RESTRICTED DISTRIBUTION
JUNE 1961

UNIVAC® Solid-State Computer

Manual 1

PROCESSOR

BASIC SYSTEM and 9200-Word Drum Option

Remington Rand Univac®
DIVISION OF SPERRY RAND CORPORATION

315 PARK AVENUE SOUTH
NEW YORK 10, N.Y.

Published by
UNIVAC ENGINEERING CENTER, PHILADELPHIA
Post Office Box 500
Blue Bell, Pa.

RESTRICTED DISTRIBUTION

JUNE 1961

Part 1

80-COLUMN SYSTEM

CONTENTS

Heading **Title** **Page**

SECTION 3. DESCRIPTION OF LOGIC CIRCUITS

ILLUSTRATIONS

TABLES

Section 1
INTRODUCTION

Manual 1, part 1 explains the fundamentals of computer logic and the application of these fundamentals to the theory of operation of the Processor in the UNIVAC® Solid-State Computer (USSC) system. The operation of the Processor in the basic 80-column-card system is covered in this part of the manual. This information applies to the following Processor models:

Model 131 —80-column system, convertible to magnetic tape

Model 131A—80-column system, convertible to magnetic tape and Randex*

Model 159 —80-column system, convertible to magnetic tape, Randex, and either 5000- or 9200-word drum

Model 139 —80-column system, convertible to magnetic tape, Randex, and either 5000- or 9200-word drum, and CPP

Models 131 and 131A have a 5000-word memory which is described under headings 3-87 to 3-109. The 5000-word memory and its expanded version (9200 word) for models 159 and 139 are described under headings 3-110 to 3-119.

Additions or changes to the logic or circuitry of the Processor described in this manual which are peculiar to a specific model of the USSC system are described in the Theory and Maintenance manual which covers that model.

It is not the purpose of this manual to detail all the logic operations carried out by the Processor. Section 2, however, presents the basic concepts of logic circuitry. Sections 3 and 4 apply the knowledge of logic circuitry to detailed explanations of computer circuits and some of the more complex functions of the Processor. Section 5 explains the operation of all the types of instructions, except those which control the input-output devices. The operation of magnetic amplifiers and other circuit elements are presented in section 6.

Note that all references to drawings in this manual refer to logic drawings in manual 2. References to figures indicate illustrations within this manual.

1-1. FUNCTIONAL DESCRIPTION

Figure 1-1 shows a typical functional arrangement of the system. Variations of this arrangement are possible through program control and are covered in manual 8. Table 1-1 lists the major characteristics of the UNIVAC Solid-State Computer system. Complete information on the input-output units can be found in manual 3.

The USSC system is composed of the Processor, the card reader, the read-punch unit, and the high-speed printer.

The first step in processing is to store the program for the problem. A program, compiled by a programmer, is a sequence of instructions punched on tabulating cards. The instructions on the program cards are read by the card reader and transferred to the computer-storage unit. The program automatically specifies and controls the operations required to solve a given problem.

Punched tabulating cards, bearing input information to be processed, then are placed in the card reader (figure 1-1). The card reader reads or senses the information represented by punched holes on the cards. The information is transferred from the card reader to the Processor where the program stored previously in the magnetic storage drum controls the processing and computation of the input information.

The computed results or data processed in the Processor are sent, as output, to the printer and to the read-punch unit. The output information may be printed on paper by the printer or punched on other tabulating cards in the read-punch unit or both. The cards placed in the input bin of the read-punch unit can be either prepunched with information or blank. In either case the input card is read at a read station before being punched with computer output at the punch station. The card is read again for checking purposes at a second read station and transferred to one of two output stackers.

1-2. PROCESSOR

The Processor consists of the computing and processing circuitry, the storage drum, the operator's control panel and keyboard, and the system power supplies. The storage device is a cylindrical, magnetically coated drum with

*Trademark of Sperry Rand Corporation.

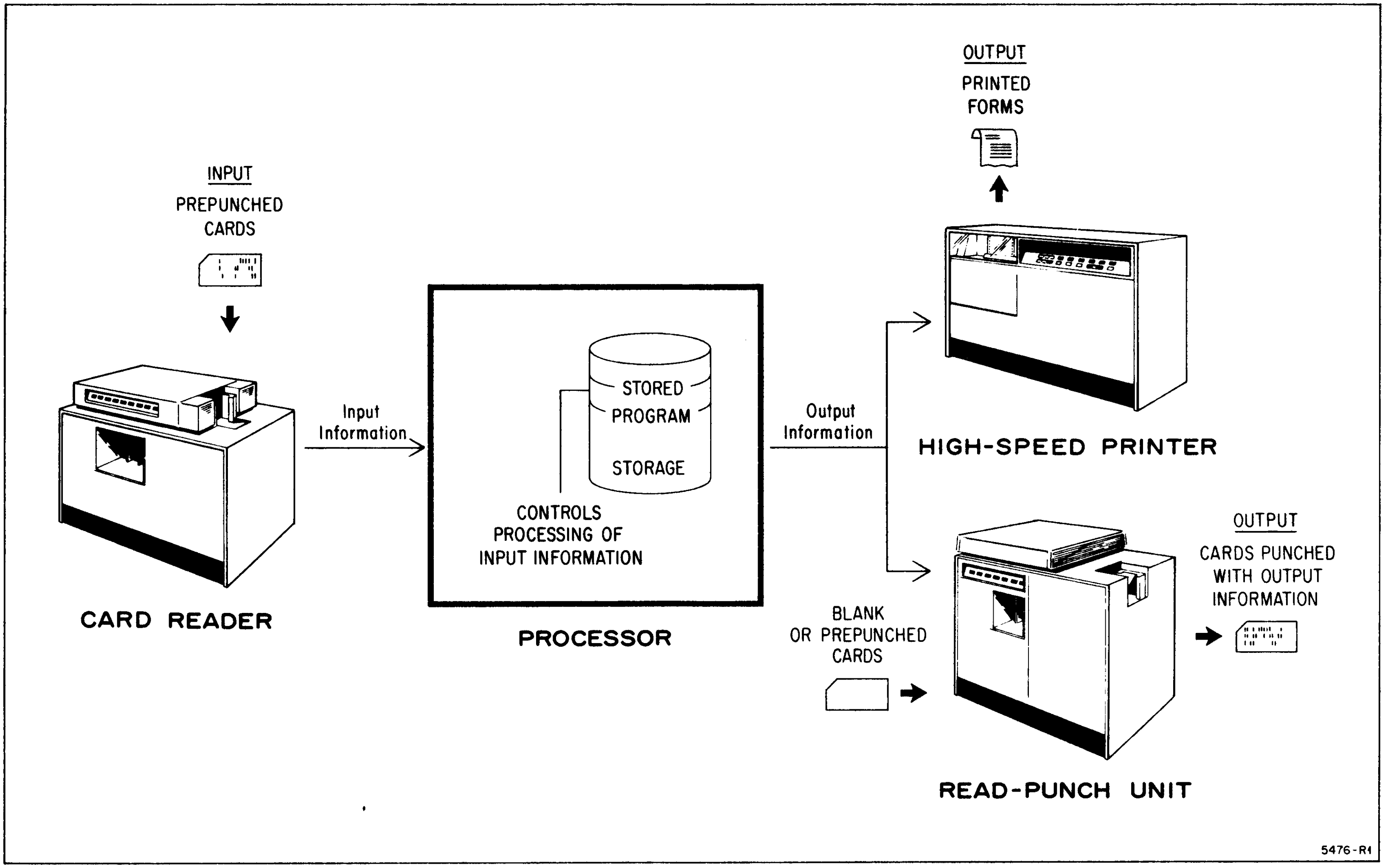


Figure 1-1. Typical Arrangement of UNIVAC Solid-State Computer

Table 1-1. Characteristics of the USSC System

Input-output characteristics are listed in manual 3. A repertory of 34 instructions is available for the programmer. This table is concerned with the arithmetic, comparison, transfer, translate, shift, input-output, and miscellaneous instructions. See section 2 for complete instruction list.

Arithmetic Speeds	
Addition	85 microseconds; addition of two ten-digit numbers
Subtraction	85 microseconds; subtraction of two ten-digit numbers
Multiplication	119 microseconds (minimum) 1785 microseconds (maximum)
Division	425 microseconds (minimum) 1955 microseconds (maximum)
Computer Words	
Data word	12 digit positions (ten digit positions, one sign position, and a space between words)
Instruction word	12 digit positions (ten digit positions, one sign position, and a space between words)
Information Flow	
Bits	In parallel
Digits	In series
Words	Serial-parallel
Speed	707,000 digits per second
Codes	
USS-6	Four-bit, binary coded biquinary with two zone bits
Card	12-bit 80-column card code
Storage	
Total capacity	5000 or 9200 words (main storage)
Capacity per band	200 words
Total number of bands	25 (5000-word drum); 46 (9200-word drum)
Standard-access bands	20 for 5000-word drum; 38 for 9200-word drum
Capacity	4000 words for 5000-word drum; 8600 words for 9200-word drum
Access time	3400 microseconds, maximum (200 word times)
Fast-access bands	5 (4 heads per track) for 5000-word drum; 3 (4 heads per track) for 9200-word drum
Capacity	1000 words (5000-word storage); 1600 words (9200-word storage)
Access time	850 microseconds, maximum (50 word times)
Speed	17,670 rpm
Timing band	Provides channel addresses and synchronizes generation of timing pulses
Sprocket tracks (2)	Provide basic half-pulse time frequency for clock
High-speed printer buffer tracks (2)	Store 13 words, which constitute a printed line, from main storage
Card-reader buffer tracks (4)	Store 16 words (two input cards)
Read-punch unit buffer tracks (4)	Store 24 words (two input cards and one output card)
Input Keyboard	
Keys	0 through 9
Non-numeric	Typed by simultaneous operation of numeric keys
Signs	Plus and minus keys
Control	Control keys
Timing	
Half-pulse time	0.707 microsecond (time for a signal to be transferred through a magnetic core)
Pulse time	1.414 microseconds (time for passage of one digit)
Word time	17 microseconds (time for passage of one word)

a capacity of either 5000 or 9200 computer words. In addition to this 5000- or 9200-word capacity, which is known as main storage, other areas of the drum are set aside for timing purposes and input-output buffer storage. The buffer-storage areas store information coming from or going to the input-output units. Buffer storage enables the slower input-output units to keep pace with the faster speeds of the Processor and also allows the units of the system to operate in parallel.

All the computing and processing circuitry explained in this manual is compactly contained on printed wiring boards, known as packages, within the Processor cabinet. Arithmetic, control, and processing operations take place within the magnetic-amplifier, transistor, diode, and miscellaneous circuitry on these packages. The clock circuits, which synchronize the operation of the magnetic-amplifier components, and most of the system power supplies also are located in the Processor cabinet. (See figure 1-2.)

Three of the four logical units necessary to a data-processing system are contained in the Processor. These units (figure 1-2) are: the control unit, the storage unit, and the arithmetic unit. A general description of each of the three units follows; detailed descriptions of the units can be found in section 3.

1-3. CONTROL UNIT. This unit controls all the operations of the system including the input-output unit. The major components of the control unit are: register C, the static register, the instruction decoder, the function encoder, and the operator's control panel.

Complete instruction words read from storage are stored in register C. Parts of these words are compared with

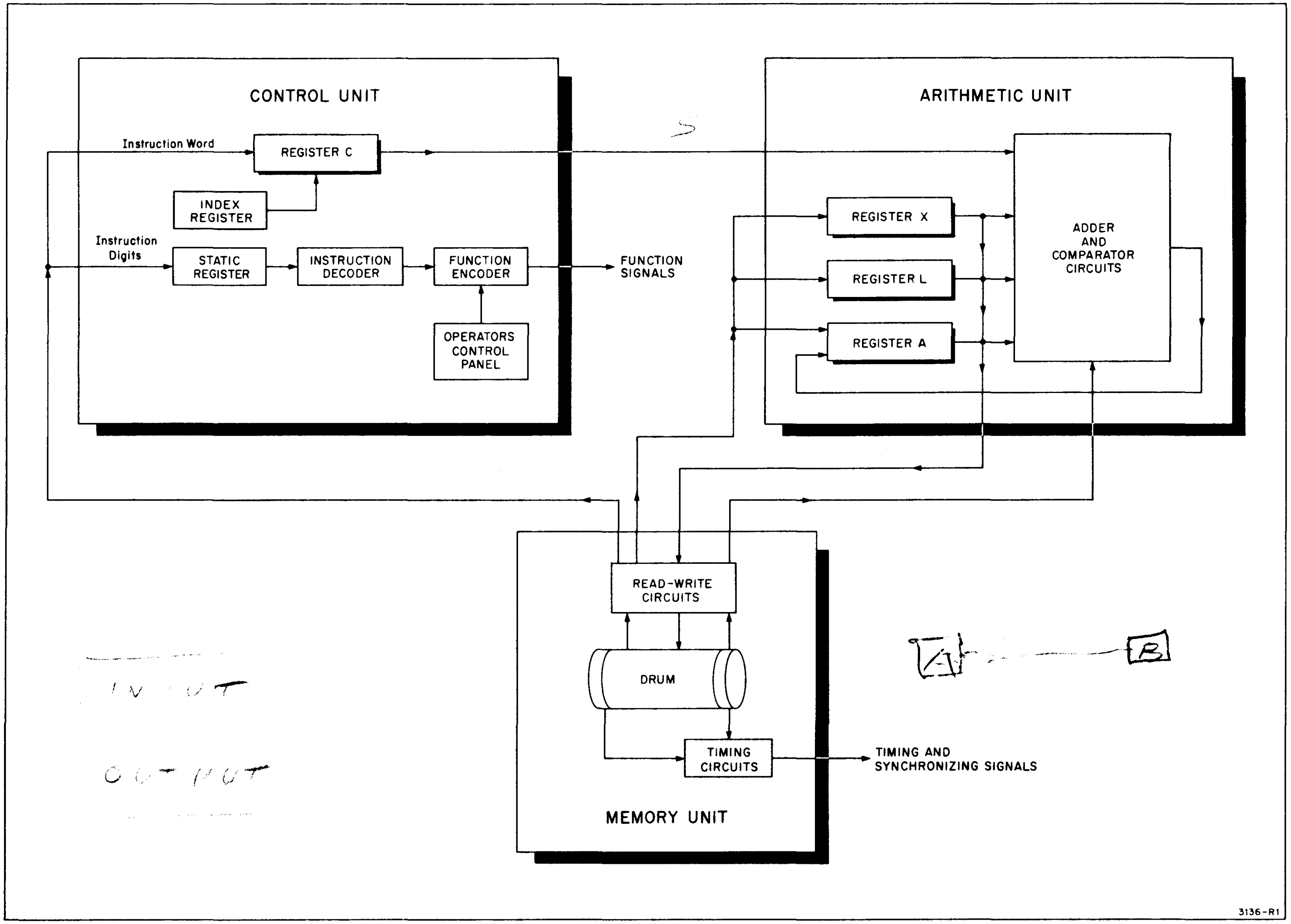


Figure 1-2. General Block Diagram of Processor Units

storage addresses in the adder and comparator circuits in order to locate stored information or other instruction words. The two instruction-code digits of the instruction word stored in register C are also stored in the static register. These digits are interpreted by the instruction decoder and converted into a function signal. This function signal is converted into a number of other function signals by the function encoder. The function signals control various computer circuits which cause the stored instruction to be executed.

The operator's control panel contains pushbutton switches and indicators which enable the operator to control manually the automatic operation of the system. Abnormal conditions in any of the four units also are indicated at the operator's panel. A keyboard on the panel enables the operator to type information into the Processor or to alter the stored program.

1-4. STORAGE UNIT. The main component of the storage unit is the magnetic-storage drum. Instructions or data are stored magnetically on the drum. Instructions are read from the drum to control calculations or processing; data is read from the drum to be operated on by the instruction. The read-write circuits of the drum control reading from or writing onto the drum. Permanently recorded signals on the drum go to the timing circuits which synchronize and time all computer operations.

1-5. ARITHMETIC UNIT. Although the arithmetic unit has many components, the most important components are the three arithmetic registers and the adder and comparator circuits. All arithmetic and comparison operations are performed in the arithmetic unit. Each of the three registers, A, L, and X, stores temporarily the ten digits of a word, which is usually the operand in an arithmetic instruction. The operands are operated on in the adder and comparator circuits, and the results are returned to one of the three registers.

1-6. CARD READER

The card reader reads information from punched input cards and transfers the information to the Processor at a maximum rate of 450 cards per minute. The input cards are placed in the input bin. From the input bin, cards are automatically transported through the read stations to the output stackers. Each card is read by brush sensing at two read stations. The second read station enables the programmer to check the accuracy of the information read at the first read station. After the two readings, the cards are transported to one of three output stackers. The program determines which stacker is used.

1-7. READ-PUNCH UNIT

The read-punch unit also reads information from input cards, but it is not a normal method of input to the system. The cards inserted into the read-punch-input bin are usually blank cards or cards punched with a small amount of information. Under the control of the program, the card is read by brush sensing at the first read station and is sent to the punch station where processed or computed information is punched into the card. The card moves to the second read station where the original information on the card and the information added by punching are checked. The card is then transferred to one of two output stackers as determined by the program. The read-punch unit reads and punches cards at a rate of 150 per minute.

1-8. PRINTER

The information computed by the Processor can be printed by the printer on many types of continuous forms at a speed of 600 lines per minute. Each line of print can contain a maximum of 130 characters. The number of spaces between lines is specified by the program. The printer contains 65 printwheels, each of which contains all the characters available for printing. All the characters to be printed on any one line are printed during the time necessary for one complete revolution of the printwheels. Under control of the program, the paper is spaced before or after the printing of each line.

1-9. COMPUTER LANGUAGE

The purpose of this section is to familiarize the reader with the basic language of the computer.

1-10. COMPUTER WORDS

Information is processed by the computer in units known as computer words. A computer word is a group of digits constituting the smallest unit of information which can be processed or stored. Each computer word consists of ten digits and a sign and is presented in a space 12 digits long. The 12th position is used as the spacing between words and is included as part of the word. Each digit of a word consists of six bits. (Refer to heading 1-14.) A bit is a binary character (either 0 or 1) which, when combined with other bits in a certain order, can represent a numeric digit. The bits of a digit are transferred in parallel, and the digits are transferred in series. To facilitate identification of digits in a computer word, the method shown here is used throughout this manual:

In this method, the 12 digit positions of a computer word are designated p0 through p11. The digits of a word are transferred throughout the computer with the least-

Computer Word

Position Number	p11	p10	p9	p8	p7	p6	p5	p4	p3	p2	p1	p0
Digit	SBW	10	9	8	7	6	5	4	3	2	1	0

significant digit (LSD) first, which is the reason for placing digit 1 or p1 at the right end of the word and the most-significant digit (MSD), digit 10 or p10, at the left end of the word.

Computer words are divided into two groups, data words and instruction words. Data words contain information to be processed; instruction words contain instructions and addresses of words to be used in the instruction. Instruction words, therefore, control processing and computations; data words are part of the information processed or computed. The difference between data and instruction words is a programming consideration only. The computer does not differentiate between the two.

1-11. DATA WORD. A data word consists of information digits to be processed or computed. The 12 digit positions of a data word contain ten digits, an algebraic sign, and a space between words (SBW). In the example given, the p0 digit position contains the sign; the p1 through p10 digit positions contain the numeric digits (even though they may be 0 digits), and the p11 digit position is the SBW and contains a 0 digit.

1-12. INSTRUCTION WORD. An instruction word consists of ten numeric digits, a space between words, and includes a sign if index registers are part of the system. The digits of an instruction word are divided into three groups as shown:

Instruction Word

SBW	instruction code	m address	c address	Sign
p11	p10 p9	p8 p7 p6 p5	p4 p3 p2 p1	p0

Digits p10 and p9 are the instruction-code digits. Every instruction consists of two digits. For example, when the programmer wishes to program an add instruction, he places the digits 7 and 0 in the p10 and p9 digit position of the instruction word.

Digits p8 through p5 comprise the m address, which is usually the storage address of the operand to be used in executing the instruction. In an add instruction, the operand must be read from the m address in Processor storage before addition can take place. The programmer inserts the numerical address of the word which is to be read in the m-address position of the instruction word. Digits p4 through p1 make up the c (control) address, which is the address of the next instruction. To construct a continuous program, in which one instruction follows another in the desired order, each instruction word must contain the address in Processor storage of the next instruction word. The address of this next instruction word is known as the c address.

1-13. SYSTEM CODES

The computer employs two types of coding for numeric (and alphabetic) information processed within the system, computer (USS-6) code and card code. The punched tabulating cards used as the main method of input and output of the system are punched in card code, in which digits are represented by 12 bits. The information in card code on the input cards is read from the cards under control of the program and transferred to the translator circuits of the Processor. These circuits translate the 12-bit card code into six-bit USS-6 code for internal use in the Processor. When processing of the input information is complete, the six-bit USS-6 code is translated back into 12-bit card code which can be punched on the output tabulating cards.

1-14. USS-6 CODE (MACHINE AND PRINTER). The USS-6 code consists of six bits; four bits represent the numeric value of a character, and the two zone bits differentiate between numerics and non-numerics and between groups of letters and symbols. (Refer to table 1-2). The USS-6 code is also known as machine code because it is the code used in internal machine operations. The USS-6 code is also the printer code. Each character is composed of six bits as shown below:

	Numeric Bits	
Zone Bits	Binary	Quinary
6 5	4	3 2 1

The numeric bits are weighted, from left to right, 5 4 2 and 1, so that they represent the numeric digits 0 through 9. For example, a 1 bit in the bit-4 position represents the digit 5. The numeric bits are further subdivided into binary and quinary bits because the logic circuits of the system differentiate between them. Bits 1, 2, and 3 are referred to as the quinary part of the numeric bits because the pattern repeats at 5. Bit 4 is the binary bit because it is a 0 for the numeric digits 0 through 4, and it is a 1 for the numeric digits 5 through 9.

The zone bits designate whether the four numeric bits are for numeric characters, alphabetic characters, or symbols. Table 1-2 shows that both zone bits for all numeric digits are 0's. For example, a numeric combination of 0001 with 00 zone bits represents the digit 1. The same numeric combination with 01 zone bits represents an A, with 10 zone bits a J, and with 11 zone bits a /.

1-15. 80-COLUMN CARD CODE. The 80-column card stores information in the form of punched holes across 80 vertical columns of a card. A punched hole indicates a 1 bit while the absence of a punched hole in a hole position indicates an 0 bit. Each of the 80 columns contains 12 hole positions or rows, labelled Y, X, and 0 through 9 from top to bottom. A numeric character is represented by a single hole in the

Table 1-2. USS-6 Machine and Printer Code

Zone				Numeric
00	*01*	*10*	*11*	
)	&			0111
space	:	*	%	0110
— (minus)	. (period)	$	, (comma)	0101
0*			+	0000
1†	A	J	/	0001
2	B	K	S	0010
3	C	L	T	0011
4	D	M	U	0100
5	E	N	V	1000
6	F	O	W	1001
7	G	P	X	1010
8	H	Q	Y	1011
9	I	R	Z	1100
' (apost)	‡			1111
;				1110
(				1101

*Represents plus when used in the sign position, p0.
†Represents minus when used in the sign position, p0.

row (0 through 9) which corresponds to a character. (Refer to table 1-3.) For example, a punched hole in row 3 represents a 3. An alphabetic character or symbol is represented by a number of holes in one row.

Table 1-3. 80-Column Card Code

Rows YXO				Rows 1-9
000	*100*	*010*	*001*	*1 2 3 4 5 6 7 8 9*
1	A	J	/	1 0 0 0 0 0 0 0 0
2	B	K	S	0 1 0 0 0 0 0 0 0
3	C	L	T	0 0 1 0 0 0 0 0 0
4	D	M	U	0 0 0 1 0 0 0 0 0
5	E	N	V	0 0 0 0 1 0 0 0 0
6	F	O	W	0 0 0 0 0 1 0 0 0
7	G	P	X	0 0 0 0 0 0 1 0 0
8	H	Q	Y	0 0 0 0 0 0 0 1 0
9	I	R	Z	0 0 0 0 0 0 0 0 1
space	&	— (minus)	0	0 0 0 0 0 0 0 0 0
‡	. (period)	$	, (comma)	0 0 1 0 0 0 0 1 0
+	:	*	%	0 0 0 1 0 0 0 1 0
)				1 0 0 1 0 0 0 1 0
' (apost)				0 0 0 1 0 1 0 1 0
;				0 0 0 1 1 0 0 1 0
(				0 0 1 0 1 0 0 1 0

The 80-column card is divided into eight card words of ten columns each. (See figure 1-3.) Each word is further subdivided into three parts of four rows each, known as the unprimed, primed, and double-primed words. The J and I designations are input words read by the reader and read-punch unit, respectively, and the O words are output words.

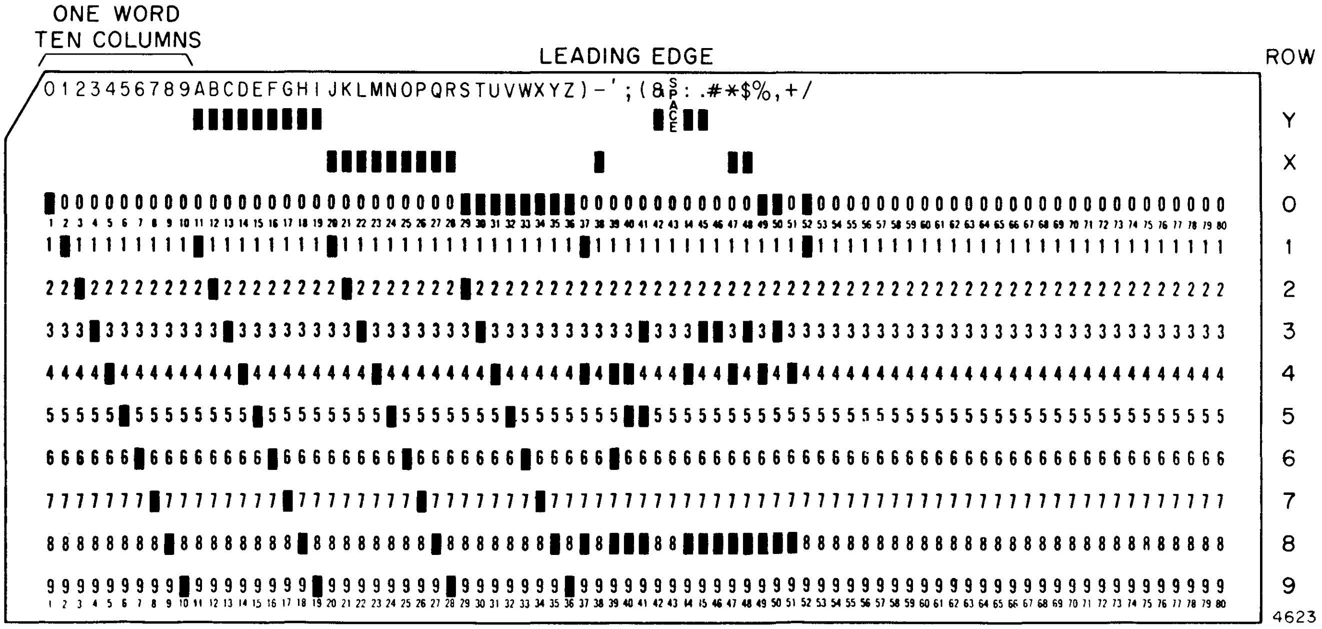


Figure 1-3. 80-Column Card Divided Into Words

Section 2
LOGIC CIRCUITRY

2-1. INTRODUCTION

This section gives the necessary background to understand Processor operations that are described in subsequent sections. It also discusses information to be found in the logic drawings of manual 2.

Whenever possible, the description is related to the logic drawings by examples of actual circuits. In certain instances, however, simplified versions of typical logical circuits are described as an introduction to the actual circuits shown on the drawings.

2-2. GENERAL

The logic drawings represent the Processor and synchronizer circuitry in symbolic form. Because most of the circuitry consists of magnetic amplifiers and diodes, the logic drawings contain, principally, diode and magnetic-amplifier symbols connected by lines. Signals are indicated on the drawings by reference designations or numbers.

This section explains the meaning of the logic symbols and describes how the elements that the symbols represent form the basic Processor circuits. It describes types of signals and the way they are represented on the drawings. In addition, information is given concerning the method by which timing signals are generated and used to synchronize operations throughout the Processor.

Only logical functions of the circuit elements are described here. The electronic theory and operation of magnetic amplifiers and other circuit elements are presented in section 6.

2-3. SIGNAL POLARITY

Information is represented in circuitry by high or low signals. In this section, high- and low-signal conditions are distinguished from each other for illustrative purposes as shown in figure 2-1a.

A high signal is a positive voltage. A low signal is a 0-voltage condition. A high signal may be thought of as representing a pulse condition and a low signal as representing a no-pulse condition. Actually, a low signal is not negative, as suggested in figure 2-1a. It is shown as negative only to avoid the confusion that might result if the 0-voltage condition were represented by a straight line.

2-4. DIODE CIRCUITS

The basic logical element of the circuitry is the diode circuit. A diode circuit is symbolized on logic drawings as a segment of a circle (figure 2-1b). A diode circuit performs the logical functions of a gate (figure 2-1c), or a buffer (figure 2-1b, 2). When used as a connecting diode (figure 2-1b, 3), it performs no logical function.

Diodes transmit current in one direction but oppose current flow in the opposite direction, thereby preventing any signal from affecting the source of other signals.

2-5. GATES

A gate is a circuit that produces a prescribed output signal only when every one of its input lines is in a prescribed state. A gate is represented by the diode symbol with a dot in the center.

A gate produces a low-output signal only when the signal on every one of its input lines is low. (See figure 2-1c.) If

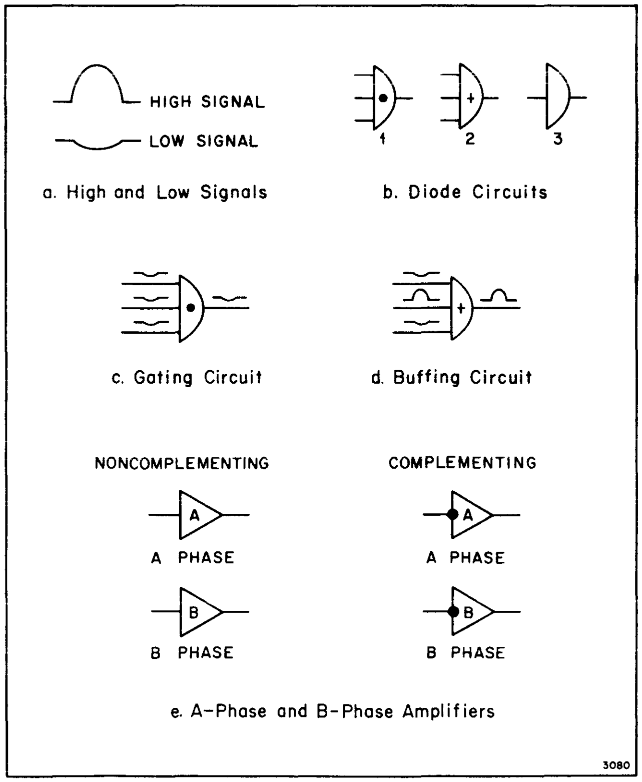


Figure 2-1. Basic Logical-Circuit Elements

one or more of the input signals is high, the output signal is high. High signals cannot be used to perform the gating function since the same high output is produced whether one or all of the input signals are high.

2-6. BUFFERS

A buffer is a circuit that produces a prescribed output signal when any one of its input lines is in a prescribed state. A buffer is represented by the diode symbol with a plus sign in the center. The buffer output is high when any one of its input lines is high. (See figure 2-1d.)

2-7. DIFFERENCES BETWEEN GATES AND BUFFERS

The difference between a gate and a buffer is a difference of logical function. The same circuit that is a gate for low signals is also a buffer for high signals. The necessary definitions are made on the logic drawing. A diode circuit is represented as a gate if its essential function is to perform a gating function; otherwise, it is represented as a buffer or a connecting diode.

2-8. CONNECTING DIODE

A connecting diode (figure 2-1b, 3) is represented by a diode symbol without a dot or plus sign in its center. A connecting diode has only one input line and one output line and has no logical significance other than to prevent the interaction of signals. Connecting diodes are shown on the logic drawings to distinguish the magnetic amplifiers that they connect from another type of magnetic amplifier which does not require a connecting diode. (Refer to heading 2-43.)

2-9. MAGNETIC AMPLIFIERS

The basic symbol for the magnetic amplifier is a triangle. The point of the triangle indicates the direction of signal flow. Additional notation shows amplifiers as belonging to one of two electronic categories, and one of two logical categories.

2-10. ELECTRONIC DIFFERENCES

A central sine-wave clock circuit supplies power pulses to all the magnetic amplifiers to retime and reshape the input signals. Magnetic amplifiers can be classed electronically in two groups: those that receive A-phase power pulses, and those that receive B-phase power pulses.

The generation of power pulses is synchronized with the revolving storage drum by sprocket tracks on the drum (figure 2-2). Each sprocket track contains 2400 permanently recorded pulses. These pulses are read continually from the drum at the rate of one every 1.414 microseconds, which is the basic pulse rate of the computer. The sprocket pulses are fed to the sine-wave clock which generates an A- and B-phase power pulse for each sprocket pulse it receives. These power pulses differ by half a pulse time and are 180 degrees out of phase. The letter A or B inside the triangle indicates the phase of the power pulse that drives the amplifier.

All A-phase amplifiers in the computer receive an A-phase power pulse simultaneously. One half-pulse time later, all B-phase amplifiers receive a B-phase power pulse. Amplifiers connected in series must alternate in phase so that an A-phase amplifier always precedes and follows a B-phase

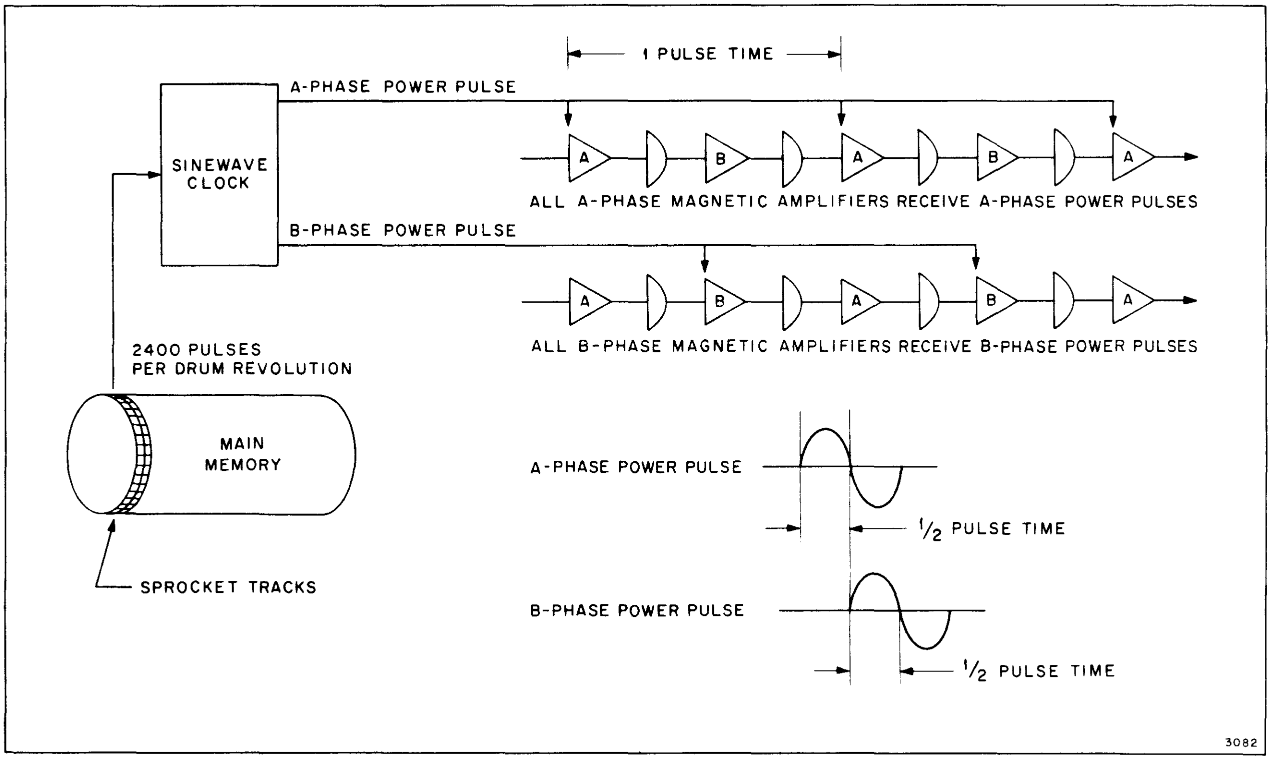


Figure 2-2. A-Phase and B-Phase Power Pulses

amplifier. Each power pulse has a positive half cycle and a negative half cycle (figure 2-2). During the half-pulse time that the power pulse is negative, the magnetic amplifier receives an input signal. For example, an A-phase amplifier receives an input signal during the half-pulse time that its A-phase power pulse is negative. During the next half-pulse time, the A-phase amplifier delivers an output signal to the B-phase amplifier next in line at the time the B-phase amplifier receives the negative half of its B-phase power pulse. This sequence of operations entails a delay of one half-pulse time in the passage of a pulse through an amplifier or one full-pulse time through both an A-phase and B-phase amplifier. The operation of the magnetic-amplifier circuits is discussed in detail in section 6.

2-11. LOGICAL DIFFERENCE

Logically there are two types of magnetic amplifiers, complementers and noncomplementers. A noncomplementer produces an output signal of the same polarity as the input signal, whereas the complementer inverts the input signal. It is distinguished from a noncomplementer on the logic drawings by a dot added to the basic amplifier symbol (figure 2-1e).

2-12. DECODING AND ENCODING MATRIXES

To simplify their presentation, it is often convenient to represent large gating and buffing arrays as decoding and encoding matrixes. Representing a gating or buffing array in this form does not change its logical significance.

2-13. DECODING MATRIX

A decoding matrix is an array of gates that convert a combination of several signals into one signal. A typical decoding matrix is the instruction decoder shown in drawing 1-3A. The same circuit is redrawn in figure 2-3a to illustrate that it is a standard gating circuit. The dots at the intersections of the lines of the instruction decoder represent diodes in the actual circuitry, and each of the vertical lines represents a single gating circuit. For example, the vertical line labelled 26(B) represents a gating circuit with input signals STR1, $\overline{\text{STR2}}$, STR4, $\overline{\text{STR5}}$, STR6, $\overline{\text{STR7}}$, and STR8. The output of the circuit after inversion by an A-phase complementer is the 6A signal. When all the inputs to the gate are low, the complemented output signal 6A is high; if one or more of the input signals is high, the output signal 6A is low. Each of the vertical lines in the decoding matrix of drawing 1-3A represents a gating circuit such as shown in figure 2-3a.

2-14. ENCODING MATRIX

Whereas a decoding matrix is an array of gates that convert a combination of several signals into one signal, an encoding matrix is an array of buffers that convert one signal into a combination of several signals. A typical encoding matrix is the function encoder in drawing 1-4A. As in the decoding matrix, the dots at the intersection of the lines represent diodes in the actual circuitry.

The 6A signal generated by the instruction decoder, as previously described, is one of the many inputs to the function encoder. The 6A input line is connected by diodes to the lines which produce output signals 6, 55+, 67, 77, and 82A.

To illustrate how function-signal outputs are generated, part of the function encoder is redrawn as an array of buffers in figure 2-3b. The high 6A signal from the instruction decoder generates low function signals 6, 67, 77, and high function signals 55+ and 82A. When signal 6A is high, the other inputs to the buffers are low.

Other A-phase signals from the instruction decoder also can generate some of the function signals that are generated by the 6A signal. For example, a high 9A signal generates function signals 67, 77, and 82A. However, it does not generate signals 6 and 55+. In the actual circuitry, A-phase signals, other than those shown in figure 2-3b, can generate one or more of the function signals 6, 55+, 67, 77, and 82A. To simplify the example, these signals are not shown in the figure.

2-15. COMPUTER TIMING

All Processor operations are precisely synchronized. The movement of information within the Processor or between

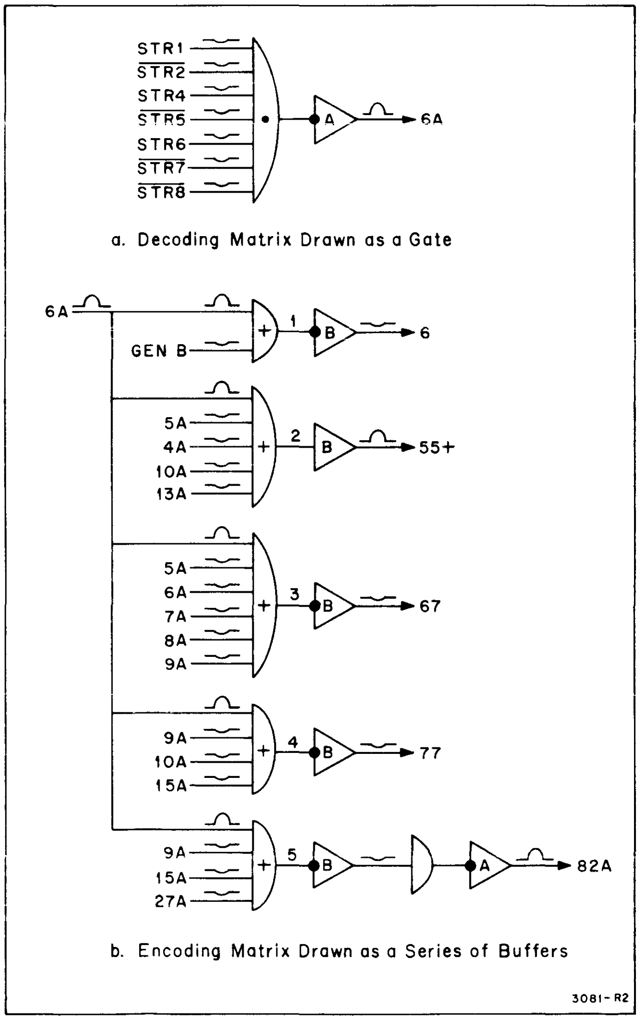


Figure 2-3. Decoding and Encoding Matrixes

the Processor and input-output units is synchronized by timing signals generated by the cycling unit.

2-16. PULSE TIME

The basic unit of computer timing is the pulse time. This is the time required for a pulse to pass a given point in the circuitry. A pulse is delayed one half-pulse time in passing through a magnetic amplifier or one full-pulse time in passing through both an A-phase and a B-phase magnetic amplifier. Because the four signals that represent the four bits of a decimal digit are transmitted in parallel (heading 2-35), one pulse time is also the time necessary for a digit to pass a given point in the circuitry.

2-17. WORD TIME

The basic unit of Processor information is the 12-digit word, which requires 12 pulse times to pass a point in the circuitry. These 12 pulse-time intervals, or one word time, are designated t0 to t11 (figure 2-4). Each of the 12 pulse-time intervals corresponds to the time necessary for a pulse to pass an A-phase and B-phase amplifier in the circuitry, and each pulse-time interval is divided into an A and B phase. Consequently, there are 24 distinct time intervals, each of one half-pulse time duration, to each word time. The timing signals that synchronize operations of the Processor are based on the 24 time intervals of a word time. During every word time, the cycling unit generates timing signals for each of the 24 time intervals.

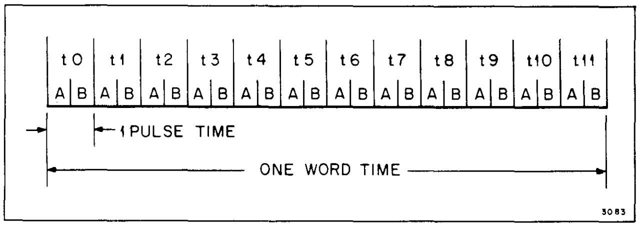


Figure 2-4. Twenty-four Time Intervals of Word Time

In addition to the main-storage bands, the storage drum has a timing band that contains 200 locations. The timing-band locations are, however, not addressable by the programmer. They contain permanently recorded pulses that perform various synchronizing functions, one of which is to synchronize generation of the timing signals with the rotation rate of the drum. For this purpose, a special code combination (1101) is recorded in one of the 12 digit spaces in each of the 200 locations of the timing band. This special combination, read from the timing band once each word time of a drum revolution, is fed to the cycling unit to generate timing signals.

2-18. TIMING SIGNALS

The cycling unit (drawing 1-18A) consists of a string of magnetic amplifiers connected by diodes. This circuit constitutes a one-word time delay; that is, a word time is necessary for a pulse to travel through the entire circuit. Such a pulse must pass through 24 magnetic amplifiers, each of which causes a delay of one half-pulse time.

The special code combination that initiates generation of the timing signals is fed into gate 102 on the TS1, $\overline{\text{TS2}}$, TS3, and TS4 lines. (The notation TS1, $\overline{\text{TS2}}$ is explained under heading 2-41.) Once each word time, the code combination 1101 from the timing band produces low signals on all the input lines of gate 102. (See drawing 1-18A.) Thus, gating takes place, and once each word time the A-phase amplifier following gate 102 generates a high signal (t7A+). This timing signal is high for only a half-pulse time; at all other time intervals it is low. The plus (+) sign following the signal designation indicates that it is a high signal only during the half-pulse time interval t7A.

The high t7A+ signal is an input to a B-phase complementer and a B-phase noncomplementer so that a half-pulse time later, the B-phase timing signals (t7B+ and t7B—) of the t7 time interval are generated. (The + or — following each t7B signal indicates that it is a high or low signal, respectively, only during the half-pulse time interval t7B.) As the signal generated by the A-phase amplifier at the output of gate 102 progresses through the cycling unit, it generates A- and B-phase timing signals for each of the 24 half-pulse time intervals.

In addition to generating A- and B-phase signals for each time interval, the cycling unit generates complemented and noncomplemented versions of the A- and B-phase signals during most of the 24 time intervals. (See drawing 1-16A.) For example, during the t0 time interval, the cycling unit generates signals t0A—, t0A+, t0B+, and t0B—.

In certain instances, timing signals, such as the t0B+ signal, are generated on two output lines. Logically, the two signal outputs are identical. They come from two sources to satisfy an engineering requirement for additional driving power because the signals are needed to synchronize an unusually high number of operations throughout the Processor. To distinguish between the two signal outputs, one is written primed (t′0B+) and the other is written unprimed (t0B+). If index registers are included, additional timing signals are generated.

2-19. RELATION BETWEEN TIMING SIGNALS AND WORD TIME

One of the 200 locations of the timing band is shown in detail in the upper-left hand corner of figure 2-5. The permanently recorded pulses in this location are read by the timing-band read heads in one word time. Consequently, the special code combination (1101), in the digit position corresponding to the t6 time interval, is read once every word time or 200 times per drum revolution. Because the code combination 1101 is located in the t6 digit space, the timing signals for the t7A time interval are the first signals generated by the cycling unit after the 1101 combination is read. Then, for one word time, the timing signals for the remaining time intervals are generated by the cycling unit. At the end of one word time, the combination 1101 in the next timing-band location begins the cycle again.

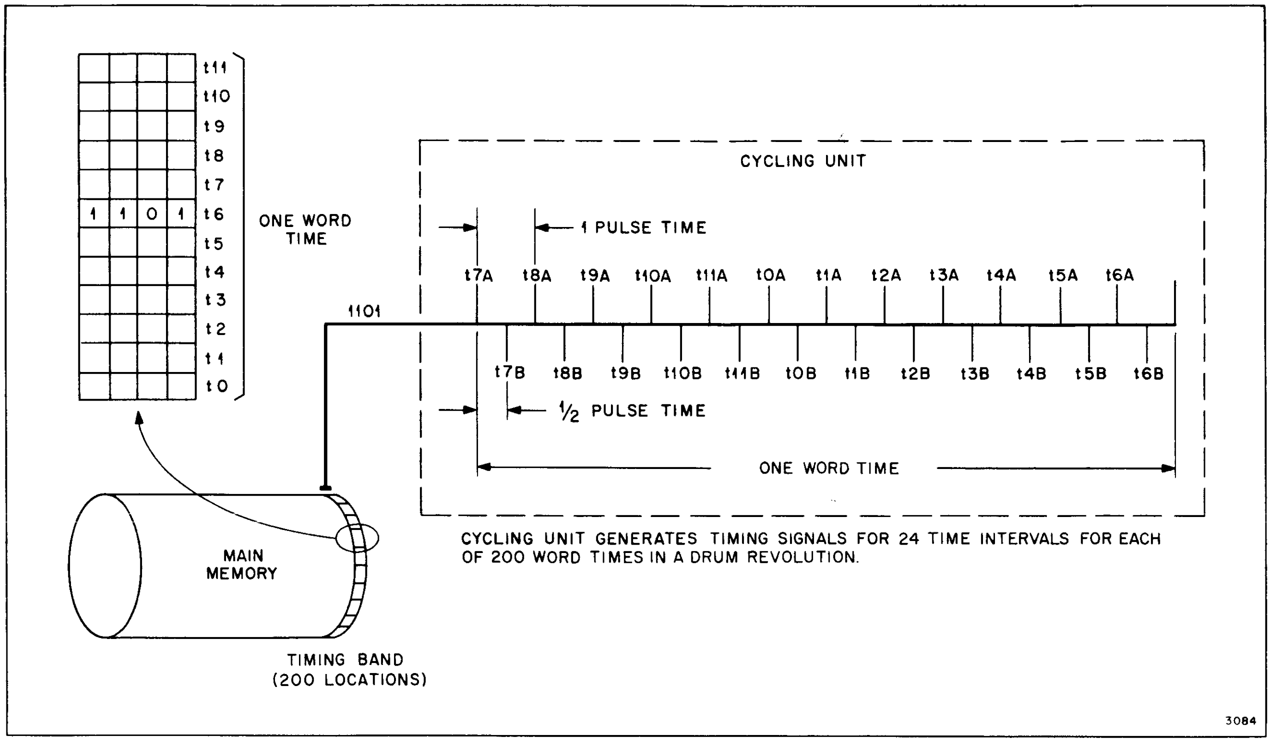

t11
t10
t9
t8
t7
1 1 0 1 t6
t5
t4
t3
t2
t1
t0
ONE WORD TIME
1101
CYCLING UNIT
1 PULSE TIME
t7A t8A t9A t10A t11A t0A t1A t2A t3A t4A t5A t6A
t7B t8B t9B t10B t11B t0B t1B t2B t3B t4B t5B t6B
½ PULSE TIME
ONE WORD TIME
CYCLING UNIT GENERATES TIMING SIGNALS FOR 24 TIME INTERVALS FOR EACH OF 200 WORD TIMES IN A DRUM REVOLUTION.
MAIN MEMORY
TIMING BAND (200 LOCATIONS)
3084

Figure 2-5. Relation Between Timing Signals and Word Time

2-20. SIGNAL NOTATION

In addition to the timing signals already described, there are three types of signals in the Processor: function, control, and information signals. Function and control signals control the execution of instructions in accordance with the program. Information signals represent information processed by the Processor. The notations used to designate these three types of signals distinguish them from each other and from the timing signals. The use of the function, control, and information signals is described in detail in section 4. Only the notations used to designate these signals are described here.

2-21. FUNCTION SIGNALS

Most function signals are designated FS1, FS6, FS77, etc. Some function signals have a plus (+) sign following the number to indicate that they are high signals during the execution time of instructions which have them as function signals. At all other times, they are low signals. The opposite is true for signals without a plus sign. Although most of the function signals are B-phase signals generated by B-phase amplifiers, there are a few A-phase function signals generated by A-phase amplifiers. These are distinguished by an A placed after the signal designation (FS14A, FS20A). On the logic drawings, however, all function signals are placed inside circles (figure 2-6).

2-22. CONTROL SIGNALS

A control signal is designated by an alphabetic notation that is often an abbreviation of the name of the signal. As examples, EP is the abbreviation for the ending-pulse signal, CT for conditional transfer, TS for time selection, and CP for complement.

2-23. INFORMATION SIGNALS

Information signals transferred within the Processor are designated by an alphanumeric notation. A letter or letters indicate the source of the signal, and a number indicates the bit position of the signals. For example, a six-bit character leaves register A on the A1, A2, A3, and A4 lines for numeric bits, and leaves register X on the X1 and X2 lines, for the two zone bits. The letter identifies the register-output lines while the number identifies the bit position.

2-24. LOGICAL FUNCTIONS

In the logical circuitry, the gating function is usually performed by a diode circuit and an amplifier in series. A typical gating circuit is shown in figure 2-6. The output signal from the A-phase complementer is high only when all the input signals to the gate are low.

The gating circuit in figure 2-6 is typical of many gating circuits within the Processor in that the input signals include a timing signal plus one or more signals of another type. In addition to timing signal t11B—, this circuit has as inputs, FS1 and two control signals, OF and TS. The meaning of these function and control signals is explained in section 4. The polarity of timing signal t11B— is known for any time interval of any word time. The signal is high except for the half-pulse time interval t11B, when it is low. The polarities of the function signal and the two

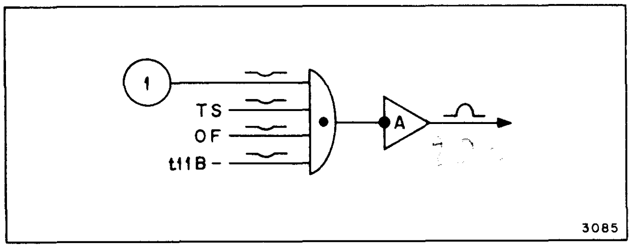


Figure 2-6. Typical Gating Circuit

control signals do not change in accordance with any regular time sequence but depend on the program being processed.

In descriptions of gating circuits in this manual the following three expressions are often used: making a gate permissive, alerting a gate, and blocking a gate. The meaning of each of these expressions is described under headings 2-25 through 2-27, with reference to figure 2-6.

2-25. MAKING A GATE PERMISSIVE

Most of the gating functions in the computer are performed during a definite time interval. The gating circuit in figure 2-6, for example, can perform the gating function only during the half-pulse time t11B when this signal is low. During any other time interval, the high timing signal prevents a gating function regardless of the polarities of FS1 and control signals TS and OF. If these signals are low at t11B, a gating action takes place and the gate is said to be made permissive. The term making a gate permissive means that a circuit which is designated as a gate operates as a gate at some definite time.

2-26. ALERTING A GATE

Function signal 1 is normally high, but goes low during the execution time of an instruction for which it is one of the function signals. Assuming that such an instruction is being executed, FS1 goes low at t0B, the beginning of some word time. The timing signal t11B—, however, does not go low until 11 pulse times later so that a gating action cannot take place before t11B. Whether gating takes place depends on the polarities of the OF and TS signals at t11B. When FS1 goes low, it is said to alert the gate. In this manual, the expression alerting a gate means that a signal input to a gate goes low one or more pulse times earlier than the time interval of a possible gating action and remains low at least until the interval ends.

2-27. BLOCKING A GATE

The gating circuit in figure 2-6 can perform the gating function only during the time interval t11B, when the t11B— signal is low. At any other time interval, the high timing signal blocks the gate, preventing it from performing the gating function regardless of the polarities of the other input signals. The term blocking a gate is used in this manual to mean that a signal input to a gate is high and, therefore, blocks any gating action.

2-28. FLIP-FLOPS

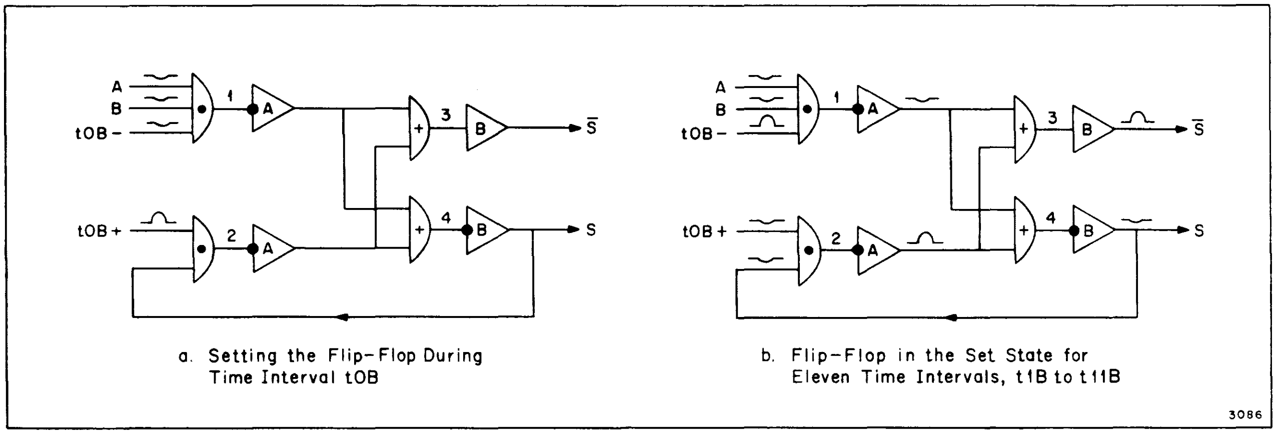
Gates and buffers are used in a circuit known as a flip-flop, which is used for a variety of purposes. The following paragraphs explain the basic flip-flop and describe a few of its uses.

2-29. THE BASIC FLIP-FLOP

The flip-flop is a storage element that can maintain either of two stable states, set or restored. It remains in one state until an input signal changes it to the other state. A simple flip-flop is shown in the set state in figure 2-7b and in the restored state in figure 2-8b. The notations used to designate the two outputs of this particular flip-flop, S and $\overline{S}$, mean that the two signals are always opposite in polarity; that is, if $\overline{S}$ is low, S is high, and if $\overline{S}$ is high, S is low. The unbarred and barred notation is used to identify the outputs of many of the flip-flops. For example, the complement (CP) flip-flop has two output signals, CP and $\overline{CP}$; the conditional-transfer flip-flop has outputs CT and $\overline{CT}$. The use of barred and unbarred notations for circuits other than flip-flops is discussed under heading 2-41.

2-30. SETTING THE FLIP-FLOP

The flip-flop in figure 2-7a is set when gate 1, the set gate of the flip-flop, is made permissive by low signals



a. Setting the Flip-Flop During Time Interval t0B

b. Flip-Flop in the Set State for Eleven Time Intervals, t1B to t11B

Figure 2-7. Setting a Typical Flip-Flop

A and B at time interval t0B. The timing signal t0B— to gate 1 is low and the timing signal t0B+ to gate 2 is high only during the time interval t0B. One pulse time later, signal S goes low and signal $\overline{S}$ goes high. The flip-flop is kept in the set state by the circulating low S signal which is gated with the normally low timing signal into gate 2. Gate 2 is made permissive so that a high signal into buffers 3 and 4 keeps signals S low and $\overline{S}$ high for at least the remainder of the word time. (See figure 2-7b.)

At the end of the word time, the flip-flop can remain in the set state or it can be restored. If signals A and B are still low at the next t0B time interval, gate 1 is made permissive again and the flip-flop remains set for another word time. During this same t0B time interval, the t0B+ signal into gate 2 (the restore gate of the flip-flop) does not restore the flip-flop because the high signal into buffers 3 and 4 overrides the low signal from the restore gate into these same buffers. This action illustrates an important property of flip-flops in the circuitry. If a flip-flop receives a set and a restore signal during the same time interval, the set signal always takes precedence.

2-31. RESTORING THE FLIP-FLOP

To restore the flip-flop, set gate 1 must not be permissive at t0B; that is, signals A or B, or both, must be high at t0B. The timing signal into restore gate 2 goes high at t0B. Figure 2-8a shows how a flip-flop is restored by the timing signal into restore gate 2. One pulse time later, at t1B, signal S is high and signal $\overline{S}$ is low. Figure 2-8b shows the flip-flop held in the restored state for one word time by the circulating high S signal into gate 2 and the normally high timing signal into gate 1. At the next t0B time interval, the flip-flop will be set only if both signals A and B into gate 1 are low. Otherwise, it will remain in the restored state.

2-32. THE SET AND THE RESTORED STATES

The set and restored states are easily distinguished. The flip-flop in figure 2-7b is in the set state; S is low and $\overline{S}$ is high. The same flip-flop is shown in the restored state in figure 2-8b; S is high and $\overline{S}$ is low. The state of this flip-flop, therefore, indicates the polarity of the output signals. This is true of most flip-flops which have output signals with unbarred and barred notations. For example, the polarities of the two output signals of the complement flip-flops are indicated by the state of the flip-flop. When the CP flip-flop is set, CP is low and $\overline{CP}$ is high; when it is restored, CP is high and $\overline{CP}$ is low. In most flip-flops that have a barred and unbarred output, the flip-flop is set when the unbarred signal is low and restored when the barred signal is low.

2-33. THE READ FLIP-FLOP

The flip-flop shown in figures 2-7 and 2-8 does not exist in the logical circuitry, although its logic is the same as that of the flip-flops in the system. A description of the main-storage-read flip-flop on drawing 1-21A, for example, shows the similarity between it and the typical flip-flop described under headings 2-29 through 2-32. The main-storage-read flip-flop controls the reading of information from main storage.

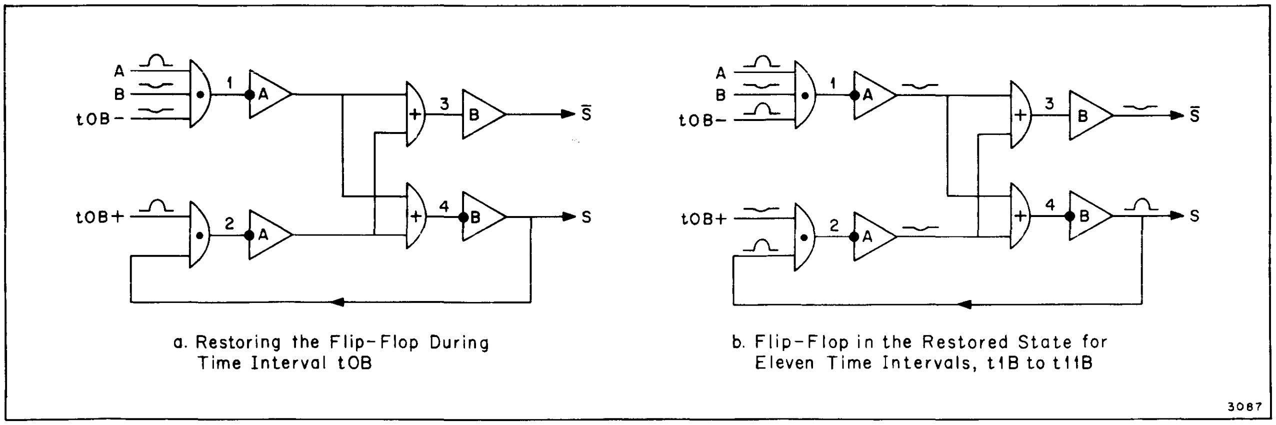
The read flip-flop resembles the flip-flop in figures 2-7 and 2-8 in that its state can be changed once every word time and it stays in one state for at least one word time. The read flip-flop, however, has several set gates, only one of which operates at any time. Another important difference is that there is only one output signal, RD, from the read flip-flop.

If any set gate has all-low input signals at t8B, the read flip-flop is set for one word time; RD goes low at t9B and remains low for at least one word time. At the end of this word time, the t8B+ signal into restore gate 43 restores the flip-flop unless a set gate has all-low input signals. If the flip-flop is restored, it remains in this state until a later t8 time interval, when it can be set by all-low input signals on a set gate.

2-34. COUNTERS

The read flip-flop illustrates one use of a single flip-flop. Two or more flip-flops can be used to perform other logical functions such as counting.

A counter is a circuit that counts the number of times a repeated operation is performed. For example, a row



a. Restoring the Flip-Flop During Time Interval tOB

b. Flip-Flop in the Restored State for Eleven Time Intervals, t1B to t11B

Figure 2-8. Restoring a Typical Flip-Flop

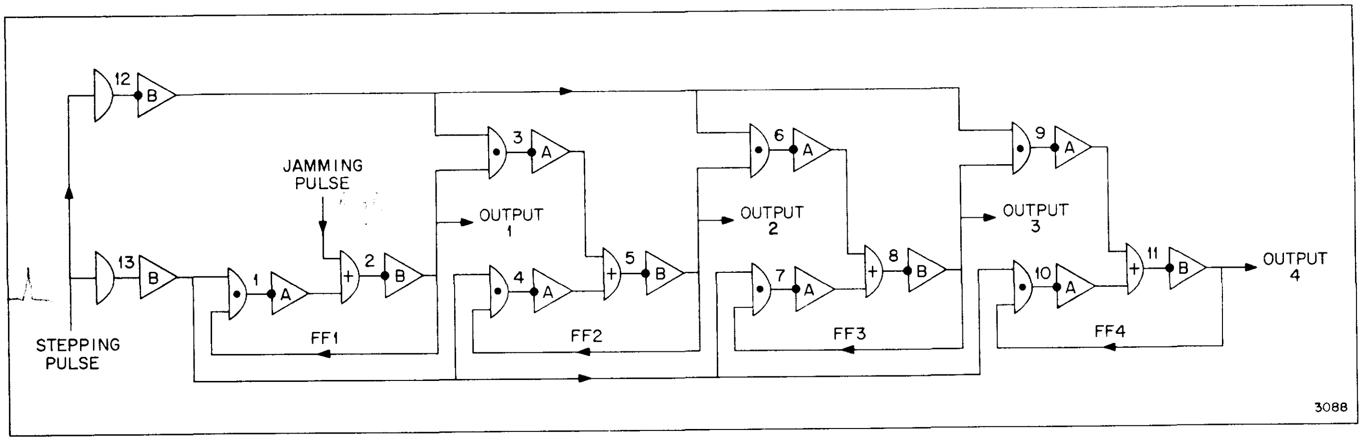


Figure 2-9. Typical Counter Circuit

counter counts the rows of a card being read by the card reader. Because there are 12 rows on a card, such a component must be able to count from 1 to 12.

A typical counter is shown in figure 2-9. It can count from 1 to 4 and is set to an initial reading of count 1 only by a jamming pulse which is an input to buffer 2 of FF1. The counter counts 1 if FF1 is set; that is, if output signal 1 is low; it counts 2 if FF2 is set, and so forth. The counter is stepped to counts of 2, 3, and 4 by successive stepping pulses which are inputs to the two connecting diodes. The counter can be stepped to the next count only if it has been counting the previous count.

The complete operating sequence of the counter is as follows: Initially the four flip-flops of the counter read 0; that is, all four flip-flops are in a restored state in which output lines 1, 2, 3, and 4 are high. A high jamming pulse into buffer 2 jams the counter to the 1 count by setting FF1 (output line 1 goes low). The first high stepping pulse is complemented by B-phase complementer 12 (figure 2-9) so that it is a low input to gate 3 where it is gated with the low output of FF1. Gate 3 is made permissive and sets FF2. The same stepping pulse goes through B-phase noncomplementer 13 (figure 2-9), the output of which is a high input to gate 1. Flip-flop 1 is restored.

The low-output signal from FF2 is gated with the next low stepping pulse at gate 6 to set FF3. The high stepping pulse to gate 4 restores FF2. The next low stepping pulse to gate 9 is gated with the low output of FF3 and sets FF4. The high stepping pulse to gate 7 restores FF3. Flip-flop 4 is restored by the next high stepping pulse to gate 10. The initial jamming pulse plus three stepping pulses enable the counter to count from 1 to 4.

If the counter had additional flip-flops, it could be made to count from 1 to any desired number. In the logical circuitry, counters similar to the one shown in figure 2-9 are used to count from 1 to 4, 1 to 10 and other series.

2-35. SHIFT REGISTER

A shift register is a circuit that consists of a number of flip-flops connected in series. It is the intermediate-

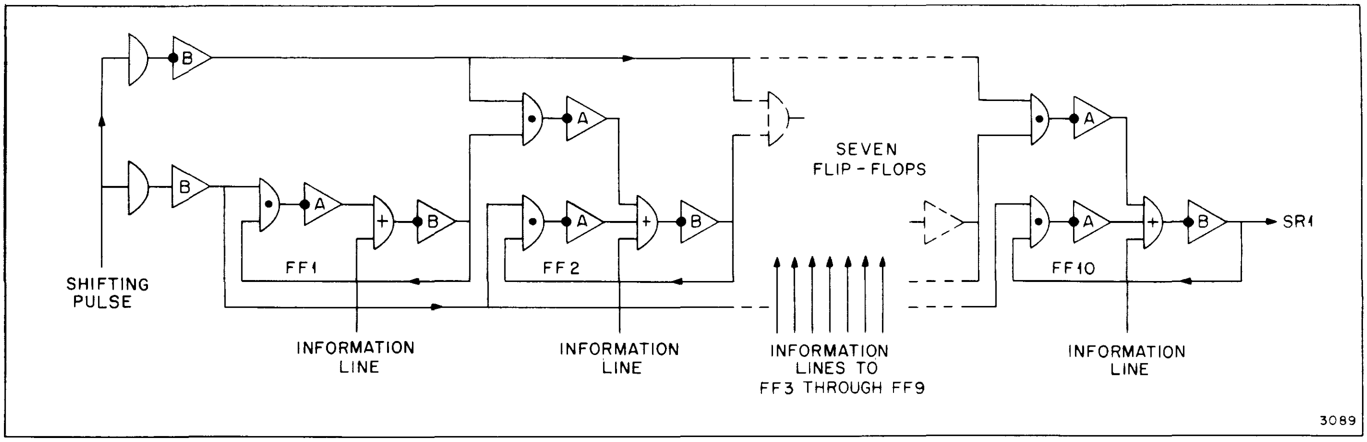


Figure 2-10. Typical Shift Register

storage unit that converts information flow from parallel to serial form or from serial to parallel form. A conversion from parallel to serial is needed when information on a punched card, read in parallel by the input-output units, must be written serially into a buffer track of the storage drum. Conversely, a conversion from serial to parallel form is needed when output information is stored serially in the register and is transferred in parallel to the punch actuators of the read-punch unit. The flip-flops of a shift register resemble those of the counter circuit in the logic drawing. A shift register, however, does not count the number of times an operation is performed, but temporarily stores information. The shift registers consist of ten flip-flops, connected in series, storing ten information bits, one bit per flip-flop.

A typical shift register is shown in figure 2-10. The individual flip-flops resemble those of the counter circuit (figure 2-9) but differ from them in that each flip-flop has an information-input line to its buffer. Also, the output of each flip-flop goes only to the set gate of the next flip-flop in line.

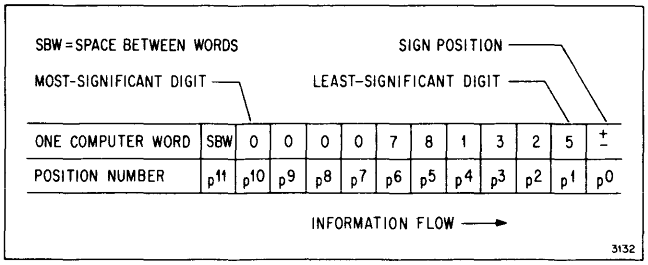
In a typical operation, ten bits of information enter the ten flip-flops in parallel on the ten information lines. A high signal on an information line is a binary 1 that sets the flip-flop to which it is an input. The output line of any flip-flop, therefore, is low if it is storing a binary 1. When the ten bits of information have been stored in the ten flip-flops, each flip-flop is storing a 0 or a 1, according to what was on its information line. Each flip-flop continues to store its pulse until a shifting pulse moves each bit of information one flip-flop to the right, so that if FF1 had a low pulse, FF2 now has a low pulse, and so forth. One pulse time after the first shifting pulse, the pulse that was circulating in FF10 is shifted to the SR1 (shift register 1) line. The shifting operation for the shift register is identical with the stepping operation of the counter circuit.

After ten shifting pulses (assume one pulse time between each shifting pulse), all ten pulses originally stored in the ten flip-flops have been shifted to the SR1 line. Thus, ten bits of information that were available in parallel before they entered the shift register are put on the SR1 line serially, one pulse each pulse time (the rate at which they are written on the drum).

2-36. TIMING REFERENCE

Most of the logic drawings are marked in one or more places with timing references. These references make it possible to determine the location of any of the 12 digits of a word in the circuitry during any of the 24 intervals of a word time.

For convenient reference to the 12 digits of a word, each digit is assigned a position number. The notation used to indicate the 12 positions of a computer word is shown in figure 2-11, in which each digit of the number 0000781325 is assigned a digit-position number. When p0 is referred to, the reader knows that this is the sign (+ or —) of the word, whereas p1 is the least-significant digit, a 5 in this example, and so forth. Although this is only a six-digit number, it always moves in the computer as a complete word. The p7 to p10 digits of figure 2-11, therefore, are 0's.



ONE COMPUTER WORD	SBW	0	0	0	0	7	8	1	3	2	5	±
POSITION NUMBER	p^{11}	p^{10}	p^{9}	p^{8}	p^{7}	p^{6}	p^{5}	p^{4}	p^{3}	p^{2}	p^{1}	p^{0}

Figure 2-11. Digit Positions of a Computer Word

A typical timing-reference notation is p0 = t0B. This notation is placed on a drawing to indicate the exact location in the circuitry where p0 (the sign of the word) is at time interval t0B. Knowing this and knowing that a pulse is delayed half a pulse time in passing through a magnetic amplifier, the time at which p0 arrives at any point in the circuitry can be calculated. Reference timing is discussed in greater detail in the following paragraphs.

2-37. CIRCULATING REGISTERS

The four circulating registers (C, A, X, and L) are basically similar in that each can store one numeric computer word. They are called circulating registers because the word of information is stored dynamically; that is, it constantly circulates in the register. The plus or minus sign associated with each computer word is not stored in the register but is stored in flip-flops. A computer word circulating in a register occupies 12 digit positions, counting the sign position and the space-between-words position, but only ten of these positions are occupied by information digits.

(The uses of a circulating register are discussed in section 3. Under headings 2-38 through 2-40, register L is discussed in detail to show how information enters, circulates through, and leaves a register.)

As shown in drawing 1-12A, register L consists of four subregisters, numbered 4 through 1 from top to bottom. Each subregister stores ten bits; subregister 1 stores the ten least-significant bits, subregister 2 stores the ten next-higher-order bits, and so forth. For example, if the number 0000871342 were stored in register L, the distribution of the bits would be as shown in figure 2-12. The zone bits of an alphabetic word are stored in subregisters 1 and 2 of another register.

2-38. INPUT GATES

A new word of information enters register L through a set of input gates. One group of input gates is numbered 2A, 2B, 2C, and 2D, each of which has an input from the M-buffer circuit; M1 goes to subregister 1, M2 goes to subregister 2, and so forth. The gates numbered 3A, 3B, 3C, and 3D are also input gates of register L and function in the same manner as the input gates just mentioned, except that the word of information which enters by these gates comes from the S buffer as indicated by the input lines, S1, S2, S3, and S4.

2-39. RECIRCULATING GATES OF REGISTER L

Gates 1A, 1B, 1C, and 1D are the recirculating gates of register L. A word of information in the register continues to circulate until cleared by the 57+ signal on the recirculating gates. Function signal 57+ is normally low, but during the execution step of an instruction that has 57+ as one of its function signals, 57+ goes high. The high 57+ signal blocks the recirculation gates and prevents the word of information in the register from circulating.

The notation p0 = t0B above gate 1D (drawing 1-12A) is the timing reference for the drawing. This notation means that p0, the sign, is an input to the recirculating gates or a set of input gates at time t0B. Although the sign is present as an input to the gates at t0B, it is blocked by timing signal t0B+. With the aid of timing reference, the timing of a digit at any point in the register can be established. For example, digit p1 is an input to gates 1A, 1B, 1C, and 1D at t1B; p2 is an input at t2B, and so forth. Because it takes one pulse time for a pulse to travel through an A- and B-phase amplifier, the timing notation p0 = t0B in front of gate 1D agrees with the notation p0 = t11B after gate 4D.

BIQUINARY CODE

DECIMAL VALUE	BITS			
	4	3	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	0	0
6	1	0	0	1
7	1	0	1	0
8	1	0	1	1
9	1	1	0	0

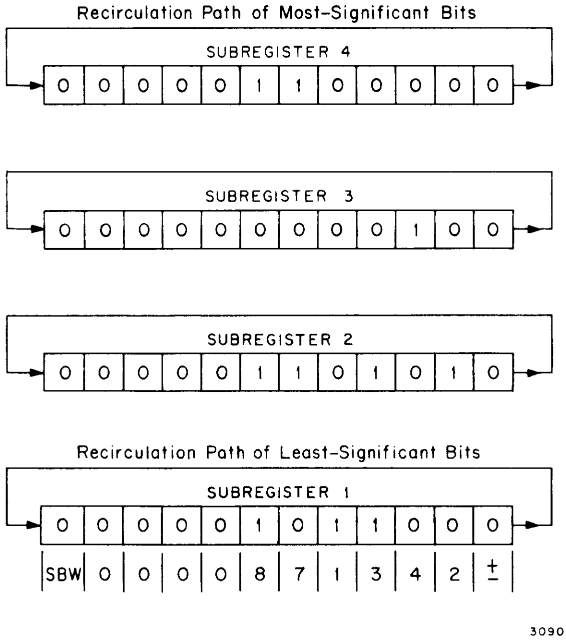


Figure 2-12. Distribution of Bits in a Circulating Register

One of the simplest instructions is the 30 (ML) instruction which transfers the contents of the storage location designated by m to register L. The function signals for the 30 instruction are 8, 57+, and 67. Only FS8 and FS57+ are considered here since FS67 does not appear on drawing 1-12A. During the execution of the 30 instruction, FS8 and FS57+ allow a new word from the M buffers to enter register L. Function signal 57+ goes high for one word time to clear the information circulating in the register. When 57+ goes high, FS8 goes low for one word time so that a new information word on the MT1, MT2, MT3, and MT4 lines can enter the register through input gates 2A, 2B, 2C, and 2D.

Information entering the register on the MT lines consists of 1 bits and 0 bits. A 1 on any of the MT lines is a low signal that is gated with low FS8. The output of the gate goes through two complementers so that a binary 1 in a subregister is a low signal. A binary 0 on an MT line is a high signal that circulates in a subregister as a high signal. The biquinary code for a decimal 7, for example, is 1010. On the MT lines a 7 would be: MT1 high, MT2 low, MT3 high, and MT4 low, as shown in the following example:

1	0	1	0
MT4	MT3	MT2	MT1
low	high	low	high

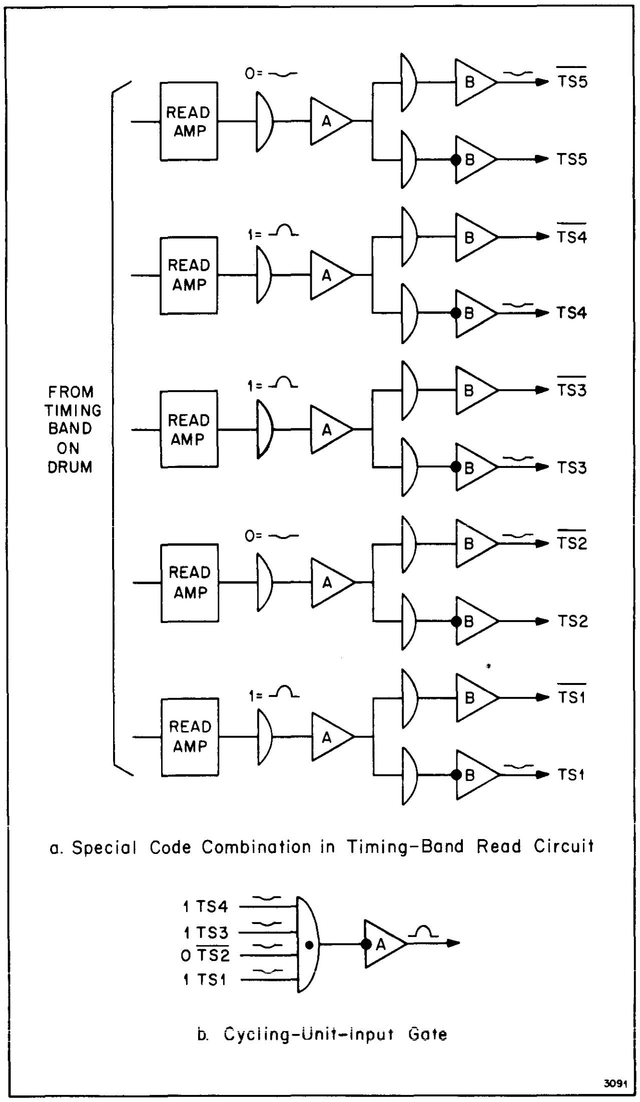
It takes one word time for a computer word to enter register L. At the end of this word time, FS57+ goes low so that the new word circulates in the register, and FS8 goes high to block the input gates.

2-40. OUTPUT GATES OF REGISTER L

The output gates of register L are gates 4A, 4B, 4C, and 4D, all alerted by FS66. During most operations, the information word in register L is not being sent to the L1, L2, L3, and L4 lines because FS66 is high, blocking gates 4A, 4B, 4C, and 4D. When this signal goes low during an instruction, the information word in register L is gated out to the L lines and continues to circulate in the register. Function signal 66 is generated by instructions 82, 87, 85, and 55, as indicated on the logic drawings by the presence of the letters CE, CGT, MPY, and D near FS66. These mnemonics indicate the instructions. (Refer to heading 2-45.)

2-41. INFORMATION-GATING CIRCUIT

The timing signals used throughout the computer are generated by the cycling unit, which is activated once each word time by special code combination 1101. The



a. Special Code Combination in Timing-Band Read Circuit

b. Cycling-Unit-input Gate

Figure 2-13. Special Code Combination in Timing-Band Read Circuit

1101 combination is gated at the input gate of the cycling unit. This gating action is explained in detail in the next four paragraphs.

