NAVSHIPS 94103

VOLUME I

TECHNICAL MANUAL

for

UNIVAC DIGITAL TRAINER

SECTIONS 1 THROUGH 4 AND 6

UNIVAC®
DIVISION OF SPERRY RAND CORPORATION
MILITARY OPERATIONS · UNIVAC PARK · ST. PAUL, MINN.

10 SEPTEMBER 1962

LIST OF EFFECTIVE PAGES

PAGE NUMBERS	CHANGE IN EFFECT	PAGE NUMBERS	CHANGE IN EFFECT
Title Page	Original	A-1 to A-5	Original
ii	Change I	B-1 to B-13	Original
iii to ix	Original		
1-1 to 1-10	Original		
2-1 to 2-2	Original	Volume II	
2-3 to 2-4	Change I	Title Page	Original
2-5 to 2-9	Original	5-ii	Original
3-1 to 3-17	Original	5-1 to 5-10	Original
4-1 to 4-80	Original	5-11 to 5-12	Change I
4-81	Change I	5-13 to 5-48	Original
4-82 to 4-87	Original	5-49 to 5-52	Change I
4-88 to 4-92	Change I	5-53 to 5-57	Original
4-93 to 4-114	Original	5-58	Change I
6-1 to 6-32	Original	5-59 to 5-61	Original

TABLE OF CONTENTS

SECTION 1 - GENERAL INFORMATION

SECTION 2 - INSTALLATION

SECTION 3 - OPERATOR'S SECTION

SECTION 3 - OPERATOR'S SECTION (Cont.)

SECTION 4 - PRINCIPLES of OPERATION

SECTION 4 - PRINCIPLES of OPERATION (Cont.)

SECTION 4 - PRINCIPLES of OPERATION (Cont.)

SECTION 4 - PRINCIPLES OF OPERATION (Cont.)

SECTION 6 - TROUBLESHOOTING AND REPAIR

APPENDIX A - ADAPTER

APPENDIX A - ADAPTER (Cont.)

APPENDIX B - COMPUTER MATHEMATICS

APPENDIX B - COMPUTER MATHEMATICS (Cont.)

LIST OF ILLUSTRATIONS

SECTION 1 - GENERAL INFORMATION

SECTION 2 - INSTALLATION

SECTION 3 - OPERATOR'S SECTION

SECTION 4 - PRINCIPLES of OPERATION

SECTION 4 - PRINCIPLES of OPERATION (Cont.)

SECTION 4 - PRINCIPLES of OPERATION (Cont.)

SECTION 5 - FUNCTIONAL SCHEMATICS

SECTION 5 - FUNCTIONAL SCHEMATICS (Cont.)

SECTION 6 - TROUBLESHOOTING AND REPAIR

APPENDIX A

APPENDIX B

LIST OF TABLES

SECTION 1

GENERAL INFORMATION

1-1. INTRODUCTION.

The UNIVAC® Digital Trainer is a small, stored program computer capable of processing a large quantity of data very rapidly. The Trainer performs arithmetic and logical functions by manipulating binary numbers in automatic or manual modes of operation. Except for its small memory, short word length, and limited input/output section, the Digital Trainer has all of the features of a large general purpose stored program computer. (See table 1-1.)

TABLE 1-1. UNIVAC DIGITAL TRAINER CHARACTERISTICS

Size	9" x 26" x 58"
Power Requirements	115V ±10%, single phase 60 cycle. 750 watts total. UNIVAC Digital Trainer = 600 watts Flexowriter® = 150 watts
Cooling	No special cooling required.
Operation	Parallel binary mode single address logic, 15-bit word length, 512 storage locations, 8-microsecond memory cycle, 20-microsecond average instruction time.

a. GENERAL. - A computer is a device capable of accepting data, instructions to perform operations on that data, executing the specified instructions, and producing the results of the operations. Any problem that can be solved by numerical techniques can be handled and solved by computer operations. Any given problem must be analyzed and resolved into a collection of smaller problems, each of which can be solved by the application of the basic computer operations.

(1) A general purpose digital computer consists of interdependent electrical circuits and storage devices that function to perform the desired operations. Data to be manipulated internally is given a binary numerical representation which is then represented by a state or condition of a component in the computer. Operations are performed on the data by applying a series of sequential signals to the devices storing the data. The results of the operations are then made available to the external equipment.

(2) The major sections of a computer are as follows:

CONTROL - The control section senses an instruction and initiates properly timed command signals which control the execution of the instruction.

ARITHMETIC - The arithmetic section performs the arithmetic and/or logical functions, and provides temporary storage for the results.

STORAGE - The storage section is that portion of the computer which is used for permanent storage of data. This permanent storage area is commonly called the computer memory.

INPUT/OUTPUT - The input/output section is the means by which coded data is entered into the computer and the results of computer operations are then made available to external equipment.

b. CONTENTS OF INSTRUCTION MANUAL. - This instruction manual consists of six sections contained in two volumes.

SECTION 1 - General Information — gives an introduction to basic Trainer concepts and a brief description of the Trainer set, including the principle registers and modes of operation.

SECTION 2 - Installation — provides the necessary information for the preparation and installation of the Trainer for operation.

SECTION 3 - Operator's Section — contains information necessary for an operator to make preparations and interpret Trainer operations during the execution of an instruction. It also contains a detailed repertoire of instructions and information on coding and programming the Trainer.

SECTION 4 - Principles of Operation — presents in detail the operation of the various Trainer sections. Because of the relatively simple electronic principles involved, the emphasis of explanation is on the logical performance of the circuits rather than on the electrical operation.

SECTION 5 - Functional Schematics — contains functional schematics of all portions of the Trainer. These schematics, contained in Volume II, offer a graphic presentation of the logical circuit interconnections.

SECTION 6 - Maintenance and Troubleshooting — presents a tabulation of routines used for maintaining and repairing the Trainer.

1-2. GLOSSARY OF TERMS.

Many terms, while having other meanings in other fields, have specific meanings as they pertain to a computer. The special terms and their specific meaning in computer language are listed as follows:

ABORT - The condition in a computer which results in the next sequential instruction being skipped.

ACCESS TIME - The time interval, characteristic of a memory or storage device, between the instant information is requested from memory and the instant the next request for information from memory can be made.

ACKNOWLEDGE - The indication of the status of data on the input/output lines. Abbreviated ACK.

ADDRESS - A coded number that specifically designates a computer register or other internal storage location. Information is referenced by its address. Portions of computer control are responsible for directing information to or from an addressed location.

ADDRESSABLE - Capable of being referenced by an instruction; e.g., Enter A (f = 01)

ARITHMETIC - A section where logical processes are performed, and operands and results are stored temporarily.

BINARY CELL - An information storing element that can have one or the other of two stable states.

BINARY NUMBER - A number system with two symbols (zero and one) which has two as its base just as the decimal system uses ten symbols (0 - 9) and has 10 as its base.

BINARY POINT - The radix point in the binary system.

BIT - A binary digit, zero or one, represented in a computer by the state of a stage.

BOOTSTRAP - A routine, normally input, contained in memory, used for program loading.

BORROW - In subtraction, a borrow is the additional subtraction of a one from the next partial difference. It occurs when a digit of the minuend is a zero and the corresponding digit of the subtrahend is a one.

BRANCH POINT - A point in a program where a decision is made on the basis of current arithmetic results.

BUFFER - A mode of operation that involves interequipment data transfer.

CAPACITY - The upper and lower limits of the numbers that may be processed in a computer's register.

CLEAR (verb) - To restore a storage or memory device to the zero state.

CODED PROGRAM - A procedure for solving a problem by means of a digital computer. It may vary in detail from a mere outline of the procedure to an explicit list of instructions coded in the language of the machine.

COMMAND - One of a set of signals or groups of signals resulting from an instruction. Commands initiate the individual steps of the instruction.

COMPLEMENT - In reference to one's complement binary arithmetic this simply means switching the one bits of a number to zero, and the zero bits to one.

CONTROL - The circuits of a computer that translate the instruction code, and generate the commands that cause the instruction to be performed.

COUNTER - A device capable of increasing or decreasing its own contents upon receipt of separate input signals.

CORE MATRIX - A magnetic core memory plane containing an array of cores, each of which represents the same column for each storage register in the magnetic core storage system.

CORE STORAGE - A type of storage system in which the magnetic core is the basic storage element.

DEBUG - To isolate and remove all malfunctions from a computer, or all mistakes from a routine or program.

DIGIT - One of a set of characters that are used as coefficients of powers of the radix in the positional notation of numbers.

DUMP - Transfer of information from one piece of equipment to another. Generally involves an output of data from a computer.

ENABLE - A signal of given polarity, applied to a gate circuit, that will allow a computer command to be generated.

FLOW DIAGRAM - A graphical representation of a sequence of operations.

FUNCTION CODE - The portion of the instruction word (2^{14}-2^{09}) that specifies to the control section which particular instruction is to be performed.

HALF-SUBTRACT - The bit by bit subtraction of two binary numbers with no regard for borrows. Abbreviated as HS. The complement of the half-subtract is half-subtract not, abbreviated as $\overline{HS}$.

INSTRUCTION - A word which is a coded directive to the control section to initiate a prescribed sequence of steps necessary to effect a particular logical operation.

INSTRUCTION CODE - Designators constituting an instruction word. The designators are listed below as they appear from highest to lowest order significance in the instruction word:

f = Function code (4 bits)
j = Branch condition (2 bits)
k = Operand interpretation (1 bit)
b = Address modification (1 bit)
y = Operand (9 bits)

INPUT/OUTPUT - A section providing the means of communication between a computer and external equipment, or other computers. Input/output operations involve units of external equipment, certain computer registers, and portions of a computer control section.

INTERRUPT - (1) Internal: indicates the termination of an input or output buffer. (2) External: signal on the data lines that requires computer attention.

JUMP - An instruction which may, depending upon the contents of a given register or position of a given switch, cause the normal sequence of instructions to be interrupted, and an instruction at a remote address to be executed.

LOAD - To enter information into a computer or a storage location.

LOGICAL PRODUCT - The bit by bit multiplication of two binary numbers.

LOOP - Repetition of a group of instructions in a routine or program.

MALFUNCTION - Non-operation of the computer due to a component failure.

MARGIN - A measure of the tolerance of a circuit; the range between an established operating point and the point at which the first circuit starts to fail.

MASTER CLOCK - The primary source of timing signals.

MEMORY - Any device into which information can be introduced, stored, and then extracted at a later time.

MODULUS - The number of values a register can represent. For example, if only the integers from -15 to +15 can be represented in a register, the modulus of this register is 31.

NON-VOLATILE STORAGE - Storage media that retain information in the absence of power. These include magnetic tapes, drums, and cores.

OCTAL NUMBERS - A number system using eight symbols (0 - 7) and having eight as its base.

OPERAND - Coded data representing a number that is involved in computer operations or results from computer operations.

OVERFLOW - The condition which arises when the result of an arithmetic operation exceeds the capacity of the number representation in the computer.

PARALLEL TRANSMISSION - The system of information transfer in which the characters of a word are transmitted simultaneously over separate lines.

PROGRAM - A sequence of coded computer instructions and necessary operands for the solution of a problem.

PAPER TAPE READER - An input device that reads the coded information contained on a punched paper tape by means of photoelectric cells.

RADIX - The number of individual characters used in a number system. Decimal uses ten characters (0 - 9), radix 10. Octal uses eight characters (0 - 7), radix eight. Binary uses two characters (0 - 1), radix two.

READ - To extract information.

REAL-TIME - Computer operation with regard to a specific event or time.

REGISTER - A storage device, usually made up of a series of flip-flops capable of storing a computer word. The condition of the flip-flops ("1" or "0") is usually indicated on a computer maintenance console by a series of neon indicators.

ROUTINE - A sequence of instructions that cause a computer to execute a specific part of a computer program.

SCALE - To shift a binary number either right or left in a register so that the number can be used for further computation in a computer.

SERIAL TRANSMISSION - A system of information transmission in which the characters of a word are transmitted in sequence over a single line as contrasted to parallel transmission.

SET - To change the state of a register to some value other than zero; to change the state of a stage of a register from "0" to "1".

SHIFT - Displacement of an ordered set of characters, one or more columns to the right or left.

SIGN DIGIT - A character used to designate the algebraic sign of a number.

SINGLE ADDRESS (INSTRUCTION) CODE - A system whereby any instruction of a given repertoire will reference only one address of storage while the instruction is being executed.

STAGE - An electronic device or building block with bistable characteristics that enable the device to hold or represent a zero or one.

STORAGE - Consists of devices in which information is set aside for immediate or future use.

TRANSLATION - A change of information from one language or means of representation to another.

TRANSFER - A change of information from one location to another.

VOLATILE STORAGE - Opposite of nonvolatile storage. Information is lost in the event of a power interruption.

WORD - Information coded for computer representation as a series of bits.

1-3. PHYSICAL DESCRIPTION.

The UNIVAC Digital Trainer is a digital computer designed for use in indoctrinating new personnel in the principles of digital data processing. The Trainer receives data, performs logical processes on data, and supplies output data. The Trainer functions are performed by four major sections: Input/Output, Storage, Arithmetic and Control (paragraph 1-1). (See table 1-2.)

a. TRAINER CABINET. (see figure 1-1.) The Trainer is housed in three sections, which are assembled into a single unit 9 inches deep, 26 inches high, and 58 inches wide. The center section contains the operator's console and the power supply. Each of the outer sections contain two chassis of logic modules. These modules are in the form of plug-in printed circuit boards (cards) with mounted parts. The cards are identified by chassis location using the grid system, numbered 1 through 30 from left to right, and A through G from top to bottom.

TABLE 1-2. TRAINER PHYSICAL CHARACTERISTICS

Height	26 inches
Width	58 inches
Depth	9 inches
Weight	
Center section	80 pounds
Left section	90 pounds
Right section	90 pounds
Flexowriter weight	75 pounds

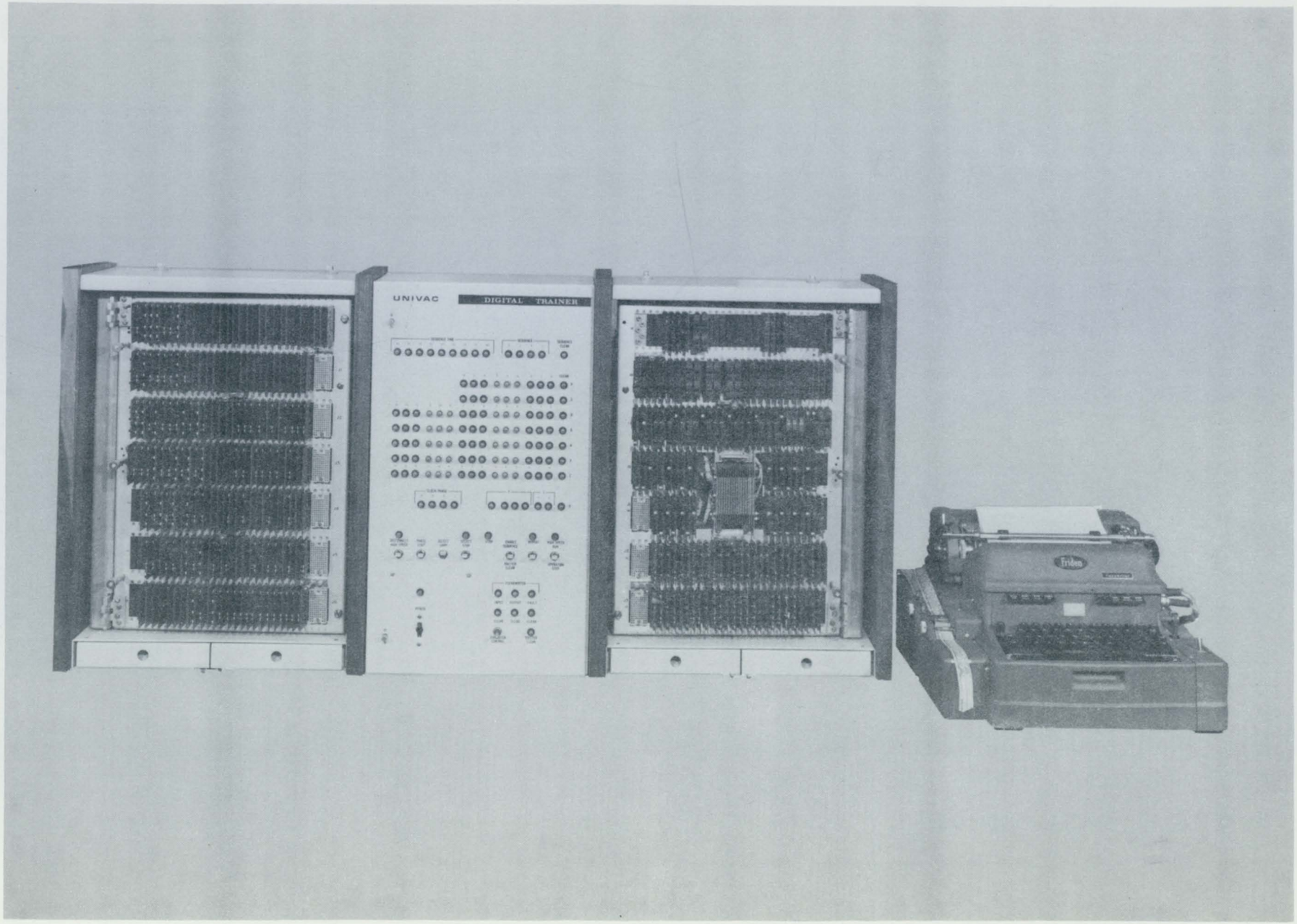


Figure 1-1. Trainer and Flexowriter (Front View)

b. CHASSIS. - The left section of the Trainer contains two logic chassis. The front chassis is labeled A1 and the rear A2. The right section of the Trainer contains one memory and one logic chassis. The front chassis contains the storage or memory stack and is labeled A3. The rear chassis A4 consists mainly of input/output logic.

(1) LOGIC CHASSIS. - The three logic chassis weigh approximately 25 pounds each. On one side of both chassis A2 and A4 there are four connectors which are mated with connectors on the center section. These connectors provide power and signal distribution in the Trainer. They are all ninety pin connectors, although all pins are not used. On the front of each chassis there are test jacks which provide test points on the output of each circuit on that chassis. Circuit repair is usually accomplished by removing and replacing the printed circuit assembly which contains the defective component. Power and signal connections are made with these circuits through a 15 pin connector on bottom of the card.

(2) MEMORY CHASSIS. - This chassis contains the 512 word magnetic core memory stack and its associated circuits. Also on the front of this chassis there are test jacks which provide test points for all the circuits on this chassis.

c. PRINTED CIRCUIT CARDS. - Most of the circuits of the Trainer are contained on printed circuit cards serving as replaceable modules. (See figure 1-2.) This modular approach makes repair easier and helps keep Trainer down-time to a minimum. Each card contains one to four transistors depending on the required inputs to a circuit. Power and signal connections are made with the chassis through the 15-pin connector. Generally pins one, two, three, and four are reserved for the power connections. The printed circuit board itself is made of layers of glass fabric, impregnated and bonded with an epoxy-resin compound. A thin layer of copper is deposited on the board and an etching process leaves only the desired copper conductors on the board. The components and 15-pin connector are attached and soldered to the conductors. The board, but not the connector, is then dipped in a epoxy solution which provides an effective seal around the board and components. Any card can be identified by noting the last three digits of the number stamped on the connector.

d. WIRING. - Circuit and interassembly wiring connections are made through taper pin connectors. The color of the wires is significant only in power distribution and is as follows:

Black = Ground
Green = -3 volts
Yellow = +15 volts
Violet = -15 volts

1-4. FUNCTIONAL DESCRIPTION.

a. GENERAL. - The Digital Trainer is a high-speed digital computer that can rapidly process complex data. Major features of the Trainer include the following:

1) An internal, high-speed, magnetic core storage with a cycle time of eight microseconds and a capacity of 512, 15-bit words.

2) A repertoire of 40 single address instructions, some of which can be modified by designators to produce a total of 60 different instructions.

3) 15-bit word length.

4) Internally stored program.

5) Arithmetic - parallel, one's complement, subtractive.

6) Instruction time - add, 24 microseconds
- multiply, 130 microseconds
- divide, 130 microseconds

7) Clock speed - 2.5 megacycles, 4-phase.

8) Power requirements - 60 cycle, 115 VAC, 750 watts. (Including Flexowriter)

9) Two input/output channels

10) Provision for address or operand modification.

(1) BASIC LOGIC CIRCUIT. - The basic logic circuit of the Trainer is a transistor-inverter amplifier with diode gate inputs (see figure 1-3). This circuit is designed to operate with nominal static voltages of 0 volts and -3 volts representing the "0" and "1" binary states respectively. The circuit output (collector) will be approximately zero volts ("0") when one or more of the inputs is -3 volts ("1"). If all the inputs are zero volts ("0"), the collector output is clamped at -3 volts ("1"). The inputs to the transistors are through the diodes and are isolated from one another. There may be from one to 10 inputs to a single circuit. There is only one output pin, but one circuit may furnish outputs to as many as five circuits. The outputs are not isolated from one another.

(2) FLIP-FLOP. - A flip-flop is a bistable device; that is, a circuit capable of being placed in two stable conditions. These conditions are used to represent binary ones and zeros. A flip-flop is formed by the the interconnection of the two basic logic circuits as shown in figure 1-4. Note that the flip-flop is formed by applying the output of each circuit as an input to the other (There are other inputs and outputs, but the illustrated connections make up the flip-flop,) Thus, if one circuit has a -3 volt ("1") output, the other circuit has this as an input and has a zero volt ("0") output. The two sides and the two conditions of the flip-flop are further defined as follows: the one side and zero side designations have been assigned arbitrarily to the two elements of the flip-flop. The one side normally receives pulses which cause it to have a zero volt ("0") output; thus, the zero side has a -3 volt ("1") output. This condition is called the Set state. Pulses applied to the zero side cause it to have a zero volt ("0") output, and this is called the Clear state. If the flip-flop is set, it is storing a "1" and

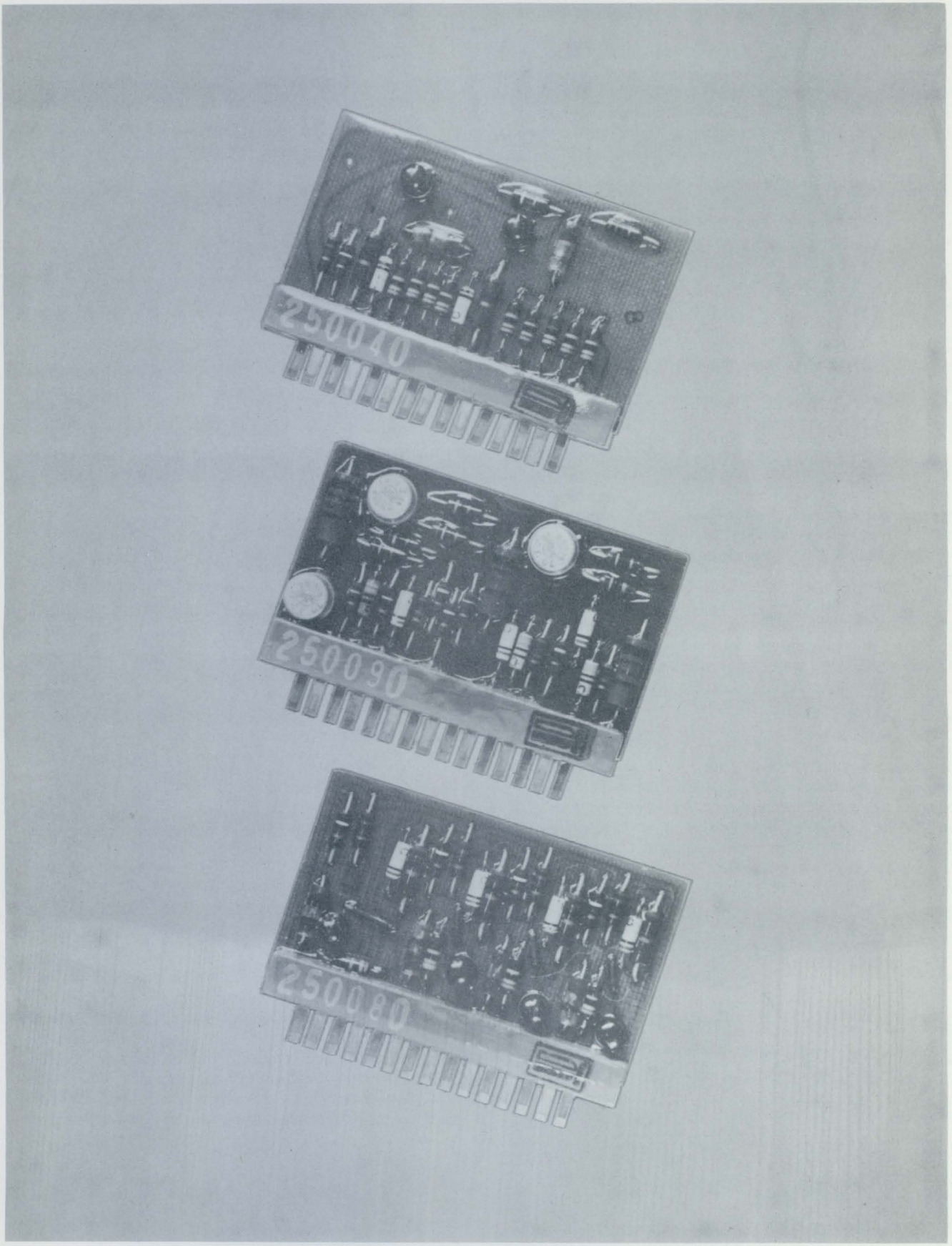


Figure 1-2. Printed Circuit Cards

if it is cleared, it is storing a "0". Also associated with most flip-flops is an Indicator-Driver circuit by which the flip-flop can be set manually. A neon lamp is ionized when the flip-flop is set. The registers of the Trainer are made up of combinations of flip-flops. The flip-flops are then set or cleared depending upon the signals applied. The registers of the computer are considered as volatile storage: when power is turned off, the information stored in the registers is lost.

(3) TIMING. - The clock system consists of four phases, equally spaced in time, which are distributed to all the gate circuits of the computer. These timing pulses (called phase one, two three, four) control and synchronize the transfer of all data throughout the Trainer. A clock pulse is a "0" (zero volts) for approximately 400 millimicroseconds out of each 1600 millimicroseconds. The clock pulses occur at a 2.5 megacycle rate.

b. TRAINER SECTIONS. - The Trainer functions are divided among four sections of the computer: input/output, storage, arithmetic, and control. These sections are integrated in such a way that their functions are dependent upon operations of the other. This sectional division is for the purpose of explanation, rather than to describe any physical separation. Refer to the block diagram, Section V, figure 5-1, for section layout.

(1) CONTROL SECTION. - The control section consists of registers and timing sequences used to control all Trainer functions and to execute the instructions of the internally stored program.

(a) CONTROL SECTION REGISTERS. - The two registers in the Trainer control section are the U and P registers.

The U register (six bits) is the principle control section register. It holds the instruction (function code and designators) while it is being executed. The bits of the U register are applied to a translator that enables the commands necessary to execute the instruction in the U register.

The P register contains the address of the next sequential instruction to be executed. After the current instruction has been executed, the Trainer references the P register to find out what instruction to execute next.

(b) CONTROL SEQUENCES. - There are four major sequences that cause the instructions of the internally stored program to be executed:

A SEQUENCE - The A sequence reads up the instruction to be executed. That is, it obtains an instruction from memory and puts it into the U register where it can be translated an enable the instruction to be executed. The P register is also incremented by one to hold the address of the next instruction to be read up.

B SEQUENCE - The B sequence performs operand modification or indexing. When the b designator of an instruction is set to "1", the contents of the B register (memory address 000) is added to the lower nine bits of the instruction before it is executed.

C SEQUENCE - The C sequence controls the logical or arithmetic operations as prescribed by the function code of the current instruction being executed.

D SEQUENCE - The D sequence controls the acquisition or disposition of the operand as prescribed by the current instruction being executed.

(2) ARITHMETIC SECTION. - The arithmetic section is comprised of registers and special circuits which perform the logical or arithmetic operations of the computer. These operations include addition, subtraction, multiplication, division, shifting, and some special operations. These operations are performed by using the Adder circuit, and special data manipulations between registers.

(a) The A, D, Q, and X registers (15 bits each) are involved in all Trainer arithmetic. The A and Q registers are addressable (can be referenced by an instruction); the D and X registers are not. The normal word length is 15 bits, but there are provisions for a 60-bit product (multiplication) and a 60-bit dividend (division).

(b) The Add instruction (f = 04) is accomplished by placing the augend in the X register and the addend in the D register. The Adder combines these and produces a sum which is entered into the A register.

(c) The Subtract instruction (f = 05) also uses the D and X registers. However, for subtraction, the minuend is placed in the X register and the complement of the subtrahend is placed in D. The adder combines these and produces the difference which is entered into the A register.

(d) The multiplication operation (f = 10, j = 3) is performed by placing the multiplicand in the D register and the multiplier in Q. The recycling of the C sequence controls the shifts and additions necessary to produce the product. The product will be stored in Q if single length (15 bits) and in AQ if double length (30 bits), with A holding the most significant bits. Both the multiplier and multiplicand must be positive numbers when using the multiply instruction.

(e) The division process (f = 11, j = 3) is performed by placing the divisor in the D register and the dividend in AQ. The recycling of the C sequence controls the shifts and subtractions necessary to produce the quotient and remainder. The quotient will be stored in the Q register, and the remainder, if any, in A. Both the divisor and dividend must be positive numbers when using the divide instruction.

(f) The Shift instructions enable a right shift or left shift of the A, Q, or AQ (double length) registers. The maximum shift count is 15_{10}. During right shifts, the lower order bits are lost and the sign of the number is extended (sign extension). For left shifts, the lower order bits are replaced by the higher order bit as the shift progresses (circular).

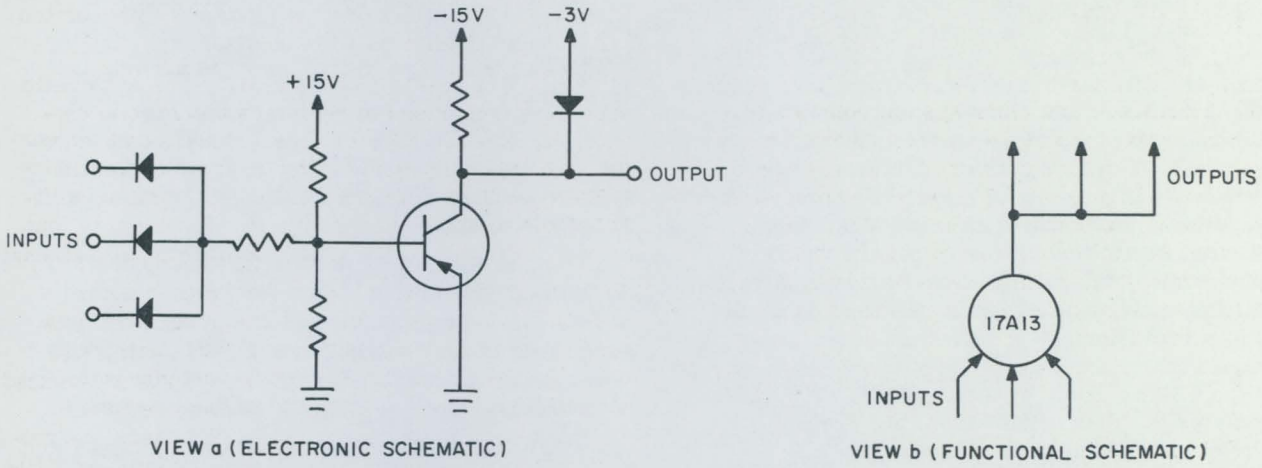


Figure 1-3. Basic Logic Circuit

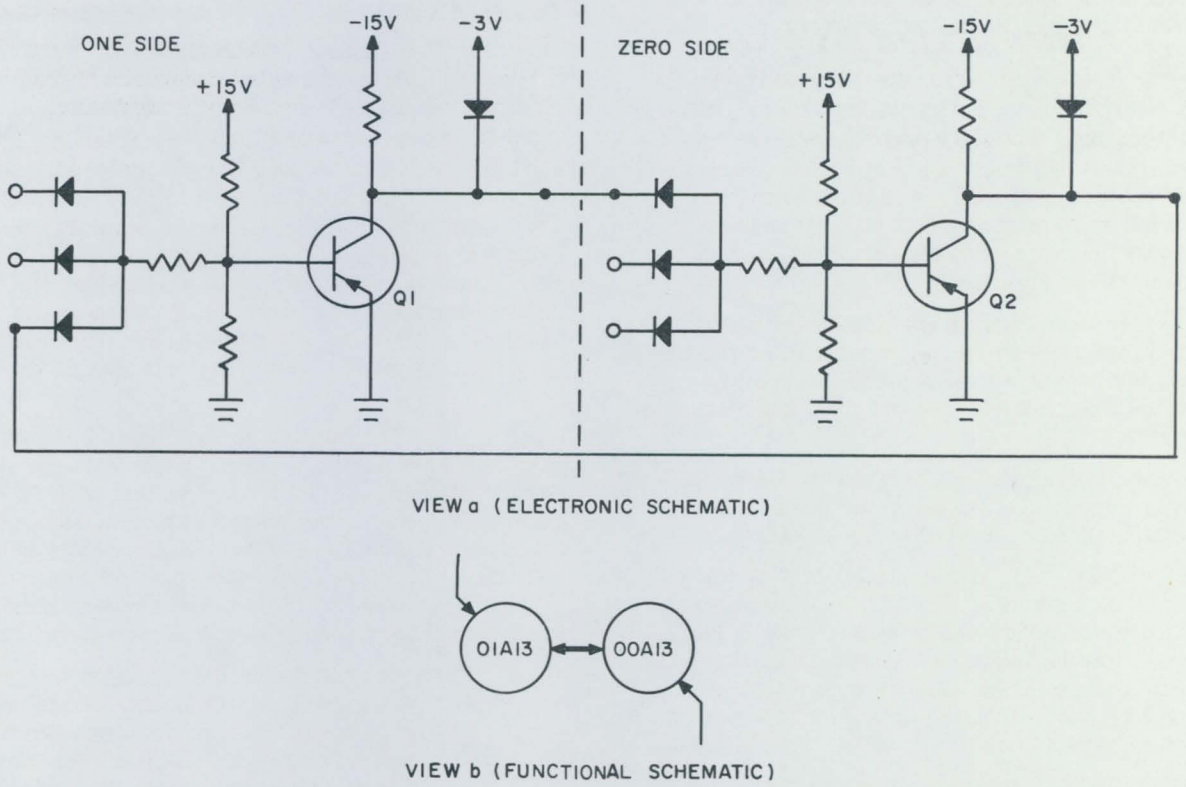


Figure 1-4. Flip - Flop

(g) The Adder (15 bits) is a subtractive device with end-around borrow capabilities which uses the basic principles of one's complement binary arithmetic.

(3) INPUT/OUTPUT SECTION. - Communications between the Trainer and peripheral equipment are conducted via the Q register through two channels labeled "0" and "1". A Flexowriter with paper tape read/punch is the primary input/output device. It communicates with the Trainer via channel zero which consists of the lower six bits of the Q register. Channel one takes input to, and output from, all 15 bits of the Q register. The 15 input/output lines feed an adapter unit which consists of output line drivers, input amplifiers, and control circuits. The adapter unit may then be connected to a variety of standard peripheral units, such as high-speed paper tape readers/punches, magnetic tape units, or magnetic drum. All input/output information transfers occur under program control. The Flexowriter can also be operated in the off-line mode by the flip of a switch. This disassociates the Flexowriter from the Trainer for the purpose of preparing paper tapes, etc.

(4) STORAGE SECTION. - The storage section consists of a magnetic core memory and associated address, transfer, and control circuits.

(a) The memory, constructed of modular arrays of ferrite cores, has a capacity of 512 words, 15 bits per word. A word is referenced by loading the S register with the proper address and the associated translator selects the proper cores. Current pulses are then applied to the cores and the data at the selected address is readout; however, the data is written back into memory as a part of the Memory cycle. The time required for the Read/Write cycle is eight microseconds, during which time the memory lockout is active. This lockout prevents other memory references from occurring at the same time.

Each core of memory is a bistable element capable of storing a "1" or "0" depending upon the direction of magnetization. The time required to drive a core from one state to the other is approximately 1.2 microseconds. Each core is at the intersection of four wires by which its state can be changed and sampled. (See figure 1-5.)

All data exchange with memory is accomplished through the Z register.

c. SPECIAL FEATURES. - In addition to the normal high-speed operation, the controls on the maintenance console allow the Trainer to be run in a variety of modes.

(1) HIGH-SPEED RUN - OPERATION STEP. - In the High-Speed Run mode, the Trainer operates at full speed, stepping through the program at an average rate of one instruction every 24-32 microseconds. In the Operation Step mode, one program step is performed for each switch depression.

(2) REPEAT. - The Repeat mode is used to store manually loaded data inserted via the console registers at consecutive addresses in memory. This switch may also be used to call up data from consecutive addresses for memory.

(3) PHASE STEP AND DISCONNECT HIGH SPEED. - These controls allow the operator to step through the timing sequences within each instruction in a program. The operator can then observe the commands of an instruction as they occur, via the registers and the row of timing sequence indicators.

(4) The CLEAR COMPUTER CONTROL switch allows the Flexowriter to be disconnected from the Trainer and used off-line for the production of program tapes or independent print out of data on paper tapes.

SECTION 2

INSTALLATION

2-1. UNPACKING AND HANDLING.

The UNIVAC Digital Trainer will be received in a number of packages, depending upon the amount of spare parts ordered with the basic Trainer. Each of the assembly units, the control console with power supply, and the Flexowriter will be individually packaged. The following precautions should be carefully observed to avoid damage to the equipment.

a. Handle packages with care. Do not puncture containers or packages before opening and removing contents.

b. Do not drop or jar the packages. To avoid damage, place each of the units or assemblies on a sturdy bench or table as they are unpacked.

2-2. POWER REQUIREMENTS AND DISSIPATION.

The total power dissipation of the Trainer and Flexowriter is approximately 750 watts. All circuits are supplied with power from one main power supply source located in the center or control console cabinet. The Trainer requires 60 cycle, 115 VAC, single phase power which makes it possible to plug it into any convenient outlet. There are no fans or cooling systems employed with the Trainer. Chassis and cards are left exposed to allow normal air circulation in the equipment room to dissipate heat. For greatest reliability, it is suggested that the Trainer be operated in an air-conditioned room, if conditions permit.

2-3. DIMENSIONS AND WEIGHT (approximate, uncrated).

The Trainer is 26 inches high, 9 inches deep and 58 inches long. The associated Flexowriter unit with tape punch and reader is 10 inches high, 17-1/2 inches deep, and 10 inches long. Over-all weight of the equipment is 335 pounds.

2-4. INSTALLING THE TRAINER.

Installation of the Trainer consists of preparing the site or area. Figures 2-1 and 2-2 should be referred to during the installation of the equipment to help identify the various plug and jacks called out in this section.

The installer should ensure that all parts of the Trainer have been included in the shipment. These parts, when referring to right or left sides with respect to the front of the Trainer, should include:

1. Cabinet assembly, left side, including the control and arithmetic chassis with circuit cards (RRU 265701 and 265702).
2. Cabinet assembly, right side, including the memory and Flexowriter chassis with circuit cards (RRU 265701 and 265704).
3. Control Panel assembly with power supply (RRU 265706).
4. One Flexowriter.
5. One power cord with connectors.
6. One input/output adapter (optional).
7. Two cables with 90-pin male connectors.
8. One Flexowriter cable with connectors.
9. One isolation transformer.
10. One input/output adapter cable with connectors (optional).
11. One input/output adapter power cable with connectors (optional).

Equipment that should be made available to the installer but is not supplied as a part of the equipment is as follows:

1. Oscilloscope Tektronix type 535, or equivalent.
2. VOM or VTVM
3. Screwdriver and pliers
4. Soldering iron
5. Needle nose pliers
6. Wire cutter.

The three main subsections, the control panel and the two logic chassis, are held together by thumbscrews which, during shipment are screwed into the pull-out stabilizing stands on the bottom of each logic chassis subsection.

a. SUBSECTION ORIENTATION - The subsections should be oriented on a table so that the chassis containing the memory stack is on the right side of the control panel.

b. PREPARING THE CONTROL PANEL CABINET.

(1) Open front of control panel. Remove the four screws holding T1 and T2 transformers to the bottom of the control panel cabinet.

(2) Remove the eight screws on the back of the control panel cabinet that holds the power supply in place.

CAUTION

The power supply subassembly is hinged on the bottom of the control panel cabinet. Removal of the screws identified in steps (1) and (2)will release this subassembly. Care should be exercised to ensure that the power supply does not fall backwards during the removal of the supporting screws.

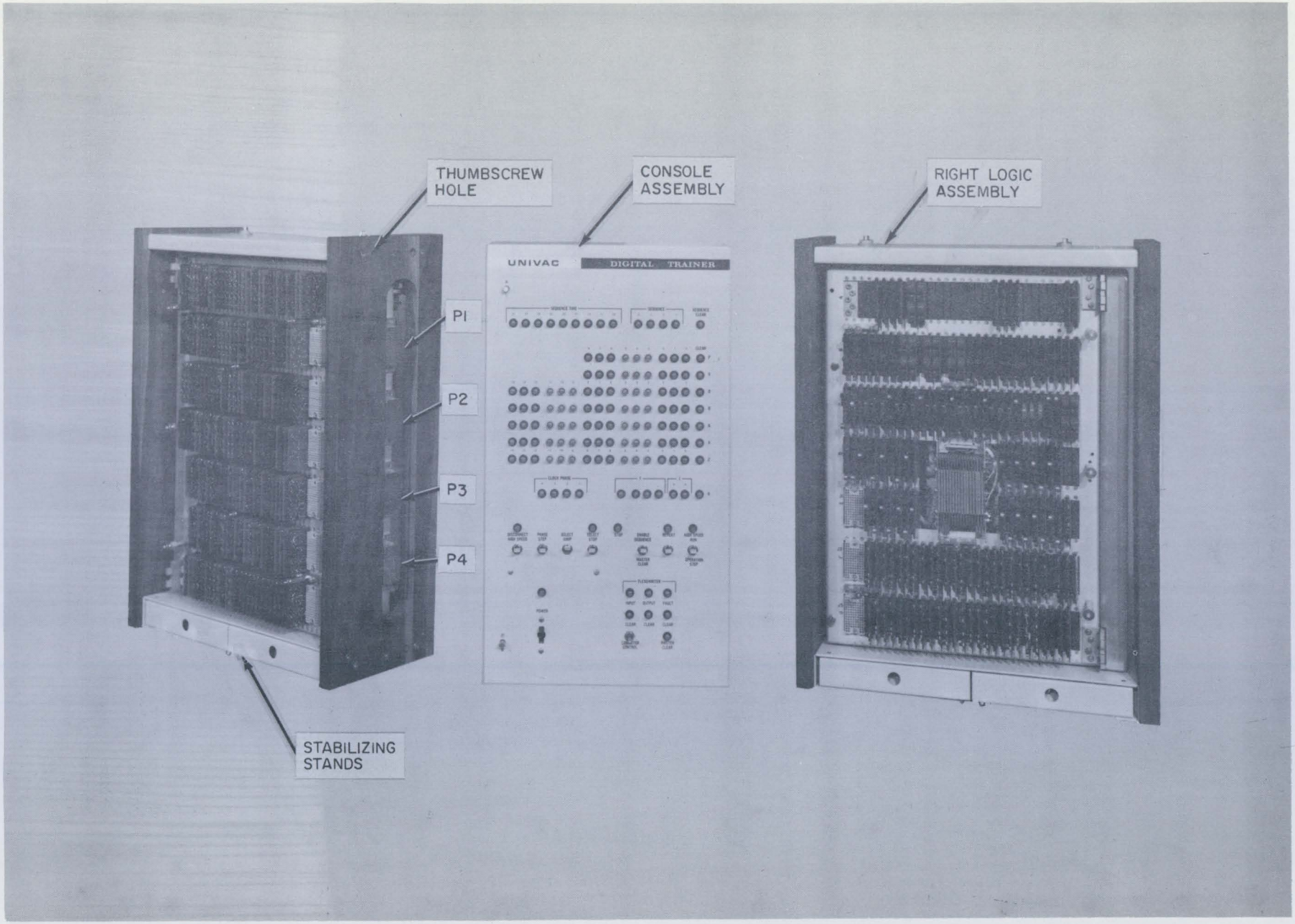


Figure 2-1. Trainer Subassemblies

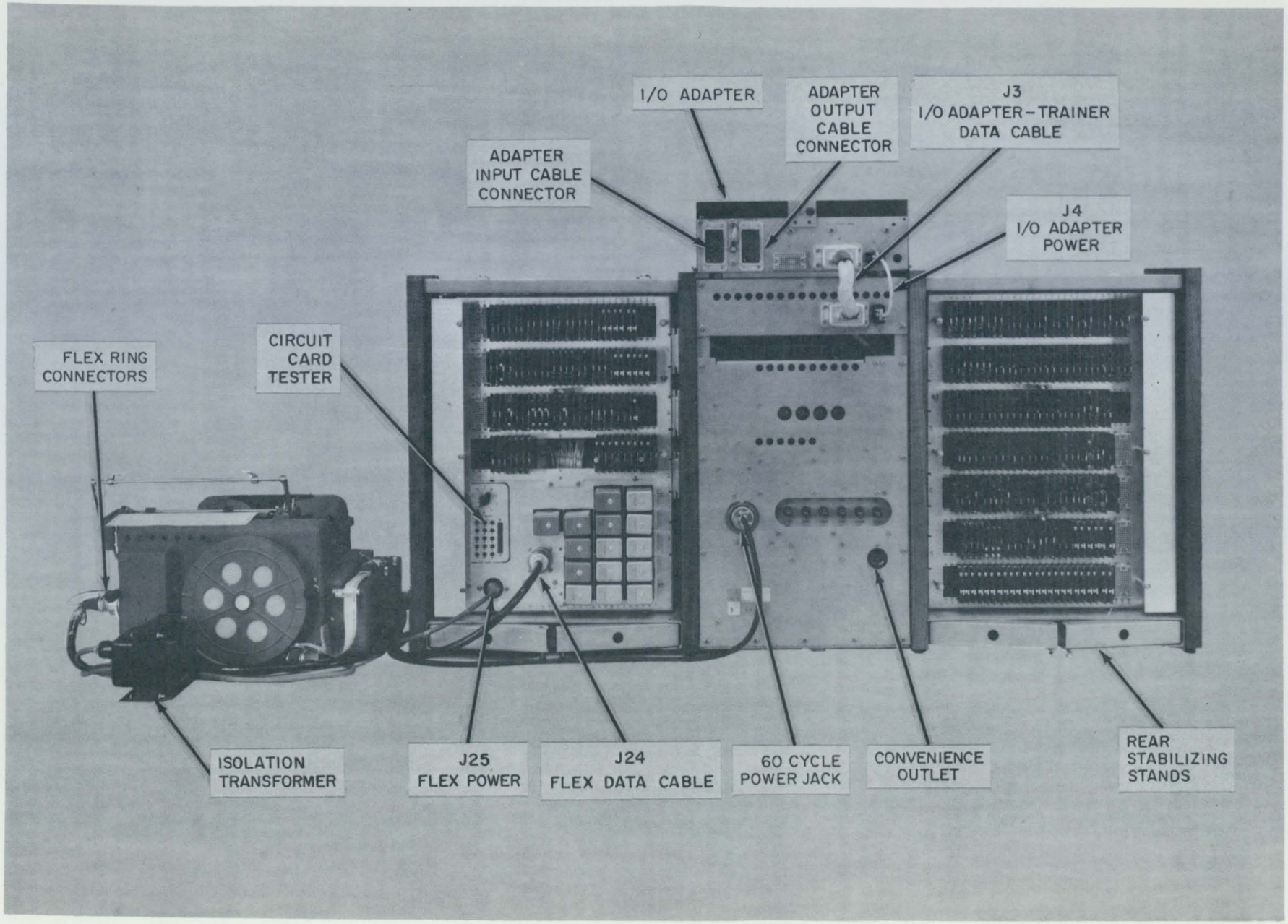


Figure 2-2. Trainer (Rear View)

(3) Lower the power supply subassembly. Ensure proper support of the power supply while it is in this position.

c. SUBSECTION ASSEMBLY.

(1) Remove the four thumbscrews from each of the logic subsections.

(2) Pull the stabilizing stand bar outward to help support the cabinets.

(3) Fit the logic chassis subsections against the control panel. Align the connector wells and thumbscrew holes of each subsection.

(4) Insert thumbscrews through the two upper and lower holes on each side of the control panel subassembly. If the logic chassis have been properly fitted to the control panel, the thumbscrews should screw into the logic chassis subsections. If not, realign the subsection until this is possible. Tighten all thumbscrews.

d. INTER-SUBSECTION CABLING.

(1) Locate and note the cables with connectors contained in the control panel subsection. Four of these cables are wired into the control panel subsection and are identified by the following terms:

(a) A2J3 (assembly 2, jack 3). This male connector is inserted into the female connector on the left-hand logic chassis, second from the bottom (J3).

(b) A2J4 (assembly 2, jack 4). This male connector is inserted into the left-hand logic chassis, bottom connector (J4).

(c) A4J3 (assembly 4, jack 3). This male connector is inserted into the right-hand chassis, second from the bottom connector (J3).

(d) A4J4 (assembly 4, jack 4). This male connector is inserted into the right-hand chassis, bottom connector (J4).

(e) The two cables with 90 pin connectors are inserted into the two upper connectors (A2J1 to A4J1 and A2J2 to A4J2). These cables provide continuity between logic chassis subsections and are wired one to one; that is, the pin of one male connector is connected directly to the same pin of the other male connector. Connect the male connector A2J1 to J1 of the left-hand chassis (top connector). The other end of this cable is connected to the upper connector of the right chassis (A4J1 to J1).

(f) Repeat the same procedure for the remaining cable and the second from the top connector (A2J2 to J2 left and A4J2 to J2 right).

(g) Insure that each of these connectors has been fully seated and are tight.

(h) Return the power supply subassembly back to its upright position and replace the eight screws in the back of the power supply panel. Tighten each screw securely.

(i) Replace the four screws in the transformer base plate and tighten securely.

e. FLEXOWRITER CONNECTIONS.

(1) Locate and identify the Flexowriter control cable. This cable has a ring connector on one end and two ring connectors on the other end.

(2) Insert the single-ended ring connector into J1 located on the I/O control, chassis 4.

(3) Insert the two ring connectors into the respective connectors located on the Flexowriter. Tighten the rings to ensure proper contact is made on both of the Flexowriter cables.

(4) Insert the power cord from the isolation transformer into J2 located on the I/O control, chassis 4.

(5) Plug the Flexowriter power cord into the isolation transformer.

f. INPUT/OUTPUT ADAPTER CABLING.

(1) Position the input/output adapter on top of the control panel cabinet.

(2) Connect the input/output adapter data cable between J3 on the back of control panel cabinet to J3 of the adapter.

(3) Connect the power cable between J4 of the Trainer to J4 of the adapter.

(4) Connect the external equipment input cable to J1 of the adapter and the output cable to J2.

CAUTION

If the input/output adapter is not included in the shipment, the adapter plug provided should be inserted. This plug will ground the inputs to the Q register that are normally fed by the input/output adapter and will eliminate stray pickup.

g. POWER CONNECTIONS - Make sure the power switch is in the OFF position. Insert power cord into the power connector located on the rear of the control panel cabinet. Insert male power cord into a power outlet that will supply 60 cycle, 115 VAC power at 750 watts.

2-5. INSPECTION AND CHECKOUT.

a. VISUAL INSPECTION. - Make certain all circuit cards are firmly seated into the connectors. Inspect the cards to see if proper spacing is maintained between each card. Examine for possible damage to the over-all Trainer if it has been moved or crated since its last use. Make a thorough visual check for loose or broken taper pins, broken or bared wires, and possible broken or damaged connectors.

b. FLEXOWRITER CHECKOUT. - The preliminary check of the Flexowriter will determine the operating ability of the unit as an individual device.

(1) Turn on power and master clear the Trainer by depressing the MASTER CLEAR switch on the panel.

(2) Place the COMPUTER CONTROL switch in the up position. The Flexowriter motor should start. (Be sure the Flexowriter switch is in the ON position.)

(3) Depress PUNCH ON switch on the front of the Flexowriter. Depress TAPE FEED. The punch will start to punch a leader (sprocket holes only).

(4) Depress each of the symbol, number, and letter keys on the keyboard. For each key depressed, one frame of paper tape should be punched.

(5) Depress TAPE FEED again and remove tape from punch. Insert the tape in the reader and depress START READ. The tape should move through the reader and an exact duplicate of the typewritten copy should be made. The punch should also be making a duplicate copy of the tape.

(6) When this check is completed, return the COMPUTER CONTROL switch to the downward position. The Flexowriter motor should turn off.

c. MASTER CLOCK PHASE CHECK. - The procedure outlined in this topic should be used to check the master clock phases on the logic chassis. Use an oscilloscope, Tektronix type 535, or equivalent, with a dual trace preamplifier. Use AC FAST SWEEP with EXTERNAL TRIGGER.

(1) Check terminal board four on chassis one for phase one (refer to chassis map, figure 5-54 of section 5 for location of terminal board).

(2) Use phase one for the external sweep trigger input to the oscilloscope. Use channel A probe for phase one, place scope on dual trace.

(3) Use channel B probe to observe phase two at terminal board five. The start of phase two should occur 0.4 microsecond after the start of phase one.

(4) Move the channel B probe to terminal board six to observe phase three. The start of phase three should occur 0.8 microsecond after the start of phase one.

(5) Move channel B probe to terminal board seven to observe phase four. The start of phase four should occur 1.2 microseconds after the start of phase one. The end of phase four should occur 1.6 microseconds after the start of phase 1.

2-6. FUNCTIONAL CHECK.

A functional check is made by executing each instruction in the repertoire. Some instructions are checked without using memory; others are checked using memory. Normal operation of the Trainer should be used for all checks. If any malfunction is encountered during the execution of an instruction, phase step through the instruction. This provides a means for detecting the exact location of the malfunction. It should be remembered, however, that a memory reference cannot be accomplished during the Phase Step mode of operation.

Master clear the Trainer prior to the execution of each instruction.

a. ENTER LOGICAL PRODUCT.

(1) Load 00077 into the Q register.

(2) Load 00005 into the X register. (X register indicators show complement of flip-flops.)

(3) Enable the timing chain and the D sequence.

(4) Depress the OPERATION STEP switch.

(5) The results of the logical product should appear in the A register (00005).

b. STORE A.

(1) Load 76 into the U register.

(2) Load A with any information.

(3) Load 00100 into the X register.

(4) Enable the timing chain and the D sequence.

(5) Depress OPERATION STEP switch.

(6) The information contained in the A register should be stored in the memory location 00100.

c. ENTER A.

(1) Load 06 into the U register.

(2) Load 00100 into the X register.

(3) Enable the timing chain and the D sequence.

(4) Depress OPERATION STEP switch.

(5) The information stored by the steps in b. above should again appear in the A register.

d. ENTER Q.

(1) Load 12 into the U register.

(2) Repeat steps (2) through (4) as outlined above in c.

(3) The information stored by the steps in b. above should appear in the Q register.

e. SELECTIVE SET.

(1) Load 14 into the U register.

(2) Load 03111 into the A register.

(3) Load 20433 into the X register.

(4) Enable the timing chain and the C sequence.

(5) Depress OPERATION STEP switch.

(6) The A register should contain 23533.

f. ADD.

(1) Load 20 into the U register.

(2) Enter any information into the X and A registers.

(3) Enable the timing chain and the C sequence.

(4) Depress OPERATION STEP switch.

(5) Sum of the X and A registers should appear in the A register.

g. SUBTRACT.

(1) Load 24 into the U register.

(2) Enter any information into the A register.

(3) Load X with a number smaller than that inserted into the A register.

(4) Enable the timing chain and the C sequence.

(5) Depress OPERATION STEP switch.

(6) The difference between the original numbers of X and A should appear in the A register.

h. ENTER B.

(1) Load 30 into the U register.

(2) Load X with 11.

(3) Enable the timing chain and the D sequence.

(4) Depress OPERATION STEP switch.

(5) The information should be loaded into the B register (address 000).

(6) Master clear the trainer.

(7) Load 07 into the U register.

(8) Load 00067 into the X register.

(9) Enable the timing chain and the B sequence.

(10) Depress OPERATION STEP switch.

(11) The information stored during steps controlled by b. should appear in the A register.

i. B REGISTER INDEX (B decrement)

(1) Load 31 into the U register.

(2) Load X with 00001.

(3) Enable the timing chain and the B sequence.

(4) Depress OPERATION STEP switch.

(5) The B register's contents should now be decremented by one.

(6) Master clear the trainer.

(7) Load 12 into the U register.

(8) Enable the timing chain and the D sequence.

(9) Depress OPERATION STEP switch.

(10) The contents of the B register should be displayed in the Q register (00010).

j. ENTER B REGISTER.

(1) Load 32 into the U register.

(2) Load 00100 into the X register.

(3) Enable the timing chain and the B sequence.

(4) Depress OPERATION STEP switch.

(5) Data contained at address 100 should now be entered into the B register (address 000).

(6) To inspect the contents of B, perform steps (6) through (10) of i.

k. B REGISTER INDEX (B increment).

(1) Load 33 into the B register.

(2) Load the value of address 100 into the X register.

(3) Enable the timing chain and the B sequence.

(4) Depress OPERATION STEP switch.

(5) The skip condition should be honored and the B register should be incremented by one.

(6) Depress OPERATION STEP again. The A sequence should be initiated, advance to T23, and re-enable itself again with the P register advanced (skip satisfied).

(7) To check for the contents of the B register (incremented by one) perform steps (6) through (10) of i.

l. STORE B REGISTER.

(1) Load 35 into the U register.

(2) Load any address into the X register.

(3) Enable the timing chain and the B sequence.

(4) Depress OPERATION STEP switch.

(5) The contents of the B register should be stored at the address loaded into X.

(6) To inspect that address location, perform steps (6) through (10) of i., but include the address of the stored B register in the X register.

m. RETURN JUMP.

(1) Load 36 into the U register.

(2) Load the address to be jumped to in the X register.

(3) Enter an address into the P register. (This address is to be stored at the location specified by the address in X.)

(4) Enable the timing chain and the D sequence.

(5) Depress OPERATION STEP switch.

(6) The P register should contain the original contents of X + 1 and the original contents of P should be stored at the address specified by X.

(7) Master clear the trainer.

(8) Load 06 into the U register.

(9) Load the same address in X as was done in step (2).

(10) Enable the timing chain and the D sequence.

(11) Depress OPERATION STEP switch.

(12) The original contents of P should be displayed in the A register.

n. SHIFT INSTRUCTIONS.

(1) Enter 40 into the U register.

(2) Enter the shift count into the X_{L4} register.

(3) Insert a "1" into Q13 position. (A "1" in Q14 will extend the sign and fill Q with "1's" for the number of shifts executed.)

(4) Enable the timing chain and the C sequence.

(5) Depress the OPERATION STEP switch.

(6) The bit in the Q register should be repositioned to the right the number of places specified by the number entered into X_{L4}.

(7) Repeat the above steps with 41 and 42 loaded into the U register and bits set into the A register. These are the instructions for all the shifts of A and AQ right. The results should be the same as was indicated for the Q right shift.

o. MULTIPLY.

(1) Enter the multiplicand into the A register.

(2) Using the Store A instruction (b,), store the multiplicand in any address in memory (address in X).

(3) Master clear the trainer.

(4) Load 43 into the U register.

(5) Enter the multiplier into the Q register.

(6) Load the X register with the address of the multiplicand.

(7) Enable the timing chain and the D sequence.

(8) Depress OPERATION STEP switch.

(9) The product should be displayed in the AQ registers; A holding the most significant bits of the product exceeds the capacity of Q.

p. DIVIDE.

(1) Enter the dividend into the A register.

(2) Using the Store A instruction (b.), store the dividend into any address in memory (address in X).

(3) Master clear the Trainer.

(4) Load 47 into the U register.

(5) Enter the divisor into the Q register.

(6) Load the X register with the address of the dividend stored in step 2.

(7) Enable the timing chain and the D sequence.

(8) Depress the OPERATION STEP switch.

(9) The quotient should appear in the Q register.

q. ADDRESS SUBSTITUTE INSTRUCTION.

(1) Load 54 into the U register.

(2) Load any number into the Q register.

(3) Load an address into the X register.

(4) Enable the timing chain and the D sequence.

(5) Depress the OPERATION STEP switch.

(6) The lower nine bits of the Q register should be entered into the lower nine bits of the memory address specified by the X register leaving the upper six bits unchanged.

(7) To inspect the memory location, execute the steps of c. using the same address in X as in step (3).

(8) Repeat these steps with 56 entered into U and a number in A. The results should be the same except A lower nine bits is stored instead of Q lower nine.

r. A/Q COMPLEMENT INSTRUCTION.

(1) Load 55 into the U register.

(2) Load any number into the Q register.

(3) Enable the timing chain and the D sequence.

(4) Depress the OPERATION STEP switch.

(5) The original contents of Q should be complemented and displayed in the Q register.

(6) Repeat the same steps, 56 in U and any number in A. The contents of A should be complemented and replaced in A.

s. JUMP INSTRUCTIONS.

(1) Load A with the instruction code outlined in step (9).

(2) Store A in memory using the steps shown in 2-6,b, Functional Check.

(3) Master clear the trainer.

(4) Enable the timing chain and the A sequence.

(5) Set the P register to the address where the instruction was stored in step (2).

(6) Depress the OPERATION STEP switch.

(7) The U register should contain the Jump Instruction code with the A sequence re-enabled. The P register should contain the original address plus one.

(8) Depress the OPERATION STEP again. If the jump has been satisfied, the P register should contain the Jump address plus one. However, if the jump is not satisfied, the P register will have been advanced to the next sequential address.

(9) Repeat the above steps for the following:

(a) $A \neq 0$ Jump.

1) Store the following instruction in memory, step (1): 60200.

2) Before step (4), load A with any number.

3) Jump should be satisfied.

4) Repeat the same procedure with A register, with the A register cleared. Jump should not be satisfied.

(b) A = 0 Jump.

1) Store the following instruction in memory, step (1): 61200.

2) Before step (4), load A with any number.

3) Jump condition should not be satisfied.

4) Repeat the same procedure with the A register cleared. The jump condition should be satisfied.

(c) A Negative Jump.

1) Store the following instruction in memory, step (1): 62200.

2) Before step(4) , depress A14 making the A register negative.

3) Jump condition should be satisfied.

4) Repeat the same procedure with the A register cleared. The jump condition should not be satisfied.

(d) Q Negative Jump.

1) Store 63200 in memory, step (1).

2) Before step (4), depress Q14 making Q negative.

3) Jump condition should be satisfied..

4) Repeat the same procedure with the Q register cleared (positive). Jump condition should not be satisfied.

(e) Unconditional Jump.

1) Store 64200 in memory, step (1).

2) Execution of the instruction should satisfy the jump at all times.

(f) Select Jump.

1) Store 65200 in memory, step (1).

2) Execute the instruction. The jump should not be satisfied.

3) Set the SELECT JUMP switch. Execute the instruction again. This time the jump should be satisfied.

(g) STOP.

1) Store 66200 in memory, step (1).

2) Execute the insturction. The STOP indicator should light and the jump should be satisfied.

(h) Select Stop.

1) Store 67200 in memory, step (1).

2) Execute the instruction. The jump condition should be satisfied.

3) Set the SELECT STOP switch in the up position.

4) Execute the instruction again The jump condition should be satisfied. Also the SELECT STOP indicator should be on.

t. INPUT/OUTPUT INSTRUCTIONS. - The input/output instructions are checked by running programs through the Trainer. This check will provide a means of determining the Flexowriter hookup as well as the ability of the Trainer to execute a small routine. The programs can be loaded into the memory manually or by paper tape if a load program (bootstrap) is available.

ADDRESS	INSTRUCTION	REMARKS
100	10 020	Enter Q with function code
101	53 000	Enable output and turn on motor
102	30 000	Clear B register
103	71 104	Jump if no output request
104	11 000	Enter Q with B register
105	51 000	Output Q
106	33 077	Increment B register
107	64 103	Jump to 103
110	64 102	Jump to 102

(1) OUTPUT TEST. - In order to load the test program manually into memory, master clear the Trainer and enable the D sequence. Load 76 into the U register (storeA) and 00100 into the P register. Engage the REPEAT switch, enter the first instruction into the A register, and depress OPERATION STEP switch. The P register will be advanced to the next sequential address with the U register remaining undisturbed. Manually clear the A register, insert the next instruction, and depress the OPERATION STEP again. Repeat the procedure until all instructions have been entered into memory.

After loading the input/output test program, master clear the Trainer and enable the timing chain and the A sequence. Return the REPEAT switch to its normal position. Master clear the Flexowriter circuits by the pushbutton in the lower right-hand corner of the control panel. The first instruction should be loaded into the U register. Continue to depress the OPERATION STEP switch until the Flexowriter motor starts. After this point, the HIGH-SPEED RUN may be engaged. Set the P register to 00100 and depress the OPERATION STEP.

The program will send codes beginning with 00 advancing through 77 to the Flexowriter. This will force the Flexowriter to type out every number, letter, and symbol associated with it. The program will continue to recycle until the OPERATION STEP switch is depressed. There is no program stop instruction included.

(2) INPUT TEST. - The following input test can be inserted into Trainer memory in the same manner explained above. This program does not store or process the information received from the Flexowriter. Rather, its only use is to indicate to the operator that data is being received properly.

ADDRESS	INSTRUCTION	REMARKS
111	10 040	Enter Q with function code
112	53 000	Enable Input and start motor
113	71 113	Jump if no input request
114	50 000	Input Flex word
115	64 113	Jump to 113

After the program has been loaded, the P register set to 111, enable the A sequence and timing chain. Be sure the REPEAT switch is returned to the normal position. Depress the OPERATION STEP and note the contents of the U register. The first instruction should now be loaded into U. Continue to depress the OPERATION STEP switch until the Flexowriter motor starts and the white light on the front panel of the Flexowriter lights. Engage the HIGH-SPEED RUN switch (up position). Depress any of the keys of the Flexowriter keyboard. The Flex code associated with the key should appear in the Q register.

Table 4-30 in section four of this manual shows the flex codes for each character of the keyboard. When each key is depressed, the Q register will be cleared and the new code displayed.

The same procedure can be employed using the Flexowriter reader. Instead of depressing the keys, a tape can be prepared and inserted into the Reader. After the initial start of the program, the 71 instruction at address 113 will continue to be executed until the START READ button is depressed. This action should start the tape movement through the Reader at the rate controlled by the exchange of control signals between the Trainer and Flexowriter. Because of the speed, the operator will not be able to read the characters being entered into the Q register, but the flashing indicators of the lower six bits of Q will indicate that data is being received.

The tests outlined in this topic are only preliminary tests. They can be augmented as necessary when the need arises. They are not meant to be replacements for acceptance tests or any other special routines. If a malfunction is encountered during the preliminary checkout, refer to section six for detailed information on maintenance routines.

SECTION 3

OPERATOR'S SECTION

3-1. GENERAL.

Although the Trainer basically has an automatic mode of operation, the console indicators and switches allow the operator to run the Trainer with considerable flexibility and in slower modes of operation. Basically the switches and indicators affect the speed and logic of the Trainer. By proper operation of the switches and indicators, the operator is able to step through, in a slower mode of operation, the instructions and sequence of operations. The buttons and switches also can be used to set a single stage of a register or clear an entire register, without the use of a programmed instruction.

a. TURN ON PROCEDURE. - To apply 60 cycle power to the Trainer, close the POWER ON switch. The corresponding red indicator will light if the Trainer is receiving power.

3-2. OPERATIONS, MODES, CONTROLS, and INDICATORS.

After power has been applied to the Trainer, one of several modes of operation can be selected by operation of various switches. (See figure 3-1.)

a. HIGH SPEED. - The most common mode of operation is the High-Speed mode. This is the automatic mode of operation in which the Trainer executes sequential instructions. The High-Speed mode of operation is entered by momentarily depressing the MASTER CLEAR - ENABLE SEQUENCE switch, which clears all the registers. Momentarily placing this same switch in the up position enables the A sequence. The address of the first instruction is then inserted into the P register. When the HIGH SPEED RUN - OP STEP switch is momentarily in the up position, the green RUN indicator will light signifying that the Trainer is executing instructions in the High-Speed mode of operation.

b. OPERATION STEP. - The Operation-Step mode allows one instruction to be executed, then an automatic stop. This mode of operation is entered by momentarily depressing the OP STEP switch. After executing one instruction, the Trainer stops after completing the A sequence (reading up) of the following instruction. Depressing the OP STEP switch again will cause this instruction to be executed and the Trainer will stop after completing the next A sequence. This mode of operation is particularly useful when debugging a routine. It allows for inspection of each instruction and its result.

(1) Another function of the OP STEP switch is to stop the Trainer when it is in the High-Speed Run mode. Depressing the OP STEP switch will stop the Trainer after it completes the next A sequence.

c. PHASE STEP. - The Phase-Step mode allows one clock phase to be produced each time the PHASE STEP switch is depressed. To enter the Phase-Step mode, momentarily depress the OP STEP switch. (Exit from high-speed operation.) The HIGH SPEED DISCONNECT switch must be in the up position. This will disable the generation of phase pulses at the normal rate. Now, each time the PHASE STEP switch is depressed, one clock phase will be generated, and the operator can view the results of all commands associated with the clock phase. The PHASE indicators, one, two, three, and four will light in turn as successive clock phases are generated. During the Phase-Step mode of operation, the memory timing chain is disabled. This prevents any memory reference, so such data must be manually inserted via the SET pushbuttons. The DISCONNECT HIGH SPEED switch must be returned to the neutral position before High-Speed or Op-Step mode can be used.

d. REPEAT. - The Repeat mode causes the instruction in the U register to be executed repeatedly. The main function of this mode of operation is to load the Bootstrap and Inspect and Change programs. This is accomplished by manually inserting the instructions into the Q register and executing a Store Q Repeat instruction. The contents of the Q register will be stored at consecutive addresses for each depression of the OP STEP switch. The Repeat mode is entered by the following procedure:

1. Master clear and Enable D sequence.
2. Set the REPEAT switch to the up position. Red indicator should light.
3. Set the U register to 74. (Store Q.)
4. Set the instruction to be stored in Q.
5. Set the S register to the address receiving the contents of Q.
6. Set the P register to S + 1.

If the OP STEP switch is depressed, the contents of the Q register will be stored in the address in S. Also, the S register is incremented by one. This causes the instructions in Q to be stored in consecutive addresses each time the OP STEP switch is depressed.

e. SELECT JUMP. - The SELECT JUMP switch provides manual intervention in a program, if desired. This is accomplished by executing a jump instruction (f = 15) with a j designator of one. If the SELECT JUMP switch is in the up position, the program will jump to the address designated by the lower nine bits of the instruction. If the SELECT JUMP switch is in the neutral position the program will ignore the Jump instruction and execute the next sequential instruction.

f. SELECT STOP. - The SELECT STOP switch provides manual intervention in a program, if desired. This is accomplished by executing a Stop instruction (f = 15) with a j designator of three. If the SELECT STOP switch is in the up position, the program will jump to the address designated by the lower nine bits

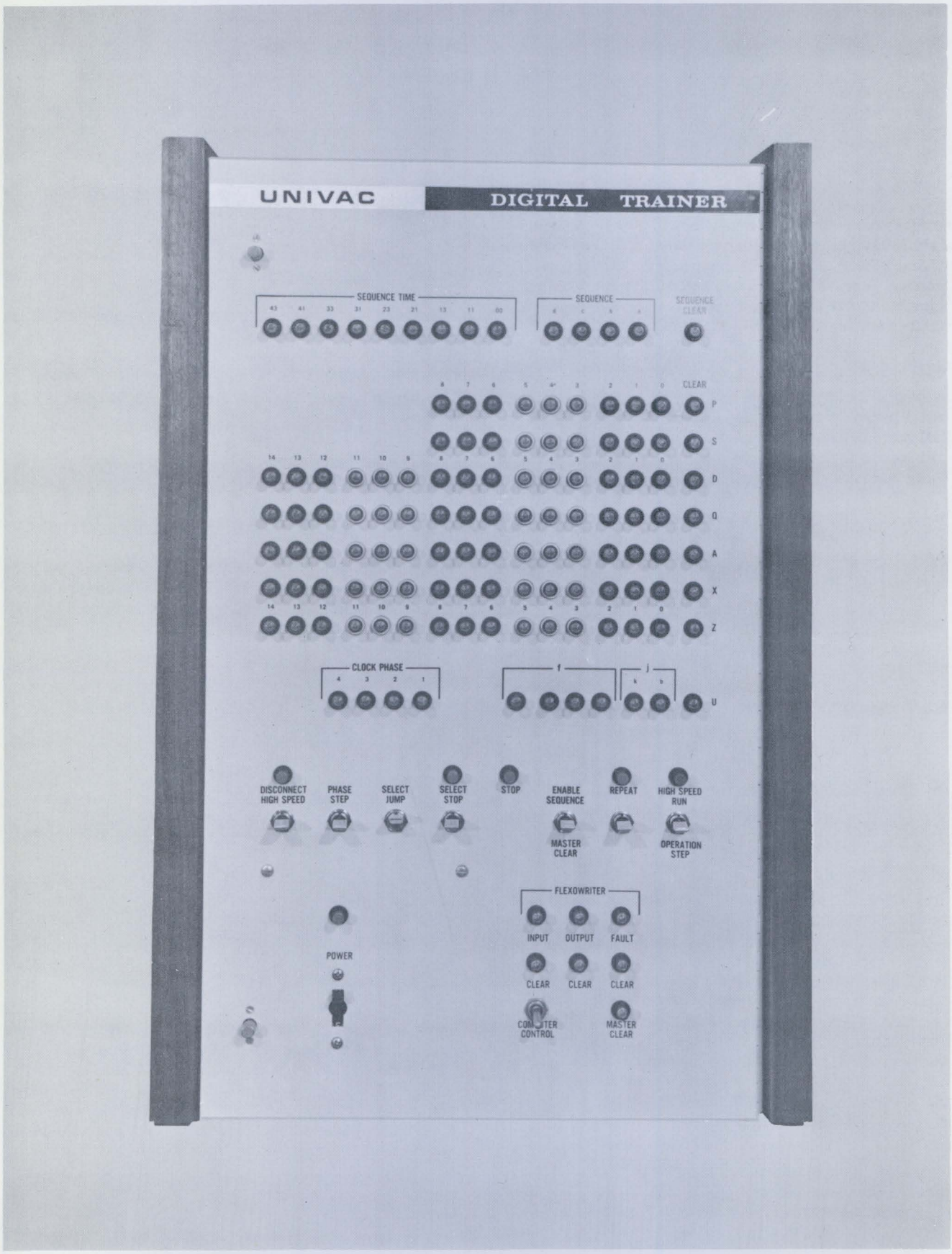


Figure 3-1. Operator's Console

of the instruction, and stop. The corresponding red indicator will light indicating that the program has jumped and stopped. After the Trainer stops, it may be started again by momentarily setting the HIGH SPEED-OP STEP switch in the up position. If the SELECT STOP switch is in the neutral position, the program will jump to the address designated by the lower nine bits of the instruction, but the Trainer will not stop.

g. COMPUTER CONTROL SWITCH. - The Trainer and Flexowriter operate independently of each other when this switch is in the up position. This allows the Flexowriter to be used in the off-line mode for preparing paper tapes. When the switch is in the down position, the Flexowriter is an input/output device and is controlled by the Trainer.

h. FLEXOWRITER CONTROLS. - Fault, Output, and Input modes all have SET and CLEAR buttons along with a MASTER CLEAR button which will clear all Flexowriter controls, The Output mode includes all outputs from the Trainer. The Input mode includes all inputs to the Trainer. The FAULT indicators light only because of a programmed fault. The Fault, Input, and Output modes are all controlled by using the External Function instruction (f = 12, j = 3). See the repertoire of instructions for the External Function codes.

i. REGISTER CONTROLS. - The remaining array of indicators and buttons on the console are associated with registers, designators, and sequences of the Trainer. Each register of the Trainer is represented by a block of indicators. Any stage of a register can be set by depressing the associated indicator. This will set the flip-flop, and light the indicator. Depressing the button on the right side of a register will clear all of the stages of that register. The sequence timing chain and sequence designators can be set and cleared in a like manner.

j. STOP INDICATOR. - When a stop instruction (f = 15, j = 2) is executed the program will jump to the address designated by the lower nine bits of the instruction and stop. The STOP indicator will light indicating the program has jumped and stopped. This is the normal Stop instruction to use for the termination of a program.

TABLE 3-1. CONSOLE CONTROLS AND INDICATORS

	POSITION	INDICATOR	ACTION
POWER	Up (L)*	Red	Applies 60 cycle, 110V power to Trainer.
MASTER CLEAR-	Down (M)*	--	Clears all registers, etc.
ENABLE SEQUENCE	Up (M)	--	Enables the ~~A sequence~~ TIMING CHAIN.
HIGH SPEED- OPERATION STEP	Up (M)	Run (green)	Initiates normal High-Speed operation.
	Down (M)	Run (green)	Executes one instruction each time the OP STEP switch is depressed.
REPEAT	Up (L)	Red	Causes the instruction in the U register to be repeated each time the OP STEP switch is depressed. The S register is incremented by one.
DISCONNECT HIGH SPEED	Up (L)	Red	Disables Master Clock.
PHASE STEP	Down (M)	Phase 1, 2, 3, 4	Generates one clock phase each time the switch is depressed.
SELECT JUMP	Up (L)	--	If f = 15 and j = 1, jump to address in Y.
SELECT STOP	Up (L)	Red	If f = 15 and j = 3, jump to address in Y and stop.
COMPUTER CONTROL	Up (L)	--	Isolates Flexowriter from Trainer.

*(M) - Momentary position
*(L) - Locking position

3-3. MANUAL READ AND WRITE.

The following procedures for reading and writing manually can be used by the programmer in performing operations used in loading, testing, and debugging a program. Manual operations involve principally the correction and checking of the stored program being run. Although the following procedures can be used, it is not intended that they be adhered to rigidly for all cases of program correction and checking. In the following paragraphs on reading and writing manually, the Q register was chosen for illustration. The same operations can be accomplished by using instructions which reference the A register. Special subroutines such as the Inspect and Change routine also provide considerable flexibility for program checking and correction. (See 3-4, Programming.)

a. MANUAL READ. - Reading manually involves the use of the Enter Q instruction (f = 02), or Enter A instruction (f = 01).

(1) ENTER Q (f = 02). When reading from address to Q, transfer information from one address of core storage to the Q register as follows:

STEP 1. Place the MASTER CLEAR switch momentarily in the down position.

STEP 2. Enable the D sequence.

STEP 3. Set the U register with f = 02, k = 1, and b = 0.

STEP 4. Set the S register equal to the address of the word to be entered in Q.

STEP 5. Place the OPERATION STEP - HIGH SPEED switch momentarily in the down position.

With step five completed, the Trainer stops, and the desired word is available for inspection in the Q register.

(2) ENTER Q (f = 02). When reading from consecutive addresses to Q, transfer information from consecutive memory addresses to the Q register as follows:

STEP 1. Place the MASTER CLEAR switch momentarily in the down position.

STEP 2. Enable the D sequence.

STEP 3. Place the REPEAT switch in the up (locking) position. The red REPEAT indicator should light.

STEP 4. Set the U register with f = 02, k = 1, and b = 0.

STEP 5. Set the S register equal to the address of the word to be entered in Q.

STEP 6. Set the P register equal to the S register plus one.

STEP 7. Place the OPERATION STEP - HIGH SPEED switch momentarily in the down position.

At the completion of step seven, the desired word has been transferred from core storage to the Q register, and the S and P registers have been advanced by one. Consecutive words will now be read up from memory with each momentary depression of the OPERATION STEP switch.

b. MANUAL WRITE. - The procedure for manually entering a word into storage can be varied to meet the needs of the programmer. The methods given here involve the use of the Store Q instruction, f = 17, k = Ø.

(1) WRITE FROM Q TO ONE ADDRESS. - Write from Q into one address of core storage as follows:

STEP 1. Momentarily place the MASTER CLEAR switch in the down position.

STEP 2. Enable the D sequence.

STEP 3. Set the U register with f = 17, k = 0, and b = 0.

STEP 4. Set the S register to the desired storage address.

STEP 5. Set the word to be stored into the Q register.

STEP 6. Momentarily place the OPERATION STEP - HIGH SPEED switch in the down position.

When step six is completed, the word in Q will have been transferred to the desired memory location.

(2) WRITE FROM Q TO CONSECUTIVE ADDRESSES. - Write from Q into consecutive addresses of core storage as follows:

STEP 1. Place the MASTER CLEAR switch momentarily in the down position.

STEP 2. Enable the D sequence.

STEP 3. Place the REPEAT switch in the up (locking) position. The red REPEAT indicator should light.

STEP 4. Set the U register with f = 17, k = 0, and b = 0.

STEP 5. Set the S register to the desired storage address.

STEP 6. Set the P register equal to the S register plus one.

STEP 7. Set the word to be stored into the Q register.

STEP 8. Momentarily place the OPERATION STEP - HIGH SPEED switch in the down position.

When step eight is completed, the word in Q will have been transferred to the desired memory location, and the S register and P register have been advanced by one. By repeating steps seven and eight, words will be stored in consecutive addresses of memory.

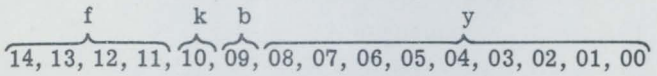
3-4. PROGRAMMING.

a. INTERNAL PROGRAMMING CONCEPTS. - Once the program is written and coded in an acceptable

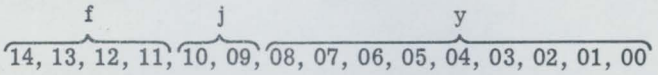
form, it is entered into the storage section of the Trainer. From this point, the Trainer, upon proper initiation executes the instructions of the program. The instructions are stored in sequential addresses of memory and are executed sequentially, beginning at the lowest address. From its address location, the instruction is moved to the control section, where the computer translates the instruction to determine the method of execution.

b. INSTRUCTION WORD. - All operations in the Trainer are controlled by a 15-bit instruction word. The 15 bits of the instruction word can be noted as i14, i13, i12, ... i02, i01, i00. Different bits of the Instruction word perform different functions, and have been assigned symbols for ease of reference. All of the instructions of the repertoire fit one of the two instruction formats.

Form I (Function codes of 00 through 05, and 17.)



Form II (Function codes of 06 through 16.)



The instruction word designators or symbols perform the following function:

(1) f - Function Code Designator (Bits 14-11). - The f designator in a Form I instruction determines the type of instruction to be executed (Enter, Add, Subtract, etc.). Form I instructions have function codes of 00 through 05, and 17. The f designator in a Form II instruction determines a group of four instructions. For example, an f designator of 14 selects four Arithmetic Jump instructions. A further selection of one of these four instructions is the function of another designator. Form II instructions have function codes of 06 through 16.

(2) j - Function Code Modification Designator (Bits 10-09). - The j designator is used only in Form II instructions. Its function is to select one of a group of four instructions already selected by the f designator. For example, if the f designator is 14, the instruction to be executed is some type of arithmetic jump. The j designator then selects one of the following four Arithmetic Jump instructions:

j = 00 - A ≠ 0 jump
j = 01 - A = 0 jump
j = 10 - A negative jump
j = 11 - Q negative jump

} Arithmetic Jump instructions

(3) k - Operand Interpretation Designator (Bit 10). - The k designator, in Form I instruction, determines the source of the operand. If k = 1, then Y is the address of the operand. If k = 0, then Y is the operand.

(4) b - Address Modification Designator (Bit 09). - The b designator, in Form I instructions, determines if Y will or will not be modified. If b = 1, the contents of the B register will be added to Y. If b = 0, Y will not be modified.

(5) Y - Operand Designator (Bits 08-00). - The function of the Y designator is relative to the b and k designators. The b designator determines if Y will be modified. The k designator determines if Y is the operand or the address of the operand. For instructions with a j designator, Y is the operand, or the address of the operand, depending upon the value of j, and the Function code.

(6) Symbol Conventions. - The following symbols are used throughout the descriptive material and repertoire of instructions:

A	-	The A register or Accumulator (15-bit Arithmetic register).
A_n	-	The nth bit of the A register.
B	-	The B or Index register. Located at memory address 0.
D	-	The D register (15-bit Arithmetic register).
Q	-	The Q register (15-bit Arithmetic register).
S	-	The Memory Address register (nine bits).
P	-	The P or Program Address register (nine bits).
U	-	The U or Program Control register (six bits).
X	-	The X or Exchange register (15-bit Arithmetic register).
Z	-	The 15-bit nonaddressable Memory Exchange register.
( )	-	contents of
RNI	-	Read or execute the next instruction.
L(Y)(Q)	-	Bit by bit multiplication of Y_n and Q_n. Logical product.
Y	-	Indicates the operand is the lower nine bits of the instruction word (k = 0 and b = 0).
Y + B	-	Indicates the operand is the lower nine bits of the Instruction word, plus the contents of the B register (k = 0 and b = 1).
$\underline{Y}$	-	Indicates the operand is the contents of the address indicated by the lower nine bits of the Instruction word (k = 1 and b = 0).
$\underline{Y + B}$	-	Indicates the operand is the contents of the address indicated by the lower nine bits of the Instruction word, plus the contents of the B register (k = 1 and b = 1).

3-5. REPERTOIRE OF INSTRUCTIONS.

The Trainer is a self-modifying, single address machine. This means that the operand, or address required for the execution of an instruction, can be modified automatically during a programmed sequence. Modification is accomplished by adding the contents of the B register to the Y portion of the Instruction word. During the execution of a program, instructions are read sequentially from magnetic core storage, except after Jump or Skip instructions. To be executed, an instruction must be transmitted from memory to the Z register, and then to the U register. While the instruction is in the U register, it is translated to enable the commands needed to execute the instruction. See table 3-2 for Instruction Repertoire.

TABLE 3-2. INSTRUCTION REPERTOIRE

CODE	INSTRUCTION	CODE	INSTRUCTION
00	Enter Logical Product	12 j = 0	Input
01	Enter A	12 j = 1	Output
02	Enter Q	12 j = 3	External Function
03	Selective Set	13 j = 0	Address Substitute Q
04	Add	13 j = 1	Complement Q
05	Subtract	13 j = 2	Address Substitute A
06 j = 0	Store Y in B	13 j = 3	Complement A
06 j = 1	B Register Index on Ø	14 j = 0	A ≠ 0 Jump
06 j = 2	Enter B Register	14 j = 1	A = 0 Jump
06 j = 3	B Register Index on Y	14 j = 2	A Negative Jump
07 j = 1	Store B Register	14 j = 3	Q Negative Jump
07 j = 2	Return Jump	15 j = 0	Unconditional Jump
10 j = 0	Right Shift Q	15 j = 1	Select Jump
10 j = 1	Right Shift A	15 j = 2	Stop
10 j = 2	Right Shift AQ	15 j = 3	Select Stop
10 j = 3	Multiply	16 j = 0	Input Jump
11 j = 0	Left Shift Q	16 j = 1	Output Jump
11 j = 1	Left Shift A	16 j = 2	Interrupt Jump
11 j = 2	Left Shift AQ	17 k = 0	Store Q
11 j = 3	Divide	17 k = 1	Store A

The instructions for the Trainer follow. To simplify explanation, consider Y the operand, regardless of its source.

00 Enter Logical Product L(Y) (Q)→A

The logical product of Y and (Q) is derived and entered into the A register. The k and b designators will determine the source of Y. A logical product is simply the ANDing of "1" bits. Hence, if Y_n and Q_n are both "1's", A_n will receive a "1". Any other combination of Y_n and Q_n will cause A_n to receive a "0".

01 Enter A (Y)→A

The A register will be cleared and Y will be entered into A. The k and b designators will determine the source of Y.

02 Enter Q (Y)→Q

The Q register will be cleared and Y will be entered into Q. The k and b designators will determine the source of Y.

03 Selective Set Set A_n if Y_n = 1

The A register is not cleared prior to the start of this operation. A selective set is simply the ORing of "1" bits. Hence, if Y_n OR A_n is a "1", A_n will be a "1". If Y_n and A_n are both "0", A_n will be a "0". The k and b designators will determine the source of Y.

04 Add (A) + Y→A

Add Y to the previous contents of the A register, and put the sum into the A register. The k and b designators will determine the source of Y.

05 Subtract (A) - Y→A

Subtract Y from the previous contents of the A register, and put the difference into the A register. The k and b designators will determine the source of Y.

06 j = 0 Store Y in B

The lower nine bits of this instruction word will be stored in the B register.

06 j = 1 B Register Index (Decrement)

If B - Y will equal Ø, the next instruction will be skipped, and the subtraction will not occur. "B final" will be the same as "B initial". If B - Y will not equal Ø, the difference will be stored in the B register, and the next instruction will be executed. Y is the lower nine bits of the Instruction word.

06 j = 2 Enter B Register (Y)→B

The contents of address Y will be entered into the B register. Y is the lower nine bits of the Instruction word.

06 j = 3 B Register Index (Increment)

Skip the next instruction if B = Y, and add one to the contents of the B register. Execute the next instruction if B ≠ Y, and add one to the contents of the B register. Y is the lower nine bits of the instruction word.

07 j = 1 Store B at Y

The contents of the B register will be stored at address Y. Y is the lower nine bits of the Instruction word.

07 j = 2 Return Jump

The contents of the P register will be stored in the lower nine bits of address Y. The next instruction to be executed is at address Y + 1.

10 j = 0 Right Shift Q

The Q register will right shift Y places, open ended, with sign extension. The shift count is contained in the lower four bits of Y. The upper five bits of Y is ignored. Maximum shift count is decimal 15.

10 j = 1 Right Shift A

The A register will right shift Y places, open ended, with sign extension. (For details, see Right Shift Q.)

10 j = 2 Right Shift AQ

AQ, as a 30-bit, double-length, register will right shift Y places, open ended, with sign extension. The lower bit of A will shift into the upper bit of Q. (For details, see Right Shift Q.)

10 j = 3 Multiply Q (Y)→AQ

The product of Q times the contents of Y will enter the 30-bit, double-length AQ register. Both the multiplicand (Y) and the multiplier (Q) must be positive numbers. The leftmost bit of A will contain the sign of the product.

11 j = 0 Left Shift Q

The Q register will left shift Y places. This is a circular shift. The 2^{14} bit of Q will enter the 2^0 bit position. The shift count is contained in the lower four bits of Y. The upper five bits of Y is ignored. Maximum shift count is decimal 15.

11 j = 1 Left Shift A

The A register will left shift Y places. This is a circular shift. (For details, see Left Shift Q.)

11 j = 2 Left Shift AQ

AQ, as a 30-bit, double-length register, will left shift Y places. This is a circular shift. The 2^{14} bit of A will enter the 2^0 bit of Q and the 2^{14} bit of Q will enter the adjacent 2^0 bit of A. (For details, see Left Shift Q.)

11 j = 3 Divide

The 30-bit, double-length dividend in the AQ register will be divided by the contents of Y. The quotient will enter the Q register and the remainder will enter the A register. Both the dividend (AQ) and the divisor (Y) must be positive numbers. The 2^{14} bit of A is the sign of the 30-bit dividend in AQ. To divide only the contents of Q by Y, the A register must first be cleared.

12 j = 0 Input

The external equipment will input to the Q register on channel Y. Input on channel zero will load the lower six bits of the Q register. Input on channel one will load all 15 bits of the Q register. The lower bit of Y selects the desired channel. The Trainer will generate an Input Acknowledge signal to the external equipment.

12 j = 1 Output

The Trainer will output from the Q register to the external equipment on channel Y. Before this instruction is executed, the Q register must be loaded with the data to be transmitted. This instruction will not load the Q register. Output on channel zero will transmit the lower six bits of the Q register. Output on channel one will transmit all 15 bits of the Q register. The lower bit of Y selects the desired channel. The Trainer will generate an Output Acknowledge signal to the external equipment.

12 j = 2 No Op Code

No instruction translation has been assigned to tnis code. The P register will be advanced normally. All registers will remain unchanged.

12 j = 3 External Function

Transmission from the Q register to the external equipment will occur, accompanied by the External Function signal. The External Function signal identifies the information bits from the Q register as a function code to the external equipment. The lower bit of Y selects the channel upon which the External Function will occur. The Flexowriter unit is connected to channel zero. One of a variety of peripheral equipments may be

connected to channel one. Before this instruction is executed, the Q register must be loaded with the desired External Function code format. For the Flexowriter unit it is as follows:

Q REGISTER BIT NUMBERS

5	4	3	2	1	0
Enable Input	Enable Output	Master Clear Flexowriter Unit	Disable Input	Disable Output	Fault

13 j = 0 Address Substitute Q

The lower nine bits of the Q register will be stored at address Y. The upper six bits of address Y will remain unaltered.

13 j = 1 Complement Q

The contents of the Q register will be complemented. "1" bits of Q will be switched to "0", and "0" bits will be switched to "1". The lower nine bits of the instruction (Y) is ignored.

13 j = 2 Address Substitute A

The lower nine bits of the A register will be stored at address Y. The upper six bits of address Y will remain unaltered.

13 j = 3 Complement A

The contents of the A register will be complemented. "1" bits of A will be switched to "0", and "0" bits will be switched to "1". The lower nine bits of the instruction (Y) is ignored.

14 j = 0 A Not Zero Jump

If the contents of the A register is zero, execute the next sequential instruction. If the contents of the A register is not zero, execute the instruction at address Y. Y is the lower nine bits of the Instruction word.

14 j = 1 A Zero Jump

If the contents of the A register is not zero, execute the next sequential instruction. If the contents of the A register is zero, execute the instruction at address Y. Y is the lower nine bits of the Instruction word.

14 j = 2 A Negative Jump

If the A register contains a positive number, execute the next sequential instruction. If the A register contains a negative number, execute the instruction at address Y. Y is the lower nine bits of the Instruction word.

14 j = 3 Q Negative Jump

If the Q register contains a positive number, execute the next sequential instruction. If the Q register contains a negative number, execute the instruction at address Y. Y is the lower nine bits of the Instruction word.

15 j = 0 Unconditional Jump

The normal sequence of instructions is interrupted, and the lower nine bits (Y) of the instruction is the address of the next instruction to be executed.

