Front View of the Calculator

DESCRIPTION OF A MAGNETIC DRUM CALCULATOR

BY

THE STAFF OF THE COMPUTATION LABORATORY

VE RI TAS

CAMBRIDGE, MASSACHUSETTS
HARVARD UNIVERSITY PRESS
1952

The opinions or assertions contained herein are the private ones of the writers and are not to be construed as official or reflecting the views of the Navy Department or of the naval service at large.

Distributed in Great Britain by
GEOFFREY CUMBERLEGE
OXFORD UNIVERSITY PRESS
LONDON

Library of Congress Catalog Card Number 52-8623

Printed in the United States of America

STAFF OF THE COMPUTATION LABORATORY

Howard H. Aiken

Professor of Applied Mathematics, Harvard University
Director of the Computation Laboratory

Gerrit Blaauw
Jean E. Bryant
William Burkhart
Robert J. Burns
Allen G. Christensen
Carmela M. Ciampa
Charles A. Coolidge, Jr.
Peter Elias
John T. Farren
Samuel T. Favor
Marion Fennessy
Ralph E. Frazier
Harrison Fuller
J. Orten Gadd, Jr.
Kenneth C. Hanna
John A. Harr
Robert L. Hawkins
Richard Hofheimer
Grace M. Hopper
Frank Innes
Betty L. Jennings
Theodore Kalin
Lloyd C. Kentfield
Goran Kjellberg
Marshall Kincaid
Burton Lester
Peter Lindley
Marcel Linsman
E. Edward Lucchini
Frederick G. Miller
William Minty
Benjamin L. Moore
Gosta Neovius
Glen Poorte
William Porter
Willy Pouliart
Charles H. Richards
John W. Roche
Clarence Ross
Morris Rubinoff
Heinz Rutishauser
Jacquelin Sanborn
Warren L. Semon
Theodore Singer
Dexter Smith
Leonard W. Smith
Ambrose Speiser
Peter F. Strong
Helene V. Thoman
Frank L. Verdonck
An Wang
Charles Wayne
Holly B. Wilkins
Robert E. Wilkins
Way Dong Woo

PREFACE

In January 1948, prior to the completion of the Mark II Calculator, the Bureau of Ordnance of the United States Navy requested the Staff of the Computation Laboratory to undertake an investigation of automatic digital computing machinery with particular reference to the use of electronic components. This research culminated in the construction of the magnetic drum calculator, Mark III, on which work was started during the summer of 1948. The machine was completed in March 1950, at which time it was removed to the Naval Proving Ground, Dahlgren, Virginia. Operation on a production basis was started about a year later.

The electronic and relay circuits used in the machine were designed by Benjamin LaBree Moore, Way Dong Woo, Morris Rubinoff, Charles Allerton Coolidge, Jr., Gerrit Blaauw, Marshall Kincaid, Richard Hofheimer, Charles Henry Richards, and others. In addition, valuable contributions were made by Ambrose Speiser and Heinz Rutishauser of the Eidgenössische Technische Hochschule, Zurich; Gosta Neovius and Goran Kjellberg of Matematikmaskinnämndens Arbetsgrupp, Kunglig Tekniska Högskolan, Stockholm; Burton R. Lester and Charles Wayne of the General Electric Company; Marcel Linsman of the University of Liége and Willy Pouliart of the Bell Manufacturing Company of Belgium. Charles Manneback of the University of Louvain also paid several visits to the Computation Laboratory and made numerous valuable suggestions. The electrical wiring of the machine was carried out under the supervision of William Porter. After preliminary testing at the Computation Laboratory, the machine was disassembled, transported to the Naval Proving Ground, and reassembled there under the supervision of Frederick Gaylord Miller.

The mechanical parts of the machine, including the tape read-record mechanisms and the magnetic drum units, were engineered by Robert Edward Wilkins with the assistance of Samuel Favor and Dexter Smith.

The approximation formulas used to compute the elementary functions were devised in large part by John Allen Harr. The coding of the routines to evaluate these formulas was carried out by Clarence Ross of the staff of the Naval Proving Ground, Dahlgren, Virginia.

The present volume describing the Mark III Calculator was written largely by Helene Valeska Thoman and Peter Fallis Strong. The chapter on the elementary functions was written by John Harr, and the chapter on problem preparation is in large measure due to Joseph Orten Gadd, Jr. Richard Hofheimer and Marshall Kincaid rendered much valuable assistance in the preparation of other chapters forming the book.

The illustrative figures used in this volume were drawn by Carmela Maria Ciampa, and the photographic plates were made by Paul Donaldson, photographer of Cruft Laboratory. The manuscript was prepared for publication by Jacquelin Sanborn. The films used in making the plates from which the book was printed were prepared by Paul Donaldson, assisted in part by Robert Joseph Burns. It is a pleasure to acknowledge the continual coöperation of the Harvard University Press in making possible the publication of this and other volumes of the Annals of the Computation Laboratory.

The construction of the calculator and the preparation of this volume fulfill the requirements of Task 2 of Contract NOrd 10449 between the Bureau of Ordnance and Harvard University. The Staff of the Computation Laboratory wish to express their gratitude to the Chief of the Bureau of Ordnance for the privilege of carrying out the work and publishing this volume.

Howard Aiken

Cambridge, Massachusetts
March 1952

CONTENTS

Chapter		Page
I	Organization of the Calculator. . . .	1
II	Basic Circuits.	41
III	Storage System.	62
IV	Arithmetic Units.	86
V	The Elementary Functions.	115
VI	Numerical Input and Output Devices. .	142
VII	Sequencing and Control.	189
VIII	Instructional Tape Preparation Table.	229
IX	Operation of the Calculator	264
X	Problem Preparation and Solution of Typical Examples.	272
Appendix I	Constants in Fast Storage	307
Appendix II	List of Codes	309
Index.		311

LIST OF PLATES

	Front View of the Calculator	*Frontispiece*
I	Magnetic Drum Storage Unit.	3
II	Pole Piece (exploded view).	5
III	Slow Storage Bracket.	6
IV	Typical Chassis	13
V	Instructional Storage Drum.	14
VI	Tape Read-Record Mechanisms	15
VII	Printer Control Panels.	16
VIII	Typewriters	17
IX	Power Supply Control Panels	18
X	Numerical Tape Preparation Table. . . .	19
XI	Instructional Tape Preparation Table. .	20
XII	Main Control Panel.	27
XIII	Chassis: Underside	42
XIV	Racks of Chassis.	43
XV	Cables (exploded view).	44
XVI	Component Containers (exploded view). .	45
XVII	Gear Box.	63
XVIII	Constant Storage Bracket.	69
XIX	Fast Storage Bracket.	71
XX	Magnetic Clutch	143
XXI	Magnetic Cam Unit	150
XXII	Two Tape Read-Record Mechanisms: Front View. .	152
XXIII	Two Tape Read-Record Mechanisms: Rear View. .	153
XXIV	Printer Tape Mechanism.	171
XXV	Printer Relay Panel	172
XXVI	Typewriter.	173
XXVII	Printer Control Panel (close-up). . . .	174
XXVIII	Cam Unit.	175

CHAPTER I

ORGANIZATION OF THE CALCULATOR

Mark III Calculator is an electronic computing machine in which numerical quantities and coded operating instructions are stored with the aid of magnetic pulse recording techniques. The calculator's internal number storage system is composed of eight aluminum alloy cylinders, Fig. 1.1 and Plate I, gear-connected, and driven at approximately 6900 rpm by a 3450 rpm wound rotor induction motor. The surfaces of the storage drums are coated with a thin film composed of finely divided magnetic oxides of iron suspended in a plastic lacquer, and applied to the drums with an artist's air brush. Recording and playback pole pieces, Fig. 1.2 and Plate II, mounted on stationary brackets in close proximity to the surfaces of the drums, not only provide a means of recording on and reading from the magnetic medium, but assist in the selection of stored quantities as well.

The pole pieces consist of split Mu-Metal cores, each half being composed of twelve laminations 0.006 inch thick, cemented together and stacked to a total thickness of approximately 0.1 inch. The cores are wound with either one 250-turn coil or two 300-turn coils as required by playback and recording pole pieces, respectively. The active tips of the two halves of the core are carefully lapped and polished. The air gap thus formed supplies the leakage field by which the presence of sharp current pulses in the coils may be recorded as short dipoles in the rapidly moving magnetic medium as it passes under the pole piece tips. In order to insure a high degree of reliability in operation, only ten pulses are recorded per inch of circumference on the surfaces of the number storage drums, and twenty pulses per inch on the surface of the drum reserved for the storage of operating instructions. This higher density in the case of instructional storage, in combination with the larger circumference and slower speed of the instructional storage drum, causes the pulse repetition rate to be the same for both drum systems, thus permitting the same basic design for playback and record units.

The pole pieces are mounted in machined brass or molded plastic housings, accordingly as magnetic shielding is or is not necessary due to the close proximity of recording and playback units. Since the housings are approximately 1/2 inch thick, they are mounted on stepped spiral brackets, Plate III, having lands separated by 1/8 inch measured along the axis of a drum. This construction permits economical use of the

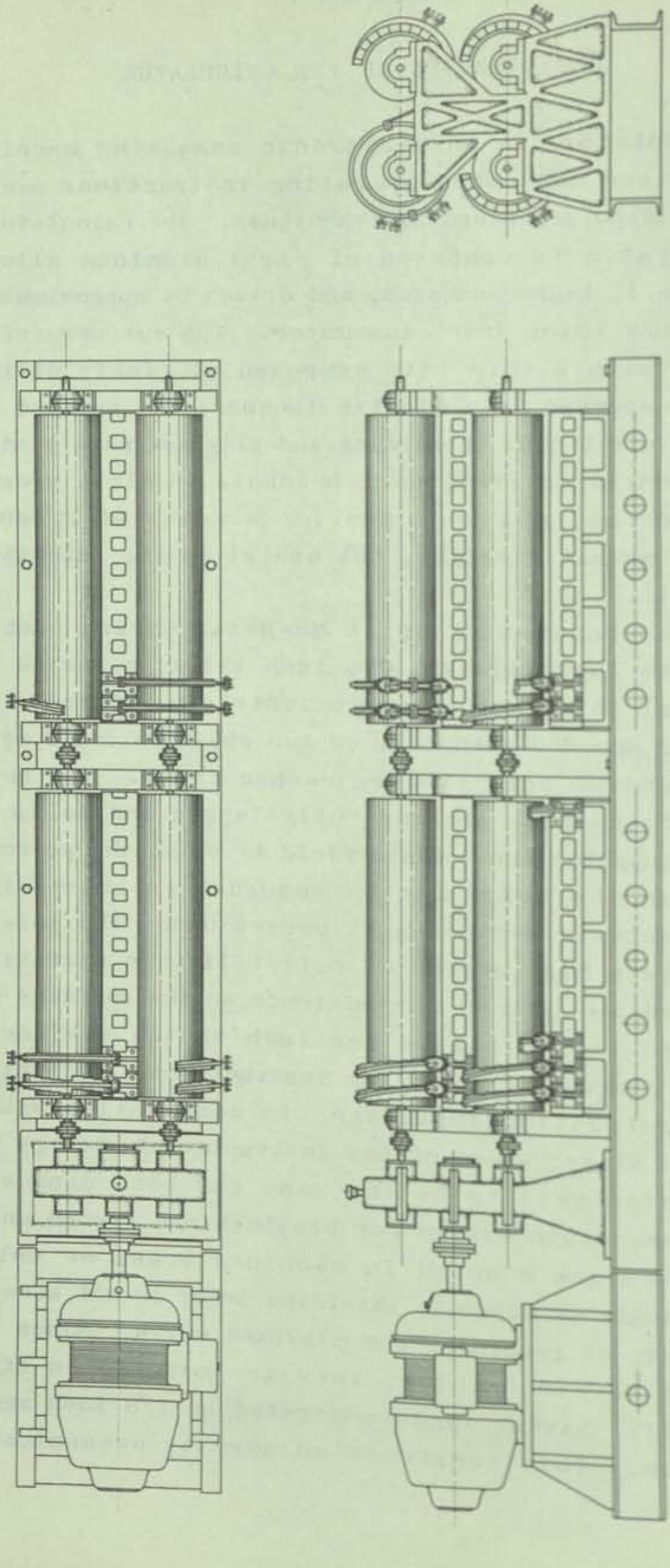


Fig. 1.1—Magnetic drum storage unit.

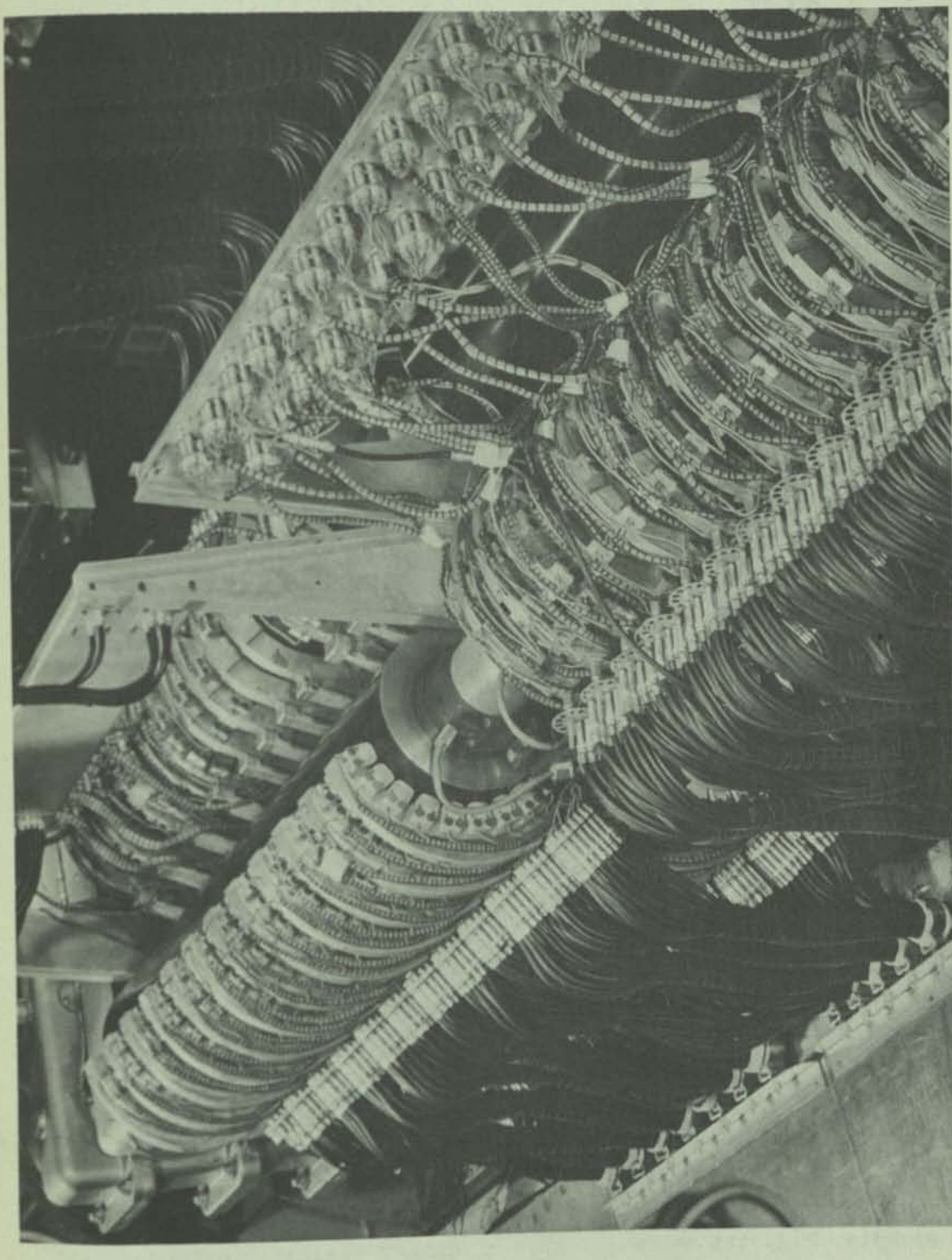


Plate I *Magnetic Drum Storage Unit*

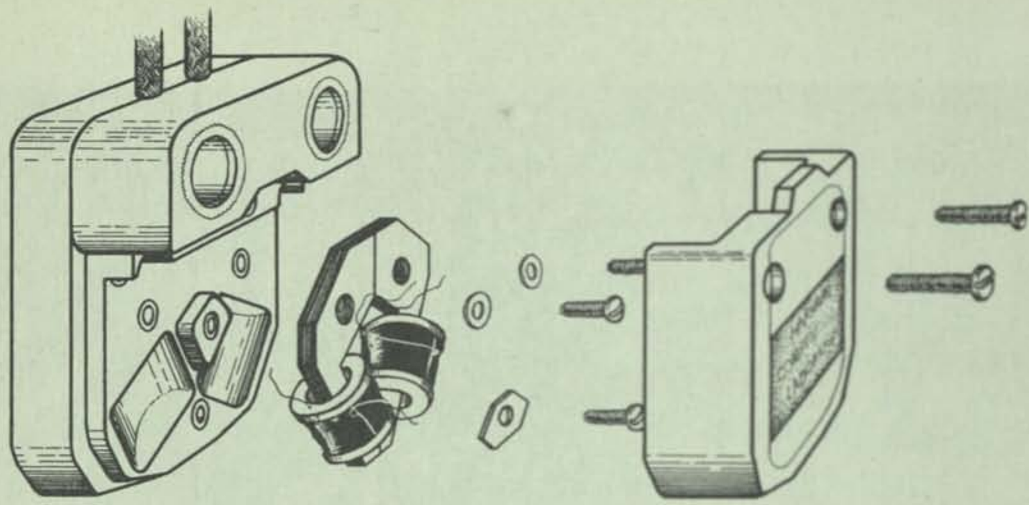


Fig. 1.2—*Exploded view of a pole piece.*

drum surfaces, and provides 0.025 inch between recording channels to prevent cross-talk.

The two coils of a record pole piece are wound in opposition with a common ground connection between them. A sharp current pulse applied to one of the coil input terminals causes a positive or negative magnetic dipole to be recorded in the magnetic medium, depending on the direction of the current. In either case, a dipole recorded by the other coil will be of the opposite polarity for the same direction of current flow. A recorded dipole, in passing under a playback pole piece, will generate an electromotive force at the output terminal of its coil. The resulting pulse forms, a positive peak followed by a negative peak, Fig. 1.3(*a*), or a negative peak followed by a positive peak, Fig. 1.3(*b*), are taken to represent the binary digits 0 and 1, respectively. Thus, of the two coils provided on a record pole piece, one is reserved for recording 0's, and the other for recording 1's.

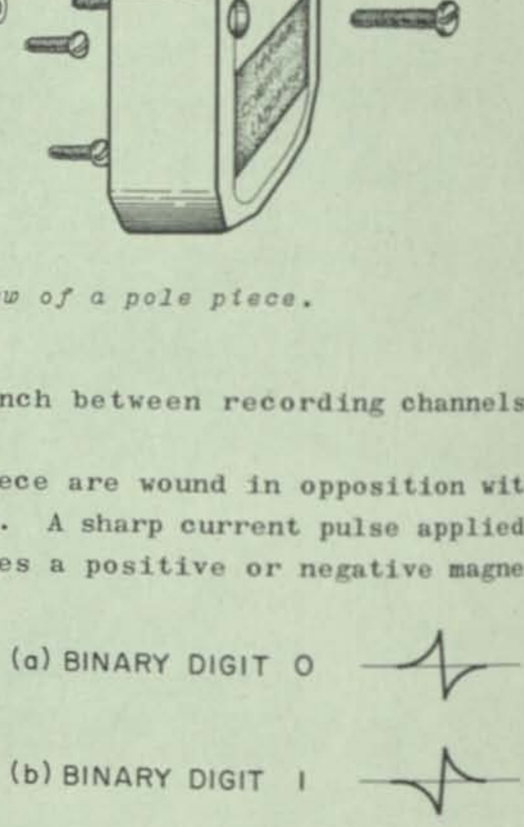


Fig. 1.3—*Wave forms of played back signals.*

Signals coming from a playback pole piece are delivered to a playback unit, where they are amplified and converted into a rectangular envelope. For example, Fig. 1.4 shows a series of binary digits, 0101100111, (*a*) at the terminal of the playback coil, and (*b*) at the output of the playback unit. If the played back information is to be rerecorded, the output of

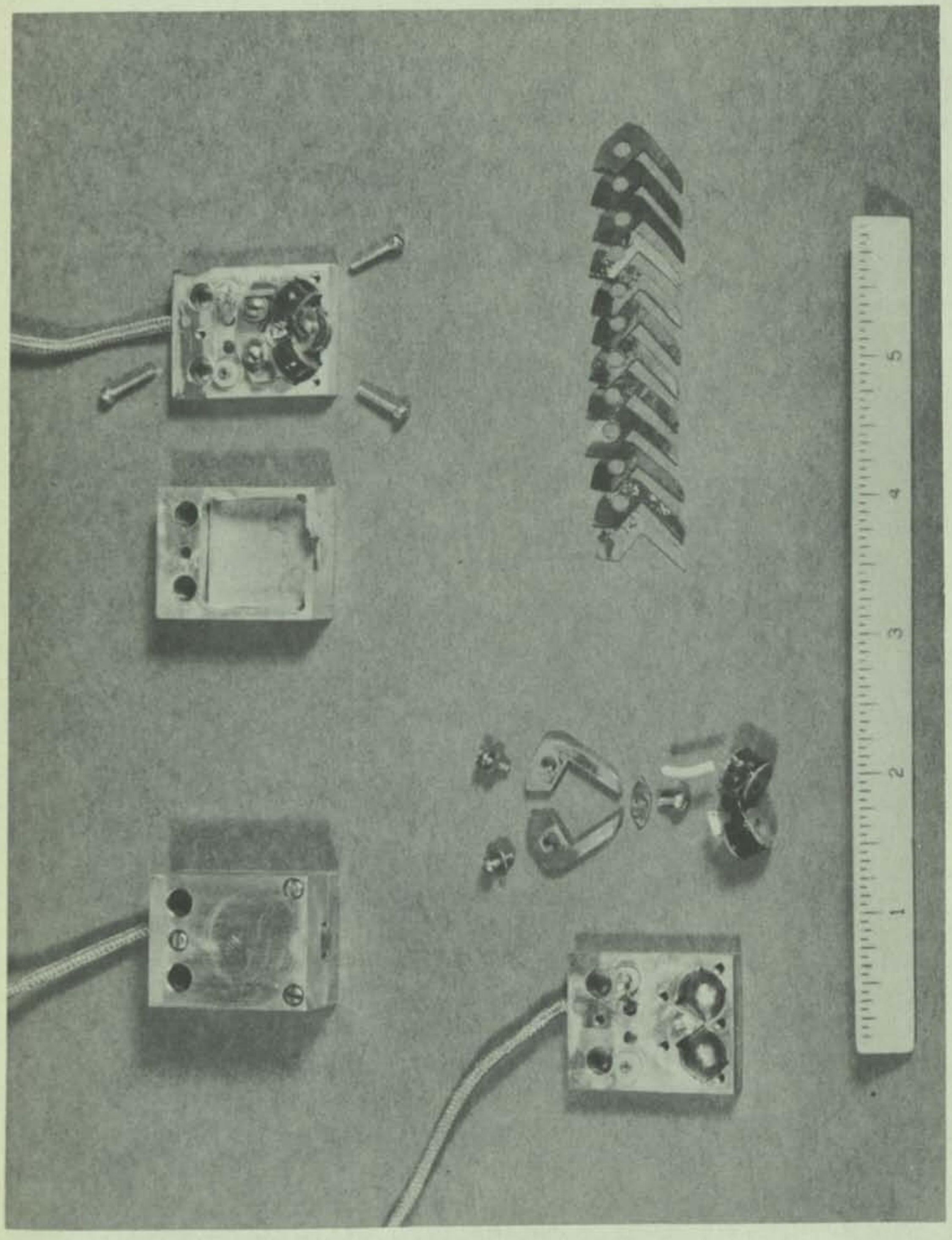


Plate II *Pole Piece (exploded view)*

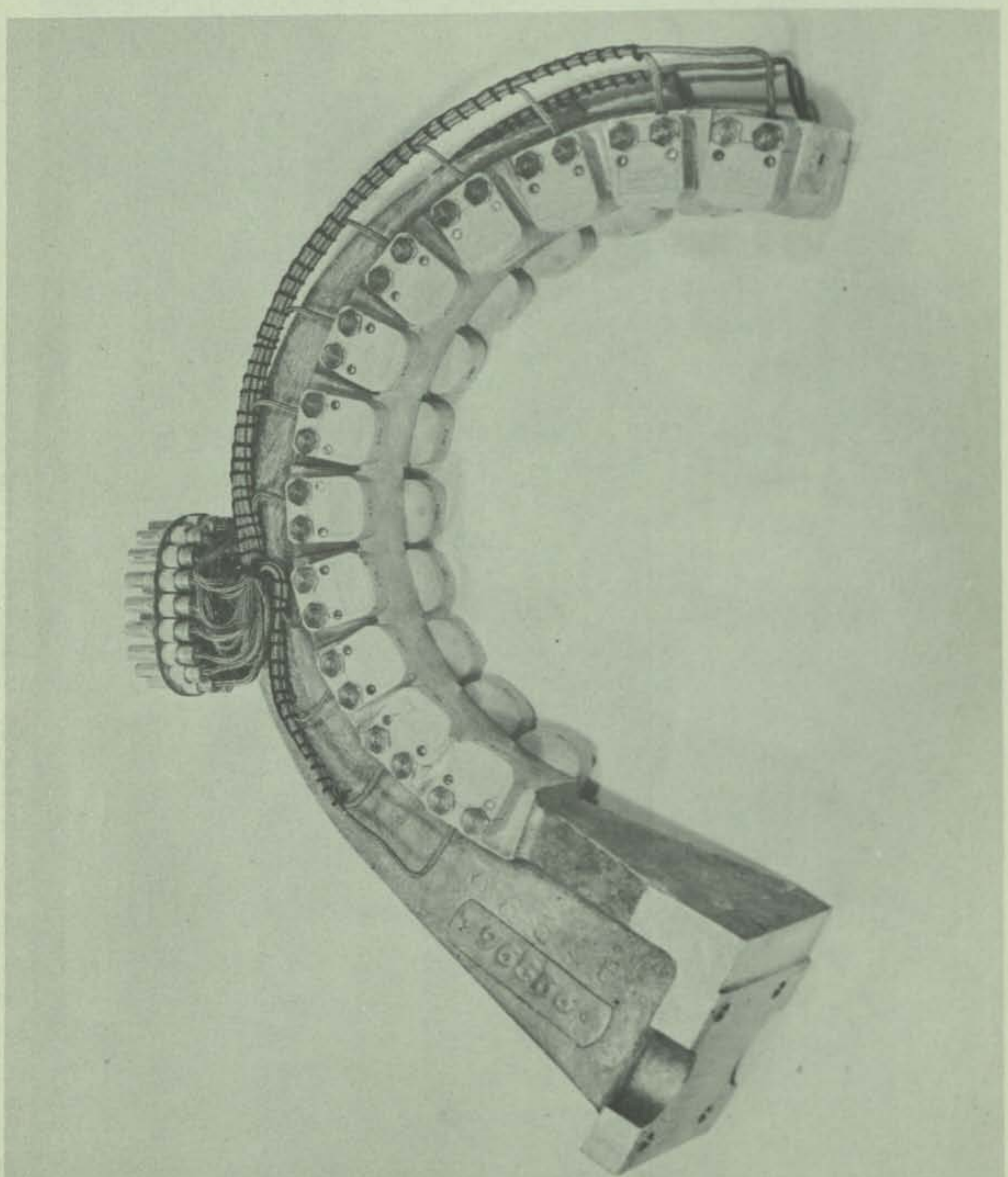


Plate III *Slow Storage Bracket*

the playback unit is delivered to a record unit, which gates or modulates a set of standard record pulses, and delivers them to the two coils of a record pole piece, as shown in Fig. 1.4(*c*).

Magnetic drum storage has the pleasing characteristic that information once recorded remains on the drum surface indefinitely, unless deliberately altered. Moreover, provided that accurately timed record pulses

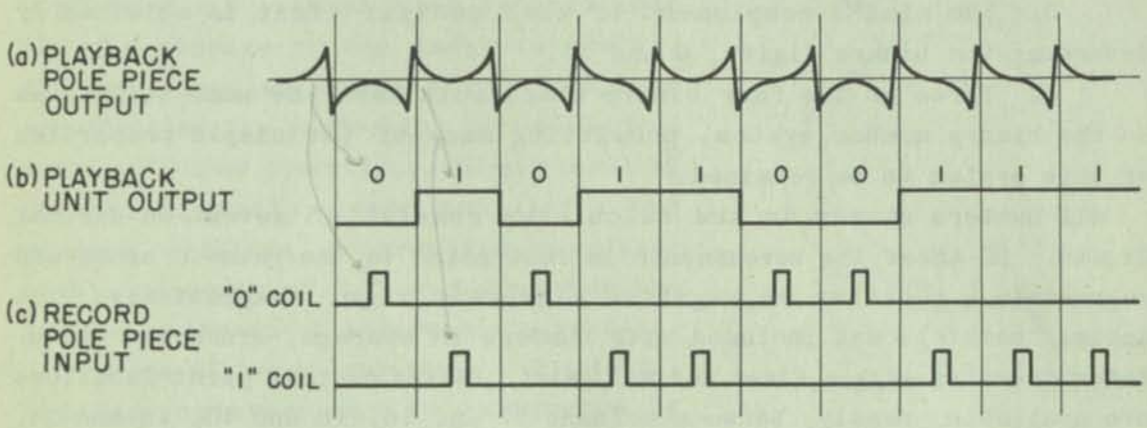


Fig. 1.4—*Typical series of wave forms.*

are used, it is not necessary to erase the surface, by demagnetizing it, in order to change a binary digit from 0 to 1, or vice versa. This is because the record current is sufficient to saturate the medium in either direction, with the result that the recorded dipoles either remain unchanged or suffer a complete reversal of polarity when subjected to a record pulse of like or opposite sign, respectively. Thus it becomes possible to record a succession of quantities in a given channel with complete assurance that the last information delivered to the storage drums will be stored there independent of any previously recorded quantities.

TABLE 1.1

Decimal digits 2*,4,2,1 notation

	2*	4	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	0	1
8	1	1	1	0
9	1	1	1	1

The accurately timed pulses required in the operation of the record and playback units are themselves obtained from the storage drums, where channels of evenly spaced dipoles have been permanently recorded.

In order to store numerical information using only the binary digits 0 and 1, a coding system has been adopted in which each decimal digit is represented by four binary digits with the weights 2*,4,2,1, as shown in Table 1.1. Here the asterisk (*) serves to distinguish between the two binary digits which are of equal weight. Table 1.1 reveals the following desirable features:

1. Examination of the 1 component alone indicates whether a digit is odd or even.

2. Examination of the 2* component alone indicates whether a digit is less than 5, or 5 or greater.

3. The nine's complement of each decimal digit is obtained by inverting the binary digits, 0 and 1.

4. Three of the four binary components have the same weights as in the binary number system, permitting many of the simple properties of this system to be retained.

All numbers stored in the calculator consist of seventeen decimal digits. Of these the seventeenth is restricted to the value 0 or 9, and represents a positive or negative algebraic sign, respectively. The decimal point is not included with numbers in storage, since the calculator operates with a fixed decimal point. Seven decimal point locations are available, namely, between columns 17 and 16, 16 and 15, 15 and 14, 13 and 12, 12 and 11, 10 and 9, and 1 and 0. Before starting the calculator on the solution of a problem, one of these locations is chosen by manually presetting the decimal point controls. The choice of an operating decimal point, however, has no effect on the storage system, but merely alters the output connections of the multiply unit.

Since four binary digits are required to represent a decimal digit, four parallel channels on a storage drum provide a single channel for the storage of decimal digits. All stored quantities enter and leave the storage drums serially. The lowest order digit, preceded by two non-informative digits, is followed by the digits in the second, third, ..., sixteenth columnar positions, the algebraic sign, and a third non-informative digit, in that order. The three non-informative digits are supplied for control purposes. Hence each stored quantity is represented by a time space of twenty pulses. For example, the numbers π and $-e$ are stored as follows when the operating decimal point lies between columns 9 and 10:

			17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1		
	2*	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	0	0
π	4	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	1	0	0
	2	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	0	1	0	0	0
	1	0	0	0	0	0	0	0	0	1	1	0	1	1	1	0	0	1	0	0	0
	2*	0	1	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	0
$-e$	4	0	1	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	0
	2	0	1	0	0	0	0	0	0	1	0	0	1	1	1	0	1	1	1	0	0
	1	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0

With a drum diameter of eight inches and a density of ten pulses per inch, a total of 240 binary digits can be stored around the drum in one channel. The periphery is divided into twelve segments, called number times, each of which contains twenty digit or pulse times, Fig. 1.5. Ten of the segments, numbered 0-9, constitute storage registers, as each of them is available for the storage of one complete number. The remaining two segments are blank, and provide time for performing various arithmetic and other operations. Twenty parallel decimal channels, each containing ten storage registers, provide the calculator with a capacity of 200 registers for the internal storage of intermediate results.

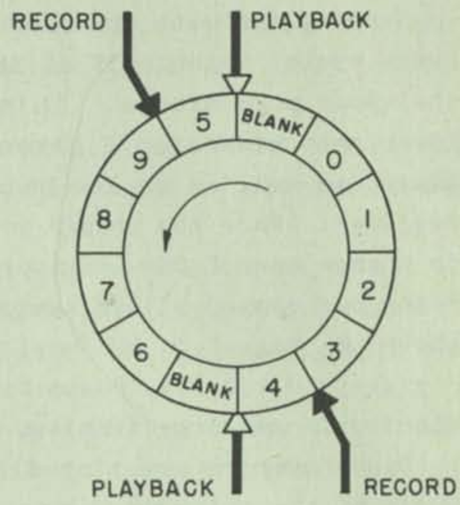


Fig. 1.5—*Cross section of storage drum.*

Any one of the calculator's 200 internal storage registers may be designated by specifying the channel in which it lies and the time phase in which it passes under a pole piece. In order to simplify references to storage registers, the channels are identified by twenty letters of the alphabet. As previously stated, the time phase or number time is denoted by one of the digits, 0-9. Then the storage register located, for example, on channel x in time phase 6 may be referred to as x_6. Later it will be shown that the preparation table for instructional tapes capitalizes on the representation described here, and greatly simplifies the preparation of problems for introduction to the machine.

The use of separate record and playback pole pieces, indicated as black and white arrowheads, respectively, in Fig. 1.5, increases the speed and flexibility of the calculator by making it possible to record in and read from a channel at the same time. Since a storage register is accessible only at the times when it is passing under a record or a playback pole piece, the speed of the calculator is effectively doubled by mounting two record and two playback pole pieces on each of the binary channels included in the group of twenty decimal channels. This arrangement is indicated schematically in Fig. 1.5, where one pole piece, connected to a four-wire bus, represents the four parallel pole pieces required for the four binary components. Thus each storage register is scanned by both a record and a playback pole piece during every half-revolution of the storage drum. For a drum speed of 6900 rpm, the time required for a half-revolution is 0.00435 second, or roughly four milliseconds, and this, the time of access to a storage register, is the periodic time during which the machine carries out one cycle of its operation.

Sensing circuits within the machine are provided to determine, during every cycle, which half of the storage drum periphery is passing under the upper pole pieces. This is because the choice between upper and lower pole pieces on a given channel necessarily depends on the drum phase, as well as on the half of the periphery containing the desired register. Once the proper pole piece has been selected, it remains only to choose one of five number times, 0,1,2,3,4, the same number times being associated with diametrically opposed registers. The arrangement shown in Fig. 1.5, of registers 6,7,8,9,5, in that order, opposite registers 0,1,2,3,4, respectively, has been adopted in order to simplify electronic sensing circuits.

Since numbers are played back from storage serially, the arithmetic units of the calculator are serial in their operation. Two parallel four-wire busses, *A* and *B*, carry selected quantities from storage to an arithmetic unit, while a third bus, *C*, returns the computed result thereto. This bus configuration was adopted since the elementary arithmetic operations provide one computed result from two operands. Note

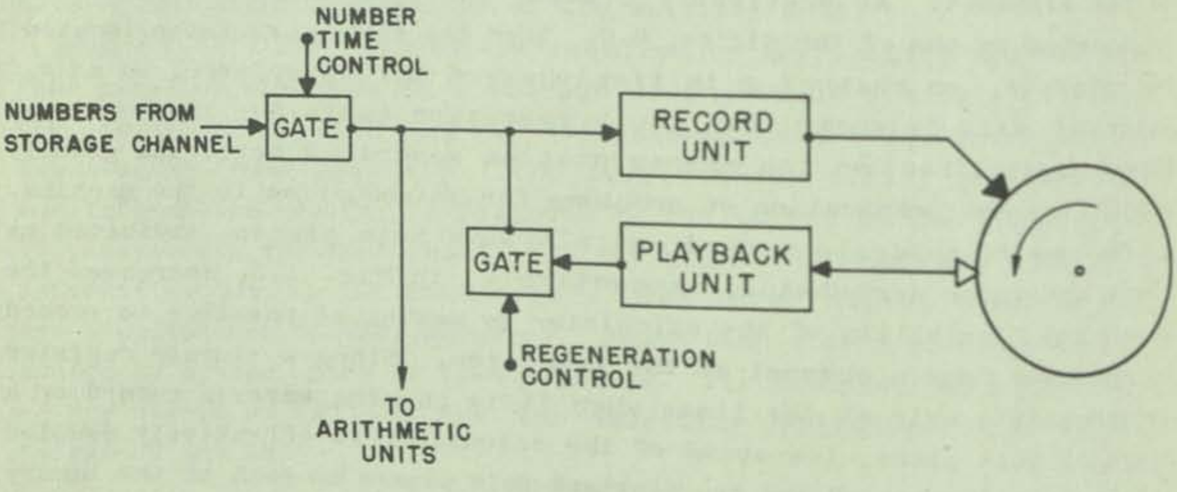


Fig. 1.6—*Diagram of regeneration channel.*

that two operands selected from storage will not, in general, be stored in the same time phase. Therefore, numbers going to an arithmetic unit are first recorded at their number times on decimal channels referred to as the *A* and *B* transfer channels. Each of the four binary components of a transfer channel, Fig. 1.6, has two pole pieces so spaced that as the last digit of a number is recorded, the first recorded digit reaches the playback pole piece, and can be gated back to the record unit. Thus, a number once introduced to a transfer channel is rerecorded or regenerated around the channel. Clearly, any two quantities in storage, regardless of their time phase, can with the aid of the *A* and *B* transfer channels be made available to an arithmetic unit as early as the beginning of 4 number time in any cycle.

A similar *C* transfer channel regenerates the output of the arithmetic unit, in order that computed results may be returned to storage at any desired time phase.

All of the foregoing elementary ideas are to be treated in detail in Chapter II. However, enough has so far been said, regarding magnetic recording techniques, to proceed with an over-all description of the calculator.

The floor plan, Fig. 1.7, indicates the physical layout of the machine. The magnetic drum storage unit is in the rear projection of the calculator. Computing and control circuits, consisting of thermionic vacuum tubes, electromechanical relays, and associated components, are arranged on chassis, Plate IV, and mounted in racks, which stand behind the front panel and on both sides of the drum unit. Near the center of the front panel is a tape mechanism for reading coded instructions from magnetic tape to the drum reserved for instructional storage, Plate V. At the right are the eight tape read-record mechanisms, Plate VI, by means of which numbers are transferred between the internal drum storage and external tape storage, as may be required in the solution of a problem. At the left are the five independent printer control panels, Plate VII, each of which operates one of the typewriters in front of the calculator, Plate VIII. Reels of tape containing computed results are mounted on the tape reading mechanisms, and manual adjustments are made to determine the typography of the printed page. The power required to operate the machine is obtained from motor generator sets, controlled from the rear of the right wing, Plate IX. The magnetic tapes on which both numerical data and instructions are introduced to the calculator are prepared on two auxiliary units, the numerical tape preparation table, Plate X, and the instructional tape preparation table, Plate XI.

The internal organization of the calculator will be described with the aid of the diagram, Fig. 1.8. The heavy lines indicate four-wire busses for the transmission of the four binary components of a decimal digit. All the gates are controlled by voltages derived periodically from the instructional storage drum. These govern the operation of the

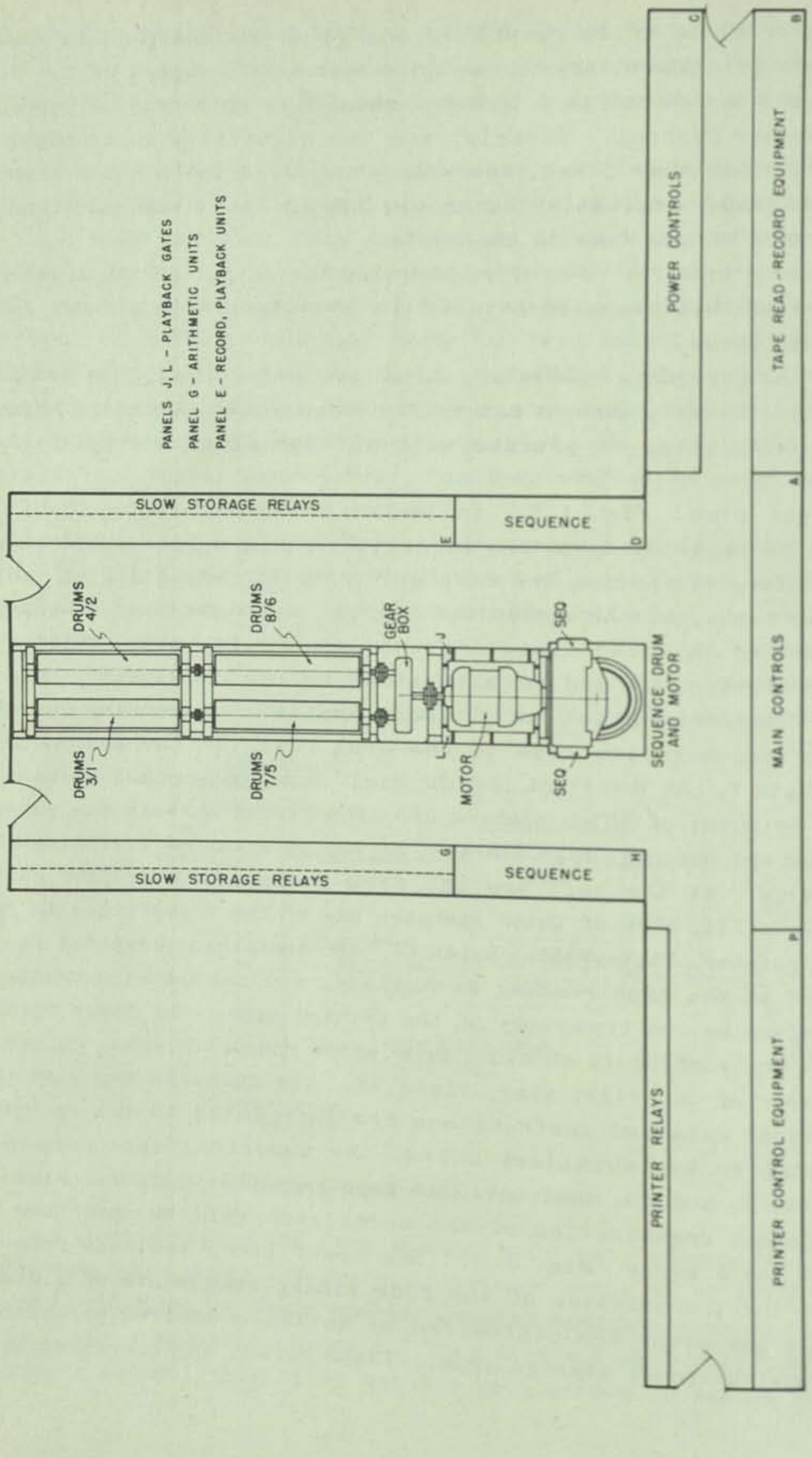


Fig. 1.7—*Floor plan of the calculator.*

Plate IV *Typical Chassis*

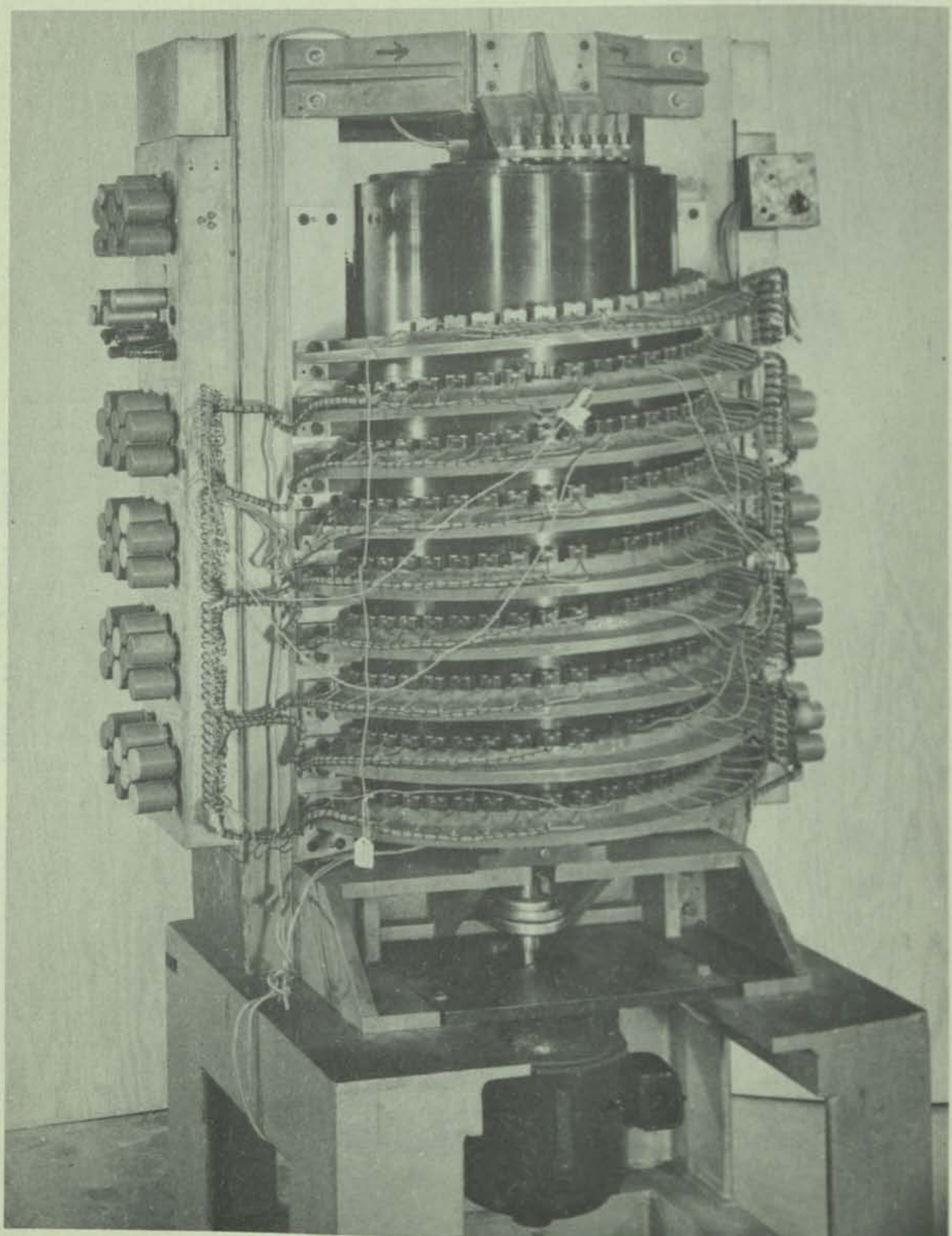


Plate V *Instructional Storage Drum*

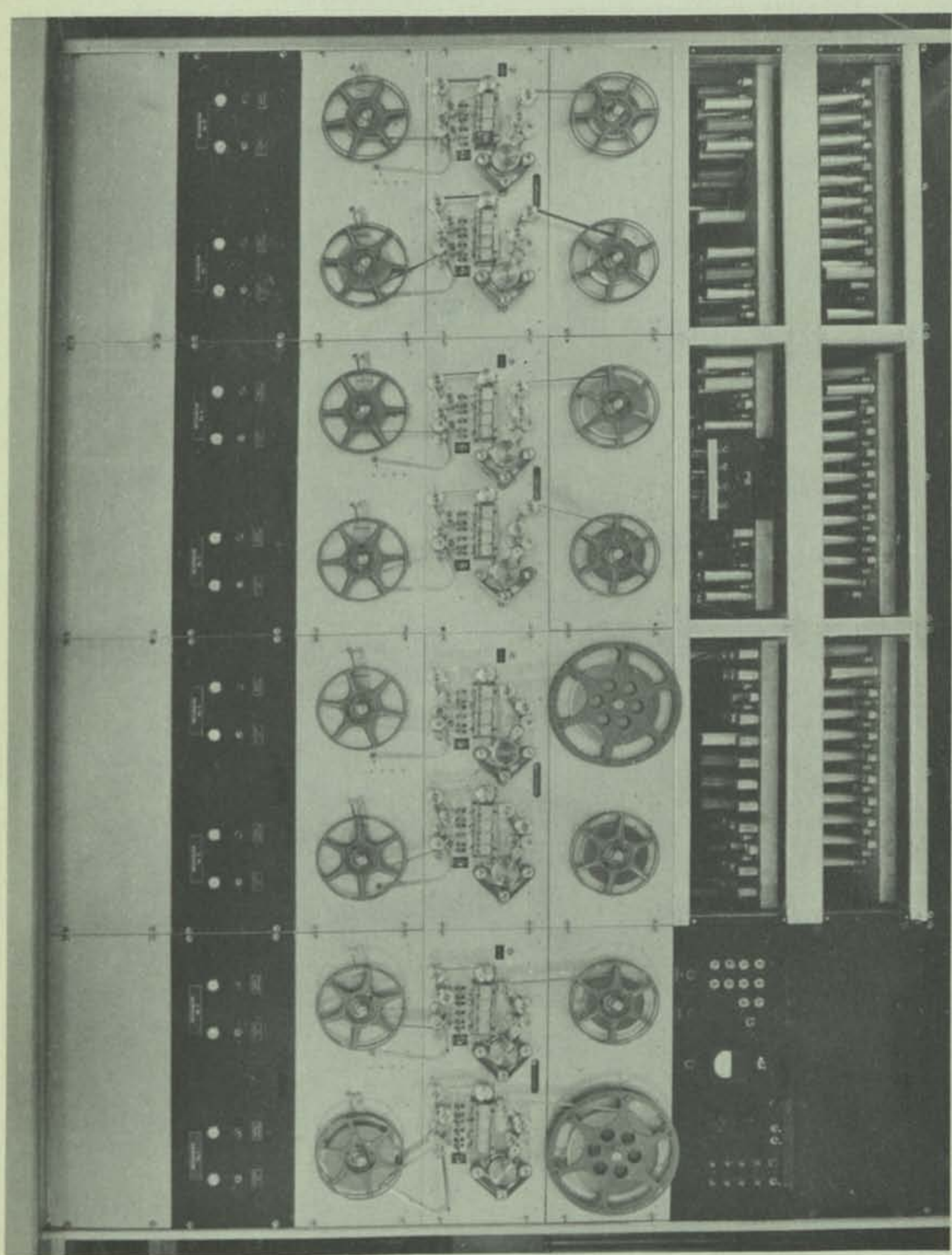


Plate VI *Tape Read-Record Mechanisms*

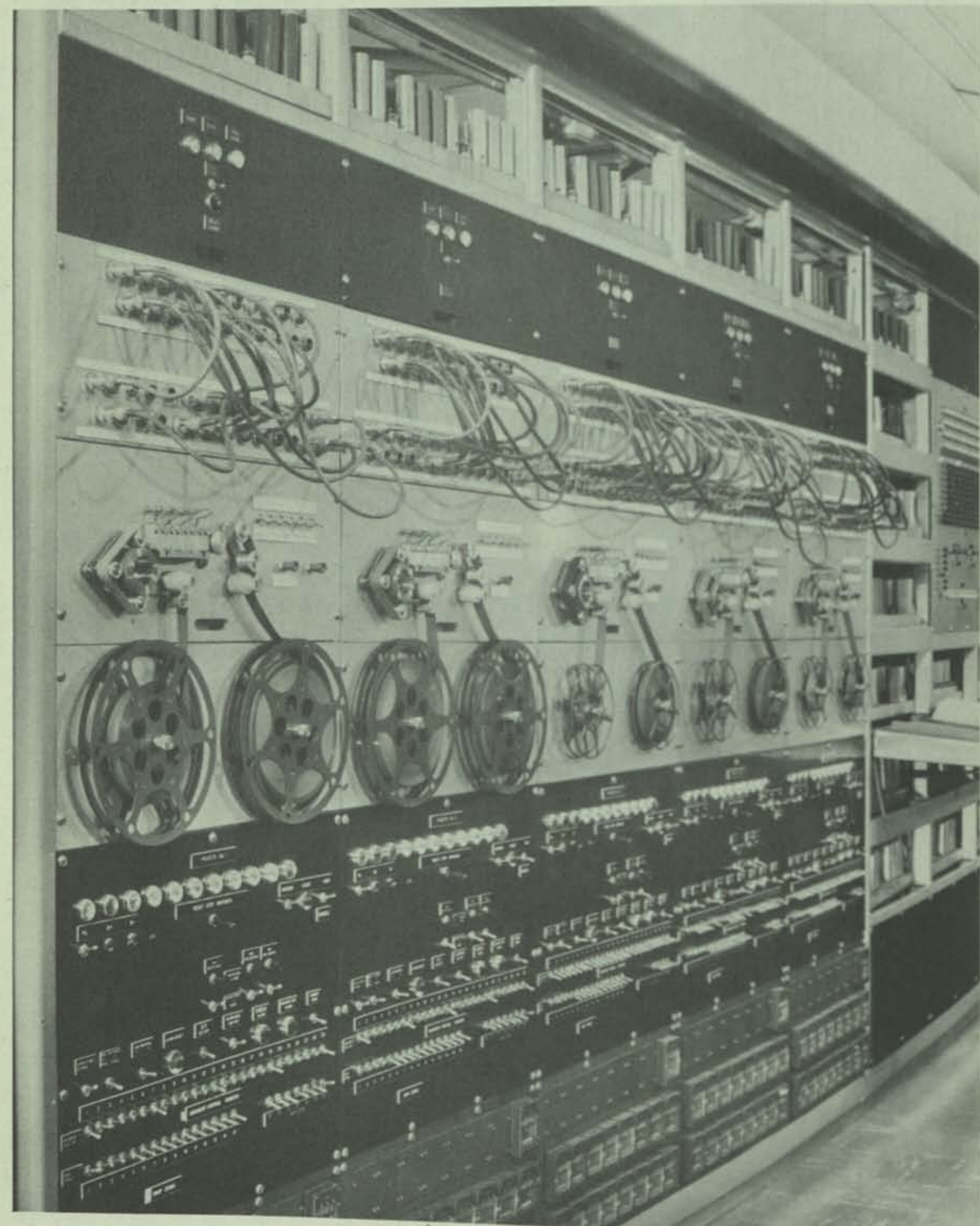


Plate VII *Printer Control Panels*

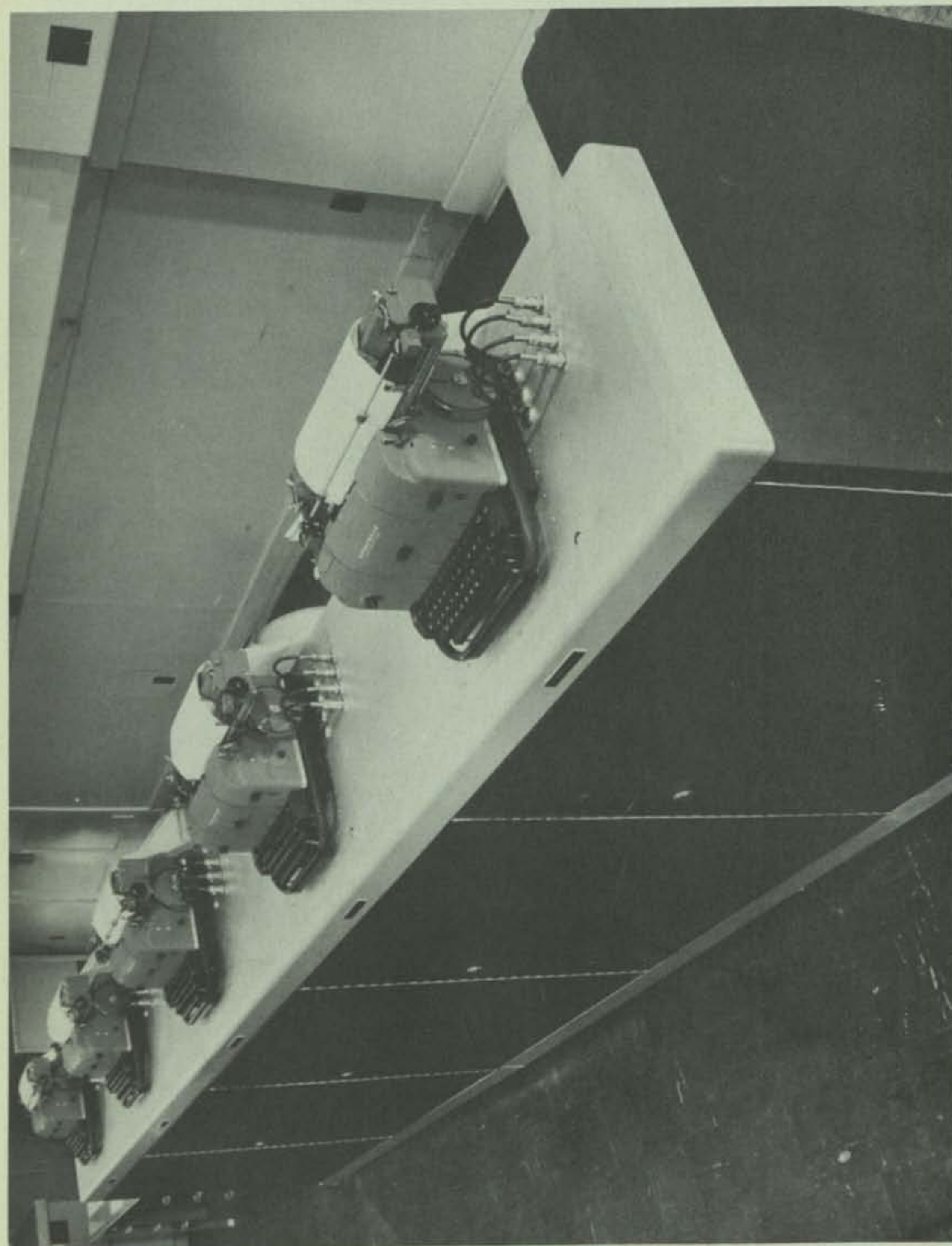


Plate VIII *Typewriters*

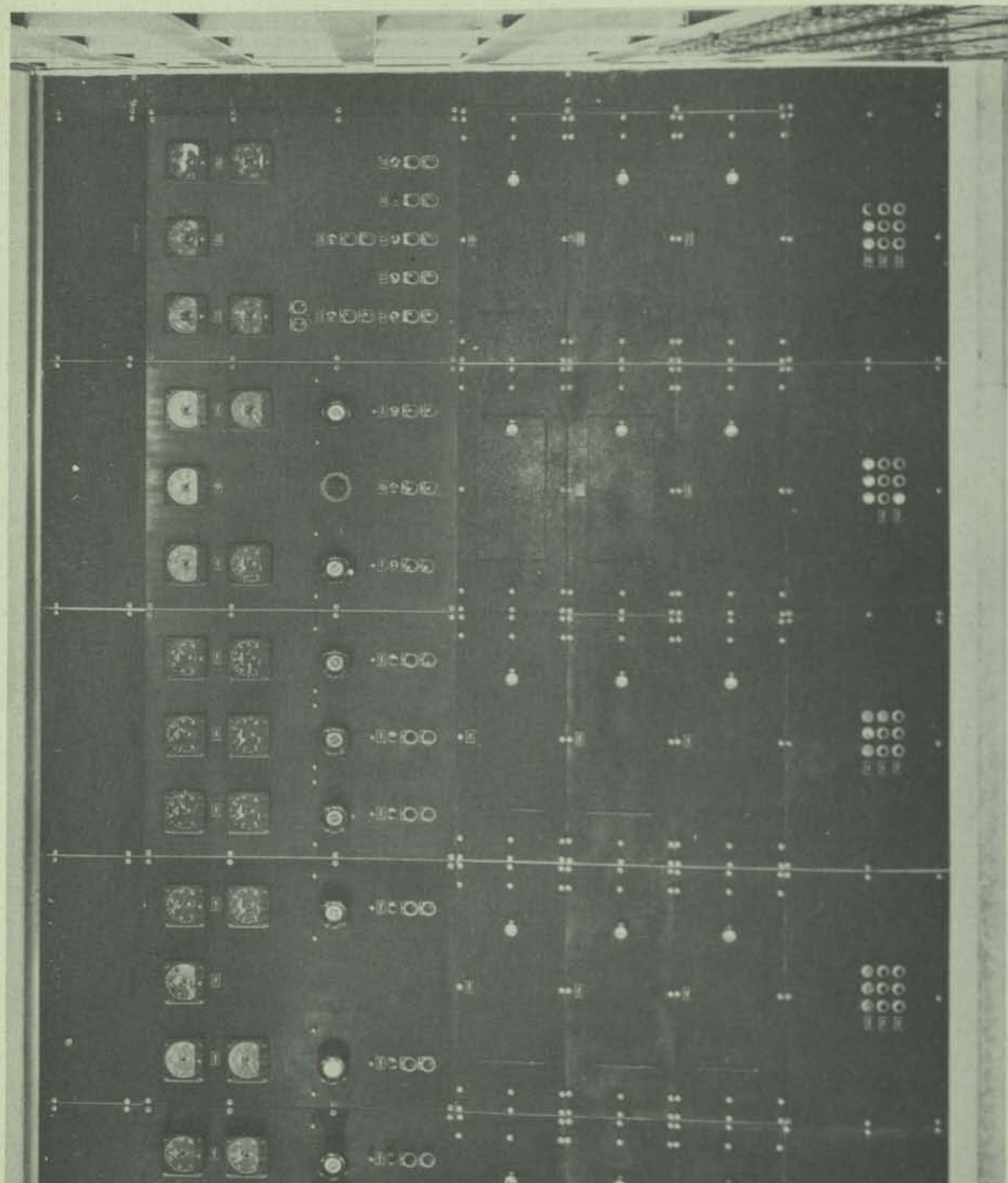


Plate IX Power Supply Control Panels

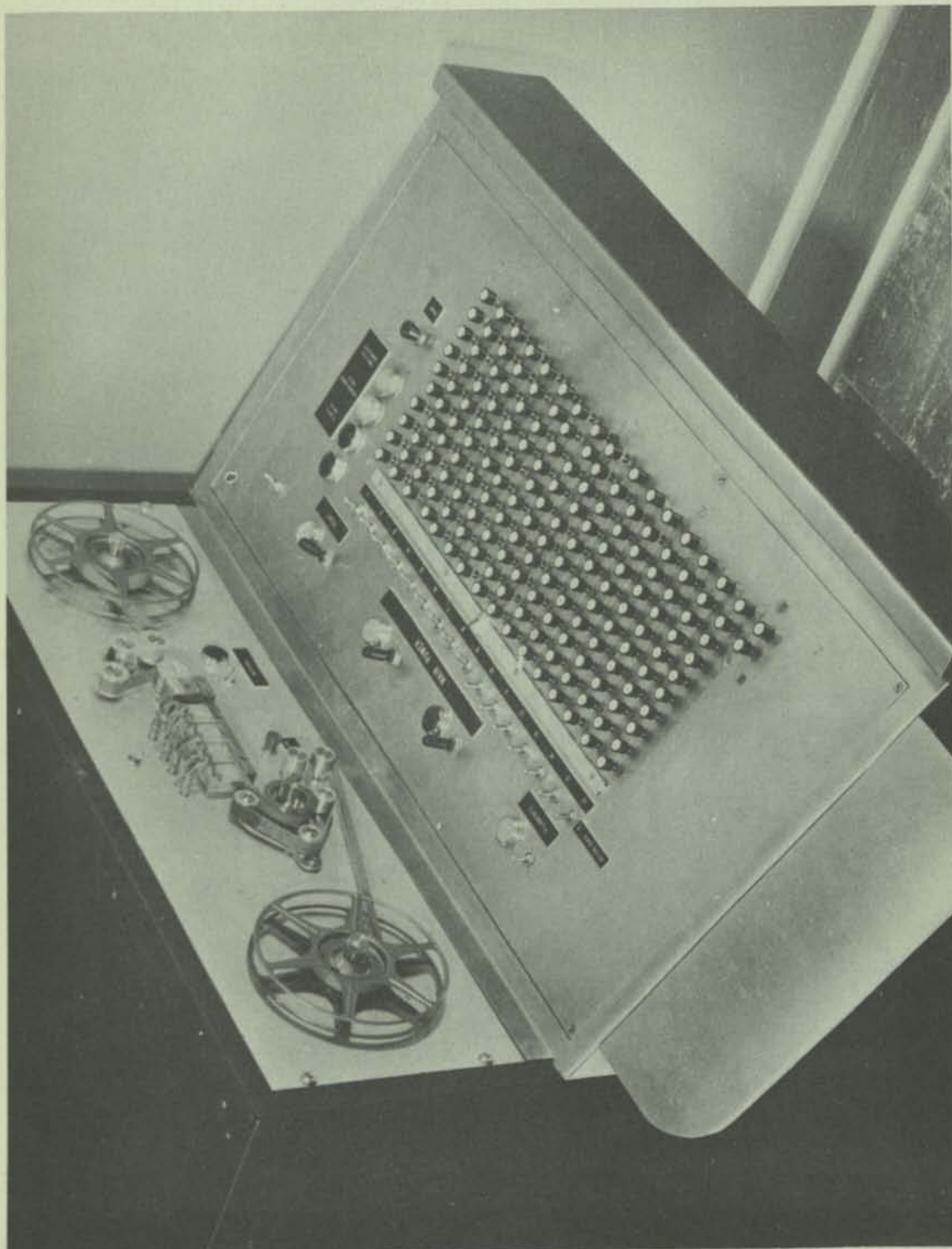


Plate X *Numerical Tape Preparation Table*

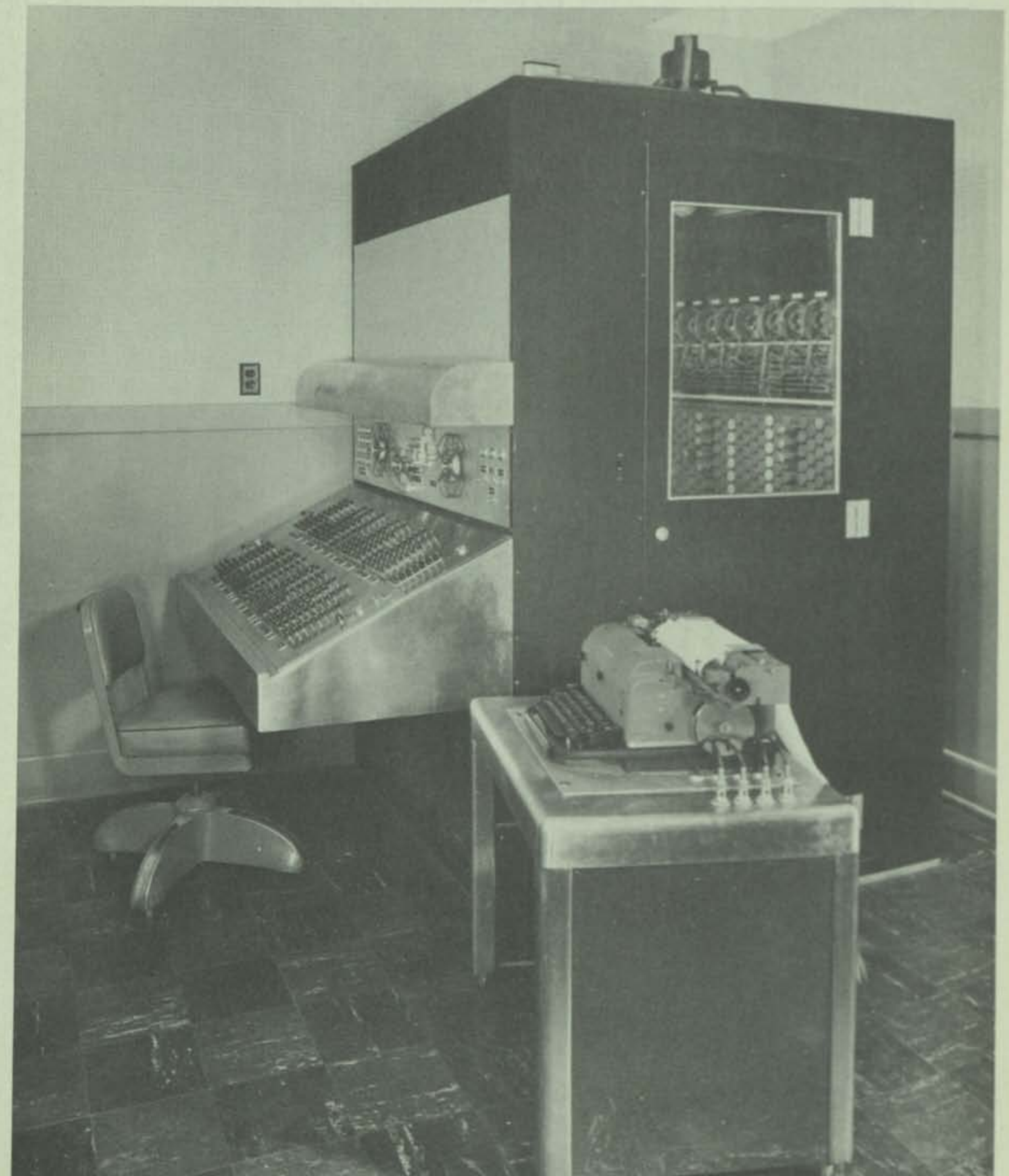


Plate XI *Instructional Tape Preparation Table* calculator by directing the flow of numbers along the A,B,C bus system.

The units shown in the diagram provide the following facilities:

1. *Twenty channels for the storage of* 200 *intermediate results.* The channels, indicated in the upper right portion of the diagram, are identified by the letters $a,b,c,d,e,f,g,h,p,q,r,s,t,u,v,w,x,y,z,\gamma$. Associated with the record pole pieces on these channels are individual record units, all fed in parallel by the C bus and supplied with record pulses by the control circuits, according to the coded instructions. The playback pole pieces, on the other hand, have connections to A and B selection gates, through which signals are delivered to A and B playback units. These, in turn, send on the information, amplified and converted into a rectangular envelope, to the number-time gates leading to the A and B transfer channels. Thus because the signals from the playback pole pieces are gated directly, just two decimal (eight binary) playback units are required instead of one for each pole piece.

2. *One constant register channel*, shown to the left of channel a in the diagram. The ten numbers recorded here from a manually operated set of switches are generally increments of the independent variable, check tolerances, and other parameters. Registers on this channel are also used under manual control to read starting values into the storage registers of the calculator and to store values required in testing. As indicated in the diagram, numbers may be recorded in this channel only from the switches, but playback connections to both the A and the B busses are provided.

3. *Fifteen constant channels*, indicated in the upper left of the diagram. These channels, numbered 1-15, store 150 permanently recorded quantities, most of which are needed in the subroutines for the computation of the functions x^{-1}, $x^{-1/2}$, $\log_{10} x$, 10^x, $\cos x$, and $\tan^{-1} x$. There are no record pole pieces on the constant channels since the numbers were recorded during the construction of the calculator and should not be altered in any way. Two playback pole pieces on each channel make the constants available within any cycle, but connections to the B bus only are provided. However, as will be seen later, this in no way restricts the coding for the function subroutines.

The storage channels thus far described are all located on two of the eight storage drums. Since these channels are electronically selected for both record and playback, they are referred to as "fast" storage channels, having registers accessible in approximately four milliseconds.

4. *Four hundred channels for interpolational storage*, indicated

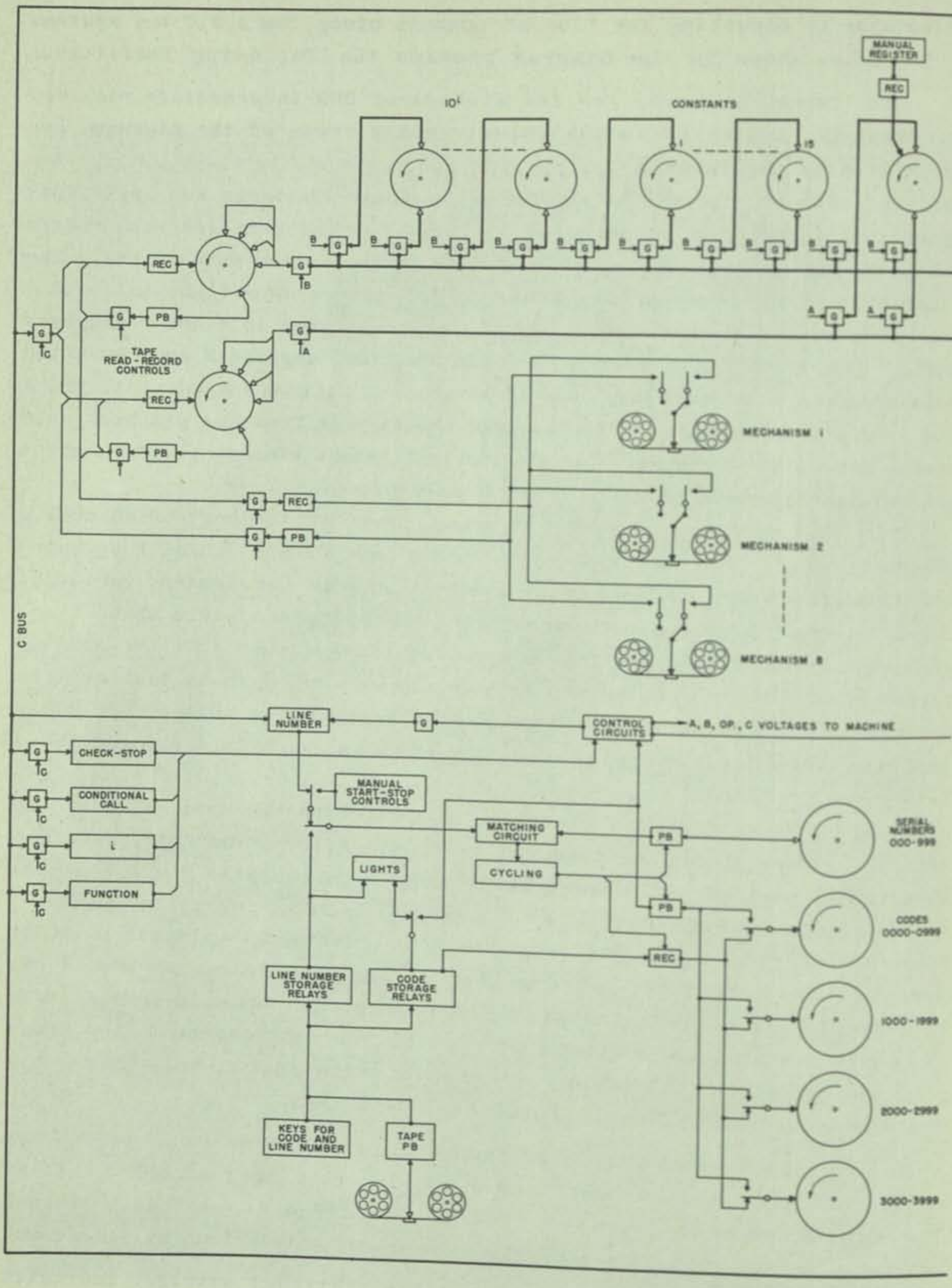


Fig. 1.8—*Over-all diagram of the calculator.*

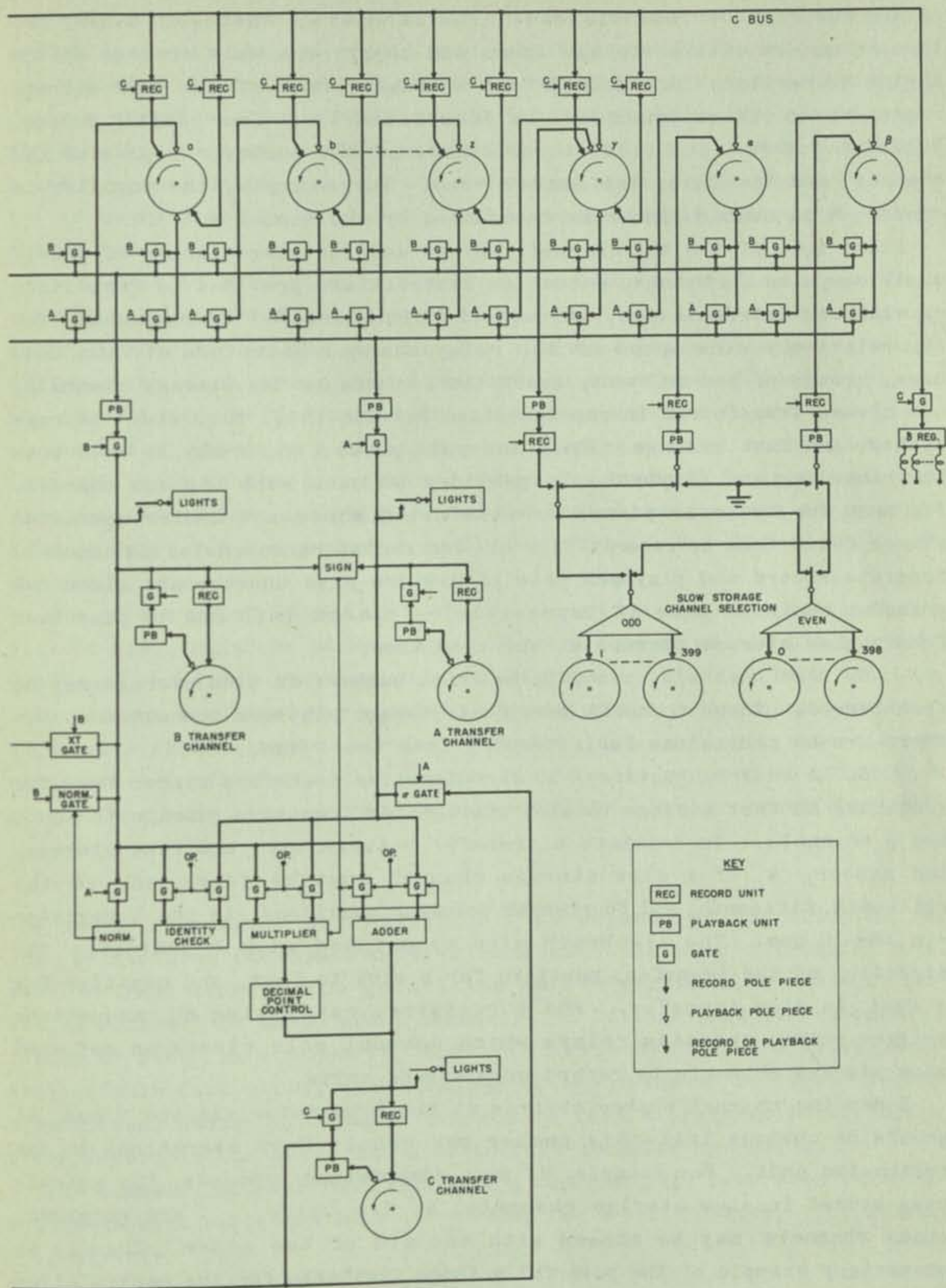


Fig. 1.8 (continued)

at the far right of the diagram. The channels, numbered 0-399, are located on six of the storage drums and comprise a bulk storage system having connections to and from fast storage. Each of the 4000 storage registers on these channels is identified by a four digit number, $0000 \leq r \leq 3999$. The first three digits of the number r represent the channel, and the last, the number time. For example, the register in channel k at number time i is designated by the number $r = 10k+i$.

Since some of the 400 channels may be used to store tables of previously computed functions, selection controls are provided to facilitate curvilinear interpolation, through the ninth order. To compensate for the relatively slow speed of the relay channel selection circuit used here, groups of ten or twenty quantities, on one or two storage channels, are always transferred in one operation between this, the "slow" storage system, and fast storage. Only one pole piece, which may be used both for recording and playback, is provided on each slow storage channel. The need for two pole pieces is eliminated since a transfer operation always reads from or records into all ten registers on a selected channel. Separate record and playback pole pieces are also unnecessary since the transfer controls make it impossible to record into and to play back from a slow storage channel at the same time.

5. *Two channels*, α *and* β, to which numbers in slow storage may be transferred. These channels have fast storage playback connections, but there are no provisions for recording from the C bus.

6. *A control register*, δ, for directing transfers either from the γ channel in fast storage to slow storage, or from slow storage to the α and β channels. To initiate a transfer between slow and fast storage, the number, k, of a slow storage channel must be delivered, in the sixteenth, fifteenth, and fourteenth columnar positions, to the δ register via the C bus. The algebraic sign associated with k indicates the direction of the transfer, positive for a slow to fast, and negative for a fast to slow transfer. The δ register, consisting of vacuum-tube trigger pairs, operates relays which connect pole pieces on selected slow storage channels to record or playback units.

Since the channel number arrives at the δ register via the C bus, it should be obvious that this number may result from operations in the arithmetic unit. For example, if as a computation proceeds, the quantities stored in slow storage channels, k, $k+c$, $k+2c$, ... are required, these channels may be chosen with the aid of the adder. This is an elementary example of the powerful methods available for the manipulation of the slow storage system.

When a positive slow storage channel number, $+k$, is delivered to the δ

register, for a slow to fast transfer, the kth and $(k + 1)$st channels are chosen, with the aid of odd-even selection pyramids. Pole pieces on the selected channels are connected to playback units. These, in turn, deliver signals to record units on the α and β channels. When the command to transfer is given, record pulses are supplied for two cycles to the α and β record units, causing the twenty quantities from channels k and $k+1$ to be recorded in channels α and β, respectively, in corresponding number times.

In a fast to slow storage transfer, just ten quantities, from the γ channel, are recorded in slow storage. As indicated in the diagram, there is provided on the γ channel a third playback pole piece, which delivers the quantities stored there, via an associated playback unit, to the slow storage record unit. This will be connected to the pole piece on channel k, whenever a negative slow storage channel number, $-k$, has been delivered to the δ register. When the command for a transfer is given, record pulses are supplied for two cycles, causing the ten quantities in γ_0-γ_9 to be recorded at number times 0-9, respectively, in channel k.

One of the chief advantages of the selection controls provided for slow to fast transfers becomes apparent when some or all of the channels 0-399 are used for interpolational storage. Let $f(x)$ be a function computed and tabulated for equal increments of the independent variable, successive tabular entries being recorded in a series of slow storage registers, r, $r+1$, $r+2$, ... Then $f(x)$ can be reduced to a function $g(r)$ of the slow storage register number by a linear transformation,

$$r = ax + b.$$

The whole process of interpolation then depends on the selection of appropriate slow storage register numbers. Since in a transfer from slow to fast storage the quantities on two channels, k and $k+1$, are always recorded in the α and β channels, it follows that ten successive values of $g(r)$ for $r = 10k+i$, $10k+i+1$, ..., $10k+i+9$, are always made available in fast storage, regardless of the value of i. This insures a sufficient number of tabular entries in fast storage to permit ninth order interpolation after one slow-to-fast transfer operation.

For example, suppose that values of a function $f(x)$ have been recorded in slow storage registers $1000 \leq r \leq 1499$, for $x = 1.00, 1.05, \ldots, 25.95$, respectively. Then

$$r = 20x + 980.$$

If a ninth order interpolation is to be performed for $x = 3.66$, then

$$r = 1053.20.$$

When, in addition, a central-difference formula is to be used, there are required in fast storage five tabulated values on each side of the interpolation interval, namely, those values of $f(x)$ stored in registers 1049-1058. Hence delivery of $k = +104$ to the δ register insures that channels 104 and 105, containing the required values, will be connected to playback units. Note that the value of r was diminished by 4 before deriving k for this selection. After the transfer, the quantities from registers 1049-1058 will be stored in registers α_9, β_0, β_1, ..., β_8, respectively. They may be chosen for delivery to the arithmetic unit, as required by the interpolation formula, with the aid of sensing circuits controlling the α and β channels. These circuits, mentioned briefly under item 12 of this chapter, are described in detail in Chapter III.

7. *Three transfer channels, A, B, C,* indicated in the center of Fig. 1.8 and previously described. Numbers on these channels may be delivered under manual control to a set of lights at the top of the main control panel. For example, Plate XII shows the number, 0909···0909, as it appears to the operator. This visual transcription of quantities on the transfer channels, in combination with the manual controls for operating the calculator, is of great assistance in testing, in checking, and in tracing sources of error.

8. *One sixteen-digit serial adder.* To add two positive quantities in the calculator, augend and addend are first selected from storage and regenerated on the *A* and *B* transfer channels. During blank number time of the following cycle, the numbers are added serially in digit pairs, and the sum is recorded on the *C* transfer channel. It should be noted that the adder is also used for subtraction, by adding the nine's complements of negative numbers. Therefore, a sum computed during blank number time is always delivered to the adder again, during 0 number time, in order to pick up the end-around carry, if any, resulting from the use of nine's complements.

When an augend and addend are selected from storage in cycle n, their sum is not available until the end of 0 number time, relative to a playback pole piece, in cycle $n+1$. However, the sum may still be returned to storage in the 0 (or any other) time phase, due to the displacement of one number time between record and playback pole pieces. Thus the time required for one addition, including the selection of augend and addend, is one cycle.

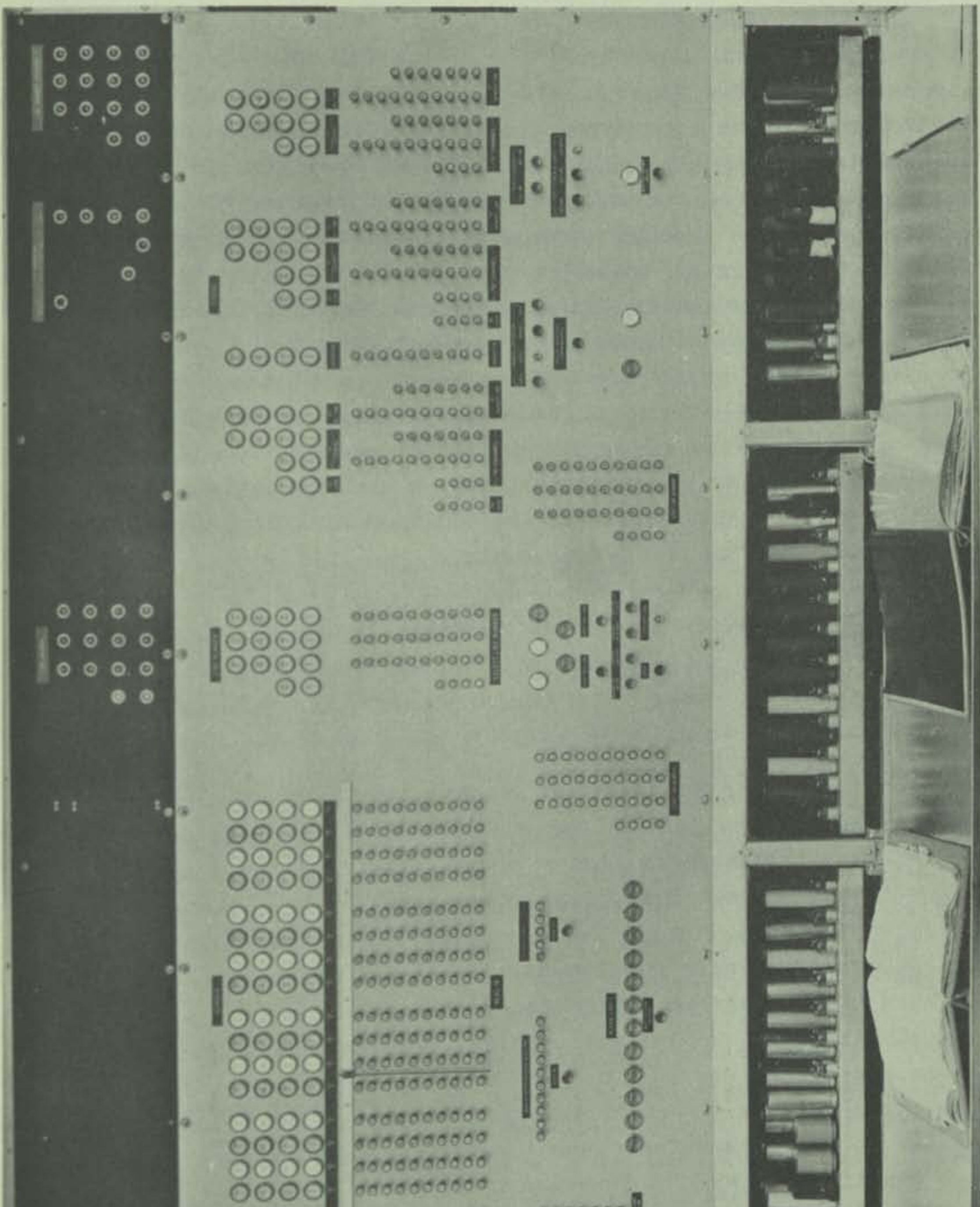


Plate XII *Main Control Panel*

A computed result which is returned to storage during cycle $n+1$ cannot be played back from storage until cycle $n+2$. Therefore, provisions are made, under control of the σ command, for delivering a quantity from the C transfer channel directly to the A transfer channel, bypassing fast storage when desired. Further arithmetic operations may then be performed immediately on a computed result, insuring that n quantities may be accumulated to form a single sum in n cycles.

9. *Sign control circuits*, consisting of vacuum-tube trigger pairs associated with the A and B transfer channels. When a quantity is recorded on the A (or B) transfer channel, its algebraic sign, stored as a 0 or 9 in the seventeenth columnar position, is combined in the A (or B) sign storage unit with a "transfer" sign, +, -, +| |, or -| |. The transfer signs, which are part of each coded instruction, permit the inversion of algebraic signs and the imposition of positive and negative absolute value signs on quantities selected from storage. As will be seen later, the preparation table for instructional tapes automatically records a subtraction as the addition of a negative number, and inverts the B transfer sign. For example,

$$x_6 - y_3 = d_4$$

is recorded in the equivalent form,

$$x_6 + (-y_3) = d_4.$$

The provisions for subtraction and the addition of negative numbers include gates, controlled by the A and B sign storage units, for the direct and inverted transmission of quantities from the A and B transfer channels to the adder. Thus, a positive sign causes a number to be read directly to the adder, while for a negative sign the binary digits are inverted, causing nine's complements to arrive at the adder.

Because of the representation chosen for the algebraic sign, the sign of the sum is equal to the sum of the signs of the augend and addend, plus the end-around carry. The sum sign controls gates (not indicated in Fig. 1.8) for the direct or inverted transmission of sums coming from the adder to the C transfer channel during 0 number time. This insures that a negative sum will be inverted from nine's complements to the usual form, the positive absolute magnitude followed by an algebraic sign. The sum sign is then recorded in the seventeenth columnar position with the sum digits on the C transfer channel.

10. *One multiplication unit*, which computes the thirty-two digit product of any two quantities stored in the calculator. Three cycles of

machine time are required for one multiplication, including the selection of quantities from storage. Multiplication is serial, all digits of the multiplicand being multiplied in succession by each of the multiplier digits. The product of each digit pair, consisting of two decimal digits, is divided into right- and left-hand components, which are combined, after proper delays, in two adders, the regular unit used for addition and an identical supplementary one. The product is recorded in storage channels having a total of seven playback pole pieces, associated with the seven possible decimal point locations. The manual selection of an operating decimal point provides a relay connection to one of these pole pieces, in order that the product may always be played back at the operating decimal point and recorded on the C transfer channel. Since the decimal point in a product lies between columns $2n$ and $2n+1$ for an operating decimal point between columns n and $n+1$, the digits played back through the various pole pieces are $n+1$, $n+2$, ..., $n+16$, for $n = 16, 15, 14, 12, 11, 9$, or 0, as dictated by the manual setting of the decimal point.

Three of the pole pieces can also be selected electronically, by a coded instruction, making three groups of product digits available regardless of the operating decimal point. One group contains product digits 31-16, that is, the product for an operating decimal point between columns 16 and 15. Thus it is possible to carry on subsidiary computations at this decimal point, as is done in the computation of the elementary functions, no matter where the operating decimal point may be.

The other available groups of product digits are the high- and low-order digits, 32-17 and 16-1, respectively. Their inclusion makes it possible to use the calculator for double-accuracy work with quantities of thirty-two digits. The high- and low-order digits constituting one quantity are then recorded in two registers on the same channel. When the low-order digits are stored at number time 0, 1, 2, 3, or 4, then number time 5, 6, 7, 8, or 9, respectively, is reserved for the sixteen high-order digits. A double-accuracy addition requires four cycles of machine time, as well as special instructions, which are supplied automatically by the preparation table for instructional tapes. On the other hand, the program of instructions for a double-accuracy multiplication, consisting of four sixteen-digit multiplications as well as several additions and shifting operations, must be supplied by the mathematician.

In a multiplication the product is computed as a positive absolute value. The proper algebraic sign, determined by sensing the A and B sign storage units, is supplied in the seventeenth columnar position on the C transfer channel.

11. *A circuit for sign choice.* The sign determined for a product may also be associated with the quantity on the *B* transfer channel, causing the sign of that quantity to be inverted when the *A* sign is negative. This selective operation is indicated by the symbol ⊙ on the keyboard of the preparation table for instructional tapes, and has a number of important uses. For example, in rounding off a quantity to n places of accuracy, the choice circuit may be used to determine the sign of the correction, $5 \times 10^{-(n+1)}$, accordingly as the quantity to be rounded is positive or negative.

12. *A sensing circuit called the i register*, which is used in the selection of fast storage registers and in shifting operations. Instead of specifying a number time 0-9 for the selection of a storage register on a given channel, the coded instructions may demand that a number time be chosen corresponding to the value of a digit previously computed and stored in the i register. The time phase selected by this method is referred to as the i number time. For example, if the digit 6 has been delivered to the i register, then x_i represents the register on channel x at number time 6. These facilities increase the flexibility of the coding, since the digit i is delivered over the *C* transfer channel and may be subjected to arithmetic operations as a computation proceeds.

The digit i must be computed in the thirteenth columnar position. When a quantity is delivered from the *C* transfer channel to the i register, vacuum-tube trigger pairs store not only the thirteenth digit but also the algebraic sign and the 1 component, in the 2*,4,2,1 notation, of the fourteenth digit as well. However, the selection of the i number time depends only on the thirteenth digit, the remaining information being used in the following applications.

After a transfer from slow to fast storage, the i register may be used in selecting values from the α and β channels. In this case, instead of specifying the channel, α or β, and the number time, 0-9, the coded instruction demands that from the twenty registers, α_0-β_9, that quantity be chosen which has the slow storage register number, r, previously delivered to the i register. The last digit of r, in the thirteenth columnar position, controls the number time selection, while the 1 component of the digit in the fourteenth columnar position indicates whether the kth or $(k + 1)$st channel, corresponding to α or β, respectively, contains the desired quantity.

For example, suppose that after channels 104 and 105 have been transferred to fast storage, the quantities from registers 1049-1058 are required in succession for a ninth order interpolation. When r = 1049 has been delivered to the i register, the quantity on channel α at number time 9 will be selected, the circuits previously described having already insured that this is the quantity from slow storage register 1049. When r is increased by 1 and delivered to the i register again, then the quantity on channel β at number time 0, previously transferred from slow storage register 1050, is made available; and so on. Thus the whole process of selection from slow storage may be made to rest upon the computation of control numbers to be delivered to the δ and i registers.

A shifting operation is accomplished through a multiplication by a power of ten, selected under control of the sign and digits stored in the i register. The number of columns to be shifted, $s = 0,1,\ldots,15$, must be delivered to this register in the fourteenth and thirteenth columnar positions. The algebraic sign of s determines the direction of the shift, positive or negative indicating a shift to the left or right, respectively. Although one shift requires a series of coded instructions using five cycles of machine time, this five-cycle subsequence of instructions is recorded automatically by the preparation table for instructional tapes. For example, the order, "a_3 shifted by $c_4 = b_5$," supplies all the coding necessary to shift the quantity stored in a_3 by a previously computed number of columns, represented by the quantity stored in c_4, and to record the result in b_5.

13. *Normalizing circuits*, to shift any quantity stored in the calculator to the left until the first non-zero digit is in the sixteenth columnar position. A quantity to be normalized is first delivered from the B transfer channel to an electronic counting circuit, to determine the amount of shift necessary. The output of the counting circuit is then used to initiate a shift to the left, by means of the shift circuit already described. The six-cycle subsequence of coded instructions required for a normalization is supplied automatically by the preparation table for instructional tapes. For example, the order, "normalize $d_7 = a_3$," supplies all the instructions necessary to normalize the quantity in d_7 and store the result in a_3. Normalization may be used when a function having a wide range of values is to be recorded on tape and printed as an integer followed by a decimal fraction, together with an associated power of ten. The amount of shift required in the normalization, combined with a constant dependent upon the position of the operating decimal point, supplies the required exponent.

14. *Means for the computation of the elementary functions x^{-1}, $x^{-1/2}$, $\log_{10} x$, 10^x, $\cos x$, $\tan^{-1} x$.* Included in the calculator for this purpose are 121 permanently recorded constants, an associated selective

sensing register, and a computing routine for each function. The constants consist of such quantities as $\pi/2$, $\log_{10} e$, and truncated power series coefficients, selected under code control, together with first approximations for iterative routines and similar data which may be selected under control of the sensing register. The computing routines for the functions have been recorded in one section of the instructional storage drum, making it possible to evaluate a function by referring to a group of previously recorded instructions, without repeating the functional coding. When one of the functions is selected on the preparation table for instructional tapes, the necessary commands are supplied to refer to the associated computing routine. For instance, to carry out the order, "log $e_3 = f_7$," the calculator is instructed to refer to the logarithm routine, using the quantity in e_3 as argument, to record the computed value of the logarithm in f_7, and to refer back to the main sequence of instructions.

The function x^{-1} is included in order to eliminate the need for special division circuits. Thus, any order using the division sign, ÷, on the keyboard of the instructional tape preparation table refers the calculator to a quotient computing routine, which computes the function x^{-1} for the divisor, and multiplies the result by the dividend.

The functions are computed to fifteen decimal places of accuracy. Considerable effort was expended to insure that the routines adopted be economical as well, in regard to the number of additions and multiplications required and the number of associated constants to be stored. Chapter V describes in detail the theoretical basis of the computation of the functions, including examples of coded function routines.

15. *Check circuits*. Two kinds of checking facilities are provided in the calculator. Mathematical checks on the computation may be made by subtracting from a predetermined positive tolerance the absolute value of a comparison quantity, such as the difference between two independently computed values of the same function. When the resultant check quantity is delivered via the *C* bus to a check-stop register, the calculator is stopped and an indicator lamp is lighted if and only if the sign of the check quantity is negative. Otherwise the calculator continues in operation.

It is also possible to deliver the quantities on the *A* and *B* transfer channels to an identity check circuit, for comparison of the signs and digits in corresponding columnar positions. The calculator is stopped and an indicator lamp is lighted if any two corresponding digits are not identical. This operation is used, for example, to insure that a quantity stored in one register has been correctly transferred into a second register.

16. *Eight tape read-record mechanisms*, and associated controls for transferring numbers between the external tape storage and internal drum storage. Each mechanism may be used for either reading or recording, but manual adjustments for one operation or the other must be made at the outset of each problem. Twenty-five cycles, or approximately 0.1 second, are required to read or record one quantity on tape. During this time the calculator may be used for any operations not concerned with the tape read-record system. As one set of electronic equipment serves all eight mechanisms, only one mechanism may be used at a time.

The tape used in the calculator, for the input of instructions as well as for numerical input and output, is made of paper, 5/8 inch wide, and coated with a layer of magnetic oxides. Pole pieces, having the same laminated construction as those used with the storage drums, provide a means of reading from and recording on the tape as it moves past the pole piece tips. The gap required between the magnetic medium and the scanning devices in drum recording can be eliminated in tape recording, since direct contact between the pole piece tips and the tape does not cause undue wear. Hence, it is possible to record magnetic dipoles with a density of fifty per inch on the tape. Each of the tape-handling units associated with the calculator is equipped with four pole pieces, which are mounted on the stepped lands of a block, permitting each pole piece to read or record dipoles in one of four parallel tape channels, approximately 1/8 inch apart. Of the four channels, *A*,*B*,*C*,*D*, two, namely *C* and *D*, are reserved for series of regular timing pulses which serve the same function as sprocket holes in perforated paper tapes, while the information is recorded as binary digits 0 and 1 in channels *A* and *B*. A reel of magnetic tape may be erased and reused a great number of times, the limiting factor being the care with which the tape is handled.

On numerical tapes, the twenty decimal or eighty binary digits associated with one quantity in the calculator are recorded serially, in both the *A* and *B* channels. For check purposes, corresponding digits in channels *A* and *B* are one digit out of phase. Numerical tapes may be prepared either by the calculator, in the course of a computation, or by an operator using the numerical tape preparation table. The keyboard of the table, Plate IX, contains an array of switches, for the registration of the sign and sixteen digits of each quantity. An indicator lamp is lighted if a digit is not supplied in each column; and, as a second check, each quantity must be registered twice in the keyboard. If the same sign and digits were indicated both times, the number is delivered to a tape record unit and recorded in the *A* and *B* channels, timing pulses being recorded in the *C* and *D* channels at the same time.

Since reading and recording operations involve the transfer of information from comparatively slow-moving tape to a rapidly rotating drum, or vice versa, an intermediate storage is provided in the form of two binary drum channels, indicated at the left of Fig. 1.8. In contrast to the usual drum storage, where the 2*,4,2,1 components of each decimal digit are recorded in four parallel channels, the eighty binary digits of one quantity are here stored and regenerated serially on both channels, all the 2* components, followed by all the 4, the 2, and the 1 components in that order.

In order to read one quantity from tape into fast storage, the calculator first selects one of the tape read-record mechanisms previously adjusted for reading, to provide connections between the pole pieces in this mechanism and playback units. The digits played back from the *A* and *B* tape channels are recorded and regenerated on the intermediate storage channels, the *A* digits going to one channel, and the *B* digits to the other. Four playback pole pieces on each of the intermediate storage channels make it possible to transfer the digits originally derived from the *A* and *B* tape channels, respectively, to the four-wire busses leading to the *A* and *B* transfer channels. The two sets of digits may then be compared before the quantity is used in a computation.

A quantity to be recorded on tape is delivered via the *C* bus to both of the intermediate storage channels, and once recorded, is regenerated. After the transfer of the quantity has been checked, the digits from the two channels are delivered serially to two tape record units connected to pole pieces on the selected tape read-record mechanism. Timing pulses are recorded in the *C* and *D* tape channels while the digits are recorded in channels *A* and *B*, a check being maintained between corresponding *A* and *B* digits during recording to insure an accurate output.

17. *Five printers*, for presenting computed results in a form suitable for publication. Since each printer is a separate unit, consisting of an electric typewriter controlled from one of the panels at the left front of the calculator, as many as five reels of tape may be printed simultaneously. Each typewriter requires approximately 3.5 seconds to print one sixteen digit quantity. Each printer control panel is equipped with a tape-reading mechanism, together with plugs and switches which determine the number of digits to be printed, the intercolumnar and interlinear

spacing, and other items related to the typography of the printed page. Digits are played back from the *A* and *B* tape channels and stored in relays, which are scanned by a rotary step switch in order to supply electrical impulses to solenoids under the typewriter keys. The type bars controlled by the solenoids operate contacts which provide a means of comparing each *A* digit, as it is printed, with the corresponding digit derived from the *B* channel. This checks not only that each digit is printed but also that it is the same on both tape channels, to insure completely reliable results.

18. *Sequencing and control equipment*, for preparing, storing, and delivering to the calculator the coded instructions necessary for its operation. In the calculator each coded instruction governing one cycle consists of thirty-eight binary digits, indicating the storage registers to be selected and the arithmetic operation to be performed. The allocation of the digits is shown in Table 1.2. In addition to the digits 0-9, the coded instructions make use of the combinations in the 2*,4,2,1 notation which do not represent decimal digits. These combinations, listed in Table 1.3, are referred to as starred digits, the digit values being assigned in accordance with the weights in the four columns, and the asterisk serving to distinguish them from the standard decimal digits. The binary digits constituting one

TABLE 1.2

Allocation of sequence digits

Digit	Weight	Use
1	1	*A* transfer
2	2	sign
3	1	ten's digit
4	2	*A* channel
5	1	unit's digit
6	2	*A* channel
7	4	
8	2*	
9	1	*A* number
10	2	time
11	4	
12	2*	
13	1	Operation
14	2	
15	4	
16	2*	
17	1	*B* transfer
18	2	sign
19	1	ten's digit
20	2	*B* channel
21	1	unit's digit
22	2	*B* channel
23	4	
24	2*	
25	1	*B* number
26	2	time
27	4	
28	2*	
29	1	ten's digit
30	2	*C* channel
31	1	unit's digit
32	2	*C* channel
33	4	
34	2*	
35	1	*C* number
36	2	time
37	4	
38	2*	

command are stored in thirty-eight parallel channels on the instructional storage drum. With a drum circumference of fifty inches and a density of twenty binary digits per inch, 1000 digits may be recorded around the periphery in one channel. Four parallel channels, Fig. 1.8, each containing thirty-eight binary channels and thirty-eight record-or-playback pole pieces, provide a capacity of 4000 instructions. The location of each instruction on the drum may then be indicated by a four-digit line number, 0000-3999, in which the first digit refers to one of the four parallel channels, while the last three specify the time phase. To aid in the selection of instructional storage locations, for either record or playback operations, serial numbers 000-999 have been permanently recorded around the drum periphery. The numbers are played back through associated pole pieces to indicate the last three digits of the line number.

TABLE 1.3				
Non-decimal digits 2*,4,2,1 notation				
	2*	4	2	1
2*	1	0	0	0
3*	1	0	0	1
4*	1	0	1	0
5*	0	1	0	1
6*	0	1	1	0
7*	0	1	1	1

The instructions required for a computation are most efficiently recorded on the drum from an input tape prepared on the instructional tape preparation table. The keyboard of this table, Fig. 1.9, is so arranged that instructions may be registered there as mathematical operations rather than as numerical codes. All the orders previously mentioned in this chapter can be registered by depressing keys in the various sections of the keyboard, in a manner which should be obvious from inspection of Fig. 1.9.

The preparation table automatically records on the tape the one or more coded commands needed in the calculator to carry out each order registered in the keyboard. The keys at the upper left provide for recording the line numbers, 0000-3999, to indicate where each instruction is to be stored on the drum. In all, sixty-four binary digits are associated with each command on the tape: twelve blank digits, thirty-eight

TABLE 1.4		
Allocation of line number digits		
Digit	Weight	Line Number
39	1	thousand's digit
40	2	
41	1	hundred's digit
42	2	
43	4	
44	2*	
45	1	ten's digit
46	2	
47	4	
48	2*	
49	1	unit's digit
50	2	
51	4	
52	2*	

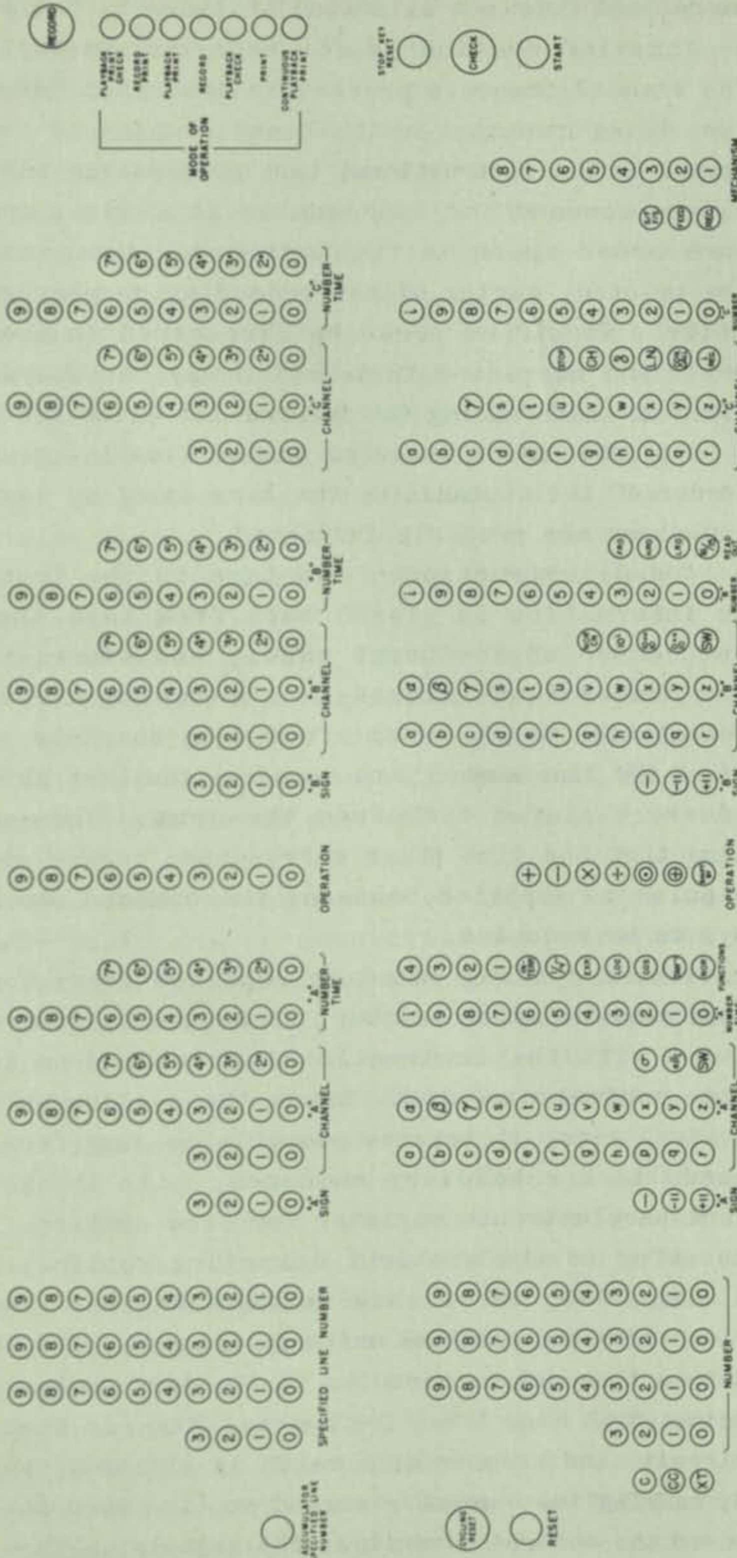


Fig. 1.9—*Keyboard of instructional tape preparation table.*

for the coded command, and fourteen allocated as shown in Table 1.4, for the line number. This information is not recorded in duplicate but divided between the *A* and *B* channels previously described for numerical tapes, timing pulses being recorded in the *C* and *D* channels at the same time. A tape prepared on the instructional tape preparation table may be played back there, one command and line number at a time, and checked by registering each order again in the keyboard. Commands and line numbers may also be printed, during either recording or playback, by an associated typewriter. Should an error be discovered in checking the tape, the line number and corrected instruction may be recorded at the end of the tape. Then in transferring the information to the instructional storage drum, the last information recorded on the line in question will be correct. The order of the commands on the tape is of no consequence, provided the line numbers are properly indicated.

To transfer the coded instructions from tape to the instructional storage drum, the information is played back from tape through the mechanism near the center of the front panel, and stored in relays, Fig. 1.8. The line number is delivered to a matching circuit, which provides a relay connection to one of the four drum channels, depending on the first digit of the line number, and compares the last three digits with the serial numbers played back from the drum. When a match is obtained, indicating that the time phase corresponds to the stored line number, a cycling pulse is supplied, causing the command stored in the code storage relays to be recorded.

In normal operation, a cycling pulse is supplied every four milliseconds, to read one command after another, in sequence, from the playback units associated with the instructional storage drum to trigger pair storage and the control circuits. Since the cycling pulse may be supplied at any desired time, it becomes possible to jump from the main sequence of commands to a subsidiary sequence, with the aid of the matching circuit and an electronic register for line numbers.

For example, to refer to the quotient computing routine, a command is given to call line 3000, where this routine begins. The regular cycling pulse delivered to the playback units is stopped while the number 3000 is delivered from the control circuits to the line number register, by connections indicated in Fig. 1.8. The number, 3000, is then supplied to the matching circuit, and as soon as a match is obtained, the cycling pulse is restored, causing the command recorded on line 3000 to be played back and delivered to the control circuits. The calculator then performs the sequence of commands starting with line 3000.

Numbers may also be delivered to the line number register via the *C* bus, by connections indicated in Fig. 1.8. In this case the line number must be in the sixteenth, fifteenth, fourteenth and thirteenth columnar positions. Then instead of calling a specific line number, the calculator may be referred to the command having the line number previously stored in the line number register. This technique may be used, for example, to return to the main sequence of commands at the end of a subsidiary sequence.

The commands to call either a specific line of coding or the line having the number stored in the line number register may be made conditional, depending on the algebraic sign of a control number previously delivered via the *C* bus to a conditional call register. A positive sign stored there will cause the conditional call to be successful, referring the calculator to the command indicated. A negative sign, indicating that the call will fail, refers the calculator to the next command in the sequence.

Provision is made on the instructional tape preparation table to introduce to the *C* bus of the calculator a numerical quantity, 0000-3999, in the sixteenth, fifteenth, fourteenth and thirteenth columnar positions. Using the switches in the lower left corner of the keyboard, Fig. 1.9, a command may be given for an external transfer, *XT*, of the digits desired. For example, the order, "XT 2316 = a_3," will cause the quantity, 2316 0000 0000 0000, to be recorded in register a_3. Since the range and columnar positions of quantities so introduced correspond to the range and columnar positions of the control numbers used in the calculator, this command is useful for the manipulation of control numbers, as well as for a limited input of numerical data.

To increase the ease and efficiency of operation, a variety of manual controls and indicator lights are provided on the front panel, Plate XII. At the right is a set of switches for the registration of a single coded instruction, to be recorded under manual control at any specified line on the instructional storage drum. It is also possible to deliver a command from the front panel directly to the control circuits, causing the calculator to operate without reference to the instructional storage drum. For check purposes, one command at a time may be inspected in lights, the command being delivered either from the input tape or the drum. A set of lights is also provided for the inspection of line numbers.

To complete the general description of the calculator, it remains only to describe the start and stop controls. The line number of the instruction with which the operator wishes to start may be indicated in a register provided for this purpose on the front panel. Pushbutton controls make it possible to start either on the specified line or, after a stop in the course of a computation, on the next line in the sequence. The calculator may be operated one line at a time as well as continuously, starting with either a specified line or the next line in the sequence. A register is also supplied for specifying the number of the line on which the calculator is to stop. Then, provided that a toggle control switch has been thrown, the calculator will stop when it reaches the specified line. The calculator may be stopped manually at any time by depressing a pushbutton, and will stop automatically if an alarm is sounded. One of several indicator lamps is then lighted, depending on which of the following conditions has occurred:

1. Tolerance-check failure,
2. Identity-check failure,
3. Check failure in the tape read-record system,
4. Insufficient tape in a tape read-record mechanism,
5. Reset failure in the tape read-record system,
6. An addition which exceeds the machine capacity,
7. Low oil pressure in the gear box, or overheated drum bearings.

Of these, the last is regarded as an emergency, and causes the main drum drive motor, the instructional storage drum drive motor, and the power supply which controls drum recording to be shut off, to prevent damage to the machine.

The general description of the calculator given in this chapter, together with the details of coding procedures, Chapter VIII, and the typical examples, Chapter X, provides the information necessary to use the calculator in the solution of problems. The remaining chapters supply the technical details required for the maintenance and improvement of the machine.

CHAPTER II

BASIC CIRCUITS

Early in the design of the calculator, it was decided to use plug-in components to simplify the building and maintenance of the machine. The standard chassis provided to accomodate such components is shown in Plates IV, XIII, and XIV. Connections between sockets, and from sockets to filament transformers, are permanently wired on the underside of the chassis, Plate XIII. Connections between chassis are made with cables, Plate XV, plugged to jacks along the edges of the chassis.

Circuits consisting of resistors, condensers, inductors, and crystal rectifiers are mounted on standard eight-pin bases, Plate XVI, which fit the chassis sockets. An aluminum cover is spun onto each base, and grounded by a connection to one of the base pins. To distinguish between the seventy or more different circuits provided in such component containers, the covers are painted in different color combinations.

In addition to the component containers and thermionic vacuum tubes, many of the electromagnetic relays used in the calculator plug into the chassis sockets. These relays are of the small telephone type, enclosed in metal containers with connections brought out to the pins of standard octal-type bases. The following contact configurations are provided: single-pole single-throw, single-pole double-throw, and double-pole double-throw.

The remaining relays fit special sockets, flush-mounted on panels of the machine, as shown in Plate XXV. These relays have six double-throw contacts each, and are of two types, single coil and double coil, distinguished by red and green molded bakelite frames, respectively. They are identical with those used in the Mark II Calculator[1].

The machine contains approximately 4500 thermionic vacuum tubes, of seven standard classifications:

1. Type 6AG7, a power output pentode, used chiefly as a cathode follower, and also to record on magnetic tape or to operate indicator lamps.

2. Type 6L6, a beam power amplifier, used to operate rotary step switches or to record on the magnetic drums.

3. Type 6V6, a beam power amplifier, used to operate the six-contact relays.

4. Type 403B, a miniature pentode, used as a Class A amplifier.

5. Type 6AS6, a miniature voltage amplifier pentode, in which the

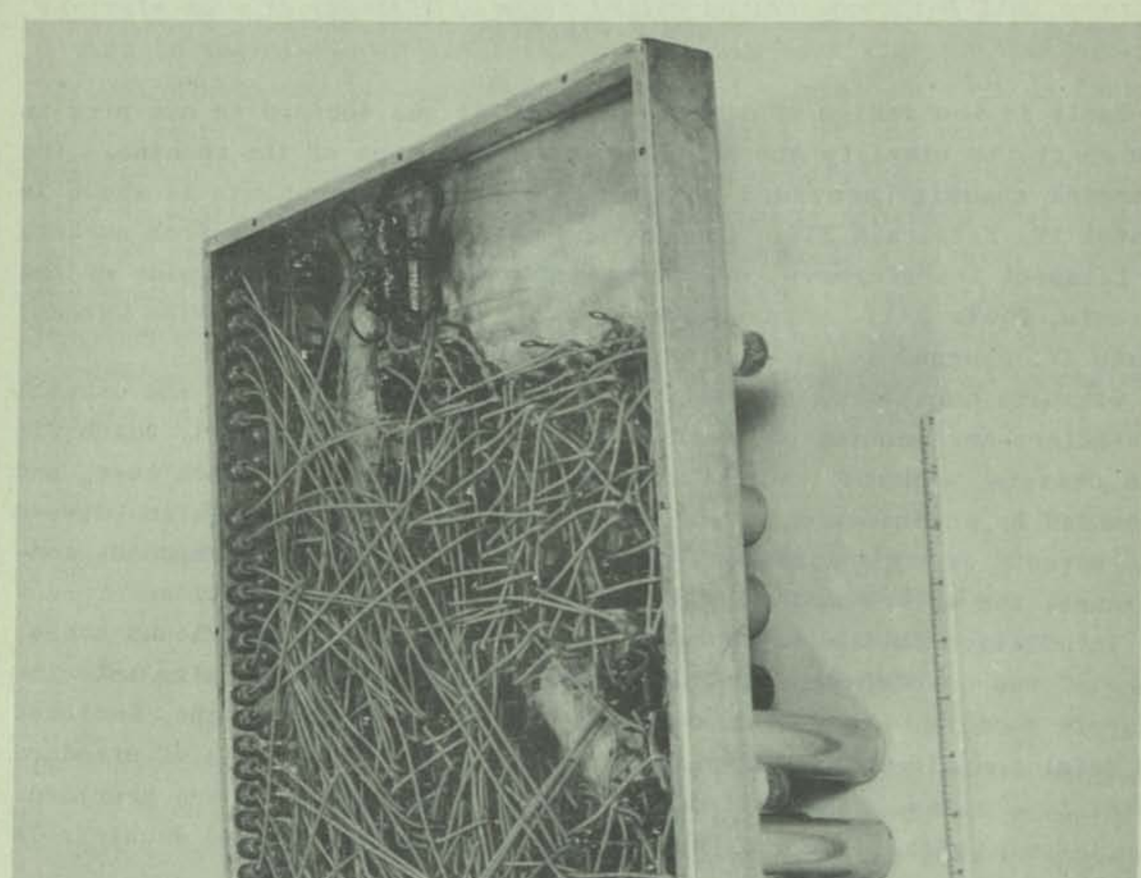


Plate XIII *Chassis: Underside*

Plate XIV *Racks of Chassis*

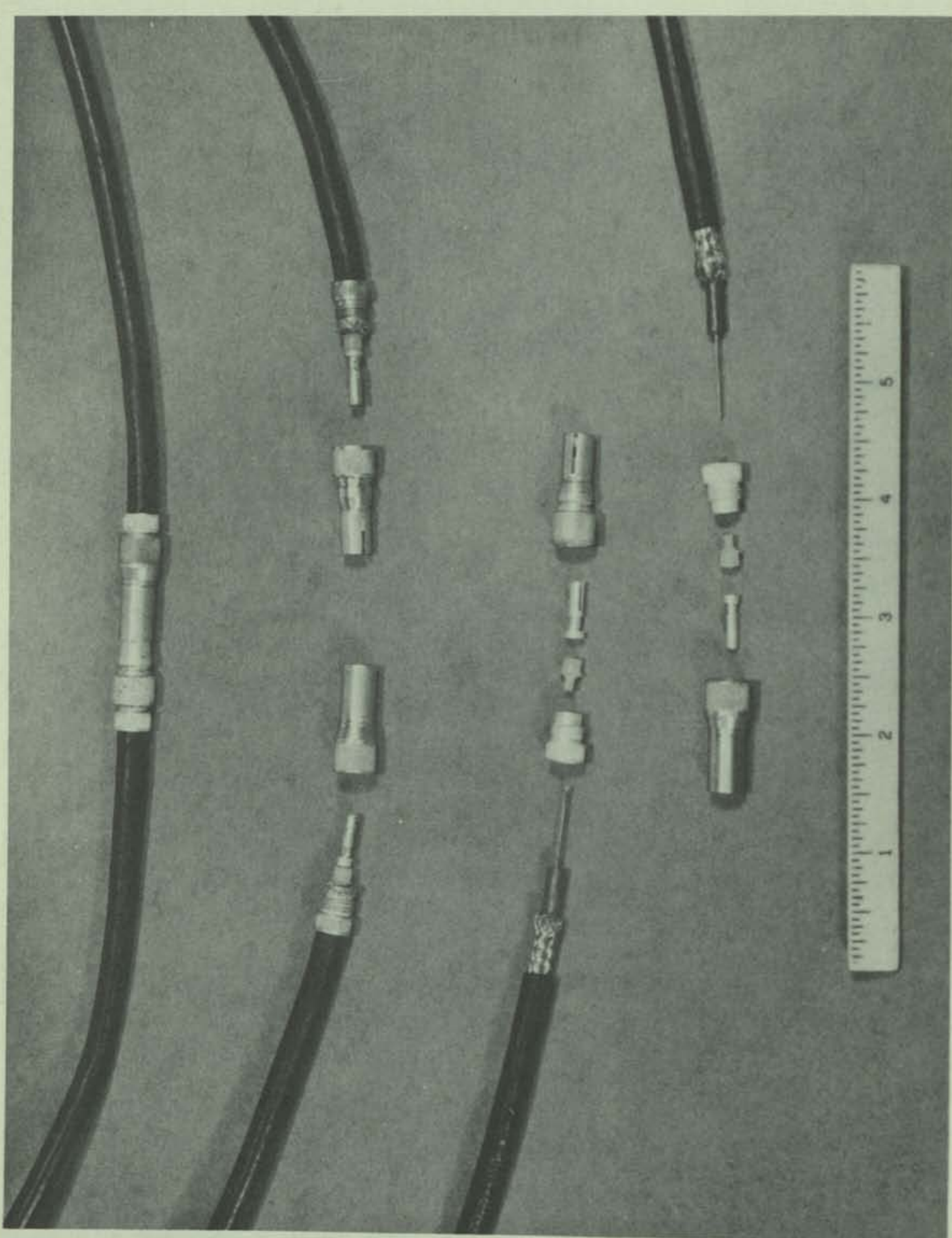


Plate XV *Cables (exploded view)*

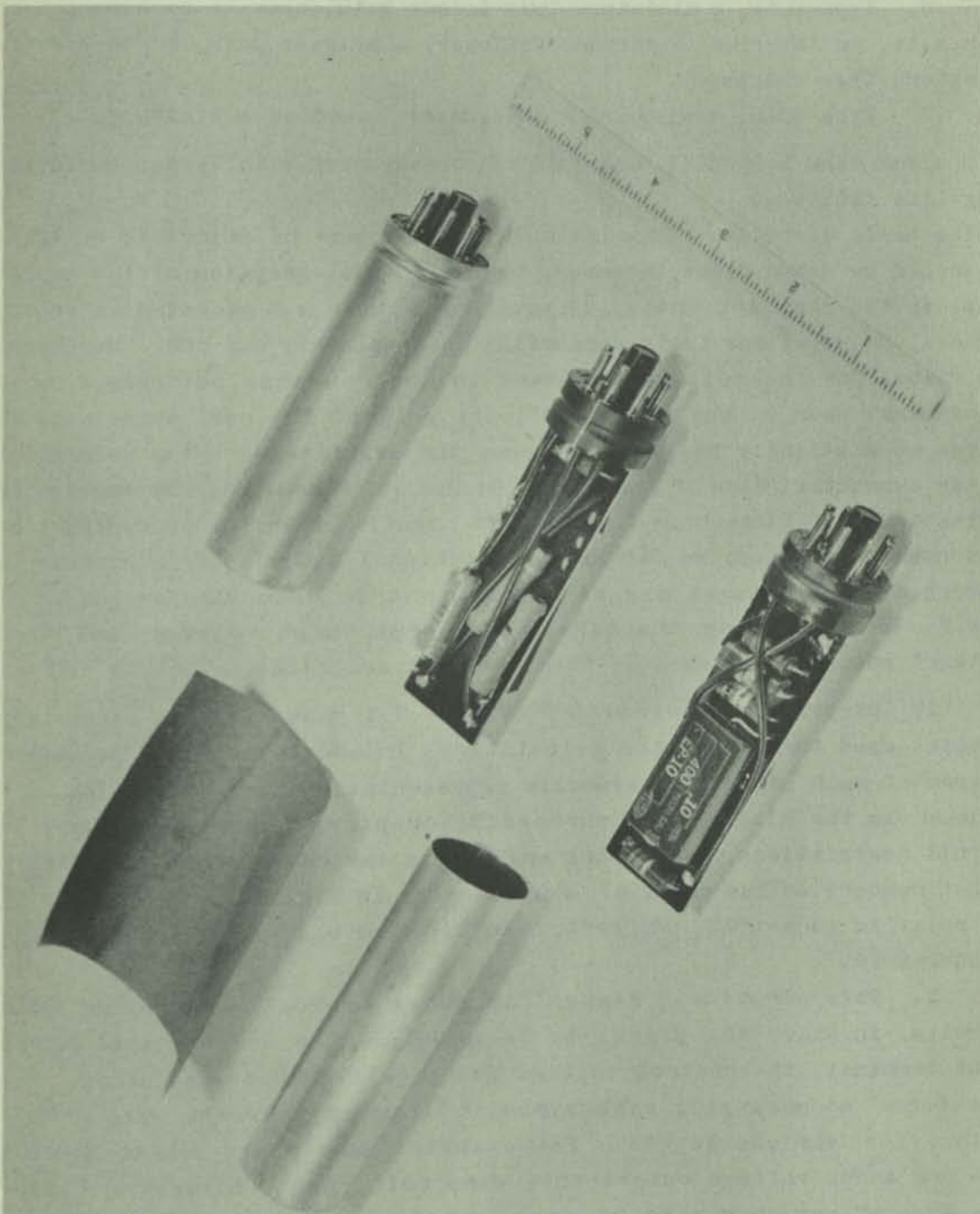


Plate XVI *Component Containers (exploded view)* control and suppressor grids have almost equal control over the plate current, used chiefly as a gate or as one tube of a trigger pair.