Figure 2-13a shows the timing-band read amplifiers which also appear on logic drawing 1-19A. This circuit is always active; that is, the timing-band read amplifiers are always reading the timing-band signals. One of the digit spaces in each of the 200 locations of the timing band contains the special code combination 1101 which is read by the read amplifiers once each word time.

As illustrated in figure 2-13a, a high signal coming off the drum has a binary value of 1, a low signal a binary value of 0. One pulse time later, these signals become the TS1 to TS5 and $\overline{\text{TS1}}$ to $\overline{\text{TS5}}$ output signals. The TS5 and $\overline{\text{TS5}}$ signals, the check-bit signals, appear as outputs from the timing-band read amplifiers, but do not appear as inputs to the cycling-unit input gate.

Figure 2-13a shows that the signals for each of the bits of the special code combination are available in two forms: complemented (TS1 to TS4) and noncomplemented ($\overline{\text{TS1}}$ to $\overline{\text{TS4}}$). Figure 2-13b shows that the input gate to the cycling unit (drawing 1-18A) has input signals TS1, $\overline{\text{TS2}}$, TS3, and TS4. When combination 1101 is present on the TS lines, the gate shown in figure 2-13b has all low signals on its input lines. The result is a high signal following the A-phase amplifier. This high signal is one of the timing signals generated for each of the 24 time intervals of a word time. Any code combination other than the special code combination results in a low-output signal following the A-phase amplifier.

The gating action described in the preceding paragraphs illustrates the important point that a signal can be inverted electronically without logically losing its identity. The 0 of the special code combination is represented by a high signal coming off the timing band. To put all low signals on the gate in figure 2-13a, it is necessary to make the $\overline{\text{TS2}}$ and not the TS2 signal an input to the gate. This low signal still represents a 0 because it is the complement version of the original signal. Electronically, the original high 0 signal has been inverted; logically it still represents a 0.

2-42. SPECIAL-CIRCUIT ELEMENTS

Special-circuit elements are discussed in this section to aid the reader in interpreting the logic drawings. A detailed discussion of these electronic circuits is included in section 6 of this manual.

2-43. ARITHMETIC AMPLIFIERS

The arithmetic amplifiers (figure 2-14a) are noncomplementers and can be either A or B phase. The letter S (in the triangle with letter A or B) designates a special type of amplifier which is simpler and more economical to build than the standard amplifier. It is not necessary to put connecting diodes between arithmetic amplifiers. A special amplifier can drive only another special amplifier or a transition amplifier, designated by the letter T in the triangle with the letter A or B. The transition amplifier can drive only a standard noncomplementer. Neither the special nor the transition-type amplifier can drive gating or buffing circuits.

The special and transition-type amplifiers are used in the recirculation loops of the arithmetic registers. The main loop is made up of a series of special amplifiers. Near an output of each subregister, a transition amplifier and a standard noncomplementer precede the output line to provide the necessary driving force.

2-44. TRANSISTOR-DRIVING AMPLIFIERS

The outputs of transistor-driving amplifiers go to transistor circuits rather than to standard amplifiers and they perform the same logical functions as standard complementers and noncomplementers. The symbols for transistor-driving amplifiers are shown in figure 2-14b. The difference between the square symbol and the triangle within the square in figure 2-14b indicates an electronic difference, not a logical difference.

2-45. INSTRUCTION MNEMONICS

Each instruction in the system is identified by its two-digit instruction code and by its mnemonic. The two-digit code is recognized by the logic circuitry; the mnemonic provides a simple, meaningful name for each instruction. Mnemonics are shown next to the function signals on the logic drawings to indicate the instructions with which the function signal is associated. For example, the mnemonic ML next to FS8 and FS57 (drawing 1-12A) indicates that these are function signals of the 30 (ML) instruction. The mnemonic AL next to FS57 indicates that this signal is also generated by the 77 (AL) instruction. Tables 2-1 and 2-2 list the instructions in alphabetic mnemonic order and in numerical sequence.

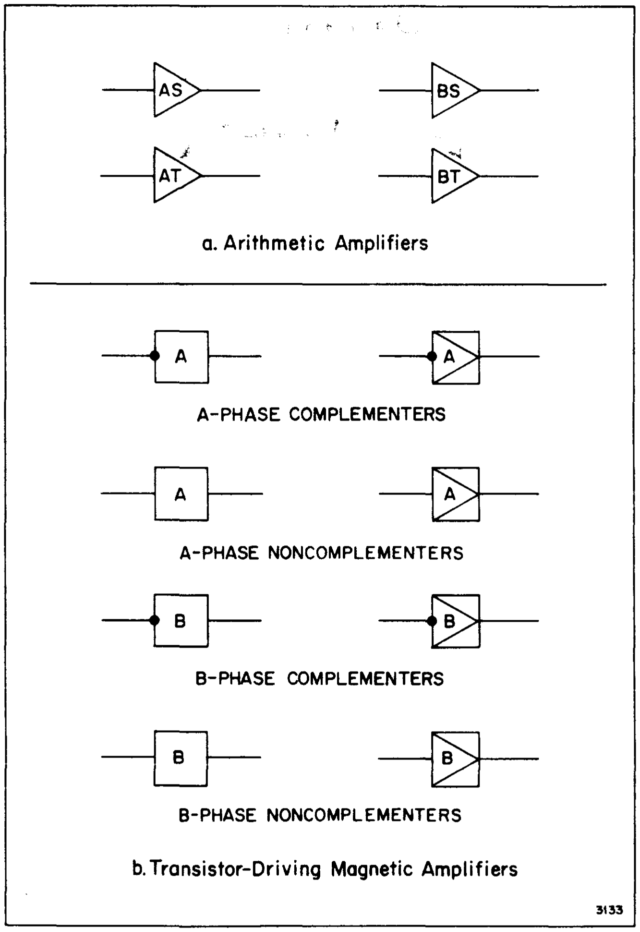


Figure 2-14. Special-Type Amplifiers

Table 2-1. Instruction Mnemonics (Alphabetic Listing)

Mnemonic	Machine Code	Function
A	70	Add
AL	77	Transfer (rA) ——> rL
AM	60	Transfer (rA) ——> m
CE	82	Compare for equality
CGT	87	Compare for greater than
D	55	Divide
EX	35	Extract
FP	81	Feed one card into punch
FR	72	Feed one card into reader
IA*	07	Increment index register
IL*	02	Load index register
LM	50	Transfer (rL) ——> m
MA	25	Transfer (m) ——> rA
ML	30	Transfer (m) ——> rL
MPY	85	Multiply
MX	05	Transfer (m) ——> rX
PBM	46	Transfer read-punch buffer ——> m
PF	16	Advance paper
PR	11	Print
RBM	96	Transfer reader buffer ——> m
S	75	Subtract
SC	none	Search
SHL	37	Shift left
SHR	32	Shift right
SI	20	Superimpose
SPS	57	Select punch stacker 1
SRS	47	Select reader stacker
ST	67	Stop
SZ	none	Staticize
TCU	12	Translate card code ——> USS-6
TEP	27	Test printer
TEPB	22	Test read-punch buffer
TERB	42	Test reader buffer
TUC	17	Translate USS-6 ——> card code
XM	65	Transfer (rX) ——> m
ZS	62	Zero suppress

*Optional

EP : [illegible]

CT Cond'l Transfer

R [illegible]

DI [illegible]

0 : [illegible]

GEN : [illegible]

[illegible]

BSX

IR [illegible]

CP [illegible]

Table 2-2. Instruction Mnemonics (Numeric Sequence)

Machine Code	Mnemonic	Function	Machine Code	Mnemonic	Function
—	SC	Search	46	PBM	Transfer read-punch buffer ——> m
—	SZ	Staticize	47	SRS	Select reader stacker
02*	IL	Load index register	50	LM	Transfer (rL) ——> m
05	MX	Transfer (m) ——> rX	55	D	Divide
07*	IA	Increment index register	57	SPS	Select punch stacker
11	PR	Print	60	AM	Transfer (rA) ——> m
12	TCU	Translate card code ——> USS-6	62	ZS	Zero suppress
16	PF	Advance paper	65	XM	Transfer (rX) ——> m
17	TUC	Translate USS-6 ——> card code	67	ST	Stop
20	SI	Superimpose	70	A	Add
22	TEPB	Test read-punch buffer	72	FR	Feed one card into reader
25	MA	Transfer (m) ——> rA	75	S	Subtract
27	TEP	Test printer	77	AL	Transfer (rA) ——> rL
30	ML	Transfer (m) ——> rL	81	FP	Feed one card into punch
32	SHR	Shift right	82	CE	Compare for equality
35	EX	Extract	85	MPY	Multiply
37	SHL	Shift left	87	CGT	Compare for greater than
42	TERB	Test reader buffer	96	RBM	Transfer reader buffer ——> m

*Optional

CM [illegible]

RTS [illegible]

BR 1 [illegible]
BR 2

RS Read [illegible] Fast

FS Fast [illegible]

PBC

CLQ [illegible]

SP [illegible]

ST [illegible]

SI [illegible]

RD [illegible]

CG

PCS [illegible] Cell Signal

SC [illegible]

CCC Card Cycle Co[illegible]

BL Buff[illegible]

[illegible] Buff[illegible]

PCS photo Cell Signal

OCT [illegible]

Section 3

DESCRIPTION OF LOGIC CIRCUITS

The purpose of this section is to give a detailed explanation of the logic circuits of the Processor, excluding the input-output synchronizers. Knowledge of the circuits is necessary for an understanding of sections 4 and 5, in which these components are referred to frequently within the more complex operations of the USSC system. The text in this section continues to refer to the logic drawings in manual 2.

3-1. GENERAL

The USSC Processor consists of four logical units: control, arithmetic, storage, and input-output. The input-output unit is detailed in manual 3. This section deals only with the control, arithmetic, and storage units.

3-2. CONTROL UNIT

The control unit interprets programmed or manually inserted instructions and controls the operations necessary to perform the instructions. Programs enter into computer storage from punched cards or the operator's keyboard. Each program instruction is transferred from storage, in the sequence specified by the program, to the control unit. Elements of the control unit translate the biquinary representation of each instruction to signals which control the internal operations of the computer in performing the instruction.

The major components of the control unit are the static register, the instruction decoder, the function encoder, register C, the operator's control panel, and, if index registers are included in the system, the three index registers and the index-register adder.

3-3. STATIC REGISTER

The static register (drawing 1-2A) consists of the static-register flip-flops, the stage-advancing gate, the ending-pulse buffer, D3 flip-flop, and the input-output (I/O) abnormal-condition flip-flop. (Refer to heading 4-41.)

3-4. STATIC-REGISTER FLIP-FLOPS. The seven static-register flip-flops store the seven bits of the two instruction digits. Because the third bit of the least-significant digit is unnecessary for the staticizing operation, the flip-flops are numbered 1, 2, 4, 5, 6, 7, and 8. Each of the seven bits of the two instruction digits is stored in its respective flip-flop until the ending pulse restores all the flip-flops.

After an ending pulse restores the flip-flops, an instruction word is read from storage into the M buffers. The seven bits of the two instruction digits (p10 and p9) of the instruction word become the MT output signals of the buffers and go to the input gates of the seven static-register flip-flops. The three bits of the least-significant digit, p9, enter FF1, FF2, and FF4 at gates 5A, 5B, and 5D. One pulse time later the four bits of the most-significant digit, p10, enter FF5, FF6, FF7, and FF8 at gates 6A, 6B, 6C, and 6D. A 0 bit on the MT lines (a high MT signal) keeps the flip-flop restored to generate a low $\overline{\text{STR}}$ output. A 1 bit on the MT lines (a low MT signal) sets the flip-flop to generate a low STR output. Thus, a seven-bit combination is stored in the seven flip-flops with low $\overline{\text{STR}}$ output indicating a 0 bit, and a low STR output indicating a 1 bit. The outputs of the seven flip-flops drive the instruction decoder and function encoder to generate signals to execute the instruction. The character "X" shown in any output combination of the static-register flip-flops, such as 0X10, is used to balance a four-bit combination or to indicate that the bit can be either a 1 or a 0. In all output combinations, the p3 bit is shown as an X because only three bits are required for the p9 digit. If the system includes index registers, this bit is required to partially designate one of the three registers.

The condition of the flip-flops can be changed by the input signals to each flip-flop and by the stage-advancing gate or ending-pulse buffer.

3-5. STAGE-ADVANCING GATE. Stage-advancing gate 26 steps FF1 and FF2 to 1 outputs to advance the staticized instruction to the next execution stage. During the arithmetic, shift, input-output test, and zero-suppress instruction, FS64 alerts gate 26. If the ending pulse (EP) is low, indicating that the instruction is not yet completed, the gate is permissive. The high output of the complementer sets FF1 and FF2 to STR1 and STR2 outputs. This steps the static register to the second stage of the staticized instruction. For example, the right-circular shift (32) instruction is staticized in the flip-flops as 0011 0X10 (STR8 is the most-significant bit; STR1 is the least-significant bit), which is the combination for 32. FS64 is generated during the first step of the instruction. The stage-advancing gate sets FF1, which was restored to an STR1 output. The stage-advancing gate steps both FF1 and FF2 unless one of these is already in the set condition, as is FF2 in this case. The staticized combination is therefore stepped to 0011 0X11 (33), which drives the encoder and decoder to execute the second step of the instruction.

3-6. ENDING-PULSE BUFFER. The ending-pulse gates sample various computer circuits for indications that an instruction is completed. The instruction is ended when these indications are present at one of the ending-pulse gates as low signals, causing the ending-pulse buffer to generate a high EP signal. The EP signal restores all seven static-register flip-flops to the barred output, such as $\overline{\text{STR1}}$ (0). When the seven flip-flops are cleared, the next-instruction digits are staticized. Table 3-1 lists the ending-pulse gates and the instructions they end.

3-7. BLOCK-READ FLIP-FLOPS. The two block-read flip-flops block the main-storage-read circuits during a search for the m or c address in which the search is for the contents of an arithmetic register rather than a storage address. The programmer addresses a register by placing one of the following bit combinations in either the p1 (for c address) or p5 (for m address) position of the instruction word: 0101, 0111, or 0110. The 0101 combination causes readout from register A; 0111 from register X, and 0110 from register L.

During a search step the contents of register C are on the S lines. If the S1 and S3 signals are both low at gate 120, register A or X is addressed. When gate 120 is permissive, the BRD1 flip-flop is set to a high block-RD1 output. If the S2 and S3 signals are both low at gate 122, register L or X is addressed. When gate 122 is permissive, the BRD2 flip-flop is set to a high block-RD2 output. Thus if only FF1 is set, register A is addressed; if only FF2 is set, register L is addressed; and if both flop-flops are set, register X is addressed. The high block-RD1 and block-RD2 signals block circuits which normally permit a readout from storage, generate function signals, and alert the output gates of the correct arithmetic registers. When STR FF1 is stepped to a 1, indicating that the search is complete, the STR1 signal makes gate 124 permissive, restoring both block-read flip-flops.

3-8. RUN FLIP-FLOPS. In a normal search-for-next-instruction operation, the static register contains all 0's. When the next instruction is found, the static register is changed to a combination other than all 0's. This search operation can take up to 200 word times to complete. Under normal conditions, then, the longest period of time that the STR contains all 0's is 200 word times, in the search-for-next-instruction operation. Under abnormal, or computer-stop conditions, however, the STR can contain 0's for a longer period because no instruction is staticized until the condition is remedied. The run flip-flops ensure that a computer-stop condition does not turn off the RUN lights on the control panels before the 200 word times have elapsed. As a result, a normal search operation does not turn off the RUN lights to indicate a computer stop.

Only one TS2 at t9B— sentinel (BT) is recorded on the timing band. It is read from the band once every 200 word times to set the first run flip-flop every drum revolution. A high ending pulse (EP) restores the flip-flop under normal conditions after every instruction. If no instruction is processed, EP remains low during the 200 word times, and the flip-flop remains set. The next time the sentinel is read from the drum, it sets the second run flip-flop if EP is still low. The set output of the second flip-flop turns off the RUN lamps to indicate a computer stop.

The first run flip-flop is jammed by FS45A or FS29A to a restored state during transfers between card buffer and main storage to keep the run signal high during the first part of the transfer. If translation is included in the transfer, the low run signal will alert the ending-pulse gate (309) after the BT sentinel sets the first run flip-flop.

Also associated with the static register are the D3 and abnormal-condition (I/O) flip-flops. (Refer to section 4.) The D3 flip-flop initiates the third execution step of the divide instruction (D3). The D3 flip-flop generates FS32 and FS32A, which control the third step of division.

3-9. INSTRUCTION DECODER

The instruction decoder (drawing 1-3A) is a network of diode gates which receives the pulse-combination outputs of the static register and converts them to function signals. Each flip-flop of the static register produces either of two possible outputs. For example, static-register FF1 produces either an $\overline{\text{STR1}}$ or an STR1 output. The barred output signifies a 0 when it is low, and the unbarred

Table 3-1. Ending-Pulse Gates and Instructions Ended

Gate	Instruction Ended	Indication
13	46 (read-punch buffer to main storage); 96 (card-reader buffer to main storage)	$\overline{\text{BBTRLAT}}$ signal on gate from buffer-band-translate flip-flop indicates instruction did not include automatic translation.
14	32 (shift-right); 37 (shift-left)	If zero shift is specified.
15	Keyboard-input operation	
16	22 (read-punch test)	If buffer is not loaded.
17	All other instructions	
18	85 (multiply)	
19	Register (A, L, or X) to main storage	
20	11 (print)	
21	16 (paper-feed)	
111	72 (reader card-cycle)	
112	27 (printer-test)	If printer is not available.
113	42 (card-reader-test)	If buffer is not loaded.
208	81 (main storage to read-punch buffer)	The $\overline{\text{BBTRLAT}}$ signal on the gate from the buffer-band-translate flip-flop indicates instruction did not include automatic translation.
309	81 (main storage to read-punch-buffer); 46 (read-punch buffer to main storage); 96 (card-reader buffer to main storage)	The BBTRLAT signal on gate from the buffer-band-translate flip-flop indicates instruction did not include automatic translation.

output signifies a 1 when it is low. Seven such outputs from the static register make permissive only one of the diode gates in the instruction decoder.

If, for instance, the 50 instruction is staticized in the static register, the pulse combination sent to the instruction decoder is STR8 (1), $\overline{\text{STR7}}$ (0), $\overline{\text{STR6}}$ (0), $\overline{\text{STR5}}$ (0), $\overline{\text{STR4}}$ (0), $\overline{\text{STR2}}$ (0), and $\overline{\text{STR1}}$ (0). The pulse combination for the first digit of the instruction is 1000, for the decimal 5. The combination for the second digit is 0X00, for the decimal 0 (X indicating the third bit, which is unused unless the system is equipped with an index register). Only the gate line labelled 50 is permissive to this 1000 0X00 combination, generating FS23A.

The signals generated by the instruction decoder are sent to the function encoder to generate the function signals that control internal computer operations. In addition, some of the signals generated by the instruction decoder are sent in parallel to control other circuits. Therefore, all signals generated by either the instruction decoder or function encoder are known as function signals.

The origin of a function signal usually can be determined by its phase. Signals originating in the instruction decoder are A-phase signals; those originating in the function encoder are generally B-phase signals. When function signals are referred to, the presence of an A, following the signal number (such as FS12A), denotes an A-phase signal; the absence of an A (such as FS76) denotes a B-phase signal. All the signals produced by the decoder to execute a specific instruction are high signals, with the exception of low signal 60A.

In addition to the STR outputs of the static register, control signals OR+, IER+, $\overline{\text{SP}}$, and $\overline{\text{SPX}}$ affect the instruction-decoder gates. The OR+, and IER+ signals from the computing circuits are present during divide and multiply instructions to block generation of certain function signals. The $\overline{\text{SP}}$ or $\overline{\text{SPX}}$ signals, which indicate a computer-stop condition, block generation of any function signals. Function signal 100 and timing signal t10B— alert the instruction decoder if index registers are included in the system.

During buffer-to-main-storage transfers with translation, the MQC serves as an instruction decoder. The eight gates requiring the BBTRLAT signal decode the reading in the MQC.

3-10. FUNCTION ENCODER

In the normal execution of an instruction, the function encoder (drawing 1-4A) receives an A-phase function signal from the instruction decoder and encodes this signal into a number of function signals. The instruction decoder performs the logical function of a gate because a number of low inputs are necessary to generate a single output signal. Although the function encoder is shown as a diode matrix similar to the instruction decoder, it performs the logical function of a buffer because any high input produces one or more output signals. A buffer on the function encoder is a vertical diode line on the matrix and every diode (shown as a dot) on that line is an input to the buffer. The high 1A signal, for example, goes to four buffers which generate function signals 1, 58+, 74, and 63. For any buffer to operate and generate a function signal, one high input must be applied to it. In this example, high signal 1A is the signal produced by the instruction decoder; therefore, all other outputs of the instruction decoder to the function encoder are low. The low-signal inputs to the four buffers have no effect on the function encoder.

The function encoder receives inputs from sources other than the instruction decoder. Input signals from the control panel, the IER, OR, and D3 flip-flops, and the printer-control circuit also generate function signals from the encoder.

During buffer-to-main-storage transfers with translation, the MQC serves as an instruction decoder. The GEN signals generated by the reading in the MQC are sent to the function encoder. For example, if the reading set up in the MQC is 0X11, a high GEN 18 signal is produced and sent to the function encoder to produce function signals 18, 55+, 56+, and 57+. These function signals simulate the 17 instruction which translates the word stored in registers A and X from USS-6 to card code. The other GEN signals are GEN 17X, GEN 77, GEN 18, GEN 76, GEN 17, GEN 99, and GEN 66.

3-11. REGISTER C

Register C (drawing 1-1A) stores the ten digits of the instruction word read from storage during a search-for-instruction operation. Four parallel subregisters store the two digits of the instruction, the four digits of the m address, and the four digits of the c address. When the instruction word is stored in register C, the p9 and p10 digits, which make up the instruction code, are also stored in the static register. Storage of the same two digits in register C is for control-panel display purposes.

3-12. OPERATOR'S CONTROL PANEL

The operator's control panel on the Processor cabinet contains the switches and indicators necessary for manual control of the computer. The engineer's panel (concealed) on the same cabinet is primarily for trouble-shooting purposes and is therefore discussed in manual 4. Each input-output unit has a separate panel for controlling the operations of the individual unit. These panels are explained in the manuals for each unit. The descriptions of the major operator-controlled operations are in section 4.

3-13. INDEX-REGISTER CIRCUIT (OPTIONAL)

An index register stores four digits which modify the m address of an instruction word stored in register C to permit the programmer to use one basic instruction in processing a large amount of information stored in sequential storage addresses. The m address of the instruction is incremented by the number stored in an index register each time a new piece of information is processed. The basic instruction is staticized by storing the entire instruction in register C and the two instruction digits in the static register. An extra word time is added to the staticize step of the basic instruction cycle during which the m address from register C is added to the contents of an index register and the resulting modified m address is returned to register C. The search for the new m address (operand) follows the modifications, and the instruction is then executed, using the contents of the m address. The extra word time in the cycle is described in detail in section 4.

The programmer can use either of two instructions, 02 or 07, to initially store the digits in the index register. The 02 instruction clears the index register of previous contents and loads it with the four digits which made up the m address of the 02 instruction. The 07 instruction does not clear the index register but instead reads out the contents of the index register onto the B lines and then onto the M lines. At the same time, the four digits of the m address of the 07 instruction are read from register C onto the S lines. The information on the M and S lines is then added in the index-register adder and the sum returned to the m-address portion of register C and to the index register. The sum is also stored in the m-address portion of register A and the remainder of register A cleared to 0's. The stored sum in register A permits the programmer to test the contents of register A against a constant stored in register L. (The 02 and 07 instructions are described in section 5.)

The three index registers and their associated circuits are shown on drawings 1-24A and 1-25A. The circuits shown on 1-24A are the three registers, the index-register read-in flip-flop, the index-register read-out flip-flop, the rC-sign flip-flop, and STR3 flip-flop which select one of the three registers. The index-register adder and its gates are shown on drawing 1-25A. The add-9800 flip-flop which is required during the additional word time of the staticize step and during the 07 instruction, is located on drawing 1-7A and is described under heading 3-58.

3-14. INDEX REGISTER

This section describes index register 1, the operation of which is similar to registers 2 and 3. The four digits stored in the register recirculate through gates 35, 37, 39, and 41 unless they are blocked by a high signal from gate 43, which is permissive when new information is read into the register. Input gates 34, 36, 38, and 40 are alerted to information from register C on the C13, C23, C33, and C43 lines if FS101 is low and gate 42 is permissive. Output gates 22, 23, 24, and 25 are alerted if gate 42 is permissive and FS100 is present. The information reads out of the register onto the B11—B14 lines and is then transmitted onto the M lines for addition to the contents of register C during the additional word time of the staticize step or during the 07 instruction. The 02 instruction does not read information out of the register.

The registers are designated, during the 02 or 07 instruction and the additional staticize word time, by the first bit of the digit which indicates the sign of the instruction stored in the rC-sign flip-flop and by the third bit of the p9 digit of the instruction code stored in the STR3 flip-flop. The decoding is as follows:

sign of rC	rC-sign FF Signal	p9 digit	STR3 FF Signal	Register Specified
0	$\overline{\text{BBA1}}$	1	BBA2	IR-1
1	BBA1	0	BBA2	IR-2
1	BBA1	1	BBA2	IR-3
0	$\overline{\text{BBA1}}$	0	$\overline{\text{BBA2}}$	none

3-15. REGISTER C SIGN FLIP-FLOP

The rC-sign flip-flop stores the first bit of the sign digit of the instruction which is on the M1 line at t0B of the staticize step. If the sign digit is a 1 for a minus (a 1 bit), gate 3 is made permissive to set the flip-flop, generating low C— and BBA1 signals and a high $\overline{\text{BBA1}}$ signal. If the sign digit is a 0 for a plus (a 0 bit), gate 2 is made permissive to restore the flip-flop. During keyboard input the high WRA signal from the word-release flip-flop (drawing 1-5A) blocks both gates 2 and 3 and either an WRC— or WRC signal jams the flip-flop to the required state.

3-16. STR3 FLIP-FLOP

The STR3 flip-flop stores the third bit of the p9 digit of the instruction which is on the M3 line at t9B. The flip-flop is initially restored at gate 8 by a high RBBA signal, which is generated at t9B of the staticize step by restoring static register FF1 (drawing 1-2A). If the third bit of the p9 digit is a 1, gate 7 is made permissive to set the flip-flop, generating low STR3 and BBA2 signals and a high $\overline{\text{BBA2}}$ signal. If the third bit is a 0, the flip-flop remains restored.

3-17. INDEX-REGISTER READ-OUT FLIP-FLOP

The index-register read-out flip-flop is set to read information out of an index register. Either gate 17 or gate 18 is made permissive at t10B of the staticize step of any instruction which specifies an index register. This setting is required if either an 07 instruction or an instruction requiring modification is staticized. The setting is not required if an 02 instruction is staticized. Therefore, in this case gate 11 is made permissive to restore the flip-flop at gate 19. The high GEN $\overline{\text{SP1}}$ or GEN $\overline{\text{SP2}}$ signals generated by set gates 17 and 18 are sent to buffer 28 (drawing 1-5A) to generate one pulse time of $\overline{\text{SP}}$ and $\overline{\text{SPX}}$ which block the gates of the instruction decoder. The high B58+ signal from the flip-flop keeps the instruction decoder blocked for the remainder of the word time by generating SP and SPX. The B58+ signal is also sent to the function encoder to generate FS58+ which blocks the contents of register A from the S lines so that the m address from register C can be read onto them. Function signal 100 goes low for one word time to alert the output gates of the selected register and also alerts the register-adder gates (drawing 1-25A). Function signal 100A— goes low to generate Bt0-3B— and Bt0-3B+ signals (drawing 1-18A).

The IBB signal which blocks set gates 17 and 18 is only high if the computer is being operated in one instruction mode with the index-register modification inhibited.

3-18. INDEX-REGISTER READ-IN FLIP-FLOP

The index-register read-in flip-flop is set to read information into an index register. A high at buffer 12 sets the flip-flop during an 02 or an 07 instruction. If an 02 instruction is staticized, gate 11 is made permissive to place a high on buffer 12. During the execution step of the 07 instruction, a high S101A signal is generated to place a high on buffer 12. Function signal 101A— goes low to make gate 43, 53, or 63 permissive to block the recirculation gate of the selected register, and FS101 goes low to alert the input gate of the selected register. The flip-flop is initially restored by a high RESIR signal generated by gate 20 (drawing 1-5A) and restored on completion of the read-in operation by the R101A signal. The R101A signal is generated at gate 17 (drawing 1-2A) when the 02 or 07 instruction is cleared from the static register.

3-19. INDEX-REGISTER ADDER

The index-register adder and its gates (drawing 1-25A) are used during the 07 instruction or the address-modification phase of the staticize step but not during the 02 instruction. The adder adds the information from register C on the S lines to the information from the selected index register on the M lines with the sum emerging on the IRA1 through IRA4 lines. Before the two numbers reach the adder, the number on the S lines is sent through the complementer circuit (drawing 1-13A and heading 3-70) and the number on the M lines is sent through the decimal-carry adder (drawing 1-13A and heading 3-71). The addition process is the same as that described in section 4.

The sum on the IRA1 through IRA4 lines is present at index-register-adder gates 1 through 12. The p5 and p6 digits, which are to be stored as they come from the adder, are sent directly back to register C on the IRC1 through IRC4 lines. The p7 and p8 digits, which may require modification before they are stored, are returned to the S lines on the IRC5 through IRC8 lines. These same digits are returned to the adder, emerge on the IRA1 through IRA4 lines, and are then returned to register C for storage with the p5 and p6 digits. Gates 9 through 12 are made permissive to these p7 and p8 digits which enter register C on the IRC5A through IRC8A lines.

3-20. ARITHMETIC UNIT

The arithmetic unit consists of the logical circuits which perform the computing functions of the system. Components of the arithmetic unit also process all the non-computing Processor instructions and some of the input-output instructions.

3-21. REGISTER A

Register A and the two other arithmetic registers, L and X, are circulating registers. The logical operation of a circulating register is described under heading 2-37. Register A (drawing 1-8A) stores the ten decimal digits of a computer word in four 12-bit subregisters. The gates and buffers that control input to and output from register A are described in the following paragraphs.

3-22. INPUT GATES. The input gates (46—49), when alerted by FS77, permit information from the M or S buffers to enter register A. When the outputs of the M buffers are to be read into register A, a blocking signal is applied to the S buffers which keeps the S inputs to register A low so that the M inputs can enter. In other instructions the M buffers are similarly blocked to enable the S-buffer outputs to enter the input gates of register A. The input gates are blocked during an index-register (07) instruction by a high signal from gate 118. The m portion of register A is filled with the sum of the index-register addition; the remainder of register A cleared to 0's.

3-23. LEFT-SHIFT GATES. During either a left-shift instruction or a left-shift in division, lengthening the recirculation loop shifts left the contents of register A. The lengthened recirculation path for the first three bits of each digit is through the S buffers (drawing 1-7A) and the complementer circuit (drawing 1-13A). The outputs of the complementer are the S1C, S2C, and S3C signals which go to the left-shift gates of register A. Because the complementer circuit can accommodate only three inputs, the recirculation path for the fourth bit of each digit is lengthened within the register and becomes the A″4 and $\overline{A''4}$ signals, which go to the left-shift gates 41 and 42 at the same time as the complementer outputs. Alerting signals, when present, make the left-shift gates permissive to the left-shifted contents of register A.

3-24. RIGHT-SHIFT GATES. During either a right-shift instruction or a right-shift in multiplication, shortening the recirculation path shifts right the contents of register A. This is done by blocking the normal recirculation path and reading out of register A one pulse time early. These early outputs, A′1, A′2, A′3, and A′4, return to register A through right-shift gates 37 to 40. Function signal 59 alerts the gates to the early output of the register.

3-25. QUOTIENT-INPUT GATES. In the division instruction, the quotient digits are computed in the multiplier/quotient counter (MQC) (heading 3-81) and stored in register X. In the final step of division (D3), the quotient is transferred from register X, through the multiplier-quotient counter, to register A by way of quotient-input gates 33 through 36. Function signal 32 alerts these gates to the quotient-digit signals Q1 through Q4. The $\overline{Q1}$ and $\overline{Q3}$ on MQC outputs are present at gates 34 and 36 to block division sentinel 0101.

3-26. MULTIPLIER-SENTINEL GATE. In the first step of multiplication a sentinel is inserted automatically into register A. The sentinel stops the process after the multiplicand has been multiplied by the last multiplier digit. Function signal 13 alerts multiplier-sentinel gate 31 which jams 1's into the A1 and A3 subregisters to store a bit combination of 0101, the multiplier sentinel, in register A.

3-27. LSD COMPLEMENTER GATE. Alerted by division-control signal OR, the least-significant-digit (LSD) complementer gate 30 complements the digit contained in the space-between-words (SBW) position of register X during division. The complemented digit becomes the LSD of register A. The contents of the SBW position of register X can be either 0 or a 9 in the division process. The $\overline{\text{X4D}}$ signal, an output of the most-significant-bit (X4) subregister, indicates whether the space-between-words digit is an 0 or a 9. If a 9 (1100) occupies the SBW position of register X, the $\overline{\text{X4D}}$ signal from register X is high. The LSD complementer complements the signal to a low and sends a 0 combination (0000) into register A. If a 0 (0000) occupies the SBW position of register X, the $\overline{\text{X4D}}$ signal from the register is low. The LSD complementer complements the signal to a high, placing an 1100 (9) combination into register A. (Refer to section 4.)

3-28. RECIRCULATION GATES. The recirculating signals of the four subregisters of register A are A1, A2, A3, and A4. These signals go from the last amplifier of the subregisters into recirculation gates 25 to 28, where they reenter the register. The recirculation gates are normally permissive to the recirculating signals except when FS55+ is high to block the recirculation gates clearing register A. Another input to the recirculation gates is the AR6 signal from the zero-suppress gate.

3-29. ZERO-SUPPRESS GATE. When FS38A is present during a zero-suppress instruction, zero-suppress gate 32 samples the Z1 through Z10 outputs of register X. If any of the Z signals are high, the gate permits the normal recirculation of the register to continue. When all Z signals become low, however, the zero-suppress gate places a high AR6 blocking signal on the recirculation gates. A high AR6A signal is also generated to place 1 bits in the A2 and A3 subregisters to create the code (00 0110) for space to suppress 0's.

3-30. CIRCULAR-SHIFT GATES. These gates (50, 61, 87, and 96) function on right-circular shift and divide instructions to transfer the shifted outputs of register X into register A. The gates, alerted by FS22, are permissive to the X'1 through X'4 early outputs of register X.

3-31. SUM-INPUT BUFFERS. During the arithmetic instructions, the sum of any addition is present on the O lines from the binary and quinary adders (headings 3-60 and 3-72). In these instructions, the sum of additions is transferred from the quinary- and the binary-adder gates to the sum-input buffers of register A (800, 63, 89, and 98).

3-32. OUTPUT-DISPLAY GATES. Use of the REGISTER SELECTOR on the operator's control panel enables the operator to type into or view the contents of a register. If the operator selects register A by depressing REGISTER SELECTOR button A, the RSA control signal is generated. This signal makes the four output-display gates of register A permissive to the contents of the four subregisters. The resulting A1-OD, A2-OD, A3-OD, and A4-OD signals are sent to buffers shown on the drawing for register C (drawing 1-1A). From register C, the contents of register A go to the control-panel display circuits.

3-33. OUTPUT CIRCUITS. Six outputs of register A are common to all four subregisters. All six outputs normally read out of the register to various circuits. Whether these signals are used at their destinations depends on the presence of controlling signals. For example, A1M through A4M outputs always read out of register A to the M-buffer-input gates. If the gates are permissive to the signals, the contents of the register are allowed to enter the M buffers. If the recirculation gates are blocked, however, the A1M through A4M signals continue to go to the M-buffer-input gates but are blocked from entering the M buffers. The A1 through A4 output signals go to the recirculation gates. The A'1 through A'4 signals go to the right-shift gates of registers A and X. The A11, A21, A31, and A41 outputs normally go to the S buffers. However, during a number of instructions, FS58+ is high. This signal blocks the outputs to the S buffers so that the buffers can be used for other purposes. The A"4 and $\overline{\text{A'4}}$ output signals, generated only by the fourth subregister, enter the left-shift gates during a left-shift instruction. The A1A through A4A outputs and the $\overline{\text{A1}}$ through $\overline{\text{A4}}$ outputs are used during translation.

3-34. REGISTER X

Register X (drawing 1-10A) stores the ten decimal digits of a computer word in four 12-bit subregisters. The 13 gates described in the following sections control input to and output from register X.

3-35. INPUT GATES. The input gates (19A to 19D) of register X receive the outputs of the M buffers when alerted by FS76. During the PRY step of the print instruction, FS42+ blocks input gates 19C and 19D to place 0 bits in subregisters X3 and X4. At the same time, FS76 makes gates 19A and 19B permissive to the primed part of the print word from storage.

3-36. RIGHT-SHIFT GATES. The right-shift gates (18A to 18D) of register X operate in the same manner as the right-shift gates of register A. During either a right-shift instruction or a right shift in multiplication, the outputs of register X (X'1 through X'4) are returned to register X one pulse time earlier than in normal recirculation. Function signal 59 alerts the gates to the early output.

3-37. ZERO-SUPPRESS INSERT-ONES GATE

During the first step of the zero-suppress instruction, gate 21 (alerted by FS36) inserts a 1 bit into subregister X3 for every digit position (except at t11B) of the word to be suppressed. The zero-suppress and comma-suppress gates override the inserted 1 bit if the input combination from registers A and X is a 0 or a comma. If the combination is other than a 0 or comma, the inserted 1 bits indicate the presence of a legitimate punching or printing digit.

3-38. ZERO- AND COMMA-SUPPRESS GATES

During the first step of a zero-suppress instruction, gate 5 (alerted by FS36) is permissive to the USS-6 bit combination for 0 (00 0000). Gate 6, also alerted by FS36, is permissive to the USS-6 bit combination for comma (11 0101). The word to be suppressed is stored in registers A and X with the four bits from register A on the S lines and the two remaining bits of each digit from register X on the X lines. These gates generate a high signal which overrides the low signal from the insert-ones gate at buffer 205.

3-39. RECIRCULATION GATES. The recirculating signals of the four subregisters of register X are X1, X2, X3, and X4. These signals are the outputs of the last amplifier of the subregisters and they reenter register X at the recirculation gates (20A to 20D). The recirculation gates are normally permissive to the recirculating signals. During a number of instructions, however, FS56+ blocks the path of recirculation, clearing the register. During the second step of the zero-suppress instruction FS37+ blocks recirculation of the X3 bit to clear subregister X3 of the 1 bits which were inserted during the first step, and AR6 inserts 0 bits into subregisters X1 and X2 to create those bits of the space code which will suppress commas and 0's. Function signal 15+ clears subregisters X2 and X4 during the first step of divide. The other two registers are not cleared because they may contain the 1 bits of a programmed sentinel.

3-40. QUOTIENT-COMPLEMENTING GATES. When alerted by an OR-control signal, the quotient-complementing gates (10A, 10B, 11, 12, 13, 17) complement the division sentinel and quotient digits in the divide instruction. The Q1 through Q4 and $\overline{Q1}$ through $\overline{Q4}$ signals represent the quotient digits which are computed in the multiplier/quotient counter (heading 3-81 and section 4).

3-41. DIVISION-SENTINEL GATE. In the first step of the divide instruction, FS15 alerts division-sentinel gate 9, which jams 1's into subregisters X1 and X3. This places the division-sentinel combination, 0101, into the least-significant-digit position of register X. Later, this sentinel, or a programmed sentinel, will signal the end of division. (Refer to section 4.)

3-42. REMAINDER-INPUT GATES. In the third step of the divide instruction (D3), the final quotient in register A and the final remainder in register X must be interchanged. The contents of register A normally read onto the S lines. Function signal 32 alerts the remainder-input gates (8A to 8D). The remainder, which was stored in register A, is transferred on the S lines to register X.

3-43. CHECK-BIT STORAGE GATES. In the print instruction, the six bits of a digit are transferred from main storage, through registers A and X, to the print-buffer band. Four of the six bits are sent through register A. two through register X. The check bit for the unprimed four bits sent through register A is stored in subregister X4 or register X. Function signal 43 alerts check-bit storage gate 22 to the DM5 check-bit signal

3-44. CHECK-BIT COMPUTER GATES. The check-bit computer gates 23 to 27 operate during the print instruction to sample the primed part of the digit coming from storage on the M1 and M2 lines and the check bit of the unprimed part of the digit already stored in register X, indicated by the X4 signal. The gates, alerted by FS65, compute a single check bit for the two parts of the digit, primed and unprimed. If any of these four gates operates, a 1 bit is stored in the fourth subregister of register X because the digit contains an even number of 1 bits and requires a 1 bit to make it odd. If none of the gates operate, a 0 bit is stored because the digit contains an odd number of 1 bits.

3-45. CIRCULAR-SHIFT GATES. During either a right-shift or a divide instruction, a circular shift takes place between registers A and X. The output of register A goes to register X through the circular-shift gates of register X, and the output of register X goes to register A through the circular-shift gates of register A. Function signal 59 alerts gates 3A to 3D to the A'1 through A'4 outputs of register A.

3-46. OUTPUT-DISPLAY GATES. The output-display gates of register X function in the same way as those of register A. When the operator selects register X by depressing REGISTER SELECTOR button X, the RSX control signal is generated. This signal makes the output-display gates permissive to the contents of register X. The outputs of the four gates, signals X1OD, X2OD, X3OD, and X4OD, go to the register-display circuit shown on drawing 1-1A. From register C, the contents of register X go to the control-panel display circuits.

3-47. OUTPUT CIRCUITS. Six types of outputs of register X are common to the four subregisters. Signals X1M through X4M go to the M buffers (drawing 1-6A). Signals X'1 through X'4 go to the right-shift gates of registers A and X and also to the M buffers. The recirculating signals X1 through X4, in addition to returning to the recirculation gates of register X, go to the MQC (drawing 1-16A) during the multiply and divide instructions. These signals go to the input gates of the MQC flip-flops. In division, the MQC provides a one-pulse delay for shifting left the contents of register X. In multiplication, the MQC stores the digits of the multiplier. (Refer to headings 4-30 and 4-35.) Z1 through Z10 outputs of subregister X3 are used in the zero-suppress instruction and go to the zero-suppress gate of register A. The X1A, X2A, and X4A signals and the $\overline{X1}$ through $\overline{X4}$ signals are used during translation.

In the print operation, subregister X4 contains the check bit for the part of a digit contained in register A. The $\overline{X4}$ and X4 outputs of subregister X4 are sampled by the check-bit computer gates in the computing of a new check bit. In operations other than printing, X4 is also a normal recirculating signal. Another output of subregister X4 is the $\overline{X4D}$ signal, which goes to the LSD complementer gate of register A during a divide instruction.

3-48. REGISTER L

Each of the four subregisters of register L (drawing 1-12A) has three input gates and two output gates. The five gates controlling input to and output from register L are described in the following paragraphs.

3-49. RECIRCULATION GATES. The upper gate of each subregister is the recirculation gate. Information normally recirculates from the output of the subregister into the recirculation gates (1A to 1D) on the L'1 through L'4 lines. During most instructions, FS57+ is low, making the gate permissive to the recirculating signals. During the storage-to-register L and register A-to-register L transfer instruction, however, FS57+ goes high, blocking the gate. This clears the register because there is no path for normal readout of the contents of the register during these instructions. The subregisters of register L are cleared so that new information can enter through either the M or S input gates.

3-50. M GATES. The outputs of the M buffers on the MT lines enter the M-input gates (2A to 2D) during the storage-to-register L transfer instruction. The M gates are shown as the center gate of each subregister in drawing 1-12A. Function signal 8 makes the M gates permissive to information on the MT lines (MT1 to MT4). At the end of the instruction, blocking signal 57+ is removed from the recirculation gates, and the information begins to recirculate.

3-51. S GATES. The outputs of the S buffers on the S lines enter the S-input gates (3A to 3D) during the register A-to-register L transfer instruction. The S gates are shown as the lower gate of each subregister in drawing 1-12A. Function signal 16 alerts the S gates to information on the S lines (S1 to S4). At the end of the instruction, the recirculation path is opened as the blocking signal goes low, and the information begins to recirculate.

3-52. OUTPUT-DISPLAY GATES. When the operator depresses the REGISTER SELECTOR for register L, control signal RSL is generated. This signal makes the output-display gates of register L permissive to the contents of the register. Output signals L1OD, L2OD, L3OD, and L4OD go to the output-display circuit shown on register C, which controls the control-panel display circuit.

3-53. OUTPUT GATES. The output gates (4A to 4D) are permissive to the contents of register L on the multiply, divide. and comparison instructions. Alerted by FS66 on these instructions, the outputs of the gates are the L1 through L4 signals. These four signals, which represent the contents of the four subregisters of register L, go to the M buffers.

In the second step of division, control signal OR+ is high, blocking the output gates even though FS66 is present. As soon as signal OR+ goes low, the gates are permissive to the contents of the register.

3-54. OUTPUTS. Signals L1M and L4M are normal outputs of the register to the M buffers. The M buffers, however, are permissive to these signals only during a register L-to-storage transfer instruction. At all other times, the input gates of the M buffers are not permissive to the signals from register L. The $\overline{L'1}$ through $\overline{L'3}$ signals and the L1A through L4A signals are used during translate.

3-55. 80-COLUMN TO USS-6 TRANSLATOR GATES

The 80-column to USS-6 translator gates translate the 80-column card code into the USS-6 machine and printer codes. (Refer to table 3-2.) The 12 bits of the card code are stored as follows: The four unprimed bits are stored in register A, the four primed bits in register L, and the four double-primed bits in register X. On a translate instruction, the code combination on the A, L, and X lines goes to the translator gates of register A (gates 7 through 22, drawing 1-9A) and register X (gates 1 through 6 and 23, drawing 1-9A). These gates, alerted by FS17, convert the 12-bit code to six-bit code. After translation, the four numeric bits are stored in register A and the two zone bits in register X.

3-56. USS-6 TO 80-COLUMN TRANSLATOR GATES

The USS-6 to 80-column translator gates translate the USS-6 internal and printing code into the 80-column card code. The six bits of the USS-6 code are stored, the four numeric bits in register A and the two zone bits in register X. On a translate instruction, the code combination on the A and X lines goes to the translator gates of register A (gates 1 through 7, 18, 29, and 30, drawing 1-9A), register X (gates 8 through 17, drawing 1-9A) and register L (gates 22 through 28, drawing 1-9A). These gates, alerted by FS18, convert the six-bit code to 12-bit code. The four unprimed bits are stored in register A, the four primed bits in register L, and the four double-primed bits in register X.

Table 3-3 illustrates the translation of all numerics, alphabetics, and special symbols. The lower-right block of the table is a sample block showing the contents of all other blocks. Note that the character appears on the top line of the block; the gates which perform the translation appear in parentheses on the middle line of the block; and the bit combination in 80-column card code appears on the bottom line. The first four bits on the bottom line is the combination found in register A; the second four bits is the combination found in register L, and the third four bits is the combination found in register X. Note that the forbidden combinations (for example, the combinations for which no character or symbol is available) are not blocked but pass through the translate gates with results which are not meaningful.

3-57. S BUFFERS

The S buffers (drawing 1-7A) transfer the outputs of registers A and C to various circuits in the Processor. The output signals of the two registers are ungated inputs to the buffers. The outputs of the buffers are the S1 through S4 and $\overline{S1}$ through $\overline{S4}$ signals, which are referred to as the S lines. The S1 through S4 unbarred outputs are forced low during the divide, storage-to-register A transfer, print, and superimpose instructions by high FS82A. Of these instructions, only the unbarred (noncomplemented) outputs are used so that when 82A is high, the S lines are low. This enables the M lines, which are not forced low, to enter the input gates of register A.

Table 3-2. Translating from 80-Column Card Code to USS-6 Code

USS-6 → ↓ rA	00(rX)	01(rX)	10(rX)	11(rX)
0111	) (gates 7 and 16) 0001 0010 0010	& (gates 4 and 18) 1000 0000 0000	X	X
0110	Space (gate 20) 0000 0000 0000	: (gates 4 and 16) 1000 0010 0010	* (gates 16 and 1) 0100 0010 0010	% (gates 6 and 16) 0010 0010 0010
0101	— (minus) (gate 19, gate 23 to block 1) 0100 0000 0000	. (gates 4 and 17) 1000 0100 0010	$ (gates 1 and 17) 0100 0100 0010	, (comma) (gates 6 and 17) 0010 0100 0010
0000	0 (gate 23 to block 6) 0010 0000 0000	Nonprinting (gate 4) 1010 0000 0000	Nonprinting (gate 2, gate 23 to block 1) 0110 0000 0000	+ (plus) (gate 3, gate 21 to block 16) 0000 0010 0010
0001	1 (gate 7) 0001 0000 0000	A (gates 7 and 4) 1001 0000 0000	J (gates 7 and 1) 0101 0000 0000	/ (gates 6 and 7) 0011 0000 0000
0010	2 (gate 8) 0000 1000 0000	B (gates 8 and 4) 1000 1000 0000	K (gates 8 and 1) 0100 1000 0000	S (gates 8 and 6) 0010 1000 0000
0011	3 (gate 9) 0000 0100 0000	C (gates 9 and 4) 1000 0100 0000	L (gates 9 and 1) 0100 0100 0000	T (gates 9 and 6) 0010 0100 0000
0100	4 (gate 10) 0000 0010 0000	D (gates 10 and 4) 1000 0010 0000	M (gates 10 and 1) 0100 0010 0000	U (gates 10 and 6) 0010 0010 0000
1000	5 (gate 11) 0000 0001 0000	E (gates 11 and 4) 1000 0001 0000	N (gates 11 and 1) 0100 0001 0000	V (gates 11 and 6) 0010 0001 0000
1001	6 (gate 12) 0000 0000 1000	F (gates 12 and 4) 1000 0000 1000	O (gates 12 and 1) 0100 0000 1000	W (gates 12 and 6) 0010 0000 1000
1010	7 (gate 13) 0000 0000 0100	G (gates 13 and 4) 1000 0000 0100	P (gates 13 and 1) 0100 0000 0100	X (gates 13 and 6) 0010 0000 0100
1011	8 (gate 14) 0000 0000 0010	H (gates 14 and 4) 1000 0000 0010	Q (gates 14 and 1) 0100 0000 0010	Y (gates 14 and 6) 0010 0000 0010
1100	9 (gate 15) 0000 0000 0001	I (gates 15 and 4) 1000 0000 0001	R (gates 15 and 1) 0100 0000 0001	Z (gates 15 and 6) 0010 0000 0001
1111	' (apost) (gates 16 and 12) 0000 0010 1010	# (gate 5, gate 21 to block 17) 0000 0100 0010	X	X
1110	; (gates 16 and 11) 0000 0011 0010	X	X	X
1101	((gates 17 and 11) 0000 0101 0010	X	X	Character (gates) 80-column code (rA) (rL) (rX)

X = forbidden combinations

Table 3-3. Translating from USS-6 Code to 80-Column Card Code

USS-6 → ↓ rA	00(rX)	01(rX)	10(rX)	11(rX)
0111	) (gates 1 and 16) 0001 0010 0010	& (gate 18) 1000 0000 0000	X	X
0110	Space (gate 10 to block 14) 0000 0000 0000	: (gates 18 and 14) 1000 0010 0010	* (gates 7 and 14) 0100 0010 0010	% (gates 5 and 14) 0010 0010 0010
0101	— (minus) (gate 10 to block 11, and gate 6) 0100 0000 0000	. (gates 18 and 11) 1000 0100 0010	$ (gates 7 and 11) 0100 0100 0010	, (comma) (gates 5 and 11) 0010 0100 0010
0000	0 (gates 3 and 4) 0010 0000 0000	Nonprinting (gates 18 and 4) 1010 0000 0000	Nonprinting (gates 7 and 3) 0110 0000 0000	+ (plus) (gate 17, gate 30 to block 5) 0000 0010 0010
0001	1 (gate 2) 0001 0000 0000	A (gates 18 and 2) 1001 0000 0000	J (gates 7 and 2) 0101 0000 0000	/ (gates 5 and 2) 0011 0000 0000
0010	2 (gate 28) 0000 1000 0000	B (gates 18 and 28) 1000 1000 0000	K (gates 7 and 28) 0100 1000 0000	S (gates 5 and 28) 0010 1000 0000
0011	3 (gate 27) 0000 0100 0000	C (gates 18 and 27) 1000 0100 0000	L (gates 7 and 27) 0100 0100 0000	T (gates 5 and 27) 0010 0100 0000
0100	4 (gate 25) 0000 0010 0000	D (gates 18 and 25) 1000 0010 0000	M (gates 7 and 25) 0100 0010 0000	U (gates 5 and 25) 0010 0010 0000
1000	5 (gate 22) 0000 0001 0000	E (gates 18 and 22) 1000 0001 0000	N (gates 7 and 22) 0100 0001 0000	V (gates 5 and 22) 0010 0001 0000
1001	6 (gate 8) 0000 0000 1000	F (gates 18 and 8) 1000 0000 1000	O (gates 7 and 8) 0100 0000 1000	W (gates 5 and 8) 0010 0000 1000
1010	7 (gate 9) 0000 0000 0100	G (gates 18 and 9) 1000 0000 0100	P (gates 7 and 9) 0100 0000 0100	X (gates 5 and 9) 0010 0000 0100
1011	8 (gate 13) 0000 0000 0010	H (gates 18 and 13) 1000 0000 0010	Q (gates 7 and 13) 0100 0000 0010	Y (gates 5 and 13) 0010 0000 0010
1100	9 (gate 12) 0000 0000 0001	I (gates 18 and 12) 1000 0000 0001	R (gates 7 and 12) 0100 0000 0001	Z (gates 5 and 12) 0010 0000 0001
1111	' (apost) (gates 15 and 16) 0000 0010 1010	# (gates 13 and 26, gate 29 to block 18) 0000 0100 0010	X	X
1110	; (gates 14 and 23) 0000 0011 0010	X	X	X
1101	((gates 11 and 24) 0000 0101 0010	X	X	Character (gates) 80-column card code (rA) (rL) (rX)

X = forbidden combinations not blocked.

GEN 77 forces the unbarred outputs low during an 81 (FP), 46 (PBM), or 96 (RBM) instruction with translation. The high GEN 77 signal is generated by a reading setup in the MQC when either an output word from a main-storage location or an input word from a buffer location is on the M lines and is to be stored in register A. The high GEN 77 signal produces function signals to accomplish this transfer. Because the input gates to register A are alerted to information on both the M and S lines, the S lines must be held low to permit only the word on the M lines to enter.

If index registers are included, information on the BC1—BC4 lines from register C and information on the IRC5—IRC8 lines from the index-register adder is gated onto the S lines.

3-58. ADD-9800 FLIP-FLOP (OPTIONAL WITH INDEX REGISTERS)

The add-9800 flip-flop is used with index registers to ensure that the modified m address is kept within the same bands as the original m address. Function signal 100 alerts the set gates of this flip-flop if the index-register read-out flip-flop (drawing 1-24A) is set. The flip-flop is set if the addition of the contents of the index register to the original m address from register C results in a modified m address in the next band. To bring the modified m address back into the original band, it is necessary to subtract 200 from the modified m address. The same result is achieved by addition instead of subtraction, by adding 9800 to the modified m address.

The modified m address is gated onto the S lines as it leaves the index-register adder. If 9800 is to be added, the add-9800 flip-flop is set, generating a high BJ7 signal at t6A and a high BJ9 signal at t7A. The BJ7 and BJ9 signals are sent to the M buffers (drawing 1-6A) with the BJ7 signal jamming the code combination for 7 (1010) and the BJ9 signal jamming the code combination for 9 (1100) onto the M lines. This places the code for 9800 onto the M lines. The 1 needed to change the number to 9800 is produced by sending the BJ7 signal into the initial-force-decimal-carry circuit (drawing 1-14A). The modified m address on the S lines is added to the 9800 on the M lines in the index-register adder, and the sum is returned to register C.

3-59. M BUFFERS

The M buffers (drawing 1-6A) normally transfer the contents of register L and the storage drum to other Processor circuits. The signals from register L, L1 through L4, and the DM-memory outputs are ungated inputs to the buffers. Therefore, they are transferred to the M lines whenever they are present at the M buffer. All other inputs to the M buffers are from the input gates, which are permissive during the instructions listed for each gate in drawing 1-6A. A permissive gate allows information signals from various computer elements to enter the M buffers and the M lines. The MT lines perform the same function as the M lines. During the input-output test instruction, FS200 and FS27A are high inputs to the buffers. They force the unbarred-M outputs low so that the S lines can read into the input gates of register A. If index registers are included, the B lines from the index registers and the BJ7 and BJ9 signals from the add-9800 flip-flop are ungated inputs.

3-60. COMPARATOR

The circuits of the comparator (drawing 1-14A) perform four major functions: comparison, addition of binary bits, determination of carry, and control. The comparator performs these functions in the arithmetic and comparison instructions and in the search for instructions or operands in the memory. Because certain components of the comparator operate differently for each of the four functions, the description of each component includes its functions. (In the computer code, the three lowest-order bits of a digit are the quinary bits. The most significant is the binary bit. References made in the following paragraphs to the quinary and binary bits pertain to this order of significance.)

3-61. QUINARY-EQUALITY CIRCUIT. In the search for either an instruction or an operand from storage, the code combinations for the many addresses on the storage drum are transferred serially to the M buffer and the M lines. The address of the instruction or operand being searched for is stored in register C and also reads into the S buffers and the S lines. The quinary-equality circuit compares the quinary bits of the addresses on the M lines with the quinary bits of the address on the S lines. When the circuit finds equality between the two lowest-order digits of the two addresses, and the correct band is chosen (heading 3-105), the desired storage address has been located and the instruction or operand contained in that location can be read out. The input gates of the circuit are controlled by the A, C, and CP control signals, which are generated during any function involving the quinary-equality circuit, to make the gates permissive. When all the M and S inputs to a quinary-equality gate are low, a condition which can occur only when the inputs are equal, a high K signal is generated. The K signal generates low signal EQ and high signal $\overline{EQ}$, both of which indicate equality of the input quinary bits. The $\overline{EQ}$ signal controls the time-selection (TS) flip-flop. The EQ signal controls the conditional-transfer (CT) flip-flop.

In the conditional-transfer (87) instruction, the contents of registers L and A are present at the M and S inputs, respectively. Information in the two registers is compared in the comparator to determine whether the contents of register A are greater than the contents of register L. The presence of low signals A and C at the input gates indicates that the contents of register A are not greater than the contents of register L. Control signals A, C, and CP alert the quinary-equality-input gates to the M and S inputs. Once again, equality at any one of the gates produces a high K signal, which in turn produces high $\overline{C}$ and low C signals. The $\overline{C}$ and C signals control gates 13 and 14 of the CT flip-flop.

In the conditional-transfer (82) instruction, the gates test for equality in the same manner as the 87 instruction.

However, the A, C, and CP signals act only as alerting signals in the 82 instruction.

During the subtraction process, the quinary-equality gates compare the minuend and subtrahend. If the two are equal, the result of the subtraction, (0), must be given a plus sign (heading 3-74). The CP signal, which indicates that subtraction is taking place, and the A and C signals alert the gates to the M and S inputs. Equality produces the K signal and also the high $\overline{\text{EQ}}$ signal which keeps the TS flip-flop from being restored. If the inputs to the gates are not equal, the $\overline{\text{EQ}}$ signal becomes a low signal which restores the TS flip-flop.

3-62. QUINARY-CARRY CIRCUIT. The quinary-carry circuit consists of all the comparator gates that generate R signals and the three output circuits which generate C, $\overline{\text{C}}$, and C′.

The quinary-carry gates alerted by the $\overline{\text{CP}}$ signal from the complement (CP) flip-flop (heading 3-70) are used during the four arithmetic instructions. In the addition process for any of these instructions, the quinary bits of the two digits being added go to the quinary-carry gates. When the addition of quinary bits results in a carry, the indication of carry is sent to the binary-addition circuits, to be introduced into the addition of the binary bits. Similarly, decimal carry from a previous addition is indicated to the quinary-carry gates by the presence of both A and C signals. In the arithmetic instructions, the operand from memory is on the M lines and the contents of register A on the S lines. The quinary-carry gates alerted by the $\overline{\text{CP}}$ signal sample specific bits of the two digits being added. For instance, gate 45 samples the M3 and S1 bits. If both bits are 1's, the quinary combination of the two digits is 1XX XX1. A 1 in the M3-bit position indicates that one of the digits being added is a 4 or greater. A 1 in the S1-bit position indicates that the other digit is a 1 or greater. In adding the digits 1 and 4, a carry to the binary bits occurs. Gate 45, then, generates a high R signal, which in turn generates low signals C and C′ and high signal $\overline{\text{C}}$. The C′ and $\overline{\text{C}}$ signals go to the binary adder; the $\overline{\text{C}}$ signal, together with the C signal, also goes to the decimal-carry adder.

During a subtraction process (addition with unlike signs or subtraction with like signs) the group of quinary-carry gates alerted by the CP signal is used to sample for quinary carry. In the subtraction process, the contents of register A on the S lines are complemented in the complementer gates of the quinary adder (heading 3-72). In parallel, the S lines and the M lines go to the quinary-carry gates. One of the gates alerted by the CP signal generates a K signal when the combination of M and S inputs is low. Control signal CP indicates that the S-line inputs are presently being complemented (9's complement). The quinary-carry gates for subtraction differ from those for addition in that the 9's complement of the S inputs to these gates must be added to the M inputs to determine quinary carry. To illustrate this, the combination at gate 46 is: M=001X, S=X00X. Referring to the computer code, the digit on the S lines can be any digit with 0 in the S2- and S3-bit position. Four possibilities exist for this combination: 0, 1, 5, and 6. Because the S bits are being complemented in the quinary adder, the same S bits at the quinary-carry gates represent the complemented version of the S digit. The 9's complement of 0, 1, 5, and 6 is 9, 8, 4, and 3 respectively. Following are additions of the quinary bits of the four possible S combinations, after complementing, to the quinary bits of the M combination, at gate 46:

```
M:X/01X=(2)        M:X/01X=(2)
  S:1/100=(4)        S:1/011=(3)
        ----               ----
   sum   6           sum    5
M:X/01X=(2)        M:X/01X=(2)
  S:0/011=(3)        S:0/100=(4)
        ----               ----
   sum   5           sum    6
```

The digit on the M lines is described above as a 2 because, with a 1 in the M2-bit position, the digit can be no less than a 2. If the addition of 2 to the S digit requires a quinary carry, it follows that any digit larger than 2 also requires a quinary carry. The first quinary addition is of the S combination for 9 (the 9's complement of 0) to the M bits. Only the quinary bits of the digit are sampled; therefore, in quinary code a 2 is added to a 4. The sum is 6 and, because any digit larger than 4 necessitates a quinary carry to the binary addition, a quinary carry is indicated. This holds true for the other three possible S combinations, all of which result in a quinary digit larger than 4. Any such combination present at the quinary-carry gates during subtraction causes a C signal to be generated. The number in parentheses in each addition is the digit formed by the quinary bits.

Gate 51 of the quinary-carry circuit is the only gate that operates in both the addition and subtraction processes. This gate is permissive to a combination of M and S inputs, common to both addition and subtraction. In either process, the presence of this combination generates an R signal.

The group of quinary-carry gates, alerted by the CP signal, is used also during the comparison (87) instruction to compare the quantities in registers A and L. If the contents of register L, on the M lines, are greater than the contents of register A, on the S lines, a high R signal is generated. The R signal generates low C and high $\overline{\text{C}}$ signals to control the CT flip-flop. Some of the comparisons made by these gates are of non-numeric combinations (denoted by the alpha symbol). The combinations 0101, 0110, and 0111 are less than 0; the combinations 1101, 1110, and 1111 are greater than 9. Non-numeric symbols in card code also can be compared during the 87 instruction by these gates.

3-63. FORCE-DECIMAL-CARRY GATES. The comparator gates labeled force-decimal carry generate the two indications of decimal carry, A and C, when the correct controlling signals are applied.

On the comparision (87) instruction, FS20 alerts gate 32 to the A+ signal. The A+ signal comes from the rA-sign flip-flop (heading 3-73) and indicates that the sign of the contents of register A is a plus. If the A+ signal is low, it becomes a high signal output of the amplifier and is sent to the binary-carry and quinary-carry circuits to generate the A and C signals. In this instruction, the A and C signals are used to alert the quinary-equality circuit. If the A+ signal is high, the A and C signals are not generated, and the quinary-equality circuit is blocked.

Gate 31 also operates during the 87 instruction to compare the binary bits of information from registers L and A. If the $\overline{S4}$ bit from register A and the M4 bit from register L are present, the digit in register L is greater than the digit in register A. The A and C signals are generated by the gate to control the CT flip-flop (heading 3-68). The CP signal, present during this instruction, alerts the gate.

Gates 30 and 31 of the force-decimal-carry circuit operate during arithmetic instructions. They generate the A and C signals to indicate decimal carry regardless of whether the quinary-carry signal, C, is present. In adding certain digits, addition of the quinary bits generates no quinary carry to the binary adder. However, in many such additions, a decimal carry to the next digit is necessary. To ensure that the A and C signals are present at the decimal-carry adder (heading 3-71), the force-decimal-carry gates generate both A and C to add a carry of 1 whenever the binary bits show a decimal carry. When both the M4 and S4 bits are 1's, as when either gate is permissive, a decimal carry is generated. The CP- and $\overline{CP}$-control signals alert the gates. The CP signal indicates that the $\overline{S4}$ bit is being complemented in the quinary-adder complementer gates and will therefore become a 1 bit. The same two gates also operate as part of the binary-equality gates during the search operation.

3-64. BINARY-EQUALITY GATES. The binary-equality gates compare the binary bits of digits on the M and S lines during search and other instructions.

In the search for an instruction or an operand, the code combinations for the storage addresses are on the M lines. The address is stored in register C and is present on the S lines. While the quinary-equality gates are comparing the quinary bits of each address digit, the binary-equality gates are comparing the binary bits of the same digit. Only the gates alerted by the CP signal operate during search operations. If the M4 and S4 bits are equal (M4 and S4 or $\overline{M4}$ and $\overline{S4}$), the binary-carry circuit generates signals which control the memory-selection circuits (drawing 1-20A), and the TS flip-flop (heading 3-69).

The binary-equality gates alerted by the CP signal are used during both the 82 and 87 comparison instructions to sample for equality of digits being compared on the M and S lines. If the bits are equal, the binary-carry circuit generates signals which control the CT flip-flop.

Any of the binary-equality gates, and gates 30 and 31, can operate during arithmetic instructions. If a subtraction process is involved, those gates alerted by the CP signal operate. If an addition process is involved, those gates alerted by the $\overline{CP}$ signal operate. In the subtraction process (CP present), the binary-equality gates send a high signal to the binary-carry and adder circuits to produce a 1 whenever the input bits are alike. When the input bits are unlike, a low signal is sent to the carry circuit and the adder to produce a 0. In the addition process (CP present), the binary-equality gates send a high signal to the carry circuit and adder to produce a 1 when the input bits are unlike. Whenever the input bits are alike the equality gates send a low signal to the carry and adder circuits to produce a 0.

3-65. INITIAL-FORCE-DECIMAL-CARRY CIRCUIT. This circuit operates during the search, add, subtract, or 82 comparison instructions. The CP flip-flop generates one of the following high signals: CP1, CP2, CP3, CP4, or CP5. These high signals go to the initial-force-decimal-carry buffer to force the outputs of the binary- and quinary-carry circuits to A and C, respectively. The A and C signals are generated initially in the instructions and search to alert various comparison gates.

If index registers are included, a high BJ7 signal from the add-9800 flip-flop enters the initial-force-decimal-carry circuit to send a carry into the index-register adder.

3-66. BINARY-CARRY CIRCUIT. The binary-carry circuit converts inputs from the initial-force-decimal-carry, force-decimal-carry, and binary-equality circuits in the A′, $\overline{A}'$, A and $\overline{A}$ binary-carry signals. The latter three signals control the memory-selection circuits and the CT flip-flop of the comparator. The A and $\overline{A}$ signals, which indicate the presence or absence of binary carry, go to the decimal-carry adder (heading 3-71). The A′ and $\overline{A}'$ signals go to the binary adder and indicate binary carry or the value of the binary bit. The signal-flow chart (drawing 1-12A) illustrates other uses of the binary-carry outputs.

3-67. BINARY-ADDER GATES. The binary-adder gates are alerted by FS50 during all the arithmetic instructions. The adder adds the outputs of the quinary- and binary-carry circuits and sends the sum to register A. The output O signal is the binary or fourth bit of the sum. The quinary bits of the sum are present on the O-output lines of the quinary adder.

3-68. CONDITIONAL-TRANSFER FLIP-FLOP. The conditional-transfer flip-flop is used primarily during the search step and the 82 and 87 comparison instructions to determine whether the computer takes the next instruction from the m or c address.

In the 82 instruction, the m address is chosen if the two words from registers A and L are equal; if they are unequal, the c address is chosen. In the 87 instruction, the m address is chosen if the contents of register A are greater than the contents of register L; if not, the c address is chosen. The CT signal controls readout of the m address from register C; the $\overline{CT}$ signal controls readout of the c address from register C.

In the staticize step of any instruction, set gate 12 of the CT flip-flop samples the $\overline{\text{STR2}}$ signal, which indicates whether the instruction involves a search for the m address. If the $\overline{\text{STR2}}$ signal is low, indicating a 0 in the STR2-bit position, the CT flip-flop is set and the CT signal is generated to control readout of the m address. After the m address is read from register C, the CT flip-flop must be restored so that the next instruction may be searched for on completion of the instruction stored in register C. The RCT1—RCT6 signals and set CC1 signals restore the flip-flop during various instructions. If the $\overline{\text{STR2}}$ signal is high, indicating a 1 in the STR2-bit position, the flip-flop is restored, generating the $\overline{\text{CT}}$ signal to control readout of the c address. Function signal 2 makes the gate permissive to the $\overline{\text{STR2}}$ signal.

At the beginning of the execution step of the 82 and 87 instructions, FS60A makes gate 1 permissive, setting the flip-flop to CT. When the CP signal is generated, the m address will be read unless comparison operations restore the flip-flop to $\overline{\text{CT}}$, thereby choosing the c address.

During any input-output test instruction, FS27 alerts set gate 11 of the flip-flop. If the jam-I2B input signal is high, indicating that an abnormal condition is present in one of the input-output units, the flip-flop is restored to $\overline{\text{CT}}$, causing the program to go to the c address for the next instruction. If the input-output units are operating normally, the jam-I2B signal is low and sets the flip-flop to generate CT and read out the m address.

During the 87 instruction, gates 13, 14, and 15 control the restore circuit of the CT flip-flop. Function signal 20 alerts all three gates which sample the sign of the contents of registers L and A (signals A— and L+) and the outputs of the binary- and quinary-carry circuits (signals A, C, $\overline{\text{A}}$, and $\overline{\text{C}}$). The A and C signals on gate 13 indicate the the A+ signal is present at the force-decimal-carry gate, 32. Table 3-4 lists the gates (drawing 1-14A) which operate to restore the CT flip-flop to $\overline{\text{CT}}$ under the various conditions in the 87 instruction. The conditions, or signals, in the first column of table 3-4 are the sign combinations possible for the contents of registers A and L. The conditions listed at the top of the table are the absolute values of the contents of both registers, as determined by the comparison circuits. The CT flip-flop remains set to generate CT under those conditions marked by X in the table.

In the 82 comparison instruction, gates 16 and 17 control the restore circuit of the CT flip-flop. Function signal 75 alerts the gates which sample the indications of equality between the contents of registers A and L. If either the $\overline{\text{EQ}}$ or $\overline{\text{A}}'$ signal is low, one of the gates sends a high signal to buffers 77 and 78 to restore the flip-flop.

The $\overline{\text{EQ}}$ signal, when low, indicates inequality of the quinary bits of the two registers. The $\overline{\text{A}}'$ signal indicates inequality of the binary bits of the two registers. If the contents of the registers are equal, the $\overline{\text{EQ}}$ and $\overline{\text{A}}'$ signals are high. The high signals block gates 16 and 17 and the CT flip-flop remains in the set condition.

During the 82 instruction, gates 18 and 19, alerted by FS75, sample the signs of registers A and L which are stored in the sign and control circuits. If the signs are unequal, A+ and L—, or A— and L+, one of the gates is permissive. The permissive gate sends a high signal to buffer 77, restoring the flip-flop. If the signs are equal, one of the inputs to both gates is high. The high signals block both gates and the flip-flop remains in the set condition.

The CT flip-flop is also controlled by the NEXT ADDRESS switch on the operator's control panel. The result of any comparisons made during the 82 and 87 instructions is indicated by the m-address and the c-address lights on the control panel. If the CT flip-flop has been set, the m lamp is lit; if it is restored, the c lamp is lit. The NEXT ADDRESS switch enables the operator to override the results of the instructions. If the m switch is energized, a high signal goes to buffer 76, which sets the flip-flop and generates CT. If the c switch is energized, a high signal goes to buffer 77, which restores the flip-flop and generates $\overline{\text{CT}}$.

3-69. TIME-SELECTION FLIP-FLOP. The time-selection (TS) flip-flop is used in controlling memory-search operations and, during the add and subtract instructions, to control the sign of the sum.

During search, the two lowest-order digits of the instruction address, p1 and p2, or operand address, p5 and p6, and the least-significant bit of the third digit, p3 or p7, are compared to address digits from the storage circuits in the comparator equality circuits. The TS flip-flop is initially set to TS, at buffer 82, before search begins. The flip-flop in the set condition indicates equality of the address digits being compared. Inequality of the digits restores the flip-flop. Function signal 1 alerts the flip-flop input gates 20 and 21 during search. These gates sample the $\overline{\text{EQ}}$ and $\overline{\text{A}}'$ outputs of the quinary- and binary-comparison circuits. If both inputs are high, indicating ~~in~~equality, the gates are blocked and the TS flip-flop remains set. If either of the signals goes low indicating inequality, the flip-flop is restored and a new search operation begins. Gate 22 operates to restore the TS flip-flop under certain conditions of the search operations. (Refer to section 4.)

Table 3-4. Conditional-Transfer Flip-Flop Gate Inputs

	A > L	A = L	A < L
A+ L+	X	Gate 13	Gate 13
A+ L—	X	X	X
A— L+	Gates 14 15	Gates ~~14~~ ~~15~~ 13	Gate ~~13~~ 14 15
A— L—	Gates 14 15	Gates 14 15	X

The CT flip-flop restore-gates 16 and 17 also control the TS flip-flop during subtraction. When two quantities involved in subtraction are equal, the sign of the 0 result must be a plus. As in the 82 instruction, gates 16 and 17 are alerted by FS75 to sample for equality of digits being compared in the binary- and quinary-equality circuits. If equality exists between the two, the gates are blocked and the TS flip-flop remains set. The TS output of the flip-flop goes to the rA-sign flip-flop to force the sign of the sum to a plus (heading 3-74). Outputs of the TS flip-flop also go to the memory-selection circuits, the static register, and the read-write circuits.

3-70. COMPLEMENTER

During arithmetic instructions the complementer circuit (drawing 1-13A) complements the quinary bits of information on the S lines. In subtraction, the minuend of the S lines must be complemented. Under such conditions the complement (CP) flip-flop is set, generating the CP-control signal. With CP present, the complementer gates generate the quinary bits of the 9's complement of the input S bits. For example, if the quinary bits of the digit 1 (001) are to be complemented, the input to the complementer circuit is $\overline{S3}$, $\overline{S2}$, S1, and CP. Gate 1 is permissive to the low S1 signal, generating low S1C. Gate 2 is blocked by the high $\overline{CP}$ signal, and gate 4 is blocked by the high $\overline{S1}$ signal. Gate 3, however, is permissive to $\overline{S2}$, S1, and CP, generating low S2C. Gate 5 is blocked by the high $\overline{CP}$ signal, and gate 6 is blocked by the $\overline{S1}$ signal. Because both gates 5 and 6 are blocked, the $\overline{S3C}$ signal is low. The outputs of the complementer, then, are $\overline{S3C}$, S2C, and S1C, or the quinary combination 011 for either a 3 or an 8. The binary circuits determine the binary bit of the combination, which in this case is a 1. Because the 9's complement of 1 is 8, the complementer and binary-adder circuits produce the biquinary-code combination for 8, which is 1011.

In addition, complementing is unnecessary; therefore, the $\overline{CP}$ signal alerts the complemeter gates so that the input bits pass unchanged through the circuits. The outputs of the complemeter circuit are low signals and are applied to the quinary-adder gates.

3-71. DECIMAL-CARRY ADDER

The decimal-carry adder (drawing 1-13A) operates during subtraction to add a 1 to the subtrahend on the M lines. The design of the computer calls for the use of the 9's complement in arithmetic operations. However, when a subtrahend is to be added to a minuend, adding the 9's complement of the minuend to the subtrahend does not produce the correct difference. Adding the 10's complement of the minuend to the subtrahend does produce the correct difference. The unit adder simulates the use of the 10's complement by adding a 1 to the subtrahend.

The A and C signals from the comparator add a 1 to the subtrahend on the M lines. During subtraction, the CP signal is generated by the complement flip-flop in the set condition and causes the initial-force-decimal-carry-circuit to generate signals A and C. These signals alert the decimal-carry-adder gates to the subtrahend bits on the M lines. Each gate is permissive to a specific input combination. For example, if the quinary combination on the M lines is 010 (2), the inputs are $\overline{M3}$, M2, $\overline{M1}$, A, and C. Gates 14, 15, and 16 are blocked to this combination, generating $\overline{M3U}$. Gate 13 is permissive to the $\overline{M1}$ and M2 signals, generating M2U. Gate 9 is also permissive, generating M1U. The output combination of the decimal-carry adder is $\overline{M3U}$, M2U, M1U. This signifies the quinary combination of 011 (3). Thus the adder has added a 1 to a quinary 2 for a quinary sum of 3.

During addition, the decimal-carry adder adds a decimal carry from a previous addition to the quantity of the M lines. At the beginning of the process, the CP flip-flop is restored and the initial-force-decimal-carry circuit generates $\overline{A}$ and $\overline{C}$. However, if a decimal carry exists from a previous addition, the carry circuits override the $\overline{A}$ and $\overline{C}$ signals to produce the A and C signals which indicate decimal carry. The decimal-carry adder adds the decimal carry of 1 to information on the M lines. The outputs of the decimal-carry adder go to the quinary adder.

3-72. QUINARY ADDER

The quinary adder consists of 18 gates, arranged in a matrix (drawing 1-13A). The circuit adds the three quinary M bits from the decimal-carry adder to the three quinary S bits from the complementer. Function signal 50 alerts the gates, one or two of which can be permissive to a single combination of low inputs. The permissive gates generate high O signals; the blocked gates generate low O signals. A high output signal indicates a 1 bit; a low output signal indicates a 0 bit. The three major outputs of the quinary adder are O1, O2, and O3 signals. These form the quinary portion of the sum of addition which is stored in register A.

3-73. SIGN-AND-CONTROL CIRCUIT

The sign-and-control circuit (drawing 1-15A) consists of three sign flip-flops, three control flip-flops, and a control circuit for display of the register signs on the operator's panel. The sign flip-flops compute and store the signs of the contents of registers A, X, and L. The control flip-flops generate signals that control various operations in the computer.

3-74. REGISTER A SIGN FLIP-FLOP. The rA sign flip-flop computes and stores the sign of the contents of register A and the sign of the result in the arithmetic instructions.

Gates 1 and 2, alerted by FS86 during multiplication and division, determine the sign of the product in multiplication and the sign of the quotient in division. The flip-flop is initially set to A— by FS13A during multiply and FS15A during divide. Gates 1 and 2 sample the signs of registers X and L from the sign flip-flops of these registers If the signs are alike, one of the gates is permissive and the flip-flop is restored to A+. If the signs are unlike, both gates are blocked and the flip-flop remains in the set condition, generating A—.

Gate 3 is alerted by FS4 during either the add or subtract instruction to compute the sign of the sum or difference. The presence of the A and C inputs to the gate indicates that the quantity in register A is equal to the word from storage. The CP signal indicates that subtraction is taking place. The A— signal is the output of the rA-sign flip-flop prior to sign computation. If gate 3 is permissive to these signals, the flip-flop is restored to A+; if not, the flip-flop remains in the set condition and stores the A— signal.

Gate 7 receives the same inputs as gate 3 except that it is permissive when the sign of the word in register A is positive and is equal to the word from memory. When the gate is made permissive, it sets the flip-flop and stores a minus sign, A—.

Gate 6 is alerted by FS4 during subtraction to force the flip-flop to A+ if the quantities being subtracted are equal. The TS signal indicates equality of the first nine digits of the two quantities; the EQ and A′ signals indicate equality of the tenth digits of the two quantities. If both quantities are equal, the gate is permissive and the flip-flop is restored to A+. This ensures that a 0 result always has a plus sign.

In a transfer of information from main storage to register A (MA instruction), the sign being transferred is received by gate 4.

The flip-flop is initially set to A— by FS6. The $\overline{\text{M1}}$ signal on the M lines represents the sign position of the word from storage. If $\overline{\text{M1}}$ is low, indicating a plus sign, the flip-flop is restored to A+; if $\overline{\text{M1}}$ is high, it blocks gate 4. Function signal 6, however, makes gate 5 permissive and the flip-flop remains in the set condition (A—).

3-75. REGISTER L SIGN FLIP-FLOP. The rL-sign flip-flop stores the sign of the contents of register L during the two transfer instructions involving the register. In both instructions, the flip-flop is set initially to L+ by FS8 and FS16. If the input sign to the flip-flop is a plus, the flip-flop remains set; if the input is minus, the flip-flop is restored to L—.