15 j = 1 Select Jump

If the SELECT JUMP switch is not selected, the Trainer will execute the next sequential instruction. If the SELECT JUMP switch is selected (up position), the normal sequence of instructions is interrupted, and the lower nine bits (Y) of the instruction is the address of the next instruction to be executed.

15 j = 2 Stop

The Trainer will jump to the address in the lower nine bits of the instruction (Y) and stop after completing the A sequence. This is the normal program stop, and the red STOP indicator will light. The Trainer will read up the instruction at address Y, but will not execute this instruction. When High Speed Run is initiated, the Trainer will execute the instruction at address Y.

15 j = 3 Select Stop

If the SELECT STOP switch is not selected, the Trainer will jump to the address in the lower nine bits of the instruction (Y), and execute the instruction. If the SELECT STOP switch is selected (up position), the Trainer will jump to the address in the lower nine bits of the instruction (Y) and stop after completing the A sequence. The red indicator above the SELECT STOP switch will light. The Trainer will read up the instruction at address Y, but will not execute this instruction. When High Speed Run is initiated, the Trainer will execute the instruction at address Y.

16 j = 0 Input Jump

If the external equipment has generated an input request, execute the next sequential instruction. If no input request has been generated, the lower nine bits (Y) of the instruction is the address of the next instruction to be executed.

16 j = 1 Output Jump

If the external equipment has generated an output

request, execute the next sequential instruction. If no output request has been generated, the lower nine bits (Y) of the instruction is the address of the next instruction to be executed.

16 j = 2 Interrupt Jump

If the external equipment has generated an interrupt, execute the next sequential instruction. If no interrupt has been generated, the lower nine bits (Y) of the instruction is the address of the next instruction to be executed.

17 k = 0 Store Q

The contents of the Q register will be stored in the memory address selected by the lower nine bits (Y) of the Instruction word. The b designator may be used (b = 1) to modify Y.

17 k = 1 Store A

The contents of the A register will be stored in the memory address selected by the lower nine bits (Y) of the Instruction word. The b designator may be used (b = 1) to modify Y.

3-6. OCTAL INSTRUCTION WORD.

The Instruction word of the Trainer consists of 15 binary bits, and, therefore, can be expressed in 5 octal digits. It is convenient to express the contents of all of the Trainer registers in octal. In order to do this, it is necessary to regroup the Form I and Form II instructions shown in paragraph 3-4, b.

a. FORM I. - An instruction that would add the contents of address 250 to the contents of the A register would be coded as follows:

f = 04, k = 1, b = 0 and y = 250

In binary, this instruction would be:

f k b y

0 1 0 0 1 0 0 1 0 1 0 1 0 0 0

By regrouping the binary bits of this instruction, it can be expressed in octal as:

2 2 2 5 0

0 1 0 0 1 0 0 1 0 1 0 1 0 0 0

It should be obvious that any of the Form I instructions can be expressed in octal by expressing the six binary bits that make up the f, k and b designators as two octal digits.

b. FORM II. - An instruction that would store the lower nine bits of the A register into address 375 would be coded as follows:

f = 13, j = 2, and y = 375

In binary, this instruction would be:

f j k

1 0 1 1 1 0 0 1 1 1 1 1 1 1 0 1

By regrouping the binary bits of this instruction, it can be expressed in octal as:

5 6 3 7 5

1 0 1 1 1 0 0 1 1 1 1 1 0 1

It should be obvious that any of the Form II instructions can be expressed in octal by expressing the six binary bits to make up the f and j designators as two octal digits.

c. For programming purposes, it is most convenient to have the repertoire of instructions in an octal format as shown in table 3-3. This eliminates the necessity of assembling the instructions in designator format, and then regrouping the bits in octal.

3-7. PROGRAMMING FUNDAMENTALS.

Any problem that can be solved by mathematical procedures can be solved by a computer. If it is determined that the problem can be best solved by a computer, it is then necessary to formulate the problem in computer language. A program of instructions and the data needed for the solution of the problem must be devised.

a. FLOW DIAGRAMS. - A flow diagram is helpful in facilitating the coding or programming of the problem. A flow diagram or flow chart indicates the flow, or steps, in the computation which leads to the solution of a particular problem. A basic flow diagram usually lists the series of simple arithmetic steps which are to be performed by a computer. It is imperative that the coder be thoroughly familiar with the over-all operations and the peculiarities of each computer instruction so he can construct the outline with regard to computer capabilities.

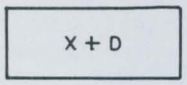
(1) Usually more than one flow diagram is formed for the more complicated problems. The first flow outline may be equations in mathematical language written in the sequence in which they will be computed, together with brief explanations of the steps involved. The second flow chart usually formulates the flow of computation in which the problem will be computed. This chart usually contains the instructions necessary for the data input, the instructions that operate on the data to obtain the solutions, and the necessary instructions for the output of the results. Many problems are such that the second type of flow diagram will consist of many charts and/or diagrams, each a more detailed presentation of the preceding charts.

(2) To make it easier to formulate and understand flow diagrams, certain symbols have been accepted. The following list of symbols (figure 3-2) gives the coder an example of the basic symbols used in drawing the second type of flow chart.

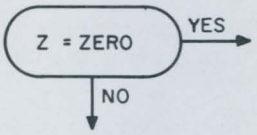
(a) LINES OF FLOW. - A solid line with an arrow touching the next element of the flow diagram usually is used to indicate the path to be followed by the computer; or more precisely, the path to be fol-

TABLE 3-3. REPERTOIRE OF INSTRUCTIONS (OCTAL FORMAT)

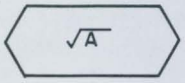
MNENOMIC	OCTAL	OPERATION
ENTLP	00	L(Q)Y → A
ENTLP b	01	L(Q)Y + B → A
ENTLP k	02	L(Q)$\overline{Y}$ + A
ENTLP kb	03	L(Q)$\overline{(Y + B)}$ → A
ENTA	04	Y → A
ENT Ab	05	Y + B → A
ENT Ak	06	$\overline{Y}$ → A
ENT Akb	07	$\overline{Y}$ + B → A
ENT Q	10	$\overline{Y}$ → Q
ENT Qb	11	Y + B → Q
ENT Qk	12	$\overline{Y}$ → Q
ENT Qkb	13	$\overline{Y}$ + B → Q
SELSET	14	Set A_N = 1 where Y_N = 1
SELSET b	15	Set A_N = 1 where $Y_N + B_N$ = 1
SELSET k	16	Set A_N = 1 where $\overline{Y}_N$ = 1
SELSET kb	17	Set A_N = 1 where $(\overline{Y + B})_N$ = 1
ADD	20	(A) + Y → A
ADD b	21	(A) + (Y + B) → A
ADD k	22	(A) + $\overline{Y}$ → A
ADD kb	23	(A) + $\overline{Y}$ + B → A
SUB	24	(A) − $\overline{Y}$ → A
SUB b	25	(A) − (Y + B) → A
SUB k	26	(A) − $\overline{Y}$ → A
SUB kb	27	(A) − $\overline{Y}$ + B → A
ENT B	30	Y → B
B DEC	31	B − Y, if B = 0 Skip NI and Bi = Bf if B ≠ Y, B − Y → B
ENT bk	32	$\overline{Y}$ → B
B INC	33	If B = Y Skip NI; If B ≠ Y NI; Add +1 to (B) in all cases
STR B	35	B → $\overline{Y}$
RJP	36	P → $\overline{Y}$: Execute Y + 1
RSHQ	40	Rt Shift (Q) by Y_{L4}
RSHA	41	Rt Shift (A) by Y_{L4}
RSHAQ	42	Rt Shift (AQ) by Y_{L4}
MUL	43	(Q)$\overline{Y}$ → AQ
LSHQ	44	Left Shift (Q) by Y_{L4}
LSHA	45	Left Shift (A) by Y_{L4}
LSHAQ	46	Left Shift (AQ) by Y_{L4}
DIV	47	(AQ)/$\overline{Y}$ Quotient → Q:R → A
INCN	50	Input from I/O → Q on CHY Bit 0
OUTCN	51	(Q) → I/O Equipment on CHY Bit 0
NO OP	52	Take NI
EXF	53	$(Q)_{L6}$ + Ext Function Bit → I/O Equipment on CH Y Bit 0
QLSU	54	Q_{L9} → $\overline{Y}$
CPQ	55	(Q) Complemented → Q
ALSU	56	A_{L9} → $\overline{Y}$
CPA	57	(A) Complemented → A
ANZJP	60	(A) ≠ 0 Jump to $\overline{Y}$.
AZJP	61	(A) = 0 Jump to $\overline{Y}$
ANGJP	62	(A) Neg Jump to $\overline{Y}$.
QNGJP	63	(Q) Neg Jump to $\overline{Y}$
UJP	64	Jump to $\overline{Y}$
SELJP	65	Jump to $\overline{Y}$ if Switch is Set
JP ST	66	Jump to $\overline{Y}$: Stop Operation
SEL ST	67	Jump to $\overline{Y}$ and Stop if Switch is set
INJP	70	Jump to $\overline{Y}$ if no Input Request
OUT JP	71	Jump to $\overline{Y}$ if no Output Request
INT JP	72	Jump to $\overline{Y}$ if no Interrupt Signal Exists
STR Q	74	(Q) → $\overline{Y}$
STR Qb	75	(Q) → $\overline{Y}$ + B
STR A	76	(A) → $\overline{Y}$
STR Ab	77	(A) → $\overline{Y}$ + B



Operation Symbol



Decision Symbol



Subroutine Symbol

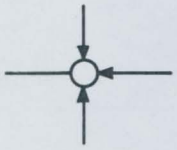
Entrance or Exit Symbol

Stop Symbol

Lines of Flow

Remote Connectors

Discontinuity Symbol



Junction Symbol (Use if there is more than one input to a symbol.)

Figure 3-2. Programming Symbols

lowed by the coder who is formulating the computer instructions from the flow diagrams.

(b) OPERATION SYMBOL. - The rectangular box usually contains a statement about a computer or mathematical operation. The content of the box may be a simple statement or a mathematical expression.

(c) DECISION SYMBOL. - An oval is used to indicate a two way decision. At this point in a program the computer must decide whether it will continue to execute instructions sequentially, or jump to a remote address and begin executing instructions.

(d) CONNECTORS AND REMOTE CONNECTORS. - To eliminate, as much as possible, crossing flow lines on a diagram, remote connectors are used to indicate a destination not easily reached in the diagram. Thus, the flow can be broken at a convenient point by terminating it in an arbitrary symbol which can be used to initiate the flow in another region of the diagram.

b. SUBROUTINE. - A subroutine is a portion of a routine that is complete in itself and can be isolated from the content of the larger main routine. It is a self-contained list of instructions for executing some particular operation, such as sin x, tan x, square root of x, etc. This routine should be coded in such a way that it may be used in a number of routines, wherever such a function is needed.

(1) SUBROUTINE ENTRANCE AND EXIT. - To utilize a subroutine, the main routine must be interrupted, and a jump provided to the correct entry point of the subroutine. The subroutine, in turn, must provide an exit back to the main routine. Both of these functions are accomplished by using a Return Jump instruction. Figure 3-3 is an example of how a Return Jump instruction provides an entrance to and exit from a subroutine. Note the commands generated by the Return Jump instruction at address 202 of the main routine.

(a) P, which is equal to 203, (address of the next instruction) is stored into the lower nine bits of Y, (700). The upper six bits of address 700 must contain an Unconditional Jump instruction = 64. This instruction at address 700 sets up the entrance back to the main routine.

(b) The computer then jumps to address 701 and begins executing the instructions of the square root calculation.

(c) After the square root is found, the computer does an unconditional jump to address 700.

(d) The unconditional jump at address 700 provides exit from the subroutine and entrance to the main routine.

It should be obvious that with such a subroutine the programmer can use the Return jump with Y = 700 any time his program requires a square root calculation. In a sense, it can be said that an instruction word coded 36700 is a square root instruction.

(2) FLOW CHARTING A SUBROUTINE. - It has been determined that a problem exists, and that the best possible way of solving this problem is by using a computer. In working the main program, it was discovered that a divide of two positive numbers was necessary, and that it would not be feasible to use the remainder in the answer. Therefore, it becomes necessary to round off the quotient to the nearest integer. PROBLEM: To round off the quotient resulting from a divide of two positive numbers. KNOWN: The divisor, the dividend, the quotient, and the remainder are all variables. ANALYSIS: By di-

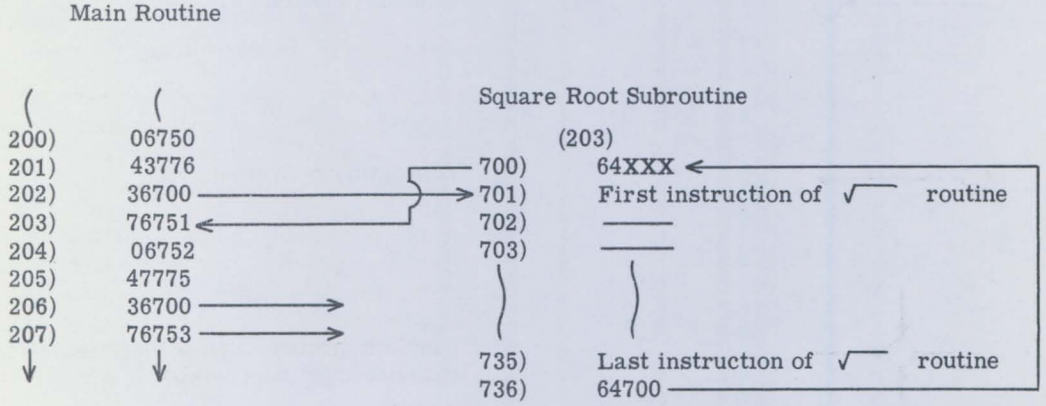


Figure 3-3. Return Jump Subroutine

viding the remainder into the divisor and then looking at this quotient to see if it is less than or equal to two, we can round off to the next highest integer. If the new quotient is greater than two, then leave the quotient of the original divide as it is.

The first step in writing a subroutine to make the roundoff is to lay out the flow diagram. It may be necessary to make several flow diagrams before preceeding with the next step. Figure 3-4 shows the basic operations necessary to accomplish quotient roundoff. This flow chart can be further amplified to that shown figure 3-5. Finally, from the detailed flow chart of figure 3-6, it is possible to code the operations in Trainer instruction format.

c. MACHINE CODING. - Each operation of the flow chart should be expressable by one instruction of the computer repertoire. If this is not possible, the flow chart must be analyzed and rewritten in more detail. After all of the operations of the routine have been expressed in instruction format, the only remaining task is to assign the instructions to their sequential addresses in computer memory. This assignment will depend upon what addresses are available or unused, and the programmer usually has considerable latitude in deciding where to store his program.

d. DEBUGGING A PROGRAM. - After the coded program has been written out completely, it must be put on some input medium (punched paper tape or magnetic tape) for loading into a computer. Once the program has been stored in computer memory, it can be executed. If the program is complex, or lengthy, it is

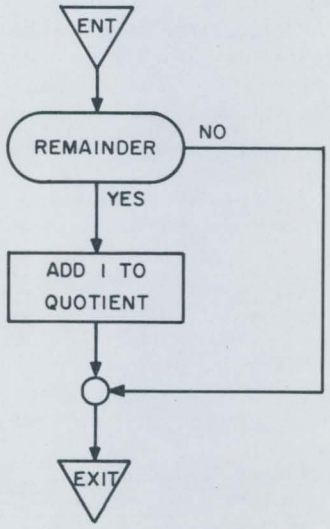


Figure 3-4. Quotient Roundoff (Preliminary)

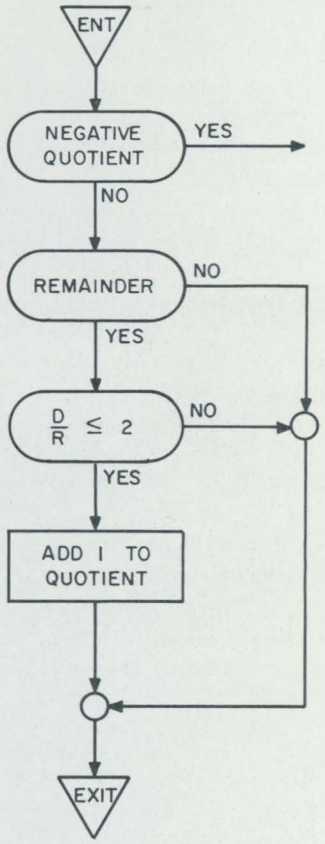


Figure 3-5. Quotient Roundoff (Secondary)

quite likely that it contains errors and will not run correctly the first time. Three common types of errors are:

(1) Errors introduced into the program during tape preparation.

(2) Coding errors created during the process of expressing each operation of the flow chart in computer instruction format.

(3) Logical Errors. Erroneous methods used to obtain the solution of a problem.

Debugging is the term applied to the process of locating and correcting errors in a program. Debugging a program usually involves a series of trial runs of the program on a computer. Each time the program fails to run properly the failure must be analyzed, and the error corrected. The methods that can be employed to debug a program vary depending on the nature of the programs, the service routine available, the availability of computer time, and the personal preference of the individual for one procedure over another. The following remarks suggest possible patterns to follow in debugging:

(1) TAPE PREPARATION ERRORS. - This type of error can be minimized by systematic checking of all input tapes before their information is read into a computer. This check can be made as follows:

(a) Use the Flexowriter to punch the tape from the programmer's manuscript.

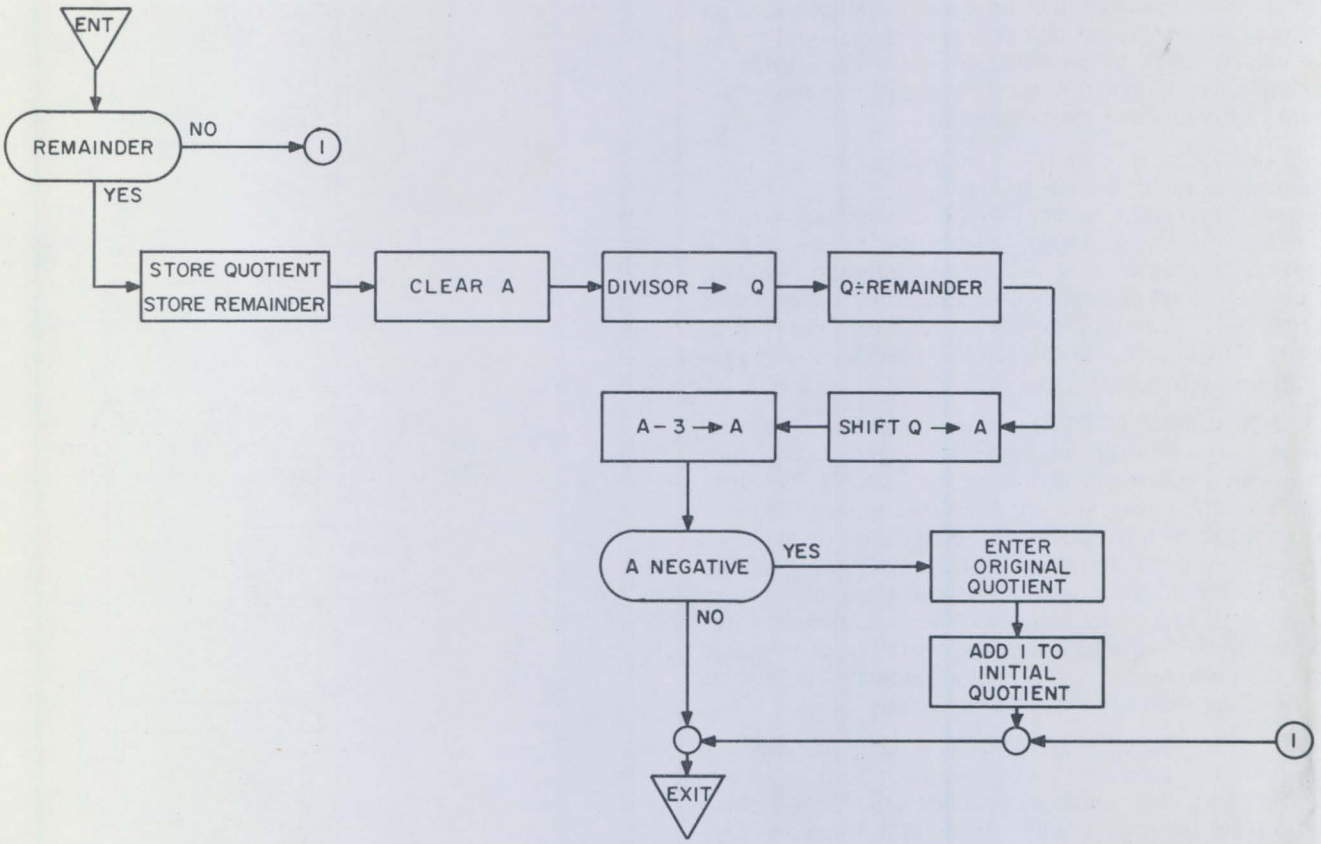


Figure 3-6. Quotient Roundoff (Detailed)

(b) Discard the resulting typed manuscript (hard copy) from the Flexowriter.

(c) Prepare a new typed manuscript from the punched paper tape.

(d) Compare this typed manuscript with the coder's manuscript to detect errors in punching.

(e) Correct all detected punching errors and prepare a new tape.

(f) Obtain from the corrected punched tape a new manuscript which should be retained by the coder to use while debugging.

(2) MANUAL DEBUGGING. - Manual debugging is the process used to locate errors in a program by controlling the operations of the program from the console and visually checking these operations as they occur and are indicated on the control panel. The following suggestions are intended to give the beginning student a starting point in developing his own debugging techniques.

(a) If the program is relatively short, the Inspect and Change routine can be used to examine all of the program instructions to see if the instructions have been properly stored in memory. Incorrectly stored instructions can be quickly located and easily changed using this routine. (See Paper Tape Control - Inspect and Change routine.)

(b) If the program is correctly loaded, but will not run properly, it is likely that the programmer has used an erroneous method in obtaining his solution. This can be detected by executing the program in the Operation Step mode. By executing one instruction at a time, and observing the contents of the registers, it is possible to detect when the program executes an unexpected operation. The programmer can then realize the mistake and make the necessary corrections.

3-8. TRAINER SOFTWARE.

Software is the term applied to the variety of programs or routines that aid in the utilization of the Trainer. Routines that compute such functions as Square Root X, Cube Root X, Sin X, Cos X, Tan X, and other often used mathematical functions can be referred to whenever necessary by the programmer. Other routines that perform input/output functions and assist in program debugging are not only convenient, but necessary tools at the disposal of the programmer.

a. INSPECT AND CHANGE. - The Inspect and Change routine causes the contents of a specified address to be entered into the Q register and the address of this data to be entered into the A register. The content of Q may then be changed manually if necessary. Q is then returned to the address from which it was taken, and the contents of A is advanced by one. The contents of the new address in A is then entered into Q to be inspected, and changed if necessary. This

routine can be used to inspect the instructions of a program to see if they are properly stored, and any discrepancies can be easily corrected. Short programs can be manually loaded with this routine. The instructions of the Inspect and Change routine are as follows:

Address	Octal Code	Remarks
001	20001	A + 1 → A
002	75000	Store Q at 000 + B
003	76000	A → B
004	13000	Enter Q with 000 + B
005	66001	Jump and Stop

To use the Inspect and Change routine, perform the following steps:

1. Master clear
2. Set P = 003
3. Set the first address to inspect in the A register.
4. Select High-Speed Run

When the Trainer stops (red STOP indicator lit), the Q register will contain the contents of the address in A. Selecting High-Speed Run will store Q back into the address in A, then increment A by one, and finally enter Q with the contents of the new address in A.

To load the Inspect and Change routine into addresses 001 - 005, use the procedure in 3-3, "Write from Q to Consecutive Addresses".

b. BOOTSTRAP. - This term applies to a routine that causes a program punched on paper tape to be loaded into the Trainer memory via the Flexowriter. The 13 instructions of this basic load routine may be set manually into the Trainer memory by means of the Inspect and Change procedure above, and are as follows:

Address	Octal Code	Remarks
006	76000	(A) → B
007	04010	Enter A with 00010
010	10000	Clear Q
011	70011	Jump if no input request
012	50000	Input
013	42003	Right Shift AQ - 3
014	61017	Jump if (A) = 0
015	46006	Left Shift AQ - 6
016	64011	Unconditional Jump
017	75000	Store Q at address Q + B
020	33777	Y = B Skip
021	64007	Unconditional Jump
022	66023	Jump and Stop

The Bootstrap routine loads paper tape programs punched in a special format termed Lemonoc which means Left-Most Octal Digit. In this format the first, or higher order, or left octal digit of a two digit Flex code is used to represent the (binary-octal) number. The lower digit of the Flex code is disposed of by shifting it off the right end of the Q register. The Flex codes used to represent the octal digits of instructions in Lemonoc format are as follows:

Flex Character	Flex Code	Octal Digit
T	01	0
R	12	1
E	20	2
W	31	3
=	44	4
+	54	5
6	66	6
7	72	7

With this format an instruction coded 61017 (jump to address 017 if A = 0) would be typed and punched as 6RTR7. Only the higher order digit of each Flex code is stored in memory. The Bootstrap routine is simple and short because no translation or conversion of the input data is needed.

When punching a paper tape in Lemonoc format, every five Flex codes punched will have their leftmost digits assembled to form an instruction. A stop code should be punched on the paper tape after the last instruction of the program is punched.

To load a program punched in Lemonoc format perform the following operations. (The Bootstrap routine must be stored in addresses 006 - 022 of the Trainer.)

1. Master clear the Trainer.
2. Master clear the Flex logic.
3. Set the paper tape in the reader with the first frame under the read head.
4. Set the Input mode. (White light on Flexowriter should light.)
5. Set the first address of the program into the A register.
6. Set the P register to 006.
7. Select High-Speed Run.
8. Select Start-Read on Flex.

The Trainer will load all of the instructions of the program in consecutive addresses and when the Stop code is read up the Flexowriter will stop. The Trainer will continue running in an input jump loop because no more data is being fed to the Trainer. Depressing the OPERATION STEP switch will stop the Trainer .

c. FLEX LOAD. - This routine is used to load a program punched in Flex format on paper tape into the Trainer memory via the Flexowriter. This standard load routine will produce a hard copy of the program as it is being loaded into the Trainer . The instructions of the Flex Load routine may be punched on paper tape in Lemonoc format and stored into the Trainer memory by the Bootstrap routine, and are as follows:

Address	Octal Code	Remarks
022	66023	Jump and Stop
023	61022	Jump if A = 0

Address	Octal Code	Remarks
024	10050	Enter Q with 50
025	53000	External Function
026	56042	Store AL9 in 042
027	70027	Jump if no input request
030	50000	Input
031	30000	Clear B
032	00077	L (Y) (Q) → A
033	27056	A - (Y + B) → A
034	61047	Jump if A = 0
035	33007	Skip NI if Y = B
036	64032	Jump
037	20025	A + 25 → A-
040	60022	Jump if A ≠ 0
041	06055	Enter A with (055)
042	76---	Store A
043	06042	Enter A with (042)
044	20001	A + 1 → A
045	64026	Jump
046	07777	Logical Product Mask
047	12046	Enter Q with (046)
050	02055	L (Y) (Q) → A
051	45003	Left Shift A - 3
052	15000	Selective Set A^n for $B^n = 1$
053	76055	Store A in 055
054	64027	Jump
055	-----	Word Assembly
056	00037	Flex Code 0
057	00052	Flex Code 1
060	00074	Flex Code 2
061	00070	Flex Code 3
062	00064	Flex Code 4
063	00062	Flex Code 5
064	00066	Flex Code 6
065	00072	Flex Code 7

When punching a paper tape in Flex format, the five characters punched before a carriage return will be assembled and stored in Trainer memory. A carriage return must be punched after every instruction. When the last instruction and carriage return is typed and punched, a period and Stop code should be punched at the end of the tape. When the tape is being loaded into the Trainer, the period will cause the Trainer to stop, and the Stop code will cause the Flexowriter to stop. During a Flex Load the Trainer will stop if any Flex code other than zero through seven or carriage return is inputted. The Flex codes are listed on table 3-4.

To load a program punched in Flex format perform the following operations. (The Flex load routine must be stored in addresses 022 - 065 of the computer.)

1. Master clear the Trainer.
2. Set the paper tape in the reader. (The first digit of the first instruction does not have to be under the read head.)
3. Set the first address of the program into the A register.
4. Set the P register to 023.
5. Select High-Speed Run.
6. Select Start-Read on Flex.

The Trainer will load all of the instructions of the program in consecutive addresses, and the period and Stop code punched at the end of the tape will cause the Trainer and Flexowriter to stop. The Flexowriter will also have printed a hard copy of the program as it was being loaded into the Trainer.

d. FLEX DUMP. - This routine will cause the Flexowriter to print and punch the contents of specific memory areas on paper tape in Flex code. The instructions of the Flex Dump routine may be punched on paper tape in Lemonoc format and stored into the Trainer memory by the Bootstrap routine.

Address	Octal Code	Remarks
022	66023	Jump and Stop
023	56000	AL9 → B
024	54051	QL9 → 051
025	10030	Enter Q with 30
026	53000	External Function
027	13000	Enter Q with B
030	35054	Store B at 054
031	30000	Clear B
032	44003	Left Shift Q - 3
033	74055	Store Q at 055
034	00007	L (Y) (Q) → A
035	20056	A + 56 → A
036	56037	AL9 → 037
037	12000	Enter Q with B
040	71040	Jump if No Output Request
041	51000	Output
042	12055	Enter Q with (055)
043	33004	Skip NI if Y = B
044	64032	Jump
045	10045	Enter Q with 45
046	71046	Jump if No Output Request
047	51000	Output
050	32054	Enter B with (054)
051	33---	Skip NI if Y = B
052	64027	Jump
053	64022	Jump
054	-----	Store Address
055	-----	Word Assembly
056	00037	Flex Code 0
057	00052	Flex Code 1
060	00074	Flex Code 2
061	00070	Flex Code 3
062	00064	Flex Code 4
063	00062	Flex Code 5
064	00066	Flex Code 6
065	00072	Flex Code 7

To dump a program or the contents of a block of addresses on paper tape in Flex code perform the following operations. (The Flex Dump routine must be stored in addresses 022 - 065 of the Trainer.)

1. Master Clear the Trainer.
2. Set the P register to 023.
3. Set the first address to be dumped into the A register.

TABLE 3-4. FLEXOWRITER CODES

FLEX LETTER		Octal	FLEX LETTER		Octal
Upper Case	Lower Case	Code	Upper Case	Lower Case	Code
A	a	30	1	1	52
B	b	23	2	2	74
C	c	16	3	3	70
D	d	22	4	4	64
E	e	20	5	5	62
F	f	26	6	6	66
G	g	13	7	7	72
H	h	05	8	8	60
I	i	14	9	9	33
J	j	32	0	0	37
K	k	36	-	-	56
L	l	11	.	=	44
M	m	07	/	+	54
N	n	06	(	,	46
O	o	03	)	.	42
P	p	15	-		50
Q	q	35	Space		04
R	r	12	Shift Up		47
S	s	24	Shift Down		57
T	t	01	Back Space		61
U	u	34	Carriage Return		45
V	v	17	Tab		51
W	w	31	Color Shift		02
X	x	27	Tab Feed		00
Y	y	25	Stop Code		43
Z	z	21	Delete		77

4. Set the last address to be dumped into the Q register.
5. Depress the Flexowriter PUNCH ON button.
6. Obtain six to eight inches of leader on the paper tape by momentarily depressing the TAPE FEED button.
7. Select High-Speed Run.
8. Select Start-Read on Flex.

When the contents of the last address have been punched, the Trainer and Flexowriter will stop. The contents of the addresses being dumped will be printed (hard copy) as they are being punched on paper tape.

SECTION 4

PRINCIPLES OF OPERATION

4-1. GENERAL.

This section describes the UNIVAC Digital Trainer in terms of the electrical characteristics, logic, and operation of the Trainer sections. The explanations include the functions and operations of the various registers, translators, and other special circuits; description of data flow; information on interdependence of sections; and an explanation of the sequence of operation.

a. DESCRIPTION. - The Trainer is made up of four sections: Control, Arithmetic, Input/Output and Memory. The sections are principally located as follows: Storage on the front right-hand chassis, Control section on the rear left-hand chassis, Arithmetic section on the front left-hand chassis, and the Input/Output or Flexowriter circuits on the rear right-hand chassis.

(1) CONTROL SECTION. - The control section of the Trainer is made up of registers, modifying circuits, and timing circuits. The parts function together, setting up timing sequences and other control operations necessary to successfully execute instructions.

(a) REGISTERS. - The following registers are located in the control section:

U register - a 6-bit program control register

P register - a 15-bit program address register

B register - a 15-bit index register located at address 000 in memory

(b) MODIFYING CIRCUITS. - The following modifying circuit is located in the control section:

The S adder - adds S + 0, +1 to P. Other operations employing this adder will be discussed later in this manual.

(c) TIMING SEQUENCES. - The timing sequences resulting from functions of the control section are as follows:

A sequence - initiates commands necessary for obtaining the next instruction word and performing preliminary operand modification.

B sequence - performs operand modifications and obtains the appropriate operand.

C sequence - initiates and controls arithmetic operations as prescribed by the current instruction word.

D sequence - stores the operand as specified by the current instruction word.

(2) ARITHMETIC SECTION. - The arithmetic section consists of the A, D, Q, and X registers, and a modifying circuit called the Adder. (The Adder combines the contents of the X and D registers.) The purpose and function of the arithmetic section is to perform addition, subtraction, multiplication, division, shifts, and logical operations according to instructions.

(a) REGISTERS. - The following registers are included in the arithmetic section:

A - a 15-bit addressable assimilator

Q - a 15-bit addressable logical function register

X - a 15-bit non-addressable exchange register

D - a 15-bit non-addressable exchange register

(3) INPUT/OUTPUT SECTION. - The input/output section serves as the communication link between the external equipment and the other Trainer sections. Communication with the Trainer is carried on in a 15-bit parallel mode, or a special channel using a 6-bit parallel mode to the Flexowriter. The input/output section is entirely dependent on program control to perform the transmission or reception of each word. It does not have the capabilities of performing automatic buffers or transfers to any external equipment.

(4) MEMORY SECTION. - The memory section is a magnetic core storage system. It is a high-speed, random access, nonvolatile storage system. That is, its speed of operation is compatible with the Trainer; its data words may be referenced in a nonsequential manner; and it retains data words through normal power sequencing.

(a) CAPACITY. - The memory section contains 512 locations (memory registers) for storage of 15-bit words. Each of the 512 memory registers is assigned a unique address.

(b) TIMING. - The time needed for one memory reference (basic memory cycle time) is eight microseconds. The readout time, or the time from which a given function assumes control of the memory minor sequence until delivery of the data from the memory, is approximately 2.5 microseconds.

(c) CONTROLS. - There are two control registers, S and the Z registers, associated with the memory. During a memory reference, the S register (Storage Address register) contains the 9-bit address word that specifies one of 512 memory registers. Data transmissions into, or out of, the selected memory register is channeled through the Z register.

b. OPERATION OF TRAINER SECTIONS. - Data is transmitted from external equipment into the Trainer and from the Trainer to the external equipment through the input/output section. The data received could be stored into Trainer memory or immediately processed depending on the controlling program. Processes such as addition, subtraction, multiplication, and division are performed by the arithmetic section. The control section regulates transmissions into and out of memory, and inputs and outputs according to the program of instructions. The control section also selects memory address locations and performs other special functions. Because of the interdependence of one section upon the other, no one section can be explained completely without introducing material which logically belongs in discussions of other sections. In the following paragraphs, reference is made to parts contained in, or functions performed in other Trainer sections. Complete explanation of the parts or functions generally are not contained at that point, but are left for a later discussion of the particular section. Explanation of the sequence of operations shows the flow of data through and between the Trainer sections. Functional schematics of the registers and other special circuits are contained in Section 5. Reference is made to the functional schematics in the following paragraphs of this section. Whether specific reference is made or not, the appropriate schematic should be used along with the text as an aid to understanding the special circuit.

4-2. BASIC CIRCUITS.

The UNIVAC Digital Trainer utilizes one basic circuit to make up the registers, timing chains, translators, etc. The only exception is in the memory section where special circuits are employed to supply proper voltages and currents required for a magnetic core storage.

Before the circuit can be examined, it should be noted that the voltage levels at the inputs to the circuits are zero volts, or a ground level, which represents a "0" and -3 volts which represents a "1". The circuit, in its most simple form, is shown in figure 4-1.

The characteristics of the NEGATIVE OR are: If any of the inputs are at a -3 volts level, the negative voltage will appear at the input to the inverter; but if all inputs are at a ground volt level, a positive output will be reflected to the input of the inverter. The applied input voltages to the inverter will be inverted. That is, if a "1" (-3 volts) was applied to input A, a negative voltage level would be reflected to the input of the inverter, which would invert it to produce a ground level or a "0" output. Table 4-1 shows the inputs in various combinations and the resultant outputs. This circuit is commonly called the NOR circuit.

TABLE 4-1. NOR CIRCUIT CHARACTERISTICS

A	B	C	R
1	1	1	0
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

a. ELECTRONIC ANALYSIS. - The basic circuit used in the Trainer is the NOR circuit, explained in block form above. Figure 4-2 (a), shows a typical circuit employing the diode gates and transistor inverter.

(1) INVERTER. - This particular circuit has three inputs and one output, while other circuits may have up to 10 inputs. The transistor, Q1, is a PNP type, and for conduction the base must be negative with respect to the emitter. The base of Q1 is connected to a voltage divider consisting of R7 and R10 between +15 VDC and ground. This places a positive potential on the base of Q1, and the transistor is cut off unless inputs are such that allow conduction. Assuming the transistor is cut off, the output will be -3 VDC ("1"). This -3 VDC output is provided by the clamping action of CR1 which does not allow its cathode to be more negative than the -3 VDC applied to the anode. Thus, when the transistor is cut off, the output is a -3 VDC. If the inputs are such that the transistor is conducting, the ground applied to the emitter is effective through the low resistance of the transistor, and the output is then ground ("0"). The inputs to Q1 are via CR4, CR5, and CR6. Assume the inputs to be -3 VDC to CR4 and 0 VDC to CR5 and CR6. The -3 VDC applied to the cathode of CR4 causes it to conduct; the voltage developed across R4 reverses the bias on CR5 and CR6 and they are cut off. The current through CR4, R4, and R7 also causes the voltage at the base of Q1 to go from approximately +0.7 VDC to a small negative voltage, allowing the transistor to conduct, and the output is ground. The transistor conducts any time one or more of the inputs is -3 VDC ("1"). The

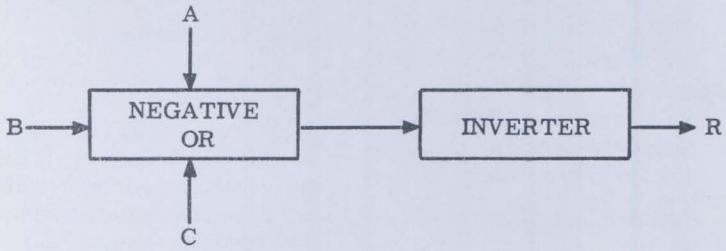


Figure 4-1. Block Diagram of the Basic Circuit

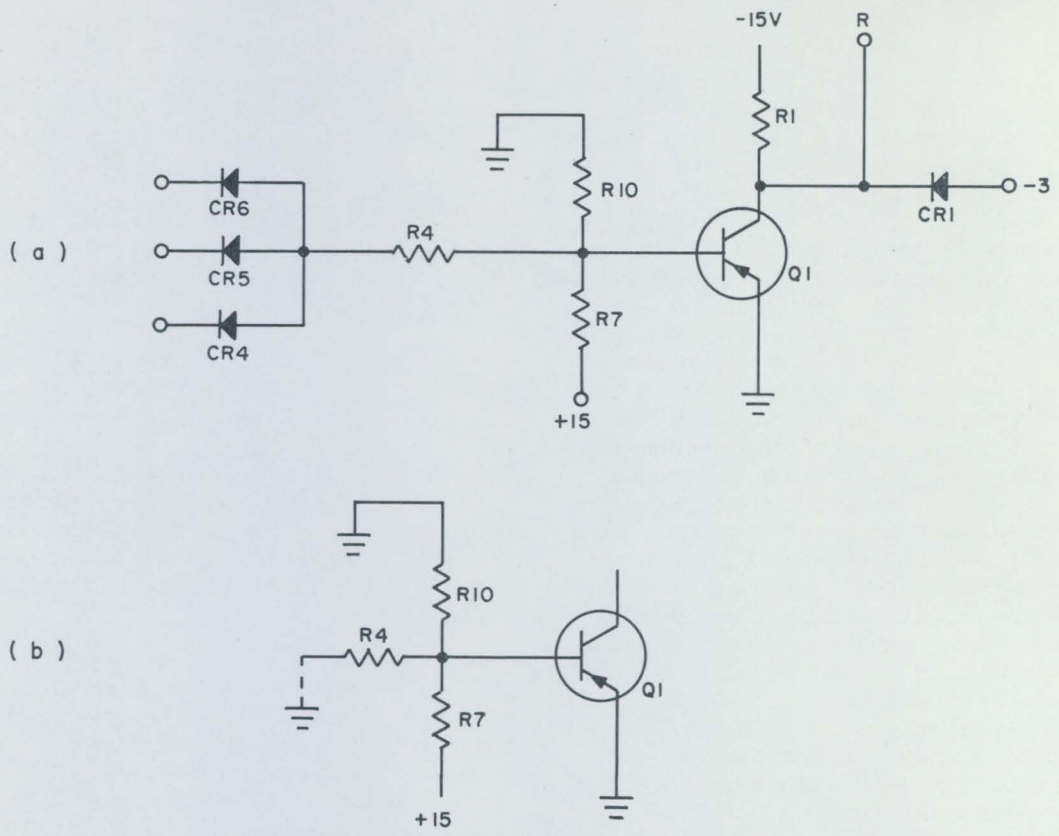


Figure 4-2. Diode Gate and Transistor Inverter

only time the transistor is cut off is when all inputs are ground. The R7 and R10 voltage divider applies a small positive voltage through R4 to the anodes of the diodes. If the cathodes are at ground they conduct a small amount, and R4 is effectively grounded as shown in figure 4-2 (b). This voltage divider now makes the potential of the base of Q1 about +0.7 VDC which will cut off the transistor, and the output is -3 VDC.

Two statements can summarize the results of the diode gate and transistor inverter circuit: (1) if any of the inputs is -3 VDC ("1") the transistor is conducting, and the output is ground ("0"); and (2) if all inputs are ground ("0") the transistor is cut off, and the output is -3 VDC ("1").

(2) FLIP-FLOP. - A flip-flop is made up of two inverters with the output of each connected as the input to the other. The purpose of this configuration is to produce a bistable element capable of storing binary numbers, in the case of registers, or produce enables to allow commands to take place at a definite time. With one inverter feeding into the input of another, one transistor will be conducting while the other is cut off. The transistors of a flip-flop are identified as the one side or the zero side. Figure 4-3 shows a flip-flop and its associated circuits. Q1 is the zero side and Q2 is the one side. For purposes of identification, each circuit has been given a code. The explanation of this code will be explained in the following paragraphs.

Assume a clear command has been initiated. The input to CR4 of 11D00 will be a "0" and at a time controlled by a master clock, the input to CR6 will rise from "1" to "0". Q1 is now cut off, and the output on pin seven is -3 VDC. This output is applied to CR4 of 00D00, the zero side of the flip-flop, which causes CR4 to conduct, and its output is ground. This output is applied to CR5 which is an input to 01D00, or the one side of the flip-flop. The other input, CR6, would be a "0" because the timing input applied to CR6, 13D00, is a "1". Q1 is conducting and its output is ground. Thus, all inputs to 01D00 are "0", the base of Q2 is a positive potential, and Q2 is cut off and has a -3 VDC output. This -3 VDC is applied to CR3 of Q1 which holds Q1 in the conducting state ("0" output), and Q2 in the cut off state ("1" output) until all inputs are applied to Q2. This state of the flip-flop is called the clear state.

Proceeding from the clear state, assume that, because of a command, the input to CR4 has changed to a "0". If the proper data is present the input on pin six, feeding CR5, will be a "0". The last input under control of the master clock, will go to "0". Under these conditions (all inputs to 13D00 are ground) Q1 is cut off, and the output of -3 VDC is applied to CR6 of 01D00, the one side of the flip-flop. This causes Q2 to conduct and its output is a "0" which is applied to CR3 of 00D00. Since timing is being controlled by a master clock, the input to 11D00 on pin 14 would be a "1" resulting in a "0" output from 11D00. With both inputs to 00D00 now a "0", Q1 is cut off and has a -3 VDC output to CR5 of 01D00 which maintains the flip-flop in this new state. That is, the one side (01D00-Q2) is conducting and has a ground ("0") output and

the zero side (00D00 - Q1) is cut off and has a -3 VDC ("1") output. This state is called the set state. The flip-flop will remain set until it is cleared.

(3) MANUAL SET. - The remaining circuit associated with this flip-flop is the Indicator Driver circuit, 99D00. The indicator driver has a direct input from the zero side of the flip-flop. When the flip-flop is set, this input is -3 VDC which causes Q1 (99D00) to conduct and supply a near ground potential on pin seven. This low potential is applied through interassembly wiring to pin three of the pushbutton indicator assembly located on the front maintenance panel. The neon lamp now has approximately 90 volts across it which is enough to cause the lamp to be ionized, thus giving an indication that the flip-flop is set. During the clear state of the flip-flop, Q1 (99D00) is cut off and the potential across the lamp is approximately 36 volts which is not enough to ionize the lamp. The 36 volt potential is the result of the -90 VDC on pin one of the indicator module and the -54 VDC is on pin two. Thus, the lamp is extinguished when the flip-flop is cleared. The other function of the indicator driver is to manually set the flip-flop when the pushbutton in the indicator module is depressed. This pushbutton is a physical part of the indicator assembly. Depressing the button causes the lamp to be ionized and a ground supplied through the interassembly wiring to the anode of CR4 (99D00). CR4 conducts and grounds the anode of CR1. The cathode of CR1 is connected to the output of Q2 (01D00). If the output of Q2 is grounded, nothing will happen because the flip-flop is already set. If, however, the output of Q2 is -3 VDC, CR1 (99D00) conducts and clamps the output of Q2 (01D00) at ground. Thus, the input to CR3 (01D00) is clamped at ground and the flip-flop is set. When the flip-flop is set, Q1 (99D00) conducts and furnishes the low potential to the lamp which allows it to remain ionized after the button is released.

(4) MANUAL CLEAR. - Also associated with each register and some sequences is a MANUAL CLEAR button. The connections for the CLEAR D pushbutton are shown in figure 4-4. The Manual Clear circuit is the same as the Manual Set circuit used with each flip-flop. However, the neon lamp is removed from the indicator module on the maintenance panel and there is no input to Q1 on the Indicator Driver circuit. Thus, pushing the MANUAL CLEAR pushbutton applies a ground through the interassembly wiring to CR4 (99D00). CR4 clamps the anode of CR1 at ground which clamps the output of 10D00 at ground. Thus, the input to 11D00, figure 4-3, is a "0" which is inverted at a time controlled by the master clock. The output of 11D00 is then applied to the zero side of the flip-flop as a -3 VDC and the flip-flop is cleared.

(5) SUMMARY. - The diode gate and transistor inverter is the building block of the Trainer. It is well to know the following fundamentals:

1. If all inputs to the gate are "0", the output from the transistor is a "1".

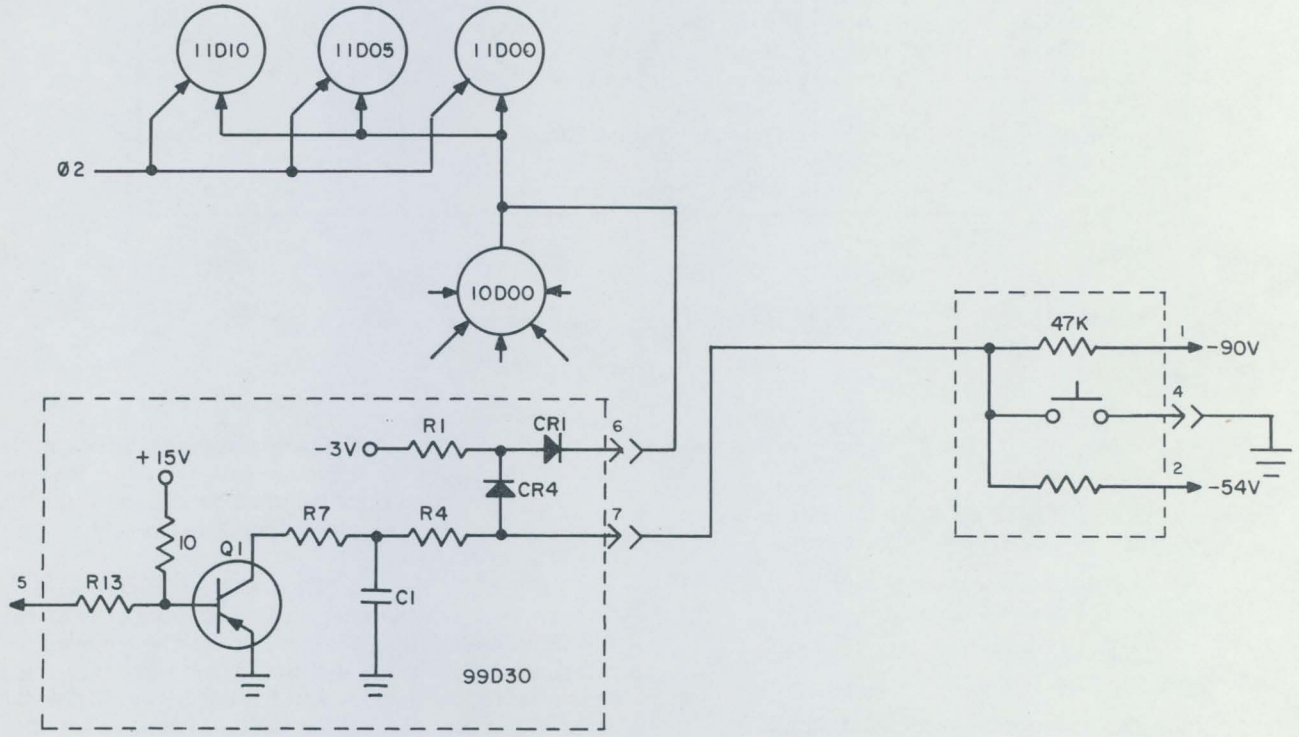


Figure 4-4. Manual Clear

TABLE 4-2. CROSS REFERENCE TO CARD SCHEMATIC DIAGRAMS

Figure No.	Card Type	Name	Number Used
4- 5	250010	Inverter	19
4- 6	250020	Inverter	3
4- 7	250040	Inverter	27
4- 8	250050	Inverter	11
4- 9	250070	Inverter	147
4-10	250080	Inverter	34
4-11	250090	Indicator Driver	45
4-12	250100	Clock Oscillator	1
4-13	250140	Chassis Driver Input	4
4-14	250150	Chassis Driver Output	2
4-15	250160	Clock Output	4
4-16	250170	Input Amplifier	9
4-17	250200	Lamp Driver	3
4-18	250240	Delay Line -3	1
4-19	250250	Delay Line -4	2
4-20	250270	Inhibit Switch	5
4-21	250280	Read/Write Driver	2
4-22	250290	Read/Write Current Generator	4
4-23	250300	Transformer Selector	2
4-24	250330	Sense Amplifier	15
4-25	250340	Gated Sense Amplifier	8
4-26	250370	Read/Write Transformer	3
4-27	250400	Inverter	1
4-28	250410	Inverter	99
4-29	250420	Flip-Flop	71
4-30	250430	Memory Driver	18
4-31	250690	Clock Amplifier	3
4-32	250700	Voltage Protection	1
4-33	250720	Punch Puller	9
4-34	250770	Pulse Delay	3
4-35	250910	Time Delay -4	3
4-36	265562	Positive Selector	16
4-37	265565	Negative Selector	16
4-38	265566	Diode Card	4
4-39	265677	Gate and Strobe Output	1
4-40	265678	Choke	10
4-41	265682	Inhibit Current Diverter	15

2. If any input to the gate is a "1", the output from the transistor is a "0".
3. A flip-flop is set when the zero side has a "1" output.
4. A flip-flop is cleared when the zero side has a "0" output.
5. The indicator lamp is ionized when the flip-flop is set.
6. The pushbutton associated with a flip-flop will set that flip-flop when depressed.
7. The pushbutton associated with a register or sequence will clear the associated circuits when depressed if the proper clock timing is present.

(6) SCHEMATIC DIAGRAMS. - Figures 4-5 through 4-41 are the schematic diagrams of each card assembly used in the Trainer. They are included for reference only. Table 4-2 is a cross reference list between the figure number and the card type. Those circuits shown that are not the basic circuits, have special applications and will be discussed in subsequent paragraphs.

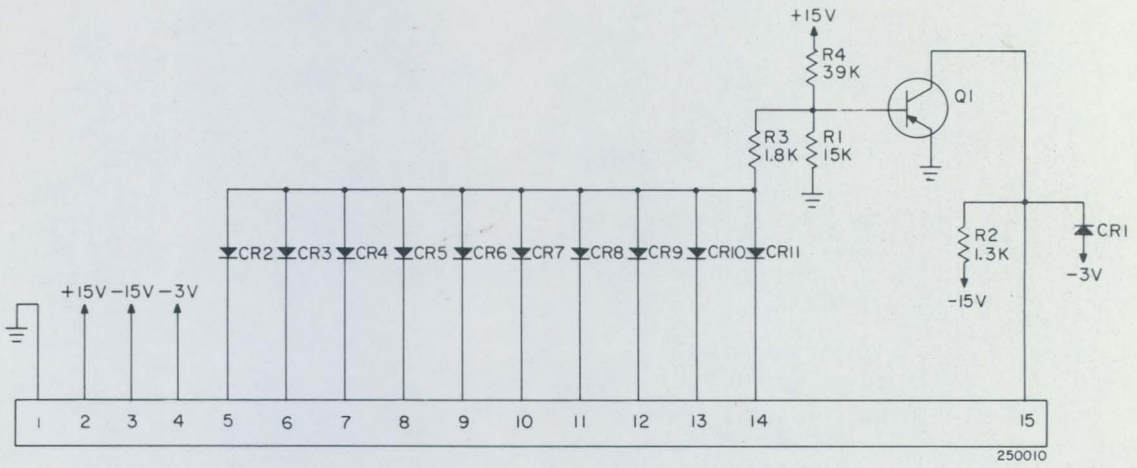


Figure 4-5. Inverter

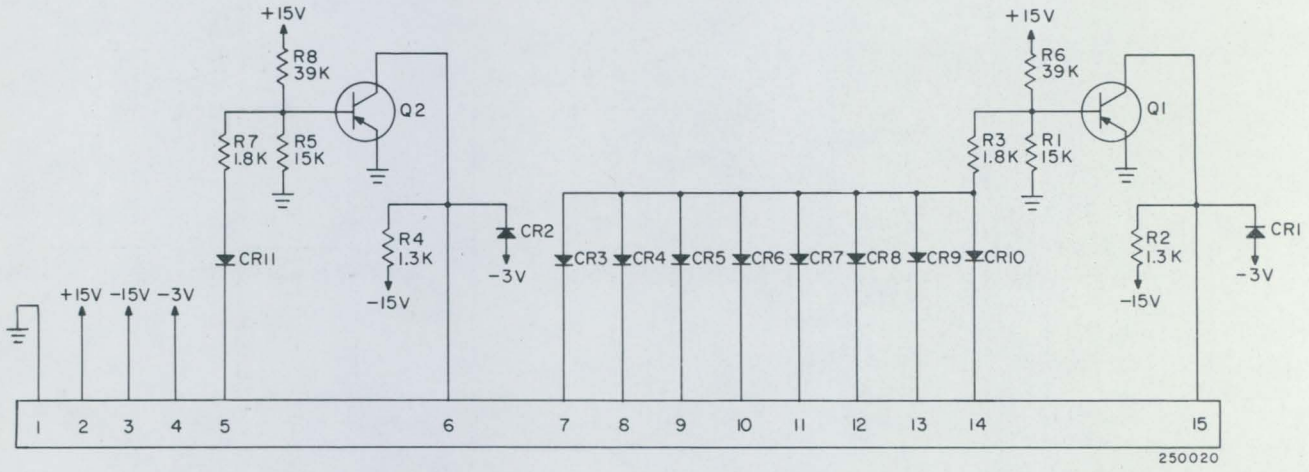


Figure 4-6. Inverter

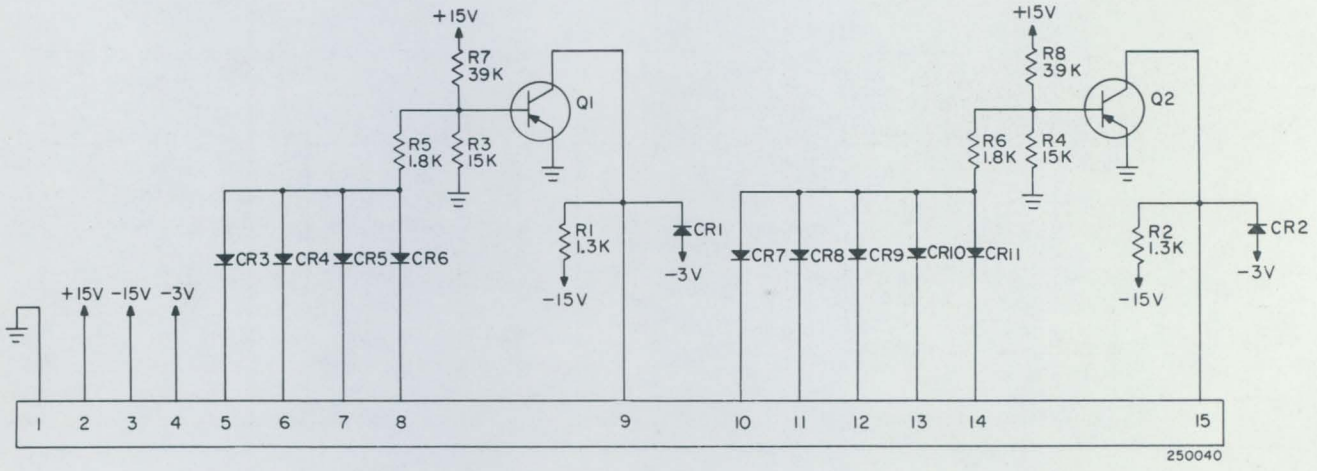


Figure 4-7. Inverter

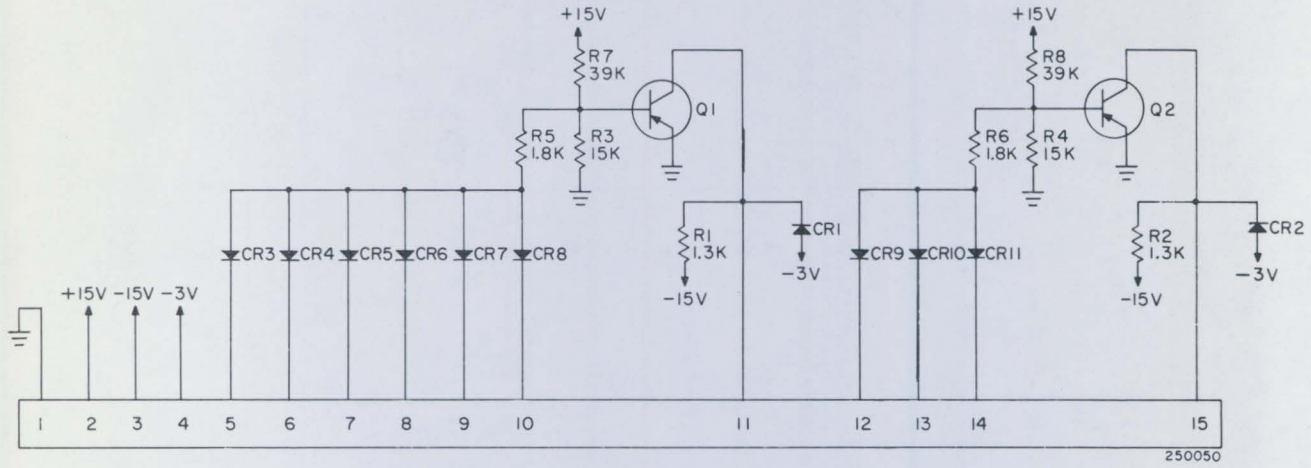


Figure 4-8. Inverter

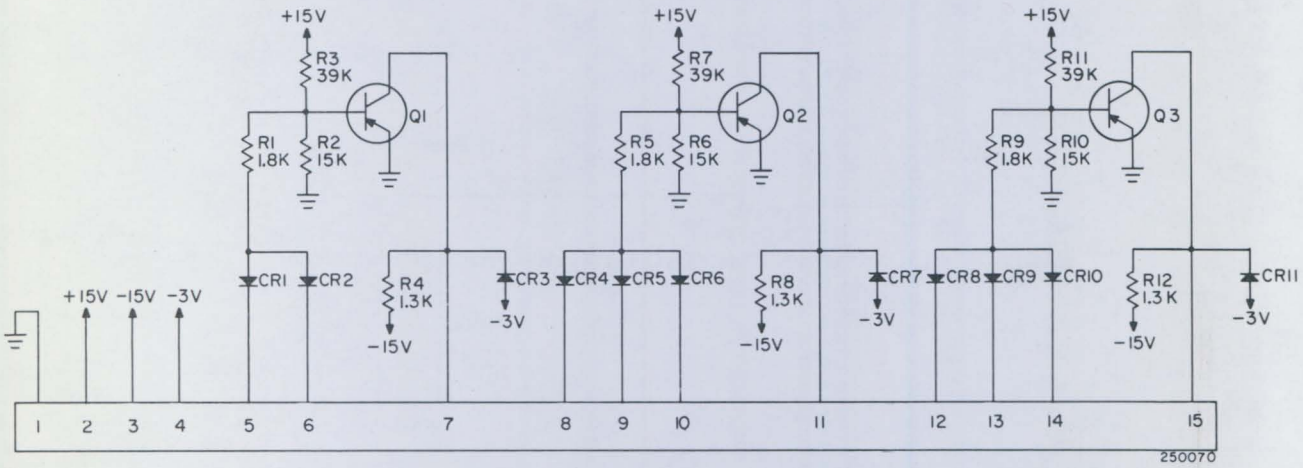


Figure 4-9. Inverter

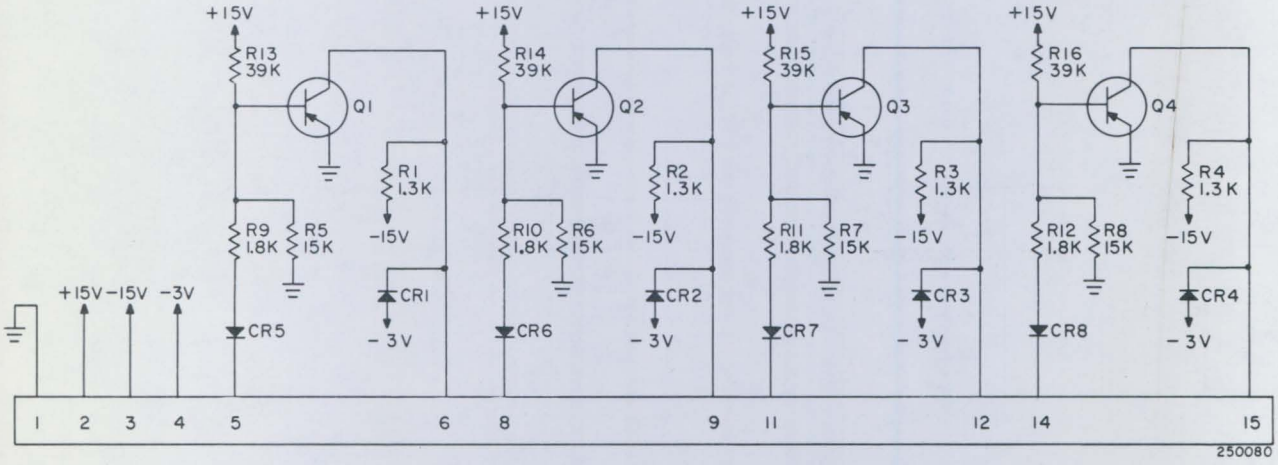


Figure 4-10. Inverter

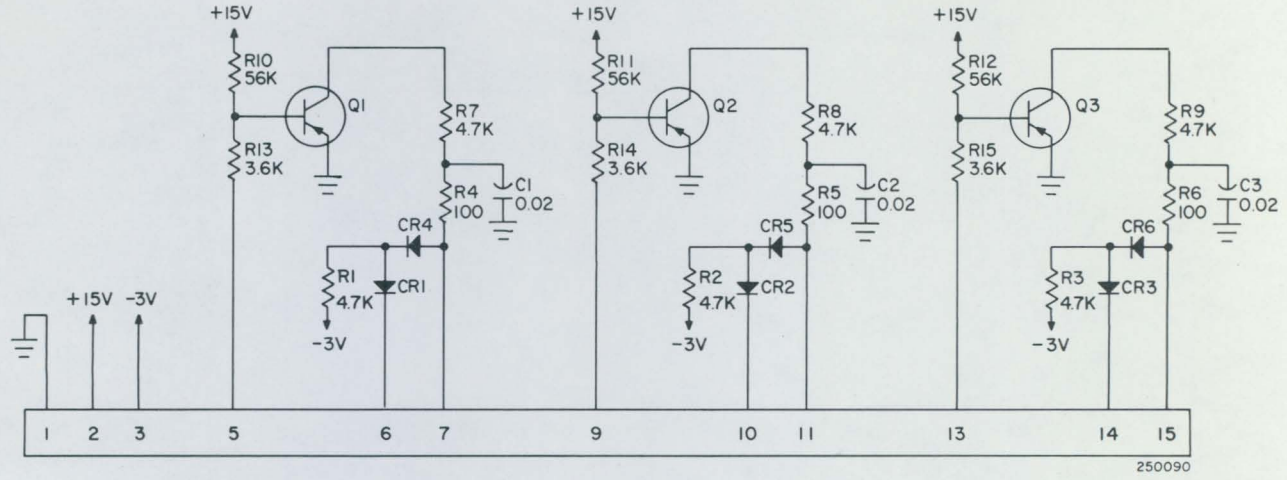


Figure 4-11. Indicator Driver

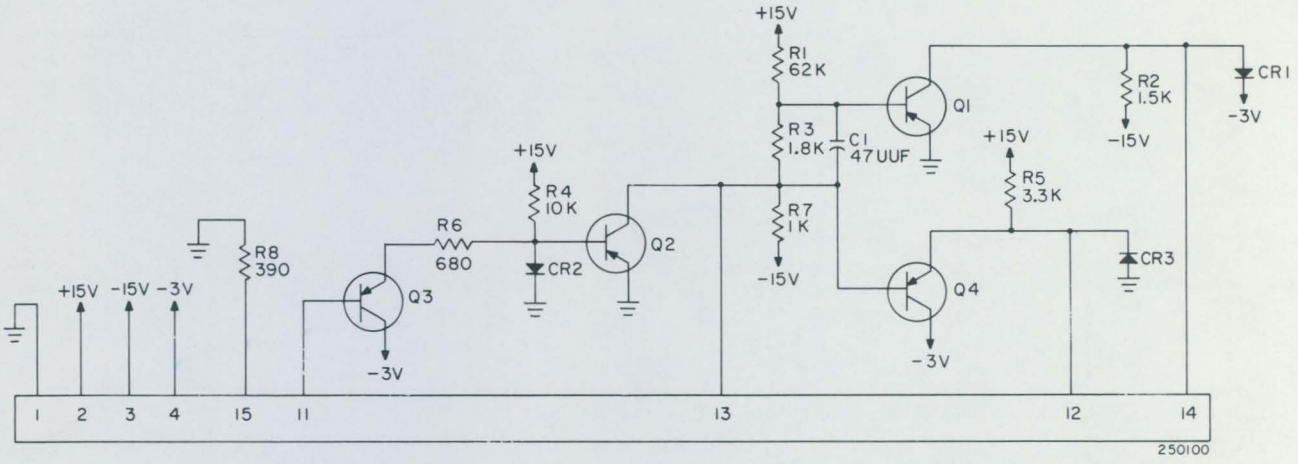


Figure 4-12. Clock Oscillator

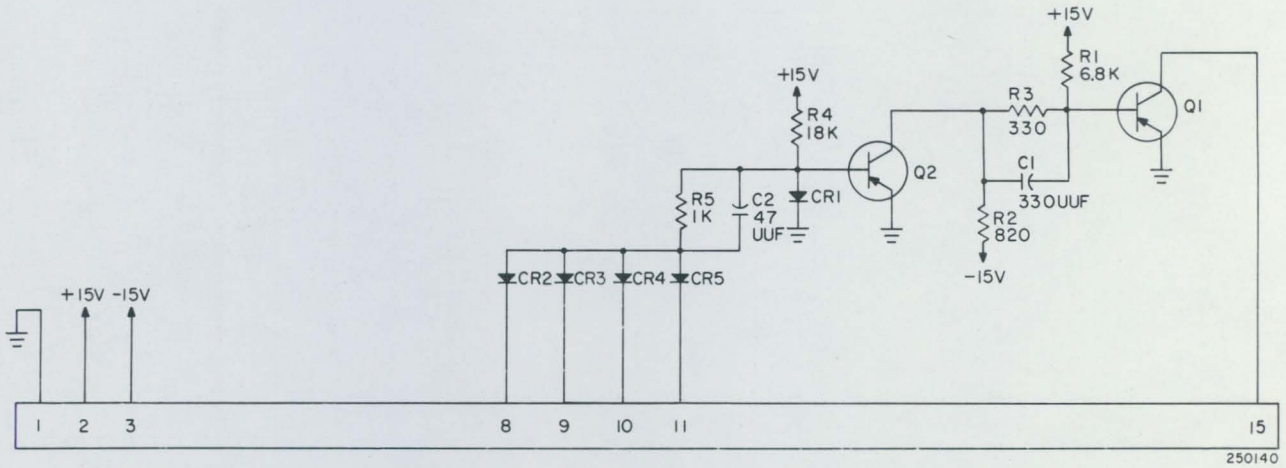


Figure 4-13. Chassis Driver

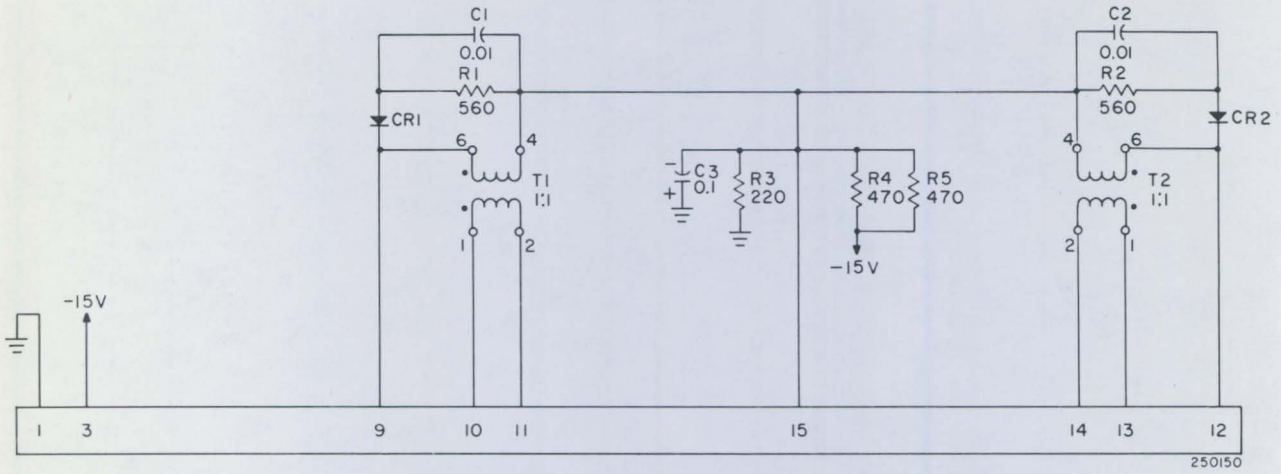


Figure 4-14. Chassis Driver Output

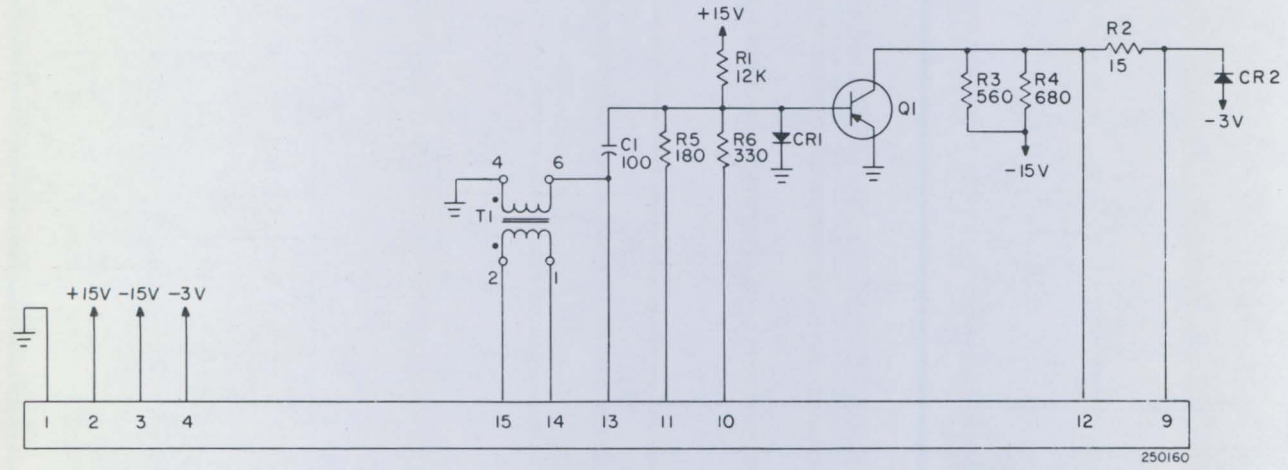


Figure 4-15. Clock Output

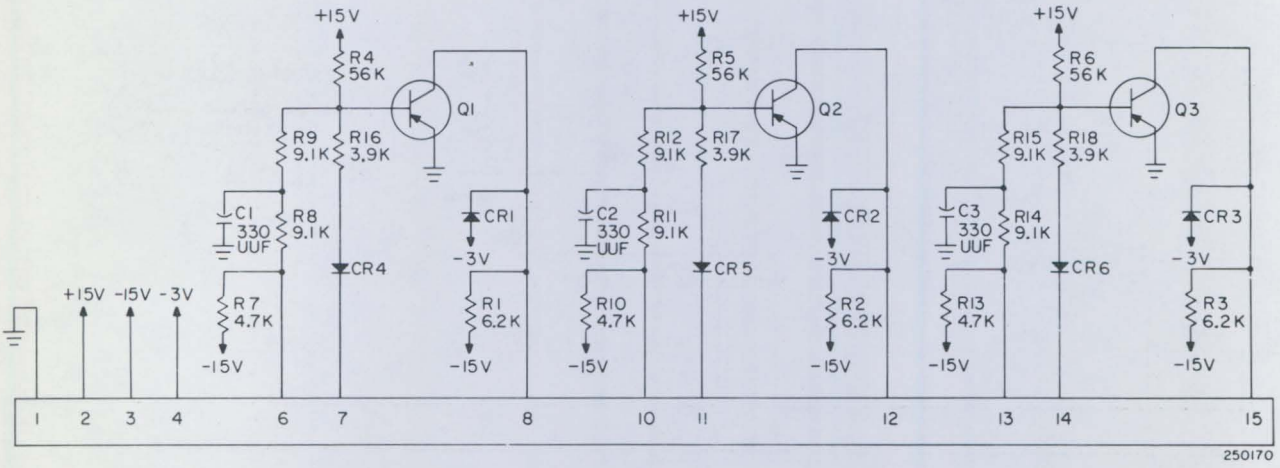


Figure 4-16. Input Amplifier

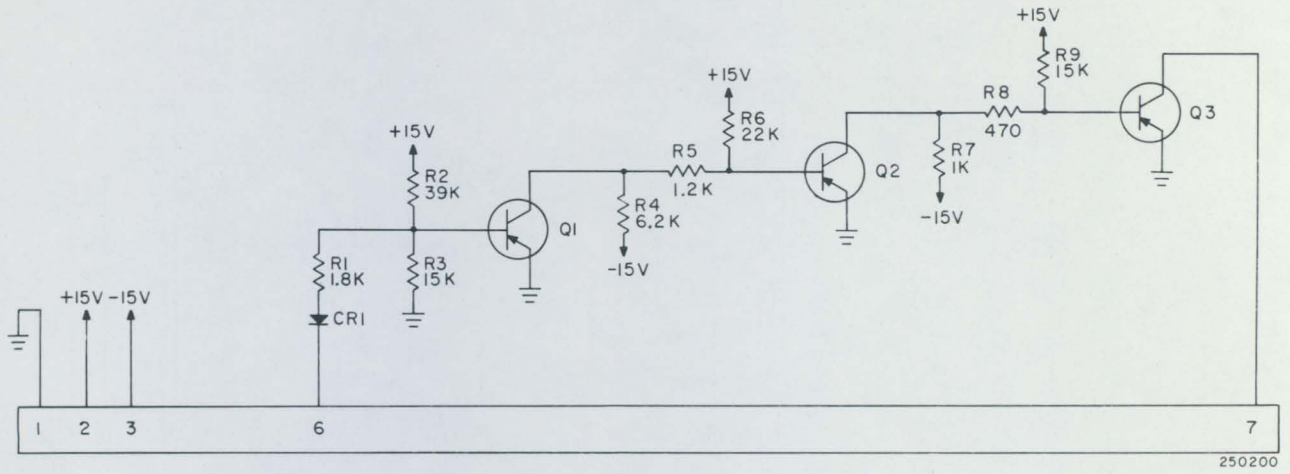
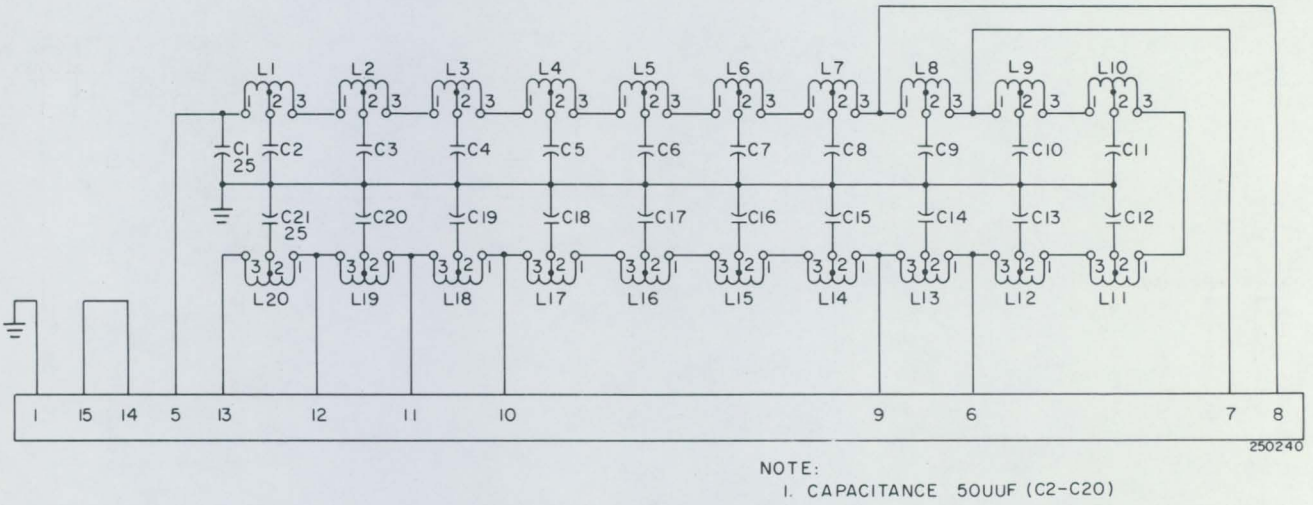
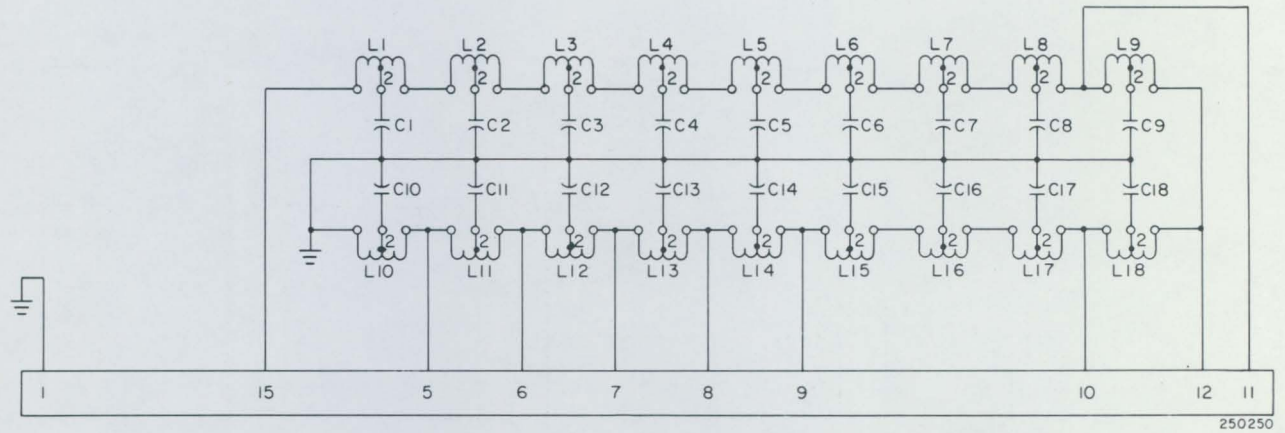


Figure 4-17. Lamp Driver



NOTE:
1. CAPACITANCE 50UUF (C2-C20)