6. Type 2C51, a miniature twin triode with separate cathodes, used as a gate, an inverter, a cathode follower, a trigger pair, or to operate telephone-type relays.

7. Type 6AL5, a miniature twin diode, used as a limiter.

Of these, the 6AS6, 2C51, and 403B are constructed with Western Electric long-life cathodes.

The basic circuits of the calculator will now be described briefly. It should be noted first, however, that with the exception of the amplifiers in the playback units, the vacuum tubes are operated as on-off devices, being either fully conducting or completely cut off. Moreover, the values for the voltage dividers in the coupling networks were so chosen that when a tube is cut off, the grid of the next stage will be driven to a slightly positive voltage, the exact value being determined by the characteristics of the tube. On the other hand, a grid voltage is of the order of fifteen or twenty volts negative when the preceding tube is conducting. Thus, so far as the functional behavior of a circuit is concerned, all voltages may at any instant be described as "high" or "low." In succeeding chapters, the terms "high voltage" and "low voltage" refer to voltages of the type just described.

1. *Miscellaneous circuits*. Figure 2.1 shows typical elementary circuits used throughout the calculator. In addition to the schematic diagram of each circuit, a symbolic representation[2] is given which will be used in the diagrams of succeeding chapters. Where necessary in circuit descriptions, both tubes and relays may be referred to by their socket numbers on the chassis. For example, in Fig. 2.1(*c*) relay *S*15 is the relay in socket 15, while triode *S*10*a* is half of the twin triode in socket 10.

2. *Gate circuits*. Figure 2.2 shows typical vacuum-tube gate circuits, in which the signal to be gated may be delivered to either input terminal, the control voltage being delivered to the other. The wave forms accompanying each symbolic diagram show the gate output voltage for various inputs. For example, the plate-loaded pentode gate has a low voltage output only when both input voltages are high. This type of gate may also be used to gate pulses of short duration, Fig. 2.2(*b*). In this case, however, condenser coupling is used, and an inverter is provided to obtain positive pulses at the output terminal.

A pair of mutually inverted gates is shown in Fig. 2.3. By coupling from the screen of one pentode to the control grid of the other, the

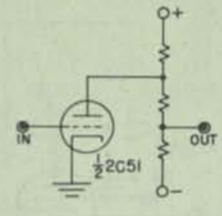
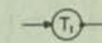
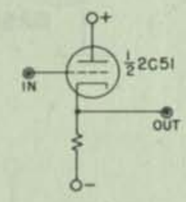
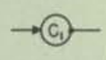
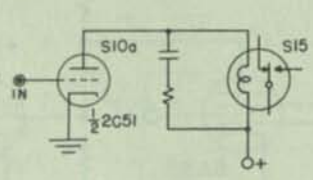
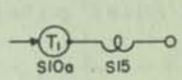
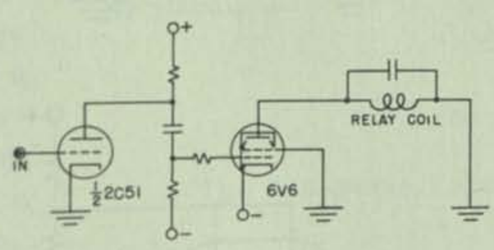
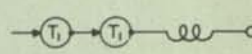
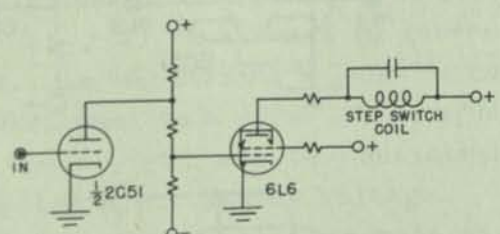
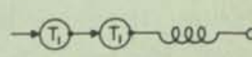
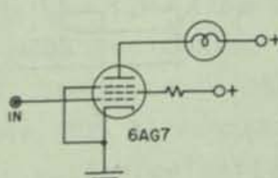
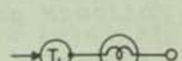
Circuit	Schematic Diagram	Symbolic Diagram
(*a*) *Inverter.*		
(*b*) *Cathode follower.*		
(*c*) *Telephone-type relay pick-up.*		
(*d*) *Six-contact relay pick-up.*		
(*e*) *Step switch.*		
(*f*) *Indicator lamp.*		

Fig. 2.1—*Miscellaneous circuits.*

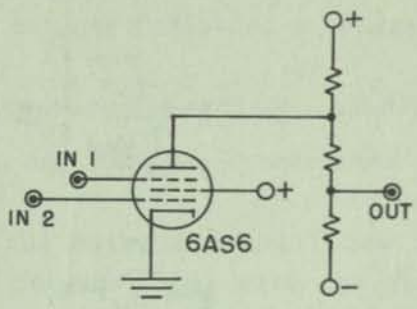
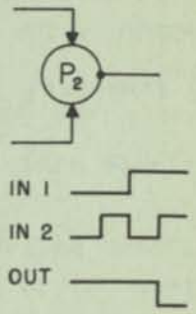
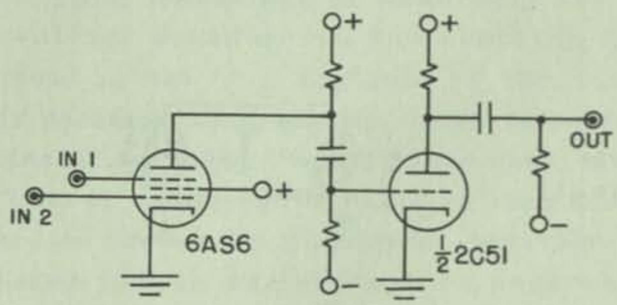
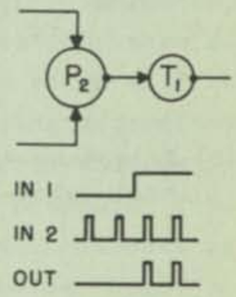
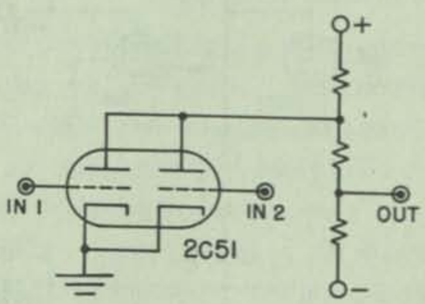
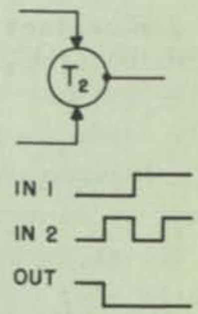
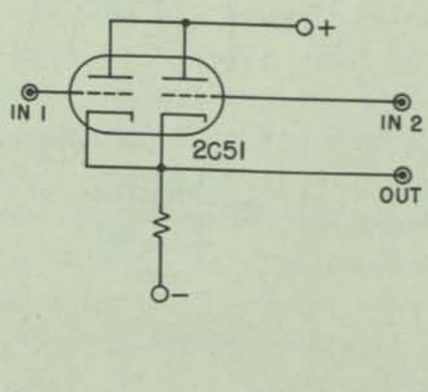
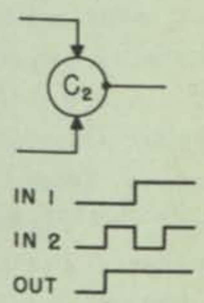
Type of Gate	Schematic Diagram	Symbolic Diagram
(*a*) *Plate-loaded pentode gate.*		
(*b*) *Pulse gate.*		
(*c*) *Plate-loaded triode gate.*		
(*d*) *Cathode follower gate.*		

Fig. 2.2—*Vacuum-tube gate circuits.*

inverse of one input voltage is obtained without the additional triode required in the symbolic diagram.

Crystal rectifier gates are used for selecting playback signals. The principle underlying the operation of these gates is that any diode, whether a thermionic tube or a semi-conductor, offers small resistance to current in the forward direction and large resistance to current in the reverse direction. Figure 2.4 shows a typical current-voltage characteristic for a germanium crystal diode. If a small a.c. voltage is superimposed on the d.c. biasing voltage, the crystal offers variational resistance to the a.c. voltage, approaching the value dv/di where $v = f(i)$ is assumed to represent the d.c. characteristic of the crystal. As the biasing voltage is changed, the resistance varies from a few hundred ohms to a value of the order of several hundred thousand ohms. Thus the crystal may be considered as a switch, which is opened or closed by the d.c. biasing voltage.

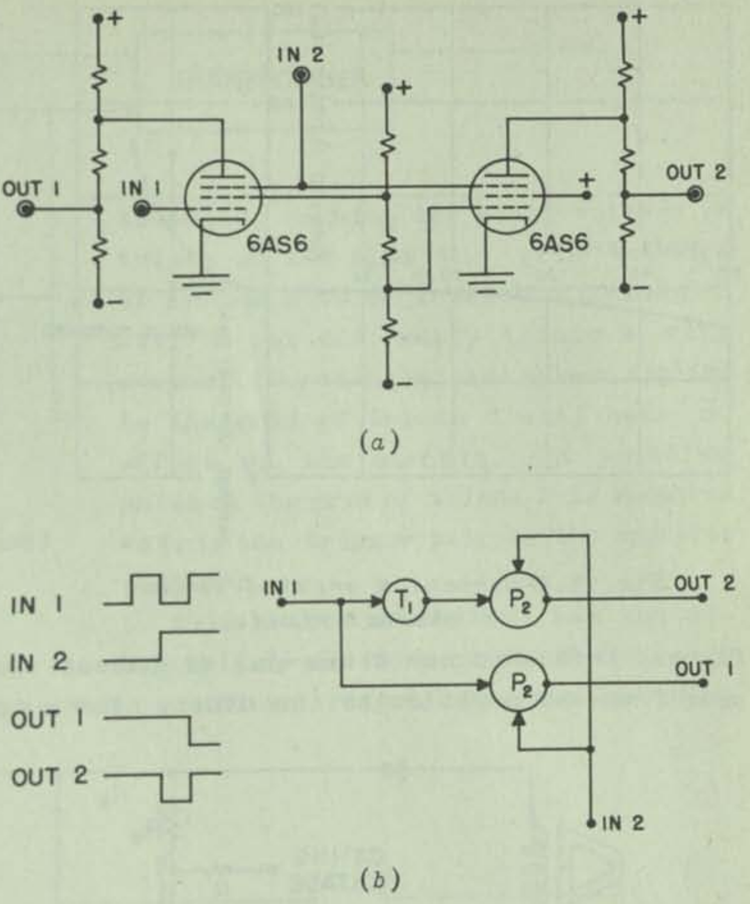


Fig. 2.3—*Mutually inverted gates.*

Figure 2.5 shows two crystal rectifier gate circuits, the arrow in the rectifier symbol indicating the direction of the low resistance path through the crystal. When a positive gating potential of about twelve volts is applied as indicated, the gate is open, and low-level signals from the corresponding pole piece will be transmitted through the crystals to the output bus. A negative gating potential of about two volts will close the gate. Resistors R_1 and R_2 isolate the playback pole piece from the gating voltage, and resistors R_3 prevent interaction of the gating voltages. Each gate circuit consists of two crystals and five

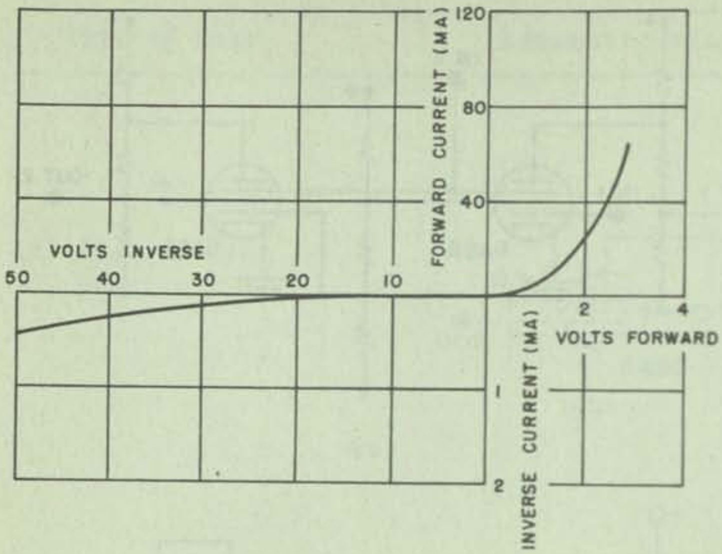
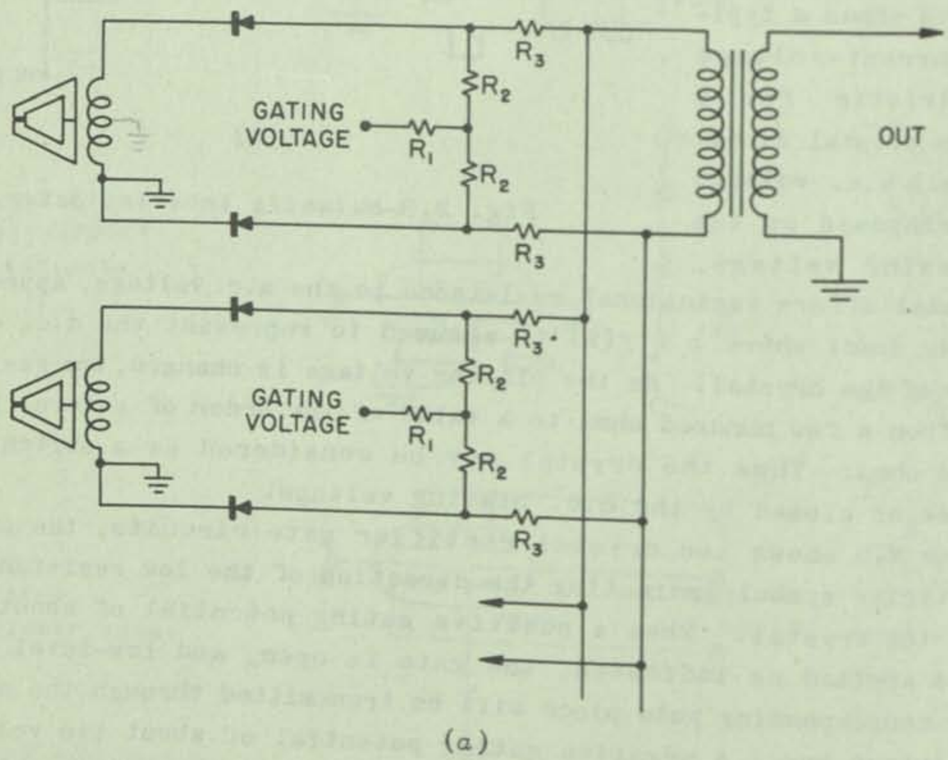


Fig. 2.4—*Germanium crystal diode characteristic.*

resistors, all mounted in a component container, Plate XVI.

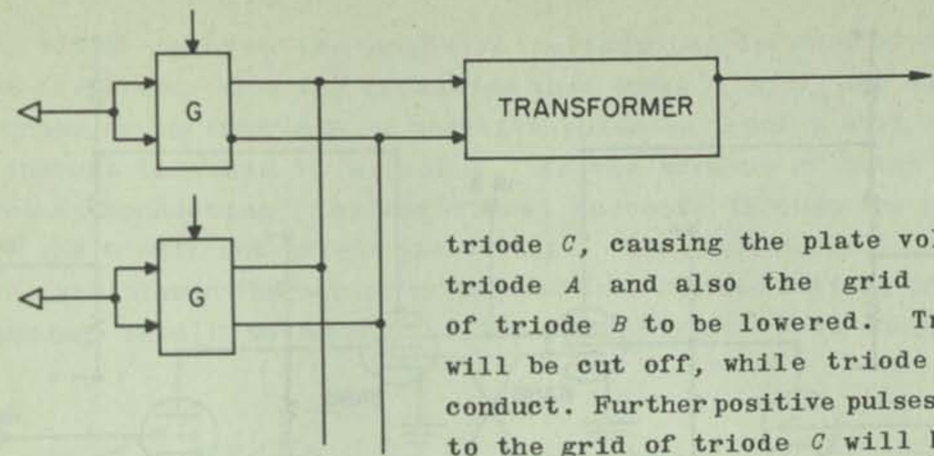
3. *Trigger pairs.* Eccles-Jordan trigger pairs are used in control circuits and for temporary storage of information. The 2C51 envelope indicated in the center of Fig. 2.6(*a*) provides the two triodes, *A* and *B*, of a trigger pair. At any given time, one triode is conducting and the other cut off. In the same figure, triodes *C* and *D* are puller tubes, serving to trip the trigger pair from one position to the other. For example, to make triode *A* of



(*a*)

Fig. 2.5—*Crystal rectifier gates.*

the trigger pair conduct, a positive voltage is applied to the grid of



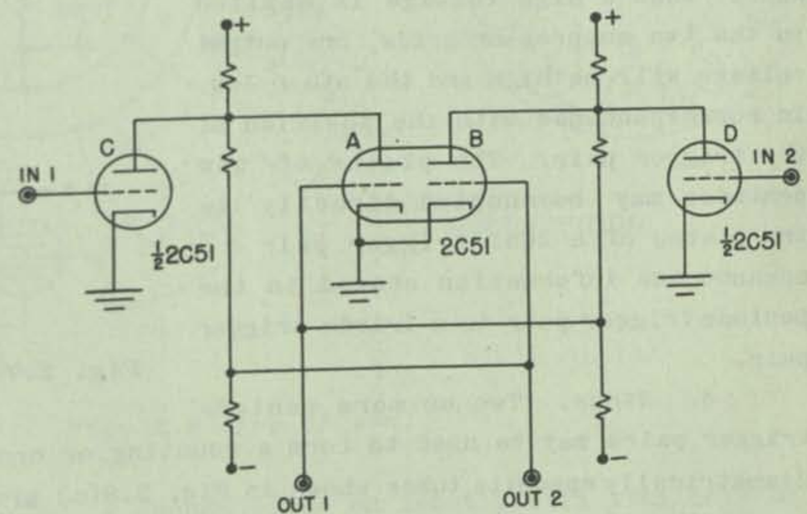
(b)

Fig. 2.5 (continued)

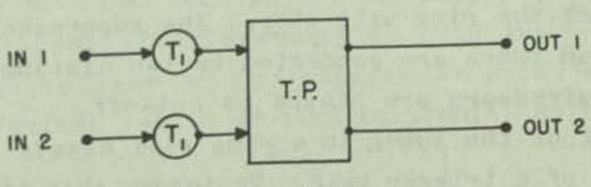
triode *C*, causing the plate voltage of triode *A* and also the grid voltage of triode *B* to be lowered. Triode *B* will be cut off, while triode *A* will conduct. Further positive pulses applied to the grid of triode *C* will have no effect on the circuit. A positive pulse on the grid of triode *D* is required to trip the trigger pair to the opposite position.

This method of control has the advantage that the tripping of the trigger pair is independent of the shape of the triggering pulses. To obtain additional reliability, the resistors in the voltage dividers have been chosen so that the circuit is insensitive to random noise, ripple, or small fluctuations in the power supply voltages. The outputs are taken from the grids of the trigger pair, and may be coupled directly to vacuum-tube control circuits. However, when the output of a trigger pair is to be delivered to several grids or to the grid of a power tube, a cathode follower stage is used to provide isolation.

A combination trigger pair and gate circuit,

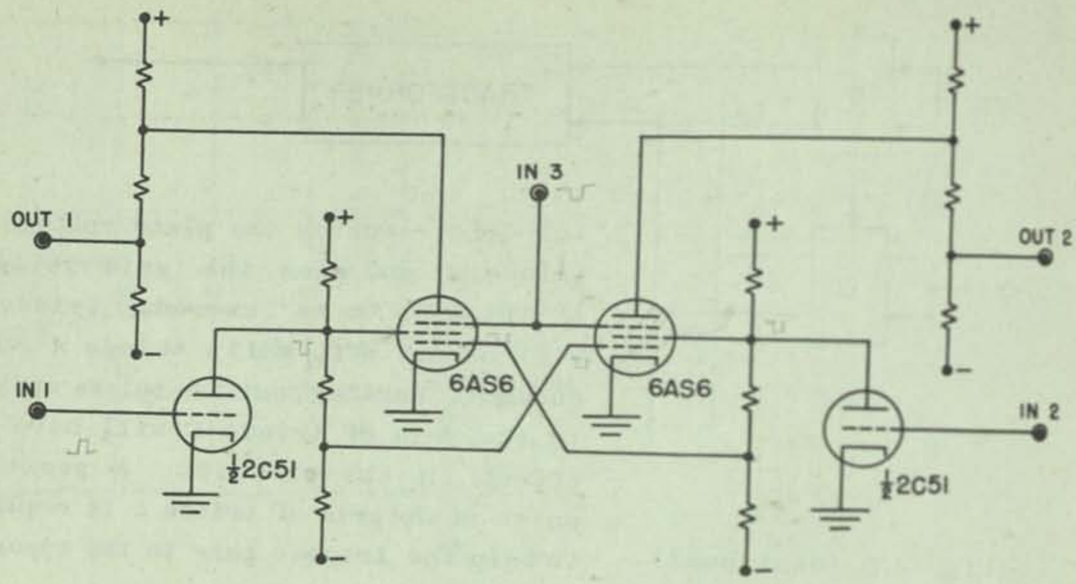


(a)



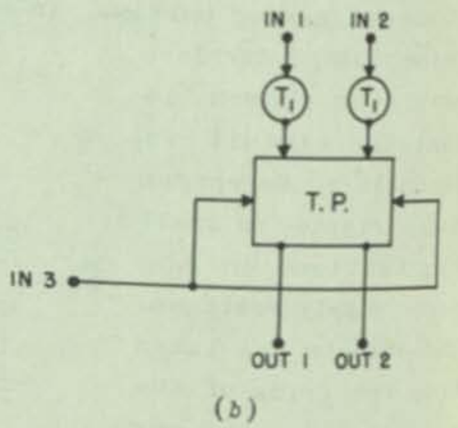
(b)

Fig. 2.6—*Triode trigger pair.*



(a)

Fig. 2.7, is obtained by connecting the cathodes, control grids, and screen grids of two 6AS6 tubes to form a trigger pair. When a high voltage is applied to the two suppressor grids, one output voltage will be high and the other low, in correspondence with the position of the trigger pair. The plates of the pentodes may be coupled directly to the plates of a 2C51 trigger pair to advance the information stored in the pentode trigger pair to a triode trigger pair.

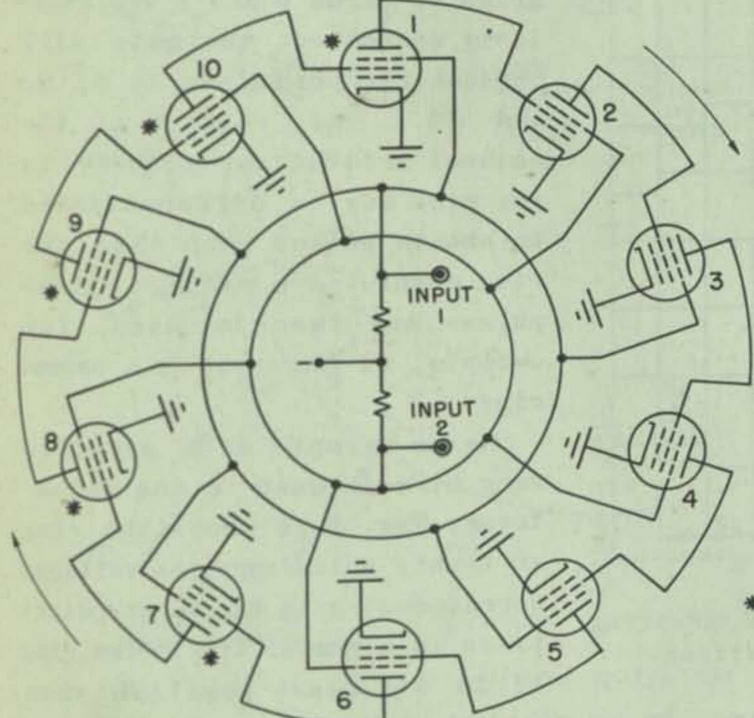


(b)

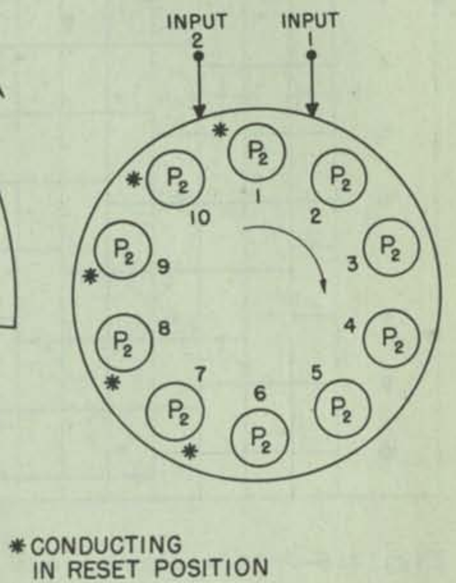
Fig. 2.7—*Pentode trigger pair.*

4. *Rings*. Two or more pentode trigger pairs may be used to form a counting or control ring[3]. When the diametrically opposite tubes shown in Fig. 2.8(a) are connected as trigger pairs, the circuit represents a ring of ten. The connections between the plate of each tube and the screen of the next determine the direction in which the ring will step. The suppressor grids of the odd- and even-numbered tubes are connected to two distinct pulse inputs. Normally all the suppressors are biased to cut-off.

Half of the tubes in a ring will always be conducting since each tube is one of a trigger pair. To insure that the proper sequence of tubes is conducting at the start, the ring may be reset by means of a manual switch, which removes the negative voltage supply from half of the voltage dividers. With the assumption that tubes 7, 8, 9, 10, and 1 have been turned on in this way, a positive pulse on input 1 will result in plate current in tubes 7, 9, and 1. As the screens of tubes 8 and 10 are already conducting, the additional currents through the plates of tubes 7 and 9 will not affect the circuit. The plate current in tube 1, however, will lower the screen voltage of tube 2, thus cutting off tube 7 and causing tube 2 to conduct. The ring has now been stepped from



(a)



(b)

Fig. 2.8—*Ring of ten.*

position 1 to position 2. A second pulse on input 2 will similarly cut off tube 8 and turn on tube 3. Alternate pulses applied to inputs 1 and 2 will cause the ring to step once for each pulse, the only requirements being that the pulses do not occur simultaneously on both input lines, and that each pulse raise the voltage on the suppressor grids above a minimum threshhold for a sufficient length of time to permit the trigger pair to trip.

In many applications it is desirable to obtain control voltages corresponding to one or more positions of the ring. This may be accomplished by connecting the control grids of two ring tubes to the input terminals of a vacuum-tube gate. The gate output, determined by the potentials on the selected control grids, Fig. 2.9, will then be a high or low voltage for one or more stable positions of the ring. For example, if the grids of tubes 1 and 7 are connected to the control and suppressor grids of a pentode gate, the gate will conduct for position 1 only, while if the grids of tubes 6 and 7 are similarly connected, the gate will conduct for positions 7, 8, 9, and 10. The voltage at the control grid of any pentode in the ring may be differentiated to obtain pulses each time the ring completes a cycle. These pulses may then be used, for example, as the input to a second ring.

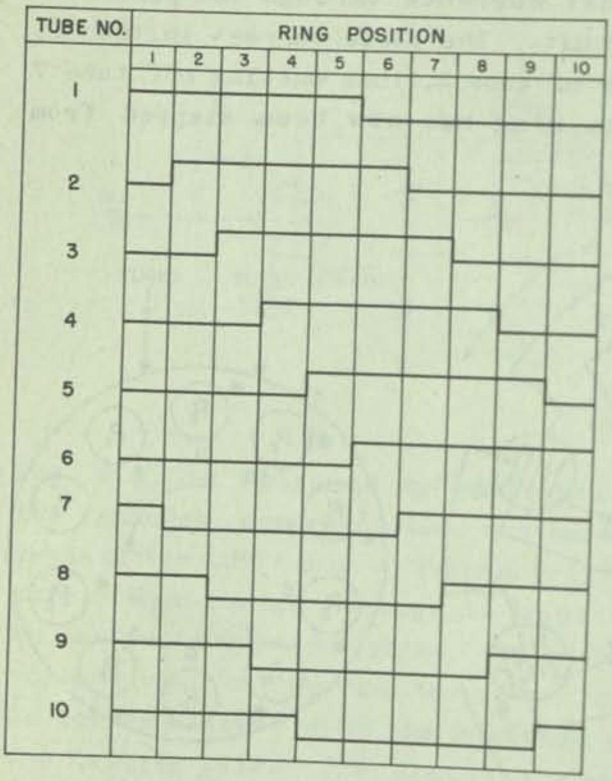


Fig. 2.9—*Grid voltages as a function of ring position.*

As an example of a specific ring circuit used in the calculator, Fig. 2.10 shows the ring of twenty which supplies voltages corresponding to the twenty digit times in a number time. The ring is in its reset position when tubes 11-20 are conducting. It is started at the beginning of each number time when a pulse is delivered from a twin triode gate to the screen grid of tube 1, making tube 1 conduct. The ring is then stepped once each digit time by means of pulses occurring at the start of even and odd digit times. The even digit-time pulses are delivered to the suppressor grids of ring tubes, 1,3,...,19, while the odd digit-time pulses are delivered to the suppressor grids of ring tubes 2,4,...,18. As tube 20 is a triode, the ring stops in its reset position until a starting pulse from the twin triode gate arrives at the screen of tube 1. This insures that the ring will start in phase when the power is first turned on. Voltages from the control grids of the ring tubes are delivered through cathode followers to a system of twin triode gates, Fig. 2.11, delivering output voltages corresponding to each of the twenty

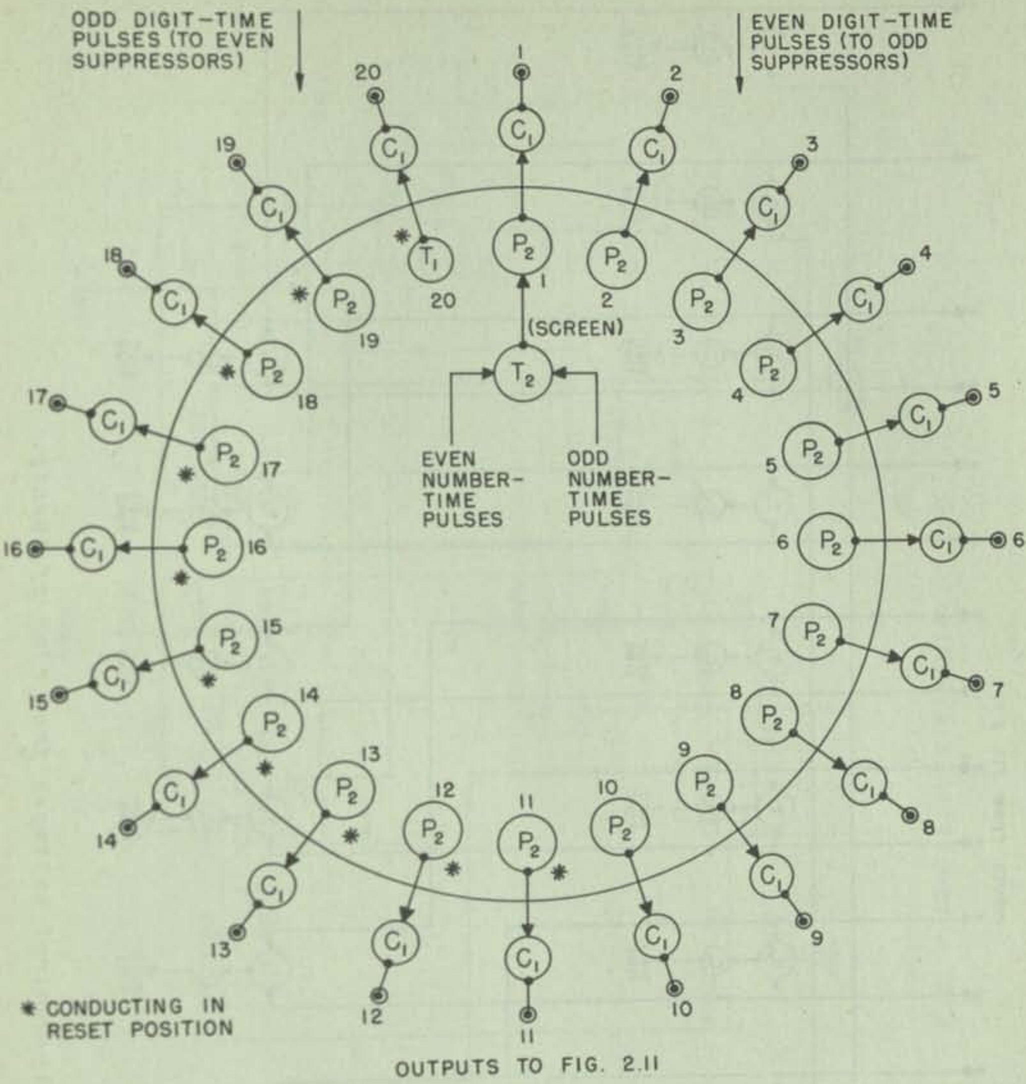


Fig. 2.10—*Ring of twenty.*

digit times. These are numbered -1,0,1,...,16,17,18, in order that digit times 1-16 may be associated with the sixteen decimal digits of a stored quantity.

The signals required to step the ring are derived from pulses permanently recorded on four drum channels. Pulses played back from these channels occur, (1) at the start of odd number times, (2) at the start of even number times, (3) at the start of odd digit times, and (4) at the start of even digit times. The pulses occurring at the start of odd and even number times are required elsewhere in the calculator. By delivering both sets of pulses to the twin triode gate shown, a starting pulse is obtained at the beginning of each number time.

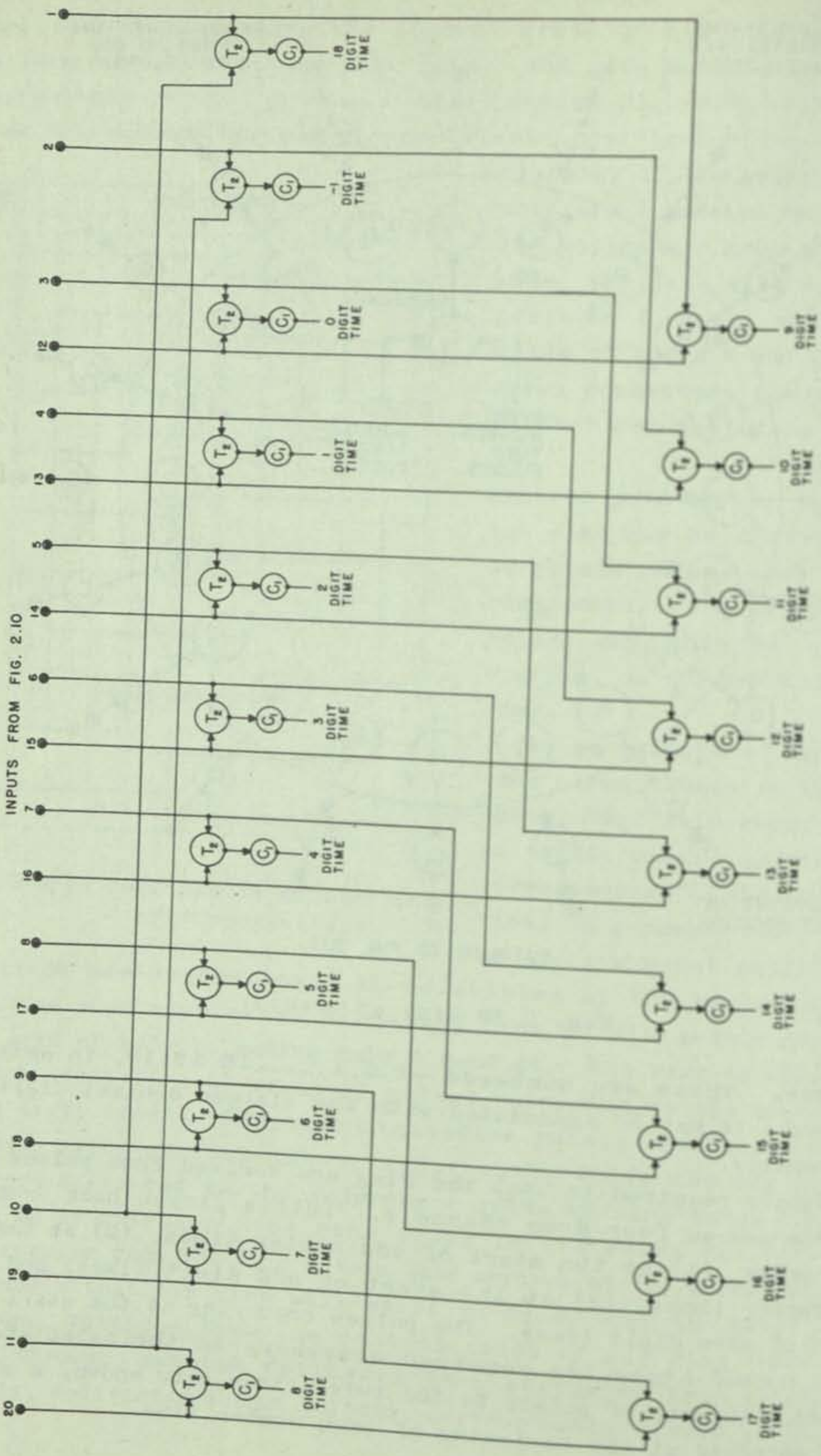


Fig. 2.11—Control voltages from ring of twenty.

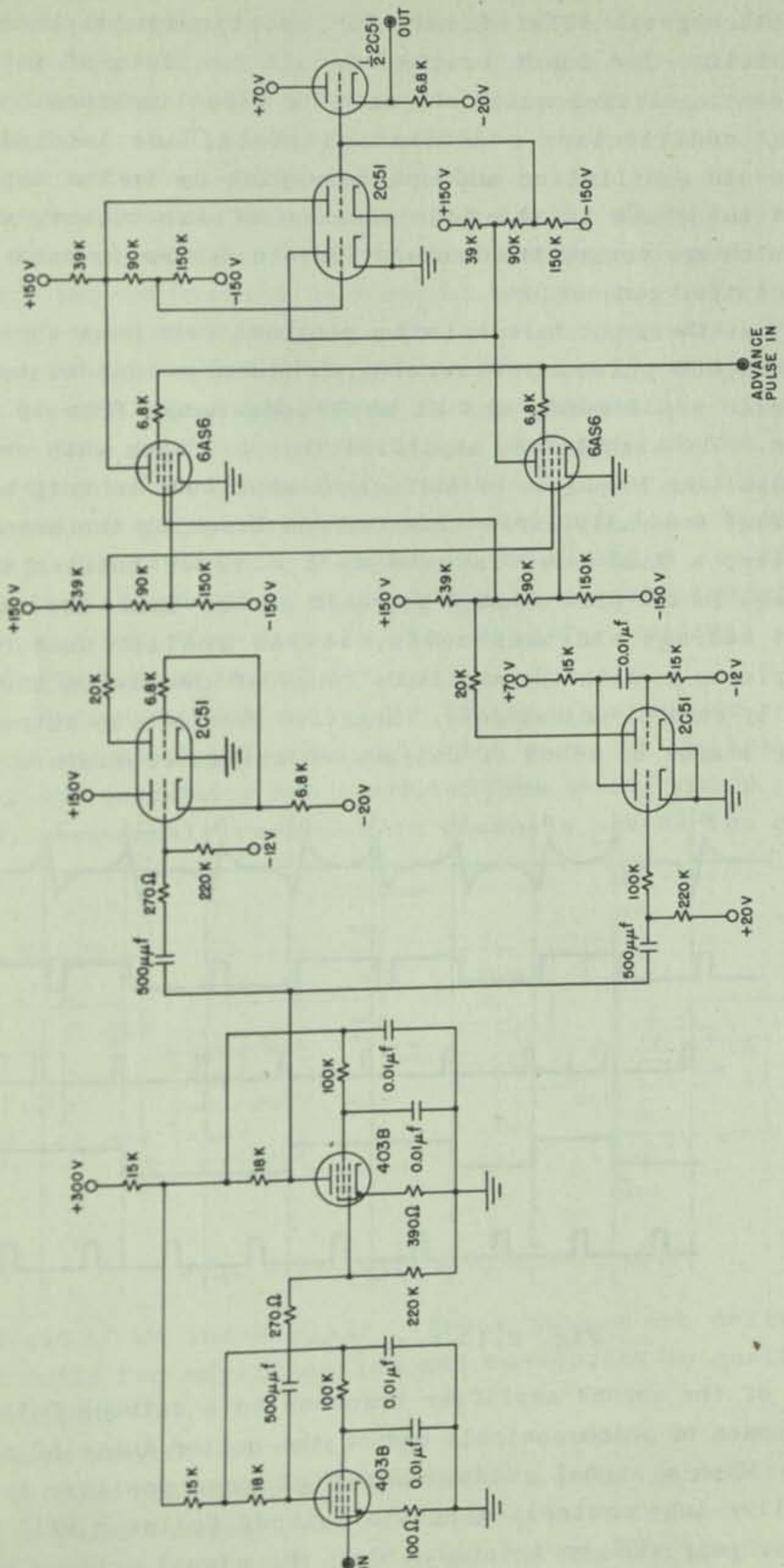


Fig. 2.12—Playback circuit.

5. *Playback unit*. The circuit for one binary playback unit is shown in Fig. 2.12. The input to the circuit consists of information pulses, representing either numerical data or coded instructions. Four such circuits, constituting a decimal playback, are located on one chassis. To avoid oscillation and spurious pick-up in the amplifiers, the four input tubes are in the four corners of the chassis while the four output tubes are toward the center. Shielded wire is used for both the input and output connections.

The signals at the input terminal of a playback unit have a repetition rate of about 28,000 pulses per second, frequency components as high as 150,000 cycles per second, and an amplitude range from 15 or 20 to 100 millivolts. The signals are amplified in two stages with negligible distortion, resulting in pulses of sufficient amplitude to trip a trigger pair. To prevent unusually large signals from blocking the second stage of the amplifier, a 6AL5 tube is used as a clipper between the first and second stage in the slow storage playback units. This tube is omitted from the fast storage playback units, as the smaller number of fast storage pole pieces permits the amplitude range of the played back pulses to be reduced by careful adjustments. Negative feedback is introduced in both amplifier stages by means of unbypassed cathode resistors.

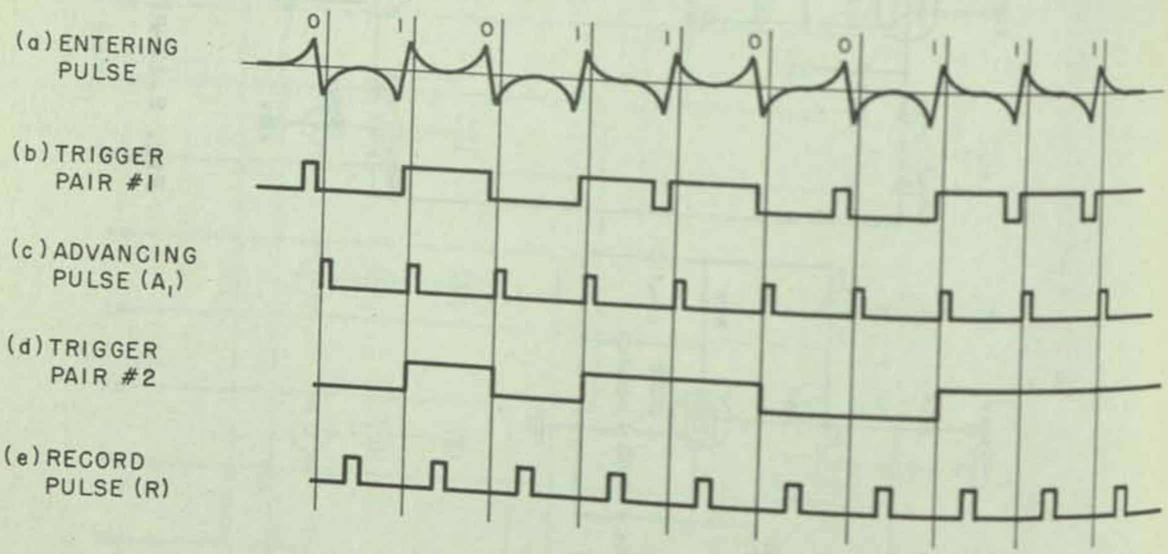


Fig. 2.13—*Wave forms*.

The output of the second amplifier branches to a cathode follower and inverter tube, each of which controls one of the puller tubes of a pentode trigger pair. When a signal voltage rises above a positive threshhold value, the puller tube controlled by the cathode follower will conduct, and the trigger pair will be tripped. When the signal voltage reaches a sufficiently negative value, the normally conducting inverter will be cut off, causing the voltage at the grid of the corresponding puller tube to rise. The trigger pair will then be tripped in the opposite direction. The wave forms in Fig. 2.13(*a*) and (*b*) show how the trigger pair is tripped for a typical series of pulses.

An accurately timed pulse occurring shortly after the crossover point of each entering signal is applied to the suppressor grids of the pentodes to advance the information to a second trigger pair. To insure reliable tripping of the second trigger pair, series resistors are used between the plates of the puller tubes and the screen grids of the pentodes. This makes it possible to read out of the pentode plate circuit even when the puller tube is fully conducting. One grid of the second trigger pair is connected to the grid of a cathode follower, which supplies a low or high voltage output from the playback unit for a 0 or 1 pulse, respectively, Fig. 2.13(*d*). Inasmuch as the timing of the playback output is determined by regular advancing pulses, independent of the exact timing of the entering signals, the need for critical adjustments of playback pole pieces is reduced.

6. *Pulse playback unit*. The timing signals which are required in the record and playback units and which serve to synchronize the operations of the calculator are generated from evenly spaced pulses of like polarity, permanently recorded in channels around the drum periphery

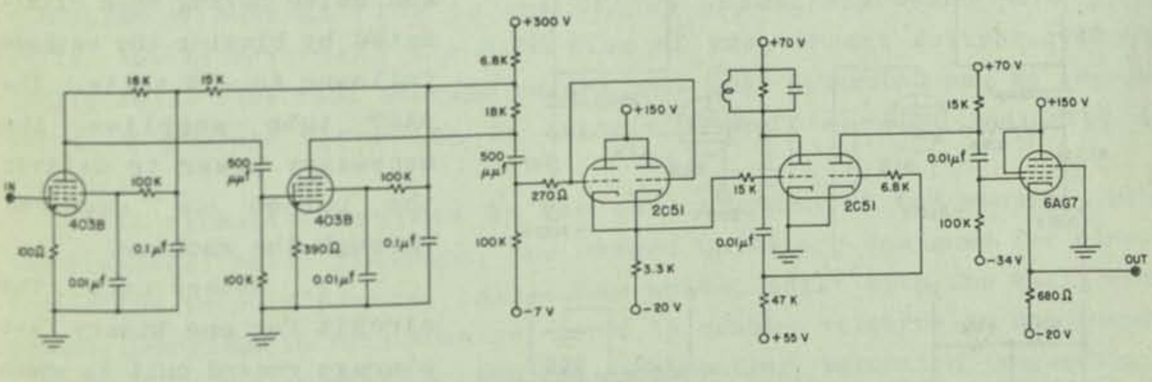


Fig. 2.14—*Pulse playback circuit.*

with the aid of an index plate. These pulses are delivered to pulse playback units for amplification and conversion to positive pulses of short duration.

The pulse playback circuit, Fig. 2.14, consists of a two-stage amplifier, a cathode follower, a triode with a pulse generator in its plate circuit, a pulse inverter, and an output cathode follower. The two-stage amplifier, the same as that used in the information playback unit, feeds a cathode follower, to avoid charging of the coupling condenser. A grid-limiting resistor is used in series with the grid of the pulse generator tube. This tube begins to conduct near the crossover point of an entering signal and is turned on very rapidly, being fully conducting after the amplified signal has risen only 3 or 4 volts. The tube remains fully conducting for 6 to 8 microseconds, but the duration of the negative output pulse is determined by the tuned L-C circuit. A 4-microsecond advancing pulse is obtained with an inductance of 10 millihenrys and a capacitance of 150 micromicrofarads, while for a 6-microsecond record pulse the values are 16 millihenrys and 250 micromicrofarads, respectively. The resistor in parallel with the condenser and inductor serves to damp the circuit, preventing unwanted oscillations. The final output from the cathode follower is a positive square-wave signal, the background ripple and noise having been eliminated by biasing the cathode follower to -34 volts. The 6AG7 tube supplies the necessary power to deliver the pulses as required through the machine.

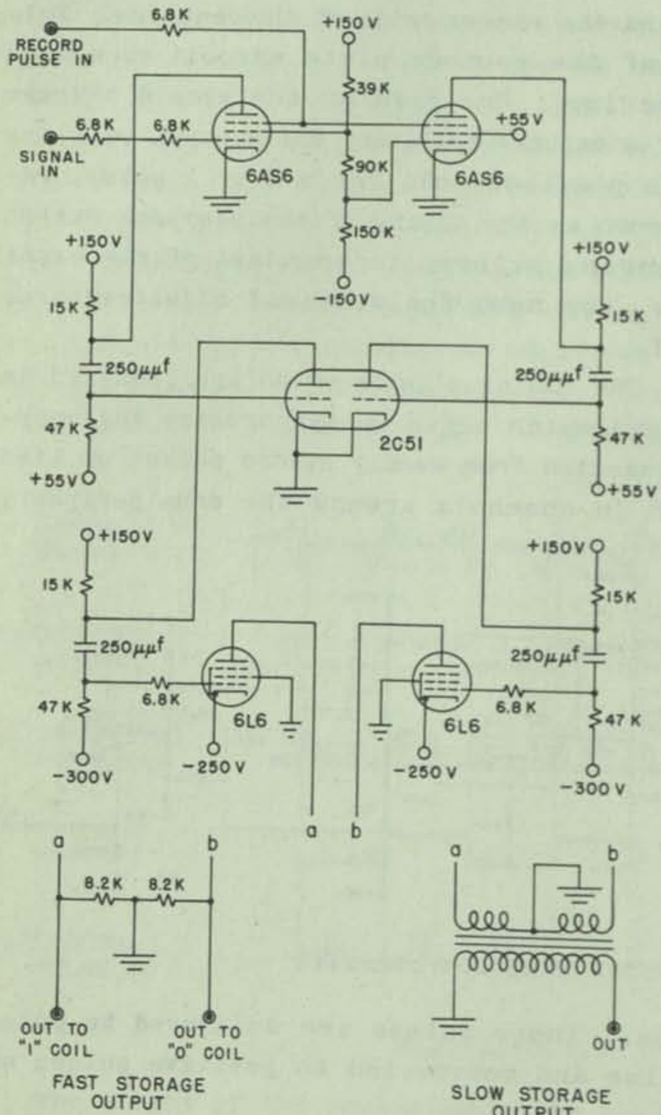


Fig. 2.15—*Record circuit.*

7. *Record unit.* The circuit for one binary fast storage record unit is shown in Fig. 2.15. The input voltage is delivered to a pair of mutually inverted gates of the type previously described. A record pulse applied to the suppressor grids of the pentodes is gated to one of the inverters

in the 2C51 envelope, which in turn drives the grid of the corresponding power tube. The duration of the record pulse is 6 microseconds, to insure that the flux in the pole piece core reaches a maximum. A damping resistor is provided in parallel with each recording coil, to reduce the oscillations which occur when the power tube is suddenly cut off. Eight binary record circuits are located on one chassis. To prevent feedback and cross talk, the pulse input and output lines are shielded, as are the connections between the 6L6 power tubes and the component container from which the output is taken.

It should be noted that since the record and playback pole pieces are in close proximity on the fast storage drum and since recording involves a leakage flux which makes complete shielding impossible, some record voltage will be induced in nearby playback pole pieces. However, the relative location of record and playback pole pieces is such that signals are played back and sensed by advancing pulses, Fig. 2.13(*a*) and (*c*), in the interval between record pulses, Fig. 2.13(*e*). Thus interference from record pulses does not affect the playback outputs.

To reduce the number of relay contacts required in the slow storage selection circuits, only one coil is provided on each slow storage pole piece. While this permits playback operations to be carried out as in fast storage, the recording procedure is necessarily different. The polarity of the recorded pulse is determined not by gating the same direction of current flow to one of two oppositely wound pole piece coils, but by reversing the direction of the current delivered to the single coil. The fast storage record unit just described may be adapted for slow storage recording by using a transformer, as indicated in Fig. 2.15.

Of the circuits required in the construction of the machine, only the playback, pulse playback, and record units are designed for direct connection to pole pieces. All other units, built with the basic components described in this chapter, serve to combine voltages in accordance with the logical operations required in computing, selection, sequencing, and control systems, making up the subject matter of succeeding chapters.

REFERENCES

1. *Description of a Relay Calculator*, Annals of the Computation Laboratory, Harvard University, XXIV; Cambridge: Harvard University Press (1949)

2. *Synthesis of Electronic Computing and Control Circuits*, Annals of the Computation Laboratory, Harvard University, XXVII; Cambridge: Harvard University Press (1951)

3. Moore, Benjamin L., Pentode Counting or Control Ring, *Review of Scientific Instruments*, 21, 4, page 337 (April, 1950)

CHAPTER III

STORAGE SYSTEM

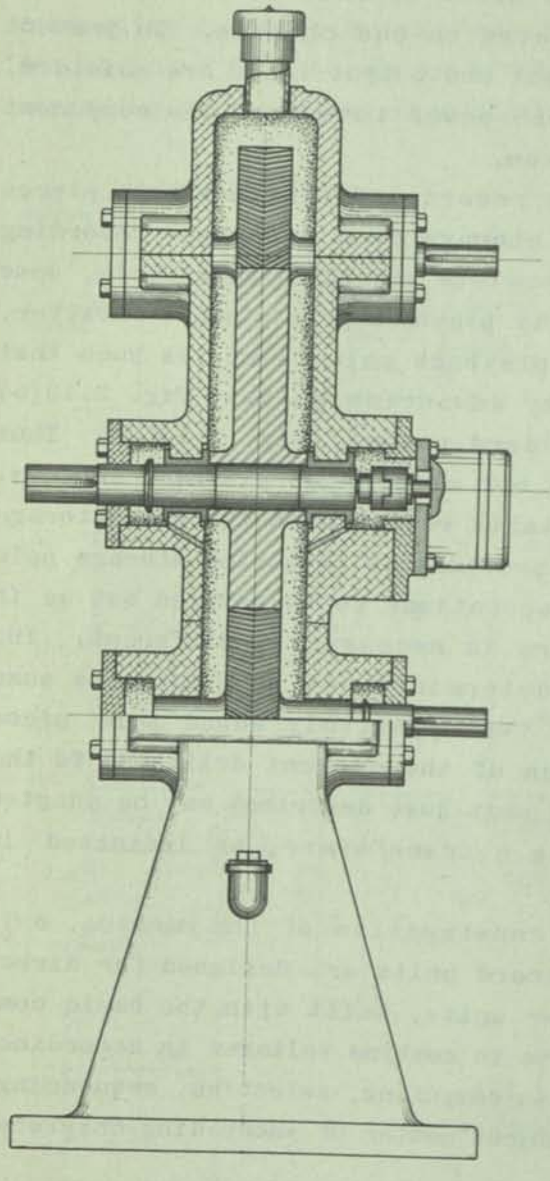


Fig. 3.1—*Gear box.*

The general arrangement of the magnetic drum storage unit has already been shown in Fig. 1.1. The 15 hp., 3450 rpm wound rotor induction motor is connected to the input shaft of the gear box, which contains one driving gear and four driven gears, Fig. 3.1 and Plate XVII. The four output shafts of the gear box drive the four pairs of drums. The gear box serves as a speed-doubling unit as well, providing a drum speed of about 6900 rpm. To minimize noise, herringbone gears and sleeve bearings are used in the gear box. A rotary pump coupled to the drive gear shaft supplies force-feed lubrication for the meshing gear teeth and for the bearings. A pressure alarm is provided which will turn off the power in the event of a failure in the lubrication system.

Each of the storage drums is 8⅛ inches in diameter, 40 inches in length, and weighs about 240 pounds. The drums are made from forged aluminum alloy billets, specially processed at the foundry to insure homogeneous forgings. Steel stub shafts are inserted on screw threads in the ends of each drum, Fig. 3.2. The drum assemblies have been carefully machined, ground, and dynamically balanced. As a result, the maximum total eccentricity on any one drum is 0.0003 inch, and the maximum amount of unbalance, 0.2 gram.

The drums are mounted on grease-lubricated, precision ball bearings,

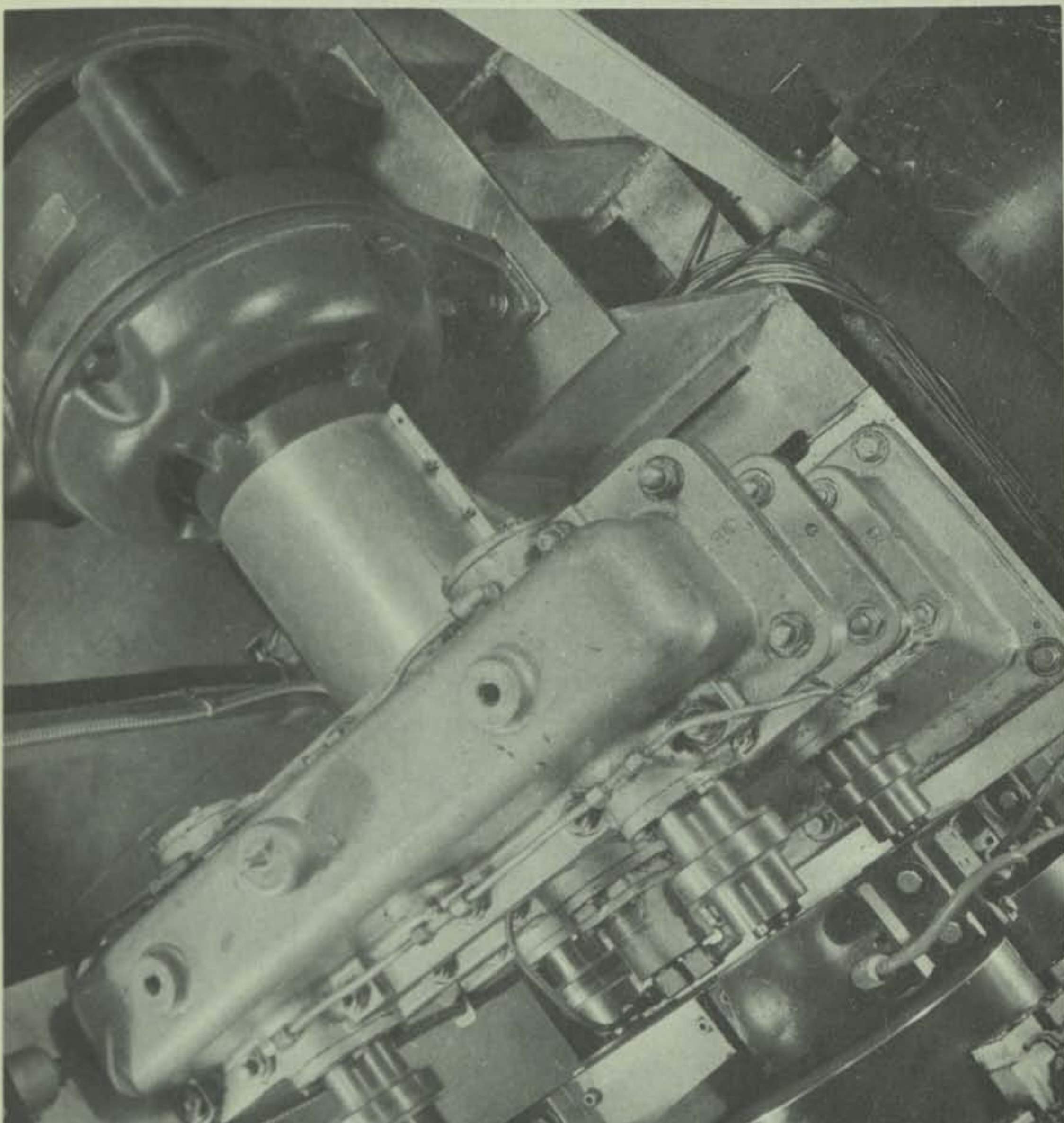


Plate XVII *Gear Box*

which are housed in cast-iron pillow blocks, Fig. 3.2. A thermocouple is mounted on the cap of each pillow block to turn off the power if the

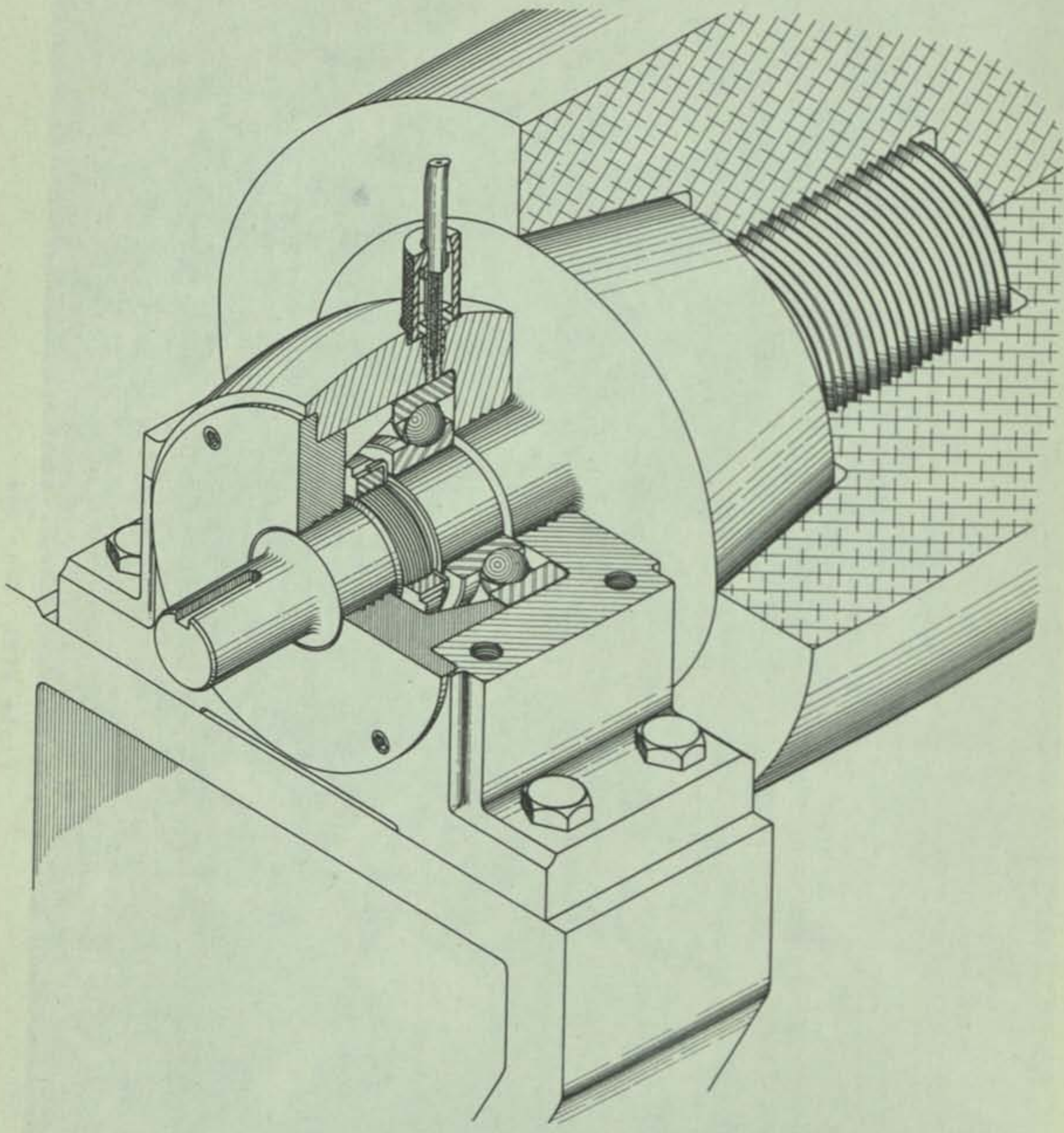


Fig. 3.2—*Pillow block and stub shafts.*

bearings become overheated. Cast-iron pedestals are provided to hold the bearing supports as well as the pole piece brackets required on each drum. Different types of brackets are used, both circular and semi-circular, depending on the pole piece arrangements on different channels.

On all the pole pieces, clearance is provided around the mounting screws to permit adjustment of the gap between the drum surface and the pole piece tips. The leads from the pole pieces are brought out to connector plates fastened onto each bracket.

The drum surfaces are sprayed first with a layer of bonding cement, and then with several applications of the iron oxide mixture described in Chapter I. They are then covered with a second layer of cement, buffed at high speed to produce a hard surface.

The mechanical connections between drums are geared flexible couplings which compensate for misalignment and restrict the torsional displacement to 0.001 inch or less on the meshing surface of the coupling. However, to insure accurate magnetic recording, it is necessary to record on each drum under control of pulses derived from the same drum. For this purpose a channel of record pulses is permanently stored on each drum in the unit.

The drums are numbered for reference as indicated in Fig. 1.7. Drums 1, 2, 3, 5, 6, and 8 contain the slow storage channels, while drum 4 contains the fast storage channels, $a,b,\ldots,z,\gamma$, and the two channels, α and β, reserved for transfers from slow to fast storage. Drum 7 contains the fifteen channels of constants, the channel associated with the manual register on the front panel, the transfer channels, the timing pulses, and special channels required in the arithmetic units and the tape read-record system.

Before describing the storage system, it is helpful to investigate the timing pulses permanently recorded on drum 7. When these are delivered to pulse playback units, the played back pulses occur at the times listed in Table 3.1, and, either directly or through ring and gate circuits, provide timing controls for the operation of the calculator as a whole. Basically, there are three sets of control pulses, A_1, R, and A_2, having the same repetition rate but displaced in time as shown in Fig. 3.3. Thus each A_1 pulse defines the start of a digit time, while the A_1 pulse occurring at -1 digit time, called the $-1A_1$ pulse, defines the start of a number time.

The pulses occurring at the start of even and odd digit times and at the start of even and odd number times control the digit-time ring of twenty described in Chapter II. A ring of six, Fig. 3.4, driven by the pulses at the start of even and odd number times, supplies voltages lasting through each of the six number times in a half-revolution of the drum. To insure that the ring starts in phase when the power is first turned on, the suppressor grid of tube 4 is pulsed only at the start

TABLE 3.1	
Timing Pulses on Drum 7	
Time of Pulse	Use
A_1	Playback
R	Record
A_2	Delay for C transfer channel
A_1, even digit times A_1, odd digit times	Digit-time ring
$-1A_1$, even number times $-1A_1$, odd number times	Digit-time ring, Number-time ring
$-1A_1$, blank number times $-1A_1$, blank number time, "even" drum cycle	Indication of drum phase

of each blank number time. The outputs of this ring will be referred to as "number-time voltages"; and similarly the name, "digit-time voltages," will refer to the outputs of the ring of twenty described in Chapter II.

The purpose of the ring of four in Fig. 3.4 is to indicate the drum phase as an "even" or "odd" half-revolution. Two outputs are delivered, one lasting from the start of blank number time to the start of the following blank number time, and the other occurring one number time later, from the start of 0 number time to the start of the following 0 number time. These two voltages are used in controlling playback and record operations, respectively, and correspond to the displacement of approximately one number time between record and playback pole pieces on the fast storage channels. The purpose of the pulses derived in Fig. 3.5

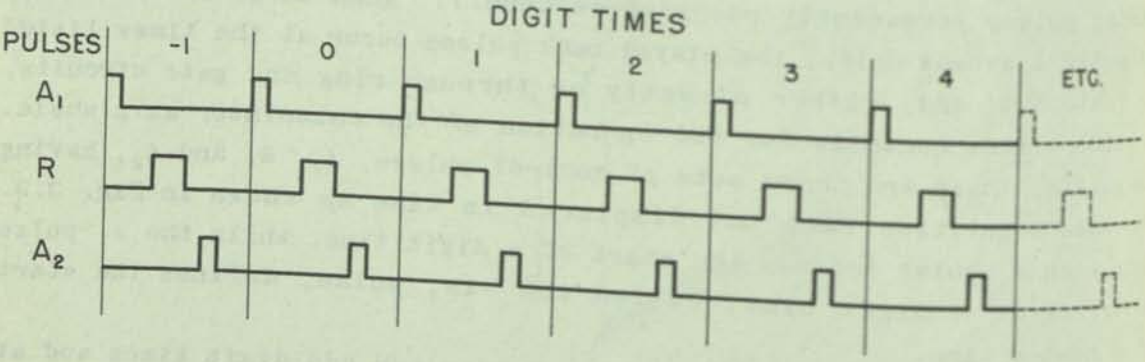


Fig. 3.3—*Timing for A_1, R, and A_2 pulses.*

will be seen in later descriptions of special circuits in the calculator.

To begin the discussion of the storage system, reference is made to Table 3.2, which lists the fast storage channels, their code numbers, and their bus connections.

The fifteen constant channels store the quantities listed in Appendix I. Although many of the constants required for the evaluation of the elementary functions are generally selected under control of the

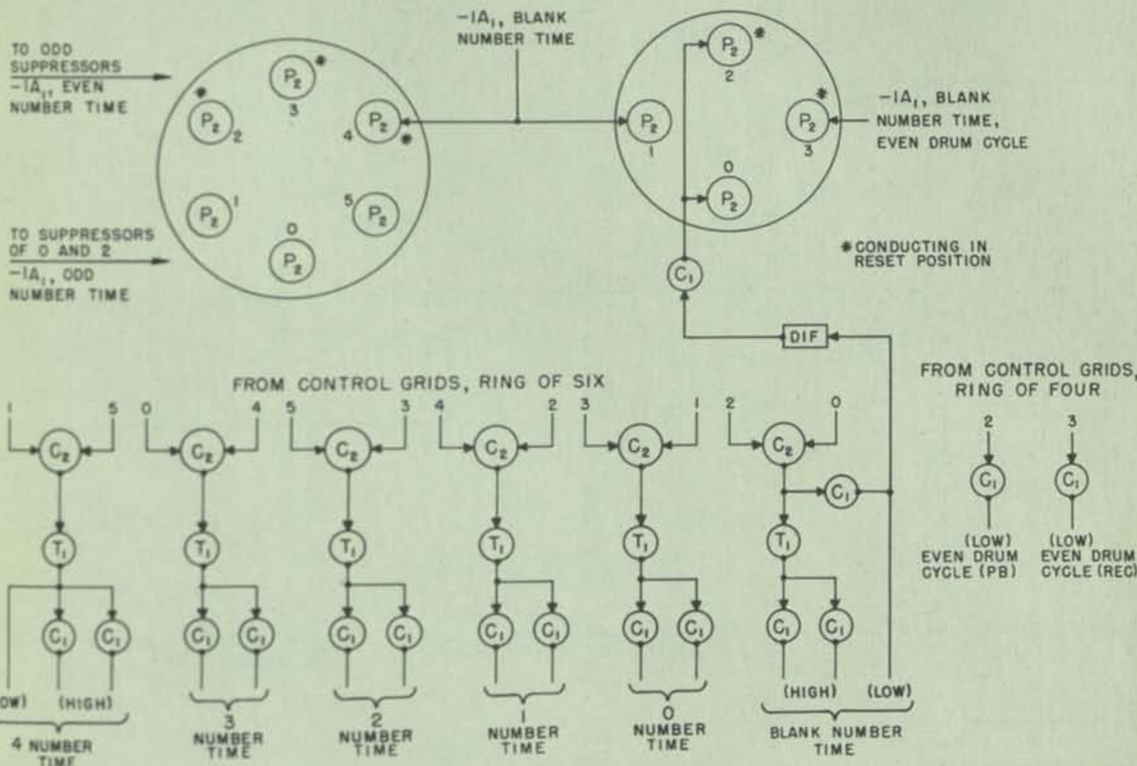


Fig. 3.4—*Number-time ring.*

function sensing register, any of them may be selected directly under code control as well. Circular brackets, Plate XVIII, are used to permit two diametrically opposite playback pole pieces to be mounted on each binary channel in the group. The numbers were recorded during the construction of the calculator by temporarily connecting the playback pole pieces to record units.

Table 3.3 lists the powers of 10 stored on channels 16-19. These values are arranged to facilitate selection under control of the i register for shifting operations. Like the constants, these numbers were recorded during the construction of the calculator. Two playback pole pieces are mounted on each channel, but since a 1, in 2*,4,2,1 notation,

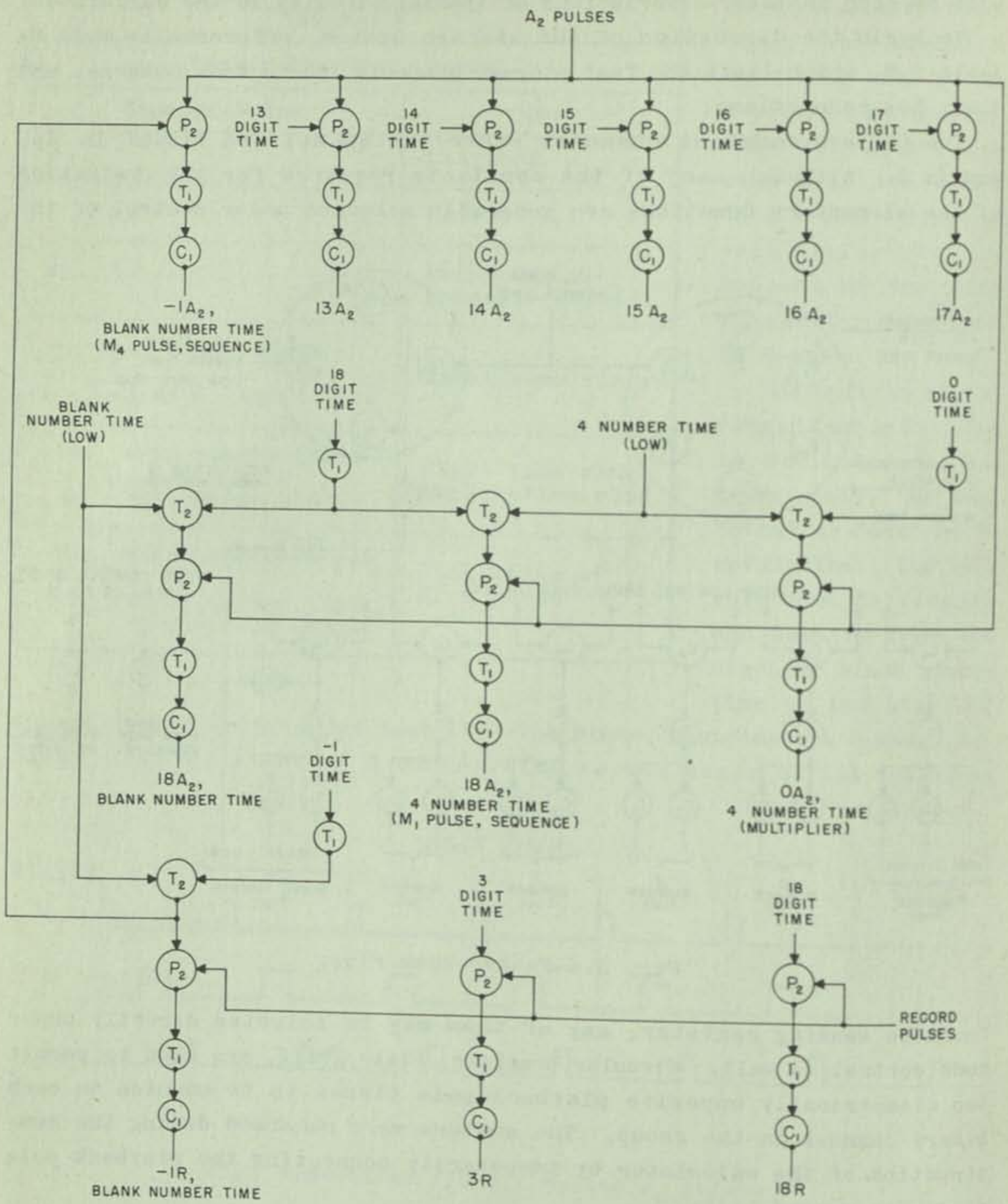


Fig. 3.5—*Cycling pulses.*

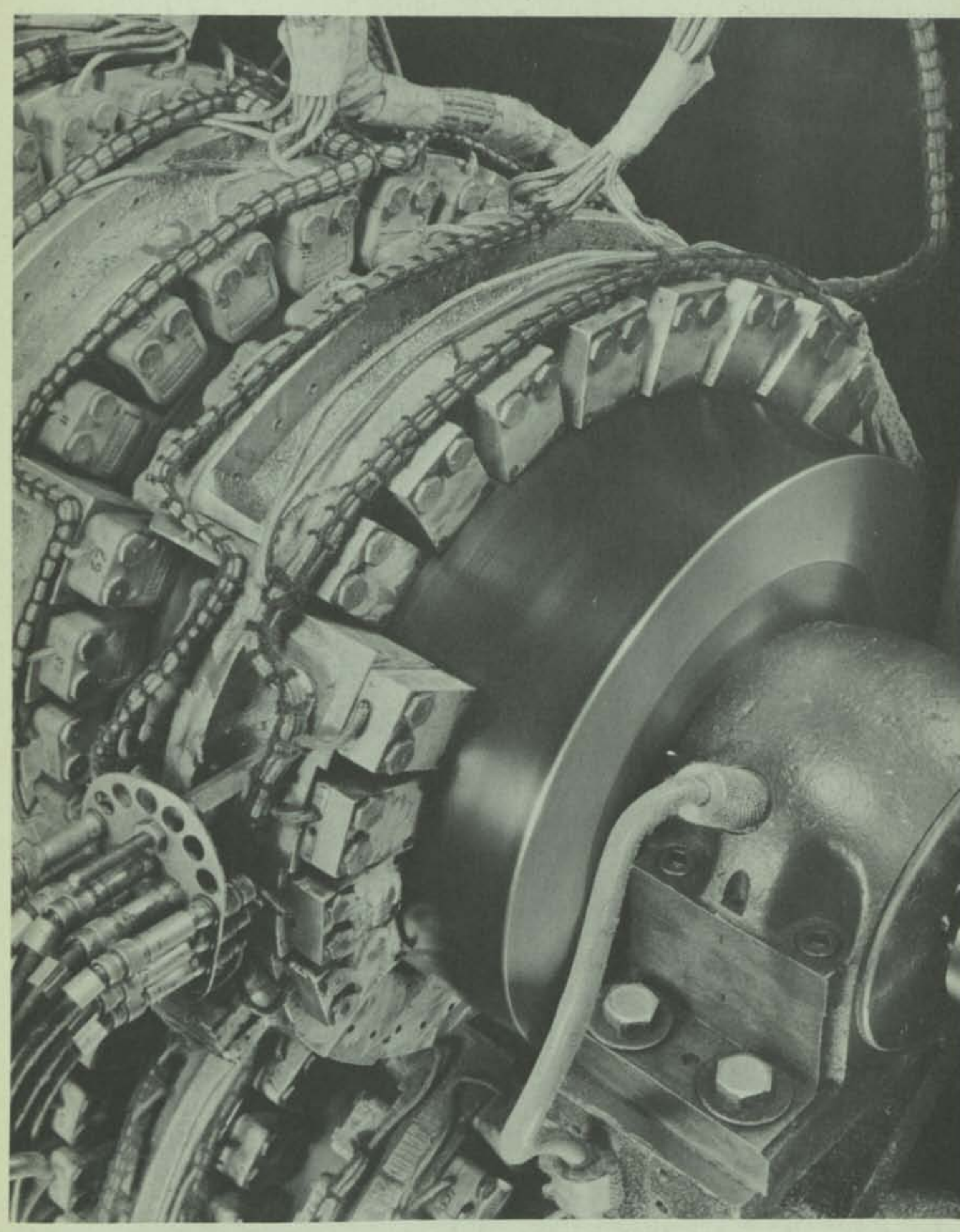


Plate XVIII *Constant Storage Bracket*

is 0001, a single binary channel of zeros replaces the three channels for the 2*, 4, and 2 components usually required in decimal channels.

Registers on channels $a,b,\ldots,z,\gamma$, may be selected by code for either record or playback operations. To mount two record and two playback pole pieces on each binary channel, the circular brackets shown in

TABLE 3.2			
Fast Storage Channels			
Channels	Codes	Playback Connections	Record Connections
constants	01 . . . 15	B bus	
powers of 10 required for shifts	16 . . . 19	B bus	
a b . . . z γ	20 21 . . . 38 39	A or B bus	C bus
α β	03* 04*	A or B bus	slow storage
manual register	05*	A or B bus	manual register on front panel
tape read-record (a) channels (b)	A10, C02* B02*, C02*	A bus B bus	C bus C bus

Plate XIX are used. Each record pole piece is advanced 20.7 digit times in the direction of drum rotation from the corresponding playback pole piece. This insures that digits are returned to storage in the correct time phase, taking into account the fact that record pulses occur approximately one-third of a digit time later than the A_1 pulses used in playback units.

On the γ channel, the third playback pole piece used in fast to slow storage transfers is ten digit times in advance of the upper playback pole piece. In order that numbers transferred to slow storage may be

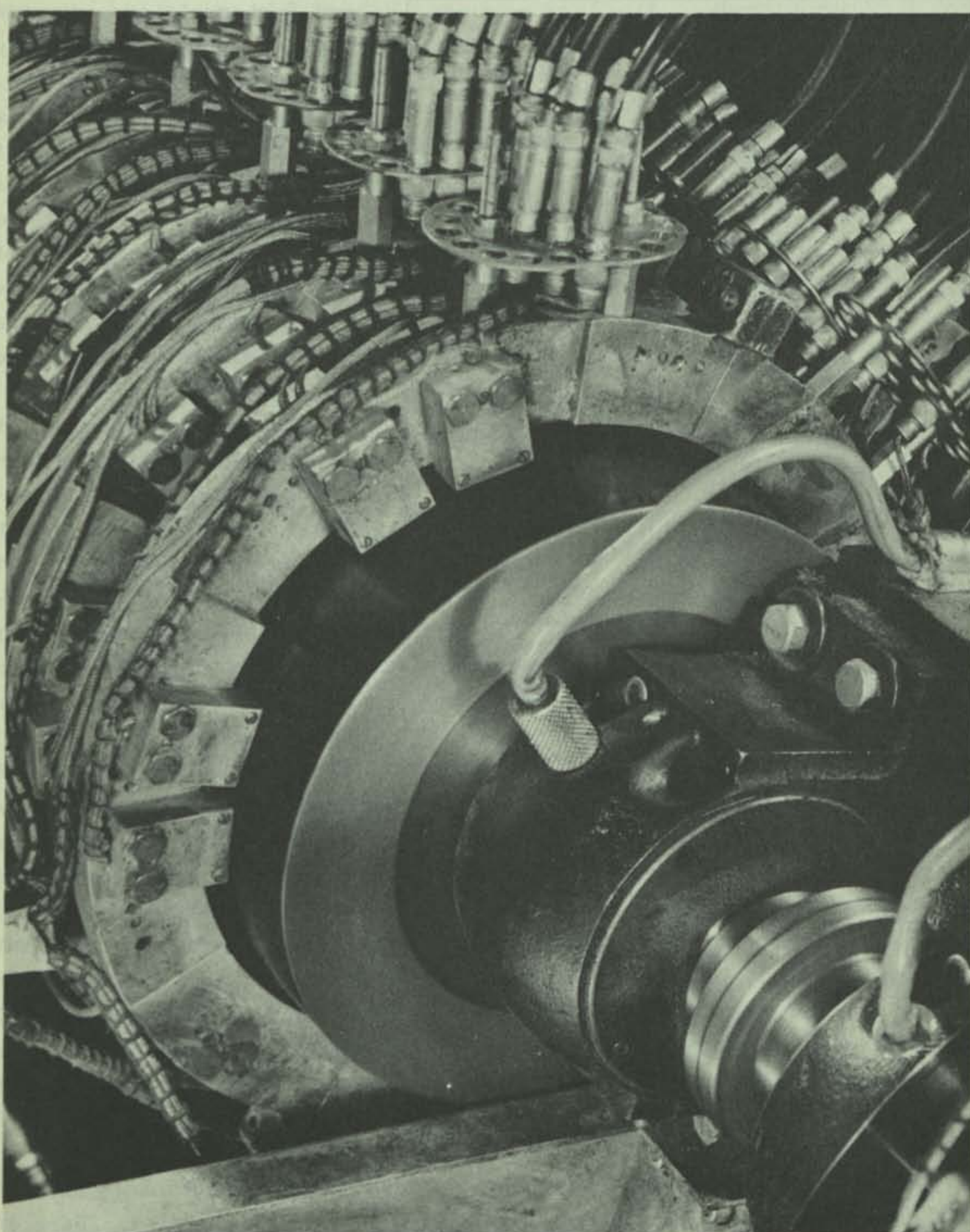


Plate XIX *Fast Storage Bracket*

TABLE 3.3		
Numbers on Channels 16-19		
Channel	Number Time	Quantity Stored
16	0 1 2 . . . 9	10^0 10^1 10^2 . . . 10^9
17	0 1 2 . . . 5 8	10^{10} 10^{11} 10^{12} . . . 10^{15} 0
18	0 1 2 . . . 9	10^0 10^{15} 10^{14} . . . 10^7
19	0 1 2 . . . 5	10^6 10^5 10^4 . . . 10^1

returned to fast storage in the same time phase, the record pole pieces on the α and β channels are advanced 11.7 digit times from the upper playback pole pieces. This spacing compensates for the delay of one digit time which occurs when signals are recorded in slow storage and played back through the same pole piece, as well as for the timing difference in record and playback pulses.

Numbers are recorded in the constant register channel using manual controls on the front panel of the calculator. A set of strip switches is provided there for registering a sixteen-digit quantity and its algebraic sign. When a number-time button, 0-9, and the button labeled "read in" are depressed, a rotary step switch is started, which scans the strip switch contacts, Fig. 3.6. Through the first four step switch wipers, the binary components of each decimal digit are delivered to the record units associated with the constant register channel. Record pulses are gated to these units as shown in Fig. 3.7 under control of a digit time voltage, supplied through wiper 5 on the step switch, and a number time voltage, corresponding to the depressed number-time button. The cam timing which controls the step switch is slow enough to insure the arrival of at least one gated record pulse on each step of the step switch.

The two tape read-record channels, used in transferring quantities to and from magnetic tape, are described in detail in Chapter VI. Here it will be noted that the single code, *C02**, causes the number on the *C* bus

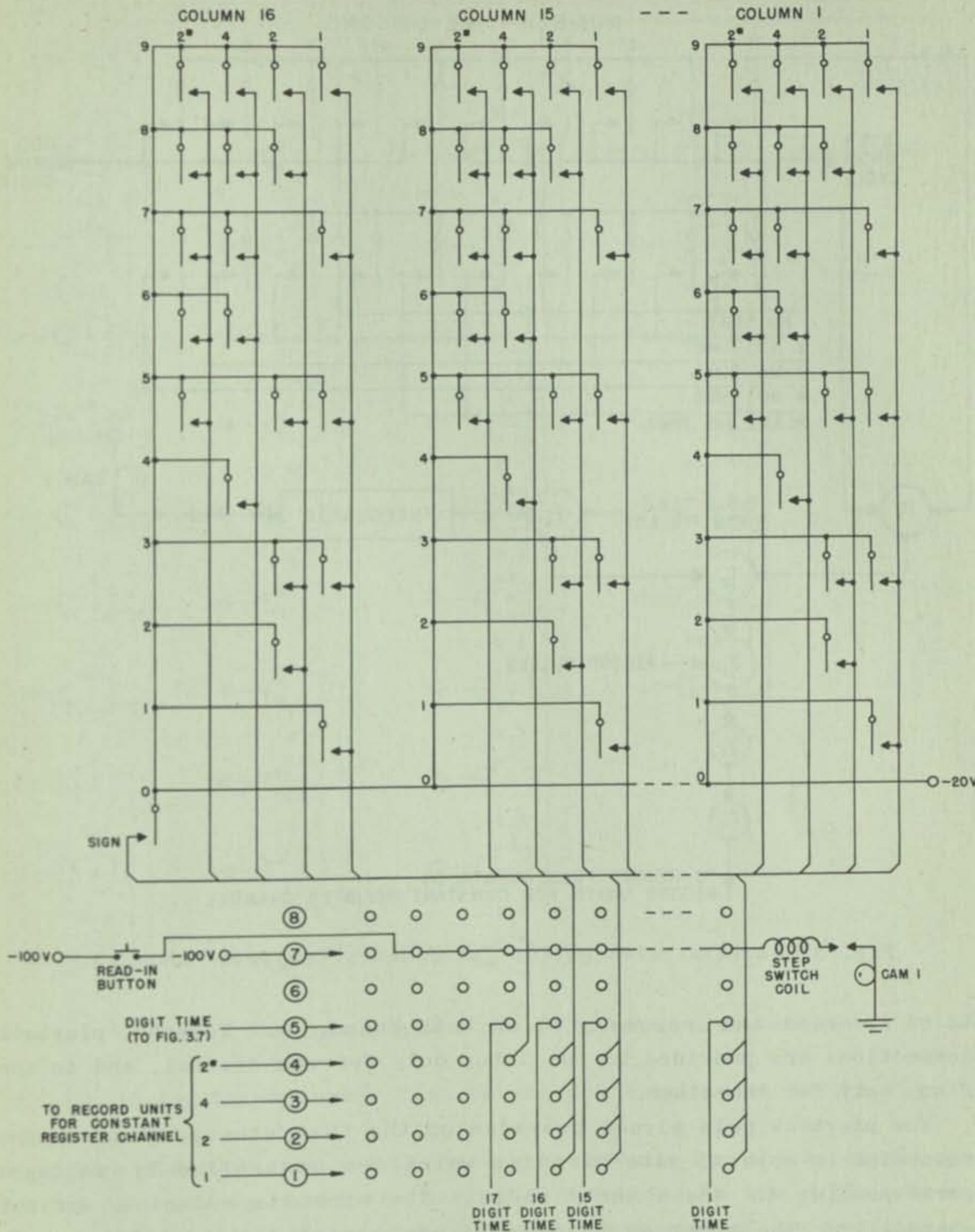


Fig. 3.6—*Strip switches and step switch for constant register channel.*

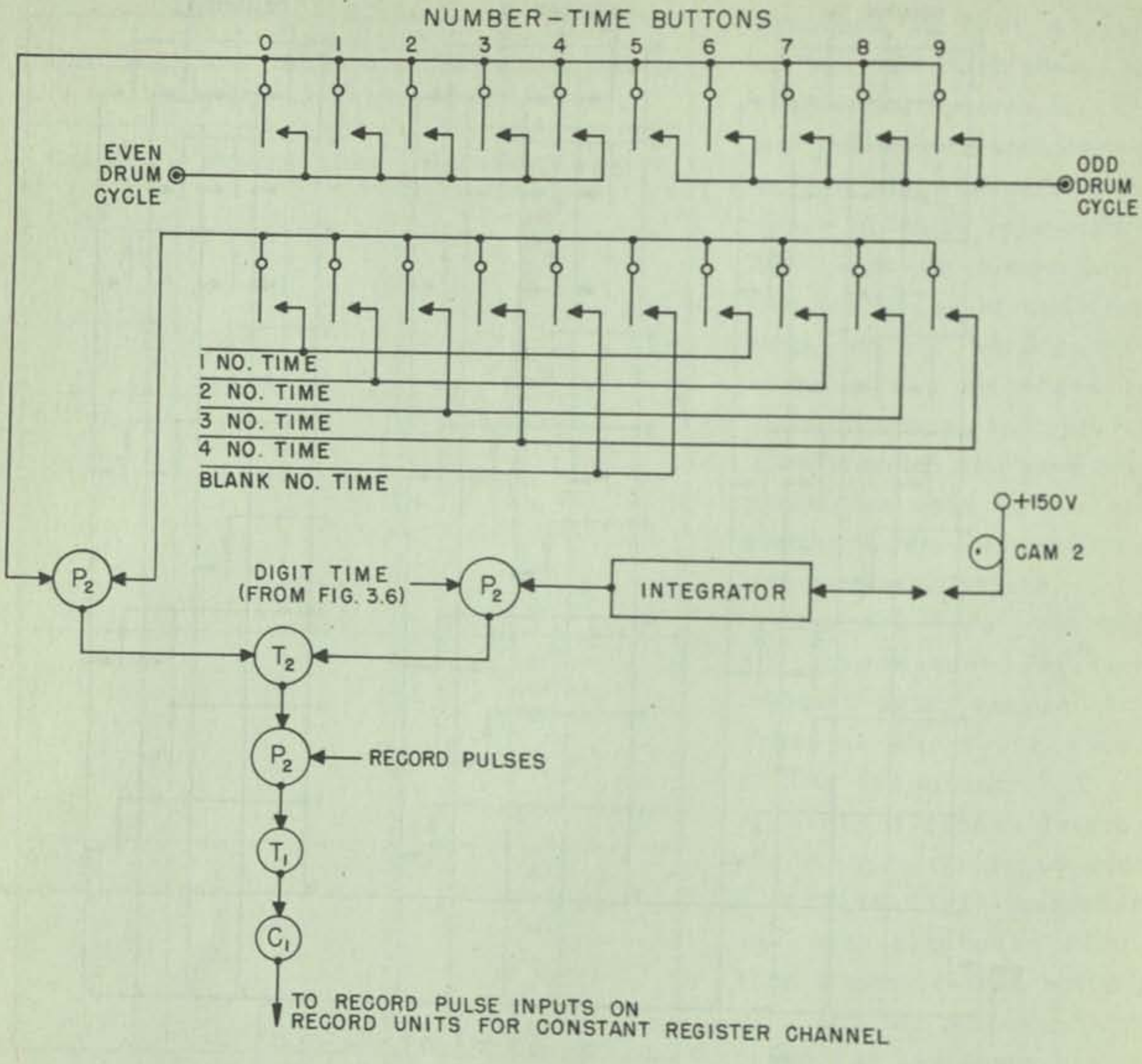


Fig. 3.7—*Record pulse gating for constant register channel.*

to be recorded and regenerated on both channels. However, playback connections are provided to the *A* bus only for one channel, and to the *B* bus only for the other.

The playback pole pieces provided on the fast storage channels are connected to crystal gate circuits which are controlled by voltages corresponding to the *A* and *B* codes. The circuits select a decimal channel and the upper or lower pole piece on this channel, causing signals to be played back to preamplifiers, each consisting of a 403B tube and a cathode follower. The signals are then delivered to the *A* or *B* playback units. Since the gating voltages last one machine cycle starting at blank number time, the transients introduced by them will

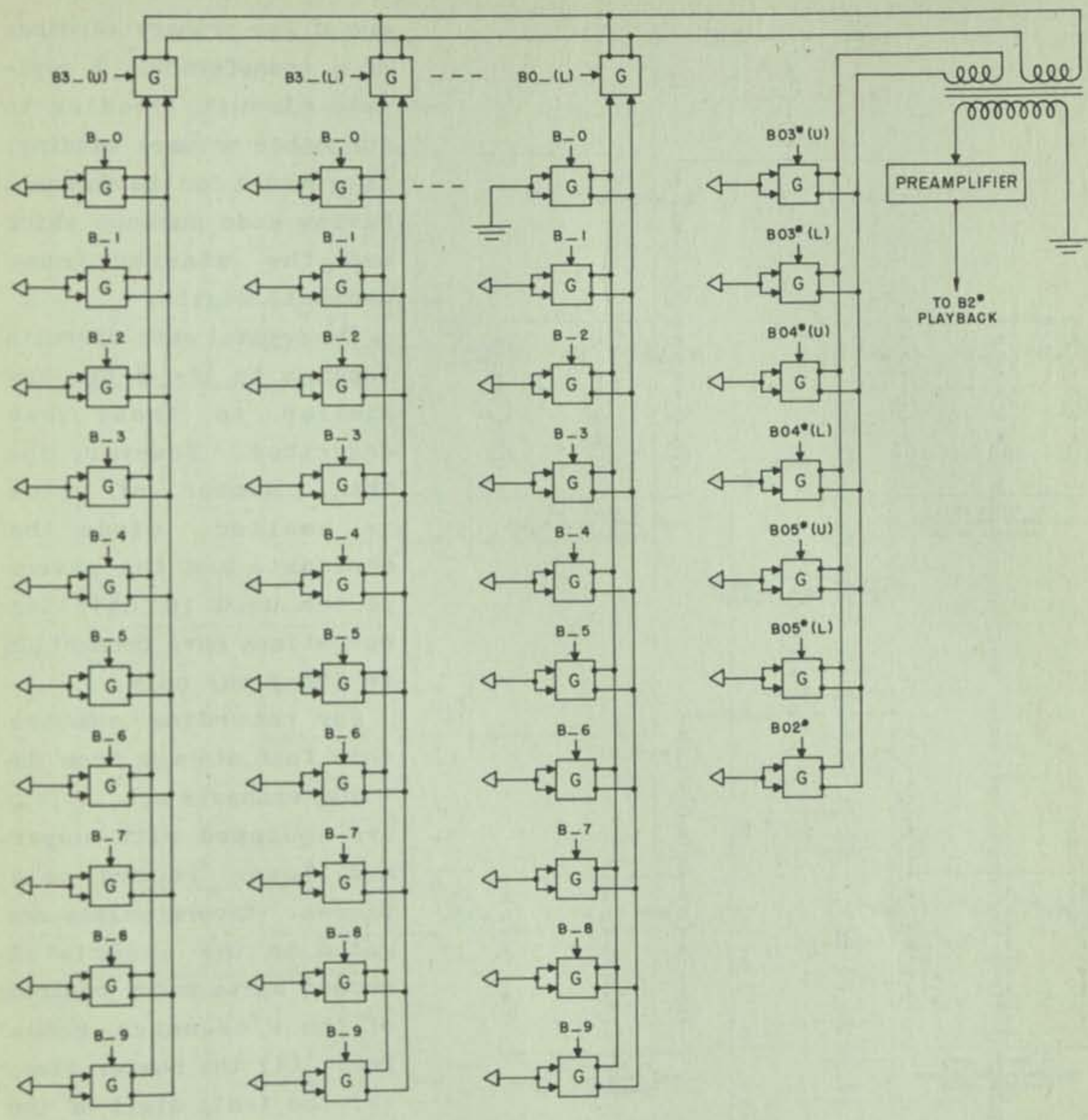


Fig. 3.8—*Playback gates for 2* components to B bus.*

die out during blank number time before any significant information is played back.

Figure 3.8 indicates the playback gate circuits for the 2* components of the fast storage channels having connections to the *B* bus. There are similar circuits for the 4, 2, and 1 components of these channels. Signals from channels 01-39 are gated under control of the *B* sequence codes for: (1) the unit's digit of the channel number; and (2) the ten's digit of the channel number and the choice of upper or lower playback pole piece. The output of the second group of gates is connected to

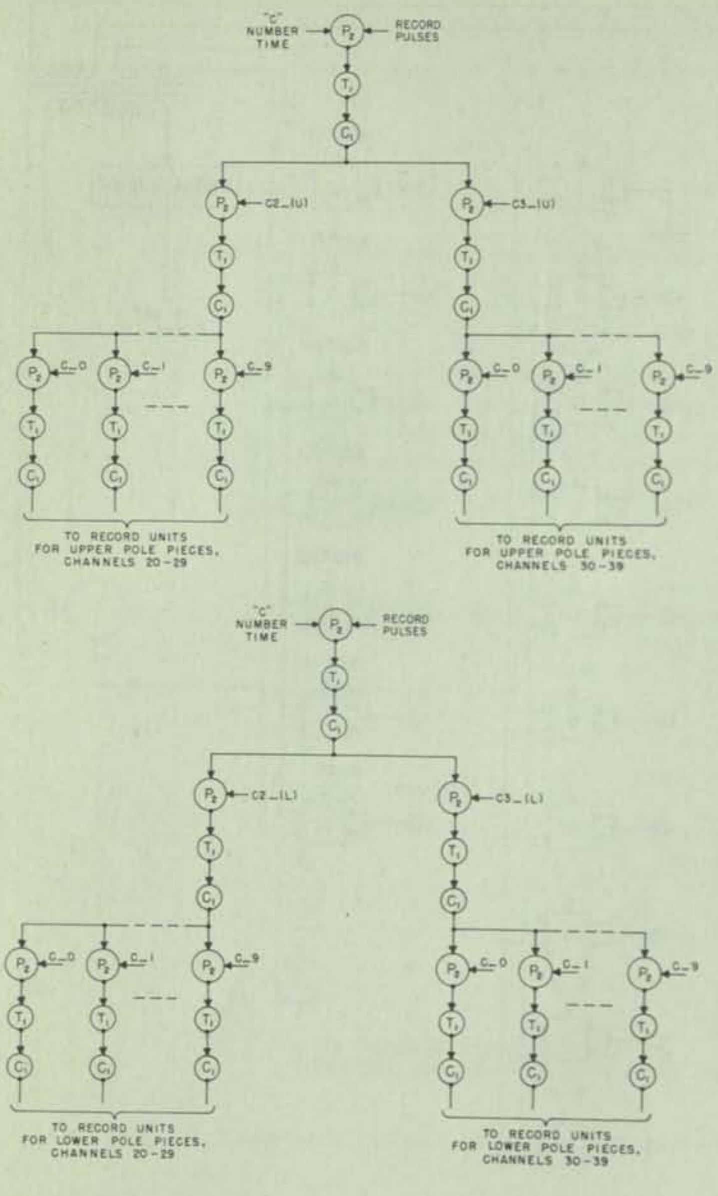


Fig. 3.9—*Fast storage record pulse gating.*

one of two primary windings on a transformer. A separate circuit, leading to the other primary winding, is provided for the channels having code numbers which use the starred (non-decimal) digits.

The crystal gate circuits leading to the *A* bus are similar to those just described. However, the total number of gates is smaller, since the constants and the powers of ten used in shifting operations have connection to the *B* bus only.

For recording numbers into fast storage from the *C* bus, channels $a,b,\ldots,z,\gamma$ are equipped with upper and lower record pole pieces. Record pulses are gated to the associated record units under control of the *C* sequence codes for: (1) the number time; (2) the ten's digit of the channel number and the choice of upper or lower pole piece; and (3) the unit's digit of the channel number. The circuit is indicated in Fig. 3.9. The *C* control voltages are delayed one cycle and one number time, relative to the *A* and *B* control voltages of a sequence command. The one-cycle delay is required so that when, for example, two numbers are selected from storage during cycle n, their sum may be returned to storage in cycle $n+1$, at the same time that two new operands are being selected. The further delay of the *C* control voltages till 0 number time of cycle $n+1$ compensates for the fact that each record pole piece on channels $a,b,\ldots,z,\gamma$, is advanced approximately

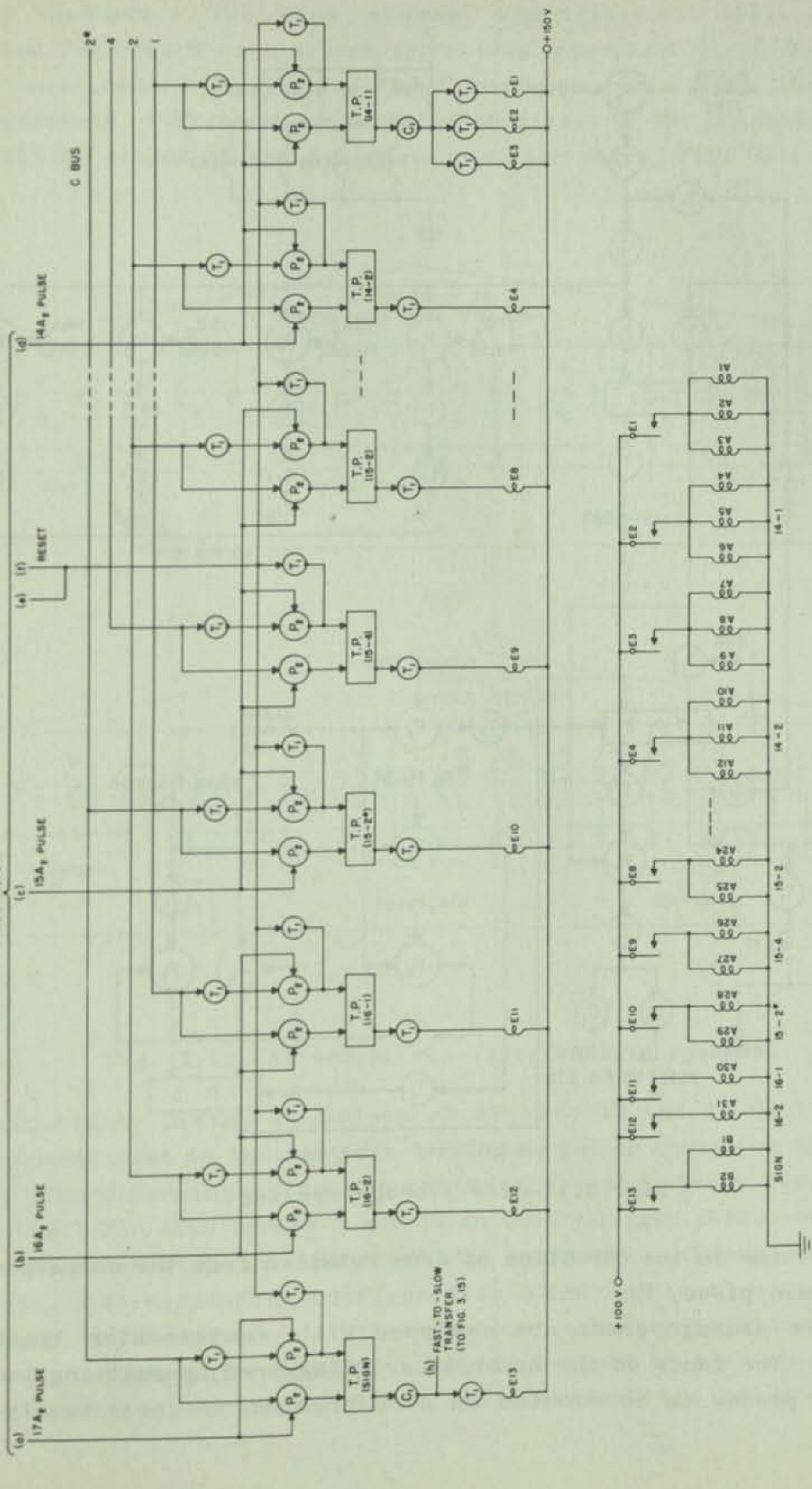


Fig. 3.10—δ register storage trigger pairs.

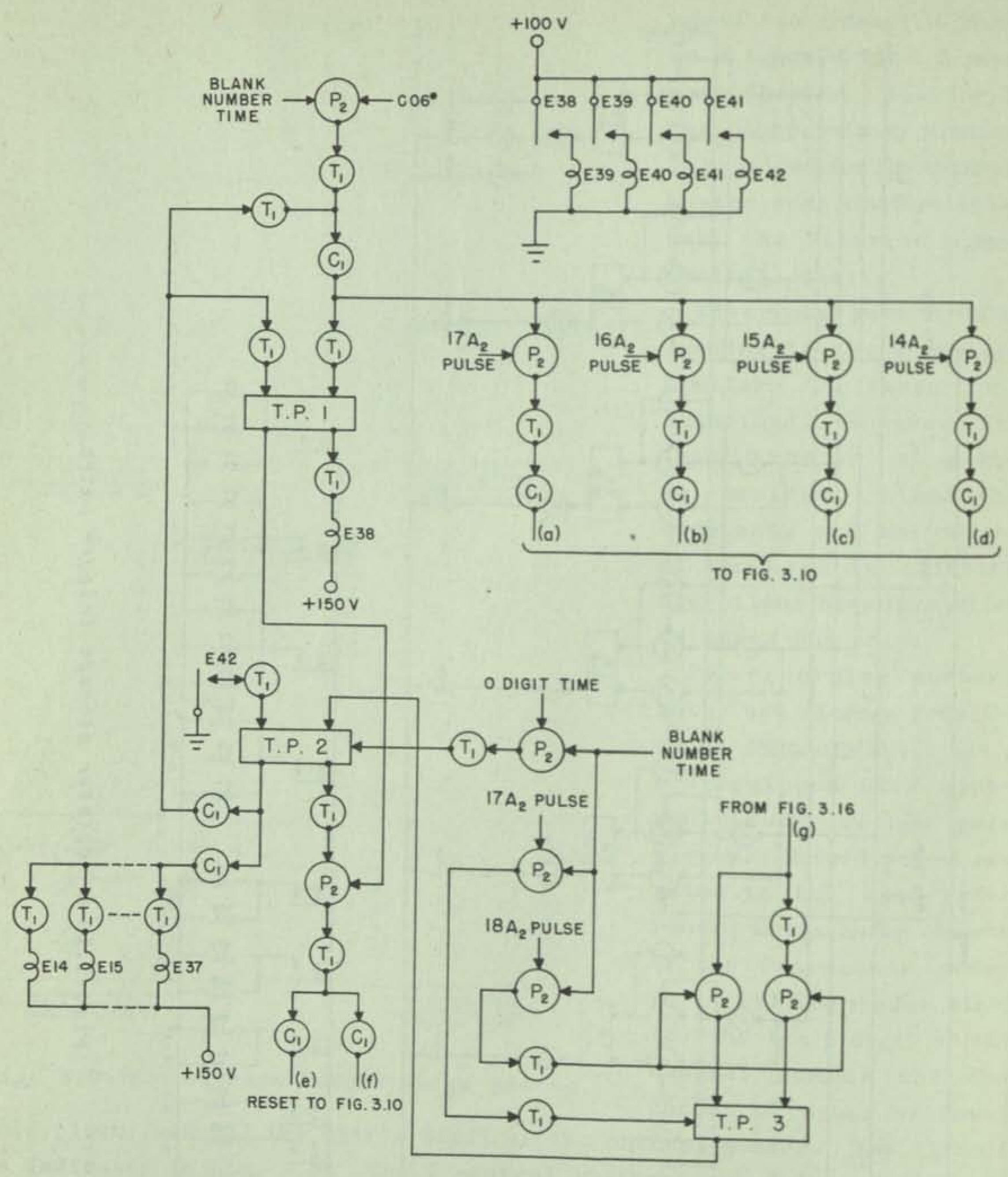


Fig. 3.11—*Slow storage cycling.*

one number time in the direction of drum rotation from the corresponding playback pole piece, Fig. 1.5.

The slow storage drums are equipped with semicircular brackets, Plate III. The lands on the brackets are staggered, permitting twenty-four pole pieces to be mounted on each bracket to cover twenty-four binary channels. The slow storage channels, k = 000,...,399, are selected for record or playback operations under control of the δ register. On a command to read to the δ register, code C06*, the sign and the sixteenth, fifteenth, and fourteenth digits of the quantity on the C bus will be stored in the δ register trigger pairs, Fig. 3.10. This is

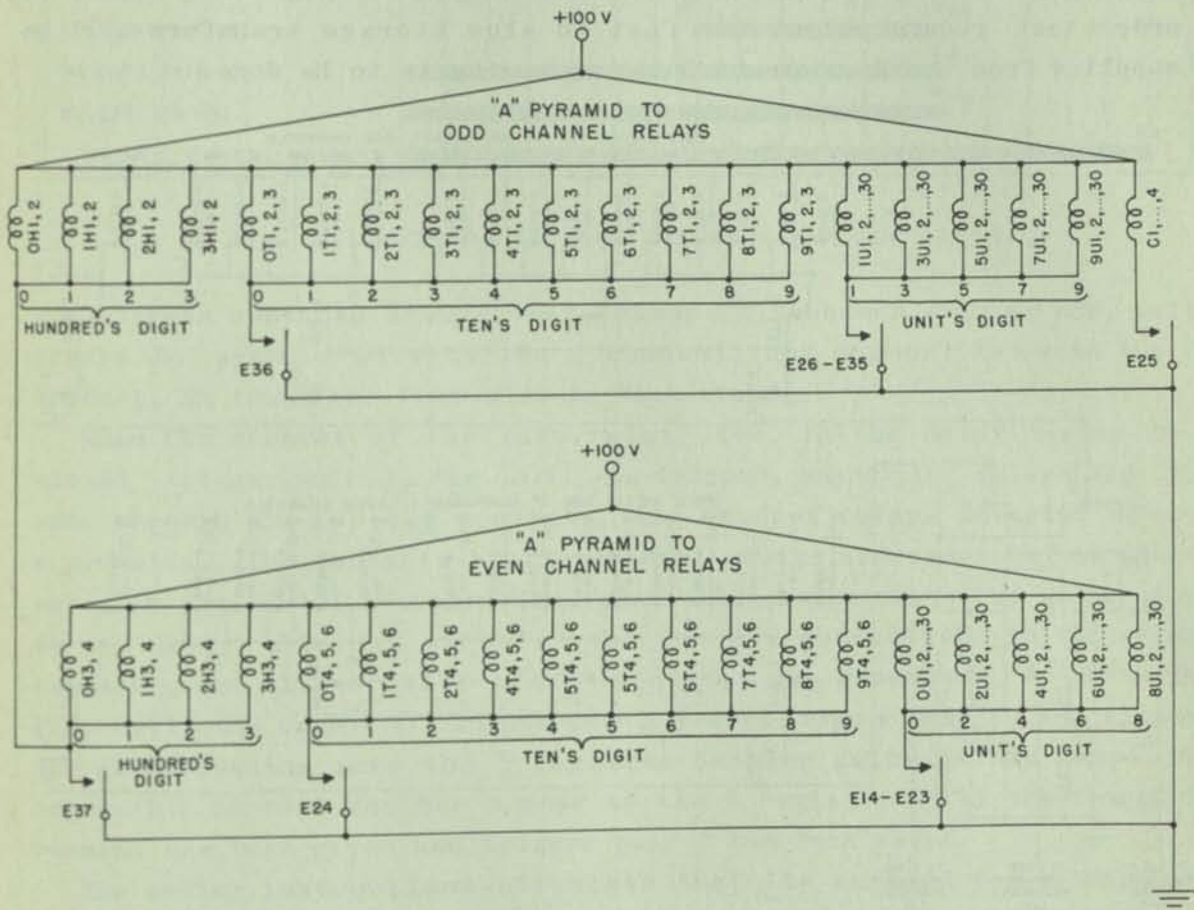


Fig. 3.12—*Channel relays selected by A pyramid.*

accomplished by gating A_2 pulses at 17, 16, 15 and 14 digit times of blank number time to the pentodes serving as puller tubes for the trigger pairs. The other inputs to the pentodes are the binary components of the C bus. For each binary 1 stored in these trigger pairs, one or more of the telephone-type relays E1-E13 is energized. Through the E relay contacts, a corresponding configuration of six-contact relays, A1,..., A31,B1,B2, is energized. At the time that a number is stored in the δ register, trigger pair 1 in Fig. 3.11 is tripped to pick up relay E38. Then a series of relays, E39-E42, is energized one after the other, providing a time delay of about 20 milliseconds while the contacts on the *A* and *B* relays are positioned.

The pyramid of *A* relay contacts makes connections to the selection relays for two slow storage channels, k and $k+1$, where k is the number stored in the δ register trigger pairs, Fig. 3.12. A connection is also made to a record pulse channel selection relay. This is necessary in order that record pulses for fast to slow storage transfers will be supplied from the drum on which the recording is to be done.

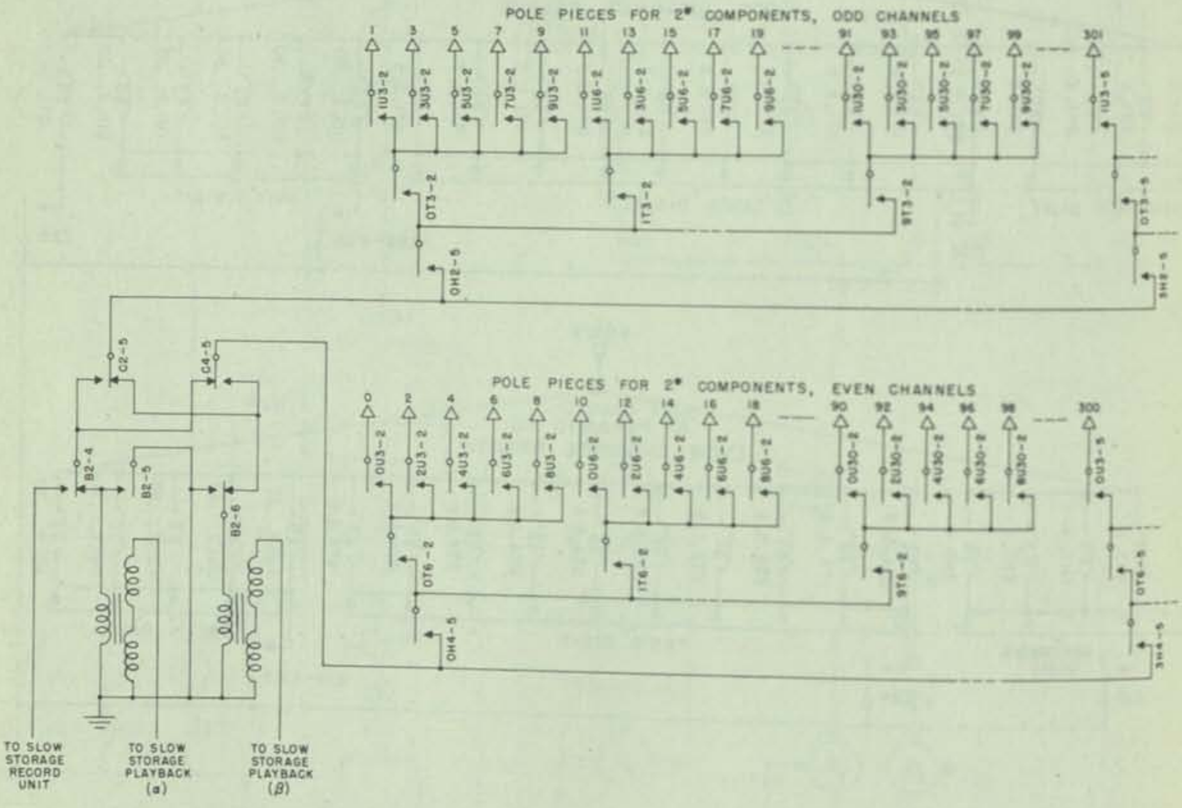


Fig. 3.13—*Slow storage channel selection.*

The *B* relays are energized when a negative sign, indicating a fast to slow storage transfer, is stored in the δ register. Through the *B* relay contacts, channel k will be connected to the slow storage record unit, and the inputs to the slow storage playback units will be grounded, Fig. 3.13. When a positive sign is stored in the δ register, indicating a slow to fast storage transfer, the *B* relays are not energized. In this case, channels k and $k+1$ will be connected to the slow storage playbacks leading to the α and β record units, respectively. Also shown in Fig. 3.13 are contacts on the *C* relays, energized through the *A* pyramid when k is

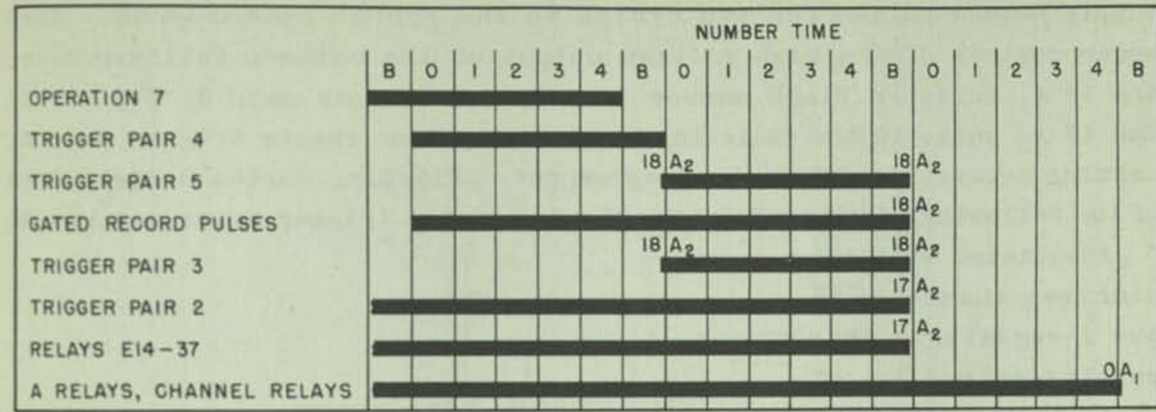


Fig. 3.14—*Timing diagram for transfer operation.*

odd. These contacts insure that channel k, whether even or odd, will always be associated with the α channel, and channel $k+1$ with the β channel, in transfers from slow to fast storage.