3-76. REGISTER X SIGN FLIP-FLOP. The rX-sign flip-flop stores the sign of the contents of register X during multiply and transfer instructions. In the divide instruction, the flip-flop stores the sign of the dividend. The dividend is stored in register A. In the multiply, divide, and Y-transfer instructions, FS78 sets the flip-flop to X+ and alerts gates 34 and 35 to the M1 signal.

At the beginning of the multiply and divide instructions, the sign of the multiplier or dividend is on the M lines (M1). In the Y instruction, the M1 bit indicates the sign of the word being transferred from storage. If the M1 signal is low, the flip-flop is restored to X—; if the M1 is high, the flip-flop is set to X+.

In the second step of the multiply instruction, FS14-1A at gate 37 jams the flip-flop to X+. The IER signal from the IER-OR flip-flop (heading 3-85) alerts gate 33 to the output of the rA-sign flip-flop, A—. The product of a multiplication is stored in both registers A and X. Therefore, the sign of register X must be identical to that of register A. If the sign of register A is a minus, as indicated by the A— signal, the flip-flop is restored to X—; if the sign of register A is +, gate 33 is blocked and the flip-flop remains set.

3-77. SIGN-DISPLAY CIRCUIT. The four REGISTER SELECTOR switches on the operator's control panel enable the operator to view the contents and the sign of any of the four circulating registers. The sign of register C is significant only if index registers are included. The input gates of the register-display circuit are alerted by the signals generated by these selector switches and cause the correct sign-indicator lamp to be lit. For example, if the operator energizes the switch for register A, the RSA signal is generated. The signal alerts gate 101 of the sign-display circuit. The gate samples the A— output of the rA-sign flip-flop. If the sign of register A is —, the A— signal is low and a high signal is sent to light the minus lamp on the control panel. If A— is high, indicating that the sign of register A is +, the gate is blocked and a high signal is sent to light the plus lamp on the control panel.

3-78. COMPLEMENT FLIP-FLOP. Although the program specifies either an add or a subtract instruction, the complement flip-flop determines from the magnitudes and signs of the quantities involved whether an addition or subtraction is required. If a subtraction is required, one of the quantities involved must be complemented; therefore, the CP-control signal is generated to control the complementing and additions. If an addition is required, complementing is unnecessary; therefore, the $\overline{\text{CP}}$ signal is generated to control circuits which perform the addition

Add and subtract instructions are distinguished at the input of the CP flip-flop by the STR4 output of the static register. The $\overline{\text{STR4}}$ signal, if low at gates 8 and 9, indicates that an add instruction is staticized in the static register. The STR4 signal, if low at gates 10 and 11, indicates that a subtract instruction is staticized in the static register. The four gates for add and subtract are alerted by FS4.

When the $\overline{\text{STR4}}$ signal is low, gates 8 and 9 sample the M1 bit, the sign bit of the word from memory. A low M1 signal indicates that the sign is —; a low $\overline{\text{M1}}$ signal indicates a + sign. These signals are compared with the A+ or A— signal, the sign of the contents of register A. Either of the two gates is permissive only to unlike signs from storage and register A. The flip-flop is restored initially to $\overline{\text{CP}}$ at buffer 19 by the t′11B+ timing signal. If the signs are unlike, the flip-flop is set to CP. If the signs are alike, the gates are blocked and the flip-flop remains restored to $\overline{\text{CP}}$.

If the subtract instruction has been staticized, the STR4 signal is low at gates 10 and 11, which also sample the signs of the word from storage and register A. When the signs of the two words are alike, the gates are permissive and the flip-flop is set to CP. Unlike signs block the gate to keep the flip-flop restored.

If complementing takes place in the first step of the add or subtract instructions, the result of the first step may be complemented in the second step of these instructions. When complementing is required during the second step of the instructions, FS74 is generated to alert gate 12, which sets the flip-flop to CP. During a storage-search operation or 82 instruction, FS74 alerts gate 12 to set the flip-flop to CP.

3-79. OVERFLOW FLIP-FLOP. The overflow (OF) flip-flop receives signals which indicate that overflow conditions are present in various computer circuits. The flip-flop is used in the add, subtract, divide, and test instructions.

In the second step of the divide instructions, the OR-control signal alerts gate 27. The gate tests for the presence of the division sentinel in the most-significant-digit position of register X. When the sentinel (0101) is present, the gate is permissive and the flip-flop is set. The OF signal controls starting the final stage of division (D3).

Improper division occurs when the MQC-countdown circuit counts below 0, indicating that the problem was programmed improperly or that an error is present. The OF flip-flop input gate 28, alerted by FS31, samples the Q2 and Q3 outputs of the MCQ flip-flops. The presence of Q2 and Q3 indicates a combination other than 0 through 10 which means that the divisor was subtracted more than ten times from the dividend. Such a combination makes the gate permissive, generating the improper division signal (DI) and setting the overflow flip-flop. The DI signal generates an ending pulse in the static register which clears the STR flip-flops to 0 and initiates a search for the c + 1 address. The third step of division (D3) does not take place when improper division occurs.

When the sum of two ten-digit quantities exceeds the capacity of register A, the programmer must provide for storage of the extra-sum digit. When this condition is present, the A, C, $\overline{\text{CP}}$, and t11B signals are low at gate 29. The A and C signals signify decimal carry at the end of addition. Function signal 4 alerts the gate during addition. When the gate is permissive to its input signals, the flip-flop is set to OF.

Control signal jam I2A, a high output of the I/O abnormal-condition flip-flop (drawing 1-2A), sets the overflow flip-flop. The OF signal controls the overflow-delay flip-flop which in turn controls the reading of the c + 1 address. The overflow flip-flop is restored to $\overline{\text{OF}}$ by the OF2+ and GCB+ control signals. Signal OF2+ is the output of the overflow-delay flip-flop; GCB+ is generated when the operator presses the GENERAL CLEAR button on the control panel.

3-80. OVERFLOW-DELAY FLIP-FLOP. After overflow results from addition or an abnormal condition, the program continues normally into a search for the next instruction. The programmer anticipates overflow by programming into memory-location c + 1 (the next storage location on the memory after the specified c address) the instruction he wants the computer to follow when overflow occurs. When the next instruction is located by the comparison operation, the TS flip-flop is set. The overflow-delay flip-flop generates a signal (OF2+) which keeps the TS flip-flop set to $\overline{\text{TS}}$ to delay for one word time the reading of the selected address. As a result, the next address is read from the c + 1 address.

During the third step of division, D3, the OF2+ signal is generated by FS32A. The OF2+ signal in turn restores the overflow flip-flop to the normal $\overline{\text{OF}}$ condition.

3-81. MULTIPLIER/QUOTIENT COUNTER

The multiplier/quotient counter (MQC) (drawing 1-16A) consists of four flip-flops, a clear circuit, and a countdown circuit. The p7 digit of an instruction is set up in the MQC flip-flops during the staticize step of every instruction. It is significant, however, only during select-stacker or shift instructions to designate the stacker or the number of times to shift left or right. The multiplier/quotient counter is also used, as its name implies, during multiply and divide instructions.

The MQC also serves as an instruction decoder during buffer-to-main-storage transfer instruction with translation. The buffer-band-translate flip-flop is set, generating a low BBTRLAT signal, to indicate a transfer with translation of each input or output word. The reading set up in the MQC for specific instructions is stepped down each word time, and the outputs are decoded by gates alerted by the BBTRLAT signal (drawing 1-3A). The permissive gate generates a high signal which is decoded in the function encoder to function signals that control the steps needed to translate the words in transit between buffer and main storage.

3-82. MQC FLIP-FLOPS. In the left or right circular-shift instructions, the p7 digit designates the number of shifts required. The countdown circuit counts down to 0 from the number stored in the flip-flops. In the staticize step of either shift instruction, FS2 alerts the input gates of the flip-flops to inputs from the M lines at time interval t7B—. The four numeric bits of the p7 digit of the instruction word from storage enter the flip-flops during this time interval. The outputs of the flip-flops are the Q1 through Q4 signals which alert the gates of the countdown circuit.

In the multiply instruction, the flip-flops are cleared initially to 0 by a high signal from the clear-MQC circuit, so that a multiplier digit can be read into the circuit from register X. If the multiplier contains five digits, they will be stored, one by one, in the MQC flip-flops. The least-significant-multiplier digit from register X is the first digit of the multiplier to be stored in the flip-flops. The four numeric bits of the digit, X1 through X4, enter the flip-flops at gates 4A, 4B, 4C, and 4D, all alerted by the IER-OR control signal (heading 3-85). A low X signal denotes a 1 bit; a high X denotes a 0 bit. The four bits become the Q outputs of the flip-flops. In the second stage of multiplication, FS61 alerts the gates of the countdown circuit. When the MQC has counted down to 0 for each multiplier digit, the Q outputs of the flip-flops initiate a new phase of the multiplication process at the IER flip-flop. (Refer to section 4.)

At the beginning of the divide instruction, in the D1 step, the clear-MQC circuit clears the flip-flops to 0 so they can be used as a left-shift path for the contents of register X. When the IER-OR signal alerts gates 4A, 4B, 4C, and 4D in the second and third steps of division, the X1 through X4 outputs of register X enter the MQC flip-flops to be stored for one pulse time. Then, the X-input bits become the Q outputs and return to register X. In this way, the entire contents of register X are left-shifted. (Refer to section 4.)

During the second step of division, the flip-flops and the countdown circuit monitor the number of divisor-to-dividend additions, and these numbers become the final-quotient digits. In preparation for this quotient-counting operation, the four flip-flops are set initially to a count of 10. The OR-control signal from the OR flip-flop makes gate 1 permissive, jamming 1 bits into flip-flops 1, 2, and 4. The output of the four flip-flops, then, is 1101, the numeric-bit combination for the digit 10. The 10 combination is stored in the flip-flops until the first divisor-to-dividend addition is accomplished, when the countdown circuit places the combination for 9 in the flip-flops. The countdown circuit reduces by a count of one the digit stored in the flip-flops with each ensuing addition. After the final addition of each add step takes place, the Q outputs of the MQC go to register X and become the final-quotient digits.

In the card-reader select-stacker (47) instruction, the p7 digit indicates the stacker which is to receive the card. In the staticize step of the instruction, the p7 digit on the M lines is stored in the MQC flip-flops as in the shift instructions. The Q outputs of MQC which result from the p7 digit are sampled in the synchronizer circuits to select the specified stacker.

3-83. COUNTDOWN CIRCUIT. The countdown circuit consists of a group of 26 gates which reduce or step down by one the digit stored in the MQC flip-flops. The output of one or more of these gates changes the state of the flip-flops to store the new digit. Some of the countdown-circuit gates are permissive during the shift, divide, and multiply instructions while others are permissive during the buffer-to-main-storage transfer instruction with translation.

For an explanation of the countdown during each transfer with translate instruction, refer to manual 3. The countdown procedure for the divide, shift, and multiply instructions is explained in detail in the following paragraphs.

Primary step gate 50, requiring FS61, is made permissive at t1B to alert secondary step gates 14 through 18, 20, and 21. Gates 20 and 21 are blocked by a high BBTRLAT signal from the restored buffer-band-translate flip-flop. Note that the BLN2 signals on gates 18 and 17, the BLN1 signal on gate 15, and the $\overline{\text{BBTRLAT}}$ signal on gate 14 are all low during these instructions.

Table 3-5 shows the various Q-input combinations to the gates from the MQC flip-flops, the gates which are permissive to the input combinations, the function of the output signals of the gates, and the output combinations of the flip-flops.

3-84. CLEAR-MQC CIRCUIT. The clear-MQC circuit generates a CLQ signal that clears the flip-flops to 0. The high CLQ signal blocks gates 5, 6, 7, and 8 of the flip-flops to restore the four flip-flops to $\overline{\text{Q1}}$, $\overline{\text{Q2}}$, $\overline{\text{Q3}}$, and $\overline{\text{Q4}}$ outputs. Function signal 62 at gate 19 normally generates the CLQ signal. In the second (D2) step of division, the CLQ signal is generated by control signal ORA+ at buffer 42. In the final step of division (D3), the signal is generated by FS32A at buffer 42.

3-85. IER-OR FLIP-FLOPS

The IER and OR flip-flops (drawing 1-17A) control the phases of multiplication and division. The IER flip-flop is set initially during the first step of multiplication at gate 11, which samples for input signals indicating that the first step of the instruction (MPY1) is complete and that the second step (MPY2) is ready to begin. The STR inputs to the gate, when low, indicate that the static register has sequenced to the MPY2 step of the instruction. The $\overline{\text{Q1}}$ through $\overline{\text{Q4}}$ signals indicate that the MQC flip-flops have been cleared to 0. During the add steps of the same multiply instruction, the same Q inputs indicate that the countdown circuit has counted down to 0 from the specified number of additions for each multiplier digit. As soon as the MQC counts to 0. as indicated at gate 11, the IER flip-flop is set, generating high signals IER A1+, IER A2+, and IER+, and low signals IER-OR, and IER. These signals control the operations of the multiply instruction. (Refer to heading 4-34.)

During the divide instruction, the OR flip-flop provides control signals for the division process. In the first step of division, D1, FS15 makes gate 9 permissive to set the flip-flop and generate high signals ORA+ and OR+ and low signals OR and IER-OR. These signals initiate the first phase (complementing and left-shift phase) of the D2 step. At the end of this phase, timing signal t11B+ restores the OR flip-flop, generating low ORA+, OR+, and IER-OR, and high OR, which initiate the add phase of the D2 step. All complement and shift phases thereafter are initiated at gate 10, which is alerted by FS31 and is made permissive by input signals indicating decimal carry.

3-86. MULTIPLY-DIVIDE ENDING TEST SWITCH

The MULTIPLY-DIVIDE ENDING TEST switch on the engineer's panel is primarily for maintenance. When energized, the switch produces a low output which alerts multiply gate 100 and divide gate 101. Both gates require the TS1 and t′11B— signals which indicate a specific storage location. During troubleshooting of the multiply

Table 3-5. MQC-Countdown Gates

USS-6 Code (Numeric Bits Only)	Gate	Inputs	Outputs	Output Code of MQC
Q4 Q3 Q2 Q1 10 = 1 1 0 1	17	Q1	N2B restores FF1	1100 = 9
9 = 1 1 0 0	15	Q3 $\overline{Q1}$	N4 sets FF1, FF2 N4B restores FF3	1011 = 8
8 = 1 0 1 1	17	Q1	N2B restores FF1	1010 = 7
7 = 1 0 1 0	16	Q2 $\overline{Q1}$	N3 sets FF1 N3B restores FF2	1001 = 6
6 = 1 0 0 1	17	Q1	N2B restores FF1	1000 = 5
5 = 1 0 0 0	18	$\overline{Q1}$ $\overline{Q2}$ $\overline{Q3}$	N1 sets FF3 N1B restores FF4	0100 = 4
4 = 0 1 0 0	15	Q3 $\overline{Q1}$	N4 sets FF1, FF2 N4B restores FF3	0011 = 3
3 = 0 0 1 1	17	Q1	N2B restores FF1	0010 = 2
2 = 0 0 1 0	16	Q2 $\overline{Q1}$	N3 sets FF1 N3B restores FF2	0001 = 1
1 = 0 0 0 1	17	Q1	N2B restores FF1	0000 = 0
0 = 0 0 0 0	14 18	$\overline{Q4}$ $\overline{Q3}$ $\overline{Q2}$ $\overline{Q1}$ $\overline{Q1}$ $\overline{Q2}$ $\overline{Q3}$	N5 sets FF2 N1 sets FF3 N1B restores FF4	0110 = Count below 0 (indicates improper division)

or divide instruction, the instruction is placed in a specific storage location so that the last phase of the second step coincides with the 0p0 pulse. This pulse, generated by the cycling unit, synchronizes the maintenance oscilloscope with the specific storage location (TS1 and t'11B—). The programmed instruction should end at the same time the 0p0 pulse is generated; that is, at t'11B—, which is the time when the TS1 signal from storage should occur.

In the second step of division, if the MULTIPLY-DIVIDE ENDING TEST switch is energized, FS31 makes gate 101 permissive to generate DEB and DEA control signals. Signal DEB jams the OR flip-flop to the add phase, and signal DEA initiates the final step of division by setting the D3 flip-flop. The final step of a divide-end operation is an interchange of the contents of registers A and X. In this interchange, the contents of register X are shifted through the MQC. The final-quotient digit, computed in MQC, is placed in the p1 position of register A.

In the second step of multiplication, if the MULTIPLY-DIVIDE ENDING TEST switch is energized, the reading in the static register makes gate 100 permissive to generate a high ME signal which sets the IER flip-flop and jams the MQC flip-flop to initiate multiply sentinel 0101. The final step of a multiply-end operation is a right-circular shift of the contents of registers A and X.

3-87. STANDARD-STORAGE UNIT

The storage unit performs two major functions, word storage and computer timing. The storage unit stores instruction words and data words which process input data from punched cards or the manual keyboard. The stored information can be read from the drum for use in the Processor or input-output synchronizers. The storage unit sends various types of timing pulses to the control, arithmetic, and input-output units to time the internal operations of the system.

The major component of the storage unit is the rotating storage drum. Figure 3-1 shows the simplified, functional arrangement of the standard equipment (the 5000-word storage drum). The following paragraphs describe 5000-word storage.

3-88. 5000-WORD STORAGE

Information is stored on the surface of the drum in the form of magnetized areas which are written and read by read-write heads. The four bits of a digit are written in parallel across the drum. A check bit is written in the fifth bit position. The digits of a word are written serially around the surface of the drum. A recorded word, therefore, consists of 12 digit positions around the drum, each digit consisting of five bits across the drum. The main-storage area of the drum can store 5000 of these 12-digit information words.

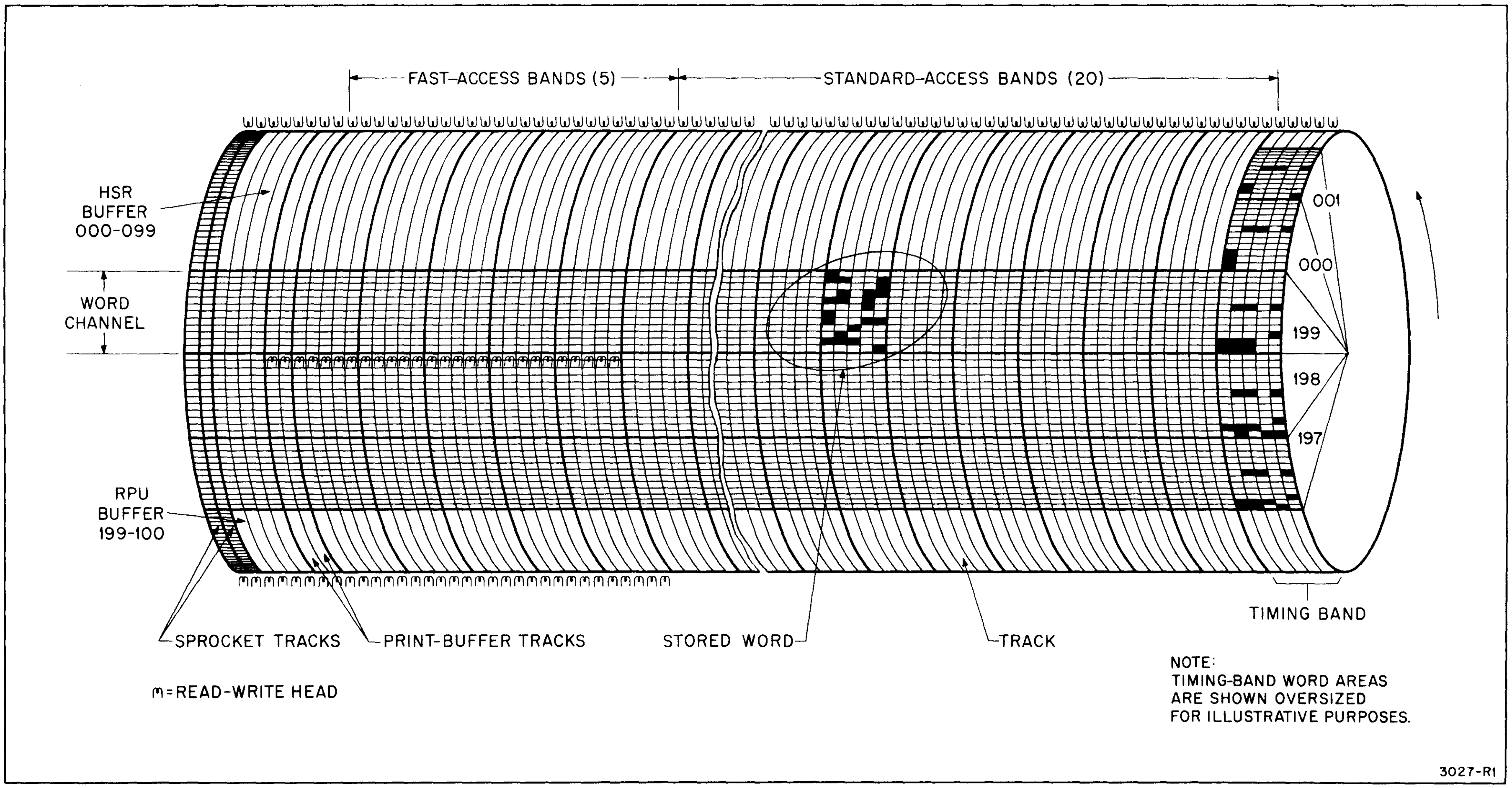


Figure 3-1. Storage Drum Showing Computer Characteristics

The main-storage area is divided into 25 areas, known as bands, around the circumference. Each band has 200 word-storage locations and consists of five tracks.

Of the 25 main-storage bands, 20 are designated standard-access (normal-access) bands (locations 0000-3999) and five are designated fast-access bands (locations 4000-4999). Each fast-access band is serviced by four read-write heads per track (20 heads per band), which are 90 degrees apart. Figure 3-1 shows only three of the four heads of a fast-access track; the fourth head is on the underside of the drum. The use of four heads makes any storage location on a fast-access band available within a maximum of 50 word times, or one-quarter of a drum revolution. Each standard-access track is serviced by only one read-write head per track (five heads per band). Any storage location on a standard-access band is available within a maximum of 200 word times, or one complete drum revolution.

Figure 3-2 illustrates the bit pattern formed by a stored computer word. A computer word consists of 12 digits: ten information digits, one digit for the sign, and one digit position for the space between words. The stored word in the figure is —6187203459—. Tracks 1 through 4, with biquinary bits weighted 1, 2, 4, and 5 (in that order) store the bits of the digits. Track 5 stores the check bit (heading 3-98). The example shows the serial-parallel method of storage with the digits of a word written serially and the bits written in parallel.

The storage locations for the 25 bands are listed below. Bands 1 to 20 are the standard-access bands with a total capacity of 4000 words; bands 21 to 25 are the fast-access bands with a total capacity of 1000 words.

Standard-Access Bands

Band 1	000-0199	Band 11	2000-2199
Band 2	0200-0399	Band 12	2200-2399
Band 3	0400-0599	Band 13	2400-2599
Band 4	0600-0799	Band 14	2600-2799
Band 5	0800-0999	Band 15	2800-2999
Band 6	1000-1199	Band 16	3000-3199
Band 7	1200-1399	Band 17	3200-3399
Band 8	1400-1599	Band 18	3400-3599
Band 9	1600-1799	Band 19	3600-3799
Band 10	1800-1999	Band 20	3800-3999

Fast-Access Bands

Band 21	4000-4199
Band 22	4200-4399
Band 23	4400-4599
Band 24	4600-4799
Band 25	4800-4999

In figure 3-1, the standard-access band containing the stored-word example is band number 6, reading from right to left and excluding the timing band. The stored word is in the next-to-last location of band 6 which contains storage locations 1000 to 1199, or address 1198. The addressing system is explained in detail under heading 4-4.

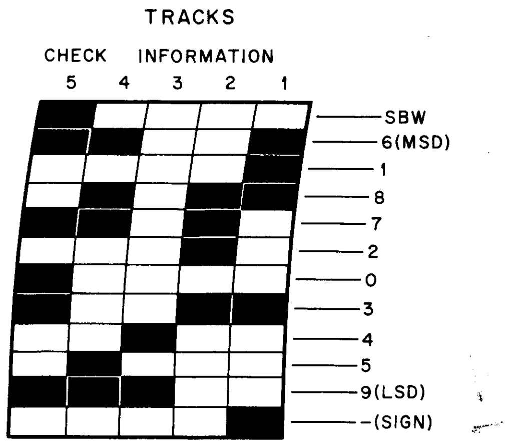


Figure 3-2. Word-Storage Pattern

3-89. BUFFER-STORAGE AREAS

In addition to the 5000-word capacity of the main-storage bands, a complete band and two tracks are used for input-output buffer storage. Figure 3-1 shows that the print-buffer area consists of two tracks, and that four tracks are divided between the high-speed-reader buffer (locations 0000-0099) and the read-punch-unit buffer (locations 100-199).

The buffer-storage areas compensate for differences in the speed of operation between the relatively slow input-output units and the fast rate of computation. For example, the time to advance and print one line of information is approximately 30 drum revolutions. If the information to be printed were available to the printer only from main-storage locations, the Processor could not process other instructions for nearly 30 drum revolutions because the read-write heads which serve main storage would be occupied with the transfer of information for three drum revolutions and actual printing would take an additional 27 drum revolutions. To save computer time, print information is transferred from main storage to a buffer-storage area. Once the transfer to the buffer areas is complete, the read-write heads which serve main storage are free.

Buffer-storage areas are provided also for information processed by the card reader and the read-punch unit. Information read from punched cards by either the card reader or the read-punch unit is transferred to a buffer-storage area and then to main storage. Information which is to be punched on cards first is transferred from main storage to a buffer-storage area and then to the read-punch unit.

The print-buffer area consists of two tracks which store print information for one line. Eight read-write heads serve the two print-buffer tracks. Each track is served by four heads mounted at 90-degree intervals around the circumference of the drum.

The card-buffer-storage area consists of four tracks referred to as the card-buffer band. The card-buffer band is divided into two parts; one half serves the card reader, while the other half serves the read-punch unit. The card-buffer band is served by eight read-write heads, two for each of the four tracks. The heads are mounted at 180-degree intervals around the drum's circumference.

3-90. TIMING BAND. The timing band, five tracks wide, controls the addressing of storage locations and generation of timing signals in the cycling unit.

Each of the 200 word-storage locations around the drum is identified by a combination of digits stored permanently in the timing band. The expanded view of the timing band (figure 3-3) shows the code combinations stored in a section of the timing band. When storage search takes place, the recorded combinations on the timing band are read from the drum. The 200 unique timing-band addresses are read, one by one, by the timing-band read circuits. The TS signals, indicating storage addresses, are transferred to the M lines and compared with the desired address. When the desired address corresponds to the address being read by the timing-band heads, the proper channel is selected and the information is read from that location.

A time-selection address is located on the word channel preceding the word with which it is associated. This allows one word-time delay for the addressed word to be read by the read-write heads.

Only five timing-band word areas are shown in figure 3-3. The three lowest-order digit positions of each word area indicate a channel across the drum. Because the timing-band addresses differ by one word time from the main-storage addresses, the digits 197 on the figure indicate that any word stored across the drum in the associated channel is located in the 196th position in one of the bands. The band is selected in the band-selection circuits. (Refer to heading 4-6.) The time-selection addresses stored in the timing band are numbered 000 through 199.

A 1 bit is stored in track 5 for any digit containing an even number of 1 bits. For example, the code combination for time-selection address 000 does not contain any 1 bits. To ensure that an odd number of 1 bits is read from the drum for any stored character, a 1 bit is added in the fifth bit position. Because the addresses can be numbered only from 000 to 199, the most-significant digit of an address can be only a 0 or a 1. For this reason, the most-significant digit of the channel address is identified by a single bit, the bit in the track 1 position.

The timing combination shown in figures 3-1 and 3-3 is recorded permanently in the t6 position of every timing-band word. The combination 1101 is read by the timing-band read circuits and sent to the cycling unit to generate the signals which time the internal operations.

Signals known as input-output sentinels control the input-output synchronizers and are also stored permanently in the timing band, but are not shown in figures 3-1 and 3-3.

3-91. SPROCKET TRACK. There are 2400 bit positions in a single track around the drum. In the two sprocket tracks, a permanent bit is stored in each of the 2400 positions. Two heads per track read the sprocket bits to ensure that at least one is read every pulse time and sent to the sinewave clock circuits which generate synchronized A- and B-phase power pulses. The power pulses control the magnetic-amplifier elements throughout the computer circuitry.

3-92. TIMING-BAND READ CIRCUITS

The timing-band read circuits (drawing 1-19A) consist of five amplifier circuits (which convert the combinations stored on the timing band into time-selection signals), the timing-band flip-flop, and the timing-error flip-flop. The TS-signal outputs of the read circuits can be grouped into three signal categories: storage-address, timing-combination, and control signals.

The storage-address signals, which are the three lowest-order digits of each word in the timing band, go to the comparator of the arithmetic unit, via the M buffers, to be used in the search operations. These signals are stored in the t0, t1, and t2 positions of the timing band.

Once every word time, the timing combination (1101) is read from the t6 position into the cycling unit. The cycling unit uses the 1101 combination to generate signals which time internal computer operations.

The TS outputs of the timing-band read circuits, combined with the timing signals from the cycling unit, form input-output sentinels which control the input-output synchronizers.

The timing-band flip-flop ensures that the t2 position of location 199 of the timing band contains a 0 because the next storage address is 000. During the initial recording of the timing band, t2 of location 199 is recorded first and t1 of location 199 is recorded last. Because the recording is stopped at t1, it may overlap into the t2 position, causing extraneous information to be recorded. To prevent the reading of this extraneous information, readout of the 000 combination is simulated by the timing-band flip-flop.

The flip-flop is set at gate 20 by the A sentinel (TS1 at t9B) in location 198 to generate high JB and JA signals which jam the timing-band read circuits to indicate 0 bits on the TS1 through TS4 tracks and a 1 bit (check bit) on the TS5 track.

The timing-error flip-flop, also shown in drawing 1-19A, is explained under heading 4-43.

3-93. CYCLING UNIT

The cycling unit (drawing 1-18A), controlled by the timing combination from the timing band, generates the timing signals which synchronize computer operations. The cycling unit consists of the delay line which produces the timing signals, the cycling-unit-error flip-flop, and the every-other-word flip-flop.

Every word time, the timing combination, TS4, TS3, $\overline{\text{TS2}}$, and TS1 (1101), is transferred from the timing band, through the timing-band read circuits, to the cycling unit. The combination makes gate 102 permissive at t6B to generate high signal t7A+. One half-pulse time later, high t7B+ and low t7B— signals are generated. The input signal continues through the cycling unit, which is actually a delay line, generating the variations of timing signals. The A and B phases of signals t0 through t11 are generated in high (+) and low (—) states. The high timing signals in general are blocking signals, while low signals are alerting signals on gates.

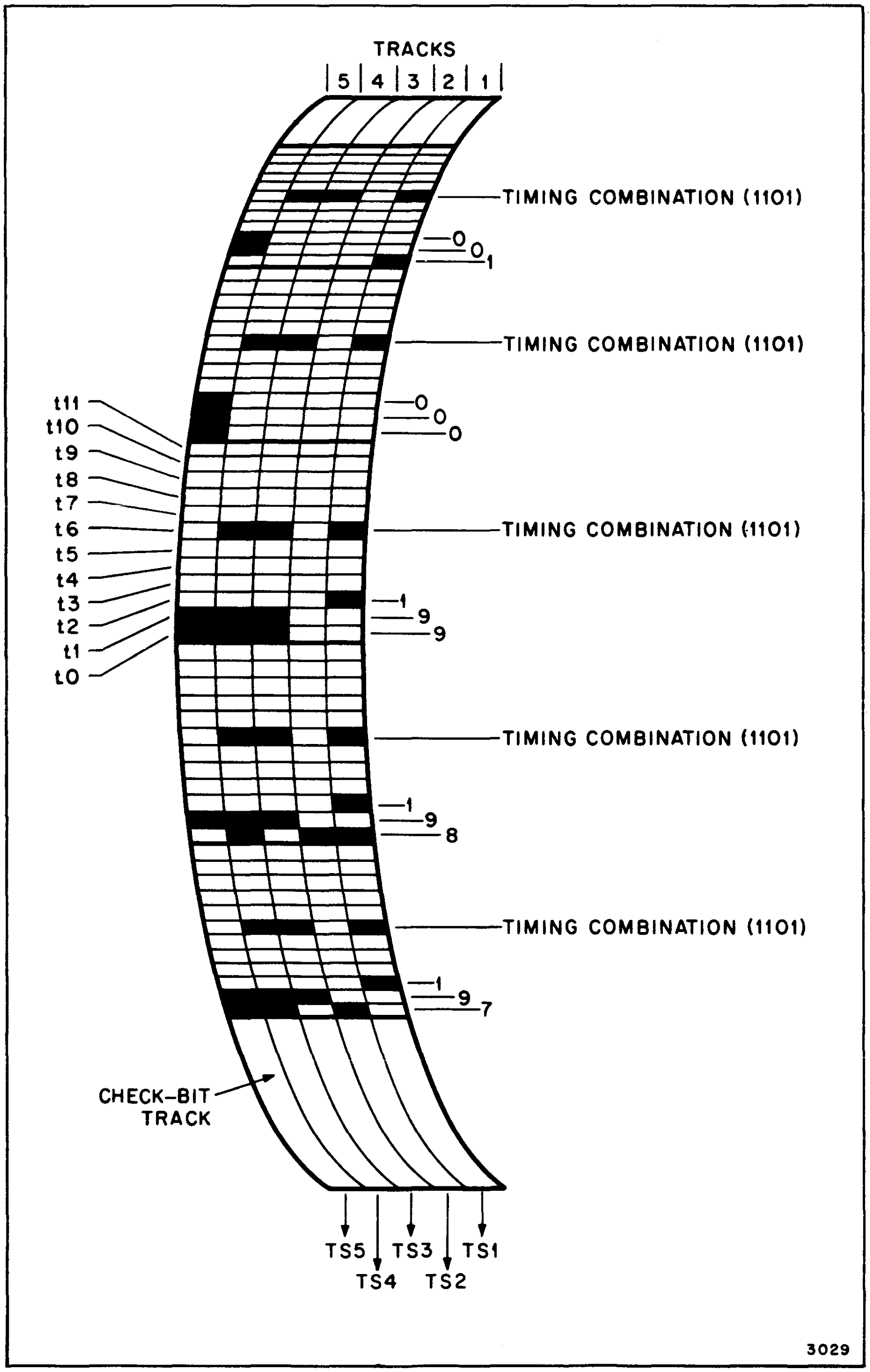

TRACKS
5
4
3
2
1
TIMING COMBINATION (1101)
0
0
1
TIMING COMBINATION (1101)
0
0
0
TIMING COMBINATION (1101)
1
9
9
TIMING COMBINATION (1101)
1
9
8
TIMING COMBINATION (1101)
1
9
7
t11
t10
t9
t8
t7
t6
t5
t4
t3
t2
t1
t0
CHECK-BIT TRACK
TS5
TS4
TS3
TS2
TS1
3029

Figure 3-3. Timing Band, Expanded View

The every-other-word (EW) flip-flop synchronizes signals from the control panel, or input-output units, with Processor timing. The BT sentinel (TS2 at t9B) generates a high set EW signal which sets the flip-flop, generating low signal EW and high signals $\overline{EW}$ and EWA. The flip-flop remains set until time interval t11B, when the high t11B+ signal at gate 48 overrides low EW, restoring the flip-flop. The flip-flop remains restored for one word time until low signals t11B— and $\overline{EW}$ at gate 47 set it again. The flip-flop thus changes state every word time.

Most external signals (generated by manually or mechanically operated switches) are sent to the set gate of a synchronizing flip-flop under the control of a signal from the EW flip-flop. The output signal of the synchronizing flip-flop alerts another gate control led by the same output signal from the EW flip-flop. A timing signal must also be present at both gates. Because the low output of the EW flip-flop and the timing signal occur together only once every other word time, the external signal must recirculate in the synchronizing flip-flop for two word times. During these two word times, if the signal is a weak or partial signal, it will build up or die out. If it dies out, the synchronizing flip-flop is set again the next time the EW flip-flop output signal and the timing signal occur together because a switch signal is present for more than two word times.

3-94. INPUT-OUTPUT SENTINELS

The input-output sentinels control transfer of information between the input-output units and the buffer-storage areas, and between the buffers and main storage. The sentinels are 0 bits or 1 bits recorded on tracks of the timing band in specific positions of one or more word locations, and are identified by one or more alphabetic, and sometimes numeric, characters such as BT for begin transfer and TO for transfer over.

Table 3-6. Input-Output Sentinel Chart

Timing Signals	Timing-Band Combination Signals			
	TS4	TS3	TS2	TS1
t11	Q4	Q5	Q6	Opo
t10	F	E	D	C
t9			BT	A
t8			B	
t7				Y
t6	1	1	0	1
t5			K	TO
t4		Q3	Q2	Q1
t3	J	ST	I	H
t2	R3		R1	X
t1	X	X	X	X
t0	X	X	X	X

As shown in table 3-6, each timing-band word location contains a timing-band address indicated by X's, a 1101 timing combination, and may also contain one or more sentinels. The timing-band address requires nine bit positions, and the timing combination requires four bit positions. The remaining bit positions of each timing-band word are available for the recording of input-output sentinels. Table 3-6 is a composite of all 200 word locations of the timing band, and shows the relative position of each sentinel in the location or locations in which it is recorded.

Some sentinels are recorded in only one location of the timing band. For example, the A sentinel is recorded as a 1 bit on track 1 at the t9 position of timing-band location 198. The gates controlled by the A sentinel require low TS1 and t9B signals, which occur together only in the 198 location and restrict the operation of the gate to once a drum revolution. The F sentinel is typical of sentinels which are recorded in more than one location around the band, and is recorded as a 1 bit on track 4 at the t10 position of 14 word locations: 004, 013, 022, 045, 054, 066, 085, 098, 107, 129, 138, 151, 169, and 182. The gates controlled by the F sentinel require low TS4 and t10B signals, which occur together 14 times during one drum revolution.

3-95. WRITE CIRCUIT

The components of the write circuit (drawings 1-21A and 1-22A) write information on the drum. Information is transferred from the M buffers to the write circuits and converted to the form necessary for recording. The components of the write circuit are described in the following paragraphs.

3-96. WRITE-PEDESTAL GENERATOR. The read circuits must be deenergized before writing. During any instruction which involves writing information into storage, a high signal is applied to buffer 104 of the write-pedestal generator package. The output of the generator is a write pedestal which disconnects the read amplifiers of the standard-access heads. The IR (inhibit-read) output of the generator disconnects the read amplifiers of the fast-access heads.

3-97. WRITE-INPUT CIRCUITS. Four bits of each digit to be written are transferred into the write-input circuits on the $\overline{W}$ lines. The barred M outputs ($\overline{M1}$ through $\overline{M4}$) from the minuend buffers are transferred through amplifiers and enter the write-input circuits of the standard- and fast-access write circuits as barred W lines.

3-98. CHECK-BIT COMPUTER. The check-bit computer selects the check bit for each digit written on the drum and also checks the accuracy of information read from the drum. The read check is made to ensure that no bits have been affected by failure of any of the transmitting elements. When a digit to be written contains an even number of 1 bits, a 1 check bit is written. When a digit contains an odd number of 1 bits, a 0 check bit is written.

3-99. WRITE FLIP-FLOP. The write flip-flop controls writing in a main-storage location by alerting the input gates of the phase-modulation coder. Set gate 5B is made permissive to write the contents of registers A, X, or L. Set gates 5A, 100, and 101 are made permissive during an instruction transferring the contents of the card buffer into main storage. The $\overline{\text{BBTRLAT}}$ signal on these gates indicates a transfer without translation. Control signal RSC2 or RSC5 required at these gates indicates that the card-buffer-read flip-flop is set. Set gate 5C is made permissive during an instruction transferring the contents of the card buffer into main storage. The BBTRLAT signal on this gate indicates a transfer with translation.

The low set output of write flip-flop W6 alerts the input gates to the standard-access phase-modulation coder (drawing 1-21A) and to the fast-access phase-modulation coder (drawing 1-22A). The low restored output, $\overline{\text{W6}}$, alerts set gate 5C, and the high set output, IR2D, generates the write pedestal.

The four bits of each digit to be written are sent in parallel on the M lines to the write-input circuits and to the check-bit computer. An even number of 1 bits in a combination generates a high $\overline{\text{CK}}$ signal. An odd combination of 1 bits generates a low $\overline{\text{CK}}$ signal. For example, if the numeric bits for 7 (1010) enter the check-bit computer, the inputs are M4, $\overline{\text{M3}}$, M2, $\overline{\text{M1}}$. Only diode gate 26F is permissive to generate a high $\overline{\text{CK}}$ signal which writes a 1 bit into the fifth bit position on the drum. The check-bit computer operates when this same information is read from the drum to check for parity error. (Refer to heading 4-42.)

3-100. PHASE-MODULATION CODER. The phase-modulation coder receives an input pulse and transforms it to two positive sinewaves, one of which is delayed one half-pulse time. The phase of the two output pulses determines whether a 1 or 0 is written on the drum. Figure 3-4 shows a typical phase-modulation circuit with a 1 input (a 1 is a high-input signal). The high-input pulse is complemented to a low by input complementer 1. Gates 2 and 2A, alerted by the set output of the write flip-flop, are permissive to the low pulse. Complementing amplifier 2 complements the signal to a high pulse, which goes through a buffer to one end of the write transformer. Complementer 2A complements the low-input signal to a high pulse which goes to amplifier 3. Amplifier 3 delays the pulse for one half-pulse time, and sends it to the opposite end of the write transformer. The first output of the transformer occurs one half-pulse time earlier than the second output. The first pulse draws current in a positive direction at the center-tapped write-head winding, supplying the positive portion of the current waveshape.

The second pulse draws current in a negative direction at the head winding to supply the negative portion of the waveshape. The current waveshape of a 1 recorded on the drum is shown in figure 3-5a.

A 0 input to the coder (a 0 is a low-input signal) also produces two high outputs to the write transformer, but with reversed timing. The low input is complemented to a high by input complementer 1. The high signal blocks gates 2 and 2A but is buffed into amplifiers 4 and 4A. The output of amplifier 4 goes directly to one end of the write transformer. The output of amplifier 4A goes to the opposite end of the transformer via amplifier 5 which delays the pulse for one half-pulse time. The input to both ends of the write transformer is a high pulse, but the first input precedes the delayed input by one half-pulse time. When the first output precedes the second, the negative portion of the current waveshape on the drum precedes the positive portion (figure 3-5b). A 011 combination appears on the drum as a low-high, high-low, high-low configuration (figure 3-5c).

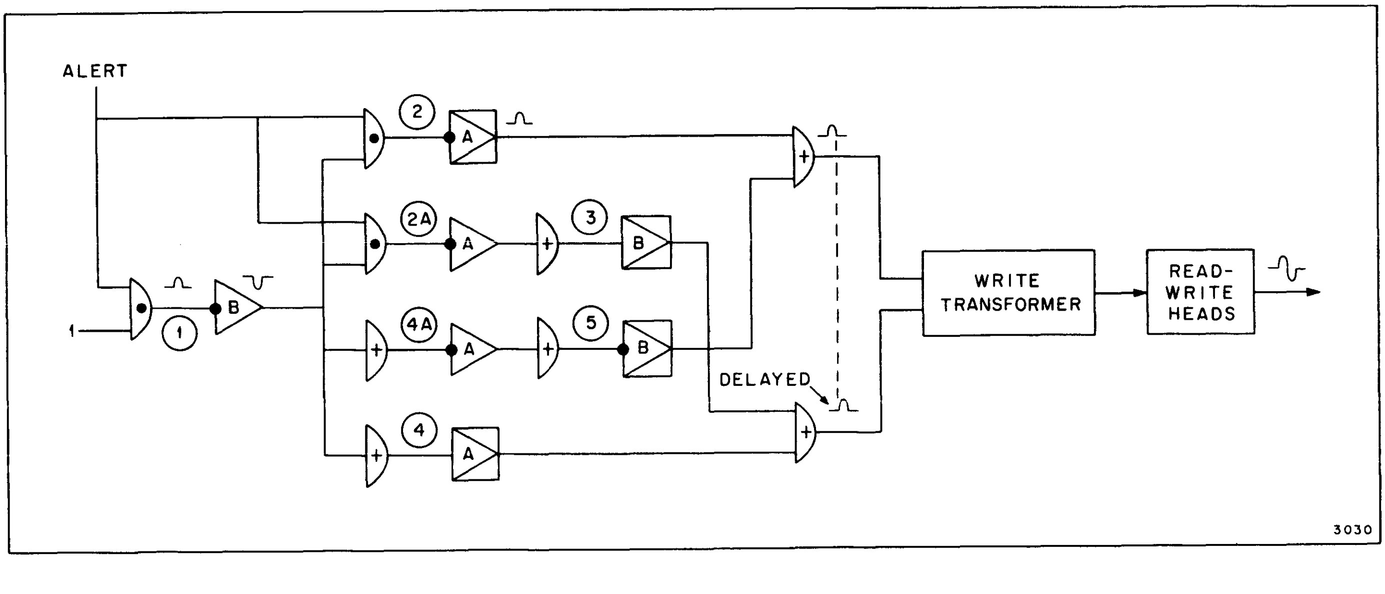


Figure 3-4. Phase-Modulation Coder

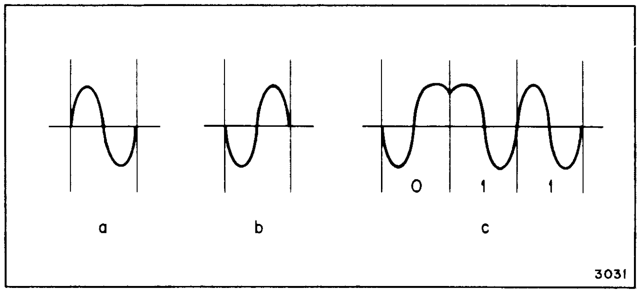


Figure 3-5. Phase-Modulated Waveshapes

The check bit is also phase modulated, at gate 25. The $\overline{\text{CK}}$ signal is high when a 1 bit is to be added in the fifth bit position and low when a 0 is to be written.

The outputs of the write transformers of the standard-access read-write circuits are the WD signals which go to the memory switches. The write-transformer outputs of the fast-access circuits are the WF signals which also go to the memory switches.

3-101. READ CIRCUIT

The components of the read circuit (drawings 1-21A and 1-22A) read information from the storage drum. The major components of the read circuit are the read flip-flop, the read-output circuits, the check-bit computer, the check-timing flip-flop, and the main-storage-check flip-flop. (The latter two components are explained under heading 4-42.)

3-102. READ FLIP-FLOP. The read flip-flop controls readout from the drum. When the flip-flop is set, a low-output signal makes the gates of the read-output circuits permissive to output from the drum. When the flip-flop is restored, the gates are blocked to inhibit the read circuits.

The flip-flop is set at gate 42C during a print instruction to read information from main storage and transfer it to the print buffer. The gate is alerted by FS42 and $\overline{\text{STR4}}$ to indicate that the print instruction, and not a paper-advance instruction, is staticized. The TS2 and t8B— signals, a B sentinel, make the gate permissive.

In a search operation, the flip-flop is set at gate 42A to read from a selected address. Function signal 1 alerts the gate, and the TS and $\overline{\text{OF}}$ signals indicate that the search operation is successful. This gate is blocked by high block-RD1 and block-RD2 signals during a register-search operation.

The flip-flop is set at gate 42E to transfer an output word, read from main storage, into the card buffer. The $\overline{\text{BBTRLAT}}$ signal required at the gate indicates that the output word, to be punched into a card, is transferred without translation. Gate 42D, requiring a BBTRLAT signal, is made permissive by a reading setup in the MQC to transfer an output word with translation.

The set output of the flip-flop (RD) alerts gates 45A and 45B, one of which is made permissive by FM (fast-memory) or SM (standard-memory) signals. The FM signal indicates that the information is to be read from a fast-storage band by a fast-access read-write head. If FM is present, gate 45B generates the RGF signal to alert the read-output gates of the fast-access read circuits (drawing 1-22A). This allows the fast-access read heads to read information from the fast-access bands.

If the SM signal is present, rather than the FM signal (only one can be present at a time), gate 45A operates to alert the output gates of the standard-storage circuits. The SM signal indicates that information is to be read from the standard-access bands of main storage.

3-103. READ-OUTPUT CIRCUIT. The four information bits of a digit read from a standard-access band become the DM′1 through DM′4 signals. The four bits read from a fast-access band become the DM15 through DM45 signals. These signals go directly to the M buffers and onto the M and MT lines. The M and MT lines transfer the outputs of the read circuit to the components specified by the staticized instruction. The M lines also return the read outputs to the check-bit computer, which checks for an odd number of 1 bits. (Operation of the check-bit computer, the check-timing flip-flops, and the main-storage-check flip-flop in reading erroneous combinations from the drum is explained under heading 4-42.)

3-104. MEMORY-SELECTION CIRCUITS

The memory-selection circuits control the memory switches, which energize a specific read-write head for reading or writing. Head selection, band selection, and switch selection are accomplished by the memory-selection circuits (drawing 1-20A).

3-105. BAND-SELECTION FLIP-FLOPS. The band-selection flip-flops are divided into two groups, those which determine the 100's digit (p3) of the desired storage address, and those which determine the 1000's digit (p4) of the address. For example, if the address to be written into or read from is 3689, the p3 band-selection flip-flops generate signals which indicate the 6 digit of the address and the p4 flip-flops generate signals which indicate the 3 digit. Digits 8 and 9 are determined in the comparator.

The outputs of register C, containing the desired address, are on the S lines during memory selection. The bits of the p3 digit of the address are sampled by the five p3 band-selection flip-flops (table 3-7). The table shows that if the p3 digit of the address is, for example, 2 or 3, the MS2 flip-flop is set, generating signal MS2, which goes to the switch-selection-gate matrix.

The p4 digit determines the 1000's digit of the desired address. Only the two lowest-order bits of p4 are required to determine the 1000's digit because the digit ranges from only 0 to 4. The two bits are read directly from register C (signals C13, C14, C23, and C24) to the MS10 and MS20 flip-flops. The flip-flops are set or restored by the signals from register C. Their outputs (MS10 and MS20) indicating the fourth, or 1000's digit of the address, go to the switch-selection-gate matrix.

Table 3-7. Band-Selection Flip-Flops (p3)

p3 Digit	p3 Bits S4	S3	S2	S1	Flip-Flop Output
0	0	0	0	0	MS0
1	0	0	0	1	MS0
2	0	0	1	0	MS2
3	0	0	1	1	MS2
4	0	1	0	0	MS4
5	1	0	0	0	MS4
6	1	0	0	1	MS6
7	1	0	1	0	MS6
8	1	0	1	1	MS8
9	1	1	0	0	MS8

The fast/standard flip-flop determines whether the desired location is the fast- or standard-access area of storage. The C33 and C34 signals from register C indicate whether the third bit of the p4 digit is a 1 or a 0. If it is a 0, the p4 digit is less than 4; therefore, the desired address is in locations 0000-3999. These are standard locations and as a result the SM signal is generated. If the third bit of p4 is a 1, the p4 digit is a 4; therefore, the desired address is in locations 4000-4999. Because these are fast-access locations, the FM signal is generated by the fast/standard flip-flop. The FM or SM outputs also go to the switch-selection-gate matrix.

3-106. CLEAR-BAND-SELECTION CIRCUIT. The band-selection flip-flops, which process the p3 and p4 digits, operate in parallel with and at the same time as the time-selection circuits which process the pl and p2 digits. If the comparison circuits determine that the pl and p2 digits are unequal, the TS flip-flop generates $\overline{TS}$, which indicates that the band-selection flip-flops are to be restored. Function signal 1 alerts the gate which samples the $\overline{TS}$ signal. If $\overline{TS}$ is low, the circuit generates a high output to restore the eight band-selection flip-flops before the switch-selection circuits are affected. The FS30+ signal on gate 62 is high during instructions involving the input-output buffers in which a specific band is chosen but not a specific location in that band. Therefore, FS30+ blocks gate 62 so that the band-selection flip-flops are not cleared, but continue to select a band. The RCT3, 4, 6, or 8 signals, generated on completion of an instruction involving the input-output buffers, clears the band-selection flip-flops.

Although FS1A is high during a search operation, the effect of FS1 overrides that of FS1A at gate 64. However, when any instruction other than search is in process, FS1A is low to clear the band-selection flip-flops.

3-107. HEAD-SELECTION FLIP-FLOPS. The two lowest-order digits of the desired address are compared with the same two digits of the timing-band address in the comparator circuits. These two digits narrow the search to four possible channels. The desired location then is in one of four quadrants of the drum. Head-selection 1 flip-flop and head-selection 2 flip-flop determine which of the four quadrants contains the desired information. The outputs of the flip-flops go to the memory-switch-selection matrix. (More detailed information on the two flip-flops is contained under heading 4-5.)

3-108. SWITCH-SELECTION CIRCUITS. The switch-selection gates, shown as a matrix in drawing 1-20A, sample the outputs of the band- and head-selection flip-flops to determine the memory switch to be energized. Only one of the switches is energized by the 13 inputs to the gates, and the output of the gates goes to a switch driver. The elements of a switch driver are shown within the left driver in the drawing. The outputs of the switch drivers are the FS0 through FS38 signals, one of which energizes the correct memory switch. The numbers within the switch-driver blocks on the drawing signify the bands controlled by each driver.

If the storage location is within the 1000-1199 standard-access band, gate 6A is permissive to the SM, MS0, MS10, and $\overline{MS20}$ low-input signals. The switch-driver output is the FS10 signal which energizes the correct switch to read from the chosen band. The same switch driver also serves the fast-memory switches. In addition, the driver selects a switch which energizes one of the four heads of a fast-access band. The FS10 signal can be generated also by input signals FM, MS2, $\overline{QS2}$, and QS1 at gate 6B. The 4203 designation refers to band 4200, head 3. The top number in each switch-driver block refers to the standard-access band, the lower number refers to both the fast-access band and the head of that band.

3-109. MEMORY SWITCH. The memory switches (drawing 1-23A) select a group of ten heads specified by the FS-signal inputs. The ten selected heads include five standard-access heads and five fast-access heads which record the four information bits and one check bit. The read and write circuits determine which five of the ten heads are selected. For example, if the FS10 signal energizes switch 1000, ten heads are chosen. However, the FS10 signal was generated with the help of the low FM signal, indicating fast-access storage. In reading from the drum, the SM signal is therefore high and blocks the standard-read circuits from reading, while the low FM signal allows the fast-read circuits to read. As a result, the SM signal blocks the transistor flip-flops of the read units to block readout to the DM′ line, while the FM signal makes the transistor flip-flops permissive, allowing information from fast-access read heads to enter the DM lines. In writing on the drum, the SM and FM signals block the input gates of the phase-modulation coders of either the standard or fast circuits.

The WF (fast-storage) and WD (standard-storage) signals are the inputs to the selected heads when writing, or the outputs of the heads when reading from the drum.

3-110. EXPANDABLE STORAGE UNIT

The expandable storage unit consists of a standard magnetic drum with decreased head spacing which makes it expandable to a 9200-word storage unit. The expandable unit covered under headings 3-110 to 3-113 refers

Table 3-8. 9200-Word-Drum Storage Locations

Band	Location	Band	Location
Section A, Standard Access			
1	0000-0199	11	2000-2199
2	0200-0399	12	2200-2399
3	0400-0599	13	2400-2599
4	0600-0799	14	2600-2799
5	0800-0999	15	2800-2999
6	1000-1199	16	3000-3199
7	1200-1399	17	3200-3399
8	1400-1599	18	3400-3599
9	1600-1799	19	3600-3799
10	1800-1999	20	3800-3999
Section A, Fast Access			
21	4000-4199	24	4600-4799
22	4200-4399	25	4800-4999
23	4400-4599		
Section B, Standard Access			
26	5000-5199	35	6800-6999
27	5200-5399	36	7000-7199
28	5400-5599	37	7200-7399
29	5600-5799	38	7400-7599
30	5800-5999	39	7600-7799
31	6000-6199	40	7800-7999
32	6200-6399	41	8000-8199
33	6400-6599	42	8200-8399
34	6600-6799	43	8400-8599
Section B, Fast Access			
44	8600-8799	46	9000-9199
45	8800-8999		

to the 5000 word operation; the expanded unit covered under heading 3-114 to 3-119 refers to the unit expanded for 9200-word capacity. The following sections describe those storage circuits which operate differently from that of the standard 5000-word circuits. Circuits which operate identically for either the standard or expandable type of storage are described under headings 3-89 to 3-94. The logic circuits for the 5000- and 9200-word storage are shown in manual 2, drawings 1-20C, 1-21C, 1-22C, and 1-23C.

The physical size of the expandable-memory unit is the same as the standard 5000-word drum. The read-write heads were made smaller and placed closer together to provide for the increased storage. The expanded memory unit has two sections, A and B. Section A (0000 — 4999) has standard and fast storage with WD and WF memory lines. Section B (5000 — 9199) also has standard and fast storage, but the memory lines are WDD for normal memory and WFF for fast memory.

For models 159 and 139, the only difference from models 131 and 131A is that standard-access read-write circuits (drawing 1-21A) and fast-access read-write circuits (drawing 1-22A) are combined into one illustration (drawing 1-21C), less the 9200-word logic circuits which are shown as 9200-word storage only. (Physically, the expandable 5000-word storage (models 159 and 139) and the nonexpandable 5000-word storage (models 131 and 131A) are not interchangeable.)

The 9200-word storage unit is divided into two areas, section A and section B, as shown in figure 3-6. Section A consists of 25 bands, each having 200 word-storage locations. Twenty bands are designated standard-access (locations 0000 — 3999); five are designated fast-access (locations 4000 — 4999). Section B consists of 21 bands, each band having 200 word-storage locations. Eighteen bands are designated standard-access (locations 5000 — 8599); three are designated fast-access (locations 8600 — 9199).

Table 3-8 lists the storage locations for the 46 bands of storage which are available with the 9200-word drum. Bands 1 to 20 are the standard-access bands for the A section of the drum, with a total capacity of 4000 words; bands 21 to 25 are the fast-access bands for the A section of the drum with a total capacity of 1000 words. Bands 26 to 43 are the standard-access bands for the B section of the drum, with a total capacity of 3600 words; bands 44 to 46 are the fast-access bands for the B section of the drum with a total capacity of 600 words.

3-111. 5000-WORD STORAGE

The circuits which are added in the revision from the standard 5000-word unit logic to the expandable 5000-word unit are the read-inhibit circuit and the write-switch flip-flop.

3-112. READ-INHIBIT CIRCUIT. The read-inhibit circuit (drawing 1-22C) inhibits the input windings of the RAP-S circuit. Each RAP-S circuit has two input windings: one for standard-memory access, the other for fast-memory access. During a write instruction, both outputs from the read-inhibit circuit (RDINHNM, read-inhibit-normal-memory, and RDINHFM read-inhibit-fast-memory), are generated to inhibit all RAP-S input windings. The read-output circuit is gated with a DROG1 signal which is generated from the output of the main-storage-read flip-flop.

3-113. WRITE-SWITCH FLIP-FLOP. The write switch and the fast/standard flip-flop outputs are gated together to select one of the MSN (memory-select negative) package circuits. The MSN output energizes one of the output windings of the WXP-S (write transformers) to permit writing onto standard or fast memory. Each WXP-S circuit has two output windings: one for standard access and one for fast-memory access. An RMSN (reset-memory-selection negative) signal, generated when writing is complete, resets both flip-flops. The write-switch flip-flop was added because the inputs to the MSN package cannot be less than one word time. The write-input circuit gates the information to be written and gates the output of the write flip-flop (W6) during a write instruction.

3-114. 9200-WORD STORAGE (EXPANDED MEMORY)

To expand the memory from 5000 words to 9200 words capacity the following additional logic-circuit modifica-

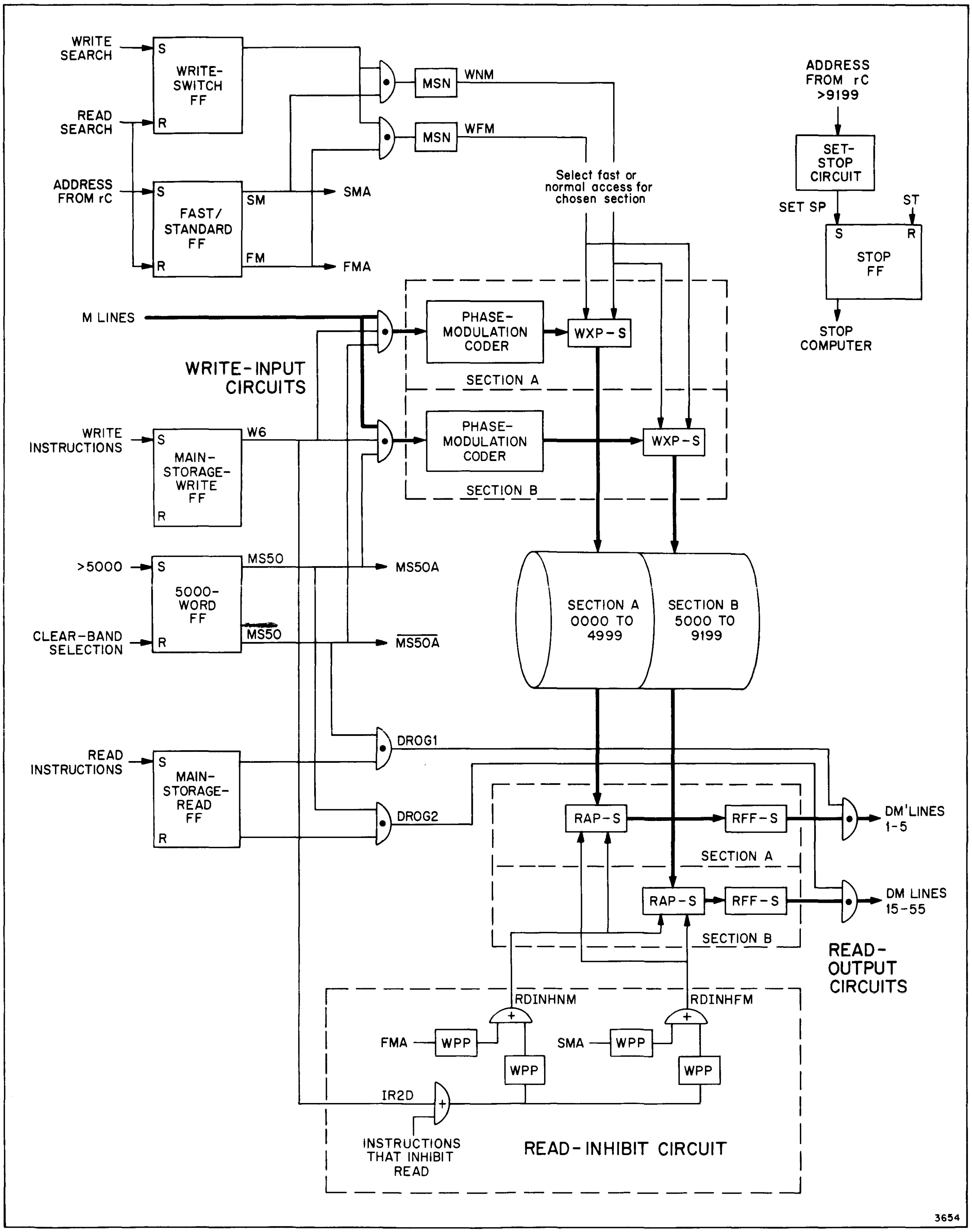


Figure 3-6. 9200-Word Memory, Block Diagram

tions are needed: a set-stop circuit, a 5000-word flip-flop, a band-selection flip-flop, and a fast/standard flip-flop.

3-115. SET-STOP CIRCUIT. The set-stop circuit (drawing 1-20C) detects when the programmed address is over 9199. This circuit samples the special output of the band-selection flip-flops (refer to heading 3-118), and generates a high SET SP signal. The SET SP signal sets the stop flip-flop (drawing 1-5B, manual 2) to stop computation.

3-116. 5000-WORD FLIP-FLOP. The output of the 5000-word flip-flop (drawing 1-20C) selects which section of the storage unit is to be used during a read or a write instruction.

When reading, a low MS50 (output of the 5000-word flip-flop) produces a DROG2 signal to read out from section B. A low $\overline{\text{MS50}}$ output produces a DROG1 signal to read out from section A.

During a write operation, a low MS50A signal alerts the write-input circuit of section B, and a low MS50A signal alerts the write-input circuit of section A.

The 5000-word flip-flop is controlled by the p4 digit of the storage address from register C, which specifies whether the band is above or below address 5000.

3-117. MEMORY-SELECTION CIRCUITS

3-118. BAND-SELECTION FLIP-FLOPS. The MS2, MS4, MS6, and MS8 band-selection flip-flops generate additional outputs (for example, signal MS2A1) which are applied to the set-stop circuit.

3-119. FAST/STANDARD FLIP-FLOP. The fast/standard flip-flop determines whether the desired location is in the fast or standard-access area of section A or section B storage. The RMSN signal resets this flip-flop. The OF2+ signal at gate 74 (drawing 1-20C) inhibits this gate during an overflow or c + 1 condition.

The C33 and C34 signals from register C indicate whether the third bit of the p4 digit is a 1 or a 0. If it is a 0, the p4 digit is less than 4 and the desired address is in locations 0000—3999 (section A) or locations 5000—8599 (section B) of standard memory. If the third bit of p4 is a 1 (the digit is 4 or 9) the desired address is in locations 4000 — 4999 (section A) or 9000 — 9199 (section B) of fast memory. On drawing 1-20C, gates 62 and 63 and buffer 64 produce outputs, set FM3 or set FM4, when an address is between 8600 and 8999. These outputs set the fast/standard flip-flop for fast-memory access of section B.

CONTENTS

Heading **Title** **Page**

SECTION 4. THEORY OF OPERATION

TABLE OF CONTENTS (cont)

SECTION 5. INSTRUCTIONS

ILLUSTRATIONS

TABLES

Section 4

THEORY OF OPERATION

4-1. INTRODUCTION

Sections 2 and 3 of this manual have presented the logical elements, and their combination into specific Processor circuits. This section will explain computer functions which rely on various configurations of these fundamental circuits. Instructions which are not described in this section are discussed generally in section 5. A condensed description of all instructions is contained in manual 5. Drawings referenced in this section are contained in manual 2.

4-2. BASIC OPERATION CYCLE

An operation cycle consists of the locations and the subsequent execution of a specified instruction. Usually, the computer completes an operation cycle in four steps:

(1) Search for instruction,

(2) Staticize instruction,

(3) Search for operand, and

(4) Execute instruction.

Certain instructions, such as register-to-register transfers, may eliminate the search for operand.

The exact drum-address location sought for by the search for instruction usually is specified by the previous instruction. When the new instruction is located, the search step ends, and the staticize step is initiated. The entire instruction word is transferred from the drum to register C. The p9 and p10 digits (operation code) of the word are transferred to the static register where they are decoded and encoded into function signals which control the operation of the computer (figure 4-1). The computer progresses into the search-for-operand step if the operation code indicates the need for an operand to complete the instruction. Location of the operand initiates the execute step, during which the instruction is completed. A completed instruction produces an ending pulse which clears the static register, generates search signals, and starts the sequence for another cycle of operation.

4-3. SEARCH-FOR-INSTRUCTION STEP

A search for instruction must do the following:

Match the stored address with the timing-band address;

Select from 25 (or 46) drum bands the one specified by the stored address; and

Activate the read-write heads for the correct word location and place that word on the M lines.

A distinct drum address is read from the timing band for each word time. This address repeatedly fixes the surface of the drum relative to the read-write heads. When the timing-band address is compared with an address previously stored in register C, the selection logic circuits are able to determine at which time the read-write heads should be activated to read the desired instruction word.

Viewed in cross-section (figure 4-2), the drum is divided into four quadrants which are designated 00, 01, 10, and 11. Each quadrant contains 50 words for a total of 200 words around the drum circumference. The timing band, which is a permanent data band around the drum contains a discrete address for each word location. This address is detected one word time before the actual word, thereby permitting sufficient time for stored- and detected-address comparison and for the generation of control signals.

Along the length of the drum are read-write heads which are distinct for each band. The timing-band head operates constantly on read. All other heads are activated for read or write only when the timing-band address matches the stored address.

4-4. STANDARD- AND FAST-ACCESS STORAGE. The drum band that is chosen for storing information is determined by the stored address. Depending on whether the standard 5000-word drum or the expanded 9200-word drum is used, either one of 25, or one of 46 bands may be chosen (refer to table 3-8). Five bands between address locations 4000 through 4999, and three bands between locations 8600 through 9199 if using expanded storage, are referred to as fast-access bands. These fast-access bands have four read-write heads per band, corresponding to the four drum quadrants. Depending on the drum position at any given time, the head closest to the quadrant containing the desired location will be selected. Each standard-access band has one read-write head, corresponding to head 00 on figure 4-2. The maximum number of word times necessary to reach a location in standard-access operation is 199; this occurs when the desired location has just passed under the 00 head. If fast-access operation were used in this instance, access time would be reduced by three-fourths. This is the maximum saving in time realized by using the fast-access storage bands.

4-5. FAST-ACCESS OPERATION. Differentiation between standard- and fast-access operation is established by the presence or absence of a 1 in the four-weighted bit of the p4 digit in the stored address. This bit is detected by the fast/standard flip-flop (drawing 1-20B). For computers with expanded storage, two further differentiations are necessary. If the stored address is greater than 5000, the 5000-word flip-flop (drawing 1-22B) is set for a low MS50 output, thus alerting only the B portion of the drum (refer to the discussion under heading 3-116). This differentiation applies to both standard- and fast-access storage. Detection of addresses 86XX, 88XX, or 9XXX in expanded storage generates a low FM (fast-access memory) signal from the fast/standard flip-flop (drawing 1-20B) exactly as a 4XXX address does in 5000-word storage. Once the A or B portion of the drum is selected with expanded storage, all logical operations are identical to those required for the 5000-word drum.

4-6. FAST-ACCESS HEAD SELECTION. Head selection in fast-access storage depends on where the desired address is on the drum and what its position is with respect to the read-write heads when it is called for. The relationship between drum address and read-write heads is fixed (figure 4-2).

The drum, for address identification purposes, may be thought of first as halves and then as quarters with distinct relationships between drum halves and quarters and between the addresses contained in each half and quarter. Two drum-address differences exist between drum halves. The address MSD's on one half are always OXX while they are 1XX on the other half. Each drum half contains two quadrant identities (figure 4-2) whose MSB's match within that half but differ between the halves.

Opposite quadrants are similar in two ways. The middle digit of addresses in opposite quadrants is always 5 or greater for one quadrant pair and less than 5 for the other quadrant pair. The LSB's of opposite quadrant-pair identities are always identical (10—00, 01—11). These drum-address relationships are important in both fast- and standard-access quadrant identification and head selection.

The fast-access storage address contained in register C is a four-digit number from 4000 to 4999. The address read read from the timing band at any time is a three-digit number from 000 through 199. Both addresses are expressed in biquinary code, a characteristic of which is used with the fixed drum-address relationships to simplify quadrant identification and head selection. As shown in table 4-1, if the MSB and the LSB of a biquinary digit are quarter-added (binary addition disregarding the carry), the result indicates whether that digit is odd or even. This principle is most important when applied to the p3 digit of the stored address. This digit (the 100's digit) may be any number from 0 through 9, but any number above 1 is meaningless to the drum because drum-address locations extend only to 199 within a band. Therefore, the odd or even quarter-add result of the p3 stored-address digit is sufficient to identify the desired location as in the 100-199 half of the drum (figure 4-2, quadrants 10-11) or the 000-099 half (quadrants 00-01). From this point on, quadrant identification and head selection are a simultaneous process of elimination which is accomplished by quarter-adds of various bits from the p2 and p3 digits of both addresses.

Given a certain pattern of quarter-adds and direct digit comparisons, computer logic solves the following problem:

> Address 4XXX specifies a word location in quadrant yy at the same time that the 00 head is reading word location XXX in quadrant zz. Which one of four possible heads is now being approached by the specified location?

To illustrate the solution to this problem, assume address 4419 is stored in register C (S lines) when timing-band address 119 is detected on the M lines. In address 4419, location 019 of band 4400 is specified. Address 119 specifies that location 119 is under the 00 head in the timing band. The problem is to activate the next head under which drum location 019 in band 4400 will pass.

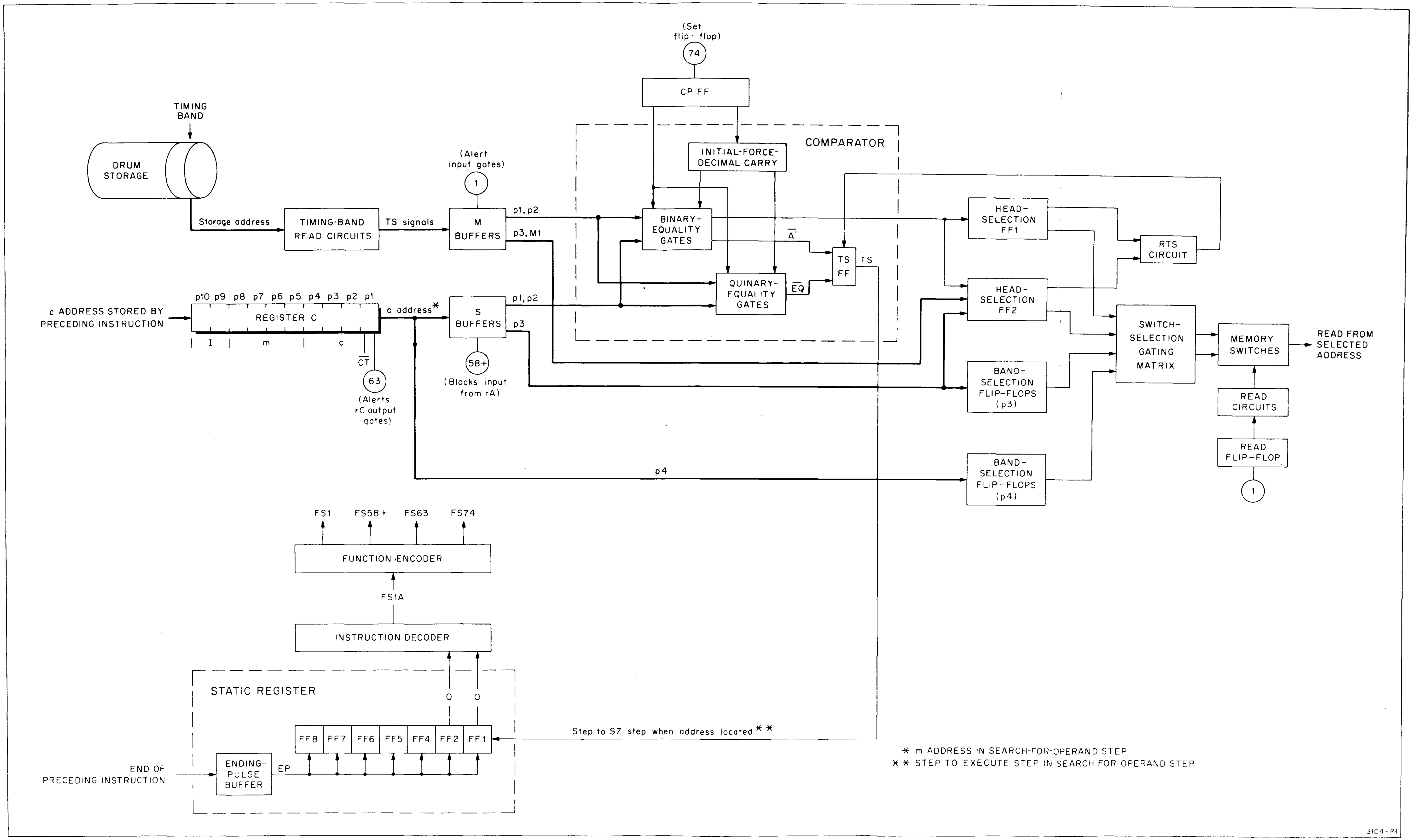
Location 019 is in the 00 quadrant. The 00 head of the timing band is, at this instant, reading location 119 in the 10 quadrant. The read-write head over the 00 quadrant, which contains the desired drum location, is head 10 (figure 4-4). Therefore, head 10 must be activated by Processor logic so that it will read out of location 019 when that location reaches the head.

To determine whether the p3 digit is even or odd, the S1 and S4 bits are quarter-added (refer to table 4-1). If the digit is odd, the number represented is either 1 (41XX), 3 (43XX), 5 (45XX), 7 (47XX), or 9 (49XX). If the digit is even, the number represented will be 2, 4, 6, or 8. Either even or odd identifies a half of the drum since an even digit places the address in the 000-through-099 half, and an odd digit selects the 100-through-199 half. In the example shown in table 4-1, the result is even, thus restricting the address to either 019

Table 4-1. Oddness and Evenness of Biquinary Combinations

	Biquinary Code				Decimal Equivalent	Even or Odd*
	MSB			LSB		MSB LSB
	0	0	0	0	0	0 and 0 = 0
	0	0	0	1	1	0 and 1 = 1
	0	0	1	0	2	0 and 0 = 0
	0	0	1	1	3	0 and 1 = 1
p3 →	0	1	0	0	4	0 and 0 = 0
	1	0	0	0	5	1 and 0 = 1
	1	0	0	1	6	1 and 1 = 0
	1	0	1	0	7	1 and 0 = 1
	1	0	1	1	8	1 and 1 = 0
	1	1	0	0	9	1 and 0 = 1

*0 = even; 1 = odd



* m ADDRESS IN SEARCH-FOR-OPERAND STEP

** STEP TO EXECUTE STEP IN SEARCH-FOR-OPERAND STEP

Figure 4-1. Search Step of Basic Operation Cycle

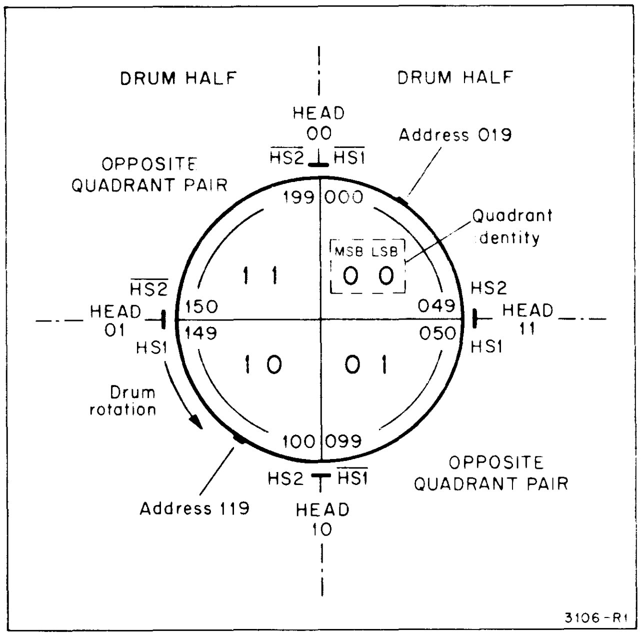


Figure 4-2. Drum-Address Relationship

or 069. To reduce the identification to a specific quadrant and to activate the read-write head closest to that quadrant, the S4 and S1 odd-or-even result is quarter-added to M1 of the p3 digit and to A or $\overline{A}$, which is the result of quarter-adding the M4 and S4 bits of p2. The M1 bit of the p3 digit is significant because the timing-band address can have only a 1 or a 0 in this position (the 100's digit). When the result of quarter-adding (S4 + S1) to M1 is odd, the drum half that is not under the 00 head contains the desired location. When the result is even, the drum half under the 00 head contains the desired location. In the example, ($\overline{S4}$ + $\overline{S1}$) + M1 is odd. Therefore, that half of the drum under the 00 head is eliminated along with the heads (00, 11) that are over the drum half at that instant.

If M4 is greater than or equal to S4 (M4 > S4). A is low and is assigned a logical value of 0. If M4 is less than S4 (M4 < S4), $\overline{A}$ is low and is assigned a logical value of 1.

If A is generated (M4 ≧ S4), the quadrant under the 00 head, or its opposite, contains the desired location. The drum half that contains the desired quadrant of the opposite pair has been specified by the quarter-addition of (S4 + S1) + M1. If the result of this quarter-addition is even, the desired location is in the quadrant under the 00 head. If the result is odd, the location is in the opposite quadrant.

If $\overline{A}$ is generated (M4 < S4), the first two digits of the stored address are greater than the timing-band address by 50 or more. This automatically eliminates the quadrant under the timing band and its opposite because the stored address must be either 050 to 099 or 150 to 199 (quadrant 01 or 11).