Figure 4-18. Delay Line -3



NOTES:
1. ALL CAPACITORS ARE 25UUF
2. ALL COILS ARE 4 UH

Figure 4-19. Delay Line -4

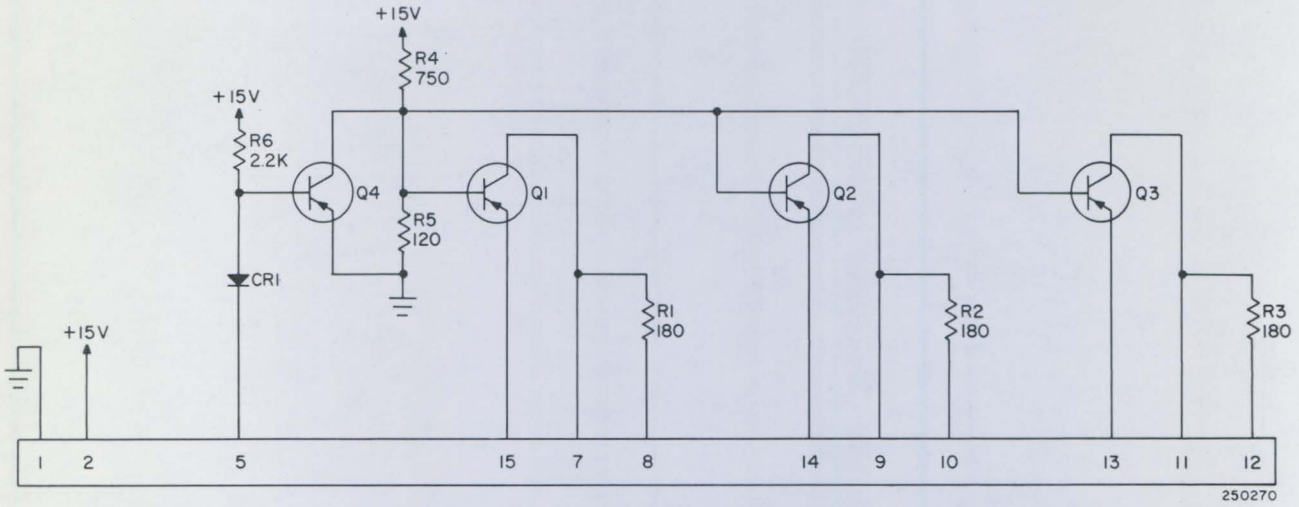


Figure 4-20. Inhibit Switch

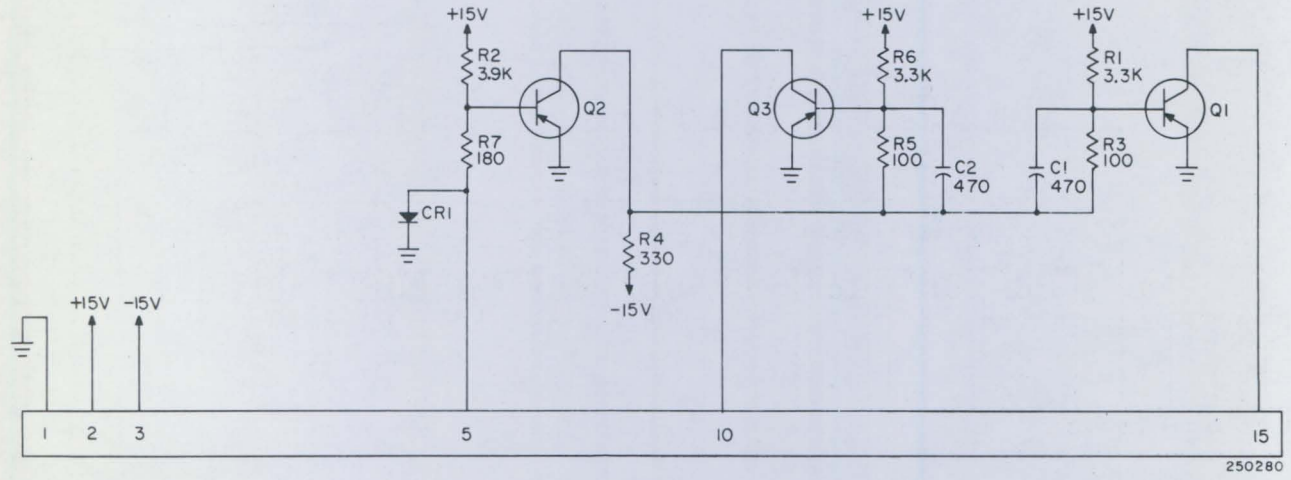


Figure 4-21. Read/Write Driver

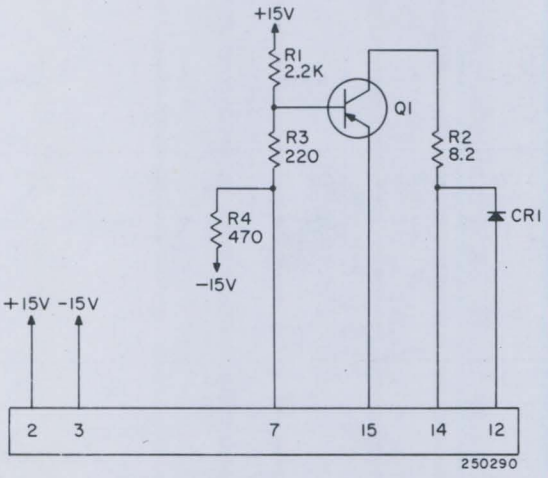


Figure 4-22. Read/Write Current Generator

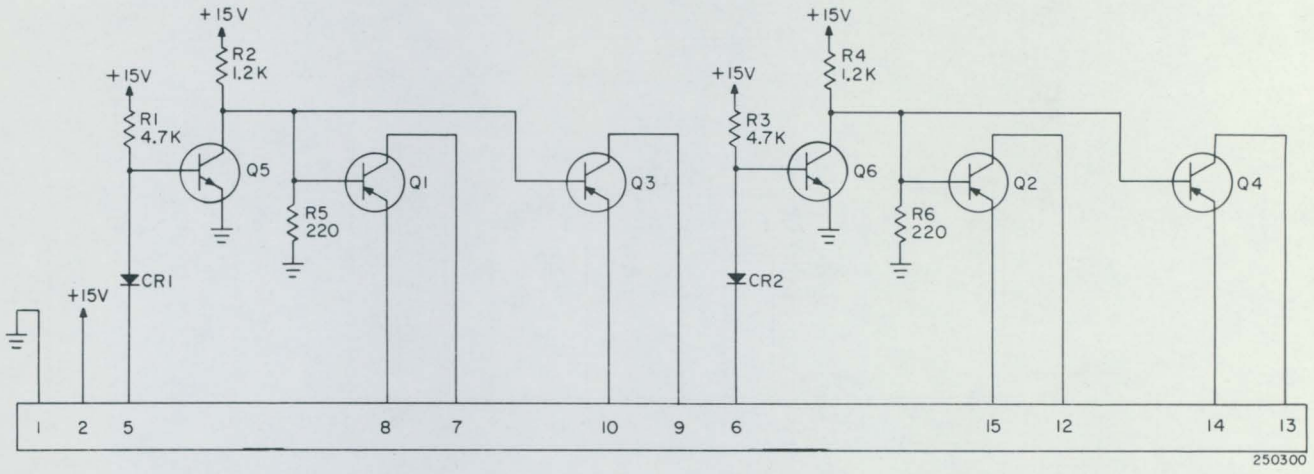


Figure 4-23. Transformer Selector

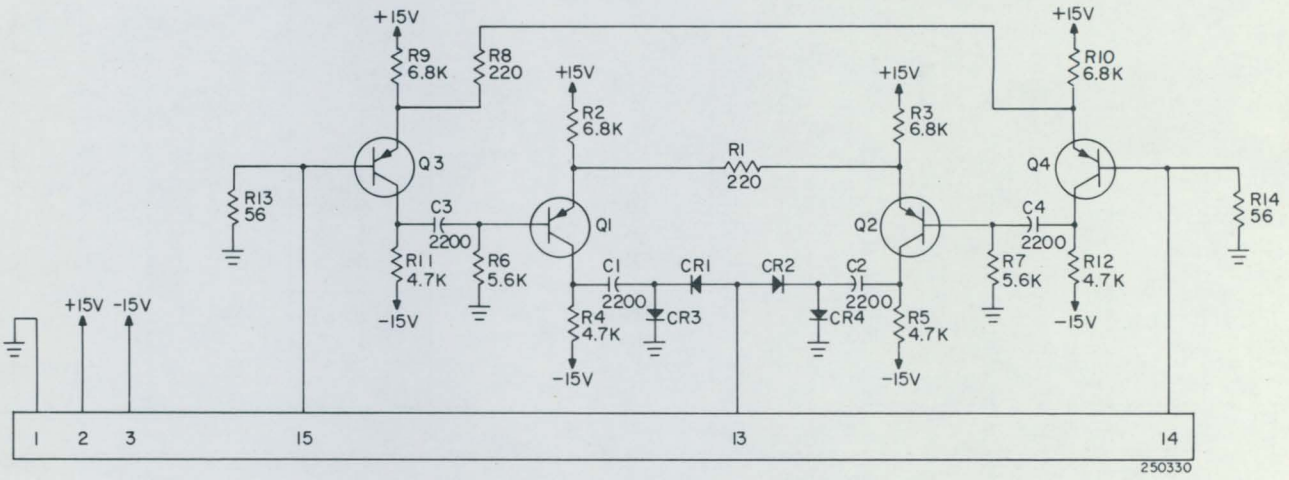


Figure 4-24. Sense Amplifier

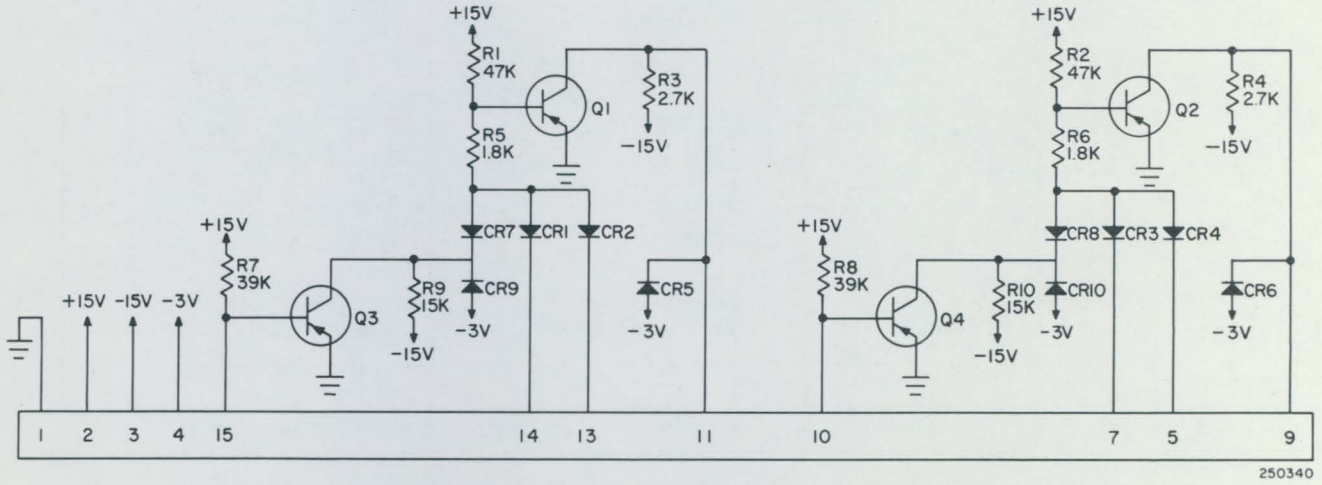


Figure 4-25. Gated Sense Amplifier

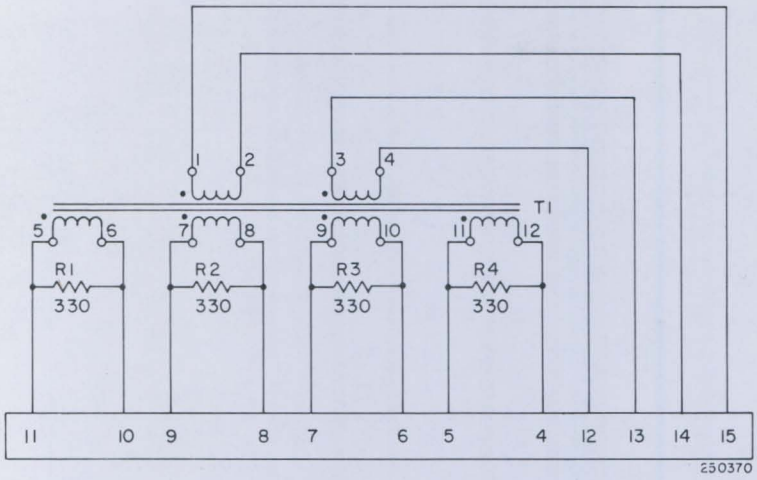


Figure 4-26. Read/Write Transformer

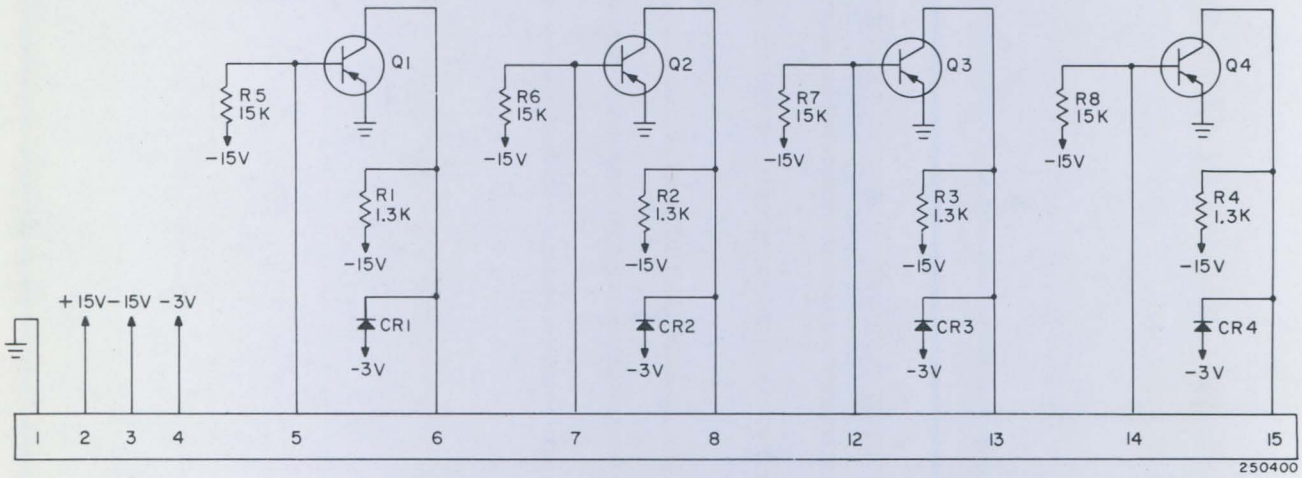


Figure 4-27. Inverter

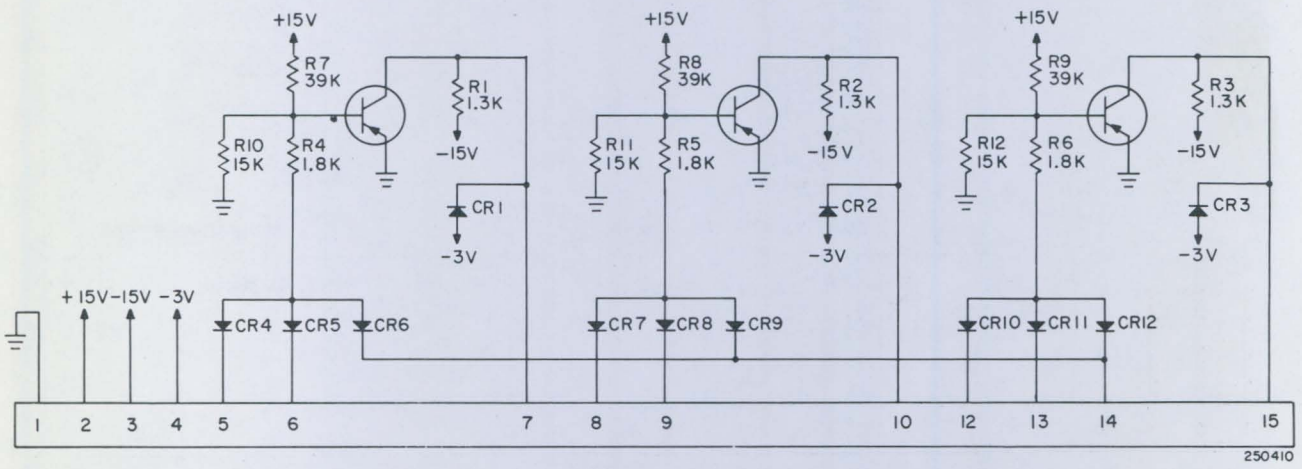


Figure 4-28. Inverter

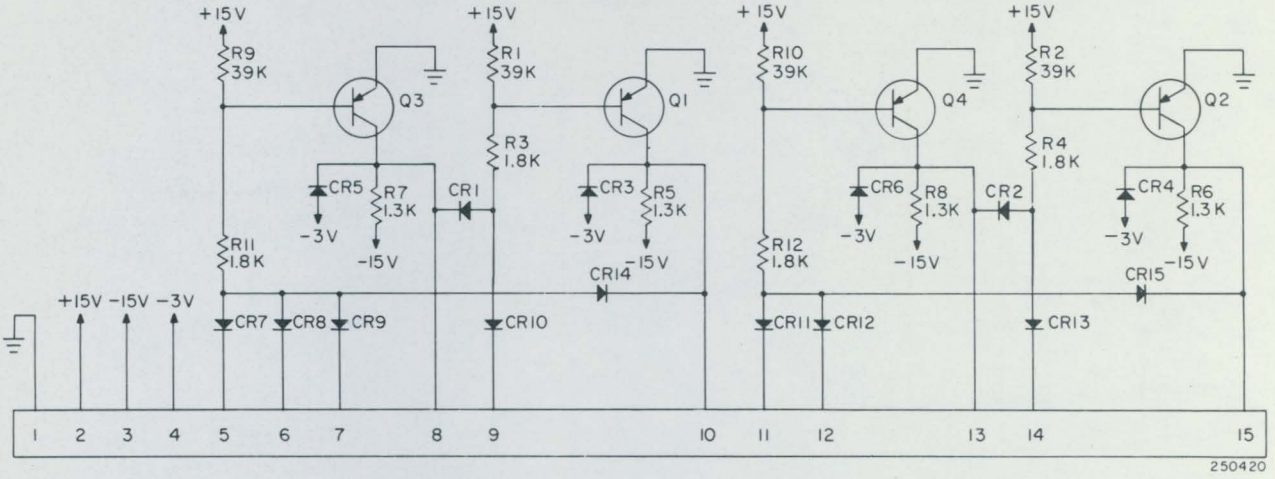


Figure 4-29. Flip-Flop

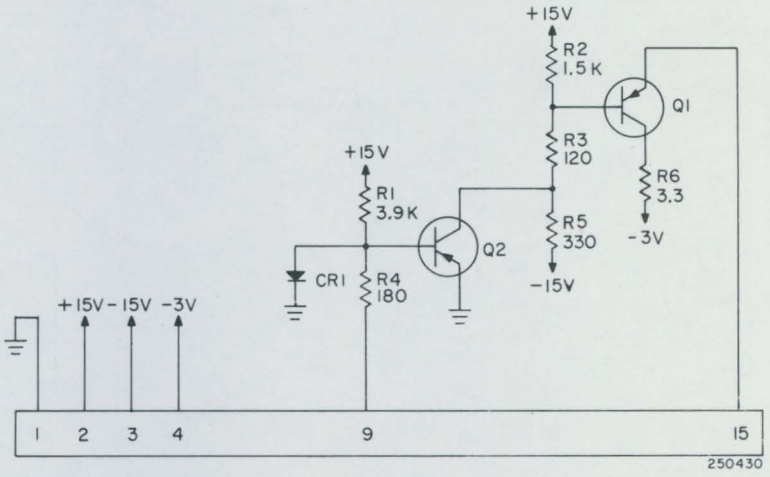


Figure 4-30. Memory Driver

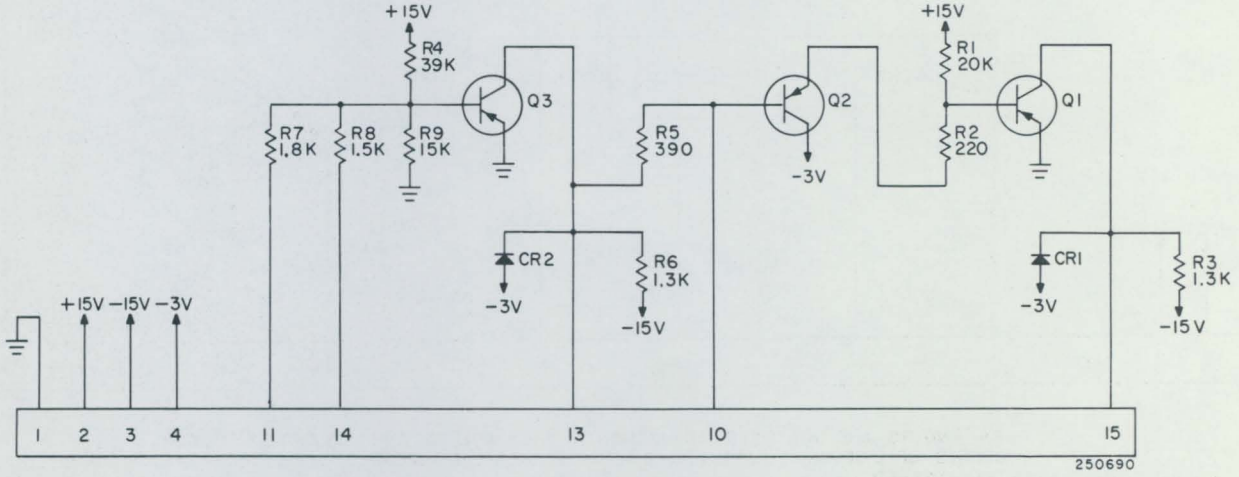


Figure 4-31. Clock Amplifier

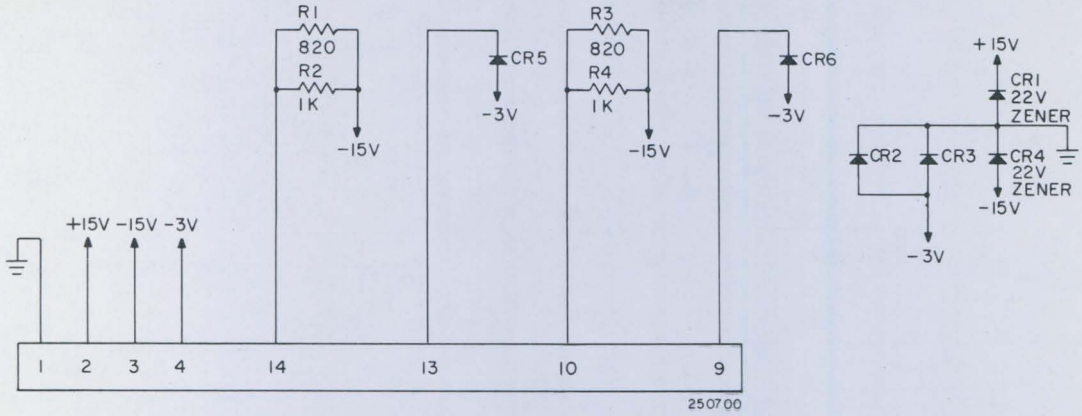


Figure 4-32. Voltage Protection

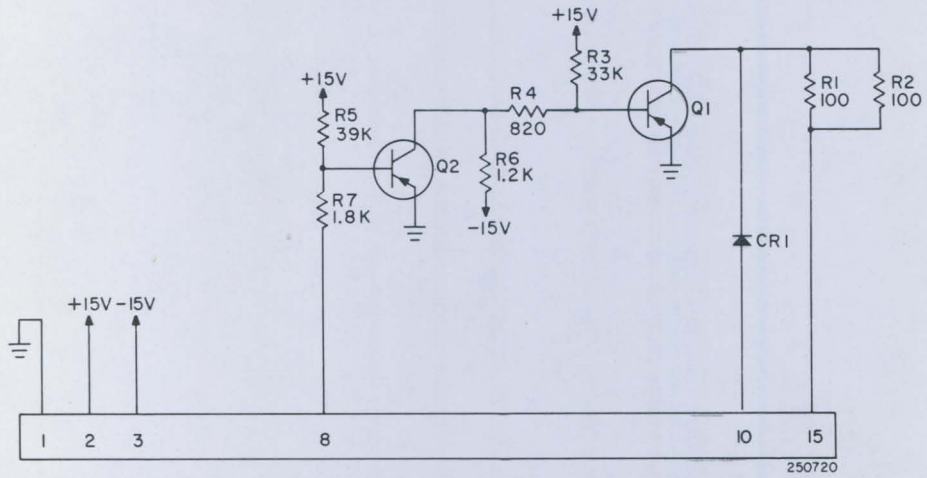
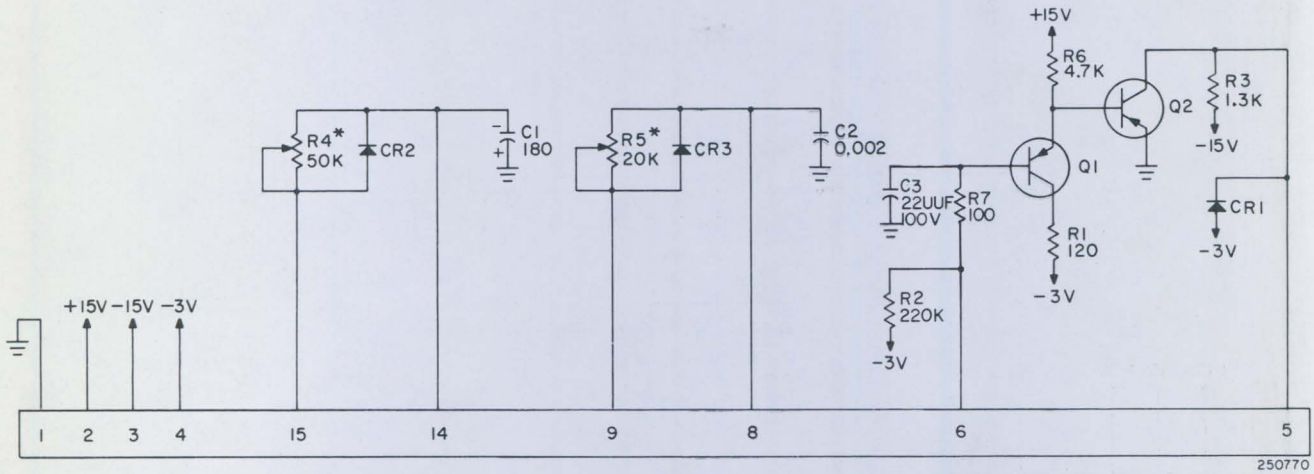


Figure 4-33. Punch Puller



NOTE:
R4 AND R5 ARE NOT FIELD ADJUSTMENTS. THEY ARE FACTORY ADJUSTED TO GIVE DELAYS OF 2 SECONDS ±10% AND 2 MICROSECONDS ±15% RESPECTIVELY. THEY ARE THEN SEALED.

Figure 4-34. Pulse Delay

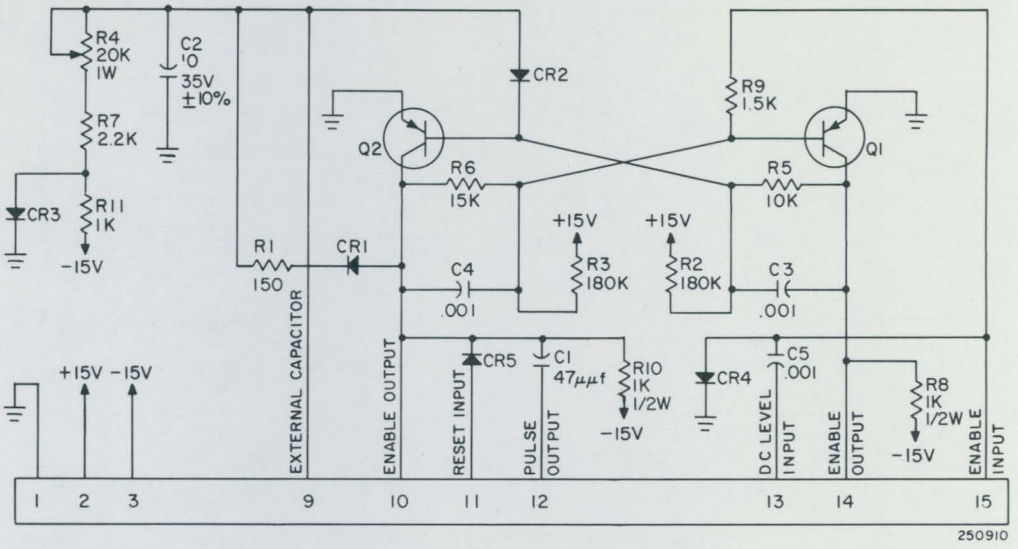


Figure 4-35. Time Delay -4

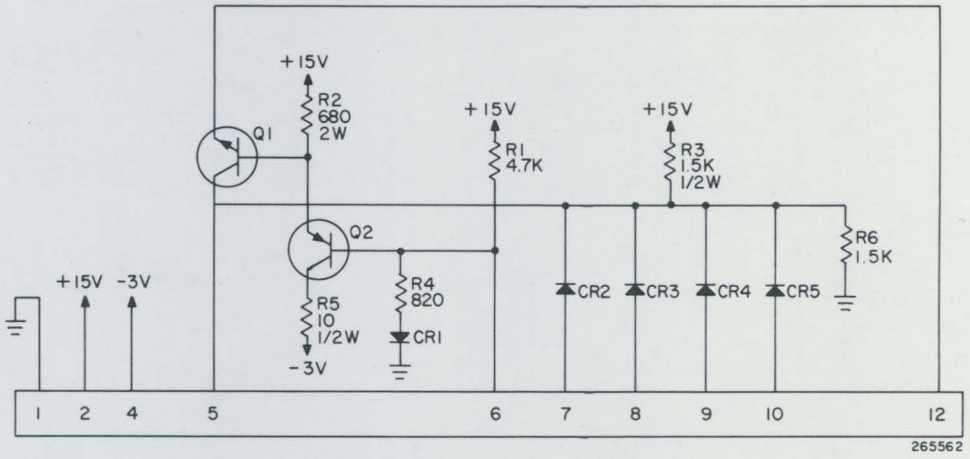


Figure 4-36. Positive Selector

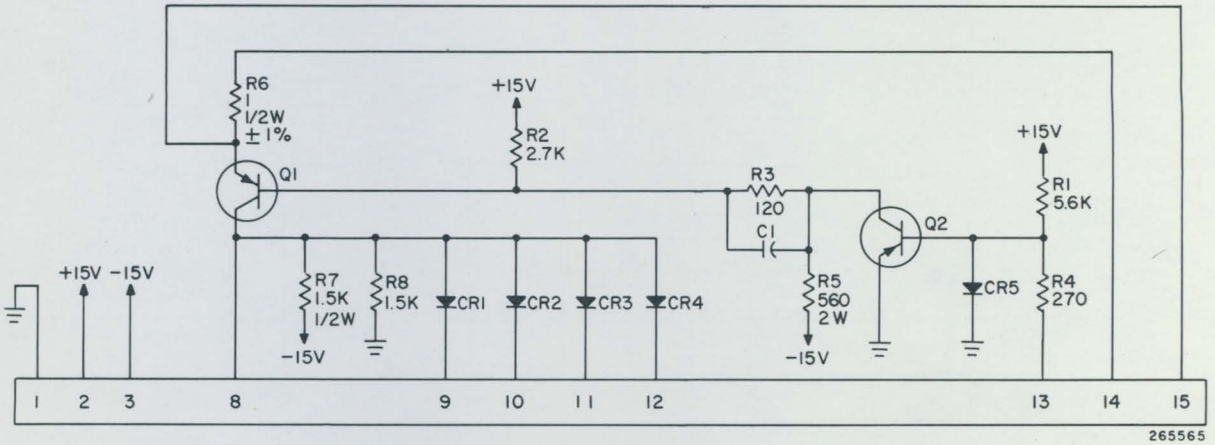


Figure 4-37. Negative

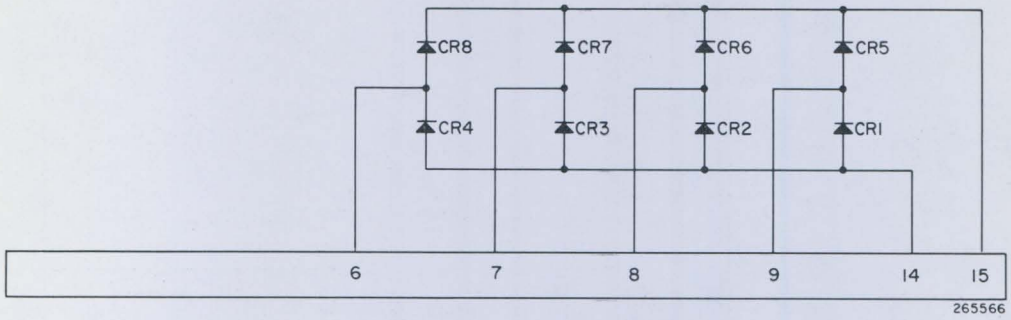


Figure 4-38. Diode Card

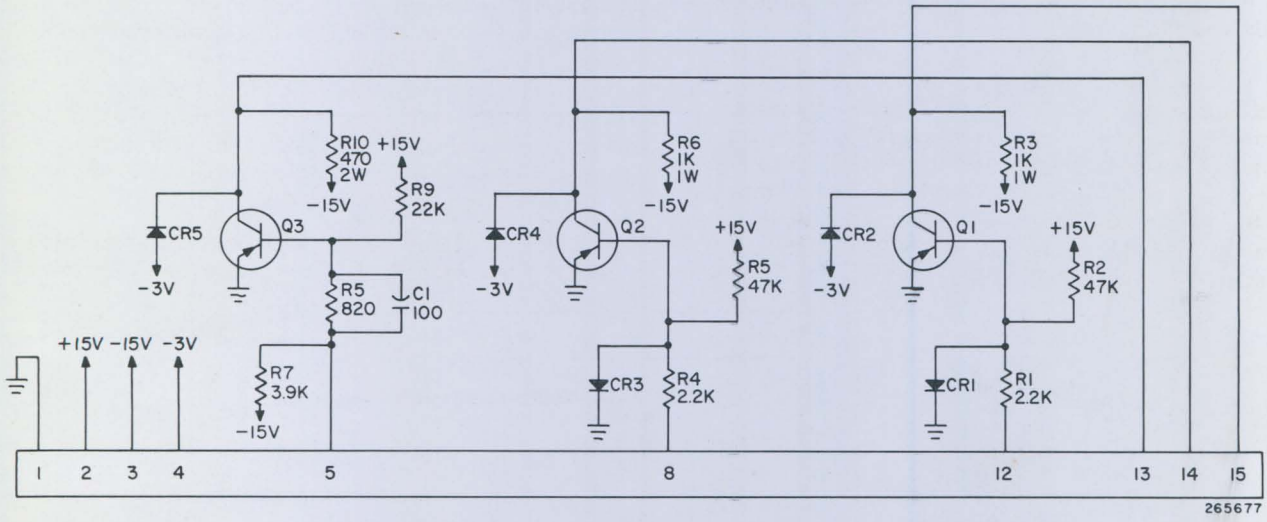


Figure 4-39. Gate and Strobe Output

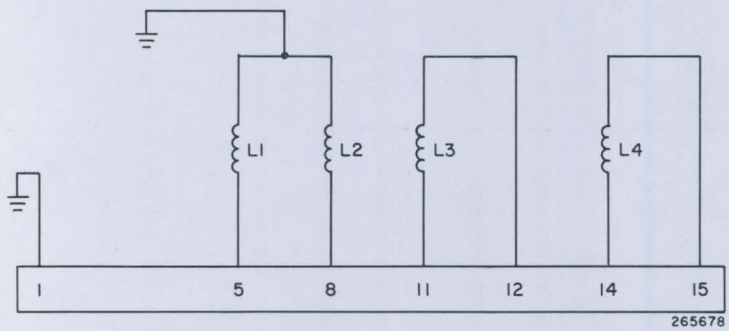


Figure 4-40. Choke

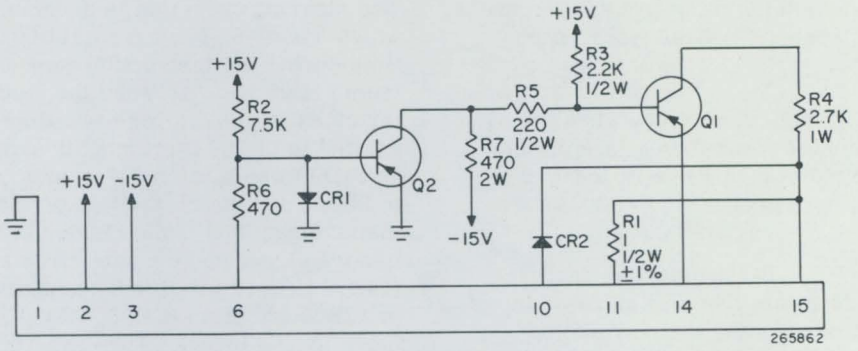


Figure 4-41. Inhibit Current Diverter

(7) FUNCTIONAL SCHEMATICS.- The functional schematics in Section 5, volume II show the logical functions of the Trainer and are referenced in this section. A logic symbology diagram is shown in figure 4-42 and should be referred to in the following explanations.

(a) LOGIC SYMBOL IDENTIFICATION. - Each independent circuit is represented by a logic symbol. The symbol used is a circle with inputs shown as arrows pointing into the circle. The number of arrows indicate the number of diodes making up the gate of that circuit. Identification of these symbols is accomplished by giving each symbol a term. For an example of this, examine the 2^0 position of the D register. The terms for this stage would be as indicated in figure 4-42.

The base letter indicates the register to which the logic symbol applies. If this was a part of the A register, A would be the base letter, etc. The left digits indicate the side of the flip-flop: 01 always represents the "1" side while 00 represents the "0" side. The right digits indicates the stage the circuit is a part of. The term 01A14 would indicate the "1" side of a flip-flop stage 2^{14} power of the A register. The circuit shown in figure 4-42 is the logic representation of the schematic examined in figure 4-3.

Also contained in the logic symbol associated with 01D00 is a term 1E4. This indicates the physical lo-

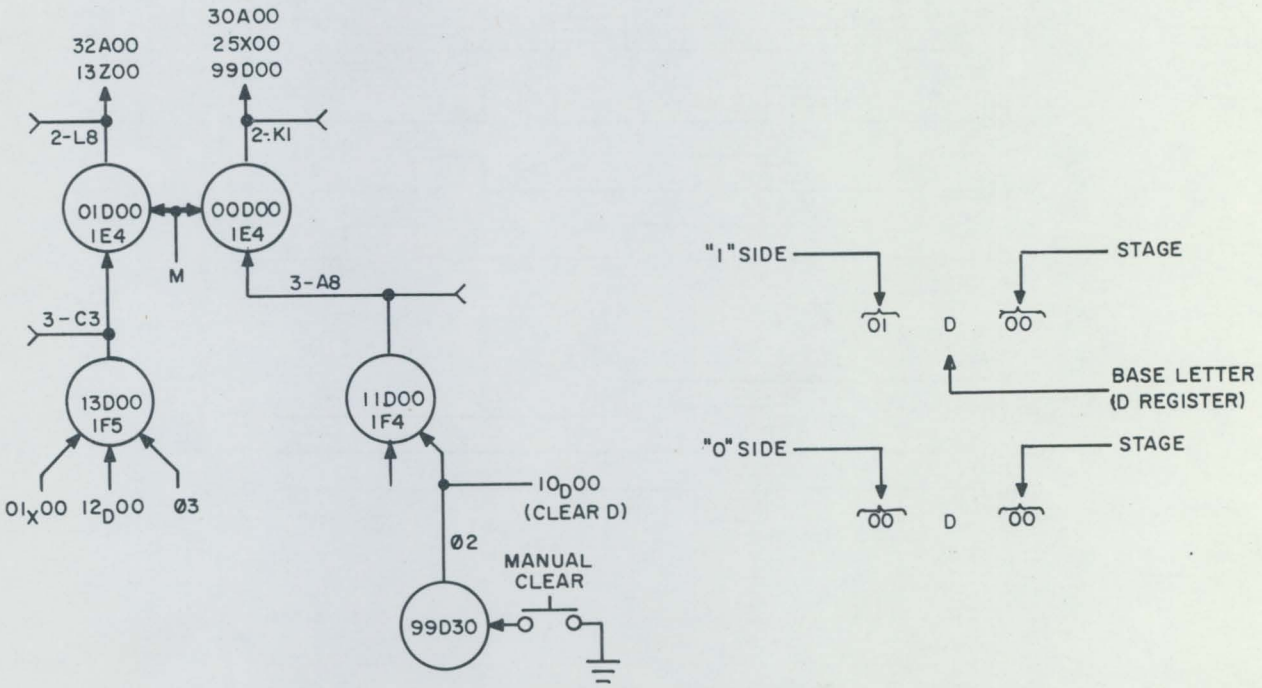


Figure 4-42. Logic Symbology

cation of the card that contains this circuit. The one shows the card to be located on chassis one, at coordinates E4.

(b) TEST POINTS. - Each circuit has a test point on its output which is located on a jack on the side of each chassis. Test points can be located exactly by the test point callout; e.g., 3-C3, is the test point at jack three, coordinates C3.

(c) CARD TYPE. - To identify the card type, it is necessary to locate the card on the chassis maps as indicated by the logic symbols. The card type is identified by a 6-digit number. See table 4-2.

(d) TEXT NOTATIONS. - Throughout this text, the diode gate and transistor inverter are referred to as a gate or inverter, depending on its primary logic function. Flip-flops, although composed of two inverters, are referenced by using the letter associated with the flip-flop and the digits representing the bit position of the flip-flop. Thus, 00A00 and 01A00 flip-flop becomes A00. Therefore setting A00 is the operation which causes 00A00 to have a "1" output and 01A00 to have a "0".

(e) DATA TRANSMISSION. - The transmission of data to a register is normally accomplished by two operations. First, the register is prepared for the transmission by clearing the register. The actual transmission then takes place. The clearing operation is accomplished by 11D00 which has a "0" input from 10D00 at the proper time during the instruction execution. The output from 11D00 is a "1" when the input from the master clock is a "0" (phase two time). This "1" is applied to 00D00 forcing a "0" output and is applied to 01D00 which has a "1" output. This is applied to 00D00 and the cleared state of the flip-flop is maintained. The transmission of data at this stage is normally through a gate circuit such as 13D00 (13D00 is used for X→D transmission). The inputs to 13D00 are from 01X00, 12D00 and the master clock (phase three). The phase three inputs are "0" for approximately 0.4 microseconds of each cycle of the master clock. (Refer to figure 4-43. Thus 13D00 is phase enabled at phase three times. The input from 12D00 is a "0" when a time in the instruction execution is reached when this transmission must take place. The remaining input, from 01X00, determines the bit which is entered into D00. If this input is a "0", the output from 13D00 is a "1" and D00 is set. If the 01X00 input is a "1", the output from 13D00 is a "0" and D00 remains cleared. The only time 13D00 has a "1" output is when all inputs are "0's". The flip-flop is cleared on phase two and the transmission occurs on phase three. This same procedure is generally used in all registers; i.e., clear on one clock phase and transmit on one of the next clock phases. The output of 00D00 is applied directly to other circuits and also to 99D00, the indicator driver.

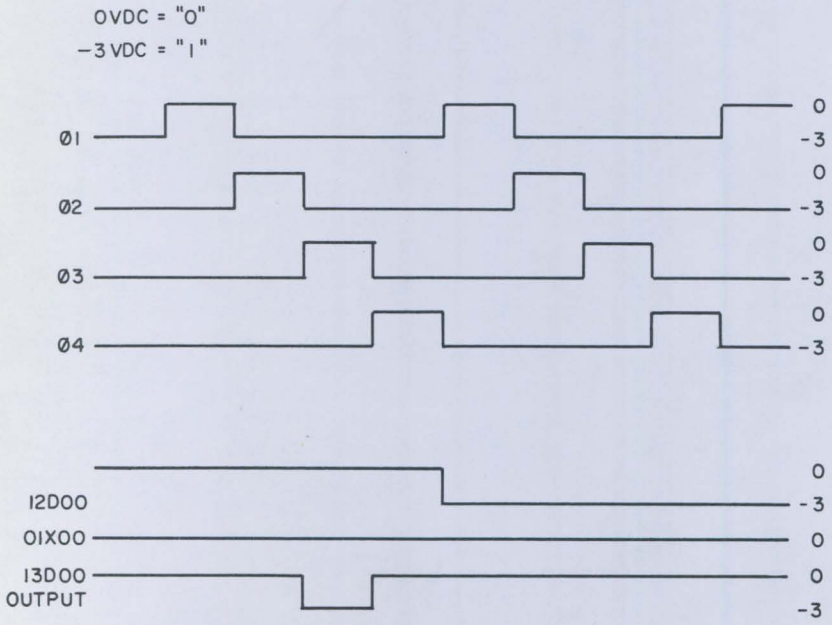


Figure 4-43. Trainer Clock Pulse

4-3. OPERATION OF THE MASTER CLOCK AND CONTROL CONSOLE.

a. GENERAL. - The control console with its indicators is the means by which the operator can control the Trainer and read the contents of the registers. By manipulating the controls the operator can operate the Trainer in high-speed run, execute instructions one at a time, or step through the clock phases. Push-button control enables the operator to change or alter the contents of any register.

b. MASTER CLOCK. - The master clock provides the main timing signals for Trainer control. The clock system utilizes a four phase output with a complete cycle time of 1.6 microseconds. Each phase utilizes 0.4 microseconds of cycle time. There are four indicators on the console which, when lighted, indicate the clock phase enabled. The enables to light the indicators are provide by J30 and J31. The particular indicator lighted is a function of the state of J30 and J31.

Clock phases are normally generated automatically under control of the output from the clock oscillator. During high-speed operation, automatic generation of clock phases is necessary to allow rapid execution of routines. When the Trainer is in the Phase Step mode, every momentary activation of the PHASE STEP switch allows the generation of one clock phase.

Note

The following logical analysis of the master clock may seem contradictory with regard to the level of inputs and outputs. However, the cards used in the clock circuits utilize more than one circuit per card, or utilize transformer coupling, making it possible to have "0" outputs for "0" inputs.

(1) LOGICAL ANALYSIS. - (See figures 5-2 through 5-4) When the Trainer is in high-speed operation, the HIGH SPEED DISCONNECT switch is in the position shown on figure 5-2. The 72Y42 circuit, (Input Amplifier, figure 4-14) without an input applied, will reflect a "1" input to itself and output a "0". This is inverted by 12J42 to a "1" input to 13J42. Regardless of other inputs, the "1" will force the 13J42 to output a "0", inverted again to a "1" by 14J42. The output of 14J42 is then applied as a constant Clear signal to J41 and J40. The output of these flip-flops are constant enables to the master clock (figure 5-4). The "0" outputs from 00J40 and 00J41 provide one of the enable inputs to the chassis drivers, 90Y1-, and also to gates located under the J30 and J31 flip-flops. Now, assume that phase one is to be enabled: The output from 01J31 is a "0" which is another enable input to 90Y11.

Oscillator 90Y00, in the meantime, keeps recycling to provide two outputs which are 180 degrees out of phase. The output varies between "0" and "1" with each output being applied, alternately, as an input to 90Y02 and 90Y04. When the input to 90Y04 is a "0" the output from the Shaper circuit is also a "0". This output is applied as the final enable input to 90Y11. The output from 90Y11 now becomes a "0" which is coupled to the clock output circuits by 90Y21 at the same signal level. This signal is applied as the input to the 92Y01 circuit, which supplies the clock signal for phase one to all the logic chassis. The "0" output from the 92Y01 circuit is connected to a bus bar on each chassis for distribution. At the same time, the "1" output from 90Y00 forces the output from 10J30 to a "0". Now, 10J32 with "0" inputs from 00J41, 01J31, and 10J30 has a "1" output which sets J30 in preparation for phase two.

When the oscillator changes phase, the "0" output from 90Y00 is applied as an input to 90Y02 which produces a "0" output. The inputs to 90Y12 are now the "0" outputs from 90Y02, 00J40, and 01J30. The "0" output from 90Y12 is coupled to the output circuits at the same signal level by 90Y22. This signal is applied as the input to the 92Y02 circuit. The "0" output from 92Y02 is used to drive the clock buses for phase two on the logic chassis. The "1" output from 90Y00, in the meantime, forces the output from 90Y06 to a "0" which is an enable input to 10J33. Now, 10J33 with "0" inputs from 00J40, 01J30, and 90Y06, has a "1" output which clears J31 in preparation for phase three.

When the oscillator recycles again, the "0" output from 90Y00 is applied as an input to 90Y04 which now has a "0" output. The inputs to 90Y13 are the "0" outputs from 00J41, 90Y04, and 00J31. The "0" output from 90Y13 is coupled to the output circuit at the same signal level by 90Y23. This signal is applied as the input to 92Y03 circuit. The "0" output from the 92Y03 is used to drive the clock buses for phase three on the logic chassis. The "1" output from 90Y00, meanwhile, forces the output from 10J30 to a "0" which is an enable input to 10J34. Now, 10J34 with "0" inputs from 00J41, 00J31, and 10J30 has a "1" output which clears J30 in preparation for phase four.

As the oscillator recycles, the "0" output from 90Y00 is applied as an input to 90Y02 which now has a "0" output. The inputs to 90Y14 are the "0" inputs from 90Y02, 00J30, and 00J40. The "0" outputs from 90Y14 is coupled to the output circuit at the same level by 90Y24. The "0" output from 92Y04 is used to drive the clock buses for phase four on the logic chassis. At the same time, the "1" output from 90Y00 forces the output from 90Y06 to a "0" which is an enable input to 10J31. Now 10J31, with "0" inputs from 00J40, 00J30, and 90Y06, has a "1" output which sets J31 in preparation for phase one. The clock now recycles to enable phase one and continues to recycle to enable each phase in turn as long as the Trainer is in high-speed operation.

(a) PHASE STEP MASTER CLOCK LOGIC. - (See figure 5-2.) Under certain conditions, it may be necessary to prevent the master clock from recycling automatically, and instead, utilize manual control.

The PHASE STEP switch, in conjunction with HIGH SPEED DISCONNECT switch, is used to perform this function. When both switches are in the normal position, they have no effect on the automatic recycling of the clock. When the HIGH SPEED DISCONNECT switch is activated, every momentary activation of the PHASE STEP switch allows the generation of one clock phase.

When the Trainer is in normal high-speed operation, the "0" output from 72Y42 is inverted to a "1" by 12J42. This output goes through two more stages of inversion, 13J42 and 14J42, to clear J41 and J40. The "0" outputs from 00J41 and 00J40 provide the constant enables to the master clock. With the PHASE STEP in the normal position, the "0" input to 72Y41 is inverted to a "1" and sets J42 which supplies a "0" to 72Y40. With an open from the switch to 72Y40, the circuit, being an input amplifier, will reflect a "1" input to itself and unconditionally output a "0" to 14J40 which inverts the signal to a "1". This signal is one of the four inputs to 15J40 and 15J41 which forces the output to "0's". The "0" inputs to 00J41 and 00J40 will have no effect on the flip-flops.

When the HIGH SPEED DISCONNECT switch is activated and the PHASE STEP remains in the normal position, the "0" input to the 72Y42 inverter is combined with the output of 00G70 which is the zero side of the Op Step flip-flop. This flip-flop will be set only when the Trainer is in high-speed run operations, therefore, the HIGH SPEED DISCONNECT switch cannot be used to disable the master clock if the Trainer is running in high speed. Assuming the Trainer to be stopped, the input from 00G70 will be a "0" to 72Y42 which inverts it to a "1" and outputs to 12J42. The second inversion is accomplished by 12J42 and the "0" output is one of three inputs to 13J42. The inputs from 01L17 is from the memory timing chain which will input a "1" whenever memory is being used. The memory timing chain cannot be phase stepped. The third input is from 14J43 which will input a "1" when the Trainer is being master cleared. The inputs to 13J42 then, are the "0" inputs from 12J42, indicating that the HIGH SPEED DISCONNECT switch is activated and the Trainer is not in high-speed run, 01L17, indicating that memory is available and 14J43, indicating the Trainer is not being master cleared. The "1" output from 13J42 is inverted to a "0" by 14J42 which removes the constant Clear enable to J40 and J41. The two other outputs from 13J42 to 15L11 and 13L11 will disable the memory timing chain so memory cannot be referenced during the phase stepping of the master clock.

If, phase four was generated previous to activating the HIGH SPEED DISCONNECT switch, the inputs to 24J42 are the "0" inputs from 90Y06 and 00J40. The "1" output from 24J42 sets J43 and the "0" output from 01J43 is applied as an enable to 17J40. As the HIGH SPEED DISCONNECT switch is being activated, phase one is being enabled in the normal manner (high-speed operation). The output from 10J30 is forced to a "0" by the "1" inputs from 90Y00. Now 17J40, with "0" inputs from 01J43 and 10J30, has a "1" output which sets J40. When J40 is set, phase two cannot be enabled. At the same time, 21J42 with "0" inputs from 00J41 and 10J30 has a "1" output which clears J43. When the oscillator recycles, the "1" output from 90Y00 forces the output from 90Y06 to a "0". Now 17J41, with "0" inputs from 00J43 and 90Y06, has a "1" output which sets J41. With both J40 and J41 set, no clock phases can be generated. The oscillator keeps recycling, but no clock enables are transmitted to the logic chassis. The circuits remain in this condition until activation of the PHASE STEP switch. Also, as long as the PHASE STEP switch is in the normal position, J42 is set.

In the preceding paragraph, phase two was the initial clock phase assumed to be disabled. The first function of phase step is to enable phase two. When the PHASE STEP switch is momentarily depressed, the ground level signal input to 72Y40 combined with the output from 01J42 produces a "1" and forces a "0" output from 14J40. The "0" is applied to both 15J40 and 15J41. With 15J40 now having "0" inputs from 14J40, 00J43, and 01J41, the only input required is from 10J30. As the oscillator recycles, a "1" output from 90Y00 forces the output from 10J30 to a "0", which is the final enable required by 15J40. The "1" output from 15J40 clears J40 in preparation for phase two. When the oscillator changes phase, phase two is enabled in the normal manner with the required "0" output from 00J40. Now 24J42, with "0" inputs from 00J40 and 90Y06, has a "1" output which sets J43 and clears J42. The clearing of J42 removes the enable from 72Y40 and prevents more than one clock phase from being generated during each activation of the PHASE STEP switch. Since J41 had been previously set, phase three cannot be enabled at this time. However, the oscillator continues to cycle and a "1" output from 90Y00 forces the output from 10J30 to a "0". This signal, coincident with the "0" output from 01J43, enables 17J40 to set J40. Phase four cannot be enabled with J40 set. When the PHASE STEP switch is returned to its normal position, J42 is again set. On the next activation of the switch, the same enabling process takes place with the "0" output from 14J40 being used to enable 15J41. This circuit also has enable inputs from 01J43 and 01J40. As the oscillator output changes phase, the "1" output from 90Y00 forces the output from 90Y06 to a "0" which fully enables 15J41. The "1" output from this circuit clears J41 to provide the enable for phase three.

When the oscillator recycles again, phase three is enabled in the normal manner. Now 21J42, with "0" inputs from 00J41 and 10J30, has a "1" output which clears J42 and J43. The "1" output from 01J42 again disables 72Y40. Since J40 had been previously set, phase four cannot be enabled. The oscillator continues to cycle and, on the next output phase change, the "1" output from 90Y00 forces the output from 90Y06 to a "0". This output coincident with the "0" output from 00J43 enables 17J41 to set J41. Phase one cannot be enabled with J41 set. When the PHASE STEP switch is returned to its normal position, J42 is set again in preparation for the next pulse. The phase

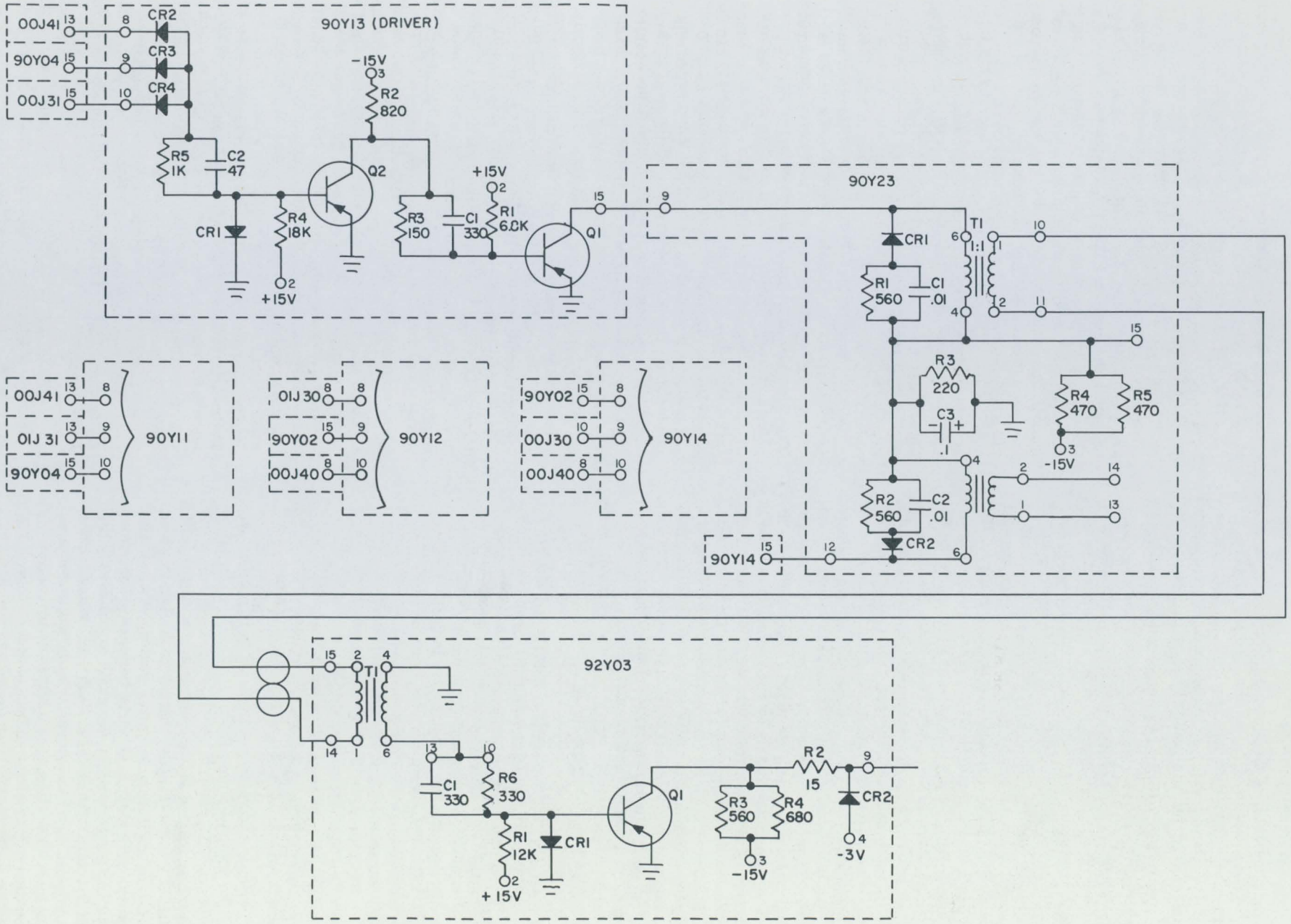


Figure 4-45. Master Clock Driver Circuits

step logic thus allows the generation of one clock phase upon each activation of the switch, and the circuits return themselves to a stable state. Phase one is generated in the same manner as phase three and phase four is generated in the same manner as phase two.

(2) ELECTRONIC ANALYSIS. - The basic oscillator for the clock system is a delay line oscillator with a frequency of 1.25 megacycles. The period or frequency of oscillation depends on the delay time of the delay line (see figure 4-44). For example, if Q2 had been cut off, its collector begins to drop to -15 VDC, but is clamped at -3 VDC by Q4. With Q2 cut off, Q3 must also be cut off because of the circuit configuration. The change in collector voltage of Q2, 0 VDC to -3 VDC, is applied to the delay line as feedback to the base of Q3. The change in voltage charges the delay line and moves as a wave front down the line. After an interval of time determined by the constants of the line, the base of Q3 is driven negative. Q3 turns on and drives Q2 into conduction. The collector of Q2 now rises to 0 VDC. The delay line discharges through Q2 towards ground. When the line has discharged, the base of Q3 becomes positive with respect to the emitter and turns off. The base of Q2 goes positive at this time and turns off. The same time interval is used in the charge and discharge of the line. The amount of delay in the line is approximately 0.40 microseconds. The oscillator keeps cycling to generate a symmetrical square wave output. The delay line is terminated in its characteristic impedance, R8, to prevent any reflections on the line. The delay line is illustrated in the functional schematics as 90Y01.

CAUTION

DL-3 (90Y01) and DL-4 (90Y03 and 90Y05) are multitapped delay lines. The taps to be utilized are determined by factory test. Do not change taps unless it is definitely determined that the wrong tap is in use. The output from the oscillator is symmetrical with the pulses being approximately 0.40 microseconds in width at the half-amplitude points. The output from the shapers is a positive pulse with the width being determined by the amount of delay in DL-4.

The square wave output from Q2 is fed to Q1 and Q4. The positive output from Q2 is utilized by Q4 to furnish a 0 VDC input to 90Y04. The negative output from Q2 is utilized by Q1 to furnish a 0 VDC input to 90Y02. C1 in the base circuit of Q1 is used to speed up the rise and fall time of the input pulse to cause more effective switching of the transistor. Since both 90Y02 and 90Y04 work in the same manner, only one of the circuits will be described. The output from 90Y02 is utilized in the generation of phase two and phase four. The output from 90Y04 is utilized in the generation of phase one and phase three. The 0 VDC output from Q4 (90Y00) drives Q3 into cut off. The collector of Q3 drops toward -15 VDC, but is clamped at -3 VDC. Connected to the base of Q2 is a delay line (DL-4) which is used to shape the pulse and limit the time of Q2. The delay line effectively places the base of Q2 at DC ground. When the voltage change appears at the base of Q2 (-3 VDC), Q2 turns on. The voltage change moves down the line as a wave front. At the ground end, the wave is reflected back as a voltage change in the positive direction. When this voltage change reaches the base of Q2, the transistor is turned off. Regardless of the width of the input pulse (within limits), the conduction time of Q2 is limited to twice the delay time of the line. In this case, the line provides a delay of 0.18 microsecond in one direction. The wave is not reflected back down the line because of the termination provided by R5 and CR2 to ground via the -3 VDC power supply. When Q2 is conducting, the -3 VDC output drives Q1 into saturation. The 0 VDC output from Q1 is an enable input to the clock driver.

The chassis drivers deliver a positive current pulse output to generate a clock phase. Figure 4-45 illustrates a typical driver with its associated output stage. The state shown is for phase three but the three other phases are exactly the same in configuration and operation. When the three input signals are at a ground level, Q2 is cut off. If any or all input signals are at a -3 VDC level, Q2 is conducting. When Q2 is cut off, the negative potential at its collector drives Q1 into conduction. The ground level signal from Q1 is fed to the output coupling circuit 90Y23. The capacitors in the bases of Q2 and Q1 are to facilitate switching times by increasing the rate of rise and fall times.

The coupling circuit consists of 1:1 impedance transformer with no phase inversion from primary to secondary. The positive going pulse is coupled to Clock Driver circuit, 92Y03. This pulse is coupled across the 1:1 input transformer and applied as a negative input to Q1. The transistor turns on and its output goes from -3 VDC (clamp voltage) to 0 VDC. This signal is fed to the logic chassis as the signal for phase three. The other phases are generated in the same manner, utilizing their own circuits.

Since all four output coupling circuits are interconnected through their bias supplies, provisions are made to minimize any bias voltage changes. The junction of R3 and C3 is held at approximately -3 VDC regardless of which clock phase is enabled. The constant voltage is obtained by holding a charge on C3 fairly constant over a wide change in load current. In this manner, the coupling circuit can be time shared by each clock phase with the signal level remaining fairly constant. A clamping circuit connected across T1 consists of CR1 in series with parallel network R1 and C1. This circuit clamps out negative transients that may appear as the transistors are switching. This action attempts to maintain the square wave symmetry of the output pulse.

c. CONSOLE CONTROL. - The console control (figure 4-46) located in the center of the Trainer serves as a tool for investigating failures, and as a

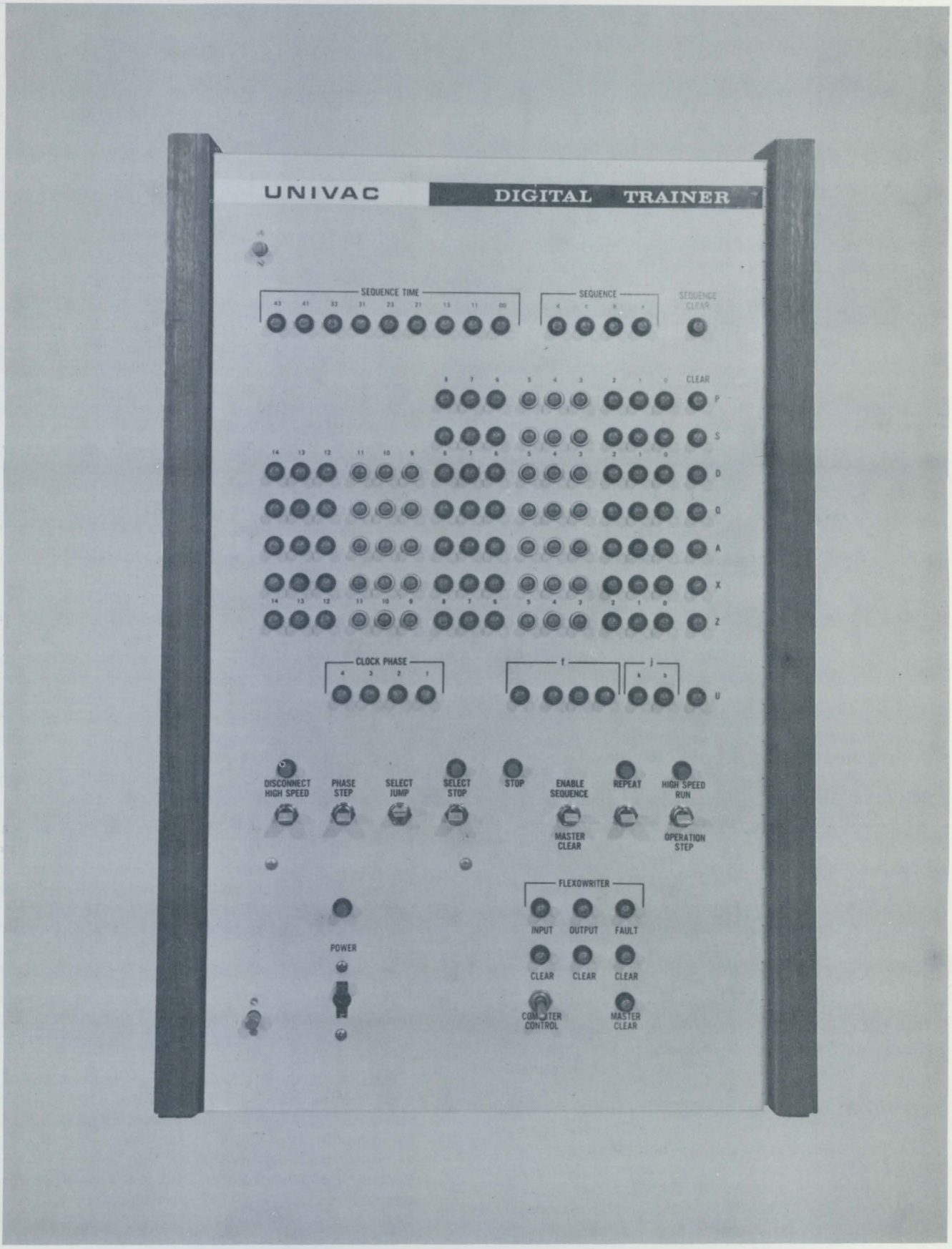


Figure 4-46. Control Console

control console for executing various routines stored, or to be stored, in the Trainer. The logic circuits (figures 5-2 and 5-3) associated with the console are integral parts of the Trainer.

(1) MASTER CLEAR. - The MASTER CLEAR switch (S3) is a momentary double throw switch with off center position. If the MASTER CLEAR switch is momentarily depressed and if the Trainer is not in high-speed operation, a Clear command is generated. The "1" output from 00G70 disables 72Y48 and 72Y43. This is a safety factor to prevent accidentally clearing the Trainer during normal operation. If the switch is depressed, contacts one through four and five close the Run flip-flop (G70) is cleared, and the "0" inputs to 72Y43 will be inverted to a "1". The signal is again inverted by the 12J43 and 14J43 to a "1" output. The output from 14J43 and 13J42 will override the HIGH SPEED DISCONNECT switch if it was activated by forcing a "0" output from 13J42. A second inversion by 14J42 will return the signal to a "1" which will clear J41 and J40 flip-flops, thereby restoring the necessary enables to the master clock to allow clock phases to be generated.

The other outputs from 14J43 to 80N00 and 80N02 supply inputs to the master clear circuits on figure 5-2. The "1" inputs to 80N00 and 80N02 are inverted by 80N01, 80N03, and 80N05 to a "1" and are applied to the various clear circuits throughout the Trainer This action will clear all registers. The master clear action will not have an effect on the stored memory section. When the MASTER CLEAR switch, S3, is momentarily placed in the up position, Enable sequence, a "0" signal is applied to 72Y48 and is inverted to a "1" which is the input to 11T00. This is the first flip-flop of the main timing chain on figure 5-9. The timing chain is now considered enabled, but it cannot be initiated, or run, until the Op Step, G73, flip-flop is set.

(2) HIGH-SPEED RUN. - High-Speed Run is the normal operational mode of the Trainer. The HIGH SPEED RUN - OP STEP switch is a three-position switch, normal center, momentary up (HIGH SPEED RUN) and momentary down (OP STEP). In the normal position, a ground level is applied to 72Y71 with "0" signals from 12J74 and the switch contacts four to six. The output of 72Y71, now a "1", is applied to the J72 flip-flop setting the one side. This flip-flop will enable the 15J73 and the 15J72 inverters. When the S1 switch is placed in the up position momentarily, contacts one to three break and one to two close. This action will remove the "0" input to 72Y74, thereby removing the enable input from 12J74 to 72Y71. The constant "1" output from 72Y71 is now removed as an input to J72. The "0" input to 72Y70 is inverted twice by 72Y70 and 12J70. When the master clock produces phase three, the "1" output of 15J72 is applied as an input to the G70 flip-flop, setting it. It also supplies "1" signals to the stop flip-flops, 00J46 and 00J47, clearing them.

With the Run flip-flop (G70) set, the output from 00G70 is a "1". This signal is an input to the 73Y70 Indicator Driver circuit, which will light the RUN indicator. The output to the three 72Y-- circuits, considered during discussion of HIGH SPEED DISCONNECT, ENABLE SEQUENCE, and MASTER CLEAR, now are disables to these circuits. Also, 00G70 is providing a constant "1" input to the Op Step flip-flop keeping it set to the one state. The output from 01G73 now a "0" will supply a constant enable to 13T11 which will allow the timing chain to be initiated (figure 5-9) after the remaining conditions are met. The input to 00G73 from 61N20 is provided by the timing chain and will be a "1" each time an instruction is executed, but as long as the input to 01G73 remains a "1" from 00G70, it will have no effect.

There are three ways the Trainer can be stopped after High-Speed Run has been initiated. These are: (1) the execution of a Program Stop instruction which will satisfy the conditions and force a "1" output from 13J47 clearing G70; (2) the execution of a Select Stop which will clear G70 from the 13J46 inverter; or, (3) placing the HIGH SPEED RUN switch momentarily in the down position. With G70 cleared, the next time the timing chain produces a "1" input from 61N20 to 00G73, the Op Step flip-flop will clear and remove the enable to 13T11, thereby stopping the timing chain.

Between the time the G70 is cleared and the timing chain clears G73, the 17J73 inverter will be fully enabled by the first phase one after G73 is set. When phase one is generated, the output of 17J73 is forced to a "1", clearing the J72 flip-flop. If the switch were returned to its normal position before G70 was cleared, there would be no effect on the clearing signal from 17J73 because of the constant "1" output from 72Y71, but if the routine or program was completed before the switch was returned to neutral, the clearing of J72 would take place and disable the 15J72 inverter. This is done to insure that G70 will not get set a second time for one setting of the switch.

(3) OP STEP. - The Operation Step mode allows the operator to control the rate of instruction execution. This feature is extremely important for debugging of programs and maintenance of the Trainer Depressing the HIGH SPEED RUN - OP STEP switch momentarily will allow one instruction to be executed at high speed. At the completion of that instruction, the Trainer will read up the next instruction and stop.

Logically, the setting of J72 will occur as it did when High-Speed Run was considered with the switch in the neutral position. Depressing S1 closes contacts four and five applying a "0" to the input of 72Y73. This signal is inverted by the 72Y73 and 12J73 inverters, resulting in a "0" applied to 15J73. The 15J73 now has "0" inputs from 01J72 and 12J73. When phase three is generated, the output from 15J73 will clear the two Stop flip-flops and the Run (G70) flip-flop. It also sets G73 which, in turn, supplies the proper enable to the timing chain, allowing it to run. The 17J73 inverter is now fully enabled by the first phase "1" after G73 is set. At this time 12J73 will output a "1" and clear the J72 flip-flop. The following phase one will force a "1" output clearing J72. This disables the 15J73 inverter so that G73 cannot be reset with the

initial depressing of the switch. When the instruction has been executed, the timing chain, via 81N20, will clear G73 removing the enable to 13T11, thus stopping the timing chain. To reinitiate the action, switch S1 must be returned to the neutral position to again set J72. Depressing the switch again will result in the execution of another instruction and the same action will result.

(4) SELECT STOP. - The Select Stop mode will allow manual control over a program. It is utilized with a 15 instruction and j equal to 3. Associated with the Select Stop mode is a Jump function which allows a jump to a special routine at the completion of a program and will stop the Trainer if the SELECT STOP is engaged. The operator then has the option of stopping the program after a given routine is completed or allowing the program to continue.

If the SELECT STOP switch (S4) is engaged, the "0" is combined with the input 91F15 to 72Y46. The 91F15 input is from the Function Code translator, translating a function code that is equal to 15. Assuming the code is being translated, the "1" output of 72Y46 would be inverted back to a "0" by 12J46 and would be one of the needed inputs to 13J46. Other inputs are 20U03, which will be a "0" if the j designator is equal to 3, and 11T41, a command from the timing chain. With all conditions satisfied, the output of 13J46, now a "1", will clear the Run flip-flop (G70) and set the Stop flip-flop (J46). From 00J46, a "1" is applied to the 76Y46 Indicator Driver circuit turning on the SELECT STOP indicator. During the reading of the next instruction, the G70 flip-flop will be cleared, removing the enable to the timing chain, stopping the program.

The jump features associated with the Select Stop instruction will not be discussed at this time, but will be included in a later paragraph dealing with all jump instructions.

(5) STOP. - The normal or program stop is associated with the 15 instruction and the designator equals two. This instruction is commonly employed at the end of a program. For a stop condition to take place, it is necessary to satisfy the 13J47 inverter. The 91F15 will be a "0" input when the Function Code translator is translating for f = 15, and the input from 20U02 is translating for j = 2. The 11T41, the result of the timing chain, will force the output of 13J47 to a "1", clearing G70 and setting J47. The "1" output from 00J47 to 77Y46 will turn on the STOP indicator and the Trainer is stopped at the next instruction.

Also associated with the Stop instruction is an unconditional jump which will be discussed, along with the SELECT JUMP switch, in a later paragraph dealing with all jump instructions.

The last control remaining is REPEAT which will be considered in the discussion of the main sequences.

4-4. OPERATION OF THE CONTROL SECTION.

a. GENERAL. - The control section, which provides control for all operations of the Trainer, consists of registers, translators, branch conditioners, and the main timing chain. It is this section that initiates and controls the sequence of events and commands that allow instructions to be completed.

b. CONTROL SECTION REGISTERS. - The control section consists of the U, P, and B registers. Each register performs an important function essential to the execution of a given instruction.

(1) B REGISTER. - The B register utilized in the Trainer is not, in reality, a register made up of hardware (flip-flops, gates, etc.) such as the other registers. Rather, address 000 of the main memory section is reserved for this purpose. During the modifying of a given instruction, the B register is automatically called from memory and utilized as a normal register. All references to the B register for indexing, entering, and storing is done by the timing chain under the influences of the sequence enables.

(2) P REGISTER. - (See figures 5-12 and 5-13.) The P register is a 9-bit non-addressable register. It is commonly called the Program Address register because it holds the address of the next instruction to be executed. The inputs to the P register are from the S register adder. The transmission to P is gated on phase one using 13P-- (S + 0, 1→ P) gates.

(a) REGISTER OPERATION. - The Command enable, S + 0, 1→ P, can be generated in the A or D sequence depending upon certain conditions. During the normal sequence of instructions, the A sequence initiates Clear S, P→ S, placing the address of the next instruction in S. Later in the A sequence, the command to Clear P, S + 1→ P is initiated. This is done in anticipation of reading the next sequential instruction. The other origin of the S + 0→ P command is the D sequence. Inverter 64N52 will have a "1" output if the instruction is f = 07 and j = 2, the Return jump instruction. However, prior to this command, the contents of the P register are transmitted to memory via the Z register. The S register at this time holds the memory address where P is to be stored as a result of Adder→ S command generated by the D sequence. During the next A sequence, the P register, already advanced by one, is transmitted to the S register, so the jump to Y + 1 is accomplished.

The P register is cleared using 10P00 prior to the data transmissions previously mentioned. In addition, master clear is accomplished by using 80N01 and normal clear can be performed by depressing the CLEAR pushbutton associated with the P register.

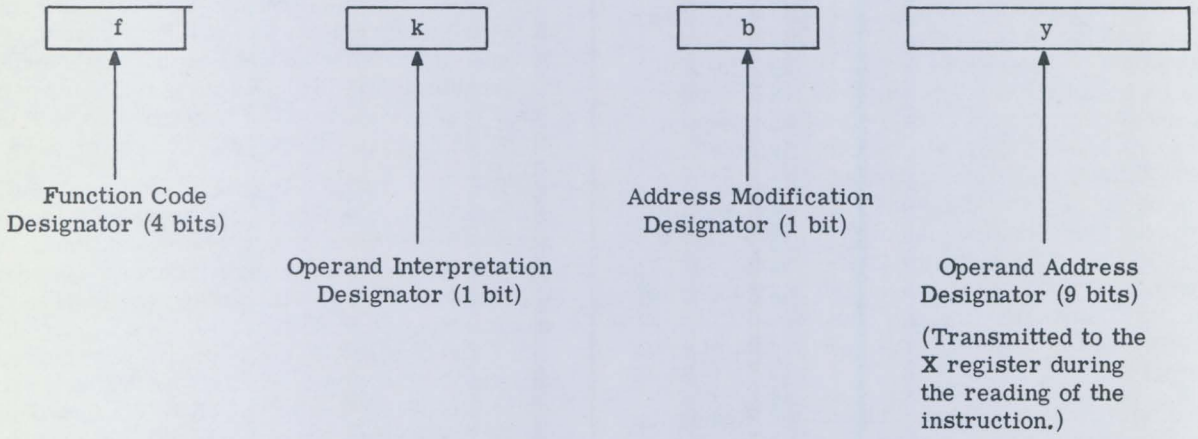
(3) U REGISTER. - (See figure 5-5.) Each instruction, as it is read from a memory location, is placed in the Z register. From this register the instruction word is broken up into two groups. The upper six bits, containing f, k, and b, or f and j, is placed in a 6-bit U register. The lower nine bits of the Instruction word are transmitted to the X register and will remain there until the Function code is translated, and it is determined what the lower nine bits will be used for in the execution of the instruction. The U register holds the instruction while it is being

executed. All operations that are necessary to execute an instruction are governed by the contents of U. The Function Code Designator circuits, consisting of the four higher order stages of U, hold the code for the current instruction. The Function code specifies the function to be performed by the instruction. A total of 16 function codes make up the repertoire of instructions. However, for some codes, the j designator is employed to expand the repertoire. With the combined use of f and j, the Trainer will respond to a total of 40 instructions.

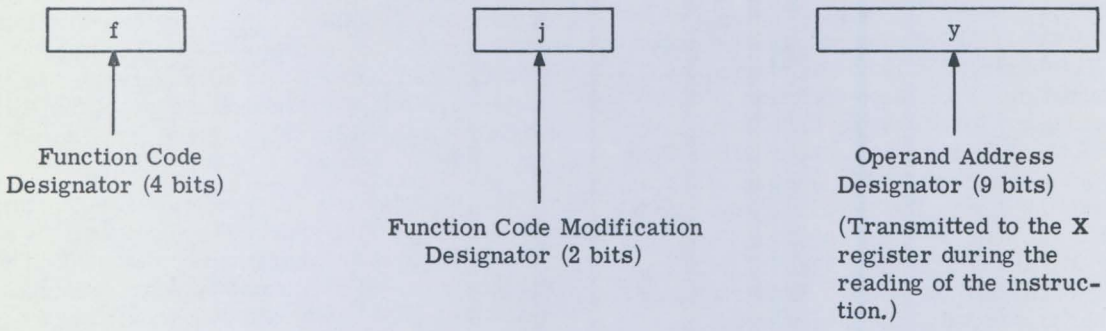
The Operand Interpretation Designator circuit specifies the manner in which the operand will be treated. The k designator will determine the location of the operand for f = 00 through 05. If k = 0, the operand is in the lower nine bits of the Instruction word. If k = 1, the operand is located in memory and the lower nine bits is the address. The exception to this is f = 17, where k is used in the same manner as j.

The Index Modification designator, b, specifies if the B register will be used to modify the lower nine bits of the instruction word before the instruction is executed. If the b designator is a 0, the B register (located at address 000) will not alter the Instruction word, but if b is equal to one, the B register is read from memory and added to the lower nine bits of the Instruction word before it is used as an operand or address.

For certain instructions, the k and b designators are used to make up the 2-bit j designator. This designator is used to modify the Function code, e.g., f = 10 and j = 0; the instruction is translated to be a right shift of the Q register; f = 10 and j = 1 will result in the right shift of A, etc. The two instruction formats employed by the Trainer are shown in figure 4-47.



Instruction Format for f = 00 - 05



Instruction Format for f = 06 - 16

Figure 4-47. Instruction Formats

The Operand Address designator comprises the nine lowest order bits of the Instruction word. These lower nine bits are transmitted to the X register when the upper six bits are loaded into the U register. While its usual function is to specify the memory address location of an operand, for certain instructions, it can be used for the operand itself.

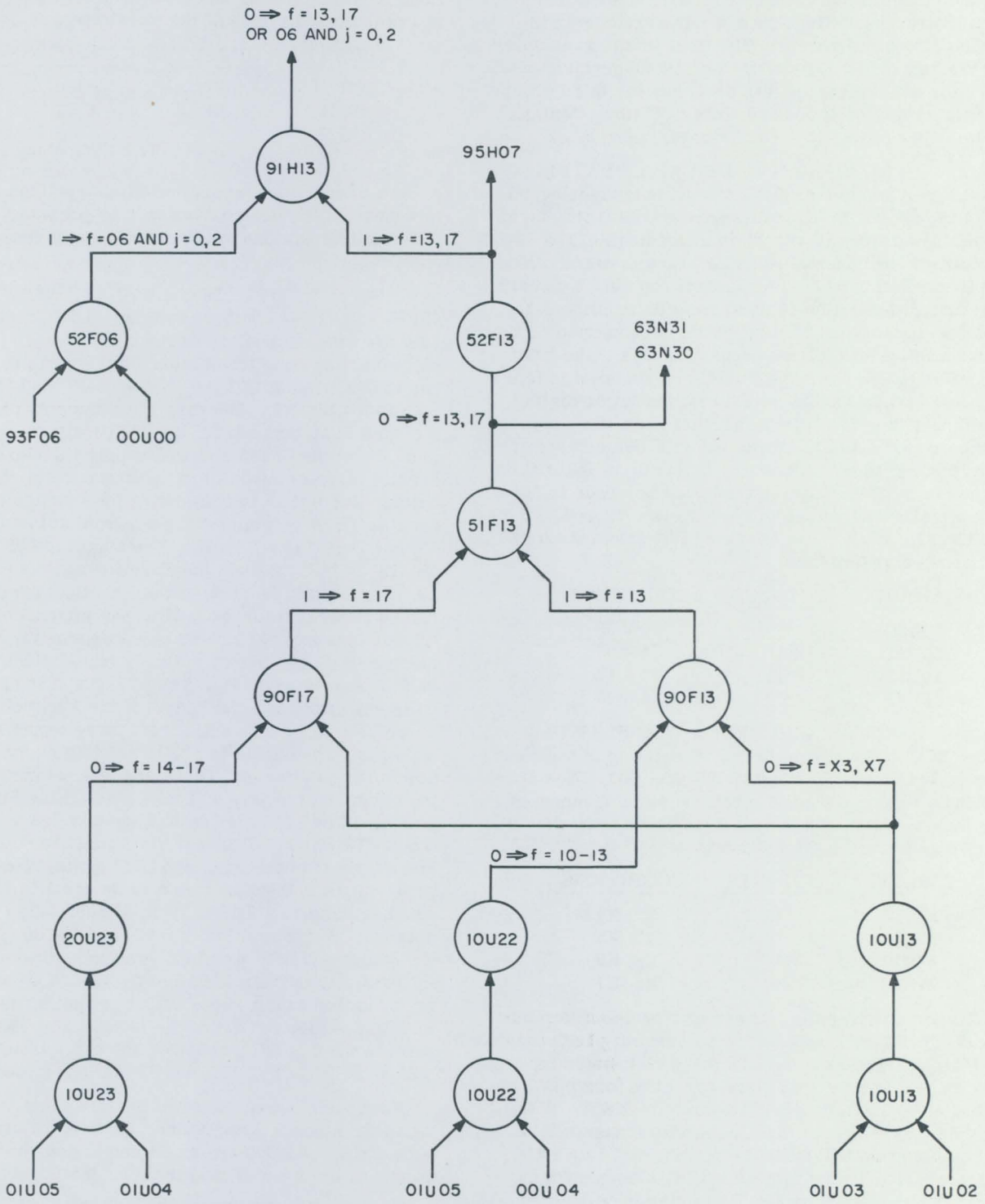


Figure 4-48. Function Code Translation (f = 13, 17, or 06 and j = 0 or 2)

(a) FUNCTION CODE TRANSLATOR. - (See figure 5-6 and 5-7.) The Function Code translator is composed of circuits which sense the six bits of U and produce outputs that effect the functions specified by each instruction. The Function Code translator is broken into three groups. The first group translates for the highest two bits of U, the second group translates for the two center bits of U, and the third group translates for the lowest two bits of U (the j designator).

(1) HIGHEST ORDER TRANSLATION. - The highest group translation is made by sensing U05 and U04. The 10U2- circuits have inputs from stages of the U register and translate accordingly. The 10U22 circuit will be considered as an example of the other 10U2- circuits. When the inputs to the 10U22 circuits are "0", U04 must be cleared and U05 must be set. Considering the input from 10U05, the Function code is somewhere in the area of 10-17 codes. The input from 00U04 will limit the Function code down to four possible codes. If U04 is cleared, the lower digits could be either "1's" or "0's"; therefore, the only numbers that could be combined with U04 cleared is zero through three. Thus, the output of 10U22 will be forced to a "1" output when the Function code is 10 through 13. The output will be inverted to a "0" by the 20U22 circuit. The following list shows the highest group translations.

CIRCUIT	OUTPUTS	CONDITIONS
10U20	"1"	00 - 03
10U21	"1"	04 - 07
10U22	"1"	10 - 13
10U23	"1"	14 - 17

(2) SECOND ORDER TRANSLATION (X0 - X7). - The second order translation of the Function code is made by sensing U03 and U02. This translation is accomplished in much the same manner as the higher order. The translation is made by the 10U1- series as shown in the following list.

CIRCUIT	OUTPUTS	CONDITIONS
20U10	"0"	X0, X4
20U11	"0"	X1, X5
20U12	"0"	X2, X6
20U13	"0"	X3, X7

To demonstrate the manner of translation for this order, 10U11 will be examined. Assuming both inputs to 10U11 to be "0's", the U03 stage of U would have to be cleared, and the U02 stage set. The four bits indicating the Function code then would be XX01. Without being able to examine U04 in this translation, it is possible that U04 could be holding a "1" or a "0", therefore, the translation of 10U11, when producing a "1" output, indicates the U register to be holding either X5 or X1.

It is the combination of the highest and second order translations that indicate the function codes and set up the proper enables to allow the proper sequence of events to execute the instruction.

(b) THE j DESIGNATOR TRANSLATION. - The remaining group translates for the j designator. This translation employs the 10U0- circuits and utilizes inputs from U01 and U00. The following list shows the translations of the j designator.

CIRCUIT	OUTPUT	CONDITIONS
20U00	"0"	j = 0
20U01	"0"	j = 1
20U02	"0"	j = 2
20U03	"0"	j = 3

As shown by the repertoire of instructions in certain cases, the j designators will select one of four possible instructions specified by a given function code.

(c) COMPLETE TRANSLATIONS. - The 90F-- and 91F-- combine a higher order translation with the second order translation for a single, complete function code translation. For example, 90F10 has inputs from 20U22 ("0" if f = 10-13) and 20U10 ("0" is f = X0, X4). Because X0 is the only unit digit included in 10 through 13, the 90F10 will be inverted to a "0" by the 91F10 and will supply its outputs to certain inverter circuits in the sequence (6-N-- circuits). The 90F13 is translating for function codes equal to 13 by combining outputs from 20U22 (f = 10-13) and 20U13 (f = X3, X7). The output of 90F13 then will be a "1" only when the Function code is equal to 13. For certain function codes, identical commands can be generated for more than one instruction. The translations are performed by combining the combinations of single and/or multiple translations; e.g., 91H13 (see figure 4-48). The 90F13 and 90F17 circuits will produce a "1" output if the Function code is 13 or 17. The 51F13 will invert these inputs to a "0" and supply the enable to 63N31 and 63N30, two inverters in the sequences. Therefore, the command generated by these circuits will take place if the Function code is 13 or 17. The 51F13 also supplies a "0" to the 52F13 which will invert the signals back to a "1" and supply it to the input of 91H13 again, resulting in a "0" output if the Function code is equal to 13 or 17. Another input to 91H13 is from 52F06. When its inputs are "0" from 93F06 (f = 06) and 00U00 (j = 0 or 2) the output of 91H13 will be forced to a "0" output. The full translation from 91H13 is f = 13, 17, 06 and j = 0 or 2. In this example the output is applied as a "0" to 64N40, which is part of the D sequence. When that point in the timing is reached, the other input to 64N40 will be a "0" and will initiate an X→D transmission.

No translation is required for those instructions using the k and b designators. Rather, the direct outputs of U01 and U00 are examined to determine the values of the k and b designators. (See figure 5-5.)

(4) S REGISTER. - Although the primary function of the S Register is to supply the memory section with address enables during a memory reference, the control and arithmetic sections also utilize the S register during certain function codes. When used as a control register, it receives the sum of the B register

and the lower half of the Instruction word (f = 00-05 and 17 and b = 1). Also, it is through the use of the S register that the B register indexing is accomplished. For function codes 10 and 11, the S and Z registers are used as arithmetic registers which store and decrement the shift count. This is accomplished by loading the shift count into the Z register. The complement of the Z register is transmitted to S, which, through the properties of the S + 0, + 1 adder will add 1 to the contents of the S register. The complement of the sum is transmitted to the Z register. The result is the subtraction of one from the initial contents of the Z register. This counting down action is performed each time a shift of one is completed. When Z_{L4} is equal to zero, the shifting action associated with the instruction is completed. More details of the S and Z registers as well as shift operations, will be discussed more fully in the following paragraphs, but it is necessary to be aware of these facts before the instructions can be examined in the sequence.

(a) S + (0, 1) ADDER. - (See figure 5-38.) The S + (0, 1) is a 9-bit, additive, open ended (no end-around carry) type adder. Its purpose is to modify the contents of S by the addition of plus one or zero. Zero is added to S when it is desired to transmit the normal contents of the S register. This would be done when f = 00 - 05 and k = 0. The sum of the B register and the lower nine bits of the Instruction word would be loaded into the S Register. Because k is equal to zero, indicating that the modified lower half of the instruction word is the operand, the transmission of S + 0 → D would occur. The S + 1 properties are used to advance the P register to the address of the next sequential instruction. The addition of one to S is also utilized during the incrementing of the B register during the execution of an 06 and j = 3 instruction.

(1) ADDER OPERATION. - The modification of the S register is determined by the G10 flip-flop. The contents of the S register is incremented by plus zero or one depending on the state of this flip-flop.

The setting of G10 to the one state will occur every time the timing chain is initiated via 13T11. This is done without consideration of the sequence enables controlling the timing chain. The clearing of G10 is controlled by 62N20, which will be a "1" when the Function code is 00 - 05 with k equal to zero, and the sequence timing chain is not under control of the A sequence. The outputs from the Adder are available to the Z, D, and P registers. The Z register has the ability to receive only the complement of S + 0 or 1, while D and P will receive the normal outputs. In considering the outputs of the Adder, it must be remembered that these outputs are actually enables to gates that control the setting of the various stages of a register (see figure 4-49).

To analyze the S adder, it must be noted that for each bit position in the S register, there are two inverters in the Adder. One of the two inverters, 20S--, is used to transmit the binary state of the flip-flops in the S register to the corresponding stage of the D register. This inverter is used when:

1. There is no carry for S-- from the stage of immediate lower order, where S-- equals any value from 01 through 08.
2. Plus one is not added for S00 to (S).

Assuming the G10 flip-flop to be cleared, the "0" input will partially enable the 20S00 inverter. Meanwhile, the 01G10 will input a "1" signal to the 21S-- circuits. The "1" input will force the 21S-- circuit to output a "0" to the remaining 20S-- circuits and also partially enable the 22S-- circuits. The remaining inputs to the 20S-- inverters are those taken from the zero side of the S register. Each stage from this point operates in an identical fashion; therefore, 2^0 and 2^1 will be considered as examples for the S + 0 explanation.

Assume the 2^0 position of the S register to be holding a binary one and the 2^1 stage to be holding a binary zero. The input to 20S00 from 00S00 (holding a binary one) is a "1" which will force 20S00 to output a "0". The 22S00 circuit is now receiving two "0" levels from 20S00 and 21S00 ("1" input from 01G10). The inverted output of 22S00, now a "1", is applied to 17Z00 which will disable the gate and leave 2^0 stage of Z cleared. It is also inverted again by 23S00 which will produce a "0" input to 17D00 and 13P00. These gates, now enabled, will allow the setting of the 2^0 stages of both D and P upon arrival of the command. In the 2^0 position, the output of 21S00 will enable the 20S01 with a "0" input and is combined with the "0" from 00S01 (holding a binary zero). The output of 20S01, being a "1", will be inverted to a "0" by 22S01 and will enable gate 17Z01, allowing that stage to be set. The 23S01 will invert the "0" output from 22S01 to a "1" which will disable gates 13P01 and 17D01. As these examples show, the Z register will receive the complement of the number contained in the S register while the P and D will receive the actual contents of the S register.

The second inverters associated with the stages of the S register are the 21S-- series and are enabled when S + 1 is utilized. The output from 01G10, now applies a "0" input to the 21S-- inverters while the 20S00 is forced to a "0" output as a result of the "1" input from 00G10. The 2^0 power of the S register is examined to determine if that stage contains a binary one by the input to 21S00 from 01S00. If 01S00 is a "0", 21S00 will output a "1" which is inverted by 22S00 and 23S00 to disable the gates feeding the 2^0 stage of both P and D. This stage of these registers will remain in the cleared state. The "1" output from 21S00, representing a carry, will disable the normal path (20S01) for 2^1 power and will result in a partial enable input to 22S01. If it is assumed that the 2^1 stage is holding a binary "0" which can absorb the carry, the 01S01 input to 21S01 will be a "1" overriding the "0" input from 01S00. The output from 21S01 will fully enable the 22S01 allowing the corresponding stages of

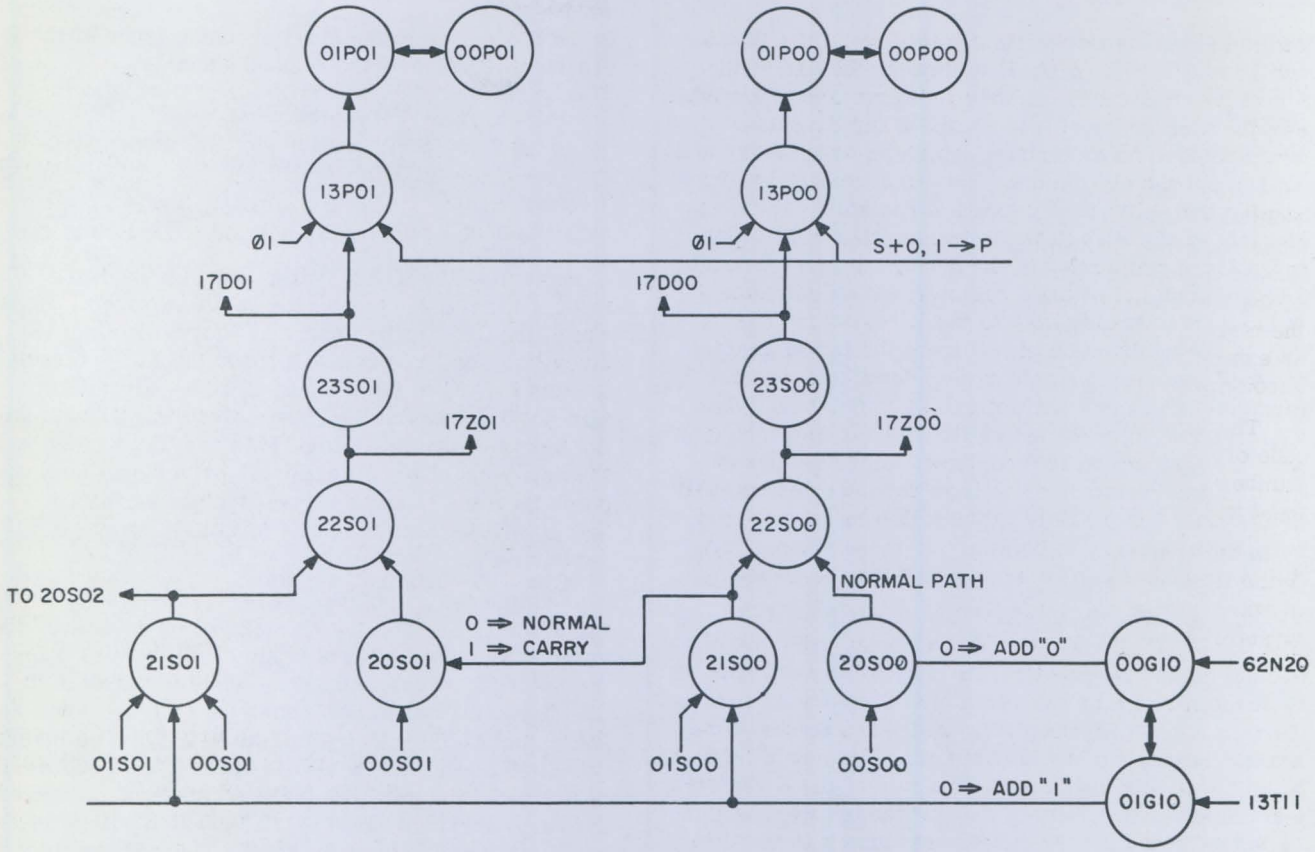


Figure 4-49. S + 0, 1 Adder and Associated Outputs

D or P to be set. The 21S01 also will enable the normal path (20S02) for the 2^2 power of S to be transmitted. All other stages of the Adder will have the normal path enabled by the "1" input from 01S01 to each of the higher order stages.

To summarize, the S + 1 action causes the Adder to examine each stage of the S register looking for the first stage equal to zero. When the binary zero is found, that stage is set to a binary one. All lower order stages (holding binary ones) are returned to zero while all higher order stages will have their normal paths enabled allowing a direct transmission of S for these stages. Table 4-3 shows the contents of the S register with plus one added to it. Both the carry and normal path (21S-- and 20S--) outputs are shown for this problem.

c. SEQUENCES. - The sequences, or main timing chain of the Trainer, is the source of all command timing. It is through this method that the logical sequence of events occur to fulfill the necessary action to perform and complete a given instruction. The timing chain, under control of the A sequence enables, will load the U register with the instruction which is

TABLE 4-3. S + 0, 1 ADDER

REGISTER STAGE	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
S REGISTER CONTENT	1	0	1	0	1	1	1	1	1
+1									
CARRY PATH OUTPUT	21S08	21S07	21S06	21S05	21S04	21S03	21S02	21S01	21S00
	0	0	0	0	1	1	1	1	1
NORMAL PATH OUTPUT	20S08	20S07	20S06	20S05	20S04	20S03	20S02	20S01	20S00
	0	1	0	0	0	0	0	0	0
RESULT S + 1 to P	P08	P07	P06	P05	P04	P03	P02	P01	P00
	1	0	1	1	0	0	0	0	0

immediately translated by the Function Code translator. It is the outputs of this translator that enable certain inverters in the timing chain to allow the proper commands to be generated when combined with the output of the timing chain and the sequence enables.

(1) SEQUENCE LOGIC TERMS. - (See figures 5-9 through 5-11.) The terms used to identify the flip-flops and inverters in the sequences are useful aids that should be employed to help the maintenance man. One of the two base letters used is the letter T reflecting the term Timing. The term 11T33 will be examined as an example of the logic translation. This term is associated with the seventh flip-flop in the timing chain.

The left-hand digit (11) identifies this to be the one side of the flip-flop; the zero side is assigned an even number (10) in this location. The right digit (33) is used to identify clock cycles and phasing.

In this example, the 3X shows the master clock is in the third cycle since the timing chain was initiated. In other words, the phase four that enabled the 13T33 inverter was generated during the third clock cycle after the 13T11 inverter at the beginning of the timing chain was fully enabled. The X3 shows the phase of the clock that will be used to load information into a register that is being affected by any command enable generated by this flip-flop; i.e., the D register receiving possible commands from the inverters 64N61, 64N60, and 62N60 under control of the 11T33 flip-flop will be loaded with new information the following phase three after 11T33 is set. See figure 4-50. Because all registers are cleared before receiving new information, the phase preceding the loading will accomplish the clearing action. Therefore, if the D register is loaded on phase three, the clearing of the D register will occur on phase two. As another example to the process and to summarize, the 11T41 flip-flop was set to the "1" state during the fourth cycle of the master clock. The registers that will be affected by any command as a result of this flip-flop being set will be cleared on phase four and loaded on phase one.

Other circuits that utilize the T term are located on figure 5-8, sequence enables. Three of the four sequence enables are identified in table 4-4.

TABLE 4-4. SEQUENCE ENABLE IDENTIFICATION

SEQUENCE	SEQUENCE ENABLE FLIP-FLOP	SEQUENCE ENABLE CONTROL FLIP-FLOP	SLAVE CIRCUITS
B	T12	T02	03T12
C	T13	T03	03T13
D	T14	T04	03T14
			03T14
			03T14
			03T14

The A sequence is identified by the --T01 circuits and will apply the necessary enables when the three sequence enable flip-flops are in the cleared or zero state. The second base letter used is N and identifies the command enable inverters of the timing chain. These inverters receive the outputs of the Function Code translator and other decision making circuits to produce command enables at the time determined by the timing chain. The left-hand digit associates the sequence enable involved with it as outlined in table 4-5.

TABLE 4-5. COMMAND INVERTER IDENTIFICATION

INVERTER	SEQUENCE
61N--	A
62N--	B
63N--	C
64N--	D

The right-hand digit has no relative meaning to the function of the inverter; rather, it is a convenient method employed in locating the inverter in the timing chain. For example, the inverters associated with the T11 flip-flop are the 6-N1- series and the 6-N2- series. By this means, each inverter can be associated with the various flip-flops of the timing chain.

(2) SEQUENCE ENABLES. - (See figure 5-8.) The sequence enables are identified as the A, B, C, and D sequences. The enables applied to the timing chain inverters originate from the state of three flip-flops, T12, T13, and T14. The remaining three flip-flops, T02, T03, and T04 are the Sequence Enable Control flip-flops. Assuming the Trainer is master cleared, each of the flip-flops will receive a "1" input to the zero side from 80N05. Now the 02T01 inverter has "0" inputs from 00T12, 00T13, and 00T14. The output of 02T01 is forced to a "1" which is applied to its slave inverters 05T01 and 03T01. The signal is returned to a "0" and is applied to the 61N-- series inverters of the timing chain as the A sequence enables.