When the contact of the last relay, *E*42, in the delay series has closed, trigger pair 2, Fig. 3.11, is tripped, energizing relays *E*14-*E*37 and, through the *E* relay contacts, the channel relays selected by the *A* pyramids. The contacts of the channel relays are connected as indicated in Fig. 3.13. To minimize crosstalk between channels, each relay is shielded, shielded wire is used for all connections to the relay contacts, and all normally closed contacts are grounded. As shown in Fig. 3.11, the output of trigger pair 2 resets trigger pair 1, and closes the gates leading into the δ register trigger pairs. This makes it impossible to read another number to the δ register until the transfer command has been given and trigger pair 2 has been reset.

The coding instructions stipulate that the command for a transfer between fast and slow storage be given at least twelve cycles after a number has been delivered to the δ register, to be sure the relays have been positioned. The command, operation-code 7, requires first, that record pulses be gated for two cycles to either the α and β record units or to the slow storage record units, and second, that the δ register be reset. The timing for these operations is indicated in Fig. 3.14. At the start of 0 number time, trigger pair 4, Fig. 3.15, is tripped causing a high voltage output from the cathode follower gate in the center of the drawing. The 18 A_2 pulse in the following blank number time trips trigger pair 5, to deliver a high voltage output from the cathode follower gate for another cycle. Depending on whether a plus or minus sign is stored in the δ register, one of two pulse gates will be opened, to supply record pulses for two cycles to the proper record units. Also under control of the high voltage output of the cathode follower gate, the 18 A_2 pulse in blank number time trips trigger pair 3, Fig. 3.11. The 17 A_2 pulse in the following blank number time resets trigger pair 2, causing relays *E*14-*E*37 to be dropped out. Finally, during 0 digit time of the following blank number time the δ register trigger pairs are reset.

Associated with the storage channels is the *i* register. This permits selection of fast storage registers under control of previously computed numbers as well as under code control. On a command to deliver a number to the *i* register, code *C*15*, the sign, the 1 component of the fourteenth digit, and the complete thirteenth digit of the number on the *C* bus are stored in the *i* register trigger pairs, Fig. 3.16. This is accomplished by gating A_2 pulses at 17, 14, and 13 digit times to the pentode gates serving as trigger pair puller tubes.

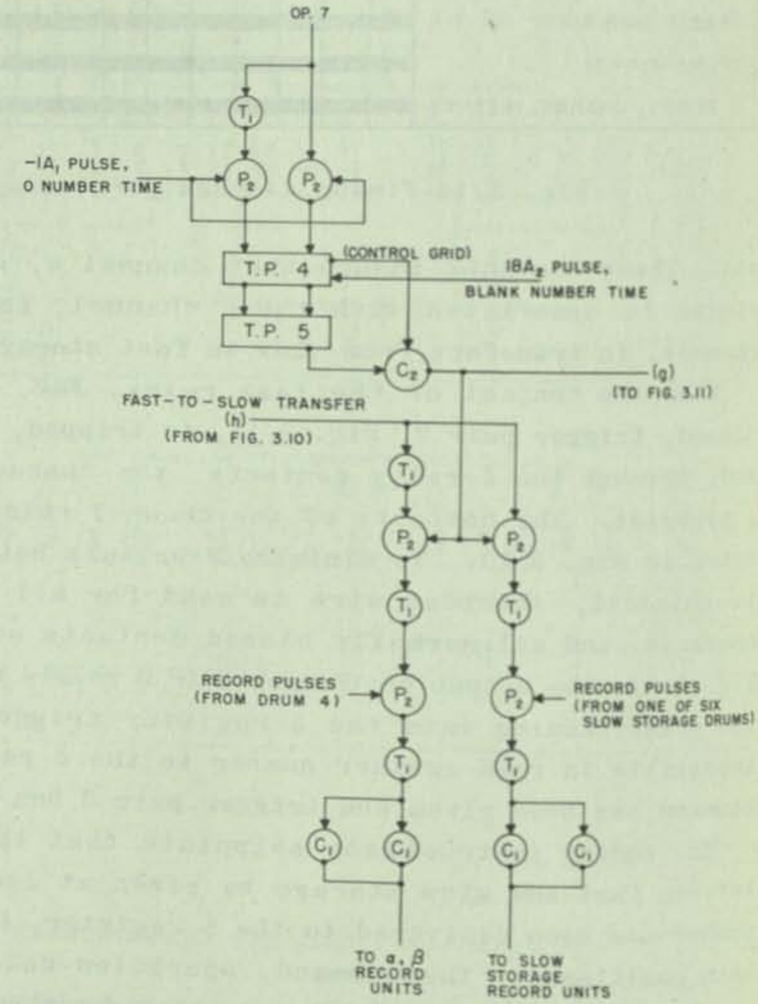


Fig. 3.15—*Record pulse gating for transfers.*

When the *A*, *B*, or *C* number time in a coded instruction is designated as *i*, code 2*, a number time will be selected corresponding to the thirteenth digit stored in the *i* register. To do this, two signals are delivered to the sequence control circuits. One, indicating whether the thirteenth digit is in the range 0-4 or 5-9, is obtained from the trigger pair storing the 2* component of the thirteenth digit. The second, specifying one of five number times, 0, 1, 2, 3, or 4, is obtained

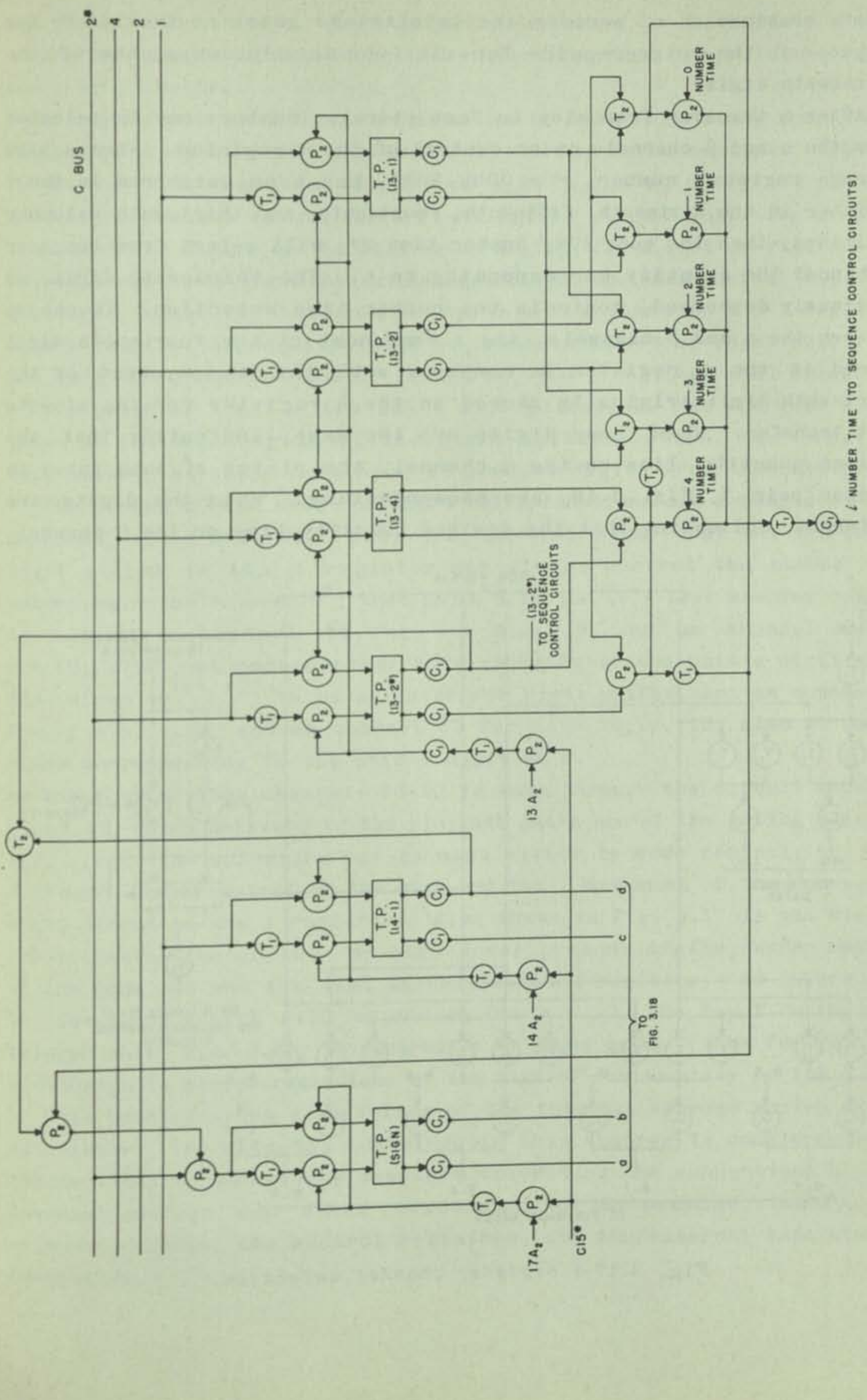


Fig. 3.16—ι register trigger pairs.

from a combination of pentode and twin triode gates controlled by the outputs of the trigger pairs for all four binary components of the thirteenth digit.

After a transfer from slow to fast storage, numbers may be selected from the α and β channels under control of the *ι* register. When a slow storage register number, *r* = 0000-3999, has been delivered to the *ι* register in the sixteenth, fifteenth, fourteenth, and thirteenth columnar positions, then the code *A*08, number time 2*, will select from the α or β channel the quantity corresponding to *r*. The thirteenth digit, as previously described, controls the number time selection. To choose between the α and β channels, the 1 component of the fourteenth digit stored in the *ι* register is compared with the 1 component of the fourteenth digit originally stored in the δ register for the slow to fast transfer. When these digits are the same, indicating that the desired quantity lies on the α channel, the plates of both tubes in trigger pair 1, Fig. 3.16, are nonconducting. When the digits are different, indicating that the desired quantity lies on the β channel,

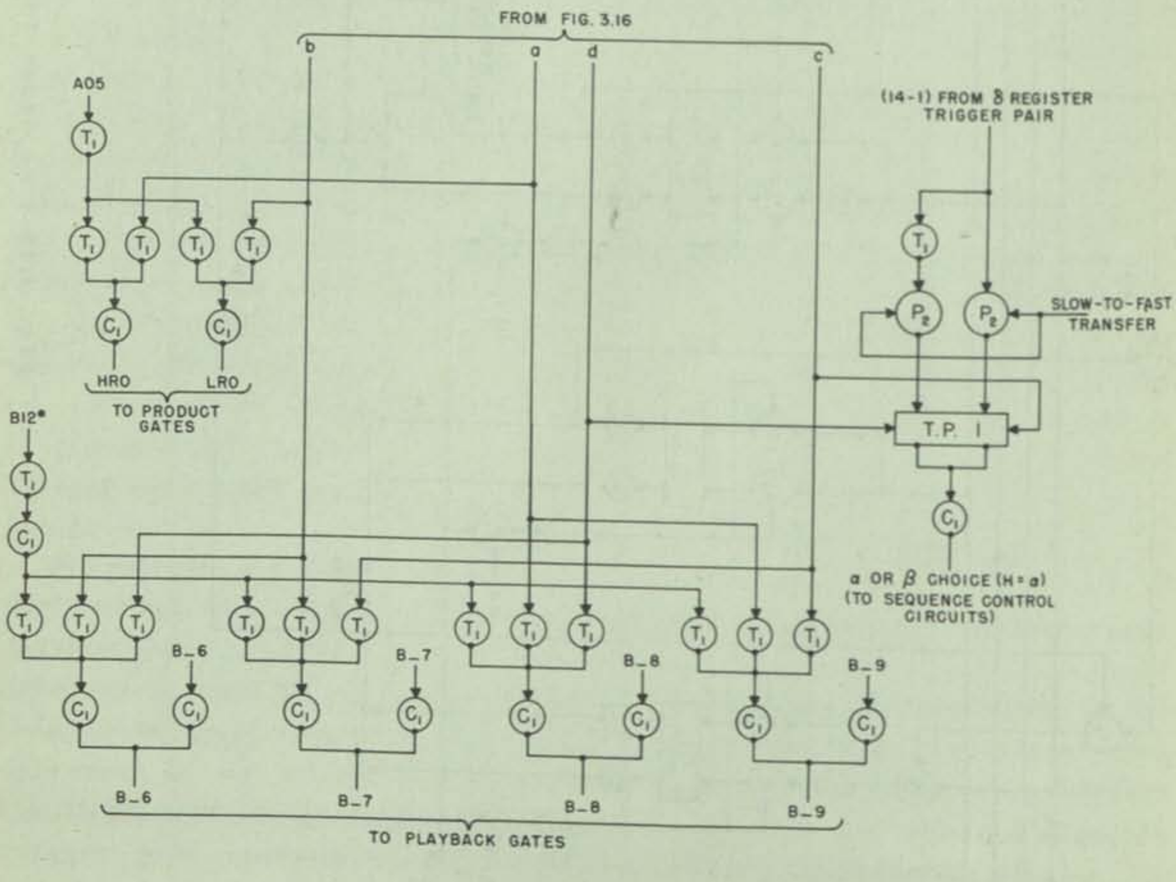


Fig. 3.17—*ι* *register channel selection.*

one plate will conduct. Corresponding to these two situations, a high or low voltage is delivered to the sequence control circuits, to select the α or β channel.

A third function of the i register is to control shifting operations. The method used to shift a quantity s columns is to multiply it by a 1 in the $(s + 1)$st machine column for a left shift, or by a 1 in the $(17 - s)$th machine column for a right shift. In the resulting product, the low-order digits or the high-order digits, for a left or right shift, respectively, constitute the shifted quantity. The amount of shift, s, is delivered in the thirteenth and fourteenth columns to the i register, with a positive or negative sign to indicate a left or right shift. If the code B12*, number time 2*, is given, the i register will select the proper power of ten for the shift multiplication. If the code A05 is given on the third cycle of the shift multiplication, the i register will select the proper read-out of product digits from the multiplier.

The powers of ten used for shift multiplications are arranged on four channels, codes B16, B17, B18, B19, in such a way that the thirteenth digit stored in the i register may always control the number time selection. The values 10^s, that is, a 1 in the $(s + 1)$st machine column, are stored on channel 16 for $s = 0,\ldots,9$, and on channel 17 for $s = 10,\ldots,15$, at number times corresponding to the unit's digit of s. The values $10^{16-|s|}$, to be selected for right shifts, are on channel 18 for $-s = 0,\ldots,9$, and on channel 19 for $-s = 10,\ldots,15$, also at number times corresponding to the unit's digit of s.

The choice among channels 16-19 is made through the circuit shown in Fig. 3.17, which delivers to the playback gates one of the gating voltages $b_6,\ldots,b_9$. The selection may be made either by code control, or, on a B12* code, under control of the sign and the 1 component of the fourteenth digit stored in the i register. Also shown in Fig. 3.17 is the circuit for the selection of low- or high-order product digits, under control of the code A05 and the sign stored in the i register. To insure that the low-order digits will be chosen for $s = -0$, the input to the sign trigger pair, Fig. 3.16, is controlled by three gates. Thus for $|s| = 0$, a plus sign is stored regardless of the sign of the quantity on the C bus.

This concludes the description of the internal storage system of the calculator. The material contained in this chapter is complemented by the sections in succeeding chapters describing the connections between internal storage and other components of the machine, namely, the arithmetic units, the control registers, and the external tape storage facilities.

CHAPTER IV

ARITHMETIC UNITS

The arithmetic units provided in the calculator are an adder and a multiplier. Descriptions of these units and associated circuits will be given in this chapter, preceded by a discussion of the *A*, *B*, and *C* transfer channels which serve as intermediate storage between the arithmetic units and the fast storage system.

Figure 4.1 shows the circuits controlling the *A* transfer channel. The heavy lines indicate the four parallel wires for transmission of

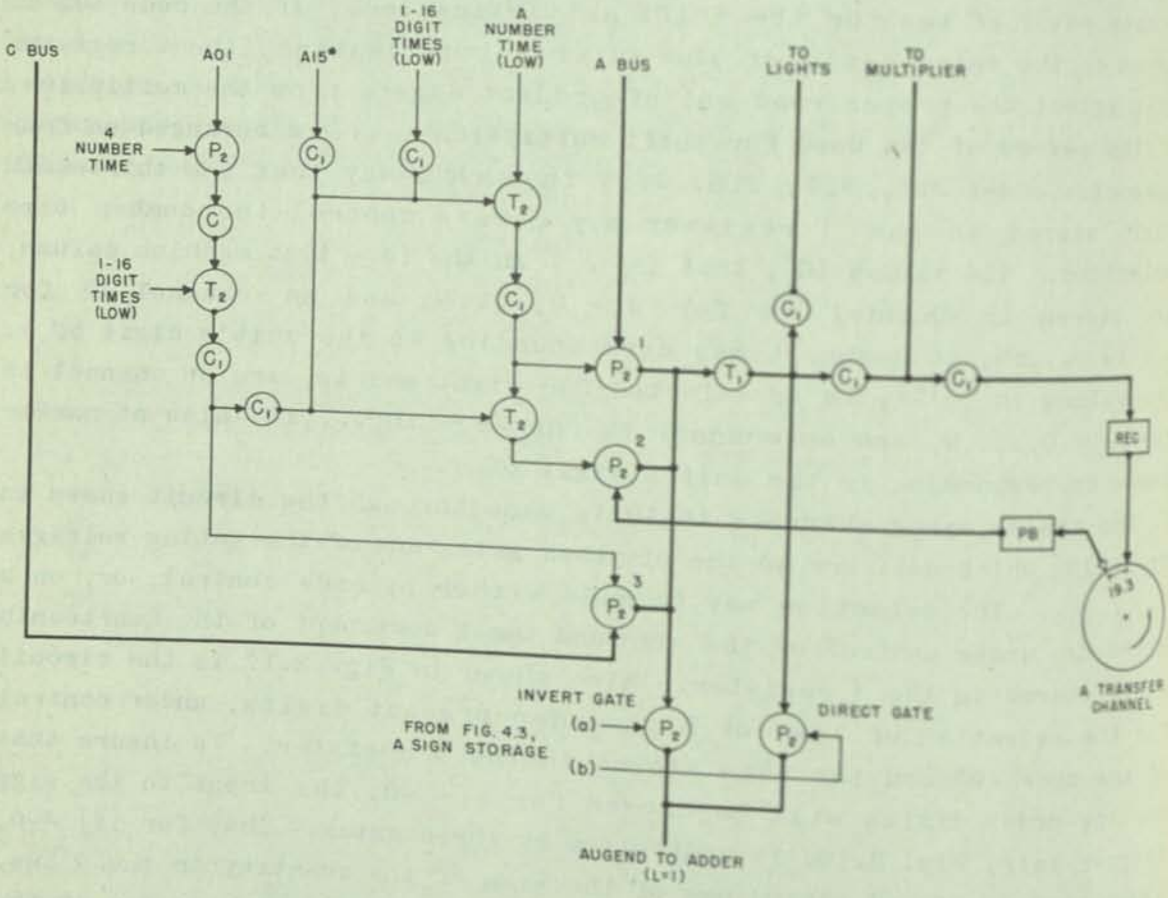


Fig. 4.1—*A transfer channel*.

decimal digits. The number recorded on the channel may come from one of three pentode gates, 1, 2, or 3, of which not more than one is open at a time. Gate 1, the *A* number-time gate, is opened during 1-16 digit times of the *A* number time, to deliver the digits of the number selected from storage by the A codes. Gate 2 is the regeneration gate, generally open during 1-16 digit times of all number times except the A number time, to provide a circuit from the transfer channel playback to the record unit. Gate 3 delivers C bus information to the A transfer channel at 4 number time under control of code $A01$, the σ command. Since record pulses occur approximately one-third of a digit time later than the A_1 pulses used in playback units, the record and playback pole pieces on the transfer channel are 19.3 digit times apart. This causes a digit recorded in 6 digit time, for example, to be played back approximately one number time later by the 6 A_1 pulse, and then recorded again by the 6 record pulse.

When the command to record zeros on the A transfer channel is given, code $A15$*, a high voltage is delivered to two twin triode gates, as indicated in Fig. 4.1. The outputs of the twin triodes close gates 1 and 2, and as gate 3 is normally closed, the voltage on the bus connecting the outputs of gates 1, 2, and 3 will be high. As a result, the input voltage to the record unit will be low and zeros will be recorded.

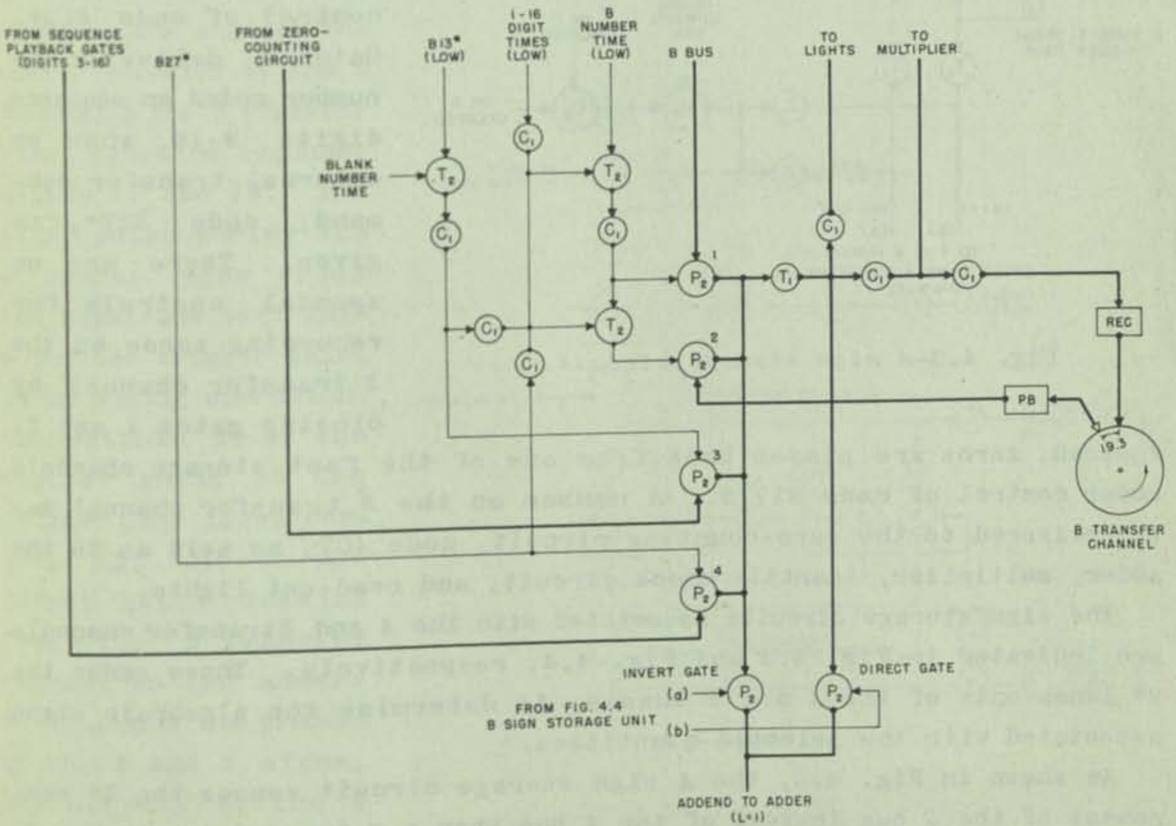


Fig. 4.2—*B transfer channel.*

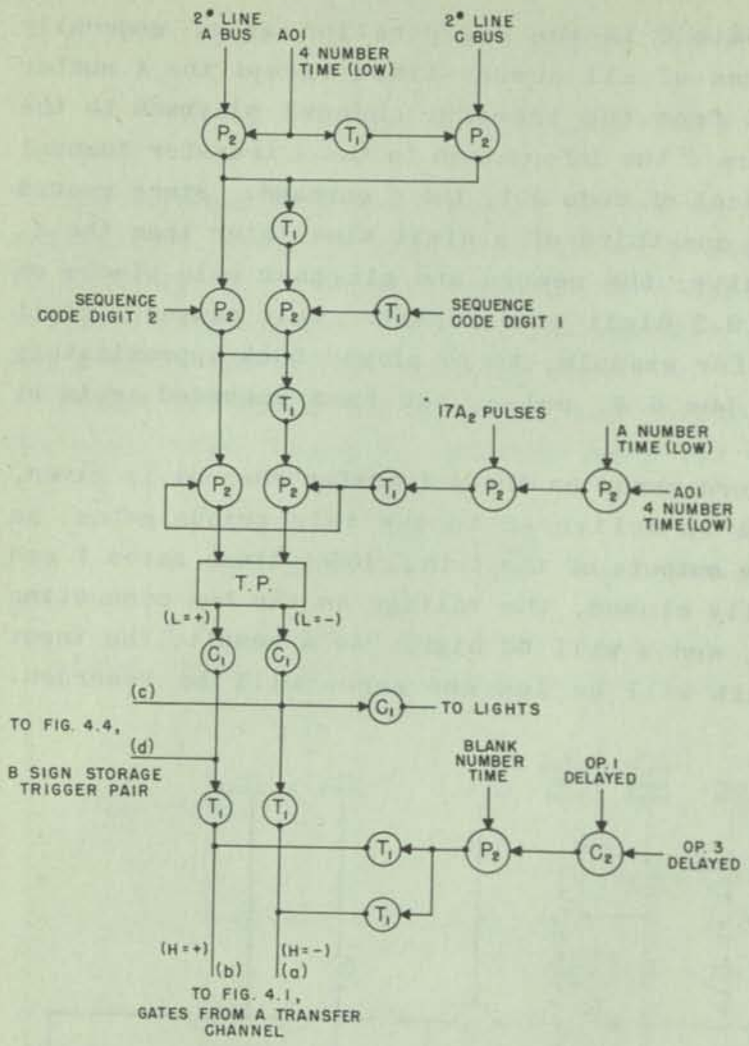


Fig. 4.3—*A sign storage circuit.*

The *B* transfer channel controls are shown in Fig. 4.2. Here the input may come from one of four pentode gates, 1, 2, 3, or 4, of which not more than one is open at a time. Gate 1, the *B* number time gate, and gate 2, the regeneration gate, operate in the same way as gates 1 and 2 on the *A* channel. Gate 3, for normalizing control numbers coming from the zero-counting circuit, is opened at 0 number time under control of code *B*13*. Gate 4 delivers the number coded on sequence digits 3-16 when an external transfer command, code *B*27*, is given. There are no special controls for recording zeros on the *B* transfer channel by closing gates 1 and 2. Instead, zeros are played back from one of the fast storage channels under control of code *B*17 8. A number on the *B* transfer channel may be delivered to the zero-counting circuit, code *A*07, as well as to the adder, multiplier, identity check circuit, and read-out lights.

The sign storage circuits associated with the *A* and *B* transfer channels are indicated in Fig. 4.3 and Fig. 4.4, respectively. These sense the 2* lines only of the *A* and *B* busses, to determine the algebraic signs associated with the selected quantities.

As shown in Fig. 4.3, the *A* sign storage circuit senses the 2* component of the *C* bus instead of the *A* bus when a σ command is given. In either case, the digit and its inverse are delivered to a sign-computation

TABLE 4.1		
Coding of *A* Transfer Sign		
A Transfer Sign	Sequence Digit 2	1
positive	0	0
positive absolute value	0	1
negative absolute value	1	0
negative	1	1

circuit, which multiplies the stored sign by the *A* transfer sign, represented by sequence digits 1 and 2 as shown in Table 4.1. There is a high or low voltage on the common plate connection from the two pentodes in the upper portion of Fig. 4.3 when the resultant sign is plus or minus, respectively. This sign is read into a trigger pair by the 17 A_2 pulse, either during the *A* number time or, on a σ command, during 4 number time. The trigger pair output may be used to control the direct and invert gates leading from the *A* transfer channel to the adder, or, in combination with the *B* sign, to determine the sign of a product.

The *B* sign storage circuit, in Fig. 4.4, combines the sign of the number selected by the *B* codes with the *B* transfer sign, coded on sequence digits 17 and 18. The 17 A_2 pulse during the *B* number time reads the resultant sign into a pentode trigger pair. In an adding operation, the outputs from the control grids of the trigger pair tubes control the direct and invert gates leading from the *B* transfer channel to the adder. To determine the product of the *A* and *B* signs, the outputs of the *A* sign storage trigger pair are connected to

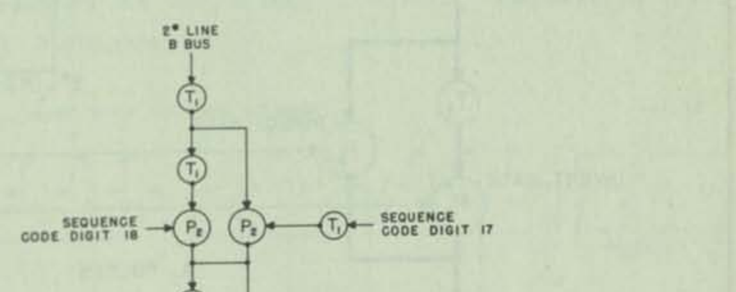


Fig. 4.4—*B sign storage circuit.*

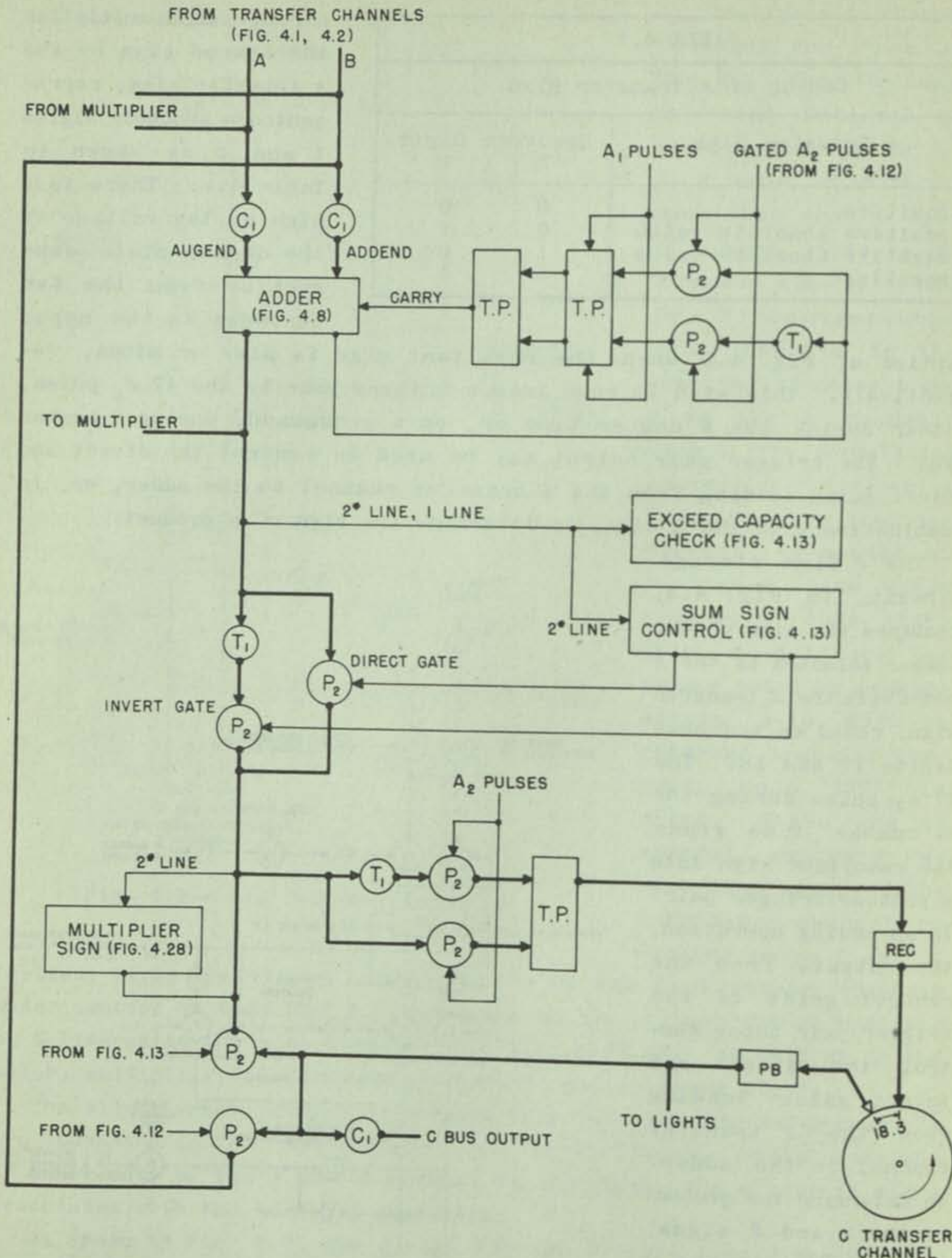


Fig. 4.5—*C transfer channel.*

the suppressor grids of the *B* trigger pair tubes. The connections are made in such a way that the common plate voltage from the *B* trigger pair tubes is high or low for a positive or negative product, respectively. In sign-choice operations, the product sign controls the direct and invert gates between the *B* transfer channel and the adder, while in multiplication the product sign is stored in a trigger pair and supplied later with the product on the *C* bus.

The *C* transfer channel is shown in Fig. 4.5, along with various circuits to be described in connection with the adder. The single input to the *C* transfer channel is from the adder, to which products from the multiplier must be delivered to be recorded on the *C* channel. The sum of the two digits which enter the adder at 6 digit time, for example, may not come through the add circuit in time to be recorded by the 6 record pulse, as will be explained later. Therefore, the sum digit is stored in a trigger pair by the 6 A_2 pulse and recorded in the following digit time. To compensate for this delay, the displacement between record and playback pole pieces is 18.3 digit times, instead of 19.3 digit times as on the *A* and *B* channels.

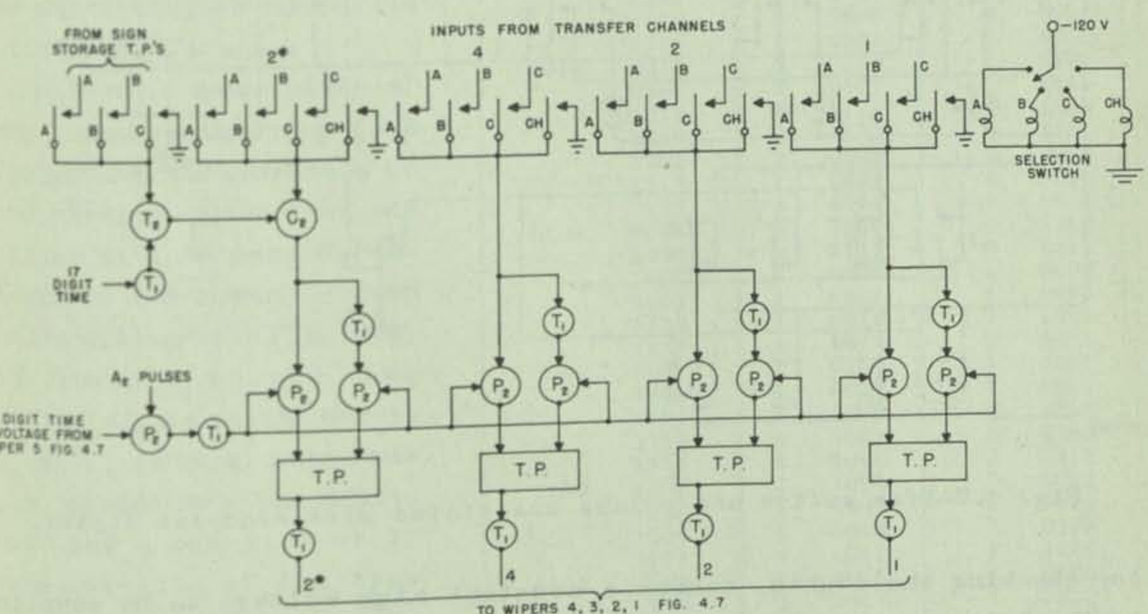


Fig. 4.6—*Control trigger pairs for read-out lights.*

To read the number on any one of the transfer channels to lights, a button, *A*, *B*, or *C*, is depressed on the front panel. This energizes a relay, Fig. 4.6, providing connections from the four output lines of the selected channel to trigger pair puller tubes. A button labeled "*CH*"

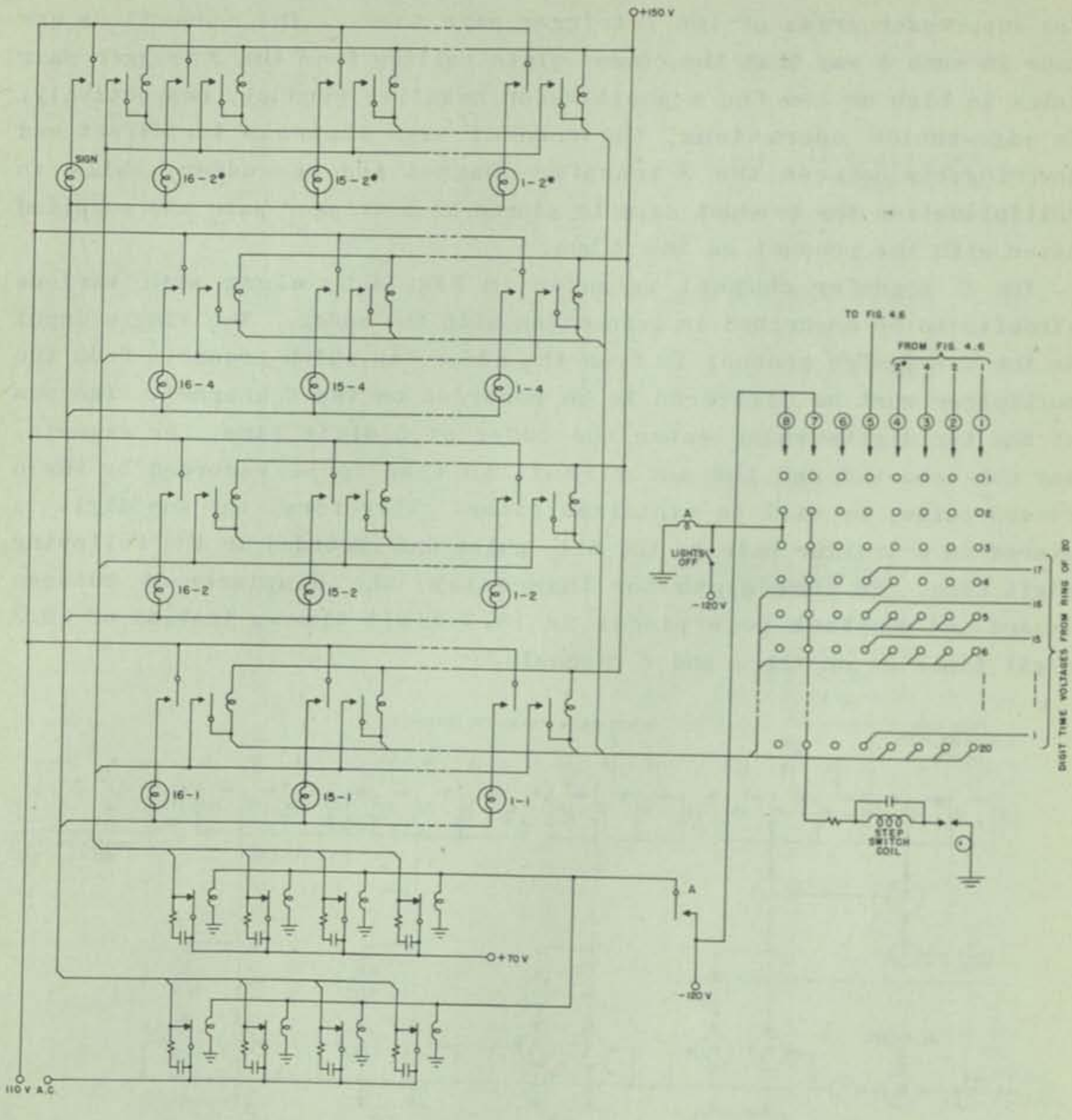


Fig. 4.7—*Step switch and relays associated with read-out lights.*

for checking the lights, causes a constant high voltage to be supplied to the trigger pair puller tubes. When the *A* or *B* channel is selected, a connection is also made to the corresponding sign storage unit, since algebraic signs are not stored on the *A* and *B* transfer channels. When the "read out" button on the front panel is also depressed, a step switch, indicated in Fig. 4.7, is started, causing the number on the selected channel to be stored in a set of relays controlling the lights. Through the fifth step switch wiper, digit-time voltages are supplied to the circuit in Fig. 4.6, gating A_2 pulses to the trigger pair puller tubes. The cam timing on the step switch is slow enough to insure the arrival of at least one gated A_2 pulse on each step of the step switch. Thus successive digits from the transfer channels are stored in the trigger pairs, the outputs of which are connected to four of the step switch wipers. These in turn scan the relays controlling the lights. The lights stay on until the "off" button is depressed or until another number is displayed.

With this description of the transfer channels as a background, the operation of the circuits used for addition will be considered. A static add circuit, shown in block diagram in Fig. 4.8, is provided to add two decimal digits and the carry, 0 or 1, from a previous sum, producing a sum digit, 0-9, and a carry, 0 or 1. Four circuits of the type shown in Fig. 4.9 are connected in parallel, to add the four pairs of components according to the rules for binary addition. Thus in Fig. 4.9, the two 1 components plus the carry from the previous sum are added. The result is a sum digit and a carry, the latter being delivered to the circuit for adding the 2 components.

TABLE 4.2

Addition Table

Inputs to Adder		Binary Sum		Translated Sum	
0000	0000	0	0000	0	0000
	0001	0	0001	0	0001
	0010	0	0010	0	0010
	0011	0	0011	0	0011
	0100	0	0100	0	0100
	1011	0	1011	0	1011
	1100	0	1100	0	1100
	1101	0	1101	0	1101
	1110	0	1110	0	1110
	1111	0	1111	0	1111
0001	0001	0	0010	0	0010
	0010	0	0011	0	0011
	0011	0	0100	0	0100
	0100	0	0101	0	1011
	1011	0	1100	0	1100
	1100	0	1101	0	1101
	1101	0	1110	0	1110
	1110	0	1111	0	1111
	1111	1	0000	1	0000
0010	0010	0	0100	0	0100
	0011	0	0101	0	1011
	0100	0	0110	0	1100
	1011	0	1101	0	1101
	1100	0	1110	0	1110
	1101	0	1111	0	1111
	1110	1	0000	1	0000
	1111	1	0001	1	0001
0011	0011	0	0110	0	1100
	0100	0	0111	0	1101
	1011	0	1110	0	1110
	1100	0	1111	0	1111
	1101	1	0000	1	0000
	1110	1	0001	1	0001
	1111	1	0010	1	0010
0100	0100	0	1000	0	1110
	1011	0	1111	0	1111
	1100	1	0000	1	0000
	1101	1	0001	1	0001
	1110	1	0010	1	0010
	1111	1	0011	1	0011
1011	1011	1	0110	1	0001
	1100	1	0111	1	0010
	1101	1	1000	1	0011
	1110	1	1001	1	0100
	1111	1	1010	1	1011
1100	1100	1	1000	1	0010
	1101	1	1001	1	0011
	1110	1	1010	1	0100
	1111	1	1011	1	1011
1101	1101	1	1010	1	0100
	1110	1	1011	1	1011
	1111	1	1100	1	1100
1110	1110	1	1100	1	1100
	1111	1	1101	1	1101
1111	1111	1	1110	1	1110

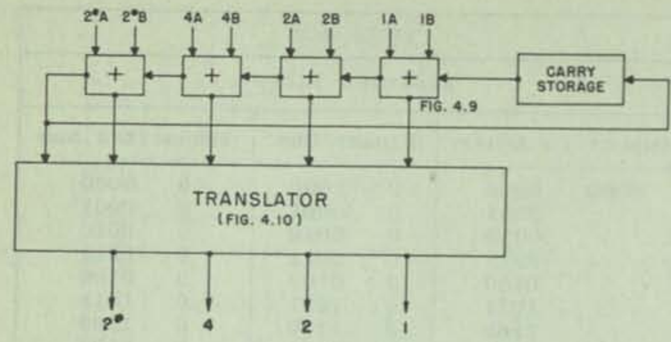


Fig. 4.8—*Block diagram of static add circuit.*

The combined output of the four parallel circuits representing the binary sum, is delivered to a translation circuit, Fig. 4.10, for conversion to 2*,4,2,1 notation. The possible binary sums and their 2*,4,2,1 equivalents are listed in Table 4.2. The static add circuit does not produce an output instantaneously, because the capacity in the voltage dividers delays the carry numbers passing from one binary add circuit to the next and retards the setting of the gates in the translator. This is the reason for using A_2 pulses to read sums from the adder into trigger pair storage, prior to recording on the C transfer channel.

The cycling circuits for addition are under control of the operation-code 1. Since a subtraction is treated as the addition of a negative number, the same code is supplied by the instructional tape preparation table when either an add or subtract operation is selected. In the latter case the B transfer sign is automatically inverted. For an addition coded on line n, the A and B control voltages cause numbers to be selected from storage and recorded on the A and B transfer channels. Since the voltage corresponding to operation-code 1 is also available for one cycle starting at blank number time, cycle n, it must be delayed in order to control the addition of the selected quantities during blank and 0 number times, cycle n+1. The delay circuit, Fig. 4.11, makes the command available for one cycle starting, arbitrarily, at 2 number

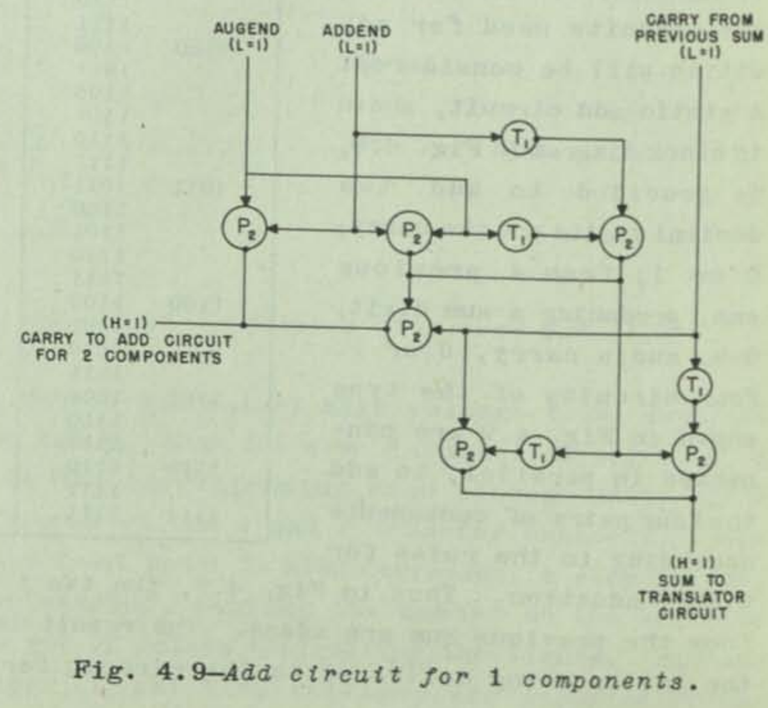


Fig. 4.9—*Add circuit for 1 components.*

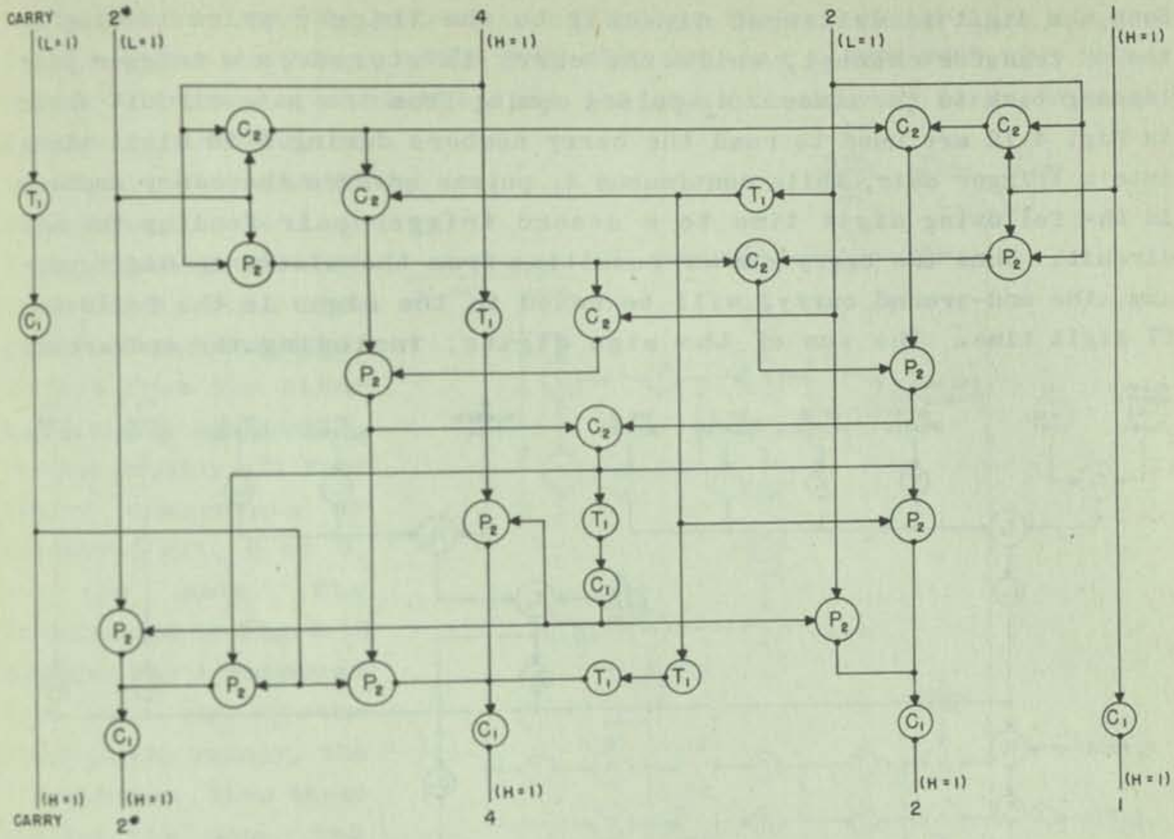


Fig. 4.10—*Translator*.

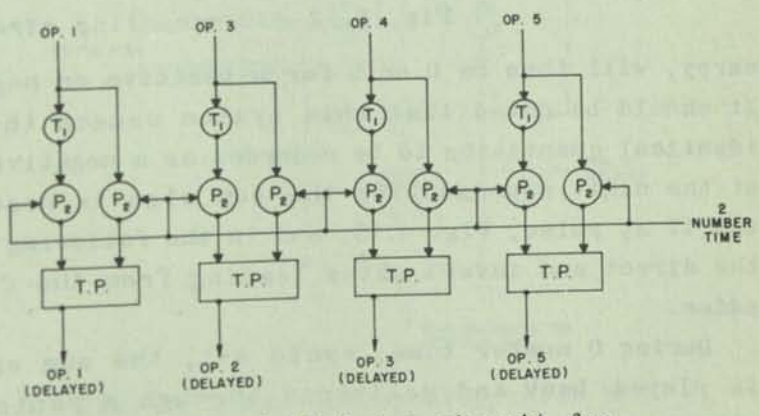


Fig. 4.11—*Delay circuit for operation-codes* 1, 3, 4, 5.

time, cycle n. Similar circuits, included in Fig. 4.11, delay the commands for double-accuracy addition, operation-codes 3 and 4, and sign-choice, operation-code 5, since these commands, when given on cycle n, also require the use of the adder on cycle $n+1$.

During blank number time, cycle $n+1$, the delayed add command permits the A and B sign storage circuits, Figs. 4.3 and 4.4, to open the direct or invert gates leading from the transfer channels to the static add circuit. The numbers selected from storage or their nine's complements are added serially in digit pairs, beginning with the lowest order digits.

Each sum digit is delivered directly to the trigger pairs leading to the *C* transfer channel, while the carry is stored in a trigger pair leading back to the adder. A_2 pulses coming from the gate circuit shown in Fig. 4.12 are used to read the carry numbers during 1-16 digit times into a trigger pair, while continuous A_1 pulses advance the carry numbers in the following digit time to a second trigger pair feeding the add circuit. Thus the carry number resulting from the sixteenth digit pair sum, the end-around carry, will be added to the signs in the following 17 digit time. The sum of the sign digits, including the end-around

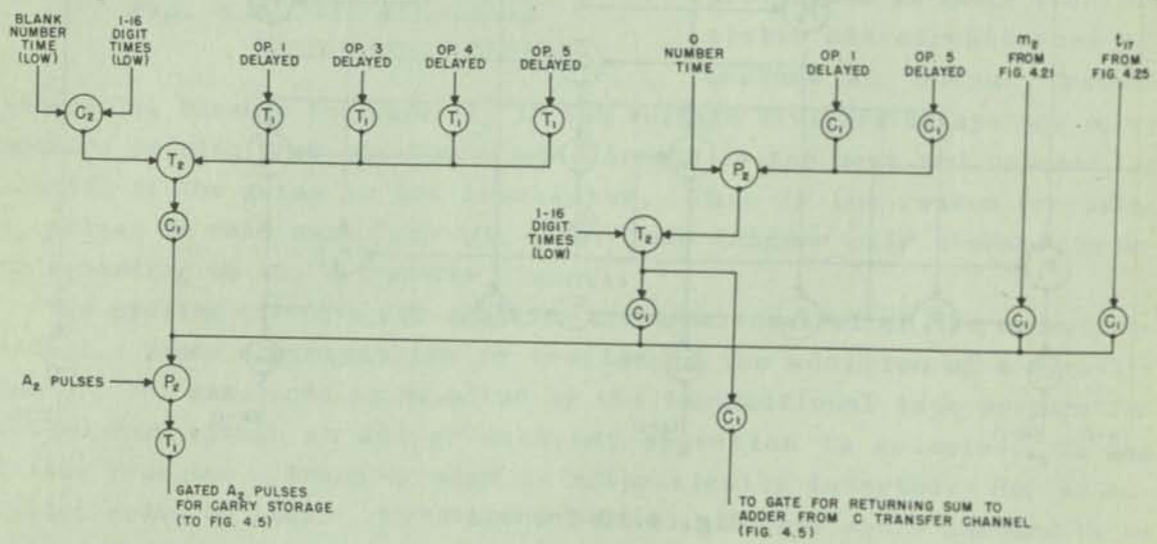


Fig. 4.12—*Adder cycling circuits.*

carry, will then be 0 or 9 for a positive or negative sum, respectively. It should be noted that this system causes the difference between two identical quantities to be recorded as a negative zero. The 2* component of the digit representing the sum sign is read into a trigger pair by the 17 A_2 pulse, Fig. 4.13, and in the following number time will control the direct and invert gates leading from the *C* transfer channel to the adder.

During 0 number time, cycle $n+1$, the sum on the *C* transfer channel is played back and delivered through a pentode gate to the adder as addend, to pick up the end-around carry. Zeros are supplied as the augend, since the gates from the *A* transfer channel are closed. The sum is now sent on either directly or inverted, depending on the sum sign, to the *C* transfer channel. The sum sign is also recorded on the channel as the seventeenth digit. After 0 number time, the *C* regeneration gate is opened, causing the *C* playback output to be regenerated and made continuously available for return to fast storage.

The exceed-capacity alarm circuit stops the calculator if a sum, x, falls outside the range, $-9999\ 9999\ 9999\ 9999 < x \leq 9999\ 9999\ 9999\ 9999$. The circuit makes use of the fact that an excessively large or small sum causes the sum of the sign digits, at 17 digit time, to be 1 or 8, respectively. In both these cases, the 1 component of the sum digit differs from the other three binary components, whereas usually all four binary components of the sum digit, 0 or 9, are the same. The circuit shown in Fig. 4.13 compares the 1 component with only one of the other three, namely, the 2* component. When these are not the same, the pentode gate serving as a trigger pair puller tube will be open, permitting the 17 A_2 pulse to trip the trigger pair and sound an alarm. The circuit is disabled, however, during a double-accuracy addition.

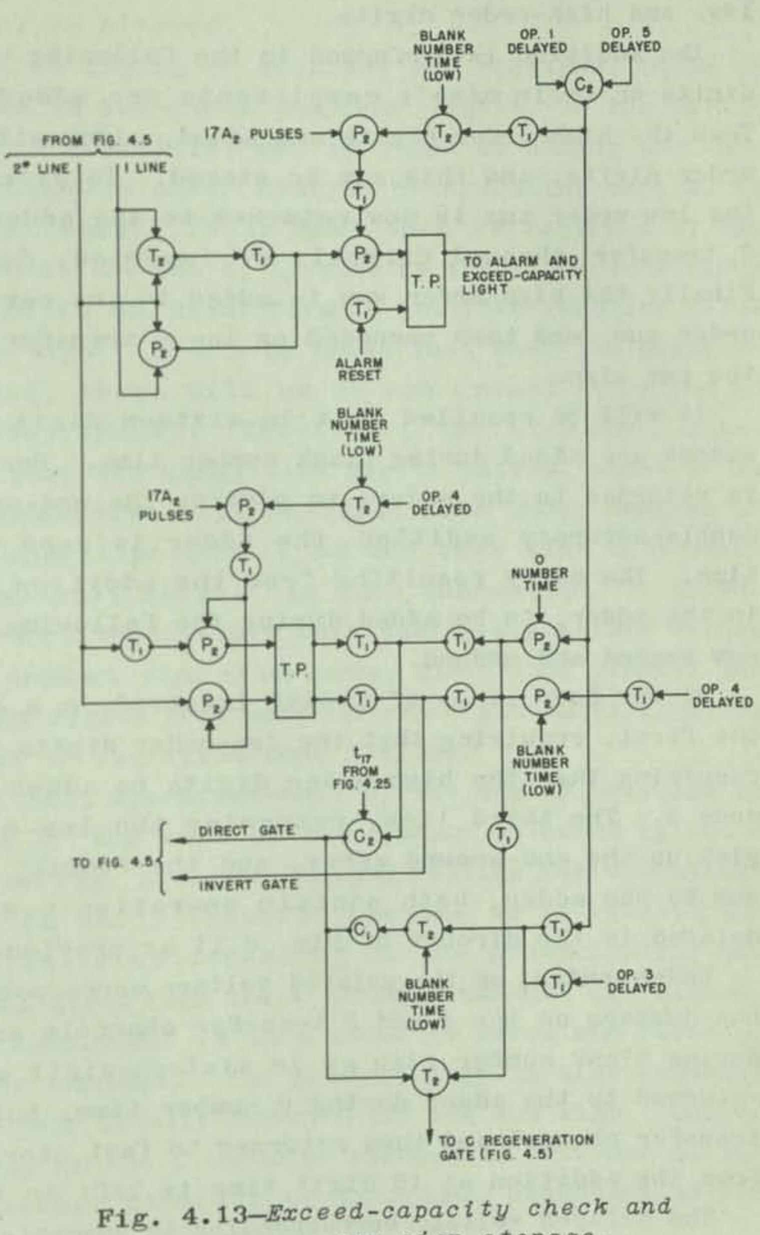


Fig. 4.13—*Exceed-capacity check and sum sign storage.*

To perform a double-accuracy (thirty-two digit) addition using the adder just described requires four cycles of machine time. The four lines of coding for this operation are supplied automatically by the instructional tape preparation table when the key marked ⊕ in the "operations" column is selected, along with keys for the storage registers containing the low-order digits. For example, the order indicated as "$x_0 \oplus y_2 = z_3$" will supply the coding to add the numbers having low-order digits stored in x_0 and y_2, and high-order digits stored in x_5 and y_7, respectively. The low- and high-order sum digits will be recorded in z_3 and z_8. The algebraic sign is stored twice, with both the low- and high-order digits.

The addition is performed in the following way. First, the low-order digits or their nine's complements are added, and the sum is stored. Then the high-order digits are added, along with the carry from the low-order digits, and this sum is stored. To pick up the end-around carry, the low-order sum is now returned to the adder, and is recorded on the C transfer channel directly or inverted, depending on the sum sign. Finally the high-order sum is added to the carry, if any, from the low-order sum, and then recorded on the C transfer channel under control of the sum sign.

It will be recalled that in sixteen-digit addition, the augend and addend are added during blank number time. During 0 number time the sum is returned to the adder, to pick up the end-around carry. However, in double-accuracy addition the adder is used only during blank number time. The carry resulting from the addition at 16 digit time is left in the adder, to be added during the following blank number time to the new augend and addend.

Of the four lines of coding involved in a double-accuracy addition, the first, requiring that the low-order digits be added, and the second, requiring that the high-order digits be added, both contain operation-code 3. The third line, returning the low-order sum to the adder to pick up the end-around carry, and the fourth, returning the high-order sum to the adder, both contain operation-code 4. These commands are delayed in the circuit of Fig. 4.11 as previously mentioned.

Under control of the delayed voltage corresponding to operation-code 3, the numbers on the A and B transfer channels are delivered to the adder during blank number time as in sixteen-digit addition. The sum is not returned to the adder during 0 number time, but is regenerated on the C transfer channel and then returned to fast storage. The carry resulting from the addition at 16 digit time is left in the adder.

The delayed voltage corresponding to operation-code 4 opens the direct gate from the B transfer channel to the adder, Fig. 4.4, to eliminate the usual B sign control when the low- and high-order sum digits are returned to the adder to pick up carry numbers. Zeros are recorded on the A transfer channel and supplied as augend, the code $A15$* having been given by the instructional tape preparation table. Under control of operation-code 4, pulses are gated through the circuit in Fig. 4.12, in order to store carry numbers in the adder. The sum digits are recorded either directly or inverted, depending on the sum sign. The gates leading to the sum sign trigger pair are closed since the sum sign has already been computed and must not be altered.

When more than thirty-two digits of accuracy are required, three or more registers must be used to store each complete quantity in the calculator. An addition of such quantities may be coded by adding the successive groups of sixteen digits under control of operation-code 3, to form sixteen-digit partial sums. These must then be returned to the adder under control of operation-code 4, to pick up the end-around carry and to invert negative sums to the usual form, a positive absolute value followed by the algebraic sign. It may be noted that when two positive quantities are being added, there will be no end-around carry and no conversion from nine's complements. Therefore, the time for a high-accuracy addition of two positive quantities may be halved, since it is sufficient to form the successive sixteen-digit sums under control of operation-code 3, using operation-code 1 on the last line as a check that the sum is not excessively large. In this connection, it is advisable in high-accuracy work to multiply only positive absolute values and to supply the proper product sign afterwards, since this permits the high- and low-order product digits from each sixteen-digit multiplication to be summed under control of operation-code 3 alone.

In a sign-choice operation, operation-code 5, the number selected by the *B* codes is multiplied by the sign of the number selected by the *A* codes. As may be seen from Fig. 4.4, the delayed voltage for operation-code 5 causes the number on the *B* transfer channel to be read to the adder either directly or inverted depending on the product sign, not the *B* sign. As both of the gates from the *A* transfer channel are closed, the number or its nine's complement is then added to zeros and recorded directly on the *C* transfer channel. The product sign is also recorded and stored in the trigger pair usually reserved for the sum sign. During 0 number time, the quantity on the *C* transfer channel is returned to the adder and then recorded either directly or inverted, depending on the product sign.

This concludes the descriptions of circuits immediately associated with the adder and employed in the operations of addition, subtraction, double-accuracy addition, and sign-choice. In considering the multiplication circuits, the main components of the multiplier will be described first, followed by a discussion of the cycling controls and the operation of particular circuits.

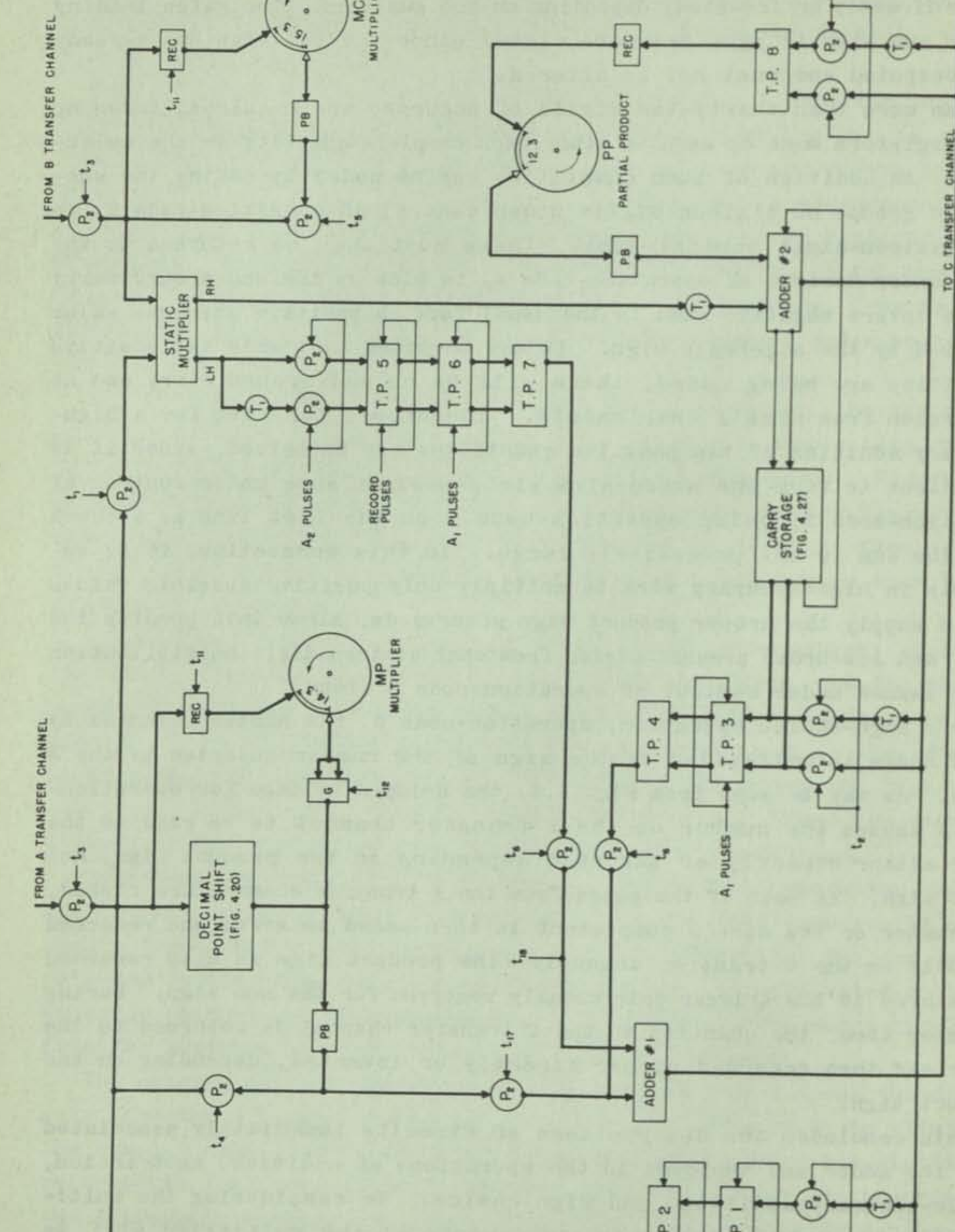


Fig. 4.14—Over-all diagram of multiplication circuits.

Referring to Fig. 4.14, an over-all diagram of the multiplication circuits, it is seen that the multiplicand, having been selected from storage and recorded on the *B* transfer channel, may be gated to the record unit for the decimal channel labeled *MC*. The sixteen multiplicand digits, once recorded, are regenerated on the *MC* channel, since the record and playback pole pieces on this channel are separated by 15.3 digit times. The input to the *MC* record unit also feeds a static multiply circuit, causing the multiplicand digits to be delivered there serially, starting with the lowest order digit, and continuously, as long as either the input gate or the regeneration gate is open.

The decimal channel labeled *MP* in Fig. 4.14 stores the digits of the multiplier, coming from the *A* transfer channel. The lowest order digit of the multiplier is gated to trigger pairs in the static multiply circuit, where it will be used to multiply all sixteen multiplicand digits. Because the record and playback pole pieces on the *MP* channel are 14.3 digit times apart, each digit of the multiplier is played back fifteen digit times after it is recorded and may then be gated back to the record unit. Thus at intervals of sixteen digit times, the second, third, ..., sixteenth multiplier digits will be played back and gated to the trigger pairs of the static multiply circuit, where each will multiply the multiplicand digits.

In a single digit time, the static multiply circuit, outlined in Fig. 4.15, computes the product of one multiplier and one multiplicand digit. By means of vacuum-tube pyramids, Fig. 4.16, the entering digits are translated to decimal notation and delivered to the gates, Fig. 4.17, which serve to multiply the two digits. The output leads from the gates are connected as shown in Figs. 4.18 and 4.19, to deliver the product in

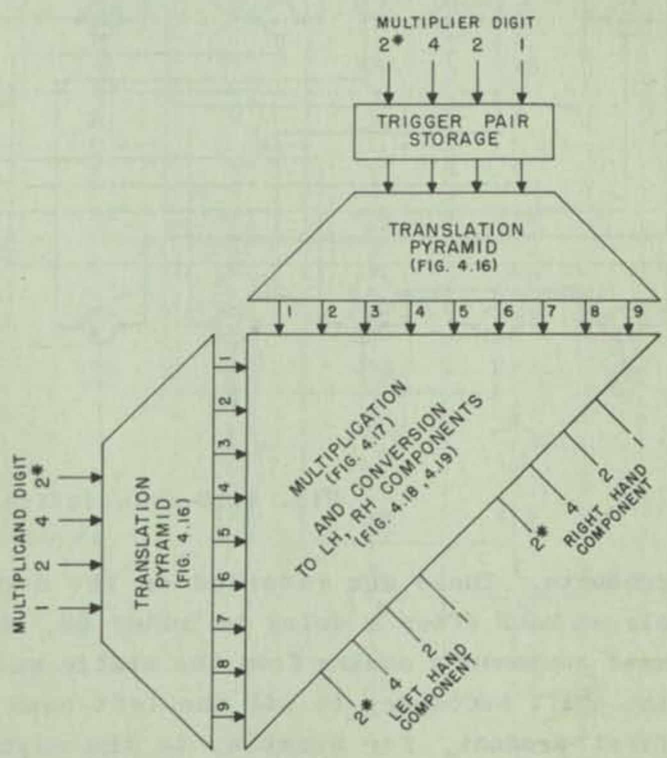


Fig. 4.15—*Block diagram of static multiply circuit.*

2*,4,2,1 notation. In each product the ten's digit is referred to as the left-hand component, and the unit's digit as the right-hand component.

Two static add circuits, #1 and #2, are provided for combining successive left-hand and right-hand components to obtain the thirty-two product digits. Adder #1 is the same circuit used for the operations of addition, subtraction, double-accuracy addition, and sign-choice. Adder #2 is a duplicate circuit, with special controls on the storage of carry numbers as will be described later.

Left-hand components from the static multiply circuit are delivered to adder #1, where they are added to sums from adder #2 to form partial

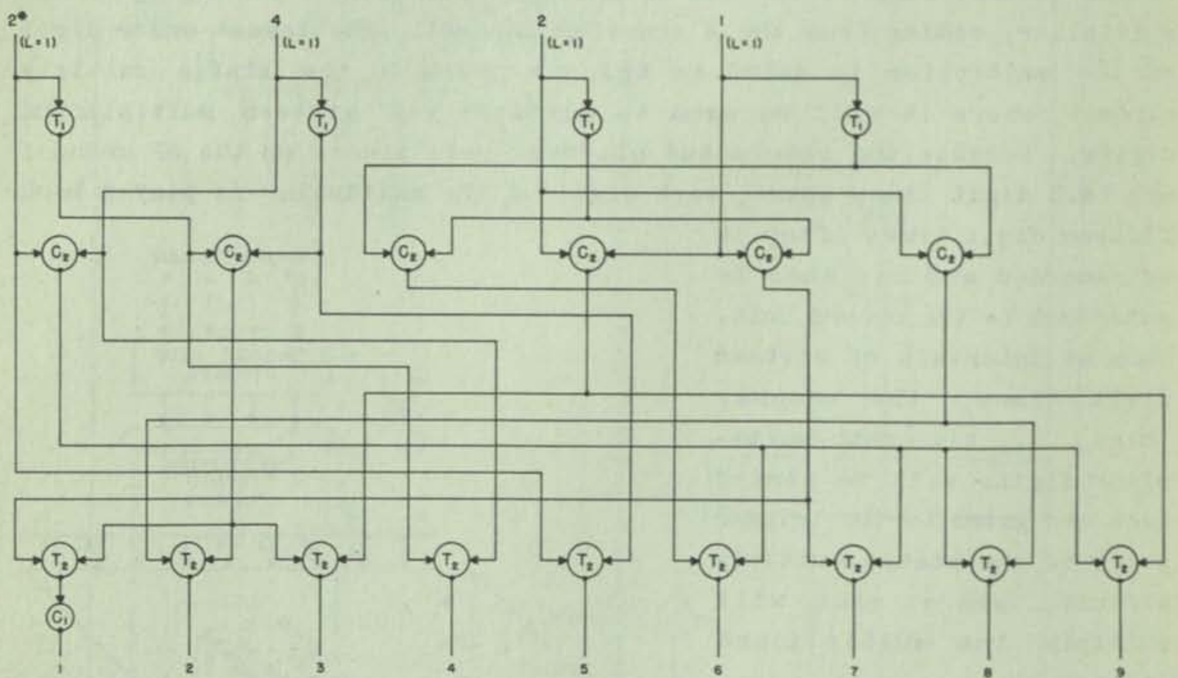


Fig. 4.16—*Translation pyramid.*

products. These are recorded on the decimal channel labeled *PP*, and played back after a delay to adder #2, where they are added to right-hand components coming from the static multiply circuit. This provides the shift necessary to add the left-hand component resulting from the first product, for example, to the right-hand component resulting, sixteen digit times later, from the product of the second multiplier digit and the first multiplicand digit.

The decimal point shift circuit stores the thirty-two product digits resulting from one multiplication. The final low-order product digits are recorded there in the course of a multiplication, and on the *MP*

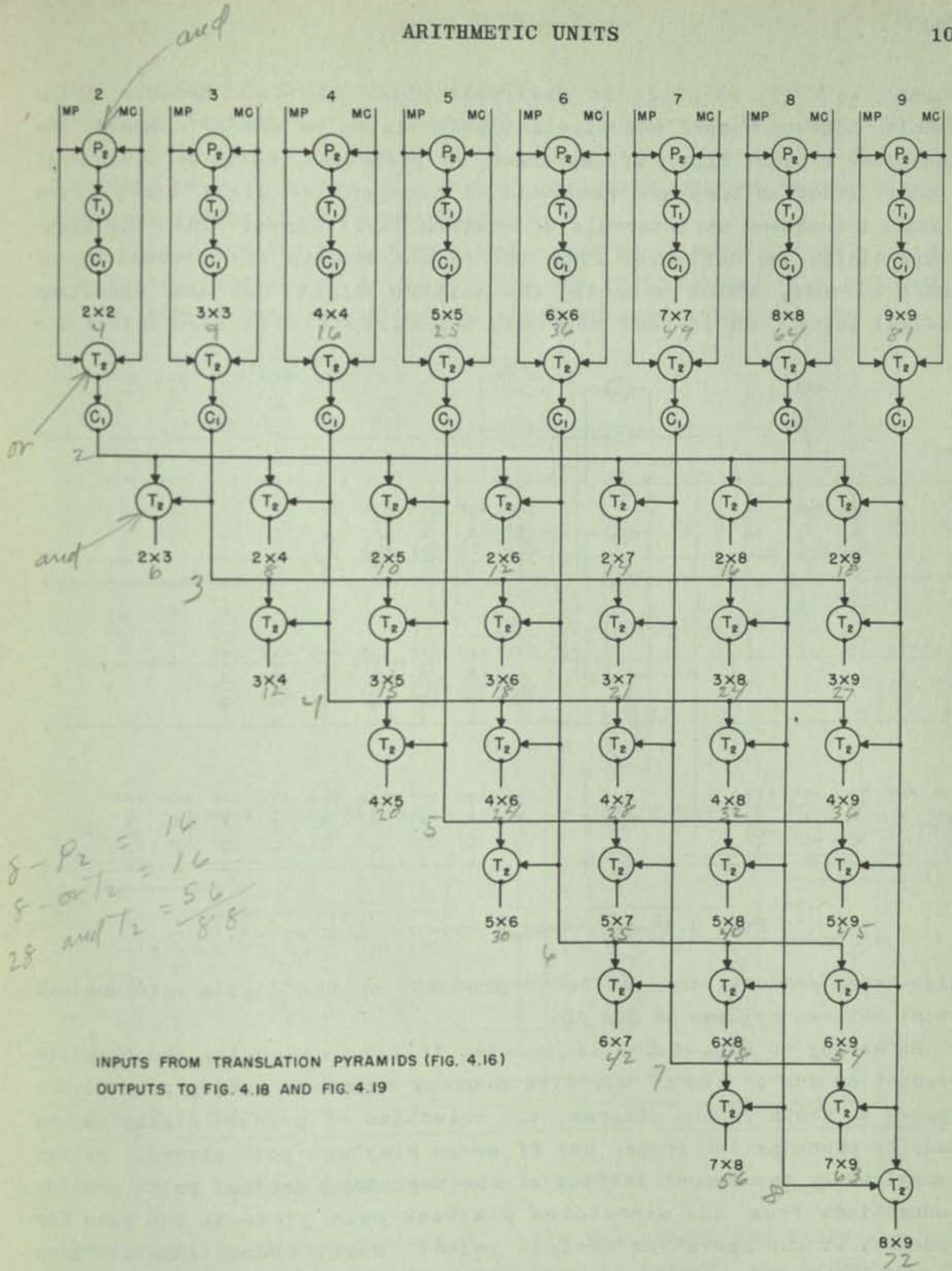


Fig. 4.17—*Multiplication circuit.*

channel as well, in place of used multiplier digits. Because of the spacing between record and playback pole pieces on the *MP* channel, the low-order product digits are advanced, or precessed, until at the end of a multiplication they are recorded at consecutive digit times, even though introduced at intervals of sixteen digit times. Then the high-order digits are delivered from the *PP* channel to the decimal point shift circuit, which selects the sixteen digits at the operating decimal point. Under code control, the circuit will also select the

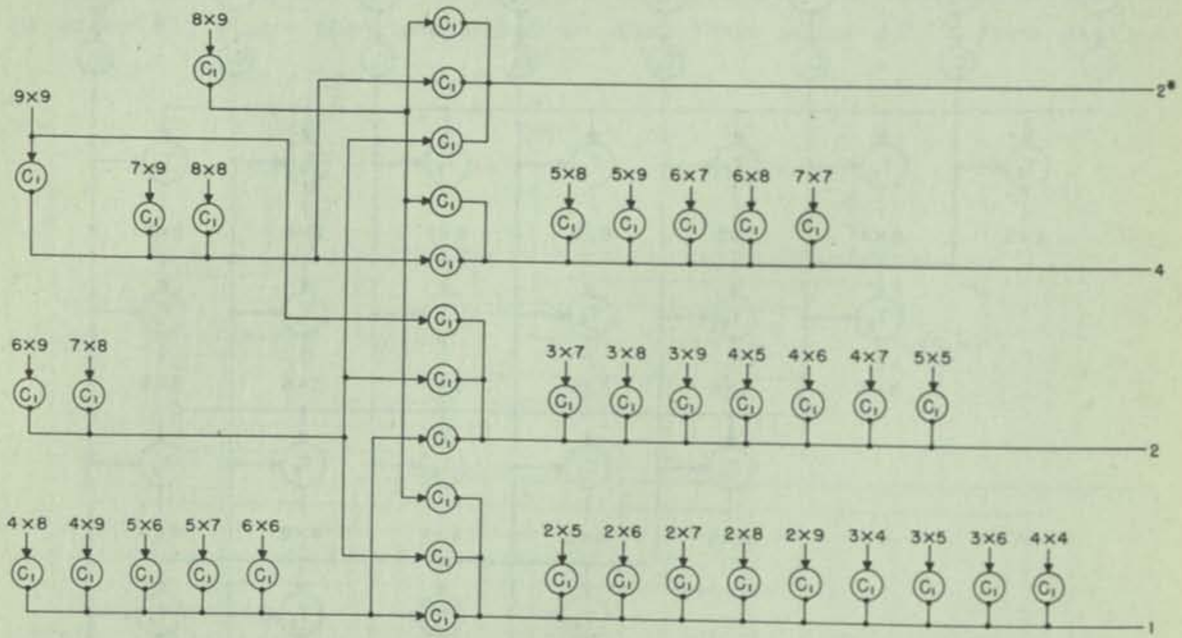


Fig. 4.18—*Left-hand component of product.*

high-order product, the low-order product, or the digits with decimal point between columns 16 and 15.

Referring to Fig. 4.20, it is seen that by recording the complete product on one or two of the five decimal channels having pole pieces spaced as shown in the diagram, the selection of product digits may be made by choosing the proper one of seven playback pole pieces. Relays energized by the manual setting of the operating decimal point provide connections from the associated playback pole piece to the gate for products at the operating decimal point. Relay connections are also made to one of the record pole pieces on the four channels at the right when the operating decimal point is between columns 15 and 14, 13 and 12, 12 and 11, or 10 and 9. However, the product is recorded on the channel at the far left independent of the choice of operating decimal point, since the playback pole pieces on this channel deliver the digits which are available under code control. Because of the upper and lower record pole pieces on the channel at the left, the product digits are recorded on both halves of the drum periphery and may be called for by code regardless of the drum phase.

A multiplication is performed under control of operation-code 2. When the button labeled "x" is selected on the instructional tape preparation table, three lines of coding are supplied automatically. The first of

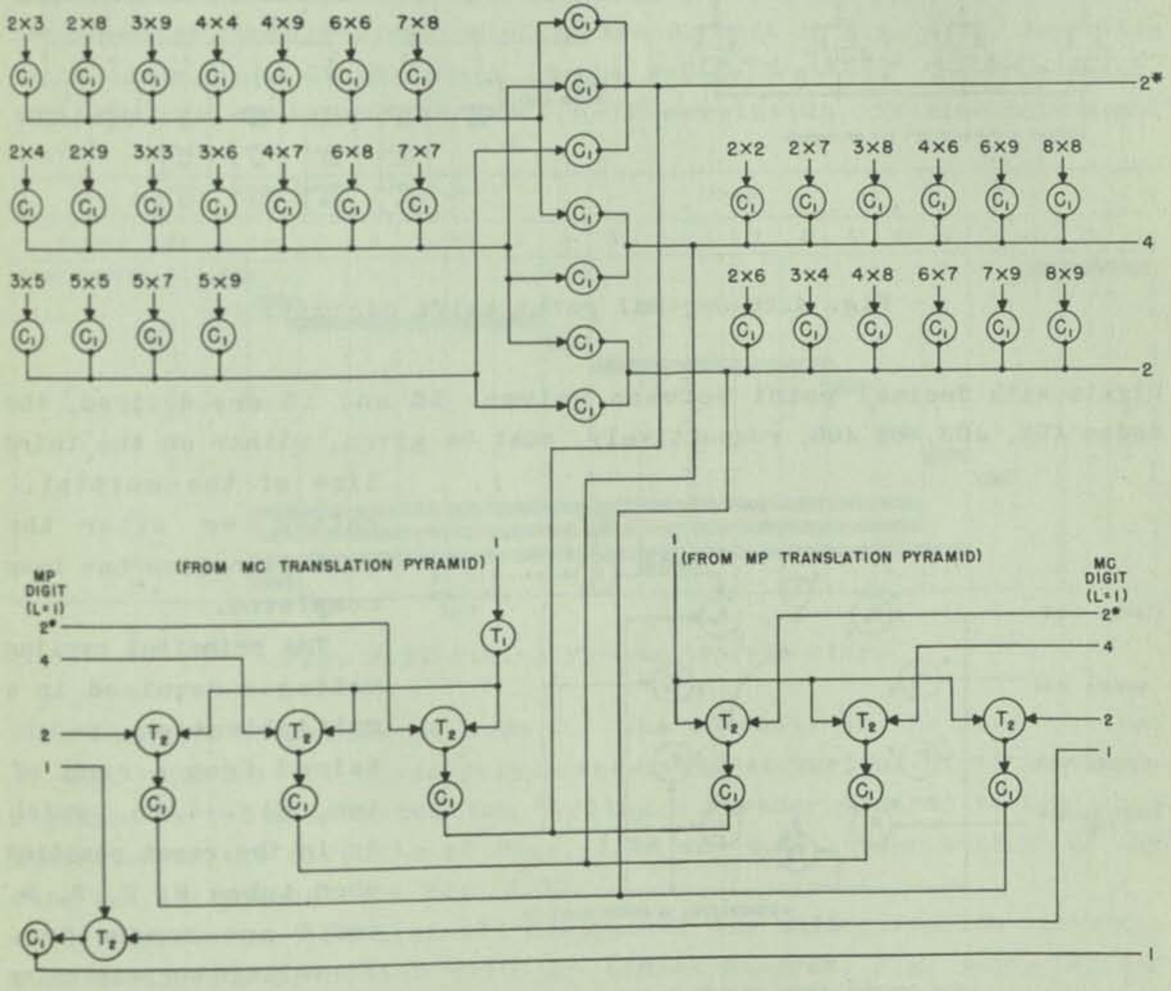


Fig. 4.19—*Right-hand component of product.*

these contains operation-code 2, while the second and third are blank (all zeros), to allow the time required to compute the product of the numbers selected by the *A* and *B* codes on the first line. The product will be delivered to the *C* transfer channel at the operating decimal point. However, if the high-order product, the low-order product, or the

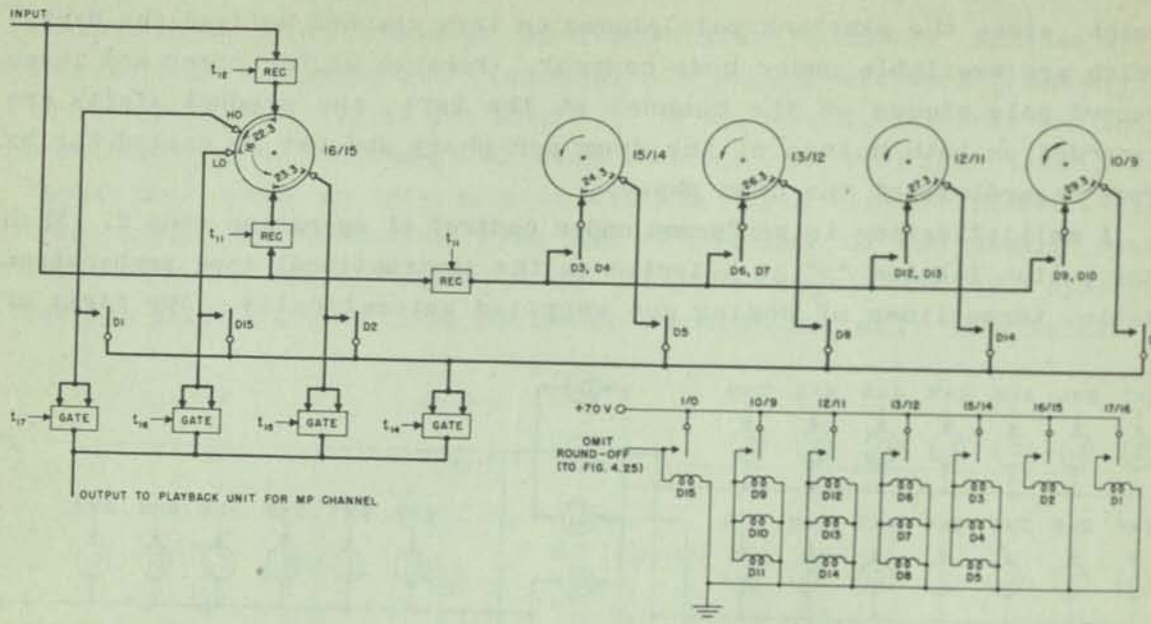


Fig. 4.20—*Decimal point shift circuit.*

digits with decimal point between columns 16 and 15 are desired, the codes *A*02, *A*03, or *A*06, respectively, must be given, either on the third line of the multiplication or after the multiplication has been completed.

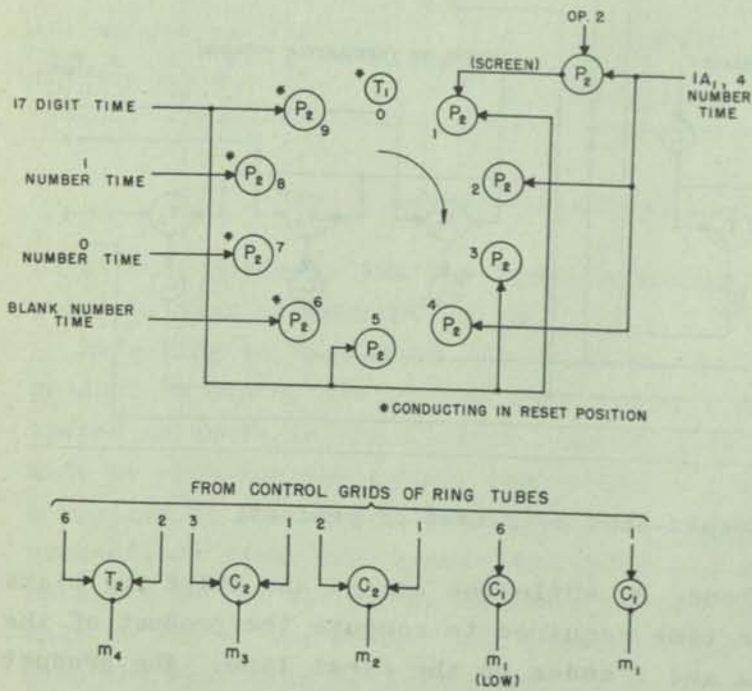


Fig. 4.21—*Multiplier ring of ten.*

The principal cycling voltages required in a multiplication are obtained from a ring of ten, Fig. 4.21, which is in the reset position when tubes 6, 7, 8, 9, and 0 are conducting. The voltage corresponding to operation-code 2 opens a pentode gate, admitting the 1 A_1 pulse during 4 number time to the screen grid of tube 1 and advancing the ring to position 1. The timing diagram, Fig. 4.22, indicates the ring positions during one multiplication, and the duration of the cycling voltages, m_1, m_2, m_3, m_4, derived from the ring tubes.

At the start of a multiplication, 1 digit time, 4 number time, cycle n, a timing scheme is adopted composed of sixteen "multiply" number times, each containing just sixteen, rather than twenty digit times. To deliver a voltage during the last digit time of each multiply number time, pulses recorded at sixteen-digit intervals in two channels on drum 7 are delivered to playback units. The output derived from one or the other channel is selected by the circuit in Fig. 4.23, depending on the drum phase at the start of the multiplication. The drum phase, indicated as an "even" or "odd" half-revolution, is also determined

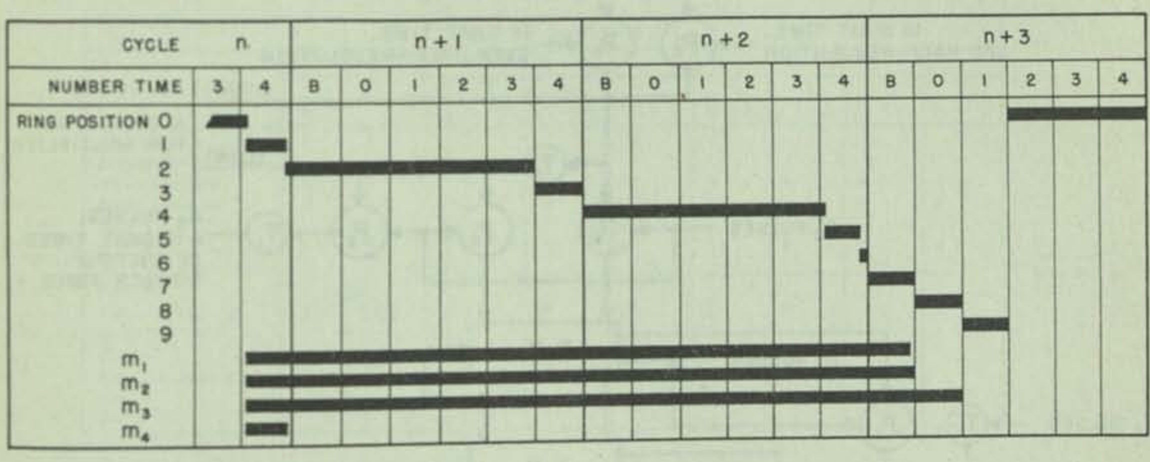


Fig. 4.22—*Timing for multiplier ring.*

from channels of pulses on drum 7. The circuits in the lower portion of Fig. 4.23 deliver gated pulses occurring at various digit times, as required by the multiplication cycling. Further control voltages are generated in the circuits of Figs. 4.24 and 4.25, under control of the ring and the pulses from Fig. 4.23.

By considering the over-all diagram of the multiplication circuits, Fig. 4.14, in conjunction with the timing diagram, Fig. 4.26, for the control voltages $t_1, t_2, \ldots, t_{18}$, a detailed study of the steps involved in one multiplication may be made. To illustrate the flow of digits, an example has been carried out in Table 4.3 for a four-column multiplier having the same controls as the unit included in the calculator. In the example, the circled digits are the carry numbers resulting from the last addition performed in adder #2 during each multiply number time. Each of these must be added not to the following sum in adder #2, but in adder #1 to the last left-hand component produced in each multiply number time. This is accomplished through the circuit shown in Fig. 4.27.

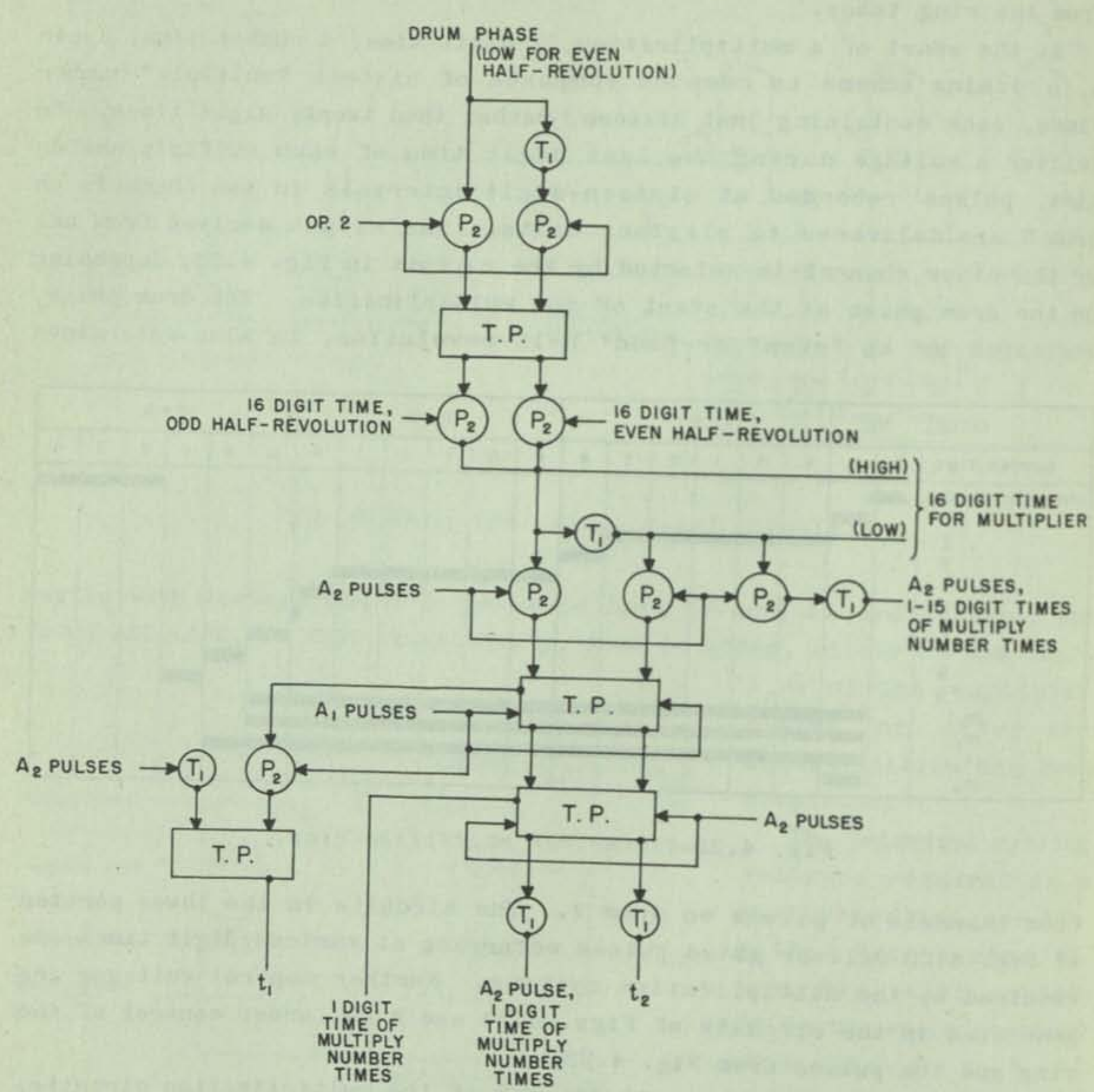


Fig. 4.23—*Multiplier cycling voltages.*

During the last digit time of each multiply number time, the regular trigger pairs for the storage of carry numbers from adder #2 are reset by an A_2 pulse, and the carry number is advanced to trigger pair 9 instead. In the following digit time it is advanced to trigger pair 3. Here there is no coincidence with a sum from adder #2, as the adder #2 output at this time, being a final product digit, is gated to trigger

TABLE 4.3

Digit Flow in Four-Column Multiplier

4321 × 9876 = 42674196

Digit Time →	Multiply Number Time 1											
	1			2			3			4		
	A_1	R	A_2	A_1	R	A_2	A_1	R	A_2	A_1	R	A_2
In Static MP	6											
Multiplier MC	1			2			3			4		
AR to Adder #2	6			2			8			4		
PP to Adder #2	0			0			0			0		
T.P. 1			6									
T.P. 2												6
Dec. Pt. Shift Ckt.		6			7			8			9	
T.P. 3						2			8			4
T.P. 4 & Adder #1							2			8		
LR to T.P. 5			0			1			1			2
T.P. 6					0			1			1	
T.P. 7 & Adder #1							0			1		
T.P. 8									2			9
PP											2	

Digit Time →	Multiply Number Time 2											
	1			2			3			4		
	A_1	R	A_2	A_1	R	A_2	A_1	R	A_2	A_1	R	A_2
In Static MP	7											
Multiplier MC	1			2			3			4		
AR to Adder #2	7			4			1			8		
PP to Adder #2	2			9			5			2		
T.P. 1			9									
T.P. 2												9
Dec. Pt. Shift Ckt.		6			8			9			6	
T.P. 3			0			3			7			0
T.P. 4 & Adder #1	4			0			3			7		
LR to T.P. 5			0			1			2			2
T.P. 6		2			0			1			2	
T.P. 7 & Adder #1	1			2			0			1		
T.P. 8			5			2			3			8
PP		9			5			2			3	

Digit Time →	Multiply Number Time 3											
	1			2			3			4		
	A_1	R	A_2	A_1	R	A_2	A_1	R	A_2	A_1	R	A_2
In Static MP	8											
Multiplier MC	1			2			3			4		
AR to Adder #2	8			6			4			2		
PP to Adder #2	3			8			2			3		
T.P. 1			1									
T.P. 2												1
Dec. Pt. Shift Ckt.		9			9			6			9	
T.P. 3			1			5			7			5
T.P. 4 & Adder #1	0			1			5			7		
LR to T.P. 5			0			1			2			3
T.P. 6		2			0			1			2	
T.P. 7 & Adder #1	2			2			0			1		
T.P. 8			2			3			5			8
PP		8			2			3			5	

TABLE 4.3 (continued)

Digit Time →	Multiply Number Time 4											
	1			2			3			4		
	A_1	R	A_2	A_1	R	A_2	A_1	R	A_2	A_1	R	A_2
In Static MP	9											
Multiplier MC	1			2			3			4		
AR to Adder #2	9			8			7			6		
PP to Adder #2	5			8			7			3		
T.P. 1			4									
T.P. 2												4
Dec. Pt. Shift Ckt.		1			6			9			1	
T.P. 3			0			7			5			0
T.P. 4 & Adder #1	5			0			7			5		
LR to T.P. 5			0			1			2			3
T.P. 6		3			0			1			2	
T.P. 7 & Adder #1	2			3			0			1		
T.P. 8			7			3			7			6
PP		8			7			3			7	

Digit Time →	Start of Standard Number Time															
	1			2			3			4			5			
	A_1	R	A_2	A_1	R	A_2	A_1	R	A_2	A_1	R	A_2	A_1	R	A_2	
In Static MP	0															
Multiplier MC	0															
AR to Adder #2	0			0			0			0			0			
PP to Adder #2	7			6			2			4			0			
T.P. 1			7			6			2			4				
T.P. 2				7			6			2			4			
Dec. Pt. Shift Ckt.		4			7			6			2			4		← complete product
T.P. 3			1			0										
T.P. 4 & Adder #1	0			1			0									
LR to T.P. 5			0			0			0							
T.P. 6		3			0			0								
T.P. 7 & Adder #1	2			3			0			0						
T.P. 8			2			4			0			0				
PP		6			2			4			0			0		

pair 1. The carry number is then advanced to trigger pair 4, and added in adder #1 to the proper delayed left hand-component.

It should be noted that information may be gated through any one of five crystal rectifier circuits, Figs. 4.14 and 4.20, to the playback unit generally associated with the *MP* channel. Under control of t_{12}, the gate for the playback pole piece on the *MP* channel is kept open at all times except blank and 0 number times, cycle $n+3$, of a multiplication. At this time, the circuit of Fig. 4.25 delivers either t_{13}, t_{14}, t_{15}, or t_{16} instead, to open gates associated with one of the playback pole pieces in the decimal point shift circuit. These gates are opened at the start of blank number time, like the fast storage playback gates, to allow the transients introduced by the gating voltages to die out. During 0 number time, the selected product digits are delivered from the playback unit through adder #1 to the *C* transfer channel under control of t_{17}. The same control voltage causes the algebraic sign of the product, previously computed in the *A* and *B* sign storage circuits

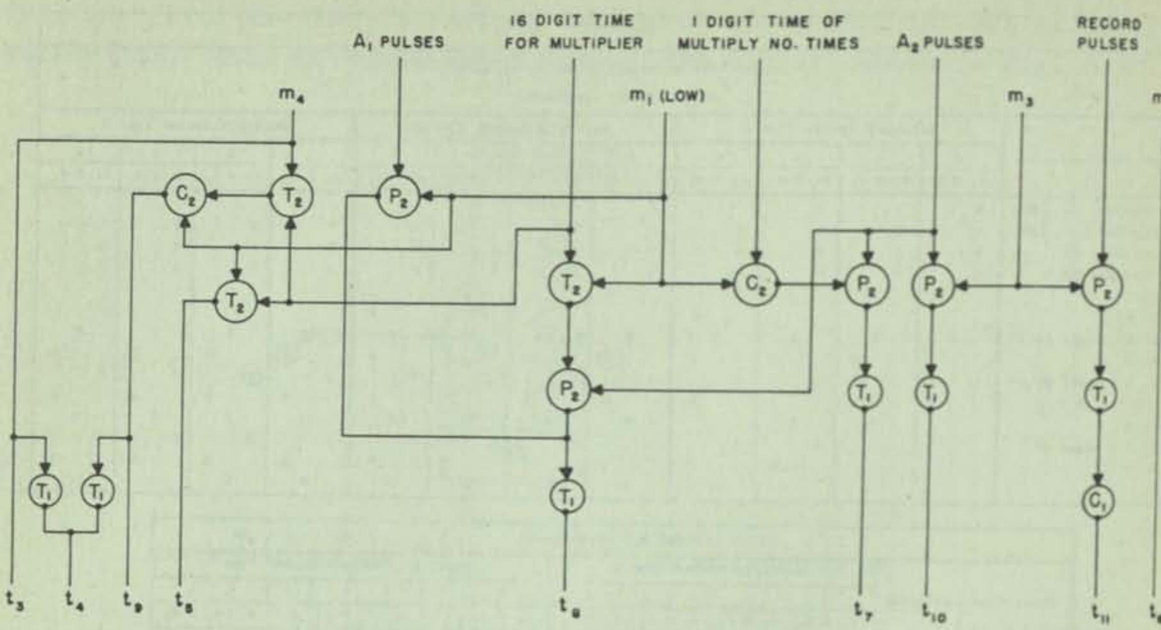


Fig. 4.24—*Multiplier control voltages* t_3-t_{11}.

and stored in a trigger pair, Fig. 4.28, to be recorded on the *C* transfer channel at 17 digit time.

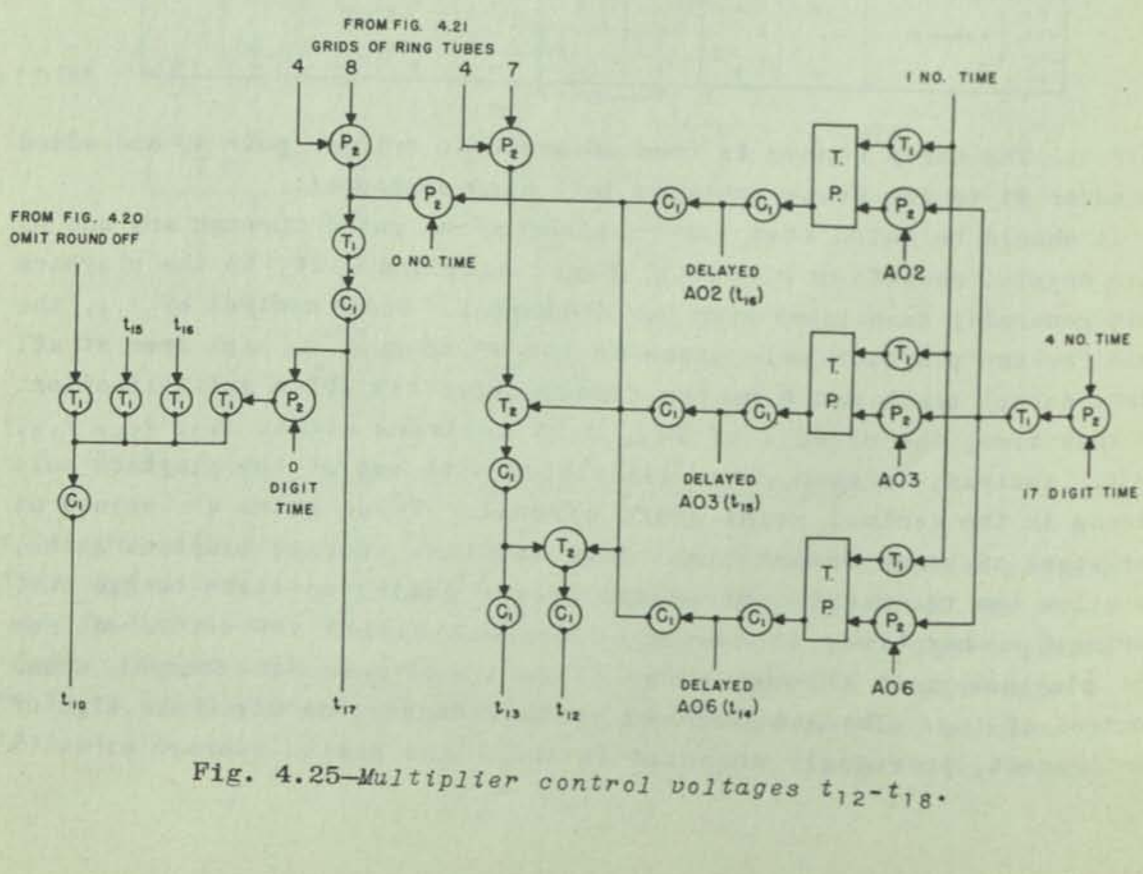


Fig. 4.25—*Multiplier control voltages* t_{12}-t_{18}.

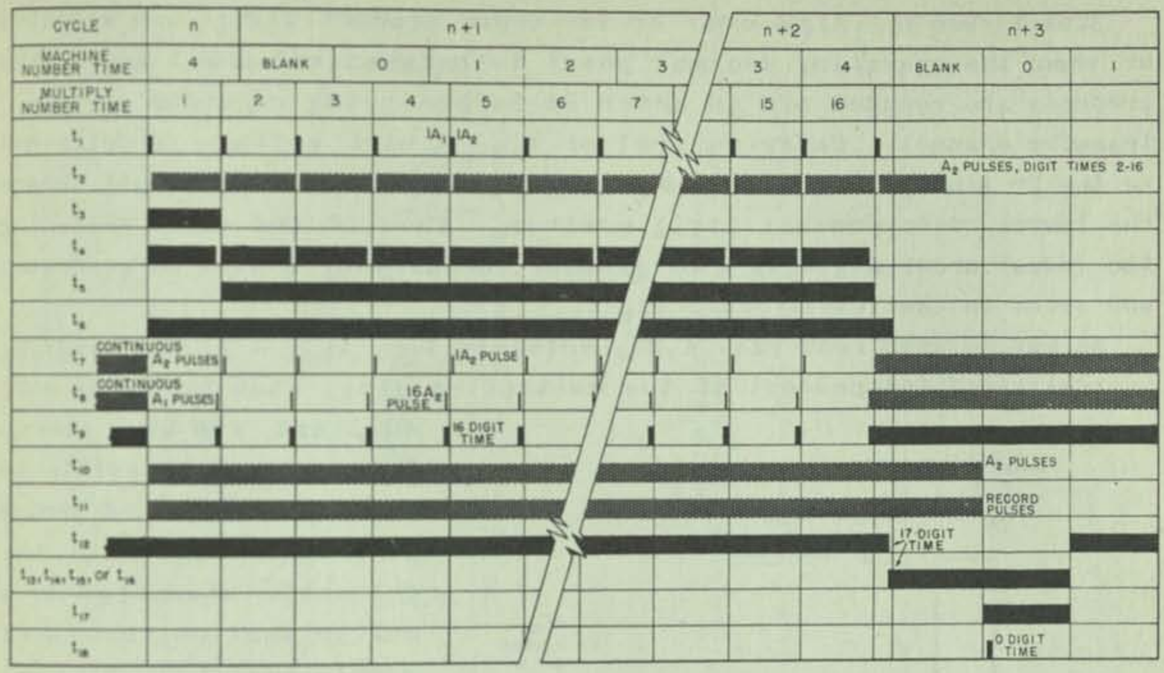


Fig. 4.26—*Timing of control voltages for over-all diagram.*

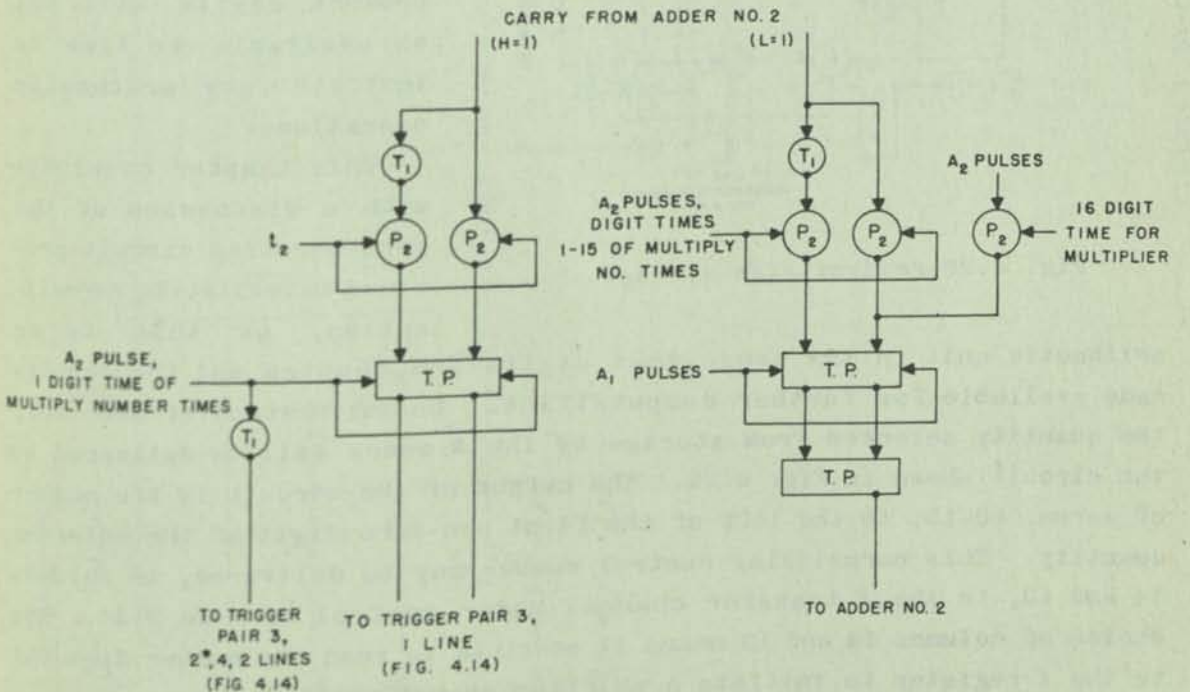


Fig. 4.27—*Storage of carry numbers, adder #2.*

Except when the high-order or low-order product digits are selected or when the operating decimal point is between columns 1 and 0, the products are rounded off in adder #1 before being recorded on the *C* transfer channel. Under control of t_{18}, a high voltage is delivered to the 2* line of the addend at 0 digit time, 0 number time, just before the lowest order product digit arrives. Then if the digit preceding the lowest order digit is 5 or greater, a carry of 1 will be generated and added to the lowest order digit.

As may be seen from Fig. 4.25, voltages t_{14}, t_{15}, t_{16}, t_{17}, and t_{18} are delivered independent of the multiplier ring, when the codes *A*02, *A*03, and *A*06 are given. This makes it possible to read the associated groups of product digits to the *C* transfer channel after a multiplication has been completed and the ring has stopped cycling. No *B* code or operation-code is given with a code *A*02, *A*03 or *A*06, since the selected product digits will not be available in time to initiate any arithmetic operations.

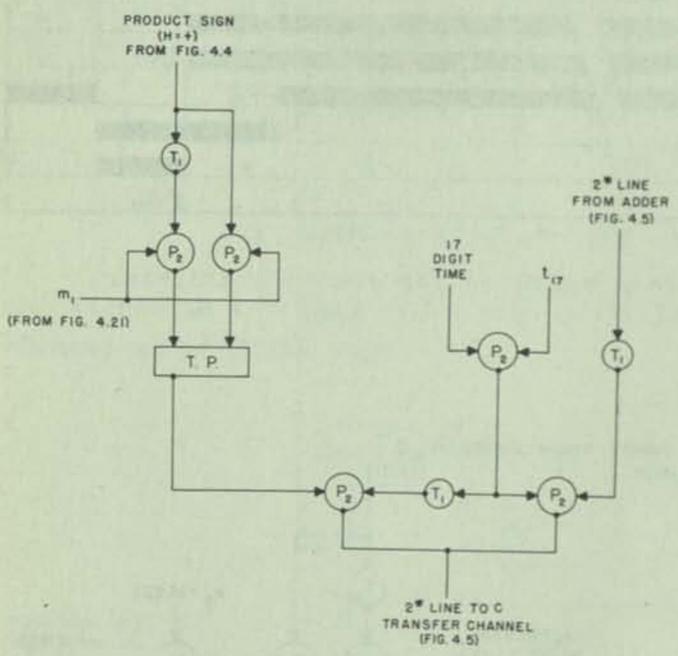


Fig. 4.28—*Product sign storage.*

This chapter concludes with a discussion of the zero-counting circuit provided to facilitate normalization, as this is an arithmetic unit in the sense that digits are counted and the results made available for further computations. Under control of code *A*07, the quantity selected from storage by the *B* codes will be delivered to the circuit shown in Fig. 4.29. The output of the circuit is the number of zeros, 00-15, to the left of the first non-zero digit of the entering quantity. This normalizing control number may be delivered, in columns 14 and 13, to the *B* transfer channel under control of code *B*13*. The choice of columns 14 and 13 makes it possible to read the number directly to the *l* register to initiate a shifting operation.

The zero-counting circuit consists essentially of an input ring of

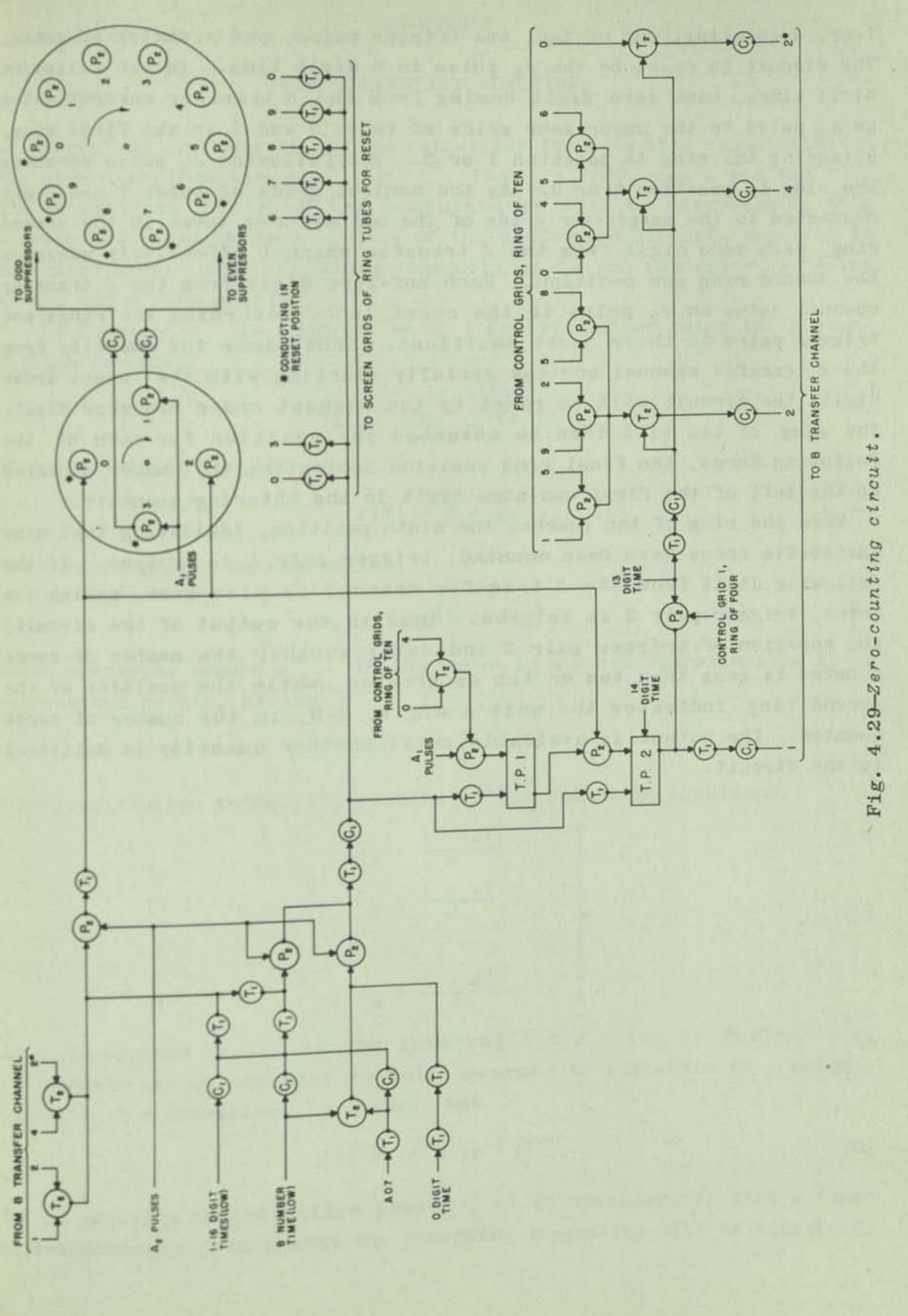


Fig. 4.29—*Zero-counting circuit.*

four, a counting ring of ten, two trigger pairs, and a number of gates. The circuit is reset by the A_2 pulse in 0 digit time. In the following digit times, each zero digit coming from the B transfer channel gates an A_2 pulse to the suppressor grids of tubes 0 and 2 in the first ring, advancing the ring to position 1 or 3. The following A_1 pulse advances the ring to position 2 or 0. As the control grids of tubes 1 and 3 are connected to the suppressor grids of the odd and even tubes in the second ring, each zero digit from the B transfer channel effectively advances the second ring one position. Each non-zero digit from the B transfer channel gates an A_2 pulse to the reset line, returning the rings and trigger pairs to their reset positions. Thus since the quantity from the B transfer channel arrives serially starting with the lowest order digit, the circuit will be reset by the highest order non-zero digit. The ring of ten will then be advanced one position for each of the following zeros, the final ring position indicating the number of zeros to the left of the first non-zero digit in the entering quantity.

When the ring of ten reaches the ninth position, indicating that nine successive zeros have been counted, trigger pair 1 is tripped. If the following digit from the B transfer channel is also zero, making ten zeros, trigger pair 2 is tripped. Thus in the output of the circuit, the position of trigger pair 2 indicates whether the number of zeros counted is less than ten or ten or greater, while the position of the second ring indicates the unit's digit, 0-9, in the number of zeros counted. The output is available until another quantity is delivered to the circuit.

CHAPTER V

THE ELEMENTARY FUNCTIONS

Provisions are made within the calculator for automatic computation of the elementary functions x^{-1}, $x^{-1/2}$, $\log_{10} x$, 10^x, $\cos x$ and $\tan^{-1} x$. The theoretical basis of the computations will be discussed in this chapter, followed by descriptions of the associated circuits required in the calculator. Complete coding instructions for the functions are given at the end of the chapter.

