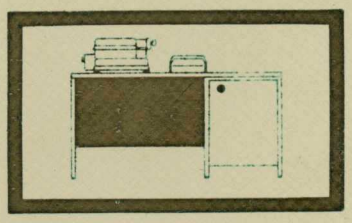
RECOMP

COMPUTER SYSTEMS



THEORY OF OPERATION

AUTONETICS
A DIVISION OF NORTH AMERICAN AVIATION, INC
Industrial Products
3400 E. 70TH STREET LONG BEACH CALIFORNIA

RECOMP

COMPUTER SYSTEMS

THEORY OF OPERATION

Publication 555-A-14

28 MAY 1962

AUTONETICS Industrial Products
A DIVISION OF NORTH AMERICAN AVIATION, INC. 3400 E. 70TH STREET, LONG BEACH, CALIFORNIA

CUSTOMER SERVICE
RECOMP MAINTENANCE

BULLETIN

TITLE: Scope of RECOMP Computer Systems Theory of Operation Manual
NUMBER: R3-MA-3
DATE: 23 July 1962
ATTACHMENTS: None
PAGE: 1 OF 1
SUPERSEDES: None

This bulletin outlines the scope applicability of the RECOMP Computer Systems Theory of Operation manual. The manual is written to apply to systems wired for floating point capability during manufacture; however, most of the information applies to all RECOMP III systems and certain portions apply also the RECOMP II systems. Manual organization is such that future expansion of the document can be readily achieved both in regard to the RECOMP II system and optional equipment.

The RECOMP Computer Systems Theory of Operation manual is intended principally for training. A second, but equally important purpose is as a major reference document in the field when special, detailed information on operation is required for solution of complex maintenance problems. This is in contrast to the RECOMP III Service manual which contains only that information customarily required in maintaining the equipment.

Information in the manual at present mainly comprises physical and functional descriptions of the equipment. The logic description will be distributed later. Material in the manual is divided into the following categories: introduction to the system; number systems, codes, and formats; general description of the system and its elements, including a listing of physical and operating specifications; basic procedures and precautions necessary during operation; principles and sequences utilized in performing the various system operations; and descriptions on operation of circuits and mechanical assemblies.

The portions in sections 1, 2, 3, 4, and 6 applying to both RECOMP II and RECOMP III systems are those concerning number systems conversion, fixed point numeric format, command format, memory design and operation, power supply, signal and control circuits (except logic driver), pre-operating preparations, operating precautions, loop and main memory selection, and channel and sector selection.

All of the information in section 5 is pertinent to RECOMP II but that pertaining to the names and functions of the modes and phases, indexing, floating point arithmetic, and serial-parallel progression which are applicable to RECOMP III only.

Thorough familiarity with the scope and organization of this manual prior to actual use will prove beneficial at the time of use. This familiarity can readily be achieved by studying the table of contents, lists of illustrations and tables, and the introductory paragraphs of each section which briefly outline the content of the sections.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

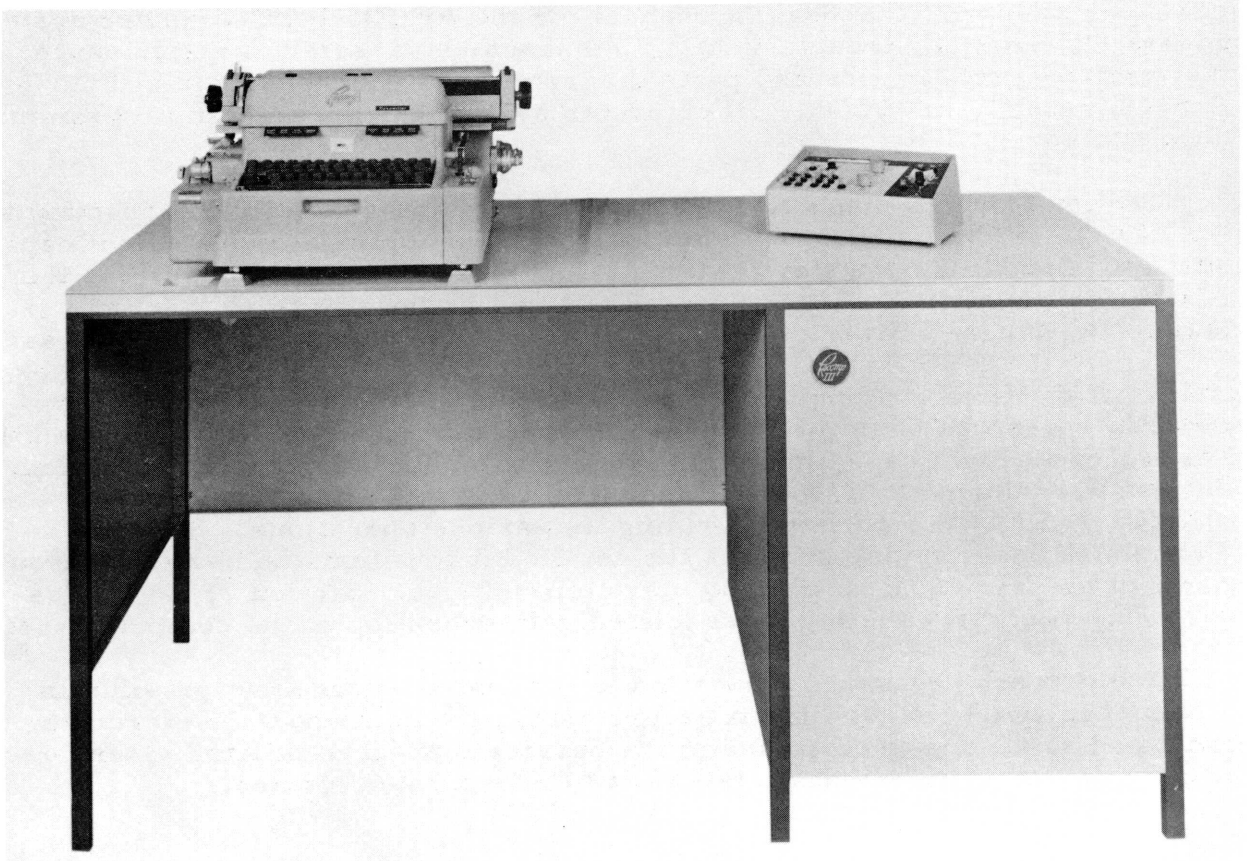
LIST OF TABLES

PREFACE

Comprising seven sections, this manual presents the physical and functional descriptions, operating instructions; and logic description on the RECOMP III computer system. The Service manual for the RECOMP III computer system covers checkout and troubleshooting; disassembly-assembly; maintenance; test routines; schematics and parts diagrams; and checkout logic and wiring. Programming of a RECOMP III computer system is explained in programming documents.

Section 1 of this manual, Introduction, contains a general introduction to the RECOMP III computer system and its various optional input-output capabilities. Section 2, Number Systems, Codes, and Formats, describes the number systems, codes, and formats utilized by the computer system. Section 3, General Description and Specifications, contains a general description of the RECOMP III computer system and its elements, followed by a listing of physical and operating specifications. Section 4, Operating Procedures, presents basic procedures for operation of the RECOMP III computer system in normal usage, including precautions necessary during operation. Section 5, Principles of Operation, outlines the methods and sequences utilized by RECOMP III in performing its various operations. Section 6, Functional Description, contains the detailed discussions on how the various parts of the RECOMP III computer system function. Section 7, Logic Description, describes the logic associated with operation of the computer system.

Within each section, information is presented as much as possible in a sequence which will facilitate understanding the equipment. Referencing from one section to another and to the Service manual is utilized extensively to aid in locating information pertinent to interrelated elements.



R3-21

RECOMP III COMPUTER SYSTEM

Section 1
INTRODUCTION

GENERAL

RECOMP III is a solid state, general purpose, digital computer system designed especially for scientific computation, industrial calculations, and data reduction. Standard configuration is a fixed-point-arithmetic computer unit, a control console, a Flexowriter input-output unit, and a tape rewind unit, all located within or on top of a desk. Optionally, the system may incorporate floating point arithmetic capability and other types of input-output equipment. RECOMP III can be adapted to handle as many as three additional optional input-output devices such as high-speed paper tape punch-reader units, card punch-reader units, and plotters. Figure 1-1 illustrates the main elements of the computer system; equipment enclosed in dash lines represents optional devices and capabilities.

The Flexowriter of the RECOMP III system comprises a keyboard, printer, paper tape reader, and paper tape punch. Function of the keyboard is manual entry of commands and data into the computer, that of the printer is recording of data on paper under computer control. The tape reader enters commands and data from punched paper tape into the computer; the tape punch

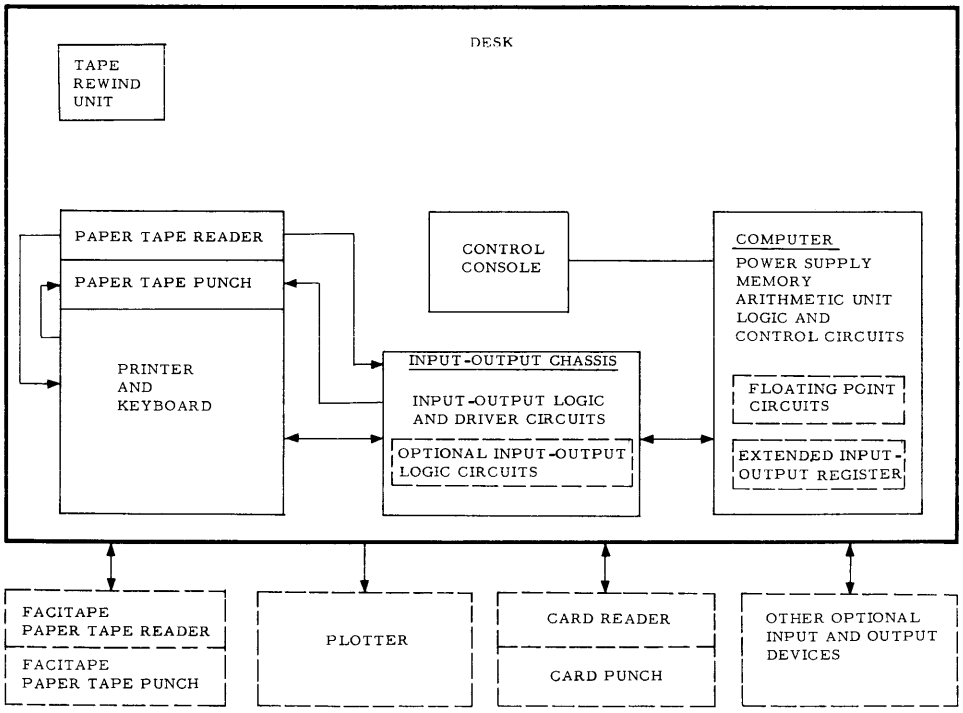


Figure 1-1. RECOMP III Computer System Block Diagram

records information from the computer on paper tape. Auxiliary functions of the Flexowriter when it is not operated with the computer are (1) preparation of punched paper tapes for subsequent input of commands and data into the computer, (2) reproduction of punched tapes, and (3) printing of contents of punched tapes. The tape rewind unit is used for feeding tape to the paper tape reader and for rewinding read or punched tapes.

The computer unit contains the elements for storage of data and commands, control of internal operations, computation and processing of data, and regulation and distribution of power. Facilities for external control of the system are provided by switches and indicators on the control console.

Principal features of the RECOMP III computer system are its ease of programming and simplicity of operation. Other features are (1) its large word size and memory capacity in relation to its cost, (2) its ability to operate with as many as four inputs and four outputs, (3) its ability to read and punch paper tape in any code of five through eight channels, (4) its portability and accessibility for servicing, and (5) its use of printed circuitry to keep maintenance at a minimum. RECOMP III operates from any standard 115-volt a-c outlet.

Two units of special test equipment are available for testing computer unit circuits. These are a system tester and a component tester. The system tester is used to test computer flip-flops and logic gates and can also be utilized for checking operation under marginal conditions. When testing the computer with the system tester, the computer is fully energized as for normal operation. The component tester is used to individually test computer etched-circuit plug-in boards.

Section 2

NUMBER SYSTEMS, CODES, AND FORMATS

GENERAL

The number systems, character codes, and information formats utilized by the computer system are described in general in this section. Detailed information on the individual number systems, codes, and formats as they are used and acted upon during specific operations is presented in other sections of this manual to facilitate understanding of the associated subjects.

Number system used in the RECOMP III computer system is binary. Character coding can be in any ones-and-zeros code of five-to-eight channels. Keyboard entry and printer output, however, use a predetermined, fixed 6-channel ones-and-zeros code called alphanumeric. Internal memory storage formats recognized by RECOMP III circuitry are fixed point binary and floating point binary.

NUMBER SYSTEMS

The computer uses only the binary number system of zeros and ones in its internal operations. However, through programming, information may be entered into and recorded by its input-output equipment in the decimal, octal, or combined octal-binary number systems. The information may also be programmed for entry and output in the non-number system called alphanumeric, but usually computations are not performed on this information.

Conversion between the binary number system used internally and the decimal, octal, or combined octal-binary number systems used with input-output is accomplished by the computer with an internally stored program. Alphanumeric information is not converted because it is represented internally in binary by a ones-and-zeros configuration identical to that of the code used. However, regardless of the manner in which information is represented externally, the information is always transferred between the input-output equipment and the computer in a ones-and-zeros representation.

The decimal number system, consisting of the 10 digits, 0 through 9, is used in keyboard entry and printer output of numerical data because it is most readily understood by the operator. Octal, comprising the eight digits, 0 through 7, is used in entry and output of commands and memory locations because it is easier to write a command or location in the octal than in the binary system. Binary, consisting of the two digits 0 and 1, is used by the computer in its internal operations because it is simple and economical to mechanize.

Conversion of Octal or Binary to Decimal

Manual methods of conversion between number systems vary, depending on the systems being converted and whether the numbers are integers or fractions.

To convert octal or binary to decimal, each digit, starting from the most significant, is first separately converted to its decimal equivalent. The converted digits are then added to produce the final decimal number.

Example: Convert octal 257.025 to decimal

$2 \times 8^2 + 5 \times 8^1 + 7 \times 8^0 + 0 \times 8^{-1} + 2 \times 8^{-2} + 5 \times 8^{-3} =$

$128 + 40 + 7 + 0 + 0.03125 + 0.009765625 = 175.041015625$

Example: Convert binary 101.1010 to decimal

$1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 0 \times 2^{-4} =$

$4 + 0 + 1 + 0.5 + 0 + 0.125 + 0 = 5.625$

Conversion of Decimal Integers to Octal or Binary

To convert decimal integers to octal or binary, divide the number by 8 for octal or 2 for binary, and note the remainder to one side. Divide the new quotient by 8 for octal, or 2 for binary and again note the remainder to one side. Continue the preceding steps until a zero quotient is obtained. Then write the remainders in reverse order of occurrence.

Example: Convert decimal 216 to octal

		Remainder
8)	216	
8)	27	0
8)	3	3
	0	3

216 decimal = 330 octal

Example: Convert decimal 216 to binary

		Remainder
2)	216	
2)	108	0
2)	54	0
2)	27	0
2)	13	1
2)	6	1
2)	3	0
2)	1	1
	0	1

216 decimal = 11011000 binary

Conversion of Decimal Fractions to Octal or Binary

To convert decimal fractions to octal or binary, multiply the decimal fraction successively by 8 for octal or by 2 for binary and then place the decimal point in the product at the same position as in the multiplicand. The first digit to the left of the decimal point is the most significant digit of the octal or binary fraction. Multiply all digits of the product, except the integer part, by 8 for octal or 2 for binary. In the succeeding product, the digit to the left of the decimal point is the next most significant octal or binary digit. Repeat the preceding steps on the product fraction until the desired accuracy is obtained.

Example: Convert decimal fraction 0.312 to octal

```
0.312          0.312 decimal = 0.23757 octal
  x8
2.496
  x8
3.968
  x8
7.744
  x8
5.952
  x8
7.616
```

Example: Convert decimal fraction 0.312 to binary

```
0.312          0.312 decimal = 0.010011 binary
  x2
0.624
  x2
1.248
  x2
0.496
  x2
0.992
  x2
1.984
  x2
1.968
```

Conversion of Octal to Binary

To convert octal numbers to binary numbers, replace each octal digit by a group of three binary digits.

Example: Convert octal 7654.63 to binary

```
 7    6    5    4   .   6    3   =
111  110  101  100  .  110  011  =
111110101100.110011 binary
```

Conversion of Binary to Octal

To convert binary numbers to octal numbers, place the binary digits in groups of three starting from the binary point; then replace each binary group with its octal equivalent.

Example: Convert binary 1110011101100111.111101001 to octal

1 110 011 101 100 111 . 111 101 001 =

1 6 3 5 4 7 . 7 5 1 =

163547.751 octal

A table in the RECOMP III Service Manual provides a quick means for obtaining the decimal value of binary digits for the powers 0 through 39 and octal digits for the powers 0 through 13, which in both instances is the word capacity of the RECOMP III computer.

CODES

The RECOMP III computer and the Flexowriter paper tape reader and punch can operate with any 5- to 8-channel code. However, the Flexowriter keyboard and printer function only with an established 6-channel code. This code (figure 2-1) uses 52 different binary combinations to represent all letters of the alphabet, numbers, special character, and functions. All functions, (except upper case, lower case, and backspace), special characters, and numbers have a punchout in the No. 6 channel.

Because only a 6-channel code is used by the keyboard and printer while the computer always handles input and output in an 8-channel code, two of the eight channel positions are always represented in the computer as zeros.

INTERNAL INFORMATION FORMATS

A computer word consists of 41 bits (binary digits) which may represent (1) a fixed point numerical quantity to be operated upon, (2) computer commands that specify the operations to be performed, (3) alphanumeric information that usually is only stored for subsequent use in identification or explanation of computation results, or (4) a floating point numerical quantity to be operated upon. In all formats, the right-most bit, called the sync bit, is used for computer synchronization and is not available in programming.

The representation of bit times for the computer is opposite to the bit position designation used by programming (figure 2-2). It is essential to remember this difference when reading any programming publication.

Fixed Point Numeric Format

In the fixed point numeric format, (figure 2-3), the sign of the quantity (1 for plus and 0 for minus) occupies the leftmost bit position, and the absolute value of quantity occupies the remaining 39 of the 40 information (programmable) bit positions. The most significant bit is in bit position 40 and

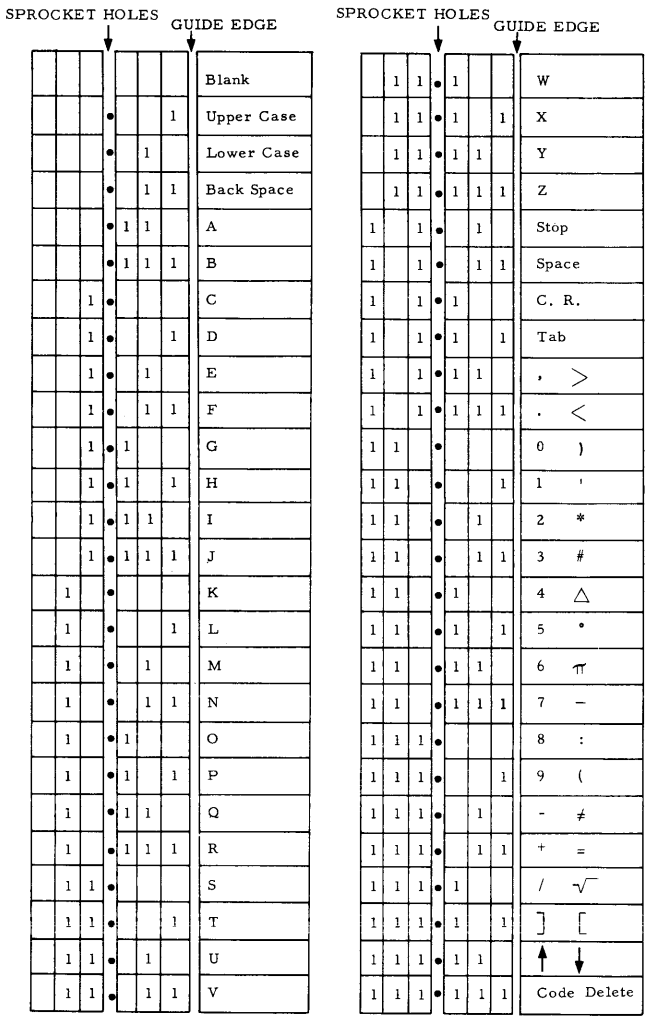


Figure 2-1. Keyboard-Printer Code Configuration

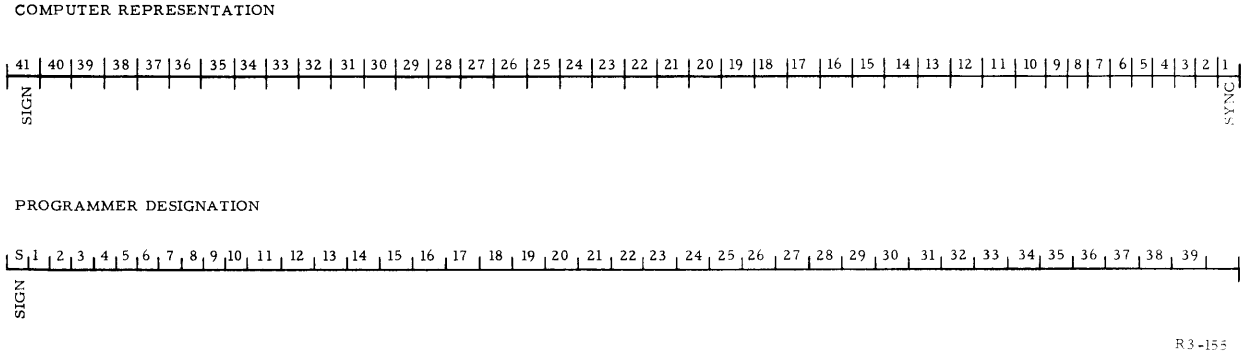


Figure 2-2. Comparison of Computer Representation and Programmer's Designation

the least significant is in bit position 2. Normally the binary point (analogous to decimal point) is thought of as being between the sign and the most significant bit, that is, a numerical quantity is considered as less than one (a fraction) in absolute value. Each number word occupies a definite location in the memory. To control processing of a number, its location must be specified in the given command. The specific location is referred to, first by channel, then by sector.

Alphanumeric Format

Each character in alphanumeric format (figure 2-3) is represented in the computer by eight bit positions. Thus a 40-bit word can contain up to five alphanumeric characters.

NOTE

THE TERM ALPHANUMERIC IS USED IN THIS MANUAL ONLY TO SIGNIFY INPUT OR OUTPUT WITH THE KEYBOARD OR PRINTER, RESPECTIVELY, IN THE ESTABLISHED 6-CHANNEL CODE.

An alphanumeric character is entered into and output from the computer memory unaltered from its binary representation at the input-output device.

Command Format

All commands (figure 2-3) are single address and stored two to a word. A sign bit, an operation code usually consisting of two octal digits (six bits), and an address containing four octal digits and one binary digit (13 bits) make up a command.

The sign of a command is indicated by a 1 for positive or a 0 for negative. The sign has no significance in a left command. In a right command a 0 specifies indexing and a 1 specifies no indexing.

The operation code determines the input-output, arithmetic, or logic operation to be performed. Although most operation codes contain two octal digits, a few also utilize the highest, next to highest, or lowest bit of the address (bit positions 34 or 14, 33 or 13, 22 or 2, respectively). A command using one of these address bits as part of the operation code must contain a one in the required position to order the operation desired.

Meaning of the address portion varies, depending on the nature of the command. Usually the address portion is the memory location of an operand, with the four octal address digits referring to the memory channel and sector of the operand. The first two octal digits (six bits) specify the channel and the second two octal digits (six bits) indicate the sector. Depending on whether it is a zero or one, the additional (13th) bit specifies the first or second command in a word (except when it is part of the operation code). However,

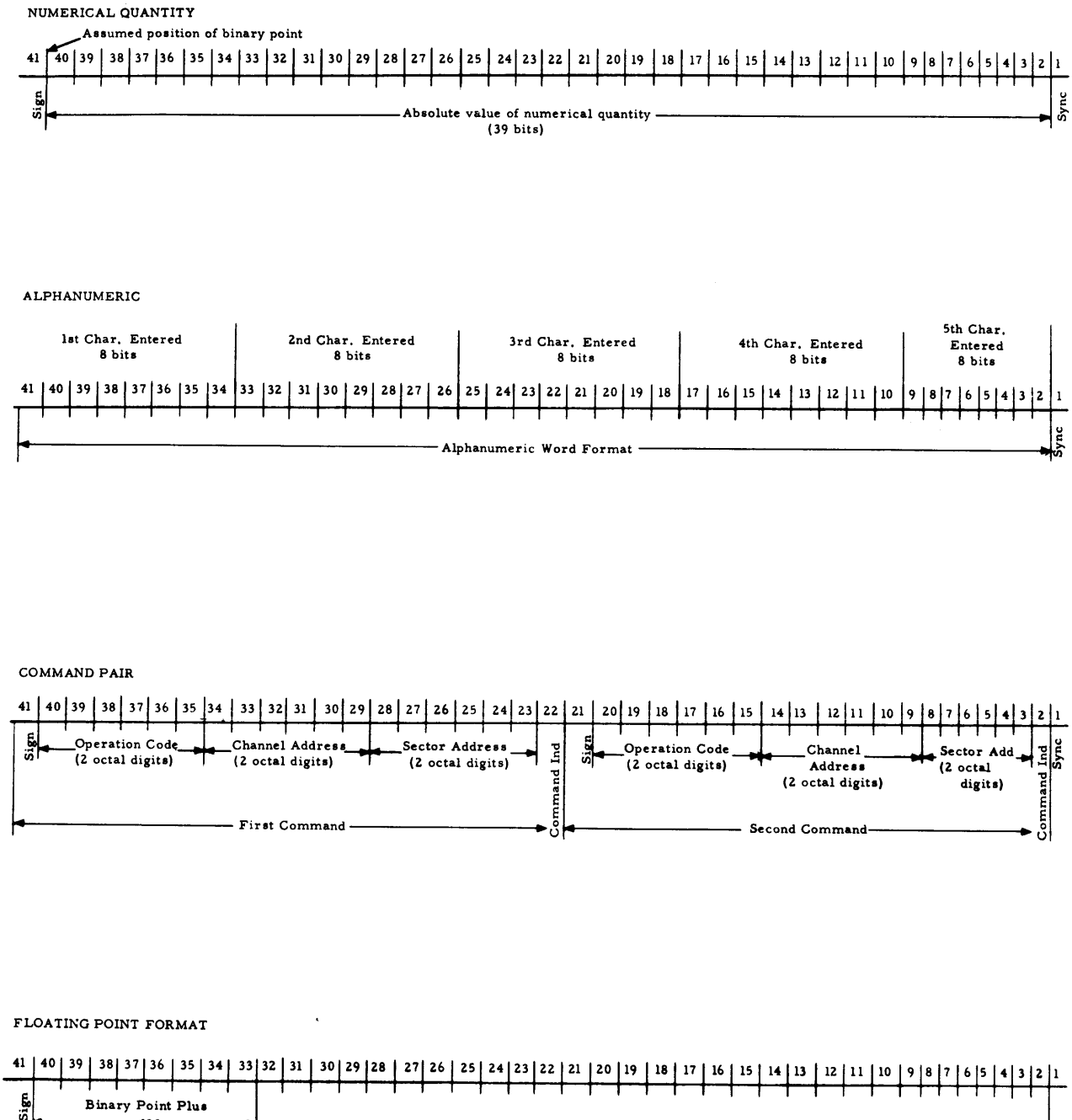


Figure 2-3. RECOMP III Word Formats.

as a first or second command indicator, this bit is significant only in transfer of control, store address, and load and store index commands. Other functions of the address portion are (1) specifying the number of binary shifts to be made or (2) indicating the number of characters to be input or output. In many commands the address portion has no significance. The commands available in RECOMP III are presented in section 3 of this manual.

Floating Point Numeric Format

In a floating point numeric word, (figure 2-3), the binary point is, in effect, carried along with the number during computations. The programmer need not be concerned with scaling and overflow because the arithmetic operations automatically scale the numbers correctly.

A floating point word contains only 31 data bit positions, the remaining 8 bit positions being utilized for the exponent. Thus, where the number (unless it is a zero) is normalized (bit 32 is a 1-digit), the binary point is counted from bit 32 instead of from bit 40. The binary point plus 128 is used instead of a signed binary point (this is known as "excess 128"). Thus, -1.0 would appear in a word as follows:

-10000001 1.00000000000000000000000000000000

Maximum numeric capacity of a floating point word would thus be 31 one's (approximately 1) times 2^{127} or about 10^{38}.

The smallest number would thus be 1/2 times 2^{-128} or about 10^{-38}. Floating point numbers are also commonly thought of as fractions times a power of 2 (the exponent).

Section 3

GENERAL DESCRIPTION AND SPECIFICATIONS

GENERAL

This section contains a general description of characteristics, a brief description of functions, and detailed specifications of the computer system and its various elements. The specifications, divided into physical and operating categories, are presented following the general description and discussion of functions.

DESCRIPTION AND FUNCTIONS

The standard RECOMP III computer system comprising a computer unit, control console, input-output unit, and tape rewind unit is all housed within or on top of the desk. Elements within the computer unit include a magnetic disk memory, power supply, circuit boards, and cooling blowers. The basic computer has facilities for incorporating additional input-output equipment and floating point arithmetic capability. This is accomplished by adding accessory circuit boards as described later in this section. Installation of these accessory circuits does not change the general operating concept from that of a standard RECOMP III system.

Five power and signal cables interconnect the elements of a standard RECOMP III computer system. One power cord to a standard 115-volt outlet supplies all source power to an entire standard system.

General descriptions and discussions of the functions of each element of the computer system are contained in following paragraphs. Detailed functional descriptions of each computer system element are found in section 6 of this manual.

Memory

The magnetic memory disk provides (1) main information storage, (2) most of the storage space for the rapid access loops and arithmetic registers, and (3) the storage for basic timing signals and the permanent memory load (bootstrap) and diagnostic routines. Information is stored in channels (tracks) on the disk in the form of flux patterns representing binary digits. Nominal disk rotation rate is 3450 rpm. The disk is driven by a rotor which is ball-bearing mounted on a non-rotated shaft. (figure 6-27). Diaphragms attach the disk to the rotor and the non-rotating shaft to the headplate and memory frame. These diaphragms provide stiff coupling between the memory components, but are flexible enough to ensure parallel rotation between disk and headplate. When operating, proper separation between disk and headplate (approximately 50 microinches) is maintained by an air thrust bearing. The 50-microinch separation ensures satisfactory information recording and reproduction with the levels of voltages used.

Main Memory Storage

Main memory storage comprises 64 channels of 64 word sectors (locations) each for a total main storage capacity of 4096 words. Each word consists of 41 bits (39 data, 1 sign, and 1 synchronization). The synchronization bit is not used for programming. A word can represent a number, a pair of commands, or five alphanumeric characters.

All word sectors in main storage are directly addressable except the last 16 sectors in the 64th channel. The addresses for these 16 sectors, commonly called the "gray" area, are reserved for designating word locations in the rapid-access loops. Information can be stored in the 16 non-addressable main memory locations, however, through use of special programming techniques.

Locations of words in main memory are addressed by four octal digits from 0000 through 7757. The first two digits identify the memory channel in which a specific word is located; the last two denote the sector or word location within the channel.

In addition to the 64 main storage channels, the disk also contains one additional channel capable of utilization for permanent storage. This channel, which is inaccessible by normal programming-operating procedures, is called the origin-sector channel and contains permanent bootstrap memory load and diagnostic routines. These built-in routines facilitate program loading and easy, fast computer checkout.

Average access time to stored information in main memory is approximately 8.50 milliseconds; minimum access time being 0.81 milliseconds, and maximum, 17.82 milliseconds. Main memory storage is the nonvolatile type, that is, the information is not subject to self-erasure and is not destroyed by power interruption.

Rapid Access Loops

Two recirculating loops, L and V, of eight words each provide rapid access to stored information when addressed. Both loops are contained in one memory channel, along with the registers. Loop addresses are specified by octal digits 7760 through 7767 for the L-loop and 7770 through 7777 for the V-loop. The least significant digit specifies which of the eight word sectors in a loop is addressed. Access to loop information averages 0.95 milliseconds, with a minimum of 0.81 milliseconds and maximum of 2.70 milliseconds. Information contained in the loops is volatile in that it is subject to erasure in the event of power interruption.

Registers

Five recirculating registers, each containing storage space for one word, are located in the same memory channel as the loops. Part of each register is on the memory disk and part is in flip-flops, which form recirculating circuitry. Characteristics of the registers are given in the following paragraphs.

Accumulator Register

The accumulator (A) register consists of six flip-flops and 35 bit positions

on the disk. This register is the principal operand register of the computer, holding and acting upon the operand during most command executions. Prior to execution of an arithmetic command, the A-register holds the number against which a number from memory is applied; following execution of an arithmetic command, it holds the result of an arithmetic operation.

Following some instructions (for example, addition or subtraction) the contents of the A-register may overflow from position 1, that is, the result of the operation may be too large to be contained in the register. Information which overflows is lost. To notify the operator when an overflow occurs, an overflow trigger is activated and the OVERFLOW indicator on the control console lights. When the trigger is on, the instruction TOV (transfer on overflow) must be executed to extinguish the indicator. To find and correct an overflow condition may require operation of the computer through one command at a time.

Number Register

The number (B) register consists of three flip-flops and 38 bit positions on the disk. Every number or command transferred out of or entered into memory is routed through the B-register. During execution of arithmetic commands, the B-register holds the number from memory that is applied against the number in the A-register. The B-register is not available to the programmer.

Remainder Register

The remainder (R) register consists of four flip-flops and 37 bit positions on the disk. This register has two principal uses:

1. After the execution of a fixed-point multiplication instruction, the R-register temporarily contains the least significant half of the product and in this respect is considered an extension of the A-register.
2. During execution of a fixed-point division instruction, the least significant half of the dividend appears in the R-register; after division, the remainder appears temporarily in this register.

Command Register

The command (Z) register consists of two flip-flops and 39 bit positions on the disk. This register holds the pair of commands to be executed, and is not available to the programmer.

Location Counter and Index Register

The location counter and index (G) register is a storage register consisting of two flip-flops and 39 bit positions on the disk. In addition to being a location counter and index register, part of the G-register also is a character counter during input-output and shifting operations. The location counter portion is in the left half of the register (bit positions 22 through 34), the index portion in the right half (bit positions 2 through 14). Input-output character counting portions of the register occupy bit positions 35 through 40 in the left half of the register and bit positions 15 through 20 in the right half.

For most command executions, the address in the location counter is increased to indicate the next half-word location each time a command is executed. The exception occurs on a transfer of control command when the location counter is set to the address of the command being executed. Thus commands are automatically executed in address sequence, unless the sequence is interrupted by transfer of control commands.

The index register portion automatically modifies the address of a right command if indexing had been ordered. A 0-bit must be placed in the sign position of the right command to obtain indexing. Character counter portions of the register count the characters input or output until the desired number is input or output, or tally the number of binary positions a number is shifted.

Timing Channels

Basic timing sources for computer control are provided by clock and sector track signals permanently recorded in the clock and origin-sector channels of the memory disk. A sine wave originating in the clock channel produces equally spaced pulses which control triggering of electronic components. Since there are as many clock pulses in the clock channel as there are binary digit (bit) spaces in the information channels, one clock pulse defines one bit time. Sector pulses recorded in the sector channel synchronize each memory location with memory head scanning in order to transfer words to and from the correct memory location. Additional appropriate timing signals for control and arithmetic operations are provided by a bit counter (composed of flip-flops) and timing flip-flops and gates.

In addition to supplying the signals for memory write-read synchronization, the origin-sector channel also provides storage for memory load (bootstrap) and diagnostic routines. The first eleven words of the sector channel comprise the bootstrap routine; the remainder of the channel contains the diagnostic routine.

Normally not addressable by a program, the sector channel can be read, however, through a special operating procedure. This procedure utilizes the LOCATION RESET switch in a special manner to cause an instruction to be read from the sector channel instead of a main memory location. If computation is started at location 0000.0, the instructions stored in the sector channel will then be executed. It is impossible to write into the sector channel; memory write commands will write only into addressable locations.

Power Supply

Primary power for the computer is supplied from a single-phase, 60-cycle, 115-volt source. Paralleled connectors on the rear of the desk permit 115-volt connections for the standard and optional input-output devices. During the computer start process, power control circuitry of the power supply switches power to the memory, blowers, and secondary power supplies at the appropriate fixed times. The timing of power application is very important to the memory positioning procedure.

Secondary power supplies provide d-c voltages for operations of the computer and control console. These secondary supplies are mounted on six etched circuit boards which plug into receptacles on the power supply chassis: the +6, -6, and +0.75-volt board in J1; the -12 and +75-volt board in J2; the -18, -100 and -3-volt board in J3; the relay control network board in J4; the power supply network board in J5; and the power supply filter board in J6. Uses of the secondary power supplies are as follows:

-100-Volt Power Supply - Unregulated -100-volt power is supplied for the neon indicators and as reference voltage for the other power supplies.

-18-Volt Power Supply - -18-volt power with regulation of three percent is supplied to all logic gates in the computer.

-12-Volt Power Supply - -12-volt power with regulation of three percent is supplied to logic flip-flops, read amplifiers, and clock circuits.

-6-Volt Power Supply - -6-volt power with regulation of three percent is supplied to the logic flip-flops and the read amplifiers.

-3-Volt Power Supply - -3-volt power is supplied to the emitters of the 64 information channels write switches.

+75-Volt Power Supply - +75-volt power with regulation of three percent is supplied to clock power amplifier No. 1 circuit board.

+6-Volt Power Supply - +6-volt power with regulation of three percent is used mainly in the logic flip-flops.

+0.75-Volt Power Supply - +0.75-volt power with regulation of five percent is derived from 6-volt regulated power and supplied as bias cutoff of the driver stage of the read switching network.

Signal and Control Circuits

The signal and control circuits operating in conjunction with the memory and registers, provide basic timing and gating, perform arithmetic computations, temporarily store information, and provide initial timing for the magnetic memory and synchronization of the logic circuits. Some of these circuits also control transfer of information between the computer and input-output equipment. Operations performed (commands executed) are listed in table 3-2.

Virtually all of the signal and control circuits are mounted on plug-in circuit boards in the computer, the remainder being located in the other units of the computer system. Each circuit board is of a particular building-block type to allow interchangeability among boards of a single type and to allow a modular-replacement form of maintenance when a particular circuit fails to function.

Types and numbers of plug-in circuit boards which mount in the RECOMP III computer (figure 3-1) are as follows:

Clock power amplifier No. 1	1	
Clock power amplifier No. 2	1	
Logic network (gates)	38	
Flip-flop (four per board)	*13	
Write amplifier	3	
Write switch	8	
Read amplifier	17	
Read switch	8	
Logic driver	6	
Floating point arithmetic logic	8	(optional)
Extended input-output register logic	1	(optional)

* Including two boards for optional floating point arithmetic and extended input-output register.

The computer is internally wired to the connectors for the boards required for operation of optional equipment. Circuit boards for the standard input-output equipment are located on the input-output chassis in the desk. The following circuit descriptions are written in terms of general functional stages of each circuit board.

Clock Power Amplifier

A clock signal synchronizes all computer operations by triggering the flip-flops at a constant nominal frequency of 153 kilocycles per second. Because the source of the clock signal is a permanently recorded sine wave on the memory disk, all operations are synchronized with information transfers to and from memory as well as with each other. The clock signal consists of a squared voltage wave that is false (approximately ground level) for 1.75 microseconds and true (approximately -6 volts) for 4.62 microseconds.

Clock power amplifier No. 1 shapes the clock signal and partially amplifies it. This board has two outputs. One, the strobe signal, is transmitted directly to all read amplifiers in the computer to gate memory reading. The other output is transmitted to clock power amplifier No. 2. Clock power amplifier No. 2 inverts the signal and provides the final current amplification required to drive the computer logic flip-flops.

Logic Network

Computer operation is specified by logic equations. These equations are mechanized by means of standardized "and", "and-and", and "and-or" diode logic gates, most of which are mounted on the logic network circuit boards.

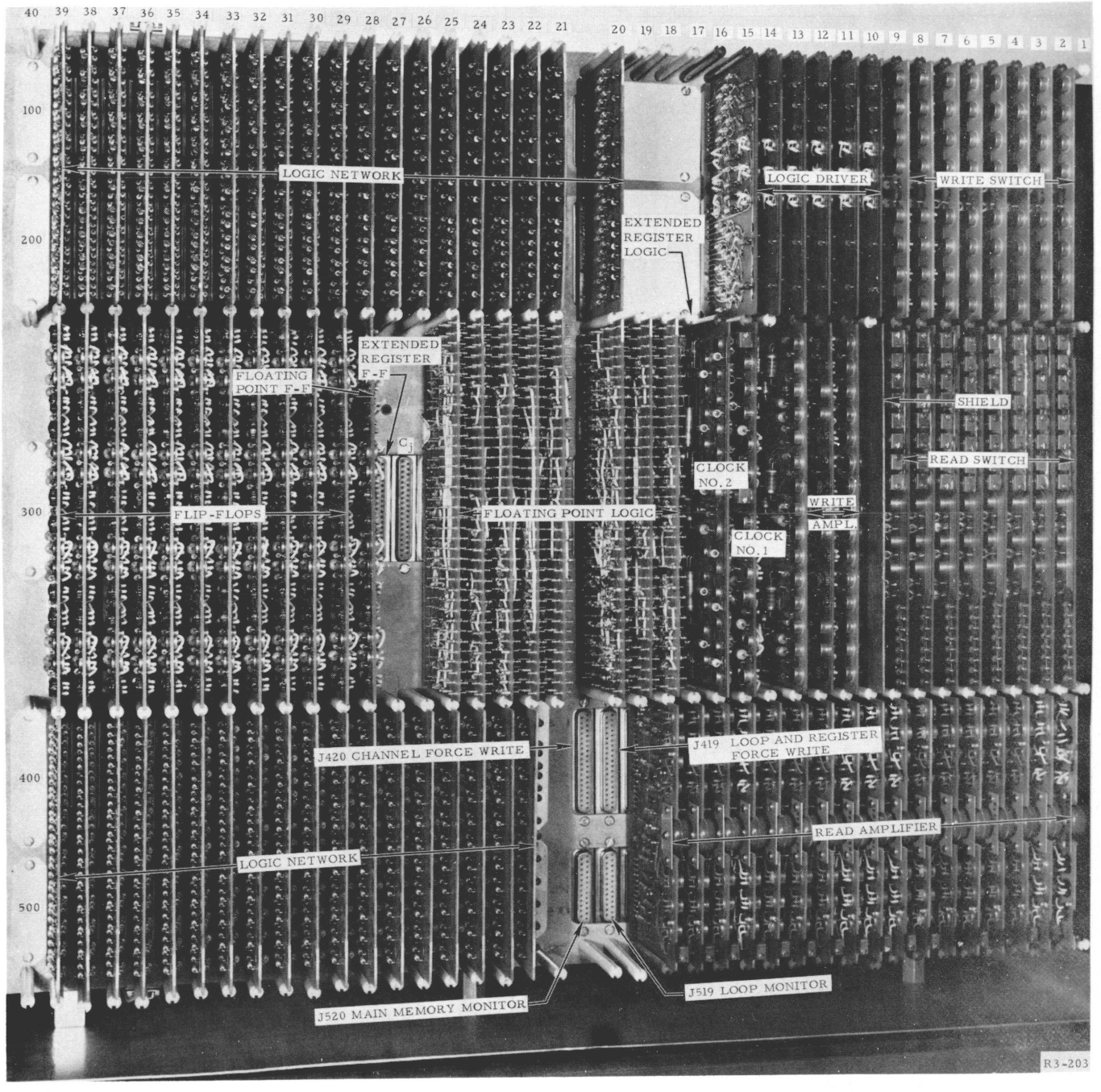


Figure 3-1. Computer Circuit Board Locations

The "and" gate has the characteristic of giving a true output signal only if all its inputs are true. "And-and" gates are used to mechanize logic involving cascaded "and" gates in which the first "and" gate is the primary gate and the second "and" gate is the secondary gate. "And-or" gates are used to complete the mechanization of most computer logic.

Flip-Flop

The flip-flop is a bi-stable circuit capable of storing a single bit (0 or 1) of information. Combinations of flip-flops are used for counting and storing, and for controlling various types of computer operations.

Logic Driver

Each RECOMP III logic driver circuit board contains three transistorized drivers which mainly provide supplemental ground current to those flip-flops used in a large number of logic gates. The logic drivers are applied to outputs of flip-flops or primary gates in such manner as to limit the ground current loading on a flip-flop to 80 milliamperes or less. Each logic driver board also contains several primary "and" gates; these gates are similar in configuration to those on the logic network boards. The primary gates on the driver boards are not connected on the board to the drivers but are externally wired to them if a primary gate has secondary loading sufficient to require the supplemental driving current.

Write Switching Network

Write switching network circuit boards contain circuitry for selection of a particular information channel write head.

Write Amplifier

Three write amplifier circuit boards, each containing three amplifier flip-flops, supply write head current for recording information on the memory disk. Two write amplifier flip-flops are common to each of the 64 information channels; the remaining amplifiers are used for the high-speed loops and the 1-word registers.

Read Switching Network

Each read switching network selects the output of one information channel read head from eight read head outputs during any given clock pulse time.

Read Amplifier

The read amplifier circuit boards amplify and shape signals from the memory read heads. RECOMP III used 17 read amplifier boards; eight for the 64 information channels, one each for the two 8-word loops (L and V), one each for the five 1-word registers, one for the clock channel, and one for the

origin-sector channel. For the information channels, read amplifier input signals come from the read switching network; for the other channels, input signals come directly from the read heads.

Arithmetic Elements

In performing its operations, the computer basically utilizes two elements: the arithmetic registers (A, B, and R) and an adder-subtractor formed by two flip-flops. Except for the registers, which are partially contained on the memory disk, the computation elements are all located on circuit boards of the computer. The same elements are used for both fixed and floating point operations. However, in RECOMP III, some additional logic circuitry is required to enable the computer's built-in floating point capability. The arithmetic elements are described in detail in sections 6 and 7 of this manual.

Control Console

The control console (figure 3-2) contains the switches and indicators for applying power to the computer system and for monitoring and controlling computer operations. In addition, the rear of the console (figure 3-3) has test jacks for monitoring timing and register signals. Dimensions of the RECOMP III control console are 11 inches wide, 7 inches deep, and 5 inches high. Weight is approximately 4 pounds.

Input-Output Unit

Standard input-output unit of the RECOMP III computer system is a modified Model FL Flexowriter (figure 3-4) incorporating a keyboard, printer, tape reader, tape punch, code selector, and code translator. Speed of operation of the tape reader, tape punch, and printer is a nominal 10 characters per second.

The engineering-type keyboard, used in entering information into the computer, has all numeric and alphabetic characters of an ordinary typewriter. Both the reader and punch are capable of accepting variously-coded tapes containing from five to eight channels of information. The keyboard and printing mechanism, however, operate only in the established 6-channel code.

When not in use as an input-output unit, the Flexowriter can be utilized for tape preparation, tape reproduction, and tape content printout. Tape reproduction from a master tape is possible in any code of from five to eight channels. When utilizing the keyboard for preparation of input tapes, and during printout of information read from a tape, only the established 6-channel code is usable.

The Flexowriter is 17 1/2 inches wide, 20 inches deep, 10 inches high, and weighs approximately 85 pounds.

Circuits for operation of this equipment are located within the unit and on input-output driver boards No. 1 and No. 2, and input-output logic board No. 1 in the input-output chassis of the desk.

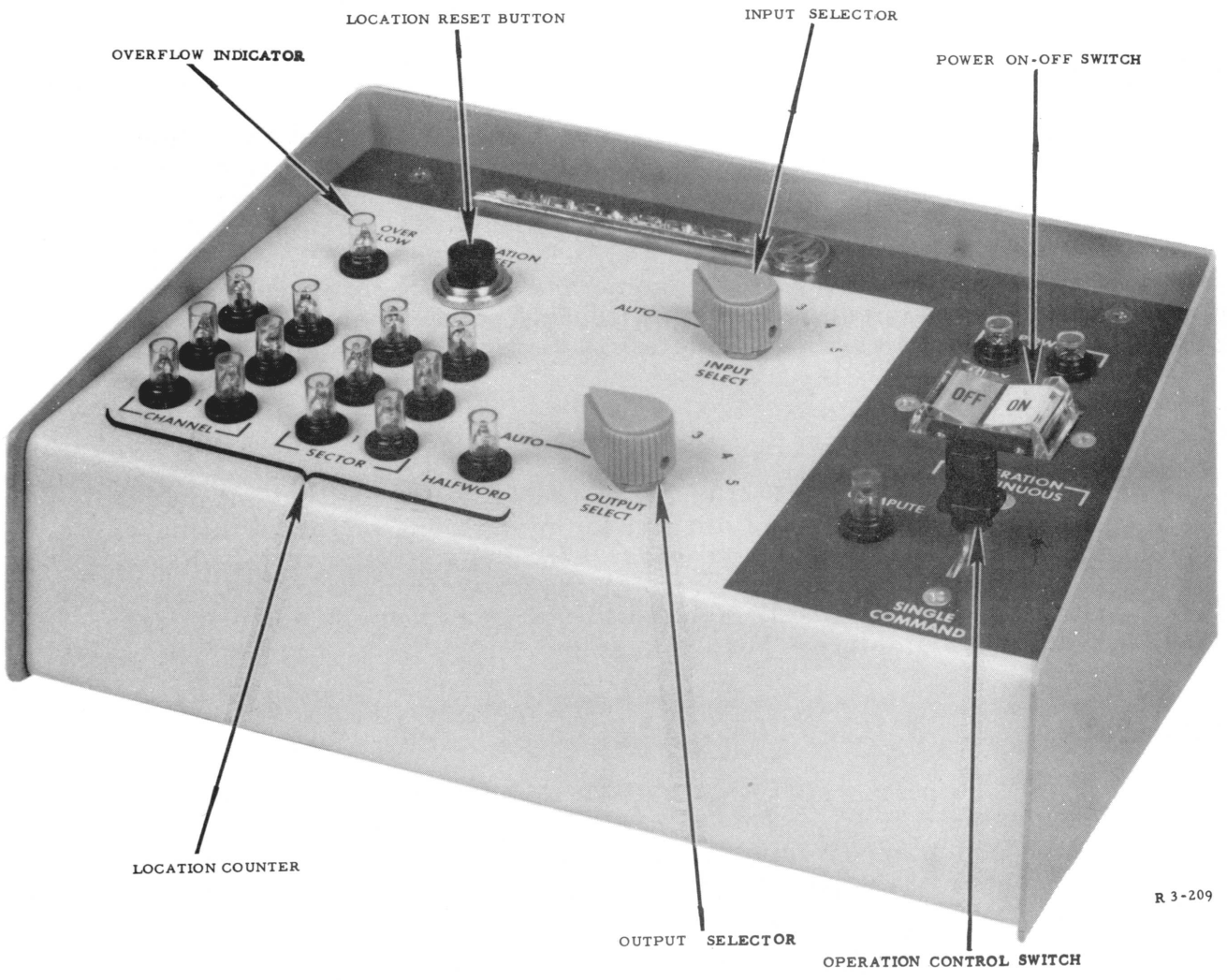


Figure 3-2. RECOMP III Control Console.

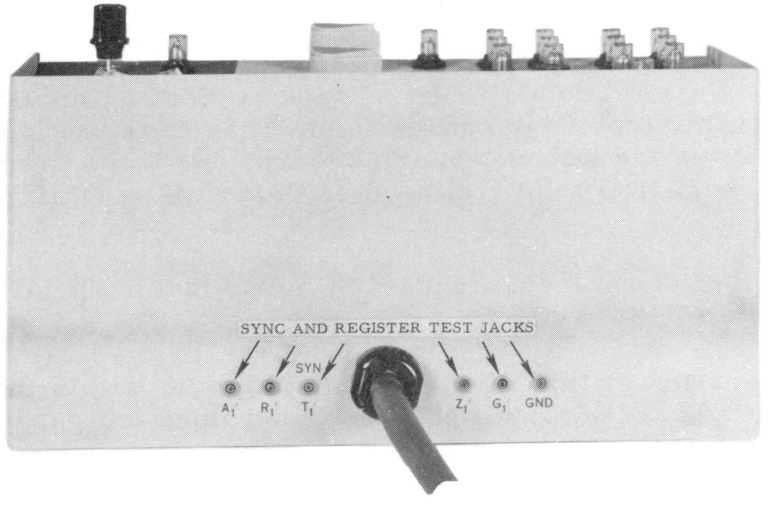


Figure 3-3. Control Console Rear.

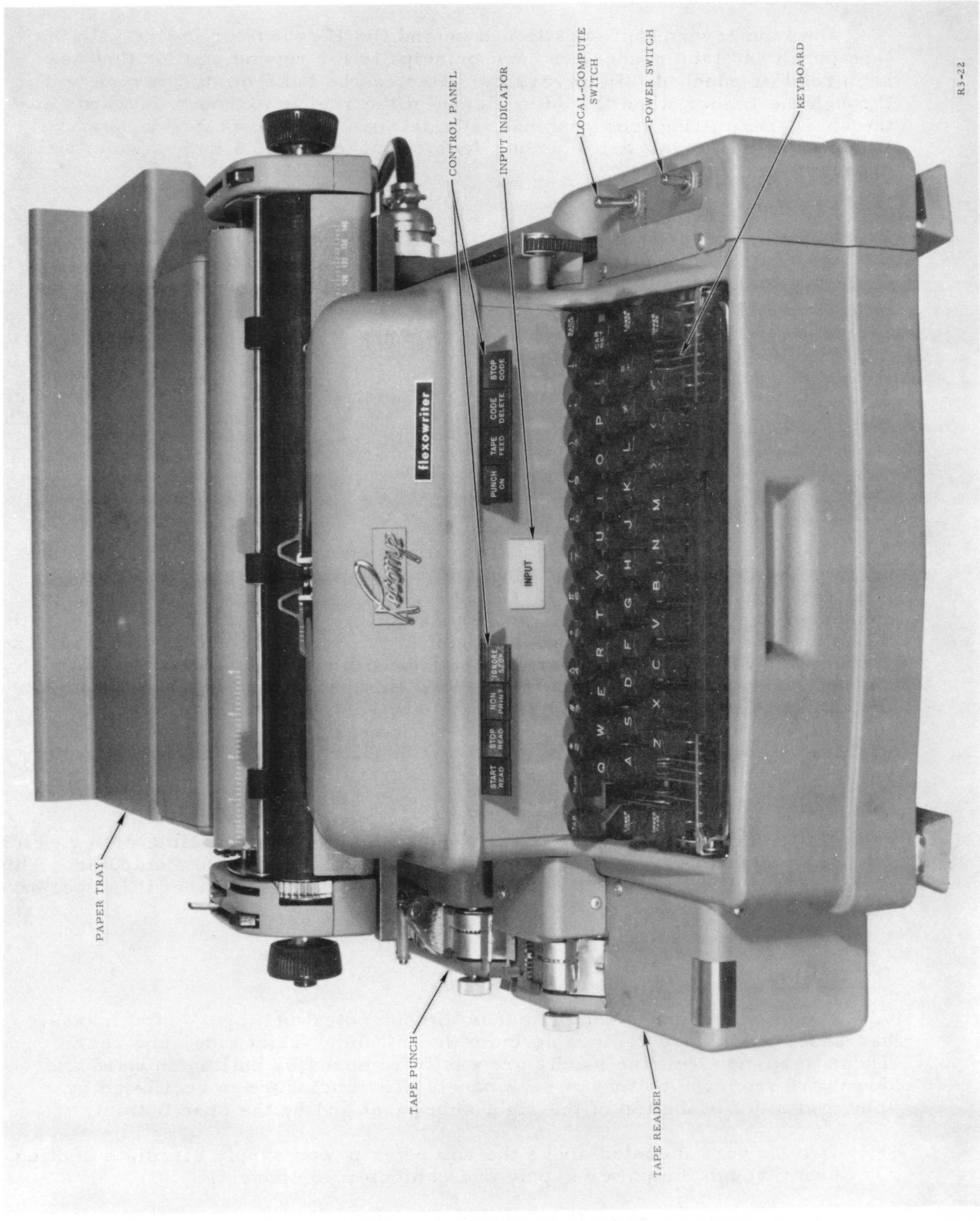


Figure 3-4 RECOMP III Flexowriter

RECOMP III Tape Rewind Unit

The tape rewind unit, positioned behind the Flexowriter in line with the tape punch and tape reader, is used principally for rewinding tape that has been read or punched. However, tape also can be fed from the rewind unit through the reader when the outer flange of the reel is extended outwards on the shaft, permitting free running. Diameter of the reel is five inches. Dimensions of the unit are 7 inches deep, 5 inches high, 6 inches wide; weight is approximately 6 pounds.

Desk

Principal purpose of the desk is to provide compact, neat-appearing housing and mounting for the computer and other elements of a RECOMP III computer system. The desk is designed to approximately the same overall dimensions as those of a standard size office desk. Structure of the desk frame, however, is made of welded steel to support the computer within the desk enclosure and for the control console, input-output unit, and tape rewind unit on top of the desk. Desk connectors and the non-computer circuits within the desk are described briefly in the following paragraphs.

Rear Connector Panel

The rear connector panel contains all receptacles for interconnections between the computer, control console, and input-output units (refer to illustration in Service manual). One 115-volt a-c main power receptacle is connected to source power; two accessory power jacks, wired in parallel with the main power receptacle, provide connections for input-output devices. The main power receptacle is covered to provide protection from high-voltage shocks. Receptacles J901 through J909 on this panel provide connections for the cables to the control console and all input-output equipment used in conjunction with the computer. The 5- or 8-channel tape selector switch (S1) enables circuitry for reading either 5- or 8-channel tape. The switch reverses the order of channels 1 through 5 except channel 3 which remains in the same location with either tape.

Also mounted on the rear connector panel is the running time meter which registers total hours and tenths of hours operating time of the computer. This meter is wired in parallel across the 115-volt a-c circuit so that it is operative only during power-on condition of the computer.

Desk Interior

The desk enclosure which houses the computer and input-output chassis has panels which are removable from the left side, right side, and rear. These snap-type release panels are easily removed by pulling outward and downward from the bottom of each panel. The enclosure is ventilated by louvered slots in the top of the right side panel and by the open bottom.

Two blowers mounted above the computer power supply circulate cooling air down through the power supply and computer components.

The input-output chassis is mounted on the inside surface of the rear connector panel within the desk interior (figure 3-5). Input-output logic and

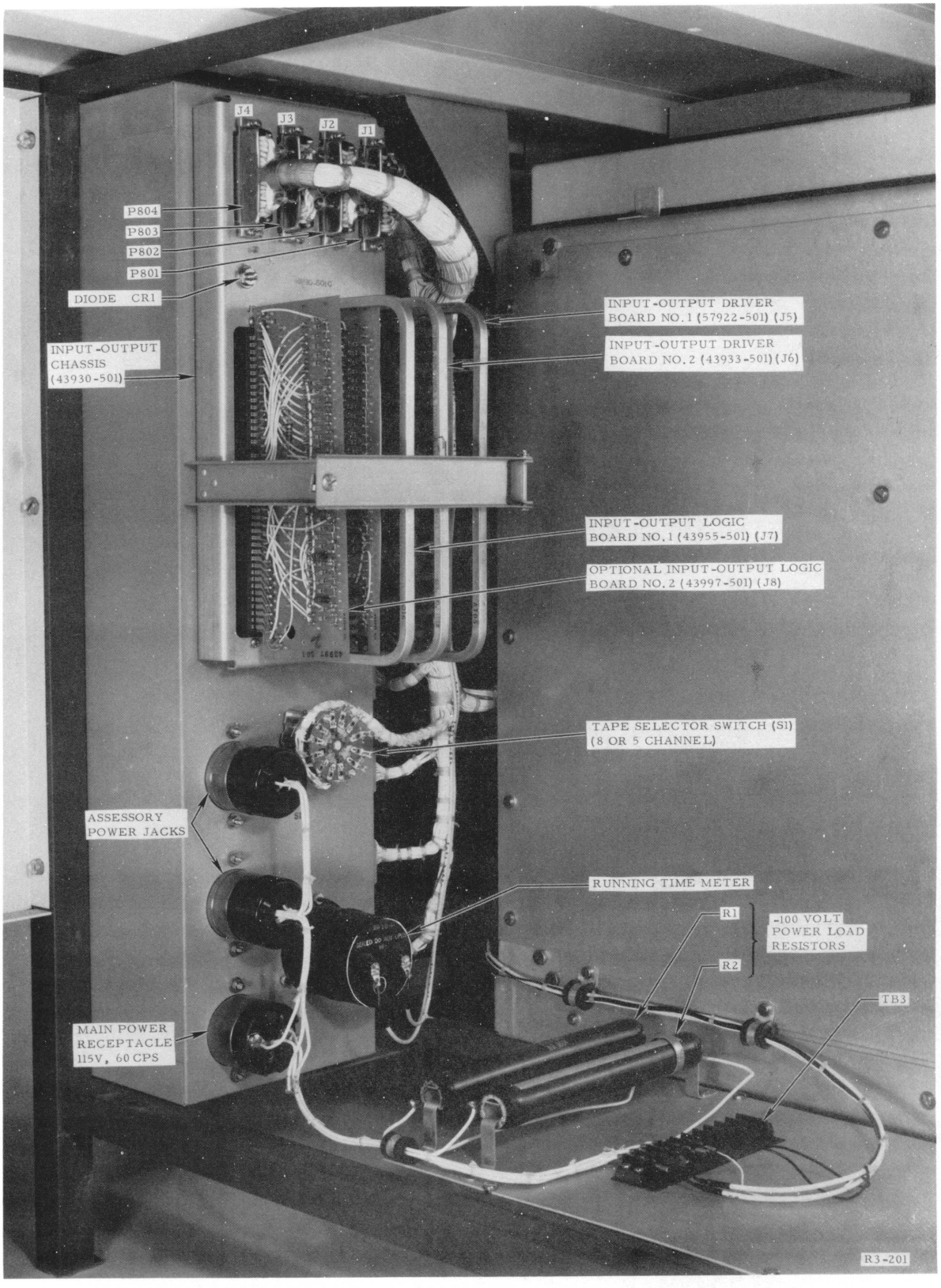


Figure 3-5. Desk Interior.

driver circuit boards plug into receptacles J5 through J8 which are wired to receptacles J1 through J4. Wire bundle plugs P801 through P804 plug into J1 through J4, respectively; wiring from these plugs is routed to the control panel and input-output equipment receptacles J901 through J909, to the power supply, and to the computer.

Located in the bottom of the enclosure is terminal board No. 3 (TB3), the terminals of which provide connections for 115-volt a-c source power from the power receptacle in the rear connector panel to the computer chassis, and connections for the -100-volt power load resistors (R1 and R2) located behind TB3. Power load resistor R2 is adjustable to compensate for voltage deviations which may be encountered at different installation facilities (refer to section 4 of Service Manual). Terminal board No. 3 (TB3) and power load resistors R1 and R2 are covered to provide protection against high-voltage shocks or burns.

Accessory Equipment for Enabling Optional Capabilities

RECOMP III computer is wired internally to accommodate the optional floating point arithmetic capability and to utilize various optional input-output devices. Additional computer circuit boards are required, however, as accessory equipment for implementation of these capabilities.

To provide the computer with floating point arithmetic capability, eight special logic circuit boards and one standard flip-flop circuit board are plugged into the computer chassis. Refer to figure 3-1 for identification and location of these boards.

When any input or output device other than the standard RECOMP Flexowriter input-output unit is used with the computer, one additional special input-output logic circuit board is plugged into connector J8 of the input-output chassis. This board, Input-Output Logic Board No. 2, provides logic gating circuits for all optional input and output devices used with a RECOMP III computer.

To provision the computer for input-output with a 12-channel card reader-punch, one additional standard flip-flop circuit board and one extended input-output register logic circuit board are plugged into the computer chassis in the locations shown on figure 3-1. Input-Output Logic Board No. 2 is also required to obtain this capability. The additional circuits on these three boards extend the computer's standard 8-channel input-output register and logic circuitry to enable 12-channel operations.

PHYSICAL SPECIFICATIONS

Size -- 30 inches high by 30 inches deep by 60 inches long (desk) plus approximately 6 cubic feet for standard input-output unit, control console, and tape rewind unit.

Weight -- Approximately 250 pounds, exclusive of input-output equipment.

Power Requirement -- Standard 115-volt, 60-cycle, single-phase, at 350 watts.

Permissible Line Voltage Variation -- 105 to 125 volts.

Ambient Temperature Operating Range -- 50° to 110 °F.

Permissible Internal Operating Temperature Range (Computer Only) -- 50 °F to 120° F; thermostatically-operated switch stops computer at 125 °F.

Air Conditioning Requirement -- Normally none.

Cooling -- Two blowers in computer.

Permissible Relative Humidity -- To 95 percent.

Circuits -- Solid state, with printed circuit boards used whenever possible.

Exterior Finish -- Beige, cocoa brown, madrone, and gold.

OPERATING SPECIFICATIONS

Type -- General purpose digital.

Internal Number System -- Binary.

Clock Frequency -- 153 kc (nominal).

Operating Mode -- Serial, from internally stored program.

Command Type -- Single address.

Word Length -- 41 bits (1 synchronization, 40 information including sign where applicable).

Programmed Word Formats -- Command (two commands per word), fixed point numeric, floating point numeric, and alphanumeric.

RECOMP III Commands -- 50 in standard system, 4 additional floating point arithmetic when system incorporates this capability; for listings and functions of commands executed, refer to tables 3-1 and 3-2.

RECOMP III Manuals Controls -- Power on-off, continuous and single cycle compute, halt, input select, output select, and location reset on control console; power on-off, local-compute, start read, stop read, non-print, ignore stop, punch on, tape feed, code delete, and stop code on standard input-output unit; on-off switch on tape rewind unit.

RECOM P III Operation Indicators -- On, ready, compute, overflow, and 13 location indicators on control console; input light on typewriter.

Table 3-1. Mnemonic-Octal Conversion Table for RECOMP III Computer Commands.

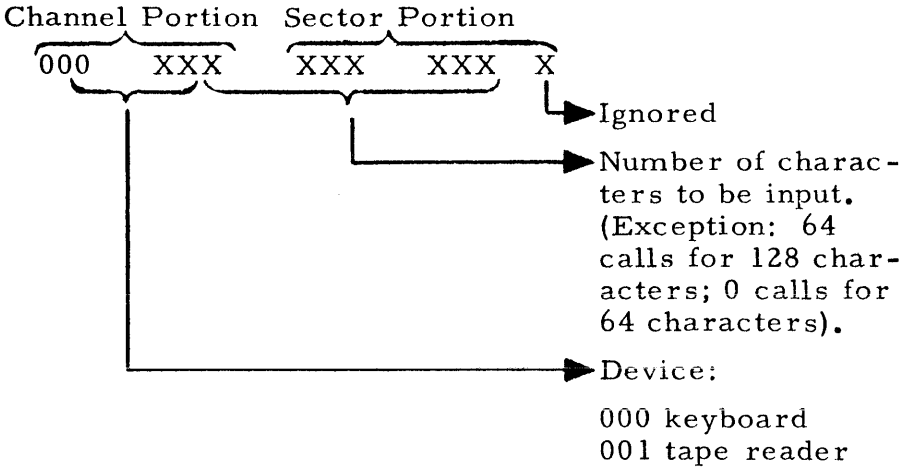
Mnemonic Symbol	Octal Operation Code	Mnemonic Symbol	Octal Operation Code	Mnemonic Symbol	Octal Operation Code
ADD	73	CLA	37	CTV	41
ALS	02.0	CLS	36	DIV	66
ARS	03	CME	35.1	EXT	70
ASC	02.4	CMG	35.0	FAD	(75)
ASV	02.2	CMP	17..0	FDV	(26)
CAR	57	CMZ	16.4	FMP	(23)
CAZ	16.0	CSA	13	FSB	(74)
CFL	44	CTL	40	HTR	71

Table 3-1. Mnemonic-Octal Conversion Table for RECOMP III Computer Commands (Continued)

Mnemonic Symbol	Octal Operation Code	Mnemonic Symbol	Octal Operation Code	Mnemonic Symbol	Octal Operation Code
ICH	00	RCA	77	SUB	72
LDI	31.1	RCS	76	TIX	10
LLS	42.0	RND	17.4	TLB	15
LRS	43	SAN	12.4	TMI	53
LSC	42.4	SAP	12.0	TNZ	11
LSV	42.2	SLC	25.0	TOV	52
MPY	63	STA	65	TPL	55
NOP	54	STI	25.1	TRA	51
OCH	01	STO	45	TZE	50
OVN	13.4	STR	05	XAR	56

Table 3-2. List of RECOMP III Computer Commands.

Octal Operation Code	Mnemonic Symbol	Description
00	ICH	Input 1 to 128 8-bit characters (all but last 5 are lost) into A from input device named in address (or set into input switch on the console). The console switch overrides device selected in address, so it should be set to AUTO to allow program to specify its input device(s). The form of the address is as follows:

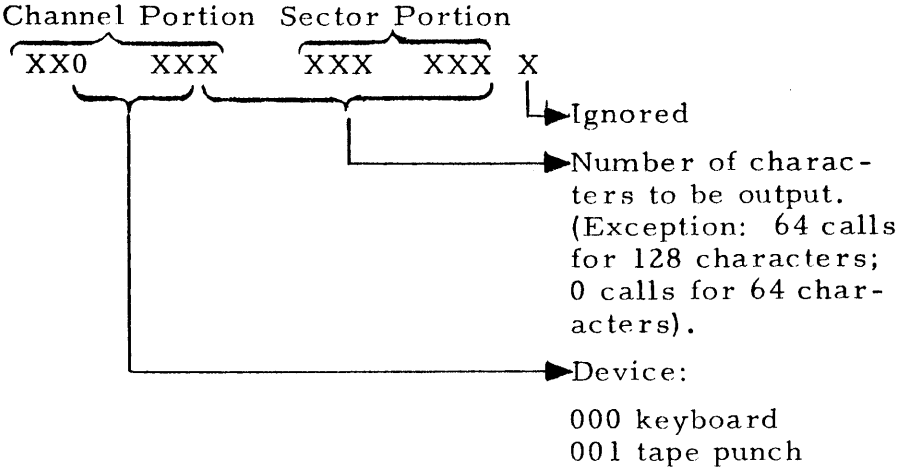


As each character is input into right-most 8 bits of A, A is shifted left 8 bits through (including) sign position. Since typewriter keyboard uses a 6 bit code, the high 2 bits of each character input from it will be zeros. Either of the tape readers will read 8 bits per character with a hole entering as one and no-hole as zero.

(If first 2 bits of address are not zeros, first character input will have its low two bits forced to ones corresponding to the ones in the address).

Table 3-2. List of RECOMP III Computer Commands (Continued)

Code	Symbol	Description
01	OCH	Output 1 to 128 characters from A to output device named in address (or set at output switch on console). Console switch overrides device selected in address, so it should be set to AUTO to allow program to specify its output device(s). The form of the address is as follows:



As each character is output from the 8 left-most bit positions (including sign) of A, A is shifted left 8 bits and the 8 bits of last previous character output enter the right end of A. After output of the first character, these bits will be 00100 1XX, where XX are the first 2 bits of address. In general, any output of over 5 characters will be meaningless, although 5 out of every 6 characters will be the original A-register contents.

In output to Flexowriter keyboard, the 2 high bits of each character are ignored. Characters with no key representation are recorded as blanks.

02	ALS	Shift contents of A left number of places in least significant bit of channel portion and 6 bits of sector portion of address (0 to 127 decimal). Bits leaving left end of A are lost. Bits entering right end of A will be zeros. Sign position of A is neither shifted nor affected.

NOTE

ALL SHIFT COMMANDS UTILIZE THE SAME SEVEN BITS TO INDICATE THE AMOUNT OF SHIFT, EITHER 12-18 FOR A LEFT-HALF COMMAND OR 32-38 FOR A RIGHT-HALF COMMAND.

Table 3-2. List of RECOMP III Computer Commands (Continued)

Code	Symbol	Description
*02.2	ASV	Shift contents of A left amount of shift specified. Turn overflow indicator on if any of bits lost from left end of A were one.
*02.4	ASC	Shift contents of A left amount of shift specified or until A is normalized (left-most bit of A is a one), whichever occurs first. Decrement I by number of places actually shifted. (I = index register).
03	ARS	Shift contents of A right amount of shift specified. Bits leaving right end of A are lost. Bits entering left end of A will be zeros.
05	STR	Replace contents of word addressed by contents of R.
10	TIX	Subtract 1 at b = 38 from index register. If index register is not now zero, take next command from half word specified in address. This command (the TIX) must be in a left half word itself, but it can transfer to either half word.
11	TNZ	If contents of A are not equal to zero, take next command from half-word specified in address.
12	SAP	Make sign of A positive.
*12.4	SAN	Make sign of A negative.
13	CSA	Reverse sign of A.
*13.4	OVN	Turn Overflow indicator on.
15.	TLB	If lowest order (right-most) bit of A is 1, take next command from half-word specified in address.
16.	CAZ	Replace contents of A with positive zero.
*16.4	CMZ	Replace contents of A with negative zero.
17	CMP	Complement A. (Reverse every bit of A including the sign).
*17.4	RND	If highest order bit of R is 1, increase magnitude of A by 1 in the least significant position. If overflow occurs, overflow indicator will turn on and sign A will be reversed.
23	FMP	Multiply A by floating point word addressed. R is destroyed. If product lies outside permissible range, turn overflow indicator on.

Table 3-2. List of RECOMP III Computer Commands (Continued)

Code	Symbol	Description
25	SLC	Replace left-hand address portion of word addressed with complement of location counter (which will contain the location of this command).
*25.1	STI	Replace right-hand address portion of word addressed with contents of index register.
26	FDV	Divide A by floating point word addressed. R is destroyed. If quotient lies outside permissible range, turn overflow indicator on.
*31.1	LDI	Replace contents of index register with right-hand address portion of word addressed. Half word bit of address of this command must be one to effect loading of index.
35	CMG	Compare contents of A with contents of word addressed. Turn overflow indicator on if word addressed is arithmetically greater than contents of A. Turn the overflow indicator off if the word addressed is arithmetically less than the contents of A. If word addressed and contents of A are both positive and equal, do nothing. If word addressed and contents of A are both negative and equal, reverse overflow indicator. For the purpose of this command, plus zero is considered to be greater than minus zero.
*35.1	CME	If contents of A are not equal to word addressed, turn overflow indicator on. For the purpose of this command, plus zero does not equal minus zero. Note that, although the half word bit of the address is 1, this command refers to a full word.
36	CLS	Clear and subtract. Replace contents of A with negative of contents of word addressed.
37	CLA	Clear and add. Replace contents of A with contents of word addressed.
40	CTL	Replace words in L-loop, starting with word whose least significant address digit is the same as that of address, with consecutive words starting with word addressed, until 7767 is filled. For example: CTL 12300 will replace words 7760-7767 with words 1230-1237. CTL 12360 will replace words 7766 and 7767 with words 1236 and 1237. CTL 12370 will replace word 7767 with word 1237.

Table 3-2. List of RECOMP III Computer Commands (Continued)

Code	Symbol	Description
41	CTV	Replace words in V-loop until word 7777 is filled. This command is completely analogous to CTL 40.
42	LLS	Shift contents of A and contents of R left amount of shift specified. Make sign of A same as sign of R. Bits leaving left end of A are lost. Bits entering right end of R will be zeros. Bits leaving left end of R enter right end of A.
*42.2	LSV	Shift contents of A and contents of R left amount of shift specified. Turn overflow indicator on if any of bits lost from left end of A were 1. Make sign of A same as sign of R.
*42.4	LSC	Shift contents of A and contents of R left amount of shift specified or until A is normalized (leftmost bit of A is 1), whichever occurs first. Decrement I by number of places actually shifted. Make sign of A same as sign of R.
43	LRS	Shift contents of A and contents of R right the amount of shift specified. Make sign of R same as sign of A. Bits leaving right end of R are lost. Bits entering left end of A are zero. Bits leaving right end of A enter left end of R.
44	CFL	Replace words in memory from L-loop. Analogous to CTL 40.
		NOTE COPY FROM OR TO L COMMANDS WITH LOOP ADDRESSES (7760-7777) WILL READ INTO OR FROM THE SO-CALLED "GRAY" AREA WHICH IS THE LAST 16 WORD SECTORS OF CHANNEL 77. HOWEVER, THE CTV COMMAND CANNOT BE USED TO COPY FROM THE "GRAY" AREA.
45	STO	Replace contents of word addressed by contents of A.
50	TZE	If contents of A are equal to zero (plus or minus), take the next command from half-word specified in address.
51	TRA	Transfer. Take next command from half-word specified by address instead of sequentially. A half-word bit of 1 indicates right-half command.
52	TOV	If overflow indicator is on, turn it off and take next command from half-word specified in the address.

Table 3-2. List of RECOMP III Computer Commands (Continued)

Code	Symbol	Description
53	TMI	If sign of A is negative, take next command from half-word specified in address.
54	NOP	No operation. This command has no effect.
55	TPL	If sign of A is positive, take next command from half-word specified in address.
56	XAR	Exchange contents of A with contents of R.
57	CAR	Replace contents of R with contents of A.
63	MPY	Mulitply contents of A by contents of word addressed, and replace contents of A and R with product. Both A and R will have sign of product.
65	STA	Replace address portion of half word addressed by corresponding address portion of A.
66	DIV	Divide contents of A and R (dividend has sign of A) by contents of word addressed and replace contents of A by quotient and contents of R by remainder (with original sign of A). If overflow occurs, overflow indicator will turn on.
70	EXT	Extract (bitwise logical and). Compare contents of A with contents of word addressed bit by bit, including sign position. Whenever corresponding bits of A and word addressed are both 1's, leave 1 in A. Whenever either of the corresponding bits of A and word addressed is zero, put zero into that position of A. (A plus sign is one; a minus sign is zero).
71	HTR	Halt. When computing is resumed, take next command from half-word specified in address.
72	SUB	Subtract arithmetically contents of word addressed from contents of A and put difference in A. If overflow occurs, overflow indicator will turn on, and sign of A will be reversed.
73	ADD	Add arithmetically contents of word addressed to contents of A and put sum in A. If overflow occurs, overflow indicator will turn on, and sign of A will be reversed.
74	FSB	Subtract floating point word addressed from A. R is destroyed. If difference lies outside permissible range, turn overflow indicator on.
75	FAD	Add floating point word addressed to A. R is destroyed. If sum lies outside permissible range, turn overflow indicator on.

Table 3-2. List of RECOMP III Computer Commands (Continued)

Code	Symbol	Description
76	RCS	Replace the contents of R with contents of A. Then replace contents of A with negative of contents of word addressed.
77	RCA	Replace contents of R with contents of A. Then replace contents of A with contents of word addressed.

NOTE

* THE OPERATION CODES USUALLY CONSIST OF TWO OCTAL DIGITS EACH. HOWEVER, SOME OPERATION CODES UTILIZE THE HIGHEST, NEXT TO HIGHEST, OR LOWEST BIT OF THE ADDRESS. THESE ARE LISTED AS XX.4, XX.2, OR XX.1, RESPECTIVELY.

Section 4
OPERATING PROCEDURES

GENERAL

Basic procedures for operation of the computer system in normal usage are presented in this section. Those procedures concerning operation during maintenance are given in the service manual. Procedures in this section cover pre-operating preparations, operating precautions, and system and input-output equipment operating procedures. Controls and their functions are listed in table 4-1.

Operation of a standard computer system in the compute mode requires proper positioning of switches on both the control console and input-output equipment.

Preparation and/or reproduction of RECOMP III tapes (Flexowriter local mode of operation) utilizes only switches of the Flexowriter, with power being supplied through the power ON/OFF switch at the right side of the keyboard. Control of the RECOMP III tape rewind unit, which is connected to the auxiliary power receptacle on the Flexowriter, is through an ON/OFF switch near the tape reel.

PRE-OPERATING PREPARATIONS

Usually, preparation of the computer system for operation consists of (1) verifying interconnections between elements of the system, (2) providing tape for the punch and paper for the printer as required, (3) connecting the system to the power source and (4) selecting the appropriate position of the 5- or 8-channel tape select switch. Before connecting the system to the power source, the system should be checked to ensure that all controls are in off or neutral position.

In addition to the preceding customary preparations, pre-operating steps may also, on occasion, be extended to include (1) operation of the input-output units in local mode of operation and (2) verification of the system's operational status through execution of the RECOMP III origin-sector channel diagnostic test routine.

OPERATING PRECAUTIONS

Several operating precautions must be adhered to as a safeguard against possible damage to the computer system. Following are the major operating precautions:

Do not restrict the flow of air around the computer by storing boxes or cabinets in front of the air vent.

Ensure that the disk memory is completely stopped prior to moving the computer by allowing at least 15 minutes from power turnoff for complete deceleration.

Do not bump or jar the computer during the time the disk memory is accelerating or decelerating.

Always follow the established procedure for applying power to the computer system to ensure that the system will not be affected by power transients.

Always remove power from the computer system according to the established procedure; never unplug the power cable from the wall receptacle to remove power.

After removing power from the computer system, do not reapply power until after 10 seconds have elapsed; the memory disk must completely drop away from the headplate before reapplying power to ensure against damage to the disk.

Never operate the computer at temperatures below 50 degrees F.; the air thrust bearing in the computer memory does not function properly below this temperature and damage to the memory may result.

If an auxiliary source is to supply power either permanently in place of or temporarily as an emergency substitute for line power, the computer should be modified with a failsafe circuit to ensure against possible damage to the memory resulting from sudden and excessive voltage and frequency fluctuations. The modification is relatively minor and should be made prior to installation of a system.

Do not touch electronic components with hands as poor connections will result from grease or grime.

Do not operate the LOCATION RESET switch on the control console for computer synchronization or origin-sector channel readout until the OPERATION switch has been set to halt position (refer to control console description in Section 6, Functional Description).

SYSTEM OPERATING PROCEDURES

General procedures for manually controlling computer system operation are presented in the following paragraphs. Operating procedures special to execution of specific programs are normally included with the descriptions of such programs.

Computer System Start Procedure

Following are the procedures for starting the computer system:

1. On RECOMP III control console, ensure that OPERATION selector switch is at halt position.

2. On control console, depress POWER ON switch (POWER ON indicator lights immediately).

NOTE

AFTER APPROXIMATELY 40 SECONDS, THE READY INDICATOR LIGHTS, SIGNIFYING THAT THE COMPUTER IS NOW READY FOR OPERATION.

3. On input-output equipment, place switches at proper positions for compute mode of operation and type of program to be executed.

Computer System Stop Procedure

To shut down the computer system proceed as follows:

1. On RECOMP III control console, place OPERATION selector switch at halt position.
2. On input-output equipment, place power ON/OFF switches at OFF.
3. On control console, place POWER ON/OFF switch at OFF.

NOTE

WHEN POWER IS REMOVED, POWER ON AND READY INDICATORS EXTINGUISH IMMEDIATELY AND MEMORY DISK RETRACTS FROM PROXIMITY OF HEAD PLATE.

CAUTION

THE COMPUTER SHOULD NEVER BE MOVED WHILE THE MEMORY DISK IS ROTATING. ALLOW AT LEAST 15 MINUTES AFTER POWER REMOVAL FOR MEMORY DE-CELERATION BEFORE MOVING THE COMPUTER.

NEVER DEPRESS THE POWER ON SWITCH ON THE CONTROL CONSOLE FOR RESTART UNTIL AFTER 10 SECONDS HAVE ELAPSED.

Computer Operating Procedures

Most computer operations are directed by internally stored programs. However, some operations can be manually directed from the control console. Also in the RECOMP III system, when operating under program control, certain operations can be directed from an input-output unit through an internally stored program, such as from the Flexowriter through the RECOMP III Load-Start Program. Functioning of the operating controls and indicators is described in Section 6, Functional Description.

Regardless of the type of input-output device used with the computer, all switches or controls must be at off or neutral position prior to applying power. During operation, all switches establishing compute or local mode of operation and input-output device designation must be at the proper position before beginning computations or off-line operations. Procedures for operating the input-output equipment are presented separately for the compute and local modes of operation. Table 6-3 lists the switch positions which select the various RECOMP III input-output devices.

Use of RECOMP II Format Tapes

A RECOMP II Format Tape Input program, when utilized in conjunction with the 5-channel position of the RECOMP III tape select switch, enables input to RECOMP III of RECOMP III programs punched in 5-channel RECOMP II format. The RECOMP III program may have been prepared in either RECOMP II command, alphanumeric, or location format. Tapes may have been prepared on a tape preparation device such as the VersaTape or a Baudot-coded Flexowriter, or produced by the RECOMP II computer system in either its off-line or on-line modes of operation. The RECOMP II format is then converted to a RECOMP III version and the data stored in the RECOMP III memory. Conversion is effected without repunching and subsequent re-reading of the data by a RECOMP III Flexowriter or other input-output device.

The RECOMP II Format Tape Input program occupies 53 memory locations and is relocatable. It ignores leading tape feed and accepts L, C, F, S and H formats as described in the RECOMP II programming manual. A/ or an N are not accepted. Multiple carriage returns are handled the same as by the RECOMP II. Letter shift and figure shift characters (RECOMP II octal codes 33 and 37) are ignored except in alphanumeric format.

Entry of the RECOMP II Format Tape Input program into RECOMP III requires use of the Load-Start Program (refer to section 5 of RECOMP III Service Manual). The procedure for input into RECOMP III from a 5-channel RECOMP II format tape is as follows:

1. Ensure that Load-Start Program is stored in memory.
2. Enter RECOMP II Format Tape Input Program into memory.
3. Load prepared RECOMP III program punched on 5-channel tape into tape reader.

CAUTION

TAPE MUST BE POSITIONED WITH THE THREE CODE HOLES INWARD OF THE FEED (SPROCKET HOLE) PINS.

4. Adjust tape table guide against outer edge of tape.
5. Place tape select switch (in desk well) at 5.

6. On control console, verify that OPERATION selector switch is at halt position.

7. Verify that INPUT SELECT switch on control console is at AUTO position.

8. On control console, depress and release LOCATION RESET button, then place OPERATION selector switch at CONTINUOUS position.

9. On keyboard, strike sxxxxx followed by carriage return (where xxxxx represents first memory location of RECOMP II Format Tape Input program).

NOTE

THE TAPE CONTAINING THE RECOMP III PROGRAM MUST DESIGNATE THE FIRST MEMORY LOCATION INTO WHICH THE PROGRAM WILL BE ENTERED.

10. When loading of RECOMP III program is completed, place OPERATION selector switch at halt position and tape select switch at 8.

RECOMP II Format Tape Input Program Printout

NOTE

MEMORY LOCATIONS LISTED IN THE FOLLOWING PRINTOUT ARE FOR PURPOSES OF EXAMPLE ONLY.

```
1000  -4010000+4110100
   1  +5177641+4300040
   2  +3777600+4220030
   3  +5277640+1240000
   4  +4577600+0002010
   5  +7010110+0200250
   6  +7210070+4577670
   7  +3610200+5310631
1010  +1577641+4577710
   1  -0000000-0000171
   2  +1077611+3110271
   3  +4300020+3777600
   4  +4220010+5177630
   5  +7777760+7210470
   6  +6577761+0570000
   7  +3110461+5277641
1020  -6577761+5110331
   1  -3777710+5400030
   2  -6510341+5110341
   3  -3710460+5110570
   4  +7777760+5110610
   5  -3777760+5110260
   6  +6577741+5177741
   7  -3777710+5400070
1030  -3777600+5177750
   1  +3607411+4110260
   2  -3777710+5400040
1033  +4577750+3110401
   4  +5177641+7177240
   5  -3710240+5110370
   6  -3710150+5110330
   7  +5110571-0000000
1040  -3777710+5400050
   1  -3710470+5110570
   2  -3710200+5110330
   3  -3777710+5400020
   4  -3777760+5110220
   5  -3777710+5400060
   6  -3777710+5400000
   7  -3777710+5400010
1050  +3177651+0002010
   1  +4300050+3777670
   2  +4200040+4220010
   3  -1200000+5277641
   4  +1240000+4577670
   5  +1077601+5400100
   6  +4010010+5177640
   7  -5177721+4577751
1060  +4010500+5177600
   1  +6510621+7210470
   2  +6577761+0530170
   3  +5110600+3710310
   4  +0100050+7177641
```

Flexowriter Operating Procedure

The Flexowriter is the standard input-output device used with the RECOMP III computer system. Operating procedures for this equipment are presented in following paragraphs.

Compute Mode of Operation

The compute mode of operation with the Flexowriter as the input-output device utilizes only the power ON-OFF and LOCAL-COMPUTE switches of the Flexowriter. Operating procedures for this mode are normally included with the program-execution instructions accompanying computer programs and routines.

Local Mode of Operation

When not in use as an input-output device, the Flexowriter can be utilized for tape preparation, tape reproduction, and tape content printout.

To prepare a tape, use the following procedure:

1. Place Flexowriter power switch at ON.
2. Place LOCAL-COMPUTE switch at LOCAL.
3. Depress PUNCH ON switch.
4. Operate keyboard as required for desired format until tape preparation is completed.
5. Return all switches to off or neutral conditions.

To reproduce a tape, with or without simultaneous tape content printout, proceed as follows:

1. Insert tape to be reproduced into tape reader.
2. Place Flexowriter power switch at ON.
3. Place LOCAL-COMPUTE switch at LOCAL.
4. Depress PUNCH ON switch.
5. Depress IGNORE STOP switch.

NOTE

DEPRESSING THE IGNORE STOP SWITCH BEFORE STARTING THE READING PROCESS CAUSES THE READER TO IGNORE STOP CODES ON TAPE AND CONTINUE NORMAL OPERATION.

6. Depress NON-PRINT switch.

NOTE

IF TAPE CONTENT PRINTOUT IS DESIRED SIMULTANEOUSLY WITH TAPE DUPLICATION, ALL SWITCH SETTINGS ARE THE

SAME EXCEPT START READ IS DEPRESSED INSTEAD OF NON-PRINT. A PRINTOUT WILL BE INTELLIGIBLE AND REPRODUCTION CORRECT ONLY IF THE TAPE CONTENT IS IN THE STANDARD RECOMP III 6-CHANNEL CODE.

7. Return all switches to off or neutral condition upon completion of reproducing tape.

To produce tape content printout without punching tape, perform the following procedure:

1. Place Flexowriter power switch at ON.
2. Place LOCAL-COMPUTE switch at LOCAL.
3. Depress START READ switch.

NOTE

IF A STOP CODE IS READ DURING TAPE CONTENT PRINTOUT, THE START READ SWITCH MUST BE DEPRESSED TO RESUME READER OPERATION. DEPRESSING THE IGNORE STOP SWITCH CAUSES THE READER TO IGNORE STOP CODES ON TAPE AND CONTINUE NORMAL OPERATION. A TAPE CONTENT PRINTOUT WILL BE INTELLIGIBLE ONLY IF TAPE CONTENT IS IN THE STANDARD RECOMP III 6-CHANNEL CODE.

4. Depress STOP READ switch to stop reader operation.
5. Return all switches to off or neutral condition.

Table 4-1. RECOMP III Operating Controls and Indicators with Functions

Control or Indicator	Ref. Des.	Location	Functions	
			Compute Mode	Local Mode
POWER ON/OFF	S1	Control Console	Applies and removes power to and from computer.	
OPERATION	S2	Control Console	Selects CONTINUOUS, SINGLE COMMAND, or halt action of computer.	
LOCATION RESET	S5	Control Console	Generates origin-sector channel read signal when fully depressed; during depression and release generates signal resetting location counter to zero.	

Table 4-1. RECOMP III Operating Controls and Indicators with Functions (Continued)

Control or Indicator	Ref. Des.	Location	Functions	
			Compute Mode	Local Mode
INPUT SELECT	S3	Control Console	Permits automatic designation, or manually selects input device for operation with computer.	
OUTPUT SELECT	S4	Control Console	Permits automatic designation, or manually selects, output device for operation with computer.	
POWER ON INDICATOR	L1	Control Console	Indicates application of 115-volt a-c power to computer through contacts of POWER switch.	
READY INDICATOR	L2	Control Console	Indicates computer-ready condition.	
COMPUTE	L3	Control Console	Indicates compute state.	
OVERFLOW	L4	Control Console	Indicates overflow condition in computer.	
Location Counter Display	L5 thru L17	Control Console	Indicates memory address of next instruction to be executed when computer is in halt condition.	
ON/OFF	S1	Flexowriter	Controls applications of 115-volt a-c power to Flexowriter.	
LOCAL/ COMPUTE	S2	Flexowriter	Selects mode of operation of Flexowriter.	
START READ	S3	Flexowriter		Automatically starts tape reader operation.
STOP READ	S4	Flexowriter		Stops tape reader operation.
NON-PRINT	S5	Flexowriter		Allows direct reproduction of tape from reader to punch without producing printed copy.

Table 4-1. RECOMP III Operating Controls and Indicators with Functions (Continued)

Control or Indicator	Ref. Des.	Location	Functions	
			Compute Mode	Local Mode
IGNORE STOP	S6	Flexowriter		Causes reader not to recognize and stop on STOP code.
PUNCH ON	S7	Flexowriter		Energizes punch circuits to standby mode, ready to punch tape on receipt of information from reader or keyboard.
TAPE FEED	S8	Flexowriter		Causes tape feed (punches sprocket holes only) as switch is depressed.
CODE DELETE	S9	Flexowriter		Punches all six channels of tape to delect a code punched in error.
STOP CODE	S10	Flexowriter		Punches a code which, when read by reader, causes reader to stop.
Tape Select	S1	Desk Well	Selects 5-or 8-channel tape circuitry during input operations.	
INPUT	PL	Flexowriter		Indicates when computer is ready for input from keyboard or tape reader.
On/Off	S1	Tape Rewind	Operates tape rewind unit.	
Circuit Breakers	CB1, CB2	Power Supply (In Desk)	Provides a-c circuit protection for computer.	

Section 5
PRINCIPLES OF OPERATION

GENERAL

Basically, all general purpose digital computing systems contain the same major elements and operate according to the same general principles. The major elements always present are input, storage, arithmetic and information processing, output, control, and power supply. General functional principles employed by virtually all systems in most of their operations are, briefly:

Coded information, consisting of both commands and data, is entered from an input device into the computer's internal storage element.

The commands and data are retrieved from internal storage and utilized for computations or processing.

As desired, the result of computation or processing is either (1) placed back into internal storage for further computation or processing, or for output to an external device or (2) is retained in the arithmetic element for immediate use in additional computation or processing.

When desired, information is withdrawn from internal storage and transmitted out of the computer to an output device which records the data in either a coded or binary form.

RECOMP is the same as the other general purpose digital computing systems in these basic general aspects. However, in the type of major elements used and in the manner of effecting the general functional principles, RECOMP differs somewhat from other systems.

This section describes, in general terms, the manner in which RECOMP effects its various operations. Because the general theory of operation can be more readily understood with the supposition that information is already in the computer system, the discussion assumes that commands and data have been entered into storage from an input device and that the computation or processing will be performed on that data. Detailed explanations of the operations discussed are given in Section 6, Functional Description, and Section 7, Logic Description.

Information on the specific type of elements comprisng the RECOMP system is contained in section 3 and, consequently, is omitted from this discussion. However, familiarity with the characteristics of these elements as well as a general acquaintance with the operations ordered by the various commands (section 3), is essential to understanding the information presented in subsequent paragraphs.

OPERATIONS CONTROL

The operations control function ensures that all actions are performed in specific predetermined action sequences, with variations possible only as a result of predetermined circumstances. Both the predetermined sequences and the variations in sequence are established by specific signals from various electronic components and electromechanical parts such as flip-flops, gates, cam-actuated contacts, and switches. Most of these control signals include, as an integral part, timing signals to establish the exact time at which a specific sequence will begin or terminate. Thus, when the system is functioning correctly, a specific action sequence can be caused only by one unique signal or combination of signals.

The major action sequences are commonly referred to as operating modes. These modes, signified in RECOMP III by the states of several flip-flops, are designated sync, idle, command, direction, operand, execute, input, output, and display. Functions of these modes are summarized in table 5-1.

Action sequences within the major modes are called sub-modes or mode phases, and are specified in RECOMP III by the states of two flip-flops. The phases are designated 1 through 4. As in the major modes, the phase entered is determined by settings of flip-flops and other components. Actions specified by phases include such operations as transferring a command pair from memory into the B-register, selecting a specific memory location at which writing or reading will occur, and changing the content of the location counter. Not all mode phases are used in each mode. Phases utilized in the different modes and their functions are given in table 5-1.

Settings of the flip-flops used for designating the modes and phases are established principally from (1) the flip-flops containing the operation code of the command being executed, and (2) signals generated from switch positions established through operator action. Thus, in actuality, program commands and the operator direct system operations even though the control may appear to be derived solely from electronic and electromechanical devices.

Program and Manual Direction

Because the program and operator control is superimposed over that established by the computer, the program commands and operator actions must be correct. During execution of a command, the action specified by the operation code can be altered only by the logic of the command, which may or may not allow the action to follow one of several alternative courses. For example, in a transfer command, the operation code specifies a transfer of control if a certain condition is fulfilled. However, whether the transfer actually occurs or the computer continues to execute commands in sequence depends on success or failure, respectively, in meeting the conditional requirements. As another example, an add operation code always causes execution of the addition command. But the manner in which the result is derived depends on whether either or both quantities involved are negative or positive numbers.

While alternative courses of action during execution of a command are fixed by logic design, deviations in computer operations are possible through programming. In RECOMP III, two program modifiers may be incorporated into certain commands by the programmer which, when interpreted by the computer, cause deviations in program execution. Both modifiers use command bit positions other than those ordinarily used for the operation code. These positions are the sign position of the right command of a pair and the halt word indicator position of a transfer of control command. The sign position of the right command, when minus (0), indicates indexing is to be performed against an address of a right command operand; that is, the address originally specified is changed in direct relation to the number contained in the index register. The half word indicator in a transfer of control command specifies whether program execution sequence will proceed to the left or right command of a pair at the location specified in the command.

After the program of commands and data has been entered into internal storage and program execution has been initiated, system operation is virtually automatic in that actions specified by the commands are carried out in proper manner and sequence without any significant operator assistance. Execution of the individual commands is then under control of the components which establish the modes and phases employed in effecting the operations specified.

Mode Selection and Sequencing

Components establishing the modes and phases consist principally of flip-flops and gates. Their signals, along with those from various switches and contacts, constitute the mode and phase logic that determines and controls the operations of the entire system. The logic for determining a particular mode and phase and for sequencing from one mode and phase to another, is represented by flip-flop states whose settings originate from variety of sources and circumstances. Principal source is signals from secondary gates; primary controlling circumstance is the nature of the operation code. Other circumstances include binary value of the half-word indicator, existence of an indexing requirement, and positions of switches.

The nature of the operation code produces considerable variation in the selection of modes and phases used by the different commands and in the sequencing from one mode and phase to another. For example, the operation code of the halt and transfer command causes operations to proceed from the operand mode through the operand direction, and display modes to the idle mode. The operation code of the divide command, however, causes operations to progress from operand mode through operand, direction, and execute modes to command or operand mode.

Although configuration of the operation code is the main factor governing most mode and phase selection and sequencing, two modes, sync and idle, utilize other determinants. Sync mode is established initially by signals from the power supply, idle mode by signals from the digit counter in conjunction with mode control flip-flops. Entry into the command mode at the start of program execution is also initiated at a non flip-flop source, the signal originating at the OPERATION switch. Subsequent entry into command mode, however, is caused by flip-flop and gate signals.

Selection of the appropriate mode and phase at the beginning of execution of a command is accomplished during operand mode when the flip-flops containing the operation code and other determinants establish the mode and phase flip-flop settings. Sequencing from one mode and phase to another during execution of a command is then accomplished through various mode and phase control flip-flop and gate signals, along with timing signals.

Phase-Duration Timing and Termination

The flip-flop and gate signals governing operation sequencing and duration vary in configuration, but a timing signal is always present in one form or another. Timing originates as a memory disk clock signal which is then applied to all flip-flops. Certain flip-flops constituting the digit (bit) counter, in turn, generate special timing signals for determining duration and sequencing of all computer operations.

Initial timing following power turn-on is provided by power supply relays. Upon completion of logic synchronization, the various configurations of the digit counter flip-flops provide signals to timing flip-flops and gates whose signals are always coincident with a bit time or word time. The timing flip-flop and gate signals are, in turn, then used alone or in conjunction with other flip-flop and gate signals to control mode and phase selection, duration, termination and sequencing.

Circumstances initiating termination include conditions such as operation for the specific number of word times required to execute a command, completion of transfer of information between memory and registers, fulfillment of a condition, completion of the number of shifts specified, and number of characters input or output.

Termination of a phase constitutes entry into a next mode or phase. However, the next mode or phase entered depends solely on the nature of the operation specified by the command being or just executed. Progression may be into a phase of the same mode or a different mode, and may bypass phases within a mode or modes which normally would occur next in sequence. However, regardless of the cause of termination and the subsequent mode or phase entered, the transition always is performed at the predetermined time established through logic.

INFORMATION ENTRY

Both commands and data can be entered into internal storage from various input devices. These devices may include equipment such as a keyboard, tape reader, punched card reader, or other units meeting signal requirements of the computer. Selection of the device to be used for input can be achieved by either the input command or a switch. Entry of the commands and data can be accomplished manually, such as from the keyboard, or automatically by the input device under computer control, as with a tape or punched card reader. Regardless of the entry method, however, the input process is always under control of the input mode logic in respect to coordination and timing.

Equipment Coordination and Timing

Entry of information into the RECOMP III computer is always coordinated with other system actions and performed in specific, timed sequences. Immediately

following entry into the input mode, a signal is sent from the computer selecting and starting the input device to be used. This signal also signifies that the computer is ready to accept information. If the input device is operating and ready to transmit information to the computer, the input device then sends a timing pulse to the computer. This pulse indicates that the input device is transmitting the first character. (When information is being entered manually, operator depression of the appropriate key on the keyboard results in mechanical action that is translated into the coded signals transmitted to the computer. If the information is being entered automatically by a device, the selected unit reads the storage medium and translates the coded information into the signals sent to the computer.)

During transmission of the first character, the input device timing pulse to the computer discontinues to allow the computer to transfer the character from its input components to the A-register. Continued presence of the computer select-and-start signal, however, causes continued operation of the input device and generation of another timing pulse indicating transmission of the next character to the computer.

The input process continues until the quantity of characters specified by the command to be entered has been transmitted to the computer. When the last character has been entered, the character tally in the character counter initiates logic signals which cause termination of the input command. Transition from the input mode is either to the command or operand mode, depending on whether the input command was the right or left command of a pair, respectively.

Information Flow

Although information always is transmitted from the input device to the computer in the form of characters, the method of handling the information in the computer differs somewhat according to the quantity of characters entered. When the quantity of characters is insufficient to form one computer word, the characters are input sequentially until the quantity specified is entered, after which the input command is terminated automatically. If the quantity of characters to be entered forms one or more computer words, the quantity specified is likewise input in sequence and the input command is terminated automatically. However, only the last five characters entered are retained in the A-register, the remainder being lost.

Information flow on input is always from the input device to the D and C (and in some instances E) flip-flops, then to the A-register. Execution of a Store command then transfers the characters in the A-register through the B-register into internal storage. The bits constituting one character are transmitted between the input device and the flip-flops on parallel lines. However, from the flip-flops to the A-register and from this register through the B-register into memory, the bits are transferred serially. All information entered may be placed into main memory or the rapid-access loops as desired.

A count of the number of characters entered is tallied by the character counter portion of the G-register. This count is used in the input termination procedure.

INFORMATION STORAGE

All information entered into memory is placed there via the B-register. Likewise, information retrieved from storage is also always routed through this register. The only memory writing-reading processes not utilizing the B-register are information transfers between the loops and main memory. Following paragraphs describe the writing and reading processes, storage location selection, and coordination and timing of writing and reading with other computer operations.

Writing and Reading Processes

The writing and reading processes employ a principle in which information is recorded on a rotating disk as flux patterns representing binary digits and read from it in the form of voltage pulses.

When recording into main memory (except from a rapid-access loop) information flow proceeds from the B-register to the write amplifier flip-flops, then to a specific write head chosen by a flip-flop selection matrix. In writing into loops and registers, the information flow is the same except no selection by a matrix is required. Information flow from a loop to main memory proceeds from a loop read amplifier direct to the main memory write amplifiers.

Actual recording of information on the disk is accomplished with write currents originated on circuit boards as a positive or negative voltage for a one or zero binary digit, respectively. Because the current direction changes only at transitions between unlike digits, the method of recording is termed the non-return-to-zero method.

Reading of information from memory is accomplished by read heads constantly detecting the information on the disk and sending corresponding signals to read switch boards or, in the loops and registers, to the read amplifiers. The read heads sense only the changes in magnetic state. A change from a zero to one binary digit results in a positive pulse and a change from a one to zero binary digit results in a negative pulse. In a succession of like binary digits, the first bit is signified by a pulse and the remaining digits by a 0-volt level (figure 6-26).

From the main memory, the 64 read head signals are routed to eight read switch boards which each select one signal from among eight as determined by flip-flop states. The eight selected signals, in turn, are applied to eight read amplifier boards from which one signal (from one read head) is chosen for use. Unless the information is being transferred from main memory to a rapid-access loop, this signal from the one read head is then transmitted to the B-register. In main-memory-to-loop transfers, information is sent from a main memory read amplifier directly to a loop write amplifier.

Storage Location Selection

The manner of selecting the specific memory location into which information will be recorded or from which it will be read, is the same for both commands and data. However, the method of selection differs somewhat for main memory and the rapid-access loops in that selection of a specific channel is unnecessary for the latter.

Selection is accomplished in two steps during both recording and reading. The initial step determines whether a main memory or loop location is designated. This action is accomplished by sensing for zero bits in the most significant octal digit of the memory address. If a loop designation is detected, several flip-flops are set accordingly to signify loop; similarly, if a main memory location is sensed, flip-flops are set to signify main memory and to represent the channel in which the location exists. The second step, identical for both main memory and loops, consists of selecting the word sector within the channel or loop. This selection is accomplished by a comparison-for-agreement between the sector address specified and the sector count in the origin-sector channel. When agreement occurs, the computer begins routing the signals from the B-register (or loop) to main memory or from main memory to the B-register (or loop).

Writing-Computation Coordination and Timing

Most store-into-memory operations comprise commands specifically and solely for that purpose. However, a few store operations (such as store address and store index) combine pre-processing of information with writing into memory. These combined operations require more complex coordination and timing than is necessary when only storing to ensure that the various functions are performed at the proper time. Both this coordination and timing are effected with the mode and phase control logic as directed by the operation code of the command.

For example, in the store address command, operations are extended in the operand mode into phase 3 for transferring the information from memory into the B-register. Then, the sequence proceeds to phase 2 of the direction mode for transferring the contents of the A-register into the B-register. When this latter transfer is completed, the execute mode is entered to place the contents of the B-register into main memory. In this command, operation in the execute mode provides the additional time for storing above that needed for an ordinary store operation.

Reading-Computation Coordination and Timing

Many computer operations require reading from storage in conjunction with computation and information processing. Commands involving reading combined with computation or processing include all fixed point arithmetic commands, the extract command, and the floating point arithmetic commands. Coordination and timing involved in these joint operations are provided by mode and phase control logic as directed by the operation code.

The add command provides a typical example of combined reading-computation operations. In this command, operation is extended in the operand mode into phase 3 to transfer data from memory into the B-register, and then is sequenced

into phase 2 of the direction mode to determine, from the signs of the numbers, whether the addition command will utilize the addition or subtraction process. The calculation process also is started during the direction mode and is then completed during the execute mode, with the answer being retained in the A-register. In this command, a longer direction mode plus operation in the execute mode provides the additional time needed for the combined operation over that which would have been required for only reading information from memory into the A-register.

COMPUTATION AND PROCESSING

All general purpose digital computers perform many functions other than solving arithmetic problems; among these are the comparing, integrating, and converting of data of all types. As a group, these operations upon data are often referred to as information processing to distinguish them from true arithmetic calculations, which are frequently called computations. However, the latter designation is commonly applied to all operations ordered by program commands.

Actual problem solution usually involves operations of both a calculation and processing nature. Many arithmetic calculations are performed on data previously processed in some manner, such as the addition of numbers changed in value through reversal of sign. Similarly, processing is frequently performed on the result of a calculation.

Many commands of a non-arithmetic nature are utilized solely in handling information and directing program execution sequence. Such commands, for example, enter and output information, transfer control to a nonsequential memory location, or transfer information between loop and main memory. These operations, however, also are not truly information processing because no transformation of data occurs.

Mathematical Calculations

RECOMP performs all calculations, both fixed and floating point, through the simple processes of addition and subtraction. Multiplication and division are performed principally through repeated successive additions or subtractions, with other operations included as applicable to the requirements of the type of calculation involved.

Addition and subtraction are carried out by an adder-subtractor circuit operating in conjunction with the A-, B-, and R-registers. The adder-subtractor circuit is composed of two flip-flops, one of which is used for registering the sum or difference resulting from each addition or subtraction of bits and the other for noting carry or borrow from each addition or subtraction. The registers hold the numbers involved in the calculation.

The specific arithmetic operation desired is specified by the operation code and controlled by the mode and phase logic. Both the fixed and floating point calculations are accomplished in the same manner as in hand calculations, and are performed serially from least significant to most significant bit. All calculations are accomplished through application of a number from memory against another number in the A-register. The number from memory is transferred into the B-register by the arithmetic command; the number in the

A-register was either placed there by a command executed prior to the arithmetic instruction or retained there from a previous operation.

A general, typical example of the manner in which the computer performs an arithmetic calculation is provided by the add command. This command first causes a number to be retrieved from memory and placed in the B-register. Addition is then started by adding the least significant bit from each the number in the A-register and the number in the B-register. This addition is accomplished by combining signals from one flip-flop of each register in a manner determined by whether the bits being added are ones or zeroes. The result of this addition is then utilized (1) to set the sum-difference flip-flop of the adder-subtractor to the appropriate state, and (2) if any carry resulted, to set the carry-borrow flip-flop of the adder-subtractor to its appropriate state. Output of the sum-difference flip-flop is then transmitted to the B-register, while that of the carry-borrow flip-flop is applied to the subsequent addition of the next most significant bit from each the A-and B-registers. This add-and-carry process continues until all bits in the A-and B-registers have been operated upon. Simultaneous with the addition process, the A-register is cleared to zero to prepare it for receiving and retaining the result. Then when the addition of all bits is completed, the contents of the A-and B-registers are added to place the result in the A-register; this process is equivalent to replacing the zeroes in the A-register with the sum in the B-register. Subtraction is accomplished by essentially the reverse of the addition process.

In floating point arithmetic, all calculations are performed in generally the same manner as in fixed point arithmetic and utilize the same processes. However, prior to and following the fixed point addition-subtraction phase of the operation, a denormalizing or normalizing (scaling) process is performed on the floating point numbers and result.

Non-Mathematical Information Processing

Most operations of the computer system are of a non-mathematical (non-calculation) nature. As previously indicated, these operations are of two general types: (1) information processing, consisting of varied manipulations upon data, and (2) handling and control, comprising such actions as transfer of information and control of program execution sequence. However, some non-mathematical (non-arithmetic) operations constitute or can be used for semi-arithmetic functions. As examples: a compare operation determines which of two quantities is greater, and a shift command can be used in place of multiply or divide when the multiplier or divisor is equivalent in value to a power of two. Such semi-arithmetic operations are sometimes considered mathematical functions to distinguish them from operations utilized for processing only, such as the RECOMP III Load Index command which places the address at the location specified into the index register. This operation is used to expedite transferring blocks of data from memory for classification and re-storing in separate portions of memory.

Regardless of whether the information processing operations are performed for semi-arithmetic or genuine processing purposes, however, the majority is accomplished in the A-register. Usually, if the information to be acted upon is not in the A-register from a previous operation it is placed there by a "transfer data" command of some type which transfers the information from memory (or another register) to the A-register. The information is then acted on according to the operation specified (such as shifted, compared, or altered) and left in the

A-register. If the processed information will not be used in calculation or some other subsequent operation, it is then stored back into memory.

A few non-mathematical operations utilize the R-register or both the A- and R-registers instead of the A-register. These operations include exchange contents of A-and R-registers, compare contents of A and R registers, and some shift commands.

All non-mathematical operations are specified by the operation code and controlled by the mode and phase logic. Sequencing from one non-mathematical operation to another and interuse of non-mathematical operations with calculations are, of course, always governed by the program as established by the programmer.

INFORMATION OUTPUT

Transfer of information out of internal storage to an external device is accomplished in essentially the reverse of the procedure used for information input. Output devices may include a paper tape punch, card punch, or any other equipment compatible with the RECOMP computer. Selection of the device to receive the information is accomplished automatically by the command in the computer program or manually by a switch. Unlike input, however, output of information is only possible under computer control.

Equipment Coordination and Timing

Information output is always coordinated with other computer action and performed in specific, timed sequences. Immediately after entering output mode, the computer sends a select-and-start signal to the designated output device. At essentially the same time that it selects the device, the computer also transfers the first character from the A-register through the output components to the output unit.

After sending the first character to the output unit, the computer then enters into a "waiting action" while the device reproduces the character. While the character is being reproduced, the output device transmits a timing pulse to this effect to the computer. The computer then sends another select-and-start signal to the device to reinitiate the output process for the second character. This process continues until the quantity of characters specified by the command to be output have been reproduced by the device.

When the last character has been reproduced, the character tally in the character counter initiates logic signals which cause termination of the output command. Transition from the output mode is to either the command or operand mode according to whether the output command was, respectively, the right or left command of a pair.

Information Flow

The information itself is always transmitted from the computer to the output device in the form of characters. However, the method of handling the flow of information in the computer differs slightly according to the quantity of characters output. If the quantity of characters to be output is less than that

constituting one computer word, the characters are output sequentially until the specified quantity is transmitted. Output is then terminated and any characters remaining in the A-register are left there. When the quantity of characters specified to be output is greater than that comprising one computer word, the five characters in the A-register are output in sequence until the quantity specified is recorded. However, only the first five characters recorded will be meaningful.

Regardless of whether the quantity of characters output constitutes less or more than one computer word, the information flow is always from the A-register through the D and C (and in some instances E) flip-flops to the output device. When information to be output is not in the A-register from a previous operation, it is placed there from either main memory or the rapid-access loops by a "transfer-to-A" type command. Information transfer from memory to the A-register occurs via the B-register.

Transfer of bits comprising characters occurs serially between memory and the B-register and from this register through the A-register to the flip-flops. From the flip-flops to the output device, however, the bits constituting one character are transmitted on parallel lines.

A count of the number of characters output is tallied by the character counter portion of the G-register; this count is used in the output termination process.

OPERATION EXECUTION SEQUENCE

During execution of a program, operations continue until halted by a stop command in the program or the operation switch on the control console. The individual commands are executed one at a time in sequence, unless a transfer of control command causes computation to continue at a non-sequential location. Similarly, the individual modes and phases involved in execution of a command are performed in a specific sequence. Thus, progression of computer operations through both a computer program and the various modes and phases is essentially both serial and sequential.

Serial — Parallel Progression

Although program, mode, and phase progression takes place serially, some actions within phases occur in parallel. The more common types of actions which occur in parallel in RECOMP III include (1) examining information for a specific condition during a transfer of information and then setting a flip-flop accordingly; (2) recirculating the contents of loops and registers while other computer actions are in process; and (3) simultaneous copying of information from one register to both another register and group of flip-flops, such as from the G-register to the Z-register and the C flip-flops, during the same bit times. Specific examples of simultaneous action in RECOMP III include such joint operations as:

During a portion of the last word time of the Sync mode, the location counter is zeroed, the location counter display indicators on the control console are extinguished, and the digit counter is synchronized.

During part of phase 1 of the operand mode, while the Z-register is copying the B-register, (1) the B-register contents are also being transferred to the D and C flip-flops, (2) the B-register contents are being examined to detect main memory or loop address, and (3) if indexing is required, the contents of the index register are subtracted from the contents of the B-register.

Parallel action progression during computer operations reduces the time required for all actions involved in an individual operation and/or enables more actions to be accomplished by one operation. Also reducing the time required for performing operations of a computer program is the capability of locating and transferring a pair of commands at a time from main memory or the loops to the B-register.

Program Execution Sequence

Simultaneous transfer of both instructions in a command pair from memory to the Z-register greatly reduces the command access time from that which would be required for locating and transferring each instruction individually.

Sequence of command execution in RECOMP normally is as follows, assuming the memory location of the first instruction to be executed has been entered into the location counter from the keyboard or other input device:

The memory location specified by the location counter is selected and the pair of commands at that address is transferred through the B-register into the Z-register. Next, the command of the pair to be executed, which can be the left or right command, is interpreted and, if specified, its operand procured from memory and placed in the appropriate register. This command is then executed; that is, the function specified by the operation code is performed.

If the command executed was the left command of a pair, the right command of a pair is then interpreted, its operand (if any) located and transferred from memory, and the command executed. However, if the command executed was the right command of a pair, a new pair of commands is located and placed into the Z-register.

Succeeding pairs of commands are located, transferred, and executed in sequence coinciding with sequential advancement of the location counter unless a transfer of control is specified in the program. If a transfer is encountered, the next sequential address specified by the location counter is placed by the non-sequential location in the transfer of control command, and sequential execution then resumes from the new location.

The location counter is advanced one-half location count each time a command is executed. This advance occurs in half and full location count sequence until interrupted by a transfer to a new location. When a transfer is detected, the location counter is set to the new address and sequential advancement is then continued from the new location.

Transfer of control occurs whenever the program contains (1) an unconditional transfer command or (2) a conditional transfer command and the conditional requirement is fulfilled. Both types of transfers and the accompanying changes in the location counter content are always performed under control of the mode and phase logic as established by the operation code of the command.