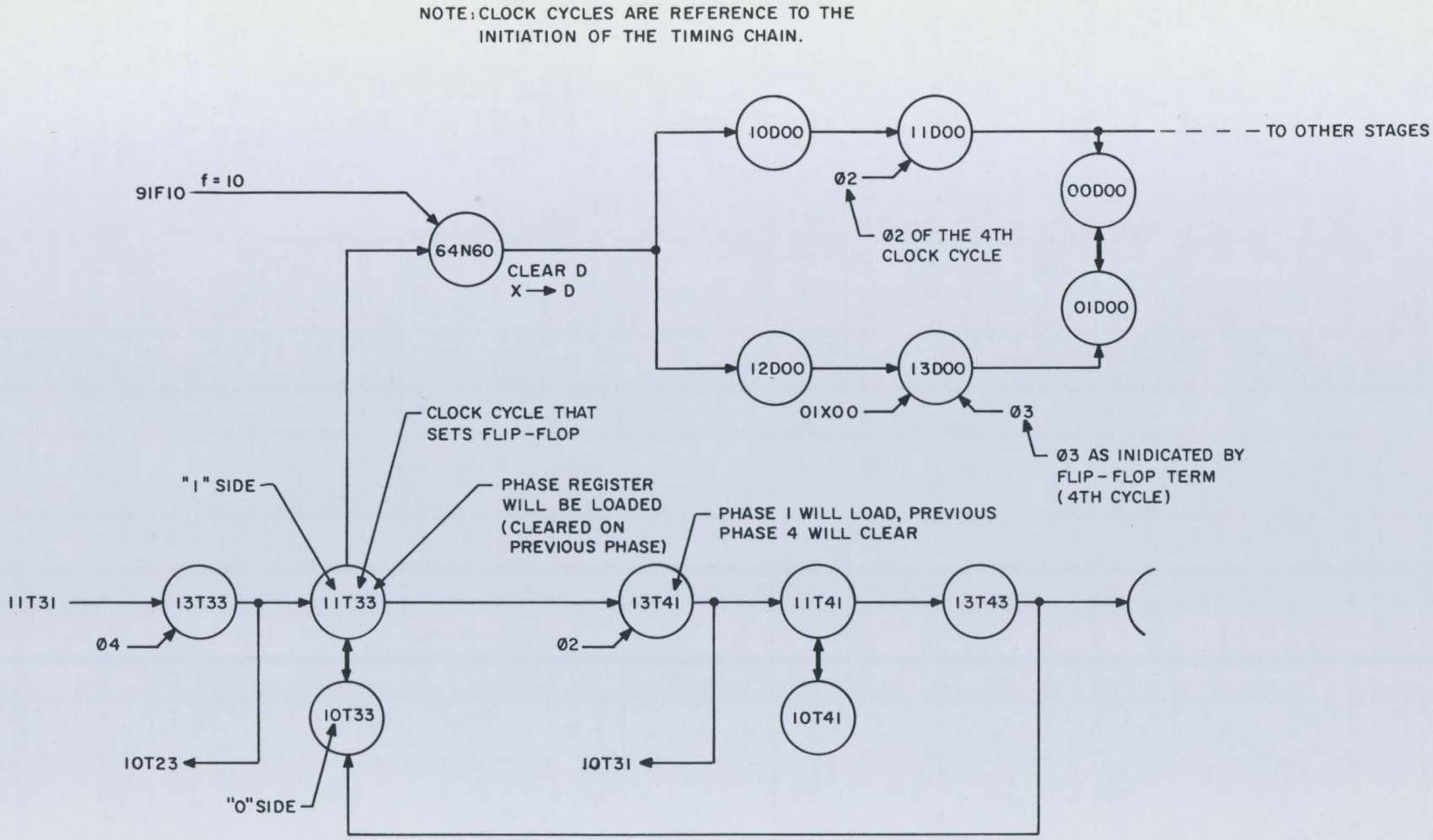


Figure 4-50. Timing Chain Terms

When the timing chain is initiated, the proper commands will be generated to accomplish the reading of the Instruction word from memory which will be inserted into the U register. Assuming the B sequence is the next sequence enables to be used, the 61N80 inverter (if f = 00-07, 17) will output a "1" setting the T02 flip-flop. This command is under the control of the T41 timing chain flip-flop which was set at phase two (see figure 4-51). The following phase 1 will set the T43 flip-flop causing a "0" input to the 13T00 inverter at the beginning of the timing chain and the 21T12 inverter.

The "0" input to 13T00 is combined with phase 2 forcing its output to a "1", thereby setting the T00 flip-flop to the one state and clearing the T41 flip-flop. The "0" input to 21T12 is inverted to a "1" at phase 3 time and will unconditionally clear the three sequence enable flip-flops if any of them were set. This action will occur at the end of each run of the timing chain. The 15T12 gate controlling the setting of the B enable flip-flop is now fully enabled with "0" inputs from 01T02, 11T00, and phase 1. The B enables generated by 01T12 and the inverter 03T12 are now applied to the timing chain. The "1" input to 02T01 from 00T12 will force the output to a "0" which is again inverted by 03T01 and 05T01 to a "1" removing the A sequence enables from the timing chain. The timing chain which was enabled, must wait until all of the conditions are satisfied controlled by 13T11 before it can be initiated again. When all of the enables are present, the B sequence enables will allow the proper commands to be generated that will accomplish another phase of the instruction execution.

The C and D sequences are enabled in the same manner, based on the same timing. It should be stated that all sequences are not required for every instruction and will result in the omission of these sequences. Table 4-6 shows the enabling of the various sequences for given instructions. The A sequence is automatically enabled when the instruction is completed by clearing the sequence enable control flip-flops controlled by 10T33. There will be no command to set any of the sequence enable control flip-flops again until the A sequence is completed.

(3) MAIN TIMING CHAIN. - (See figures 5-9 through 5-11.) The main timing chain is located on two drawings. The command inverters associated with the A, B, and D sequences are shown on figures 5-9, and 5-10 and those controlled by the C sequence enables are on figure 5-11.

The timing chain is made up of nine flip-flops which will complete one cycle in approximately 6.4 microseconds. During one cycle the timing controlled by the sequence enables are generated. If three sequences are required to complete an instruction, the timing chain will be recirculated three times, each time under the control of different sequence enables.

The timing chain is enabled by the use of the MASTER CLEAR - SEQUENCE ENABLE switch on

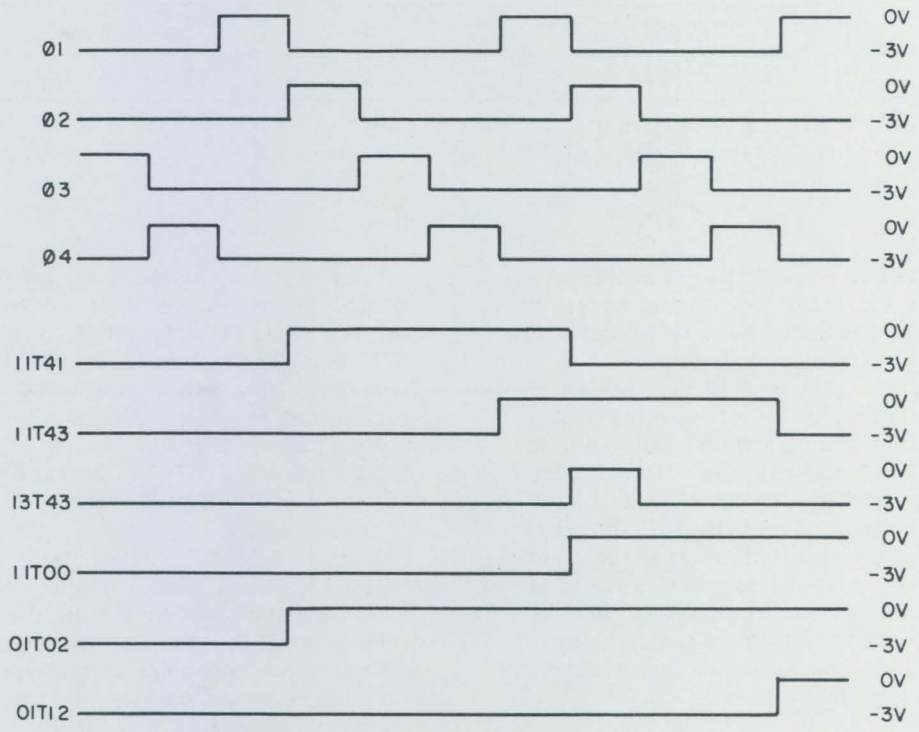


Figure 4-51. Sequence Enable Timing

the maintenance panel. After the Trainer is master cleared, the same switch, momentarily placed in the up position, will force the 72Y48 input amplifier to a "1" output. This signal is applied as one of the inputs to 11T00 setting the flip-flop to the one state. The timing chain is now considered enabled. The "0" output from 11T00 is applied as one of the five inputs to the 13T11 inverter. It is this inverter that must be satisfied before the timing chain can be initiated. The remaining four inputs that must apply "0" inputs to 13T11 are: 1) 01L17 indicating the memory section is immediately available; 2) 01G73, the run enable is present as controlled by the Op Step flip-flop; 3) 10T43, the zero side of the last flip-flop in the timing chain has been cleared; and, finally, 4) phase 2 of the master clock. When all of these conditions are present, the timing chain is said to be initiated and the setting of T11 will take place. The following phase 4 will be combined with the "0" output of 11T11 forcing 13T13 to a "1" output which will set 11T13. Under normal circumstances, each flip-flop will be set, or in the one state, for one complete clock cycle. Figure 4-52 shows the relationship of the T11, T13, and T21 flip-flops.

TABLE 4-6 SEQUENCE ENABLING

SEQUENCE ENABLED	COMMAND INVERTER	ENABLING SEQUENCE	CONDITIONS
B	61N80	A	f = 00-05, 07 and b = 1 f = 06 and j ≠ 0
C	61N82	A	f = 10, 11 and j ≠ 3 and $Z_{L4} \neq 0$ or f = 12, or f = 13 and j = 1 or 3
	62N80	B	f = 06 and j = 1 or 3
	63N80	C	f = 10, 11 and $Z_{L4} \neq 0$ *
	64N80	D	f = 00-05, 10 or 11
D	61N81	A	f = 06 and j = 0, f = 00-05, 07 and b = 0, f = 10, 11 and j = 3, f = 10, 11 and j ≠ 3 and $Z_{L4} = 0$ f = 13 and j = 0 or 2, f = 17
	62N81	B	f ≠ 06 and j ≠ 1 or 3
	63N81	C	f = 06

* Used to recirculate the C sequence during multiply, divide, and shift instructions.

Assume the T21 flip-flop to be set and the A sequence enables are applied to the command inverter. The commands to be generated from this point of the timing chain is to Clear P and transmit S + 1 → P by the 61N30 inverter. The controlling flip-flop is set on phase two; therefore, the "0" enables from 11T21 are made almost immediately available to the 61N30 inverter. Combined with the "0" enable from the A sequence, the 61N30 is forced to a "1" output. This gives the circuits most of phase two and all of phase three to properly enable the various gates and inverters to accomplish the command on receipt of the controlling phases of the clock. Phase four is the next to be generated resulting in the clearing of the P register, followed by phase one that will accomplish the transmission of S + 1 → P.

The following phase two will enable 12T21 inverter clearing T21 which will remove the enable from 61N30. Therefore, the command cannot be repeated a second time for the original setting of T21.

For certain function codes and/or decision making circuits, the timing chain can be temporarily stopped. This is controlled by any of the inputs to 13T31 from 12T31, 14T31, or 18T31 going to a "1". These conditions will be discussed in a later paragraph in this manual. When the timing chain has reached the last flip-flop (11T43), the beginning is enabled again via 13T00 to prepare for the next sequence.

(a) A SEQUENCE. - The A sequence enables are used to read the instruction word from memory, insert the function code into the U register and increment the P register. This sequence will be used as an example to demonstrate the relationship between the sequence enables, the timing chain, and command enables. For most instructions, the A sequence is identical.

Assume the timing chain to be enabled and the A sequence enables present to the command enable

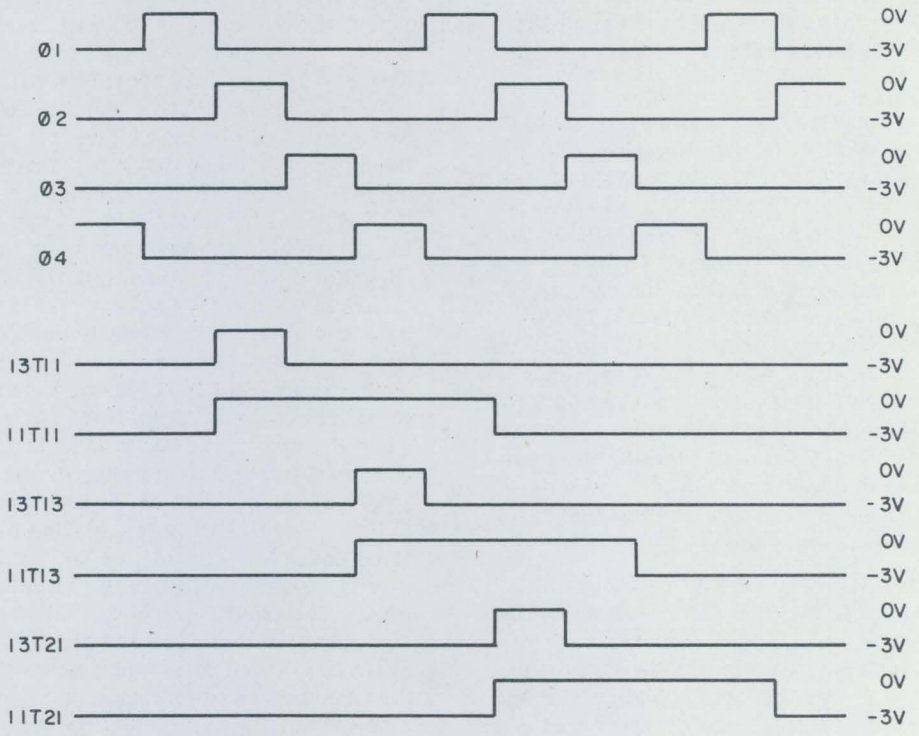


Figure 4-52. Timing Chain Timing

inverters. The 13T11 inverter will now be fully enabled with "0" inputs from 11T00, 01L17 (memory available), 01G73 (high-speed run enable), 10T43 (the last flip-flop of the timing chain is cleared), and phase two. The "1" output will set the T11 flip-flop which will apply "0" inputs to the 61N10, 61N11, 61N12, 64N10, 64N11, and 61N13 inverters. Considering each inverter individually, 61N10 will be fully enabled. The input from 00T13 (C sequence) will be a "0" input, except when the C sequence is enabled. The other input is from 60N10 which will have a "1" input from 09T14 (D sequence) forcing a "0" input to 61N10. The command to Clear S and Initiate Memory is now generated. The next inverter is 61N11 which is being enabled by the 03T01 (A sequence enable) and 00G50. The 00G50 flip-flop will input a "0" if no jump is to be satisfied. The output of 61N11 is now a "1" generating the command to Clear S and transmit the contents of P→S. The P register has been holding the address of the instruction to be read.

The next inverter, 61N12, will be disabled by the "1" input from 61N11. The inputs to 61N12 from 61N11 and 00T13 will be "1's" when the A or C sequence enables are present; therefore, this inverter can only be utilized during the B or D sequences. The 64N10 and 64N11 inverters are disabled by the D sequence enables which now will input a "1" to these circuits. The 61N13 will be enabled by the input from 00T13 which will output a "1" only during the C sequence, and the output of 60N13 which is receiving a "1" input from 07T14 (D sequence) forcing its output to a "0". The resultant command from the 61N13 inverter is to Clear Z. This is done in preparation to receive the word being read from memory. Thus, the commands generated as a result of setting 11T11 are; Clear S, P→S, Clear Z, and Initiate Memory.

On the next phase four, the timing chain is allowed to advance to T13. At the same time, the clearing to T00 takes place to insure that a second initiation of the timing chain does not take place. The output of 11T13 is applied as a partial enable to the 61N20, 62N20, 62N21, 62N22, and 64N20. All of these inverters will be disabled except 61N20. The 62N20 will be disabled by the "1" input from 02T01, which will be a "0" for all sequences except A. The 62N21 and 62N22 inverters will be disabled by a "1" input from 03T12, the B sequence enable, and 64N20 will be disabled by the "1" input from 01T14 (D sequence).

The 61N20 inverter, now fully enabled by "0" inputs from 11T13 and 03T01, the A sequence enable will generate a command to Clear D and clear the Op Step flip-flop. The G73 flip-flop (Op Step) will be restored to the set state (back to one only if the high speed flip-flop was set). If the timing chain was initiated as a result of the OP STEP switch being depressed, the G73 flip-flop will disable the 13T11 inverter at the beginning of the next sequence. Therefore, it would be necessary to depress the switch a second time to execute the instruction being read by this A sequence. When the instruction is executed, the next A sequence will read the next instruction to be executed and again clear G73 from the output of

61N20, resulting in the disabling of inverter 13T11 again. This action allows the operator to read the contents of all the registers before proceeding with the program.

The following phase two will completely enable the 13T21 inverter, setting the T21 flip-flop, and at the same time clearing T11. The "0" enables from T21 are applied to 61N30, 63N36, 63N35, 64N30, and 64N31. Of these, only the 61N30 inverter will be fully enabled. The 63N36 and 63N35 inverters will be enabled only during the C and D sequences, as controlled by the output of 62N35. The 00T14 will be a "1" input to 62N35 during the D sequence and the 00T14 will apply a "1" input during the C sequence. Because the A enables are present, the T14 and T13 flip-flops are in the cleared state; therefore, both inputs are "0's" forcing a "1" output from 62N35 causing the disable to both 63N35 and 63N36. Both 64N30 and 64N31 required the D sequence enable; therefore, they also are disabled at this time.

The 61N30 inverter receiving a "0" input for 03T01 (A sequence) will output a "1" resulting in the generation of Clear P and S + 1 → P that is being read by this A sequence. Previously the command to transmit the P → S was initiated. This command loaded the S register with the same contents that were contained in the P register. In the anticipation that the next sequential address holds the instruction that will be executed next, the P register is advanced by adding one to the contents of S via the S + 0, 1 adder during the transmission of S to P. During the execution of the instruction, except for jump instructions, the P register will remain unchanged until the next A sequence.

Running in parallel with the main timing chain is the memory timing chain which was initiated at the beginning of the A sequence. Approximately 2.5 microseconds after the command to initiate memory, the information from memory is entered into the Z register. The timing of the memory timing chain is utilized in the controlling of inverters 61L10, 61L11, and 61L12. When the memory timing chain has been advanced to the L13 flip-flop, "0" enables are applied to these inverters. The 61L12 inverter will be fully enabled with "0" inputs from 05T01, A sequence; 72Y44, REPEAT switch not engaged; and 01L13. The command output will be to clear the U register and transmit the upper six bits of Z to U. The 61L11 inverter is disabled by the "1" inputs from 60L11. Both the 00T14, D sequence enabled, and 00T12, the B sequence enable, will be "0's", resulting in a "1" output from 60L11. The 61L10 will be enabled with "0" inputs from 60L10 and 01L13. The 01T12, B sequence enable, will apply a "1" input to 60L10, forcing a "0" output from it. The resultant commands from 61L10 will clear the X register and transmit the lower nine bits of the instruction word to the X register from Z.

The main timing chain is allowed to advance to set the T23 flip-flop at phase four. The output of 13T23 will also clear the T13 flip-flop. The "0" enables from 11T23 will be applied to the 64N40 which is disabled by 01T14, the D sequence, and 61N40 now fully enabled by the 03T01 circuit, representing the A sequence. When the master clock produces phase three, the output of 61N40 will clear the abort flip-flop. This flip-flop is used to determine skip and no skip conditions. Its operation will be discussed in a later paragraph. The output of 11T23 is also applied to the input of the 13T31 inverter and is combined with the outputs of three other inverters. The 12T31 will output a "1" only if the A sequence enable is present (05T01), and the decision has been made to skip; that is, not execute the instruction being read up (the Abort flip-flop, 01G51). Assuming the Abort flip-flop to be cleared, the input to 12T31 would be a "1", forcing the output to a "0", and partially enabling the 13T31 inverter. The 14T31 will output a "1" if the B sequence enable is present (03T12), the Function code is equal to 06 (93F06), and memory is not available (00L17). The "1" input from the B sequence will force a "0" output from 14T31, applying another enable to the 13T31 circuit. The 18T31 has three inputs to consider. The 51F10 is translating for the Function code to be equal to 10 or 11; the 03T14 is the D sequence enable; and the 00L17 is the memory not available flip-flop. The input from 03T14 will be a "1" forcing a "0" output from 18T31 and at phase two, the timing chain is allowed to advance to the T31 flip-flop. The three inverters just discussed are used to temporarily stop the timing chain when the Function code is equal to 06, 10, or 11 until the memory section is available. It is also used to stop the timing chain when it has been determined that the instruction just loaded into memory is not to be executed. If this condition does exist, the T23 flip-flop will also apply an enable to the 19T00 inverter at the beginning of the timing chain where it is combined with the 03T01, the A sequence enable, the 01G51, the Abort flip-flop, and phase one. This will re-enable the timing chain to read the next instruction, skipping over the instruction just read.

Assuming the instruction being read is to be executed, the setting of 11T31 will supply "0" inputs to four inverters. The 64N52, 64N51, and 64N50 all require the D sequence enable, while the 62N50 inverter has the B sequence enable as an input. Therefore, all of these inverters are disabled during the A sequence.

Phase four will continue the advancing of the timing chain by enabling the 13T33 inverter, resulting in setting the T33 flip-flop. The "0" output of T33 will be applied to the 64N61 and 64N60, both of which require the D sequence enable. The remaining inverter, 62N60, will be enabled only during the B sequence. As in the case of the T31 flip-flop, no commands are generated at this time during the A sequence. However, from 10T33, the zero side of this flip-flop, a "1" output is applied to 00T02, 00T03, and 00T04, which will clear the sequence control enable flip-flops in preparation for enabling the next sequence. This is done at this time during every run of the timing chain.

The advancing of the chain will continue on phase two which will enable 13T41, applying a "1" input to 11T41. The three inverters supplied with "0" enables from 11T41 are controlled by the D sequence, and are therefore disabled. From the zero side of the flip-flop, a "1" output is available to 16T43 which forces a

"0" output from that inverter. The "0" enable from 16T43 is applied to eight inverters where, for the first time during the A sequence, the new function code is considered. The contents of the U register and the output of the Function Code translator will determine the next sequence that will be enabled. Of the eight inverters, the A sequence is applied to three. These are the 61N80, 61N81, and 61N82 inverters. Figure 4-53 shows the breakdown of the function codes and the sequences enabled. Jump instructions, f = 14 - 16, are not involved with the branching to the other sequences. Rather, they are handled by the A sequence at the end of the timing chain (11T43). These jump instructions will be considered in detail in a later paragraph in this section.

The 13T43 inverter is enabled by phase one which will set the final flip-flop of the timing chain. The output of 11T43 is combined with the following phase three to force 21T12 to a "1" output. This will clear the Sequence Enable flip-flops if any were set. With the Sequence Enable flip-flops clear the status of the Sequence Control Enable flip-flops will determine which sequence will be executed next.

The "0" output to 13T00 will enable the gate at the beginning of the timing chain and will set the T00 flip-flop on the next phase two. From the zero side of T43 (10T43) a "0" output is supplied to 29G50 which will sample the jump circuitry and will determine if a jump condition is to be satisfied (f = 14 - 16).

The phase one generated after the setting of the T00 flip-flop will set the next Sequence Enable flip-flop (figure 5-9), which will apply the proper enables on the command inverters in readiness for the next run of the timing chain. The "1" output of 13T00 will also clear T41. On the generation of phase one, the 12T43 inverter will clear T43.

(b) SUMMARY. - The A sequence is given the responsibility to read the next instruction word from memory and properly distribute the work into the U and X registers. It is also during this sequence that the P register is advanced and the next proper sequence is enabled. The action is controlled by command inverters enabled by the A sequence and the setting of the timing chain. These inverters are shown in table 4-7.

TABLE 4-7. A SEQUENCE COMMANDS

CIRCUIT	TIMING FLIP-FLOP	COMMANDS
61N10	11T11	Clear S - Initiate Memory
61N11		Clear S - P → S
61N13		Clear Z
61N20	11T13	Clear D - Clear Op Step flip-flop
61N30	11T21	Clear P - S + 1 → P
61L12 }	MEMORY	Clear U - Z_{U6} → U
61L10 }	TIMING	Clear X - Z_{L9} → X
		(Stop A sequence and start again if the Abort flip-flop is set. 61N40 - clear Abort flip-flop.)
10T33	11T23	*Clear Sequence Control Enable flip-flop.
61N80 }	11T41	
61N81 }		Set proper Sequence Enable Control flip-flop.
61N82 }		
21T12		*Clear Sequence Enable flip-flop.
11T43	11T43	Sample Select Stop circuits.
		Sample Stop circuits.
		Sample Jump circuits.
		*Enable Timing Chain.

*Will occur for each run of timing chain without regard to sequence enable.

The B, C, and D sequences will not be covered in the detail that was afforded to the A sequence. The action of the timing chain is the same regardless of the sequence enables applied. Therefore, only the inverters and the commands will be listed in tabular form for each of the instructions performed by the Trainer. This table will be accompanied by a brief explanation of the high points and to present the over-all picture of the instruction.

(c) INSTRUCTION MODIFICATION. - Modification of the lower nine bits of an instruction word can be accomplished when the Function code is equal to 00-05 and 17. This modification, controlled by the b designator, will or will not add the contents of the B register (address 000) to the lower half of the instruction word. Assuming the Function code read into the U register during the A sequence was within these limitations, the b designator combined with the Function code will enable one of two inverters (see figure 4-53). With the b designator equal to zero, the D sequence is directly enabled from the 61N81 leaving the D register cleared and the lower nine bits of the Instruction word in the X register. It is

Completely wrong

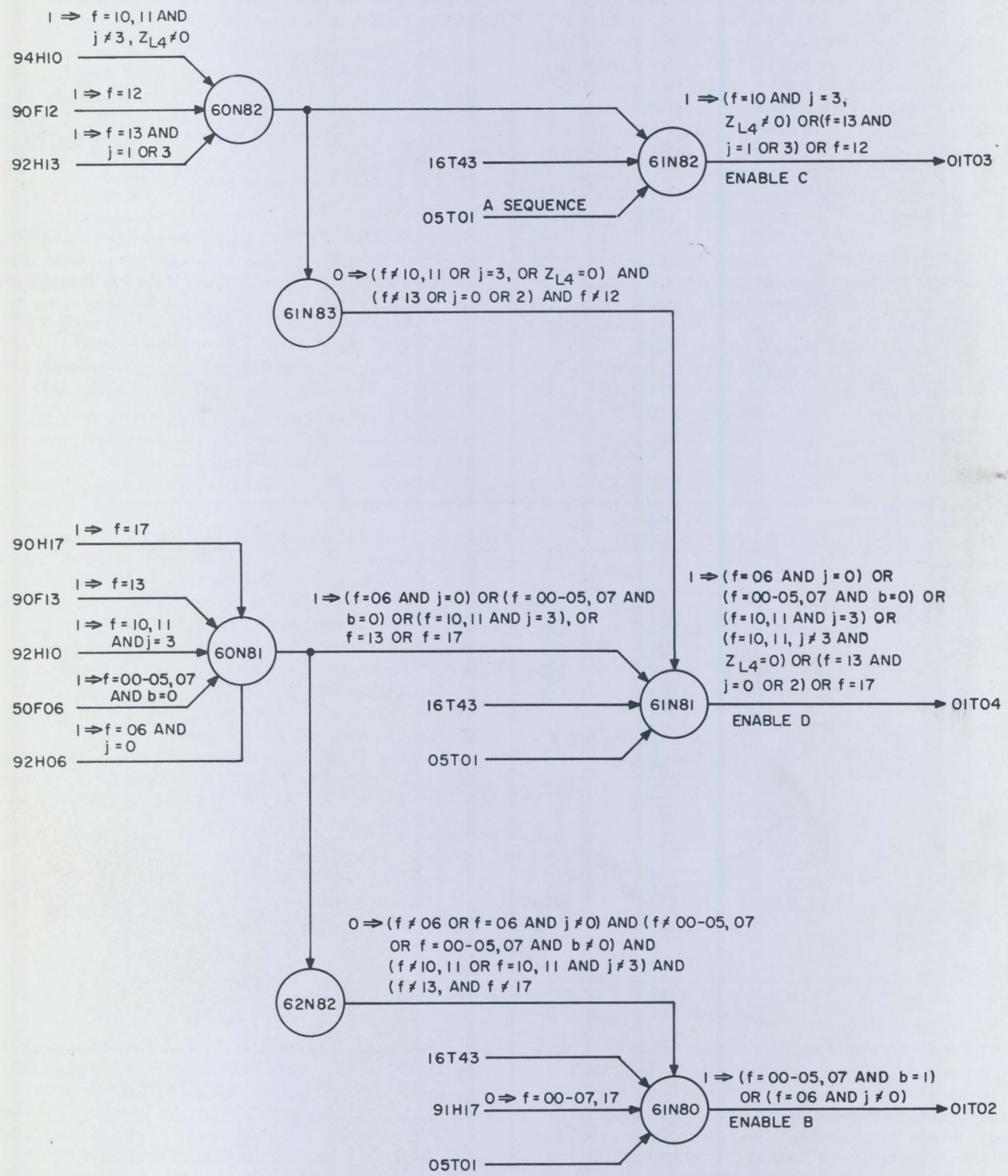


Figure 4-53. Sequence Enabling

TABLE 4-8. INSTRUCTION MODIFICATION COMMANDS

SEQUENCE	TIMING FLIP-FLOP	INVERTER	COMMAND	COMMENTS
B	11T11	61N10	Clear S, Initiate Memory	Address B Register
		61N13	Clear Z	
	11T13	62N22	X → D	Y → D
	MEMORY	61N11	$Z_U \rightarrow X_U$	
	TIMING	61N10	$Z_L \rightarrow X_L$	B register → X
	11T41	62N81	Enable D	

during this sequence that the operand interpretation is accomplished. If the b designator is equal to one, the B sequence is included to complete the modification before the instruction is allowed to continue. The enabling of the B sequence is done by the 61N80 inverter and will generate the commands shown in table 4-8.

During the A sequence, the positioning of data was: 1) The U register received the upper six bits of the instructions; 2) The X register was loaded with the lower nine bits; and 3) The D register was cleared.

The B sequence, in clearing the S register, will read the contents of the B register from address 000. The lower nine bits of the Instruction word is moved to the D register. The Z to X transmission will position the B register in the X register. The X and D registers are two arithmetic registers that feed the main Adder. Under command of the D sequence that follows next, the sum of X and D will be transmitted to the S register as the modified instruction word.

(d) OPERAND INTERPRETATION. - For function codes within the group of 00 - 05 inclusive, the location of the operand can be selected from one of two possible places as determined by the k designator. These are: 1) k = 0, the lower nine bits of the instruction word; or 2) k = 1, the contents of memory located at the address specified by the lower nine bits of the Instruction word. The interpretation of the operand will occur after modification of the instruction word takes place (if b = 1) and is under control of the D sequence. Tables 4-9 and 4-10 show the commands generated by the D sequence for k = 0 and k = 1. From the commands shown in table 4-9, the sum of X plus D will form the operand which is loaded into the S register from the output of the Adder (b = 0, y + 0; b = 1, y + b). The contents of S is then returned to the X register via the D register.

When the k designator is equal to one, the contents of the memory at the address specified by X + D must be referenced. Table 4-10 shows the commands generated by the D sequence to accomplish this task.

In this interpretation, the contents of memory is transmitted to the X register via the Z register. No matter what the point of origin is for the operand, it will be contained in the X register at the completion of the D sequence.

(e) INSTRUCTION EXECUTION. - After the operand has been positioned in the X register, the data manipulation is ready to take place. During this phase of the instruction execution, the C sequence usually has control of the Trainer. For some instructions, partial data handling is accomplished in one of the previous sequences. For the function codes equal to 00 - 05, only the D and C sequences will be considered. The A and B sequences remain unchanged for these instructions.

f = 00 - Enter Logical Product. - The logical product formed by this instruction is a bit by bit product of two data words. An example of a logical product is shown in figure 4-54.

TABLE 4-9. k = 0 OPERAND INTERPRETATION COMMANDS

SEQUENCE	TIMING FLIP-FLOP	INVERTER	COMMAND	COMMENTS
D	11T11	61N12	Adder → S	y + 0 → S
		61N13	Clear Z	
	11T13	62N20	Clear S + 1	
		64N20	Clear D, S + 0 → D	Operand (Y) → D
	11T21	64N31	Clear S, D → X	Y → X
	11T41	64N80	Enable C	

TABLE 4-10. k = 1 OPERAND INTERPRETATION COMMANDS

SEQUENCE	TIMING FLIP-FLOP	INVERTER	COMMAND	COMMENTS
D	11T11	61N10	Clear S, Initiate Memory	
		61N12	Adder → S	Address Operand
		61N13	Clear Z	
	MEMORY	61L10	$Z_L \rightarrow X_L$	Operand (Y) → X
	TIMING	61L11	$Z_U \rightarrow X_U$	
	11T41	64N80	Enable C	

X_i	101	001	110	011	011	OPERAND
Q	001	001	010	001	111	MASK
X_f	001	001	010	001	011	LOGICAL PRODUCT

Figure 4-54. Logical Product

The initial contents of X is the operand. The Q register which was previously loaded with the Mask, a constant used to form the logical product, is used to clear those stages of the X register where Q is equal to one. The remaining bits are unaltered. Therefore, X final contains the logical product of the operand and the Q register.

The first commands for this function code are generated as part of the D sequence during the reading of the operand which was transmitted to X under the control of the 64N31 inverter. When the timing chain is advanced to the T41 flip-flop, the 64N70 will become fully enabled with "0" inputs from 91F00 ("0" when f = 00), 05T14, the D sequence enable, and T41. The command to transmit Q to X is generated and will clear the X register where the stages of Q are equal to "1's".

As a point of clarification, the X register is the only register in the Trainer that is cleared to all ones (see figure 4-55). When the X register receives data from the transmitting register, it is always from the "0" side to the "0" side of X. This feature allows instructions such as the logical product to be performed. The X register will be discussed in greater detail as part of the arithmetic section.

The C sequence is enabled from the D sequence and the timing chain begins another cycle. The command inverters enabled by the C sequence for this function code are 63N57 and 63N55. Both of these inverters receive an enable from 63N56, which requires a "1" input from any one of three circuits. The 90F03 will be a "1" when the Function code is equal to 03; the 52F00 will be a "1" when the Function code is 00, 01, 04, or 05, and 90H13 will be a "1" when the function code is equal to 13, and j is one or three. For a 00 function code, the input from 52F00 will be a "1" and will force 63N56 to a "0" output. This enable is applied to both 63N57 and 63N55. The 63N57 will also receive a "0" input from 90F03 (f ≠ 03) and the Clear A command is generated. The 63N55 inverter will also be fully enabled with "0" inputs from 63N56 and 51F04 (f ≠ 04, 05). The command to transmit X to A is produced. This will position the logical product in the A register thereby completing the instruction.

At the end of the timing chain, the C sequence does not have the ability to enable another sequence and will unconditionally be cleared by 11T43. The clearing of the Sequence Enable flip-flop will allow the A sequence enables to be applied to the timing chain and the next instruction is read from memory.

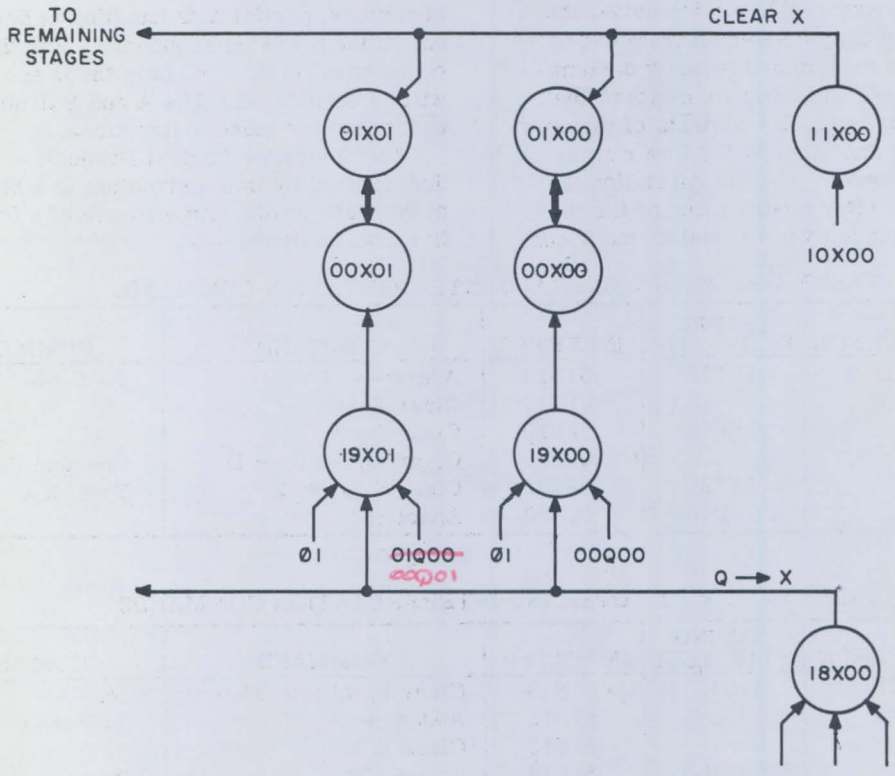


Figure 4-55. X Register Logic

f = 01 - Enter A. - This instruction is used to enter the A register with the operand as determined by the k and b designators. The C sequence will utilize the same inverters as during the logical product instruction. With the operand in the X register, the enabling of the 63N57 and 63N55 will generate the commands to Clear A and transmit X → A.

f = 02 - Enter Q. - The Enter Q instruction has the same logical function as the 01 Function code except the Q register is involved instead of the A register. The C sequence will control the movement of the operand from the X register by the use of the 63N30 inverter. When the timing chain has advanced to the T23 flip-flop, the 15T23 inverter will be fully enabled with "0" inputs from 11T23 and 03T13. The "1" output will be inverted by 16T23 which will apply "0" inputs to six command inverters. The 63N30 inverter is now fully enabled with "0" inputs from 16T23 and 53F10 (a "0" output when f = 02, 10, or 11, and j ≠ 1). The Clear Q and transmit X → Q commands are now generated completing the instruction.

f = 03 - Selective Set. - The logical operation of this instruction is to superimpose the operand with the contents of the A register. This is accomplished by transmitting the X register (operand) contents to the A register without clearing A. Those stages of A that are equal to one will remain unchanged, while those stages that are equal to zero will be changed to ones if the operand is a one in that stage.

The modification of the instruction and the operand interpretation is accomplished in the usual way during the B and D sequence. The C sequence is allowed to advance to T33 without a command generated for this function code. The "0" enable from 11T33 will be combined with the "0" from the C sequence (01T13) to force a "1" output from 15T33. The "1" is returned to a "0" enable by the 16T33 which will apply a partial enable to four command inverters. With f = 03, the input to 63N56 from 90F03 will be a "1" forcing its output to a "0". The "0" enable is applied to 63N57 and 63N55. The 63N57 inverter will be disabled by the 90F03 ("1" if f = 03) and will not allow the A register to be cleared. The 63N55 will be fully enabled with "0" inputs from 63N56, 16T33, and 51F04 ("0" if f ≠ 04, 05). The A register is now loaded with the contents of X without being cleared, resulting in setting A_n for Y_n = 1.

f = 04, 05 - Add, Subtract. - In the execution of the Add instruction, the operand is added to the contents of the A register. Because the Adder receives its inputs from the X and D registers, the contents of the A register must be moved to the X register after the contents of X is moved to D. The operand and instruction modification is accomplished in the usual way for this instruction. Table 4-11 shows the commands generated by the C sequence during the Add instruction.

TABLE 4-11. ADD INSTRUCTION COMMANDS

SEQUENCE	TIMING FLIP-FLOP	INVERTER	COMMANDS
C	11T13	63N10	Clear D, X → D
	11T21	63N22	Clear X, A → X
	11T33	63N57	Clear A
	11T53	63N51	Adder → A

It should be noticed that the T53 flip-flop is employed during this instruction. The addition of this flip-flop is done to allow additional time for the Adder to complete the addition before the gating to the A register occurs. This flip-flop is set by 15T33 and will apply a "0" enable to the input of 63N58. The input from 11T00 will not be a "0" until the timing chain is enabled again, approximately 2.8 microseconds after the T53 flip-flop is set.

For the Subtract instruction, f = 05, the same commands are issued with the exception of the X → D controlled by 63N10. Instead, the operand is complemented during the transmission from X to D. The 63N11 inverter will be enabled with the T13 flip-flop set. The "0" inputs to 63N11 are: 03T13, the C sequence enable; 91F05, f = 05; and 11T13. The command to send X' → D is issued in preparation for the subtract.

f = 06; j = 0 - Store Y in B. - For this instruction, and all others utilizing the j designator, the operand is fixed; that is, the option in obtaining the operand by the use of the k designator is removed. The j designator is employed to increase the repertoire of instructions by altering the Function code. This designator is translated from the two lowest order bits of the U register that were used to indicate the values of the k and b designators in the previous instructions. Table 4-12 shows the sequence and commands employed during this instruction. The D sequence is enabled by the A sequence after the instruction is read from memory.

TABLE 4-12. f = 06; j = 0 - STORE Y IN B INSTRUCTION COMMANDS

SEQUENCE	TIMING FLIP-FLOP	INVERTER	COMMANDS	COMMENTS
D	11T11	61N10	Clear S, Initiate Memory	Address B register.
		64N10	Inhibit Memory lower	Destroy the contents
		64N11	Inhibit Memory upper	of the B register.
		61N13	Clear Z	
	11T23	64N40	X → D	Operand to D register
	11T31	64N50	D → Z	Operand to Z register

The 61N10 command inverter will clear the S register (the address of the B register) and reference memory. Inhibiting memory upper, and memory lower, will not allow the contents of that memory address location to be entered into the Z register during the read cycles; therefore, the Z register will remain cleared. The operand, to be stored in the B register, is moved from the X register to D and finally to the Z register. This is done before the write cycle of the memory timing chain is initiated. The result will be that during the write cycle, the contents of Z is written into address location 000, which will load the B register with the operand. For this instruction, the A and D sequence enables are the only ones required.

f = 06; j = 1 - B Register Index. - The execution of this instruction will subtract the contents of the B register by the operand (B - Y). If, as a result of the subtract, the B register is equal to zero, the next sequential instruction will be skipped; that is, it will not be executed. If the skip is not satisfied, the result of the subtraction will be entered into the B register. Table 4-13 shows the commands generated as a result of this instruction.

TABLE 4-13. f = 06; j = 1 B-REGISTER INDEX INSTRUCTION COMMANDS

SEQUENCE	TIMING FLIP-FLOP	INVERTER	COMMANDS	COMMENTS
B	11T11	61N10	Clear S, Initiate Memory	Address B register.
		61N13	Clear Z	
	11T13	62N21	X' → D	Complemented operand to D
	MEMORY	61L11	$Z_U \rightarrow X_U$	B register → X
	TIMING	61L10	Clear X, $Z_L \rightarrow X_L$	
	11T23	14T31	Disable 13T31	Stop timing chain until memory cycle is complete.
	11T31	62N50	Set Abort if B - Y = 0	Skip satisfied.
C	11T41	62N80	Enable C	
	11T21	63N23	Clear S, Adder → S	B - Y → S
	11T33	63N52	Clear D, S → D	
D	11T41	63N81	Enable D	
	11T11	61N10	Clear S, Initiate Memory	Address B register.
		64N10	Inhibit Memory lower	Destroy B register content.
		64N11	Inhibit Memory upper	
		61N13	Clear Z	
	11T31	64N50	D → Z	

Under the control of the B sequence, the B register is read from address 000 (61N10). The operand, the lower nine bits of the instruction words, is complemented during the transmission to the D register in preparation for the subtraction process (62N21). The contents of the B register is then loaded into the X register and its outputs, combined with the outputs of the D register are applied to the Arithmetic adder. The decision to skip or not is determined by the 62N50 inverter. The inputs, 50A71 and 50A81, are taken directly from the Adder circuitry and will apply "0" enables if the difference between X and D is equal to "0". Setting the Abort flip-flop (01G51) will cause the Trainer to skip over the next sequential instruction.

The C sequence is enabled by the B sequence and will transmit the results of the subtraction to the S register (63N23). The 63N52 inverter will move the data from the S to the D register in preparation to store the difference back into the B register. The D sequence is the enabled to complete the instruction.

Address 000 (B register) is again referenced, but this time the contents of B is destroyed by inhibiting memory upper and memory lower (64N10 and 64N11). This action will now allow the data from memory to enter the Z register during the read cycle of the memory timing chain. Before the write cycle occurs, the transmission of D to Z will occur and the new value is written into the B register location.

f = 06; j = 2 - Enter B Register. - In the translation of this instruction, the lower nine bits of the instruction word becomes the address of the operand. This operand will be inserted into the B register. Table 4-14 shows the commands generated by this instruction after the B sequence is enabled.

TABLE 4-14. f = 06; j = 2 - ENTER B REGISTER INSTRUCTION COMMANDS

SEQUENCE	TIMING FLIP-FLOP	INVERTER	COMMAND	COMMENTS
B	11T11	61N10	Initiate Memory, Clear S	
		61N12	Adder → S	Address Operand
		61N13	Clear Z	
	11T13	62N21	X' → D	
	MEMORY	61L10	Clear X, Z_L → X_L	Operand → X
	TIMING	61L11	Z_U → X_U	
	11T23	14T31	Disable 13T31	Stop timing chain until the memory cycle is completed.
	11T33	62N60	Clear D	
	11T41	62N81	Enable D Sequence	
D	11T11	61N10	Initiate Memory, Clear S	Address B register.
		64N10	Inhibit Memory Lower	Destroy contents
		64N11	Memory Upper	of B register.
	11T23	64N40	X → D	Operand to D register
	11T31	64N50	D → Z	Operand to Z register before write cycle
	11T43	21T12	Clear B, C, or D sequence enable flip-flops	

During the B sequence, the operand is read from memory and is temporarily stored in the X register. The D sequence will address the B register, but will destroy its contents by inhibiting memory from the Z register during the read cycle of the memory timing chain. During the read cycle, the operand is moved from the X to the D register. Before the write cycle begins, the operand is transmitted to the Z register by the 64N50 inverters. It is during the write cycle of the memory timing chain that the operand is written into the B register.

f = 06; j = 3 - B Register Index on Y. - This instruction will compare the operand to the B register. If they are equal (Y = B) the next sequential instruction is skipped. If the operand is not equal to the B register, the B register is incremented by one and returned to address 000. Table 4-15 shows the commands generated by the sequences of this instruction. The B sequence is enabled by the output of 61N80 at the completion of the A sequence.

Although it is not reflected in the commands for this instruction, the S + 0, 1 adder is set to a plus one each time the timing chain is initiated from 13T11. The only means to clear the G10 flip-flop is by the use of the 62N20 inverter. This circuit will be enabled only when the upper bit of j or k is a "0" (02U01), the Function code is 0X (02U05), and the A sequence is not enabled. For this instruction, the output of 02U05 would be a "0" and because the C sequence enabled, the input from 02T01, would also be a "0". The input from 02U01 would be a "1" for this instruction because of the j designator's value of three. Because of this, the G10 flip-flop will remain set and add one to the contents of the S register during its transmission to the D register.

f = 07; j = 1 - Store B Register. - With the utilization of this instruction, the contents of the B register can be stored at any address location as specified by the lower nine bits of the Instruction word. Table 4-16 shows the commands generated for the execution of this instruction. The B sequence will be enabled at the completion of the A sequence by the use of the 61N80 inverter.

Under the control of the B sequence, the B register was read to the Z register. The 61L10 inverter, normally responsible for moving the lower nine bits of the

TABLE 4-15. f = 06; j = 3 - B REGISTER INDEX INSTRUCTION COMMANDS

SEQUENCE	TIMING FLIP-FLOP	INVERTER	COMMAND	COMMENTS
B	11T11	61N10	Initiate Memory, Clear S	Address B register
		61N13	Clear Z	
	11T13	62N21	$X' \rightarrow D$	Complemented operand to D
	MEMORY	61L10	Clear X, $Z_L \rightarrow X_L$	B register $\rightarrow$ X
	TIMING	61L11	$Z_U \rightarrow X_U$	
	11T23	14T31	Disable 13T31	Stop timing chain until the memory cycle is completed.
	11T31	62N50	Set Abort if Y = B	Sample possible skip.
	11T33	62N60	Clear D	
	11T41	62N80	Enable C sequence	
C	11T21	63N23	Clear S, Adder $\rightarrow$ S	B register $\rightarrow$ S
	11T33	63N52	S $\rightarrow$ D, Clear D	S + 1 to the D register (The B register is now incremented.)
	11T41	63N81	Enable D sequence	
D	11T11	61N10	Initiate Memory, Clear S	Address B register
		64N10	Inhibit Memory Lower	Destroy B register contents.
		64N11	Inhibit Memory Upper	
		61N13	Clear Z	
	11T31	64N50	D $\rightarrow$ Z	

TABLE 4-16. f = 07; j = 1 - STORE B REGISTER INSTRUCTION COMMANDS

SEQUENCE	TIMING FLIP-FLOP	INVERTER	COMMAND	COMMENTS
B	11T11	61N10	Initiate Memory, Clear S	Address B register
		61N13	Clear Z	
	11T13	62N20	Clear S + 1	
	MEMORY	61L11	$Z_U \rightarrow X_U$	Upper 6 bits of the B register $\rightarrow$ X
	TIMING			
	11T41	62N81	Enable D	
D	11T11	61N10	Initiate Memory, Clear S	
		61N12	Adder $\rightarrow$ S	Operand to S
		64N10	Inhibit Memory Lower	Destroy old contents of address specified by Y
		64N11	Inhibit Memory Upper	

addressed word to X is disabled by the combined inputs of 91F07 (f = 07) and 01T12 (B sequence) to the 60L10 inverter. The upper six bits are transmitted to the X register in the normal manner.

To summarize, the B register, which is to be stored, is still located in the Z register. The X register contains the lower nine bits of the Instruction word with the upper six bits of the B register in X upper. When the command is to transmit the Adder output to S, the sum of X lower nine bits and the cleared D register are transmitted to the S register, resulting in X lower nine to S. If the X register held something other than "0's" in the upper six bits because of the $Z_U \rightarrow X_U$ transmission, they would now be ignored.

The second reference to memory, under control of the D sequence, will address the location of memory specified by the operand. The contents of this location will be destroyed because of the inhibiting of memory upper and memory lower during the read cycle. The contents of Z (the B register) will be written into the address location during the write cycle of the memory timing chain, completing the instruction, Store B at address specified by Y.

If the 07 instruction is executed with the value of j equal to zero, the same commands will be generated along with one additional command. The 61N13 inverter will be disabled by the 60N13 circuit if j is equal to one. Examination of the 60N13 circuit shows that the direct translation of j (20U01) will be a "1"

input, if j is equal to zero. Therefore, the 61N13 inverter will be enabled when f = 07; j = 0, and will clear the Z register during the D sequence. At this point of the instruction execution, the contents of the B register was stored in the Z register. With Z now cleared, the write cycle of the memory timing chain will write the contents of Z into the location specified by the operand resulting in the clearing of that memory location.

f = 07; j = 2 - Return Jump. - The jump condition is satisfied by this instruction and will result in the address of the next sequential instruction (S + 1) to be stored, so that this instruction can be executed when the subroutine (specified by the Return Jump instruction) is completed. Table 4-17 shows the commands generated by the Return Jump instruction after the D sequence is enabled by A.

TABLE 4-17. f = 07; j = 2 - RETURN JUMP INSTRUCTION COMMANDS

SEQUENCE	TIMING FLIP-FLOP	INVERTER	COMMAND	COMMENTS
D	11T11	61N10	Clear S, Initiate Memory	Address memory specified by Y.
		61N12	Adder → S	
		64N10	Inhibit memory lower	Destory memory lower at address specified by Y.
		61N13	Clear Z	
	11T21	64N30	P → Z	Store P at address Y.
	11T31	64N52	Clear P, S + 1 → P	Y + 1 → P
	11T43		Enable A sequence	

The lower 9 bits of address Y is destroyed in this instruction by the Inhibit command generated from the term 64N10. Because the Z register remains cleared during the memory cycle, the contents of the D register (address of the next sequential address) are transmitted to the Z register before the write cycle is started by the memory timing chain. This action will store the P register in the memory address location specified by Y. The S register which contains Y, is transmitted to the P register via the S + O, 1 adder, adding one to Y. During the next A sequence, the P→S command will read the instruction at Y + 1 and execute it. When the subroutine is completed, the stored P register can be read and, through the properties of another jump instruction, return to the main program.

This instruction is usually used when it is desired to continue the main program with the next sequential instruction after a subroutine has been completed. Considerations should be given to the possible destruction of the various registers such as A, Q, and B during the subroutine. A good practice is to store these registers in memory during the first instructions of the subroutine, execute the subroutine, and finally, replace the original contents of A, Q, and B before returning to the main program.

f = 10; j = 0 - Right Shift Q. - During the right shift instructions, the lowest order stage of the register (2^0) is open ended; that is, data from the 2^0 position is destroyed as the bits are shifted. The sign position of the register (2^{14}) is extended through the register for each shift executed. For example, if 2^{14} stage is equal to one, indicating a negative value, and a shift of three places is to be executed, 2^{14}, 2^{13}, and 2^{12} stages would be overshifted and 2^2, 2^1, and 2^0 would be destroyed.

The shift count is controlled by the S + 0, 1 adder, the Z and S registers. By the use of these circuits, the Z register is decremented by one each time the shift of one is completed. When Z lower four bits are equal to zero, the shift instruction is considered complete and the next A sequence is enabled. Table 4-18 shows the sequence of events for the shift instruction. It must be remembered that the lower nine bits of the instruction is still contained in the Z register as well as the X register, at the completion of the A sequence. The maximum number of shifts that can be executed is 15 decimal, because only the lower four bits of the Z register are examined.

The shifting properties of the Trainer are obtained by transmitting the contents of the Q register to the X register, misplaced one stage to the right (f = 10; j = 0) or left (f = 11, j = 0). The direct transmission path is utilized to return the contents of Q, now shifted one place, back to the Q register. Figure 4-56 shows a data word shifted two places by the transmission of the X and Q registers. This shift action continues until the lower four bits of the Z register are equal to zero. The decrementing of the Z register is shown in figure 4-57.

During the right shift of the A register (f = 10; j = 1), the same transmissions occur, except the A register is involved as was the Q register in the example shown. The command inverters (63N20 and 63N30) are disabled for the shift of A and are replaced with 63N40 (A_{R1} → X) and 63N50 (Clear A, X → A).

TABLE 4-18. f = 10; j = 0 - RIGHT SHIFT Q INSTRUCTION COMMANDS

SEQUENCE	TIMING FLIP-FLOP	INVERTER	COMMAND	COMMENTS
C	11T21	63N20	Clear X, $Q_{R1} \rightarrow X$	Shift of one place during transmission to X.
		63N24	Clear S, $Z_{L4}' \rightarrow S$	Complemented $Z_{L4} \rightarrow S$
	11T23	63N30	Clear Q, $X \rightarrow Q$	Return shift data to Q
	11T31	63N43	Clear Z, $(S_{L4} + 1)' \rightarrow Z$	See explanation.
	11T41	63N80	$Z_{L4} \neq 0$, Enable C	Shift again.
			$Z_{L4} = 0$, Enable A	Shift completed.

During the Right Shift AQ instruction (f = 10; j = 3), both the A and Q registers are shifted to the right. The sign position (2^{14}) of the A register is extended and the lowest order stage of Q (2^0) is open ended. For example, if a right shift of AQ was executed with a shift count of 15, the A register would be shifted into the Q register and the initial contents of Q would be destroyed.

Under the control of the C sequence, the Q register is the first to be shifted by the enabling of the 63N20 command inverter. The commands to Clear X and transmit $Q_{R1} \rightarrow X$ will occur. As the Q register is transmitted, misplaced by one stage to the right, the 2^{14} stage of X is loaded with the 2^0 stage of the A register. The shifted contents of Q is returned to that register by the 63N30 inverter which will Clear Q and transmit $X \rightarrow Q$. The shifting of the A register is next with the output of 63N40, Clear X, $A_{R1} \rightarrow X$. The return of data to the A register is accomplished by the 63N52 inverter commands, Clear A, $X \rightarrow A$. As it can be seen, the shifting of Q and A utilizes the same inverters employed when each register was shifted individually. The shift count is maintained in the normal manner as illustrated in figure 4-57.

The recycling of the C sequence is controlled by the 63N80 inverter during all shifts. The inputs to this circuit will remain "0's" from 01T13 (C sequence), 55F10 (f = 19, 11), and 30Z00 ($Z_{L4} \neq 0$), until the 30Z00 circuit senses that the Z register has been decremented to "0". When the C sequence cannot be enabled, the A sequence enables appear in the timing

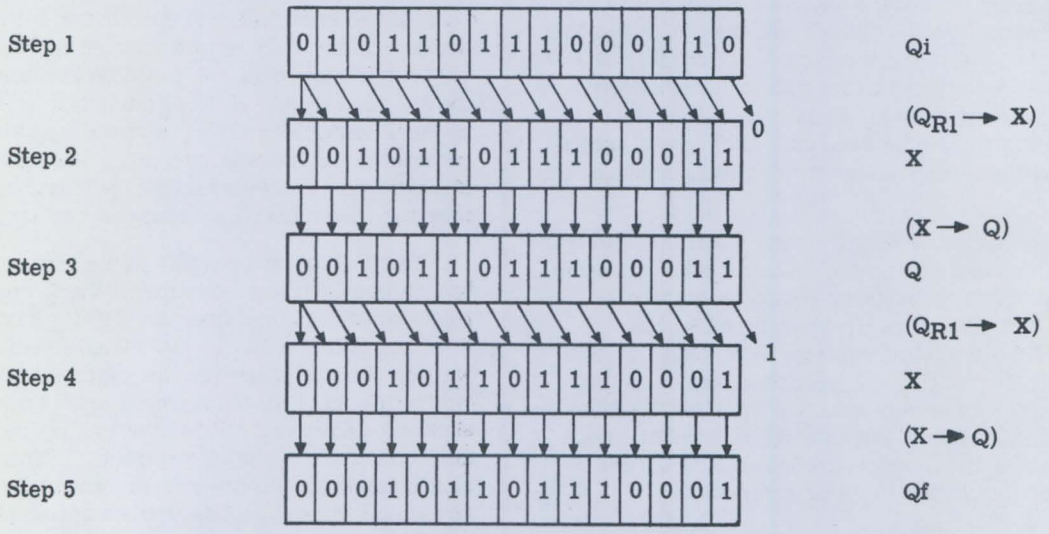


Figure 4-56. Right Shift Q Transmissions

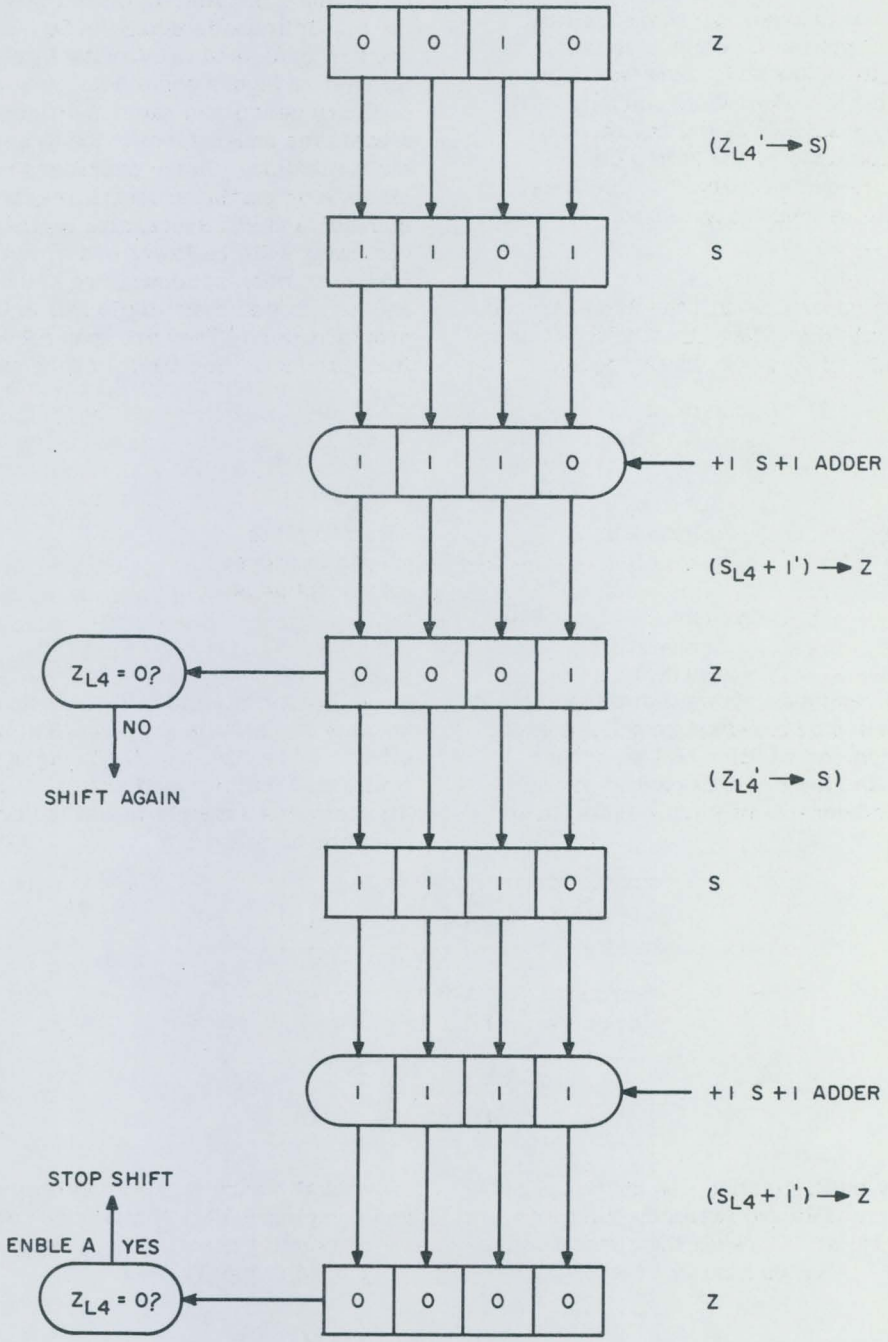


Figure 4-57. Shift Count Control

chain to read the next instruction. Thus, the number of shifts to be performed will result in the number of times the C sequence will be recirculated.

For any shift (not multiply or divide) to be performed, the initial contents of the Z register (shift count) must be unequal to zero. Assume the shift instruction was read into the U register and, because of program modification, the shift count was found to be equal to zero (shift A and/or Q no places). To prevent any erroneous shifting of the register, the Z_{L4} translator is examined by the 94H10 circuit. Before the C sequence can be enabled by the A sequence, three conditions must be satisfied: 1) f = 10, 11; 2) j $\neq$ 3; and 3) $Z_{L4} \neq 0$.

Only under these conditions will the 94H10 circuit provide a "1" input into the 60N82 allowing the C sequence to be enabled. If $Z_{L4} = 0$, the A sequence enables would remain applied to the timing chains and would result in the reading of the next instruction.

f = 10; j = 3 - Multiply. - Multiplication performed in the Trainer makes use of the shifting capability of the arithmetic section and the Adder → A step. Initially, the multiplier is placed in the Q register and the multiplicand is placed in D. The double-length product is formed in AQ with A holding the most significant or higher order bits. An examination of ordinary pencil and paper multiplication will provide a basis for understanding the Trainer operation of multiplication. In the pencil and paper method, intermediate or partial products are formed by using the multiplicand and successive digits of the multiplier, beginning with the lower order digits. These partial products, when set down, are shifted relative to one another so that their digits fall in the columns of proper order. They are then added to form the final product. See the following example:

```
Multiplicand        15
Multiplier        x 14
     Partial       60
     Products
                                Shift to place
                  ┌──────       digits in proper
                 15             order columns.
Product         210
```

Working through this example shows that this pencil and paper method relies on previous knowledge of the decimal multiplication and addition tables, proper attention to carries and the order of partial product digits, as well as the concepts of simple multiplication and addition.

In binary multiplication, no carries can occur because the results are always single digits; they are either "0" or "1" (and "1" being only when both multiplier and multiplicand are "1"). See figure 4-58. The previous example is worked out in binary multiplication as follows:

```
                1 1 1 1   (15)
                1 1 1 0   (14)
Partial         0 0 0 0
Products      1 1 1 1
            1 1 1 1
          1 1 1 1
1 1 0 1 0 0 1 0   (210)
```

Notice that in binary multiplication, the partial products are either the multiplicand (when the current multiplier digit is "1") or "0" (when the current multiplier digit is a "0"). Notice also that the addition of several binary numbers at once can be confusing and complex. This complexity would require a corresponding complexity of design if this method were used in the Trainer.

MULTIPLICAND

MULTIPLIER	0	1
0	0	0
1	0	1

Figure 4-58. Binary Multiplication

During Trainer multiplication, each partial product is accumulated in a running total as it is formed, so that only two binary numbers are being added at one time. To ensure that each partial product is added with proper regard for the order of its digits, the running total is shifted to the right each time the partial product is added. When the multiplier bit is "0", the partial product is also "0", thus, this step requires only a shift. In other words, multiplication by a "1" consists of adding the multiplicand and shifting. Multiplication by a "0" consists of merely shifting. This process, using the previous example, is shown in figure 4-59. (It is assumed in the example that both numbers are positive and no sign bit is used.) The Trainer does not contain the hardware to perform sign correction or detection during the multiply instruction as do full scale computers, therefore the multiplier and the multiplicand must be made positive in preparation for the instruction execution by the program.

		Multiplier Digits
Multiplicand	1 1 1 1	
Partial Product	0 0 0 0	0
(No Add)	- - - -	
Right Shift	0 0 0 0 0	
Partial Product	1 1 1 1	1
Sum	1 1 1 1 0	
Right Shift	0 1 1 1 1 0	
Partial Product	1 1 1 1	1
Sum	1 0 1 1 0 1 0	
Right Shift	1 0 1 1 0 1 0	
Partial Product	1 1 1 1	1
Sum	1 1 0 1 0 0 1 0	
Right Shift	1 1 0 1 0 0 1 0	

Figure 4-59. Trainer Multiplication

The D sequence is enabled by the A sequence through the use of the 61N81 inverter. It is during this sequence that the operand (multiplicand) will be read from memory at the address specified by the lower *nine bits of* the instruction word. After the multiplicand is made available from memory to the Z register, it is transmitted to the D register via X. The timing chain is temporarily disabled during this sequence to ensure the contents of Z will not be changed or cleared before the write cycle of the memory reference is completed. When the timing chain is allowed to advance again, the clearing of the Z and A registers occurs in preparation for the C sequence. Table 4-19 shows the commands generated during the execution of the multiply instruction.

TABLE 4-19. f = 10, j = 3 - MULTIPLY INSTRUCTION COMMANDS

SEQUENCE	TIMING FLIP-FLOP	INVERTER	COMMAND	COMMENTS
D	11T11	61N10	Clear S, Initiate Memory	Address
		61N12	Adder → S	Multiplicand
		61N13	Clear Z	
	MEMORY	61L10	Clear X, $Z_L \rightarrow X_L$	
	TIMING	61L11	$Z_U \rightarrow X_U$	Multiplicand to X
	11T23	18T31	Disable 13T31	Wait for the completion of the memory cycle.
	11T33	64N60	Clear D, X → D	
	11T41	64N71	Clear Z, Clear A	See text.
		64N80	Enable C	
C	11T21	63N20	Clear X, $QR_1 \rightarrow X$	
		63N24	Clear S, $Z_{L4}' \rightarrow S$	Shift AQ right
	11T23	63N30	Clear Q, X → Q	one place and
	11T31	63N40	Clear X, $A_{R1} \rightarrow X$	decrement shift count.
		63N43	Clear Z, $(S_{L4}+1)' \rightarrow Z$	
	11T53	63N51	Clear A, Adder → A	If $Q2^0 = 1$
		63N50	Clear A, X → A	If $Q2^0 = 0$
	11T41	63N80	Enable C if $Z_{L4} \neq 0$	Recirculate C sequence.

The registers that are involved during this instruction and their use in the execution are:

Z S S adder	}	Control of shift count and shift count decrementing
D	-	Holds the multiplicand.
Q	-	Holds the multiplier.
A	-	Forms the partial product.
X	-	Performs the shifts and holds the contents of A during the Add step.
AQ	-	Holds the product as completion of instruction.

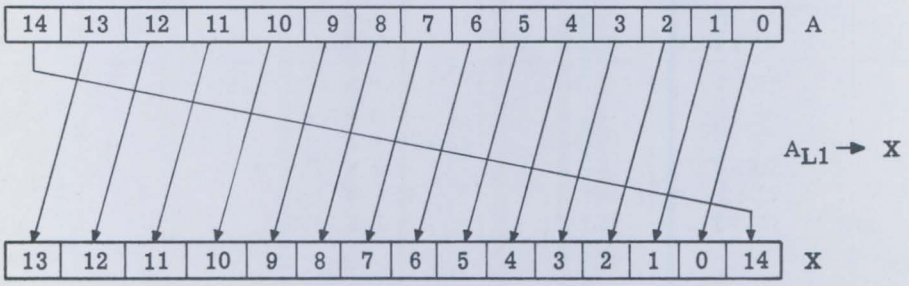
After the C sequence is enabled, the shifting actions start immediately, first Q followed by the A register. Because the multiplier is contained in the Q register, provisions are provided to compute the 2^0 bit so that it can be examined at the end of the C sequence. When the A register is transmitted to X ($A_{R1} \rightarrow X$) one of two paths are considered to return the partial product to A. These paths are controlled by 63N51 (Adder → A) and 63N50 (X → A).

For the multiply instruction, 63N51 will receive a "0" input from 81F04, which is translating for f = 10, j = 3, and $Q2^0 = 1$. If these conditions are satisfied, the sum of A, shifted one place to the right, and the multiplicand will be gated to the A register (~~X + 0~~) (X+D) to form the partial product. If $Q2^0$ was equal to zero, the output from 81F04 will be a "1" and will force 63N51 to a "0" output. The "0" from 63N51 is applied to 63N50 and is combined with "0" enables from 10U00 ("0" if $j \neq 0$) and 55F10 ("0" if f = 10, 11). This path will complete the normal shift by transmitting the contents of X back to A. This action is continued until all stages of Q (multiplier) have been sampled.

The S adder, Z and S registers are utilized in this instruction to control the shift count. A shift of 16 places will be required to properly position the product in the AQ register. Because Z_{L4} is the only bit of the Z register that is considered during shifts, its representation can only be 15 places. The 16th shift is obtained by clearing the Z register (64N71) and counting the Z register down by one. The result of this action, during the C sequence will transmit the complement of Z lower four to the S register setting S lower four to "1's". The addition of one to S by the S adder will result in the lower four bits being returned to "0's". The carry will be extended to the higher order bits, but is ignored for this purpose. From the output of the Adder, the complemented sum is returned to Z lower 4 and will reflect a total of 1111 (decimal 15) shifts to be completed. The translation of $Z_{L4} = 0$ is now sampled for the first time during the multiply. Because Z is not equal to zero, the C sequence is recirculated again and will continue to do so until $Z_{L4} = 0$. The results obtained is a shift of 16 places. Figure 4-60 shows this technique.

f = 11; j = 0 or 1 - Left shift A or Q. - The left shift is a circular shift. That is, the 2_{14} bit is moved into the 2^0 bit position of the same register. The left shift of A or Q is accomplished in the same manner so only one of these registers will be considered in this explanation. The shift count is taken from the lower four bits of the Instruction word and is processed in the same way it was during the right shift instruction.

The shift of a register to the left is controlled by separate gates other than those employed during the right shift. The transmission of $A_{L1} \rightarrow X$ will misplace each bit one position to the left. The paths made available to accomplish the end-around or circular shift is illustrated in the following example:



The C sequence is enabled directly from the A sequence by the 61N82 inverter. Notice that the Z_{L4} translation is sampled and must be unequal to zero before the shift can begin. If Z_{L4} is equal to zero, the A sequence will be enabled again and no shift will be executed. The commands generated by the Left Shift A instruction is shown in table 4-20.

The shifting of the Q register (f = 11; j = 0) is done in the same way as the shift of the A register. The commands generated for the left shift of Q are controlled by 63N21 (Clear X, $Q_{L1} \rightarrow X$) and 63N30 (Clear Q, X → Q).

f = 11; j = 2 - Left Shift A and Q. - The shift of the A and Q registers together is a circular or end-around shift as it was during the shifting of the individual register. As illustrated in figure 4-61, the 2^{14} stage of Q is loaded into the 2^0 position of the A register, and the 2^{14} stage of A is entered into the 2^0 position of Q. In this figure, five bit registers are considered for simplicity.

When the command to shift $Q_{L1} \rightarrow X$, the 2^{14} stage of the A register is provided a path to the 2^0 position of the Q register. The G41 flip-flop is utilized as temporary storage place for the 14th

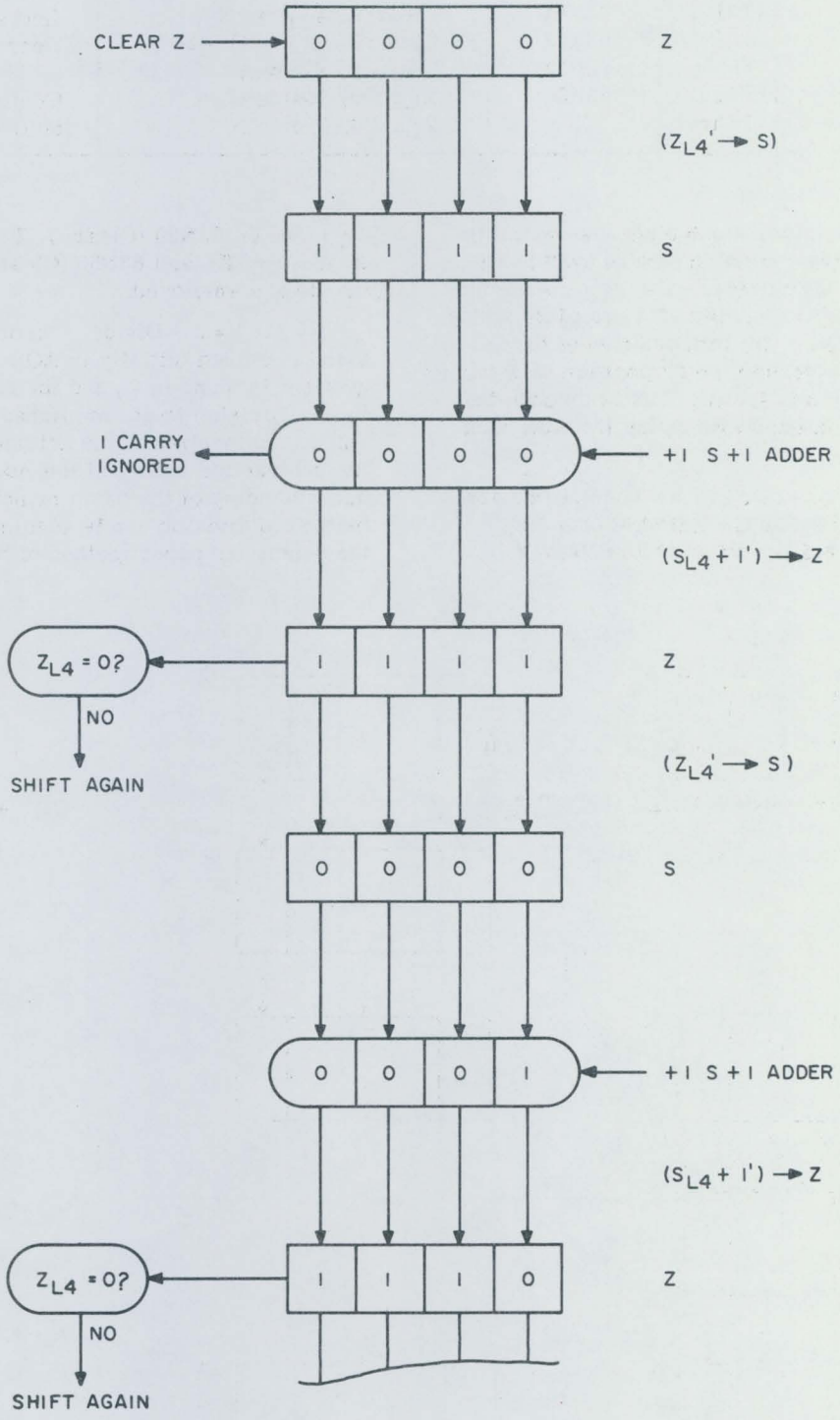


Figure 4-60. Shift Count for Multiply and Divide Instructions

TABLE 4-20. LEFT SHIFT INSTRUCTION COMMANDS

SEQUENCE	TIMING FLIP-FLOP	INVERTER	COMMAND	COMMENTS
C	11T21	63N24	Clear S, $Z_{L4} \rightarrow$ S	
	11T31	63N41	Clear X, $A_{L1} \rightarrow$ X	Left shift one place.
		63N43	Clear Z, $(S_{L4} + 1)' \rightarrow$ Z	Decrement shift count.
	11T53	63N52	Clear A, X $\rightarrow$ A	
	11T41	63N80	$Z_{L4} \neq 0$, Enable C	Continue to shift.
	11T43		$Z_{L4} = 0$, Enable A	Shift completed.

position of the Q register, which otherwise would be lost. The normal transmission path is used to return the data to the Q register. The $A_{L1} \rightarrow$ X command will misplace the contents of A one place to the left in the X register. The 14th position of the Q register is now entered into the 2^0 position of X from G41. The completion of the AQ shift is through the use of the X $\rightarrow$ A command restoring the data back into the A register.

The commands generated by the C sequence are the same as they were for the left shift of a single register (see table 4-20). The 63N21 (Clear X, $Q_{L1} \rightarrow$ X), 63N30 (Clear Q, X $\rightarrow$ Q), 63N41 (Clear X, $A_{L1} \rightarrow$ X), and 63N50 (Clear A, X $\rightarrow$ A) commands are employed.

f = 11; j = 3 - Divide. - In division the 30 bit dividend is stored initially in AQ, the divisor in D, the quotient is found in Q, and the remainder, if any, is in A. Division is accomplished by making use of the shifting properties of the arithmetic section and by the subtraction ability of the Adder. A general understanding of the basic principle of the Trainer method of division can be obtained by first reviewing the pencil and paper method of division. In division

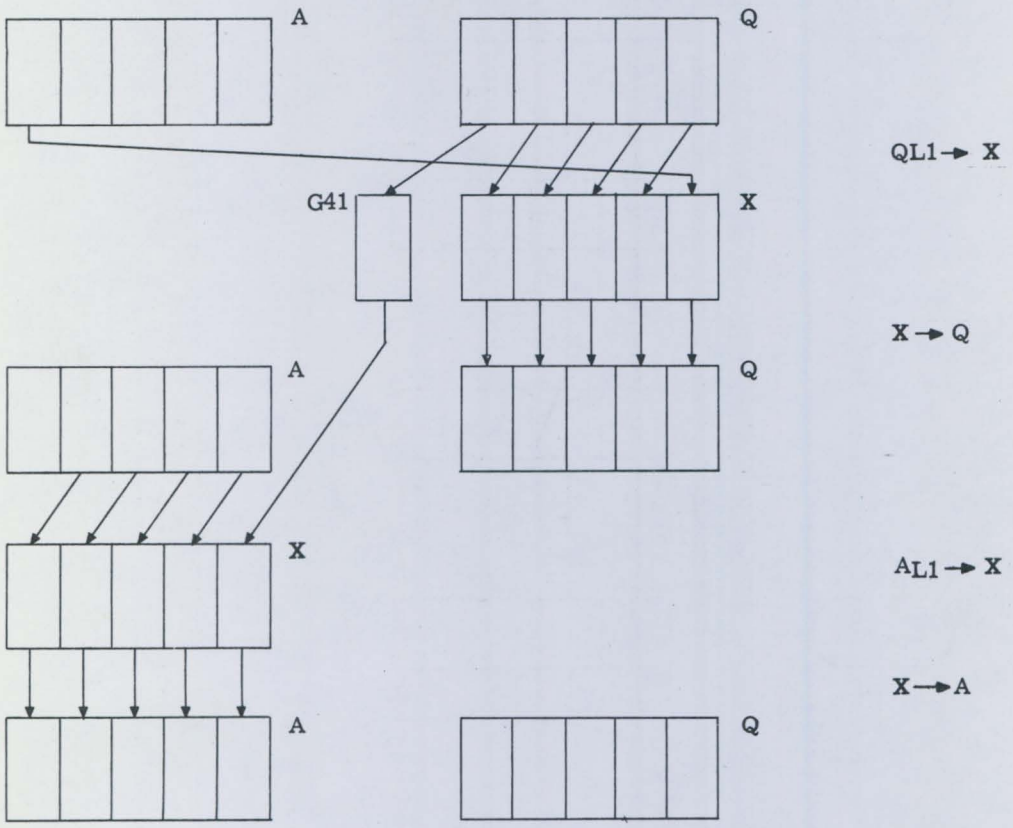


Figure 4-61. Left Shift of the A and Q Registers

the quotient indicates how many times the divisor can be subtracted from the dividend. The quotient is formed one bit at a time by a series of partial divisions as follows:

```
        251
   2 ) 503
       -2
       -2
        10
        -2
        -2
        -2
        -2
        -2
        03
        -2
         1   remainder
```

The highest order digit of the quotient, two, is obtained by counting the number of times the divisor can be subtracted from the first partial dividend, five. The second partial dividend, 10, is formed from the remainder of the previous partial division and the second digit of the initial dividend which is taken as the lowest digit. Since two is contained in 10 five times, the second digit of the quotient is five. This procedure is continued until each digit of the initial dividend has been used in a partial dividend.

Trainer division is similar to pencil and paper division. The two methods differ in certain respects, however:

(1) Numbers are expressed in binary rather than decimal notation. The consequence of this is that with binary numbers the division can be subtracted only once, if at all, for each partial dividend. In division of decimal numbers, the divisor can be subtracted as many as nine times from the partial dividend. Thus, the use of binary numbers simplifies the process considerably from this standpoint.

(2) In division by pencil and paper method, there is no fixed limit to the size of the divisor, dividend, or quotient. If one wishes to spend the time, he can divide a dividend of 100 digits by a divisor of two digits and get a quotient of perhaps 50 digits. In the Trainer, however, there is a fixed limit to the size of the three numbers. The divisor cannot exceed 15 bits (14 data bits and the sign bit) and the dividend cannot exceed 30 bits (29 data bits and the sign bit). Furthermore, the relative sizes of the divisor and the dividend must be such that there are no more than 15 bits in the quotient. For example, in the Trainer, a dividend of 15 or more significant digits cannot be divided by a divisor of one digit, because in this case the quotient would exceed the maximum.

(3) The stair step aspect of the pencil and paper example should be noted: there is a step corresponding to each partial division. This gives the appearance that the divisor was shifted one place to the right before each subtraction. The same effect will be obtained if the dividend were to be shifted one place to the left.

(4) Further, is should be noted that there is a partial dividend for each digit in the quotient. Regardless of the actual size of the number, Trainer division procedure always assumes and operates as though they are the maximum size; namely, 15 bits for the divisor and the quotient, and 30 bits for the dividend. Briefly, this means that the zeros to the left of the most significant bit are not disregarded as they are in the pencil and paper method. The Trainer always performs 15 partial divisions in order to form a 15-bit quotient. A simplified divide sequence is shown on figure 4-62.

In order to understand Trainer division, an understanding of the operation of the Arithmetic adder for subtraction is necessary. The X register, during addition, subtraction, multiplication, and division is loaded with the contents of the A register. This allows the X and D register to be combined in the Adder with the results loaded directly in A. Because the Adder is a subtractive, end-around borrow type, the normal reflected input for an add process would be X plus D' to accomplish an add by subtracting. Therefore, during the subtract process the operand is complemented during the transmission of X → D. The selected input to the Adder from D then would be the normal value, and the difference of X and D would appear as the Adder output. The following example shows in a simplified manner, the reflected inputs to the Arithmetic adder.

Add process - (Add by subtraction requires the second rank input to the Adder to be complemented.)

Normal contents of D	000101 = 5
Normal contents of X	001001 = 9
Reflected contents of D to Adder (D')	111010
	001111
End-around borrow	1
Adder output (X -D')	001110 = 14

Subtraction process -

Normal contents of operand in X	000101
X' → D	111010
A' → X	001001
Reflected contents of D to Adder (D')	000101
Adder output (X - D')	000100

As it can be seen in the Adder, during subtraction D is apparently complemented and subtracted from X. Therefore, if X is greater than or equal

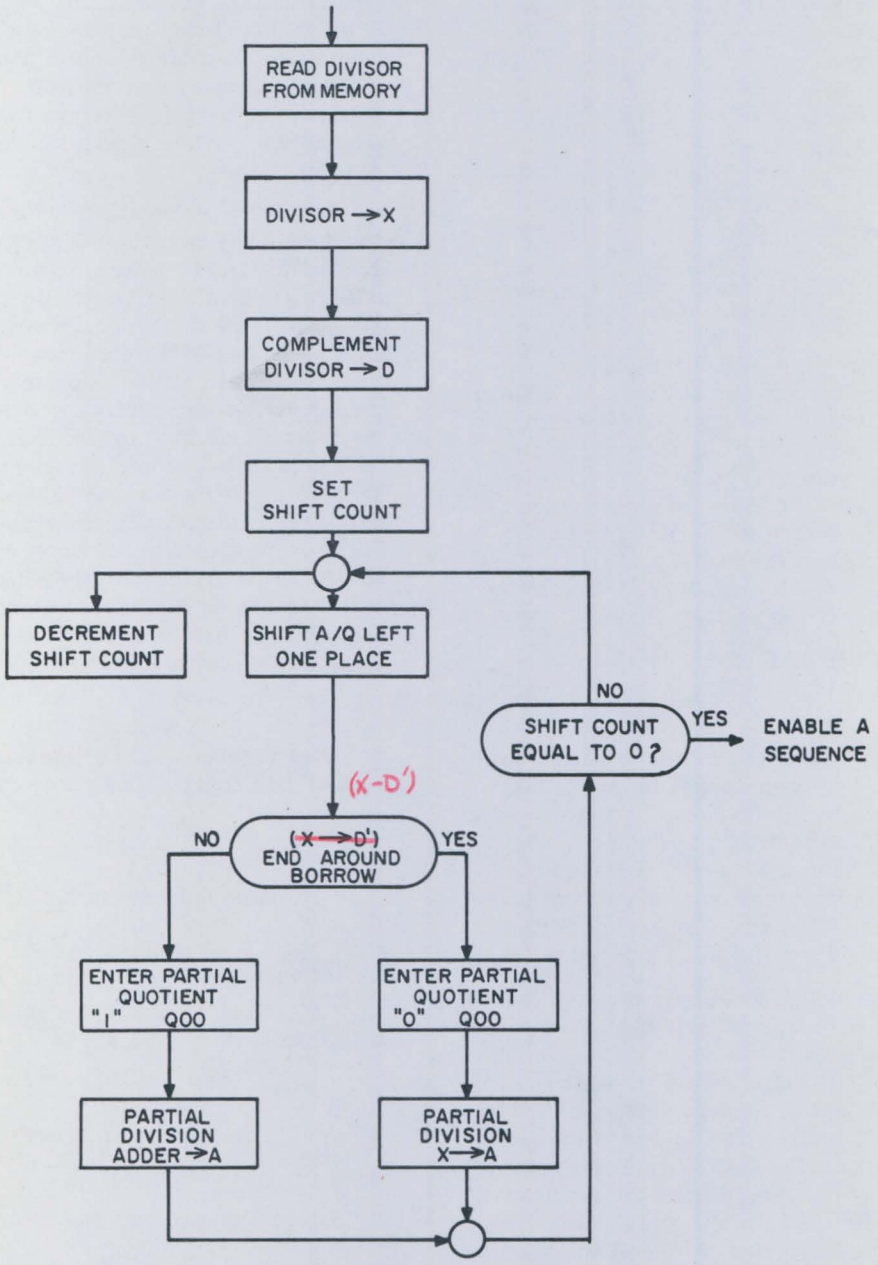


Figure 4-62. Simplified Divide Sequence

to D', a subtraction is done with no end-around borrow. If, however, X is less than D', an end-around borrow is initiated. If an end-around borrow is generated, the 62A00 will be a "1" and will cause the disabling of the 63N51 command inverter (C sequence). This is used in the divide sequence to indicate that the divisor is larger than the dividend, and to disable Adder → A. If D' can be subtracted from X (no end-around borrow), a partial division is done and a "1" is entered in Q00 as a digit of the quotient. If D' cannot be subtracted from X (end-around borrow requested), a "0" is entered into Q00 indicating that a partial division was attempted but not completed because the divisor was larger than the partial dividend. AQ (the dividend) is then shifted left one place. In the divide instruction the Trainer senses if X is greater than or equal to D' and, if true, "1" → Q00 . If X is less than D', 0 → Q00 . AQ has been shifted left one place. This sense and shift procedure is performed 15 times. When the procedures stop, the quotient is in Q and the remainder is in A. The shift count is maintained in much the same way as in a multiply instruction, except only a shift of 15_{10} places is requested during the Divide instruction and is automatically entered into the Z register. As each shift and sense procedure is completed, the shift count is decreased by one. When the shift count equals zero, it causes the divide operation to stop. The circuits and operations of the divide instruction are summarized in table 4-21. Notice that the Trainer gives the programmer no choice but to gain the divisor from memory at the address specified by the lower nine bits of the instruction word.

TABLE 4-21. f = 11; j = 3 - DIVIDE INSTRUCTION COMMANDS

SEQUENCE	TIMING FLIP-FLOP	INVERTER	COMMAND	COMMENTS
D	11T11	61N10	Clear S, Initiate Memory	Address divisor
		61N12	Adder → S	
		61N13	Clear Z	
	MEMORY	61L10	Clear X, Z_L → X_L	Divisor to X
	TIMING	61L11	Z_U → X_U	
	11T23	18T31	Disable 13T31	Wait for completion of the memory cycle.
	11T33	64N61	Clear D, X' → D	Complement divisor → D.
	11T41	64N72	Set Z = 17 (octal)	Set Z to shift count.
		64N80	Enable C	
C	11T21	63N24	Clear S, Z_{L4}' → S	
		63N21	Clear X, Q_{L1} → X	
	11T23	63N30	Clear Q, X → Q	Shift Q left one place.
	11T31	63N43	Clear Z, $(S_{L4} + 1)'$ → Z	
		63N41	Clear X, A_{L1} → X	
	11T53	63N51	Clear A, Adder → A	If no EAB*, enter partial quotient.
		63N59	Set "1" → Q00 if no EAB	
		63N50	Clear A, X → A	If EAB
	11T41	63N80	Enable C	If $Z_{L4} \neq 0$
	11T43		Enable A	If $Z_{L4} \neq 0$

* EAB - end-around-borrow

It is assumed that the program has advanced through the A and D sequences and into the C sequence. In the first step of the example, the shift of Q is immediately initiated and the commands Q_{L1} → X and X → Q are generated. This leaves the 2^0 position of Q (Q00) available to receive the partial quotient. Next comes A_{L1} → X. The values of the dividend (X) and the divisor (D) are now compared through the use of the Adder. Assuming there was an end-around borrow (X is less than D'), "0" → Q00 occurs, indicating that a division was attempted, but that the result of the subtraction was not transmitted to A because the divisor (D') was larger than X. The Trainer has now completed the first partial quotient, and shifted AQ. The shift count is now sampled, and if found to be not equal to zero, will allow the C sequence to be recirculated to continue the divide instruction. This same series of steps will be done until, for the first time, X is greater than or equal to D'. When this condition occurs, a subtraction can be performed, and a "1" is entered in Q00 to indicate this. Since X ≥ D, there is no end-around borrow indicated by 62A00 (Adder) and Adder → A occurs. This action performs the partial division. The sampling of the

dividend and divisor will continue subtracting or not subtracting, depending on the end-around borrow condition, until the shift count is equal to zero. At this time, the remainder (result of the last subtraction and/or shift) is in the A register and the quotient is in the Q register.

In the Trainer process of division, there are two possible fault conditions. The first occurs when the divisor is equal to zero and the second occurs when the quotient is more than 15 bits in length including the sign bit. The Trainer will carry out the division regardless of whether one of these fault conditions exists. However, the divide instruction does provide for the detection of such a fault after the division has been performed. Divide faults are as follows:

The fault resulting from a divisor which is equal to zero can be detected by programming the Arithmetic jump (f = 14; j = 3 Q negative jump) instruction. When the divisor is zero, the quotient is always negative. The only other condition that will cause Q to be negative at this time, is the occurrence of the quotient overflow fault. Therefore, the fault condition of the divisor being equal to zero can be detected by jumping to another part of the program if Q is negative. Let us consider what the quotient is if the divisor is equal to zero. Since the divisor is zero, the condition, X is greater than or equal to D', is satisfied for each of the partial divisions. For each partial division in which the condition is met, the partial quotient entered into Q00 is a "1". Therefore, when the divisor is zero, each bit of Q (including the sign bit) is a "1" at the end of the division phase. Thus, if the divisor is zero, Q is always negative when the jump evaluation is made. The jump, when satisfied, should jump to a subroutine designed to remedy the fault.

The second fault condition, commonly called quotient overflow, occurs when the proper quotient for a division contains more than 14 bits and the sign bit. The division procedure in the Trainer provides for only a 15-bit quotient. When the divisor is excessively small in comparison with the dividend, a quotient with more than 15 bits can occur. Normally, suitable steps are taken in the program to avoid this quotient overflow condition. Sometimes it occurs because the exact size of the divisor and the dividend are not known in advance. For this reason it is desirable to have a method of detecting whether an overflow has resulted during division. An examination of the quotient resulting from a division, when an overflow does not occur, will aid in understanding the method used to detect an overflow condition. The first partial division produces the first partial quotient bit which will be, finally, the sign bit when the 15 partial divisors have been performed. Thus, the first partial quotient is the sign bit of the final quotient. As part of the program, steps should be taken to ensure that the dividend and the divisor are positive, which will give a positive quotient, if there is no divide fault. Therefore, the first partial quotient should be "0". In order for the first partial quotient to be "0", the first partial dividend must be less than the divisor (X is less than D'). If the first partial dividend is larger than, or equal to, the divisor (X is greater than, or equal to, D'), the first partial quotient will be a "1". Therefore Q will be negative at the completion of the Divide instruction. It has been shown that if Q is positive, an overflow condition does not exist, while if Q is negative, an overflow condition does exist. The handling of this fault could be accomplished in the same manner as was the divisor equal to zero fault. A subroutine that is designed to remedy the fault, should it occur, should be utilized.

f = 12 - Input/Output Instructions. - Communications between the Trainer and any external equipment (typewriter, magnetic tape units, paper tape units, etc.) are accomplished by programming function code equal to 12 which is combined with the j designator to provide the necessary control signals and gating action required. There are three input/output instructions generated with f = 12. These are:

1) j = 0. Gate the data from the cable or external equipment into the Q register and send the Input Acknowledge Control signal

2) j = 1. Transmit the contents of the Q register and send the Output Acknowledge signal to the external equipment

3) Send the contents of the Q register to the external equipment accompanied with the External Function control signal. The External Function code that is made available from the Q register must be a control function to the equipment receiving the code; for example, turn on motor, turn off motor, etc.