The function x^{-1} is computed by an iterative method using the Newton-Raphson formula,

$$z_{m+1} = z_m - \frac{f(z_m)}{f'(z_m)}. \tag{1}$$

When specialized for the case,

$$f(z) = x - \frac{1}{z},$$

the formula yields an expression,

$$z_{m+1} = z_m(2 - xz_m), \tag{2}$$

which converges toward x^{-1}. If the error in a first approximation z_0 is represented by e/x, giving

$$z_0 = \frac{1 + e}{x},$$

then substitution in Eq. (2) results in the following equations,

$$\left.\begin{aligned} z_1 &= \frac{1 - e^2}{x}, \\ z_2 &= \frac{1 - e^4}{x}, \\ &\cdots\cdots \\ z_m &= \frac{1 - e^{2^m}}{x}. \end{aligned}\right\} \tag{3}$$

If x is assumed to lie in the interval $1 \leq x < 10$, it follows that reciprocals may be computed to full accuracy by a machine of p columns capacity in m iterations, provided that

$$|e| < (5 \times 10^{-p})^{2^{-m}}. \tag{4}$$

Before applying the iterative process, it is necessary to find a first approximation, z_0, such that the foregoing inequality will be satisfied.

To do this, set

$$z_0 = \frac{1 + e(x)}{x} = a_1 + a_2 x + \cdots + a_n x^{n-1}.$$

Then,

$$e(x) = -1 + a_1 x + a_2 x^2 + \cdots + a_n x^n.$$

The value, $|e_{max}(x)|$, can be made as small as possible throughout the whole of a certain subinterval of the range $1 \leq x < 10$ if $e(x)$ is represented by a Tchebychef polynomial[1], which has the property,

$$|e_{max}| = |e(x_0)| = |e(x_1)| = |e_1| = |e_2| = \cdots = |e_{n-1}|,$$

where x_0 and x_1 are the limits of the subinterval in question, and the $e(x_i)$ are the n-1 maximum and minimum values of $e(x)$ in that same interval.

The general Tchebychef polynomial for the interval (x_0, x_1) is

$$T_n(x) = A_n \cos\left[n \cos^{-1}\left(\frac{2x - x_0 - x_1}{x_1 - x_0}\right)\right],$$

where A_n is arbitrary. For example, when $n = 2$

$$T_2(x) = A_2 \cos\left[2 \cos^{-1}\left(\frac{2x - x_0 - x_1}{x_1 - x_0}\right)\right].$$

Thus when

$$A_2 = \frac{-(x_1 - x_0)^2}{x_0{}^2 + 6x_0x_1 + x_1{}^2},$$

then

$$e(x) = -1 + a_1 x + a_2 x^2,$$

where

$$a_1 = \frac{8(x_0 + x_1)}{x_0{}^2 + 6x_0x_1 + x_1{}^2}, \tag{5}$$

$$a_2 = \frac{-8}{x_0{}^2 + 6x_0x_1 + x_1{}^2}. \tag{6}$$

Also,

$$e_m = \frac{(x_1 - x_0)^2}{x_0{}^2 + 6x_0x_1 + x_1{}^2}$$

$$= \frac{(k - 1)^2}{k^2 + 6k + 1}, \tag{7}$$

where $k = x_0/x_1$.

Clearly an expression similar to Eq. (7) may be derived from each Tchebychef polynomial of degree n. These expressions may be used together with the Ineq. (4) to relate the variables p, m, and $k = x_0/x_1$. Since k^q must be approximately equal to $1/10$, where q is the number of arcs of degree n-1 used to approximate z_0 in the interval $1 \leq x < 10$, it follows that a suitable study of the data here given may be made to yield that set of z_0 for which the total amount of stored information is a minimum or for which the number of arithmetic operations necessary to the computation of x^{-1} is a minimum, whichever is desired. Actually, this choice also depends on the storage system of the calculator and the methods employed for the computation of two other functions, $x^{-1/2}$ and $\log_{10} x$. Hence further comments will be made after these functions have been discussed.

To compute the function $x^{-1/2}$, Eq. (1) is specialized for the case,

$$f(z) = x - \frac{1}{z^2}.$$

Then,

$$z_{m+1} = \frac{1}{2} z_m(3 - xz_m{}^2), \tag{8}$$

and z_{m+1} converges toward the value $x^{-1/2}$. If

$$z_0 = \frac{1 + e}{x^{1/2}},$$

then, by Eq. (8),

$$z_1 \doteq \frac{1 - 3e^2/2}{x^{1/2}}, \tag{9}$$

$$z_2 \doteq \frac{1 - 27e^4/8}{x^{1/2}},$$

$$\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot$$

$$z_m \doteq \frac{1 - (3/2)^r \; e^{2m}}{x^{1/2}},$$

where $r = 2^m - 1$. Assume x lies in the interval $1 \leq x < 10$, recognizing that multiplication by a power of ten to put x within this interval may require multiplication of z_m by $10^{-1/2}$. It follows that $x^{-1/2}$ may be computed to full accuracy by a machine of p columns capacity in m iterations, provided that

$$|e| < [5(2/3)^r 10^{-p}]^{2^{-m}}. \tag{10}$$

In order to find z_0, consider the function,

$$e(s) = -1 + a_1 s + a_3 s^3, \tag{11}$$

and let a_1 and a_3 be determined such that in the interval $s_0 \leq s < s_1$,

$$|e(s_0)| = |e(s_1)| = |e_m|,$$

where e_m is the maximum value of the function in the interval in question. It may be shown that

$$a_1 = \frac{2(s_0^2 + s_0 s_1 + s_1^2)}{s_0^2 s_1 + s_0 s_1^2 + 2S}, \tag{12}$$

$$a_3 = \frac{-2}{s_0^2 s_1 + s_0 s_1^2 + 2S}, \tag{13}$$

$$e_m = 1 - \frac{2(s_0^2 s_1 + s_0 s_1^2)}{s_0^2 s_1 + s_0 s_1^2 + 2S},$$

where

$$S = \left[\frac{s_0^2 + s_0 s_1 + s_1^2}{3}\right]^{3/2}.$$

By setting $k = (s_0/s_1)^2$, it follows that

$$e_m = 1 - \frac{2(k + \sqrt{k})}{k + \sqrt{k} + 2[(k+\sqrt{k}+1)/3]^{3/2}}. \tag{14}$$

In Eq. (11), set $x = s^2$, to obtain an expression for z_0 in the interval $x_0^{-1/2} \leq x^{-1/2} < x_1^{-1/2}$, namely,

$$z_0 = \frac{1 + e}{x^{1/2}} = a_1 + a_3 x.$$

Inequality (10) and Eq. (14) may now be used to determine the number of constants which must be stored in a machine of p columns capacity if the function $x^{-1/2}$ is to be computed to full accuracy in m iterations.

The computation of the function $\log_{10} x$ is based on the equations,

$$\log_{10} x = \log_{10} a + (\log_{10} e)[z - \frac{z^2}{2} + \frac{z^3}{3} - \cdots + \frac{(-1)^{n+1}}{n} z^n] + R, \tag{15}$$

and

$$z = \frac{x}{a} - 1,$$

where

$$R < (\log_{10} e)(\frac{z^{n+1}}{n+1}). \tag{16}$$

Suppose the interval $1 \leq x < 10$ to be divided into subintervals with each of which there is associated a value of a. Further, let the values of $1/a$ and $\log_{10} a$ be stored in the calculator. Assuming $x_0 \leq x < x_1$ to be one of the subintervals, set

$$\frac{x_0}{a} - 1 = -(\frac{x_1}{a} - 1).$$

Then the most appropriate value of a in the interval is

$$a = \frac{x_0 + x_1}{2}. \tag{17}$$

Hence the maximum value of z in the interval is

$$z_m = \frac{2x_1}{x_0 + x_1} - 1$$

$$= \frac{1 - k}{1 + k}, \tag{18}$$

where $k = x_0/x_1$. Inequality (16) and Eq. (18) may now be used to determine, as a function of the number of constants stored, the number of arithmetic operations required for the computation of $\log_{10} x$.

The fact that in the storage system of the calculator ten numbers are stored around the drum periphery in one decimal channel suggests that the interval $1 \leq x < 10$ be broken into ten subintervals for the three foregoing functions. Then by storing constants appropriate to x in each interval at the ten number times, a single argument-sensing circuit may be used to control the number-time selection on the channels containing the parameters necessary to the computations. For simplicity and economy, it is desirable to sense only the first two digits of x. On the other hand, if the sensing circuits placed no limitation on the design of the apparatus, the subintervals should be delimited by the numbers $t^0, t^1, \ldots, t^{10}$, where $t = 10^{1/10}$, making $k = 0.7943$ for all intervals. The values, t^n, are given in Table 5.1, together with certain two-digit approximations more suitable for sensing purposes. The table also lists the values of k resulting from the adoption of the approximate values of t^n. As the difference from the theoretical value of k never exceeds 0.0443, the approximate limits may be adopted with complete assurance that the limitations on the design of sensing circuits due to engineering considerations will not increase the number of stored constants nor the number of arithmetic operations required for the computation of the functions x^{-1}, $x^{-1/2}$, and $\log_{10} x$.

Using the minimum value of k and assuming that the first approximations for x^{-1} and $x^{-1/2}$ are to be obtained using polynomials of degree

TABLE 5.1			
Limits of Sensing Intervals			
Interval	Theoretical Limits	Approximate Limits	k
0	$1.000 \leq x < 1.259$	$1.0 \leq x < 1.2$	0.833
1	$1.259 \leq x < 1.585$	$1.2 \leq x < 1.5$	0.800
2	$1.585 \leq x < 1.995$	$1.5 \leq x < 2.0$	0.750
3	$1.995 \leq x < 2.512$	$2.0 \leq x < 2.5$	0.800
4	$2.512 \leq x < 3.162$	$2.5 \leq x < 3.0$	0.833
5	$3.162 \leq x < 3.981$	$3.0 \leq x < 4.0$	0.750
6	$3.981 \leq x < 5.012$	$4.0 \leq x < 5.0$	0.800
7	$5.012 \leq x < 6.310$	$5.0 \leq x < 6.0$	0.833
8	$6.310 \leq x < 7.943$	$6.0 \leq x < 8.0$	0.750
9	$7.943 \leq x < 10.000$	$8.0 \leq x < 10.0$	0.800

two and three, respectively, reference to Eqs. (7) and (14) shows that

$e_m = 0.0103$ for x^{-1},

$e_m = 0.0039$ for $x^{-1/2}$.

With these values of e_m, it can be seen from Eqs. (3) and (9) that the number of iterations required for the improvement of the reciprocal and reciprocal square root is three in both cases. If either approximating polynomial for x^{-1} or $x^{-1/2}$ were increased by one in degree, ten more constants would have to be stored. Since the added multiplication required to evaluate z_0 would not replace an iterative operation, no gain would be made in speed.

TABLE 5.2		
First Approximation Constants for x^{-1}		
Interval	a_1	$-a_2$
0	1.825 726	0.829 876
1	1.490 683	0.552 105
2	1.154 639	0.329 897
3	0.894 410	0.198 758
4	0.730 290	0.132 780
5	0.577 320	0.082 474
6	0.447 205	0.049 689
7	0.365 145	0.033 195
8	0.288 660	0.020 619
9	0.223 603	0.012 422

The values of the constants required for the computation of x^{-1}, $x^{-1/2}$, and $\log_{10} x$ insofar as the sensing circuit is concerned are computed from Eqs. (5), (6), (12), (13), and (17). These values are listed in Tables 5.2, 5.3, and 5.4. In addition, the numbers 0.5, 1.5, and 2.0 are required by the iteration formulas for x^{-1} and $x^{-1/2}$. The factors 1.0 and $10^{-1/2}$ must be provided to correct the reciprocal square root in the event that an even or odd number of columns of shift,

TABLE 5.3

First Approximation Constants for $x^{-1/2}$

Interval	a_1	$-a_3$
0	1.433 410	0.434 967
1	1.295 342	0.320 499
2	1.140 234	0.217 933
3	1.003 367	0.148 954
4	0.906 568	0.110 039
5	0.806 267	0.077 051
6	0.709 488	0.052 663
7	0.641 041	0.038 905
8	0.570 117	0.027 242
9	0.501 684	0.018 619

respectively, was employed to bring the argument into the interval $1 \leq x < 10$. Finally, the function $\log_{10} x$ requires that the series coefficients in Eq. (15) be made available.

Sixteen terms of the series in Eq. (15) must be retained in order to make the maximum error in $\log_{10} x$ less than 5×10^{-16}. However, the twelfth, thirteenth, fourteenth, fifteenth and sixteenth terms of the series may be removed by using Tchebychef polynomials of order 12,13,...,16, for the interval (-1,1). These are:

$$T_{12}(x) = x^{12} - 3x^{10} + \frac{27}{8}x^8 - \frac{7}{4}x^6 + \frac{105}{256}x^4 - \frac{9}{256}x^2 + \frac{1}{2048},$$

$$T_{13}(x) = x^{13} - \frac{13}{4}x^{11} + \frac{65}{16}x^9 - \frac{39}{16}x^7 + \frac{91}{128}x^5 - \frac{91}{1024}x^3 + \frac{13}{4096}x,$$

$$T_{14}(x) = x^{14} - \frac{7}{2}x^{12} + \frac{77}{16}x^{10} - \frac{105}{32}x^8 + \frac{147}{128}x^6 - \frac{49}{256}x^4 + \frac{49}{4096}x^2 - \frac{1}{8192},$$

$$T_{15}(x) = x^{15} - \frac{15}{4}x^{13} + \frac{45}{8}x^{11} - \frac{275}{64}x^9 + \frac{225}{128}x^7 - \frac{189}{512}x^5 + \frac{35}{1024}x^3 - \frac{15}{16384}x,$$

$$T_{16}(x) = x^{16} - 4x^{14} + \frac{13}{2}x^{12} - \frac{11}{2}x^{10} + \frac{165}{64}x^8 - \frac{21}{32}x^6 + \frac{21}{256}x^4 - \frac{1}{256}x^2 + \frac{1}{32768}.$$

Using these polynomials, Eq. (15) may be rewritten as

$$\log_{10} x = \log_{10} a + a_0 + a_1 z + \cdots + a_{11} z^{11} + R',$$

TABLE 5.4

Constants Required in the Computation of $\log_{10} x$

Interval	a	$1/a$	$\log a$
0	1.10	0.90909 09090 90909	0.04139 26851 58225
1	1.35	0.74074 07407 40741	0.13033 37684 95006
2	1.75	0.57142 85714 28571	0.24303 80486 86294
3	2.25	0.44444 44444 44444	0.35218 25181 11362
4	2.75	0.36363 63636 36364	0.43933 26938 30263
5	3.50	0.28571 42857 14286	0.54406 80443 50276
6	4.50	0.22222 22222 22222	0.65321 25137 75344
7	5.50	0.18181 81818 18182	0.74036 26894 94244
8	7.00	0.14285 71428 57143	0.84509 80400 14257
9	9.00	0.11111 11111 11111	0.95424 25094 39325

where

$$a_0 = 0.00000\ 00000\ 00001$$
$$a_1 = 0.43429\ 44819\ 03244$$
$$a_2 = -0.21714\ 72409\ 56402$$
$$a_3 = 0.14476\ 48273\ 12253$$
$$a_4 = -0.10857\ 36177\ 52932$$
$$a_5 = 0.08685\ 88920\ 14708$$
$$a_6 = -0.07238\ 29806\ 84856$$
$$a_7 = 0.06204\ 27994\ 03830$$
$$a_8 = -0.05423\ 35532\ 60322$$
$$a_9 = 0.04819\ 56528\ 39172$$
$$a_{10} = -0.04572\ 21594\ 19618$$
$$a_{11} = 0.04177\ 76646\ 78944$$

$$|R'| \leq 1.5 \times 10^{-15}.$$

To simplify the computation of 10^x, where x is in the range $-15 \leq x \leq 15$, the function is considered in the form,

$$10^x = 10^{-15}(10^{15+x}).$$

Then the exponent,

$$15+x = m.abc\ldots$$

is positive for any value of x. To compute 10^x, the two factors $10^{0.a}$ and $10^{0.0\,bc\ldots}$ are determined, and their product shifted m-15 columns. The digit a may be delivered to the i register to control the selection of $10^{0.a}$, provided that the ten values, $10^{0.a}$, Table 5.5, are stored on one of the constant channels in the calculator. The factor, $10^{0.0\,bc\ldots}$, may be evaluated by the series expansion,

TABLE 5.5
Constants Required in the Computation of 10^x
$10^{0.0} = 1.00000\ 00000\ 00000$
$10^{0.1} = 1.25892\ 54117\ 94167$
$10^{0.2} = 1.58489\ 31924\ 61113$
$10^{0.3} = 1.99526\ 23149\ 68880$
$10^{0.4} = 2.51188\ 64315\ 09580$
$10^{0.5} = 3.16227\ 76601\ 68379$
$10^{0.6} = 3.98107\ 17055\ 34973$
$10^{0.7} = 5.01187\ 23362\ 72723$
$10^{0.8} = 6.30957\ 34448\ 01932$
$10^{0.9} = 7.94328\ 23472\ 42815$

$$10^z = 1 + t + \frac{t^2}{2!} + \cdots + \frac{t^n}{n!} + R, \tag{19}$$

where

$$t = z \log_e 10$$

and

$$R < \frac{t^{n+1}}{(n+1)!(1-t)}.$$

As the maximum value of z is 0.1, the value of t does not exceed 0.2303. Then twelve terms of the series in Eq. (19) must be retained, making the remainder less than 6.04×10^{-17}. The tenth, eleventh, and twelfth terms of the series may be removed through the use of the following Tchebychef polynomials for the interval (0,1):

$$T_9(x) = x^9 - \frac{9}{2}x^8 + \frac{135}{16}x^7 - \frac{273}{32}x^6 + \frac{1287}{256}x^5 - \frac{891}{512}x^4 + \frac{693}{2048}x^3 - \frac{270}{8192}x^2$$
$$+ \frac{81}{65536}x - \frac{1}{131072},$$

$$T_{10}(x) = x^{10} - 5x^9 + \frac{85}{8}x^8 - \frac{25}{2}x^7 + \frac{2275}{256}x^6 - \frac{1001}{256}x^5 + \frac{2145}{2048}x^4 - \frac{330}{2048}x^3$$
$$+ \frac{825}{65536}x^2 - \frac{25}{65536}x + \frac{1}{524288},$$

$$T_{11}(x) = x^{11} - \frac{11}{2}x^{10} + \frac{209}{16}x^9 - \frac{561}{32}x^8 + \frac{935}{64}x^7 - \frac{1001}{128}x^6 + \frac{11011}{4096}x^5$$
$$- \frac{4719}{8192}x^4 + \frac{4719}{65536}x^3 - \frac{605}{131072}x^2 + \frac{121}{1048576}x - \frac{1}{2097152}.$$

Using these polynomials, Eq. (19) may be rewritten,

$$10^z = a_0 + a_1 t + \cdots + a_8 t^8 + R', \tag{20}$$

where

$$\begin{aligned}
a_0 &= 1.00000 \quad 00000 \quad 00000\\
a_1 &= 0.99999 \quad 99999 \quad 99970\\
a_2 &= 0.50000 \quad 00000 \quad 03472\\
a_3 &= 0.16666 \quad 66665 \quad 12241\\
a_4 &= 0.04166 \quad 66701 \quad 05326\\
a_5 &= 0.00833 \quad 32903 \quad 70962\\
a_6 &= 0.00138 \quad 92035 \quad 64614\\
a_7 &= 0.00019 \quad 70747 \quad 85410\\
a_8 &= 0.00002 \quad 78384 \quad 16808
\end{aligned}$$

$$|R'| < 1.04 \times 10^{-16}.$$

Further economies through the use of the polynomial $T_8(x)$ lead to an excessively large remainder. Hence Eq. (20) was adopted, requiring that the constant storage include the coefficients a_i, as well as the values $10^{0.a}$ and $\log_{10} e$, for the computation of 10^x.

To evaluate the function

$$\cos x = \cos |x|,$$

let

$$|x| = 2\pi I + 2\pi J,$$

where I is an integer and J a decimal fraction. It follows that

$$\begin{aligned}\cos |x| &= \cos 2\pi J \\ &= -\cos (2\pi J - \pi).\end{aligned}$$

Now let

$$4z = 2\pi J - \pi.$$

Then $\cos z$ may be calculated by the series,

$$\cos z = 1 - \frac{z^2}{2!} + \frac{z^4}{4!} - \cdots + \frac{(-1)^n z^{2n}}{(2n)!} + R. \tag{21}$$

Since

$$\cos 2z = 2\cos^2 z - 1,$$

it is evident that

$$\cos x = -\cos 4z = 1 - 2\cos^2 (2z).$$

As the maximum value of z is $\pi/4$, nine terms of the series in Eq. (21) must be retained, making

$$|R| < 2.019 \times 10^{-18}.$$

Of these terms, the eighth and ninth may be removed through the use of the following Tchebychef polynomials of orders 14 and 16 for the interval $(-1,1)$:

$$T_{14}(x) = x^{14} - \frac{7}{2}x^{12} + \frac{77}{16}x^{10} - \frac{105}{32}x^8 + \frac{147}{128}x^6 - \frac{49}{256}x^4 + \frac{49}{4096}x^2 - \frac{1}{8192},$$

$$T_{16}(x) = x^{16} - 4x^{14} + \frac{13}{2}x^{12} - \frac{11}{2}x^{10} + \frac{165}{64}x^8 - \frac{21}{32}x^6 + \frac{21}{256}x^4 - \frac{1}{256}x^2 + \frac{1}{32768}.$$

Equation (21) may then be written

$$\cos z = a_0 + a_2 z^2 + a_4 z^4 + \cdots + a_{12} z^{12} + R', \tag{22}$$

where

$$\begin{aligned} a_0 &= 1.00000\ 00000\ 00000 \\ a_2 &= -0.49999\ 99999\ 99993 \\ a_4 &= 0.04166\ 66666\ 66472 \end{aligned}$$

$$\begin{aligned}
a_6 &= -0.00138\ 88888\ 86998 \\
a_8 &= 0.00002\ 48015\ 78540 \\
a_{10} &= -0.00000\ 02755\ 52341 \\
a_{12} &= 0.00000\ 00020\ 63047
\end{aligned}$$

$$|R'| < 4.92 \times 10^{-17}.$$

Again, further economies through the use of the Tchebychef polynomial $T_{12}(x)$ lead to an excessively large remainder. Hence Eq. (22) was adopted as the basis of the computation of the cosine function. The constants required for the computation are the values $1/2\pi$, $\pi/4$, $\pi/2$, and the series coefficients given above.

TABLE 5.6

Sensing Intervals for $\tan^{-1} x$

Interval	b	arc tan b
$0 \leq \lvert x\rvert < 0.40$	0.21	0.20699 21942 19821
$0.40 \leq \lvert x\rvert < 1.00$	0.66666 66666 66667	0.58800 26035 47568
$1.00 \leq \lvert x\rvert < 2.40$	1.5	0.98279 37232 47329
$2.40 \leq \lvert x\rvert < \infty$	4.76190 47619 04762	1.36380 41325 75076

The function $\tan^{-1} x$ is computed by using the following relations:

$$\tan^{-1}(-x) = -\tan^{-1} x$$

$$\tan^{-1}|x| = \tan^{-1} b + \tan^{-1} w$$

$$w = \frac{|x| - b}{1 + b|x|}$$

$$\tan^{-1} w = w - \frac{w^3}{3} + \frac{w^5}{5} - \cdots + \frac{(-1)^{n-1}w^{2n-1}}{2n-1} + R.$$

If the range $0 \leq x < \infty$ is divided into four intervals as shown in Table 5.6, then a value of b and $\tan^{-1} b$ may be associated with any argument x, and the value of w may be computed, where $w_m \leq 0.21$. The function $\tan^{-1} w$ is computed by the series expansion, in which eleven terms are required to make the remainder less than 5×10^{-16}. Four of these terms may be removed by using the following Tchebychef polynomials, normalized to the interval (-1,1):

$$T_{15}(x) = x^{15} - \frac{15}{4}x^{13} + \frac{45}{8}x^{11} - \frac{275}{64}x^9 + \frac{225}{128}x^7 - \frac{189}{512}x^5 + \frac{35}{1024}x^3 - \frac{15}{16384}x,$$

$$T_{17}(x) = x^{17} - \frac{17}{4}x^{15} + \frac{119}{16}x^{13} - \frac{221}{32}x^{11} + \frac{935}{256}x^9 - \frac{561}{512}x^7 + \frac{357}{2048}x^5$$

$$- \frac{51}{4096}x^3 + \frac{17}{65536}x,$$

$$T_{19}(x) = x^{19} - \frac{19}{4}x^{17} + \frac{19}{2}x^{15} - \frac{665}{64}x^{13} + \frac{1729}{256}x^{11} - \frac{2717}{1024}x^9 + \frac{627}{1024}x^7$$

$$- \frac{627}{8192}x^5 + \frac{285}{65536}x^3 - \frac{19}{262144}x,$$

$$T_{21}(x) = x^{21} - \frac{21}{4}x^{19} + \frac{189}{16}x^{17} - \frac{119}{8}x^{15} + \frac{735}{64}x^{13} - \frac{5733}{1024}x^{11} + \frac{7007}{4096}x^9$$

$$- \frac{1287}{4096}x^7 + \frac{2079}{65536}x^5 - \frac{385}{262144}x^3 + \frac{21}{1048576}x.$$

The series may then be rewritten,

$$\tan^{-1} w = a_1 w + a_3 w^3 + \cdots + a_{13} w^{13} + R',$$

where

$$\begin{aligned}
a_1 &= 0.99999 \quad 99999 \quad 99983\\
a_3 &= -0.33333 \quad 33333 \quad 18890\\
a_5 &= 0.19999 \quad 99964 \quad 47985\\
a_7 &= -0.14285 \quad 67570 \quad 61678\\
a_9 &= 0.11108 \quad 95377 \quad 16742\\
a_{11} &= -0.09025 \quad 92174 \quad 25174\\
a_{13} &= 0.06680 \quad 63477 \quad 93420
\end{aligned}$$

$$|R'| < 5 \times 10^{-16}.$$

Thus the constants required for the computation of $\tan^{-1} x$ are the values of b and $\tan^{-1} b$ and the series coefficients. As the intervals associated with b do not correspond to those used with the functions x^{-1}, $x^{-1/2}$, and $\log_{10} x$, a different technique must be used for sensing the arc tangent argument. A digit 0, 1, 2, or 3 is computed, for arguments in the four intervals 0,1,2,3, respectively. Then by storing the values b and $\tan^{-1} b$ at number times 0-3 on two constant channels, the i register may be used to select the proper values of b and $\tan^{-1} b$.

Based on the theory given above, it is possible to code the functional routines in various ways, depending on the kind of checks to be included and on whether the time or the number of lines of coding for each function is to be minimized. However, the functional coding should always include certain features if the corresponding buttons on the instructional tape preparation table are to be used to refer to the functional subsequences.

For example, when the order, $x_0 \div y_3 = z_6$, is indicated on the instructional tape preparation table, the following lines of coding are recorded:

$$\begin{array}{ll} (n) & x_0 + 0 = r_2 \\ (n+1) & 0 + y_3 = r_0 \\ (n+2) & XT\ (n+4) = r_9 \\ (n+3) & C3000 \\ (n+4) & 0 + r_1 = z_6. \end{array}$$

Thus any coding for the division subsequence should assume the dividend to be in register r_2 and the divisor in r_0. The routine should start on line 3000, and store the quotient in r_1. The line number, n+4, of the command to which the calculator must refer back, should be delivered from r_9 to the line number register. The last command in the routine should be a call on the line number register. When the calculator returns to line n+4, the quotient will be transferred to z_6, and the main course of computation proceeds.

Similarly, when one of the buttons for the functions $x^{-1/2}$, $\log_{10} x$, 10^x, $\cos x$ or $\tan^{-1} x$ is selected to indicate an order $f(x_0) = y_1$, the lines of coding recorded are:

$$\begin{array}{ll} (n) & 0 + x_0 = r_0 \\ (n+1) & XT\ (n+3) = r_9 \\ (n+2) & C3\text{---} \\ (n+3) & 0 + r_1 = y_1. \end{array}$$

TABLE 5.7

Starting Lines for Function Routines

Function	Starting Line
quotient	3000
reciprocal square root	3060
logarithm	3200
exponential	3130
cosine	3264
arc tangent	3340

The starting line numbers, 3---, for the different functions are listed in Table 5.7. To avoid destroying previously computed results, it is advisable to use storage registers on channel r only in the course of functional computations, and to prohibit the use of r registers in problem coding.

The constants required in the functional computations have been recorded as indicated in Appendix I. They are stored with decimal point between columns 15 and 16 so that the computations may be carried out at this decimal point, regardless of the location of the operating decimal point. To simplify the shifting operations necessitated by the two decimal point locations, one between columns 15 and 16 and the other between columns n and $n+1$, it is convenient to store in the calculator the parameters listed in Table 5.8. For example, in the coding given at the end of this chapter it is assumed that these values have been recorded in registers on channel s as part of the set-up procedure for each problem.

TABLE 5.8

Parameters Used in Function Computations

Number	Decimal Point	Storage Register
$15-n$	13/12	s_0
15	$(n+1)/n$	s_3
$30-n$	1/0	s_4
1.4	$(n+1)/n$	s_1
1.0	$(n+1)/n$	s_2
0.5	$(n+1)/n$	s_5
$\log_{10} 2$	$(n+1)/n$	s_6

The quantity $15-n$ is used in all the functional computations to determine the shift required to return a functional value, with decimal point between columns 15 and 16, to the operating decimal point. The second quantity, 15 at the operating decimal point, is added to an argument x before the function 10^x is evaluated. The quantity, $30-n$, is also used in exponential computations to obtain the amount of shift,

$$(15-n) - (m-15) = 30-n-m,$$

required to return the value $10^{0.abc\ldots}$ to the operating decimal point. The quantities 1.4 and 1.0 at the operating decimal point are used in the evaluation of $\tan^{-1} x$, to compute the digit, 0, 1, 2, or 3, indicating in which of four intervals x lies. The logarithm subsequence uses the quantities 0.5 and $\log_{10} 2$ in checking the computation.

The function coding given at the end of this chapter includes a tolerance check on each computed value. If the check fails, the computation is automatically repeated. The calculator stops if the check fails a second time. Division is checked by comparing the dividend with the product of the divisor and the quotient. The reciprocal square root is checked by multiplying the square of the computed value by the original argument, giving a product which should be equal to 1. The relation, $\log_{10} x = \log_{10} x/2 + \log_{10} 2$, is used to check the logarithm. This requires that the logarithm of half the argument be computed and that the constant, $\log_{10} 2$, be stored. The exponential function, 10^x, is

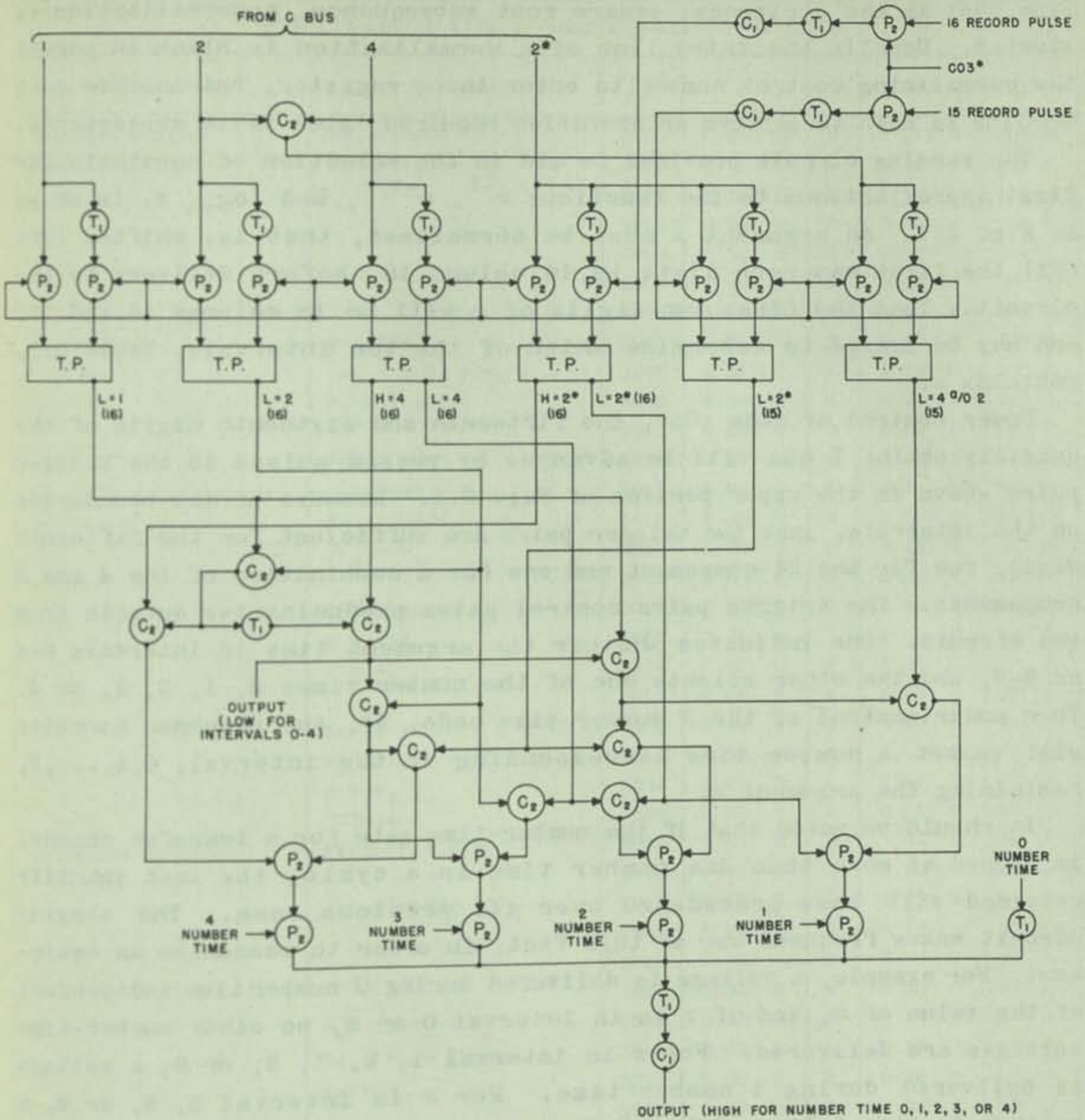


Fig. 5.1—*Function sensing circuit.*

checked by computing the value 10^{-x}. The product, $10^{x} \times 10^{-x}$, should then be equal to 1. The relation, $\sin^2 x + \cos^2 x = 1$, is used to check the cosine. This requires the computation of $\sin x = \cos (\pi/2 - x)$. The arc tangent is checked by means of the relation $\tan^{-1} x + \tan^{-1} 1/x = \pi/2$, which requires the computation of $\tan^{-1} 1/x$.

Since the number of lines of coding reserved for function computations is limited, operations are interposed where necessary. For example, on line 3061 in the reciprocal square root subsequence, a normalization is started. Usually the third line of a normalization is blank to permit the normalizing control number to enter the t register, but in this case the line is used to perform an operation required later in the subsequence.

The sensing circuit provided to aid in the selection of constants for first approximations to the functions x^{-1}, $x^{-1/2}$, and $\log_{10} x$, is shown in Fig. 5.1. An argument x must be normalized, that is, shifted left till the first non-zero digit is in column 16, before delivery to the circuit. Then the first two digits of x will be in columns 16 and 15, and may be sensed to determine which of the ten intervals, Table 5.1, contains x.

Under control of code C03*, the fifteenth and sixteenth digits of the quantity on the C bus will be advanced by record pulses to the trigger pairs shown in the upper portion of Fig. 5.1. Because of the boundaries on the intervals, just two trigger pairs are sufficient for the fifteenth digit, one for the 2* component and one for a combination of the 4 and 2 components. The trigger pairs control gates producing two outputs from the circuit. One indicates whether the argument lies in intervals 0-4 or 5-9, and the other selects one of the number times 0, 1, 2, 3, or 4. Then under control of the B number-time code, 3*, the sequence circuits will select a number time corresponding to the interval, 0,1,...,9, containing the argument x.

It should be noted that if the number-time gate for a transfer channel is opened at more than one number time in a cycle, the last quantity recorded will take precedence over all previous ones. The sensing circuit makes frequent use of this fact, in order to economize on equipment. For example, a voltage is delivered during 0 number time independent of the value of x, and if x is in interval 0 or 6, no other number-time voltages are delivered. For x in interval 1, 2, 7, 8, or 9, a voltage is delivered during 1 number time. For x in interval 2, 8, or 9, a voltage is also delivered during 2 number time, and for x in interval 9, during 3 number time as well.

REFERENCES

1. Lanczos, C., Trigonometric Interpolation of Empirical and Analytical Functions, *Journal of Mathematics and Physics*, XVII, pages 123-199 (1938)

DIVISION CODING WITH AUTOMATIC ROLLBACK AND CHECK						
Expression	Command	Line No.	*A*	OP	*B*	*C*
Rollback control	$XT + \|2000\| = r_8$	3000	0·20 0	0	1 27* 0	30 8
Normalize divisor	$Norm\ \|r_0\| = r_3$	3001	0 07 0	0	1 30 0	00 0
		3002	0 15* 0	1	1 13* 0	15* 0
		3003	0 00 0	0	0 00 0	00 0
		3004	1 30 0	2	0 12* 2*	00 0
		3005	0 00 0	0	0 00 0	00 0
		3006	0 05 0	0	0 00 0	30 3
Sensing	σ = 03*	3007	0 01 0	1	0 17 8	03* 0
	$-a_0 + NCN_1 = r_4$	3008	3 31 0	1	1 13* 0	30 4
a_2x	$r_3 \times$ (-023*)	3009	0 30 3	2	3 02 3*	00 0
		3010	0 00 0	0	0 00 0	00 0
	16/15 *RO*	3011	0 06 0	0	0 00 0	00 0
$a_1 + a_2x$ = 1st approx.	σ + 013* = r_5	3012	0 01 0	1	0 01 3*	30 5
Iteration	$XT + \|3000\| = r_6$	3013	0 30 0	0	1 27* 0	30 6
	$-r_3 \times r_5$	3014	3 30 3	2	0 30 5	00 0
		3015	0 00 0	0	0 00 0	00 0
	16/15 *RO*	3016	0 06 0	0	0 00 0	00 0
	$\sigma + 112$	3017	0 01 0	1	0 11 2	00 0
	$\sigma \times r_5$	3018	0 01 0	2	0 30 5	00 0
		3019	0 00 0	0	0 00 0	00 0
	16/15 *RO* = r_5	3020	0 06 0	0	0 00 0	30 5
	$r_6 - 10^{15} = CC$	3021	0 30 6	1	3 17 5	05* 0
	$\sigma = r_6$	3022	0 01 0	1	0 17 8	30 6
	*CC*3014	3023	0 30 1	4	0 17* 0	00 0
Normalize dividend	$Norm\ \|r_2\| = r_1$	3024	0 07 0	0	1 30 2	00 0
		3025	0 15* 0	1	1 13* 0	15* 0
	$XT + \|0002\| = r_3$	3026	0 00 0	2	1 27* 0	30 3
		3027	1 30 2	2	0 12* 2*	00 0
		3028	0 00 0	0	0 00 0	00 0
		3029	0 05 0	0	0 00 0	30 1
	$r_4 - NCN_2 = r_6$	3030	0 30 4	1	2 13* 0	30 6
	$-\sigma - r_3 = t$	3031	3 01 0	1	3 30 3	15* 0
	$r_1 \times r_5$	3032	0 30 1	2	0 30 5	00 0
		3033	0 00 0	0	0 00 0	00 0
	16/15 *RO* = r_1	3034	0 06 0	0	0 00 0	30 1
	0 + 118	3035	0 15* 0	1	0 11 8	00 0
Shift rounding no.	$\sigma \times 10^t$	3036	0 01 0	2	0 12* 2*	00 0
		3037	0 00 0	0	0 00 0	00 0
	SRO	3038	0 05 0	0	0 00 0	00 0
Rounded quotient	$\sigma + r_1 = r_5$	3039	0 01 0	1	0 30 1	30 5
Shift quotient to $(n+1)/n$	r_5 shifted by $r_6 = r_5$	3040	0 15* 0	1	0 30 6	15* 0
	$r_0 \odot r_2 = r_7$	3041	0 30 0	5	0 30 2	30 7
		3042	0 30 5	2	0 12* 2*	00 0
		3043	0 00 0	0	0 00 0	00 0
		3044	0 05 0	0	0 00 0	30 5

DIVISION CODING WITH AUTOMATIC ROLLBACK AND CHECK (continued)

Expression	Command	Line No.	*A*	OP	*B*	*C*
Attach sign	$r_7 \odot r_5 = r_1$	3045	0 30 7	5	0 30 5	30 1
	$r_9 = LN$	3046	0 30 9	1	0 17 8	04* 0
Check:	$r_1 \times r_0$	3047	0 30 1	2	0 30 0	00 0
		3048	0 00 0	0	0 00 0	00 0
		3049	0 00 0	0	0 00 0	00 0
	$\sigma - r_2$	3050	0 01 0	1	3 30 2	00 0
	$-\lvert\sigma\rvert + 118 = CC$	3051	2 01 0	1	0 11 8	05* 0
		3052	0 00 0	0	0 00 0	00 0
	*CC*0000	3053	0 00 0	0	0 17* 0	00 0
	$r_8 - 10^{15} = CC$	3054	0 30 8	1	3 17 5	05* 0
	$\sigma = r_8$	3055	0 01 0	1	0 17 8	30 8
Rollback	*CC*3001	3056	0 30 0	1	0 17* 0	00 0
	$-s_2$ = stop	3057	0 15* 0	1	3 31 2	16* 0
		3058	0 00 0	0	0 00 0	00 0
		3059	0 00 0	9	0 00 0	00 0

RECIPROCAL SQUARE ROOT CODING WITH AUTOMATIC ROLLBACK

Expression	Command	Line No.	*A*	OP	*B*	*C*
Rollback control	$XT + \lvert 2000 \rvert = r_8$	3060	0 20 0	0	1 27* 0	30 8
Normalize argument	$Norm\ r_0 = r_3$	3061	0 07 0	0	0 30 0	00 0
		3062	0 15* 0	1	1 13* 0	15* 0
	$XT + \lvert 3000 \rvert = r_7$	3063	0 30 0	0	1 27* 0	30 7
		3064	0 30 0	2	0 12* 2*	00 0
		3065	0 00 0	0	0 00 0	00 0
		3066	0 05 0	0	0 00 0	30 3
Sensing	σ = 03*	3067	0 01 0	1	0 17 8	03* 0
$-.5r_3 = r_2$	$-r_3 \times 110$	3068	3 30 3	2	0 11 0	00 0
		3069	0 00 0	0	0 00 0	00 0
	16/15 *RO* $= r_2$	3070	0 06 0	0	0 00 0	30 2
a_3x	$r_3 \times$ -043*	3071	0 30 3	2	3 04 3*	00 0
		3072	0 00 0	0	0 00 0	00 0
	16/15 *RO*	3073	0 06 0	0	0 00 0	00 0
1st approx. $= a_3x + a_1$	σ + 033* $= r_1$	3074	0 01 0	1	0 03 3*	30 1
Iteration	$r_7 - 10^{15} = CC$	3075	0 30 7	1	3 17 5	05* 0
	$\sigma = r_7$	3076	0 01 0	1	0 17 8	30 7
$-.5r_3r_1$	$r_1 \times r_2$	3077	0 30 1	2	0 30 2	00 0
		3078	0 00 0	0	0 00 0	00 0
	16/15 *RO*	3079	0 06 0	0	0 00 0	00 0
$-.5r_3r_1^2$	$\sigma \times r_1$	3080	0 01 0	2	0 30 1	00 0
		3081	0 00 0	0	0 00 0	00 0
	16/15 *RO*	3082	0 06 0	0	0 00 0	00 0

RECIPROCAL SQUARE ROOT CODING WITH AUTOMATIC ROLLBACK (continued)						
Expression	Command	Line No.	A	OP	B	C
$-.5r_3r_1^2 + 1.5$	$\sigma + 111$	3083	0 01 0	1	0 11 1	00 0
$r_1(-.5r_3r_1^2 + 1.5)$	$\sigma \times r_1$	3084	0 01 0	2	0 30 1	00 0
		3085	0 00 0	0	0 00 0	00 0
	16/15 $RO = r_1$	3086	0 06 0	0	0 00 0	30 1
	CC3075	3087	0 30 7	5	0 17* 0	00 0
p	$s_0 - NCN = t$	3088	0 31 0	1	2 13* 0	15* 0
$.5p$	$\sigma \times 117$	3089	0 01 0	2	0 11 7	00 0
		3090	0 00 0	0	0 00 0	00 0
	$HRO = r_5$	3091	0 02 0	0	0 00 0	30 5
Odd even choice	15* 0 - $12t = CC$	3092	0 15* 0	1	3 12 2*	05* 0
$Q = -(15 - n + .5p)$	$-s_0 - r_5 = r_4$	3093	3 31 0	1	3 30 5	30 4
	CC3099	3094	0 30 9	9	0 17* 0	00 0
$Q = -(15 - n + p - 1/2)$	$r_4 + 115 = r_4$	3095	0 30 4	1	0 11 5	30 4
$r_1 \times 10^{-1/2}$	$r_1 \times 113$	3096	0 30 1	2	0 11 3	00 0
		3097	0 00 0	0	0 00 0	00 0
	16/15 $RO = r_1$	3098	0 06 0	0	0 00 0	30 1
	$XT + \|0002\|$	3099	0 00 0	2	1 27* 0	00 0
	$-\sigma - r_4 = t$	3100	3 01 0	1	3 30 4	15* 0
	15* + 118	3101	0 15* 0	1	0 11 8	00 0
Shift rounding no.	$\sigma \times 10^t$	3102	0 01 0	2	0 12* 2*	00 0
		3103	0 00 0	0	0 00 0	00 0
	SRO	3104	0 05 0	0	0 00 0	00 0
Rounded r_1	$\sigma + r_1 = r_6$	3105	0 01 0	1	0 30 1	30 6
Shift to $(n+1)/n$	r_6 shifted by $r_4 = r_1$	3106	0 15* 0	1	0 30 4	15* 0
		3107	0 00 0	0	0 00 0	00 0
		3108	0 30 6	2	0 12* 2*	00 0
		3109	0 00 0	0	0 00 0	00 0
		3110	0 05 0	0	0 00 0	30 1
	$r_9 = LN$	3111	0 30 9	1	0 17 8	04* 0
Check:	$r_1 \times r_1$	3112	0 30 1	2	0 30 1	00 0
		3113	0 00 0	0	0 00 0	00 0
		3114	0 00 0	0	0 00 0	00 0
	$\sigma \times r_0$	3115	0 01 0	2	0 30 0	00 0
		3116	0 00 0	0	0 00 0	00 0
		3117	0 00 0	0	0 00 0	00 0
	$\sigma - s_2$	3118	0 01 0	1	3 31 2	00 0
	$-\|\sigma\| + 118 = CC$	3119	2 01 0	1	0 11 8	05* 0
		3120	0 00 0	0	0 00 0	00 0
	CC0000	3121	0 00 0	0	0 17* 0	00 0
	$r_8 - 10^{15} = CC$	3122	0 30 8	1	3 17 5	05* 0
	$\sigma = r_8$	3123	0 01 0	1	0 17 8	30 8
Rollback	CC3061	3124	0 30 6	1	0 17* 0	00 0
	$-s_2$ = stop	3125	0 15* 0	1	3 31 2	16* 0
		3126	0 00 0	0	0 00 0	00 0
		3127	0 00 0	9	0 00 0	00 0

LOGARITHM CODING WITH AUTOMATIC ROLLBACK AND CHECK

Expression	Command	Line No.	A	OP	B	C
Rollback control	$XT + \lvert 2000 \rvert = r_8$	3200	0 20 0	0	1 27* 0	30 8
	$r_0 = r_6$	3201	0 30 0	1	0 17 8	30 6
Check control	$XT + \lvert 1000 \rvert = r_7$	3202	0 10 0	0	1 27* 0	30 7
Normalize argument	$Norm\ r_6 = r_6$	3203	0 07 0	0	0 30 6	00 0
		3204	0 15* 0	1	1 13* 0	15* 0
	$XT + \lvert 1100 \rvert = r_3$	3205	0 11 0	0	1 27* 0	30 3
		3206	0 30 6	2	0 12* 2*	00 0
		3207	0 00 0	0	0 00 0	00 0
		3208	0 05 0	0	0 00 0	30 6
Sensing	$\sigma = 03*$	3209	0 01 0	1	0 17 8	03* 0
	$-10^{12} = r_5$	3210	0 15* 0	1	3 17 2	30 5
	$r_6 \times 053*$	3211	0 30 6	2	0 05 3*	00 0
		3212	0 00 0	0	0 00 0	00 0
	$16/15\ RO$	3213	0 06 0	0	0 00 0	00 0
$z = r_6/a - 1$	$\sigma - 070 = r_2$	3214	0 01 0	1	3 07 0	30 2
Compute log z by series	$15* + 119$	3215	0 15* 0	1	0 11 9	00 0
	$\sigma \times r_2$	3216	0 01 0	2	0 30 2	00 0
		3217	0 00 0	0	0 00 0	00 0
	$16/15\ RO = r_4$	3218	0 06 0	0	0 00 0	30 4
	$r_5 + 10^{12} = r_5$	3219	0 30 5	1	0 17 2	30 5
	$\sigma = t$	3220	0 01 0	1	0 17 8	15* 0
	$r_3 - 10^{14} = CC$	3221	0 30 3	1	3 17 4	05* 0
	$\sigma = r_3$	3222	0 01 0	1	0 17 8	30 3
	$r_4 + 12t$	3223	0 30 4	1	0 12 2*	00 0
	$CC3216$	3224	0 32 1	6	0 17* 0	00 0
	$\sigma - 12t$	3225	0 01 0	1	3 12 2*	00 0
Mantissa	$\sigma + 063* = r_2$	3226	0 01 0	1	0 06 3*	30 2
	$XT + \lvert 0002 \rvert$	3227	0 00 0	2	1 27* 0	00 0
	$-\sigma + s_0 = t$	3228	3 01 0	1	0 31 0	15* 0
	$15* + 118$	3229	0 15* 0	1	0 11 8	00 0
	$\sigma \times 10^t$	3230	0 01 0	2	0 12* 2*	00 0
		3231	0 00 0	0	0 00 0	00 0
	SRO	3232	0 05 0	0	0 00 0	00 0
Rounded r_2	$\sigma + r_2 = r_2$	3233	0 01 0	1	0 30 2	30 2
Shift mantissa to $(n+1)/n$	r_2 shifted by $-s_0 = r_2$	3234	0 15* 0	1	3 31 0	15* 0
	$s_0 - NCN = r_4$	3235	0 31 0	1	2 13* 0	30 4
		3236	0 30 2	2	0 12* 2*	00 0
		3237	0 00 0	0	0 00 0	00 0
		3238	0 05 0	0	0 00 0	30 2
$-(15-n) + 1.5$	$-s_0 + 116$	3239	3 31 0	1	0 11 6	00 0
$-12 + n$	$\sigma + 116 = t$	3240	0 01 0	1	0 11 6	15* 0
For checking	$r_1 - s_6 = r_5$	3241	0 30 1	1	3 31 6	30 5
Shift characteristic to $(n+1)/n$	$r_4 \times 10^t$	3242	0 30 4	2	0 12* 2*	00 0
		3243	0 00 0	0	0 00 0	00 0

LOGARITHM CODING WITH AUTOMATIC ROLLBACK AND CHECK (continued)						
Expression	Command	Line No.	A	OP	B	C
	SRO	3244	0 05 0	0	0 00 0	00 0
Char. + man. = log	$\sigma + r_2 = r_3$	3245	0 01 0	1	0 30 2	30 3
	$-r_7 + 10^{15} = CC$	3246	3 30 7	1	0 17 5	05* 0
	$\sigma = r_7$	3247	0 01 0	1	0 17 8	30 7
	*CC*3254	3248	0 32 5	4	0 17* 0	00 0
	$r_3 = r_1$	3249	0 30 3	1	0 17 8	30 1
$x/2$	$r_0 \times s_5 = r_6$	3250	0 30 0	2	0 31 5	30 6
		3251	0 00 0	0	0 00 0	00 0
		3252	0 00 0	0	0 00 0	00 0
	*C*3203	3253	0 32 0	3	0 07* 0	00 0
Check logarithm	$r_5 - r_3$	3254	0 30 5	1	3 30 3	00 0
	$-\lvert\sigma\rvert + 118 = CC$	3255	2 01 0	1	0 11 8	05* 0
	$r_9 = LN$	3256	0 30 9	1	0 17 8	04* 0
	*CC*0000	3257	0 00 0	0	0 17* 0	00 0
	$r_8 - 10^{15} = CC$	3258	0 30 8	1	3 17 5	05* 0
	$\sigma = r_8$	3259	0 01 0	1	0 17 8	30 8
Rollback	*CC*3201	3260	0 32 0	1	0 17* 0	00 0
	$-s_2$ = stop	3261	3 31 2	1	0 17 8	16* 0
		3262	0 00 0	0	0 00 0	00 0
		3263	0 00 0	9	0 00 0	00 0

EXPONENTIAL CODING WITH AUTOMATIC ROLLBACK AND CHECK						
Expression	Command	Line No.	A	OP	B	C
Rollback control	$XT + \lvert 2000\rvert = r_8$	3130	0 20 0	0	1 27* 0	30 8
	$r_0 = s_9$	3131	0 30 0	1	0 17 8	31 9
10^x, 10^{-x} control	$XT + \lvert 2000\rvert = r_7$	3132	0 20 0	0	1 27* 0	30 7
Argument + 15	$s_9 + s_3 = s_9$	3133	0 31 9	1	0 31 3	31 9
$(16-n)$ to t reg.	$s_0 + 10^{12} = t$	3134	0 31 0	1	0 17 2	15* 0
	$15^* + 178 = r_6$	3135	0 15* 0	1	0 17 8	30 6
Split integral, fractional parts	$s_9 \times 10^t$	3136	0 31 9	2	0 12* 2*	00 0
		3137	0 00 0	0	0 00 0	00 0
Fraction	$LRO = r_5$	3138	0 03 0	0	0 00 0	30 5
Integer = I	*HRO*	3139	0 02 0	0	0 00 0	00 0
$30-n-I$ (1/0)	$-\sigma + s_4$	3140	3 01 0	1	0 31 4	00 0
	$\sigma \times 10^{12}$	3141	0 01 0	2	0 17 2	00 0
		3142	0 00 0	0	0 00 0	00 0
$30-n-I$ (13/12)	$LRO = r_3$	3143	0 03 0	0	0 00 0	30 3
	$r_5 \times 10^{13}$	3144	0 30 5	2	0 17 3	00 0
		3145	0 00 0	0	0 00 0	00 0
000 *abc*...	$HRO = t$	3146	0 02 0	0	0 00 0	15* 0

EXPONENTIAL CODING WITH AUTOMATIC ROLLBACK AND CHECK (continued)						
Expression	Command	Line No.	*A*	OP	*B*	*C*
	$r_5 \times 10^1$	3147	0 30 5	2	0 16 1	00 0
		3148	0 00 0	0	0 00 0	00 0
$bc\ldots$	$LRO = r_5$	3149	0 03 0	0	0 00 0	30 5
Choose $10^{0.a}$	$15* + 07t = r_4$	3150	0 15* 0	1	0 07 2*	30 4
	$r_5 \times 089$	3151	0 30 5	2	0 08 9	00 0
		3152	0 00 0	0	0 00 0	00 0
$t = (0.0bc\ldots) \log_e 10$	$16/15\ RO = r_2$	3153	0 06 0	0	0 00 0	30 2
Compute series	$XT + \lvert 0700 \rvert = r_5$	3154	0 07 0	0	1 27* 0	30 5
	$15* + 080$	3155	0 15* 0	1	0 08 0	00 0
	$\sigma \times r_2$	3156	0 01 0	2	0 30 2	00 0
		3157	0 00 0	0	0 00 0	00 0
	$16/15\ RO = s_9$	3158	0 06 0	0	0 00 0	31 9
	$r_6 + 10^{12} = t$	3159	0 30 6	1	0 17 2	15* 0
	$\sigma = r_6$	3160	0 01 0	1	0 17 8	30 6
	$-r_5 + 10^{14} = CC$	3161	0 30 5	1	0 17 4	05* 0
	$-\sigma = r_5$	3162	3 01 0	1	0 17 8	30 5
	$CC3166$	3163	0 31 6	6	0 17* 0	00 0
	$s_9 + 08t$	3164	0 31 9	1	0 08 2*	00 0
	$C3156$	3165	0 31 5	6	0 07* 0	00 0
	$s_9 + 070$	3166	0 31 9	1	0 07 0	00 0
	$\sigma \times r_4$	3167	0 01 0	2	0 30 4	00 0
		3168	0 00 0	0	0 00 0	00 0
	$16/15\ RO = r_4$	3169	0 06 0	0	0 00 0	30 4
	$XT + \lvert 0002 \rvert$	3170	0 00 0	2	1 27* 0	00 0
	$-\sigma + r_3 = t$	3171	3 01 0	1	0 30 3	15* 0
	$15* + 118$	3172	0 15* 0	1	0 11 8	00 0
Shift rounding no.	$\sigma \times 10^t$	3173	0 01 0	2	0 12* 2*	00 0
		3174	0 00 0	0	0 00 0	00 0
	SRO	3175	0 05 0	0	0 00 0	00 0
Rounded r_4	$\sigma + r_4 = r_4$	3176	0 01 0	1	0 30 4	30 4
Shift r_4 to $(n+1)/n$	r_4 shifted by $-r_3 = r_2$	3177	0 15* 0	1	3 30 3	15* 0
	$r_7 - 10^{15} = CC$	3178	0 30 7	1	3 17 5	05* 0
		3179	0 30 4	2	0 12* 2*	00 0
		3180	0 00 0	0	0 00 0	00 0
		3181	0 05 0	0	0 00 0	30 2
	$r_7 - 10^{15} = r_7$	3182	0 30 7	1	3 17 5	30 7
	$CC3197$	3183	0 31 9	7	0 17* 0	00 0
Check: $10^x \times 10^{-x}$	$r_1 \times r_2$	3184	0 30 1	2	0 30 2	00 0
		3185	0 00 0	0	0 00 0	00 0
		3186	0 00 0	0	0 00 0	00 0
	$\sigma - s_2$	3187	0 01 0	1	3 31 2	00 0
	$-\lvert\sigma\rvert + 118 = CC$	3188	2 01 0	1	0 11 8	05* 0
	$r_9 = LN$	3189	0 30 9	1	0 17 8	04* 0
	$CC0000$	3190	0 00 0	0	0 17* 0	00 0
	$r_8 - 10^{15} = CC$	3191	0 30 8	1	3 17 5	05* 0

EXPONENTIAL CODING WITH AUTOMATIC ROLLBACK AND CHECK (continued)						
Expression	Command	Line No.	A	OP	B	C
	$\sigma = r_8$	3192	0 01 0	1	0 17 8	30 8
Rollback	CC3131	3193	0 31 3	1	0 17* 0	00 0
	$-s_2$ = stop	3194	3 31 2	1	0 17 8	16* 0
		3195	0 00 0	0	0 00 0	00 0
		3196	0 00 0	9	0 00 0	00 0
Set up to compute 10^{-x}	$r_2 = r_1$	3197	0 30 2	1	0 17 8	30 1
	$-r_0 = s_9$	3198	3 30 0	1	0 17 8	31 9
	C3133	3199	0 31 3	3	0 07* 0	00 0

COSINE CODING WITH AUTOMATIC ROLLBACK AND CHECK						
Expression	Command	Line No.	A	OP	B	C
Rollback control	$XT + \lvert 2000 \rvert = r_8$	3264	0 20 0	0	1 27* 0	30 8
	$r_0 = r_6$	3265	0 30 0	1	0 17 8	30 6
Check control	$XT - \lvert 1000 \rvert = r_7$	3266	0 10 0	0	2 27* 0	30 7
$r_6 \times 10/2\pi = 10J$	$\lvert r_6 \rvert \times 136$	3267	1 30 6	2	0 13 6	00 0
		3268	0 00 0	0	0 00 0	00 0
		3269	0 00 0	0	0 00 0	00 0
$10J \times \pi/20 = J\,\pi/2$	$\sigma \times 137$	3270	0 01 0	2	0 13 7	00 0
		3271	0 00 0	0	0 00 0	00 0
	16/15 RO	3272	0 06 0	0	0 00 0	00 0
$z = J\,\pi/2 - \pi/4$	$\sigma - 138 = r_4$	3273	0 01 0	1	3 13 8	30 4
Compute cos z by series:	$XT + \lvert 0006 \rvert = r_5$	3274	0 00 0	6	1 27* 0	30 5
z^2	$r_4 \times r_4$	3275	0 30 4	2	0 30 4	00 0
		3276	0 00 0	0	0 00 0	00 0
	16/15 $RO = r_2$	3277	0 06 0	0	0 00 0	30 2
	$15^* + 178 = \ell$	3278	0 15* 0	1	0 17 8	15* 0
	$\sigma = r_4$	3279	0 01 0	1	0 17 8	30 4
Iteration	$\sigma + 13\ell = r_3$	3280	0 01 0	1	0 13 2*	30 3
	$r_4 + 10^{12} = r_4$	3281	0 30 4	1	0 17 2	30 4
	$\sigma = \ell$	3282	0 01 0	1	0 17 8	15* 0
	$r_5 - r_4 = CC$	3283	0 30 5	1	3 30 4	05* 0
	$r_3 \times r_2$	3284	0 30 3	2	0 30 2	00 0
		3285	0 00 0	0	0 00 0	00 0
	16/15 RO	3286	0 06 0	0	0 00 0	00 0
	CC3280	3287	0 32 8	0	0 17* 0	00 0
cos z	$\sigma + 10^{15} = r_2$	3288	0 01 0	1	0 17 5	30 2
	$XT + \lvert 0002 \rvert$	3289	0 00 0	2	1 27* 0	00 0
	$-\sigma + s_0 = \ell$	3290	3 01 0	1	0 31 0	15* 0
	$15^* + 118$	3291	0 15* 0	1	0 11 8	00 0
Shift rounding no.	$\sigma \times 10^{\ell}$	3292	0 01 0	2	0 12* 2*	00 0

COSINE CODING WITH AUTOMATIC ROLLBACK AND CHECK (continued)						
Expression	Command	Line No.	*A*	OP	*B*	*C*
		3293	0 00 0	0	0 00 0	00 0
	$SRO = r_3$	3294	0 05 0	0	0 00 0	30 3
$\cos^2 z$	$r_2 \times r_2$	3295	0 30 2	2	0 30 2	00 0
		3296	0 00 0	0	0 00 0	00 0
	$16/15\ RO = r_2$	3297	0 06 0	0	0 00 0	30 2
$\cos^2 z - 1$	$\sigma - 10^{15}$	3298	0 01 0	1	3 17 5	00 0
$\cos 2z = 2\cos^2 z - 1$	$\sigma + r_2 = r_2$	3299	0 01 0	1	0 30 2	30 2
	$-s_0 = t$	3300	3 31 0	1	0 17 8	15* 0
$\cos^2 2z$	$r_2 \times r_2$	3301	0 30 2	2	0 30 2	00 0
		3302	0 00 0	0	0 00 0	00 0
	$16/15\ RO = r_2$	3303	0 06 0	0	0 00 0	30 2
$-\cos^2 2z + 1$	$-\sigma + 10^{15}$	3304	3 01 0	1	0 17 5	00 0
$\cos r_6$	$\sigma - r_2 = r_2$	3305	0 01 0	1	3 30 2	30 2
	$\sigma \odot r_3$	3306	0 01 0	5	0 30 3	00 0
Rounded $\cos r_6$	$\sigma + r_2$	3307	0 01 0	1	0 30 2	00 0
Shift $\cos r_6$ to $(n+1)/n$	$\sigma \times 10^t$	3308	0 01 0	2	0 12* 2*	00 0
		3309	0 00 0	0	0 00 0	00 0
	$SRO = r_3$	3310	0 05 0	0	0 00 0	30 3
	$r_7 + 10^{15} = CC$	3311	0 30 7	1	0 17 5	05* 0
	$\sigma = r_7$	3312	0 01 0	1	0 17 8	30 7
	*CC*3322	3313	0 33 2	2	0 17* 0	00 0
Shift $\pi/2$ to $(n+1)/n$	$-s_0 + 10^{12} = t$	3314	3 31 0	1	0 17 2	15* 0
	15* + 137	3315	0 15* 0	1	0 13 7	00 0
	$\sigma \times 10^t$	3316	0 01 0	2	0 12* 2*	00 0
		3317	0 00 0	0	0 00 0	00 0
	SRO	3318	0 05 0	0	0 00 0	00 0
$\pi/2 - r_0$ = new arg.	$\sigma - r_0 = r_6$	3319	0 01 0	1	3 30 0	30 6
	$r_3 = r_1$	3320	0 30 3	1	0 17 8	30 1
	*C*3267	3321	0 32 6	7	0 07* 0	00 0
	$r_9 = LN$	3322	0 30 9	1	0 17 8	04* 0
$\cos^2 (\pi/2 - x)$	$r_3 \times r_3 = s_9$	3323	0 30 3	2	0 30 3	31 9
		3324	0 00 0	0	0 00 0	00 0
		3325	0 00 0	0	0 00 0	00 0
$\cos^2 x$	$r_1 \times r_1$	3326	0 30 1	2	0 30 1	00 0
		3327	0 00 0	0	0 00 0	00 0
		3328	0 00 0	0	0 00 0	00 0
$\cos^2 x + \cos^2 (\pi/2 - x)$	$\sigma + s_9$	3329	0 01 0	1	0 31 9	00 0
	$\sigma - s_2$	3330	0 01 0	1	3 31 2	00 0
	$-\lvert\sigma\rvert + 118 = CC$	3331	2 01 0	1	0 11 8	05* 0
		3332	0 00 0	0	0 00 0	00 0
	*CC*0000	3333	0 00 0	0	0 17* 0	00 0
	$r_8 - 10^{15} = CC$	3334	0 30 8	1	3 17 5	05* 0
	$\sigma = r_8$	3335	0 01 0	1	0 17 8	30 8
Rollback	*CC*3265	3336	0 32 6	5	0 17* 0	00 0

COSINE CODING WITH AUTOMATIC ROLLBACK AND CHECK (continued)						
Expression	Command	Line No.	A	OP	B	C
	$-s_2$ = stop	3337	3 31 2	1	0 17 8	16* 0
		3338	0 00 0	0	0 00 0	00 0
		3339	0 00 0	9	0 00 0	00 0

ARC TANGENT CODING WITH AUTOMATIC ROLLBACK AND CHECK						
Expression	Command	Line No.	A	OP	B	C
Rollback control	$XT + \lvert 2000 \rvert = r_8$	3340	0 20 0	0	1 27* 0	30 8
	$r_0 = r_2$	3341	0 30 0	1	0 17 8	30 2
Check control	$XT + \lvert 2000 \rvert = r_7$	3342	0 20 0	0	1 27* 0	30 7
	$r_2 = s_9$	3343	0 30 2	1	0 17 8	31 9
Sense to choose b:	$\lvert \sigma \rvert - s_2 = r_5$	3344	1 01 0	1	3 31 2	30 5
	$\sigma \odot s_2$	3345	0 01 0	5	0 31 2	00 0
	$\sigma + s_1$	3346	0 01 0	1	0 31 1	00 0
	$-\sigma + \lvert s_9 \rvert = r_3$	3347	3 01 0	1	1 31 9	30 3
	$r_5 \odot 10^{12}$	3348	0 30 5	5	0 17 2	00 0
	$\sigma + 116 = r_5$	3349	0 01 0	1	0 11 6	30 5
	$r_3 \odot 115$	3350	0 30 3	5	0 11 5	00 0
	$\sigma + r_5 = t$	3351	0 01 0	1	0 30 5	15* 0
	$XT + \lvert 3392 \rvert = s_8$	3352	0 33 9	2	1 27* 0	31 8
b	$15^* + 09t = r_5$	3353	0 15* 0	1	0 09 2*	30 5
$\tan^{-1} b$	$15^* + 10t = r_6$	3354	0 15* 0	1	0 10 2*	30 6
Shift b to $(n+1)/n$	r_5 shifted by $-s_0 = r_4$	3355	0 15* 0	1	3 31 0	15* 0
		3356	0 00 0	0	0 00 0	00 0
		3357	0 30 5	2	0 12* 2*	00 0
		3358	0 00 0	0	0 00 0	00 0
		3359	0 05 0	0	0 00 0	30 4
$b \lvert s_9 \rvert$	$\sigma \times \lvert s_9 \rvert$	3360	0 01 0	2	1 31 9	00 0
		3361	0 00 0	0	0 00 0	00 0
		3362	0 00 0	0	0 00 0	00 0
$b \lvert s_9 \rvert + 1$	$\sigma + s_2 = r_2$	3363	0 01 0	1	0 31 2	30 2
Compute $1/r_2$	$XT + \lvert 3000 \rvert = r_5$	3364	0 30 0	0	1 27* 0	30 5
	$Norm\ r_2 = r_2$	3365	0 07 0	0	0 30 2	00 0
		3366	0 15* 0	1	1 13* 0	15* 0
	$-s_0 + NCN = r_3$	3367	3 31 0	1	1 13* 0	30 3
		3368	0 30 2	2	0 12* 2*	00 0
		3369	0 00 0	0	0 00 0	00 0
		3370	0 05 0	0	0 00 0	30 2
	$\sigma = 03^*$	3371	0 01 0	1	0 17 8	03* 0
	$r_3 - s_0 = t$	3372	0 30 3	1	3 31 0	15* 0
	$r_2 \times -023^*$	3373	0 30 2	2	3 02 3*	00 0

ARC TANGENT CODING WITH AUTOMATIC ROLLBACK AND CHECK (continued)						
Expression	Command	Line No.	A	OP	B	C
		3374	0 00 0	0	0 00 0	00 0
	16/15 RO	3375	0 06 0	0	0 00 0	00 0
1st approx.	$\sigma + 013^* = r_3$	3376	0 01 0	1	0 01 3*	30 3
Iteration	$r_5 - 10^{15} = CC$	3377	0 30 5	1	3 17 5	05* 0
	$\sigma = r_5$	3378	0 01 0	1	0 17 8	30 5
	$r_3 \times -r_2$	3379	0 30 3	2	3 30 2	00 0
		3380	0 00 0	0	0 00 0	00 0
	16/15 RO	3381	0 06 0	0	0 00 0	00 0
	$\sigma + 112$	3382	0 01 0	1	0 11 2	00 0
	$\sigma \times r_3$	3383	0 01 0	2	0 30 3	00 0
		3384	0 00 0	0	0 00 0	00 0
	16/15 $RO = r_3$	3385	0 06 0	0	0 00 0	30 3
	CC3377	3386	0 33 7	7	0 17* 0	00 0
	$s_8 = LN$	3387	0 31 8	1	0 17 8	04* 0
Shift r_3 to $(n+1)/n$	$r_3 \times 10^t$	3388	0 30 3	2	0 12* 2*	00 0
		3389	0 00 0	0	0 00 0	00 0
	$SRO = r_2$	3390	0 05 0	0	0 00 0	30 2
	C0000	3391	0 00 0	0	0 07* 0	00 0
$\|s_9\| - b$	$\|s_9\| - r_4$	3392	1 31 9	1	3 30 4	00 0
$(\|s_9\| - b)/(b\|s_9\| + 1)$	$\sigma \times r_2 = r_3$	3393	0 01 0	2	0 30 2	30 3
		3394	0 00 0	0	0 00 0	00 0
		3395	0 00 0	0	0 00 0	00 0
Shift to 16/15	r_3 shifted by $s_0 = r_3$	3396	0 15* 0	1	0 31 0	15* 0
	$XT + \|0004\| = r_4$	3397	0 00 0	4	1 27* 0	30 4
		3398	0 30 3	2	0 12* 2*	00 0
		3399	0 00 0	0	0 00 0	00 0
		3400	0 05 0	0	0 00 0	30 3
Compute $\tan^{-1} r_3$ by series	$r_4 = t$	3401	0 30 4	1	0 17 8	15* 0
	$r_3 \times r_3$	3402	0 30 3	2	0 30 3	00 0
		3403	0 00 0	0	0 00 0	00 0
	16/15 $RO = r_5$	3404	0 06 0	0	0 00 0	30 5
	$15^* + 178$	3405	0 15* 0	1	0 17 8	00 0
Iteration	$\sigma + 09t = r_2$	3406	0 01 0	1	0 09 2*	30 2
	$-r_4 + 083 = CC$	3407	3 30 4	1	0 08 3	05* 0
	$r_4 + 10^{12} = r_4$	3408	0 30 4	1	0 17 2	30 4
	$\sigma = t$	3409	0 01 0	1	0 17 8	15* 0
	$r_2 \times r_5$	3410	0 30 2	2	0 30 5	00 0
		3411	0 00 0	0	0 00 0	00 0
	16/15 RO	3412	0 06 0	0	0 00 0	00 0
	CC3406	3413	0 34 0	6	0 17* 0	00 0
	$\sigma + 104$	3414	0 01 0	1	0 10 4	00 0
	$\sigma \times r_3$	3415	0 01 0	2	0 30 3	00 0
		3416	0 00 0	0	0 00 0	00 0
	16/15 RO	3417	0 06 0	0	0 00 0	00 0
$\tan^{-1} r_3 + \tan^{-1} b$	$\sigma + r_6 = r_5$	3418	0 01 0	1	0 30 6	30 5

ARC TANGENT CODING WITH AUTOMATIC ROLLBACK AND CHECK (continued)						
Expression	Command	Line No.	A	OP	B	C
	$XT + \lvert 0002 \rvert$	3419	0 00 0	2	1 27* 0	00 0
	$-\sigma + s_0 = t$	3420	3 01 0	1	0 31 0	15* 0
	15* + 118	3421	0 15* 0	1	0 11 8	00 0
Shift rounding no.	$\sigma \times 10^t$	3422	0 01 0	2	0 12* 2*	00 0
		3423	0 00 0	0	0 00 0	00 0
	SRO	3424	0 05 0	0	0 00 0	00 0
Rounded $\tan^{-1} s_9$	$\sigma + r_5 = r_3$	3425	0 01 0	1	0 30 5	30 3
	$r_7 - 10^{15} = CC$	3426	0 30 7	1	3 17 5	05* 0
Shift to $(n+1)/n$	r_3 shifted by $-s_0 = r_3$	3427	0 15* 0	1	3 31 0	15* 0
		3428	0 00 0	0	0 00 0	00 0
		3429	0 30 3	2	0 12* 2*	00 0
		3430	0 00 0	0	0 00 0	00 0
		3431	0 05 0	0	0 00 0	30 3
	$r_9 - 10^{15} = r_7$	3432	0 30 7	1	3 17 5	30 7
	$r_0 \odot r_3 = r_5$	3433	0 30 0	5	0 30 3	30 5
	*CC*3451	3434	0 34 5	1	0 17* 0	00 0
Check: $\tan^{-1} x + \tan^{-1} 1/x$	$r_1 + r_5 = r_3$	3435	0 30 1	1	0 30 5	30 3
	$-s_0 + 10^{12} = t$	3436	3 31 0	1	0 17 2	15* 0
	$-r_0 \odot 137$	3437	3 30 0	5	0 13 7	00 0
$\pm\pi/2$ at $(n+1)/n$	$\sigma \times 10^t$	3438	0 01 0	2	0 12* 2*	00 0
		3439	0 00 0	0	0 00 0	00 0
	SRO	3440	0 05 0	0	0 00 0	00 0
	$\sigma + r_3$	3441	0 01 0	1	0 30 3	00 0
	$-\lvert\sigma\rvert + 118 = CC$	3442	2 01 0	1	0 11 8	05* 0
	$r_9 = LN$	3443	0 30 9	1	0 17 8	04* 0
	*CC*0000	3444	0 00 0	0	0 17* 0	00 0
	$r_8 - 10^{15} = CC$	3445	0 30 8	1	3 17 5	05* 0
	$\sigma = r_8$	3446	0 01 0	1	0 17 8	30 8
	*CC*3341	3447	0 33 4	1	0 17* 0	00 0
	$-s_2$ = stop	3448	3 31 2	1	0 17 8	16* 0
		3449	0 00 0	0	0 00 0	00 0
		3450	0 00 0	9	0 00 0	00 0
Set up for $\tan^{-1} 1/x$	$r_5 = r_1$	3451	0 30 5	1	0 17 8	30 1
	$XT + \lvert 3343 \rvert = s_8$	3452	0 33 4	3	1 27* 0	31 8
	$r_0 = r_2$	3453	0 30 0	1	0 17 8	30 2
	*C*3364	3454	0 33 6	4	0 07* 0	00 0

CHAPTER VI

NUMERICAL INPUT AND OUTPUT DEVICES

The three subjects to be dealt with in this chapter are the numerical tape preparation table, the input-output system of the calculator, and the five printers. All of these units are associated with the storage of numerical information on magnetic tape.

The mechanical devices used in these three units to move the magnetic tape past the pole piece tips are similar in many respects, differing largely in the speeds at which the tapes are driven. The tape is threaded through a system of rollers which maintain proper tension in the tape, and is passed over a driving capstan. A rubber belt presses the tape against the capstan in order to provide adequate friction. The capstan is started and stopped by a magnetic clutch of the type shown in Plate XX. While all of the tape mechanisms use a friction-driven take-up reel, it is necessary to drive the supply reel in the case of the input-output mechanisms only. A detailed description of the system controlling this drive is presented later on in this chapter.

Four pole pieces are mounted on a block in a staggered arrangement so that each covers a separate channel on the tape. These four channels on the tape are designated by the letters, *A*, *B*, *C*, and *D*. Channels *A* and *B* are reserved for coded numerical data, while channels *C* and *D* are used for timing pulses. The eighty binary digits associated with one quantity in the calculator, Table 6.1, are arranged on the tape as shown in Fig. 6.1, and have the relative timing shown in Fig. 6.2. These diagrams indicate that in contrast to the system used within the calculator, the high-order digits are recorded first on the tape to simplify the operation of the printers. For check purposes, the information is recorded in duplicate on the *A* and *B* channels, and corresponding components are one digit out of phase. One of the eighty binary digits is omitted from each information channel on the tape, as this simplifies the control circuits. However, the omitted digits are components of noninformative decimal digits, namely, the 2* component of digit 18 and the 1 component of digit -1.

In general, techniques similar to those used with the magnetic drums are employed in tape recording and playback. A reel of tape is erased, prior to recording, on a commercially available eraser, since no provisions have been made for selective alteration of previously recorded information.

The simplest of the tape-handling units is the manually operated

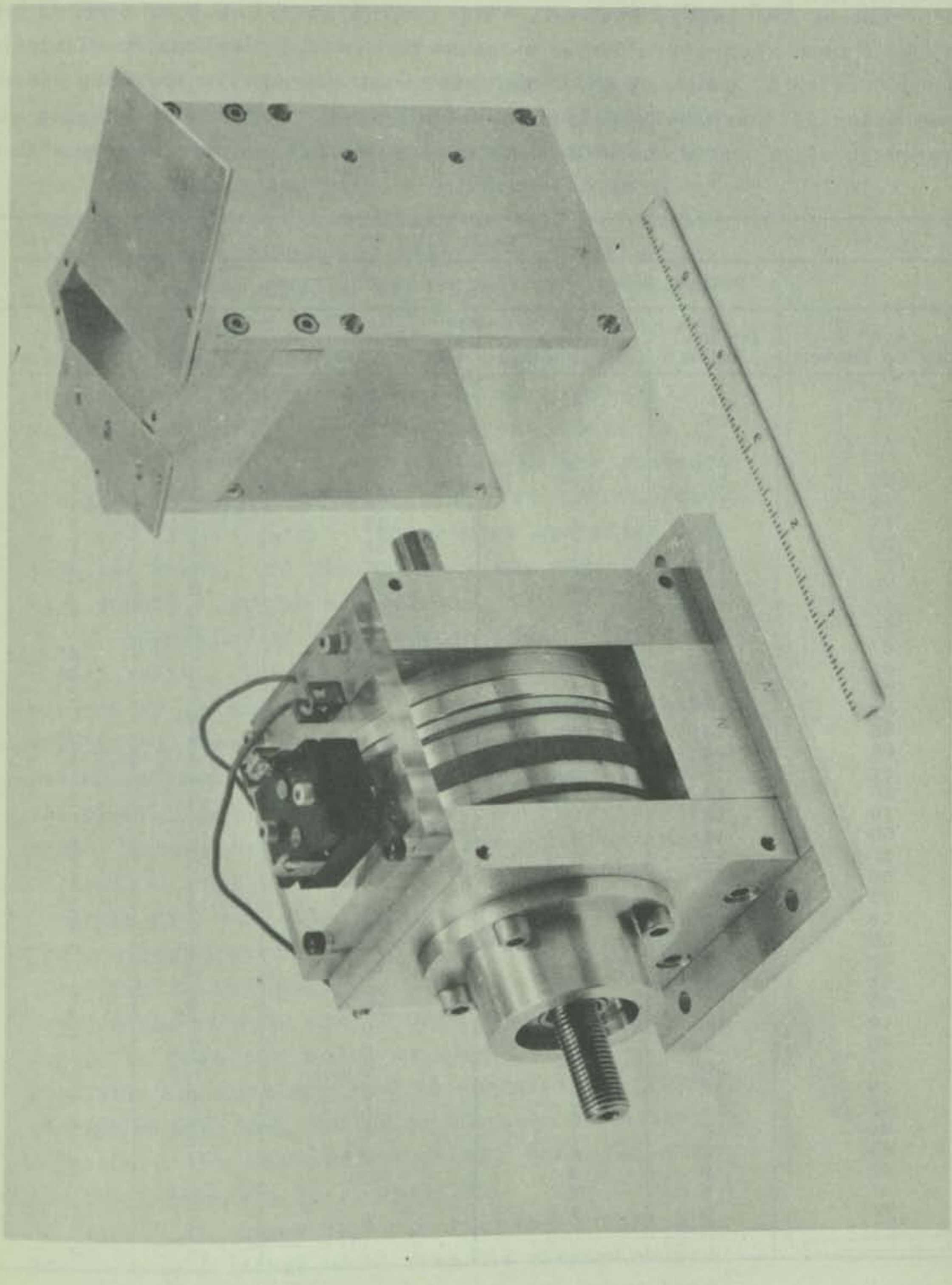


Plate XX *Magnetic Clutch*

numerical tape preparation table. The sixteen decimal digits and algebraic sign of each quantity to be recorded must be registered on keys at the front of the table, Plate X. The switch at the top of each column may be thrown when zeros occur repeatedly in that column to eliminate the necessity of punching the 0 key with each quantity. When the record pushbutton is depressed, relays are energized to store the digits and algebraic sign, and a check is made to insure that in each column either

TABLE 6.1

Pulses associated with one quantity on tape

A or *B* Pulse Number	Decimal Digit	Binary Component	*A* or *B* Pulse Number	Decimal Digit	Binary Component
79[1]	18	2*	39	8	2*
78	18	4	38	8	4
77	18	2	37	8	2
76	18	1	36	8	1
75	17	2*	35	7	2*
74	17	4	34	7	4
73	17	2	33	7	2
72	17	1	32	7	1
71	16	2*	31	6	2*
70	16	4	30	6	4
69	16	2	29	6	2
68	16	1	28	6	1
67	15	2*	27	5	2*
66	15	4	26	5	4
65	15	2	25	5	2
64	15	1	24	5	1
63	14	2*	23	4	2*
62	14	4	22	4	4
61	14	2	21	4	2
60	14	1	20	4	1
59	13	2*	19	3	2*
58	13	4	18	3	4
57	13	2	17	3	2
56	13	1	16	3	1
55	12	2*	15	2	2*
54	12	4	14	2	4
53	12	2	13	2	2
52	12	1	12	2	1
51	11	2*	11	1	2*
50	11	4	10	1	4
49	11	2	9	1	2
48	11	1	8	1	1
47	10	2*	7	0	2*
46	10	4	6	0	4
45	10	2	5	0	2
44	10	1	4	0	1
43	9	2*	3	-1	2*
42	9	4	2	-1	4
41	9	2	1	-1	2
40	9	1	0	-1	1
			-1[2]	18	1

[1] Channel *B* only

[2] Channel *A* only

a key was punched or the zero switch thrown. A check failure causes an indicator lamp to be lit. If the check passes, the keys are reset and the operator must punch the same number a second time. When the record pushbutton is depressed again, the same columnar check is made and in addition the quantity stored in relays is compared with the one indicated on the keyboard. Failure of this check causes a second indicator lamp to be lit. In the event that the first quantity registered on the keyboard was in error, the reset switch may be employed to reset the keys, storage relays, and control circuits. If the error was in the second quantity, the corrections may be made immediately on the keyboard, where the second quantity is still registered. When both checks are passed, the quantity stored in relays will be recorded on tape. At the end of the record operation the keys are reset, and the operator may register the next quantity on the keyboard.

The operation of the table is under control of a step switch, labeled #1 in Fig. 6.3. The same diagram shows the timing for the six cams which supply the necessary pulses to control the step switch and the associated relays. Table 6.2 lists the operations performed in one step switch cycle, during which one quantity is recorded on tape.

Each of the keys on the front of the table controls five contacts, as indicated in Fig. 6.4. The contacts labeled C, along with the switches (Sw.1,...,Sw.16) which may be thrown when zeros occur repeatedly in one or more columns, are connected to form the columnar check circuit. The remaining contacts are used to energize the proper storage relays, and, following the second punching, to perform the comparison check, controlled by the check relays, $A1,\ldots,A11$.

Figure 6.5 shows the electronic circuit and the two step switches which scan the storage relays and deliver digits to the record circuit. Accurately timed record and cycling pulses are derived from

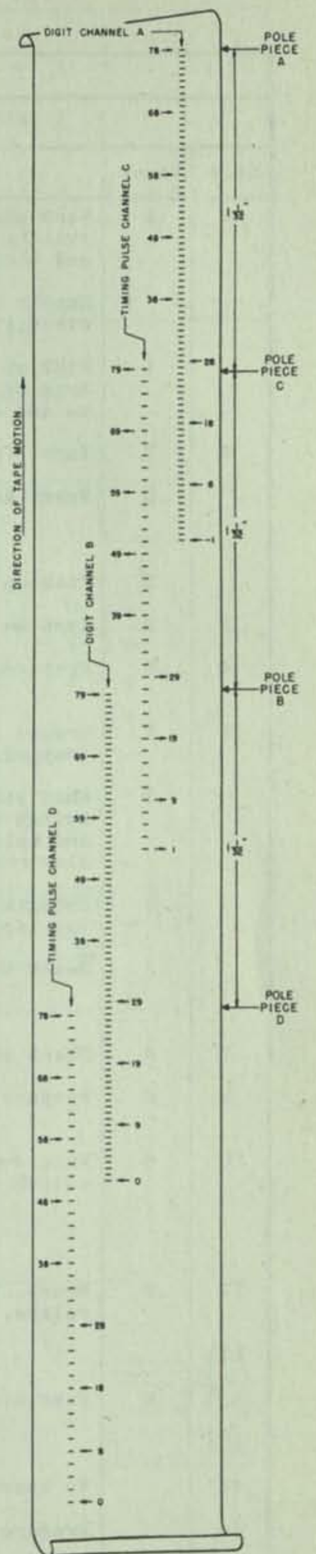


Fig. 6.1—*Location of pulses on tape.*

TABLE 6.2

Step switch 1 operation (tape preparation table)

Step	Wiper	Operation	Relays
1	2	When start button is depressed, start the step switch, provided that column-punch check has passed and that electronic circuits are in reset position.	Pick up CL101,CL102; energize step switch coil.
		Supply current to recording tubes in electronic circuit.	Pick up CL114,CL206.
	1	Pick up storage relays corresponding to depressed keys by delivering -120 v. to storage relay pick-up and check bus.	
2	7	Turn off reset position indicator lamp.	
	5	Reset keys.	Pick up CH112,CL111; energize key reset magnet.
	3	Pick up check relays, *A*1-*A*11.	
3	3	Turn on light 1.	Pick up CL109,CH110.
4	5	Prepare to stop on step 5.	Pick up CL107,CL108. Drop out CL102.
5	1	Ground storage relay pick-up and check bus for comparison check.	
	2	When start button is depressed, start the step switch again, provided that the comparison check and column-punch check have passed and that the electronic circuit is in the reset position.	Pick up CL101,CL102; energize step switch coil.
	3	Energize start coil for magnetic clutch controlling the tape.	Pick up CL103,CH104.
	5	Reset keys.	Pick up CL111,CH112; energize key reset magnet.
7	4	Start electronic circuit.	
9	5	Prepare to stop on step 10.	Pick up CL107,CL108. Drop out CL102.
10	6	When recording has been completed, start step switch again.	CL203 picked up by electronic circuit. Pick up CL101,CL102; energize step switch coil.
12	3	Reset, by dropping relay hold line and check relays. Turn off light 1.	Pick up CL105,CL106.
13 . . 18	8	Pass directly over these steps to step 19.	
19		No special operations.	
20		Prepare to stop on step 1.	Pick up CL107,CL108. Drop out CL102.

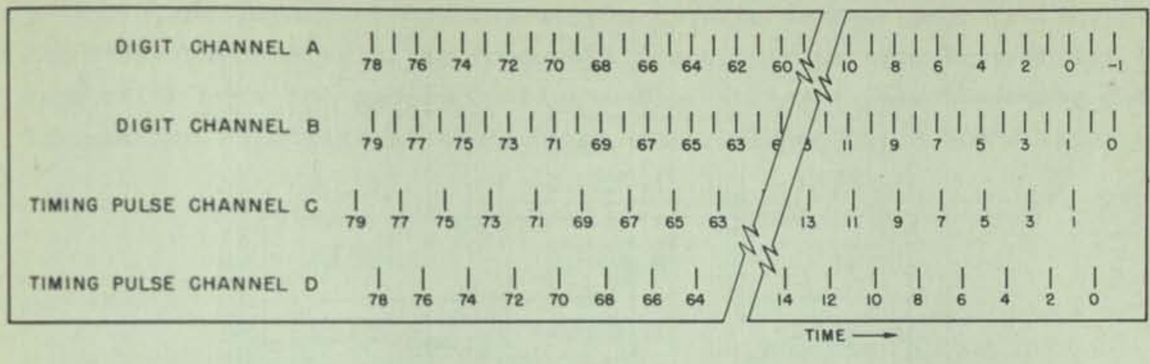


Fig. 6.2—*Relative timing of pulses on tape.*

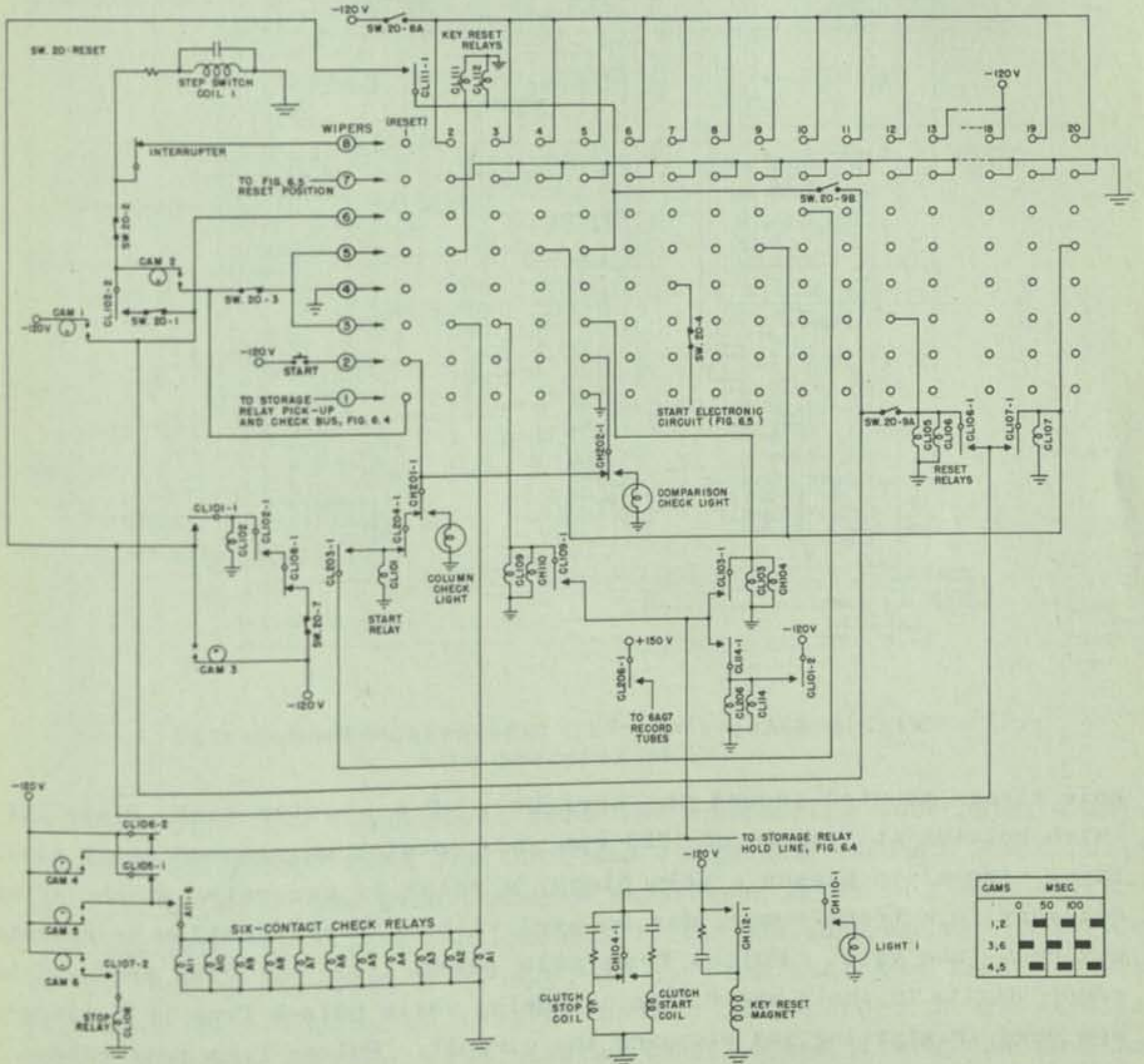


Fig. 6.3—*Step switch 1, tape preparation table.*

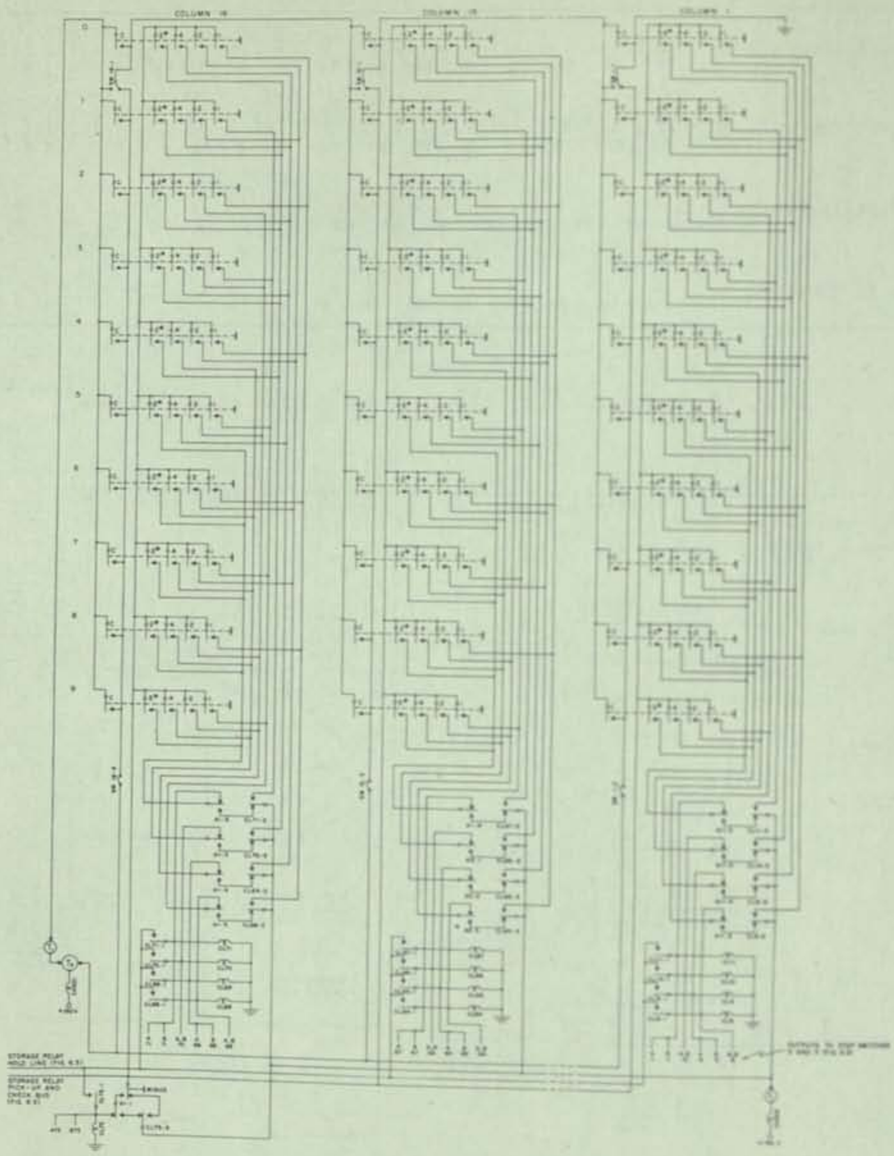


Fig. 6.4—*Key contacts, tape preparation table.*

pole pieces mounted around the periphery of a plastic disk, Plate XXI, which rotates at a speed of 1725 rpm and contains magnetized metal pins. Every time a pin passes a pole piece, a pulse is generated which, after delivery to a transformer, may be used without amplification to operate a vacuum-tube gate. Pulses from pole piece 1 on the disk are used to record digits in the *A* and *B* tape channels, while pulses from pole piece 2 are used in starting and stopping the circuit. Pulses from pole pieces 3 and 4 are referred to as *C* and *D* pulses, respectively, since they are used to record in the *C* and *D* tape channels as well as for timing signals.

The start impulse from step switch 1 trips trigger pair 1, Fig. 6.5. The first succeeding pulse from pole piece 2 trips trigger pair 2. The connection from the non-conducting tube in trigger pair 2 permits gated information to arrive at the record circuits for the *A* and *B* tape pole

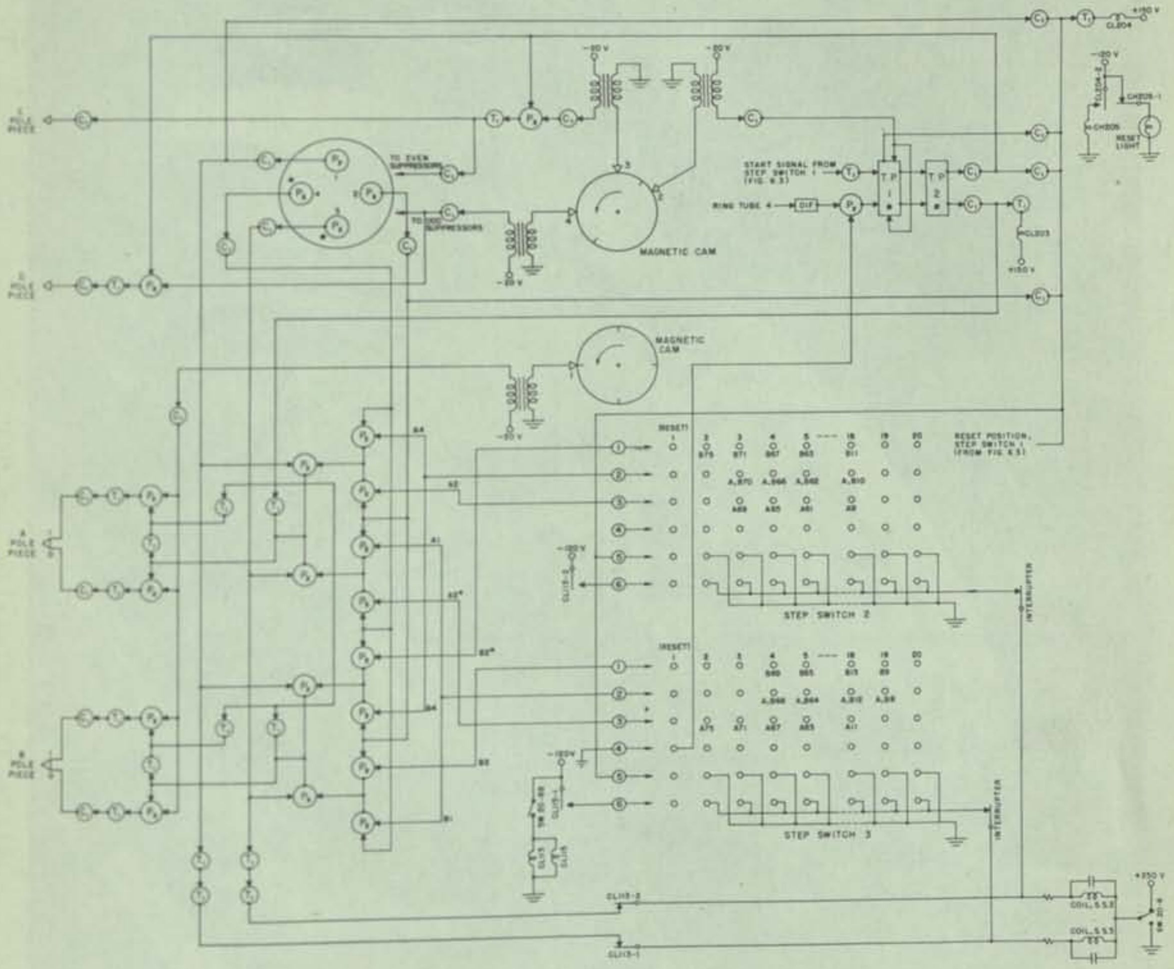


Fig. 6.5—*Step switches 2, 3, and electronic circuits, tape preparation table.*

pieces, while the connection from the conducting tube gates *C* and *D* pulses to the record tubes for the *C* and *D* tape pole pieces. The *C* pulses are also gated to the even-numbered suppressor grids in a ring of four, *D* pulses being delivered continuously to the odd suppressors. Referring to the timing diagram, Fig. 6.6, it is seen that the first *C* pulse following the tripping of trigger pair 2 is recorded on the tape and starts the ring cycling. Two step switches, 2 and 3, are then advanced alternately by the ring, to deliver the *A* and *B* digits from storage relays to pentode gates. The ring also controls these gates, causing

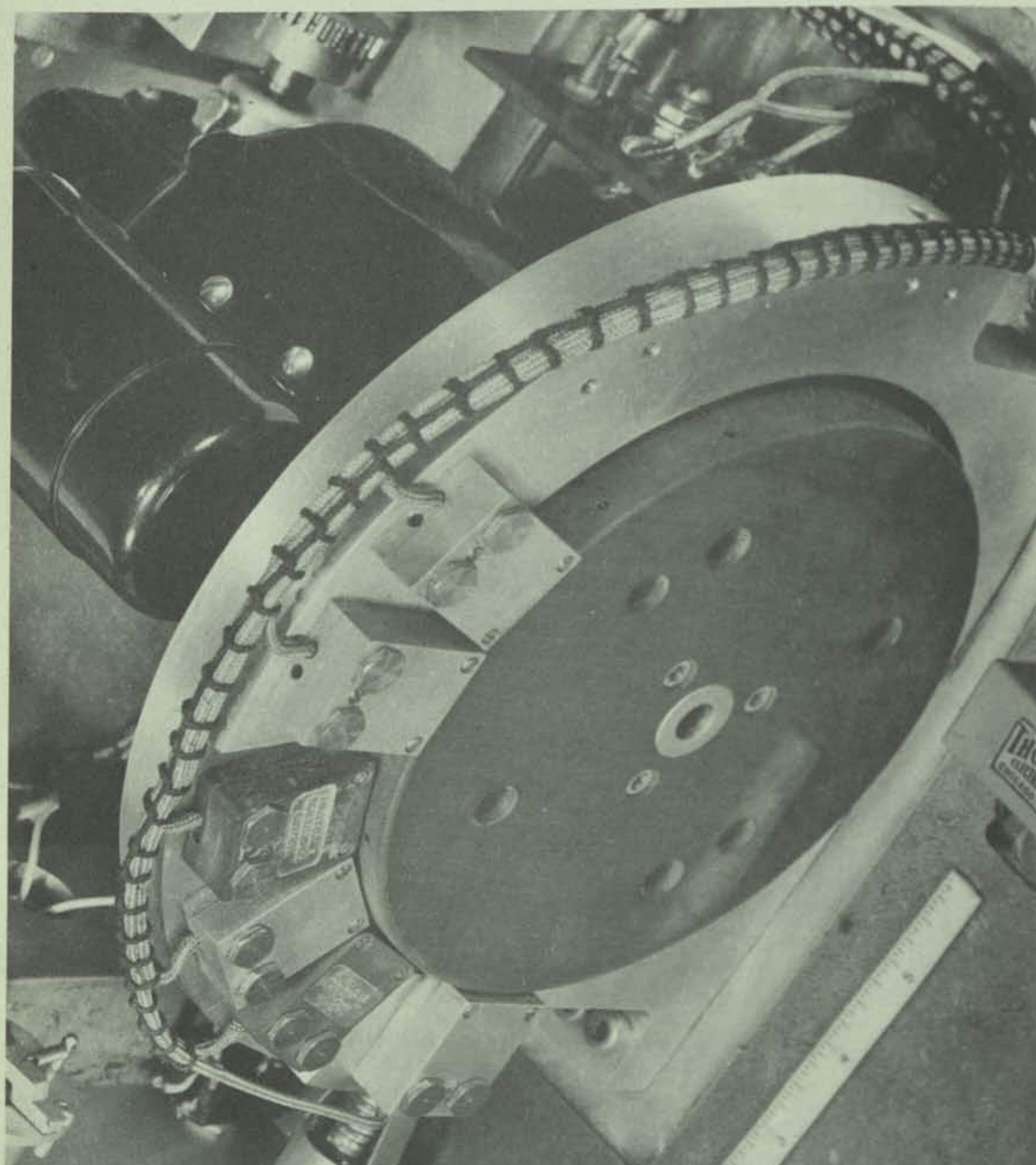


Plate XXI Magnetic Cam Unit

one *A* and one *B* digit to be delivered to the *A* and *B* record circuits at each position of the ring.

When step switch 3 returns to its reset position, a ground potential is delivered through wiper 4 to open the gate serving as a puller tube for trigger pair 1. When the last *D* pulse to be recorded advances the ring to position 4, a differentiated pulse is produced which resets trigger pair 1. The following pulse from pole piece 2 resets trigger pair 2, closing the pulse gates in the electronic circuit. At the same time, the trigger pair 2 output energizes relay CL203, to start step switch 1 again.

The reset switch (Sw.20) resets all the circuits associated with the tape preparation table, including the keys on the front. The reset

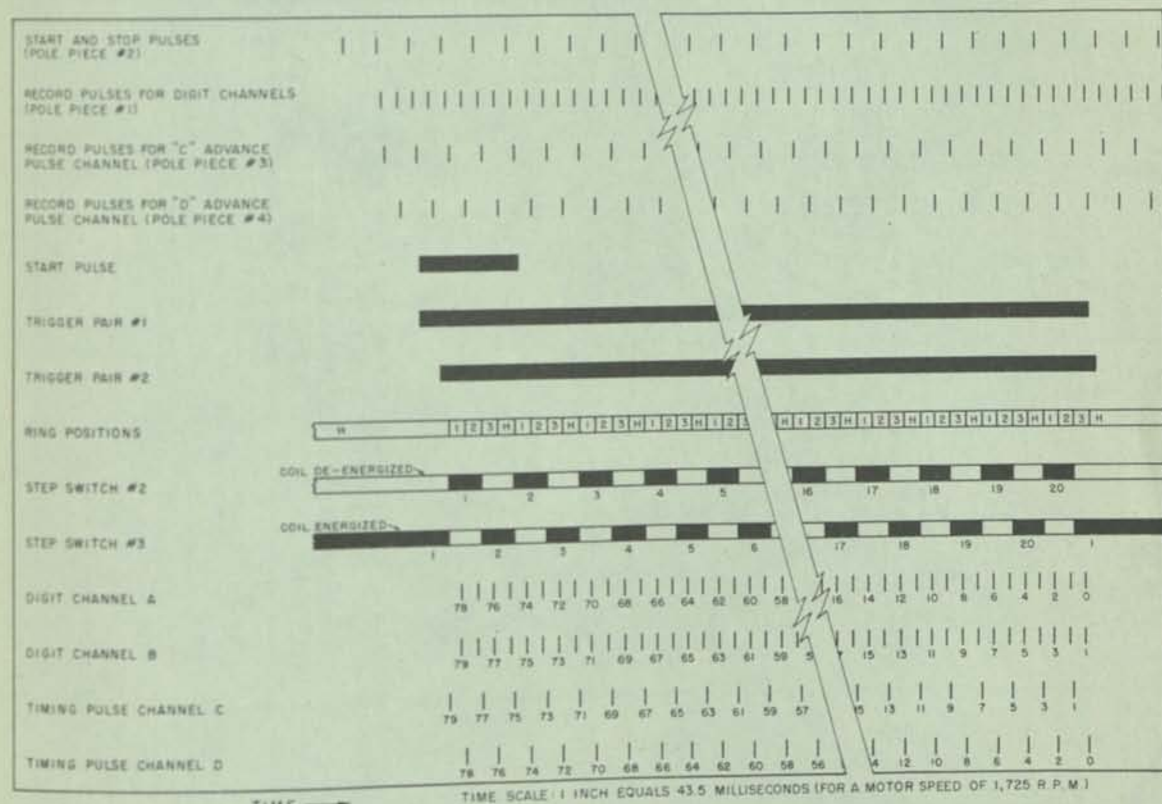


Fig. 6.6—*Timing diagram, tape preparation table.*

light, Fig. 6.5, is turned on under control of relay CL204, when the ring of four, trigger pairs 1 and 2, and the three step switches are in their reset positions.

The next units to be described are the eight tape read-record mechanisms. Two of these mechanisms, Plates XXII and XXIII, with associated

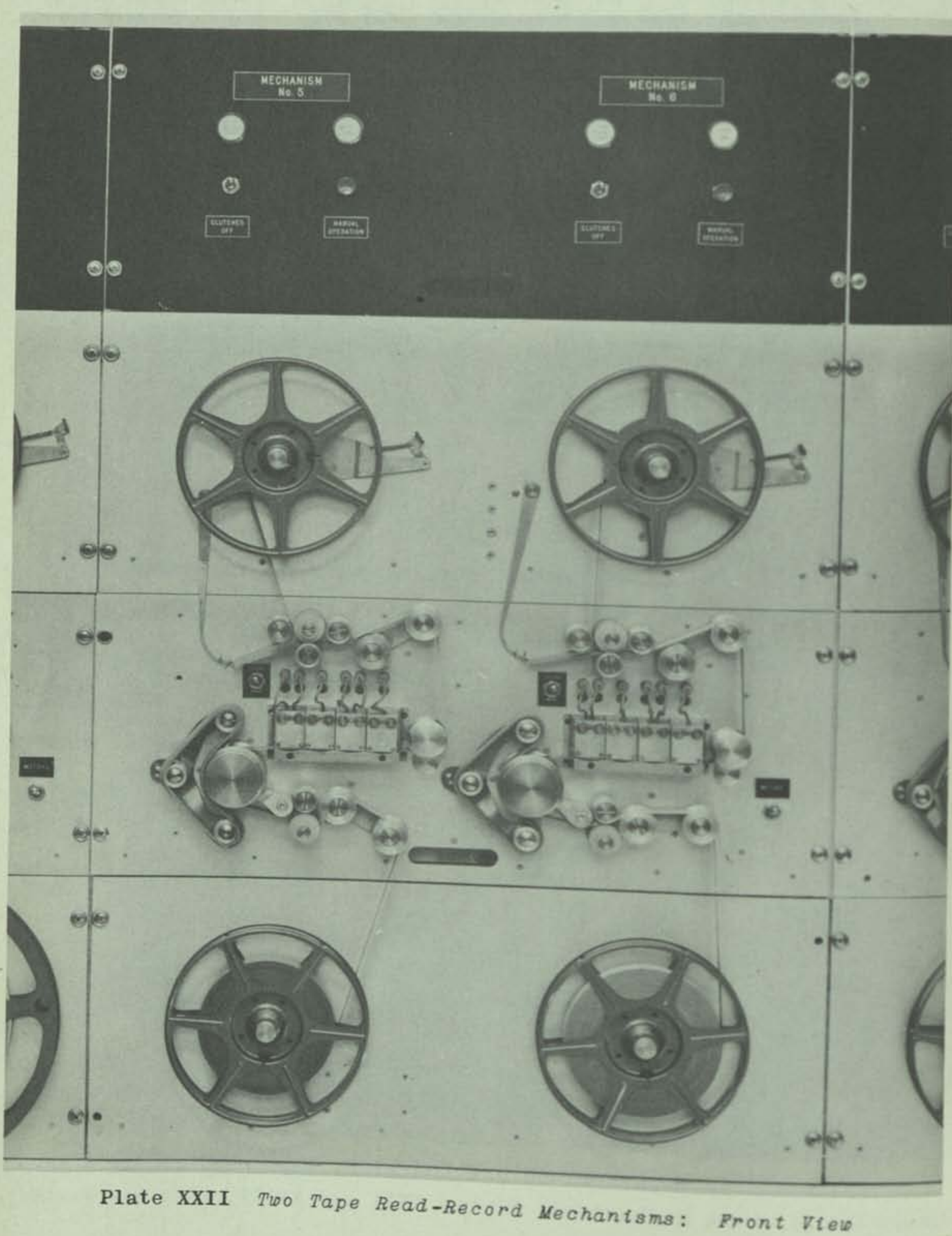


Plate XXII *Two Tape Read-Record Mechanisms: Front View*

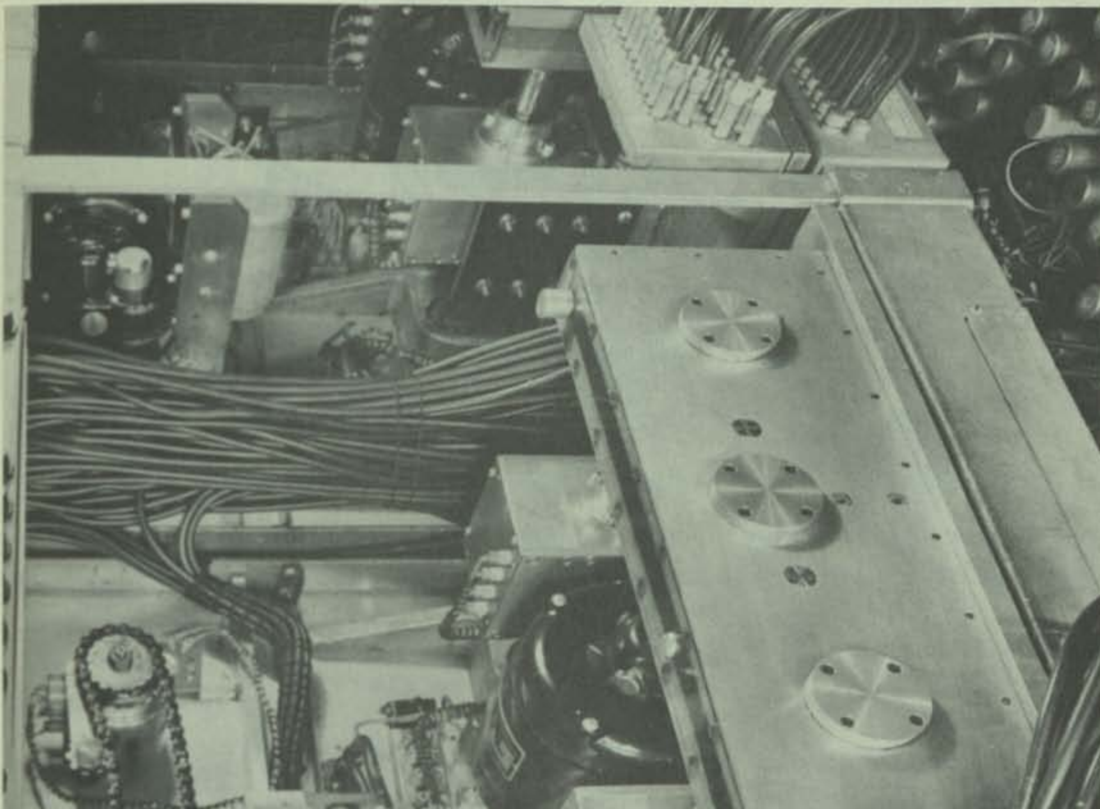


Plate XXIII *Two Tape Read-Record Mechanisms: Rear View*

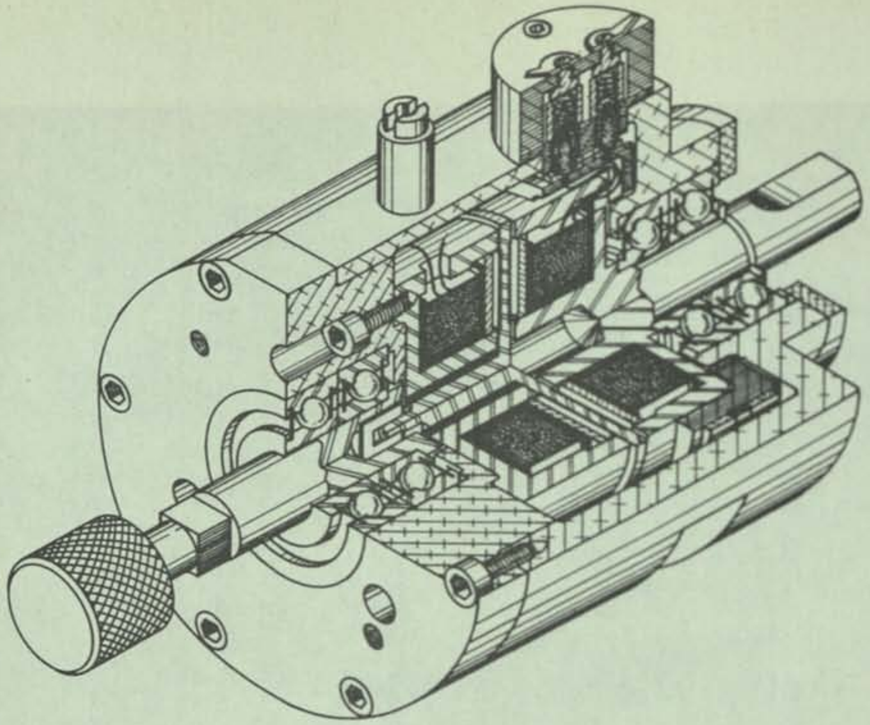
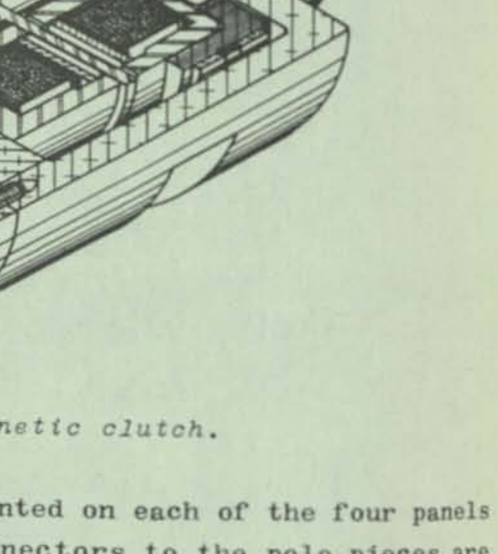


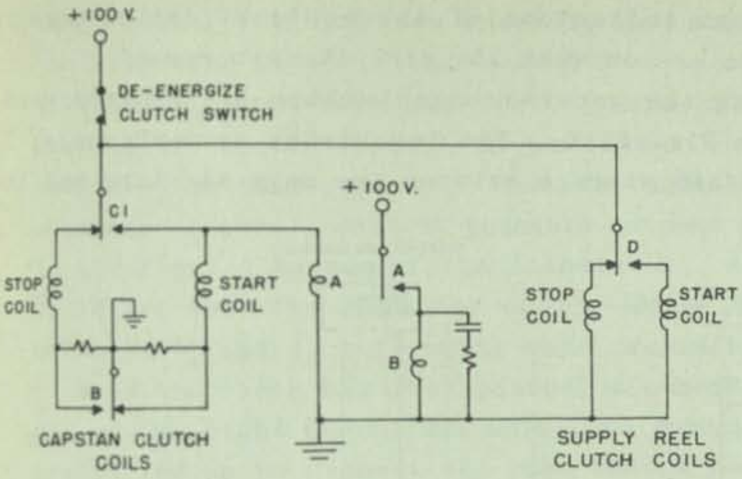
Fig. 6.7—*Tape reel magnetic clutch.*

switches, pushbuttons, and lights are mounted on each of the four panels at the right of the calculator. The connectors to the pole pieces are plugged to the upper or lower plughubs for record or read operations, respectively. The "read-record" switch must be set correspondingly. Tape is moved past the pole piece tips by a clutch-driven capstan at a speed of about 28 inches per second. A second clutch, Fig. 6.7, controlled by the circuit shown in Fig. 6.8, starts and stops the tape supply reel. When the main clutch starts the tape moving, the arm shown in Fig. 6.8(*a*) is rotated counterclockwise by the tension on the tape. In this position a contact is made to pick up relay *D*, operating the clutch for the tape supply reel, as shown in Fig. 6.8(*b*).



(*a*)

Fig. 6.8—*Clutch control circuit for one of eight mechanisms.*



(*b*)

Fig. 6.8 (continued)

Insufficient tape or broken tape in any of the eight mechanisms will cause an alarm to be sounded, and the calculator to stop. Spring contacts sense for these conditions.

The "motor" switch on each of the four panels controls three motors, which drive the tape supply and take-up reels as well as the capstans for both mechanisms (Fig. 6.9). The "clutches off" switch associated with each mechanism de-energizes both the start and stop coils of the capstan clutch, to facilitate threading the tape. The "manual operation" pushbutton transfers one quantity from the tape to the intermediate drum storage channels associated with the tape read-record system, or vice versa, depending on whether the mechanism is set for reading or recording. This makes it possible to position an input tape at the start of a problem.

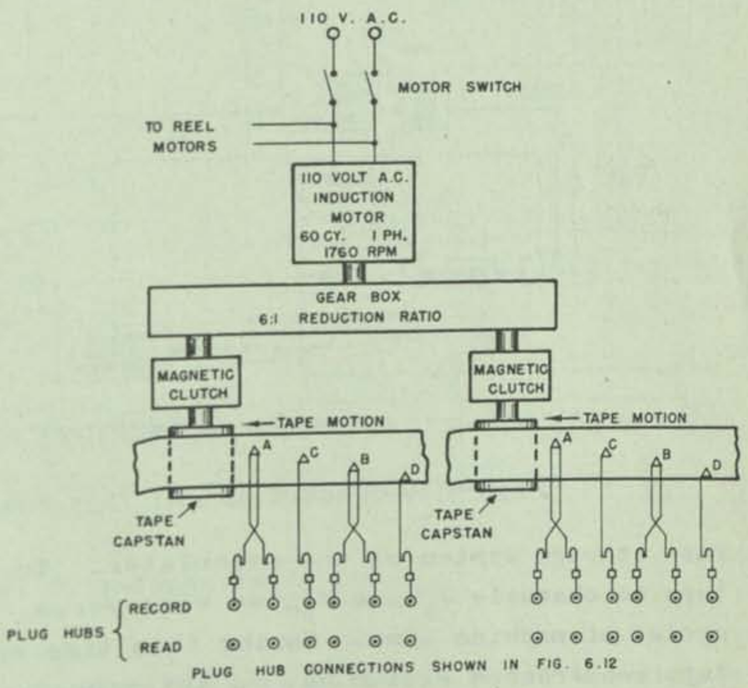


Fig. 6.9—*Two tape read-record mechanisms.*

Two lights are provided with each mechanism. The left-hand light indicates that the associated mechanism is in use, either reading or recording. The right-hand light goes on after the mechanism has been used, and remains on until another tape read or record operation is initiated. Four additional lights located beneath the input-output mechanisms provide a visual indication of the state of the electronic circuits. All four lights are on when the circuits are reset.