Returning to the example in figure 4-3, the p3 digit, being even, has established which half of the drum is desired (000-through-099 half). This indication is quarter-added with M1 to yield an odd result. (Note that if the 100's digit of the drum address and the stored address match, the result of this quarter-addition always is even). The M4—S4 quarter-addition yields equality,

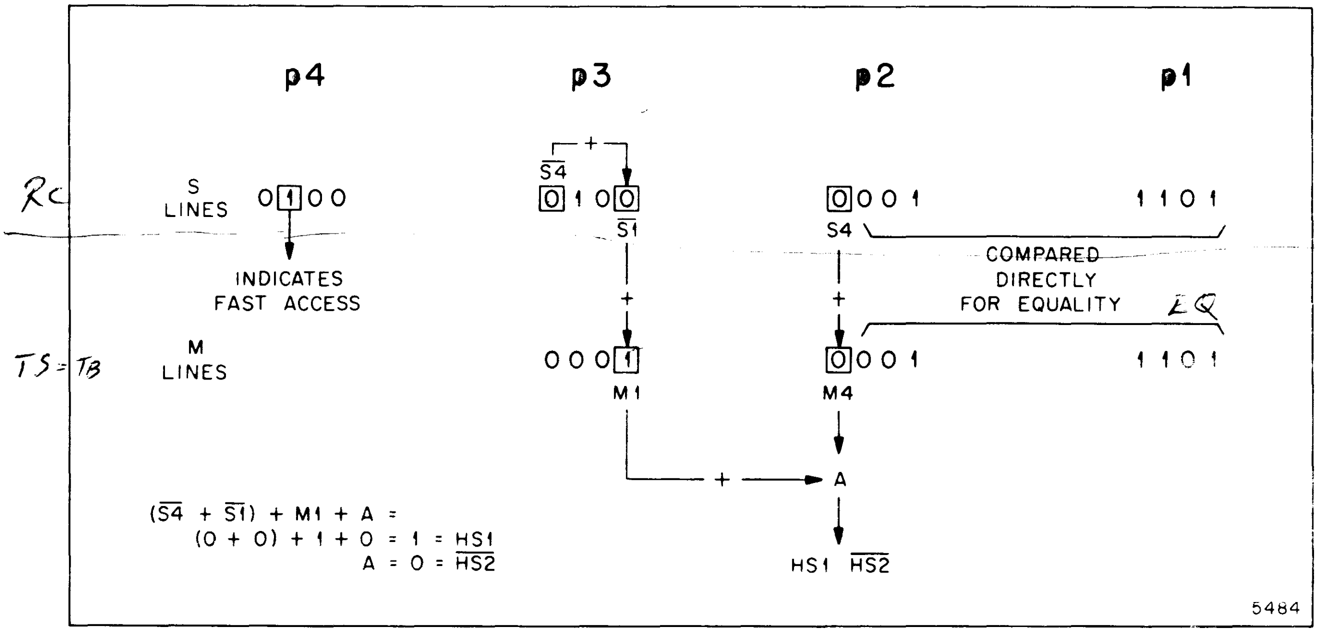


Figure 4-3. Comparison of Addresses for Fast-Access Head Selection

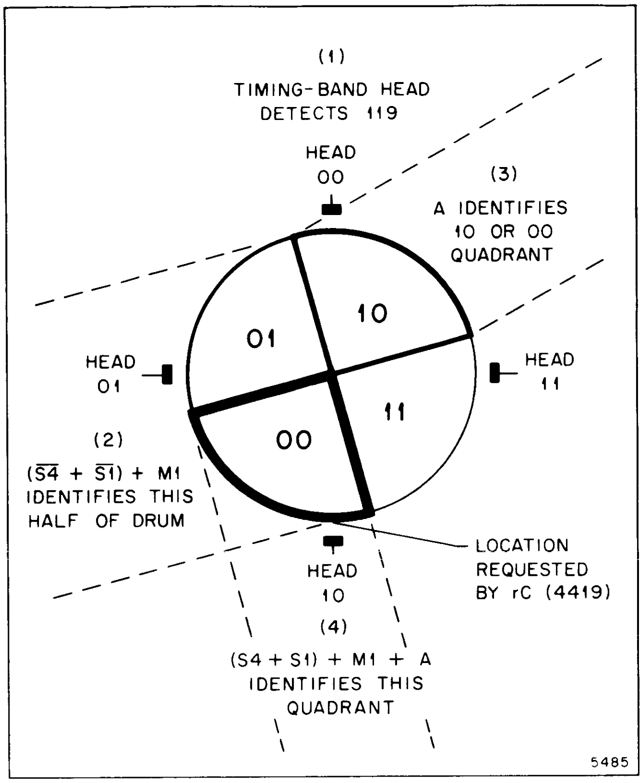


Figure 4-4. Quadrant Identification and Fast-Access Head Selection

which generates an A. Thus the final configuration developed by the example is

$(\overline{S4} + \overline{S1}) + M1 + A = (0 + 0) + 1 + 0 = 1.$

The A means that the quadrant under the 00 head (in this case quadrant 10), or its opposite, contains the desired address. The odd result of $(\overline{S4} + \overline{S1}) + M1$ pinpoints the opposite quadrant (00). Head 10 is approaching quadrant 00. The output of the head-selection-2 flip-flop, HS2, is the MSB of the head identity (figure 4-2). Therefore, $(\overline{S4} + \overline{S1}) + M1 + A$ must generate HS2 = 1. This configuration is present at gate 37C of drawing 1-20B and does generate a 1, establishing the head identity as at least 1x. The value assigned to HS1 is determined solely by A or $\overline{A}$: $\overline{A}$ generates a 1, and A generates a 0. This is possible because the LSB's of opposing quadrants are identical. Therefore, any address containing a middle digit of 5 or greater produces $\overline{A}$ which, in turn, generates HS1 = 1. Less than 5 in S4 (p2) of the stored address, or equality between M4 and S4 (p2) of both addresses, generates A for $\overline{HS1}$ = 0.

In the example in figure 4-3 an A is developed. Therefore, $\overline{HS1}$ = 0, thus selecting a final head identity of 10 (HS2, $\overline{HS1}$). Once the correct head is selected, its identity is stored in the head-selection flip-flops (drawing 1-20B) until the read operation is completed.

As a further illustration of the fast-access head-selection process, the example is shown diagrammatically in figure 4-4. The principles illustrated in figure 4-4 may be applied to any combination of stored address versus timing-band address.

Thus far, in having selected the correct head for fast-access operation, only portions of the p4, p3, and p2 digits have been sampled. These samples are sufficient to identify the address quadrant and to select the correct head, but they are not sufficient to complete the readout from the drum. Information cannot be read from the drum until the TS signal from the time-selection (TS) flip-flop (drawing 1-14B) is sampled at gate 42A of the main-storage-read flip-flop (drawing 1-21B) at t8b–. The TS signal must be low at this time, or the read flip-flop will not be set. Final indication of when to read is developed at gates 20, 21, and 22 of the TS flip-flop. Assume that all other inputs to buffers 80 and 81 are low. At this point the direct comparison between all of the p1 address bits and three of the p2 address bits (figures 4-2 and 4-3), in conjunction with head selection, completes the read enable. Referring to the diagram of the example (figure 4-4) and relating the results of the example to actual head positions, note that head 10 is directly over the desired address location. Therefore, TS must go low during this word time if the instruction word is to be read out onto the M lines beginning with t0 = p0 of the next word time. What occurs at gates 20, 21 and 22 of the TS flip-flop now must be related to the two addresses.

A t0b+ signal at buffer 82 initially sets the TS flip-flop at t1b. The $\overline{A'}$ signal, which is the result of the M4—S4 comparison of the p1 address digits, is sampled at t2b (gate 20). Since M4 equals S4 at t1 (figure 4-3), $\overline{A'}$ is high. Simultaneously, a comparison of the first three bits of the p1 digits (quinary-equality circuit) generates a high $\overline{EQ}$ signal at gate 21. Both gates 20 and 21 have a high input at t2, thus preventing the TS flip-flop from being reset. Gate 21 samples $\overline{EQ}$ at t3b and again detects a high signal. At t4 the correct head for fast-access operation has been selected and, because a head other than 00 was selected, a low RTS signal is generated (drawing 1-20B). The RTS signal is applied to gate 22 of the TS flip-flop together with SM (standard-access memory).

Since SM is high during fast-access operation, the low RTS signal is blocked and does not reset the TS flip-flop at t4. The TS signal remains low and a drum-read operation is initiated at the beginning of the next word time. The TS flip-flop could have remained set for timing-band address 019, 069, 119, or 169 since any of these addresses when combined with a stored address of 4419 will not reset the TS flip-flop. The significant address is determined solely by which head is selected.

4-7. STANDARD-ACCESS OPERATION. The principle of standard-access operation is identical to the process described for head selection during fast-access operation except that only one head can be selected, the 00 head. If the address comparison yields $\overline{A}$ or any of eight input combinations of gates 37A through 37H (drawing 1-20B), a low RTS signal is developed which resets the time-selection flip-flop, thus preventing a drum read. The TS flip-flop remains set only when the stored and detected addresses are identical. A typical example of standard-access operation yields the following configuration when the addresses match:

	p4	p3	p2	p1
S Lines	0010	0001	0001	0010
M Lines		0001	0001	0010

$$\begin{aligned} &(S4 + S1) + M1 + A \\ &= (0 + 1) + 1 + 0 = 0 = \overline{HS2} \\ &A = 0 = \overline{HS1} \end{aligned}$$

Stored address 2112 is matched with timing-band address 112. Quarter-adding in the order established for fast-access operation develops a 0 for the $\overline{HS2}$ identity. Signal $\overline{HS1}$ also is 0 because A is low. Therefore, only the 00 head is activated. The equality of the address comparison sets the TS flip-flop, and drum data is gated onto the DM lines.

4-8. BAND SELECTION. Address location is a two-dimensional process which first locates the address on the drum circumference and then specifies a particular band along the length of the drum. A band is selected by information in the p4 and p3 digits of the stored address.

The 1000's digit (p4) is decoded to determine fast- or standard-access memory (FM or SM signal) by the detection of a 4. In Processors with 9200-word drum, the 1000's digit is further decoded by gates 62 and 63 (drawing 1-22B), which are made permissive by an 86XX, 88XX or 9XXX address. These addresses indicate fast-access bands in expanded storage. If a 4XXX, 86XX, or 88XX address is detected, FM is generated and no further decoding of the 1000's digit is required. Address 9XXX is unique because it generates FM, but requires further decoding. If a fast-access indication is not detected, only the first two bits are significant because the digit is then restricted to a maximum value of 3. The maximum value of 8, which is possible in expanded storage, is treated as a 3, but the 5 is stored for selecting the B portion of the drum. The MS10-$\overline{MS10}$ and the MS20-$\overline{MS20}$ outputs of the band-selection flip-flops (drawing 1-20B) are the decoded results of the four possible bit combinations (00, 01, 10, or 11) of the first two bits of p4.

Processors that have a 9200-word-storage capacity decode the MSB of p4 in the stored address at gate 65A or 65B of the 5000-word flip-flop (drawing 1-22B). If a 1 bit is detected, a value of 1000 (5), 1001 (6), 1010 (7), 1011 (8), or 1100 (9) is implied for the p4 digit. The flip-flop is set to produce a low MS50 output, and only that section of the drum (B) which contains addresses from 5000 to 9199 is alerted.

After the A or B section of the drum is chosen, band-selection decoding of the p4 digit is identical to the procedure described in the preceding paragraph.

The 100's digit of the address is decoded by the band-selection flip-flops (p3) (drawing 1-20B). The flip-flops have five outputs (MS0, MS2, MS4, MS6, and MS8), each corresponding to one band of 200 words.

To illustrate band selection, assume a stored address of 34XX. The two least significant digits are not important for band selection. The address indicates that band 3400—3599 in standard-access memory is desired. The 1000's digit is decoded to yield MS10—MS20. The decoded 100's digit makes gate 48B (drawing 1-20B) permissive to generate MS4. As shown in the memory-selection matrix (drawing 1-20B), the signal combination at gate 18A selects only switch driver FS34 (3400—3599). The same combination results and the same switch driver is chosen for an address of 84XX in expanded storage, but the presence of an 8 in p4 alerts the B portion of the drum. Address 34XX in expanded storage would have alerted the A portion.

4-9. STATICIZE-INSTRUCTION STEP

The primary function of the staticize-instruction step (figures 4-5) is to store the two operation-code digits (p9 and p10) in the static register (STR) to control the execution of the instruction. Other functions of the staticize step are to store the ten digits of the instruction in register C and to store the p7 digit of the instruction word in the multiplier/quotient counter (MQC). The word stored in register C consists of the two operation-code digits, the address of the operand (m address), and the address of the next instruction (c address). The p7 digit of the instruction word specifies the number of shifts necessary in a shift instruction, and the card-reader output stacker to be used in a select-stacker instruction. If index registers are included, the staticize step includes an inspection of the operation code of the instruction. If the instruction code calls for modification of the m address, and extra word time is added to the staticize step for this modification (heading 4-13).

At t10B of each word time of the search step, the output of the TS flip-flop is sampled to determine whether the search was successful. If the search was successful, the TS signal is low at gate 7 of STR FF1 (drawing 1-2B). Function signal 1 alerts this gate, which also samples the $\overline{OF}$ output of the overflow flip-flop. The conditions that generate a high $\overline{OF}$ signal are described under heading 3-72. The presence of the low $\overline{OF}$ signal at gate 7 means that no abnormal conditions are present. When gate 7 is permissive to these signals, it generates a high RCT1 signal which restores the CT flip-flop (drawing 1-14B), and it sets STR FF1 to generate a low STR1 output.

At the beginning of every staticize step, the CT flip-flop is restored to prepare for the search-for-next-instruction step. If the staticized instruction requires an operand, the CT flip-flop is set at the end of the staticize step to search for the operand stored in the m address.

When STR FF1 is set to a 1 output (figure 4-5), the input to the instruction decoded is 0000 0X01, since only FF1 has changed. The decoder converts this combination to FS2A, which drives the function encoder to generate FS2, FS2+, and FS62. These signals control the staticize step.

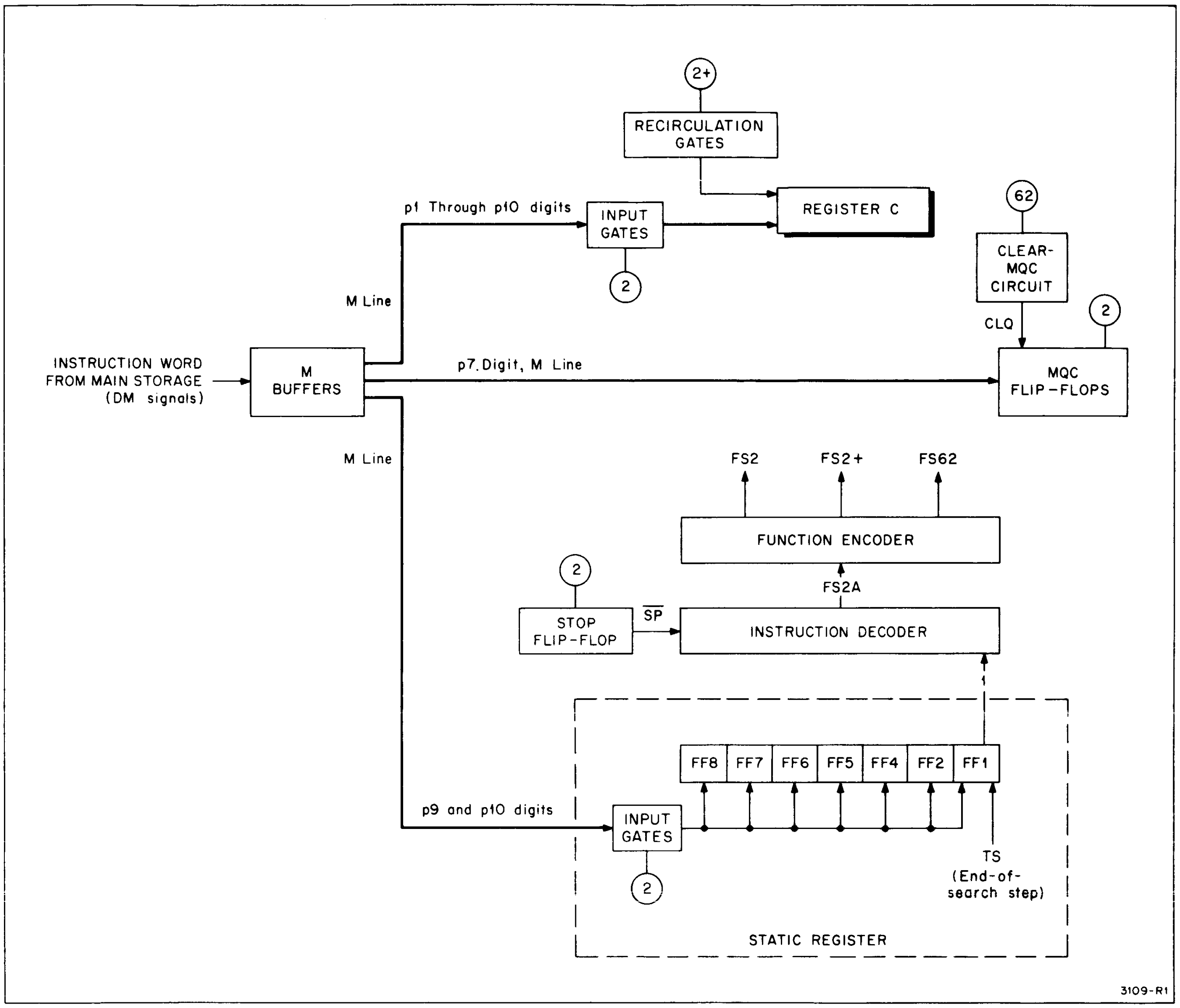


Figure 4-5. Staticize Step

4-10. STORING THE p7 DIGIT. Before storing the p7 digit in the four flip-flops of the MQC, the flip-flops must be cleared of previous information. At t0B of the staticize step, FS62 generates the CLQ signal at gate 19 of the clear-MQC circuit (figure 4-5 and drawing 1-16B). The CLQ signal restores the four flip-flops at gates 5, 6, 7, and 8.

Function signal 2 alerts the input gates of the MQC flip-flops at t7B, the time during which digit p7 is on the MT lines. The four bits of digit p7 are stored in the flip-flops for future use if the staticized instruction is a shift or select-stacker instruction.

4-11. STORING THE INSTRUCTION WORD. At t0A of the staticize step, the p0 digit of the instruction word from memory is present on the DM or DM′ lines of the read-output circuits (drawings 1-21B and 1-22B). These signals go directly to the M buffers and are on the M lines at t0B. Function signal 2+ blocks the recirculation gates of the four subregisters of register C, clearing the register; function signal 2 alerts the input gates to the digits of the instruction word on the M lines. Thus, at t0B of the staticize step, the p0 digit of the instruction word enters the permissive input gates of register C.

4-12. STATICIZING THE INSTRUCTION. Once they have performed their functions, the function signals generated by the staticize step must be removed. At t8B of the staticize step, FS2 is low at gate 114 of the static register (drawing 1-2B). This restores STR FF1 (0 output); the configuration at the input to the decoder is changed, and different function signals are developed by t11B.

At t9B the p9 digit of the instruction word is present on the M lines. (See figure 4-5.) Function signal 2 alerts the input gates of FF1, FF2, and FF4 at t9B so that the p9 digit can enter. To ensure that no function signals are generated between the time that digit p9 is staticized (t10B) and digit p10 is staticized (t11B), FS20 makes gate 20 of the stop flip-flop permissive to generate a high $\overline{\text{SPX}}$ signal (drawing 1-5A) at t10B, which blocks the instruction decoder.

At t10B, the p10 digit is present on the MT lines. Function signal 2 alerts the input gates of FF5, FF6, FF7, and FF8 of the STR at t10B so that the p10 digit can enter the register. By t11B the two instruction digits are staticized. Function signals required to execute the staticized instruction are available from the instruction decoder at t0A, and from the function encoder at t0B.

4-13. INDEX-REGISTER MODIFICATION

Index registers provide a means of altering an existing address in register C by adding that address to the contents of a special register. After the original address is modified, it is used to continue the instruction pattern. Address modifications with the index registers eliminate the necessity for returning to main storage for a new address, thus reducing program execution time.

Figure 4-6 is a block diagram of the index-register modification. It is assumed in this figure that register 1 was selected during the staticize step by setting STR FF3 and that, as a result, the index-register read-out flip-flop was set to generate function signals to perform the modification. (The timing of the index-register modification is shown in figure 4-7.)

If index registers are included in the system, the staticize step of the basic instruction cycle includes, in addition to the functions described under heading 4-9, an inspection of the instruction code to determine whether the operand is to be modified. The timing of this inspection during the staticize step is shown in figure 4-7. The third bit of the p9 digit and the sign bit of the instruction word are stored in register C to determine whether modification is required and, if it is required, which of the three index registers is specified. If modification is not required for the operand, the sign of register C is plus (the first bit of the sign digit, p0, is a 0), and the third bit of the p9 digit is a 0. In this case both the rC-sign flip-flop (drawing 1-24B) and STR FF3 will be restored, and the index-register read-out flip-flop will not be set. The staticize step will not require an additional word time.

If one of the three index registers is specified, either or both the rC-sign flip-flop and the STR FF3 flip-flop will be set. The set signals from one or both of these flip-flops set the index-register read-out flip-flop. The setting of the read-out flip-flop generates GEN $\overline{\text{SP3}}$ and B58+ signals which block the instruction decoder for one word time. During this additional word time, the contents of the specified index register are read out of the register onto the M lines and the m address from register C is read out onto the S lines.

The information on the M and S lines is sent through the add circuits of the Processor into the index-register adder (drawing 1-25B). The add circuits of the Processor consist of the complementer circuit, the decimal-carry adder, and the gates that determine carry. The resulting modified m address now can be returned to register C, except that provision must be made for the possibility that the resulting address may not be within the original band.

If the address is not within the original band, the digits 9800 are added to the modified m address before it is stored. Because this addition will affect only the p7 and p8 digits from the adder, the p5 and p6 digits from the adder are returned directly to register C. The p7 and p8 digits are sent to the S lines to be returned to the index-register adder.

If band modification is not needed, the add-9800 flip-flop is not set and the p7 and p8 digits go through the adder unchanged and return to register C. If band modification is necessary, the add-9800 flip-flop is set. The outputs of the add-9800 flip-flop place the digits 9800 on the M-line inputs to the index-register adder. The digits 9800 are added to the modified m address and the resulting sum is an address which is within the original band. The modified p7 and p8 digits are returned to register C. When all four digits are stored in register C, the index-register read-out flip-flop is restored so that the instruction decoder no longer is blocked. A search is made for the operand in the modified m address and the instruction is executed.

4-14. SELECTING INDEX REGISTER. The index register to be used during the modification is selected during the staticize step and is designated by the sign bit of the instruction and the third bit of p9 digit. These bits are stored in the rC-sign flip-flop and STR FF3. If either or both are set, the instruction is to be modified and the index-register read-out flip-flop is set. If the rC-sign flip-flop is set, the C signal makes set gate 17 permissive. If STR FF3 is set, the STR3 signal makes set gate 18 permissive. If both flip-flops are set, both gates are made permissive. If neither is set, modification is not required and the index-register read-out flip-flop is not set.

4-15. SETTING INDEX-REGISTER READ-OUT FLIP-FLOP. The index-register read-out flip-flop is set if modification is required. The flip-flop then generates low FS100, FS100A−, a high GEN $\overline{\text{SP2}}$ or GEN $\overline{\text{SP1}}$ signal, and a high B58+ signal. Function signal 100 reads the information out of the index register and the index-register adder. Function signal 100A− generates the Bt0-3B− and Bt0-3B+ signal (drawing 1-18A), which read the m address out of register C and clear the m-address portion of register C. The high GEN $\overline{\text{SP}}$ and B58+ signals block the instruction decoder for one word time while the modification is taking place. The B58+ signal also generates FS58+ to prevent the contents of register A from reading out onto the S lines.

4-16. ADDING REGISTER C TO INDEX REGISTER. The m address from register C is read onto the S lines by the Bt0-3B− signal generated by FS100A−. The m-address portion of register C also is cleared by the Bt0-3B+ signal to prepare for the modified m address.

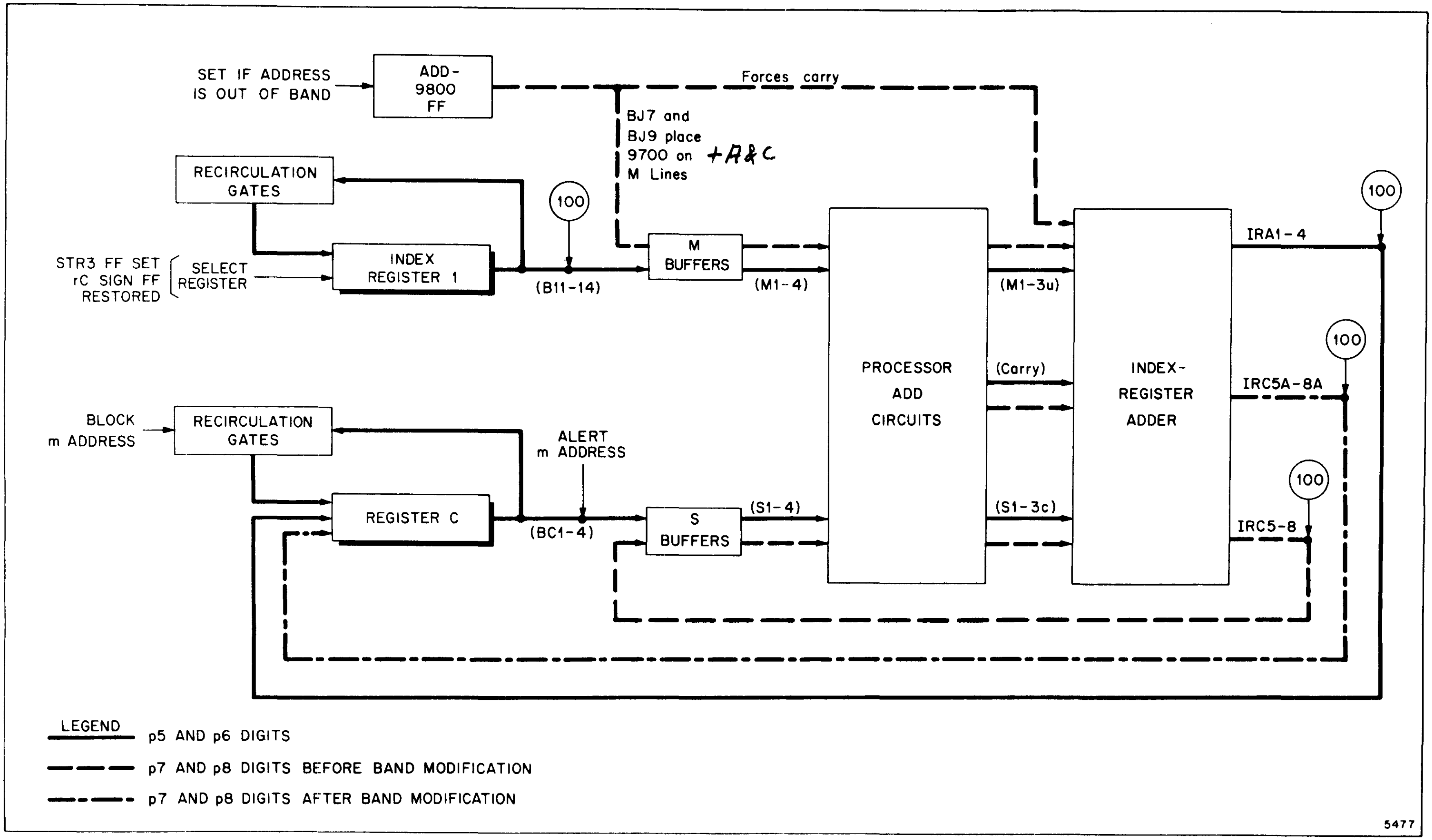


Figure 4-6. Index-Register Modification

The information on the S lines (BC1 through BC4 signals) is sent through the complementer circuit into the index-register adder. The contents of the selected index register are transferred from the register to the M lines by FS100, with the information recirculating in the index register. The information on the M lines is sent through the decimal-carry adder into the index-register adder, which adds the m address from register C and the information from the index register, and returns the p5 and p6 digits to register C. The p7 and p8 digits are sent to the S lines and again through the adder. If band modification is required, the add-9800 flip-flop is set to place the digits 9800 on the M lines. The resulting modified p7 and p8 digits then are returned to register C. The index-register read-out flip-flop is restored at t10B of the additional word time and, since the instruction decoder no longer is blocked, a search is initiated for the modified m address.

4-17. SEARCH-FOR-OPERAND STEP

An operand is defined as any one of the quantities entering into or resulting from an operation. An operand may be an argument, a result, a parameter, or a next-instruction address. Because the meaning of the four digits of the m address varies with the type of instruction, the instructions are divided into the following groups:

Instructions involving a normal search-for-operand step in which the search for the storage location designated by the m address (p8, p7, p6, and p5) is conducted in the same manner as the search-for-instruction step (characterized by a p9 digit of 0 or 5). This group is composed of the arithmetic, transfer, extract, and superimpose instructions:

Arithmetic: add (70[1]); divide (55); multiply (85); and subtract (75).

Transfers between main storage and registers: register A (60, 25); register L (50, 30); register X (05, 65).

Extract (35) and *superimpose* (20).

Instructions with a search-for-operand step in which the search is only for the band designated by the p7 and p8 digits (characterized by a p9 digit of 1 or 6). This group is composed of input-output instructions which search for a main-storage band in which output or input information is stored as follows: printer (11, 16); read-punch (81, 46); card reader (96).

Instructions in which the p8 through p5 digits represent an alternate c address. These instructions comprise a test, the result of which determines whether the next instruction is taken from the p8 through p5 digits or from the p4 through p1 digits. This group is composed of instructions that compare the contents of register A with the contents of register L, and of four instructions that test the input-output units as follows:

Register tests: compare for equality (82); compare for greater than 87).

Input-Output: test printer (27); test read-punch (22); test reader (72, 42).

Instructions in which the p7 digit is the only significant digit of the m address. These instructions do not involve search because the p7 digit is stored in the MQC during the staticize step. This group is composed of the two shift instructions, in which the p7 digit designates the number of places to shift. It also includes the two select-stacker instructions, in which the p7 digit designates the stacker. The shift and select-stacker instructions are as follows:

Shift: shift left (37); shift right (32).

Select Stacker: select RPU stacker (57); select CR stacker (47).

The code combinations for the steps of all instructions are shown in table 4-2. The table is divided into instructions that require one execution step and those that require more than one. It is subdivided further into instructions that require a search-for-operand step and those that do not. The third column, STR code, lists the code combinations for each step of each instruction after it has been staticized. The code positions marked by X are bits that are unnecessary to the staticize step. The X in the STR3 position of the LSD is included only to balance each combination with the eight bits of the biquinary code.

4-18. INSTRUCTION-CODE CHARACTERISTICS. An instruction that requires a normal search-for-operand step has a p9 digit of 0 or 5. Both the codes for 0 (0000) and 5 (1000) contain two 0's in their lowest order bits. When an instruction requiring a normal search-for-operand step is staticized, the $\overline{\text{STR1}}$ and $\overline{\text{STR2}}$ outputs of the static register generate FS1A to initiate a search operation. The $\overline{\text{STR2}}$ output also sets the CT flip-flop so that the search is for the m address.

An input-output instruction requiring a search for a band has a p9 digit of 1 or 6. When this type of instruction is staticized, FS1A is not generated because the $\overline{\text{STR1}}$ output is high. The function signal that is generated for the particular input-output instruction drives the function encoder to produce some of the same signals used during a normal search operation, as well as additional function signals associated with the instruction.

4-19. LOCATING THE OPERAND. The search-for-operand step is identical to the search-for-instruction step. Control of the operand search, however, differs from control of the instruction search. Note that figure 4-1 is a block diagram of the search-for-operand step as well as the search-for-instruction step; differences between the two steps are described by notes on the figure. Although FS63 alerts both the m-address output gates and the c-address output gates of register C, the CT flip-flop determines which address is to be read out and compared. When an instruction is staticized, the output of STR FF2 determines whether the operand is necessary.

[1] Mnemonics and functions of machine codes of instructions are contained in table 2-2.

Table 4-2. Instruction-Code Combinations

Instruction	Biquinary Code	STR Code 8 7 6 5 4 3 2 1	Steps
Instructions with One Execution Step (Search)			
05 (MX) : (m) → rX	0 0 0 0 1 0 0 0	X X X X X X 0 0 0 0 0 0 1 X 0 1	Sc 00 MX 06
20 (SI) : Superimpose (m) on (rA) → rA	0 0 1 0 0 0 0 0	X X X X X X 0 0 0 X 1 0 0 X 0 1	Sc 00 SI 21
25 (MA) : (m) → rA	0 0 1 0 1 0 0 0	X X X X X X 0 0 0 X 1 0 1 X 0 1	Sc 00 MA 26
30 (ML) : (m) → rL	0 0 1 1 0 0 0 0	X X X X X X 0 0 0 X 1 1 0 X 0 1	Sc 00 ML 31
35 (EX) : Extract (m) from (rA) → rA	0 0 1 1 1 0 0 0	X X X X X X 0 0 0 X 1 1 1 X 0 1	Sc 00 EX 36
50 (LM) : (rL) → m	1 0 0 0 0 0 0 0	1 0 0 0 0 X 0 0 1 0 0 0 0 X 0 1	LM-Sc 50 LM-Ex 51
60 (AM) : (rA) → m	1 0 0 1 0 0 0 0	1 X 0 1 X X 0 0 1 X 0 1 0 X 0 1	AM-Sc 60 AM-Ex 61
65 (XM) : (rX) → m	1 0 0 1 1 0 0 0	1 X 0 1 X X 0 0 1 X 0 1 1 X 0 1	XM-Sc 65 XM-Ex 66
Instructions with One Execution Step (No Search)			
12 (TCU) : 80 CC → USS-6 (rA, rL, and rX) → rA, rX	0 0 0 1 0 0 1 0	0 X 0 1 0 X 1 0	TCU 12
17 (TUC) : USS-6 → 80CC (rX and rA) → rA, rX, and rL	0 0 0 1 1 0 1 0	0 X 0 1 1 X 1 0	TUC 17
47 (SRS) : Select output stacker of reader	0 1 0 0 1 0 1 0	0 1 0 0 1 X 1 0	SRS 47
57 (SPS) : Select read-punch stacker 1	1 0 0 0 1 0 1 0	1 X 0 0 1 X 1 0	SPS 57
67 (ST) : Stop	1 0 0 1 1 0 1 0	1 X 0 1 1 X 1 0	ST 67
72 (FR) : Feed one card into reader	1 0 1 0 0 0 1 0	1 X 1 0 0 X 1 0	FR 72
77 (AL) : (rA) → rL	1 0 1 0 1 0 1 0	1 X 1 0 1 X 1 0	AL 77
82 (CE) : (rA) : (rL) If (rA) = (rL), go to m If (rA) = (rL), go to c	1 0 1 1 0 0 1 0	1 X 1 0 0 X 1 0	CE 82
87 (CGT) : (rA) : (rL) If (rA) > (rL), go to m If (rA) ≦ (rL), go to c	1 0 1 1 1 0 1 0	1 X 1 1 1 X 1 0	CGT 87
Optional 02 (IL) : m → index register	0 0 0 0 0 0 1 0	0 X 0 0 0 X 1 0	IL 02
Optional 07 (IA) : m — index register → IR, rC and rA	0 0 0 0 1 0 1 0	0 X 0 0 1 X 1 0	IA 07
Instructions with Two or More Execution Steps (Search)			
55 (D) : (m) ÷ (rL) → rA; remainder → rX	1 0 0 0 1 0 0 0	X X X X X X 0 0 1 0 0 0 1 X 0 1 1 0 0 0 1 X 1 1 1 0 0 1 1 X 1 1	Sc 00 D1 56 D2 58 D3 68
70 (A) : (m) + (rA) → rA	1 0 1 0 0 0 0 0	X X X X X X 0 0 1 X 1 0 X X 0 1 1 X 1 0 X X 1 1	Sc 00 A1 71 A2 73

Table 4-2. Instruction-Code Combinations (cont)

Instruction	Biquinary Code	STR Code 8 7 6 5 4 3 2 1	Steps
Instructions with Two or More Execution Steps (Search) (cont)			
75 (S) : (rA) — (m) → rA	1 0 1 0 1 0 0 0	X X X X X X 0 0 1 X 1 0 X X 0 1 1 X 1 0 X X 1 1	Sc 00 S1 76 S2 78
85 (MPY) : (m) x (rL) → rA, rX	1 0 1 1 1 0 0 0	X X X X X X 0 0 1 X 1 1 1 X 0 1 1 X 1 1 1 X 1 1	Sc 00 MPY1 86 MPY2 88
Instructions with Two or More Execution Steps (No Search)			
11 (PR) : Storage → print buffer. Print one line.	0 0 0 1 0 0 0 1	0 X 0 1 X X 0 1 0 X 0 1 X X 1 1	PR1 11 PR2 13
16 (PF) : Paper advance	0 0 0 1 1 0 0 1	0 X 0 1 X X 0 1 0 X 0 1 X X 1 1	PF1 16 PF2 18
22 (TEPB) : Test: read-punch buffer loaded.	0 0 1 0 0 0 1 0	0 X 1 0 0 X 1 0 0 X 1 0 X X 1 1	TEPB 22 Common test 23
27 (TEP) : Test: printer available	0 0 1 0 1 0 1 0	0 X 1 0 1 X 1 0 0 X 1 0 X X 1 1	TEP 27 Common test 23
32 (SHR) : Righ shift	0 0 1 1 0 0 1 0	0 X 1 1 X X 1 0 0 X 1 1 0 X 1 1	SHR1 32 SHR2 33
(rA) into rX and (rX) into rA			
37 (SHL) : Left shift (rA). Transfer 0's to vacated positions.	0 0 1 1 1 0 1 0	0 X 1 1 X X 1 0 0 X 1 1 1 X 1 1	SHL1 37 SHL2 38
42 (TERB) : Test: reader buffer loaded	0 1 0 0 0 0 1 0	0 1 0 0 0 X 1 0 0 X 1 0 X X 1 1	TERB 42 Common test 23
46 (PBM) : Read-punch buffer → I or IT interlace	0 1 0 0 1 0 0 1	0 1 0 0 1 X 0 1 0 1 0 0 1 X 1 1	PBM1 46 PBM2 48
62 (ZS) : Zero suppress USS-6 word in rA and rX	1 0 0 1 0 0 1 0	1 X 0 1 0 X 1 0 1 X 0 1 0 X 1 1	ZS1 62 ZS2 63
81 (FP) : 0 or 0T interlace → read-punch buffer. Start read-punch unit.	1 0 1 1 0 0 0 1	1 X 1 1 0 X 0 1 1 X 1 1 0 X 1 1	FP1 81 FP2 83
96 (RPM) : Card-reader buffer → J or JT interlace	1 1 0 0 1 0 0 1	1 1 0 0 1 X 0 1 1 1 0 0 1 X 1 1	RBM1 96 RBM2 98

As shown in table 4-2. the STR2 position of the staticized instruction is always 0 when a search for operand is required.

The $\overline{\text{STR2}}$ output of the static register is sampled by gate 12 of CT flip-flop (drawing 1-14B) at t10B of the staticize step. If a search must be made for the m address, the $\overline{\text{STR2}}$ signal is low, making gate 12 permissive. When the gate is permissive, the CT flip-flop is set to generate CT. which controls readout of the m address from register C before the execution step. When gate 12 is blocked by high signal $\overline{\text{STR2}}$, the CT flip-flop is restored to $\overline{\text{CT}}$, which controls readout of the c address after the execution step.

The CT signal is present at the m-address output gates of register C at t0B. The LSD of the m address, p5, goes from register C to the S buffers and the S lines and is available at the quinary- and binary-equality gates of the comparator (drawing 1-14B) at t1B. The comparison and memory-selection processes for the search-for-operand step are the same as those for the search-for-instruction step (heading 4-3).

The 60, 50, and 65 transfer instructions require a search-for-operand step in which the m address is a storage location into which information is to be written. Because these three instructions entail writing in storage, time must be allowed to prepare the write circuits. For this reason, the 60, 50, and 65 instructions generate the FS3 for use in the search operation in addition to the normal search (SC) function signals. Function signal 3 alerts gate 8 of STR FF1 to sample the TS signal at t5B. (In normal search, the TS signal is sampled three pulse times later, at t8b.) Sampling at t5B allows the write circuits to prepare for writing by the time the execute step begins. If the TS signal is low at t5B, STR FF1 is set to a 1, initiating the execute step of the 60, 50, or 65 instruction.

4-20. EXECUTE-INSTRUCTION STEP

The execute-instruction step is initiated when the seven bits of the two instruction digits are staticized and either of the two lowest order bits of the p9 digit is a 1 (search-for-operand step is unnecessary). It is also initiated when the search-for-operand step has been completed and STR FF1 has been stepped to a 1 output. In both cases the combination of bits stored in the static register represents a specific instruction and can be decoded by the instruction decoder. The combination in the static register goes to the instruction decoder and function encoder to generate one or more function signals which execute the staticized instruction.

4-21. INSTRUCTIONS WITH TWO EXECUTION STEPS. Many instructions require two execution steps. The add (70) instruction, for example, adds two quantities in the first execution steps and if necessary, complements the sum of the addition in the second execution step. The function signals generated for the first (A1) step of the add instruction include FS64. At t10B of the A1 step, FS64 restores FF1 and sets FF2 at the step gate of the static register (drawing 1-2B). Flip-flop 1 previously had been set to a 1 to accomplish the A1 step. Setting FF2 to a 1 output sets up the combination in the static register for step A2; as a result, a new set of function signals is generated to control the A2 step.

4-22. TIMING OF THE BASIC OPERATION CYCLE The four steps of the basic operation cycle require a minimum of four word times. Instructions requiring no search-for-operand step take only three word times. Instructions with more than one execution step require more than four word times. The timing of the basic operation cycle with and without a search-for-operand step is shown in figure 4-7. The timing of the various operations of the search-for-operand step is identical to that of the search-for-instruction step. Therefore the search-for-operand step shows only the timing of operations unique to it. Although many other operations occur during each word time, only those necessary to explain the timing of the whole cycle are included.

4-23. PROCESSING A TYPICAL INSTRUCTION

The following paragraphs apply the principles of the basic operation cycle to the processing of a typical instruction, the 60 instruction. In the discussion of the search-for-instruction step, the operation of the main-storage read circuits was described. The main-storage write circuits are described under this heading with the 60 instruction. A transfer of information from register A to a main-storage location is specified by this instruction.

During the search-for-instruction step, the 60 instruction is in main storage. During the staticize step the instruction word is stored in register C, and the two instruction digits, 6 and 0, are staticized. In this instruction, the contents of register A are transferred to a storage location. The address of that location is the operand in the instruction. The execution step of the instruction controls the writing of the word from register A into the specified storage location.

4-24. SEARCH FOR OPERAND

The 60 instruction is stored in the static register by the staticize step as 1001 0X00. The two lowest order bits of the p9 digit are 0's, which specify a search. This combination is detected by the (AM-LM-XM)SC and SC gate lines of the instruction decoder and generates FS1A and FS3A. (The search steps of the 60, 50, and 65 instructions require the same function signals; therefore, all are on a common instruction-decoder gate line.)

Function signal 1A generates FS58+, FS74, FS63, and FS1. Function signal 58+ blocks readout from register A onto the S lines. Function signal 74 sets the complement flip-flop (drawing 1-15B) to produce a CP signal. Function signal 63, in conjunction with the CT signal, causes the m address from register C to be read out onto the S lines. Function signal 1 alerts the input gates of the M buffer to enable the timing-band signals from the drum to enter the M lines.

The 1001 0X00 static-register output combination also generates FS3A (AM-XM-LM SC gate) which generates FS3 to alert the STR FF1 step gate. At t5B of the search-for-operand step, the output of the TS flip-flop is sampled by gate 8 of FF1. A low TS signal and FS3 make gate 8 permissive and step FF1 from a 0 output to a 1 output. The new STR output combination of 1001 0X01 (61) is present at the instruction decoder at t6B.

At t6B, FS3 sets the main-storage write flip-flop (drawing 1-21A) at gate 5B to prepare the write circuits for the execution step of the instruction.

When gate 8 of STR FF1 is permissive at t6A, a high RCT2 signal is generated which goes to the input gates of the CT flip-flop and to buffer 104 of the write circuits. The high RCT2 signal restores the CT flip-flop, generating the low $\overline{\mathrm{CT}}$ signal. The $\overline{\mathrm{CT}}$ signal controls readout of the c address from register C and generates the write pedestal. The new STR combination causes the function signals for search to end at t7A. The search-for-operand step has located the operand, which is the storage location designated by the m address.

4-25. EXECUTION STEP OF 60 INSTRUCTION

In the execution step of the 60 instruction (figure 4-8), the contents of register A are recorded in the storage address located by the search-for-operand step. During the execution step, certain function signals are generated by the new combination in the static register. These signals control transfer of the contents of register A onto the M lines and into the main-storage write circuits, the preparation of the write circuits to record on the drum, and eventually the ending of the instruction.

At t7A of the search step the AM gate line of the instruction decoder generates FS11A, which in turn, generates FS11 and FS21. These signals control writing the contents of register A onto the drum.

Function signal 11 alerts input gates 7A, 7B, 7C, and 7D of the M buffers (drawing 1-6B). These gates are permissive to the four bits of the p0 digit (A1M, A2M, A3M, and A4M) from register A at t7B, and the contents of register A enter the M lines. Gate 6 of the M buffers, also alerted by FS11, is permissive at t7B to a low A— signal. This condition indicates that the sign of the contents of register A, which has been stored in the rA-sign flip-flop, is minus. If the sign of register A is plus, low A— is absent from gate 6. The sign position of the word (t7B) on the A1M lines contains 0, indicating a plus sign. The word and its sign are sent from register A onto the M lines and into the write circuits.

At t4B of the execute step, FS21 operates static-register gate 19 to generate EP and reset-TS signals. Because the 60, 50 and 65 instructions involve writing on the drum and therefore require less time for execution than most instructions, the EP signal is generated early. At 7B the function signals for the instruction no longer are generated. The TS flip-flop is restored so that its set output cannot set the main-storage-read flip-flop (drawing 1-21B). Normally, at t8B, the read flip-flop is set at gate 42A by FS1, TS, and OF. All of these signals are present at t8B because the function signals for search are available at t7B. To keep the read flip-flop restored until reading is required, the reset-TS signal restores the TS flip-flop, keeping the TS signal high to block gate 42A of the read flip-flop. The 0 combination in the static register initiates a search for the next instruction.

4-26. MEMORY-WRITE OPERATION

During a write operation, information and check bits are transferred by the M lines to the phase-modulation coder where each input signal is coded into a phase-modulated pulse representation for recording. The resulting pulse is amplified in the write amplifier and drives the write heads.

At t7A, FS11A electronically disconnects the read amplifiers from the write circuits by generating the write pedestal. Control signal RCT2, present for one pulse time, has generated the write pedestal from t6A until t7A. For writing onto the fast-access bands, the IR output of the write-pedestal generator (WPP) disconnects the read amplifiers of the fast-access heads.

The bits of each digit to be written are sent in parallel to the check-bit computer circuit and to the write-input circuits. (The check-bit computer circuit is described under heading 4-42.) Each of the four bits of the digit to be written is delayed one pulse time by two complementing amplifiers before going to the input gates of the phase-modulation coder, so that the check bit and the information bits can be written simultaneously. Computing the check bit requires a full pulse time. At t9B the information bits and a check bit are present at the input gates to the phase-modulation coder. The write flip-flop output alerts the input gates at t8B and for a full word time thereafter. (Operation of the phase-modulation coder is explained under heading 3-100.) The bits are converted into a phase-modulated waveshape which goes directly to the read-write heads and is written onto the drum at the location specified by the m address.

4-27. ARITHMETIC OPERATIONS

4-28. ADDITION AND SUBTRACTION

4-29. ADD (70) INSTRUCTION. The add instruction adds algebraically the quantity at the storage location designated by the m address to the contents of register A, stores the result in register A, computes the sign of the sum, and stores the sign in the rA-sign flip-flop. The quantity in storage is the addend; the quantity in register A is the augend.

The add instruction requires two execution steps, A1 and A2.

The following preparations for addition will have taken place before the A1 step:

(1) Search-for-instruction step completed;

(2) Add (70) instruction staticized;

(3) Staticized instruction has initiated a search for the operand (m address), which in this instruction is the addend; and

(4) Augend stored in register A by a previous instruction, and its sign stored in the rA-sign flip-flop.

4-30. A1 Step (71). In the first step of addition, A1 (figure 4-9a), the signs of the augend and addend are compared at the input gates of the complement (CP) flip-flop. The gates determine from the two signs whether the augend is to be complemented (CP) or not complemented ($\overline{CP}$) before addition. This information controls the complementing circuit of the quinary adder. The sign of the augend is compared with the CP output of the complement flip-flop at the input gates of the rA-sign flip-flop to determine the sign of the sum. Once determined, the sign of the sum is stored in the rA-sign flip-flop. The augend in register A is sent on the S lines to the comparator and the quinary adder. The addend from the location designated by the m address is sent on the M lines to the comparator and the quinary adder. The adders add the two quantities, digit by digit, and transfer the sum on the O lines into register A.

The staticized digits of the A1 step (71) generate FS4, FS50, FS55+, FS64, and FS75 at t0B to execute the step. Function signal 4 alerts the input gates of the complement flip-flop. Gates 8 and 9 of this flip-flop sample the signs of the two quantities. If the signs are unlike, one of the gates sets the complement flip-flop to generate CP, which causes the augend digits to be complemented. If the signs are alike, gates 8 and 9 are blocked, the flip-flop remains restored to generate $\overline{CP}$, and the augend is not complemented.

Function signal 55+ blocks the recirculation gates of register A, and the digits of the augend go to the S buffers and the S lines. Function signal 50 alerts the quinary- and binary-adder gates (drawing 1-14B) at t0B.

The quinary bits of the p1 digit of the addend (M1, M2, and M3) go to the input gates of the decimal-carry adder

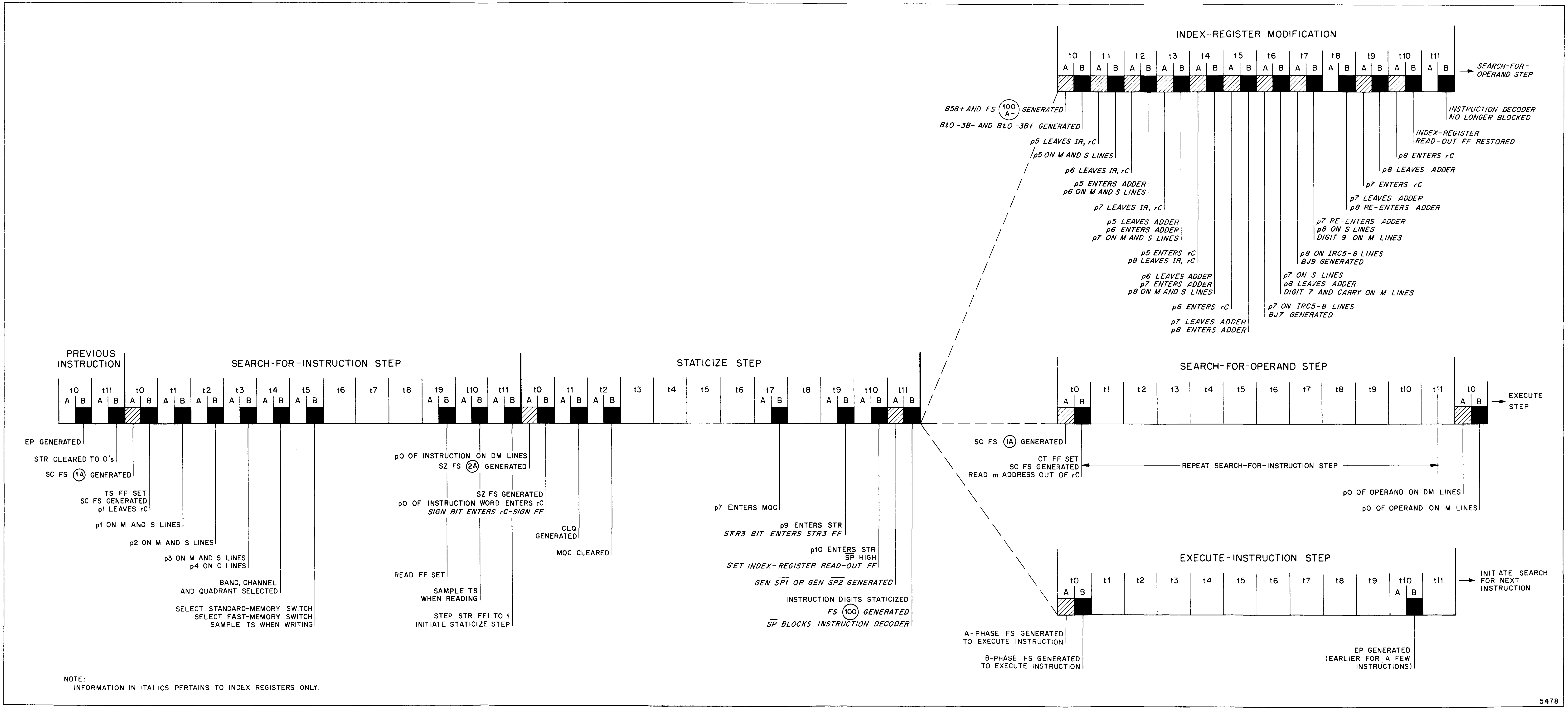


Figure 4-7. Timing of Basic Operation Cycle

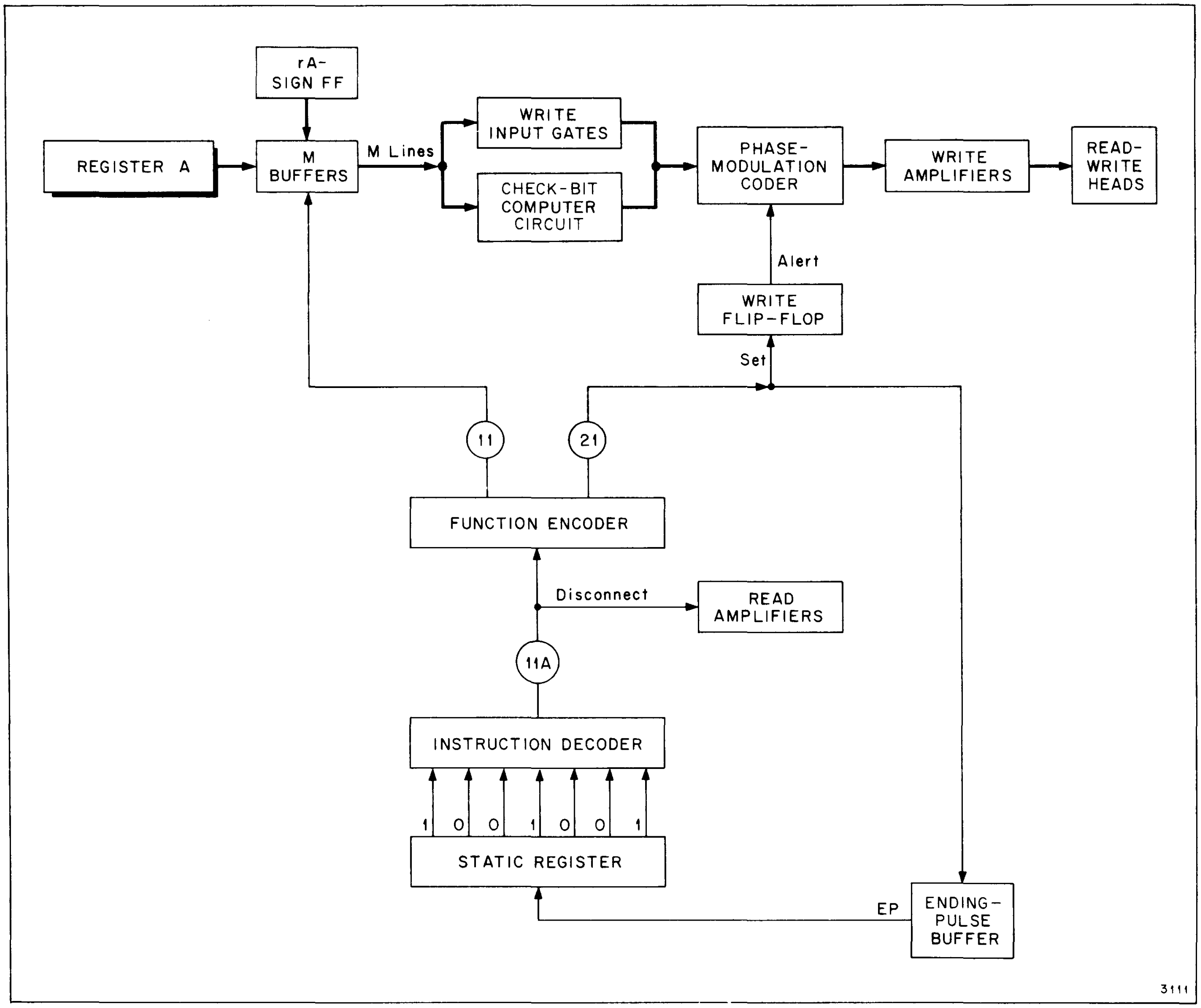


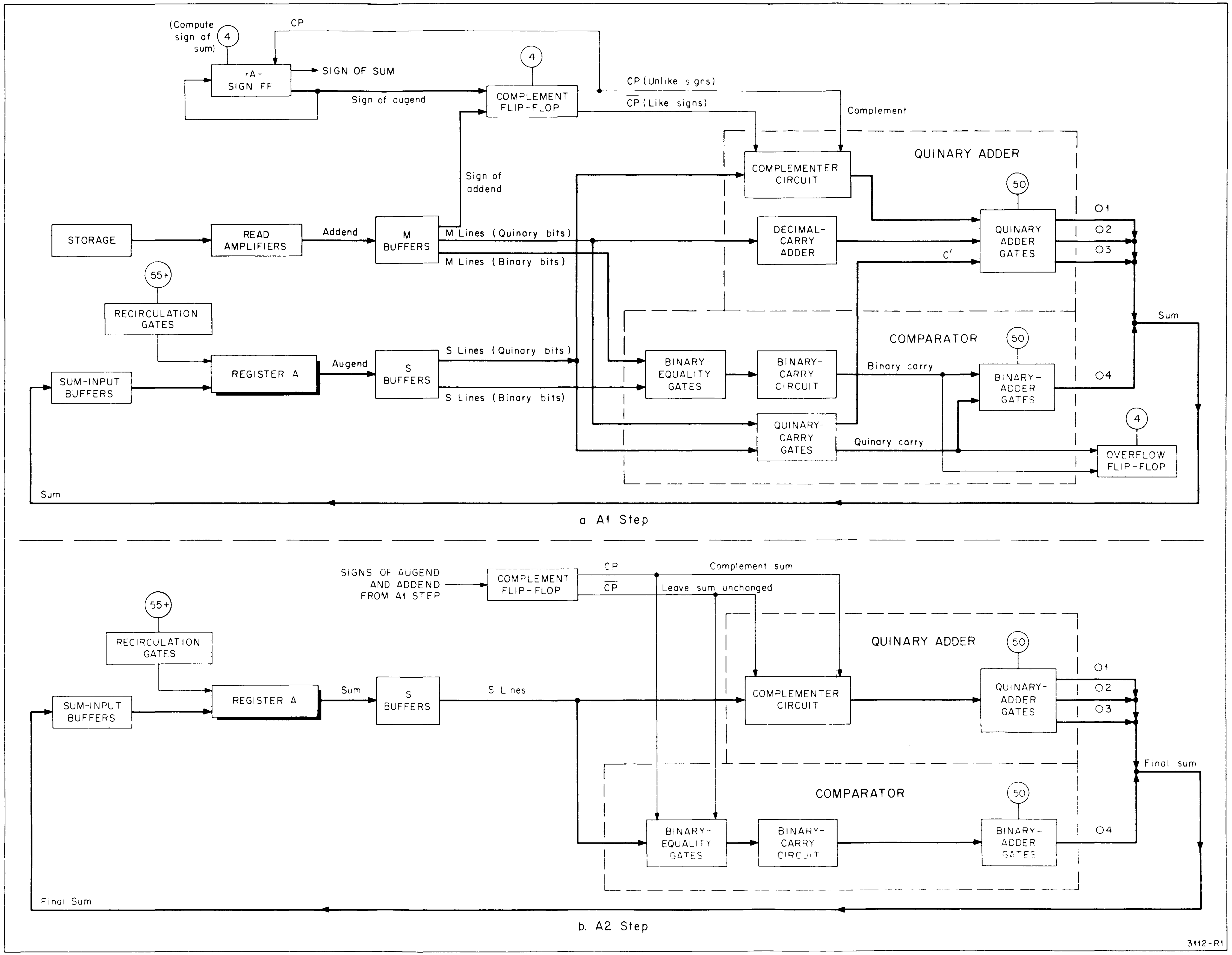
Figure 4-8. Execution Step of 60 Instruction

(drawing 1-13B) and the quinary-carry gates (drawing 1-14B). The input gates of the decimal-carry adder sample the A and C signals from the binary and quinary circuits to determine whether a carry bit is to be added to the two digits being added. Because the pl digits are the first to be added, there can be no carry, and the $\overline{A}$ or $\overline{C}$ signal, or both, will be low. The outputs of the decimal-carry adder, which represent the quinary bits of each digit of the addend, go to the quinary adder.

The quinary bits of the pl digit of the augend (S1, S2, and S3) go to the input gates of the complementer circuit (drawing 1-13B). If the CP (complement) signal alerts the circuit at gate 3 or 6, the complement of the input bits is produced. If the $\overline{CP}$ (no complement) signal alerts the circuit at gate 2 or 5, the quinary S bits pass through it unchanged.

The quinary S and M bits also are sent in parallel to the quinary-carry circuit (drawing 1-14B). Any combination of quinary bits which indicates the need for quinary carry generates C and C′ signals from the quinary-carry circuit at buffers 71 and 73. The gates are blocked by combinations which require no quinary carry, generating the low $\overline{C}$ signal at buffer 72. The low C signal, if generated, is used in conjunction with the A signal to signify a decimal carry to the addition of the next two digits. The low C′ signal signifies the presence of a quinary carry and causes a carry bit to be added in the addition of the binary bits in the binary adder.

The quinary outputs of the decimal-carry adder and those of the complementer circuit are added in quinary-adder gates 29 through 46 (drawing 1-13B). One (or two) of the adder gates is permissive to the input combinations, producing a high O signal. A high O signal indicates a 1 bit, a low O signal indicates a 0 bit.



a A1 Step

b. A2 Step

Figure 4-9. Add (70) Instruction

The binary bits of the two quantities are added in the binary-adder gates (drawing 1-14B) at the same time as the quinary bits. Initially, the M4 and S4 binary bits are compared in the binary-equality gates. The gates, alerted by $\overline{CP}$, sample the two binary bits during a normal addition. If either bit is a 1, a signal indicating the 1 bit goes to the binary-carry circuit to generate binary-carry signal A, which will be used with the C signal, if present, in the addition of the next two digits. The carry circuit also sends the A′ signal to gate 6 of the binary adder. If the $\overline{C}$ signal is low, gate 6 is permissive to the A′ signal and the O42 output is high, indicating a 1 bit in the MSB of the sum digit. No decimal carry will be generated if the $\overline{C}$ signal is low.

If the M4 and S4 bits both have a value of 1, gate 30 of the initial-force-decimal-carry circuit is permissive, and the A and C signals are generated to force an indication of a decimal carry to the next addition.

If the M4 and S4 bits both have a value of 0, the binary-equality gates are blocked, generating low signals $\overline{A}$ and $\overline{A}'$ which indicate no binary carry. The low $\overline{A}'$ signal goes to gate 7 of the binary adder which is permissive only if quinary carry (low C′ signal) is present. If the gate is permissive, the O41 output is high, indicating that a 1 is in the MSB position of the sum digit. If quinary carry is absent, the C′ signal is high, blocking gate 7 and producing a low O41 bit. A low O41 output signal indicates a 0 in the MSB of the sum digit. The O signal outputs of the quinary adder and the binary adder are sent to the sum-input buffers of register A (drawing 1-8B). The sum digits are stored in register A during the A1 step.

If the sum of two quantities with unlike signs is 0, the sign of that sum must be forced to a plus. The TS flip-flop (drawing 1-14B) is used as a control circuit to indicate this condition to the rA-sign flip-flop. Function signal 75 alerts restore gates 16 and 17 of the TS flip-flop during the A1 step. If the signs of the two quantities being added are unlike, the CP flip-flop is set to generate CP. The CP, A, and C signals are present during addition when the signs are unequal. These signals alert the quinary-equality gates, which compare the digits being added in the adders for equality. The CP signal alerts the binary-equality gates, which also compare for equality. If the first nine digits of the quantities being added are equal, the TS flip-flop remains set to TS. The TS signal goes to the rA-sign flip-flop (drawing 1-15B) to force the sign of the sum to a plus if the sum of the quantities is 0.

At t11B, FS4 alerts the input gates of the rA-sign and overflow (OF) flip-flops. Gates 3, 6, and 7 of the rA-sign flip-flop compute the sign of the sum and store it in the flip-flop. During addition, the (OF) flip-flop is set at gate 29 when the input signals indicate that the sum has exceeded the capacity of register A. When this possibility is expected, the programmer provides a routine starting in the c + 1 address, to be used only if overflow occurs. The set output of the OF flip-flop (OF) sets the overflow-delay flip-flop to OF2+ during the next search-for-instruction step. The OF2+ signal keeps the TS flip-flop set at buffer 82 to delay for one word time the reading of the selected instruction-word address. This causes the c + 1 address to be read instead of the c address.

Function signal 64 alerts gate 26 of the static register (drawing 1-2B) at t10B of the A1 step. If the EP signal is low, FF2 is set. The new setting of STR FF1 and STR FF2 (11) causes the instruction decoder and function encoder to generate function signals for the A2 step. These function signals are 50, 55+, and 67.

4-31. A2 Step (73). If the sum determined during the A1 step is the complement of the true sum, it must be complemented to its true form during the A2 step (figure 4-9b). The sum determined during the A1 step and stored in register A reads out onto the S lines. Register A is cleared by FS55+ so that the true sum, which will be determined during the A2 step, can enter it.

Function signal 50 alerts the gates of the binary adder (drawing 1-14B) and the quinary adder (drawing 1-13B). The sum on the S lines goes to the complementer circuit and the binary-equality gates for complementing if necessary. If complementing is unnecessary, the $\overline{CP}$ signal is present and the sum passes through the complementer circuit. It also passes through the quinary and binary adders, as in the A1 step, and returns to register A. No addition takes place in the adders because there is no quantity on the M lines to be added. (In subtraction, however, a 1 is automatically added to the quantity from register A in the decimal-carry adder.)

Function signal 67 operates gate 17 of the ending-pulse-buffer circuit (drawing 1-2B) at t9B. At t10B the EP signal is generated to clear the static register of the add instruction and initiate a search for the next instruction.

4-32. SUBTRACT (75) INSTRUCTION. The execution of add and subtract instructions is identical. However, they differ in control functions. The subtract instruction subtracts algebraically the contents of the storage location designated by the m address from the contents of register A, stores the difference in register A, computes the sign of the difference and stores it in the rA-sign flip-flop. The quantity in storage is the subtrahend; the quantity in register A is the minuend. The subtract instruction requires two execution steps, S1 and S2.

Although the instruction code for subtract is 75, the staticized digits generate the same function signals as the add instruction. The STR FF4 output for subtract is an STR4 signal (drawing 1-2B). As a result, gates 10 and 11 of the complement flip-flop (drawing 1-5B) operate during subtraction to determine whether the subtrahend is to be complemented.

When quantities with unlike signs are subtracted, the process is the same as when quantities with like signs are added. The sign of the subtrahend is changed, and the two quantities are added. These conditions restore the complement flip-flop because no complementing is required.

A special circuit in the function encoder operates during addition and subtraction to control the complement flip-flop and complementing in the A2 or S2 step.

4-33. SAMPLE PROBLEMS. The many similarities and the few dissimilarities of addition and subtraction may be illustrated by two sample problems. In the examples, two digits are first added and then subtracted.

4-34. **Addition.** In this sample addition problem, column A contains the digits, B contains their biquinary equivalents, C contains the bits of the digits as signals on the M and S lines, and D represents the outputs of the decimal-carry adder and complementer circuits. The M4 and S4 bits are not included because they are added within the binary-adder gates of the comparator. The addend is the contents of m, and the augend the contents of register A.

Example of Addition

A	B	C	D
+2	(m) = 0010 =	M4 M3 M2 M1 =	M3U M2U M1U
+1	(rA) = 0001 =	S4 S3 S2 S1 =	S3C S2C S1C

	Low	Low	High	High	
Sum after A1 and A2:	O4	O3	O2	O1	= 0011 or +3

Like signs in the add instruction restore the complement flip-flop to $\overline{\text{CP}}$. The quinary bits of the augend, $\overline{\text{M3}}$, M2, and $\overline{\text{M1}}$, go to the decimal-carry adder, which does not change their value. The outputs of the circuit, $\overline{\text{M3U}}$, M2U, and $\overline{\text{M1U}}$, have the same quinary value (010) as the inputs. The quinary bits of the addend, $\overline{\text{S3}}$, $\overline{\text{S2}}$, and S1, go to the complementer circuit, which does not complement because the $\overline{\text{CP}}$ signal is low. The outputs of the complementer circuit, $\overline{\text{S3C}}$, $\overline{\text{S2C}}$, and S1C, have the same value (001) as the inputs. The binary bits of the two digits, $\overline{\text{M4}}$ and $\overline{\text{S4}}$, go to the binary-carry circuit. These two bits and $\overline{\text{CP}}$ generate a low O4 output from the binary adder. A low output signal on the O lines is a 0 bit, which goes to subregister A4 of register A (drawing 1-8B).·The outputs of the decimal-carry adder and complementer circuits go to the quinary adder (drawing 1-13B). Only gate 34 of the adder is permissive to the input combination. Gate 34 produces a high O12 output which goes to the A1 and A2 subregisters of register A. Hence the two lowest order bits of the sum are 1 bits. Because only gate 34 is permissive, the other gates are blocked. As a result, inputs to the A3 subregister from the quinary adder are low O signals, indicating a 0 bit. The sum determined during the A1 step, therefore, is 0011, or 3. In the A2 step this sum recirculates unchanged through the complementer circuit because the complement flip-flop remains restored to $\overline{\text{CP}}$. The rA-sign flip-flop determines from the signs of the two quantities that the sign of the sum is a plus.

4-35. **Subtraction.** In the first step of subtraction, S1, the input M and S bits which go to the decimal-carry adder (shown in the example in column C) and the complementer are the same as in addition. In subtraction, however, the decimal-carry adder adds a 1 to the subtrahend and the complementer produces the 9's complement of the minuend if the CP signal is present. (Complementing is described under heading 4-47.)

In the example, the minuend is the contents of register A, and the subtrahend the contents of m.

At t0B of the first step of subtraction (S1), the plus signs of the two digits and the STR4 signal set the complement flip-flop to generate CP at gate 11 (drawing 1-5B). At t1A, high signal CP4 is generated for one pulse time. Signal CP4 goes to the initial-force-decimal-carry circuit (drawing 1-14B) to force generation of the decimal-carry signals, A and C. The A and C signals represent the 1 bit which is to be added to the subtrahend in the decimal-carry adder. Because signals A and C were generated by the CP4 signal, they last for only one pulse time. The quinary M inputs to the decimal-carry adder represent a decimal 1; the outputs (D in the example) represent a 2.

Example of Subtraction

S1 step

A		B	C		D
+2	(rA)	= 0010 =	S4 S3 S2 S1	→ complement →	S3C S2C S1C
+1	(m)	= 0001 =	M4 M3 M2 M1	→ add 1 →	M3U M2U M1U

	High	High	Low	Low	
Sum after S1:	O4	O3	O2	O1	= 1100 or 9

S2 step

E		F		G
1100	(from rA) =	S4 S3 S2 S1	→ complement →	S3C S2C S1C
0000	(on M lines) =	M4 M3 M2 M1	→ add 1 →	M3U M2U M1U

	Low	Low	Low	High	
Sum after S2:	O4	O3	O2	O1	= 0001 or 1

Complementing of the S input combination to the complementer circuit is not evident from the output combination, which retains the same quinary value. The result of complementing, however is evident at the outputs of the adders. Only gate 46 of the adder is permissive to the outputs of the decimal-carry adder and complementer circuits; it produces a high O3 output and low O2 and O1 outputs. The $\overline{M4}$ and $\overline{S4}$ bits, with CP, send a low signal to binary-adder gate 6. None of the quinary-carry gates alerted by CP is permissive to the M and S inputs. As a result, a low $\overline{C}$ signal alerts gate 6 to produce a high O4 output, signifying a binary 1 bit. The O input bits to register A at the end of the S1 step therefore have a value of 1100 or 9. The decimal 9 is the 10's complement of the true answer.

Because addition of the two combinations in the S1 step produced no decimal carry, the $\overline{A}$ and $\overline{C}$ signals from the comparator (drawing 1-14B) are low at t11B of the S1 step. Gates 62 and 63 of the function encoder (drawing 1-4B), alerted by the CP signal to the low $\overline{A}$ and $\overline{C}$ signals, are permissive at t11B of the S1 step. When the gates are permissive, low FS74 is generated which keeps the complement flip-flop set to generate CP and generates the CP5 signal for one pulse time. The CP5 signal forces decimal-carry signals A and C low from t1B until t2B of the S2 step.

In the S2 step, the 1100 combination (E in the example) returns to the S lines from register A as S4, S3, $\overline{S2}$, $\overline{S1}$ (F in the example). The three lowest order S bits go to the complementer, which produces a $\overline{S3C}$, $\overline{S2C}$, $\overline{S1C}$ output combination (G in the example).

During the S2 step there is no information on the M lines, and the barred M signals all are low (F in the example). The A and C signals, which are present for one pulse time, cause a decimal 1 to be added to the 0 value of the M bits. The resulting output of the decimal-carry adder (drawing 1-13B) is $\overline{M3U}$, $\overline{M2U}$, M1U (G).

The outputs of the decimal-carry adder and complementer go to the quinary-adder gates. Only gate 32 is permissive to the input combination to produce a high O1 output. while the O2 and O3 outputs of the quinary adder and the O4 output of the binary adder are low. The output combination of the quinary and binary adders, 0001 or 1, returns to the sum-input buffers of register A. The plus sign of the result is computed by the rA-sign flip-flop.

4-36. MULTIPLICATION

Multiplication is accomplished by iterative additions and shifts of the multiplicand. The multiplicand is added to itself the number of times specified by each digit of the multiplier. The multiplication process is divided into two steps, MPY1 and MPY2. The Processor can multiply a ten-digit multiplicand by a ten-digit multiplier. In such a case, the 20-digit product will be stored in registers A and X. The least significant part of the product is stored in register X; the most significant part of the product is stored in register A.

Before multiplication, the multiplicand is placed in register L by a programmed transfer instruction. In the first step of multiplication (MPY1), the multiplier is transferred from its storage location to register X. A code combination, the multiplier sentinel, is placed in the LSD position of register A. Multiplication ends when the sentinel is detected.

During the second step, MPY2, the multiplicand in register L is added to itself in the adder the number of times indicated by the LSD of the multiplier. The LSD of the multiplier is stored in the multiplier/quotient counter (MQC) which counts the number of additions. The sum of these additions is returned to register A. Next, the contents of registers A and X are shifted right one digit position, placing the LSD of register A in the MSD position of register X. The LSD in register A before each shift becomes the LSD of the product. This process of adding and shifting continues until the LSD of the product is the LSD of register X.

The multiplication sentinel also is shifted during shift operations. Initially it is shifted from the LSD position in register A to the MSD position in register X. Every ensuing shift operation shifts the sentinel one more place to the right until it reaches the LSD position of register X. Then it is shifted into the MQC where it ends the multiplication process.

The MPY2 step consists of two phases, IER (shift) and $\overline{IER}$ (add).

During each IER phase, the contents of registers A and X are shifted one place to the right; the sentinel is shifted one place, and the LSD of register X, which contains the multiplier, is shifted into the MQC to control the number of additions.

During the $\overline{IER}$ phase, the multiplicand is added to the sum of additions from the previous $\overline{IER}$ phase the number of times specified by the multiplier digit in the MQC. The new sum is returned to register A for the next $\overline{IER}$ phase.

4-37. GENERAL DESCRIPTION. The multiplication process used in the Processor is illustrated by the sample problem in figure 4-10. In the problem, the number 1234 is multiplied by 4321. Although the Processor can multiply two ten-digit numbers, the sample problem is confined to two four-digit numbers. Figure 4-10a shows the process used by the computer; figure 4-10b shows the process used in the Processor circuits. The circled digits and the final four digits in figure 4-10a are the product digits. The first computer product digit is 4, which is the LSD of the final product.

During the MPY1 step, the multiplication sentinel (S) is placed in the LSD position of register A, and the multiplier is placed in register X. During the shift phase of the MPY2 step, the multiplication sentinel is shifted from register A to the MSD position of register X. The multiplier in register X also is shifted right, causing the LSD (1) to be transferred to the MQC. During the add phase of MPY2, the multiplicand from register L is added to

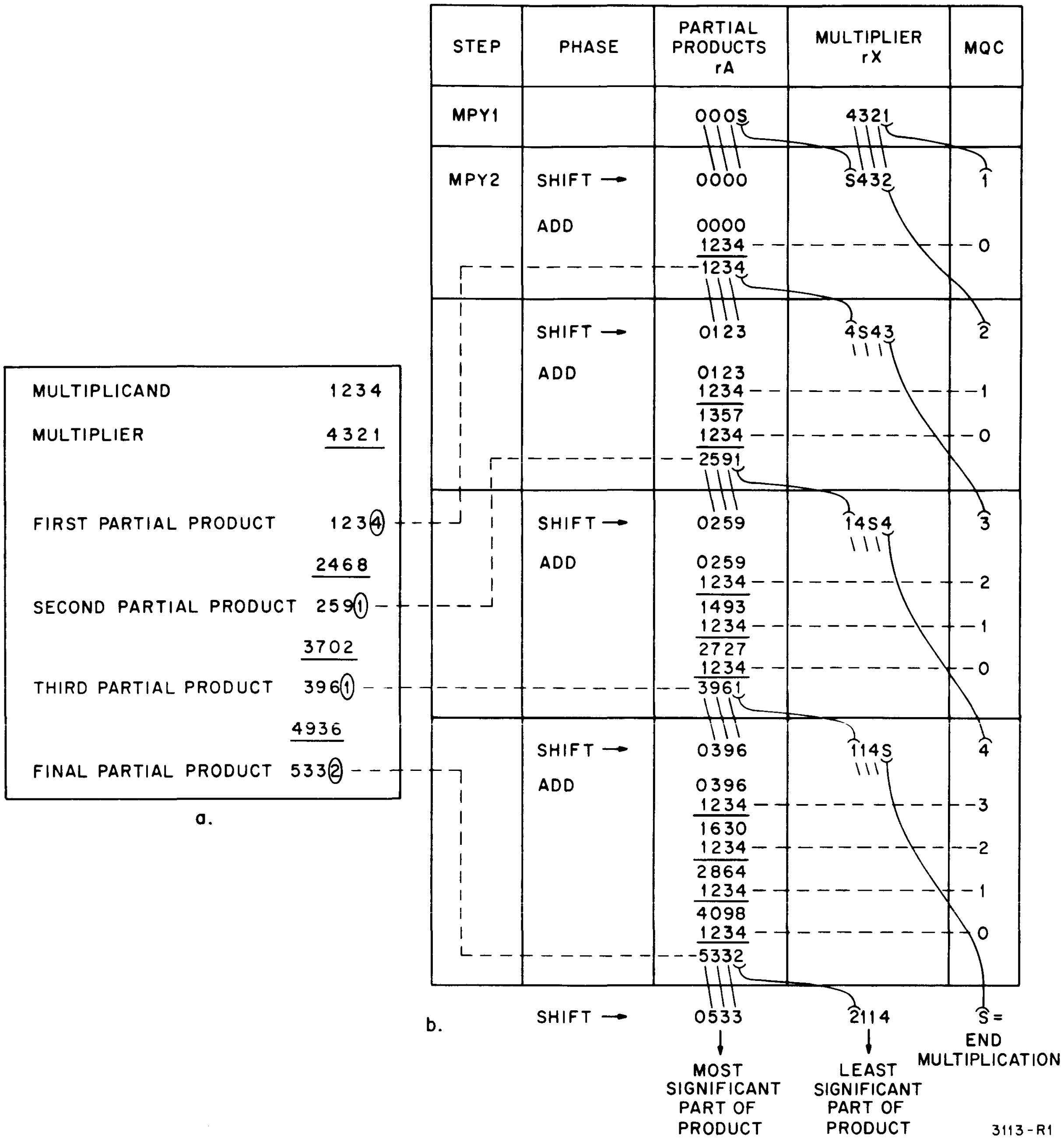


Figure 4-10. Multiplication Process

the contents of register A the number of times specified by the digit stored in the MQC. The MQC digit is 1; therefore, the multiplicand 1234 is added to the contents of register A, which at this point is 0000. The sum of the addition, 1234, is returned to register A. The MQC counts down to 0 after the necessary number of additions has occurred, and a 0 digit in the MQC initiates the next shift phase.

During the next shift phase, the contents of register A (1234) are shifted to the right, causing the LSD of the first partial product, 4, to be shifted into register X. At the same time that registers A and X are shifted, the sentinel advances to the right one more place. The LSD of the multiplier in register X, 2, is shifted into the MQC to control the number of additions in the next add phase. The multiplicand (1234) is added to the shifted contents of register A (0123) twice as specified by the MQC. The two additions produce the second partial product, 2591. The second partial product also is shifted; the LSD, 1, becomes the second quotient digit when placed in the MSD position of register X. The product digits and sentinel in register X shift again to the right, leaving the MSD vacancy into which the partial product digit, 1, is placed. The LSD of register X, 3, is transferred to the MQC. The MQC therefore permits three additions of the multiplicand, after which a 0 reading starts the next shift phase. This process of alternating shift and add phases continues until the sentinel reaches the LSD position of register X and is shifted into the MQC. The sentinel combination in the MQC ends multiplication with the most significant part of the product in register A and the least significant part in register X.