Mode	Action Progression	Phase 1 (D'_oK_c) Functions	Phase 2 (D'_o K
SYNC ($I_4I_2I_1$)	Entered from: Power application sequence. Phase progression: If Phase 3 entered initially, Phase 3 to 2. Sequences to: Phase 2 of Idle mode.		Synchronize digit counter; set location count indicators on control console; establish stat
IDLE ($I_4I'_2I_1$)	Entered from: Phase 2 of Sync mode or Phase 4 of Display mode. Phase progression: Phase 2 to 1. Sequences to: Phase 1 of Command mode.	Delay transition from Idle to Compute mode for word time.	Retain computer logic in non-compute state.
COMMAND ($I'_4I'_2I_1$)	Entered from: Phase 1 of Idle mode; or Phase 1 or 4 of Direction mode; or Phase 1, 2, or 3 of Execute mode; or Phase 2 of Input mode; or Phase 2 of Output mode. Phase progression: Phase 1 to 2 to 3. Sequences to: Phase 1 of Operand mode.	Transfer contents of G-register to Z-register and channel address bits into C flip-flops; analyze location specified by location counter to determine if command pair is in main memory or loop and establish states of flip-flops accordingly.	Compare command pair location sector bits channel and set flip-flop accordingly when a
OPERAND ($I'_4I_2I_1$)	Entered from: Phase 3 of Command mode; Phase 1 or 4 of Direction mode; Phase 1, 2, or 3 of Execute mode; Phase 2 of Input mode; or Phase 2 of Output mode. Phase progression: Phase 1 only; or Phase 1 to 2 only; or Phase 1 to 2 to 3. Sequences to: Phase 2 of Direction mode.	If entered from Command mode: transfer command pair from B-register to Z-register; if indexing was ordered, subtract contents of index register from right command address; detect if left or right command of pair will be executed first and transfer corresponding unindexed channel address and operation code from Z-register into C and D flip-flops, respectively; if left command is first also transfer unindexed sector address of left command into left character counter; analyze address of command to be executed first to determine if operand is in main memory or loop and establish states of flip-flops accordingly. If entered from Direction, Execute, Input, or Output mode: analyze address (in Z-register) of second command of pair to be executed to determine if operand is in main memory or loop and establish states of flip-flops accordingly; transfer corresponding unindexed channel address and operation code from Z-register into C and D flip-flops, respectively (sector address is not entered into character counter. In some commands, perform various additional special functions.	When operation proceeds to this phase: if le register) unindexed channel address and ope and unindexed sector address into left chara re-enter (from Z-register) unindexed or ind and D flip-flops, respectively, and unindexe counter. If operand is required from memory: compa with sector count in origin-sector channel ar
DIRECTION ($I'_4I_2I'_1$)	Entered from: Phase 1, 2, or 3 of Operand mode. Phase progression: Phase 2 only, or Phase 2 to 1 only, or Phase 2 to 4 only, or Phase 2 to 1 to 4, or Phase 2 to 4 to 1, or Phase 2 to 1 to 2 only. Sequences to: Phase 1 of Command mode; or Phase 1 of Operand mode; or Phase 1, 2, or 3 of Execute mode; or Phase 2 of Input mode; or Phase 2 of Output mode; or Phase 4 of Display mode.	When operation proceeds to this phase: increment location counter when not accomplished during phase 2; other functions performed vary according to operation specified by command being executed.	Increment location counter when phase 2 onl to operation specified by command being exe
EXECUTE ($I'_4I'_2I'_1$)	Entered from: Phase 1 or 2 of Direction mode. Phase progression: Phase 1 only, or Phase 2 only, or Phase 2 to 1 only, or Phase 2 to 3 only, or Phase 3 to 1 only, or Phase 2 of Execute mode to Phase 2 of Direction mode to Phase 1 of Execute mode, or Phase 2 of Execute mode to Phase 2 of Direction mode to Phase 2 to Phase 1 of Execute mode. Sequences to: Phase 1 of Command mode, or Phase 1 of Operand mode, or Phase 4 of Display mode.	Functions performed vary according to operation specified by command being executed.	Functions performed vary according to opera
INPUT ($I_4I'_2I'_1$)	Entered from: Phase 2 of Direction mode. Phase progression: Phase 2 to 1. Sequences to: Phase 1 of Command mode, or Phase 1 of Operand mode, or Phase 4 of Display mode.	Terminate Input command.	Enter into memory from 0 to 128 alphanume device designated by command or switch on
OUTPUT ($I_4I_2I'_1$)	Entered from: Phase 2 of Direction mode. Phase progression: Phase 2 to 1. Sequences to: Phase 1 of Command mode, or Phase 1 of Operand mode, or Phase 4 of Display mode.	Terminate Output command.	Output from memory from 0 to 128 alphanum device designated by command or switch on
DISPLAY ($I'_4I'_2I_1$)	Entered from: Phase 4 of Direction mode, or any exit phase of Direction, Execute, Input, or Output mode. Phase progression: Phase 4 only. Sequences to: Phase 2 of Idle mode.		

	Phase 1 ($D'_o K_c$) Functions	Phase 2 ($D'_o K'_c$) Functions	Phase 3 ($D_o K'_c$) Functions
		Synchronize digit counter; set location counter to zero; extinguish location counter display indicators on control console; establish states of various flip-flops.	If entered, none because transition to phase 2 is forced almost ir Sync mode.
	Delay transition from Idle to Compute mode for word time.	Retain computer logic in non-compute state.	
ode; or se 2 of	Transfer contents of G-register to Z-register and channel address bits into C flip-flops; analyze location specified by location counter to determine if command pair is in main memory or loop and establish states of flip-flops accordingly.	Compare command pair location sector bits in Z-register with sector count in origin-sector channel and set flip-flop accordingly when agreement occurs.	Transfer command pair located in Phase 2 from memory (main s during transfer determine if indexing is to be performed on right establish state of flip-flop accordingly.
ı mode; 2 of › 2 to 3.	If entered from Command mode: transfer command pair from B-register to Z-register; if indexing was ordered, subtract contents of index register from right command address; detect if left or right command of pair will be executed first and transfer corresponding unindexed channel address and operation code from Z-register into C and D flip-flops, respectively; if left command is first also transfer unindexed sector address of left command into left character counter; analyze address of command to be executed first to determine if operand is in main memory or loop and establish states of flip-flops accordingly. If entered from Direction, Execute, Input, or Output mode: analyze address (in Z-register) of second command of pair to be executed to determine if operand is in main memory or loop and establish states of flip-flops accordingly; transfer corresponding unindexed channel address and operation code from Z-register into C and D flip-flops, respectively (sector address is not entered into character counter. In some commands, perform various additional special functions.	When operation proceeds to this phase: if left command is to be executed, re-enter (from Z-register) unindexed channel address and operation code into C and D flip-flops, respectively, and unindexed sector address into left character counter; if right command is to be executed, re-enter (from Z-register) unindexed or indexed channel address and operation code into C and D flip-flops, respectively, and unindexed or indexed sector address into right character counter. If operand is required from memory: compare operand location sector bits in Z-register with sector count in origin-sector channel and set flip-flop accordingly when agreement occurs	When operation proceeds to this phase: transfer operand located mory (main storage or loop) to B-register. In some commands, perform various additional special functions.
to 4 only, ›de; or se 2 of	When operation proceeds to this phase: increment location counter when not accomplished during phase 2; other functions performed vary according to operation specified by command being executed.	Increment location counter when phase 2 only used; other functions performed vary according to operation specified by command being executed.	
only, or to ›f Execute mode. ›de, or	Functions performed vary according to operation specified by command being executed.	Functions performed vary according to operation specified by command being executed.	Functions performed vary according to operation specified by com
  ode, or	Terminate Input command.	Enter into memory from 0 to 128 alphanumeric characters as specified in command from input device designated by command or switch on control console.	
  ›de, or	Terminate Output command.	Output from memory from 0 to 128 alphanumeric characters as specified in command to output device designated by command or switch on control console.	
›ction,			

Table 5-1. RECOMP III Operating Mode and Phase Functions

Phase 3 ($D_o K'_c$) Functions	Phase 4 ($D_o K_c$) Functions	Operation Deviations
ansition to phase 2 is forced almost immediately after entry into		Phase entered in Sync mode can be 2 or 3 because initially D_o may be in either state.
	NOTES NONEXISTENCE OF LISTED FUNCTIONS IN MODES INDICATES UNUSED PHASES. OPERATION SEQUENCES IN SINGLE-CYCLE COMPUTE ARE THE SAME AS DURING CONTINUOUS COMPUTATION EXCEPT THAT, IN SINGLE CYCLE, OPERATION ALWAYS SEQUENCES TO THE IDLE MODE AFTER EACH COMMAND EXECUTED INSTEAD OF TO THE COMMAND OR OPERAND MODE. IN SOME OPERATIONS, A SPECIFIC PHASE IS ENTERED FOR ONLY ONE BIT TIME. THE DISPLAY MODE ACTUALLY CONSTITUTES PHASE 4 OF THE COMMAND MODE. HOWEVER, BECAUSE OF THE DISSIMILARITY OF FUNCTIONS ACCOMPLISHED, THE PHASE 4 ACTION SEQUENCE IS DESIGNATED DISPLAY MODE TO MORE ACCURATELY SIGNIFY THE FUNCTIONS PERFORMED. IN THE ACTION PROGRESSION COLUMN, THE WORD "ONLY" SIGNIFIES THAT AN OPERATION USING THAT PARTICULAR MODE AND PHASE PROGRESSION DOES NOT UTILIZE ALL PHASES APPLICABLE TO THAT MODE.	Idle mode is entered from Sync mode following power turn-on; Idle mode is entered from Display mode following a manual or program halt.
ated in Phase 2 from memory (main storage or loop) to B-register; f indexing is to be performed on right command address and ccordingly.		Command mode is entered from Idle mode following power turn-on and after a computation halt; Command mode is entered from Direction, Execute, Input, or Output mode when last preceding command executed is right command of a pair or a left command ordering a transfer of control.
this phase: transfer operand located during Phase 2 from me-) to B-register. n various additional special functions.		Operand mode is entered from Command mode whenever new command pair is transferred from memory to B-register; Operand mode is entered from Direction, Execute, Input or Output mode when last preceding command executed is left command of pair and did not order transfer of control. Phases used determined by configuration of command operation code; Phase 1 only used for commands not requiring operand from memory and not allowing indexing; Phases 1 and 2 only used for commands allowing indexing but not requiring operand from memory; Phases 1, 2, and 3 used for commands allowing indexing and requiring operand from memory. Special functions accomplished by Phase 1 are performed only during execution of some transfer commands; special functions accomplished by Phase 3 are performed only during execution of multiply, divide, and floating point arithmetic commands.
	When operation proceeds to this phase: enter new command pair address into location counter for transfer of control, or enter index into index register.	Phase progression during Direction mode varies as required for performing operations specified by commands. Operation sequencing to Execute, Input, or Output mode dependent on operation required in command being executed; operation sequencing to Display mode caused by execution of Halt and Transfer command or by manual computation halt; operation sequencing to Command or Operand mode dependent on whether last command execution completed in Direction mode is, respectively, (1) right command or left command ordering transfer of control or (2) is left command not ordering transfer of control.
ccording to operation specified by command being executed.		Phase progression during Execute mode varies as required for performing operations specified by commands; phase progression from Execute mode to Direction mode and back to Execute mode occurs only in floating point add and subtract commands. Command mode is sequenced to when command first executed is right command or left command ordering transfer of control; Operand mode is sequenced to when command just executed is left command of pair not ordering transfer of control; Display mode is sequenced to during manual computation halt.
		Command mode is sequenced to when Input command is right command of pair; Operand mode is sequenced to when Input command is left command of pair; Display mode is sequenced to during manual computation halt.
		Command mode is sequenced to when Output command is right command of pair; Operand mode is sequenced to when Output command is left command of pair; Display mode is sequenced to during manual computation halt.
	Establish location counter display indication corresponding to new address entered into location counter by Halt and Transfer command; when computation is halted manually, establish location counter display indication corresponding to address of next command to be executed.	Display mode is entered from Direction mode during execution of Halt and Transfer command or during manual computation halt; Display mode is entered from Execute, Input, or Output mode during manual computation halt only.

Section 6

FUNCTIONAL DESCRIPTION

GENERAL

This section describes functional operation of the electrical and mechanical component elements of the RECOMP III computer system. The functional descriptions of these elements are grouped under four major categories. . These categories are the computer, control console, input-output chassis, and input-output equipment.

COMPUTER

Functional description of the computer elements is separated according to power supply, signal circuits, memory, control circuits, loops and registers,, and arithmetic circuits.

Power Supply

The computer power supply consists of the power control circuits and the secondary power supplies. Section 3 briefly outlines the most important characteristics of the power supply. Figure 7-10 of the Service Manual shows the power supply interconnecting wiring.

Computer power supplies utilized in RECOMP III may be in a basic or modified configuration. The basic power supply is designated by part number 58308-501; the modified power supply by 58308-501-11. Principal difference is that the basic power supply contains manually-controlled power and memory interlocks and an additional initial power relay K2, which are not present in the modified power supply. These additional components permit power to be removed from the power supply power transformer and/or the magnetic memory solenoid during maintenance procedures if desired. However, removal of these components in the 58308-501-11 power supply produces no essential differences in power supply operation between the two units. Therefore, the functional description of the power supply that follows includes no references to the 58308-501 configuration. All power supply illustrations in this manual and in the Service Manual, however, include the basic power supply components and depict their functions.

Power Control Circuits

The power control circuits of the power supply switch power to other power supply circuits of the RECOMP III system. Power control circuits also provide appropriate timing for the magnetic memory starting. A 30-second time delay permits the memory motor to attain a nominal rotation speed of 3450 rpm before the memory disk is positioned. An additional 10-second time delay after disk positioning permits the snychronization of the logic circuits prior to computer operation.

Figure 6-1 illustrates the power distribution and timing functions of the power control circuits. Components applicable to 58308-501 power supplies are enclosed by dashed lines. The computer power supply receives 115-volts ac primary power when the POWER ON switch on the control console is depressed with circuit breakers CB1 and CB2 closed. Power is directed to the POWER ON indicator on the control console and to the power and memory thermostats. The power thermostat is mounted on the -18-volt power supply circuit board, and the memory thermostat is mounted on the memory housing. Each of these thermostats opens at 120°F and closes at 105°F to ensure that temperatures are under these limits for computer operation.

Power passes through the power and memory thermostats to the relay power supply bridge rectifier. The -100-volts dc output of the relay power supply energizes initial power relay K1. Contacts 9 and 10 of relay K1 maintain power to the relay power supply after the POWER ON pushbutton is released. Relay K1 contacts also apply 115-volts ac power (1) to the computer cooling blowers, (2) to the power transformer, (3) to the memory motor, and (4) to 30-second delay relay K3. The power transformer secondary windings supply ac power to the secondary power supplies which, in turn, supply operating power to all other circuits with the exception of the write switches. These switches are inhibited by open contacts 13 and 12 of relay K6. Closed contacts 8 and 10 of relay K6 present a binary 1 condition to generate delay start signal T_d which is applied to the logic circuits. Open contacts 5 and 7 of delay relay K3 inhibit the memory solenoid to delay positioning of the memory disk.

Relay K3 energizes 30 seconds after the initial application of power, and contacts 5 and 7 close to energize the memory solenoid and memory hold relay K5. Energization of the memory solenoid attracts the solenoid plate; movement of the solenoid plate, in turn, causes the memory disk to be positioned near the headplate. Sector track signals from the disk then synchronize the computer circuits in conjunction with signal T_d. Normally closed contacts 1 and 3 of relay K5 open to de-energize 30-second delay relay K3. Contacts 5 and 6 of relay K5 form a holding circuit to keep the memory solenoid energized. The closing of contacts 12 and 13 of relay K5 energizes delay relay K4 through normally closed contacts 1 and 3 of relay K6 after 10 seconds.

Relay K4 is fully energized 40 seconds after initial application of power. Contacts 5 and 7 of relay K4 close to energize relay K6 and illuminate the READY indicator on the control console. Normally closed contacts 1 and 3 of relay K6 open to de-energize delay relay K4. Contacts 5 and 6 of relay K5 form a holding circuit to keep the READY indicator illuminated. The closing of contacts 13 and 12 of relay K6 applies -3-volt power (bias) to the emitters of the write switch transistors. Contacts 8 and 10 of relay K6 open to remove signal T_d, and contacts 9 and 10 close to apply a positive voltage from the control console; this transfer inhibits the T_d gates after the synchronization period, and the computer is ready for operation.

Power is removed from the computer by depressing the POWER OFF pushbutton on the control console. This de-energizes all relays except K1, and computer operation is inhibited. The memory solenoid de-energizes, and the memory disk retracts from the headplate. Discharge current from capacitor C2 in the relay power supply keeps relay K1 energized for 0.5 second after power is removed from the computer. The memory motor continues to operate, and the power transformer continues to supply power to the secondary power supplies for this period.

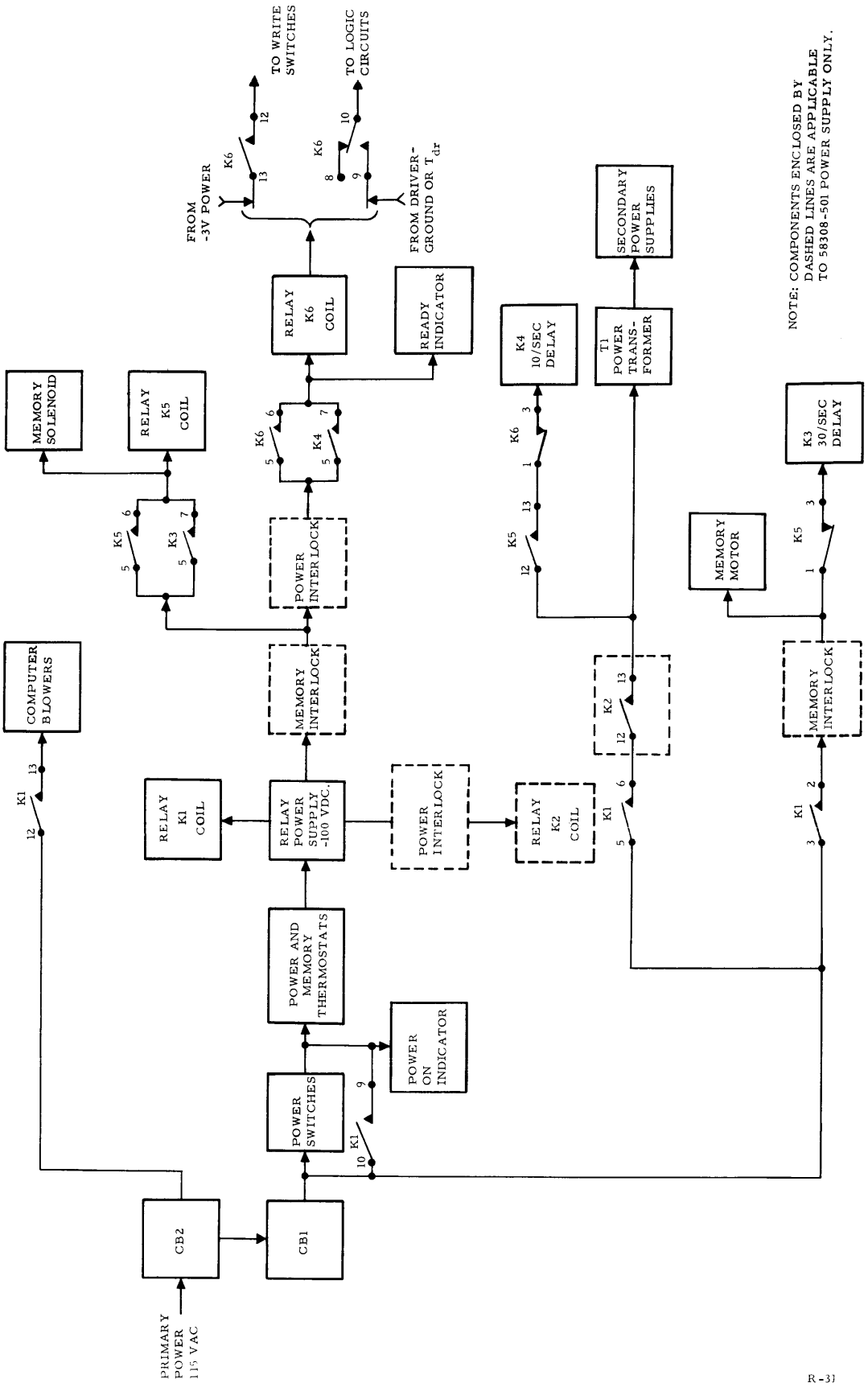


Figure 6-1. Power Control Circuits Block Diagram

CAUTION

COMPUTER SHOULD NEVER BE MOVED WHILE MEMORY DISK IS ROTATING.

ALLOW AT LEAST 15 MINUTES AFTER POWER TURNOFF FOR MEMORY DECELERATION BEFORE MOVING COMPUTER.

Secondary Power Supplies

The computer secondary power supplies provide eight dc voltages for operation of the computer system. Voltages from the power transformer secondary windings are rectified and filtered in the secondary power supplies before application to other circuits. All secondary power supply output voltages are regulated except the -100-volt supply and -3-volt supply (bias). Secondary power supply components are mounted on six circuit boards which plug into J1 through J6 receptacles on the power supply chassis.

-100-Volt Power Supply

The -100-volt power supply provides unregulated power to the neon driver circuits of the control console, and also provides a reference voltage for other secondary power supplies. This power supply (figure 7-14 of Service Manual) consists of full-wave bridge rectifier and filter. The filter is mounted on the filter network circuit board. A 1-ampere fuse on the power supply network board protects the -100-volt supply from overloads.

-18-Volt Power Supply

The -18-volt power supply provides three-percent-regulated power to all computer logic gates and signal circuits. This power supply (figure 6-2) consists of a full-wave bridge rectifier, a filter, and a series regulating circuit. The rectifier circuit is mounted on the power supply network circuit board (figure 7-20 of Service Manual); the filter on the filter network circuit board (figure 7-22 of Service Manual). The regulating circuit consists of a power amplifier stage, a differential amplifier stage, a driver amplifier, a current limiter, and an under-voltage kickout stage.

The differential amplifier stage is the comparison element which monitors the difference between the power amplifier stage output voltage and a reference voltage. This element receives a sample of the output voltage, compares it with the reference voltage, and produces a signal that is proportional to the difference. The driver amplifier amplifies the difference voltage and applies the resultant signal to the power amplifier where the output voltage is regulated as required.

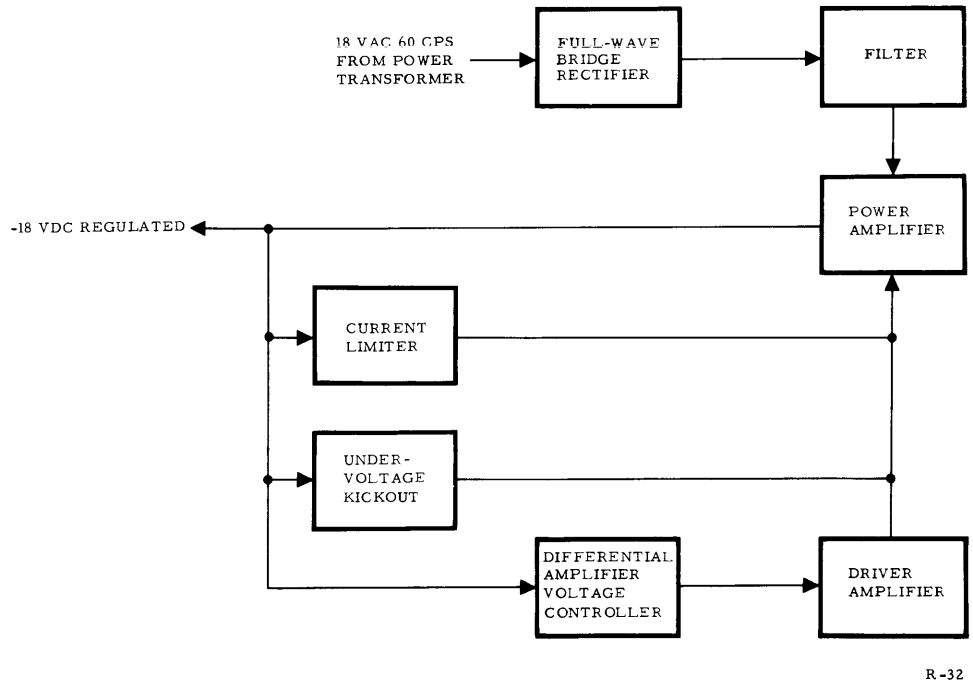


Figure 6-2. -18-Volt Power Supply Block Diagram

Current limiter and under-voltage kickout circuits provide protection to the computer circuits in the event of an abnormal increase in load current. The current limiter conducts and actuates the under-voltage kickout stage when the load current increases abnormally. The under-voltage kickout stage then disables the power amplifier stage, and power is removed from the computer circuits until the load current decreases to a normal value.

Figure 7-14 in the Service Manual is the schematic of the -18-volt power supply. This schematic also shows the circuits for the -100- and -3-volt power supplies. P-n-p power transistors Q2, Q3, Q5, Q6, Q9, and Q10 comprise the power amplifier stage. The transistors are connected in parallel in an emitter-follower configuration which is in series with the unregulated -18-volt output of the bridge rectifier and the load. Parallel configuration of the transistors provides the necessary control current gain, and series configuration of the control element ensures proper regulation for the varying load presented by the computer circuits.

Capacitors C1 and C2 filter the -18-volt power which supplies the collector voltage of the power transistors. Amplifier Q3 is the controlling transistor of the power amplifier stage. Q3 emitter is connected to the base circuits of amplifiers Q2, Q6 and Q9, and the emitters of the latter two amplifiers are connected to the base circuits of amplifiers Q5 and Q10, respectively. The emitter potential of Q3 is then the controlling factor in the conductivity of the other five power amplifier stage transistors.

Driver amplifier Q7 in turn controls Q3 in the same manner that the latter controls the power amplifier stage. Q7 is a d-c amplifier with the emitter connected to the base circuit of Q3. Q7 amplifies the difference voltage from the comparison element to a level sufficient to drive the control transistor.

Differential amplifier transistors Q4 and Q8 form the comparison element of the regulating circuit. The flow of Q8 collector current through resistors R1 and R4 establishes the base potential of Q7 by varying the unregulated voltage from the -100-volt supply. The comparison element then indirectly controls the conductivity of each transistor in the regulating circuit with the exception of the current limiter and the under-voltage kickout transistor.

Conduction of the differential amplifier is determined by the relative base potentials of Q4 and Q8. The two transistors have a common bias resistor in the emitter circuits, and, with equal negative base potentials, their conduction is equal. Emitter current through bias resistor R8 will be equally divided between Q4 and Q8. Any change in the conduction of either transistor results in an equal and opposite change in the conduction of the companion transistor. The value of emitter current through R8 will then remain constant regardless of any variation in the base potentials of Q4 and Q8. If the base potential of one transistor is held at a constant negative value, a varying negative voltage applied to the base circuit of the companion transistor will become the controlling factor in the conduction of both Q4 and Q8.

Zener diodes CR2 and CR3 form the reference element of the regulating circuit. The reference element diodes are connected between the output of the power amplifier stage and ground, and supply a constant -12.4-volts potential to the base of Q4. Each diode provides a 6.2-volt drop under breakdown and is connected in a configuration that ensures a constant reference voltage regardless of load current variations and environmental temperatures. In this configuration, each diode is actually composed of a breakdown diode and a forward-biased diode connected in series. As the load current and environmental temperature vary, the opposite temperature coefficients of the two diodes tend to cancel, and the inherent resistance remains practically constant. The series connection of CR2 and CR3 provides further temperature compensation by reducing the total resistance which would be present if only one diode were utilized. This configuration of the reference element ensures a constant reference voltage over a wide temperature range.

Potentiometer R15 and resistor R16 form the sampling element of the regulating circuit. A small portion of the power amplifier stage output voltage is applied to the base circuit of Q8 through the adjustable contact of R15. The differential amplifier is set to the equilibrium state by adjusting potentiometer R15 for a value of sample voltage proportionate to the reference voltage at the base of Q4 which provides the required -18-volt output. Current through R8 is then a combination of the emitter currents of both Q4 and Q8. Any difference between the output voltage and the reference voltage then appears at the base circuit of Q8. The differential amplifier then amplifies the difference voltage and applies it to the driver amplifier as described. Capacity multiplier C6 is connected between the output voltage and the base of Q8 to

improve the gain characteristics of the amplifier. C7 is connected across the output terminals to improve ripple suppression and reduce the effects of transient signals from the load.

Assuming that the output voltage increases above the normal -18-volt value, C6 couples the negative increase to the base of Q8. The conduction of Q8 increases, but as the emitter potential starts to decrease negatively, the conduction rate of Q4 decreases and the total emitter current remains the same. The increase in Q8 collector current drives the base potential of Q7 in a positive direction, and the increased conduction of the driver amplifier reduces the base drive to control transistor Q3. As the conduction of Q3 decreases, the emitter potential of the other power transistors becomes more positive, their conduction decreases, and the output voltage decreases accordingly. When the output voltage decreases to the normal -18-volt value, no difference voltage is present at the base of Q8, and the differential amplifier resumes the equilibrium state. A decrease in the output voltage results in reverse operating conditions for the circuits just described. Periodic adjustment of potentiometer R15 corrects variations in the output voltage due to changes in circuit parameters.

The current limiter and under-voltage kickout stages are normally inoperative until the load current increases abnormally. Current limiter Q11 is held at cutoff by a bias developed across diode CR4, a "stabistor" which maintains a relatively constant 0.65 voltage drop for various current amplitudes. CR4 is forward-biased by the negative output voltage and holds the Q11 emitter at a more negative potential than the base for values of load current below those indicated. When the load current increases to the abnormal value, the voltage drop acress R18 is sufficient to overcome the bias the Q11 conducts. Conduction of Q11 reduces the base drive to control transistor Q3, and the output voltage decreases.

Under-voltage kickout transistor Q1 is normally held at cutoff by a bias voltage developed across Zener diode CR1 in the base circuit. CR1 holds the base potential of Q1 at -15 volts, and the transistor is cut off with the emitter at the normal -18-volt potential of the output voltage. When the output voltage decreases to -15 volts due to the action of the current limiter, Q1 conducts and further reduces the base drive to Q3. As the output voltage continues to decrease, the action becomes regenerative and Q1 conducts harder until all base drive to Q3 is cut off. Output voltage falls to zero, and the computer circuits become inoperative. Operation of the computer is inhibited until the source of the overload is removed. Power to the computer must be removed and then restored before the regulating circuit will return to normal operation.

Capacitor C4, in conjunction with R2, provides a long-time constant charge path which prevents the activation of the under-voltage kickout stage when power is initially applied to the regulating circuit from the -100-volt source. Capacitor C3, in conjunction with R1, delays the application of a large negative voltage to the base of Q7 until the regulating circuit reaches equilibrium after power turn-on. Without this power delay circuit, Q7 would remain cut off and Q3 would go into saturation from the high negative base voltage supplied from the -100-volt source across R3. The current limiter would then conduct as a result of this condition and activate the under-voltage kickout stage.

Marginal testing of the -18-volt power supply is possible through the application of a variable voltage from the system tester. The test voltage is applied across R17 to the base circuit of Q4. Any change in the conduction of Q4 due to a change in bias voltage produces an equal and opposite change in the conduction of Q3.

-12-Volt Power Supply

The -12-volt power supply provides three-percent-regulated power to the logic flip-flops, clock, read and write amplifier, and read switching circuits. This power supply (figure 6-3) consists of a full-wave bridge rectifier, filter, and a series regulating circuit. The rectifier circuit is mounted on the relay control circuit board, the filter circuit on the power supply network circuit board, and the regulating circuit on the -12-, +75-volt secondary power supply.

Configuration of the regulating circuit is the same as that of the -18-volt power supply except for variations in the number and value of some circuit components. Operation of the -12-volt regulating circuit is similar to that of the -18-volt regulating circuit.

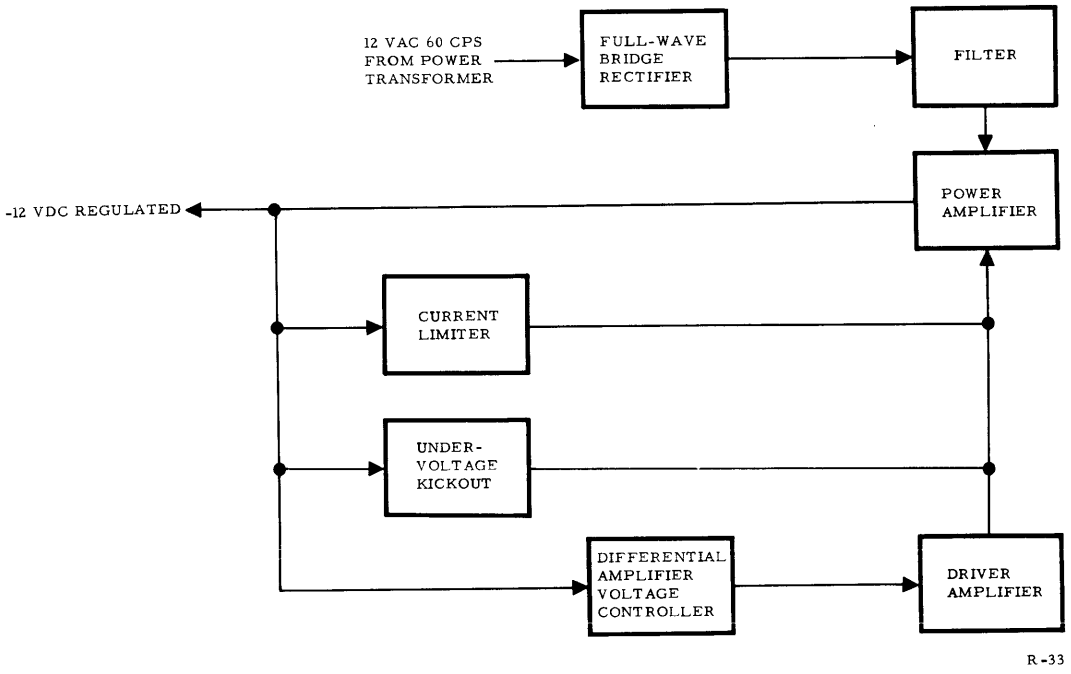


Figure 6-3. -12-Volt Power Supply Block Diagram.

-6-Volt Power Supply

The -6-volt power supply provides three-percent-regulated power to the logic flip-flop, clock, and read amplifier circuits. This power supply (figure 6-4) consists of a full-wave bridge rectifier, filter, and a series regulating circuit. The rectifier circuit is mounted on the relay control circuit board, the filter on the power supply network circuit board, and the regula-

ting circuit on the +6-, -6-, +0.75-volt secondary power supply.

Configuration of the regulating circuit is the same as that of the -18-volt power supply except for variations in the number and value of some circuit components. Operation of the -6-volt regulating circuit is similar to that of the -18-volt regulating circuit.

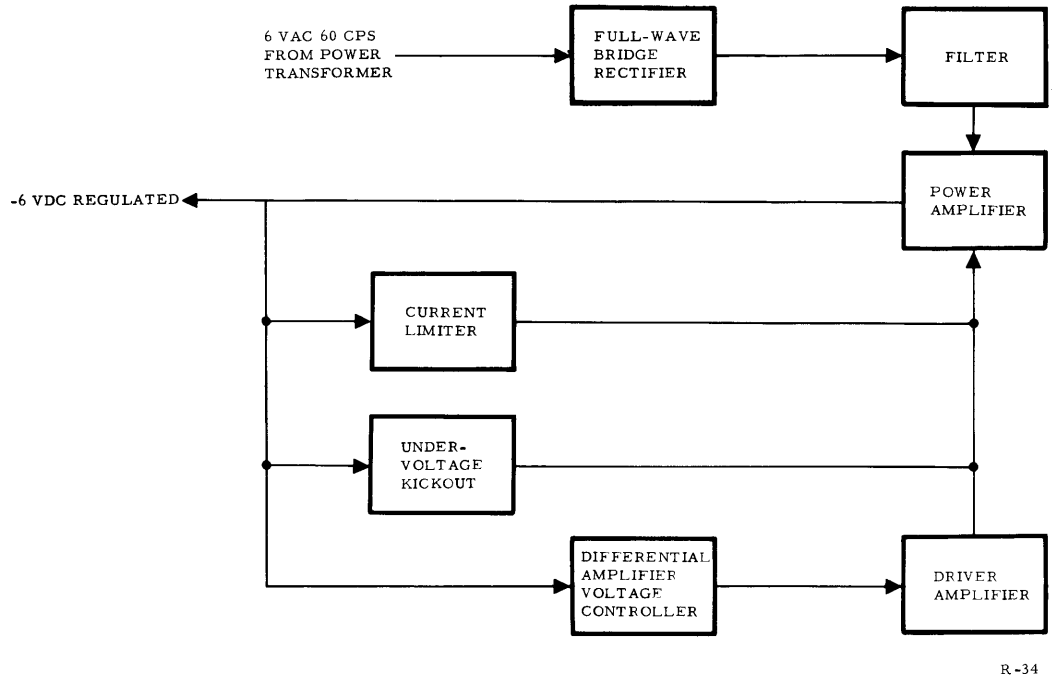


Figure 6-4. -6-Volt Power Supply Block Diagram

+6-Volt Power Supply

The +6-volt power supply provides three-percent-regulated power to the logic flip-flop, read and write amplifier, read switching, and clock circuits. This power supply (figure 6-5) consists of a full-wave bridge rectifier, filter, and a series regulating circuit. The filter is mounted on the power supply filter network circuit board, and the remaining circuits are on the +6-, -6-, +0.75-volt secondary power supply.

Configuration of the regulating circuit is the same as that of the -18-volt power supply except that the under-voltage kickout stage is eliminated. There are also variations in the number and value of some circuit components. Operation of the +6-volt regulating circuit is similar to that of the -18-volt regulating circuit in compensating for load variations.

+75-Volt Power Supply

The +75-volt power supply provides three-percent-regulated power to the clock circuits. This power supply (figure 6-6) consists of a full-wave bridge rectifier, filter, and a series regulating circuit. The filter is mounted on the power supply filter network circuit board, and the remaining circuits

are on the -12-, +75-volt secondary power supply.

Configuration of the regulating circuit is in the same as that of the -18-

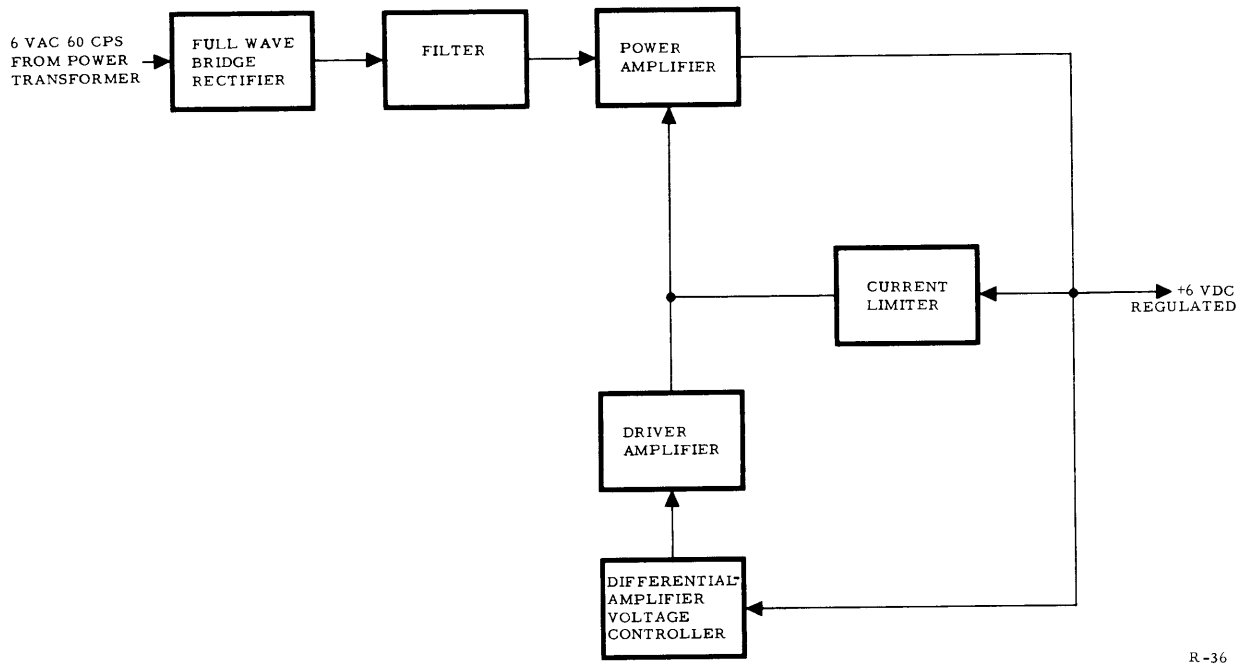


Figure 6-5. +6-Volt Power Supply Block Diagram.

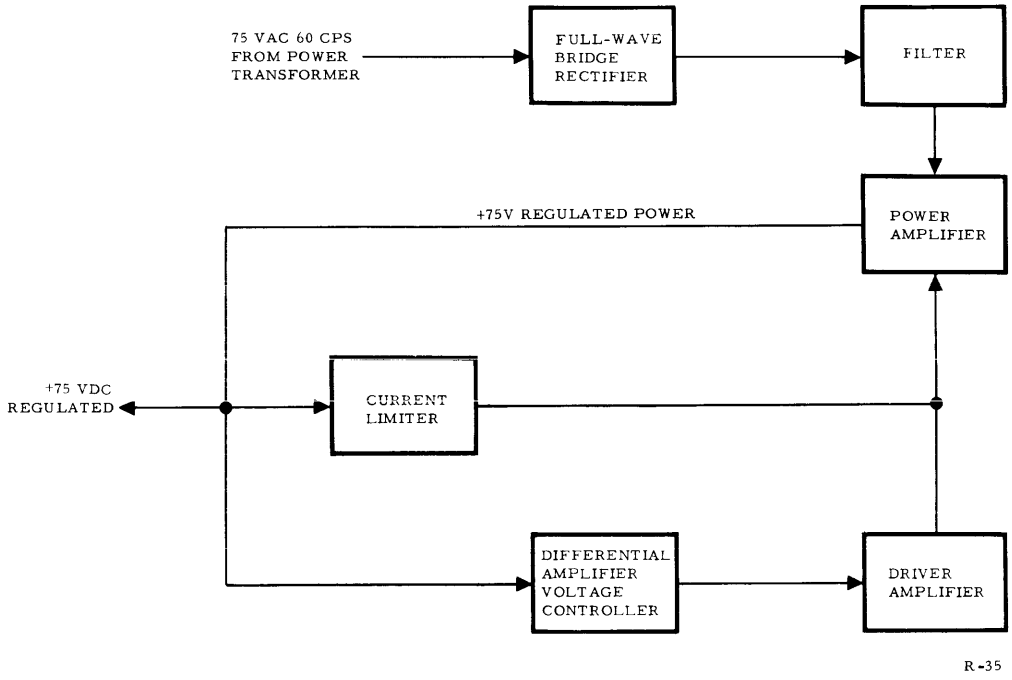


Figure 6-6. +75-Volt Power Supply Block Diagram.

volt power supply except that the under-voltage kickout stage is eliminated. There are also variations in the number and value of some circuit components. Operation of the +75-volt regulating circuit is similar to that of the -18-volt regulating circuit in compensating for load variations.

-3-Volt Power Supply

The -3-volt power source, mounted on the -3-, -18-, -100-volt secondary power supply, provides bias potential to the write switching circuits and to transistors that function as switches for the indicator lamps on the control console. This power supply (figure 7-14 of Service Manual) consists of three identical diodes series-connected between power ground and the emitters of the 64 information channel write switching transistors. Relay K6 controls the initial application of -3-volt power to the write switches (refer to Power Control Circuits in this section). Current through the diodes occurs only when one of the write switches conducts through the write head circuits. Each of the diodes has a nominal 1-volt drop under load.

+0.75-Volt Power Supply

The +0.75-volt power supply, mounted on the +6-, -6-, +0.75-volt secondary power supply, provides five-percent-regulated power to the read switching driver circuits. This power supply (figure 6-7) consists of a power amplifier and a voltage control amplifier. The power amplifier carries the majority of load current, and the voltage control amplifier controls the conduction of the power amplifier. Any change in the output voltage effects a change in the voltage amplifier base bias. The resultant increase or decrease in voltage amplifier collector current varies the power amplifier base bias to produce a change in the output voltage to compensate for load variations.

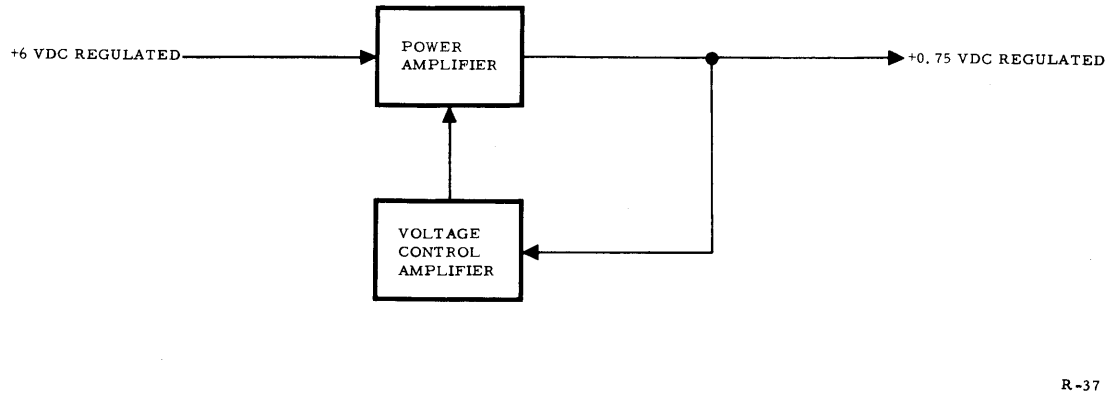


Figure 6-7. +0.75-Volt Power Supply Block Diagram.

Signal Circuits

The computer signal circuits provide basic timing and gating, perform arithmetic and logic computations, store information, and supply necessary

voltage and current amplification. Some of these circuits also control the transfer of information between the input-output equipment and the computer.

Block diagrams and simplified schematics accompany the functional descriptions of signal circuits in this section. Complete schematics of the signal circuits are contained in the Service Manual.

Clock Circuits

The clock circuits shape and amplify the clock signal which synchronizes all computer operations (refer to Timing Control in this section). These circuits consist of Clock Power Amplifiers No. 1 and No. 2. Figure 6-8 illustrates the major functions of the clock circuits; figure 6-9 shows typical clock circuit waveforms and voltage levels.

Clock Power Amplifier No. 1

The clock power amplifier No. 1 circuits provide two outputs. One output is a square-wave clock signal which goes to clock power amplifier No. 2 for further amplification. The second output is an inverted clock signal, the strobe, which goes to the computer read amplifiers to gate memory reading (figure 6-8).

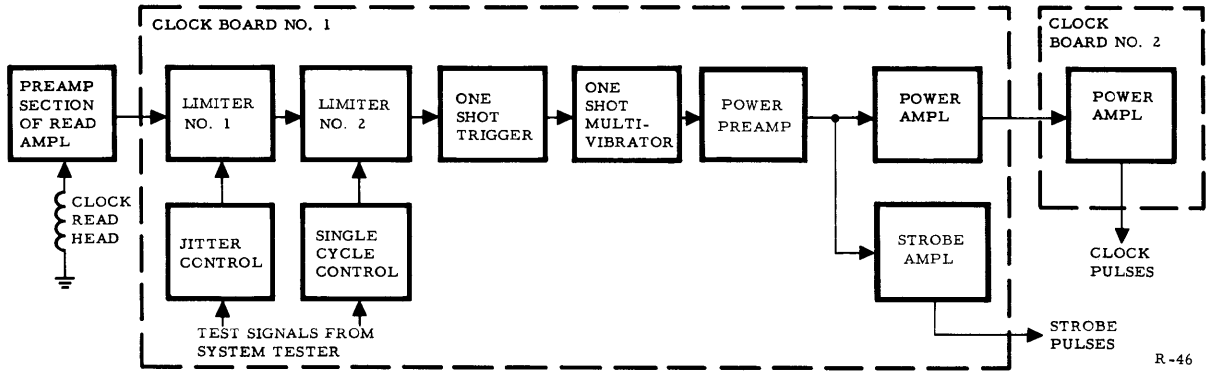


Figure 6-8. Clock Circuit Block Diagram.

Input to clock power amplifier No. 1 is a 4.5-volt peak-to-peak sine wave from the preamplifier section of the clock read amplifier (figure 6-9). This signal is applied to the base of first limiter stage transistor Q1(refer to schema in Service Manual). On the positive portion of the input cycle, Q1 is cut off. The emitter-base circuit of Q2 is then forward-biased, and Q2 conducts. Q2 collector voltage decreases to -2 volts. On the negative portion of the input cycle, Q1 conducts heavily. Q1 collector current clamps the Q1 and Q2 emitter circuits to -12 volts, and Q2 cuts off. Q2 collector voltage increases to -11 volts as C2 discharges through R3. Output of the first limiter stage is approximately a square wave, alternating between -2 and -11 volts.

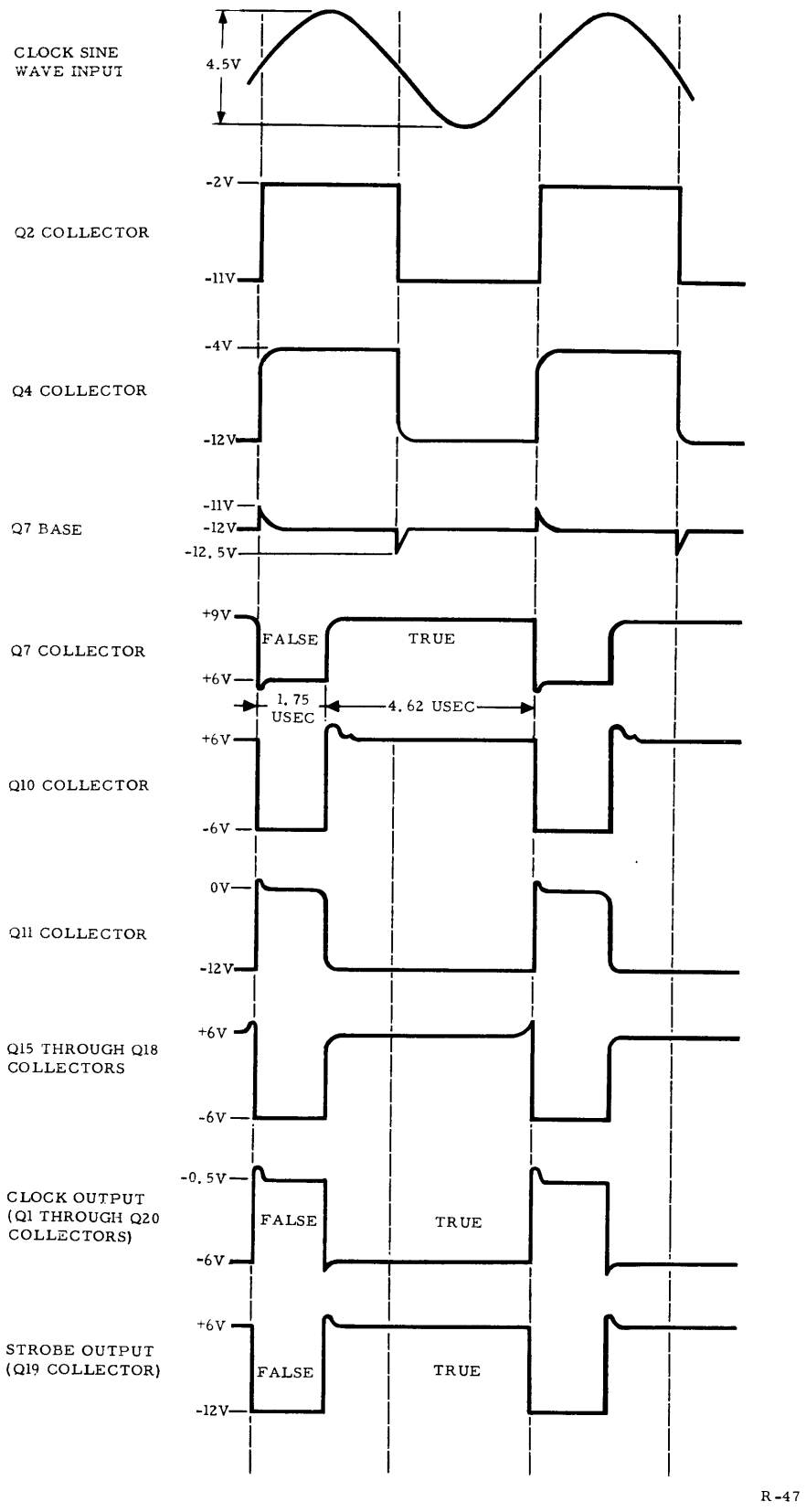


Figure 6-9. Clock Circuit Waveforms

The second limiter stage consists of Q3 and Q4, and circuit operation is identical to that of the first limiter stage. Q4 collector voltage alternates between -4 and -12 volts.

A multivibrator trigger circuit shapes the square-wave output of the second limiter stage into a negative trigger. Q7 is normally biased to cutoff by diode CR3. C5 and R13 form a short-time constant circuit which differentiates the square-wave output of the second limiter stage into positive and negative voltage pulses. The positive transition of the differentiated square-wave drives Q7 into conduction. Q7 collector voltage decreases from 9 to 6 volts, and this negative transition forms the trigger for the following stage.

A one-shot multivibrator establishes the time duration of the clock pulse. Application of the negative trigger pulse from Q7 drives Q8 into conduction, and its collector voltage goes positive from -18 to +6 volts. C6 and C8 couple this positive pulse to the base of Q10 (diode CR8 is reverse-biased at this time). Q10 cuts off, and its collector voltage goes from +6 to -6 volts. Diode CR8 disconnects the base of Q10 from the timing circuit when the positive pulse disappears. Q10 collector voltage is obtained from a voltage divider network composed of R19 and R23. The +6 volts at TP5 is one leg of the divider; the -18-volt supply is the other. C6, R17, and R20 determine the time that Q10 is cut off, and thus the pulse duration of the negative portion of the square-wave output. Potentiometer R17 adjusts the multivibrator output for the required 1.75-microsecond pulse duration of the negative portion.

The one-shot multivibrator circuit incorporates several provisions to improve multivibrator switching and stability. Diodes CR4, CR6, CR7, and CR8 assure non-saturable operation by maintaining collector voltages at a greater value than the base potentials. Q9 speeds the return of the multivibrator to its stable state by supplying an additional positive voltage on the emitter of Q10 as it begins to conduct due to the discharge of C8. Returning the base of Q10 to the -18-volt supply through R17 ensures greater multivibrator frequency stability.

C9 and C10 couple the square-wave output of the one-shot multivibrator to the bases of Q11 and Q12, respectively. Q11 is a power preamplifier, and Q12 functions as a capacity discharger. Q11 is in a quiescent state with +6 volts on the base across R30 prior to the triggering of the multivibrator. When Q10 cuts off to establish the time duration of the negative portion of the clock pulse, the negative increase in collector voltage triggers Q11 into conduction. Q11 collector voltage decreases from -12 volts to approximately zero, and this positive transition back-biases CR12 at the Q12 emitter to keep Q12 cut off even with the negative pulse from Q10 present at the base. Current flow through R24, CR10, and R25 keeps Q11 in conduction for the same interval that Q10 is cut off.

When the multivibrator reverts back to its stable state and Q10 again conducts, the positive transition present at the Q10 collector cuts Q11 off. As Q10 collector voltage goes negative, CR12 conducts and applies the negative transition to the Q12 emitter. This negative voltage, in coincidence with the negative voltage coupled to the Q12 base by C10, triggers Q12 into conduction. Q12 conduction rapidly discharges the load capacity that shunts R26. The load is composed of power amplifiers Q15 through Q18. Thus, Q11 amplifies and inverts the one-shot multivibrator output signal, and Q12 improves the fall time of the waveform trailing edge.

The positive pulse generated by Q11 is applied to four clock power amplifiers and a strobe amplifier connected in parallel. These amplifiers are normally cut off, and conduct upon application of the signal from Q11. The clock power amplifiers' paralleled output is a +6- to -6-volt pulse which furnishes the input to clock power amplifier No. 2. The strobe amplifier output is a +6- to -12-volt pulse which furnishes the input to the computer read amplifiers.

Q13, Q14, and Q20 function as capacity dischargers in a manner similar to that already described for Q12. These transistors conduct when the power amplifiers and the strobe amplifier cut off, and discharge the load shunt capacitance to improve the trailing edge characteristics of the clock pulse and the strobe pulse.

Clock amplifier No. 1 circuit board contains two special circuits used in marginal testing of the computer. A jitter circuit produces jitter of the clock pulse for computer functional tests; single-cycle clock circuit produces single clock pulses for computer static tests. The system tester contains the operating controls for these circuits (refer to Sections 2 and 4 of the Service Manual for test procedures and operating instructions). The circuit descriptions that follow are of those components located on clock amplifier No. 1 circuit board which are used specifically in conjunction with the system tester control circuits.

Jitter control amplifier Q5 is normally conducting with the base held at ground potential in respect to an emitter potential of -6 volts. Negative pulses with an amplitude of 0 to -12 volts are applied from flip-flop C_1 through CR1, C3, and R4 to the base of Q2 in the first limiter stage. With no control voltage applied to the base of Q5, these C_1 pulses have no effect upon Q5 conduction. When a control potential of -25 volts is applied to the Q5 base from the system tester, Q5 cuts off and its collector potential follows the amplitude variations of the C_1 input pulses. C3 couples the -12-volt pulse to the base of Q2, and this input alternately triggers Q2 in conjunction with the clock pulses. These simulated clock pulses developed in the first limiter stage establish bunched pairs of pulses in the clock power amplifiers. The spacing (jitter) between the bunched pairs is controlled by potentiometer R4.

Q6 is the single-cycle clock trigger, and is normally held at cutoff by a -25-volt potential at the base applied from the system tester. With Q6 in a quiescent state, the collector potential is -12 volts. A negative trigger is applied from the system tester to produce the single-cycle clock pulse. Q6 conducts when the trigger is applied, and the collector potential rises to approximately zero volts. C5 and R13 differentiate this positive pulse and apply it to the base of the multivibrator trigger Q7. As a result, the one-shot multivibrator generates a single-clock pulse which is transmitted to the computer logic gates to test logic gates one at a time.

Clock Power Amplfier No. 2

Clock power amplifier No. 2 provides the final current amplification necessary to drive the computer logic gates. The circuit consists of 20 power amplifiers, Q1 through Q20, connected in parallel (refer to schematic in Service Manual). The amplifiers are normally cut off with +6 volts on the bases and the emitters at ground potential. Diodes CR12 through CR21 clamp the collectors to -6 volts.

On application of the +6- to -6-volt pulse from clock power amplifier No. 1, the amplifiers conduct, and collector output potential goes from -6 volts to approximately -0.5 volt. Pulse durations of the negative and positive portions of the square-wave output are established by the one-shot multivibrator described under clock power amplifier No. 1. The negative portion is 1.75 microseconds, and the positive portion is 4.62 microseconds. Thus, the clock pulse and the strobe pulse are 180 degrees out of phase with each other. The positive transition of the clock pulse establishes the timing for all computer logic circuits.

Resistors R1 through R17, R19, R20, R23, and R25 balance the currents in the parallel-amplifier configuration. Diodes CR5 through CR10 are anti-saturation devices. Q21, Q22, and Q23 constitute capacity discharge circuits similar to that described under clock power amplifier No. 1. Diodes CR11, CR22, and CR23 ensure that the capacity dischargers cut off when bases and emitters are at the same -12-volt potential.

Logic Circuits

A standard RECOMP III computer chassis contains 38 logic network circuit boards which are identical in configuration and function. Each logic network circuit board contains 14 available gates, the functioning of which is described in this section. Circuit functions of the 11 "and" gates located on each of the six logic driver circuit boards are identical to that for "and" gates described in this section.

The RECOMP III computer chassis is also wired to accept other types of logic circuit boards for use with special input-output equipment or to provide floating point arithmetic capability. If a 12-channel card reader-punch input-output unit is used with RECOMP III, or if other optional devices using from 9- to 12-channel input and/or output are operated with the computer, a special extended register logic circuit board is plugged into receptacle J317 of the computer chassis along with an additional standard flip-flop circuit board at receptacle J327. When the RECOMP III computer is equipped with floating point arithmetic capability, eight additional special floating point logic circuit boards are plugged into receptacles J318 through J325 of the computer chassis, along with an additional standard flip-flop circuit board at receptacle J328. The special extended register and floating point logic circuit boards contain the same type of logic gates as those of the 38 standard logic network circuit boards; however, the physical arrangement of the gates and components varies on the extended register and each of the floating point logic boards from that on the standard logic network boards. A schematic of the extended register logic board is shown on figure 7-52 of the RECOMP III Service Manual; a typical schematic and parts location diagrams of the eight floating point logic circuit boards are shown on figures 7-45 through 7-51 of the RECOMP III Service Manual.

Functional description of the logic circuits contained in the following paragraphs applies to those gates which are typical to all logic circuit boards in the RECOMP II and III computers.

A number of logic gates are located on each logic circuit board. The gates are groups of diodes connected in various circuit configurations to control transmission, storage, selection, and computation of data within the computer. Several inputs may be utilized in a gate, but there is always only

one output. Signals applied to the diodes at each input control the operation of the gate. A gate is said to be enabled when a given number of inputs produces an output, and is said to be inhibited when a given number of inputs results in no output. In RECOMP computer logic circuits, "and" gates are enabled when all input signals are at approximately -12 volts, and "or" gates are enabled when one or more input signals are at approximately -12 volts.

Internal computer operation uses the binary system, and computer data is in the form of binary-1 and binary-0 digits (refer to Internal Information Formats in section 2). Outputs of logic flip-flops are used to represent binary quantities which are employed in the computer to enable or inhibit logic gates and control other circuit functions. Generation of the signals which represent computer binary information is described in this section in analysis of the logic flip-flop circuits.

In logic notation, such as that used in section 7 of this manual and in section 6 of the Service Manual, the diodes of a gate are represented by signal terms rather than by the component identifications, and all terms representing both the input and output diodes of a gate are commonly referred to as a logic equation.

"And" Gate

The "and" gate has the characteristic of producing a true output signal only when all the input signals to the gate are true. Thus, the "and" gate output signal is the Boolean product of the input signals, and the "and" gate symbol is identified by the multiplication sign. View a of figure 6-10 illustrates a typical three-input "and" gate. CR1, CR2, and CR3 are the "and" diodes and CR4 is the disconnect diode (sometimes referred to as the "or" diode). The input signals are applied to the "and" diodes, and the output signal is taken from the disconnect diode.

The diode cathodes are connected together through a common load resistor to the -18-volt supply. Conduction of the "and" diodes controls the bias and conduction of the disconnect diode. The voltage level of the input signals, in turn, controls the bias of the "and" diodes. When all input signals are true (-12 volts), bias of the "and" diodes is equalized, and conduction of these diodes is effectively cut off. The disconnect diode then conducts heavily to produce a true output signal, the current path of which is from the -18-volt supply through the load resistor. When one of the input signals is false (-0.5 volt) and the remaining signals are true, the "and" diode with the false input has the greatest potential difference between anode and cathode. The false input diode conducts heavily and becomes the controlling factor in the circuit. Cathode potential rises to the -0.5-volt amplitude of the false input signal and back-biases the true input diodes to cutoff. The disconnect diode conduction is negligible and results in a false output signal. When all input signals are false, "and" diode conduction is equal which inhibits the gate by back-biasing the disconnect diode to -0.5 volt.

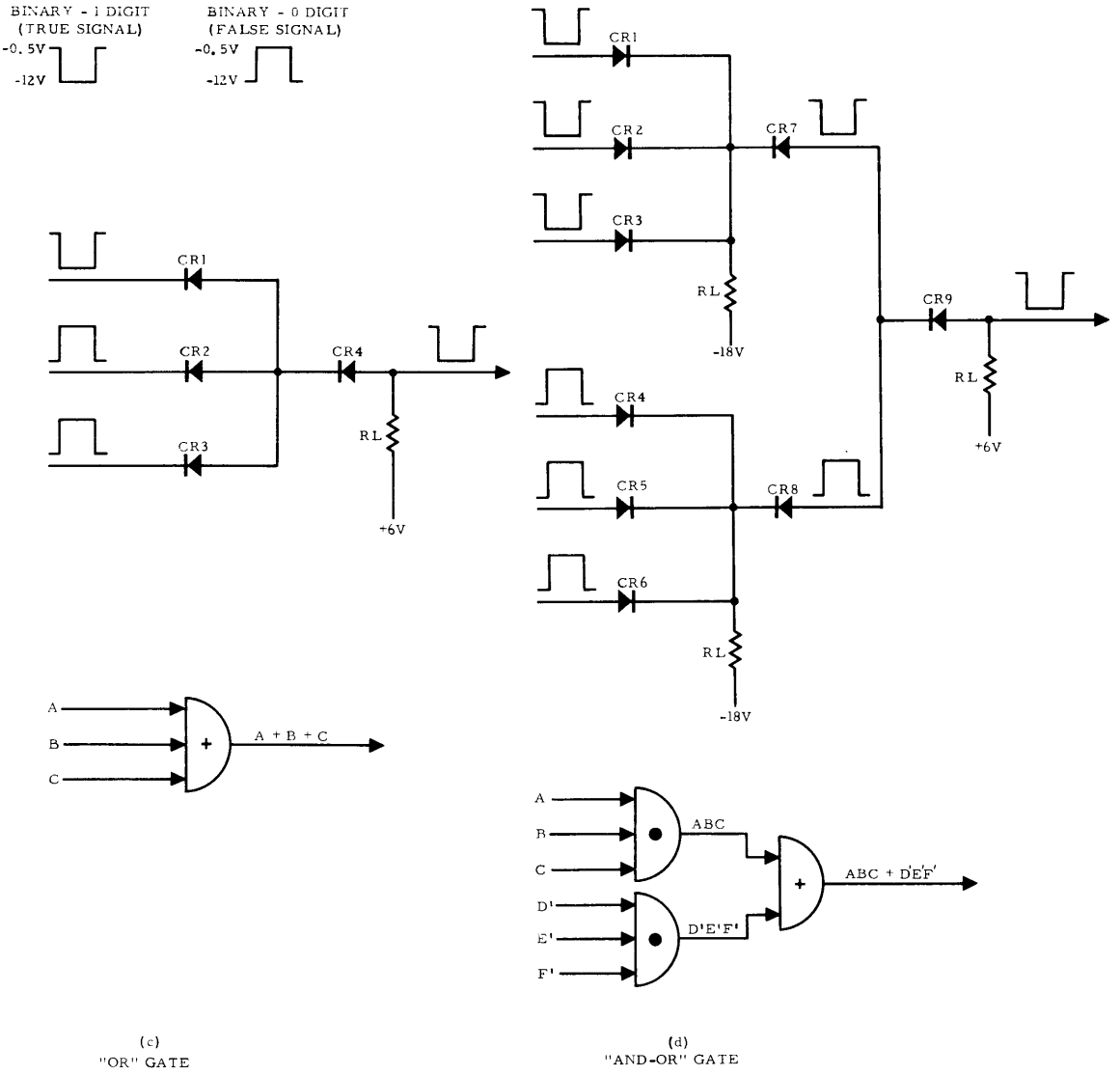


Figure 6-10. Typical Logic Gates and Symbols (Sheet 1)

"And-And" Gate

The "and-and" gate is a combination of "and" gates connected in cascade to combine logic gates each of which have at least one identical gate input. A typical "and-and" gate (view b of figure 6-10) consists of one or more primary "and" gates and one or more secondary "and" gates. The output of a primary gate constitutes one of several inputs to each of the secondary gates. A true or false output from the primary gate combines with other secondary gate inputs to enable or inhibit the secondary gates as previously described. To drive approximately 60 or more gates of RECOMP III computers, a logic driver circuit is utilized to amplify the flip-flop output drive current before application to "and" gates (refer to Logic Driver in this section).

"And-and" gate output signals are also the Boolean produce of the input signals as shown in figure 6-10, view b, and the circuit symbol for each primary and secondary gate is the same as that for the "and" gate.

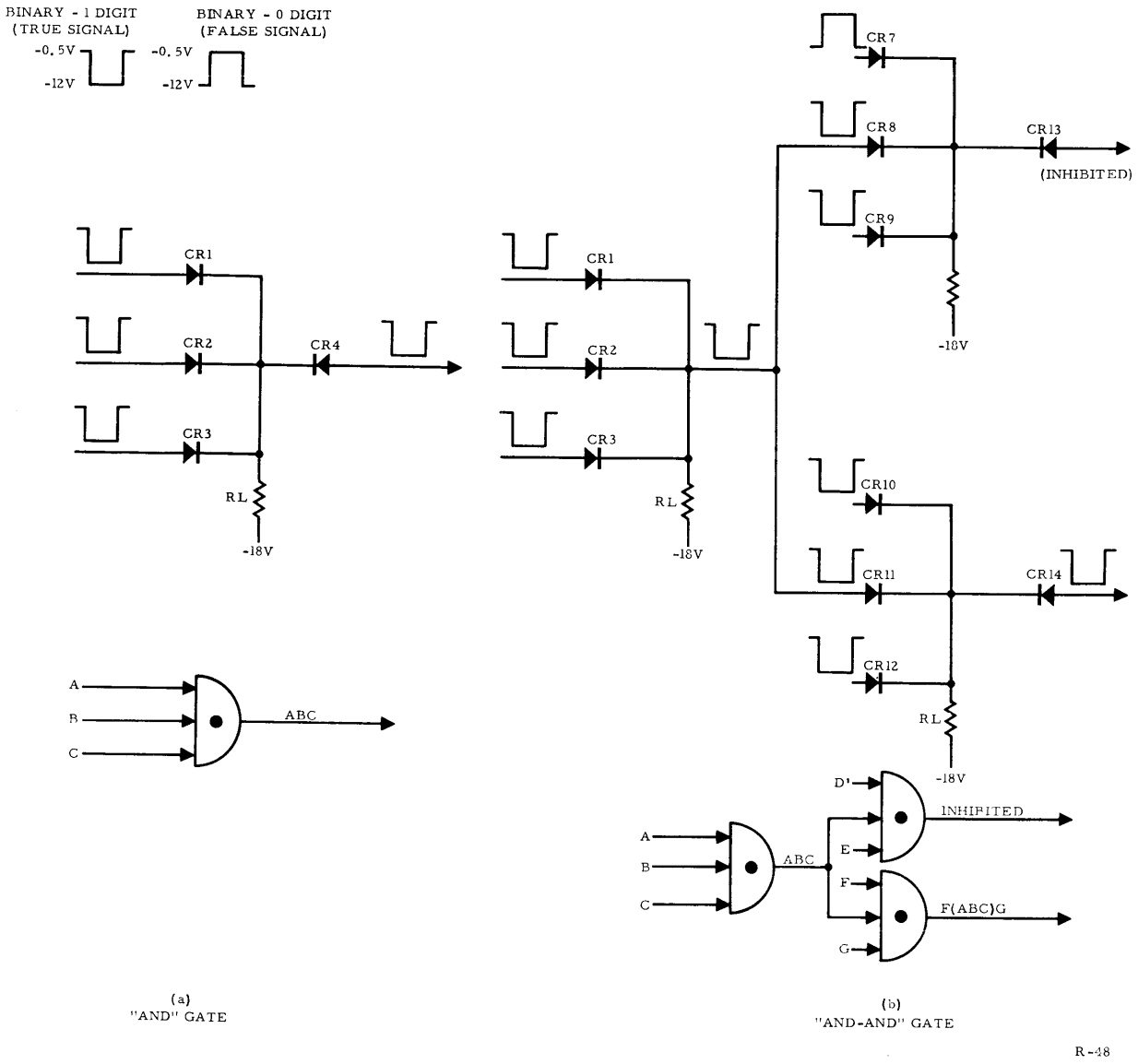


Figure 6-10. Typical Logic Gates and Symbols (Sheet 2)

"Or" Gate

The "or" gate has the characteristic of producing a true output signal if one or more of the input signals is true. View c of figure 6-10 illustrates a typical three-input "or" gate. CR1, CR2, and CR3 are the "or" diodes, and CR4 is the series "or" diode. The anodes of the "or" diodes are connected together in the circuit and in series with the series "or" diode. The series "or" diode anode is connected through a load resistor to the +6-volt supply.

As illustrated in figure 6-10, view c, when one input signal is false, the false input diode does not have the greatest potential difference between anode and cathode as in the "and" gate previously described. The true input diodes' conduction is greater, and they become the controlling factors in the circuit. Potential at the anodes drops to the -12-volt amplitude of the true input signals and back-biases the false input diode to cutoff. The true-input diode and series "or" diode then conduct heavily to produce a true output signal. When

all input signals are true, "or" diode conduction is equal and the gate is enabled in a like manner. Only when all input signals are false and the potential at the series "or" diode cathode rises to -0.5 volt, is the gate inhibited to produce a false output signal.

The "or" gate output signal is then the Boolean sum of the input signals as shown in figure 6-10, view c, and the "or" gate circuit symbol carries the addition sign.

"And-Or" Gate

An "and-or" gate is a combination of "and" and "or" gates connected to combine input signals of "and" gates and conduct the true-level output of one or more enabled "and" gates. The typical "and-or" gate shown in view d of figure 6-10 consists of two "and" gates and an "or" gate. The individual gates function as previously described for the two different types of gates. A true output from either "and" gate results in a true output from the "or" gate. Only when both "and" gate output signals are false does the "or" gate produce a false output.

The "and-or" gate output signals are the Boolean product and sum of the input signals as shown in figure 6-10, view d, and each "and" gate and "or" gate has a separate circuit symbol as previously described.

Logic Network Circuit Boards

All of the types of logic gates covered in the foregoing functional description are located on the basic logic network circuit boards. However, some logic gate components for signals entering or leaving a particular logic circuit board may not all be mounted on the same circuit board.

On the schematic of the basic logic network board (Service Manual), connector P1, pin 1 provides the -18-volt supply for each circuit board, and R1 through R14 are the gate load resistors. The diodes are all of the same type, and their function in a particular gate depends upon the method of connection on the circuit board and at the receptacles in the computer.

Diodes CR1 through CR9 form an eight-input "and" gate with the output at P1-14 on the schematic. CR1 is the disconnect diode, but any of the other diodes may also serve this function if required, depending upon the wiring at the connector. The "and" gates may be portions of "and-and" or "and-or" gates which have their remaining components located on other circuit boards. Diodes CR17 through CR28 form an "and-or" gate with CR22 and CR23 as the "or" portion of the gate. The series "or" diode in this instance is located on another circuit board. Each logic network board also contains one series "or" diode (CR40) which may be used to combine the outputs of other gates.

Diodes CR34 through CR38 and CR39 through CR43 form two "and" gates with a common input at P2-6. With the two "and" gate outputs connected to CR40 as shown on the schematic they form an "and-or" gate.

Logic Flip-Flop Circuits

RECOMP II system contains 42 logic flip-flop circuit boards, and the basic RECOMP III system,11. One additional logic flip-flop board is used in RECOMP III when the computer is equipped with the optional floating point arithmetic capability; another is added to form the extended input-output register when RECOMP III computer includes this optional item (refer to section 3).

Each logic flip-flop circuit board contains four identical logic flip-flop circuits. The circuit employed for each flip-flop is basically an Eccles-Jordan type bistable multivibrator with an amplifier stage integral to each output of the flip-flop. Nonsaturable operation of the flip-flop transistors is controlled by low-dissipation, high-speed, base-to-collector clamping circuits.

A logic flip-flop is fundamentally a memory element and is used in the computer as an information storage device with fast information transfer capability. Principal characteristics of the RECOMP logic flip-flop circuit which make it useful for computer application as a memory element are:

1. Two stable operating states.
2. Positive internal regenerative action which instantaneously carries the circuit from one state to another.
3. Circuit switching action which is designed to deliver maximum gain to the conducting transistor during regeneration process and simultaneous turn-off action to the transistor stage being driven to a nonconducting state.
4. Ability of the circuit to remain in a stable state, that is, retain (remember) information for an arbitrary length of time when no trigger pulse is applied.

The two stable operating states of computer logic flip-flops are termed 1-set state ("true-level" unprimed output) and 0-set state ("true-level" primed output). Refer to figure 6-11. In RECOMP computer logic a "true-level" signal has a potential of approximately -12 volts and a "false-level" signal is just below circuit ground at -0.5 volt. When a logic flip-flop is in a 1-set state, the unprimed output signal is at -12 volts and the primed output is at -0.5 volt. The converse is true when the flip-flop is in a 0-set state. Both output voltage levels are used in logic application. The -12-volt output is used to trigger other flip-flops or to enable logic gates, while the -0.5-volt output is used to inhibit flip-flops and logic gates.

The two voltage levels at the outputs of the flip-flop power amplifiers are applied in computer logic to represent binary ones and binary zeros. A 1-set state of the flip-flop circuit produces a binary-one condition at its unprimed output. When the flip-flop is 0-set, a binary-zero condition is produced at its primed output. The state of a flip-flop circuit remains unchanged until caused to change to the opposite state by application of appropriate input signals to the corresponding side of the flip-flop.

Logic flip-flops function singly or in combination with other flip-flops and gates in the computer as elements for information storage and transfer. Computer functional capabilities and operations in which logic flip-flops are

employed include: (1) extension and utility of the computer memory, (2) computer operation and mode selection, (3) data comparison, (4) arithmetic computation, and (5) selection of auxilliary input-output equipment when these units are operated in conjunction with the computer.

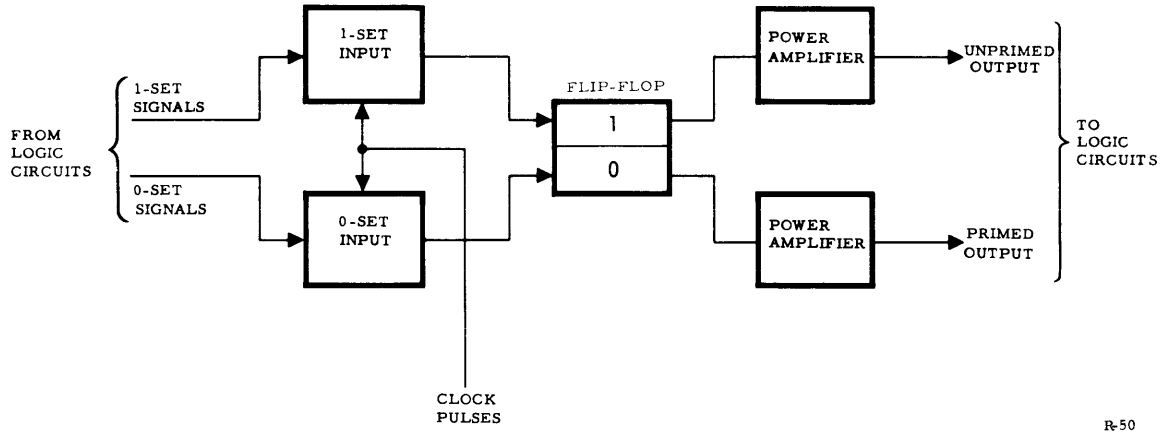


Figure 6-11. Logic Flip-Flop Block Diagram.

Flip-Flop Circuit Operation

Flip-flop circuit elements and their operation (figures 6-11 and 6-12) are symmetrical in configuration. Components of the 1-set side of the circuit are a mirror image (identical in value and method of circuit connection) of the 0-set side of the circuit. Difference of the two sides of the flip-flops lies only in its state of operation as determined by which side of the circuit receives a trigger pulse. The one exception to the circuit symmetry is that the 1-set side has an external connection for an input signal through a resistor (R6 on figure 6-12) which is used for testing purposes in forcing the flip-flop to change its state.

Operation of a typical flip-flop is illustrated in figure 6-11. The 1-set and 0-set input signals are outputs of logic gates and are gated with a clock pulse to the respective sides of the flip-flop circuit. A true-level input at the 1-set side and near ground condition at the 0-set input, causes the flip-flop to become 1-set during the true-to-false transition of the clock pulse. The unprimed output is approximately -12 volts, and the primed output is at -0.5 volt. A true-level input signal present at the 0-set input with near ground condition at the 1-set input causes the flip-flop to become 0-set during the true-to-false transition of the next clock pulse. In this state, the primed output signal is at approximately -12 volts, and the unprimed output amplifier conducts at near ground potential producing a false level output at -0.5 volt.

Figure 6-12 is the simplified schematic diagram of a logic flip-flop circuit. Q2 and Q3 are the flip-flop 1-set and 0-set n-p-n junction transistors which are connected in common emitter configuration. The collector of each transistor is cross coupled by an r-c network to the base of the other which provides

regenerative paths for the circuit to change state. Outputs of Q2 and Q3 supply the base drive current for p-n-p transistor amplifiers Q4 and Q1 respectively. The flip-flop change of state is caused by a trigger pulse which is developed at the input timing circuit of the corresponding flip-flop base drive network. (Generation of the trigger pulse is described in this section under the heading, Flip-Flop Timing Circuits.) The following circuit analysis describes voltage development paths of each flip-flop stable state and the voltage changes which occur during the change-of-state and regenerative process.

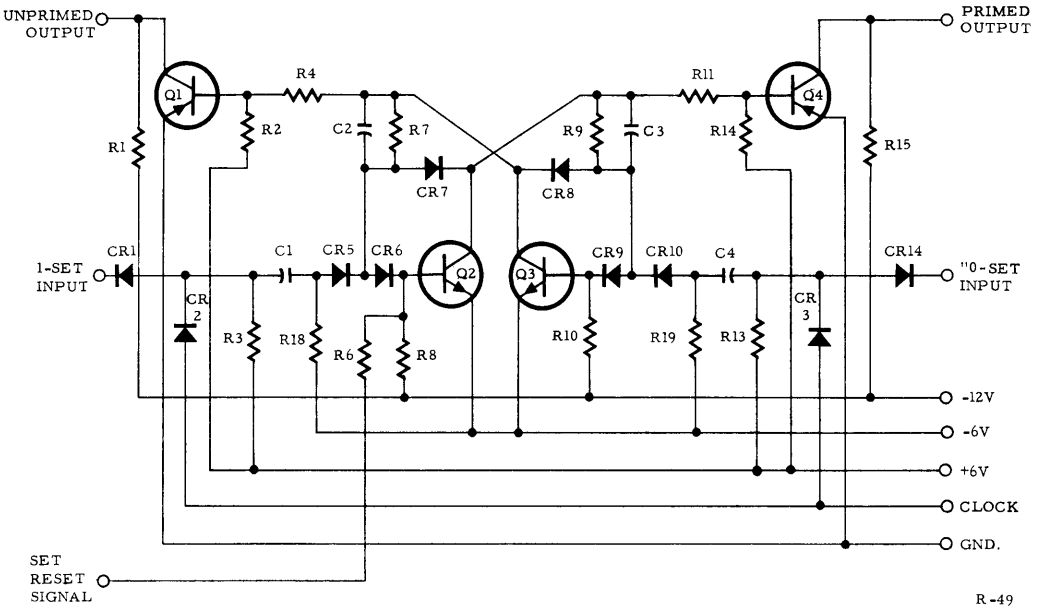


Figure 6-12. Logic Flip-Flop Simplified Schematic

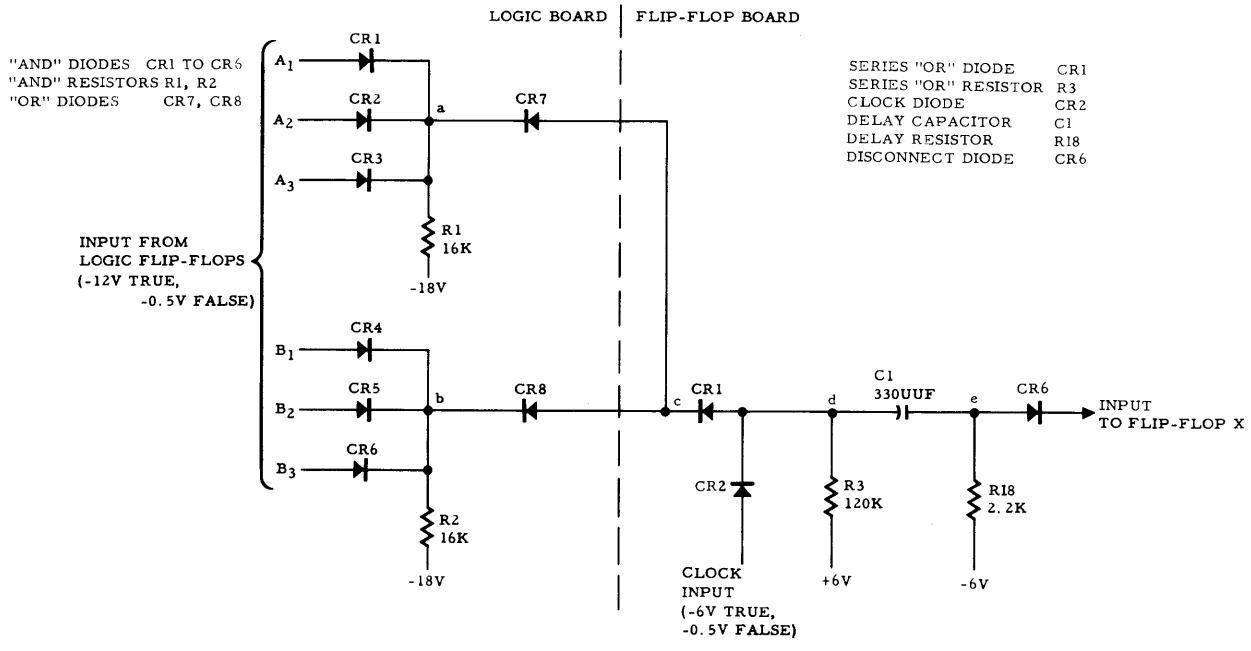


Figure 6-13. Typical Logic Flip-Flop Gating and Timing Circuit

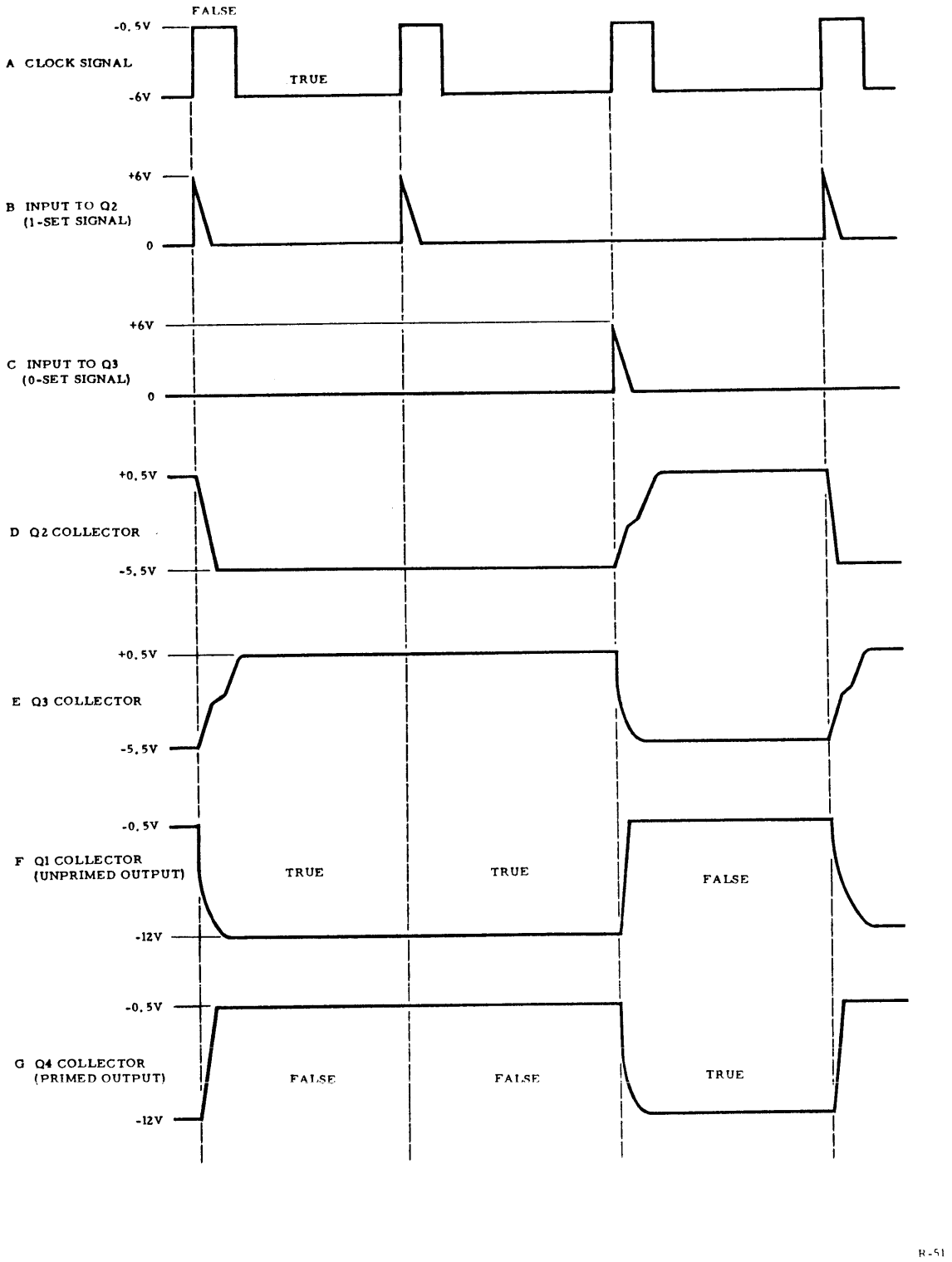


Figure 6-14. Logic Flip-Flop Waveforms

If the flip-flop is in a 0-set state, Q3 is conducting and Q2 is cut off. Transistor Q3 is in a positive-biased condition with its emitter at -6 volts, its base at approximately -5.7 volts, and its collector at approximately -5.5 volts. Transistor Q2 is in a negative-biased condition with its emitter also at -6 volts and its base equal to, or slightly more negative than, -6.5 volts. The negative voltage at Q2 base is produced by the combination of voltages applied across R8 from the -12-volt supply and across R7 from Q3 collector output. Q2 collector voltage is +0.5-volt. This voltage, applied across R9 in conjunction with the -12-volt supply across R10 to the base of Q3, maintains a positive bias at Q3 which keeps it conducting in the 0-set state.

P-n-p output amplifier transistors Q1 and Q4 are connected in a common grounded-emitter configuration (figure 6-12). With the flip-flop circuit in a 0-set state, the -5.5 volts at Q3 collector is applied through R4 to the base of Q1, dropping the base potential negative in respect to its emitter. In this negative-biased condition, Q1 conducts, producing a -0.5-volt potential at the unprimed output. Meanwhile, the base-to-emitter potential of Q4 amplifier is at a positive bias with respect to Q4 collector. In this state, Q4 is cut off, with its collector remaining at approximately the -12-volt potential of the supply across R15.

The flip-flop changes to the 1-set state when triggered by a +6-volt pulse applied to the base of Q2 (provided an equivalent trigger pulse is not applied simultaneously to the base of Q3). Capacitors C3 and C2 act as commutating couplers from the collector of one transistor to the base of the other, and improve switching time between stable states. This regenerative action provides maximum gain to the triggered transistor, and simultaneous turn-off action to the transistor being driven to the non-conducting state.

When the trigger pulse is applied to the base of Q2, a forward-biased condition is established at Q2 causing it to conduct. The circuit becomes regenerative and a rapid transition to the opposite state occurs. As Q2 conducts, its previous collector potential of +0.5-volt drops to a more negative value which is coupled to the base of Q3 across capacitor C3. With less potential applied across Q3, the conduction of this transistor decreases. The more positive voltage developed at Q3 collector, in turn, is coupled back to the base of Q2 across capacitor C2 and increases the positive bias of Q2. Consequently, Q2 rate of conduction is increased. Regenerative action continues until Q2 is conducting at near saturation and Q3 becomes negative biased to cutoff.

Transistors Q2 and Q3 are operated non-saturated by means of low-dissipation, high-speed collector-to-base clamping circuits. Diodes with different forward-voltage drops are connected in parallel between the base and collector of Q2 and Q3 to control the amount of bias on the transistor that is conducting. In figure 6-12 germanium diodes CR7 and CR8 are connected to the collectors, and silicon diodes CR6 and CR9 are connected to the bases of the flip-flop transistors. To maintain the collectors at a more positive potential than the base circuits, the germanium diodes conduct with a 0.3-volt drop as compared to a 0.6-volt drop for the silicon diodes. While regenerative action of the flip-flop circuit causes the conducting transistor to progressively increase in conduction toward saturation, the clamping network allows the conducting transistor to increase in conduction until it approaches its saturation level. At this time, the feedback diode becomes forward-biased, conducts, and clamps the collector of the conducting transistor to +0.3-volt above its base-to-emitter potential.

During the transition from a 0-set state to a 1-set state, and after attainment of circuit equilibrium, diodes CR6 and CR7 (figure 6-12) provide base drive control and clamping action for transistor Q2. When the trigger pulse is applied to the base of Q2, causing it to conduct, CR6 is conducting to drive Q2 toward saturation. Meanwhile, CR7 remains reverse biased to cutoff until Q2 collector reaches approximately -5.5 volts. At this point, CR7 becomes forward biased and conducts to clamp the base circuit at approximately 0.3-volt more negative than its collector. This is accomplished by the forward-voltage drop differential between diodes CR6 and CR7. Degenerative action produced by this clamping network controls Q2 conduction to approximately 0.3 volt below the transistor's operating level of saturation.

Flip-Flop Timing Circuits

Triggering of all flip-flops is synchronized with clock signals originating in the memory to establish the correct time relationships between the flip-flops and other logic circuits. The input circuits of each flip-flop generate a trigger pulse if a true-level input signal is present at one of the inputs during the true-to-false transition of the next clock pulse.

Figure 6-13 illustrates operation of a flip-flop input circuit that generates the trigger pulse. Assume that input signals $A_1A_2A_3$ in figure 6-13 are true. "Or" diode CR7 conducts, and the voltages at points "a" and "c" start to fall toward a negative value. Diode CR1 also starts to conduct, but clock diode CR2 is cut off by the -6-volt true clock signal present at the anode. The voltage at point "d" starts to fall negatively at an exponential rate determined by the r-c time constant of capacitor C1 and resistor R18. When the voltage at point "d" reaches -6 volts, CR2 conducts and clamps the voltage to that level. C1 charges to -6 volts, and it is at this time that the capacitor stores the information that flip-flop "X" is to be 1-set at the true-to-false transition of the next clock pulse.

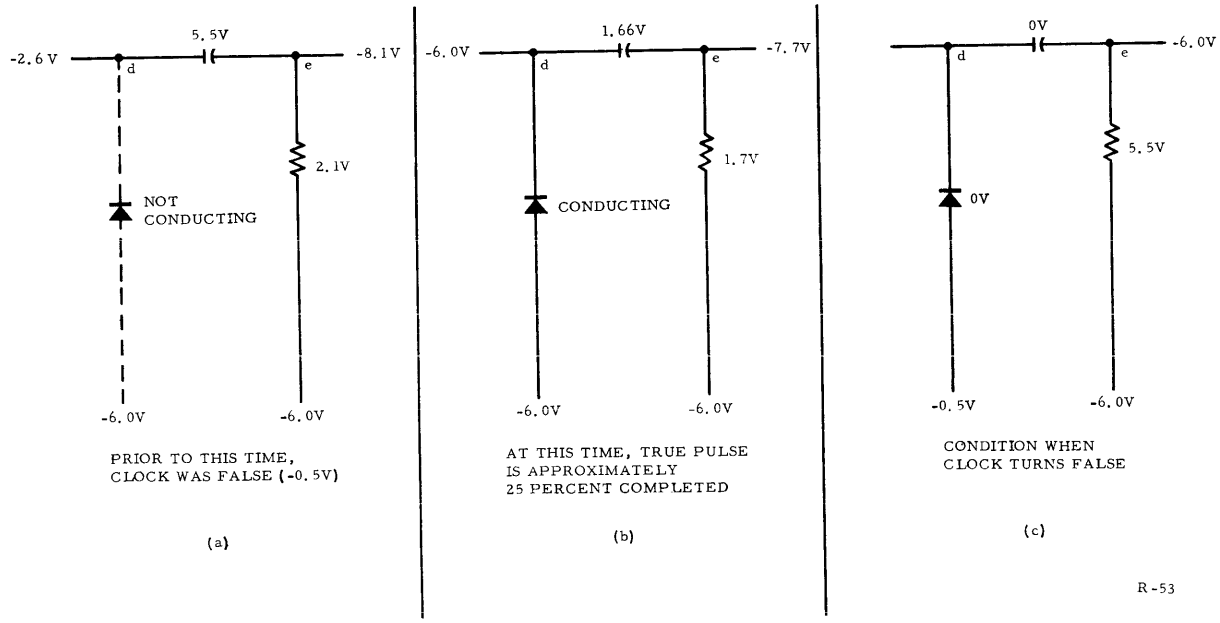


Figure 6-15. Potentials at Delay Section of Timing Circuits.

The potentials at points "d" and "e" of figure 6-13 are illustrated in figure 6-15 at (a) the time the clock signal goes true, (b) at approximately 25 percent of the clock true period, and (c) at the time the clock signal goes false. Observe in (a) that points "d" and "e" are at -2.6 volts and -8.1 volts, respectively, when the clock signal goes true.

In (b) of figure 6-15 the clock diode conducts when the voltage at point "d" reaches -6 volts, and the voltage at point "e" has decreased to -7.7 volts. At the occurrence of the true-to-false transition of the clock signal in (c) the voltage across the capacitor is zero and point "e" is at -6 volts. At the time the clock signal goes false, points "d" and "e" are driven in a positive direction because the diode is conducting and the clock signal rise time is very short. Point "e" in figure 6-13 rises toward ground potential, and diode CR6 conducts to transmit a positive trigger pulse to the flip-flop circuit.

The input impedance of the n-p-n flip-flop transistor is very low at the time the trigger pulse is applied, and capacitor C1(figure 6-13) discharges very rapidly to return point "e" to a nominal -6-volt potential by the end of the clock false period.

Had both input "and" gates been enabled at the true clock period, C1 would have charged up to the clamping point at almost twice the rate, since the "and" gate load resistors then are electrically in parallel. With additional "and" gates enabled, C1 reaches its ultimate charge faster, but the clamping action of CR2 is the same.

When all of the "and" gates are inhibited, CR2 remains reverse-biased and C1 uncharged. Then, as the clock signal goes false, CR2 is unbiased, point "e" remains at -6 volts, and no trigger pulse is generated by the timing circuit.

Logic Driver Circuits

The logic driver circuit boards are located on the RECOMP III computer chassis in the 100-200 series row at connector numbers 10 through 16. Each board contains three transistorized drivers (figure 7-34 of RECOMP III Service Manual) which, under certain circumstances, are used to amplify the signal current from the logic primary "and" gates and the logic flip-flops. This amplification is required when a single primary gate, or flip-flop, drives more than approximately 60 secondary gates.

Eleven primary "and" gates are also mounted on each logic driver board which are not directly connected internally on the board to the driver stages. Some of these primary gates, however, are wired through the connectors to the amplifiers where they have sufficient secondary loadings to require the additional driving current.

The prime reason that logic drivers are used in RECOMP III is to provide a large amount of ground current for operation of many secondary gates. Each of the three logic drivers on a board consists of two emitter-follower amplifier stages. The first stage is a diode-coupled p-n-p transistor; the second, an n-p-n transistor (figure 6-16). Output of each stage retains the same phase relationship as is applied to the input. The operation of each stage is contingent upon the type of input signal (true or false) received.

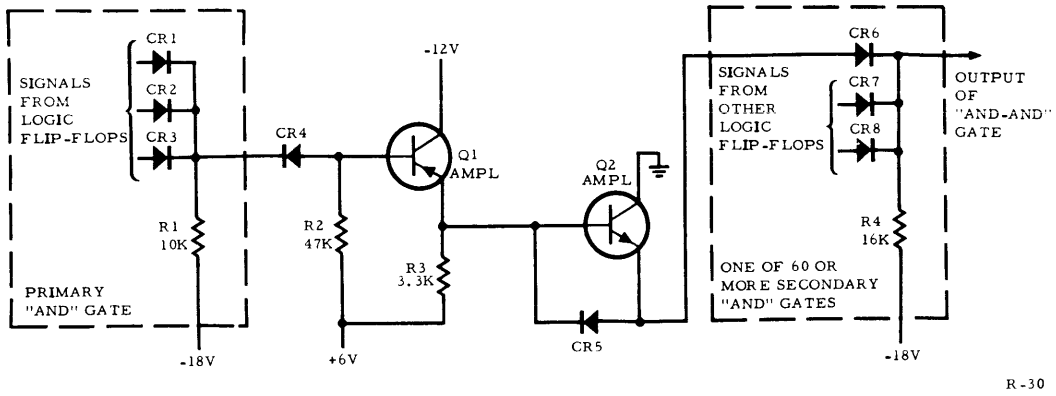


Figure 6-16. Logic Driver Simplified Schematic

P-n-p transistor Q1 is connected in the circuit as an emitter-follower current amplifier and is always in a conducting state. When an input signal to the driver circuit is false (-0.5 volt) Q1 conduction reproduces the positive voltage (approximately +0.2 volt) which is present at its base (figure 6-16). This positive voltage is applied to the base of n-p-n transistor Q2 and establishes a positive bias with respect to its emitter, and consequently causes Q2 to conduct. Having a grounded collector, Q2 conducts with sufficient current amplification to drive many secondary logic gates with a false level (approximately -0.5 volt) signal. Diode CR5 is reverse-biased and remains non-conductive during false-level input signals.

When a true level (-12-volt) signal is applied to the logic driver input, transistor Q1 amplifies the current of the -12-volt signal. This negative potential, applied to the base of Q2, causes Q2 to become negative biased and its conduction to be cut off. Diode CR5, however, becomes forward biased and conducts the amplified -12-volt signal with sufficient current to drive many secondary gates.

Write Switching Circuits

Eight identical circuit boards contain the write switching network circuits for selecting the specific memory write head to record a computer word into the main memory channel addressed by a command. Each board contains eight transistor switches, one switch of the eight for each of eight main memory write heads. All eight boards thus contain 64 switches for the 64 main memory channel write heads.

Figure 6-17 shows the matrix form of the write switching network circuitry and illustrates the method by which the write head of an addressed channel is selected. Unprimed and primed outputs of channel selector flip-flops C_1 through C_6 supply input signals to two selecting "and" gates located on the write switching network circuit boards. Flip-flops C_1 through C_3 supply signals to one selecting "and" gate on a specific board; the board which will receive

these signals is determined by wire routing of the various combinations of primed and unprimed outputs of the C_1, C_2, C_3 flip-flops. Output of the $C_1C_2C_3$ gate is connected to a bus which routes a signal to one switch input "and" gate on each of the eight write switching boards. Flip-flops C_4 through C_6, meanwhile, supply signals to another selecting "and" gate on a specific board; the board receiving these signals is determined similarly through wire routing of the various combinations of primed and unprimed outputs of the C_4, C_5, and C_6 flip-flops. Output of the $C_4C_5C_6$ gate is connected to a bus which routes a signal to all eight switch input"and" gates on that selected write switching board.

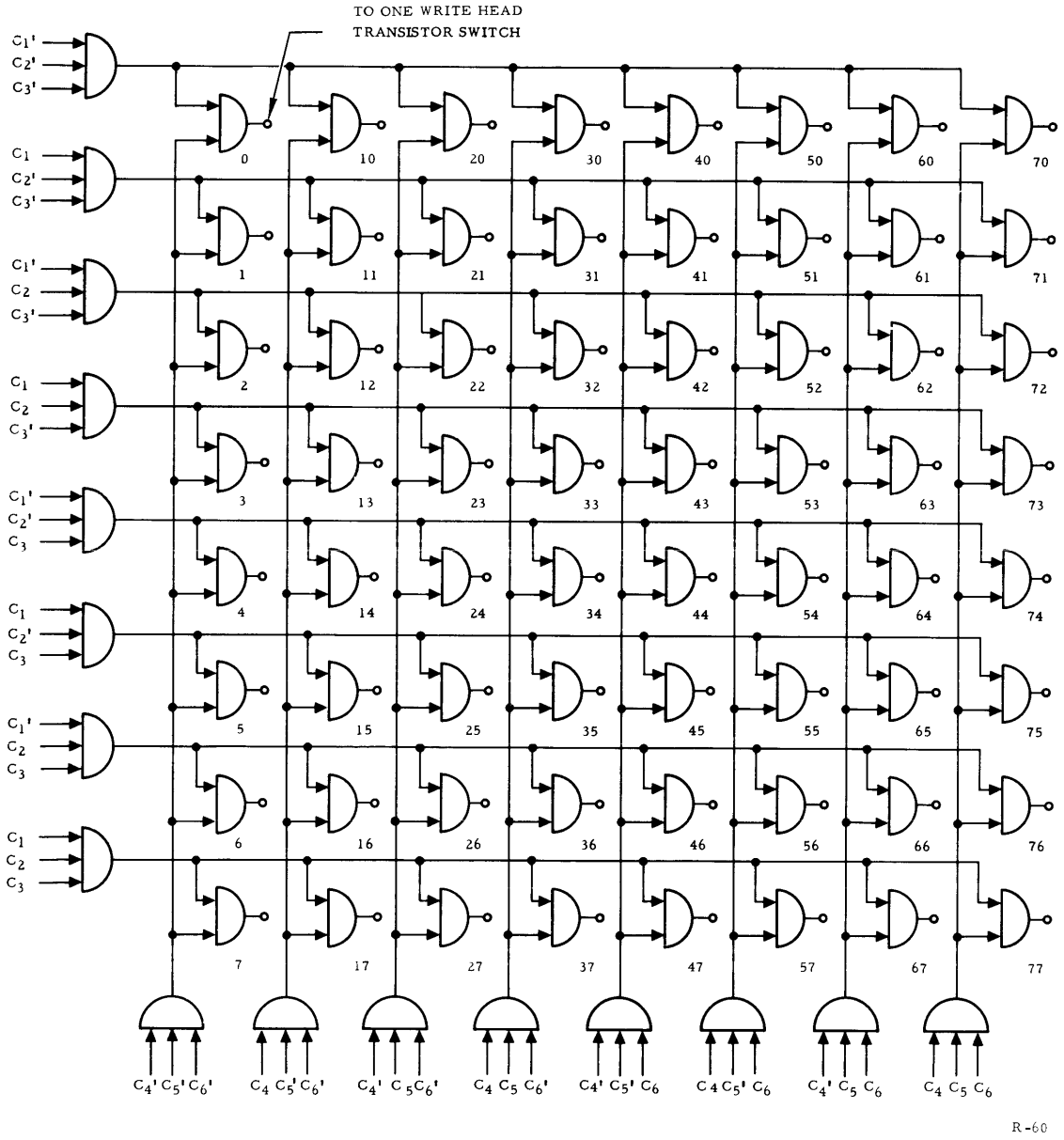


Figure 6-17. Write Head Selection Gating Matrix.

Only one of the eight possible combinations of signals from each of the two flip-flop groups can all be true during any one given time. Therefore, only one transistor switch input "and" gate can be enabled during a particular time, and only the write head of the addressed channel is permitted to record information at that time.

As an example of head selection, if C_1', C_2', C_3', C_4', C_5', and C_6' were all true during a particular word time, a selecting "and" gate would be enabled on the board that is wired to receive $C_1'C_2'C_3'$ configuration and a selecting "and" gate would be enabled on the board that is wired to receive $C_4'C_5'C_6'$ configuration. Output of $C_1'C_2'C_3'$ is routed to a same pin (and a same switch input "and" gate) on each board; output of $C_4'C_5'C_6'$, to all eight switch "and" gates on the board receiving this signal configuration. The two signals, where coincidient, then enable the switch input "and" gate for the transistor switch of write head 00 (figure 6-17). Conduction of this switch allows the write head for channel 00 to record information in that channel.

The schematic of the write switching network circuit board is presented in the Service Manual. Diodes CR17 through CR19 and diodes CR36 through CR38 form the two signal bus "and" gates. Diodes CR20 through CR35 form the switch input "and" gates of the eight transistor switches. Ouptuts of the two signal bus gates are connected to the switch input gates to form cascaded "and-and" gates. The switch input "and" gates control the base voltage of the transistors.

Figure 6-18 is a simplified schematic of one write switch circuit. The collector of the transistor switch is connected to the center tap of the corresponding write head coil. When the input signals from the channel selector flip-flops are all true, the switch input "and" gate is enabled and the -12-volt output overcomes the potential barrier between the transistor emitter and base. The transistor conducts and establishes a -3-volt potential at the center tap of the write coil. Current goes through the collector output line and one-half of the write coil when either isolation diode at the input gate conducts. The M_{w1} and

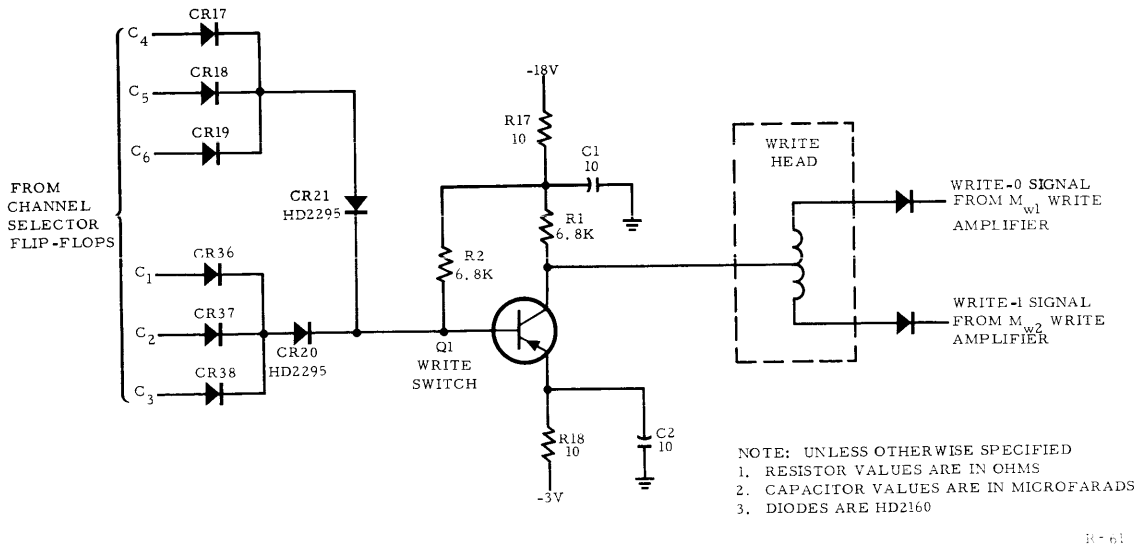


Figure 6-18. Simplified Schematic for One Write Switch

M_{w2} information channel write amplifiers supply -12-volt write signals to the corresponding write -0 or write -1 inputs of the 64 write heads simultaneously. Only that isolation diode of the addressed channel write head, which is forward-biased by the -3-volt potential at the center tap and the applied -12-volt write signal, conducts and permits write current to flow through the corresponding half of the write coil.

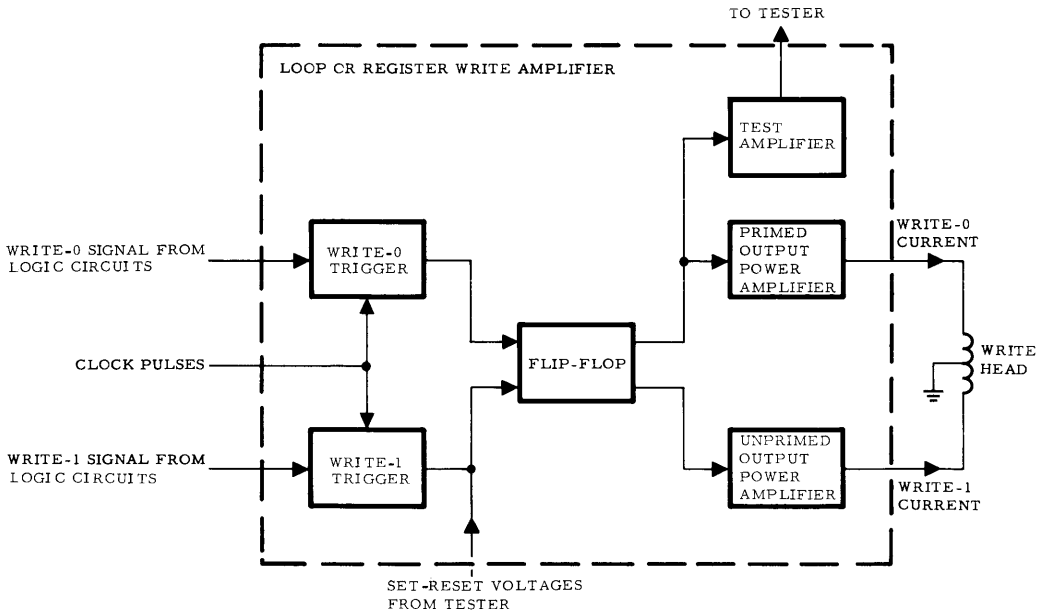
The write switching network circuit boards also contain the isolation diodes utilized at the write head coil inputs. Write signals from either of the write amplifiers are applied simultaneously to the cathodes of eight diodes on the board indicated and to similar diodes matrices on each of the seven remaining circuit boards.

Write Amplifier Circuits

Three write amplifier circuit boards, located in the computer chassis, generate the pulses of write current which record information on the magnetic memory disk. Each write amplifier circuit board contains three write amplifier circuits. Seven of the available nine write amplifier circuits are used with the two rapid-access loops and the five registers; the remaining circuits are used with the 64 main memory channels.

Loop and Register Write Amplifiers

Figure 6-19 illustrates a typical loop and register write amplifier circuit. The circuit consists of a flip-flop, an amplifier stage at each flip-flop output, and an additional amplifier that provides signals for monitoring and, in some instances, logic gate inputs.



R-62

Figure 6-19. Typical Loop or Register Write Amplifier Block Diagram

Write-1 and write-0 signals from logic gates enable the input "and-or" diodes of the write amplifier during the clock signal true interval. At the true-to-false transition of the clock signal, the enabled diode and the flip-flop timing circuit generate a pulse which triggers the flip-flop. A write-0 signal results in conduction of the 0-input flip-flop transistor, and the primed-output power amplifier conducts through one-half of the write head coil to record a binary 0 on the magnetic disk. A write-1 signal results in conduction of the 1-input flip-flop transistor, and the unprimed-output power amplifier conducts through the opposite half of the write head coil to record a binary 1 on the disk.

The schematic of the write amplifier circuit board is given in the Service Manual. On the circuit board illustrated, write amplifier No. 2 is utilized with rapid-access loop V, while write amplifiers Nos. 1 and 3 are the main memory write amplifiers M_{w1} and M_{w2}. The configuration and operation of write amplifier No. 2 is typical of all loop and register write amplifiers, and this circuit is used as a reference in the functional description that follows.

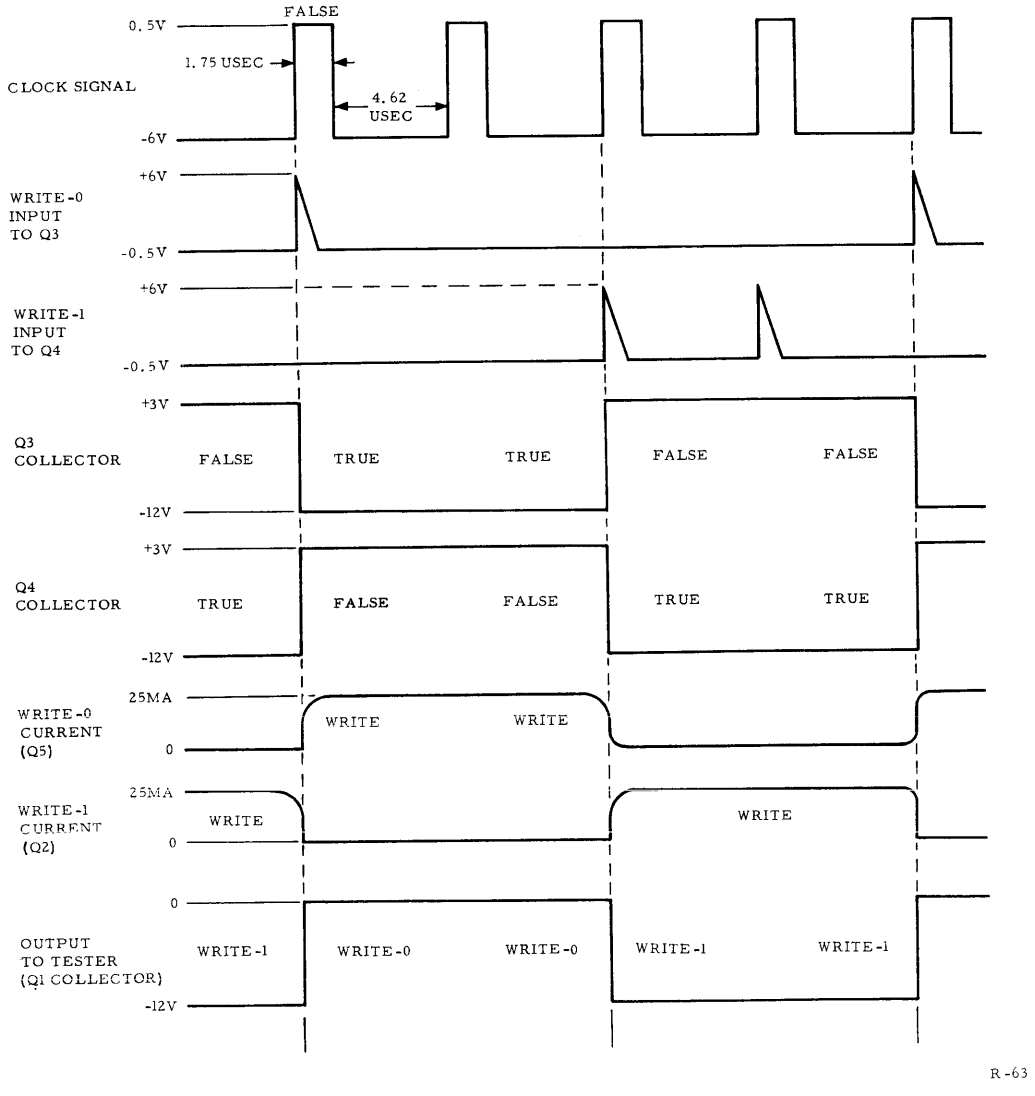


Figure 6-20. Write Amplifier Waveforms

The flip-flop consists of n-p-n transistors Q3 and Q4, the collectors of which supply the base drive for output amplifiers Q2 and Q5. Gating and timing of the flip-flop is identical to that of the other computer flip-flops (refer to Flip-Flop Timing in this section) except that resistor R20 is connected to -12 volts rather than -6 volts. Amplitude of the input trigger pulse is from -0.5 to +6 volts, as illustrated in figure 6-20. This illustration also shows typical write amplifier waveforms and voltage levels.

Assuming that the flip-flop is in the write-1 state, transistor Q3 is cut off by the conduction of Q4 through load resistor R6. The Q3 collector potential is +3 volts and the base potential is -14 volts. A positive write-0 pulse applied to the base of Q3 triggers that transistor into conduction, and the negative increase in Q3 collector voltage is coupled through capacitor C3 and cuts Q4 off. The collector and base potentials of Q4 assume the same cutoff values as those previously applicable to Q3.

Prior to the conduction of Q3, power amplifier Q5 was cut off by the +3-volt collector voltage of Q3 applied to the base, and current through the write-0 half of the write head coil was zero. The negative increase in Q3 collector voltage from +3 to -12 volts drives Q5 into conduction and the emitter potential increases to -12 volts. Q5 emitter is connected to ground through the resistor-diode combination of R5 and CR3 and the write-0 half of the write head coil. CR3 conducts when forward-biased by the -12-volt emitter potential of Q5, resulting in current going from ground through the write-0 half of the coil and the power amplifier to the -12-volt supply. The 25 milliamperes of write current through the write head records a binary 0 in the selected loop or register sector of the loop-register channel on the magnetic disk.

A write-1 pulse applied to the base of Q4 triggers the flip-flop to the opposite state, and power amplifier Q2 conducts to record a binary 1 on the disk in a like manner. Test power amplifier Q1 is connected in parallel with Q2 to provide write-1 signals for the system tester (when connected). When Q4 conducts, the negative increase in collector potential drives Q1 into saturation, and the output is a true signal with an amplitude of 0 to -12 volts. Diode CR4 permits rapid recovery of Q1 from the saturated state by cutting off the base drive to the test power amplifier when the collector potential of Q4 goes positive at the occurrence of a write-0 input pulse.

Information Channel Write Amplifiers

Figure 6-21 is a block diagram of information channel write amplifiers M_{w1} and M_{w2}. These amplifiers are identical in configuration to the loop and register amplifiers except that load resistors R115 and R315 are 150 ohms. The unprimed-output power amplifiers are not used, and the primed-output power amplifiers provide both the write-1 and write-0 current. The two flip-flops thus provide four (2^2) possible combinations of output signals to represent write commands from the computer logic as follows:

AMPLIFIER M_{w1}	AMPLIFIER M_{w2}
0	0
1	0
0	1
1	1

In condition 10, M_{w1} is 1-set and M_{w2} is 0-set by a write-1 input signal, and the M_{w2} primed-output power amplifier produces the write-1 current. Conversely, in condition 01, M_{w2} is 1-set and M_{w1} is 0-set by a write-0 input signal and the M_{w1} primed-output power amplifier produces the write-0 current. In condition 11, both M_{w1} and M_{w2} are 1-set and both primed-output power amplifiers are cut off for a "not write" command. Condition 00, or "both write", has no application in computer logic and is never used.