A block diagram showing the general organization of the tape read-record system is given in Fig. 6.10. The two binary drum channels, I_a and I_b, serve as intermediate storage between the magnetic tape and the

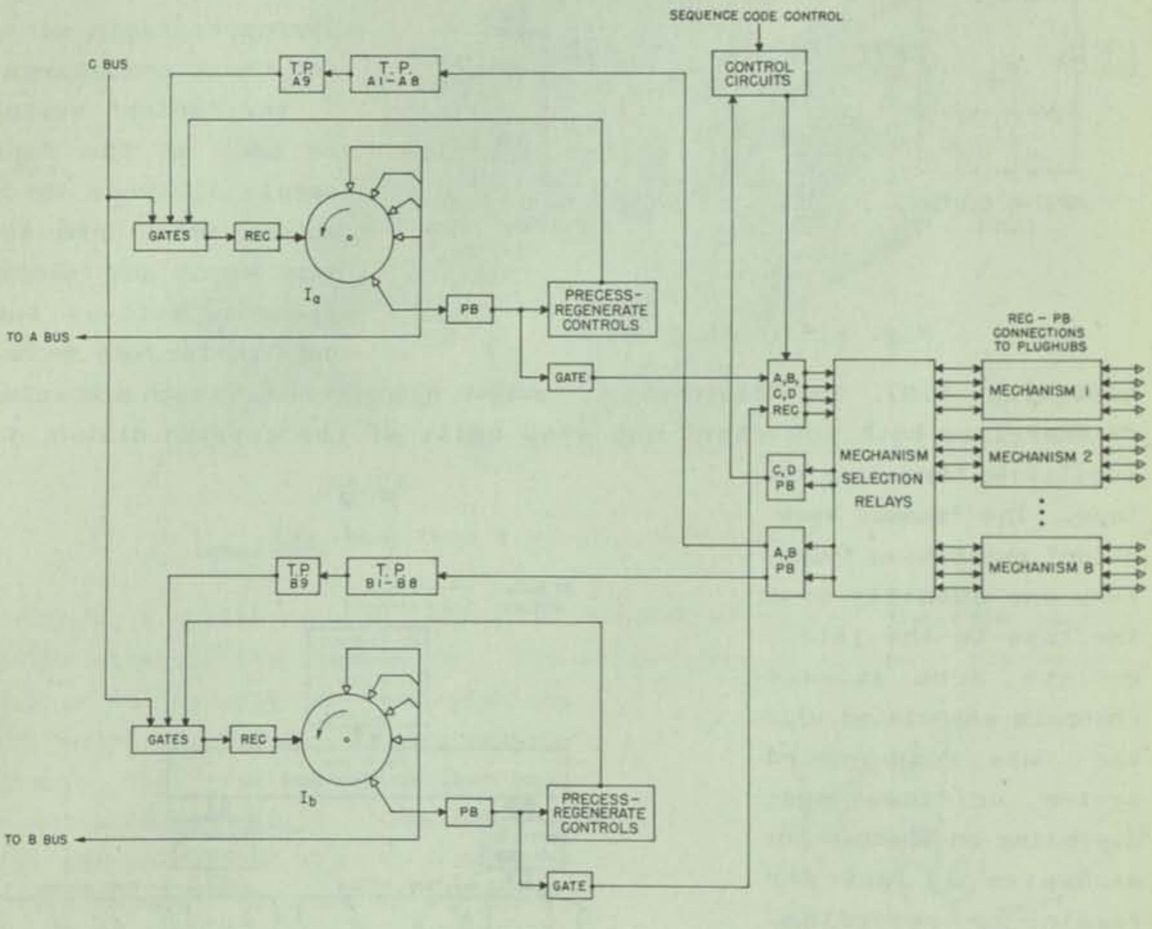


Fig. 6.10—*Block diagram, tape read-record system.*

fast storage system of the calculator. To transfer one quantity from tape to channels I_a and I_b, or vice versa, requires about twenty-five cycles of machine time. During this time operations not involving the tape read-record system may be interposed. An automatic wait circuit delays the execution of the second of two successive commands involving the input-output system until the operations initiated by the first command have been completed.

All quantities to be recorded on tape are derived from the C bus. Before the quantity on the C bus can be recorded on tape, a command must be given to cause this quantity to be stored on channels I_a and I_b. On each channel the complete quantity is recorded serially three times in the following manner: the twenty 2* components, followed by the twenty 4, 2, and 1 components; then the 2* components again, etc. Once recorded, the digits are continuously regenerated through the playback pole pieces provided for this purpose. The four additional playback pole pieces on channels I_a and I_b make it possible to read information on these channels to the A and B busses of the calculator. A check may therefore be made to insure that the quantity on the C bus was correctly transferred to channels I_a and I_b before the tape recording is initiated.

When the check has been passed, a command is given to record via any one of the eight mechanisms which has been preset for recording. Relays are picked up to connect the tape record circuits to the pole pieces on the selected mechanism, and the corresponding clutch coil is energized

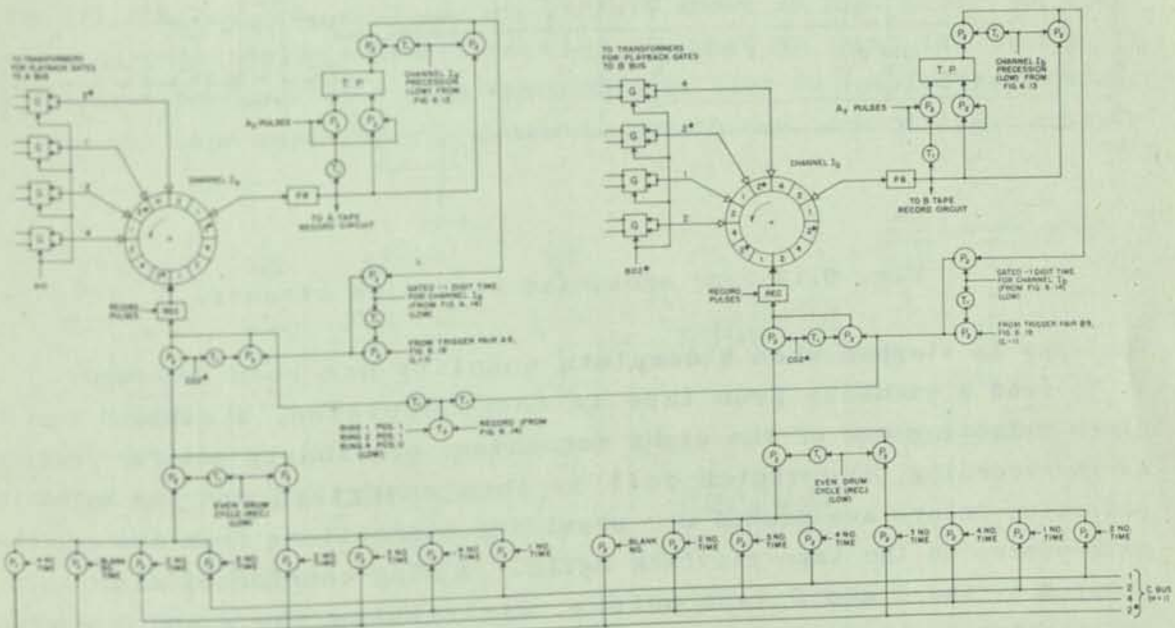


Fig. 6.11—*Channels I_a and I_b.*

to start the tape moving. The control circuits cause the digits on channels I_a and I_b to be precessed instead of regenerated, such that at intervals of one number time successive binary components will be delivered from the I_a and I_b playback units to the A and B tape record circuits, respectively. The control circuits also deliver pulses derived from the main storage drums to the C and D tape record circuits. A check is maintained between corresponding A and B digits during recording, and

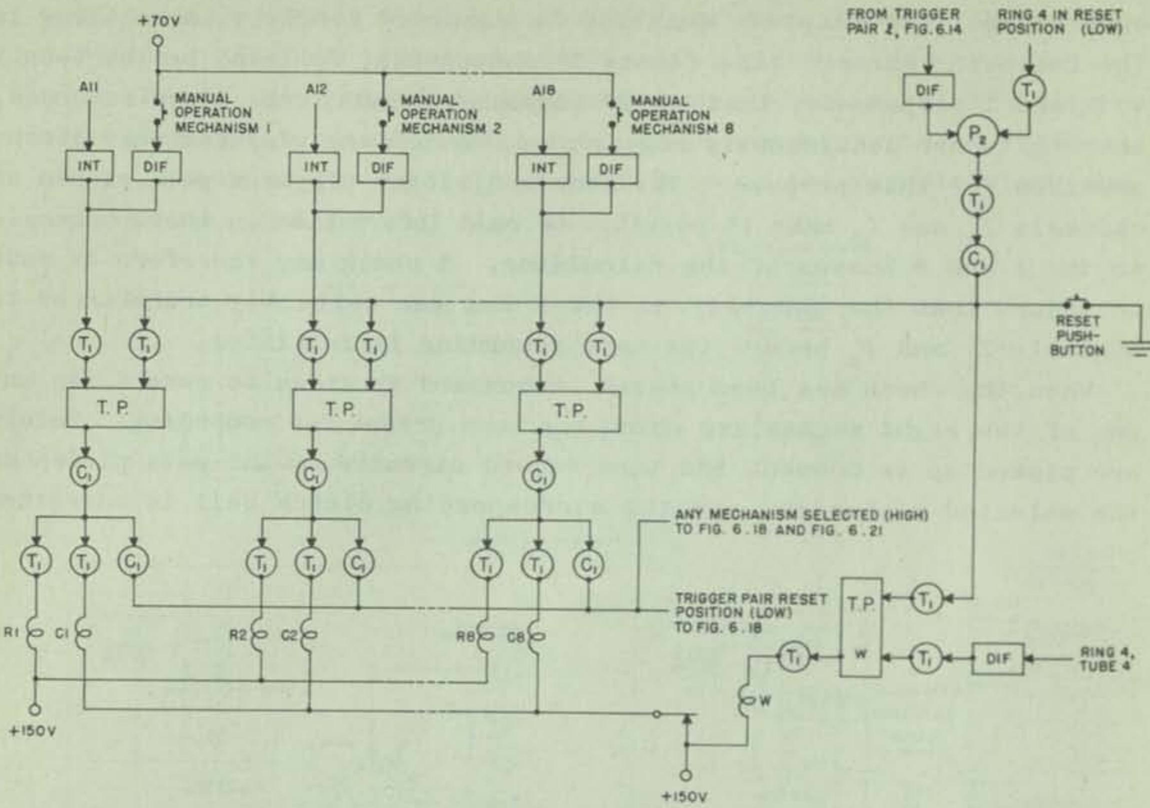


Fig. 6.12—*Tape mechanism selection circuit.*

the tape is stopped when a complete quantity has been recorded.

To read a quantity from tape to the calculator, a command must be given selecting one of the eight mechanisms previously set for reading. As in recording, the clutch coil is then energized and the mechanism selection relays are picked up, providing connections from the selected pole pieces to the tape playback units. A ring counter of eight tubes, stepped by the C and D tape pulses, distributes the A and B playback information to trigger pairs $A1$-$A8$ and $B1$-$B8$, respectively. A second ring of eight, controlled by pulses derived from the storage drums of the calculator, successively advances the A and B digits from these trigger pairs to trigger pairs $A9$ and $B9$. The digits in trigger pairs $A9$ and $B9$ are then recorded on channels I_a and I_b, respectively, and are precessed until the whole quantity has been played back from tape and recorded serially on the intermediate storage channels. At this time the tape is stopped and the digits on channels I_a and I_b are regenerated.

A second command is then given to read the quantity on channel I_b to any desired fast storage register. A check to compare the quantity in the fast storage register with the number on channel I_a insures that all transfers have been carried out correctly and that the two quantities read from the tape were identical.

The detailed operation of the tape read-record system may be considered by referring to Figs. 6.11-6.21. Figure 6.11 shows channels I_a and I_b and the associated control circuits at the start of 0 number time on an even drum cycle. A quantity is transferred from the C bus to these channels under control of code $C02^*$. In order that the 2*, 4, 2, and 1 components will always be recorded in the sectors indicated in the diagram, two sets of number time gates are provided for each channel, to be used on odd and even drum cycles. The displacement of one number time between each digit on channel I_a and the identical digit on channel I_b corresponds to the displacement between digits on the A and B tape channels.

The commands to select a mechanism for either reading or recording, codes $A11, A12, \ldots, A18$, control the circuit shown in Fig. 6.12. An integration circuit delays the selection so that in case the system is completing a previous operation there will be time to rescind the command. Provided the tape read-record system is not in use, one of eight trigger

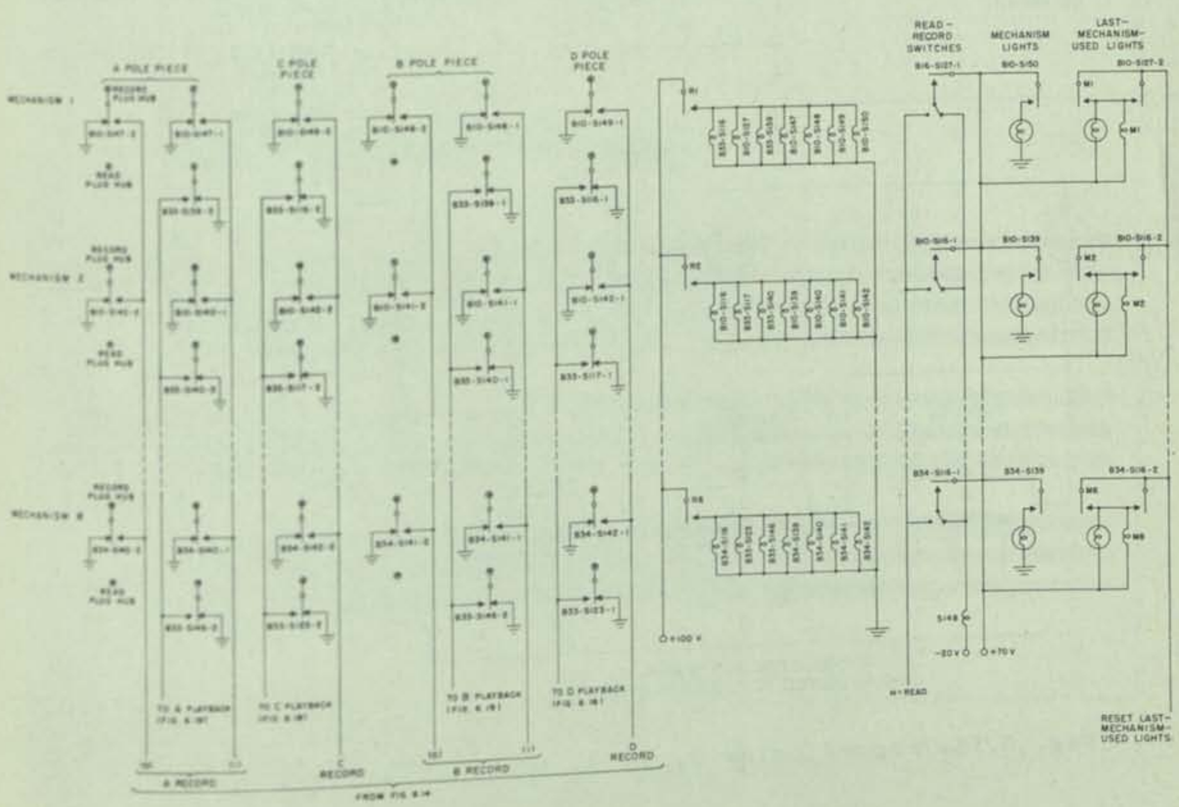


Fig. 6.13—*Read-record connections, last-mechanism-used lights.*

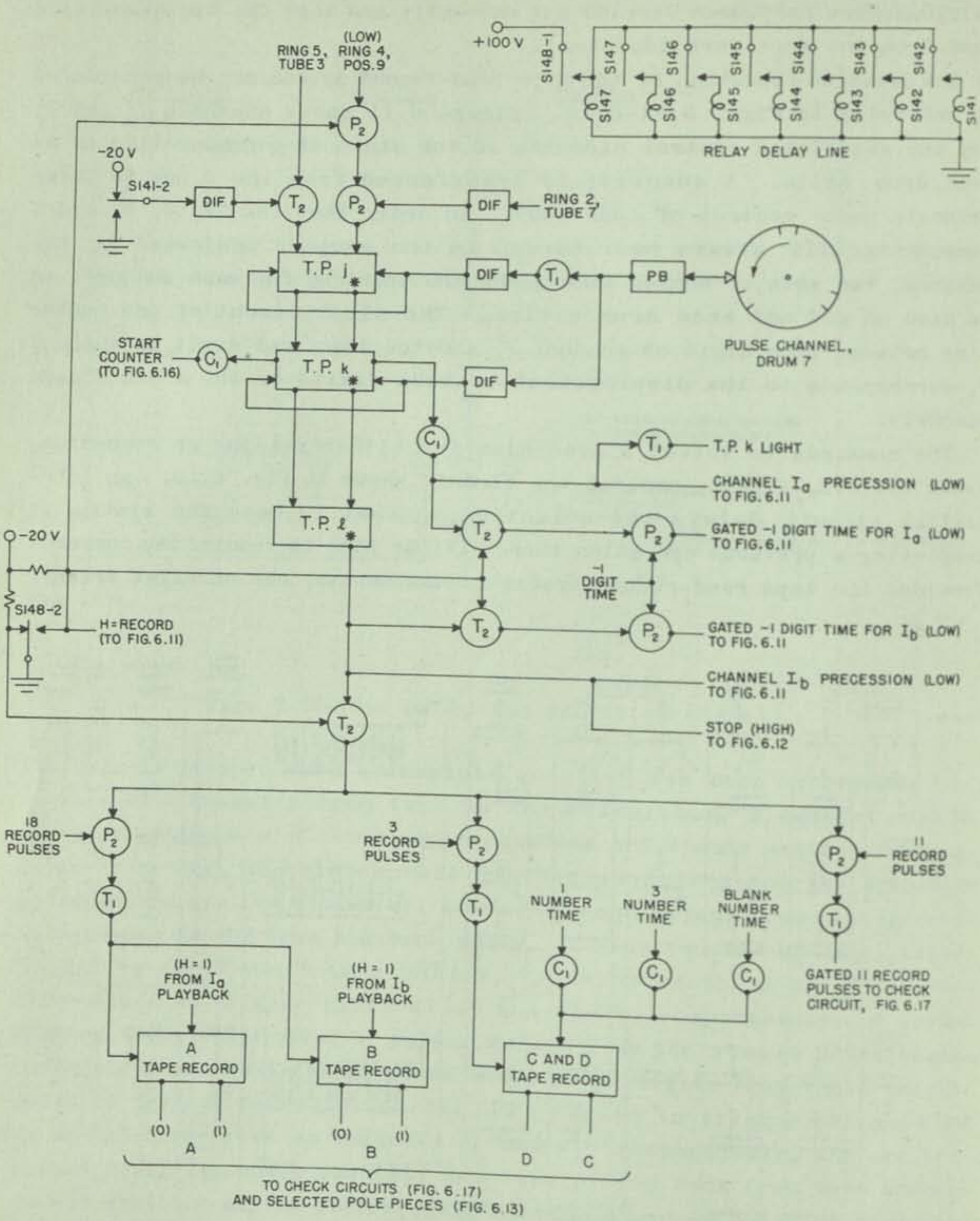


Fig. 6.14—*Trigger pairs j, k, ℓ, tape read-record system.*

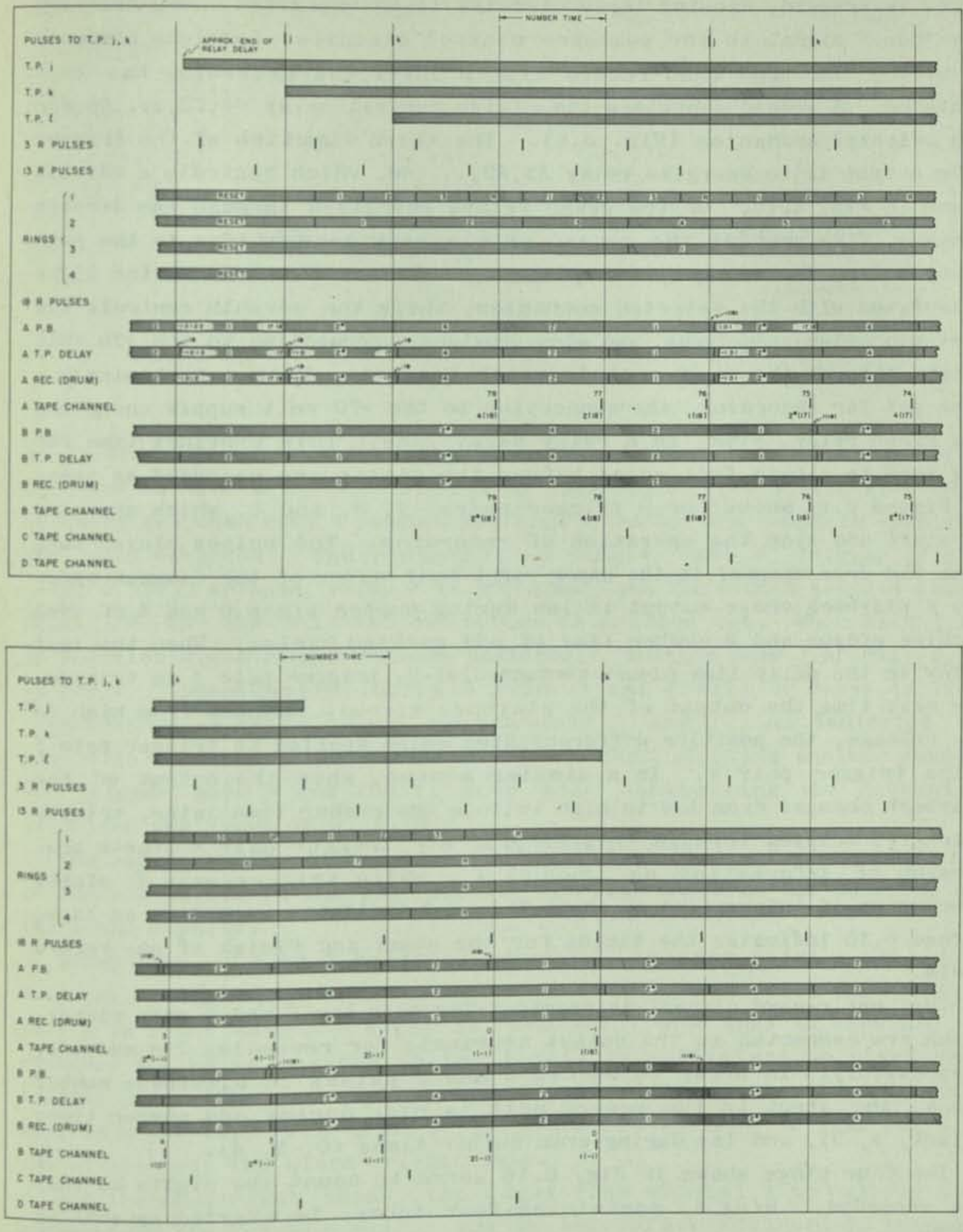


Fig. 6.15—*Tape record timing diagram.*

pairs is tripped, causing three circuits to be energized. One delivers the "busy" signal to the sequence control circuits, to block commands involving the tape read-record system until the recording has been finished. A second energizes the clutch control relay $C1, C2, \ldots, C8$ for the selected mechanism (Fig. 6.8). The third function of the trigger pair output is to energize relay $R1, R2, \ldots, R8$, which controls a circuit shown in Fig. 6.13. Of the seven relays energized through the R relay contact, five provide the record or playback connections to the pole piece coils, depending on the plugging. The sixth turns on the light associated with the selected mechanism, while the seventh controls the last-mechanism-used light, and also provides a connection to the +70 volt supply through the read-record switch contact. When a mechanism has been set for recording, the connection to the +70 volt supply energizes the first relay, $S148$, in a relay delay line. This provides time for the tape to attain full speed before the digits are recorded on tape.

Figure 6.14 shows three trigger pairs, j, k, and ℓ, which are used to start and stop the operation of recording. The pulses played back from the drum channel in the upper right-hand corner of the diagram operate a playback whose output is low during number times 0 and 4 of even machine cycles and 2 number time of odd machine cycles. When the last relay in the delay line closes contact $S141$-2, trigger pair j is tripped. The next time the output of the playback circuit changes from high to low voltage, the positive differentiated pulse applied to trigger pair j trips trigger pair k. In a similar manner, when the output of the playback changes from low to high voltage one number time later, trigger pair ℓ is tripped through trigger pair k. Trigger pair k starts precession of information on channel I_a, while trigger pair ℓ starts precession of information on channel I_b and initiates recording on tape. Figure 6.15 indicates the timing for the start and finish of one record cycle.

Just one record circuit is provided for both the C and D pole pieces, which are connected to the output terminals for recording 1's and 0's, respectively. In order to record C and D pulses in alternate number times, the input to the record unit is high during odd number times (blank, 1, 3), and low during even number times (0, 2, 4).

The four rings shown in Fig. 6.16 serve to count the digits as they are recorded. Ring 1, containing four tubes, is started as soon as trigger pair k has been tripped, and is advanced regularly by the record pulses at 3 and 13 digit times. Rings 2, 3, and 4, containing eight, four, and ten tubes, respectively, are advanced by pulses obtained from

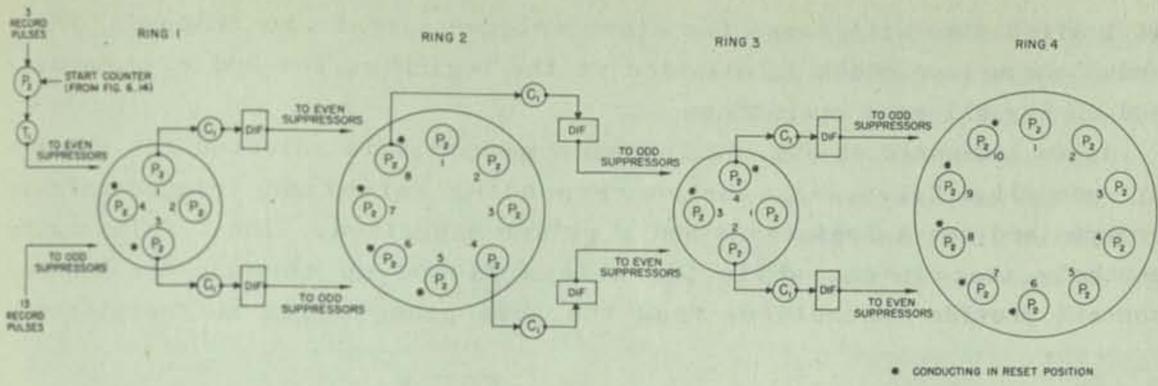


Fig. 6.16—*Tape record check circuit.*

the preceding rings, as indicated in the diagram. Since tape will continue to move after the clutch is disengaged, the clutch stop coil is energized when ring 4 reaches position 4, before the complete quantity has been recorded. Under control of ring 4, tube 4, trigger pair w (Fig. 6.12) is tripped, relay W is energized, and the clutch control relay $C1,\ldots,C8$, for the selected mechanism is dropped out. When ring 4 is in position 9 and ring 2 reaches position 7, trigger pair j of Fig. 6.14 is reset. Consequently, trigger pairs k and ℓ will be reset causing regeneration to take place on channels I_a and I_b, as indicated in Fig. 6.15. The ring counter is prevented from starting another count, and trigger pair w and the trigger pair controlling the mechanism selection are reset (Fig. 6.12).

The record check circuit shown in Fig. 6.17 checks for the presence of current pulses in the pole pieces, and also compares the polarity of the A and B pulses.

Every time a C or D pulse is recorded, trigger pairs r and s are tripped. Trigger pair r is reset when an A digit is recorded and trigger pair s is reset when a B digit is recorded. Note that in case a 0 is recorded the pulse induced in the 1 coil is sufficient to reset the trigger pair. When any of these events fails to occur, the record pulses applied to the suppressor grids of the pentodes comprising trigger pairs r and s will trip the alarm trigger pair.

For the comparison check, the digit from channel A is delayed one number time in trigger pair m and is thereafter advanced to trigger pair q. The digit from channel B is read directly into trigger pair p. If and only if these digits are identical, the output from trigger pair q is low. In case the output of trigger pair q is high, the record pulse at 1 digit time will cause the alarm trigger pair to be tripped. The *A* and *B* comparison check is disabled at the beginning and end of recording and during all read operations.

As is indicated in Fig. 6.12, when a mechanism is selected for reading by code *A*11,*A*12,...,*A*18, the corresponding selection trigger pair is tripped and the associated *R* and *C* relays energized. The *C* relay again controls the clutch, while the relays picked up through the *R* relay contact provide connections from the pole piece coils to the playback

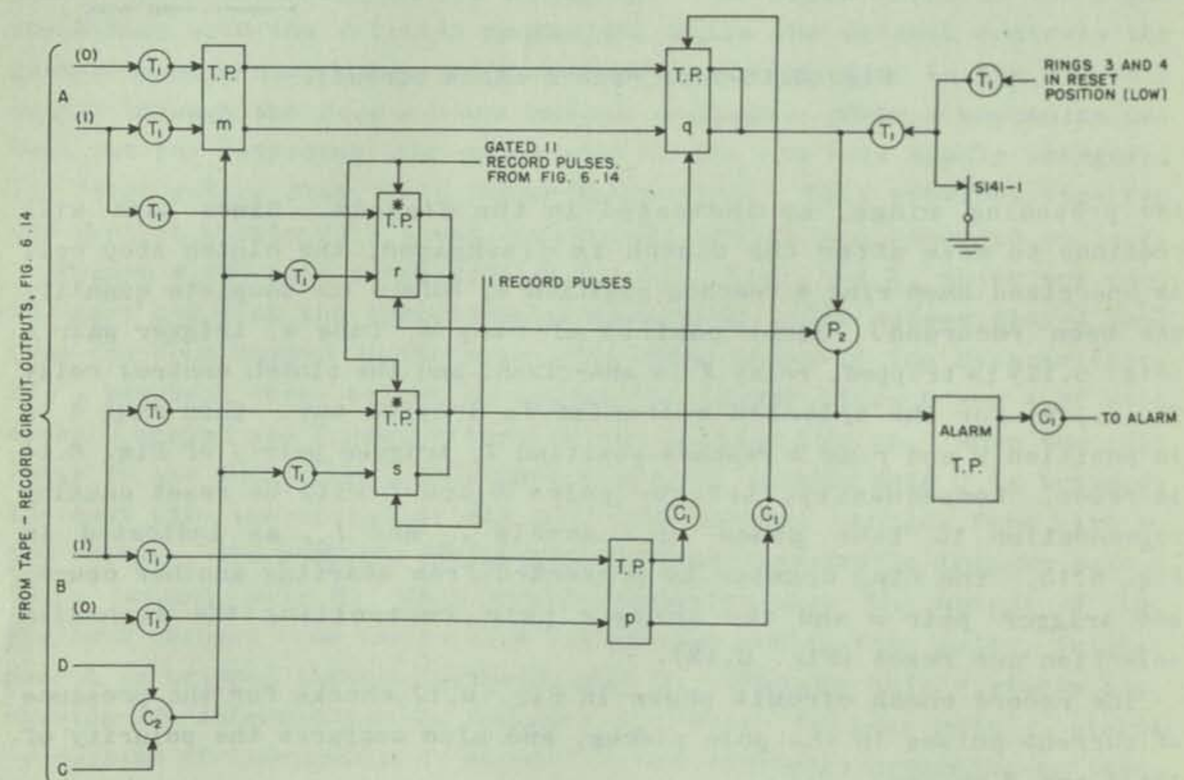


Fig. 6.17—*Ring counter, tape read-record system.*

circuits. The connection to the +70 volt supply through the read-record switch contact provides the signal to open a pulse gate, Fig. 6.18, admitting differentiated pulses from the *C* playback to the even suppressor grids in ring 5. Differentiated *D* pulses are delivered to the odd suppressor grids, causing the ring to step once for each *C* or *D* pulse. It is this ring of eight tubes which distributes the *A* and *B* playback information in trigger pairs *A*1-*A*8 and *B*1-*B*8.

The information in the *A* and *B* tape channels is delivered to playback circuits through the pole piece coils used for recording 1's. This preserves the convention that a high voltage output from a playback represents a 1 and a low voltage, a 0. The tape playback circuits used here are similar to the fast storage playback circuits, but have different circuit constants. Also, the final trigger pairs in the *A* and *B* playback circuits are omitted. The playback outputs are distributed to trigger pairs *A*1-*A*8, *B*1-*B*8, Fig. 6.19, under control of differentiated pulses from ring 5. Thus the pulses from ring 5 serve the same purpose as the *A*1 pulses used in fast storage playbacks in that they arrive shortly after the crossover point of the entering pulse and sense the polarity of the *A* and *B* digits.

The first *C* pulse played back from the tape steps ring 5 to position 1, as indicated in Fig. 6.18 and the timing diagram, Fig. 6.20. At this time no *A* or *B* digits have yet been played back, so that no information is stored in trigger pairs *A*1 and *B*1. When ring 5 is stepped to position 2 by the first *D* pulse, the *A* and *B* playback outputs, representing the 4 and 2* components of digit 18, respectively, are stored in trigger pairs *A*2 and *B*2. The next *C* pulse steps ring 5 to position 3, causing the *A*2-18 and *B*4-18 digits to be stored in trigger pairs *A*3 and *B*3. This process continues until the complete quantity has been read from tape.

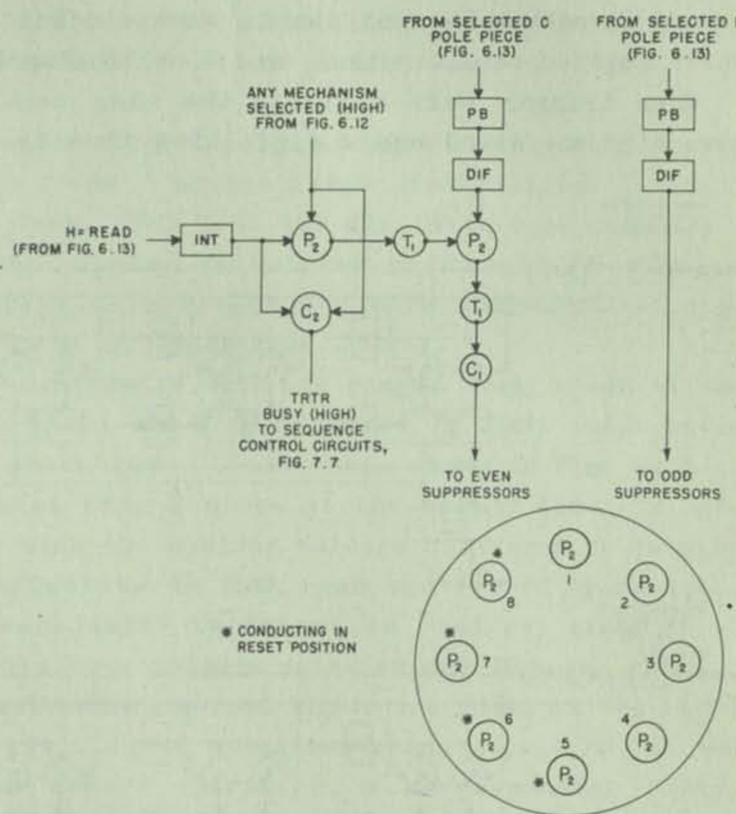


Fig. 6.18—*Ring 5, tape read-record system.*

Meanwhile, when ring 5 reaches position 3, trigger pair *j* of Fig. 6.14 is tripped. As in recording, trigger pairs *k* and *ℓ* are subsequently tripped, the ring counter is started and the digits recorded on channels I_a and I_b are precessed. At -1 digit time, precession is interrupted and the digits stored in trigger pairs *A*9 and *B*9 are recorded on channels I_a and I_b respectively. In this way, one digit from tape is recorded on channels I_a and I_b in each number time. Once recorded the digits are precessed around the channel so that they will be correctly located when the reading has been completed.

At the beginning of each tape reading operation the record circuits of channels I_a and I_b are controlled by the circuit shown in the center of Fig. 6.11. This circuit causes 0's to be recorded in channel I_a and 1's to be recorded on channel I_b. Thus if no new information is recorded on either channel due to some electrical failure, the identity check applied to channels I_a and I_b will stop the machine.

When trigger pair *k* starts the ring counter, ring 2 will begin advancing one *A* and one *B* digit at a time to trigger pairs *A*9 and *B*9.

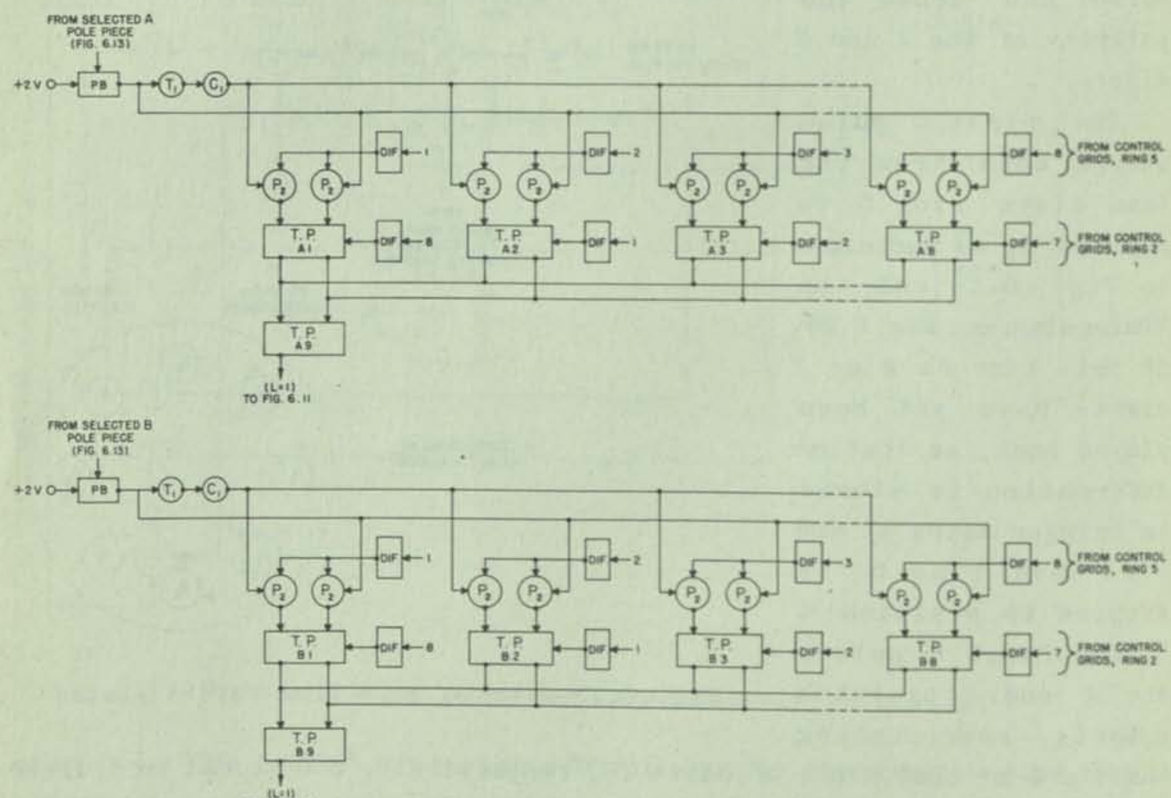


Fig. 6.19—*Trigger pairs A*1-*A*9, *B*1-*B*9, *tape read-record system.*

To make sure that ring 2 will not overtake ring 5 and read out some information twice, a conditional stop is made by resetting trigger pair *J*, Fig. 6.14, each time ring 2 reaches position 7. If ring 5 is in position 3, 4, 5, or 6, or reaches one of these positions before a cycling pulse arrives at the suppressor grids of trigger pair *J*, then trigger pair *J* will be tripped out of the reset position, and the cycling pulse will not change the position of trigger pair k. However, if trigger pair j remains reset long enough for a cycling pulse to arrive and reset trigger pair k, then rings 1, 2, 3, and 4 are stopped, and the channel I_a regeneration gate is opened. One number time later trigger pair ℓ is tripped, causing the channel I_b digits to regenerate also. Meanwhile, ring 5 continues to store information from the tape in trigger pairs $A1,\ldots,A8,B1,\ldots,B8$. As soon as ring 5 reaches position 3, trigger pair j is tripped out of its reset position, and, under control of the cycling pulses, the counter rings and the precession controls are started again.

The clutch is stopped as in recording, when ring 4 reaches position 4. Ring 5 stops when all the C and D pulses associated with one quantity on tape have been played back. The last two digits in tape channels A and B are therefore never stored in trigger pairs nor recorded on channels I_a and I_b. However, since they are both components of non-informative digits, this is of no importance.

Two check circuits are associated with the rings. One, shown at the right of Fig. 6.21, insures that all five rings are in their reset positions at the start of an operation. The second, shown in Fig. 6.21 on the left, checks to see that ring 2 steps at the proper times by comparing two ring positions with the cycling voltage delivered to trigger pair J. The checks are effective in both read and record operations. The check on ring 2 is especially important in reading, since it is ring 2 which advances digits into trigger pairs $A9$ and $B9$.

The controls for the last-mechanism-used lights are shown at the right in Fig. 6.13. When code $A11,\ldots,A18$ energizes relay $R1,\ldots,R8$, at the start of either a read or record operation, a corresponding relay, $M1,\ldots,M8$ is energized, turning on the associated light. When ring 4 reaches position 1, relay N is energized and held, regardless of which mechanism was selected. This causes the M relay for the mechanism just selected and also the M relay for the last mechanism selected to be dropped out. When ring 4 reaches position 6, relay N is dropped out. Then the M relay for the mechanism in use is energized again, and held until another mechanism is selected.

This chapter concludes with a description of the five printers. Each is a separate unit consisting of the following parts:

1. a tape playback mechanism, Plate XXIV, and playback circuits, for reading numerical information from magnetic tape;
2. relays of the six-contact type, mounted on the rear panels of

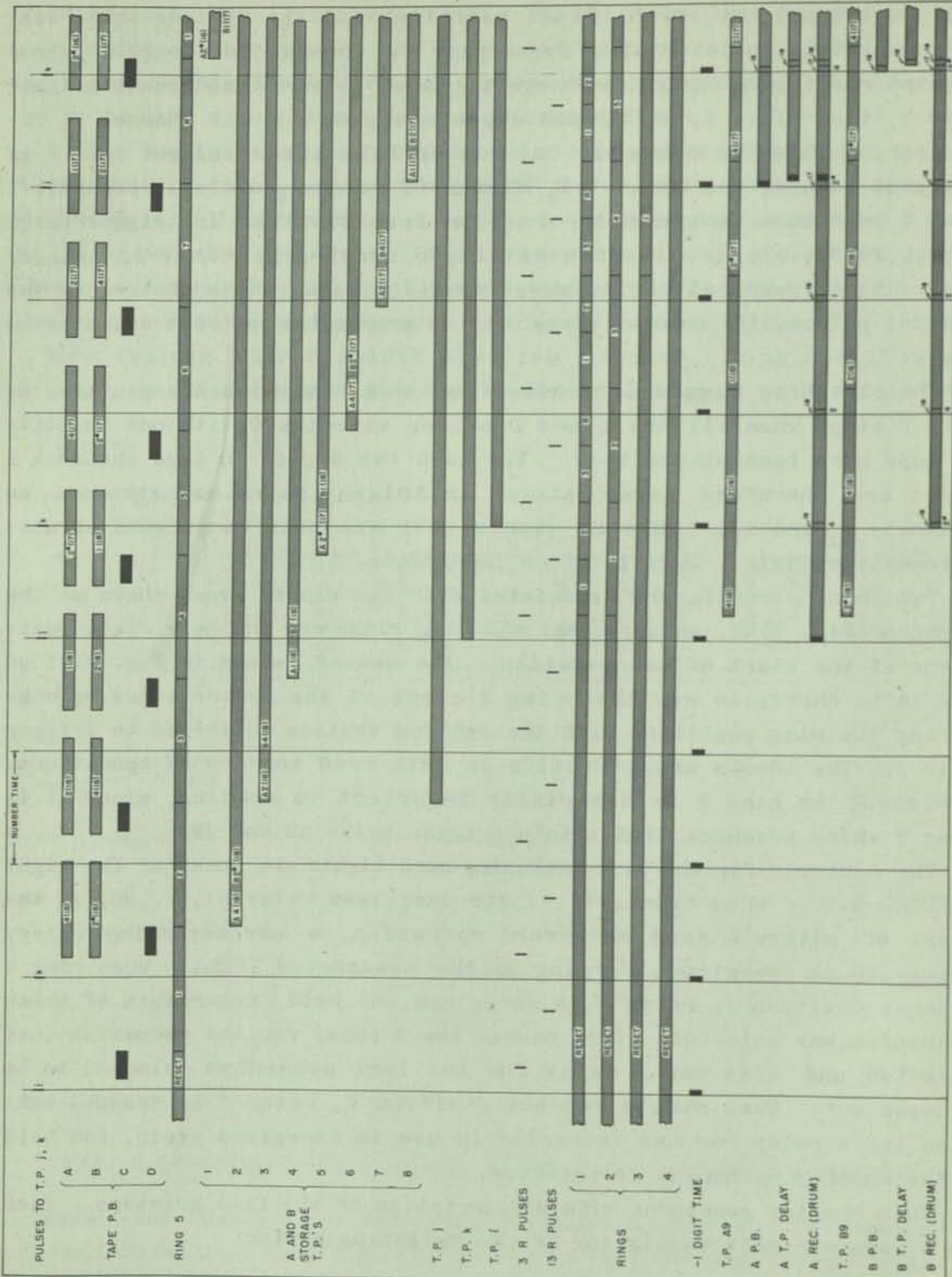


Fig. 6.20—*Tape read timing diagram.*

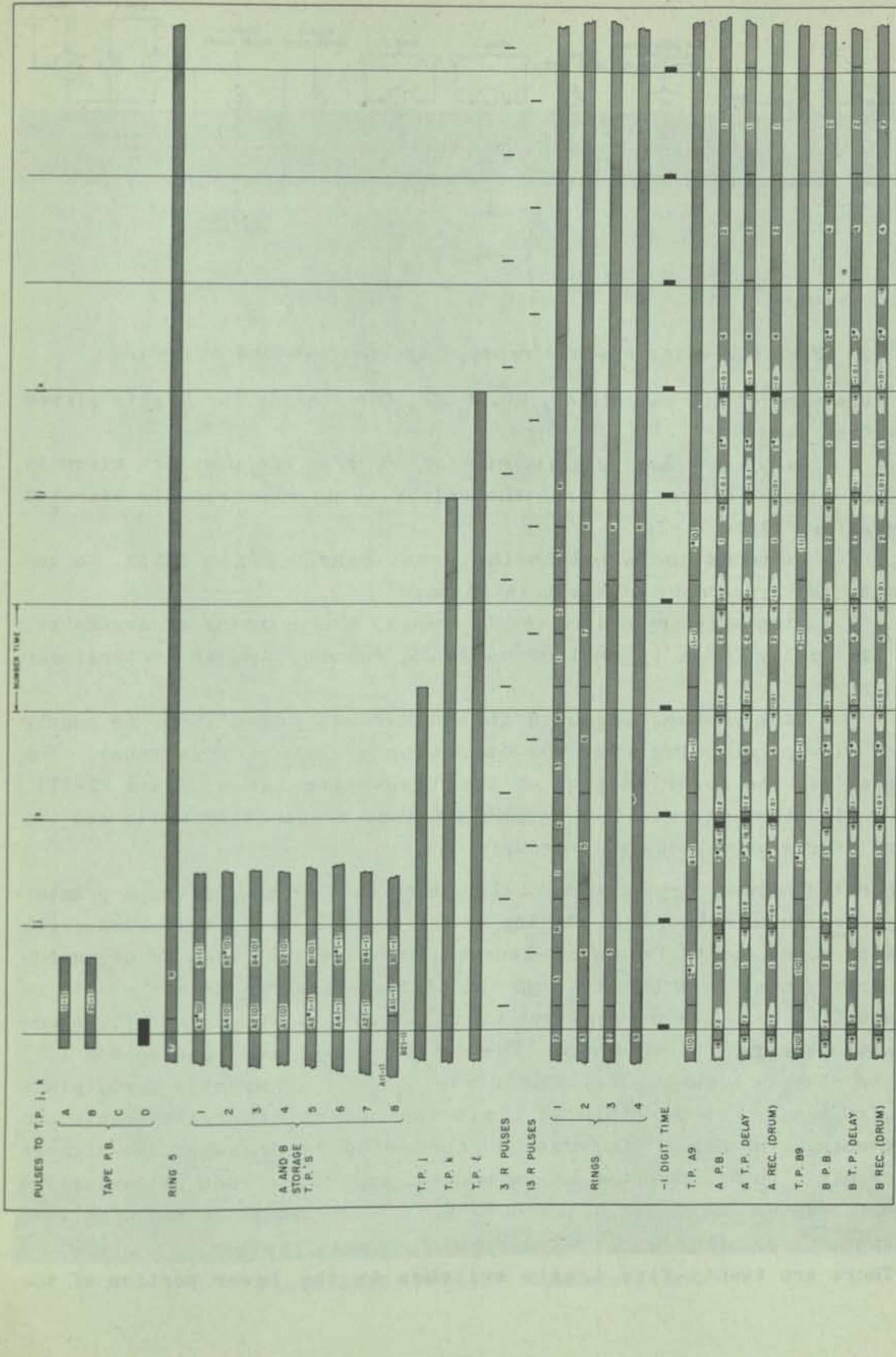


Fig. 6.20 (continued)

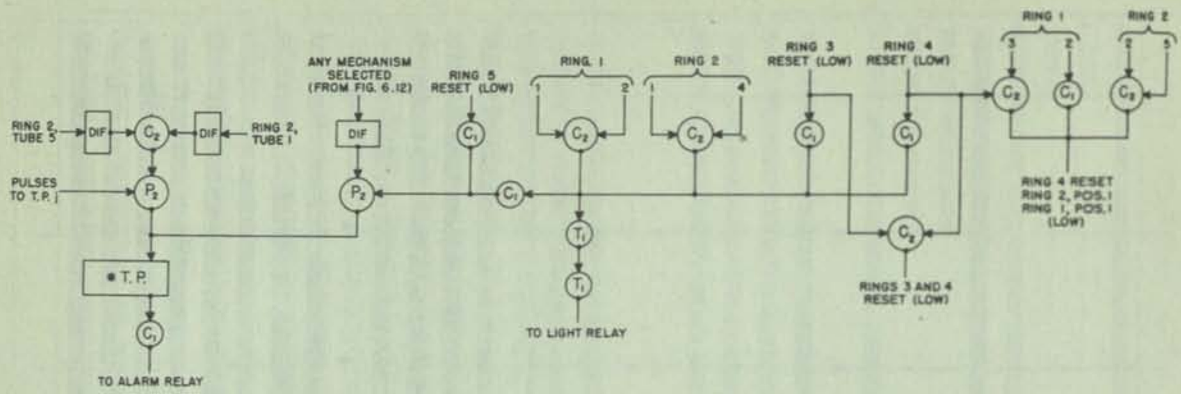


Fig. 6.21—*Ring check circuit, tape read-record system.*

the left wing of the calculator, Plate XXV, for storing the digits played back from tape;

3. step switches for delivering digits from the playback circuits to the storage relays, and from the relays to an electrically operated typewriter, Plate XXVI;

4. switches and plugs on the front panel, Plate XXVII, to determine the typography of the printed page;

5. step switches and relays to control the printing of arguments, the time delay for a typewriter carriage return, and the interlinear spacing;

6. cams of the type used in the Mark II Calculator, to supply timed electrical pulses for the operation of the step switches. The cams are in the lower section of the typewriter table, Plate XXVIII, along with the cams required by the two tape preparation units and the calculator's manual control circuits.

First consider the manual controls, which must be set before a printer is placed in operation. At the top of the control panel, the print jacks must be connected to the twenty-four column-selection jacks to determine the arrangement in which the sign and digits of each quantity are to be printed. A positive or negative quantity will be preceded by a space or a minus sign, respectively. The decimal point (*DP*) and spaces (*SP*) may be plugged between digits as desired. The argument control plugs (AC_1,AC_2) make it possible to limit the number of digits printed in arguments. The first plug must be connected in front of the highest order digit to be included in arguments, and the second following the lowest order argument digit. If desired, the decimal point may be omitted in argument printing by throwing the *D.P. cut-off* switch.

There are twenty-five toggle switches in the lower portion of the

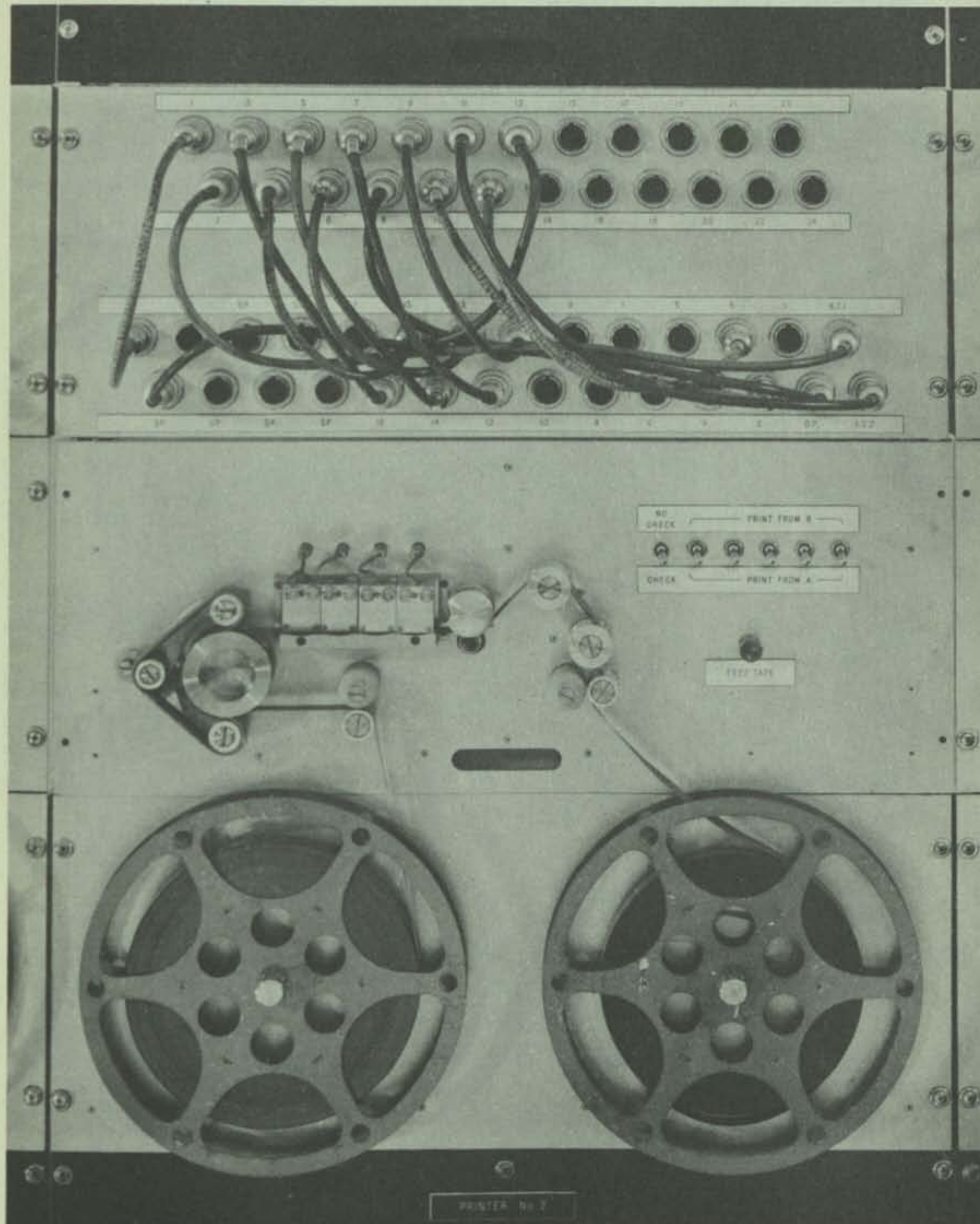


Plate XXIV *Printer Tape Mechanism*

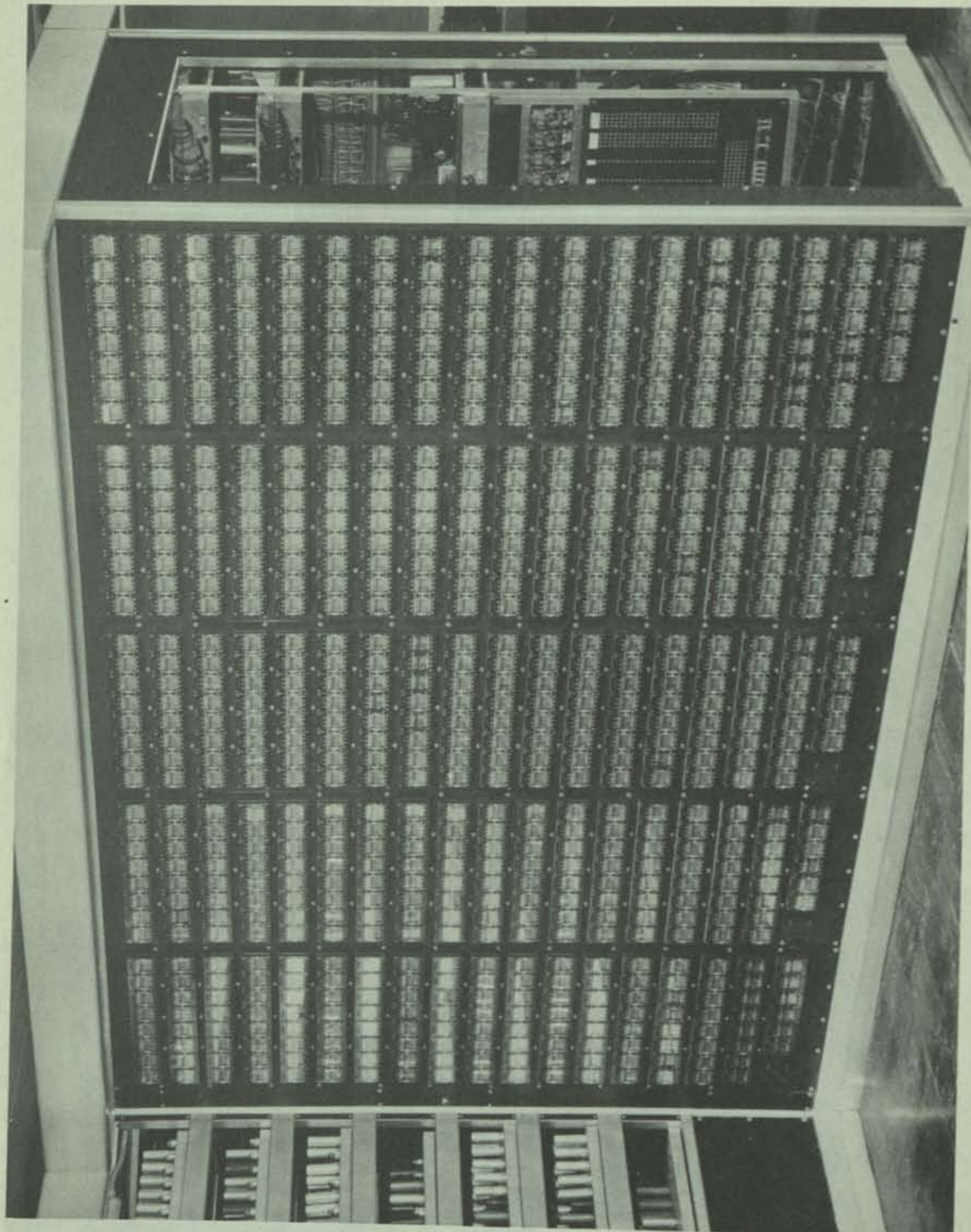


Plate XXV *Printer Relay Panel*

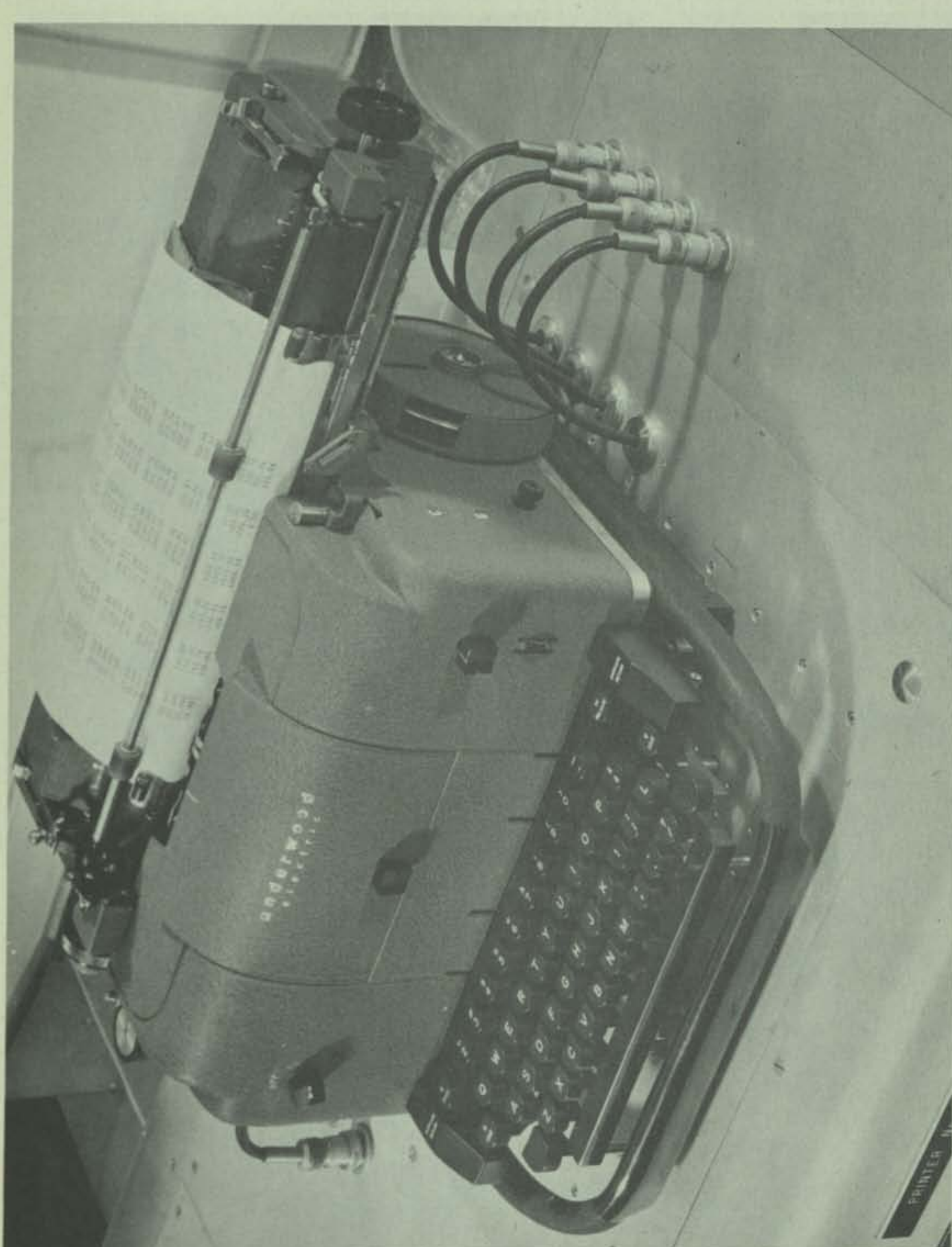


Plate XXVI Typewriter

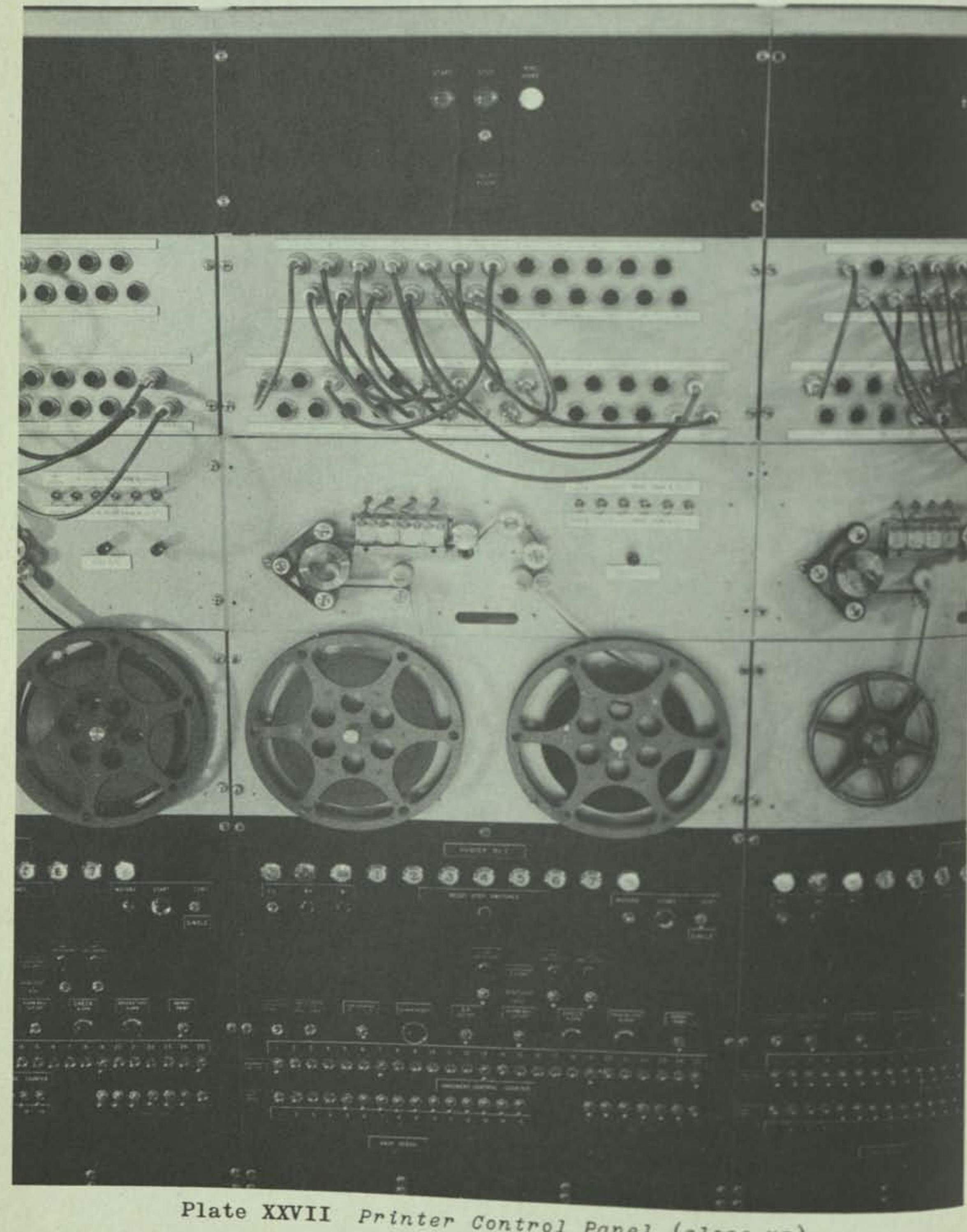


Plate XXVII *Printer Control Panel (close-up)*

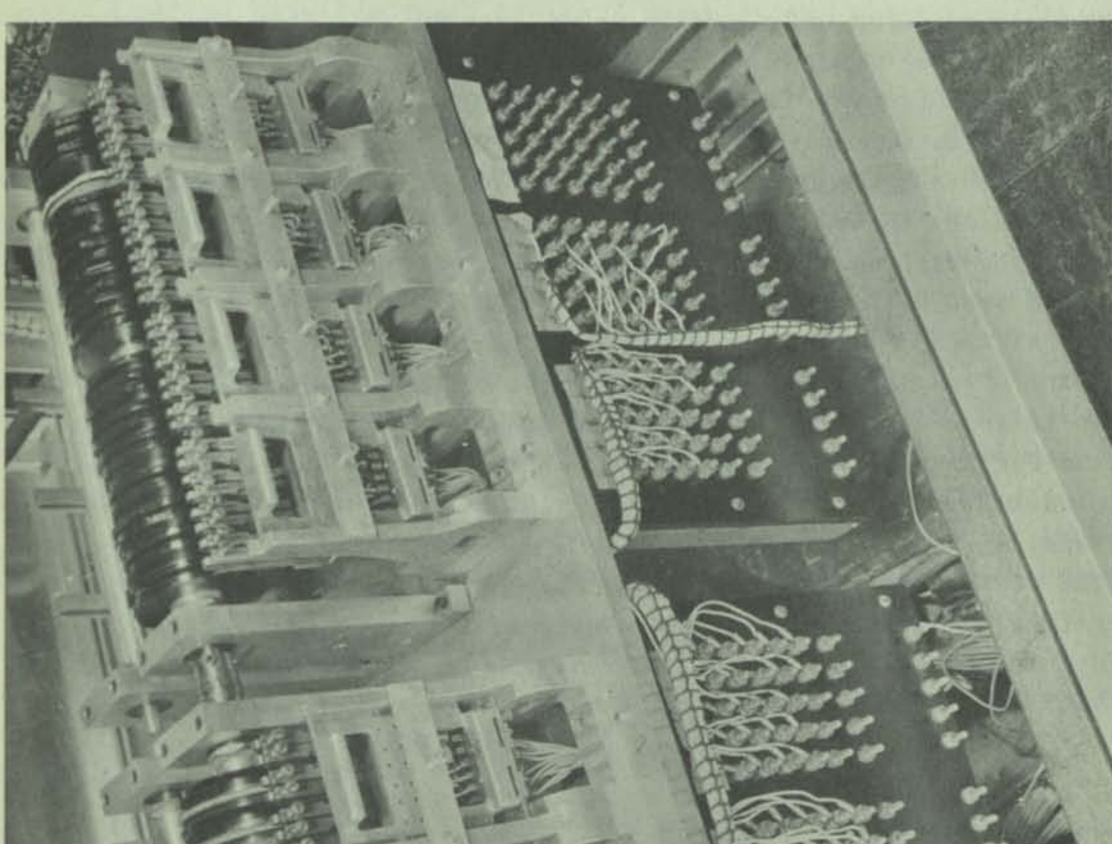


Plate XXVIII *Cam Unit*

front panel, associated with the twenty-five steps of a step switch. The step switch is advanced one step for each quantity played back from tape, causing the quantity to be printed as an argument only when the step switch is on a step for which the corresponding switch has been thrown. Thus if every argument is to be followed by n functional values, then every $(n+1)$st switch must be thrown. The four switches immediately below the last four argument control switches are provided to reset the step switch, omitting when necessary the controls on one or more of the last four steps. If twenty-five divided by $n+1$ leaves a remainder r, then r of the lower switches must be thrown.

The fifteen drop-zero switches for columns 2 through 16 prevent the printing of zeros in front of the highest order non-zero digit when this digit is to the left of the decimal point. When a zero occurs in a column for which the drop-zero switch has been thrown, a zero is not printed unless one or more of the higher order digits was not zero. Instead, a space or minus sign is printed, depending on the information in the next lower column and on the algebraic sign. For example, if the operating decimal point is between columns 12 and 13, drop-zero switches in columns 14, 15 and 16 are thrown, and spaces plugged to print the decimal digits in groups of four, then for these numbers recorded on tape,

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

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

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

the printed results will appear as follows:

(a) 1.2345 6789 1234

(b) -0.0012 3456 7890

(c) 5000.6789 1234 5678 .

The amount of information to be printed on one line is determined by adjusting the margin controls on the typewriter carriage, the capacity being ninety-five characters. A microswitch on the right margin of the typewriter provides for a carriage return at the end of each line.

Interlinear spacing is determined by the positions of two switches. One provides a blank line between either every three or every five lines of print. The other, in the on position, marks the end of a page by supplying thirteen blank lines, either after every ten groups of five lines each or after every twenty groups of three lines each.

Five print switches are provided to determine whether the information recorded on the *A* or *B* tape channels is to be printed. There is also a check switch, determining whether or not the *A* and *B* digits are to be compared before printing the *A* digits. Normally the print switches are in the *A* position and the check switch in the check position, causing the *A* and *B* digits played back from tape to be stored in two sets of relays, the print and check relays. Each *A* digit is then delivered to the typewriter solenoids and printed. At the time of printing each *A* digit is compared with the corresponding *B* digit through contacts located in the typewriter, insuring that the digit is printed and also that it is the same on both tape channels. However, for test purposes the *A* channel digits may be printed without a check by repositioning the check switch. When the print switches are in the *B* position, the *B* information will be stored in the print relays and printed. Since in this case the *A* digits are not stored, no check may be made and the check switch must be in the no-check position.

As on the tape read-record mechanisms, insufficient tape or broken tape in a printer mechanism will cause an alarm to be sounded, through

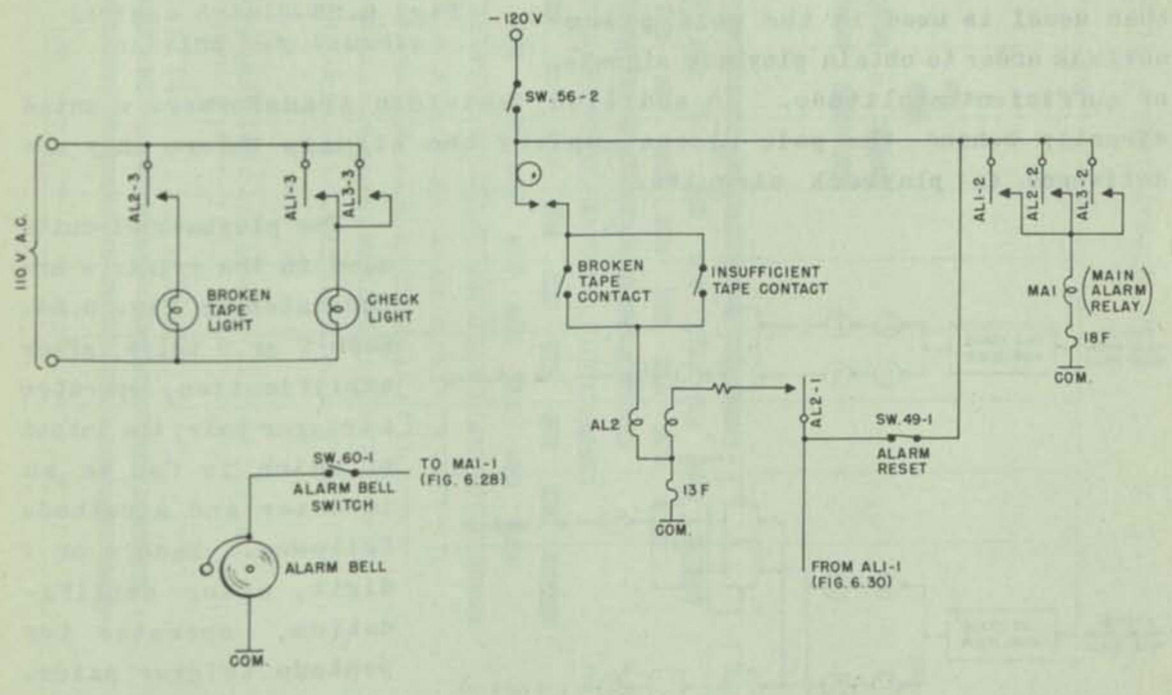


Fig. 6.22—*Printer alarm circuits.*

the circuit shown in Fig. 6.22. Spring contacts are again used to sense for these conditions.

Before starting a printer, the power must be turned on by throwing the switches associated with the filaments, the positive and negative voltage supplies, and the motors. The alarm bell cut-off switch must be in the on position to start, though it may be turned off later. The step switches and electronic circuits must be in their reset positions, indicated by associated lights. The printing may be done one quantity at a time or continuously.

When the start button is depressed, the start clutch relay is energized, Fig. 6.23, tripping the clutch control trigger pair. The clutch start-coil is then energized and the tape starts moving. As the tape is run at the comparatively slow speed of 1.4 inches per second, a larger number of turns than usual is used in the pole piece coils in order to obtain playback signals of sufficient amplitude. In addition, shielded transformers mounted directly behind the pole pieces amplify the signals before they are delivered to playback circuits.

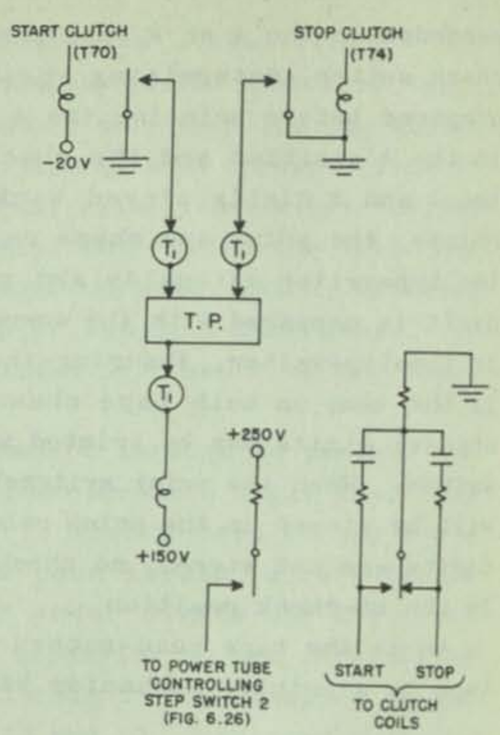


Fig. 6.23—*Clutch control circuit for printer.*

The playback circuits used in the printers are indicated in Fig. 6.24. Each *C* or *D* pulse, after amplification, operates a trigger pair, the output of which is fed to an inverter and a cathode follower. Each *A* or *B* digit, after amplification, operates two pentode trigger pairs. The *C* and *D* playback outputs are used to advance digits through the two pentode trigger pairs to triode trigger

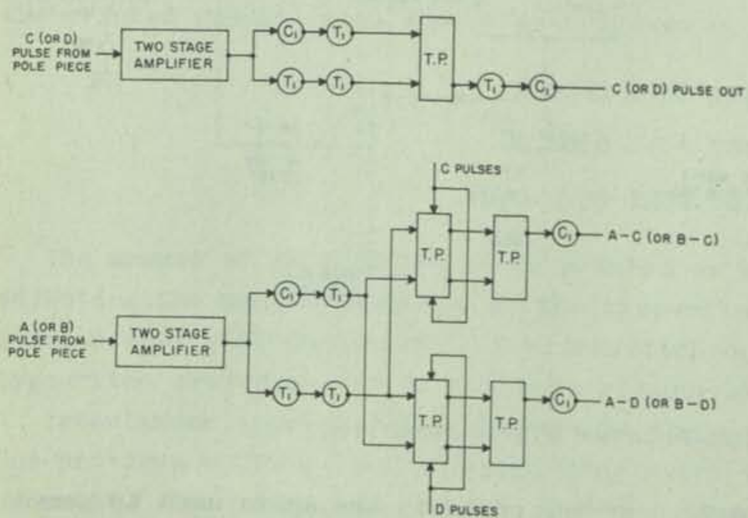


Fig. 6.24—*Printer playback circuits.*

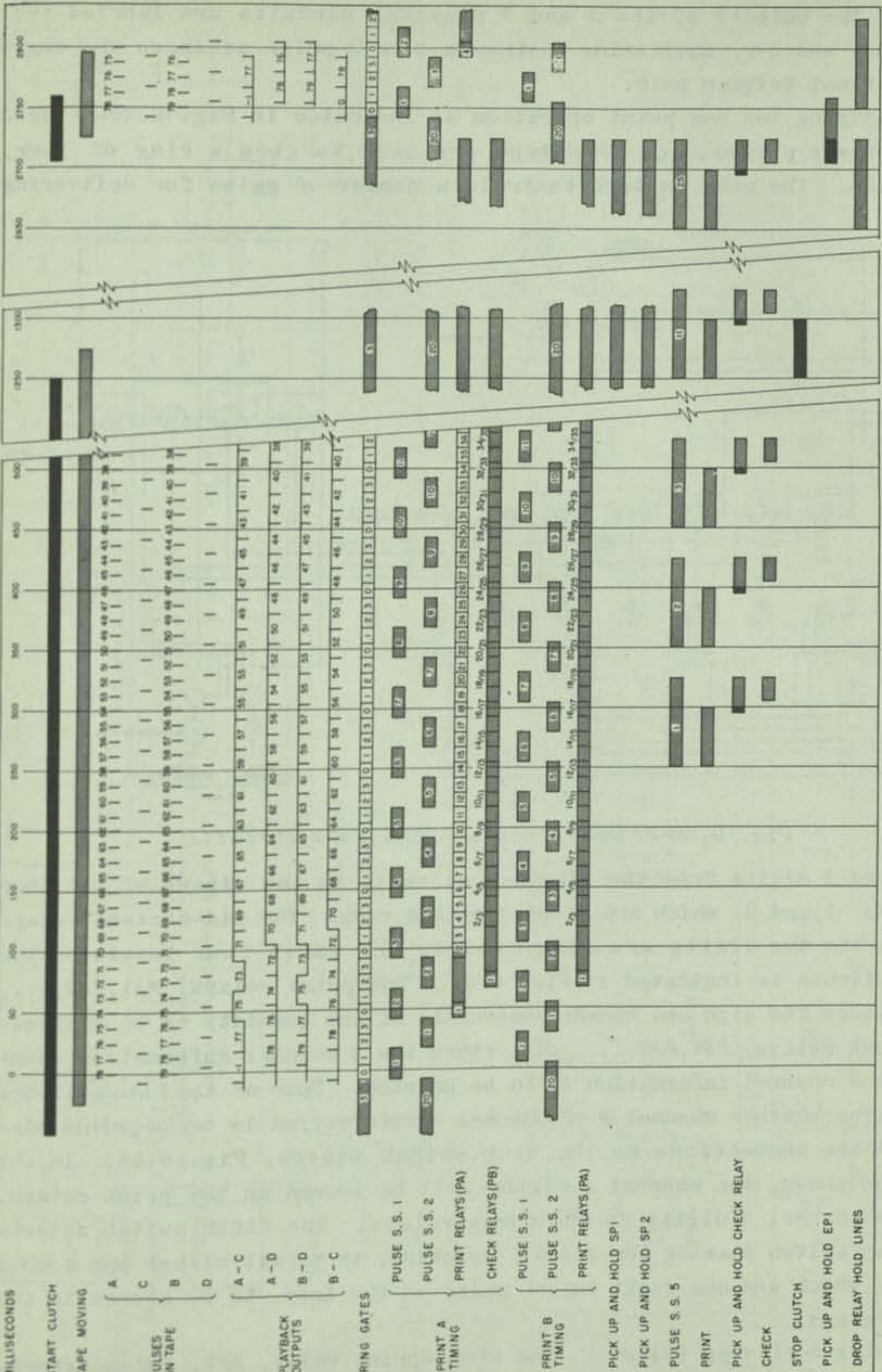


Fig. 6.25—Printer timing diagram.

pairs. The outputs of the *A* and *B* playback circuits are labeled *A-C*, *A-D*, *B-C*, and *B-D*, indicating whether a *C* or *D* pulse advances the digit to the final trigger pair.

The timing for one print operation is indicated in Fig. 6.25. The *C* and *D* pulses played back from tape are used to step a ring of four, Fig. 6.26. The ring in turn controls a number of gates for delivering

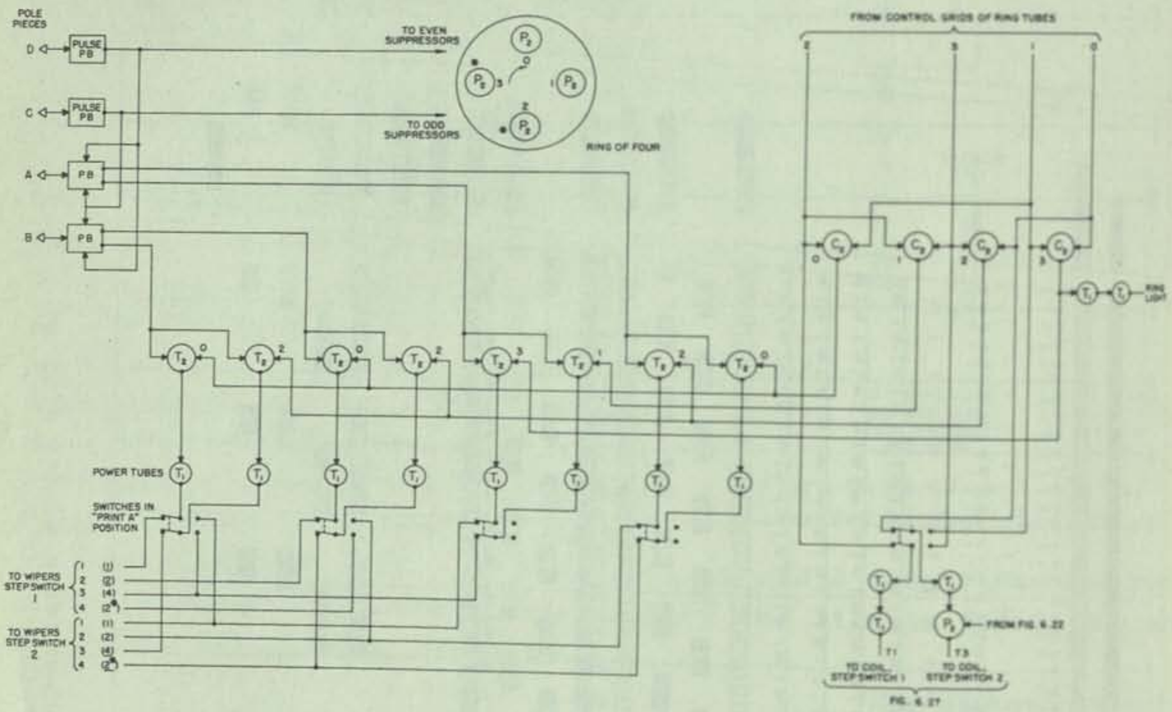


Fig. 6.26—*Printer playback and gate circuits.*

the *A* and *B* digits from the playback circuits to the wipers of two step switches, 1 and 2, which are stepped by the ring. The six-contact relays for storing the digits are connected to the first four levels of the step switches as indicated in Fig. 6.27. The print relays, *PA*1,*PA*2,..., *PA*65, store the sign and binary components of the quantity to be printed. The check relays, *PB*1,*PB*2,...,*PB*65, store the *B* channel information whenever the *A* channel information is to be printed. Four of the five switches indicating whether channel *A* or channel *B* information is to be printed determine the connections to the step switch wipers, Fig. 6.26. In the position shown, the channel *A* digits will be stored in the print relays, and the channel *B* digits in the check relays. The fifth switch adjusts the step switch phasing for *A* and *B* printing, to permit either the *A* or *B* digits, which are one digit out of phase on the tape, to be stored in the print relays.

On step 5 of step switch 2, the start-print relay, *SP*1, is energized. A cam-timed voltage energizes relay *SP*2 through the *SP*1-1 contact. These

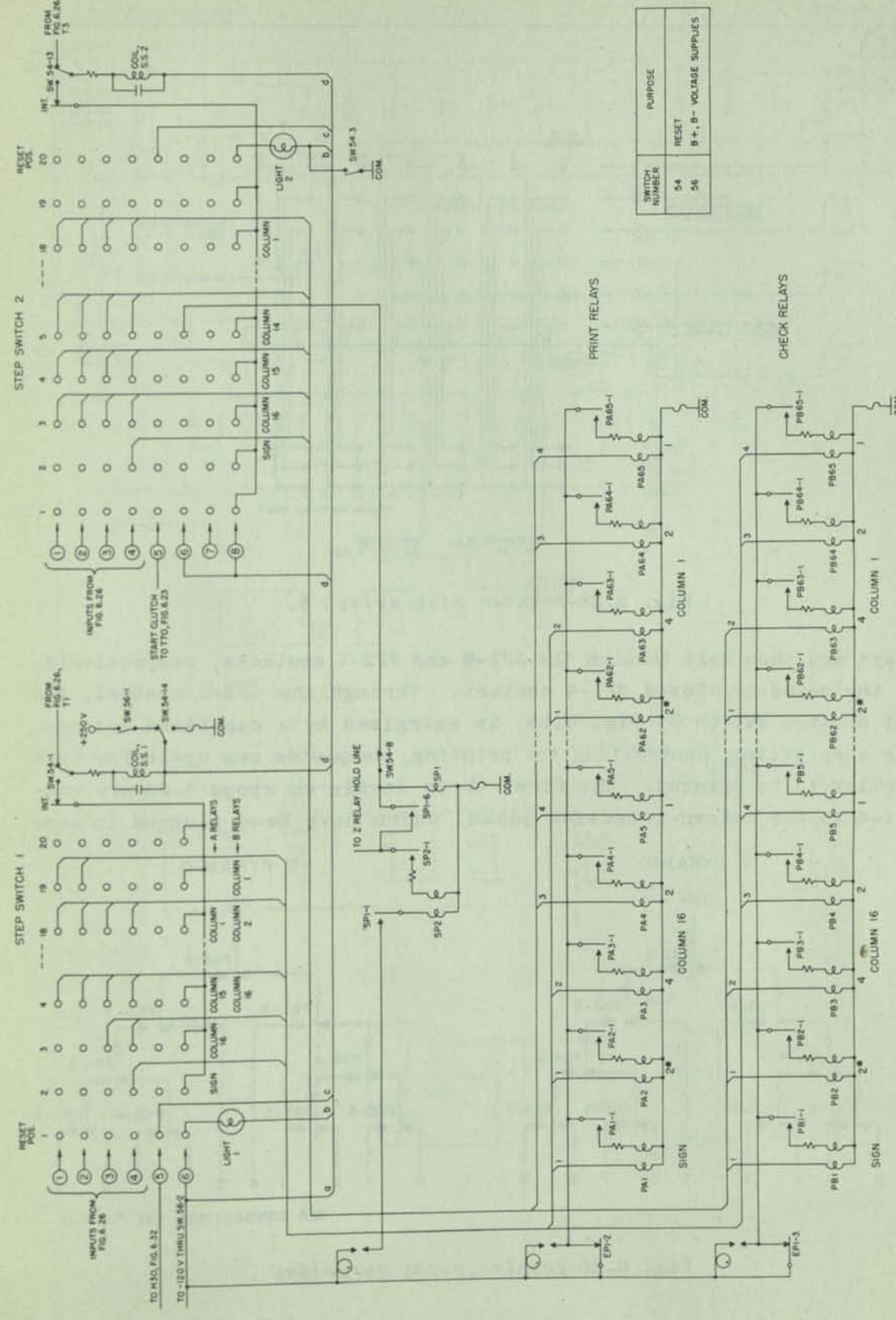


Fig. 6.27—Step switches 1 and 2, print and check relays.

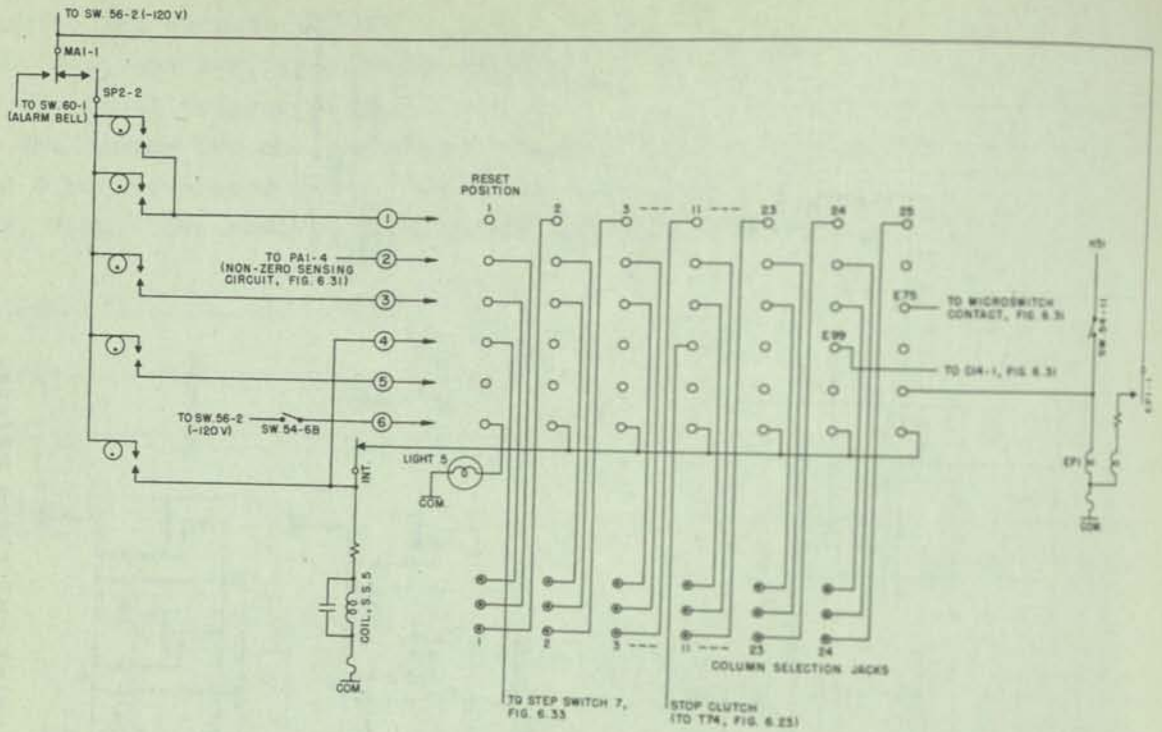


Fig. 6.28—*Printer step switch 5.*

relays are then held through the *SP*1-6 and *SP*2-1 contacts, respectively, and the normally closed *EP*1-4 contact. Through the *SP*2-2 contact, the coil of step switch 5, Fig. 6.28, is energized by a cam-timed voltage. This step switch, controlling the printing, completes one cycle for each quantity to be printed. The first three levels on steps 1-24 are connected to the column-selection jacks, which have been plugged in some

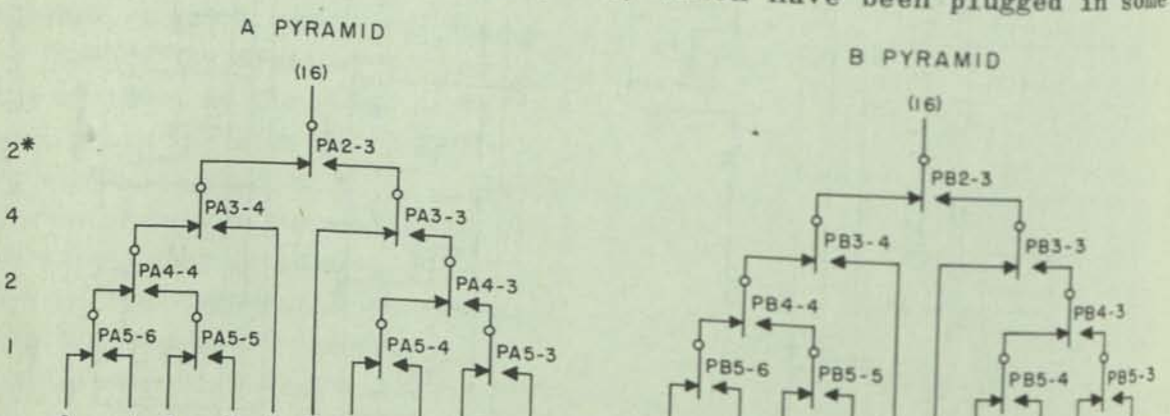


Fig. 6.29—*Printer relay pyramids.*

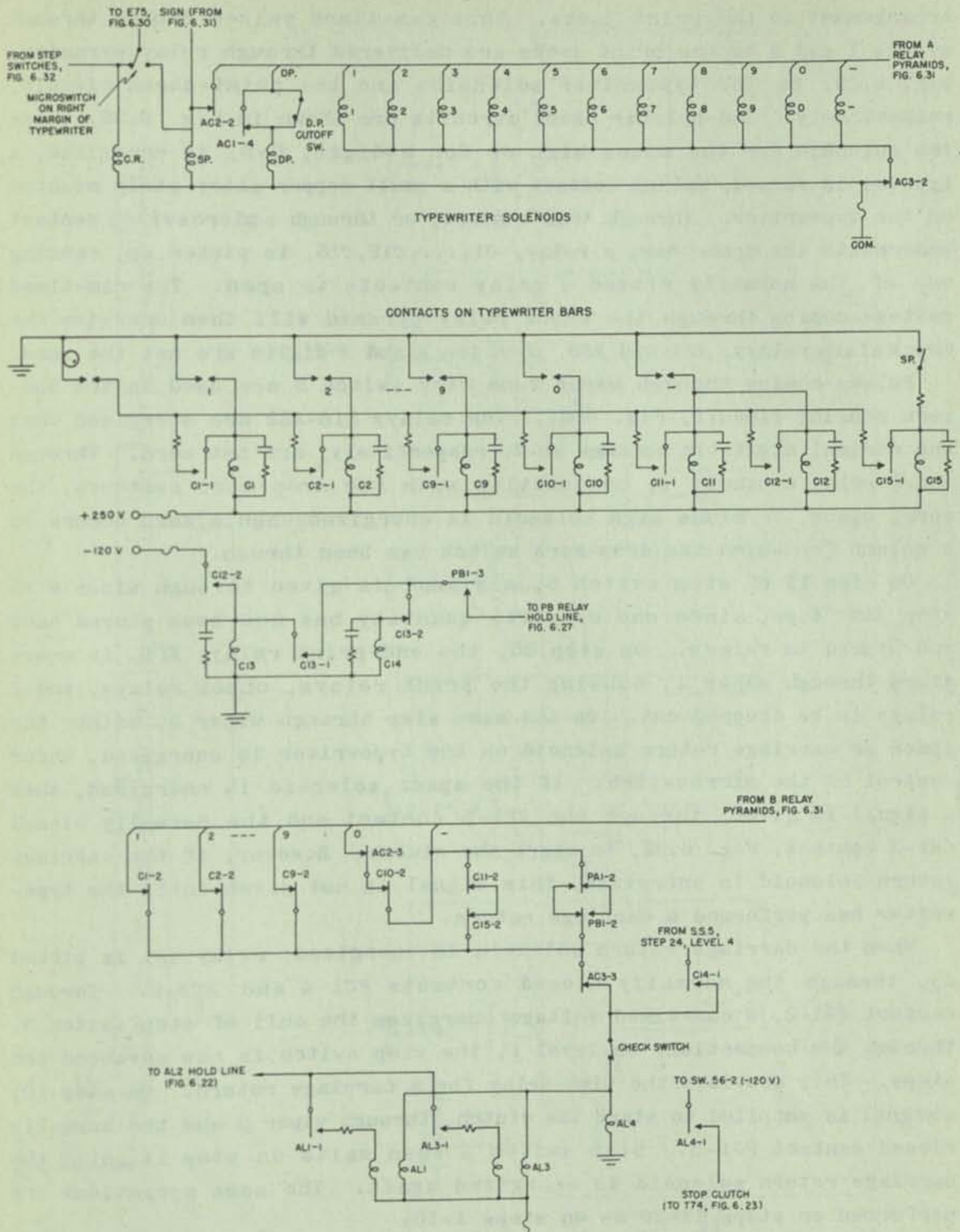


Fig. 6.30—*Printer check circuits.*

arrangement to the print jacks. Thus cam-timed pulses coming through wipers 1 and 3 to the print jacks are delivered through relay pyramids, Fig. 6.29, to the typewriter solenoids and the print-check circuit, respectively. The printer check circuits are shown in Fig. 6.30. When the solenoid for the minus sign or for a digit, 0-9, is energized, a type bar is raised, making contact with a small copper alloy strip mounted on the typewriter. Through this contact, or through a microswitch contact underneath the space bar, a relay, *C*1,...,*C*12,*C*15, is picked up, causing one of the normally closed *C* relay contacts to open. The cam-timed voltage coming through the check relay pyramid will then energize the check alarm relays, *AL*1 and *AL*3, when the *A* and *B* digits are not the same.

Pulses coming through wiper 2 on step switch 5 are used in the non-zero sensing circuit, Fig. 6.31. The relays *Z*16-*Z*22 are energized when the decimal digits in columns 16-2, respectively, are not zero. Through the *Z* relay contacts in conjunction with the drop-zero switches, the zero, space, or minus sign solenoid is energized when a zero occurs in a column for which the drop-zero switch has been thrown.

On step 11 of step switch 5, a signal is given through wiper 6 to stop the tape, since one complete quantity has now been played back and stored in relays. On step 25, the end-print relay, *EP*1, is energized through wiper 1, causing the print relays, check relays, and *Z* relays to be dropped out. On the same step through wiper 3, either the space or carriage return solenoid on the typewriter is energized, under control of the microswitch. If the space solenoid is energized, then a signal is given, through the *EP*1-5 contact and the normally closed *CR*1-3 contact, Fig. 6.32, to start the clutch. However, if the carriage return solenoid is energized, this signal is not given until the typewriter has performed a carriage return.

When the carriage return solenoid is energized, relay *CR*1 is picked up, through the normally closed contacts *PC*1-4 and *PC*3-2. Through contact *CR*1-2, a cam-timed voltage energizes the coil of step switch 3. Through the connections on level 1, the step switch is now advanced ten steps. This provides the time delay for a carriage return. On step 10, a signal is supplied to start the clutch, through wiper 3 and the normally closed contact *PC*1-3. Step switch 3 then waits on step 11 until the carriage return solenoid is energized again. The same operations are performed on steps 11-20 as on steps 1-10.

The same impulse which picks up relay *CR*1 energizes the coil of step switch 6, causing it to step once at the beginning of each carriage return delay. On step 8 of step switch 3, a cam-timed voltage is

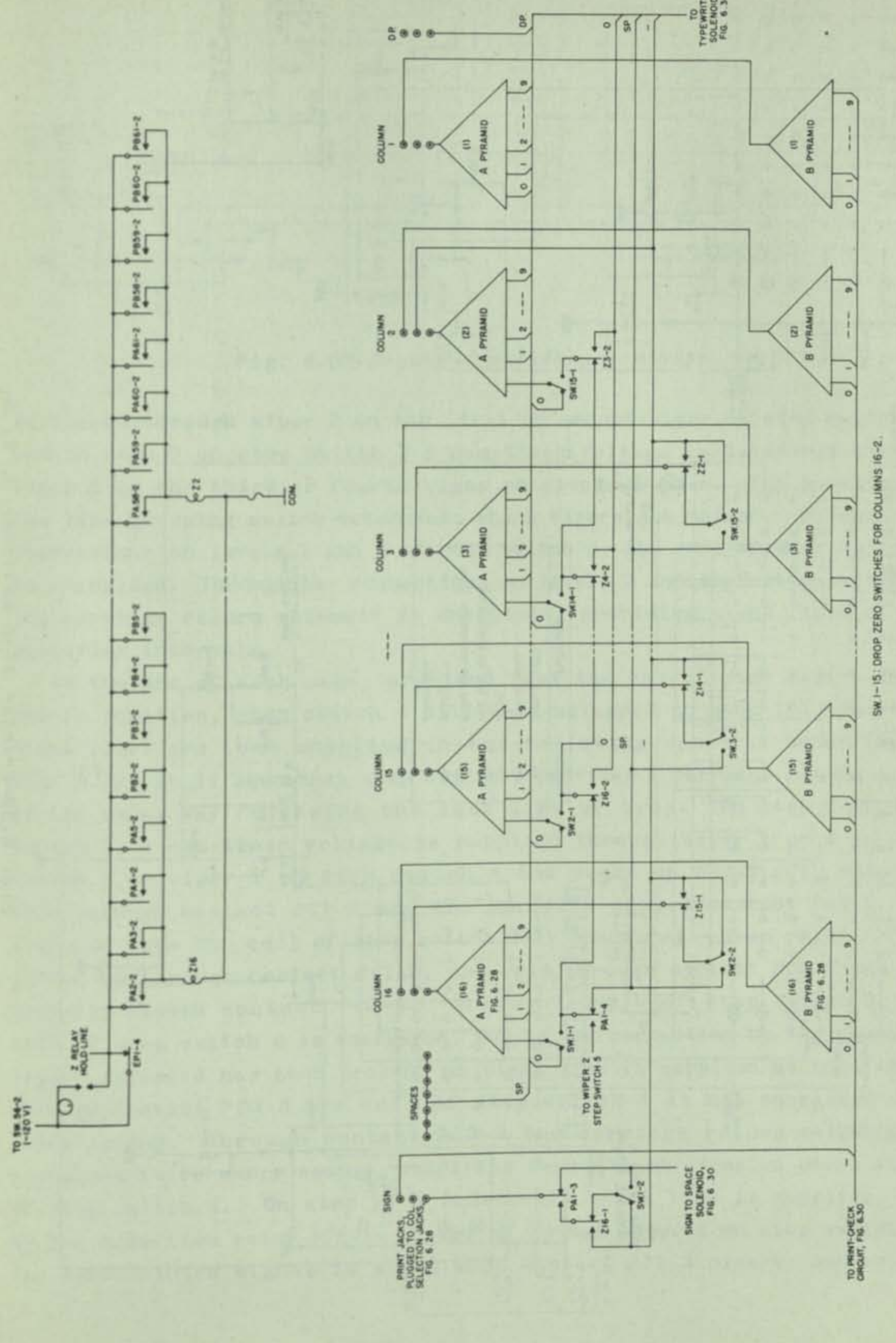


Fig. 6.31—*Non-zero sensing circuit.*

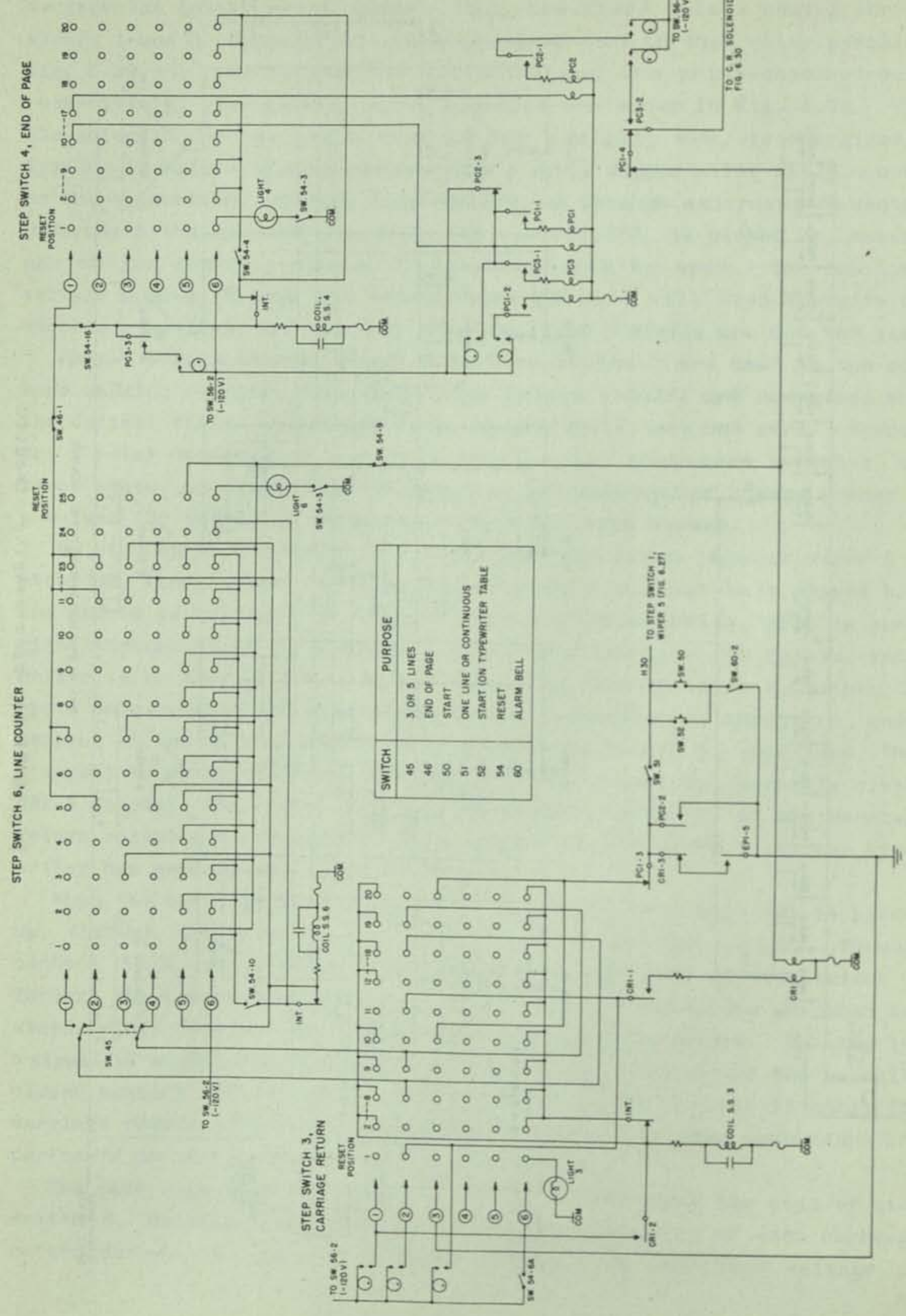


Fig. 6.32—Printer step switches 3, 4, 6.

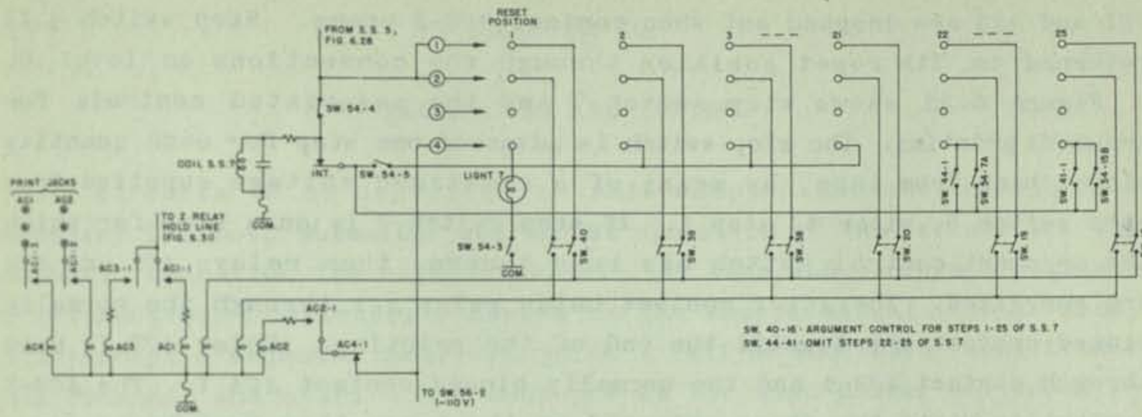


Fig. 6.33—*Argument printing controls.*

delivered through wiper 2 to the first or second wiper on step switch 6; and on step 9 of step switch 3 a cam-timed voltage is delivered through wiper 2 to the third or fourth wiper on step switch 6. The position of the line-grouping switch determines which wipers are pulsed. Through the connections on levels 1 and 2 of step switch 6, the coil of step switch 4 is energized. Through the connections on levels 3 and 4 of step switch 6, the carriage return solenoid is energized, providing blank lines at the specified intervals.

At the end of each page, provided that the end-of-page switch is in the on position, step switch 4 will have advanced to step 10. Thirteen blank lines are then supplied in the following manner. Referring to Fig. 6.32, it is seen that step switch 3 provides a carriage return delay in the usual way following the last line of type. On step 8 of step switch 3, a cam-timed voltage is supplied through wiper 1 or 2 on step switch 6 to wiper 1 on step switch 4 and picks up relay *PC*1, which is held through contact *PC*1-1 and the normally closed contact *PC*2-3. At the same time the coil of step switch 4 is energized. Then relay *PC*3 is picked up through contact *PC*1-2, and held through contact *PC*3-1 and the normally closed contact *PC*2-3. Through step 9 of step switch 3, the coil of step switch 6 is energized, but as the connection to the carriage return solenoid has been broken, no blank line is supplied at this time. Through contact *PC*3-3 the coil of step switch 4 is now energized once every second. Through contact *PC*3-2 the carriage return solenoid is energized twice every second, supplying twelve blank lines on steps 11-16 of step switch 4. On step 17 a thirteenth blank line is supplied, and at the same time relay *PC*2 is picked up through wiper 1 on step switch 4. The start clutch signal is given when contact *PC*2-2 closes; and relays

*PC*1 and *PC*3 are dropped out when contact *PC*2-3 opens. Step switch 4 is returned to its reset position through the connections on level 6.

Figure 6.33 shows step switch 7 and the associated controls for argument printing. The step switch is advanced one step for each quantity played back from tape, by means of a cam-timed voltage supplied from step switch 5, wiper 4, step 1. If step switch 7 is on a step for which the argument control switch has been thrown, then relays *AC*1 and *AC*2 are energized. The *AC*1-1 contact holds relay *AC*1 through the normally closed contact *EP*1-4 until the end of the printing. Relay *AC*2 is held through contact *AC*2-1 and the normally closed contact *AC*4-1. The *AC*2-2 contact prevents the space solenoid on the typewriter from being energized except by a plus sign. Thus as step switch 5 scans the print jacks preceding the first *AC* plug, a typewriter solenoid, space or minus, is energized only for the algebraic sign. When step switch 5 reaches the first *AC* plug, relay *AC*4 is energized through contact *AC*1-3. Relay *AC*2 is dropped out when contact *AC*4-1 opens. After the digits between the *AC* plugs have been printed, step switch 5 reaches the second *AC* plug and energizes relay *AC*3 through contact *AC*1-2. Contact *AC*3-2 opens and prevents further digits, except plugged spaces, from being printed.

CHAPTER VII

SEQUENCING AND CONTROL

The circuits to be described in this chapter provide the controls necessary for both automatic and manual operation of the calculator. In automatic operation the calculator may: 1) read and perform a sequence of instructions previously recorded on the instructional storage drum; 2) interrupt a sequence under control of a call or successful conditional call command, and start a new sequence at the line number delivered to the line number register; 3) introduce a wait when a read or record code is given while the tape read-record system is still engaged in a reading or recording operation; 4) supply codes under control of the *t* register and function sensing register; 5) stop under control of the check-stop register.

The manual controls enable the operator to do the following: 1) record instructions on the instructional storage drum from tape, either one line at a time or continuously, and to inspect the codes and line numbers in lights when desired; 2) cause the calculator to start either on a specified line or on the line following the last line performed, and having started, either to perform just one line or to continue in sequence; 3) read to lights either a specified line of coding or the last line performed by the calculator; 4) stop the calculator either immediately or when it arrives at a specified line; 5) cause the calculator to perform a command registered on the control panel either once or repeatedly, for testing; 6) read a command registered on the control panel either to lights or to the instructional storage drum.

A block diagram of the sequencing circuits is shown in Fig. 7.1. In normal operation a command is played back and advanced to trigger pair 1. A cycling pulse, derived from the main storage drums, then advances the *A*, operation, and *B* codes to the appropriate control pyramids. The *C* code is advanced to trigger pair 2, and delayed before delivery to the *C* control pyramid. One cycle later the next command is played back from the drum and advanced in a similar fashion.

Whenever a command is given to call a particular line of coding, the cycling circuits stop the pulses used to read out commands. The serial numbers played back from the drum are compared in the matching circuit with the number stored in the line number register. When a match is obtained, the cycling pulses are restored. The desired line of coding will then be advanced to the control pyramids and operation is resumed.

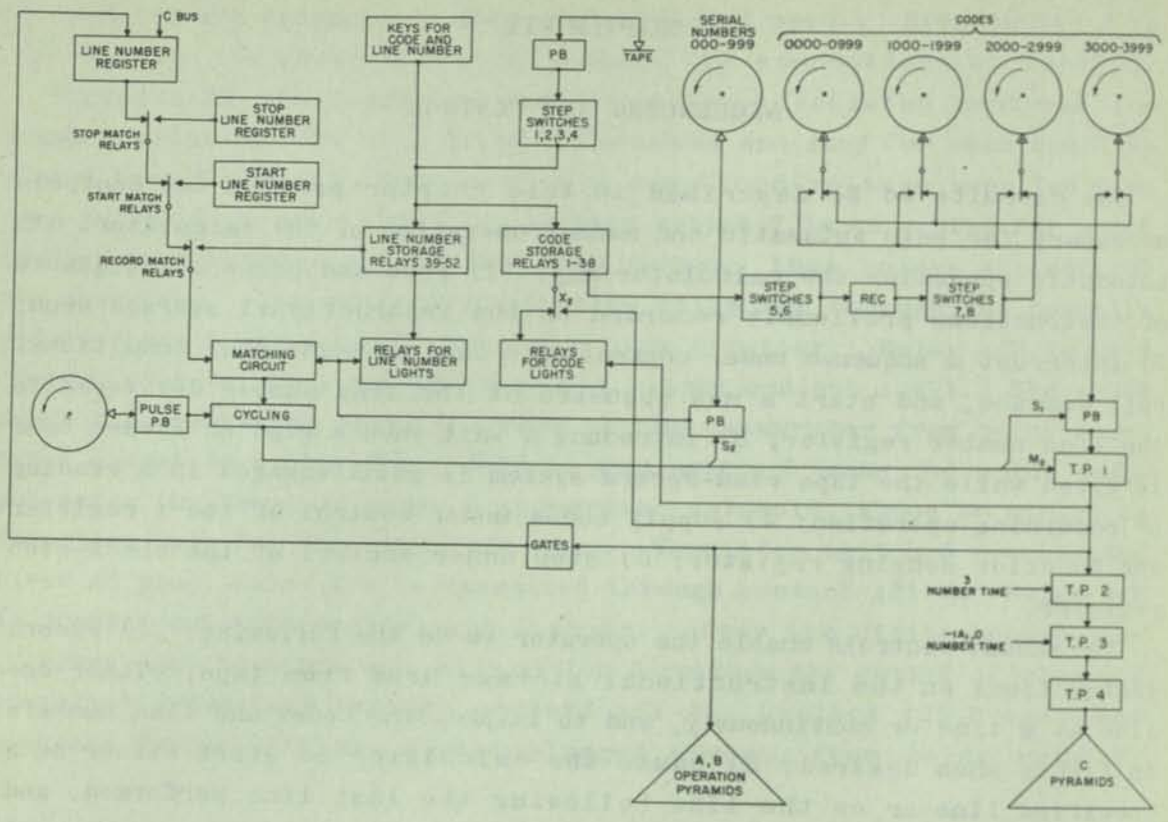


Fig. 7.1—*Block diagram of sequencing circuits.*

The matching circuit is used in a similar way to find a particular line on which the calculator is to start, stop, or on which a command is to be recorded.

The instructional storage drum is shown in Plate V and Fig. 7.2. This drum is mounted vertically and driven at a speed of about 1725 rpm. The drum is 50 inches in circumference and 32.8 inches in length. Digits are recorded with a density of about twenty per inch around the periphery of the drum, allowing 1001 digits to be recorded in each channel.

One digit of the 1001 digits recorded in a channel is non-informative, while the remainder represent coded instructions. Advance pulses are supplied to the sequence playbacks so that every 125th digit is read out. In this manner all the digits recorded on a channel may be read out in 125 revolutions of the drum. The positions of the serial numbers which identify the coded information are shown in Fig. 7.3. The binary channels on the instructional storage drum are allocated as shown in Table 7.1.

The number storage drums and the instructional storage drum are driven by separate motors. Compensation for the lack of synchronism between

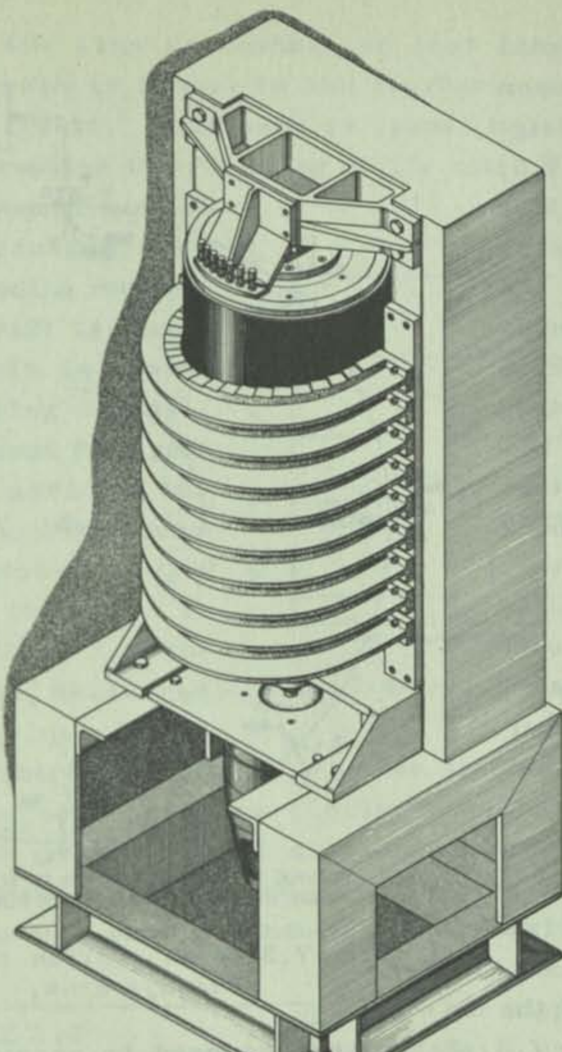


Fig. 7.2—*Instructional storage drum.*

these drums is made in the electronic circuits associated with the sequencing system. Two sets of pulses are used to control these circuits. The pulses M_1, M_2, M_3, and M_4, occurring at the times indicated in Table 7.2, are derived from the main storage drums. The pulses S_1, S_2, S_3, S_4, and S_5, timed as shown in Fig. 7.4, are obtained from five channels on the instructional storage drums.

An additional pulse channel, indicated in Fig. 7.5, is used to control the read-out of commands and serial numbers from the playback circuits. Of the three pole pieces provided on the channel, one is for recording 1's, the second is for playback, and the third is for recording 0's. In normal operation, a 1 digit is recorded, and played back 125 sequence pulse times later by an S_1 pulse.

When the go signal is supplied to open the pentode gate designated $S99$ in Fig. 7.5, each 1 digit played back causes the following to take place: 1) the circuit which records 0's is started to erase the 1 digit; 2) another 1 is recorded on the cycling channel so that it may be played back 125 digit times later, causing the next coded command to be read out; 3) an S_2 pulse is gated to the line number playback circuits, and an S_1 pulse to the code playback circuits, in order to read out a command and its line number.