The Trainer has two communication paths available to it. These are designated as channel zero and channel one and are addressed or selected by the 2^0 bit of the instruction word. If 2^0 is a "0", channel zero is selected, while if 2^0 is equal to "1", the selection of channel one is made. Channel zero is reserved for communication to and from the typewriter and will input and output to the chassis located on the right rear of the Trainer. This chassis contains all of the circuitry associated with the typewriter and will examine only the lower six bits of the Q register. Channel one is utilized for that equipment external to the Trainer. Through the use of an adapter, the full 15 bits of the Q register is transmitted in a parallel state on channel one. Further details will be found in the input/output section of this manual.

The C sequence is enabled directly from the A sequence. There are only four command inverters that will be enabled when the Function code is equal to 12. These are 63N38, 63N32, 63N33, and 63N34 under the control of the T23 timing chain flip-flop.

When the j designator is equal to zero, the 63N38 and 63N32 inverters will be partially enabled with "0" inputs from 91F12 (f = 12), 20U00 (j = 0), and the T23 timing chain flip-flop. The control to these inverters is the fourth input to 63N38 which is the 2^0 flip-flop of the X register now holding the lower nine bits of the Instruction word. If X 2^0 stage is holding

a "1" (channel one) the input to 63N38 will be a "0" and will force the output to a "1". This action will cause the entire Q register to be cleared and the data contained on channel one (15 bits) to be gated into the Q register. If, however, the 2^0 position of X is holding a "0" (channel zero), the input to the 63N38 inverter will force its output to a "0" which is applied to 63N32 making it fully enabled. The "1" output from 63N32 will clear only the lower six bits of the Q register and gate the 6-bit data word from the typewriter circuitry into the Q register. In both cases, the Input Acknowledge is retained to the external equipment indicating the data has been gated into the Trainer. The selection of the channel that will transmit the Input Acknowledge is controlled by other circuitry and will be discussed in the input/output section.

The j = 1 translation will transmit the Output Acknowledge control signal on the channel specified by the 2^0 bit of the X register (lowest order bit of the Instruction word). The 63N33 inverter will be fully enabled during this instruction with "0" inputs from 91F12 (f = 12), 20U01 (j = 1), and T23. Again as during input, the channel selection is controlled by the input/output circuitry. The Q register must be loaded with the Data word that is to be transmitted before this instruction is executed.

The External Function instruction (j = 3) gives the Trainer the capability to remotely control the operation of the connected external equipment. The Function code that is to be transmitted to the external equipment must be loaded into the Q register. These codes will be recognized by the equipment as commands to turn on the motor, turn off the motor, master clear, set input or output modes, etc., when accompanied with the External Function code control signal. The codes required by each equipment vary and the programmer must become acquainted with these codes to properly utilize the instruction.

The command to send the External Function signal is controlled by the 63N34 inverter. This circuit will be fully enabled with "0" inputs from 91F12 (f = 12), 20U03 (j = 3), and the T23 flip-flop. The "1" output will be applied to two gates where the decision is made to determine the channel that will transmit the External Function control signal.

f = 13; j = 0 - Address Substitute Q.- It may be desired to change the lower nine bits of a certain word during a given program execution. The execution of this instruction will destroy the lower nine bits of a data word contained in memory, address specified by the lower nine bits of the Instruction word, and replace the destroyed data with the lower nine bits of the Q register.

TABLE 4-22. f = 13; j = 0 - ADDRESS SUBSTITUTE Q INSTRUCTION COMMANDS

SEQUENCE	TIMING FLIP-FLOP	INVERTER	COMMAND	COMMENTS
D	11T11	61N10	Clear S, Initiate Memory	Address data word
		61N12	Adder → S	
		61N13	Clear Z	
		64N10	Inhibit Memory Lower	(Destroy lower 9 bits) of data word.)
	11T21	63N30	Clear X, Q → X	Contents of Q → X
	11T23	63N40	X → D	
	11T31	64N51	$D_L \rightarrow Z_L$	Q lower 9 bits to Z Z upper not altered
			Enable A sequence.	

Only the A and D sequences are utilized to complete this instruction, as can be seen by the commands. The first four commands as generated by the 61N10, 61N12, and 61N13 inverters will reference and address the Data word that is to be altered. The command to Inhibit Memory Lower will allow only the upper six bits of the data word to be entered into the Z register. This action will also disable the command to transmit Z → X by the output of the L20 flip-flop. The Q register's contents are now moved to the D register via A. Only the lower nine bits of D (containing the Q register) are transmitted to Z. Thus, the Z register will now have the upper six bits of the Data word from memory and the lower nine bits of the Q register. During the write cycle, the new or altered word is returned to the memory location completing the instruction.

The address Substitute A instruction is identical to the instruction just discussed except that the 63N30 inverter (Clear X, Q → X) is disabled and replaced with the 63N35 inverter (Clear X, A → X). The decision or selection of which inverter is to be enabled is controlled by the 02U10 input to 63N36. Notice the 51F13 (f = 13, 17) is available to both inverters. The 02U01 is the upper bit of the j designator and is from the zero side of that flip-flop. Therefore, if the output of 02U01 is a "0", j has to have the value of either "0" or "1". This will allow the 63N36 to be fully enabled. But if the output from 02U01 is a "1" (flip-flop in "1" state) the j designator must be equal to two or three and will force the 63N36 to output a "0". This "0" will be applied to the 63N35 inverter which will now be fully enabled and will allow the A register

to be transmitted to X. All other commands are the same for both instructions.

f = 13, j = 1 or 3 - Complement A/Q. - The complementing of the A or Q registers is performed with the execution of this instruction, with the selection of the register that will be involved, made by the upper bit of the j designator. This selective action is accomplished in the same manner as explained in the previous paragraph.

TABLE 4-23. COMPLEMENT A/Q INSTRUCTION COMMANDS

SEQUENCE	TIMING FLIP-FLOP	INVERTER	COMMAND	COMMENTS
C	11T21	63N36	Clear X, Q → X	if j = 1
		63N35	Clear X, A → X	if j = 3
	11T23	63N31	X' → D	Complement A or Q.
	11T31	63N42	D → X	
	11T33	63N53	Clear Q, X → Q	if j = 1
		63N57	Clear A	
		63N55	X → A	if j = 3

As shown in table 4-23, the C sequence is enabled by the 92H13 (f = 13; j = 1 or 3) with a "1" input to 60N82. This circuit will output a "0" to the 61N82 inverter and, combined with the A sequence enable, will force the output to a "1" which will enable the C sequence. The "0" output from 60N82 is also applied to the 61N83 inverter which will invert the signal to a "1". The "1" output of 61N83 will disable the 61N81 inverter, the path that was used to enable the D sequence during the address substitute instructions.

Under the control of the C sequence, the A or Q register, as specified by j, is transmitted to the X register. The complementing action occurs during the data movement between the X and D registers. The complemented data is again returned to the X register and finally to the A or Q register. The A sequence is then enabled to read the next instruction.

f = 15; j = 2 and 3 - STOP, SELECT STOP. - Through normal programming procedures, there are two ways the Trainer can be stopped. This is accomplished by clearing the High-Speed Run flip-flop previously covered in an earlier paragraph (4-2, c, (2)). The circuitry involved in clearing the High-Speed Run flip-flop is controlled by the last flip-flop of the timing chain, T41[3], which provides outputs to 13J46 and 13J47.

Assuming the Trainer to be in High-Speed Run and the instruction being executed is f = 15; j = 2. At the end of the A sequence, the outputs from T41[3] will be a "0" and will be applied as inputs to 13J46 and 13J47 (see figure 5-3). The 13J47 inverter will now be fully enabled with the combined "0" inputs from 11T41[3], 91F15 (f = 15), and 20U02 (j = 2) and will force its output to a "1". The "1" signal is applied as inputs to 00G70, clearing the High-Speed Run flip-flop and 01J47 setting the Stop flip-flop. A "1" is applied from 00J47 to the 77Y46 Indicator Driver which will light the STOP indicator.

Associated with both the Stop and Select Stop instructions is an unconditional jump. That is, when the Stop instruction is executed, a jump condition is satisfied before the stop occurs. In the execution of the Select Stop, a stop will only occur if the SELECT STOP switch is engaged, but the jump condition will be accomplished unconditionally.

The Op Step flip-flop is still in the "1" state providing the necessary enable to the timing chain to allow it to recycle. Because there was no path provided for this instruction to enable another sequence, the A sequence enables are still present to the commands inverters. As the timing chain is advanced under the influence of the A sequence, the 61N30 command inverter will clear the Op Step flip-flop. When the timing chain attempts to recycle to complete the execution of the instruction, the 13T11 gate will be disabled by the absence of the "0" input from 01G73. The Trainer, is therefore brought to a stop with the next instruction to be executed in the U register and its lower nine bits in the X register. To start the Trainer again, it would require the resetting of the High-Speed or Op Set flip-flops by the activation of the console control switches.

The Select Stop instruction (f = 15; j = 3) is performed in the same manner except that the 13J46 inverter is sampled. The clearing of the High-Speed Run flip-flop will occur and the SELECT STOP indicator will light. This instruction is written into a program and is normally used when a stop is desired after the execution of a given number of instructions have been completed. Note that the stop condition can be satisfied only if the SELECT STOP switch is engaged. If the instruction is executed without the switch engaged, the jump condition will be satisfied, but stop will not take place and the Trainer will continue in the High-Speed mode.

f = 17; k = 0, 1 - STORE A/Q. - The storing of the A or Q registers in a memory location is accomplished by this instruction. Note that the B register modification is available for this instruction as it was for function codes 00-05. The k designator is interpreted

as the Function code modifier for this instruction. Table 2-24 shows the commands generated by the D sequence for the storing of one of these registers.

After the Function code is entered into the U register during the A sequence, the b designator is immediately examined. If the b designator is equal

TABLE 4-24. f = 17, k = 0, 1-STORE A/Q INSTRUCTION COMMANDS

SEQUENCE	TIMING FLIP-FLOP	INVERTER	COMMAND	COMMENTS
D	11T11	61N10	Clear S, Initiate Memory	
		61N12	Adder → S	Address and destroy
		64N10	Inhibit Memory Lower	contents of memory
		64N11	Inhibit Memory Upper	where A or Q register
		61N13	Clear Z	is to be stored.
	11T21	63N36	Clear X, Q → X	if k = 0
		63N35	Clear X, A → X	if k = 1
	11T23	64N40	X → D	
	11T31	64N50	D → Z	
			Enable A	

to zero, it is combined with the translation of f = 17 forcing a "1" output from 90H17 which is applied as an input to the 60N81 inverter. The "1" signal is inverted by 60N81 and will enable 61N81 which will enable the D sequence. However, if b is equal to one, the input from 90H17 will be "0" and combined with the remaining inputs, will cause 60N81 to output a "1". This "1" output will disable 61N81 but is also applied as the only input to 63N82 which will invert the signal and apply its output to the 61N80 inverter. The output of 61N80 will enable the B sequence. The reading of the B register will occur during the B sequence as shown in table 4-8. From the B sequence, the D sequence is enabled by utilizing the 62N81 inverter. The Adder → S command will add the B register to the lower nine bits of the Instruction word (Y + B → S) to make up the address where the register (A or Q) is to be stored. If b was equal to zero, the B sequence will not be enabled and the command Adder → S will load Y + 0 → S. Only the lower nine bits of the Instruction word would be reflected to the S register to make up the address.

During the read cycle of the memory timing chain, the contents of memory at the address location where the register is to be loaded is destroyed by the Inhibit Memory command. The register to be stored is then moved to the Z register via X and D. The contents of Z is then written into the memory location during the write cycle of memory timing.

(f) JUMP INSTRUCTIONS. - (See figure 5-14.) Jump evaluation is made for instructions 14, 15, and 16, to determine if the conditions specified by the j designator are satisfied. There are eight circuits which make the jump evaluation for these instructions.

f = 14, j = 0. The evaluation of f = 14 is accomplished by 31G50. The input from 20U00 will be a "0" when j is equal to zero and will be combined with the "0" input from 21A70, translating for the A register not equal to zero. Thus, a "1" output from 31G50 means the jump condition is satisfied. The "1" is inverted to a "0" by 15G50 and will apply the output to the 13G50 inverter. The two other inputs to 13G50 are 20U10 (X4) and 20U23 (14-17) thus, f = 14. When the timing chain is allowed to set the T43 flip-flop, the zero side (10T43) will supply a "1" input to 29G50. The "1" is inverted to a "0" and applied to ~~four~~ five inverters looking for the various conditions to set the Jump flip-flop (G50). This action forces a "1" output from 13G50 which initiates the jump.

f = 14; j = 1. The evaluation for j = 1 is accomplished by the combined inputs to 17G50. The inputs from 20A71 and 20A81 will be "0" when the A register is equal to zero. When the 17G50 circuit outputs a "1", the jump evaluation is satisfied and will result in setting the G50 flip-flop if the Function code is 14.

f = 14; j = 2. The evaluation for j = 2 is accomplished by the 19G50 circuit. Combined with 20U02 (j = 2) is the input from 01A14. This input will be a zero if the 2^{14} bit of the A register is set indicating the contents of the A register is negative. The "1" output of 19G50 will initiate the jump when combined with f = 14.

f = 14; j = 3. The evaluation for j = 3 is accomplished by the 21G50 circuit. The input from 20U03 will be a "0" if j is equal to three. The 01Q14 is the sign position of the Q register and will be a "0" when the Q register is negative. When the output of 21G50 is a "1" and the Function code is 14, the jump is satisfied and the setting of the G50 flip-flop will occur.

f = 15. The evaluation of j = 0, 2, and 3, is translated as an unconditional jump; that is, the j designator is not conditioned on the value or contents of any register or operation of the Trainer for the jump to occur. Execution of this instruction will result in the jump condition being satisfied.

The input to the 25G50 will be a "0" from 20U01 when the j designator is equal to one. That means

that when j ≠ 1, the input from 20U01 will be a "1", which will be inverted by the 25G50 circuit. The "0" output (j = 0, 2 and 3) is applied to the 23G50 circuit where it is combined with the inputs from 91F15 (f = 15) and 29G50. Thus, if the j value is anything other than one, the jump condition is satisfied.

When the value of j is equal to one, the 27G50 and 25G50 circuits are partially enabled by the input from 20U01. The second input to 27G50 will be a "0" if the SELECT JUMP switch is engaged (see figure 5-3). With the 27G50 circuit fully enabled, the output to the 25G50 circuit will be a "1". This signal is inverted to a "0" by the 25G50 circuit and applied as a "0" to the 23G50 circuit. The jump condition, with f = 15, will now be initiated by the setting of the G50 flip-flop.

f = 16. The group of jumps associated with function code equal to 16 are considered as input/output jumps in that they are conditioned on the state of control signals produced and received by circuits of the input/output section. To fully understand the jump evaluation conditions required for this instruction, it is necessary to have knowledge of the control signals and the communication properties of the input/output section. Therefore, the jump evaluation for f = 16 will be discussed as part of the input/output section. When conditions are met and the G50 flip-flop is set, the jump will take place in the same manner as it will for the jumps already discussed.

When a jump condition is satisfied and the G50 flip-flop is set, a "1" output is applied to the 61N11 gate of the timing chain. It is through this signal that the timing chain is notified that a jump condition has been satisfied. In order to maintain an over-all picture of the events for a jump, assume the A sequence is enabled and will read the Jump instruction from memory. As usual the upper six bits of the instruction will be transmitted to the U register where it will be translated immediately while the lower nine bits (address to jump to) is captured by the X register. Notice also, that the D register was cleared by the 61N20 circuit. When the timing chain has been advanced to the setting of the T43 flip-flop, a "1" output is made available to the 29G50 circuit which will sample any possible jump condition. For this example, it will be assumed that a jump has been satisfied and the G50 flip-flop is now set. Because the jump instructions are not included to the circuits that will enable other sequences, the timing chain is recirculated still under the control of the A sequence enable. The setting of the T11 flip-flop will supply the normal enable, but this time 61N11 (P → S) is disabled because of the "1" input from the zero side of the Jump flip-flop, 00G50. The 61N11 is unconditionally forced to a "0" output and will enable 61N12 resulting in an Adder → S command (Y + 0 → S). The instruction at the address specified by the lower nine bits of the Jump Instruction word is now read to the U register. When the T21 flip-flop is set, the command to Clear P and transmit S + 1 → P will occur. The normal A sequence will continue to read the instruction.

(g) SKIP EVALUATION. - A skip evaluation is made for instruction 06 and j equal to 1 or 3 (B register index). If the skip condition is satisfied, the Abort flip-flop will be set as a result of the "1" output from 62N50 during the B sequence instruction commands. When the Abort flip-flop is set, the Trainer will skip over one instruction; that is, not execute the next sequential instruction of the program. The instruction (06) is executed with the Abort flip-flop in the normal manner now. When the A sequence is enabled and the timing chain is advanced, the normal commands are issued including the advancing of the P register until the T23 flip-flop is set. The normal advancing of the A sequence will be stopped by the "1" input to 13T31 from 12T31. The inputs to 12T31 will be "0's" from 05T01 (A sequence) and 01G51 (Abort flip-flop set). Also from 11T23, a "0" is applied to the 19T00 which is combined with "0" inputs from 03T01, A sequence, 01G51, Abort flip-flop and phase one. The "1" output from 19T00 will reset 11T00 and allow the timing chain to be reinitiated under control of the A sequence. It also must be remembered that memory was initiated and will take approximately eight microseconds to complete its cycle time. Therefore, T00 cannot initiate the timing chain again until the memory cycle is complete and a "0" input is available from 01L17 and 13T11. The clearing of the T23 flip-flop is accomplished by fully enabling the 12T23 inverter with "0" inputs from 11T00 and phase four. The second run of the timing chain will read and execute the next instruction, resulting in the skipping of one instruction.

(4) A REPEAT. - The Repeat mode associated with the Trainer can be activated only by manually engaging the REPEAT switch (S2) on the console control panel. This is a convenient method of loading a program into memory at sequential addresses without having to reload the U register or indicate the next sequential address each time. Although this is one manner in which the Repeat mode could be employed, the explanation is not intended to place a restriction on its use. The utilization of the type of operation is left up to the imagination of the operator.

When S2 is closed (figure 5-2), ground is applied to the 72Y44 input amplifier forcing its output to a "1" which serves as inputs to 61N12 and 61L12. The "1" signal is also applied as an input to the 12J44 circuit and is inverted back to a "0" and is applied as a partial enable to the 64N15 circuit. The ground level will also complete the path necessary to light the REPEAT indicator on the console control panel.

Trainer operation, with the exception of the affected circuits, 61N12, 61L12, and 64N15 will be normal. Before an application is discussed, the affected circuits will be reviewed. The normal use of the 61N12 circuit would Clear S and transmit Adder → S during the B or D sequences. The 61L12 circuit is controlled by the memory timing chain and generates the commands to Clear U and Z_U and U. The 64N15 circuit is disabled during normal operation. During

the Repeat mode, the "0" input from 12J44 is combined with 07T14, the D sequence, to disable the clearing of the S register.

Assume that is is desired to store data that would normally be inserted into the Q register at sequential address locations. The initial set up would be:

(1) Manually insert the store Q instruction into the U register (f = 17, b = 0)

(2) Manually insert the first address where Q is to be stored in the P register

(3) Enable the sequence or timing chain

(4) Engage the REPEAT switch and,

(5) Depress the OP STEP switch to the down position.

The A sequence will be initiated and Clear S and transmit P → S and S + 1 → P in the normal manner. The instruction, or data at the address specified by P, will be read from memory but will not be allowed in the U register since the control circuit 61L12 is now disabled. Thus, the manually inserted instruction is retained. Notice, that as the timing chain is advanced, controlled by the A sequence, the 61N20 inverter will clear the Op Step flip-flop causing the Trainer to stop after the completion of the A sequence. At this time, the D sequence indicator should be lit and the timing chain enabled. The OP STEP switch must be depressed a second time to initiate the timing chain. This will allow the timing chain to advance under the influence of the D sequence. During the D sequence, which is employed by the Store Q instruction (see table 4-24), the normal command to Clear S and transmit Adder → S (61N12) is disabled and will not occur. Also, during the D sequence, the 61N10 will be enabled, generating the commands to Clear S and Initiate Memory. Although memory will be initiated, the Clear S command is blocked by the "1" output from 64N15, which is completely enabled by "0" inputs from 12J44 (Repeat) and 07T14 (D sequence). This disabling action will keep the S register at the same values it was when the P register loaded it and will store the Q register at that location.

The Trainer will stop at the end of the next A sequence, after the P register has been advanced again. In this example, with the Store Q instruction still in the U register, the D sequence light should be lit (b = 0). The Q register can now be manually cleared and new information inserted; that is, information to be stored at the next sequential address. Depressing the OP STEP switch will now store the Q register at the second sequential address.

To summarize, the Repeat mode will continue to execute the same instruction contained in the U register. Each time the OP STEP switch is engaged, the normal sequence of events will occur for a given instruction, except the loading of the U register with Z upper six bits during the A sequence; the clearing of the S register and Adder → S commands during the D sequence. If a Store instruction is entered into the U register, the register that is to be stored will be entered into each sequential address (S + 1 → P) location for each time the OP STEP switch is depressed to the down position.

4-5. OPERATION OF THE ARITHMETIC SECTION.

a. GENERAL. - The arithmetic section contains the A, D, Q, and X registers. In addition to these registers, there is a modification circuit called the Adder. The arithmetic section uses special procedures derived from the fundamental processes of one's complement binary arithmetic. The operations of addition, subtractions, multiplication, division, and shifting are made in the arithmetic section (see figure 4-63).

(1) A REGISTER. - The A register (15 bits) is the principle Arithmetic register. After addition or subtraction, the sum or difference is found in the A register. After multiplication, the most significant half of the product remains in the A register. For a divide operation, the remainder is located in the A register. The registers of the arithmetic section do not have the capability of shifting within themselves. A shift can be accomplished within the arithmetic section as follows: To shift the contents of A, transmit A_{R1} → X, Clear A, and transmit X → A. The term A_{R1} → X means transmit A_{29} → X_{28}, A_{28} → X_{27} . . . . . A_{01} → X_{00}. Since the contents of X are the initial contents of A, shifted right one place, the transmission of X → A will position the initial contents of A, shift right one place, back into A completing a right shift of one place. A left shift is accomplished by misplacing the data bits in the X register one place to the left during the transmission of A to X (A_{L1} → X). The command to transmit X → A will complete the shift by placing the initial contents of A back into A shifted left one place.

(a) A REGISTER LOGIC. - The A register is shown on figures 5-15 through 5-17. The register is made up of three sections with five stages in each section, or a total of 15 stages. Each stage contains two inverters connected through printed circuit wiring on a card (250420 circuit) to make a flip-flop and two circuits 13A-- and 15A-- which function as input gates. There are also command enable circuits 12A-- and 14A-- which receive the commands from the main timing chain and enable the proper circuits to fulfill the command. Usually each command enable serves one section (five stages). For example, 14A00 serves stages 00-04, 14A05 serves 05-09, and 14A10 serves stages 10-14.

Note

Since data is gated into the register by command, all transmissions are discussed in terms of the receiving register.

(b) A REGISTER COMMUNICATIONS. - Transmission of data occurs when certain conditions exist. These conditions exist when a bit from the transmitting register is present, the phase three input is "0", and the Command Enable input is "0". For example, bits from the X register are always present

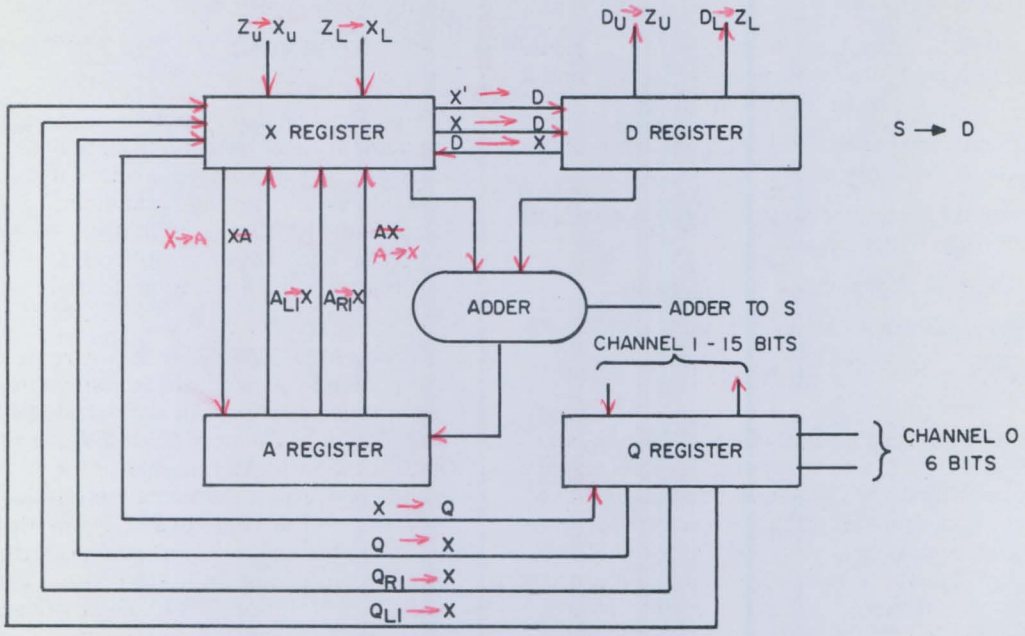


Figure 4-63. Arithmetic Section, Block Diagram

as inputs to the input gates of the A register. The input gates are phase enabled at phase three time; and if the control section has interpreted an instruction to transmit X → A, the command line X → A must also be a "0" at which time the data in X is entered in the A register.

(c) CLEAR A OPERATION. - The Clear A operation can be done manually or by command from the control section. In either case the output 11A-- is a "1" during phase two which forces the zero side of the flip-flops to have an output of "0". This is the clear state. All commands to Clear A are combined as inputs to the 10A00 circuit. When anyone of these inputs go to a "1", the output of 10A00, "0", will apply this "0" output to the three 11A-- term cards. The following phase two will provide the "0" enable to allow the A register to be cleared.

Manual clear of the A register is controlled by a CLEAR pushbutton located to the right of data bit indicators on the control panel. When this pushbutton is depressed, ground is applied to the input of the 99A30 circuit. The 99A30 is not an inverter, but will output a ground signal when ground is applied as an input (see figure 4-4.). The "0" output of 99A30 effectively grounds the output of 10A00 and provides "0" enables to the input of 11A-- circuits. When phase two is generated, the A register is cleared. Note that the clearing of the register occurs one phase prior (Ø2) to the loading phase (Ø3).

(d) SET OPERATION. - Manual set is accomplished by depressing the pushbutton associated with each stage of the register. Pushing the button will cause the feedback to the zero side to ground which then will output a "1". When the zero side has a "1" output, the stage is set (see figure 4-3).

(e) A EQUAL ZERO TRANSLATOR. - The A equal to zero translation is utilized during the Jump Evaluation instruction (f = 14, j = 1 and A not equal to zero for the f = 14, j = 0). This translator, located on figure 5-17, consists of four circuits (20A-- and 21A70). The 20A70 and 20A80 circuits are supplied with output from the zero side of each flip-flop in the A register. If A is equal to zero, all inputs to these two circuits will be "0" forcing the outputs of 20A70 and 20A80 to a "1". The 20A81 and 20A71 will invert the signal back to a "0" and apply this enable to the 17G50 circuit or Jump evaluation, figure 5-14. If A is not equal to zero, either or both 20A70 and/or 20A80 will output a "0". This "0" output will be inverted again by the 20A81 and/or 20A71 and will disable the 17G50 circuit. The "1" is also applied to the 21A70 circuit which will invert the "1" to a "0" and apply the "0" as an enable to the 31G50 circuit. This will satisfy the conditions for a jump evaluation of f = 14; j = 0 and A $\neq$ 0.

(2) D REGISTER. - The D register (15 bits) participates in all the Arithmetic operations, addition, subtraction, multiplication, and division. The D

register contains the addend, subtrahend, multiplicand, or divisor, respectively. The D register can be cleared or set in the same manner as the A register. Clear D can be either a manual operation or part of an instruction. Manual set is accomplished by depressing the button associated with each stage of the register. All data leaving the arithmetic section to other sections must pass through the D register.

(a) D REGISTER LOGIC. - The D register is shown on figures 5-18 through 5-20. The D register is made up of three sections with five stages per section. Each section has command enable circuits for the various commands, for example: 12D00 serves the first section for command X→D and 12D05 serves the second section. Each stage of the register contains the basic flip-flop with the circuits used as input gates. These circuits allow the transfer of data when certain conditions exist. These conditions are: coincidence between the data bit, clock phase three, and the command enable (all signals concerned being "0's".)

(b) D REGISTER COMMUNICATIONS. - The three sources of data available to the D register are from the X register controlled by the 13D-- gates, the S adder, 17D-- gates, and from the zero side of the X register, 15D--. The command to transmit X' →D gives the Trainer the ability to complement a data word which is accomplished by gating the zero side of the X register to the one side of the D register. This is the only place in the Trainer where complementing is accomplished.

Commands generated by the timing chain to control the data flow into the D register are applied as inputs to the command enable circuits 12D--, 14D-- and 16D--. When any of the inputs are a "1", the output of the command enable circuits is forced to a "0" and is applied as inputs to fives gates in one D register section. The clearing of the D register is controlled by the 10D00 circuit. When any of the ten inputs go to a "1", the output of 10D00 will be a "0". This "0" is an enable to the three 11D-- circuits which, at phase two time, will output a "1" and Clear D. Notice that the D register is cleared on phase two and set on the following phase three.

Communication outputs are taken from both sides of the flip-flop. From the zero side, outputs are available to the X register (25X--) and the Arithmetic adder (30Z--). The 99D-- terms are the Indicator Drivers that control the indicators on the control panel. Outputs are available from the one side to the Arithmetic adder (32A--) and the Z register (13Z--).

(3) Q REGISTER. - The Q register (15 bits) participates in the divide and multiply operations. It is also the communication register for the input/output section. In the multiplication process, the multiplier is in Q and the double-length product is found in AQ. In a division operation, the dividend is in AQ and the quotient will be found in Q.

When used as the input/output communication register, the Q register will hold the data to be sent to external equipment (data or external function) and will receive the data from the external equipment when the Input instruction is executed.

(a) Q REGISTER LOGIC. - The Q register (figures 5-21 through 5-23) is composed of three sections. These sections are broken down into one group of six stages; section one, one group of four stages; section two: and five stages for section three. The abnormal breakdown of Q is to facilitate the input/output communication requirements. Each stage has a flip-flop and circuits for gating the data into the stage. Also, each section has its own command enable circuits. For example, 12Q00 provides the command enable of the first section for the command X→ Q; 12Q05 serves the second section; and 12Q10 serves the third section. Data is gated into the Q register by the occurrence of three coincident signals: a data bit from some source, phase three a "0", and the proper command enable, a "0". From within the Trainer, the Q register can only receive data words from the X register.

When the Divide instruction is being executed, the partial quotient is entered into the Q register. This is accomplished on a bit by bit basis from the output of the 92H11 circuit feeding directly into the one side of 01Q00. If a subtraction without an end-around borrow is possible, the partial quotient is a "1" and Q00 is set by 92H11. During the shift of one place to the left, Q00 is cleared (no end-around during divide). If the next partial division results in no subtraction (X is less than D), the Q00 flip-flop remains cleared and is shifted again to the left. This would result in the first partial quotient now being shifted into Q02, the second in Q01, and the third to be inserted in Q00 after the sampling of the Adder is completed. This continues until the register has been shifted 15 times which will leave the quotient properly positioned in Q and the remainder in A.

During the Multiply instruction, the multiplier is in Q. Each bit is examined to determine if an add will or will not occur. The output to 13G41 from 01Q00 captures each bit of the multiplier as the Q register is shifted to the right. At the completion of the Multiply instruction, the product is in the AQ registers and the multiplier is destroyed by a right shift.

When the Q register is utilized as the communication register, the register is broken into two word groups. That is, for communications on channel zero, the lower six bits of the Q register are involved. Channel zero is connected internally to chassis four of the Trainer which is used to communicate with an electric typewriter. The outputs made available to the typewriter are taken from the one side of the flip-flops 01Q00 - 01Q05. These outputs are made available to the 23C-- terms which are the control gates for flip-flops on chassis four. During the input mode, the Q register, lower six stages, receive data from the 11C-- term circuits. The gating action is controlled by executing an input instruction (f = 12; j = 0) with the lowest order bit of the Instruction word equal to zero. During the execution of this instruction,

the lower six bits will be cleared by the input from 63N32 to 10Q00 before the gating action occurs. The remaining stages of Q, Q06 - Q14, will not be altered by this instruction.

The output made available from the 02Q-- circuits to the 60Y-- term circuits are utilized for output communications on channel one. When communications utilizing channel one are employed, the full 15 bits of the Q register are transmitted to the external equipment via an adapter which is made available with the Trainer. Inputs from the adapter, channel one, are from the 50Y-- term circuits which gives the Q register the property of receiving as well as transmitting 15 bits in parallel.

(4) COMPARING A, D, AND Q REGISTERS. - In comparing the A, D, and Q registers, it can be seen that they are very much alike. Any of the three registers may be cleared by an instruction or by manual means (CLEAR button). They are all 15-bits in length and divided into three sections. Each section has its own command enable circuits, and each stage has its own input gate circuits. Each has the inputs gated into the stages during the clock phase three and each can be cleared during clock phase two.

The registers are the same except for data transmissions and the section breakdown for the Q register. All of the input signals are not from the same source; all of the output signals do not go to the same place; and the number of bits involved in a transmission is not always the same. The A, D, and Q registers all communicate with the X register.

(5) X REGISTER. - The X register (15 bits) is the communication or exchange register of the arithmetic section. Almost all data exchanged within the arithmetic section is through the X register. The X register is also used in the arithmetic processes with the section. As stated before, it is the transmission of data to and from the X register that performs the shifts necessary in some arithmetic operations.

(a) X REGISTER LOGIC. - The X register (figures 5-24 through 5-31) is similar to the A, D, and Q registers. It can be cleared manually or by part of an instruction, and has the provisions for being set manually. The X register is divided into three sections of five stages per section. As with A, D, and Q, each section has its own command enable circuits and each stage has several different input data circuits. The clock phases, however, are different. The X register is cleared during clock phase four, and data is gated into X during clock phase one. The X register is the opposite of A, D, and Q in operation; register flip-flops are set, the zero side of the flip-flop having outputs of "1's". In the X register, clear pulses are applied to the one side of the flip-flops causing them to have an output of "0". This causes the zero side to have an output of "1". Thus, the X register is cleared to "1's". This facilitates certain operations such as the formation of logical products. To set the X register, set pulses are applied to the zero side causing it to have an output of "0". This forces the one side to have a "1" output, and the X register is set to "0's".

(b) X REGISTER COMMUNICATIONS. - The transmission of data into and out of X is different because of the different method required to set and clear the flip-flops. In the transmissions X → A or X → Q, the input is to the ~~one~~ ZERO side of the stages and, conversely, the outputs from X are from the one side. In the transmission D → X or Q → X, however, the inputs to X are to the zero side and in order to set the register properly, the outputs from D, Q or A must come from the zero side. If the outputs were taken from the one side, it would result in a complemented number in X. In transmissions from X, the output is taken from the one side of the stages. If the output is taken from the zero side of the stages, it results in a complemented number in the receiving register. This is done in certain cases. For example, to complement D: D → X, Clear D, X' → D (X' → D means to take the output from the zero side and transmit it to D); $D_f = D'_j$. In this manner numbers can be complemented as required during arithmetic operations.

(c) ARITHMETIC PROCESSES INVOLVING X. - During the execution of the shifts and multiply instructions, the X register utilizes special circuits to accomplish end-around shifts, or the shifting of AQ together. Each of these special paths will be examined separately.

During the right shift of the Q register (f = 10, j = 0) the shift is open ended with the sign extended. Assuming the command to transmit Q right one place to X was issued, the 20X-- command enable circuits will enable the 21X-- gates. Note that for each stage of X, the input from Q is misplaced one stage to the right. For example, the 12X12 gates feeding the 12th stage of X has an input from 00Q13. The 2^{14} stage of X will have two gates partially enabled by the input from 10U00 ("1" if j = 0), and the 21X14, which will sample the 2^{14} stage of the Q register. This means that the 14th stage of Q is loaded into both X13 and X14, thus extending the sign position.

The shift of A to the right is accomplished in the same manner except the 14X-- which will enable the 15X13 gates under each stage of X. The 14th position of A is entered into both the 14th and 13th stage of X which will extend the sign of A.

When A and Q are shifted together , the sign position of A only will occur and the 2^0 stage of Q will be open ended. The first register to be shifted is the Q register, which will again enable the 21X-- gates by the $Q_{R1} \rightarrow X$ command. At this time, the 2_{14} stage of X will now be loaded with the contents of A00 by the utilization of the 29X14 gate now enabled by 10U00. The 21X14 gate is disabled by the "1" input from 20U00 (j = 0). Therefore, when the Q register receives the contents of X, the 2^0 position of A is included and entered into the 2^{14} position of Q. The normal shifting of A will occur with the sign extension being accomplished as stated previously.

The left shift of either A or Q is an end-around shift; that is, 2^{14} position is entered into the 2^0 position of that register. The command, $Q_{L1} \rightarrow X$ will enable the 23X-- gates associated with each stage of X. The 23X00 gate (2^0 stage) is enabled by the 14th position of the Q register which is combined with the 20U00 circuit (j = 0). This will accomplish the end-around operation. The shifting of A left will enable the 17X-- gates and, at stage 2^0, the translation of j = 1 is combined with the output of 00A14 (2^{14} stage) to gate the data to X. The shift of AQ, however, requires the need for two special flip-flops, G40 and G41. During the shifting of the Q register, the 2^{14} stage of A must be inserted into the 2^0 position of Q. At the same time, the 2^{14} position of Q must be retained or temporarily stored until the shift of the A register is generated.

The command to shift the Q register left ($Q_{L1} \rightarrow X$) and j is equal to 2, the 13G40 gate is fully enabled by the combined inputs from 20U02 (j = 2), 22X10 ($Q_{L1} \rightarrow X$), and 00A14 (2^{14} stage of A). The output, dependent on the contents of A14 will set the G40 flip-flop. Notice that during this command the 2^0 position of X will not be able to gate data in from the 14th position of the Q register because of the disable from 20U00 (j = 0). The output of G40, if it was set, will not apply a "1" output from 00G40 to the 28X00 inverter forcing its output to a "0". This is combined with the inputs from 10U01 ($j \neq 1$) and phase one. The phase one that will enable the gates for each stage of X will also enable the 29X00 gates. The "1" output from 29X00 is applied to the input of 00X00 resulting in A^{14} being transmitted to Q^0.

At the same time these events are occurring, the 2^{14} position of Q is one of the inputs available to the 15G41 gate. This input is combined with 22X05 ($Q_{L}1 \rightarrow X$) and 03U01 (j = 2,3) and will set the G41 flip-flop if Q^{14} is holding a binary "1". The G41 flip-flop will act as a temporary storage location for the Q^{14} position which otherwise would be lost. The action of the G41 flip-flop, regardless of whether it is set or cleared ($Q2^{14}$ a "0" of "1"), will not interfere with the transmission of A^{14} to X^{00} previously described. The input to 02G41 from 16X10 will remain a "1" until the command to transmit $A_{L1} \rightarrow X$ occurs. Therefore, the 02G41 circuit will be forced to output a "0" which will enable 28X00 until the left shift of A command is generated.

When the command to transmit $X \rightarrow Q$ occurs, the Q register will receive the initial contents of Q shifted left one place and the 2^{14} position of the A register in the 2^0 position of Q. The data originally contained by the 2^{14} stage of Q is being retained by the G41 flip-flop. The clearing of the G40 flip-flop will be done by the 12T21 inverter in the timing chain which will force the output of 00G40 to a "0", and which will partially enable the 28X00 circuit. This will allow the 02G41 circuit to control the output of 28X00 when the command $A_{L1} \rightarrow X$ (16X10) is generated. The command to shift A left one ($A_{L1} \rightarrow X$) will enable the 17X-- gates. At the same time the 02G41 inverter will be enabled by the 16X10. The G41 flip-flop, if it was set, will output a "1" to the 02G41 circuit forcing a "0" output from it. This "0" is combined with the "0" enable from 00G40 to force a "1" output to the 29X00. The applied "1" signal input to 29X00 will unconditionally produce a "0" output which will be applied to the zero side of the X00 flip-flop. This input to X00 will not change the state of the flip-flop and will cause X00 to remain in the one state, the original state of Q^{14}. The G41 flip-flop is cleared when the timing chain is recirculated in preparation for the next shift.

To summarize, figure 4-61 shows the general parts used for the transfer of data during the AQ left shift instruction. As shown, the highest order bit is retained by the G41 flip-flop during the $Q_{L1} \rightarrow X$ transmission. The figure shows a direct transmission of A^{14} to X^{00}, but in reality the bit passes through the G40 flip-flop and then to X^{00}. The result is the same. When the A register is shifted, the G41 flip-flop will be gated to the X^{00} stage thereby loading the Q^{14} position into that location. The transfer of data to Z will then complete the shift of AQ one place. This process is repeated until the shift count is equal to zero.

When the multiply instruction is executed (see table 4-19), the A and Q registers are immediately shifted to the right by the C sequence. This means the first bit of the multiplier, Q00, would normally be lost before it could be examined. To compensate for this, the value of Q00 is captured by the G41 flip-flop through the use of 13G41. This circuit will be fully enabled with "0" inputs from 20X10 ($Q_{R1} \rightarrow X$), 20U03 (j = 3), and 01Q00 if Q 2^0 contains a binary one. The output of 90H10 is the circuit that will examine the multiplier bit at the end of the C sequence. Another circuit employed during multiply is the 11X14 inverter. During the right shift of the A register, this circuit will not allow sign extension to occur as it normally does during an AQ right shift instruction. When the j value is equal to three, the input from 10U03 will be a "1" forcing a "0" output to be applied to the 12X14 circuit. Here it is combined with the "0" inputs from 63N22 and 63N35, both of which are commands to transmit $A \rightarrow X$ and are not generated at this time. The "1" output of 12X14 will disable the 15X14, thereby disabling the sign extension properties of A during the multiply.

The 15G40 circuit with combined "0's" from 20U03 (j = 3) and 22X00 ($Q_{L1} \rightarrow X$) will set the G40 flip-flop everytime the command to shift Q left one place occurs during a divide instruction. The Divide instruction is the only time the j designator can be at the value of three and a $Q_{L}1 \rightarrow X$ command is generated. As a result of this setting, the zero side of the flip-flop, 00G40, will apply a "1" to the input of 28X00 which will invert the signal to a "0" and apply it to the input of the 29X00 circuit. Here it is combined with 10U01 ("0 " if $j \neq 1$) and phase one of the clock. The result is that the 2^0 position is cleared in the X register (which relates to $Q2^0$ stage) each time the shifting left of Q is accomplished. When the partial quotient is returned to the Q register, the stage, already a zero, can remain in that state or be changed to a one depending on the evaluation of the divisor and the dividend (see table 4-21).

b. ADDER. - The Adder in the arithmetic section of the Trainer is composed of 15 binary stages. Each stage is alike with regard to circuits and functional operation. The 15 stages of the Adder consist of gates and inverters in a logical array for carrying out the required addition and subtraction operation in the arithmetic section during an operational sequence.

(1) HALF-SUBTRACT PROCESS. - The Adder is of the subtractive, one's complement, end-around borrow type. In accomplishing its addition and subtraction operations, the Adder utilizes a half-subtract process. In this process, each bit position is considered as an independent subtraction problem. The problem is carried out using the conventional rules for binary subtraction without utilizing a borrow from a higher order bit position (see table 4-25). Before the final result is obtained, any borrow generated by the half-subtract process must be satisfied. The borrow being generated by any bit position is satisfied by a higher order position. The borrow is carried through each position, even end-around, until it is satisfied. Every "1" in the borrow row indicates a borrow request to that position from a lower order bit. In table 4-26, 2^{01} needs a borrow from position 2^{04} which cannot be satisfied; therefore, the borrow becomes an end-around to be satisfied by position 2^{00}. In obtaining the final result, the half-subtract result is complemented when a "1" appears in the borrow row and is repeated when a "0" appears in the borrow row (see table 4-26).

TABLE 4-25. HALF-SUBTRACT PROCESS

BIT POSITION	2^{04}	2^{03}	2^{02}	2^{01}	2^{00}
MINUEND	0	0	1	0	1
SUBTRAHEND	1	1	0	1	0
HALF SUBTRACT	1	1	1	1	1

TABLE 4-26. HALF-SUBTRACT WITH FINAL RESULT

BIT POSITION	2^{04}	2^{03}	2^{02}	2^{01}	2^{00}
MINUEND	0	0	1	0	1
SUBTRAHEND	1	1	0	1	0
HALF SUBTRACT	1	1	1	1	1
BORROWS	1	0	1	0	1
FINAL RESULT	0	1	0	1	0

(2) ADDITION (SUBTRACTIVE). - In the Trainer, the two inputs to the Adder are from the X register, holding the contents of A, and the D register. These two inputs are not gated into the Adder. No command enable signals are needed for entry into the Adder. The contents of the two registers are always combined in the Adder, regardless of the instruction in the Trainer. The output of the Adder, however, is not transmitted until an Adder →A or Adder →S command enable is received. Due to the characteristics of the Adder logic, the contents of the D register appear to be complemented when being combined with the contents of X in the Adder. This apparent complementing of the contends of D allows for the use of the halfsubtract process in the arithmetic operations. To accomplish an Add instruction (f = 04) a number, Y, transmitted to D is added to the contents of A register which is transmitted to the X register.

The Adder modifies the inputs from X and D registers to give the resultant sum (X + D). In effect, the Adder accomplishes addition by subtraction due to the apparent complementing of the contents of D. Consider the example in which X = 5_{10} and D = 5_{10}:

NORMAL ADDITION

D = 0 0 1 0 1 = 5_{10}
X = 0 0 1 0 1 = 5_{10}
X + D = 0 1 0 1 0 = 10_{10}

ADDER

Y = 0 0 1 0 1
X = 0 0 1 0 1
D' = 1 1 0 1 0
HALF SUBTRACT = 1 1 1 1 1
BORROWS = 1 0 1 0 1
FINAL RESULT X - D' = 0 1 0 1 0 = 10_{10}

The Adder accomplished the addition of X + Y by actually doing a X - D' operation. The resultant sum, as can be seen, is the same for both operations. The Adder accomplishes a subtract instruction (f = 05) in essentially the same manner except that the original data transmission to D enters the complement of Y in D, apparently complemented in the Adder to give (D'). Using the half-subtract process, the problem becomes X - (D') which provides the difference of (X) - (Y).

(3) ADDER LOGIC. - (See figures 5-32 through 5-35.) Since any stage of the Adder is identical logicwise to every other stage, detailed discussion of one stage applies to all stages. The makeup of the Adder is such that the circuits can be separated into composite circuits, each having definite functional operations.

(a) INPUT. - The input circuits of each stage of the Adder (see figure 4-64) sample the state of X and D register flip-flops. Each of the stages has identical input circuits. The inputs from the X register and the D register are combinations of "1's" and "0's". These are combined by the input circuits to generate outputs that indicate: (1) the borrow request status of each stage; (2) indicate the borrow enable status of each stage, and (3) are used in the half-subtract process of each stage. The inputs to the stage enables are from the one side of the flip-flops in the X and D registers, while the inputs to the stage request circuits are from the zero side of the flip-flops in the two registers. The request output indicates whether a particular stage needs, or does not need, a borrow.

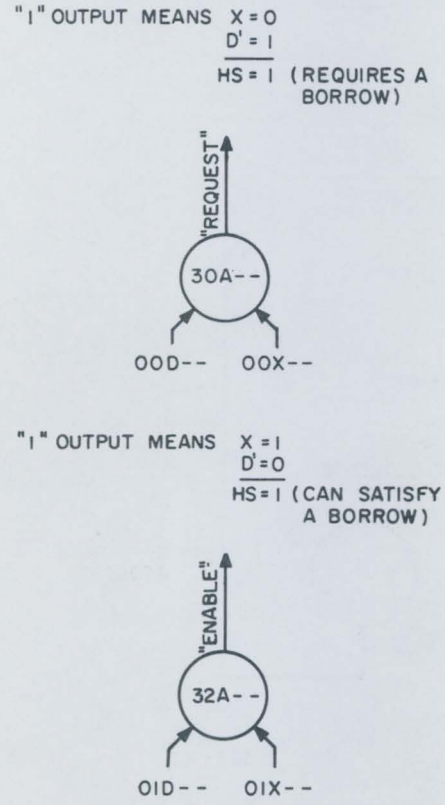


Figure 4-64. Simplified Schematic of Input

The enable output indicates whether a particular stage can, or cannot, satisfy a borrow request from another stage without a reborrow. These enables and request outputs are transmitted to other circuits in the Adder.

(1) EXECUTING THE HALF-SUBTRACT. - The enable and request outputs from table 4-27 are also used as the basis for carrying out the half-subtract process in the Adder. The enable output from the 32A-- circuit along with the request output from 30A-- are combined in the 31A-- circuit. The output from 31A-- is a logical "0" or "1" which is always the complement of the half-subtract result ($\overline{HS}$). This output ($\overline{HS}$) is applied to the 70A-- circuit which also has inputs from the 30A-- and 71A-- circuits of the preceding stage of the Adder. Because of this logical procedure of the half-subtract process the contents of D appears to be complemented in the Adder.

(b) BORROW REQUEST. - A borrow request can be generated by any stage of the Adder, or the Adder can forward a borrow request generated by a previous stage. The initial condition for a borrow request to be generated is that both the X and D registers are in the cleared or zero state. This stage of the registers will reflect the value of X equal to zero and D' equal to one. The half subtract will, of course, be one, with a borrow generated to the next stage.

The borrow request, a "1" output from the 30A-- circuit will apply its signal to four circuits. One of these circuits is the 61A-- circuit which will be forced to output a "0" (see figure 4-65). This signal is applied to the 71A-- circuit of the next higher order stage as a request to borrow from that stage. The remaining inputs to the 71AXX + 1 circuit will determine if the borrow can be satisfied by this stage. The 32A-- circuit will apply a "1" input only if the borrow can be satisfied by this stage without requiring a reborrow. The 30AXX + 1 will input a "1" only if this stage requires a borrow. Assume the borrow can be satisfied by the 71AXX + 1 stage. This would be identified by the "1" input from the 32AXX + 1 circuit. The inversion of this signal would be accomplished by the 71AXX + 1 circuit and would not forward the borrow request to the next higher order stage.

When the next higher order stage cannot satisfy the borrow request, the 71AXX + 1 circuit will be fully enabled by "0" inputs from 61AXX, borrow requested, 32AXX + 1, borrow cannot be satisfied by this stage, and 30AXX + 1, no borrow generated by this stage. The "1" output of the 71AXX + 1 circuit will be applied to the 61AXX + 1 forcing its output to a "0" which is applied to the next higher order stage as a continued borrow request that is not satisfied. From this point, the borrow is continually propagated through the Adder until a stage that can satisfy the request is found.

If a stage has received a borrow request, and it also requires a borrow, the "0" input to the 71A-- circuit from 61A-- making the request is overridden by the "1" signal into it from the 30A-- circuit

TABLE 4-27. SUMMARY OF INPUT CIRCUIT SIGNALS

INPUTS					OUTPUTS					SIGNAL FUNCTIONS
	I	II	III	IV		I	II	III	IV	
00X-- 00D--	0 0	0 1	1 0	1 1	30A--	1	0	0	0	Borrow request from all stages.
01X-- 01D--	1 1	1 0	0 1	0 0	32A--	0	0	0	1	Borrow enable from all stages.

Explanation of Input/Output signal combinations:

I: When X = 0 and D = 0, the "1" output from 30A-- indicates an unconditional borrow request from a stage. The "0" output from 32A-- indicates that a stage cannot satisfy any borrow request without a borrow.

II: When X = 0 and D = 1, the "0" output from 30A-- indicates that a stage is not generating a borrow request. The "0" output of 32A-- is the same as in I.

III: When $\bar{A}$ = 1 and D = 0, exactly as in II.

IV: When $\bar{A}$ = 1 and D = 0, the "0" output from 30A-- indicates no borrow (as in II). The "1" output from 32A-- indicates that a stage can satisfy any borrow request without a reborrow.

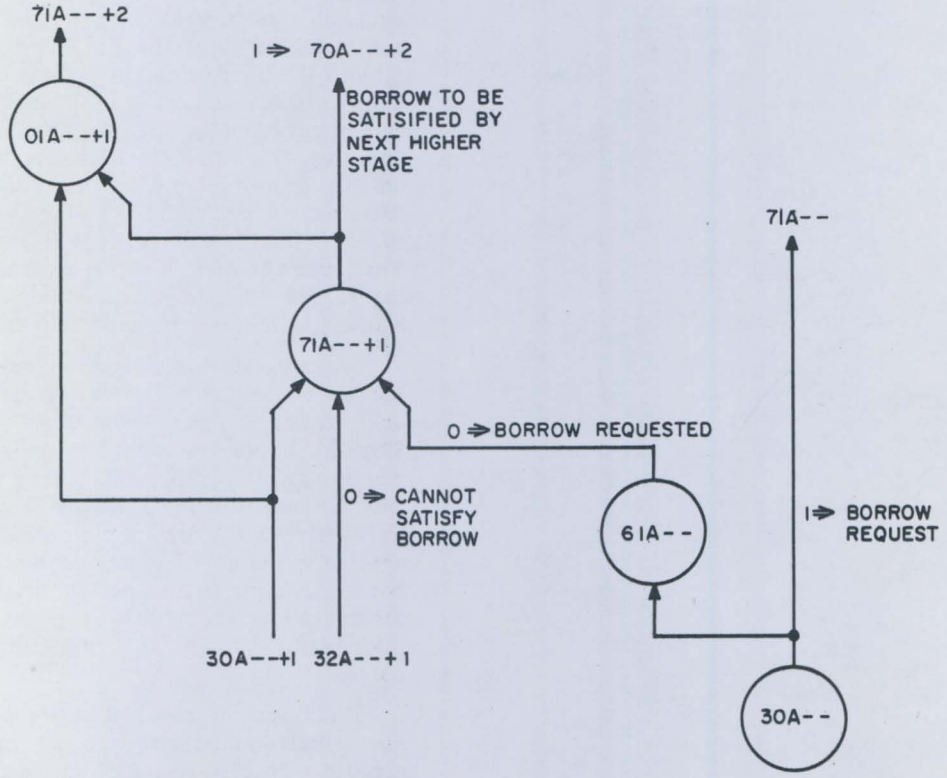


Figure 4-65. Simplified Schematic of the Borrow Request

associated with that stage. The borrow request, however, is continued by the "1" input to the 61A-- circuit in that stage.

(c) INTRASTAGE AND OUTPUT CIRCUITS. - When a no borrow condition exists, the intrastage circuits will perform the subtraction by the half-subtract method. As it was stated previously, the 31A-- circuit will output the complement of the half-subtract ($\overline{HS}$). This signal is applied to the 70A-- circuit associated with this stage. The other output circuit, 71A--, has the direct outputs of the half-subtract (HS) applied to it. To fully understand and illustrate the operation of these circuits, various values of the X and D registers will be considered.

When the borrow request conditions were considered, the 70A04 circuit received a "1" input from either the 30A-- circuit or the 71A-- circuit, both of which indicated a borrow request from some lower stage of the Adder. This "1" input will always force a "0" output from the 70A-- circuit which will partially enable the 72A--. The output of the 71A-- circuit, then, was the deciding factor as to what the output of the stage would be.

Assume a borrow has been generated from 30A03. The "1" input to 61A04 would be inverted to a "0" and applied to the 71A04 circuit indicating a borrow requested. The "1" output of 30A03 would also be applied to the 70A04 circuit which will be forced to a "0" output and will partially enable the 72A04 circuit. Notice that the "1" input from 30A03 will override any further input that may occur from 31A04 or 71A03. Therefore, the borrow will disable the normal path ("1" input to 70A04) and the borrow path (71A04) as the deciding circuit to determine the output of this stage.

If the borrow can be satisfied by the 2^4 stage, the output of 32A04 will be a "1". This is applied as an input to 71A04 forcing its output to a "0". The "0" outputs are combined at the 72A04 output circuit which will produce a "1" output. This signal will disable the gate to the A register, and A, stage 2^4, will remain cleared.

To review this action, the following illustration is offered.

	2^4	2^3
X =	1	0
D' =	0	1
HS =	1	1
BORROW =	1	0
RESULT =	0	1

If the 2^4 stage was unable to satisfy the borrow ("0" output from 32A04), the 71A04 circuit would be fully enabled with "0's" from 61A04 (2^3 requests a borrow), 32A04 (2^4 cannot satisfy the borrow request), and 30A04 (no borrow generated by 2^4 stage). This condition would force the output of 71A04 to a "1" which will be applied to the inputs of 61A04 (try to borrow from 2^5 stage), 70A05 (disable the normal path for 2^5 stage), and 72A04. The input to 72A04 will be inverted back to a "0" which will allow the setting of the 2^4 stage of the A register to one. The following illustration demonstrates this action.

	2^5	2^4	2^3
X =	1	0	0
D' =	0	0	1
HS =	1	0	1
BORROW =	1	1	0
RESULT =	0	1	1

Notice that the intended result for the 2^4 stage was to be a "0", but was complemented, because of the borrow generated by the 2^3 stage, and the borrow was carried to the 2^5 stage where the borrow of 2^3 was satisfied. Each stage of the Adder operates in the same manner.

When a borrow request is not generated by some previous stage, or the stage considered (see figure 4-66), the inputs to the 71A-- circuit will be as follows. The inputs from 30A-- will be "0", no borrow requested by this stage, 61A-- will be a "1", no borrow generated in previous stages and finally, the HS from 32A--. The "1" input from 61A-- will disable the 71A-- circuit and force its output to a "0", which will partially enable the 72A-- output circuit. The borrow path for this stage is now considered disabled and will allow the 70A-- circuit to make the decision based on the contents of X and D' for this stage. "0" inputs from 71A-- and 30A-- circuits of the previous stage indicate a no-borrow condition in the previous stage (30A--) or any other stage preceding this stage of the Adder (71A--) that has not been satisfied.

The HS inputs to the 31A-- circuit are now considered. As it was previously stated, the output of 31A-- will be the complemented HS input ($\overline{HS}$). This signal ($\overline{HS}$) will be inverted back to HS by the 70A-- circuit and applied as the input to the 72A-- output circuit. The output of 72A-- is then applied as the input to the gates of the A and S registers.

Assume the X register is holding a "1" and the D register is holding a "0". The reflected input to the Adder would be X = 1, D' = 1, and the result should be "0" in that stage of the A register. The combined inputs to the 30A-- circuit from X and D for this example would result in an output ($\overline{HS}$) of a "0". The same would be true for the 32A-- circuit resulting in a "0" ($\overline{HS}$) output from this circuit. Both the 32A-- and the 30A-- circuits apply their output to the 31A--. With both inputs now a "0", the 31A-- is forced to a "1" output or the complemented half subtract ($\overline{HS}$). This "1" signal is input to and inverted back to the HS condition by the 70A-- circuit. The combined "0" inputs of 70A-- and 71A-- to 72A-- will force a "1" output from 72A--. This signal will disable the 15A-- gate and the A register will remain cleared

in this stage when the Adder is gated to the A register. The result then reflects:

X = 1	=	X = 1
D = 0		D' = 1
	HS	= 0
	BORROW	= 0
	ADDER A RESULT	= 0

The same conditions and levels will result when X and D are both holding zeros for a given stage. The reflected inputs to the Adder would be.

X = 0	=	X = 0
D = 1		D' = 0
	HS	= 0
	BORROW	= 0
	ADDER A RESULT	= 0

For this example, both the 32A-- and 30A-- circuit would output a "0" signal to the 31A-- (HS). The inversion of the signal to a "1" by 31A-- would be applied to the 70A-- circuit which will restore the signal back to a "0" or HS. The combined "0" inputs from 70A-- and 71A-- will force the output of 72A-- to a "1" which will disable the Adder → A gate. The A register, for this stage, will remain a "0" during the command Adder → A.

The last combination that can be considered without generating a borrow is when the X register is equal to a one and D is equal to one. The reflected input to the Adder will be X equal to one and D' equal to zero as shown in the following illustrations:

X = 1	=	X = 1
D = 1		D' = 0
	HS	= 1
	BORROW	= 0
	RESULT	= 1

The HS output of the 32A-- circuit will be a "1" for this example while the 30A-- circuit is a "0". The "1" input to the 31A-- circuit will be a "0" or HS. This is again inverted to HS or a "1" by the 70A-- circuit which will force the 72A-- circuit to a "0" output. When the command Adder → A is generated, the gate will be fully enabled and the flip-flops for this stage of A will be set to the one state.

The last condition, X = 0; D = 0, reflects X = 0; D' = 1. The HS from the 30A-- circuit will be a "1" for this example forcing outputs of "0" from 61A-- (borrow request to next highest order) and 31A-- reflecting HS of "0". The "0" signal input will force the 70A-- circuit to a "1" which is applied to 72A--. The inversion of this signal will enable the gate associated with this stage of A and will set the flip-flop when the command Adder → A is generated. The result of this example is:

X = 0	=	X = 0
D = 0		D' = 1
	HS	= 1
	BORROW	=10
	RESULT	= 1 + borrow to next higher order

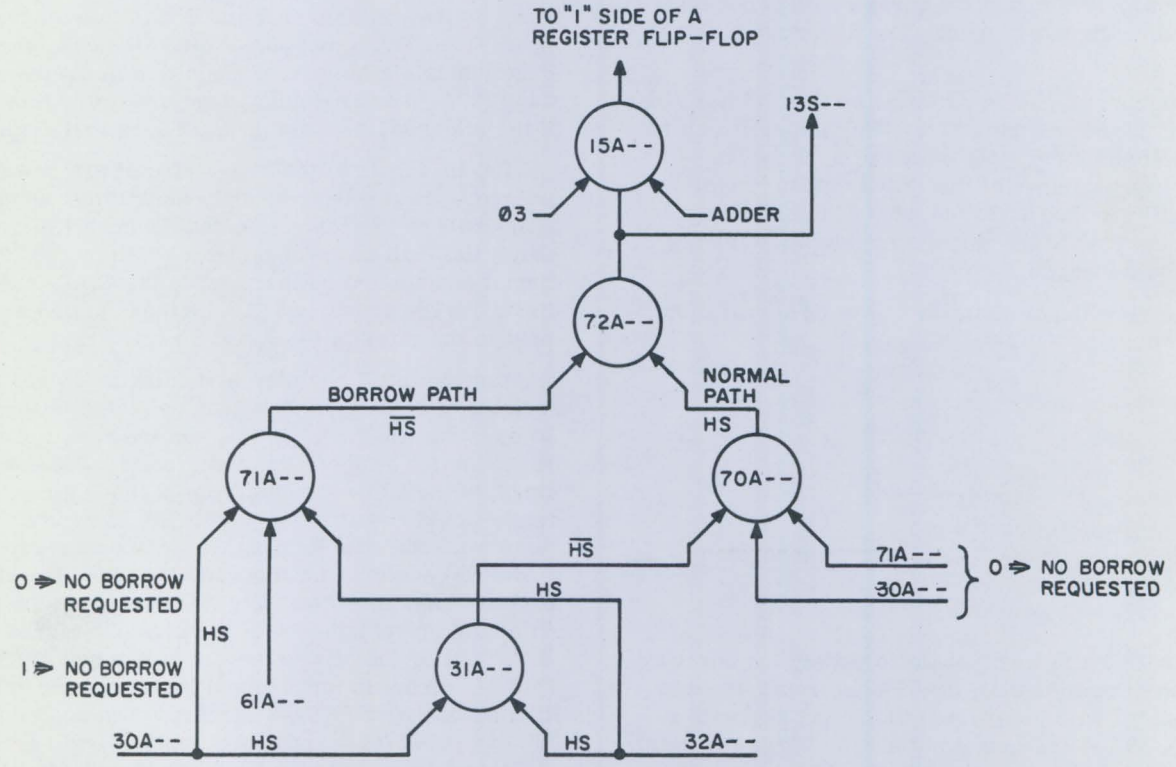


Figure 4-66. Simplified Schematic, Intrastage and Output Circuits

(d) ADDER EQUAL ZERO EVALUATION. - The B register index instruction (f = 06, j = 1 or 3) utilizes the output of the Adder to determine if the results of a subtraction would result in the difference of zero. When the 06 instruction is executed with the j value equal to one, the operand and the B register are contained in the X and D registers. Although the subtraction never takes place, the results of the subtraction can be sampled directly from the Adder. This is accomplished by sampling the output of the 71A-- and 70A-- output circuits.

It has already been determined that the 71A-- circuit will output a "0" if no borrow is generated. If the values of X and D' for a given stage are the same, the HS will be a "0" (see examples on the preceding page). Therefore, if X and D are the same, each stage will have the HS of "0" reflected from the 70A-- circuits. Because each stage is satisfied within itself, no borrows are generated. Therefore, the outputs of all of the 71A-- circuits will be a "0". The results of this condition indicate that the two values contained in X and D are equal to each other.

The outputs of the 70A-- and 71A-- circuits are sampled by the 50A70 and 50A80 circuits. If all inputs to 50A80 and 50A70 are "0's" (Y = B or Y - B, B = 0), the outputs will be a "1" for both circuits. These signals are inverted back to "0's" by the 50A71 and 50A81 and are applied as inputs to the 62N50 command inverters in the timing chain. The output of circuits 50A71 and 50A81 to the 62N50 circuit will determine if a skip condition will or will not be satisfied. Notice that only the lower nine bits of the Adder are involved in this evaluation.

(e) ADDER USE DURING DIVIDE. - The partial quotient is determined by the Adder output during divide instruction (f = 11; j = 3). The divisor, held in the D register, is compared to the dividend which was moved from the A register to X during the left shift command. The two values are immediately subtracted from each other in the Adder circuits. If the D register is larger than the value contained in X, an end-around borrow will occur; therefore, the partial quotient will be zero (D > X). However, if X is equal to, or greater than D, the subtraction can take place without an end-around borrow. This will result in the subtraction being completed by gating Adder→A and inserting a partial quotient of one into Q (X ≥ D).

The 4-A-- circuits are employed by the Adder to inform the divide logic of the result of the possible subtraction. The inputs to 40A00 and 40A01 are from the 30A-- circuits (borrow requests) of each stage of the Adder. If a request to borrow is generated by any stage, its associated 30A-- circuit will output a "1" as described in a previous paragraph. This "1" signal is also supplied as an input that will force either the 40A00 or 40A01 to output a "0". The "0" signal is returned to a "1" by the inversion properties of 41A00 or 40A01 (depending on the stage generating the borrow request) and applied as an input to 43A00. Once again 43A00 will invert the "1" to a "0" and enable circuit 45A00. If the borrow request cannot be satisfied by a higher order stage, and end-around borrow will occur, and a "1" will be applied as an input to 61A00 from either 71A14 (borrow requested by lower order stage but not satisfied by stages up to and including 2^{14}) or 30A14 (2^{14} requests a borrow). The output of 61A00, now a "0", is combined with the "0" from 43A00 which will output a "0" to 71A00 (borrow requested from 2^0 stage) and to 62A00, which will invert the signal to a "1". The "1" output from 62A00 will disable the divide logic and will indicate the dividend smaller in value than the divisor. The results will be a X→A command followed by another left shift of AQ.

However, if a borrow was requested by a given stage and this borrow was satisfied without resulting in an end-around borrow, both inputs to the 61A00 circuit from 71A14 and 30A14 would be "0". The "1" output of 61A00 will force the 45A00 circuit to output a "0" which will be inverted two more times by the 47A00 and 62A00 circuits. The 62A00 circuit will now output a "0" to the 90H11 circuit indicating that the dividend is greater than the divisor (X > D). Now subtraction will actually occur by the command Adder → A, and a partial quotient of "1" is inserted into Q00.

When the dividend and the divisor are equal to each other (X = D), the HS properties of the 30A-- circuits should all be equal to "0's". All of the 30A-- circuits are combined into the 40A00 and 40A01. If all half-subtracts are "0's" (no borrows requested), the output of both 40A00 and 40A01 will be "1's" which will be returned to "0's" by 41A00 and 41A01, respectively. The 43A00 will now be fully enabled with "0" inputs from 41A00 and 41A01 and will apply a "1" to the 45A00 circuit. For this condition, the two inputs from 71A14 and 30A14 will be "0's" to 61A00 forcing it to output a "1" to 45A00. Both inputs to 45A00 will be "1's" under these conditions. The "0" output of 45A00 will be inverted two more times by 47A00 ("1") and 62A00 ("0") as before. The "0" output of the 62A00 will gate the Adder→A and enter a partial quotient of "1" into the Q00 position.

To summarize, 62A00 output will be a "0" when the dividend is equal to or greater than the divisor because no end-around borrow has occurred (X→D). However, 62A00 will output a "1" if a borrow has been requested and could not be satisfied by the higher order stages, and an end-around borrow has occurred (X→ D).

(f) ADDER SUMMARY. - The circuits contained in each stage of the Adder are listed with brief explanations as to the functions they perform.

30A--	will output a "1" which represents a borrow requested from the stage associated with it.
32A--	will output a "1" when the stage associated with it can absorb a borrow without reborrowing.
31A--	will output the complement of the half-subtract (HS).
61A--	will output a "0" when a borrow request is generated by a given stage, or the

stage preceding it cannot absorb a borrow request.

70A-- will output the half subtract of a given stage when no borrows have been requested by stages preceding the stage associated with the 70A-- circuit. This circuit controls the normal path to the output circuit.

71A-- will output a "1" when a borrow request is received and cannot be satisfied by the stage associated with it. This circuit also controls the borrow path to the output circuit.

72A-- is considered the output circuit. Its inputs are controlled by 71A-- and 70A--, one of which will always supply a "0" input, while the other determines the output level 72A-- will produce. The output is applied directly to the gates of A and S.

50A-- samples the output of the 70A-- and 71A-- looking for the difference between two numbers to be equal to zero. The outputs are applied to the 62N50 command inverter and are used only during B register index instructions.

4-A-- circuits are employed during the Divide instructions. The results of the borrow requests, generated and satisfied before an end-around borrow occurs, will determine if X > D or X < D. a "0" output from 45A00 indicates X < D while a "1" output indicates X > D.

4-6. OPERATION OF THE INPUT/OUTPUT SECTION.

a. GENERAL. - The transfer of data to and from the Trainer is accomplished by the input/output section which utilizes two input/output channels. Data is received from and transmitted to external equipment in a Parallel mode. Channel zero is used to communicate with the electric typewriter and has the capability of handling 6-bits paralled in and out of the Trainer. Channel one is a 15-bit paralled channel which uses an adapter to communicate to any other type of external equipment. The actual data transfer to and from the Trainer is through the Q register. Along with the data, control signals are also transmitted from the Trainer to the external equipment, and to the Trainer from the external equipment. The Trainer is designed to use a DC level input/output system. Signals are DC levels which may be changed upon interchange of control information. Signals may exist for microseconds or days, depending on the nature of the particular task. The control lines, which are carried in the same cables as the data lines, have the same voltage levels. Hence, delay times, rise and fall times, and storage times are similar. The circuits and relays associated with the electric typewriter are located on chassis four of the Trainer and operate independently from the Trainer logic. These circuits, under control of the typewriter timing and control signals to the Trainer, are generated in such a way that the chassis appears to be a part of external equipment. Because of this type of breakdown, this chassis will be considered as being part of the typewriter rather than the Trainer. The discussion of the typewriter will be limited to the chassis, its circuits, relays, and inputs and outputs. The mechanical operation of the typewriter will be presented in another manual. The adapter that is associated with channel one communications will not be discussed in this section, but will be covered in appendix A to this manual.

b. DATA AND CONTROL SIGNALS. - The communication control signals and data to and from the Trainer is completely under program control. The instructions associated with f = 12 are designed to permit the control section to generate commands that control the inputs and outputs along with the proper control signals. Figure 4-67 shows the Trainer receiving input data from equipment B and sending output data to equipment A.

The direction of information flow is shown in figure 4-67. Data request signals are always sent from the external equipment to the Trainer. Acknowledge signals are always sent from the Trainer to the external equipment. The third set of control signals, called Interrupt signals in the input cable and External Function signals in the output cables, are always sent in the same direction as data flow signals.

(1) COMMUNICATION. - The sequence of events for the two cases of communication between the Trainer and external equipment follows. The terms input and output are always used with respect to the Trainer, not the external equipment.

(a) The normal input sequence for data transfer from external equipment to the Trainer is:

1. Enter Q with External Function code to activate Input mode for external equipment.
2. Trainer sends the External Function code signal (f = 12; j = 3).
3. Trainer waits for external equipment to place data word on data lines by sampling the Input Request line (f = 16; j = 0).
4. Trainer gates data into the Q register and sends the Input Acknowledge signal (f = 12; j = 0).
5. External equipment senses the Input Acknowledge line.
6. External equipment drops the data lines and the Input Data Request line.

Steps three through six are repeated for every data word until the number of words specified by the program has been transferred.

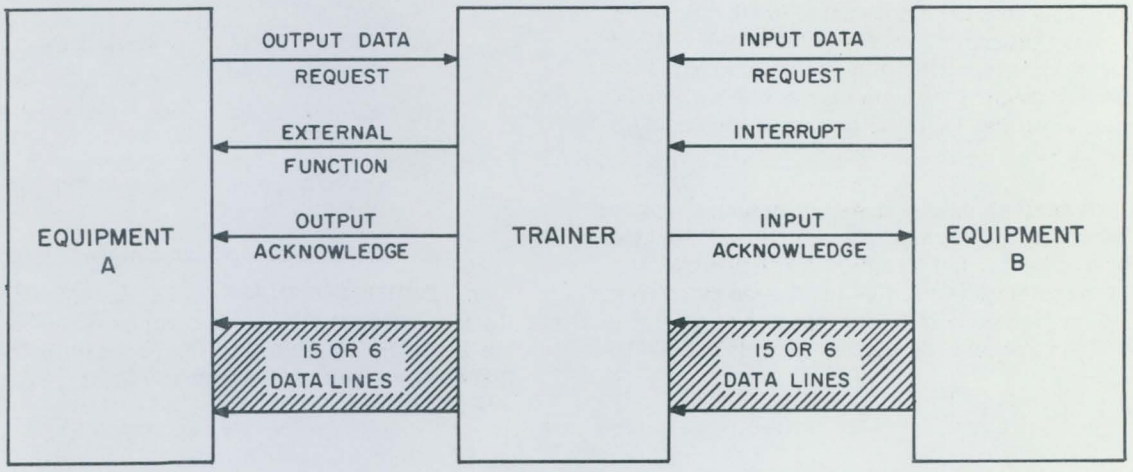


Figure 4-67. Interconnections, Trainer to External Equipment for Input and Output

(b) The sequence for external equipment transmitting an Interrupt code to the Trainer is:

1. External equipment places the Interrupt code on the input data lines.
2. External equipment sets the Interrupt line.
3. Trainer detects the Interrupt signal (f = 16; j = 2).
4. Trainer gates the Interrupt code into the Q register (f = 12; j = 0).
5. Trainer sets the Input Acknowledge line.
6. External equipment samples the Input Acknowledge line.
7. External equipment drops the Interrupt line.

Note that the Input Acknowledge is the Trainer response to either an Input Data request or to an Interrupt. To eliminate misinterpretation of the Input Acknowledge signal, external equipment must not interrupt until its last Input Data request has been acknowledged by the Trainer. Under emergency conditions, when data loss is of secondary importance, the Input Data request may be dropped and the Interrupt raised. When these conditions prevail, an Input Acknowledge signal that occurs after the Interrupt is raised, is an answer to the Interrupt.

(c) The normal output sequence for data transfer from the Trainer to external equipment is:

1. Trainer enters the Q register with External Function code to activate the Output mode for the external equipment.
2. Trainer sends the External Function code signal (f = 12; j = 3).
3. External equipment sets the Output Data request indicating that it is in a condition to accept data.
4. Trainer detects Output Data request (f = 16; j = 1).
5. Trainer sets the Output Acknowledge line indicating that data lines are ready for sampling (f = 12; j = 1).
6. External equipment may drop Output Data request any time after detecting Output Acknowledge.
7. External equipment samples the data lines.
8. Trainer drops the Output Acknowledge.

Steps three through eight are repeated for every data word until the number of words specified in the program has been transferred.

(2) FUNCTION CONTROL. - External function codes are carried over the same lines used for output

data, but control signals used with external function codes are carried in different control lines from those that carry the request or acknowledge signals. When the External Function code control line is set, this is an indication to the external equipment that the data on the lines is a function to be performed, i.e., turn on motor, turn off motor, master clear, etc. When the Output Acknowlege line is set, the external equipment will regard the data on the line as words to be processed; that is, to be typed out, punched on paper tape, etc.

The external equipment does not return a control signal indicating the receipt of the External Function code. The only way the Trainer is notified that the external equipment has received and responded to the External Function code is by the receipt of the Input or Output Data request, depending on which mode was activated.

The following sequence gives the steps for transmitting an External Function code from the Trainer to external equipment.

1. Trainer enters External function into the Q register.
2. Trainer sets the External Function line (f = 12; j = 3).
3. External equipment detects the External Function line.
4. External equipment samples the code on the data lines.
5. Trainer drops the External Function line.

c. INPUT/OUTPUT CONTROL. - The control of the input/output section is completely under control of the main timing chain and the program being executed. Instructions associated with the input/output section are as follows:

FUNCTION CODE	INSTRUCTION	OPERATION
f = 12; j = 0	Input	Data to Q; send Input Ack on channel Y
f = 12; j = 1	Output	Q out; send Output Ack on channel Y
f = 12; j = 3	External Function	Q out; send External Function on channel Y
f = 16; j = 0	Input Jump	Jump if no Input request.
f = 16; j = 1	Output Jump	Jump if no Output request.
f = 16; j = 2	Interrupt Jump	Jump if no Interrupt.

It can be seen from the Input/Output instruction that the activities of the input/output section can be broken down into five groups. These are: 1) External Function; 2) Output; 3) Input; 4) Input/Output Jumps, and 5) Interrupts.

(1) EXTERNAL FUNCTION. - The External Function signal associated with the proper External Function code gives the Trainer the ability to control remotely the connected external equipment. Table 4-29 shows the external function codes recognized by the electric typewriter. The proper code must be entered into the Q register in preparation for sending the external function. The External Function instruction can now be executed by the control section. The 2^0 bit of the Instruction word will determine the channel on which the External Function signal will be sent. If the 2^0 bit is equal to zero, channel zero will be selected. If the 2^0 bit is equal to one, the External Function line will be set on channel one.

Assume that the function code is entered into the Q register. The data in Q immediately is made available to both channels and will allow the lines to change. The channel on which the External Function control signal is transmitted will determine which external equipment will respond to the code. After the A sequence has read the External Function instruction into the U register and the lower nine bits of the instruction to the X register, the C sequence will be enabled by the 61N82 command inverter. Once the timing chain is allowed to advance, the only command generated is from the 63N34 command inverter controlled by the T23 flip-flop. The "1" output of 63N34 will be inverted by the 63N37 inverter which will apply "0" inputs to the 03E04 and 05E04 circuits (see figure 5-14). The 2^0 bit of the X register is combined with the External Function bit command at these gates. If X, 2^0 stage, contains a "1" at this time, the 03E04 circuit is fully enabled and will produce a "1" output to the 04E04 inverter. The "0" output of 04E04 will be applied to six 330-- circuits. These circuits will gate the lower six bits of the Q register into the electric typewriter control (figure 5-48) where the immediate translation of the Function code will take place. However, if the X register, stage 2^0, contains a "1", the 05E04 gate will be fully enabled, the E04 flip-flop will be set, and the zero side will output a "1" which will be made available to the special Input/Output adapter (figure 5-53). The "1" input to 60Y16, output driver, will set the External Function

TABLE 4-28. EXTERNAL FUNCTION CODES FOR ELECTRIC TYPEWRITER

FUNCTION	BIT POSITION 5	4	3	2	1	0	OCTAL REPRESENTATION
ENABLE INPUT	1	0	0	0	0	0	40
ENABLE OUTPUT	0	1	0	0	0	0	20
MASTER CLEAR	0	0	1	0	0	0	10
FAULT	0	0	0	0	0	1	01

line on channel one. Card 99052 is an indicator driver which will light the EXTERNAL FUNCTION indicator on the adapter panel. The E04 flip-flop will remain set until the timing chain has again been enabled and initiated to the 13T23 inverter or approximately 8 microseconds.

The External Function control line for channel zero will gate the Function code from Q to typewriter control immediately, while the control line for channel one is set for approximately 8 microseconds. The time allocated to channel one is allowed because of the various timing considerations and types of external equipments that possibly could be connected to it. However, channel zero is internally connected to typewriter logic on chassis four, which makes possible the immediate gating of the Function code from Q into the typewriter control translator.

(2) OUTPUT MODE. - The Trainer is operating in the Output mode when it is transmitting data to external equipment. To accomplish the Output mode, a program of instructions that will enter the Q register with the data to be transmitted and send the Output Acknowledge, must be executed once each time a data word is sent to the external equipment. Necessary steps must be taken to assure that the external equipment has had ample time to sample the initial contents of the Q register before the data in Q is changed. One method of accomplishing this is offered by the flow diagram in figure 4-68. Following this method, or a facsimile thereof, will insure that the external equipment has received the transmitted data before the Q register is altered.

Before the Output operation can be discussed, it is necessary that the normal, or static state of the Output control signal circuitry be examined (see figure 5-14 and 4-69). During the static state, the output request lines will be at "0" levels (no output request). The combined "0" inputs from 13O10, channel zero, and 50Y15 channel one, will force the output of 13000 to a "1" output. The "1" output of 13000 will set the 000 flip-flop which in turn will partially enable the 15000 circuit, although the output of 13000 will still force a "0" output from 15000. The 17000 will be fully enabled with the execution of an Output Jump instruction (f = 16; j = 1, Jump if no Output request) which, for these conditions, will set the Jump flip-flop (01G50). This then, is the static state the circuitry will assume without further action by the Trainer or the external equipment.

After it has been determined that the external equipment has generated the Output Data request, indicating it is immediately available to accept a data word, the Q register must be entered with the word to be transmitted. The receipt of the Output requests from either channel zero or one will appear as a "1" signal to the 13000 inverter. The "1" signal is inverted to a "0" which will fully enable the 15000 inverter. The "1" level output of 15000 will now disable the 17000 circuit thereby disabling further Output jump conditions. Assuming the next

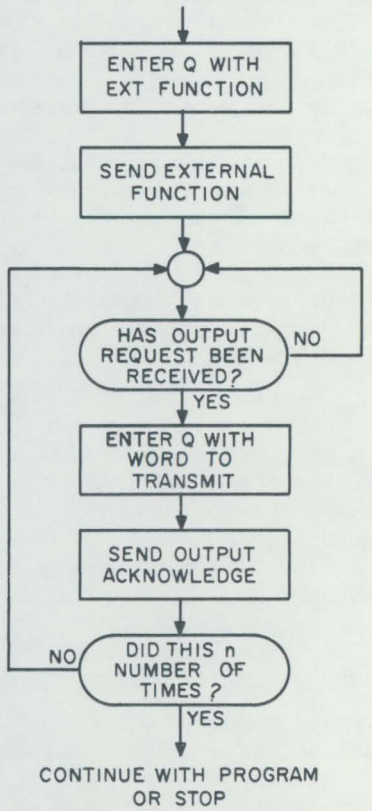


Figure 4-68. Flow Diagram for Output Operations

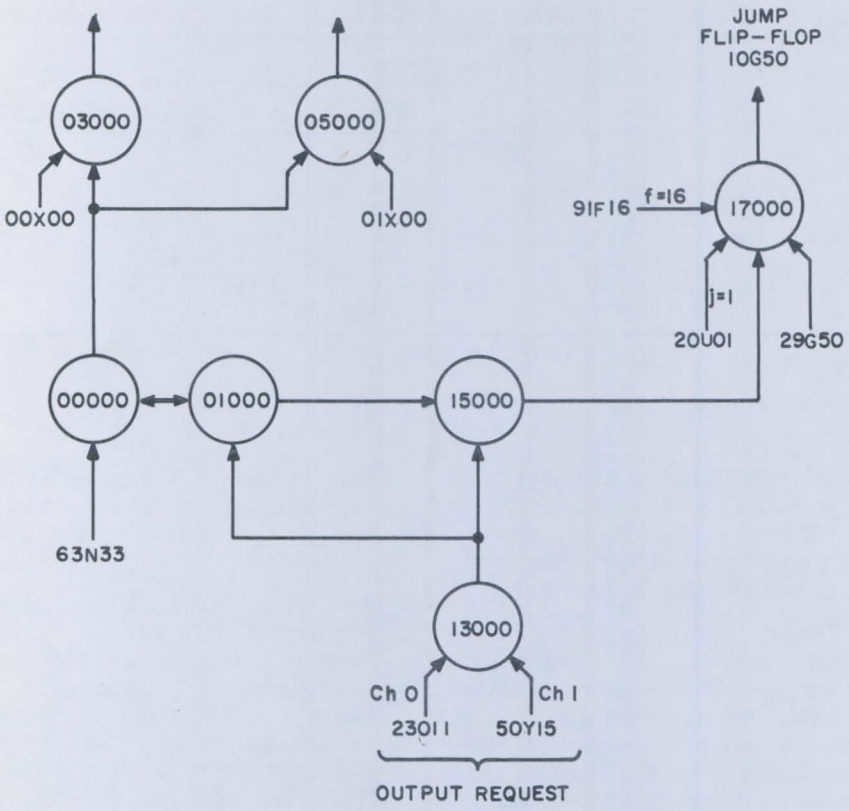


Figure 4-69. Output Control Signal Circuitry

instruction to be executed is the Output instruction (f = 12; j = 1) the A sequence, as for all instructions, will place the Function code in the U register and the lower nine bits in X. The C sequence is enabled directly from the A sequence via command inverter 61N82. Under control of the T23 flip-flop and function code, all inputs to the 63N33 will be "0's" which will allow the command Send Output Acknowledge to be generated. This command, now a "1" will clear the 000 flip-flop. A "1" is now produced by 01000 and applied to 15000 which will invert the signal and again enable the 17000 inverter. Output jump instructions will not be satisfied even though the initial Output request is still present.

From the zero side of the 000 flip-flop "0's" are applied to the 03000 and 05000 circuit. This enable is combined with the 2^0 bit of the X register to determine which channel will output the Output Acknowledge control signal. If the X register is holding a "1" in the 2^0 bit position, the "0" output of 00X00 will enable the 03X00 and the Output Acknowledge will be transmitted on channel zero. However, if the 2^0 bit position is holding a "1", the 05000 circuit will be fully enabled by the "0" output from 01X00 and channel one will output the Output Acknowledge signal.

The external equipment will sample the Output Acknowledge, and upon receipt of it, gate the data from the line into the logic associated with it, and drop the Output Data request.

When the request line is returned to a "0" level, indicating the external equipment has received the output data, the 13000 circuit will once again be fully enabled. The "1" output to 15000 will maintain the "0" input to 17000, while the "1" input to 01000 switches the state of the flip-flop. The "1" output from the zero side will now remove the Output Acknowledge control signal and the "0" output from the one side will partially enable the 15000 circuit, and the circuitry is returned to the static state.

When the external equipment has processed the received data word and is ready to receive additional data, the Output Data request is again returned to a "1" level. Through the inversion properties of 13000, the 15000 now will be fully enabled and will apply a "1" to the input of the 17000 circuits. This will disable any possible Output Jump instruction indicating the Output Request has been received and the entire output operation is recycled.

If for some reason the Output Data request is not returned to "0" after the Output Acknowledge line has been set, no further controlled communications can be accomplished. This is caused by the 000 flip-flop taking on the properties of a logical one-shot. Assume the Output request was received by a "1" input to 13000. When the request was acknowledged, the 000 flip-flop was cleared. The "0" output from the zero side of 000, combined with the output of the 2^0 bit position of X, will select the channel of which the

Output Acknowledge is sent. The "1" output from the one side of 000 will be inverted by the 15000 circuit, which will continue to enable 17000, indicating no Output Data Request. The only way the 000 flip-flop can be returned to the one state is by the initial Output Data Request returning to a "0". When this condition prevails, the sampling of 17000 by the Output Jump instruction will continuously be satisfied, indicating that the Output Data Request has not been received.

Consideration must also be given to the fact that if the 000 flip-flop remains set as outlined in the previous paragraph, the Trainer will continue to execute instructions in a normal manner. Because the X register is involved in many arithmetic operations besides receiving the lower nine bits of the Instruction word for each instruction that is read, the 2^0 bit position could be changed to hold ones and zeros continuously. This bit position has its output applied directly to 03000 and 05000. This set of circumstances could result in the Output Acknowledge signal transmission on channel zero or one at all times.

Notice should also be directed to the fact that the Output Data Request control lines of both channels are terminated as inputs to the 13000 circuit. It cannot be determined which channel produced the Output Data Request during the sampling of 17000 with the Output Jump instruction. Because of this, a programming limitation must be placed on the input/output section to ensure that only one piece of external equipment, on channel zero or one, be activated at any one time.

(3) INPUT MODE. - The logic circuits that control the input control signals are identical to those employed by the Output mode of operation. The primary difference is that during the command to send the Input Acknowledge, special considerations are given to the gating of data into the Q register depending on which channel is transmitting the data to the Trainer. The static state of the circuits 17I00, 13I00, and the I00 flip-flop are the same as those considered during the Output discussion.

After the External Function code has been sent to the external equipment, the assembly, or reading, of one data word will take place. When the external equipment is ready to transmit the word to the Trainer, it sends the Input Data Request signal. This signal will appear as a "1" input to 18I00 which will output a "0" to 19I00. The "1" from 19I00 is applied to 17I00 which will disable the jump conditions for an Input Jump (Input Data request received). The "1" output of 19I00 will also force 13I00 to a "0" output which will have no effect on the state of the I00 flip-flop.

The reading of the Input instruction (f = 12; j = 0) by the A sequence will place the upper six bits of the Instruction word in the U register and the lower nine bits in X. The C sequence is enabled by the A sequence via command inverter 61N82. Under the control of the T23 flip-flop and the Function code, both the 63N32 and 63N38 command inverters are partially enabled (see figure 4-70). The 2^0 bit of the X register

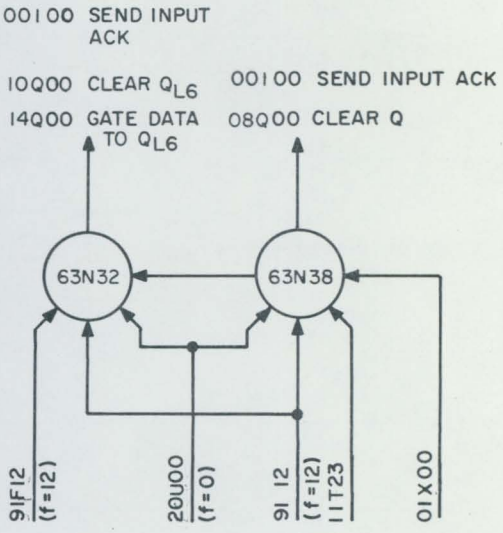


Figure 4-70. Data to Q Register Selection, Input Mode

will be the deciding factor as to which inverter will issue the command. If the input to 63N38 from 01X00 (the one side of bit 2^0) is a "0", indicating input data from channel one, command inverter 63N38 will be fully enabled and will clear the entire Q register in preparation for the receipt of the data. However, if the 01X00 input is a "1", the output of command inverter 63N38 will be forced to a "0" output, which will fully enable 63N32. The "1" output of 63N32 will clear only the lower six bits of the Q register and gate the six bits from the typewriter logic to the Q register (14Q00). The command to generate the Input Acknowledge is also issued.

The "1" levels, be it from 63N38 or 63N32, will clear the I00 flip-flop. The "0" output is applied to the 03I00 and 05I00 circuits where they are again combined with outputs from the 2^0 position of the X register. If channel one is selected, the output of 05I00 will be applied to circuit contained in the adapter to allow the gating of the data to the Q register. Unlike channel zero, the gating of the data on channel one is not clock phased (see figure 4-71.).

(a) INPUT ON CHANNEL 0. - As it can be seen by the timing chart in figure 4-71, the commands issued by the 63N32 will last for the length of time the 11T23 flip-flop is set. The following phase two will clear only the lower six bits of the Q register in preparation to receive data from the typewriter logic which will occur on the next phase of the clock (Ø3). The sending of the Input Acknowledge is delayed for

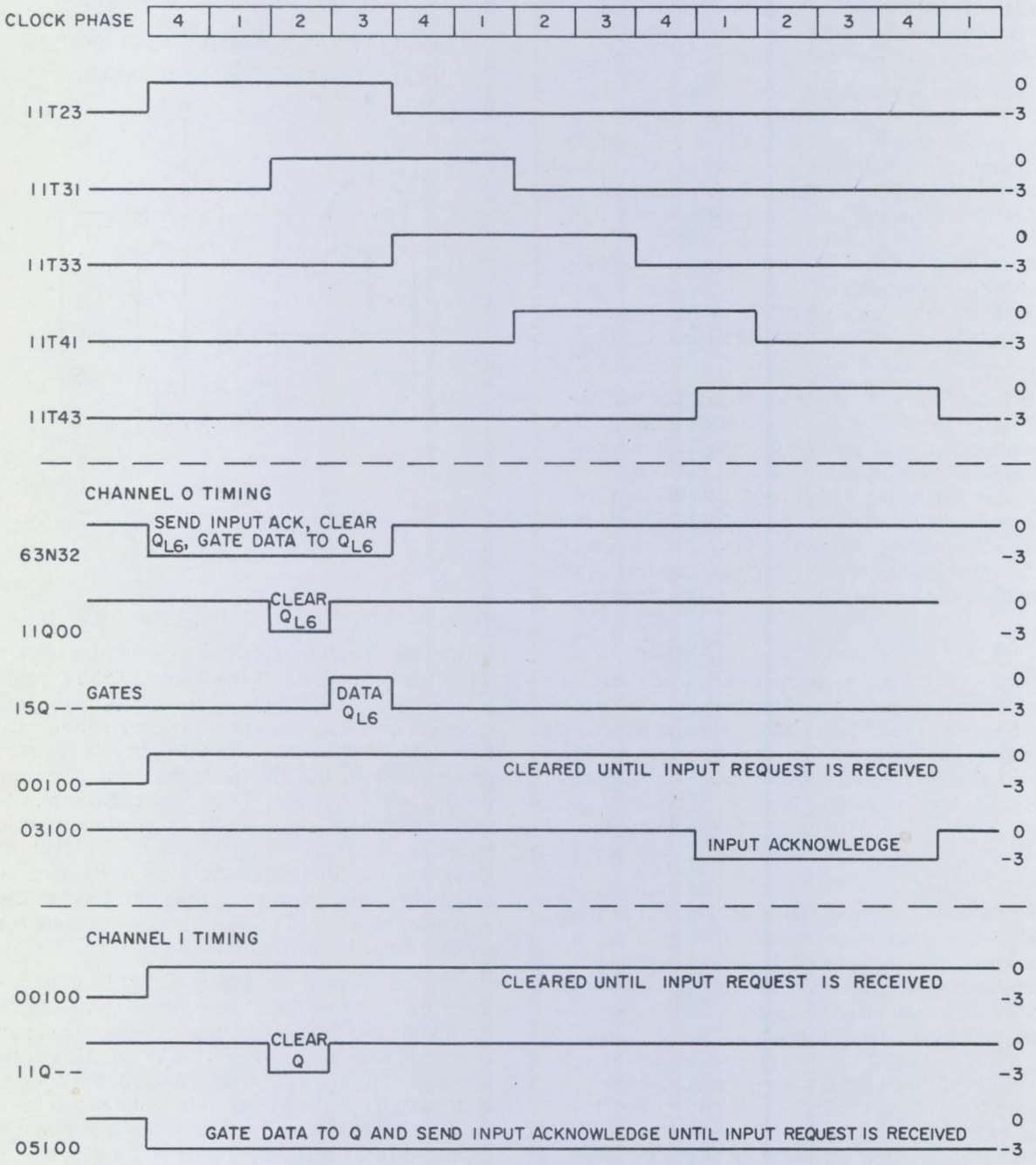


Figure 4-71. Input Data Timing

approximately 2 microseconds after the data is entered into Q. The generation of the Acknowledge signal will clear the output register of the typewriter logic; therefore, it is withheld until the Q register is properly entered with the data. The time that the Input Acknowledge signal is available to the typewriter circuits is controlled by the input of 11T43 to 03I00. This signal is made available for one clock cycle or 1.6 microseconds.