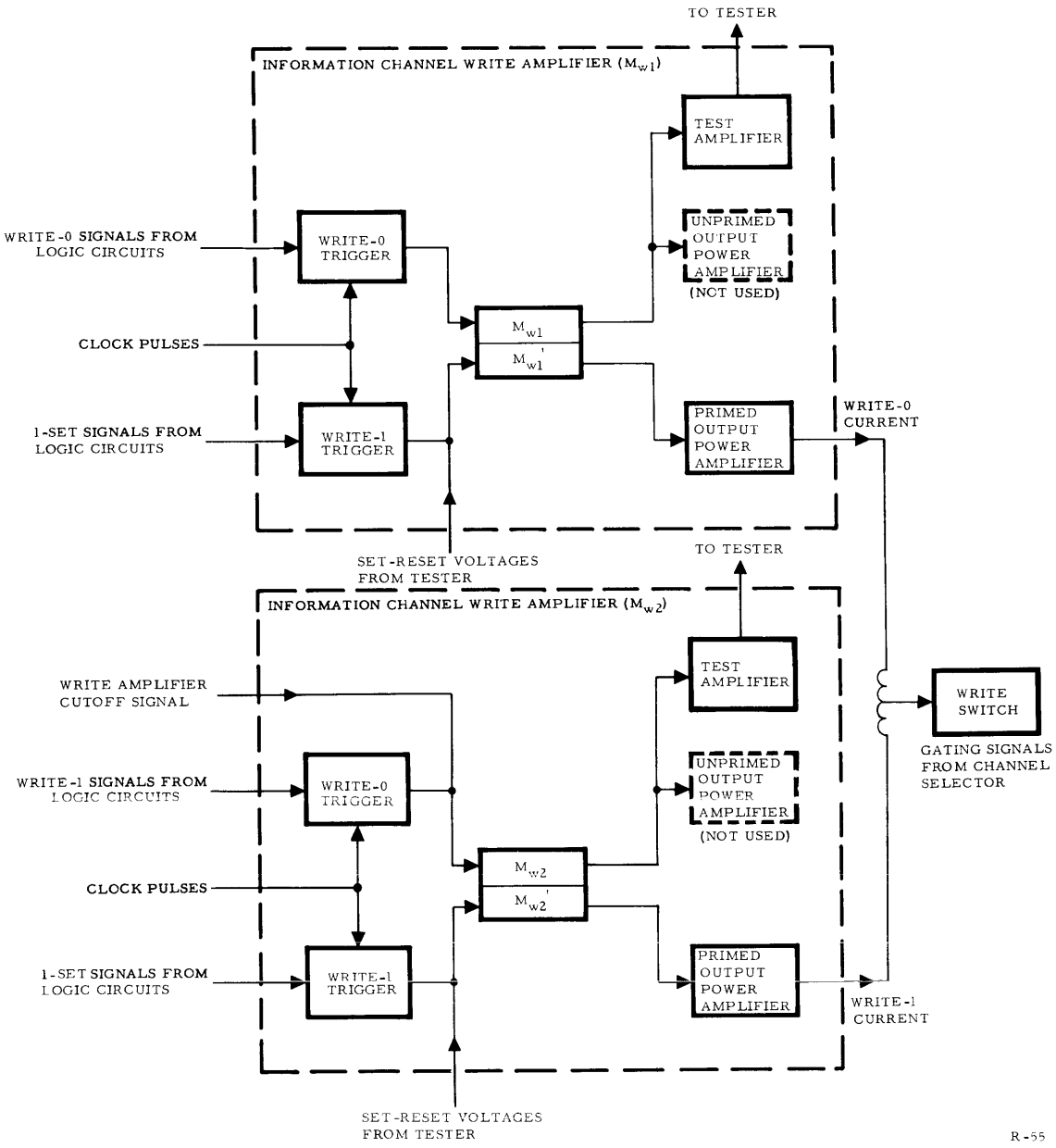


Figure 6-21. Information Channel Write Amplifier Block Diagram

Operation of the information channel write amplifiers is essentially the same as that of the loop and register amplifiers. The center tap of the write head coils is not grounded, but is connected to the collectors of the write switching transistors. Conduction of the information channel primed-output amplifiers occurs only when the write switch of a selected channel conducts and registers a -3-volt potential at the center tap of the selected write head coil. The write-1 or write-0 current through the corresponding half of the coil (as in the loop and register amplifiers) then records the information on the disk.

Write Amplifier Cutoff Circuit (TB2)

The write amplifier cutoff circuit board (TB2), located on the inside of the hinged computer chassis (figure 7-6 of RECOMP III Service Manual), provides a signal to cut off main memory information channel "write-1" amplifier M_{w2} at bit time 1.

In addition to the cutoff circuit, the board also contains circuits which serve as decoupling elements for the input gates to the logic flip-flops (Figure 7-40 of RECOMP III Service Manual). Capacitors C5 through C16 and C18 filter the logic flip-flop supply voltages to ensure that transient voltages do not trigger the flip-flops.

The write amplifier cutoff signal ensures that correct information is stored in the main memory channels synchronous with the timing of the memory by prohibiting a binary 1 of the preceding word sector from being recorded at bit time 2 of the succeeding word sector. (The computer records the sync bit at bit time 1, the least significant bit of the 40-bit information word at bit time 2, the most significant bit of the information word at bit time 40, and the sign bit at bit time 41.)

Sync bit pulse T_1, clock pulse C, and the output of a one-shot multivibrator provide the input signals to an "and" gate composed of diodes CR4, CR5, and CR7 (figure 7-40 of RECOMP III Service Manual). Strobe pulse C' is the input to the one-shot multivibrator composed of p-n-p transistors Q1 and Q2. The output of the "and" gate through threshold diode CR9 is the write amplifier cutoff signal C/O. Figure 6-22 shows the timing relationships between the input waveforms and the output waveform of the write amplifier cutoff circuit.

The sync bit pulse is true only at bit time 1 of each word sector time; the clock pulse alternates between false and true levels 41 times in each word sector time; the strobe signal (complement of the clock signal) alternates between true and false levels at the clock signal rate. When the output of the one-shot multivibrator is true at bit time 1, the "and" gate is enabled and produces the write amplifier cutoff signal.

Zener diode CR9 is the threshold diode of the CR4, CR5, CR7 "and" gate. With the anode of CR9 connected to the -18-volt supply across resistor R8, and when any one of the input signals are false, CR9 breaks down and maintains a -7-volt drop at this anode. The write amplifier cutoff signal is then at -7-volts during the period that the "and" gate is inhibited and does not affect operation of the write amplifier circuits.

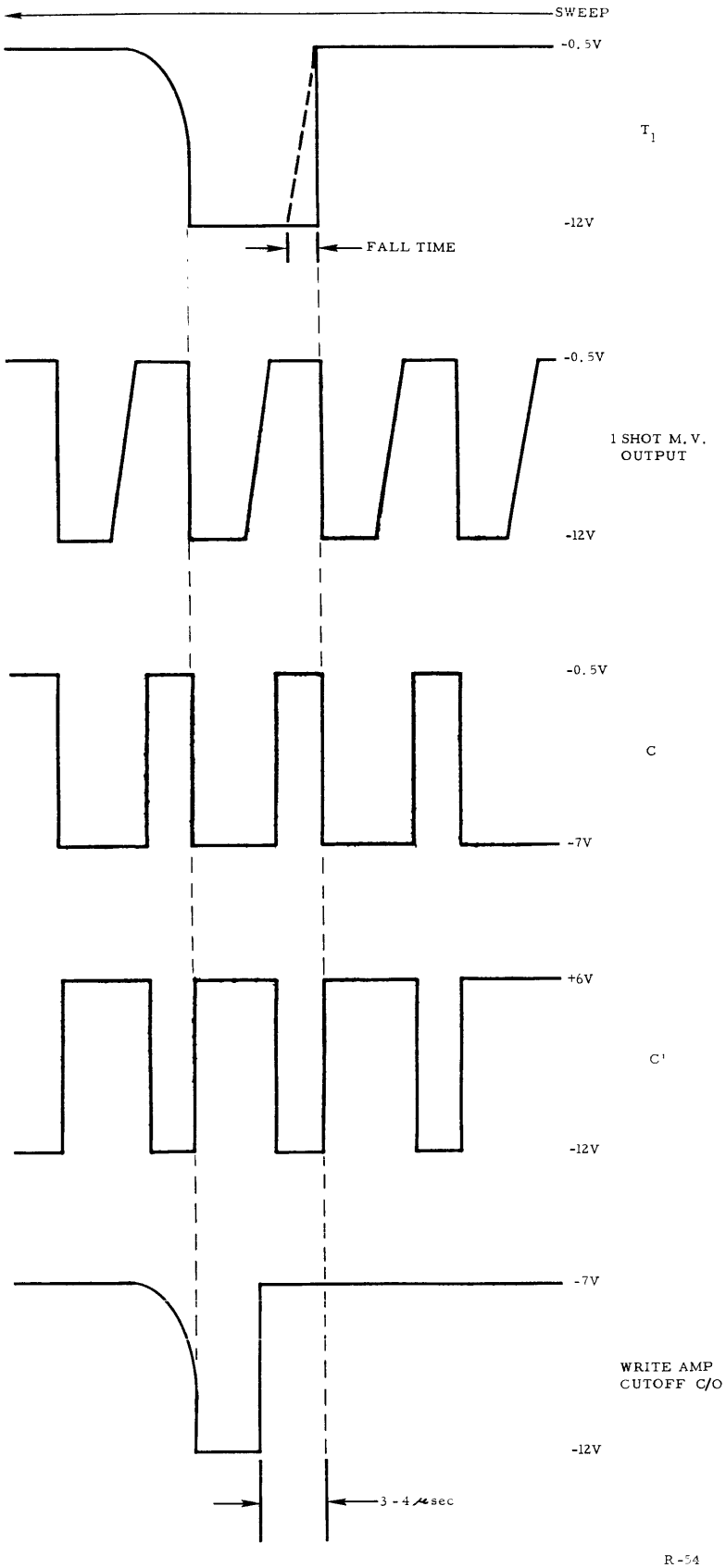


Figure 6-22. Write Amplifier Cutoff Circuit Waveforms

Q1 of the one-shot multivibrator normally conducts with Q2 cut off by +6 volts applied across R6 to the base. The collector voltage of Q2 is -12 volts at cutoff and is at ground potential when Q2 is conducting. The "and" gate will then be enabled when Q2 registers a true output in coincidence with T_1 and C true signals at the other two "and" gate inputs. As sync bit pulse T_1^1 is only true at bit time 1, the "and" gate can only be enabled during that period.

Input signals to the write amplifier cutoff circuit are not perfect square waves as figure 6-22 illustrates. To ensure positive cutoff of the write amplifier, the "and" gate is not enabled at the false-to-true transition of T_1, but is delayed beyond the fall time period of the waveform. The one-shot multivibrator provides the necessary 3-to-4 microsecond delay.

The strobe signal applied to the write amplifier cutoff circuit continuously alternates between +6- and -12-volt levels. C17 and R9 form a short-time constant to the strobe pulse and differentiate the square wave into positive and negative spikes. Diode CR11 limits the positive portion of the waveform by clamping it to ground potential. The trigger pulse applied to the base of Q2 through capacitor C2 is then alternating between 0 and -12 volts with a 6.32-microsecond interval between pulses. The strobe pulse will then trigger the one-shot multivibrator at the beginning of each bit time of the word, and the false output inhibits the "and" gate for approximately 3 microseconds of the 6.32-microsecond bit time period. After 3 microseconds, Q2 reverts to the stable state and registers a true output until again triggered by the strobe pulse.

Thus, the "and" gate is enabled approximately 3 to 4 microseconds after the beginning of bit time 1 when all gate inputs are true. Zener diode CR9 conducts to register the -12-volt write amplifier cutoff signal at the output. This negative voltage, applied to the base circuit of information channel write amplifier M_{w2}, results in a false output from the primed-output power amplifier to prohibit the inadvertent recording of a binary-1 bit at bit time 2.

Read Switching Circuits

Eight identical circuit boards comprise the computer read switching network circuits. Each circuit board has eight transistor switches, with the eight boards thus containing the 64 read switches for the 64 read heads which reproduce information from the main memory channels. Information from the channel which contains the registers and high-speed loops does not require selection by switching circuits and is therefore read directly into the read amplifiers (refer to Read Amplifier Circuits).

Information is reproduced by the read heads from all storage channels simultaneously and continuously throughout computer operation. The eight read switching network circuits select eight of these 64 available read head signals, as determined by the least significant octal digit of the channel address, and apply them to the eight main memory read amplifier circuits. Logic gates then select the amplifier circuit corresponding to the most significant octal digit of the channel address and apply its output to the main memory read flip-flop.

Figure 6-23 illustrates how each read switching network circuit selects one read head signal from among the eight available. The diagram also illustrates the additional selection of the read amplifier circuit by the logic gates which determine if the output of a particular read switching network circuit is that of the information channel addressed.

Signals from the read heads of channels 0 through 7 are applied to transistor switches M_{r0} through M_{r7}, respectively, during each bit time. A transistor driver amplifier on each read switching network circuit board controls the conduction of eight switches, one on the circuit board on which it is located and one on each of the other seven associated circuit boards.

In figure 6-23, the driver 0 on the circuit board illustrated controls the conduction of switch M_{r0} and that of seven other switches indicated at the gating signal output. Signals from the channel selector flip-flops C_1, C_2, and C_3 in

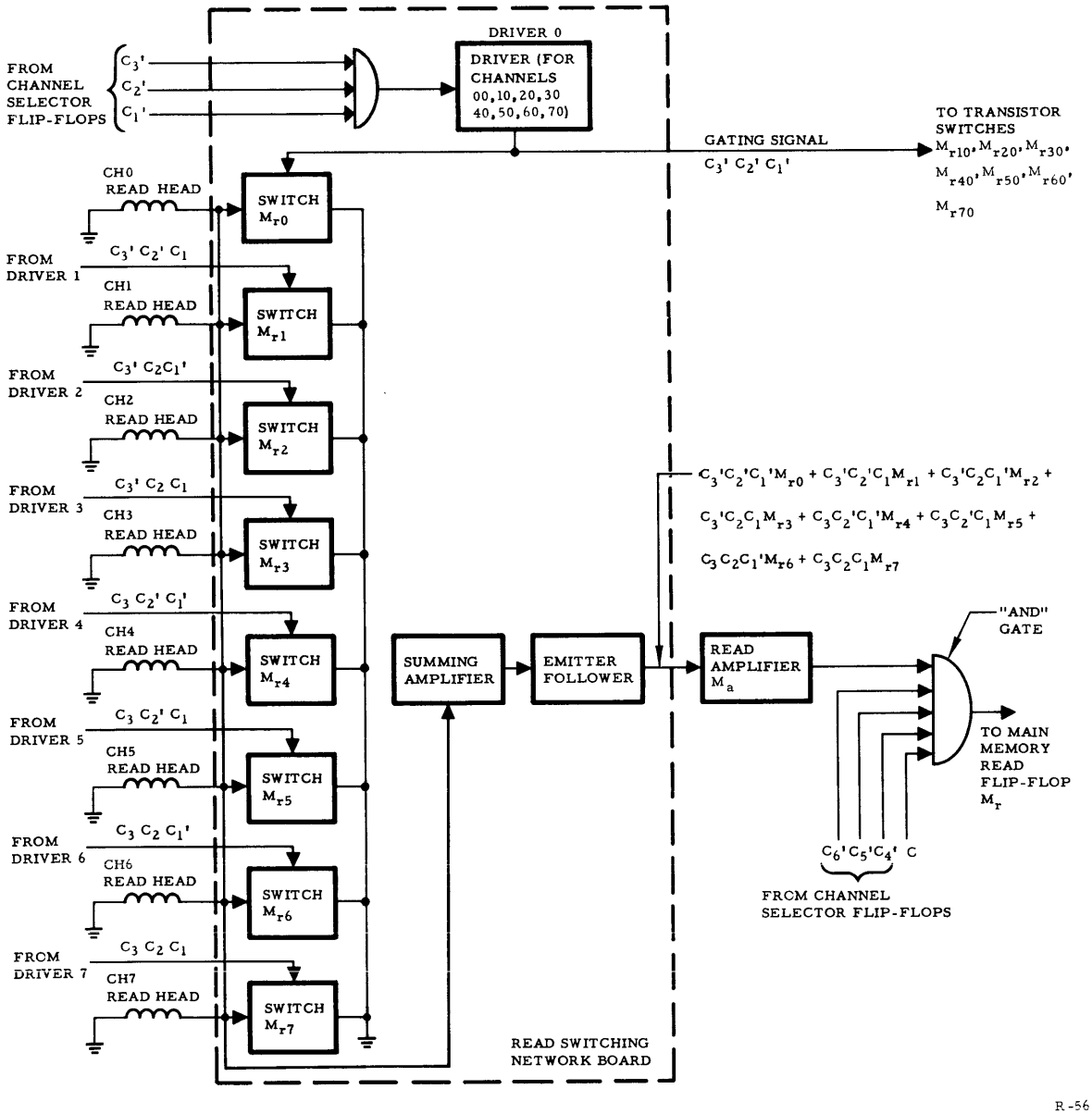


Figure 6-23. Read Switching Network Block Diagram

turn control the conduction of driver 0 and the other drivers. In the example in figure 6-23, driver 0 conducts only when its input "and" gate is enabled by true signals from the primed outputs of flip-flops C_1, C_2, and C_3. The "and" gate inputs of the seven remaining drivers require all true signals from a different combination of the three flip-flops' unprimed and primed outputs before they are enabled. Thus, during any given time, only one combination of channel selector gate input signals can be true and only one driver of the read switching network can conduct. The possible number of combinations is $2^3(8)$, or one combination for each driver.

Conduction of driver 0 cuts off switch M_{r0} and the seven other switches indicated as gated by the driver. Switches M_{r1} through M_{r7} and the remaining switches continue to conduct with their associated drivers cut off. The signal from the read head of channel 0 passes through to the summing amplifier and the emitter-follower, which couples the signal to main memory read amplifier M_a. Signals from the read heads of channels 10, 20, 30, 40, 50, 60, and 70 are also simultaneously coupled to their associated read amplifiers in a like manner from the other circuit boards. The 56 other read head signals are shorted to ground by the conduction of their respective switches.

Outputs of read amplifier M_a and the other seven memory read amplifiers are applied to "and" gates whose other inputs are combinations of signals from channel selector flip-flops C_4, C_5, and C_6. Again, only one of the possible eight signal combinations can be true at any given time, and only one of the "and" gates is enabled to pass the signal from the addressed channel to main memory read flip-flop M_r.

Transistor switches Q1 through Q8 are in parallel with corresponding read heads in the computer memory (the schematic of the read switching network board is given in the Service Manual). Driver Q11 controls the conduction of switch Q8 and one switch on each of the seven other network circuit boards. Summing amplifier Q9 amplifies the selected read head signal current, and emitter-follower Q10 applies the output signal to the associated read amplifier.

Zener diode CR4 in the base circuit of Q11 is the threshold diode for an "and" gate comprising diodes CR1, CR2, and CR3. A -0.5-volt potential exists at point "a" on figure 6-24 when the "and" gate is inhibited by false signals from channel selector flip-flops C_1, C_2, and C_3. CR4 is reverse-biased and Q11 is cut off by the +6-volt supply across R39 at point "b". With Q11 cut off, the negative voltage at point "c" is applied to the gate line which supplies the base drive for switch Q8 and the other seven associated switches. The eight transistors conduct heavily and short the read head signals to ground.

When the "and" gate is enabled by true signals at the inputs, CR4 breaks down from the -12-volt potential at point "a", and the diode maintains a 7-volt drop across its terminals. The potential at point "b" decreases to +0.65 volts, and Q11 conducts when the base potential falls below that of the +0.75-volt emitter potential. Saturation of the driver establishes a positive potential at point "c" which cuts off Q8 and the other seven associated switches. The signals from the corresponding read heads pass through to the summing amplifier while the remaining switches short out the 56 other signals as described previously.

A potentiometer at the output of each transistor switch permits adjustment of the output signal amplitude. The grounded-base configuration of summing

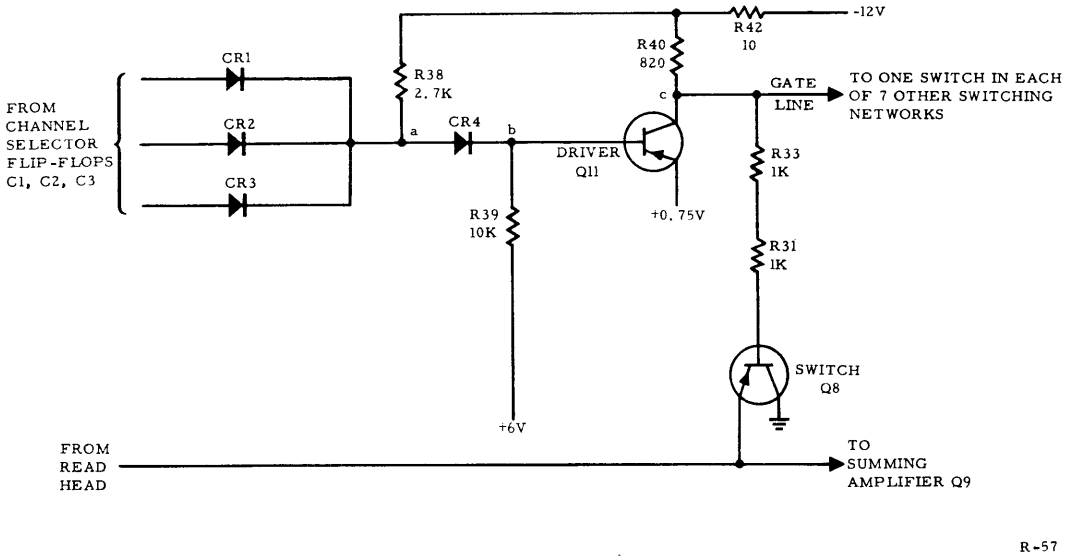


Figure 6-24. Simplified Schematic For One Read Switch

amplifier Q9 provides an input resistance of approximately 50 ohms to the read head signal. This value of resistance is low in comparison to the series resistance of 10 kilohms, and amplitude adjustments of the read head signal are independent of each other. The series 10-kilohm resistor provides damping to prevent signal overshoot.

Read Amplifier Circuits

The computer contains 17 identical read amplifier circuit boards which amplify and shape the read head signals before applying them to the logic gates. The 17 read amplifier circuit boards are utilized as follows: eight for main memory information channels, one for each of the five registers, one for each of the two rapid-access loops, one for the clock channel, and one for the origin-sector channel. Input signals for the information channel amplifiers are derived from the read switching networks; the input signals to the remaining amplifiers are received directly from the read heads.

Figure 6-25 is a block diagram of a read amplifier circuit. The pre-amplifier amplifies the weak read head signals and applies them to a strobe switch (a modified Schmitt trigger) circuit. This switch synchronizes memory reading with clock timing to ensure that information transfer occurs simultaneous with other computer logic operations. The flip-flop power amplifier shapes the read head signal and amplifies the current sufficiently to drive the logic gates.

A 1-bit memory read signal input to the read amplifier circuit during a false clock interval produces a true signal at the unprimed output and a false signal at the primed output. When a 0-bit signal is input to the read amplifier circuit, the output signals are reversed. The read amplifier is reset at the end of each word sector time by the positive transition of a pulse from memory synchronization flip-flop T_{41}.

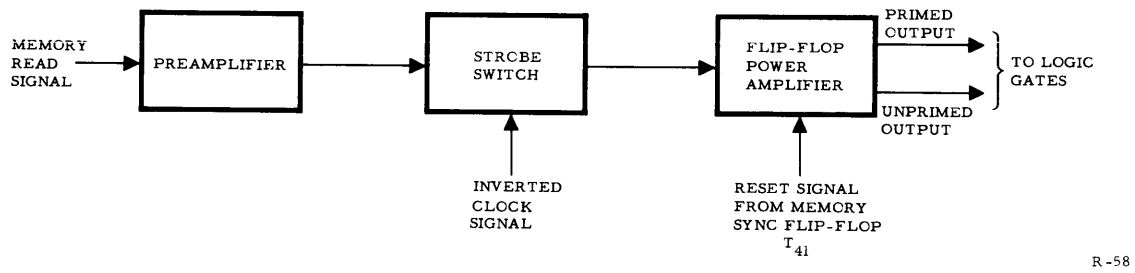


Figure 6-25. Read Amplifier Block Diagram

Each read amplifier circuit's preamplifier consists of five transistors connected in alternate emitter-follower and amplifier configurations (refer to schematic of read amplifier board in Service Manual). The memory read signals are routed to emitter-follower Q1 which presents a low-capacity input to the memory read heads. Amplifier Q2 has a gate control (potentiometer R5) in the emitter circuit. The output signal at the collector of Q2 passes through a resistance-capacitance network composed of R3 and C3 which limits the high frequency response to 120 kilocycles. Noise decoupling networks are used throughout the preamplifier. Amplitude of the preamplifier output signal at TP5 is approximately six volts peak-to-peak.

The preamplifier output signal is impedance-coupled through C8 and L1 to the base of strobe switch transistor Q6. Inductor L1 provides low d-c resistance to improve the d-c stability of the transistor bias. The strobe switch transistors are n-p-n type, and the strobe signal is applied across R17 to the emitters. The strobe signal alternates between +6 to -12 volts.

Operation of the strobe switch circuit must be considered in relation to that of the power amplifier flip-flop. The reset pulse from memory synchronization flip-flop T_{41} is applied to the read amplifier circuit through CR4. Then the signal is applied through CR7 to the logic gates, and capacitor C11 couples the signal to the base of the flip-flop transistor Q8. Positive transition of the pulse cuts Q8 off, and capacitor C13 couples the negative increase in collector voltage to the base of Q9, causing that transistor to conduct. Collector voltage of Q9 rises to -0.5-volt and that of Q8 increases negatively to -9 volts. Therefore, at reset the read amplifier circuit unprimed output is false and the primed output is true.

The strobe switch circuit contains a feedback line from the base of Q7 through CR6 to Q9 collector. At reset, current through this feedback line establishes a potential of -5.4 volts on the base of Q7. The bias on Q6 is -6 volts applied across inductor L1, and the resultant -0.6 threshold bias at the base of Q6 prohibits the triggering of the input transistor by transient noise voltages.

Figure 6-26 shows the read amplifier waveforms and voltage levels. Assume that the power amplifier flip-flop has just been triggered by the reset pulse resulting in the unprimed output false and the primed output true. When reset, both strobe switch transistors are cut off because of the +6-volt strobe

signal applied at the emitters and the approximate +6-volt potential at the collectors. At the negative transition of the strobe signal, Q7 conducts when the emitter voltage decreases to the base potential of -5.4 volts. The conduction of Q7 clamps the emitter voltage of both transistors at a -5.4 volt level, and Q6 remains cut off with the -6-volt supply present at the base. Negative increase in Q7 collector voltage applied to the base of flip-flop transistor Q9 has no effect as the transistor is conducting at near saturation. The read amplifier unprimed output remains at a false level and the primed output remains at a true level.

A positive 1-bit read head signal from the preamplifier applied to the base of Q6 does not cause that transistor to conduct until the threshold bias of -0.6-volt is overcome. When Q6 becomes positive biased, the transistor conducts and clamps the emitter potentials of both transistors to -6 volts for the duration of the strobe signal true period. With Q6 conducting,the negative increase in collector voltage is applied to the base of Q8, and that transistor conducts and cuts off Q9. The read amplifier unprimed output then produces a true signal and the primed output produces a false signal when a 1-bit read head signal appears at the input. With Q9 cut off, current through the feedback line to the base of Q7 decreases and the base voltage increase to -6.6 volts. Q7 remains cut off for the duration of the strobe signal true period by the -0.6-volt bias between the base and the emitter. Both Q6 and Q7 are cut off during the false period of strobe signal, but the read amplifier outputs remain at the same true and false levels with the conduction of Q8 keeping Q9 cut off.

The positive memory read signal decreases to zero in approximately one clock interval. However, the read amplifier outputs will continue to indicate a 1-bit condition until application of a 0-bit read head signal. Q6 continues to conduct at the negative transition of each strobe signal, but the negative increases in collector voltage applied to the base of the conducting Q8 have no effect on the state of the flip-flop.

A negative 0-bit read head signal applied to the base of Q6 during the strobe signal true interval cuts the transistor off instantaneously, and Q7 conducts when the negative voltage at the emitter exceeds the -6.6-volt potential at the base. Negative increase in Q7 collector voltage is applied to the base of Q9 and that transistor conducts and cuts off Q8. The read amplifier unprimed output then produces a false signal and the primed output produces a true signal when a 0-bit read head signal appears at the input. Feedback current from Q9 collector again establishes the -0.6-volt bias at the base of Q6 which keeps the transistor cut off for the duration of the strobe signal true interval. Q7 continues to conduct during the strobe signal true period, and both transistors are cut off during the strobe signal false period.

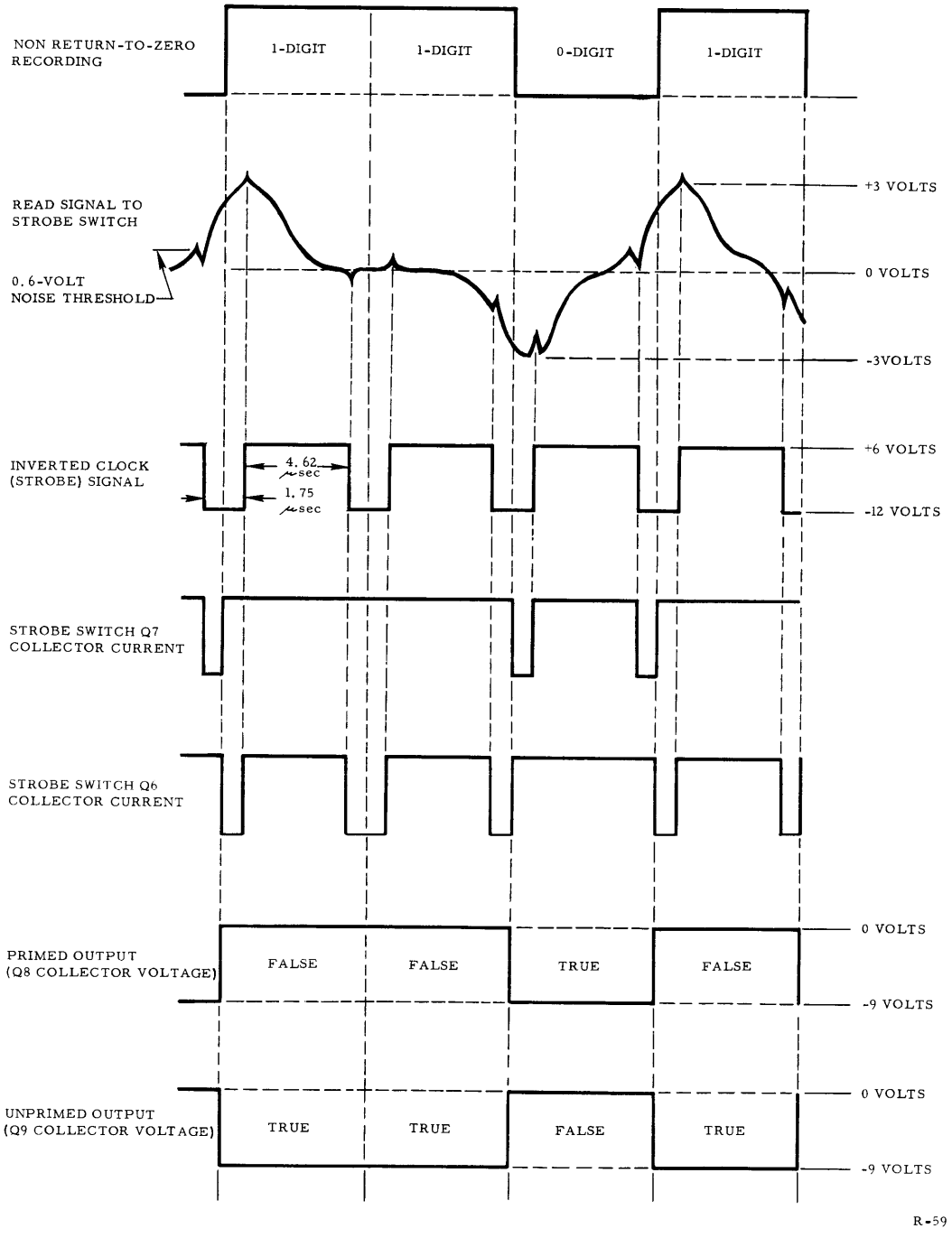


Figure 6-26. Read Amplifier Waveforms

Memory

Basically, a RECOMP computer memory consists of a motor-driven, ferrous-oxide coated disk which rotates at a synchronous speed beneath a stationary headplate containing write, read, and erase heads. The coated surface of the disk stores information in the form of flux patterns representing binary digits. These patterns are placed on the disk in the form of concentric channels or tracks which serve several purposes. Most of the channels constitute general storage for commands and data, and are referred to as main memory. Remaining channels provide a rapid-access portion for commands and data, portions of the arithmetic and control registers, and basic computer timing and control signals.

Write heads electrically record the flux patterns representing the binary digits, and read heads electrically reproduce the patterns. Erase heads for the rapid-access loop and register channel erase all information in this channel immediately after the information is reproduced by the read heads. Erasure is accomplished by magnetically changing the recorded flux patterns on the disk to one polarity.

Mechanical Construction

Figure 6-27 illustrates the mechanical construction of the computer memory. A flexible diaphragm attaches the magnetic disk to the rotor, which is mounted to a non-rotating shaft with ball bearings held in place by snap rings. The shaft is supported by two parallel diaphragms, one located in the inner edge of the headplate and one in the lower part of the frame. These diaphragms provide stiff coupling between the members but are flexible enough to permit disk rotation parallel to the headplate.

A solenoid assembly, mounted to the memory frame, surrounds the non-rotating shaft. The solenoid plate (armature) is attached to one end of the non-rotating shaft. A pre-loaded compression spring mounted between the plate and the solenoid housing keeps the solenoid plate and, consequently, the disk retracted while the solenoid coil is de-energized. The spring pressure on the solenoid plate prevents the non-rotating shaft from positioning the memory disk near the headplate until the solenoid coil is energized. Power control circuit interlocks ensure that information cannot be recorded or read until the disk is rotating at 3450 revolutions per minute and is positioned approximately 50 microinches from the headplate.

Disk Positioning

The memory disk does not move into normal operating position until 30 seconds after the initial application of power to the computer. This time delay permits the disk to attain the correct rotation speed necessary and to ensure sufficient air bearing between the disk and headplate. (refer to Control Power Circuits in this section).

Memory motor rotation begins with the initial application of power to the computer. The motor, a split-phase induction-type incorporating a capacitance start, has a nominal operating speed of 3450 revolutions per minute. When the disk reaches operating speed, current energizes the solenoid coil after the

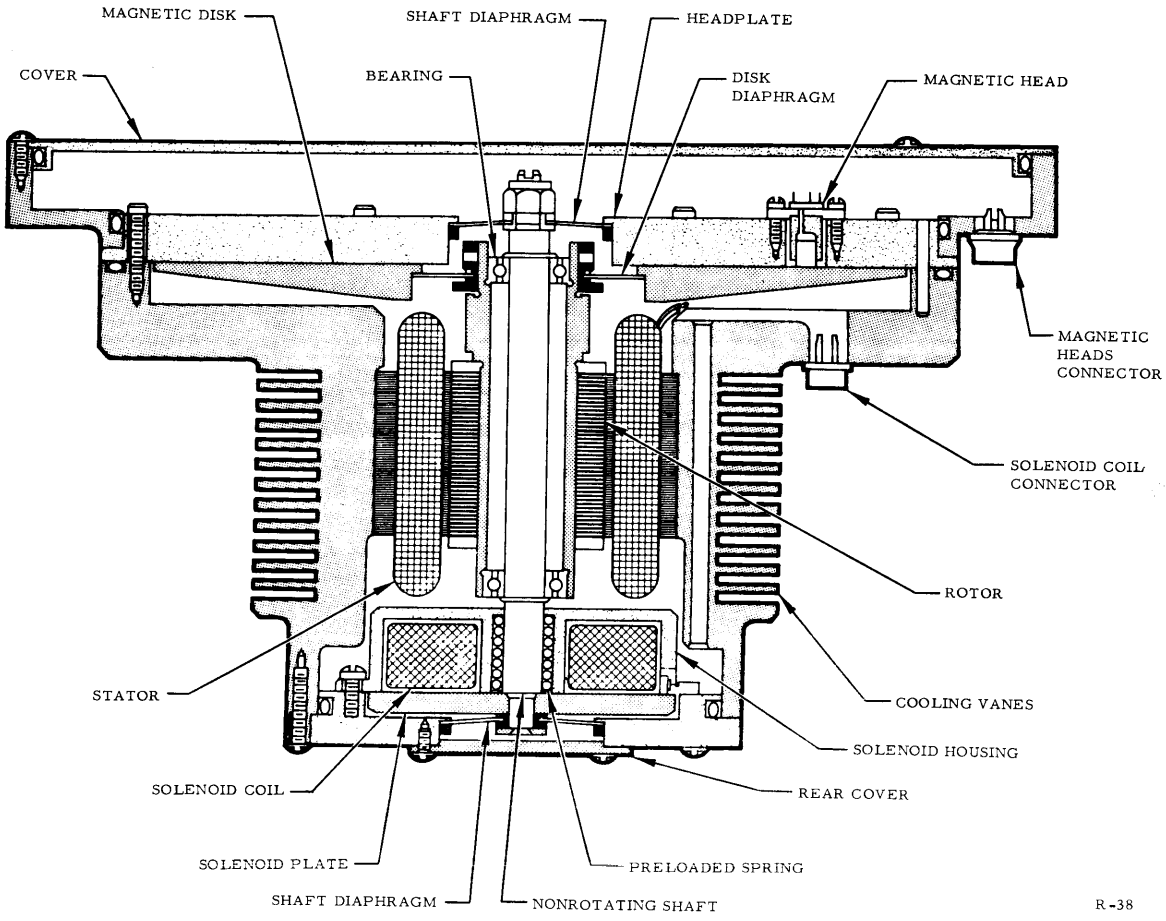


Figure 6-27. Magnetic Memory Cross-Sectional Diagram.

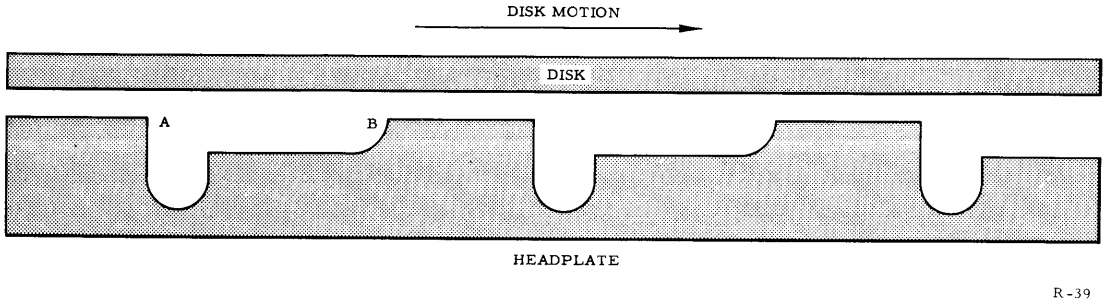


Figure 6-28. Air Thrust Bearing Functional Diagram.

time delay, and the solenoid plate forces the non-rotating shaft, rotor, and disk into correct position near the headplate. After the additional 10-second time delay for synchronization, the computer is ready for operation.

An air thrust bearing (figure 6-28) establishes the gap between the rotating disk and the headplate. The air bearing is developed by eight grooved radial sections in the headplate, each occupying 45 degrees on the headplate surface. Air moves in the direction of disk motion as the disk rotates, expanding in area A and compressing in area B. Any tendency of the disk to move toward the headplate is resisted by the greater air pressure at the high compression points. Thus, the correct gap is maintained by the air cushion produced by opposing forces of air pressure developed at the headplate versus solenoid thrust through the non-rotating shaft.

Information Storage

The channels on the disk surface are separated from each other by 0.03 inch, except for the two inside channels which are separated by 0.02 inch. Radius of each channel from the center of the disk varies from 1.89 inches for the inside channel to 3.87 inches for the outside channel. Main memory utilizes 64 channels, and the registers and rapid-access loops occupy one channel. The disk also contains a clock channel and an origin-sector channel in which timing and control information is recorded at the time of manufacture.

Each of the channels except the loop-register channel provides non-volatile information storage. The non-volatile channels retain stored information for an indefinite period, and the reproduction of information from these channels does not alter the character of the contents. Contents of the main memory channels are subject to erasure or modification to meet the requirements of the programmer, but the contents of the clock and origin-sector channels can be modified only by the manufacturer.

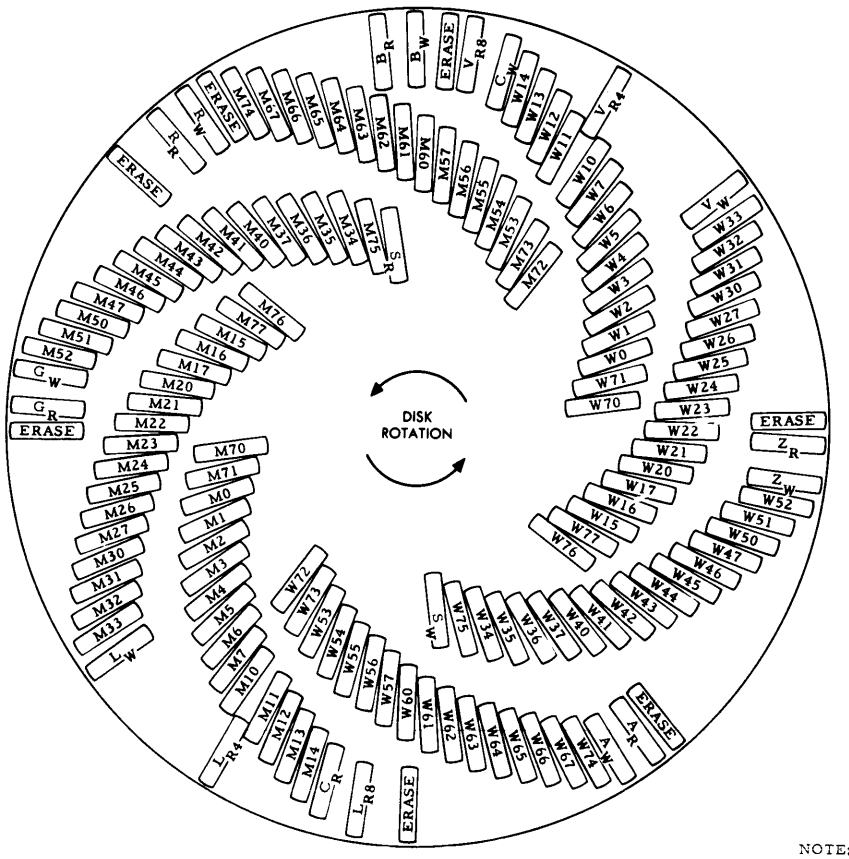
The loop-register channel provides only volatile information storage, as the loop-register functions utilize logic flip-flops in conjunction with the channel on the disk. Information is not retained in the loop-register channel if source power is interrupted or fails and when the computer power is turned off. The loops and registers and their associated flip-flops constitute a closed transmission loop for certain computer commands, and the loop contents may be recirculated during operation to preserve the information for future use.

Each main memory channel has corresponding write and read heads which are located on the stationary memory headplate. The loop-register channel has individual write and read heads for each of the two rapid-access loops and the five registers. The loops and registers also have erase heads positioned such that they erase recorded digits after reading. Figure 6-29 illustrates the head positions on the memory disk in relation to disk rotation.

The write and read heads function in association with write and read amplifiers in recording and reproducing information. All amplifiers are mounted on circuit boards located in the computer chassis.

Main memory channels utilize two write and eight read amplifiers, whereas each loop and register has its own individual write and read amplifier. The clock and origin-sector read heads each have an associated read amplifier,

but no write amplifiers are provided for these channels as information is recorded in them only by the manufacturer.



NOTE: LR_4 AND VR_4 NOT USED IN RECOMP III

MAIN MEMORY = CHANNELS NUMBERED OCTALLY 0 TO 77
HIGH-SPEED MEMORY = LOOPS L AND V
REGISTERS = A, B, R, G, Z
CLOCK CHANNEL = C
ORIGIN-SECTOR CHANNEL = S

M = MAIN MEMORY READ HEAD
W = MAIN MEMORY WRITE HEAD
SUBSCRIPT R = READ HEAD
SUBSCRIPT W = WRITE HEAD
ERASE = ERASE HEAD

R3-145

Figure 6-29. Magnetic Memory Head Positions Diagram.

Write Heads

A write head (figure 6-30) consists of a center-tapped coil wound around a metal core formed of thin, high-permeability T-laminations, two low-loss magnetic blocks, and two ceramic insulating inserts. The lamination edges form the pole tip which serves as the writing medium.

Both the main memory and loop-register write heads record information in word sectors of their respective channels as selected by the computer program. However, the clock and origin-sector write heads are utilized only at the time of the manufacture and perform no functions in normal computer operations.

Figure 6-31 illustrates the principles of perpendicular recording utilized in the computer memory. Write current flows through the coil and sets up flux lines around the core. The flux concentrates at the small area

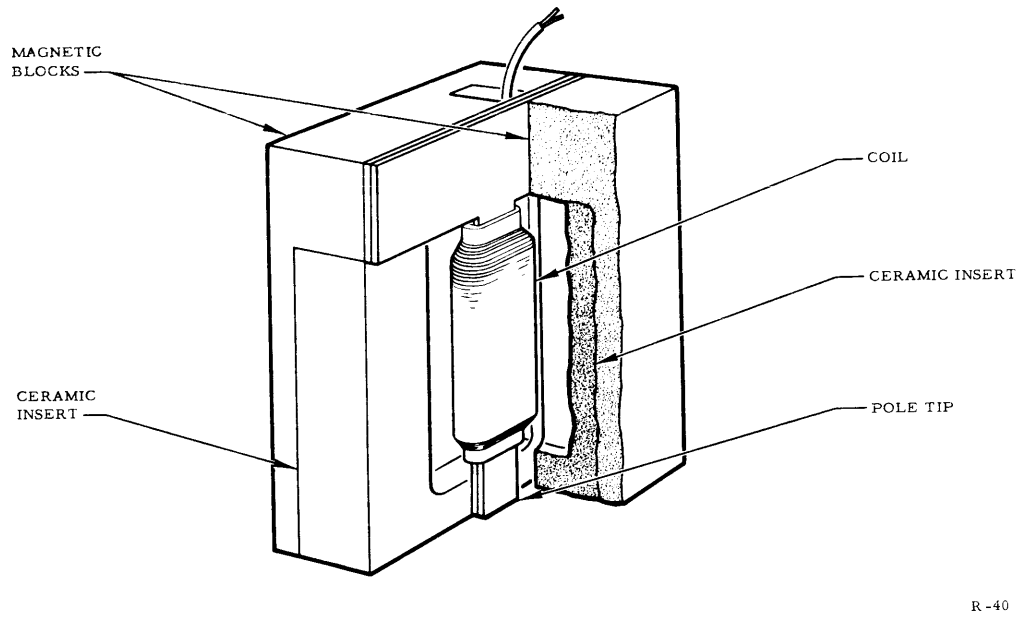


Figure 6-30. Write Head Construction Diagram.

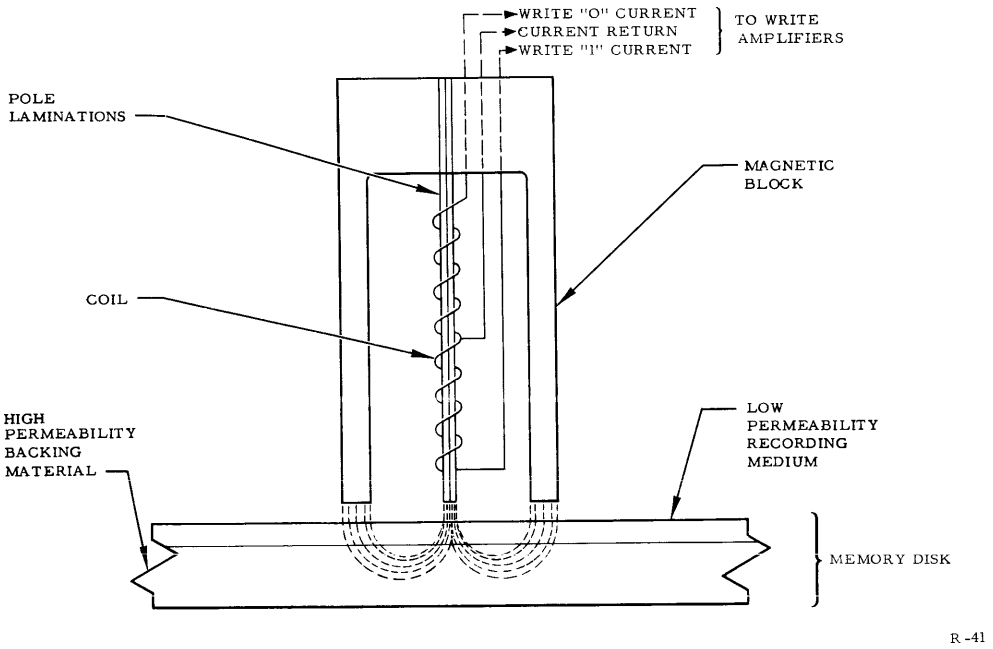


Figure 6-31. Perpendicular Recording Diagram.

of the pole tip and sets up vertical magnets as it flows downward through the ferrous-oxide recording medium on the memory disk. Low permeability of the recording medium, as opposed to high permeability of the backing material, tends to keep the flux concentrated as it passes through the recording medium. In the backing material the flux diffuses to a lower density and has little effect upon the magnets already produced as it re-enters the recording medium. The flux return path after leaving the memory disk is through the magnetic blocks to the center section of the T-laminations.

Relationship between the write current and the induced flux pattern in the recording of binary information is illustrated in figure 6-32. Square-wave pulses of write current flow through the coil from opposite directions to the center tap for binary-1 and binary-0 signals. This method of recording is termed "non-return to zero" because the write current is always at either a positive or negative value and the voltage level changes only at the transition between recording of unlike digits.

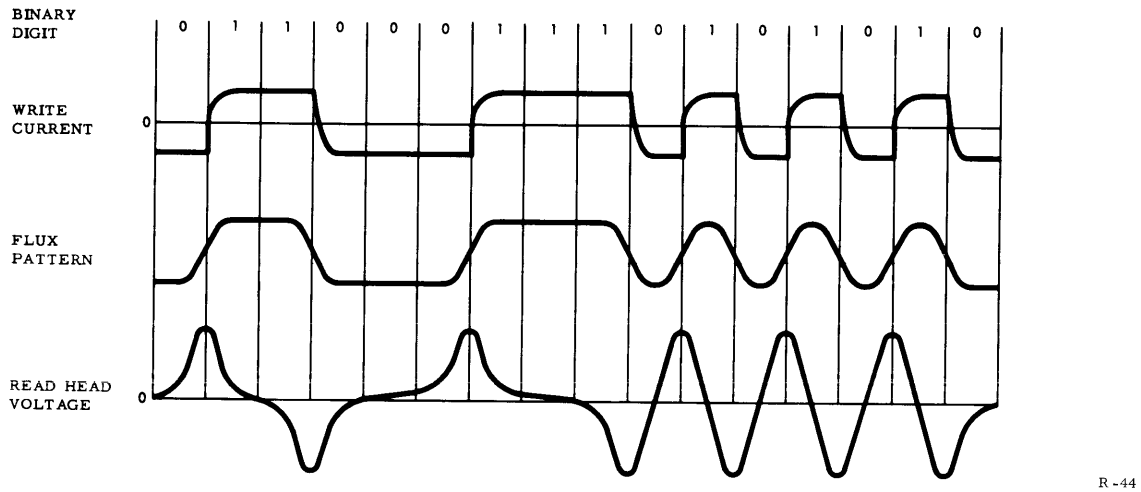


Figure 6-32. Writing and Reading Waveforms.

Read Heads

A read head (figure 6-33) consists of a single-ended coil wound around a metal core formed of T-laminations, and two low-loss magnetic blocks shaped to form a shielding extension around the pole tip.

Both the main memory and loop-register read heads reproduce information from word sectors of their respective channels as selected by the computer program. The clock and origin-sector read heads continuously monitor their channels and transmit the recorded timing and control information to various computer circuits as required.

Figure 6-34 illustrates the principles of the reading of perpendicularly recorded digits from the memory disk. The read head senses only the changes in the magnetized state of the recording medium which induces essentially triangular voltage pulses in the coil. A positive pulse represents a change from a binary 0 to a binary 1, and a negative pulse represents a change from a binary 1 to a binary 0 (figure 6-32). For a succession of similar binary

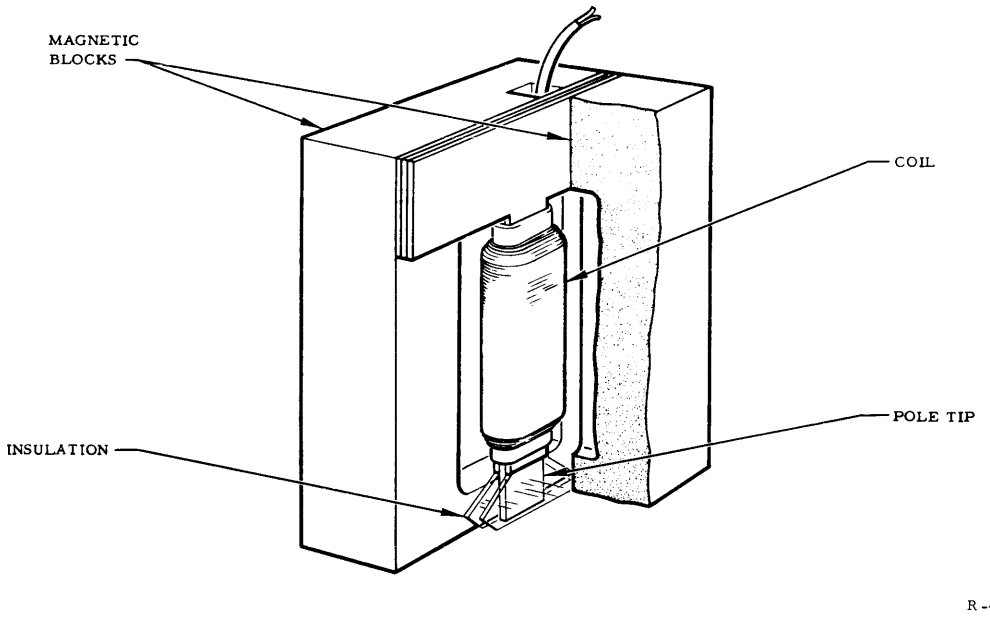


Figure 6-33. Read Head Construction Diagram.

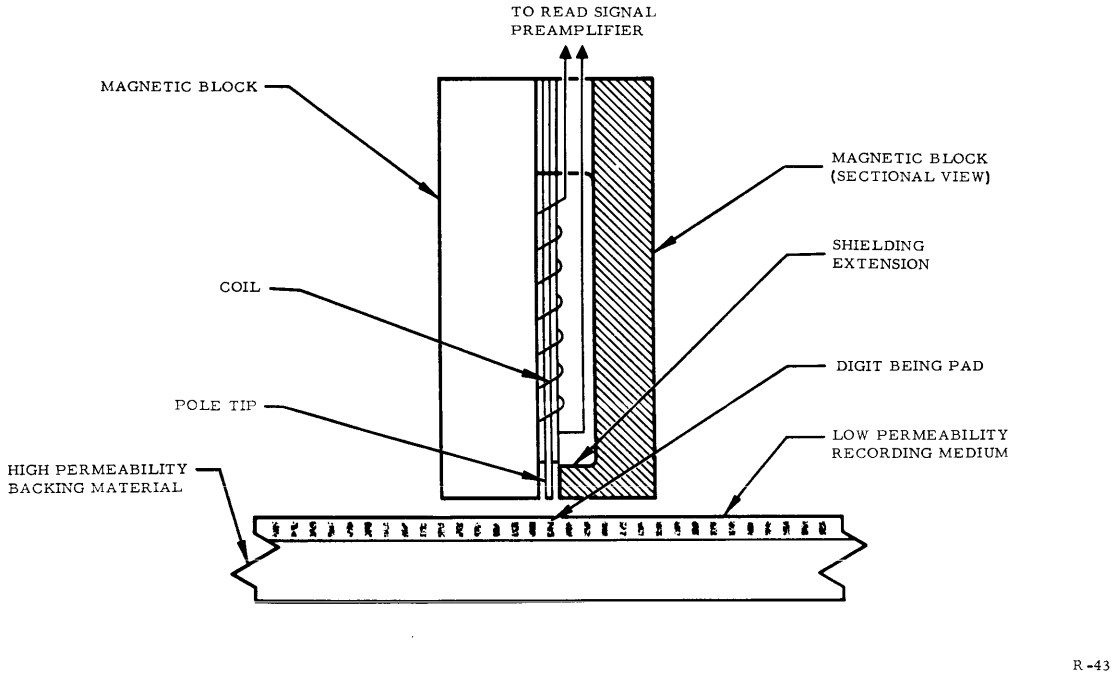


Figure 6-34. Perpendicular Reading Diagram.

digits (bits), the first bit is reproduced as a pulse and the succeeding bits have a 0-volt level. The magnetic block extensions around the pole tip shield the coil from adjacent recorded channels to ensure that only the information in the channel addressed will be reproduced.

Erase Heads

All seven loop-register erase heads are similar in construction to the write and read heads, but a permanently-magnetized wire replaces the laminated core. The erase head magnetic field changes previously recorded flux patterns on the memory disk to the same polarity. Each recorded bit is automatically erased after the read head reproduces the bit and stores it in a read amplifier.

Main Memory Storage Arrangement

Each of the 64 main memory storage channels is divided into 64 memory locations or sectors, each with a storage capacity of one word composed of 41 bits (one sync bit and 40 information bits). The storage capacity of each channel is thus 64 words, or 2624 bits. Each 40-bit information word can represent a pair of commands, a number, or five alphanumeric characters. Each bit occupies a space in a channel equal to one clock pulse time; each word, a space equal to 41 clock pulse times. The bit times thus are established by the clock pulses generated by the sine wave recorded in the clock channel (refer to Timing Control in this section).

Both the main memory channels and the word sectors within each channel are designated by the octal numbers 00 through 77. Each memory location thus has an "address" composed of four octal digits with the first two specifying the channel and the last two the sector within the channel. The computer records one word in (or reads one word from) a designated word sector of a designated channel as directed by commands stored in the memory. These commands specify whether recording or reading is to be performed and contain the addresses of the channel and sector in or from which recording or reading, respectively, is to occur. The bits of a word are processed in serial order in the direction of disk rotation, that is, the least significant information bit is recorded (or read) first (refer to figure 2-3).

Write heads are numbered octally W_0 through W_{77}, and read heads are numbered octally M_0 through M_{77}. The write heads are separated from the read heads by 180 degrees on the memory headplate.

The last 16 word sector addresses of channel address 77 are reserved for designating word locations in the rapid-access loops. Since these sector addresses are not available for specifying word locations in main memory, the addressable storage capacity of main memory is 4080 words.

Timing Control

Timing control signals recorded on the memory disk synchronize the serial operations of the computer. A sine wave in the clock channel estab-

lishes the basic bit times during each cycle of operation, and special bit patterns in the origin-sector channel synchronize the computer after initial application of power. Following initial synchronization, sector pulses synchronize memory writing and reading processes with other computer operations. These basic timing signals operate in conjunction with a binary digit (bit) counter and special timing flip-flops and gates in the control of every computer sequential operation.

The sine wave in the clock channel has a frequency of 2624 cycles for each revolution of the memory disk (figure 6-35). Each cycle defines one bit time in the memory storage format. The sine waves are shaped into square-wave signals in the clock power amplifier circuits, where their time durations are adjustable to provide clock pulses with a 1.75-microsecond true portion and a 4.62-microsecond false portion. These clock pulses provide the timing for the triggering of all logic flip-flops in the computer.

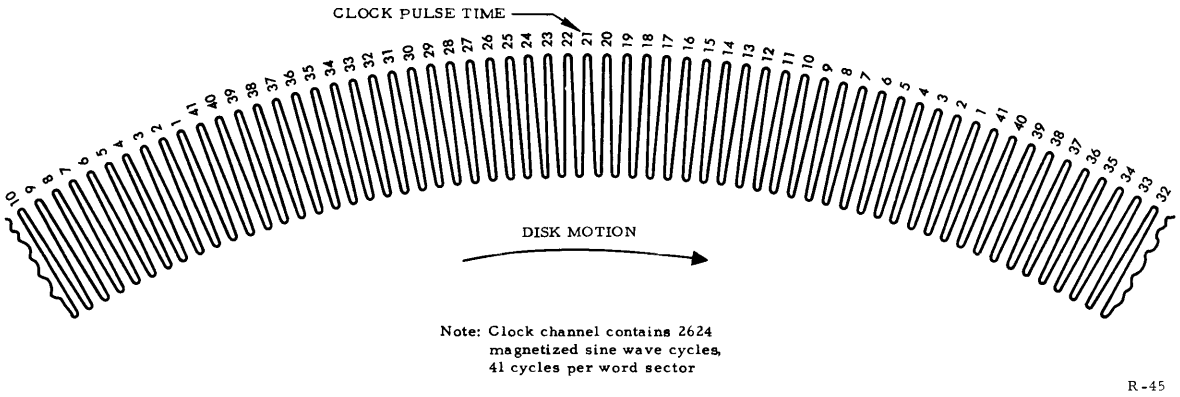


Figure 6-35. Clock Channel Waveforms.

Outputs of the six logic flip-flops comprising the bit counter define the bit times T_1 through T_{41} which constitute one word time. After the initial application of power to the computer, the flip-flops change state at the true-to-false transition of each clock pulse. The clock pulses thus establish the bit times. However, the computer timing requires synchronization of the bit counter output with that of the clock channel to ensure that the counter begins the bit count at time T_1 of the first word in sector 00. The following paragraphs describe how this is accomplished.

Word sectors in the origin-sector channel are numbered octally in the same manner as those of the main memory. Sector 77 of the origin-sector channel contains recorded binary 0's in every bit position in contrast to each of the other sectors which contain at least one binary 1 in some bit position. These binary-1 bits prevent the counter from completing a full cycle until the occurrence of sector 77. The last bit position of channel 76 contains a binary 1 which resets all bit counter flip-flops to zero just prior to the occurrence of sector 77. The binary 0's in sector 77 then have no effect upon the state of the counter flip-flops, and the counter completes a full cycle and 1-sets timing flip-flop T_1 at the first digit time of sector 00 in coincidence with the first clock pulse.

Timing control flip-flops T_1 and T_{41} provide the signals signifying the beginning and end of each word. T_1 is not set to the true state until the bit counter is synchronized with the clock signal; however, T_{41} is initially set to the true state at the beginning of synchronization, then alternately 0-sets itself with each binary 0 and is 1-set with each binary 1 detected in the origin-sector channel. At sector 77, T_{41} remains in the false state, which allows the bit counter to complete its cycle and thus effect synchronization with the clock signal and of the computer. The logic involved in this synchronization is described in section 7 of this manual.

Flip-flops N_1 and N_5 provide additional timing control over various functions as determined by the computer operational modes.

Logic Control

Logic control ensures that all operations specified by the instructions are performed in the correct mode and in proper sequence. Basic control elements are logic flip-flops operating in conjunction with the two control registers and the origin-sector channel. The states of the flip-flops define the logic control functions of mode designation, operation selection, loop and main memory selection, channel selection, and sector selection. Detailed explanations of the logic involved in these control functions are in section 7.

Mode Designation

Three flip-flops comprise the primary elements controlling the computer through the various modes and in the transitions from one mode to another. Two other flip-flops are also utilized as control elements to designate the phases of operations within modes. However, both of these flip-flops also perform other functions on a time sharing basis.

Operation Selection

Six flip-flops define the type of operation to be performed by the computer. Settings of these flip-flops are determined by the bit configuration in the operation code of each command; outputs of the flip-flops then control the operation code gating matrix to define the type of operation specified. Some commands, however, use address bits in addition to those of the operation code in defining the type of operation. Address bits used for this purpose are those of the most significant, next most significant, and half-word indicator positions. The configuration of 1- or 0-bits in these positions results in settings of flip-flops whose outputs are used in logic designating the operation to be performed.

A command indicator bit in a transfer command (bit position 22 or 2) determines whether transfer of control will be to the first (left) or second (right) command of the command pair at the location specified. A flip-flop is 0-set to indicate execution of the left command and 1-set to indicate execution of a right command.

Loop-Main Memory Selection

The process of locating commands and operands in memory requires designation of main memory or loop location. This designation is accomplished by analyzing the address of the command pair or operand desired and setting flip-flops as required.

Designation of main memory or loop for commands begins with copying the address of the next command pair to be executed from the location counter to the Z-register. Simultaneous with this transfer, the bits representing the address are monitored for zeros and ones in strategic positions. A 0-bit in any position 27 through 34 indicates a main memory location and a flip-flop is 0-set for this condition. A 0-bit in position 26 indicates an L-loop location; a 1-bit in this position, a V-loop location. If a loop location is specified, the flip-flop designating loop location is 0-set for an L-loop address and 1-set for a V-loop address. Setting or resetting of either flip-flop completes the loop-main memory selection process; this action is followed by selection of the word sector in which the command pair or operand is located and read out into the B-register.

The address of an operand, when required, is analyzed for loop or main memory location in a manner similar to that for commands. However, the analysis is made simultaneous with transfer, from the B-register to the Z-register, of the command pair containing the operand. A 0-bit in any position 27 through 34 for left command or 7 through 14 for right command indicates a main memory location. A 0-bit in either position 26 for left command or position 6 for right command indicates, respectively, an L-loop or V-loop location.

Loop-main memory selection for an operand at an indexed address is performed in the same manner as that at an unindexed address. Determination of indexing requirement is made while the command pair is being transferred from memory to the B-register, and is performed by analyzing the sign bit of the right command. A 0-bit in this position (bit position 21) signifies the operand is to be procured from an indexed location; a flip-flop is then set to indicate this requirement.

Channel Selection

The channel address code, held by six flip-flops, controls the gating of the write and read switching networks in the selection of a designated memory channel. After the desired sector has been located during the sector selection phase, outputs of these flip-flops enable the write or read head of a selected channel.

Sector Selection

Sector selection determines the sector location of a designated command or operand in the memory. The Z-register, the origin sector channel, and several associated flip-flops comprise the sector selection elements.

Selection of the desired sector begins when the sector address in the Z-register is serially compared with the recorded sector digits in the origin-sector channel. A flip-flop registers the output of the origin-sector read head as it passes over the origin-sector channel on the memory disk. The sector agreement flip-flop is set to the false state at the beginning of the comparison period and does not change state if sector agreement occurs. When sector agreement occurs, a flip-flop is set to the true state to gate the contents of the selected memory location into the B-register. The comparison procedure is the same for the location of either a command or operand.

Loops and Registers

All of the 8-word loops and 1-word registers of the computer are of the recirculating type; that is, information therein constantly passes through a continuous closed loop formed in part by memory disk bit positions and in part by flip-flops. Most of the storage is comprised of the disk bit positions, with the amount of storage in flip-flops varying between the individual loops and registers. The L- and V-loops and the G- and Z-registers each have only one write amplifier flip-flop and one read amplifier flip-flop. Of the remaining registers, the A-register has six flip-flops, the R-register has four, and the B-register has three (including the write and read amplifier flip-flops). Two additional flip-flops are often used as an integral part of the R-register; however, these independent logic elements only extent the loop-length of the register under certain circumstances.

Figure 6-36 illustrates operation of a typical recirculating loop or register during recirculation. The register shown is the A-register in which flip-flop A_3 is the read amplifier and flip-flop A_{39}, the write amplifier. Operation of the loops and the other registers during recirculation is similar except for the number of flip-flops involved.

Writing and reading processes are similar for all loops and registers. Information to be recorded enters the loop or register at the write amplifier flip-flop, data to be reproduced leaves the loop or register at the read amplifier flip-flop. Reading from a loop or register does not alter its contents.

During recirculation, the write head records each bit of a word in that portion of the channel allocated to the register. As the disk rotates, the read head reproduces each successive bit and the read amplifier flip-flop immediately stores each bit until the occurrence of the next clock pulse. A binary 1 from the disk sets the read amplifier flip-flop to the true state, and the unprimed output produces an output representing a binary 1; a binary 0 from the disk resets the read amplifier flip-flop to the false state, and the primed output produces an output representing a binary 0. The flip-flops circulate each successive bit around the loop as they change state in coincidence with each clock pulse. Thus, for example, a bit stored in A_3 (figure 6-36) produces a like bit at the output of A_2 when the next clock pulse occurs. A_2 does not change state if a succeeding bit is the same as the previous bit, but continues to register the same output. This serial progression of the bits aroung the loop continues until a bit originally copied from the disk is recorded back onto it by the write head.

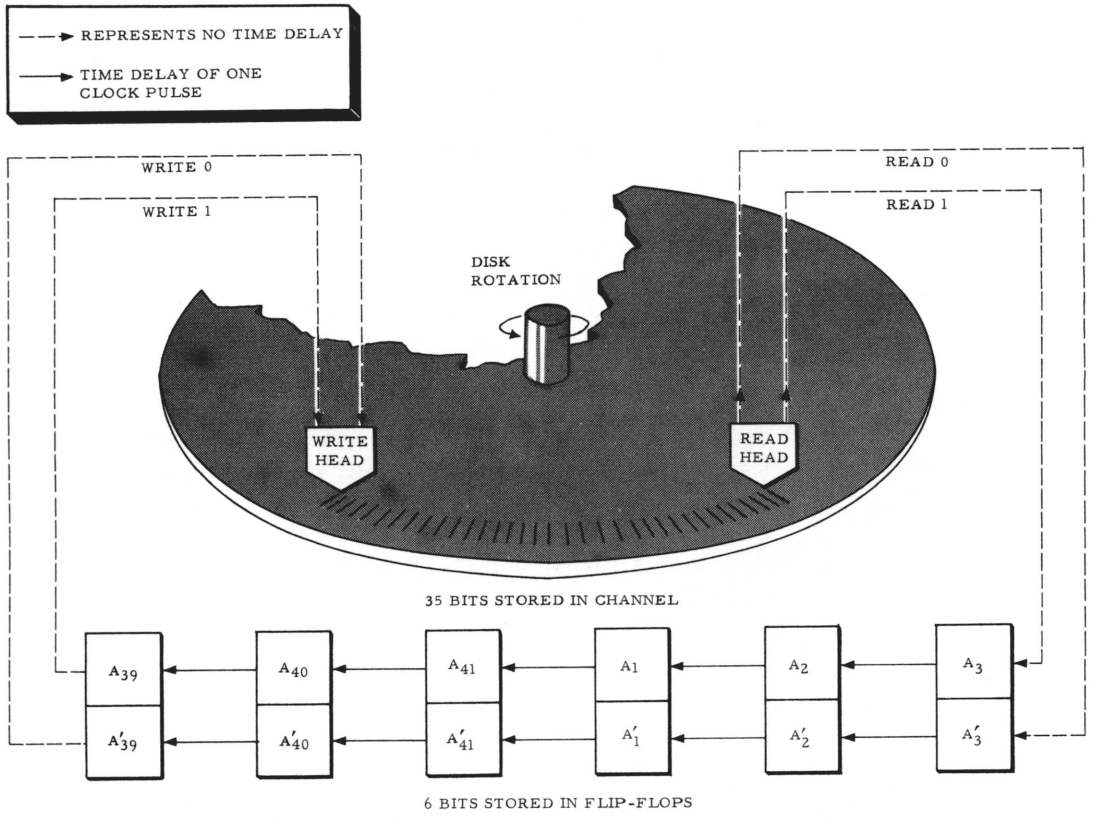


Figure 6-36. Typical Circulating Register (A-Register).

Substitution or revision of the contents also is accomplished in essentially the same manner in all loops and registers. New information, in the form of 0- and 1-bits, is transmitted via a write amplifier flip-flop to the appropriate write head which, in turn, records the new data on the disk. When the disk rotation places the new information under the read head, the new contents are reproduced by the read head and sent to the read amplifier in place of the old data. Recirculation then is resumed in the same manner as previously described.

The processes of writing, reading, and recirculation in the loops and registers are directed by the computer logic as prescribed in the various modes and commands. These processes are described generally in the following paragraphs and in detail in Section 7, Logic Description, in this manual.

Rapid-Access Loops

Each loop has a capacity of eight words and utilizes 8 of the last 16 word sectors of main memory channel 77 as addresses. The L-loop addresses are numbered octally from 7760 through 7767; the V-loop, from 7770 through 7777.

Writing into or reading from both loops can be accomplished directly (without transfer from or to main memory) by several commands when a loop location is specified. Transfers of information can be effected into each of the loops separately from any of the 64 main memory channels. Information can be transferred only from the L-loop, however, to any of the 64 main memory channels. Quantity of words transferred between main memory and a loop can be any number from one through eight, with the exact quantity determined by the least significant octal digit of the main memory address. Transfer of words is always in agreement with ascending memory address sequence.

Each loop consists of a write amplifier, a read amplifier, and 326 consecutive bit positions in the loop-register channel. The bits circulate around the loop as the disk rotates. The L-loop write head is designated L_w and the read head, L_r; the V-loop write head is identified V_w, and the read head V_r. Additional read heads for each loop are located midway between the write and primary read heads, but are not utilized in the RECOMP III computer. The primary write and read heads for each loop are separated by eight word lengths on the memory disk.

Control Registers

The control registers (Z and G) and their associated logic circuits perform many of the control functions required in the location, storage, modification, and execution of commands and in the handling of data. Consisting of a write and read amplifier and 39 bit positions in the loop-register channel, the Z-register stores the command pair being executed; it is not addressable by the programmer. The G-register, also consisting of a write and read amplifier and 39 bit positions in the loop-register channel, contains storage space for the location counter, index register, and character counter (figure 6-37).

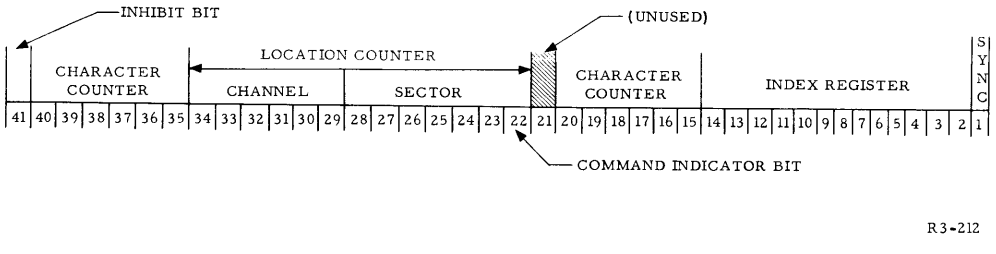


Figure 6-37. G-Register Functional Bit Allotments.

Occupying bit positions 22 through 34 of the G-register, the location counter stores the memory address of the next command to be executed.

These bit positions correspond to the bit positions allocated to the sector and channel addresses of the left command in a command pair (figure 2-3). Thus, the bits in positions 23 through 28 indicate the sector location of the next command pair, and the bits in positions 29 through 34 indicate the channel location of the next command pair. Normally, the computer executes the left command of a pair first, but transfer of control commands permit the execution of the right command of a sequential or non-sequential location first. Commands to be executed are normally stored in consecutive memory locations, and the address in the location counter is increased sequentially by one-half count each time a command is executed. Each full count then indicates the location of the next command pair. The computer thus is automatically directed to execute the commands in sequence until a transfer of control command is indicated. When a transfer of control command is encountered, the location counter is set to the memory location specified in the transfer command. This new location indicates to the computer logic the location of the next command to be executed. The LOCATION COUNTER indicators on the control console display the address location of the next command when the computer is in a non-compute mode.

The index register occupies bit positions 2 through 14 in the G-register. Indexing constitutes the modification of memory addresses by the subtraction of indexing bits in bit positions 3 through 14 from the address bits in the corresponding bit positions of a selected command. A binary 0 in bit position 21 of a command orders indexing, which greatly reduces programming time by permitting a given command to be executed repeatedly using different data each time.

Located in bit positions 35 through 40 and 15 through 20 in the G-register, the character counters tally the quantity of characters input and output and the number of shifts performed on a word. During input or output operations the complements of bits in positions 3 through 8 (for right commands), or 23 through 28 (for left commands), are transferred to bit positions 15 through 20 and 35 through 40, respectively, of the G-register. The appropriate counter then counts the number of characters input or output from 1 to 128. Counting of characters is terminated when the counter contains all binary 1's. During shifting operations, the character counter functions in a similar manner to count the number of shifts indicated. Bit positions 35 through 40 are utilized for left commands and 15 through 20 for right commands. The character counters are also utilized for counting word times during miltiplication and division.

Arithmetic Registers

The arithmetic registers A, B, and R operating in conjunction with various logic circuits perform the basic computer functions of addition, subtraction, multiplication, and division (including those of floating point when the computer incorporates this capability). In the execution of arithmetic commands, the registers operate also with two flip-flops which, during computation, form an adder-subtractor. The three arithmetic registers also perform many semi-arithmetic and control functions. In these functions, however, the arithmetic registers usually operate with logic circuits other than those used in arithmetic operations.

The A-register (accumulator register) stores information in six flip-flops and in 35 bit positions of the loop-register channel. In arithmetic computations, this accumulates all or a portion of the result of each computation. In most other operations, the A-register holds and acts upon the operand.

The B-register (number register) utilizes three flip-flops and 38 bit positions in the loop-register channel. In arithmetic operations the B-register holds the number from memory that will be applied against a number in the A-register.

The R-register (remainder register) stores its contents in four flip-flops and in 37 bit positions in the loop-register channel. At the completion of multiplication, this register contains the least significant half of the product. In the division process, the R-register holds the least significant half of the dividend prior to the division and the remainder of the quotient at completion. (The A-register contains the other portions of the quotient, product, and dividend in these computations.)

Arithmetic Elements and Processes

Principal computer elements involved in performing the arithmetic processes are the three arithmetic registers and the adder-subtractor consisting of sum-difference flip-flop S and computation carry-borrow flip-flop K_a. Other computer elements such as the memory write-read circuitry and the control registers are utilized also; however, these other elements perform associated functions which generally enter into virtually all computer operations.

The registers and adder-subtractor form the principal computation elements for both the fixed and floating point arithmetic processes. However, when the computer is equipped with floating point arithmetic capability, four additional flip-flops and a number of both primary and secondary gates aid in controlling the computation processes.

Nine circuit boards are added to enable floating point arithmetic capability. These are a flip-flop board and eight floating point logic boards. The flip-flop board is the same as the other flip-flop boards utilized in the computer. However, the configurations of the gates on the floating point logic boards differ from the configurations on the other logic boards in the computer and vary from one floating point logic board to another.

The floating point flip-flop circuit board is installed at connector J328 in the computer chassis and contains flip-flops F_a, F_m, E_c, and N_z. The computer utilizes F_a during floating point addition and subtraction functions and F_m during floating point multiplication and division functions. Flip-flop E_c is the comparison flip-flop for floating point addition, subtraction, and division; flip-flop N_z is the normalization flip-flop for floating point addition, subtraction, and multiplication.

Floating point logic circuit boards No. 1 through No. 8 (figures 7-45 through 7-51 of the RECOMP III Service Manual) connect to J325 through J318, respectively, and contain gating elements for the control of the floating point arithmetic functions. Because arrangement of the gating circuits on the individual boards differs, the boards are not interchangeable. Operation of the logic gates on the boards is the same, however, as those described in this section under Logic Circuits.

All computations are performed in the binary numbers system and in serial progression beginning with the least significant bit of each word. The computer performs multiplication and division in variations of the addition and subtraction computations. General procedures used by the computer in the performance of the arithmetic processes are described in the following paragraphs. The procedures of the fixed point arithmetic processes are discussed first; these are followed by analyses of the floating point arithmetic processes. Detailed procedures are presented in Section 7, Logic Description, of this manual.

Fixed Point Arithmetic Processes

In fixed arithmetic, the programmer assumes the binary point (analogous to decimal point) to be between the sign of a number and the most significant digit of the number. Thus, the number is considered to have an absolute value of less than one.

For fixed point addition and subtraction, numbers to be operated upon must have equal exponents prior to computation to obtain meaningful results. This equality can be obtained by either (1) manually scaling the numbers to equal exponents before entering them into the computer or (2) by shifting them, through programming, to equal scaling at some time following their entry.

In fixed point multiplication and division, numbers need not have equal exponents prior to performing computations with them. However, in division, hand-scaling or shifting of the numbers through programming to a lesser exponent may be necessary to ensure against obtaining a quotient equal to or greater than one which causes overflow. Also, in both multiplication and division, shifting of the result through programming, may be desirable to obtain a more easily interpreted result.

Binary Addition

In the execution of an add command, the computer algebraically adds an addend from a selected memory location to an augend in the A-register. Because the computer performs absolute addition of both positive and negative quantities, the procedure employed in executing an addition command depends upon the signs of the operands (numbers to be added). The computer performs absolute addition when the operands have like signs and performs absolute subtraction when the operands have unlike signs (refer to Binary Subtraction).

In executing an add command, the computer first copies the selected word from memory and stores it in the B-register. The contents of the B-register

are then added to that of the A-register, which then stores the sum of the two quantities until it is processed further or placed in memory. Original contents of the selected word are unaltered and remain in the same memory location.

The computer utilizes the same computation methods to perform addition as those employed in manual calculations. First, the computer adds the two least significant digits, one from each operand, records the sum digit, and temporarily stores the carry digit. Then, at the next bit time, the computer adds the carry digit to the sum of the next least significant digit of each operand. The computer continues this serial binary addition, proceeding successively from least significant digits to most significant digits.

Table 6-1 is a truth table for the serial binary adder utilized in the computer. A_1 and B_1 are inputs to the A-register and B-register respectively. K_a is the carry flip-flop, and S is the sum flip-flop. Assume that the augend is initially stored in the A-register and the addend has been transferred from the memory to the B-register. During execution of the addition command, the digits are shifted right so that flip-flops A_1 and B_1 contain corresponding bits at any given time. The digits in A_1, B_1, and the carry digit from the previous bit time in flip-flop K_a determine the sum digit, S, and the new carry digit. The truth table lists the eight possibilities that can occur. Note that the carry flip-flop, K_a, changes from 0 to 1 only when A_1 and B_1 are both true and from 1 to 0 only when A_1 and B_1 are both false. These are the conditions which prevail for the absolute addition of operands which have like signs. When the signs of the operands are unlike, the adder becomes a subtractor as noted previously, and K_a remains in the same state. K_a must be zero at the beginning of each computation cycle.

The adder requires two computation cycles (two word-sector times) to complete the execution of an addition command. The A-register and B-register contain the operands at the beginning of the first cycle. During the first bit time, the computer compares the signs of the A- and B- register, leaves operation code flip-flop D_1 set or resets it to its zero state to indicate whether absolute addition or absolute subtraction respectively, is to be performed, and sets R-register flip-flop R_1 to the true state. K_a had been set to zero at the end of the previous word-sector time. Beginning with the second bit time of the first cycle, flip-flop R_1 gates the operand digits from the A- and B-registers into the adder, starting with the least significant digits. Simultaneously, a zero is gated into the A-register. Also, with each shift right, sum flip-flop S registers the sum of the two digits at the output and a sum digit from the previous bit time is shifted back into the B-register. At the end of bit time 40 of the first computation cycle, the B-register contains the sum of the contents of the A- and B-registers, the A-register contains the original binary sign bit of the augend, the sum flip-flop contains the sign of the addition computation, and the carry flip-flop contains any carry bit resulting from the computation. During bit time 41, the sum flip-flop copies or reverses the A-register sign bit depending upon the absolute values of the operands. If the A-register operand was greater than that of the B-register operand (absolute add), flip-flop S copies the original sign of the A-register operand. If the A-register operand was smaller than that of the B-register operand (absolute subtract), flip-flop

S reverses the original sign of the A-register operand.

During the second computation cycle the computer first transfers the sign of the sum into flip-flop C_2, then places the contents of the B-register (A +B) into the A-register. At the end of the second computation cycle, the sign is copied from flip-flop C_2 into the sign bit position of the A-register. Thus, at the end of the second computation cycle, the A-register contains the sign and the sum of the addition computation.

Table 6-1. Truth Table for Binary Adder.

Inputs To Adder at Time t			Outputs From Adder at Time t+1	
A_1	B_1	K_a	S	K_a
0	0	0	0	0
0	0	I	1	0
0	1	0	1	0
0	1	I	0	1
1	0	0	1	0
1	0	1	0	1
1	I	0	0	1
1	I	I	1	I

Legend: A_1--operand digit from A-register
B_1--operand digit from B-register
K_a--carry digit flip-flop
S---sum digit flip-flop

Binary Subtraction

In executing a subtract command, the computer algebraically subtracts a subtrahend from a selected memory location from a minuend in the A-register. As in binary addition, the procedure employed by the computer to execute a subtraction command depends upon the signs of the operands (numbers to be subtracted). Absolute subtraction is performed when the operands have like signs, absolute addition when the operands have unlike signs (refer to Binary Addition).

The computer first copies the selected word from the memory and stores it in the B-register. Next, the contents of the B-register are subtracted from that of the A-register, which stores the difference sum until it is processed further or placed in memory. Original contents of the selected word remain unaltered in the same memory location.

The same computation methods are used in the computer to perform subtraction as those employed in manual calculations. The computer first performs a subtraction on the two least significant digits, one from each operand, records the difference digit, and temporarily stores the borrow digit. This serial binary subtraction then continues, proceeding successively from least significant digits to most significant digits.

Table 6-2 is a truth table for the serial binary subtractor utilized in the computer. Note that the difference digits are identical in all cases with the sum digits of the binary adder. This means that the sum flip-flop can also be utilized to generate the difference in binary subtraction. K_a which now contains the borrow binary digit, changes from 0 to 1 only when A'_1 and B_1 are true and from 1 to 0 only when A_1 and B'_1 are true. These are the conditions which prevail for the absolute subtraction of operands which have like signs. When the signs of the operands are unlike, the subtractor becomes an adder, and K_a remains in the same state.

Table 6-2. Truth Table for Binary Subtractor.

Inputs to Subtractor at Time t			Outputs From Subtractor at Time t+1	
A_1	B_1	K_a	S	K_a
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Legend: A_1-- operand digit from A-register
B_1-- operand digit from B-register
K_a-- borrow digit flip-flop
S--- difference flip-flop

As with the adder, the subtractor requires two computation cycles to complete execution of a subtraction command and performs the computation in the same serial manner. The state of operational code flip-flop D_1 again determines whether the circuit will function as an adder or a subtractor in the execution of a subtract command. If the operands have like signs, D_1 remains

in the false state to indicate subtraction; if the operands have unlike signs, D_1 is then set to the true state to indicate addition. These conditions are the opposite of those which prevail for an add command.

When performing a subtraction, the logic associated with the arithmetic elements differs in several respects from that involved during an addition. Inputs to the subtractor from the A- and B- registers are a combination of the unprimed and primed outputs of A_1 and B_1, which change the state of K_a rather than a combination of two primed or two unprimed outputs which supply the inputs to K_a when used as an adder. Also, since in subtraction there is no requirement to add the carry digit in each calculation, the B-register loop is shortened by one bit time as a compensation. The computer achieves this compensation by transferring the difference into flip-flop B_{40} rather than into B_{41} as is done in addition. In effect, then, flip-flop S replaces flip-flop B_{41} in the 41 - bit B-register.

Binary Multiplication

In the execution of a multiplication command, the computer algebraically multiplies a multiplicand from a selected memory location by a multiplier in the A-register. The product that results from the computation is composed of 78 binary digits, with the 39 most significant digits of the product appearing in the A-register and the 39 least significant digits in the R-register. Sign of the product appears in the sign bit position in both the A- and R-registers.

The multiplication process consists of successive multiplications of the multiplicand by each digit of the multiplier, a left-shifting of the result of each step by one digit, and the addition of the result of each step to the result of the previous step to form a partial product. Addition of the results of the final step to the previous partial product yields the final product. The adder-subtractor performs the necessary addition in each step of the multiplication computation.

Two computation cycles are required to complete the execution of a multiplication command. The first cycle extends through one word-time and the second cycle extends through 39 word-times. Timing for the second cycle is controlled by the character counter which terminates the computation at the end of the 39thword-time.

During the first computation cycle, the algebraic sign of the product is determined and appropriately stored. The signs of the multiplicand and the multiplier are sensed and the result, which will be the sign of the product, is stored in flip-flop C_2 for subsequent copying into the sign bit position of both the A-register and R-registers at the completion of computation.

During the second computation cycle, flip-flop S in the adder-subtractor generates a new partial product during each word time of the cycle. Each new partial product is the sum of the previous partial product and either the multiplicand or zero. The computer then transfers the partial product back into the A-register and shifts it one digit to the right as required. The

shift is accomplished by transferring the partial product into A_{39} rather than A_{40}, thus decreasing the A-register length by one bit position. Logic used for the sum and carry flip-flops in the adder-subtractor is identical to that used for addition and subtraction.

Binary Division

In executing a division command, the computer algebraically divides a double-length dividend in the A- and R-registers by a divisor from a selected memory location. The R-register holds the 39 least significant digits of the dividend. At the end of the divide process, the 39-bit quotient is in the A-register and the remainder in the R-register. Sign of the quotient appears in the sign bit position of the A-register, the sign of the remainder is the same as the sign of the quotient.

The division process consists of successive subtractions of the divisor from the full and reduced dividend, the addition of any over-borrow to the reduced dividend, and a left-shifting of the reduced dividend to properly position the divisor at each step. The adder-subtractor circuit performs the necessary addition and subtraction in each step of the division computation.

Two computation cycles are required to execute a division command. The first cycle extends through one word-time and the second cycle through 40 word-times. Timing for the second cycle is controlled by the character counter, which terminates the computation at the end of the 40th word-time.

During the first computation cycle, the algebraic sign of the quotient and remainder are determined and stored. The algebraic sign of the quotient is determined by sensing the signs of the dividend and the divisor. Sign of the quotient is then stored in flip-flop C_2 until it is copied into the sign bit position of the A-register at the end of the computation. The sign of the remainder, which is always the same as that of the dividend, is copied immediately into the sign bit position of the R-register where it remains until the computation is completed. Also during the first cycle, the computer performs a subtraction of the contents of the B-register (divisor) from the contents of the A-register (dividend) to test for possibility of overflow (dividend equal to or greater than divisor) and begins the division process. The computer then transfers the difference from sum-difference flip-flop S into the A-register. This action produces a one-digit left shift of the A-register. The left shift results in a reduced dividend and constitutes the first step in the division process. If there was no overflow, the computer then begins the second computation cycle. But if overflow was detected during the initial subtraction, overflow error flip-flop J_0 is 1-set at the end of the first computation cycle to light the OVERFLOW indicator on the control console.

In the second computation cycle, the computer successively subtracts the divisor from a reduced dividend during each word-time of the cycle except the last. The computer transfers the difference-sum back into the A-register at the end of each word-time and shifts the reduced dividend one digit left. At the end of each word time (except the first) the adder-subtractor either continues the subtraction process, or reverts to addition to

combine an over-borrow with the reduced dividend as required. Word-time 40 of the second cycle is reserved to de-complement the remainder when the contents of the A-register are the complement of the R-register with respect to the divisor. This ensures that the quotient precedes the remainder in readout of the result.

Floating Point Arithmetic Processes

In floating point format, a word consists of a sign bit, an 8-bit characteristic, a 31-bit mantissa, and a sync bit (figure 2-3). The characteristic (exponent) is a binary integer with a range of $2^8=256$; the mantissa appears as a binary fraction less than one, which is the same form as fixed point numbers. If the most significant bit to the right of the assumed binary point is a binary 1 (indicating the fraction has a value between 0.5 and 1.0), the mantissa is said to normalized.

As in fixed point addition and subtraction, floating point numbers to be operated upon must have equal exponents before computation begins. However, in the floating point addition and subtraction processes the computer performs the denormalizing and normalizing (scaling) necessary to obtain equal exponents and a properly-scaled result. Likewise, the computer automatically compensates for overflow (dividend equal to or greater than divisor) in floating point division and automatically produces an easily interpreted result in both floating point multiplication and division.

The two operands in a floating point arithmetic computation are initially transferred to the A- and B-registers as in fixed point computations. Result of all floating point computations appears in normalized form in the A-register.

Floating Point Addition and Subtraction

The two operands in the A-and B-registers must have equal exponents before addition or subtraction on the fractional portions can be accomplished. When the exponents are initially equal, standard fixed point addition or subtraction occurs. If the exponents are unequal, the operand with the smaller exponent is denormalized in the A-register until exponent equality is attained. The smaller operand is denormalized by successively shifting the mantissa one digit to the right, inserting zeros in the vacated bit positions, and augmenting the exponent by a value of one at each shift until the two exponents are equal.

Flip-flop E_c is utilized during the first word time of addition and subtraction to compare the numerical value of the two operands. If the B-register operand is smaller in value, it is transferred to the A-register; if the A-register operand has the smaller value, the contents of the two registers are recirculated.

During the second word time, the operation is terminated if the most significant fractional digits of each operand are zeros. Comparison of the exponents also occurs at this time, with E_c as the comparison element. If the exponent in the A-register is not equal to that in the B-register, the A-register operand is denormalized during succeeding word times until the two exponents are equal. Following equalization of the exponents, fixed point addition or

subtraction is performed for two word times, utilizing the adder-subtractor. If the result of the computation in the A-register is not already in normalized form, the mantissa is normalized by shifting it left and then the exponent is decremented by one for each bit shift of the mantissa until exponent normalization occurs. As during denormalization to equalize the exponents, normalization of the result utilizes a varying quantity of word times as determined by the amount of normalization required.

Floating Point Multiplication

In floating point multiplication, the multiplier is initially in the A-register and, during the first word time, the multiplicand is transferred to the B-register and the character counter is set up as a 32-digit counter. During the second word time, the multiplier fraction is transferred to the R-register, the exponents are added utilizing the fixed point adder logic, and the sum of the exponents is transferred to the B-register. At this time the B-register contains the multiplicand fraction and the sum of the exponents, and computation is terminated if either of the factors is zero.

During the following 31 word times, the fractional portions of the operands in the B- and R-registers are multiplied utilizing the fixed point multiplication logic. However, each of the fractions contains only 31 instead of 39 digits. At the end of these 31 word times both the product and the exponent are in the A-register, the exponent having been transferred to it from the B-register during the latter part of the 31st word time. If the result is not already in normalized form, one additional word time (the 34th in the complete process) is required. During this final word time, the fractional portion of the result is normalized by shifting it left and the new exponent is decremented by one through subtraction.

Floating Point Division

In floating point division, the dividend is initially in the A-register and, during the first word time, the divider is transferred to the B-register and the character counter is set up as a 32-digit counter. Also, consideration is given to a possible overflow condition, which will occur if the dividend is equal to or greater than the divisor. Assuming normalized, a 1-bit denormalization of a larger dividend prior to division computations ensures that no overflow occurs.

Comparison flip-flop E_C is utilized to determine the comparative values of the fractional portions of the two operands during the first word time. If the divisor is larger than the dividend, the computer initiates the fixed point divide sequence. However, if the dividend is larger than the divisor, the dividend fraction is denormalized by one bit to make the divisor the larger.

During the second word time, one subtraction is performed on the fractional portions and the difference is transferred to the A-register and precessed left one digit. An overborrow occurs when the divisor is larger than the dividend, and this condition must be corrected to avoid an extra subtraction which would affect the exponent. A 1-bit is added to the A-register to compensate for the denormalization and the overborrow. The two exponents are also subtracted during the second word time, and the difference

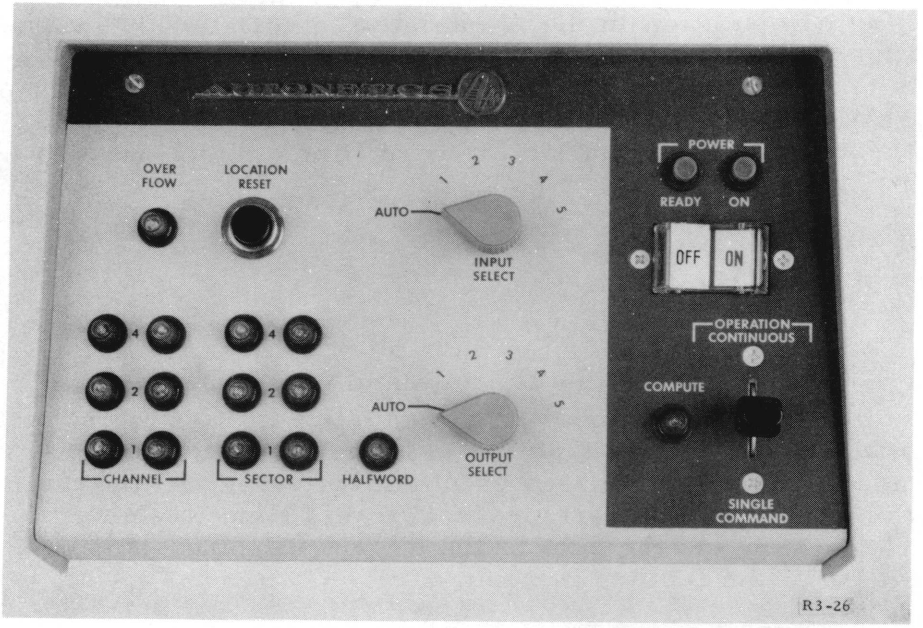


Figure 6-38. Control Console Panel.

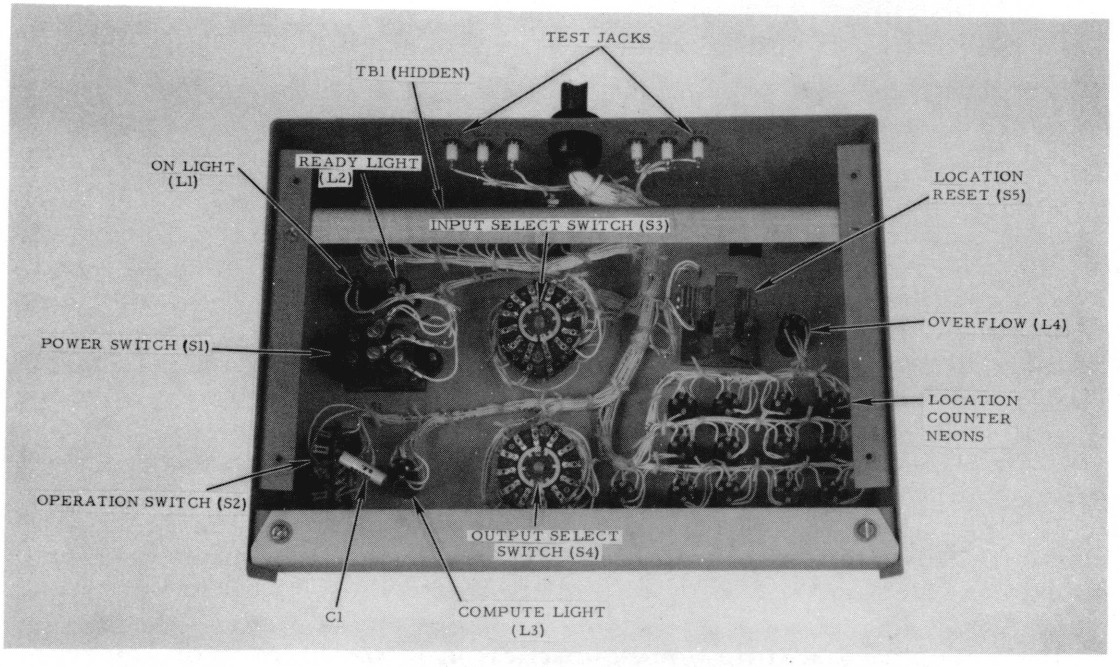


Figure 6-39. Control Console Interior.

is transferred to the B-register.

When dividing a large number in the A-register by a small number in the B-register, an exponent overflow will occur. If the reverse is true, an exponent underflow condition will occur. Flip-flop J_0 is utilized to detect these conditions, but termination does not occur in either case. If the dividend or divisor is a zero, the operation is terminated during the second word time.

After the end of the second word time, the computer enters the standard fixed point divide sequence for 31 word times. The quotient bit obtained at each word time is placed in the R-register which holds each bit until all quotient bits are transferred to the A-register during the word time following the 31st word time of the fixed point sequence. Also during this same final word time (the 34th of the complete process), the new exponent also is transferred to the A-register from the B-register. No normalization is necessary on the result of a floating point division.

CONTROL CONSOLE

The control console provides the principal means for manually directing and visually monitoring operations of the computer system. In addition to the operating controls and indicators, the control console also contains test jacks for external monitoring of various internal signals during maintenance. During input-output operations, various control console elements operate in conjunction with the controls and indicators on a designated device.

Interconnecting wiring diagram of the control console is shown on figure 7-54 of the Service Manual. Connector P901 on the control console cable assembly connects to receptacle J901 in the computer desk well. P901 and J901 have complementary pin numbers with Red-A1 on P901 connecting to Red -A6 on J901. The remaining seven sections of the connector and receptacle have similar connections. Components of the control console are illustrated on figures 6-38 and 6-39; their functions are described in the following paragraphs.

Controls

All controls on the control console, except the POWER ON-OFF switch, receive -12 volts from the computer power supply. Contacts of the control console switches then generate logic, -12-volt, true signals which define the computer operational modes, select peripherial input-output devices, synchronize the logic circuits for location reset, and initiate the bootstrap routine.

POWER ON-OFF Switch

Momentary depression of the ON portion of the ON-OFF switch applies 115-volt, 60-cycle, single phase power to the computer power supply which, in turn, supplies -100-volt d-c power and controls 115-volt a-c power to the computer system elements.

When stopping the computer, depression of the POWER OFF switch for a period of 1/2 second removes power from the computer.

OPERATION Selector Switch

The OPERATION selector switch is a single-pole, triple-throw switch which, through its various positions, determines several general courses of action to be followed by the computer (figure 6-40). Two of the three switch positions are labeled on the control console; these are the CONTINUOUS and SINGLE COMMAND positions. The center switch position, which is not identified on the control console, is the halt position. In the SINGLE COMMAND position, the selector switch acts as a momentary position switch, returning to the halt position automatically upon release.

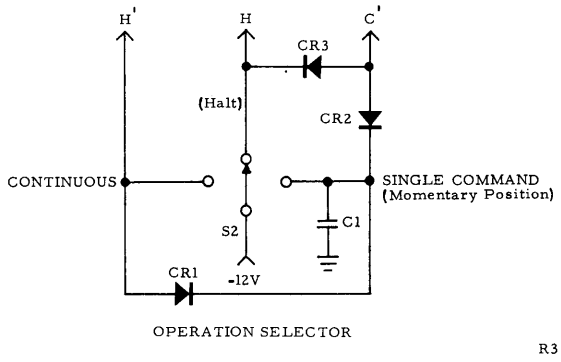


Figure 6-40. Operation Selector Simplified Schematic.

Normally, the OPERATION selector switch is set to the halt position before the initial application of power to the computer. When power is applied, the -12 volts present at the movable switch contact produces halt signal H and causes diode CR3 on TB1 of the control console to conduct and produce single command signal C'. Signal H and other signals cause a 0-bit to be inserted in the sign position of the G-register to enable computation to occur when the OPERATION selector switch is placed at CONTINUOUS or SINGLE COMMAND position. However, signal C' serves no function until computation occurs and then is halted by return of the OPERATION selector switch to halt from CONTINUOUS or SINGLE COMMAND position.

Placing the OPERATION selector switch in CONTINUOUS or SINGLE COMMAND position causes generation of compute signal H'. This signal along with others initiates the compute sequence. When computation is terminated by movement of the switch to halt position, two actions occur: first, H and other signals cause the location counter to be set to zero and second, C' and other signals 1-set the sign position of the G-register which, in turn, causes

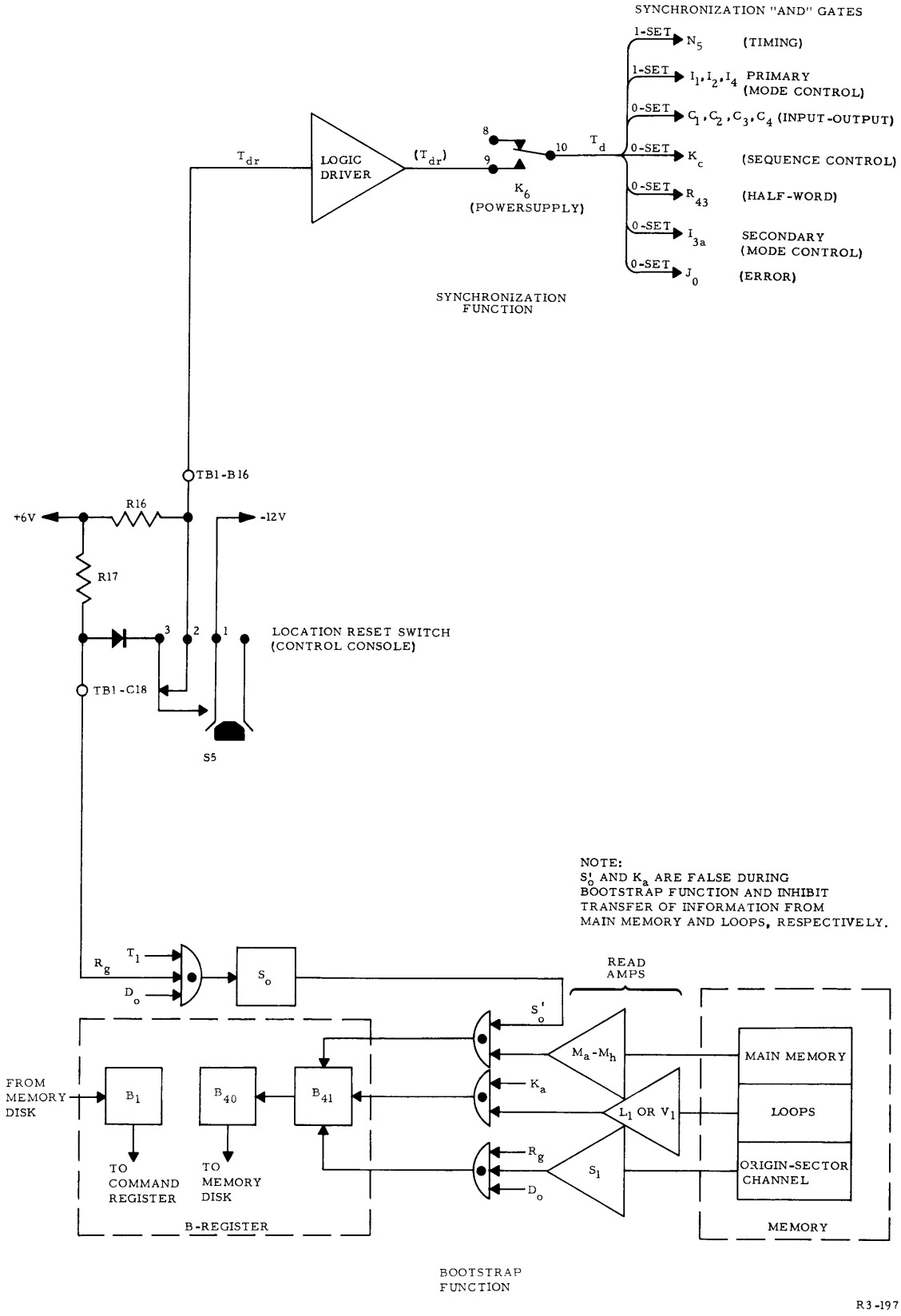


Figure 6-41. Location Reset Switch Functional Diagram.

a 1-bit to be circulated through the G-register to 1-set G_1 at the next T_{41} time. The 1-setting of G_1 then terminates the compute sequence at the end of the current command, and the computer sequences to the idle mode.

In the SINGLE COMMAND position, the OPERATION selector switch generates signals H' and C' simultaneously through the conduction of diodes CR1 and CR2 , respectively, on TB1 of the control console. Signal H' initiates the compute sequence, and C' terminates the sequence after the execution of a single command. Capacitor C1 ensures that single command operation is not affected by switch contacts bounce.

LOCATION RESET Switch

The LOCATION RESET switch is a dual-purpose, pushbutton switch utilized for logic initializing and origin-sector channel readout (figure 6-41). This switch should never be actuated for either function until the OPERATION selector switch has been set to the halt position.

Depression of the LOCATION RESET switch generates origin-sector channel read signal R_g and, during depression and release, momentarily generates logic initializing signal T_{dr} by the application of -12 volts from contact 1 to contacts 3 and 2, respectively. R_g serves no function with the OPERATION selector switch set to the halt position. T_{dr} serves the same function as does delay start signal T_d during computer synchronization.

(T_d results from an open condition at normally-closed contacts 8 and 10 of power supply relay K6 at the initial application of power to the computer. This open condition enables seven single-input "and" gates when the clock signal is received following positioning of the disk. The output signals of the "and" gates maintain the computer in the synchronization mode during the last 10 seconds of the 40-second delay period. Following this 10-second delay, contacts 9 and 10 close to apply a ground to the logic driver; this grounding inhibits the logic initializing gates).

During operation subsequent to computer synchronization, a +6-volts from terminal board TB1 in the control console clamps the logic driver output at ground potential until generation of the T_{dr} signal by depression of the LOCATION RESET switch. The -12-volt T_{dr} signal from the LOCATION RESET switch drives the input transistor of the logic driver into conduction. This conduction supplies a true signal through relay K6 contacts to enable the logic initializing gates (figure 6-41). The computer momentarily enters the synchronization mode and then returns to the idle mode when the LOCATION RESET switch is fully depressed. This momentary initializing occurs again during release of the switch. As the computer enters the synchronization mode, the location counter is set to zero and remains 0-set until the computer leaves the idle mode to begin computation.

The origin-sector channel read function of the LOCATION RESET switch is initiated only after the OPERATION selector switch is set to CONTINUOUS or SINGLE COMMAND from the halt position while the LOCATION RESET switch is held depressed.

As the LOCATION RESET switch generates R_g, signal H' from the OPERATION selector switch initiates the compute sequence. R_g then combines with other signals to 1-set the sector agreement flip-flop S_o and flip-flop B_{41} of the B-register to initiate reading from the origin-sector channel. The two flip-flops remain in the true state as long as the LOCATION RESET switch is held depressed. During this period B_{41} copies the recorded load/start (bootstrap) and diagnostic routines from the origin-sector channel and the B-register write amplifier flip-flop records them into the B-register. The Z-register then copies the B-register read amplifier flip-flop in the usual manner for execution of the commands in the routines. Signals from two flip-flops inhibit the transfer of information from main memory and loops during origin-sector channel readout.

Complete depression of the LOCATION RESET switch must be maintained while the OPERATION selector is returned to the halt position after the completion of the origin-sector channel reading operation. Release of the LOCATION RESET switch again generates signal T_{dr} and the computer momentarily re-enters the synchronization mode before again sequencing to the idle mode.

INPUT SELECT and OUTPUT SELECT Switches

The INPUT SELECT and OUTPUT SELECT switches are identical, six-position, rotary switches which permit automatic or manual selection of input and output devices for operation with the computer. For automatic device selection, coding in the computer program designates the device; for manual device selection, the contacts of the switches generate logic signals which specify the unit to be used. The computer logic assigns a higher priority to a manually-selected device than to a program-selected unit. This priority gives the operator flexibility in choosing the input or output device to be used without changing the computer program.

Any one of five input and five output devices can be manually selected by these two switches. The selected device may be either of two Flexowriter input or output methods, or any one of three optional input or three optional output devices which are compatible with the RECOMP III computer (refer to table 6-3). The AUTO positions of the INPUT and OUTPUT SELECT switches permit device selection by a computer program. Of the five switch positions permitting manual device selection by an operator, positions 1 and 2 select Flexowriter keyboard-printer and paper tape reader-punch, respectively, and positions 3, 4, and 5 select optional input-output devices.

The states of flip-flops C_4, C_3, and C_2 control the selection of the input or output device to be operated by the computer. These three flip-flops are set to the states indicated for each device in table 6-3 by coding a computer program or by manually positioning the INPUT and OUTPUT SELECT switches. For automatic device selection by a program, the flip-flops are set to the indicated states by signals from three coded bit positions in the channel address portion of a computer command. In manual device selection, however, flip-flops are set by three logic signals generated by the INPUT and OUTPUT SELECT switches when set to positions 1 through 5. The flip-flop

states are the same both for input and for output, with selection between input or output device occuring as the computer enters either the input or output mode as directed by the command. Computer signal I_a defines the input mode and signal I_o the output mode. The applicable signal is then combined with the output signals of the three flip-flops to generate a drive signal to the designated input or output device.

Table 6-3. Input-Output Device Selection

SWITCH POSITIONS	DEVICE SELECTED		CONTROL SIGNALS (I and O)	FLIP-FLOP STATES
	INPUT	OUTPUT		
AUTO	AS PROGRAMMED			AS PROGRAMMED
1	KEYBOARD	PRINTER	40, 30, 20	$C_4' C_3' C_2'$
2	TAPE READER	TAPE PUNCH	40, 30, 21	$C_4' C_3' C_2$
3	DEVICE NO. 3		40, 31, 20	$C_4' C_3 C_2'$
4	DEVICE NO. 4		40, 31, 21	$C_4' C_3 C_2$
5	DEVICE NO. 5		41, 30, 20	$C_4 C_3' C_2'$

View a of figure 6-42 illustrates the automatic selection of an input-output device. During analysis of an instruction, flip-flop D_6 copies flip-flop G_1 of the G-register, which causes the three coded bits in channel address positions 32, 31, and 30 (or 12, 11, and 10) to set C_4, C_3, and C_2, respectively, to the states corresponding to the code of a designated device. This action is performed under control of timing signal N_7. Mode control flip-flop I_4 must be in the false state before N_7 becomes a true signal and gates the coded bits into the three flip-flops.

Manual selection of an input-output device is illustrated in view b of figure 6-42. The INPUT and OUTPUT SELECT switches continuously generate three logic signals when set to positions 1 through 5. These signals serve no function until I_a or I_o is generated as the computer enters either the input or the output mode. The three signals from the switch contacts are then combined with I_a or I_o in device selection gates whose output signals set C_4, C_3, and C_2 to the states which select the designated device. Output signals from these flip-flops are then combined with I_a or I_o in a device drive signal gate which, in turn, generates a drive signal to the designated device.

Manual (switch) device-selection signals take precedence over program device-selection signals when the INPUT or OUTPUT SELECT, switch is set

from AUTO to a numbered position. Flip-flop I_4 is always in the true state when the computer is in either the input or the output mode, and the "and" gate which generates timing signal N_7 is inhibited at these times. Because N_7 must be true to control entry of the device designation code into C_4, C_3, and C_2, and because the computer remains in input or output mode until the specified number of characters has been transferred, a program cannot select a device once the computer enters the input or output mode. Automatic device selection can only occur during instruction analysis, which always takes place prior to entry into the input or output mode. Signals from the INPUT or OUTPUT SELECT switch in conjunction with I_a or I_o can, however, change the states of the three flip-flops controlling device selection. Therefore, if a program has been coded to designate a device for an input or output command, automatic selection will occur during instruction analysis; but if an INPUT or OUTPUT SELECT switch is rotated to a numbered position, manual selection will assume and maintain control of device selection as soon as the computer enters an input or output mode.

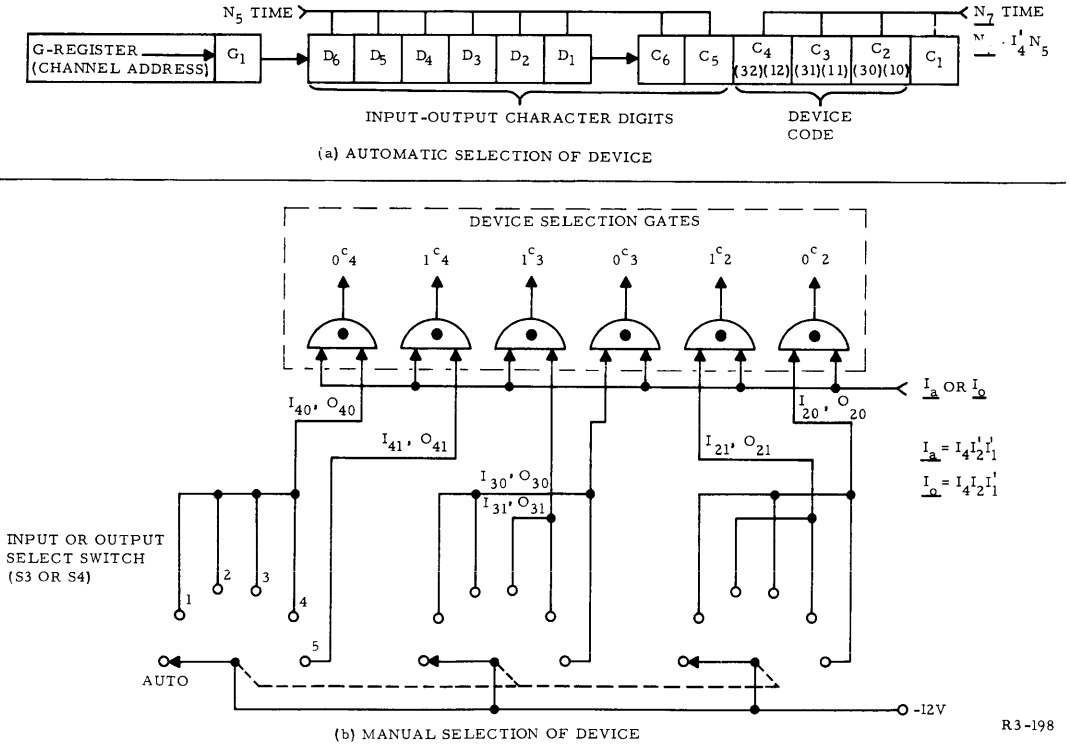


Figure 6-42. Automatic and Manual Selection of Input-Output Devices.

Indicators

The control console contains 17 neon lamp assemblies which provide visual indications of various computer functions while the system is in operation. Except on the POWER ON and READY indicators, each neon lamp has a transistor as an integral part of the lamp assembly. The transistors function as switches to control the illumination of these indicators.

View a of figure 6-43 shows the typical supply and signal voltages at the terminals of a lamp assembly incorporating a transistor switch. The neon lamp is connected between the -100-volt supply and ground, and the lamp is illuminated while the transistor is at cutoff. Collector and emitter of the p-n-p type transistor in the lamp assembly are connected to the -100- and -3-volt power supplies, respectively. The transistor is at cutoff until driven into conduction by the application of a -12-volt signal to the base. When a -12-volt input signal is present, the transistor conducts and causes the -100-volt potential at the lamp to drop below the threshold level required for lamp illumination, thus turning off the indicator. The -100-volt potential returns to its former value when the transistor goes into cutoff with the removal of the -12-volt input signal, and the lamp again illuminates.

POWER ON Indicator

A two-terminal neon lamp, the POWER ON indicator is illuminated by application of 115 volts a-c through contacts of the POWER ON switch as it is depressed. Relay K1 contacts form a holding circuit which maintains the power-on condition and keeps the indicator illuminated through normally closed contacts of the POWER OFF switch after the POWER ON switch is released. The indicator is turned off when the POWER OFF switch is depressed.

READY Indicator

The READY indicator, a two-terminal neon lamp, is illuminated by the application of -100-volt power from the bridge rectifier 40 seconds after the initial application of power to the computer, provided the internal temperatures of the power supply and memory are within operating limits. Depression of the POWER OFF switch causes the indicator to be turned off immediately.

COMPUTE Indicator

The COMPUTE indicator is normally illuminated during the computer operation, being extinguished only during the synchronization and idle modes. During these modes, flip-flops I_1 and I_4 are in the true state, and their output signals are combined in a gate which generates signal $\underline{C_{1o}}$ to inhibit illumination of the COMPUTE indicator.

OVERFLOW Indicator

Illumination of the OVERFLOW indicator occurs whenever an overflow condition exists in the computer. A true signal from the primed output of overflow flip-flop J_o inhibits the illumination of the indicator until the flip-flop is set to the true state by an overflow condition.

Location Counter Display Indicators

The location counter display consists of thirteen neon lamps which,

following a programmed or manual halt, indicate the memory address of the next instruction to be executed. (During computation, the display indicates various information according to the states of the time-shared flip-flops controlling illumination of the lamps. The rapid change of the display prevents interpretation).