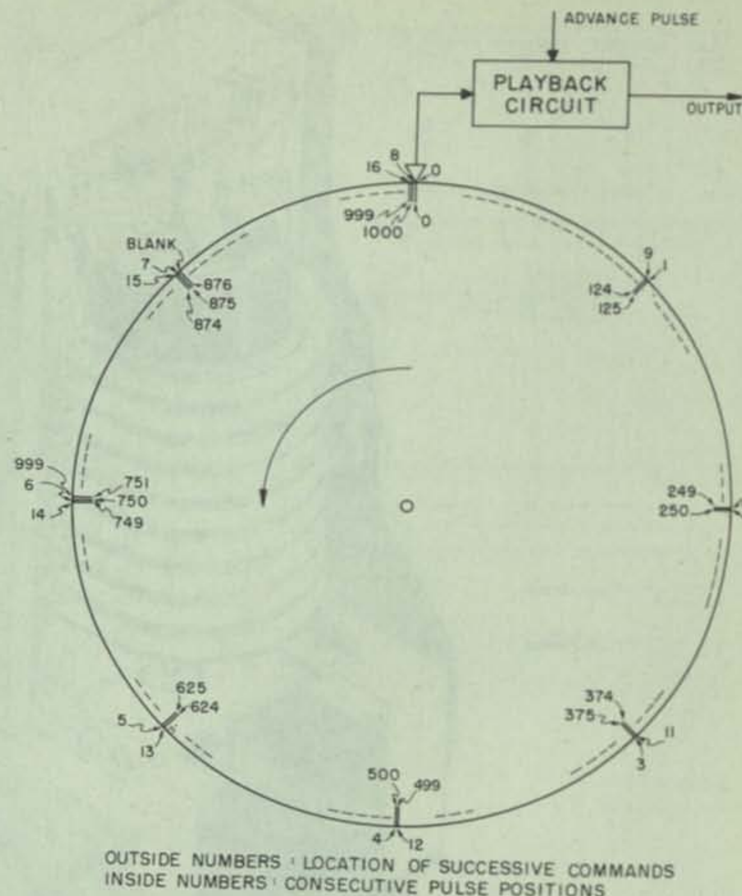


Fig. 7.3—*Cross section of instructional storage drum.*

As indicated in the lower left of Fig. 7.5, S_4 pulses are used to record 0's in the erase circuit. In order to insure complete erasure the S_4 pulses are accurately timed relative to the S_3 pulses used to record 1's. Thus a 1 digit on the cycling channel is normally erased 121 digit times after being played back. If the go signal is removed, however, the erase circuit will be stopped allowing the last 1 digit recorded to remain on the channel. When the go signal is restored this digit may be used to start

TABLE 7.1

Allocation of channels on instructional storage drum

Information	Number of Channels
Cycling pulses	6
Serial Numbers 000-999	12
Commands 0000-0999	38
Commands 1000-1999	38
Commands 2000-2999	38
Commands 3000-3999	38

the calculator on the line following the last line executed.

A regeneration path is formed by the playback, gate *S*120, gate *S*98, and the 1 record circuit. This path is opened by *S*120 whenever a call occurs, thus interrupting the recording of 1's until a match is obtained.

In normal operation, when the calculator is executing successive lines of coding recorded on the drum, gate *S*121 is opened every time a 1 digit is played back from the cycling channel. The S_2 pulses obtained from gate *S*121 are used to advance the line numbers from the line number playbacks to associated trigger pair storage. As indicated in the block diagram, Fig. 7.1, the line numbers may be delivered either to the matching circuit or to the relays controlling the line number display lights. In case a command is to be recorded on a particular line of the drum, however, gate *S*121 is opened by the record match signal obtained from a relay contact. In this manner continuous S_2 pulses are supplied to the line number playbacks so that each line number may be examined in the matching circuit. Continuous S_2 pulses are also supplied by gate *S*121 in case a particular line of coding is called. In this case a continuous high voltage is applied to the grid of gate *S*121 by trigger pair 2. This trigger pair may be tripped by an M_3 pulse when a call is made automatically by a coded instruction. It may also be tripped by a differentiated pulse from a relay contact which is closed when a manual call is made as in starting the calculator or reading information to lights.

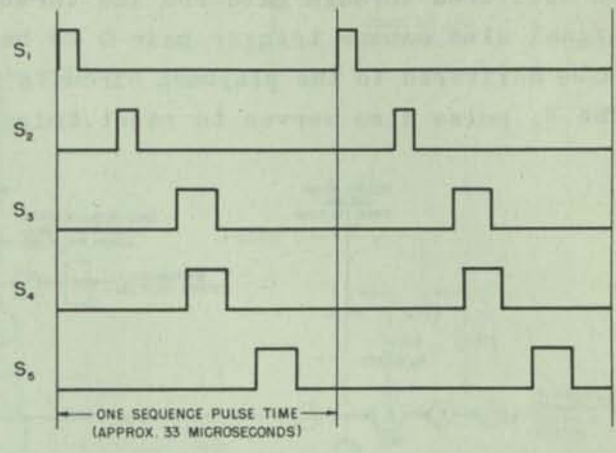


Fig. 7.4—*Relative timing of sequence cycling pulses.*

TABLE 7.2			
Timing of M_1, M_2, M_3, and M_4 pulses			
	Type of Pulse	Digit Time	Number Time
M_1	A_2	18	4
M_2	A_1	-1	blank
M_3	R	-1	blank
M_4	A_2	-1	blank

The output from trigger pair 2 also prevents a 1 from being recorded on the cycling channel until the desired line has been found. This is accomplished by closing the gate *S*120 and opening the gate *S*119. Then a played back 1 digit trips trigger pair 3. An S_1 pulse, gated at 1, 2, 3, or 4 number time, is read through trigger pair 3 to trip trigger pair 4. As soon as a match signal is obtained from the matching circuit, indicating that the desired line has been found, a signal to record a 1 is delivered through gate $S96$ and through the combining gate $S98$. This signal also causes trigger pair 5 to be tripped, permitting an S_1 pulse to be delivered to the playback circuits for the thirty-eight code digits. The S_1 pulse also serves to reset trigger pairs 2 and 3, and is used in

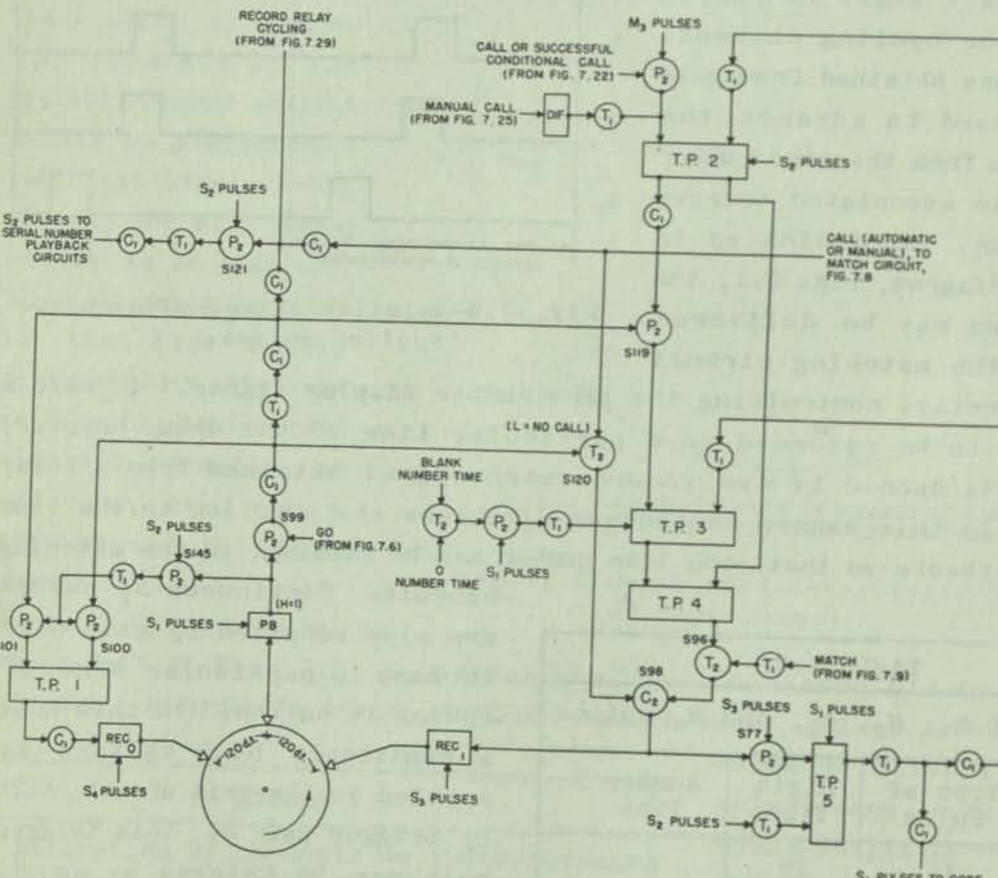


Fig. 7.5—*Cycling pulse control circuit.*

the circuits shown in Fig. 7.6. When trigger pair 2 is reset the next S_2 pulse resets trigger pair 4, thus restoring the circuit to its normal condition.

Referring to Fig. 7.6, it is seen that a gated S_1 pulse from the circuit shown in Fig. 7.5 resets trigger pair 6, cutting off the go signal. The first following M_1 pulse resets trigger pair 7, the output of which is differentiated and used to trip trigger pair 8. The following

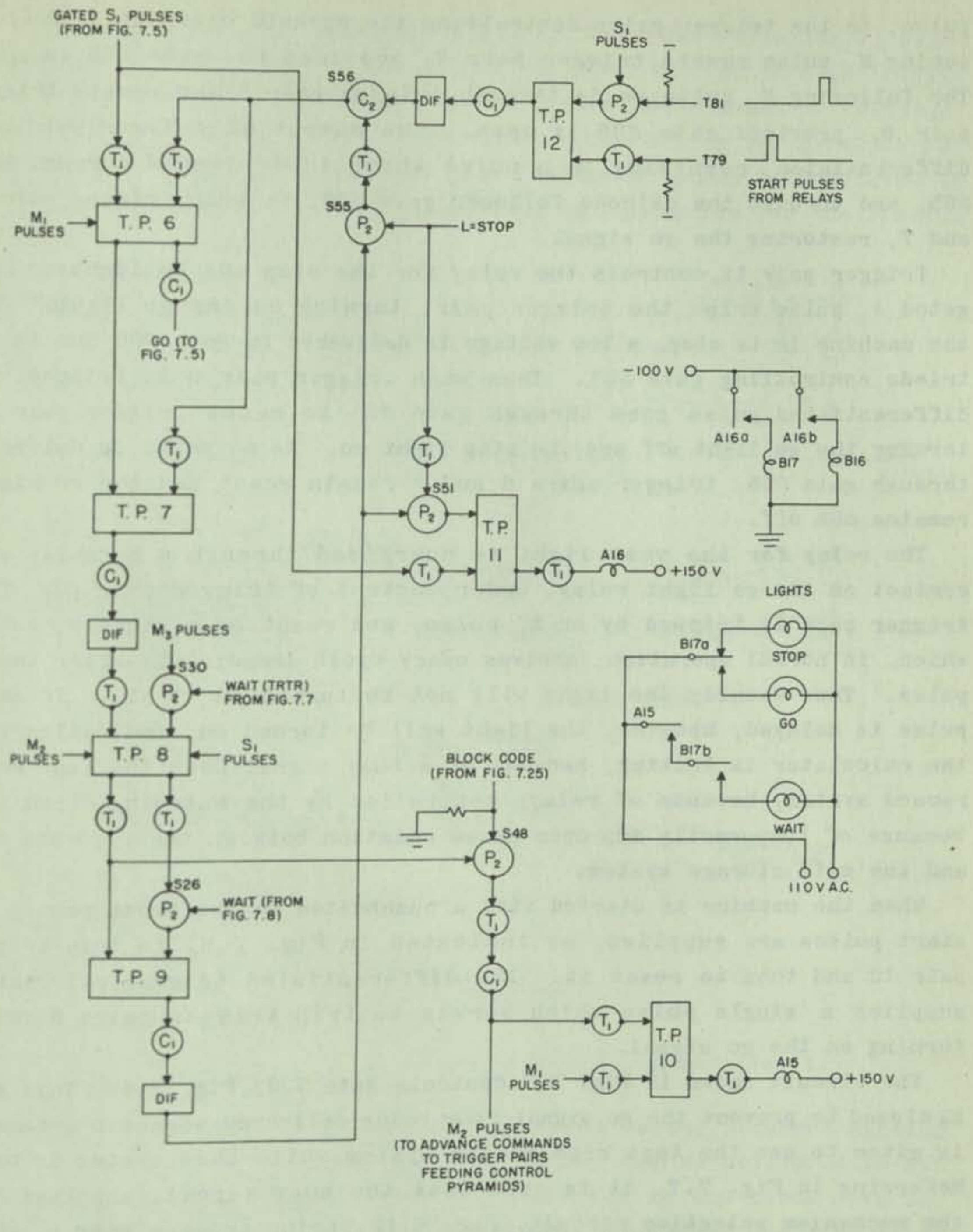


Fig. 7.6—*Sequence cycling circuit.*

M_2 pulse reads through trigger pair 8 and serves two purposes: 1) it trips trigger pair 9; 2) provided the gate $S48$ is open, it arrives at terminal $T22$ and advances a command, read out by the previous gated S_1

pulse, to the trigger pairs controlling the pyramid circuits. The following M_3 pulse resets trigger pair 8, provided the gate $S30$ is open. The following S_1 pulse reads through trigger pair 8 and resets trigger pair 9, provided gate $S26$ is open. The output of trigger pair 9 is differentiated, resulting in a pulse which is delivered through gate $S55$, and through the cathode follower gate $S56$, to trip trigger pairs 6 and 7, restoring the go signal.

Trigger pair 11 controls the relay for the stop and go lights. Each gated S_1 pulse trips the trigger pair, turning on the go light. When the machine is to stop, a low voltage is delivered to gate $S55$ and to the triode controlling gate $S51$. Then when trigger pair 9 is tripped, the differentiated pulse goes through gate $S51$ to reset trigger pair 11, turning the go light off and the stop light on. As no pulse is delivered through gate $S55$, trigger pairs 6 and 7 remain reset and the go signal remains cut off.

The relay for the wait light is energized through a normally open contact on the go light relay, under control of trigger pair 10. This trigger pair is tripped by an M_1 pulse, and reset by a gated M_2 pulse, which, in normal operation, arrives every cycle immediately after the M_1 pulse. Thus usually the light will not be turned on at all. If an M_2 pulse is delayed, however, the light will be turned on, indicating that the calculator is waiting, because of a busy signal from the tape read-record system, because of relays controlled by the matching circuit or because of temporarily improper phase relation between the sequence drum and the main storage system.

When the machine is started with a pushbutton on the front panel, two start pulses are supplied, as indicated in Fig. 7.6, to trip trigger pair 12 and then to reset it. The differentiated trigger pair output supplies a single pulse which serves to trip trigger pairs 6 and 7, turning on the go signal.

The circuit shown in Fig. 7.7 controls gate $S30$, Fig. 7.6. This gate is closed to prevent the go signal from being delivered whenever a command is given to use the tape read-record system while this system is busy. Referring to Fig. 7.7, it is seen that the busy signal, supplied from the mechanism selection circuit, Fig. 6.12, trips trigger pair a, which will be reset by an M_1 pulse as soon as the busy signal is removed. As may be seen from the list of codes, Appendix II, the commands to be blocked while the tape read-record system is busy are $A10$-$A18$, $B02*$, and $C02*$. The corresponding voltages from the control pyramids are combined as shown in Fig. 7.7 to deliver a high voltage through the trigger pair, supplying the signal to close gate $S30$, Fig. 7.6. The following M_3 pulse trips trigger pair b, permitting the next M_4 pulse to read through trigger pair b and reset the code trigger pairs before the command is executed.

In normal operation, the trigger pairs controlling the pyramids for the A, B, and operation codes are reset by the M_1 pulse as indicated in Fig. 7.7. The trigger pairs controlling the pyramids for the C codes are reset by a voltage lasting through 1 number time, except in a multiplication. In this case the C code determining where the product is to be stored must be delayed until the product has been computed. The delay is provided by the m_1 voltage from the multiplication cycling circuits.

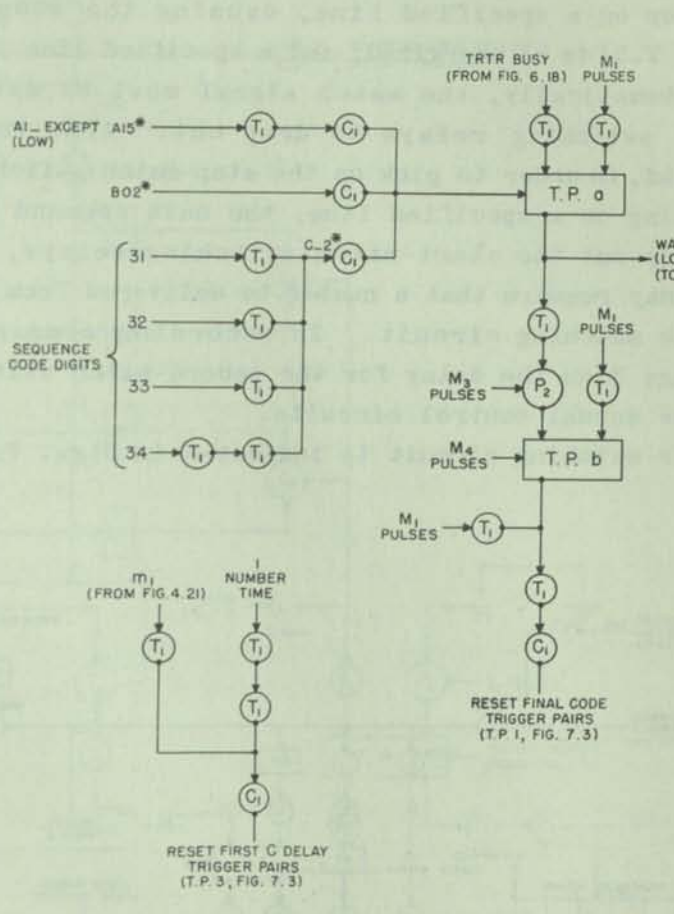


Fig. 7.7—*Wait signal circuit.*

The gate $S26$, Fig. 7.6, is controlled by the matching circuit, which will now be described. This circuit compares the line numbers of commands being read from the drum with the number of a specified command, on which the machine is to start, continue computing, stop, or record a new command. When the two numbers agree, a match signal is delivered to the sequence cycling circuits, Fig. 7.5. The specified line number may be delivered to the matching circuit from several sources indicated in the block diagram, Fig. 7.3. These are the line number register, the start and stop match registers, and the line number storage relays. The latter store information which either has been played back from an instructional tape or has been designated in the *select line number* register on the front panel.

Due to the fact that relays are used to connect the matching circuit to one of the possible inputs and to select one of the four channels of commands on the instructional storage drum, delays must be included in the matching circuit to make sure that all relays have been properly positioned before a match signal is given and, in some cases, before the next line of coding is read. For example, whenever a shift is made from one of the four channels of commands to another, a delay is required while the selection relays are positioned. If the machine has been set to stop on a specified line, causing the stop-match switching relays (Fig. 7.3) to be energized, and a specified line is called, either manually or automatically, the match signal must be delayed to allow the stop-match switching relays to drop out. Also the next command must be delayed, in order to pick up the stop-match switching relays again. After starting on a specified line, the next command must be delayed in order to drop out the start-match switching relays, since the first command read may require that a number be delivered from the line number register to the matching circuit. In recording commands on the instructional storage drum the delay for the record-match switching relays is provided in the manual control circuits.

The matching circuit is indicated in Figs. 7.8 and 7.9. At the right

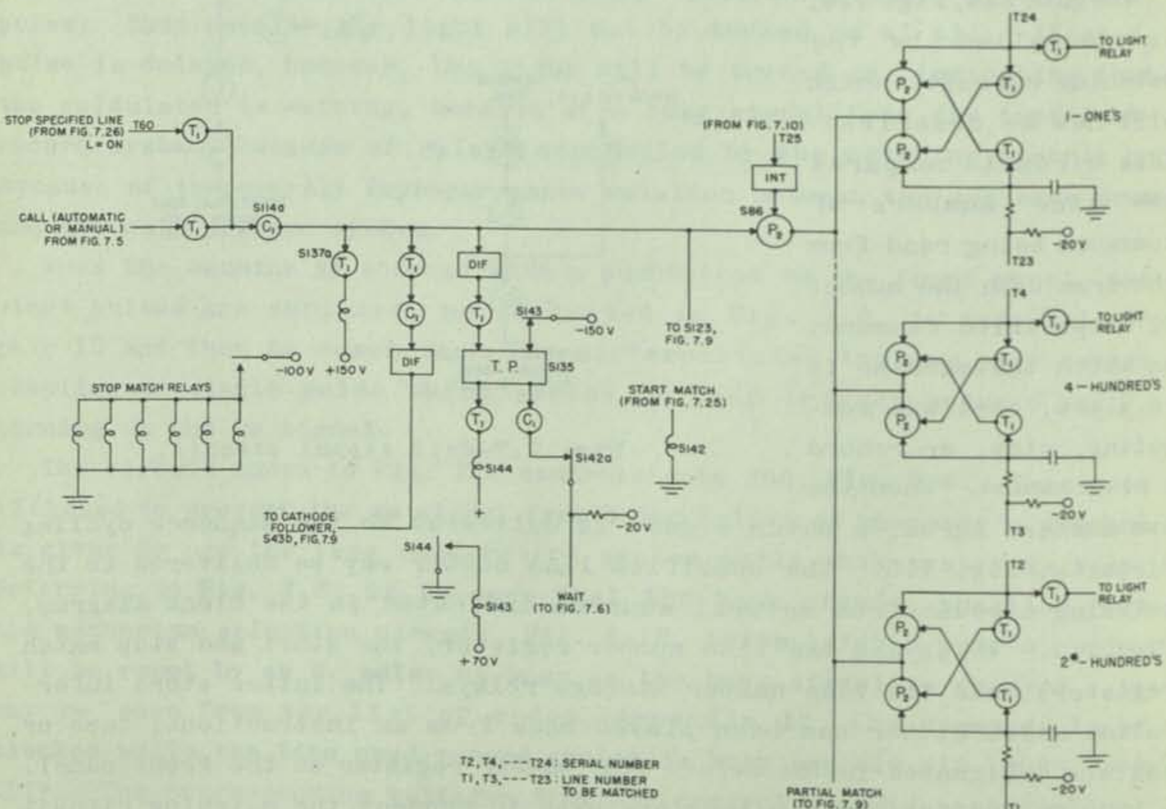


Fig. 7.8—*Matching circuit.*

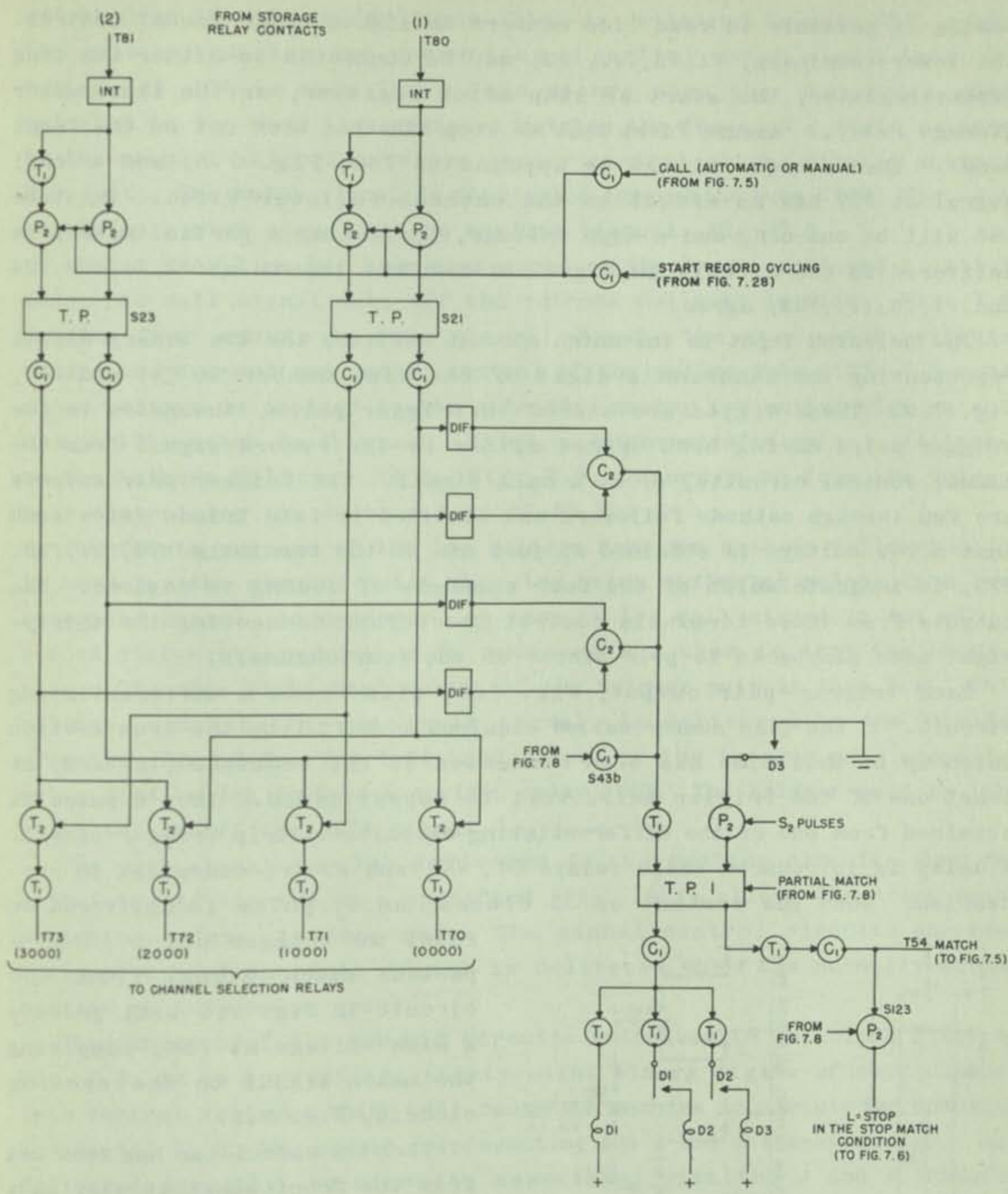


Fig. 7.9—*Match gating and channel selection circuit.*

of Fig. 7.8 are the comparison circuits for the three lower digits of the entering line numbers. The inputs to the upper terminals, *T*2, *T*4, ..., *T*24, are from the output trigger pairs of the line number playbacks. A puller tube connected to each of these inputs controls a light relay, making it possible to read line numbers to lights under manual control. The lower terminals, *T*1,*T*3,...,*T*23, may be connected to either the line number register, the start or stop match register, or the line number storage relays. Assume first that no stop line has been set on the front panel. Then a high voltage is applied to *T*60, Fig. 7.8, and a call signal at *T*77 has no effect on the cathode follower *S*114*a*. The tube *S*86 will be cut off, and a high voltage, indicating a partial match, is delivered to the circuit in Fig. 7.9 when the inputs at *T*2,*T*4,...,*T*24 and *T*1,*T*3,...,*T*23, agree.

The selected input to the match circuit delivers the two binary digits representing the thousand's digit of the line number to *T*81 and *T*80, Fig. 7.9. These digits are stored in trigger pairs, the gates to the trigger pairs having been opened either by the record signal from the manual control circuits, or by a call signal. The trigger pair outputs are fed through cathode followers and combined in twin triode gates such that a low voltage is obtained at just one of the terminals *T*70,*T*71,*T*72, *T*73, to indicate which of the four channels of coding is desired. The outputs from these terminals control the relays connecting the thirty-eight code playbacks to pole pieces on the four channels.

Each trigger pair output, Fig. 7.9, also feeds a differentiating circuit. If the line number called requires a shift from the drum section which up to this time has been connected to the sequence playback, at least one of the trigger pairs will be repositioned. Then a pulse is obtained from one of the differentiating circuits to trip trigger pair 1. A delay is introduced while relays *D*1, *D*2, and *D*3 are energized in succession. When the contact on *D*3 closes, an S_2 pulse is delivered to reset the trigger pair. Then a partial match signal from the circuit in Fig. 7.8 will produce a high voltage at *T*54, supplying the match signal to the cycling circuit, Fig. 7.5.

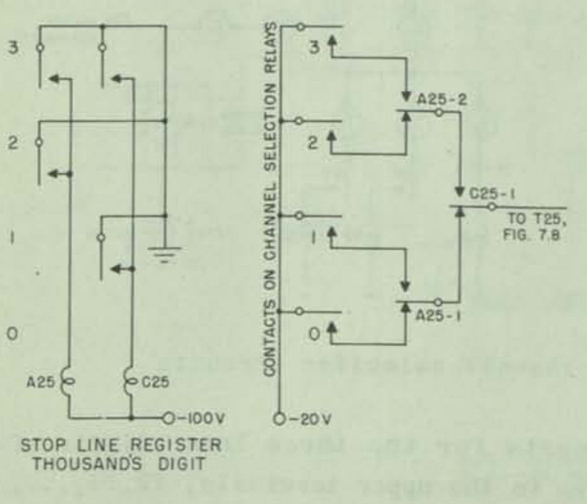


Fig.7.10—*Stop line-thousand's digit.*

When the calculator has been set from the front panel to stop on a particular line, the output of the cathode follower *S*114*a*, Fig. 7.8, is high. The stop match relays are energized by the puller tube *S*137*a* connecting the stop line register to the matching circuit. The high output from the cathode follower *S*114*a* is delivered to gate *S*86, making the gate output low and preventing a partial match signal until the voltage at *T*25 is made low. This occurs when the thousand's digit selected in the stop line register matches the thousand's digit selected by the channel selection relays, under control of the circuit shown in Fig. 7.10. The match signal is then gated through the tube *S*123, Fig. 7.9, supplying a stop signal to the cycling circuit, Fig. 7.6.

When a stop line has been registered on the front panel and a call is made, the call signal cuts off the cathode follower in *S*114*a*, Fig. 7.8, causing four events to take place. First, the stop match switching relays are dropped out, under control of the puller tube *S*137*a*. Second, a low voltage is applied to the gate *S*86, making its output high as soon as a partial match is obtained. Third, a differentiated pulse is delivered to the cathode follower *S*43*b*, Fig. 7.9, in order to trip the trigger pair controlling the delay relays *D*1,*D*2,*D*3.

The fourth purpose of the low voltage from the cathode follower *S*114*a* is to delay the command until the stop match switching relays have been energized again. Accordingly, as soon as the call signal is cut off, a second differentiated pulse is produced and used to trip the trigger pair *S*135. The right-hand output of the trigger pair is then low. This voltage, constituting the wait signal, is delivered to the cycling circuits, Fig. 7.6. The left-hand output of the trigger pair energizes relay *S*144, which in turn energizes relay *S*143. The trigger pair is then reset to cut off the wait signal.

The wait signal is also delivered to the cycling circuits when the calculator is started on a specified line. This allows the start match switching relays to drop out. The manual control circuits energize relay *S*142, and the wait signal is delivered when the normally closed contact on this relay opens.

The function of the pyramid circuits indicated in the block diagram, Fig. 7.3, is to convert the thirty-eight binary digits of each command into control voltages which will cause the machine to execute the command. The digits 1, 2, 17, and 18, representing the *A* and *B* transfer signs, are delivered directly to circuits associated with the *A* and *B* transfer channels (see Chapter IV). The digits 19-28, representing the *C* channel and number time, are delayed in trigger pairs one cycle and one number time, as explained in Chapter III, before delivery to the trigger pairs controlling the *C* pyramids.

Figure 7.11 shows the pyramid for the four digits, 13, 14, 15, and 16, used for operation codes. Special controls are introduced by codes

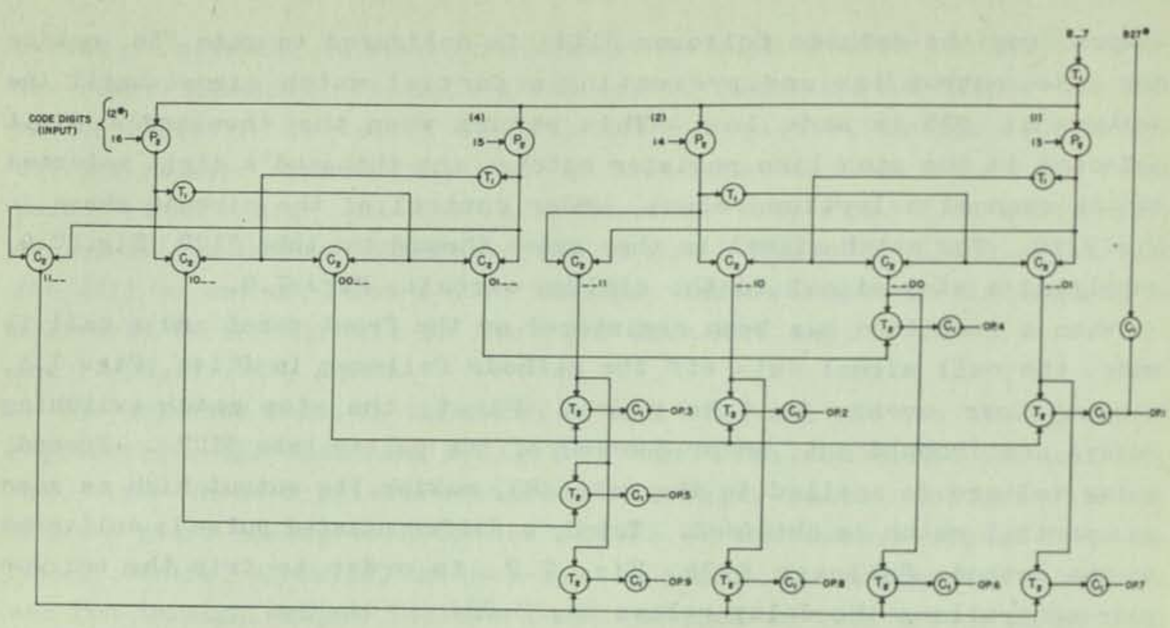


Fig. 7.11—*Operation-code pyramid.*

*B*_7* (that is, any *B* code whose second digit is 7*) and *B*27*. The call, conditional call, and external transfer commands, having codes *B*_7*, use code digits 3-16 to represent a number in the range 0000-3999. In these cases no control voltage derived from digits 13-16 should be delivered to the calculator to initiate an operation. As will be seen later, provisions are also made in these cases to prevent digits 3-12, usually used for the *A* channel and number time, from operating through the *A* pyramids. In an external transfer, the number coded on digits 3-16 is delivered to the *B* transfer channel, added to zeros, and made available on the *C* bus. Therefore, operation code 1, for an addition, is supplied with the code *B*27* independent of the operation code pyramid. As will be seen later, the code *A*15*, to record zeros on the *A* transfer channel, is also supplied for the code *B*27* independent of the *A* pyramids.

The circuits shown in Figs. 7.12, 7.13, and 7.14 are provided in triplicate, for the digits representing the *A*, *B*, and *C* channel number and number-time codes. In the number-time pyramid, Fig. 7.12, the output is a high voltage during 0, 1, 2, 3, or 4 number time, corresponding to number-time codes 0 or 6, 1 or 7, 2 or 8, 3 or 9, or 4 or 5, respectively. In the *C* number-time pyramid, the number-time voltages are delayed to account for the displacement of one number time between record and playback pole pieces.

As may be seen from Fig. 7.13, the inputs representing the unit's digit of a channel number produce a high voltage at one of sixteen output terminals. The outputs, *A*_0,...,*A*_9,*B*_0,...,*B*_9, from the *A* and *B* circuits control the first level of the playback selection gates, while the outputs *C*_0,...,*C*_9, from the *C* circuit control the first level distribution gates for record pulses.

The components of the ten's digit of a channel number are delivered to the circuit in Fig. 7.14. A low voltage output *A*0_,*B*0_,...,*C*2_, is produced for use in the ten's-unit's combination circuits, Fig. 7.15. The circuit in Fig. 7.14 also produces outputs controlling the second level playback or record gates, *A*0_(*L*),*B*0_(*L*),...,*C*3_(*L*). These outputs are determined not only by the ten's digit of the channel number but also by a voltage indicating the drum phase and whether the selected number time is in the range 0-4 or 5-9.

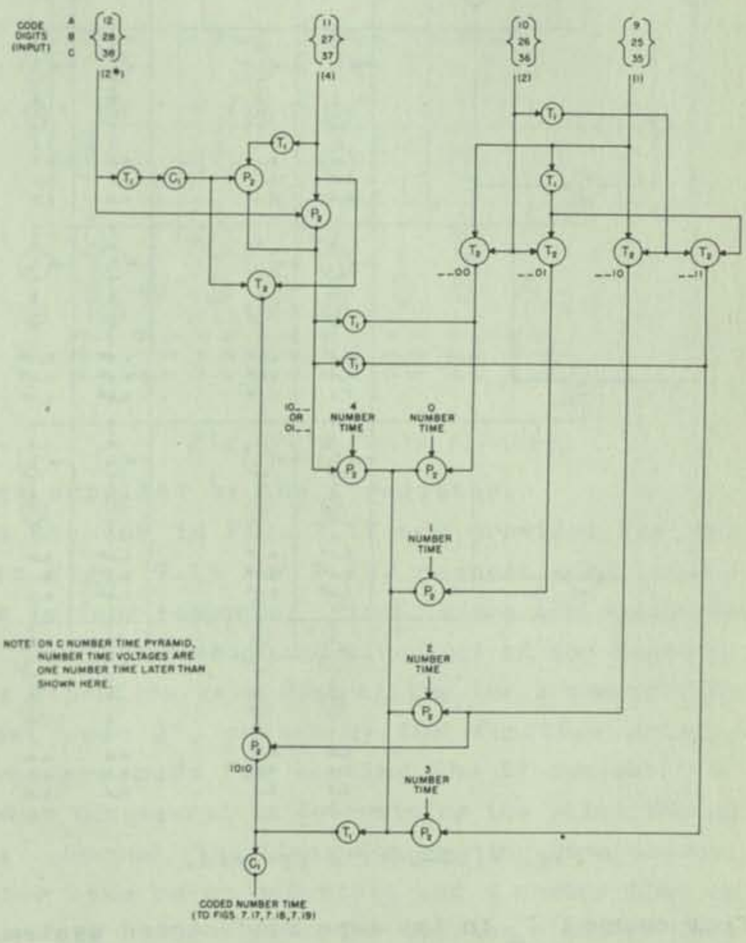


Fig. 7.12—*Number time pyramid.*

When a code *B*_7* is given and the number coded on sequence digits 3-16 is 00_,...,19_, some provision must be made to prevent a command *A*00-*A*19 from being delivered to the calculator. This is accomplished by the cathode follower gate shown in Fig. 7.14, which feeds the *A* ten's pyramid and insures that for a *B*_7* code, the *A* ten's pyramid will receive a digit 2 or 3 as input.

The circuits shown in Figs. 7.15 and 7.16 combine the ten's and unit's digits of the *A*, *B*, and *C* channel numbers. The outputs are used as required in controlling the calculator or as inputs to further pyramid circuits. The code *B*27* supplies the code *A*15* independent of the *A* code digits as described above. Both the commands *B*27* and *A*15* are

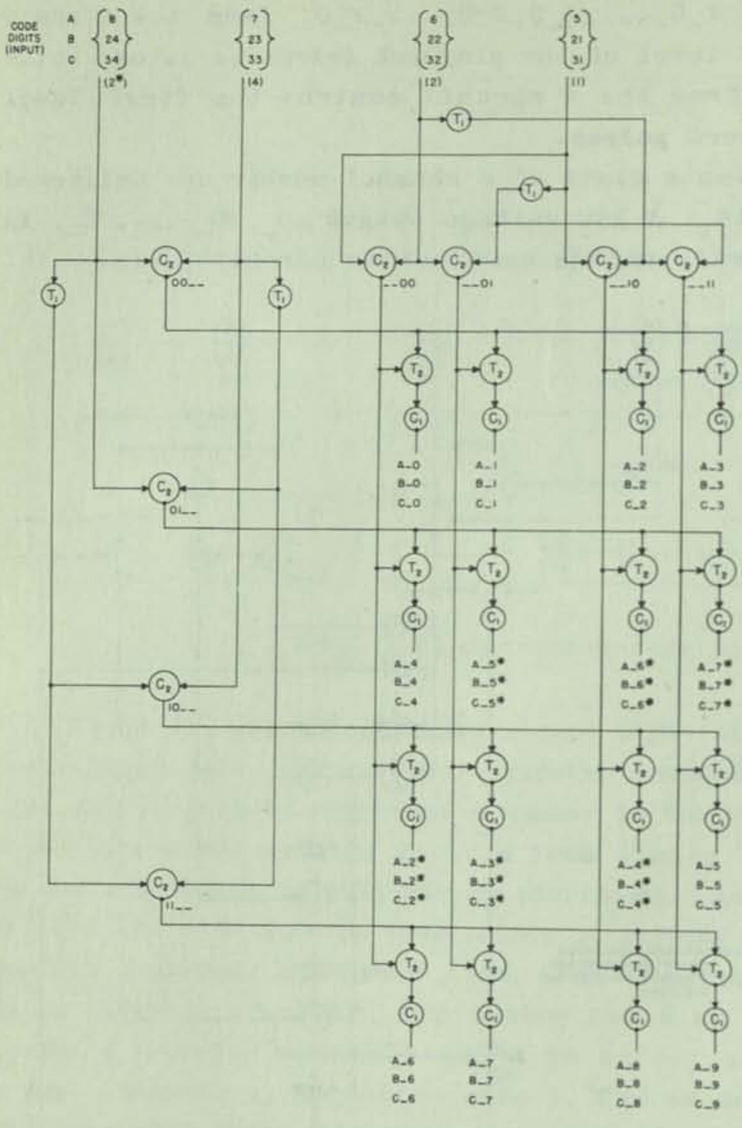


Fig. 7.13—*Unit's pyramid.*

delayed until the start of 0 number time since both require the use of the adder and, without the delay, would interfere with an addition coded on the previous line.

The circuit shown in Fig. 7.17 provides voltages controlling the number-time gate on the *A* transfer channel, indicating the drum phase as required in the *A* circuit of Fig. 7.14, and controlling the *A* bus gates for the α, β, and manual constant register channels. First consider the output controlling the *A* number-time gate. This will be a low voltage during either the number time represented by code digits 9-12, the number time selected by the *i* register, or a number time suitable for reading from channel I_a in the tape read-record system. Provisions are made to abrogate the *A* number time in a number of cases, namely, for code *B*_7*, *A*00, *A*02, *A*03, *A*05, *A*06, and *A*1_ except *A*10. This prevents any of these commands, which do not use the *A* transfer channel, from destroying the last quantity recorded on the *A* transfer channel.

The second output of the circuit of Fig. 7.17 is high on an even cycle if the number time selected is between 5 and 9 and high on an odd cycle if the number time selected is between 0 and 4. This voltage is used as shown in Fig. 7.14 to select the upper or the lower pole piece on the channel called for by the *A* code.

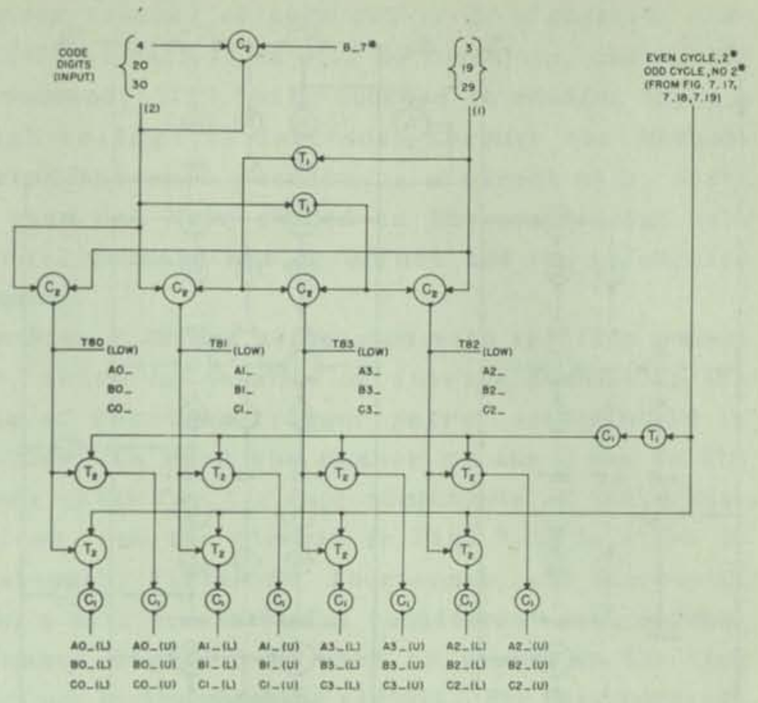


Fig. 7.14—*Ten's pyramid.*

The circuits controlling the *A* bus gates of the α, β, and manual constant registers having codes 03*, 04*, 05*, are shown in the upper portion of Fig. 7.17. Voltages from the *A* ten's-unit's combination circuit, Fig. 7.15, are combined with the drum phase voltage to select the upper or lower playback pole pieces on these channels. Channels α and β may also be selected under control of the *i* register, by means of code *A*08 in combination with a voltage supplied by the *i* register.

Circuits comparable to the one in Fig. 7.17 are provided for the *B* and *C* codes as shown in Figs. 7.18 and 7.19, respectively. The *B* circuit differs from the *A* in four respects. First, since the constants may be played back to the *B* transfer channel under control of the function sensing register, there is a pentode gate controlling the *B* number time output for the number time, code 3*, chosen by the function sensing register. Also, provisions are made for sensing the 2* component of the function number time when necessary, in determining the selection of upper or lower pole pieces. Second, the times for reading from channel I_b to the *B* bus are 2 number time on an odd cycle and 4 number time on an even cycle. Third, there are no provisions for calling a number from the α or β channel under control of the *i* register. Finally, the *B* number time is abrogated only for codes *B*00, *B*07*, and *B*17*. It is not abrogated in an external transfer, *B*27*, in order that the algebraic sign coded on the *B* sign digits may be associated with the digits being externally transferred.

In the *C* circuit, Fig. 7.19, the *i* register number time is delayed one number time by a trigger pair delay circuit. The *C* number time is abrogated only for code *C*00.

The special control circuits provided in the calculator will now be

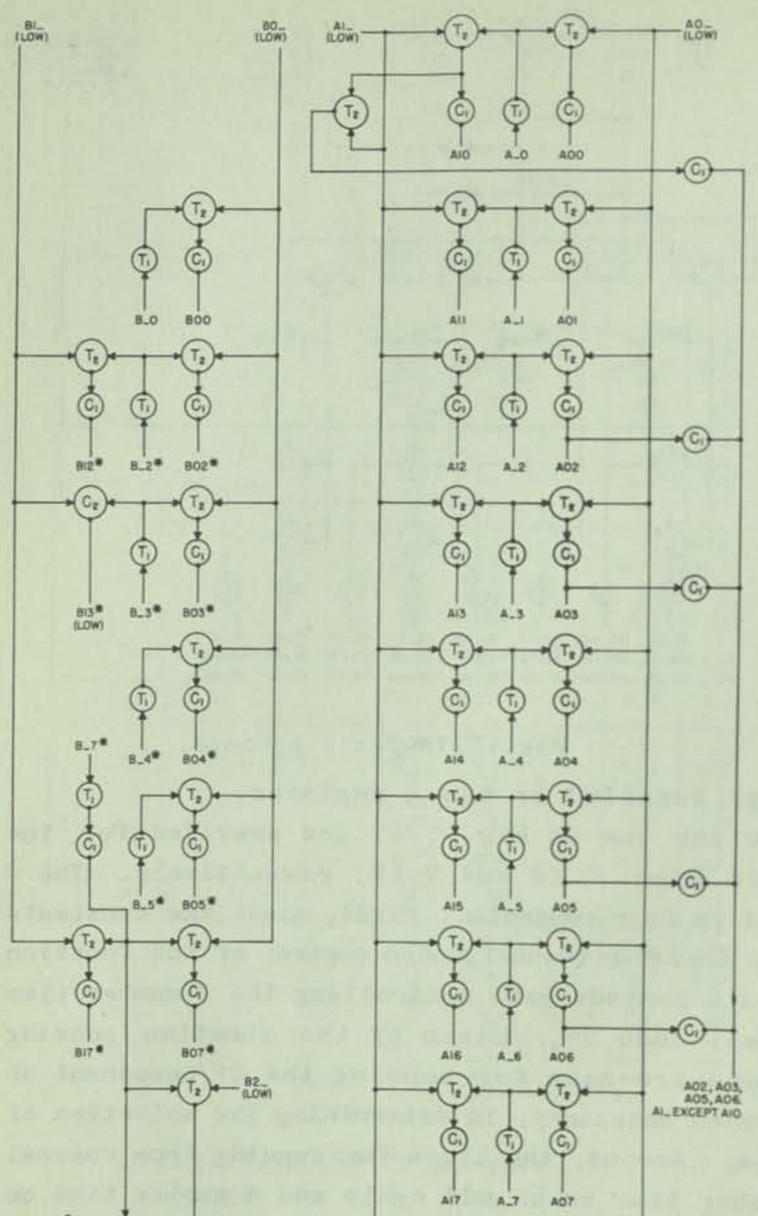


Fig. 7.15—*Ten's-unit's combination* (A,B).

described. The checkstop register, Fig. 7.20, is a trigger pair, which, under control of code *C*16*, stores the sign of the quantity on the *C* bus. When the operation code given is 8 or 9, for check or stop, respectively, and if a negative sign has been stored, this circuit stops the calculator, energizes the alarm relay, and turns on either the check or the stop light. If a positive sign has been stored, the circuit has no effect and the calculator continues in operation.

The identity-check circuit, Fig. 7.21, operates under control of operation code 6. During 4 number time, the sign and digits of the quantity selected under the *A* codes are compared with the sign and digits of the quantity selected under the *B* codes. A check failure sounds an alarm to stop the calculator and turns on the identity-check failure light.

The conditional call register, Fig. 7.22, stores the sign of the quantity on the *C* bus, under control of code *C*05*. If a positive sign has been stored, the conditional call light will be turned on, indicating that a conditional call command, *B*17*, will succeed in reading through the trigger pair. A high voltage is delivered through the cathode follower gate *S*132, causing the same results as a direct call, *B*07*. However, if a negative sign has been stored in the conditional call register, a conditional call command has no effect and the calculator proceeds to the next command.

The remaining gates in Fig. 7.22 are associated with the line number register. This register, which is capable of storing numbers in the range 0000-3999, consists of fourteen trigger pairs, as indicated in Fig. 7.23. The command *C*04*, to read the number on the *C* bus to the line number register, opens gates for the four components of the *C* bus. Gated A_2 pulses are supplied from the circuit in Fig. 7.22 to store in the trigger pairs the sixteenth, fifteenth, fourteenth, and thirteenth digits on the *C* bus. When a call or successful conditional call occurs, the number coded on sequence digits 3-16 will be stored in the line number register for comparison in the matching circuit. For this purpose, fourteen pentode gates, Fig. 7.23, are opened at 16, 15, 14, or 13 digit time. Corresponding A_2 pulses then store the digits in the trigger pairs. Both the digit-time voltages and the gated A_2 pulses are derived from the circuit in Fig. 7.22. They are cut off when the line called is 0000, to permit the number previously stored in the line number register to control the matching circuit.

For an external transfer command the circuit in Fig. 7.22 delivers gated digit-time voltages to the circuit in Fig. 7.23, permitting the number coded on digits 3-16 to be fed through the first set of gates used in calls and successful conditional calls. The number is then delivered to the *B* transfer channel, through gates indicated in Fig. 4.2.

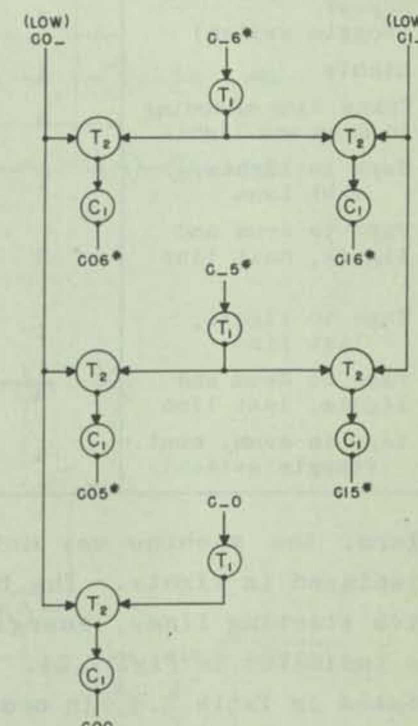


Fig. 7.16—*Ten's-unit's combination (C).*

The remaining sequence circuits are associated with the manual controls on the front panel of the calculator, Plate XII. The various operations are performed under control of cycling step switches and relays mounted behind the front panel.

With the exception of the stop and reset controls, each button may be operated only under certain conditions. For example, if there is an

TABLE 7.3

Buttons on main control panel

Button	Start Line No.	Relays Energized at Start of Cycle	Step Switch	Wipers
Start next line	3	*B*5,*C*1,*C*2,*C*3	1	4,5,6
Start spec. line	3	*B*2,*B*5,*C*1,*C*2,*C*3,*D*1	1	2,4,5,6
Run next line and stop	3	*B*1,*B*5,*C*3,*C*4,*D*1	1	3,4,5,6
Run spec. line and stop	3	*A*4,*A*5,*B*5,*C*3,*C*4,*D*1	1	2,3,4,5,6
Read last line to lights	2	*A*3,*C*1,*C*2,*D*2	1	1,5,6
Read spec. line to lights	2	*A*1,*A*2,*A*3,*B*5,*C*3,*C*4, *D*1,*D*2	1	1,2,3,4,5,6
Operate	3	*D*10,*E*12,*G*11	2	1,5,6
Repeat (toggle switch)	5	*C*10,*D*10,*E*12,*G*11	2	1,5,6
Lights	1	*A*10,*E*12,*G*12	2	1,4,6
Trans. line of coding to drum and lights	3	*G*9,*A*10,*E*12,*G*12,*B*12, record match relays	2	1,4,6
Tape to lights, next line	1	*E*9,*E*12,*E*13,*H*12	2	1,3,6
Tape to drum and lights, next line	3	*D*9,*E*9,*E*13,*E*12,*H*12, *A*12,*B*12, record match relays	2	1,3,6
Tape to lights, last line	1	*C*9,*C*13,*E*12,*H*13,*C*12	2	1,2,6
Tape to drum and lights, last line	3	*A*9,*C*9,*C*13,*E*12,*H*13, *A*12,*B*12,*C*12	2	1,2,6
Tape to drum, cont. (toggle switch)	4	*A*11,*B*11,*C*11,*E*11,*F*11, record match relays	—	- - -

alarm, the machine may not be started; however, information may be displayed in lights. The buttons listed in Table 7.3 are connected to five starting lines, energized through normally closed relay contacts as indicated in Fig. 7.24. The relays are energized under the conditions listed in Table 7.4, in order to prevent conflicting operations.

Step switch 1, Fig. 7.25, controlled by the buttons, Fig. 7.26, operates the machine from commands previously recorded on the drum. One cycle of the step switch is ten steps, the same connections being provided on steps 11-20 as on steps 1-10. Each of the associated buttons energizes relays $C1$ and $C2$, which are held by contact $C1a$ through the normally closed contact $B6a$. The coil of the step switch is energized on step 1 through wiper 6 and contact $C2a$. The step switch is advanced by a 4 lobe, 4 rps cam until it reaches step 8. Then relay $B6$ is picked up, and, provided the control button has been released, relays $C1$ and $C2$ will be dropped out, and the step switch moves on to step 11. If, however, the start button is still depressed when the step switch reaches step 8, relay $C2$ will

TABLE 7.4

Relays controlling starting lines

Relays	Energized for
$C10$	"repeat" operation
$B16$	machine running
$D16$	alarm condition
$A17, C17$	"tape to drum, continuous" operation

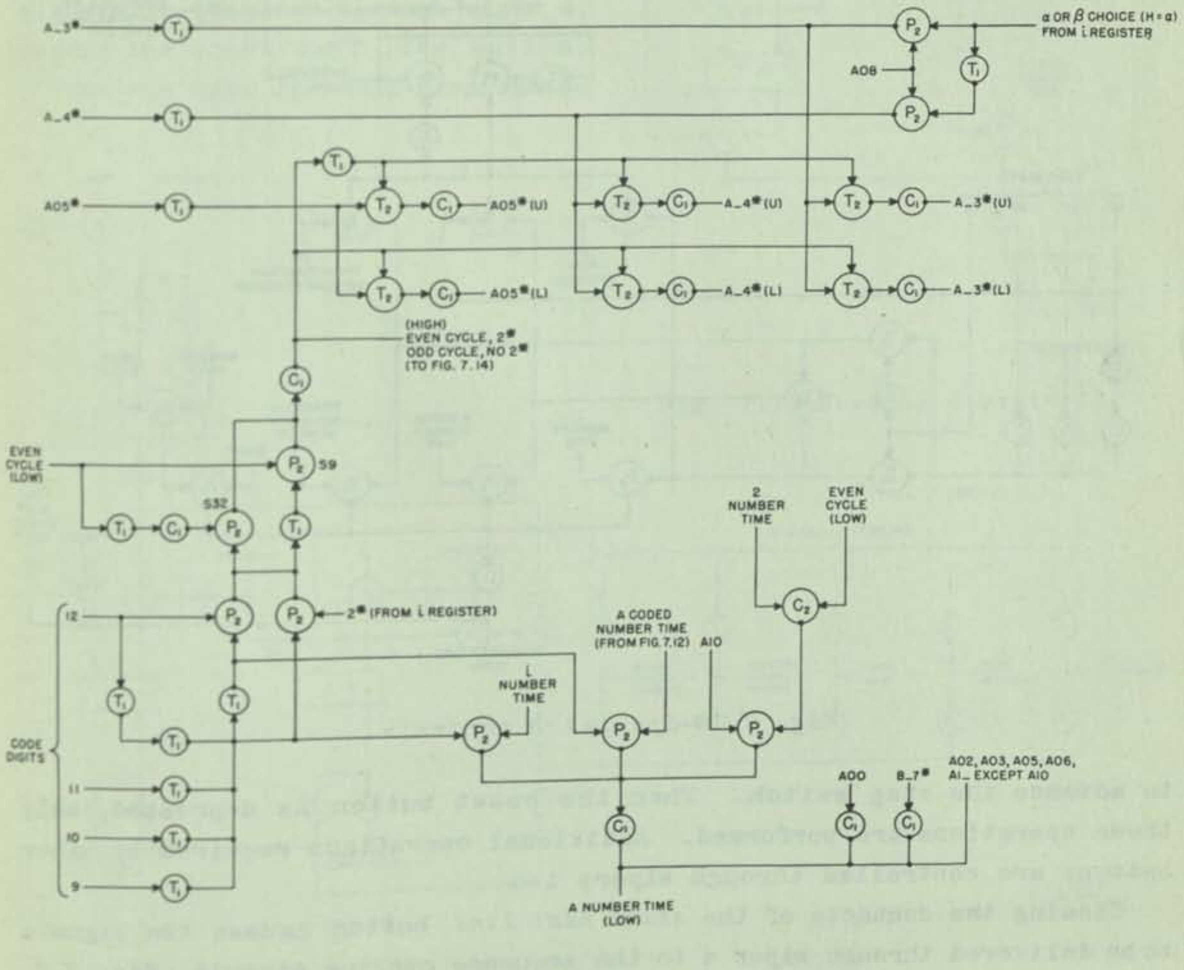


Fig. 7.17—*Special A controls.*

remain energized and the step switch will remain on step 8 until the button is released.

When a button associated with step switch 1 is depressed, the following operations occur: 1) The storage relays, keys, and light relays are reset; 2) the alarm bell is rung as a signal that the step switch is cycling; and 3) at the end of the cycle the relay hold line is interrupted. These operations are controlled through wiper 5, which is connected to a second cam, whose timing is the same as that of the cam used

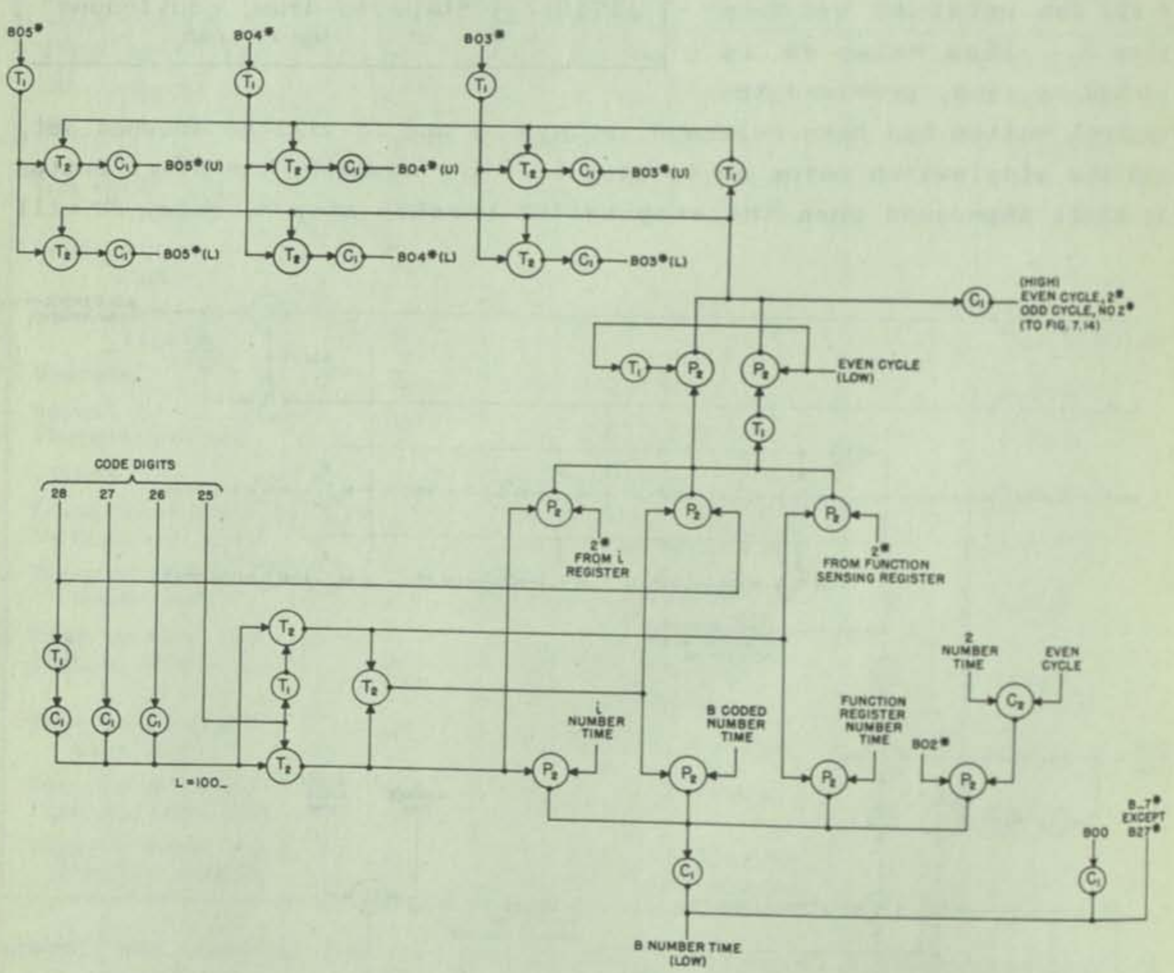


Fig. 7.18—*Special B controls.*

to advance the step switch. When the reset button is depressed, only these operations are performed. Additional operations required by other buttons are controlled through wipers 1-4.

Closing the contacts of the *start next line* button causes two signals to be delivered through wiper 4 to the sequence cycling circuit, Fig. 7.6, to start the calculator.

The *start specified line* button starts the calculator on the line indicated in the start line register. For this purpose the start-match relays must be energized to deliver the number in the start line register to the matching circuit. Relay *S*142 in the matching circuit is energized to delay the next command, giving the start-match relays time to drop out. These relays are energized through wiper 2. In addition, the start signals are supplied through wiper 4, as for the *start next line* button.

The *run next line and stop* button

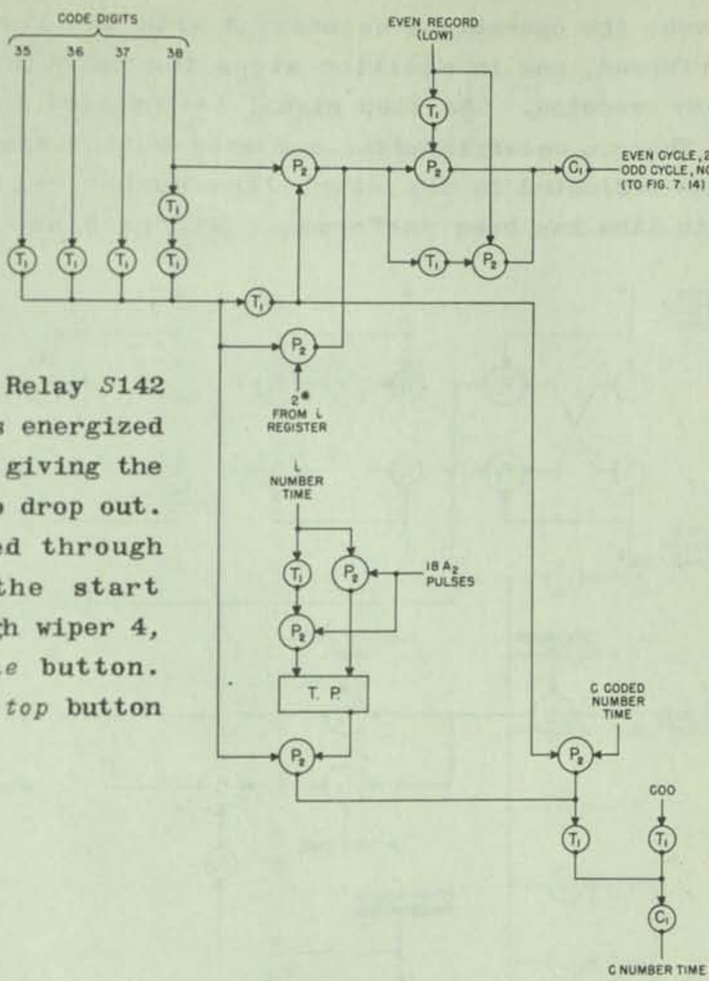


Fig. 7.19—*Special C controls.*

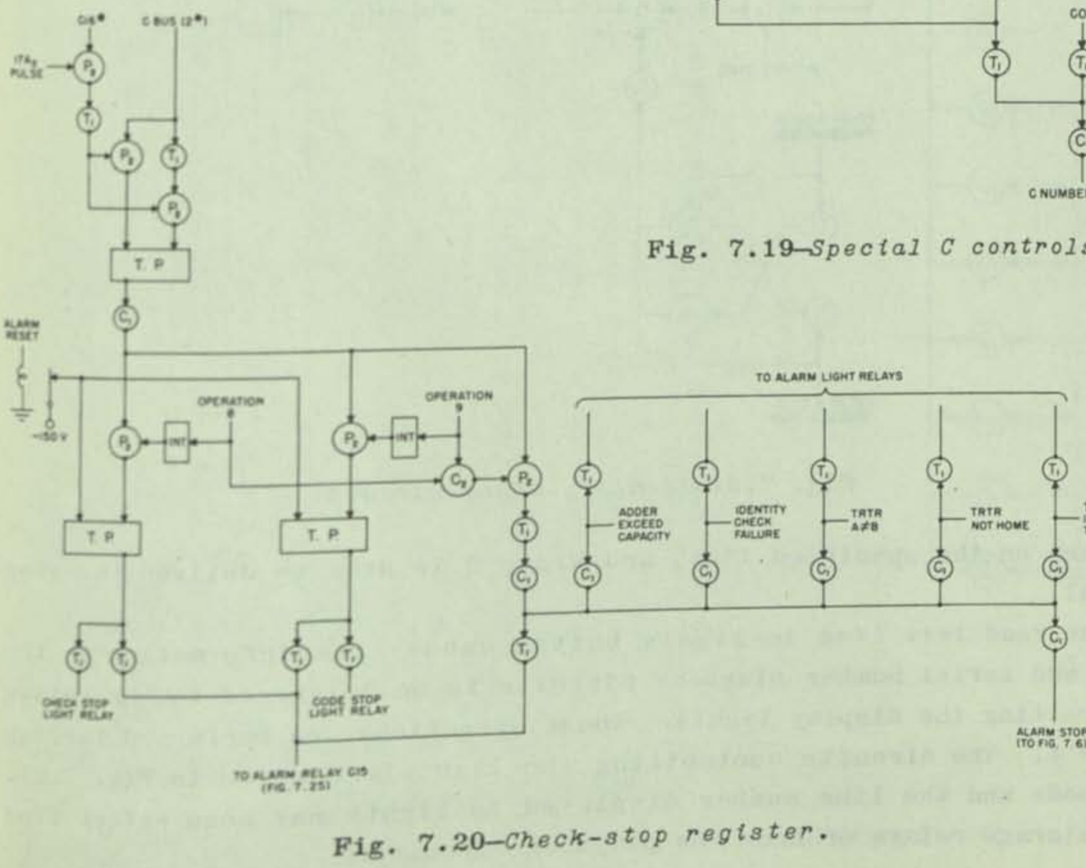


Fig. 7.20—*Check-stop register.*

causes the operations associated with the *start next line* button to be performed, and in addition stops the calculator after one command has been executed. The stop signal is supplied through wiper 3.

The *run specified line and stop* button starts the calculator on the line indicated in the start line number register, and stops it after this line has been performed. Wipers 2 and 4 are used to start the

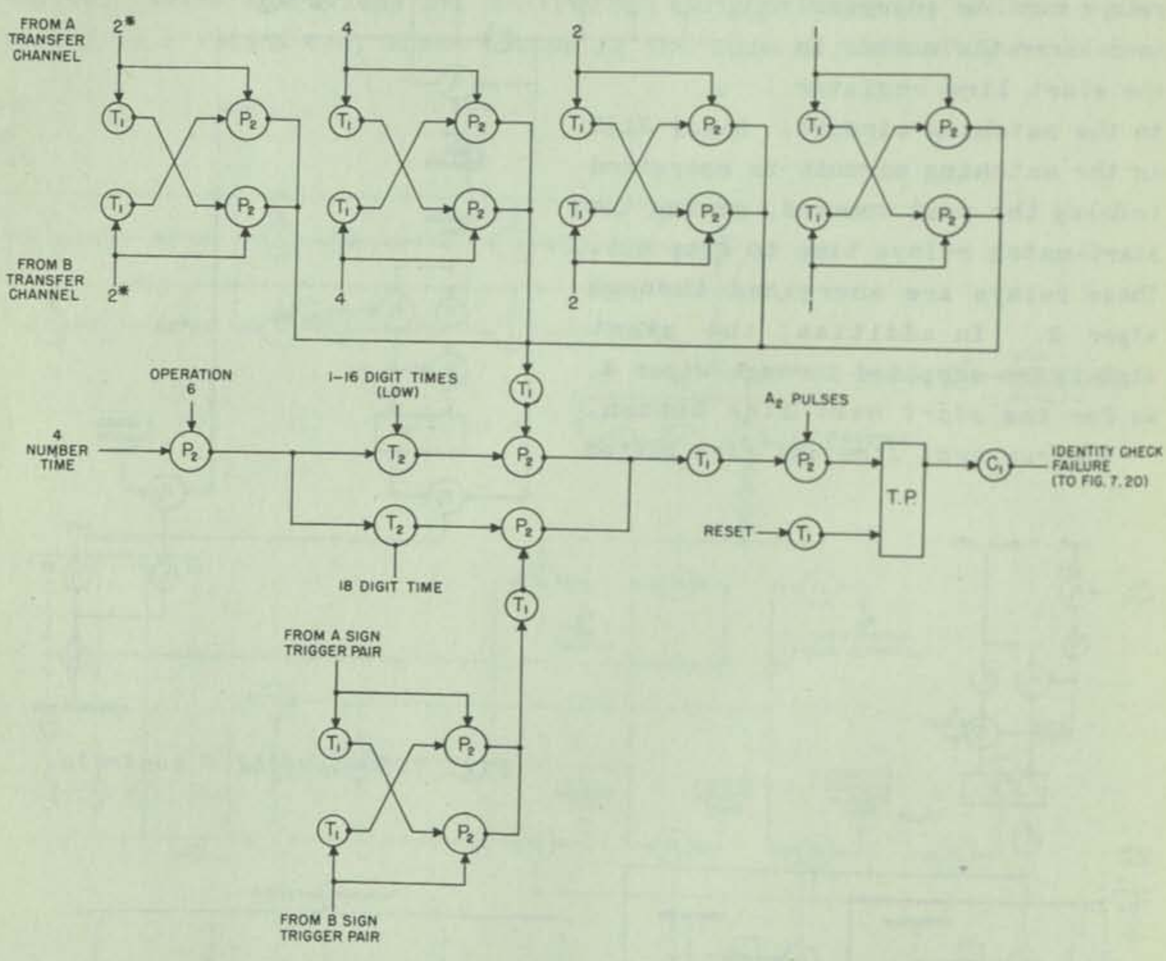


Fig. 7.21—*Identity-check circuit.*

machine on the specified line, and wiper 3 is used to deliver the stop signal.

The *read last line to lights* button causes the information in the code and serial number playback circuits to be delivered to the relays controlling the display lights. These operations are performed through wiper 1. The circuits controlling the lights are shown in Fig. 7.27. The code and the line number displayed in lights may come either from the storage relays or from the playback circuits.

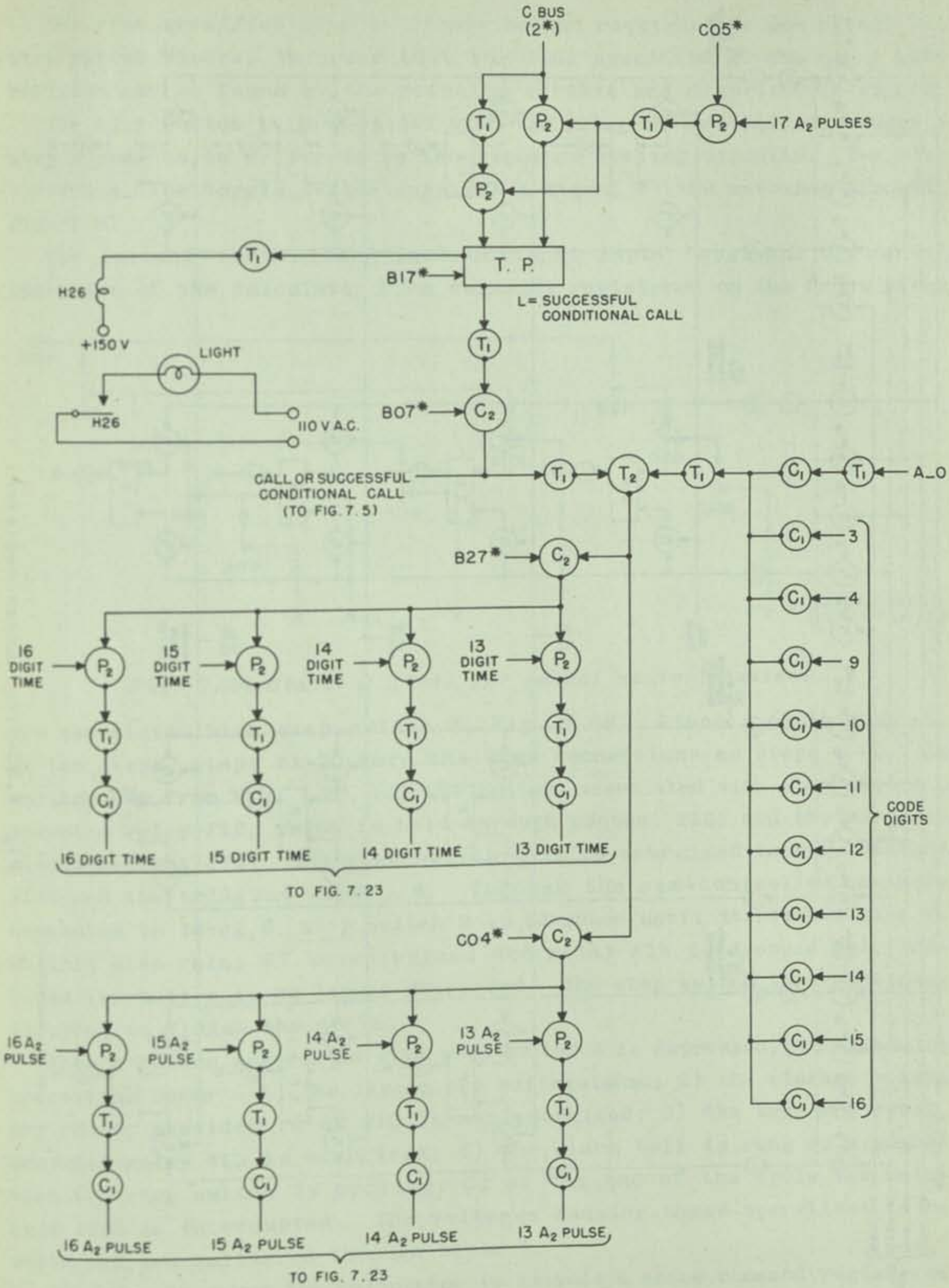


Fig. 7.22—*Control circuits for line number register.*

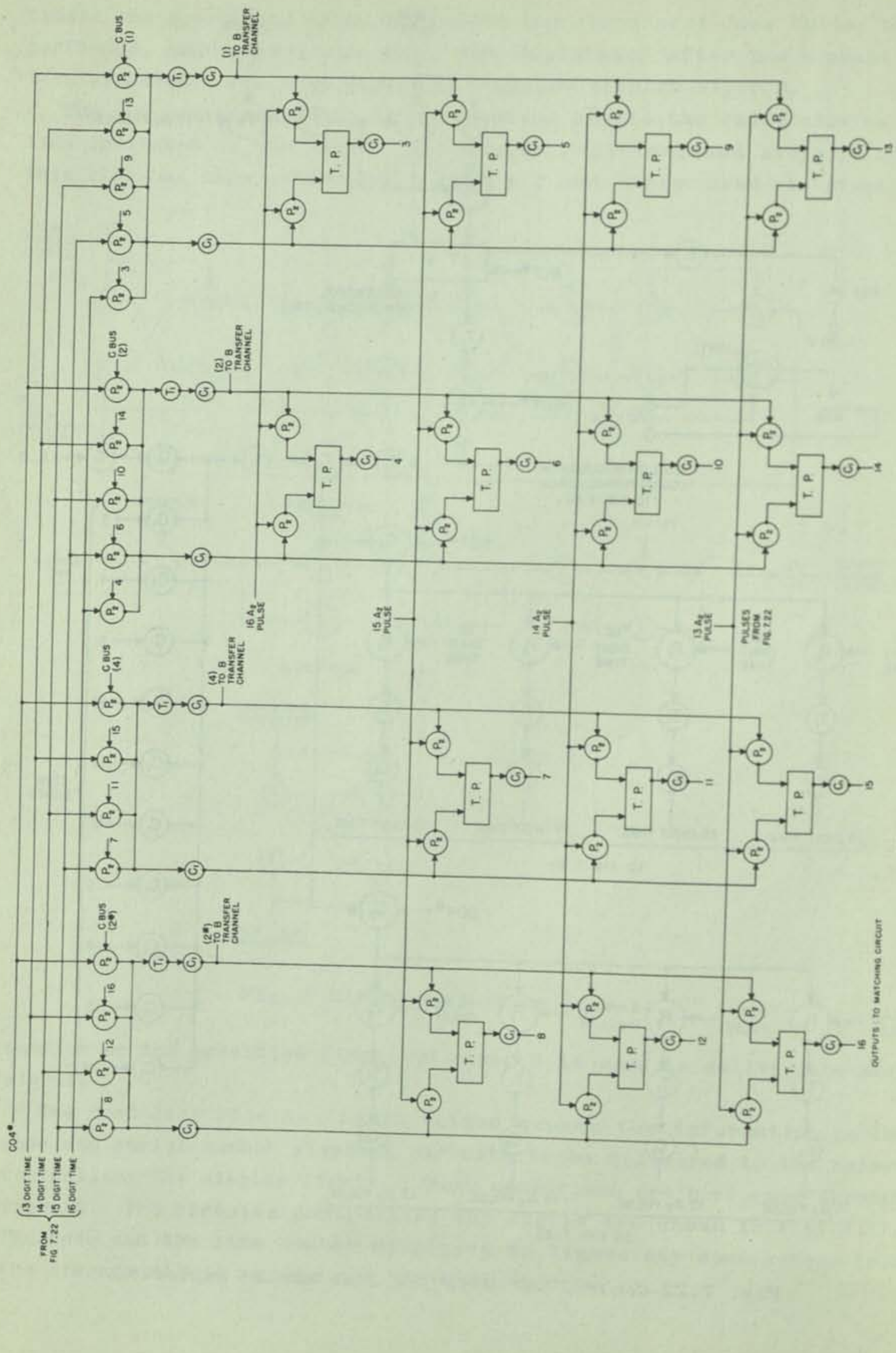


Fig. 7.23—Line number register.

The *read specified line to lights* button requires the use of all the step switch wipers, in order that the line specified in the start line register may be found by the matching circuit and displayed in lights.

The *stop* button is in parallel with the relay contact *D6b* and causes a stop signal to be delivered to the sequence cycling circuits. The *stop specified line* toggle switch supplies a signal to the matching circuit, Fig. 7.8.

The buttons controlling instructional input tapes and the manual operation of the calculator from commands registered on the front panel

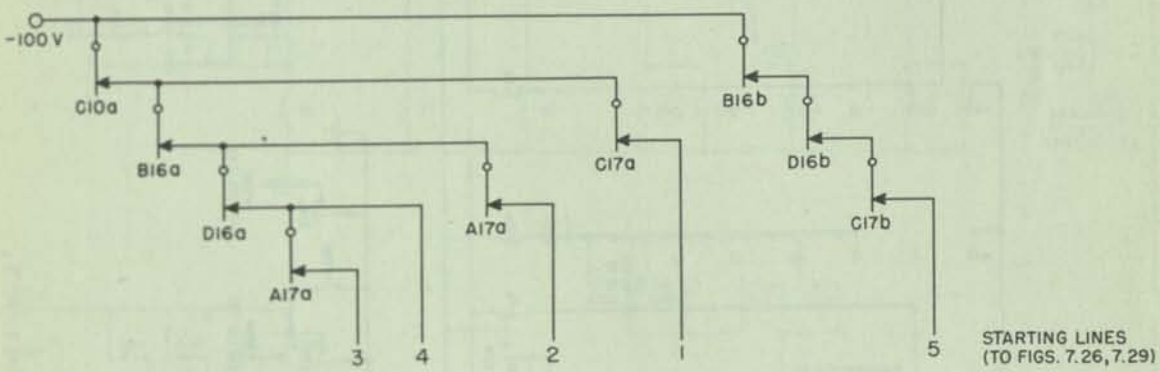


Fig. 7.24—*Starting lines for manual control buttons.*

are associated with step switch 2, Fig. 7.28. Since a cycle consists of ten steps, steps 11-20 have the same connections as steps 1-10. As may be seen from Fig. 7.29, all the buttons associated with step switch 2 energize relay *E12*, which is held through contact *E12a* and the normally closed contact *D7a*. The step switch coil is energized through contact *E12b* and the bridging wiper, 6. Through the cam-controlled contacts connected to level 6, step switch 2 is advanced until it reaches step 7. On this step relay *D7* is energized and relay *E12* is dropped out, provided the button is no longer depressed. The step switch then continues stepping to finish the cycle.

When a button associated with step switch 2 is depressed, the following operations occur: 1) The lights are extinguished; 2) the storage relays are reset, provided relay *C12* is not energized; 3) the keys are reset, provided relay *E13* is energized; 4) the alarm bell is rung as a signal that the step switch is cycling; 5) at the end of the cycle the relay hold line is interrupted. The voltages causing these operations to be performed are delivered through wiper 1.

In order to cause the calculator to execute a coded command registered in the keys, the *operate* button is used. Under control of wiper 5 on

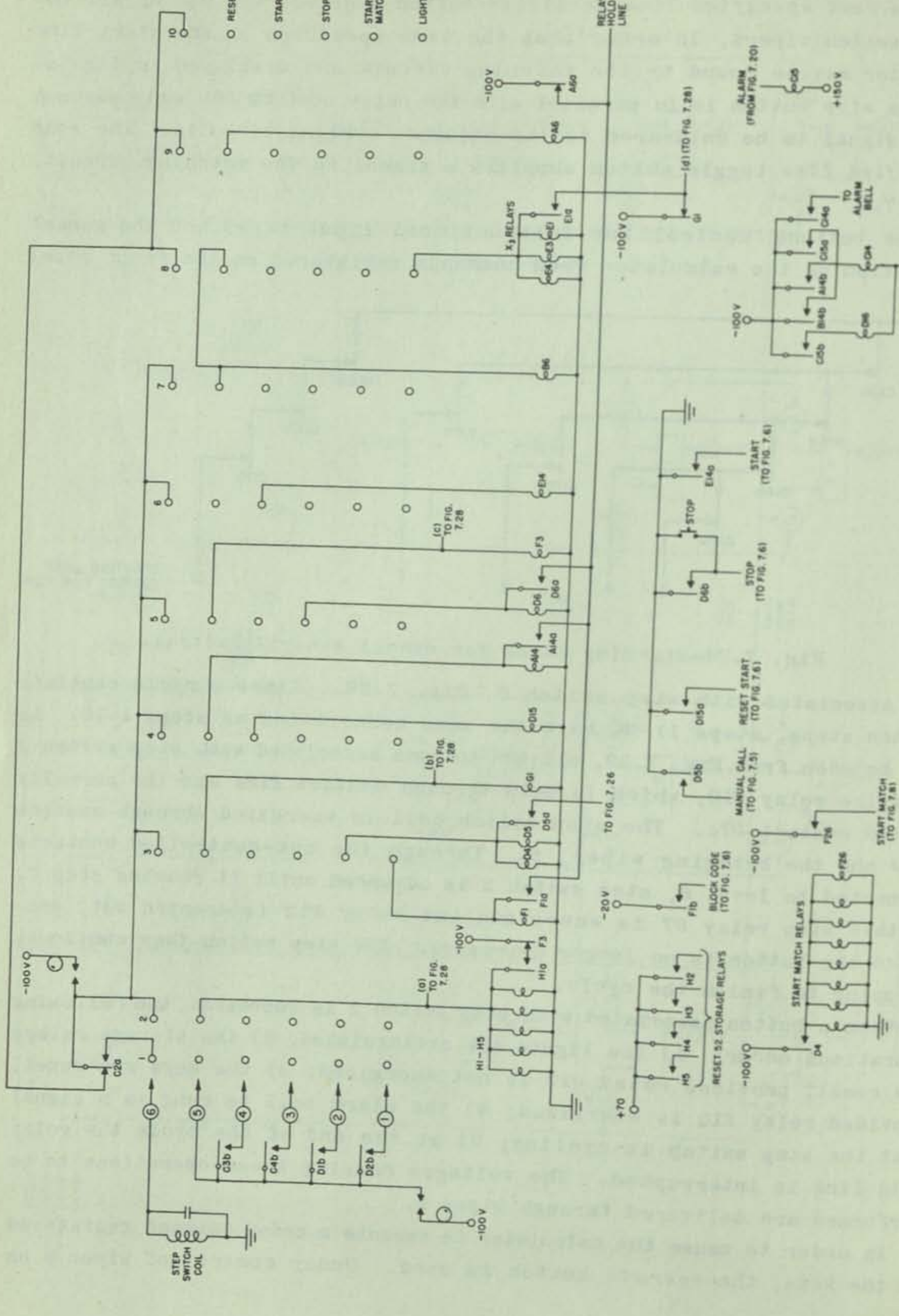


Fig. 7.25—*Step switch 1.*

step switch 2, the information in the keys is delivered to the storage relays, and the X_2 relays, Figs. 7.3 and 7.27, are energized. When step switch 2 is on step 7, relay *F*8 is picked up through wiper 5. Contact *F8b* energizes the operate relay shown in the upper left-hand corner of Fig. 7.30. Trigger pair 1 is tripped by an M_2 pulse. The

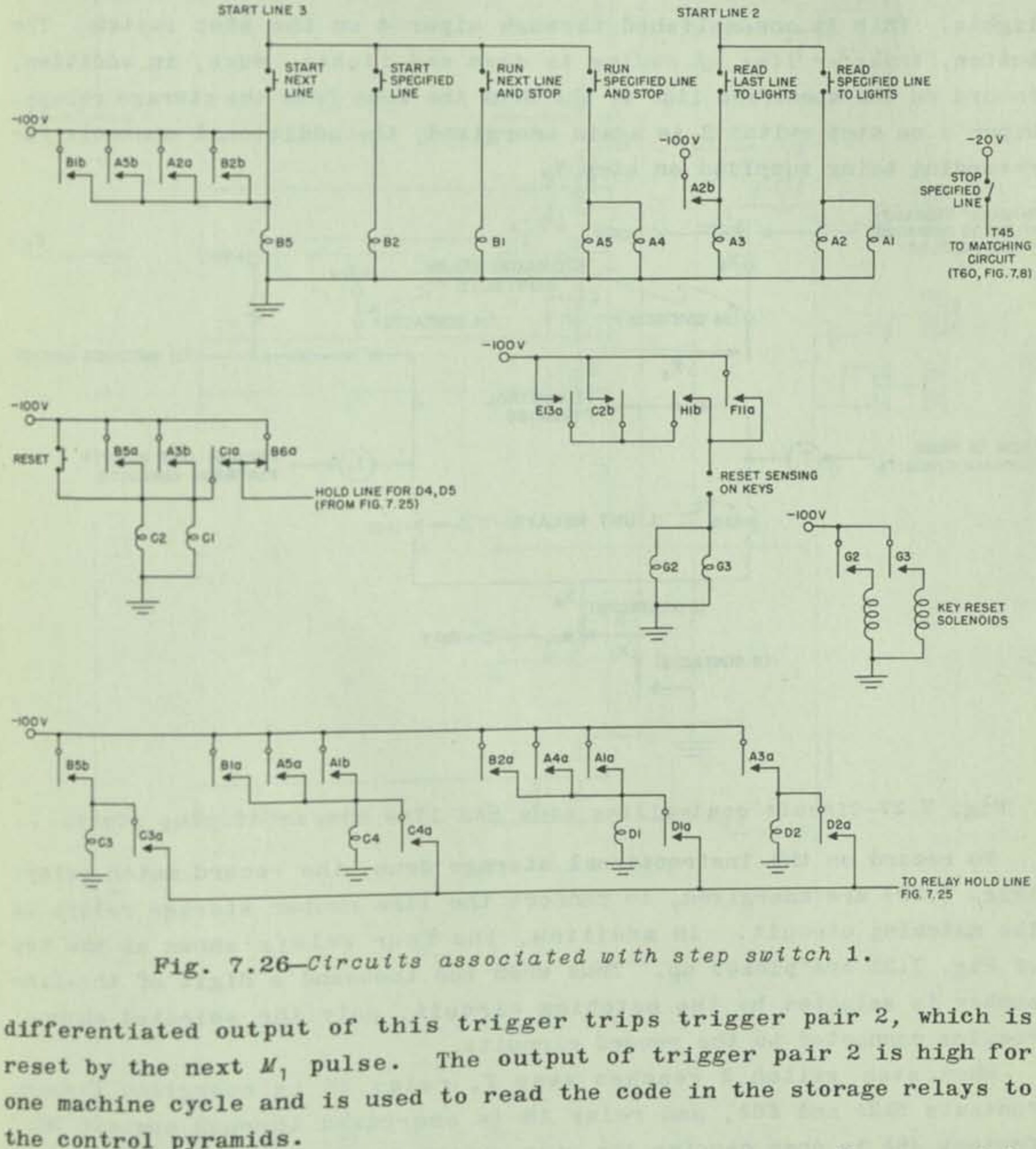


Fig. 7.26—*Circuits associated with step switch* 1.

differentiated output of this trigger trips trigger pair 2, which is reset by the next M_1 pulse. The output of trigger pair 2 is high for one machine cycle and is used to read the code in the storage relays to the control pyramids.

To perform one command repeatedly, for test purposes, the command is registered in the keys, and the toggle switch *repeat* is thrown. Up to step 7 the step switch cycling is the same as for the *operate* button. Then the step switch stops, and pulses read the command from the storage relays to the control pyramids, until the toggle switch is thrown. The step switch then completes the cycle.

The *lights* button causes the information in the keys to be delivered to the storage relays, and then to the relays controlling the display lights. This is accomplished through wiper 4 on the step switch. The button, *transfer line of coding to drum and lights*, must, in addition, record on the specified line of the drum the code from the storage relays. Wiper 4 on step switch 2 is again energized, the additional controls for recording being supplied on step 7.

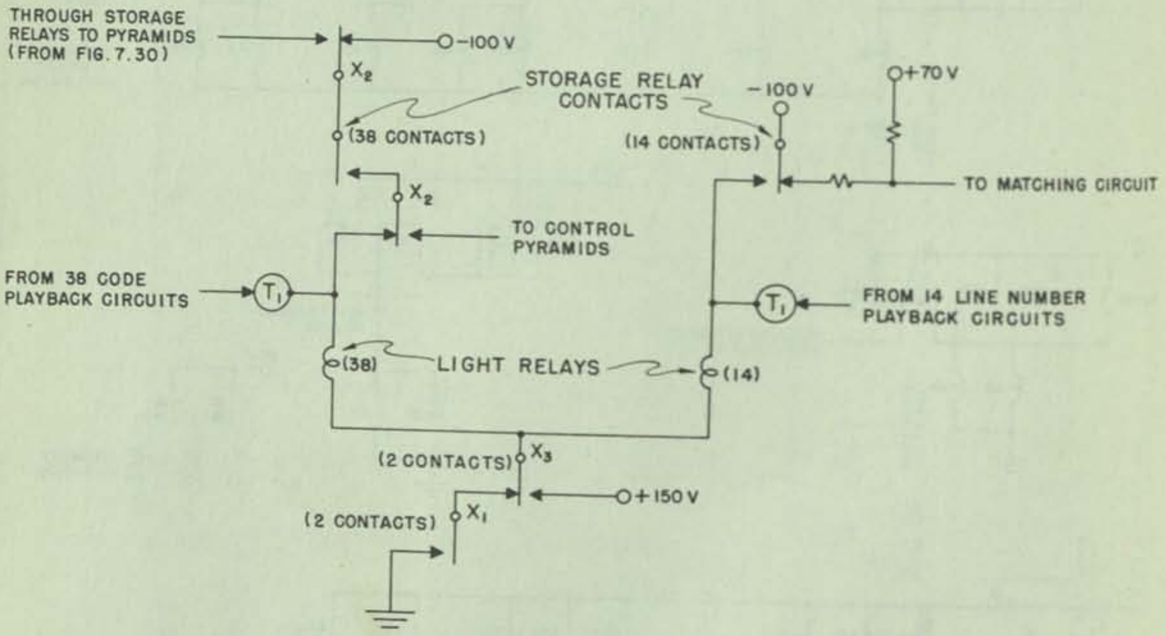


Fig. 7.27—*Circuit controlling code and line number display lights.*

To record on the instructional storage drum, the record match relays (Fig. 7.29) are energized, to connect the line number storage relays to the matching circuit. In addition, the four relays shown at the top of Fig. 7.31 are picked up. Thus when the thousand's digit of the line number is selected by the matching circuit, only the selected channel remains connected to the record circuits.

When step switch 2 reaches step 7, relay *B*8 is energized through contacts *B*12*b* and *E*8*b*, and relay *A*8 is energized through contact *B*8*b*. Contact *A*8*b* is open causing the step switch to stop on step 8 until the recording has been completed. Contact *B*8*a* supplies the signal to start the record cycling controls.

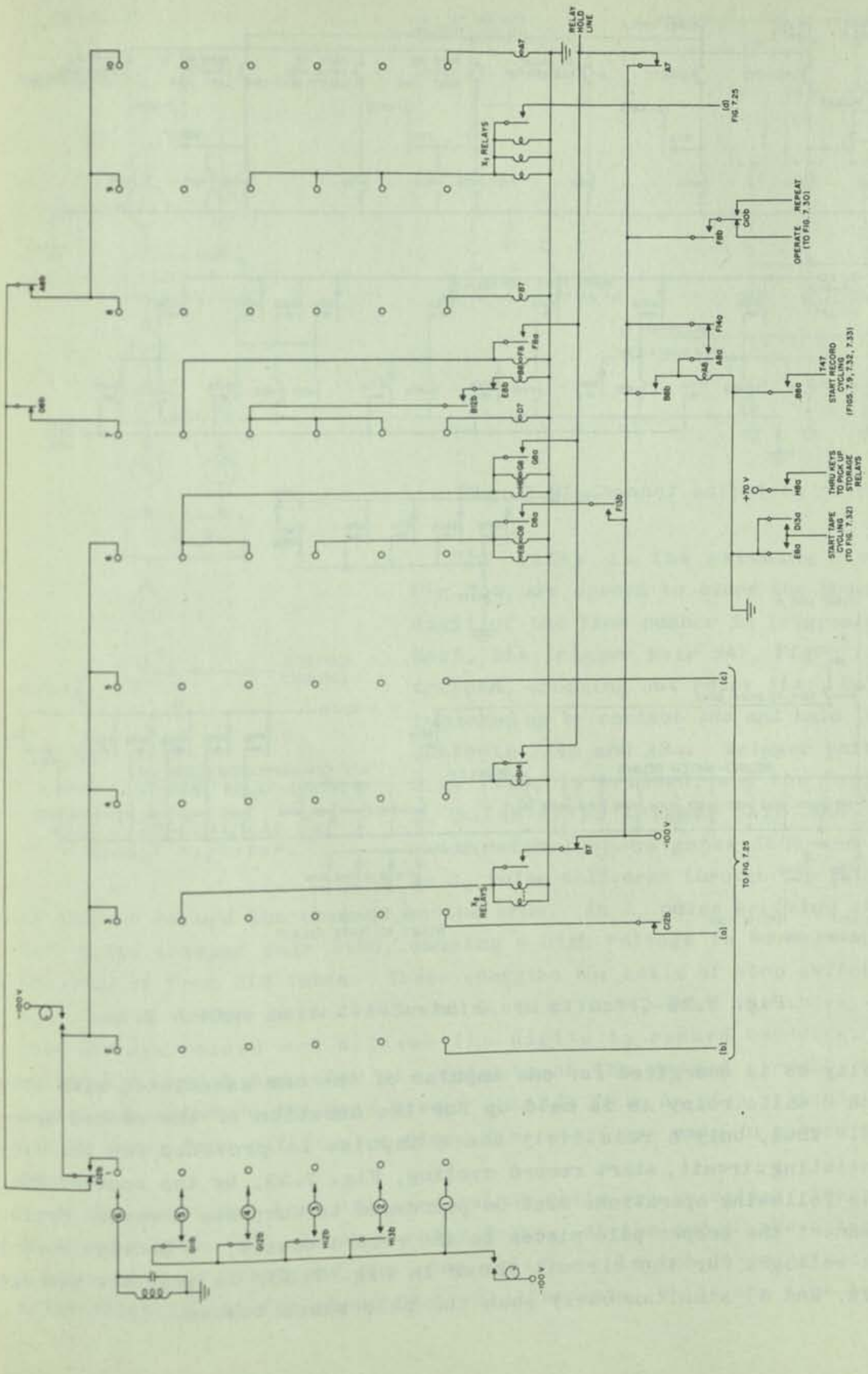


Fig. 7.28—*Step switch 2.*

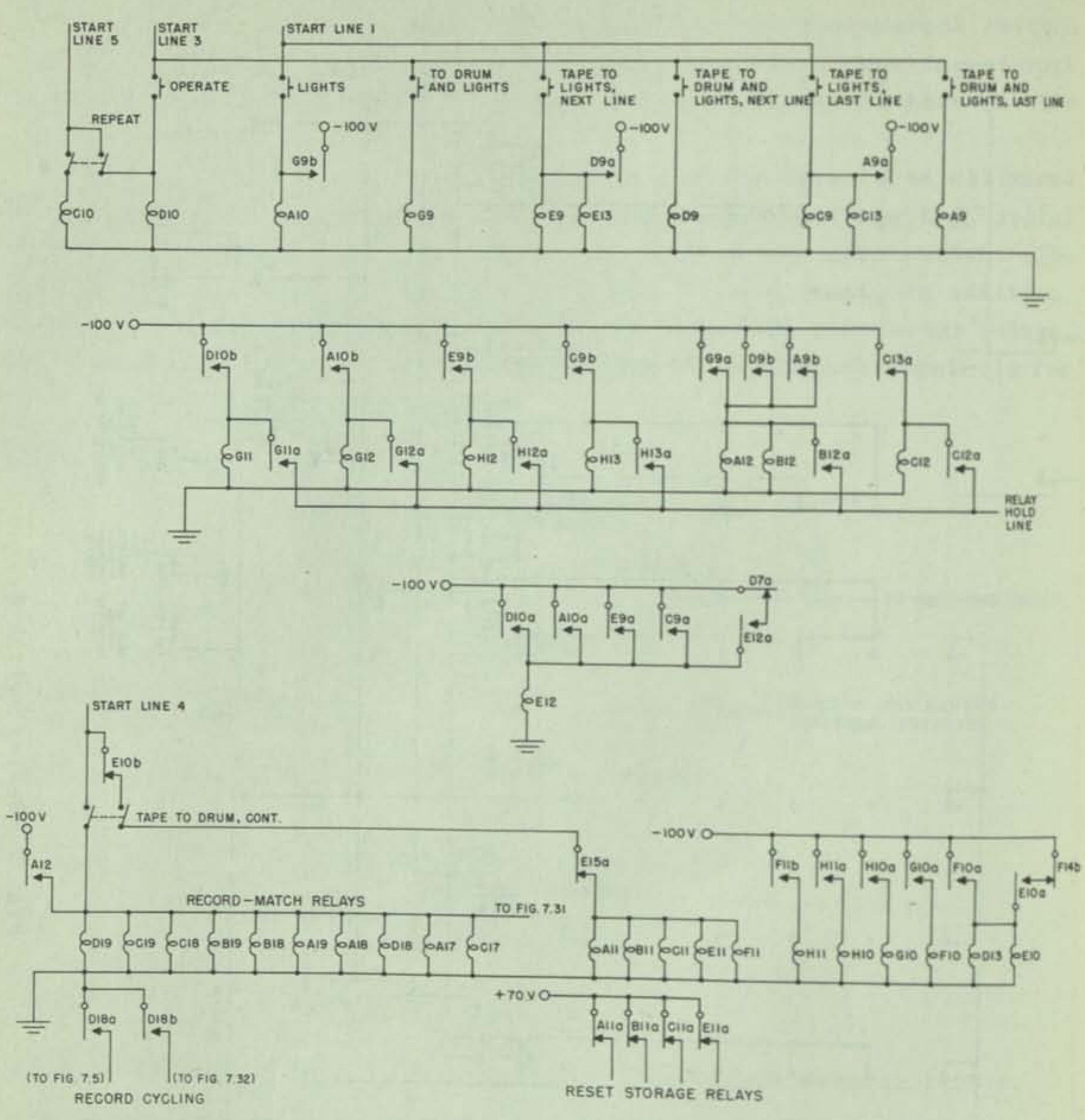


Fig. 7.29—*Circuits associated with step switch 2.*

Relay *B*8 is energized for one impulse of the cam associated with step switch 2 while relay *A*8 is held up for the duration of the record operation. Thus, only a relatively short impulse is provided for the differentiating circuit, start record cycling, Fig. 7.33, by the contact *B*8*a*.

The following operations must be performed to complete a record cycle: *a*) connect the proper pole pieces to the record busses; *b*) provide record match voltages for the circuit shown in Fig. 7.33; *c*) scan the storage relays; and *d*) simultaneously scan the pole piece busses.

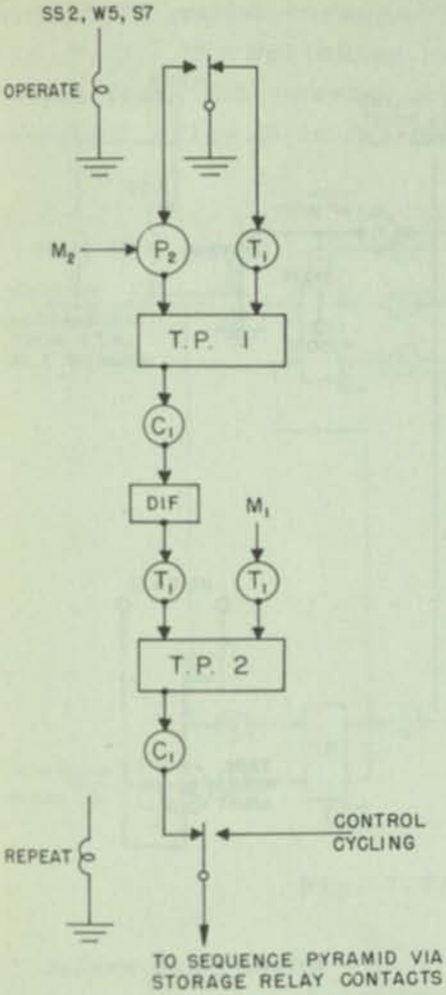


Fig. 7.30—*Controls for manual register.*

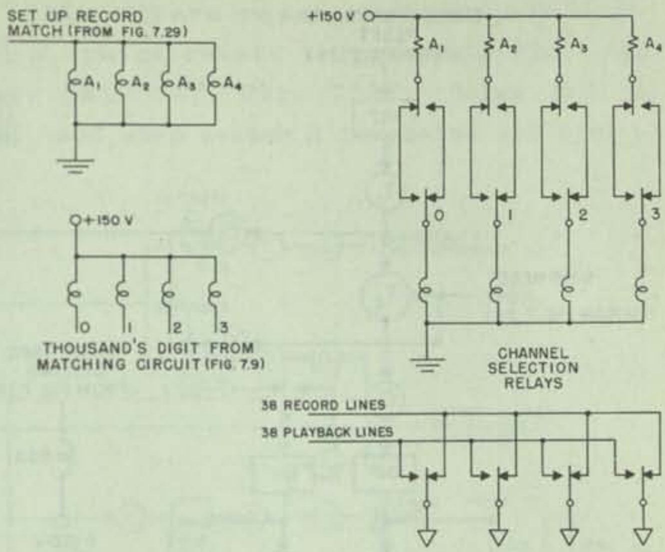


Fig. 7.31—*Channel selection relays.*

The gates in the matching circuit, Fig. 7.9, are opened to store the thousand's digit of the line number in trigger pairs. Next, the trigger pair *S*47, Fig. 7.32, is tripped, dropping out relay *F*14. Relay *A*8 is picked up by contact *B*8*b* and held through contacts *F*14*a* and *A*8*a*. Trigger pair *S*22, Fig. 7.33, is tripped, and the following S_2 pulse trips trigger pair *S*86. The match signal opens gates *S*108 and *S*119. An S_5 pulse delivered through the gate *S*108 is used to record the command on the drum. An S_1 pulse arriving at gate *S*119 trips trigger pair *S*106, causing a high voltage to be delivered to the grids of four 6L6 tubes. These energize the coils of step switches 5, 6, 7, and 8. Step switches 5 and 6, Fig. 7.34, scan the thirty-eight code storage relays and deliver the digits to record circuits. Step switches 7 and 8 transfer the outputs of the record circuits to the busses connected to the pole pieces (Fig. 7.35). As step switches 5 and 6 both sense four relays at a time, five steps are made in recording one command.

At the same time that the step switch coils are energized, the output of trigger pair *S*106 starts a relay delay. At the end of the delay, the trigger pair is reset and the step switch coils are de-energized. The relay delay is set for about forty milliseconds, while the time between

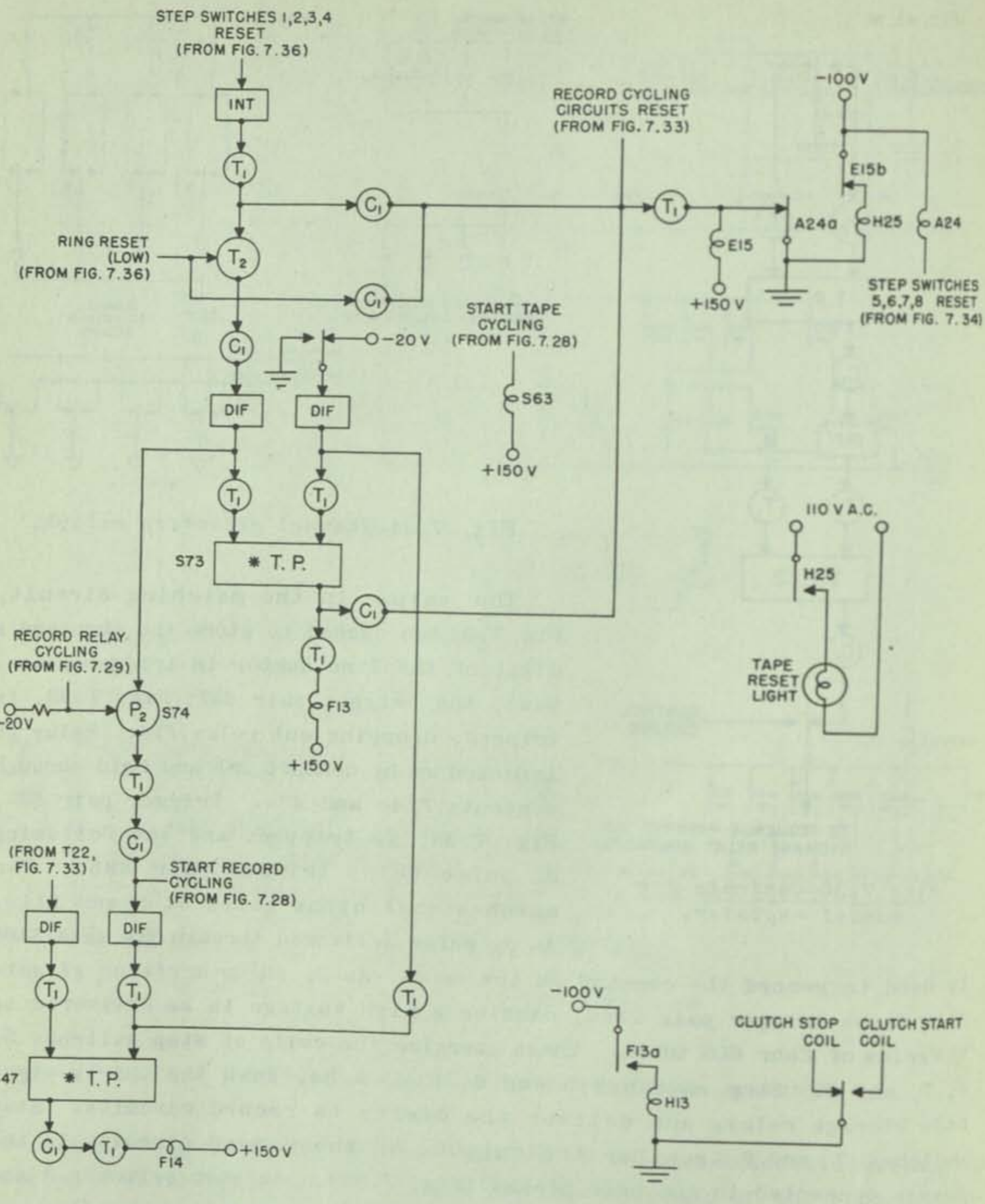


Fig. 7.32—*Record and playback cycling controls.*

match signals is approximately thirty-three milliseconds. Thus eight digits are recorded on the drum at every other match signal. On step 5, 10, 15, or 20 of the four associated step switches (5, 6, 7, and 8), a circuit to ground through the sixth wipers resets trigger pair *S*132, Fig. 7.33. The following gated S_1 pulse resets trigger pair *S*86. The output from *T*22 resets trigger pair *S*47, Fig. 7.32. Relay *F*14 is energized, relay *A*8 is dropped out, and step switch 2 completes its cycle.

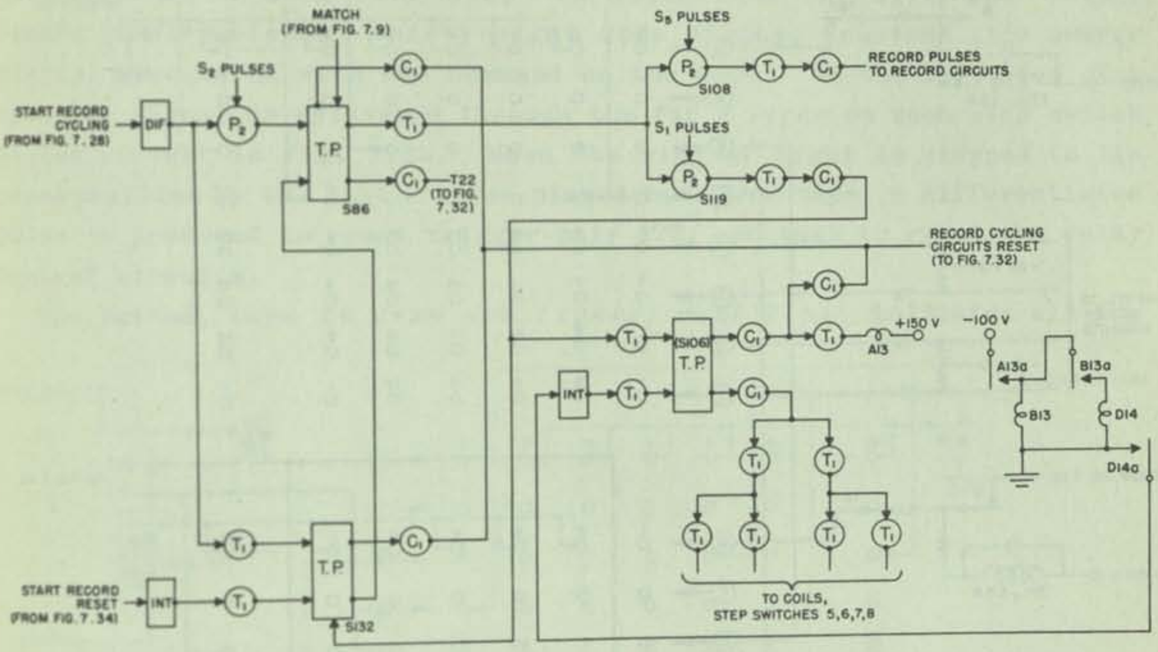


Fig. 7.33—*Record cycling controls.*

Before depressing the button, *transfer line of coding to drum and lights*, or any of the buttons associated with the instructional input tape, the light *tape circuits reset* must be on. This light indicates that the electronic circuits used in tape reading have been reset and that the step switches (1,2,3,4) used in reading from tape to the storage relays and the step switches (5,6,7,8) used in transferring coded commands from the storage relays to the drum are all on the first step of a cycle. The circuit controlling the light is shown in Fig. 7.32. These circuits may be reset manually by depressing the button *reset sequence tape input circuits*.

The button, *tape to lights, next line*, causes one command and its line number to be played back from tape, delivered to the storage relays and then to the relays controlling the display lights. The operations are performed through wiper 3, which, on step 6, energizes relays *D*8 and *E*8. The contact *D8b* is opened to stop the step switch on step 7 until the tape reading has been completed. Through contact *E8a*, relay *S*63, Fig. 7.32, is energized, tripping trigger pair *S*73. Trigger pair *S*47 is also tripped, dropping out relay *F*14. The output from trigger pair *S*73 causes relay *F*13 to be picked up. Contact *F*13*b* holds relays *D*8 and

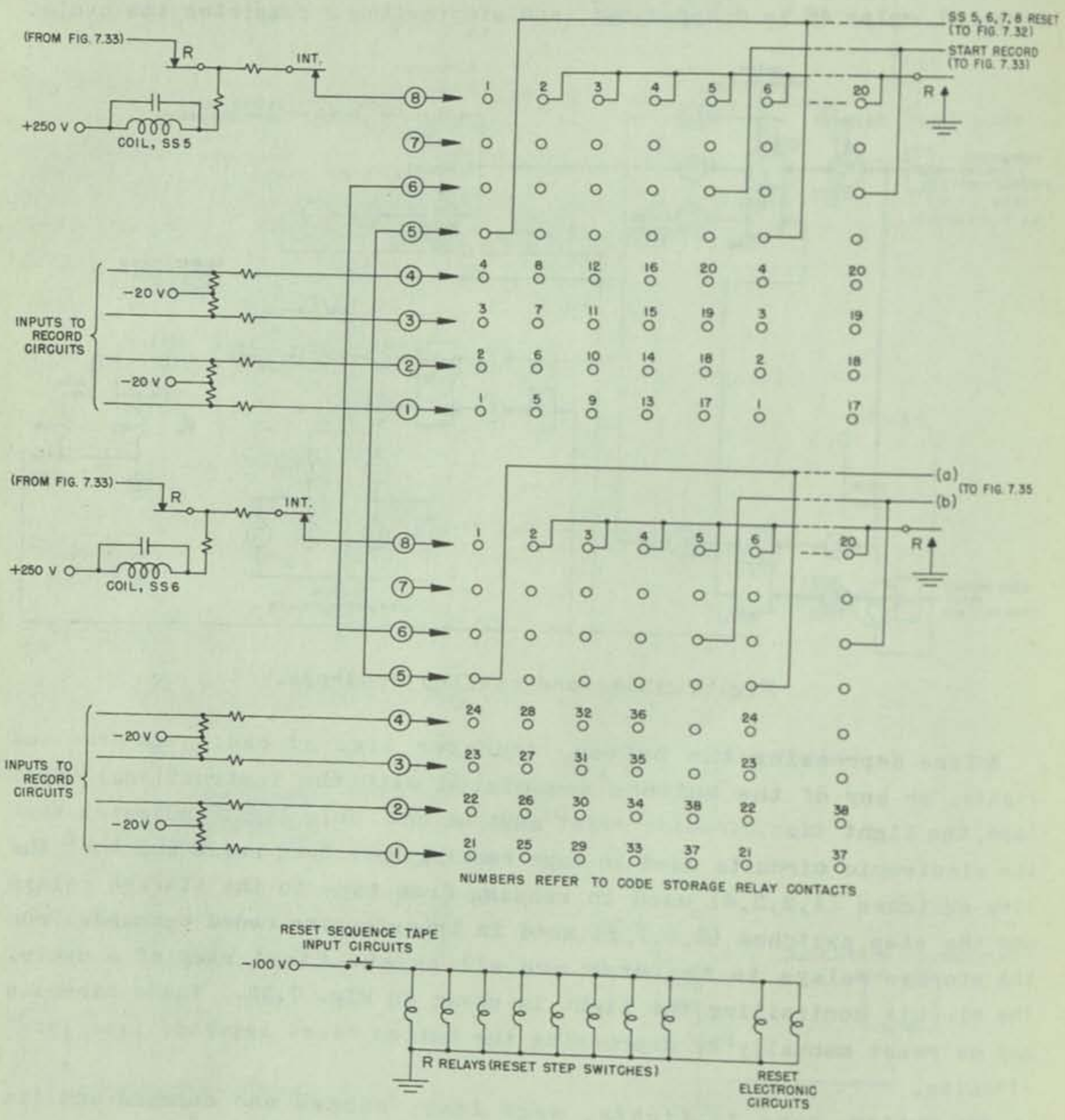


Fig. 7.34—*Step switches* 5, 6.

*E*8, while through contact *F*13*a*, the clutch relay, *H*14, is energized, causing the tape to start moving.

The circuits used to play back digits from tape and deliver them to step switches 1, 2, 3, and 4 are indicated in Fig. 7.36. *C* and *D* pulses from the tape step a ring of eight, while voltages from the ring control the step switches, and gate the *A* and *B* digits to the step switch wipers. The storage relays are connected to corresponding levels on the step switches, as indicated in Fig. 7.37. In playing back one command, the step switches advance four steps to distribute the sixty-four binary digits (twelve blanks, thirty-eight code digits, fourteen line number digits) associated with one command on the tape. On the last step of a cycle, a signal is delivered through the fifth wiper on each step switch to the circuit in Fig. 7.32. When the ring of eight is stepped to its reset position by the last *D* pulse played back from tape, a differentiated pulse is produced to reset trigger pair *S*73, and thus to reset the relay control circuits.

The button, *tape to drum and lights, next line*, initiates all the

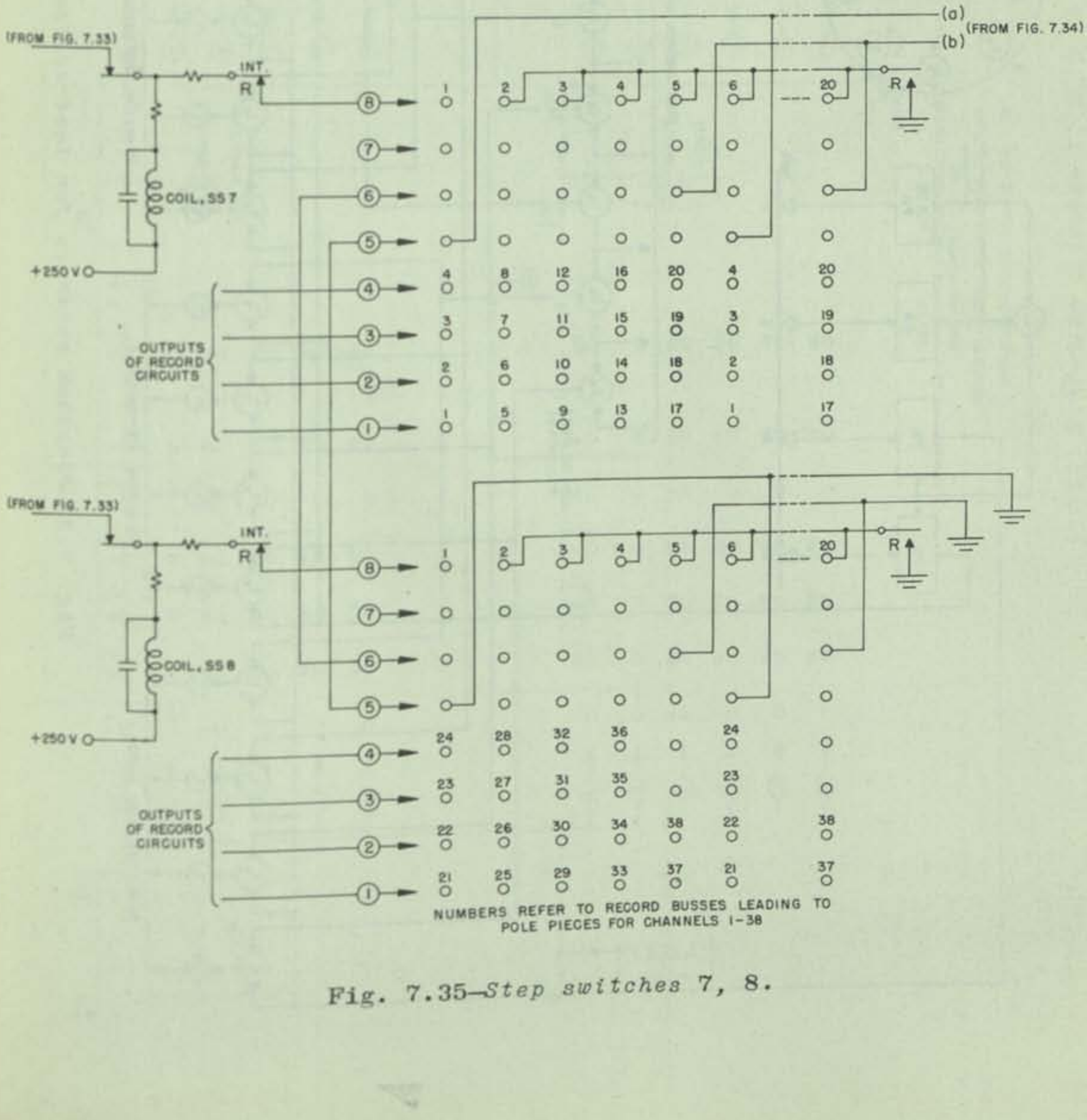


Fig. 7.35—*Step switches* 7, 8.