4-38. INITIAL CONDITIONS. The following conditions must have taken place before the MPY1 step can be initiated:

The multiplicand is in register L and its sign in the rL-sign flip-flop;

The instruction word is located in the search-for-instruction step;

The multiply (85) instruction is staticized in the static register;

The multiplier in the storage location designated by m is read from the drum in the search-for-operand step; and

STR FF1 is jammed to 1. The combination in the static register is 1011 1X01, which generates function signals at t0A and t0B to carry out the MPY1 step.

4-39. MPY1 STEP. The sign of the final product in multiplication is determined during the MPY1 step (figure 4-11a) by the rA-sign flip-flop, which compares the sign of the multiplier and the multiplicand. The sign of the multiplicand was stored in the rL-sign flip-flop (drawing 1-15B) during the transfer instruction. The sign of the multiplier enters the rX-sign flip-flop at t0B at gates 34 and 35, which are alerted by FS78. At t0A, the rA-sign flip-flop is set to A– by FS13A to prepare for comparison of the signs of the multiplier and multiplicand. At t11B of the MPY1 step, FS86 alerts gates 1 and 2 of the rA-sign flip-flop. If the signs of the multiplier and multiplicand are alike (X+ and L+, or X– and L–), the flip-flop is restored to A+. If the signs are unlike, gates 1 and 2 are blocked and the flip-flop remains set to A–.

At t0B, FS62 operates gate 19 of the clear-MQC circuit (drawing 1-16B). At t1A the CLQ signal clears the MQC flip-flops, which receive the LSD of the multiplier in the MPY2 step. At t0B, FS56+ blocks the recirculation gates of register X, and FS76 alerts the input gates to the multiplier which enters on the M lines.

Register A is cleared during the MPY1 step so that it can receive the partial products during the MPY2 step. At t0B, FS55+ blocks the recirculation gates of register A (drawing 1-8B). At t1B, FS13 operates the multiplier-sentinel gate of register A to jam 1's into subregisters A1 and A3. As a result, the LSD of register A is the multiplier-sentinel combination, 0101. Register A now contains the sentinel and nine 0 digits. Function signal 58+ blocks the normal output gates of register A so that the sentinel remains within the register.

At t10B of the MPY1 step, FS64 alerts STR step gate 26 (drawing 1-2B) which jams STR FF2 to a 1 output to initiate the MPY2 step. Function signals for the MPY2 step are generated at t0A and t0B of the MPY2 step.

At t11B, gate 11 of the IER flip-flop (drawing 1-17B) samples the STR output signals, which indicate that the MPY2 step has been staticized, and the $\overline{Q}$ (0) outputs of the MQC flip-flop, which indicate that the circuit has been cleared. If these two conditions are met, the IER flip-flop is set at t0B to initiate the IER phase of the MPY2 step. The flip-flop generates low control signals IER at t0B and IER-0R at t1B, high control signals IERA1+ from t0A to t0B only, IERA2+ at t1A, and IER+ at t0B. These signals control the IER phase of the MPY2 step.

4-40. MPY2 STEP. The MPY2 step generates FS55+ and FS66. These signals are present during both the IER and $\overline{\text{IER}}$ phases of MPY2. Other function signals, however, are generated during each of the two phases by the outputs of the IER flip-flop.

4-41. IER Phase. At the beginning of the IER (shift) phase, control signal IERA1+ generates FS59 for one pulse time from t0B to t1B. Beginning at t1B, control signal IERA+ is high, and it generates FS59 and FS56+.

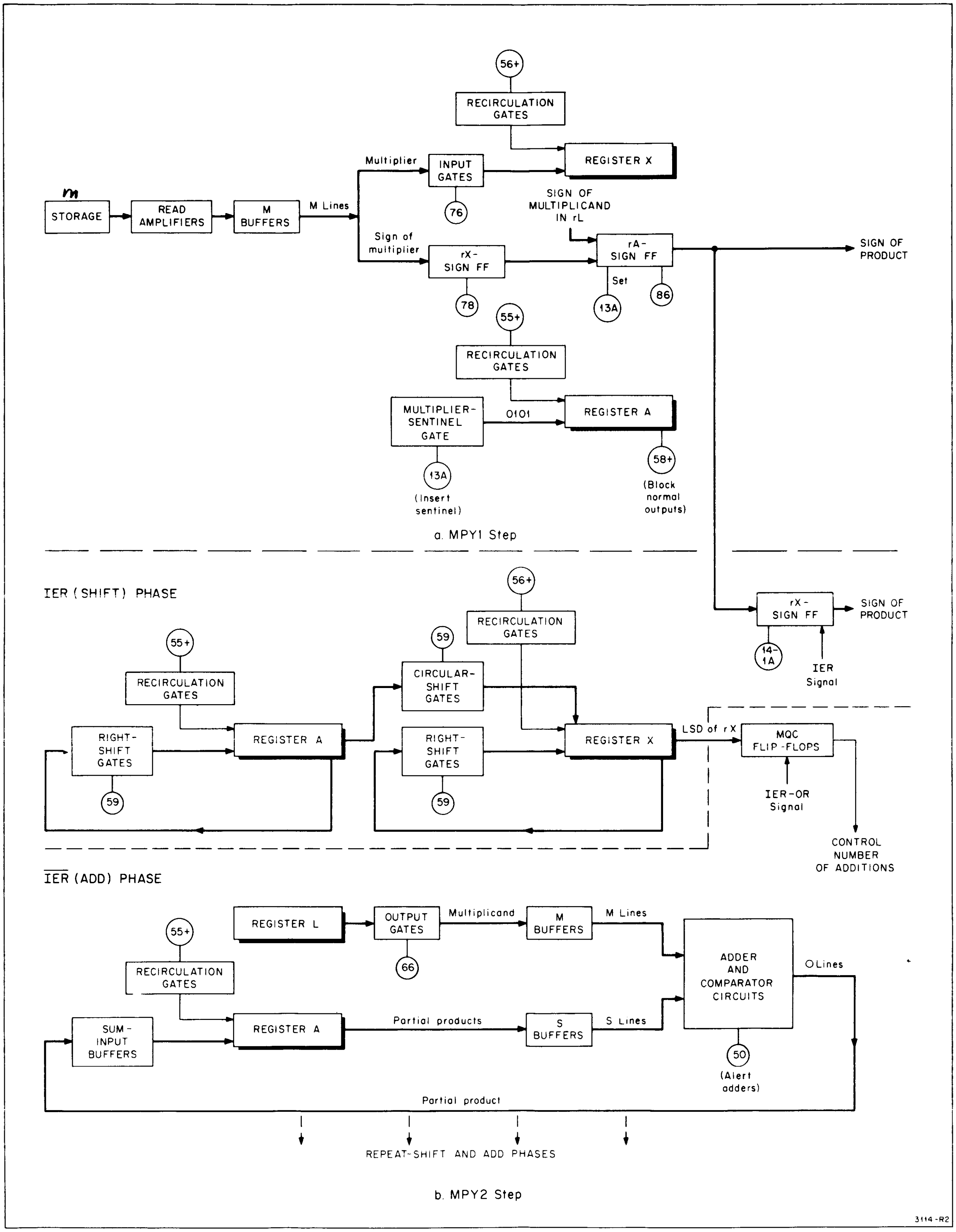
At t0A of the IER phase, FS14-1A jams the rX-sign flip-flop to X+. At t0B of the IER phase, the low IER control signal alerts the rX-sign flip-flop at gate 33 (drawing 1-15B) which samples the sign of register A from the rA-sign flip-flop. If the sign of register A is minus, the rX-sign flip-flop is restored to X–; if it is plus, the flip-flop remains set to X+. Thus the signs of both parts of the final product will be identical.

Function signal 59 alerts the right-shift gates of registers A and X and the circular-shift gates of register X so that both registers can be shifted right. Function signal 56+ blocks the recirculation gates of register X; FS55+ blocks the recirculation gates of register A. During the first IER phase, the right-shift operation causes the multiplication sentinel, in the LSD position of register A, to be shifted to the MSD position of register X. At the same time, the LSD of the multiplier in register X is shifted into the MQC flip-flops. Control signal IER-0R alerts the input gates of the four MQC flip-flops to the outputs of the register-X subregisters.

Function signal 66 alerts the output gates of register L, which stores the multiplicand during multiplication. The multiplicand in register L is transferred to the M lines and into the added circuits in preparation for the add phase ($\overline{\text{IER}}$). The partial product stored in register A after each shift phase is on the S lines and is also an input to the adder circuits.

During every IER phase, the IER control signal alerts gate 18 of the ending-pulse circuit (drawing 1-2B) to sample for the 0101 sentinel in the MQC. When the sentinel is present in the MQC flip-flop, the gate is permissive and the EP signal is generated to end multiplication.

4-42. $\overline{\text{IER}}$ Phase. During this phase, the contents of register A are added in the adder to the multiplicand from



a. MPY1 Step

b. MPY2 Step

Figure 4-11. Multiply (85) Instruction

register L. The $\overline{\text{IER}}$ phase is initiated at t11B of the IER phase by the t11B+ timing signal at gate 40 of the IER flip-flop (drawing 1-17B). The signal restores the flip-flop at t0A, and the polarities of the signals generated by the flip-flop during the IER phase are reversed for the $\overline{\text{IER}}$ phase. The IER+ output, which is now low, goes to the instruction decoder. If the staticized MPY2 signals are present and the IER+ signal is low, FS14-2A is generated, which in turn generates FS50 and FS61. Function signal 50 alerts the binary and quinary adders; FS61 alerts the countdown gates of the multiplier-quotient counter.

After each shift phase, the multiplicand is added to the partial product in the binary and quinary adders, just as in addition (heading 4-9). The sum of each addition is returned to register A. The MQC-countdown circuit counts down to 0 from the multiplier digit stored in it during the shift phase. When the reading in the MQC flip-flops is 0, the IER flip-flop is set to initiate the IER phase again.

After the last multiplier digit has been processed and the necessary additions have been performed, the sentinel is shifted into the MQC flip-flops. During the next shift phase, the IER signal alerts ending-pulse gate 18 (drawing 1-2B). If the sentinel (0101) is stored in the MQC, the Q1 and Q3 signals from the MQC flip-flops are low. Only the sentinel has 1 bits in these positions. The EP signal is generated and, at t0B, a search for the next instruction is initiated.

4-43. DIVISION

All of the arithmetic operations previously described are basically reiterative processes which may or may not use some form of 9's complement in the execute phase. Addition is a simple combination of two numbers with or without complement; subtraction is a similar combination: and multiplication is a controlled succession of adds. Division, though combining some aspects of all of the preceding arithmetic operations, is essentially a successive series of controlled subtractions.

The division process consists of the following arithmetic operations:

(1) Nine's-complement the dividend and multiply the result by 10;

(2) Repeatedly add the divisor to the dividend until a carry is generated;

(3) Subtract from 10 the number of times the divisor-dividend add is performed and complement the result (this number becomes the MSD of the quotient);

(4) Nine's-complement the number from which the carry was generated, multiply by 10, and repeatedly add the divisor to this number until a carry is generated;

(5) Again subtract the number of divisor-dividend adds from 10, but do not complement the result (this number is used directly as the second digit of the quotient); and

(6) Repeat operations 2 through 5 as many times as the problem requires.

To illustrate this procedure, the number 348 will be divided by 27. In the following example, the numbers in parentheses to the left correspond to the arithmetic operations just described:

```
(1)      348
         651
       x  10
        6510  (zeros in the LSD are ignored)
(2)       27  1 add
        9210
          27  2 add
      1 1910
(3)       10
         - 2
           8  = 1 (MSD of quotient)
(4)      191
         808  (Note that it is necessary to multiply
       x  10  by 10 to shift the relationship of divisor
        8080  to dividend.)
          27  1 add
        8350
          27  2 add
        8620
          27  3 add
        8890
          27  4 add
        9160
          27  5 add
        9430
          27  6 add
        9700
          27  7 add
        9970
          27  8 add
      1 0240  = remainder
(5)       10
         - 8
           2  = LSD of quotient
```

The answer thus is 12, with a remainder of 24. This remainder could be treated as the first number of arithmetic operation (1) and the entire procedure repeated until there is no longer a remainder or until the quotient reaches the desired number of places.

4-44. GENERAL DESCRIPTION. In the Processor, a division instruction is executed by performing the arithmetic sequence described under the preceding heading in three steps: D1, D2, and D3. During the D1 step, the numbers to be divided are assigned to registers, and the divide (55) instruction is staticized. During the D2 step, several internal sequences are performed that parallel arithmetic operations (1) through (6), as described previously. The D2 step is terminated by either a programmed or automatic sentinel, thus permitting a fixed or variable quotient length. During the D3 stage, the answer is, in effect, staticized, and the next instruction is initiated.

Table 4-3 contains a complete sample division problem, both with and without a programmed sentinel, as it is executed in the Processor. In the MQC column is listed the state of the MQC at each successive addition. In the Phase column is indicated whether the Processor is in a shift-complement (OR) or an add ($\overline{\text{OR}}$) phase. In the OF and SBW columns are shown the overflow and space-between-words states at each step, and at which point a carry occurs.

4-45. INITIAL CONDITIONS. The following preparations for division precede the D1 step:

(1) The divisor is placed in register L by a previous instruction;

(2) The programmed sentinel (0101) is inserted in register X if it is to be used;

(3) The search-for-instruction step is completed;

(4) The divide (55) instruction is staticized; and

(5) The dividend (operand) is located during the search-for-operand step.

4-46. D1 STEP. During the first execution step of division, D1, in which registers A and X are prepared for division, the rX- and rA-sign flip-flops are prepared to compute the sign of the quotient and the remainder, the MQC is cleared, the automatic sentinel is placed in register X, and, at the end of D1, the static register is stepped to D2. Figure 4-12 is a block diagram of the D1 step. The divisor has been stored in register L by a previous instruction, and a programmed sentinel (0101) may have been placed in register X by a previous instruction.

Function signal 55+ blocks the recirculation gates of register A (drawing 1-8B) to clear the register. Function signal 77 makes the input gates of register A permissive to the information on the MT and S lines. While the dividend on the MT lines is read into register A, FS82A holds the outputs of the S buffers low.

Function signal 15+ blocks the recirculation gates of subregisters X2 and X4 of register X. The automatic sentinel, generated by FS15 at division sentinel gate 9 (drawing 1-10B), enters the sign position of register X. If no programmed sentinel is stored, this sentinel will end division after ten OR cycles.

Function signal 62 makes gate 19 of the clear-MQC circuit (drawing 1-16B) permissive at t0B to generate the CLQ signal which clears the MQC.

The sign of the divisor has been stored in the rL-sign flip-flop by a previous instruction. At the beginning of D1, the rX-sign flip-flop is jammed to X+ at t0B by FS78 at gate 35 (drawing 1-15B). If the sign of the dividend from main storage is minus, gate 34 also becomes permissive, changing the sign to minus. If the sign is plus, the rX-sign flip-flop remains as it is. The signs of the divisor and dividend are sampled during the D3 step to determine the sign of the quotient.

At t11B of the D1 step, FS15 makes gate 9 permissive to set the OR flip-flop, initiating the first OR phase of D2 at t0B and generating the IER-OR, ORA+, and OR control signals. At t10B of D1, FS64 makes gate 26 (drawing 1-2B) permissive to set STR FF1 and STR FF2. The static-register flip-flop combination now is 1000 1X11, which causes D2 function signals to be generated.

4-47. D2 STEP. The D2 step consists of alternating shift and complement phases (OR) and add phases ($\overline{\text{OR}}$). The first phase of D2 always is the OR phase which is followed by the $\overline{\text{OR}}$ phase.

4-48. **OR Phase.** Figure 4-13 is a block diagram of the first OR phase initiated by FS15 to set the OR flip-flop at the end of D1. The high ORA+ control signal from the OR flip-flop sets the complement flip-flop (drawing 1-15B) producing a low CP signal. Signal ORA+ also generates a CLQ signal at buffer 42 of the clear-MQC circuit (drawing 1-16B), which clears the MQC. The following operations take place during the OR phase:

(1) Complementing and Left Shift of Register A. Function signal 55+ blocks the recirculation of register A (drawing 1-8B). At the same time the dividend is read out of register A onto the S lines by way of the A11, A21, A31, and A41 lines. The quinary bits of each character of the dividend on the S1, S2, and S3 lines enter the quinary-adder complementer circuit (drawing 1-13B). The CP signal alerts the complementing gates to complement the first three (quinary) bits. The quinary bits (S1C, S2C, and S3C) reenter register A at left-shift gates 43 through 45 (drawing 1-8B), alerted by FS71. While circulating through the complementer circuit, the first three bits are delayed (shifted left or, in effect, multiplied by ten) one pulse time. The fourth or binary bit of each character in register A is delayed one pulse time (shifted left) within register A and reenters register A at left-shift gate 42 as an $\overline{A''4}$ signal. Gate 42, alerted by FS71 and CP, complements the fourth bit to complete shifting and complementing the characters of the dividend.

(2) Complementing and Left Shift of Register X. The contents of register X, including the sentinel or sentinels, leave the register and enter the input gates of the MQC flip-flops (drawing 1-16B), alerted by the IER-OR control signal. The MQC flip-flops, cleared by the CLQ signal, supply one pulse time of delay for the X1, X2, X3, and X4 outputs of register X. One pulse time after entering the flip-flops, the X input bits become the Q outputs of the flip-flops and return to register X at quotient-complementing gates 10A, 10B, 11, 12, 13, and 17 (drawing 1-10B). The OR control signal alerts these gates, which complement the Q inputs from the MQC. The entire contents of register X are complemented and shifted left. During the first OR cycle only the sentinel or sentinels are stored in register X, whereas in succeeding OR cycles the quotient digits also are stored in register X.

(3) Complementing the LSD of Register A. During the first OR phase, the original dividend is complemented and shifted left. In an ordinary register-A left-

Table 4-3. Division Process With and Without Programmed Sentinel

PROBLEM STORED IN COMPUTER:

rL	rA
0012000000	$\overline{0001550000}$

(a) With Programmed Sentinel

Step	MQC	Phase	OF	Dividend			Quotient		
				SBW	rA MSD LSD	SIGN	SBW	rX	SIGN
D1	0				0001550000			00P0000000	Z
D2	10	0R odd		9	9984499999			9P9999999Z	
	9	0R			+12				
				9	9996499999				
	8				+12				
			1	0	0008499999			9P9999999Z	
	10	0R even		9	9915000000			P0000000Z1*	
	9	0R			+12				
				9	9927000000				
	8				+12				
				9	9939000000				
	7				+12				
				9	9951000000				
	6				+12				
				9	9963000000				
	5				+12				
				9	9975000000				
	4				+12				
				9	9987000000				
	3				+12				
				9	9999000000				
	2				+12				
			1	0	0011000000			P0000000Z1	
D3	0				0000000012 Final quotient			0011000000 Final remainder	

*Set overflow (0F) flip-flop.

(b) Without Programmed Sentinel

Step	MQC	Phase	OF	Dividend			Quotient		
				SBW	rA MSD LSD	SIGN	SBW	rX	SIGN
NOTE: Same as steps D1 and D2 with programmed sentinel.									
(D2)	10	0R odd		9	9889999999			9999999Z87	
	9	0R			+12				
				9	9901999999				
	8				+12				
				9	9913999999				
	7				+12				
				9	9925999999				
	6				+12				
				9	9937999999				
	5				+12				
				9	9949999999				
	4				+12				
				9	9961999999				
	3				+12				
				9	0073999999				
	2				+12				
				9	9985999999				
	1				+12				
				9	9997999999				
	0				+12				
			1	0	0009999999			9999999Z87	

Table 4-3. Division Process With and Without Programmed Sentinel (cont)

(b) Without Programmed Sentinel (cont)

Step	MQC	Phase	OF	Dividend			Quotient		
				SBW	rA MSD LSD	SIGN	SBW	rX	SIGN
(D2)	10	OR even		9	9900000000			000000Z129	
	9	OR		9	+12 9912000000				
	8			9	+12 9924000000				
	7			9	+12 9936000000				
	6			9	+12 9948000000				
	5			9	+12 9960000000				
	4			9	+12 9972000000				
	3			9	+12 9984000000				
	2			9	+12 9996000000				
	1		1	0	+12 0008000000			000000Z129	
	10	OR odd		9	9919999999			99999Z8708	
	9	OR		9	+12 9931999999				
	8			9	+12 9943999999				
	7			9	+12 9955999999				
	6			9	+12 9967999999				
	5			9	+12 9979999999				
	4			9	+12 9991999999				
	3		1	0	+12 0003999999			99999Z8708	
	10	OR even		9	9960000000			0000Z12916	
	9	OR		9	+12 9972000000				
	8			9	+12 9984000000				
	7			9	+12 9996000000				
	6		1	0	+12 0008000000			0000Z12916	
	10	OR odd		9	9919999999			999Z870833	
	9	OR		9	+12 9931999999				
	8			9	+12 9943999999				
(D2)	7	(OR)		9	+12 9955999999				
	6			9	+12 9967999999				
	5			9	+12 9979999999				
	4			9	+12 9991999999				
	3		1	0	+12 0003999999			999Z870833	
	10	OR even		9	9960000000			00Z1291666	
	9	OR		9	+12 9972000000				
	8			9	+12 9984000000				
	7			9	+12 9996000000				
	6		1	0	+12 0008000000			00Z1291666	
	10	OR odd		9	9919999999			9Z87083333	
	9	OR		9	+12 9931999999				
	8			9	+12 9943999999				
	7			9	+12 9955999999				
	6			9	+12 9967999999				
	5			9	+12 9979999999				
	4			9	+12 9991999999				
	3		1	0	+12 0003999999			9Z87083333	
	10	OR even		9	9960000000			Z129166666*	
	9	OR		9	+12 9972000000				
	8			9	+12 9984000000				
	7			9	+12 9996000000				
	6		1	0	+12 0008000000			Z129166666	
D3	0				1291666666 Final quotient			0008000000 Final remainder	

*Set overflow (OF) flip-flop.

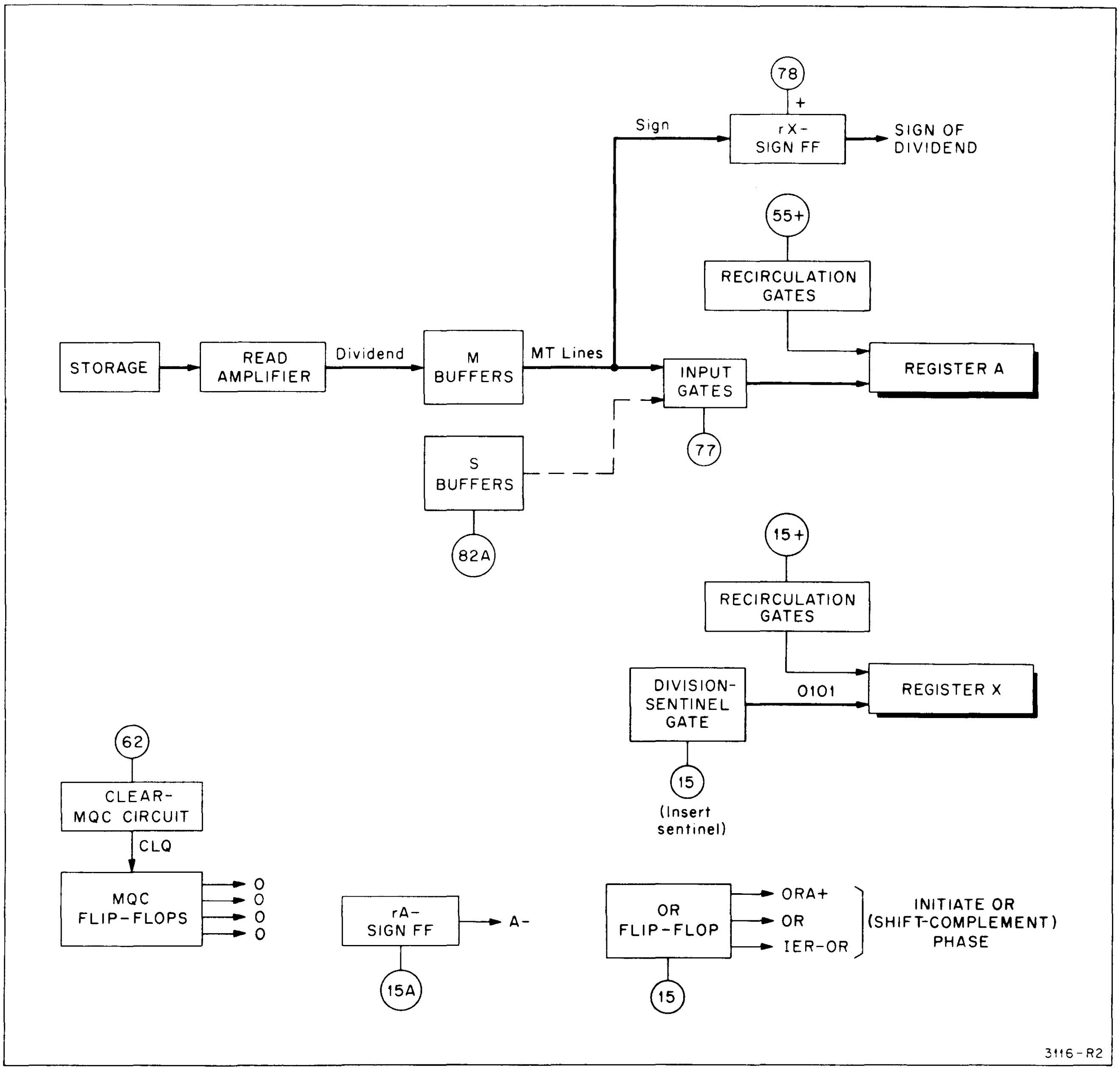


Figure 4-12. D1 Step of Division

shift (37) instruction, the vacancy left in the LSD position is filled with a 0. In division, however, the LSD position of register A is filled with a 9 during the odd OR phases and with a 0 during the even OR phases to conform with the state (complemented or true form) of the other information in register A.

The LSD-complementer gate 30 of register A (drawing 1-8B) is alerted by the OR output of the OR flip-flop. At t1B, the gate samples the X4D output of register X. Only the X4D bit is necessary to determine whether a 9 or a 0 is stored in the SBW position of register X. If the SBW of register X is 0, a 9 is inserted in register A because gate 30 is made permissive; if the SBW of register X is 9, a 0 is inserted in register A because gate 30 is blocked.

(4) Jam MQC. At t11B of every OR phase, the OR signal makes gate 34C of MQC FF4 (drawing 1-16B) permissive to generate the set-1101 signal which jams 1's into MQC flip-flops 1, 3, and 4. This action stores a 10 combination (1101) in the MQC, which is counted down during the OR phase.

(5) Initiate OR Phase. At the end of the OR phase, a high t11B+ signal restores the OR flip-flop (drawing 1-17B) to initiate the $\overline{\text{OR}}$ phase.

4-49. $\overline{\text{OR}}$ Phase. Figure 4-14 is a block diagram of the $\overline{\text{OR}}$ phase, during which the divisor from register L and the dividend from register A enter the quinary- and binary-adder circuits where the two quantities are added. The MQC-countdown circuit monitors the number of

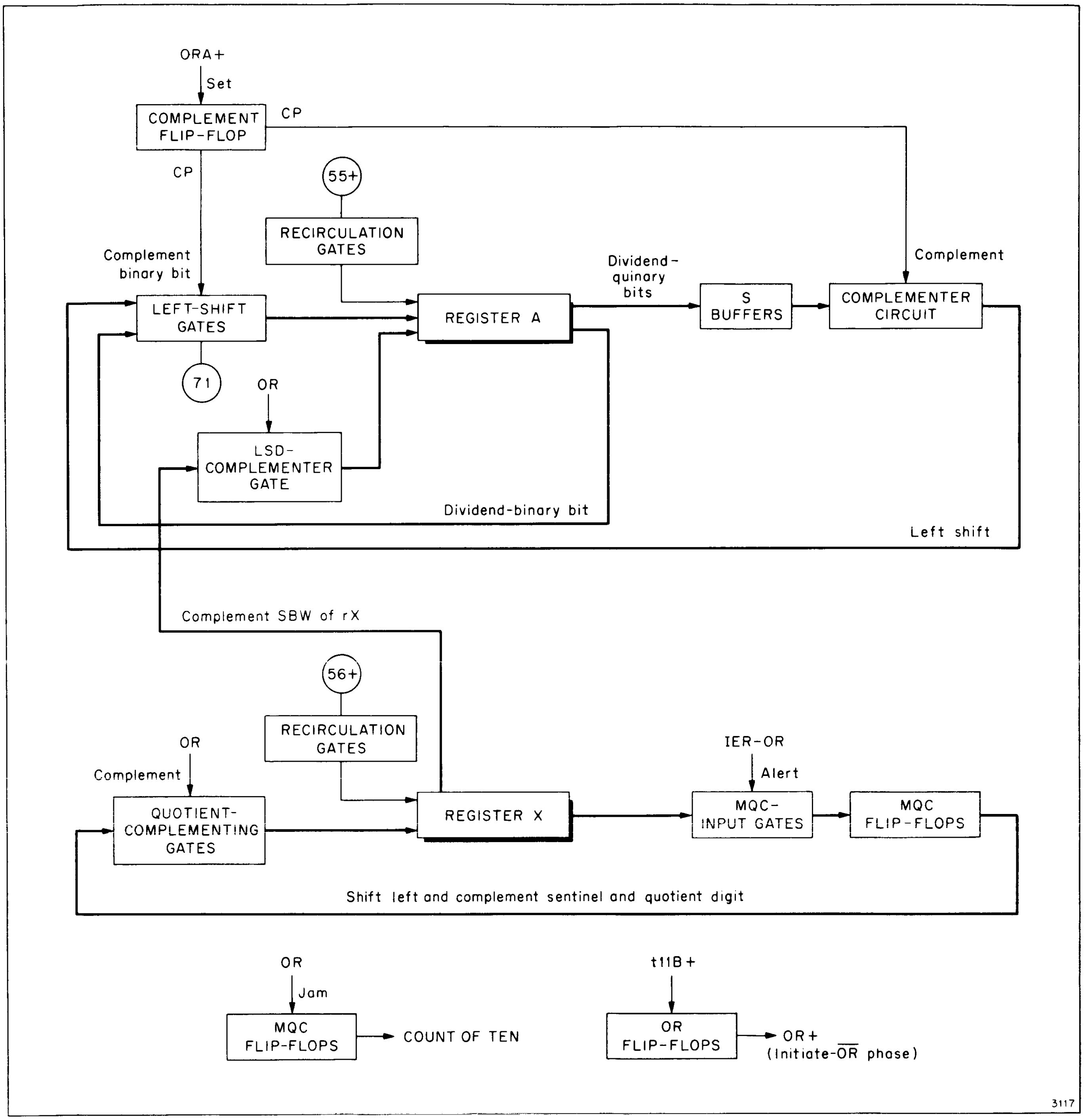


Figure 4-13. D2 Step of Division, OR Phase

additions. The sum of the divisor-to-dividend additions returns to register A for temporary storage. Addition and the end-of-D2 step are performed as follows:

Addition. The divisor, already placed in register L by a previous instruction, is recirculating them. Function signal 66 and the OR flip-flop output, OR+, which is low at this time, make register L output gates 4A through 4D (drawing 1-12B) permissive. The divisor in register L is read onto the L lines to the M buffers and M lines which carry information to the quinary and binary adders. Register L continues to recirculate because the recirculation gates are not blocked. Function signal 55+ blocks the recirculation path of register A (figure 4-14), and the dividend is read out of register A onto the S buffers and S lines. The contents of register A are read continuously onto the S lines unless blocked by FS58+. The S4 binary bits on the S lines are read into the comparator. The three lowest order (quinary) bits are read into the quinary-adder circuit (drawing 1-13B), as in the addition process

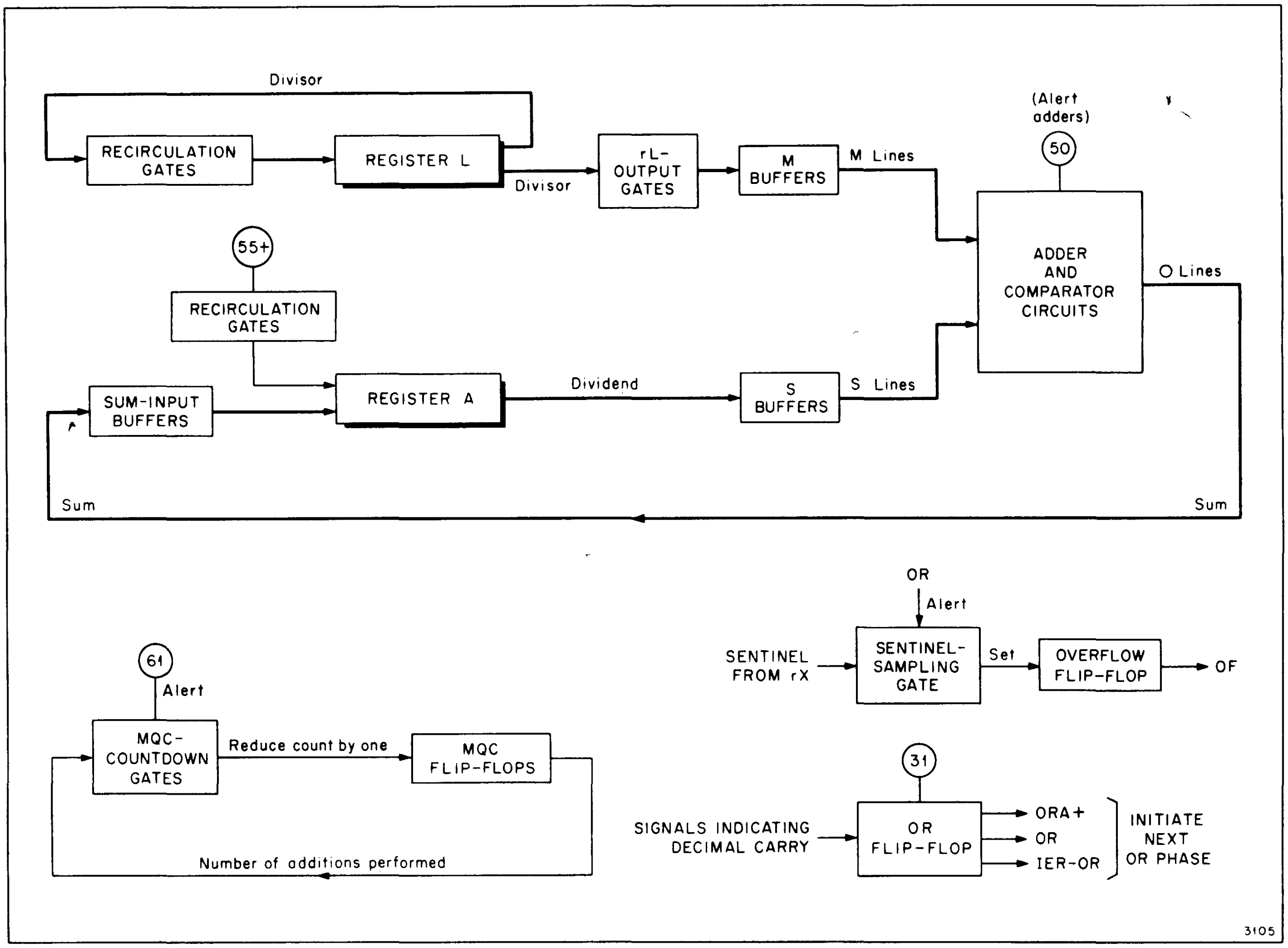


Figure 4-14. D2 Step of Division, $\overline{OR}$ Phase

(heading 4-29). Function signal 50 alerts the quinary adder and comparator to the M and S inputs. The divisor and dividend are added and the sum read onto the O lines, which return the sum to register A.

With each addition, the MQC-countdown gates (drawing 1-16B), alerted by FS61, reduce by one the number stored in the MQC flip-flops every t1B of the $\overline{OR}$ phase. When decimal carry occurs, the remainder is on the O lines and the MQC contains the number of additions performed.

Throughout the D2 step, FS31 alerts gate 10 of the OR flip-flop (drawing 1-17B) to sample the $\overline{OF}$, A, C, S3, and S4 signals. The A and C signals at t11B are generated because the addition of the p10 digit produced a decimal carry to the p11 digit. The S3 and S4 signals (1100 or 9) are present because the p11 digit is a 9. If a decimal carry is added to the final digit to be added, a 9 (carry condition) will occur. The low $\overline{OF}$ signal is present when the OF flip-flop is restored, as in the add phases. If all of these signals are present, the flip-flop is set to initiate the OR phase, which causes the OR, ORA+ and IER-OR control signals to be generated.

End of D2 Step. During every OR phase of D2, the automatic division sentinel is shifted toward the MSD position of register X. At t0B of the tenth OR cycle, the sentinel that samples gate 27 (drawing 1-15B) examines the X1 and X3 bits of register X, which are present when the sentinel is in the MSD position of register X. If the gate is alerted by OR indicating that the last addition has ended, the OF flip-flop is set. An OF signal is generated to step to D3 as soon as the OR phase is completed, determined by a decimal carry represented by A and C signals. A programmed sentinel may make this sampling gate permissive before the automatic sentinel reaches the MSD position.

4-50. D3 STEP. During the D3 step (figure 4-15) the contents of registers A and X are interchanged, with the final quotient ending in register A and the final remainder in register X. The sign of the quotient is computed during the D3 step to end the instruction. The two phases of the D3 step are described under the following two headings.

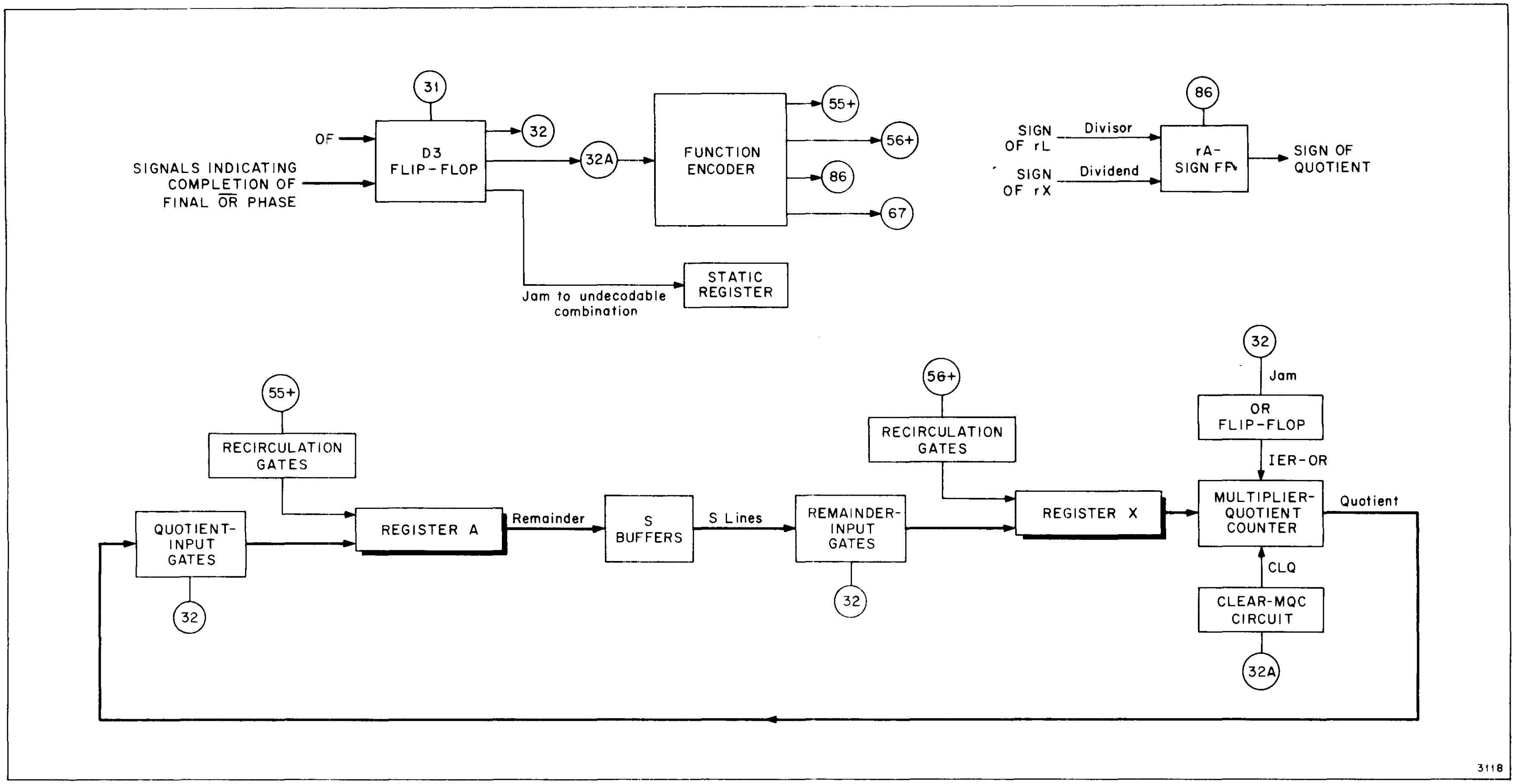


Figure 4-15. D3 Step of Division

4-51. Interchanging Register A and Register X. Function signal 31 alerts set gate 1 of the D3 flip-flop (drawing 1-2B) throughout the D2 step. When the following signals are present at gate 1, it is permissive, initiating the D3 step of division:

OF (present when the sentinel is in the MSD position of register X);

A and C (present when a carry exists);

S3 and S4 (present when the digit 9 is in the SBW of of register A); and

STR8 (present when incorrect division has not occurred; incorrect division generates EP, which restores all STR flip-flops).

Because OF is low, $\overline{\text{OF}}$ is high to block set gate 10 (drawing 1-17B) of the OR flip-flop to prevent the setting of the flip-flop.

Coincidence of the signals just listed at t11B of the tenth $\overline{\text{OR}}$ phase sets the D3 flip-flop (drawing 1-2B), which generates low FS32 at t0B and high FS32A at t1A. When the D3 flip-flop is set, static register FF5 is set, which results in a combination that cannot be decoded by the instruction decoder. Therefore, no function signals are generated by the instruction decoder, and the function signals from the D3 flip-flop control the D3 step until the flip-flop is restored at t11B.

Function signal 32A causes the function encoder to generate FS55+, FS56+, FS86, and FS67. Function signal 55+ blocks recirculation of information in register A. The final remainder is read from register A onto the S buffers and S lines. Function signal 32 alerts the register-X input gates to the contents of register A, which are on the S lines.

Function signal 56+ blocks recirculation of information in register X. The quotient is read from register X and enters the input gates of the MQC flip-flops. These gates are alerted by the IER-OR control signal generated by FS32A. Function signal 32A also generates CLQ to restore each MQC flip-flop every pulse time, unless the flip-flop is set by the information from register X. As the quotient in register X is shifted during the one-pulse-time delay supplied by the MQC flip-flops, the remainder on the S lines is read into register X. The outputs of register X are read through the MQC flip-flops and onto the Q lines. Function signal 32 alerts the quotient-input gates of register A to the quotient on the Q lines. Thus, the remainder is transferred into register X and the quotient is transferred into register A. Function signal 67 generates an EP signal at t9B. By t11B, the EP signal ends the D3 step, thus completing the instruction.

4-52. Sign of the Quotient. During the D3 step, the sign of the quotient is computed at gates 1 and 2 of the rA-sign flip-flop (drawing 1-15B). Function signal 86 alerts these gates to sample the sign of the divisor, which is stored in the rL-sign flip-flop, and the sign of the dividend, which is stored in the rX-sign flip-flop. The rA-sign flip-flop is initially set to an A– signal. If the signs of the original divisor and dividend indicate that the quotient should be positive, the flip-flop is restored to generate an A+ signal for the sign of the quotient.

4-53. ERROR CIRCUITS

The main-storage-check flip-flop (drawing 1-21B), the timing-error flip-flop (drawing 1-19B), and the cycling-unit-error flip-flop (drawing 1-18B) continuously check Processor functions for parity or timing errors. The MRE, TE, or CUE signal produced by the setting of any of these flip-flops sets the stop flip-flop (drawing 1-5B) to generate a high $\overline{\text{SP}}$ signal that stops the Processor by blocking the gates of the instruction decoder to prevent function signals from being generated.

An error or abnormal operation of an input-output unit sets the I/O abnormal-operation flip-flop (drawing 1-2B). When this flip-flop is set, the occurrence of any succeeding instruction that makes use of the abnormally operating unit causes the next instruction to be taken from the c + 1 instead of the c address. The assumption is made that the first instruction of a stop routine has been stored in c + 1.

4-54. MAIN-STORAGE-CHECK FLIP-FLOP

When information is written onto a main-storage band, the check-bit-computer circuit (drawing 1-21B) ensures that an odd number of 1 bits is recorded on the drum for each digit. If a digit contains an even number of 1 bits, a low CK signal is generated to write a 1 check bit. If a digit contains an odd number of 1 bits, a low $\overline{\text{CK}}$ signal is generated to write a 0 check bit. When this same information is read from the drum, the check-bit-computer circuit checks the digit and again generates a CK or $\overline{\text{CK}}$ signal, which alerts set gates 53A and 53B of the main-storage-check flip-flop.

If the check bit read with the digit is a 0, the $\overline{\text{DM5B}}$ signal is low. If the check bit read is a 1, the DM5B signal is low. The main-storage-check flip-flop is set to generate a high MRE (main-storage-read error) signal if either of the following conditions is present:

A 0 check bit is required ($\overline{\text{CK}}$ signal low) and a 1 check bit is read (DM5B signal low); or

A 1 check bit is required (CK signal low) and a 0 check bit is read ($\overline{\text{DM5B}}$ signal low).

The main-storage-check flip-flop can be set (high MRE generated) only if the check-timing flip-flop is set. This check-timing flip-flop is set at t0B if the RD signal is present at gate 47 (main-storage-read flip-flop is set). The low output of the check-timing flip-flop alerts gates 53A and 53B of the main-storage-check flip-flop to sample for parity errors.

The high MRE signal lights the MAIN STORAGE indicator on the Processor engineering panel and the PROCESSOR/OFF NORMAL indicator on the operator's control panel, and sets the stop flip-flop. The high $\overline{\text{SP}}$ output of the stop flip-flop goes to the instruction decoder to block generation of function signals for execution of the next instruction until the error condition is cleared.

The COMPUTATION/RUN pushbutton on the Processor operator's panel must be pressed to generate a high ST (start) signal. This signal restores the main-storage-check flip-flop to a low MRE output and the stop flip-flop to a low $\overline{\text{SP}}$ output. The instruction decoder then generates function signals to resume operations.

4-55. TIMING-ERROR FLIP-FLOP

The timing-error flip-flop (drawing 1-19B) samples the output digits of the timing-band read circuits. When originally recorded on the timing band of the drum, each permanent timing combination contains an odd number of 1 bits. The timing-error flip-flop maintains a constant check on the digit outputs of the timing-band read circuits to ensure that each digit contains an odd number of 1 bits. Thus, the operation of the drum and the read circuits is constantly checked.

Gates 2, 3, 4, and 5 of the timing-error flip-flop sample the TS1, TS2, and TS3 outputs of the timing-band read circuits. Gates 6 and 7 sample the TS4 and TS5 outputs. An even combination of 1 bits alerts two of the gates, setting the flip-flop to generate a high TE (timing-error) signal. For example, gates 2 and 6 operate if the TS combination is 0 0000. This combination contains an even number (0) of 1 bits; therefore an error exists. Gate 2 operates to send a high signal to buffer 14, which sends a low signal to gate 9. Gate 6 operates to send a high signal to buffer 16, which sends a low signal to gate 9. Both inputs to gate 9 therefore are low, setting the flip-flop to generate a high TE signal.

The timing-error flip-flop is set to generate a high TE signal, indicating an error, if either of the following conditions exists:

> Each of the two groupings of gates receives an even combination (even + even = even); or
>
> Each of the two groupings receives an odd combination (odd + odd = even).

An odd input combination in one grouping plus an even input combination in the other grouping (odd + even = odd) places a high on both set gates, and the timing-error flip-flop output (TE) remains low. If an error exists the high TE signal lights the timing-error (STORAGE/ADDRESS) indicator on the Processor engineering panel and the PROCESSOR/OFF NORMAL indicator on the Processor operator's panel, and sets the stop flip-flop.

4-56. CYCLING-UNIT-ERROR FLIP-FLOP

The cycling-unit-error flip-flop (drawing 1-18B) constantly checks the synchronization of the timing-combination input to the cycling unit. This combination, TS4, TS3, $\overline{\text{TS2}}$, and TS1, must be present at t6B of every word time to synchronize Processor and input-output operations correctly. If this combination is absent at t6B, the cycling-unit-error flip-flop generates a signal that stops the Processor and lights the CYCLING UNIT error indicator on the Processor engineering panel.

Gate 3 of the flip-flop is blocked by the high t6B+ signal at t6B, which is the time that the timing combination should be present. At any time other than t6B, the t6B+ signal is low. Therefore, if the bits of the timing combination are low at the gate at any time other than t6B, the gate is permissive, setting the flip-flop. Gates 4, 5, 6, and 7 check the individual bits of the timing combination for synchronization. These gates are alerted by the t6B– signal at t6B. A malfunction is present if any one of the $\overline{\text{TS1}}$, TS2, $\overline{\text{TS3}}$, and $\overline{\text{TS4}}$ signals is low at t6B, since only the timing-combination signals should be low at that time. If the input signal to one of these gates is low at t6B, the gate is permissive. This condition again sets the flip-flop to a high CUE output. The low output signal from buffer 11 lights the CYCLING UNIT indicator on the Processor engineering panel and the PROCESSOR/OFF NORMAL indicator on the Processor control panel, and the high CUE signal sets the stop flip-flop.

4-57. INPUT-OUTPUT ABNORMAL-CONDITION FLIP-FLOP

Abnormal operation of the printer, read-punch, or card reader produces a signal (AOP, AOB, or AOR) that alerts a set gate of the I/O abnormal-condition flip-flop (drawing 1-2B). When a succeeding instruction that controls the abnormally operating unit occurs, the gate is made permissive by a function signal to set the flip-flop. High jam-I2A and jam-I2B signals generated by the set flip-flop change the static-register reading to simulate a step of a test instruction (common test) which normally transfers the contents of register C to register A and sets the CT flip-flop (drawing 1-14B). The input-output instruction is transferred from register C to register A, but the setting of the CT flip-flop is blocked by the jam-I2B signal. The jam-I2A signal sets the overflow flip-flop (drawing 1-15B) and restores the CT flip-flop so that the next search is for the c + 1 address of the input-output instruction. The assumption is made that the first instruction of a stop routine has been stored in c + 1.

For example, abnormal operation of the printer generates a low AOP signal which alerts gate 100 (drawing 1-2B). Function signal 41 of the next print instruction makes the gate permissive to set the flip-flop. The static-register contents are alerted to store the print instruction in register A and to search for the c + 1 address. The AOR signal from the card reader and the AOB signal from the read-punch unit operate identically to the AOP signal.

The CCT input to buffer 105 also generates one pulse time of the jam-I2B signal to transfer the 72 (feed-card-into-reader) instruction from register C into register A and to read the next instruction from the m address. The CCT signal is generated only if a second 72 instruction is given before the previous one is completed. The assumption is made that the first instruction of an error routine has been stored in the m address.

4-58. MANUALLY CONTROLLED OPERATIONS

The computer can operate either continuously or noncontinuously. Although it normally functions in the continuous mode, facilities are provided, mainly for troubleshooting and testing procedures, to operate the computer in the noncontinuous or step-by-step mode.

The operator controls the mode of operation of the computer by pressing one of the eight illuminating pushbuttons under OPERATION on the Processor control panel. These buttons light when pushed and are mechanically interlocked so that only one can be pressed at a time. When the CONTINUOUS button is pushed, the computer operates continuously until a programmed stop instruction is staticized, until an error occurs, or until the operator stops computation by pressing the STOP button. To place the computer in a noncontinuous mode, one of the seven buttons marked ONE OPERATION/TAPE, H.S.P., F.R., RPU, COMPARISON STOP, ONE INSTRUCTION/W/O INDEX REGIS., W INDEX REGIS., must be pressed. Logical operation of the noncontinuous modes is described under the following headings. When the CONTINUOUS pushbutton switch is energized, an NC signal is generated to synchronize the control-panel signal with Processor signals. The NC signal alerts gate 1 of the NC-synchronizing flip-flop (drawing 1-5B), which is permissive at t0B of the next EW signal. The NC signal is stored in the synchronizing flip-flop for two word times, after which it goes to gate 9 of the IOS flip-flop. Gate 9, which is alerted by the EW signal, is permissive at t0B to set the flip-flop. The IOS flip-flop generates an IOS signal which alerts set gates 23 and 24 of the stop flip-flop. When the IOS flip-flop is set, a high $\overline{\text{IOS}}$ signal is generated at buffer 15 to restore the synchronizing flip-flop.

When either gate 23 or 24 is permissive, the stop flip-flop is set to generate low SP and high $\overline{\text{SP}}$ signals. The $\overline{\text{SP}}$ signal blocks the instruction decoder so that no function signals are generated. When the COMPUTATION/RUN button is pressed, a high ST signal is generated which restores the stop flip-flop at gate 25.

4-59. ONE OPERATION/H.S.P.

When the ONE OPERATION/H.S.P. button is pressed, the computer stops on the first step of the next advance-and-print (11) instruction. An EPR signal is set to buffer 101 to set the abnormal-operation-printer (AOP) flip-flop (drawing 1-37B).

4-60. ONE OPERATION/RPU

When the ONE OPERATION/RPU button is pressed, the computer stops on the first step of the next card-cycle instruction involving the read-punch unit. AN ECC signal is generated and sent to buffer 24 of the abnormal-operation-read-punch (AOB) flip-flop (drawing 1-30B).

4-61. ONE OPERATION/F.R.

When the ONE OPERATION/F.R. button is pressed, the computer stops on the first step of the next card-cycle instruction for the card reader. An RECC signal is generated and sent to buffer 73 of the abnormal-operation-reader flip-flop (drawing 1-26B).

4-62. ONE INSTRUCTION

When the ONE INSTRUCTION/W/O INDEX REGIS. button is pressed, the Processor stops after the next instruction is set up in register C, but before it is executed. If that instruction calls for an an index-register modification, the modification will be ignored. After the instruction is set up in register C, information may be typed in from the keyboard (described under heading 4-64).

When the ONE INSTRUCTION/W INDEX REGIS. button is pressed, the Processor stops as described in the preceding paragraph, but the index-register modification (if specified) is executed. Information also may be typed in after this button is pressed and the instruction is set up in register C.

4-63. COMPARISON STOP

When the COMPARISON STOP button is pressed, the computer stops after an 82 or 87 instruction has been executed but before the next instruction is searched for. The function of both the 82 and 87 instructions is to compare two quantities and, on the basis of this comparison, to determine whether the next instruction should be taken from the m or the c address. One of the two NEXT ADDRESS illuminating pushbuttons on the Processor operator's panel lights to indicate whether the next instruction to be searched for will be in the m or c address. When the COMPARISON STOP button has been pressed, the computer stops after the comparison is made and the next address has been selected. The address determined by the comparison can be overridden by pressing the appropriate button. The m or c button changes the state of the CT flip-flop and lights the lamp associated with the chosen address. The COMPUTATION/RUN button then is pressed, and the computer searches for the next instruction.

When the COMPARISON STOP button is pressed, a QT-stop signal is generated and sent to gate 24 of the stop flip-flop. The STR signals that make gate 24 permissive are common only to the 82 and 87 comparison instructions. If the inputs to gate 24 are low, the gate is permissive at t10B, and the stop flip-flop is set. The t10B signal ensures that the comparison instruction will be completed before the stop flip-flop is set.

4-64. KEYBOARD INPUT

The control-panel keyboard enables the operator to change the contents of the arithmetic registers (A, X, or L) by typing in new information. An instruction word also is typed into register C at the keyboard to load the program into the memory at the c address specified in the instruction word. The keyboard consists of ten keys (0 through 9), a keyboard-alert (A) pushbutton to energize the keyboard, a keyboard-ready indicator, which lights to indicate that the keyboard is ready for use, and two word-release pushbuttons (+ and −). With the latter two pushbuttons, the sign of the word is inserted, and the typed word is transferred from temporary storage into the selected register. (See figure 4-16.)

At all times during computation, the contents of register

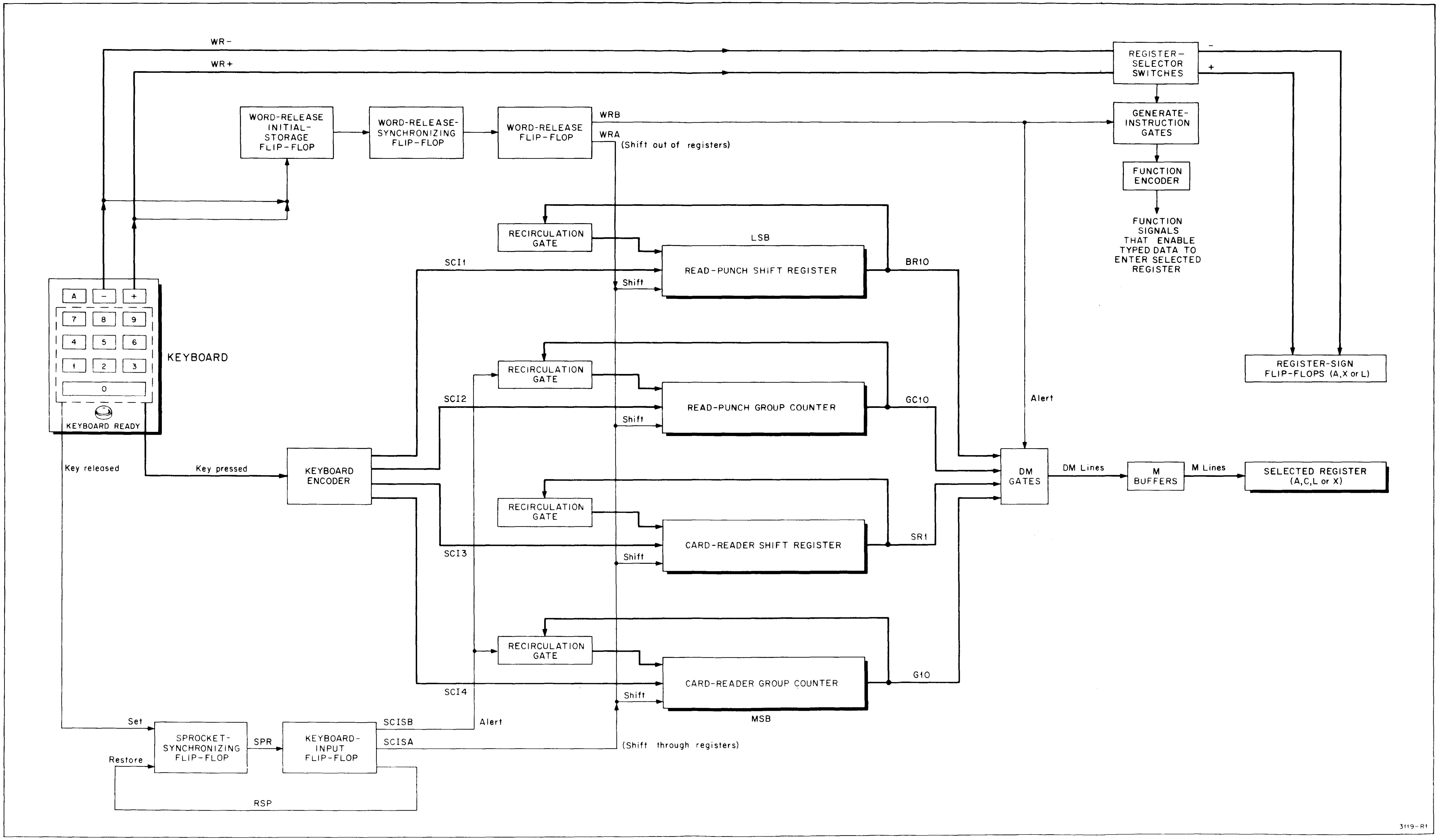


Figure 4-16. Keyboard-Input Operation

A, C, X, or L are displayed on the Processor operator's panel. Forty lights labelled REGISTER CONTENTS display the biquinary combinations of each of the ten digits contained in the register. Two sign lights (+ and −) to the left of the REGISTER CONTENTS lights indicate the the sign of the word in the register. The register to be displayed is selected by pressing one of four illuminating REGISTER SELECTOR pushbuttons on the operator's panel. These four buttons, C, A, X, and L, light when pressed and are interlocked so that only one can be operated at a time.

To type either instructions or data information into the computer, a register is selected and the computer stopped by pressing the COMPUTATION STOP button. When the computer stops, either ONE INSTRUCTION button and the keyboard-alert (A) button are pressed. After a delay of approximately 1 second, the keyboard-ready lamp on the control-panel keyboard lights to indicate that the keyboard is ready for use. The word now can be typed in, MSD first. When the complete word is typed, one of the word-release buttons (+ or −) is pressed to shift the word into the register selected and place its sign in the appropriate flip-flop. Then the OPERATION/CONTINUOUS and COMPUTATION/RUN buttons are pressed.

If a typing error is made, one of the word-release buttons is pressed to release the word into the register, the keyboard-alert button is pressed again, and the word is retyped.

Two keys can be pressed simultaneously to form bit combinations for characters not represented by keys on the keyboard. An alternate method is to type in two complete words. One of the word-release buttons then is pressed and the combination of the first and second words shifts into the selected register. A list of key combinations that can be used follows:

Biquinary Code	*Key Combination*	*Alphabetical Identity*
0101	1 and 4	A
0110	2 and 4	B
0111	3 and 4	C
1101	9 and 1	F
1110	9 and 2	G
1111	4 and 8	H

The word typed into the computer is stored temporarily in the read-punch shift register (drawing 1-29B), the card-reader shift register (drawing 1-26B), the card-reader group counter (drawing 1-27B), and the read-punch group counter (drawing 1-29B). (See figure 4-16.) The group counters function as shift registers during this phase. Each shift register stores ten bits of the 40-bit word typed. The LSD of the word is read out of the shift registers first, although the MSD is typed first. A CLIA signal, which clears all four shift registers, is generated when the keyboard-alert button is pressed. Signals necessary to shift the word are generated every time a key is released. When the entire word is typed and stored in the shift registers, one of the two word-release keys is pressed to supply the sign of the word and open the path into the selected registers.

4-65. PRESSING A KEY. Each time a key is pressed to place a digit in the four shift registers, the code for the desired digit is generated by the keyboard-encoder matrix (drawing 1-5B). The four bits of each digit are distributed onto the four output lines. The SC11 line enters the read-punch shift register, the SCI2 line the read-punch group counter, the SCI3 line the card-reader shift register, and the SCI4 line the card-reader group counter. Each bit is stored in the first flip-flop of a register. The outputs of the keyboard encoder, which is controlled by the keyboard keys, are shown in table 4-4.

A high SCI+ signal is generated from the beginning to the end of the keyboard input. The SCI+ signal goes to the card-reader group counter to block the setting of the sample-pulse flip-flop and to the read-punch group counter to block the stepping of the row counter.

Table 4-4. Keyboard Encoder

Keyboard Key	Card-Reader Group Counter	Card-Reader Shift Register	Read-Punch Group Counter	Read-Punch Shift Register	USS-6 Code (numeric bits only)
	SCI4	SCI3	SCI2	SCI1	
0					0000
1				X	0001
2			X		0010
3			X	X	0011
4		X			0100
5	X				1000
6	X			X	1001
7	X		X		1010
8	X		X	X	1011
9	X	X			1100

4-66. RELEASING A KEY. Each time a key is released a sprocket signal is generated that is synchronized with the timing of computer signals by the EW and t4B- signals which set the sprocket-synchronizing flip-flop (drawing 1-5B). This flip-flop generates a low SPR signal which sets the keyboard-input flip-flop (drawing 1-30B) at t4B, gate 37, when the EW signal is present. Three signals are generated: a low SCISB, a high SCISA, and a high RSP. The SCISA signal is available for nine pulse times (t6A through t3A), and in all four registers it shifts each bit of information nine places to the right. The recirculation path between the two group counters is completed by the SCISB signal. The other two shift registers are already recirculating. The RSP signal restores the sprocket-synchronizing flip-flop.

When a key is pressed, the four bits of the first digit stored in the four registers are shifted nine places to the right in all shift registers by the SCISA signal. This signal sends a high signal to the restore gates and a low to the set gates of the ten flip-flops of each register. Because of this shifting, the four bits of the first digit are placed in the LSD position of the four registers. When the next key is depressed, the four bits of the next digit are stored in the first flip-flop of each register. When the key is released, the SCISA signal again shifts the new bits nine places to the right and shifts the previously inserted bits the same number of places. The SCISB signal, sent to the group counters, permits the first bit in each counter to recirculate and become the second lowest order bit of the register. All ten digits of a word are read into the registers and shifted in the same manner. As a result, all digits of the word are in position in the register to be read out LSD first. (Refer to table 4-4.)

4-67. SIGNING AND RELEASING THE WORD. After all ten digits of a word have been typed, the word is given with a sign and released into the previously selected register by depressing either of the two word-release buttons (+ or –). If the + button is depressed, a WR+ signal is generated. If the – button is depressed, a WR– signal is generated. The WR+ or WR– signal is stored temporarily in the word-release initial-storage flip-flop (drawing 1-5B) and then is synchronized with computer timing by the setting of the word-release synchronizing flip-flop. The synchronized signal generated from this flip-flop sets the word-release flip-flop, which generates two signals, a high WRA and a low WRB. The WRB signal generates function signals that open the path to the selected register on the M lines and gate the information stored in all four registers onto the M lines. The WRA signal causes information to shift out of the four registers.

The WRB signal alerts DM gates 56A, 56B, 56C, and 56D (drawing 1-5B) so that the four output lines of shift registers read the contents onto the DM and M lines as follows:

Register	*Output*	*DM line*	*M line*
Read-punch shift register	BR10	DM11	M1
Read-punch group counter	GC10	DM21	M2
Card-reader shift register	SR1	DM31	M3
Card-reader group counter	G10	DM41	M4

The WRB signal also goes to generate-instruction gates 57, 58, 59 and 60, only one of which has been alerted. One of the RSC, RSA, RSL, or RSX signals alerts one of the gates, depending on which register has been selected. If the RSC signal has been generated, a generate-beta (GEN BETA) signal is produced. If the RSX signal has been generated, a generate-Y (GEN Y) signal is produced. If the RSL signal has been generated, a generate-L (GEN L) signal is produced, and if the RSA signal has been generated, a generate-B (GEN B) signal is produced. One of these four signals is sent to the function encoder to generate function signals that perform the instruction required to store the input data in the selected register. (Function signal 67, however, is not generated because the static register is not involved.) The WRB signal also is sent to gate 15 on the static register (drawing 1-2B) to generate an ending pulse if FS2 is present.

The WRA signal is sent to all four shift registers to shift the contents out of the registers by placing a high signal on the restore gates and a low on the set gates. The WRA signal also is sent to the keyboard-input flip-flop (drawing 1-30B) to generate an RSP signal which restores the word-release initial-storage and word-release syynchronizing flip-flops (drawing 1-5B).

The signal generated when an arithmetic register is selected (RSA, RSX, or RSL) is combined with either the WR+ or WR– signal to generate one of the following signals: WRA–, WRA+, WRX–, WRX+' WRL–, or WRL+. These signals go directly to the rA-, rX- or rL-sign flip-flops to store the correct sign in the correct flip-flop. If register C is selected to receive the word typed in, no sign is involved unless index registers are included.

Section 5

INSTRUCTIONS

5-1. INTRODUCTION

There are 34 instructions in the repertory. Eight are input-output instructions, which are described in the appropriate separate manuals or pertinent parts of manual 3; the remaining instructions are described in this section and in section 4 of this manual.

Instructions are described in this section as follows: (1) input-output, (2) arithmetic, (3) transfer, (4) translate, (5) miscellaneous, (6) comparison, and (7) test.

Section 4 of this manual describes in detail the search and staticize steps common to all instructions. Section 4 also describes the execution steps of the 60 transfer instruction and the arithmetic instructions, and (70), subtract (75), multiply (85), and divide (55). Knowledge of the 60 transfer, and the arithmetic instructions and the basic search-and-staticize sequence is necessary to understand the less detailed descriptions in this section. The execution sequence for the remaining six transfer instructions and the translation, comparison, miscellaneous, and input-output test instructions are described in this section at block-diagram level. The reader may assume that the initial-search and staticize-instruction steps have been completed, and that both the text and the figures in this section refer to the execution sequences.

Section 3 of this manual contains a functional description of the logical circuits that are mentioned in this section. Tables 2-1 and 2-2 of this part contain mnemonic and machine-code equivalents, together with function of instructions described in this section.

5-2. INPUT-OUTPUT INSTRUCTIONS

Input-output instructions control the transfer of information to and from the input-output units; they also control card movement, output-stacker selection, and so forth. Because a thorough understanding of these instructions requires knowledge of the input-output units, each input-output instruction is described in detail in the appropriate manual, or in the pertinent part of manual 3.

5-3. ARITHMETIC INSTRUCTIONS

The four arithmetic instructions, add (70), subtract (75), multiply (85), and divide (55), are described in detail under heading 4-27 of this part.

5-4. TRANSFER INSTRUCTIONS

The seven transfer instructions are the 60, 65, 50, 25, 05, 30, and 77 instructions.

5-5. 60, 65, AND 50 INSTRUCTIONS

The 60, 65, and 50 instructions transfer information from a register to a main-storage location. The 60 instruction, which transfers the contents of register A to a main-storage location, is described in detail in section 4. The 65 and 50 instructions are similar to the 60 instruction except that they transfer the contents of registers X and L, respectively, to main-storage locations.

5-6. 25, 05, AND 30 INSTRUCTIONS

The 25, 05, and 30 instructions transfer information from a main-storage location to a regiser. The 25 instruction transfers the contents of a main-storage location to register A. The 05 and 30 instructions are similar to the 25 instruction except that they transfer the contents of a main-storage location to registers X and L, respectively.

5-7. 25 INSTRUCTION. During the execution step of the 25 instruction (figure 5-1), the recirculation gates of register A are blocked so that the new word from storage can enter the register. The word from storage is read by the read amplifiers, goes through the M buffers, and enters the register-A input gates. A blocking signal into the S buffers forces the S lines low so that the input gates are permissive to information on the M lines only. The sign of the word from storage enters the rA-sign flip-flop. (See figure 5-1.) When the entire word from storage is in the register, the blocking signal on the recirculation gates is removed so that the word can recirculate in the register.

5-8. 77 INSTRUCTION

The 77 instruction transfers the contents of register A to register L; it is the only register-to-register transfer instruction. Because this instruction does not involve a search for an operand, the m-address portion of the instruction is ignored.

5-9. TRANSLATE INSTRUCTIONS

There are two translate instructions. The 12 instruction translates one word from card code to USS-6 code, and the 17 instruction translates one word from USS-6 code to card code. The card reader and read-punch operate with card code, while the Processor performs arithmetic operations only with USS-6 code.

5-10. 12 INSTRUCTION

When arithmetic operations are to be performed on information read by the card reader or read-punch unit,

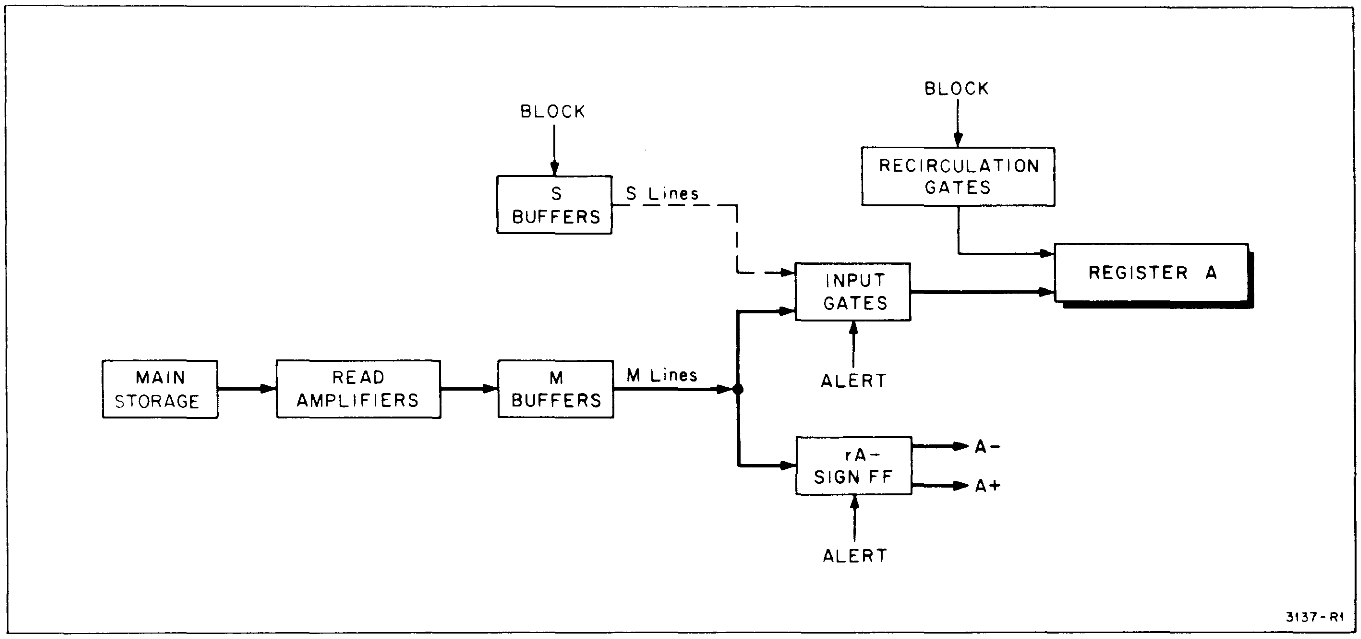


Figure 5-1. 25 (Transfer) Instruction

the 12 instruction is used to translate the information contained in registers A, X, and L from card code, in which it was supplied by the units, into USS-6 code, which is used by the Processor.

Before the 12 instruction is given, each word read in the 12-bit card code by the card reader or read-punch unit is divided automatically into three parts: unprimed, primed, and double-primed. The unprimed part is transferred into register A, the primed part into register L, and the double-primed part into register X. When the 12 instruction is given, the information stored in the registers is transferred to the translation gates (see figure 5-2); the recirculation gates of all three registers are blocked to clear the registers. The translated word, which has been divided into numeric and zone bits of the USS-6 code by the translation gates, is returned to registers A and X for storage, and the recirculation gates are opened so that the translated word can recirculate in these two registers, which now are available for arithmetic operations. Register L remains cleared.

5-11. 17 INSTRUCTION

After arithmetic operations are performed on information contained in registers A and X (in USS-6 code), the 17 instruction is given to translate output information that is to be punched into the card code used by the read-punch unit.

Before the 17 instruction is given, the numeric part of the USS-6 word is transferred into register A and the zone part into register X. When the 17 instruction is given, the information stored in the arithmetic registers is transferred to the translation gates (see figure 5-3), where it is divided into unprimed, primed, and double-primed parts. During the transfer, the recirculation gates of the registers are blocked to clear the registers. The translated word, now in card code, is returned to the registers, with the unprimed part stored in register A, the primed part in register X, and the double-primed part in register L, and the recirculation gates are opened so that the translated word can recirculate. The signs of registers A and X are jammed to plus. The word then is in suitable form for storage on the read-punch buffer band of the card buffer, from which is can be extracted for punching into a card.

5-12. MISCELLANEOUS INSTRUCTIONS

Eight instructions have been grouped in the miscellaneous category: the 20 (superimpose) instruction, the 35 (extract) instruction, the 32 (shift-right) instruction, the 37 (shift-left) instruction, the 62 (0-suppress) instruction, the 67 (stop) instruction, the optional 02 (index-register-load) instruction, and the optional 07 (index-register-add) instruction.

5-13. 20 (SUPERIMPOSE) INSTRUCTION

The 20 instruction superimposes the 1 bits of the word in the m-address location onto the word in register A. The new word placed in register A contains a 1 bit where either the original word in register A or the word from storage had a 1 bit. The sign of the original word in register A is not changed.

This instruction is used to combine the information in one word with the information in another word. Information in one word often must be combined with many other words, so that the information to be superimposed is stored as a constant.

As shown in figure 5-4, the information word in register A circulates there. Simultaneously, the word from storage enters register A on the M lines from the M buffers through the input gates of register A. The S buffers are blocked so that the register-A input gates receive informa-

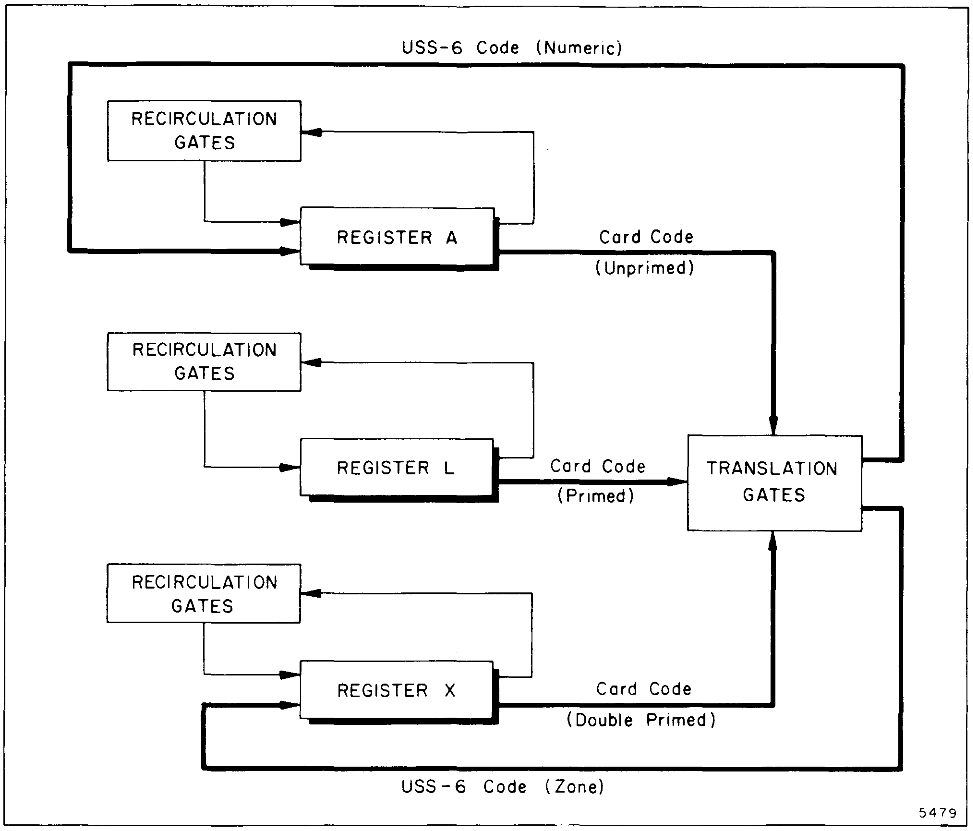


Figure 5-2. 12 (Translate) Instruction

tion on the M lines only. Thus, the 1 bits in register A recirculate, and the 1 bits in the word from main storage enter the register through the input gates.

5-14. 35 (EXTRACT) INSTRUCTION

The 35 instruction changes 1 bits to 0 bits in each decimal digit of the word in register A whenever there is a 0 bit in the corresponding bit position of the word in the main-storage location designated by the m address. The sign of register A is not changed.

This instruction is referred to as the extract instruction because it is used to extract certain digits from a word in register A. In the extract word from main storage, each bit of each digit contains a 1 bit where the corresponding digit in register A is to be unchanged, and a 0 bit where it is to be changed to 0.

For example, to extract the five MSD's of the word in register A by changing the five LSD's to 0's, the programmer places the extract word XXXXX 00000 in main storage. When 35 instruction containing this word as the m-address is given, the 0 bits in each bit position of the five LSD positions change these digits to 0's in the word in register A. Since the USS-6 code for a period is 1111, or four 1 bits, there are no 0 bits in the five MSD positions of the extract word, and the corresponding digits in the original word in register A remain unchanged.

The 35 instruction (figure 5-5) is similar to the 20 instruction (figure 5-4). In the 35 instruction, however, the word in register A does not recirculate because the recirculation gates are blocked. Instead of recirculating, it is sent to the S buffers, so that it is on the S lines at the same time that the extract word from storage is on the M lines. Corresponding digits of both words, therefore, are at the input gates of register A at the same time. The new word formed in register A as a result of this instruction contains 1 bits only in the bit positions where the S and M lines had 1 bits in corresponding bit positions.

5-15. 32 (SHIFT-RIGHT) INSTRUCTION

The 32 instruction shifts the contents of register A to the right a designated number of places and into register X and simultaneously shifts the contents of register X to the right and into register A. The number of places shifted can vary from zero through ten. The p7 digit of the m address of the instruction word specifies the number of places to be shifted. (The p7 position is the next-to-most significant digit of the m address.) The bit-code combination 1101 is used to represent 10 in this instruction.

During the staticize step of every instruction, the p7 digit of the m address is placed in the multiplier-quotient counter (MQC) where it is available if the instruction is a shift instruction.

The 32 (SHR) instruction is executed in two steps, SHR1 and SHR2.