Two groups of six lamps each comprise the CHANNEL and SECTOR displays which indicate, in binary form, the channel and sector portions of the address; the 13th or HALFWORD lamp indicates whether the first or second command of the pair at the indicated address is specified for execution. View b of figure 6-43 illustrates the grouping of the display indicators and shows the signals applied to control the illumination of each indicator.

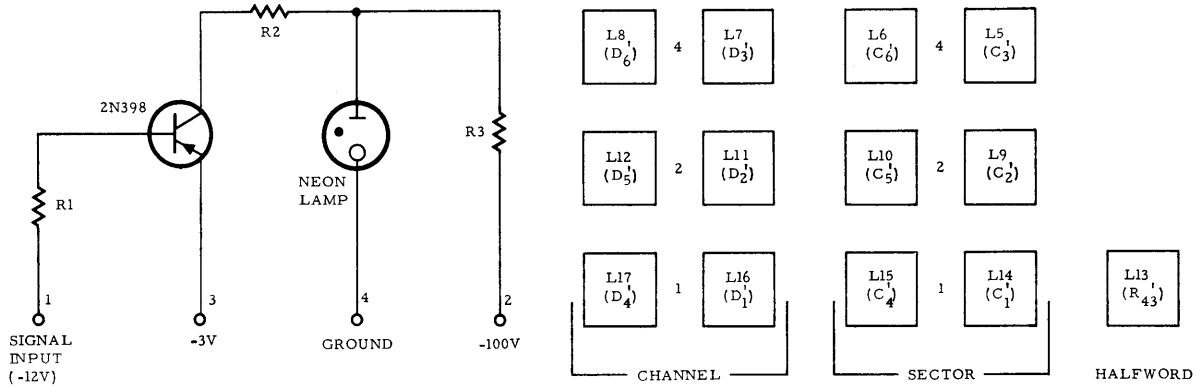


Figure 6-43. Location Counter Display Indicators, Signals and Typical Voltages.

Grouping of the lamps facilitates conversion of a memory address from the 13-bit binary form to the more convenient five-digit octal-binary form. Numbers beside each row of CHANNEL and SECTOR indicators serve as a medium for the address conversion. An illuminated indicator assumes the value indicated by its corresponding number, and an indicator which is not illuminated assumes a zero value. The three assumed values for the indicators in each row are then added to derive one of the four octal digits in each memory address. Binary representation of the HALFWORD indicator is disregarded during conversion because it only designates first half-word when not illuminated and second half-word when illuminated. Assuming that L8, L12, L11, L9, L14, and L13 in figure 6-43 were illuminated, the octal-binary form of the memory address of the next instruction would be 6203.1

During all computer operations except synchronization and idle, signals

from the primed outputs of flip-flops D_6 through D_1, C_6 through C_1, and R_{43} control the illumination of the location counter display indicators. When the controlling flip-flops are in the false state, true signals from the primed outputs of the flip-flops drive the transistor switches into conduction, and the lamp supply voltage drops below the level required for illumination. When the flip-flops are set to the true state, the drive to the transistor switches is cut off and the lamps illuminate. Thus, the indicators alternately illuminate and extinguish in varying display configurations and meanings according to the changing states and uses of these time-shared flip-flops.

When the computer is in the synchronization and idle modes, these same 13 flip-flops likewise are 0- or 1-set according to the prevailing logic. However, mode logic prevents the flip-flop states from being changed. During synchronization, all 13 flip-flops are 0-set, which extinguishes all of the indicators. When the computer is in idle following a halt, the flip-flops are in a 0- or 1-state according to the 0- and 1-bit configuration of the memory address in the location counter prior to the halt. The steady states of the flip-flops, in turn, cause the location counter indicators to constantly display the memory address of the next instruction to be executed. This constant display is established by a special operating sequence referred to as the display mode, which is initiated immediately following a programmed or manual halt and prior to entry into the idle mode.

During the display mode, flip-flop D_6 (view a of figure 6-42) copies the command indicator (halfword) bit, the six sector bits, and the six channel bits of a memory address from the location counter portion of the G-register. The bits are shifted right through the D and C flip-flops, and R_{43} copies the command indicator bit from C_1. These 1 and 0 bits set the 13 flip-flops to corresponding states to enable or inhibit illumination of the location counter display indicators, and the indicators display the memory address of the next instruction as the computer sequences to the idle mode.

Test Points

The rear of the control console contains six test points. Signals from the A-, R-, Z-, and G-registers and timing (sync) flip-flop T_1 are available for external monitoring at the test points (figure 3-2). The 6th test point is ground. Use of these test jacks for troubleshooting procedures and system maintenance is described in sections 2 and 4 of the Service Manual.

INPUT-OUTPUT CIRCUITS AND PROCESSES

The standard RECOMP III computer transmits information to an output device, or receives information from an input device, on up to eight parallel channels. Eight binary digits (bits) transmitted in parallel to or from the computer constitute one character during input-output operations. Each 8-bit character occupies eight bit positions in the alphanumeric word format, and each word contains up to five characters. An input or output command to the standard computer results in the transfer of from 1 to 128 characters between the computer and a selected input-output device.

During input operations the computer can process information contained

in any 5- to 8-channel code. Unused channels are represented in the computer as zeros which can be deleted or modified by programming. During output operations all eight channels are activated by the computer. However, information can be rearranged through programming to any bit configuration or character size required. When provisioned with the extended input-output register, the computer can transmit or receive on a maximum of 12 channels. However, the Flexowriter keyboard and printer can operate on a six-channel code only. During input operations from the keyboard, channels 7 and 8 reflect zeros within the computer. During output operations the printer reproduces only the information contained in the first six channels.

Five-channel tape can be utilized during input operations when the tape is turned over and the TAPE SELECT switch (S1) in the desk well is set to 5. With the sprocket hole in the tape engaged, the switch contacts provide compatibility for the reversed positions of channels 1 and 5 and channels 2 and 4. Channel 3 retains the same relative position, and it is not necessary to route signals on this channel through the TAPE SELECT switch.

Most of the input-output logic and driver circuits which control input and output processes are located on circuit boards mounted on the input-output chassis in the desk compartment.

Input-Output Circuits

In a standard system, the input-output chassis in the desk compartment contains input-output driver circuit boards No. 1 and No. 2 and input-output logic circuit board No. 1 which connect to J5, J6, and J7, respectively. These circuit boards provide amplification and control over the transmission of information between the computer and the Flexowriter (figure 7-1 of Service Manual). Input-output logic circuit board No. 2, which connects to J8, aids in selection and control over transmissions between the computer and optional input-output devices.

The input-output chassis also contains connectors J1 through J4 which interconnect the circuits between the computer, the chassis circuit boards, and the input-output devices.

Schematic diagrams of input-output logic circuit boards No. 1 and No. 2 are shown on figures 7-60 and 7-64, respectively, of the Service Manual. Figure 7-62 in the Service Manual is the schematic diagram for circuit board No. 1 (43995-501) in the etched circuit configuration which is utilized in later computer systems. A different input-output chassis (44126-501) must be installed in the computer when the No. 1 etched circuit board (43995-501) is used.

Input-Output Logic Board No. 1

Input-output logic circuit board No. 1 contains "and" gates for selecting Flexowriter input-output units and for amplifying drive signals for the keyboard-printer, tape punch, and tape reader. Amplification of the drive signals on input-output logic board No. 1 constitutes a reamplification as these signals receive an initial amplification in the computer logic drivers. Input-output logic board No. 1 also contains clamping and isolation circuits for various signals and functions.

Etched circuit board No. 1 (43995-501) contains the same general circuitry as the conventional No. 1 logic circuit board (43955-501) except that different reference voltages are used for the gates; also, input and output terminal numbers on the two boards are different (figures 7-60 and 7-62 of RECOMP III Service Manual). Connector J8 is wired differently in input-output chassis 44126-501 to accept 43995-501 etched logic circuit board No. 1.

Punch-Keyboard Driver Signal Gates

Signal P_{kd} is a driver signal to either the keyboard printer or tape punch when one of these Flexowriter units is selected to receive information from the computer. "And" gate No. 2 generates P_{kd} when the tape punch is selected for output, and driver transistor Q3 conducts to amplify the signal (figure 7-60 of Service Manual). When the keyboard printer is selected for output, "and" gate No. 3 generates P_{kd}, and driver transistor Q5 conducts to amplify the signal. All input signals to "and" gate No. 1 must be true before either gate No. 2 or No. 3 can generate P_{kd}. This is an "and-or" gate combination.

Operation of the two driver circuits is identical. Before the generation of P_{kd}, diodes CR15 and CR17 are reverse biased and Q3 and Q5 are cut off by approximately +6 volts at the base of each. Q4 and Q6 conduct with the collectors at ground potential and +6 volts applied across R16 and R17, respectively, to the base of each. The emitters of Q4 and Q6 are near ground potential, diodes CR20 and CR21 are reverse biased, and the circuit supplies no drive signal to the Flexowriter. This is the state of the circuit when all three "and" gates are inhibited, or when gate No. 1 is enabled and both No. 2 and No. 3 gates are inhibited, or when gate No. 1 alone is inhibited. For the first two conditions, diodes CR11, CR15, CR16, and CR17 are reverse biased, and for the last condition CR11 and CR16 are forward biased and CR15 and CR17 are reverse biased.

Gate No. 1 is enabled when output signal (O_d) from the computer and output ready signal O_r from the Flexowriter are $\overline{\text{true}}$, and the OUTPUT SELECT switch on the control console is set to positions 1 or 2. Gate No. 2 is enabled when signal P (tape punch) is true, and the OUTPUT SELECT switch is set to position 2. Gate No. 3 is enabled when signal K (keyboard printer) is true and the OUTPUT SELECT switch is set to position 1. When gates No. 1 and No. 2 are enabled, CR11, CR16, and CR17 are reverse biased and CR15 is forward biased. Q3 conducts with the base more negative than the emitter, CR20 is forward biased, and Q4 is cut off when the emitter potential goes more negative than that of the collector. CR22 is forward biased, and the driver circuit supplies a P_{kd} drive signal through additional driver amplifiers to the Flexowriter. When gates No. 1 and No. 3 are enabled, CR17 is forward biased, Q5 conducts, CR21 is forward biased, Q5 cuts off, and CR23 is forward biased to supply the P_{kd} drive signal.

Clamping Circuits

Output signal lines from the Flexowriter and the P_{kd} line are clamped to a positive potential until the occurrence of true output signals. The signal lines are connected to +6 volts applied across resistors R24 through R36 (figure 7-60 of Service Manual). Values of the clamping resistors vary according to the current required to drive the logic circuits to which the signals are applied.

The anodes of diodes CR36 and CR37 in the T_n "and" gate are also connected to a +6 volts across 1.1 kilohm resistors to ensure positive diode conduction when either of the two input signals are open.

Isolation Circuits

Signals from flip-flops D_1 through D_6, C_5, and C_6 are applied to isolation circuits before amplification in the output driver stages. The isolation circuits are single-input "and" gates consisting of diodes CR53 through CR60 and resistors R37 through R44 (figure 7-60 of Service Manual). The gate output lines are clamped near ground potential when the gates are inhibited to ensure positive cutoff of the corresponding drivers.

Input-Output Logic Board No. 2

Input-output logic circuit board No. 2 (figure 7-64 in RECOMP III Service Manual) contains conventional "and" and "or" gates for the selection, timing, and control of the optional input-output devices. Refer to the description of Input-Output Processes in this section for the functioning of input-output logic circuit board No. 2 when the computer is in the input and output modes.

Input-Output Driver Boards

Schematic diagrams of input-output driver circuit boards No. 1 and No. 2 are shown on figures 7-56 and 7-58, respectively, of the Service Manual. Both boards are similar in configuration and function, each containing driver circuits which amplify signals transmitted from the computer to the Flexowriter. Input circuits tape advance, code delete, anti-repeat, and the output circuit gate are not utilized.

Each driver board contains nine driver circuits, but one circuit on driver board No. 2 is not utilized. View a of figure 6-44 is a simplified block diagram which illustrates operation of the input-output driver circuits. Each driver circuit consists of three transistor amplifiers in series. Six of the driver circuits are connected in parallel to amplify signal P_{kd} which drives the code translator and tape punchclutches when the Flexowriter is selected for output. Each of the signals from flip-flops D_1 through D_6, C_5 and C_6 is amplified in a driver circuit, as are input-output device select signals R_{ks}, P_{ks}, and R_d.

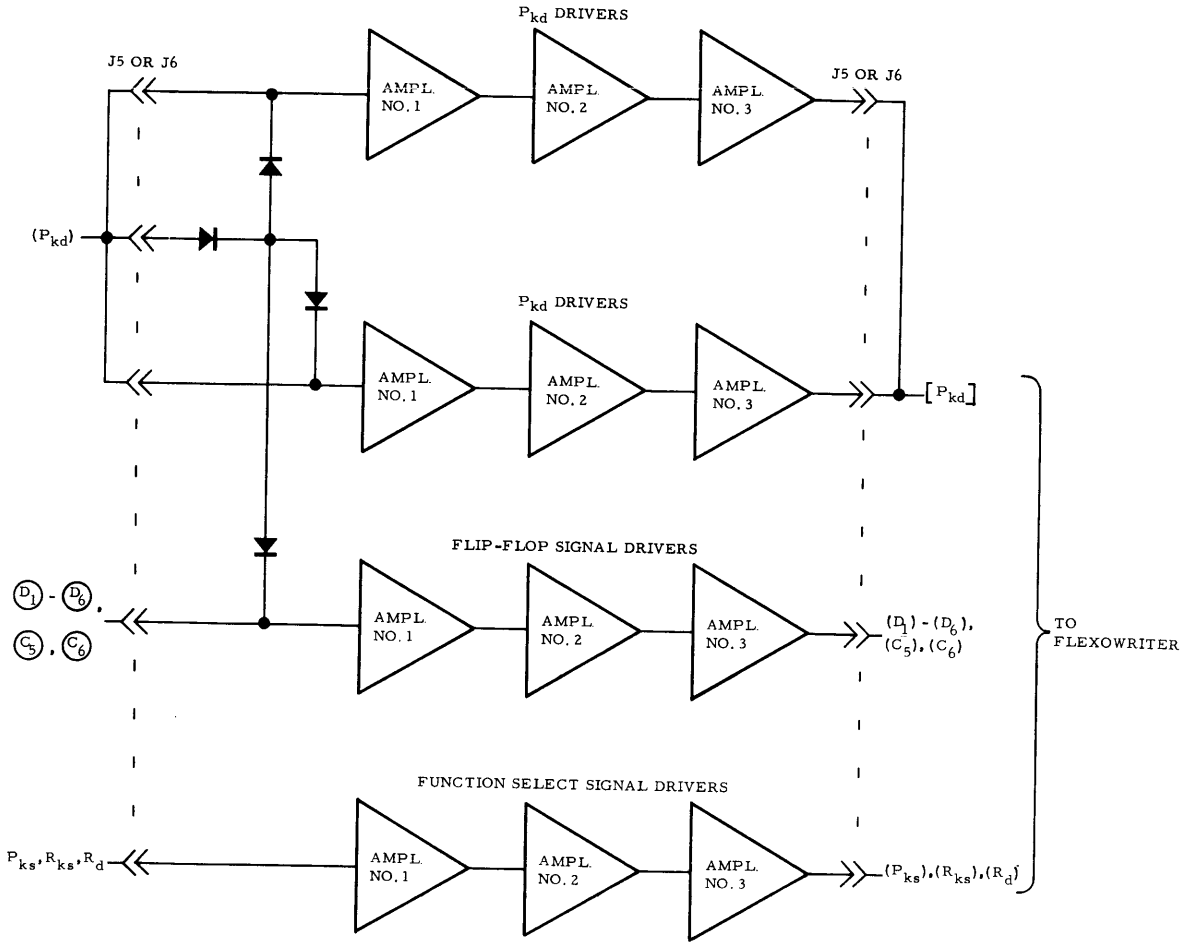
The base circuits of the No. 1 amplifiers (figure 6-44) of the drivers for the P_{kd} and flip-flop signals are connected through diodes to a common line. Signal P_{kd} is applied to the input circuits of the six P_{kd} drivers, and also to the anode of a diode whose cathode is connected to the common line. The diode must be reverse biased by a true P_{kd} signal before the No. 1 amplifiers can conduct. When the P_{kd} signal is false and when true signals are applied to the input circuits of the flip-flop signal drivers, the diodes connected between the base circuits of the No. 1 amplifiers and the common line are forward biased. The P_{kd} diode is also forward biased and conducts the flip-flop output signals to ground. Therefore, the P_{kd} drive signal must be generated before an 8-bit output character can be amplified and transmitted to the Flexowriter. Base circuits of the No. 1 amplifiers of the function select signal drivers are not connected to the common line and function independently of the P_{kd} signal.

Operation of each of the driver circuits is identical except for the variations previously noted in the input circuits. No. 1 amplifiers are at cutoff before the application of true input signals to the driver circuits (figure 7-56 of Service Manual). The -18-volt supply voltage applied across the No. 1 amplifiers load resistors keeps the No. 2 amplifiers conducting heavily. No. 3 amplifiers are held at cutoff due to the flow of collector current through the No. 2 amplifiers' load resistors.

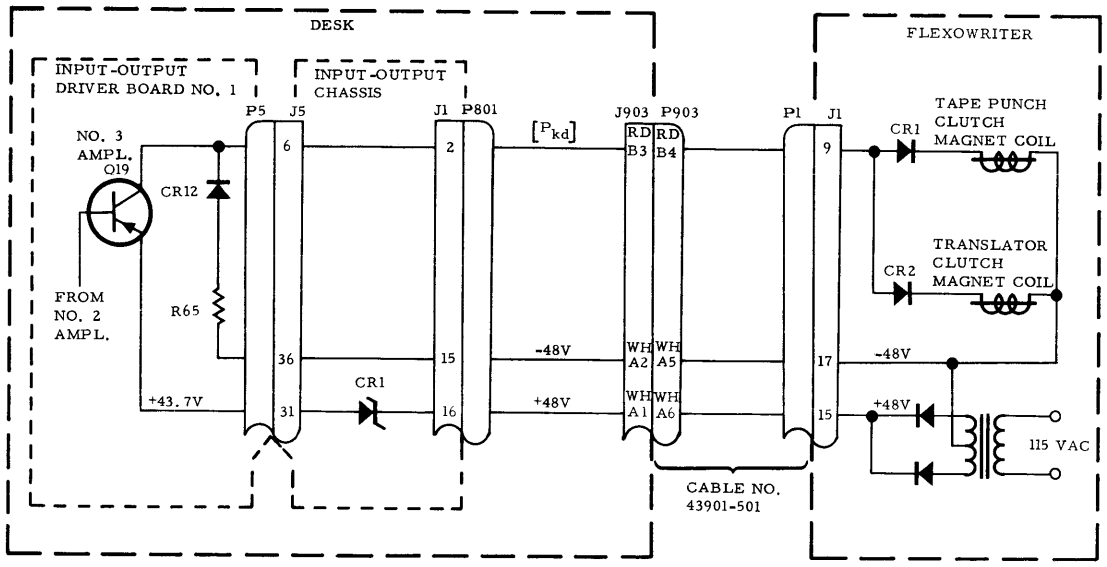
No. 1 amplifiers conduct at the occurrence of true input signals, and current flow through the load resistors cuts off the base drive to No. 2 amplifiers. As No. 2 amplifiers cut off, the -18-volt supply voltage applied across the load resistors drives No. 3 amplifiers into conduction. View b of figure 6-44 is a simplified schematic diagram which illustrates operation of a P_{kd} driver output circuit as the No. 3 amplifier conducts. Q19 is the No. 3 amplifier of the first driver circuit on input-output driver circuit board No. 1. Operation of the output circuit is typical of all of the driver circuits in that they are transmitted to the Flexowriter to energize electromagnets or relay coils which convert electrical energy into mechanical action of the printer or tape punch.

The emitter circuit of Q19 is routed through a 4.3-volt Zener diode to the +48-volt output terminal of the Flexowriter power transformer. Q19 collector is connected through the paralleled tape punch and translator clutch magnet coils to the -48-volt terminal of the transformer. The Zener diode breaks down as Q19 conducts and clamps the emitter potential at -4.3 volts in respect to the collector potential. Simultaneous conduction of the six P_{kd} No. 3 amplifiers draws sufficient current through the two magnet coils to actuate the tape punch and translator clutches. Clamping action of the Zener diode ensures a steady flow of current through the coils. Diode CR12 provides a limiting action in the collector circuit to restrict the swing of the collector potential to -48 volts.

Emitter circuits of the flip-flop signal and function-select No. 3 amplifiers are also routed through the Zener diode to the +48 volt terminal. Operation of these amplifiers is identical to that of Q19 described above, except that the collector circuits of the flip-flop signal amplifiers are connected to energize either the tape punch or translator code magnets, and those of the function-select amplifiers to energize appropriate relays.



(a) BLOCK DIAGRAM



NOTE: CR1 ON INPUT-OUTPUT CHASSIS IS A 4.3-VOLT ZENER DIODE.

(b) INTERCONNECTING SCHEMATIC

R3-211

Figure 6-44. Input-Output Driver Circuits.

Input-Output Processes

Only one input-output device may be selected for use in conjunction with the computer at a particular time. Coding in a computer program automatically selects a device when either the INPUT or OUTPUT SELECT switches are set to AUTO; positions 1 through 5 of each switch manually select one of the five devices for input or output (figure 6-42). Optional input and output devices Nos. 3, 4, and 5 may be any device that is compatible with the RECOMP III computer.

Each input-output device connected to the computer generates signals which define the active and idle states of the device. The input devices generate signal T_r when in the active state and signal T'_r when in the idle state; output devices generate signal T_p when in the active state and signal T'_p when in the idle state. When all input-output devices are in the idle state, the T'_r and T'_p signals provide true inputs to "and" gates. These "and" gates on input-output logic circuit boards No. 1 and No. 2 generate timing signal T_n when enabled. This timing signal inhibits all transfer of information between the computer and the input-output devices when at a true level. The two gate outputs are in parallel, and a false input signal to either gate inhibits both gates.

The input-output devices remain in the idle state until activated by a drive signal from the computer. When the computer is in the input mode, a drive signal (R_{ks} or R_d) is transmitted to a selected device to indicate that the computer is ready to receive information (figure 6-45). If the computer is in the output mode, a drive signal (P_{kd} or P_{ks}) is transmitted to a selected device to indicate that the computer is ready to transmit information (figure 6-46).

When activated by a drive signal, the Flexowriter generates timing signal T. This signal represents the Flexowriter input or output signals T_r or T_p, respectively, when it is in the active state. Signals T_r and T_p are applied to a six-input "or" gate on logic board No. 2, and the gate is enabled when any one of the input signals is at a true level. The T signal line from the Flexowriter and the T gate output line are joined at a clamping circuit on logic circuit board No. 1. This circuit clamps the common T signal line to the computer to a positive voltage until one of the input-output devices is activated. If either the T signal from the Flexowriter or the output of the T gate is at a true level, the true signal overrides the positive clamping voltage and enables computer logic circuits which initiate the input or output operation. The T signal remains at a true level during the period that a designated number of input or output characters are being transmitted.

Input Unit Selection

Any one of up to five devices can be selected for entering information into the computer. The Flexowriter keyboard and tape reader, as standard equipment, are always available for selection; any other compatible input device can be selected when it is connected to receptacles J904, J906, and J908 and input-output logic board No. 2 is added to the input-output chassis. For manual selection of a device, the switch contacts of the INPUT SELECT switch generate three

I-signals when the switch is set to a numbered position. The I-signals are then combined with input signal I_a in three device selection gates whose output signals set flip-flops C_2, C_3, and C_4 to the states required for the selection of a designated device (table 6-3).

When the keyboard is selected for input, the three flip-flops are set to the false states, and signals from the primed outputs are applied to a function selection gate on logic circuit board No. 1. The gate generates signal R_{ks} when output ready signal O_r from the Flexowriter is true along with I_a. R_{ks} enables manual input from the keyboard and inhibits input from the tape reader. When the tape reader is selected for input, the three flip-flop signals are applied to a drive signal gate on the same logic board. This gate generates signal R_d which enables input from the tape reader. R_{ks} and R_d are amplified in driver circuits on driver board No. 2 before they are applied to the Flexowriter.

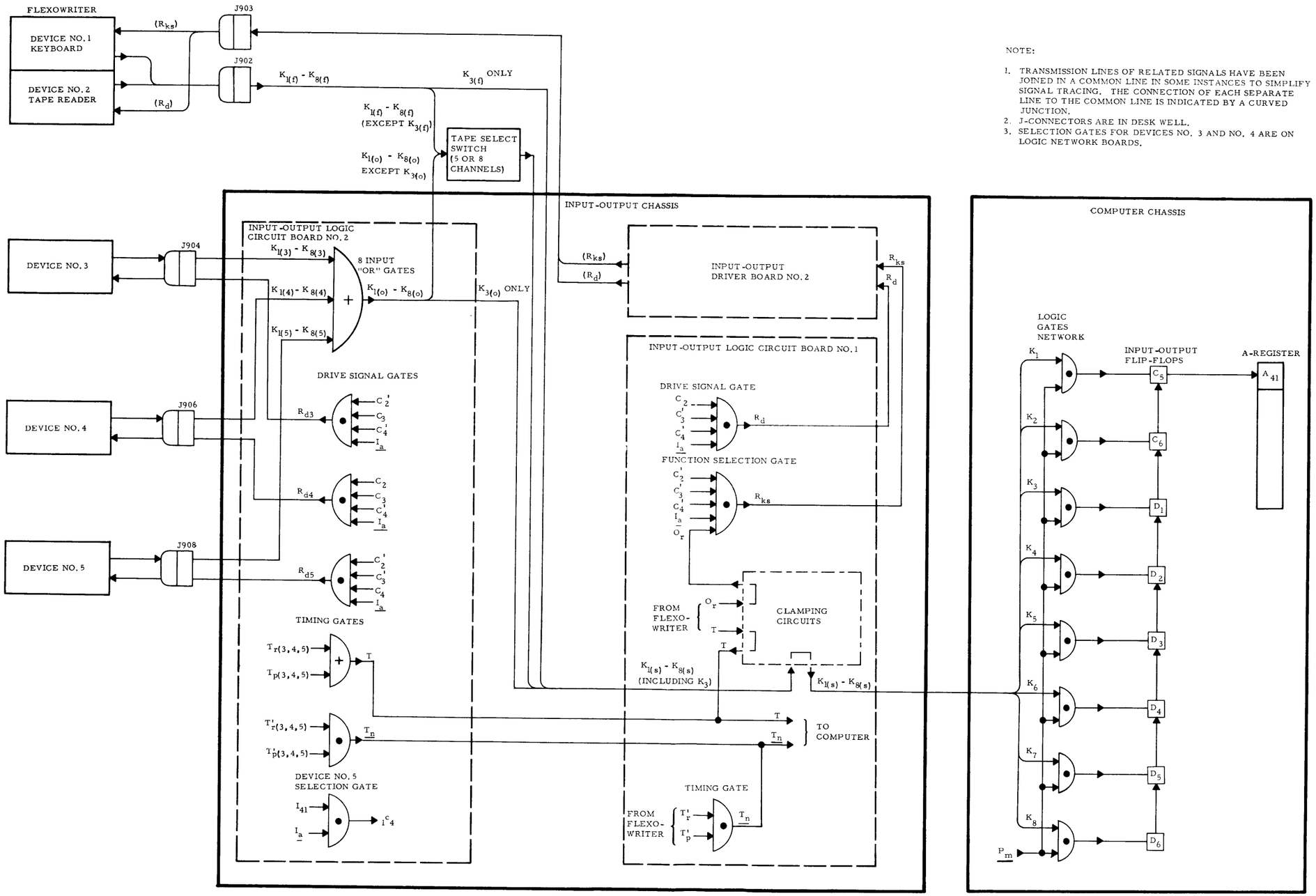
Logic circuit board No. 2 contains selection gate for input device No. 5. Selection gates for input devices No. 3 and No. 4 are located on logic network circuit boards in the computer chassis. In figure 6-45, signal I_{41} from position 5 of the INPUT SELECT switch is combined with I_a in the selection gate to set C_4 to the true state required for the selection of device No. 5. Position 5 also generates signals I_{30} and I_{20} which are combined with I_a in two logic network gates whose output signals set C_3, and C_2, respectively, to the required false states.

When input device No. 5 is selected for input, the output signals from C_2', C_3', and C_4 are combined with I_a in a drive signal gate on logic board No. 2. The enabled gate generates drive signal R_{d5} which activates the input device connected to receptacle J908. When device No. 3 or No. 4 is selected, drive signals R_{d3} or R_{d4} are generated in a similar manner to activate the device connected to receptacle J904 or J906, respectively.

Input Data Signal Routing

Presuming that device No. 5 has been activated by R_{d5}, signal T'_{r5} assumes a false level and signal T_{r5} assumes a true level. The false T'_{r5} signal inhibits the T_n "and" gate and T_n assumes a false level. The true T_{r5} signal enables the $\overline{T}$ "or" gate and transmits a true level T signal which 1-sets flip-flop S in the computer at T_{41} time, and the computer is ready to receive information from input device No. 5

During the period that T is true, device No. 5 generates (during 8-channel operation) eight output signals simultaneously ($K_{1(5)}$-$K_{8(5)}$) which represent an 8-bit character. The eight output signals are applied to eight three-input "or" gates on logic board No. 2. Two other inputs to the "or" gates are connected to the corresponding K-lines of the idle optional input devices (figure 6-45),



NOTE:

1. TRANSMISSION LINES OF RELATED SIGNALS HAVE BEEN JOINED IN A COMMON LINE IN SOME INSTANCES TO SIMPLIFY SIGNAL TRACING. THE CONNECTION OF EACH SEPARATE LINE TO THE COMMON LINE IS INDICATED BY A CURVED JUNCTION.
2. J-CONNECTORS ARE IN DESK WELL.
3. SELECTION GATES FOR DEVICES NO. 3 AND NO. 4 ARE ON LOGIC NETWORK BOARDS.

R3-179

Figure 6-45. Input Logic Functional Diagram.

and the gates are enabled by true signals from any of the three devices. The "or" gates generate signals $K_{1(o)}$-$K_{8(o)}$ when enabled, and all of these signals except $K_{3(o)}$ are routed through the TAPE SELECT switch. This switch provides compatibility between the input devices and the computer when 5-channel tape is utilized, as previously described. Signals $K_{1(f)}$-$K_{8(f)}$ (except $K_{3(f)}$) from the Flexowriter are also applied to the same contacts of the TAPE SELECT switch as the corresponding signals from the optional devices. When the Flexowriter is transmitting information on less than eight channels, the unused channel positions are grounded.

The K-signals routed through the TAPE SELECT switch are applied to clamping circuits on logic board No. 1 along with the $K_{3(o)}$ or $K_{3(f)}$ signal. These clamping circuits supply the ground current for keeping the K-signals at positive voltage when they represent binary zeros. Thus, K-signals representing binary ones at approximately -12-volt level override the positive clamping voltage, whereas K-signals representing binary zeroes at near ground potential remain at a positive level. After leaving the clamping circuits the K-signals are combined with signal P_m in eight 2-input "and" gates which are located on logic network circuit boards. P_m assumes a true level only after S has been 1-set by timing signal T. The true K-signals enable their respective gates, but the false K-signal gates remain inhibited by the positive voltage from the clamping circuits.

Outputs of the eight "and" gates are connected to the 1-set inputs of flip-flops C_5, C_6, and D_1 through D_6. K_1 (least significant bit) is applied to C_5, and K_8 (most signifcant bit) is applied to D_6. The input-output flip-flops are 0-set just prior to the entry of each character, and the true K-signals then simultaneously set the corresponding flip-flops to the true state. Flip-flops to which the false K-signals are applied remain in the false state. Each bit of an input character is then serially shifted from C_5 into flip-flop A_{41} of the A-register.

Output Unit Selection

The output unit selected to receive information from the computer can be the Flexowriter printer or tape punch, or any other compatible output device that may be connected to receptacles J905, J907, and J909. However, selection of a device connected to any of these three receptacles is also dependent on the addition of input-output logic board No. 2 to the input-output chassis. For manual selection of a device, the switch contacts of the OUTPUT SELECT switch generate three O-signals when the switch is set to a numbered position (table 6-3). The O-signals are then combined with output signal I_o in device selection gates in a manner similar to that for input device selection (figure 6-42). Logic circuit board No. 2 contains the selection gate for output device No. 5. Selection gates for output devices No. 3 and No. 4 are located on logic network boards in the computer chassis.

When the printer is selected for output of information from the computer, the output signals of C_2', C_3', and C_4' are applied along with I_o to a function selection gate on logic circuit board No. 1. The gate generates signal P_{ks},

when enabled, which permits automatic output to the printer. P_{ks} is amplified in driver circuits on driver board No. 2 before application to the computer. When the printer is not selected, the computer outputs information to the tape punch.

Output Data Signal Routing

The eight bits that constitute an output character are shifted serially from flip-flop A_1 of the A-register into flip-flop D_6 (figure 6-46). These bits are precessed through D_5-D_1, C_6, and C_5, and after the precession has been completed C_5 contains the least significant bit and D_6, the most significant bit. The eight flip-flop output signals are applied directly to the input circuits of the three optional output devices, but require amplification before being applied to the Flexowriter.

Output devices cannot accept an output character until activated by a drive signal from the computer. The computer generates signal ($\overline{O_d}$) after the character bits have been precessed through the input-output flip-flops. Signal (O_d) is applied to the three optional device drive signal gates on logic board No. 2, and to the Flexowriter drive signal gates on logic board No. 1, (figure 7-60 and 7-64 of Service Manual). The optional device signal gates generate signals O_{d3}, O_{d4}, or O_{d5} when enabled and activate devices No. 3, No. 4 or No. 5, respectively. An activated device then generates signal T_p which enables the "or" timing gate on logic board No. 2, and timing signal T assumes a true level. During the period that T_p is true, the computer also transfers the next output character from the A-register to the input-output flip-flops.

All output signals from the input-output flip-flops are applied through isolation circuits to eight driver circuits on driver boards No. 1 and No. 2. The isolation circuits consist of eight single-input "and" gates whose outputs are near ground potential until the gates are enabled by true signals from the input-output flip-flops. Output of each gate is connected to the input circuit of the No. 1 amplifier of each of the eight driver circuits. The No. 1 amplifiers are held at cutoff by the conduction of the No. 2 amplifiers, and the clamping action of the isolation gates ensures that conduction does not occur until the gates are enabled by true signals from the input-output flip-flops. Also, the No. 1 driver amplifiers cannot conduct until drive signal P_{kd} has been generated as a result of the selection of a Flexowriter operation for output.

Signal P_{kd} is generated by the Flexowriter drive signal gates on logic board No. 1 when either the printer or tape punch has been selected for output (refer to Input-Output Logic Board No. 1 in this section). P_{kd} is applied to a clamping circuit on the same logic board, amplified in a logic driver circuit located in the computer chassis, and is then applied to six parallel driver circuits located on driver boards No. 1 and No. 2.

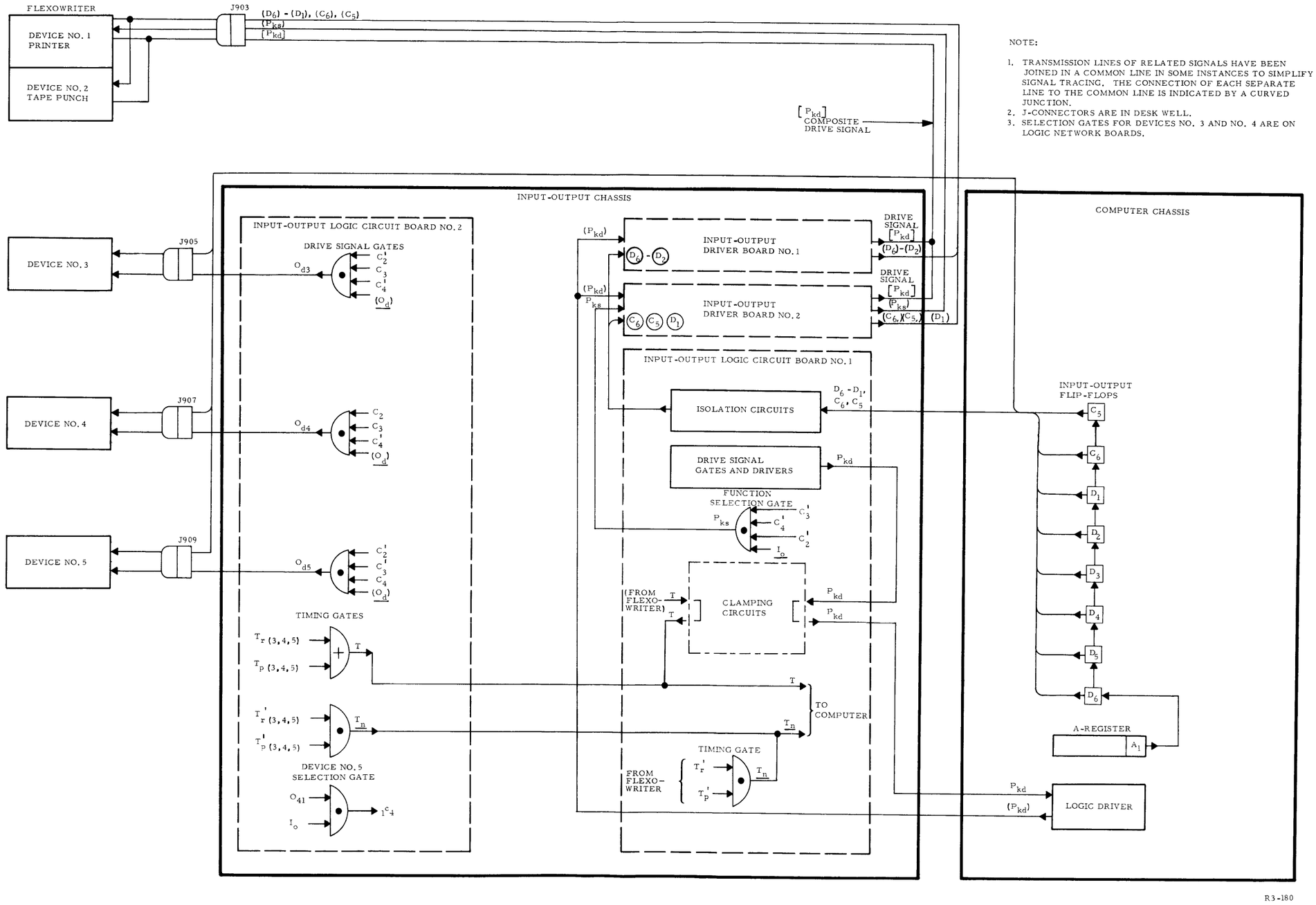


Figure 6-46. Output Logic Functional Diagram

When the P_{kd} drive signal is present at the inputs to the P_{kd} driver circuits, conduction of the flip-flop signal drivers occurs when true signals are applied from the isolation gates. The Flexowriter generates timing signal T when the composite P_{kd} signal is applied to the code translator and tape punch clutches, and signal T'_p assumes a false level during the period that T is true and inhibits the T_n timing gate. The 8-bit output character is then transmitted to the Flexowriter from the flip-flop signal drivers which generate binary ones or zeroes in response to true or false input signals. Channels 7 and 8 are always coded with zeroes when the printer is selected for output and may or may not be coded with zeroes when the tape punch receives the output.

Extended Input-Output Register Operation

The computer has the capability to transmit or receive information on a maximum of twelve channels when two additional circuit boards are connected to receptacles in the computer chassis. For example, a card reader and a card punch, when connected to receptacles J908 and J909, respectively, may be used with the computer for 12-channel input and output.

An additional standard flip-flop circuit board 57333-501 and extended register circuit board 44118-501 are connected to receptacles J327 and J317, respectively, for the extended register capability. The flip-flop circuit board contains flip-flops E_1 through E_4 which combine with D_6 through D_1, C_6 and C_5 to form a 12-bit input-output register. The extended register logic circuit board (figure 7-52 of Service Manual) contains conventional "and" gates and "or" gates which control operations between the computer and the extended register devices.

During input operations signals K_9 through K_{12} are transmitted from the card reader to flip-flops E_1 through E_4, respectively. Flip-flop E_4 thus receives the most significant bit of the 12-bit character, and E_1 precesses the four additional bits to D_6. All character bits then shift through the remainder of the input-output flip-flops into A_{41} of the A-register as previously described. The four additional K-signals are clamped to a +6 volts by clamping circuits on the extended register logic circuit board, and are then combined with signal P_m in "and" gates on the same board before application to the 1-set inputs of $\overline{E_1}$ through E_4.

During output operations E_4 copies A_1 of the A-register in the same manner that D_6 performs this function for the 8-bit character. After the 12 output bits have been precessed through the input-output flip-flops, C_5 contains the least significant bit and E_4 contains the most significant bit. Drive signal O_{d5} is then generated and activates the card punch, and the output character is applied directly to the device from the twelve flip-flops of the extended register.

Provisions For Future Applications

A number of computer signals are routed to the desk receptacles for optional input-output equipment which are not presently used. These include signals from device selection flip-flops C_2, C_3, and C_4, and input-output signals I_a and I_o. The signals are available to increase capability potential for accomodating possible future input-output equipment with special operating requirements.

FLEXOWRITER INPUT-OUTPUT UNIT

The RECOMP III Flexowriter is an input-output device consisting of a keyboard-printer (typing unit), a paper tape recorder (punch), a punched tape reproducer (reader), a code selector, and a code translator (figure 6-48). Conversion of information is performed by the code selector and the code translator. Electronic circuitry necessary for operation of the Flexowriter is mounted on etched circuit plug-in boards located in the desk.

Primary power is supplied to the Flexowriter from the 115-volt accessory jack at the rear of the desk. With switch S1 on the Flexowriter placed at ON, power is applied to all elements of the Flexowriter.

During operation with the computer, the Flexowriter (1) translates mechanical action of the keyboard (figure 6-47) or hole patterns in punched paper tape into binary coded signals for the computer and (2) converts binary coded electrical pulses from the computer into either printed copy, or punched paper tape. In local operation, coded signals resulting from mechanical action of key depression can be transmitted directly to the tape punch, or information can be transmitted directly from the reader to the punch and/or printer

When switch S2 is placed at LOCAL the Flexowriter will read and punch any and all coded paper tapes up to and including 8-channel tape. While in local mode the printer mechanism, which is 6-channel binary-code operated, receives information from the reader and prints out an alphanumeric printed copy at a nominal speed of 10 characters per second. The printer mechanism is capable of reproducing information from a punched tape in the form of printed copy, or producing a properly coded 6-channel tape by entering information through the keyboard to the tape punch.

With switch S2 at COMPUTE, the tape reader and code selector transmit in binary code up to and including 8-channels of information to the computer. The punch receives from the computer, and in turn reproduces, binary coded information up to and including 8 channels. When outputing information from the computer to the printer, the translator recognizes information formats of all properly coded 6-channel codes, and causes action by the printer. Contents of channels 7 and 8 are ignored by the translator when outputing to the printer. A keyboard action during information input to the computer, in turn, causes the code selector to transmit properly coded 6-channel codes to the computer. Input-output codes utilized by the printer and keyboard are shown in table 6-4.

Figure 6-47. Flexowriter Keyboard, Control Panel and Input Indicator Light.

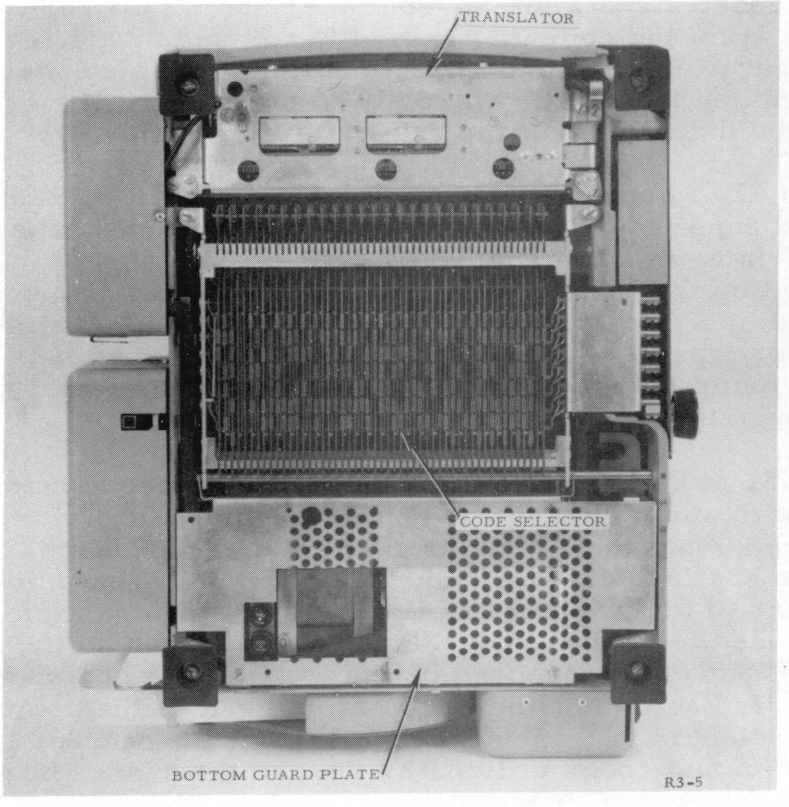
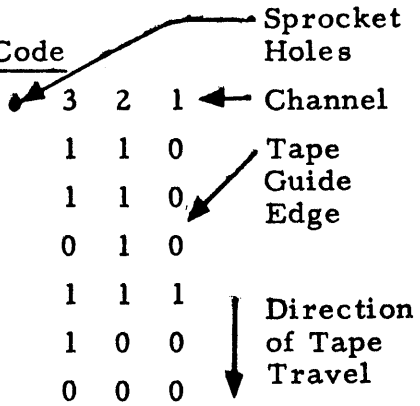


Figure 6-48. Code Selector, Code Translator.

Table 6-4. Input-Output Character Coding

A. Alphanumeric Keys

Type Bar Number	Upper Case Key	Lower Case Key	Binary Code 8	7	6	5	4	3	2	1
1	Q	q	0	0	0	1	0	1	1	0
2	A	a	0	0	0	0	0	1	1	0
3	*	2	0	0	1	1	0	0	1	0
4	Z	z	0	0	0	1	1	1	1	1
5	W	w	0	0	0	1	1	1	0	0
6	S	s	0	0	0	1	1	0	0	0
7	#	3	0	0	1	1	0	0	1	1
8	X	x	0	0	0	1	1	1	0	1
9	E	e	0	0	0	0	1	0	1	0
10	D	d	0	0	0	0	1	0	0	1
11	△	4	0	0	1	1	0	1	0	0
12	C	c	0	0	0	0	1	0	0	0
13	R	r	0	0	0	1	0	1	1	1
14	F	f	0	0	0	0	1	0	1	1
15	°	5	0	0	1	1	0	1	0	1
16	V	v	0	0	0	1	1	0	1	1
17	T	t	0	0	0	1	1	0	0	1
18	G	g	0	0	0	0	1	1	0	0
19	π	6	0	0	1	1	0	1	1	0
20	B	b	0	0	0	0	0	1	1	1
21	Y	y	0	0	0	1	1	1	1	0
22	H	h	0	0	0	0	1	1	0	1
23	–	7	0	0	1	1	0	1	1	1
24	N	n	0	0	0	1	0	0	1	1
25	U	u	0	0	0	1	1	0	1	0
26	J	j	0	0	0	0	1	1	1	1
27	:	8	0	0	1	1	1	0	0	0
28	M	m	0	0	0	1	0	0	1	0
29	I	i	0	0	0	0	1	1	1	0
30	K	k	0	0	0	1	0	0	0	0



(Continued)

Table 6-4. Input-Output Character Coding (Continued).

A. Alphanumeric Keys (Continued).

Type Bar Number	Upper Case Key	Lower Case Key	Binary Code
31	(	9	0 0 1 1 1 0 0 1
32	>	,	0 0 1 0 1 1 1 0
33	Ø	o	0 0 0 1 0 1 0 0
34	L	1	0 0 0 1 0 0 0 1
35	)	∅	0 0 1 1 0 0 0 0
36	<	.	0 0 1 0 1 1 1 1
37	P	p	0 0 0 1 0 1 0 1
38	≠	-	0 0 1 1 1 0 1 0
39	'	1	0 0 1 1 0 0 0 1
40	√	/	0 0 1 1 1 1 0 0
41	[	]	0 0 1 1 1 1 0 1
42	=	+	0 0 1 1 1 0 1 1
43	↓	↑	0 0 1 1 1 1 1 0

B. Function Keys

Type Bar Letter	Function	Binary Code
Q	Lower Case	0 0 0 0 0 0 1 0
R	Upper Case	0 0 0 0 0 0 0 1
S	Tab	0 0 1 0 1 1 0 1
T	Space	0 0 1 0 1 0 1 1
U	Blank	0 0 0 0 0 0 0 0
V	Carriage Return	0 0 1 0 1 1 0 0
W	Back Space	0 0 0 0 0 0 1 1
Y	Upper Case	0 0 0 0 0 0 0 1
Z	Lower Case	0 0 0 0 0 0 1 0

Table 6-4. Input-Output Character Coding (Continued).

C. Control Switches.

Switches	Function	Binary Code							
G	Code Delete	0	0	1	1	1	1	1	1
H	Stop Code	0	0	1	0	1	0	1	0
A	Start Read	0	0	0	0	0	0	0	0
B	Stop Read	0	0	0	0	0	0	0	0
C	Non Print	0	0	0	0	0	0	0	0
D	Ignore Stop	0	0	0	0	0	0	0	0
E	Punch On	0	0	0	0	0	0	0	0
F	Tape Feed	0	0	0	0	0	0	0	0

Control Switches

Ten switches are provided for controlling the various functions of the Flexowriter. Eight of these switches are located on the control panel and two are at the right side of the keyboard (figure 3-4).

START READ: Effective only in local mode, this switch, when depressed and released, automatically starts the tape reader operation. To stop reader operation the STOP READ switch must be actuated.

STOP READ: When this switch, effective only in local mode, is depressed and released, the reader will stop operation; in order to resume automatic reader operation, the START READ or the NON-PRINT switch must be actuated.

NON-PRINT: Actuation of this switch, effective only in local mode, will set up a state which will allow direct reproduction of tape, from reader to punch, without sending signals through the translator, keyboard, and code selector combination. With no printed copy being produced it allows reproduction of any and all codes, up to and including 8-channel operation.

IGNORE STOP: This switch, when in local mode, will set up a state that will cause the reader not to recognize and stop on the stop code. This state will be set by pressing the switch downward, and will continue until the switch is manually raised to its off position. The IGNORE STOP switch has no function in compute mode.

PUNCH ON: When this switch, effective only in local mode, is pressed downward, the punch circuits are energized in a standby mode to be ready to punch tape on receipt of information from the reader or the keyboard.

TAPE FEED: This switch, when depressed, will cause the tape to feed forward for as long a period as the switch is held depressed. During actuation of this switch, the only holes punched in tape are sprocket holes. This switch is effective only in the local mode.

CODE DELETE: While in the local mode, if this switch is held down the tape punch punches out all holes in the first 6 channels of the tape to delete a code punched in error. This switch has no function in the compute mode. When a corrected tape is then used as input information to the printer, the delete code is read by the reader, the tape will be advanced, and no printer operation will take place. Printer operation will be resumed when the tape reader reads the next character following the delete code.

STOP CODE: This switch, when depressed, while in the local mode, will cause the punch to punch a code on the tape which will be read by the reader and cause the reader to stop. However, the reader will ignore this code and continue to read if the IGNORE STOP switch was closed prior to the sensing of the stop code in the tape. The STOP CODE switch has no function in the compute mode.

POWER: This switch, located on the right side of the keyboard (figure 3-4), controls application of 115-volt ac power to the Flexowriter.

LOCAL-COMPUTE: This switch, located to the rear of the power switch (figure 3-4), selects operation modes as follows:

LOCAL -- In this mode the printer can receive properly coded 6-channel information through the tape reader and reproduce alphanumeric printed copy at ten characters per second. It is possible to duplicate up to 8-channel tapes using the reader in conjunction with the tape punch. In local mode, 6-channel tapes can be prepared using keyboard, code selector, and punch combination.

COMPUTE -- In this mode the tape reader or keyboard can transmit information to the computer. The printer or tape punch can receive information from the computer while in this mode and prepare either a printed copy containing this information (using the keyboard printer) or a punched paper tape (using the tape punch unit).

Drive System

The drive system comprising a motor unit and power drive mechanism supplies all mechanical power required to operate the various mechanisms and units of the Flexowriter.

Motor Unit

Mounted in a cradle hinged to the rear base plate, the Flexowriter drive motor is a compound-wound, constant speed, 115-volt ac motor rated at 35 millihorsepower. A differential relay (K1) is wired in series with the main motor winding which is initially energized when the power switch is turned on. K1 stays energized only during the brief starting period of the motor and closes contacts which complete the starter winding circuit. As the motor approaches operating speed (1750 rpm), starting torque (and therefore current flow) is reduced. When the current flow drops below a predetermined threshold value, the holding contacts of starting relay K1 open and the starter winding circuit is de-energized. Thus, once the motor reaches its running speed, it will operate by the circuit through its main windings only.

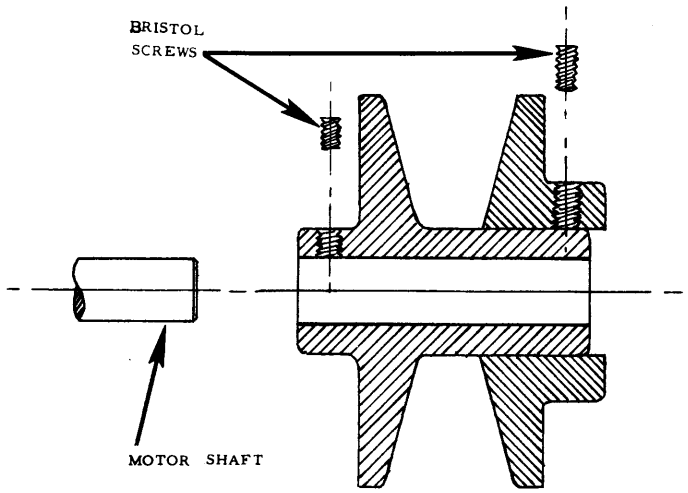


Figure 6-49. Split Pulley Cross Section.

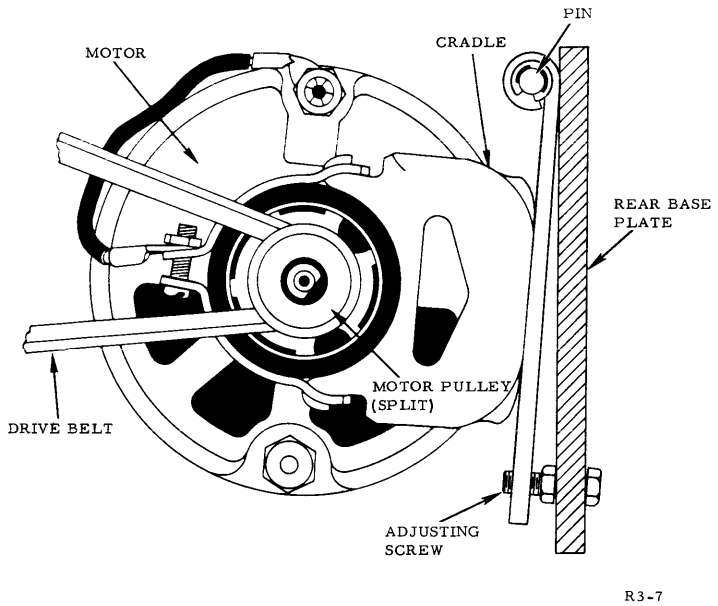


Figure 6-50. Motor Unit.

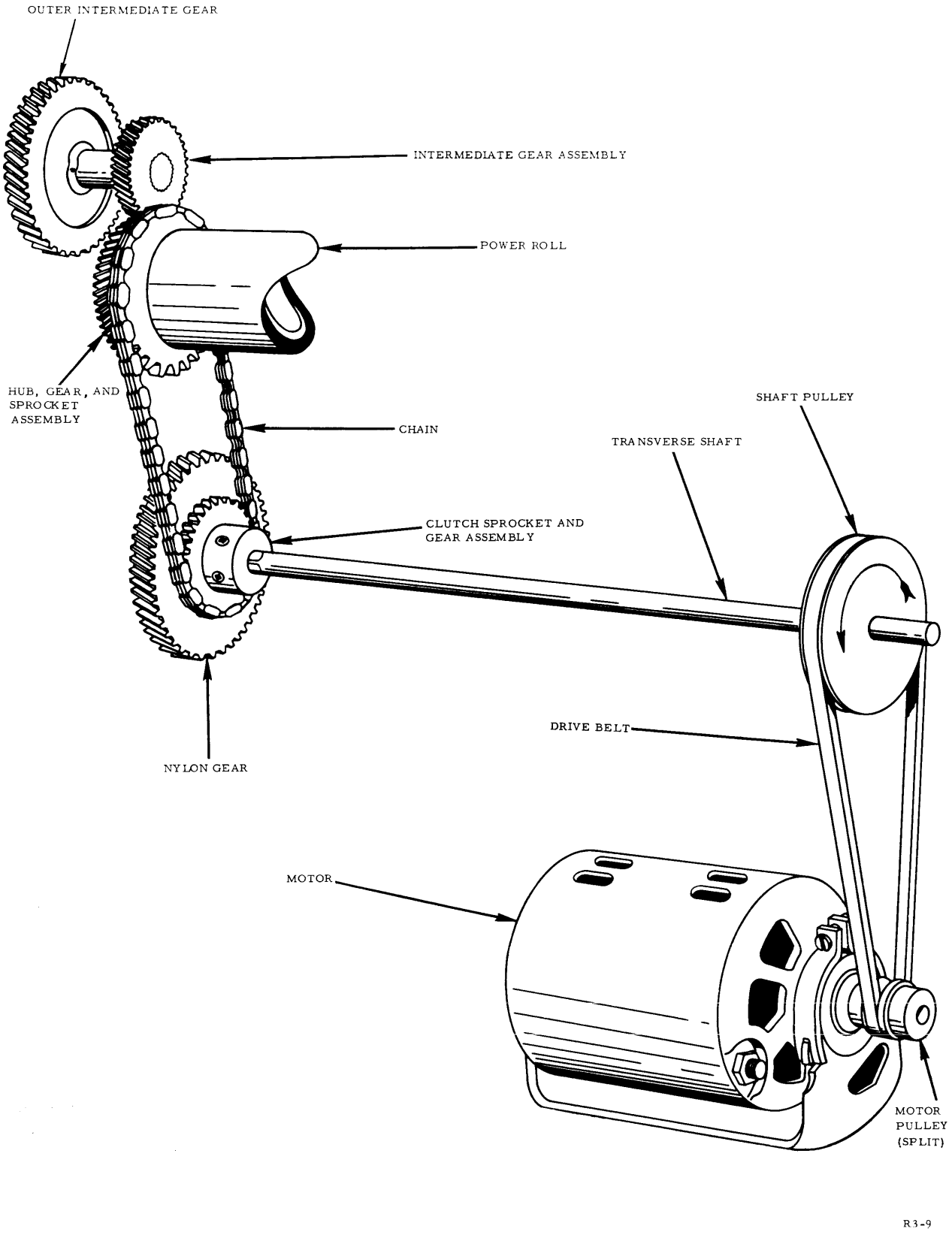


Figure 6-51. Power Drive Mechanism.

Adjusting screws with lock nuts enable adjustment of drive belt tension. A split pulley on the motor shaft allows changing the speed of the drive, if necessary (figures 6-49 and 6-50).

Power Drive Mechanism

Comprising a transverse shaft, belt or chain drive, power roll, and gear train (figure 6-51), the power drive mechanism transmits the power from the motor to the printer, reader, translator, and punch at proper operating speed. The motor is connected by a belt to a transverse shaft, which is connected by silent gears to the clutch assembly of the tape punch. A chain, or on some machines a rubber gear belt, connects the transverse shaft to the hub portion of the clutch sprocket on the power roll. The hub, in turn, is connected by silent gears to the intermediate gear assembly which drives the translator clutch and reader. Independently, the power roll furnishes the driving force for all the cam units. When a cam is released by operation of a keylever, the serrated surface of the cam engages the rubber surface of the roll, forcing the cam unit away from the power roll. This action provides the movement necessary to operate the typebar or function-key mechanisms of the machine.

Inter-Mechanism Operation

Flexowriter mechanisms involved in a specific input-output or tape preparation-reproduction operation vary, depending on the mode of operation and the specific operation selected. Table 6-5 lists the principal mechanisms involved in originating, converting, and receiving information during the various modes and operations.

Table 6-5. Inter-Mechanism Information-Flow Sequence.

From	Through	To
LOCAL MODE		
Keyboard	Code Selector	Tape Punch
Keyboard		Printer
Keyboard	Code Selector	Tape Punch and Printer
Tape Reader		Tape Punch
Tape Reader	Code Translator	Printer
Tape Reader	Code Translator and Code Selector	Printer and Tape Punch
COMPUTE MODE		
Keyboard	Code Selector	Computer
Tape Reader		Computer
Computer	Code Translator	Printer
Computer		Tape Punch

Components that produce the electromechanical actions listed in table 6-5 are illustrated in figure 6-52.

Keyboard-Printer Mechanisms

The keyboard-printer performs the mechanical operations involved in producing (1) in local mode, binary-coded electrical signals and printed copy during tape preparation and printout, respectively, and (2) in compute mode, binary-coded electrical signals and printed copy during input-output, respectively. These operations, in turn, represent a composite result of actions performed by the various mechanisms comprising the keyboard-printer. Operation of these individual mechanisms is described in the following paragraphs.

Cam Units

In most keyboard-printer operations, cam units (figure 6-53) actuate mechanical linkages to cause the various actions of the keyboard-printer mechanisms. The cam units are of two types (figure 6-54): single lobe, which requires 360 degrees rotation for complete operation; and double lobe, which requires only 180 degrees rotation for complete operation. However, operation of both types of cams is similar except for longer time usage such as with carriage return, backspace, or tab which utilize the single lobe cam.

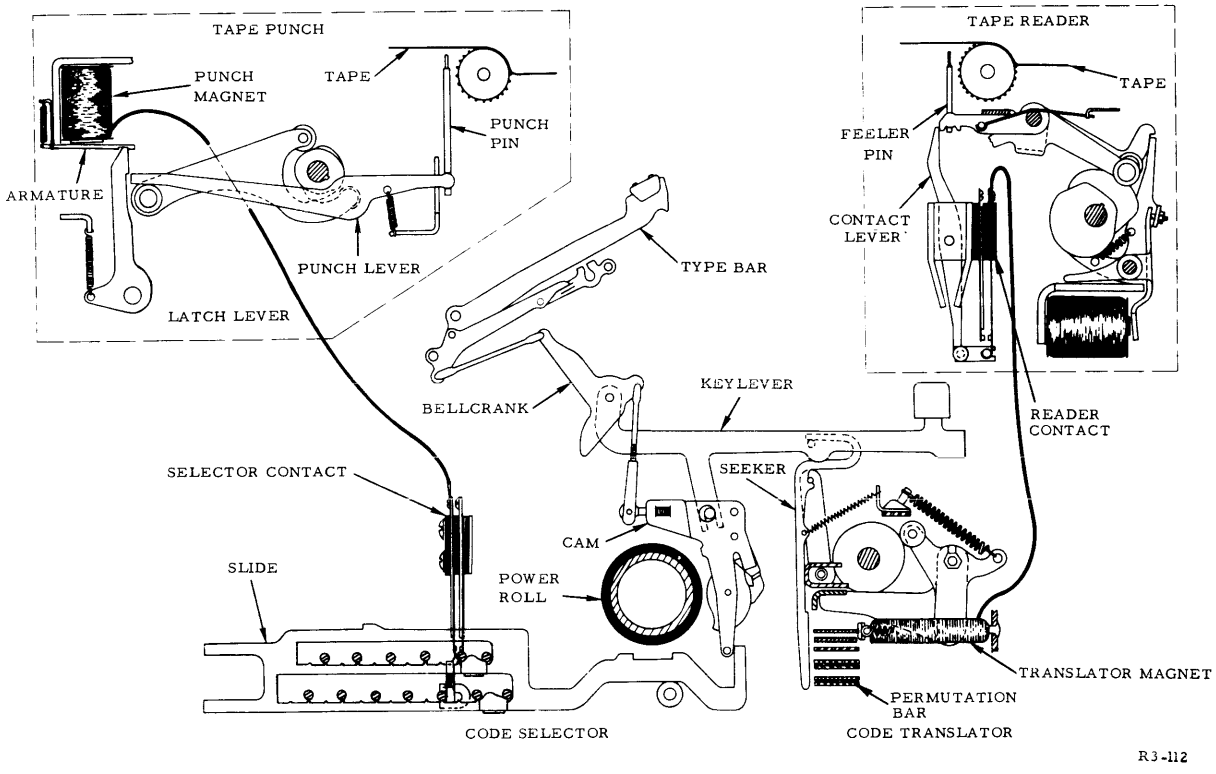


Figure 6-52. Functional Principles.

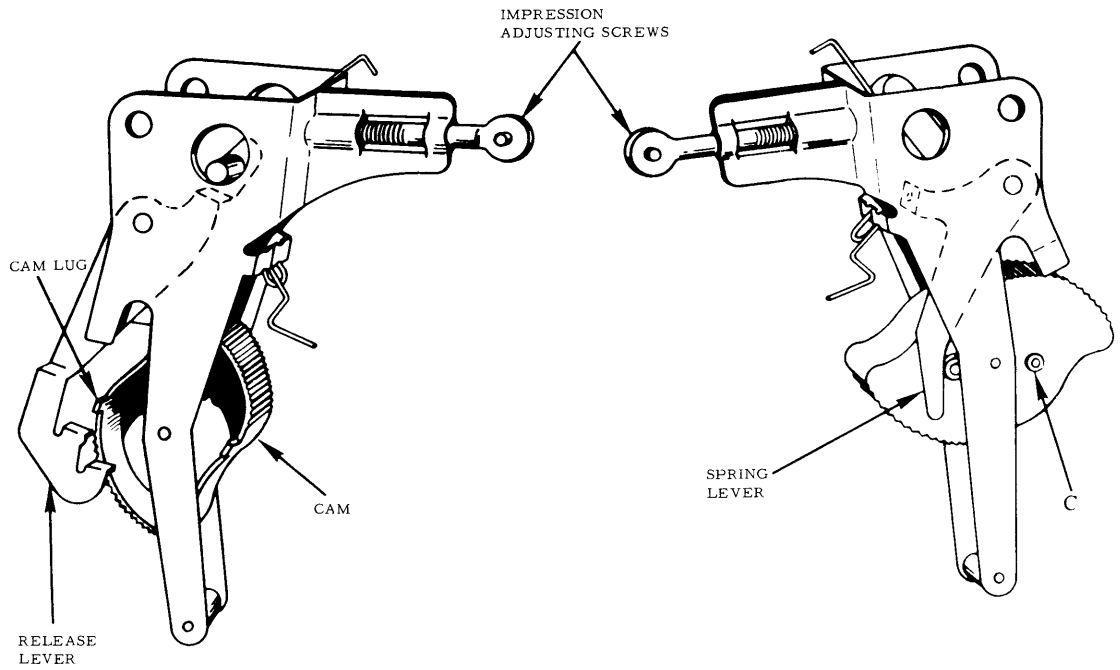


Figure 6-53. Cam Units.

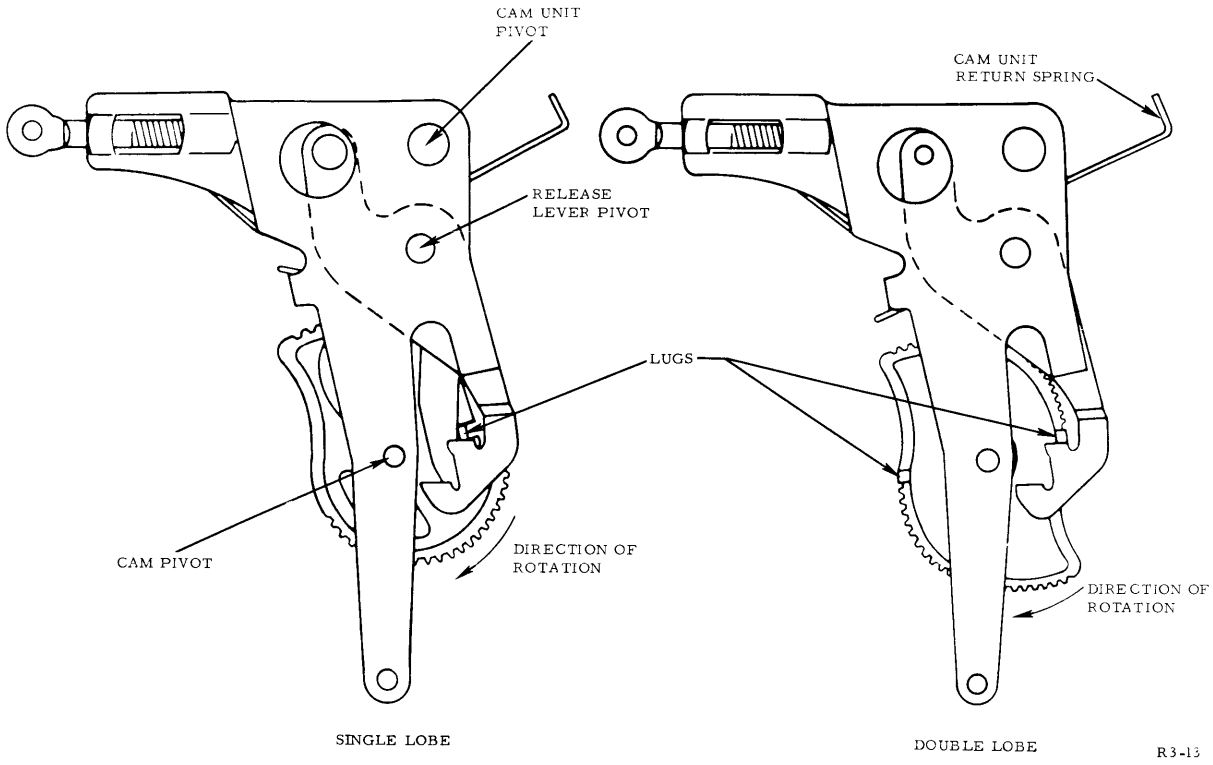


Figure 6-54. Single and Double Lobe Cams.

Keylever Operation

In all keyboard-initiated operations except upper and lower case, the keylever controls action of a respective cam unit. Keylever rows 1 and 2 operate a rear row of cams; keylever rows 3 and 4, a front row of cams (figure 6-55).

The keylevers are mounted on a keylever bearing wire held in the keylever bearing support. This mounting provides a pivot for the keylevers and the bellcranks. (figure 6-56). The keylevers are guided by the front guide comb and by the slot in the chassis of the power frame. When in their home position, the keylevers are held against the top of the slot in the front guide comb by springs and adjusting screws. The lower, forked ends of the keylevers extend through the power frame to engage the release levers on the various cam units. Keylevers must move freely and have proper tension to ensure uniform impression.

Keylever Interlock Operation

A mechanical interlock (figure 6-57) allows only one keylever to be depressed at one time, thus preventing pile-up of type bars, mispunching of codes in tape, and generation of erroneous signals to the computer. The interlock is effective on all keylevers except the lower case shift on the left side of the keyboard and the upper case shift on the right side. However, each of these two keylevers, through equalizing rods, works simultaneously with its companion keylever located on the opposite side of the keyboard.

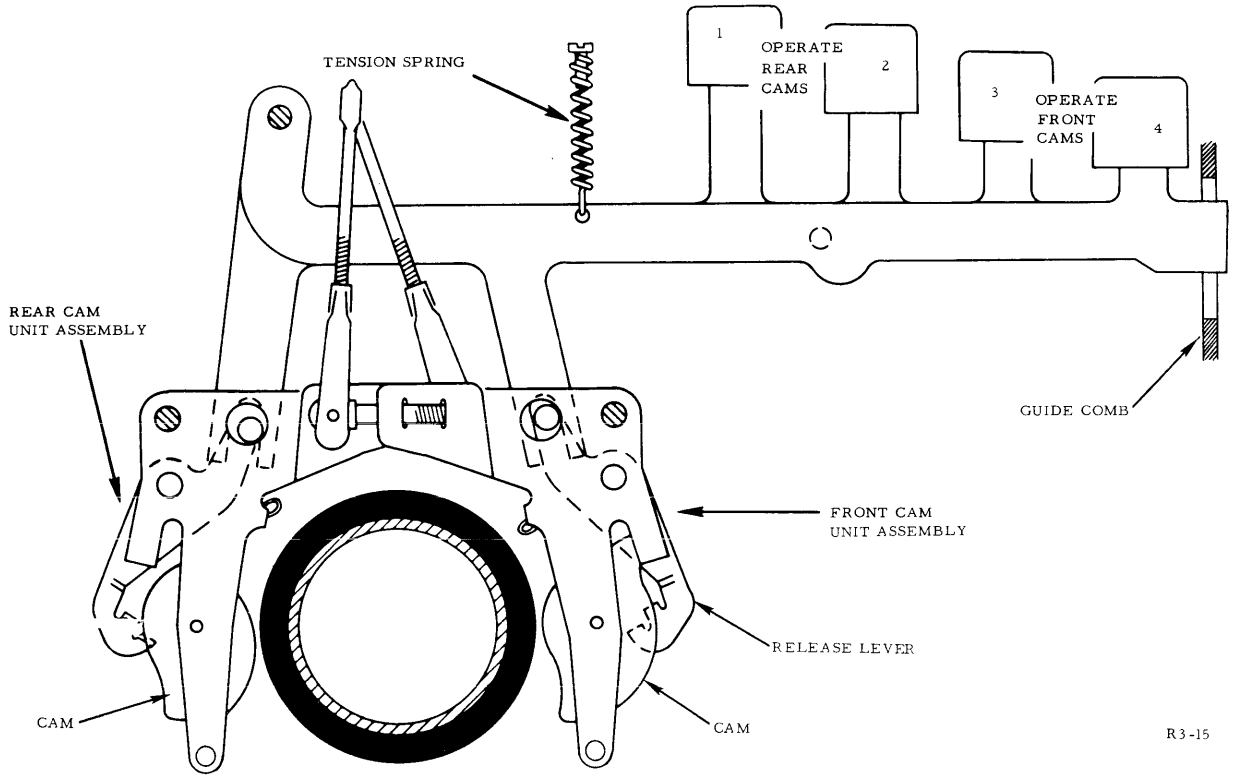


Figure 6-55. Keylever Positions and Cams.

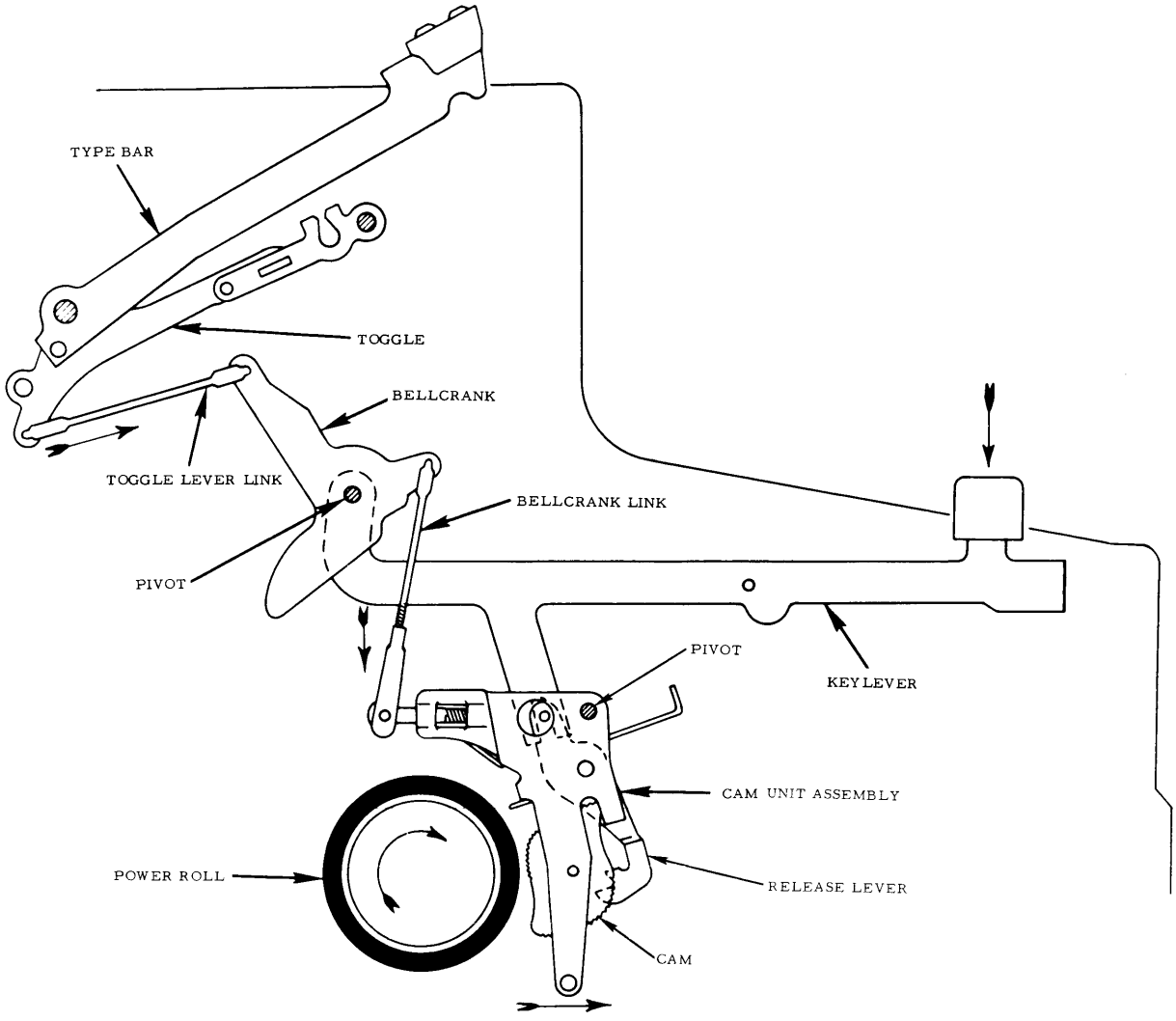


Figure 6-56. Keylever, Cam, Bellcrank, and Typebar.

This dual operation, thus, in effect, provides an interlock also for these keylevers.

The channel strip in back of the front guide comb contains one less roller than there are keylevers. As a keylever is depressed, it enters the channel strip between two rollers, forcing the other rollers on each side to be displaced laterally. This action blocks the other keylevers from entering between any rollers, which prevents another simultaneous keylever operation.

Keylever Lock Operation

To prevent operation of the keyboard should jamming or damage occur, a keylever lock and magnet (figure 6-58), locks the keyboard when any one of the following conditions exists:

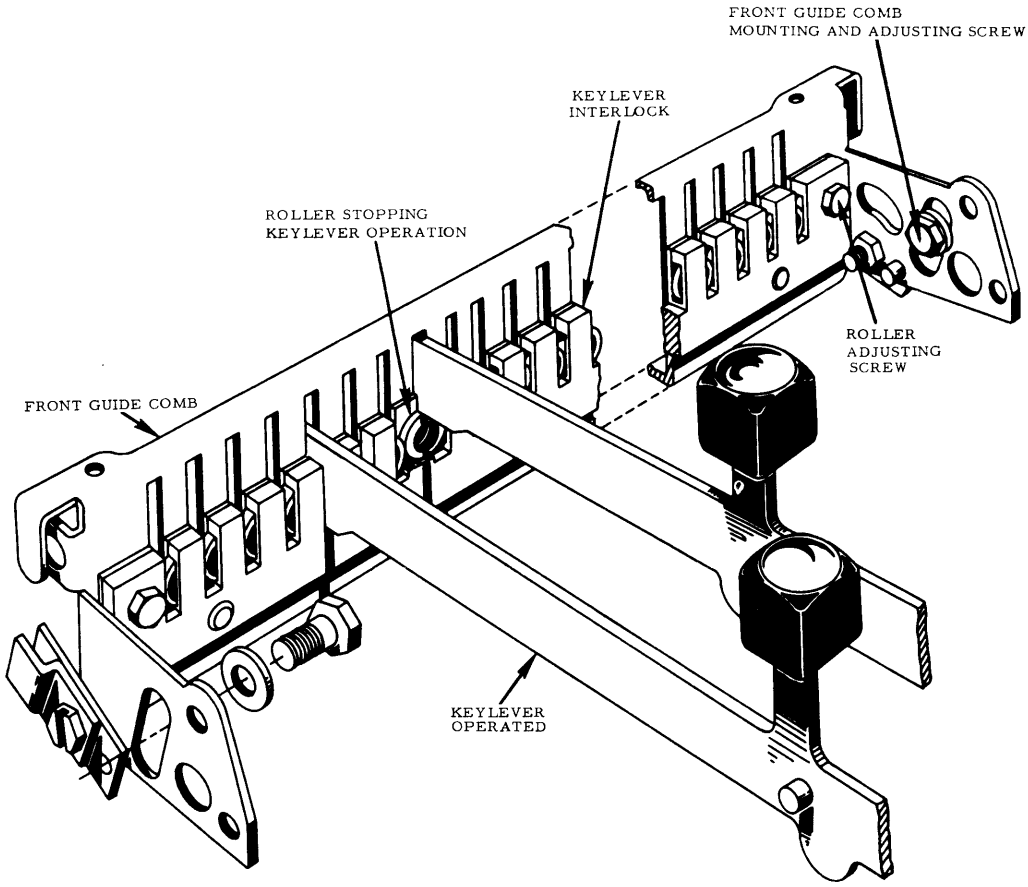


Figure 6-57. Keylever Interlock.

1. When the power switch is off, to prevent the keys from accidentally being operated and tripping the cams, which would cause pile-up of type or jamming of the power roll when the machine is turned on.
2. In local mode during carriage return, backspace, and tab operations, to prevent an operator from depressing a keylever before the machine has completed the selected operation.
3. When a tight tape condition occurs while preparing a punched tape from the keyboard or if a code bar should fail to return to its rest position at the completion of an operation.
4. In compute mode (except when the computer is transmitting to or receiving inputs from the keyboard) during carriage return, backspace, and tab operations.

The keylever lock bail pivots on the ends of the front guide comb and extends across the guide comb under the keylevers. A stud on the right end of the bail is connected by an adjustable link to the keylock magnet. When the magnet is operated, the bail is moved in an arc sufficient distance to allow the keylevers to operate.

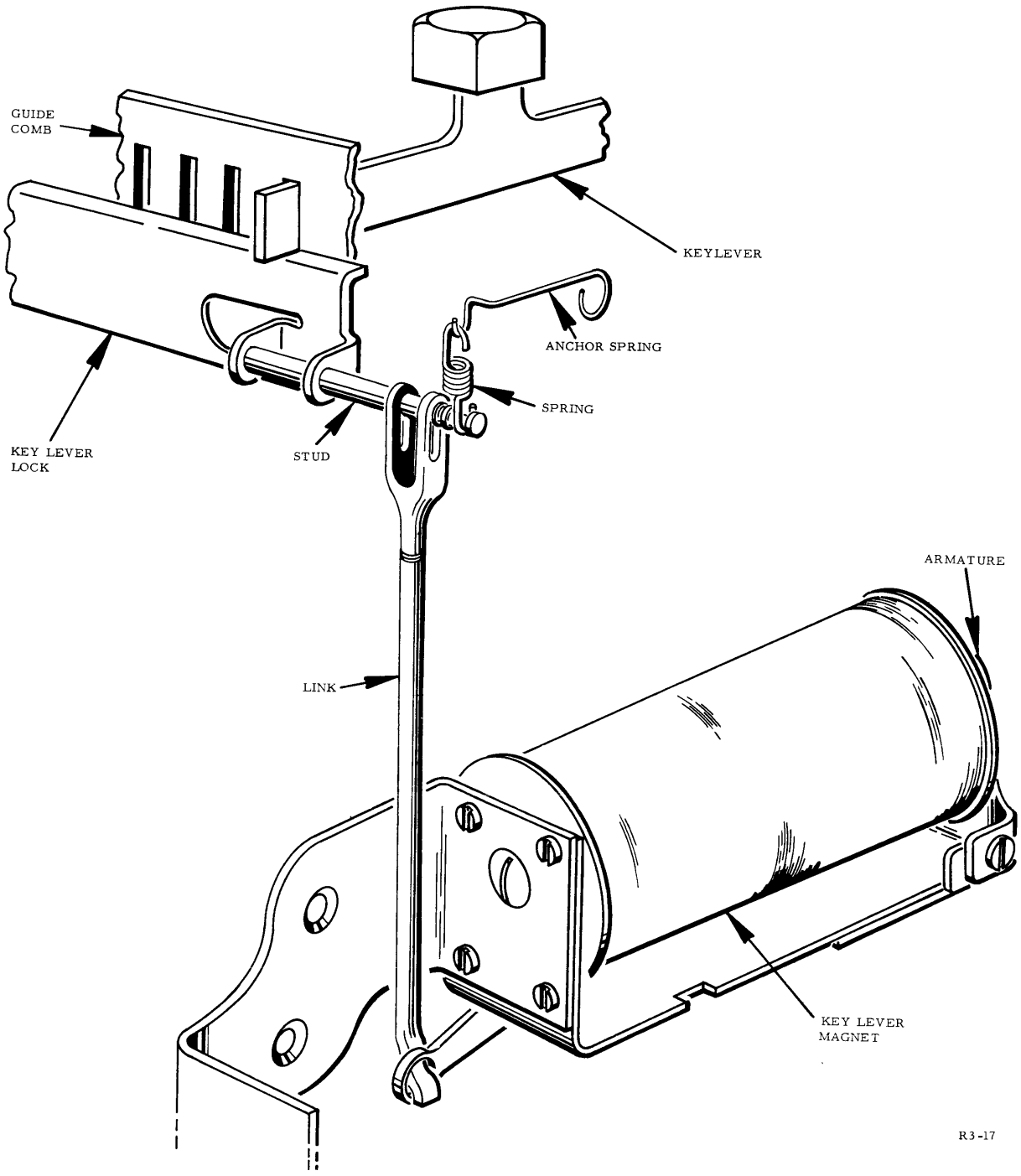


Figure 6-58. Keylever Lock.

Universal Bar Mechanism Operation

The universal bar, mounted to the type bar segment, allows the carriage to move at each operation of a type bar or the space bar by transferring motion from these bars to the escapement mechanism (figure 6-59). Shape of the type bar guide slots in the type bar segment and the location of the

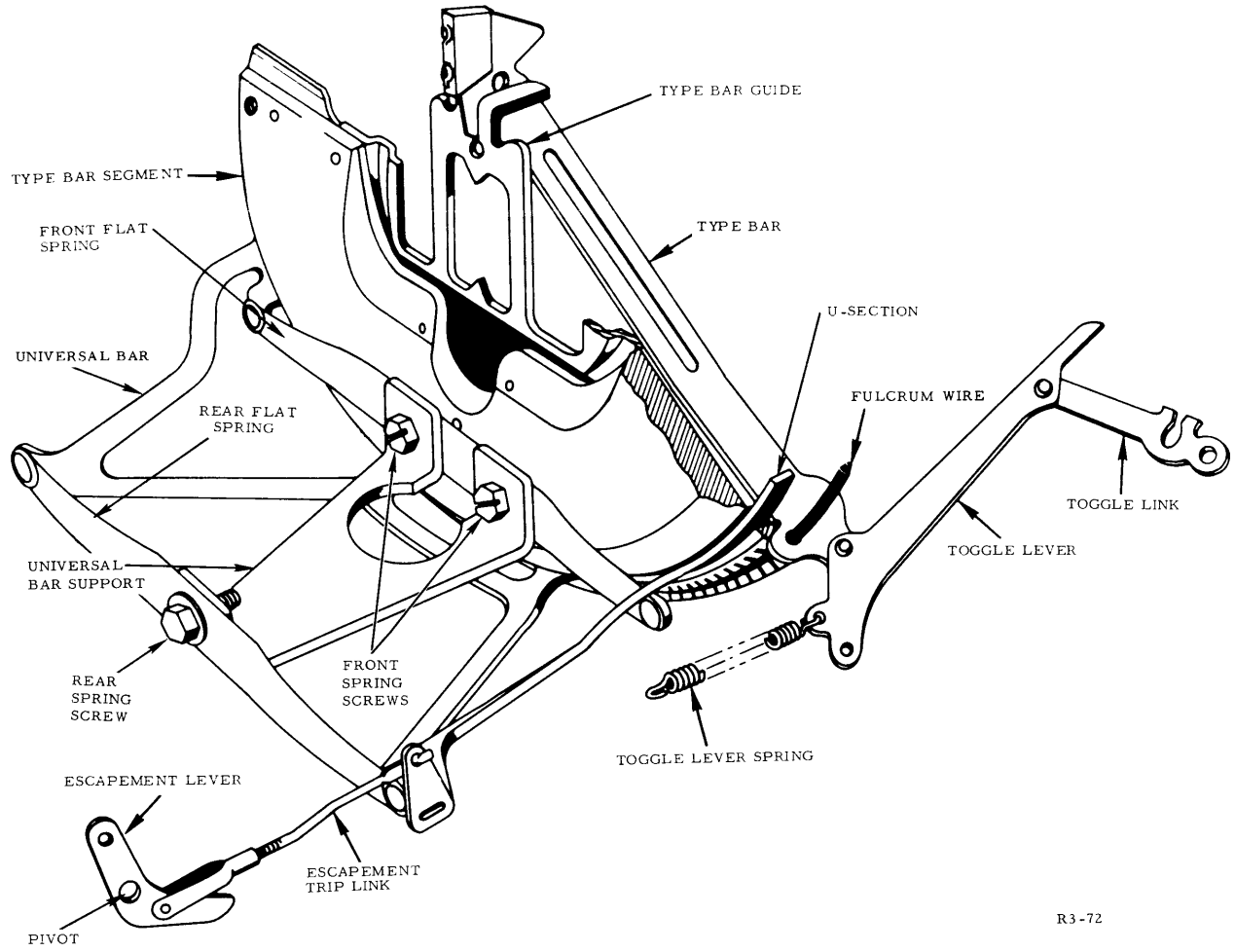


Figure 6-59. Universal Bar Mechanism.

fulcrum rod are such that the forward edge of a type bar, upon contacting the universal bar, causes the type bar to be moved slightly against its spring tension. This motion is then carried by the universal bar through the escapement link to the escapement lever.

Carriage Return

Operation of the carriage return keylever returns the carriage to the left hand margin and indexes the platen. Carriage return is accomplished by a clutch which winds up a carriage return tape. Because the tape is connected to the carriage at the line space hook lever, the tape operates the platen before pulling the carriage to the right.

When the carriage return keylever is depressed, the carriage return cam is released and engages the power roll. A downward pull is exerted on the carriage return link (figure 6-60). This motion is transferred to the return lever, causing it to move downward, and contact the return arm which is

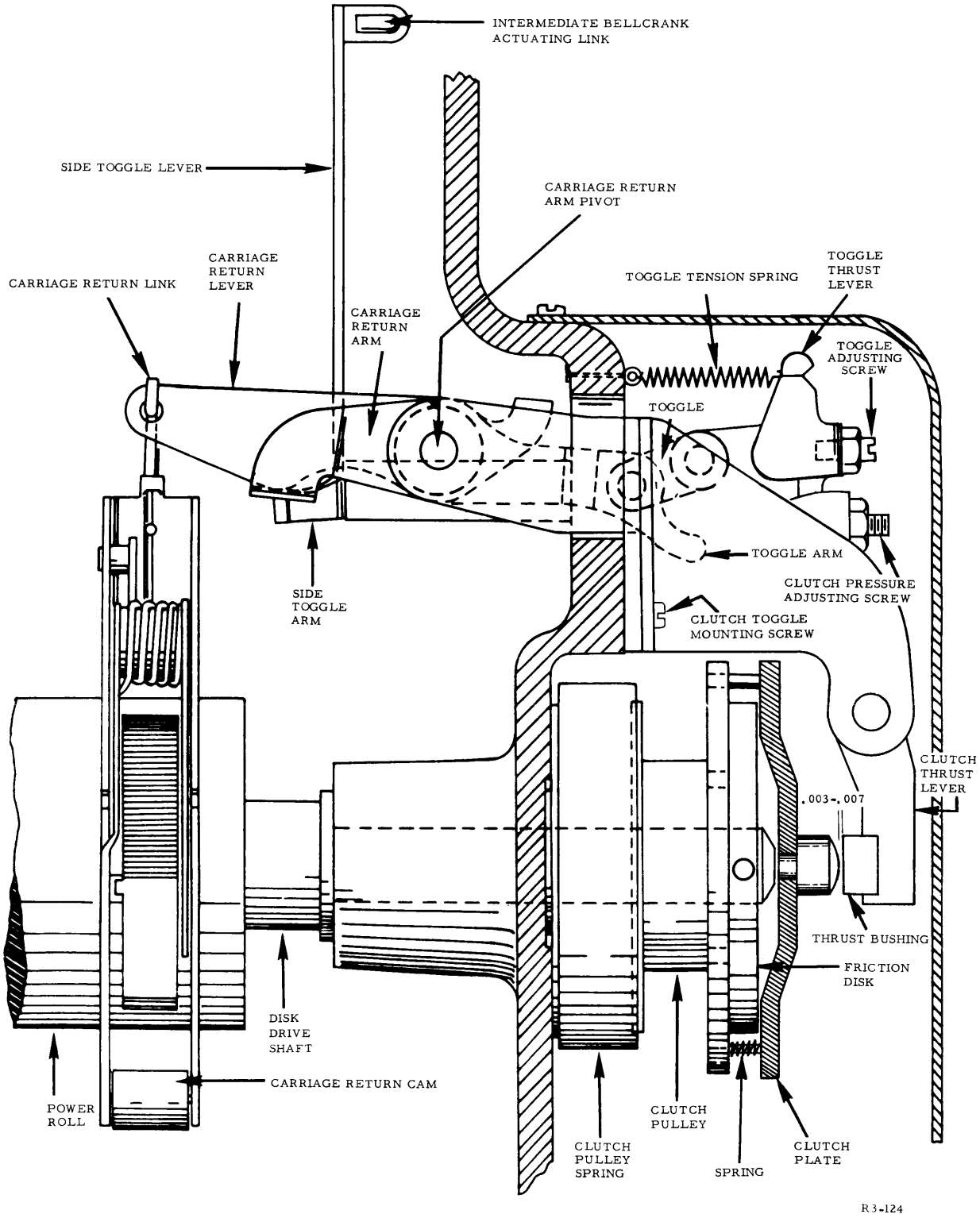


Figure 6-60. Carriage Return Clutch.

rotated counterclockwise on its pivot. Pivoting causes the opposite end of the arm to rotate upward, locking the toggle. A tension is thus exerted on the toggle tension spring which holds the toggle in the locked position.

Motion of the toggle arm brings the arm in contact with the upper toggle adjusting screw, causing the thrust arm to be rotated clockwise on its pivot, and exerting a pressure against the thrust bushing. This pressure is carried through to the clutch plate and friction disk. The friction disk is constantly rotating, being driven by the disk drive shaft extending from the power roll. When the clutch plate engages the rotating friction disk, the plate rotates and, because the pins on the plate engage the clutch pulley, transfers the motion to the pulley. The carriage return tape, being attached to the pulley, is then wound, which pulls the carriage to the right.

When the clutch is not operating, the clutch pulley spring exerts a tension against the pulley rotation, which holds the carriage return tape taut. In addition to operating the toggle, the motion of the carriage return arm also moves the toggle knockout lever downward on its pivot. This action moves the upper extension of the knockout lever and the toggle knockout link forward. The motion of the toggle knockout link rotates the intermediate toggle knockout lever. This position is then held because of the clutch toggle action explained previously.

The motion of the intermediate toggle knockout lever exerts a pull on the rear toggle knockout link, pulling the carriage return bail stop arm lever, which in turn, moves the slotted link. However, the slotted link does not move far enough to rotate the unlatching bellcrank (figure 6-61).

Motion of the carriage caused by the clutch winding the carriage return tape, moves the margin rack in the direction indicated (figure 6-61). The margin stop strikes the margin release lever and moves it in the same direction as the margin rack. This movement of the margin release lever in turn, rotates the cam operating lever which pivots at its center and exerts a push on the unlatching trip link. The push on the link moves the unlatching trip lever and releases the carriage return and tab unlatch cam. Release allows the unlatch cam to engage the power roll and rotate the unlatching bellcrank through the unlatching cam lever, operating link, and operating lever. The motion of the unlatching bellcrank provides a backward pull on the slotted link, moving the carriage return bail stop arm lever which, in turn, exerts a backward pull on the following linkage: rear toggle knockout link, intermediate toggle knockout lever, toggle knockout link and the toggle knockout lever. This backward pull unlocks the clutch toggle, releasing the pull on the carriage return tape.

Tabular Operation

The tabular mechanism permits the operator to place the printout in accurate orderly columns at a minimum of two spaces between columns. When the tabular keylever is depressed, the tab cam is released, engaging the power roll. Leverage developed by this engagement is transmitted by the tab operating link to the left of the rear rail. This action exerts a pull on the tab operating pull lever which pivots on a stud that, in turn, pulls downward on the tab operating pull link (figure 6-62).

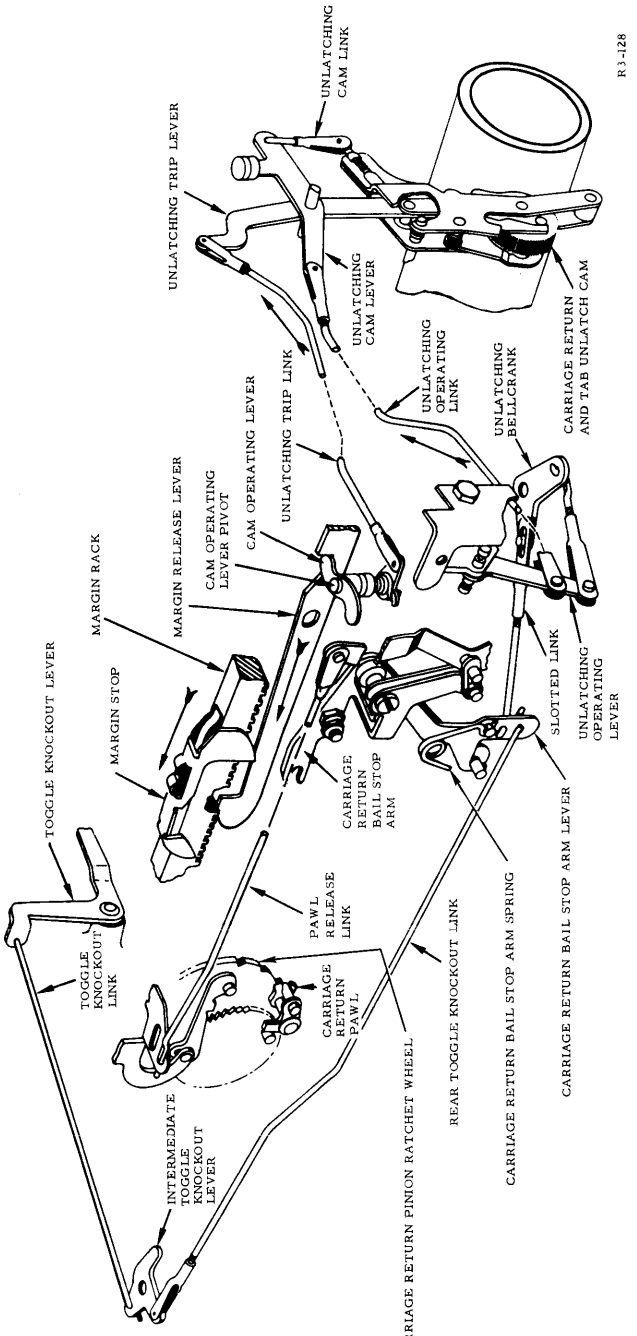


Figure 6-61. Carriage Return Linkage.

Attachment of the tab operating pull link to the tab operating lever results in transmission of motion to the tab lever. Transmission is accomplished by a short downward extension arm on the tab operating lever, which is provided with a pin connected by a spring to the tab lever pin. A guard plate, loosely pivoted on the tab lever pin and slotted to embrace the extension arm pin, is placed between the extension arm and the spring. Opposite the front or upper face of the tab lever is a similar spring which causes the tab operating lever and the tab lever to normally move in unison. However, when the striking edge of the tab lever hits a tab stop, the connecting spring stretches and prevents damage to the parts.

When the tab operating lever is rocked clockwise, the pawl release engaging pin engages and rocks the tab pawl release arm counterclockwise, thereby pushing the pawl release link to the left and operating the pawl release arm. The pawl release arm, in turn, operates the pinion disengaging arm, which disengages the pinion from the escapement mechanism. This allows the carriage to move freely under the tension of the main spring.

The tab operating lever is held in the operated position by a tab lever latch (figure 6-62). When the tab operating lever is pulled downward, the tab-lever latch snaps over the upper edge of the lever and prevents both the tab lever latch and the tab lever from returning to the normal position. The tab lever disengaging spring tends to restore the levers as a unit to the normal position, but the tab-lever latch prevents this action.

When the tab lever strikes a tab stop, a tab lever unlatching operation takes place. This unlocks the tab operating lever and allows the carriage return pawl to engage the carriage return pinion ratchet wheel, thus holding the carriage in that particular tab position. This is accomplished by the horizontal slotting of the tab lever where the tab lever stud passes through the tab lever. Therefore, when the tab lever strikes a stop, the tab lever moves to the right, moving the cam operating lever in a counterclockwise direction. Movement causes an arm, which is part of the cam operating lever, to be similarly rotated, thereby pushing the unlatching trip link to the right and moving the unlatching trip lever. This action allows the carriage return and tab unlatch cam to engage the power roll. The resultant movement of the cam actuates the unlatching cam lever, drawing the unlatching operating link and moving the unlatching operating lever. A lug, which is part of the unlatching operating lever, rocks the tab lever latch to disengage it from the tab operating lever. Thus, the tension of the tab lever disengaging spring disengages the tab lever from the tab stop, and returns the tab lever to the non-operated position.

A rebound check lever is employed when tabulating to keep carriage rebound to a minimum. As the upturned end of the tab lever engages a tab stop, the hooked portion of the rebound check lever rides under and behind to the tab stop. Thus, the upturned end of the tab lever engages one face of the tab stop, while the hooked end of the rebound check lever engages the opposite face. Therefore, carriage rebound is effectively minimized during tab operations.

Tab Governor Operation

The tab governor retards the motion of the carriage during tabulation with

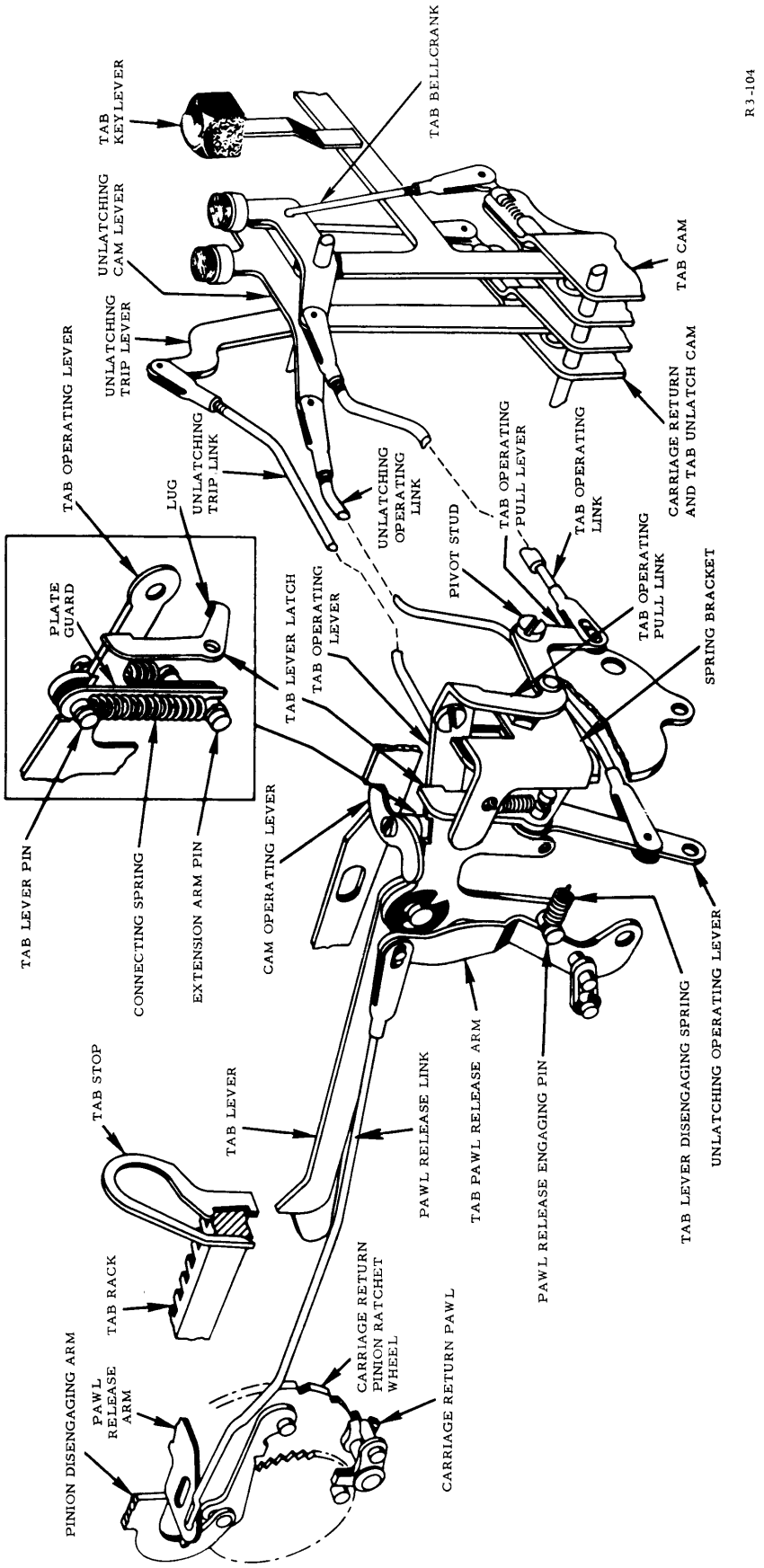


Figure 6-62. Tabular Cam Linkage.

the clutch spring (helical) providing the braking action. When the machine is running, the belt rotates the governor clutch hub, causing the clutch spring to rotate on the hub. During normal escapement operation, the carriage moves with no restrictions. However, on a tabular operation, the carriage moves only as fast as the governor clutch hub is rotating. When the speed of the carriage is equal to the speed of the governor clutch hub, a braking action occurs as a result of the helical spring grabbing on the clutch hub. Machine operation is necessary before the carriage can be moved to the left (refer to Escapement Operation).

Space Bar Operation

The space bar mechanism (figure 6-63) operates the escapement mechanism for letter spacing without printing characters. Depressing the space bar releases the cam of the space bar cam unit, allowing it to engage the power roll. Leverage developed on the cam frame pulls the bellcrank link which, in turn, pulls a dummy bellcrank downward, striking the escapement bail. The upper portion of the bail operates in a manner similar to a bell crank. Movement of the bail through the toggle lever link moves the dummy type bar in the segment toward the platen and operates the universal bar and escapement mechanism in the same manner as the other type bars.

Because there is no type head or slug on the space type bar, no character is printed. Solder is added to give the bail the necessary weight for operation.

Backspacing

With each depression of the backspace key, the backspace mechanism (figure 6-64) moves the carriage to the right one space. When the backspace keylever is depressed, the backspace cam is released, engaging the power roll. The pivot movement of the cam pulls the backspace bellcrank actuating link, which rotates the backspace bellcrank. A backspace connecting link connects one leg of the backspace bellcrank to the backspace pawl. This link causes the rotating movement of the bellcrank to move the backspace pawl downward, engaging a tooth of the escapement pinion ratchet wheel. The pawl rotates the ratchet wheel until it strikes the pawl stop. At this point, the carriage return pawl engages the next tooth of the ratchet wheel and holds the carriage in its new position (figure 6-65).

Movement of the backspace bellcrank also actuates two sets of contacts (SBS). Opening of contacts 1 and 2 de-energizes the keylever lock magnet (LKL), locking the keyboard until the completion of the backspace operation. Contacts 3 and 4, upon opening, break the output ready (O_r) signal to the computer.

A detent pawl prevents backward rotation of the escapement mechanism, thus causing the carriage return ratchet to turn in one direction only. The detent pawl assembly is mounted on the rear carriage rail.

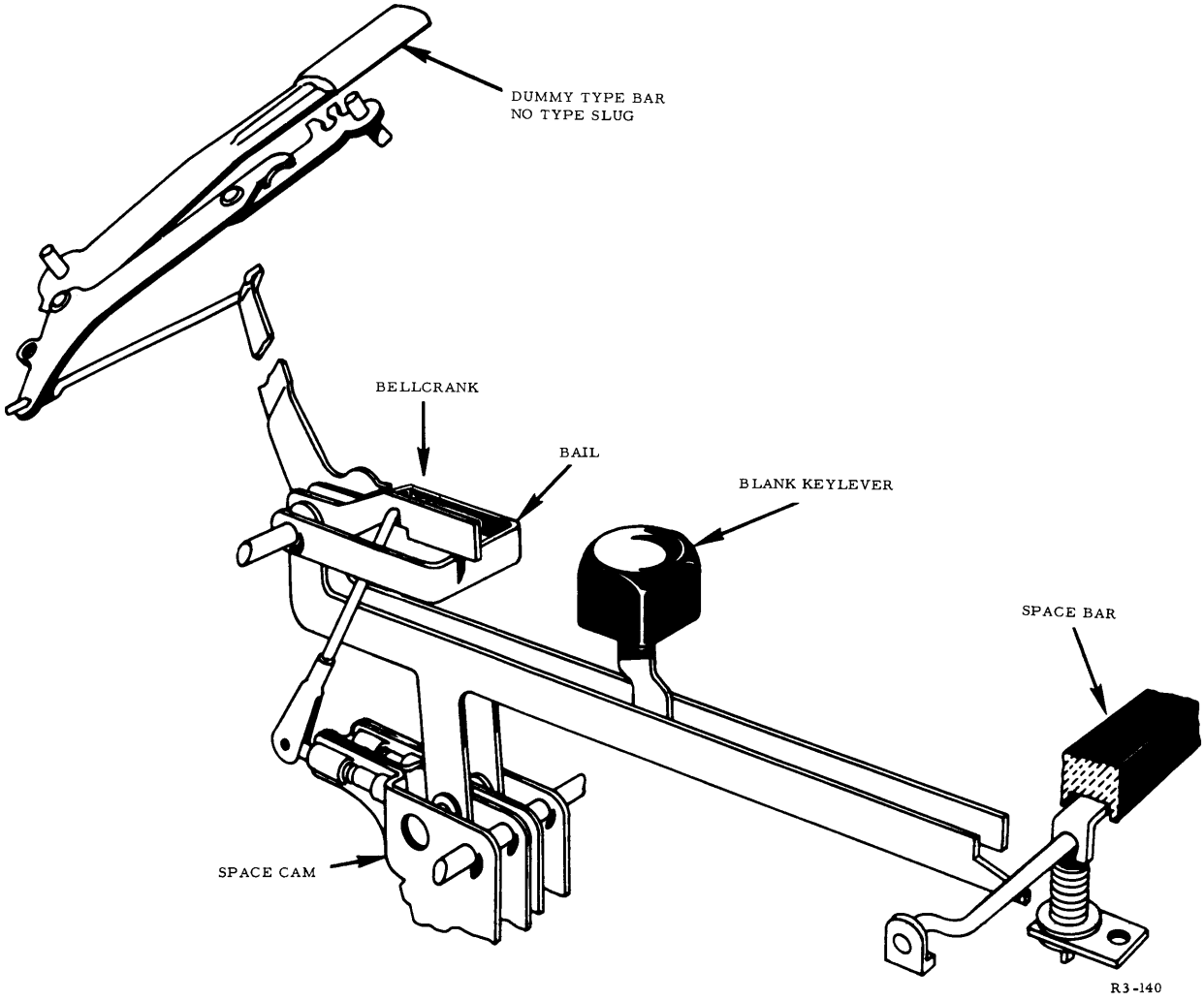


Figure 6-63. Space Bar Mechanism.

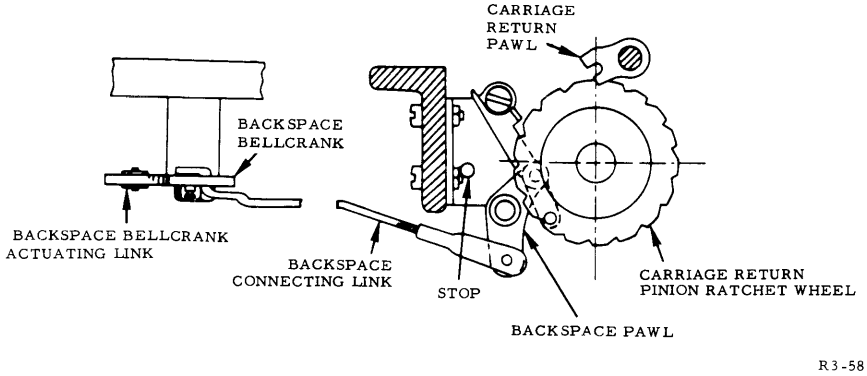


Figure 6-64. Backspace Mechanism.

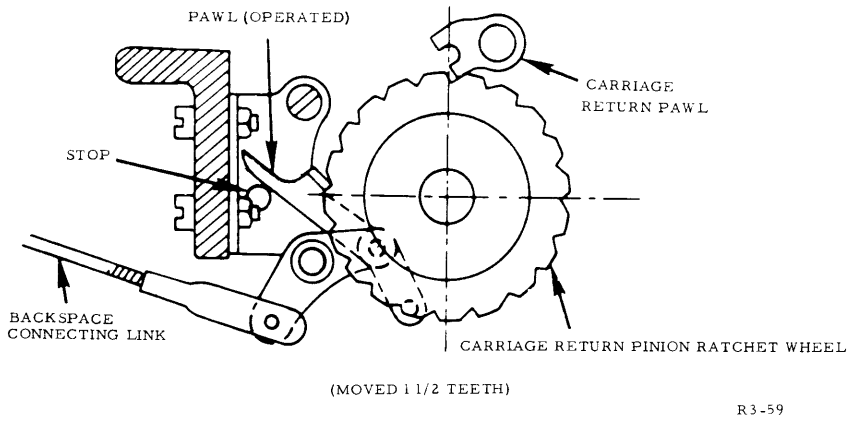


Figure 6-65. Carriage Return Pawl Engaging Halftooth.

Ribbon Feed and Direction Reversal

When any character key lever is depressed, the tail of its associated bellcrank operates a ribbon lift bar which extends across the width of the power frame (figure 6-66). A ribbon feed lever at each side of the machine is pivoted on the key lever bearing wire and has a lug bent at a right angle on each end. The lower lug engages a roller on the ribbon lift bar and the upper lug carries the ribbon feed pawl which, in turn, engages the teeth in the ribbon spool. A ribbon feed check pawl mounted directly under the feed pawl on a fixed pivot also engages the spool. The rocking motion of the ribbon feed levers causes the engaged spool to feed two teeth at a time; the check pawl prevents reverse motion of the engaged spool.

Reversal of feed direction is accomplished by the ribbon reverse bar and pawls. Each end of the ribbon reverse bar contains a pin with a ribbon reverse pawl freely mounted to the pin. The pin itself controls the position of the feed and check pawls such that only the pawls for one spool can be engaged at any one time. The rear end of the reverse pawls is forked and supported by a stud on a two-piece ribbon reverse lever. When the ribbon is unwound from one spool, a continued pull on the ribbon draws the lever toward the rear and raises the front end of the reverse pawls such that the hook of one engages the bent lug on the ribbon feed pawls. As these pawls move forward they pull the ribbon reverse bar such that the feed pawls which were engaged are thrown out of engagement and the ones on the opposite spool become engaged, thus changing the direction of the ribbon feed. A hairpin spring holds the ribbon reverse bar in either position (figure 6-67).

Letter Shift Mechanism Operation

The type basket shift raises and lowers the type basket to change between printing of upper and lower case characters. When the basket is in the raised position, the lower case characters (small letters) print; when in the lowered position, the upper case (capital letters) print.

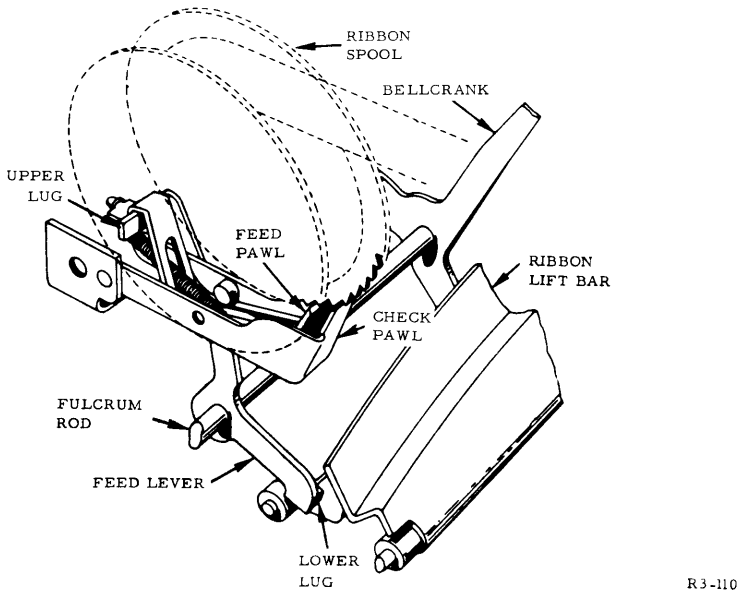


Figure 6-66. Ribbon Feed and Check Pawls.

To effect a basket shift, two double lobe letter cams are used, one for the shift down, the other for the shift up. The shift-down cam is located in the front row of cams, the first cam on the left side; the shift-up cam is located in the front row of cams, first cam on the right.

The keylever marked UPPER CASE, located on the left side of the keyboard, operates the shift-down cam directly. However, the UPPER CASE keylever on the right side of the keyboard, operates the shift-down cam through an equalizing rod extending from one side of the machine to the other. The shift-up cam is operated directly from the right keylever marked LOWER CASE, while the LOWER CASE keylever on the left side operates the shift-up cam through its associated equalizing rod. Thus, each equalizing rod transfers keylever motion from one side of the machine to the other.

When either of the shift keylevers marked UPPER CASE is depressed, the shift cam at the left side of the power roll operates. As the cam rotates, a pull is exerted on the left toggle release lever, which pivots on the ribbon lift bar fulcrum wire bearing. While the toggle release lever pivots, its upper inturned finger, which abuts the toggle lever, moves the toggle lever in a clockwise direction (figure 6-68). As the toggle lever pivots, it disengages the toggle linkage consisting of the bracket operating lever, the toggle lever, and the rock shaft operating lever. The bracket operating lever exerts a downward push on the bracket such that the basket is started down toward its upper case position. Downward motion of the bracket causes the rock shaft operating lever to move the rock shaft and thereby transmit the motion to the opposite side of the basket, resulting in a parallel basket motion.