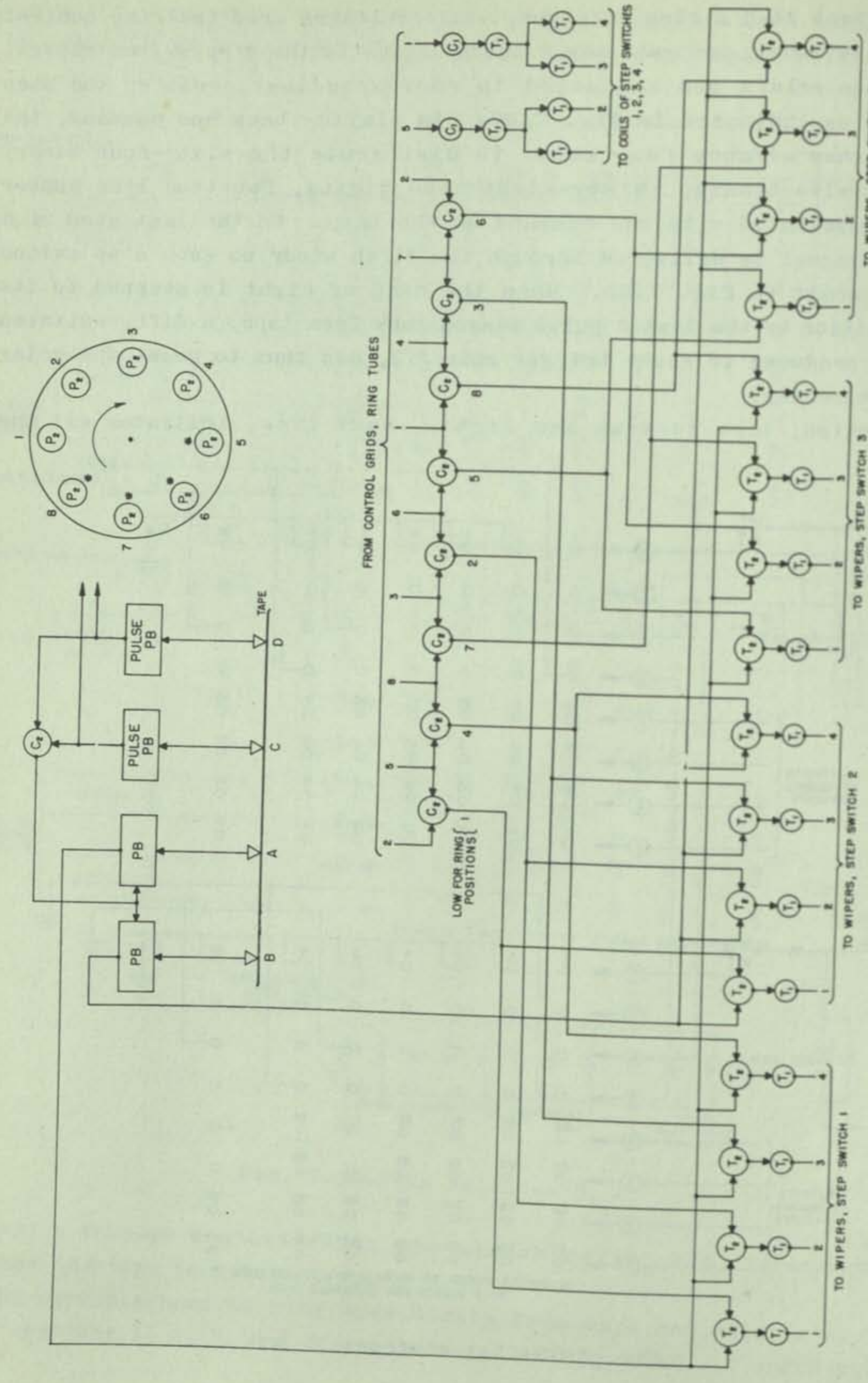


Fig. 7.36—*Playback controls for instructional tape.*

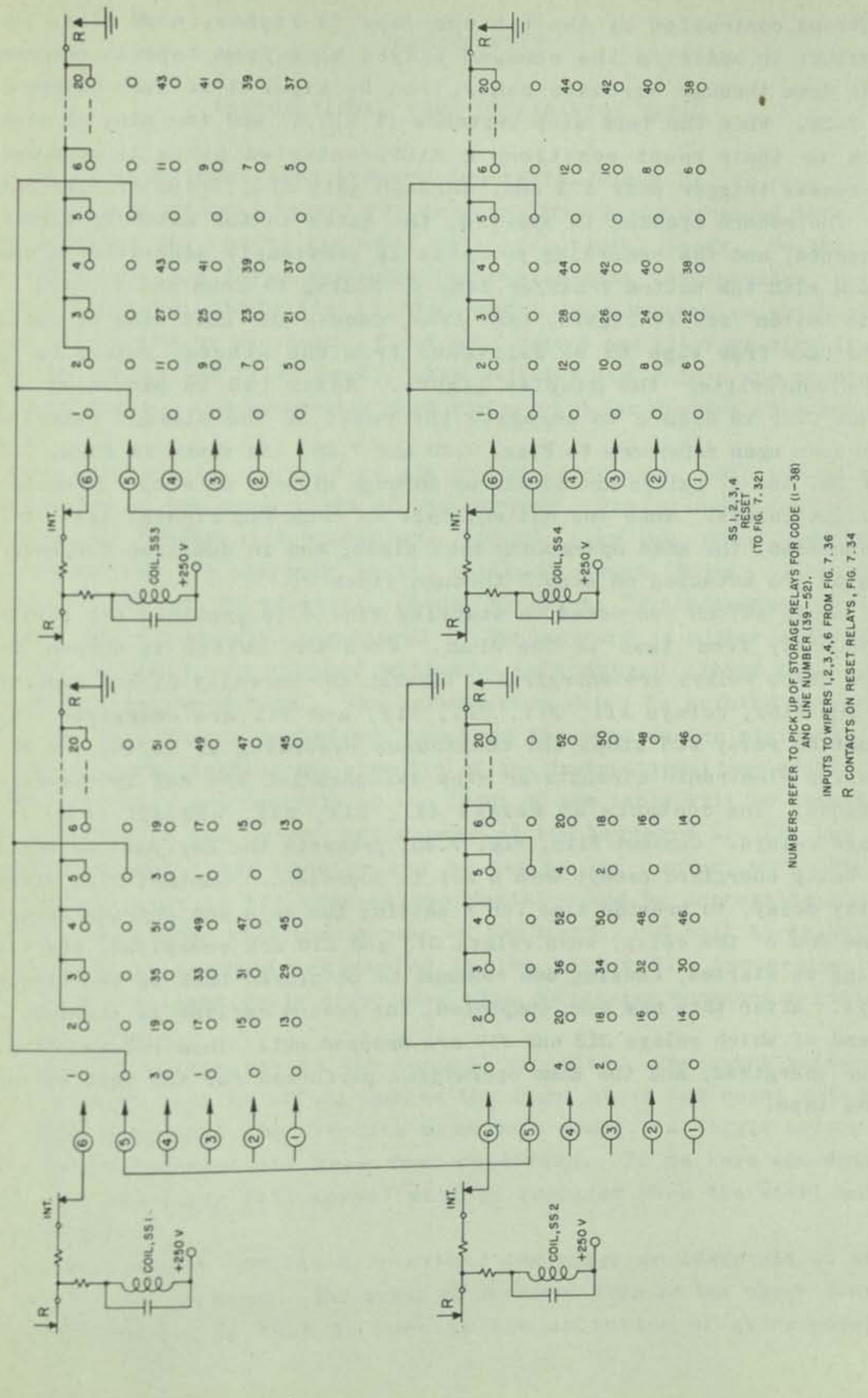


Fig. 7.37—*Step switches 1, 2, 3, 4.*

operations controlled by the buttons *tape to lights, next line,* just described; in addition the command played back from tape is recorded on the drum through circuits controlled by wiper 3 of step switch 2, Fig. 7.28. When the tape step switches (1,2,3,4) and the ring of eight return to their reset positions a differentiated pulse is produced. This resets trigger pair *S*73 and, through gate *S*74, trips trigger pair *S*47. The record cycling is started, the gates in the matching circuit are opened, and the recording proceeds as previously described in connection with the button *transfer line of coding to drum and lights.*

The button *tape to lights, last line,* causes the last line of coding played back from tape to be delivered from the storage relays to the relays controlling the display lights. Relay *C*12 is picked up and contact *C*12*b* is opened to abrogate the reset of the storage relays as may be seen upon reference to Figs. 7.25 and 7.28. As shown in Figs. 7.27 and 7.28, the X_1 relays are picked up through wiper 2 on step 9 to control the light relays. When the button, *tape to drum and lights, last line,* is depressed, the same operations take place, and in addition the record circuits are actuated on step 7 through wiper 2.

A toggle switch connected to starting line 4 is provided for reading continuously from tape to the drum. When the switch is thrown, the record match relays are energized. Through the normally closed contacts *E*10*b* and *E*15*a*, relays *A*11, *B*11, *C*11, *E*11, and *F*11 are energized. The contact on relay *E*15 stops the continuous recording if at the end of a cycle the electronic circuits or step switches 1-8 are not in the reset positions. The contacts on relays *A*11, *B*11, *C*11, and *E*11 reset the storage relays. Contact *F*11*a*, Fig. 7.25, prevents the key reset solenoids from being energized except when a key is depressed. Contact *F*11*b* starts a relay delay, to provide time for resetting the keys and storage relays. At the end of the delay, when relays *D*13 and *E*10 are energized, the tape cycling is started, causing one command to be played back to the storage relays. After this has been completed, the record cycling is started, at the end of which relays *D*13 and *E*10 are dropped out. Then relays *A*11-*F*11 may be energized, and the same operations performed for the next command on the tape.

CHAPTER VIII

INSTRUCTIONAL TAPE PREPARATION TABLE

The instructional tape preparation table, Plate XI, is used for recording instructional tapes, for checking previously prepared tapes, and for printing the coding recorded on instructional tapes. On the front of the table is a large keyboard for registering instructions, various control buttons, indicator lights, and a tape mechanism which can be used for record or playback. Electronic record and playback circuits are mounted behind the front panel, step switches and relays are mounted on the rear panel. The typewriter used for printing is mounted on a separate table.

The unit may be operated in any one of seven ways, selected by means of switches at the upper right of the keyboard. The following combinations of the four fundamental operations, record, playback, print, and check, are permitted: 1) playback, print, check; 2) record, print; 3) playback, print; 4) record; 5) playback, check; 6) print; 7) continuous playback, print. The information registered in the keyboard is either recorded or printed or both, or checked with the information played back from a previously prepared tape. The information which is printed is derived from the keys during recording, and from the tape during playback.

Before investigating the circuits of the instructional tape preparation table, the keys and controls on the front of the table will be described. The controls in the upper left corner of the keyboard are for the line number counter. This counter may be set to any number, 0000-3999, by selecting digits in the four-column register and by depressing a push-button. The number will then appear in the lights, and will be associated with the first command registered in the keyboard. Successive line numbers are automatically associated with successive commands registered in the keyboard.

The reset button resets the control circuits. The start button, at the far right, has no effect unless the light above the reset button is on, indicating that the circuits have been reset. A toggle switch may be thrown to prevent the keys from resetting. If no keys are down, a null line of coding (all zeros) will be supplied when the start button is depressed.

Commands may be indicated in either the upper or lower set of strip switches on the keyboard. The array of numbered keys in the upper section is more flexible in that it permits the selection of every possible

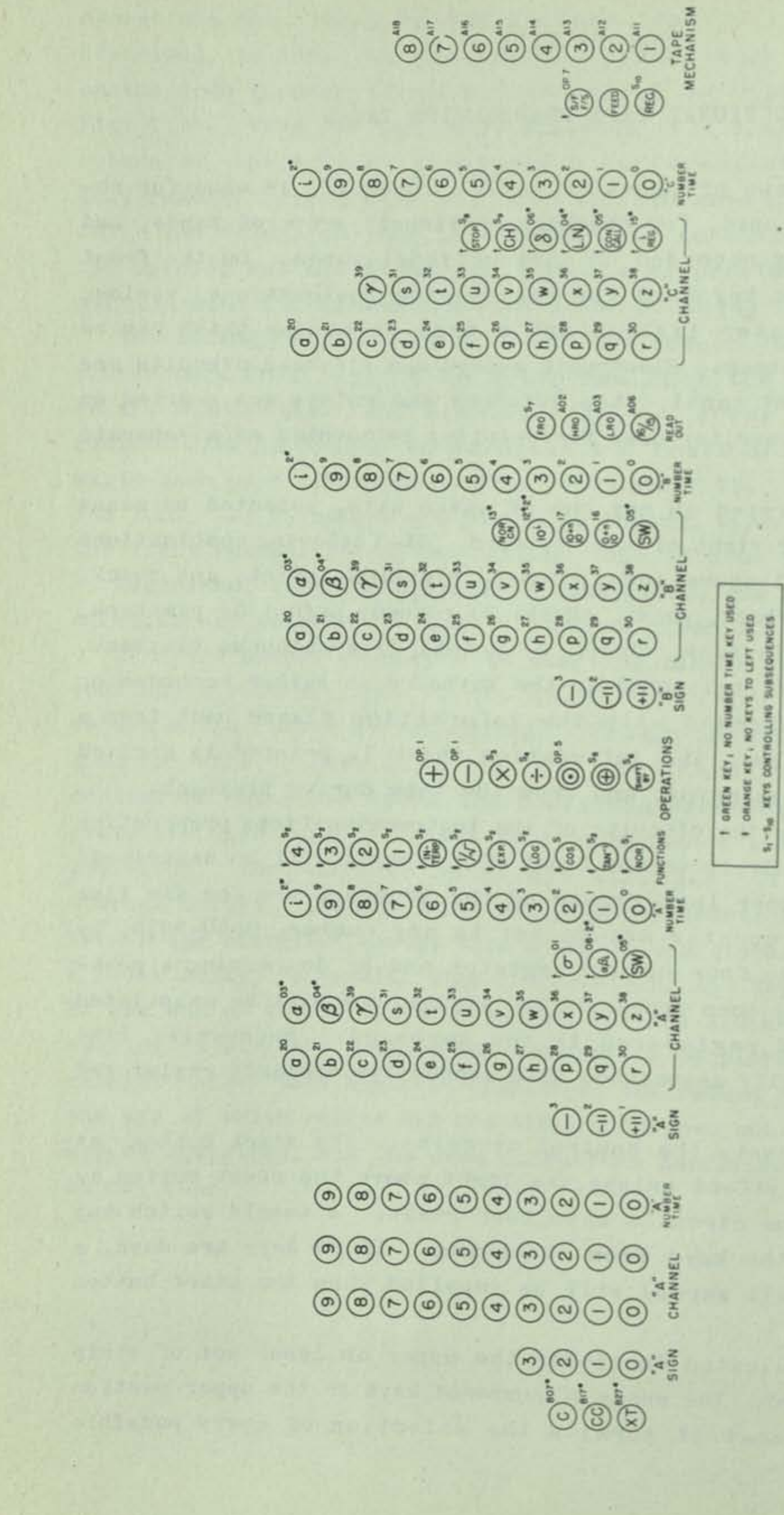


Fig. 8.1—Keys in lower section.

combination of the codes listed in Appendix II. Since certain infrequently used codes, such as those for the constants in fast storage, are not represented on the lower keys, the upper keys must be used whenever one of these codes is required.

The symbols on the lower keys represent mathematical operations, storage registers, and controls on numerical input and output tapes. When the lower register is used, it is not necessary to prepare a problem in terms of numerical codes, and furthermore certain groups of codes are supplied automatically when a single command has been registered in the keyboard.

At the left of Fig. 8.1, the keys labelled *C*, *CC*, and *XT* are used in conjunction with a four-column register to code call, conditional call, and external transfer commands. The number selected in the four-column register is coded by means of digits 3-16 of the command. If line 0000 is called, either directly or conditionally, the call will be to the line whose number has been previously stored in the line number register (*L.N.* under *C* codes). A conditional call will succeed if a positive sign has been previously stored in the conditional call register (cond. call under *C* codes). If in an external transfer the sign of the digits is of significance, then the positive absolute value or negative absolute value must be selected as the *B* transfer sign. Otherwise, the quantity being externally transferred will assume the sign of the last quantity delivered to the *B* transfer channel.

The keys for the *A* and *B* signs supply the codes for the transfer signs, minus, minus absolute value, or plus absolute value, to be associated with the quantities going to the *A* and *B* transfer channels. If no sign key is used a positive sign is supplied.

To select fast storage registers under the *A*, *B*, or *C* codes, a channel, $a,b,c,d,e,f,g,h,p,q,r,s,t,u,v,w,x,y,z,\gamma$, and a number time, $0,1,\ldots,9,\iota$, must be indicated. The ι number time refers to the digit previously stored in column 13 of the ι register. Keys for the α, β, and constant register (Sw.) channels appear under the *A* and *B* codes only, since these channels have fast storage playback connections but no provisions for recording from the *C* bus.

Under the *A* codes, the σ key calls for the last quantity recorded on the *C* transfer channel. The $\alpha\beta_{\iota}$ key selects, under control of the ι register, a quantity previously transferred from slow storage to the α or β channel. These keys are green to indicate that when they are used, no *A* number time key should be depressed. Similarly, no *B* or *C* number time key should be used when a green key under the *B* or *C* codes is selected.

TABLE 8.1

Subsequences supplied by instructional tape preparation table.

Sub-sequence	Command	Line No.	Codes				
				A	*OP*	*B*	*C*
s_1	*Norm.* $y_0 = z_0$	1	Count zeros of y	0 07 0	0	0 37 0	00 0
		2	*Norm. C.N.* = t reg.	0 15* 0	1	1 13* 0	15* 0
		3	blank	0 00 0	0	0 00 0	00 0
		4	$y \times 10^t$	0 37 0	2	0 12* 2*	00 0
		5	blank	0 00 0	0	0 00 0	00 0
		6	$SRO = z_0$	0 05 0	0	0 00 0	38 0
s_2	function $y_0 = z_0$ (except *Norm.*)	1	$y_0 = r_0$	0 15* 0	1	0 37 0	30 0
		3	Call line 3___	0 3_ _	_	0 07* 0	00 0
		2	XT 0004 = r_9	0 00 0	4	0 27* 0	30 9
		4	$r_1 = z_0$	0 15* 0	1	0 30 1	38 0
s_3	$z_0 \times y_0 = z_0$	1	$z_0 \times y_0 = z_0$	0 36 0	2	0 37 0	38 0
		2	blank	0 00 0	0	0 00 0	00 0
		3	blank	0 00 0	0	0 00 0	00 0
s_4	$z_0 + y_0 = z_0$	1	$z_0 = r_2$	0 36 0	1	0 17 8	30 2
		2	$y_0 = r_0$	0 15* 0	1	0 37 0	30 0
		4	Call line 3000	0 30 0	0	0 07* 0	00 0
		3	XT 0005 = r_9	0 00 0	4	0 27* 0	30 9
		5	$r_1 = z_0$	0 15* 0	1	0 30 1	38 0
s_5	$z_0 \oplus y_0 = z_0$	1	$z_0 + y_0 = z_0$	0 36 0	3	0 37 0	38 0
		2	$z_5 + y_5 = z_5$	0 36 5	3	0 37 5	38 5
		3	Carry	0 00 0	4	0 38 0	38 0
		4	Carry	0 00 0	4	0 38 5	38 5
s_6	z_0 shifted by $y_0 = z_0$	1	$y_0 = t$ reg.	0 15* 0	1	0 37 0	15* 0
		2	blank	0 00 0	0	0 00 0	00 0
		3	$z_0 \times 10^t$	0 36 0	2	0 12* 2*	00 0
		4	blank	0 00 0	0	0 00 0	00 0
		5	$SRO = z_0$	0 05 0	0	0 00 0	38 0
s_7	$PRO = z_0$	1	$I_b = z_0$	0 15* 0	0	0 02* 0	38 0
		2	blank	0 00 0	0	0 00 0	00 0
		3	$-I_a + z_0$	3 10 0	1	0 38 0	00 0
		4	$-\lvert\sigma\rvert + 10^0 = CSR$	2 01 0	1	0 16 0	16* 0
		5	blank	0 00 0	0	0 00 0	00 0
		6	Check	0 00 0	8	0 00 0	00 0
s_8	$z_0 + y_0$ = Stop	1	$z_0 + y_0 = CSR$	0 36 0	1	0 37 0	16* 0
		2	blank	0 00 0	0	0 00 0	00 0
		3	Stop	0 00 0	9	0 00 0	00 0
s_9	$z_0 + y_0$ = Check	1	$z_0 + y_0 = CSR$	0 36 0	1	0 37 0	16* 0
		2	blank	0 00 0	0	0 00 0	00 0
		3	Check	0 00 0	8	0 00 0	00 0
s_{10}	$z_0 + y_0$ = Rec. 1	1	$z_0 + y_0$ = Rec. reg.	0 36 0	1	0 37 0	02* 0
		2	blank	0 00 0	0	0 00 0	00 0
		3	blank	0 00 0	0	0 00 0	00 0
		4	$\sigma - I_b$	0 01 0	1	3 02* 0	00 0
		5	$-\lvert\sigma\rvert + 10^0 = CSR$	2 01 0	1	0 16 0	16* 0
		6	blank	0 00 0	0	0 00 0	00 0
		7	Check	0 00 0	8	0 00 0	00 0
		8	Select mech. 1	0 11 0	0	0 00 0	00 0

When an orange key is selected on the keyboard, no keys to the left of it may be used simultaneously. Since the function keys are all orange the argument for which a function is required must be selected under the *B* codes. Each function key controls a subsequence of commands as indicated in Table 8.1. The Norm. key supplies the six lines of coding needed to normalize a quantity, that is, to shift it to the left until the first non-zero digit is in column 16. The first command of the subsequence is to count the number of zeros to the left of the first non-zero digit in the argument. The second is to deliver the normalizing control number (*Norm. C.N.*) from the zero-counting circuit to the t register. The third line is blank, to permit the normalizing control number to be stored in the t register. On the fourth line the argument is multiplied by 10^t, the power of 10 selected by the t register. On the last line, the code for a shift read out (*SRO*) is given.

The remaining function keys each supply four-line subsequences, to refer to the sections of the instructional storage drum containing the appropriate computing routines. The first line of each subsequence transfers the argument to register r_0. The second transfers to register r_9 the line number to which the calculator must refer at the end of the function computation. The third command calls the line number, Table 8.2, at which the particular function computation begins. The fourth line of the subsequence, to which the calculator refers at the end of the function computation, transfers the quantity in r_1 to the register originally selected under the *C* codes. Thus it is assumed that the coding for each function computation starts with an argument in r_0, stores the computed function in r_1, transfers the quantity in r_9 to the line number register, and then calls the command having this line number.

TABLE 8.2

Starting lines for function computations on instructional storage drum.

Function	Start Line Number
Division	3000
Reciprocal square root	3060
Exponential	3130
Logarithm	3200
Cosine	3264
Arc tangent	3340
Interpolation	3460
Function 1	3600
2	3700
3	3800
4	3900

The coding for the computation of the functions is given in Chapter V and has been recorded on the instructional storage drum starting at the specified line numbers. Any other functions required in a particular problem may be coded starting on the lines indicated in Table 8.2.

The keys to the right of the function keys indicate the operations which may be performed on quantities selected under the *A* and *B* codes. The add and subtract operations, + and -, both supply the same operation code. However, when the - key is used, the *B* transfer sign is automatically inverted. The three lines of coding required for a multiplication are supplied when the × key is used. A division is initiated by depressing the ÷ key. This supplies a five-line subsequence, Table 8.1, in order to refer to the section of the instructional storage drum which contains the computing routine for a division.

When the ⊙ key is selected, the operation code given is for a sign choice, causing the quantity selected under the *B* codes to be multiplied by the sign of the quantity selected under the *A* codes. A double-accuracy addition is coded under control of the ⊕ key, which supplies a four-line subsequence. The registers selected under the *A*, *B*, and *C* codes must be those containing the low-order digits, and the conventions for double-accuracy work given in Chapter I should be followed. The coding for a shift operation is supplied under control of the button, *shifted by*. In this case the quantity selected by the *A* codes is the argument to be shifted. The digits in columns 13 and 14 of the quantity selected by the *B* codes determine the number of columns of shift, while the direction of the shift is left or right, depending on whether the sign of the *B* quantity is plus or minus. The coding is essentially the same as the last five lines of the subsequence for normalization. Finally, to code an identity-check operation, the 6 key in the operation column of the upper register must be used. It is advisable to code a blank line following an identity check, in order that in the event of a check failure the quantities being compared will be preserved on the *A* and *B* transfer channels for inspection.

Under the *B* registers, the Norm. C.N. key selects the normalizing control number coming from the zero-counting circuit. When the Norm. C.N. key is used, the *B* transfer signs, plus and minus, are recorded as positive and negative absolute value codes, respectively.

The powers of ten required for shifts are stored in two fast storage channels selected by the keys 10^{0+n} and 10^{10+n} under the *B* codes: the *B* number time is used to select the particular register required and also fixes the value of n. The combination 10^{10+n}, number time 8, selects the code to record zeros on the *B* transfer channel.

The key 10^{i} supplies the code for a power of 10 selected by the i register. When a positive number n, $0 \leq n \leq 15$, has been stored in the i register, $10^{i} = 10^{n}$. When a negative number n, $-15 \leq n \leq 0$, has been stored in the i register, $10^{i} = 10^{16-|n|}$. In either case, the selected power of 10 may be used in a multiplication to shift a quantity n columns, the direction of the shift being determined by the sign of n.

The three keys *HRO*, *LRO*, and 16/15 *RO* supply the codes for special products from the multiplier. These may be called for on the last cycle of a multiplication or at any time after a multiplication, before the multiplier is used again. *HRO* and *LRO* refer to the high- and low-order product digits, respectively, while 16/15 *RO* refers to the product digits, 31-16, normally delivered for an operating decimal point between columns 16 and 15.

The *FRO* key, for a feed read-out, and the feed key, at the far right

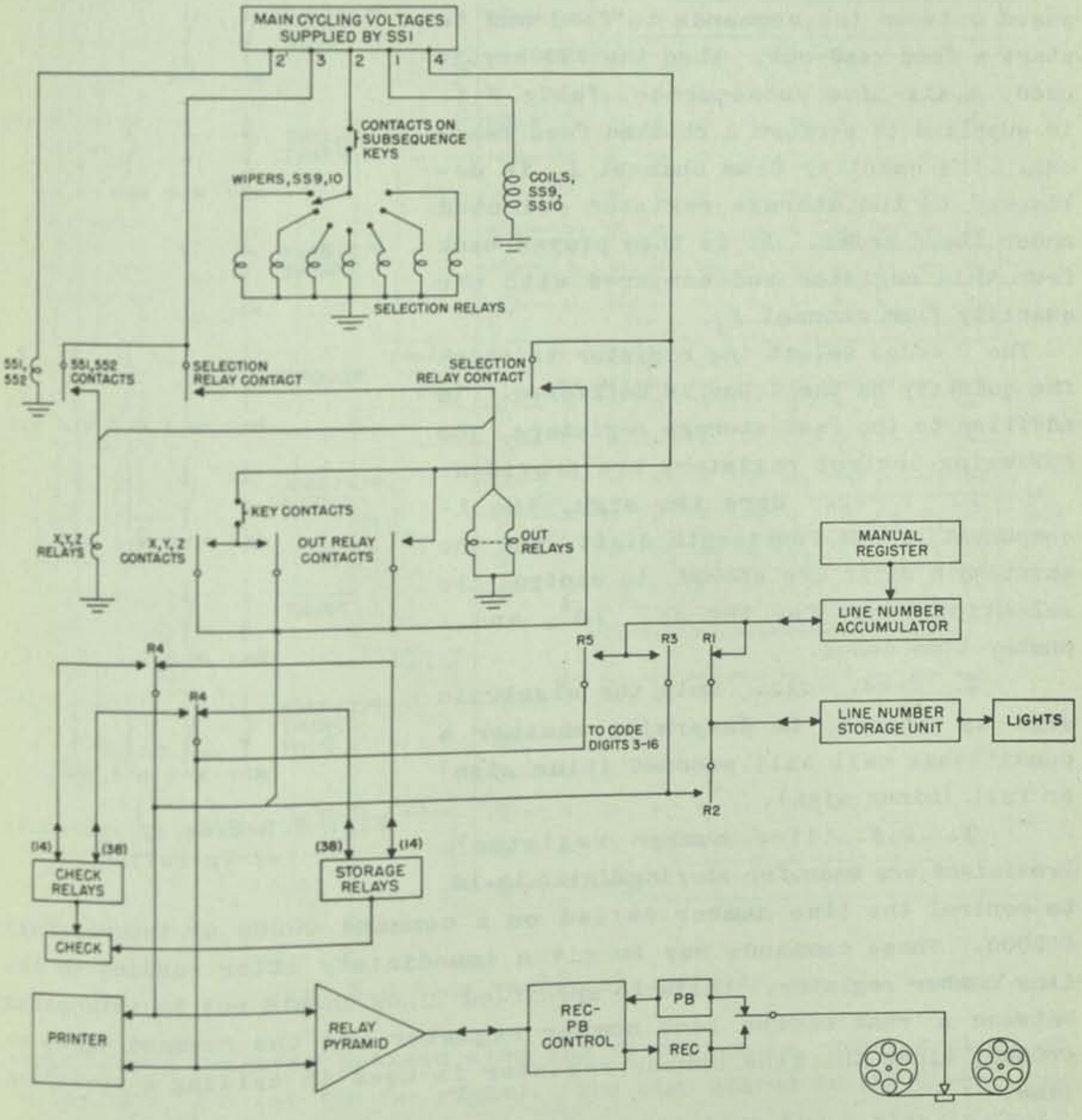


Fig. 8.2—*Block diagram.*

of the keyboard, are used in reading a quantity into the calculator from a numerical input tape. The single line of coding required to play back one quantity from tape to the two intermediate storage channels, I_a and I_b, is supplied by depressing the keys, *Feed n*, where *n* is a mechanism number, 1-8. Then either the command for a feed read-out may be given immediately after the *Feed* command, in which case the calculator automatically waits approximately twenty-five cycles until the reading has been completed; or operations not involving the tape read-record system may be interposed between the commands to feed and to start a feed read-out. When the *FRO* key is used, a six-line subsequence, Table 8.1, is supplied to perform a checked feed read-out. The quantity from channel I_b is delivered to the storage register selected under the *C* codes. It is then played back from this register and compared with the quantity from channel I_a.

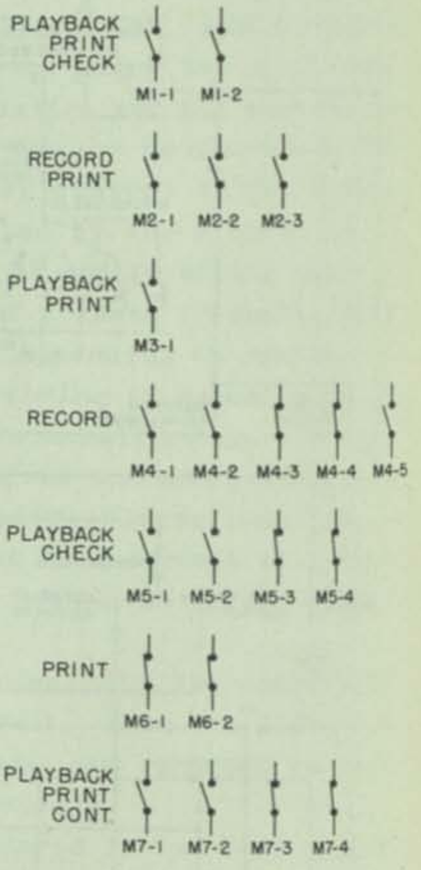


Fig. 8.3—*Mode of operation strip switch.*

The *C* codes select the register to which the quantity on the *C* bus is delivered. In addition to the fast storage registers, the following control registers are provided:

1. *i reg.* Here the sign, the 1-component of the fourteenth digit, and the thirteenth digit are stored, to control the selections made for the $\alpha\beta_i$, 10^i, and i number time codes.

2. *Cond. call.* Only the algebraic sign is stored, to determine whether a conditional call will succeed (plus sign) or fail (minus sign).

3. *L.N.* (line number register). Provisions are made for storing digits 13-16, to control the line number called on a command *C*0000 or (successful) *CC*0000. These commands may be given immediately after reading to the line number register. Calls to specified lines should not be interposed between a read to the line number register and the command *C*0000 or *CC*0000, since the line number register is used in calling a specified line.

4. δ. The algebraic sign and digits 14-16 are delivered to circuits

controlling a relay pyramid for the selection of slow storage channels. Twelve cycles must elapse between reading into this register and the

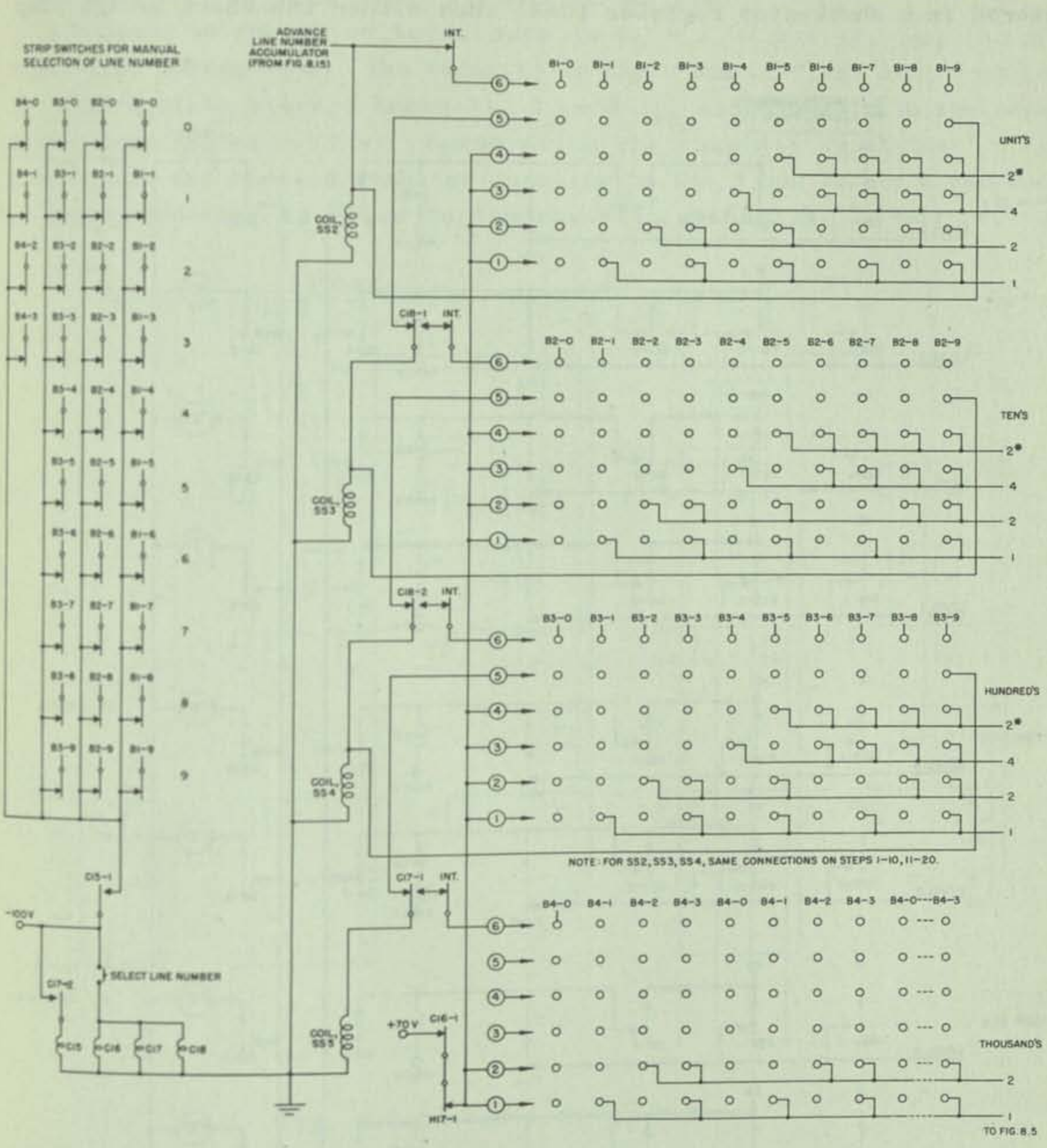


Fig. 8.4—*Line number accumulator.*

command for a transfer between slow and fast storage, under control of the key, *S/F*, *F/S* (at the far right). The sign stored in the δ register determines the direction of the transfer, positive for a slow to fast, and negative for a fast to slow storage transfer.

5. *Check.*

6. *Stop.* The algebraic sign of the quantity on the *C* bus is stored in a check-stop register (*CSR*) when either the check or the stop

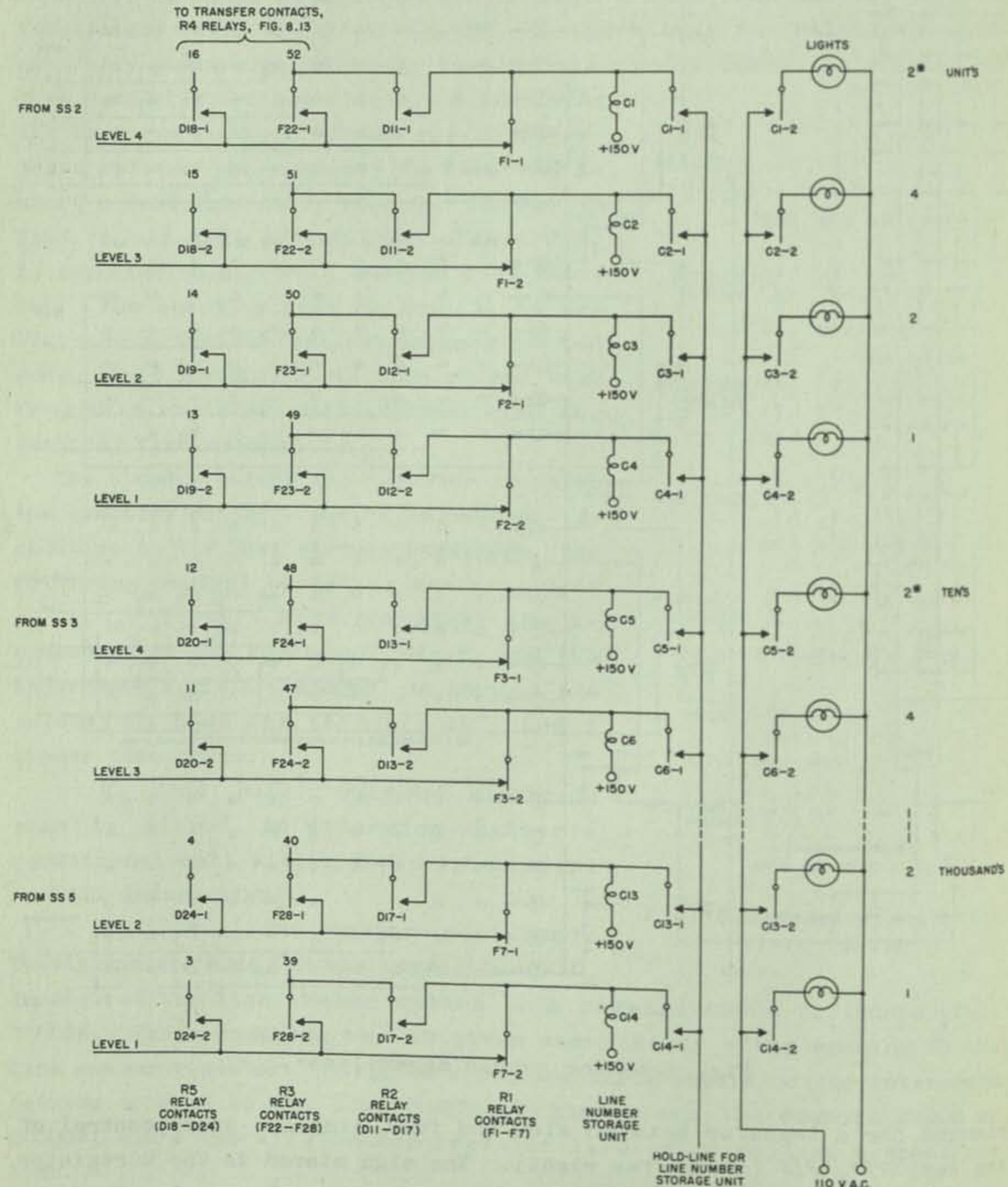


Fig. 8.5—*Line number storage unit and lights.*

key is used. Three commands are recorded as shown in Table 8.1. If a negative sign has been stored, the calculator will stop, and either the check-stop or code-stop light will be turned on.

A command to record on tape, such as $a_0 + b_1$ = Rec. 6, supplies an eight-line subsequence. The quantity on the C bus is first delivered to the intermediate storage channels, I_a and I_b, associated with the tape read-record system. After regeneration the quantity on channel I_b is played back and checked with the quantity on the C bus before a command to begin recording is given to the selected mechanism. A tape record

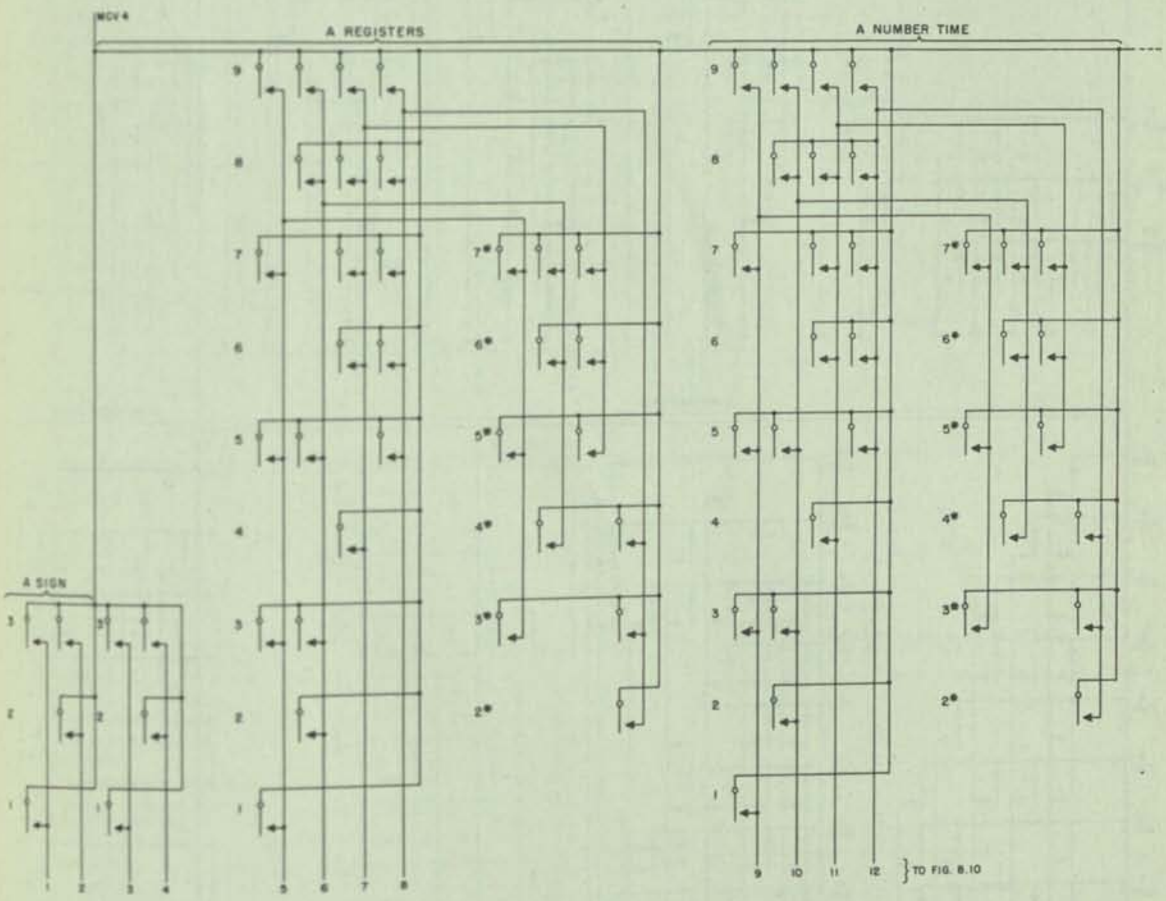


Fig. 8.6—*Contacts on upper keys.*

operation takes about twenty-five cycles, during which the calculator may perform any operations not involving the tape read-record system. However, the machine will automatically wait if any such commands are given before the recording has been completed.

To transfer a quantity from one register to another in the calculator, it is necessary to add the quantity to zero and then record from the

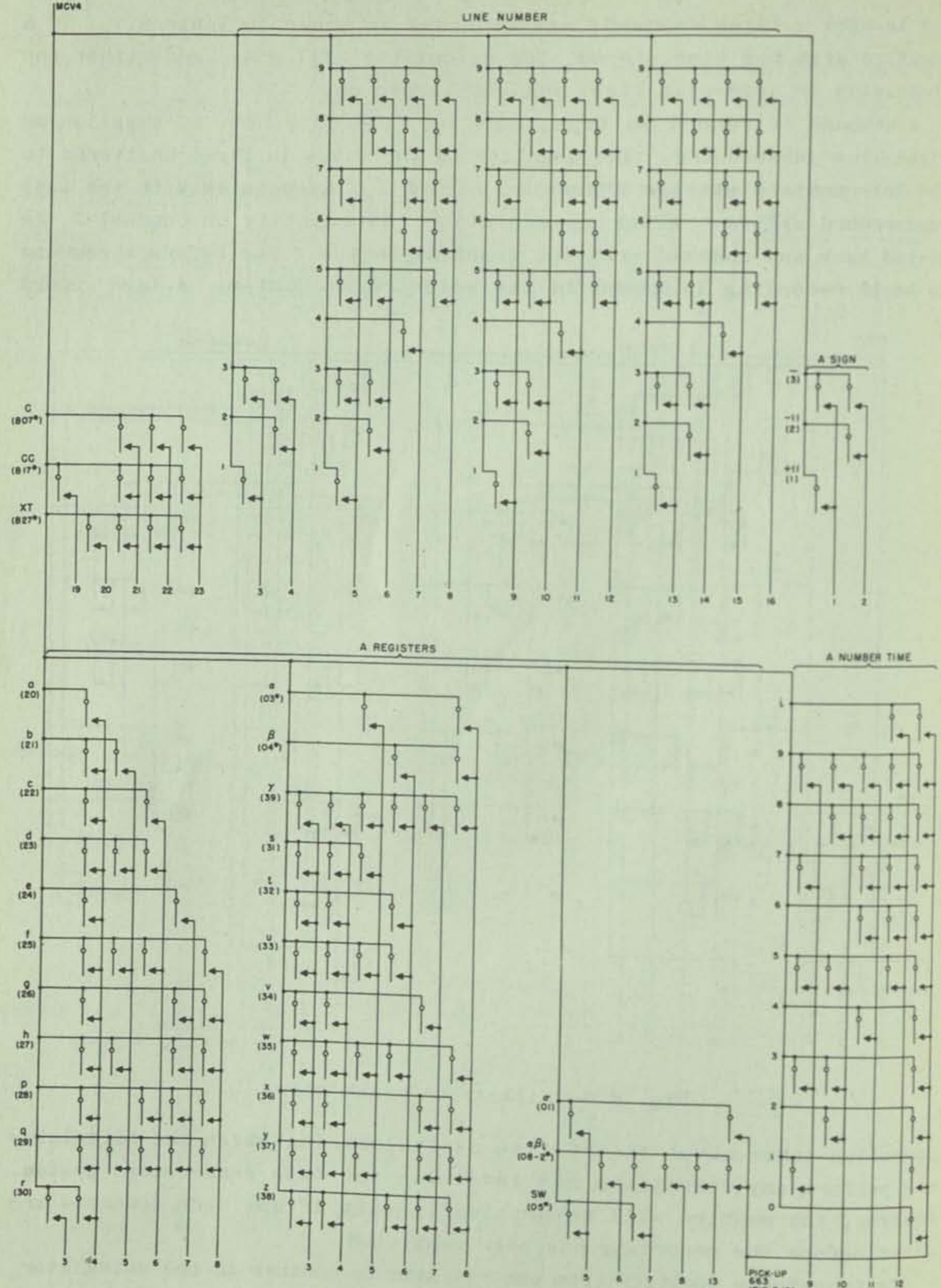


Fig. 8.7—*Contacts on lower keys — line number and A code controls.*

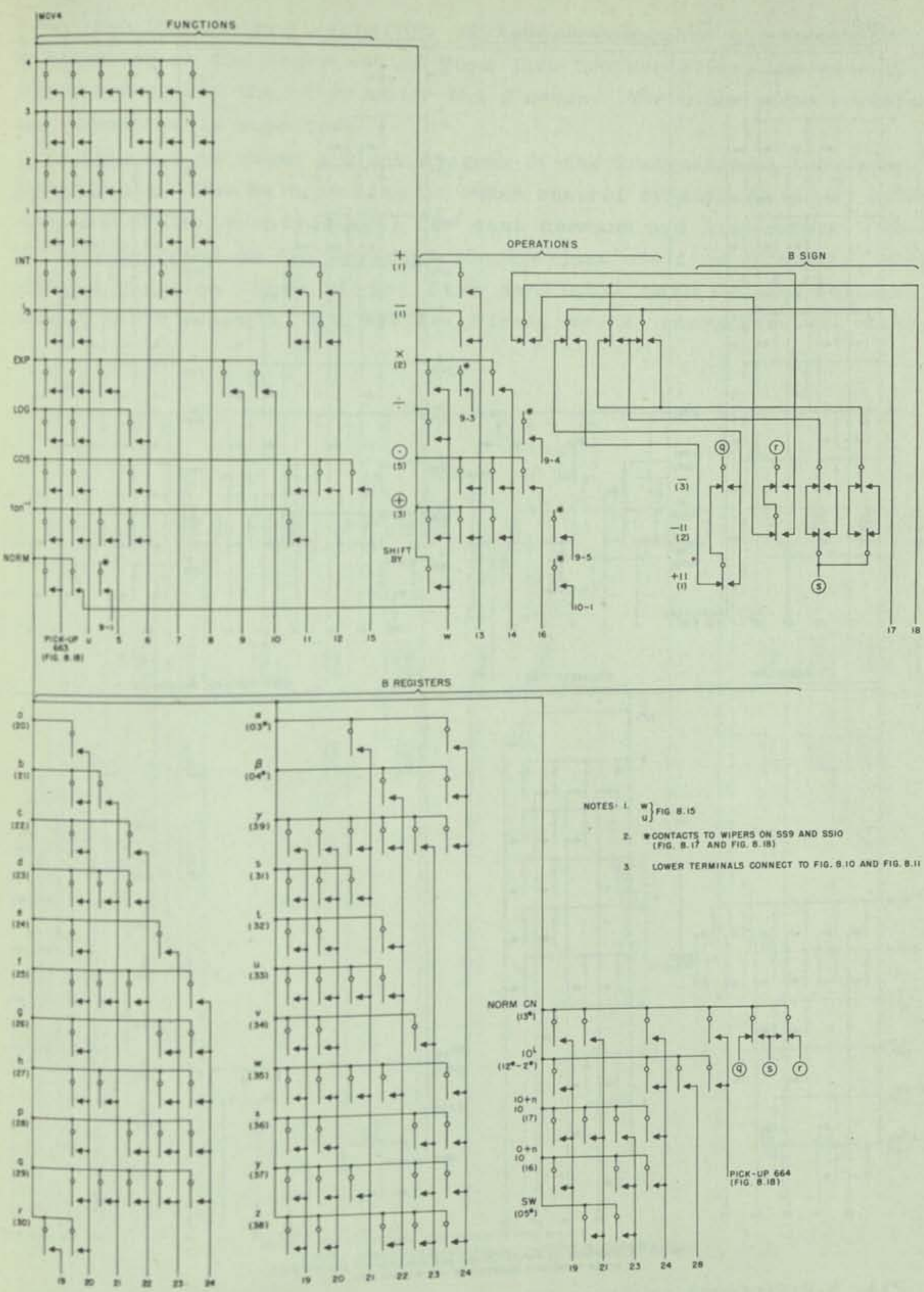
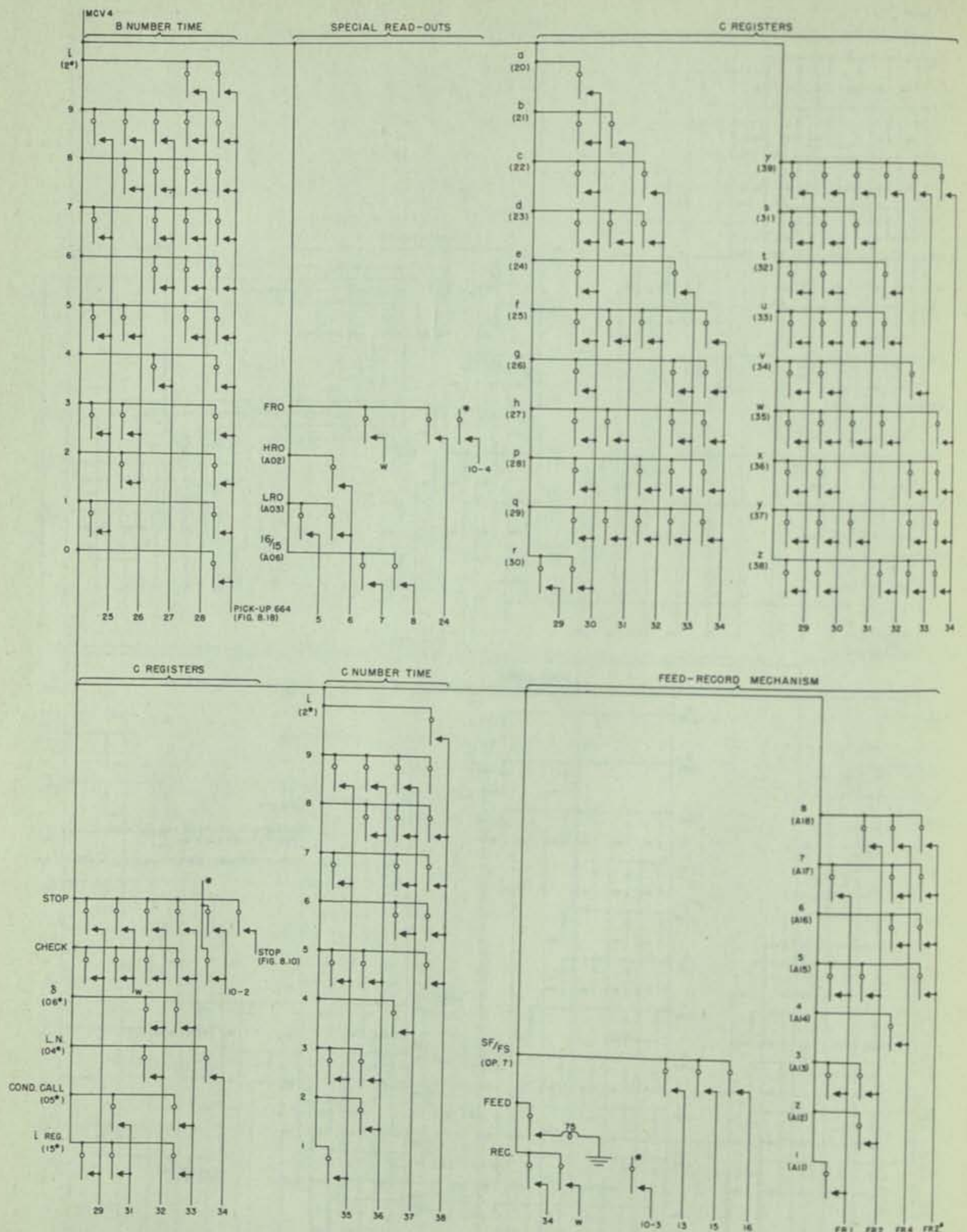


Fig. 8.8—*Contacts on lower keys — functions, operations and B registers.*

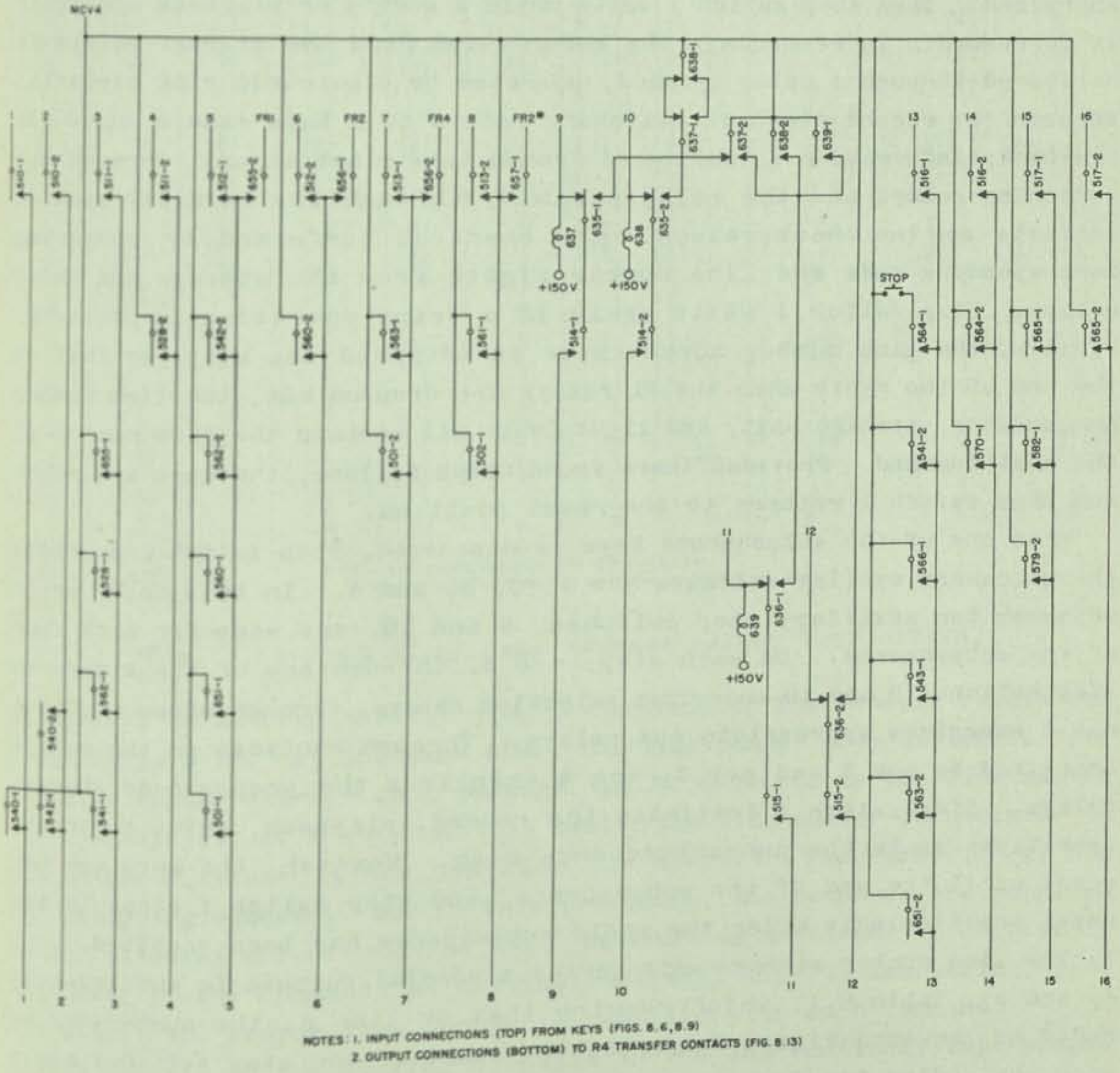


NOTES: 1. *CONTACTS TO WIPERS ON 559 AND 5510 (FIG. 8.17 AND FIG. 8.18)
2. LOWER TERMINALS CONNECT TO FIG. 8.11 AND 8.12

Fig. 8.9—*Contacts on lower keys — B number time and C code controls.*

C bus into the desired register. A transfer command may be indicated by selecting on the lower set of keys just two registers, one under the *A* or *B* codes, and the other under the *C* codes. The other codes required are automatically supplied.

In Fig. 8.2 is shown a block diagram of the instructional tape preparation table. The main cycling is under control of step switch 1, which makes one cycle (twenty steps) for each command and line number. When the keys selected on the keyboard require just one line of coding, step switch 1 supplies three of the five available main cycling voltages, namely, *mcv* 2′, *mcv* 3, and *mcv* 4. First, *mcv* 2′ energizes two relays,



NOTES: 1. INPUT CONNECTIONS (TOP) FROM KEYS (FIGS. 8.6, 8.9)
2. OUTPUT CONNECTIONS (BOTTOM) TO R4 TRANSFER CONTACTS (FIG. 8.13)

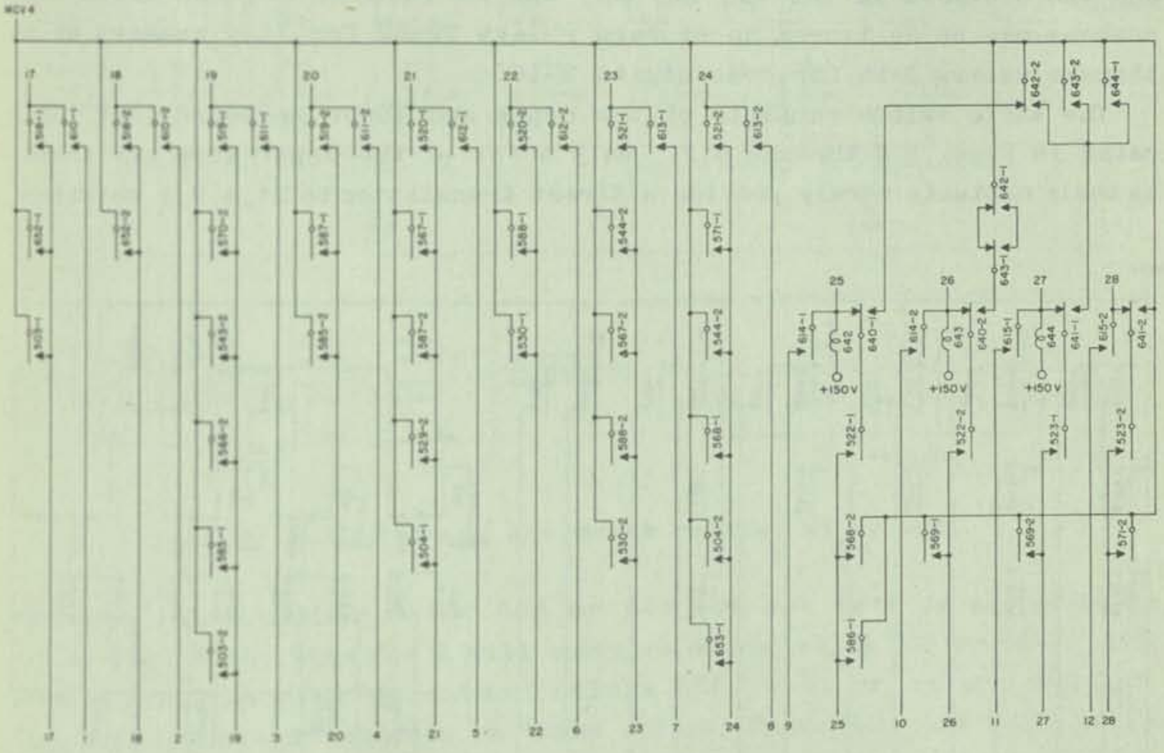
Fig. 8.10—*Control of code storage relays* 1-16.

551 and 552, through whose contacts *mcv* 3 energizes three groups of relays, *X*, *Y*, *Z*. Then *mcv* 4, through contacts on the depressed keys, through contacts on the *X*, *Y*, *Z* relays, and through the normally closed *R*4 contacts, energizes corresponding code storage relays. Relays *R*1 are energized in order to reset the storage unit and lights associated with the line number accumulator. When the *R*3 relays are picked up, the line number to be associated with the command now in the code storage relays is delivered from the line number accumulator through the *R*3 contacts, through normally closed *R*4 contacts to the fourteen line number storage relays.

If the selected mode of operation requires a check, the *R*4 relays are energized. Then step switch 1 waits while a record or playback operation is performed. In recording, the information from the storage relays is delivered through a relay pyramid, operated by electronic ring circuits, through the record-playback switching relays to a tape record unit. In playback, information is delivered from a tape playback unit through the switching relays and the relay pyramid, through the normally open *R*4 contacts to the check relays. The check is performed by comparing corresponding code and line number digits from the storage and check relays. Step switch 1 waits again if a print operation is required. Finally, the line number accumulator is advanced one step, so that at the end of the cycle when the *R*1 relays are dropped out, the line number accumulator, storage unit, and lights will all contain the line number of the next command. Provided there is no check failure, the keys are reset and step switch 1 returns to the reset position.

When one of the subsequence keys is depressed, step switch 1 provides the necessary cycling voltages *mcv* 1, 2, 3, and 4. In this case, *mcv* 1 advances two auxiliary step switches, 9 and 10, one step for each line of the subsequence. On each step, *mcv* 2, through one of the wipers on step switches 9 and 10 energizes selection relays, through whose contacts *mcv* 3 energizes appropriate out relays. Through contacts on the relays energized by *mcv* 2 and *mcv* 3, *mcv* 4 energizes the proper code storage relays. Step switch 1 initiates the record, playback, print, or check operations as in the non-subsequence case. However, the keys are not reset until the end of the subsequence, and step switch 1 stops in the reset position only after the whole subsequence has been supplied.

The line number storage unit serves a special purpose in subsequences s_2 and s_4, Table 8.1, which require that on line n, the number $n+2$ be coded on sequence digits 3-16. At the start of the step switch 1 cycle when the coding for line n would normally be supplied, the *R*1 relays and the relay hold-line for the line number storage unit are energized, leaving n stored in the line number storage unit. Then the line number accumulator is advanced one step, to $n+1$, and the line number $n+1$ and the coding for this line are supplied in the usual way. At the end of this cycle the line number accumulator is advanced again, to the number $n+2$. On the next cycle, relays $R2$ and $R5$ are energized, permitting the



NOTES: 1. INPUT CONNECTIONS (TOP) FROM KEYS (FIGS. 8.6, 8.9)
2. OUTPUT CONNECTIONS (BOTTOM) TO R4 TRANSFER CONTACTS (FIG. 8.13)

Fig. 8.11—*Control of code storage relays* 1-12, 17-28.

number $n+2$ to be read through the $R5$ contacts to the storage relays for code digits 3-16. At the same time, the line number n is delivered from the line number storage unit through the $R2$ contacts to the line number storage relays. At the end of this cycle, the line number accumulator is not advanced, since it now contains the correct line number, $n+2$, for the following command. Due to this procedure, the commands are supplied in the following order: $n+1$, n, $n+2$. However, as mentioned in Chapter I, the order of the commands on the instructional tape is immaterial, provided that the proper line number is supplied with each command.

The details of the cycling circuits in the instructional tape preparation table are shown in Figs. 8.3 through 8.18. The strip switch contacts controlling the mode of operation are shown in Fig. 8.3. The line number accumulator, Fig. 8.4, consists essentially of four step switches (*S.S.* 2, 3, 4, 5), connected as decade counters. The manual controls on the register are indicated at the left of the drawing. Figure 8.5 shows the storage unit, consisting of relays *C*1-*C*14, associated with the line number accumulator. Also shown are the display lights and the contacts on the *R*1, *R*2, *R*3, and *R*5 relays, through which line numbers may be delivered to storage relays 39-52 for line numbers or to storage relays 3-16 for code digits 3-16.

The strip switch contacts of the upper and lower keyboards are indicated in Figs. 8.6 through 8.9. Only a few of the upper keys are shown, as their contacts merely provide a direct translation to 2*,4,2,1 notation.

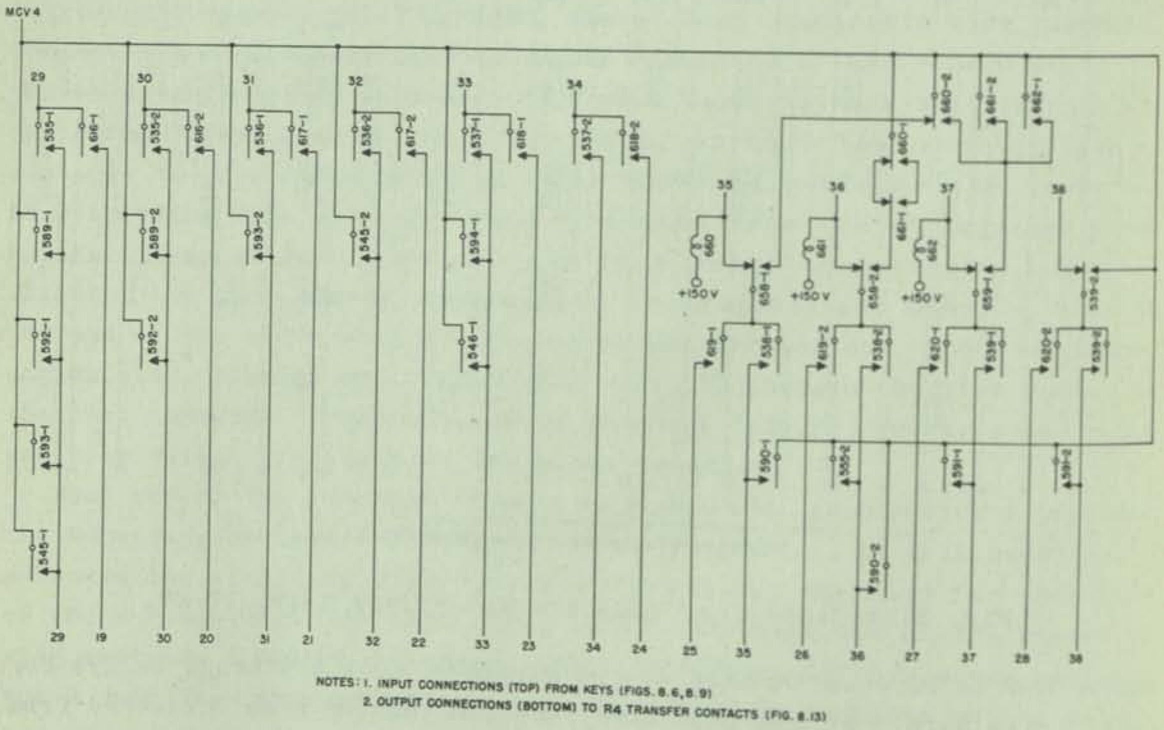


Fig. 8.12—*Control of code storage relays* 19-38.

Since the terminals numbered 1-38 in Figs. 8.6-8.9 are connected to corresponding terminals in Figs. 8.10-8.12, the code storage relays will be energized by *mcv* 4 through the contacts on the out relays picked up by *mcv* 2′, *mcv* 2, and *mcv* 3.

As shown in Figs. 8.7, 8.8, and 8.9, the contacts controlling relays 663 and 664 sense for transfer commands requiring that the codes for addition and for recording zeros on the *A* or *B* transfer channel be

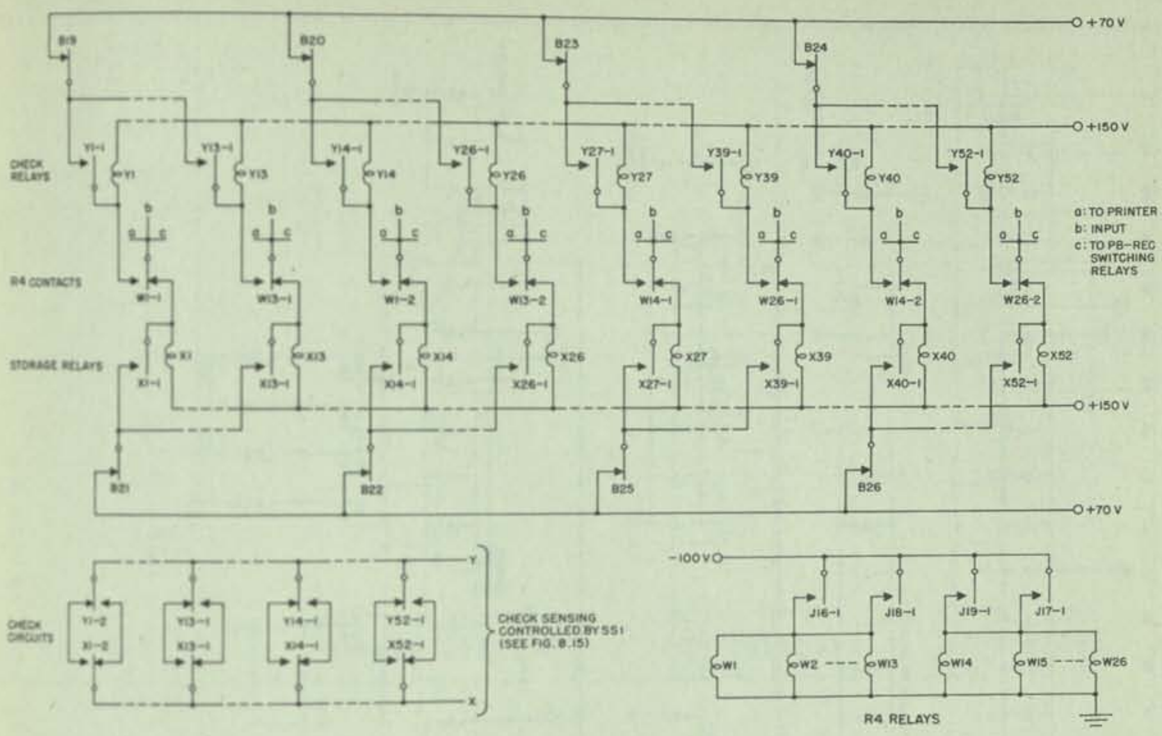


Fig. 8.13—*Storage and check relays, R4 relays.*

supplied. When either relay 663 or 664 but not both is energized by *mcv* 4, Fig. 8.18, then *mcv* 2 will energize either relay 666 or 665. This permits *mcv* 3 to pick up either relays 562, 563, or relays 566-569, Fig. 8.16. Through contacts on these relays, Figs. 8.10 and 8.11, *mcv* 4

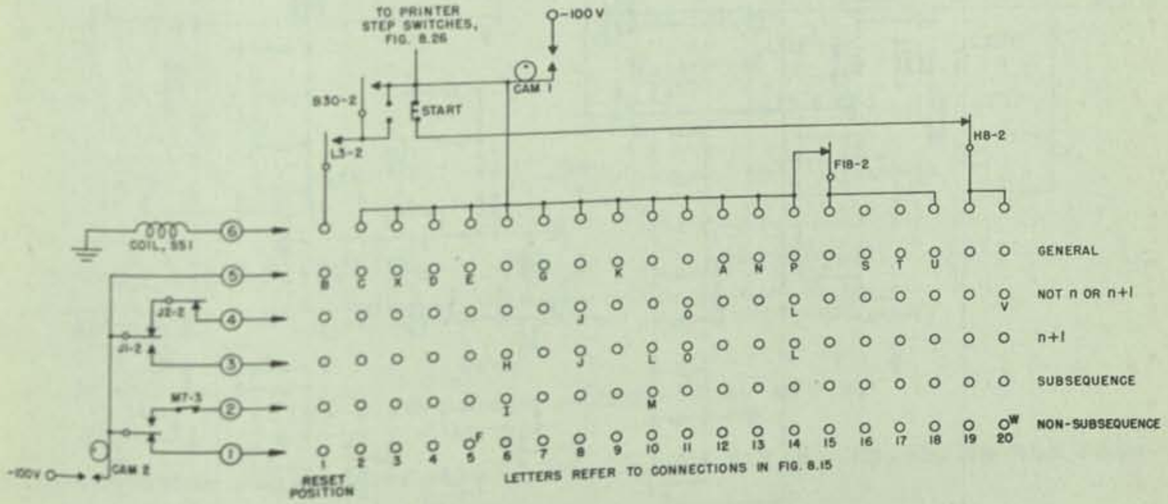


Fig. 8.14—*Step switch 1.*

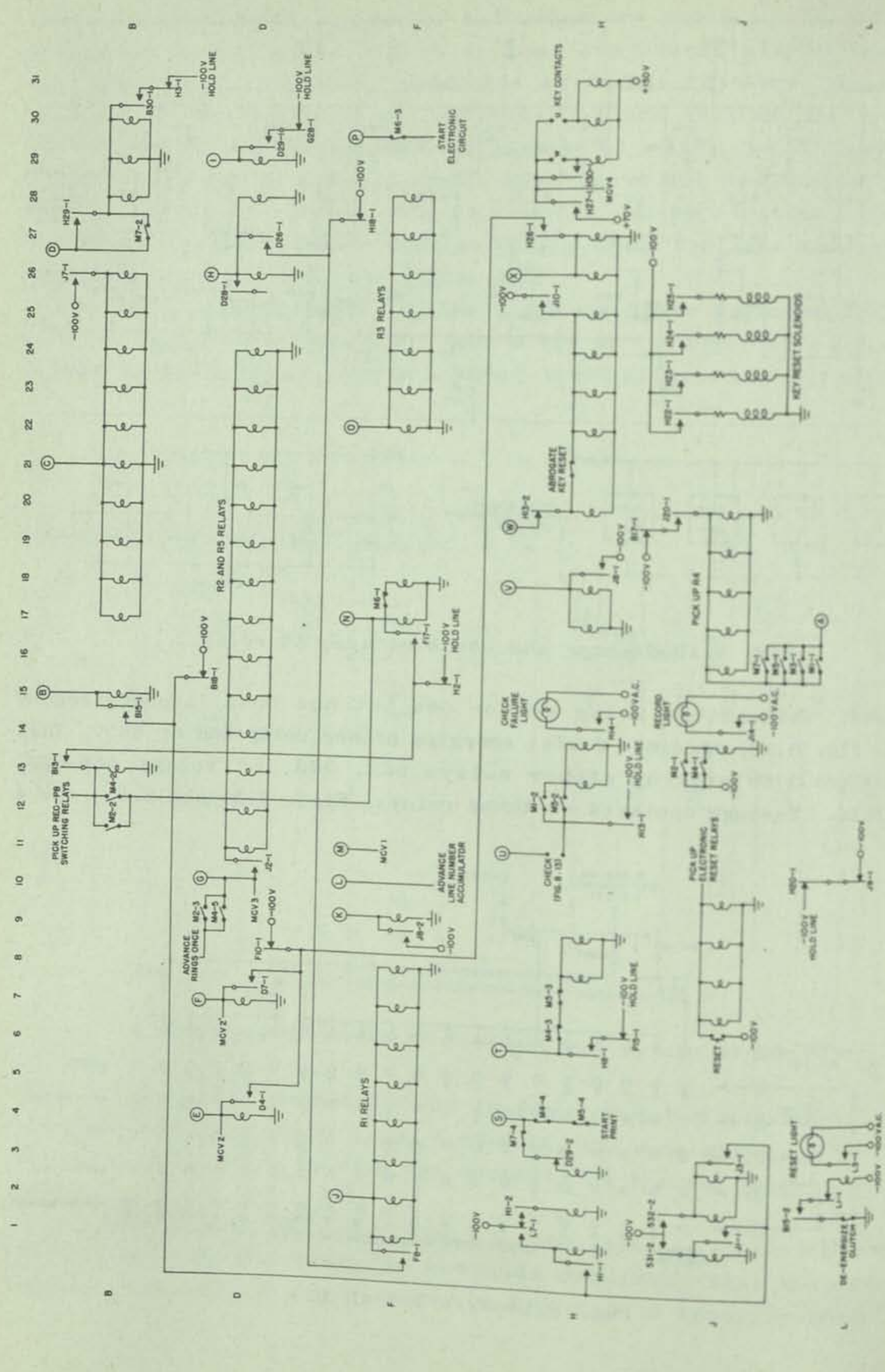


Fig. 8.15—Relays controlled by step switch 1.

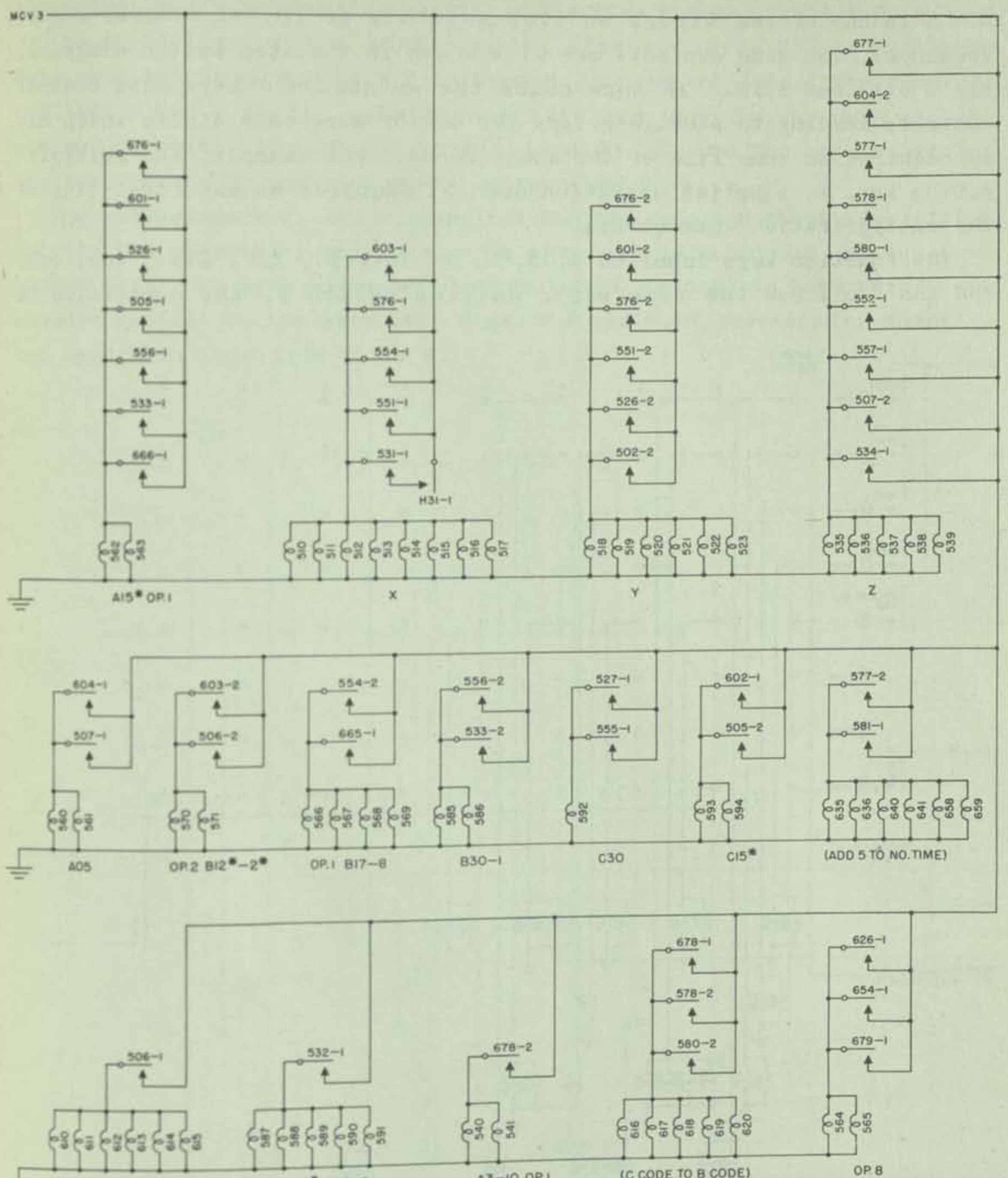


Fig. 8.16—*Relays energized by mcv* 3.

delivers either the codes *A*15*, Op. 1, or *B*17-8, Op. 1, to the code storage relays.

The contacts marked with an asterisk in Figs. 8.7 and 8.8 connect *mcv* 2 to one of the wipers on step switch 9 or 10, to control a subsequence. The same contacts are also shown in the step switch diagrams, Figs. 8.17 and 8.18. In some cases the subsequence keys also control contacts leading to storage relays for one or more code digits which may be required on some line of the subsequence. For example, the multiplication key, ×, supplies operation code 2, required on the first line of the multiplication subsequence.

The function keys labelled 4, 3, 2, 1, Interp., 1/√, Exp., log, cos, and $\tan^{-1}$ all use the same wiper on step switch 9, the connection to

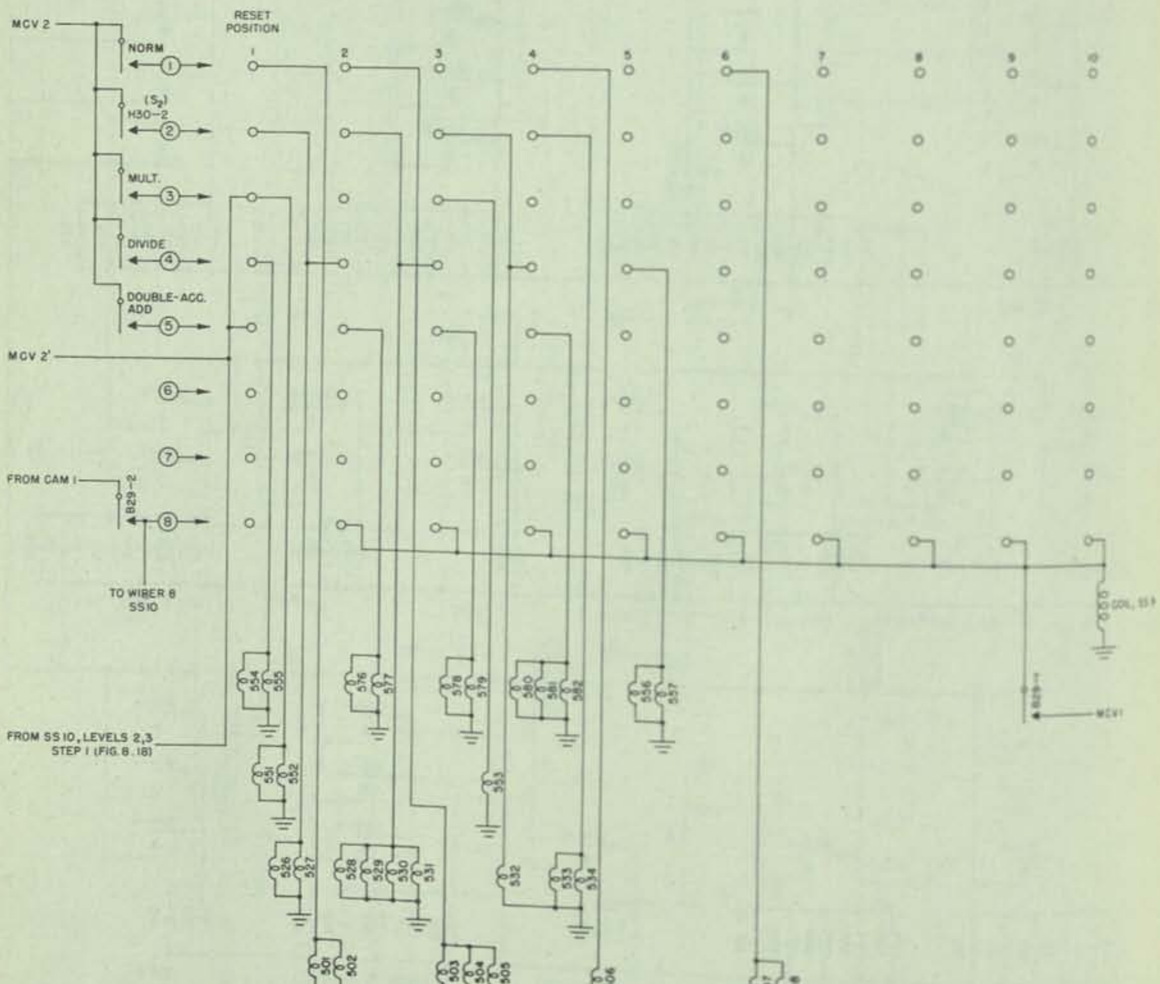


Fig. 8.17—*Step switch* 9.

which is controlled by the relay contact *H*30-2. This is because the subsequences supplied for these keys, Table 8.1, differ only in the hundred's, ten's, and unit's digits of the line number, Table 8.2, called on the third line of the subsequence. These digits are in each case delivered to the code digits 5-16 through the key contacts. The contacts labelled u and w on the subsequence keys are used in the main cycling control circuits, Fig. 8.15, to distinguish between the keys supplying the subsequence s_2 and the other subsequence keys.

The subsequences s_8 and s_9 supplied for the stop and check keys differ only in the operation code given on the last line of the subsequence. Accordingly, the same wiper on step switch 10 is used in both cases, and an extra contact on the stop key, Figs. 8.9 and 8.10, provides for changing the operation code from 8 to 9.

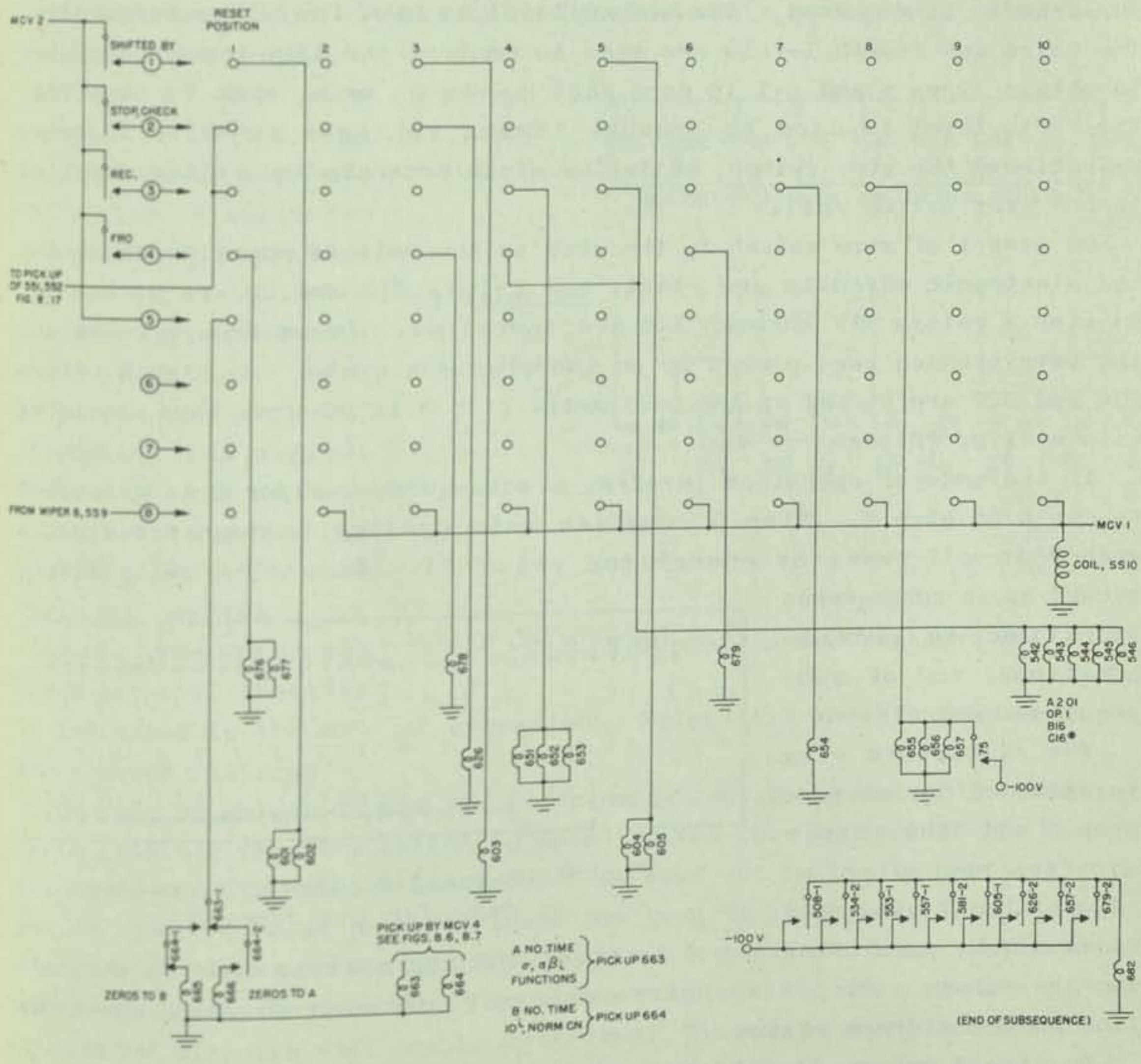


Fig. 8.18—*Step switch* 10.

When the feed key is depressed, *mcv* 4 energizes relay 75, Fig. 8.9, causing relays 655, 656 and 657, Fig. 8.18, to be energized. Then the code for the mechanism selected, *A*11-*A*18, will be delivered to the code storage relays directly, Fig. 8.10. When a mechanism is selected for recording, however, relays 655, 656, and 657 are not energized until the eighth line of the record subsequence, since in recording a check is made before the command to select a mechanism is given.

The storage and check relays are indicated in Fig. 8.13, along with the *R*4 relay contacts through which the code and line number relays are energized. Also shown are the comparison circuits used in checking.

Step switch 1, Fig. 8.14, together with the relays shown in Fig. 8.15, generates the main cycling voltages. The first level is used when no subsequence is required. The second level is used for all subsequences. The third and fourth levels are used to control the line number register to obtain lines n and n+1 in case subsequence s_2 or s_4 must be supplied. The fifth level is used to produce timing voltages required in every operation of the step switch, while the sixth controls the voltage applied to the step switch coil.

On step 1 of step switch 1, the step switch coil is energized provided the electronic circuits are reset, and relays *B*15 and *L*3 are picked up. On step 2 relays *B*17 through *B*26 are energized. These relays reset all the relays which were picked up on the previous cycle. On step 3 relays *H*26 and *H*27 are picked up and held until step 9 is reached, thus providing main cycling voltage *mcv* 4.

If the mode of operation involves a subsequence, wiper 2 is connected to cam 2 on step 4. Step 5 supplies main cycling voltage *mcv* 2 until step 9 in all cases by energizing relay *D*4. On step 6 relay *D*29 is picked up in subsequence operation to provide additional end of subsequence controls.

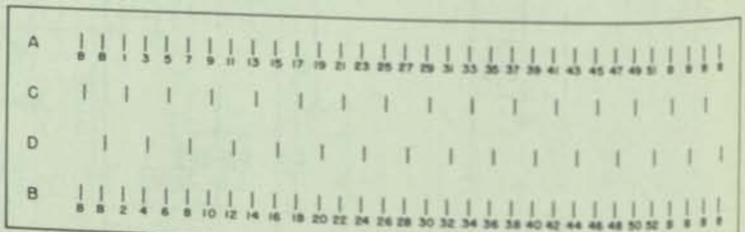


Fig. 8.19—*Relative timing of pulses on tape for one command.*

The electronic ring is advanced one step on step 7 if the cycle involves recording on tape. Step 7 also provides *mcv* 3, permitting *mcv* 4 to store the appropriate code in the code storage relays. Step 8 energizes relays *F*1 through *F*8 which reset the line number storage relays.

On step 9 relays *D*4, *D*7, *H*26, and *H*27 are dropped out and relay *F*10 is picked up. This turns off *mcv* 2 and *mcv* 4 as well as *mcv* 2′ in the non-subsequence operations. In the subsequence cycles, step 10 supplies *mcv* 1 to energize the coils of step switches 9 and 10. On step 11 the line number is transferred from the line number accumulator to the storage relays. Step 12 allows relays *J*17 through *J*20 to be picked up; these relays reposition the *R*4 relays. If the mode of operation calls for recording on the magnetic tape the record-playback switching relays are energized on step 13 allowing the information stored in the relays to be transferred to the recording circuits. In addition, step 13 allows relays *F*18 and *F*17 to be energized, thus providing power to the clutch start coil.

On step 14 the line number accumulator is advanced once and, in the case of playback or record operations, step switch 1 is made to wait until the completion of the desired electronic operation. Step 15 is blank and serves as a wait point for step switch 1. At the end of a record or playback cycle step 16 initiates the printing operation if printing is indicated and ends any subsequence which may have been supplied. On step 17, if printing is in progress relays *H*8 and *H*9 are energized in order to provide a suitable wait for step switch 1 on step 19. Step 18 is the check point if checking is indicated in the mode of operation. Relays *H*13 and *H*14 are energized for a check failure.

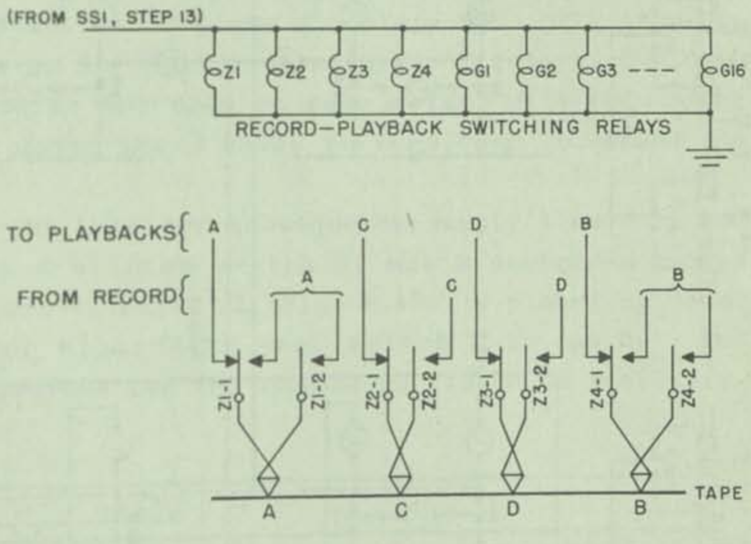


Fig. 8.20—*Pole piece connections.*

On step 19 relays *H*8 and *H*9 are de-energized at the end of the printing cycle allowing the step switch to move to step 20. On step 20 the number stored in the line number accumulator is read out to the indicator lights and the keys are reset provided there has been no check failure on step 18.

Additional circuits associated with step switch 1 are required to obtain the necessary cycling for lines n and $n+1$ of subsequences s_2 and s_4. These circuits will now be described. The relays energized by *mcv* 3, Fig. 8.16, are picked up through contacts on relays energized by *mcv* 2′

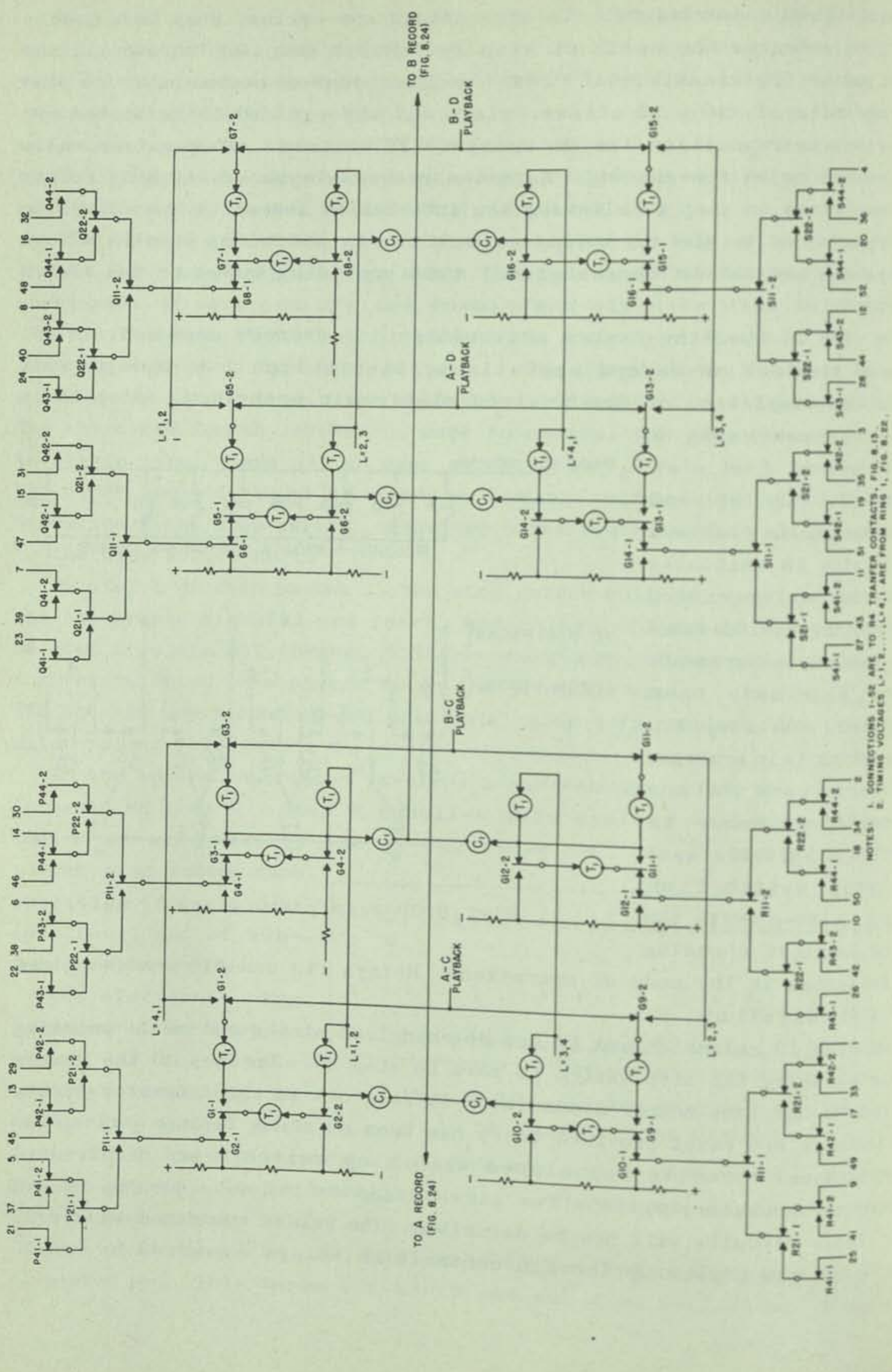


Fig. 8.21—*Relay pyramid for playback or record.*

and *mcv* 2. A number of parallel contacts are provided in most cases, since certain combinations of codes are required by more than one subsequence. For example, the X relays are energized whenever *mcv* 4 is to read through the key contacts to storage relays 1-16 (for A and operation code digits); the Y relays are energized whenever *mcv* 4 is to read through the key contacts to storage relays 17-28 (for B code digits); the Z relays are energized whenever *mcv* 4 is to read through the key contacts to storage relays 29-32 (for C code digits); relays 562 and 563 are energized whenever the codes required are $A15^*$, Op. 1; relays 635, 636, 640, 641, 658, and 659 are energized when, in a double-accuracy addition, the number-time code must be increased by 5, and so on.

Most of the relays controlling the contacts in Fig. 8.16 are energized through step switches 9 and 10, Fig. 8.17 and Fig. 8.18. For example, consider the function subsequence, s_2, associated with wiper 2, step switch 9. On the first cycle, *mcv* 2 picks up relays 526, 527. Through their contacts *mcv* 3 picks up the following relays: Y relays; 562,563; 592. The command delivered to the code storage relays will then be to add the argument selected under the B codes to zero, and to record the sum in r_0.

The next two cycles in the function subsequence supply lines $n+1$ and n, in that order. On step 2 of step switch 9, *mcv* 2 energizes relays 528-531. Through contact 531-2, relay $J1$ (Fig. 8.15) is picked up, connecting wiper 3 instead of wiper 4 on step switch 1 to cam 2. The connections on this level provide for the operations listed in Table 8.3.

TABLE 8.3

Special operations for line $n+1$

Step	Operation	Relays
6	Energize hold line for line number storage unit, and hold for two cycles.	Pick up and hold D26,D28
8	Disconnect line number accumulator from line number storage unit.	Pick up and hold R1 relays (F1-F8)
10	Advance line number accumulator (to $n+1$).	
11	Read from line number accumulator to line number storage relays.	Pick up R3 relays (F22-F28)
14	Advance line number accumulator again (to $n+2$).	

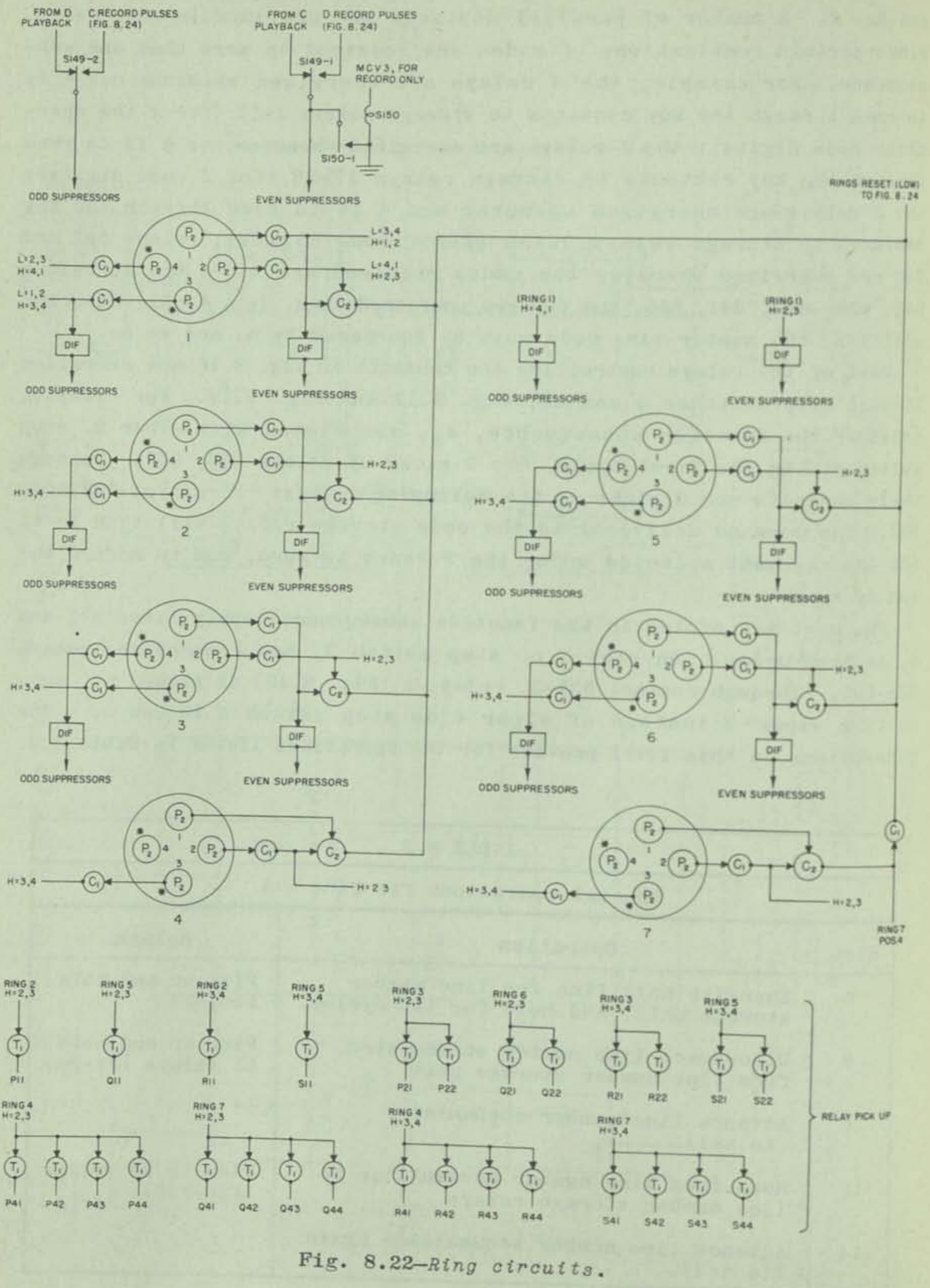


Fig. 8.22—*Ring circuits.*

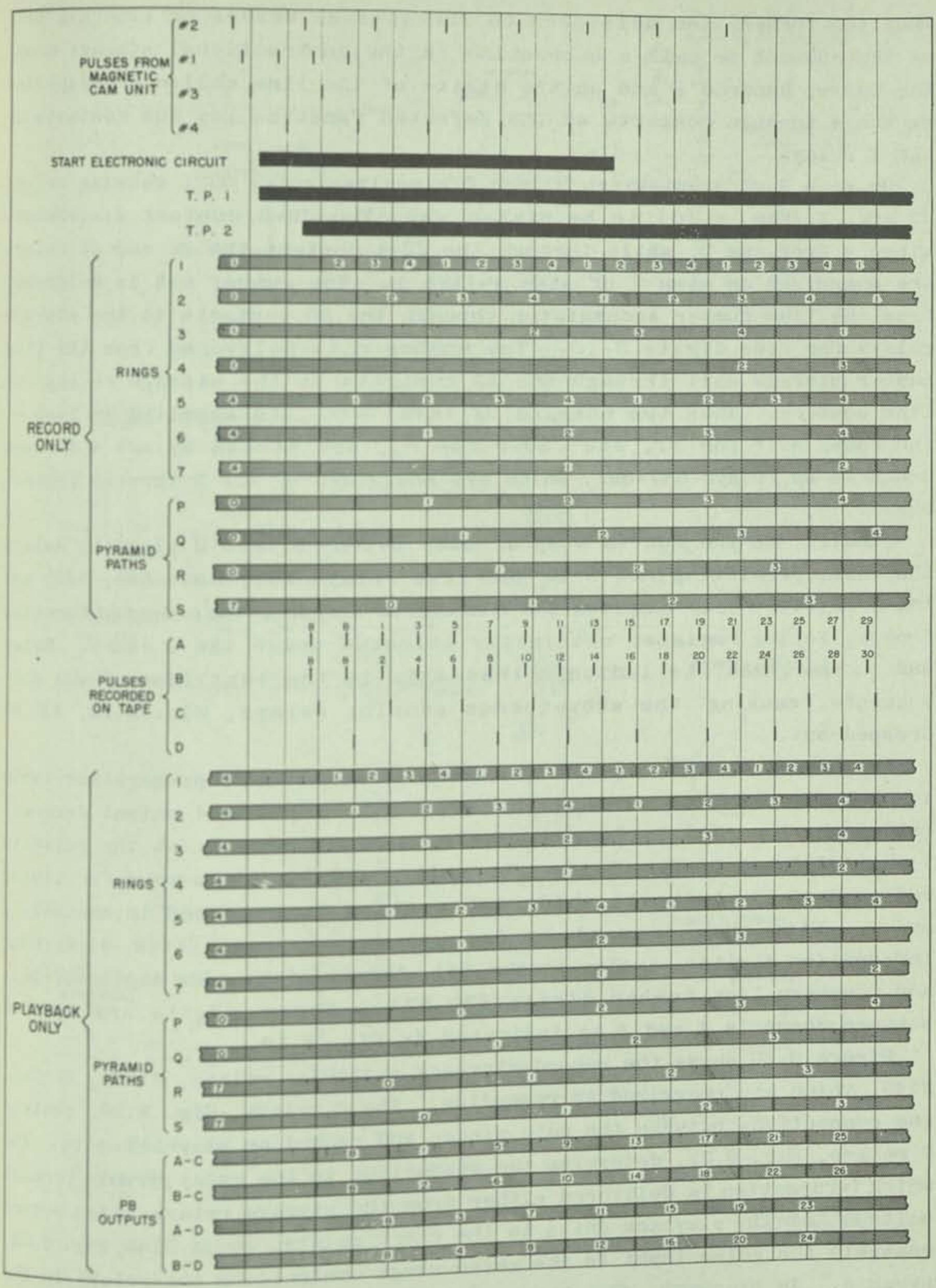


Fig. 8.23—*Timing, record and playback.*

Thus the information delivered to the storage relays on line $n+1$ will be the command to call a subroutine on the instructional storage drum. The ten's, hundred's and unit's digits of the line called are supplied by *mcv* 4 through contacts on the selected function key and contacts on the *X* relays.

On step 3 of step switch 9, *mcv* 2 energizes relay 532, causing relays *J*2 and *J*3 (Fig. 8.15) to be picked up. The *J*2-2 contact disconnects wiper 4 from cam 2, while through the *J*2-1 contact the *R*2 and *R*5 relays are energized on step 7 of step switch 1. The number $n+2$ is delivered from the line number accumulator through the *R*5 contacts to the storage relays for code digits 3-16. The number n is delivered from the line number storage unit through the *R*3 contacts to the storage relays for line numbers. Thus the command, $XT(n+2) = r_9$, is supplied on line n. The codes *B*27* for *XT*, and *C*30-9 for r_9, are stored by *mcv* 4 through contacts on relays 587-591, which are energized by *mcv* 3 through contact 532-1.

Finally, on the fourth step of step switch 9, *mcv* 2 picks up relays 533, 534, permitting *mcv* 3 to energize relays 562, 563; 585, 586; and the *Z* relays. This supplies the command to transfer the computed function from r_1 to the register originally selected under the *C* codes. Relay 682 is energized, to indicate that this is the last line of the subsequence, causing the subsequence sensing relays, Fig. 8.15, to be dropped out.

The magnetic tape used in the instructional tape preparation table is the same as that used in the numerical input and output devices. Pulses are recorded in four channels, *A*, *B*, *C*, and *D*, at the relative times indicated in Fig. 8.19. Channels *C* and *D* are reserved for timing pulses used in playback, while information is recorded in channels *A* and *B*. With each command on tape there are associated sixty-four information digits: twelve blanks (*B*), thirty-eight code digits (1-38), and fourteen line number digits (39-52). These digits are divided between channels *A* and *B* as indicated in Fig. 8.19.

Figure 8.20 shows the record-playback switching relays, *Z*1-*Z*4, *G*1-*G*16, *S*149, which are energized in recording. The *Z* relays, Fig. 8.20, control the connections between the pole pieces and record or playback units. The *G* relays, Fig. 8.21, determine the connection to the relay pyramid through which information is delivered either from the storage relays to the record units or from the playback units to the check relays. Relay *S*149, Fig. 8.22, controls the pulse input to the seven rings controlling the relays in the pyramid. In playback, the first ring is advanced by the *C* and *D* pulses played back from tape. In recording, the first ring is advanced by

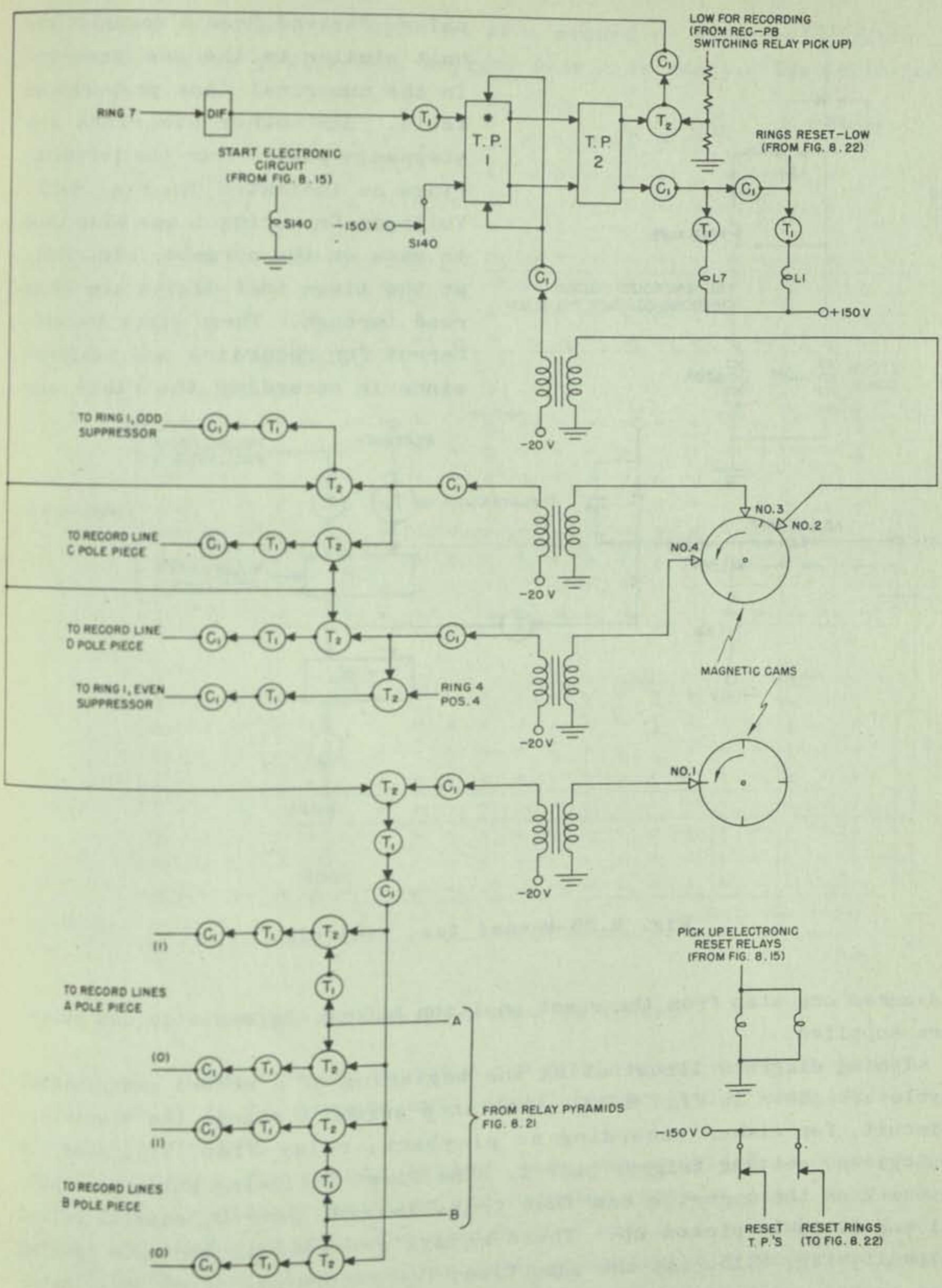


Fig. 8.24—*Main electronic control circuits.*

pulses derived from a magnetic cam unit similar to the one provided in the numerical tape preparation table. The other six rings are stepped by pulses from the preceding rings as indicated in Fig. 8.22. Voltages from ring 1 are also used to gate on the pyramid, Fig. 8.21, at the times that digits are being read through. These times are different for recording and playback since in recording the rings are advanced one step from the reset position before the magnetic cam pulses are supplied.

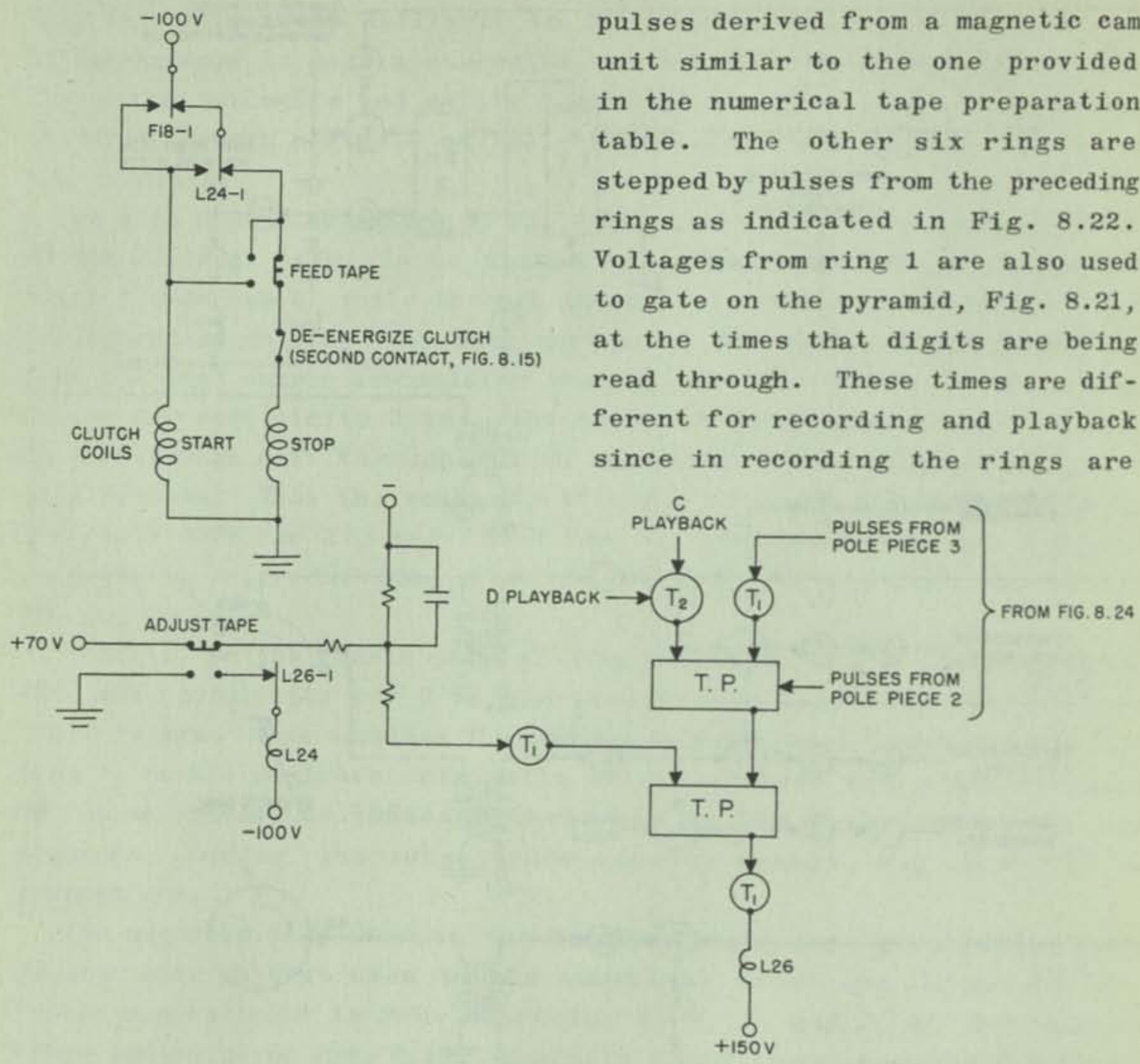


Fig. 8.25—*Manual tape controls.*

Timing diagrams illustrating the beginning of a record and playback cycle are shown in Fig. 8.23. When step switch 1 starts the electronic circuit, for either recording or playback, relay *S*140, Fig. 8.24, is energized, setting trigger pair 1. The first following pulse from pole piece 2 on the magnetic cam then trips trigger pair 2, causing relays *L*1 and *L*7 to be picked up. These relays are used in the main cycling circuit, Fig. 8.15. At the same time, for recording, three pulse gates are opened, admitting pulses from the pole pieces on the magnetic cam to the record circuits. At the end of a record or playback operation, when ring 7 reaches position 4, trigger pair 1 is reset. The following

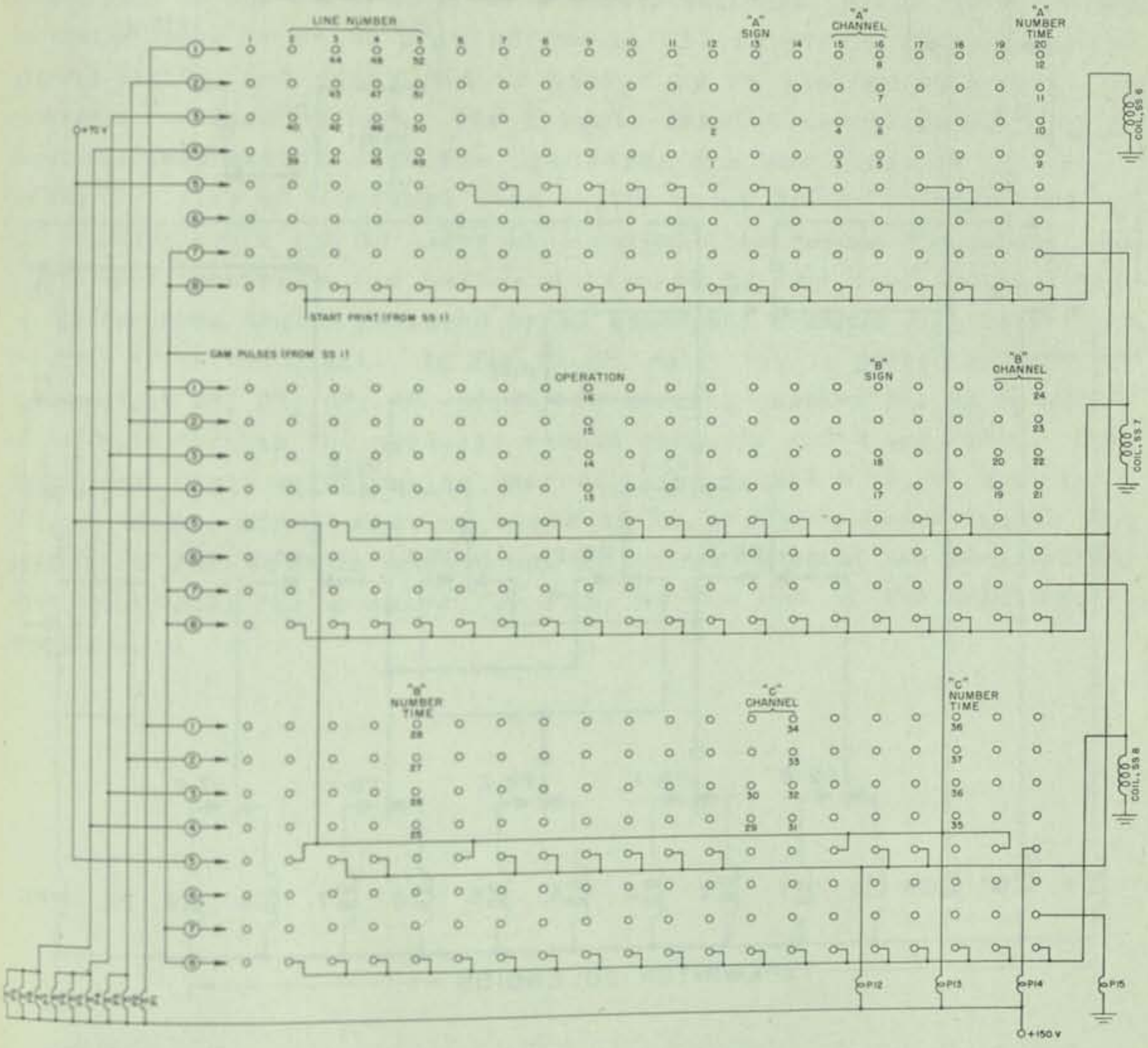


Fig. 8.26—*Printer step switches.*

pulse from pole piece 2 on the magnetic cam resets trigger pair 2, causing relays *L*1 and *L*7 to be dropped out, and thus permitting step switch 1 to resume its cycling.