(b) INPUT ON CHANNEL 1. - Channel one has the capability of handling the full contents of the Q register, or 15 bits in parallel. The output of this channel terminates in a special adapter which contains line drivers and input amplifiers which will match the line voltages necessary to communicate with magnetic tape units, magnetic drums, high-speed paper tape readers and punches, or any other piece of external equipment that may be required to complete a given installation.

When the I00 flip-flop is cleared as controlled by the 63N38 inverter, the outputs to the 60Y-- circuits will gate the data contained on the cable directly into the Q register. This enable will occur at phase four time, except for the slight inherent delay of the circuits. The following phase two, the command to Clear Q, will be issued two phases after the gating action had started. Actually, those flip-flops of Q receiving a one bit by the gating action, will try to go to the "0" state, but the inputs to the one side will remain after the Clear Q command is terminated. If any stage of Q was receiving a binary zero, the one side of the flip.flop would have a "0" input and the clearing action would take place.

Unlike channel zero, the Input Acknowledge is allowed to remain on the line to the external equipment until the initial Input Data request is dropped.

(4) INPUT/OUTPUT JUMPS. - There are three jump instructions available that allow the sampling of the input/output section. This is the only means of determining the receipt of an Input Data request, Output Data request, or an Interrupt. It is through the use of these instructions that the Trainer can determine if the external equipment can receive or is ready to transmit another data word. Notice that there is no selection of the channel for these instructions, therefore, any request, regardless of channel, will satisfy the Jump condition.

The translation of the Input jump is to jump if no Input Data request has been received. This is controlled by the 17I00 circuit which has five inputs to it: 91F16 (f = 16); 20U00 (j = 0); 29G50 (command to sample possible jump conditions from the timing chain, 10T43). The last two inputs, 01I00 and 19I00, are the results of the Input Data request status.

When the static state of this circuitry was examined, it was determined that the I00 flip-flop was set with no request present. The input to 17I00 then will be a "0" from 01I00. The lack of a request on either channel will produce "0" inputs to 18I00 forcing a "1" output which is applied to, and inverted, by 19I00. This "0" output will fully enable the 17I00 inverter, and upon command, will output a "1" that will set Jump flip-flop G50. Each Input Jump instruction executed from this point will be satisfied until an Input Data request is received.

The receipt of the Input Data request will appear as a "1" input to the 18I00 circuit, which will invert the signal to a "0". This signal is applied to 19I00 causing it to output a "1". This will act as a disable to the 17I00 circuit and the jump will not be satisfied. Instead, the next sequential instruction of the program will be read and executed.

When the I00 flip-flop is cleared by the command to send the Input Acknowledge, the "1" output from 01I00 will continue to disable the 17I00 circuit. This condition will be maintained until the request is dropped back to a "0". Through the inversion properties of 18I00, 19I00, and 13I00, the I00 flip-flop will be returned to the "1" state and will once again enable the 17I00 circuit. Input jump conditions will again be satisfied.

Output jumps are controlled in the same manner when the Function code is 16 (91F16), and the j value is one (20U01). These conditions, combined with the output of 15000 will satisfy jump conditions based upon the receipt or absence of the Output Data request. More detailed coverage of the Output jump is given in paragraph c (2), Output Mode, in this section. Once the Jump flip-flop is set the execution of the jump is accomplished as discussed in paragraph (f), Jump Instruction.

(5) INTERRUPT AND INTERRUPT JUMP. - Certain pieces of external equipment such as magnetic tape units have the capability of generating an Interrupt signal. This signal may be generated at any time the system is in operation. It is a manner in which equipment can inform the Trainer of certain conditions that should be known. These conditions may change with the various pieces of equipment that may be connected to the Trainer, so a hypothetical set of conditions will be considered.

Assume a tape unit has the capability of developing a status report, that is, a code that can be transmitted to the Trainer indicating its present status. Example of the code could be interpreted as being writing, reading, inactive, etc. Usually codes of this type would be transmitted to the Trainer upon request. When certain events such as tape break, loss of power, out of tape, etc., occur the Trainer should be informed immediately. A resultant typeout on the typewriter will inform the operator of the malfunctioning equipment so that necessary corrective action can be initiated. The Interrupt signal would be associated with this type of report. Conditions such as those above make periodic checking of the Interrupt line necessary. Figure 4-72 shows the program logic that could be used during interrupts.

(a) INTERRUPT LOGIC. - The Interrupt signal utilizes some of the same circuits normally employed by the Input operation. The I00 flip-flop is in the set state and the "0" output will partially enable

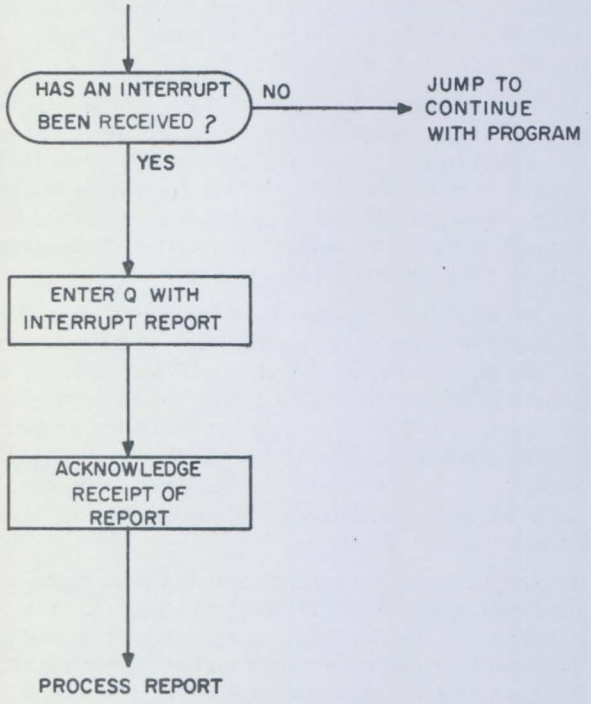


Figure 4-72. Flow Diagram for Interrupts

the 15I00 and the 17I00 circuits. When the Interrupt Jump instruction is executed and no interrupt has been generated, the 15I00 circuit will be fully enabled by "0" inputs from 91F16 (f = 16), 20U02 (j = 2), 50Y17 (no interrupt), and 29G50, an enable generated by the 10T43 flip-flop of the timing chain. For these conditions, the jump will be satisfied and executed each time the 15I00 circuit is sampled.

If the Interrupt were to occur, the reporting equipment would place the interrupt, or status report, on the input cable followed by the Interrupt signal. This signal would appear as a "1" input to the 13I00 and 15I00 circuits. The input to 15I00 will disable the circuit and the jump condition for f = 16; j = 2 will not be satisfied, indicating the receipt of an interrupt. When it has been determined that an interrupt was generated, a suggested procedure would be to store the A, Q, and possibly the B registers in memory so the contents of these registers would be preserved during the processing of the interrupt report.

The Input instruction will gate the report into the Q register where it can be determined which of the equipments generated the Interrupt, the reason for it, and finally, the action that must be taken because of it. The execution of the Input instruction will have the same affect on the circuitry as it did during normal input operation. That is, the I00 flip-flop will be cleared which will send the Input Acknowledge to the reporting equipment. This signal will indicate to the external equipment that the Interrupt has been received and to drop the interrupt line. When this occurs, 13I00 and 15I00 will receive a "0" as it did before the Interrupt was granted. The "0" input to 13I00 will be combined with the output of 19I00 ("0" if no Input Data request) to force 13I00 to output a "1". This will return I00 back to the "1" state, dropping the Input Acknowledge signal and re-enabling the 15I00 circuit.

The explanation was given in such a manner that the Interrupt could be used. It is not intended or suggested that its use be confined only to this type of operation. As an example, a piece of equipment could be designed, such as a manual keyboard type of entry, that could input data periodically through the use of the Interrupt line. Another consideration that should be brought out is that all interrupts will occur on channel one. The typewriter circuitry, which utilizes channel zero exclusively, does not have the capability of generating an interrupt signal or status report, therefore, the programmer can be assured that the Interrupt signal will be originated by one of the equipments connected to channel one.

d. THE ELECTRIC TYPEWRITER.

(1) GENERAL. - The electric typewriter was selected as the initial piece of external equipment because:

1. It has a self-contained paper tape reader.
2. It has a self-contained paper tape punch.
3. Typeouts are available directly from the Trainer.
4. It can be used to prepare a paper tape.
5. Few control and logic circuits are required.

The primary disadvantage of the Input/Output typewriter is the lack of speed.

The circuits associated with the typewriter are contained exclusively on chassis four of the Trainer. These circuits can be broken down into three main groups, namely: Control, Input, and Output. The typewriter can be controlled entirely by the output of the Trainer through the use of these circuits and associated relays. This includes the turning on and off of the typewriter motor as well as the selection of the Input or Output modes (see table 4-29). Letters and numerics can be typed out, controlled by the selection of the codes associated with them (see table 4-29). For example, if code 45 was sent to the typewriter logic, a carriage return would be executed; a code of 30 would result in the typeout of the letter A.

(2) CONTROL PANEL. - The control console for the typewriter is located in the lower right-hand corner of the main control console panel. These controls are:

INPUT: Pushbutton indicator. When the Trainer has control of the typewriter, the selection of the Input mode will turn on the Input indicator. The Input mode may also be selected manually by depressing

TABLE 4-29. TYPEWRITER CODES AND SYMBOLS

FLEX CODE TRANSLATION TABLE ALPHANUMERIC TO OCTAL											
UPPER CASE	LOWER CASE	OCTAL NUMBER	UPPER CASE	LOWER CASE	OCTAL NUMBER	UPPER CASE	LOWER CASE	OCTAL NUMBER	UPPER CASE	LOWER CASE	OCTAL NUMBER
A	a	30	J	j	32	S	s	24	2	2	74
B	b	23	K	k	36	T	t	01	3	3	70
C	c	16	L	l	11	U	u	34	4	4	64
D	d	22	M	m	07	V	v	17	5	5	62
E	e	20	N	n	06	W	w	31	6	6	66
F	f	26	O	o	03	X	x	27	7	7	72
G	g	13	P	p	15	Y	y	25	8	8	60
H	h	05	Q	q	35	Z	z	21	9	9	33
I	i	14	R	r	12	1	1	52	0	0	37

TYPEWRITER MOVEMENTS (UPPER OR LOWER CASE)			
TAPE FEED	00	SHIFT UP (TO UPPER CASE)	47
COLOR SHIFT (OF RIBBON)	02	TABULATE (TO PRESET POS.)	51
SPACE	04	SHIFT DOWN (TO LOWER CASE)	57
STOP (READING TAPE)	43	BACKSPACE	61
CARRIAGE RETURN LINE FEED	45	CODE DELETE	77

SIGNS AND PUNCTUATION MARKS				
			INTERPRETATION	
UPPER CASE	LOWER CASE	OCTAL NUMBER	UPPER CASE	LOWER CASE
-	_	56	Minus sign, Hyphen (Exponent in superscript)	Minus sign, Hyphen
.	=	44	Multiplication DOT	Equality sign
/	+	54	Fraction Bar, Virgule	Plus sign
(	,	46	Left Parens	Comma
)	.	42	Right Parens	Period
—	|	50	Underline	Vertical bar (Absolute Value sign)

the INPUT pushbutton. The CLEAR pushbutton located directly below the INPUT pushbutton will clear the Input mode.

OUTPUT: Pushbutton indicator. When the Trainer has control of the typewriter, the selection of the Output mode will turn on the OUTPUT indicator. The Output mode may also be selected manually by depressing the OUTPUT pushbutton. The CLEAR pushbutton located directly below the OUTPUT pushbutton will clear the output mode.

FAULT: Program fault code (01) does nothing but set a flip-flop which, in turn, turns on the FAULT indicator. This could be utilized in programs that check for accuracy of the data and the contents of words being sent to the Trainer. If a bit or data word is improperly read by the typewriter, the Trainer could send the fault code to typewriter logic which will give the operator a visual indication that a fault condition exists. The CLEAR pushbutton located directly below the FAULT indicator will clear the flip-flop and extinguish the FAULT indicator.

MASTER CLEAR: By depressing this pushbutton, or under Trainer control, the receipt of a function code of 10 octal to the typewriter logic will clear the registers and circuits of the typewriter. The Master Clear command may be used in lieu of the Disable Input or Disable Output commands in that it will clear all circuits.

COMPUTER CONTROL SWITCH: The position of this switch will determine if the Flexowriter will operate in the on-line or off-line mode.

(3) INPUT/OUTPUT TYPEWRITER LOGIC. - The input/output typewriter can be sectionalized into three logical parts: logic control, logic output, and logic input. Each of these operations will be discussed. Since both output and input rely on the logic control section for proper operation, the logic control section will be discussed in conjunction with input and output modes.

The input/output typewriter, designed primarily to operate with Trainer control (on-line), also can be operated manually (off-line). When manual operation is selected the typewriter unit is operative and Trainer control is disabled.

The off-line and on-line methods of operations are logical choices for the separation of the functions of the input/output typewriter. The functional analysis of each method will be discussed in the following paragraphs.

(a) OFF-LINE. - The off-line operation of the typewriter is controlled by the COMPUTER CONTROL switch on the control panel of the Trainer. When in the up position, off-line operation is selected. Figure 5-48 shows the schematic representation of this switch.

The COMPUTER CONTROL switch is the controlling factor in determining the means of energizing the start motor relay, K11, causing power to be applied to the Flexowriter (115 VAC 60 cycle). Off-line operation is accomplished by placing the COMPUTER CONTROL switch in the up position. By this action, ground is applied, through the now closed contacts of S8, to the input of the relay puller 50Y00. With -15 volts applied to pin 14 of the K11 relay and a ground path now available through 50Y00, K11 will now become energized which will supply power to the Flexowriter. It should also be noted that the output of 40O00 is held at ground during off-line operations making it impossible for the circuits of the Flexowriter control to govern the actions of the Flexowriter.

Contact 1 of S8 will also supply a ground to the input of relay puller 30Y10 (see figure 5-51). With a continuous ground applied to the input of 30Y10, relay K8 will energize. The associated contacts of K8 will provide a continuity path for the Reader Clutch of the paper tape reader and will turn on the READY light on the front of the Flexowriter. With power now applied and a path for the reader clutch provided, the Flexowriter may now be operated as an independent piece of equipment.

When the COMPUTER CONTROL switch is in the down position, the ground is removed from the 30Y10 and 50Y00 relay pullers. Both the power source and the reader clutch are now under the control of the Flexowriter circuit and the inputs received from the Trainer. The ground required by the 50Y00 circuit to energize K11 will originate only by one of three inputs to 40O00 providing a "1". The control of the reader clutch is directly under the control of the 13I10 inverter during Input operations. Complete coverage for the controlling of the reader clutch will be offered in the INPUT MODE discussion in a following paragraph.

(b) ON-LINE. - On-line operation allows the Trainer to select the function it wishes the typewriter to perform. When on-line operation is desired the COMPUTER CONTROL switch is placed in the down position. The Trainer will furnish the control signals necessary for the execution of a specific operation of the typewriter.

(1) CONTROL CIRCUITS. - Logic control circuits for the typewriter are shown on figure 5-48. The control information the Trainer sends to the typewriter is in the form of external function codes, master clear words, and input error words. The Trainer also sends input and output acknowledge signals to the typewriter, but these signals are concerned primarily with data transmission and will be discussed in the logical analysis of the input and output modes of operation.

(a) EXTERNAL FUNCTION CODE. - When the typewriter is operating as an input or output device to the Trainer, some means of control must

exist so that the Trainer may specify the function desired. Scuh control is accomplished by means of an external function code transmission from the Trainer to the typewriter. The purpose of the transmission is to initiate within the logic circuits of the external device the performance of a specific function. The functions that may be established within the typewriter are listed in table 4-28.

The External Function code enters the control circuits through the output of 04E04. When the External Function code is present, the output of 04E04 will apply a "0" input to each of the 34C--circuits. This ground level is sent as the External Function Enable signal to gate the various External Function words through their respective input amplifiers.

(b) MASTER CLEAR. - There are times in the operation of the equipment when it is necessary that the logic circuits be returned to their normal states. The Master Clear control circuit of the typewriter logic performs this function on command of the Trainer as the result of a Master Clear word transmission, or from the manual operation of a switch. The Master Clear control circuits are shown on figure 5-48.

One input to the Master Clear circuitry is the Master Clear function word transmission from the Trainer. When the Master Clear word is present, along with the External Function signal, the inputs of 34C01 are both at ground level. A "1" output of 34C01 is inverted to a "0" by 35C01 and applied as the input to 36C01. After inversion by 36C01, the "1" output will clear F00, the Fault flip-flop, I00, the Input Mode flip-flop, and O00, the Output Mode flip-flop. The "1" outputs of 01I00 and 01O00 will result in the clearing of the input and output logic and control circuits.

A second means to Master Clear is by depressing the MASTER CLEAR pushbutton on the control panel under FLEXOWRITER. When this pushbutton is depressed, the output of 35C01 is grounded reflecting a "0" input to 36C01. The distribution of the clearing signal is as previously mentioned.

(c) INPUT FAULT. - If the Trainer detects an error in the transmission of data from the typewriter, it may be programmed to generate an Input Fault function word. When the typewriter logic detects this function word, a flip-flop is set which will give the operator a visual indication of the Fault condition.

This will not affect further operations of the typewriter logic. The Input Fault function word is applied as a "0" to the input to 34C00. When the 04E04 circuit outputs a "0" indicating an External Function signal, the 34C00 will be fully enabled and will output a "1". This signal will set the Input Fault flip-flop (G00). The FAULT indicator will now be turned on.

The Input Fault flip-flop may be cleared by the transmission of the Master Clear function word from the Trainer. In case it is desired to clear the Input Fault flip-flop without destroying the contents of the Input or Output registers, an individual CLEAR switch is provided on the control panel directly beneath the FAULT indicator. Depressing this switch will ground the feedback line from 00F00. With all "0" inputs to 01F00, the one side will be forced to output a "1" to the zero side. This action will switch or clear the F00 flip-flop.

(2) OUTPUT MODE. - (See figures 5-49 and 5-50.) The typewriter operates in the output mode when it receives information from the Trainer and records the information in the form of a typewritten hard copy or code punched on paper tape.

When the Trainer wishes to transmit information to the typewriter, it must send the Enable Output function word along with the External Function signal to set up circuits in the typewriter logic in preparation for the reception of data word. The COMPUTER CONTROL switch must also be in the down position.

The static state of the output circuits are governed by the Output Mode flip-flop via 03000. Before the arrival of the Enable Output function word, the O00 flip-flop in the cleared state, the output to 02O00 is a "1". Double inversion is accomplished by the 02O00 and 03O00 circuits which will result in "1" outputs to the 13O10, 30O10, 20O00, and 10O00 circuits.

The "1" input to 13O10 will disable the generation of any Output Data requests 13O10 would try to generate. Therefore, communication between the Flexowriter and the Trainer is made impossible for output operations. The "1" input to 30O10 circuit is inverted to a "0" and applied as the only input to the 31O13 inverter. The inversion properties of 31O13 will now change the signal to a "1" which is applied as a continuous clearing signal to the zero sides of all flip-flops contained in the Output Data register. This clearing signal is also applied to the zero side of the Output Acknowledge flip-flop (30O00) maintaining its cleared state The remaining "1" outputs to 20O00 and 10O00 are clearing signals to both circuits which are zero sides of flip-flops contained in the output circuitry.

The Enable Output function word from the Trainer will fully enable the 34C02 circuit upon the arrival of the External Function control signal. The resulting "1" output of 34C02 is sent to time delay circuitry 40V01, where a 340 millisecond "0" output is immediately generated.

This 350 millisecond "0" pulse is inverted by the 41V01 circuit and a resulting "1" output is used to ensure the Input Mode flip-flop (I00) is cleared and will set the Output Mode flip-flop (O00). This "1" signal is also applied to the 02O00 circuit maintaining a disable to the Output circuits for the full 350 milliseconds. This delay is employed to allow the Trainer sufficient time to receive the last word during Input Mode or the Flexowriter logic to operate on the last word received during Output Mode before the mode changes from Input to Output or from Output to Input. It should be noted that identical time delay and circuitry is employed for Input as well as Output.

The setting of the Output Mode flip-flop results in a "1" output from 00O00 that is applied to the 40O00 inverter forcing its output to a "0". This ground level

is applied as the input to relay puller 40Y00, which will energize Start Motor relay K11. Power is now applied to the Flexowriter motor.

The "0" output from 01O00 is applied to the 02O00 circuit where it is combined with the output of 41Y01 which will return to a "0" output after 350 milliseconds. The "1" output from 02O00 is returned to a "0" by the 03O00 circuit which will remove the disables outlined in the preceding paragraphs.

Examining the four inputs to 13O10 shows the input from 03O00 is now a "0", and the 10O00 and 20O00 circuits, because they were held in the zero state, will also supply "0" inputs. The remaining input to fully enable 13O10 is from 15Y00. It will be considered at this time that the output of this circuit is a "0" that will satisfy the 13O10 gate, and the Output Data request is now generated as a "1" signal output from 13O10 to 13O00 of the Trainer Input/Output logic (see figure 5-14). The Flexowriter logic is now ready to accept the Output Data word from the Trainer. The Trainer will send the data word to the Flexowriter circuits as soon as it is instructed to do so by the program being executed. Meanwhile, Flexowriter logic circuits will remain in the Output mode unless the Trainer commands it to perform a different function by transmitting a new function word.

When the Trainer sends a data word, it enters the six 22C-- gates which gates correspond to data bits 2^0 to 2^5 of the Q register. Inputs are also made available to the 34C-- circuits which are the gates of the Flexowriter control (figure 5-48) when the word being received is accompanied with the External Function signal. When a one bit is present to any of the gates that gate will receive a "0" input and become partially enabled. In order for the data to be gated into the Data Output register, the Output Acknowledge must be present as an input to the gates.

The Output Acknowledge, a "1" signal, will appear as the output of 03O00 and the input to the 23C00 inverter and 31O00 and 11O00 flip-flops. The input to 23C00 will be inverted to a "0" and applied as inputs to the 22C-- gates. Those gates now fully enabled (Output Acknowledge and binary one bits) will output a "1", thereby setting their respective flip-flops. Each flip-flop of the Data Output register has a 23Y-- Punch Puller circuit and a relay associated with it (see figure 4-50). Those stages of the register that were set to the "1" state will apply a "0" input to the 23V-- circuits which, in turn, will pull in the relay for that stage. If the flip-flop was not set, the relay would not be energized. The closed contacts of the energized relays will now apply a B- voltage to position the shuffle bars in the Flexowriter mechanical translator.

Two flip-flops are also affected by the Output Acknowledge. The "1" input to 11O00 will force a "1" output from 10O00 (zero side). This "1" is applied to 13O10 which will now output a "0" dropping the Output Data Request. The "1" input of the Output Acknowledge to 31O00 will unconditionally set it to the one state. The output of 31O00 (one side) will partially enable the 22O00 gate which is used for operations that require a longer period of time to execute.

The rotation of TCC1 (Timing Cam Common one) will provide timing to the output logic circuits. TCC1 will close a set of switch contacts at 15° (±5°) and will open again at 65° (±5°). The common contact is connected to ground which, at 15°, will be applied to the input of 17Y00. The remaining input to 17Y00 is always enabled by a direct connection to ground.

When TCC1 has rotated to approximately 15°, a ground level is applied to pin 6 of 17Y00 which will fully enable the circuit. The "1" output is the input to the 30O10 inverter and the 16Y00 time delay circuit. The "1" signal will be inverted to a "0" by the 30O10 and back again to a "1" by the 31O03 which will clear the Data Output register. Meanwhile, the 16Y00 delay circuit, also with the "1" input from 30O10, will produce a "0" output to 15Y00 for approximately 75 milliseconds. For this period of time, 15Y00 will output a "1" which will be applied to 10O00, clearing that flip-flop and restoring the "0" enable to 13O10. The "1" signal of 15Y00 is also an input to 13O10 which will continue to disable that circuit until the time delay circuit has recovered. When the 75 millisecond delay period has passed, the "1" output of 16Y00 will once again allow 15Y00 to output a "0" which will now fully enable the 13O10 gate. The generation of the Output Data request occurs and is sent to the Trainer informing it that the Flexowriter is ready to receive another word. This action will occur for each word received from the Trainer. The Flexowriter will take 100 to 125 milliseconds to complete one cycle or to typeout one character.

Operations that require a longer period of execution time are given special consideration by the logic circuits. These operations are: carriage return, tab operations, and back space operations. When a number or letter is to be typed out, the action is completely controlled by the mechanical translators and a definite time element to accomplish this operation is established by the TCC1 cam. However, when a carriage return is ordered by the Trainer, a longer period of time is required by the Flexowriter to execute the command. To allow for this required time, a translator is employed to sample each code being received by the Data Output register. This translator consists of the 24O00, 24O01, and 24O02 circuits.

The 24O02 circuit will be fully enabled if a code of 45 (carriage return) is received. The "1" output of 24O02 will be inverted back to a "0" by the 23O00 inverter. This "0" signal will be combined with the output of 31O00 (Output Acknowledge received) by the 22O00. The resulting "1" output will be applied to the 21O00 circuit, clearing that flip-flop. From the "0" side (20O00) a "1" level is applied to 13O10 that will delay the generation of the Output Data Request until the operation code has been executed.

The "1" signal of 22O00 is also applied to time delay circuit 25Y00, which will immediately output a "0" that will be maintained for approximately 180 milliseconds. The "0" output will be inverted by 26Y00 which will apply a "1" as the input to 27Y00 forcing its output to a "0". The second input to 27Y00 is controlled by two switches connected in series to ground.

These switches, SCRT (carriage return and tab switch) and the SBS (back space), are normally in the closed position thereby maintaining a constant ground enable to one of the inputs of 27Y00. When the Carriage Return command is executed, the SCRT switch will open and remain open until the carriage has moved to the left margin. During this time, the ground enable is removed from pin 10 of 27Y00. The normal bias voltage of this circuit will continue to force a "0" output to the input of 20O00, even after the 180 milliseconds time period has passed. When the delay circuit recovers, or the SCRT switch closes again, whichever occurs last, the 27Y00 inputs will both be at a "0" level. The "1" output of 27Y00 will be applied to the 20O00 circuit, clearing the flip-flop. The output of 20O00, now a "0", will re-enable the 13O10 circuit and the Output Data Request is generated, requesting the next code from the Trainer.

The 24O00 circuit translating for a code of 61 (back space) and the 24O01 circuit translating for a code of 50 (tab) will, when fully enabled, initiate the same sequence of events as for a carriage return.

The Flexowriter has a built-in, automatic carriage return that must also be considered. If the operator or programmer has not programmed a carriage return at the end of a line, and the carriage has moved to the far side of the Flexowriter, the mechanics of the unit will automatically execute a carriage return without a command from the Trainer. The Flexowriter circuits do not compensate for this type of carriage return. Care should be exercised that programmed carriage returns be executed before reaching the rightmost margin. Failure to do so would possibly result in typing of characters during the execution of the carriage return. This could occur because the Flexowriter circuits, or Trainer, would not be aware of the carriage return and would continue to exchange command signals.

When the Output mode is terminated, whether from a Master Clear command, the enabling of the Input mode, or depressing the MASTER CLEAR pushbutton, sufficient time must be allocated to process the last word received from the Trainer. The time required is gained by the time delay of 350 milliseconds from the arrival of the Enable Input command and the time the input circuits are enabled. However, the Output Enable flip-flop will be cleared immediately upon receipt of the command. This action will remove the "1" input to 40O00 which would drop the "0" enable input to the 50Y00 relay puller. If this was permitted, K11 would de-energize, dropping power to the Flexowriter before the last character could be typed or punched on paper tape. To overcome this condition, the output of 10O00 is also fed as an input to 40O00. 10O00 will not be returned to a "0" output until the word presently being processed has been completed and the Output Data Request is being generated again. Therefore, 10O00 will maintain a "1" input to 40O00 keeping power supplied to the Flexowriter until the last character has been fully processed by the Flexowriter.

INPUT MODE. - (See figures 5-48 and 5-51.) The Flexowriter operates in the Input mode when it transmits information to the Trainer from tape-recorded data or by data inserted manually via the keyboard.

Before the arrival of the Enable Input command, the Input Mode flip-flop (I00) will be in the cleared state. The output from 01I00 will be a "1" input to the 02I00 which will be inverted and applied as a "0" input to 03I00. The "1" output of 03I00 will be made available to four of the input circuits. The "1" input to 20I10 from 03I00 will force a "0" output from 21I10. This "0" is inverted back to a "1" by 21I10 and applied as a continuous negative input to the zero side of all the Data Input register flip-flops (10C--) holding them in the cleared state. The "1" input to the 10I00 circuit will keep that flip-flop from being set until the Input mode becomes active. The one side of this flip-flop (11I00) will maintain a disable to the 11I10 inverter that will not allow the generation of the Input Data Request.

The "1" input to 21I01 will keep a continuous "0" input to time delay circuit 20Y20. 30Y20 requires a "1" input to trigger the delay properties of the circuit. The last output of 03I00 is the input to 12I10 whose output is now forced to a "0" output. This signal will be inverted back to a "1" by 13I10 and applied as a constant disable to the relay puller 30Y10. K8 will not be allowed to energize under these conditions.

When the Trainer is ready to receive information from the Flexowriter, regardless of whether it is from tape or the keyboard, the command to enable the Input mode must be transmitted to the Flexowriter accompanied with the External Function signal. When these signals are received, the 34C03 circuit of the Flexowriter control will be enabled and will apply a "1" input to delay circuit 40Y00. The output of 40Y00 will immediately output a "0" for a duration of the delay, 350 milliseconds. This signal is inverted back to a "1" by 41Y00 and is applied to the zero side of the Output Mode flip-flop, ensuring that the Output mode is inactive. The "1" is also applied to the Input Mode flip-flop (I00) setting it. The 02I00 circuit, now with a "0" input from 01I00, is still held disabled by 41Y00 for the duration of the delay. When the delay circuit recovers, the output of 41Y00 is returned to a "0" output that will fully enable inverter 02I00 forcing its output to a "1". A second inversion of this signal is accomplished by inverter 03I00 and will now change the disables ("1") to enables ("0") to the four circuits previously considered. Power is given to the Flexowriter by the "1" output of 00I00 applied to 40O00. The "0" of 40O00 will again energize K11 via 50Y00.

With the "1" signal removed from 12I10 and from 03I00 and replaced with a "0" and combined with the "0" output of 10I00, 12I10 is forced to a "1" output. The "1" is applied to 13I10 and inverted to a "0" that is reflected to the input of relay puller 30Y10. The "0" level input will energize the K8 relay giving power to the Reader Clutch and will turn on the Input Ready indicator. This indicator is the white light located above the keyboard on the front of the Flexowriter case, and indicates that the input circuits are ready to accept one frame of information.

The contacts of K8 that control the reader clutch power is wired in series with relay K23 located in the Flexowriter. This path, although not shown in these drawings, is still blocked until the K23 relay becomes energized. When the START READ switch is depressed on the Flexowriter, relay K23 will energize and hold. The Reader Clutch will then be controlled entirely by K8 from this point.

Assuming paper tape is being read, the START READ switch has been depressed, the typewriter will begin a cycle to read the first frame of tape. The sense pins of the reader will activate the mechanical translator of the typewriter which, in turn, will close selector switches (SS1-SS6) as well as the Selector Common Contact (SSC). There is one switch associated with each bit of the Data Input register or relays K9, 10, 12 through 15. Those switches that close, as a result of the punch positions of the tape, will provide a path from B- through its associated relays to a B+ source. The relays that are energized will close their normally open contacts and apply ground input to the 12Y-- circuits. Each 12Y-- that received a ground input will not output a "1", and will set its associated flip-flops. Thus, the first frame of data, now located in the Data Input register, is now applied to the gates of the Q register.

The outputs of the zero side from each flip-flop is now examined by the 21I00 circuit. If any flip-flop or a combination of flip-flops were set, as they will be upon reading a frame from tape, any number of inputs to 21I00 will be "1's". When the output of 21I00 is a "0", it indicates that the Data Input register is full and ready to transmit the data to the Q register in the Trainer. The "0" from 21I00 is combined with "0's" from 03I00 (Input Mode Active) and 20O00 (long character flip-flop) to force a "1" output from 21I01. The "1" is the input to 30Y20 that will now output a "0" signal for 18 milliseconds. This "0" is inverted by 31Y20 which will set the Output Data Request flip-flop (11I00) but will continue to disable the transmission of the request for the duration of the time delay. Meantime, the setting of the Output Data request flip-flop will force the zero side to output a "1" to 13I10. After inversion of the signal by 12I10 and 13I10, a "1" is applied to the relay puller which will now de-energize K8. This will remove power to the Reader Clutch and the Ready indicator will be turned off.

When the delay circuit is recovered, the output of 31Y20 will again enable 11I10. Combined with the "0" output from the Output Data Request flip-flop (11I00), 11I00 inverter is permitted to output a "1" which is the control signal to the Trainer indicating data is ready to be gated into the Q register.

The input circuits will remain in this condition until the Trainer senses the Input Data request line and executes an Input instruction. This instruction will gate the information held by the Input Data register into the Q register and generate the Input Acknowledge signal. This signal will appear as a "1" input from 03I00 to the 20Y10 delay circuit. The 20Y10 circuit will immediately output a "0" level for 80 milliseconds which will be inverted to a "1" by 21Y10, to a "0" by 20I10, and back to a "1" by the 21I10 circuit. The output of 21I10 will clear the Input Data register and clear the Input Data request flip-flop. With this clearing action, the output of 10I00 will again become a "0" to the input of 12I10. Combined with the "0" output of 03I00, 12I10 will produce a "1" input to 13I10 which will return the "0" enable to the input of relay puller 30Y10. Once again, K8 is allowed to energize and supply power to the Reader Clutch, as well as lighting the Input Ready indicator. Thus, one more frame is read from tape and the entire cycle is repeated.

4-7. OPERATION OF THE MEMORY SECTION.

a. GENERAL. - The memory section of the Trainer is a current operated, magnetic core memory. It uses the permanent magnetic properties of ferrite cores to store binary information. The four principle parts of the memory section are:

1) Memory Stack - contains the storage elements of the system.

2) Address selection circuits use - to select the specified memory location.

3) Data Control Circuits - control the flow of information into and out of memory.

4) Timing Control Circuits - establish the timing relationships during cycles of operation.

Chassis three contains the magnetic core storage system. This system has high-speed, random access, and non-volatile characteristics; i.e., the speed of operation is compatible with the other Trainer sections; data may be referenced in a non-sequential manner; and the system retains its data when power is removed from the Trainer. There is a total of 512_{10} locations (registers) for the storage of 15-bit words. Every storage location is assigned a separate address (000_8 through 777_8).

When a specific storage location in memory is referenced, the S register contains a 9-bit address word that specifies one of 512 storage locations. Data transmission into or out of the selected storage location (register) is channeled through the Z register.

Basic memory cycle time (the time required for one memory reference) is 8.0 microseconds. After a given function initiates memory, approximately 2.5 microseconds will elapse before the delivery of data from storage (readout time). All timing relationships in the memory section are established by the Memory Timing Chain.

(1) MEMORY BLOCK DIAGRAM. - Figure 4-73 shows the memory section in simple block diagram form. To form a sequence of events, the control section will place into the S register the specific address of the location in memory to be referenced. The contents of S is immediately broken down into five groups as shown. The results of this translation will select one core on each of 15 memory boards to make up the 15-bit word. The timing chain will be initiated by the main timing chain of the control section. The first thing to be generated from the memory timing is the Address enable which will select the memory location to be referenced. This is followed by a Read enable which will allow the Read generator to produce the necessary current to read from the selected cores. The output of the cores, amplified by sense amplifiers, are strobed by the timing chain at the proper time and gated into the Z register. The information contained by cores at the selected address location is now destroyed, but will be restored by the Write cycle of the timing chain.

The output of the Z register is made available to the inhibit generators. These circuits will inhibit the writing of a binary one in those locations where Z is equal to a binary zero. When the timing chain enables the Write generator, the data contained in the Z register, by way of the inhibit generators, will restore the data in the memory location. This automatic Read/Write cycle is part of the timing chain.

b. CORE THEORY. - Before the memory section can be examined, the theory of the ferrite magnetic core must be understood. A typical magnetic core is shown in figure 4-74. It measures 0.050 inches outside diameter, 0.030 inches inside diameter, and is 0.015 inches thick. The ferrite core is magnetized by the field produced by a current flowing in a wire that is threaded through the core. It retains a large amount of this induced flux when the current is removed. Flux lines are clockwise, or counterclockwise, around the core, depending upon the direction of the current (see figure 4-75).

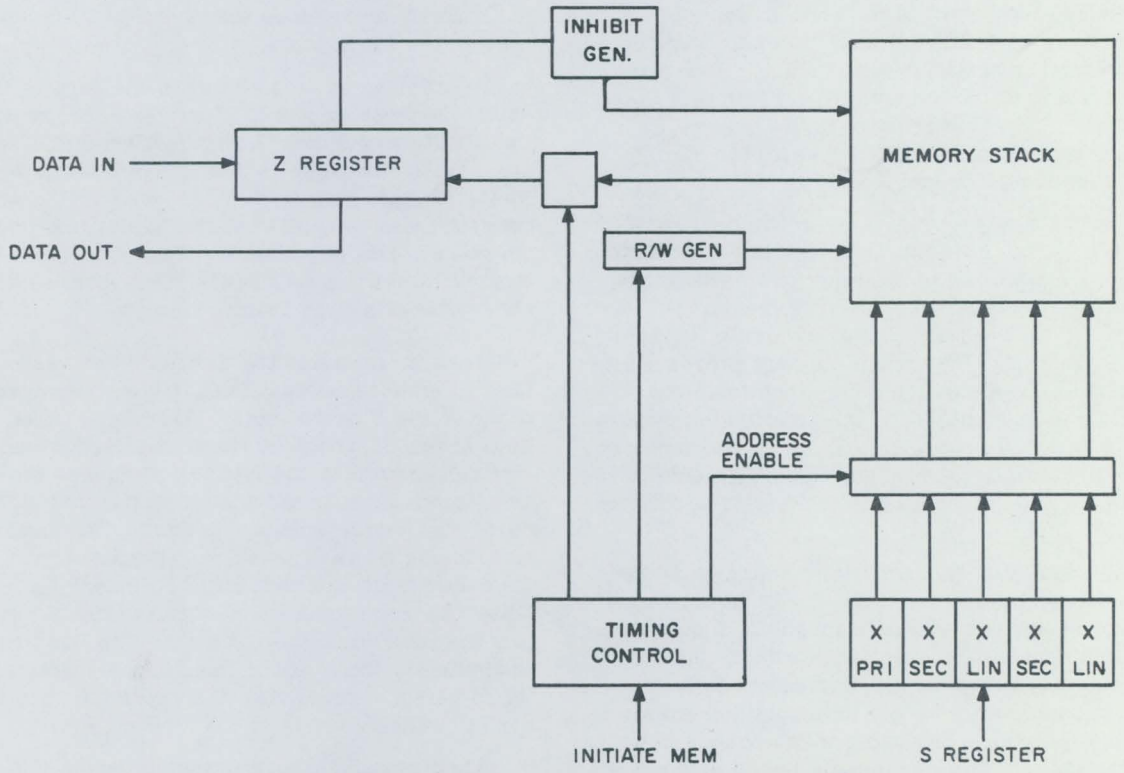


Figure 4-73. Memory Block Diagram

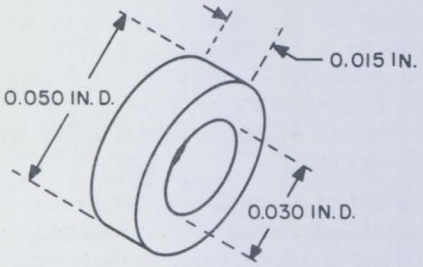


Figure 4-74. Magnetic Core

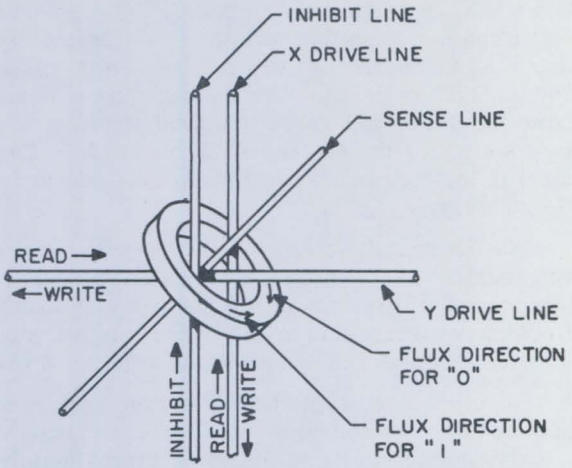


Figure 4-75. A Typical Magnetic Core Inhibit Line X Oriented

These two unique states are designated "0" or "1", depending on the orientation of the core with respect to the wiring. The time required to switch the core from one state to the other is approximately 1.2 microsecond with the current pulse used (approximately 400 milliamperes for two microseconds).

Again, drive currents of one-half current, transmitted simultaneously along two wires, X and Y, will reverse the magnetic field in the core at the point of crossing (see figure 4-76). Thus, information can still be restored or retained by the core. Reversing both half currents again changes the direction of the magnetic field. This reversal of fields induces a signal which is sensed on a third wire. The signal, or pulse, is detected and amplified.

The memory system uses many such cores (see figure 4-76). Drive current, flowing along two wires, may change the direction of the magnetic field of the core at the intersection, but will leave the other cores undisturbed. The Trainer uses this memory principle to store information. The magnetic core stores a bit of information by remembering the direction of the flow of the drive current. The information bit remembered by the magnetic core is read out by the sense wire. The principle of the magnetic core is simple and dependable and permits an extremely compact memory unit.

To have a bistable, fast switching memory device, it is necessary to have a material that is easily magnetized and has low resistance to magnetic flux. By applying an alternating magnetizing force to this material, a hysteresis loop can be obtained. This is a very square loop compared to a loop obtained from a non-ferrous material. Switching of the square loop ferrite material is faster, causing a larger induced pulse in the sense line. This allows a stored bit of information to be more easily distinguished from normal circuit noise than could be obtained from a non-ferrous material.

The first step in writing a "1" when the core is in the "0" state is to force the core into its "0" state of magnetization. This is done by applying pulses to both the X and Y lines which clears the core. The Clear pulse is followed by a write pulse applied to the X and Y drive lines which changes the state of the core to the "1" state.

In order to write a "0" when the core is in the "1" state, the first step is to clear the core, or drive it to its "0" state of magnetization. During the second step, the Write pulse is applied to the X and Y drive lines. In this step we want to write a "0", so a half-current pulse is applied to the Inhibit line in a direction opposite to that of either the X or Y lines. This effectively cancels one-half of the Write current, and the core will not switch to the "1" state.

Reading a core in the "1" state and restoring it to the "1" state is accomplished by applying a read pulse to the X and Y drive lines. This Read pulse is identical to the Clear pulse which was used during the Write operation. The Read pulse forces the cores to the "0" state, and the Sense line, passing through the core, will detect a change in flux. This output is used only during a read operation. During a write operation, the Sense lines are shut off. Following the Read pulse, the core must be restored to the "1" state so that the information is not lost. The Restore pulse is applied to the X and Y lines and is identical to the Write pulse. This pulse will drive the core back to the "1" state.

When reading a core in the "0" state and restoring it to the "0" state, the Read pulse is applied to the X

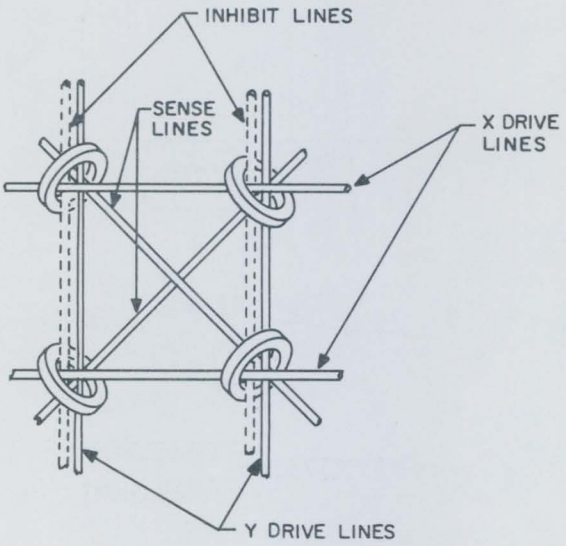


Figure 4-76. Intersection of Lines Through Magnetic Cores

and Y drive lines. Because the state of the core is already in the "0" state, very little flux change will be detected by the Sense line. The difference in the amount of flux change, in comparison to reading a "1", indicates that a "0" has been read from the core. The Read pulse is followed by a write pulse which is again applied to the X and Y drive lines. In this situation a "0" is to be restored, therefore, a half-current pulse is applied to the Inhibit line to cancel out half of the Write pulse. The core will now remain in the "0" state.

Control of the X and Y drive current in the Trainer is accomplished by the memory timing chain. The Read generator will provide approximately 400 milliamps of current on both the X and Y drive lines in such a direction that all cores for the selection address location will be driven into the "0" state. Those cores holding a "1" will switch to the "0" state, inducing a current pulse into the sense winding while those cores holding "0" will induce a minor pulse. A "1" pulse will have the amplitude of approximately 55 millivolts, while those cores holding a zero will induce a pulse of approximately 10 millivolts into the sense lines. These pulses are amplified by the sense amplifiers. The Strobe pulse, generated by the timing chain, will probe the output of the sense amplifiers at the correct time to gate information pulses only into the Z register. The output of the Z register will provide the inputs to the inhibit generators to determine which cores will remain "0" and which cores will be restored to "1". If the Z register for a given stage is holding a "0", the inhibit generator will be enabled. Those stages of Z that are holding "1" will disable the Inhibit generator and current will not be permitted to flow from these circuits.

When the command to write is generated, the X and Y drive lines are supplied current pulses that will attempt to drive all cores for the selection location to a "1" state. Those cores receiving an Inhibit current pulse will not be able to switch their states because of the cancelling current pulse of either the X or Y drive pulses. A core must receive the full current pulse of both X and Y to switch.

The capacity of a memory system depends primarily upon the number of cores a memory board contains. The memory system of the Trainer is capable of storing 512_{10} 15-bit words; therefore, its memory stack contains 7,680 individual cores.

(1) THE HYSTERESIS LOOP. - A more detailed explanation of the core theory can be offered by examining the hysteresis loop which is characteristic of the magnetic states a core can obtain. In most electronic devices such as transformers, the hysteresis loop of the material utilized in its core represents a loss of the transformer. Therefore, in the construction of a transformer, the hysteresis loop is kept to a minimum, thereby making the transformer efficient.

The hysteresis loop for ferrite material employed in the construction of memory cores has a near square loop as shown in figure 4-77. This loop shows the magnetic states to which ferrite material can be forced and held after the magnetizing force has been removed. As shown, this material then, can be driven into two separate states which gives it the properties of a bistable element.

The X and Y drive lines that are passed through the center of each core comprise the magnetizing force in the magnetic core system. These lines, when pulsed with a current, will develop a magnetic field around the wire which will affect the magnetic state of the core. When this current is increased, thereby increasing the magnetic net around the wire, the magnetism of the core is changed to correspond with the magnetizing force (current through the wire). Because each core has both X and Y drive lines passing through it, the selection of an individual core can be determined where these lines intersect. When each of the drive lines is supplied with a half-current pulse, all cores through which these lines pass will feel the half-current pulse, but will not be able to switch their states. The only core to feel the full current pulse, as a result of the half-pulses applied to X and Y, is where these lines intersect. The core at this location will change its state to correspond with the magnetizing force.

The state of magnetization of a core is shown on the hysteresis diagram which plots magnetic flux density in gauss (B) as a function of the field (induced by the current) in oersteds (H).

The diagram shown in figure 4-77 assumes that the core is already is some state of magnetization,

such as A. If a current flow with a direction (+) that produces a field (H) of a given magnitude (m), (i.e., +Hm), is applied to the Drive line, the flux density increases to saturation as indicated by +Bs. When the current is removed, the flux density retained by the core drops slightly to the level indicated as the "0" state. Another pulse of +Hm would now merely shift the core to +Bs again, and after the pulse is removed, the core would return to +Br.

If a current pulse of the same magnitude, but in the opposite direction (-Hm) is applied to the Drive line, the flux density shifts along the curve to -Bs causing a reversal of the flux density in the core. When the current pulse is removed, the residual flux density is at -Br, the state arbitrarily designated as the "1" state.

Any change in the flux of a core induces a voltage in all windings passing through the core. Hence, the induced voltage on the sense winding is sampled to see if the core switches, with +Hm applied, from -Br to +Bs. If a large induced voltage is sensed (over 50 millivolts), the core was in the "1" state and had been switched from -Br to +Bs. Because the content of the core is determined in this manner, the current pulse corresponding to +Hm is called a Read pulse. A memory utilizing this type of magnetic core storage element is referred to as a Destructive Readout memory. Therefore, a restore function is necessary to return the core to its original state.

When a core is in the "0" state, it is possible to enter a "1" into the core by applying a pulse corresponding to -Hm, or to leave the "0" by not applying the pulse. Because the data circuits and addressing circuits are separated, the -Hm is applied unconditionally. When entering a "0", the field is partially cancelled by an inhibit pulse on the Inhibit line. An inhibit pulse is the equivalent of one-half (+Hm), a half-read pulse. A write pulse of net half-amplitude, one-half (-Hm), is not sufficient to switch the core to the "1" state.

The operation of memory depends on the ability of each core to distinguish between two current levels on its read/write windings (R/W drive lines). Each core in a plane is linked by four windings. Two of these windings, the X and Y drive lines, determine the address of the core. To operate on the selected core, half-amplitude pulses are supplied to each selected drive line so the core at the intersection of the two selected drive lines is the only core that receives a net field, or full amplitude pulse, of Hm. All other cores on the two selected drive lines receive only the half-pulse field associated with one drive line.

As seen in figure 4-77, the coercive force, Hc, is the field required to switch the core. The drive currents have been selected so that $\frac{Hm}{2}$ is less than Hc, and as a result, is insufficient to switch the core. But the sum of the two drive currents, $\frac{Hm}{2} + \frac{Hm}{2} = Hm$ is greater than Hc and switches the core in just over one microsecond.

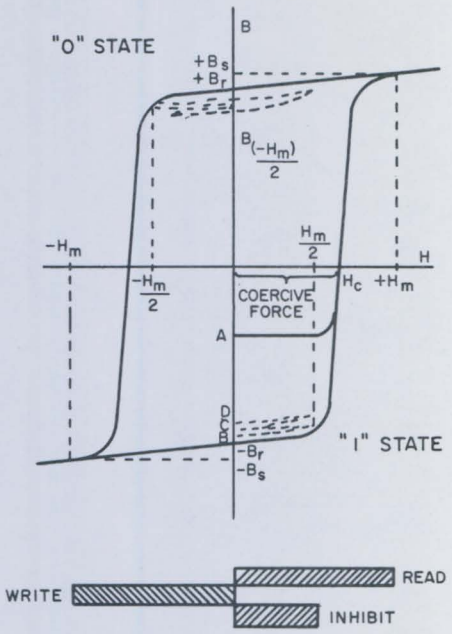


Figure 4-77. Idealized Hysteresis Loop of a Ferrite Core

When a core receives a half-current pulse, the field induces a change in the flux density of the core. Assuming that the core is in the "1" state at -Br, a half-read pulse causes the flux to shift along the hysteresis loop to the point limited by $\frac{Hm}{2}$ and then return to a slightly lower remanent value, such as point B. Since the core is now operating on a slightly smaller loop, further half pulses again reduce the remanent flux. This effect soon reaches a limit as at point D. When the core is in the "0" state (+Br), half-write pulses produce a similar effect.

The shift in flux, caused by half-current pulses, induces a small voltage on the sense winding. The amplitude of this voltage is a function of the squareness of the hysteresis loop. The squareness ratio, Rs, is defined as the ratio of the flux density value at $\frac{Hm}{2}$ to that at +Hm. Values of Rs range from a practical limit of 0.7 to an ideal limit of 1.0. A typical value for the cores is about 0.9. This and other operating margins, such as slant, Hc, and Bs are ensured by grading. A core with a low Rs, or a greater slant, has a greater shift in flux for a given half-current pulse. The small voltage, induced on the sense winding as the flux level travels along the "knee" of the hysteresis loop to a slightly lower remanent value, is a significant source of noise that tends to obscure the desired signal. This effect can be partially eliminated by strobing the core output during the maximum "1" time (see figure 4-78).

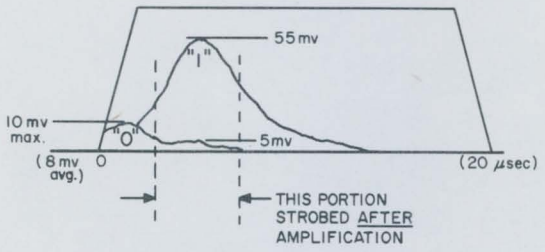


Figure 4-78. Typical Magnetic Core Output

c. MEMORY STACK AND MEMORY BOARDS. - The storage elements of the memory section are contained in one memory stack. The stack contains 15 memory boards with two end boards to which connections are made to supply the necessary drive currents to each of the memory boards. The memory boards are interconnected by spring clips that are used to connect each of the boards together.

The memory board is the basic unit of the memory stack. Each of the memory boards contain 512 magnetic cores located at the intersection of the horizontal and vertical conductors (X and Y drive lines). Because each core can store one binary bit, one memory board is all that is required to store one bit position for all possible addresses. The memory board employed in the Trainer is shown in figure 4-79. The X and Y drive lines terminate at tabs along the edges of the board. On the top surface of each board adjacent tabs are connected to alternate drive lines. An identical set of tabs is located on the lower surface of the board. These tabs lie directly under corresponding tabs on the top surface, but are electrically insulated from them. The bottom tabs are connected to drive lines that are not connected to the top tabs. (Refer to figures 4-80 and 4-81 for a simplified view of drive line connections.) A drive line connected to an upper tab on one edge of a board terminates on a lower tab at the opposite edge.

As can be seen by figure 4-79, a memory board is divided into four quadrants, two of which are employed by the Trainer's memory. A quadrant on the boards contains a 16 x 16 array of cores or a total of 256 magnetic cores. In the larger memory systems, each quadrant of a memory board has its own sense and inhibit windings associated with it. However, the memory boards employed by the Trainer are connected in such a manner that both quadrants are controlled by one inhibit generator and the output is fed into one sense amplifier. These connections are made through the solder terminals at the corners of the board.

(1) END BOARDS. - There are two end boards associated with the memory stack. These boards are identified as A1 and A2. The A1 end board is a printed circuit board that connects the driver elements of the memory stack to the driving elements on the memory chassis. It connects the X and Y drive lines on the memory boards to the X and Y selector transformer secondaries. These connections are located on the top side of the A1 board (see figure 4-82.). Terminals 1, 2, 5, 6, 7, 8, 11, and 12 are connected to the X transformer secondaries while terminals 3, 4, 9, and 10 are connected to the X transformer secondaries. Each connection made to the terminals of A1 will be expanded to drive four lines of each of the memory boards. For example, if the driving element connected to terminal one is selected, the printed circuit of End Board A1 will in turn drive four lines in the X coordinates.

End Board A2 has 48 terminals which are connected to line selectors. That is, for each of the lines that can be driven, 32X and 16Y, the individual line is terminated on the A2 end board. The line selector will select one of the driver lines and complete the path for current for that line only. Figure 4-83 shows the A2 end board and terminal connections. The electrical connections to A2 will be further identified in the following paragraphs.

d. S REGISTER AND TRANSLATOR. - Any Trainer function requiring access to memory must first transmit the address of the memory location into S. During the memory reference, outputs from the S translator enable the proper X and Y drive lines. The S register and translator are shown in figures 5-36, 5-37, and 5-39.

The clearing and loading of the S register is controlled by the main timing chain and the sequence enable applied to the timing chain. When the command is generated, the output from 10S00 is forced to a "0". This signal is an enable input to 11S00 and 11S05. At phase four, both inverters will be enabled to clear the register. On the following phase one, the input gates are enabled to load S. The inputs to S are from the P register and the Arithmetic adder. During the A sequence the address of the next instruction to be executed is transmitted to S (P → S). The input from the Arithmetic adder is the result of the B register modification operations, to site one example. For all of the above inputs, the outputs from S to the translator specify a particular memory location from which data is either read out or stored. There is also an output from S to the S adder which was discussed during the

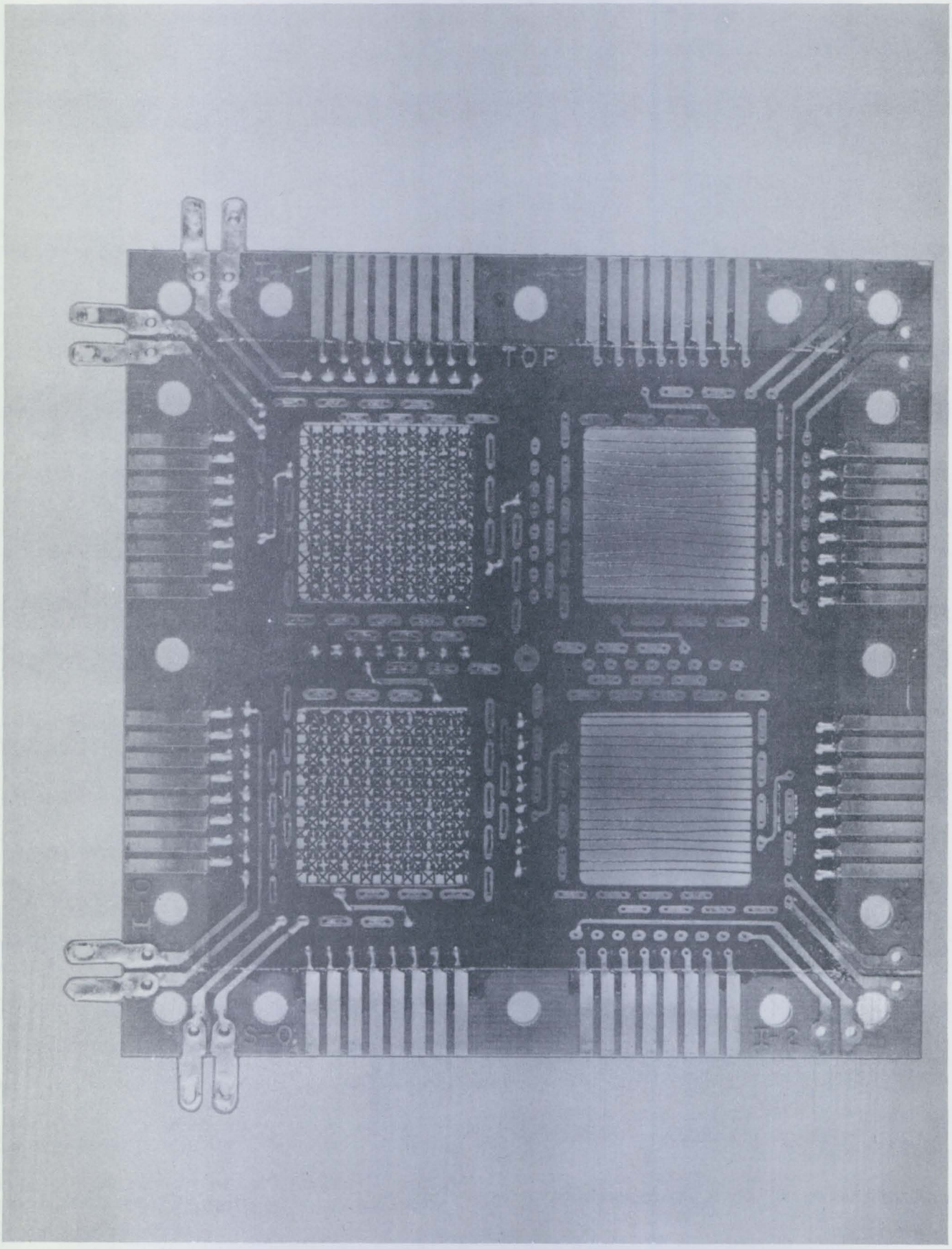


Figure 4-79. Memory Board

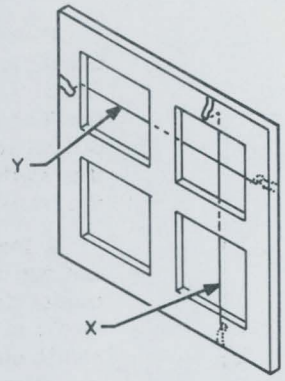


Figure 4-80. Drive Line Layout

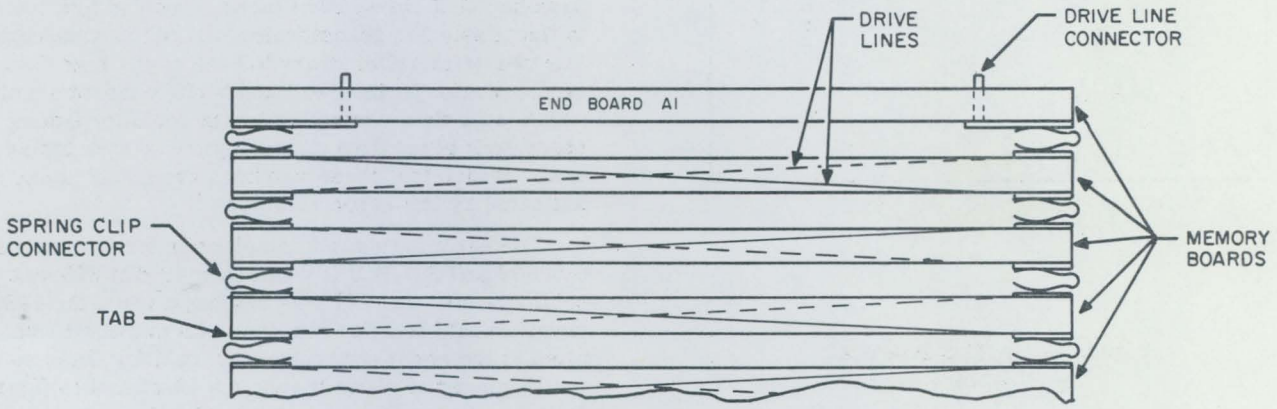


Figure 4-81. Memory Board Connections

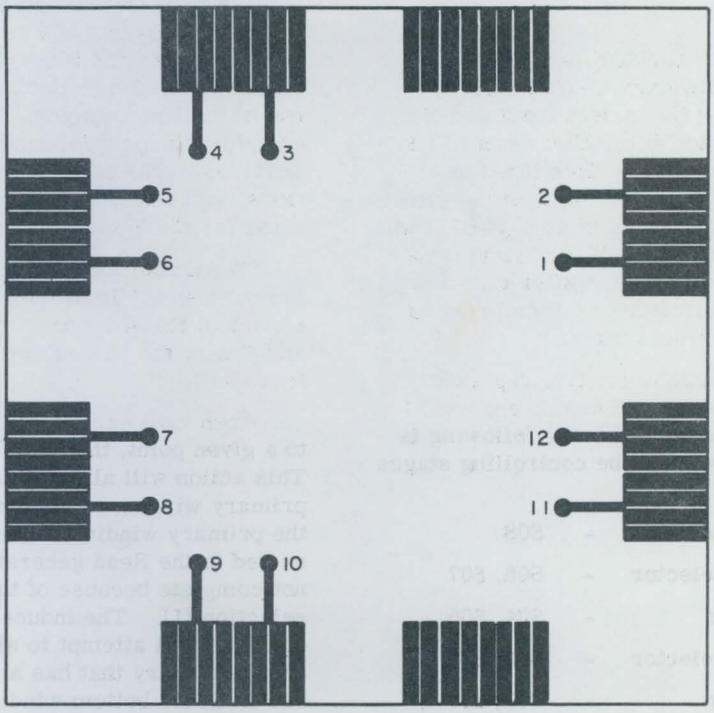


Figure 4-82. End Board A1

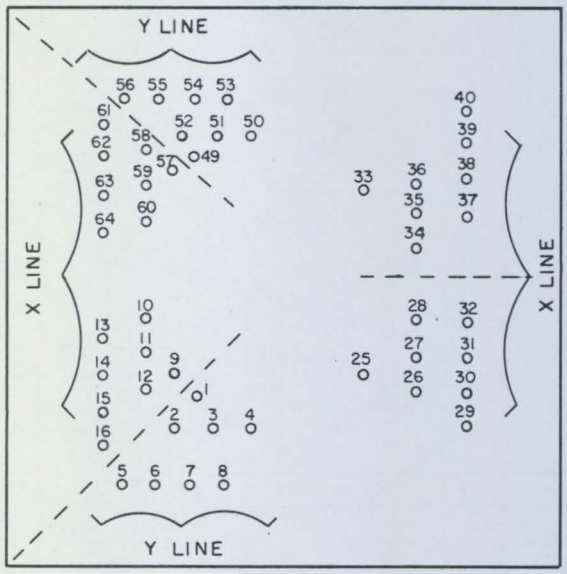


Figure 4-83. End Board A2

control section of this manual. During normal operations, this transmission returns the address incremented by one to the P register in preparation for the next A sequence which will read the next sequential instruction.

(1) ADDRESS TRANSLATION. - The S translator translates the outputs from the register to specify the unique enables that select the X and Y drive lines. The translator is enabled when L11 in the memory timing chain is set. This flip-flop remains set for both the Read and the Write sequences. The "1" output from the zero side of L11 (00L11) will be inverted to a "0" by the 40S-- circuit and is applied as an enable input to the translation circuits. The outputs from the translator are distributed to the memory circuits of chassis three.

The S translator is divided into five separate translators, each with a distinct function and each controlled by specific states of S. The following is a listing of the translations and the controlling stages of S.

1)	X primary selector	-	S08
2)	X secondary selector	-	S06, S07
3)	X line selector	-	S04, S05
4)	Y secondary selector	-	S03, S02
5)	Y line selector	-	S01, S00

Based on this breakdown, the five outputs of the translator will select a primary of a transformer (X only), the secondary and finally the line for both the X and Y coordinates. Figure 4-84 shows a simplified schematic of the X drive line translation. The switches at the bottom of the figure represent the translator output and the associated circuitry.

A look at the transformers will show that there are two primaries and four secondaries. The selection of the transformer is made by the Primary selector of the S translator (S08). In this example, the eighth bit position of the S register will select the top transformer if it is a zero and the bottom transformer if it is a one. The selection of the transformer itself will eliminate one-half of the possible lines that could be selected by the S register. That is, of the possible 32 X lines, 16 of these lines that are connected to the unselected transformer will not be considered. The two primary windings to each of the transformers are connected to the Read and Write current generators which, at a time controlled by the memory timing chain, will generate a current pulse to both transformers. Notice that these windings are out of phase as indicated by the associated dot.

The secondary winding selection for X is made from bits S06 and S07. If it was assumed that S06 was holding a one, and S07 was holding a zero, the 44S01 circuit would be fully enabled. Its output will be applied to the necessary circuitry that, for this example, would appear to close the switch identified in figure 4-84 as one for the secondary selection. Notice that the closing of this switch will attempt to complete the path for current not only to the selected transformer, but also the unselected transformer.

The final translation of X is the line selection accomplished by bits S04 and S05. Assuming S04 to be holding a zero and S05 to be holding a one, the 43S02 circuit of the S translator will effectively close switch two of the line selector. This switch will attempt to complete the path for current for line two of all secondaries. The same type of translations and selections, with exception of the primary selection, is also made for the Y line and secondaries.

To summarize up to this point, the primary selection will select 16 of the possible 32 X lines. The selection of the secondary will select four of the 16 lines, and finally the line selection will select one of the four lines.

When the memory timing chain has been advanced to a given point, the Read generator will be turned on. This action will allow current to flow through the top primary winding of the top transformer. Although the primary winding of the bottom transformer is connected to the Read generator, the path for current is not complete because of the open switch of the primary selection (1). The induced voltage into the four secondaries will attempt to allow current to flow, but the only secondary that has a path completed is the second from the bottom winding through secondary

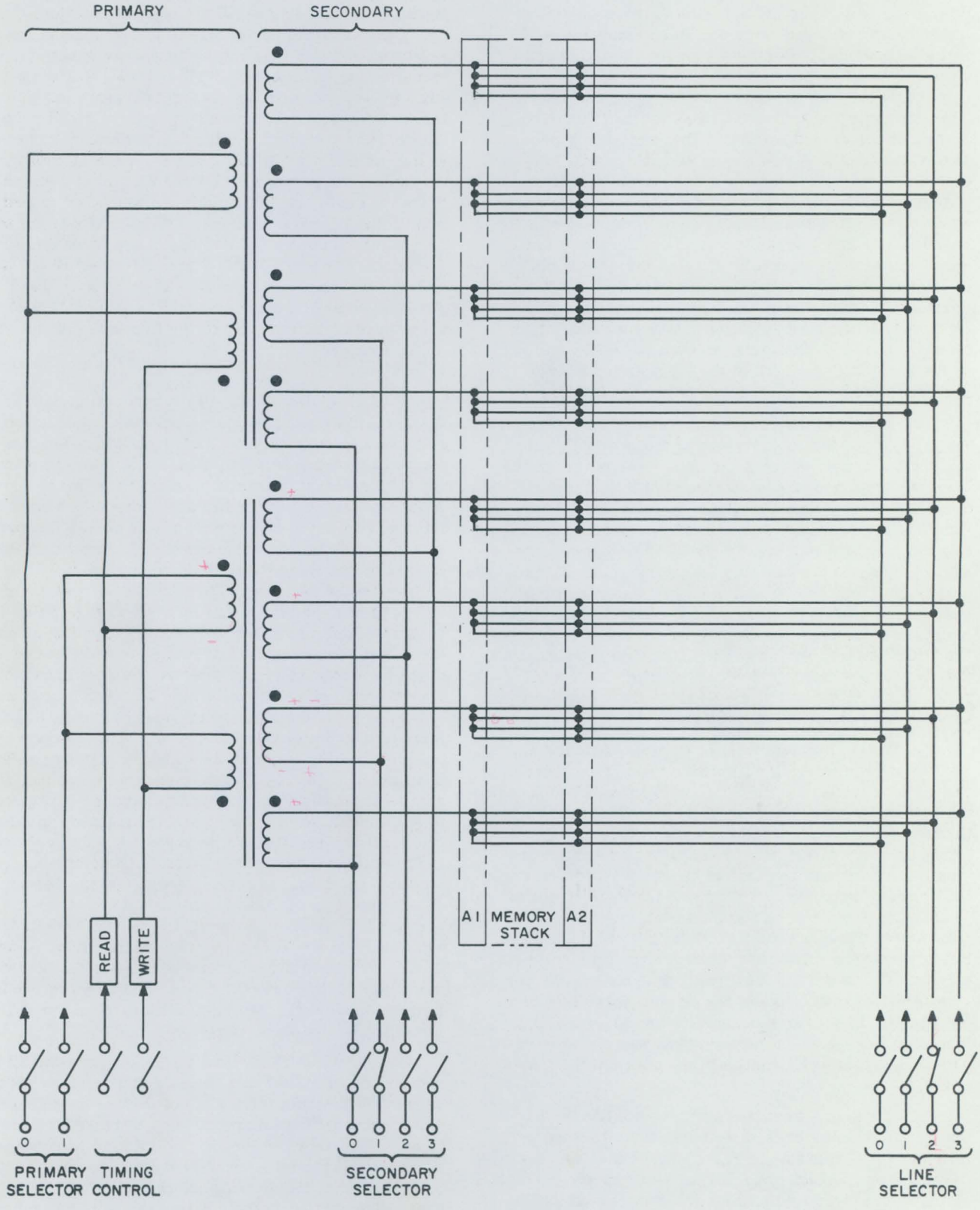


Figure 4-84. Simplified Schematic of X Drive Line Translation

selector switch one. The other side of the secondary winding is connected to a terminal on the A1 End Board which will expand the secondary input to four lines. Each of these lines will pass through the 16 memory boards and terminate on the A2 end board as individual line connections. It is to the A2 end board that the line selection is connected and made. This example has selected Line Selector two represented by the closed switch. Therefore, only one line of the four selected by the secondary will have a completed path for current to flow through. Thus, only one line in the X coordinates has been selected and current allowed to flow. When combined with the Y secondary and line selections, accomplished in the same manner, the magnetic core at the intersection of the two drive lines will be forced to a "0". If the core was holding a one, the collapse of the magnetic field will induce a pulse into the sense winding of the memory board. This pulse is then amplified and sent to the Z register. Each of the 15 memory boards are associated with each stage of the Z register.

As a result of the reading operation, all cores that were loaded at the intersection of the X and Y drive lines were driven to the zero state. As a part of the Memory sequence, the restoring of this information or data is done automatically by a write operation. The data that was just read out of the selected memory location and now held by the Z register will generate outputs to the Inhibit generators. For those stages of Z that are holding a binary one, the inhibit generators are disabled or turned off. However, those stages of Z that are holding a binary zero will enable the Inhibit generator for the memory board associated with its bit position. The current from the Inhibit generator will cancel out one-half of the write current keeping the core in the zero state.

The Write command will be generated by the memory timing chain which will enable the Write generator. The path provided for the write current will be through the bottom primary winding of the top transformer back through switch zero of the primary selector. The lower primary of the bottom transformer cannot provide a current path because of the open switch (1) of the primary selector. The induced voltage is out of phase with the previous read current, so current will now flow in the opposite direction as it did during the Read operation. However, the same path is still provided as it was during the Read cycle; that is, through line selector switch two to the A2 end board terminal, through the 15 memory boards, to the A1 end board. The secondary winding path through switch one (secondary selector) will complete the path for the current flow.

This then, is a simplified representation of the manner in which a core is selected from the translation of the S register. The switches that represent the primary, secondary, and line sections are special circuits employed by the memory section and will be discussed in detail in later paragraphs.

The discussion thus far has been directed to the selection of the X coordinates only. The selection of the Y coordinates is accomplished in the same manner except that there is no selection of the Y primary. Because of the capacity of the memory section, only one Y transformer is required thereby making the selection of the Y primary unnecessary since this one transformer is enabled for all address locations. Figure 4-84 can also be used to represent the Y selections for both the secondary section and the line section, considering only the top transformer. Reference is again made to figure 4-79, the memory plane. As shown, the number of Y lines that must receive drive currents are 16 (eight lines connected to each side of the board). The connections made to the memory boards from the A1 end board are from the four secondary windings of the Y transformer. Each secondary, as in the case for X, will drive four lines. The line selection connected to End Board A2 will complete the path for one of the four lines of the selected secondary.

e. DATA CONTROL. - Communication between the Z register and the addressed memory location is regulated by data control. Information is detected in a given bit of the selected memory location by sampling the remanent magnetization of the selected core. A net, full amplitude, read pulse, impressed on the intersecting drive lines, attempts to switch the core to a "0". If the core contained a "1", the flux reversal, caused by the Read pulse, induces a relatively large voltage on the sense windings. When a read memory reference is specified, this induced voltage pulse is amplified and strobed into the corresponding stage of the Z register where it is retained for future use. Since the contents of the selected memory location are strobed unconditionally into Z during a read memory reference, the Z register must be cleared at the start of every memory reference. This is done by the main timing chain at the time the memory reference is initiated. During a write memory reference, the data is loaded into the previously cleared Z register to subsequent storage in a selected memory location. The Write pulse attempts to write a one in every core in the selected memory location. The contents of corresponding stages of the Z register are used to determine if an inhibit is generated for a given bit to prevent the one from being written. The output from each stage of the Z register drives the inhibit winding in the corresponding memory board. If the content of any stage of the Z register is a zero an inhibit pulse is developed on the selected memory board to nullify the effect of the Write pulse.

Data control, being involved in the transfer of information out of and into the selected memory location, serves its function (memory reference) in any communication between memory and the other Trainer sections. A memory reference is initiated to sample a word (read), to store a word (write), or to perform a combination of these functions (read/write). The same cycle of operation is used within memory for each of the three - a read pulse followed by a write pulse. The difference is in the manner in which the Z register is used.

(1) Z REGISTER. - (See figures 5-40 through 5-42.) The Z register functions as a buffer between the memory section and the arithmetic and control sections of the Trainer. All information, going to or coming from memory, passes through this register. The Z register is a 15-bit register that functions at the bit plane level; specific stages in Z control specific planes (boards) in memory. For some commands the Z register is divided logically into the upper six bits and lower nine bits. The division into Z_U and Z_L is controlled by the commands generated by the timing chains.

(a) Z REGISTER INPUTS. - All stages of the Z register have inputs from memory and from the D register. Inputs from the P register are also made available to the lower nine bits of the Z register. The inputs from memory are controlled by the Strobe pulse that is initiated by the memory timing chain. The inputs from the D and P registers are controlled by command enables generated by the particular section.

Each stage in Z is associated with a sense winding in the corresponding bit plane. The sense winding is the input to a sense amplifier whose output (62Y--) is combined with the Strobe pulse and the Z_L or Z_U enable. If the three signals are "0" simultaneously, the corresponding stage in Z is set. The input from memory to Z is from 63Y-- term circuits.

The input to Z from the D register can be a half-word (lower nine bits) or a full word, depending on the instruction being executed. The command enables are generated by the main timing chain under the influence of the sequence enables. The input to Z from the P register is used during a Return Jump instruction to store P in memory lower.

Another input to the Z register is from the S + 0, 1 adder. This transmission path is used only during shift, multiply, and divide instructions. During these instructions, the Z register holds the shift count which is continuously decremented each time a shift is performed. The decrementing action is accomplished by transmitting the complement of the lower four bits of the Z register to S. Through the Adder properties, one is added to the contents of the S register and the complement of the Adder returned to the Z register. The results of these transmissions and addition will subtract one from the initial contents of the Z register lower four bits. (Details of this action were discussed in the control section.) The Z register lower four bits are continuously examined to determine if Z_{L4} is equal to zero. This is done by the 30Z00 circuit. The inputs to this circuit are from the zero sides of the first four flip-flops of the Z register. When all inputs are "0's" the output of the 30Z00 circuit will be a "1", which will be applied as inputs to the 63N80 and 94H10 circuits, indicating the shift count is zero and the instruction is completed.

(b) Z REGISTER OUTPUTS. - The Z register has outputs available to the control, arithmetic, and memory sections.

During the A sequence, the outputs from the one side of the flip-flop of stages Z09 through Z14 are made available to the U register. It is from these stages that the instructions (Function code, j, k and b designators) are obtained. The remaining 9 bits of the instruction are transmitted to the X register via the zero side of the Z register (X is set to all ones during the Clear command and cleared by the data bits during transmission times). During the A sequence, only the lower nine bits are permitted to enter the X register. During the remaining sequences, the transmissions to the X register could be the operand that was addressed as part of the instruction execution or the B register located at address 000.

Associated with each stage of the Z register are the 21Z-- circuits. These circuits will be partially enabled by the outputs of the zero side of each flip-flop of Z. These circuits are responsible for the enabling or disabling of the inhibit generators in the memory section.

If a given stage of the Z register is holding a binary zero, the output from the zero side of that flip-flop will apply a "0" input to the associated 21Z-- circuit. However, if the flip-flop is holding a binary one, the output from the zero side will be a "1" and will disable the 21Z-- circuit. The remaining input to each of these circuits will originate from the memory timing chain (00L15). When the timing chain has been advanced to the point that the Write cycle is about to start, the Inhibit command will be generated. This command will appear as a "1" input from 00L15 to the 20Z-- circuits which will invert the signal to a "0" and apply it to each of the 21Z-- circuits. Those stages of Z holding partial enables to these circuits will be forced to a "1" output. However, the circuits that are disabled by the contents of the Z register will continue to output a "0" level to the 60Y-- inhibit generators. The stages that produce a "1" output to the 60Y-- circuits will turn on the Inhibit generator prior to the Write cycle, thereby cancelling one-half of the write current. It is these cores that will remain in the "0" state. For those 60Y-- circuits that received a "0" input, the Inhibit generator was not allowed to turn on and the memory board associated with that stage will receive the full Write pulse without the inhibit current present. These cores will be returned to the "1" state.

f. TIMING CONTROL . - The generation of control signals for the operation of the memory section must occur in a fixed time relationship. This timing relationship is controlled by the memory timing chain (see figure 5-43). The timing chain provides enable signals for data transmission from memory to Z (Read cycle), and for the data transmission from Z to memory (Write cycle).

The basic memory cycle time is 8.0 microseconds. This is the minimum time between consecutive memory references. When memory is initiated, the output from 14L11 is forced to a "0" by the 61N10 command circuit. The "0" output from 14L11 is used as

an enable signal to start the memory timing chain. The functions enabled by the timing chain with respect to clock cycles are listed as follows:

Ø2.0	Initiate Memory	
Ø4.0	Clear L17	- Clear memory lockout
Ø1.1	Set L11	- Enable S translator
Ø4.1	Set L12	- Enable read Generator
Ø2.2	Set L13	- Initiate strobe
Conditional Commands	61L10	- Clear X
	61L10	- $Z_L \rightarrow X_L$
	61L11	- $Z_U \rightarrow X_U$
	61L12	- Clear U
	61L12	- $Z_U \rightarrow U$
Ø4.3	Set L15	- Enable Inhibit
Ø1.4	Set L16	- Enable Write (Z to M)
Ø3.4	Set L17	- Memory available
Ø2.5	Clear L11	- Disable S translator
	Clear L16	- Disable Write
	Clear L19	-
	Clear L20	-
Ø3.5	Clear L15	- Disable Inhibit

Refer to figure 4-85 for the actual time relationships established by the memory timing chain.

g. SPECIAL CIRCUITS. - The selection of drive current transformers, the generation of Read/Write pulses, the enabling of the sense circuits, and the generation of inhibit pulses in the memory section are accomplished by special circuits (nonstandard logic circuits). Since the memory section of the Trainer uses a magnetic core memory, the special circuits are mainly used to generate a current pulse or to provide a low impedance path for current flow.

(1) MEMORY DRIVERS. - (See figure 4-86 and 5-39.) The Memory Drive circuit is used as the inputs to the various special memory circuits. The output of the S translator serves as the inputs to the memory drivers.

When the selection of the primary, secondary, and line of a given address is made, the translating circuit (standard logic circuit) will output a "1" to a 50Y-- circuit, the Memory driver. The -3 VDC level will appear as the input on pin nine and will reflect a negative voltage level to the base of Q2. The saturation of Q2 will put the collector at ground causing a positive potential to be placed on the base of Q1 which will cut off that transistor. The emitter of Q1 is tied directly to output pin 15. This output will reflect a high impedance to the circuit connected to pin 15 of the selected Memory driver, and will have to rely on its self-biasing network. However, those nonselected memory drivers will have a ground ("0") input applied to them. This ground level will cut off Q2 and will allow the

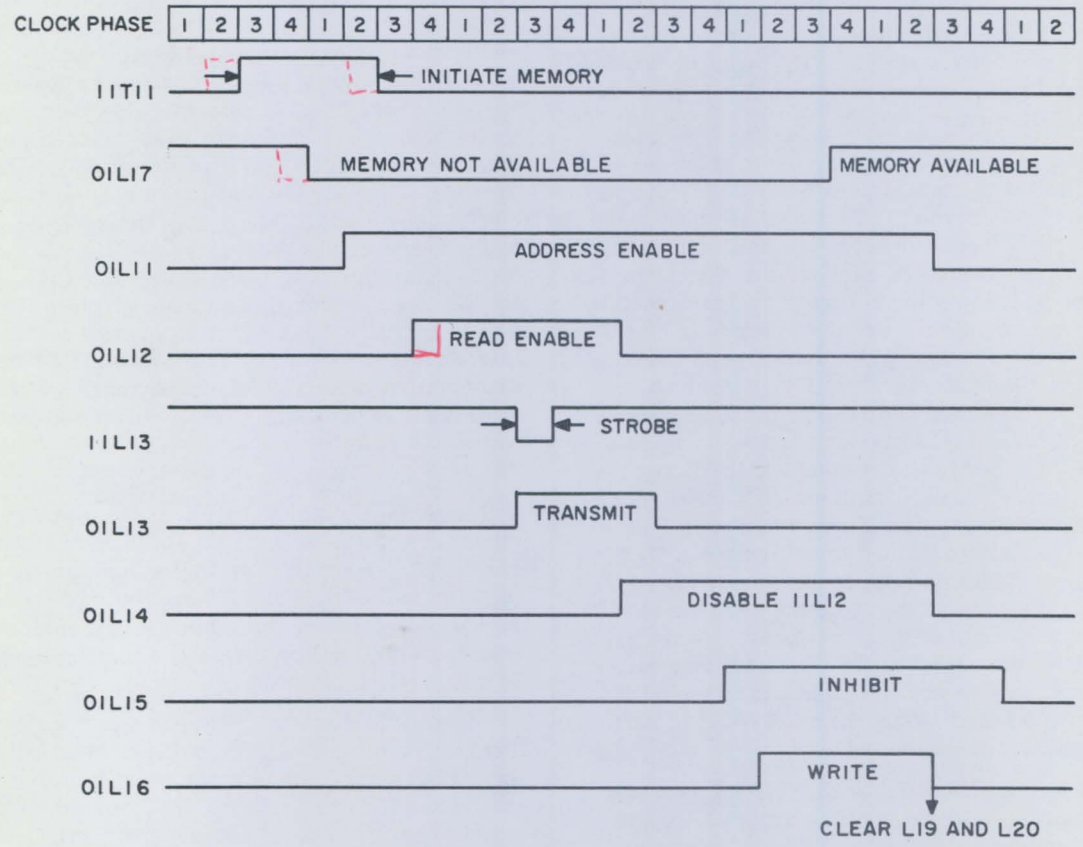


Figure 4-85. Memory Timing

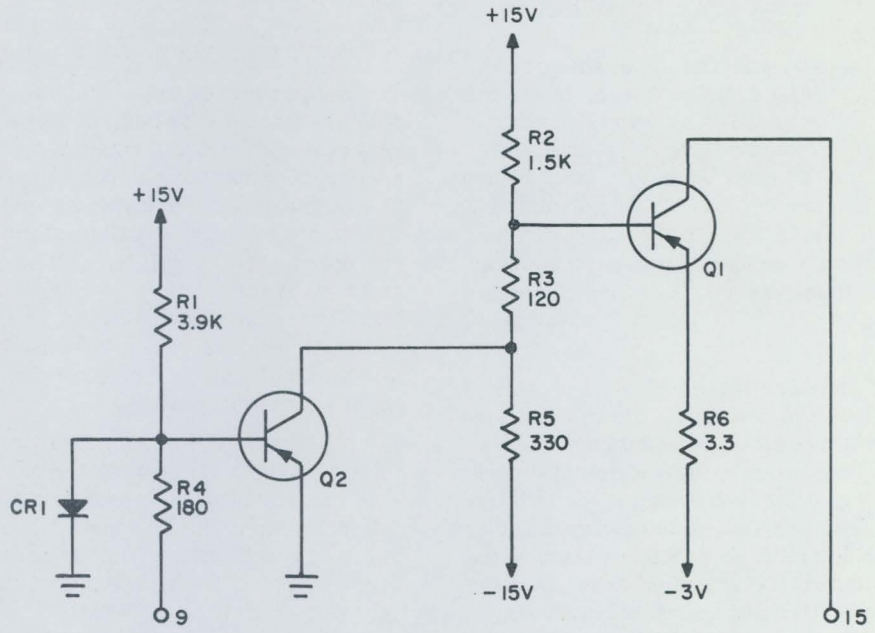


Figure 4-86. Memory Driver Circuits

normal voltage divider network of R2, R3, and R5 to place a negative voltage on the base of Q1. This voltage level will allow Q1 to conduct. The conduction of Q1 will effectively place Q1 and R6 in a series with the biasing network of the circuit connected to pin 15 of the Memory driver.

(2) READ/WRITE CIRCUITS. - The Read/Write circuits operate in an identical manner except that Read current flows in the opposite direction from Write current. The purpose of the Read and Write circuits is to allow a current pulse to flow through the primaries of a selected X and Y transformers (see figures 5-44, 5-45, and 4-87).

An example of the selection and generation of the current pulse is shown in schematic form in figure 4-87. In this example, it is assumed that X primary zero and X secondary zero are selected. The line selection could be any line in the X coordinates. The Write cycle is considered, but the explanation also refers to the Read cycle in that the circuits are identical.

When the output of the S translator is applied to the memory drivers, the input to the 50Y16 driver will force an open or a high impedance output from the 50Y16 circuit. The Read/Write driver (53Y01), now will place a positive voltage on the base of Q5 which will allow it to conduct. The reflected ground from emitter to collector will be applied to the base of both Q1 and Q3.

Before the selection is made, the -3 VDC input from 50Y12 to both circuits allows the Q2 transistors of both circuits to conduct. In the Positive Selector circuit (56Y14), the -3 VDC was applied directly to the base of Q2. This transistor, through its conduction path, applies a near ground level to the base of Q1 keeping it cut off. The negative voltage input to the base of the Q2 transistor of the negative selector (55Y14) is also allowed to conduct. The ground reflected from the emitter to collector places a positive voltage on the base of Q1 keeping it cut off.

The output of the 50Y12 memory driver will apply an open to pins 6 and 13 of the positive and negative selectors (56Y14 and 55Y14). This once again will allow the self-biasing network of these circuits to control the conduction of transistors Q2 in both circuits. In the Positive Selector circuit (56Y14), the base of Q2 will feel a positive potential applied to it. This base voltage will cut off Q2 and will allow the base of Q1 to go positive from the applied +15 volts to R2.

With the applied positive voltage to the base of Q1, the transistor will conduct with the current path from ground connected to pin 14 of the Negative selector, through the one ohm resistor (R6), to pin 15 of the Negative Selector circuit, to pin 12 of the Positive Selector circuit, through emitter to collector of Q1, and finally through R3 to the applied +15 volts. This conduction effectively places the cathode of the diode connected to pin seven at a ground potential. If, as a result of the

applied drive current, pin five of the transformer goes positive, the diode connected to pin seven will conduct and complete the path to ground.

The Negative selector receiving the open from 50Y12 will also result in the cutoff of the Q2 transistor. The voltage divider network of R2, R3, and R5 will reflect a negative voltage on the base of Q1 transistor which will, as in the Positive selector, conduct and provide a ground to the anode of the diode connected to pin nine of the negative selector. If the current pulse induced into the secondary produces a negative potential at pin five of the transformer, the diode connected to pin nine will complete the path for the current pulse to ground.

The positive and negative selectors for line selection operate in the identical manner. The only difference between the two circuits is the number of inputs available to the line selectors. The additional diodes required for line selections are located on diode boards and are connected to the positive and negative selectors as shown in figure 5-44 and 5-45. Examination of the X translation shows that there are eight sets of diodes for each of the four positive and negative selectors. This gives the line selectors the ability to control one line from each of the transformer secondaries.

Before the arrival of the Write enable from 01L16, the 51Y00 circuit (Read/Write driver) will have a -3 VDC input applied directly from the L16 flip-flop in the memory timing chain. This negative voltage will allow Q2 to conduct. The ground reflected from emitter to collector of Q2 will apply a positive voltage to the base of Q3 keeping it cut off. The output of Q3 on pin 10 will be applied to both the X write generator, which is shown in figure 4-87, and to the Y write generator (51Y01). The biasing network of R1, R3, and R4 will place a negative voltage on the base of Q1 (51Y11) and will allow it to conduct. The current path is from the negative 10 volts applied to an externally connected resistor, through the variable resistor (X write adjustment on the memory chassis), through the coil, Q1, a second coil to ground. In this state, Q1 is acting as a current diverter, keeping the current from flowing through the primary of the transformer. Notice that Q1 is connected in parallel with pin 10 of 53Y01 and pin 13 of 54Y01.

During the Write cycle of a memory reference, L16 in the memory timing chain is set. The "0" output from 01L16 turns off Q2 on 51Y00. The voltage divider network of R4, R5, and R6 places a negative voltage on the base of Q3 which drives it into saturation. The ground output from Q3 turns off Q1 on 51Y11. When Q1 is cut off, a current pulse is generated as the fields around the coils in the emitter and collector circuits collapse. The collapsing field tends to keep current flowing in the same direction as when the transistor was conducting. The coil effectively acts like a battery with the positive terminal connected to the emitter of Q1 on 51Y11 and to the emitter of Q3 on 53Y01. The switch in potential on the emitter of Q3 on 53Y01 allows the transistor to conduct. When Q3 conducts, the primary winding of the transformer has a low impedance path with respect to the Current Diverter circuit. The current, for the period of the Current pulse, is diverted through the primary of the transformer. This pulse is coupled across into the secondary windings of the transformer.

The phasing of the transformer is indicated by the dots associated with each of the windings. The Current pulse, diverted through the primary windings, will appear to be negative at pin 13 and positive at pin 12. The output induced into the secondary will appear positive at pin four and negative at pin five. The negative voltage at pin five will be applied to both pins seven and nine of the positive and negative secondary selectors. This negative potential will bias the diode connected to pin nine in the forward direction and will allow this diode to conduct providing one side of the secondary a path to ground.