The tension of the basket tension spring holds the basket in the up position; but as the basket moves downward, the tension of the spring decreases until the spring reaches a dead center position in which it no longer has a holding effect on the bracket operating lever. However, a similar spring on the right side simultaneously increases tension on the right bracket operating lever (which is reversed). This spring moves from a dead center position to an

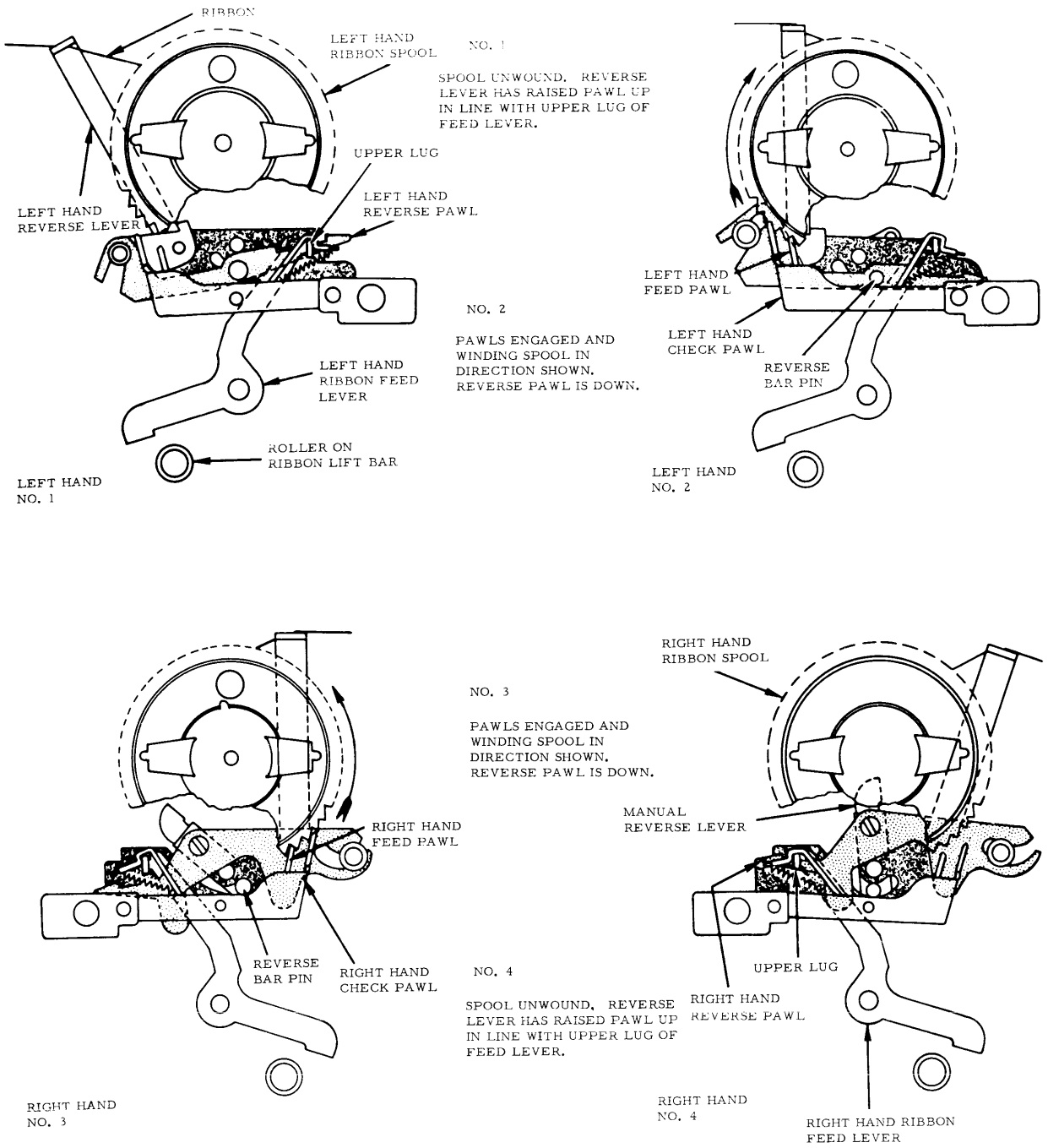


Figure 6-67. Ribbon Reverse Mechanism.

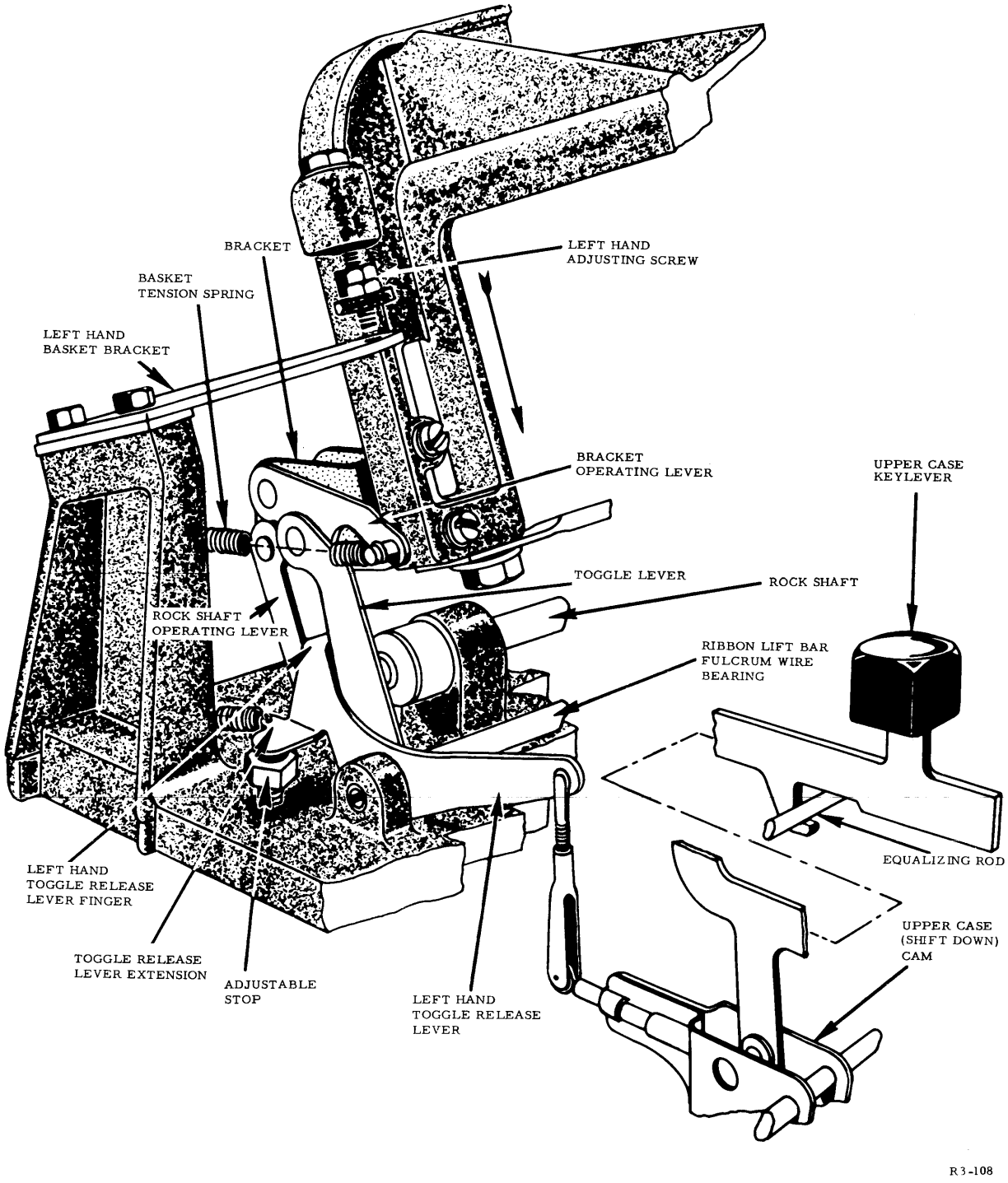


Figure 6-68. Shift Mechanism -- Lower Case Position.

effective position on its operating lever such that the basket, when it reaches the down position, is held there under substantial tension of the basket tension spring (figure 6-69).

Adjustable stops limit the amount of movement of the basket travel. Limitation of movement is accomplished by the toggle lever striking the inturned finger of the toggle release lever, which in turn, applies pressure to the release lever extension. The stop, therefore, prevents the lever from moving further.

To return the basket to the up position, either of the shift keylevers marked LOWER CASE, when depressed, operate the right shift cam. The same action previously explained takes place except that the toggle is disengaged on the right side while the left toggle holds the basket in the up position.

Paper Feed Operation

Paper is fed by the platen (when it is rotated) through pressure maintained against the plate by feed rolls mounted in deflector yokes under the platen. Compression springs, supported by screws in the carriage base, provide the necessary lift, assisted by a torsion spring attached to the left end of the feed roll actuating arm.

Release of the tension is accomplished by moving the paper release lever toward the front. This action moves a toggle lever assembly on the left end of the carriage such that the feed roll actuating shaft rotates downward toward the rear. The feed roll actuating arm is then lowered, causing the deflector and feed rolls to drop away from the platen. (figure 6-70).

A line gage card holder (figure 6-71) holds cards and multiple copies close to the platen, and also aids in aligning copies to strike a correction in the exact position of the original character. The line gage further enables an operator to locate a desired writing line.

Line Spacing

Line spacing is accomplished by the initial pull on the carriage return tape operating the platen indexing mechanism. Pull of the tape is transmitted through the hook lever assembly to the index pawl carrier assembly which causes the pawl to enter a platen ratchet tooth and rotate the platen (figure 6-72). Provision is made for spacing different numbers of ratchet teeth with each operation of the line space mechanism by changing the position of the line space lever. The change in the line space lever position, in turn, varies the position of the index pawl which then allows the pawl to enter the ratchet at a different point. An adjustable stop is provided to prevent overthrow. No manual carriage return and line space lever is provided.

Carriage and Rails

The carriage assembly provides support and movement to the platen

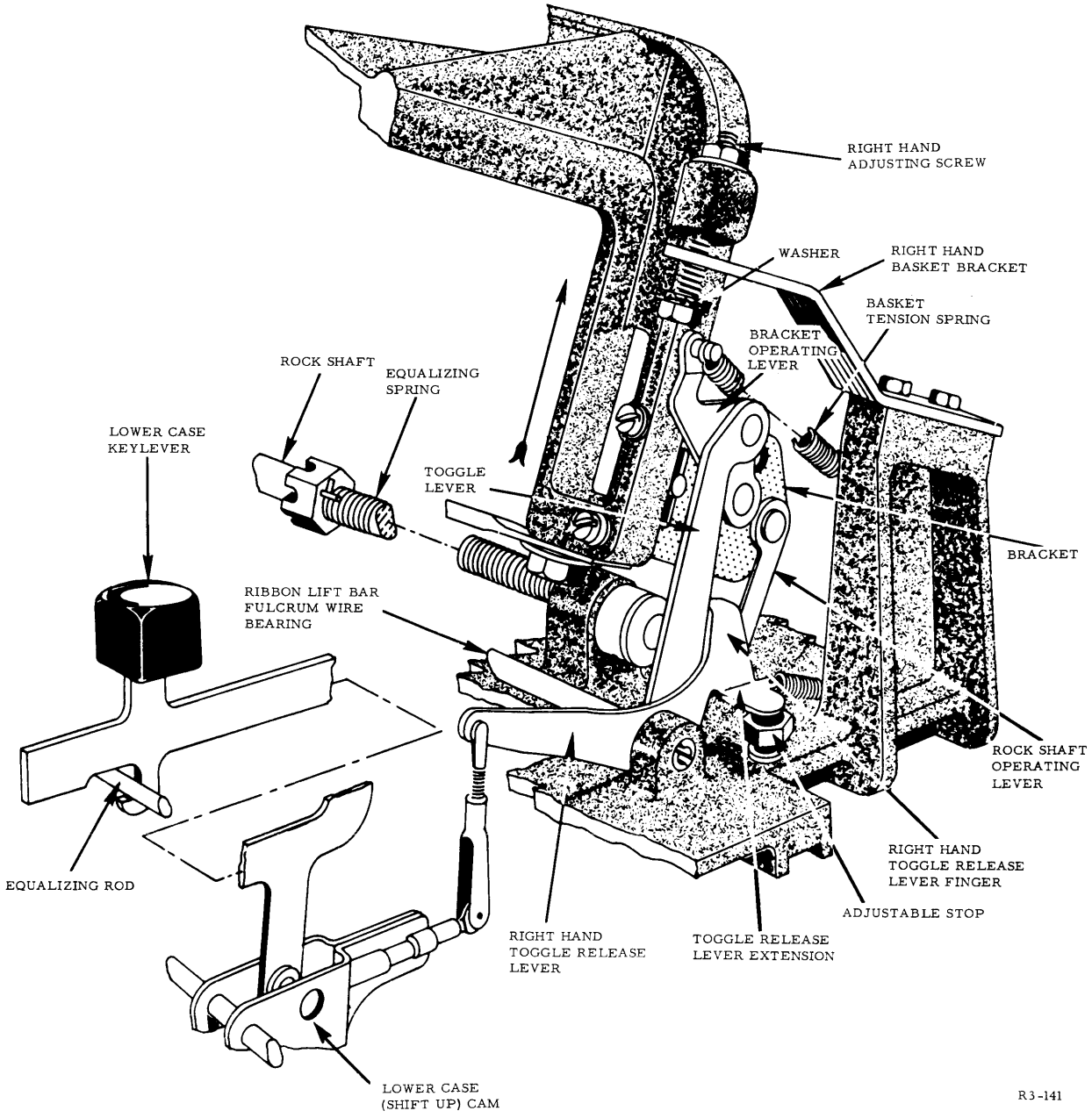


Figure 6-69. Shift Mechanism -- Upper Case Position.

such that it holds the paper in correct relation to an operated type bar. Front and rear rails provide the means by which the carriage may be moved laterally with a minimum amount of friction while simultaneously holding the surface of the platen correctly at the printing point.

Figure 6-73 pictures the carriage, rails, and a truck assembly. Six of these trucks are used with the 16-inch carriage. Purpose of the truck assemblies is support of the carriage in the rails. Each truck assembly body contains four steel rollers and a star wheel, or gear. The four surfaces formed by each rail and the side of the carriage adjoining that particular rail, enclose a square opening that constitutes a guide within which the

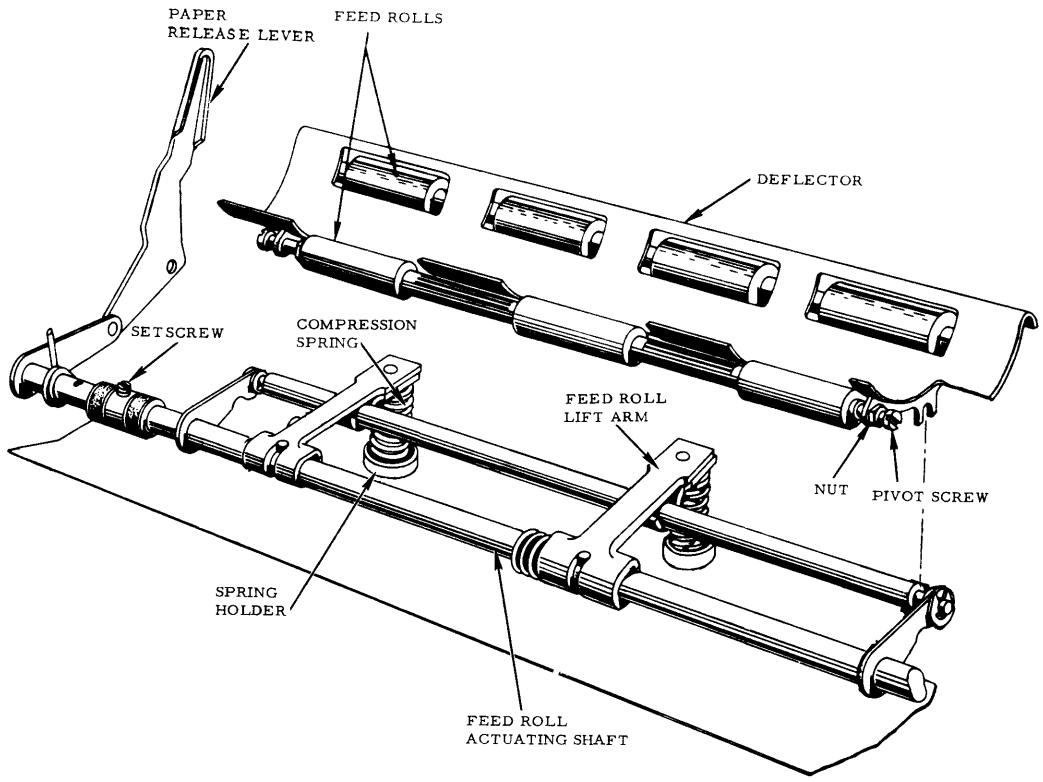


Figure 6-70. Paper Feed Mechanism.

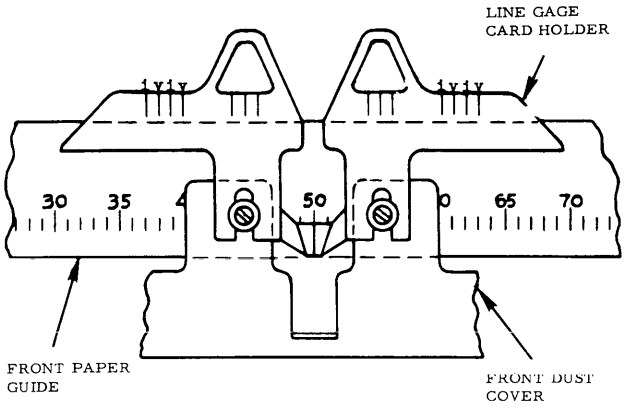


Figure 6-71. Line Gage - Card Holder.

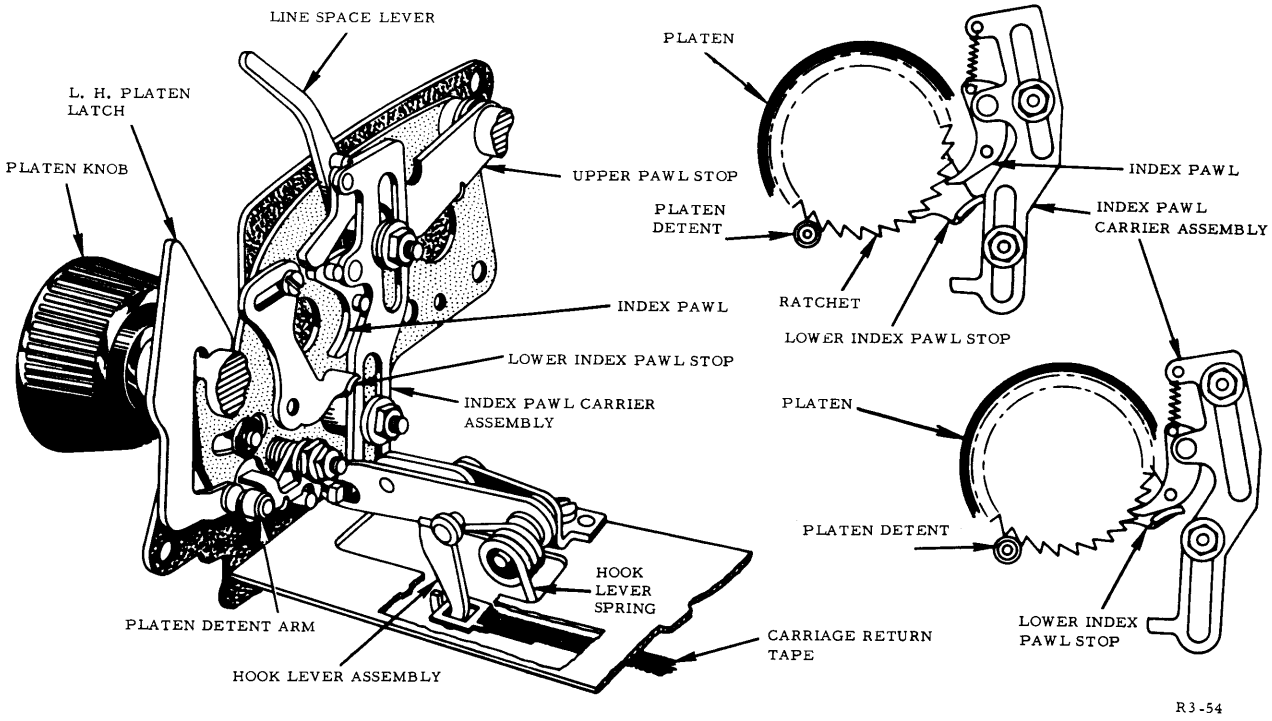


Figure 6-72. Line Space Mechanism.

trucks move. All of these surfaces are ground to a close tolerance.

The four rollers located in the truck assemblies ride along two opposite ground surfaces while the star wheels engage the rack gears which are an integral part of both the rails and carriage. Action of the star wheels is such that the trucks always move to give maximum support to the carriage in any position. Thus, the carriage is supported on all twenty-four rollers, regardless of its position along the rails.

Right and left castings of the base assembly have milled surfaces to hold the rails in true relation to the power frame and type bar segment. The rails are fastened to the base by four screws. Position of the rear rail can be adjusted forward or backward with adjusting screws. This permits adjustment for true lateral motion as well as keeping the rails spaced such that the carriage will not be loose.

Escapement Operation

The escapement mechanism provides carriage support simultaneous with movement of the carriage a uniform distance for each operation of a type or space bar. The escapement used in the RECOMP III Flexowriter is of the mono-spacing type in which all characters, both in upper and lower case have exactly the same spacing. Figure 6-74 shows the escapement mechanism together with the escapement rack and the carriage tension tape. The escapement mechanism is made up of several assemblies which are

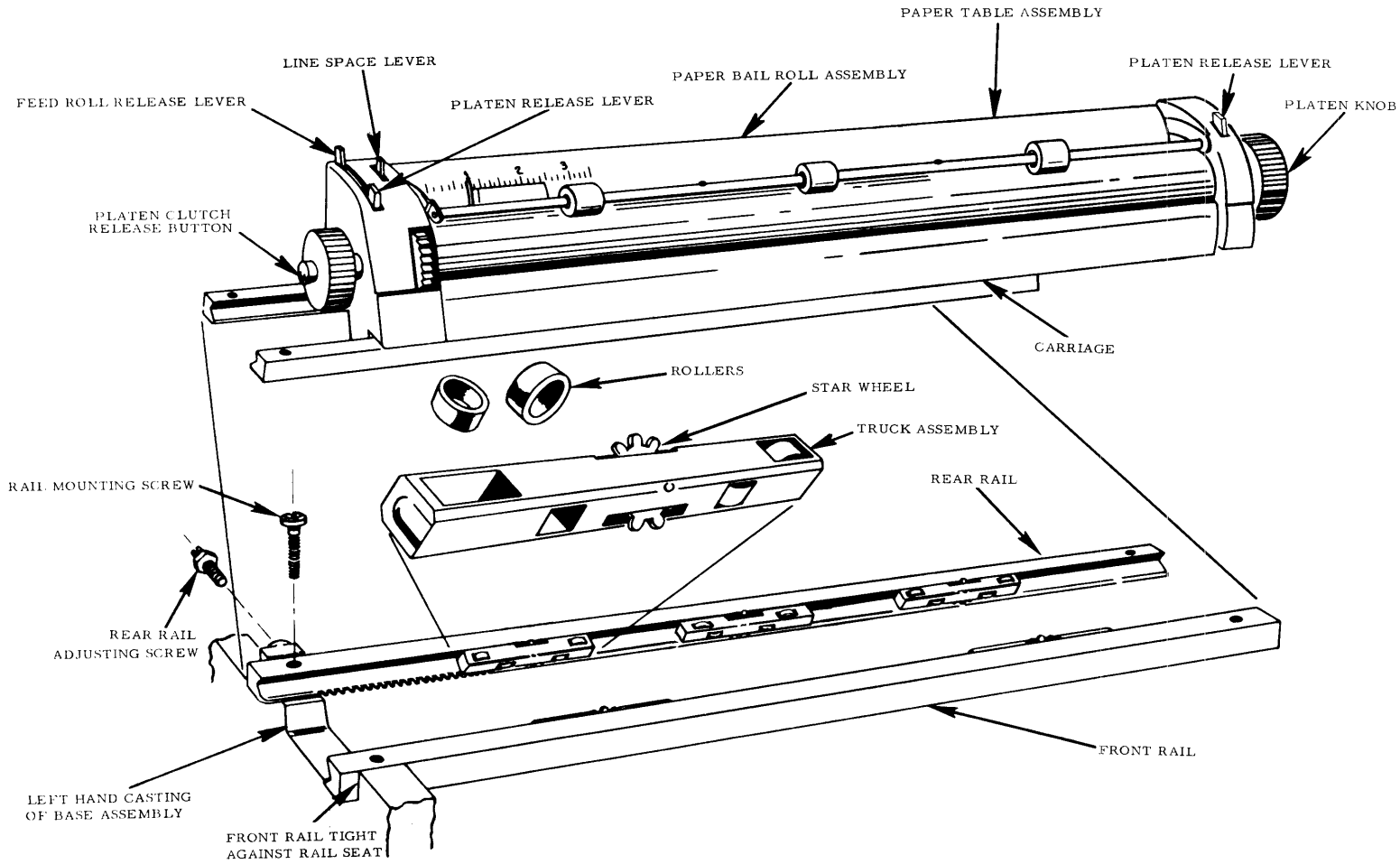


Figure 6-73. Carriage and Rails.

described separately in following paragraphs. These assemblies are:

Escapement Pinion and Ratchet Wheel -- A machined part consisting of a ratchet wheel, an escapement pinion, and a governor drive pinion.

Escapement Wheel-Hub and Carriage Return Ratchet -- A manufactured assembly consisting of an escapement wheel, a release ratchet, and a carriage return ratchet mechanism.

Escapement Rack -- A long rack gear fastened to the bottom surface of the carriage assembly.

Escapement Pawl Carrier -- An assembly consisting of a pawl carrier frame, a movable pawl, and two adjustable pawl stops.

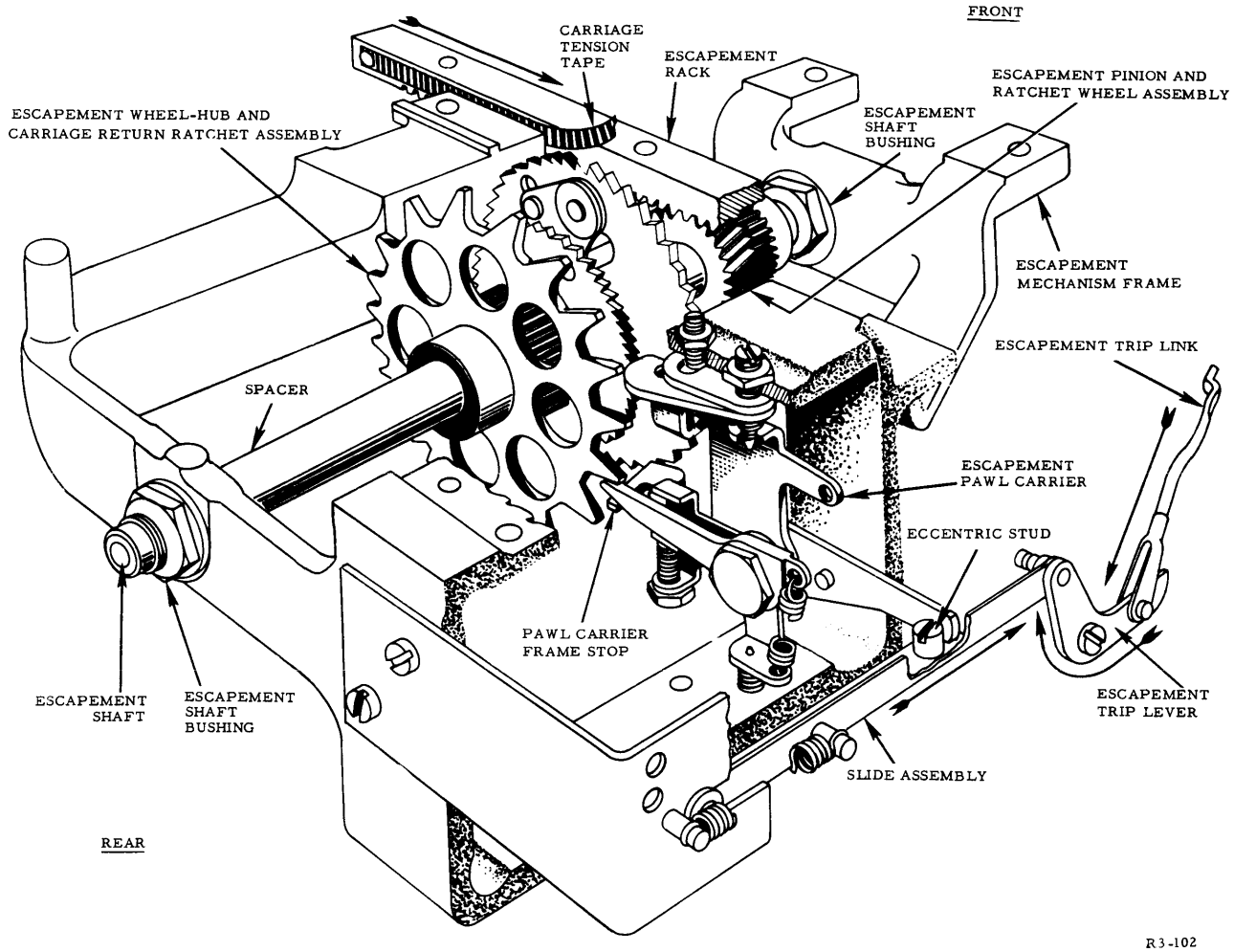


Figure 6-74. Escapement Mechanism.

Escapement Pinion and Ratchet Wheel

The escapement pinion and ratchet wheel works in conjunction with the carriage return pawl on the escapement wheel - hub and carriage return ratchet mechanism in such manner that the carriage can only move in direct relation to the escapement rack. Meshing of the escapement pinion with the escapement rack connects the escapement mechanism to the carriage assembly. The governor drive pinion meshes with a gear of the tab governor assembly to regulate the speed of the carriage during tab operation.

Escapement Wheel-Hub and Carriage Return Ratchet

Control over the amount of carriage escapement for each operation of a type or space bar is provided by the escapement wheel-hub and carriage return ratchet assembly. The release ratchet of this assembly controls the carriage return pawl which is pivoted on the carriage return ratchet mechanism. This mechanism, together with a detent pawl, prevents any reverse rotation of the escapement wheel.

Component parts of the escapement wheel-hub and carriage return ratchet mechanism always move in unison. However, the release ratchet may be moved a few degrees in a clockwise direction (when viewed from the front) without affecting the other parts of this mechanism. Moving the release ratchet in a clockwise direction lifts the carriage return pawl from engagement with the escapement ratchet wheel. This frees the escapement wheel-hub and carriage return ratchet assembly from the escapement pinion and ratchet wheel assembly. Thus, the escapement pinion can rotate in a clockwise direction (carriage movement to the right) during carriage return and backspacing operations, without any movement of the escapement wheel.

Escapement Rack

The escapement rack meshes with the escapement pinion gear of the escapement mechanism; the carriage tension tape is attached to the right end of the rack.

Escapement Pawl Carrier

Mounted on two pivots, the escapement pawl carrier is moved by the trip slide of the escapement mechanism. Carrier side motion is limited by two adjustable pawl stops. The pawl carrier is moved by the trip slide only an amount sufficient to allow the pawl to clear the escapement wheel and move to the pawl's upper stop. Simultaneously, the escapement wheel is kept from moving by a stop which engages one of the wheel's teeth. When the pawl carrier is returned to normal position, one tooth of the escapement wheel engages the pawl and moves the pawl to its lower stop. Pawl movement is always equivalent to a one-tooth-space movement of the escapement wheel. The pawl movement is then carried through the assembly to the escapement rack.

Paper Tape Punch

The tape punch perforates predetermined codes (in paper tape) which represent characters or functions initiated in the reader or keyboard, or received from the computer. This recording of information in paper tape is accomplished by converting electrical impulses received from either the code selector (keyboard), tape reader, or the input-output driver boards (computer).

When a code punch magnet is energized, the resultant upward movement of its armature releases its associated latch lever. Release allows spring tension to pivot the latch lever in a clockwise direction which, in turn, permits the latch lever recess to engage the end of the punch lever. Simultaneously, energization of the punch clutch magnet occurs, and the clutch connects the constantly running drive gear with the punch operating shaft (refer to Punch and Translator Clutches in this section).

As the punch operating shaft rotates, the punch lever frame and pivot rod assembly raises and lowersin accordance with the contour of the punch lever actuating cams. This action results in the corresponding raising and lowering of those punch levers and associated code punches engaged by latch levers (those which are to perforate the tape to represent binary ones). Latch lever engagement of the selected punch levers holds them against the drive arm pivot stud as their opposite ends are moved upward. (The punch lever corresponding to the feed hole punch position, however, is permanently held against the drive arm pivot stud, resulting in a feed hole tape perforation for each punch operating shaft revolution).

Return of the latch levers to their home position begins when a latch lever actuating stud on the inner face of the contact bail operating cam raises the latch lever restoring bail arm. This action initiates a counterclockwise pivot of the latch lever restoring bail, moving the latch levers past their armature latching position. Simultaneously, the latch lever restoring bail operates the knockoff bail to release any armature still adhering to its punch magnet core. Upon completion of the armature knock-off action, the latch lever restoring bail returns to its normal position, thus enabling re-engagement of the latch levers in their respective armature notches.

During the restoring action, the tape feed pawl cam and feed pawl lever arm rotate counterclockwise, which imparts a clockwise motion to the detent gear. Rotation of the detent gear and its associated tape feed sprocket advances the tape in 1/ 10 inch increments. Stabilization of the tape feed mechanism is accomplished by the detent, lever and spring assembly.

Three arms in the tape contact mechanism prevent punch operation when the tape runs out, tears, or binds (figure 6-75). Operation is prevented through closure of the tape punch switch contacts by one of the following means:

Hold-Down Arm -- Lifting of the tape hold-down arm closes the tape punch switch contacts by causing clockwise movement of the tape control bail through the hold-down arm and run-out arm linkage.

Run-Out Arm -- If the tape tears or runs out, the run-out arm drops below the tape table, causing tape control bail clockwise movement.

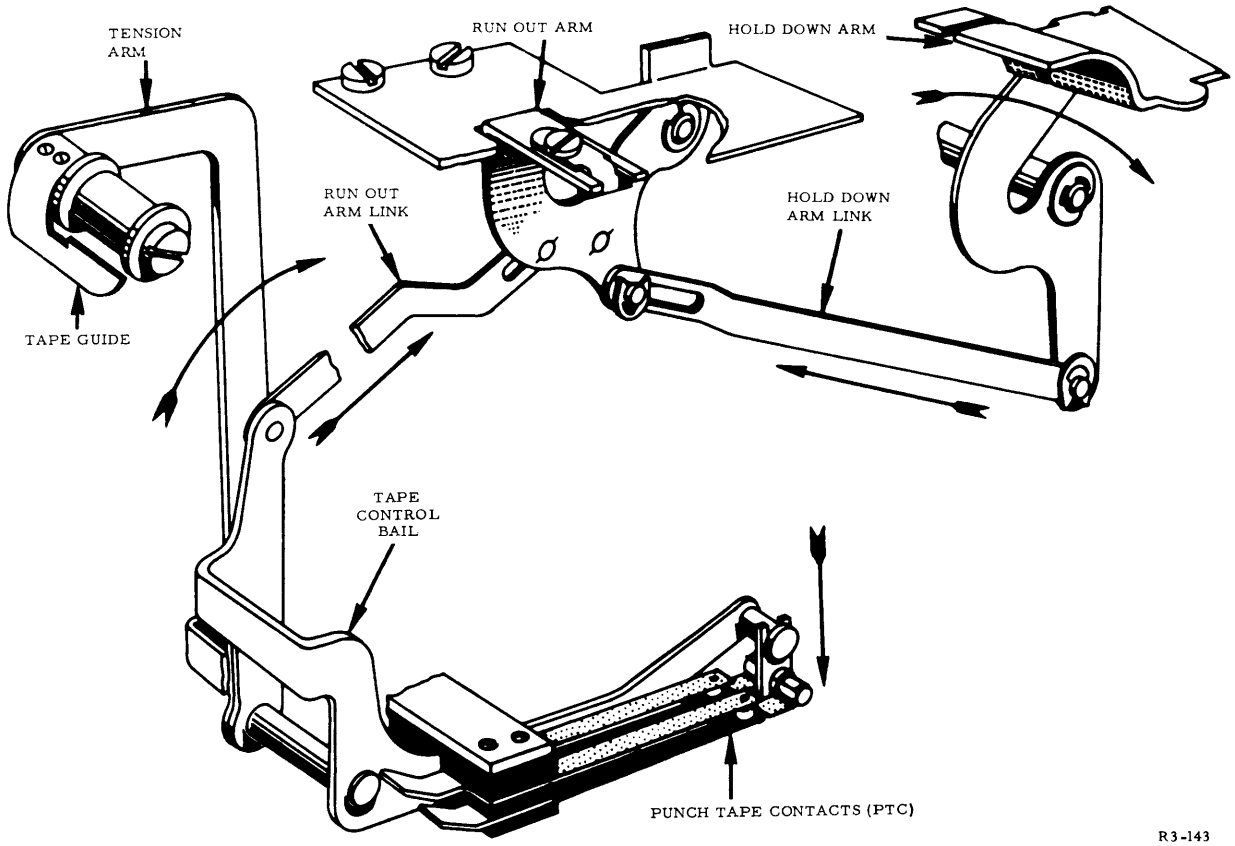


Figure 6-75. Punch Tape Contact Mechanism.

Tape Tension Arm -- If the tape binds during operation, the tension arm moves clockwise, causing tape control bail clockwise movement.

Paper Tape Reader

The paper tape senses the codes in a punched tape and translates the codes into electrical impulses for subsequent routing to either the translator, punch, or computer.

When reader magnet LR is energized, the feeler pins move up toward the tape. If any code holes are over feeler pin positions, the pins move through the holes and release the contact lever, allowing the corresponding pin contact to be operated. The contacts remain closed for 50 milliseconds. Subsequent cam action then lowers the feeler pins below the level of the tape, and the pins are then locked in position until the next cycle. The tape feed pinwheel then advances the tape 0.10 of an inch to complete a reading cycle.

Reader Shaft and Cams

Motive force for reader operation is derived from the Flexowriter main drive system through a continuously rotating shaft with three cams (figure

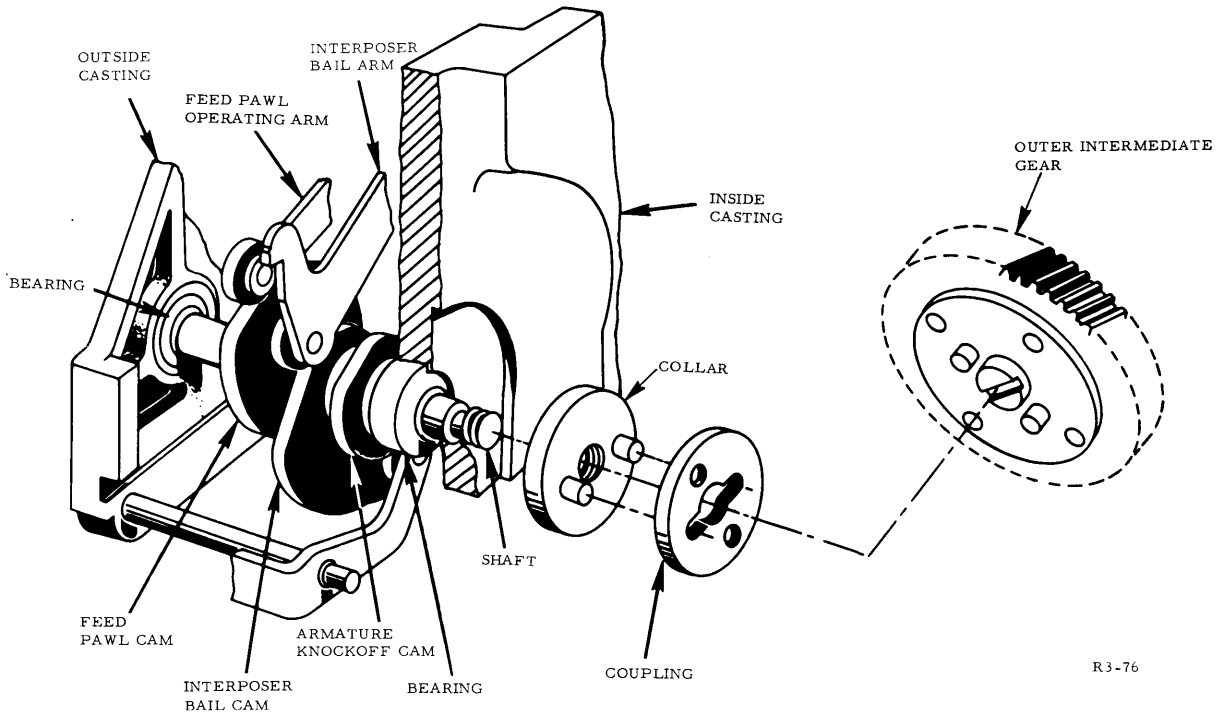


Figure 6-76. Reader Shaft and Cams.

6-76). The cams (interposer bail, feed pawl, and armature knock-off), in turn produce the various reader actions and operation timing.

Reader cycle duration is controlled by the armature knockoff cam (figure 6-77). If the magnet is energized at a point where the follower of the armature knockoff arm is riding on the high lobe of the armature knockoff cam, the armature cannot be moved to the core of the magnet. The high lobe thus prevents the reading cycle from taking place until the follower of the knockoff arm is on the low side of the knockoff cam, at which point the armature can be attracted to the energized magnet.

Feeler Pin Operation

When the armature is attracted to the energized magnet core, the upper arm of the armature moves the control arm in a clockwise direction, releasing the interposer bail arm. The interposer bail arm in following the interposer bail cam contour, then (figure 6-78) rocks and raises the interposer bail assembly. This action causes the interposers to rise because of spring tension of their respective springs, raising the feeler pins in the guide block which then enter any tape perforations (figure 6-79).

As the feeler pins are entering their respective code holes, the associated interposers rise and free the interposer shoulders from blocking the paths of their respective contact levers (figure 6-80). Accordingly, when the interposer studs rise above the contact lever bail rollers, the contact levers,

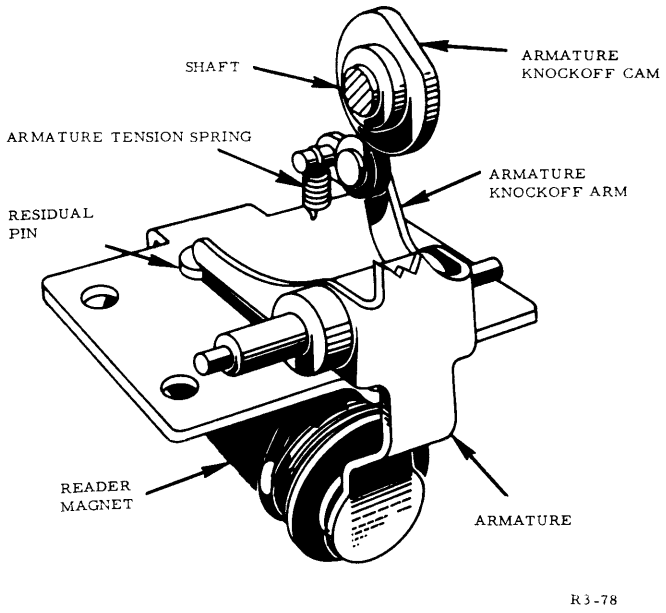


Figure 6-77. Magnet Armature Control.

now freed of the interposer shoulders, move inward under contact strap spring tension and actuate the required reader contacts.

If there is no code hole positioned over a feeler pin, the pin rises to the tape and touches it with a light pressure (figure 6-81). However, because the interposers do not release the contact levers, the reader contacts are not actuated.

As the interposer cam rotates and its high point rides in contact with the cam follower, the interposer arm and bail assembly is pivoted in a counter-clockwise direction on its pivot shaft and the control arm is moved inward by its return spring tension. Inward movement of the control arm allows it to engage the latch end of the interposer bail arm, thus keeping the bail arm from further contact with the interposer bail cam. This action causes the interposer bail to lower the interposer arms into their contact lever blocking position, depress the feeler pins below the tape level, and lower the interposer studs. Lowering of the interposer studs separates the contact lever bails, which restores the reader contacts to the non-operate position.

Tape Feed Operation

The tape feed mechanism feeds the tape the equivalent of one code position for each cycle of the reader. Tape feed occurs during the latter part of the cycle, following the reading of the code. Feeding of tape is under control of the tape feed cam (figure 6-82).

Tape advance occurs after the code feeler pins have returned to their rest position below the tape level. However, the tape advance mechanism

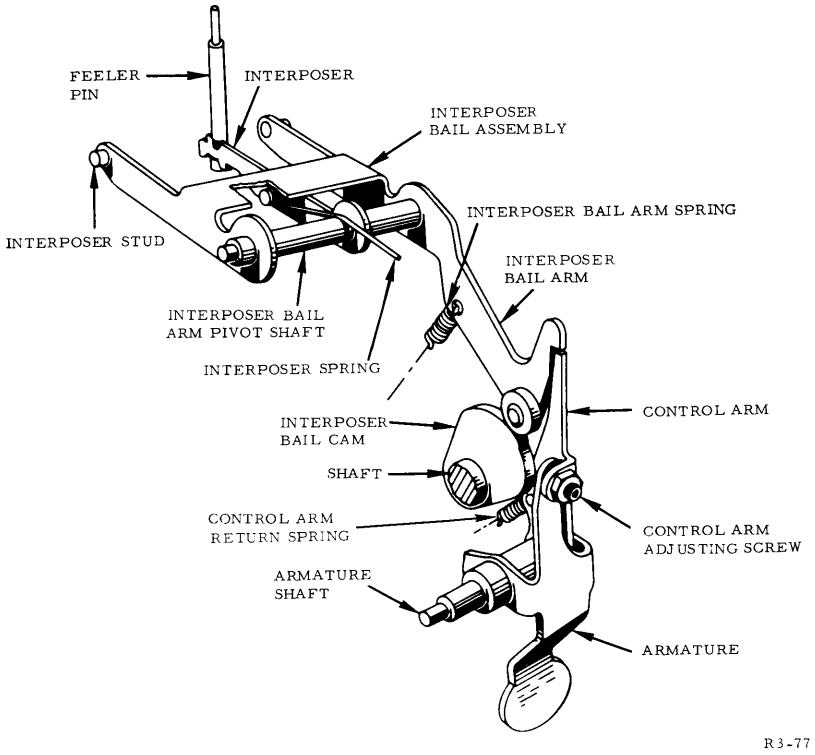


Figure 6-78. Interposer Bail Operation.

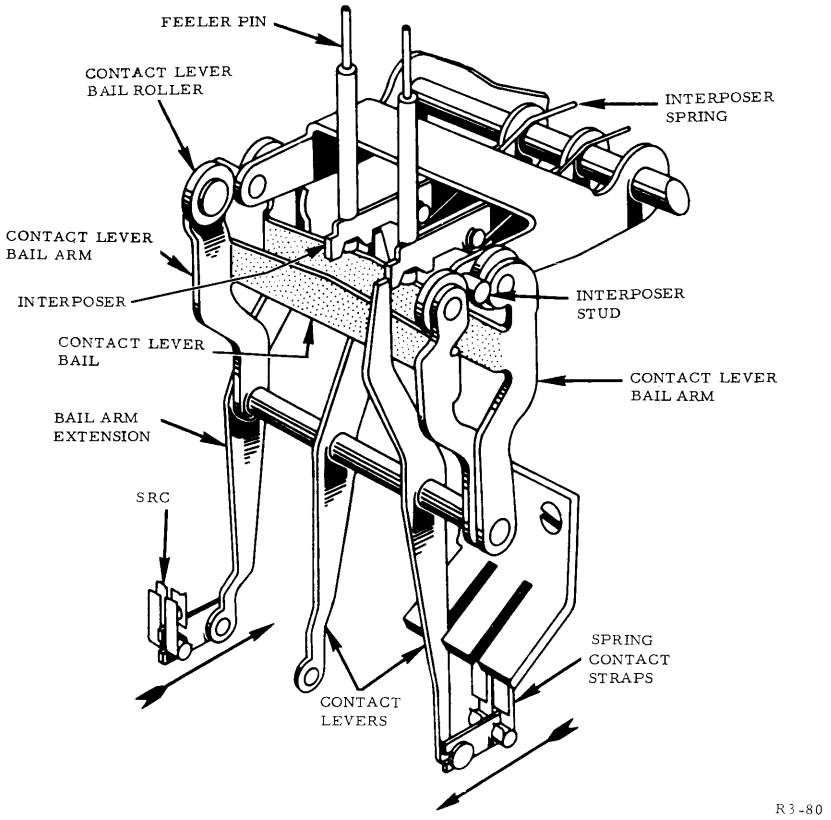


Figure 6-79. Feeler Pin and Contact Operation.

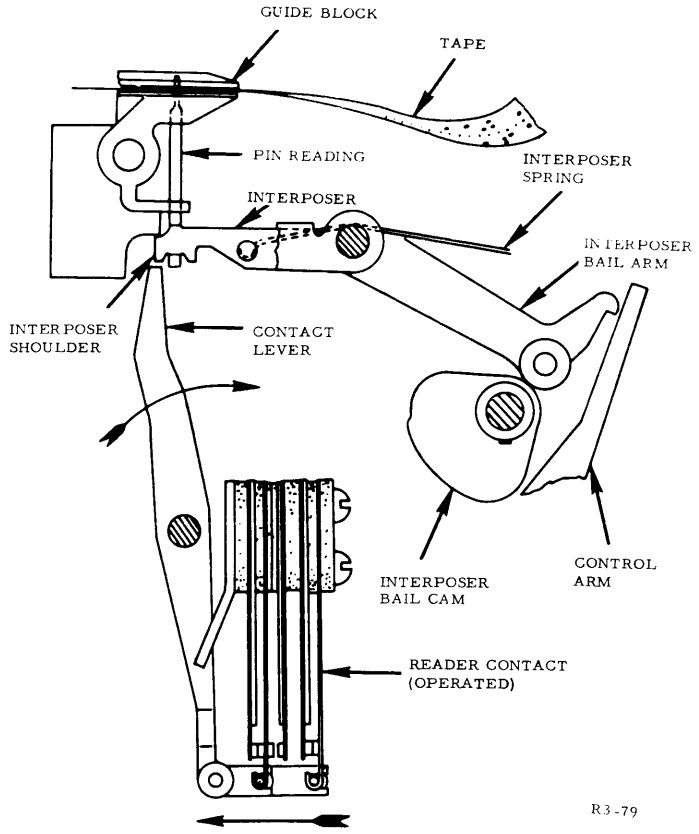


Figure 6-80. Feeler Pin Reading Code.

begins to operate before the control arm latching point engages the interposer bail arm to terminate code hole sensing.

When the interposer stud moves up from between the pin contact lever bails and the bails move toward each other, an auxiliary latch moves to the right. Then, when the cam follower of the feed pawl operating arm momentarily touches the high point of the tape feed cam, the latching end of the operating arm moves downward. The downward movement allows the feed pawl latch to move clockwise under feel pawl latch spring tension. Clockwise movement permits the feed pawl operating arm roller to follow the tape feed cam, thus operating the feed pawl, ratchet wheel, and pin wheel. An adjustable feed pawl stop controls the movement of the feed pawl such that the tape will be moved only one code position for each cycle. A detent assembly holds the feed ratchet wheel in position between tape advancements.

Tape Hold-Down Lever

The tape hold-down lever holds the tape against the pin wheel and guides the tape as it is being advanced (figure 6-83). When the tape hold-down lever is moved away from the hold position, the reader tape contact is opened preventing reader operation.

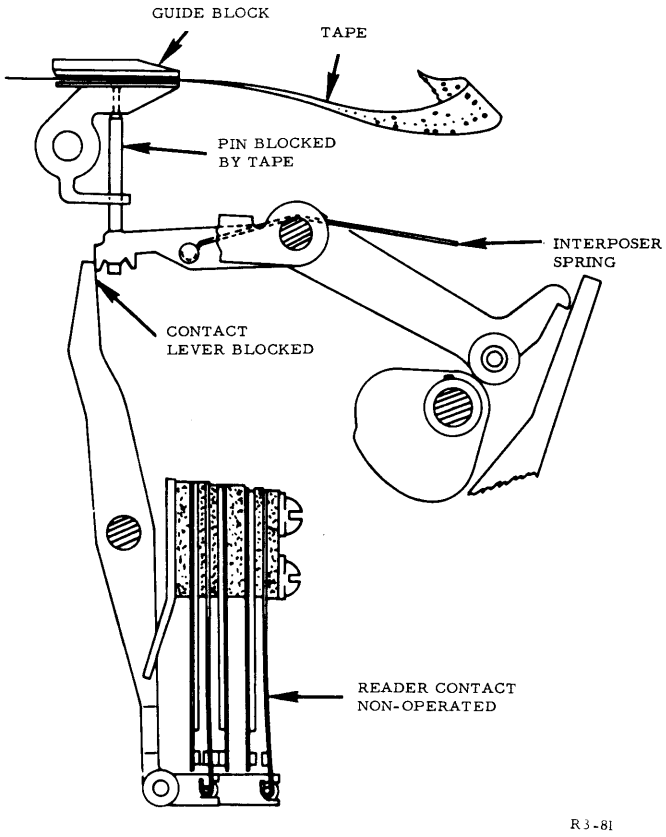


Figure 6-81. Feeler Pin Stopped by Tape.

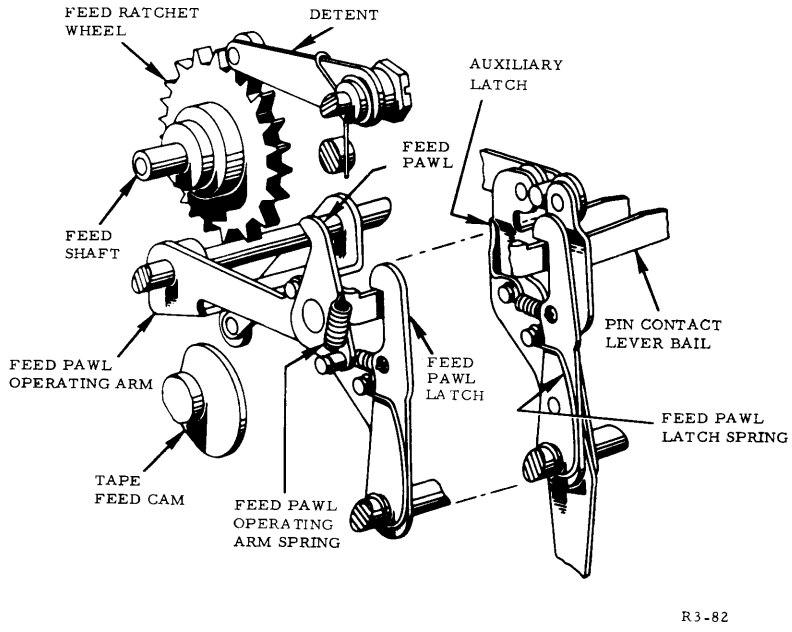


Figure 6-82. Tape Feed Mechanism.

Code Selector

Function of the code selector is to establish the appropriate binary-coded representation for each character and operation of the keyboard during its operation with the punch, punch and printer, and computer. The selector is designed to choose any code combination composed of up to 12 binary digits (bits). Operations of the RECOMP III computer system, however, use only a 6-bit code which produces a maximum of 64 code combinations.

To operate the required electrical contacts, the selector contains an assembly of 26 front and 25 rear selector slides which are mechanically actuated by the keyboard-printer front and rear cams (figure 6-84). Rotation of a cam causes a sliding movement of its respectively coded selector slide.

The selector slides are positioned to operate a group of seven lower and six upper transverse bails which operate contact shafts with tabs that, in turn actuate switch contacts. In the lower group, one bail is employed for each of the six bits of the code. The 7th bail is always operated by each selector slide. In the upper group an identical arrangement exists for code

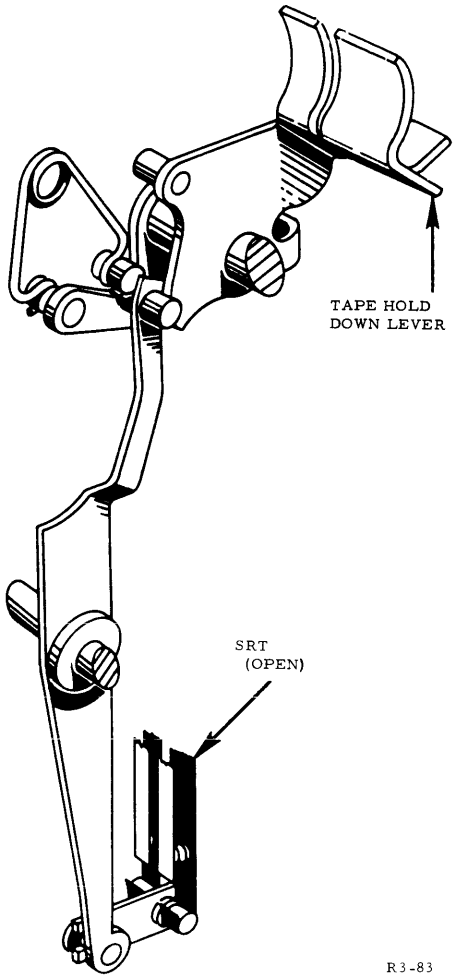


Figure 6-83. Tape Hold-Down Lever.

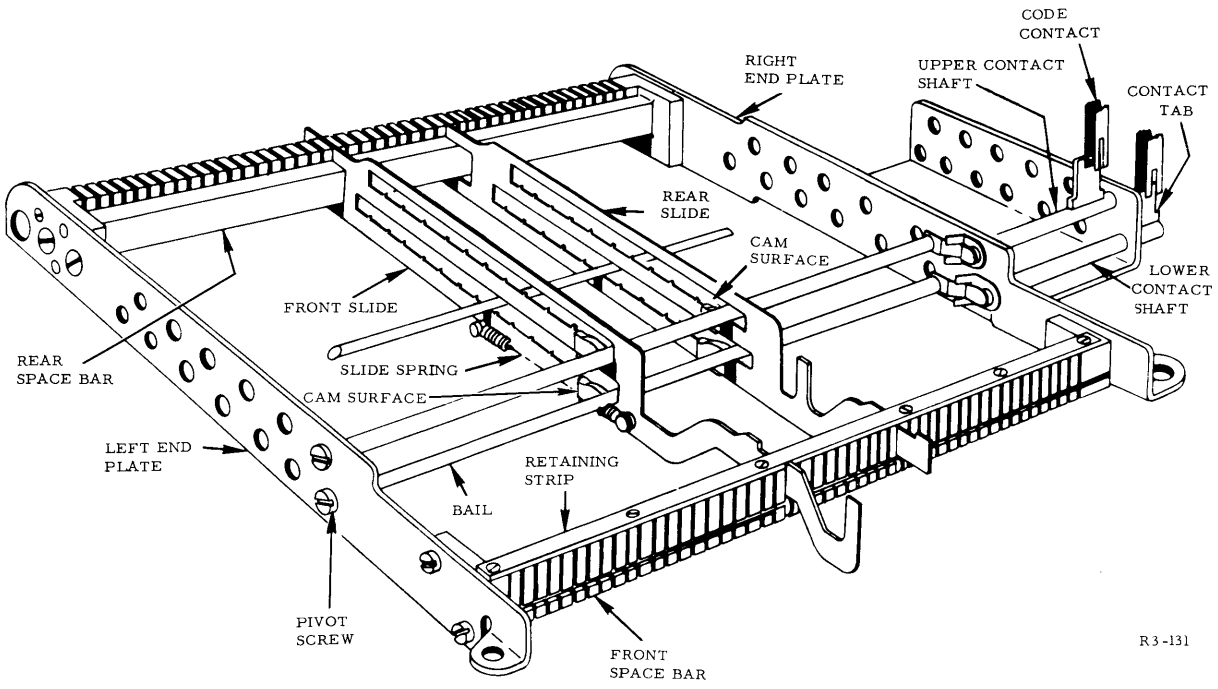


Figure 6-84. Code Selector.

channels one through five. Channel six is handled through a special dual tab and contact arrangement with the 7th lower bail.

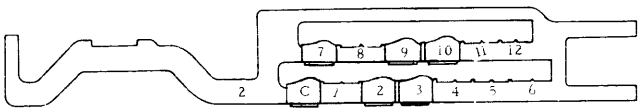
Coding of the selector slides is determined by the cam surfaces that operate the code bails (figure 6-85). A different combination of bails is operated by each individual selector slide displacement. Figure 6-86 illustrates typical slide displacement by front and rear cams, respectively; the illustration inset shows a transverse bail in the operating (raised) position resulting from selector slide action.

The transverse bails close electrical contacts located on the right selector end plate by actuating their respective contacts with the contact shaft and tab assembly. In the lower group, the contact shaft has two tabs to provide the 6th channel in Compute mode. The upper group of contacts are used during output to the computer; the lower group are utilized to actuate the punch magnets. Selection between the contact groups is dependent upon the position of the LOCAL-COMPUTE and PUNCH ON switches.

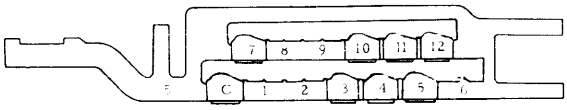
Code Translator

During operations with the printer, the code translator (figure 6-87) translates coded electrical impulses into mechanical selection and operation of printer keylevers. When operating with the punch, the translator only conveys the impulses to that device.

Translation during printing consists of energizing the code magnets for



TYP. SLIDE



TYP. SLIDE

FRONT SLIDE

TYPE BAR NUMBER	CAM SURFACE
R	7-8
	C-1
T	7-8-9-11
	C-1-2-4-6
U	7
	C
2	7-9-10
	C-2-3
4	7-8-9-10-11-12
	C-1-2-3-4-5
6	7-11-12
	C-4-5
8	7-8-10-11-12
	C-1-3-4-5
10	7-8-11
	C-1-4
12	7-11
	C-4
14	7-8-9-11
	C-1-2-4
16	7-8-9-11-12
	C-1-2-4-5
18	7-10-11
	C-3-4
20	7-8-9-10
	C-1-2-3
22	7-8-10-11
	C-1-3-4
24	7-8-9-12
	C-1-2-5
26	7-8-9-10-11
	C-1-2-3-4
28	7-9-12
	C-2-5
30	7-12
	C-5
32	7-9-10-11
	C-2-3-4-6
34	7-8-12
	C-1-5
36	7-8-9-10-11
	C-1-2-3-4-6
38	7-9-11-12
	C-2-4-5-6
40	7-10-11-12
	C-3-4-5-6
42	7-8-9-11-12
	C-1-2-4-5-6
V	7-10-11
	C-3-4-6
Z	7-9
	C-2

REAR SLIDE

TYPE BAR NUMBER	CAM SURFACE
S	7-8-10-11
	C-1-3-4-6
1	7-9-10-12
	C-2-3-5
3	7-9-12
	C-2-5-6
5	7-10-11-12
	C-3-4-5
7	7-8-9-12
	C-1-2-5-6
9	7-9-11
	C-2-4
11	7-10-11
	C-3-5-6
13	7-8-9-10-12
	C-1-2-3-5
15	7-8-10-12
	C-1-3-5-6
17	7-8-11-12
	C-1-4-5
19	7-9-10-12
	C-2-3-5-6
21	7-9-10-11-12
	C-2-3-4-5
23	7-8-9-10-12
	C-1-2-3-5-6
25	7-8-11-12
	C-2-4-5
27	7-11-12
	C-4-5-6
29	7-9-10-11
	C-2-3-4
31	7-8-11-12
	C-1-4-5-6
33	7-10-12
	C-3-5
35	7-12
	C-5-6
37	7-8-10-12
	C-1-3-5
39	7-8-12
	C-1-5-6
41	7-8-10-11-12
	C-1-3-4-5-6
43	7-9-10-11-12
	C-2-3-4-5-6
W	7-8-9
	C-1-2

R3-111

Figure 6-85. Selector Slide Coding.

a binary-coded digit and one magnet for operating a single-revolution mechanical clutch, then utilizing the magnet operation to move levers and initiate translator shaft rotation. The levers, called seekers, hook over studs on the printer keylevers which, in turn, actuate the typebars and cause other printer operations. A drive gear meshes with a power gear to provide rotation of the translator shaft when the clutch armature is released. During the shaft operation cycle, a selected seeker is moved to operate its associated keylever.

When using 6-bit code, the translator consists of two magnet yoke assemblies. These assemblies are identical except for the mounting plate and the number of coils and armatures. The magnet yoke assembly nearest the clutch end of the translator consists of four magnets and four armatures; the other magnet yoke assembly, three magnets and three armatures. In general, during a tape reader or computer output cycle, if channel one of a code contains a binary one, a circuit is completed to the coil of the number one code magnet in the translator. This circuit is similar for each of the seven magnets. Energization of the translator magnets is described in detail under Functional Circuits Operation in this section.

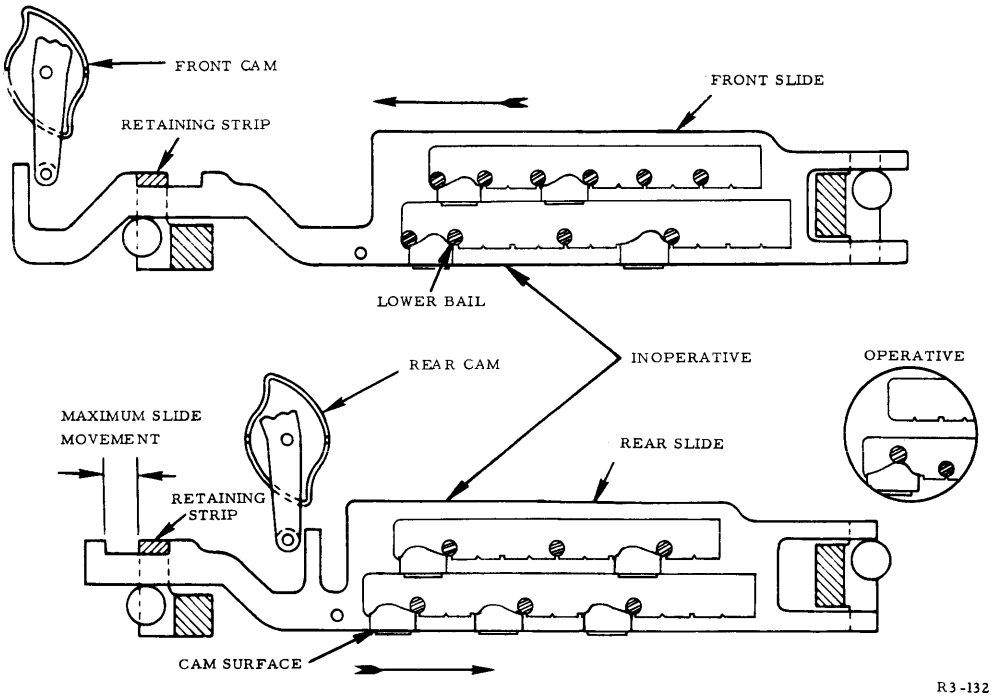


Figure 6-86. Front and Rear Slides.

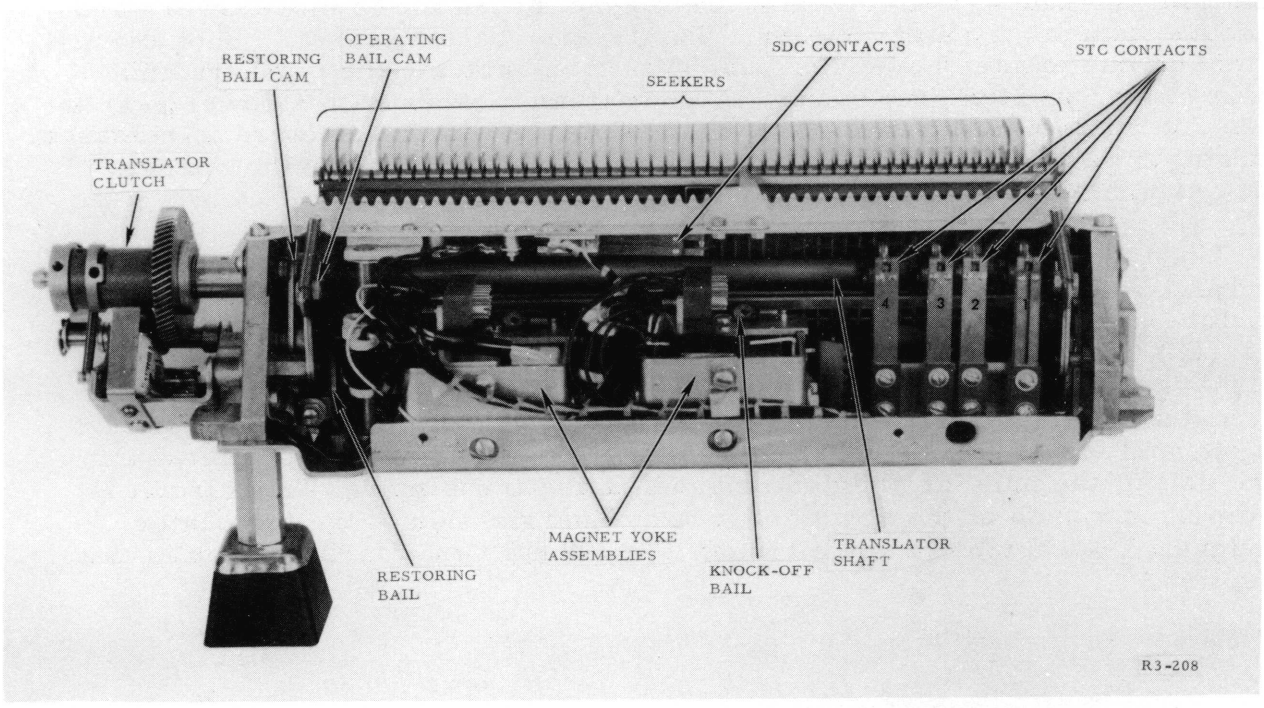


Figure 6-87. Code Translator.

Permutation Bars

Each of the code translator armatures controls a permutation bar by normally holding it in a non-operated (latched) position. When the magnet is energized, the armature is attracted, releasing the bar(figure 6-88). A plunger spring then moves the bar horizontally until the restoring stud on the bar strikes the restoring bail.

The seven permutation bars are similar, except on their projection or notched sides (figure 6-89). Projections are arranged to prevent movement of all seekers when the bar is in the normal position; and to permit seekers, corresponding to correct codes, to move into notches when the bar is in the operate position. A view of the permutation bars looking at the projection side (figure 6-90) shows the bars are not arranged in numerical order. This configuration is due to the positioning of the magnets. The bars are arranged such that, for any given combination of operated and non-operated bars, there is only one seeker position without projections. Accordingly, for each translator cycle, only one seeker operates its corresponding keylever.

As an example of permutation bar operation, (figure 6-91) if bars 2, 3, and 4 are released (moved to the left), the space opposite the letter "C" seeker is opened, while the spaces opposite the remaining seekers are closed. This operation is shown in figures 6-90 and 6-91. Note in figure 6-91 that the "C" seeker is seated in its opening by seeker spring tension.

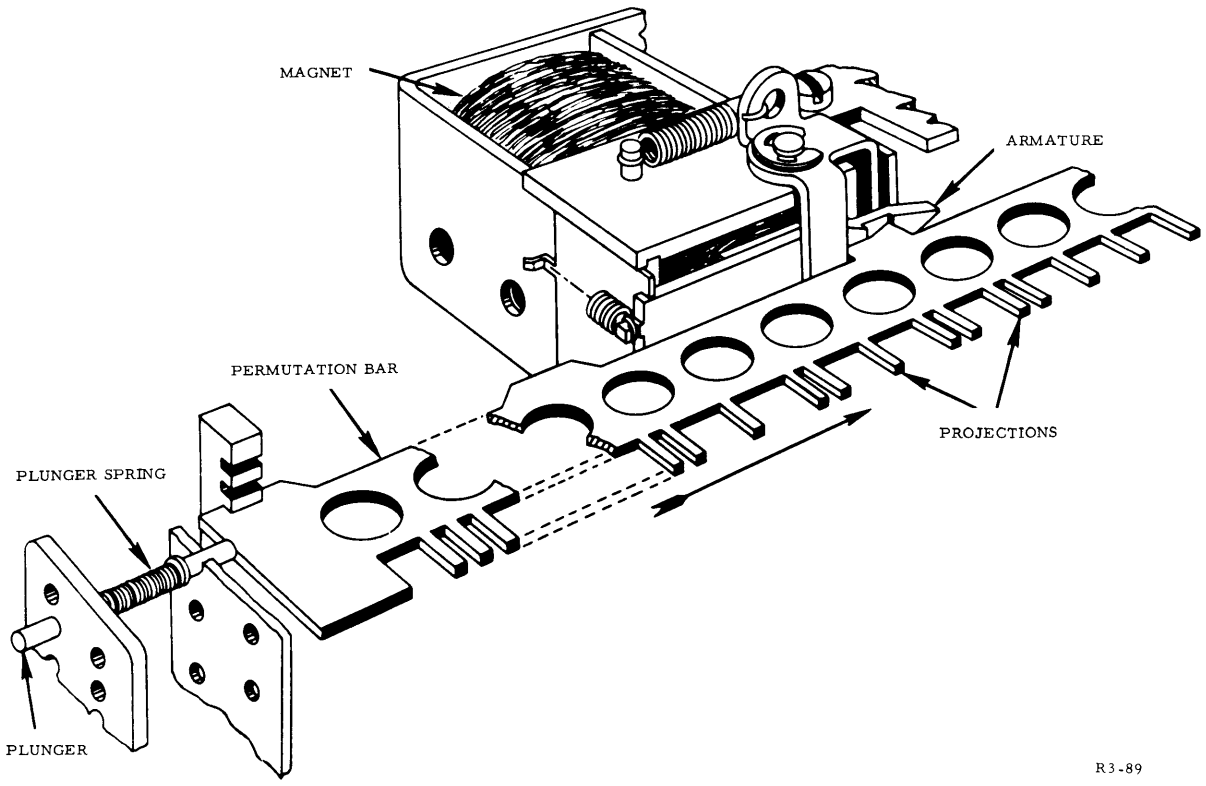


Figure 6-88. Permutation Bar Latched.

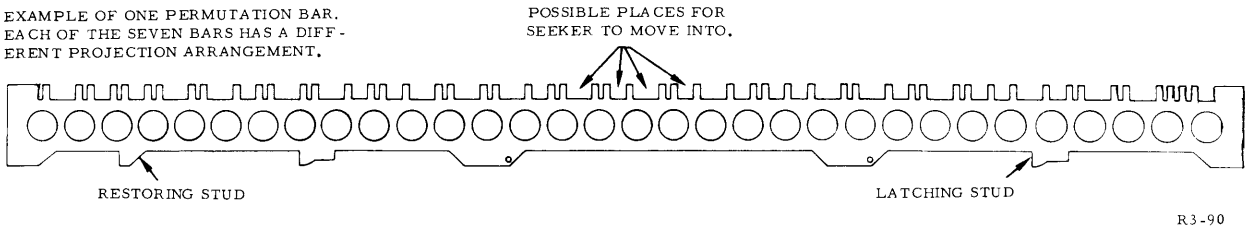


Figure 6-89. Sample Permutation Bar.

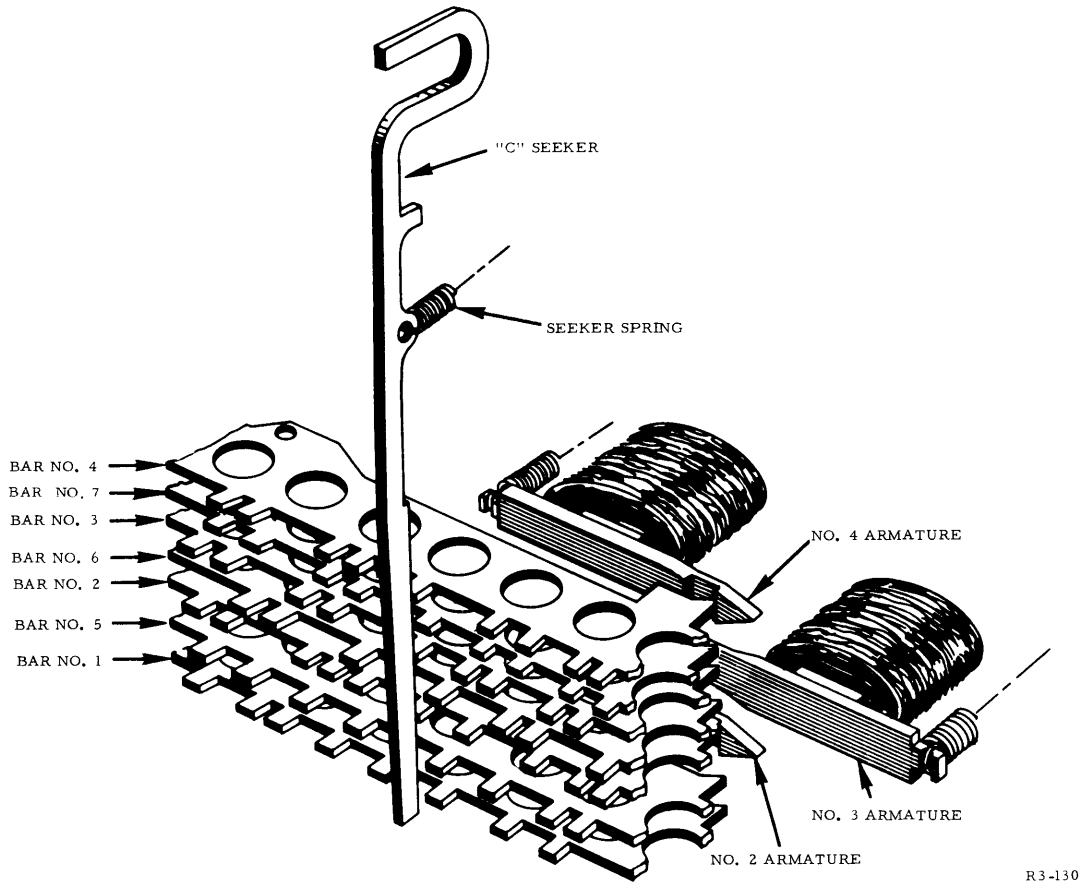


Figure 6-90. Permutation Bars Non-Operated.

Although energization of the clutch magnet occurs simultaneously with that of the code magnets, because of slower clutch action, the permutation bars move and permit proper seeker operation before another translator shaft rotation.

Translator Cams

The translator shaft contains 13 cams, 5 for mechanical and 8 for electrical actions. Of the five for mechanical action, two are identical seeker restoring bail cams, one at each end of the translator shaft. When the shaft is at rest (clutch disengaged), these cams hold the seeker restoring bail against the seekers to prevent their seating in the permutation bar crenels.

Operation of the translator clutch is similar to that of the tape punch clutch. When operated, it permits one complete rotation of the translator shaft. Detailed information on clutch operation is given under Punch and Translator Clutches in this section.

Contour of each cam is such that when the shaft rotates, the cam follower

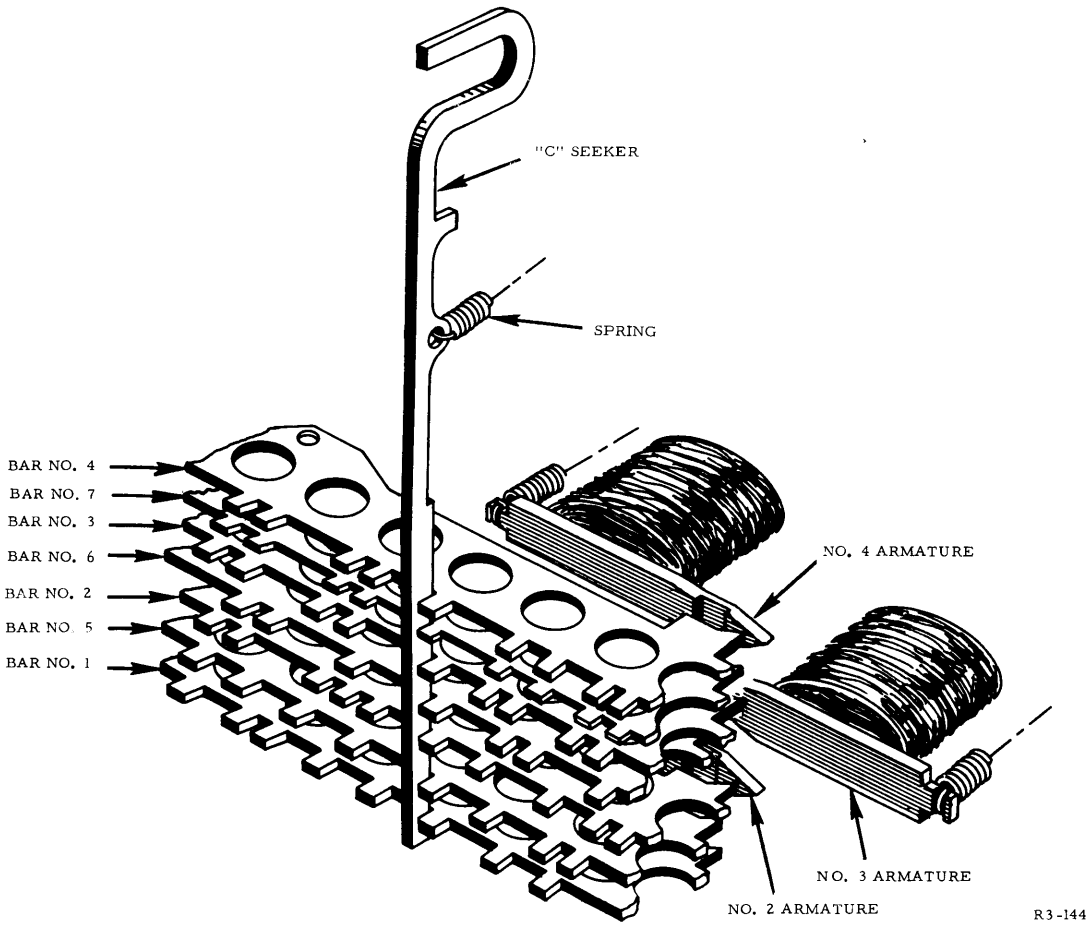


Figure 6-91. Permutation Bars Operated.

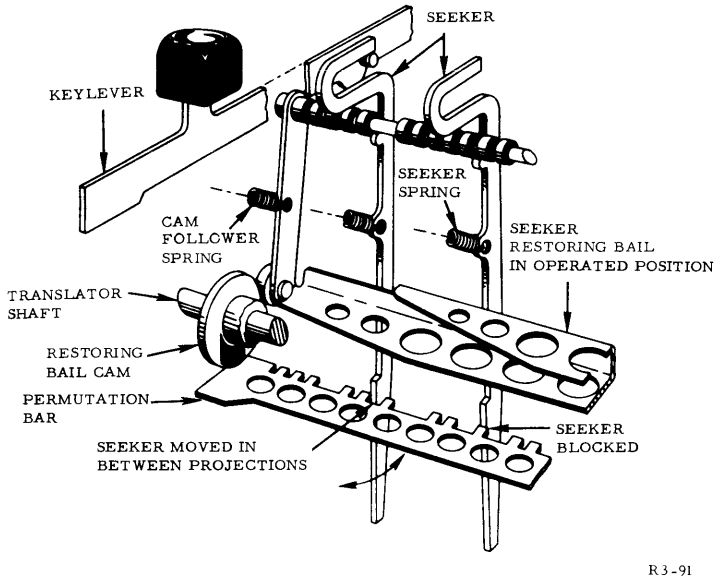


Figure 6-92. Seeker Restoring Bail.

spring holds the cam follower against the cam surface (figure 6-92). When the cam follower moves over the cam's flat contour, the seeker restoring bail is pulled inward, allowing the proper seeker to engage in the respective permutation bar crenels. As the cam approaches the end of its cycle, the seeker restoring bail allows the seeker to move away from the permutation bar crenels, thus restoring the seeker.

The two seeker operating bail (mechanical) cams, one at each end of the translator shaft, also are identical. If a seeker is allowed to move inward during the shaft cycle, just after it achieves maximum inward movement, an operating bail engages the notch on the seeker and pulls the seeker downward (figure 6-93). This action then pulls the corresponding keylever downward.

Located on the clutch end of the translator shaft (figure 6-94) the fifth mechanical cam operates the permutation bar restoring bail. During the latter part of the shaft cycle, the bar restoring bail moves all bars back beyond the latching point of their respective armatures before allowing them to latch. This bar overtravel results in the bar restoring stud moving the knock-off bail arm of the magnet yoke assembly (figure 6-95). The arm is pivoted, causing the knock-off bail to contact all the armatures and move them away from the magnet core. This knock-off action ensures positive latching of the bars with their respective armatures.

Punch and Translator Clutches

The clutch mechanisms used on the tape punch and translator are identical except for several changes on the punch clutch mechanism which adapt it for higher speed operation. Function in both instances, however, is to permit mechanical power to be transmitted from a continuously rotating gear to a drive shaft when a clutch magnet is energized. Figure 6-96 illustrates the tape punch clutch and the relative position of its component parts; figure 6-97 illustrates the translator clutch.

A disassembled clutch mechanism is shown in figure 6-98. The clutch spring is close-wound from rectangular wire. Dimension of the inside diameter of the spring when it is not expanded is maintained very accurately to ensure secure spring grip on both the gear hub and the clutch collar. The end of the spring which fits into a recess in the clutch collar has a right angle bend and rides over the clutch collar as far as the end of the clutch spring slot. The opposite end of the clutch spring rests against a pin that protrudes into the clutch sleeve inner diameter and rides over the gear hub (figure 6-99).

When the clutch magnet yoke assembly is not energized and the end of its armature is resting against the raised edge of the clutch sleeve, the clutch spring is sufficiently expanded so it does not grip the hub of the driven gear. The driven gear then rotates freely without turning the shaft. When the magnet yoke assembly is energized initiating the on cycle, the end of the armature disengages from the edge of the raised surface on the clutch sleeve outer diameter. This disengagement allows the clutch sleeve to rotate in a clockwise direction. The pin located in the sleeve inner diameter then moves away from the end of the spring, which allows the spring to contract and grip the gear hub. Because the other end of the spring is simultaneously trans-

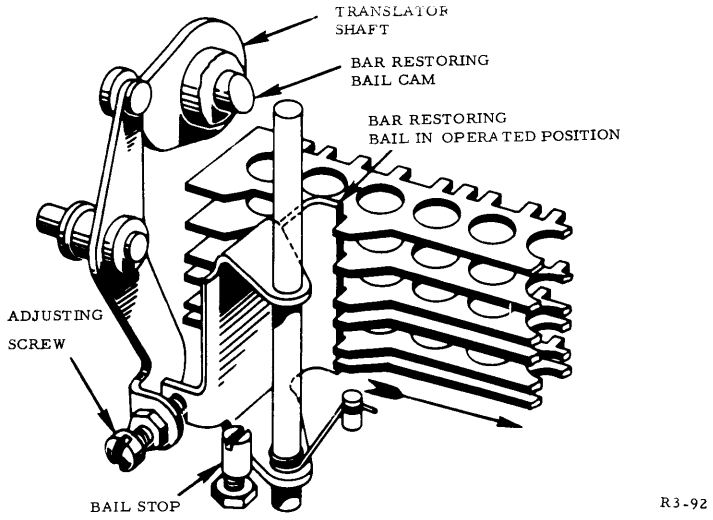


Figure 6-93. Seeker Operating Bail.

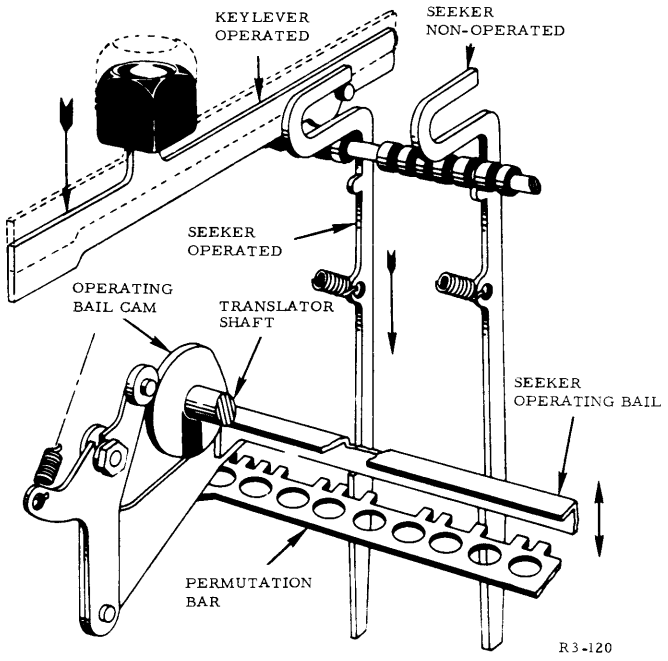


Figure 6-94. Bar Restoring Bail.

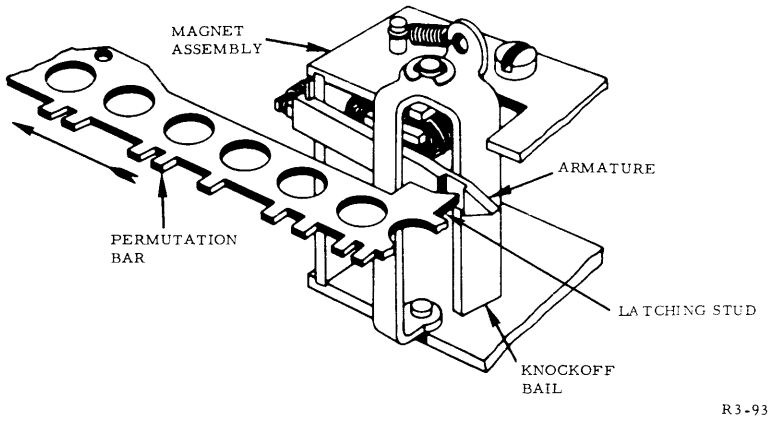


Figure 6-95. Armature Knock-off.

ferring clockwise rotary motion to the clutch collar and shaft, the spring winds tighter on the gear hub and the shaft then rotates through one punch or translator cycle.

On the punch, due to its rapid operation and the long current decay time of the magnets, the off cycle is initiated by action of the knock-off cam and overtravel stop. On the translator, operation is slower and the knock-off cam and follower are not required. Accordingly, the combination knock-off cam and overtravel stop is replaced by the overtravel stop on the translator. Both clutch mechanisms incorporate an armature stop bracket to prevent armature drag on the clutch sleeve when the magnet yoke assembly is not energized.

After the armature moves away from its magnet cores, it again engages the edge of the raised surface on the clutch sleeve. This engagement stops the sleeve rotation which unwinds the clutch spring, releasing the gear hub. Shaft rotation continues, however, until the overtravel stop comes into effect. The detent is then overthrown, and the detent lever locks the clutch mechanism with the clutch spring in the slightly unwound position, ready for the next on cycle.

Control Circuits

The typewriter control circuits consist of punch circuits, reader and translator circuits, amplifying circuits (driver boards), and control network circuits (logic network boards). Both the driver and logic network boards are mounted on a chassis in the desk. The following descriptions of the control circuits state the purpose and general operation of each circuit involved in controlling Flexowriter actions. Detailed operation of all circuits in the Flexowriter during both local and compute modes are presented by function under Functional Circuits Operation. A glossary of circuit and signal designations and definitions is given in table 6-6.

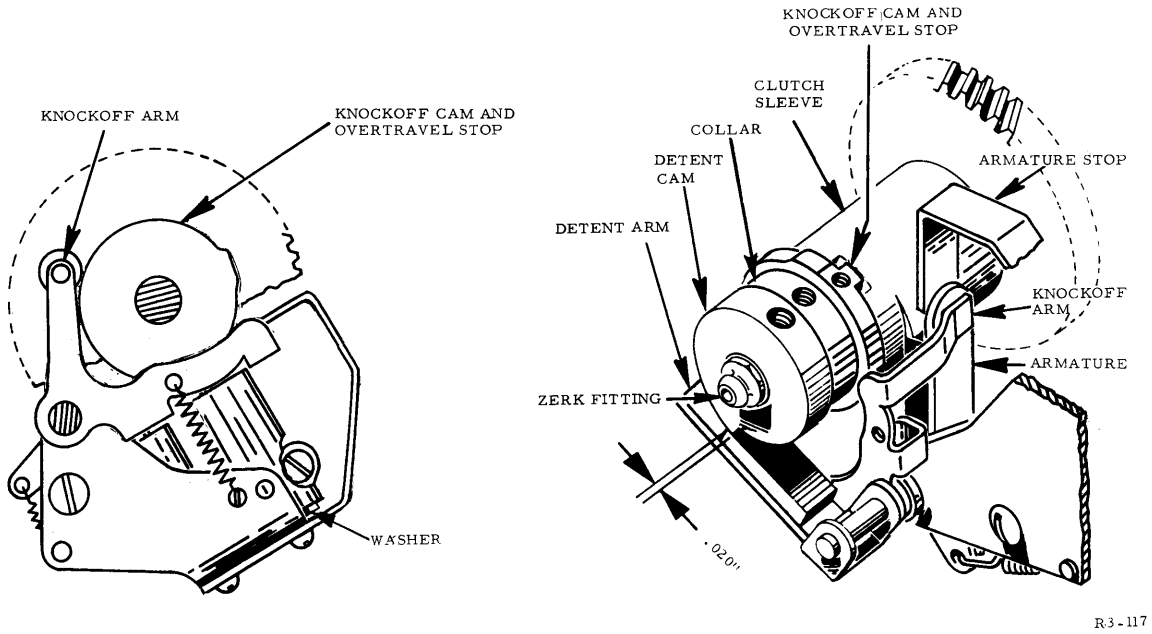


Figure 6-96. Punch Clutch Magnet Adjustments.

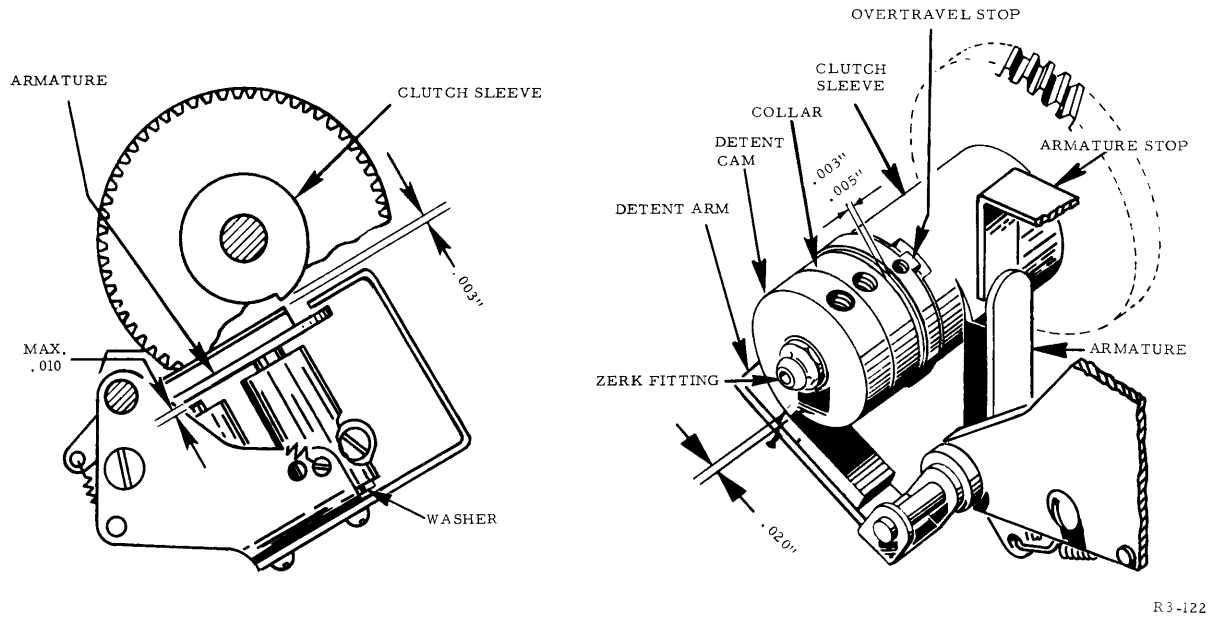


Figure 6-97. Translator Clutch Magnet Adjustments.

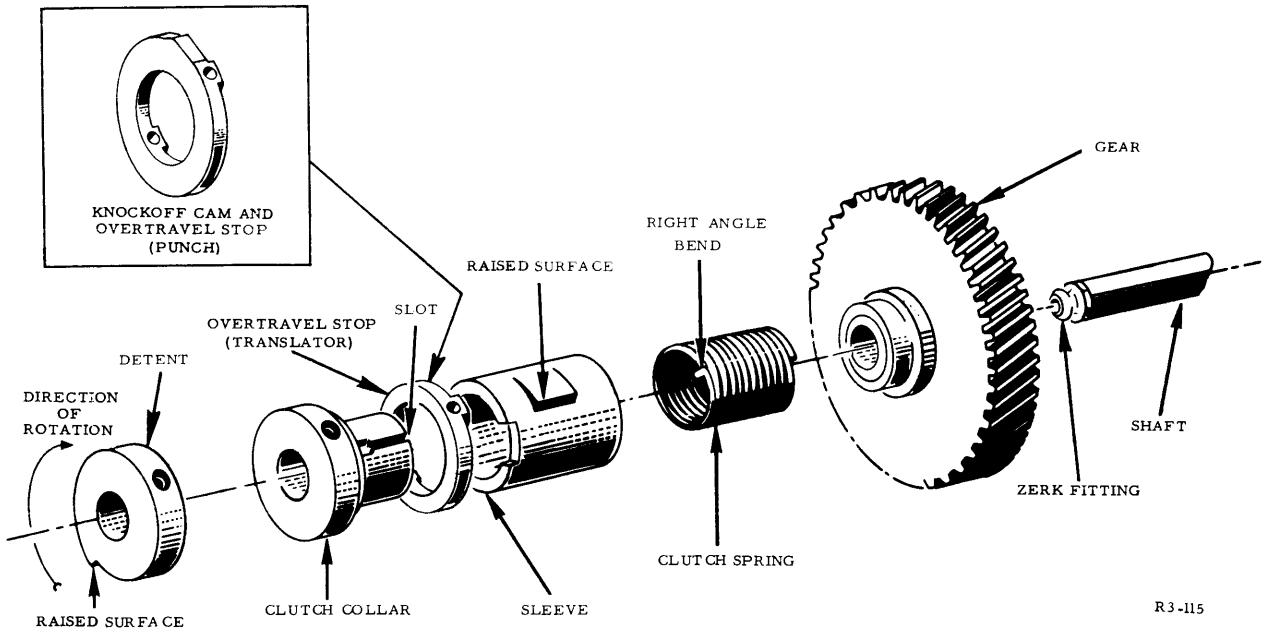


Figure 6-98. Clutch Mechanism Disassembled.

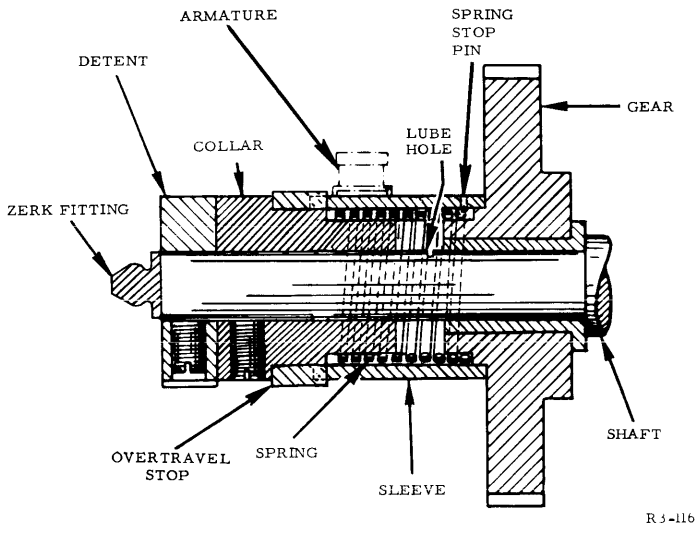


Figure 6-99. Clutch Mechanism Cutaway -- Side View.

Key Lock Magnet

All keylevers are locked when power is off. When the POWER ON switch (S1) is placed at ON, the keylock magnet is energized permitting keylever action except during the following functions: carriage return, tab, backspace, delay control, stop code, punch error, and no tape or incorrect tape tension.

Clutch Magnet Circuit

The clutch magnet circuit controls both the punch and translator clutches. Consequently, energization of the magnet causes both the punch and translator to start a cycle of operation. Operation of the clutch magnet circuit and the utilization of signals generated during the punch and translator cycles are described under Functional Circuits Operation in this section.

Anti-Repeat Circuit

Under certain conditions, the punch completes a cycle of operation before the selector common contact opens. To prevent a repeat punch cycle, an anti-repeat circuit is incorporated with the punch code and clutch magnet circuits. At the start of a punch cycle, punch lock contacts transfer to (1) complete a circuit to the anti-repeat relay and (2) open the circuits to the punch code and clutch magnet circuits. If the selector common contact is still closed when the punch lock contacts return to normal, the anti-repeat relay is held energized by its own transfer contacts because the punch lock contacts are of the make-before-break type. The anti-repeat relay then remains energized as long as the selector common contact remains closed, thus preventing energization of the punch code magnets even though the punch lock contact has returned to normal. Inability to energize the code magnets then prevents repeat operation of the punch regardless of the length of time the selector contacts remain closed.

Punch STOP CODE Circuit

This circuit causes perforation of the tape in the 2nd, 4th, and 6th channels which, when read as a code by the tape reader, automatically stops tape reader operation. Depressing STOP CODE switch energizes the code relay completing a circuit to the punch code magnets and to the clutch magnet. Although all of the code relay contacts are closed when STOP CODE switch is depressed only the 2-4-6 code magnets are energized. Regardless of the length of time the STOP CODE switch is depressed, only one cycle of the punch is possible because the anti-repeat relay was energized from the STOP CODE switch instead of the selector common contact.

Punch CODE DELETE Code Circuit

The CODE DELETE circuit causes perforations of the tape in channels 1 through 6 which are read as a code by the tape reader but do not result in any operation by the printer. Consequently, this 1-2-3-4-5-6 code is used for deleting a code in the tape which was made through an error by the operator.

Depressing CODE DELETE switch energizes the code delete relay which completes a circuit to the punch code magnets and clutch magnet. All six punch code magnets are energized. Regardless of the length of time the CODE DELETE switch is depressed, only one cycle of the punch is possible because the anti-repeat relay was energized from the CODE DELETE switch.

Tape Feed Circuit

A tape feed circuit is used to advance the tape through the tape punch unit and simultaneously perforate sprocket holes only in the tape. Pressing and holding the TAPE FEED switch energizes the punch clutch magnet, allowing continuous operation of the punch.

Manual Start and Stop Circuits

Depressing the START READ switch energizes the tape reader magnet to start a reader cycle of operation. The start circuit is completed through the reader control relay, with a holding circuit keeping the reader magnet energized after the START READ switch is released. Once the circuit to the reader magnet is established, the magnet will remain energized and the reader will operate continuously until the reader magnet circuit is broken automatically (by delay control) or by depressing the STOP READ switch.

When the STOP READ switch is depressed, the holding circuit to the reader control relay is broken. The reader magnet de-energizes, stopping reader operation.

Delay Control Circuit

The delay control circuit is essential during printer operation to delay operation of the tape reader until a function in the printer has been completed. Three functions require more operating time than is necessary for printing characters: these are carriage return, back space, and tabular. When the delay control relay is energized, the circuit to the reader magnet is opened, stopping reader operation. Transfer of the delay control relay contacts allows a holding circuit to the delay control relay to be established.

For example, if a carriage return code is read into the reader at the same time the holding circuit is completed to the delay control relay, a circuit would be completed to the translator magnets and the translator clutch magnet. The translator would operate, pulling down the carriage return lever, and starting a carriage return function. Operation of the carriage return mechanism would then open the holding circuit contacts to the delay control relay. The delay control relay contacts then would return to their normal position, but the circuit to the reader magnet would not be completed until the carriage returns to the left margin and the clutch toggle unlock. When unlocking occurs, the carriage return contact would close to complete the energizing circuit to the reader magnet, and to start reader operation.

STOP CODE Reader Halt Circuit

When a STOP CODE is read by the reader, the holding circuit for the reader control relay will be broken, thus de-energizing the reader magnet and stopping reader operation. The START READ switch must be depressed to start reader operation again.

Origin-Sector Channel Read Enable Circuit

Depression of the START READ switch while the Flexowriter is in compute mode generates R_{gf} by causing routing of -12 volts to JL2-11. However, this signal has no effect except when the START READ switch is held down simultaneous with manipulation of the OPERATION switch on the control console. The R_{gf} signal then causes execution of the Bootstrap Read routine in the origin-sector channel. R_{gf} is routed to the same terminal in the computer as R_g from the control console. Execution of the routine is performed in the same manner, regardless of the source of the R_g signal. Routing of the -12 volts to produce the R_{gf} signal is accomplished by compute mode relay K12 being energized and closing its contacts 16 and 17 and the START READ switch upon depression causing energization of K27 with subsequent closing of its contacts 3 and 4.

Functional Circuits Operation

The following individual descriptions of the circuits within the Flexowriter are presented by function to facilitate understanding and troubleshooting. Description of the compute mode operations give switch settings of the Flexowriter controls only; it is assumed that switches on the computer control console have been set to their correct settings for the particular operation described. In addition to descriptions of the circuits that are active during the various local and compute mode operations, explanations also are presented on the power supply and motor circuits.

Schematics of the individual circuits are presented with the respective circuit descriptions; a schematic of the complete Flexowriter circuitry is shown in Section 7, Schematics and Diagrams, of the Service Manual. A glossary of circuit and signal designations used in Flexowriter operation is presented in table 6-6 of this manual.

Keyboard to Computer (Compute Mode)

With switch S1 at ON and switch S2 at COMPUTE, signal (R_{ks}) from the computer energizes input control relays K8 and K9 and lights the INPUT indicator. Energizing relays K8 and K9 changes input from tape reader to keyboard using contacts SS6-SS12. As a character or function keylever is depressed, the SS contacts (table 6-6),representing the code for the character or function,transfer and put -12 volts (figure 6-100) on the selected K lines (K_{1f} through K_{8f}).

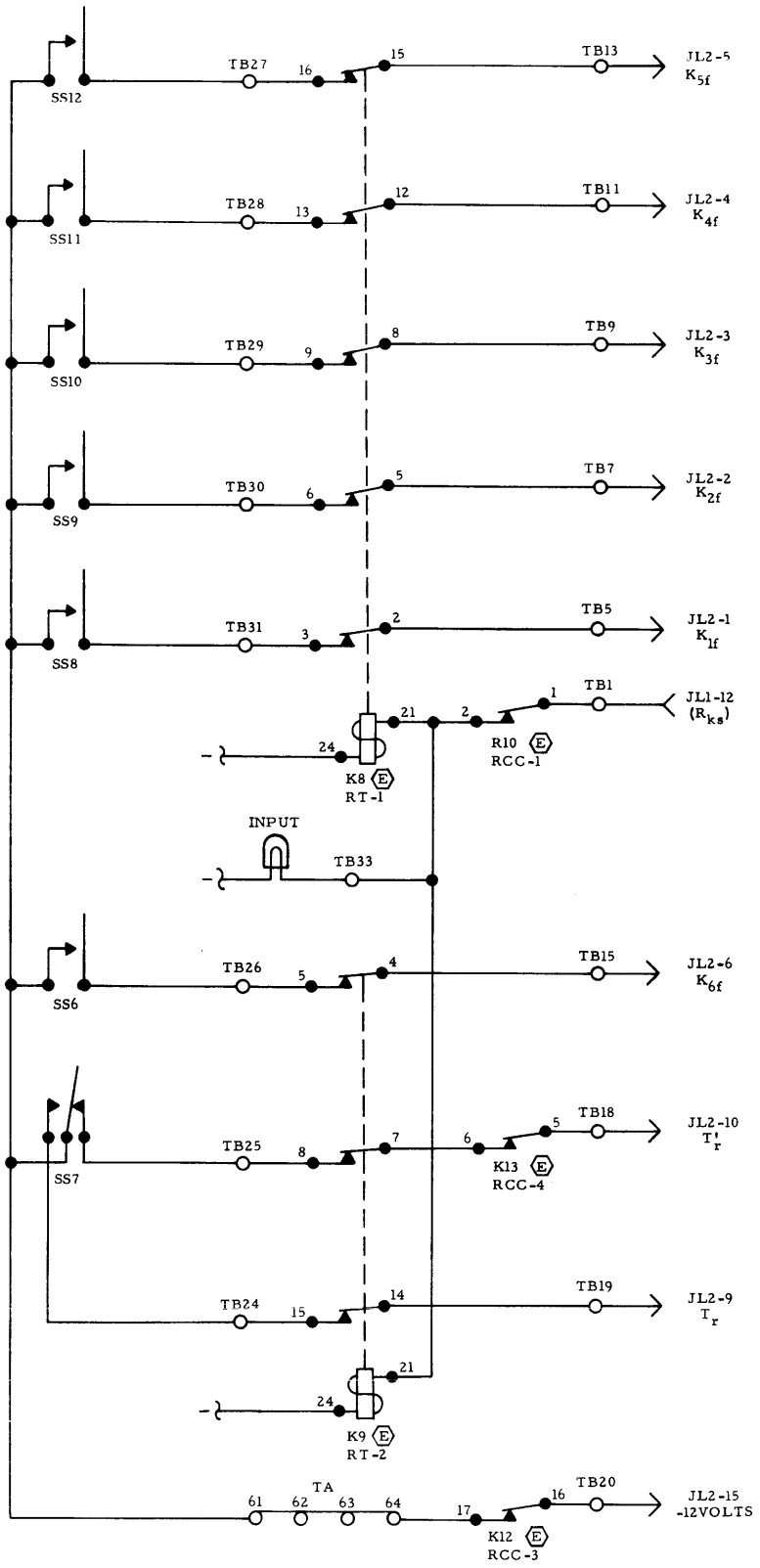


Figure 6-100. Keyboard to Computer Input Circuit (Compute Mode).

The -12 volts is supplied from the computer to the keyboard-printer through JL2-15. A -12-volt level is also applied to contact SS-7 (code selector common) which, when at rest, sends signal T'_r to the computer and, when transferred, T_r (an input sampling signal). Refer to figure 6-101 for code selector contact timing.

Tape Reader to Computer (Compute Mode)

With switch S1 at ON and switch S2 at COMPUTE, a -12-volt level is present on the reader contacts common bus line (figure 6-102) for SRC, and SR1 through SR8 contacts. Read command signal (R_d) energizes reader magnet LR through SRT contacts to start reader operation. Reader contacts, corresponding to the code sensed by the reader pins, transfer and a -12-volt level appears at the applicable input lines K_{1f} through K_{8f}. SRC contacts 5, 6, and 7 produce terms T_r and T'_r which indicate when a code is being sensed by the reader pins or when the contacts are at rest position, respectively.

Computer to Punch (Compute Mode)

With switch S1 at ON, and switch S2 at COMPUTE, the system in output mode, and the tape punch selected as the output device (figure 6-103), signal [P_{kd}] energizes both the punch and translator clutch magnets. Signals that appear at input lines JL1-1 through JL1-8 energize corresponding punch code magnets. The translator is cycled together with the punch to produce required timing signals O_r and T_p. Normally closed contacts 1 and 2 of relay K 14 furnish signal P to the computer. Figure 6-104 shows the P signal generation circuitry.

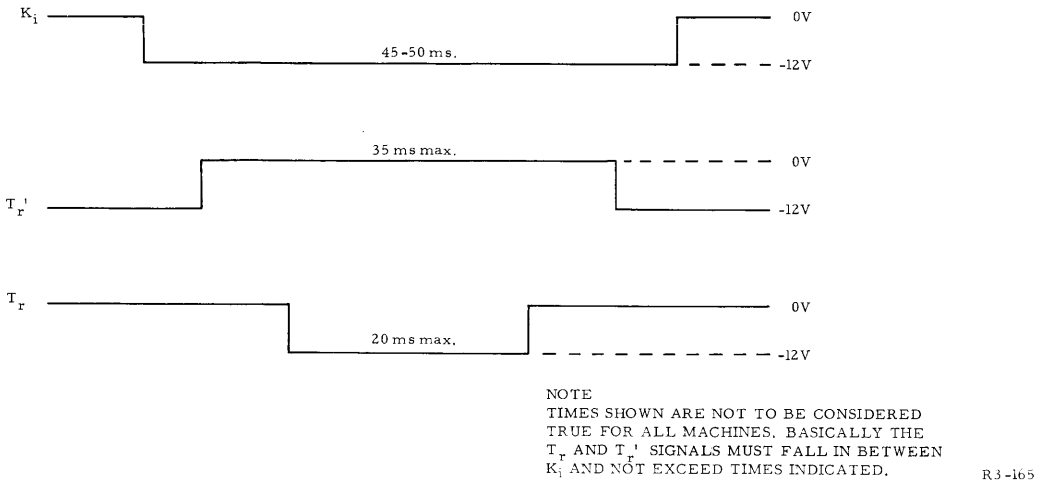


Figure 6-101. Code Selector Contact Timing.

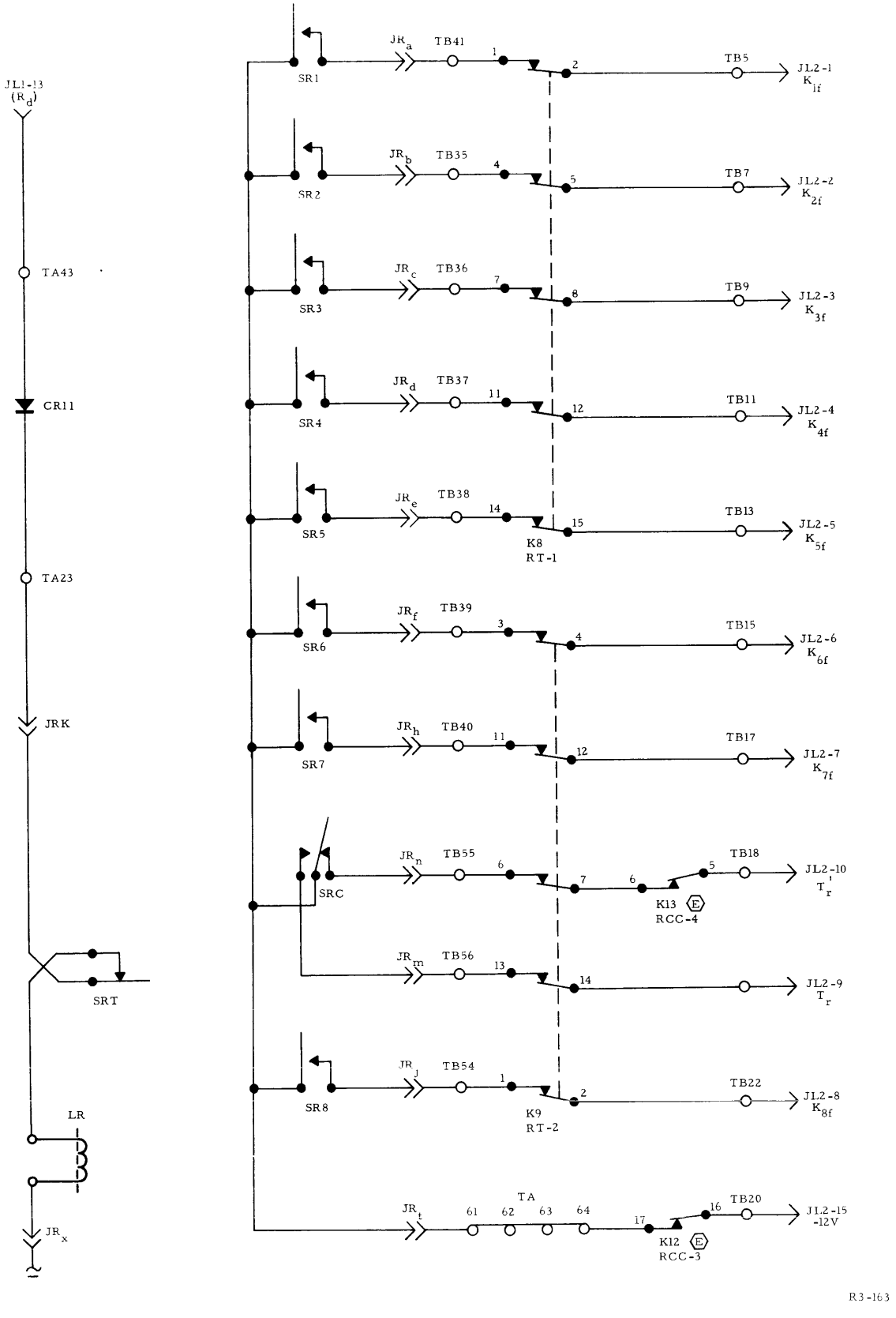


Figure 6-102. Tape Reader Input to Computer Circuit (Compute Mode).

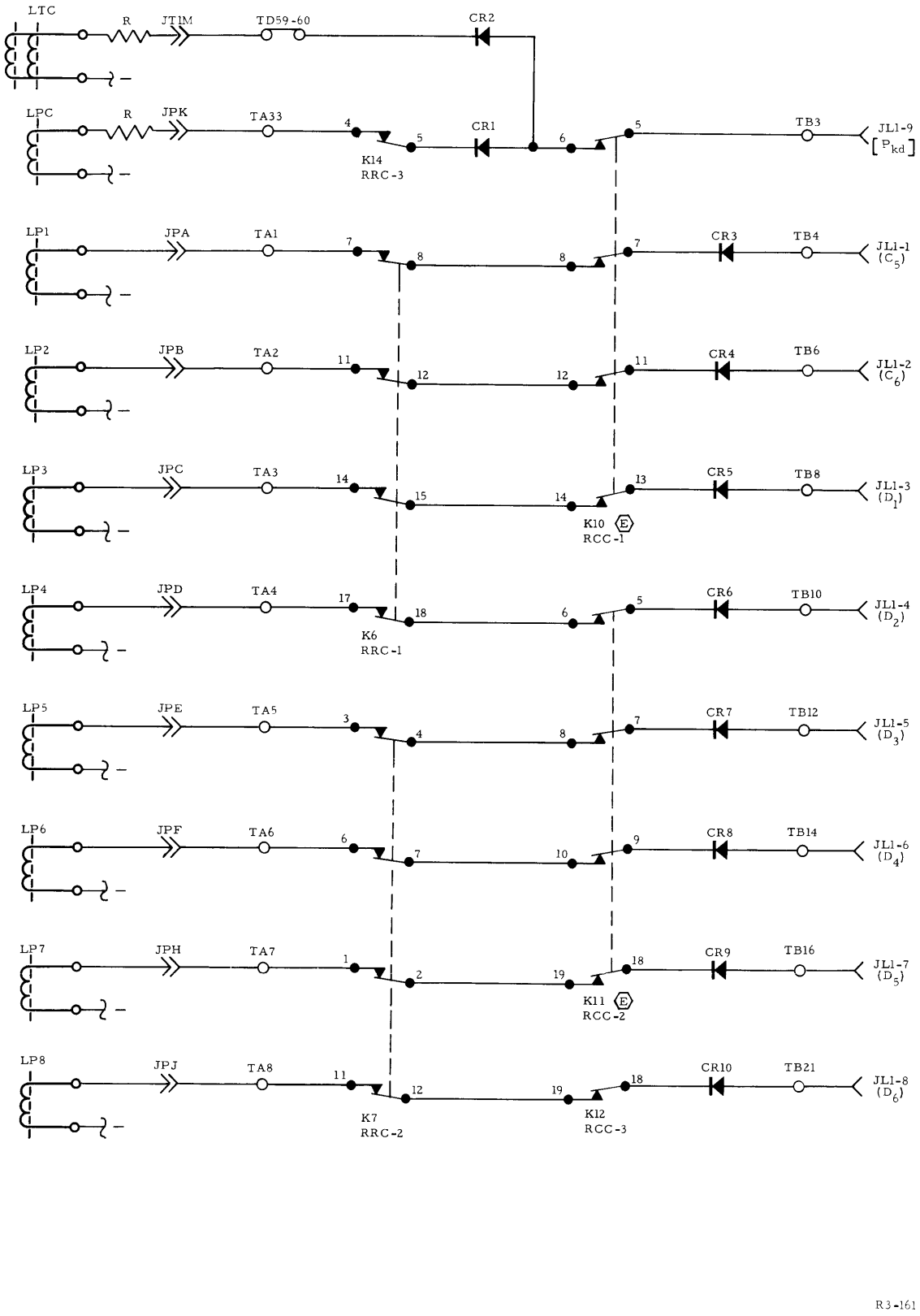


Figure 6-103. Punch Circuit With Punch Selected (Compute Mode).