Manual controls for manipulating tape are provided as shown in Fig. 8.25. To facilitate mounting and removing reels of tape, a switch may be thrown to de-energize the clutch stop coil. A pushbutton, *feed tape*, causes the tape to move continuously under the pole pieces as long as the button is depressed. A second pushbutton, *adjust tape*, causes a previously recorded tape to be positioned to the beginning of the next

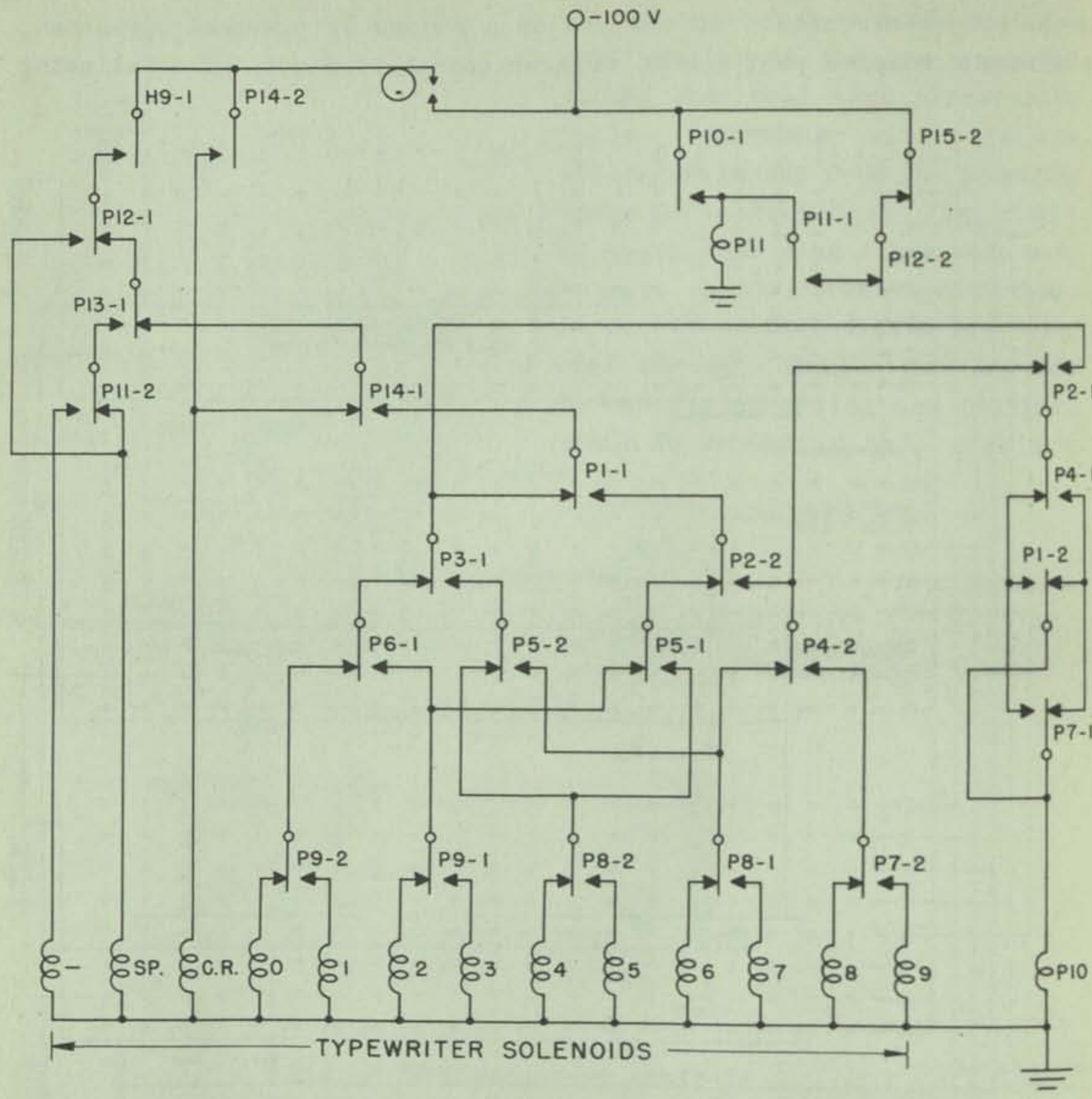


Fig. 8.27—*Pyramid for typewriter solenoids.*

command. This is accomplished by energizing the clutch start coil as long as *C* and *D* pulses are played back from the tape.

This chapter concludes with a description of the printing circuits controlling the electrically operated typewriter associated with the instructional tape preparation table. When the selected mode of operation requires printing, step switch 1 on step 16 starts the first of three step switches, Fig. 8.26. These, through wipers 1-4, scan the storage or check relays for line numbers and codes. Thus on successive steps a decimal digit of the line number or code picks up a configuration of relays *P*1-*P*9. Through the *P* relay pyramid, Fig. 8.27, a corresponding typewriter solenoid is energized, causing the digit to be printed. Through connections on the fifth level of the step switches, relay *P*12 is picked up when a space is to be printed; relay *P*13 is picked up to sense for starred digits; and relay *P*14 is picked up at the end of a cycle to provide a carriage return. The seventh level connections are used to start one step switch when the preceding one has finished its cycle. On the last step of the final step switch relay *P*15 is energized to end the printing cycle and to allow step switch 1 to resume operation. The eighth level controls the pulses delivered to the step switch coils.

For the code digits followed by an asterisk, a minus sign is printed directly after the digit. In Fig. 8.27, relay *P*10 is picked up whenever a code digit 2*, 3*, 4*, 5*, 6*, or 7* occurs, causing *P*11 to be picked up and held through the normally closed contacts *P*12-2 and *P*15-2. Following each digit which may be starred, step switch 6, 7, or 8 picks up *P*13, causing a minus sign or space to be printed, depending on the position of *P*11. *P*11 is dropped out on the next step of the step switch, when either *P*12, for a space, or *P*15, at the end of the printing, is energized.

CHAPTER IX

OPERATION OF THE CALCULATOR

The power supplies of the Mark III Calculator are controlled from the panel shown in Plate IX located at the rear of the left wing of the machine. To turn on the machine the 110 volt AC power must be turned on first. Next the storage drums may be turned on followed by the sequence drum. The three motor generator sets and the filament power should also be turned on at this time.

A minimum of twelve minutes must be allowed for the storage drums to attain full speed, and this is also a convenient length of time to warm up the filaments. When the storage drums are at full speed, the following DC voltages may be turned on: -65, -150, ±100, -20, +55, +70, +150, -120, +250, +300, -250. The order prescribed here is calculated to prevent the application of plate voltages to vacuum tubes before bias voltages are applied. This is, of course, to prevent excessive plate or grid currents which might cause damage to the electronic components of the machine. It is particularly important to turn on the record voltage (-250 V) last to prevent the possibility of recording spurious information on the drums.

Normally none of the voltages requires any adjustment, but meters and adjusting rheostats are provided so that all voltages can be set close to the nominal values. The switch controlling the cam unit located underneath the typewriter table is generally left on; if this switch happens to be off, however, it may be turned on at any time. Voltages from the cam unit are required for the operation of the printers, the main control panel, and the instructional and numerical tape preparation tables.

In turning off the power, the record voltage (-250 V) is removed first. Then the +300, +250, +150 and +70 volt powers may be turned off, followed by the filament voltage and the 110 volt AC power. When 110 volt AC is turned off all the relays controlling the motor generators, drums, and remaining DC voltages are dropped out.

Before a problem can be solved by the machine, the instructional tape and any necessary value tapes must be made. Two auxiliary machines, the instructional tape preparation table and the numerical tape preparation table, are provided for the purpose of making the required tapes. The operation of these units will now be described.

If the instructional tape preparation table power is not on, it should be turned on with the record voltage (-250 V) of the calculator off.

This is to avoid the possibility of transient surges' disturbing information recorded on the drums of the calculator. After the filaments have been lit for about ten minutes, the high voltage switches may be thrown on.

An erased reel of tape is threaded through the rollers of the tape mechanism as shown in Plate X. Next, one of the modes of operation (either *record* or *record and print*) should be selected by depressing the appropriate key at the upper right of the keyboard. The cycling reset button should be depressed in case the cycling reset light is not on.

Before recording the first line of coding, the line number accumulator must be set to the line number on which the coding starts. This is done by entering the required line number in the manual register at the upper left of the keyboard and pressing the button *set accumulator to specified line number*. The correct line number should then appear in the line number lights in 2*,4,2,1 notation.

The position of the tape should be marked and the first line of coding should be set up by depressing the appropriate keys in the main keyboard of the coding machine. The start button at the far right should be pressed to cause the proper codes to be recorded on the tape. When recording is complete the keyboard will be reset and the line number accumulator automatically advanced by the number of lines recorded on tape. The next line may now be recorded in the same way as the first line, except that the tape need not be marked nor the line number accumulator altered.

After ten or twenty lines have been recorded, it is advisable to perform a checking operation to make sure that the coding machine is operating properly. To do this the *disengage clutch* switch should be thrown and the tape rolled back manually to the place initially marked.

Now one of the modes of operation involving playback and check should be selected. The line number accumulator is once more set to the starting line number, the first line of coding is entered in the main keyboard, and the *start* button depressed. In case of a check failure, it should be determined whether the error is due to electrical failure in the coding machine or improper use of the machine. To facilitate tracing sources of error it is advisable to employ the printer in both record and playback operations. It will be assumed here that the coding machine is operating correctly and that any check failures are due to errors in setting up the keyboard. In this case the proper procedure is to make a note of the lines incorrectly recorded. These lines may be rerecorded at the end of the tape (after the line number accumulator is set to the proper line number).

The operator should now proceed to record the remainder of the program. The entire tape must be checked, and at some stage it is desirable that the coding be printed. This may be done either while recording or checking, or, using the *playback and print* mode of operation, after the checking is complete.

Any necessary value tapes should be prepared on the numerical tape preparation table. After the filament switch has been turned on for about ten minutes the high voltages may be turned on by means of the two high voltage toggle switches. The -250 volt power (record voltage) should be turned off at the main control panel while the numerical tape preparation table is turned on to avoid recording any spurious pulses on the drums of the machine.

Next a reel of erased tape is placed on the tape mechanism above the keyboard as shown in Plate X. After the tape is threaded the motor should be turned on.

Before entering the first value in the keyboard, it is necessary that the circuits of the numerical tape preparation table be reset, as indicated by the reset light in the upper right. In case this light is not on, the reset toggle switch should be thrown. When the light comes on, the switch should be turned off again. In addition to resetting the circuits, this switch resets all the keys in the keyboard.

Now the first number is entered in the keyboard by depressing the appropriately numbered key in each column. In case the numbers to be recorded on tape are such that zeros occur repeatedly in a certain column, the toggle switch at the head of that column may be placed in the "zero" position. This is the equivalent of depressing the zero key in the column, except that the toggle switch can only be reset manually. When the number is set up in the keyboard, the *start* button is depressed. At this point the light marked *first step completed* should be lit. If, however, the *column check* light comes on, it means the operator has failed to depress a key (or throw a zero switch) in one or more columns; the error should be rectified and the *start* button pushed once more.

When the *first step completed* light comes on, the keys are automatically reset. The operator must re-enter the number in the keyboard and again depress the *start* button. If the keyboard has been set up in an identical manner both the first and second times, the number will be recorded on tape, the keyboard and all electrical circuits will be reset, and the *reset* light will be turned on. If the second number set up on the keyboard does not agree with the first the *check* light will be turned on. In this case the *reset* toggle switch should be used and the entire process of recording the number should be repeated.

The succeeding numbers making up the value tape should be recorded in the same manner as the first number. Considerable care must be taken in the preparation of value tapes, since it is difficult to make a correction in case a number has been erroneously recorded.

The next step in setting up the calculator for the solution of a problem is to transfer the coding recorded on tape to the sequence drum. The reel of tape taken from the instructional tape preparation table must be rewound and placed in the tape mechanism just to the right of the main control panel. The tape should be manually adjusted to its starting position by means of the mark made when the tape was prepared. The tape mechanism drive motor should be turned on after all manual adjustments are made. Finally the tape circuits should be reset as indicated by the *tape circuits reset* light in the lower right-hand corner of the main panel. If this light is not on, the tape circuits may be reset by pressing the button located directly below the light.

The first line of coding may now be recorded on the drum. The button *tape to drum and lights, next line* should be used. The first line of coding will then be recorded on the drum and displayed in the lights. This line of coding should be checked against the coding from which the instructional tape was prepared to make sure that it is the first line and that the machine is operating properly.

To transfer the balance of the coding to the drum, the toggle switch *tape to drum and lights, continuous* may be thrown. The lines of coding on the tape will then be recorded on the drum at the rate of approximately 120 lines per minute.

When all coding has been transferred to the drum, the tape should be removed from the mechanism, rewound, and stored until the problem has been completed.

Any input number tapes required in the problem must be placed in the appropriate tape read-record mechanisms. The first step is to turn off the motors and then the clutches by means of the appropriate toggle switches. After the value tape has been rewound the reel must be placed on the upper (supply) spindle. The tape is then threaded on the mechanism as illustrated in Plate XXII. An empty reel is placed on the lower (take-up) spindle and the tape is fastened to the hub of the lower reel. The tape is then manually adjusted to the starting position indicated by the mark made when the tape was prepared.

After the tape is positioned the mechanism must be set up for reading. This is done by placing the *read-record* switch in the *read* (lower) position and plugging the six leads from the pole pieces into the lower row of plughubs. Finally the clutches and motors may be turned on.

In the preparation of a value tape it is possible to record a pair of extra numbers at the beginning of the tape to assist in making sure the tape is properly positioned and to prove that the proper tape is on the mechanism. These numbers may be read from the tape to intermediate storage and thence to lights by operating the machine manually from the front panel.

To set up a tape mechanism for recording, a reel of erased tape is placed on the upper spindle. The tape is threaded as before and attached to the hub of an empty reel placed on the lower spindle as before. The *read-record* switch is placed in the *record* position and the six leads from the pole pieces are connected to the upper plughubs. Finally the tape must be marked to indicate the position of the first number recorded.

The operating decimal point of the calculator is set by depressing one of the keys at the lower left of the main control panel. The proper location of decimal point is, of course, specified in the operating instructions for the problem at hand. Before starting the problem certain other procedures, as required by the operating instructions, may have to be carried out. In particular, it may be necessary to introduce certain quantities into the constant register channel. This is done by setting up the required number in the *read-in* keyboard at the upper left of the main panels. The number time of the register to be read into is selected by depressing the appropriate *read-in switch register selection* key, and the read-in is accomplished by pushing the *read-in* button. All numbers read into the calculator by this method should be checked by reading into lights.

After the set-up procedure has been completed, the start line number must be entered in the *start line number* register on the front panel. When a problem is placed on the machine for the first time it is generally necessary to stop the machine at selected places in the computation to make sure that the proper numbers are being computed. The number of the first line at which it is required to stop is entered in the *stop line number* register and the *stop specified line* toggle switch is thrown to the upper position. The machine is then started by depressing the *start specified line* button and will automatically stop after the line entered in the *stop line number* register has been executed.

When the stop has been reached, the contents of various selected registers may be displayed in lights and checked against hand computations prepared for the purpose. In case of error, the source of the error must be traced. This process is facilitated by appropriate use of the four buttons *run next line and stop, run specified line and stop, read last line to lights*, and *read specified line to lights*, in addition to the controls just described.

In case it happens that a certain line of coding must be changed, the number of the line should be entered in the *select line number* register and the proper codes entered in the *coding* keyboard. The coding may be transferred to the drum and lights by pushing the button labeled *transfer line of coding to drum and lights*. The codes entered in the keyboard may now be checked by examining the *coding* lights. The coding in the drum may be checked by entering the line number in the *start line number* register and pressing the button *read specified line to lights* and again examining the *coding* lights.

As soon as it is certain that all coding is correct the machine is, of course, run continuously with the *stop specified line* switch off.

It may be anticipated that during the course of computations various machine stops will occur. When the machine stops, the stop light is lit. In addition another light should come on to indicate the reason for the stop (e.g., *code stop, check stop, adder exceed*, etc.).

In case a stop occurs the number of the last line of coding executed should be determined by pressing the *read last line to lights* button and examining the line number lights. The line on which to start the calculator and the procedure which must be followed before starting should be ascertained by referring to the instructions prepared in coding the problem. In the case of check stops it is usually necessary merely to start the calculator on some specified line. In case of a coded stop it may, for example, be necessary to change the value of certain parameters by means of the read-in register, or perhaps the tape on a read-record mechanism must be changed. Eventually a code stop is reached which signifies the end of the computation. At this point a new problem must be set up on the calculator and the results of the completed problem must be printed out.

The printers, like the tape preparation tables, are independent of the calculator proper and have individual power controls. Normally all the printers are left on; in case it is necessary to turn a printer on or off, however, the record voltage (-250 V) should be first turned off at the main power supply control panel. To turn on a printer the filament switch located at the far left of the printer control panel (Plate XXVII) should be turned on first. After the filaments have been heated for about ten minutes the switches controlling the high voltages should be turned on.

The controls determining the typography must be set up in accordance with the specifications regarding the required format of the printed page. A detailed discussion of the function of these controls has already been presented in Chapter VI so that only a brief recapitulation need be given here.

The toggle switches *argument control counter* and *skip step* determine the succession of argument-style prints and function-style prints. The *drop zero* switches control the suppression of insignificant zeros; as a rule the switch in the column immediately above the decimal point and those below are off, while switches in the higher columns are turned on. The *end-of-page space* switch may be thrown to suppress the thirteen carriage returns generally given after each page. The *line grouping* switch determines whether the printed page shall consist of fifteen groups of three lines each or ten groups of five lines each. The *D.P. cut-off* switch determines whether or not the decimal point be printed in an argument-style print; this switch is, of course, effective only in case a decimal point is plugged. The *repeat print* switch is mainly for testing purposes and should be left off.

The plugging above the printer tape mechanism determines which of the sixteen digits recorded on tape are printed; furthermore, it is to be noted that only those digits bracketed by the *A.C.* 1 and *A.C.* 2 plugs will be printed in an argument-style print. In addition, plugging is used to insert the decimal point and any spaces required in their proper positions.

The total number of characters (including spaces) in a typical line of print must be counted to fix the setting of the left- and right-hand margin stops of the typewriter. The setting of the right-hand stop is important as the carriage return is supplied only when the carriage is against this stop. At the time the stops are set it is convenient to check the position of the paper in the typewriter.

The tape from which it is desired to print should be placed on the printer mechanism as shown in Plate XXIV. The switches on the mechanism panel should be set to print from *A* with check. Before starting the printer, the *ring reset* switch should be turned on. This should cause the *ring home* light to be lit. While the *ring reset* switch is still on, the *reset step switches* switch should be turned on and left on until the lights numbered 1, 2, 3, 4, 5, 6, and 7 above the switch are all on. Then the two reset switches may be turned off.

To start the printer the *start* button should be depressed; it should be noted that the *alarm bell cut-off* switch must be down for the *start* button to take effect. The switch to the right of the *start* button determines whether the printer stops after printing the first number or prints continuously. This switch may be thrown from *continuous* to *single* at any time in order to stop the printer immediately. To stop the printer at the end of a page, the *stop at end of page* switch may be used. The final pages printed by the machine should be marked with the date and the name of the problem.

In conclusion it should be remarked that all occurrences concerned with the operation of the machine are carefully recorded in a logbook. This is not only to allow analysis of the time spent in running various problems, but is also of great value to the engineering staff responsible for the maintenance of the machine. Furthermore, the mathematician responsible for the solution of a certain problem can check to make sure that all necessary steps in the operating procedure have been performed. Besides the logbook special sheets are provided for recording pertinent facts connected with the output tapes. These sheets serve as a guide when the information recorded on the tapes is printed. All marks made on tape should be numbered and dated in correspondence with the entries in the log sheets; the use of dates avoids confusion when the reel of tape is erased and used again.

CHAPTER X

PROBLEM PREPARATION AND SOLUTION OF TYPICAL EXAMPLES

A description of the meanings of the individual codes which control the operation of the machine was given in connection with the sequencing unit in Chapter VII. The object of the present chapter is to give a description of the way in which codes are used together in order to cause the calculator to solve specific problems. To this end the coding for four sample problems will be presented.

Before considering these examples, however, it is advisable to remark upon some of the salient features of the Mark III Calculator. In the first place, the speed of the machine is very great; in the space of one minute up to fifteen thousand lines of coding may be acted upon by the machine. The first consequence of this is that only problems of a considerable magnitude are suitable for solution on the Mark III Calculator. In addition all unnecessary machine stops should be avoided, since the time required for the operator to determine the cause of the stop and to carry out the procedure necessary to start the machine again is obviously very large when measured in terms of machine cycles.

The sequencing unit of the calculator permits considerable flexibility in the organization of the coding of problems. In particular, a conditional call to rerun a calculation may be coded in place of a check, as was shown in the routines to compute the elementary functions presented in Chapter V. Examination of these routines shows that numerous calls and conditional calls are made to reduce the number of lines of coding required; and while the routines used to compute the elementary functions are quite short, they serve to suggest the great range of possibilities for coding problems requiring many hundreds of lines of coding.

Finally, the number of registers comprising the internal storage system of the calculator is relatively large, in keeping with the high speed and flexibility of the machine. It should rarely be necessary to resort to external storage except for the numbers representing the final computed results which must be printed.

Before the coding of a problem for machine solution can be commenced, it is necessary to reduce the problem to a form involving only those operations which the machine can perform. Usually several numerical methods are available for the solution of a given problem; the selection of a particular method is based upon the accuracy required in the results, the time required for the computation, and the ease with which the coding can be carried out. The various alternatives must be carefully weighed by the mathematician responsible for the coding, and in the final analysis the decisions taken depend to a large extent upon his individual judgment and experience.

After the numerical method has been selected the problem may be broken down into a number of appropriate subroutines and the coding started. Most of the difficulties in coding for any machine arise from the individual peculiarities of the problem to be solved and from the form in which the coding must be written. The instructional tape preparation table of the present calculator has been designed to make the form taken by the coding resemble closely the form in which algebraic expressions are generally written down, thus reducing greatly the difficulties in the mere mechanics of coding.

Example 1. It is required to tabulate a group of seventh degree polynomials of the form

$$y = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + a_6x^6 + a_7x^7, \qquad (1)$$

in the range $0 \leq x \leq x_{max}$ at intervals of $\Delta x = h$.

For the purpose of computation it is convenient to rewrite Eq. (1) in the form

$$y = a_0 + x(a_1 + x(a_2 + x(a_3 + x(a_4 + x(a_5 + x(a_6 + xa_7)))))).$$

It is seen that seven multiplications and seven additions are required for the calculation.

While the polynomial could very well be checked by the calculation of high order differences, it is apparently faster to evaluate the sixth degree polynomial expression for

$$\Delta y = y(x + h) - y(x).$$

Let

$$y(x + h) - y(x) = \Delta y = b_0 + b_1x + b_2x^2 + b_3x^3 + b_4x^4 + b_5x^5 + b_6x^6;$$

then the values of the coefficients b_i are found to be

$$\left.\begin{aligned}
b_0 &= a_7h^7 + a_6h^6 + a_5h^5 + a_4h^4 + a_3h^3 + a_2h^2 + a_1h \\
b_1 &= 7a_7h^6 + 6a_6h^5 + 5a_5h^4 + 4a_4h^3 + 3a_3h^2 + 2a_2h \\
b_2 &= 21a_7h^5 + 15a_6h^4 + 10a_5h^3 + 6a_4h^2 + 3a_3h \\
b_3 &= 35a_7h^4 + 20a_6h^3 + 10a_5h^2 + 4a_4h \\
b_4 &= 35a_7h^3 + 15a_6h^2 + 5a_5h \\
b_5 &= 21a_7h^2 + 6a_6h \\
b_6 &= 7a_7h
\end{aligned}\right\} \qquad (2)$$

Since h is a fixed constant, the values of b_i can be computed initially from the coefficients a_i. Each new value of y can therefore be computed in two different ways and the results checked.

The coding for the problem is given in Table 10.1 and may be considered to be divided into two parts. The first part may be called the starting routine, and is used only at the start of the tabulation of each new polynomial. The second part is the tabulation routine. This routine is executed to compute each new value of the dependent variable.

The coefficients of the polynomials are introduced by means of a value tape. Each group of coefficients is preceded by a serial number whose sign is positive. The serial number is stored in a_9 for reference in case of an arbitrary point rerun. Each group of coefficients is followed by the negative sum of the coefficients for checking purposes. At the end of the value tape a negative number is recorded to control the stop register.

After the coefficients a_i have been transferred from the value tape on mechanism 1 to the storage registers a_i, the negative sum of the coefficients is transferred to a_8 and a check is performed on line 0082. The next lines of coding to line 0198 are for the purpose of computing the values of the coefficients b_i according to Eqs. (2).

It is found by summing Eqs. (2) that

$$\sum_0^6 b_i = a_1h + a_2(h^2+2h) + a_3(h^3+3h^2+3h) + a_4(b^4+4b^3+6h^2+4h) + \cdots$$

whence

$$\sum_0^6 b_i = \sum_0^7 a_i(h+1)^i - \sum_0^7 a_i.$$

Lines 0199 to 0227 are used to compute $\sum_0^7 a_i(h+1)^i$, and on line 0238 a conditional call is made to rerun the calculation of the coefficients b_i in case

$$\left| \sum_0^6 b_i - \sum_0^7 a_i(h+1)^i + \sum_0^7 a_i \right|$$

is greater than a suitable tolerance placed in switch 3. Before the call is made the rollback control register, d_0, is incremented by one in the first machine column. This register was initially set up on line 0075 to contain -1 in the first machine column, and therefore, in general, contains the number of reruns performed. On line 0084, the quantity in d_0 is checked to be less than some number placed in switch 5. The purpose of this check is simply to control the number of automatic rollbacks performed by the machine before an indication of failure is given to the operator. The setting of switch 5 should be at the discretion of the operator of the machine.

As an extra precaution, a check is performed on the difference used to control the conditional call register in case the call to the rerun line does not succeed. After this check is passed the first value of the argument, $x = 0$, is placed in x_0 and x_2. The register x_2 will in future contain twice the number in x_0 for checking purposes. Evidently the first value of the function, $y(0)$, is a_0; the quantity a_0 is therefore transferred to y_0 and the argument and function are recorded on mechanism 2.

On lines 0268 and 0269 the quantities h and $2h$, from switches 1 and 2, are added to x_0 and x_2 and the sums are placed in registers x_1 and x_3 respectively. On line 0273 a conditional call is made to rerun the computation of x_1 and x_3 in case x_3 is not twice x_1. The register d_1 is used as the control register for this rerun, being compared with switch 5.

The coding on lines 0281 through 0308 represents the evaluation of the polynomial for the argument x_1. It will be recalled that the first value of the polynomial, $y(0)$, was stored in register y_0. In this instance, however, $y(x_1)$ is stored in y_1. The coding on lines 0309 through 0332 represents the calculation of Δy. For this calculation the previous value of the argument, still stored in x_0, is used, and the result is added to the previous value of the function y_0. An automatic rerun is performed in case $|y_0 + \Delta y - y_1|$ is greater than the tolerance placed in switch 3, and a check is also performed on this difference in case the rerun is not performed. When the check is passed the argument and function are recorded on mechanism 2.

It should be noted that the lines of coding 0342-0437 are the same as lines 0246-0341 except for the fact that the rôles of registers x_0, x_2, y_0, y_2 and registers x_1, x_3, y_1, y_3 are interchanged. The use of a two-section routine of this kind facilitates the rollback since the numbers in x_0, x_2, y_0, and y_2, which are needed to compute x_1, x_3, y_1, and y_3 in the first section, are not changed in the first section, and on the other hand the numbers in x_1, x_3, y_1, and y_3, needed to compute x_0, x_2, y_0, and y_2 in the second section, are not altered in the second section.

The highest value of x for which the value of $y(x)$ is required is stored in switch 6. On lines 0254 and 0350 the argument is compared with switch 6 and on lines 0264 and 0360 a conditional call is made to line

0001. When $x = x_{max}$ this call succeeds and a new set of coefficients a_i is transferred to fast storage from the value tape on mechanism 1 to start a new item of the tabulation. When the last item is completed, the serial number read from tape to a_9 has a minus sign attached and the code stop light comes on, signifying the end of the problem.

In this example all intermediate results obtained in the course of the computation have been read into fast storage registers. This is by no means necessary, but it should be emphasized that the liberal use of registers to store intermediate results cannot fail to simplify the problem of tracing sources of error.

The operating instructions consist of the starting instructions, switch settings and rerun instructions. In addition a description of any value tapes used should be given. In this example the numerical values of the switch settings are not given since the coding is general in form. Together with the description of the value tape already presented, the operating instructions for this problem are as follows:

Starting Instructions

1. Set decimal point switch to 13/12.
2. Place value tape on mechanism 1. Set up mechanism 1 for reading. Set up mechanism 2 for recording.
3. Start line 0001.

Switch Settings

SW 0 = 1 in machine column 1
SW 1 = h
SW 2 = $2h$
SW 3 = Tolerance for b_i check
SW 4 = Tolerance for difference check
SW 5 = Rollback control in machine column 1
SW 6 = x_{max}

Rerun Instructions

Line	Instructions
0007	*FRO* check. Back up value tape and start line 0001.
0017	" " " " " " " " "
0024	" " " " " " " " "
0031	" " " " " " " " "
0038	" " " " " " " " "
0045	" " " " " " " " "
0052	" " " " " " " " "
0059	" " " " " " " " "

Rerun Instructions (continued)

Line	Instructions
0066	*FRO* check. Back up value tape and start line 0001.
0081	" " " " " " " " " "
0082	Identity check on sum of coefficients from tape. Back up value tape and start line 0001. If there is reason to believe that the numbers on the tape are incorrect, start next item by starting on line 0001 without backing up value tape.
0086	Rollback control check. Called from line 0238. Number of reruns in d_0. Start next line.
0242	Computation check. Start line 0087.
0252	Record check. Start line 0246.
0262	Record check. Start line 0256.
0267	Rollback control check. Called from line 0273. Number of reruns in d_1. Start next line.
0277	Identity check. Start line 0268.
0280	Rollback control check. Called from line 0337. Number of reruns in d_2. Start next line.
0341	Computation check. Start line 0268.
0348	Record check. Start line 0342.
0358	Record check. Start line 0352.
0363	Rollback control check. Called from line 0369. Number of reruns in d_1. Start next line.
0373	Identity check. Start line 0364.
0376	Rollback control check. Called from line 0433. Number of reruns in d_2. Start next line.
0437	Computation check. Start line 0364.

Arbitrary Point Rerun

Back up value tape. Start line 0001.

Code Stop

Line 0010. End of problem.

TABLE 10.1

Expression	Line	*A*	OP	*B*	*C*
Start line	0001				Feed 1
a_9 = Item number	0002			*FRO*	a_9
End of problem	0008			a_9	Stop
	0011				Feed 1
	0012			*FRO*	a_0
	0018				Feed 1
	0019			*FRO*	a_1
	0025				Feed 1
	0026			*FRO*	a_2
	0032				Feed 1
	0033			*FRO*	a_3
	0039				Feed 1
	0040			*FRO*	a_4
	0046				Feed 1
	0047			*FRO*	a_5
	0053				Feed 1
	0054			*FRO*	a_6
	0060				Feed 1
	0061			*FRO*	a_7
	0067				Feed 1
	0068	a_0	+	a_1	e_0
	0069	σ	+	a_2	e_1
	0070	σ	+	a_3	e_2
	0071	σ	+	a_4	e_3
	0072	σ	+	a_5	e_4
	0073	σ	+	a_6	e_5
$e_6 = \Sigma a_i$	0074	σ	+	a_7	e_6
d_0 = Rollback control	0075			-SW 0	d_0
$a_8 = -\Sigma a_i$	0076			*FRO*	a_8
Identity check	0082	$-e_6$	6	a_8	
	0083				
Rollback control check	0084	SW 5	-	d_0	Check
	0087	a_7	×	SW 1	e_0
	0090	σ	+	a_6	e_1
	0091	σ	×	SW 1	e_2
	0094	σ	+	a_5	e_3
	0095	σ	×	SW 1	e_4
	0098	σ	+	a_4	e_5
	0099	σ	×	SW 1	e_6
	0102	σ	+	a_3	e_7
	0103	σ	×	SW 1	e_8
	0106	σ	+	a_2	e_9
	0107	σ	×	SW 1	f_0
	0110	σ	+	a_1	f_1
$b_0 = ha_1+h^2a_2+h^3a_3+h^4a_4+h^5a_5+h^6a_6+h^7a_7$	0111	σ	×	SW 1	b_0

TABLE 10.1 (continued)

Expression	Line	*A*	OP	*B*	*C*
	0114	a_3	+	a_3	c_0
$f_2 = 3a_3$	0115	σ	+	a_3	f_2
	0116	*XT*0004	$+\vert$	$\vert$	c_1
$f_3 = 4a_4$	0117	σ	×	a_4	f_3
	0120	σ	+	a_4	c_3
$f_4 = 6a_4$	0121	σ	+	a_4	f_4
	0122	*XT*0007	$+\vert$	$\vert$	c_4
	0123	σ	×	a_7	h_0
	0126	a_6	×	SW 1	h_1
	0129	*XT*0005	$+\vert$	$\vert$	c_5
$f_5 = 5a_5$	0130	σ	×	a_5	f_5
$b_6 = 7a_7h$	0133	h_0	×	SW 1	b_6
$f_6 = 7a_7h^2$	0136	σ	×	SW 1	f_6
$f_7 = 10a_5$	0139	f_5	+	f_5	f_7
	0140	f_6	+	f_6	h_2
$f_8 = 21a_7h^2$	0141	σ	+	f_6	f_8
$f_9 = 35a_7h^2$	0142	σ	+	h_2	f_9
	0143	*XT*0006	$+\vert$	$\vert$	c_6
$g_0 = 6a_6h$	0144	σ	×	h_1	g_0
	0147	*XT*0015	$+\vert$	$\vert$	c_7
$g_1 = 15a_6h$	0148	σ	×	h_1	g_1
	0151	σ	+	g_0	g_2
$g_3 = 20a_6h$	0152	σ	-	h_1	g_3
	0153	f_6	+	g_0	c_8
	0154	σ	+	f_5	c_9
	0155	σ	×	SW 1	h_3
	0158	σ	+	f_3	h_4
	0159	σ	×	SW 1	h_5
	0162	σ	+	f_2	h_6
	0163	σ	×	SW 1	h_7
	0166	σ	+	a_2	h_8
	0167	σ	+	a_2	h_9
$b_1 = 2a_2h+3a_3h^2+4a_4h^3+5a_5h^4+6a_6h^5+7a_7h^6$	0168	σ	×	SW 1	b_1
	0171	f_8	+	g_1	p_0
	0172	σ	+	f_7	p_1
	0173	σ	×	SW 1	p_2
	0176	σ	+	f_4	p_3
	0177	σ	×	SW 1	p_4
	0180	σ	+	f_2	p_5
$b_2 = 3a_3h+6a_4h^2+10a_5h^3+15a_6h^4+21a_7h^5$	0181	σ	×	SW 1	b_2
	0184	f_9	+	g_3	p_6
	0185	σ	+	f_7	p_7
	0186	σ	×	SW 1	p_8
	0189	σ	+	f_3	p_9
$b_3 = 4a_4h+10a_5h^2+20a_6h^3+35a_7h^4$	0190	σ	×	SW 1	b_3
	0193	f_9	+	g_1	g_0

TABLE 10.1 (continued)

Expression	Line	A	OP	B	C
	0194	σ	+	f_5	g_1
$b_4 = 5a_5h+15a_6h^2+35a_7h^3$	0195	σ	×	SW 1	b_4
$b_5 = 6a_6h+21a_7h^2$	0198	f_8	+	g_0	b_5
$g_2 = 1+h$	0199	SW 1	+	10^{12}	g_2
	0200	σ	×	a_7	g_3
	0203	σ	+	a_6	g_4
	0204	σ	×	g_2	g_5
	0207	σ	+	a_5	g_6
	0208	σ	×	g_2	g_7
	0211	σ	+	a_4	g_8
	0212	σ	×	g_2	g_9
	0215	σ	+	a_3	r_0
	0216	σ	×	g_2	r_1
	0219	σ	+	a_2	r_2
	0220	σ	×	g_2	r_3
	0223	σ	+	a_1	r_4
	0224	σ	×	g_2	r_5
$r_6 = \Sigma a_i(1+h)^i$	0227	σ	+	a_0	r_6
$r_7 = \Sigma a_i(1+h)^i - \Sigma a_i$	0228	σ	+	a_8	r_7
	0229	b_0	+	b_1	r_8
	0230	σ	+	b_2	r_9
	0231	σ	+	b_3	s_0
	0232	σ	+	b_4	s_1
	0233	σ	+	b_5	s_2
$s_3 = \Sigma b_i$	0234	σ	+	b_6	s_3
	0235	σ	-	r_7	s_4
	0236	$+\|\sigma\|$	-	SW 3	Cond. call
Increment rollback control	0237	d_0	+	SW 0	d_0
Automatic rollback	0238	*CC*0084			
	0239	r_7	-	s_3	s_5
Computation check	0240	$-\|\sigma\|$	+	SW 3	Check
x_0 = Argument	0243	*XT*0000		$+\|\ \|$	x_0
x_2 = Twice argument	0244	σ			x_2
$y_0 = y(0)$	0245			a_0	y_0
Record argument	0246			x_0	Rec. 2
	0254	x_0	-	SW 6	Cond. call
d_1 = Rollback control	0255			-SW 0	d_1
Record function	0256			y_0	Rec. 2
Start new item	0264	*CC*0001			
Rollback control check	0265	SW 5	-	d_1	Check
x_1 = Argument	0268	x_0	+	SW 1	x_1
x_3 = Twice argument	0269	x_2	+	SW 2	x_3
	0270	x_1	+	x_1	e_0
	0271	$+\|\sigma\|$	-	SW 0	Cond. call
Increment rollback control	0272	SW 0	+	d_1	d_1
Automatic rollback	0273	*CC*0265			

TABLE 10.1 (continued)

Expression	Line	A	OP	B	C
d_2 = Rollback control	0274			-SW 0	d_2
	0275	x_1	+	x_1	e_1
Argument check	0276	σ	6	x_3	
	0277				
Rollback control check	0278	SW 5	-	d_2	Check
	0281	a_7	×	x_1	e_2
	0284	σ	+	a_6	e_3
	0285	σ	×	x_1	e_4
	0288	σ	+	a_5	e_5
	0289	σ	×	x_1	e_6
	0292	σ	+	a_4	e_7
	0293	σ	×	x_1	e_8
	0296	σ	+	a_3	e_9
	0297	σ	×	x_1	f_0
	0300	σ	+	a_2	f_1
	0301	σ	×	x_1	f_2
	0304	σ	+	a_1	f_3
	0305	σ	×	x_1	f_4
$y_1 = y(x) = \sum a_i x_1^i$	0308	σ	+	a_0	y_1
	0309	b_6	×	x_0	f_5
	0312	σ	+	b_5	f_6
	0313	σ	×	x_0	f_7
	0316	σ	+	b_4	f_8
	0317	σ	×	x_0	f_9
	0320	σ	+	b_3	g_0
	0321	σ	×	x_0	g_1
	0324	σ	+	b_2	g_2
	0325	σ	×	x_0	g_3
	0328	σ	+	b_1	g_4
	0329	σ	×	x_0	g_5
	0332	σ	+	b_0	g_6
$y_3 = \Delta y(x-h) = \sum b_i x_0^i$	0333	σ	+	y_0	y_3
	0334	σ	-	y_1	g_7
	0335	$+\lvert\sigma\rvert$	-	SW 4	Cond. call
Increment rollback control	0336	d_2	+	SW 0	d_2
Automatic rollback	0337	CC0278			
	0338	y_1	-	y_3	g_8
Computation check	0339	$-\lvert\sigma\rvert$	+	SW 4	Check
Record argument	0342			x_1	Rec. 2
	0350	x_1	-	SW 6	Cond. call
d_1 = Rollback control	0351			-SW 0	d_1
Record function	0352			y_1	Rec. 2
Start new item	0360	CC0001			
Rollback control check	0361	SW 5	-	d_1	Check
x_0 = Argument	0364	x_1	+	SW 1	x_0
x_2 = Twice argument	0365	x_3	+	SW 2	x_2

TABLE 10.1 (continued)

Expression	Line	A	OP	B	C
	0366	x_0	+	x_0	e_0
	0367	$+\|\sigma\|$	-	SW 0	Cond. call
Increment rollback control	0368	SW 0	+	d_1	d_1
Automatic rollback	0369	CC0361			
d_2 = Rollback control	0370			-SW 0	d_2
	0371	x_0	+	x_0	e_1
Argument check	0372	σ	6	x_2	
	0373				
Rollback control check	0374	SW 5	-	d_2	Check
	0377	a_7	×	x_0	e_2
	0380	σ	+	a_6	e_3
	0381	σ	×	x_0	e_4
	0384	σ	+	a_5	e_5
	0385	σ	×	x_0	e_6
	0388	σ	+	a_4	e_7
	0389	σ	×	x_0	e_8
	0392	σ	+	a_3	e_9
	0393	σ	×	x_0	f_0
	0396	σ	+	a_2	f_1
	0397	σ	×	x_0	f_2
	0400	σ	+	a_1	f_3
	0401	σ	×	x_0	f_4
$y_0 = y(x) = \Sigma a_i x_0^i$	0404	σ	+	a_0	y_0
	0405	b_6	×	x_1	f_5
	0408	σ	+	b_5	f_6
	0409	σ	×	x_1	f_7
	0412	σ	+	b_4	f_8
	0413	σ	×	x_1	f_9
	0416	σ	+	b_3	g_0
	0417	σ	×	x_1	g_1
	0420	σ	+	b_2	g_2
	0421	σ	×	x_1	g_3
	0424	σ	+	b_1	g_4
	0425	σ	×	x_1	g_5
	0428	σ	+	b_0	g_6
$y_2 = \Delta y(x-h) = \Sigma b_i x_1^i$	0429	σ	+	y_1	y_2
	0430	σ	-	y_0	g_7
	0431	$+\|\sigma\|$	-	SW 4	Cond. call
Increment rollback control	0432	d_2	+	SW 0	d_2
Automatic rollback	0433	CC0374			
	0434	y_0	-	y_2	g_8
Computation check	0435	$-\|\sigma\|$	+	SW 4	Check
	0438	C0246			

Example 2. In this example the coding for a routine to compute and check the expression

$$a^b = c \tag{3}$$

where a, b, and c are complex quantities will be given. At the start of the routine it will be supposed that the real and imaginary parts of a and b are standing in registers a_0, a_1, b_0, b_1, respectively, and it is required to place the real and imaginary parts of the result in c_0 and c_1 respectively.

In computing the complex exponential the built-in functional routines will, of course, be employed. It is therefore necessary to write out the real and imaginary parts of Eq. (3) in terms of logarithms and exponentials to the base ten. In the first place

$$a = 10^{\log_{10}(a_0+ia_1)}$$

where

$$\begin{aligned}\log_{10}(a_0+ia_1) &= \log_{10}\left\{\sqrt{a_0^2+a_1^2}\; e^{i\tan^{-1}(a_1/a_0)}\right\} \\ &= \tfrac{1}{2}\log_{10}(a_0^2+a_1^2) + i\tan^{-1}(a_1/a_0)\log_{10}e. \qquad (4)\end{aligned}$$

Here the value of $\tan^{-1}(a_1/a_0)$ lying between $-\pi$ and π, i.e., the principal value, is to be used. It follows from Eq. (4) that

$$\begin{aligned}a^b &= 10^{(b_0+ib_1)\log_{10}(a_0+ia_1)} \\ &= \rho(\cos\phi + i\sin\phi),\end{aligned}$$

where

$$\rho = 10^{\frac{1}{2}b_0\log_{10}(a_0^2+a_1^2) - b_1\tan^{-1}(a_1/a_0)\log_{10}e},$$

and

$$\phi = \tfrac{1}{2}b_1\log_e 10\log_{10}(a_0^2+a_1^2) + b_0\tan^{-1}(a_1/a_0).$$

Finally, therefore,

$$c_0 = \rho\cos\phi, \qquad c_1 = \rho\sin\phi.$$

The results c_0 and c_1 may be checked by means of the relation

$$a = c^{1/b}.$$

Since this is another complex exponential, the same coding can be used for the check as was used for the computation of c. In the case of the check calculation, however, the principal value of $\tan^{-1}(c_1/c_0)$ cannot be used since this is not necessarily equal to ϕ. To the principal value of $\tan^{-1}(c_1/c_0)$ must be added $2n\pi$, where n is a positive or negative integer, or zero, such that

$$-\pi < \phi - 2n\pi \leq \pi.$$

Evidently the calculation of the quantity $2n\pi$ must be made during the original computation of c.

The coding for the routine is given in Table 10.3. On line 0501 the rollback control counter, h_0, is set up in the manner discussed in Example 1. On line 0502 a negative number is read into h_7, which will be used to control the conditional call register in connection with the check routine. On line 0506 zero is placed in h_1, which will later be used to store the quantity $2n\pi$. On lines 0507-0510 the quantities a_0, a_1, b_0, and b_1 are transferred to the working registers x_0, x_1, x_2, and x_3.

It is necessary to comment on the calculation of $\tan^{-1}(x_1/x_0)$. The angle computed by the functional routine lies in either the first or fourth quadrant, whereas the inverse tangent required in the present calculation should lie in the third quadrant in case x_0 and x_1 are both

TABLE 10.2

Sign of x_0	Sign of x_1	Quadrant of angle computed by functional routine	Quadrant of required angle	Change required in computed angle
+	+	I	I	none
-	+	IV	II	add π
-	-	I	III	subtract π
+	-	IV	IV	none

negative, and in the second quadrant in case x_0 is negative and x_1 is positive. The situation may be summarized as shown in Table 10.2. Now if $s(x)$ be defined to take on the value ± 1 according to the sign of x, then

$$\theta = \theta' + s(x_1)\cdot\pi/2 - s(x_0)s(x_1)\cdot\pi/2,$$

where θ is the required value of $\tan^{-1}(x_1/x_0)$ and θ' is the angle computed by the functional routine. The operation of multiplying $s(x)$ by a quantity is, of course, the sign-choice operation, represented by the symbol $\odot$ on the Mark III instructional tape preparation table. The choices and additions required by Eq. (4) are made in lines 0531-0533.

On line 0569 a conditional call to the check routine is made. The first time this line is executed, h_7 contains a negative number so the call fails. An automatic rollback is made on line 0577 in case $|x_5| > \pi$; this is particularly designed to guard against the possibility that h_1 contains some multiple of 2π, which might happen if line 0506 were not executed properly. The quantity $2n\pi$, which must later be added to $\tan^{-1}(c_1/c_0)$ as previously explained, is computed and placed in h_1 on lines 0578-0594, and on the following lines $\rho \cos \phi$ and $\rho \sin \phi$ are transferred from z_0 and z_1 to c_0 and c_1.

Next the reciprocal of $b_0 + ib_1$ is computed and placed in registers y_0 and y_1, c_0 and c_1 are placed in registers x_0 and x_1, and a positive quantity is transferred to the control register h_7. On line 0619 a call is made to line 0511, the first line of the routine to compute

$$(x_0 + ix_1)^{y_0+iy_1} = z_0 + iz_1.$$

This time the conditional call to line 0620 succeeds; $z_0 + iz_1$ is compared with $a_0 + ia_1$ and an automatic rollback takes place in case these quantities do not agree.

In addition to the check on line 0505 which may be rerun by starting on the next line, check stops are coded in the routines for performing division and computing the elementary functions. The checks on these routines should be rerun by starting on the first line of the routine on which the failure occurs.

The decimal point is set at 13/12, and the switch settings are as follows:

SW 0 = 1 in machine column 1
SW 1 = Rollback control in machine column 1
SW 2 = 0001.5707 9632 6795 $(\pi/2)$
SW 3 = 0000.4342 9448 1903 $(\log_{10}e)$
SW 4 = 0002.3025 8509 2994 $(\log_e 10)$
SW 5 = 0000.1591 5494 3092 $(1/2\pi)$
SW 6 = 0006.2831 8530 7180 (2π)
SW 7 = Check tolerance

The setting of switch 7 is determined by the ranges of the variables a and b.

TABLE 10.3

Expression	Line	A	OP	B	C
h_0 = Rollback control	0501			-SW 0	h_0
h_7 = Check control	0502			-SW 0	h_7
Rollback control check	0503	SW 1		h_0	Check
$h_1 = 2n\pi = 0$	0506	XT0000		$+\lvert\ \rvert$	h_1
	0507			a_0	x_0
	0508			a_1	x_1
	0509			b_0	y_0
	0510			b_1	y_1
	0511	x_0	×	x_0	x_2
	0514	x_1	×	x_1	
$x_2 = x_0^2 + x_1^2$	0517	σ	+	x_2	x_2
$x_3 = x_1/x_0$	0518	x_1	÷	x_0	x_3
$x_4 = \log_{10}(x_0^2 + x_1^2)$	0523		log	x_2	x_4
	0527		$\tan^{-1}$	x_3	x_5
	0531	x_1	$\odot$	SW 2	g_0
	0532	σ	+	x_5	g_1
	0533	x_0	$\odot$	g_0	
	0534	$-\sigma$	+	g_1	
$x_5 = \tan^{-1}(x_1/x_0) + 2n\pi$	0535	σ	+	h_1	x_5
$x_6 = \frac{1}{2}\log_{10}(x_0^2 + x_1^2)$	0536	s_5	×	x_4	x_6
$x_7 = \log_{10}e\ [\tan^{-1}(x_1/x_0) + 2n\pi]$	0539	SW 3	×	x_5	x_7
$x_8 = \frac{1}{2}\log_e(x_0^2 + x_1^2)$	0542	SW 4	×	x_6	x_8
	0545	y_0	×	x_6	x_9
	0548	$-y_1$	+	x_7	
$x_9 = \log_{10}\rho$	0549	σ	+	x_9	x_9
	0550	y_0	×	x_5	y_2
	0553	y_1	×	x_8	
$y_2 = \phi$	0556	σ	+	y_2	y_2
$y_3 = \rho$	0557		exp	x_9	y_3
$y_4 = \cos\phi$	0561		cos	y_2	y_4
$y_5 = \sin\phi$	0565			r_3	y_5
$z_0 = \rho\cos\phi$	0566	y_3	×	y_4	z_0
	0569			h_7	Cond. call
$z_1 = \rho\sin\phi$	0570	y_3	×	y_5	z_1
Conditional call to checking routine	0573	CC0620			
	0574	SW 2	+	SW 2	
Conditional call = $\lvert x_5\rvert - \pi$	0575	$-\sigma$	+	$+\lvert x_5\rvert$	Cond. call
Increment rollback control	0576	h_0	+	SW 0	h_0
Automatic rollback	0577	CC0502			
	0578	XT0012		$+\lvert\ \rvert$	g_2
	0579	y_2	×	SW 5	g_3
	0582	σ	$\odot$	s_5	
	0583	σ	+	g_3	
	0584	σ	Shift by	$-g_2$	
	0589	σ	Shift by	g_2	
$h_1 = 2n\pi$	0594	σ	×	SW 6	h_1

TABLE 10.3 (continued)

Expression	Line	A	OP	B	C
	0597			z_0	c_0
	0598			z_1	c_1
h_7 = Check control	0599			SW 0	h_7
	0600	b_0	×	b_0	g_4
	0603	b_1	×	b_1	
	0606	σ	+	g_4	g_4
	0607			c_0	x_0
	0608			c_1	x_1
	0609	b_0	÷	g_4	y_0
$y_0 + iy_1 = (b_0 + ib_1)^{-1}$	0614	$-b_1$	÷	g_4	y_1
Compute as a check $(c_0 + ic_1)^{(b_0 + ib_1)^{-1}}$	0619	$C0511$			
	0620	z_0	-	a_0	
	0621	$+\lvert\sigma\rvert$	-	SW 7	Cond. call
	0622	z_1	-	a_1	g_5
Automatic rollback	0623	$CC0502$			
	0624	$+\lvert g_5\rvert$	-	SW 7	Cond. call
	0625				
Automatic rollback	0626	$CC0502$			

Example 3. It is required to tabulate the function $y(x)$ defined by the relation

$$y = y_0 \cos xy \tag{5}$$

in the region $0 \leq x_0 \leq 1$ for values of the parameter $0 < y_0 \leq 1$.

There are several numerical methods available for the solution of this problem, of which only one will be discussed in this example. By differentiation of Eq. (5) it may be shown that y satisfies the differential equation

$$yy'' + y(y+xy')^3 - 2(y')^2 = 0. \tag{6}$$

The solution of this equation for the boundary conditions $y(0) = y_0$, $y'(0) = 0$ is required.

Equation (6) may be rewritten in the form

$$\left.\begin{aligned} q &= -(y + px)^3 + \frac{2p^2}{y} \\ p &= \int_0^x q\,dx \\ y &= \int_0^x p\,dx. \end{aligned}\right\} \tag{7}$$

When mechanical quadrature rules are used in place of the integrals, Eqs. (7) form the basis for calculating the solution of the differential equation. In this example Milne's method will be employed, so that Eqs. (7) may be replaced by the approximate expressions

$$\left.\begin{aligned} {}^0p_5 &= p_1 + \tfrac{4h}{3}(2q_2 - q_3 + 2q_4) \\ {}^0y_5 &= y_3 + \tfrac{h}{3}(p_3 + 4p_4 + {}^0p_5) \\ {}^0q_5 &= q(x_5, {}^0y_5, {}^0p_5) \\ p_5 &= p_3 + \tfrac{h}{3}(q_3 + 4q_4 + {}^0q_5) \\ y_5 &= y_3 + \tfrac{h}{3}(p_3 + 4p_4 + p_5) \\ q_5 &= q(x_5, y_5, p_5). \end{aligned}\right\} \tag{8}$$

In Eqs. (8) the quantities y_i, p_i, and q_i are the values of y, p, and q for $x = x_i$, where $x_{i+1} - x_i = h$. The quantities ${}^0y_5$, ${}^0p_5$, and ${}^0q_5$ are first approximations to y_5, p_5, and q_5.

In order to determine y_5, p_5, and q_5, the previous values of y, p, and q appearing in Eqs. (8) must be known. Since only $y(0)$, $p(0)$, and $q(0)$ are known a priori, Eqs. (8) are not suitable for starting the solution. The method of Runge and Kutta will be used to obtain the required starting values. It may easily be shown that $y(x)$ and $q(x)$ are even functions, while $p(x)$ is an odd function. Some economy can be effected, therefore, by evaluating y and its derivatives for $x = -h$, 0, h, and $2h$ rather than for $x = 0$, h, $2h$, and $3h$. If y_0, p_0, and q_0 are the values of y, p, and q for $x = 0$, then $y_1 = y(h)$, $p_1 = p(h)$, and $q_1 = q(h)$ may be evaluated approximately by the following formulae:

$$\left.\begin{aligned}
f_1 &= hp_0 \\
g_1 &= h\cdot q(x_0, y_0, p_0) \\
f_2 &= h(p_0 + \tfrac{1}{2}g_1) \\
g_2 &= h\cdot q(x_0 + \tfrac{1}{2}h, y_0 + \tfrac{1}{2}f_1, p_0 + \tfrac{1}{2}g_1) \\
f_3 &= h(p_0 + \tfrac{1}{2}g_2) \\
g_3 &= h\cdot q(x_0 + \tfrac{1}{2}h, y_0 + \tfrac{1}{2}f_2, p_0 + \tfrac{1}{2}g_2) \\
f_4 &= h(p_0 + g_3) \\
g_4 &= h\cdot q(x_0 + h, y_0 + f_3, p_0 + g_3) \\
p_1 &= \tfrac{1}{6}(g_1 + 2g_2 + 2g_3 + g_4) + p_0 \\
y_1 &= \tfrac{1}{6}(f_1 + 2f_2 + 2f_3 + f_4) + y_0 \\
q_1 &= q(x_1, y_1, p_1).
\end{aligned}\right\} \qquad (9)$$

Equations (9) may also be employed to calculate y_2, p_2, and q_2 by advancing the subscripts of x, y, p, and q.

The defining relation, Eq. (5), may be used to check each value of y as it is computed. It is seen from Eqs. (8) that if y_5 is known to be correct, then p_5 is correct. The value of q_5, on the other hand, is not checked by a check on y_5, since q_5 does not enter into the computation of y_5. The difference between q_5 and ${}^0q_5$ should, however, be small, and this fact may be used to check q_5.

The errors in y, p, and q will, of course, depend upon the value of h selected. It is not usually a simple matter to make accurate evaluation of the magnitude of the error accumulated in the step-by-step solution of a differential equation. In the present instance it is estimated that, for $h = 0.01$, results good to at least six decimal places will be obtained. Solutions will be tabulated on a mesh of 0.01 in y_0.

The coding for the problem appears in Table 10.5. Lines 1201-1219 are used to set up various constants needed in the problem. It will be seen that y_0 is set up with the value 1.01; on line 1220 $h_1 = 0.01$ is subtracted from this to give the first value of $y_0 = 1.00$. Next x_5, y_5, and p_5 are set up with the starting values x_0, y_0, and p_0 and a call is made to line 1365. The coding on lines 1365-1386 is used to evaluate $q(x,y,p)$. The call on 1386 is back to line 1234, since this is the number placed in a_0. On lines 1234-1269 x, y, p, and q are recorded on mechanism 1.

The conditional call to line 1387 fails, so the transfers on lines 1268-1274 are executed next. On lines 1275-1359 the values of y, p, and q at $x = h$ are computed according to the formulae given in Eqs. (9). It will be noticed that calls are made to line 1365 each time the differential equation is to be evaluated. Further it will be remarked that the results are placed in registers x_5, y_5, p_5, and q_5; these registers were, of course, freed by the transfers coded on lines 1268-1274. On line 1359 a call is made to line 1234; the new values of x, y, p, and q are recorded and the entire Runge-Kutta process is repeated for $x = 2h$. In this case the conditional call to line 1387 succeeds and the main integration routine is started.

A list of the contents of certain storage registers, after the execution of line 1387, is given in Table 10.4 in order to clarify the status of the problem at the start of the main integration routine. Clearly, all the quantities required by Eqs. (8) are available.

TABLE 10.4

Register	Contents
p_2	$p(-h)$
y_3	$y(0)$
p_3	$p(0)$
q_3	$q(0)$
x_4	h
y_4	$y(h)$
p_4	$p(h)$
q_4	$q(h)$
x_5	$2h$
y_5	$y(2h)$
p_5	$p(2h)$
q_5	$q(2h)$

It is convenient to proceed from one point of the integration to the next by cyclic permutation of the subscripts appearing in Eqs. (8). This may be accomplished by means of the i register of the calculator. At the start of the calculation, on lines 1215-1219, the digits 2, 3, 4, 5, and 1 are transferred to registers z_1, z_2, z_3, z_4, and z_5. Then if the i register contains any of the digits 1-5 initially, the line of coding $z_i = i$ reg. will advance the i register to the next digit in the cycle.

In the coding of the present example unity is read into register t_1 on line 1224, and when line 1388 is executed, this is transferred to the i register. On line 1392 z_i is transferred to t_0 and thence to the i register on line 1394. Therefore at this point the i register and t_0 contain the next digit in the cycle after t_1.

On lines 1393 and 1395, the quantities 1388 and 1397 are transferred to registers c_0 and c_1. Register c_0 contains the line number to be called for the automatic rerun on the main integration tape, while c_1 contains the line to be called after all checks are passed. It should be noted that lines 1397-1404 are the same as lines 1388-1395, except that registers t_0 and t_1, and the line numbers 1388 and 1397 are interchanged.

Equations (8) are evaluated and checked by means of the coding on lines 1405-1499. The two conditional calls on lines 1450 and 1459 are for automatic rollback to the line stored in c_0; the call on 1499 is to the line stored in c_1. Usually this is line 1397 or 1388, in order to let the calculator proceed to the next point of the integration. When $x = 4.99$, however, the conditional call coded on line 1481 succeeds and register c_1 is changed to 1220. When line 1220 is called the value of y_0 is decreased by 0.01 and a new solution of the differential equation is started.

It should now be clear that the main integration routine is, in effect, divided into two sections in the same fashion as the routine of Example 1. In the present case, however, duplication of the main body of coding is obviated by the use of the t register and the line number register.

Starting Instructions

1. Decimal point 15/14.
2. Mechanism 1 used for recording.
3. Start line 1201.

Switch Settings

SW 0 = Rollback control in machine column 1
SW 1 = 00.16 6666 6666 6667
SW 2 = 1 in machine column 9
SW 3 = 1 in machine column 9
SW 4 = 04.99 0000 0000 0000

Rerun Instructions

Line	Instructions
1228	Rollback control check. Called from line 1350. Number of reruns in t_2. Start next line.
1240	Record check. Start line 1234.
1248	Record check. Start line 1242.
1257	Record check. Start line 1251.
1265	Record check. Start line 1259.
1391	Rollback control check. Called from 1450 or 1459. Number of reruns in t_2. Start next line.
1400	Rollback control check. Called from line 1450 or 1459. Number of reruns in t_2. Start next line.

Rerun Instructions (continued)

Line	Instructions
1458	Computation check. Start line 1499.
1462	Computation check. Start line 1499.
1469	Record check. Start line 1463.
1477	Record check. Start line 1471.
1489	Record check. Start line 1483.
1497	Record check. Start line 1491.

Code Stop

Line 1223. End of problem.

TABLE 10.5

Expression	Line	A	OP	B	C
Start line. $y_0 = y(0,y)$	1201	$XT0100$		$+\lvert\ \rvert$	y_0
$h_0 = \Delta x = h$	1202	$XT0001$		$+\lvert\ \rvert$	h_0
$h_1 = \Delta y$	1203	$XT0001$		$+\lvert\ \rvert$	h_1
$p_0 = y'(0,y)$	1204	$XT0000$		$+\lvert\ \rvert$	p_0
	1205	$XT0000$		$+\lvert\ \rvert$	x_0
	1206	$XT0400$		$+\lvert\ \rvert$	b_0
Switch 4 = maximum x	1207	SW 4	-	h_0	t_3
	1208	y_0	+	h_1	y_0
t_2 = Rollback control	1209			-10^0	t_2
	1210	SW 1	+	SW 1	
$h_3 = h/3$	1211	σ	$\times$	h_0	h_3
$h_4 = 4h/3$	1214	σ	+	h_0	h_4
	1215	$XT0002$		$+\lvert\ \rvert$	z_1
	1216	$XT0003$		$+\lvert\ \rvert$	z_2
	1217	$XT0004$		$+\lvert\ \rvert$	z_3
	1218	$XT0005$		$+\lvert\ \rvert$	z_4
	1219	$XT0001$		$+\lvert\ \rvert$	z_5
	1220	y_0	-	h_1	y_0
End of problem	1221	σ			Stop
	1224	$XT0001$		$+\lvert\ \rvert$	t_1
	1225	$XT0005$		$+\lvert\ \rvert$	t reg.
Rollback control check	1226	SW 0	-	t_2	Check
	1229			x_0	x_5
	1230			y_0	y_5
	1231			p_0	p_5
	1232	$XT1234$		$+\lvert\ \rvert$	a_0
	1233	$C1365$			
Record x	1234			x_5	Rec. 1
Record y	1242			y_5	Rec. 1
	1250	x_5	-	h_0	Cond. call
Record y'	1251			p_5	Rec. 1
Record y''	1259			g_5	Rec. 1
End of starting procedure	1267	$CC1387$			
	1268			y_4	y_3
	1269			p_4	p_3
	1270			g_4	g_3
	1271			x_5	x_4
	1272			y_5	y_4
	1273			p_5	p_4
	1274			g_5	g_4
$f_1 = hy'$	1275	h_0	$\times$	p_4	f_1
$g_1 = hy''(x,y,y')$	1278	h_0	$\times$	g_4	g_1
	1281	σ	$\times$	s_5	
	1284	σ	+	p_4	p_5
$f_2 = h(y' + \frac{1}{2}g_1)$	1285	σ	$\times$	h_0	f_2
	1288	s_5	$\times$	f_1	

TABLE 10.5 (continued)

Expression	Line	A	OP	B	C
	1291	σ	+	y_4	y_5
	1292	s_5	×	h_0	
	1295	σ	+	x_4	x_5
	1296	$XT1298$		$+\vert\ \vert$	a_0
	1297	$C1365$			
$g_2 = hy''(x+\frac{1}{2}h, y+\frac{1}{2}f_1, y'+\frac{1}{2}g_1)$	1298	g_5	×	h_0	g_2
	1301	σ	×	s_5	
	1304	σ	+	p_4	p_5
$f_3 = h(y'+\frac{1}{2}g_2)$	1305	σ	×	h_0	f_3
	1308	s_5	×	f_2	
	1311	σ	+	y_4	y_5
	1312	$XT1314$		$+\vert\ \vert$	a_0
	1313	$C1365$			
$g_3 = hy''(x+\frac{1}{2}h, y+\frac{1}{2}f_2, y'+\frac{1}{2}g_2)$	1314	g_5	×	h_0	g_3
	1317	σ	+	p_4	p_5
$f_4 = h(y'+g_3)$	1318	σ	×	h_0	f_4
	1321	y_4	+	f_3	y_5
	1322	x_4	+	h_0	x_5
	1323	$XT1325$		$+\vert\ \vert$	a_0
	1324	$C1365$			
	1325	f_4	+	f_3	
	1326	σ	+	f_3	
	1327	σ	+	f_2	
	1328	σ	+	f_2	
	1329	σ	+	f_1	
$\Delta y = \frac{1}{6}(f_1+2f_2+2f_3+f_4)$	1330	σ	×	SW 1	
	1333	σ	+	y_4	y_5
	1334	σ	×	x_5	a_1
$g_4 = hy''(x+h, y+f_3, y'+g_3)$	1337	g_5	×	h_0	g_4
$a_2 = \cos xy$	1340		cos	a_1	a_2
$a_3 = y_0 \cos xy$	1344	σ	×	y_0	a_3
	1347	σ	-	y_5	
	1348	$+\vert\sigma\vert$	-	SW 2	Cond. call
Increment rollback control	1349	t_2	+	10^0	t_2
Automatic rollback	1350	$CC1224$			
	1351	g_4	+	g_3	
	1352	σ	+	g_3	
	1353	σ	+	g_2	
	1354	σ	+	g_2	
	1355	σ	+	g_1	
$\Delta y' = \frac{1}{6}(g_1+2g_2+2g_3+g_4)$	1356	σ	+	SW 1	
	1357	σ	+	p_4	p_5
t_2 = Rollback control	1358			-10^0	t_2
	1359	$C1234$			
	1360	p_t	+	p_8	
	1361	σ	×	h_3	

TABLE 10.5 (continued)

Expression	Line	*A*	OP	*B*	*C*
$y_i = y_{i-2} + \frac{h}{3}(y'_i + 4y'_{i-1} + y'_{i-2})$	1364	σ	+	y_6	y_i
	1365	x_i	×	p_i	
$a_i = y + xy'$	1368	σ	+	y_i	a_i
	1369	p_i	+	p_i	
	1370	σ	÷	y_i	
$b_i = 2\frac{(y')^2}{y}$	1375	σ	×	p_i	b_i
	1378	a_i	×	a_i	
	1381	σ	×	a_i	
$g_i = y'' = 2\frac{(y')^2}{y} - (y+xy')^3$	1384	$-\sigma$	+	b_i	g_i
	1385			a_0	*LN*
	1386	*C*0000			
$y'(-x) = -y'(x)$	1387			$-p_4$	p_2
	1388			t_1	i reg.
Rollback control check	1389	SW 0	-	t_2	Check
	1392			z_i	t_0
	1393	*XT*1388		$+\lvert\ \rvert$	c_0
	1394			t_0	i reg.
	1395	*XT*1397		$+\lvert\ \rvert$	c_1
	1396	*C*1405			
	1397			t_0	i reg.
Rollback control check	1398	SW 0	-	t_2	Check
	1401			z_i	t_1
	1402	*XT*1397		$+\lvert\ \rvert$	c_0
	1403			t_1	i reg.
	1404	*XT*1388		$+\lvert\ \rvert$	c_1
$p_6 = y'_{i-4}$	1405			p_i	p_6
	1406			z_i	i reg.
Increment rollback control	1407	t_2	+	10^0	t_2
$g_6 = 2y''_{i-3}$	1408	g_i	+	g_i	g_6
	1409			z_i	i reg.
	1410	*XT*1432		$+\lvert\ \rvert$	a_0
$y_6 = y_{i-2}$	1411			y_i	y_6
$g_7 = y''_{i-2}$	1412			g_i	g_7
$p_7 = y'_{i-2}$	1413			p_i	p_7
	1414			z_i	i reg.
$g_8 = 2y''_{i-3} - y''_{i-2}$	1415	g_6	-	g_7	g_8
$g_9 = 2y''_{i-1}$	1416	g_i	+	g_i	g_9
$g_8 = 2y''_{i-1} - y''_{i-2} + 2y''_{i-3}$	1417	σ	+	g_8	g_8
	1418	g_9	+	g_9	
$g_9 = 4y''_{i-1} + y''_{i-2}$	1419	σ	+	g_7	g_9
	1420	b_0	×	p_i	
$p_8 = 4y'_{i-1} + y'_{i-2}$	1423	σ	+	p_7	p_8
	1424	x_i	+	h_0	x_6
	1425			z_i	i reg.
	1426	g_8	×	h_4	

TABLE 10.5 (continued)

Expression	Line	A	OP	B	C
$p_t = {}^0y'_t = y'_{t-4} + \frac{4h}{3}(2y''_{t-1} - y''_{t-2} + 2y''_{t-3})$	1429	σ	+	p_6	p_t
$x_t = x_{t-1} + h$	1430			x_6	x_t
	1431	$C1360$			
	1432	g_t	+	g_9	
	1433	σ	×	h_3	
$p_t = {}^1y'_t = y'_{t-2} + \frac{h}{3}({}^0y''_t + 4y''_{t-1} + y''_{t-2})$	1436	σ	+	p_7	p_t
$w_0 = {}^0y''$	1437			g_t	w_0
	1438	$XT1440$		$+\vert\ \vert$	a_0
	1439	$C1360$			
	1440	x_t	×	y_t	y_7
$w_t = {}^0y'' - {}^1y''$	1443	w_0	-	g_t	w_t
	1444	$+\vert\sigma\vert$	-	SW 3	Cond. call
$y_8 = \cos xy$	1445		cos	y_7	y_8
	1449			c_0	LN
Automatic rollback to line 1388 or 1397	1450	$CC0000$			
$y_9 = y_0 \cos xy$	1451	y_8	×	y_0	y_9
	1454	σ	-	y_t	d_t
	1455	$+\vert\sigma\vert$	-	SW 2	Cond. call
Computation check $\vert{}^0y'' - {}^1y''\vert$	1456	$-\vert w_t\vert$	+	SW 3	Check
Automatic rollback to line 1388 or 1397	1459	$CC0000$			
Computation check $\vert y - y_0 \cos xy\vert$	1460	$-\vert d_t\vert$	+	SW 2	Check
Record x	1463			x_t	Rec. 1
Record y	1471			y_t	Rec. 1
	1479	x_t	-	t_3	Cond. call
t_2 = Rollback control	1480			-10^0	t_2
Start new item	1481	$CC1500$			
	1482			c_1	LN
Record y'	1483			p_t	Rec. 1
Record y''	1491			g_t	Rec. 1
Call line 1397, 1388, or 1220	1499	$C0000$			
	1500	$XT1220$		$+\vert\ \vert$	c_1
	1501	$C1482$			

Example 4. It is required to tabulate the function $y(x)$ defined by the relation

$$x = \frac{166.9215y^7}{(7y^2-1)^{5/2}} - 1 \tag{10}$$

in the range $1 \leq x < 50$. The solution required passes through the point (1, 1.0466); the value of y increases with increasing x. In order to demonstrate the use of the slow storage system, Eq. (10) will be solved by inverse interpolation.

It may easily be shown that $y(50) \doteq 6.3$. In order to avoid the occurrence of numbers of excessive magnitude, Eq. (10) may be rewritten in the form

$$x = Cy^2[y(y^2-1/7)^{-\frac{1}{2}}]^5 - 1 \tag{11}$$

where $C = 166.9215 \times 7^{-5/2} \doteq 1.28755911798795$.

The problem resolves itself into two parts. The first part consists of the tabulation of $x(y)$ from Eq. (11), and the second part is the inverse interpolation in this table to obtain $y(x)$. The method selected for inverse interpolation is essentially Aitken's process. If $x(y)$ is tabulated on a mesh of 0.01 in y, third order interpolation is required to obtain six places of decimals in $y(x)$. The formulae are then as follows:

$$\left.\begin{aligned}
z_2 &= z_1 - 0.01\frac{d_0}{d_4} \\
f_0 &= 0.01\frac{d_0}{d_4} - 0.02\frac{d_0}{d_5} \\
f_1 &= 0.01\frac{d_0}{d_4} - 0.03\frac{d_0}{d_6} \\
z_3 &= z_2 - f_0\frac{d_1}{d_7} \\
f_2 &= f_0\frac{d_1}{d_7} - f_1\frac{d_1}{d_8} \\
y(x) &= z_3 - f_2\frac{d_2}{d_9} \\
&\text{where} \\
d_0 &= x_0 - x \qquad & d_5 &= d_2 - d_0 \\
d_1 &= x_1 - x \qquad & d_6 &= d_3 - d_0 \\
d_2 &= x_2 - x \qquad & d_7 &= d_2 - d_1 \\
d_3 &= x_3 - x \qquad & d_8 &= d_3 - d_1 \\
d_4 &= d_1 - d_0 \qquad & d_9 &= d_3 - d_2
\end{aligned}\right\} \tag{12}$$

and

$$z_1 = y(x_0).$$

Here x_0, x_1, x_2, and x_3 are tabular values of $x(y)$, and x is the argument for which $y(x)$ is required. It is assumed that this argument lies between x_1 and x_2.

The coding for the problem is given in Table 10.6. Lines 2001-2100 constitute the routine to compute $x(y)$ and store the results in slow storage. On line 2011, 0.88 is transferred to an argument counter, y_1, and on the next line a call is made to line 2029. Here 0.98 is read to y_0, -0098 is transferred to the δ register and y_0 is transferred to y_2. On lines 2033-2060, Eq. (11) is evaluated and the result placed in γ_i. In this case, since y_2 was read to the i register on line 2047, $x(0.98)$ is placed in γ_8. The quantity in y_2 is incremented on line 2062. The call on line 2063 is to line 2033, since this is the number in c_8. The lines 2033-2063 are repeated with the result that $x(0.99)$ is placed in γ_9. This time, however, the call on line 2063 is to line 2013, since this is the number in c_9. The two values of $x(y)$ just computed are required later in the difference check. On line 2013 $x(0.98)$ and $x(0.99)$ are transferred from fast storage to slow storage registers 0098 and 0099, respectively. Since the decimal point for the problem is 15/14 and y is incremented in steps of 0.01, the argument y is suitable for use as a slow storage control number without alteration.

On line 2016, c_9 is changed to 2064. On line 2017 the argument 0.90 is read into y_0. This is incremented by 0.10 and the result transferred to y_1 on line 2021. Register y_0 is also read to y_3 on line 2024. Lines 2026-2032 are the same as lines 2018-2024 except that the rôles of y_0 and y_1 are interchanged. This is to provide for two-section operation as in the previous example.

Next the coding on lines 2033-2063 is executed ten times, to store the values $x(1.00), x(1.01), \ldots, x(1.09)$ in registers $\gamma_0, \gamma_1, \ldots, \gamma_9$. When the call to line 2064 is made, these quantities are transferred to slow storage registers 0100,0101,...,0109. Next the argument in y_3 is read positively to the δ register and on line 2078 the transfer from slow to fast storage is performed. The quantities in registers γ_i are thus transferred from slow storage to registers β_i, while the two previously computed results are transferred to α_8 and α_9. These twelve quantities are differenced on lines 2081-2096; the values of $x(y)$ in α_8 and α_9 are introduced to insure continuity between the groups of ten values of $x(y)$. It will be noted that d_4 is positive in case the absolute value of the difference d_3 is less than the tolerance of 10^{-8}, and d_5 is negative since $y_0 = 0.90$ and $y_1 = 1.00$. On account of the choice on line 2098, the conditional call on line 2100 will succeed only if d_4 is negative.

In this case line 2018 is called and an automatic rollback effected. If the conditional call fails, line 2026 is called, and the next ten values of $x(y)$ are computed, transferred to slow storage, and differenced. On this round d_5 is positive and the rôles of lines 2018 and 2026 are interchanged. A remark should be made regarding the conditional call on line 2069. When d_4 is negative, which is the case when a rerun is being made, the call succeeds and the reset of the automatic rollback control counter t_0 is omitted.

When the argument y_2 reaches the value 6.4, the conditional call on line 2080 succeeds and the inverse interpolation routine is started. On lines 2102-2115 are coded various preliminary operations. The numbers in slow storage registers 0090-0109 are transferred to fast storage, and $-\beta_8$, i.e., $-x(1.08)$, is read into register a_0. Then 0100 is read into the δ register and the contents of slow storage registers 0100-0119 are transferred to fast storage. On lines 2132-2152 the numbers in α_0 through β_3, together with a_0, are differenced and an automatic rollback is performed in case the absolute value of the difference is greater than 10^{-8}.

In Eqs. (12) it is assumed that the argument x lies between x_1 and x_2. The first argument for which inverse interpolation is to be performed is $x = 1.00$, which certainly lies between $x(1.00)$ and $x(1.19)$, or in other words between α_0 and β_9; indeed steps will be taken to insure that the argument x always lies between α_1 and β_2 and hence that the quantities x_0, x_1, x_2, x_3 of Eqs. (12) always are to be found among the registers α_0 through β_3. There remains therefore the problem of selecting the α or β register containing x_0; this may be solved by using the $\alpha\beta_t$ code and performing a linear interpolation from α_0 to α_9. Let

$$z = \frac{0.09}{\alpha_9-\alpha_0}(x-\alpha_0) + z_0,$$

where z_0 is the value of y such that $\alpha_0 = x(z_0+0.01)$, and let

$$z_1 = 0.01\ \text{Int.}(100z),$$

where the symbol Int.(100z) denotes the integral part of 100z. After z_1 is transferred to the t register, $\alpha\beta_t$ is the required register containing x_0 of Eqs. (12). The quantity z_0 and the ratio $w_8 = 0.09/\alpha_0-\alpha_9$ are obtained on lines 2157-2176. On line 2178 a call is made; in the first instance this is to line 2179. The computation of the quantity z_1 is coded on lines 2180-2194.

The evaluation of Eqs. (12) proceeds in a straightforward fashion and is coded on lines 2197-2290. It will be observed that the coding on lines 2033-2063 is employed to evaluate Eq. (11) for checking purposes. To facilitate the automatic rollback a two-section routine is used, the coding on lines 2291-2403 being essentially the same as the coding on lines 2179-2290, except that the argument registers x_0 and x_1 are interchanged. When the quantity $\frac{0.09}{\alpha_9 - \alpha_0}(x - \alpha_0)$, which is stored in b_5, becomes greater than 0.11 the argument x lies between β_1 and β_2. At this point it is possible to bring the next ten values of $x(y)$ from slow to fast storage. The conditional calls on lines 2290 and 2402 are for this purpose. It should be noted that z_1 is the proper slow storage control number, so that the coding starting on line 2116, previously described, will cause the transfer of the required values of $x(y)$.

Starting Instructions

1. Set decimal point switch to 15/14.
2. Set up mechanism 1 for recording.
3. Start line 2001.

Switch Settings

SW 1 = 00.14 2857 1428 5714
SW 2 = 01.28 7559 1179 8795
SW 3 = 05.00 0000 0000 0000
SW 4 = Rollback control in machine column 1.

Rerun Instructions

Line	Instructions
2020	Rollback control check. Called from line 2100. Number of reruns in t_0. Start next line.
2028	Rollback control check. Called from line 2101. Number of reruns in t_0. Start next line.
2121	Rollback control check. Called from line 2155. Number of reruns in t_0. Start next line.
2160	Rollback control check. Called from line 2267 or 2379. Number of reruns in t_0. Start next line.
2270	Computation check. Start line 2157.
2277	Record check. Start line 2271.
2285	Record check. Start line 2279.
2382	Computation check. Start line 2157.
2389	Record check. Start line 2383.
2397	Record check. Start line 2391.

Code Stop

Line 2124. End of problem.

TABLE 10.6

Expression	Line	A	OP	B	C
Start line	2001	XT2033		+| |	c_0
	2002	σ			c_1
	2003	σ			c_2
	2004	σ			c_3
"Call line" numbers	2005	σ			c_4
	2006	σ			c_5
	2007	σ			c_6
	2008	σ			c_7
	2009	σ			c_8
	2010	XT2013		+| |	c_9
$y_1 = y$	2011	XT0088		+| |	y_1
	2012	C2029			
$x(.98)$ and $x(.99)$ to slow storage	2013				F/S
t_0 = Rollback control	2015			-10^0	t_0
c_9 = "Call line" number	2016	XT2064		+| |	c_9
$y_0 = y$	2017	XT0090		+| |	y_0
Rollback control check	2018	SW 4	-	t_0	Check
Increment argument	2021	y_0	+	10^{13}	y_1
Slow storage control	2022	$-\sigma$			δ
	2023			y_1	y_2
	2024			y_0	y_3
	2025	C2033			
Rollback control check	2026	SW 4	-	t_0	Check
Increment argument	2029	y_1	+	10^{13}	y_0
Slow storage control	2030	$-\sigma$			δ
	2031			y_0	y_2
	2032			y_1	y_3
$w_0 = y^2$	2033	y_2	$\times$	y_2	w_0
$w_1 = y^2 - 1/7$	2036	σ	-	SW 1	w_1
$w_2 = Cy^2$	2037	w_0	$\times$	SW 2	w_2
$w_3 = (y^2 - 1/7)^{-\frac{1}{2}}$	2040		$1/\sqrt{\ }$	w_1	w_3
$w_4 = y(y^2 - 1/7)^{-\frac{1}{2}}$	2044	σ	$\times$	y_2	w_4
	2047			y_2	i reg.
$w_5 = y^2(y^2 - 1/7)^{-1}$	2048	w_4	$\times$	w_4	w_5
	2051	σ	$\times$	w_4	
$w_6 = y^5(y^2 - 1/7)^{-5/2}$	2054	σ	$\times$	w_5	w_6
	2057	σ	$\times$	w_2	
$\gamma_t = Cy^7(y^2 - 1/7)^{-5/2} - 1$	2060	σ	-	10^{14}	γ_t
	2061			c_t	LN
Increment argument	2062	y_2	+	10^{12}	y_2
	2063	C0000			
10 values of x to slow storage	2064				F/S
Slow storage control	2066			y_3	δ
	2067			$-d_4$	Cond. call
t_1 = maximum y	2068	XT0639		+| |	t_1
Call only on reruns	2069	CC2076			

TABLE 10.6 (continued)

Expression	Line	A	OP	B	C
	2070				
	2071				
	2072				
t_0 = Rollback control	2073			-10^0	t_0
	2074				
	2075				
	2076	y_2	-	t_1	Cond. call
	2077	y_0	-	y_1	d_5
Last 20 values of x to fast storage	2078				S/F
Commence inverse interpolation	2080	$CC2102$			
Difference last 12 values of x	2081	$-\beta_1$	-	β_6	d_0
	2082	σ	+	β_0	
	2083	σ	+	β_2	
	2084	σ	+	β_5	
	2085	σ	+	β_7	d_1
	2086	$-\alpha_9$	-	β_8	d_2
	2087	d_1	+	d_1	
	2088	σ	+	d_1	
	2089	σ	+	d_0	
	2090	σ	+	d_2	
	2091	σ	×	SW 3	
	2094	σ	+	d_2	
	2095	σ	+	α_8	
	2096	σ	+	β_9	d_3
	2097	$-\lvert\sigma\rvert$	+	10^6	d_4
	2098	σ	$\odot$	d_5	Cond. call
Increment rollback control	2099	t_0	+	10^0	t_0
	2100	$CC2018$			
	2101	$C2026$			
Slow storage control	2102	$XT0090$		$+\lvert\ \rvert$	δ
z_1 = Slow storage control number	2103			10^{14}	z_1
$x_1 = x$	2104	σ	-	10^{12}	x_1
t_0 = Rollback control	2105			-10^0	t_0
b_0 = "Call line" number	2106	$XT2179$		$+\lvert\ \rvert$	b_0
b_1 = Shift control number	2107	$XT0013$		$+\lvert\ \rvert$	b_1
b_2 = Shift control number	2108	$XT0012$		$+\lvert\ \rvert$	b_2
b_3 = maximum x	2109	$XT5000$		$+\lvert\ \rvert$	b_3
b_4 = 0.09	2110	$XT0009$		$+\lvert\ \rvert$	b_4
	2111	$XT0011$		$+\lvert\ \rvert$	b_5
	2112	$XT0000$		$+\lvert\ \rvert$	a_4
	2113				
First 12 values of x to fast storage	2114				S/F
$z_8 = z_1$	2116			z_1	z_8
	2117	$+\lvert a_4\rvert$	-	10^6	Cond. call
Slow storage control	2118			z_8	δ

TABLE 10.6 (continued)

Expression	Line	A	OP	B	C
Rollback control check	2119	SW 4	-	t_0	Check
End of problem	2122	b_3	-	x_0	Stop
	2125	CC2130			
	2126				
	2127			$-\beta_8$	a_0
	2128				
	2129				
20 values of x to fast storage	2130				S/F
	2132	$-\alpha_3$	-	β_0	a_1
	2133	σ	+	α_2	
	2134	σ	+	α_4	
	2135	σ	+	α_9	
	2136	σ	+	β_1	a_2
	2137	$-\alpha_1$	-	α_5	
	2138	σ	-	β_2	a_3
	2139	σ	-	α_8	
	2140	σ	+	a_2	
Difference 14 values of x	2141	σ	+	a_2	
	2142	σ	+	a_2	
	2143	σ	+	a_1	
	2144	σ	$\times$	SW 3	
	2147	σ	+	a_0	
	2148	σ	+	a_3	
	2149	σ	+	α_0	
	2150	σ	+	α_6	
	2151	σ	+	α_7	
	2152	σ	+	β_3	a_4
	2153	$+\lvert\sigma\rvert$	-	10^6	Cond. call
Increment rollback control	2154	t_0	+	10^0	t_0
Automatic rollback	2155	CC2116			
t_0 = Rollback control	2156			-10^0	t_0
	2157	α_9	-	α_0	w_9
Rollback control check	2158	SW 4	-	t_0	Check
$w_8 = \dfrac{.09}{\alpha_9 - \alpha_0}$	2161	b_4	$\div$	w_9	w_8
	2166	x_8	shift by	$-b_1$	
	2171	σ	shift by	b_1	
$x_0 = 0.1$ Int.$(10\ x_8) - 0.01$	2176	σ	-	10^{12}	x_0
	2177			b_0	LN
	2178	C0000			
Increment argument	2179	x_1	+	10^{12}	x_0
	2180	σ	-	α_0	
	2181	σ	$\times$	w_8	b_6
$b_6 = w_8(x - \alpha_0)$	2184	σ	+	x_0	
	2185	σ	shift by	$-b_2$	
	2190	σ	shift by	b_2	x_1
$x_1 = 0.01$ Int.$[100(b_6 + x_0)]$					

TABLE 10.6 (continued)

Expression	Line	A	OP	B	C
	2195	σ			t reg.
$t_1 = t$ register control	2196	z_1	+	10^{12}	t_1
	2197	$\alpha\beta_t$	-	x_0	d_0
	2198			t_1	t reg.
	2199	t_1	+	10^{12}	t_1
	2200	$\alpha\beta_t$	-	x_0	d_1
	2201			t_1	t reg.
	2202	d_1	-	d_0	d_4
	2203	$\alpha\beta_t$	-	x_0	d_2
	2204	t_1	+	10^{12}	t reg.
	2205	d_2	-	d_0	d_5
	2206	d_2	-	d_1	d_7
	2207	$\alpha\beta_t$	-	x_0	d_3
	2208	σ	-	d_0	d_6
	2209	d_3	-	d_1	d_8
	2210	d_3	-	d_2	d_9
	2211	d_0	$\times$	10^{12}	e_0
	2214	σ	$\div$	d_4	e_1
	2219	$-\sigma$	+	z_1	z_2
	2220	e_0	+	e_0	e_2
	2221	σ	$\div$	d_5	
	2226	$-\sigma$	+	e_1	f_0
	2227	e_0	+	e_2	
	2228	σ	$\div$	d_6	
	2233	$-\sigma$	+	e_1	f_1
	2234	d_1	$\div$	d_7	
	2239	σ	$\times$	f_0	e_3
	2242	$-\sigma$	+	z_2	z_3
	2243	d_1	$\div$	d_8	
	2248	σ	$\times$	f_1	
	2251	$-\sigma$	+	e_3	f_2
	2252	d_2	$\div$	d_9	
	2257	σ	$\times$	f_2	
	2260	$-\sigma$	+	z_3	t reg.
	2261	σ			y_2
	2262	XT2264		$+\lvert\ \rvert$	c_t
	2263	C2033			
	2264	x_0	-	γ_t	h_0
	2265	$+\lvert\sigma\rvert$	-	10^8	Cond. call
Increment rollback control	2266	t_0	+	10^0	t_0
Automatic rollback	2267	CC2157			
Computation check	2268	$-\lvert h_0\rvert$	+	10^8	Check
Record x	2271			x_0	Rec. 1
Record y	2279			y_2	Rec. 1
	2287	b_6	-	b_5	Cond. call

TABLE 10.6 (continued)

Expression	Line	A	OP	B	C
t_0 = Rollback control	2288			-10^0	t_0
	2289	$XT2291$		$+\lvert\ \rvert$	b_0
Conditional call to S/F transfer routine	2290	$CC2116$			
Increment argument	2291	x_0	+	10^{12}	x_1
	2292	σ	-	α_0	
$b_6 = w_8(x - \alpha_0)$	2293	σ	×	w_8	b_6
	2296	σ	+	z_0	
	2297	σ	shift by	$-b_2$	
$z_1 = 0.01$ Int.$[100(b_6 + z_0)]$	2302	σ	shift by	b_2	z_1
	2307	σ			t reg.
t_1 = t register control	2308	z_1	+	10^{12}	t_1
	2309	$\alpha\beta_t$	-	x_1	d_0
	2310			t_1	t reg.
	2311	t_1	+	10^{12}	t_1
	2312	$\alpha\beta_t$	-	x_1	d_1
	2313			t_1	t reg.
	2314	d_1	-	d_0	d_4
	2315	$\alpha\beta_t$	-	x_1	d_2
	2316	t_1	+	10^{12}	t reg.
	2317	d_2	-	d_0	d_5
	2318	d_2	-	d_1	d_7
	2319	$\alpha\beta_t$	-	x_1	d_3
	2320	σ	-	d_0	d_6
	2321	d_3	-	d_1	d_8
	2322	d_3	-	d_2	d_9
	2323	d_0	×	10^{12}	e_0
	2326	σ	÷	d_4	e_1
	2331	$-\sigma$	+	z_1	z_2
	2332	e_0	+	e_0	e_2
	2333	σ	÷	d_5	
	2338	$-\sigma$	+	e_1	f_0
	2339	e_0	+	e_2	
	2340	σ	÷	d_6	
	2345	$-\sigma$	+	e_1	f_1
	2346	d_1	÷	d_7	
	2351	σ	×	f_0	e_3
	2354	$-\sigma$	+	z_2	z_3
	2355	d_1	÷	d_8	
	2360	σ	×	f_1	
	2363	$-\sigma$	+	e_3	f_2
	2364	d_2	÷	d_9	
	2369	σ	×	f_2	
	2372	$-\sigma$	+	z_3	t reg.
	2373	σ			y_2
	2374	$XT2376$		$+\lvert\ \rvert$	c_t

TABLE 10.6 (continued)

Expression	Line	*A*	OP	*B*	*C*
	2375	*C*2033			
	2376	x_1	-	γ_t	h_0
	2377	$+\lvert\sigma\rvert$	-	10^8	Cond. call
Increment rollback control	2378	t_0	+	10^0	t_0
Automatic rollback	2379	*CC*2157			
Computation check	2380	$-\lvert h_0\rvert$	+	10^8	Check
Record x	2383			x_1	Rec. 1
Record y	2391			y_2	Rec. 1
	2399	b_6	-	b_5	Cond. call
t_0 = Rollback control	2400			-10^0	t_0
	2401	*XT*2179		$+\lvert\ \rvert$	b_0
Conditional call to S/F transfer routine	2402	*CC*2116			
	2403	*C*2179			

APPENDIX I

CONSTANTS IN FAST STORAGE

Channel and Number	Time	Constant
01	0	1825 7260 0000 0000
	1	1490 6830 0000 0000
	2	1154 6390 0000 0000
	3	0894 4100 0000 0000
	4	0730 2900 0000 0000
	5	0577 3200 0000 0000
	6	0447 2050 0000 0000
	7	0365 1450 0000 0000
	8	0288 6600 0000 0000
	9	0223 6020 0000 0000
02	0	0829 8760 0000 0000
	1	0552 1050 0000 0000
	2	0329 8970 0000 0000
	3	0198 7580 0000 0000
	4	0132 7800 0000 0000
	5	0082 4740 0000 0000
	6	0049 6890 0000 0000
	7	0033 1950 0000 0000
	8	0020 6190 0000 0000
	9	0012 4220 0000 0000
03	0	1433 4100 0000 0000
	1	1295 3420 0000 0000
	2	1140 2340 0000 0000
	3	1003 3670 0000 0000
	4	0906 5680 0000 0000
	5	0806 2670 0000 0000
	6	0709 4880 0000 0000
	7	0641 0410 0000 0000
	8	0570 1170 0000 0000
	9	0501 6840 0000 0000
04	0	0434 9670 0000 0000
	1	0320 4990 0000 0000
	2	0217 9330 0000 0000
	3	0148 9540 0000 0000
	4	0110 0390 0000 0000
	5	0077 0510 0000 0000
	6	0052 6630 0000 0000
	7	0038 9050 0000 0000
	8	0027 2420 0000 0000
	9	0018 6190 0000 0000

Channel and Number	Time	Constant
05	0	0909 0909 0909 0909
	1	0740 7407 4074 0741
	2	0571 4285 7142 8571
	3	0444 4444 4444 4444
	4	0363 6363 6363 6364
	5	0285 7142 8571 4286
	6	0222 2222 2222 2222
	7	0181 8181 8181 8182
	8	0142 8571 4285 7143
	9	0111 1111 1111 1111
06	0	0041 3926 8515 8226
	1	0130 3337 6849 5007
	2	0243 0380 4868 6295
	3	0352 1825 1811 1363
	4	0439 3326 9383 0264
	5	0544 0680 4435 0277
	6	0653 2125 1377 5345
	7	0740 3626 8949 4245
	8	0845 0980 4001 4258
	9	0954 2425 0943 9326
07	0	1000 0000 0000 0000
	1	1258 9254 1179 4167
	2	1584 8931 9246 1113
	3	1995 2623 1496 8880
	4	2511 8864 3150 9580
	5	3162 2776 6016 8379
	6	3981 0717 0553 4973
	7	5011 8723 3627 2723
	8	6309 5734 4480 1932
	9	7943 2823 4724 2815
08	0	0000 0278 3841 6808
	1	0000 1970 7478 5410
	2	0001 3892 0356 4614
	3	0008 3332 9037 0962
	4	0041 6666 7010 5326
	5	0166 6666 6651 2241
	6	0500 0000 0000 3472
	7	0999 9999 9999 9970
	8	not used
	9	0023 0258 5092 9940

Channel Number	Time	Constant
09	0	0210 0000 0000 0000
	1	0666 6666 6666 6667
	2	1500 0000 0000 0000
	3	4761 9047 6190 4762
	4	0066 8063 4779 3420
	5	-0090 2592 1742 5174
	6	0111 0895 3771 6742
	7	-0142 8567 5706 1678
	8	0199 9999 9644 7985
	9	-0333 3333 3331 8890
10	0	0206 9921 9421 9821
	1	0588 0026 0354 7568
	2	0982 7937 2324 7329
	3	1363 8041 3257 5076
	4	0999 9999 9999 9983
	5	0301 0299 9566 3981
	6-9	not used
11	0	0500 0000 0000 0000
	1	1500 0000 0000 0000
	2	2000 0000 0000 0000
	3	0316 2277 6601 6838
	4	not used
	5	0000 5000 0000 0000
	6	0001 5000 0000 0000
	7	5000 0000 0000 0000
	8	0000 0000 0000 0050
	9	0041 7776 6467 8944

Channel Number	Time	Constant
12	0	-0045 7221 5941 9618
	1	0048 1956 5283 9172
	2	-0054 2335 5326 0322
	3	0062 0427 9940 3830
	4	-0072 3829 8068 4856
	5	0086 8588 9201 4708
	6	-0108 5736 1775 2932
	7	0144 7648 2731 2253
	8	-0217 1472 4095 6402
	9	0434 2944 8190 3244
13	0	0000 0000 0206 3047
	1	-0000 0002 7555 2341
	2	0000 0248 0157 8540
	3	-0001 3888 8888 6998
	4	0041 6666 6666 6472
	5	-0499 9999 9999 9993
	6	1591 5494 3091 8953
	7	0157 0796 3267 9490
	8	0785 3981 6339 7448
	9	not used
14	0-9	not used
15	0-9	not used

APPENDIX II

LIST OF CODES

A Sign		*A* Channel		*A* No.	Time	Operation	
positive	0	null	00	0	0	null	0
pos. abs.	1	σ	01	1	1	add	1
neg. abs.	2	*HRO*	02	2	2	multiply	2
negative	3	*LRO*	03	3	3	double-accuracy	3
				4	4	add	4
		SRO	05	5	5	sign choice	5
		16/15	06	6	6	identity check	6
		count zeros	07	7	7	*S/F*,*F/S* transfer	7
		αβ	08	8	8	check	8
				9	9	stop	9
		channel I_a	10				
		1	11	i	2*		
		2	12				
		select tape mech. 3	13	null	4*		
		4	14				
		5	15				
		6	16				
		7	17				
		8	18				
		channel a	20				
		b	21				
		c	22				
		d	23				
		e	24				
		f	25				
		g	26				
		h	27				
		p	28				
		q	29				
		r	30				
		s	31				
		t	32				
		u	33				
		v	34				
		w	35				
		x	36				
		y	37				
		z	38				
		γ	39				
		α	03*				
		β	04*				
		sw.	05*				
		zeros	15*				
code digit:	1 2		3 4 5 6 7 8		9 10 11 12		13 14 15 16
weight:	1 2		1 2 1 2 4 2*		1 2 4 2*		1 2 4 2*

B Sign		*B* Channel		*B* No.	Time	*C* Channel		*C* No.	Time
positive	0	null	00	0	0	null	00	0	0
pos. abs.	1		01	1	1			1	1
neg. abs.	2		02	2	2			2	2
negative	3		03	3	3			3	3
			04	4	4			4	4
			05	5	5			5	5
		constants	06	6	6			6	6
		See	07	7	7			7	7
		Appendix I	08	8	8			8	8
			09	9	9			9	9
			10						
			11	ι	2*			ι	2*
			12	function	3*				
			13	null	4*			null	4*
			14						
			15						
			16						
		powers of	17						
		10 See	18						
		Table 3.3	19						
		channel a	20			channel a	20		
		b	21			b	21		
		c	22			c	22		
		d	23			d	23		
		e	24			e	24		
		f	25			f	25		
		g	26			g	26		
		h	27			h	27		
		p	28			p	28		
		q	29			q	29		
		r	30			r	30		
		s	31			s	31		
		t	32			t	32		
		u	33			u	33		
		v	34			v	34		
		w	35			w	35		
		x	36			x	36		
		y	37			y	37		
		z	38			z	38		
		γ	39			γ	39		
		I_b	02*			I_a and I_b	02*		
		α	03*			function reg.	03*		
		β	04*			line no. reg.	04*		
		sw.	05*			cond. call reg.	05*		
		call	07*			δ reg.	06*		
		10^{ι}	12*			ι reg.	15*		
		Norm C. N.	13*			check-stop reg.	16*		
		cond. call	17*						
		XT	27*						
17 18		19 20 21 22 23 24		25 26 27 28		29 30 31 32 33 34		35 36 37 38	
1 2		1 2 1 2 4 2*		1 2 4 2*		1 2 1 2 4 2*		1 2 4 2*	

α and β storage channels, 24-26, 30,70,72,80-81,84,204-5,231

$\alpha\beta_\iota$ code, 84,205,231,299

A codes, 35,74,76,87,105,189,197, 202-5,233-34,243,255 (*see* Appendix II)
- $\alpha\beta_\iota$, 84,205,231,299
- ι number time, 82
- normalize, 88,112
- record zeros, 87,98,202-3,249, 255
- σ, 28,87-89,231
- shift read-out, 85
- special product read-outs, 85, 105-6,112
- tape read-record mechanism selection, 159,196
- transfer sign, 89

Adder, 26,28,29,90-91,96,204
- special use in high-accuracy addition, 99
- static add circuit, 93-95,102
- translation circuit, 94,95

Addition, 26,93-99,202,233
- double-accuracy, 29,95,97,98, 234,255
- exceeding machine capacity, 40, 97
- high-accuracy, 99

Aitken's process, 297

Alarm bell, 40,210,215

Alarms, 40
- check, stop, 206
- exceed-capacity, 97
- gear box lubrication system, 62
- identity check, 206
- printer, 177,184
- tape read-record, 155,164

Amplifiers
- beam power, 41,47
- class A, 41,46,58,59,74

Arc tangent, 21,31,125-27,250,284
- check, 129
- coding, 139-41
- constants required in computation, 126

Argument control printing, 170, 187-88

Arithmetic units, 10 (*see* Chapter IV)(*see also* Adder and Multiply unit)

Automatic rollback, 128,131-41, 274-75,277,284-85,291,299,300

B codes, 35,74-76,105,189,197, 202-5,233-34,243,255 (*see* Appendix II)
- call, 202
- channel I_b, 196
- conditional call, 202,207
- external transfer, 39,88,202-3, 205,207,231,258
- function number time, 130
- ι number time, 82
- normalize control number, 88, 112
- record zeros, 88,234,249
- shift number, 85
- transfer sign, 89

Bearings, 40,62,64

Binary number system, 8

Brackets, pole piece, 1,64,65
- constant storage, 67,69
- fast storage, 70,71
- slow storage, 6,78

Busses, 10
- *A*, 70,74,76,88,157
- *B*, 21,70,74-76,157
- *C*, 21,24,32,34,39,70,72,76,79, 82,87,88,91,130,156-57,202, 206-7,236,238,239,243

Cables, 41,44

Call command, 127,189,201,207,231, 290,298

Cam unit, 170,175,209-10,220,264

Carriage return, 176,184,187

Carry
- end-around, 26,96,98-99
- in adder, 93-94,96
- in multiply unit, 107-9,111
- round-off, 112

Cathode followers, 41,46-48,51,54, 58-60,74

C codes, 35,76,82,189,197,202,205, 243,255,258 (*see* Appendix II)
- check-stop, 206
- conditional call register, 207
- δ register, 79
- function sensing register, 130
- ι number time, 82
- ι register, 82
- line number register, 207
- tape read-record channels, 72, 159,196

Chassis, 11,13,41-43
- playback, 58

Chassis—continued
record, 61
socket numbers on, 46

Check-stop register, 32,189,206, 211,238,274

Clutch, magnetic, 142-43,154-55

Coding
function, 131-41
problem, 278-82,286-87,293-96, 301-6

Coding systems
2*421, 7,8
binary number system, 8
non-decimal digits, 35-36

Component containers, 41,45,50

Conditional call command, 39,189, 207,231,272,275,285,290-91, 298-300

Conditional call register, 39, 206-7,213,231,236,275,284

Constant channels, 21,31,32,67,70, 119-27,205 (*see* Appendix I)

Constant register channel, 21, 72-74,204-5,231,268

Control pyramids, 189,196-97, 201-5,217-18

Cosine, 21,31,123-25,127,250
check, 129
coding, 137-39
constants required in computation, 124-25

Crystal rectifier gates, 45,49-51, 74-76,109

Cycling pulses, 66,68
in adder, 96
in line number register, 207, 213
in multiplier, 106-111
in sequencing, 38,189,192,194

δ register, 24,26,77-82,84,236-37, 298

Decimal point, 8,29,235,268
in function computation, 127-28
in multiplier, 102,104-6,112
in printing, 170,176,270

Digit time, 9
ring, 54-56,65

Diodes
crystal rectifier, 49,50
in playback unit, 58
vacuum-tube, 46

Dipole, magnetic, 1,4,7,33

Division, 32,38,127,234
check, 128

Division—continued
coding, 131-32

Drum phase, 10,105,107,203-5

Elementary functions (*see* Functions)

End-around carry, 26,28,96,98,99

End-of-page control, 187-88,270-71

Examples of coding, 273,283,288, 297 (*see also* Functions)

Exponential function, 21,31,122-23, 127,250
check, 128
coding, 135-37
constants required in computation, 122
evaluation of complex, 283-87

External transfer command, 39,88, 202-3,205,207,231,258

Fast-slow storage transfers, 24-26,30,70,80-82,298

Fast storage channels, 9,10,21,30, 70,74,76,82,231,273-306

Feed command, 235-36,252

Feed read-out command, 235-36, 276-77

Floor plan, 11,12

Functions, 21,29,31-32,67,272,285 (*see* Chapter V)
coding of, 126-41
decimal point, 127-28
on instructional tape preparation table, 233,250,255
starting lines on drum, 233

Function sensing register, 32,67, 119-20,129-30,189,205

γ channel, 24-25,70,72

Gates (*see* Crystal rectifier gates and Vacuum-tube gates)

Gear box, 40,62,63

I_a and I_b channels (*see* Tape read-record channels)

Identity check circuit, 32,40,166, 206,212,234,277

Input-output devices (*see* Chapter VI) (*see also* Tape read-record mechanisms, Numerical tape preparation table, and Printers)

Instructional storage drum, 11,14, 32,36-39,189-92,198,233-34, 258,267 (*see* Storage drums)
recording on, 218-23

Instructional tape preparation table, 9,11,20,28-32,36-39, 94,97-98,105,273
(*see* Chapter VIII)
function computation, 126-27, 233,250,255
instructional tape, 36,38,252, 258,261-62,265
keyboard, 229-43
lights, 229,244,246,253,265
line number accumulator, 237, 244-45,253,258
operation, 264-66
organization, 243-58
printing circuits, 262-63,265
record and playback circuits, 258-60
rings, 256-60
step switches, 237,243-44, 247-48,250-58,260-63
strip switches, 229,236-37, 239-42,246
timing diagram, 257,260

Instructional tapes, 36,38,215, 223-26,229,252,258,261-62, 265,267

Interpolation, 24-26,30,250,297

i number time, 30,82,231

Inverters, 46-47,58-60

i register, 30,82-83,189,236, 290-91,298
α,β selection, 84,205
computation of arc tangent, 126
computation of exponential, 122
normalization, 233
number time selection, 82,204, 231
shifting operation, 31,67,85, 234

Keyboard of instructional tape preparation table, 36-37
fast storage registers, 231
functions, 232-33
general description, 229-31
operations, 233-34
special *A* registers, 231
special *B* registers, 234-36
special *C* registers, 236-43

Lights, 39
check,stop, 32,206,239,269
conditional call, 207
for instructional tape preparation table, 229,244,246, 253,265
for numerical tape preparation table, 33,145-46,151,266
for printer, 178,270
for tape read-record system, 155-56

Lights—continued
identity check, 206
last-mechanism-used, 159,162, 167
line number and code, 189,193, 199-200,210,212,215,218,223, 228,269
stop and *go*, 196,269
tape circuits reset, 223,267
transfer channel, 26,91-93,268
tubes used to operate, 41,47
wait, 196

Lights pushbutton, 208,218

Line number, 36-39,189,192-93,197, 199,200
in function computation, 233, 251
lights, 189,193,199-200,210, 212,215,218,223,228,269

Line number accumulator, 237, 244-46,253,258,265

Line number register, 38-39,127, 189,197-98,200,207,213-14, 231,236,291

Logarithm, 21,31-32,118-19,121,127, 250
check, 128
coding, 134-35
constants required in computation, 121-22

Magnetic cam unit, 148,150,260-61

Magnetic clutch, 142-43,154-55

Magnetic dipole, 1,4,7,33

Magnetic recording technique, 1,4, 7,33,142

Magnetic tape, 11,33
arrangement of sequence digits on, 36,38,252,258
instructional, 261-62,265
numerical, 142-45,147,167, 266-67,274,276-77

Main control panel, 27,39,40,92, 196-97,207-28,267

Mark II Calculator, 41,170

Matching circuit, 38,189-90, 193-94,197-201,207,211,215, 218,221,228

Milne's method, 288

Motors, 40,264
instructional storage drum, 190
numerical storage drum, 1,62, 190
tape read-record mechanism, 155,266-68

Multiplicand (*MC*) channel, 101

Multiplication, 28-29,100-12,197, 234-35,250
double-accuracy, 29
shift, 85
timing, 107

Multiplier (*MP*) channel, 101,109

Multiply unit, 8,100-12
decimal point, 102,104-6
input translation circuit, 101-2
MC channel, 101
MP channel, 101,109
PP channel, 102,104
special product read-outs, 85, 105-6,112,235
static add circuit, 102
static multiply circuit, 101-3

Newton-Raphson formula, 115

Non-decimal digits, 35-36,76,263

Non-zero sensing circuit, 176, 184-85

Normalization, 31,112-14, 233-34
control numbers, 88,233,234
in function computation, 130

Number time, 9,10,55
control pyramid, 202-3
gates to transfer channels, 86, 88,130,204,209
t, 30,82,231
multiply, 107
pushbutton selection, 72,74
ring, 65-67
selected by function sensing register, 119-20,130,205

Numerical storage drums, 1-3,8-11, 21,24,62,65
(*see also* Storage drums)
motor, 62,190,264
phase of, 105,107,203,205

Numerical tape preparation table, 11,19,33-34,142-51,264,266-67

Operate pushbutton, 208,215,217-18

Operation-codes, 35,112,189,197, 255 (*see* Appendix II)
associated with adder, 94-95, 249,255
check, stop, 206,251
control pyramid, 201-2
for double-accuracy addition, 98-99
for identity check, 206
for multiplication, 105-6,250
for sign-choice operation, 99
for slow-fast storage transfer, 81

Partial product (*PP*) channel, 102, 104

Pillow block and stub shafts, 64

Playback unit, 4,21,61,74
circuit, 57-59
for *MP* channel, 109
for slow storage channels, 80
printer, 178,180
sequence, 190-94,200
tape read-record system, 165
wave forms, 4,7,58-59

Pole piece brackets, 1,64,65
constant storage, 67,69
fast storage, 70,71
slow storage, 6,78

Pole pieces
α,β,γ channels, 70,72
A and B transfer channels, 87
C transfer channel, 91
channels I_a and I_b, 157
constant channels, 21,67,70
construction, 1,4,5,61,65
fast storage channels, 9,10,21, 49,70,74-76,78
instructional storage drum, 36, 191,220
multiplier channels, 29,101, 104-5
slow storage channels, 24,61,78

Pole pieces, tape, 33,142
instructional tape preparation table, 253
printer, 178

Power supplies, 11,18,177,264,269

Printer, 34,167-88
alarms, 177,184
argument control, 170,187-88
check circuits, 183-84
control panel, 11,16,170,174
end-of-page control, 187-88
for instructional tape preparation table, 262-63,265
lights, 178,270
non-zero sensing circuit, 176, 184-85
operation, 269-71
playback unit, 178,180
plugging, 170,182,188,270
pole pieces, 178
print and check relays, 180-84
ring, 180
step switches, 170,176,180-88
tape mechanism, 171
timing diagram, 179
toggle switches, 170,176-78,180, 187,270-71

Problem preparation
(*see* Chapter X)

Product read-out, 105-6,109,112,235
in shift operation, 85,233

Product sign, 29,89,91,99,109-12

Pulse playback unit, 59-61,65

Pushbuttons
- *A,B,C* (lights) 91
- *check* (lights) 91-92
- *number-time*, 72,74
- numerical tape preparation table, 144-46
- *off* (lights) 93
- printer, 178
- *read-in*, 72
- *read-out*, 92
- sequencing, 196,208-28
- tape read-record system, 155

Read last line to lights pushbutton, 208,212,269

Read specified line to lights pushbutton, 208,215,269

Reciprocal, 21,31,32,115-17 (*see also* Division)
- first approximation constants, 120

Reciprocal square root, 21,31, 117-18,127,250
- check, 128
- coding, 132-33
- first approximation constants, 121

Recording technique, magnetic, 1, 4,7,33,142

Record match relays, 198,208,218, 220,228

Record unit, 7,21
- α and β, 80-81
- circuit, 60-61
- instructional storage drum, 221
- slow storage, 80-81
- tape read-record system, 162
- wave forms, 7,58

Record zeros
- on *A* transfer channel, 87,98, 202-3,249
- on *B* transfer channel, 88,234, 249

Regeneration
- on channels I_a and I_b, 34,157, 239
- on transfer channels, 11,87-88, 96

Registers
- check-stop, 32,189,206,211,238
- conditional call, 39,206-7,213, 236,275,284
- constant storage, 31-32,120-27, Appendix I
- δ, 24,77-82,84,236,298
- function sensing, 32,119-20, 129-30,189,205

Registers—continued
- i, 30-31,82-85,122,126,189, 204-5,231,233-34,236,290-91, 298
- line number, 38,39,127,189, 197-98,200,207,213-14,236,291
- select line number, 197,269
- start line number, 197,200, 211-12,215,268-69
- stop line number, 197,200-1,268

Relays, 41,47
- controlling line number and code lights, 193,210,212,218, 223
- instructional storage channel selection, 197-98,201,221
- line number and code storage, 38,197,200,210,212,215,223, 225,227-28
- print and check, 180-84
- record match, 198,208,218,220, 228
- slow storage selection, 79-81
- start match, 198,201,211
- stop match, 198,200-1

Repeat toggle switch, 208-9,217-18

Rerun procedures, 276,291,300

Reset sequence tape input circuits pushbutton, 223

Rings
- circuits, 52-56
- digit-time, 54-56,65
- instructional tape playback circuit, 225-26,228
- instructional tape preparation table, 256-60
- multiply unit, 106
- number-time, 65-67
- numerical tape preparation table, 149,151
- odd-even, 66-67
- printer, 180
- tape read-record system, 158, 161-63,165-67
- zero-counting circuit, 112-14

Round-off, 30,112

Runge-Kutta method, 289-90

Run next line and stop pushbutton, 208,211-12,269

Run specified line and stop pushbutton, 208,212,269

σ command, 28,87-89,231

Select line number register, 197, 269

Sequence code digits, 201 (*see A, B, C,* and Operation-codes)

Sequence code digits—continued
allocation of, 35
drum storage of, 36
transfer from tape to drum, 38, 267

Sequencing, 35-40,272
(*see* Chapter VII)
cycling circuits, 192-98,200-1
cycling pulses, 38,189
manual controls, 189,207-28
playback units, 190-94,200
wait signal, 156,189,196-97,201

Serial numbers on instructional storage drum, 36,38,189-90, 192

Shifting operation, 31,67,76,85, 112,234
in function computation, 120, 128
product read-out, 85,233

Sign
A and *B*, 89
in check-stop register, 206
in conditional call register, 39,206-7
in double-accuracy addition, 98
in shifting operation, 31,85
in slow-fast storage transfer, 80-81
sum, 28,96-97,99

Sign-choice operation, 30,91,95, 99,234,284

Slow-fast storage transfers, 24-26,30,70,80-82,237,298

Slow storage channels, 21,24, 79-81,237,297-99

Start controls, 39,40,201

Start line number register, 197, 200,211-12,215,268-69

Starting lines for function routines, 127,233

Starting procedures, 267-68,276, 291,300

Start match relays, 198,211

Start next line pushbutton, 208, 210-12

Start specified line pushbutton, 208,211,268

Static add circuit, 93-95
used in multiplier, 102

Static multiply circuit, 101-3

Step switches
associated with read-out lights, 92-93
for constant register channel, 72-73

Step switches—continued
instructional tape preparation table, 237,243-44,247-48, 250-58,260-63
numerical tape preparation table, 145-47,149,151
printer, 170,176,180-88
tube used to operate, 41,47
used with main control panel, 208-28

Stop controls, 39,40,269
code, 206,276-77,292,300
on specified line, 200-1,215, 268
pushbutton, 215

Stop line number register, 197, 200,201,268

Stop match relays, 198,200-1

Stop specified line toggle switch, 198,200-1,215,268-69

Storage channels, 67,81
(*see* Appendix I)
α and β, 24-26,30,70,72,80-81, 84,204-5,231
constant, 21,31,32,67,70, 120-27,205
constant register, 21,72-74, 204-5,231,268
fast, 9,10,21,30,70,74,76,82, 231,273-306
for shift operation, 85,234
γ, 24,25,70,72
instructional, 36,190,192, 197-98
location of, 65
slow, 21,24,79-81,237,297-99
tape read-record (I_a and I_b) 34,72,156-62,165-67,236,239

Storage drums, 1,7,11,65,190-91, 264,266 (*see* Instructional storage drum and Numerical storage drums)

Storage system, 1-3,8-10,21-26, 117,119,272 (*see* Chapter III)

Strip switches
constant register, 21,72-73
decimal point, 104-6
instructional tape preparation table, 229,236-37,239-42,246
numerical tape preparation table, 144,148

Subtraction, 26,28,94,233

Sum sign, 28,96,97,99

Switches, toggle
numerical tape preparation table, 145,151
printer, 170,176-78,180,187, 270-71
repeat, 208-9,217-8

Switches, toggle—continued
stop specified line, 198,200-1, 215,268-69
tape read-record mechanisms, 154-55,162,164,267-68
tape to drum, continuous, 208-9,228,267

Tape, magnetic, 11,33
arrangement of sequence digits on, 36,38,252,258
instructional, 261-62,265
numerical, 142-45,147,167, 266-67,274,276-77

Tape read-record channels (I_a,I_b) 72,156-62,165-67,204-5,236, 239

Tape read-record mechanisms, 11, 15,33-34,40,142,151-56,159, 267-68

Tape read-record operation, 34, 159-67,239
check, 163-64,167
feed read-out, 235-36

Tape read-record system
busy signal, 162,196
channels I_a and I_b, 72,156-62, 165-67,204-5,236,239
failure, 40
last-mechanism-used light, 162, 167
lights, 155-56
playback and record units, 162, 165
pushbuttons, 155
rings, 158,161-63,165-67
timing diagrams, 161,168-69
toggle switches, 154-55,162, 164,267-68

Tape to drum and lights, last line pushbutton, 208,228

Tape to drum and lights, next line pushbutton, 208,225,228,267

Tape to drum, continuous toggle switch, 208-9,228,267

Tape to lights, last line push-button, 208,228

Tape to lights, next line push-button, 208,223,228

Tchebychef polynomials, 116-17, 121,123-26

Timing diagrams
instructional tape preparation table, 257,260
multiply unit, 109,111
numerical tape preparation table, 151
printer, 179

Timing diagrams—continued
slow-fast storage transfer, 81
tape read, 168-69
tape record, 161

Timing pulses
on drums, 7,55,59,65-66,107, 191-93,221-23
on instructional tapes, 38
on numerical tape preparation table, 145,148
on numerical tapes, 33,142

Toggle switches
numerical tape preparation table, 145,151
printer, 170,176-78,180,187, 270-71
repeat, 208-9,217-18
stop specified line, 198,200-1, 215,268-69
tape read-record mechanisms, 154-55,162,164,267-68
tape to drum, continuous, 208-9,228,267

Tolerance check, 40,128-29

Transfer channels
A, 10,11,21,26,28,32,34,86,87, 89,92,94,96,98,99,201,202, 204,231
B, 10,11,21,26,28,32,34,88,89, 91,92,94,98,99,112,114,201-2, 205,207,231
C, 11,26,28,29,90,91,94,96,98, 99,105,109-12,201
lights, 26,91-93,268
number-time gates, 86,88,130, 204,209
recording of zeros on, 87,88, 98,202,234
regeneration, 11,87,88,96
sign storage circuits, 28, 88-91,95

Transfer command, 239,243

Transfer line of coding to drum and lights pushbutton, 208, 218,223,228,269

Transfer signs, 28,88-91,94-95, 201,231,234

Translation circuit
for multiplier inputs, 101-2
in adder, 94-95

Trigger pairs
$A1$-$A8$,$B1$-$B8$, 158,164,167
$A9$,$B9$, 158,165-66
check-stop register, 206
δ register, 77-81
function sensing register, 130
i register, 82-84
j,k,ℓ, 160,162-63,165-67
line number register, 207,214

Trigger pairs—continued
  matching circuit, 200
  pentode, 46,51-53,58-59
  triode, 50-51

Typewriters, 11,17,34-35,170,173, 176-77,184,187,263,270

Vacuum-tube gates, 46,48-49,54,60

Vacuum tubes
  method of operation, 46
  types used, 41,46,47
  voltages for operation, 264

Wave forms, 4,7,58-59

Zero-counting circuit, 88,112-14, 233,234