During the SHR1 step (figure 5-6), the digit in the MQC

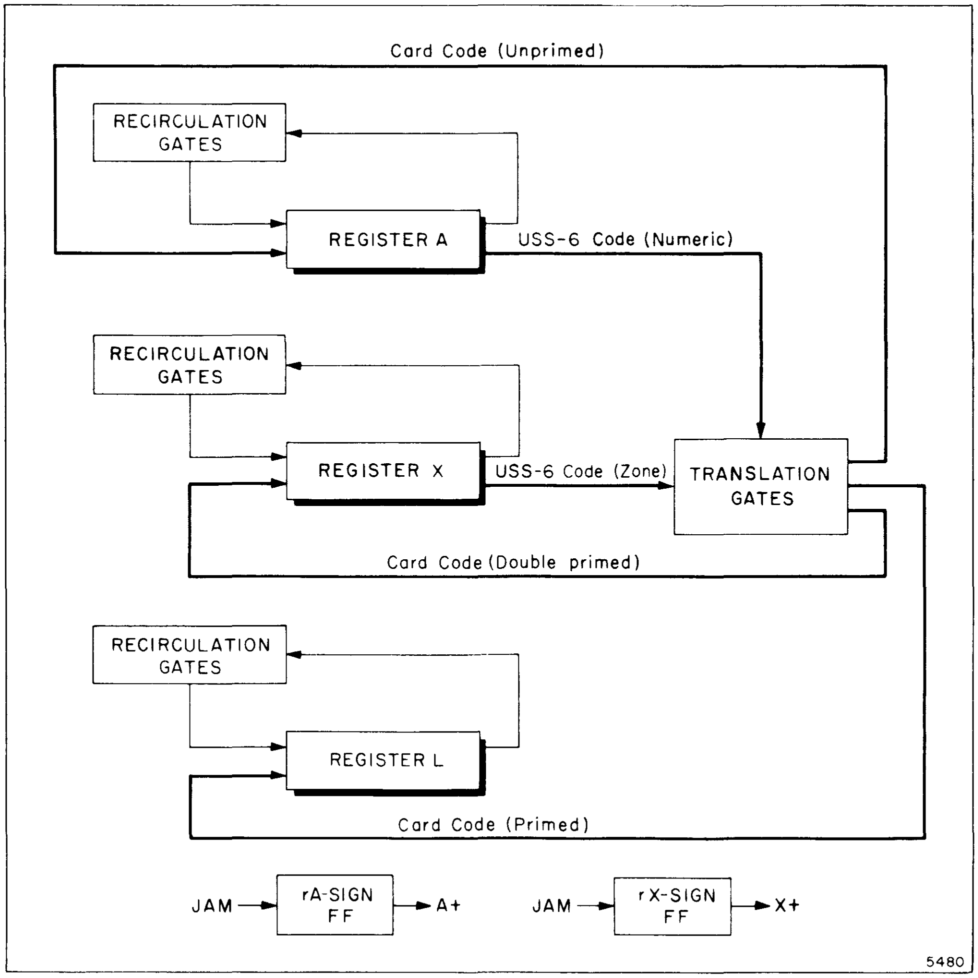


Figure 5-3. 17 (Translate) Instruction

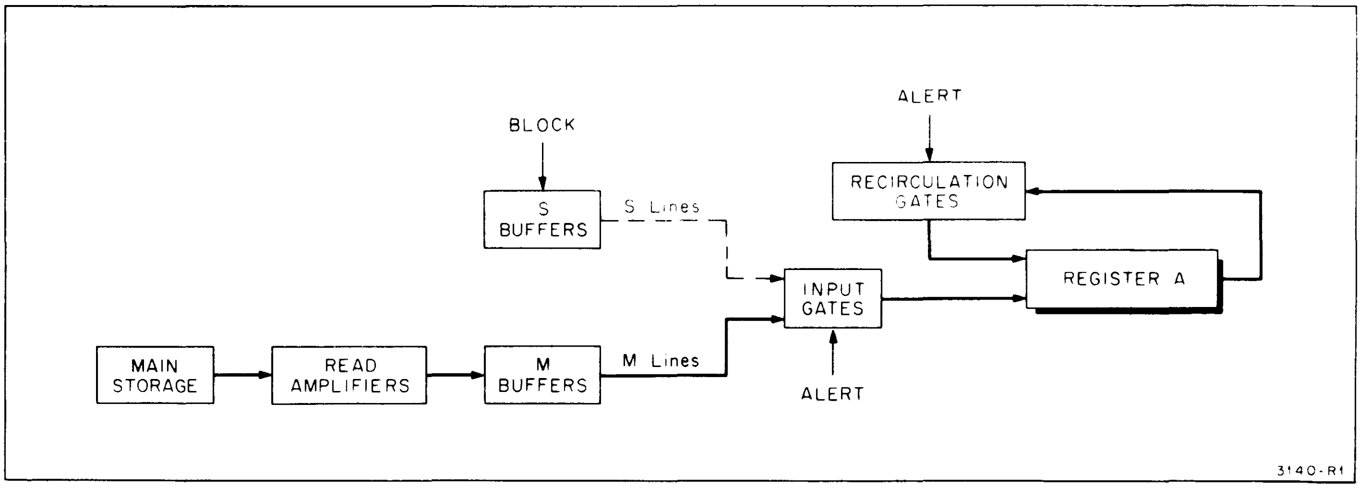


Figure 5-4. 20 (Superimpose) Instruction

(the p7 digit of the m address) is examined. If the digit in the MQC is a 0, an ending pulse clears the static register to all 0's and the next instruction is searched for. If the digit in the MQC is a 1, the static register is stepped to the SHR2 step of the 32 instruction.

During the SHR2 step of this instruction, the contents of registers A and X are shifted to the right by blocking the normal recirculation gates of registers A and X and circulating the contents of registers A and X through a loop shortened by one digit. During the SHR2 stage, the con-

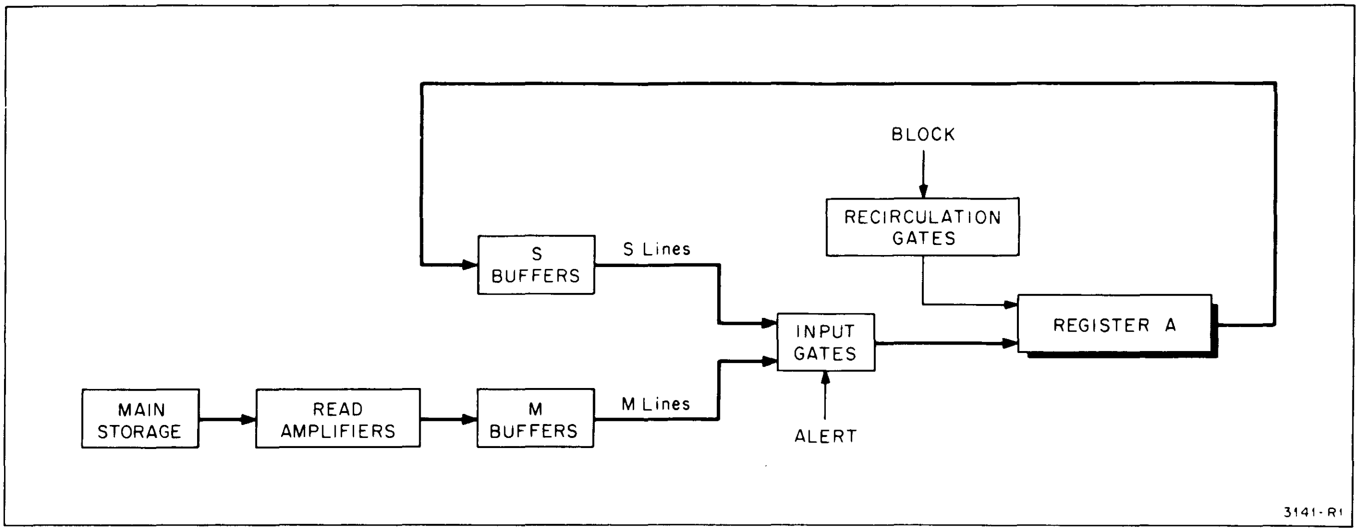


Figure 5-5. 35 (Extract) Instruction

tents of registers A and X are circulated through the shortened loop the number of times specified by the MQC digit. During each circulation, the contents of registers A and X are shifted one place to the right (figure 5-6). A circular-shift operation also takes place, in which the LSD of register A becomes the MSD of register X, and the LSD of register X becomes the MSD of register A. During each shift operation, the MQC countdown gates count down by one until the MQC digit is 0. An MQC digit of 0 generates an ending pulse.

5-16. 37 (SHIFT-LEFT) INSTRUCTION

The 37 instruction shifts the contents of register A to the left the number of places specified by the MQC. The MSD's are shifted out and 0's are placed in the LSD positions. The number of places shifted may vary from zero through ten. The code combination 1101 is used to represent 10 in this instruction.

The 37 instruction is executed in two steps, SHL1 and SHL2 (see figure 5-7).

During the SHL1 step, the MQC digit is examined. This digit is the p7 digit of the m address, which is placed in the MQC during every staticize-instruction step. If the MQC digit is a 0, an ending pulse clears the static register and the next instruction is searched for. If the MQC digit is a 1, the static register is stepped to the SHL2 step of the 37 instruction.

During the SHL2 step, the normal recirculation gates of register A are blocked, and the contents of register A are shifted to the left by circulating the contents through a loop lengthened by one digit. The contents of register A are delayed one pulse time in the loop before they reach the left-shift gates, so that the entire contents of register A shift one place to the left each time the contents circulate. The binary bit of each digit is delayed by adding one pulse time in subregister 4 of register A. The quinary bits are delayed one pulse time as they pass through the S buffers and the complementing circuit before reentering register A. (The bits are only delayed in the complementing circuit; they are not complemented.) As a result of each left shift of one digit position, the MSD is shifted out and a 0 appears in the LSD position. During each shift operation, the MQC countdown gates count down by one until the MQC digit is 0. An MQC digit of 0 generates an ending pulse.

5-17. 61 (0-SUPPRESS) INSTRUCTION

The 62 instruction suppresses all 0's and commas to the left of the MSD in a USS-6 coded word. During a 62 (0-suppress) instruction, the USS-6 code combination for a 0 (00 0000) or a comma (11 0110) is changed to the USS-6 code combination for a space (00 0110). For printing, each word is stored after it is 0-suppressed, whereas in punching, each word is first 0-suppressed, then translated from USS-6 to card code and stored for punching. Information is in card code; it is printed in USS-6 code.

The word to be suppressed is stored in registers A and X, with four numeric bits of each digit 1 in register A, and the two zone bits in subregisters 1 and 2 of register X. Because subregister 3 of regiser X (X3) is not storing any part of the word, it is used as a temporary storage unit.

The two steps of the 0-suppress instruction, ZS1 and ZS2 are shown in figure 5-8. During ZS1 a 1 bit is placed in subregister X3 for every digit of the word. Two gates are made permissive by the code combinations for 0 and comma to change the 1 bits just inserted to 0 bits for each 0 or comma in the word. When the ZS1 step is completed, subregister X3 contains a pattern of 0 and 1 bits that correspond to the word being 0-suppressed. The static register is stepped to ZS2.

During the ZS2 step, the word in register A recirculates through the normal recirculation gates. The outputs of each of the ten bits in subregister X3 are available in parallel. These ten outputs are inputs to the 0-suppress

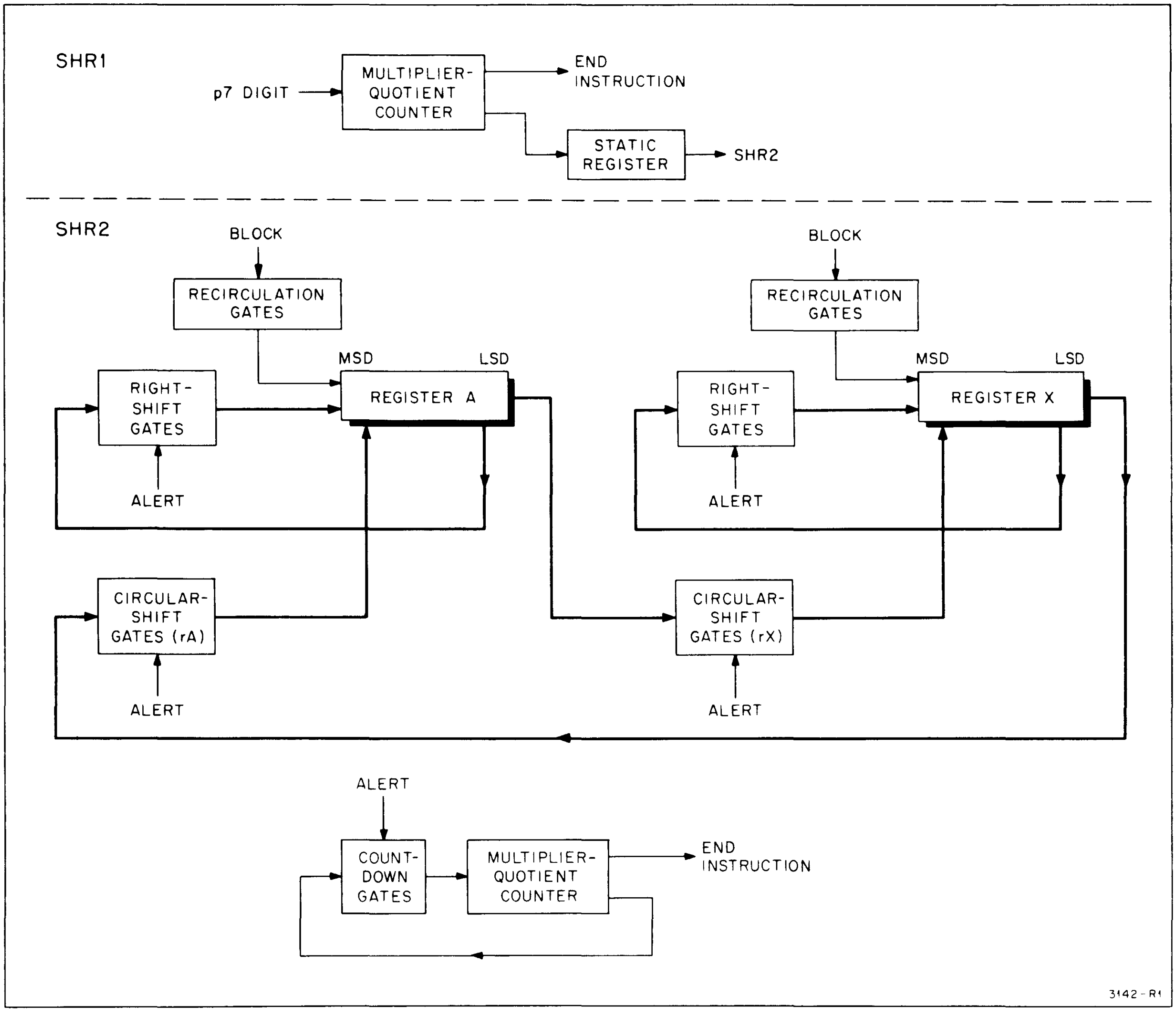


Figure 5-6. 32 (Right-Shift) Instruction

gate of register A. The recirculation gate of subregister X3 is blocked to clear the subregister as it recirculates. The output of the register-A 0-suppress gate is an input to the recirculation gates of register A and to the recirculation gates of subregisters X1 and X2. As long as a 1 bit is in subregister X3 (which means that significant digits still must enter the recirculation gates of register A and subregisters X1 and X2), the 0-suppress gate of register A is blocked so that the significant digits can reenter the registers. When binary 1's no longer are in subregister X3, the 0-suppress gate of register A is made permissive; its output blocks the normal recirculation gates of register A and the recirculation gates of subregisters X1 and X2 to replace the code for commas and 0's (nonsignificant) with 00 0000. Because the USS-6 space code is not 00 0000 but 00 0110, a second signal is generated from this 0-suppress gate (which inserts 1 bits into subregisters A2 and A3) to produce the correct space code to replace the nonsignificant 0's and commas. The instruction ends when the entire word has been 0-suppressed and a search initiated for the c address.

5-18. 67 (STOP) INSTRUCTION

The 67 instruction stops the Processor. When in the static register, the 67 instruction generates a signal that sets the stop flip-flop and clears the static register to 0's. The set stop flip-flop generates a signal that is sent to the instruction decoder to prevent any function signals from being generated.

With the static register cleared to 0's and no function signals being generated, the 67 instruction word (which includes the m and c address) continues to circulate in register C. The Processor is started again by pressing the COMPUTATION/RUN pushbutton on the operator's control panel. The next instruction is at the m address if the NEXT ADDRESS/m pushbutton is lit; it is at the

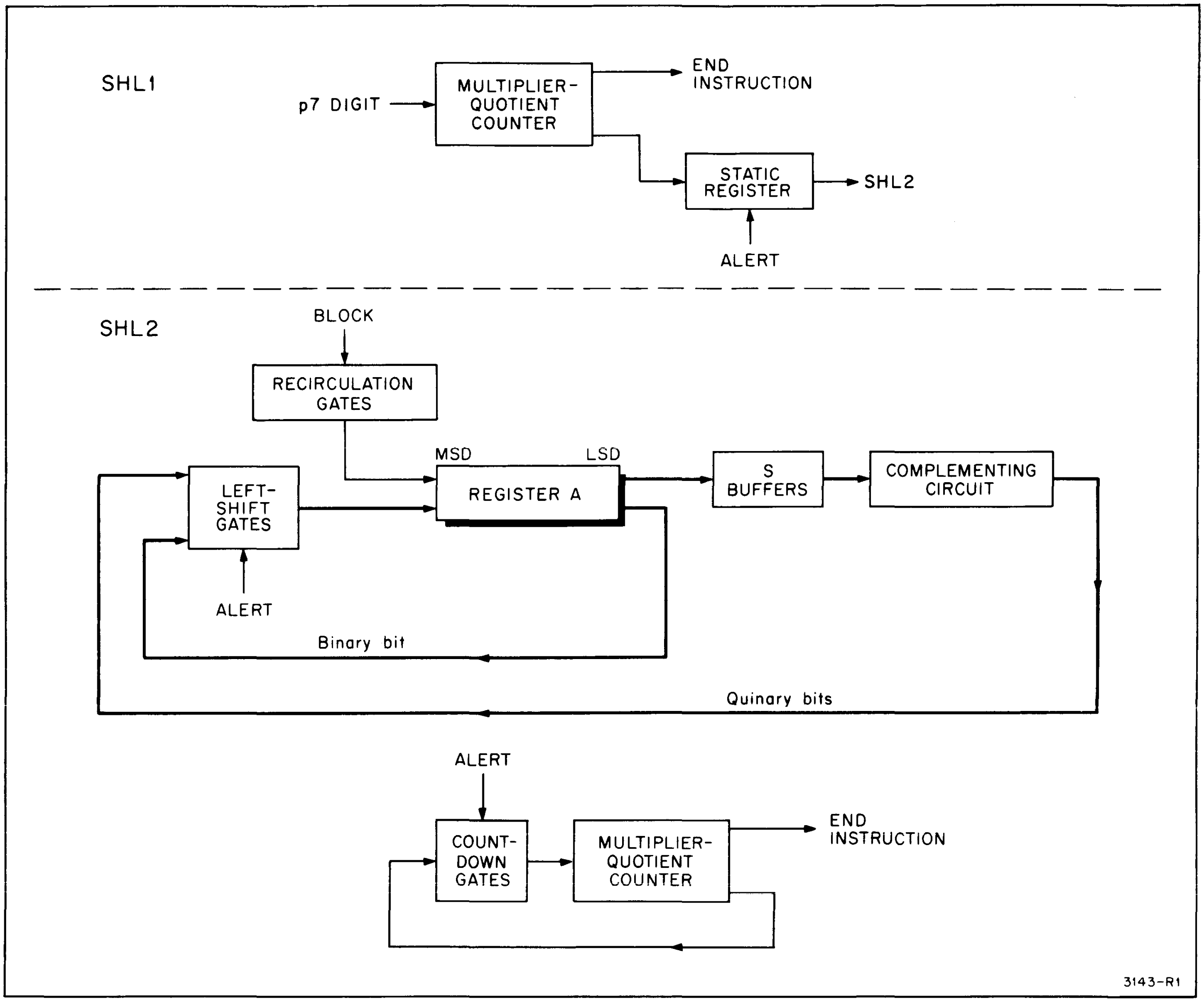


Figure 5-7. 37 (Left-Shift) Instruction

c address if the NEXT ADDRESS/c pushbutton is lit. However, the operator may select the next address manually by pressing either NEXT ADDRESS pushbutton. When pressed, these pushbuttons control the conditional-transfer (CT) flip-flop.

5-19. COMPARISON INSTRUCTIONS

Two comparison instructions compare the contents of registers A and L and, as a result of the comparison, direct the Processor either to the m or the c address. The 82 instruction compares for equality or inequality of the contents of registers A and L. The 87 instruction compares for greater-than and less-than conditions.

The Processor can be stopped after executing an 82 or 87 instruction by pressing the COMPARISON STOP pushbutton on the operator's control panel. When this pushbutton is pressed, one of the NEXT ADDRESS pushbuttons lights to indicate whether the next instruction is to be taken from the main-storage location designated by the m address or the c address. The address determined by the comparison can be changed by pressing the other NEXT ADDRESS pushbutton. The comparison-stop operation is described in detail under heading 4-63.

5-20. 82 INSTRUCTION

The 82 instruction compares the contents of register A [(rA)] with the contents of register L [(rL)]. If (rA) and (rL) are algebraically equal, the next instruction is in the storage location designated by the m address. If (rA) and (rL) are not algebraically equal, the next instruction is in the main-storage location designated by the c address.

Figure 5-9 is a block diagram of the 82 instruction. The state of the CT flip-flop determines which address is to be read from main storage. If the CT flip-flop is set, the m

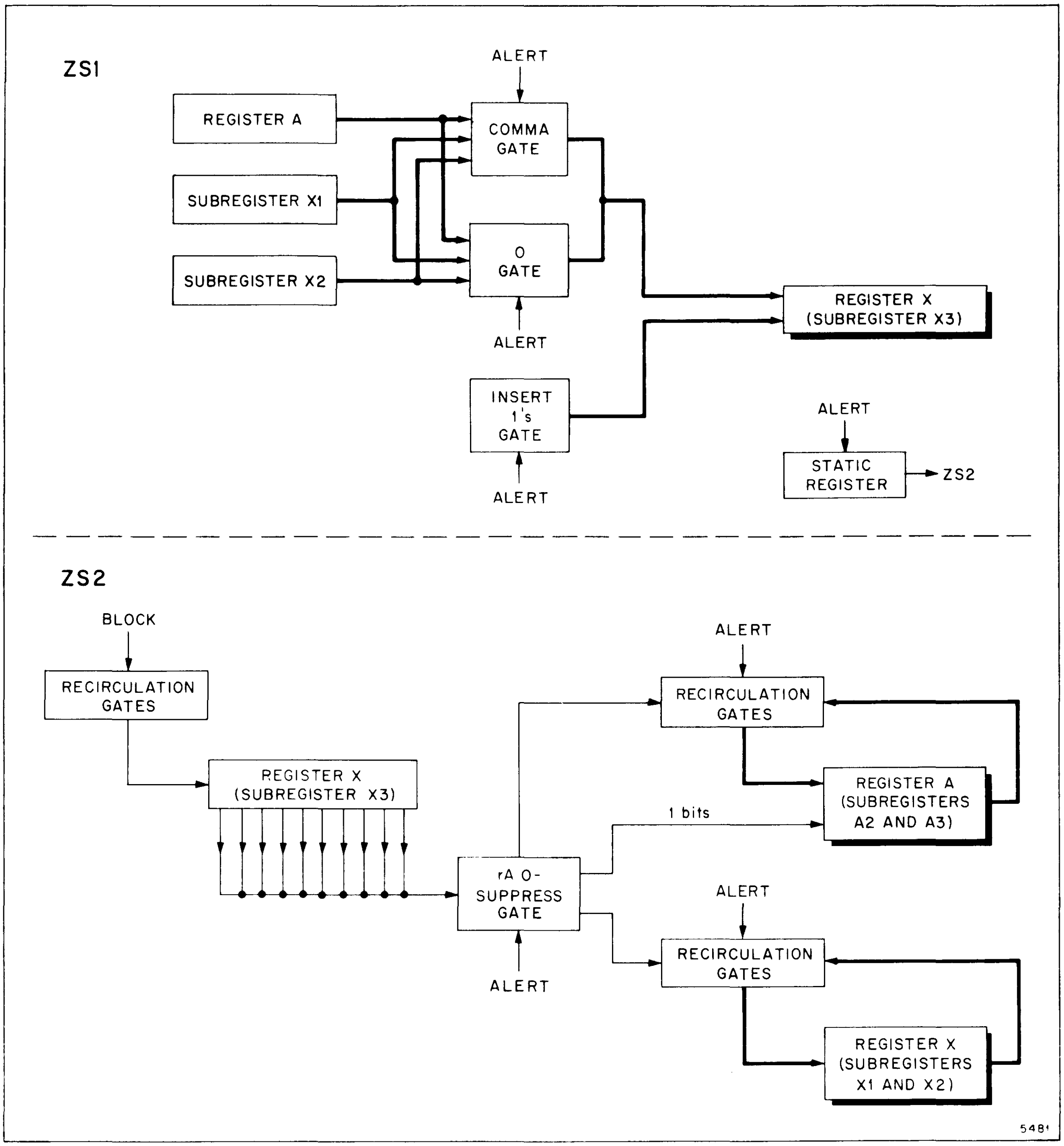


Figure 5-8. 62 (0-Suppress) Instruction

address is read from storage; if the CT flip-flop is restored, the c address is read. At the beginning of the instruction, the flip-flop is set. If (rA) and (rL) are equal, the CT flip-flop remains set and the m address is read from main storage; if (rA) and (rL) are not equal, the CT flip-flop is restored and the c address is read.

As shown in figure 5-9, (rA) and (rL) are sent to the binary- and quinary-equality gates by way of the S and M lines. The binary-equality gates sample the binary bits of each digit on the M and S lines for equality, while the quinary-equality gates sample the quinary bits of each digit on the S and M lines for equality. (A detailed description of the operation of these circuits is contained under headings 3-62 and 3-64.)

In addition to the S and M lines, the CP signal from the set complement flip-flop is an input to the binary- and quinary-equality circuits of the comparator. The complement flip-flop is initially set at the beginning of the comparison to alert the binary- and quinary-equality gates during the word time that (rA) and (rL) are compared.

Signals A and C also are inputs to the quinary-equality gates. When the complement flip-flop is set at the begin-

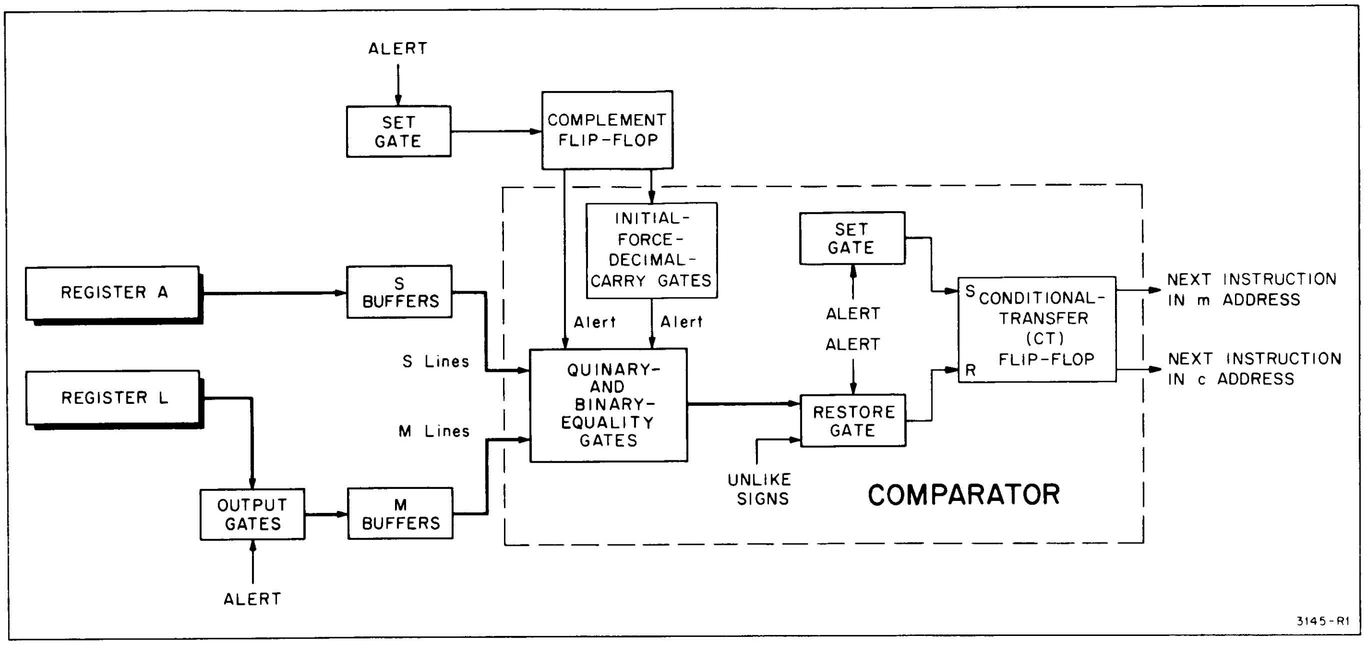


Figure 5-9. 82 (Equality-Comparison) Instruction

ning of the comparison, a CP5 signal is generated for one pulse time. This signal goes to the initial-force-decimal-carry gates, which cause A and C to be initially generated so that the quinary-equality gates are permissive to the bits of the first four digits to be compared for equality on the M and S lines. For example, suppose the following words are compared:

(rL) on M lines	00000 06917
(rA) on S line	00000 06917

The initial generation of the A and C signals by the initial-force-decimal-carry gates make the quinary-equality gates permissive to the 7's, which are the first two digits to be compared. Because they are equal, A and C signals are generated again. Because all succeeding digits are equal, A and C continue to be generated until the end of the comparison.

Four gates can restore the CT flip-flop during the 82 instruction. The $\overline{EQ}$ and $\overline{A'}$ output signals of the binary- and quinary-equality circuits are sent to either of the four restore gates to restore the CT flip-flop if either the binary or quinary part is not equal. During this instruction, the outputs of the rA- and rL-sign flip-flops are inputs to the other two restore gates to restore the CT flip-flop if the signs of (rL) and (rA) are not the same.

5-21. 87 INSTRUCTION

The 87 instruction compares (rA) and (rL). If (rA) is algebraically greater than (rL), the next instruction is in the storage location designated by the m address. If (rA) is algebraically less than or equal to (rL), the next instruction is in the storage location designated by the c address.

As shown in figure 5-10, during an 87 instruction (rA) and (rL) are sent to the comparator by way of the S and M lines. Both lines are inputs to the quinary-equality circuit, the quinary- and binary-equality gates, and the initial-force-decimal-carry gates. At the beginning of the comparison, the complement flip-flop is set to generate CP, which alerts the quinary-equality circuit and some of the equality and decimal-carry gates to information on the S and M lines. During the comparison, the complement flip-flop may be either set (CP) or restored ($\overline{CP}$) to alert different gates of these circuits.

not tru f.f.

The alerted quinary- and binary-equality gates sample the digits on the S and M lines as described under headings 2-20, 3-61 and 3-64.

When the sign of (rA) is plus, A and C signals are initially generated by the quinary- and binary-carry circuits (through the intial-force-decimal-carry gates) to operate a restore gate of the CT flip-flop, the state of which determines whether the Processor will take the next instruction from the m address or the c address (figure 5-10). After they are initially generated, the A and C signals also go to the quinary-carry circuit and the quinary-equality gates. When the sign of (rA) is minus, no A and C signals are generated, and the CT flip-flop remains set. (The operation of the comparator is described in detail under heading 3-60.)

Table 5-1 lists the sign and value conditions under which the CT flip-flop is set or restored. Two conditions restore the flip-flop when the sign of (rA) is plus: (rA) equals (rL), or (rA) is less than (rL). If (rA) equals (rL), the binary- and quinary-equality gates continue to generate A and C.

If (rA) is less than (rL), A and C are again present, but they are generated by the quinary-carry gates that indicate (rA) is less than (rL). (Refer to heading 3-62.) For

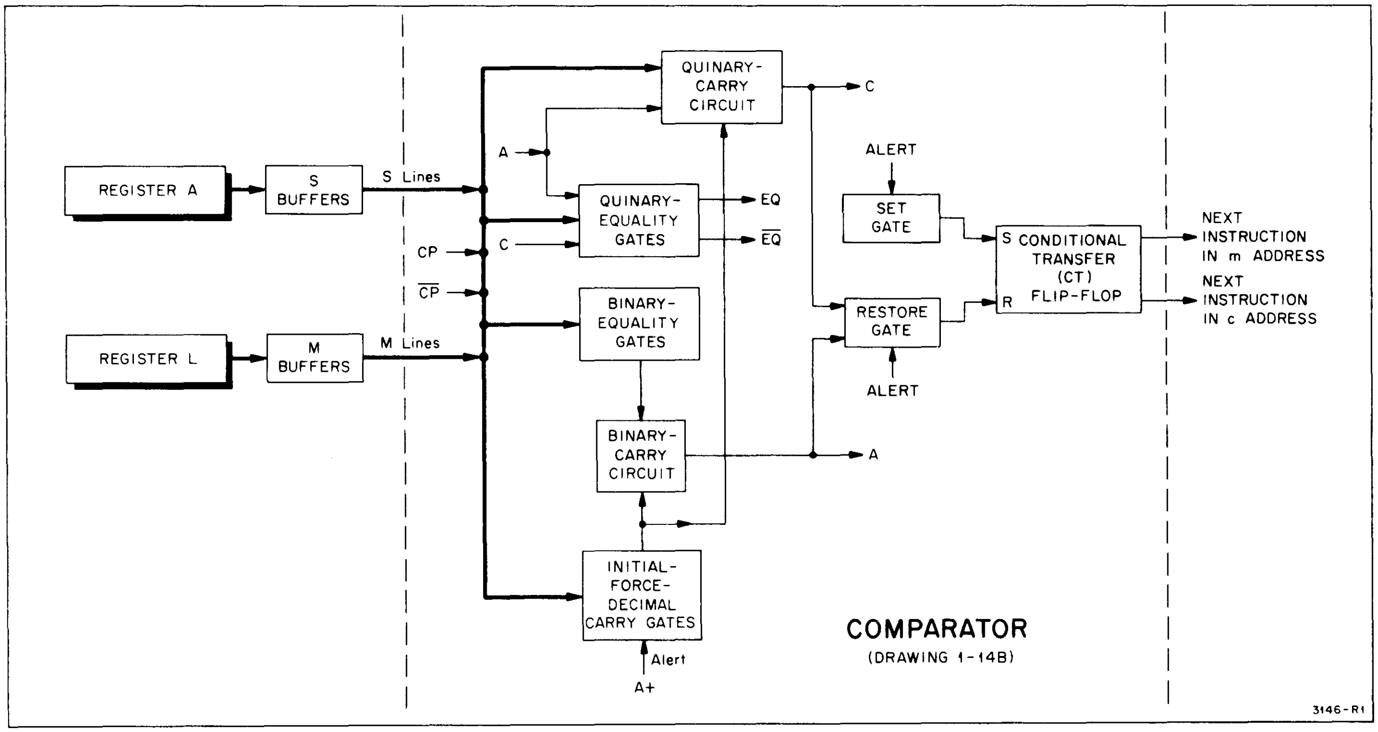


Figure 5-10. 87 (Greater-Than-Comparison) Instruction

example, suppose the two words being compared are the following:

(rL) on M lines 00000 07809

(rA) on S lines 00000 07431

When the 9 and 1 are compared, A and C signals are generated because (rA) is less than (rL). When the 0 and 3 are compared, A and C signals no longer are generated because (rA) is greater than (rL), and there are no gates that generate A and C under this condition. When the two most significant nonequal digits, 8 and 4, are compared, A and C are generated again by the gates of the quinary-carry circuit that indicate (rA) is less than (rL). The A and C signals thus generated alert the quinary-equality gates so that the succeeding digits, which are all equal, continue to generate A and C signals.

Table 5-1. State of Conditional-Transfer Flip-Flop After Sign and Value Comparisons of (rA) and (rL)

Sign		Value		
(rA)	(rL)	(rA) > (rL)	(rA) = (rL)	(rA) < (rL)
+	+	Set	Restored	Restored
+	–	Set	Set	Set
–	+	Restored	Restored	Restored
–	–	Restored	Restored	Set

One condition keeps the CT flip-flop restored when the sign of (rA) is minus: the signs of both (rA) and (rL) are minus, and (rA) is less than (rL). When (rA) is less than (rL), the quinary-carry gates that indicate (rA) is less than (rL) generate A and C, and the CT flip-flop remains set. For example, suppose the two words being compared are the following:

(rL) on M lines 00000 05924

(rA) on S lines 00000 05384

The A and C signals are not generated by the appropriate quinary-carry gates until the two most significant nonequal digits, 9 and 3, are compared. Because A and C are not initially generated, the equality gates are blocked and A and C are not generated when the two 4's are compared. They are not generated when the 8 and 2 are compared because there are no gates that indicate A and C when (rA) is greater than (rL). However, when the 9 and 3 are compared, A and C are generated to make the binary-equality gates permissive. The signals continue to be generated when the remaining digits are compared because they are all equal.

5-22. TEST INSTRUCTIONS

Three input-output test instructions (22, 27, and 42) are provided, one for each of the three input-output units. These instructions are used to determine whether a particular input-output unit is available or whether it is still processing a previous instruction. For example, the input-output test for the printer tests to determine whether the printer is free to accept a print order or is still processing a previous print order. If the printer is not available, the Processor continues processing the main program instead of waiting until the printer is available. The instructions are distributed throughout the program to test the availability of the input-output units at convenient intervals.

Each of the three input-output instructions has two stages. The first stage tests whether the unit is available. If it is not available, an ending pulse is generated and the Processor goes to the c address for the next instruction. Because each of the three instructions tests a different unit, the first stage of each instruction is different.

If the test performed during the first stage shows that the unit is available, the static register is stepped to the second stage. The second stage of each test instruction is identical; it is referred to as a common test. (See figure 5-11.) During this stage, the contents of register C are transferred to register A, and the next instruction is read from the m address; this is done because the Processor must return to it in order to continue the program. The m address to which the Processor is directed may contain the beginning instruction of a subroutine which transfers the c address of the test instruction from register A to storage, gives one of the input-output instructions, and returns the Processor to the main program.

5-23. 22 INSTRUCTION

The 22 instruction tests to determine whether the read-punch buffer of the card-buffer area is loaded with information from the read-punch unit. If the buffer is not loaded, the next instruction is in the storage location designated by the c address. If the buffer is loaded, the next instruction is in the storage location designated by the m address, and the contents of register C are transferred to register A.

The instruction analysis for the 22 instruction lists the signals in the first stage, during which the test is made to determine whether the read-punch buffer is loaded. If the buffer is not loaded, an ending pulse is generated. If the buffer is loaded, the static register is stepped to the second stage of the instruction.

5-24. 27 INSTRUCTION

The 27 instruction tests to determine whether the printer is available. If the printer is not available, the next instruction is taken from the storage location designated by the c address. If the printer is available, the next instruction is taken from the storage location designated by the m address, and the contents of register C are transferred to register A.

The instruction analysis of the 27 instruction (manual 5, table 1-15) lists the signals in the first stage, during which the test is made to determine whether or not a print or paper-feed operation is taking place. If the printer is not available, the static register is stepped to the second stage of the instruction.

5-25. 42 INSTRUCTION

The 42 instruction tests the card-reader buffer of the card-buffer area to determine whether it is loaded with information read by the card reader. If the card-reader buffer is not loaded, the next instruction is taken from the storage location designated by the c address. If the buffer is loaded, the next instruction is taken from the storage location designated by the m address, and the contents of register C are transferred to register A.

The instruction analysis of the 42 instruction (manual 5, table 1-20) lists the signals in the first stage, during which the test is made to determine whether or not the card-reader buffer is loaded. If the buffer is not loaded, an ending pulse is generated. If the buffer is loaded, the static register is stepped to the second stage of the instruction.

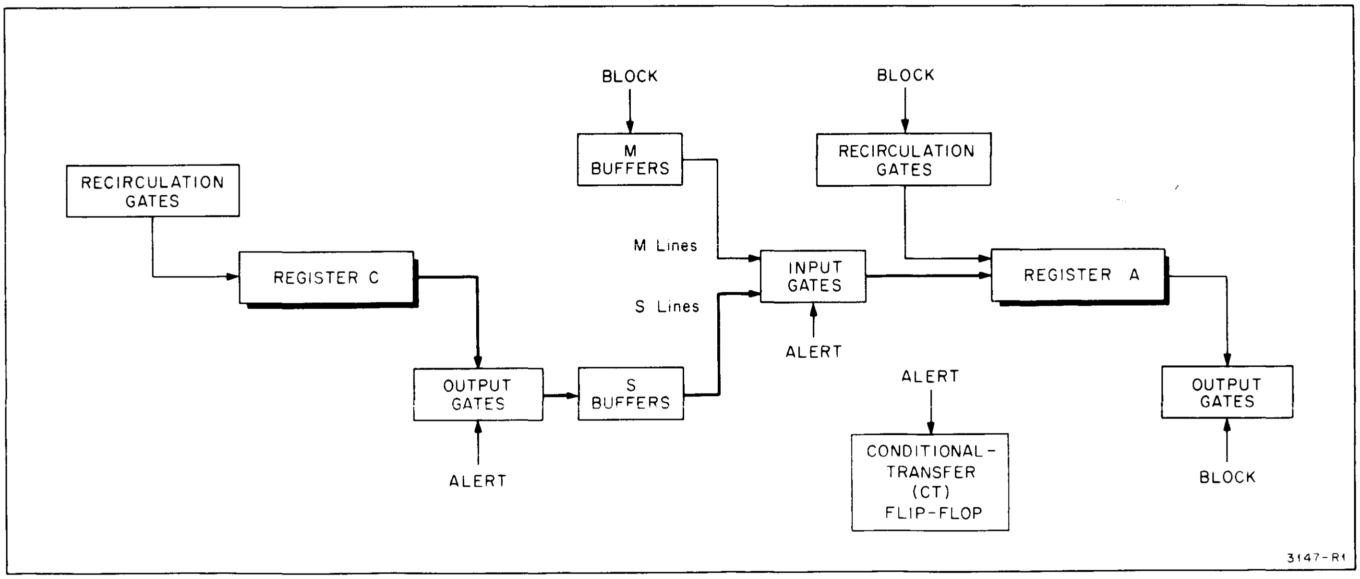


Figure 5-11. 22, 27, and 42 (Test) Instructions, Second Stage

5-26. OPTIONAL INSTRUCTIONS

If index registers are included in the system, 'he 02 and 07 instructions are provided to initially load the index registers.

5-27. 02 INSTRUCTION

The 02 instruction transfers the four digits of the m address from register C into the specified index register. During the staticize step of the instruction, the instruction code (02) is stored in the static register and the entire instruction word transferred to register C. During the same step, the sign and the third bit of the p9 digit are stored in the rC-sign flip-flop and the STR3 flip-flop, and the index-register read-in flip-flop is set. (See figure 5-12.)

During the execution step of the 02 instruction, the index-register read-in flip-flop generates signals that block the recirculation gates of the selected index register to clear the previous contents and alert the input gates of the same register to the four digits of the m address from register C. The contents of the m address from register C are ungated outputs and, because register C continues to recirculate, the contents of register C, including the m address, remain unchanged. The p5 through p8 digits from register C enter the selected register, where they are stored until used to modify an operand.

5-28. 07 INSTRUCTION

The 07 instruction adds the four m-address digits from register C to the four digits from the selected index register and stores the sum in the m-address positions (p5 through p8) of register C during a word time added to the staticize step. This part of the instruction is identical to the index-register modification described under heading 4-13 and shown in figure 4-6. The instruction decoder is blocked during this word time.

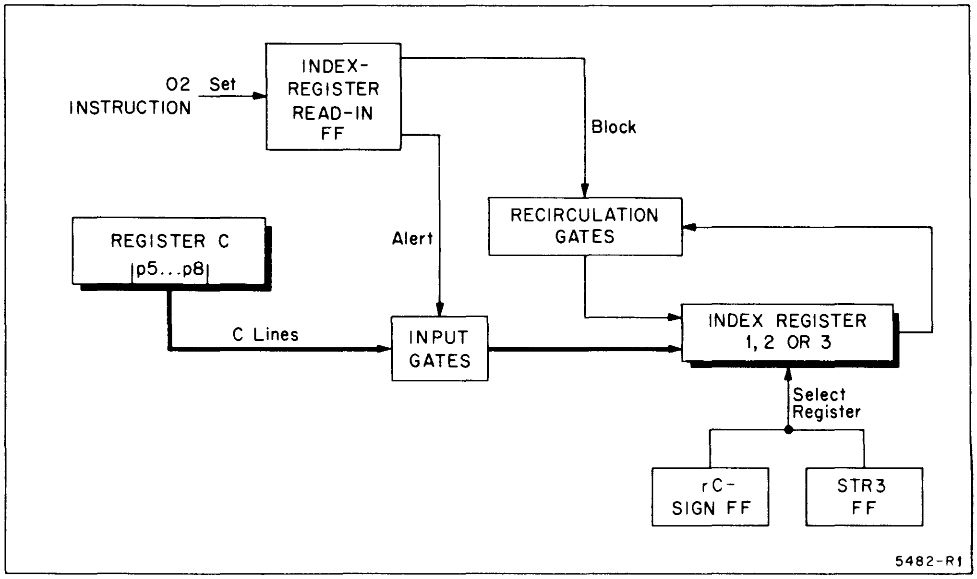


Figure 5-12. 02 (Index-Register-Load) Instruction

When the instruction decoder no longer is blocked, it generates a signal that sets the index-register read-in flip-flop. The set outputs of this flip-flop block the recirculation gates of the selected index register and alert the input gates to the m-address digits from register C. (See figure 5-13.) The instruction decoder and function encoder generate other function signals to transfer the digits from register C onto the S lines and into the input gates of register A. The function signals also block the recirculation gates of register A, clear the S lines, and block the M lines so that the digits from register C can be transferred into the p5 through p8 positions of register A. The rA-sign flip-flop then is set to generate A+.

The instruction ends with the sum stored in registers C and A and in the index register. The sum is stored in register A so that the contents of that register can be compared with a constant in register L.

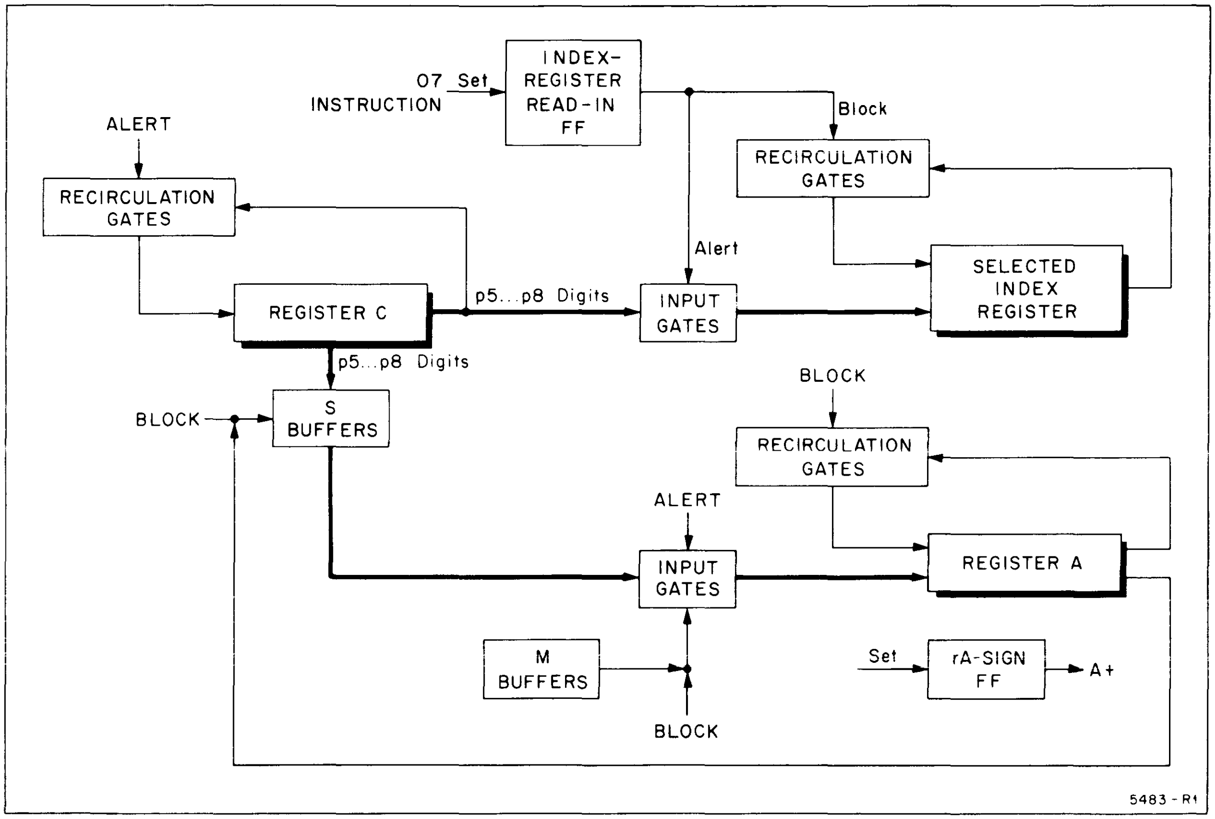


Figure 5-13. 07 (Index-Register-Add) Instruction, Execution Step

To Holders of USSC Manual 1, Processor, Basic System:

Enclosed is section 6 of part 1 of manual 1. Please insert these pages in the appropriate place.

Please change the titles of the figures on pages 6-25 through 6-31 to the following:

Page	Title
6-25	Figure 6-21. Clock System, Schematic, Part 1 of 2
6-27	Figure 6-21. Clock System, Schematic, Part 2 of 2
6-29	Figure 6-22. Clock Variations for Processor With Core Storage, Part 1 of 2
6-31	Figure 6-22. Clock Variations for Processor With Core Storage, Part 2 of 2

Thank you.

J. Mulligan

CONTENTS

Heading **Title** **Page**

SECTION 6. ELECTRONIC CIRCUITRY

ILLUSTRATIONS

Figure **Title** **Page**

Section 6

ELECTRONIC CIRCUITRY

6-1. INTRODUCTION

The electronic circuitry consists primarily of solid-state components such as semi-conductor diodes, transistors, and magnetic amplifiers. The use of vacuum tubes is limited to the minimum required for power and clock circuitry.

The solid-state components are packaged on printed circuit cards. These cards are combined into functional groups (logical or electronic) and assigned specific identities and purposes. The individual circuits and several of the functional groups into which these circuits are combined are described in this section.

For the following circuit discussions, current flow is assumed from negative to positive.

6-2. GENERAL

The circuits comprise the power system and the logical system. The power system is subdivided into the clock, power control, and power supplies. The clock supplies power for the magnetic amplifiers and synchronizes the computer by way of synchronizing pulses from the drum sprocket track.

The logical system is subdivided into logic and storage-control circuits. The logic circuits consist almost wholly of magnetic amplifiers, in slightly varying configurations, that are combined to form the logical elements (gates, buffers, flip-flops, and registers). The storage-control circuits are special-purpose combinations of diodes, magnetic amplifiers, and transistors that control the drum-storage functions of read, write, and head selection.

6-3. SYSTEM WITH INCREASED STORAGE

The USSC drum in the expanded 9200-word system stores 4200 additional words. Consequently, several circuits are added to the storage-control circuitry to permit system operation at increased capacity. These circuits are unique to the expanded storage system and are discussed under heading 6-31.

6-4. SIGNAL CONVENTIONS

System information is expressed as combinations of 0 and 1 bits, represented by low or high signals. Figure 6-1 shows typical computer information in which all pulses are either at the zero-volt reference level or at some positive potential. The positive alternation of a sine wave is designated logical 0: signals at the zero-volt reference level are designated logical 1's. Figure 6-1 illustrates +10-volt blocking pulses with each +18-volt signal. The blocking pulses are associated solely with the electronic operation of the magnetic amplifiers; they have no logical significance.

6-5. RECURRING CIRCUIT TYPES

Two basic circuit types comprise the logic configurations. These include diode circuits for gating and buffing, and complementing and noncomplementing magnetic amplifiers. Magnetic amplifier-diode circuit combinations form all of the higher order logic configurations (flip-flops, registers, matrixes, and others).

The material contained under headings 6-6 through 6-16 discuss typical diode and magnetic-amplifier circuit packages. Although the application of these two circuit

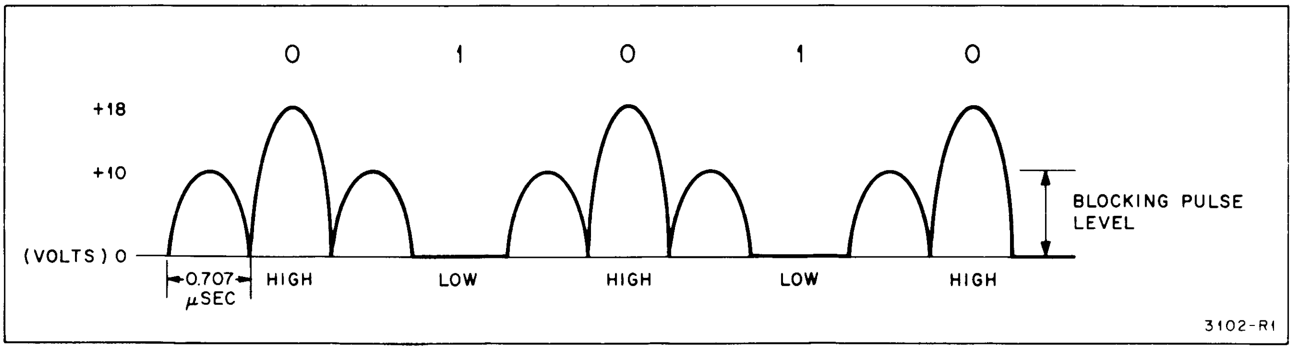
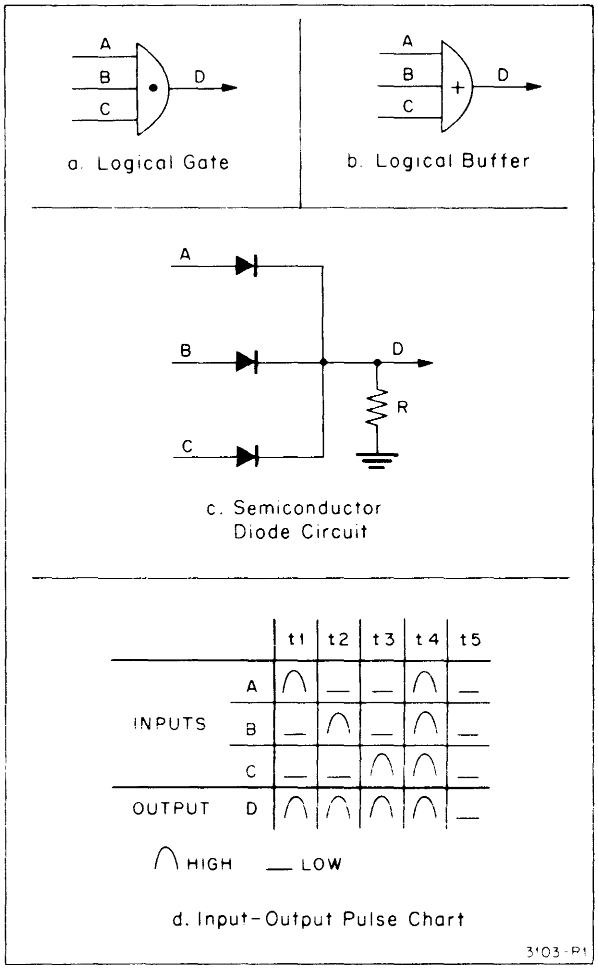


Figure 6-1. Electrical Characteristics of Information Pulses

packages may vary within the system, the theory of operation for each variation is essentially similar to the typical circuit package analyses. Accompanying drawings and wave shapes are idealized and should not be referenced for maintenance.

6-6. GATES AND BUFFERS

In the diode configuration shown in figure 6-2c, the positive-going pulse (high) applied at A causes the diode to conduct; thus, the entire pulse voltage is developed across resistor R. Point D goes positive and the output is a high. Similarly, if a high is applied at B or C, a high develops at D. If no voltage is applied to A, B, or C (all inputs low), none of the diodes conduct and no voltage is developed across resistor R. Thus, point D remains at ground potential and the output is a low.



		t1	t2	t3	t4	t5
INPUTS	A	⋂	—	—	⋂	—
	B	—	⋂	—	⋂	—
	C	—	—	⋂	⋂	—
OUTPUT	D	⋂	⋂	⋂	⋂	—

⋂ HIGH — LOW

Figure 6-2. Gate and Buffer Circuits

Figure 6-2d shows the output at D for several combinations of inputs at A, B, and C. If lows are present at A, B, and C the output at D is a low. This diode configuration functions as a logical gate. If a high is present at A, B, or C, a high is present at D. This diode configuration functions as a logical buffer. There is no physical or electronic difference between a gate and a buffer. The terms gate and buffer designate the logical purpose of the circuit.

6-7. MAGNETIC AMPLIFIERS

The magnetic amplifier is a transformer in the form of a ring-type saturable core (toroid) with two or more separate windings. Amplification is achieved when a large current through one winding is controlled by the effect on the core of a smaller current through another winding. The magnetic amplifiers used in this system produce an output current that is greater than the input current. At 707 kilocycles, the signal frequency used in the computer, the standard magnetic amplifier has a power gain of approximately 7.

The magnetic amplifiers are of two basic configurations —complementing and noncomplementing. A complementing magnetic amplifier logically inverts the input signal: thus a low (0-volt) input results in a high (+18-volt) output, and a high input results in a low output. The noncomplementing magnetic amplifier produces an output signal which is logically identical to the input signal. Through both types of magnetic amplifiers, the output signals lag the input signals by 180 degrees. Thus, at a frequency of 707 kilocycles there is a pulse-time delay of 0.707 microseconds through each amplifier.

The operation cycle of a magnetic amplifier is divided into two parts of equal duration. During the first half of the cycle, an input signal is applied to the primary winding. The input signal sets the core so that the secondary winding presents either a high or a low impedance to any signal passed through the secondary. During the second half of the cycle a pulse is applied to the secondary or output winding and, depending upon the core setting established during the first half cycle, a large or small current flows through the winding. The magnitude of the current determines the output of the magnetic amplifier.

6-8. HYSTERESIS LOOPS. When a changing current is passed through a coil wound around a ferro-magnetic material (core), magnetic flux density increases through the core until saturation is reached. If the current is further increased, flux density does not increase and the inductive reactance of the coil becomes effectively zero for further current changes. Thus the coil presents only a slight resistive impedance to current flow beyond saturation.

When current is removed from the coil, the core maintains a residual flux density (remanence) which is slightly below saturation. If a changing current is passed through the coil in the opposite direction, the coil opposes current flow until the current becomes great enough to overcome the remanence of the core. New flux lines are then developed until the core is saturated in the opposite direction.

Figure 6-3 shows an idealized hysteresis loop. Flux density is plotted along the vertical line (–B to +B), and current is plotted along the horizontal line (–H to +H).

For current change in a positive direction, flux density increases to $+B_s$ (saturation) and remains there for any further positive current changes. When current is removed, flux density settles to approximately $+B_r$ (posi-

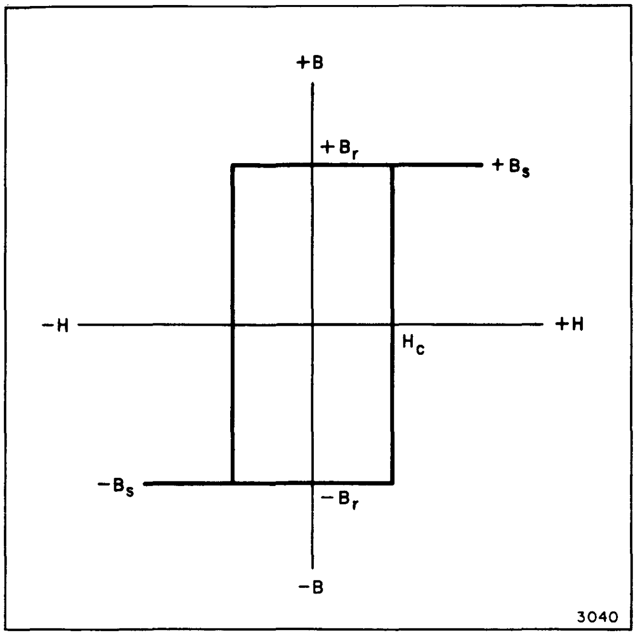


Figure 6-3. Idealized Hysteresis Loop

tive remanence). Current flow in a negative direction reverses the direction of the flux lines until negative saturation ($-B_s$) is reached. When current is removed, the flux density is determined by the negative remanence ($-B_r$) of the core.

6-9. CORE SWITCHING CHARACTERISTICS. The core is useful in computer applications because the impedance of the coil wound around a saturated core is very high for a current pulse opposite in polarity to the core remanence ($-B_r$ or $+B_r$); the core must be switched to its opposite saturation polarity before current can begin to flow through the coil. However, the coil presents a very low impedance to current in the same direction as the core remanence. For example, positive current through a core at $+B_r$ would encounter an impedance only until the flux density is increased from $+B_r$ to $+B_s$. Beyond $+B_s$ the impedance of the coil becomes negligible for further positive current.

Thus, if pulse duration and amplitude are only sufficient to switch the core, the pulse energy is transferred to the magnetic field of the core, and the pulse is effectively blocked from succeeding circuits. Conversely, a pulse that does not have to switch the core is only slightly attenuated through the coil.

6-10. POWER AMPLIFICATION. Power amplification by the use of saturable reactors is achieved with two coils wound around a single core. Current through either coil will saturate the core. Thus, if a relatively small current through one coil changes the state of the core, a large current in the same direction may pass through the second coil, but current in the opposite direction through the second coil is blocked. Therefore, power amplification may be obtained through a magnetic amplifier just as it may be obtained through a relay; the presence or absence of a small control current through one part of the device determines whether a larger current will flow through another part of the device. The power gain through the magnetic amplifiers in the USSC system is measured in terms of drive units (4.3 milliamperes). Approximately five drive units are required to drive a magnetic amplifier, and 35 drive units are produced at its output.

6-11. COMPLEMENTING MAGNETIC AMPLIFIER. The complementing magnetic amplifier (complementer) produces an output that is always the complement of the input, but which lags the input by one-half pulse time. Figure 6-4b shows typical input and output waveforms.

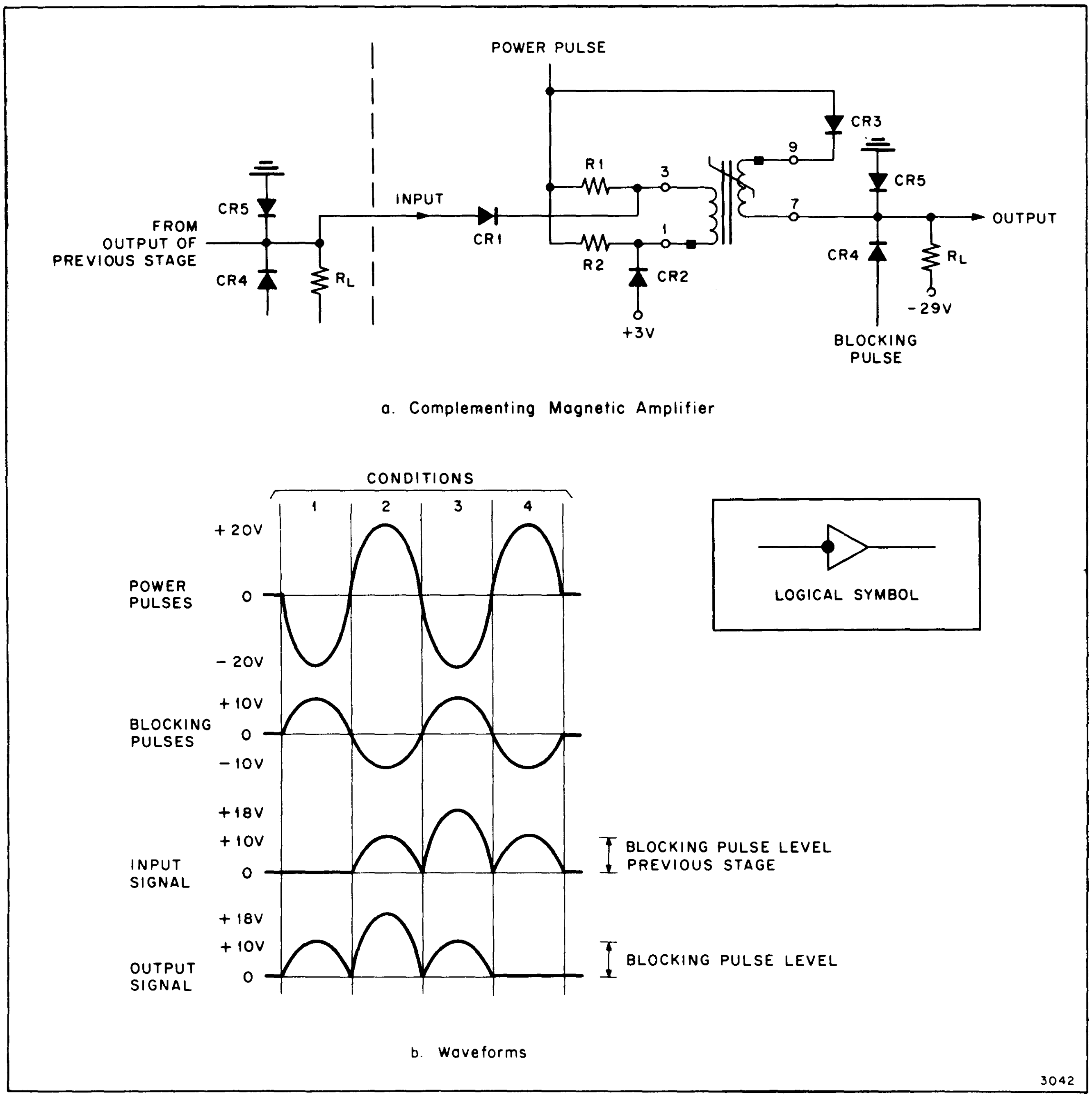
Because the power and blocking pulses are supplied by the clock (heading 6-34), the 180-degree phase difference between the pulses remains fixed. Positive input signals are applied only when the power pulse is negative and the blocking pulse positive. The square polarity symbols shown on the core windings (figure 6-4a) indicate that the signals at the dotted ends of the two coils are always of the same polarity. When a signal is applied to either winding, a voltage is induced in the other winding that causes the dotted end of the second coil to assume the same polarity as the dotted end of the first coil.

The operation of the complementing magnetic amplifier is shown as four separate circuit conditions (figure 6-4b). Each condition is a result of a discrete combination of an input signal, a power pulse, and a blocking pulse.

Condition 1. The signal combination for condition 1 is a 0-volt input signal, a negative power pulse, and a positive blocking pulse. Terminal 3 of the primary winding (figure 6-4a) is clamped to ground through input diode CR1 and diode CR5 of the previous stage. Terminal 1 of the primary winding is clamped to +3 volts through CR2. Current flows from the negative clock source through R1, from terminal 3 to terminal 1 through the primary winding, and through CR2 to +3 volts. Regardless of the core setting before the power pulse, this current flow drives the core to $+B_s$ (figure 6-3).

Condition 2. During condition 2 the power pulse is positive, and the blocking pulse is negative. The positive power pulse reverse-biases diodes CR1 and CR2. The primary winding is thus clamped at the power-pulse potential, and no current flows through the input coil. The positive power pulse forward-biases CR3, and current flows from the -29-volt supply, through R_L, the secondary winding, and CR3 to the positive clock source, and drives the core to $+B_s$. Since the core is already reset to $+B_r$, the secondary winding presents a very low impedance to the positive power pulse. Hence, practically the entire power-pulse voltage is developed across R_L, and this voltage becomes the circuit output. The negative blocking pulse reverse-biases CR4; thus the output circuit consists solely of CR3, the magnetic amplifier secondary, and R_L. This satisfies the first requirement of the complementer; that is, a high output is developed for a low input signal.

Condition 3. During condition 3 the power pulse is negative, the blocking pulse is positive, and a positive signal input is applied to the circuit. The negative power pulse



a. Complementing Magnetic Amplifier

b. Waveforms

Figure 6-4. Complementing Magnetic Amplifier

clamps terminal 1 of the primary winding to +3 volts through CR2. The positive input signal raises the potential at terminal 3 of the primary winding above +3 volts, providing a circuit for current from the negative clock source through R2, from terminal 1 to terminal 3, through the primary winding, and through CR1 to the positive input potential. This current sets the core to $-B_s$ and cause terminal 1 of the primary winding to go negative with respect to terminal 3. A negative potential is induced at terminal 9, but the positive blocking pulse forward-biases CR4, thus counteracting the effect of the negative induced voltage. Diode CR3 is kept reverse-biased, and spurious output pulses are prevented.

Condition 4. During condition 4 the power pulse is positive and the blocking pulse is negative. The input signal is a +10-volt blocking pulse from the previous stage. The positive power pulse reverse-biases CR1 and CR2, clamping the primary winding at the power-pulse potential, and no current flows through the primary winding. The power pulse forward-biases CR3 and current flows from the −29-volt supply, through R_L, the secondary winding, and CR3 to the positive clock source. The current flow switches the core from $-B_r$ to $+B_r$. During switching, the core secondary presents a very high impedance to the power pulse. Therefore, the current flow in the output circuit is very small and the output is 0 volts.

This satisfies the second requirement of the complementer; that is, a low output is developed for a high input signal.

To prevent the switching current through the secondary winding from causing a spurious output across R_L when the power pulse is positive, terminal 7 is clamped to ground potential through CR5. If a positive power pulse occurs and the secondary winding presents a high impedance, a small current flows through the secondary winding. The current through CR5, however, decreases by the same amount, and the net current change through R_L is zero. If the secondary winding presents a low impedance, the large increase in current caused by a positive power pulse cannot be neutralized and an output develops.

Resistors R1 and R2 are current limiters. If the signals described in condition 3 are present, R1 limits the current flow from the negative clock source through R1 and CR1 to the positive input potential. If the primary winding is shifted to a low-impedance state prior to the completion of the input and power pulses, R2 limits the current flow from the negative clock source through R2, the primary winding, and CR1 to the positive input. This prevents the output of the previous amplifier from being short-circuited. If no input is provided, the current from the negative clock is limited through R1, the primary winding, and CR2 to +3 volts.

The power pulse duration is only long enough to switch the core. Since the core offers a high impedance during switching and a low impedance when saturation is reached, an extraneous output signal develops if the power pulse is of longer duration than the switching time.

6-12. NONCOMPLEMENTING MAGNETIC AMPLIFIER. The noncomplementing magnetic amplifier (noncomplementer) produces a low output for a low input signal, and a high output for a high input signal (figure 6-5b).

The noncomplementer is very similar to the complementer. The two basic changes are the addition of a bias winding (figure 6-5a), and the use of the blocking pulse potentials in place of the fixed +3-volt potential to the primary. A current through the bias winding is not sufficient to switch the core, but it does either inhibit or add to the effect of the input signal through the primary winding.

Assume the signal configuration of condition 1 (figure 6-5b). The core is at $+B_r$ and presents a high initial impedance to a negative switching current. Current flows between the negative power pulse source and the positive blocking pulse source through the primary winding. However, the initial impedance of the primary is high, causing the potential at terminal 3 to go negative. Diode CR1 conducts, and terminal 3 is clamped to ground potential through CR5. Simultaneously, current flow through the bias winding augments the switching effect of the primary winding current. The large negative current through the bias and primary windings causes a similarly large negative voltage to be induced into the secondary winding. This voltage, if unchecked, would forward-bias CR3 and thus permit part of the power pulse to flow through the secondary, thereby inhibiting the effect of the switching current. This induced voltage is cancelled by the blocking pulse through CR4.

For condition 2, the core is at $-B_r$, the power pulse is positive, the blocking pulse is negative, and the input is at the +10-volt blocking pulse level. Diodes CR1 and CR2 are reverse-biased and no current flows through the primary winding.

Current produced by the power pulse flows through the bias winding to ground, and through the secondary winding by way of CR3 to –29 volts. The secondary current switches the core from $-B_r$ to $+B_s$, but to prevent any part of the power pulse from appearing at the circuit output, the bias winding induces a current into the secondary which opposes the switching current. The effect of the bias winding current is to limit the switching excursion to $+B_r$; thus the secondary impedance remains high for the duration of the power pulse.

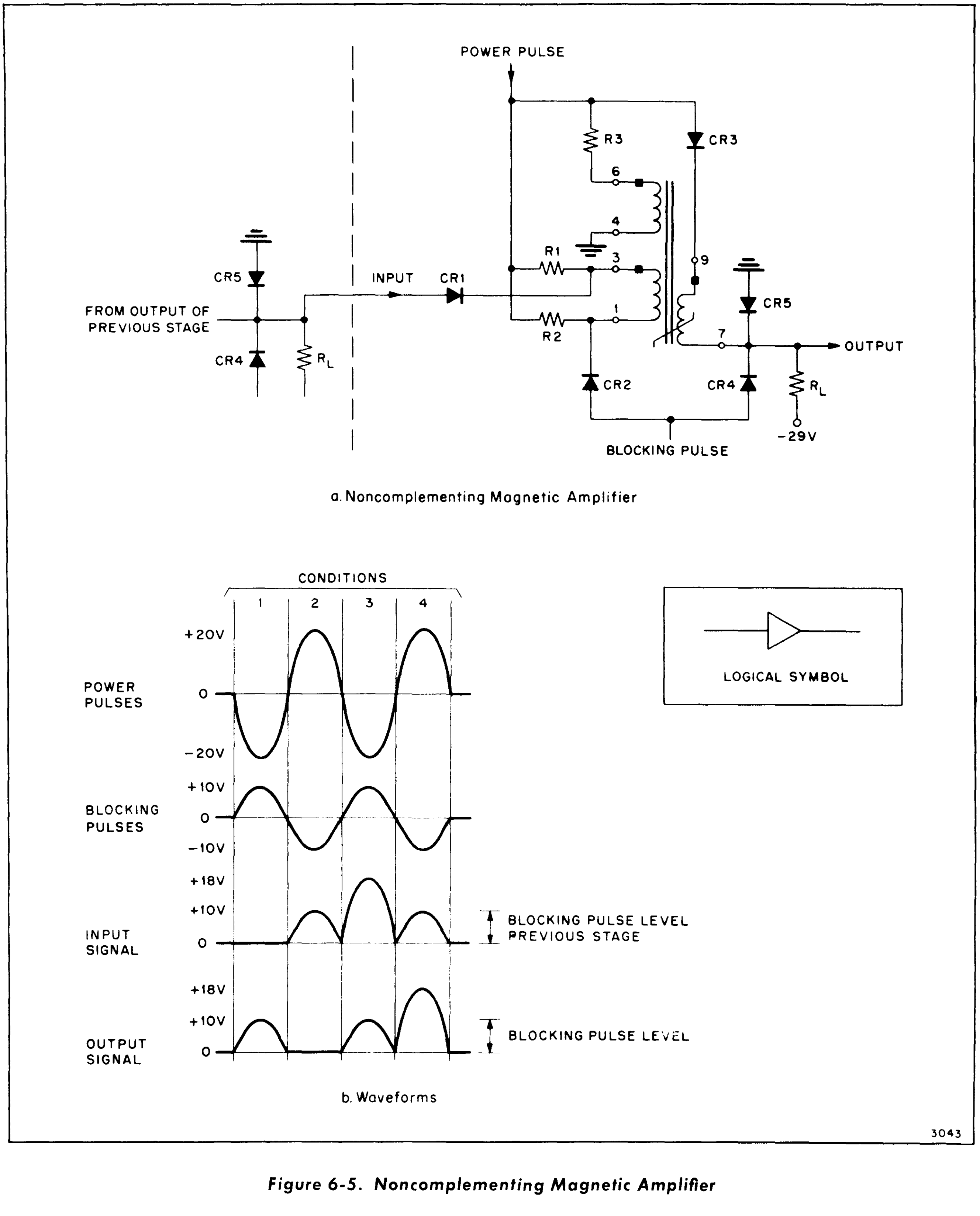
Condition 3 is practically identical to condition 1, except the input signal is approximately +18 volts. Diodes CR1 and CR2 are forward-biased thus placing terminal 3 of the primary winding at the input potential and terminal 1 at the blocking pulse potential. The net difference across the winding is 8 volts. Terminal 3 is positive in relation to terminal 1; thus current flows to switch the core from $+B_r$ to $-B_r$. However, current also flows through the bias winding and opposes the current through the primary winding. Consequently, the primary field is inhibited by the bias field and the core remains at $+B_r$.

When the next positive power pulse occurs during condition 4, the secondary presents a low impedance and the power pulse appears at the circuit output.

In figures 6-4 and 6-5 the input and output signals contain a +10-volt signal in addition to the +18 to +20-volt information pulses. These +10-volt signals are developed by the blocking pulses and have no logical significance. The blocking pulses applied to a stage appear in its output 180 degrees out of phase with the regular output signal and approximately one-half the amplitude of the output. These blocking pulses appear at the input of the driven stage, but they are not of the correct phase to have any logical effect on the stage.

6-13. MAGNETIC AMPLIFIER CIRCUIT VARIATIONS

6-14. DOUBLE-POWER CORES. When it is necessary to drive more cores than is possible with a standard single-power core (more than 35 drive units), a special double-power core is used instead of two standard cores connected in parallel. The double-power core has about twice the volume of magnetic material of a single-power core, and therefore requires twice the input power: 10 drive units for a complementer and 14 drive units for a noncomplementer. The output of a double-power core is 105 drive units.



a. Noncomplementing Magnetic Amplifier

b. Waveforms

Figure 6-5. Noncomplementing Magnetic Amplifier

The circuit for a double-power core is practically identical to that of a single-power core. The only change is that resistors of different value are used, and the output diode (CR3) is replaced by three diodes in parallel because of the increased current requirements of the circuit.

6-15. ARITHMETIC REGISTERS. An arithmetic register stores information for any desired period. Because of the inherent delay of magnetic amplifiers, one pulse time is required to pass a signal through two magnetic amplifiers in cascade. A series of 24 magnetic amplifiers provides sufficient delay to store one word (12 digits). If the output of the last amplifier is applied to the input of the first amplifier, a recirculating path is established which can store the word for an indefinite period. An arithmetic register consists of a string of noncomplementers designed only to delay information.

The noncomplementers used in the arithmetic registers have an S (special) or a T (transition) designation that indicates they are slightly different from standard noncomplementers, and they lack bias windings. Therefore, the input to one of these cores must be increased to 10 drive units. Since one core drives only one other core in the register, the output of each core is reduced to about 10 drive units and the spurious-output-suppressor circuits are omitted. Since there is only one input line to each core and no isolation of inputs is required, the input diodes are also omitted. All low-power units that do not contain an input diode and a spurious-output suppressor have an S designation.

Because the output of an arithmetic register may be supplied to several circuits simultaneously, a low-power S unit cannot be used in the output stage. The output stage of an arithmetic register is a standard noncomplementer with an output of 35 drive units. The input to the standard unit must have a ground reference level and contain no spurious signals. Thus the stage immediately preceding the output stage must contain a spurious-output suppressor and a pulldown circuit. When a low-power core contains this output circuit (transition circuit), it has a T designation.

Figure 6-6 shows a simplied arithmetic register circuit capable of providing a two pulse-time delay. The arithmetic registers used in the system are similar except for additional S-type noncomplementers.

6-16. TRANSISTOR AMPLIFIER PACKAGE (TAP). Frequently magnetic amplifier, either complementer or noncomplementer, may drive a relay or transistor stage. In such applications the magnetic amplifier output must contain only information signals and no blocking pulses.

Since the blocking pulses must be removed from the output windings, diode CR4 of the standard magnetic-amplifier circuit is removed. To replace the function of the blocking pulses on the core, the diode equivalent to CR5 is returned to +3 volts instead of ground potential. This ensures that spurious outputs are not developed in the magnetic amplifier. The remainder of the special driver-package circuit is identical to the standard single-power magnetic-amplifier circuit.