Figure 6-104. P Signal Generation.

Computer to Printer (Compute Mode)

With switch S1 at ON and switch S2 at compute, signal (P_{ks}) energizes relays K6, K7, and K14 (RRC1, RRC2, and RRC3) to select the printer for output (figure 6-105). K14 energizes last, allowing contacts on relays K6 and K7 to switch the output circuit from the tape punch to the translator for printer action and to stabilize before the K signal becomes true. The K signal is used in the gate for signal [P_{kd}] which, among other functions, energizes the translator clutch magnet and translator code magnets LT1 through LT7. Magnet LT7 is energized so that a blank key operation will occur if no code magnets have been energized. (Translator contacts STC-2 are in operation during the translator cycle and are included in the O_r signal line). Signals appearing on JL1-1 through JL1-6 energize corresponding translator code magnets.

Contacts of STC-1 transfer during a translator cycle to generate signals T_p and T'_p. Normally closed contacts of STC-1 transmit T'_p and closing of the normally open contacts transmit T_p. A-12-volt level is applied to STC-1 common contact from JL2-15. Refer to figure 6-106 for T_p and T'_p signal generation.

Output Ready (Compute Mode)

With switch S1 at ON and switch S2 at COMPUTE, the output ready (O_r) signal is sent to the computer from the keyboard-printer. O_r when true, is at a -12-volt level and is present only when switch S2 is at COMPUTE and when all other contacts of the O_r interlock circuit are in the proper position. The -12 volts for the O_r signal enters the keyboard-printer at JL2-15 and is sent through contacts 16 and 17 of relay K12 to the O_r signal line. Refer to figure 6-107 for O_r circuit. Contacts included in the O_r signal line are as follows:

Normally closed contacts 1 and 2 of relay K21; this relay is energized by either a carriage return or a tab operation which opens contacts 1 and 2.

Contacts STC-2 which interrupt the O_r signal during a translator cycle, opening at 65 degrees to 351 degrees of translator operation.

Contacts 4 and 5 of SCRT which are open during entire duration of either a carriage return or a tab operation.

Contacts 3 and 4 of SBS which are open during a back space operation.

Contacts 1 and 2 of relay K24 if the computer is transmitting to the tape punch; K24 is energized and contacts 1 and 2 opened if the SPT contacts are closed.

Contacts 1 and 2 of LKL are closed if the computer is transmitting to the translator except during carriage return, tab, or backspace operation.

Contacts 2 and 3 of relay K6 which provide O_r signal to JL2-16 when the computer is transmitting to the printer; contacts 1 and 2 of relay K6 if the computer is transmitting to the tape punch.

Delay Control During Printing (Compute Mode)

When the computer is transmitting to the printer, computer operation must be delayed for the completion of any printer operation that requires more time than the ordinary printing operation. This delay is necessary to prevent the computer from transmitting new information to the printer before completion of the previous operation. Operations requiring additional time are carriage return, backspace, and tabulation.

To delay operation of the computer, the O_r circuit must be interrupted. Delay control (figure 6-108) is accomplished by three separate operations of the circuit: delay control qualifying, primary delay, and function control delay. These circuit operations are described in following paragraphs:

Delay Control Qualifying (Compute Mode)

Delay control qualifying is accomplished by the operation of translator seekers whose respective keylevers will require longer than one printing cycle to complete. These seekers operate the SDC (switch delay control) which energizes relay K21.

Primary Delay (Compute Mode)

Primary delay is accomplished by relay K21 (delay control relay). Because SDC is operated for only approximately 10 milliseconds, K21 must interrupt the O_r circuit through completion of the operation of the keylever which requires delay.

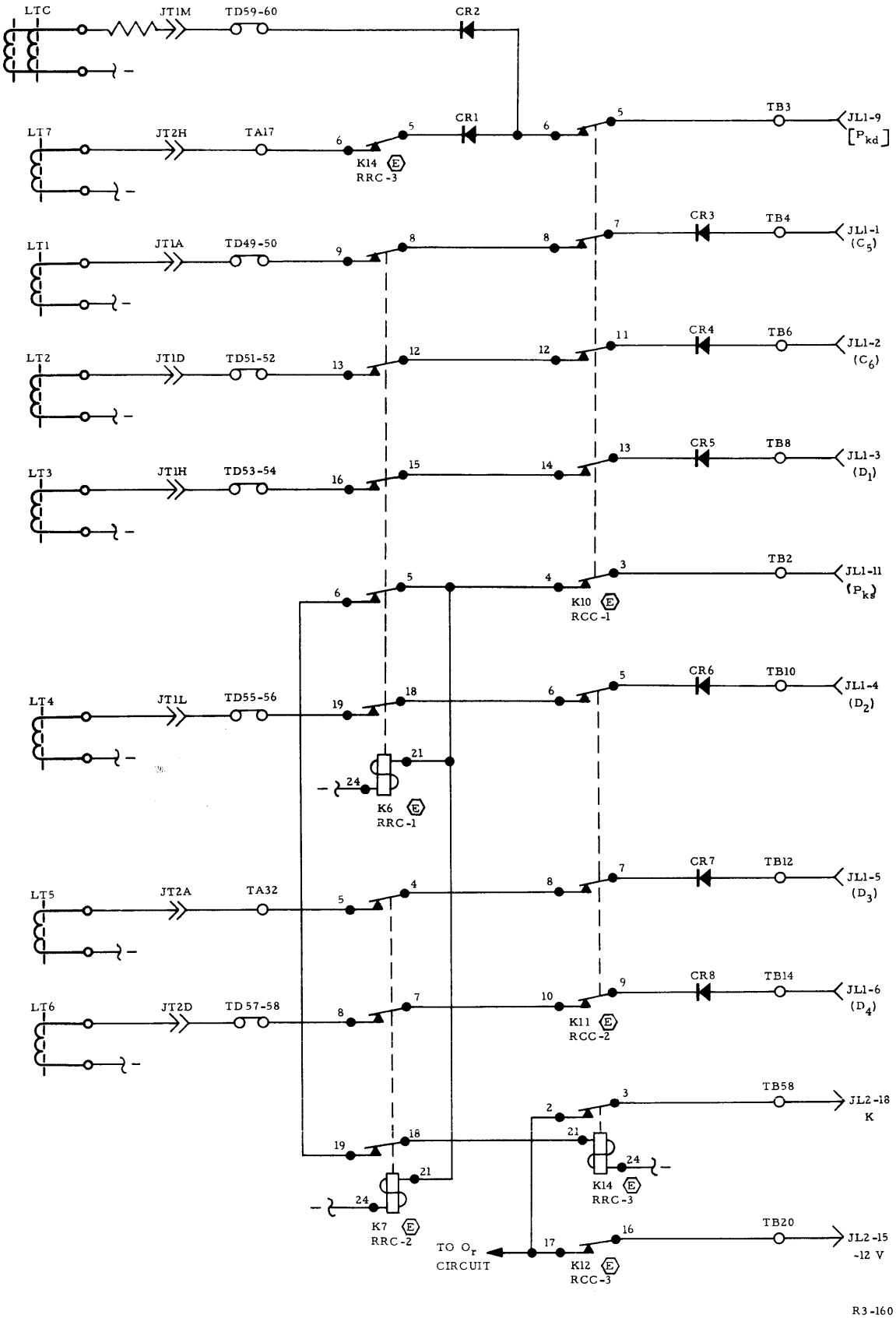


Figure 6-105. Computer to Printer Output Circuit (Compute Mode).

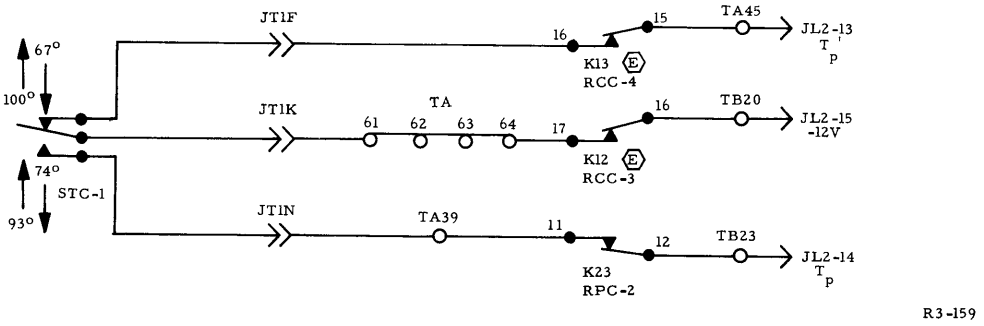


Figure 6-106. T_p and T_p' Signal Generation.

Function Delay (Compute Mode)

If, for example, a tab code is sent to the translator, SDC initially energizes K21 which remains energized through its own contacts 3 and 4 and through normally closed SCRT contacts 1 and 2. Normally closed contacts 1 and 2 of K21 open when the relay is energized, which interrupts the O_r circuit. As soon as the tab keylever is operated, the linkages cause SCRT to transfer, opening the circuit to K21. SCRT contacts 4 and 5 continue to interrupt the O_r circuit for the duration of the tab operation. When a tab stop strikes the tab lever, SCRT is restored by the unlatch cam and the O_r circuit is again completed through SCRT contacts 4 and 5. Delay control operation is identical during backspace and carriage return operation except SBS or SCRT transfer, depending on the operation selected.

Transfer of SCRT or SBS breaks the circuit to LKL, which disables the keyboard. Translator contacts STC-2 are also included in the O_r circuit. Additional information on the O_r circuit is given under Output Ready (Compute Mode).

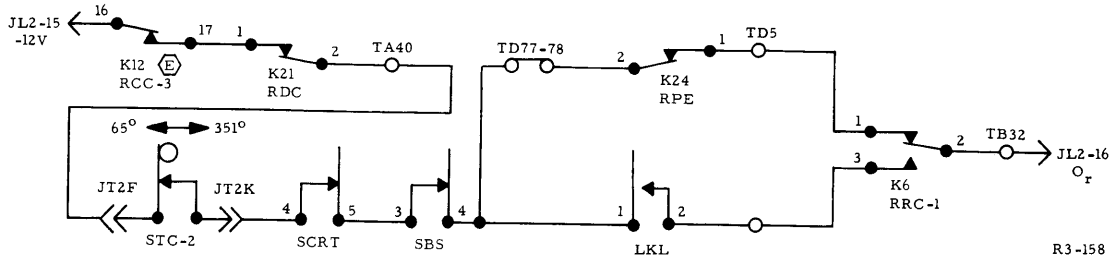


Figure 6-107. Output Ready Circuit (Compute Mode).

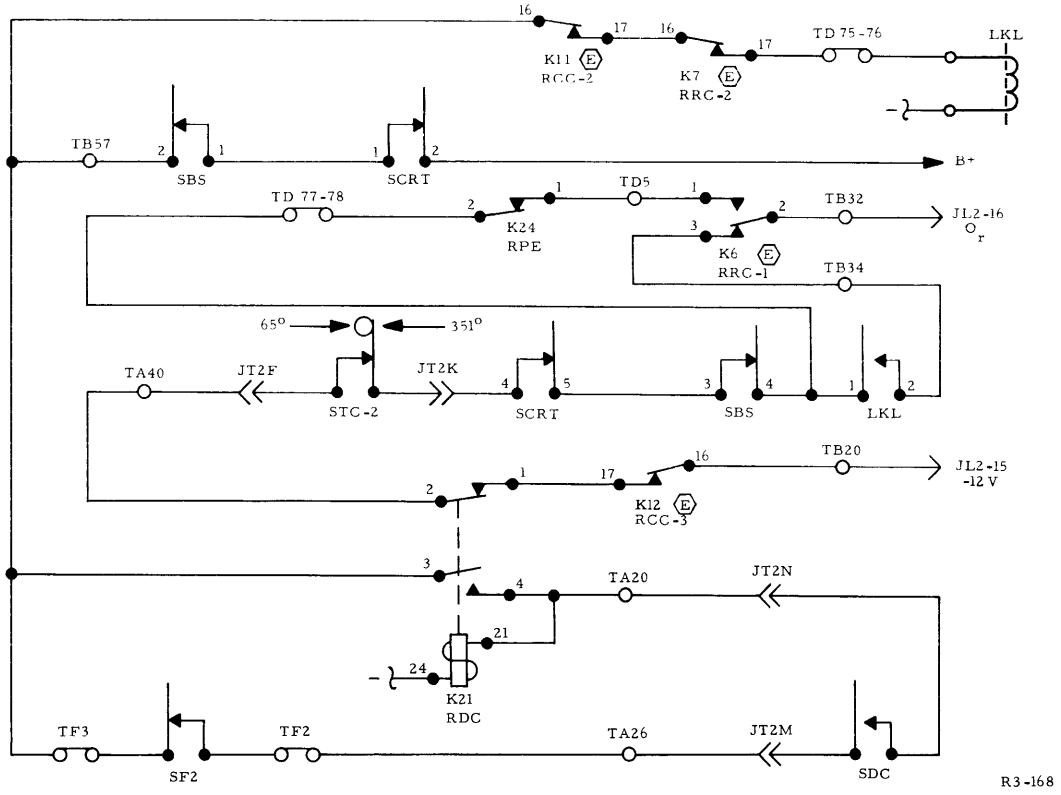


Figure 6-108. Delay Control Circuit (During Printing, Compute Mode).

Automatic Carriage Return (Local and Compute Modes)

In both local and compute modes, an automatic carriage return occurs when the carriage reaches a predetermined position in its travel. Automatic return is initiated by cams attached to a field switch actuating cam mounting plate at the rear of the carriage. The cams operate field switch actuators and are positioned such that they transfer field switch 2 (SF2) contacts slightly before field switch 3 (SF3) contacts.

Relay K21 is energized by the transferred SF2 contacts (figure 6-109) and then, held through its contacts 3 and 4 and contacts 1 and 2 of SCRT and 1 and 2 of SBS. Contacts 1 and 2 of K21 open, interrupting the O_r circuit, and contacts 13 and 14 open the LR circuit to halt reader action if it is operating.

The normally open contacts of SF3 close immediately after SF2 transfers and send the carriage return code to the translator through the normally closed contacts of K31. Both coils of K31 and K23 are then energized by the closing of SF3. The energizing of K31 opens its contacts and disrupts the carriage return code path to the translator magnets, thus allowing the magnets to be energized one time only. K23 is held energized through its contacts 3 and 4 and the normally open contacts 2 and 3 of SCRT. Normally closed contacts 1 and 2 of K23 are opened and interrupt the circuit of the SSC contacts, thereby stopping any code selector-to-punch operation which might be attempted.

Operation of the translator carriage return seeker initiates the mechanical operation of the carriage return, which transfers the SCRT contacts. When

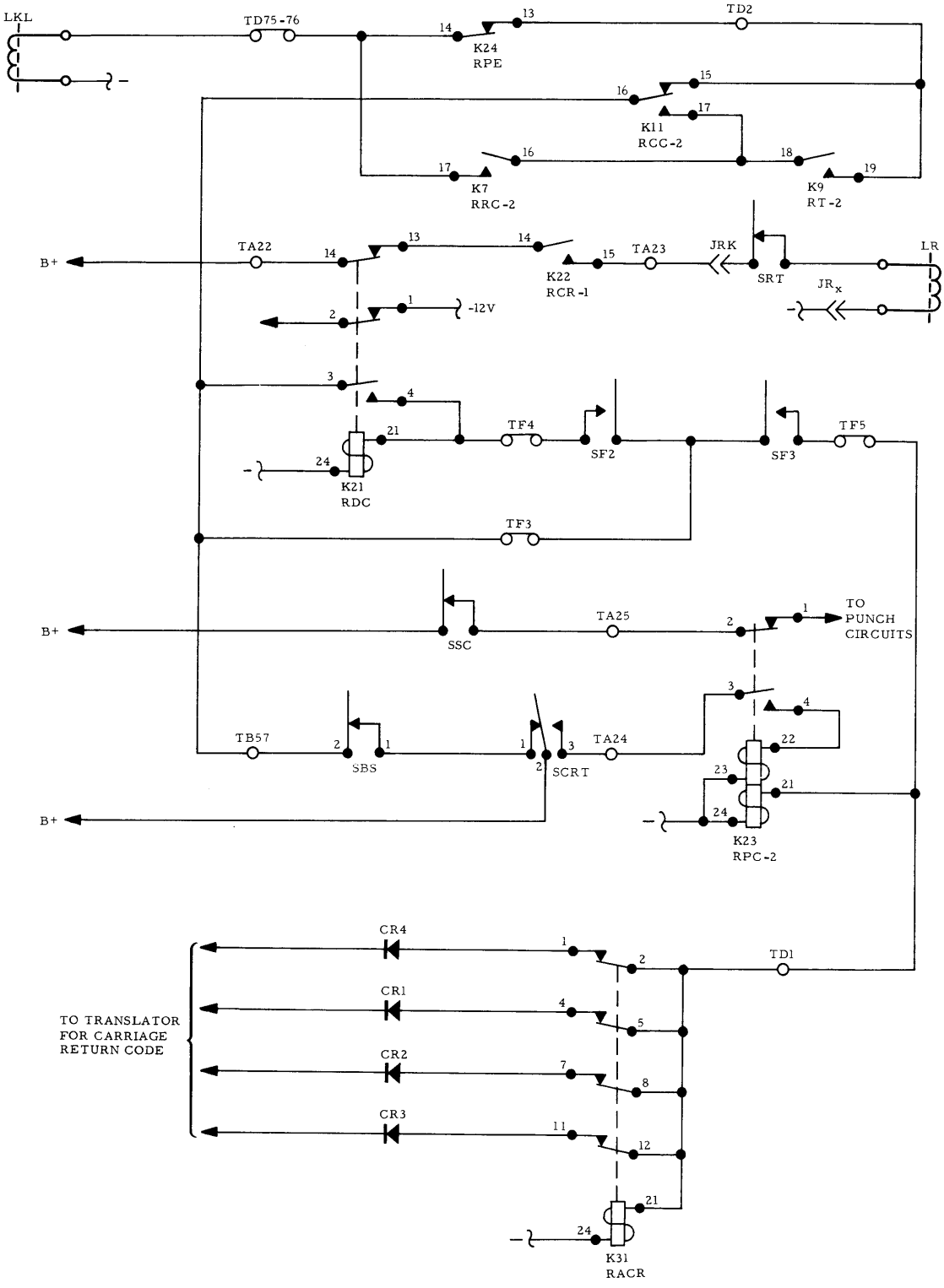


Figure 6-109. Automatic Carriage Return (Local-Compute Modes).

SCRT contacts 1 and 2 break, relays K21 and K31 are de-energized. However, relay K23 is held energized as previously described. The O_r circuit is held open until the completion of the cycle as a result of SCRT contacts 4 and 5 being open. At the close of the cycle the SCRT contacts return to their rest positions, and the O_r circuit is completed to enable a new cycle to be initiated.

Keyboard to Punch (Local Mode)

With switch S1 at ON, switch S2 at LOCAL, and switch S7 (PUNCH ON) depressed relay K5 is energized, closing circuits from normal open contacts SSC and SS1 through SS6 to punch clutch magnet and punch code magnets LP1 through LP6 (figure 6-110). Depressing a character or function keylever actuates the code selector and closes the SS contacts representing the code for that keylever. Immediately after the SS contacts transfer, the SSC contact transfers, ensuring completion of all circuits to the punch code magnets simultaneously.

Positive 48 volts (B+) is always present on the movable contact of SSC and is routed through normally closed contacts 1 and 2 of K23, SPL, 13 and 14 of K2, and 11 and 12 of K24 to LPC and the common bus of SS contacts 1 through 6. The B+ from the common bus passes through any closed SS contacts and contacts of relay K5 to the punch code magnets.

Because the punch operates faster than the printer and the code selector, an anti-repeat relay (K2) is provided. At approximately 90 degrees of punch unit operation, the SPL contacts close and energize K2 which is then held energized until SSC opens. If SSC fails to open by the time SPL returns to its rest position, relay K24 (relay punch error) is energized and its 11 and 12 contacts open to break the punch circuit to guard against a punching error. K24 can be de-energized only by opening SSC and depressing S9 or opening S1 momentarily.

Reader to Punch or to Punch and Printer (Local Mode)

To duplicate a tape only, switch S1 must be at ON, switch S2 at LOCAL, switch S7 (PUNCH ON) depressed, and switch S5 (NON-PRINT) depressed. If tape content printout is desired simultaneous with tape duplication, all switch settings are the same except S5 is not depressed and S3 (START READ) is depressed. However, the tape content must be in the RECOMP III 6-channel code for intelligible printout.

The RECOMP III 6-channel code is necessary for intelligible printout because the translator, which converts the coded electrical pulses into the mechanical motion needed to operate the keylevers, is capable of correctly translating this 6-channel code only. When punching and printing other than 6-channel information the punch would produce this error because the translator indirectly causes tape selector and, consequently, tape punch action and the printout would be unintelligible.

Reader to Punch Circuit Operation

Punch control relay K5 energizes, completing circuit path to the punch

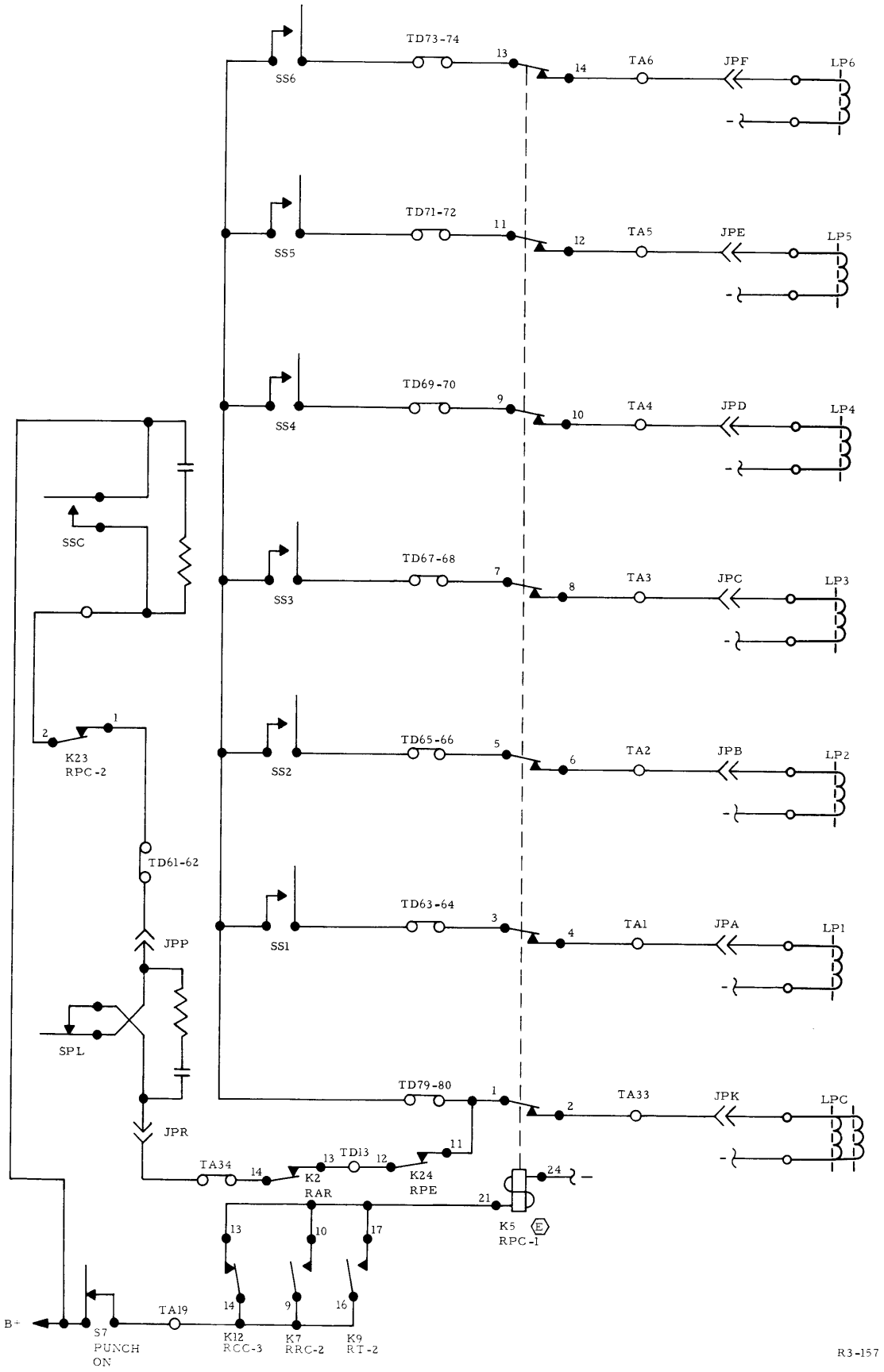


Figure 6-110. Punch Magnet Circuit (Local Mode).

clutch and punch code magnets (figure 6-111). Depressing switch S5 energizes non-print relays K3, K4, and K20 and reader control relay K22. The non-print relays, which are double-coil relays, are held energized by a holding circuit to one of the coils through the normally closed contacts of S4. With the non-print and punch control relays energized, output of the tape reader is sent directly to the tape punch.

Switch S4 (STOP READ) is adjusted to a make-before-break setting. If S4 should be depressed while duplicating a tape and it was not adjusted to a make-before-break setting, the holding circuit to the non-print and reader control relays would be broken and these relays would return to rest position. De-energizing of the non-print relays would then transfer the output of the reader to the punch instead of the translator. Relay K22 returning to its rest position opens the circuit to the reader clutch magnet LR. If the non-print relays return to their rest positions just as the reader starts a new reading cycle, the output of the reader is sent to the translator instead of the punch for completion of that particular cycle.

With the contacts of S4 correctly adjusted, the common contact closes with the normally open contact before breaking the holding circuit to the non-print and the reader control relays. Closing of the common contact with the normally open contact develops a second holding circuit to the non-print relays utilizing their other coils. A repeat depression of S4 causes opening of the normally closed contacts and breaks the holding circuit to the reader control relay (K22) and the first-energized coils of the non-print relays. The de-energizing of K22 opens the circuit to LR which halts reader action at completion of the cycle the reader is in at that moment.

If, at the instant S4 is depressed, the reader shaft has just passed its zero degree (rest) position, the reader completes this cycle. When S4 is released, the holding circuit to the second-energized coils of the non-print relay through the normally open contacts of S4 is broken. However, the non-print relays do not de-energize if the reader is sensing a code at this time, because the reader common contacts SRC are then transferring. The SRC contacts, which transfer each time the reader cycles, hold the non-print relays energized until completion of the sensing period. This holding action ensures that the last code sensed by the reader is transmitted to the tape punch and not to the translator.

Reader to Punch and Printer Circuit Operation

During simultaneous tape duplication and content printout, the information read by the tape reader is (1) routed to the translator and converted into mechanical action for printing, then (2) mechanical action of the printer is reconverted into electrical signals by the code selector and routed to the punch for punching. Circuit operation is the same as that described for reader-to-printer (local mode) and printer-to-punch (local mode) combined. Termination is accomplished by depressing S4, which de-energizes K22 and halts reading. However, because the non-print relays are not energized, a code sensed by the reader during depression of S4 is only printed.

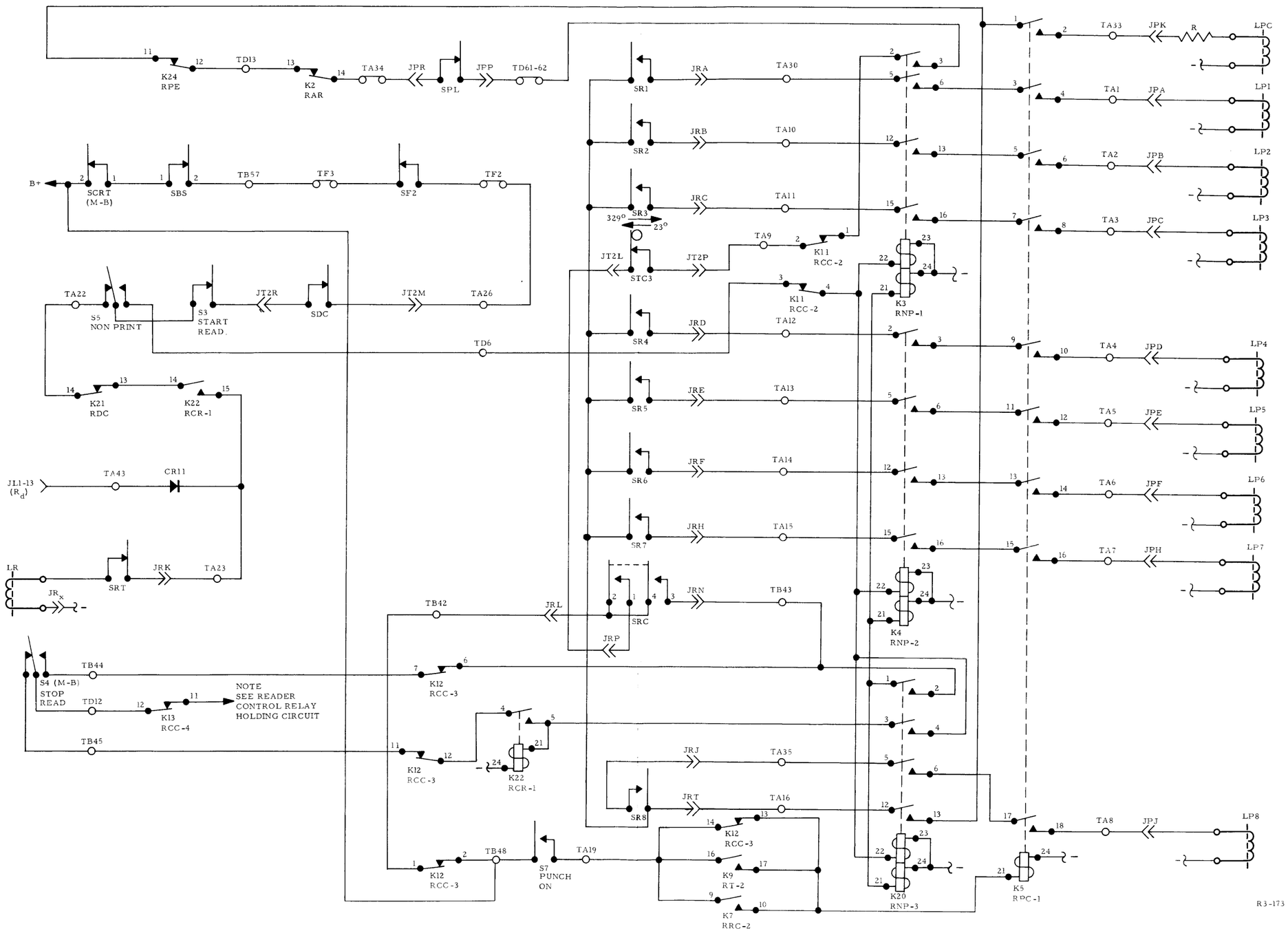


Figure 6-111. Reader to Punch Circuit (Local Mode).

Reader to Printer (Local Mode)

With switch S1 at ON, switch S2 at LOCAL, and tape in reader with loading gate closed, the reader tape contacts (SRT) will close, completing circuit path to LR (figure 6-112). Depressing S3 (START READ) energizes K22 (reader control relay) which is then held by a holding circuit through its contacts 4 and 5 and the normally closed contact of S4. Reader cycle-initiate magnet LR is energized through SRT and contacts on K22, K21, S5, S3, SDC, SF2, SBS, and SCRT. The B+ current for LR passes through these contacts to control reader operation.

Reading stops if the reader loading gate is opened or if a carriage return, tab, backspace, or stop code is read. However, after completion of a carriage return, tab, or backspace operation, the reader resumes operation because K22 is still energized. If a stop code was read, START READ switch S3 must be depressed to resume reader operation. Reading can also be stopped manually by depressing S4 (STOP READ) which de-energizes K22. Resumption of operations after de-energization of K22 requires depression of S3.

Immediately after the reader contacts transfer, the reader common contacts (SRC), which have B+ supplied to them through contacts 1 and 2 of K12, transfer and energize LTC and LT7 through STC-3. The B+ through STC-3 is also supplied to the common buss of reader contacts SR1 through SR8. From the reader contacts, the B+ goes to the translator magnets through the normally closed contacts on the non-print relays. The translator then converts the code holes sensed by the reader pins into the mechanical motion needed to produce printed copy.

Delay Control During Printing (Local Mode)

When the reader is sending information to the printer, reader operation must be interrupted momentarily when completion of any printer operation takes longer than one ordinary reader cycle. This delay is necessary to prevent the reader from causing two operations of the printer to overlap. Printer operations which require longer than one reader cycle are carriage return, tabulation, and backspace.

Delaying of the reader is accomplished by opening the circuit to reader magnet LR (figure 6-113) without de-energizing relay K22. Delay control is accomplished by three operations of the circuit (delay control qualifying, primary delay, and function control delay) which are described in the following paragraphs.

Delay Control Qualifying (Local Mode).

Delay control qualifying is accomplished by the operation of the translator seekers whose respective keylevers will require longer than one ordinary reader cycle to complete. These seekers operate the S DC (switch delay control) which opens the circuit to the reader clutch prior to 360-degree rotation of the reader shaft, or the point at which LR control arm is offset by a cam. In addition, SDC causes K21 to energize.

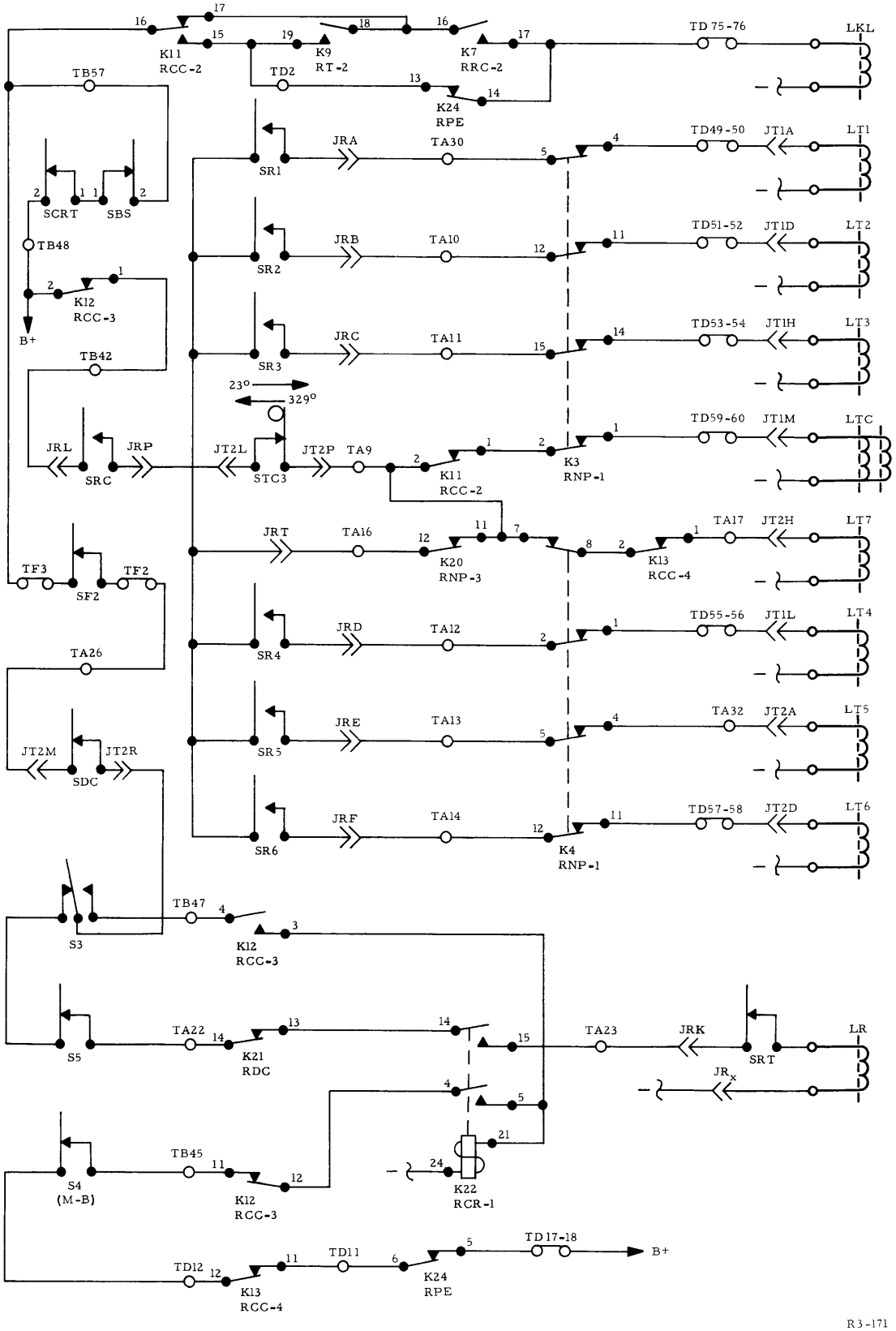


Figure 6-112. Reader to Translator Circuit. (Local Mode).

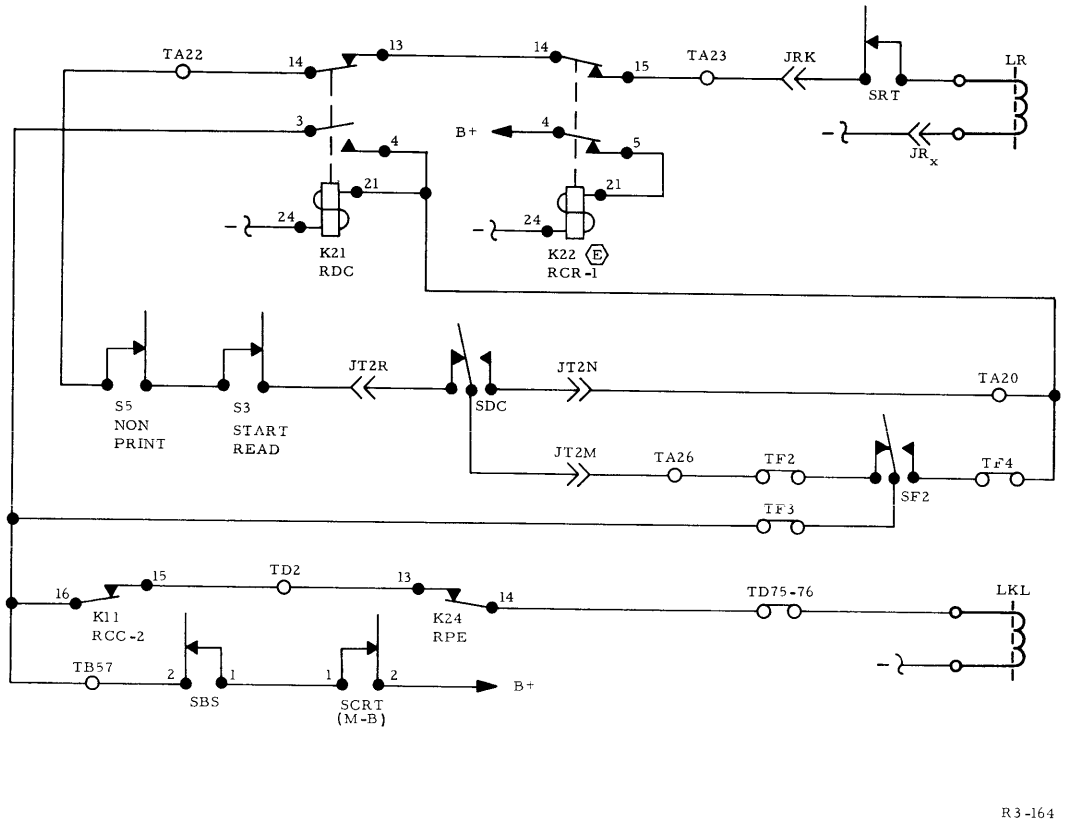
Primary Delay (Local Mode)

The primary delay is accomplished by relay K21. SDC is operated for only about 10 milliseconds, K21 must continue to open the circuit to LR after SDC is restored. A holding circuit through contacts 3 and 4 keeps K1 energized through completion of the operation of the keylever which requires delay.

Function Delay (Local Mode)

If, for example, a carriage return code is read by the reader, SDC initially energizes K21 which remains energized through its own contacts 3 and 4 and through the normally closed SCRT contacts 1 and 2. As the carriage return keylever is operated the SCRT contacts transfer, opening the circuit to K21. SCRT continues to hold the LR circuit open for the duration of the carriage return function. When the carriage reaches the end margin, SCRT is restored by the unlatch cam, and the circuit to LR is completed allowing the reader to continue normal reading.

Delay control operation is identical during backspace and tab operations except either SBS or SCRT will transfer depending on the operation selected. Transferring of SCRT or SBS also breaks the circuit to LKL which disables keyboard operation during tab, carriage return, and back space operations.



R3-164

Figure 6-113. Delay Control Circuit (During Printing, Local Mode).

Tape Feed (Local Mode)

The tape feed circuit permits the operator to place a length of leader tape containing no information at the beginning or end of a program, or to separate groups of information on tape with areas of tape containing no information. Code for tape feed causes punching of sprocket holes only.

Depressing switch S7 energizes relay K5 (figure 6-114). When switch S8 is depressed, B+ goes to the punch clutch magnet through contacts 13 and 14 of relay K11, contacts 13 and 14 of relay K2, contacts 11 and 12 of relay K24, and contacts 1 and 2 of relay K5. Tape continues to feed until S8 is released. S8 is inoperative in the compute mode.

Punch STOP CODE (Local Mode)

Depressing the STOP CODE switch (S10) energizes relay K26 which is then held through its normally open contacts 1 and 2 (figure 6-115). The B+ is sent to the SPL common contact through normally open contacts 5 and 6 of K26. From the normally closed SPL contact, B+ is supplied to the punch clutch magnet and through three sets of normally open contacts of K26 to the punch code magnets corresponding to the stop code.

One stop code is punched for each depression of switch S10 because the anti-repeat relay was energized from the STOP CODE switch instead of the selector common contact.

Punch CODE DELETE Code (Local Mode)

An incorrect code punched while preparing a tape can be overpunched with the delete code by placing the incorrect code on the tape over the punch pins and depressing the CODE DELETE switch. The code delete circuit then causes all six channels of the tape to be punched which, when read by the tape reader, will be ignored. Energization of the anti-repeat relay from the CODE DELETE switch prevents more than one code delete cycle even though the switch is held depressed.

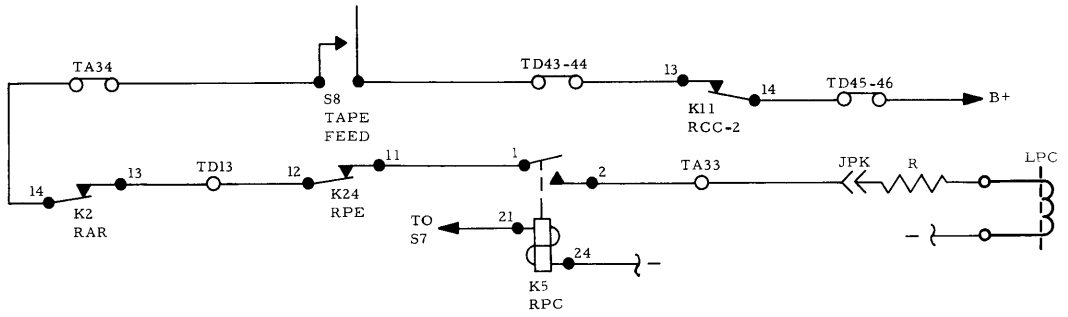


Figure 6-114. Tape Feed Circuit (Local Mode).

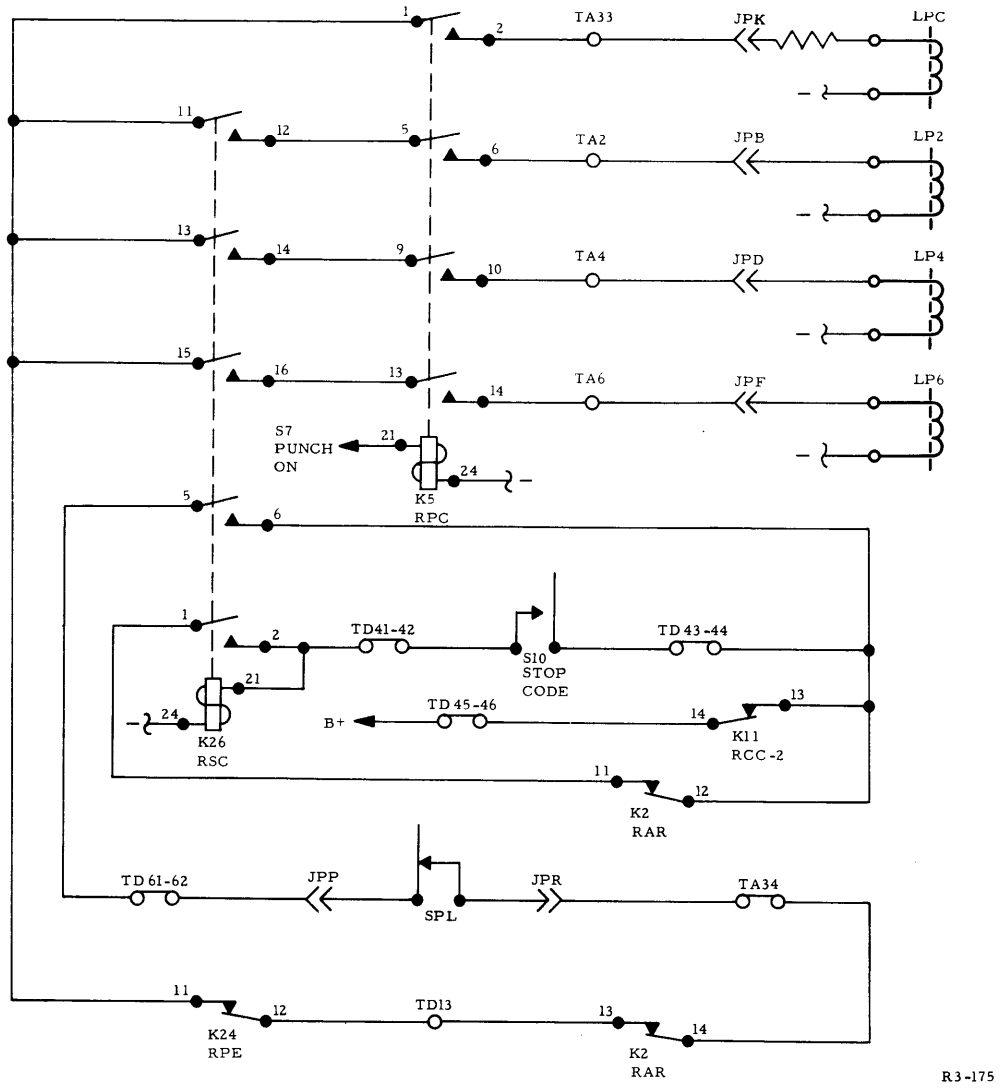


Figure 6-115. Punch STOP CODE Circuit (Local Mode).

Multiple CODE DELETE codes can be punched, however, by simultaneously holding both the TAPE FEED and CODE DELETE switches depressed. Punching of multiple CODE DELETE codes is possible because the anti-repeat relay is not energized, S8 keeping SPL energized which in turn opens the circuit to K2. Opening of this circuit causes normally closed contacts 13 and 14 of K2 to remain closed.

When punching one CODE DELETE code, depressing switch S9 energizes relay K25, which is then held through its contacts 1 and 2 (figure 6-116). The B+ from the normally closed tape feed contacts, through the normally open contacts 6 and 7 of relay K25, is applied to the SPL common contact. From the normally closed SPL contact, B+ is then sent to the punch clutch magnet and the common contacts of six normally open contacts of relay K25. When relay K25 is energized, punch code magnets LPI through LP6 are energized through the normally open contacts of relays K25 and K5. Operation of

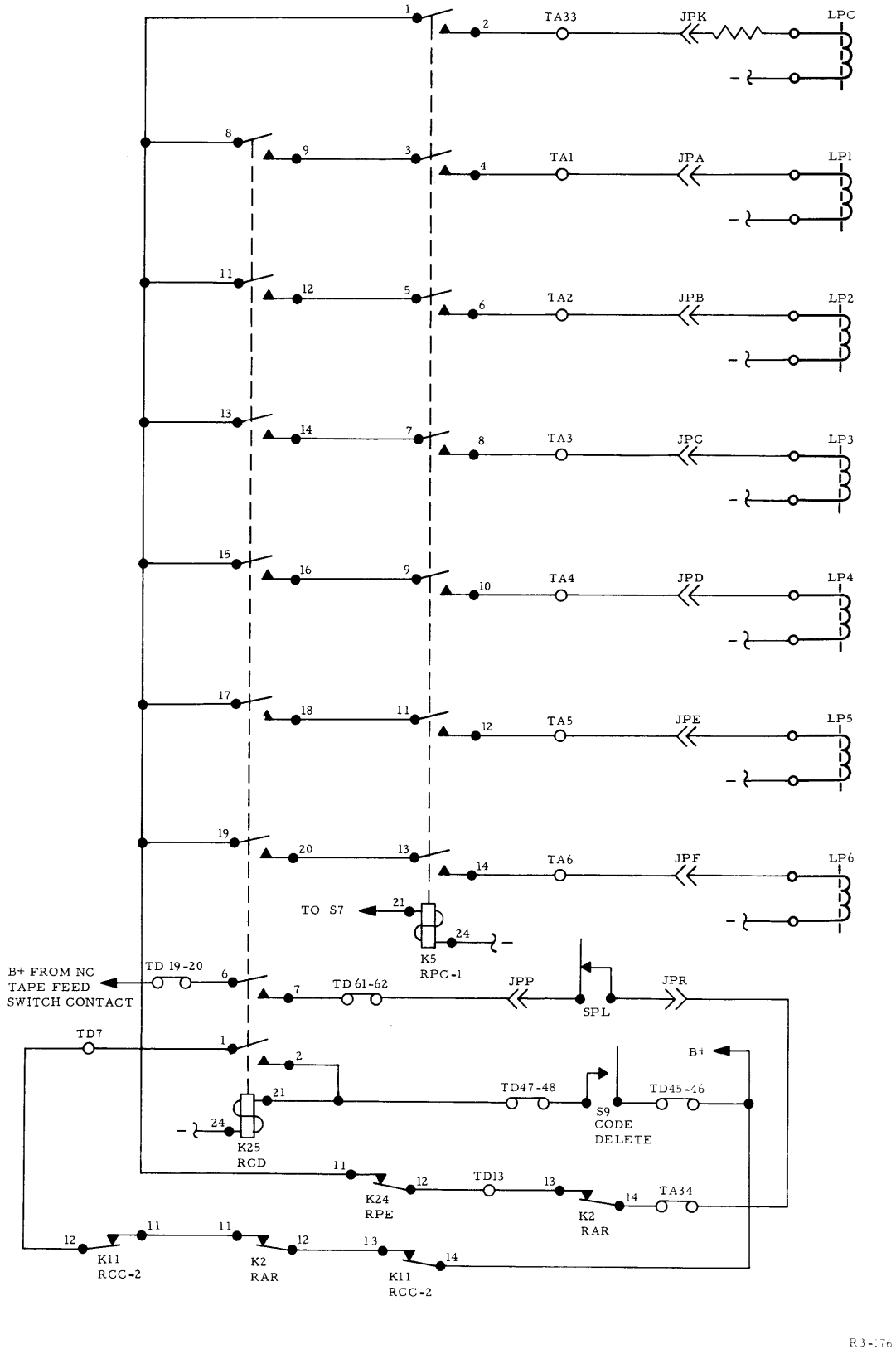


Figure 6-116 Punch CODE DELETE Code Circuit (Local Mode).

anti-repeat relay K2 from the CODE DELETE switch permits only one CODE DELETE code to be punched. Upon release of switch S9, both relays K25 and K2 are de-energized.

Relay Holding and Ignore Stop (Local Mode)

The reader control relay holding and ignore stop circuitry provides the means of (1) stopping reader action if a stop code is read by the reader; (2) continuing reader operation, although a stop code is read, if the IGNORE STOP switch is depressed; or (3) continuing reader operation although a stop code configuration (2-4-6) in another character is read and the IGNORE STOP switch is or is not depressed. In effect, this last capability constitutes an override of the stop code configuration during reading of characters containing this configuration as part of their codes.

Depressing S3 (figure 6-117) energizes relay K22 which is then held energized with a holding circuit established by the normally closed contacts in reader contact groups SR2, SR4, and SR6. If stop code holes (2-4-6) only are read by the reader, the holding circuit to K22 is broken, de-energizing reader magnet LR and stopping reader operation.

The IGNORE STOP switch (S6) is in parallel with reader contact groups SR2, SR4, and SR6. If S6 is depressed, the reader continues to operate even though a stop code is read by the reader.

When a 2-4-6 configuration constitutes only a portion of the binary-coded ones in a character, the reader continues its operations because one of the normally open contacts in reader contact groups SR1, SR3, and SR5 (which are in parallel with contacts SR2, SR4, and SR6) is operative.

Flexowriter Power Supply

Power for operating all relays and magnets of the Flexowriter is obtained from its own self-contained power supply (figure 6-118). This power supply is comprised of a step-down (115 v-48v) ac power transformer and a full wave selenium rectifier with an output of approximately 48-volts dc. With the power cord plugged into the ac power receptacle in the desk well, a 115-volt ac potential is applied across the input terminals of the transformer when power switch S1 is placed at ON. The dc supply circuit is protected by a 2-ampere slo-blo fuse (F2). Two 0.1-ufd, 200-volt capacitors are included in the power supply circuit to prevent transient voltages in the dc supply, caused by operation of switch S1, from reaching the computer.

Output of the power supply is sent to the computer through JL1-15 and JL1-17 for computer control of various relays and magnets in the Flexowriter. An auxiliary ac outlet is provided at the rear of the tape punch to supply power to the tape rewind unit. This auxiliary outlet should not be used with any equipment drawing over 10 watts.

Motor Circuit

The motor circuit (figure 6-119) consists of a 35-millihorsepower

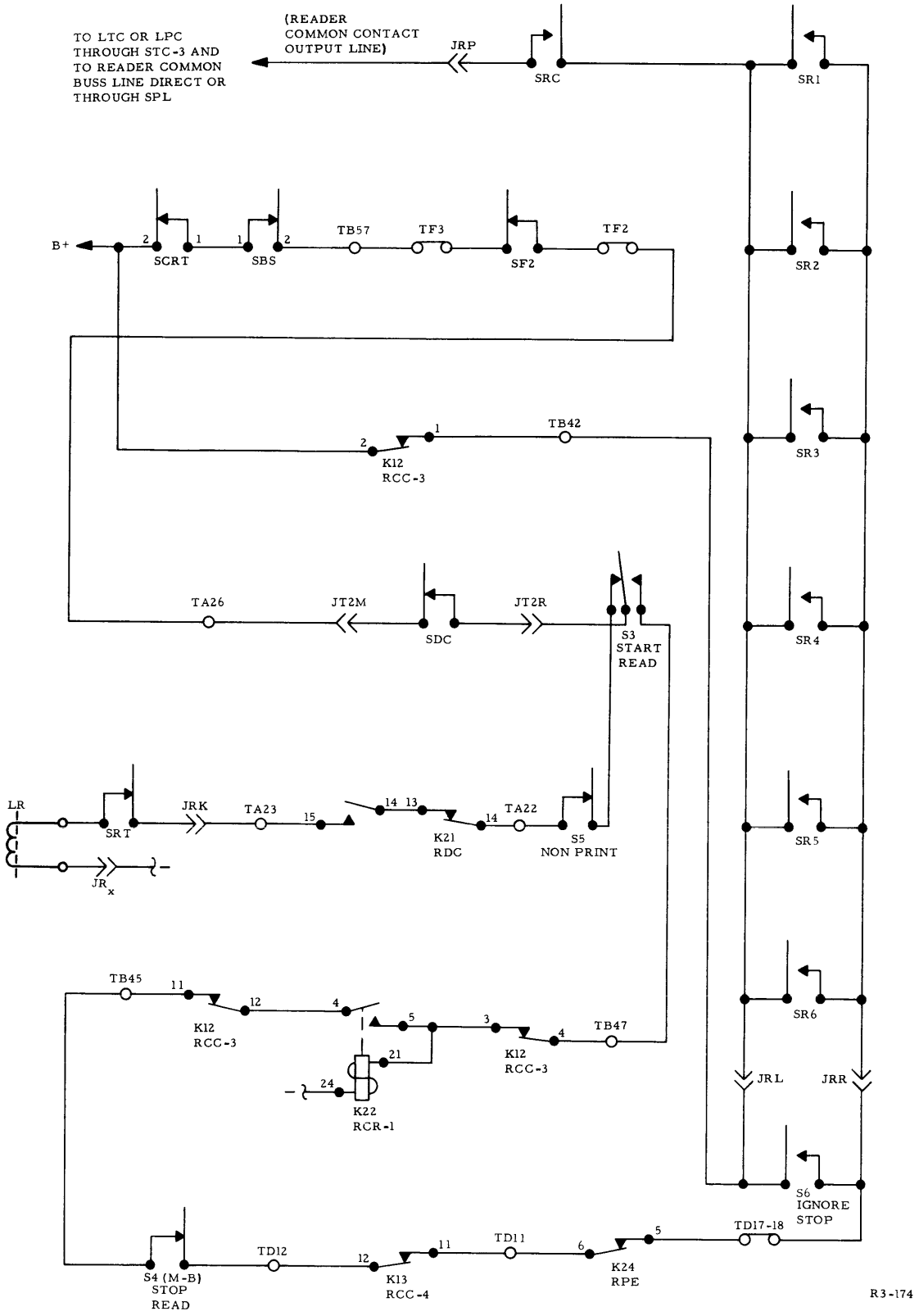


Figure 6-117. Reader Control Relay Holding and Ignore Stop Circuits (Local Mode).

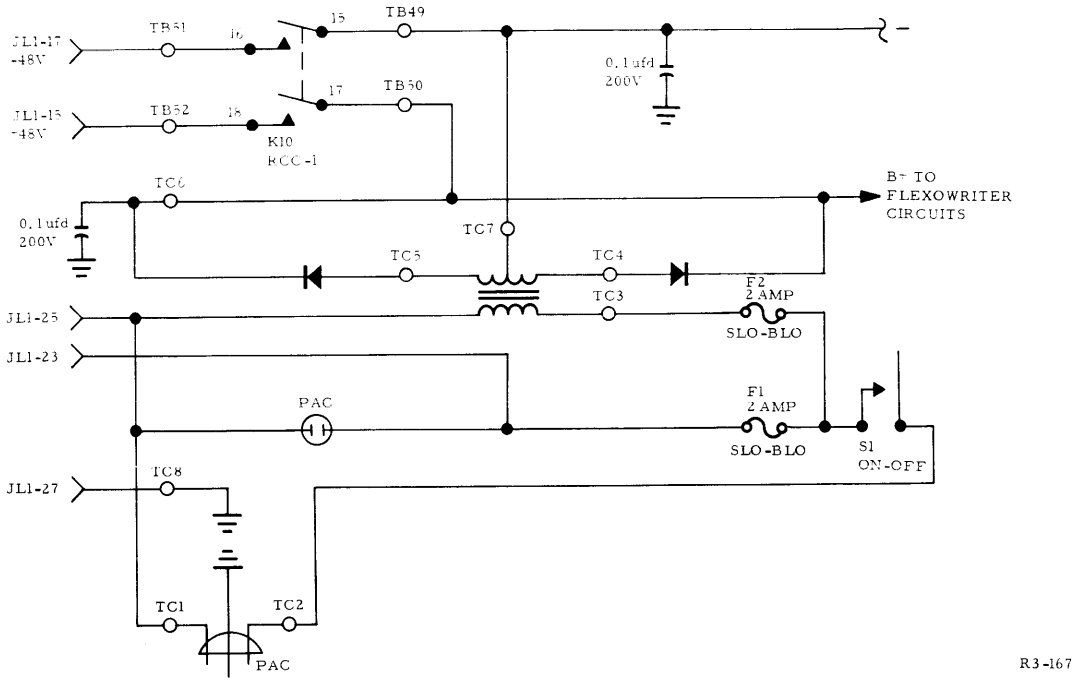


Figure 6-118. Power Supply.

constant speed motor (1750 RPM) operating on 115 volts ac, a power switch located on the right side of the keyboard, a two ampere fuse for circuit protection, and a starting relay that controls a starter winding in the motor.

Designated S1, the power switch controls the 115-volt line input to the power supply and motor. S1 is a single-pole, single-throw toggle switch mounted on the switch panel to the right of the keyboard. Closing S1 completes the 115-volt circuit to the main windings of the motor through the coil of the starting relay. The starting relay, designated K1, is a differential relay which is energized and de-energized by different levels of current flow through its coil.

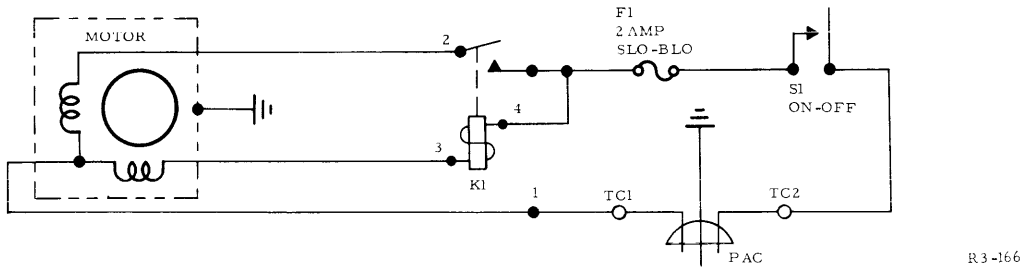


Figure 6-119. Motor Circuit.

The initial surge of current through the motor energizes relay K1. This coil is in series with the main winding of the motor, and, when energized, its contacts 2 and 4 complete a circuit to the starter winding of the motor which supplies the necessary added starting torque. Contacts 2 and 4 of K1 remain closed until the motor reaches its running speed of 1750 RPM. At this speed the current drawn by the motor, flowing through coil of K1, is not sufficient to hold K1 contacts closed. The motor then operates by the circuit through its main windings only.

Table 6-6. Glossary of Flexowriter Circuit and Signal Designations

A. MAGNETS

Designation	Definition
LKL	Keyboard lock (operates one set of normally open contacts)
LP	Punch code (eight coils in parallel)
LPC	Punch clutch (two coils in parallel)
LR	Reader cycle initiate
LT	Translator code (seven coils in parallel)
LTC	Translator clutch (two coils in parallel)

B. RELAYS

K1	Motor Start
K2 (RAR)	Anti-repeat
K3, 4, 20 (RNP-1, 2, 3)	Non-print
K5, 23 (RPC-1, 2)	Punch control
K6, 7, 14 (RRC-1, 2, 3)	Output select, punch or translator
K8, 9 (RT-1, 2)	Input select, reader or keyboard
K10, 11, 12, 13 (RCC-1, 2, 3, 4)	Computer control
K21 (RDC)	Delay control
K22, 27 (RCR-1, 2)	Reader control (1, Local; 2, Compute)
K24 (RPE)	Punch error
K25 (RCD)	Code delete
K26 (RSC)	Stop code
K28 (RRS)	Reader stop (during Compute only)
K31 (RACR)	Automatic carriage return

C. CONTACTS

LKL	Keylock
SBS	Backspace

C. CONTACTS (Continued)

Designation	Definition
SCRT	Carriage return and tab
SDC	Delay control (translator)
SF	Field (two sets of contacts)
SPL	Punch latch
SPT	Punch tape
SR	Reader code
SRC	Reader common
SRT	Reader tape
SS	Selector slide
SSC	Selector slide common
STC	Translator (four sets of contacts)

D. SWITCHES

S1	ON-OFF Power (motor and power supply)
S2	LOCAL-COMPUTE
S3	START READ
S4	STOP READ
S5	NON-PRINT
S6	IGNORE STOP
S7	PUNCH ON
S8	TAPE FEED
S9	CODE DELETE
S10	STOP CODE

E. MISCELLANEOUS

PAC	Auxiliary ac outlet, 10 watts maximum capacity (for tape rewind unit)
PL	INPUT panel light

F. SIGNALS TO COMPUTER

Designation	Definition
K	Printer chosen as output device
$K_{1(f)}$ to $K_{8(f)}$	Information input lines to computer
O_r	Output ready
P	Punch chosen as output device
R_{gf}	Origin-sector channel read enable
T_r'	Selector slide contacts not transferred (keyboard and tape reader at rest)
T_r	Selector slide contacts transferred (keyboard or tape reader active)
T_p'	Printer and tape punch at rest
T_p	Printer or tape punch in operation

G. SIGNALS FROM COMPUTER

$(C_5), (C_6), (D_1)$-(D_6)	Eight information lines from computer flip-flops transferring amplified, coded output signals to either translator or punch
(P_{kd})	Output drive signal for printer or tape punch, composed of $(\underline{O_d})O_r PC_4'C_3'C_2 + (\underline{O_d})O_r KC_4'C_3'C_2'$
(P_{ks})	Selects printer for output, composed of $I_4I_2I_1'C_4'C_3'C_2'$
(R_d)	Tape reader drive signal, composed of $I_4I_2'I_1'C_4'C_3'C_2$
(R_{ks})	Selects keyboard for input, composed of $I_4I_2'I_1'C_4'C_3'C_2'O_r$

RECOMP III - POWER ON

The following is a detailed description of the computer process from the moment the "Power - ON" switch is depressed until we are ready to initiate the "INTERNAL BOOTSTRAP" routine.

There are many operations that are taking place simultaneously. This renders it virtually impossible to describe the operations with the utmost clarity in their actual sequence. As much as practicable we will follow the exact sequence.

Sequence I - Power ON Relay Sequence

Sequence II - Logic sequence for synchronization.

A - Computer signal T_d

"ERROR RESET" button.

As long as K6 is de-energized Td is available to the computer at pin #10 of K6 relay. K6 remains de-energized for the first 40 seconds after the "ON" button was depressed.

During the first 30 seconds (after the "ON" button was depressed) the Memory Motor is getting the disk up to speed. At the end of 30 seconds, relay K4 energizes and activates the Memory Solenoid. This positions the disk near the headplate.

The CLOCK and Sector Channel readouts are available; "Td" is still available for 10 seconds before K6 energizes and clamps it at a ground potential. (When K6 is de-energized Td is "floating" and appears as a TRUE input at the gate.)

It is during these 10 seconds that the computer is properly synchronized.

At the end of this 10 second period K4 is energized.

1. The "Ready Neon" is turned ON.
2. K6 is energized.
3. The computer is in IDLE.
4. The Main Memory Write switches are turned "ON" by K6.

Our concern, at this time, is to understand the computer logic processes that occur during the 10 second interval.

The Initial Logic Operation of the computer is

$1^{I}4 = Td$	$0^{I}3a = Td$	$0^{C}4 = Td$
$1^{I}2 = Td$	$0^{J}o = Td$	$0^{C}3 = Td$
$1^{I}1 = Td$	$0^{R}43 = Td$	$0^{C}2 = Td$
$0^{K}c = Td$	$1^{N}5 = Td$	$0^{C}1 = Td$

The configuration of $I_4\ I_2\ I_1\ K_c'$ forces us into the SYNC MODE - phase 2 or 3. (Since "Do" is not immediately specified it can be either phase 2 or phase 3.)

During the SYNC MODE we will:

1. Obtain synchronization of our "bit counter" ($P_6 \longrightarrow P_1$) with the output of the SECTOR CHANNEL.
2. Set the Location Counter to zero (0).
3. Zero the Neon Display on the Control Console.
4. Terminate the SYNC MODE and proceed to the IDLE MODE.

SYNC MODE - OPERATION #1

The P-counter logic is such that initial synchronization will take place only after SECTOR - 77_8 has been read from the SECTOR CHANNEL.

Sector 77_8 is the only word in the SECTOR CHANNEL, that contains all zeroes -

(Notes: - Reference - RECOMP III - Technical Bulletin #1, page 6)

Since the other "words" all contain pertinent information;

S_1 (the L.S.B. READ Flip-Flop - of the SECTOR CHANNEL will be continually forced TRUE EACH WORD TIME.)

This will: - $1^{T}41 = S_1 \underline{I_q} I_2$ $\quad\quad$ $\underline{I_q} = I_4 I_1$

And $0^{Do} = T_{41} I_1$ Do - Forcing phase 2 (if we were not already there)

T_{41} continually RESETS the P-counter

$0^{P}6 = T_{41}$ $\quad\quad$ $0^{P}3 = T_{41}$

$0^{P}5 = T_{41}$ $\quad\quad$ $0^{P}2 = T_{41}$

$0^{P}4 = T_{41}$ $\quad\quad$ $0^{P}1 = T_{41}$

When "Sector 77_8" is read, S_1 remains prime. The P-counter is allowed to count up until T_{40} is reached. ($\underline{T_{40}} = P_6 P_5 P_4 P_3' P_1'$)

$\underline{T_{40}}$ can only be reached during the readout of sector 77_8.

The Sync mode is then terminated by

$$0^{I}2 = \underline{I_q} \underline{T_{40}}$$

OPERATION 2 - Zero the Location Counter

During the LAST WORD TIME of the SYNC MODE the location counter (Bits 22 through 34) are ALL RESET TO ZERO.

This is accomplished by: $0^{G}41 = \underline{I_q}\ I_2\ P_6^*$

Left Hand G-register information is recirculated via:

$$1^{G}41 = G_1\ \underline{I_q}\quad P'_6$$

$$0^{G}41 = G'_1\ \underline{I_q}$$

(Note: It is incidental that the CHARACTER COUNTER is also zeroed at this time).

Normal Recirculation Logic $\quad 1^{G}41 = G_1\ K'_g\ T'_{41} \qquad 1^{K}g = \underline{I_q}\ I_2$

$$0^{G}41 = G'_1\ K'_g\ T'_{41}$$

*P_6 could not come true until the last word time.

The LOCATION COUNTER is correctly RESET

Note ------ Brief explanation of NEON INDICATORS

1. Neon is "ON" unless -12 volts is present at pin #1
2. Prime side of $D_6 \longrightarrow D_1$ and $C_6 \longrightarrow C_1$ and R_{43} are wired to the NEONS.

OPERATION 3 -

ZERO THE NEON DISPLAY on the CONTROL CONSOLE.

The D-and C-registers control the Neon Indicator.

Resetting the D-and C-registers will zero the display

$$0^{D}6 = I_4\ N_5\ K'_a$$

$$0^{D}5 = D'_6\ N_5$$

$$0^{D}4 = D'_5\ N_5$$

$$0^{D}3 = D'_4\ N_5$$

$$0^{D}2 = D'_3\ N_5 \qquad 0^{C}6 = D'_1\ N_5$$

$$0^{D}1 = D'_2\ N_5 \qquad 0^{C}2 = C'_6\ N_5$$

$$0^{C}4 = 0^{C}3 = 0^{C}2 = 0^{C}1 = 0^{R}43 = I^{N}5 = Td$$

$$0^{K}a = S'\ I_4\ T_{41} \qquad 0^{S} = I_4\ \underline{T_n}\ T_{41}$$

$$\underline{T_n} = T'_q\ T'_r\ T'_{fp}\ T'_{fr}$$

It appears that the NEON DISPLAY could be reset before this last word time.

It is really impracticable to determine because N_5 is continually being double triggered by:

$$0^{N}5 = T_{41} \qquad \& \qquad 1^{N}5 = Td$$

Both occuring during the 10 second period.

It is accurate to state that during the last word time the $D_6 \longrightarrow D_1$, C_6 and C_5 are positively reset because T_{41} is inhibited for an entire word time.

We have spoken of this last word time in Sync as though it were an absolute certainty.

$0^{I}2 = \underline{I_q}\ \underline{T_{40}}$ is correct logic, BUT

UNTIL THE TEN (10) SECOND PERIOD HAS EXPIRED WE WILL REMAIN IN THE SYNC MODE.

Because $1^{I}2 = Td$ which is present until K6 is energized.

K6 energized grounds Td.

Transition is now to the IDLE-MODE.

COMMAND MODE

Transition from the IDLE MODE to the COMMAND MODE will always be into Phase 1. Under these circumstances our sequence will always be

Phase 1; Check the G-register for "MAIN MEMORY" or "LOOP" address designation. Terminate after one word-time. Transition is to Phase 2.

Phase 2; Search for SECTOR AGREEMENT between the G-register and the Sector Channel outputs. When Sector Agreement is reached, terminate the Phase and go to Phase 3.

Phase 3; Read one (1) word from Main Memory or the loop INTO THE B-REGISTER (this word REPRESENTS THE COMMAND PAIR) Terminate after one word-time. Transition is ALWAYS TO PHASE 1 of the OPERAND MODE.

PHASE 1:

Phase 1 is entered by $0^{I}4 = I_4 \underline{K_{41}}$

During Phase 1 we will analyze the contents of the LOCATION COUNTER. From this analysis we will determine if the address of the COMMAND-PAIR is a MAIN MEMORY or a LOOP ADDRESS. (THE LOCATION COUNTER SPECIFIES INSTRUCTIONS ADDRESS ONLY).

The designation of loop or Main Memory is given us by K_a.

K_a = Loop Address

K'_a = Main Memory

K_a is "SET" at T_1 of Phase 1.

$1^{K}a = I_1\ K_c\ T_1$ = ASSUME "LOOP ADDRESS" INITIALLY.

$+P'_5\ P'_3\ P'_2\ K_c\ I_1\ G'_o$

We then monitor bit positions 27⟶34 for ZERO (0) bits.

Zero bits in positions 27⟶34 = MAIN MEMORY

$0^{K}a = Z'_{41}\ I_1\ N_1\ K_c$

N_1 = Timing Flip-Flop for Address Analysis

$1^{N}1 = G'_o\ \underbrace{P_6\ P'_5\ P'_3\ P_2\ P_1}_{T\ 27}\ K_c\ D'_o \quad + \underbrace{P'_6\ P'_5\ P'_3\ P_2\ P_1}_{T\ 7}\ K_c\ D'_o\ I_1$

$0^{N}1 = \underbrace{P_5\ P_4\ P_2}_{T35\ T15}\ K_c\ D'_o$

N_1 = TRUE = T_8 true T_{15} and T_{28} true T_{35}

The contents of the G-register are copied by the Z-register via Z_{41}: -

$1^{Z}41 = G_1\ I_c\ K_c\ D'_o$

$0^{Z}41 = G'_1\ I_c\ K_c\ D'_o$

G'_o will be true throughout the COMMAND MODE via: -

$0^{G}o = \underline{I_c}\ D'_o + \underline{I_c}\ N'_1\ T_{41}$

The S - flip-flop detects which LOOP is specified by monitoring bit 26.

$$1^S = C_6\ N_5\ I_1\ \underline{T_{13}}$$

$$0^S = C_6'\ N_5\ I_1\ \underline{T_{13}}$$

$$\underline{T_{13}} = \underbrace{P_5'\ P_3\ P_2\ P_1'}_{T_{13}\ or\ T_{33}}$$

At the same time the G-register is being copied by the Z-register via Z_{41}; the CHANNEL BITS are going to be shifted into the C-register.

The only way information can be shifted into the C-register (except FLOATING POINT & INPUT-OUTPUT operations) is that it joins the D-register to form a twelve (12) bit shift register.

($C_6 \longrightarrow C_1$ is the LEAST SIGNIFICANT HALF.)

N_5 = Timing flip-flop

$$1^N5 = \underline{I_c}\ K_c\ \underline{T_{22}}$$

$$0^N5 = I_4'\ \underline{T_{20}}$$

$$\underline{T_{22}} = P_6\ P_5'\ P_3'\ P_2'\ P_1$$

$$\underline{T_{20}} = P_5\ P_4\ P_3'\ P_1'$$

$$1^D6 = G_1\ \underline{I_c}\ N_5 \qquad 1^D5 = D_6\ N_5 \qquad 1^D4 = D_5\ N_5$$

$$0^D6 = G_1'\ \underline{I_c}\ N_5 \qquad 0^D5 = D_6'\ N_5 \qquad 0^D4 = D_5'\ N_5$$

$$1^D3 = D_4\ N_5 \qquad 1^D2 = D_3\ N_5 \qquad 1^D1 = D_2\ N_5$$

$$0^D2 = D_4'\ N_5 \qquad 0^D2 = D_3'\ N_5 \qquad 0^D1 = D_2'\ N_5$$

$$1^C6 = D_1\ N_5 \qquad 1^C5 = C_6\ N_5 \qquad 1^C4 = C_5\ \underline{N_7}$$

$$\underline{N_7} = I_4'\ N_5$$

$$0^C6 = D_1'\ N_5 \qquad 0^C5 = C_6'\ N_5 \qquad 1^C4 = C_5'\ \underline{N_7}$$

$$1^C3 = C_4\ \underline{N_7} \qquad 1^C2 = C_3\ \underline{N_7} \qquad 1^C1 = C_2\ \underline{N_7}$$

$$0^C3 = C_4'\ \underline{N_7} \qquad 0^C2 = C_3'\ \underline{N_7} \qquad 0^C1 = C_2'\ \underline{N_7}$$

(Bit 26, will read out of C_6 at T_{33} ---------- Monitored by S.)

4A SUPPLEMENT

The recirculation of the G-register is:

$$1^{G}41 = G_1 K'_g T'_{41}$$

$$0^{G}41 = G'_1 K'_g T'_{41}$$

where $0^{K}g = \underline{T_{20}}$

$$1^{K}g = G'_o I'_2 N_5 P_1 P_6 \underline{T_{14}} + G'_o I'_4 P_6 P_1 \underline{T_{14}}$$
$$T_{34} \qquad\qquad T_{34}$$

K_g is reset at T_{40} of IDLE and will not come true until T_{35} of Phase 1.

Only the left-hand Character Counter is affected.

By the end of the word time we have

1. Shifted G-register into the Z-register.
2. Set "K_a" to indicate LOOP or MAIN MEMORY address.

   K_a = LOOP

   K'_a = MAIN MEMORY
3. Shifted CHANNEL NUMBER into the C-register.
4. Set "S" to indicate WHICH LOOP was selected.

   S = V - LOOP

   S' = L - LOOP
5. Phase 1 is terminated after ONE WORD TIME

   WE NOW: - $0^{R}42 = \underline{K_{41}}$

   $0^{S}o = \underline{T_{41}}$

   $0^{K}c = I'_4 I_1 \underline{K_{41}}$

   and continue R_{43} = True; which was $1^{R}43 = I'_4 I_1 T_1$

TRANSITION IS IMMEDIATELY to Phase 2.

Phase 2 is entered by $0^{K_c} = I_4' \; I_1 \; \underline{K_{41}}$

During Phase 2 we will compare the Sector portion of the Z-register (Bits 23 through 28) at Z_1 with the SECTOR CHANNEL readout of S_1.

When all six (6) bits agree we will consider that SECTOR AGREEMENT has been accomplished and terminate the phase.

S_o = SECTOR DISAGREEMENT FLIP-FLOP

Z_1 and S_1 are compared by S_o

$$1^{S_o} = Z_1 \; S_1' \; N_1 \; K_c' \; D_o' \; T_{41}' + Z_1' \; S_1 \; N_1 \; K_c' \; D_o' \; T_{41}'$$

$$0^{S_o} = T_{41}$$

Where N_1 = Timing Flip-Flop

$$1^{N_1} = G_o' \; K_c' \; \underline{T_{22}}$$

$$0^{N_1} = \underbrace{P_5 \; P_4' \; P_1'}_{T_{28}} K_c' + K_a \underbrace{P_4 \; P_2}_{T_{25} \text{ (for loops)}} K_c'$$

WHEN SECTOR AGREEMENT is reached we will terminate phase 2 and go to phase 3.

S_o' will be true at T_{40}.

$$1^{D_o} = S_o' \; \underline{K_o} \; D_o' \; I_4' \; I_1 \qquad \left| \; \underline{K_o} = K_c' \; T_{41} \right.$$

Phase 3 - is entered by $1^{D_o} = D_o' \; \underline{K_o} \; S_o' \; I_4' \; I_1$

During Phase 3 we will gate one word from Main Memory or the loop into the B-register.

$$1^{B_{41}} = M_r \; D_o \; K_a' \; S_o' + L_1 \; \underline{D_{10}} \; S' + V_1 \; \underline{D_{10}} \; S \qquad \left| \; \underline{D_{10}} = D_o \; K_a \right.$$

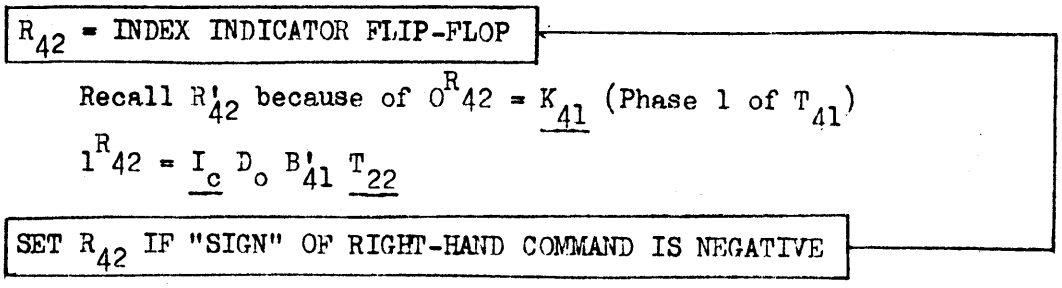
$$0^{B_{41}} = M_r' \; D_o \; K_a' \; S_o' + L_1' \; \underline{D_{10}} \; S' + V_1' \; \underline{D_{10}} \; S$$

INDEXING INVESTIGATION: -

When the COMMAND PAIR is being gated out of memory and into the B-register we must detect INDEXING notation.

This is done by sampling the Right-Hand Commands "SIGN" bit position.

This bit is available to us, out of B_{41}, at $\underline{T_{22}}$.



Phase 3 is terminated by: $0^{D}o = \underline{I_c}\ D_o\ T_{41}$

$1^{K}c = \underline{I_c}\ D_o\ \underline{K_o}$

$1^{I}2 = \underline{I_c}\ D_o\ \underline{K_o}$

Transition is to OPERAND MODE - Phase 1.

OPERAND MODE

Transition from the COMMAND MODE - Phase 3 is ALWAYS into the OPERAND MODE - Phase 1.

This is accomplished by: $0^{D}_{o} = \underline{I_c}\ D_o\ T_{41}$

$$1^{K}_{c} = \underline{I_c}\ D_o\ \underline{K_o}$$

$$1^{I}_{2} = \underline{I_c}\ D_o\ \underline{K_o}$$

In the OPERAND MODE the commands, as designated by their OPERATION CODES, are separated into three distinct types, they may be catagorized as:

TYPE #1 = OPERATION CODE = * $D'_5\ D_4$

TYPE #2 = OPERATION CODE = * $D'_5\ D'_4$

TYPE #3 = OPERATION CODE = * D_5

* Where the confiruration of the Operation Code will be placed in the D-register ($D_6 \longleftrightarrow D_1$) during the first word-time of the OPERAND MODE.

In addition to the normal computer functions that are carried out during the OPERAND MODE, the "TYPE" of command will designate HOW LONG THE OPERAND MODE WILL BE MAINTAINED; THE EXTENT OF OPERATIONS DURING THE MODE; and IF WE WILL PERMIT INDEXING.

For ALL COMMAND TYPES; Phase 1, which is maintained for ONE WORD TIME, WILL BE THE SAME.

Phase 1.

During Phase 1 we will

1. Gate C(B) into the Z-register. (B-register contains the Command-Pair)
2. If "INDEXING" was detected the INDEX-register will be SUBTRACTED from the corresponding B-register bit positions as the C(B) is being gated into the Z-register.
3. Gate the UNINDEXED CHANNEL Address and the OPERATION CODE into the "C"- and "D" registers respectively.
4. Detect Main Memory or Loop address of the "OPERAND"
5. UNINDEXED Sector Number shifts into the appropriate character counter.

Copy C(B) into Z-register - INDEX or NO-INDEX.

$$1^{Z}_{41} = B_1\ \underline{I_n}\ K_c\ G'_o\ R'_{42} = \text{Copy Logic, No Index}$$

$$+ B_1\ G_1\ R_{43}\ \underline{I_x} + B_1\ G'_1\ R'_{43}\ \underline{I_x}$$

$$= \text{C(B) minus C(Index)}$$

$$+ B'_1\ G_1\ R'_{43}\ \underline{I_x} + B'_1\ G'_1\ R_{43}\ \underline{I_x}$$

$$0^{Z}_{41} = B'_1\ \underline{I_n}\ K_c\ G'_o\ R'_{42} = \text{Copy Logic, No Index}$$

$$+B_1\ G_1\ R'_{43}\ \underline{I_x} + B_1\ G'_1\ R_{43}\ \underline{I_x}$$

$$+B'_1\ G_1\ R_{43}\ \underline{I_x} + B'_1\ G'_1\ R'_{43}\ \underline{I_x}$$

$\boxed{(\text{Where } \underline{I_x} = I_2\ I_1\ N_5\ R_{42})}$ $\qquad R_{42} = \text{TRUE} = \text{INDEX}$

$$I^{N}_5 = T_1\ \underline{I_n}\ D'_o + G'_1\ \underline{T_{22}}\ \underline{I_n}\ D'_o \qquad R_{42} = \text{PRIME} = \text{NO-INDEX}$$

$$0^{N}_5 = \underline{T_{20}}\ I'_4 + T_{41}$$

G'_o = True because of $0^{G_o} = \underline{I_c}\ D'_o$ = Command Halt Phase 1

we can now "set" GO IF - $1^{G_o} = G_1\ \underline{T_{22}}\ \underline{I_n}\ K_c$

A "one" bit at T_{22}, from G_1, tells us that our $\frac{1}{2}$ word-bit of the LOCATION COUNTER indicates a RIGHT HAND COMMAND.

G_o is PRIMED WHEN THE INDEX REGISTER is reading out of G_1

R_{42} = True = INDEX UNTIL T_{14} - then

$$0^{R_{42}} = I_{3a}\ K_c\ \underline{T_{14}}$$

and recirculate via the copy logic for Z_{41}

R_{43} = Carry Flip-Flop (for INDEXING)

$$1^{R_{43}} = I'_4\ I_1\ T_1 + B_1\ G_1\ \underline{I_x}$$

$$0^{R_{43}} = B'_1\ G'_1\ \underline{I_x}$$

* Transfer Op Code & Channel info into the D & C FF.s

$$1^{D_6} = B_1\ G'_o\ \underline{I_n}\ N_5\ K_c$$

$$0^{D_6} = B'_1\ G'_o\ \underline{I_n}\ N_5\ K_c$$

Right Hand Op & Chan. Code

(initial read-in; left or right may still be executed first)

The condition of G_1 at T_{22} is important to determine WHICH HALF OF THE COMMAND PAIR WILL BE WORKED ON.

$I_f\ N_5$ comes true again : -

$$1^{N_5} = G'_1\ \underline{T_{22}}\ \underline{I_n}\ D'_o$$

it means the $\frac{1}{2}$ word bit = 0

WORK ON LEFT HAND COMMAND.

$\underline{T_{22}}\ G'_1$ = 0 = Left Hand Command

$\underline{T_{22}}\ G_1$ = 1 = Right Hand Command

$$1^{D}6 = Z_1\ \underline{I_n}\ K'_c\ D'_o\ N_5$$

$$0^{D}6 = Z'_1\ \underline{I_n}\ K'_c\ D'_o\ N_5$$

* Copy indexed channel into D, indexed sector into G (char. ctr.) phase ? of $\underline{I_n}$

Explanation:

1. C (B) is gated into Z-register.
2. C (I) is subtracted from C (B)
3. $G'_o\ R'_{42}$ = Normal Copy Logic
4. $G_1\ \underline{T_{22}}\ \underline{I_n}\ K_c$ = Location Counter says DO Right Hand Com.

   $$1^{G}o = G_1\ \underline{T_{22}}\ \underline{I_n}\ K_c$$

5. Stop copy of C (B) by the Z-register and Recirculate =

   $$1^{Z}41 = Z_1\ G_o$$

   $$0^{Z}41 = Z'_1\ G_o$$

6. The D & C flip-flops were already loaded -

   $$1^{D}6 = B_1\ G'_o\ \underline{I_n}\ N_5\ K_c$$

   $$0^{D}6 = B'_1\ G'_o\ \underline{I_n}\ N_5\ K_c$$

   = Do not reload

But if the location counter says LEFT HAND COMMAND

$$1^{N}5 = G'_1\ \underline{T_{22}}\ \underline{I_n}\ D'_o$$

and G_o will remain Primed.

Load the D & C F.F. again with the Left-Half information from the B-register =

$$1^{D}6 = B_1\ G'_o\ I_n\ N_5\ K_c \text{ etc.}$$

and continue Z_{41} copying B_1

This will not affect the Indexing of the Right Hand Channel and Sector numbers because of R_{42}

$$0^{R}42 = I'_{30}\ K_c\ T_{14}$$ which turns $\underline{I_x}$ OFF

$$\boxed{\underline{I_x} = I_2\ I_1\ N_5\ R_{42}}$$

After indexing the Z_{41} copy B_1 is again true.

It is only possible to load the LEFT HAND CHARACTER Counter during this phase.

The Character counter is filled when the SECTOR NUMBER READS OUT OF C_1

$$1^{G}41 = C'_1\ \underline{I_n}\ K_g\ D'_o + G_1\ K'_g\ T'_{41}$$

$$0^{G}41 = C_1\ \underline{I_n}\ K_g\ D'_o + G'_1\ K'_g\ T'_{41}$$

Where K_g must be true

$$1^{K}g = G_o\ I'_4\ \underline{T_{14}}\ \overbrace{P'_6\ P_1}^{T_{14}} + G'_o\ I'_4\ \underline{T_{14}}\ \overbrace{P_6\ P_1}^{T_{34}}$$

This logic can never come true during Phase 1

$$0^{K}g = \underline{T_{20}}$$

Detection of a MAIN MEMORY or LOOP ADDRESS is accomplished in the same manner as in the COMMAND MODE = Phase 1.

K_a = LOOP $\qquad$ K'_a = MAIN MEMORY

S = V-Loop; S' = L-loop

$$1^{K}a = I_1\ K_c\ T_1 \qquad 0^{K}a = Z'_{41}\ I_1\ N_1\ K_c$$

Where: $1^{N}1 = P'_6\ P'_5\ P'_3\ P_2\ P_1\ K_c\ D'_o\ I_1 + G'_o\ P_6\ P'_5\ P'_3\ P_2\ P_1\ K_c\ D'_o$

$$0^{N}1 = K_c\ D'_o\ P_5\ P_4\ P_2 = T_{15} \text{ or } T_{35}$$

$$1^{S} = C_6\ N_5\ I_1\ \underline{T_{13}} \qquad 0^{S} = C'_6\ N_5\ I_1\ \underline{T_{13}}$$

AT THE CONCLUSION OF THE WORD-TIME ALL TYPE #1 COMMAND WILL TERMINATE THE OPERAND MODE.

Transition is ALWAYS to the Direction Mode --- Phase 2.

Termination of the OPERAND MODE is via $O^{I}1 = \underline{I_n}\ D'_5\ D_4\ T_{41}$

ALL COMMANDS WITH $D'_5\ D_4$ Operation - Codes are Type #1. They are:

OPERATION CODE	MNEMONIC CODE	D-REGISTER CONFIGURATION						
10	TIX	D'_6	D'_5	D_4	D'_3	D'_2	D'_1	———
11	TNZ	D'_6	D'_5	D_4	D'_3	D'_2	D_1	———
12	SAP	D'_6	D'_5	D_4	D'_3	D_2	D'_1	———
12.4	SAN	D'_6	D'_5	D_4	D'_3	D_2	D'_1	C_6 C'_5
13	CSA	D'_6	D'_5	D_4	D'_3	D_2	D_1	———
15	TLB	D'_6	D'_5	D_4	D_3	D'_2	D_1	———
16	CAZ	D'_6	D'_5	D_4	D_3	D_2	D_1	———
16.4	CMZ	D'_6	D'_5	D_4	D_3	D_2	D'_1	C_6 C'_5
17	CMP	D'_6	D'_5	D_4	D_3	D_2	D_1	———
17.4	RND	D'_6	D'_5	D_4	D_3	D_2	D_1	C_6 C'_5
50	TZE	D_6	D'_5	D_4	D'_3	D'_2	D'_1	———
51	TRA	D_6	D'_5	D_4	D'_3	D'_2	D_1	———
52	TOV	D_6	D'_5	D_4	D'_3	D_2	D'_1	———
53	TMI	D_6	D'_5	D_4	D'_3	D_2	D_1	———
54	NOP	D_6	D'_5	D_4	D_3	D'_2	D'_1	———
55	TPL	D_6	D'_5	D_4	D_3	D'_2	D_1	———
56	XAR	D_6	D'_5	D_4	D_3	D_2	D'_1	———
57	CAR	D_6	D'_5	D_4	D_3	D_2	D_1	———

Transition to the DIRECTION MODE - Phase 2 is via $O^{I}1 = \underline{I_n}\ D'_5\ D_4\ T_{41}$ $O^{K}c = I'_4\ I_1\ \underline{K_{41}}$

Phase 2 FIRST WORD-TIME

During the FIRST WORD-TIME of Phase 2 the CHANNEL register ($C_6 \longrightarrow C_1$), the OPERATION CODE- register ($D_6 \longrightarrow D_1$), and the CHARACTER Counter are reloaded.

It is during this word time that the INDEXED Channel number is gated into the C-register, and the INDEXED Sector number is gated into the appropriate CHARACTER Counter, and the Operation Code Register is reloaded ($D_6 \longrightarrow D_1$)

$$1^{D}6 = Z_1 \underline{I_n} K'_c D'_o N_5$$

$$0^{D}6 = Z'_1 \underline{I_n} K'_c D'_o N_5$$

Note: - The actual command that will be executed (LEFT OR RIGHT) was determined during Phase 1; and G_o Flip-Flop was set accordingly. This G_o logic will be felt by N_5, causing proper gating into D_6

$$1^{N}5 = T_1 \underline{I_n} D'_o + G'_1 \underline{T_{22}} \underline{I_n} D'_o$$

$$0^{N}5 = \underline{T_{20}} I'_4 + T_{41}$$

IT IS OF NO CONSEQUENCE that the COMPLEMENT OF "B-register" was copied into the CHARACTER COUNTER during Phase 1. (This will happen for LEFT HAND COMMANDS).

The CHARACTER COUNTER IS LOADED VIA:-

$$1^{G}41 = C'_1 \underline{I_n} K_g D'_o$$

$$0^{G}41 = C_1 \underline{I_n} K_g D'_o$$

AT THE CONCLUSION OF THE FIRST WORD TIME OF PHASE 2, ALL TYPE #2 COMMANDS WILL TERMINATE THE OPERAND MODE.

Transition is always to the DIRECTION MODE - Phase 2, by

$$0^{I}1 = \underline{I_n}\ \underline{K_o}\ D'_5$$

In order to enter Phase 2 we must retain $K'_c\ D'_o$

1. K'_c = There is no possible logic, at this time, that would turn K_c "ON".
2. D'_o = Possibility exists for $1^{D}o = S'_o\ \underline{K_o}\ I'_4\ I_1\ D'_o$

But for $D'_5\ D'_4$ (Type #2) commands we will

$1^{S}o = \underline{I_n}\ D'_5\ D'_4\ T_1\ K'_c$ which assures us

So at T_{41} $\qquad$ ($0^{S}o = T_{41}$ - ONLY)

ALL COMMANDS WITH $D'_5\ D'_4$ OPERATION CODES are TYPE #2. They are:

For TYPE #2 and Type #3 COMMANDS transition is to Phase 2 of the OPERAND MODE.

$$0^{Kc} = I'_4\ I_1\ \underline{K_{41}} \qquad 6.$$

OPERATION CODE	MNEMONIC CODE		D-REGISTER CONFIGURATION						
00	ICH	D'_6	D'_5	D'_4	D'_3	D'_2	D'_1		
01	OCH	D'_6	D'_5	D'_4	D'_3	D'_2	D_1		
02	ALS	D'_6	D'_5	D'_4	D'_3	D_2	D'_1		
02.2	ASV	D'_6	D'_5	D'_4	D'_3	D_2	D'_1	C'_6	C_5
02.4	ASC	D'_6	D'_5	D'_4	D'_3	D_2	D'_1	C_6	C'_5
03	ARS	D'_6	D'_5	D'_4	D'_3	D_2	D_1		
05	STR	D'_6	D'_5	D'_4	D_3	D'_2	D_1		
40	CTL	D_6	D'_5	D'_4	D'_3	D'_2	D'_1		
41	CTV	D_6	D'_5	D'_4	D'_3	D'_2	D_1		
42	LLS	D_6	D'_5	D'_4	D'_3	D_2	D'_1		
42.2	LSV	D_6	D'_5	D'_4	D'_3	D_2	D'_1	C'_6	C'_5
42.4	LSC	D_6	D'_5	D'_4	D'_3	D_2	D'_1	C_6	C'_5
43	LRS	D_6	D'_5	D'_4	D'_3	D_2	D_1		
44	CFL	D_6	D'_5	D'_4	D_3	D'_2	D'_1		
45	STO	D_6	D'_5	D'_4	D_3	D'_2	D_1		

Transition to the DIRECTION MODE - Phase 2 is via

$$0^{I_1} = \underline{I_n}\ \underline{K_o}\ D'_5$$

For ALL type #3 commands Phase 2 will be maintained until SECTOR AGREEMENT between Z_1 and S_1 is reached.

Phase 2 AFTER THE FIRST WORD-TIME

During Phase 2, after the FIRST WORD-TIME, we SEARCH FOR SECTOR AGREEMENT.

S_o = SECTOR DISAGREEMENT FLIP-FLOP

$$0^{S}_{o} = T_{41}$$

$$1^{S}_{o} = K'_c\ D'_o\ Z_1\ S'_1\ N_1\ T'_{41} + K'_c\ D'_o\ Z'_1\ S'_1\ N_1\ T'_{41}$$

And when SECTOR AGREEMENT is indicated

$$1^{D}_{o} = S'_o\ \underline{K_o}\ I'_4\ I_1\ D'_o$$

And transit from Phase 2 into Phase 3.

Phase 3 - MAINTAINED FOR ONE WORD-TIME

During Phase 3 we will gate one word from Memory into the B-register

$$1^{B}_{41} = M_r\ D_o\ K'_a\ S'_o + L_1\ \underline{D_{10}}\ S' + V_1\ \underline{D_{10}}\ S$$

$$\boxed{\underline{D_{10}} = D_o\ K_a}$$

$$0^{B}_{41} = M'_r\ D_o\ K'_a\ S'_o + L'_1\ \underline{D_{10}}\ S' + V'_1\ \underline{D_{10}}\ S$$

ALL TYPE #3 COMMANDS WILL TERMINATE THE OPERAND MODE at the conclusion of ONE WORD-TIME OF Phase 3 ------ via

$$0^{D}_{o} = I_1\ D_o\ T_{41} \qquad 0^{I}_{1} = \underline{N_{11}} \qquad \boxed{\underline{N_{11}} = I_2\ I_1\ D_o\ T_{41}}$$

Transition is to the DIRECTION MODE - Phase 2

TYPE #3 COMMANDS ARE:

OPERATION CODE	MNEMONIC CODE	D-REGISTER CONFIGURATION					
25.1	STI → D'_6	D_5	D'_4	D_3	D_5	D_1	+COMMAND INDICATOR
31.1	LDI → D'_6	D_5	D_4	D'_3	D'_2	D_1	+COMMAND INDICATOR
35	CMG → D'_6	D_5	D_4	D_3	D'_2	D_1	+COMMAND INDICATOR
35.1	CME → D'_6	D_5	D_4	D_3	D'_2	D_1	+COMMAND INDICATOR
36	CLS → D'_6	D_5	D_4	D_3	D_2	D'_1	
37	CLA → D'_6	D_5	D_4	D_3	D_2	D_1	
63	MPY → D_6	D_5	D'_4	D'_3	D_2	D_1	
65	STA → D_6	D_5	D'_4	D_3	D'_2	D_1	
66	DIV → D_6	D_5	D'_4	D_3	D_2	D'_1	
70	EXT → D_6	D_5	D_4	D'_3	D'_2	D'_1	
71	HTR → D_6	D_5	D_4	D'_3	D'_2	D_1	
72	SUB → D_6	D_5	D_4	D'_3	D_2	D'_1	
73	ADD → D_6	D_5	D_4	D'_3	D_2	D_1	
76	RCS → D_6	D_5	D_4	D_3	D_2	D'_1	
77	RCA → D_6	D_5	D_4	D_3	D_2	D_1	

ALL COMMANDS WITH - D_5 - are TYPE #3 COMMANDS.

Direction Mode

During direction mode several various functions may occur. These functions for the most part are controlled by the command being operated upon.

The only description covered here will be for incrementing the Location Counter which will occur each command operation.

The location counter will be incremented during phase one or phase two of Direction Mode. This will be accomplished by adding a binary one to the "G" register at T_{22} time using a 1 - input Adder.

$$1^{K}_{g} = \underline{NR_3}\ I_2\ \underline{T_{21}}\ F'_a \qquad (\underline{NR_3} = I_4\ I'_1)$$

$$(\underline{T_{21}} = T'_{41}\ P_5 P'_4 P'_3 P'_2)$$

$$0^{K}_{g} = G'_1 K_g I'_1 D_o'$$

$$1^{G}_{41} = G'_1 K_g I'_1 D'_o \qquad 0^{G}_{41} = G_1 K_g I'_1 D'_o$$

$$G_1 K'_g T'_{41} \qquad G'_1 K'_g T'_{41}$$

The logic for K_g states that during Direction $\underline{NR_3} I_2\ F'_a$ are true; "Set" K_g at T_{21} time.

This will inhibit the normal recirculation of the "G" register and add a one at T_{22} time. K_g will then act as a carry flip-flop until it is "reset" to zero by reading a zero at G_1.

After K_g is "zero set" the normal recirculation logic is again used.

ALL other operations, occuring during the DIRECTION MODE, will be covered as each command's execution is explained.

CLEAR ADD _____#37 _____CLA

CLEAR SUBTRACT____#36_____CLS

CLEAR ADD_____#77 _____RCA

CLEAR SUBTRACT ____#76_____RCS

Both the CLA (#37) and the RCA (#77) commands cause the C(W) to replace the C(A), including the "sign."

In CLA, however, the C(A) is lost; where as in the RCA, the C(A) replaces the C(R).

Both the CLS (#36) and the RCS (#76) commands cause the C(W) to replace the C(A), with a "SIGN" reversal.

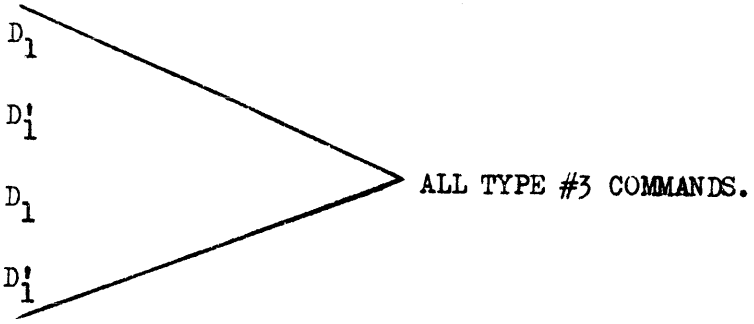
In CLS, however, the C(A) is lost; whereas in the RCS, the C(A) replaces the C(R).

CLA - #37 - $D_6' D_5 D_4 D_3 D_2 D_1$

CLS - #36 - $D_6' D_5 D_4 D_3 D_2 D_1'$

RCA - #77 - $D_6 D_5 D_4 D_3 D_2 D_1$

RCS - #76 - $D_6 D_5 D_4 D_3 D_2 D_1'$



Review OPERAND MODE - INDEXING Permitted.

Phase 1.

(1) Gate C(B) into the Z-register (INDEXING Permitted)

(2) Gate UNINDEXED Op-Code and Chan-Code into the D and C-registers.

(3) Analyze for Loop or Main Memory address

Phase 2.

(1) Gate INDEXED and Chan-Code into the C-registers, reload D-register.

(2) Load CHARACTER counter with COMPLEMENTED SECTOR NUMBER.

(3) Search for SECTOR AGREEMENT.

Phase 2- Gate C(W) into the B-register

Transition is to the DIRECTION MODE - Phase 2 via:

$$0^{D}o = I_1 \, D_o \, T_{41}$$

$$0^{I}1 = \underline{N_{11}}$$

$$\underline{N_{11}} = I_2 \, I_1 \, D_o \, T_{41}$$

Phase 2 is LEFT AFTER ONE (1) BIT-TIME

$$1^{K}c = \underline{NR_3} \, D_4 \, T_1 \, F'_a$$

$$\underline{NR_3} = I'_4 \, I'_1$$

and we enter Phase 1, at $\underline{T_2}$

During Phase 1 - the A-register is shifted RIGHT

$$1^{A}41 = B_1 \, \underline{M_{oo}} \, D'_2$$

$$0^{A}41 = B'_1 \, \underline{M_{oo}} \, D_2$$

$$\underline{M_{oo}} = I_{3a} \, D_5 \, D_4 \, D_3 \, T'_{41}$$

Where: -

$$1^{I}3a = \underline{N_{11}} \, D_2$$

$$0^{I}3a = I_{3a} \, \underline{K_{41}}$$

And for the RCA and RCS commands the A-register is gated into R_{41}

$$1^{R}41 = A_1 \, I_{3a} \, I_2 \, D_6 \, D_4 \, D_3 \quad *$$

$$0^{R}41 = A'_1 \, I_{3a} \, I_2 \, D_6 \, D_4 \, D_3 \quad *$$

$$1^{A}1 = A_2 \, D_6 \, 0_2$$

$$0^{A}1 = A'_2 \, D_6 \, I_2$$

* NOTE: _ D_6 is primed for CLA AND CLS, thus restricting these equations to RCA AND RCS.

and a ZERO SYNC BIT is assured the R_1 position by:

$$0^{R}1 = I_{3a} \, D_5 \, D_6 \, \underline{T_{40}}$$

The correct "SIGN" bit is assured the A-register by B_{40} and C_2.

$$1^{B}40 = B_{41}\ \underline{M_{oo}}\ D_1 = D_1 \text{ true} = \text{CLA or RCA - No SIGN REVERSAL}$$

$$+ B'_{41}\ \underline{M_{oo}}\ D'_1 = D'_1 \text{ true} = \text{CLS or RCS - SIGN REVERSAL}$$

$$0^{B}40 = B'_{41}\ \underline{M_{oo}}\ D_1 = \text{No SIGN REVERSAL}$$

$$+ B_{41}\ \underline{M_{oo}}\ D'_1 = \text{SIGN REVERSAL}$$

B_{40} copies the "SIGN" from B_{41} at T_1. (At T_2 the correct "SIGN" is copied by C_2)

$$1^{C}2 = \underline{U_1}\ D_2\ T_1 = \text{Preset and ASSUME POSITIVE SIGN}$$

$$0^{C}2 = B'_{40}\ \underline{M_{oo}}\ D_5\ T_2 = \text{Reset if SIGN IS NEGATIVE}$$

C_2 then deposits the correct "SIGN" in A_{41} at T_{41}

$$1^{A}41 = \underline{K_{41}}\ C_2\ I_{3a}$$

$$0^{A}41 = \underline{K_{41}}\ C'_2\ I_{3a}$$

At the conclusion of the word time all assignments have been handled.

Termination occurs with

$$1^{I}1 = I'_1\ \underline{K_{41}}$$

Transition is to the COMMAND or OPERAND MODE.

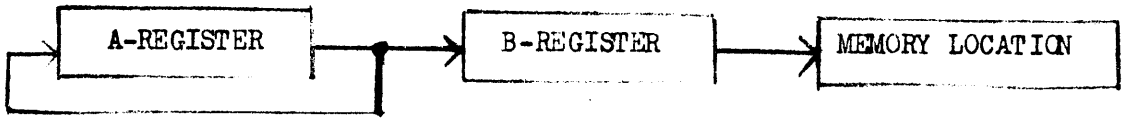
OPERATION CODES

STORE #45 -	STO:	$D_6\ D'_5\ D'_4\ D_3\ D'_2\ D_1$
STORE - R #05 -	STR:	$D'_6\ D'_5\ D'_4\ D_3\ D'_2\ D_1$
STORE ADDRESS #65 -	STA:	$D_6\ D_5\ D'_4\ D_3\ D'_2\ D_1$
STORE INDEX #25.1 -	STI:	$D'_6\ D_5\ D'_4\ D_3\ D'_2\ D_1$

The similarities of the four "store" commands facilitate their presentation as a group.

Let us first discuss what we will expect from each command

(1) STO - #45: This command will cause the contents of the A-register replace the contents of that memory location specified by the Channel and Sector portion of the command.

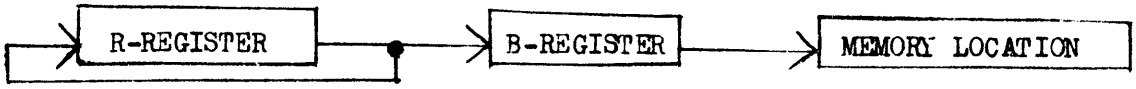


The C(A) remains the same

The C(R) remains the same

Indexing IS Permitted

(2) STR - #05: The contents of the "R" register replace the contents of that memory location specified by the Channel and Sector portion of the command.

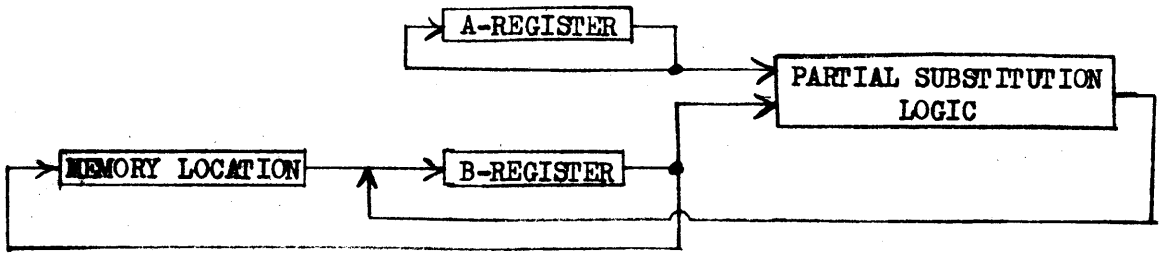


The C(A) remains the same

The C(R) remains the same

Indexing IS Permitted

(3) STA - #65: The Address (Channel and Sector) portion of the Memory Location bit positions 2 through 14 or 22 through 34, is replaced by the corresponding bit position configuration in the "A" register. The "B" register is used to accomplish this partial substitution.

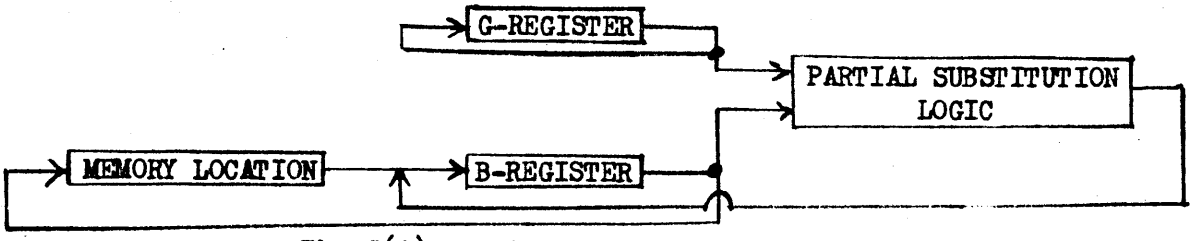


The C(A) remains the same

The C(R) remains the same

INDEXING IS PERMITTED

(4) STI - #25.1: The contents of the INDEX REGISTER replaces the address portion (Channel and Sector) of the Memory Location specified by the command.



The C(A) remains the same

The C(R) remains the same

The C(G) remains the same

INDEXING IS PERMITTED

It can be seen from the Command pictorials that the similiarity between the commands exists in the following areas:

(a) All require "writing" into either the Loops or Main Memory.

(b) All "writing" will be accomplished from the "B" register.

(c) All necessary data must be placed in the "B" register prior to writing.

We may now theorize as to method of command presentation

(a) STO - #45: - Gate A-register into B-register and PREPARE TO WRITE

(b) STR - #05: - Gate R-register into B-register and PREPARE TO WRITE

(c) STA - #65: - Gate Memory Location into B-register. Replace the correct address portion of the B-register (using partial -substitution - logic) with those bit positions of the A-register, and return this modified configuration to the B-register. PREPARE TO WRITE

(d) STI - #25.1: - Gate Memory Location into B-register. Replace the RIGHT HAND address portion of the B-register (using partial-substitution-logic) with those bit positions (INDEX REGISTER) of the G-register and return this modified configuration to the B-register. PREPARE TO WRITE

STO - #45 Is a type #2 command. This means that it will terminate the OPERAND MODE ($\underline{I_n}$) after passing through Mode $\emptyset$ 1, and completed ONE WORD-TIME in Mode $\emptyset$ 2 of $\underline{I_n}$. Transition will be into the DIRECTION MODE, $\emptyset$ 2, where it will remain for one word-time.

(1) During $\emptyset$ 1 of OPERAND ($\underline{I_n}$) the address is sensed as Main Memory or Loop by monitoring $G_{27} \longrightarrow G_{34}$ at Z_{41}. It must be recalled that the C(B) had been placed in the B-register during the Command Mode $\emptyset$ 3; and is now being copied into the Z-register.

K_a = Loop or Main Memory sensing Flip-Flop

S = L-Loop or V-Loop sensing Flip-flop

$$1^{K}a = I_1 K_c T_1 + I_1 K_c G_o' P_5' P_3' P_2 \qquad 1^{N}1 = P_6' P_5' P_3' \overline{P_1 K_c D_o' T_7} + P_6 P_5' P_3' \overline{P_2 P_1 K_c D_o' G_o' T_{27}}$$

$$0^{K}a = Z_{41}' I_1 N_1 K_c \qquad 0^{N}1 = P_5 P_4 P_2 K_c D_o' (T_{15} \text{ or } T_{35})$$

Since a Loop address must be at least 7760, the eight (8) most significant bits of the address MUST BE "ONES" for qualification.

K_a = Loop Address $\qquad$ K_a' = MAIN MEMORY

Whether a Loop Address is indicated or not, S-Flip Flop will be set to indicate which loop is to be selected.

$$1^{S} = C_6 N_5 I_1 \underline{T_{13}} \qquad 0^{S} = C_6' N_5 N_1 \underline{T_{13}} \qquad 1^{N}5 = T_1 \underline{I_n} D_o' + G_1' T_{22} \underline{I_n} T_{41}$$

$$0^{N}5 = \underline{T_{20}} I_4' + T_{41}$$

---

* C_6 will contain the determining bit attitude at T_{13} time.
Bit position six (6) the third (3rd) most significant bit of the Sector Address will have been shifted into D_6 and down through the D-register; appearing in C_6 at $\underline{T_{13}}$.

S = V-Loop $\qquad$ S' = L-Loop

* - Transfer of the Operation Code and the Channel Address into the "D" and "C" registers, respectively, is accomplished during $\emptyset$ 1.

(2) The forgoing procedure, as outlined in "(1)" will be the same for STO #45; STR #05; STA #65 and STI #25.1.

Transition for the four (4) commands in this group will be into Mode $\emptyset$ 2 by: -

$$0^{K_c} = I_4' \; I_1 \; \underline{K_{41}} \qquad \left[K_{41} = K_c \; T_{41} \right.$$

It is now evident that we will remain in Mode $\emptyset$ 1 for ONE WORD-TIME maximum.

(3) Entering Mode $\emptyset$ 2 we have the following configuration in the Mode Flip-Flops:

$$I_4' \; I_2 \; I_1 \; K_c' \; D_o'$$

The commands will be split into two groups, because of their Operation Codes, at the end of the first (1st) word time in $\emptyset$ 2.

During $\emptyset$ 2 we will: -

(a) The Indexed Channel number is gated through the D-register and into the C-register.

(b) The Indexed Character Count (the $1^{\underline{s}}$ complement sector count) is gated through the "D" and "C" registers; and into the Character Count portion of the G-register.

(c) The Operation Code is RE-ENTERED into the D-register.

$$1^{D_6} = Z_1 \; \underline{I_n} \; K_c' \; D_o' \; N_5 \qquad 0^{D_6} = Z_1' \; \underline{I_n} \; K_c' \; D_o' \; N_5$$

$$1^{N_5} = T_1 \; \underline{I_n} \; D_o' + G_1' \; \underline{T_{22}} \; \underline{I_n} \; D_o'$$

$$0^{N_5} = I_{20} \; T_4' + T_{41}$$

(a) Indexed Character Count
(b) Op. Code
(c) Sector No. Gated Into D_6

$$1^{G_{41}} = C_1' \; \underline{I_n} \; K_g \; D_o' \qquad 0^{G_{41}} = C_1 \; \underline{I_n} \; K_g \; D_o'$$

$$1^{K_g} = G_o \; I_4' \; P_6' \; P_1 \; \underline{T_{14}} + G_o \; I_4' \; P_6 \; P_1 \; \underline{T_{14}}$$

$$0^{K_g} = \underline{T_{20}}$$

The "COMPLEMENTED" Sector-Count is gated into the "CHARACTER-COUNTER".

Again note the Operation Code for the Commands

STO - #45 =	D_6	D'_5	D'_4	D_3	D'_2	D_1	$D'_5\ D'_4$ = TYPE # 2 Command
STR - #05 -	D'_6	D'_5	D'_4	D_3	D'_2	D_1	
STA - #65 =	D_6	D_5	D'_4	D_3	D'_2	D_1	D_5 = TYPE #5 Command
STI- #25.1 =	D'_6	D_5	D'_4	D_3	D'_2	D_1	

STO #45 and STR #05 - Type #2 commands will terminate the Operand Mode (I_n) after one word-time of $\emptyset$ 2. Transition will be to the Direction Mode $\emptyset$ 2 for the gating of the necessary data into the "B" register.

$$0^{I}1 = \underline{I_n}\ D'_5\ \underline{K_o} = \text{Transition to Direction Mode } \emptyset\ 2$$

$$\underline{K_o} = K_c\ T_{41}$$

The Mode phase ($\emptyset$) 2 of the Direction Mode is held for one word-time only ---> under all circumstances.

$$1^{B}41 = A_1\ D_6\ D'_2\ I'_1\ R_{42} \qquad \text{(B-register copies A-register for STO \#45)}$$

$$0^{B}41 = A'_1\ D_6\ D'_2\ I'_1\ R_{42}$$

$$1^{R}42 = D'_2\ D_1\ \underline{U_1}\ T_1\ R_{43} \qquad 1^{R}43 = I'_4\ I_1\ T_1$$

$$0^{R}42 = \underline{K_{41}}\ * \qquad 0^{R}43 = - - \frac{N}{A} - -$$

$$\underline{K_{41}} = K_c\ T_{41}\ (* \text{ } K_c \text{ is PRIMED at this time})$$

$$1^{A}41 = A_1\ D_1\ \underline{I_3} \qquad \text{(A-Register)} \qquad \underline{I_3} = I'_4\ I'_1\ I'_{3a}\ D'_o\ **$$

$$0^{A}41 = A'_1\ D'_2\ I'_4\ \underset{**}{FA'} \qquad \text{(Recirculation)}$$

** I_{3a} & FA: are both PRIMED during the entire execution of all four (4) commands

$$1^{R}41 = R_1\ I'_{3a} \text{ (R-Register)}$$

$$0^{R}41 = R'_1\ I'_{3a} \text{ (Recirculation)}$$

$1^{B}41 = R_1\ \underline{NR_1}\ D'_5$ (B-register copies R-register for STR #05)

$0^{B}41 = R'_1\ \underline{NR_1}\ D'_5$

$\underline{NR_1} = R_{42}\ D'_6\ I'_1\ FM'$ (FM will be primed for entire command)

Transition from $\emptyset$ 2 for these Type #2 commands is to the Execute Mode - Phase 2 ($U_i\ D'_o$) by:

$$0^{I}2 = \underline{NR_3}\ \underline{K_o}\ D'_4 \qquad \underline{NR_3} = I'_4\ I'_1$$

STA #65 and STI #25.1 are type #3 commands and WILL NOT TERMINATE the Operand Mode after one word-time of Phase 2.