Pin four of the transformer secondary, now positive, will apply its potential to the A1 end board where it is made available to four lines in the memory boards. The completed path for one of these lines will be provided by the Positive Line selector. This selector will apply a ground to the cathode of the diodes tied to it. With the arrival of the Positive pulse, a path to ground through the forward biasing of the diode will be provided. Thus, the current path for this selection will be through the Negative Secondary selector, to pin five of the transformer secondary, through the secondary to pin four, through A1 end board, through the memory boards, and finally, through the Positive Line selector. Coincident with these events, the Y drive lines would be driven with a current pulse in the same manner just explained.

The Read cycle of the memory chain is accomplished in the same manner except that the second primary (pins 15 and 14) is used. Notice that the phasing of the transformer would force the current in the opposite direction to the Write current. The Read cycle that would have preceded the Write cycle just explained would have utilized the Negative Line selector and the Positive Secondary selector to complete the path for the current pulse.

(3) SUMMARY. - Before the selection of any address is made (Address enable), the Current generator is providing a constant current through the Q1 transistor of the Current diverter. The fields built up around the coils connected to the emitter and collector of Q1 will be the source of potential for the Current pulse to drive the cores. When the address enables become present, the selection of the primary, secondary, and line is accomplished by allowing the secondary and line positive and negative selectors to conduct. Each of the enabled selectors will provide a ground potential to the diodes connected to the Q1 transistors of each circuit. The primary selection through the conduction of Q5, will arm the Q3 and Q1 transistors (53Y01). During the Read cycle of the memory timing, the Read/Write drivers (51Y00) will cut off Q1 of the 51Y11 circuit stopping the current flow through the coils. The collapse of the magnetic field around these coils acts as a battery and applies a potential to the emitter of the armed transistors in the Primary Selector circuit. The Current pulse is then provided a low impedance path through the transistors of the Primary selector

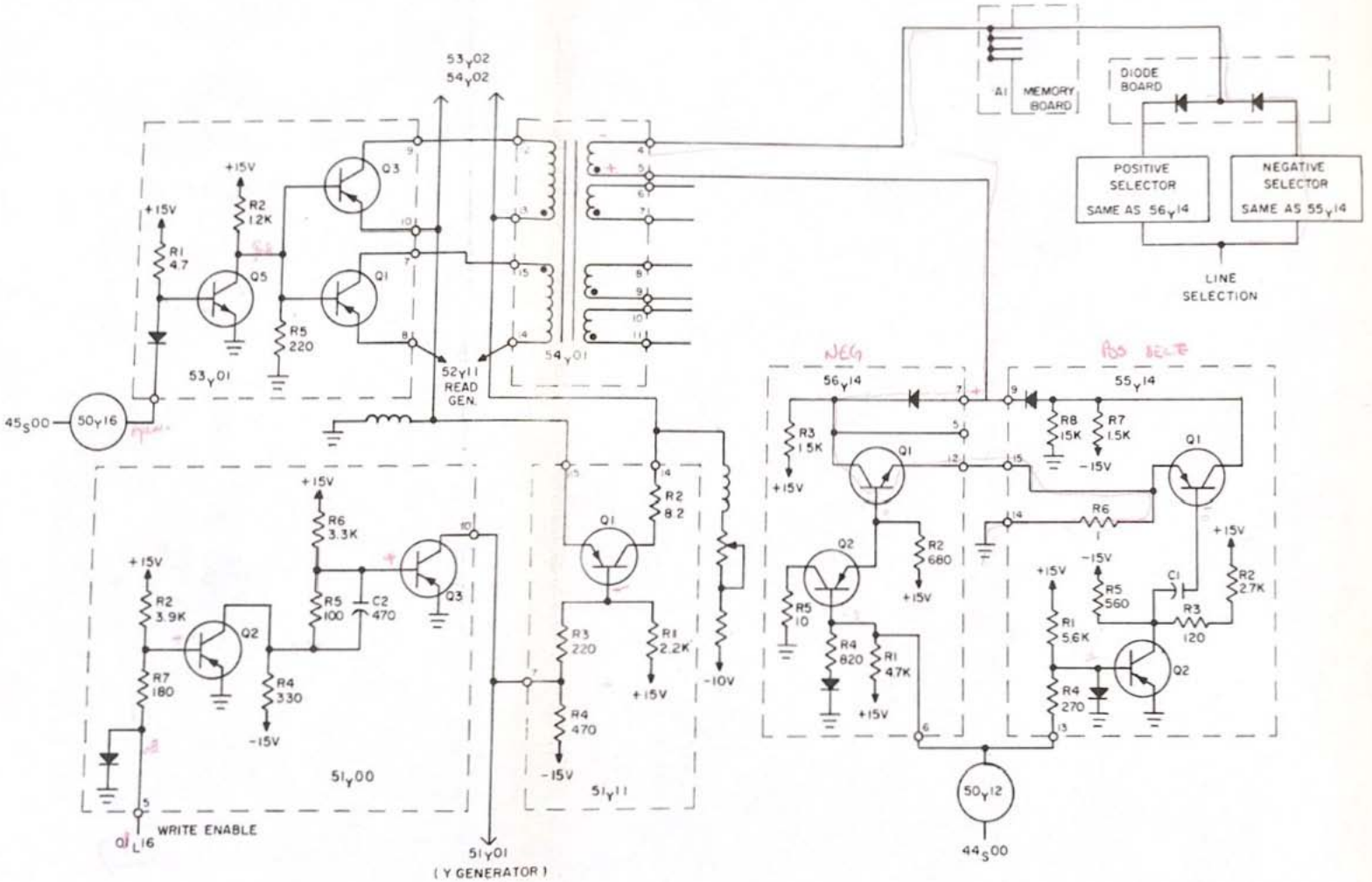


Figure 4-87. Address Selection Circuitry

and the primary of the transformer. This pulse is inductively coupled into the secondary windings. Each of these secondary windings is connected on one side to the A1 end board, and the other to the secondary selectors. Of the selected primary, only one of the secondaries is selected and of the four driver lines, only one line is selected. The current path for the secondary winding is provided through the secondary selectors, through the winding of the transformer, to the A1 end board, through each of the 15 memory boards, and finally terminated at the line selectors.

The Write cycle is accomplished in the same manner except another current diverter is used and is connected to the second primary of the selected transformer. This action, and the phasing of the transformer, will cause current to flow in the opposite direction through the memory boards than it did during the Read cycle. It must also be remembered that the drive circuits for the X and Y are coincident with each other. The control of the current diverters (current generators) for both X and Y is accomplished by the Read/Write drivers (52Y00 for read and 51Y00 for write).

There are four current diverters (current generators) located on the memory chassis. Each of these generators is individually adjustable by controls located in the upper left-hand corner of the memory chassis. These controls determine the current that will be diverted through the primary windings during the Read and Write cycles, thus affecting the drive current pulses passing through the memory boards. These adjustments are made at the factory during checkout of the memory system and any further adjustments should be executed with extreme caution.

(4) SENSE CIRCUITS. - The sense circuits connect the outputs from the memory boards to the Z register. Each of the sense circuits is associated with a stage of the Z register. The sense circuits are shown in figure 5-46, with an example shown in schematic form in figure 4-88.

The input to a sense amplifier is diagonally from opposite sense windings on a memory board. Under static conditions, the output from the Sense amplifier is clamped at ground. If a core in a memory board contains a "1", and a full read pulse is applied, the resulting flux reversal induces a voltage in the sense windings. Either input to 62Y00 goes negative with respect to the other depending upon the polarity of the induced signal on the sense winding. For example, assume pin 15 goes negative with respect to pin 14. The more negative signal on the base of Q3 increases its conduction. At the same time, the conduction of Q4 is decreasing. The voltage drops across R9 and R10 become unequal allowing current flow through R8. The current flow through R8 effectively places R9 and R10 in parallel from the +15 VDC supply. The decreased emitter resistance allows Q3 to conduct more heavily which tends to bias Q4 further towards cutoff. The collector potential of Q3 rises in a positive direction, while the collector potential of Q4 decreases in a negative direction. These voltage changes are coupled to the base of Q1 and of Q2, respectively. Q1 and Q2 are connected in a circuit configuration that is exactly the same as that of Q3 and Q4. Thus, Q2 conducts heavily and Q1 tends toward cutoff. The positive voltage change at the collector of Q2 is coupled to the junction of CR4 - CR2 where the voltage level is clamped at ground potential by CR4. The negative voltage change at the collector of Q1 is coupled to the junction of CR1 - CR3 and is applied as a "1" input to the sense amplifier output circuit. The transistor is driven into conduction and a "0" is applied to the cathode of CR7. Before the Amplifier Output circuit can be enabled, two additional signals must be applied.

If a core held a "0" and a full read pulse was applied, the output from the Sense Amplifier circuit would be the same as under static conditions. The Sense Amplifier output circuit is held at a "0" output.

To fully enable the Sense Amplifier output circuitry, the Memory → Z and the Strobe enable must be present. During a memory reference, if the $M_U \rightarrow Z_U$ and/or the $M_L \rightarrow Z_L$ transmissions are enabled, L20 and/or L19 are cleared. The "1" outputs from 01L19 and/or 01L20 drive the gate circuits on figure 5-46. The gate circuits, shown in figure 4-88 are conventional inverters that are driven into saturation. Their "0" outputs are applied as enable inputs to the sense amplifier output circuits. The output of the 68Y00, controlled by L20, is applied and will affect the 63Y00 through 63Y08 sense amplifier output circuits. The 63Y09 through 63Y14 circuits are controlled by the output of the L19 flip-flop via 68Y03. When the L19 and/or L20 flip-flops are set, the data being received from the memory boards will be destroyed because of the disable output of the 68Y00 and/or 68Y03 circuits.

During the memory reference when the 11L13 inverter is outputting a "1", (Ø2), the signal is applied to the 65Y00 circuit as a strobe enable. This enable indicates that a normal strobe of core memory is to be made. The "1" is applied to pin five of the 65Y00 circuit. The negative input will drive Q3 into saturation and will output a "0" on pin 13. This enable will be applied to all of the sense amplifier output circuits. This signal will remain as an enable to the 63Y-- circuits for the duration of Ø2 or approximately 360 milliseconds.

Assume a memory reference is being made and the Read cycle is enabled. The output of the sense amplifier will be a "1" if the core being read was holding a "1". The output of the Sense amplifier will be applied to pin 10. This input will drive Q4 into saturation resulting in a ground input to the cathode of CR8. If the Memory → Z is present, a ground will also be applied to CR3 of the Sense Amplifier output circuit. With the arrival of the strobe, the final required enable will be present to the cathode of CR4. The positive potential is reflected to the base of Q2, will cut off that transistor, and a -3V signal ("1") will be applied directly as an input to the one side of the Z register flip-flop. If any of the above mentioned signals should appear at the input as a -3 VDC level, the output of the 63Y-- circuit will continue to output a "0" which, during a memory reference, is an indication of a "0" read from the core. (Physical connections from the memory boards to the sense amplifiers are shown in figure 4-89).

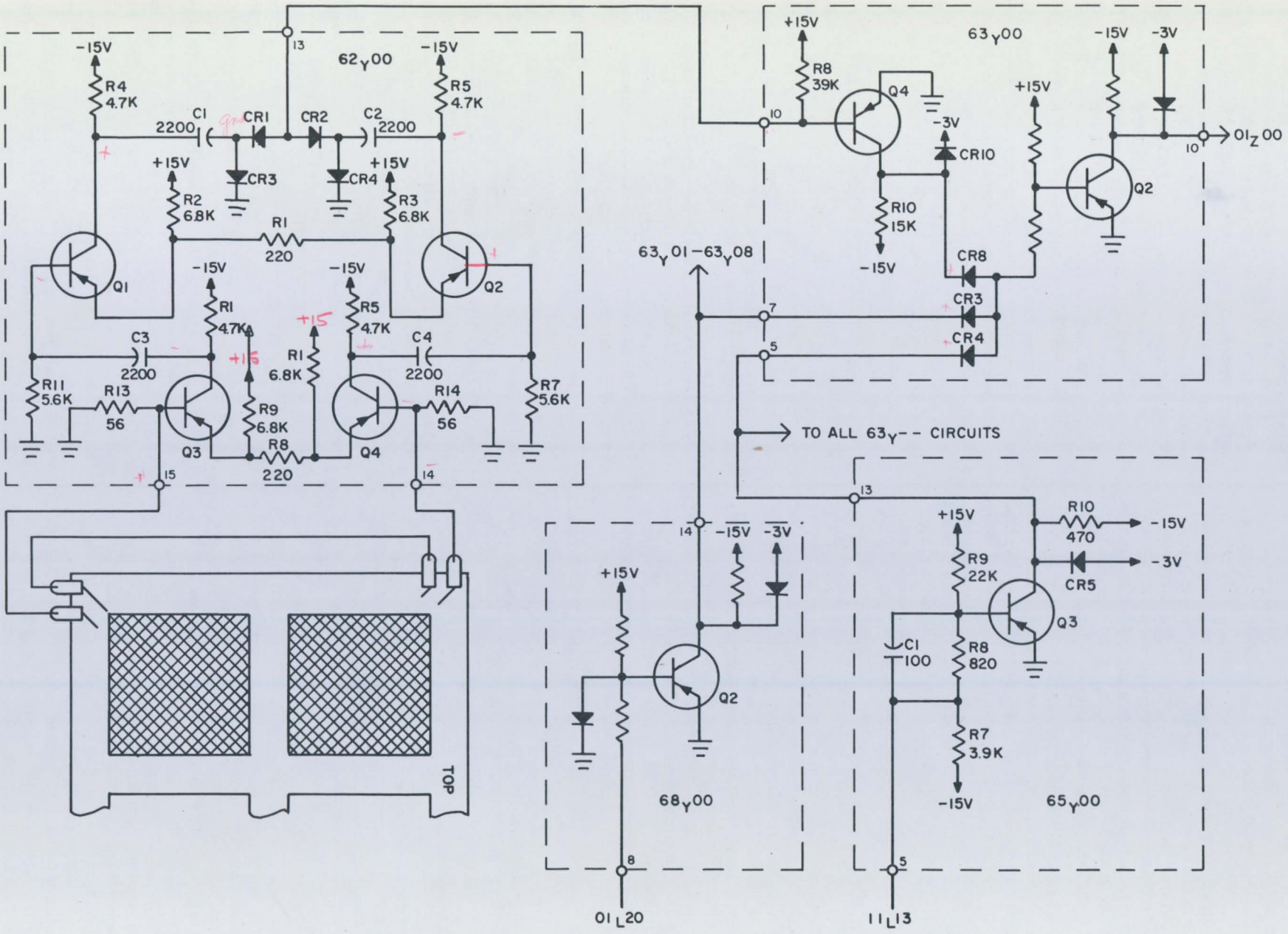


Figure 4-88. Sense Circuitry

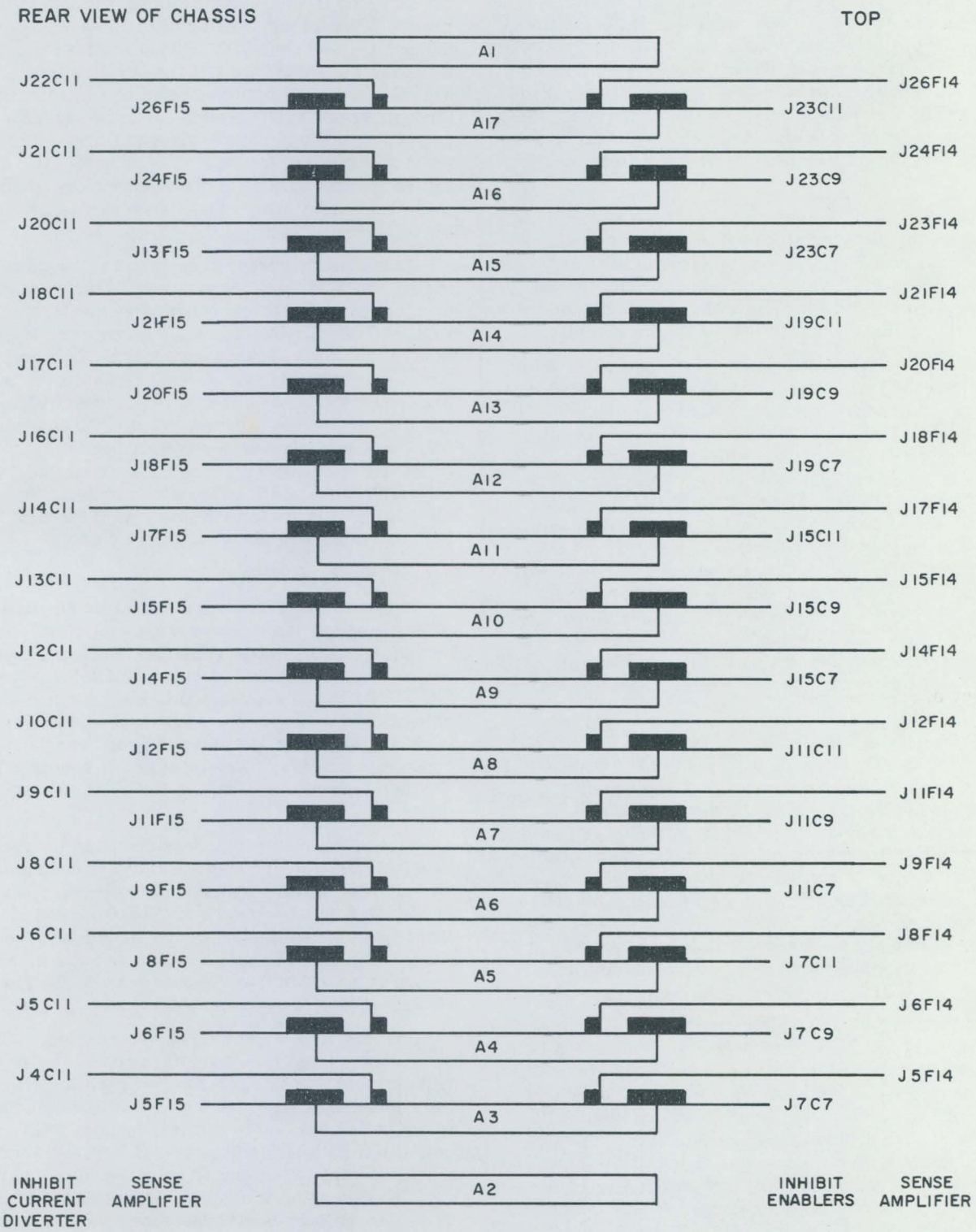


Figure 4-89. Memory Stack Connections

(5) INHIBIT CIRCUITS. - One clock phase prior to the Write cycle of any memory reference, an inhibit enable is generated by the L15 flip-flop of the memory timing chain. The purpose of the inhibit circuits is to cancel out one-half of the Write current pulse for the memory boards that read out a "0" during the Read cycle. The memory boards that supply a "1" output to the Z register must be restored to a "1" during the Write cycle. For these stages of the Z register, the Inhibit current must be disabled, which will allow the core to feel the full Write pulse and return to the "1" state.

For each of the memory boards there is one Inhibit Current Diverter circuit (60Y--) and one Inhibit Enabler circuit (69Y--). These circuits are shown on figure 5-47 with a typical stage shown in schematic form in figure 4-90, which will also be used as an example during the discussion of the inhibit circuits.

Under the static conditions, Q1 of the Inhibit Enable circuit (69Y00) cannot conduct since the base is positive with respect to the emitter which is connected through a coil to ground. Before the Inhibit Enable is generated, the "0" input to the 20Z00 circuit is inverted to a "1" which is applied to the 21Z-- circuits associated with each state of the Z register. The "0" output from each of the 21Z-- circuits will be applied as the input to the inhibit current diverter circuits. Referring to figure 4-90, the "0" input will keep Q2 cutoff. The self-biasing network made up of R3, R5, and R7 in the base circuit of Q1 allow it to conduct. The path of current for Q1 is from the -10 volt supply through the resistor, through the coil, R1, R4, Q1 and finally through the coil connected to the emitter of Q1.

If, during the Read cycle, the output of the 63Y00 circuit was a "1", stage 2^0 of Z would have been set. The "1" output from the zero side of the flip-flop (00Z00) is applied as one of the inputs to 21Z00. This "1" will force a "0" output from 21Z00 even after the arrival of the Inhibit enable. Under these conditions, the Inhibit Current Diverter circuit will remain in the same state as the static condition. This will allow the memory board associated with this stage of the Z register to feel the full Write Current pulse and will return the selected core to the "1" state.

Assume a "0" was read from the memory board during the Read cycle of the memory timing chain. That stage of the Z register would have remained in the cleared state and would result in a partial enable ("0") being applied to the 21Z00 circuit. The setting of the L15 flip-flop in the memory timing chain will cause a "1" input to 20Z00 which will invert the signal and apply it as a final enable to the 21Z-- circuits. In figure 4-90, the 21Z00 will now output a "1" (-3 VDC) to the Inhibit Current Diverter circuit which will drive Q2 into conduction and saturation. The reflected ground from emitter to collector will place the base of Q1 at a positive potential causing it to cut off. The turning off of Q1 will cause the magnetic field built up around the coils to collapse and generate a voltage that will try to keep current flowing in the circuit. This potential, acting like a battery, is applied as a positive to the emitter of Q1 in the Inhibit Enable circuit (69Y00). Current now has a path from the -10 volt supply through the 45 ohm resistor, the coil, to pin 11 which is jumpered to pin 15, through the one ohm resistor, the inhibit winding of the memory board, through Q1, and finally to ground through the coil connected to the emitter of Q1. A parallel path is also provided from the coil connected to the collector through R1 of 69Y00 through Q1, and to ground. The impedance of this circuit will now allow any great amount of current to flow. The resistor, R1, is connected into the circuit to damp out the overshoot that may occur from stray capacitance that is developed in the inhibit windings.

This inhibit current is allowed to flow through the entire Write cycle of the memory timing chain. Approximately 0.8 microseconds after the Write current is turned off, the Inhibit enable is removed from the input to 20Z00. This will produce a "1" output from 20Z00 which in turn will force a "0" output from the 21Z-- circuits. With the "0" applied as the input to the Inhibit Current Diverter circuit, Q2 is again cut off. The self-biasing network connected to the base of Q1 will allow that transistor to conduct again and the circuits to return to the static state. The physical connections made from the inhibit circuits to the memory stack are shown in figure 4-89.

4-8. POWER SUPPLY.

Power to the Trainer and indicator circuits is supplied by one main power supply physically located behind the main control console. Voltages generated by this supply include -15 volts, +15 volts, -3 volts, and -10 volts which are used in the logic and memory circuits of the Trainer. Additional voltage levels include the -54 volts and -90 volts used for the control console indicators. The input power required by the Trainer's supply is 60 cycle, 117 volts AC, single phase.

The entire power supply is shown in figure 5-52. The line voltages are applied to two transformers, T1 and T2. The T2 transformer secondary output is rectified by the network of CR5, 6, 7, and 8. The filtering is accomplished by the L1 coil and the C2 capacitor. The +15 volt line is fused at 2 amps (F2) before being applied to the buss bars on the four chassis of the Trainer.

Transformer T1 has three secondaries. The output voltages from these windings are 100 volts, 60 volts, and 19 volts. The 100 volt secondary winding is rectified by CR18, fused by F6 at 0.25A and applied to the indicators on the control console. This is the only use for the -90 volt potential. The -54 volt potential is supplied by the 60 volts secondary with AC rectification being accomplished by CR14, 15, 16, and 17. This voltage is also fused by F5 at 0.25 amps and applied to the indicators on the control console.

The -15 voltage is generated by rectifying the 19 volts secondary through the use of CR1, 2, 3, and 4. The filtering of this voltage is accomplished by the L2 coil and the C1 capacitor. The line is fused by F3 at 20 amps before being applied to the circuits in

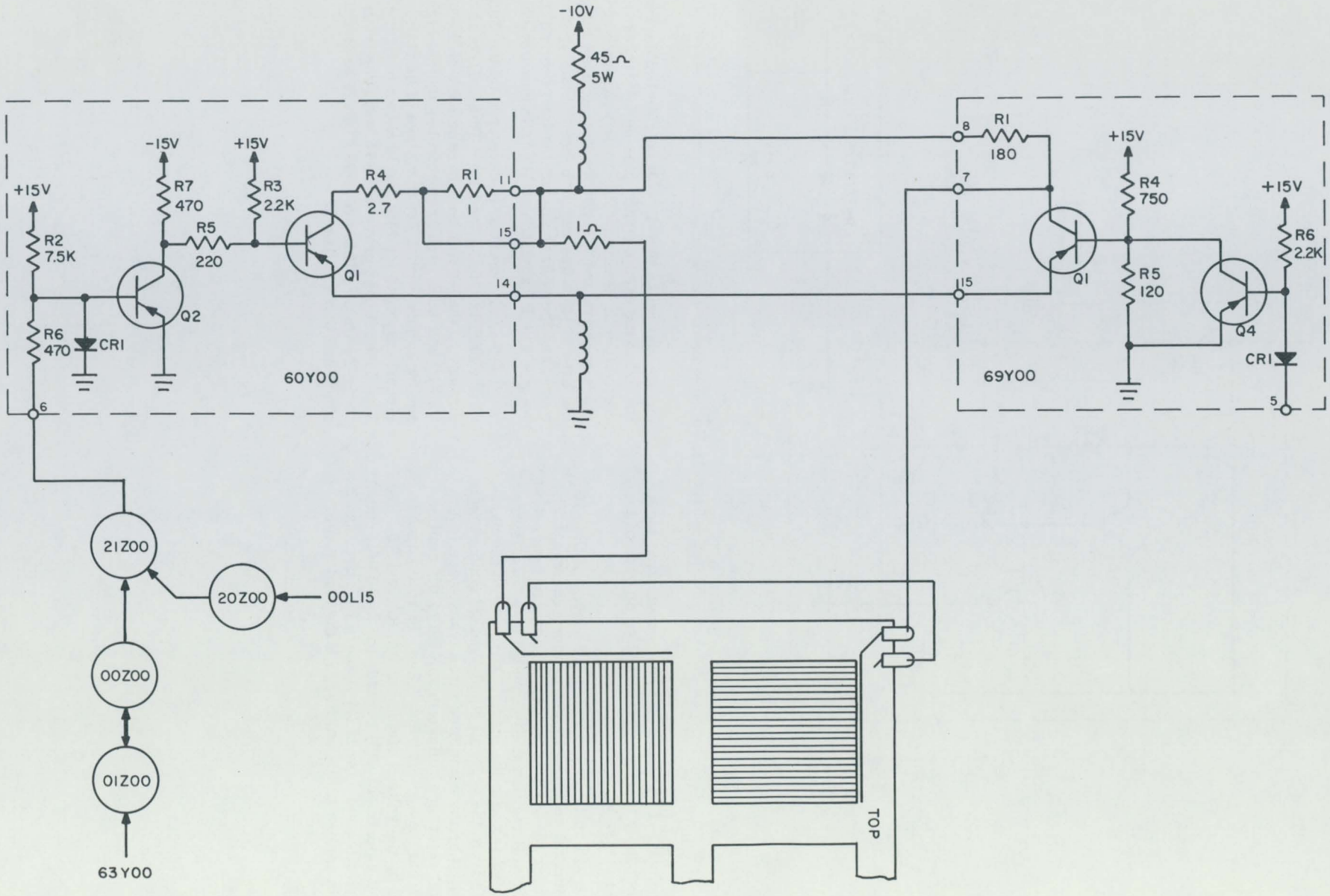


Figure 4-90. Inhibit Circuits

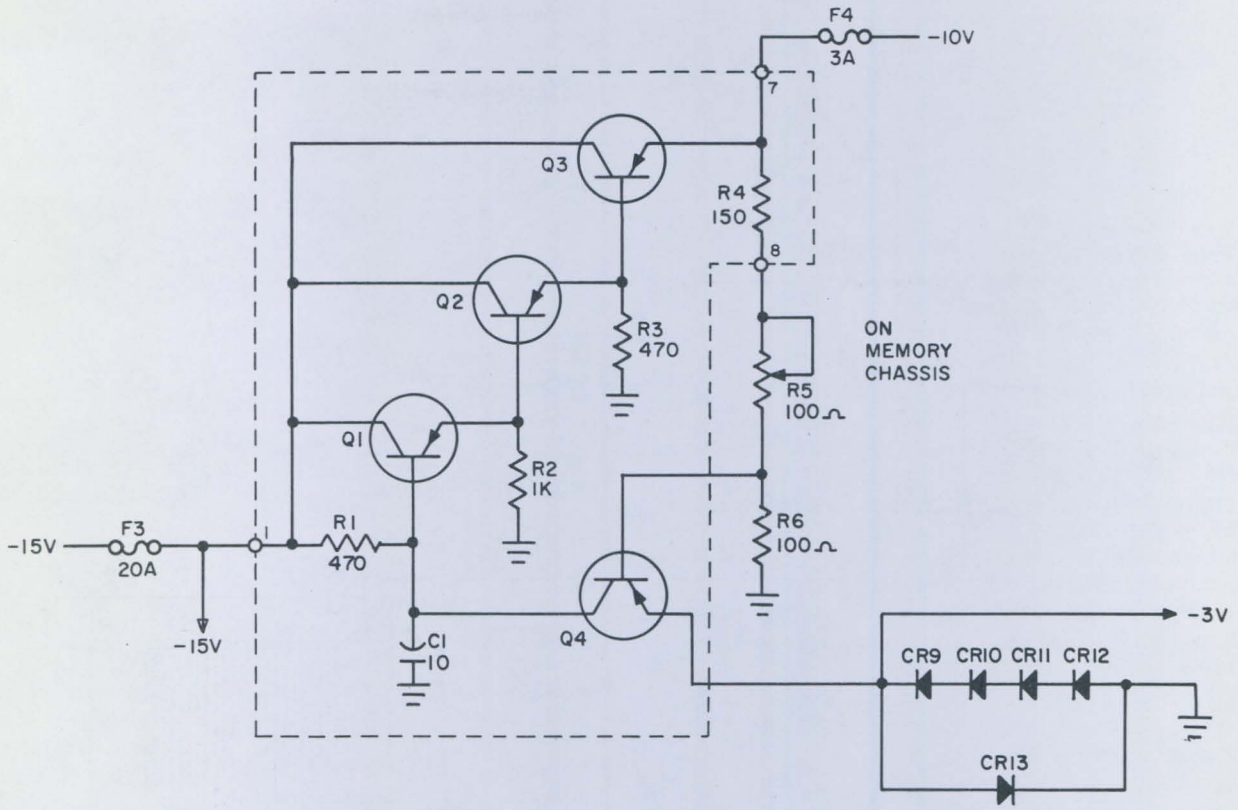


Figure 4-91. -10 Volt Regulator

the Trainer. Buss bars that are connected to the various supply voltages are identified on each of the chassis maps.

The -15 volt supply is also the source of the -10 volt potential required by the memory chassis. The circuit for the -10 volt supply is shown in figure 4-91. The -10 volt regulation is provided by a transistor (Q3) in series with the load. The input to the regulator is from the -15 volt supply and Q3 conduction is controlled by the remaining three transistors. A constant reference voltage is applied to the emitter of Q4 from the -3 volt buss. The -10 volt output is sensed by the voltage divider of R4, R5, and R6. The -10 volt adjustment (R5) is made for a -10 volt output at which time the base of Q4 is more negative than the -3 volts applied to the emitter. If the load should change and cause the -10 volt potential to decrease, the following events occur: bias on Q4 decreases and conduction increases. This makes the base of Q2 more negative and it conducts more. This makes the base of Q3 more negative, the transistor conducts more, and the voltage across the load returns to -10 volts. The R5 adjustment is on the memory chassis.

The -3 volt potential is developed by the Zener diodes of CR9, 10, 11, and 12. This voltage is applied directly to the -3 volt buss bars on each of the chassis. The 15 volt buss bars are protected against high voltage by a diode with a Zener voltage of 22 volts. The circuit employed for this use is shown in figure 4-32. The Zener diodes prevent the 15 volt buss from rising above 22 volts. This prevents damage to circuits which use these voltages.

SECTION 6

TROUBLESHOOTING AND REPAIR

6-1. GENERAL.

To make certain that the Trainer performs properly, perform all inspections and tests contained in this section before placing the Trainer in general operation. Perform the inspections and tests before service and repair to determine the extent of repair required. Perform the inspections and tests after repair and perform appropriate parts of the tests between repair procedures to make certain that repairs are made correctly. A record should be kept of failures, operating time, repair time, failure analysis, and failure causes. If a failure occurs while performing tests found in this section, the cause of failure must be determined so that necessary corrective action can be taken. Troubleshooting consists of visual inspection, component testing, and operational testing to provide statistical samples of reliability. The results provide information to determine whether reliability meets requirements, whether reuse is feasible, or whether repair is warranted. This section contains all the maintenance routines, their absolute address, a description of each instruction, and flow diagrams for each of the routines.

6-2. PREVENTATIVE MAINTENANCE.

a. GENERAL. - Maintenance procedures used with the Trainer differ from normal procedures used with other electronic equipment. Trainer maintenance requires a logical analysis of a malfunction rather than an electronic analysis. It is essential that maintenance personnel become familiar with transmission paths within the Trainer. This is helpful in determining the failing area of the Trainer by taking note of the visual indications on the console when a malfunction occurs. It is also essential that maintenance personnel become familiar with the programs and routines being run in the Trainer. In this way, when a malfunction occurs, the correct maintenance routine can be loaded into the Trainer to check the failing area.

The maintenance routines isolate malfunctions to a particular section or functional part of the Trainer. When a malfunction is isolated to a particular section, it can be further isolated to a printed circuit card or cards by using an oscilloscope and a multimeter.

Before attempting any maintenance, personnel should be thoroughly familiar with the contents of Section 4 (Principles of Operation) and Section 5 (Functional Schematics). During repair, the functional schematics will provide pertinent information for maintenance personnel.

6-3. TEST EQUIPMENT AND SPECIAL TOOLS.

Test equipment and special tools required for inspecting and testing the Trainer are listed in table 6-1.

TABLE 6-1. TEST EQUIPMENT AND SPECIAL TOOLS

Oscilloscope, Tektronix, Model 535 (Dual Trace Preamp)
Voltmeter, DC, Weston Model 93, 0.5 percent accuracy or equivalent
Soldering iron
Needle nose pliers
Wire cutters

6-4. TEST CONDITIONS.

The Trainer should be tested under normally expected operating conditions. An air-conditioned room is desirable, if available.

6-5. VISUAL INSPECTION.

Visual inspection of the Trainer should be made of its parts and its subassemblies for signs of damage. Make certain all circuit cards are firmly seated into the connectors. Inspect the cards to see if proper spacing is maintained between each card. Make a thorough visual check for loose or broken taper pins, broken or bared wires, and possible broken or damaged connectors.

6-6. ELECTRICAL TESTS.

Electrical tests consist of testing the voltage output of the power supply contained in the Trainer. Measure the voltages specified in table 6-2.

6-7. CONSOLE CONTROLS AND INDICATORS.

The console (operator's panel) allows the operator to control the functions of the Trainer. A list of controls and a brief description of their function follows.

a. MASTER CLEAR SWITCH. - When the MASTER CLEAR switch is in the momentary down position, all flip-flop indicators on the console panel will be extinguished. In the momentary up position, the first flip-flop of the timing chain is enabled. The function of the MASTER CLEAR switch is disabled when the Trainer is in the high-speed mode of operation.

b. HIGH SPEED SWITCH. - When the HIGH SPEED switch is in the momentary up position, the Trainer runs at normal speed (if the timing chain was previously enabled). In the center position, no action is initiated. The RUN indicator is lit when the Trainer is operating in the High-Speed mode. Any stop condition, or entering the Operation Step mode, extinguishes the RUN indicator.

c. OPERATION STEP SWITCH. - When the OPERATION STEP switch is placed in the momentary down position, the Trainer stops at the beginning of the next sequence to be executed. The sequence to

TABLE 6-2. VOLTAGE MEASUREMENTS

Voltage Supply	Voltage Range (volts)	Maximum Ripple Voltage (volts)	Test Points *
-3	-2.7 to -3.3	--	TB3-5
-10	-9 to -12	0.2	TB3-7
-15	-12.5 to -16.5	0.5	TB3-1
+15	+12.0 to +16.5	0.2	TB3-3
-54	-48 to -60	0.5	TB3-12
-90.5	-80 to -100	Peak (1/2 wave rectified)	TB3-14

*TB3 is located inside the control panel on the back frame

be initiated will be dependent upon the instruction being executed. The A sequence will always be the last sequence to be completed. Placing the switch in the momentary down position again causes the Trainer to run and then stop at the beginning of the next sequence to be initiated again. No action is initiated with the switch in the center position.

d. PHASE STEP SWITCH. - One clock phase is produced each time the PHASE STEP switch is placed in the momentary down position. The center position causes no action. To enter the Phase Step mode, HIGH SPEED DISCONNECT switch must be in the up position. A memory reference cannot be made during phase step operation.

e. SELECT STOP SWITCH. - Placing the SELECT STOP switch in the up position causes the Trainer to stop when the particular instruction (f = 15, j =3), Jump and Stop if the SELECT STOP switch is selected, is executed. The jump is executed and the Trainer stops at the beginning of the sequence which is enabled by the last A sequence. The SELECT STOP indicator will light when the above conditions are met. The STOP indicator will light when the Stop instruction (f = 15, j = 2) is executed. The Trainer will stop after the completion of the next A sequence.

f. SELECT JUMP SWITCH. - Placing the SELECT STOP switch in the up position establishes a condition whereby a jump from one address in memory to a specified address in memory occurs if a particular instruction (f = 15, j = 1) is executed when the SELECT JUMP switch is engaged. During all other times, the next sequential instruction will be executed.

g. HIGH SPEED DISCONNECT SWITCH. - When the HIGH SPEED DISCONNECT switch is in the up position, the corresponding indicator glows. When the OPERATION STEP switch is placed in the down position and memory is not being used, Trainer operation will cease and the Phase Step mode is permitted. The center position of the HIGH SPEED DISCONNECT switch extinguishes the indicator and disables the phase step mode of operation.

h. MANUAL SET AND CLEAR SWITCHES. - The MANUAL SET and CLEAR switches allow individual flip-flops to be set and cleared for testing and operating purposes. These switches are pushbuttons located in modules on the console panel.

i. FLIP-FLOP INDICATORS. - Each flip-flop indicator displays the contents of the flip-flop to which it is connected.

j. PHASE INDICATORS. - After each clock phase is generated, a corresponding PHASE 1, PHASE 2, PHASE 3, or PHASE 4 indicator glows until the succeeding clock phase has been generated. As an example: PHASE 1 indicator glows after clock phase one has been generated and remains glowing until clock phase two has been generated. At this time, PHASE 2 indicator glows, etc.

k. RUN INDICATOR. - The RUN indicator glows whenever the Trainer is operating at high speed. The RUN indicator is extinguished by depressing the OPERATION STEP switch or the execution of a stop instruction takes place.

l. REPEAT SWITCH. - The REPEAT SWITCH allows the operator to repeat over and over the same instruction contained in the U register. The sequential addressing of memory is continued. This mode of operation is used primarily for manually loading a program into Trainer memory and should not be used during high-speed operations.

6-8. TESTING REGISTER STAGE INDICATORS AND SET AND CLEAR PUSHBUTTONS.

To test each register stage (flip-flop) indicator, MANUAL SET, and MANUAL CLEAR pushbutton, attach an oscilloscope to the test point on the output of the flip-flop to which the indicator under test is connected. Manually set and clear the flip-flop until the indicators are operating properly. Minus three volts should be registered on the oscilloscope attached to the zero side of a flip-flop when the indicator is lit, and zero volts when the indicator is extinguished with the exception of the X register. Refer to the appropriate functional schematic in Section 5 for test point information. Demonstration of the sequence indicators and the designator indicators requires the Trainer to be in Phase Set mode of operation. To clear these indicators, hold down the associated MANUAL CLEAR pushbutton and depress PHASE STEP switch several times.

6-9. TESTING OPERATION STEP, PHASE STEP, AND HIGH SPEED DISCONNECT.

The phase indicators, DISCONNECT HIGH SPEED switch, and PHASE STEP switch are tested as follows:

a. Manually insert an instruction (20) in the U register.

b. Manually insert 00001 into the X register.

c. Place the DISCONNECT HIGH SPEED switch in the up position. The indicator associated with it should glow.

d. Enable the timing chain and the C sequence.

e. Depress the PHASE STEP SWITCH. For each time it is depressed, the Trainer will step through one clock phase at a time. The phase indicator lights.

f. Hold the OPERATION STEP switch in the down position and continue to depress the PHASE STEP switch until the timing chain starts to advance.

g. Continue to momentarily depress the PHASE STEP switch until the timing chain is again enabled. The sequence of events for this instruction should add one to the contents of the A register.

h. Return the DISCONNECT HIGH SPEED switch to the normal position. Note that depressing the PHASE STEP switch no longer has any effect and clock phases are not generated.

i. Master clear the Trainer. Place the REPEAT switch in the up position.

j. Manually insert an instruction (74) into the U register.

k. Enable the timing chain and the D sequence.

l. Manually store the following program in memory addresses 000 through 004, using the following steps.

001 30 000
002 20 001
003 33 017
004 64 002
005 66 001

(1) Manually insert the first instruction into the Q register and set the P register to 001.

(2) Depress the OPERATION STEP switch. The Trainer should store this instruction in memory at address 001. The P register should now be advanced to the memory address 002 and the D sequence should be enabled.

(3) Manually clear and insert the next instruction into the Q register.

(4) Repeat this procedure until all instructions have been entered into the memory of the Trainer.

6-10. TESTING THE HIGH SPEED RUN AND NORMAL STOP.

The routine that was loaded as part of the OPERATION STEP (paragraph 6-9, l) will be used to check the HIGH SPEED switch and the normal stop operations of the Trainer.

a. Master clear the Trainer.

b. Enable the timing chain. Check to be sure that the A sequence is enabled.

c. Place the OPERATION STEP - HIGH SPEED switch in the up position.

d. If the routine was loaded properly and the Trainer executed the instructions normally, the A register will contain octal 00017 when the STOP indicator is turned on. If this sum has not been accumulated by the A register, Master Clear the Trainer and use the OPERATION STEP switch to step through the routine. During this time, the operator should watch each instruction as it is being executed to ensure that the instructions of the routine have been properly inserted and loaded.

e. If it is determined that all controls are operating properly, return the REPEAT switch to the center position and Master Clear the Trainer.

6-11. MEMORY TESTS.

a. MAGNETIC CORE STORAGE TESTS. - This series of routines, or tests, demonstrates the capacity of memory, the ability of individual addresses to be manually altered, and sensitivity to internally generated noise. Memory capacity should be 512 15-bit words. It should be possible to manually alter individual addresses. Memory should function properly, remaining insensitive to internally generated noise, for worst possible patterns.

(1) MEMORY TEST PROCEDURES. - (Instructions and addresses are in octal notation.) Perform the following memory test procedures.

(a) To test word length, manually insert words and check by reading manually until it is determined that all 15 bits can be altered.

(b) To establish that the memory capacity is 512 words, perform the following steps:

1) Master Clear the Trainer.

2) Enable the timing chain and the D sequence.

3) Manually load an arbitrary pattern into the A register.

4) Manually load 76 into the U register.

5) Engage the HIGH SPEED switch and allow the Trainer to cycle through the memory addresses. The P register will advance normally through each sequential address allowing the contents of the A register to be stored at all addresses of memory. After the P register has been advanced through address 777, depress the OPERATION STEP switch.

6) Manually read various addresses of memory to examine for the pattern initially inserted into the A register.

(c) To establish that the memory is capable of operating properly at the maximum repetition rate, perform the following test procedures.

1) Manually load an arbitrary pattern into any selected memory cell.

2) Manually insert the instructions at addresses 001 and 002.

001 12 XXX (the lower nine bits is the address of the pattern loaded in step one.)
002 64 001.

3) Set the P register to 001.

4) Enable the A sequence and the timing chain.

5) Engage the HIGH SPEED switch. The Trainer repeats the two instructions which will enter the Q register with the pattern loaded at address XXX.

6) Allow the Trainer to run for 10 minutes or longer, and check to ensure that the information inserted in the memory cell and the Q register is unaltered.

7) Repeat the above steps for other addresses as necessary.

b. MEMORY TEST (BRAINWASH) PROGRAM. - The objective of this diagnostic test program is to test the ability of the Trainer to accept and retain various bit configurations within memory. A series of patterns is used, ranging from the simplest to one generating the maximum amount of noise in the magnetic core matrix sense winding. The test is diagnostic in that the individually faulty bits in the memory array are located exactly, and associated with a specific testing sequence. This information, along with knowledge of the logical structure of the memory allows determination of the probable cause of memory failure.

(1) BRAINWASH #1. - Brainwash #1 is loaded by the standard procedure in the addresses below 377 and checks memory between 400 and 777. First, a check is made by storing and reading back zeros at all addresses above 400. If this is successful, it stores and reads back ones at all addresses above 400. Last, load memory with alternate ones and zeros, complement the complete pattern seven times, and check each address for correctness.

This test will cycle ten times and stop at address 067, unless an error occurs. The program may be recycled from here by enabling High-Speed.

(2) PROGRAM CHARACTERISTICS. - Program characteristics are as follows:

Computer stops at address:

116 - Q = correct pattern, A = error pattern; at first error address enable High Speed and stop;

122 - Q = first error address in lower three characters, A = last error address. If A = 400, we have used up our B index without finding a correct pattern. If it is desired to check error area, use Inspect and Change routine starting at address Q.

156 - Same as 116 above. Enable High Speed and stop.

162 - Same as 122 above.

232 - Same as 116 above. Enable High Speed and stop.

213 - Q = Error Address

226 - Same as 116 above. Enable High Speed and stop.

221 - Q = Error Address

After any error stop, program will continue by enabling High Speed.

(3) BRAINWASH #2. - This program is stored in memory by the standard procedure at addresses above 400 and check memory from address 001 to 377. It uses the same format as Brainwash #1.

There is a special Inspect and Change routine at address 664 since the standard Inspect and Change is destroyed by the running of this test.

(4) Program characteristics are as follows: ALARMS and/or REMARKS: If computer stops at address:

516 - Q = correct pattern, A = error pattern of first error address; enable High Speed and stop at

522 - Q = first error address, A = last error address. If it is desired to check error area, then use Inspect and Change routine starting at address Q.

556 - Same as 516 above. Enable High Speed and stop at

562 - Same as 522 above.

632 - Same as 516 above. Enable High Speed.

613 - Q = error address

626 - Same as 516 above. Enable High Speed and stop at

621 - Q = error address

662 - Same as 516 above. Enable High Speed and stop at

634 - Q = error address

To continue program after error stop, depress the HIGH SPEED switch.

DS 4596

BRAINWASH #1

022	66023	STOP
023	61022	JP A=0
024	10050	ENT Q=50
025	53000	EXT. FUNC
026	56042	STR AL9--042
027	70027	JP NO IR
030	50000	INPUT
031	30000	CLR B
032	00077	LYQ -- A
033	27056	SUBT A-(Y+B)
034	61047	JP A=0
035	33007	B=7 SKIP

```
036   64032   JP
037   20025   ADD A+25
040   60022   JP A≠0
041   06055   ENT A=WORD
042   76      STR A --
043   06042   ENT A=STR ADDRESS
044   20001   ADD A+1
045   64026   JP
046   07777   MASK
047   12046   ENT Q=MASK
050   02055   L(Y)Q -- A
051   45003   LSH A 3
052   15000   SET AN FOR BN=1
053   76055   STR A -- WORD ASSY
054   64027   JP
055           WORD ASSY
056   00037   CODE 0
057   00052   CODE 1
060   00074   CODE 2
061   00070   CODE 3
062   00064   CODE 4
063   00062   CODE 5
064   00066   CODE 6
065   00072   CODE 7
066   66067   STOP
067   04010   ENT A=10
070   76250   STR A -- PRG. INDEX
071   10000   CLR Q
072   30000   ENT B=0
073   75400   STR Q -- (400+B)
074   07400   ENT A=(400+B)
075   62102   JP A NEG
076   36241   RJP CK 0
077   33377   B= 377 SKIP
100   64073   JP
101   64125   JP
102   05400   ENT A=B+400
103   56110   STR AL9 --
104   75400   STR Q--(400+B)
105   07400   ENT A = (400+B)
106   62113   JP A NEG
107   36241   RJP CK 0
110   06      ENT A= (  )
111   64115   JP
112   66113   STOP
113   33377   B=377 SKIP
114   64104   JP
115   66116   STOP
116   05377   ENT A = ERROR ADDRESS
                (LAST)
117   74252   STR Q -- 252
120   12110   ENT Q = ERROR ADDRESS
                (FIRST)
121   66122   STOP
122   12252   ENT Q=252
123   33400   B=400 SKIP
124   64073   JP
125   30000   CLR B
126   55000   COMPLEMENT Q
127   75400   STR Q -- (400+B)
130   07400   ENT A=(400+B)
131   62133   JP A NEG
132   64140   JP
133   57000   COMPLEMENT A
134   36241   RJP CK 0
135   33377   B=377 SKIP
136   64127   JP
137   64165   JP
140   05400   ENT A=B+400
141   56150   STR AL9 -- 150
142   75400   STR Q -- (400+B)
143   07400   ENT A =(400+B)
144   62146   JP A NEG
145   64153   JP
146   57000   COMPLEMENT A
147   36241   RJP CK 0
150   06      ENT A= (  )
151   64155   JP
152   66153   STOP
153   33377   B=377 SKIP
154   64142   JP
155   66156   STOP
156   05377   ENT A=ERROR ADDRESS
                (LAST)
157   74252   STR Q -- 252
160   12150   ENT Q=ERROR ADDRESS
                (FIRST)
161   66162   STOP
162   12252   ENT Q=252
163   33400   B=400 SKIP
164   64127   JP
165   04007   ENT A=7
166   76251   STR A -- STR RPT INDEX
167   30000   CLR B
170   75400   STR Q -- (400+B)
171   55000   COM PLEMENT Q
172   33377   B=377 SKIP
173   64170   JP
174   30000   CLR B
175   07400   ENT A=(400+B)
176   57000   COMPLEMENT A
177   77400   STR A -- (400+B)
200   33377   B=377 SKIP
201   64175   JP
202   06251   ENT A=STR RPT INDEX
203   24001   SUBT A-1
204   76251   STR A -- STR RPT INDEX
205   61207   JP A=0
206   64174   JP
207   30000   CLR B
210   07400   ENT A=(400+B)
211   62230   JP A NEG
212   60230   JP A≠0
213   33377   B=377 SKIP
214   07400   ENT A=(400+B)
215   62217   JP A NEG
216   64224   JP
217   57000   COMPLEMENT A
220   36241   RJP CK 0
221   33377   B=377 SKIP
```

222	64210	JP
223	64234	JP
224	13377	ENT Q=(377+B)
225	66226	STOP
226	11400	ENT Q=400+B
227	66221	STOP - JP
230	13377	ENT Q=(377+B)
231	66232	STOP
232	11400	ENT Q=B+400
233	66213	STOP - JP
234	06250	ENT A = PRG RPT INDEX
235	24001	SUBT A-1
236	76250	STR A -- PRG RPT INDEX
237	61253	JP A=0
240	64071	JP
241	64000	CK 0
242	60244	JP A≠0
243	64241	JP
244	06241	ENT A=RJP ADDRESS
245	20003	ADD A+3
246	76241	STR A -- RJP ADDRESS
247	64241	JP
250	____	PROGRAM REPEAT INDEX
251	____	STR REPEAT INDEX
252	____	STORAGE
253	66067	STOP - JP

See figures 6-1 thru 6-2 for flow diagrams.

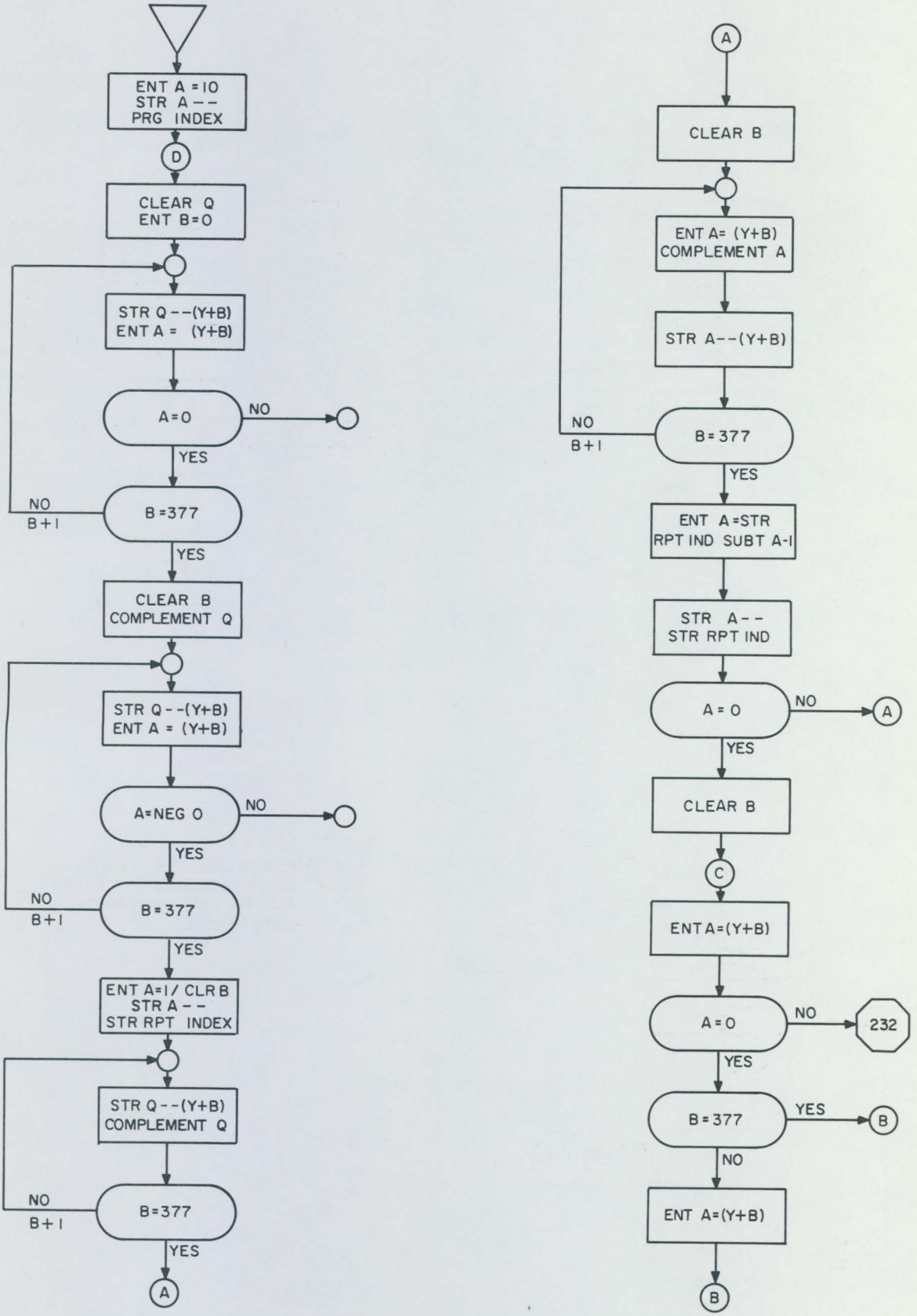


Figure 6-1. Brainwash #1 Flow Diagram - I

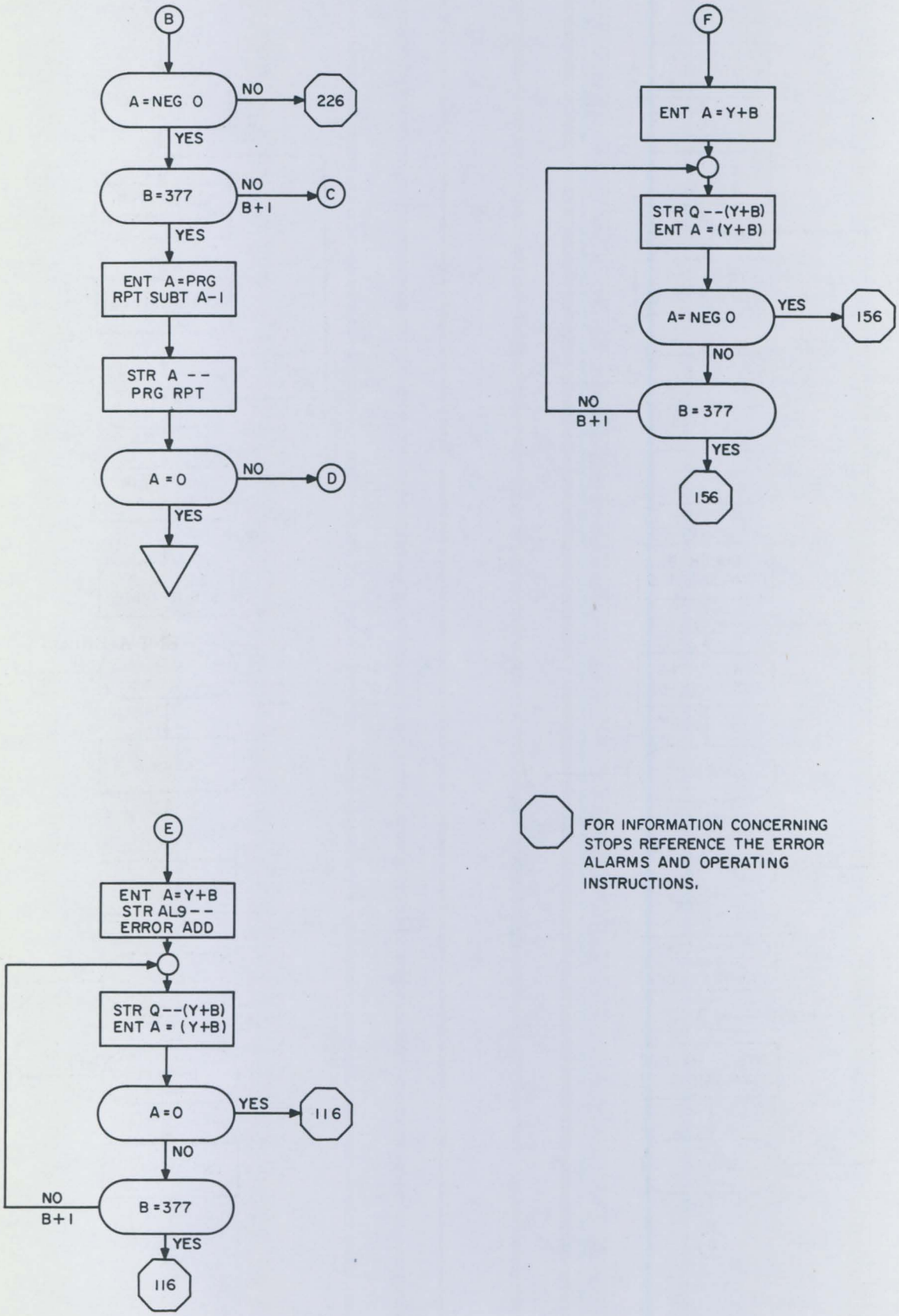


Figure 6-2. Brainwash #1 Flow Diagram - II

```
422  66423  STOP
423  61422  JP A=0
424  10050  ENT Q=50
425  53000  EXT FUNC
426  56442  STR AL9 -- 442
427  70427  JP NO IR

430  50000  INPUT
431  30000  CLR B
432  00077  LYQ -- A
433  27456  SUBT A=(Y+B)
434  61447  JP A=0
435  33007  B=7 SKIP
436  64432  JP
437  20025  ADD A+25

440  60422  JP A≠0
441  06455  ENT A=455
442  76___  STR A
443  06422  ENT A=422
444  20001  ADD A+1
445  64426  JP
446  07777  MASK
447  12446  ENT Q = MASK

450  02455  L(Y)Q -- A
451  45003  LSH A3
452  15000  SET A_N for BN=1
453  76455  STR A -- 455
454  64427  JP
455  _____  WORD ASSY
456  00037  CODE 0
457  00052  CODE 1

460  00074  CODE 2
461  00070  CODE 3
462  00064  CODE 4
463  00062  CODE 5
464  00066  CODE 6
465  00072  CODE 7
466  66467  STOP
467  04010  ENT A=10

470  76650  STR A -- 650
471  10000  CLR Q
472  30001  ENT B=1
473  75000  STR Q -- (0+B)
474  07000  ENT A=(0+B)
475  62502  JP A NEG
476  36641  RJP CK 0
477  33377  B=377 SKIP

500  64473  JP
501  64525  JP
502  05000  ENT A=B
503  56510  STR AL9 -- 510
504  75000  STR Q -- (0+B)
505  07000  ENT A=(0+B)
506  62513  JP A NEG
507  36641  RJP CK 0
510  06___  ENT A=ERROR CONTENTS
511  64515  JP
512  66513  STOP
513  33377  B=377 SKIP
514  64504  JP

515  66515  STOP
516  05777  ENT A=ERROR ADDRESS
            (LAST)
517  74652  STR Q -- 652
520  12510  ENT Q=ERROR ADDRESS
            (FIRST 1)
521  66522  STOP
522  12652  ENT Q=652
523  33400  B=400 SKIP
524  64473  JP
525  30001  ENT B=1
526  55000  COMPLEMENT Q
527  75000  STR Q -- (0+B)

530  07000  ENT A=(0+B)
531  62533  JP A NEG
532  64540  JP
533  57000  COMPLEMENT A
534  36641  RJP CK O
535  33377  B=377 SKIP
536  64527  JP
537  64565  JP

540  05000  ENT A=B
541  56550  STR AL9 -- 550
542  75000  STR Q -- (0+B)
543  07000  ENT A -- (0+B)
544  62546  JP A NEG
545  64553  JP
546  57000  COMPLEMENT A
547  36641  RJP CK O

550  06---  ENT A=ERROR CONTENTS
551  64555  JP
552  66553  STOP
553  33377  B=377 SKIP
554  64542  JP
555  66556  STOP
556  05777  ENT A=ERROR ADDRESS
            (LAST)
557  74652  STR Q -- 652
560  12550  ENT Q=ERROR ADDRESS
            (FIRST)
561  66562  STOP
562  12652  ENT Q=652
563  33400  B=400 SKIP
564  64527  JP
565  04007  ENT A=7
566  76651  STR A -- REPEAT INDEX
567  30001  ENT B=1

570  75000  STR Q -- (0+B)
571  55000  COMPLEMENT Q
572  33377  B=377 SKIP
573  64570  JP
574  30001  ENT B=1
575  07000  ENT A=(0+B)
576  57000  COMPLEMENT A
577  77000  STR A -- (0+B)

600  33377  B=377 SKIP
601  64575  JP
602  06651  ENT A=REPEAT INDEX
603  24001  SUBT A-1
```

```
604   76651   STR A -- REPEAT INDEX
605   61607   JP A=0
606   64574   JP
607   30001   ENT B=1
610   07000   ENT A=(0+B)
611   62630   JP A NEG
612   60630   JP A≠ 0
613   33377   B=377 SKIP
614   07000   ENT A=(0+B)
615   62617   JP A NEG
616   64624   JP
617   57000   COMPLEMENT A
620   36641   JP CK O
621   33376   B=376 SKIP
622   64610   JP
623   64654   JP
624   13776   ENT Q=(B-1)
625   66626   STOP
626   11000   ENT Q=B
627   66621   STOP - JP
```

See figures 6-3 thru 6-4 for flow diagrams.

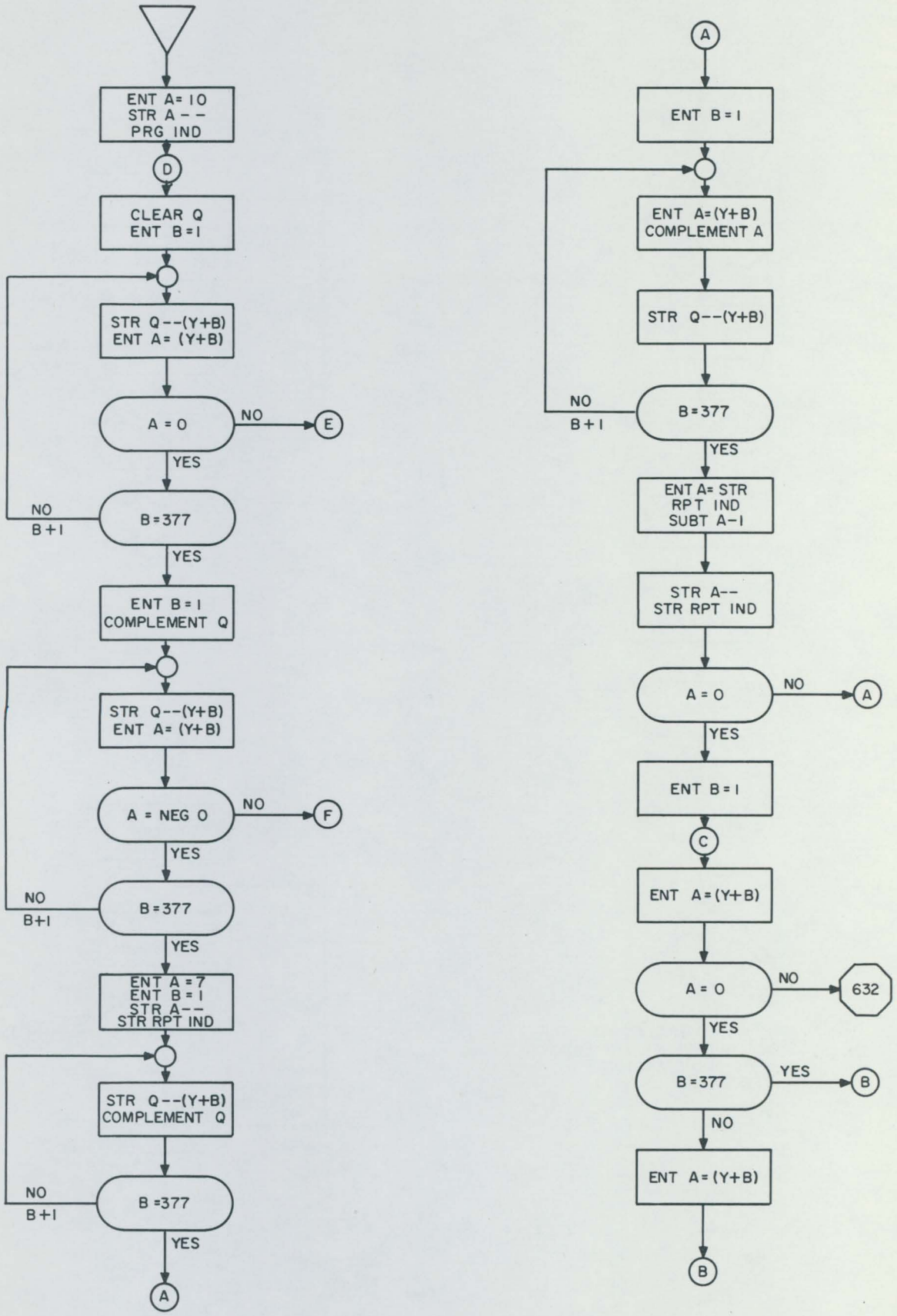


Figure 6-3. Brainwash #2 Flow Diagram - I

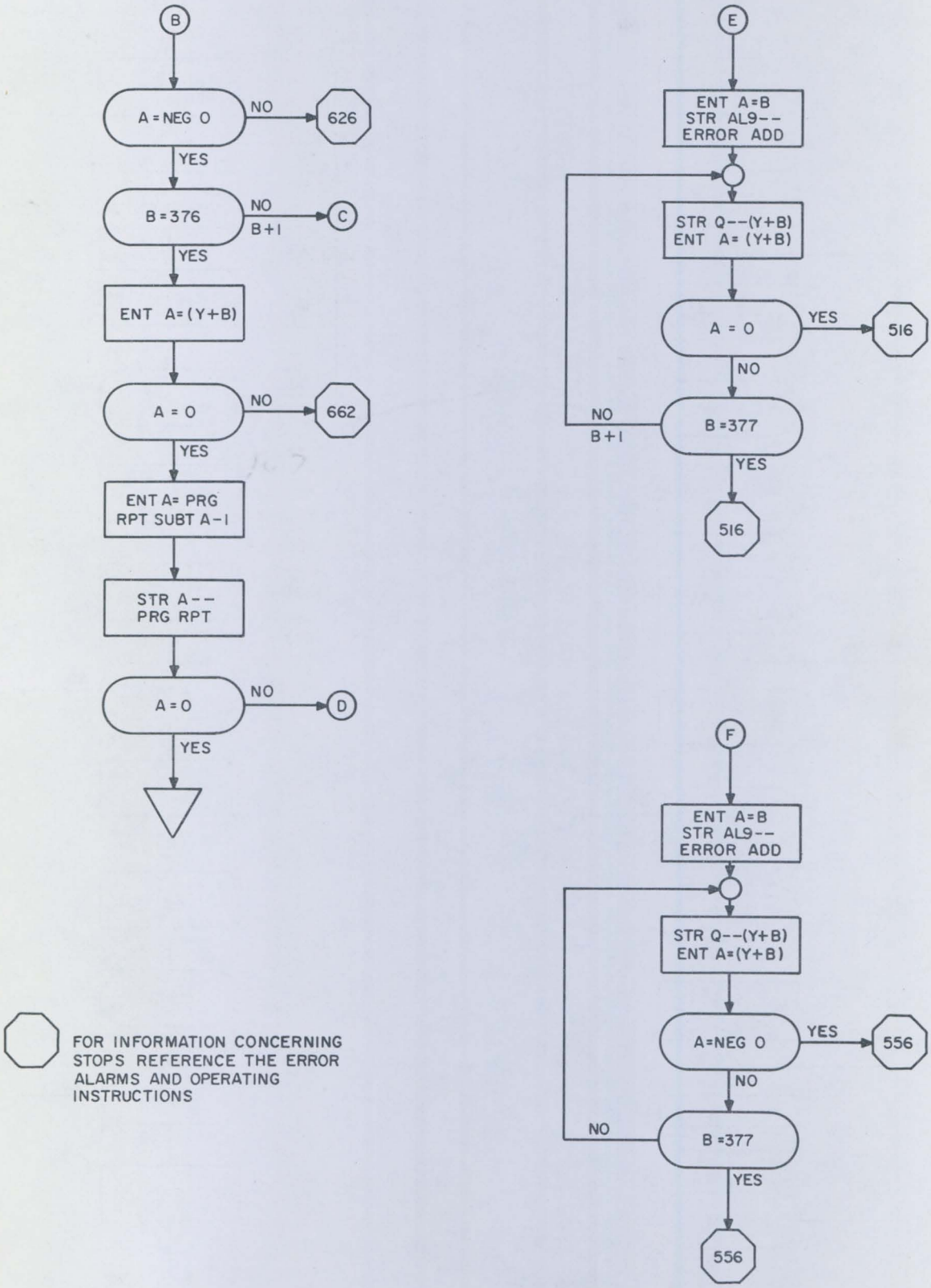


Figure 6-4. Brainwash #2 Flow Diagram - II

6-12. COMMAND TESTS.

a. DESCRIPTION. - The Command Test checks all function codes possible with the exception of 50, 51, 53, 54, 56, 70, 71, and 72 instructions. These instructions are used in the Load Routine and/or Input/Output test, both of which should have been run before this test.

This test is loaded with the standard procedure listed at beginning. The starting address of the Command routine is 067. The test will cycle ten times and stop at 067 unless an error occurs. The program may be recycled from here by enabling High Speed.

b. ALARMS and/or REMARKS. - If the Trainer stops, check the address in the P register and the reference table containing stop addresses and their meanings. Unless otherwise specified after examination of error, the program will continue to cycle after depressing High Speed.

6-13. FLICK (Flex I/O Check).

a. DESCRIPTION - This program checks the capabilities of the Flexowriter as an input/output device. The program is loaded by the standard procedure. The starting address is at 067. A tape called Flex codes must be positioned in the Reader on the first frame (01) and read in when the Flexowriter goes to input.

The Trainer will stop at address 117 at this time; turn on the punch and feed a small leader. Enable High Speed and the test message will output; the Trainer will stop at address 144; feed a small trailer, turn off punch, tear off and position message in reader. Enable High Speed and the test message will be checked by the Trainer after which the Trainer will stop at address 117, if it is desired to rerun test, reference the beginning of this paragraph.

Test Message:

8+7+1+0 = 9+6+2-1
THIS IS A ROUTINE MADE TO INSURE THAT A FAULT IN THE FLEX WILL NOT OCCUR IT CHECKS ALL THE LETTERS LIKE J, Q, AND Z ALSO THE NUMBERS LIKE 5, 4, AND 3 IN CASE YOU SHOULD NOTE THAT THE POEM TENDS TO VARY RUN IT AGAIN, AS THIS TEST IS HAIRY

b. Program characteristics are as follows:

The Select Stop key must be set for error display. The Trainer will stop at address:

114: Q = Code read
A = Actual code
To resume enable High Speed

162: At this time a subtract has taken place, the received message minus the actual message in the A register. To continue, enable High Speed, or test may be restarted at the beginning of output test message, address 117.

DS 4596 UDT COMMAND

022	66023	STOP
023	61022	JP A=O
024	10050	ENT Q=50
025	53000	EXT. FUNC.
026	56042	STR AL9 -- 042
027	70027	JP NO. IR
030	50000	INPUT
031	30000	CLR B
032	00077	LYQ -- A
033	27056	SUBT A-(Y+B)
034	61047	JP A=O
035	33007	B=7 SKIP
036	64032	JP
037	20025	ADD A+25
040	60022	JP A≠O
041	06055	ENT A=WORD
042	76	STR A
043	06042	ENT A=STR ADDRESS
044	20001	ADD A+1
045	64026	JP
046	07777	MASK
047	12046	ENT Q=MASK
050	02055	L(Y) Q -- A
051	45003	LSH A 3X
052	15000	SET A_N for B_N=1
053	76055	STR A -- WORD
054	64027	JP
055		WORD
056	00037	CODE O
057	00052	CODE 1
060	00074	CODE 2
061	00070	CODE 3
062	00064	CODE 4

```
063     00062   CODE 5
064     00066   CODE 6
065     00072   CODE 7
066 --> 66067   STOP
067     04010   ENT A=10

070     76455   STR A
071     06442   ENT A=O-O
072     60074   JP A≠O
073     61075   JP A=0
074     66075   STOP
075     57000   COMPLEMENT A
076     62100   JP A NEG
077     66100   STOP

100     12441   ENT Q=7-7
01      63103   JP Q NEG
02      66103   STOP
03      61105   JP A=0
04      60106   JP A≠0
05      66106   STOP
06      55000   COMPLEMENT Q
07      63111   JP Q NEG

110     64112   JP
11      66112   STOP
12      04001   ENT A=1
13      62115   JP A NEG
14      60116   JP A≠0
15      66116   STOP
16      65120   SEL JUMP
17      64122   JP

120     67122   SEL STOP
21      66122   STOP
22      45016   LSH A 16X
23      62125   JP A NEG
24      66125   STOP
25      45001   LSH A 1X
26      60130   JP A≠0
27      66130   STOP

130     42001   RSH AQ 1X
31      60134   JP A≠0
32      63134   JP Q NEG
33      66134   STOP
34      40016   RSH Q 16X
35      63137   JP Q NEG
36      66137   STOP
37      46001   LSH AQ 1X

140     60142   JP A≠0
41      66142   STOP
42      44016   LSH Q 16X
43      63147   JP Q NEG
44      40016   RSH Q 16X
45      45016   LSH A 16X
46      62150   JP A NEG
47      66150   STOP

150     41016   RSH A 16X
51      36427   RJP CK O
52      66153   STOP
53      46017   LSH AQ 17X
54      61156   JP A=0
55      66156   STOP
56      63160   JP Q NEG
57      66160   STOP

160     46016   LSH AQ 16X
61      62163   JP A NEG
62      63164   JP Q NEG
63      66164   STOP
64      42016   RSH AQ 16X
65      61167   JP A=0
66      66167   STOP
67      63171   JP Q NEG

170     66171   STOP
71      44017   LSH Q 17X
72      63174   JP Q NEG
73      66174   STOP
74      42017   RSH AQ 17X
75      63177   JP Q NEG
76      64200   JP
77      66200   STOP

200     30002   ENT B=Z
01      31001   B=0 SKIP
02      64204   JP
03      66204   STOP
04      31001   B=0 SKIP
05      66206   STOP
06      30677   ENT B=677
07      35443   STR B

210     30000   CLR B
11      32443   ENT B
12      33700   B=700 SKIP
13      64215   JP
14      66215   STOP
15      33700   B=700 SKIP
16      66217   STOP
17      30007   ENT B=7

220     12441   ENT Q=7-7
21      04777   ENT A=00777
22      46006   LSH A 6X
23      36427   RJP CK O
24      66225   STOP
25      04000   CLR A
26      05770   ENT A=00770+B
27      46006   LSH AQ 6X

230     36427   RJP CK O
31      66232   STOP
32      04000   CLR A
33      07432   ENT A=(Y+B)
34      36427   RJP CK O
35      66236   STOP
36      10000   CLR Q
37      04000   CLR A

240     10077   ENT Q=00077
41      46017   LSH AQ 17X
42      11770   ENT Q=00770+B
43      44006   LSH Q 6X
44      46011   LSH AQ 11X
45      36427   RJP CK O
46      66247   STOP
47      04000   CLR A

250     13432   ENT Q=(Y+B)
51      46017   LSH AQ 17X
```

```
 52   36427   RJP CK O
 53   66254   STOP
 54   04000   CLR A
 55   55000   COMPL Q
 56   00077   L(77) Q -- A
 57   42006   RSH AQ 6X

260   01770   L(770+B) Q -- A
 61   46006   LSH AQ 6X
 62   36427   RJP CK O
 63   66264   STOP
 64   04000   CLR A
 65   12441   ENT Q=(Y)
 66   02441   L(Y) Q -- A
 67   36427   RJP CK O

270   66271   STOP
 71   04000   CLR A
 72   03432   L(Y+B) Q -- A
 73   36427   RJP CK O
 74   66275   STOP
 75   04000   CLR A
 76   10000   CLR Q
 77   14077   SET A_N FOR Y_N =1

300   42006   RSH AQ 6X
 01   15770   SET A_N FOR Y_N =1
 02   46006   LSH AQ 6X
 03   36427   RJP CK O
 04   66305   STOP
 05   04000   CLR A
 06   16441   SET A_N FOR Y_N=1
 07   36427   RJP CK O

310   66311   STOP
 11   04000   CLR A
 12   17432   SET A_N FOR Y_N=1
 13   36427   RJP CK O
 14   66315   STOP
 15   06442   ENT A=O --O
 16   61320   JP A=O
 17   66320   STOP

320   20077   ADD A+77
 21   45011   LSH A 11X
 22   62324   JP A NEG
 23   66324   STOP
 24   21770   ADD A+777
 25   61327   JP A = 0
 26   66327   STOP
 27   22444   ADD A+(Y)

330   41010   RSH A 10X
 31   23436   ADD A+(Y+B)
 32   61334   JP A=0
 33   66334   STOP
 34   57000   COMPLEMENT A
 35   26445   SUBT A - 77770
 36   60340   JP A≠0
 37   66340   STOP

340   25000   SUBT A-B
 41   61343   JP A=0
 42   66343   STOP
 43   57000   COMPLEMENT A
 44   27436   SUBT A - 77770
 45   60347   JP A≠0
 46   66347   STOP
 47   24007   SUBT A - 00007

350   61352   JP A=0
 51   66352   STOP
 52   12446   ENT Q=55372
 53   43447   MULT Q(Y)
 54   26450   SUBT
 55   61357   JP A=0
 56   66357   STOP
 57   46017   LSH AQ 17X

360   26451   SUBT
 61   61363   JP A=0
 62   66363   STOP
 63   06450   ENT A=25630
 64   12451   ENT Q=76366
 65   47452   DIVIDE
 66   26453   SUBT
 67   61371   JP A=0

370   66371   STOP
 71   46017   LSH AQ 17X
 72   26454   SUBT
 73   61375   JP A=0
 74   66375   STOP
 75   12441   ENT Q=7-7
 76   74456   STR Q
 77   06456   ENT A

400   36427   RJP CK O
 01   66402   STOP
 02   57000   COMPLEMENT A
 03   76456   STR A
 04   12456   ENT Q
 05   46017   LSH AQ 17X
 06   61410   JP A=0
 07   66410   STOP

410   55000   COMPLEMENT Q
 11   75447   STR Q
 12   06456   ENT A
 13   36427   RJP CK O
 14   66415   STOP
 15   57000   COMPLEMENT A
 16   77447   STR A
 17   12456   ENT Q

420   46017   LSH AQ 17X
 21   61423   JP A = 0
 22   66423   STOP
 23   06455   ENT A = INDEX
 24   24001   SUBT A-1
 25   61457   JP A=0
 26   64070   JP
 27   64000   JP

430   57000   COMPLEMENT A
 31   60437   JP A≠0
 32   76456   STR A
 33   06427   ENT A
 34   20001   ADD A+1
 35   76427   STR A
 36   06456   ENT A
 37   57000   COMPLEMENT A
```

440	64427	JP	450	25630
41	77777		51	76366
42	00000		52	35555
43	00000		53	04456
44	03400		54	56750
45	77770		55	00000
46	55372		56	00000
47	36527		57	64066

UDT COMMAND

STOP ADD	REG IND	INST FAIL	
75	A=00000	06	If A≠00000 then ENT A=00000 failed
		60	If A=00000 then JP A≠0 jumped or
		61	JP A=0 failed to jump. If A=00000 restart program at 071 and OP STEP to see which jump failed.
100	A=77777	57	A≠77777 then COMPLEMENT A instruction failed
		62	A=77777 then JP NEG instruction failed
103	Q=77777	12	Q≠77777 then ENT Q=77777 failed
		63	Q=77777 then JP Q NEG instruction failed
106	A=77777	61	JP A=0 jumped or
		60	JP A≠0 failed to jump. Restart program at 103 and OP STEP to determine which jump failed.
112	Q=00000	55	Q≠00000 then COMPLEMENT Q instruction failed
			Q=00000 then JP Q NEG instruction failed
116	A=00001	04	A≠00001 then ENT A instruction failed
		62	A=00001 then JP A NEG jumped or
			JP A≠0 failed to jump. Restart program at 112 and OP STEP to determine which jump failed.
122	SEL STOP must be set	65 67	If SEL STOP is set then SEL JUMP failed or SEL STOP failed. If the stop is a SEL STOP then this is a legal stop and just depress HI SPEED.
125	A=40000	45	A≠40000 then LSH A 16X failed
130	A=00001	45	End around feature LSH A failed
134	A=00000 Q = 40000	42	A≠00000 then RSH AQ failed completely Q≠40000 then RSH AQ transfer A -- Q failed
137	Q=77777	40	Q≠77777 then RSH Q sign extension failed
142	A=00001 Q=77776	46	A≠00001 then LSH AQ transfer Q -- A failed Q≠77776 then end around feature LSH AQ failed
150	A=40000 Q=00000	44 40	Q≠00000 then LSH Q End around failed or RSH Q sign extension positive failed. Enter A and Q as specified by STOP 142, restart program 142 and OP STEP to determine failure.
153	A=77777	41	A≠77777 then RSH A with sign extension failed
156 160	A=00000 Q=77777	46	If either register is wrong then LSH AQ failed End around feature is checked here.
164	A=67777 Q=40000	46	If either register wrong then LSH AQ failed.
167 170	A=00000 Q=77777	42	If either register wrong then RSH AQ failed.
174	Q=77777	44	Q≠77777 then LSH Q end around failed

200	Q=00000	42	Q≠00000 then RSH AQ failed
204	B=1	30	B≠1 then ENT B failed
		31	B=1 then B=0 SKIP failed
206	B=0	30	B≠0 then ENT B failed
		31	B=0 then B=0 SKIP failed
215	B=700	35	Address 443 = 677. If not STR B failed.
		32	If B≠700 then ENT B failed. Restart at 206 and OP STEP to determine
		33	If B=700 then B=Y SKIP failed.
217	B=701	33	Check same as for 215 except B=701.
225	A=77777	04	If upper three character A wrong then ENT A failed.
232	A=77777	05	If upper two character A wrong then ENT A failed. If middle character wrong the B modify failed.
236	A=77777	07	A≠77777 then ENT A failed or A=76456 then B modify failed
247	A=77777	10	If upper two character wrong then ENT Q=Y failed.
		11	If lower three characters wrong then ENT Q=Y+B failed.
254	A=77777	13	A≠77777 then ENT Q=(Y+B) failed.
264	A=77777	01	If upper three characters wrong then LP Y+B Q-- A failed.
		00	If lower two character wrong then LP YQ-- A failed.
271	A=77777	02	A≠77777 then LP (Y) Q -- A failed
275	A=77777	03	A≠77777 then LP (Y+B) Q -- A failed
305	A=77777	15	If upper three characters wrong then Set A_N for $Y+B_N=1$ failed.
		14	If lower two characters wrong then Set A_N for $Y_N=1$ failed.
311	A=77777	16	A≠77777 then Set A_N for (Y)=1 failed
315	A=77777	17	A≠77777 then Set A_N for (Y+B)=1 failed
320	A=00000	06	If A≠00000 then ENT A = (Y) failed
324	A=77000	20	A≠77000 then ENT A = Y failed
327	A=00000	21	A≠00000 then ENT A=Y+B or ENT A=Y failed. Restart
		20	program at 315 and OP STEP to determine which instruction failed.
334	A=00000	22	A≠00000 then ENT A=(Y) or ENT A=(Y+B) failed.
		23	Clear A, restart program at 327 and OP STEP to determine failure.
340	A=00007	26	A≠00007 then SUBT A-(Y) failed
343	A=00000	25	A≠00000 either SUBT A-(Y) or SUBT A-B failed.
		26	Restart program at 334 and OP STEP to determine failure.
347	A=00007	27	A≠00007 then SUBT A- (Y+B) failed
352	A=00000	24	A≠00000 either SUBT A-Y or SUBT A- (Y+B) failed. Restart at 343 and OP STEP to determine failure.
357	A=00000	43	A≠00000 the MULT Q(Y) failed. Upper digits were wrong.
363	A=00000	43	same as for 357 except lower digits wrong. Restart at 352 and after executing MULT instruction A=25630 Q=76366.
371	A=00000	47	A≠00000 DIVIDE failed. Remainder wrong.
375	A=00000	47	A≠00000 DIVIDE failed. Quotient wrong. Restart at 363 and after executing DIVIDE A=04456 Q=56750.
402	A=77777	74	Address 456≠77777 then STR Q failed
		06	Address 456=77777 then ENT A failed

410	A=00000	76	Address 456≠00000 then STR A failed
		12	Address 456=00000 then ENT Q failed
415	A=77777	75	Address 456≠00000 then STR Q failed
		06	Address 456=00000 then ENT A failed
423	A=00000	77	Address 456≠77777 then STR A failed
		12	Address 456=77777 then ENT Q failed

See figures 6-5 thru 6-9 for flow diagrams.