6-17. DRUM-STORAGE CIRCUITS

The drum-storage circuits are divided into three groups: read, write, and head selection. A general analysis of the circuit groups is presented under heading 6-18. The detailed analysis of each circuit is accompanied by a simplified schematic diagram that is intended solely as a tutorial guide. The actual schematic diagram of a circuit package, showing test points, location data, and other maintenance information, should be referenced for any purpose other than a general theoretical understanding

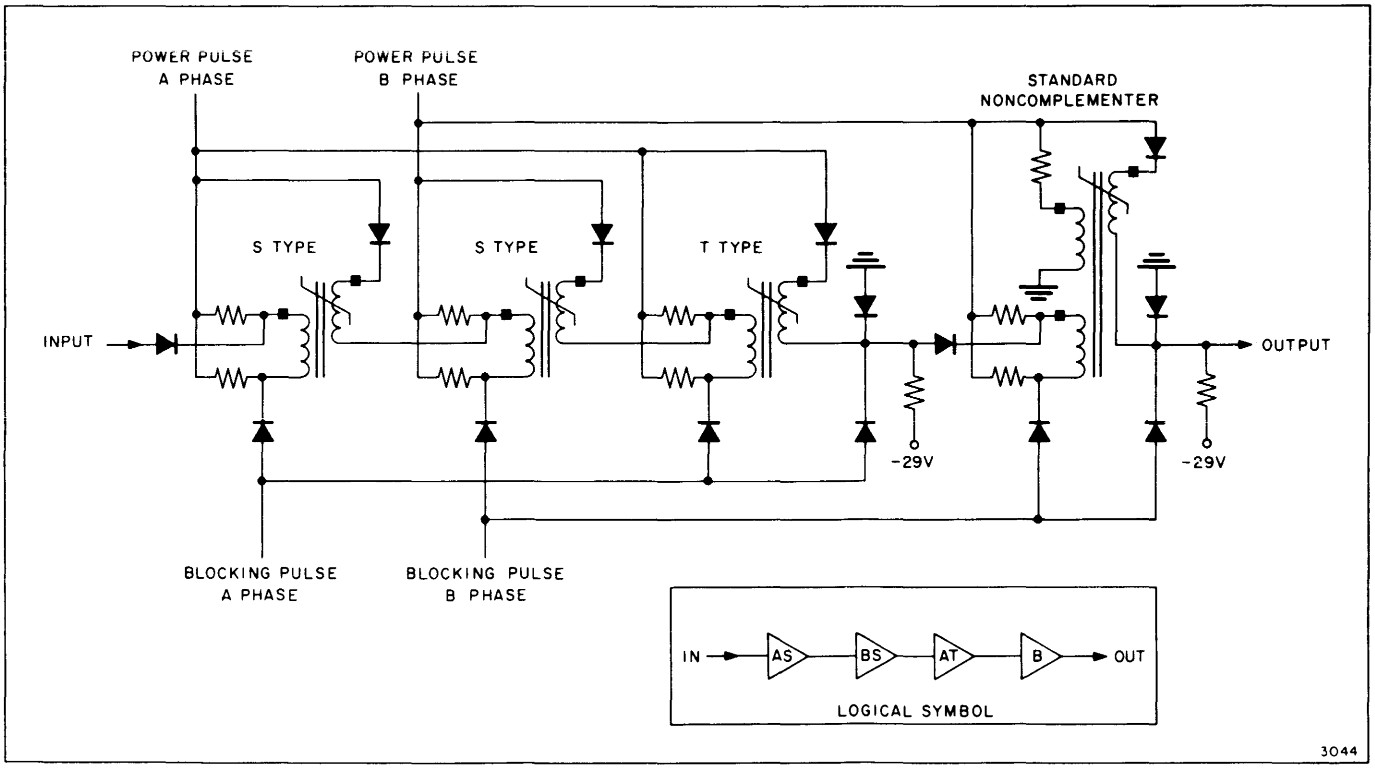


Figure 6-6. Arithmetic Register

of the circuit. Illustrated waveshapes are idealized representations of true waveshapes: they are not intended for maintenance purposes.

6-18. GENERAL

The main component of the storage system is a rotating drum (figure 6-7) with a magnetically sensitive surface. Onto this surface, bits of data are recorded as discretely magnetized areas whose polarities represent a binary 1 or 0.

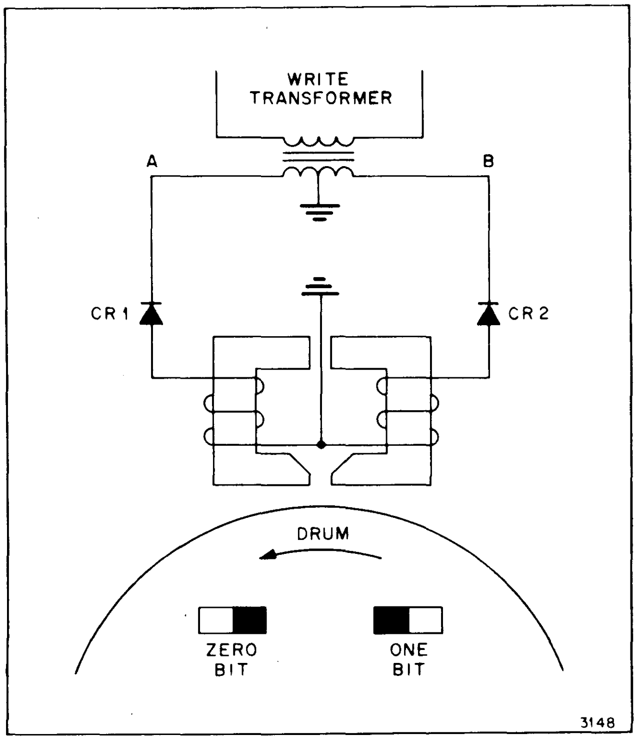


Figure 6-7. Recording Method

Information is written onto or read from the drum surface by magnetic recording heads located within the drum housing. The five bits (four information bits and a check bit) which make up a digit are recorded or read in parallel from five tracks on the drum circumference.

The central processor controls information sent to the drum surface. (Refer to section 4 for a logical description of the read-write process.) Figures 6-8 and 6-9 show the circuits that complete the processor-initiated information flow to or from the drum surface.

For a write operation, information is supplied to the write circuitry by way of the m lines. This information is amplified through a phase-modulation coder and converted from pulsed information to a sine-wave through a write-transformer (WXP-S) package. The WXP-S package produces a push-pull output current that passes through a diode cluster and drives the read-write heads. All read-write heads receive the same write current simultaneously, but only the heads whose circuits are completed by way of the head-selection network (refer to section 4, head selection) have an effect on the drum surface. Thus the diode cluster ensures that only the selected heads will present a completed circuit for write current. The read circuitry is blocked during a write operation by a write-pedestal package (WPP) that prevents the incoming write information from being interpreted as read data.

During a read operation the head-selection procedure is identical, but the write circuits are inhibited, and the WPP circuit alerts the read amplifier packages (RAP-S). When the correct drum address is detected, the circuits of the selected read-write heads are completed, but the function of the heads is now passive rather than active. Signals induced into the read-write heads are amplified through the RAP-S packages and applied to read flip-flops (RFF-S). The RFF-S packages store the binary identity of the read signals until signals from the probe and clear generator packages (PBC) cause the contents of the RFF-S packages to be gated onto the DM′ lines. Immediately after the gating signal, all read flip-flops are cleared to 0's.

The read circuitry is essentially a phase-sensitive network that reconverts the read signal from a sine wave of two possible phases into a single-ended pulse of distinct polarity. The pulse resulting from a binary 1 does not set the read flip-flop: the pulse from a binary 0 sets the read flip-flop.

6-19. WRITE CIRCUITRY

Information is recorded on the drum when the read-write head is pulsed with a current which, in turn, magnetizes small areas of the drum surface (figure 6-7). Recording current passes through the head first in one direction and then in the other; the order of these directions depends upon whether a 0 or 1 bit is recorded.

To write a 0 bit, a voltage is induced into the secondary of the write transformer such that point A (figure 6-7) goes negative and point B goes positive with respect to the grounded centertap. Current flows from ground through the centertap of the write transformer, through CR1, through half of the head winding, and back to ground through the head-winding centertap. During the next alternation, point B goes negative and point A goes positive with respect to ground. During this alternation current flows through the other half of the head winding.

To record a 1 bit, the order of the magnetizing currents is reversed. Point B is driven negative on the first alternation, and point A goes negative on the second alternation. Changing the order of the magnetizing currents changes the polarity of the recorded bits. In the example, the relative polarities of the 0 and 1 bits recorded are as shown in figure 6-7. The shaded area represents a north pole.

6-20. DIODE CLUSTER. The diode cluster above the drum housing (figure 6-8) contains the head diodes which are shown as CR1 and CR2 in figure 6-7.

When none of the matrix-selector packages (MSP) (figure 6-9) are driven, diodes CR1 and CR2 (and the

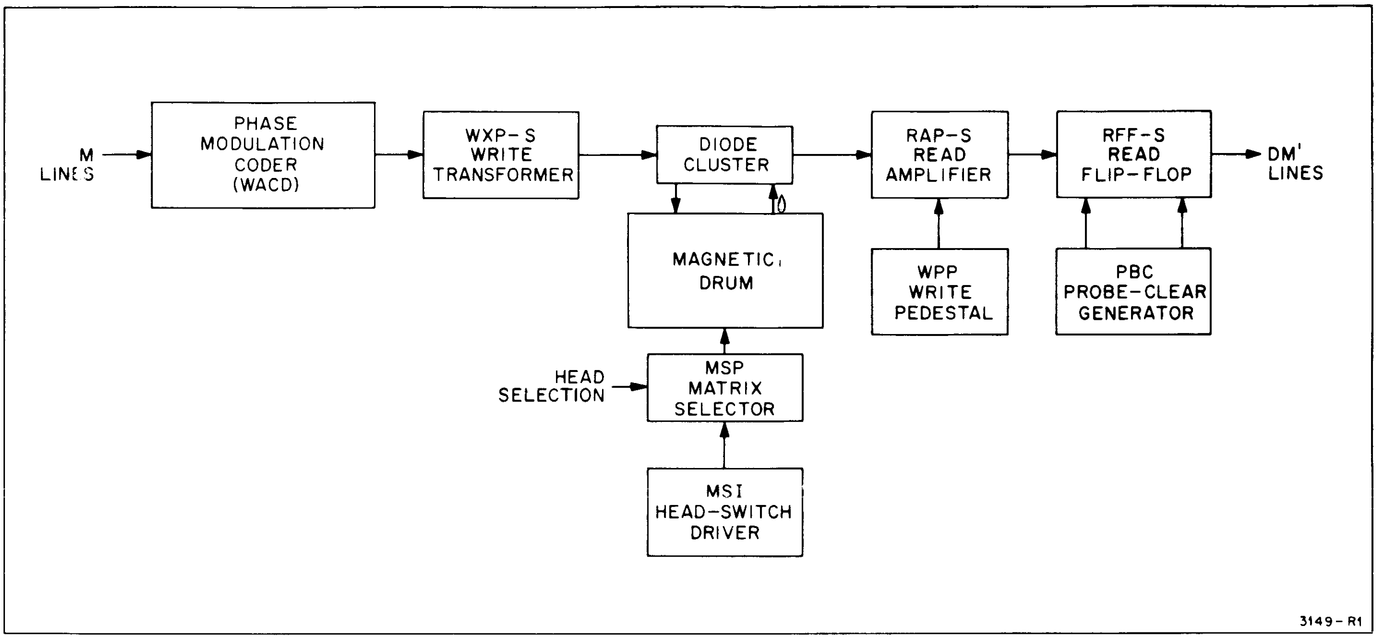


Figure 6-8. Storage Unit, Block Diagram

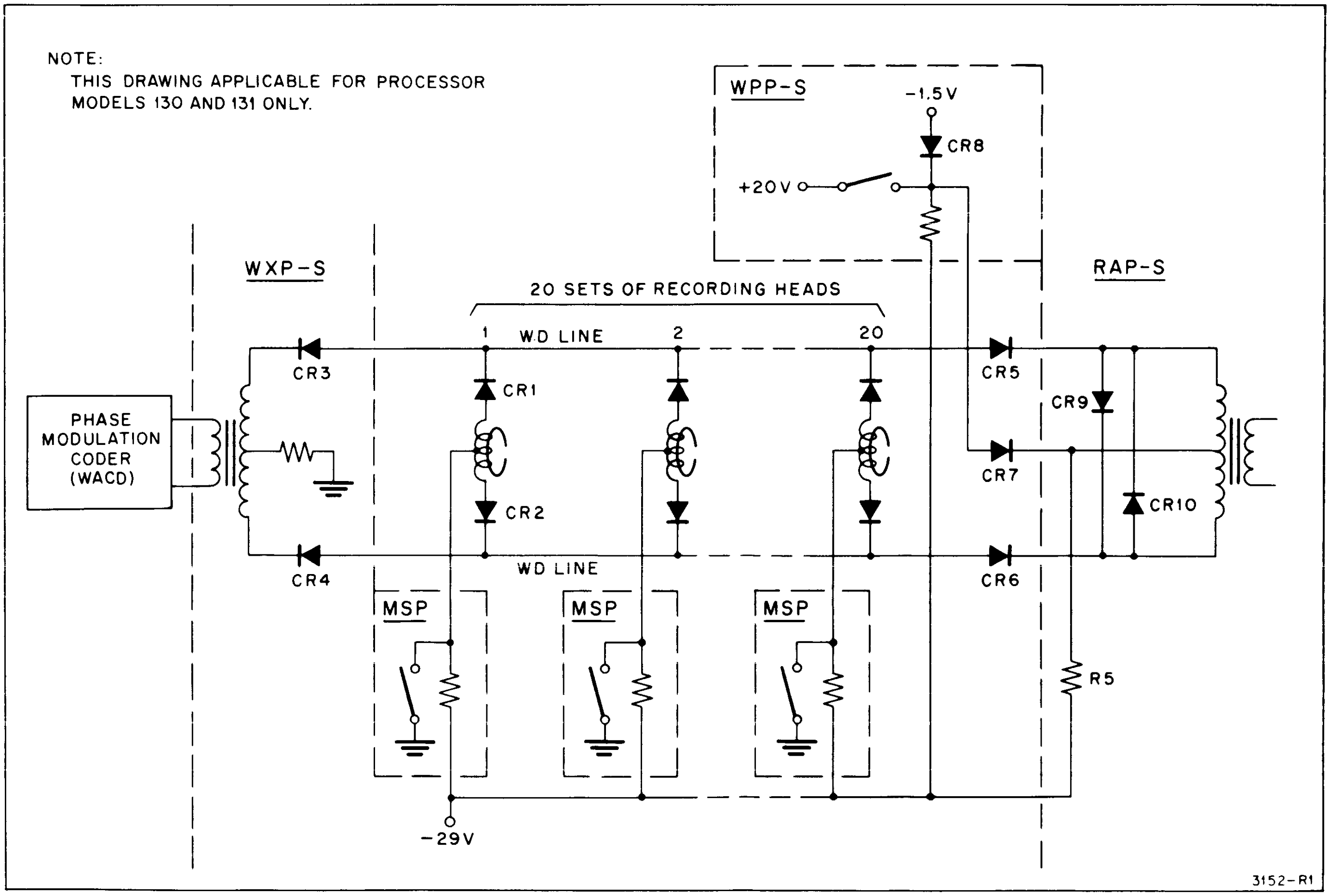

NOTE:
THIS DRAWING APPLICABLE FOR PROCESSOR MODELS 130 AND 131 ONLY.
WPP-S
-1.5V
CR8
+20V
WXP-S
RAP-S
20 SETS OF RECORDING HEADS
1
WD LINE
2
20
CR3
CR1
CR5
CR9
PHASE MODULATION CODER (WACD)
CR7
CR10
CR2
WD LINE
CR4
CR6
MSP
MSP
MSP
R5
-29V
3152-R1

Figure 6-9. Read-Write Circuit Interconnections

equivalent diodes in the other heads) are reverse-biased by the potential through the head-winding centertap and the –29-volt supply. Under this condition no current flows through any of the heads.

To write with head number 1, the MSP connected to that head returns the head-winding centertap to ground potential. The reverse bias on CR1 and CR2 is removed and a write signal is developed across the write transformer. Current flows through head number 1 first in one direction and then in the other. The diodes of each of the remaining heads isolate them from this circuit.

6-21. WRITE TRANSFORMER PACKAGE (WXP-S). The write transformer (figure 6-10) employs a centertapped secondary to provide push-pull operation for the modified return-to-zero phase-modulation recording system. The primary is not centertapped because it is connected across the outputs of the driving magnetic amplifiers. Since no centertap is used, the primary current is absorbed by resistive sinks at each end of the winding. These sink circuits use a d-c supply because the drive phase at each end of the primary changes for 0's and 1's.

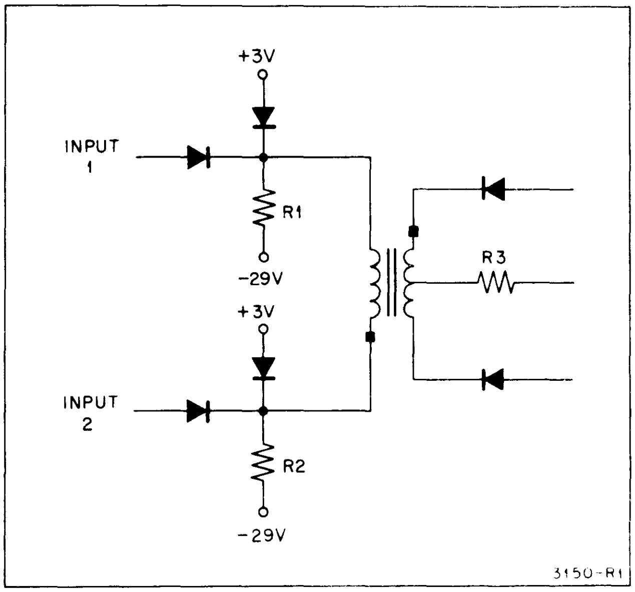


Figure 6-10. Write Transformer Circuit (WXP-S)

The sink circuits also act as current-clamping circuits which limit the maximum writing current in the secondary. The secondary voltage is set to establish head current within one-half pulse time and yet clamp the 0-to-1 transition currents at the same level as the nontransition currents. The idealized voltage and current waveforms in figure 6-11 clarify the operation of the limiter. Resistor R3 in the centertap of the secondary is used solely for monitoring purposes. The waveform for the current through or the voltage across this resistor is shown in figure 6-11.

Because of the reactive components in the write circuitry, the primary current lags the driving voltages. The primary current also accounts for the peaked voltage waveforms at both inputs. If e1 is positive (figure 6-11), current flows from the –29-volt supply through R2, the primary winding, and the input diode to the positive source. When the current through R2 exceeds the sink current, the potential at input 2 goes more positive than +3 volts. This increase in voltage is shown as a peaked voltage on the e2 waveform. When the primary current increases to maximum in the other direction, a similar peak develops at input 1. The difference between the input voltage and the peaked voltage is the actual primary driving voltage during that interval.

The current through the center of the secondary is a unidirectional pulsating current which approaches the waveform shown. Writing current is approximately 110 milliamperes for a 300 milliampere output from the phase modulation coder.

6-22. PHASE-MODULATION CODER (WACD). Four double-powered magnetic amplifiers, one for each phase for both 0's and 1's, are required to drive one WXP-S package. These magnetic amplifiers differ from the standard magnetic amplifiers in that they do not use blocking pulses and their ouputs are clamped to +3 volts instead of ground. The phase-modulation coder is also referred to as a write amplifier and complementer driver (WACD) package. The operation of the phase-modulation coder shown in figure 3-4 is described under heading 3-100.

6-23. READ CIRCUITRY

Figure 6-9 shows the read-write circuit interconnections in which the heads are the common elements between the read and write circuits. Twenty heads are connected to a headbar, although only three are shown here. The matrix-selector and write-pedestal packages are both shown as switches to aid in the understanding of their functions rather than their methods of actuation.

6-24. WRITE-PEDESTAL PACKAGE (WPP-S). The write-pedestal circuit disconnects the read amplifier from the WD lines during write operations to prevent the write signals from blocking the read circuits.

As shown in figure 6-12, the centertap potential on the primary of the read-amplifier input transformer is normally held at +1.5 volts. This voltage is maintained through a circuit consisting of diodes CR7 and CR8, and resistor R5. During the write operation the write-pedestal circuit causes a +20-volt signal to be applied to the centertap of the read-amplifier input transformer. This is several volts higher than any write signals on the WD lines and ensures that the coupling diodes CR5 and CR6 are reverse-biased during the write operation to prevent the write signals from entering the read-amplifier circuit.

Since five bits are being either written or read simultaneously, one write-pedestal package drives five read amplifiers simultaneously. A transistor amplifier package (TAP) provides the driving signals for the write-pedestal circuit and keeps the circuit free of blocking pulses.

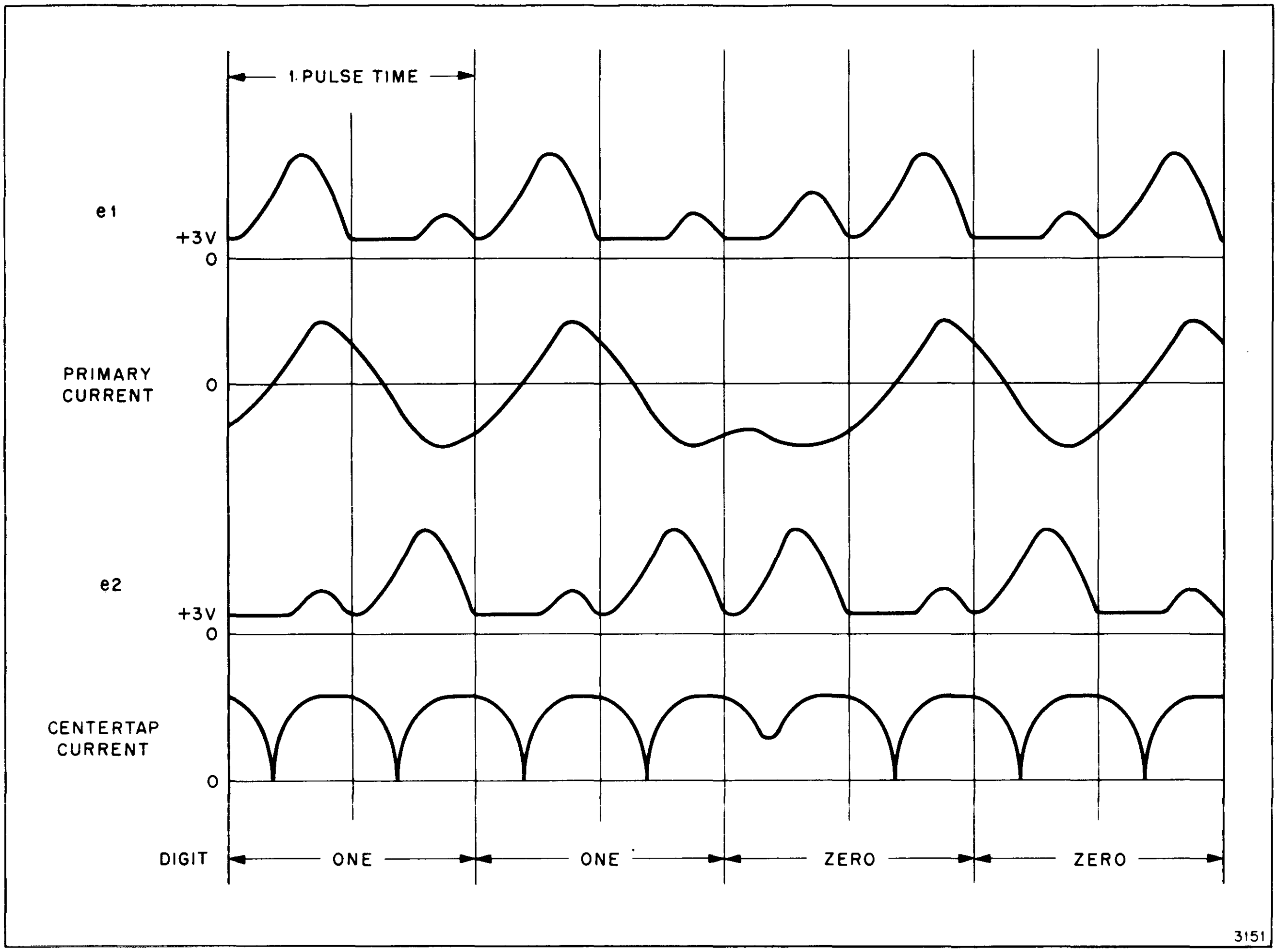


Figure 6-11. Write Transformer Package Waveforms

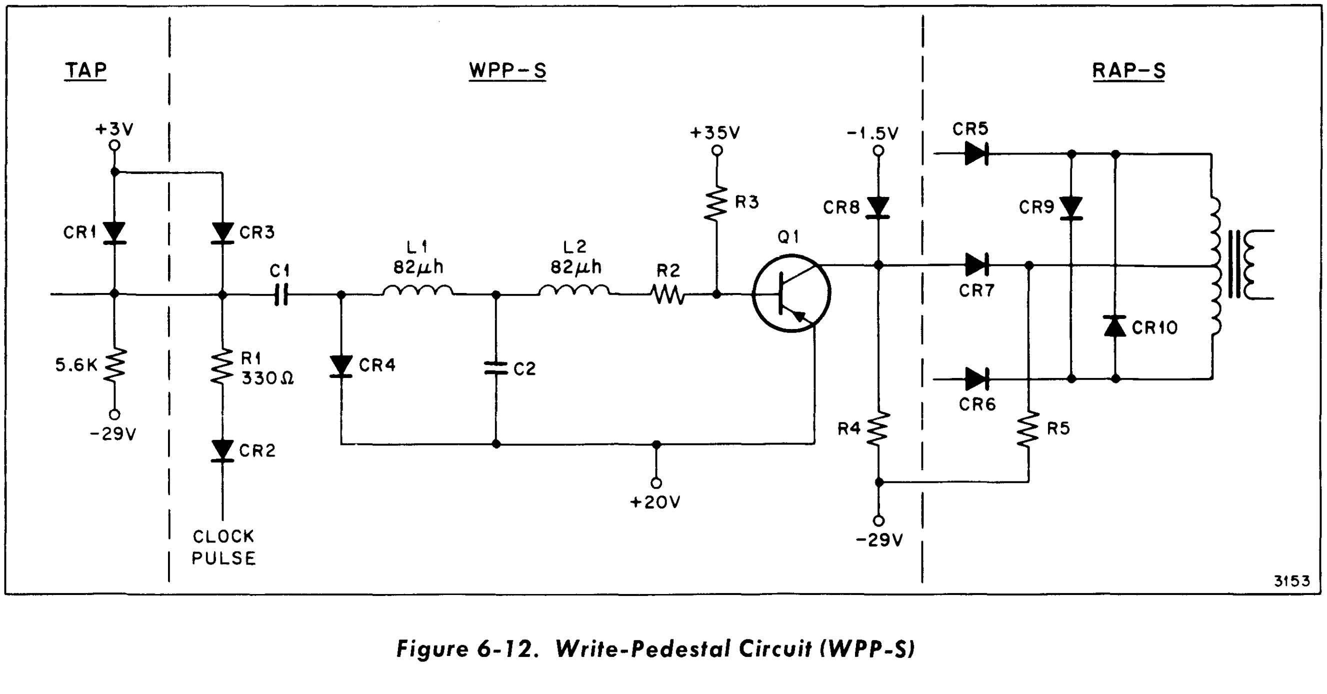


Figure 6-12. Write-Pedestal Circuit (WPP-S)

The input signal to the write-pedestal circuit consists of a train of 16 pulses, one per pulse time. The pulses begin four pulse times before a write operation and continue until the writing is completed. Therefore, the write-pedestal voltage on the transformer centertap must rise within four pulse times. One word time is allowed for the write-pedestal voltage to fall, because it is not possible to read immediately after writing.

The write-pedestal circuit (figure 6-12) consists of a d-c restorer (CR4) and a choke-input T-type filter (L1, L2, C2). The filter output drives the base of transistor Q1.

Diode CR4 establishes the d-c level of the input signals and provides signal inversion. The TAP output is coupled by capacitor C1 to CR4, whose cathode is returned to +20 volts. With no output from the TAP, CR4 is forward-biased and a small current flows from +20 volts through CR4, L1, L2, R2, and R3 to +35 volts. The TAP output is clamped to +3 volts by CR1 and CR3. Coupling capacitor C1 therefore has 17 volts across its terminals. When an output pulse appears, C1 discharges through CR4 to 20 minus E_{max} (E_{max} is the peak output voltage of the TAP). When the TAP output returns to +3 volts, CR4 is reverse-biased and a negative pulse of amplitude E_{max} minus 3 volts is developed across the filter input. Applying a negative pulse to the filter causes Q1 to conduct and thus provide a low impedance path to +20 volts for the centertap of the read-amplifier input transformer.

The filter converts the block of 16 pulses into a single pulse of 16-pulse-time duration. This ensures that a read amplifier remains isolated during the entire writing operation.

A low-impedance source is required to drive the write-pedestal circuit. When the output pulse begins to fall, the output impedance of the TAP is high. To lower the output impedance of the TAP, a 330-ohm pulldown resistor (R1) is connected to the clock through CR2.

6-25. READ-AMPLIFIER PACKAGE (RAP-S). The read-amplifier circuit (figure 6-13) consists of two class-A, common-base, transformer-coupled transistor stages plus input and output transformers. Also included are limiting, differentiating, and phase-correcting networks which enable the read amplifier to accept a signal from the head, amplify it, and transmit it to the read flip-flop in the correct phase.

The read-amplifier input transformer (T1) changes the input from a balanced signal to a single-ended signal. The input transformer is coupled to the selected head by d-c current which flows from the –29-volt supply through resistor R5 to the centertap of T1 primary, where it divides between the halves of the primary winding and through both halves of the head winding to the driven matrix selector (MSP).

While the currents through both halves of the input winding are approximately equal, it is only their sum that is a constant. Unequal currents through the head diodes or the coupling diodes CR5 and CR6 cause unequal currents through the input winding of T1. When a head is selected for reading, an unbalanced current causes a d-c step voltage across the primary of T1.

To keep this initial d-c imbalance out of the amplifier, an RL network follows the input transformer to eliminate the effects of this step voltage by differentiation.

Coil L1 is the inductive component of the RL differentiating network. The head windings comprise an RL equivalent circuit. To compensate for the inductive component, an RC circuit (R1-C1) is placed in series with the secondary of T1. The differentiating circuit has a time constant of 0.4 microsecond.

The amplitude response produced by the differentiation circuit is accompanied by an unfavorable phase response which shifts the signal components enough to limit the

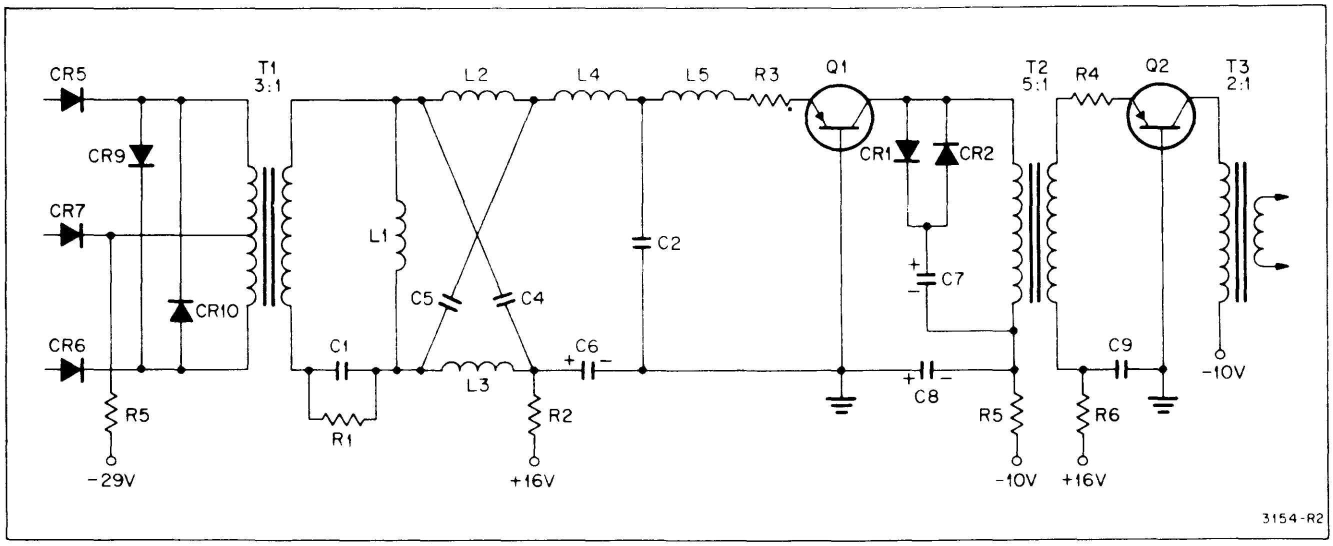


Figure 6-13. Read-Amplifier Circuit (RAP-S)

allowable timing variations at the probe gate. To correct this phase distortion an all-pass lattice filter (L2, L3, C4 and C5) is inserted after the differentiator. A π-type filter (L4, L5, and C2) also provides a fixed delay at the signal frequencies and reduces high-frequency noise. The transmission delay of the differentiator and correction networks is one-half pulse time. To allow for this delay, all writing is performed one-half pulse time earlier than would otherwise be necessary.

Stable current gain through the stage is achieved with two transistorized amplifier stages. Diodes CR1, CR2, and resistor R4 form a limiting network; CR9 and CR10 across the primary of T1 are protector diodes which ensure that no writing signals are transmitted to the read-amplifier circuit. These diodes operate only if the write-pedestal voltage (+20 volts) is not applied to the centertap of the T1 primary during a write operation. The amplitude of any normal read signal is insufficient to forward-bias CR9 or CR10.

A head normally develops a read signal of about 250 microamperes in the matched inputs of the read amplifier. Neglecting all losses, a current gain of 3 is realized in T1, a gain of 5 in T2, and a gain of 2 in T3. The unit, therefore, has a total current gain of 30. This produces an output current of 7.5 milliamperes. When the alpha and compensator losses are included, a more realistic output current of about 4 milliamperes is realized from a read-amplifier package.

6-26. READ FLIP-FLOP PACKAGE (RFF-S). The read flip-flop output (figure 6-14) is such that no signal (0 volt) is developed to represent a 1-bit, and a signal (+18 volts) is developed for a 0 bit.

When no input is supplied by the preceding RAP-S circuit and the probe pulse is at +20 volts, transistors Q1 and Q2 are at cutoff. Transistor Q3 is forward-biased through R6 and R7; Q3 conducts and clamps the base of Q4 to +19 volts, thus biasing Q4 to cutoff. The base current Q3, through R7, holds the base of Q5 below +19 volts while the emitter of Q5 is connected to the +19-volt supply. Transistor Q5 conducts heavily and approximately +19 volts is developed at its collector to reverse-bias Q6. Thus, during the nondriven state, Q3 and Q5 conduct and all other transistors in the read flip-flop circuit are biased to cutoff.

If a signal from the RAP-S is impressed across the secondary of T3 such that the base of Q1 is driven negative with respect to the emitter, current flows from the +18-volt supply through R2 and Q1 to the +19-volt supply. This current causes the potential on the collector of Q1 and the emitter of Q2 to become approximately +19 volts. If at this time a probing pulse arrives which lowers the base of Q2 below the emitter voltage, Q2 turns on, and current flows from the +16-volt supply through R7, R6, Q2, and Q1 to the +19-volt supply. This current raises the potential on the base of Q3 to +19 volts and turns off Q3. As Q3 turns off, the collector of Q3 and the

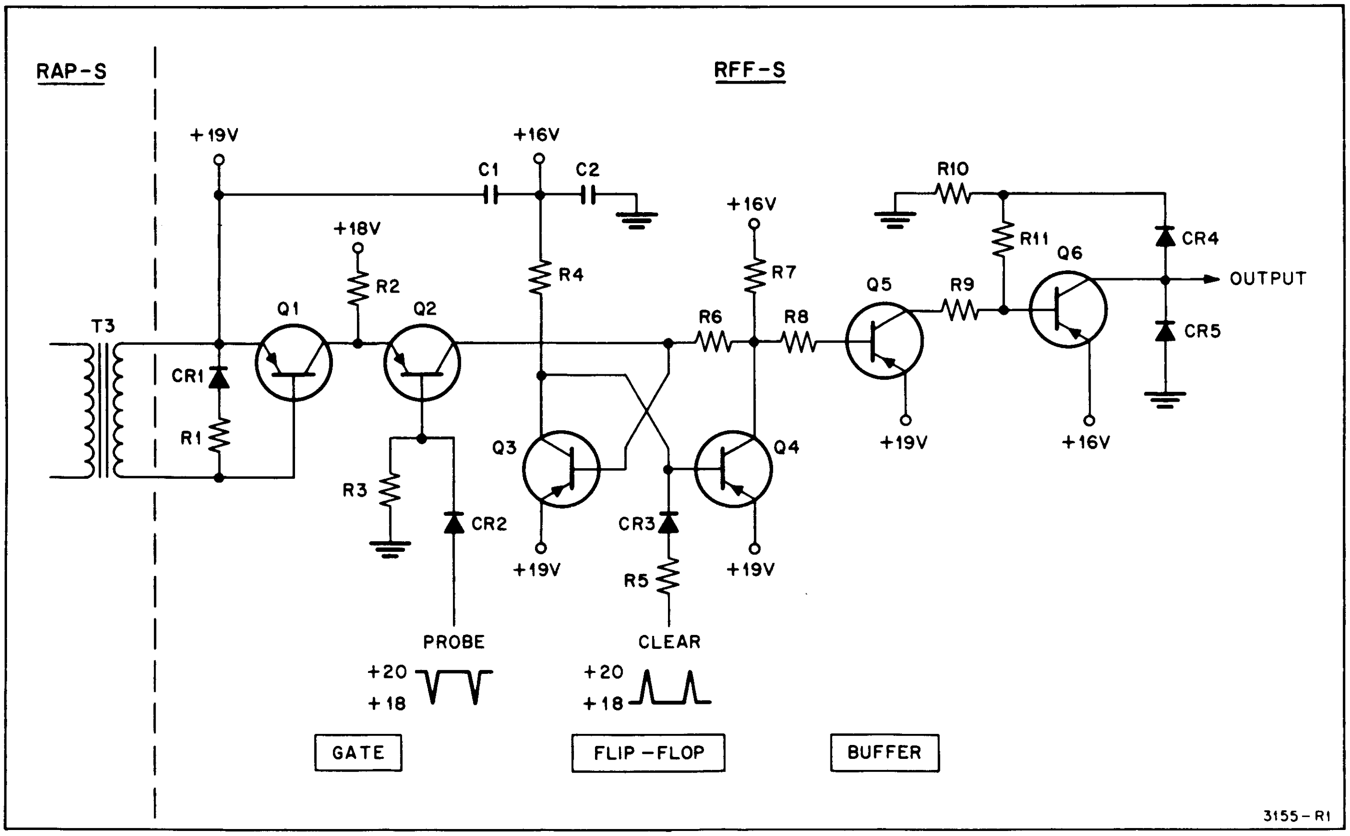


Figure 6-14. Read Flip-Flop Circuit (RFF-S)

base of Q4 approach +16 volts and Q4 turns on. Current from the +16-volt supply through R7 and Q4 to the +19-volt supply places the collector of Q4 and the base of Q5 at about +19 volts. This voltage turns off Q5, and the collector of Q5 and the base of Q6 fall below +16 volts. Transistor Q6 then conducts heavily, developing an output signal of approximately +16 volts.

If transistor Q6 is saturated, its collector and emitter would be at essentially the same potential (+16 volts). To prevent Q6 from saturating, CR4 conducts before saturation is reached and holds the collector of Q6 below +16 volts. Diode CR5 prevents the collector of Q6 from going below ground potential when Q6 is at cutoff.

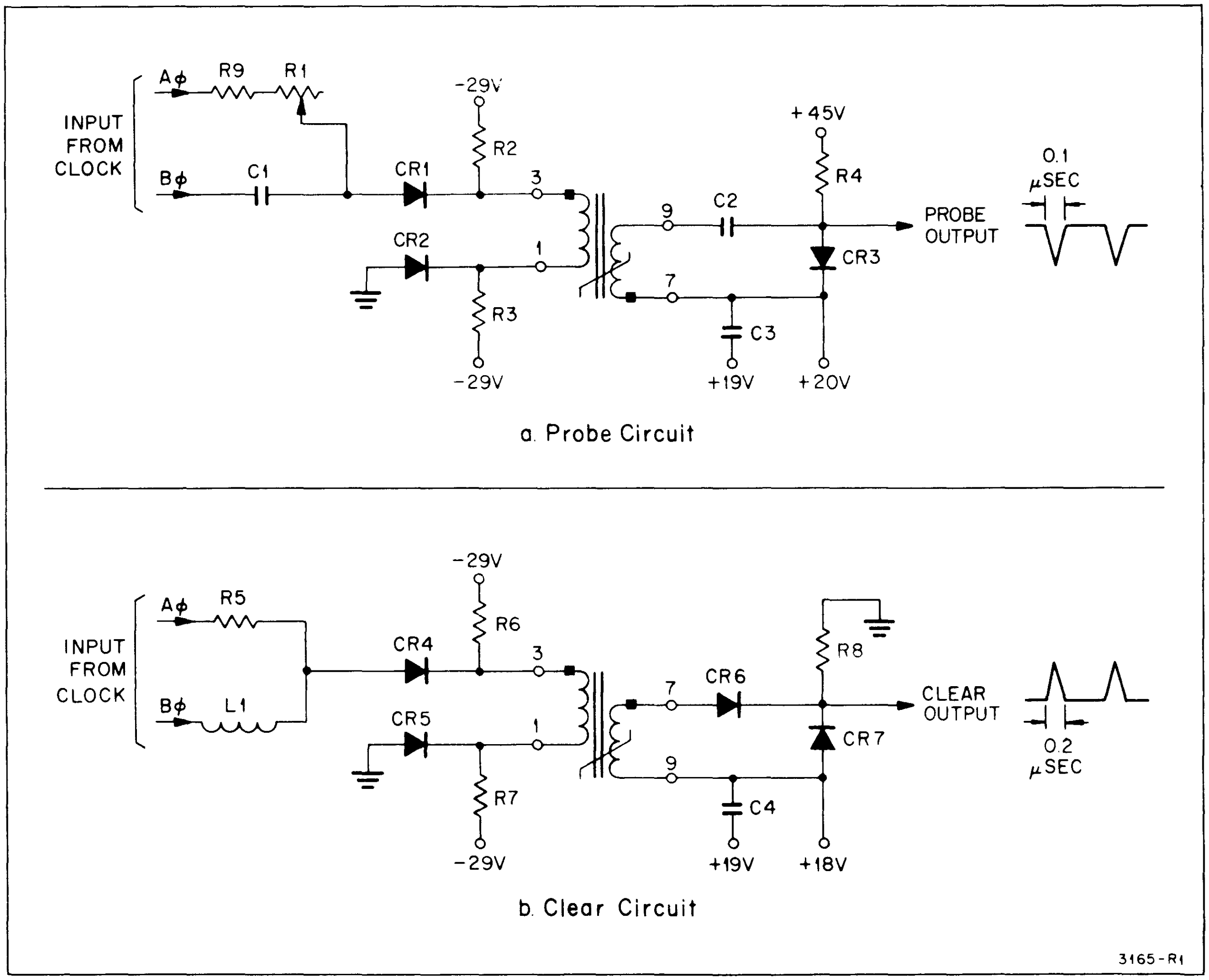
Transistor Q2 remains on only for the duration of the probe pulse (0.1 microsecond). Transistor Q3 is held off, however, by feedback from Q4. When Q4 turns on, +19 volts is applied to the base of Q3 through R6. Transistor Q3 is held off while Q4 is on, and Q4 stays on until the +20-volt clear pulse is applied through R5 and CR3 to its base. The clear pulse (0.1 to 0.2 microsecond in duration) is sufficient to turn off Q4, at which time the collector potential of Q4 drops to +16 volts. This voltage, applied to the bases of Q3 and Q5, causes Q3 and Q5 to turn on as Q4 turns off.

A negative-going input signal applied to Q1 causes it to turn on and place a load on T3. If the input signal polarity is reversed, Q1 remains off, leaving T3 unloaded. To balance the load on T3, CR1 and R1 are placed across the secondary of T3. When Q1 is off, CR1 conducts. This arrangement keeps the load on T3 relatively constant.

Capacitors C1 and C2 are bypass capacitors which prevent undesirable a-c coupling among the various stages.

6-27. PROBE AND CLEAR PACKAGE (PBC). The probe and clear circuits (figure 6-15) are both clock-driven units. The probe circuit produces a negative-going output spike for every positive alternation of the A-phase clock, and the clear circuit produces a positive-going output spike for every positive alternation of the B-phase clock.

The probe circuit (figure 6-15a) continuously develops negative-going pulses of 0.1-microsecond duration by saturating a small core of square-loop magnetic tape with current from the sine-wave clock. The clock input is applied through an RC phase shifter (R1-C1) so that the



a. Probe Circuit

b. Clear Circuit

Figure 6-15. Probe and Clear Circuits (PBC)

probe timing can be adjusted. Normally, terminal 1 of the input winding is clamped to ground through CR2. The d-c reset current that flows from the -29-volt supply through R2, the input winding, and CR2 to ground ensures that the core is reset to $-B_s$. When a positive clock pulse occurs, CR1 is forward-biased and a current from the -29-volt supply through R3, the input winding, and CR1 to the positive clock saturates the core to $+B_s$. When the positive clock pulse has elapsed, the core is slowly reset by the d-c reset circuit.

In the output circuit the output terminal is normally clamped to +20 volts through CR3. When the core is saturated to $+B_s$, a voltage induced in the output winding drives terminal 9 negative with respect to terminal 7. This negative-going pulse is applied to the output terminal through C2 and causes the output voltage to go well below +20 volts for the duration of the pulse (0.1 microsecond).

The clear circuit (figure 6-15) continuously develops positive-going pulses of 0.2-microsecond duration and operates like the probe circuit. The B-phase clock input is applied through an RL phase shifter (R5-L1). Terminal 1 of the input winding is clamped to ground potential through CR5. A d-c reset current from the -29-volt supply through R6, the input winding, and CR5 to ground resets the core to $-B_s$. A positive B-phase clock input signal forward-biases CR4, and current from the -29-volt supply through R7, the input winding, and CR4 to the positive clock saturates the core to $+B_s$. The d-c reset current slowly resets the core after the positive input signal has elapsed.

In the output circuit current flows from ground through R8, where it divides between CR6 and CR7 to the +18-volt supply. The current through CR6 and the output winding aids in resetting the core to $-B_s$. The output terminal is normally clamped to +18 volts through CR7. When the positive input signal saturates the core to $+B_s$, a voltage is induced in the output winding which drives terminal 7 positive with respect to terminal 9. The positive-going signal on terminal 7 is applied to the output terminal through CR6 and causes the output voltage to go well above +18 volts for the duration of the pulse (0.2 microsecond).

The clear pulses are generated about 0.1 microsecond before the end of the B-phase pulse to reset the RFF-S package. Figure 6-16 shows the read-circuit waveforms.

6-28. HEAD-SELECTION CIRCUITRY

6-29. MATRIX-SELECTOR POSITIVE PACKAGE (MSP). The matrix-selector positive circuit (figure 6-17) is a transistor (Q1) with its collector connected to the centertaps of five heads in a storage band. The collector also returns to the -29-volt supply through resistor R5. When Q1 is not conducting, all head windings are at -29 volts. This potential reverse-biased the diodes from the information lines to the heads (CR1 and CR2 in figure 6-8): therefore, the heads are disconnected from the information (read or write) lines. When Q1 conducts, the collector potential is clamped to ground through the emitter. Diodes CR1 and CR2 of the diode cluster (figure 6-7) are now forward-biased and the heads are connected to the information lines for reading or writing.

In the absence of an input signal (figure 6-17), transistor Q1 is biased to cutoff through CR1.

When a negative input signal is applied, Q1 conducts and the collector and head-winding centertap are grounded through Q1.

The negative input signal to the MSP is a train of half sine waves from a pair of magnetic amplifiers on the MSI package. Capacitor C1 filters the input signal, and resistor R3 limits the input current.

Resistor R4 and capacitor C2 attenuate ripple and also provide a low impedance to the initial write current pulse through the heads. When Q1 conducts, all of the collector current flows through resistor R5. In about 2 microseconds this current reaches a value of approximately 120 milliamperes. Three microseconds later, if the heads are to be used for writing, the write current through the collector uses from 120 to 500 milliamperes. During this rise time Q1 tends to saturate and thus attenuate the first few pulses of write data. The combination of R4 and C2 reduces this attenuation by limiting the voltage drop at the collector of Q1 to 1 volt.

6-30. HEAD-SWITCH DRIVER PACKAGE (MSIS). The head switch driver package (figure 6-18) contains magnetic amplifiers which cause the MSP to conduct when an input (selection signal) is received. Three noncomplementers provide outputs of both phases and approximate a d-c input signal to the MSP. Each head-switch driver package drives one MSP. Two input terminals are provided on the driver package so that selection signals may come from two separate sources without any external diode buffers. Since the magnetic amplifiers are noncomplementing, output signals are developed only when an input signal is applied.

When a group of heads is to be selected for reading or writing.,a selection signal is applied to the input of the MSI several cycles before the appearance of the selected word. The first input signal (B phase) sets noncomplementers NC1 and NC3 to the low-impedance state. On the next half cycle, the A-phase output from NC1 sets NC2 to the low-impedance state, and the A-phase output from NC3 provides an input signal to the head switch transistor. On the next half cycle NC2 delivers a B-phase input to the transistor. Thus a B-phase input signal is converted to output signals which occur during both halves of the power pulse to approximate a d-c drive for the head-switch transistor.

The output circuit of NC2 and NC3 is modified to develop a negative rather than a standard positive output waveform. The polarity of the output windings and output diodes (CR12, CR13) is reversed, and a negative power pulse of the correct phase is used.

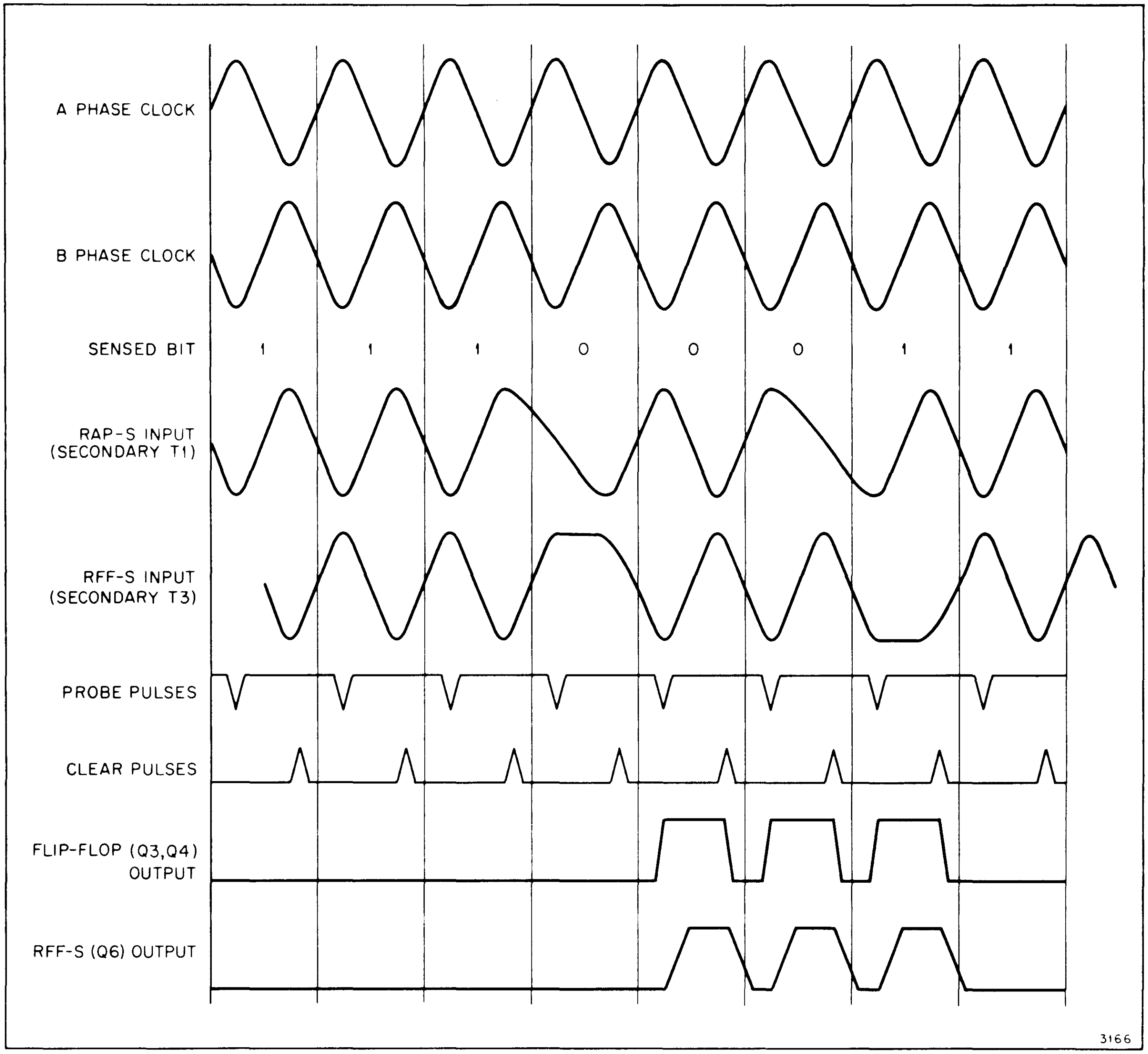


Figure 6-16. Read-Circuit Waveforms

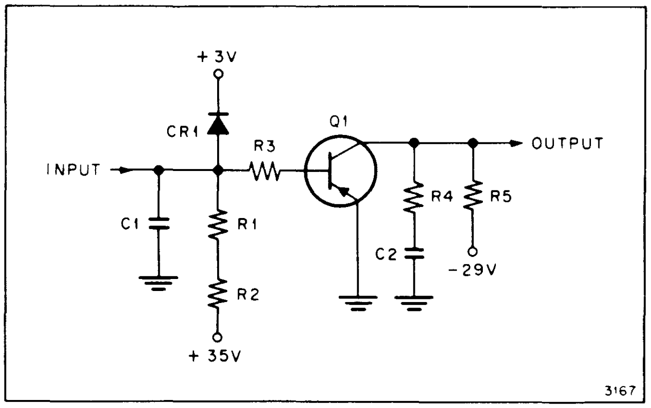


Figure 6-17. Matrix-Selector Circuit (MSP)

Blocking pulses are eliminated from the output circuits of NC2 and NC3 because they tend to turn on Q1 in the MSP circuit. The blocking pulses in a standard non-complementer ensure that the output diode remains reverse-biased while the core is being reset to the high-impedance state. For the same purpose. a d-c bias voltage (+3 volts in base circuit of the MSP) is used here to ensure that the blocking half cycle of the power pulse is of larger amplitude than the power half cycle.

The output circuits of NC2 and NC3 are connected to the base of Q1 in the MSP. The base input circuit of the MSP also serves as output clamp and spurious signal suppressor for the magnetic amplifiers.

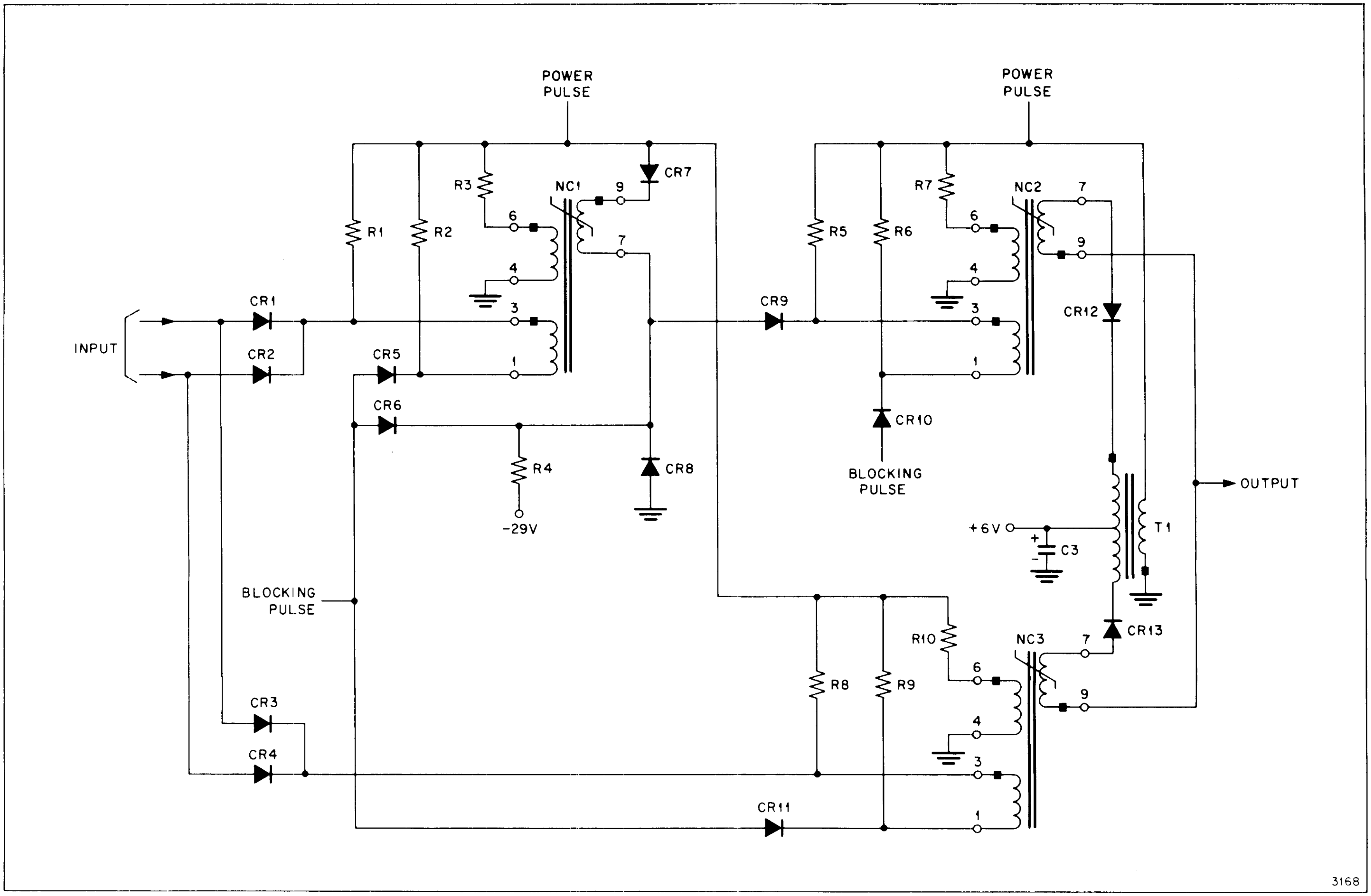


Figure 6-18. Head-Switch Driver Circuit (MSIS)

6-31. EXPANDED-STORAGE CIRCUITRY

In all processor models except 130 and 131, the following circuits are used for both 5000- and 9200-word operation. These are referred as expanded-storage circuits, but they will also be present in 5000-word systems.

Additional storage is obtained in the 9200-word version of the magnetic drum by decreasing the size of the read/write heads. The 9200-word drum is structurally different and contains more read-write heads than other drum versions.

Logically, the drum is divided into two halves, designated A and B. Section A stores 4000 words of normal memory (addresses 0000 to 3999) and 1000 words of fast memory (addresses 4000 to 4999): section B stores 3600 words of normal memory (addresses 5000 to 8599) and 600 words of fast memory (addresses 8600 to 9199). Total storage capacity in the expanded memory drum is 9200 words: 7600 words of normal memory and 1600 words of fast memory. The logical operation of the system with expanded storage is described in Section 3 under headings 3-110 through 3-119.

The function of the electronic circuitry for the expanded-storage system is similar to the standard system (models 130 or 131). Figure 6-19 shows the expanded storage read/write circuit interconnections. Alterations to existing circuit packages and additional circuit packages are as follows.

6-32. WRITE CIRCUITRY. The WXP-S circuit has two output windings of which only one winding is used in the standard storage configuration: both are used in the expanded-storage operation. The windings are differentiated according to their use for normal or fast-memory operations. A winding is selected by a matrix selector-negative (MSN) package. The operation of this package is similar to the MSP package (figure 6-17), but its output is either 0 or +28.5 volts. Two additional transistor stages are added to the MSN package to achieve correct output voltage polarities.

Two MSN circuits on one package are used for all the WXP-S circuits. When the write circuitry is alerted by a WRITE signal in conjunction with either an FM or NM signal, the center tap of one of the WXP-S output windings is returned to ground through one of the MSN circuits. The other output windings of the WXP-S circuits are blocked by the remaining MSN circuit.

The addition of the MSN package and the differentiation between fast and normal memory permits the same number of MSP packages to be used for expanded storage as is used for standard storage. Considering drum section A in the expanded storage system, 20 sets of read/write heads are required for normal memory operation (one head per track, five tracks per band, totalling 20 bands), and 20 sets are required for fast memory (four heads per track, totalling five bands). However, by logically separating fast and normal memory, only 20 MSP circuits are required for all 40 sets of heads. Since a further logical separation is made between drum sections in expanded storage, the same 20 MSP circuits are used for the entire 9200-word drum.

6-33. READ CIRCUITRY. The read circuits are modified for two reasons. First, the amplitude of the readback signal from the expanded storage drum is less than the readback from the standard storage drum; second, a differentiation is made between fast and normal memory to keep the number of MSP circuits constant. The major modification is to the RAP-S package. This now becomes the RAP-S5 package, denoting the fifth in a series of modified packages. The operation of the RAP-S5 package is similar to the RAP-S package (heading 6-25), but an additional stage of amplification is added and two instead of one input windings are used.

The WPP-S circuit (figure 6-12) is designated (A) in figure 6-19. It is unchanged, and its basic function, to prevent write signals from entering the read circuitry, is unaltered. However, two new WPP-S circuits, designated (B), are added to select the fast or normal memory input windings on the RAP-S5 packages. Electronically, the WPP-S(B) circuits are identical to the WPP-S(A) circuits (heading 6-26). The only difference is that backboard wiring connections are returned to different voltages to determine either +3 or +20-volt output levels. The read circuitry operates when the WPP-S(A) circuit is not inhibited by an INHIBIT READ signal and when one of the WPP-S(B) circuits are alerted by an RDINHNM or an RDINHFM signal. The WPP-S(A) applies −1.5 volts to the centertaps of all of the RAP-S5 input windings. Depending upon which WPP-S(B) circuit is selected, the centertap of one winding on each RAP-S5 package is returned to +3 volts. Thus, only one winding of each RAP-S5 winding pair is forward-biased to −1.5 volts through both a WPP-S(A) and (B) circuit.

6-34. CLOCK CIRCUITRY

Several versions of the clock are used with different processor models. The theoretical analysis of each version is similar to the version described in this manual, and, with the exception of the processor clock used in the system with core storage, they will not be described in this section.

The clock is a linear, high-power, narrow-band amplifier which amplifies the low-level sprocket signal from the storage drum to drive the magnetic amplifiers. In addition to developing power pulses and blocking pulses for the computing circuitry, the clock is the central timing device for the system.

The clock circuitry, shown as a block diagram in figure 6-20 and a schematic in figure 6-21, includes a phase-control system which maintains a fixed phase relationship between the sprocket signals and the clock output and a fast and a slow automatic gain control (AGC) system to maintain a relatively constant-amplitude output voltage.

A 60:1 step-down transformer matches the clock output to the low-impedance magnetic amplifier load. The clock output is distributed by heavy, low-inductance bus bars.

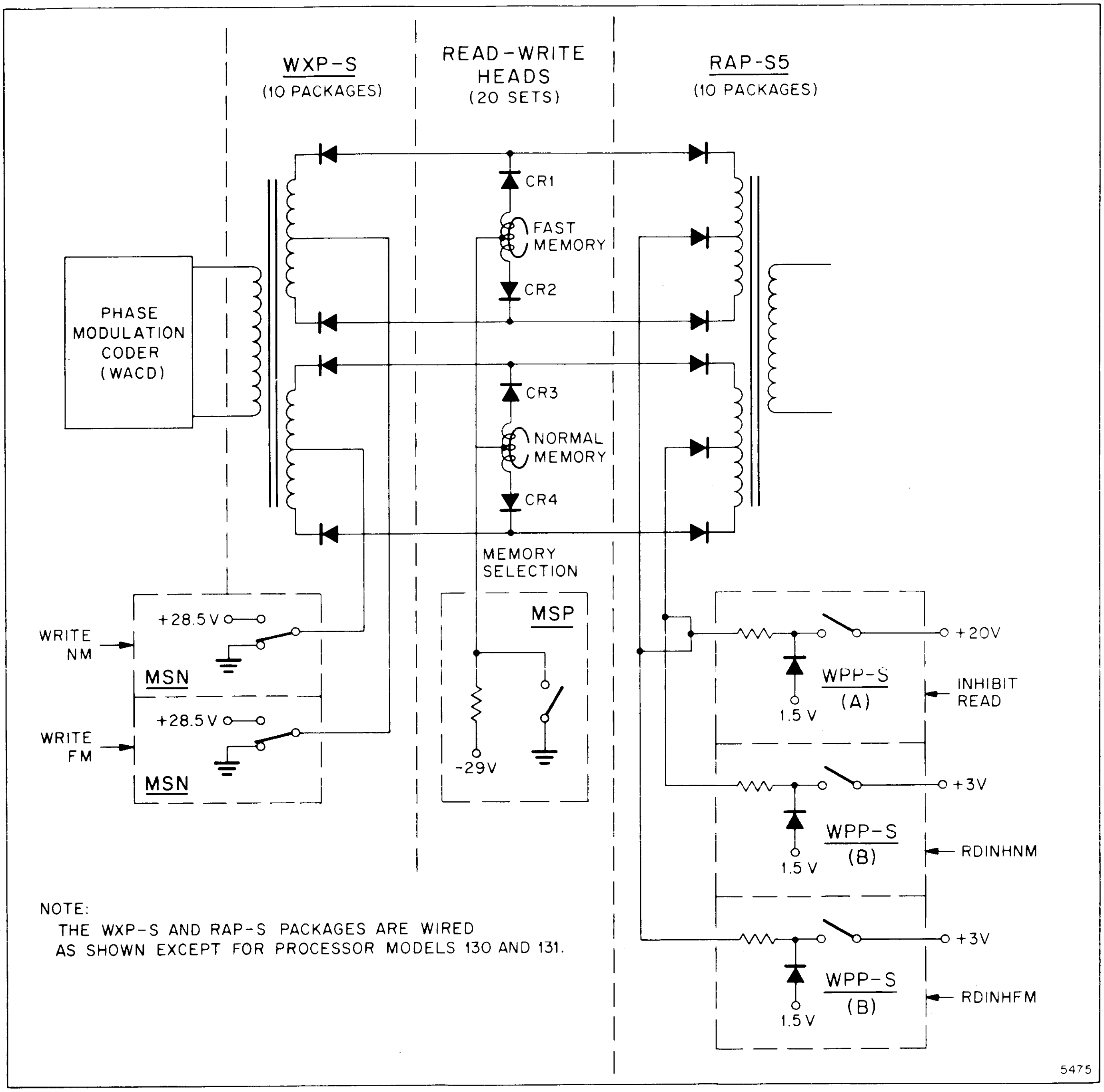


Figure 6-19. Expanded Storage Circuit Interconnections

Physically, the clock consists of two separate units: a driver chassis and an output chassis. The driver chassis contains all of the low-level amplifier circuitry, and the output chassis contains six 4X250B power output tubes with associated tank circuitry.

The general characteristics of the clock are as follows:

(1) Frequency. Both the input and the output sine-wave signals center around 707 kilocycles.

(2) Voltage gain. The clock input is a 0.5-volt peak-to-peak sine wave: the output is a 36-volt peak-to-peak sine wave.

(3) Voltage regulation. The maximum change in load is 70 watts. A 70-watt load change varies the output voltage no more than 10 per cent.

(4) Power gain. The clock increases the low-power sprocket signals to 1 kilowatt.

If more than one clock is used, all clock output signals must be identical in phase and amplitude.

The synchronizer-clock driver stage, located on each clock-driver chassis, supplies the driving signals to another clock. By adjusting this stage, the input to the second clock is made equal in amplitude and in phase to the input to the first clock.

6-35. OPERATION

The block diagram in figure 6-20 shows the type of signal (sine wave, spike, and so forth) and the phase relationships at each stage at a given time. The relative amplitudes or power levels of the clock signals are not shown in this figure.

The two sprocket tracks recorded on the drum are read by four equally spaced heads (two per track) which are connected for read only. The heads are wired in series-parallel, and since the recorded information consists only of a 707 kilocycle signal, the heads theoretically receive identical signals at all times. However, irregularities in the recording medium and variations in the drum-to-head spacing modulate the signals received by the heads. The effect of these modulations is lessened by the series-parallel connection of the four heads, and by using the total induced voltage across the heads as the input to the clock circuitry. Thus a missing pulse in one pulse position on the track would only reduce the clock input signal a small amount below the normal level.

The 0.5-volt peak-to-peak sprocket output is applied to the primary of a 1:3 step-up transformer. The transformer secondary supplies the driving signal to the synchronizer clock-driver stage. The voltage amplifier half of this stage supplies a minimum signal of about 7-volts peak-to-peak to the L-C network in its plate circuit. In the cathode follower stage, the tank circuit provides a variable phase-shift of the signal. Increasing the capacitance of the tank circuit increases the phase angle between the amplifier output and the cathode-follower input. The cathode-follower output is developed at coaxial connector number 1.

If a synchronizer clock is used in conjunction with the processor clock, connector number 1 is connected to the primary of the synchronizer-clock input transformer. Capacitor C7 in the synchronizer clock driver circuit is adjusted to make the phase of the synchronizer clock identical to the processor clock. Resistor R7 is adjusted for an initial sprocket signal amplitude that is identical between the processor and synchronizer clock.

The secondary of the input transformer supplies the a-c signal to the sprocket-voltage amplifier (V1), which develops an output signal of 25-volts peak-to-peak. This signal is r-c coupled to the peaking-core driver (V3), where it is further amplified to drive a tank circuit which utilizes the input winding of a peaking core as part of its inductance. The peaking core saturates near the zero crossing of each sprocket alternation.

To control the input to V3, a 50-kilohm potentiometer (R3) is used for the grid resistor. Too large an input signal grid causes overshoots or damped oscillations in the output. Too small an input signal causes unreliable triggering. Thus resistor R3 is adjusted so that one clean spike is provided to V3 for each sprocket alternation.

A flux change in the primary induces a voltage in the peaking-core secondary. The induced signal is an alternately positive and negative spike with a peak-to-peak amplitude of 20 volts and a pulse width of 0.1 microsecond. Because of the fast-saturating core, the magnitude and phase of the input to the smoothing-tank driver stage are not solely dependent upon the amplitude of the core-driver output. The 1-kilohm resistor across the peaking-core secondary damps out any oscillations or overshoots in the core; thus a clean input is provided to the smoothing-tank amplifier.

Since the bias on the smoothing-tank driver (V4) is approximately −3 volts, the net input signal is approximately +7 volts. The oscillating current in the tank circuit transforms the spiked output of the amplifier into a sine wave with a peak of 125 volts. The smoothing tank maintains a constant input to the V5.

The signal developed across the smoothing tank circuit is coupled to V5 through an attenuator circuit. The drive control (R5) is adjusted so that only a small percentage of the total smoothing-tank voltage is supplied to V5. Thus the V5 and succeeding stages operate in the most linear portions of their characteristic curves. For all practical purposes there is a constant amplitude input to V5. The output of V5 is amplified through one half of V6 and its output is coupled to the phase-splitter driver (V13). The output of V13 is applied to a 1:1 transformer with a single-ended primary and a balanced secondary. The secondary provides two sine-wave outputs equal in amplitude but opposite in phase. Each of these signals is then applied to the push-pull drivers (V14 and V15). Both of the push-pull drivers are biased class-A, so each provides a sine-wave output. The outputs of the drivers are equal in amplitude but opposite in phase.

The push-pull stage in the output consists of two banks of three 4X250B power tubes (V16 through V21) connected in parallel. This stage is biased class-B at −70 volts. The grids are driven to cathode potential, but the tubes draw little or no grid current. The push-pull stage drives a smoothing-tank circuit the output of which is transformer coupled to the processor circuits. The clock output consists of two 36-volt peak-to-peak sine waves of opposite phase.

Associated with the output stage, but physically separate, are four blocking-pulse transformers, a current and voltage monitoring panel, a set of high-voltage power supplies, and a plate overcurrent relay. The overcurrent relay turns off the d-c voltage when the total plate current of V16 through V21 exceeds 1.48 amperes.

6-36. AUTOMATIC GAIN CONTROL — FAST AGC SYSTEM

The fast AGC system uses a difference amplifier to maintain each cycle of the clock output at the same amplitude. If a change in the load causes the amplitude of one cycle of the clock output voltage to vary, the fast AGC system changes the drive on the final stages of the clock until the clock output voltage returns to the required 36-volt peak-to-peak level.

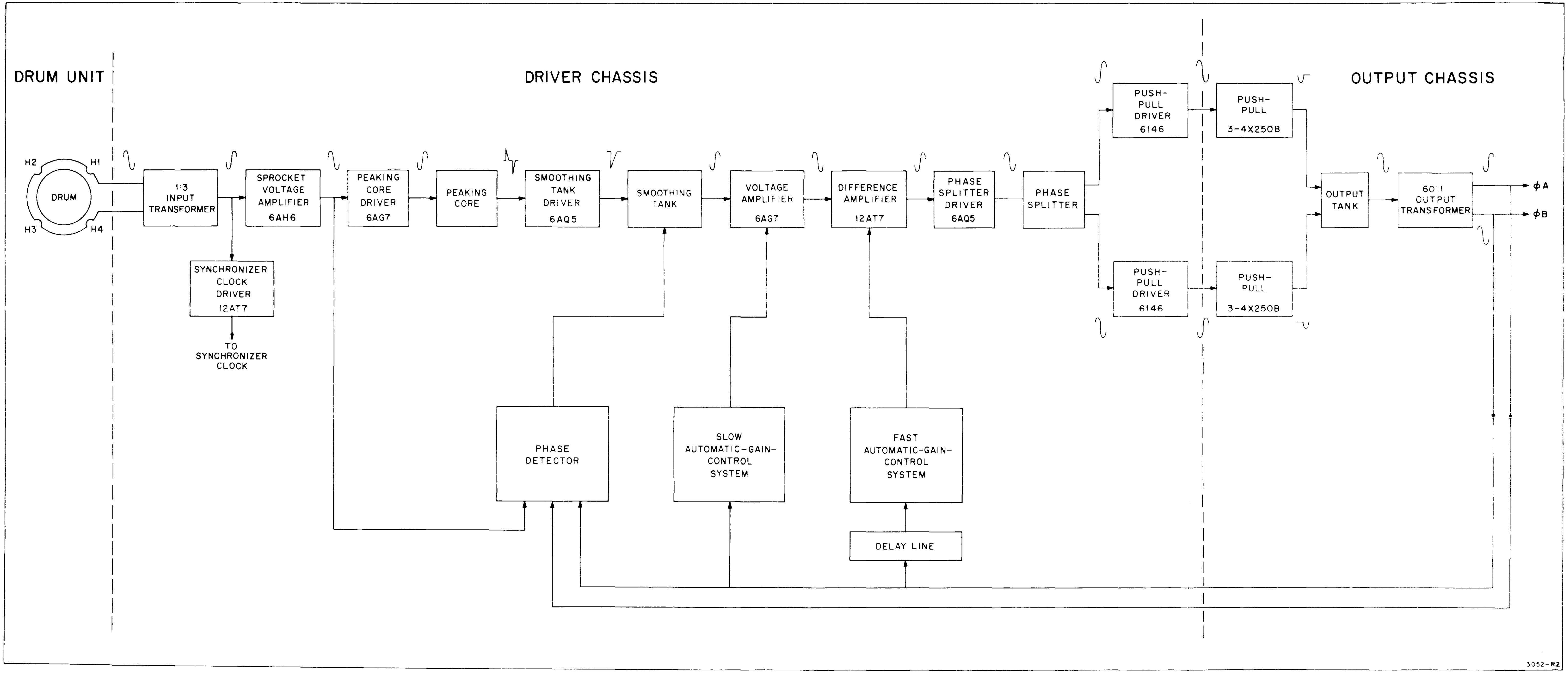


Figure 6-20. Clock System, Block Diagram

6-37. OPERATION. As described under heading 6-35, a constant-amplitude sine-wave signal is applied to V6. A signal is also taken from one side of the secondary of the output transformer and applied through a delay line to pin 7 of the cathode follower portion of V6. The delay line is adjusted so that the signal applied to pin 7 is in phase with the signal on pin 2 of V6. Both halves of V6 use a common 15 kilohm cathode resistor. The signal at pin 2 is relatively constant, but the signal to pin 7 is supplied from the clock output transformer, and it may vary with changing load conditions. Therefore, the grid-to-cathode bias of the amplifier portion is controlled almost wholly by the current through the cathode portion.

If the clock output signal remains normal (approximately 36-volts peak-to-peak), the combined cathode currents from both halves of the tube maintain a reference bias for the amplifier portion. This reference bias causes the sprocket signal to be amplified by an amount that, in turn, always results in a 36-volt peak-to-peak signal from the output chassis.

If the clock output signal decreases, the common cathode current of V6 decreases proportionally. The net result is less degenerative feedback and, consequently, greater sprocket signal amplification. Thus the output chassis is driven harder and the clock output signal voltage is increased until the normal output is reached.

Degenerative feedback increases if the clock output signal voltage is greater than normal. Thus the sprocket signal is amplified less and the output chassis produces a smaller signal until the normal level is reached.

6-38. AUTOMATIC GAIN CONTROL SLOW AGC SYSTEM

The slow AGC circuit is a slow acting amplitude-stabilization circuit. It affects the clock output only if several successive cycles of the clock output voltage drift from their normal amplitude. If an abnormal condition (for example a change in supply voltage, a defective tube, or a component variation) causes the clock output to vary, the slow AGC system adjusts the amount of drive on the last stages of the clock circuitry and returns the clock output voltage to the required 36-volt peak-to-peak level.

6-39. OPERATION. The slow AGC system (figure 6-21) consists of three tubes: a voltage regulator, V9, which maintains 75 volts between its electrodes: an amplifier, V8: and a diode, V7. Four resistors are connected in series across V9. Because of the voltage-regulating action of V9, the total voltage across the resistors is 75 volts, and the voltage across R2 is 75 volts thus clamping the cathode of V8 at –75 volts. The combination R1-C1, R12, R6, and R7 are connected in series between grid and cathode of V8. The opposing voltages across this network determine the bias on the tube. Capacitors C1 and C2 filter any a-c component to provide an almost steady d-c potential between cathode and grid of V8. The sum of the voltages across resistors R8 and R9, connected between grid and cathode, determines the bias on V5.

Either phase of the clock output voltage may be applied to C3. When a positive alternation is applied to C3, V7a conducts. Current flows from the –150-volt supply through R10, R12, R1, and V7a. Capacitor C1 is charged to the value of the voltage drop across R1, and this voltage becomes the effective bias for V8. Several input cycles are required to charge C1 to a constant potential, and, by virtue of a relatively slow discharge time through R1, several cycles are required to reduce the charge on C1. Thus a constant bias is developed for V8, and this bias, changes only if the voltage drop across R1 becomes larger or smaller for several successive input cycles.

During negative input alternations, CR1 conducts and the input signal is bypassed to ground through C2.

Diode V7b operates only when power is initially supplied to the system. During this period C1 is being charged, and the bias on V8 is determined by the voltage-divider tap of R12. Thus V8 is biased to cutoff and its plate potential is 0 volts. If no other voltage were present, V5 would be unbiased and an excessively large clock output signal would result. Turn-on bias control voltage is supplied to V5 through V7b. When power is initially supplied to the system, the 0.5 microfarad capacitor in the cathode circuit of V7b slowly charges through a 4.7 megohm resistor to –29 volts. Diode V7b conducts and a negative bias is supplied to V5 until C1 is fully charged. Tube V8 then conducts and V7b is reverse-biased until system power is removed.

6-40. **Normal Clock Output Voltage.** When the clock output voltage is normal, no change in voltage is desired. A signal of normal amplitude is applied to C3. On the positive alternation of the clock output signal, V7 conducts and maintains the normal voltage across the R1-C1 combination. Since the bias on V8 remains constant, the plate current remains steady and the clock output voltage remains unchanged.

6-41. **Abnormal Clock Output Voltage.** When the clock output voltage is low, the amplitude of the signal applied to C3 is below normal. Since C3 is charged to a potential less than normal, V7 conducts less, and less positive bias is developed across the R1-C1 combination over several cycles. Less positive bias lowers the plate current through V8 and reduces the bias on V5. Tube V5 produces a greater output signal which drives the remaining stages harder and increases the clock output voltage to the required 36-volt peak-to-peak level.

If only one cycle of the clock output voltage is low, the voltage across R1-C1 changes very little because of the filtering action of C1. Several low cycles are required to produce any measurable voltage decrease across R1-C1.

When the clock output is high, a larger positive bias is developed across R1-C1. The plate current through V8 increases, and V5 is thus biased more negative. The output stage is driven less, and the output voltage is reduced accordingly.

6-42. PHASE-CONTROL SYSTEM

The phase-control unit monitors both the A-phase and B-phase clock outputs and compares the phase of these signals with that of the sprocket output signal in a phase-detector circuit. If the clock output voltages shift from their desired relationship, a current-variable inductor (increductor) adjusts the phase shift through the clock so that the output signals and the sprocket signal are again placed in the correct phase relationship. This minimizes storage-unit errors.

The principal factors that produce a change in the phase of the clock signals are as follows:

(1) Changes in drum speed due to variations in line frequency or voltage.

(2) Changes in the clock-supply voltages.

(3) Component drift.

(4) Clock mistuned.

(5) Changes in clock loading.

The phase-control system is a control loop consisting of a phase detector, an amplifier, a clock system, and an increductor. A change in phase between the sprocket reference signal and the clock output signals produces a d-c voltage change at the output of the phase detector. This voltage change is converted to a current change in V10. The current change is then sent through the control coil of the increductor. The variable inductor adjusts the phase shift through the clock so that the phase detector output is at a minimum. The phase detector produces no voltage when the clock outputs are in quadrature with the reference signal; thus the action of the loop is to maintain a phase shift of nearly 90 degrees through the clock. The actual phase shift maintained by the closed loop is 90 degrees minus the off-set or error angle necessary to produce the required correction signal.

6-43. OPERATION. The phase-detection unit consists of three tubes (figure 6-21). Both halves of V12 constitute a phase discriminator, V11 is the stable element, and V10 is a d-c amplifier.

The voltage-regulating action of V11 develops a constant 75 volts across the combination of R4 and R11. Tube V11 also maintains a 75-volt drop across R17, thus clamping the cathode of V10 at –75 volts. The bias on V10 is determined by the summation of the voltages developed across R4, R16, and R15. The two 1000-$\mu\mu$f capacitors filter any a-c components to maintain a steady d-c bias on V10. If the voltage between points A and B of the phase-discriminator output circuit equals zero (no discriminator output), the bias on V10 is determined by R4.