Type #3 commands require an OPERAND from Memory, and provision for "Searching" is provided when $\emptyset$ 2 is maintained in excess of one word-time. When "Searching" is completed (an indefinite number of word times are required) during $\emptyset$ 2, transition will be made to $\emptyset$ 3.

During Phase #3 the Operand will be gated out of Memory and into the B-register.

(a) $0^{S_o} = T_{41}$ $\qquad S'_o$ = Sectors Agree

$1^{S_o} = Z_1\ S'_1\ K'_c\ D'_o\ N_1\ T'_{41} + Z'_1\ S_1\ K'_c\ D'_o\ N_1\ T'_{41}$ $\qquad S_o$ = Disagreement

S_o = Sector Disagreement Flip-Flop.

(b) As soon as we achieve S'_o at T_{41} Phase #2 is terminated and we enter Phase #3.

$1^{D_o} = D'_o\ \underline{K_o}\ I'_4\ I_1\ D'_o$

Phase #3 is maintained for one word time; during which the contents of Memory is gated into the B-register.

$1^{B}41 = M_r\ D_o\ K'_a\ S'_o + L_1\ \underline{D_{10}}\ S' + V_1\ \underline{D_{10}}\ S$

$0^{B}41 = M'_r\ D_o\ K'_a\ S'_o + L'_1\ \underline{D_{10}}\ S' + V'_1\ \underline{D_{10}}\ S$

TRANSITION is now into the DIRECTION MODE - Phase #2 for one word time.

M_r = Memory Read F.F.

$\underline{D_{10}} = D_o\ K_a$

K_a = Loop Address

K'_a = Main Memory Address

S = V-Loop Selection

S' = L-Loop Selection

$$0^{D}o = I_1\ D_o\ T_{41}$$

$$0^{I}1 = \underline{N_{11}} = I_2\ I_1\ D_o\ T_{41}$$

Additional Note: - During $\underline{I_n}$ the "command indicator" is sampled by R_{43}.

$$1^{R}43 = I'_4\ I_1\ T_1$$

$$0^{R}43 = Z'_1\ G'_o\ \underline{T_{22}}\ \underline{I_n}\ D_o + Z'_1\ G_o\ \underline{T_2}\ \underline{I_n}\ D_o$$

Upon entering the "DIRECTION MODE" $I'_4\ I_2\ I'_1\ D'_o\ K'_c$ the gate U1 is true.

$\underline{U_1} = I'_4\ I_2\ I'_1\ D'_o$ - We then turn on

R_{42} (which will be used as the "Gating" flip-flop) during the appropriate half ($\frac{1}{2}$) word.

$$1^{R}42 = {}^{(1)}\ D'_2\ D_1\ \underline{U_1}\ T_1\ R_{43} + {}^{(2)}\ D_5\ D'_2\ \underline{U_1}\ \underline{T_{21}}\ D'_4\ R'_{43}$$

$$0^{R}42 = D_5\ I'_{3a}\ \underline{T_{14}}\ I'_1$$

$\underline{T_{14}} = T_{14}$ or T_{34}

(1) Turn "ON" R_{42} for RIGHT HAND portion of the word IF the Command Indicator was true (R_{43})

(2) Turn (ON) R_{42} for LEFT HAND portion of the word IF the COMMAND INDICATOR was false (R'_{43})

(3) Reset R_{42} for normal recirculation after "partial substitution is accomplished in STA & STI Commands (where D_5 is true)

During the time that R_{42} is true: -

(1) Normal recirculation of the "B" register stops, and

(2) The correct bit positions from the A-register or the G-register complement are copied by the B-register

"A" is inserted into "B" for the #65-STA (Store Address) Command.

The Complement of G

"G" is inserted into "B" for the #25.1-STI (Store Index) Command.

$$1^{B}41 = A_1\ R_{42}\ D_6\ D'_2\ I'_1 \longrightarrow \boxed{D_6\ D'_2 = \text{True for \#65-STA}}$$

$$+\ G'_1\ \underline{NR_1}\ D_5 \longrightarrow \boxed{\underline{NR_1} = R_{42}\ D_6\ I'_1\ FM' \text{ for \#25.1-STI}}$$

$$+\ B_1\ I'_4\ D'_o\ R'_{42} = \longrightarrow \boxed{\text{Normal Recirculation}}$$

$$0^{B}41 = A'_1\ R_{42}\ D_6\ D'_2\ I'_1$$

$$+\ G_1\ \underline{NR_1}\ D_5$$

$$+\ B'_1\ I'_4\ D'_o\ R'_{42}$$

While this partial substitution is taking place the "A" and "R" registers will be recirculated.

$$1^{R}41 = R_1\ I'_{3a} \qquad 1^{A}41 = A_1\ \underline{I_3}\ D_1$$

$$0^{R}41 = R'_1\ I'_{3a} \qquad 0^{A}41 = A'_1\ D'_2\ I'_4\ FA'$$

The forgoing is accomplished during one word time in the Direction Mode.

Termination of the Direction Mode and transition to the Execute Mode is by: -

$$0^{I}2 = \underline{NR_3}\ \underline{K_o}\ D'_4 \qquad \left[\underline{NR_3} = I'_4\ I'_1\right.$$

THE EXECUTE MODE

The configuration of the EXECUTE MODE is:

$$I'_4\ I'_2\ I'_1\ K'_c\ D'_o$$

$$\left[\text{Execute } \emptyset\ 2 = U_i\ D'_o\right.$$

$$\left.\underline{U_i} = I'_4\ I'_2\ I'_1\right.$$

Note:

It must be understood that when the Execute Mode - Phase 2 is entered, the "B" contains the "configuration" that is to be stored in Memory.

Since Sector Search for WRITING is required ONLY THE FIRST FIVE (5) BITS of the Sector Address must agree while the last bit (or most significant bit) **MUST DISAGREE.**

N_1 = Comparison Gating Flip-Flop

$$1^{N_1} = G_o \underline{T_2} K'_c + G'_o \underline{T_{22}} K'_c$$

$$0^{N_1} = P_5 P'_4 P'_1 K'_c + \underline{W_s} P_2 P_1$$

$P_5 P'_4 P'_1 = T_8$ time

$P_2 P_1 = T_7$ time

N_1 is found to be TRUE from $T_3 \rightarrow T_7$ or $T_{23} \rightarrow T_{27}$

$\underline{W_s} = I'_4 I'_1 D'_4 D_3 D'_2 D'_o$

S_o = Comparison F.F.

$$1^{S_o} = K'_c D'_o N_1 T'_{41} Z_1 S'_1 + K'_c D'_o N_1 T'_{41} Z'_1 S_1$$

$$+ Z_1 S_1 \underline{W_s} \underline{T_8} + Z'_1 S'_1 \underline{W_s} \underline{T_8} + Z_1 S_1 \underline{W_s} T_{28} + Z'_1 S'_1 \underline{W_s} \underline{T_{28}}$$

$$0^{S_o} = T_{41}$$

We should remain conscious of the fact that the COMPARISON FOR SECTOR AGREEMENT is between the INDEXED Z-REGISTER and the SECTOR CHANNEL output.

IMPORTANT NOTE:

The configuration in the "B" register does not recirculate for STO & STR Commands. The "B" register will recirculate for STA, STI, & SLC Commands.

(1) $1^{R_{42}} = D'_2 D_1 \underline{U_1} T_1 R_{43} + D_5 D_2 \underline{U_1} \underline{T_{21}} D'_4 R'_{43}$

(2) There is no recirculation logic, for the "B" register for STO & STR Commands because we cannot reset R_{42}.

(3) The B_{41} copy A_1 or R_1 remains true.

$$1^{B_{41}} = A_1 I'_1 R_{42} D_6 D'_2 + R_1 \underline{NR_1} D'_5$$

$$0^{B_{41}} = A'_1 I'_1 R_{42} D_6 D'_2 + R'_1 \underline{NR_1} D'_5$$

$\underline{NR_1} = I'_1 R_{42} F'_m D'_6$

(4) For STA, STI, & SLC we can "Reset" or "Set" R_{43} when the Operand Mode Phase 3 is entered via $0^{R_{43}} = Z'_1 G'_o \underline{T_{22}} \underline{I_n} D_o + Z'_1 G_o \underline{T_2} \underline{I_n} D_o$

(5) R_{42} is Reset: - $0^{R_{42}} = D_5 I'_{3a} \underline{T_{14}} \underline{T'_1}$ and will not come on again.
(NOTE: - D_5 = true for STA, STI, SLC only)

(6) B recirculates manually via: $1^{B_{41}} = B_1 I'_4 D'_o R'_{42}$

$$0^{B_{41}} = B'_1 I'_4 D'_o R'_{42}$$

Sector Agreement is explicit with $S'_o\ T_{41}$ and we terminate Mode $\emptyset$ 2 and enter Mode $\emptyset$ 3 - $\underline{U_i}\ D_o$

$$1^{D_o} = S'_o\ \underline{U_i}\ D'_4\ D'_2\ D'_o\ T_{41}$$

During Mode $\emptyset$ 3, which will be maintained for one (1) word time, we will WRITE the contents of the B-register into the specified loop or MAIN MEMORY ADDRESS.

M_{w1} and M_{w2} = MAIN MEMORY WRITE FLIP-FLOPS

L_{41} = L-Loop WRITE FLIP-FLOP

V_{41} = V-Loop WRITE FLIP-FLOP

MAIN MEMORY WRITE FLIP-FLOP CONFIGURATIONS: -

M_{w1} M'_{w2} = WRITE A ONE (1)

M'_{w1} M_{w2} = WRITE A ZERO (0)

M_{w1} M_{w2} = NO WRITING.

$$1^{M_{w1}} = B_1\ \underline{M_6}\ \underline{MTM}\ T'_1 + I_1$$

$$0^{M_{w1}} = B'_1\ \underline{M_6}\ \underline{MTM} + \underline{M_6}\ K'_a\ D_3\ T_1 \text{ (sync)}$$

$$1^{M_{w2}} = B'_1\ \underline{M_6}\ \underline{MTM} + T_1 \text{ (sync)}$$

$$0^{M_{w2}} = B_1\ \underline{M_6}\ \underline{MTM}\ T'_1$$

$\underline{M_6} = I'_2\ I'_1\ D_o\ D'_4\ D'_2$

$\underline{MTM} = K'_a\ D_3\ D_1$

Note that $\underline{MTM}$ is included in all the writing minterms. It, itself, includes K'_a in its basic minterm configuration.

Writing in the LOOPS occurs when K_a = True,

and S' = L-Loop; S = V-Loop.

$$1^{L_{41}} = B_1\ \underline{M_6}\ \underline{LTM}\ S'$$

$$0^{L_{41}} = B_1\ \underline{M_6}\ \underline{LTM}\ S'$$

$$1^{V_{41}} = B_1\ \underline{M_6}\ \underline{LTM}\ S$$

$$0^{V_{41}} = B'_1\ \underline{M_6}\ \underline{LTM}\ S$$

$\underline{LTM} = K_a\ D_3\ D_1$

As previously stated- Execute Mode -$\emptyset$ 3 ($U_i\ D_o$) is terminated after one (1) word time

$$1^{K_c} = \underline{U_i}\ D_3\ D_1\ D_o\ \underline{T_{40}}$$

$$1^{I_1} = \underline{K_{41}}\ I'_1$$

$$0^{D_o} = I_1\ D_o\ T_{41} + \underline{K_{41}}\ G'_1\ N'_1$$

INSERT INTO THE FOUR STORE COMMANDS

SLC - #25.0

SLC - Store Location Counter

The SLC command acts in the same manner as the STI (#25.1).

(1) Because the command indicator bit will be a zero (0) we will leave R_{43} primed.

(2) R_{43} primed will cause R_{42} true for the LEFT-HAND portion of the word.

(3) This will cause the partial substitution to occur for the Left-hand address portion of the C(W).

(4) The substitution will be the complement of the A-register.

CAZ - #16

CMZ - #16.4

The CAZ (#16) command will cause the C(A) to be forced to ZEROS (0's) and a POSITIVE SIGN to be placed in the SIGN BIT position.

The CMZ (#16.4) command will cause the C(A) to be forced to ZEROS (0's) and a NEGATIVE SIGN to be placed in the SIGN BIT position.

$$\text{CAZ - \#16} - D_6' \; D_5' \; D_4 \; D_3 \; D_2 \; D_1' \; C_6'$$

$$\text{CMZ - \#16.4} - D_6' \; D_5' \; D_4 \; D_3 \; D_2 \; D_1' \; C_6$$

Both CAZ and CMZ are TYPE # 1 commands.

Review OPERAND MODE FOR TYPE #1 commands.

Phase 1: -

(1) C(B), which is the command-pair is gated into the Z-register.

(2) The UNINDEXED Channel address, and Operation Code are gated into the C-and D-registers respectively.

(3) Analyze address.

(4) Terminate the Operand Mode after ONE WORD-TIME. Transition is to the Direction Mode - Phase 2.

Direction Mode - Phase 2.

Entered via:

$$0^{K}c = I_4' \; I_1 \; \underline{K_{41}}$$

$$0^{I}1 = \underline{I_n} \; D_5' \; D_4 \; T_{41}$$

Phase 2 is maintained for ONE BIT-TIME

$$1^{K}c = \underline{NR_3} \; D_4 \; T_1 \; F_a' \qquad \left[\underline{NR_3} = I_4' \; I_1'\right.$$

Transition is into Phase 1

During Phase 1 the A-register is "reset" to ZERO (0).

$$0^{A}41 = I_4' \, T_1 \, F_a'$$

$$1^{A}41 = \text{No "SET" logic available.}$$

The zero placed in A_{41} shifts down the entire register.

$$0^{A}40 = A_{41}' \, D_3$$

$$0^{A}39 = A_{40}' \, D_4$$

$$0^{A}1 = A_2' \, D_5'$$

The "SIGN" is placed in C_2.

$$1^{C}2 = U_1 \, T_1 \, D_2 \text{ (Cannot be reset for CAZ)}$$

$$0^{C}2 = I_{3a} \, D_6' \, D_5' \, D_4 \, D_1' \, C_6 \text{ (MUST be reset for CMZ)}$$

$$\begin{cases} 1^{I}3a = \underline{U_1} \; \underline{D_{52}} \; T_1 \\ \underline{U_1} = I_4' \, I_2 \, I_1' \, D_o' \\ \underline{D_{52}} = D_5' \, D_2 \end{cases}$$

The CORRECT SIGN is then placed in A_{41} at T_{41} via

$$1^{A}41 = C_2 \, I_{3a} \, \underline{K_{41}} \qquad \text{(For CAZ)}$$

$$0^{A}41 = C_2' \, I_{3a} \, \underline{K_{41}} \qquad \text{(For CMZ)}$$

The R-register was recirculated:

$$1^{R}41 = R_1 \, D_6'$$

$$0^{R}41 = R_1' \, D_6'$$

Reset $I_{3a} = 0^{I}3a = \underline{K_{41}}$

Termination occurs at the end of the word-time in Phase 1.

$$1^{I}1 = I_1' \, \underline{K_{41}}$$

CSA - #13

The CSA Command causes the "SIGN" in the A-register to be reversed.

CSA - #13 - $D_6' D_5' D_4 D_3' D_2 D_1$

CSA is a TYPE #1 Command.

Review OPERAND MODE for TYPE #1 commands.

(1) Gate C(B) into the Z-register.

(2) Gate Channel and Operation Code configuration into the C-and D-registers respectively. (UNINDEXED CONFIGURATION)

(3) Analyze Address

(4) Terminate after ONE WORD-TIME. Transition is to the DIRECTION MODE - Phase 2.

Direction Mode - Phase 2.

Entered via: -

$$0^{K}c = I_4' I_1 \underline{K_{41}}$$

$$0^{I}1 = \underline{I_n} D_5' D_4 T_{41}$$

and terminate after ONE BIT-TIME with transition to Phase 1

$$1^{K}c = \underline{NR_3} D_4 T_1 F_a'$$

Phase 1:

During Phase 1, we will recirculate the A-register contents, BUT CAUSE THE SIGN BIT TO BE REVERSED.

$$1^{A}41 = A_1 \underline{U_1} \underline{D_{53}}$$

$$0^{A}41 = A_1' \underline{U_1} \underline{D_{53}}$$

$$1^{A}40 = A_{41} I_2$$

$$0^{A}40 = A_{41}' I_2$$

$$\underline{U_1} = I_4' I_2 I_1' D_0'$$

$$\underline{D_{53}} = D_5' D_3'$$

A_{41} is inhibited, AT SIGN TIME, by D_3 -

$$1^{D_3} = I_{3a}\, D_6'\, D_5'\, \underline{T_{40}}$$

$$\left\{\begin{array}{l} 1^{I_{3a}} = \underline{U_1}\ \underline{D_{52}}\ T_1 \\ \underline{D_{52}} = D_5'\ D_2 \end{array}\right.$$

and the "SIGN" is copied by A_{41}

$$1^{A_{41}} = C_2\ I_{3a}\ \underline{K_{41}}$$

$$0^{A_{41}} = C_2'\ I_{3a}\ \underline{K_{41}}$$

A_{41} obviously copies C_2. C_2 receives the sign by.

$$1^{C_2} = \underline{U_1}\ T_1\ D_2$$

$$0^{C_2} = A_{40}\ \underline{T_2}\ I_{3a}\ D_6'\ D_4\ D_1\ C_6'\ D_5'$$

with termination at the end of the WORD-TIME.

$$1^{I_1} = I_1'\ \underline{K_{41}}$$

COPY A to R - CAR - #57

The CAR command causes the contents of the R-Register to be replaced by the contents of the A-Register. The address portions of the command has no effects.

$$CAR = D_6\ D_5'\ D_4\ D_3\ D_2\ D_1$$

CAR is a type #1 command. The OPERAND MODE - Phase 1 is terminated after one word-time, with transition to the DIRECTION MODE - Phase 2.

$$0^{I_1} = \underline{I_n}\ D_5'\ D_4\ T_{41}$$

$$0^{Kc} = I_4'\ I_1\ \underline{K_{41}}$$

Phase 2 is terminated after one bit-time

$$1^{Kc} = \underline{NR_3}\ D_4\ T_1\ F_a'$$

and we $\quad 1^{I_{3a}} = \underline{U_1}\ \underline{D_{52}}\ T_1$

$$\underline{I_n} = I_4'\ I_2\ I_1$$

$$\underline{NR_3} = I_4'\ I_1'$$

$$\underline{U_1} = I_4'\ I_2\ I_1'\ D_o'$$

$$\underline{D_{52}} = D_5'\ D_2$$

Command execution is completed in Phase 1.

$$1^{R_{41}} = A_1\ I_{3a}\ I_2\ D_6\ D_4\ D_3$$

$$0^{R_{41}} = A_1'\ I_{3a}\ I_2\ D_6\ D_4\ D_3$$

As the A-register recirculates its information

$$1^{A_{41}} = A_1\ \underline{U_1}\ D_6\ D_5'\ D_1$$

$$0^{A_{41}} = A_1'\ \underline{U_1}\ D_6\ D_5'\ D_1$$

Termination occurs with

$$1^{I_1} = I_1'\ \underline{K_{41}}$$

XAR - #56

The XAR Command causes the C(A) and the C(R) to be exchanged. At the termination of this command the original contents of the A-register will be in the R-register; while the original contents of the R-register will be in the A-register.

$$\text{XAR - \#56} - D_6\ D_5'\ D_4\ D_3\ D_2\ D_1'$$

The XAR Command is a TYPE #1 Command.

Review of OPERAND MODE for TYPE #1 Commands

1. Gate the C(B) into the Z-register
2. Gate Operation and Channel Code into the D-and C-registers respectively. (CHAR-Counter receives the Sector number)
3. Analyze address
4. Terminate after ONE WORD-TIME
5. TRANSITION is to the DIRECTION MODE - Phase 2.

DIRECTION MODE - Phase 2

The DIRECTION MODE is entered via: -

$$0^{I}1 = \underline{I_n}\ D_5'\ D_4\ T_{41}$$

and is maintained for ONE BIT-TIME until

$$1^{K}c = \underline{NR_3}\ D_4\ T_1\ F_a'$$

transition is to Phase 1.

During Phase 1 the A- and the R-register contents are exchanged.

A_{41} copies R_1

$$1\ A_{41} = R_1\ \underline{U_1}\ D_6\ \underline{D_{52}}\ D_3\ D_1'$$

$$0\ A_{41} = R_1'\ \underline{U_1}\ D_6\ \underline{D_{52}}\ D_3\ D_1'$$

---

and R_{41} copies A_1

$$1R_{41} = A_1\ I_{3a}\ D_6\ D_4\ D_2\ I_2$$

$$0R_{41} = A_1'\ I_{3a}\ D_6\ D_4\ D_2\ I_2$$

---

where: $= 1I_{3a} = U_1\ \underline{D_{52}}\ T_1$

$$0I_{3a} = \underline{K_{41}}$$

The command terminates at the end of the word-time

$$1I_1 = I_1'\ \underline{K_{41}}$$

COMPARE - CMG - #35

COMPARE - CME - #35.1

The CMG (#35) command causes the contents of the Memory Location to be COMPARED with the C(A) the result of this comparison will effect the J_o (overflow indicator Flip-Flop) in the following way:

1. If $C(A) > C(W) = 0^{J_o}$
2. If $C(A) < C(W) = 1^{J_o}$
3. If C(A) = C(W) and the "SIGNS" are POSITIVE DO NOT CHANGE THE STATE OF "J_o".
4. If C(A) = C(W) and the "SIGNS" are NEGATIVE, REVERSE THE STATE OF "J_o"
5. If C(A) is equal to the C(W) BUT THE SIGNS" were different: -
   a. J_o was FALSE → (+) A and (-) W = A > W = No Change
      → (-) A and (+) W = W > A = 1^{J_o}
   b. J_o was TRUE → (+) A and (-) W = A > W = 0^{J_o}
      → (-) A and (+) W = W > A = No Change

The CME (#35.1) command causes the contents of the Memory Location to be COMPARED with the C(A). The result of this comparison will effect the J_o Flip-Flop in the following way.

1. If $C(A) \neq C(W)$: 1^{J_o}
2. If C(A) = C(W): NO CHANGE

CMG - #35 - $D_6' D_5 D_4 D_3 D_2' D_1$

CME - #35.1 - $D_6' D_5 D_4 D_3 D_2' D_1 (R_{43})$

Both CMG and CME are TYPE #3 Commands

Review of OPERAND MODE for TYPE # 3 Commands

Phase 1 -

1. Gate C(B) into Z-register (INDEXING PERMITTED)
2. Gate Op-Code and Chan-Code into C- and D-registers.
3. Analyze address for LOOP or MAIN MEMORY

Phase 2.

1. Gate Indexed Chan-Code and Sector-Code into the Chan-register and character counter
2. Reload the D-register
3. Search for Sector Agreement.

Phase 3.

1. Gate C(W) into the B-register.

Transition is to the DIRECTION MODE - Phase 2 which is MAINTAINED FOR ONE(1) BIT-TIME.

$$1^K c = \underline{NR_3}\ D_4\ T_1\ F_a' \qquad \left| \underline{NR_3} = I_4'\ I_1' \right.$$

Transition is to Phase 1

At this time the C(W) is in the B-register. The comparison will be between the C(A) and the C(B).

J_o is the Comparison Indicating Flip-Flop.

$$1^{J_o} = A'_1\ B_1\ I_3\ D_5\ D_4\ D_3 \quad (B > A \text{ for CMG};\ A \neq B \text{ for CME})$$

$$+ J'_o\ A'_1\ T_{41}\ \underline{I_3}\ D_5\ D_4\ D_3\ R'_{43} \quad (\text{"Sign investigation for CMG})$$

$$+ A_1\ B'_1\ D_5\ D_4\ D_3\ R_{43} \quad (A \neq B \text{ for CME})$$

$$0^{J_o} = T_d$$

$$+ A_1\ B'_1\ \underline{I_3}\ D_5\ D_4\ D_3\ R'_{43} \quad (A > B \text{ for CMG})$$

$$+ J_o\ B'_1\ T_{41}\ D_5\ D_4\ D_3\ R'_{43} \quad (\text{Sign investigation for CMG})$$

$$\underline{I_3} = I'_4\ I'_1\ D'_o\ I'_{3a}$$

R_{43} indicates a ONE (1) in the COMMAND INDICATOR bit-position

$$1^{R_{43}} = I'_4\ I_1\ T_1$$

$$0^{R_{43}} = Z'_1\ G'_o\ \underline{T_{22}}\ \underline{I_n}\ D_o$$

$$+ Z'_1\ G_o\ \underline{T_2}\ \underline{I_n}\ D_o$$

The A-register was recirculated normally

$$1^{A_{41}} = A_1\ \underline{I_3}\ D_1$$

$$0^{A_{41}} = A'_1\ I'_4\ D'_2\ F'_a$$

Termination occurs at the conclusion of ONE (1) word time via:

$$1^{I_1} = I'_1\ \underline{K_{41}}$$

ADDITION - SUBTRACTION CHARTS

Regardless of the "signs" or "magnitudes", of either A-register or B-register, ALL CASES of ADDITION or SUBTRACTION will develop into one of the four (4) cases listed.

CASE #I: A + B where the SUM IS LESS THAN ONE (1)
CASE #II: A + B where the SUM IS GREATER THAN ONE (1)
CASE #III: A - B where A is GREATER than B
CASE #IV: A - B where A is LESS than B

	Register	Magnitude	Signs	Operation	Operation Will Be	Type
①	A B	A > B Sum < 1	+ +	Add		
②	A B	A < B Sum < 1	+ +	Add		
③	A B	A + B Sum > + 1	+ +	Add		
④	A B	A > B Sum < 1	+ −	Add		
⑤	A B	A < B	− +	Add		
⑥	A B	A > B	− −	Add		
⑦	A B	A < B	− −	Add		
⑧	A B	A > B	+ +	Sub		
⑨	A B	A < B	+ +	Sub		
⑩	A B	A > B	+ −	Sub		
⑪	A B	A < B	− +	Sub		
⑫	A B	A > B	− −	Sub		
⑬	A B	A < B	− −	Sub		

ADDITION (#73)

SUBTRACTION (#72)

The command ADDITION specifically causes the contents of the LOOP, or MAIN MEMORY location, as specified by the CHANNEL and Sector configuration of the command in the Z-register, to be algebraically added to the contents of the A-register.

The command SUBTRACTION specifically causes the contents of the LOOP, or MAIN MEMORY location, as specified by the CHANNEL and Sector configuration of the command in the Z-register, to be algebraically subtracted from the contents of the A-register.

Either command leaves the contents of the R-register, and the MEMORY location, unaltered.

Both ADD and SUBTRACT are TYPE #3 Commands. This means that they will terminate the OPERAND MODE ($\underline{I_n}$) at T_{41} of MODE - Phase #3. During Mode - Phase #3 the contents of the MEMORY location will be shifted into the B-register.

Transition from the OPERAND - Mode ($I_4' \; I_2 \; K_c' \; D_o$)

will be into the DIRECTION - Mode ($I_4' \; I_2 \; I_1' \; K_c' \; D_o'$)

Phase #2. This will be accomplished via logic

$$0^{I_1} = \underline{N_{11}} \qquad \left| \underline{N_{11}} = I_1' \; I_2 \; D_o \; T_{41} \right.$$

$$0^{D_o} = I_1 \; D_o \; T_{41}$$

The OPERATION CODES for the two commands are:

ADDITION = $\longrightarrow D_6 \; D_5 \; D_4 \; D_3' \; D_2 \; D_1$

SUBTRACTION = $\longrightarrow D_6 \; D_5 \; D_4 \; D_3' \; D_2 \; D_1'$

Entering the DIRECTION MODE causes primary gate $\underline{M_o}$ to become true: -

$$\underline{M_o} = I_{3a}\, D_5\, D_4\, D_3'$$

Where $I_{3a} = \underline{N_{11}}\, D_2$

Before proceeding it is necessary to consider the total manipulations that we wish to, and will perform to accomplish the task of ADDITION or SUBTRACTION.

1. Conversion of all algebraic conditions that might be encountered in either ADD or SUBTRACT into standard ADDITION or SUBTRACTION.
2. Achieve the correct "SIGN" for both "SUM" or "DIFFERENCE" when the correct answer is placed in the A-register.

The question of whether to perform standard addition or subtraction depends not only upon the command, but upon the algebraic signs of the contents of both "A" and "B" registers.

Rule #1	Addition of "A" and "B" registers is required for ADD Commands; IF THEY HAVE THE SAME SIGN.
Rule #2	Addition of "A" and "B" registers is required for SUBTRACT Commands; IF THEY HAVE OPPOSITE SIGNS.
Rule #3	Subtraction of "A" and "B" registers is required for ADD Commands; IF THEY HAVE OPPOSITE SIGNS.
Rule #4	Subtraction of "A" and "B" registers is required for SUBTRACT Commands; IF THEY HAVE THE SAME SIGN.

Considering the four (4) stipulated rules we may now state the four exclusive "cases" that are possible for the ADD-SUBTRACT Commands. Regardless of the Command (ADD or SUBTRACT), the "SIGN" configuration, or the magnitudes of "A" and "B", each combination will fall under one of the four "cases".

Case I	A + B where the sum is LESS THAN ONE (1).
Case II	A + B where the sum is GREATER THAN ONE (1) (Overflow condition).
Case III	A - B where A is GREATER THAN B.
Case IV	A - B where A is LESS THAN B.

Examples of Case I:

(+)	+.6	(+)	+.2	(+)	-.2	(-)	+.6	(-)	-.2
	+.2		+.6		-.6		-.2		+.6
	+.8		+.8		-.8		+.8		-.8

Examples of Case II:

(+)	+.6	(+)	-.6	(-)	+.6	(-)	-.5
	+.5		-.5		-.5		+.6
	+1.1		-1.1		+1.1		-1.1

Examples of Case III:

(+)	+.6	(-)	+.6	(-)	-.6
	-.1		+.1		-.1
	+.5		+.5		-.5

Examples of Case IV:

(+)	-.1	(-)	+.1	(-)	-.1
	+.6		+.6		-.6
	+.5		-.5		+.5

It is now obvious, from the foregoing examples, that the importance of the "SIGN" plus THE MAGNITUDE of the two registers (A and B) must be considered in content before proper placement under a specific "case" can be considered.

It is necessary, however, to first denote the operation as either addition or subtraction (occurs during the first bit-time of the Direction Mode). This is accomplished by logic acting upon D_1 during T_1 of the Direction Mode.

$$1^{D}1 = A'_{41}\ B_{41}\ D'_{1}\ \underline{M_o}\ I_2\ T_1 \quad + A_{41}\ B'_{41}\ D'_{1}\ \underline{M_o}\ I_2\ T_1$$

→ (Command is SUB - #72; Signs are OPPOSITE;) (Perform Addition by Rule #2)

$$0^{D}1 = A_{41}\ B'_{41}\ \underline{M_o}\ D_1\ I_2\ T_1 \quad + A'_{41}\ B_{41}\ \underline{M_o}\ D_1\ I_2\ T_1$$

→ (Command is ADD - #73; Signs are OPPOSITE;) (Perform Subtraction by Rule #3)

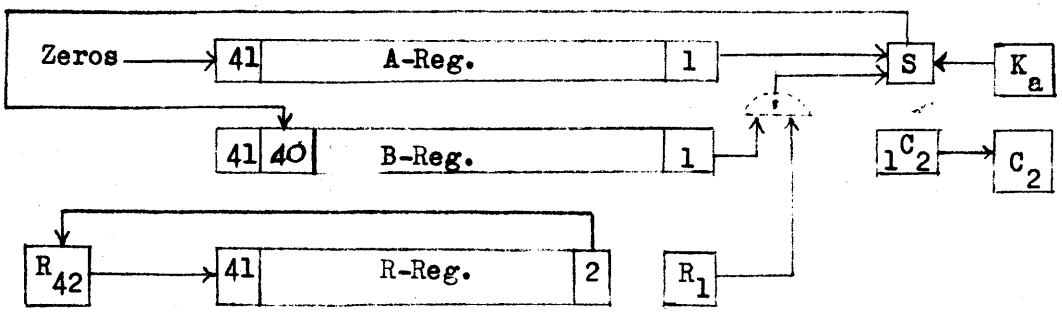
When the Signs are THE SAME, the specified operation is carried out.

We may now proceed to the actual operations of addition or subtraction. Execution of both commands will take two (2) word-times. One (1) word-time in the Direction Mode and one (1) word-time in the Execute Mode.

During the Direction Mode both "A" and "B" registers will be shifted right. Their entire contents will be monitored at A_1, and B_1, and added together. K_a will act as the "carry" or "borrow" flip-flop while R_1 will be the over all gating flip-flop for the addition or subtraction. S flip-flop will be the "SUM' or "DIFFERENCE" element. During the Direction Mode:

1. The "Sum" or "Difference" is copied into B_{40} (because of the one (1) bit delay in S)
2. The A-register is reset to zero.
3. The R-register recirculates $R_2 \longrightarrow R_{42}$.
4. $1^{C}2$ on the assumption that the "sign" of the answer will be positive.

This would be pictorically represented by: -



* Representation of B_1 and R_1 as a single "AND" signal is essentially for illustration. The inspected logic will show that whenever B_1 is used, R_1 will be included. For B_1' the "gate" would be false, hence R_1 is redundant.

When considering the output of S and K_a from the inputs A_1, B_1, and K_a, the use of the truth table simplifies matters.

$$0^{K_a} = N_{11}\, D_2 \quad (K_a \text{ is initially ZERO})$$

$$1^{S} = A_1\, B_1'\, K_a'\, I_{3a} + \underline{R_{10}}\, K_a'\, I_{3a} + B_1\, \underline{R_{01}}\, K_a'\, I_{3a} + \underline{S_{00}}\, K_a\, I_{3a} + \underline{R_{10}}\, K_a\, I_{3a} + \underline{S_{11}}\, K_a$$

$$0^{S} = \underline{S_{00}}\, K_a'\, I_{3a} + \underline{R_{00}}\, K_a'\, I_{3a} + B_1\, \underline{R_{01}}\, K_a\, I_{3a} + A_1\, B_1'\, K_a\, I_{3a} + \underline{R_{10}}\, I_{3a}\, K_a + \underline{S_{11}}\, K_a'\, T_{41}'$$

$$\underline{R_{10}} = A_1\, R_1'$$

$$\underline{R_{01}} = A_1'\, R_1$$

$$\underline{S_{00}} = A_1'\, B_1'\, C_4'$$

$$\underline{R_{00}} = A_1'\, R_1'\, C_4'$$

$$\underline{S_{11}} = A_1\, B_1\, R_1\, I_{3a}$$

The additional logic for K_a during ADD or SUBTRACT is:

$$1^{K}a = \underline{S_{11}}\ D_1\ D_2\ T'_{41} + B_1\ \underline{R_{01}}\ D'_1\ I_{3a}\ T'_{41}\ \left[C'_{3x}\right]\ D_5 \qquad 0^{C}3 = I_{3a}\ P'_5$$

$$0^{K}a = \underline{S_{oo}}\ D_1\ I_{3a}\ K_a + \underline{R_{oo}}\ D_1\ I_{3a}\ K_a + A_1\ B'_1\ D'_1\ I_{3a}$$
$$+ \underline{R_{10}}\ D'_1\ I_{3a} + I_{3a}\ T_{41}$$

For either addition or subtraction R_1 is always "set" at T_1

$$1^{R}1 = \underline{M_o}\ T_1$$

$$0^{R}1 = I_{3a}\ D_5\ D_6\ \underline{T_{40}}$$

The A-register is "reset" to zero by:

$$0^{A}41 = I'_4\ T_1\ F'_a$$

There is no $1^{A}41$ logic that can come true during this word-time.

The output of "S" is copied by B_{40} via

$$1^{B}40 = S\ \underline{M_o}\ I_2$$

$$0^{B}40 = S'\ \underline{M_o}\ I_2$$

The contents of the R-register is recirculated by

$$1^{R}42 = R_2\ \underline{M_o}\ T'_{41} \qquad 1^{R}41 = R_{42}\ \underline{M_o} \qquad 1^{R}40 = R_{41}\ D_4$$

$$0^{R}42 = R'_2\ \underline{M_o} \qquad 0^{R}41 = R'_{42}\ \underline{M_o} \qquad 0^{R}40 = R'_{41}\ D_4$$

It is now necessary to distinguish "cases" where the knowledge of COMMAND, and SIGN, are not explicit enough.

Case I: - Sign and Command are sufficient.

Case II: - If we generate a "CARRY" (K_a = true) at T_{41}, we are assured that the SUM is greater than ONE (1). This condition is OVERFLOW. A Overflow causes J_o flip-flop to "SET".

$$1^{J}o = K_a \; T_{41} \; I_2 \; D_1 \; \underline{NR_{50}} \qquad \left[\underline{NR_{50}} = I_{3a} \; D_5 \; D_4 \; D_3' \; FM'\right.$$

J_o will cause the overflow neon to be turned on. If the next command is TRANSFER ON OVERFLOW - (#52) the neon will be turned off because J_o will be reset.

Case II: - The greater magnitude of the A-register is detected at T_{41} of the DIRECTION MODE because no "borrow" indication in K_a (K_a') will be present. In this case the correct "DIFFERENCE" will be IN THE B-REGISTER at T_{41} of the Direction Mode. During the one word-time that we spend in the EXECUTE MODE, we will simply Add this correct answer to the zeroes in the A-register and deposit the unaltered, correct answer, into the A-register.

Case IV: - The greater magnitude of the B-register is detected at T_{41} of the DIRECTION MODE because of the "borrow" indication in K_a (K_a = true), will be present. In this case the COMPLEMENT ($2^{\underline{S}}$) of the correct "DIFFERENCE" will be IN THE B-register at T_{41} of the Direction Mode. During the one word-time that we spend in the Execute Mode we will Subtract this complemented answer from the zeros in the A-register and deposit the correct, "RECOMPLEMENTED" answer into the A-register.

In Cases I, II, and III we will always have the correct answer in the B-register at T_{41} of the DIRECTION MODE. We will always Add this correct answer to the zeros in A during the word-time spent in the EXECUTE MODE.

In Case IV we will always have the Twos complement of the correct answer in the B-register at T_{41} of the DIRECTION MODE. We will always subtract this complemented answer from the zeros in the A-register during the word-time spent in the EXECUTE MODE.

At T_{41} of the DIRECTION MODE the logic comes true for transition into the EXECUTE MODE.

$$0^{I}2 = \underline{M_o}\ \underline{K_o} \qquad\qquad \underline{K_o} = T_{41}\ K'_c$$

Transition is into EXECUTION MODE - PHASE 2.

Gate U_i comes true

$$\underline{U_i} = I'_4\ I'_2\ I'_1$$

C_2, which had been "set" during the DIRECTION MODE ($1^{C}2 = \underline{U_1}\ T_1\ D_2$) samples the state of S for "negative sign" implications.

$$0^{C}2 = S'\ \underline{M_o}\ I'_2\ T_1$$

K_c will be forced TRUE at T_1 of $\underline{U_i}$

$$1^{K}c = I'_4\ I'_1\ D_4\ T_1$$

forcing us into PHASE 1 after one bit-time.

The correct "sign" will be copied into A_{41} at T_{41}, from C_2

$$1^{A}41 = C_2\ I_{3a}\ \underline{K_{41}}$$

$$0^{A}41 = C'_2\ I_{3a}\ \underline{K_{41}}$$

For correct execution of Case III examples it is necessary to "set" D_1 as we enter EXECUTE.

$$1^{D}1 = K'_a \; T_{41} \; FA' \; \underline{NR_{50}}$$

so that addition might occur.

The logic for the B-register remains the same, as does the logic used to recirculate the contents of the R-register.

During EXECUTE the A-register picks up the result from S at A_{40}

$$1^{A}40 = S \; I'_2 \; \underline{NR_{50}}$$

$$0^{A}40 = S' \; I'_2 \; \underline{NR_{50}}$$

Addition or Subtraction is then completed.

Transition is to $\underline{I_c}$ or $\underline{I_n}$ via

$$1^{I}1 = I'_1 \; \underline{K_{41}}$$

EXTRACT - #70 - EXT

The EXTRACT COMMAND will cause the contents of the A-register to be compared with the Operand, in the Memory location specified by the address portion of the command.

Where this comparison shows a coincidence of BINARY"ONES", at a specific bit position, a binary one is retained, by the A-register, in that bit position. For those bit positions where a lack of BINARY ONES are coincident the A-register MUST place a zero.

The contents of the R-register and the Memory location are not changed.

EXT - #70 - $D_6\ D_5\ D_4\ D_3'\ D_2'\ D_1'$

Extract is a Type #3 command. Type #3 commands specify that the OPERAND MODE be maintained until T_{41} of Phase 3.

Summary of Operand Mode -

Phase 1 - Sense Address and load the D and C registers with the UNINDEXED configuration from the B-register. Maintain for one-word-time.

Phase 2 - Load the Channel Register and the character counter with the INDEXED configuration from the Z-register. Reenter the Operation Code into the D-register. Search for Sector Agreement between the Z-register and the Sector channel. Remain until $S_o'\ T_{41}$.

Phase 3 - Gate the Operand from Memory and into the B-register. Maintain for one-word-time.

Transition is from Operand Mode - Phase 3 into the DIRECTION MODE - Phase 2 where we remain for ONE BIT-TIME.

$$0^{I}1 = \underline{N_{11}}$$

$$0^{D}o = I_1 D_o T_{41}$$

$$\underline{N_{11}} = I_2 I_1 D_o T_{41}$$

ENTER DIRECTION MODE - Phase 2 -

$$1^{K}c = \underline{NR_3} D_4 T_1 F'_a$$

$$\underline{NR_3} = I'_4 I'_1$$

ENTER Phase 1, after ONE BIT - TIME

The coincidence of "Ones" is used to "set" A

$$1^{A}41 = A_1 B_1 \underline{I_3}$$

$$\underline{I_3} = I'_4 I'_1 D'_o I'_{3a}$$

Zeros in either A or B is used to "reset" A

$$0^{A}41 = A'_1 D'_2 F'_a I'_4 + B'_1 I_3 D_5 D'_1$$

Sync bit will be zero via

$$0^{A}41 = I'_4 T_1 F'_a$$

The R-register recirculates because of

$$1^{R}41 = R_1 I'_{3a}$$

$$0^{R}41 = R'_1 I'_{3a}$$

Termination is via:

$$1^{I}1 = \underline{K_{41}} I'_1$$

COPY TO L-LOOP - #40 - CTL

COPY TO V-LOOP - #41 - CTV

The Commands CTL - #40 " and "CTV - #41", will force the replacement of words in the specified loop. This replacement will start with the word whose least significant address "OCTAL DIGIT" is the same as that of the address specified in the command.

The replacement of words, in the loop, will be with consecutive words, starting with the word addressed, until 7767 or 7777 is filled.

Examples:

CTL - 1230 will replace words 7760 to 7767 with 1230 to 1237

CTL - 1235 will replace words 7765 to 7767 with 1235 to 1237

CTL - 1237 will replace the word at 7767.

The explanation for CTV is completely anelogous to the CTL command. *

The OPERATION CODES are:

CTL - #40 - $D_6\ D_5'\ D_4'\ D_3'\ D_2'\ D_1'$

CTV - #41 - $D_6\ D_5'\ D_4'\ D_3'\ D_2'\ D_1$

Both CTL and CTV are Type #2 Commands.

During Phase - 1 of the OPERAND MODE the address is sensed, in the usual manner, as MAIN MEMORY by monitoring G_{27} through G_{34} at Z_{41} flip-flop.

* NOTE: It is impossible for the CTL and CTV commands to copy from one loop into the other. If a loop address is specified in the command, the replacement words will come from the "Grey Area" of Main Memory.

The Operation Code and the Channel configuration are transferred into the "D" and "C" registers from the B-register.

Command Mode - Phase 1 is, of course, terminated after one word-time by:

$$0^{K}c = I'_4\ I_1\ \underline{K_{41}}$$

and we enter the OPERAND MODE - Phase 2

The typical quality of type #2 commands is that they remain in Phase 2 of the OPERAND MODE for only one word-time. During this word-time three main duties are always accomplished.

1. The (POSSIBLY INDEXED) CHANNEL-ADDRESS is gated into the C-register through the D-register.
2. The (POSSIBLY INDEXED) CHARACTER COUNT is gated into the G-register.
3. The OPERATION CODE is REENTERED INTO the D-register.

Phase 2 is terminated by: -

$$0^{I}1 = \underline{I_n}\ D'_5\ \underline{K_o}$$

The DIRECTION MODE - Phase 2 is entered.

The DIRECTION MODE is entered by both commands. <u>CTL and CTV do not use this mode to accomplish any of their execution.</u>

The Direction Mode is terminated after one word-time by:

$$0^{I}2 = \underline{NR_3}\ K_o\ D'_4 \qquad \left[\underline{NR_3} = I'_4\ I'_1\right.$$

The EXECUTE MODE - Phase 2 is entered.

Our configuration, at this time, is $I'_4\ I'_2\ I'_1\ K'_c\ D'_o$.

We will proceed to search for Sector Agreement for writing using the usual comparison logic.

$$1^{S}o = K'_c\ D'_o\ Z_1\ S'_1\ N_1\ T'_{41} + K'_c\ D'_o\ Z'_1\ N_1\ T'_{41}$$

$$0^{S}o = T_{41}$$

$$1^{N}1 = G_o\ \underline{T_2}\ K'_c + G'_o\ \underline{T_{22}}\ K_c$$

$$0^{N}1 = \underbrace{P_5\ P'_4\ P'_1\ K'_c}_{T_8 \text{ or } T_{28}}$$

When Sector-Agreement is reached, Phase #2 is terminated in favor of Phase 3 by: -

$$1^{D}o = S'_o\ \underline{U_i}\ D'_4\ D'_2\ D'_o\ T_{41}$$

Since gating to the Loop is variable, from one (1) to eight (8) words, "D_o" is not going to be reset until a zero (0) is detected in the most significant bit position of the SECTOR TRACK. A zero is permanently recorded in the most significant bit position for sectors 07, 17, 27, 37, 57, 67, 77 of the sector track only.

$$1^{K}c = S'_1\ \underline{U_i}\ D_o\ \underline{T_{40}}$$

$$0^{D}o = \underline{K_{41}}\ G'_1\ N'_1$$

The actual writing into the loops is accomplished by

$$^{L}1_{41} = M_r \underline{M_6} D'_3 D'_1$$

$$^{L}0_{41} = M'_r \underline{M_6} D'_3 D'_1$$

$$^{V}1_{41} = M_r \underline{M_6} D'_3 D_1$$

$$^{V}0_{41} = M'_r \underline{M_6} D'_3 D_1$$

$$\underline{M_6} = I'_2 I'_1 D_o D'_4 D'_2$$

Both A and R registers recirculate normally

$$^{A}1_{41} = A_1 \underline{U_1} \underline{D_{53}} + A_1 D'_5 I'_4 T'_1$$

$$^{A}0_{41} = A'_1 \underline{U_1} \underline{D_{53}} + A'_1 D'_2 I'_4 F'_a + T_1 I'_4 F'_a$$

$$^{R}1_{41} = R_1 I'_{3a}$$

$$^{R}0_{41} = R'_1 I'_{3a}$$

$$\underline{U_1} = I'_4 I_2 I'_1 D'_o$$

$$\underline{D_{53}} = D'_5 D'_3$$

The normal recirculation of L and V-loops is by

$$^{L}1_{41} = L_1 D'_o + L_1 I_2 + L_1 I_1 + L_1 D_3 K'_a + L_1 D_2$$

$$^{L}0_{41} = L'_1 D'_o + L'_1 I_2 + L'_1 I_1 + L'_1 D_3 K'_a + L'_1 D_2$$

$$^{V}1_{41} = V_1 D'_o + V_1 I_2 + V_1 I_1 + V_1 D_2$$

$$^{V}0_{41} = V'_1 D'_o + V'_1 I_2 + V'_1 I_1 + V'_1 D_2$$

The loops are recirculated unless new information is being filled in by copy to the loops or a Store Command.

When copy to V-loop occurs, the L-loop must be recirculated by special logic

$$1^{L_{41}} = L_1 \; S$$

$$0^{L_{41}} = L_1' \; S$$

Where S is forced prime every word-time by during CFL, CTL, CTV and STO by

$$0^{S} = \underline{I_3} \; K_a' \; T_{40} \; \underline{N_{10}}$$

If the command is CTV the state of S is reversed at T_{41} by: =

$$1^{S} = D_6 \; D_3' \; D_1 \; \underline{I_3} \; \underline{N_{10}} \; T_{41}$$

$$\begin{cases} \underline{I_3} = I_4' \; I_1' \; D_0' \; I_{3a}' \\ \underline{N_{10}} = D_4' \; D_2' \end{cases}$$

COPY FROM L-LOOP - #44 - CFL

The Command CFL forces the replacement of words in Main Memory by words in the L-Loop. This replacement will start at the memory locations specified by the address in the command.

The replacement of words in Main Memory will be consecutive words, starting with the specified address and continuing until L-Loop sector 7767 has been written (into Main Memory).

Examples: -

CFL - 1230 will replace words 1230 to 1237 with 7760 to 7767

CFL - 1235 will replace words 1235 to 1237 with 7767 to 7767

CFL - 1237 will replace word 1237 with 7767

The logic for CFL is similar to that used for the STO Command, except that gating lasts from one to eight word times, and the information is transferred from the L-Loop to Main Memory.

The OPERATION CODE is

CFL - #44 $D_6\ D_5'\ D_4'\ D_3\ D_2'\ D_1'$

The Logic for CFL is the same as CTL & CTV EXCEPT for the "work" implication during EXECUTE-MODE-PHASE 3.

Writing is accomplished into Main Memory via

$1^{M}_{w1} = L_1\ \underline{M_6}\ \underline{L_m}\ T_1'$ $\quad 1^{M}_{w2} = L_1'\ \underline{M_6}\ \underline{L_m}$

$0^{M}_{w1} = L_1'\ \underline{M_6}\ \underline{L_m}$ $\quad 0^{M}_{w2} = L_1\ \underline{M_6}\ \underline{L_m}\ T_1'$

$\underline{M_6} = I_2'\ I_1'\ D_0\ D_4'\ D_2'$

$\underline{L_m} = D_3\ D_1'$

ALL Recirculation Logic is the same.

LOAD INDEX - #31.1

The command LOAD INDEX causes the configuration (contents) of the INDEX REGISTER to be replaced by the right hand address portion of the word addressed.

Load INDEX is a type #3 command. We will remain in the OPERAND MODE until we have gated the contents of the MEMORY LOCATION, as specified in the Z-register, into the B-register.

Review: - OPERAND MODE - Phase 1,

1. The "Operand" address is sensed as MAIN MEMORY or Loop by monitoring $G_{27} \longrightarrow G_{34}$ at Z_{41} during N_1
2. The Operation Code and the Channel Address are shifted into the "D" and "C" registers from the B-register.
3. If the transition was from the COMMAND MODE - Phase 2 the command - pair would be gated into the Z-register from the B-register. INDEXING would occur if specified.
4. Transition is to Phase 2.

Operand Mode - Phase 2.

1. During the first (1st) word time: -
   a. The Operation Code and the INDEXED (if specified) Channel Address are gated into the "D" and "C" registers from the Z-register during N_5.
   b. The INDEXED (if specified) CHARACTER COUNT is gated into the G-register from C_1 during K_g.
2. Phase 2 is continued until Sector Agreement is detected by S'_o.
3. Transition is then to Phase 3.

Phase 3.

1. The "operand" is gated out of Memory by D_o, and into the B-register.
2. Phase 3 is maintained for one (1) word time.

Transition is to the Direction Mode - Phase #2 for 1 bit time, and then into Phase 4 at T_1. -

$$1^{D}o = \underline{U_1}\ \underline{M_7}\ T_1 \qquad \left| \underline{M_7} = D_4\ D_3'\ D_2'\ D_1 \right.$$

$$1^{K}c = \underline{NR_3}\ D_4\ T_1\ F_a' \qquad \left| \underline{NR_3} = I_4'\ I_1' \right.$$

LOAD INDEX LOGIC EXPLANATION

The "COMPLEMENT" of the B-register is shifted into the G-register by the main two signals N_5' and R_{42}

$$1^{G}41 = B_1'\ \underline{H_o}\ N_5'\ R_{42}\ D_5$$

$$0^{G}41 = B_1\ \underline{H_o}\ N_5'\ R_{42}\ D_5$$

LDI - #31.1 $D_6'\ D_5\ D_4\ D_3'\ D_2'\ D_1$

Where $\underline{H_o} = I_1'\ D_o\ I_{3a}'$

R_{42} is True via: - $\quad 1^{R}42 = D_2'\ D_1\ R_{43}\ \underline{U_1}\ T_1$

Where $\underline{U_1}\ I_4'\ I_2\ I_1'\ D_o'$ and R_{43} had been previously set by

$$1^{R}43 = I_4'\ T_1\ I_1 \qquad (T_1 \text{ of phase 3 in the OPERAND MODE})$$

R_{43} was unable to Reset

$$0^{R}43 = Z_1'\ G_o'\ \underline{T_{22}}\ \underline{T_n}\ D_o + Z_1'\ G_o\ \underline{T_2}\ \underline{T_n}\ D_o$$

NOTE: The Command Indicator had to be a "one" (1) making

Z_1 ture at $\underline{T_2}$ or T_{22}

$$0^{N}5 = T_{41}$$

$$1^{N}5 = K_c \; D_o \; \underline{T_{21}}$$

And $0^{R}42 = I'_{3a} \; K_c \; \underline{T_{14}}$ terminates the timing to include only the Address bits.

The command is terminated by

$$1^{I}1 = I'_1 \; \underline{K_{41}}$$

ALS - #02 - Accumulator Left Shift.

ARS - #03 - Accumulator Right Shift.

ASV - #02.2 - Accumulator Left Shift - Overflow

ASC - #02.4 - Accumulator Left Shift - Normalize

LLS - #42 - Long Left Shift, A-and R-registers.

LRS - #43 - Long Right Shift, A-and R-registers.

LSV - #42.2 - Long Left Shift-Overflow - A and R registers.

LSC - #42.4 - Long Left Shift - Normalize - A and R registers.

The ARS and ALS commands will cause the C(A) to be shifted "right" or "left", respectively, the number of bit positions specified by the Sector configuration and the "least significant bit" of the Channel configuration. These bit positions allow a maximum of a 127 bit shift. Bits leaving the A-register, due to the shift, are lost. Zeros are placed in the bit positions that have been vacated by the shift. The "sign" is not effected.

The ASV and ASC commands have the same effect as the ALS command, with exceptions:

1. ASV will TURN ON THE OVERFLOW INDICATOR, if a binary one (1) is shifted past the most significant bit position.
2. ASC will terminate when the C(A) either achieves the desired shift or becomes NORMALIZED whichever occurs first. The C(I) is decremented as per the number of bit positions shifted.

The LLS and LRS commands will cause the A and R registers to shift left or right as though they were one 78 bit register. The degree of shift is determined in the same manner as for ARS and ALS. The "sign" bit positions are ignored in the shifting process. Bits shifted out of the left end of "A" or the right end of "R" are lost. Zeros are placed in the bit positions that have been vacated by the shift.

For LLS: make the sign of "A" the same as the sign of "R".

For LRS: make the sign of "R" the same as the sign of "A".

The LSV and LSC commands have the same effect as the LLS command with exceptions:

1. LSV will TURN ON THE OVERFLOW INDICATOR, if a binary one (1) is shifted past the most significant bit position of the A-register.
2. LSC will terminate when the degree of shift has been achieved, or the A-register becomes NORMALIZED, WHICHEVER OCCURS FIRST.
3. The C(I) is decremented as per the number of bit positions shifted.

ALS - #02 - $D_6' D_5' D_4' D_3' D_2 D_1' C_6' C_5'$

ARS - #03 - $D_6' D_5' D_4' D_3' D_2 D_1 C_6' C_5'$

ASV - #02.2- $D_6' D_5' D_4' D_3' D_2 D_1' C_6' C_5$

ASC - #02.4- $D_6' D_5' D_4' D_3' D_2 D_1' C_6 C_5'$

LLS - #42 - $D_6 D_5' D_4' D_3' D_2 D_1' C_6' C_5'$

LRS - #43 - $D_6 D_5' D_4' D_3' D_2 D_1 C_6' C_5'$

LSV - #42.2- $D_6 D_5' D_4' D_3' D_2 D_1' C_6' C_5$

LSC - #42.4- $D_6 D_5' D_4' D_3' D_2 D_1' C_6 C_5'$

All shift commands are TYPE #2 commands.

Review of the OPERAND MODE - for TYPE #2 commands.

Phase 1:

1. Gate the C(B) into the Z-register.
2. Gate the UNINDEXED Channel number and Sector number into the C-register and CHARACTER COUNTER, respectively. Load the D-register.

3. Analyze the address.

4. Terminate after ONE WORD TIME. Transition is to Phase 2.

Phase 2.

1. Reload the C-register and the CHARACTER COUNTER with the INDEXED Channel & Sector configuration. Reload the D-register.

2. Terminate after ONE WORD TIME with transition to the DIRECTION MODE - Phase 2.

DIRECTION MODE - Phase 2.

All of the SHIFT COMMANDS enter the DIRECTION MODE - Phase 2 via

$$0^{I}1 = \underline{I_n}\ D'_5\ \underline{K_o}$$

During Phase 2 we will:

1. Zero the B-register - for ALL COMMANDS (SHIFT)

$$0^{B}40 = \underline{U_1}\ \underline{D_{52}} + B'_{41}\ D'_4$$

$$0^{B}41 = B'_1\ I_{3a}\ D_2$$

$$\underline{D_{52}} = D'_5\ D_2$$

$$\underline{U_1} = I'_4\ I_2\ I'_1\ D'_o$$

2. Set I_{3a}: $\quad 1^{I}3a = \underline{U_1}\ \underline{D_{52}}\ T_1$

3. Store the proper sign in C_2.

$$1^{C}2 = \underline{U_1}\ T_1\ D_2 \qquad \text{(Assume Positive Sign)}$$

$$0^{C}2 = A'_{40}\ D'_6\ \underline{D_{52}}\ D'_4\ \underline{U_1}\ T_2 \qquad \text{(Force Negative)}$$

(R-register sign governs C_2)

$+R'_{40}\ D_6\ \underline{D_{52}}\ D'_1\ \underline{U_1}\ T_2$ (Force Negative)

A-register sign governs C_2)

$+ A'_{40}\ D_6\ \underline{D_{52}}\ D_1\ \underline{U_1}\ T_2$ (Force Negative)

4. "Set" D_3 for ALL LEFT SHIFT COMMANDS

$$1^{D_3} = I_{3a}\ D_5'\ D_1'\ \underline{T_{40}}$$

5. Check for NORMALIZED A-register, Commands LSC and ASC.

If either of these commands finds the A-register NORMALIZED, we have immediate transition to Phase 1, at T_{40}, and command termination at T_{41}.

$$1^{K_c} = A_1\ T_{40}\ \underline{U_1}\ \underline{D_{52}}\ D_4'\ D_1'\ C_6$$

(Only applicable for LSC & ASC)

K_c true at T_{41} would cause $1^{I_1} = I_1'\ \underline{K_{41}}$

6. Recirculate the A-and R-registers.

$$1^{A}41 = A_1\ \underline{U_1}\ \underline{D_{53}}$$

$$0^{A}41 = A_1'\ \underline{U_1}\ \underline{D_{53}}$$

$$1^{R}41 = R_1\ \underline{I_{2d}}\ D_4' + R_1\ D_6'$$

$$0^{R}41 = R_1'\ \underline{I_{2d}}\ C_4' + R_1'\ D_6'$$

$$\underline{D_{53}} = D_5'\ D_3'$$

$$\underline{I_{2d}} = I_2\ D_5'\ D_3'$$

$$1^{C}4 = I_{3a}\ D_5'\ D_4'\ \underline{T_{40}}$$

$$0^{C}4 = \underline{N_{R3}}\ T_1\ D_2$$

For both the A-and R-registers ALL BUT THE SIGN BIT is recirculated.

7. Allow the CHARACTER COUNTER to count up.

$$1^{G}41 = G_1\ K_g'\ T_{41}' + G_1'\ K_g\ I_1'\ D_o'$$

$$0^{G}41 = G_1'\ K_g'\ T_{41}' + G_1\ K_g\ I_1'\ D_o'$$

$$1^{K_g} = G_o\ I_4'\ \underline{T_{14}}\ P_6'\ P_1 + G_o'\ I_4'\ \underline{T_{14}}\ P_6\ P_1$$

$$0^{K_g} = \underline{T_{20}} + G_1'\ K_g\ I_1'\ D_o'$$

8. Terminate the DIRECTION MODE - Phase 2 after ONE WORD TIME with transition to the EXECUTE MODE - Phase 2.

EXECUTE MODE - Phase 2 is entered via - $0^{I}2 = \underline{NR_3}\ \underline{K_o}\ D'_4$

During EXECUTE MODE - Phase 2 the degree of shift specified, and the type of shift specified will be accomplished for ALL SHIFT COMMANDS

ARS Fig. 1

A-register RIGHT SHIFT (Figure 1) is accomplished by causing "S" to copy A_1, and then A_{39} copies "S". The S-flip-flop is simply a delay flip-flop.

$$1^{S} = A_1\ B'_1\ K'_a\ I_{3a}$$

$$0^{S} = \underline{S_{oo}}\ K'_a\ I_{3a}$$

$$1^{A}39 = S\ \underline{M_{SR}}\ T'_1$$

$$0^{A}39 = S'\ \underline{M_{SR}}\ T'_1 + A'_{40}\ T_1\ I'_2$$

$$0^{K}a = I_{3a}\ T_{41}$$

$$\underline{S_{oo}} = A'_1\ B'_1\ C'_4$$

$$\underline{M_{SR}} = D'_4\ D'_3\ D_1\ I_{3a}\ I'_2$$

Leading zeros (0) are copied from the primed A_{40}: $0^{A}40 = K'_a\ \underline{M_{SR}}\ F'_a$

K_a will remain primed, for all shift commands because B_1 cannot come true.

The A-register will continue RIGHT SHIFTING in this manner, until TERMINATION IS INDICATED.

The correct sign will be forced into A_{41} at the last bit-time of commands execution: -

$$1^{A}41 = C_2\ I_{3a}\ \underline{K_{41}}$$

$$0^{A}41 = C'_2\ I_{3a}\ \underline{K_{41}}$$

ALS, ASV, ASC
Fig. 2

The A-register LEFT SHIFT (Fig. 2) is accomplished by causing "S" to copy A_1 and then A_{41} copies "S".

$$1^S = A_1\ B'_1\ K'_a\ I_{3a}$$

$$0^S = \underline{S_{oo}}\ K'_a\ I_{3a}$$

$$1^{A}41 = S\ \underline{M_{SL}}\ T'_1\ D'_o$$

$$0^{A}41 = S'\ \underline{M_{SL}}\ D'_o$$

$$\underline{M_{SL}} = D'_4\ D_3\ I_{3a}\ C'_4$$

with "trailing zeros" inserted into A_{41} at T_1

$$0^{A}41 = T_1\ I'_4\ F'_a$$

An OVERFLOW is detected for ASV by

$$1^{J}o = A_1\ \underline{T_{40}}\ D'_5\ \underline{M_{SL}}\ C_5$$

A NORMALIZED A-REGISTER WILL TERMINATE THE COMMAND for ASC.

$$1^{K}c = S\ \underline{T_{40}}\ D'_5\ \underline{M_{SL}}\ C_6$$

(Otherwise termination will occur when the prescribed degree of shift has been achieved)

The contents of the INDEX REGISTER will be increased by one (1) each word time we remain in Phase 2, FOR ASC & LSC ONLY.

$$1^{G}41 = G_1\ K'_g\ T'_{41}\ \text{(Normal Recirculation)}$$
$$+\ G'_1\ K_g\ I'_1\ D'_o\ \text{(Add logic)}$$

$$0^{G}41 = G'_1\ K'_g\ T'_{41}$$
$$+\ G_1\ K_g\ I'_1\ D'_o$$

K_g = Carry Flip-flop: $1^{K}g = T_2\ \underline{M_{SL}}\ D'_5\ C_6$ (ASC ONLY)

$$0^{K}g = G'_1\ K_g\ I'_1\ D'_o + \underline{T_{20}}$$

And the correct sign will be forced into A_{41} at the last bit-time of commands execution

$$1^{A}41 = C_2\ I_{3a}\ \underline{K_{41}}$$

$$0^{A}41 = C_2'\ I_{3a}\ \underline{K_{41}}$$

The A-register will continue LEFT SHIFTING in this manner until TERMINATION IS INDICATED.

LRS
Fig. 3

A-and R-registers LONG RIGHT SHIFT (Fig. 3) is accomplished by causing "S" to copy A_1, and A_{39} to copy "S". (This takes care of the A-register right-shift.) At the same time R_{40} receives the L.S.B. of the A-register by copying A_2 at T_{41}. The R-register right-shift is completed by allowing R_{41} to copy R_2.

(1) $1^{S} = A_1\ B_1'\ K_a'\ D_o'$

$0^{S} = \underline{S_{oo}}\ K_a'\ D_o'$

---

(2) $1^{A}39 = S\ \underline{M_{SR}}\ T_1'$ (Not true for Right Shift

$0^{A}39 = S\ \underline{M_{SR}}\ T_1' + A_{40}'\ T_1\ I_2'$ $\left[\underline{M_{SR}} = D_4'\ D_3'\ D_1\ I_{3a}\ I_2'\right.$

$0^{A}40 = K_a'\ \underline{M_{SR}}\ F_a'$ (Leading Zeros)

---

(3) $1^{R}40 = A_2\ \underline{M_{SD}}\ T_{41}$ $\left[\underline{M_{SD}} = D_6\ D_4'\ D_3'\ D_1\ I_{3a}\right.$

$0^{R}41 = A_2'\ \underline{M_{SD}}\ T_{41}$

---

(4) $1^{R}41 = R_2\ \underline{M_{SD}}\ I_2'\ T_{41}'$

$0^{R}41 = R_2'\ \underline{M_{SD}}\ I_2'\ T_{41}'$

Leading zeros (0) are forced into the MSB position of the A-register via:

$$0^{A}39 = A'_{40}\ T_1\ F'_a$$

Where: -

$$0^{A}40 = K'_a\ \underline{M_{SR}}\ F'_a$$

The correct sign is forced into the A-and R-registers via:

$$1^{A}41 = C_2\ I_{3a}\ \underline{K_{41}}$$

$$0^{A}41 = C'_2\ I_{3a}\ \underline{K_{41}}$$

$$1^{R}41 = C_2\ \underline{M_{SD}}\ \underline{K_{41}}$$

$$1^{R}41 = C'_2\ \underline{M_{SD}}\ \underline{K_{41}}$$

The A-and R-registers will continue RIFHT SHIFTING in this manner until TERMINATION IS INDICATED.

LLS, LSV, LSC
Fig. 4

The A-and R-registers LONG LEFT-SHIFT (Fig. 4) is accomplished by causing "S" to copy $^{(1)}$ A_1 and A_{41} to copy $^{(2)}$ "S". (This takes care of the A-register LEFT-SHIFT). The LEFT-SHIFT of the R-register is caused when R_{43} and R_{42} are inserted in the R-register recirculation, with $^{(3)}$ R_1 omitted, to cause a one (1) bit left precission. The M.S.B. of the R-register is transferred to the L.S.B. position of the A-register by allowing "S" to copy R_{40} and then A_{41} copies $^{(3)}$ "S".

1. $1^{S} = A_1\ B'_1\ K'_a\ I_{3a}$

   $0^{S} = \underline{S_{oo}}\ K'_a\ I_{3a}$

2. $1^{A}41 = S\ \underline{M_{SL}}\ T'_1\ D'_o$

   $0^{A}41 = S'\ \underline{M_{SL}}\ D'_o$

3. $1^{R}43 = R_2\ I_{3a}\ T'_{41}$

$0^{R}43 = R'_2\ I_{3a}$

$1^{R}42 = R_{43}\ \underline{M_{SL}}\ T'_{41}\ T'_1\ D_6$

$0^{R}42 = R'_{43}\ \underline{M_{SL}}\ T'_1\ D_6$

$1^{R}41 = R_{42}\ \underline{M_{SL}}\ D_6\ D'_o$

$0^{R}41 = R'_{42}\ \underline{M_{SL}}\ D_6\ D'_o$

$\underline{M_{SL}} = D'_4\ D_3\ I_{3a}\ C'_4$

---

with the MSB copied into the LSB of A-register via:

$1^{S} = I_{3a}\ C_4\ T_{41}$ (Pre "SET" "S" to one)

$0^{S} = R'_{40}\ C_4\ I_{3a}\ T_1$ (Reset if MSB is zero)

The MSB is copied from R_{40} into "S" at, effectively, T_1.

It is then copied by A_{41} with the standard recirculation logic.

$1^{A}41 = S\ \underline{M_{SL}}\ T'_1\ D'_o$

$0^{A}41 = S'\ \underline{M_{SL}}\ D'_o$

And the trailing zeros are inserted into R_{42} at T_1

$0^{R}42 = I_{3a}\ \underline{D_{52}}\ T_1$

An OVERFLOW is detected for LSV by:

$1^{J}o = A_1\ \underline{T_{40}}\ D'_5\ \underline{M_{SL}}\ C_5$

A NORMALIZED A-REGISTER WILL TERMINATE THE COMMAND FOR LSC.

$1^{K}c = S\ \underline{T_{40}}\ D'_5\ \underline{M_{SL}}\ C_6$

(Otherwise termination will occur when the prescribed degree of shift has been achieved.)

* The INDEX REGISTER increase has been described for ASC. The exact logic is applicable for LSC.

And the correct sign will be forced into A_{41} at the last bit-time of commands execution.

$$1^{A}41 = C_2\ I_{3a}\ \underline{K_{41}}$$

$$0^{A}41 = C_2'\ I_{3a}\ \underline{K_{41}}$$

$$R_1\ C_4\ D_1'$$

$$R_1'\ C_4\ D_1'$$

$$1^{C}4 = I_{3a}\ D_5'\ D_4'\ \underline{T_{40}}$$

$$0^{C}4 = T_1\ \underline{NR_3}\ D_2$$

R_{41} copies the correct sign at T_{41}. Sync is likewise recirculated at T_1 while C_4 is true.

The A-and R-registers will continue LEFT SHIFTING in this manner until TERMINATION IS INDICATED.

SHIFT LOGIC TERMINATION

The number of bit positions that we shift right or left will coincide with the number of word-times that we spend in the EXECUTE MODE - Phase 2.

The CHARACTER COUNTER will show an increase of one (1), in its readout value, each word-time of the EXECUTE MODE.

The readout of the CHARACTER COUNTER configuration and the C_1 flip-flop will tell us when we should terminate shifting operations (with the exceptions for ASC and LSC already covered.)

The number of bits shifted can be from zero (0) through 127. This information is placed in the SECTOR NUMBER and the C_1 flip-flop. The COMPLEMENT of the Sector Number is placed in the CHARACTER COUNTER during OPERAND MODE - Phase 2.

The number of additions necessary to achieve all ones ($1^{\underline{s}}$) in the CHARACTER COUNTER is EQUAL TO the original, UNCOMPLEMENTED, SECTOR NUMBER.

Since the CHARACTER COUNTER IS only six (6) bits, the total shift available from it alone are zero (0) through 63. If a shift in excess of 63 bits is desired we would "set" C_1.

C_1 is then "reset" the first time the CHARACTER COUNTER reaches its maximum configuration. Sixty-four word times later we would again have this maximum configuration WITH C_1 = FALSE. We would then terminate the command.

The original state of C_1 is determined by the amount of shift specified.

$$\left.\begin{aligned} {}^{1}C_1 &= C_2 \underline{N_7} \\ {}^{0}C_1 &= C_2' \underline{N_7} \end{aligned}\right\} \quad \text{Original shift-in of information}$$

The CHARACTER COUNTER increases its magnitude.

$${}^{1}G_{41} = G_1 K_g' T_{41}' \longrightarrow \text{Recirculation}$$

$$+ G_1' K_g I_1' D_o' \longrightarrow \text{Complement - Add}$$

$${}^{0}G_{41} = G_1' K_g' T_{41}' \longrightarrow \text{Recirculation}$$

$$+ G_1 K_g I_1' D_o' \longrightarrow \text{Complement - Add}$$

Where:

$${}^{1}K_g = G_o I_4' \underline{T_{14}} P_6' P_1 + G_o' I_4' \underline{T_{14}} P_6 P_1$$

$${}^{0}K_g = G_1' K_g I_1' D_o' + \underline{T_{20}}$$

If C_1 were originally TRUE: -

$${}^{0}C_1 = G_1 K_g I_{3a} \underline{T_{20}}$$

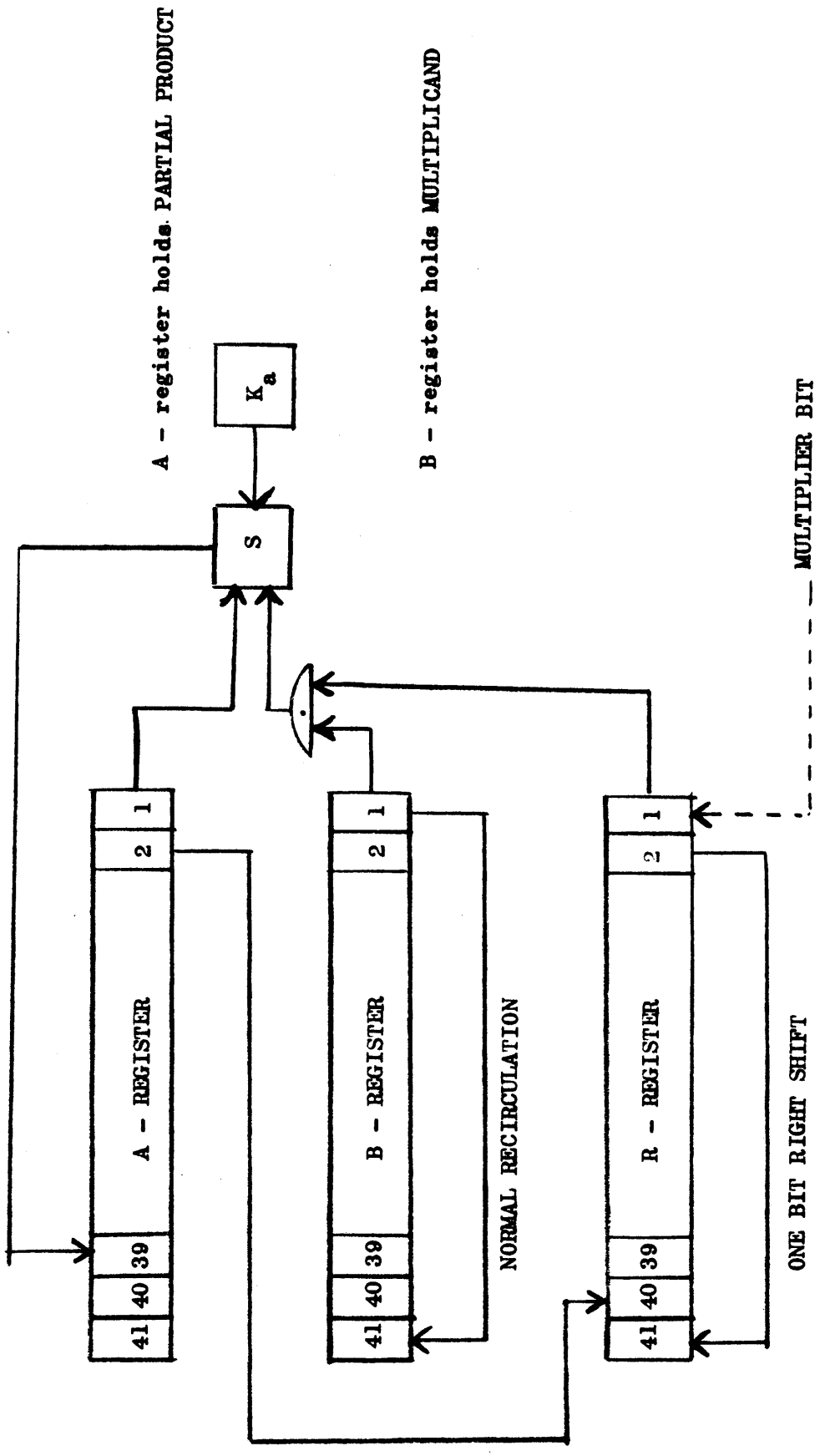
FINAL TERMINATION FOR ALL SHIFT COMMANDS IS TO

$${}^{1}K_c = G_1 K_g F_a' I_{3a} \underline{T_{20}} C_1'$$

Transition is to Phase 1 for one (1) bit-time when we will insert the sign bit, and then

$${}^{1}I_1 = I_1' \underline{K_{41}}$$

MULTIPLY PICTORIAL DIAGRAM



MULTIPLY - MPY - #63

The MPY command causes the Operand, specified by the Channel and Sector portion of the command, to be MULTIPLIED by the contents of the A-register. The result will be a seventy-eight (78) bit product. The thirty-nine (39) most significant bits of the product will appear in the A-register. The thirty-nine least significant bits, of the product, will appear in the R-register. The correct sign will appear in the sign position in both A-and R-registers.

General

Multiplication of binary numbers follows the same pattern as multiplication of decimal numbers demonstrated in the examples of Figure 1 (a) and (b). Starting at the least significant end, digits of the multiplicand are successively multiplied by the least significant digit of the multiplier, and the result recorded as in Figure 1. The process is repeated using each digit of the multiplier in turn, the result at each step being shifted left corresponding to the position of the multiplier digit. When all multiplier digits have been exhausted, the results are added together to obtain final product.

Ignoring signs for the moment, let us now consider the operation in terms of the computer. The addition process that yields the final product would appear to require an adder having 39 inputs, not to mention temporary storage for 39 words. This problem is overcome by performing the addition as the multiplication proceeds, obtaining a new "partial product" after each multiplier bit is used up. In (c) of Figure 1 for example, the least significant multiplier bit (1) times the multiplicand (1010) yields the first partial product (1010).

The second least significant multiplier bit (0) times the multiplicand (1010) yields the quantity 0000 which is shifted one bit left. We then add to the first partial product to obtain the second partial product (01010). In general, the n^{th} least significant multiplier bit times the multiplicand is shifted and added to the $(n-1)^{th}$ partial product. The sum is the n^{th} partial product. When finally n becomes equal

to the number of bits in the multiplier, the n^{th} partial product is the final product and the multiplication process is completed.

In the example shown, the multiplier (1101) has four bits; hence the fourth partial product is the final product.

(a)	(b)	(c)
Decimal	Binary	Binary with Partial Products
10	1010	1010 Multiplicand (B-register)
13	1101	1101 Multiplier (R-register)
30	1010	01010 First P.P. (A-register)
10	0000	00000
130	1010	001010 Second P.P.
	1010	01010
	10000010	0110010 Third P.P.
		01010
		10000010 Fourth P.P. = Final Product
		Most Significant (A-reg.) | Less Significant (R-reg.)

Multiplication Example (Assuming a 4 bit register)

Figure 1

Several things may be observed from the example above. A "1" or "0" multiplier bit times the multiplicand yields the multiplicand itself or a string of 0's respectively. This means that a PARTIAL PRODUCT IS FORMED BY EITHER ADDING OR NOT ADDING THE SHIFTED MULTIPLICAND TO THE PREVIOUS PARTIAL PRODUCT.

In RECOMP this addition is performed by the same "adder" that is used for the ADD and SUB commands. The option of adding or not adding the B-register to the A-register is provided by flip-flop R1. R1 gates B1 into the adder and must therefore contain the applicable multiplier bit. THIS IS ACCOMPLISHED BY INITIALLY PLACING THE MULTIPLIER IN THE R-REGISTER AND SUBSEQUENTLY SHIFTING IT ONE BIT TO THE RIGHT BEFORE PERFORMING EACH ADDITION.

Note also that the length of the partial product starts at one word and increases until it becomes two words long as the final product. This appears to indicate another arithmetic register is required. Once a multiplier bit has performed its function it is no longer needed, and may therefore be discarded to make room for an additional partial product bit.

The next least significant partial product bit is then right shifted out of A and into R, and a multiplier bit is discarded from R. Finally the final double-length product occupies the entire A-and R- registers.

$$\text{MPY - \#63 - } D_6\ D_5\ D_4'\ D_3'\ D_2\ D_1$$

MPY is a Type #3 command. The operand mode is terminated after the operand (multiplier) is read into the B-register during Phase 3.

OPERAND MODE PHASES REVIEW

Phase 1 - Sense Address.

Load D & C with UNINDEXED configuration

Phase 2 - Load C-register and Character counter with INDEXED configuration

- Reload D-register

Search for & READ sector agreement.

Phase 3 - Gate OPERAND into the B-register.

Transition from OPERAND - Phase 3 is to the DIRECTION MODE - Phase 2

$$0^{I_1} = \underline{N_{11}}$$

$$0^{D_o} = I_1\ D_o\ T_{41}$$

$$1^{I_{3a}} = \underline{N_{11}}\ D_2$$

$$\underline{N_{11}} = I_2\ I_1\ D_o\ T_{40}$$

During Phase 2 there are several functions that must be performed.

1. The contents of the A-register are placed in the R-register.
2. The A-register is cleared to zero.
3. The B-register must be recirculated.
4. The algebraic sign of the PRODUCT is determined and stored.

$$C(A) \longrightarrow R$$

(1)

$$1^{R_{41}} = A_2\ \underline{M_1}\ I_2$$

$$0^{R_{41}} = A_2'\ \underline{M_1}\ I_2\ T_{41}'$$

$$1^{R_{40}} = R_{41}\ D_o'\ T_{41}' +$$

$$0^{R_{40}} = R_{41}'\ D_o'\ T_{41}' +$$

$$\underline{M_1} = I_{3a}\ D_5\ D_4'\ D_3'$$

R_{41} copies A_2 forcing the L.S.B. of the A-register to be entered into R_1 at the end of the word time.

(2) $0^{A}41 = I'_4\ T_1\ F'_a$ ← Zero the A-register

$1^{A}40 = A_{41}\ I_2$

$0^{A}40 = A'_{41}\ I_2$

---

(3) $1^{B}41 = B_1\ I_{3a}\ D_2$

$0^{B}41 = B'_1\ I_{3a}\ D_2$ → Recirculate B

---

(4) Sense "sign" of the PRODUCT

$1^{C}2 = U_1\ T_1\ D_2$ ← (Assume sign to be POSITIVE)

$0^{C}2 = U_1\ D_5\ D'_4\ D_2\ A_{40}\ B'_{40}\ T_2 + U_1\ D_5\ D'_4\ D_2\ A'_1\ B_1\ T_{41}$ → (Reset C_2 if signs are opposite. Opposite signs give Negative Product)

Transition is now made to EXECUTE MODE - Phase 2

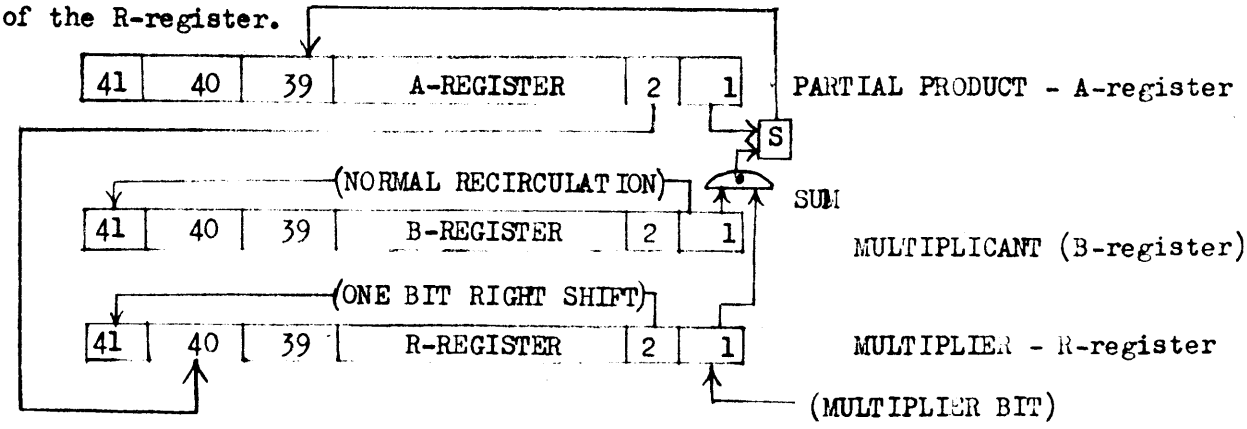
$0^{I}2 = NR_3\ D'_4\ K_o$

$NR_3 = I'_4\ I'_1$

$K_o = K'_o\ T_{41}$

We will remain in Phase 2 for thirty-nine (39 WORD TIMES) while the actual MULTIPLICATION is taking place.

During each word-time of Phase 2 we will either Add the MULTIPLICAND to the previous partial product and shift the L.S.B. into the M.S.B. of the R-register, IF THE MULTIPLIER WAS A ONE (1); or we will simply shift (WITH NO ADDITION) the L.S.B. into the M.S.B. of the R-register.



In practice the Multiplication Sequence proceeds: -

(1) The B-register (MULTIPLICAND) recirculates normally for the entire 39 word-times.

(2) "S" flip-flop (and accompanying logic) is used in the same manner as during the "ADD or SUBTRACT" commands.

(3) R_1, which holds the MULTIPLIER-BIT, is used to activate or inhibit the "adder". If the multiplier bit is a one (1) we add "B" to "A". If the multiplier bit is a zero (0) the "adder" is inhibited and we effectively ADD "ZEROS" to the A-register. A_{39} receives the "SUM (Partial Product) information from the S-flip-flop.

(4) When the A-register receives the "partial product" (Sum) from the S-flip-flop, it is via A_{39}. This has the effect of shifting the "Partial Product" ONE BIT TO THE RIGHT each word-time. The least significant bit of the partial product is shifted out of A_2 and into R_{40}.

(5) Each word-time the R-register recirculates R_2 to R_{41}. This causes the R-register to shift one bit to the right each word time. The M.S.B. is constantly supplied by A_2 at T_{41}. The L.S.B. is shifted into R_1 at T_{41}, to be used as the multiplier bit during the next word time.

(6) Termination is detected by the Character Counter (Left or Right Character counter depending on the command, Left or Right, being executed.)

The "complement" of 39 is forced into the appropriate counter. The Character counter then counts UP until its configuration is all binary ones. (This will take 39 word times). This configuration is detected and termination occurs.

(1a) B-register recirculation: -

$$1^{B}41 = B_1\ I_{3a}\ D_2$$

$$0^{B}41 = B_1'\ I_{3a}\ D_2$$

(2a) S flip-flop addition logic

Copy Logic = R_1'

$$1^{S} = \underline{R_{oo}}\ K_a\ I_{3a} + A_1\ B_1'\ K_a'\ I_{3a} + B_1\ \underline{R_{01}}\ K_a'\ I_{3a}$$

$$+ \underline{S_{oo}}\ I_{3a}\ K_a + \underline{R_{10}}\ I_{3a} + \underline{S_{11}}\ K_a$$

$$0^{S} = \underline{R_{10}}\ K_a\ I_{3a} + A_1\ B_1'\ K_a\ I_{3a} + B_1\ R_{01}\ K_a\ I_{3a}$$

Copy Logic = R_1'

$$\underline{R_{oo}}\ K_a'\ I_{3a} + \underline{S_{oo}}\ I_{3a}\ K_a + \underline{S_{11}}\ K_a'\ T_{41}'$$

$$\underline{R_{01}} = A_1'\ R_1$$

$$\underline{S_{oo}} = A_1'\ B_1'\ C_4'$$

$$\underline{R_{10}} = A_1\ R_1'$$

$$\underline{R_{oo}} = A_1'\ R_1'\ C_4'$$

WHERE: -

$$1^{K}a = \underline{S_{11}}\ D_1\ T_{41}'\ D_2$$

$$0^{K}a = I_{3a}\ T_{41} + \underline{S_{oo}}\ K_a\ D_1\ I_{3a}$$

$$+ \underline{R_{oo}}\ D_1\ I_{3a}\ K_a$$

$$\underline{S_{11}} = A_1\ B_1\ R_1\ I_{3a}$$

K_a remains PRIME WHEN R_1 is PRIME.

(3a) R_1 copies R_2 at T_{41} only

$$1^{R}1 = R_2\ \underline{M_1}\ T_{41} \longrightarrow \text{(ON at } T_1 \text{ time)}$$

$$0^{R}1 = I_{3a}\ D_5\ \underline{T_{40}}\ D_6 \qquad \text{(OFF at } T_{41} \text{ time)}$$

The L.S.B. is in R_2 at T_{41} each word-time. If it is a ONE (1) we can "set" R_1. If it is a zero (0) we will leave (or "reset") R_1 prime.

(4a) A_{39} receives the "Partial Product".

$$1^{A}39 = S\ \underline{M_{SR}}\ T_1'$$

$$0^{A}39 = S'\ \underline{M_{SR}}\ T_1'$$

$$\underline{M_{SR}} = D_4'\ D_3'\ D_1\ I_{3a}\ I_2'$$

While A_{40} monitors the carry F.F. K_a

$$1^{A}40 = K_a \underline{M_{SR}} F'_a$$

$$0^{A}40 = K'_a \underline{M_{SR}} F'_a$$

Note: - The Addition of bits A_{40} and B_{40} read back into A_{39} at T_{41}. A_{40} copies the final (for that word-time) CARRY indication from K_a at T_{41} time).

Since the CARRY DOES BELONG to the Partial Product, as the M.S.B., it is affixed to the A-register magnitude as it shifts toward the "adder".

$$1^{A}39 = A_{40} T_1$$

$$0^{A}39 = A'_{40} T_1$$

(Note: - At T_1, S contains irrelevant information)

(5a) R-register recirculation with a ONE-BIT RIGHT SHIFT is from R_2 to R_{41}

$$1^{R}41 = R_2 \underline{M_{SD}} I'_2 T'_{41}$$

$$0^{R}41 = R'_2 \underline{M_{SD}} I'_2 T'_{41}$$

$$\underline{M_{SD}} = D_6 D'_4 D'_3 D_1 I_{3a}$$

At the end, T_{41}, of the first word-time of Phase 2, the A_{40} bit would be going into A_{39}.

The L.S.B. of the Partial-Product would be reading out of A_2 and going into the now vacated R_{40}

$$1^{R}40 = A_2 \underline{M_{SD}} T_{41}$$

$$0^{R}40 = A'_2 \underline{M_{SD}} T_{41} I'_2$$

The L.S.B. of the R-register (originally R_2) is reading out of R_2 at T_{41}. We place it in R_1 to be used as the new MULTIPLIER-BIT.

$$1^{R}1 = R_2 \underline{M_1} T_{41}$$

$$0^{R}1 = I_{3a} D_5 \underline{T_{40}} D_6$$

(6a) The complement ($1^{\underline{S}}$) of thirty-nine (39) was forced into the appropriate CHARACTER COUNTER during OPERAND MODE - Phase 3

$$1^{G}41 = D_o\ K_g\ \underline{T_{14}}\ P'_6 = T_{18}$$

$$0^{G}41 = D_o\ K_g\ P_5\ P_4\ P'_3\ P_2 = T_{15}$$

$$+ D_o\ K_g\ \underline{T_{20}}\ D_6 = T_{20}$$

$$1^{K}g = G_o\ I'_4\ \underline{T_{14}}\ \overbrace{P'_6\ P_1}^{T_{14}} + G'_o\ I'_4\ \underline{T_{14}}\ \overbrace{P_6\ P_1}^{\overline{T}_{34}}$$

$$0^{K}g = T_{20} + G'_1\ K_g\ I'_1\ D'_o$$

The Character counter then counts UP via

$$1^{G}41 = G_1\ K'_g\ T'_{41} + G'_1\ K_g\ I'_1\ D'_o$$

$$0^{G}41 = G'_1\ K'_g\ T'_{41} + G_1\ K_g\ I'_1\ D'_o$$

Where $0^{K}g = G'_1\ K_g\ I'_1\ D'_o$

TERMINATION OCCURS WHEN

$$1^{K}c = G_1\ K_g\ F'_a\ I_{3a}\ \underline{T_{20}}\ C'_1$$

$$0^{C}1 = \underline{M_1}\ T_1$$

Where K_g = Reset = $G'_1\ K_g\ I'_1\ D'_o$

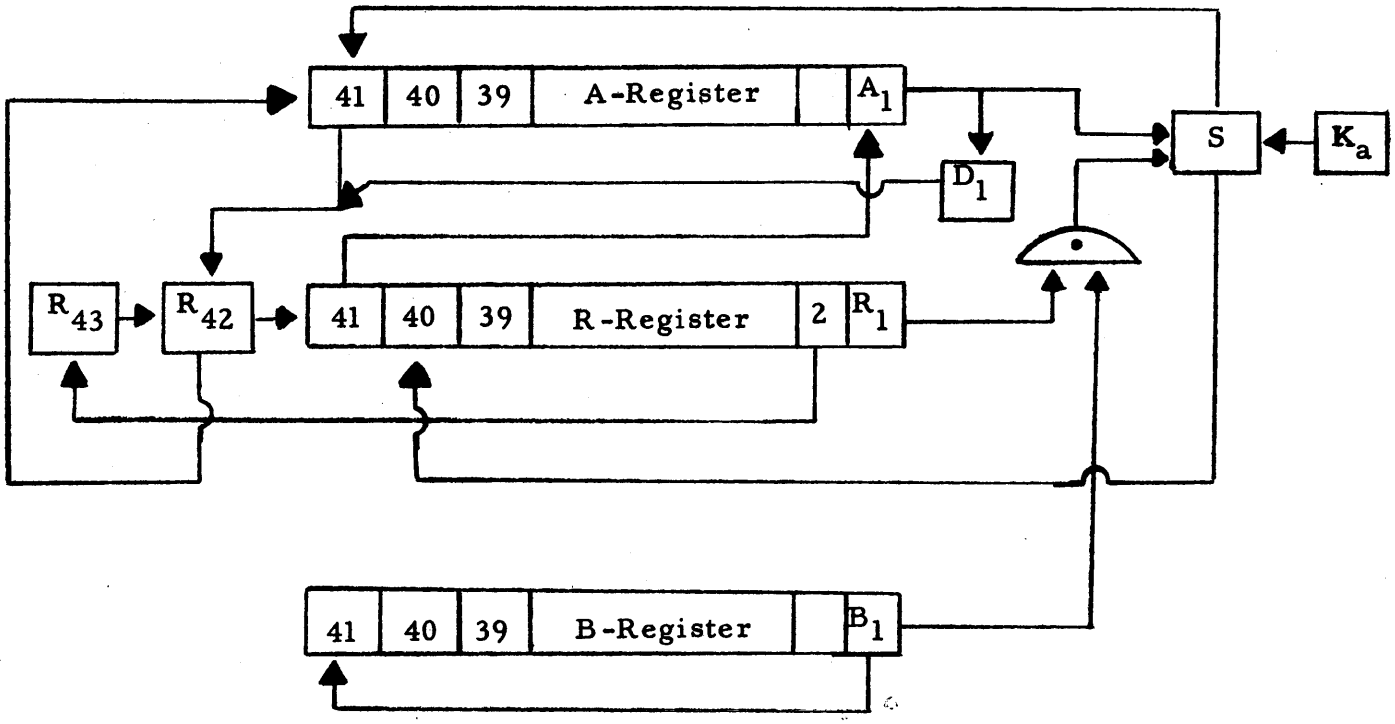
NOTES: A zero in G_1 would $0^{K}g$ before $\underline{T_{20}}$ causing

K'_g at T_{20}. If G_{20} is not a "1" at T_{20}

then G'_1 = True

$$1^{I}1 = I'_1\ \underline{K_{41}}$$

DIVIDE #66



LOGIC

Divide Command (DIV)

General

Division in binary, as in decimal, may be performed by successive subtraction of the divisor, properly positioned, from the full and reduced dividends.

LINE 1		147	Full dividend
2	-	11	Divisor
3		037	No overborrow, ∴ q1 = 1
4	+	11	
5		927	Overborrow, ∴ q1 = 1
6	+	11	
7		037	Restore reduced dividend
8	-	11	Shift divisor
9		026	No overborrow, ∴ q2 = 1
10	-	11	
11		015	No overborrow, ∴ q2 = 2
12	-	11	
13		004	No overborrow, ∴ Q2 = 3
14	-	11	
15		993	Overborrow, ∴ q2 = 3
16	+	11	
17		004	Restore Remainder

DECIMAL EXAMPLE

FIGURE A

Quotient = 13, Remainder = 4

In the decimal example, Figure A, the two digit divisor, (11), is first subtracted from the two most significant dividend digits (14), in order to obtain the most significant (tens) digit of the quotient. This quotient digit, is equal to the number of subtractions possible before an overborrow occurs, When an overborrow occurs, (Line #5), the reduced dividend, is restored by adding the divisor, (Line #6), back in again. The divisor is then shifted one digit right and the same process is repeated to obtain the next most significant digit, This will be the least sigificant digit. The entire quotient, 13, has now been found 004, is the remainder. Note that 9 subtractions is the maximum required to obtain a given quotient digit.

The same operation is performed in binary as illustrated in Figure B. Quotient bits obtained in descending order of significance are 0 or 1 according to whether subtraction of the divisor did or did not cause an overborrow. If overborrow occurs, restoration of the reduced dividend is again achieved by adding back the divisor. Note that a single subtraction is required to determine a given quotient bit in binary as against 9 in decimal. This method requires as many subtractions as there are ONE (1) quotient bits and as many additions as there are zero (0) quotient bits.

An improvement of the previous method yields the so-called Von Neumann division that is closest to that used in RECOMP III and illustrated in Figure C.

In the previous method an overborrow requires addition to restore the reduced dividend, followed by subtraction of the right shifted divisor. Clearly, this two-step operation is equivalent to a single addition of the shifted divisor (compare line #7 of Figure B with Line #5 of figure 6).

In the Von Neumann process, this technique is used to obtain each quotient bit as a result of a single addition or subtraction. This will depend upon the operation performed and the existence of an overborrow or overcarry.

RECOMP III actually uses a modified format of the Illustrated Von Neumann process. This modified method is shown in the following table. Four distinct cases must be considered and are summarized in the table following.

LINE 1	−10010011	Full dividend	
2	10110000	Divisor	
3	11100011	Overborrow, ∴ q1 = 0	
4	+10110000	Add	
5	10010011	Restore dividend	
6	−01011000	Shift divisor	FIGURE B
7	00111011	No overborrow, ∴ q2 = 1	
8	−00101100	Shift divisor	
9	11111001	Overborrow, ∴ q4 = 0	
10	+00010110		
11	00001111	Restore reduced dividend	
12	−00001011	Shift divisor	
13	00000100	No overborrow, ∴ q5 = 1	

Quotient = 1101, Remainder = 100

Binary - Restore

LINE 1	−10010011	Full Dividend	
2	10110000	Divisor	
3	11100011	Overborrow, ∴ q1 = 0 and add	
4	+01011000	Shift divisor	
5	00111011	Overcarry, ∴ q2 = 1 and subt.	FIGURE C
6	−00101100	Shift divisor	
7	00001111	No overborrow, ∴ q3 = 1 and subt.	
8	−00010110	Shift divisor	
9	11111001	Overborrow, ∴ q4 = 0 and add	
10	+00001011	Shift divisor	
11	00000100	Overcarry, ∴ q5 = 1	

Quotient = 1101, Remainder = 100

Von Neumann Method - Non Restore

Rules for Modified Von Neumann Division as performed in RECOMP III

Present Operation	Previous Carry (or) Borrow (A_1)	Present Carry (or) Borrow (K_a)	Next Operation	Quotient Bit Is
SUBTRACT	NO	YES	ADD	0
SUBTRACT	YES	N/A	SUB	1
ADDITION	NO	N/A	ADD	0
ADDITION	YES	YES	SUB	1

In order to perform this division in RECOMP, several things must be considered. We must provide for a possible double length dividend at the start. The A-and R-registers fulfill this requirement. Also, the computer must be capable of performing both addition and subtraction of the divisor in B upon the full or reduced dividend in A. The adder-subtractor logic previously described is used, WITH FLIP-FLOP D1 TRUE FOR ADDITION AND FALSE FOR SUBTRACTION. Note from the example, of Figure C, that the divisor is shifted right for each operation. In the computer it is more convenient to maintain the original orientation of the divisor by recirculating B and then left-shift the reduced dividend in A and R instead. As each new quotient bit is determined, it is written in the lowest order bit position of the R-register. Simultaneously, another dividend bit (if double length) is transferred from R to A while the leading A-register bit no longer needed, is dropped.

Divide Command (Code 66)

The command DIV. causes the contents of the A and R registers, (the 39 most significant bits of the dividend being in A and the 39 lease significant bits of the dividend in R,) TO BE DIVIDED BY THE CONTENTS OF MEMORY LOCATION M. The unrounded quotient appears in A and the remainder in R. The sign of the R register is the same as the original sign of the A register.

DIVIDE = #66 = $D_6\ D_5\ D_4'\ D_3\ D_2\ D_1'$

Divide #66 is a type #3 command. We will remain in the OPERAND MODE until the Operand is gated out of the Memory location and into the B-register.

OPERAND MODE - Phase review

Phase 1 - Sense Address

Load D and C registers with UNINDEXED configuration.

Phase 2 - Load C register and character counter with INDEXED configuration

Reload D-register

Search for and read Sector Agreement.

Phase 3 - Gate operand into the B-register.

The transition from the OPERAND MODE - Phase 3 is to the DIRECTION MODE - Phase 2.

$0^{I}1 = \underline{N_{11}}$ $\quad\quad$ $\lceil \underline{N_{11}} = I_2\ I_1\ D_o\ T_{40}$

$0^{D}o = I_1\ D_o\ T_{41}$

and $\quad 1^{I}3a = N_{11}\ D_2$

During Phase 2 there are several functions that must be performed.

(1) The C(B) is SUBTRACTED from the C(A)

(2) The DIFFERENCE is returned to the A-register via A_{41}. This causes a ONE (1) BIT LEFT SHIFT.

(3) "S" flip-flop is the SUM/DIFFERENCE element.

(3a) R_1 is the GATING flip-flop for addition or subtraction.

(4) The SIGN of the Quotient is sensed and stored.

(5) The B-register is gated into the Subtractor and recirculated normally.

(6) R-register recirculation is via R_{43} copying R_2. This causes a ONE (1) BIT LEFT SHIFT.

(7) Set J_o if A > B.

C(B) is Subtracted from C(A)

(1a) $1^S = \underline{R_{oo}}\ K_a\ I_{3a} + A_1\ B_1'\ K_a\ I_{3a}$

$+B_1\ \underline{R_{o1}}\ K_a'\ I_{3a} + \underline{S_{oo}}\ I_{3a}\ K_a$

$+\underline{R_{10}}\ K_a'\ I_{3a} + \underline{S_{11}}\ K_a$

$0^S = \underline{R_{oo}}\ K_a'\ I_{3a} + A_1\ B_1'\ K_a'\ I_{3a}$

$+B_1\ \underline{R_{o1}}\ K_a\ I_{3a} + \underline{S_{oo}}\ I_{3a}\ K_a'$

$+\underline{R_{10}}\ K_a\ I_{3a} + \underline{S_{11}}\ K_a'\ T_{41}'$

$\underline{R_{oo}} = A_1'\ R_1'\ C_4'$

$\underline{R_{10}} = A_1\ R_1'$

$\underline{R_{01}} = A_1'\ R_1$

$\underline{S_{oo}} = A_1'\ B_1'\ C_4'$

$\underline{S_{11}} = A_1\ B_1\ R_1\ I_{3a}$

Where: $1^{K_a} = B_1\ \underline{R_{o1}}\ I_{3a}\ T_{41}'\ D_5\ D_1'$

$0^{K_a} = I_{3a}\ T_{41} + A_1\ B_1'\ D_1'\ I_{3a}$

$+\ \underline{R_{10}}\ D_1'\ I_{3a}$

(2a) The DIFFERENCE is returned to the A-register via A_{41}

$1^{A_{41}} = S\ T_1'\ D_o'\ \underline{M_{s1}}$

$0^{A_{41}} = S'\ D_o'\ \underline{M_{s1}}$

$\underline{M_{s1}} = D_4'\ D_3\ I_{3a}\ C_4'$

where $0^{C_4} = \underline{NR_3}\ T_1\ D_2$

The ONE-BIT DELAY IN "S" CAUSES A ONE-BIT LEFT SHIFT IN THE A-REGISTER.

(3a) R_1 is the Gating flip-flop for the Subtractor

$1^{R_1} = \underline{M_2}\ D_o'\ T_1 + \underline{M_2}\ D_1\ T_1$

$0^{R_1} = I_{3a}\ T_{40}\ D_6\ D_5$

$\underline{M_2} = I_{3a}\ D_5\ D_4'\ D_3$

R_1 will be TRUE from $T_2 \longrightarrow T_{40}$

(4a) The SIGN of the Quotient is sensed and stored by C_2

$$1^{C}2 = \underline{U_1}\ T_1\ D_2$$

$$0^{C}2 = \underline{U_1}\ D_5\ D'_4\ D_2\ A_{40}\ B'_{40}\ T_2$$

$$+\ \underline{U_1}\ D_5\ D'_4\ D_2\ A'_1\ B_1\ T_{41}$$

(5a) B-register recirculation

$$1^{B}41 = B_1\ I_{3a}\ D_2$$

$$0^{B}41 = B'_1\ I_{3a}\ D_2$$

The B-register will recirculate for the entire command.

(6a) The R-register recirculates via R_2 to R_{43}

$$1^{R}43 = R_2\ I_{3a}\ T'_{41}$$

$$0^{R}43 = R'_2\ I_{3a}$$

$$1^{R}42 = R_{43}\ \underline{M_{s1}}\ T'_{41}\ T'_1\ D_6$$

$$0^{R}42 = R'_{43}\ \underline{M_{s1}}\ T'_1\ D_6$$

$$1^{R}41 = R_{42}\ \underline{M_{s1}}\ D_6\ D'_o$$

$$0^{R}41 = R'_{42}\ \underline{M_{s1}}\ D_6\ D'_o$$

The recirculation of the R-register between R_2 and R_{43} causes a ONE (1) BIT LEFT SHIFT.

Terminate DIRECTION MODE - Phase 2 by

$$0^{I}2 = \underline{NR_3}\ \underline{K_o}\ D'_4$$

ENTER EXECUTE MODE - Phase 2

(7a) Set overflow F.F. if DIVIDEND $>$ DIVISOR at 1st word time of Phase 2

$1^{J}o = \underline{M_2}\ K'_a\ I_2\ T_{41}$ = no overborrow = A $>$ B

To summarize what we have accomplished during the DIRECTION MODE - Phase 2

(1) The B-register was subtracted from the A-register.

(2) The difference was placed into A_{41} from S. This caused a ONE- BIT LEFT SHIFT of the A-register partial quotient. The B-register was recirculated normally.

(3) R_1 was used as a Gating term for the subtractor.

(4) R-register recirculation was from R_2 to R_{43}. This caused a ONE (1) BIT LEFT SHIFT. The M.S.B. is in R_{41} at the end of the word time (left shift causes previous $C(R_{40})$ to be shifted to R_{41}).

(5) The SIGN of the quotient was sensed and stored in C_2.

(6) Terminate Direction Mode - Phase 2 in favor of the EXECUTE MODE - Phase 2.

The EXECUTE MODE - Phase 2 will be maintained for thirty-nine (39) word times. During these 39 word-times the Von Neumann Division process will take place. The functions are: -

(1) Transfer the M.S.B. of the R-register to the L.S.B. position of the A-register.

(2) Continue shift into A_{41} of the SUM/DIFFERENCE operations of S.

(3) "Set" or "Reset" D_1 when a carry or borrow is detected. D_1 will indicate if the next operation is addition or subtraction.

(4) Set R_{42} with the appropriate "Quotient" Bit. The "Quotient" Bit is derived from D_1.

(5) Terminate Phase 2 after 39 word times.

(1a) The M.S.B. of the R-register is transferred to the L.S.B. position of the A-register each word-time. The M.S.B. of the A-register is discarded. The shift of the double - length DIVIDEND (in A and R-registers) has the same effect as shifting the DIVISOR.. (Illustrated in Figure C.)

$$1^{A}1 = R_{41}\ \underline{M_2}\ I'_2\ D'_o\ FM'\ T_1$$

If R_{41} were False a zero (0) would have been placed in A_1 at T_1 via

$0^{A}41 = I'_4\ T_1\ F'_a$ which shifted a zero (0) into A_1 at T_{41}

Since $1^{A}1 = A_2\ D'_o\ T'_1\ I'_2$ A_1 copies A_2 at T'_1

$0^{A}1 = A'_2\ D'_o\ T'_1\ I'_2$ the previously placed zero (0) will remain unless R_{41} is true.

(2a) Continue shift-in to A_{41} of the SUM/DIFFERENCE operations of S.

$$1^{A}41 = \underline{M_{s1}}\ D'_o\ T'_1\ S$$

$$0^{A}41 = \underline{M_{s1}}\ D'_o\ D'_o\ S'$$

(3a) "Set" or "Reset" D_1 for Carry-Borrow TRUE or FALSE respectively.

$$1^{D}1 = A'_1\ K_a\ \underline{M_2}\ T_{41} \qquad \underline{M_2} = I_{3a}\ D_5\ D'_4\ D_3$$

If we are doing Subtraction (D'_1) and had NO OVERBORROW in the previous subtraction; and we have one now; we change to addition. (Set D_1). If we had a previous OVERBORROW we continue subtraction.

If we are doing Addition (D_1) and had no previous OVERCARRY, we continue with addition. IF we had a previous OVERCARRY; and we have one now; we change to subtraction (Reset D_1)

(4a) The correct "QUOTIENT" bit is forced into R_{42} (Consider Von Neumann Table)

$$1^{R}42 = D_1' \underline{M_2} D_6 I_2' T_1 = \text{If Subtraction is next}$$

$$0^{R}42 = D_1 \underline{M_2} I_2' T_1 = \quad \text{If Addition is next}$$

This Quotient bit will be placed in R_2 at the end of the word-time.

It will then be left shifted each word time with each succeeding quotient bit being placed in R_2.

The M.S.B. of R is going to the L.S.B. of A_1.

The M.S.B. of A is discarded.

This process continues for thirty-nine (39) word times.

At the end of thirgy-nine (39) word times the QUOTIENT is in "R" and the REMAINDER is in "A".

Phase 2 is now terminated.

Termination is detected by the CHARACTER COUNTER.

The "complement" of thirty-nine (39) was forced into the appropriate character counter during OPERAND MODE - Phase 3.

$$1^{G}41 = D_o K_g \underline{T_{14}} P_1' D_6 = T_{18} \text{ or } T_{38}$$

$$0^{G}41 = D_o K_g P_5 P_4 P_3' P_2 = T_{15} \text{ or } T_{35}$$

$$+ D_o K_g T_{20} D_6 \qquad = T_{20} \text{ or } T_{40}$$

The Character Counter then counts up until its configuration is all binary ones (1^{S})

$$1^{G}41 = G_1 K_g' T_{41}' + G_1' K_g I_1' D_o'$$

$$0^{G}41 = G_1' K_g' T_{41}' + G_1 K_g I_1' D_o'$$

where:

$$1^{K}g = G_o I_4' \underline{T_{14}} P_6' P_1 + G_o' I_4' \underline{T_{14}} P_6 P_1$$

$$0^{K}g = T_{20} + G_1' K_g I_1' D_o'$$

Phase 2 termination then occurs by

$$0^{C}1 = G_1 K_g I_{3a} \underline{T_{20}}$$

$$1^{D}o = \underline{M_2} C_1' \underline{K_o}$$

Transition is to Phase 3.

NOTE: Check $0^{K}g$ = Logic =

$G_1' K_g I_1' D_o'$ detects a zero (0) in the character counter.

K_g = No zeros

G_1 = No zeros

T_{20} = End of character counter.

During Phase 3

At T_1 of Phase 3 we have

(1) The QUOTIENT bits in the R-register

(2) The LEAST SIGNIFICANT QUOTIENT BIT in R_{42}

(3) The REMAINDER is $A_3 \longrightarrow A_{41}$

(4) The SIGN in C_2

During Phase 3 we must

(1) Gate the Quotient into the A-register

(2) Place the correct REMAINDER in the R-register.

(3) Place correct SIGN in A_{41} and R_{41}

(4) Terminate after one-word time

(1a) Gating of the QUOTIENT into the A-register

At T_1, R_{42} holds the least significant Quotient bit.

$$1^{R}42 = D_1' D_6 \underline{M_2} I_2' T_1 \qquad 0^{R}42 = D_1 \underline{M_2} I_2' T_1$$

at the same time R_{43} is receiving the 2nd LEAST SIGNIFICANT Quotient bit.

$$1^{R}43 = R_2 I_{32} T_{41}' \qquad 0^{R}43 = R_2' I_{3a}$$

AND

A_{41} copies R_{42}

$$1^{A}41 = R_{42}\ \underline{M_2}\ D_o\ T'_{41}$$

$$0^{A}41 = R'_{42}\ \underline{M_2}\ D_o\ T'_{41}$$

Because R_{43} starts its copy of R_2 at T_1 and R_{42} receives its information at T_1, the Quotient shift into the A-register from bits 2 through 40 will get the information into A in the correct positions.

(2a) At T_1 the remainder is in the A-register

If D_1 = TRUE = ADDITION = REMAINDER MUST BE DECOMPLEMENTED

If D_1 = FALSE = SUBTRACTION = REMAINDER is O.K.

For D_1 = TRUE A-and B-registers are Added.

$1^{R}1$ for Adder Logic

$$1^{R}1 = \underline{M_2}\ D_1\ T_1$$

$$0^{R}1 = I_{3a}\ D_5\ D_6\ T_{40}\ \text{(Every word time)}$$

because the A-register is LEFT SHIFTED ONE-BIT we exclude A_2 from the right shift logic into A_1.

$$1^{A}1 = A_3\ \underline{M_2}\ D_o$$

$$0^{A}1 = A'_3\ M_2\ D_o$$

The rest of the register shifts normally.

The output of S is then copied by R_{40}

$$1^{R}40 = S\ M_2\ D_o$$

$$0^{R}40 = S'\ M_2\ D_o$$

The result of the Decomplement (D_1) or the regular copy (D_1') is written into R as the correct remainder.

The signs are forced into A_{41}

$$1^{A}41 = C_2\ I_{3a}\ \underline{K_{41}}$$

$$0^{A}41 = C_2'\ I_{3a}\ \underline{K_{41}}$$

The R-register will have the same sign as the original sign of A.

$$1^{R}42 = A_{41}\ \underline{M_2}\ I_2\ T_1 \text{ (Deposited in } R_2 \text{ at } T_{41})$$

$$0^{R}42 = I_4\ \underline{T_n}$$

The R-register was originally reset.

It is set positive if the sign of A is positive and shifted left until it arrives in R_{41} at T_1 of Phase 3.

Termination occurs

$$1^{K}c = G_1\ K_g\ F_a'\ I_{3a}\ \underline{T_{20}}\ C_1'$$

$$1^{I}1 = I_1'\ \underline{K_{41}}$$

RND - #17.4

The RND command causes the magnitude of the A-register to be increased by ONE (1), IF THE MOST SIGNIFICANT BIT OF THE R-REGISTER IS A ONE (1).

If "OVERFLOW" occurs in the A-register the overflow indicator is turned ON.

If the "most significant bit" of the R-register is zero (0) the command is processed without performing any operation.

RND - #17.4 - $D_6' D_5' D_4 D_3 D_2 D_1 C_6$

---

is a TYPE #1 COMMAND

Review OPERAND MODE for TYPE #1 COMMANDS.

Phase 1: -

1. Gate the C(B) into the Z-register.
2. Gate the UNINDEXED Channel Number and Sector number into the C-register and Character counter; respectively. Load the D-register.
3. Analyze the Address.
4. Terminate after ONE WORD-TIME.

Transition is ALWAYS TO THE DIRECTION MODE - Phase 2

Direction Mode - Phase 2

Entered via: - $0^{I}1 = \underline{I_n} D_5' D_4 T_{41}$

$0^{K}c = I_4' I_1 \underline{K_{41}}$

Phase 2 is maintained for ONE BIT-TIME

$1^{K}c = \underline{U_1} \underline{D_{52}} T_1 F_a'$ $\quad \left[\underline{D_{52}} = D_5' D_2 \right.$

at T_1 we check the MSB of the R-register

$$0^{D}2 = R'_{40}\ \underline{I_3}\ \underline{NR_8}\ T_1$$

$$\begin{cases} \underline{NR_8} = D'_6\ D'_5\ D_3\ D_1\ C_6 \\ \underline{I_3} = I'_4\ I'_1\ D'_0\ I'_{3a} \end{cases}$$

If R_{40} is a zero (0) we "RESET" D_2

I_{3a} is also "Set" at T_1

$$1^{I}3a = U_1\ \underline{D_{52}}\ T_1$$

And C_2 is "Set" at T_1. C_2 will hold the "SIGN" bit.

$$1^{C}2 = U_1\ T_1\ D_2$$

Transition is now to Phase 1.

The A-register is either increased by ONE (1), or it is recirculated:

$$1^{A}41 = A_1\ D'_5\ D'_2\ I'_4\ T'_1 \longrightarrow (\text{Recirculation - } D'_2)$$

$$+A'_1\ I_{3a}\ D'_6\ \underline{D_{52}}\ D_3\ D_1\ T'_{41} \longrightarrow (\text{Complement-Add})$$

$$0^{A}41 = A'_1\ D'_2\ F'_a\ I'_4 \longrightarrow (\text{Recirculation - } D'_2)$$

$$+A_1\ I_{3a}\ D'_6\ \underline{D_{52}}\ D_3\ D_1\ T'_{41} \longrightarrow (\text{Complement-Add})$$

The "COMPLEMENT-ADDITION" Logic will become FALSE when a ZERO (0) is read out from A_1

$$0^{D}2 = A'_1\ I_{3a}\ \underline{NR_8}$$

and the remaining portion of the A-register is recirculated.

The "SIGN" is placed in C_2 via:

$$1^{C_2} = \underline{U_1}\ T_1\ D_2$$

$$0^{C_2} = A'_{40}\ \underline{U_1}\ T_2\ D_6\ \underline{D_{52}}\ D_1$$

And copied by A_{41} at T_{41}

$$1^{A_{41}} = C_2\ I_{3a}\ \underline{K_{41}}$$

$$0^{A_{41}} = C'_2\ I_{3a}\ \underline{K_{41}}$$

If the A-register should "overflow" we would still have D_2 at T_{41} the A-register would have to be ALL ONES (1) we would turn on: -

$$1^{J_o} = D_2\ I_{3a}\ \underline{NR_8}\ T_{41}$$

And terminate the command via:

$$1^{I_1} = I'_1\ \underline{K_{41}}$$

The CMP- #17 command causes the contents of the A-register to be replaced by its COMPLEMENT, ONES (1^s) COMPLEMENT , INCLUDING THE SIGN.

$$CMP - \#17 - D'_6\ D'_5\ D_4\ D_3\ D_2\ D_1\ C'_6$$

CMP is a TYPE #1 command.

Review OPERAND MODE FOR TYPE #1 COMMANDS

1. Gate C(B) into the Z-register
2. Gate channel and operation code configuration into the C-and D-registers respectively. (UNINDEXED CONFIGURATION)
3. Analyze address
4. Terminate after ONE WORD-TIME. Transition is to the DIRECTION MODE - Phase 2.

Direction Mode - Phase 2.