A common reference signal is supplied by the sprocket signal from the plate of V1, and developed across coil L1. Any voltage developed across this coil is common to V12a and b. The plate circuit of V12a includes R13, L1, and R15; the plate circuit of V12b includes R14, L1, and R16.

Resistors R13 and R14 form part of a voltage-divider network across the clock output transformer. When the A phase of the clock output is positive and the B phase is negative, current flows from the B-phase side of the transformer through C8, R18, R14, R17, and C7 to the A-phase side of the transformer. With respect to the common connection (reference point) between R13 and R14, the other end of R14 is negative and the other end of R13 is positive. The voltages across R13 and R14 are always equal in amplitude and 180 degrees out of phase. If the polarity of the A-phase and B-phase clock outputs are reversed, the polarity of the voltages across R13 and R14 is also reversed, but their phase relationship remains 180 degrees out of phase.

It is possible for the discriminator to operate under two conditions other than the normal zero-output condition: with the phase of the clock output advanced, or with the phase retarded.

6-44. **Phase of Clock Output Advanced.** The phase of the clock output may shift ahead of the correct phase relationship with the sprocket (reference) signal because of changes in clock loading during a given program. To correct this condition, the inductive reactance of the second tank circuit is increased. The tank circuit then appears capacitive and the phase of the output signals shifts back until the clock output and the sprocket signal resume the correct phase relationship.

The voltages across R13 and R14 are still equal in amplitude and 180 degrees out of phase, but when the voltages across the resistors are vectorially added to the reference voltage across L1, the voltage drop across V12b is greater than across V12a. Increasing the voltage across V12b increases the plate current. The current through R16 and V12b becomes greater than that through R15 and V12a. The voltage developed across R16 becomes greater than the voltage across R15, causing the cathode of V12a to go negative and thereby increasing the bias on V10. As the bias increases, the plate current and the current through the bias and control windings of the increductor decreases. As the control current decreases, the inductance (and hence the inductive reactance) of the signal winding increases, detuning the second tank circuit and shifting the phase of the clock outputs into the correct phase relationship with the sprocket signal. When the correct phase relationship is attained, no further corrective voltages are developed.

6-45. **Phase of Clock Output Retarded.** When the phase of the clock output lags the correct position in relation to the sprocket (reference) signal, the inductive reactance of the second tank circuit is decreased. The tank circuit then appears inductive, and the phase of the output signals shifts ahead until the clock output and the sprocket signal resume the correct phase relationship.

The voltages across R13 and R14 are still equal in amplitude and 180 degrees out of phase, but when the voltages across the resistors and the reference coil are added vectorially, the voltage drop across V12a is greater than the drop across V12b. Tube V1 draws more plate current than V12b, and the current through R15 and V12a becomes greater than that through R16 and V12b. The voltage across R15 becomes greater than the voltage across R16, and point A of the discriminator output circuit becomes positive with respect to point B. As point A becomes more positive the bias on the V10 decreases, and both the plate current and the current through the bias and control windings of the increductor increase. As the control current increases, the inductance (and hence the inductive reactance) of the signal winding decreases, detuning the second tank circuit and shifting the phase of the clock outputs into the correct phase relationship with the sprocket signal. When the correct phase relationship is attained, no further corrective voltages are developed between points A and B of the detector output circuit.

The sensitivity of the phase detector is approximately 45 millivolts per degree of phase shift. The output voltage has a unique amplitude and polarity for each phase angle between zero and 180 degrees: below zero and above 180 degrees the phase-detector output repeats itself. Thus mistuning could cause the operating point of the detector to move out of the correct operating region, and the phase loop would introduce a phase error instead of correcting one. Under clock alignment procedure, the maintenance manual describes sensing tests that guarantee correct operation of the phase detector.

6-46. **Change of Sprocket Frequency.** If the drum-motor powerline frequency varies, there is a corresponding change in the drum speed, and hence a sprocket frequency change. Even with a variation in sprocket frequency the clock phase shift remains constant at about 90 degrees.

Assume a drum speed that produces a sprocket frequency of 707 kilocycles. Since all three tank circuits (figure 6-21) are tuned exactly to this frequency, they appear resistive, and the clock phase shift is 90 degrees.

Now assume a decrease in power-line frequency. As the sprocket frequency decreases below 707 kilocycles, each tank circuit appears inductive and produces a leading output voltage.

The phase of the clock output signals is now advanced from its normal relationship to the sprocket signal. Closing the phase-control loop while the sprocket frequency is low causes V12b to be driven harder than V12a. The phase-detector output is thus negative and less current flows through the increductor control and bias windings: hence, the inductive reactance of the second tank circuit increases. The resonant point of the second tank circuit then decreases, and the circuit contributes less and less to the total clock phase shift. When the second tank circuit contributes no phase shift, the total clock phase shift is determined by the algebraic addition of the phase shifts produced by tanks 1 and 3 plus the normal 90-degree phase shift introduced by the remaining clock circuitry, but the second tank circuit introduces a lagging phase shift as its resonant frequency continues to decrease. This lagging phase shift cancels the leading phase shift introduced by tanks 1 and 3. The total clock phase shift is restored to 90 degrees, and the signals applied to the phase detector are again in quadrature.

All changes in the resonant point of the second tank circuit are brought about by the changing current in the increductor control and bias windings.

The phase-control system corrects for an increase in sprocket frequency in the same way, but the sign of the phase shift introduced into the tank circuits is opposite to that just described.

6-47. VOLTAGE AND CURRENT MONITORS. The B-supply, screen-grid, and control-grid bias voltages applied to V16 through V21 (figure 6-22) can be monitored at the meter panel by rotating the voltage selector (SW57). By rotating the current selector (SW48), the increductor current, the cathode current of the individual output tubes, and the total plate current of all six output tubes can be measured. Use of the voltage and current monitor selectors is specified in Manual 4, Part 3, Section 2, table 2-2.

6-48. CORE-STORAGE CLOCK VARIATIONS

The core-storage processor clock shown in figure 6-22 has the following circuit variations:

The synchronizer clock-driver stage is removed and the input is derived directly from a cathode follower (not shown). The cathode follower output supplies the sprocket signal for all system clocks.

The fast AGC system is removed, and all gain control is derived from the slow AGC system. Its operation is identical to the other clock versions.

The six 4X250B power tubes (V16 through V21) in the output chassis (figure 6-21) are replaced with two 4X1000B tubes (V16 and V17, figure 6-22). Voltage to the output drivers is increased to 2000 volts, and output power is 1.5 kilowatts.

6-49. POWER CONTROL AND POWER SUPPLIES

6-50. POWER TURN-ON PROCEDURE

Power is turned on by pressing the turn-on bar (DC READY) which actuates the DRUM ON, AC ON, and DC ON switches. The power-control circuits are shown as a block diagram in figure 6-23. The power turn-on operation takes place in three steps.

When the turn-on bar is first perssed, the blowers in all units turn on. The turn-on bar should be held down a few seconds until the blowers reach normal operating speed. At this time the storage-drum airflow switches close, the DRUM ON indicator illuminates, and the motor-generator set which drives the drum motor is energized (figure 6-23). To provide sufficient time for the drum to reach its normal operating speed, a 12-minute delay is actuated for main power.

After the DRUM ON indicator illuminates, press the turn-on bar a second time. Pressing the AC ON pushbutton check 1,3 - repeat 1 2 and 3 till no improvement - then 4 - then 5 and 6 repeat 5

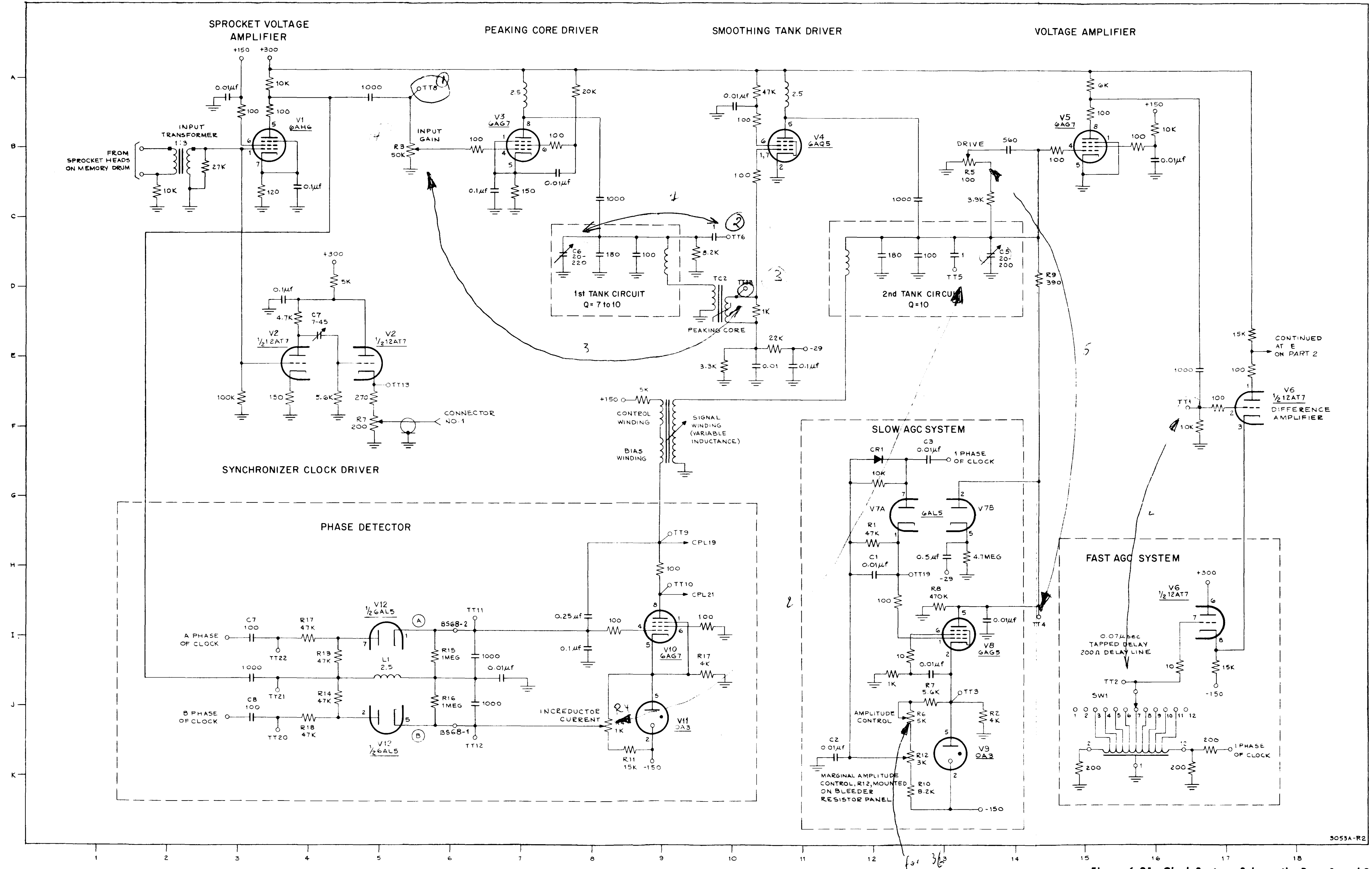
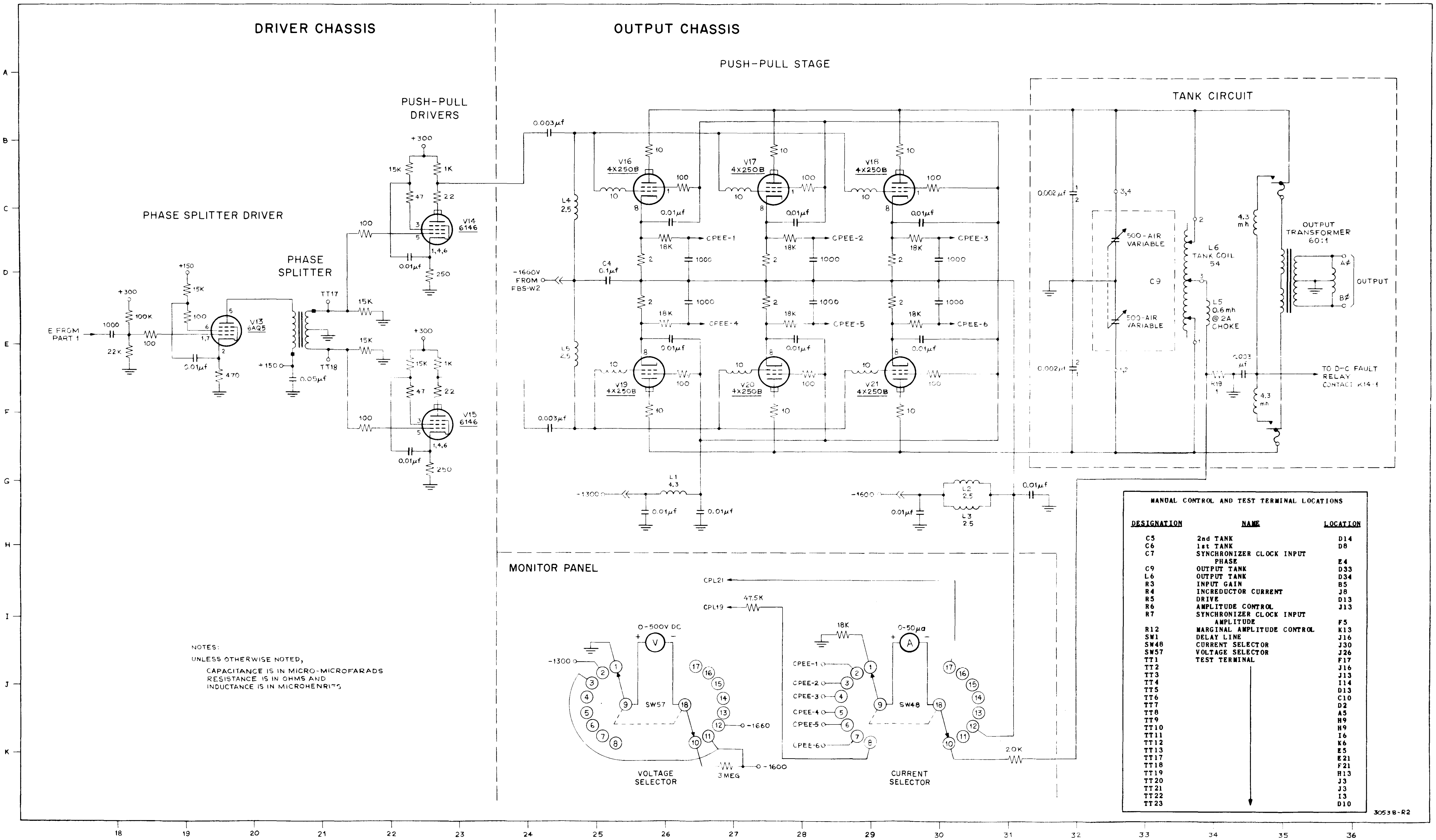


Figure 6-21. Clock System, Schematic, Parts 1 and 2

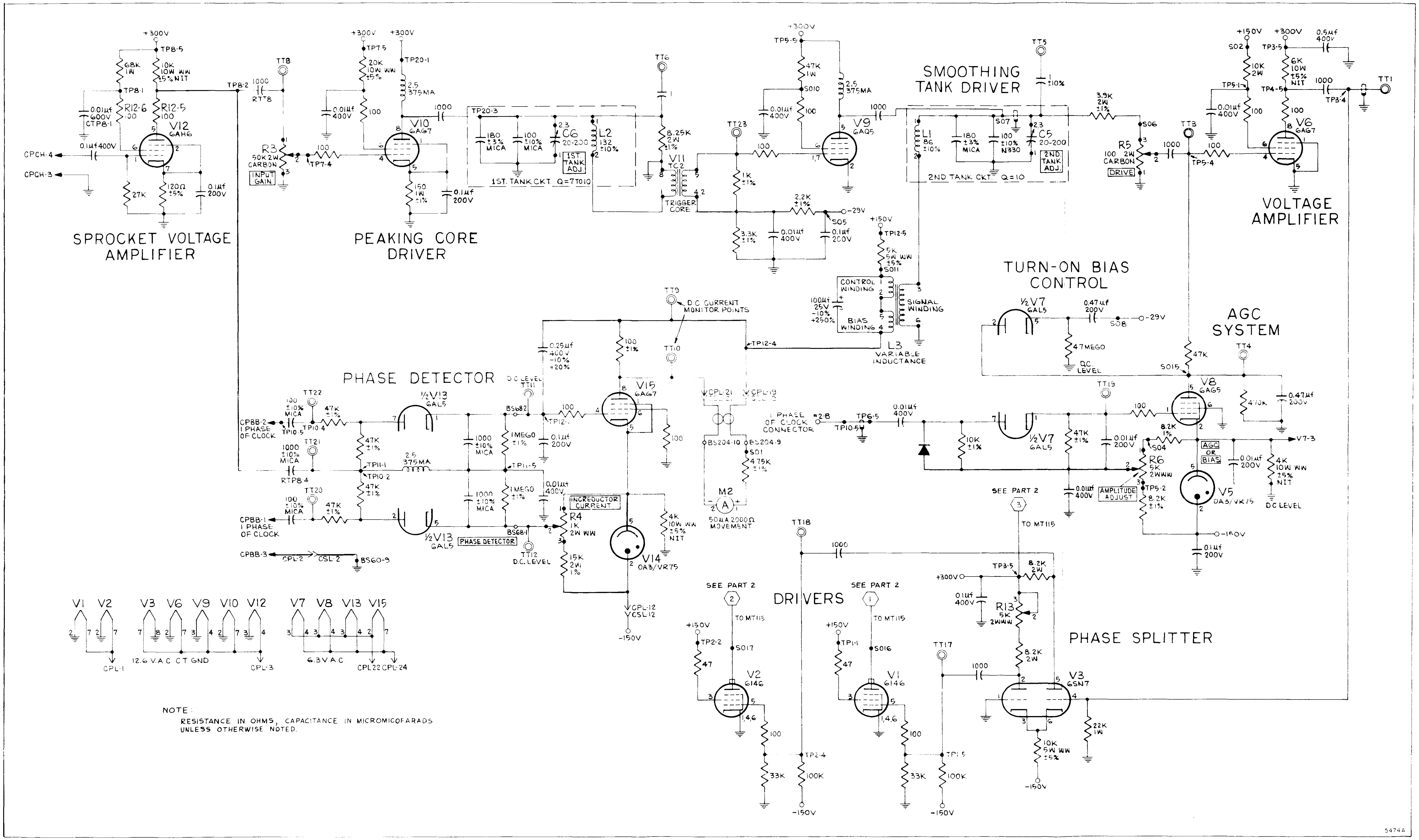


NOTES:
UNLESS OTHERWISE NOTED,
CAPACITANCE IS IN MICRO-MICROFARADS
RESISTANCE IS IN OHMS AND
INDUCTANCE IS IN MICROHENRIES

MANUAL CONTROL AND TEST TERMINAL LOCATIONS

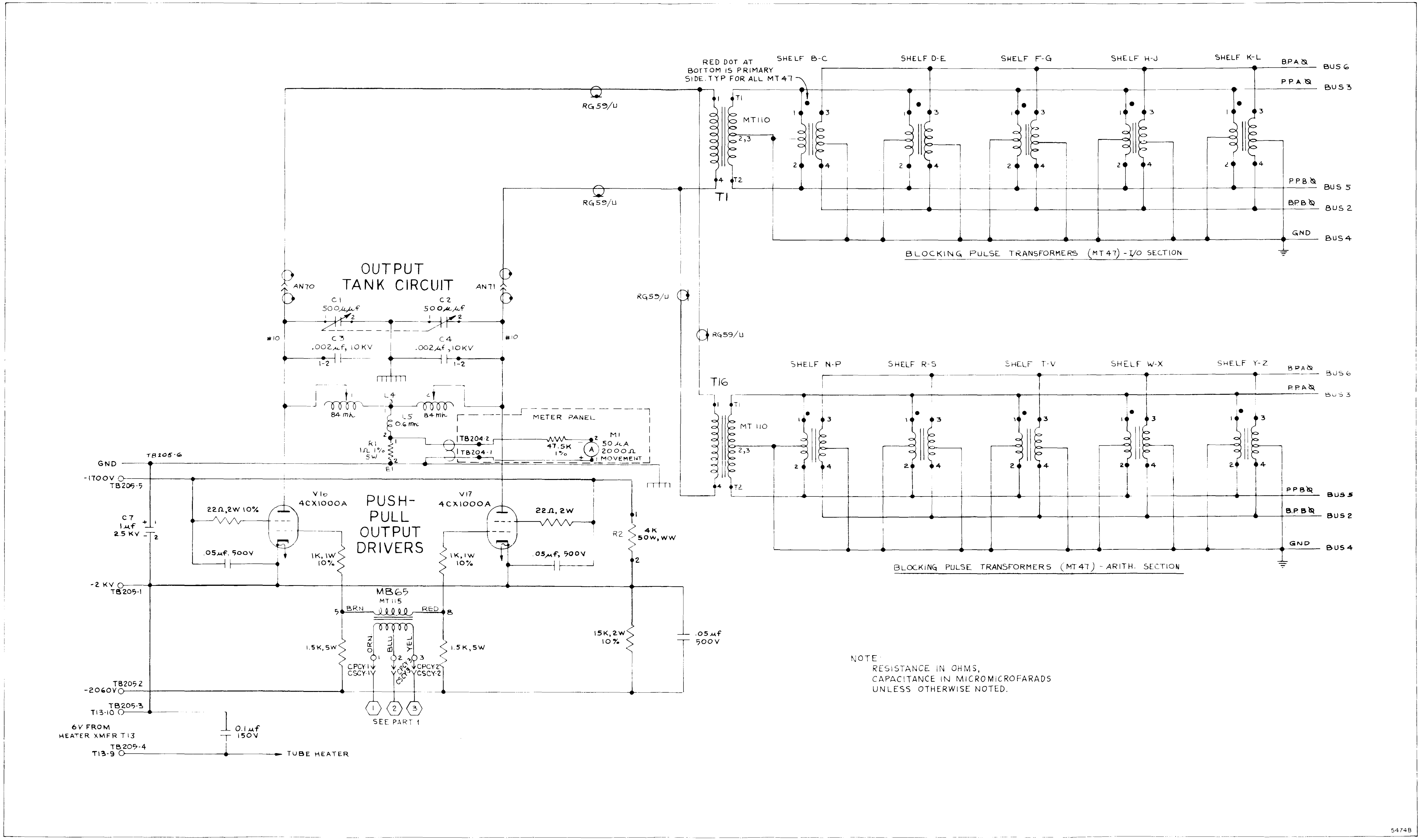
DESIGNATION	NAME	LOCATION
C5	2nd TANK	D14
C6	1st TANK	D8
C7	SYNCHRONIZER CLOCK INPUT PHASE	E4
C9	OUTPUT TANK	D33
L6	OUTPUT TANK	D34
R3	INPUT GAIN	B5
R4	INCREDUCTOR CURRENT	J8
R5	DRIVE	D13
R6	AMPLITUDE CONTROL	J13
R7	SYNCHRONIZER CLOCK INPUT AMPLITUDE	F5
R12	MARGINAL AMPLITUDE CONTROL	K13
SW1	DELAY LINE	J16
SW48	CURRENT SELECTOR	J30
SW57	VOLTAGE SELECTOR	J26
TT1	TEST TERMINAL	F17
TT2		J16
TT3		J13
TT4		I14
TT5		D13
TT6		C10
TT7		D2
TT8		A5
TT9		H9
TT10		H9
TT11		I6
TT12		K6
TT13		E5
TT17		E21
TT18		F21
TT19		H13
TT20		J3
TT21		J3
TT22		I3
TT23		D10

Figure 6-21. Clock System, Schematic, Parts 1 and 2 (cont)



NOTE:
RESISTANCE IN OHMS, CAPACITANCE IN MICROMICOFARADS UNLESS OTHERWISE NOTED.

Figure 6-22. Clock Variations For Processor With Core Storage, Parts 1 and 2



NOTE:
RESISTANCE IN OHMS,
CAPACITANCE IN MICROMICROFARADS
UNLESS OTHERWISE NOTED.

Figure 6-22. Clock Variations For Processor With Core Storage, Parts 1 and 2 (cont)

provides power to the drive motors in all units. In addition, the filament transformers are energized and the AC ON indicator illuminates. A delay of 1½ minutes is actuated to provide sufficient warm-up time for the filaments before d-c is applied to the tubes. Twelve minutes after the turn-on bar is initially pressed, signals from the 12-minute and the 1½-minute delay are gated through a series of relay contacts to illuminate the DC READY indicator in the turn-on bar.

After the DC READY indicator illuminates, press the turn-on bar a third time. The a-c power is then applied to the d-c power supplies through the DC ON switch and the DC ON indicator illuminates, indicating d-c power is available to all units.

Although a-c is available to all units the second time the turn-on bar is pressed, the drive motors in the reader, printer, or punch must be turned on manually.

6-51. A-C DISTRIBUTION

The a-c distribution to all units (except the marginal-check fuse-alarm circuits which receive a-c through the main-line switch) is initiated during the power turn-on operation. Figure 6-24 shows the a-c distribution circuits.

When the turn-on bar is initially pressed, the relay 4 contacts close and energize the blowers. With the exception of the processor blowers, all blowers are connected across a 120-volt source and ACN. The 120-volt source is one side of the 240-volt, single-phase, 3-wire line. The processor blowers receive 240 volts, since they are connected across the two outside lines.

The relay 6 contacts also close when the turn-on bar is initially pressed and energize the motor of the motor-generator set. The motor is a 60-cycle unit, but the generator produces a 300-cycle output to drive the drum motor. This higher frequency is required because the drum motor speed is approximately 18,000 rpm. Five seconds after the motor-generator set is energized, the relay 34 contacts close, placing the drum-motor load on the generator. Five seconds are required for the generator output voltage to reach normal operating level. With the relay contacts in the position shown in figure 6-24, the drum-motor current flows through the starting fuses. In 12 minutes the drum motor attains normal operating speed, and the relay 8 contacts switch. The drum-motor current then flows through the run fuses.

The second time the turn-on bar is pressed, the relay 11 contacts close and energize the clock filament transformer, the RPU relay supply, and the marginal-check cabinet. Although 120 volts are available to the motors in the card reader and read punch, the motors must be started manually. The HSP filament transformers receive 240 volts, since they are connected across both sides of the line through relay 11.

The third time the turn-on bar is pressed, the relay 12 contacts close and supply a-c to the power supplies. A relay in the primary winding of the printer-actuator supply delays a-c to the printer-actuator supply. The delay provides adequate time to establish the bias voltages before the development of the high d-c potential.

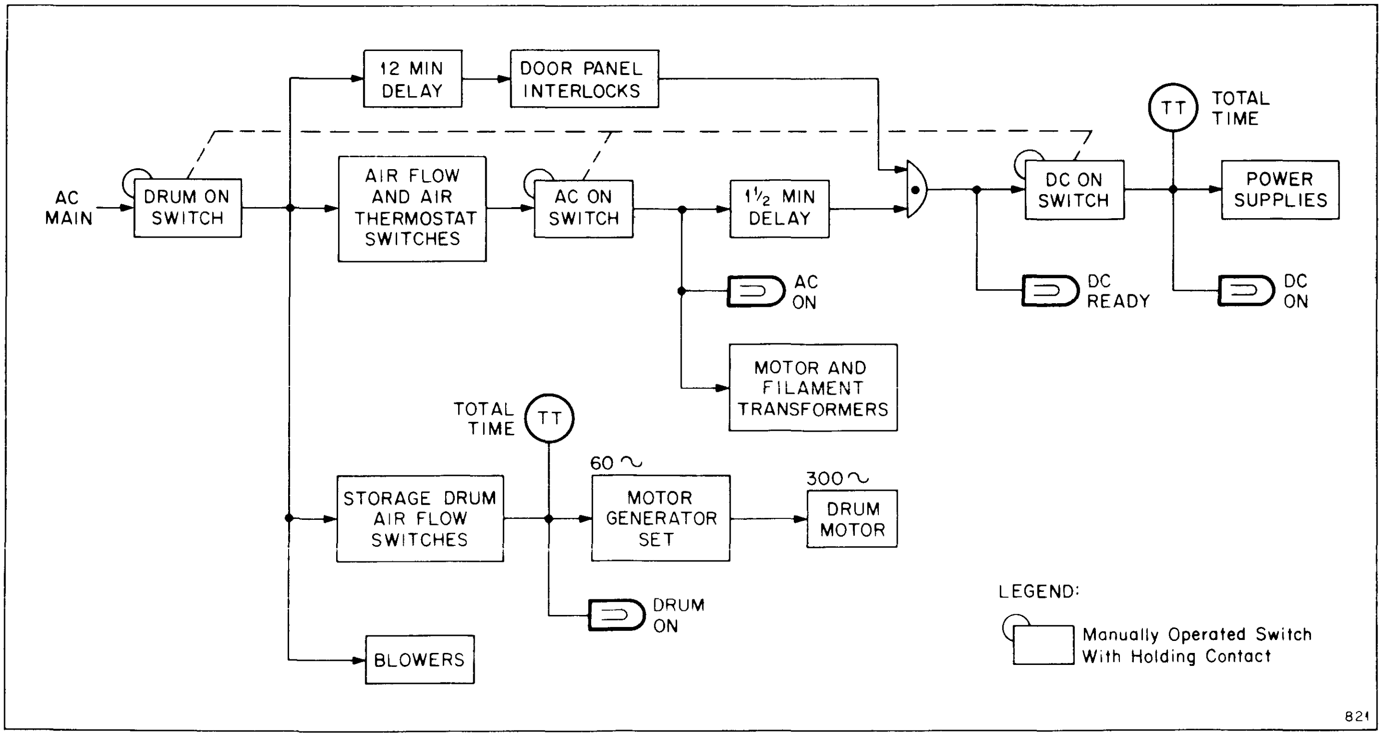


Figure 6-23. Power Control Circuits, Block Diagram

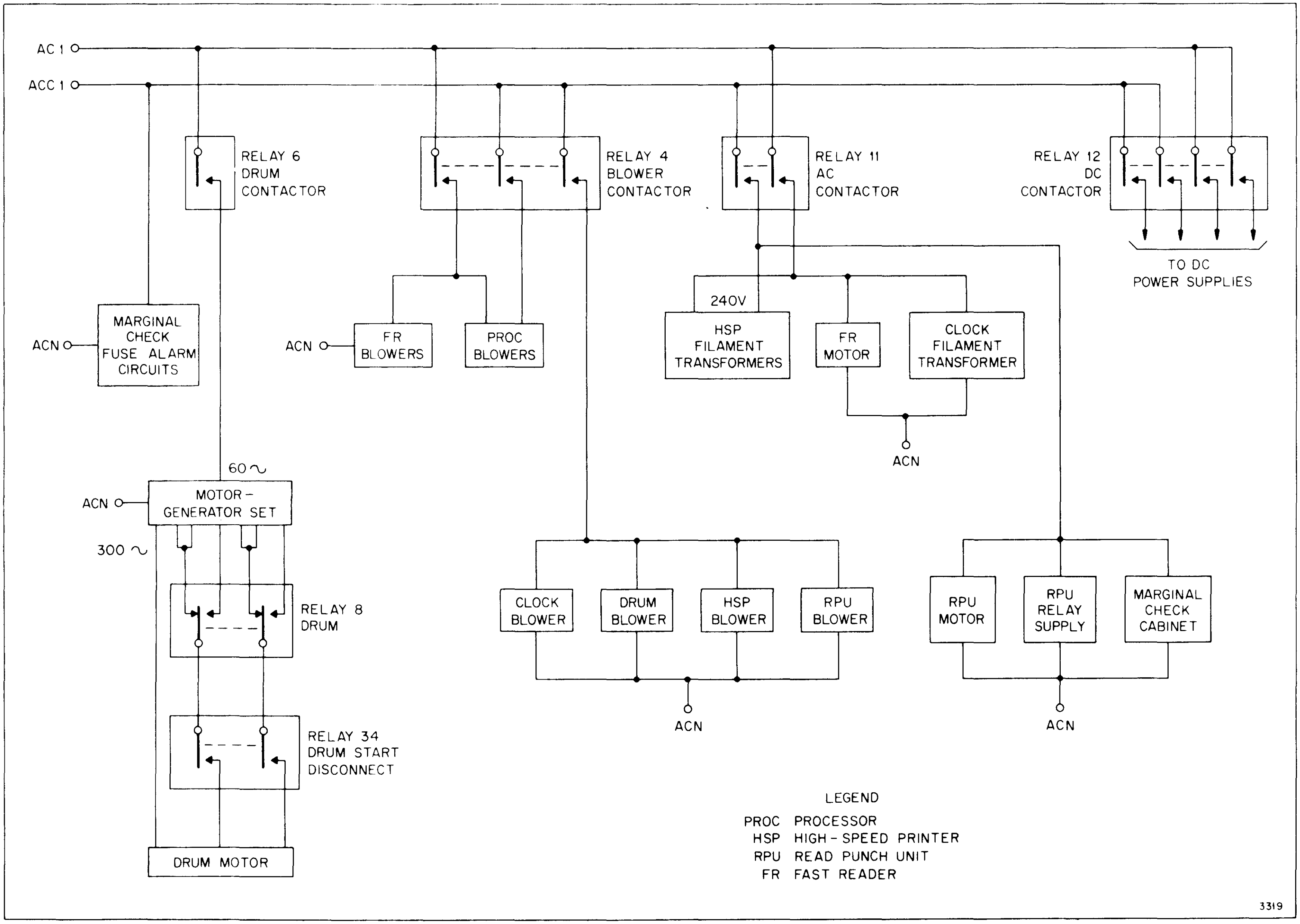


Figure 6-24. A-C Distribution Circuits, Block Diagram

6-52. POWER-CONTROL CIRCUITS

The following description and figure 6-25 explain how the blowers, drum, a-c, and d-c are turned on. When the turn-on bar is pressed, the DRUM ON, AC ON, and DC ON switches are all actuated simultaneously.

When the turn-on bar is initially pressed, relay 4 is energized through the DRUM ON switch. The blower motors are then energized through contacts 1, 2, and 3 of relay 4. A circuit is also completed through thermal-delay relay 3, which has a drop-out time of 2 minutes. Relay 4 and the blower motors remain energized through the K3-1 contacts for 2 minutes after the system is turned off.

When the blowers reach normal operating speed, all the airflow switches operate. Relay 36 is then energized through the DRUM ON and storage-drum airflow switches. Relay 37 is also energized through the DRUM ON switch and the series of airflow and air-temperature switches. The K37-1 contacts require 8 seconds to close. When the K36-1 contacts close, relays 6 and 34, the total time indicator, and the DRUM ON indicator energize through the normally closed DRUM OFF switch, K36-1, K15-1 normally closed, and K33-3 normally closed.

The motor-generator set is energized through the K6-1 contacts, and in 5 seconds the drum motor is energized when the K34-1 contacts close. The K6-2 contacts hold relay 4 energized and the K6-3 contacts short circuit the normally open DRUM ON switch.

Relay 7 is energized when the DRUM ON switch is first closed, but the K7-1 contacts do not switch for 12 minutes. This allows the drum to attain normal operating speed.

Relays 36 and 37 have 8- and 12-second dropouts respectively. These delayed dropouts prevent the system from going off the line due to a momentary interruption in the air flow. If the air flow is interrupted or if the temperature rises above a predetermined point, one or more of the airflow or air-temperature switches operates, illuminating the appropriate indicator lamp and turning off a-c power or drum power.

Eight seconds after the turn-on bar is initially pressed, the K37-1 contacts close and the turn-on bar is pressed a second time. Relay 11 and the AC ON indicator energize through the AC ON and AC OFF switches, K16-1 normally closed, and K37-1. The K11-1 and 2 contacts energize the filament transformers, and the K11-3 contacts short circuit the AC ON switch. A circuit is also completed through the AC ON switch, K10-2 normally closed, and relay 9, but the relay 9 contacts do not switch for 1½ minutes. This delay provides time for the filaments to warm up before the application of d-c.

When the K9-1 contacts switch, a circuit is completed through K11-3, the AC OFF switch, K9-1, and relay 10. The K10-2 contacts open relay 9 and hold relay 10 closed. Twelve minutes after the turn-on bar is initially pressed, the K7-1 contacts switch and energize relay 8. Power to the drum motor is switched from the start fuses to the run fuses by way of contacts 2 and 3 of relay 8. A circuit is also completed through K7-1, the normally closed contacts of the three over-current relays, the normally closed contact of the processor-interlock switch, K10-1, R22, and the three DC READY indicators in series.

When the DC READY indicator illuminates, the turn-on bar is pressed for the third time. Relay 12 and the total-time indicator are energized through K7-1 normally open contacts, the overcurrent relays, the processor-interlock switch, K10-1 normally open contacts, the DC ON normally open contact, the DC OFF switch, K17-1 normally closed contacts, and K18-1 normally closed contacts. The DC ON indicator illuminates through R23. The power supplies are energized through contacts 1, 2, 3, and 4 of relay K12. The K12-5 and K32-1 contacts short circuit the normally open DC ON switch. Unless the clock output is sufficient to hold relay 32 energized through CR4 and CR5, the K32-1 contacts open and turn off d-c. During clock alignment this circuit can be short circuited by pressing the CLOCK ALARM BYPASS pushbutton.

A positive potential must be developed to check the negative d-c fuse-fault circuits, and a negative potential to check the positive d-c fuse-fault circuits. When a-c is applied to the half-wave rectifiers CR2 and CR3 through K32-1, K12-3, and K13-1, negative and positive potentials are developed across C5 and C6 respectively.

6-53. DRUM ALARM CIRCUITS

6-54. POWER SUPPLY. The drum alarm circuits use d-c microameters, with holding coils, as indicators. A half-wave rectifier circuit supplies the d-c current to the holding coils. See figure 6-25.

Diode CR1 makes up the rectifying unit, and C4 is the filater capacitor across which dc is developed. Bleeder resistor R16 discharges C4 when the system is turned off. Power is normally supplied to the unit through the normally closed contacts of K33-1 and K6-2. If power is manually turned off, the K6-2 contacts open and deenergize the unit. If power is turned off automatically because of a drum alarm, power is supplied to the unit through the normally open contacts of K33-1 and the fault indication is retained until it is cleared manually.

6-55. HEAD-SPACING DETECTOR. The continuously operating head-spacing detector circuit detects an indicates any change in the normal spacing between the heads and the drum. If this spacing changes appreciably, the system turns off.

The four heads, connected in parallel, are positioned over a permanently recorded timing band. If the drum speed is constant, the amplitude of the sine wave induced in the head windings is a function of the coupling between the heads and the drum surface. As the head-to-drum spacing decreases, the amplitude of the induced voltage increases. If the spacing increases, the voltage decreases.

The induced voltage is transformer-coupled to a half-wave rectifier circuit by T1 (figure 6-25). Transistor Q1 is connected as a conventional diode, and dc is developed across C2. A meter is connected across C2 which indicates when the drum is rotating even though no other power is being supplied.

Assume that the head-to-drum spacing increases above the desirable limit. The voltage induced in the heads and the voltage across C2 decreases, causing the indicator to move counterclockwise. The indicator arm closes the LO contacts and completes a circuit from the negative side of C4 through the normally closed TEST AND CLEAR switch, the hold coil, the LO contacts, K8-1 normally open contacts, relay 33, and K14 to the positive side of C4. The hold coil keeps the LO contacts closed until the TEST AND CLEAR switch is opened manually.

The DRUM indicator is illuminated through K33-1 and 2. The K33-1 normally open contacts bypass all control switches and maintain dc across C4 even though power turns off. The K33-4 contacts bypass the K8-1 contacts and hold relay 33 energized even though power is off. When power is first turned on, the K8-1 contacts hold the circuit inoperative until the drum attains normal operating speed.

The filter circuit consisting of C1, C2, L1, and L2 not only provides a steady d-c output, but also absorbs any inductive surges in the meter movement when the hold coil is deenergized. This prevents any unwanted erasures at the heads.

6-56. STATOR TEMPERATURE DETECTOR. This circuit measures and indicates the temperature of the drum-motor stator windings and turns off the power if the temperature becomes too high.

The detector unit is a thermocouple, the output of which varies as a function of the stator temperature. As the temperature increases, the output increases and the indicator moves clockwise. When a predetermined temperature is reached, the indicator arm closes the contacts and completes a circuit from the negative side of C4 through the normally closed TEST AND CLEAR switch, the hold coil, meter contacts, K8-1, relay 33, and R14 to the positive side of C4. Relay 33 initiates a power turn off and a DRUM indication, while the hold coil keeps the meter contacts closed and relay 33 energized until the TEST AND CLEAR switch is opened manually.

6-57. BEARING TEMPERATURE RISE DETECTORS. Two temperature-rise detector circuits, one for the drum bearings and the other for the drum motor bearings, function in the same manner. Since expansion is the physical consideration, the difference between the bearing and ambient temperatures is measured and indicated. When the temperature difference reaches a predetermined limit, power is turned off.

Two series-opposing thermocouples are connected in each circuit. When the drum is inoperative, the bearing and ambient temperatures are equal and the indicator reads zero. When the drum is in operation, the bearing temperature exceeds the ambient temperature, and the indicator moves clockwise. The amount of indicator movement is a function of the temperature difference.

When the temperature difference is unsafe, the indicator closes the meter contacts and completes a circuit from the negative side of C4 through the normally closed TEST AND CLEAR switch, the hold coil, the meter contacts, relay 33, and R14 to the positive side of C4. Relay 33 initiates a power turn off and a DRUM indication, while the hold coil keeps the meter contacts closed and relay 33 energized until the TEST AND CLEAR switch is opened manually.

6-58. POWER TURN-OFF PROCEDURE

To turn off power, proceed as follows:

(1) Press the DC OFF pushbutton. Relays 12 and 30 (figure 6-25) open and disable the power supplies and the write circuits.

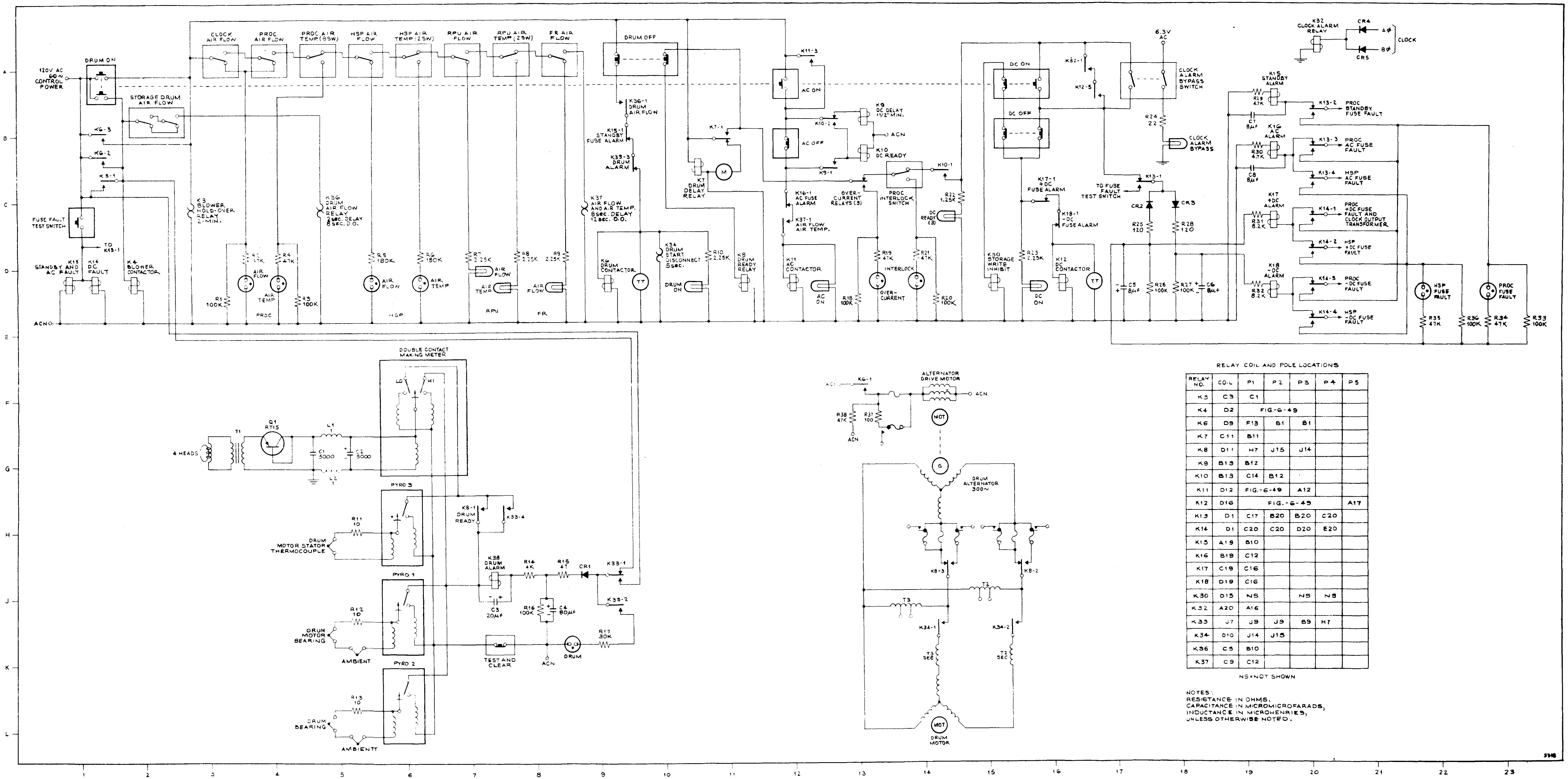
(2) Press the AC OFF pushbutton. Relay 11 opens and deenergizes the filament transformers and the motors of the reader and punch unit.

(3) Press the DRUM OFF pushbutton. Relays 6 and 8 open and deenergize the drum-motor generator set, the drum-alarm circuits, and all blower motors.

To turn off only dc, press the DC OFF pushbutton. To disable both ac and dc, press both the DC OFF and the AC OFF pushbuttons. Although it is possible to turn off both ac and dc by pressing only the AC OFF pushbutton, or to turn off the entire system by pressing only the DRUM OFF pushbutton, these two procedures are not recommended because of the possibility of undesired drum erasures. When the DC OFF pushbutton is pressed first, the write circuits are disabled by relay 30 and erasures are prohibited.

6-59. POWER SUPPLIES. The power supplies are designed to maintain an almost constant d-c output under widely varying input and load conditions. Figure 6-26 shows in block form the system d-c power supplies. The a-c inputs to the power are first applied to a voltage-stabilizing transformer (VST) which, in turn, supplies a steady a-c input to the rectifiers. Although the rectifying and filtering stages differ in construction from one power supply to another, they are of conventional design and their operation requires no explanation.

6-60. VOLTAGE-STABILIZING TRANSFORMER. The voltage-stabilizing transformer circuit is a phase-shifting network made up of a linear inductance and the parallel combination of a fixed capacitance and a nonlinear inductance. The net combination of capacitance and nonlinear inductance is more or less capacitive, depending upon the level of the input line voltage. If the line voltage increases, the fixed capacitor current increases linearly, while the nonlinear inductor current increases exponentially. Thus the net voltage across the capacitor, and hence the output voltage, decreases. Although the output voltage decreases relative to the input, it remains constant relative to the load.



RELAY COIL AND POLE LOCATIONS

RELAY NO.	COIL	P1	P2	P3	P4	P5
K3	C3	C1				
K4	D2	FIG.-6-49				
K6	D9	F13	B1	B1		
K7	C11	B11				
K8	D11	H7	J15	J14		
K9	B13	B12				
K10	B13	C14	B12	·		
K11	D12	FIG.-6-49		A12		
K12	D16	FIG.-6-49				A17
K13	D1	C17	B20	B20	C20	
K14	D1	C20	C20	D20	E20	
K15	A19	B10				
K16	B19	C12				
K17	C19	C16				
K18	D19	C16				
K30	D15	NS		NS	NS	
K32	A20	A16				
K33	J7	J9	J9	B9	H7	
K34	D10	J14	J15			
K36	C5	B10				
K37	C9	C12				

NS=NOT SHOWN

NOTES:
RESISTANCE IN OHMS,
CAPACITANCE IN MICROMICROFARADS,
INDUCTANCE IN MICROHENRIES,
UNLESS OTHERWISE NOTED.

Figure 6-25. Power Control Circuits, Schematic Diagram

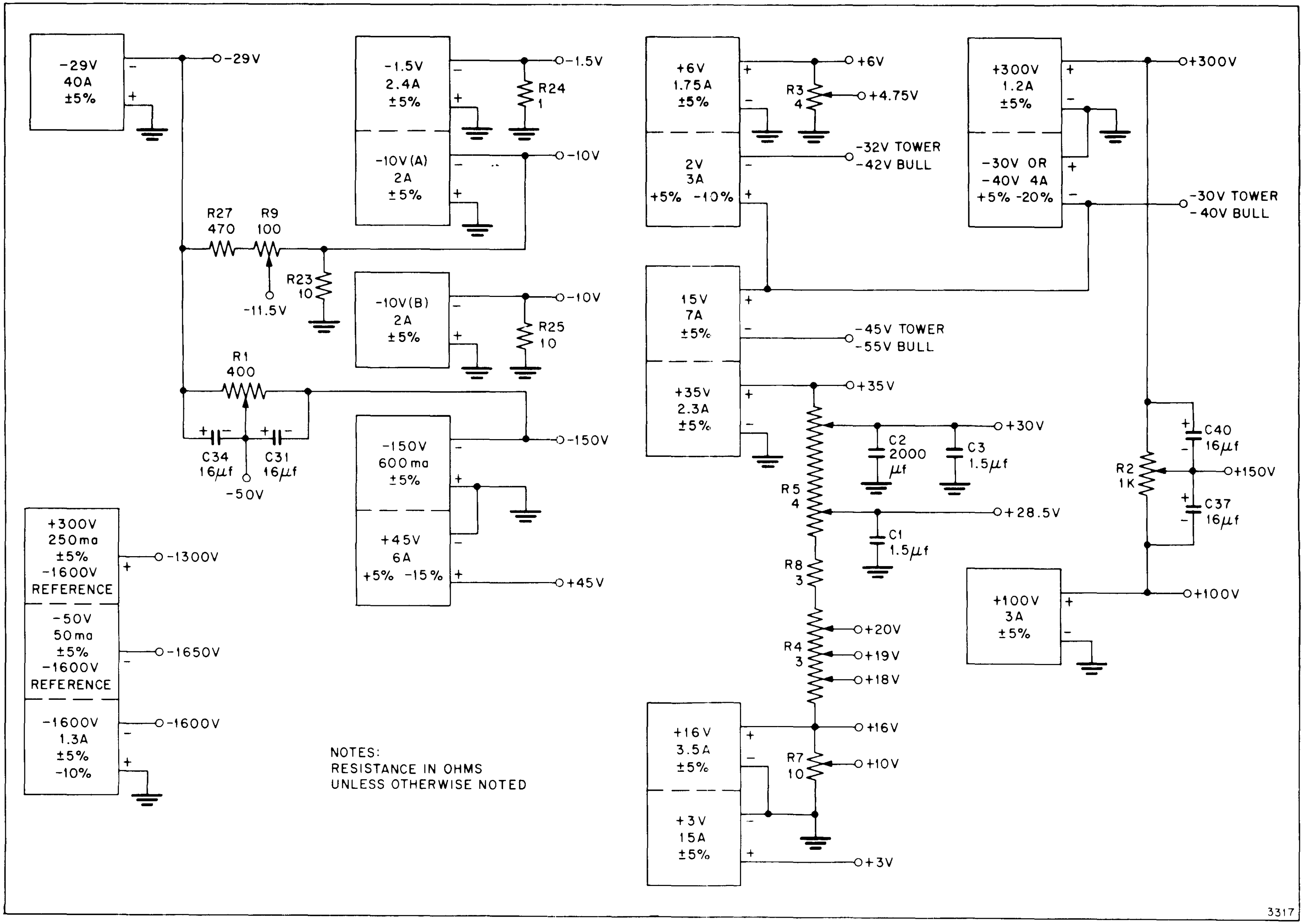


Figure 6-26. D-C Power Supplies, Block Diagram

If the input line voltage decreases. the current through the nonlinear inductance decreases at a faster rate than through the capacitor. The net voltage across the capacitor is thus larger. and the output voltage increases relative to the input.

The individual inductances and capacitance for the VST are as shown in figure 6-27. This figure shows typical VST use with processor power supplies. Leakage inductance between the primary and secondary provides the linear inductance. The secondary winding and core section provide the nonlinear inductance. A compensating winding improves stabilization.

6-61. POWER-SUPPLY INTERCONNECTIONS. Many of the voltages required by the system are developed in single power-supply units and applied directly to a load. However, many more voltages are obtained from interconnections among the various power supplies. as shown in figure 6-26.

The –15-volt and the –30-volt supplies are connected in series to develop output voltages of –30 and –45 volts. The 2-volt supply is connected in series with the –30-volt supply to provide –32 volts. If the bull punch is to be used instead of the tower punch. the –30-volt supply is replaced by a –40-volt supply and the preceding voltages are increased by 10 volts to –40. –42. and –55 volts.

Although both the +35-volt and +16-volt supplies are connected directly across a load. they are connected together through R5. R8. and R4. Intermediate voltages between +16 and +35 volts may then be obtained by positioning the movable arms of R4 and R5. Voltages between 0 and +16 volts are available across R7. The movable arm which is set at the +10-volt level makes this voltage available for use throughout the system. The +3-volt level is developed in a separate supply because of the high current requirements at this voltage.

Other power supplies such as the –29-volt and 150-volt supplies are connected in a similar manner but fewer connections are provided for obtaining intermediate voltages. Capacitors C31 and C34 bypass noise impulses around R1 and prevent fluctuations in the –50-volt level.

In the high-voltage supplies. 1600 volts is taken as the reference level. The 1660-volt level is obtained by connecting a –60-volt supply and the 1600-volt supply in series. The –1300-volt level is obtained by connecting a +300-volt supply and the –1600-volt supply in series opposing.

6-62. VOLTAGE MONITOR

The system is equipped wtih a centralized voltage-monitoring system. figure 6-28. by which all the d-c voltage levels and both phases of the clock output can be monitored with a single meter. Since all the voltages have a small allowable percentage of deviation. the meter measures only the deviations.

Since the meter contains a d-c movement. the clock output must be rectified before it can be monitored. To monitor the A-phase clock output. for example. switch 8 is set to + and switch 7 is set to position V. The A-phase clock output is rectified by CR1 and applied to the choke input filter consisting of L1 and C1. Diode CR2 shunts any leakage current through CR1 to ground. The multiplier resistance is composed of R1 and part of R3. (With a normal clock output of 36-volts peak-to-peak. R3 is adjusted so that the meter indicates a deviation of zero.) Current flows from ground through terminal 18 deck B. the meter movement and R18. terminal 81 deck A. the movable contact on R3. and R1 to the positive side of C1. If the clock output is normal. the reading is center scale. If the output is off normal. the reading indicates the deviation in percentage off normal.

To monitor a given voltage. selector switches 7 and 8 must be set as indicated on the chart in figure 6-28. To monitor the +3-volt supply. for example. the POLARITY SELECTOR. switch 8. is set to + and the VOLTAGE SELECTOR. switch 7. is set to position A. This setting places a 4.39 kilohm multipler (R30) in series with the meter movement. Current then flows from ground through terminal 1 deck B. the meter movement and R18. terminal 1 deck A. and R30 to the +3-volt supply. If the output of the +3-volt supply is exactly +3 volts. just enough current flows through the meter movement to give a center scale reading.

If output of the 3-volt supply rises to +3.15 volts. additional current flows through the meter movement and the meter reads +5 per cent. If the output falls to +2.85 volts. less current flows through meter and the meter reads –5 per cent. The maximum allowable deviation for the +3-volt supply is ±5 per cent. If the monitor indicates a deviation greater than ±5 per cent, examine the power-supply circuit to determine the cause of the abnormal reading.

6-63. FUSE FAULT CURRENT OVERLOAD CIRCUITRY

Most computer circuits employ indicator (grasshopper) fuses. If the circuit current requirements are low. a single indicator fuse is used. If the current requirements are high. a heavy cartridge fuse is placed in parallel with a smaller indicator fuse. If an overload occurs. both fuses open simultaneously. The cartridge fuse opens the current line to the faulty circuit. and the indicator fuse illuminates to signify a fuse fault on the engineer's panel. When a fuse fault occurs. the system power is turned off either partially or completely. depending upon which circuit contains the open fuse.

If a fuse fault occurs in the processor blower circuits. the standby alarm. relay 15. is energized through the indicator. On positive alternations of the supply voltage. current flows from ACN through R29. K15. the normally closed contacts of K13-2. CR1. and the indicator to the positive ACC1. Capacitor C7 maintains an almost steady current through relay 15. The K15-1 contacts open the drum contactor. relay 6. and initiate the turn off of all power.

If a fuse fault occurs in the 1600-volt supply circuit. the a-c alarm. relay 16. is energized through the indicator.

On positive alternations of the supply voltage, current flows from ACN through R30, K16, the normally closed contacts of K13-3, CR4, and the indicator to the positive ACC1. The K16-1 contacts open the a-c contactor, relay 11, and initiate the turn off of ac and dc (figure 6-25). The drum and blowers are not turned off when relay 16 is energized.

If a fuse fault occurs in the 20-volt probe circuit, the +d-c alarm, relay 17, is energized through the indicator. Current flows from the negative side of C5 through R31, K17, the normally closed contacts of K14-1, R3, and the indicator to the +20-volt supply. The K17-1 contacts open the d-c contactors, relay 12, and turn off dc (figure 6-25).

If the current in the primary windings of the clock output transformer exceeds 1.5 amperes, one or both grasshopper fuses connected in series with the primary windings open. Current then flows from the negative side of C5 (figure 6-29) through R31, K17, the normally closed contacts of K14-1, the 4.3-millihenry choke coil (figure 6-21), the fuse fault indicator (figure 6-29), L5, and R19 to ground. The dc is turned off when the relay 17 contacts switch.

A plate overcurrent relay (figure 6-21) is incorporated as an additional safety feature in the clock output circuit. Since R19 is connected in the d-c return lead of V16 through V21, the total d-c plate current of the output stage can be monitored across this resistor. If the current through R19 is slightly under 1.5 amperes, the current through K1 is sufficient to switch the relay contacts. When the K1-1 contacts switch, relay 17 is energized by current which flows from the negative side of C5 (figure 6-29) through R31, K17, K14-1 normally closed contacts, and the K1-1 normally open contacts (figure 6-21) to ground. When the K17 contacts switch, dc is turned off, and the plate overcurrent relay deenergizes. The dc is also turned off if the d-c alarm, relay 18, energizes. The K18-1 contacts open the d-c contactor, relay 12, and turn off dc (figure 6-25).

Since an automatic power turn off can be caused by conditions other than an open fuse, a fuse-fault indication is available. To determine whether a fuse is open, press the FUSE FAULT TEST pushbutton. Pressing this pushbutton energizes relays 13 and 14 through the main-line switch (figure 6-29). When the K13-1 contacts switch, current flows from the main line through K13-1 normally open contacts, CR2, R25, and R26-C5 during negative alternations of the supply voltage. A negative potential then develops at the top of C5.

If a fuse fault has occurred in the processor blower circuits, pressing the FUSE FAULT TEST pushbutton causes current to flow from the negative side of C5 through R34, the FUSE FAULT indicator lamp, K13-2 normally open contacts, CR1, the fuse indicator, and R1 to ACN. In the absence of a ground-return resistor corresponding to R1, the circuit would be completed from the indicator through the drum-blower fuses and the blowers to ACN. Both return methods are used in the a-c circuits of the system.

If a fuse fault has occurred in the 20-volt probe circuit, pressing the FUSE FAULT TEST pushbutton causes current to flow from the negative side of C5 through R34, the FUSE FAULT lamp, K14-1 normally open contacts, R3, the fuse indicator, and the bleeder resistor in the +20-volt power supply to ACN.

If the FUSE FAULT indicator illuminates when the FUSE FAULT TEST pushbutton is pressed, one or more fuses have opened.

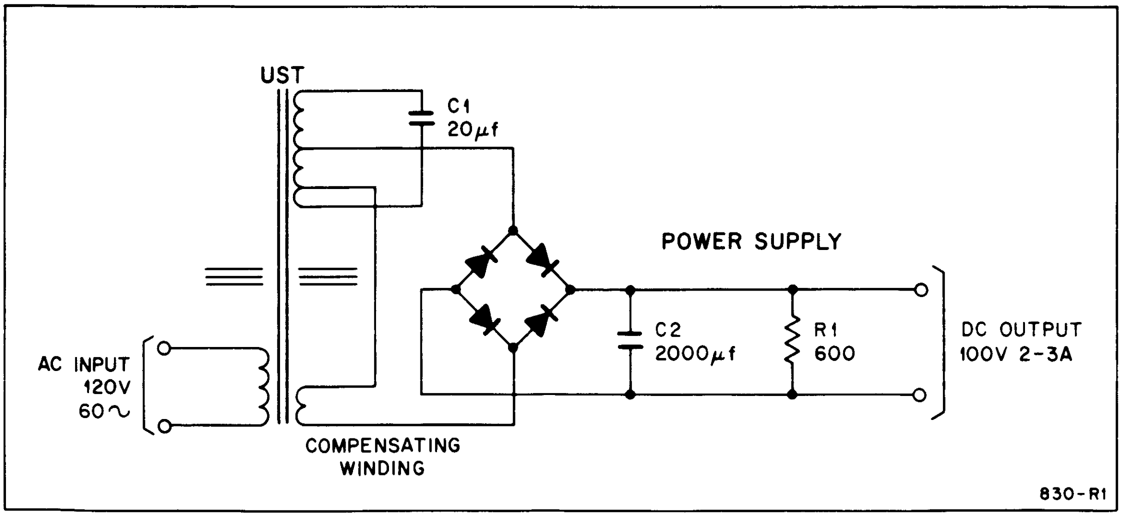
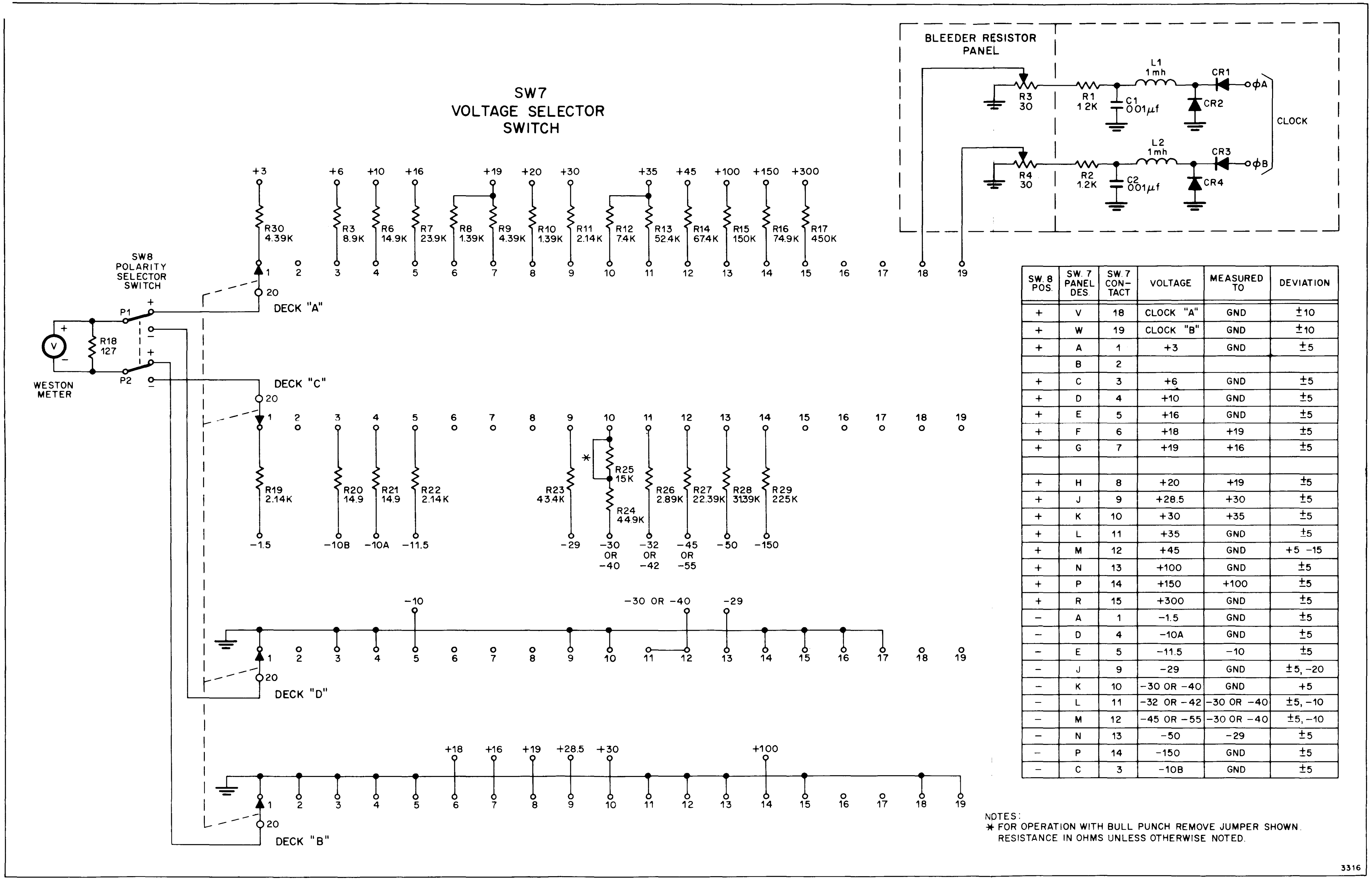


Figure 6-27. Typical Use of Voltage Stabilizing Transformer With Power Supply



SW. 8 POS.	SW. 7 PANEL DES.	SW. 7 CON-TACT	VOLTAGE	MEASURED TO	DEVIATION
+	V	18	CLOCK "A"	GND	±10
+	W	19	CLOCK "B"	GND	±10
+	A	1	+3	GND	±5
	B	2			
+	C	3	+6	GND	±5
+	D	4	+10	GND	±5
+	E	5	+16	GND	±5
+	F	6	+18	+19	±5
+	G	7	+19	+16	±5
+	H	8	+20	+19	±5
+	J	9	+28.5	+30	±5
+	K	10	+30	+35	±5
+	L	11	+35	GND	±5
+	M	12	+45	GND	+5 -15
+	N	13	+100	GND	±5
+	P	14	+150	+100	±5
+	R	15	+300	GND	±5
–	A	1	-1.5	GND	±5
–	D	4	-10A	GND	±5
–	E	5	-11.5	-10	±5
–	J	9	-29	GND	±5, -20
–	K	10	-30 OR -40	GND	+5
–	L	11	-32 OR -42	-30 OR -40	±5, -10
–	M	12	-45 OR -55	-30 OR -40	±5, -10
–	N	13	-50	-29	±5
–	P	14	-150	GND	±5
–	C	3	-10B	GND	±5

NOTES:
* FOR OPERATION WITH BULL PUNCH REMOVE JUMPER SHOWN.
RESISTANCE IN OHMS UNLESS OTHERWISE NOTED.

Figure 6-28. Voltage Monitor Circuit

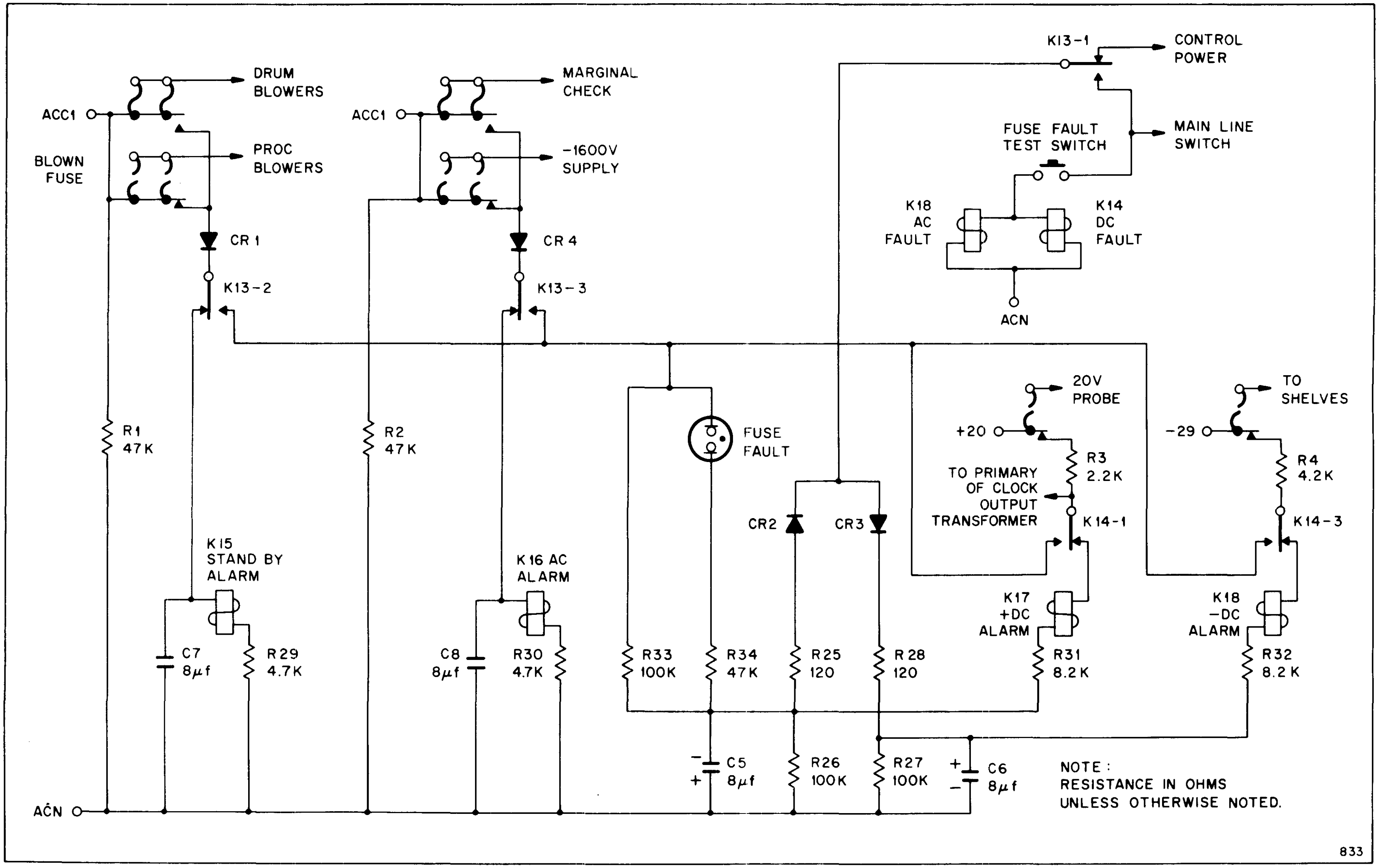


Figure 6-29. Typical Fuse-Fault Circuit

UNIVAC® SOLID-STATE 90 COMPUTER
X-6 ASSEMBLY — CODING CHART

Remington Rand Univac

OPERATIONS: CHARACTER GENERATING PROGRAM: FOR EDITING PAGE____ OF ____ PGS

	CARD TYPE	OPER. NO.	CARD NO.	a ADDRESS	KEY	INST. CODE		m ADDRESS	c ADDRESS	REMARKS	rA (K)	rX (T)	rL (Y)	WT
	1 2	3 4 5	6 7 8	11 12 13 14 15	16	17 18 19	20 21	22 23 24 25	26 27 28 29 30	46-85				
1								1400		ABCDEFGHIJ				
2								1401		ABCDEFGHIJ	PRIME	PART OF	1400	
3								1402		KLMNOPQRST				
4								1403		KLMNOPQRST	PRIME	PART OF	1402	
5				RR CODE				1404		UVWXYZ				
6								1405		UVWXYZ	PRIME	PART OF	1404	
7								1406		.$-#/*,&%:				
8								1407		.$-#/*,&%:	PRIME	PART OF	1406	
9								1408		+;'()				
10								1409		+;'()	PRIME	PART OF	1408	
11				USS-4 CODE				1410		0123456789	CAN ALSO	BE USED	FOR ARITHM.	
12													FUNCTIONS	
13		SAMPLE		2201		25		1402	2206					
14	P REQUIRED			2206		37		0500	2212					
15	IN P7			2212		32		0900	2220					
16	OF 4216			2220		37		0600	2228	ΔΔΔ PΔΔΔΔΔΔ				
17	PRIMED 4217			2228		60		4216	2234	4216 UNPRIMED				
18				2234		25		1403	2240					
19				2240		37		0500	2245					
20				2245		32		0900	2250					
21				2250		37		0600	2255	ΔΔΔPΔΔΔΔΔΔ				
22				2260		60		4217	2265	4217 PRIMED				
23														
24														
25														

UNIVAC® SOLID-STATE 90 COMPUTER
X-6 ASSEMBLY — CODING CHART

Remington Rand Univac

OPERATIONS: ____________ PROGRAM: ____________ PAGE ____ OF ____ PGS

	CARD TYPE	OPER. NO.	CARD NO.	a ADDRESS	KEY	INST. CODE		m ADDRESS	c ADDRESS	REMARKS	rA (K)	rX (T)	rL (Y)	WT
	1 2	3 4 5	6 7 8	11 12 13 14 15	16	17 18 19	20	21 22 23 24 25	26 27 28 29 30	46-85				
1				0945		60		2330	0950					
2				0825		26		0830		$\overset{m}{4206}$ ϕcard # location				
3			3	0830		60		4206	0835					
4				0950		25		1806	0955	$\overset{m}{1806}$ const. of 0000000001				
5				0955		70		4206	0960					
6				0960		60		4206	0965	add & store # of card m 4206				
7				0965		25		4212	0970					
8				0970		70		4201	0975	add & store				
9				0975		60		4201	0980	amount 4201				
10				0828		25		4201	0985					
11				0985		17			0990	zero suppress Translate to RR				
12				0995		60		2294	0997	& transfer				
13				0997		65		2299	1050	amount to P interlace				
14				0980		25		4206	1005					
15				1005		17			1010					
16				0990		62			0995	Translate to RR				
17				1010		62			1015	zero suppress				
18				1015		60		2378	1020	and card # to P interlace				
19				1020		65		2383	1025					
20				1025		27		1030	1025	Print Detail cards / first clear				
21				1030		11		2202	0835	unused Printerlace				
22										FEED CARD at 835				
23				1050		11		2204	1060	Print GT				
24				1060		67		4999	0814	STOP				
25														

UNIVAC® SOLID-STATE 90 COMPUTER
X-6 ASSEMBLY — CODING CHART

Remington Rand Univac

OPERATIONS: ______ PROGRAM: ______ PAGE ____ OF ____ PGS

	CARD TYPE	OPER. NO.	CARD NO.	a ADDRESS	KEY	INST. CODE	m ADDRESS	c ADDRESS	REMARKS	rA (K)	rX (T)	rL (Y)	WT
	1 2	3 4 5	6 7 8	11 12 13 14 15	16	17 18 19 20	21 22 23 24 25	26 27 28 29 30	46-85				
1			1					0814	START				
2				0835		72		0827	ENTER AT0835 on				
3			4	0827		96	2200	0830	second card				
4				0830		96	2200	0840					
5				0840		25	2211	0845	J20				
6				0845		60	2365	0850					
7				0850		25	2216	0855	J'20				
8				0855		60	2370	0860					
9				0860		25	2231	0865	J21 Transfer Name				
10				0865		60	2250	0870	from				
11				0870		25	2236	0875	J'21 J To P				
12				0875		60	2255	0880	interlace				
13				0880		25	2251	0885	J22				
14				0885		60	2334	0890					
15				0890		25	2256	0895	J'22				
16				0895		60	2339	0900					
17			2	0814		26	0820		GT				
18				0820		66	4201	0825	Clear 4201 to Ø				
19				0900		25	2271	0915					
20				0915		05	2276	0920	Translate				
21				0920		12		0925	to MC				
22				0925		60	4212	0930	Translated amount				
23				0930		25	2271	0935	J23				
24				0935		60	2325	0940	Transfer Amount				
25				0940		25	2276	0945	J'23 from J to P interlace				

UNIVAC® SOLID-STATE 90 COMPUTER
X-6 ASSEMBLY — CODING CHART

Remington Rand Univac

OPERATIONS: ______________ PROGRAM: ______________ PAGE ______ OF ______ PGS

	CARD TYPE / OPER. NO. / CARD NO.	a ADDRESS	KEY	INST. CODE	m ADDRESS	c ADDRESS	REMARKS	rA (K)	rX (T)	rL (Y)	WT
	1 2 3 4 5 6 7 8	11 12 13 14 15	16	17 18 19 20	21 22 23 24 25	26 27 28 29 30	46-85				
1		~~2704~~		~~60~~	~~4201~~	~~2750~~					
2		~~2750~~		~~00~~	~~2710~~		CLEAR COMPL.DRUM				
3		~~2710~~			~~TYPE~~	~~1~~					
4											
5											
6											
7		~~2704~~		~~60~~	~~4201~~	~~2750~~					
8		~~2750~~									
9											
10	CONSTANT	KEY	RL	00	0000	0000				0000000000	
11	CONSTANT	KEY	RX	00	0001	0000			0000010000		
12	(RL) → M	KEY	RA	50	0000	4999	m 0000 0000000000	5000004999			
13	RX +	4999		70	0004/3	0004/1					
14											
15											
16	CONSTANT	KEY	RL	00	0000	0000				0000000000	
17	CONSTANT	KEY	RX	00	0001	0000			0000010000		
18	ADD RX + RA	KEY	RA	70	0004/3	0004/1		700004/30004/1			
19	(RA) → M	KEY	RC	60	4999	0000	DEPR RUN | m 4999 | 700004/30004/1				
20	(RL) → M	KEY	RA	50	0000	4999	DEPR.RUN | m 0000 | 0000000000	5000004999			
21								0000010000	RX		
22	ADD RX + RA	4999		70	0004/3	0004/1		5000014999			
23		RA		50	0001	4999	m 0001 | 0000000000				
24		4999		70	0004/3	0004/1		5000024999			
25		RA		50	0002	4999	m 0002 | 0000010000				

UNIVAC® SOLID-STATE 90 COMPUTER

X-6 ASSEMBLY — CODING CHART

Remington Rand Univac

OPERATIONS: ______ PROGRAM: ______ PAGE ____ OF ____ PGS

	CARD TYPE	OPER. NO.	CARD NO.	a ADDRESS	KEY	INST. CODE		m ADDRESS	c ADDRESS	REMARKS	rA (K)	rX (T)	rL (Y)	W T
	1 2	3 4 5	6 7 8	11 12 13 14 15	16	17 18 19	20	21 22 23 24 25	26 27 28 29 30	46-85				
1				2815		70		4415	2820	WORD 0-1-2-3-4				
2				2820		70		4410	2825	ADD				
3				2825		70		4405	2830	PRINT [illegible] 51-60				
4				2830		70		4401	2835	No 2				
5				2835		17			2840					
6				2840		60		4018	2845					
7				2845		65		4023	2850					
8				2850		11		4002	2704					
9														
10														
11										IF 0001 in $\overset{m}{4201}$				
12										than ingrement				
13										m 4005 by 3				
14														
15														
16														
17														
18														
19														
20														
21														
22														
23														
24														
25														

UNIVAC® SOLID-STATE 90 COMPUTER

• X-6 ASSEMBLY – CODING CHART

~~WORD [illegible] 4401~~

Remington Rand Univac

WORD 0-1-2-3-4

OPERATIONS: ______

PROGRAM: ADD ______

PAGE ______ OF ______ PGS

PRINT RESULT 50- 60

	CARD TYPE	OPER. NO.	CARD NO.	o ADDRESS	KEY	INST. CODE		m ADDRESS	c ADDRESS	REMARKS	rA (K)	rX (T)	rL (Y)	WT
	1 2	3 4 5	6 7 8	11 12 13 14 15	16	17 18 19	20	21 22 23 24 25	26 27 28 29 30	46-85				
1									2704					
2				2704		72			2709					
3				2709		96		4200	2715					
4				2715		25		4201	2719	Nr 1				
5				2719		05		4206	2724					
6				2724		12			2730					
7				2730		60		4401	2735					
8				2735		25		4221	2740					
9				2740		05		4226	2745					
10				2745		12			2750					
11				2750		60		4405	2760					
12				2760		25		4241	2765					
13				2765		05		4246	2770					
14				2770		12			2775					
15				2775		60		4410	2780					
16				2780		25		4261	2785					
17				2785		05		4266	2790					
18				2790		12			2795					
19				2795		60		4415	2800					
20				2800		25		4281	2805					
21				2805		05		4286	2810					
22				2810		12			2815					
23														
24														
25														

S M

0 x

1 0 0

0 1 1

0 0 1

M [illegible] 2 3 4
[illegible] for 4, 9

S for [illegible] 7
M 4 or 3 8

35 S = 0 to 9
M = 4 or 9
36 S = 1, 3, 6 or 8
M = 3 or 8
37 S = 2, 3, 7 or 8
M = 2, 3, 7 or 8
38 S = 3 or 8
M = 1, 3, 6 or 8
39 S = 4 or 9
M = 0 to 9
40 S = 4 or 9
M = 1, 3, 6 or 8
41 S = 4 or 9
M = 4 or 9
42 S = 3 or 8
M = 2, 3, 7 or 8
43 S = 4 or 9
M = 2, 3, 7 or 8
44 S = 2, 3, 7 or 8
M = 4 or 9
45 S = 1, 3, 6 or 8
M = 4 or 9

---

46 S = 0, 1, 5 or 6
M 2, 3, 5, 6, 7 or 8
47 S = 0, 1, 2, 3 or 4
M = 4 or 9
50 S = 0 or 5
M = 1 or 6
51 S = 2 or 7
M = 3 or 8