Entered via: -

$$0^K_c = I'_4\ I_1\ \underline{K_{41}}$$

$$0^I_1 = \underline{I_n}\ D'_5\ D_4\ T_{41}$$

and terminated after ONE BIT-TIME with transition to Phase 1.

$$1^K_c = \underline{NR_3}\ D_4\ T_1\ F'_a$$

Phase 1: -

During Phase 1 the COMPLEMENTING of the A-register, including the SIGN, is accomplished.

$$1^A_{41} = A'_1\ I_{3a}\ D'_6\ \underline{D_{52}}\ D_3\ D_1\ T'_{41}$$

$$0^A_{41} = A_1\ I_{3a}\ D'_6\ \underline{D_{52}}\ D_3\ D_1\ T'_{41}$$

$$\begin{cases} 1^I_{3a} = \underline{U_1}\ \underline{D_{52}}\ T_1 \\ 0^I_{3a} = K_{41} \\ \underline{D_{52}} = D'_5\ D_2 \\ \underline{U_1} = I'_4\ I_2\ I'_1\ D'_0 \end{cases}$$

The COMPLEMENT of the SIGN is placed in A_{41} via: -

$$1^{A}41 = C_2 \; I_{3a} \; \underline{K_{41}}$$

$$0^{A}41 = C_2' \; I_{3a} \; \underline{K_{41}}$$

A_{41} obviously copies C_2 at T_{41}. The correct sign was placed in C_2 by

$$1^{C}2 = U_1 \; T_1 \; D_2 \qquad \text{(Initially "set" position)}$$

$$0^{C}2 = A_{40} \; \underline{T_2} \; I_{3a} \; D_6' \; D_5' \; D_4 \; D_1 \; C_6'$$

(Force C_2 - prime if sign of "A" was positive)

Terminate at the end of the word time -

$$1^{I}1 = I_1' \; \underline{K_{41}}$$

TRANSFER COMMANDS

TRA - #51, TIX - #10, TLB - #15, TZE - #50, TNZ - 11, TOV - #52, TMI - #53, TPL - #55, HTR - #71

TRA - #51: Transfers control of the program to the command specified by the address configuration. Transfer may be to either "left" or "right" hand command. Transfer is UNCONDITIONAL.

HTR - #71: Will cause computer operations to HALT, while placing the specified address configuration in the LOCATION COUNTER. This address configuration is also displayed on the CONTROL CONSOLE NEONS. This command is UNCONDITIONAL.

TIX - #10, TLB - #15, TZE - #50, TNZ - #11, TOV - #52, TMI - #53 and TPL - #55, are all CONDITIONAL TRANSFERS. Transfer will occur only if the specific conditions are met. If conditions are not met the commands will have the same effect as a "No Operation".

TIX - #10: Decrement the INDEX REGISTER by one (1) (actually Add a one (1) to the COMPLEMENTED (INDEX configuration) IF the INDEX-register WAS NOT ZERO or IS NOT NOW ZERO transfer control of the program to the command specified by the address. TIX MUST BE A LEFT-HAND COMMAND.

TLB - #15: If the LEAST SIGNIFICANT BIT of the A-register is a one (1) the computer TRANSFERS control of the program to the command specified by the address.

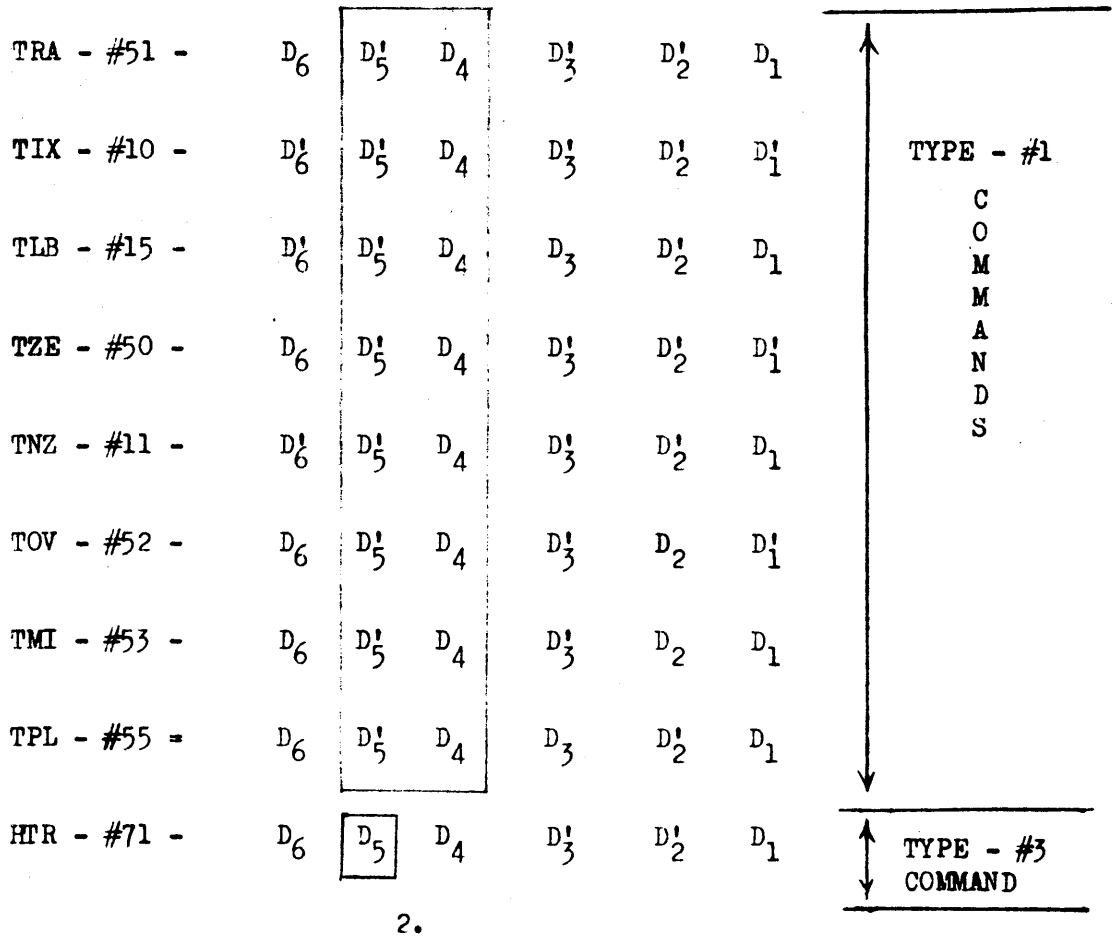
TZE - #50: If the contents of the A-register IS EQUAL TO ZERO (plus or minus) the computer TRANSFERS control of the program to the command specified by the address.

TNZ - #11: If the contents of the A-register IS NOT EQUAL TO ZERO, the computer TRANSFERS control of the program to the command specified by the address.

TOV - #52: If the OVERFLOW Indicator is "ON", TURN IT "OFF". The computer then TRANSFERS control of the program to the command specified by the address.

TMI - #53: If the SIGN of the A-register is NEGATIVE, the computer TRANSFERS control of the program to the command specified by the address.

TPL - #55: If the SIGN of the A-register is POSITIVE, the computer TRANSFERS control of the program to the command specified by the address.



Review of OPERAND MODE for TYPES # 1 and #3 commands.

Phase 1. Gate C(B) into the Z-register (can be INDEXED)
Gate Channel and Operation code configurations into the C and D registers respectively. Analyze address and terminate after ONE-WORD TIME.

Phase 2. Gate INDEXED configuration into the C-register and CHARACTER counter from the Z-register. (Channel and Sector numbers respectively). Reload D-register. Search for and achieve SECTOR AGREEMENT.

Phase 3. Gate C(W) into the B-register. Terminate after ONE-WORD TIME.

DURING OPERAND MODE - Phase 1:

All "Conditional Transfer Commands" are checked for "condition-met" the OPERATION CODE WILL BE CHANGED TO A UNCONDITIONAL TRANSFER #51 - configuration. The command would then be processed as a regular TRA. (TIX - #10 is the single exception).

TZE - #50 - $D_6\ D_5'\ D_4\ D_3'\ D_2'\ D_1'$ - Transfer on ZERO.

The A-register is recirculated during Phase 1

$$1^{A}41 = A_1\ I_1\ T_1' \qquad 0^{A}41 = A_1\ I_1$$

So monitors the A_1 output.

$$0^{S}o = T_{41} \qquad 1^{S}o = A_1\ \underline{I_n}\ K_c\ T_{41}'\ T_1'$$

If S_o' = True at T_{41} = the A-register must have been zeros.

$$1^{D}1 = S_o'\ \underline{I_n}\ T_{41}\ \underline{M_5}\ D_2' \qquad \underline{M_5} = D_6\ D_5'\ D_4\ D_3'$$

TNZ - #11 - $D_6' D_5' D_4 D_3' D_2' D_1$ - Transfer on NOT ZERO

The condition of S_o is detected at T_{41}

S_o = True at T_{41} = A-register must have a one (1) in it.

$$1^{D_6} = S_o \underline{I_n} T_{41} D_5' \underline{D_{42}} D_1 \qquad \left[\underline{D_{42}} = D_4 D_2'\right.$$

TMI - #53 - $D_6 D_5' D_4 D_3' D_2 D_1$ = Transfer on Minus sign

A_1 is monitored at T_{41} (Sign Bit)

$$0^{D_2} = A_1' \underline{I_n} T_{41} \underline{M_5} \qquad \left[\underline{M_5} = D_6 D_5' D_4 D_3'\right.$$

TPL - #55 - $D_6 D_5' D_4 D_3 D_2' D_1$ = Transfer on Plus sign.

A_1 is monitored at T_{41} (Sign Bit)

$$0^{D_3} = A_1 \underline{I_n} T_{41} D_6 D_5' \underline{D_{42}}$$

TOV - #52 - $D_6 D_5' D_4 D_3' D_2 D_1'$ = Transfer on OVERFLOW

If J_o = True

$$1^{D_1} = J_o \underline{M_5} D_2 \underline{I_n} T_{41}$$

$$0^{D_2} = \underline{M_5} D_1' \underline{I_n} T_{41}$$

$$0^{J_o} = \underline{I_n} T_{41} \underline{M_5} D_2 D_1'$$

TLB - #15 - $D_6' D_5' D_4 D_3 D_2' D_1$ = Transfer if L.S.B. = 1

The L.S.B. of the A-register reads out of A_3 at T_{41}

$$0^{D_3} = A_3 \underline{I_n} T_{41} D_6' D_5' \underline{D_{42}}$$

$$1^{D_6} = S_o \underline{I_n} T_{41} D_5' \underline{D_{42}} \qquad [S_o = \text{A-register has a one (1) in it.}]$$

We can now see that if "conditions are met" the Operation Code changes to

TRA - #51 - $D_6\ D_5'\ D_4\ D_3'\ D_2'\ D_1$

TZE - #50 - $D_6\ D_5'\ D_4\ D_3'\ D_2'\ D_1'$

$$1^D1 = S_o'\ \underline{I_n}\ T_{41}\ M_5\ D_2 = \boxed{D_6\ D_5'\ D_3'\ D_2'\ D_1}$$

TNZ - #11 - $D_6'\ D_5'\ D_4\ D_3'\ D_2'\ D_1$

$$1^D6 = S_o\ \underline{I_n}\ T_{41}\ D_5'\ \underline{D_{42}}\ D_1 = \boxed{D_6\ D_5'\ D_3'\ D_2'\ D_1}$$

TMI - #53 - $D_6\ D_5'\ D_4\ D_3'\ D_2\ D_1$

$$0^D2 = A_1'\ \underline{I_n}\ T_{41}\ \underline{M_5} = \boxed{D_6\ D_5'\ D_3'\ D_2'\ D_1}$$

TPL - #55 - $D_6\ D_5'\ D_4\ D_3\ D_2'\ D_1$

$$0^D3 = A_1\ \underline{I_n}\ T_{41}\ D_6\ D_5'\ \underline{D_{42}} = \boxed{D_6\ D_5'\ D_3'\ D_2'\ D_1}$$

TOV - #52 - $D_6\ D_5'\ D_4\ D_3'\ D_2\ D_1'$

$$0^Jo = \underline{I_n}\ T_{41}\ \underline{M_5}\ D_2\ D_1' \qquad 1^D1 = J_o\ \underline{M_5}\ D_2\ \underline{I_n}\ T_{41}$$

$$0^D2 = M_5\ D_1'\ \underline{I_n}\ T_{41}$$

TLB - #15 - $D_6'\ D_5'\ D_4\ D_3\ D_2'\ D_1$

$$0^D3 = A_3\ \underline{I_n}\ T_{41}\ D_6'\ D_5'\ \underline{D_{42}}$$

$$1^D6 = S_o\ \underline{I_n}\ T_{41}\ D_5'\ \underline{D_{42}}\ D_1$$

TRA #51 - TIX # 10 - HTR #71

The TRA #51 command and TIX #10 are not effected during the OPERAND MODE - Phase 1.

Transition for the TRA and TIX (as well as those "Conditional Transfers") move directly to the DIRECTION MODE - Phase 2 via

$$0^{I}1 = \underline{I_n}\ D'_5\ D_4\ T_{41} \qquad\qquad 0^{K}c = I'_4\ I_1\ \underline{K_{41}}$$

The HTR #71 command, being a TYPE #3 command, enters the DIRECTION MODE - Phase 2, from the OPERAND MODE - Phase 3 via: -

$$0^{I}1 = \underline{N_{11}} \qquad 0^{D}o = I_1\ D_o\ T_{41} \qquad \left| \underline{I_{11}} = I_2 I_1 D_o T_{41} \right.$$

DIRECTION MODE - Phase 2

Phase 2 is maintained for ONE BIT-TIME for ALL of the Transfer Commands UNDER ALL CIRCUMSTANCES.

TRA #51 - goes to Phase 4 via

$$1^{K}c = \underline{NR_3}\ D_4\ T_1\ F'_a$$

$$1^{D}o = \underline{U_1}\ \underline{M_7}\ T_1$$

and the Address configuration of the Command is gated from the Z-register into the LOCATION COUNTER

$$\left.\begin{aligned} 1^{G}41 &= Z_1\ \underline{H_o}\ N_5\ G'_o\ P_6 \\ 0^{G}41 &= Z'_1\ \underline{H_o}\ N_5\ G'_o\ P_6 \end{aligned}\right\} \text{L.H. Command gating}$$

$$1^{N}5 = K_c\ D_o\ \underline{T_{21}} + \underline{U_1}\ \underline{M_7}\ T_1\ D_6$$

LEFT RIGHT

$$0^{N}5 = D_o\ \underline{T_{14}}$$

$$\underline{NR_3} = I'_4\ I'_1$$

$$\underline{M_7} = D_4\ D'_3\ D'_2\ D_1$$

$$\underline{U_1} = I'_4\ I_2\ D'_1\ D'_o$$

$$\underline{H_o} = I'_1\ D_o\ I'_{3a}$$

$$\underline{T_{14}} = P_5\ P_4\ P_3$$

The gating of RIGHT HAND COMMAND ADDRESS configurations is accomplished by: -

$$1^{D}6 = Z_1\ G_o\ N_5\ I'_{3a}\ I_2\ I'_4$$

$$0^{D}6 = Z'_1\ G_o\ N_5\ I'_{3a}\ I_2\ I'_4$$

$$\boxed{\underline{N_7} = I'_4\ N_5}$$

$1^{D}5 = D_6\ N_5$ $\quad$ $1^{D}4 = D_5\ N_5$ $\quad$ $1^{D}3 = D_4\ N_5$ $\quad$ $1^{D}2 = D_3\ N_5$ $\quad$ $1^{D}1 = D_2\ N_5$

$0^{D}5 = D'_6\ N_5$ $\quad$ $0^{D}4 = D'_5\ N_5$ $\quad$ $0^{D}3 = D'_4\ N_5$ $\quad$ $0^{D}2 = D'_3\ N_5$ $\quad$ $0^{D}1 = D'_2\ N_5$

$1^{C}6 = D_1\ N_5$ $\quad$ $1^{C}5 = C_6\ N_5$ $\quad$ $1^{C}4 = C_5\ \underline{N_7}$ $\quad$ $1^{C}3 = C_4\ \underline{N_7}$ $\quad$ $1^{C}2 = C_3\ \underline{N_7}$

$0^{C}6 = D'_1\ N_5$ $\quad$ $0^{C}5 = C'_6\ N_5$ $\quad$ $0^{C}4 = C'_5\ \underline{N_7}$ $\quad$ $0^{C}3 = C'_4\ \underline{N_7}$ $\quad$ $0^{C}2 = C'_3\ \underline{N_7}$

$$1^{C}1 = C_2\ \underline{N_7} \qquad 1^{R}43 = C_1\ N_5\ D_o$$

$$0^{C}1 = C'_2\ \underline{N_7} \qquad 0^{R}43 = C'_1\ N_5\ D_o$$

This Address information is held "static" in the D-register, the C-register, and R_{43} until N_5 is again the LOCATION COUNTER.

$$1^{G}41 = R_{43}\ N_5\ G_o\ P_6\ \underline{H_o}$$

$$0^{G}41 = R'_{43}\ N_5\ G_o\ P_6\ \underline{H_o}$$

Phase 4 is maintained until the end of the word-time with transition to the COMMAND MODE - Phase 1

$$0^{I}2 = \underline{H_o}\ I_2\ T_{41}$$

$$1^{I}1 = I'_1\ \underline{K_{41}}$$

$$0^{D}o = G'_1\ N'_1\ \underline{K_{41}}$$

TIX # 10 goes to Phase 1 from Phase 2 at T_1 via -

$$1^{K}c = \underline{NR_3}\ D_4\ T_1\ F'_a$$

where the INDEX REGISTER is increased by one (1). and is monitored for a zero output.

$$1^{G}41 = G_1\ K'_g\ T'_{41} + G'_1\ K_g\ I'_1\ D'_o$$

$$0^{G}41 = G'_1\ K'_g\ T'_{41} + G_1\ K_g\ I'_1\ D'_o$$

$$1^{K}g = \underline{NR_3}\ T_2\ D'_6\ D'_3\ \underline{D_{42}}\ N'_5$$

$$0^{K}g = G'_1\ K_g\ I'_1\ D'_o$$

} Decrement of the INDEX Register.

* If the INDEX REGISTER is not zero (0) or one (1) we will TRANSFER.

** (Since the C(I) is the ones complement zero (0) or one (1) is indicated by binary zero (0) is present we do not have a true zero or one count so we will transfer.)

$$1^{D}o = G'_1\ R_{42}\ \underline{I_3}\ D'_6\ \underline{D_{42}}\ D'_1$$

$$1^{R}42 = \underline{I_3}\ D'_6\ D'_1\ P'_6\ P'_3\ P_2\ P'_1$$

and transition is IMMEDIATELY to Phase 4.

When in Phase 4 the LEFT-HAND ADDRESS will be gated into the LOCATION COUNTER.

$$1^{G}41 = Z_1\ \underline{H_o}\ \underline{N_5}\ G'_o\ D_6$$

$$0^{G}41 = Z'_1\ \underline{H_o}\ \underline{N_5}\ G'_o\ D_6$$

The L.H. address is the only one left available to us by the time we enter Phase 4, the TIX command is restricted to LEFT-HAND Command position only.

Termination is the same as for a TRA - Command

$$0^{I}2 = \underline{H_o}\ I_2\ T_{41}$$

$$1^{I}1 = I'_1\ \underline{K_{41}}$$

$$0^{D}o = G'_1\ N'_1\ \underline{K_{41}}$$

with transition to the COMMAND MODE - Phase 1

The TNZ command is the EXCEPTION to the foregoing rules for "Conditional Transfer" commands: -

Because at T_{41} of $\underline{I_n}$ we: $1^{D}6 = S_o\ \underline{I_n}\ T_{41}\ D'_5\ \underline{D_{42}}\ D_1$

our D-register configuration will be

$$D_6\ D'_5\ D_4\ D'_3\ D'_2\ D_1 = \#51 = \text{NORMAL TRANSFER}$$

But if our "CONDITIONS ARE NOT MET" we would have $D'_6\ D'_5\ D_4\ D'_3\ D'_2\ D_1$

$$\underline{M_7} = \text{True} = D_4\ D'_3\ D'_2\ D_1$$

at T_1 of the DIRECTION MODE

Normally if conditions are not met we have transition into Phase 1 via

$$1^{K}c = \underline{NR_3}\ D_4\ T_1\ F'_a$$

and termination at the end of the word-time.

TNZ, however, forces "D_o" ON even when the "conditions are NOT MET".

$$1^{D}o = \underline{U_1}\ \underline{M_7}\ T_1$$

and transition is to Phase 4 at T_1

During Phase 4, for a "condition met transfer" we would gate from Z-into the LOCATION COUNTER.

This is controlled by N_5.

$$1^{N}5 = \underline{U_1}\ \underline{M_7}\ T_1 D_6 \longleftarrow$$ = Cannot come true because TNZ had D'_6

at T_{14} we $0^{D}o = \underline{H_o}\ N_5'\ \underline{T_{14}}\ D_1\ I_2\ P_1$

which returns ut to Phase 1 where termination will occur at the end of the word time.

HTR - #71 - Enters the DIRECTION MODE - Phase 2 from Phase 3 of the OPERAND MODE. (#71 is a TYPE #3 command.)

$$0^{I}1 = \underline{N_{11}}$$

$$0^{D}o = I_1\ D_o\ T_{41}$$

and goes to Phase 4 at T_1

$$1^{D}o = \underline{U_1}\ \underline{M_7}\ T_1$$

$$1^{K}c = \underline{NR_3}\ D_4\ T_1\ F_a'$$

$$\underline{N_{11}} = D_o\ T_{41}\ I_2\ I_1$$

$$\underline{M_7} = D_4\ D_3'\ D_2'\ D_1$$

$$\underline{H_o} = I_1'\ D_o\ I_{3a}'$$

$$\underline{NR_3} = I_4'\ I_1'$$

$$\underline{U_1} = I_4'\ I_2\ I_1'\ D_o'$$

During Phase 4 we

1. Cause a ONE (1) to be written into the "INHIBIT BIT" position of the G-register (G_{41})

$$1^{N}1 = \underline{H_o}\ S_o\ \underline{T_{40}}\ I_2$$

$$0^{N}1 = \underline{H_o}\ I_2\ T_{41}$$

$$1^{S}o = \underline{H_o}\ D_6\ D_5\ T_2$$

$$0^{S}o = T_{41}$$

and the ONE (1) is written in via

$$1^{G}41 = N_1\ T_{41}\ P_6$$

$$0^{G}41 = I_1'\ N_1'\ T_{41}$$

2. The Standard TRA Logic is all true for the HTR during Phase 4. EXCEPT THE TERMINATION LOGIC.

$$0^{I}2 = \underline{H_o}\ I_2\ T_{41}$$

$$1^{I}1 = I_1' \underline{K_{41}}$$

But K_c and D_o remain "ON".

$$0^{K}c = \underline{U_1}\ F_a'\ \underline{NR_{50}} = \text{NOT TRUE}$$
$$+\ I_4'\ I_1\ \underline{K_{41}} = \text{NOT TRUE}$$

$0^{D}o = G_1'\ N_1'\ \underline{K_{41}} = N_1$ is true at T_{41} for HTR,

Transition is then to the COMMAND MODE - Phase 4, WHICH IS THE "SPECIAL DISPLAY" Phase.

During the DISPLAY PHASE, which is maintained for ONE WORD-TIME, we will gate the Location Counter into the Display-register

D_6	D_5	D_4	D_3	D_2	D_1	C_6	C_5	C_4	C_3	C_2	C_1	R_{43}

MSB LSB.

$$1^{D}6 = G_1\ N_5\ \underline{I_c} \qquad 1^{R}43 = C_1\ N_5\ D_o$$
$$0^{D}6 = G_1'\ N_5\ \underline{I_c} \qquad 0^{R}43 = C_1'\ N_5\ D_o$$

$$1^{N}5 = K_c\ D_o\ \underline{T_{21}}$$
$$0^{N}5 = \underline{T_{14}}$$

Transition is then to IDLE MODE - Phase 2 at the end of the word time.

$$1^{I}4 = I_1\ D_o\ \underline{K_{41}}$$
$$0^{D}o = I_1\ D_o\ T_{41}$$
$$0^{K}c = I_4'\ I_1\ \underline{K_{41}}$$

where the computer halts until the appropriate signals are generated.

NOTE: Display logic when the "COMPUTE" switch is placed in "HALT" position.

HALT POSITION produces signal = C'

$$1^{N}1 = \underline{T_{40}}\ C'\ I_1'$$

The "inhibit bit" is generated by

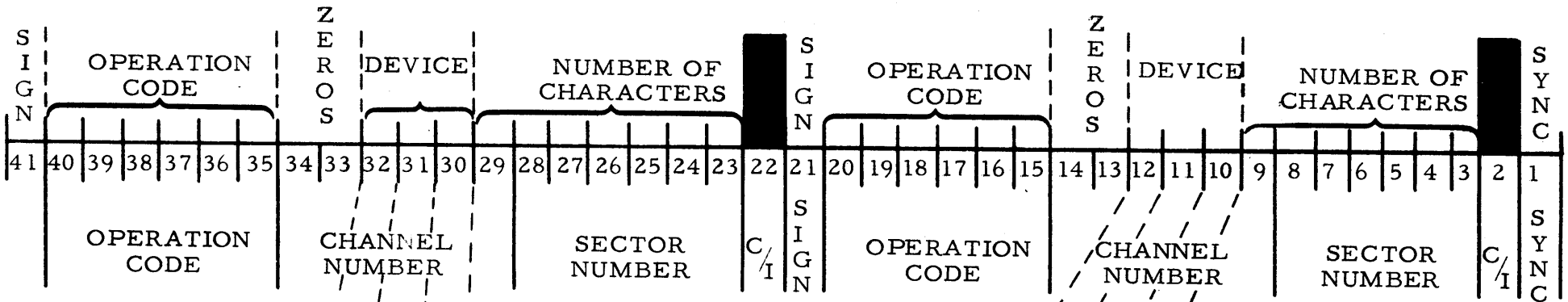
$$1^{G}41 = N_1\ T_{41}\ P_6$$

Transition is now to the Command Mode Phase #4 via

$$0^{I}2 = N_1\ \underline{K_{41}} \qquad 1^{I}1 = I_1'\ \underline{K_{41}} \qquad 1^{D}o = \underline{K_{41}}\ N_1$$

which is the DISPLAY PHASE. The logic now follows the same sequence as the HTR command.

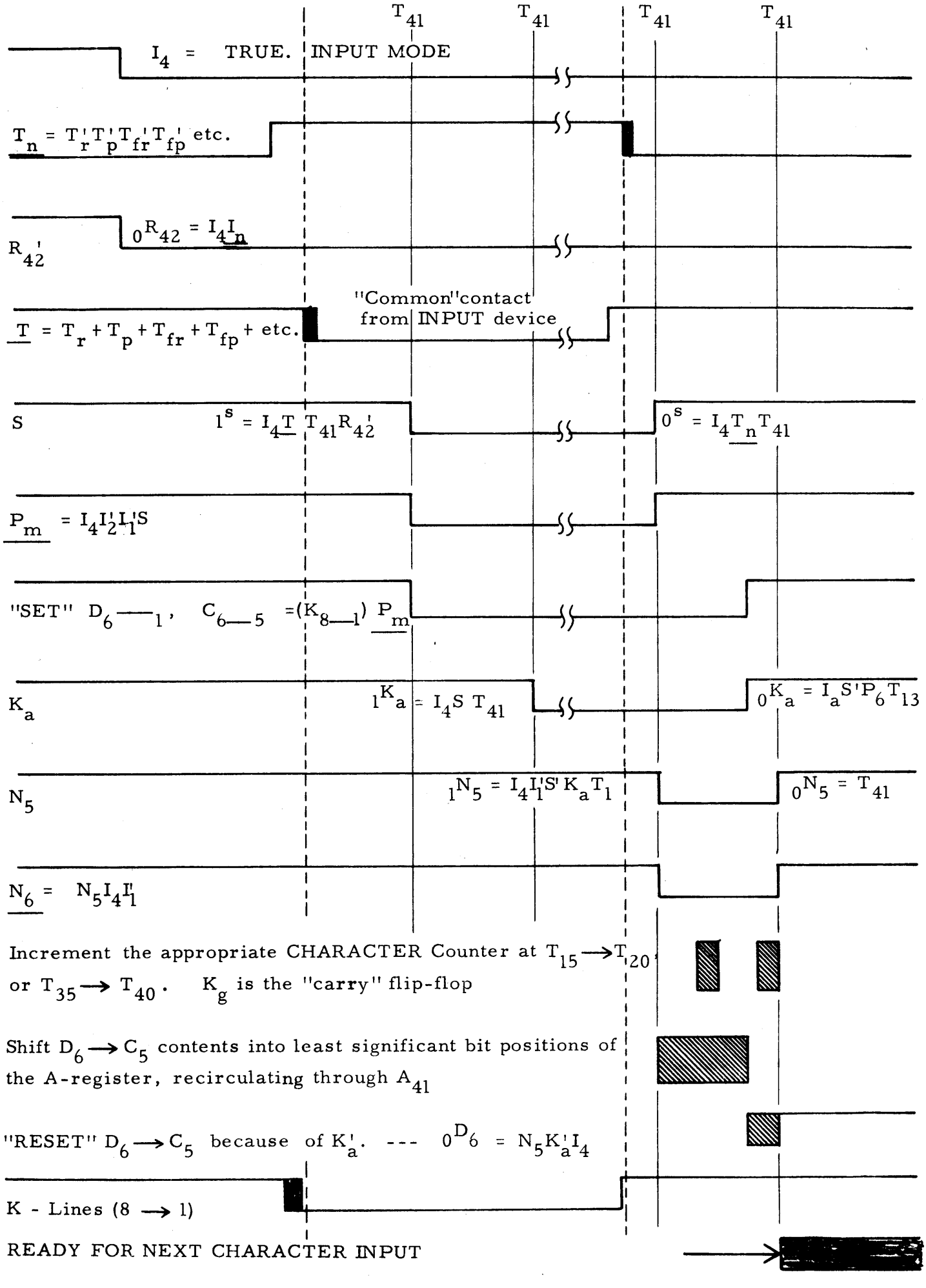
INPUT-OUTPUT WORD FORMAT



0	0	0	KEYBOARD
0	0	1	TAPE PUNCH or READER
0	1	0	FACITAPE UNIT
0	1	1	OPTIONAL
1	0	0	CARD READER

0	0	0	KEYBOARD
0	0	1	TAPE PUNCH or READER
0	1	0	FACITAPE UNIT
0	1	1	OPTIONAL
1	0	0	CARD READER

TIMING SEQUENCE FOR INPUT - ICH #00



INPUT - #00 - ICH

The ICH command causes the selected piece of peripheral equipment to Input, into the A-register, from 0 to 128, eight-bit characters. All but the last five (5) characters are lost.

Since the "Control Console Input Switch" overrides the selection of the Input device by the address in the command; it is necessary that this switch be placed in the "AUTO" position to allow the command to specify the Input device.

ICH - #00 - D_6 D_5 D_4 D_3 D_2 D_1

ICH - #00 is a TYPE #2 command

Review OPERAND MODE for TYPE #2 commands.

Phase 1: -

(1) Gate the C(B) into the Z-register.

(2) Gate the UNINDEXED Channel number and Sector number into the C-register and CHARACTER COUNTER, respectively. Load the D-register with the OPERATION CODE.

(3) Analyze the address.

(4) Terminate after one word-time. Transition is to Phase 2.

Phase 2: -

(1) Reload the C-register and the CHARACTER COUNTER with the INDEXED Channel and Sector configuration. Reload the D-register with the Operation Code.

(2) Terminate after one word-time, with transition to the DIRECTION MODE - Phase 2.

DIRECTION MODE - Phase 2

Entered from the OPERAND MODE via: -

$$O^{I}1 = \underline{I_n}\ D'_5\ \underline{K_o}$$

During the DIRECTION MODE - Phase 2 we will simply INCREMENT the LOCATION COUNTER.

(Refer: - Direction Mode Description)

Phase 2 is maintained for one word-time.

Transition is to the INPUT MODE - Phase 2 via:

$$1^{I}4 = \underline{I_3}\ D'_6\ \underline{D_{53}}\ \underline{K_o}$$

$$O^{I}2 = \underline{NR_3}\ \underline{K_o}\ D'_4$$

$$\left|\begin{array}{l} \underline{I_3} = I'_4\ I'_1\ D'_o\ I'_{3a} \\ \underline{D_{53}} = D'_5\ D'_3 \\ \underline{K_o} = K'_c\ T_{41} \\ \underline{NR_3} = I'_4\ I'_1 \end{array}\right.$$

Upon entering the INPUT MODE - Phase 2 the Device Selection configuration, contained in flip-flops $C_4\ C_3\ C_2$, is immediately checked, and the appropriate PRIMARY SIGNAL Generated.

$\underline{RKS}$ = INPUT FROM FLEXOWRITER KEYBOARD

$$\underline{RKS} = C'_4\ C'_3\ C'_2\ I_4\ I'_2\ I'_1\ \underline{O_r}$$

(O_r = SIGNAL FROM FLEXOWRITER TO COMPUTER SIGNIFIES THAT THE FLEX HAS COMPLETED ITS LAST FUNCTION)

R_d = INPUT FROM FLEXOWRITER READER

$$R_d = C'_4\ C'_3\ C_2\ I_4\ I'_2\ I'_1$$

($\underline{RKS}$ = TURN ON "$\underline{\text{INPUT-LIGHT}}$".)

The setting of the INPUT SWITCH to positions 1 5 produces the following signals

$\#1 = C_4' C_3' C_2'$ $\#2 = C_4' C_3' C_2$

$\#3 = C_4' C_3 C_2'$ $\#4 = C_4' C_3 C_2$

$\#5 = C_4 C_3' C_2'$ AUTO = PROGRAM CONTROL

Positions: - 1, 2, 3, 4	of Section 1 = $I_{40} = 0^C4$
Positions: - 5	of Section 1 = $I_{41} = I^C4$
Positions: - 1, 2, 5	of Section 2 = $I_{30} = 0^C3$
Positions: - 3, 4	of Section 2 = $I_{31} = I^C3$
Positions: - 1, 3, 5	of Section 3 = $I_{20} = 0^C2$
Positions: - 2, 4	of Section 3 = $I_{21} = I^C2$

$1^C4 = I_{41} I_a$ $1^C3 = I_{31} I_a$ $1^C2 = I_{21} I_a$

$0^C4 = I_{40} I_a$ $0^C3 = I_{30} I_a$ $0^C2 = I_{20} I_a$

The INPUT SIGNALS are detected on the $K_8 \longrightarrow K_1$ lines by the INPUT REGISTER $D_6 \longrightarrow D_1$, C_6 and C_5 at $\underline{P_m}$ time

$1^D6 = K_8 \underline{P_m}$ $1^D4 = K_6 \underline{P_m}$ $1^D2 = K_4 \underline{P_m}$ $1^C6 = K_2 \underline{P_m}$

$1^D5 = K_7 \underline{P_m}$ $1^D3 = K_5 \underline{P_m}$ $1^D1 = K_3 \underline{P_m}$ $1^C5 = K_1 \underline{P_m}$

$\underline{P_m} = I_4 I_2' I_1' S$

where: -

$1^S = \underline{T}\, T_{41}\, R_{42}'$

$\underline{T} = T_R + T_P + T_{FR}$ etc.

$0^R42 = I_4 \underline{T_n}$

$\underline{T_n} = T_R' T_P' T_{FR}' T_{FP}'$ etc.

$\underline{T}$ = Common Contact (S_{s7}) = Accept information on the $K_8 \longrightarrow 1$ lines

$\underline{T_n}$ = ALL INPUT Devices are at rest.

Sequence: -

(1) $T_n = T'_R\ T'_P\ T'_{FR} =$ Operation Complete Input device at rest $= 0^R42 = I_4\ \underline{T_n}$

(2) $\underline{T} = \overbrace{T_R + T_P} + T_{FR} + T_{FP}$ etc. = FLEX Common contact Input signal is coming. INPUT LINES Activated

(3) $1^S = I_4\ \underline{T}\ T_{41}\ R'_{42}$

(4) $\underline{P_m} = I_4\ I'_2\ I'_1\ S$

(5) $1^{D_{6-1}} = 1^{C_{6-5}} = (K_8 \text{----- } 1)\ \underline{P_m}$

NOTE: - $D_6 - 1$ will initially be "Primed" because of the Operation Code #00 for ICH. Caution must be taken to assure us that C_6 and C_5 are "Primed" by proper programming. Lack of proper programming for C'_6 and C'_5 will effect the 1st. character input.

(6) $1^K a = I_4\ S\ T_{41}$ (6a) $0^S = I_4\ \underline{T_n}\ T_{41}$

(7) $1^N 5 = I_4\ I'_1\ S'\ K_a\ T_1$

(8) $1^K g = P_1\ I'_2\ G_o\ N_5\ \underline{T_{14}}\ P'_6$
$+ P_1\ I'_2\ G'_o\ N_5\ \underline{T_{14}}\ P_6$
$1^G 41 = G_1\ K'_g\ T'_{41} + C'_1\ K_g\ I'_1\ D'_o$
$0^G 41 = G'_1\ K'_g\ T'_{41} + C_1\ K_g\ I'_1\ D'_o$
$0^K g = G'_1\ K_g\ I'_1\ D'_o$

K_g = Carry Flip-Flop
N_5 = Gating Flip-Flop
$\underline{N_6} = I_4\ N_5\ I'_1$
$\underline{I_a} = I_4\ I'_2\ I'_1$

(9) $1^A 41 = C_5\ \underline{N_6}$ (9a) $1^D 6 = A_1\ \underline{N_6}\ K_a\ C'_4$
$0^A 41 = C'_5\ \underline{N_6}$ $0^D 6 = A'_1\ \underline{N_6}\ C'_4 + N_5\ K'_a\ I_4$
$0^K a = I_a\ S'\ P_6\ \underline{T_{13}}\ (T_{33})$

(10) $0^{N}5 = T_{41}$

Input Sequence

(1) The INPUT MODE is entered via

$$1^{I}4 = I_3\ D'_6\ \underline{D_{53}}\ \underline{K_o} \text{ and } 0^{I}2 = \underline{NR_3}\ \underline{K_o}\ D'_4$$

(2) R_{42} is Primed.

(3) No action will take place until we attempt to INPUT a character from one of the Input devices.

(4) The $\underline{T_n}$ Primary is made false.

(5) The K-lines are energized.

(6) The common-contact is "made" - producing $\underline{T}$.

(7) "Set" S-flip flop.

(8) $\underline{P_m}$ primary comes true.

(9) "Set" $D_6 \longrightarrow C_5$ as K-lines indicate. $(K_8 \longrightarrow 1)\ \underline{P_m}$.

(10) "Set" K_o at the end of the word time.

No further action until the Input devices cycle is complete. Then we:

(1) Break the common contact. $\underline{T}$ no longer present.

(2) The K-lines are de-energized.

(3) $\underline{T_n}$ contact is made.

(4) "Reset" S-flip-flop at the end of the word time.

(5) $\underline{P_m}$ gate goes false.

(6) "Set" N_5 at T_1 of next word-time.

(7) Gate $\underline{N_6}$ comes true.

(8) Shift $D_6 \longrightarrow C_5$ into the A-register, recirculating with the A-register. $D_6 \longrightarrow C_5$ go to the L.S.B. positions.

(9) Increment the appropriate CHARACTER COUNTER.

(10) "Reset" $D_6 \longrightarrow C_5$ from $T_{34} \longrightarrow T_{41}$.

(11) "Reset" N_5 at T_{41} causing $\underline{N_6}$ primary to go false.

(12) Wait for next input-character.

We will stop the input of Characters when we detect all ones (1's) in the CHARACTER COUNTER, and C_1'

$\circ C_1 = G_1 K_g \underline{T_{20}} I_4$	(If the number of characters was greater than 63)
$1 K_c = I_4 I_1' G_1 K_g \underline{T_{20}} C_1'$	(Correct number of characters have been entered into the Computer)

$1 I_1 = I_1' \underline{K_{41}}$

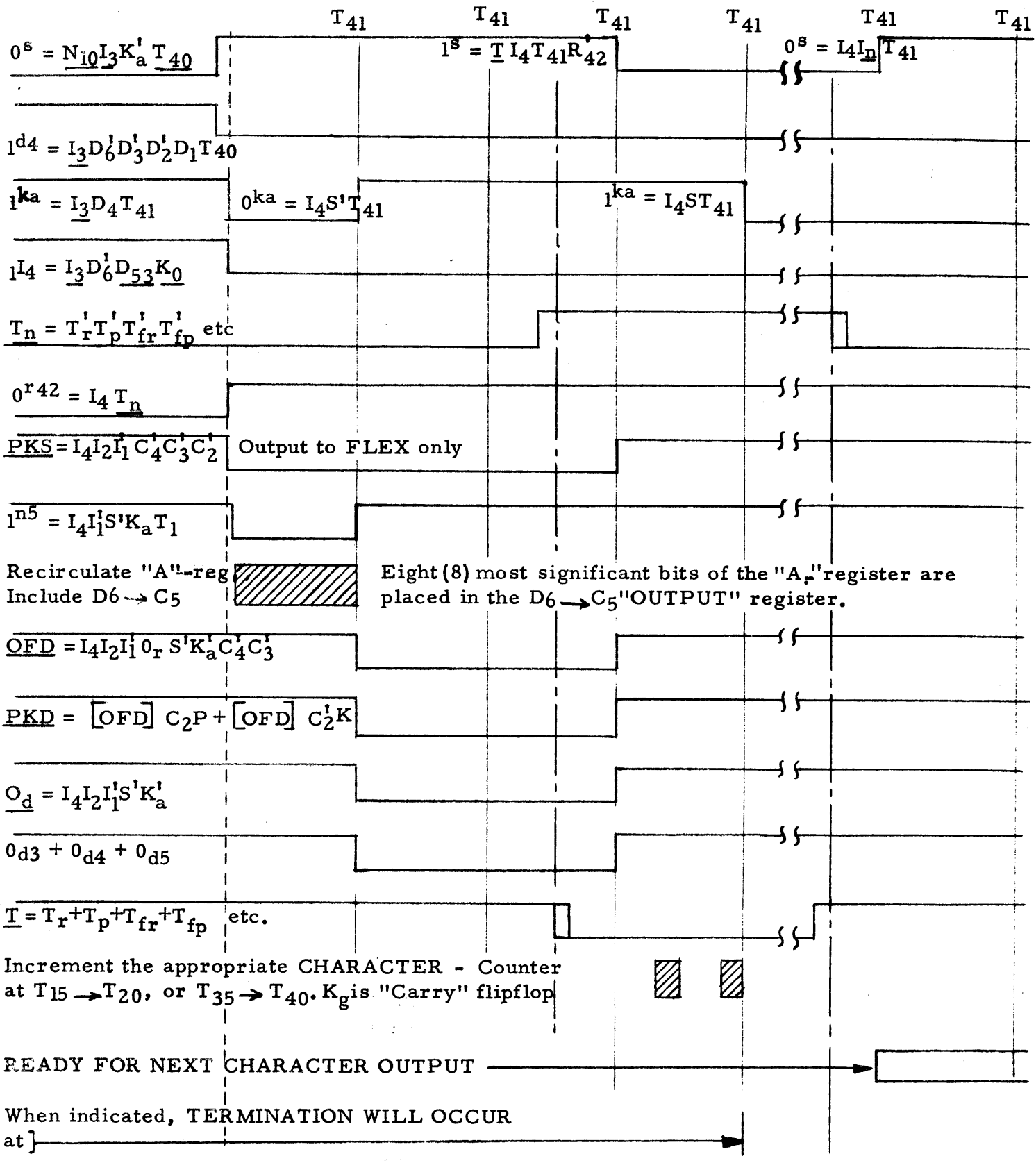
$\circ I_4 = I_4 \underline{K_{41}}$

With transition to the OPERAND or COMMAND MODE - Phase 1.

TIMING SEQUENCE FOR OUTPUT - OCH - #01

DIRECTION MODE

OUTPUT MODE - Phase 2



Final Character Output is indicated by

$0^{C_1} = G_1K_g\underline{T_{20}}I_4$ $\quad 1^{k_c} = I_4I_1G_1K_g\underline{T_{20}}C'_1$

$1^{I_1} = I'_1\underline{K_{41}}$ $\quad 0^{I_4} = I_4K_{41}$

Transition is to the OPERAND or the COMMAND MODES

OUTPUT - OCH - #01

The OCH command causes the computer to OUTPUT from the A-register to the selected piece of peripheral equipment. Output will be from 0 to 128 eight bit characters.

Since the "Control Console Output Switch" overrides the selection of the Output Device, by the address in the command; it is necessary that the switch be placed in the "Auto" position to allow the command to specify the Output Device.

OCH - #01 - $D'_6\ D'_5\ D'_4\ D'_3\ D'_2\ D_1$

OCH - #01 is a TYPE #2 command

Review OPERAND MODE for TYPE # 2 commands.

Phase 1: -

(1) Gate the C(B) into the Z-register.

(2) Gate the UNINDEXED Channel number and sector number into the C-register and CHARACTER Counter, respectively. Load the D-register with the Operation Code.

(3) Analyze the address.

(4) Terminate after ONE WORD-TIME. Transition is to Phase 2.

Phase 2: -

(1) Reload the C-register and the CHARACTER counter with the INDEXED Channel and Sector configuration. Reload the D-register with the Operation Code.

(2) Terminate after ONE WORD-TIME. Transition is to the DIRECTION MODE. Phase 2.

DIRECTION MODE - Phase 2

Entered from the OPERAND MODE via

$$0^{I}1 = \underline{I_n}\ D'_5\ \underline{K_o}$$

During the DIRECTION MODE we will: -

$$1^{D}4 = \underline{I_3}\ D'_6\ D'_3\ D'_2\ D_1\ \underline{T_{40}}$$

$$0^{S} = \underline{I_3}\ K'_a\ \underline{T_{40}}\ \underline{N_{10}}$$

$$1^{K}a = \underline{I_3}\ D_4\ T_{41}$$

$$\underline{I_3} = I'_4\ I'_1\ D'_o\ I'_{3a}$$

$$\underline{N_{10}} = D'_4\ D'_2$$

$$\underline{D_{53}} = D'_5\ D'_3$$

and make the transition to OUTPUT MODE by

$$1^{I}4 = \underline{I_3}\ D'_6\ \underline{D_{53}}\ \underline{K_o}$$

and enter into Phase 2.

We will remain in Phase 2 until we have OUTPUT the specified number of characters.

OUTPUT SEQUENCE

$\underline{T_n}$ = Peripheral equipment is at rest.

(1) "Set" N_5 = Gating flip-flop; and "Reset" R_{42} = TIMING

$$1^{N}5 = I_4\ I'_1\ S'\ K_a\ T_1 \qquad 0^{R}42 = I_4\ \underline{T_n}$$

(PKS primary gate = True IF the KEYBOARD was selected)

$(\underline{PKS} = I_4\ I_2\ I'_1\ C'_4\ C'_3\ C'_2)$

---

(2) Include $D_6 \longrightarrow C_5$ in the recirculation logic for the A-register. This developes a eight (8) bit left shift of the A-register, and places the eight (8) H.S.B.^{s} in the $D_6 \longrightarrow C_5$ Output register.

$$1^{D}6 = A_1\ \underline{N_6}\ K_a\ C'_4$$

$$0^{D}6 = A'_1\ \underline{N_6}\ K_a\ C'_4$$

$$\underline{N_6} = I_4\ N_5\ I'_1$$

C'_4 = NOT CARD EQUIPMENT

$1^{A}41 = C_5\ \underline{N_6}$ $\qquad\qquad$ $0^{A}41 = C_5'\ \underline{N_6}$

(3) The appropriate CHARACTER Counter IS NOT incremented at this time.

(4) "RESET" K_a at the end of this WORD TIME

$0^{K}a = I_4\ S'\ T_{41}$

(5) At T_1 of the 2nd word-time of the OUTPUT MODE: -

(a) OFD Primary = TRUE if FLEX. was specified.

$\underline{OFD} = I_4\ I_2\ I_1'\ \underline{O_r}\ S'\ K_a'\ C_4'\ C_3'$

OFD = Output Drive to FLEX.

$\underline{O_r}$ = OUTPUT READY SIGNAL

(b) PKD Primary = True if FLEX was specified.

$\underline{PKD} = \underline{OFD}\ C_2\ P + \underline{OFD}\ C_2'\ K$

P = Punch has been selected. (absence of PKS. K_{14} "not picked")

K = Keyboard has been selected.

(PKS is present. K_{14} "is picked").

(c) $\underline{O_d}$ Primary = True (No QUALIFICATION)

$\underline{O_d} = I_4\ I_2\ I_1'\ S'\ K_a'$

$\underline{O_d}$ = Output Drive

These "SELECTION" and "DRIVE" signals will initiate the mechanical sequence, in the Output Device, to reproduce the configuration of $D_6 \longrightarrow C_5$.

Additional "OUTPUT DRIVE" signals

$O_{d3} = (O_d)\ C_4'\ C_3\ C_2'$ = Facit Punch Drive

$O_{d4} = (O_d)\ C_4'\ C_3\ C_2$ = Optional

$O_{d5} = (O_d)\ C_4\ C_3'\ C_2'$ = Card Equipment Drive

General conditions at this time.

K_a = Primed

S = Primed

$\underline{T_n}$ = Computer is at rest.

---

(6) Output Drive signals have just come true.

Now: -

Keep S' initially

$$0^S = \underline{T_n}\ I_4\ T_{41}$$

The computer remains in this "static" condition until the peripheral equipment signals that it has accepted the information from the $D_6 \longrightarrow C_5$ flip-flops.

(On the $K_8 \longrightarrow 1$ lines)

This signal will be "$\underline{T}$"

(7) $$1^S = \underline{T}\ I_4\ T_{41}\ R_{42}'$$

This allows Primary Gate $\underline{NR_{13}}$ to come true, and

$$1^{K_g} = \underline{NR_{13}}\ G_o\ P_6' + \underline{NR_{13}}\ G_o'\ P_6$$

$$\underline{NR_{13}} = \underbrace{P_5\ P_4\ P_3\ P_1}_{T_{14} \text{ or } T_{34}}\ S\ K_a'$$

K_g "set" at this time will allow us to increment the CHARACTER COUNTER.

$$1^{G_{41}} = G_1\ K_g'\ T_{41}' + G_1'\ K_g\ I_1'\ D_o'$$

$$0^{G_{41}} = G_1'\ K_g'\ T_{41}' + G_1\ K_g\ I_1'\ D_o'$$

$$0^{K_g} = G_1'\ K_g\ I_1'\ D_o'$$

(8) As we "Set" S, we loose our Output Drive Signals.

(a) $\underline{OFD}$ (c) $\underline{O_d}$

(b) $\underline{PK_d}$ (d) etc. immediately

(9) At the end of this word-time we "set" K_a

$$1^{K}a = I_4 \; S \; T_{41}$$

This 2nd static state is maintained until the peripheral equipment completes its cycle. Cycle completion is indicated by $\underline{T_n}$

(10) $$O^{S} = I_4 \; \underline{T_n} \; T_{41}$$

This completes the OUTPUT CYCLE for ONE CHARACTER.

The sequence would normally start again by

$$1^{N}5 = I_4 \; I_1' \; S' \; K_a \; T_1$$

unless we force termination at the end of the word-time when the CHARACTER COUNTER WAS INCREMENTED BY

$$O^{C}1 = G_1 \; K_g \; \underline{T_{20}} \; I_4$$ (If the number of Characters was greater than 63)

$$1^{K}c = I_4 \; I_1' \; G_1 \; K_g \; \underline{T_{20}} \; C_1'$$ (Correct number of characters have been output from the computer)

$$1^{I}1 = I_1' \; \underline{K_{41}}$$

$$O^{I}4 = I_4 \; \underline{K_{41}}$$

With transition to the OPERAND or COMMAND MODE - Phase 1

FLOATING ADDITION #75

FLOATING SUBTRACTION #74

The FAD and FSB commands cause the contents of the Memory Location, specified by the address, to be algebraically Added or Subtracted from the contents of the A-register. The numbers will automatically scale themselves.

The binary point is counted from bit 33 instead of bit 40. The binary point plus 128 is used instead of a "signed" exponent. This is known as excess 128.

$$FAD = D_6\ D_5\ D_4\ D_3\ D_2'\ D_1 = \#75$$

$$FSB = D_6\ D_5\ D_4\ D_3\ D_2'\ D_1' = \#74$$

All Floating Point numbers must be normalized. If they are not, the computer will assume them to be zeros.

FAD and FSB are TYPE #3 commands.

The operations during the OPERAND MODES - Phases 1 and 2 are standard. There is an additional factor which must be considered: -

During the OPERAND MODE - Phase 3 -

The C(W) is gated into the B-register.

$$1^B41 = M_r\ D_o\ K_a'\ S_o' + L_1\ \underline{D_{10}}\ S' + V_1\ \underline{D_{10}}\ S.$$

$$0^B41 = M_r'\ D_o\ K_a'\ S_o' + L_1'\ \underline{D_{10}}\ S' + V_1'\ \underline{D_{10}}\ S$$

While the A-register recirculates: -

$$1^A41 = A_1\ I_1\ T_1' \qquad 1^A40 = A_{41}\ I_{3a}'$$

$$0^A41 = A_1'\ I_1 \qquad 0^A40 = A_{41}'\ I_{3a}'$$

E_c = Magnitude comparison Flip-Flop.

E_c compares the A-register magnitude to the B-register magnitude.

$$1^{E}c = A'_{41}\ B_{41}\ I_1\ R'_{42} = A < B$$

$$0^{E}c = A_{41}\ B'_{41}\ I_1\ R'_{42} = A > B$$

$$+ T_2\ F'_m \text{ (Initial Reset)}$$

Where R'_{42} had occured by: $- 0^{R}42 = \underline{K_{41}}$

at T_{41} of the word-time the gate $\underline{N_{11}}$ comes true

$$\underline{N_{11}} = I_2\ I_1\ D_o\ T_{41}$$

and we transit to the DIRECTION MODE - Phase 2

$$0^{I}1 = \underline{N_{11}}$$

$$0^{D}o = I_1\ D_o\ T_{41}$$

$$1^{I}3a = \underline{N_{11}}\ D_6\ D_4\ D_3$$

and gate $\underline{M_{oo}}$ is true at T_1: $\underline{M_{oo}} = I_{3a}\ D_5\ D_4\ D_3\ T'_{41}$

and we: $- 1^{R}42 = \underline{M_{oo}}\ T_1\ D'_2\ E'_c$ (E'_c=A > B)

$$0^{A}41 = I'_4\ T_1\ F'_a \qquad \text{(SYNC)}$$

$$0^{B}41 = R'_{42}\ D'_o\ I'_4\ B'_1 \qquad \text{(SYNC)}$$

$$1^{F}a = \underline{M_{oo}}\ D'_2 \qquad (\underline{M_{oo}} \text{ is explicitly } T_1)$$

and transit to Phase 1 is forced by: -

$$1^{K}c = \underline{NR_3}\ D_4\ T_1\ F'_a$$

The "Sign" of B is copied by C_5.

$$0^{C}5 = B'_{41}\ I_{3a}\ D_5\ T_1$$

$$1^{C}5 = B_{41}\ I_{3a}\ D_5\ T_1$$

The sign of the subtrahand C(W) is reversed if the command was FSB. (This is standard for algebraic subtraction)

$$1^{B}40 = B'_{41}\ \underline{M_{oo}}\ D'_1 + B_{41}\ \underline{M_{oo}}\ D_1 + B_{41}\ D_3\ F_a$$

$$0^{B}40 = \underbrace{B_{41}\ \underline{M_{oo}}\ D'_1}_{\text{SIGN REVERSAL}} + \underbrace{B'_{41}\ \underline{M_{oo}}\ D_1}_{\text{COPY}} + \underbrace{B'_{41}\ D_3\ F_a}_{\text{FALSE AT } T_1}$$

for algebraic subtraction the sign of the subtrahend is changed, AND THEN the Operation is <u>CHANGED TO ADDITION</u>. $\longrightarrow 1^{D}1 = \underline{M_{oo}}\ D'_2$ (D_1 <u>becomes</u> Prime at T_2)

and K_a **is preset to zero: -** $0^{K}a = \underline{M_{oo}}$

where $1^{C}2 = F_a\ I_2\ T_1$ (assume positive) ← Preset for a positive sign.

The condition of C_5 is the determining factor for the proper sign.

C_5 monitors the sign of the B-register at T_1

$$1^{C}5 = I_{3a}\ B_{41}\ T_1\ D_5$$

$$0^{C}5 = I_{3a}\ B'_{41}T_1\ D_5$$

NOTE:

The "sign" of the answer for FLOATING ADDITION or FLOATING SUBTRACTION will always be determined by the B-register sign because: -

1. The larger magnitude will always be placed in the B-register.
2. The "sign" of the sum or difference will always be the sign of the larger magnitude.

(Review of algebraic addition and subtraction)

Note: - It will be necessary to DENORMALIZE the fractional portion of the register whose magnitude is the smaller: -

(1) E_c = TRUE = A < B = DENORMALIZE ORIGINAL CONTENTS OF THE A-REGISTER.

(2) E_c = PRIME = A > B = DENORMALIZE THE ORIGINAL CONTENTS OF THE B-REGISTER

The Denormalize process is anticipated: -

(1) IF A > B then EXCHANGE A and B.

(2) IF A < B then DO NOT Exchange A and B.

at the end of the word time we should have the smallest magnitude in the A-register.

Note: - DENORMALIZE LOGIC ONLY EXISTS FOR THE A - REGISTER. No Denormalize operation is done in this Phase.

Transition BACK TO PHASE #2 is forced at T_2 and reset C_2 if C_5 is primed

$$0^K_c = T_2 \, M_{oo} \, D'_2 \qquad 0^C_2 = F_a \, I_2 \, T_2 \, C'_5$$

and the A-register is exchanged with the B-register IF conditions so specified by R_{42}

$$1^A_{41} = B_1 \, I_{3a} \, D'_2 \, R_{42} + A_1 \, I_{3a} \, D'_2 \, R'_{42}$$

$$0^A_{41} = B'_1 \, I_{3a} \, D'_2 \, R_{42} + A'_1 \, I_{3a} \, D'_2 \, R'_{42}$$

$$1^B_{41} = A_1 \, R_{42} \, D'_2 \, D_6 \, I'_1 + B_1 \, I'_4 \, D'_o \, R'_{42}$$

$$0^B_{41} = A'_1 \, R_{42} \, D'_2 \, D_6 \, I'_1 + B'_1 \, I'_4 \, D'_o \, R'_{42}$$

E_c is reset at T_2 in preparation for checking the quality of exponents: $0^E_c = T_2 \, FM'$

NOTE: - At the end of the word-time the A-register will always be GREATER THAN or EQUAL TO the B-register.

At T_{32} we prepare for the equality check of exponents: -

$$\left.\begin{aligned} {}^{1}C_6 &= I_{3a}\, D_5\, \underline{T_{32}} \\ {}^{0}C_6 &= I_{3a}\, D_5\, P_4'\, P_2' \end{aligned}\right\} \qquad C_6 = \text{True from } T_{33} \longrightarrow T_{41}$$

we reset R_1 to inhibit the "adder" $\longrightarrow {}^{0}R_1 = F_a\, I_2\, \underline{T_{32}}$

and we check the most significant fraction bit for a binary one (1).

$${}^{1}K_c = A_1'\, B_1'\, \underline{T_{32}}\, D_2'\, F_a$$

NOTE: A programming qualification stipulates that ALL floating point numbers MUST BE NORMALIZED. Any floating point number that has a zero (0) in the m.s.b. position will be considered a ZERO (0). If A, and B, are BOTH primed at $\underline{T_{32}}$ we assume that both functions are zero (0) and transit to Phase 1, in preparation for TERMINATION at T_{41} via ${}^{1}I_1 = I_1'\, \underline{K_{41}}$

If the floating point numbers are properly normalized they are checked for equality during C_6 = true, by E_c

$$\left.\begin{aligned} {}^{1}E_c &= A_1\, B_1'\, K_a'\, \underline{TE} \\ &\quad + A_1'\, B_1\, K_a'\, \underline{TE} \\ ({}^{0}E_c &= T_2\, FM') \end{aligned}\right| \qquad \underline{TE} = I_1'\, C_6\, FM'$$

where K_a is the "carry" flip-flop. K_a WILL NOT BE "SET" during this word-time because we ARE NOT incrementing the exponent configuration. K_a was "reset" by

$${}^{0}K_a = \underline{M_{oo}}$$

CASE I

E_c at T_{41} = Exponents are UNEQUAL - Make transition to the EXECUTE MODE - Phase 2, for the Equalization sequence.

CASE II

E_c' at T_{41} = Exponents are EQUAL - REMAIN in this DIRECTION MODE - Phase 2, and start the Addition sequence

CASE I We will consider "Case I" initially: UNEQUAL EXPONENTS.

Transition is to the EXECUTE MODE - Phase 2 via

$$0^{I}2 = F_a\ E_c\ \underline{K_o}$$

and at T_{41} we prepare for the Equalization Process by: -

$$0^{D}4 = F_a\ E_c\ T_{41} \qquad 0^{D}3 = F_a\ N'_z\ T_{41} \qquad 1^{D}2 = F_a\ T_{41}$$

D_2 = True will cause B-register recirculation for remainder of command: $1^{B}41 = B_1 D_2 I_{3a}$

$0^{B}41 = B'_1 D_2 I_{3a}$

Upon entering the EXECUTE MODE our Operation code is

$$D_6\ D_5\ D'_4\ D'_3\ D_2\ D_1 = \#63$$

At T_1 of the EXECUTE MODE our general gate and flip-flop configuration is: -

$\underline{M_{oo}}$ = False $(D_5\ D_4\ D_3\ I_{3a}\ T'_{41})$

$\underline{M_{sd}}$ = True $(D_6\ D'_4\ D'_3\ D_1\ I_{3a})$

$\underline{M_1}$ = True $(D_5\ D'_4\ D'_3\ I_{3a})$

$\underline{M_{sr}}$ = True $(D'_4\ D'_3\ D_1\ I_{3a}\ I'_2)$

R'_1; K'_c; D'_o; F_a; F'_m; N'_z; K'_a;

E_c; I_{3a}; C'_6 C_1;

and at T_1 we: - $0^{C}1 = F_a\ T_1$ $\qquad 0^{C}4 = \underline{NR_3}\ T_1\ D_2$ $\qquad \underline{NR_3} = I'_4\ I'_1$

OPERATIONS DURING EXECUTE MODE - Phase 2

(1) Right shift the A-register, FRACTIONAL PORTION ONLY, and insert leading zeros.

$$1^{S} = \underline{R_{10}}\ K'_a\ I_{3a} \qquad \underline{R_{10}} = A_1\ R'_1$$

$$0^{S} = \underline{R_{oo}}\ K'_a\ I_{3a} \qquad \underline{R_{01}} = A'_1\ R_1$$

"S" = One bit delay. Takes the place of A_{40}) to compensate we let A_{39} copy "S".

$$\underline{S_{oo}} = A'_1\ B'_1\ C'_4$$

$$\underline{R_{oo}} = A_1\ R'_1\ C'_4$$

$$\underline{S_{11}} = A_1\ B_1\ R_1\ I_{3a}$$

$$1^{A}39 = S\ \underline{M_{sr}}\ T'_1$$

$$0^{A}39 = S'\ \underline{M_{sr}}\ T'_1$$

We limit the bits, right shifted to the fraction by: - $1^{D}4 = F_a \underline{T_{33}}$; this causes $\underline{M_{sr}}$ to go FALSE, and primary NR_{50} to come TRUE.

With $\underline{NR_{50}}$ = True we allow A_{40} to copy S.

$$1^{A}40 = S\ I_2'\ \underline{NR_{50}}$$

$$0^{A}40 = S'\ I_2'\ \underline{NR_{50}}$$

Insertion of leading zeros was accomplished by: -

$$0^{A}40 = F_a\ N_z'\ D_4'\ K_c'$$

$$1^{A}39 = A_{40}\ D_4$$

$$0^{A}39 = A_{40}'\ D_4$$

A_{40} is initially "Reset", and kept that way.

A_{39} copies A_{40} at $T_{34} \longrightarrow T_{41}$ (which is the correct time for the m.s.b. position). The other C(A) had been shifted.

The EXPONENT portion ($T_{33} \longrightarrow T_{40}$) is incremented by the action K_a (carry flip-flop) with R_1'

$$1^{S} = \underline{R_{oo}}\ K_a\ I_{3a}$$

$$0^{S} = \underline{R_{10}}\ K_a\ I_{3a}$$

S copies the complement of A_1 when K_a is true and R_1 is primed.

$$1^{K}a = F_a\ D_4'\ \underline{T_{32}}\ K_c'$$

(K_a is true at $\underline{T_{33}}$)

$$0^{K}a = \underline{R_{oo}}\ D_1\ I_{3a}\ K_a$$

(K_a is reset when A_1' reads out)

The logic for "S" and "K_a" constitutes a One's complement - "Adder".

The check for EQUAL Exponents is accomplished again by E_c

$$1^{E}c = A_1\ B_1'\ K_a'\ \underline{TE} + A_1\ B_1\ K_a\ \underline{TE} + A_1'\ B_1\ K_a'\ \underline{TE} + A_1'\ B_1'\ \underline{TE}$$

$$0^{E}c = T_2\ F_m'$$

The check performed by E_c actually checks for an INEQUALITY of ONE (1).

NOTE: - The readout of A_1 at $T_{35} \longrightarrow T_{40}$ is the "old" configuration that is now being increased by One (1). This "increased" configuration will be in the A-register at T_{41}.

The equality check by E_c takes this delay into consideration.

If at T_{41} we have E_c = Continue Shift-Increment logic by:

$$0^D4 = F_a\ T_{41}\ E_c$$

E'_c at T_{41} - $\quad 1^I2 = F_a\ N'_z\ E'_c\ \underline{K_o}\ I'_2$

If at T_{41} we have E'_c = Exponents are equal = Make transition to the DIRECTION MODE - Phase 2 and start the Addition sequence. (The same logic as a Case II will now apply)

CASE II - EQUAL EXPONENTS

If the Exponents are initially EQUAL; we will detect this fact at T_{41} of the DIRECTION MODE (First word-time) by having E'_c. In this "Case" we WILL NOT terminate the MODE. We WILL change the OP-COPE TO #73. via:

$$0^D3 = F_a\ N'_z\ T_{41} \qquad\qquad 1^D2 = F_a\ T_{41}$$

This will react on the following gates

$$\underline{M_o} = \text{True} = D_5\ D_4\ D'_3 \qquad\qquad \underline{M_{oo}} = \text{False}$$

This 2nd word time of Phase 2 is the same as standard Add - #73 command. The exceptions involve provision for proper handling of the mantissa and the exponent.

NOTE: The "sign" of the subtrahand was reversed and the OP-CODE changed to ADD, if the original command was FSB

At T_1 we set-up for the 1st word of our Addition sequence.

(Repeater) $\quad 1^{R}1 = \underline{M_o}\ T_1$ = Turn ON add/sub. logic.

(Repeater) $\quad 0^{D}1 = A'_{41}\ B_{41}\ D_1\ \underline{M_o}\ I_2\ T_1$

$\quad + A_{41}\ B'_{41}\ D_1\ \underline{M_o}\ I_2\ T_1$

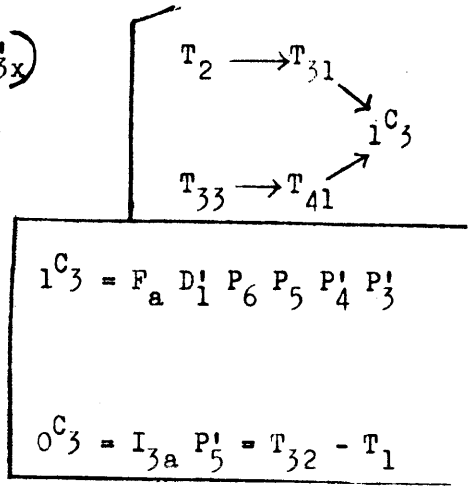
{ Command is Addition; Since SIGNS ARE OPPOSITE = Subtract, or D_1 remains TRUE.

K_a was reset by $\underline{M_{oo}}$ at T_{41}. We will now use it as the "Carry/Borrow" element.

$1^{K}a = \underline{S_{11}}\ D_1\ D_2\ T'_{41} + B_1\ \underline{R_{01}}\ D'_1\ I_{3a}\ D_5\ T'_{41}$ (C'_{3x})

$0^{K}a = \underline{S_{oo}}\ D_1\ I_{3a}\ K_a + \underline{R_{oo}}\ D_1\ I_{3a}\ K_a$

$\quad + A_1\ B'_1\ D'_1\ I_{3a} + \underline{R_{10}}\ D'_1\ I_{3a} + I_{3a}\ T_{41}$



BUT FOR FLOATING POINT - THE ADDER/SUB IS TURNED OFF AT -

$0^{R}1 = F_a\ I_2\ \underline{T_{32}}$

The absence of R_1 will cause "S" * to copy A_1 (Exponent)

$1^{S} = \underline{R_{10}}\ K'_a\ I_{3a}$

$0^{S} = \underline{R_{oo}}\ K'_a\ I_{3a}$

* EXCEPT FOR THE SPECIAL CASE OF OVERFLOW

During the entire word-time B_{40} copies S.

$1^{B}40 = S\ \underline{M_o}\ I_2$ $\qquad$ $0^{B}40 = S'\ \underline{M_o}\ I_2$

while the A-register was being zeroed by: $0^{A}41 = F_a\ I_2\ T_1$

During this operation the possible outcome will be

1. Case I for Add/Sub. = Correct Sum and Exponent in B.

2. Case II for Add/Sub. = The sum in B lacks the MSB (which is the OVERFLOW BIT). The OVERFLOW bit has caused the EXPONENT to be increased by One (1).

3. Case IV for Add/Sub. = The condition is OVERBORROW. The Two's (2^{S}) complement of the MANTISSA is caused by this OVERFLOW condition, while the exponent remains intact.

1 will transit to the EXECUTE MODE - Phase 2, $0^{I}2 = I_2\ \underline{M_o}\ \underline{K_o}$

2 and 3 will transit to the EXECUTE MODE - Phase 1.

NOTE: - A standard "case 3" for add/subtract is impossible. B > A would have produced the exchange of the B-register and the A-register during the first word-time of the DIRECTION MODE. The A-register will always be less than the B-register and produce a standard "case 4" for add/subtract.

SPECIAL

Another condition might arise wherein the "Difference" obtained is ZERO (0) when "B" is subtracted from "A". This special case will cause the command to terminate at the end of the word-time.

Detection of this special case is achieved in the following manner.

C_1 = Main timing for the ZERO test.

$$1^{C}1 = F_a \underline{T_2} \qquad\qquad 0^{C}1 = F_a T_1$$

C_5 = Zero sampling element.

$$0^{C}5 = F_a \underline{T_2} \qquad\qquad 1^{C}5 = F_a C_1 S T_1'$$

Then: -

$$1^{K}c = F_a I_2 C_5' D_1' \underline{P_6 T_{14}}$$

(NO ONE'S (1^s) detected) (Subtraction)

The DIRECTION MODE - Phase 1 is entered immediately and maintained until the end of the word time. The command is terminated by

$$1^{I}1 = I_1' \underline{K_{41}}$$

The A-register would contain a ZERO EXPONENT and a ZERO (0) for the mantissa, with the correct algebraic "sign"

Cases II

This case will indicate Overflow (K_a) at the L.S.B. time for the EXPONENT (T_{33})

E_c = OVERFLOW INDICATOR (Exponents are equal now)

(During 1st. word of Add) $\quad 1^{E}c = A_1 B_1 K_a \underline{TE} + A_1' B_1' K_a \underline{TE}$

and transition is to the EXECUTE MODE - Phase 1

$$1^{K}c = F_a E_c D_2 I_2 T_{41}$$

$$0^{I}2 = F_a E_c \underline{K_o} +$$

and $0^{D}4 = F_a E_c T_{41}$

$$1^{C}6 = I_{3a} D_5 \underline{T_{32}}$$

$$0^{C}6 = I_{3a} D_5 P_4' P_2'$$

During this "Overflow" phase we will RIGHT SHIFT the mantissa portion of the B-register by ONE (1) bit as it enters the A-register. We will then enter the "Overflow" bit into the most significant bit position of the mantissa.

During T_{33} to T_{40} of Phase 2	In anticipation of this one-bit right shift the EXPONENT was incremented by ONE (1) immediately before transition into this Phase

Logic for incrementing the EXPONENT Count - T_{33} to T_{40}

NOTES:

1. Both exponents, in the A-and B-registers are equal.
2. K_a is true at T_{33}, indicating overflow.
3. To reset K_a we need $0^{K}a = \underline{R_{oo}}\ I_{3a}\ K_a$
4. B_{40} copies S = $1^{B}40 = S\ I_2\ \underline{M_o}$
5. Reset R_1; $0^{R}1 = F_a\ I_2\ \underline{T_{32}}$
6. Logic for incrementing is "COMPLEMENT - ADDED"

$$\left.\begin{aligned} 1^{S} &= \underline{R_{oo}}\ K_a\ I_{3a} \\ 0^{S} &= \underline{R_{10}}\ K_a\ I_{3a} \end{aligned}\right\} \text{Complement Logic} \qquad \left.\begin{aligned} 1^{S} &= \underline{R_{10}}\ K'_a\ I_{3a} \\ 0^{S} &= \underline{R_{oo}}\ K'_a\ I_{3a} \end{aligned}\right\} \text{Copy Logic}$$

7. Incrementing the EXPONENT does not/cannot change the sign

Transit to EXECUTE Mode - Phase 1.

The right shift is accomplished by

$$1^{S} = B_1 \underline{R_{01}} K'_a I_{3a}$$

$$0^{S} = \underline{S_{00}} K'_a I_{3a}$$

(1.) because $0^{K_a} = I_{3a} T_{41}$

with K_a initially primed and the A-register all zeroed, WE CANNOT "SET" K_a

---

(2.) because $1^{R_1} = F_a K_c T_1$ and R_1 will remain TRUE for the entire word-time

and

$$1^{A_{39}} = S \underline{MSR} T'_1$$

$$0^{A_{39}} = S' \underline{MSR} T'_1$$

causing a ONE BIT-RIGHT SHIFT.

$$\underline{MSR} = D'_4 D'_3 D_1 I_{3a} I'_2$$

Until we

$$1^{D_4} = F_a \underline{T_{33}}$$

---

and A_{40} will copy "S" for straight copy

$$1^{A_{40}} = S I'_2 \underline{NR_{50}}$$

$$0^{A_{40}} = S' I'_2 \underline{NR_{50}}$$

$$1^{A_{39}} = A_{40} D_4$$

$$0^{A_{39}} = A'_{40} D_4$$

$$\underline{NR_{50}} = I_{3a} D_5 D_4 D'_3 F'_m$$

(Starts at T_{34} - this will insert a)
(one (1) bit into A_{32} at T_{41})

The Overflow bit was inserted into A_{40} via

$$1^{A_{40}} = F_a N'_z D'_4 K_c T'_1$$

Case II commands are terminated at this point because the correct answer is in "A" and it is normalized.

For Cases I and IV. For add/sub will transit, at the conclusion of the First (1st) word time in the DIRECTION MODE - Phase 2 too the EXECUTE MODE - Phase 2 via

$$0^{I}2 = \underline{M_o}\ \underline{K_o}\ I_2$$

Case I we will simply add the C (A) + C (B) through "S" and copy into A_{40}.

$$1^{A}40 = S\ I_2'\ \underline{NR_{50}}$$

$$0^{A}40 = S'\ I_2'\ \underline{NR_{50}}$$

$$1^{R}1 = \underline{M_o}\ T_1$$

$$0^{R}1 = I_{3a}\ D_6\ D_5\ \underline{T_{40}}$$

$$1^{S} = \underbrace{B_1\ \underline{R_{01}}\ K_a'\ I_{3a}}_{\text{Case I}} + \underbrace{\underline{S_{oo}}\ K_a\ I_{3a}}_{\text{Case IV}}$$

$$0^{S} = \underbrace{\underline{S_{oo}}\ K_a'\ I_{3a}}_{\text{Case I}} + \underbrace{B_1\ \underline{R_{01}}\ K_a\ I_{3a}}_{\text{Case IV}}$$

---

* $C_{3x} = C_3$ for Floating Point

Special jumper connection

$$0^{C}3 = I_{3a}\ P_5'$$

$$1^{C}3 = F_a\ D_1'\ P_6\ P_5\ P_4'\ P_3'$$

C_3' is true from $T_2 \rightarrow T_{31}$.

It is then forced true for one-bit time at T_{32} when "K_a" is reset and "D_1" is set via

$$1^{D}1 = F_a\ N_z'\ I_2'\ \underline{T_{32}}$$

Where for Case IV we must RE-COMPLEMENT the complemented answer in the B-register.

$$1^{K}a = B_1\ \underline{R_{01}}\ D_1'\ I_{3a}\ T_{41}'\ D_5\left(C_{3x}'\right)$$

$$0^{K}a = F_a\ D_4\ D_1'\ \underline{T_{32}}$$

The EXPONENT <u>WAS NOT COMPLEMENTED</u> and need only be copied directly by "S", and then "A_{40}".

During this word-time in the EXECUTE MODE - Phase 2 we must check to determine IF OUR ANSWER (that is going from "B" to "S" to "A_{40}") is ALREADY NORMALIZED.

This is accomplished by monitoring the state of "S" at T_{33}.

$S\ T_{33}$ = Answer is Normalized

$S'\ T_{33}$ = Answer is NOT Normalized.

$1^{K}c = S\ T_{33}\ F_a\ D_5\ N_z$

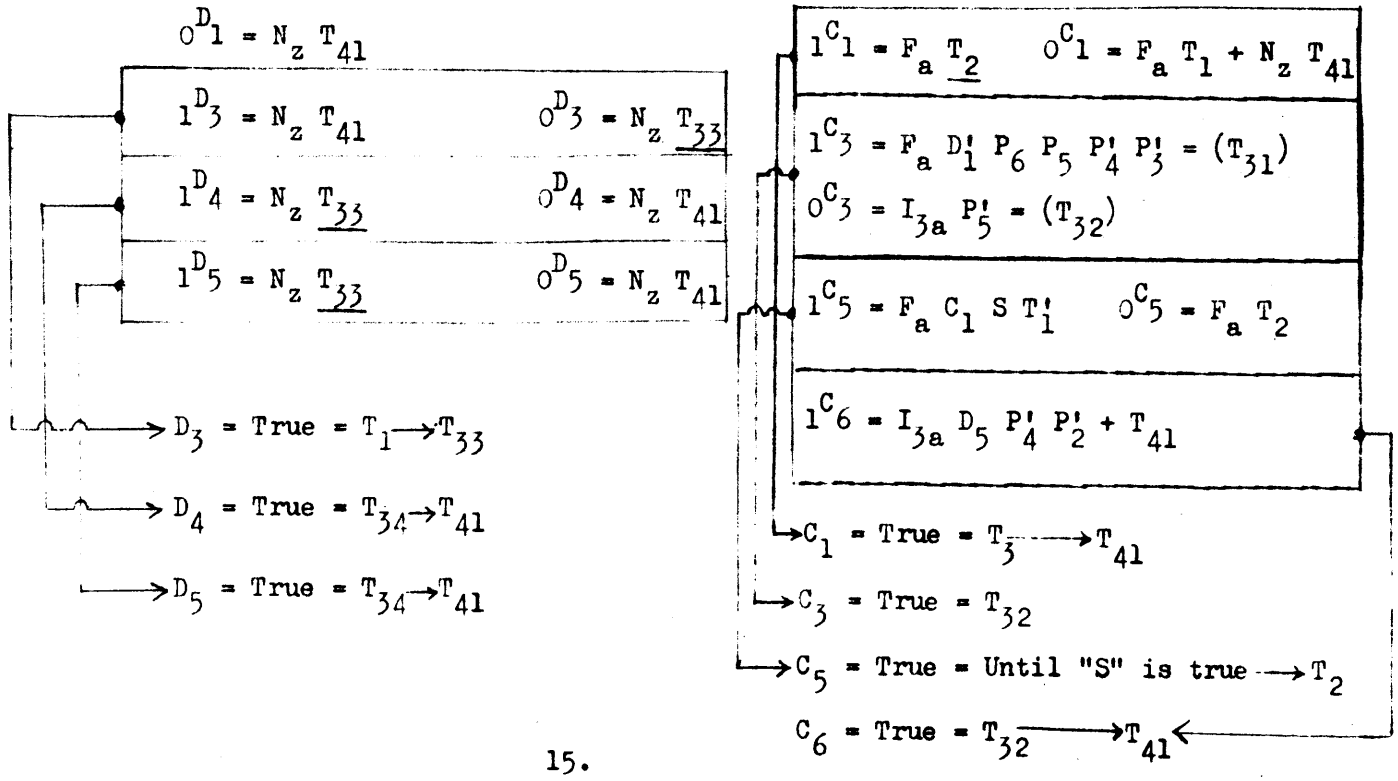
where N_z is turned ON to indicate NORMALIZE operations.

$1^{N}z = F_a\ I_2'\ D_4\ T_{32}$

IF our answer was normalized ($S\ T_{33}$) the command is terminated at T_{41} by: -

$1^{I}1 = I_1'\ K_{41}$

IF the answer IS NOT normalized we REMAIN in the EXECUTE MODE - Phase 2, and proceed with the normalize operations.



$$0^{S} = F_a\ N_z\ T_1$$

$$1^{K}a = N_z\ \underline{T_{32}}\ D_5' + F_a\ D_4'\ \underline{T_{32}}\ K_o'$$

$$0^{K}a = F_a\ D_4\ D_1'\ \underline{T_{32}}$$

NORMALIZE METHOD

The Normalize Process involves the LEFT SHIFT of the A-register until the M.S.B. of the Mantissa is a binary one (1).

Right-shift is accomplished via: -

$$1^{S} = A_1\ R_1'\ K_a'\ I_{3a}$$
$$0^{S} = \underline{R_{oo}}\ K_a'\ I_{3a} + **$$

"S" flip-flop constitutes a ONE BIT DELAY

$$1^{A}41 = S\ \underline{M_{s1}}\ T_1'\ D_o'$$

$$0^{A}41 = S'\ \underline{M_{s1}}\ D_o'$$

$$\underline{M_{s1}} = D_4'\ D_3\ I_{3a}\ C_4'$$

$$\underline{NR_{50}} = I_{3a}\ D_5\ D_4\ D_3'\ F_m'$$

$$\underline{M_o} = I_{3a}\ D_5\ D_4\ D_3'$$

$$0^{D}4 = N_z\ \underline{T_{33}}$$
$$1^{D}4 = N_z\ \underline{T_{33}}$$
$$1^{D}5 = N_z\ \underline{T_{33}}$$

Terminates Right Shift and causes primary gates $\underline{NR_{50}}$ and $\underline{M_o}$ = TRUE.

** NOTE: - The trailing zeros were inserted by: - $0^{S} = F_a\ N_z\ T_1$

The EXPONENT is now Decremented by one (1) to indicate the Left-Shift of the A-register K_a is "set" to accomplish the decrementation

$$1^{K}a = N_z\ \underline{T_{32}}\ D_5' + F_a\ D_4'\ \underline{T_{32}}\ K_c'$$

$$0^{K}a = A_1\ R_1'\ D_1$$

and "S" acts as the Difference element.

$$1^{S} = \underline{R_{oo}}\ K_a\ I_{3a} + \underline{R_{10}}\ K'_a\ I_{3a}$$

$$0^{S} = \underline{R_{10}}\ K_a\ I_{3a} + \underline{R_{00}}\ K'_a\ I_{3a}$$

$$\underline{R_{oo}} = A'_1\ R'_1\ C'_4$$

$$\underline{R_{10}} = A_1\ R'_1$$

and A_{40} copies "S", which does not allow any precession.

$$1^{A}40 = S\ \underline{NR_{50}}\ I'_2$$

$$0^{A}40 = S'\ \underline{NR_{50}}\ I'_2$$

and termination is detected by monitoring "S" at T_{32} for a binary one (1).

$$1^{K}c = D_3\ N_z\ \underline{T_{32}}\ S$$

with command termination occuring at T_{41} via:

$$1^{I}1 = I'_1\ \underline{K_{41}}$$

The correct sign bit is inserted into A_{41} via:

$$1^{A}41 = C_2\ I_{3a}\ \underline{K_{41}}$$

$$0^{A}41 = C'_2\ I_{3a}\ \underline{K_{41}}$$

Transition from $\underline{U_i}\ D_o$ Upon Completion of the Commands Execution.

When the commands execution is completed in $\underline{U_i}\ D_o$ the configuration of the mode and phase control flip-flop is:

$$I_4'\ I_2'\ I_1'\ K_c'\ D_o$$

Termination of $\underline{U_i}\ D_o$ is done by

$$1^{K_c} = \underline{U_i}\ D_3\ D_1\ D_o\ \underline{T_{40}}$$

$$1^{I_1} = \underline{K_{41}}\ I_1'$$

There is, however, other pertinent logic, that was set-up in modes prior to $\underline{U_i}\ D_o$, that will affect us at this time.

The state of the G_o flip-flop is of primary importance now.

G_o' = A "LEFT HAND" command is now being executed.

G_o = A "RIGHT HAND" cimmand is now being executed.

The logic for G_o is listed below. Each minterm is numbered and circled to facilitate their discussion in the succeeding paragraph

$$1^{G_o} = \textcircled{1} G_1\ \underline{T_{22}}\ \underline{I_n}\ K_c + \textcircled{2} I_1'\ \underline{K_{41}}$$

$$0^{G_o} = \textcircled{3} \underline{I_c}\ D_o' + \textcircled{4} \underline{I_c}\ N_1'\ T_{41}$$

Minterm (1) $G_1\ \underline{T_{22}}\ \underline{I_n}\ K_c$ = If the Location Counter, Half Word Bit Position, is TRUE (1), during Phase #1 (or Phase #4) of the Operand Mode, a "RIGHT HAND" command is being executed. "Set" G_o to coincide.

Minterm (2) $I_1' K_{41}$ = During any of the four (4) modes, represented by I_1', when we are in Phases #1 or #4, SET G_o at T_{41}. (This minterm has more dynamic significance when studied with the applicable commands.)

Minterm (3) $I_c D_o'$ =

Upon entering the Command Mode - Phase #3; RESET G_o to indicate a "LEFT HAND" Command is being executed.

Minterm (4) $I_c N_1' T_{41}$ -

In all Phases of I_c, RESET G_o at T_{41}. (N_1' will be true).

We can now see that during I_c; G_o will always be forced "False" because of Minterm (3) or Minterm (4).

As stated previously, the " state" of G_o is extremely important, at this time, because of Mode Control flip-flop I_2.

$$1^{I_2} = G_o' N_1' I_2' K_{41}$$

$$0^{I_2} = G_o I_1' K_{41}$$

Because certain commands will complete their EXECUTION Phase in the Direction Mode, while others will complete theirs in the EXECUTION Mode. It is necessary to determine for both cases if transition will be I_c or I_n.

If a "LEFT HAND" Command is being ooncluded: - Transition will be to I_n. (With possible exception of Transfers).

If a "RIGHT HAND" Command is being concluded: - Transition will be to I_c. (With possible exception of Transfers).

Remember that D_o is still TRUE for the first (1st) word-time following transition from either DIRECTION or EXECUTE MODES.

$1^{I}1$ = transition to Operand ($\underline{I_n}$) or Command ($\underline{I_c}$) Mode

G_o = which half command had been worked on.

$1^{I}2$ = specifies operand mode ($\underline{I_n}$)

$0^{I}2$ = specifies command mode ($\underline{I_c}$)

Transition will be into Phase - #4 of either mode for one (1) word time until:

$$0^{D}o = I_1\ D_o\ T_{41}$$

When, under the described circumstances, we are in Phase - #4 of OPERAND ($\underline{I_n}$), or COMMAND ($\underline{I_c}$), computation stops. The "Neon Display" on the control console is made to indicate the configuration of the LOCATION COUNTER.

The "count" in the LOCATION COUNTER had previously been augmented during the DIRECTION MODE - Phase #2

$$1^{G}41 = G_1' K_g I_1' D_o' + G_1 K_g' T_{41}'$$

$$0^{G}41 = G_1 K_g I_1' D_o' + G_1' K_g' T_{41}'$$

where:

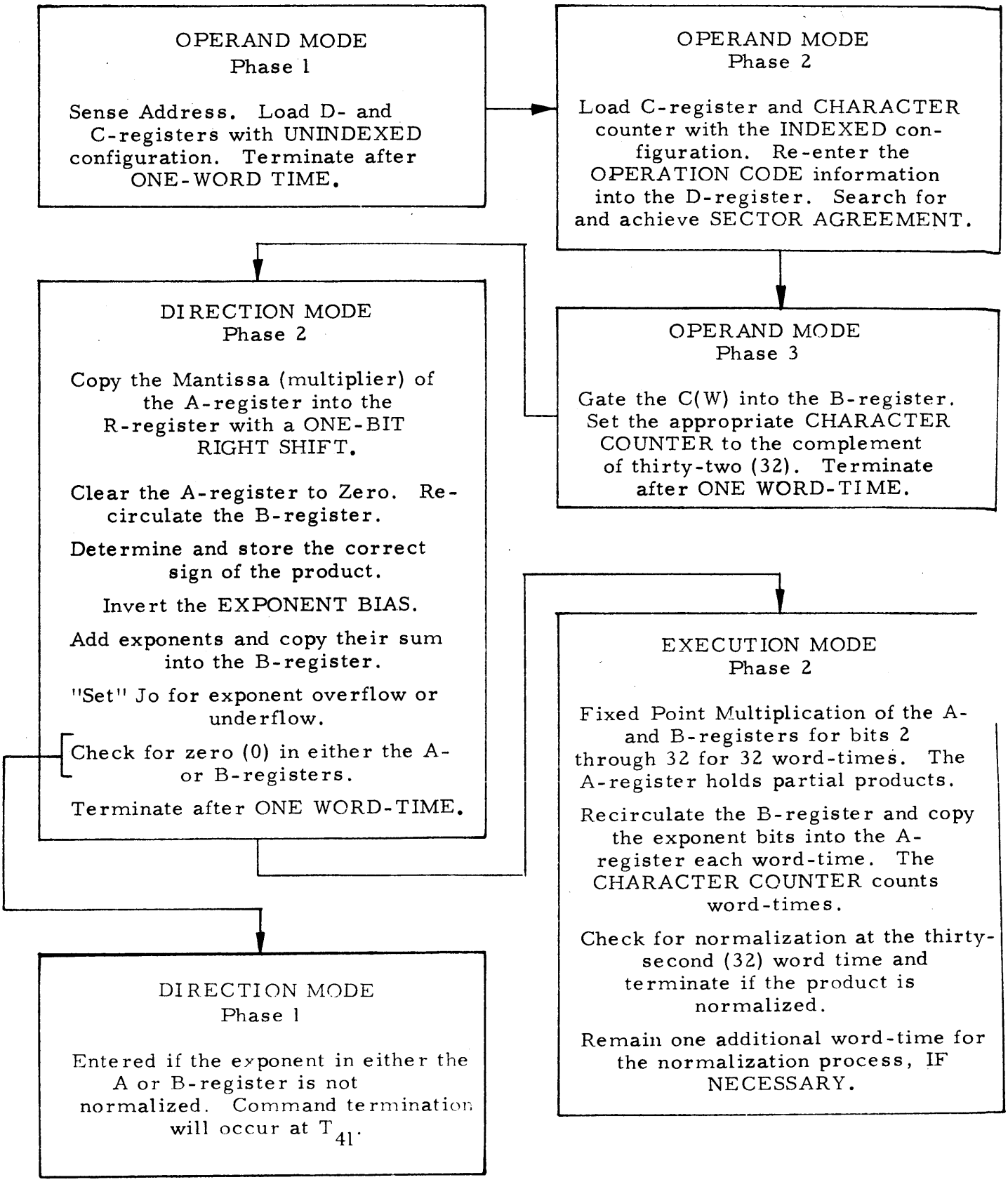
$$1^{K}g = \underline{NR_3}\ I_2\ \underline{T_{21}}\ FA'$$

$$0^{K}g = G_1' K_g I_1 D_o'$$

$$\underline{NR_3} = I_4' I_1'$$

The logic for G_{41}, as shown, is in the standard "One-Input-Adder" configuration.

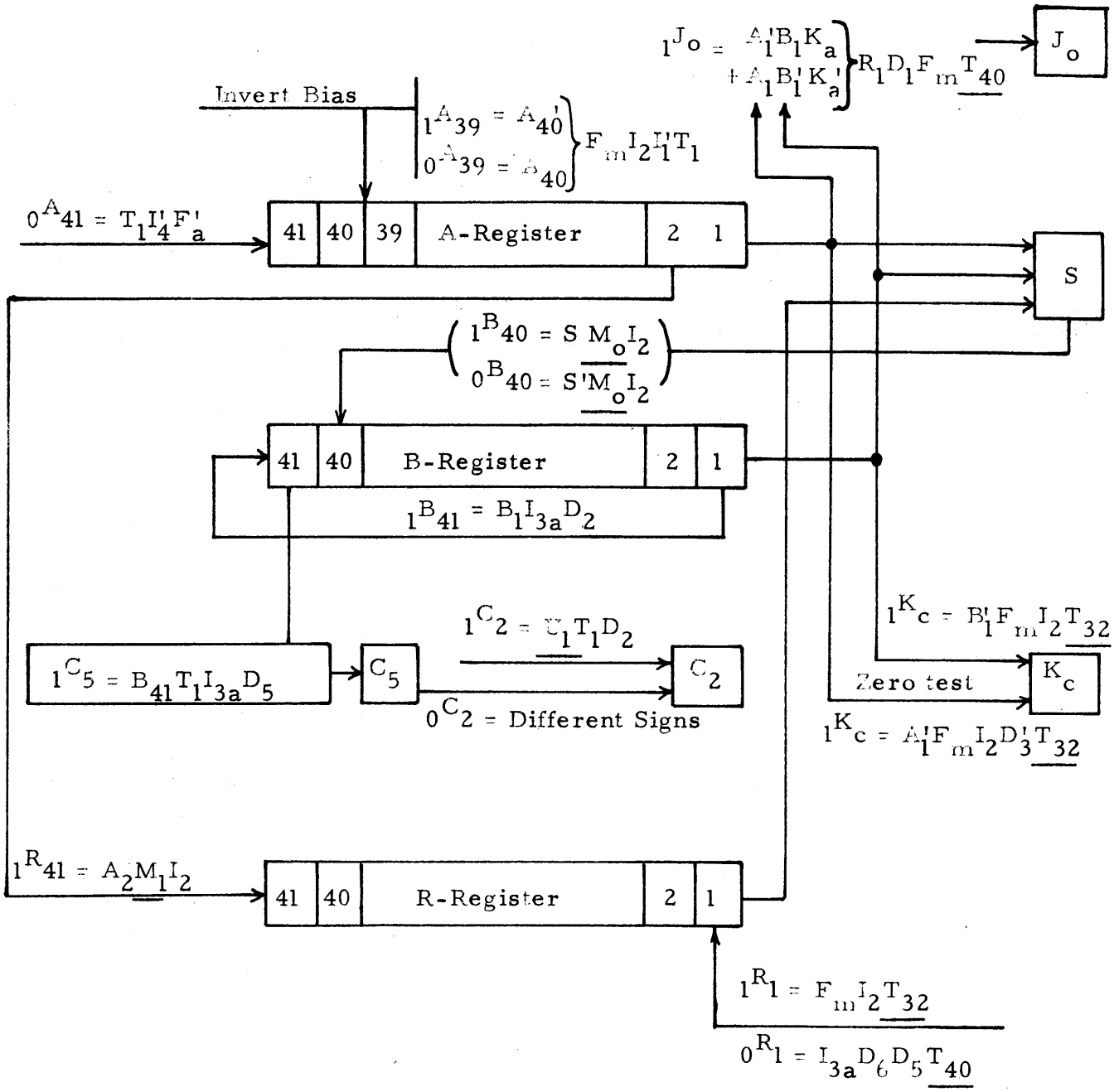
FLOATING MULTIPLY - #23

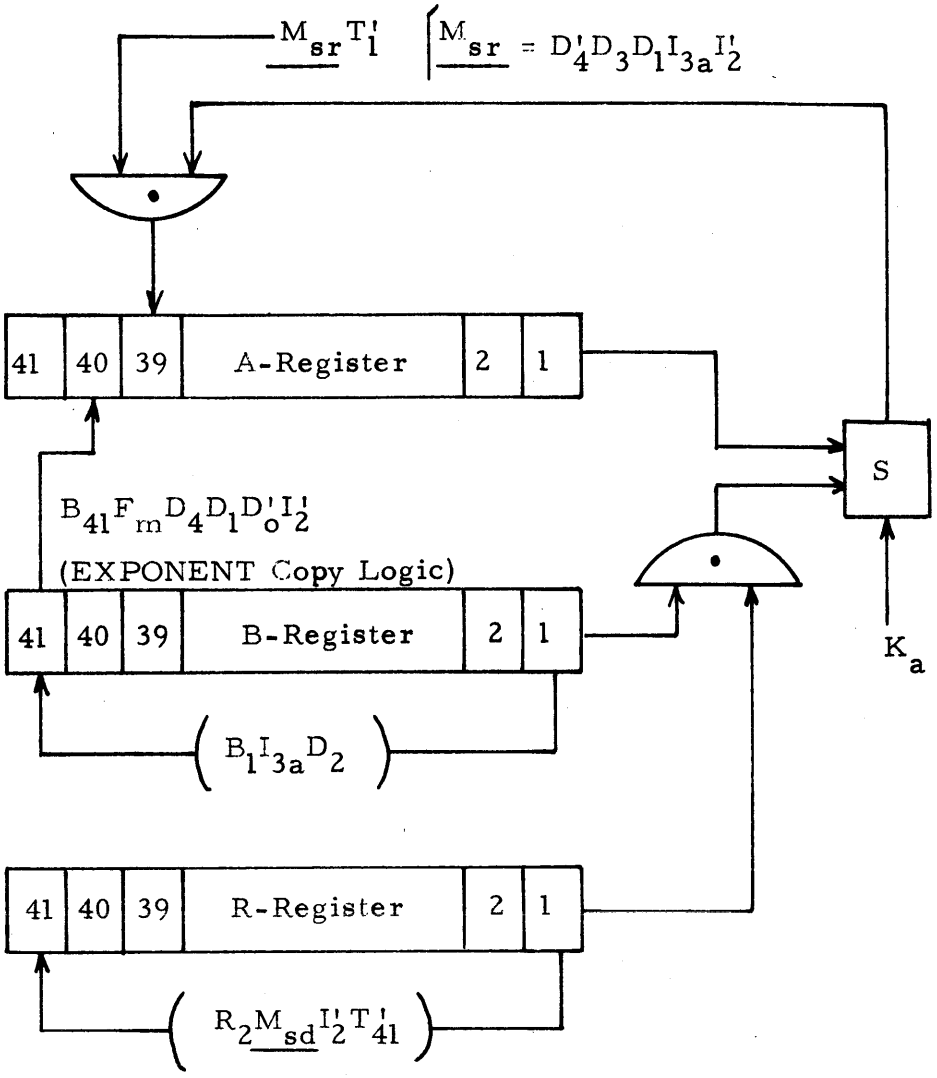


FLOATING MULTIPLY -#23

OPERAND MODE - Phase 1	OPERAND MODE - Phase 2	DIRECTION MODE Phase 2	DIRECTION MODE Phas 2	EXECUTE MODE Phase 2
	OPERAND MODE - Phase 3			

FLOATING MULTIPLY - DIRECTION MODE
PHASE 2





FLOATING MULTIPLY -
EXECUTION MODE - Phase 2

Floating Multiply #23 - $D_6' D_5 D_4' D_3' D_2 D_1$

Floating Multiply is a Type #3 command.

Phase #3 of OPERAND MODE -

(1) Gate C(W) into the B-register

(Standard Gating)

$$1^B41 = M_r D_o K_a' S_o' + L_1 \underline{D_{10}} S' + V_1 \underline{D_{10}} S$$

$$0^B41 = M_r' D_o K_a' S_o' + L_1' \underline{D_{10}} S' + V_1' \underline{D_{10}} S$$

While the A-register recirculates

$$1^A41 = A_1 I_1 T_1' \qquad 1^A40 = A_{41} I_{3a}'$$

$$0^A41 = A_1' I_1 \qquad 0^A40 = A_{41}' I_{3a}'$$

A count of 32 is loaded into the appropriate character counter of the G-register during the OPERAND MODE - Phase 3

$$1^G41 = D_o K_g D_6' T_{20} \qquad (T_{20} \text{ or } T_{40})$$

$$0^G41 = D_o K_g P_5 P_4 P_3' P_4 \qquad (T_{15} \text{ or } T_{35})$$

Where $1^K g = G_o I_4' \underline{T_{14}} P_6' P_1 + G_o' I_4' \underline{T_{14}} P_6 P_1$

K_g is the final determining factor concerning which Character counter will be utilized.

During the last bit-time Floating Multiply Logic comes true

$$1^{Fm} = \underline{N_{11}} D_6' D_4' D_2 \qquad \underline{N_{11}} = I_2 I_1 D_o T_{41}$$

The OP-CODE is forced to #63

via $1^D6 = \underline{N_{11}} D_4' D_3'$

While the Arithmetic Flip-Flop is set via:

$$1^{I_{3a}} = \underline{N_{11}}\ D_2$$

Transition is to the DIRECTION MODE - Phase #2 via: -

$$0^{I_1} = \underline{N_{11}}$$

$$0^{D_o} = I_1\ D_o\ T_{41}$$

and the $\underline{M_1}$ gate is true at T_1 by virtue of I_{3a}.

$$\underline{M_1} = I_{3a}\ D_5\ D'_4\ D'_3$$

The Multiplier bits, (the A-register contents of bit positions 2 through 32) are transferred to the R-register, WITH A ONE-BIT-RIGHT-SHIFT.

$$1^{R_{41}} = A_2\ \underline{M_1}\ I_2$$

$$0^{R_{41}} = A'_2\ \underline{M_1}\ I_2\ T'_{41}$$

$1^{R_{40}} = R_{41}\ D'_o\ T'_{41}$ $\qquad$ $0^{R_{40}} = R'_{41}\ D'_o\ T'_{41}$

At the same time the A-register is filled with zeros -

* $0^{A_{41}} = I'_4\ T_1\ F'_a$

$$0^{A_{40}} = A'_{41}\ I_2$$

$$0^{A_{39}} = A'_{40}\ F_m\ I_2\ T'_1$$

* Note - there is no "Set" logic available for A_{41} during this word time.

The contents of the B-register is recirculated

$1^{B_{41}} = B_1\ I_{3a}\ D_2$ $\qquad$ $1^{B_{40}} = B_{41}\ D_5\ D'_4$

$0^{B_{41}} = B'_1\ I_{3a}\ D_2$ $\qquad$ $0^{B_{40}} = B'_{41}\ D'_4$

The correct "sign" of the Mantissa is stored in the C_2 Flip-Flop via

$$1^{C_2} = \underline{U_1}\ T_1\ D_2 = \text{Assume Positive Sign}$$

Exhibiting a difference of signs

$$0^{C_2} = F_m\ I_2\ \underline{T_2}\ A'_{40}\ C_5 = \begin{cases} \text{B reg. is Positive,} \\ \text{A reg. is Negative} \end{cases}$$

$$+\ A_{40}\ B'_{40}\ \underline{T_2}\ \underline{U_1}\ D_5\ D'_4\ D_2 = \begin{cases} \text{B reg. is Negative,} \\ \text{A reg. is Positive} \end{cases}$$

Where C_5 gives the "sign" of the B-reg.

$$1^{C_5} = B_{41}\ T_1\ I_{3a}\ D_5$$

$$0^{C_5} = B'_{41}\ T_1\ I_{3a}\ D_5$$

The most significant bit of the exponent in the A and B registers, the A_{40} and B_{40} bits, are designated the "Exponent Bias" Bit. It is so called because its state designates the exponent as a positive or negative magnitude.

To perform the prescribed addition of exponents the Exponent Bias of the A-register is inverted as it shifts from A_{40} to A_{39}. This action will allow us to utilize the standard "fixed point" adder logic to gain the correct "sum" of the exponents with the correct "Exponent Bias".

Examples

	40	39	38	37	36	35	34	33			40	39	38	37	36	35	34	33	
A-reg.	1	0	0	0	0	0	1	1	131	INVERSION OF A-REG. EXPONENT BIAS	0	0	0	0	0	0	1	1	
B-reg.	0	1	1	1	1	1	1	0	126		0	1	1	1	1	1	1	0	
											1	0	0	0	0	0	0	1	129=+1

A-reg.	0	1	1	1	1	1	1	0	126=-2	INVERSION OF A-REG. EXPONENT BIAS	1	1	1	1	1	1	1	0	
B-reg.	1	0	0	0	0	0	1	1	131=+3		1	0	0	0	0	0	1	1	
											1	0	0	0	0	0	0	1	+129

A-reg.	1	1	1	1	1	1	1	0	+126	INVERSION OF A-REG. EXPONENT BIAS	0	1	1	1	1	1	1	0	
B-reg.	1	0	0	0	0	0	1	1	+3		1	0	0	0	0	0	1	1	
									(OVERFLOW CONDITION)		1	0	0	0	0	0	0	1	-127

A-reg.	0	1	1	1	1	1	1	0	-126	INVERSION OF A-REG. EXPONENT BIAS	1	1	1	1	1	1	1	0	
B-reg.	0	0	0	0	0	0	1	1	-3		0	0	0	0	0	0	1	1	
									(UNDERFLOW CONDITION)		1	1	1	1	1	1	1	0	+127

The Inversion Logic is

$$1^{A}_{39} = A'_{40}\, F_m\, I_2\, I'_1\, T_1 \quad \text{(Invert Bias)}$$
$$+ A_{40}\, F_m\, I_2\, T'_1 \quad \text{(Direct copy logic)}$$

$$0^{A}_{39} = A_{40}\, F_m\, I_2\, I'_1\, I'_1\, T_1 \quad \text{(Invert Bias)}$$
$$+ A_{40}\, F_m\, I_2\, T'_1 \quad \text{(Direct copy logic)}$$

The "Overflow" and "Underflow" conditions will both cause the OVERFLOW NEON to be lighted.

This is accomplished by the proper "Set" of the J_o flip-flop.

$$1^{J}o = A_1' B_1 K_a K_c' R_1 D_1 F_m \underline{T_{40}} \qquad \text{(OVERFLOW)}$$

$$+ A_1 B_1' K_a' K_c' R_1 D_1 F_m \underline{T_{40}} \qquad \text{(UNDERFLOW)}$$

Where R_1 must be "Set" to allow the standard fixed point adder logic to operate.

$$1^{R}1 = F_m I_2 \underline{T_{32}}$$

$$0^{R}1 = I_{3a} D_5 \underline{T_{40}} D_6$$

From T_{33} through T_{40} bit times only.

The sum of the exponents is written into bit positions T_{33} through T_{40} of the B-register via:

$$1^{B}40 = S \underline{M_o} I_2 \qquad \left[\underline{M_o} = I_{3a} D_5 D_4 D_3' \right.$$

$$0^{B}40 = S' \underline{M_o} I_2$$

The $\underline{M_o}$ Gate is TRUE because the OP-CODE was changed, for the addition of exponents, from #63 to #73 (ADD) via

$$1^{D}4 = F_m \underline{T_{33}} I_2$$

$$0^{D}4 = F_m T_{41}$$

At the end of this single word-time in the DIRECTION MODE, Phase 2 we have;

(1) Copied the Multiplier bits from the "A" to "R" register.

(2) Cleared the A-register to zeros.

(3) Recirculated the Mantissa of the B-register.

(4) Added the Exponents of the A and B registers and copied their sum into the Exponent bits of the B-register.

(5) Determined, and stored, the sign of the product.

(6) Detected the "Overflow" or "Underflow" condition of the sum of exponents with J_o.

One additional check is performed during this word time. It is, "check for a zero magnitude", in either the multiplier or multiplicand (A- or B-register mantissa).

If either the multiplier or the multiplicand is zero the resultant product will be zero.

The command will be terminated at the end of the word-time in this case.

The zero test is performed at T_{32} time, and if the M.S.B. is zero we will "SET" the K_c flip-flop.

$$1^{K_c} = F_m \; I_2 \; B_1' \; \underline{T_{32}} + F_m \; I_2 \; A_1' \; \underline{T_{32}} \; D_3'$$

transition would then be to Phase 1 where termination would occur at T_{41} in the normal manner.

At the end of the single word-time in the DIRECTION MODE (Except for the zero multiplier or multiplicand situation) transition will be to the EXECUTE MODE - Phase 2 via: -

$$0^{I_2} = I_2 \; \underline{M_o} \; \underline{K_o}$$

$\underline{M_o} = I_{3a} \; D_5 \; D_4 \; D_3'$

$\underline{K_o} = K_c' \; T_{41}$

Recall D_4 was "Set" for the addition of Exponents

We will remain in Phase 2 for the next THIRTY-TWO (32) WORD TIMES while the actual multiplication of mantissa is taking place.

The actual multiplication process is identical to that used in the fixed point Multiply #63 command with the addition of those logical equations which allow us to preserve the Exponent. However, for Floating Multiplication, some special gates are used (to accomplish the same effect as the fixed point multiplication) which allow us to inhibit the multiplication logic from T_{33} through T_{41} in order to preserve the Exponent.

Let us briefly review what our registers contain and the configuration of our applicable flip-flops.

(1) The A-register contains all zeros

(2) The B-register contains the mantissa of the multiplicant [the C(W)] and the correct sum of the exponents.

(3) The R-register contains the multiplier [the C(A)] which has been right-shifted one (1) bit position to occupy R_1 through R_{40}.

(4) The "sign" of the product is stored in C_2 flip-flop.

(5) Flip-flop D_4 was "RESET" at T_{41} (= F_m T_{41}) causing OPERATION CODE change to #63 (MPY)

(6) F_m flip-flop remains TRUE

For thirty-two word times the following multiplication sequence takes place.

In practice the Multiplication Sequence proceeds: -

(1) The B-register (Multiplicand) recirculates normally for the entire thirty-two (32) word-times.

$$1^B41 = B_1 \; I_{3a} \; D_2$$

$$0^B41 = B_1' \; I_{3a} \; D_2$$

(2) "S" flip-flop (and accompanying logic) is used in the same manner as during the ADD or SUBTRACT commands.

$$1^{S} = \underline{R_{00}}\ K_a\ I_{3a} + A_1\ B_1'\ K_a'\ I_{3a} + B_1\ \underline{R_{01}}\ K_a'\ I_{3a}$$
$$+ \underline{S_{00}}\ I_{3a}\ K_a + K_a'\ \underline{R_{10}}\ I_{3a} + \underline{S_{11}}\ K_a$$

$$0^{S} = \underline{R_{10}}\ K_a\ I_{3a} + A_1\ B_1'\ K_a\ I_{3a} + B_1\ \underline{R_{01}}\ K_a\ I_{3a}$$
$$+ \underline{R_{00}}\ K_a'\ I_{3a} + \underline{S_{00}}\ I_{3a}\ K_a + \underline{S_{11}}\ K_a'\ T_{41}'$$

$$\underline{R_{01}} = A_1'\ R_1$$
$$\underline{R_{10}} = A_1\ R_1'$$
$$\underline{R_{00}} = A_1'\ R_1'\ C_4'$$
$$\underline{S_{00}} = A_1'\ B_1'\ C_4'$$
$$\underline{S_{11}} = A_1\ B_1\ R_1\ I_{3a}$$

Where: -

$$1^{K_a} = \underline{S_{11}}\ D_1\ T_{41}'\ D_2$$
$$0^{K_a} = I_{3a}\ T_{41} + \underline{S_{00}}\ K_a\ D_1\ I_{3a}$$
$$+ \underline{R_{00}}\ D_1\ I_{3a}\ K_a$$

(3) R_1 holds the multiplier-bit which is used to activate or inhibit the "Adder".

If the multiplier bit is a one (1) we will add the mantissa (Bits 2$\longrightarrow$32) of the B-register to the mantissa (Bits 2$\longrightarrow$32) of the A-register. If the multiplier bit is a zero (0) the "adder" is inhibited and we effectively add zeros to the A-register.

$$1^{R_{41}} = R_2\ \underline{M_{sd}}\ I_2'\ T_{41}'$$
$$0^{R_{41}} = R_2'\ \underline{M_{sd}}\ I_2'\ T_{41}'$$

$$\underline{M_{sd}} = D_6\ D_4'\ D_3'\ D_1\ I_{3a}$$

Causing a one(1) bit right shift each word-time, so the next multiplier bit can be placed in R_1 at T_{41} if it is a binary one (1).

$$1^{R}1 = F_m\ T_{41}\ D_3'\ N_z'\ R_2$$

The "adder" is inhibited each word-time of this phase during bits 33 ⟶ 41. This will allow us to maintain the correct exponent that was determined during the DIRECTION MODE.

$$0^{R}1 = F_m\ \underline{T_{32}}\ I_2'$$

D_4 flip-flop is "set" during the exponent bit-times by: -

$$1^{D}4 = F_m\ \underline{T_{33}}\ D_3'$$

$$0^{D}4 = F_m\ T_{41}$$

The action of D_4 will cause primary gates

MSL and MSD to go "False".

and the A-register will copy the exponent bits from the B-register every word time of the multiplication sequence.

$$1^{A}40 = B_{41}\ F_m\ D_4\ D_1\ D_o'\ I_2'$$

$$0^{A}40 = B_{41}'\ F_m\ D_4\ D_1\ D_o'$$

The A-register receives the "partial product" (sum of the A and B registers) from the S-flip-flop.

$$1^{A}39 = S\ \underline{MSR}\ T_1'$$

$$0^{A}39 = S'\ \underline{MSR}\ T_1'$$

$$\underline{MSR} = D_4'\ D_3'\ D_1\ I_{3a}\ I_2'$$

(Note the presence of D_4')

The A_{39} copy of the S flip-flop causes the partial product to SHIFT RIGHT ONE (1) BIT.

Note: - Unlike the MULTIPLY #63 command, FLOATING MULTIPLY will not place the least significant half of the product in the R-register. The complete, truncated, product will be in the A-register.

The least significant bit of each partial product is lost each word-time.

The termination of the MULTIPLY SEQUENCE is detected by the readout of the CHARACTER COUNTER and the K_g flip-flop.

At this point we will either terminate the command IF the PRODUCT IS NORMALIZED, or we will NORMALIZE the PRODUCT and terminate when the Normalize process is complete.

The CHARACTER COUNTER counts up via: -

$$1^{G}41 = G_1\ K'_g\ T'_{41} + G'_1\ K_g\ I'_1\ D'_o$$

$$0^{G}41 = G'_1\ K'_g\ T'_{41} + G_1\ K_g\ I'_1\ D'_o$$

$$1^{K}g = G_o\ I'_4\ \underline{T_{14}}\ P'_6\ P_1 + G_o\ I'_4\ \underline{T_{14}}\ P_6\ P_1$$

$$0^{K}g = \underline{T_{20}} + G'_1\ K_g\ I'_1\ D'_o$$

The C_1 flip-flop is used to check the most significant bit of the product for normalization.

$0^{C}1 = \underline{M_1}\ T_1 \leftarrow$ (Assume Normalized product)

$1^{C}1 = F_m\ D'_3\ K'_a\ \underline{T_{33}} \leftarrow$ *(Product is not normalized)

*Note: -

$1^{K}a = \underline{S_{11}}\ D_1\ D_2\ T_{41}$ $\qquad$ $\underline{S_{11}} = A_1\ B_1\ R_1\ I_{3a}$

$0^{K}a = \underline{S_{oo}}\ D_1\ I_{3a}\ K_a$ $\qquad$ $\underline{S_{oo}} = A'_1\ B'_1\ C'_4$

During the last word time of the multiply sequence the mantissa portion of A and B registers are added together.

We know that A_{39} copies "S" causing a one-bit-right shift of the product bits.

If K_a (carry flip-flop) is "reset" by logic at T_{32}-time, and appears as $K'_a\ \underline{T_{33}}$, our M.S.B. of the product would be a zero (0) reading out of "S" and into A_{39} at T_{33}. (This zero would go into A_{32} at T_{41} because of the one-bit right shift.)

If the product IS NORMALIZED, termination will occur via: -

$$1^{K}c = G_1\ K_g\ C_1'\ I_{3a}\ F_a'\ \underline{T_{20}}$$

$$0^{K}c = F_m\ C_1\ I_2'\ T_1' \longleftarrow \text{(For Left-Hand commands)}$$

which will produce the termination primary signal

$$\underline{K_{41}}$$

If the product IS NOT NORMALIZED the Multiplication Sequence Phase is maintained for an additional word-time.

The N_z flip-flop is "set" at $\underline{T_{20}}$ or $\underline{T_{40}}$ in preparation for normalization and F_m is "reset"

$$1^{N}z = F_m\ G_1\ K_g\ \underline{T_{20}}\ D_3'$$

$$0^{F}m = N_z\ T_{41}$$

the OPERATION CODE is forced to FORTY-SIX (#46) for a LLS (#42) command.

$$0^{D}5 = N_z\ T_{41}$$

$$0^{D}4 = N_z\ T_{41}$$

$$1^{D}3 = N_z\ T_{41}$$

$$0^{D}1 = N_z\ T_{41}$$

$$0^{C}1 = N_z\ T_{41}$$

and primary gate $\underline{MSL} = D_4'\ D_3\ I_{3a}\ C_4'$ comes true.

S is "reset" at T_1 each word-time by: -

$$0^{S} = \underline{R_{oo}}\ K_a'\ I_{3a} \qquad \underline{R_{oo}} = A_1'\ R_1'\ C_4'$$

$$+\ \underline{S_{oo}}\ K_a'\ I_{3a} \qquad \underline{S_{oo}} = A_1'\ B_1'\ K_a'$$

Where A_1 sync bit is always zero at T_1 via: -

$$0^{A}41 = F_a'\ T_1\ I_4'$$

and B_1, sync bit, will always be primed at T_1. R_1 multiplier bit may be zero or a one.

A zero is copied into the L.S.B of the A-register via: -

$$0^{A}41 = F'_a\ T_1\ I'_4 \longleftarrow \text{(Sync Bit)}$$

$$1^{A}41 = S\ \underline{MSL}\ T'_1\ D'_o \longleftarrow$$

(S will be zero at T_2 and will copy "A_1" from then on)

$$0^{A}41 = S'\ \underline{MSL}\ D'_o$$

The S flip-flop copies A_1: -

$$1^{S} = \underline{R_{10}}\ K'_a\ I_{3a}$$

$$0^{S} = \underline{R_{oo}}\ K'_a\ I_{3a}$$

A_{41} copying S causes a one-bit left shift.

R_1 remains primed the entire word-time.

$$0^{R}1 = I_{3a}\ D_6\ D_5\ \underline{T_{40}}$$

(There is no logic to "set" R_1 at this time)

During the exponent bit-times we change the OPERATION CODE

$$1^{D}5 = N_z\ \underline{T_{33}} \qquad \underline{M_o} = I_{3a}\ D_5\ D_4\ D'_3$$

$$1^{D}4 = N_z\ \underline{T_{33}}$$

$$0^{D}3 = N_z\ T_{33}$$

Gate $\underline{MSL}$ goes False and primary gate $\underline{M_o}$ comes true.

TO SUBTRACT #72

and "set" $K_a = 1^{K}a = N_z\ \underline{T_{32}}\ D'_5$

$$0^{K}a = A_1\ B'_1\ D'_1\ I_{3a}$$

$$+\ \underline{R_{10}}\ D'_1\ I_{3a}$$

S will copy the complement of A_1 as long as K_a remains true, and then straight copy A_1 when K_a goes false.

$$1^{S} = \underline{R_{oo}}\ K_a\ I_{3a}$$

$$0^{S} = \underline{R_{10}}\ K_a\ I_{3a}$$

→ Complement Copy

The Normalize process will never exceed one word-time.

Note: - The R-register, which is really of no consequence to us at this time, will continue its one-bit right shift of the mantissa bits.

$$1^{R}41 = R_2 \underline{MSD}\ I_2'\ T_{41}'$$

$$0^{R}41 = R_2'\ \underline{MSD}\ I_2'\ T_{41}'$$

The corrected exponents is placed in the proper bits of the A-register by

$$1^{A}40 = \underline{NR_{50}}\ I_2'\ S \qquad \left[\underline{NR_{50}} = I_{3a}\ D_5\ D_4\ D_3'\ F_m'\right.$$

$$0^{A}40 = \underline{NR_{50}}\ I_2'\ S'$$

The final termination occurs by

$$1^{K}c = D_3\ N_z\ \underline{T_{32}}\ S$$

where the output from S will go into A_{32} at T_{41}.