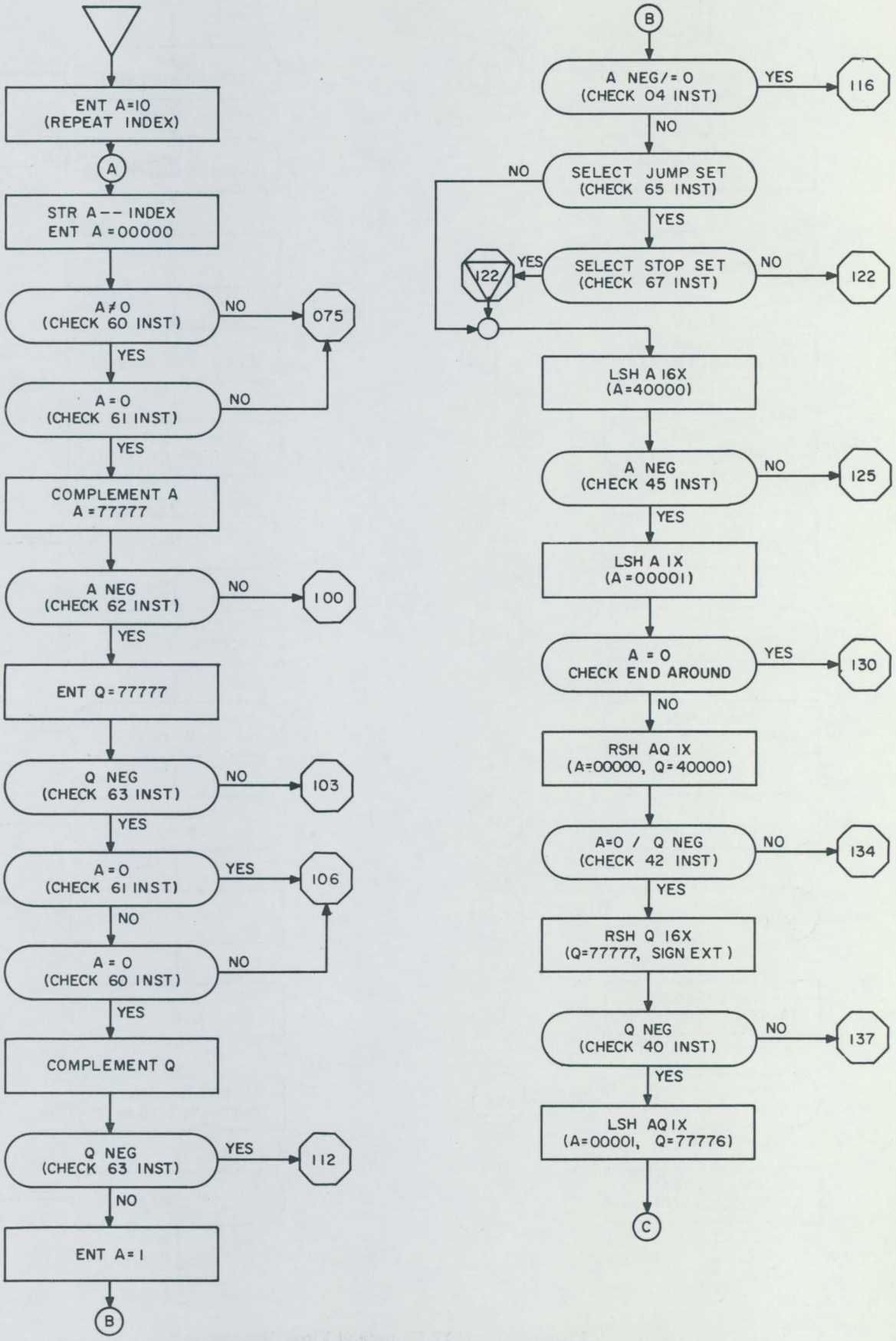
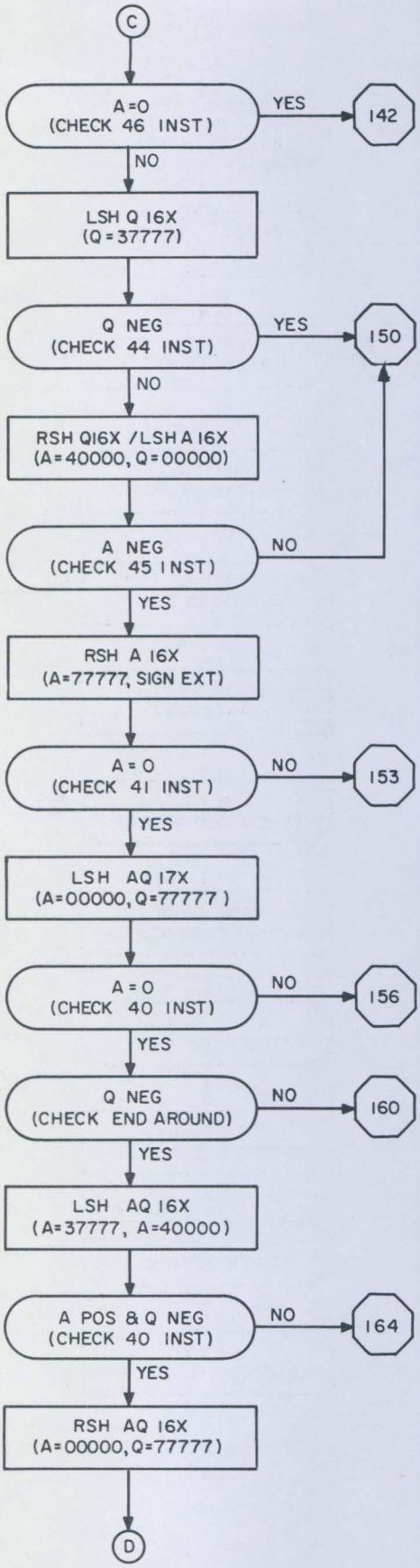


Figure 6-5. UDT Command Flow Diagram - I



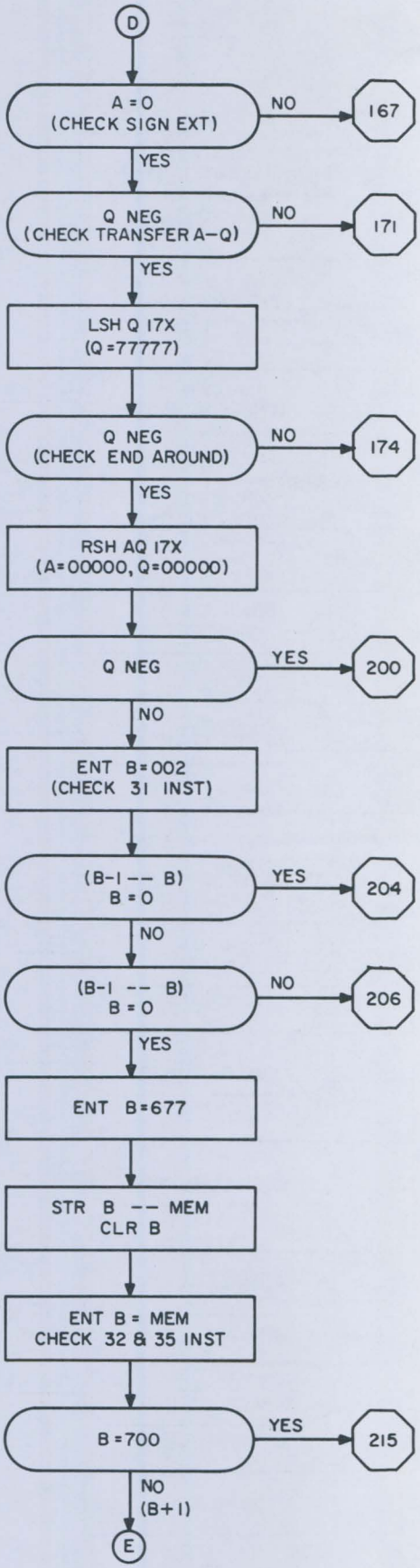


Figure 6-6. UDT Command Flow Diagram - II

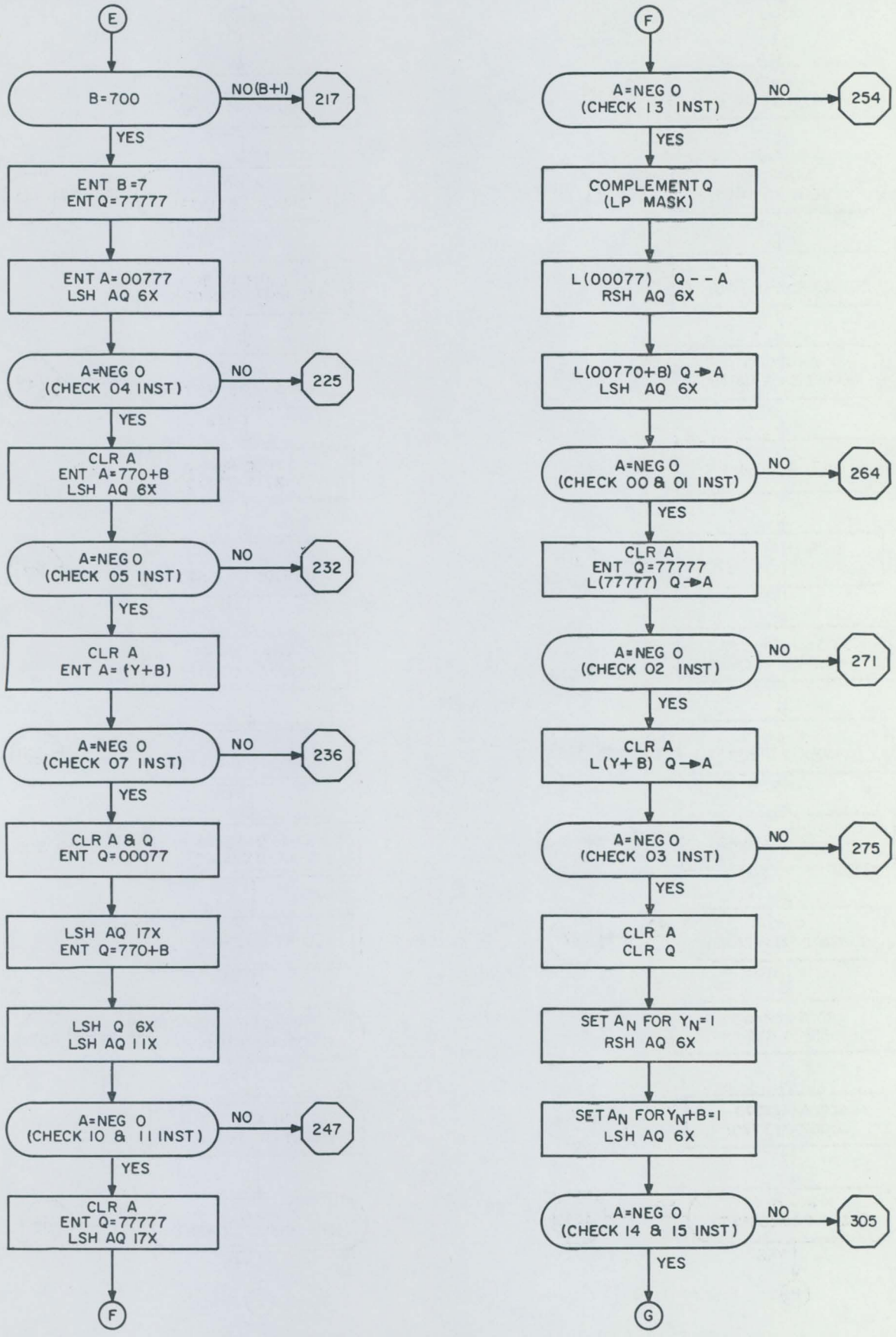


Figure 6-7. UDT Command Flow Diagram - III

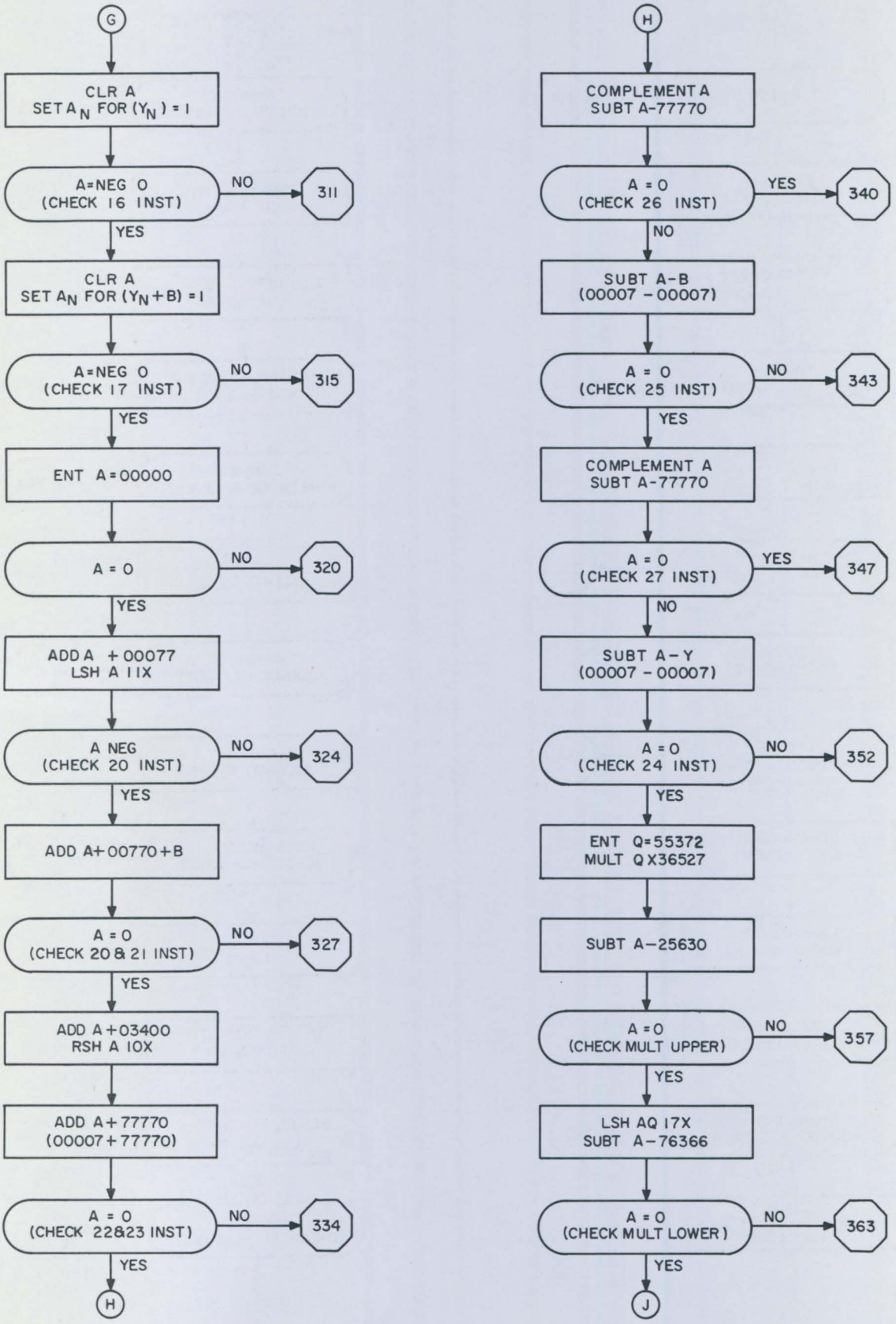


Figure 6-8. UDT Command Flow Diagram - IV

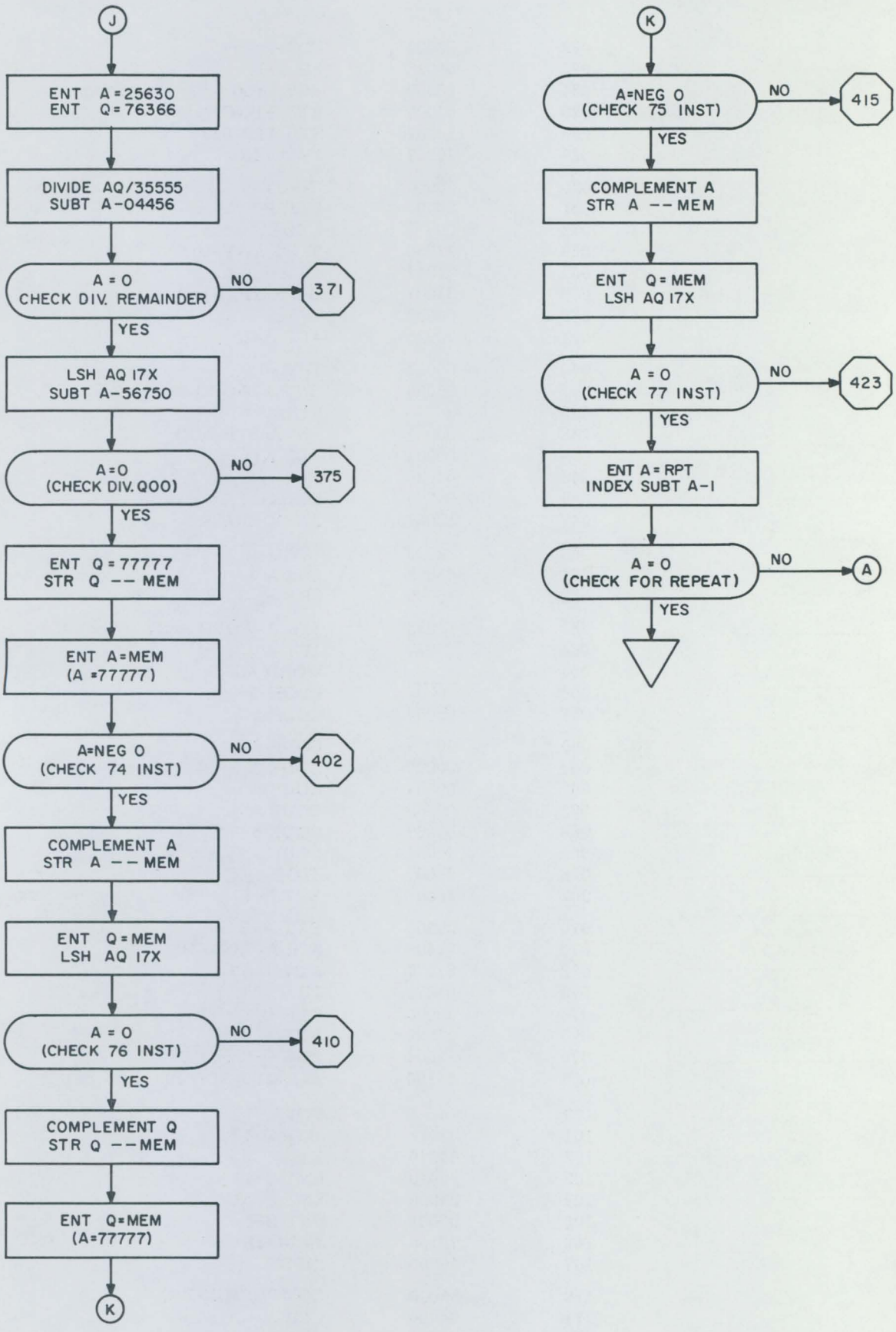


Figure 6-9. UDT Command Flow Diagram - V

UDT

FLICK

```
022     66023     STOP
023     61022     JP A=0
024     10050     ENT Q=50
025     53000     EXT FUNC
026     56042     STR AL9-042
027     70027     JP NO IR

030     50000     INPUT
031     3000      CLR B
032     00077     LYQ-A
033     27056     SUBT A-(Y+B)
034     61047     SP A=0
035     33007     B=7 SLIP
036     64032     JP
037     20025     ADD A+25

040     60022     JP A≠0
041     06055     ENT A=WORD
042     76        STRA
043     06042     ENTA=STR ADD.
044     20001     ADD A+1
045     64026     JP
046     07777     MASK
047     12046     ENT Q=MASK

050     02055     L(Y)Q-A
051     45003     LSH A 3
052     15000     SET A_N FOR B_N = 1

053     76055     STR A-WORD ASSY
054     64027     JP
055               WORD ASSY
056     00037     CODE O
057     00052     CODE 1

060     00074     CODE 2
061     00070     CODE 3
062     00064     CODE 4
063     00062     CODE 5
064     00066     CODE 6
065     00062     CODE 7
066     66067     STOP
067     30001     ENT B=1

070     05000     ENT A=B
071     77400     STR A- 400 + B)
072     33077     SKIP B=77
073     64070     JP
074     30000     CLR B
075     04004     ENT A=04
076     13364     ENG Q=(364 + B)
077     54100     STR QL9-ADD 100

100     76        STR A
101     33012     SKIP B=12
102     64076     JP
103     10040     ENT Q=40
104     53000     EXT FUNC
105     30001     ENT B=1
106     70106     JP NO IR
107     50000     INPUT

110     55000     COMPLEMENT Q
111     04000     CLR A
112     03400     L (400 + B) Q-A
113     60174     JP A≠0
```

```
114     33074     SKIP B = 74
115     64106     JP
116     66117     STOP
117     10020     ENT Q=20

120     53000     EXT FUNC
121     06203     ENT A -(203)
122     76204     STR A-(204)
123     56130     STR AL9
124     20001     ADD A+1
125     56131     STR AL9
126     20001     ADD A+1
127     76204     STR A-(204)

130     06___     ENT A
131     12___     ENT Q
132     61143     JP A=0
133     30000     CLR B
134     46006     LSH AQ6
135     71135     JP NO O.R.
136     51000     OUTPUT
137     33004     SKIP B=4

140     64134     JP
141     06204     ENT A=204
142     64123     JP
143     66144     STOP
144     06203     ENT A-(203)
145     76204     STR A-(204)
146     56160     STR AL9
147     10040     ENT Q=40

150     53000     EXT FUNC
151     30000     CLR B
152     46006     LSH AQ6
153     70153     SP NO IR
154     50000     INPUT
155     33004     SKIP B=4
156     64152     JP
157     30000     CLR B

160     26___     SUBT
161     60177     JP A≠0
162     06204     ENT A=204
163     20001     ADD A+1
164     56160     STR AL9
165     76204     STR A-(204)
166     24363     SUBT A - 363
167     61116     JP A=0

170     46017     LSH AQ 17
171     33001     SKIP B=1
172     64160     JP
173     64151     JP
174     55000     COMPLEMENT Q
175     07400     ENT A=(400 + B)
176     67114     STOP (SEL)
177     67162     STOP (SEL)

200     00000
201     00000
202     00000
203     00205     ADDRESS
204     00205     WORKING ADDRESS
205     57605     MESSAGE
206     45254
207     72543
```

	UDT
210	74433
211	54665
212	47456
213	52454
214	7-105
215	14240
216	41424
217	04300
220	41203
221	34011
222	40620
223	04073
224	02220
225	04010
226	30414
227	06243
230	41220
231	45010
232	53001
233	04300
234	42630
235	34110
236	10414
237	06040
240	10520
241	04261
242	12027
243	04311
244	41111
245	04060
246	30104
247	03161
250	63412
251	45140
252	10416
253	05201
254	63624
255	04301
256	11104
257	01052
260	00411
261	20010
262	12012
263	24041
264	11436
265	20043
266	25746
267	47355
270	74647
271	04300
272	62204
273	21453
274	01124
275	03040
276	10520
277	04063

300	40723
301	20122
302	40411
303	14362
304	00462
305	57466
306	44647
307	04300
310	62204
311	57704
312	54714
313	06041
314	63024
315	20042
316	50334
317	04240
320	50334
321	11220
322	40603
323	01200
324	40105
325	30010
326	40105
327	20041
330	50320
331	07040
332	12006
333	22240
334	40103
335	04173
336	01225
337	45123
340	40604
341	14010
342	43013
343	30140
344	65746
345	47043
346	02404
347	01051
350	42404
351	01202
352	40104
353	14240
354	40530
355	14122
356	56161
357	61616
360	15050
361	50505
362	05742
363	00000
364	00410
365	00440
366	00441
367	00443

370	00453
371	00455
372	00463
373	00465
374	00467
375	00471
376	00473
377	00000

See figures 6-10 thru 6-11 for flow diagrams.

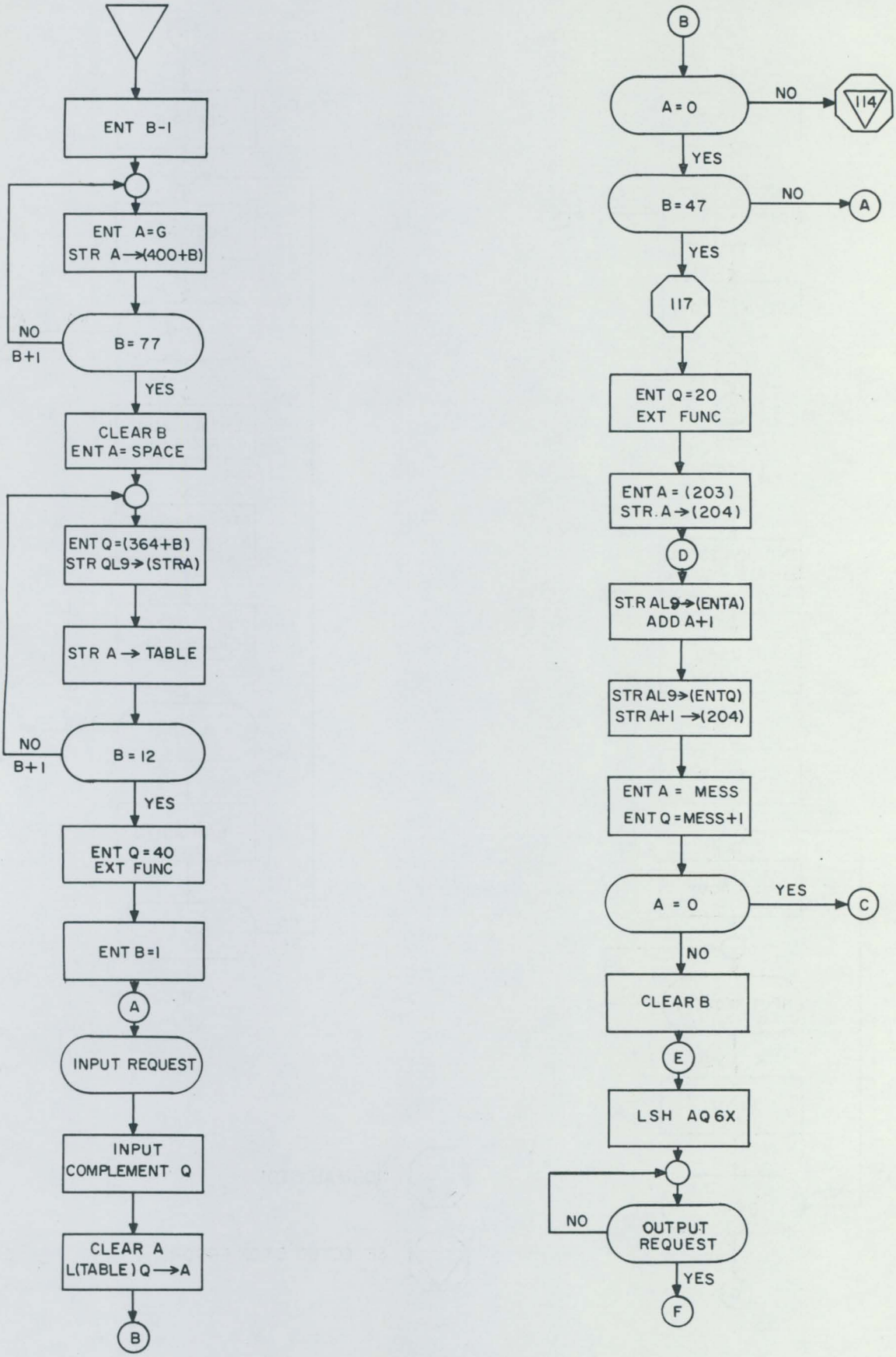


Figure 6-10. Flick Flow Diagram - I

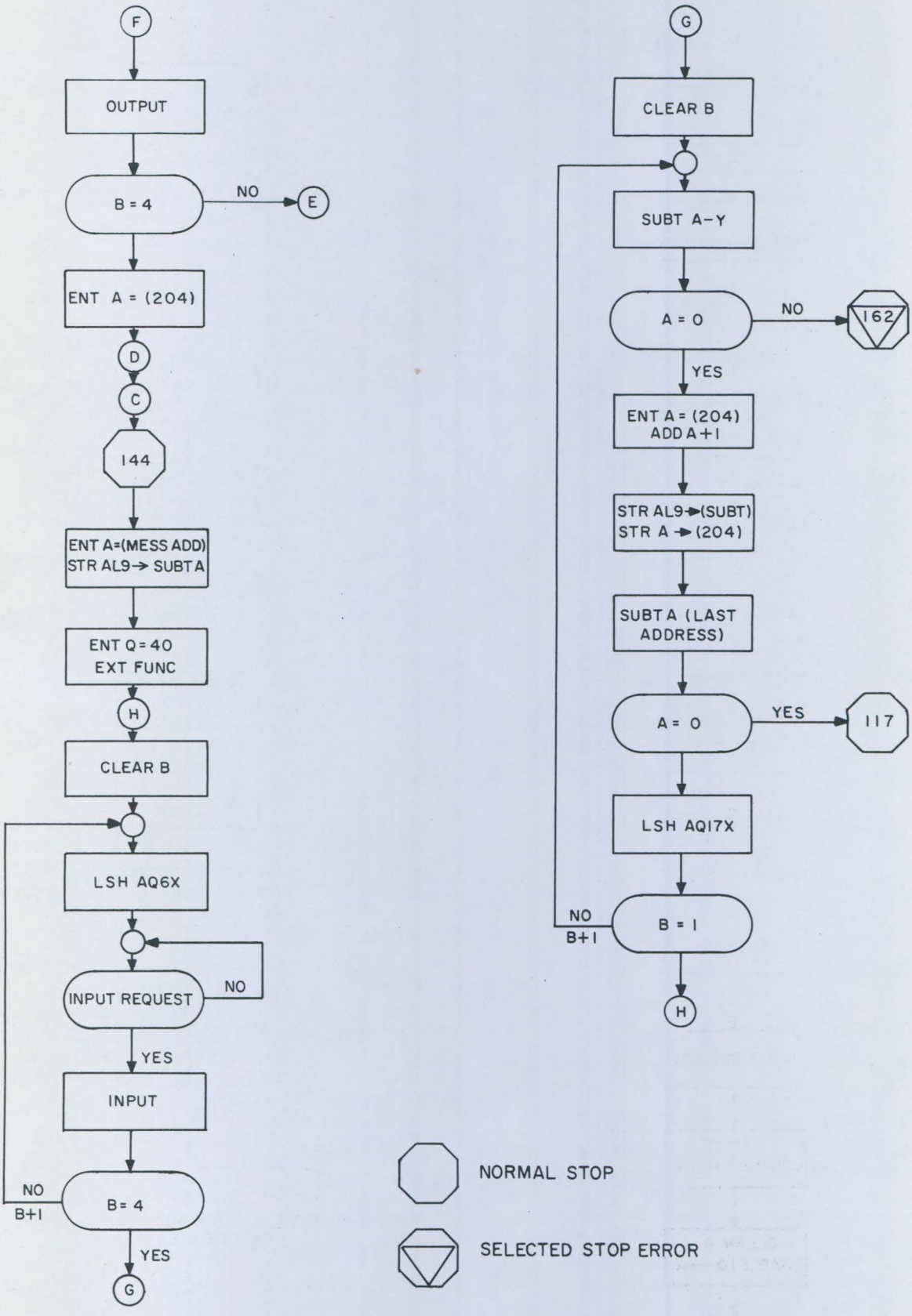


Figure 6-11. Flick Flow Diagram - II

6-14. ENDURANCE RUN.

a. GENERAL. - The endurance run is the Command test, Brainwash #1, and Brainwash #2 continuously being recycled 32,768 times. The following procedure should be followed for running each of these tests.

(1) Command Test

(a) Load the Command tape.

(b) Set P = 070

(c) Set A = 77777.

(d) Operate the HIGH SPEED RUN - OPERATION STEP switch to the up position.

(e) When the test is completed, the Trainer will stop with P = 070. An error shall be indicated by any stop with P $\neq$ 070.

(2) Brainwash #1

(a) Load the Brainwash #1 tape.

(b) Set P = 070

(c) Set A = 77777

(d) Operate the HIGH SPEED RUN - OPERATION STEP switch to the up position.

(e) When the test is completed, the Trainer will stop with P = 070.

(f) Master Clear the Trainer.

(3) Brainwash #2

(a) Load the Brainwash #2 tape.

(b) Set P = 470

(c) Set A = 77777

(d) Operate the HIGH SPEED RUN - OPERATION STEP switch to the up position.

(e) When the test is completed, the Trainer will stop with P = 470.

6-15. REPAIR.

a. GENERAL. - Servicing the Trainer consists mainly of removing and replacing defective card assemblies. Other repairs may consist of replacing defective wiring or components. When attempting any repairs, always make a thorough visual inspection of the chassis, cards, jacks, and associated wiring. When replacing blown fuses, determine the cause of failure before installing a new fuse.

CAUTION

Do not install fuses having a current rating greater than the fuse removed. When making continuity checks with an ohmmeter, observe polarity of the probes so as not to damage any transistors.

When it is determined that a malfunction exists, refer to the functional schematics (Section 5) to draw logical conclusions as to the possible circuit causing the malfunction. Section five will also identify the location of the various test points associated with each of the circuits. When the circuits have been isolated as the possibility of the cause of trouble, manually insert various patterns in the registers that will change the output of these circuits. Monitor the output of each suspected circuit with an oscilloscope until the faulty circuit is found. Replacement of this printed circuit card should restore the Trainer back to normal operation. Components mounted directly on terminal boards should not be ignored as possible defective components.

Once the defective printed circuit card is found, the repair of this circuit can be made by determining the component causing the failure. The card Tester located on the right rear chassis should be utilized along with the schematic diagram of the particular circuit card under test. The card schematics can be found in Section four, Principles of Operation.

The Logic Card tester will apply the normal voltages to the card under test. Note that the INPUT SELECT switch does not allow inputs to be generated into pins one through four, and 15. Referring to the card schematics, note that these pins are used for the applied voltage levels and pin 15 for all circuits is an output.

With the card inserted into the card jack, each pin of the circuit is available for monitoring. Inputs should be applied to each of the input pins of the circuit and the output of the circuit should be monitored. From this test, the component causing the failure should be easily found. However, further checks can be made with an ohmmeter (diodes, transistors, and values of resistance).

During the testing procedure, consideration should be given to the rise and fall time of the circuit under test. Pulse rise and fall times are measured between 10 and 90 percent levels of amplitude; pulse duration time is measured at the 50 percent level of amplitude. Circuits under test will indicate a rise and fall time of approximately 40 nanoseconds (millimicroseconds, 10^{-9} seconds).

The Oscilloscope waveform may indicate overshoot and ringing during operation. These conditions may be disregarded. The tester built into the Trainer chassis is intended for the checking of logic circuit cards only. The special circuits employed by the memory section, the input/output adapter, the indicator drivers, and the input amplifier circuits connected to the control switches should be checked with caution.

b. CIRCUIT BOARD REPAIR. - Replacing of the component should be accomplished with great care to prevent further damage to other components on the card, to the printed circuit, or to the card itself. Following the steps outlined below will help guard against any further damage to the circuit card.

(1) After the defective component is located on the printed circuit card, hold the card up to a bright light to determine the physical connections of both ends (three in the case of a transistor failure).

(2) Cut the defective component from the circuit board.

(3) Using a pencil soldering iron (25 watt) remove each end of the component leads from the printed circuit.

CAUTION

Excessive heat applied to the printed circuit will result in damage to the board. This step of component replacement should be carefully executed.

(4) Ensure the holes from which the leads were removed are clear of excess solder. Clearing of these holes can be accomplished by applying a small amount of heat and inserting a small article through the hole at the same time.

(5) Bend the leads of the replacement component to fit the holes on the board. When inserting the leads of the component, do not force the wire into the board. Inspect for excess solder. Remember to observe polarity during diode replacement.

(6) Solder each lead of the component onto the printed circuit, and cut the excess leads.

(7) Inspect the entire board to determine that none of the printed circuit wiring has been shorted by solder drippings.

(8) Insert the circuit card into the Logic Card tester and test each circuit on the board before using the card in the Trainer.

APPENDIX A
TRAINER ADAPTER

A-1. GENERAL

Communications to and from the Trainer via channel one must pass through an adapter which will properly match the line impedance and supply the necessary voltage levels to the external equipments. The adapter is a separate piece of hardward connected to the cable connectors in the rear of the Trainer and is usually positioned above the control console.

a. SPECIAL CIRCUITS. - In the adapter special circuits (nonstandard logic circuits) are used as line drivers and amplifiers. The line drivers drive the control and data lines to external equipment. The input amplifiers receive their inputs from the control and data line drivers of the external equipment. When the Input amplifier is used on the Data line, it requires a gate signal. When the Input amplifier is used on a control line, no gate signal is used. The output from either of the two line drivers is 0 VDC for a binary "1" and a -12.5 VDC for a binary "0". The output voltages are the inputs to the input amplifiers. The output from the input amplifiers is -3 VDC for a binary "1" and 0 VDC for a binary "0".

(1) GATED INPUT AMPLIFIER. - (See figure A-1.) This circuit is used to receive the Data input from external equipment. The circuit configuration is a ground emitter amplifier using a PNP transistor. The circuit is such that with both inputs disconnected completely, the transistor is in saturation, the base being slightly negative with respect to emitter. The output is approximately 0 VDC since the collector approaches the emitter potential when the transistor conducts saturation current. Under normal operating conditions, with both inputs connected, the transistor remains saturated until the data input rises to 0 VDC and the Gate input signal is 0 VDC. When the Gate signal is -3 VDC, CR5 is enabled keeping the base potential negative with respect to the emitter and keeping the transistor in saturation. In this manner, regardless of what signal appears on the Data line, no information is entered into the Q register. When the command is generated that will allow the gating of data to the Q register, the gate signal will rise to 0 VDC. Circuit CR5 is disabled due to the negative voltage appearing at the base of the transistor from the voltage divider between +15 VDC and -15 VDC. The transistor remains in saturation. Now, the Input Data signal exercises control. When the data input is 0 VDC, the junction of the 4.7K resistor and 9.1K resistor becomes effectively ground. The voltage divider action is such that the base becomes positive with respect to the emitter. The transistor is turned off and the output drops toward -15 VDC from 0 VDC. However,

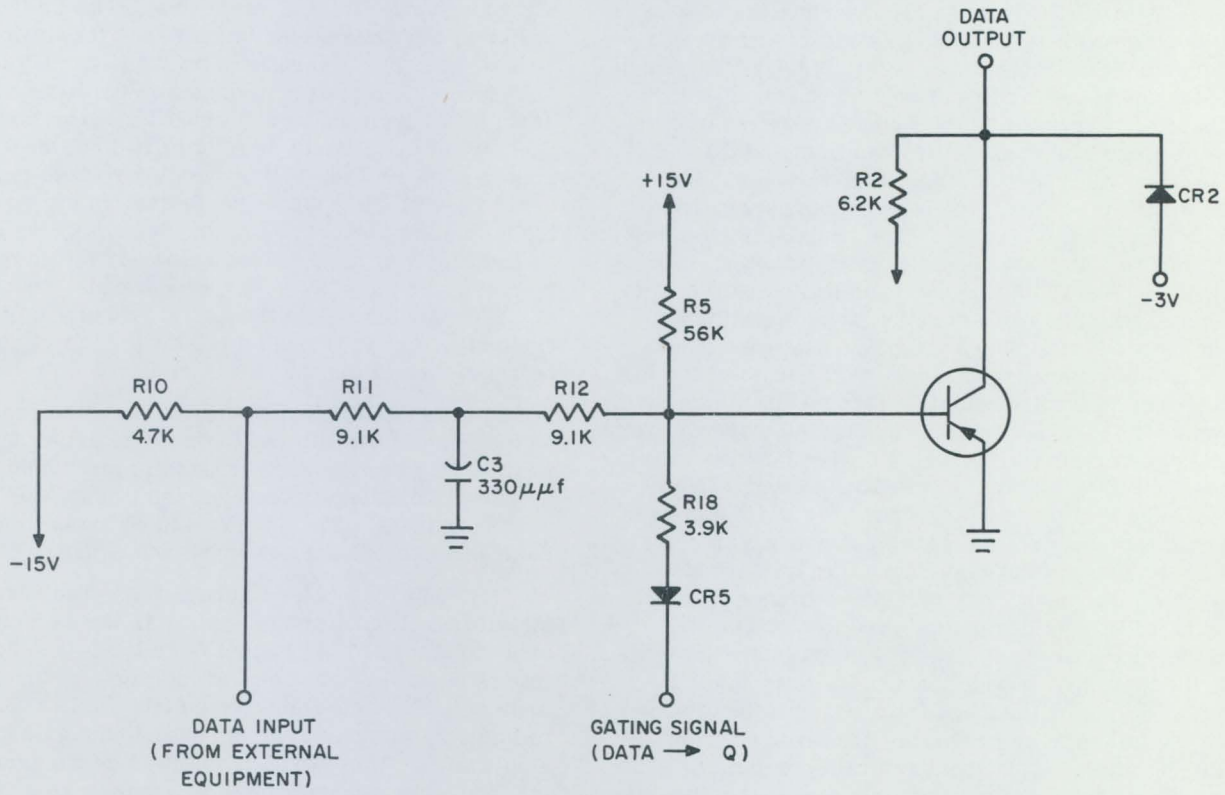


Figure A-1. Input Amplifier

CR2 clamps the output at -3 VDC. When the data input is -12.5 VDC, the base again becomes negative with respect to the emitter and the transistor goes into saturation. The output rises from -3 VDC to 0 VDC. Therefore, when the channel is gated, a data input of a "0" or a "1" is repeated by the output as a "0" or "1" respectively. This is the time when information is entered into the Q register by channel one utilizing the 15 input amplifiers. The input circuit, consisting of the two 9.1K resistors and the 330 micromicrofarad capacitors, forms a filter network which prevents ringing of the Input Step function.

(2) INPUT AMPLIFIER FOR CONTROL SIGNAL. - The input amplifiers in the adapter, used to receive control signals from external equipment are not gated. The circuit is as shown in figure A-1, with the gate signal line open. Under this condition, the transistor is normally in saturation. When a "1" or a "0" appears on the Control line, the transistor switches as it did for the same input on the Data line. Essentially, the circuit operation is the same as for a data amplifier when the channel is gated.

(3) DATA LINE DRIVER. - This driver transmits data to input amplifiers in external equipment. As seen in figure A-2, the circuit consists of one inverter, two transistors in a complementary symmetry configuration, and two transistors used as a current limiting circuit. The over-all circuit performs to provide the drive required by the data input amplifiers in the external equipment. The input to the Data Line driver is 0 VDC or -3 VDC. The input is to an inverter which drives the complementary symmetry output stage. With 0 VDC input to the inverter, the base is held positive with respect to the emitter and the transistor is cut off. The output voltage is -15 VDC which is fed to the base circuits of the output stage. Since the base of both transistors becomes negative with respect to the emitter, Q2 (PNP) conducts and Q3 (NPN) is cut off. When the input to the inverter switches from 0 VDC to -3 VDC, the base becomes negative with respect to the emitter. The transistor attempts to conduct saturation current causing its output to rise toward 0 VDC. With the change in base bias, the output stage switches with Q3 going into conduction and Q2 going to cutoff. When the output stage switches, there is a degenerative feedback to the input of the inverter from the output stage. This feedback effectively prevents the inverter from switching immediately from cutoff to saturation. Instead, a controlled rise time of the Input Step function to the Output circuit is obtained such that the transition time for switching from the cutoff state to the saturation state is greater than 3.0 microseconds, but less than 6.0 microseconds. Now, when the input to the inverter switches from -3 VDC to 0 VDC, the transistor goes toward cutoff. With the change in base bias, the output stage switches with Q2 going into conduction and Q3 going toward cutoff. Here, the feedback from output to input controls the fall time by preventing the inverter from switching immediately from saturation to cutoff. The transition of the fall time approximates that of the rise time. The output from the complementary symmetry circuit varies between 0 VDC on the Data line when Q3 conducts and -12.5 VDC on the Data line when Q2 conducts. The remaining portion of the circuits containing Q4 and Q5 is for current limiting. The arrangement of the circuit is such that Q4 conducts initially. When the current through Q4 increases, Q5 increases because its base potential rises above that of the emitter due to the voltage drop across R6. The current through Q5 produces a voltage drop across R5 which tends to decrease conduction through Q4 by increasing the base bias of Q4. Therefore, if Q4 attempts to conduct heavily, Q5 samples the voltage change across R6 and produces a controlling voltage drop across R5 which limits the current through Q4. The initial surge current required to charge the data lines is obtained from C3. When there is no data on the lines, the input amplifier sees a binary "0", -12.5 VDC, on the line.

(4) CONTROL LINE DRIVER. - This driver transmits control signals to an input amplifier in external equipment. As seen in figure A-3, the circuit consists of six transistors: one inverter, two transistors in a complementary symmetry circuit, two transistors for current limiting, and one transistor used to meet high impedance requirements when the Trainer has its power removed. The circuit is used in the same manner as the Data Line driver with the difference being that the Control Line driver presents a high impedance to the line when power is off to avoid falsely putting a "1" on the line. The functional operation of the Control Line driver is the same as for the Data Line driver with the addition of the extra circuit.

When Trainer power is turned off, Q5 along with the isolation diodes CR2, CR3, and CR4 present a high impedance to the Control line. The Input amplifier in the external equipment sees this high impedance or effective binary zero on the line. CR4, a 10V Zener diode, is incorporated in the base circuit of Q5 to set the turn on and turn off voltage levels of Q5. When the Trainer is first turned on, the -15V supply must rise to -10V before any voltage is applied to the base of Q5. When the Trainer is turned off, the voltage is removed from the base of Q5 as soon as the supply voltage decays below -10V. In this way, transients in the -15V supply do not cause conduction of Q5. This maintains the high impedance on the Control line and prevents a false signal (a "1") from being applied to the line.

b. ADAPTER LOGIC. - The adapter consists of 15 gated input amplifiers, 15 data line drivers, three input amplifiers for control signals, three control line drivers, six indicator drivers, and two standard inverter circuits. The adapter can be logically broken down into two categories: input and output.

(1) INPUT. - (See figure 5-53.) The receipt of all data on channel one is applied as inputs from the cable to the gated input amplifiers (50Y--). Following the normal sequence of events, the external equipment will place the Data word on the cable. This data is immediately available as inputs to the gated input amplifiers. This would be followed by the arrival of the Input Data request which will appear as a "0" volt level input to the 50Y16 input amplifier. The -3 VDC output from 50Y16 will be transmitted to the Trainer

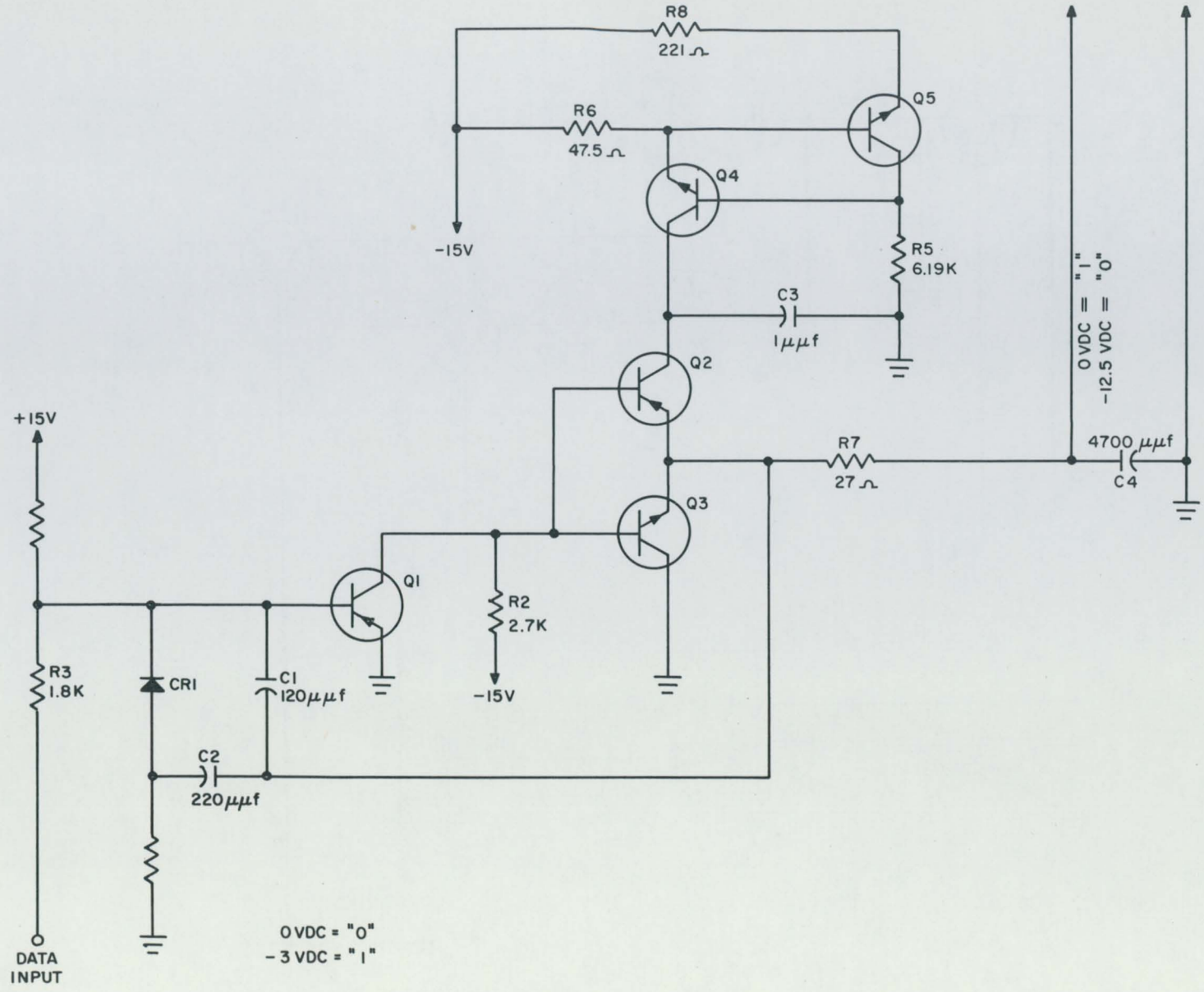


Figure A-2. Data Line Driver

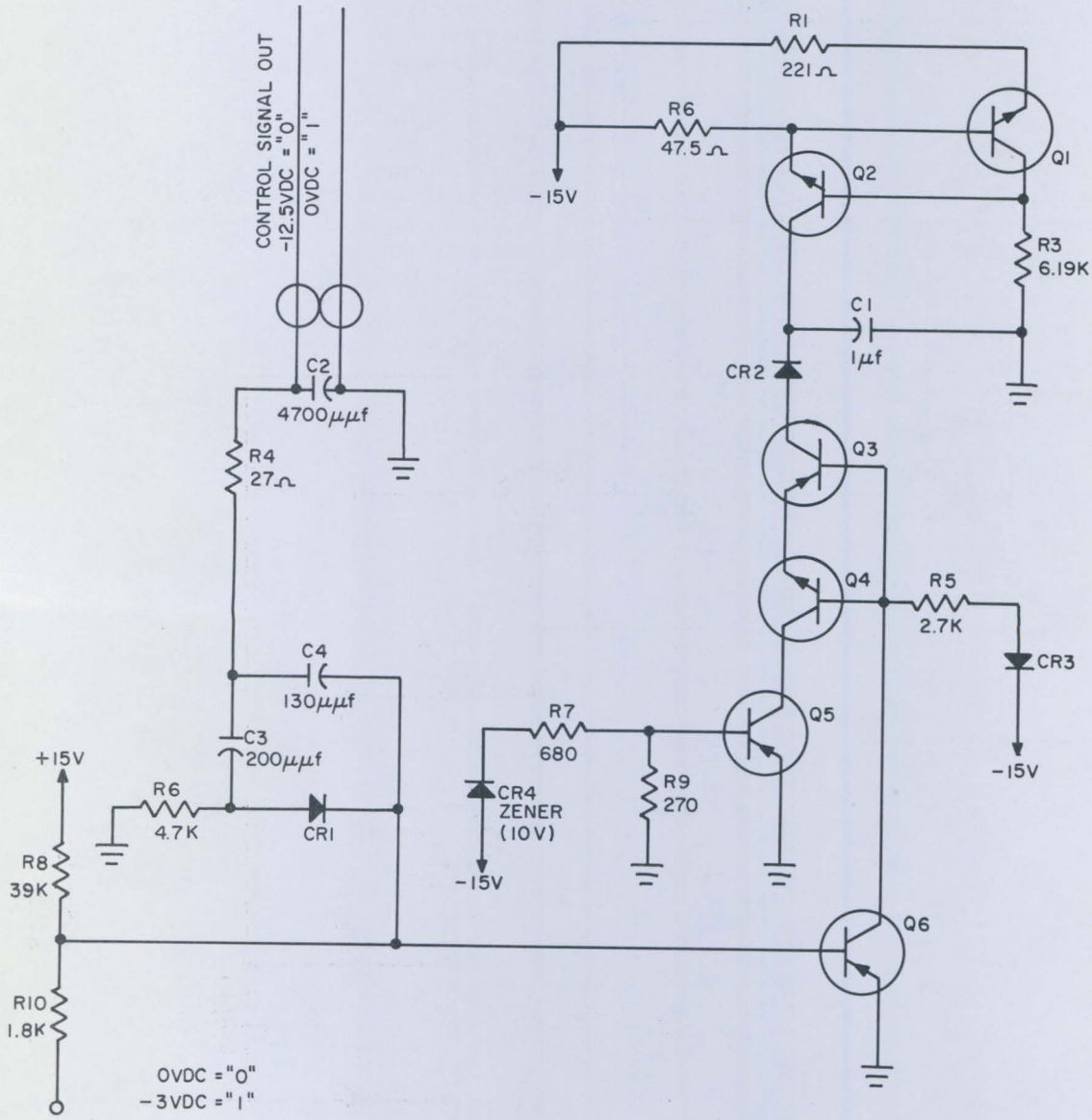


Figure A-3. Control Line Driver

logic (18I00) via an external cable to a connector in the rear of the control console. The output of 50Y16 is also applied to indicator driver 99I50, which will light the INPUT REQUEST indicator on the adapter control panel.

When the Input instruction is executed and channel one is specified by the 2^0 bit of the Instruction word, the command to send the Input Acknowledge will be generated by the 05I00 circuit in the Trainer logic. The "1" output of 05I00 will be applied to the 60Y17 Control line driver and two standard inverters, 60I00 and 60I08. The output of 60Y17 will output a "0" level on the Input Acknowledge control line which is an indication to the external equipment that the data has been received. The outputs of 60I00 and 60I08, now "0" are applied to the gating inputs of all of the gated input amplifiers. Those amplifiers that are receiving "0" inputs from the cable will output a "1" or -3 VDC to the associated stage of the Q register. However, if the data on the line is a "0" or -12.5 VDC, the output of the amplifier is a "0" and that stage of Q will remain cleared.

This is the only gating action that will occur between the input from the cable to the setting of the Q register. Inputs associated with interrupt codes will utilize the same circuit and cable as normal inputs during data transfer except the Control line into the 50Y17 input amplifier will be charged up instead of the 50Y16.

(2) OUTPUT. - The contents of the Q register, regardless of what it is being used for will be applied as inputs to the data line drivers (60Y--). This data, although available on the cable, cannot be accepted by a piece of external equipment unless it has the Output Acknowledge control signal transmitted with it. When the contents of the Q register is to be outputted to external equipment, regardless of whether it is an external function or data to be processed, the charging of the cable will immediately take place. One of two signals will accompany the data; Output acknowledge or External function (60Y15 or 60Y16). A "1" input to either circuit will cause the circuit to output a "0" indicating to the external equipment that the cable should be sampled. The indicator associated with the control will also light on the adapter control panel.

When the external equipment is ready to receive another data word, the input to the 50Y15 circuit will be "0" representing an Output Data request. The "1" output of 50Y15 (-3 VDC) will be applied to the 13000 circuit in the Trainer logic.

APPENDIX B

COMPUTER MATHEMATICS

B-1. COMPUTER MATHEMATICS.

Several number systems, or more specifically, systems of numerical notation, are commonly used in digital computers. All of these systems have the common function of identifying quantities of magnitudes. The values expressed in the different systems are identical, but the methods of notation vary. It is important to understand these systems and their interrelationships in order to understand the functioning of digital computers.

a. NUMBER SYSTEMS. - A number system may be defined by two principle characteristics: its radix and its modulus.

(1) RADIX. - The radix, or base, of a system is a number equal to the number of unique symbols used to represent all the discrete numerical values or quantities encompassed by the system. For example, a system with a radix of 10, encompassing an infinite number of values or quantities, may use no more than ten unique symbols to represent any particular value.

(2) MODULUS. - The modulus of a system is the number of discrete quantities which the system can distinguish uniquely. An example of a modulus is found in the twelve-hour system (modulus 12) of telling time. Seven o'clock is indicated by the number 7 whether it be morning or evening.

(3) DECIMAL SYSTEM. - The decimal system with radix 10, mentioned above, is the most commonly used system. It contains ten unique characters, which represent respectively, ten (zero through nine) successive quantities. The use of decimal notation has obvious advantages: its symbols and arithmetic processes are clearly understood, and conversion, either mental or physical is unnecessary for utilizing data produced by a decimal computer.

(4) BINARY SYSTEM. - The binary system radix two, uses only two unique characters, "0" and "1", to represent quantities or magnitudes. For quantities greater than one, the two binary characters are repeated in a manner to be described. This system has particular advantages for use in digital computers since the digits may be conveniently represented electrically and physically by two states; i.e., on and off, signal and no signal, etc.

(5) OCTAL SYSTEM. - The octal system, radix eight, contains eight unique elements which are represented by the numerals zero through seven. It is very convenient to represent large binary numbers in octal notation, and examples of this conversion are as follows.

b. REPRESENTATION OF VALUES IN A NUMBER SYSTEM. - Any quantity, Q, may be expressed in completely general terms as follows:

$$Q = A_n r^n + A_{n-1} r^{n-1} + A_{n-2} r^{n-2} + \ldots + A_2 r^2 + A_1 r^1 + A_0 r^0$$

where $A_n \cdot A_{n-1} \ldots A_1 \cdot A_0$

is a set of coefficients whose largest value is one less than the radix of the system.

and $r^n \cdot r^{n-1} \ldots r^1 \cdot r^0$

is a set of decreasing powers of the radix of the system, limited in magnitude only by the number of terms required to express Q.

If the radix of the numbering system has been established, the quantity, Q, may be expressed in shorthand notation by listing only the coefficients of the various powers of the radix in descending order of the powers as follows:

$$A_n\ A_{n-1}\ A_{n-2} \ldots A_2\ A_1\ A_0$$

(1) REPRESENTATION OF VALUES IN DECIMAL NOTATION. - Values are expressed in decimal (radix ten) notation in accordance with the following rule:

$$Q = A_n \times 10^n + A_{n-1} \times 10^{n-1} + \ldots + A_1 \times 10^1 + A_0 \times 10^0$$

Thus, the quantity 137 is indicated by the decimal digits 137, and in terms of general equations:

$$Q = 137 = 1 \times 10^2 + 3 \times 10^1 + 7 \times 10^0$$

(2) REPRESENTATION OF VALUES IN BINARY NOTATION. - Values are expressed in binary (radix two) notation in accordance with the following similar rule:

$$Q = A_n \times 2^n + A_{n-1} \times 2^{n-1} + \ldots + A_1 \times 2^1 + A_0 \times 2^0$$

As an example, binary number 10 001 001 is represented in binary notation as:

$$1 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

By summing all terms of the equations:

$$1 \times 2^7 = 128$$
$$1 \times 2^3 = 8$$
$$1 \times 2^0 = 1$$
$$137$$

Therefore,

BINARY		DECIMAL
10 001 001	=	137

(3) REPRESENTATION OF VALUES IN TERNARY NOTATION. - Any numbering system can be expressed by a general equation. As an example, decimal 137 yields the following ternary expression.

DECIMAL		TERNARY
137	=	12002_3

Or,

DECIMAL	TERNARY
137	$= 1 \times 3^4 + 2 \times 3^3 + 0 \times 3^2 + 0 \times 3^1 + 2 \times 3^0$

The numerical equivalence is shown by converting all terms of the equation to their decimal equivalent and summing:

$$1 \times 3^4 = 81$$
$$2 \times 3^3 = 54$$
$$2 \times 3^0 = \underline{2}$$
$$137$$

c. CONVERSION FROM SYSTEM TO SYSTEM. - Generally, conversion from one numbering system to another is performed by using the remainder method of conversions.

(1) Divide the number being converted by the radix of the numbering system being converted to.

(2) Continue the divide operations, keeping track of the quotient until the remainder is smaller than the divisor (radix). This first remainder is represented as the coefficient to the lowest order power of the general equations.

(3) Determine the next coefficient to the next higher power by continuing the divide operation using the quotient of the previous divide operation as the dividend.

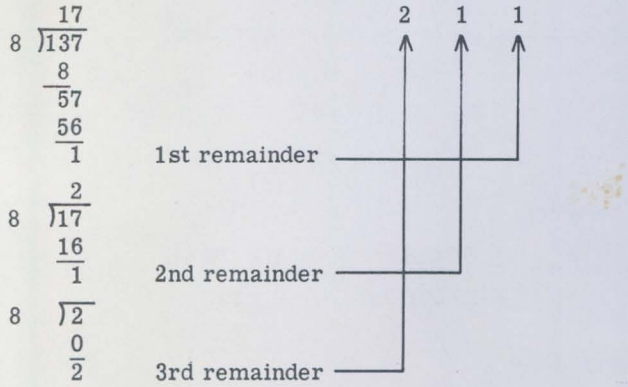
(4) Continue the above operations, keeping track of the coefficients raised to the proper power, until the dividend is smaller than the divisor. This dividend is then the highest order coefficient.

The coefficients of the terms of the radix, placed in the proper order of the general equations, are the representation of the quantity in the new numbering system. The use of these rules is illustrated in the following numerical examples.

(a) DECIMAL-TO-OCTAL CONVERSION

DECIMAL		OCTAL
137	=	?

Using the remainder method of conversions,



Therefore,

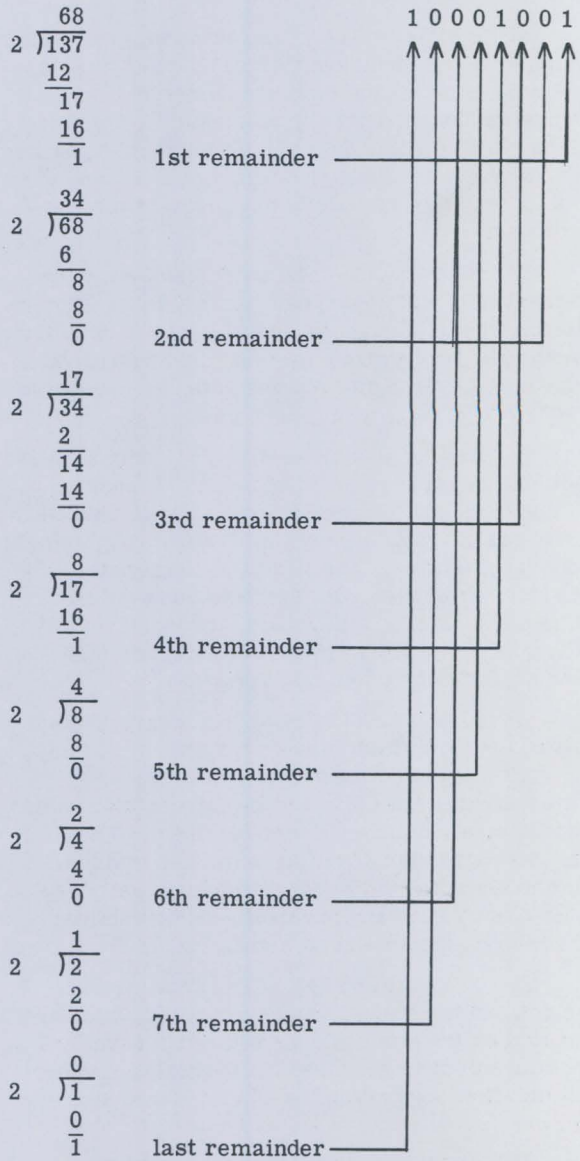
DECIMAL		OCTAL
137	=	211

This can be proved by using the general equation to convert the octal number back to its decimal equivalent.

$$2 \times 8^2 + 1 \times 8^1 + 1 \times 8^0 = 137_{10}$$

(b) DECIMAL-TO-BINARY CONVERSION. - The remainder method may also be used to convert decimal to binary. However, with larger numbers it is more convenient to convert them to octal first, then to binary.

DECIMAL		BINARY
137	=	?



Therefore,

DECIMAL		BINARY
137	=	10 001 001

This can be proved by using the general equation to convert the binary number back to its decimal equivalent.

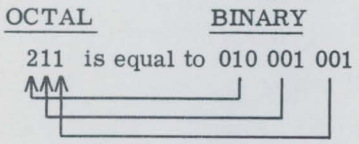
$$1 \times 2^7 + 1 \times 2^3 + 1 \times 2^0 = 137_{10}$$

The octal numbering system is used as an intermediate system between decimal and binary because of the difficulty of working with large binary numbers. From the table below it can be seen that three ascending, consecutive binary digits can be represented by a single octal digit, and, conversely, an octal digit can be represented by three consecutive binary digits.

DECIMAL	BINARY	OCTAL
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7
8	001 000	10
9	001 001	11
10	001 010	12

To convert an octal number into binary, convert each digit of the number into its binary equivalent, keeping track of the bits, and place them in the proper order of ascending powers.

Example:



Conversely, to convert binary to octal, reverse the operation.

BINARY	OCTAL
010 001 001	211

d. LIMITATIONS ON SIZE OF NUMBERS. - Operations may be performed on a quantity, regardless of its size, provided there is sufficient time and paper. This is also true, at least in theory, of computations performed by automatic calculators. In practice, the size of the number which a computer can handle at any one time is limited by the modulus of the system.

Consider the familiar desk calculator having a ten decimal digit capacity; the machine can express 10^{10} different numbers. It cannot distinguish 10^{10} from the number zero or distinguish between numbers which differ by integral multiples of 10^{10}. A simpler example of this limitation is found in the odometer used to record mileage in an automobile. Since this instrument generally operates with a modulus of 10^5, it cannot distinguish between two numbers which differ by integral multiples of 100,000 miles.

It is important to understand clearly the distinction between the symbolic representation of a number and the number itself. In the example of the desk calculator cited above, 10^{10} different representations of numbers are possible. These could represent the numbers from zero to $10^{10} - 1$, inclusive, or they could be any other set of 10^{10} values. If it is desired to represent both positive and negative numbers in the machine, then these 10^{10} different representations could be used to represent 5×10^9 positive numbers and 5×10^9 negative numbers. The range of numbers covered would then be -5,000,000,000 through +4,999,999,999. In this case zero is considered a positive number.

e. REPRESENTATION OF NEGATIVE VALUES IN DECIMAL NOTATION. - Negative numbers are indicated by the absolute value preceded by a minus sign in manual computations. This method of negative number indication is, in general, not suitable for calculators. Consider the case of the desk calculator handling a series of subtractions. The desk calculator uses a reversing gear and runs the register wheels backwards in performing subtraction. Assume that a positive number is initially entered into the machine, and that the subtraction process is continued enough times for the value in the register to go through zero and become negative. It will then be noted that the numbers shown by the register wheels pass through zero to all nines and then continue to decrease with each subtraction, whereas in the usual manual notation the numbers would be preceded by a minus sign and begin to increase. The solution to this apparent discrepancy depends upon recognition of the fact that the machine is of limited capacity and, as pointed out, can express only 10^{10} different numbers. The range of indication is 0 to $10^{10} - 1$. Any quantity outside this range of indication is represented in the machine by the quantity plus or minus an integral multiple of 10^{10}. The required integral multiple is necessary to make the indicated number fall between the aforementioned limits of zero and $10^{10} - 1$. For example, minus 137 is represented as $-137 + 10^{10}$ or 9999999863.

The number 9999999863, obtained by subtracting 137 from 10^{10}, is called the ten's complement of 137 with respect to 10^{10}. Note that the process used is equivalent to subtracting the rightmost nonzero digit from 10 and all other digits from nine, or that all digits are subtracted from nine and then a one is added.

Two examples of obtaining the ten's complement with respect to 10^8 are shown below.

```
 1 0 0 0 0 0 0 0 0      9 9 9 9 9 9 9 9
   0 0 0 0 0 5 4 8      0 0 0 0 0 5 4 8
 ---------------      ---------------
 9 9 9 9 9 4 5 2      9 9 9 9 9 4 5 1
                                    1  Add Complement
                      ---------------
                      9 9 9 9 9 4 5 2
```

99999452 is the ten's complement of 548. Notice that in both examples, the rightmost digit was handled separately.

The method of negative representation, used in desk calculators, is not directly applicable to computers because they cannot be made to run backward conveniently, or economically. To avoid this difficulty, the complement system (system of representing negative values) is used. The basic rules of complementing, which are applied to computer use, are as follows:

(1) In an additive computer, (one which performs all arithmetic operations by adding) simply add to perform the add process. To subtract, first complement the subtrahend, and then add.

(2) In a subtractive computer, (one which performs all arithmetic operations by subtracting) simply subtract to perform the subtract process. To add, first complement the addend, and then subtract.

The following examples are to support the above rules using ten's complement arithmetic with a system having a modulus of 10^8. Ignore all end-around borrows, and end-around carries.

ADDITIVE COMPUTER

Add Process

```
0 0 0 0 7 8 4 3
0 0 0 3 4 0 9 1
---------------
0 0 0 4 1 9 3 4
```

Subtract Process

```
0 0 0 0 0 8 7 5                          0 0 0 0 0 8 7 5
0 0 0 0 0 5 4 8 —— ten's complement ——   9 9 9 9 9 4 5 2
---------------                          ---------------
0 0 0 0 0 3 2 7                          0 0 0 0 0 3 2 7
```

SUBTRACTIVE COMPUTER

Add Process

```
0 0 0 0 7 8 4 3                          0 0 0 0 7 8 4 3
0 0 0 4 5 0 9 1 —— ten's complement ——    9 9 9 5 4 9 0 9
---------------                          ---------------
0 0 0 5 2 9 3 4                          0 0 0 5 2 9 3 4
```

Subtract Process

```
0 0 0 0 0 8 7 5
0 0 0 0 0 5 4 8
---------------
0 0 0 0 0 3 2 7
```

The nine's complement system may be used to eliminate the need for handling of the rightmost digit, separately.

It is obtained simply by subtracting each digit from nine (treating all digits alike) which in effect, reduces the modulus of the system by one.

Consider that the basic computing or counting mechanism is decimal in nature, each position of the mechanism having ten stable states. Upon reaching the tenth state and the addition of one more unit, the original count reverts to its zero state and a carry is generated which is transmitted by the mechanism to the next position.

The principles outlined above are the same for negative numbers except that a different method of notation is used.

In the nine's complement system, there are two representations for zero. For instance, the complement of 01 is 98. Adding one to this number yields 99, which is a representation for zero. Subtracting 01 from 01 yields 00. Therefore, by reducing the modulus of a system by one, two representations of zero are obtained. The two zero possibilities are not really a disadvantage because one possibility is always automatically excluded by the system of arithmetic used by the machine. When the modulus is reduced by one, an extra corrective step must be supplied at the transition point. This step is known as end-around carry in additive machines and end-around borrow in subtractive machines.

End-around carry means that whenever the number in the system passes through zero, in the additive sense, a quantity one is automatically added to the right hand digit of the number to correct for the transition point. This quantity one is the digit which would ordinarily appear in the next left column if another column were available.

End-around borrow means that whenever the number in the system passes through zero, in the subtractive sense, a quantity one is automatically subtracted from the right hand digit of the number to correct the transition point. This quantity one is the digit which would ordinarily be borrowed from the next column if another column were available.

The use of nine's complement arithmetic is demonstrated below with a system having a modulus of 10^8. The same basic rules apply in regard to subtractive and additive devices.

(a) ADDITIVE COMPUTER

Add Process

Pencil and paper method:

```
+ 7658
- 4593
-------
+ 3065   answer
```

Machine method:

+ 7658 is equal to 0000 7658
- 4593 is equal to 9999 5406

Add: end-around carry

```
0000 7658
9999 5406
---------
0000 3064
       1
---------
0000 3065
```

Subtract Process

Pencil and paper method:

```
+ 8346
+ 9347
-------
- 1001
```

Machine method:

+ 8346 is equal to 0000 8346
+ 9347 is equal to 0000 9347

Complement the subtrahend in order to subtract in an additive computer, therefore:

0000 9347 becomes 9999 0652

Add: 0000 8346
9999 0652
9999 8998

Answer: 9999 8998 is equal to -1001 in nine's complement arithmetic.

(b) SUBTRACTIVE COMPUTER

Add Process

Pencil and paper method:

+ 83476
- 94387
+177863

Machine method:

+ 83476 is equal to 000 83476
+ 94387 is equal to 000 94387

Complement the addend in order to add in a subtractive computer. Therefore:

000 94387 becomes 999 05612

Subtract: end-around borrow
00083476
99905612
00177864
1
00177863

Answer: 001 77863 in nine's complement arithmetic

Note

An end-around borrow was generated, borrowing one from the least significant digit.

Subtract Process

Pencil and paper method:

- 83476
+ 99652
-183128 answer

Machine method:

- 93476 is equal to 999 16523
+ 99652 is equal to 000 99652

Subtract: 99916523
00099652
99816871 answer

Answer: 99816871 is equal to -183128

Note

No end-around borrow was generated.

The basic differences can readily be seen between nine's complement and ten's complement arithmetic. In ten's complement arithmetic, the rightmost digit must be treated separately, and end-around borrows and end-around carries are ignored. In nine's complement arithmetic, all digits are treated alike but an extra step must be supplied any time the system passes through zero. This correction factor of one is known as end-around borrow in a subtractive machine, and end-around carry in an additive machine. In ten's complement arithmetic, the zero quantity is always represented as zero. When the modulus of a system is decreased by one (nine's complement arithmetic), two zero quantities are represented. However, in an additive computer, the number 000...0 never results from an arithmetic operation.

f. BINARY NOTATION. - The remarks dealing with negative number representation in machines using decimal notation may be applied to negative number representation in machines using binary notations.

There are two systems of negative representation in binary notation which are similar to the ten's and nine's complement systems. These are the two's complement, which is analogous to the ten's complement, and the one's complement, which is analogous to the nine's complement system.

Consider figure B-1 which is a tabulation of the sixteen decimal values from zero through 15, and the corresponding binary representations.

Negative values in the two's complement system are formed in a manner similar to that used in the forming of ten's complements. Figure 4-2 shows a system having a modulus of sixteen as used in figure B-1, but having an equal number of positive and negative values. Zero is considered to be a positive value. The negative values are formed by subtracting the absolute value from 2^4. The values -8 through +7 are now represented by the same 16 binary values used in figure B-1.

The two's complement system possesses the same characteristics for machine adaption as the ten's complement system; the rightmost nonzero digit must be treated individually. The one's complement system presents the same characteristics as the nine's complement system; all digits are treated alike.

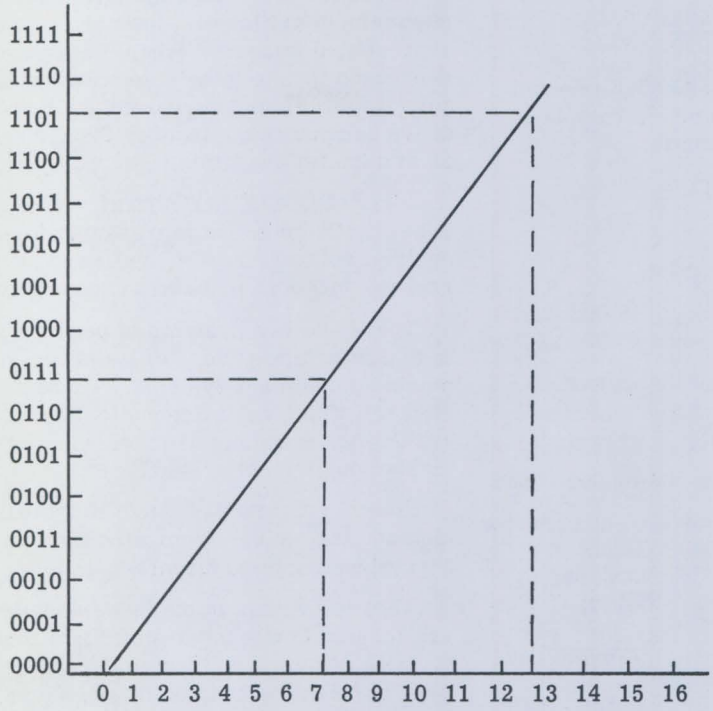
Figure B-3 depicts the 15 decimal values which may be represented by four binary digits, plotted against one's complement representations.

The negative values in figure B-3 are obtained by subtracting the absolute values from $2^4 - 1$. For example, to obtain the complement of +5, subtract this quantity from $2^4 - 1$.

$$\begin{array}{l} 1\,1\,1\,1 = 2^4 - 1 \\ 0\,1\,0\,1 = +5 \\ \hline 1\,0\,1\,0 = -5 \end{array}$$

The one's complement of a binary number is simply obtained by changing "1's" to "0's" and "0's" to "1's". This is illustrated in the tabulation below. Note that all negative numbers have a "1" in the leftmost place, as they did in the two's complement system.

0 = 0000	-0 = 1111
1 = 0001	-1 = 1110
2 = 0010	-2 = 1101
3 = 0011	-3 = 1100
4 = 0100	-4 = 1011
5 = 0101	-5 = 1010
6 = 0110	-6 = 1001
7 = 0111	-7 = 1000



0 = 0000
1 = 0001
2 = 0010
3 = 0011
4 = 0100
5 = 0101
6 = 0110
7 = 0111
8 = 1000
9 = 1001
10 = 1010
11 = 1011
12 = 1100
13 = 1101
14 = 1110
15 = 1111

NOTE

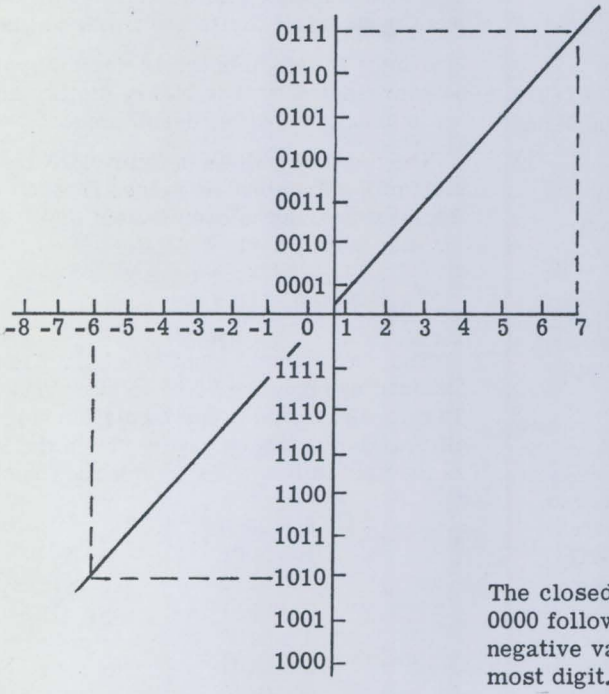
A total of 2^4, or 16, values can be expressed. The modulus of this system is 2^4 and the binary numbers listed may be considered a closed loop with 0000 following 1111.

Figure B-1. Decimal vs Binary Positive Values

$-5 = 2^4 - 5$

$-5 = 10000 - 0101$

$-5 = 1011$

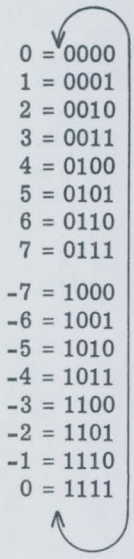


0 = 0000
1 = 0001
2 = 0010
3 = 0011
4 = 0100
5 = 0101
6 = 0110
7 = 0111
-8 = 1000
-7 = 1001
-6 = 1010
-5 = 1011
-4 = 1100
-3 = 1101
-2 = 1110
-1 = 1111

NOTE

The closed loop is still maintained and 0000 follows 1111. Note also that all negative values have a one as the left-most digit.

Figure B-2. Decimal vs Binary (Two's Complement Representation)



NOTE

The same closed loop is used, but two values for zero are possible. The modulus of the system has been reduced from 2^4 to $2^4 - 1$.

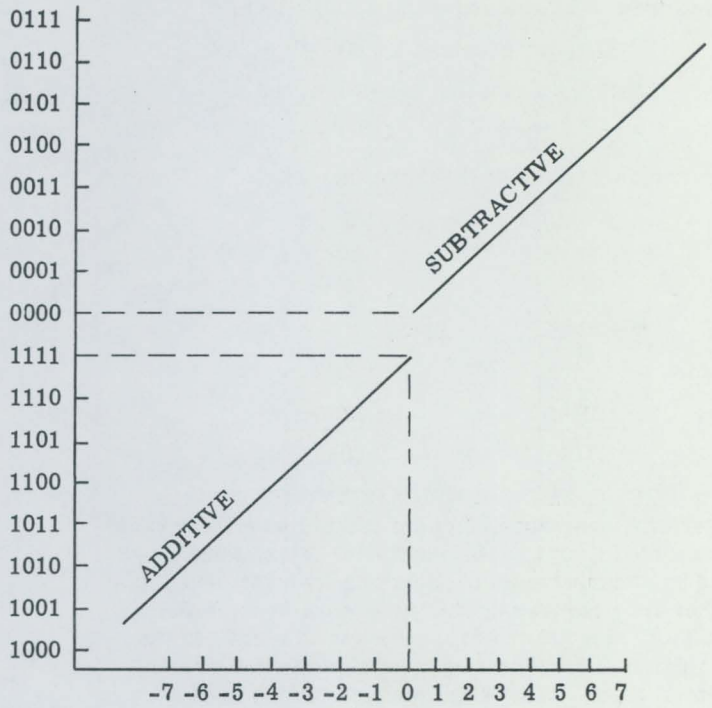


Figure B-3. Decimal vs Binary (One's Complement Representation)

g. FRACTIONS. - Computers are generally classified as being either fractional or integral. In an integer computer, the binary point is understood to be at the extreme right end of the binary number, therefore, all represented quantities are greater than one. Whereas, in a fractional computer, the binary point is understood to be located between the sign bit and the most significant digit. Therefore, all represented quantities are less than one.

The fractional bits are multiplied by the negative powers of two to determine their true value. The bit to the immediate right of the binary point is multiplied by 2^{-1}, the next lower order bit by 2^{-2}, the next by 2^{-3}, etc. As an example, the decimal equivalence of binary fraction 0.101100 is developed as follows:

BINARY		DECIMAL
0.101100	=	?

In binary notation 0.101100 is represented as:

$$1 \times 2^{-1} \times 0 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4} + 0 \times 2^{-5} + 0 \times 2^{-6}$$

By summing all terms of the equation:

$$1 \times 2^{-1} = 1/2$$
$$1 \times 2^{-3} = 1/8$$
$$1 \times 2^{-4} = 1/16$$
$$11/16$$

Therefore:

BINARY		DECIMAL
0.101100	=	0.6875

When working with positive binary fractions the above procedure may be used with both the one's and two's complement systems. To convert negative binary numbers, a slightly different procedure must be utilized as demonstrated by the following examples.

One's Complement:

BINARY		DECIMAL
1.01101	=	?

The number can be changed directly to decimal by using one of two methods. Ignoring the sign, find the one's complement of the number and treat the ones as having significance. More simply, treat the zeros as having significance. In binary notation, the number is represented as:

$$1 \times 2^{-1} + 0 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} + 0 \times 2^{-5}$$

By summing all terms of the equation:

$$1 \times 2^{-1} = 1/2$$
$$1 \times 2^{-4} = 1/16$$
$$9/16$$

Therefore:

BINARY		DECIMAL
1.01101	=	-0.5625

Two's Complement:

BINARY		DECIMAL
1.10011	=	?

Find the two's complement of the number, ignore the sign, and treat the ones as having significance.

0.10011 becomes 0.01101

In binary notation the number is represented as:

$0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} + 0 \times 2^{-4} + 1 \times 2^{-5}$

By summing all terms of the equation:

$1 \times 2^{-2} = 1/4$

$1 \times 2^{-3} = 1/8$

$1 \times 2^{-5} = 1/32$

$13/32$

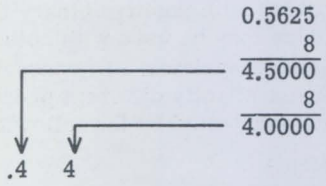
Therefore:

BINARY		DECIMAL
1.10011	=	-0.40625

h. DECIMAL TO BINARY CONVERSION. - As a matter of convenience, decimal fractions are generally converted first to octal fractions. This is accomplished by successive multiplications by eight. The integers that are produced by this procedure become the octal digits raised to the proper power of eight. In the event a decimal fraction cannot be represented exactly as an octal digit, continue the multiplications until a satisfactory approximation is reached.

DECIMAL		OCTAL
+0.5625	=	?

Using the multiply method:



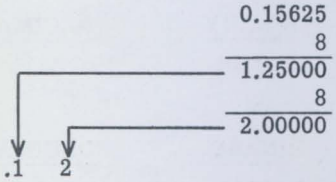
Therefore:

DECIMAL		OCTAL
+0.5625	=	.44
OCTAL		BINARY
0.44	=	.100100

To convert negative decimal numbers to octal and binary a slightly different procedure is used. Methods are illustrated for both one's and two's complement arithmetic.

DECIMAL		OCTAL
-0.15625	=	?

Using the multiply method, treat the number as if it were positive.



DECIMAL		OCTAL
.15625	=	.12

One's Complement:

Find the seven's complement of the number as follows:

77
.12
.65

Therefore:

DECIMAL		OCTAL
-0.15625	=	.65
OCTAL		BINARY
.65	=	1.110101

Two's Complement:

Find the eight's complement of the number as follows:

100
.12
.66

Therefore:

DECIMAL		OCTAL
-0.15625	=	.66
OCTAL		BINARY
.66	=	1.110110

i. BINARY TO DECIMAL CONVERSION. - As mentioned previously, the octal numbering system is generally used as an intermediate step between binary and decimal. Binary to octal conversions are performed by dividing the binary number into groups of three bits, starting at the binary point and working right. This procedure is demonstrated in the following example.

BINARY		OCTAL
0.101010	=	?
101 010		.52

In octal notation the number is represented as:

$5 \times 8^{-1} + 2 \times 8^{-2}$

By summing the terms of the equation:

$5 \times 8^{-1} = 5/8$

$2 \times 8^{-2} = 2/64$

$21/32$

Therefore:

BINARY		OCTAL
0.101010	=	+.65625

The octal numbering system was used in the above examples to demonstrate the effectiveness of having an intermediate numbering system.

There are other methods of conversions which can be used just as effectively. With familiarity, the user generally determines the methods which are best suited to his needs.

(1) ADDITION. - The process of addition is merely a procedure whereby counting is accomplished in an orderly and cyclical manner. Each time the cycle (number of unique symbols) is completed, a carry is provided. This is a general rule of the addition of quantities and it applies to all numbering systems.

Addition in the binary system is accomplished using the same general rule. Since only two digits are involved, it is necessary to remember only four digit combinations. They are:

0 + 0 = 0

1 + 0 = 1

0 + 1 = 1

1 + 1 = 0 plus a carry of 1 to the next higher rank.

The addition of 00010011 to 00110101 may be analyzed as follows:

128's	64's	32's	16's	8's	4's	2's	1's		DECIMAL
0	0	1	1	0	1	0	1	=	53
0	0	0	1	0	0	1	1	=	19
0	1	0	0	1	0	0	0	=	72

1's 1 + 1 = 0 and a carry of 1

2's 0 + 1 = 1 plus the 1's carry = 0 and a carry of 1

4's 1 + 0 = 1 plus the 2's carry = 0 and a carry of 1

8's 0 + 0 = 0 plus the 4's carry = 1

16's 1 + 1 = 0 and a carry of 1

32's 1 + 0 = 1 plus the 16's carry = 0 and a carry of 1

64's 0 + 0 = 0 plus the 32's carry = 1

128's 0 + 0 = 0

CHECK: 01001000 = 72

Octal addition may be performed using the same basic rules of addition. As an example, add:

```
 770146
  11135
1001303
```

The ensuing matrix may be used as an aid when adding in the octal numbering system. The letter c represents a carry of one to the next higher power.

AUGEND

ADDEND	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	0+c
2	2	3	4	5	6	7	0+c	1+c
3	3	4	5	6	7	9+c	1+c	2+c
4	4	5	6	7	0+c	1+c	2+c	3+c
5	5	6	7	0+c	1+c	2+c	3+c	4+c
6	6	7	0+c	1+c	2+c	3+c	4+c	5+c
7	7	0+c	1+c	2+c	3+c	4+c	5+c	6+c

SUM

(2) SUBTRACTION. - Generally, the introductory remarks concerning addition also apply to subtraction. However, in subtraction the counting process is reversed.

Binary subtraction, like binary addition, follows the same pattern as the corresponding decimal operation. Because only two digits are involved, the results of only four combinations need be remembered:

1 - 1 = 0

0 - 0 = 0

1 - 0 = 1

0 - 1 = 1 with a borrow demanded from the next left column

As an example, subtract 00010011 from 00111010

128's	64's	32's	16's	8's	4's	2's	1's		DECIMAL
0	0	1	1	1	0	1	0	=	58
0	0	0	1	0	0	1	1	=	19
0	0	1	0	0	1	1	1		39

1's 0-1 = 1 and a borrow of 1

2's 1-1 = 0 minus the 1's borrow = 1 and a borrow of 1

4's 0-0 = 0 minus the 2's borrow = 1 and a borrow of 1

8's 1-0 = 1 minus the 4's borrow = 0

16's 1-1 =: 0

32's 1-0 = 1

64's 0-0 = 0

128's 0-0 = 0

CHECK: 00100111 = 39

Octal subtraction may be performed by following the basic rules of subtraction. As an example, subtract:

```
 4600175
- 123456
 4454517
```

The subsequent matrix may be used as an aid when subtracting in the octal numbering system. The letter b represents a borrow from the next higher power.

MINUEND

SUBTRAHEND	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	7+b	0	1	2	3	4	5	6
2	6+b	7+b	0	1	2	3	4	5
3	5+b	6+b	7+b	0	1	2	3	4
4	4+b	5+b	6+b	7+b	0	1	2	3
5	3+b	4+b	5+b	6+b	7+b	0	1	2
6	2+b	3+b	4+b	5+b	6+b	7+b	0	1
7	1+b	2+b	3+b	4+b	5+b	6+b	7+b	0

DIFFERENCE

j. TWO'S COMPLEMENT ARITHMETIC. - The two's complement system of notation may be used to perform the add and subtract arithmetic processes. Remember that the two's complement system is similar to the ten's complement system and that the rightmost digit must be treated separately. Also, no end-around carries or borrows are involved in this type of arithmetic.

The subsequent examples are used to demonstrate the basic rules applying to additive-subtractive computers, using two's complement arithmetic, with a system having a modulus of 2^8.

(1) ADDITIVE COMPUTER

Add Process

Pencil and paper method:

+22
-73
-51 answer

Machine method:

00010110 is equal to +22
10110111 is equal to -73
11001101 is equal to -51

Subtract Process

Pencil and paper method:

-35
+43
-78 answer

Machine method:

11011101 is equal to -35
00101011 is equal to +43

Complement the subtrahend in order to subtract in an additive computer. Therefore:

00101011 becomes 11010101

Add:
11011101
11010101
10110010

10110010 is equal to -78 in two's complement arithmetic

(b) SUBTRACTIVE COMPUTER

Add Process

Pencil and paper method:

+41
+12
+53 answer

Machine method:

00101001 is equal to + 41
00001100 is equal to +12

Complement the addend in order to add in a subtractive computer. Therefore:

00001100 becomes 11110100

Subtract:
00101001
11110100
00110101

00110101 is equal to +53 in two's complement arithmetic

Subtract Process

Pencil and paper method:

-48
-53
+ 5 answer

Machine method:

11010000 is equal to -48
11001011 is equal to -53

Subtract:
11010000
11001011
00000101

00000101 is equal to +5 in two's complement arithmetic

k. ONE'S COMPLEMENT ARITHMETIC. - The one's complement system of negative representation of binary numbers may also be used to perform the add and subtract arithmetic processes. From previous examples, it was found that the one's complement system is similar to the nine's complement system because the modulus of the system being used is reduced by one. Because of this, all digits are treated alike, however, an extra step known as end-around carry in an additive computer and end-around borrow in a subtractive computer must be supplied any time the system passes through zero.

The following examples are used to demonstrate the basic rules applying to additive-subtractive computers, using one's complement arithmetic, with a system having a modulus of 2^8. The same problems will be used so that the differences between one's complement and two's complement arithmetic can be noted.

(1) ADDITIVE COMPUTER

Add Process

Pencil and paper method:

+22
-73
-51 answer

Machine method:

00010110 is equal to +22
10110110 is equal to -73

Add:
00010110
10110110
11001100

11001100 is equal to -51 decimal

Subtract Process

Pencil and paper method:

-35
+43
-78 answer

Machine method:

11011100 is equal to -35
00101011 is equal to +43

To subtract in an additive computer, complement the subtrahend. Therefore:

00101011 becomes 11010100

Add: end-around carry

```
  11011100
  11010100
  10110000
         1
  10110001
```

1010001 is equal to -78 in one's complement arithmetic

(2) SUBTRACTIVE COMPUTER

Add Process

Pencil and paper method:

```
 +41
 +12
 +53 answer
```

Machine method:

00101001 is equal to +41
00001100 is equal to +12

Complement the addend in order to add in a subtractive computer. Therefore:

00001100 becomes 11110011

Subtract: end-around borrow

```
  00101001
  11110011
  00110110
         1
  00110101
```

00110101 is equal to +53 in one's complement arithmetic

Subtract Process

Pencil and paper method:

```
 -48
 -53
 + 5 answer
```

Machine method:

11001111 is equal to -48
11001010 is equal to -53

Subtract:

```
  11001111
  11001010
  00000101
```

00000101 is equal to +5 in one's complement arithmetic

Note the similarity between the one's complement and the nine's complement system, and the similarity between the two's complement and the ten's complement system. It can be seen that each numbering system has a number of advantages and disadvantages.

There are a number of factors which are considered when selecting a particular numbering system for computer use. They are:

1. Speed
2. Hardware requirements
3. Maintenance and operation
4. Reliability
5. Compatibility, and
6. Special cases.

The selection of word length (modulus) is generally a compromise among a number of factors, such as available memory capacity, operating speed, required accuracy of numerical data, etc.

1. MULTIPLICATION. - Regardless of the notation that is used (decimal, ternary, binary, etc.), multiplication is always performed by three basic operations. Conventionally, they are:

1. Form the partial product, by multiplying the least significant multiplier digit times the multiplicand.
2. Repeat the above operation, using the next higher multiplier digit and shift the partial product left one place. Continue the operations until all multiplier digits have been sampled.
3. Form the product by adding all the partial products.

An example of decimal multiplication is shown as follows:

Multiply 257 by 127:

```
  257
  127
 1799
  514
 257
32639
```

Note that this is the conventional method of multiplication. The process of (1) starting with the least significant digit and (2) shifting the partial products left, may be reversed, as demonstrated by the following decimal example.

Multiply 257 by 127:

```
  257
  127
  257
   514
  1799
 32639
```

In performing binary multiplication, the same operations are performed. However, there are no multiplication tables to learn. If the multiplier contains a "1 "1", the multiplicand is added. If the multiplier contains a "0", the multiplicand is not added. A shift to the left occurs after the inspection of each multiplier digit. As an example, multiply 00111010 by 00101101.

128's	64's	32's	16's	8's	4's	2's	1's	Column
0	0	1	1	1	0	1	0	Multiplicand
0	0	1	0	1	1	0	1	Multiplier
0	0	1	1	1	0	1	0	one's column multiplier is "1", add, shift
-	-	-	-	-	-	-	-	two's column multiplier is "0", no add, shift
0	0	1	1	1	0	1	0	four's column multiplier is "1", add, shift
0	0	1	1	1	0	1	0	8's column multiplier is "1", add, shift
-	-	-	-	-	-	-	-	16's column multiplier is "0", no add, shift
0	0	1	1	1	0	1	0	32's column multiplier is "1", add
0	1	0	1	0	0	0	1	1 0 0 1 0 Product

CHECK: 0 1 0 1 0 0 0 1 1 0 0 1 0 = 2610

58 x 45 = 2610

The same basic rules apply to octal multiplication as demonstrated by the following example.

```
  1247
   305
  6503
 0000
3765
405203
```

The following table may be used as an aid when performing octal multiplications. The listed products are for multiplicands having only one digit. When they are greater than one, the left digit of those products having more than one digit, represent carries to the next higher power.

MULTIPLICAND

MULTIPLIER	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1		1	2	3	4	5	6	7
2			4	6	10	12	14	16
3				11	14	17	22	25
4					20	24	30	34
5						31	36	43
6							44	52
7								61

PRODUCT

m. DIVISION. - Manual decimal division is conventionally performed by what is known as the restoring method. In this method, the divisor is subtracted repeatedly from the dividend until the sign of the remainder changes. The dividend is then added (restored) one time, to insure a positive remainder. The divisor is then shifted to the right and the process is repeated. As an example, divide decimal 1634 by 25.

```
      065
25 ) 1634
     00      0 x 25
     163
     150     6 x 25
     134
     125     5 x 25
       9     remainder
```

In the example above, 25 is subtracted from 16, to yield a -9. The divisor, 25, is immediately added back to the dividend and 0 is entered as the first quotient digit. In the next operation, the seventh trial subtraction of 25 (175) from 163 creates a negative remainder so the divisor is added once to the dividend to restore it. Digit six is entered as the second digit of the quotient. The process is continued in the same manner until it is completed.

Binary restoring division is performed in the same fashion as decimal division. It is illustrated in the following example.

```
                00000101
10101 ) 01110101
        10101       Step 1
        0100001     Step 2
          10101     Step 3
          01100     Remainder
```

Step 1: 11101 10101 - Therefore, enter "1" as the first first digit of the quotient and subtract.

Step 2: 10000 10101 - Therefore, enter "0" as the second digit of the quotient but do not subtract. Bring down the "1" from the dividend.

Step 3: 100001 10101 - Therefore, enter a "1" as the third digit of the quotient and subtract.

The same basic rules also apply to octal division as demonstrated by the following example.

```
          1247
305 ) 405203
        305
       1002
         612
       1700
       1424
        2543
        2543
        0000
```

n. OVERFLOW. - An overflow condition is defined as the fault which arises any time the modulus of the register, holding the answer, is exceeded by the answer.

Generally, in most computing devices, overflow faults can occur during the add, subtract, and divide arithmetic operations. It can never occur during a multiply process if the register holding the product is one less than twice the length of the registers holding the multiplier or the multiplicand.

Overflow can only occur in an add operation if the two numbers being added have the same sign. When the first condition is satisfied and the sign of the sum changes, overflow has occurred.

Overflow can only occur in a subtract operation if the two numbers being subtracted have unlike signs. When the first condition is satisfied and the sign of the minuend is equal to the sign of the subtrahend, overflow has occurred.

Overflow can occur in a divide operation if the quotient exceeds the modulus of the register holding the answer. A check is generally performed at the beginning of the divide operation to determine if this fault exists.

BW #1

```
66067 04010
76250 10000 30000 75400 07400 62102 36241 33377
64073 64125 05400 56110 75400 07400 62113 36241
06000 64115 66113 33377 64104 66116 05377 74252
12110 66122 12252 33400 64073 30000 55000 75400
07400 62133 64140 57000 36241 33377 64127 64165
05400 56150 75400 07400 62146 64153 57000 36241
06725 64155 66153 33377 64142 66156 05377 74252
12150 66162 12252 33400 64127 04007 76251 30000
75400 55000 33377 64170 30000 07400 57000 77400
33377 64175 06251 24001 76251 61207 64174 30000
07400 62230 60230 33377 07400 62217 64224 57000
36241 33377 64210 64234 13377 66226 11400 66221
13377 66232 11400 66213 06250 24001 76250 61253
64071 64221 60244 64241 06241 20003 76241 64241
00000 00000 00000 66067
```

66 - 253

BW #2

```
66467 04010
76650 10000 30001 75000 07000 62502 36641 33377
64473 64525 05000 56510 75000 07000 62513 36641
06000 64515 66513 33377 64504 66516 05777 74652
12510 66522 12652 33400 64473 30001 55000 75000
07000 62583 64540 57000 36641 33377 64527 64565
05000 56550 75000 07000 62546 64553 57000 36641
06000 64555 66553 33377 64562 66556 05777 74652
12550 66562 12652 33400 64527 04007 76651 30001
75000 55000 33377 64570 30001 07000 57000 77000
33377 64575 06651 24001 76651 61607 64574 30001
07000 62630 60630 33377 07000 62617 64624 57000
36641 33376 64610 64654 13776 66626 11000 66621
13776 66632 11000 66613 06650 24001 76650 61653
64471 64000 60644 64641 06641 20003 76641 64641
00000 00000 00000 66467 07000 62660 60660 64634
13776 66662 11000 66634 20001 75000 76000 13000
66664
```

466 - 670

CHADDER

LOADS - 66-241
START 67

```
66067 04010
76200 06173 22175 61075 66071 06175 22173 61101
66075 30001 12174 06176 36201 12000 54107 41016
61113 36206 66104 36206 45001 44001 33016 64104
36201 10000 46001 61126 36206 66120 30002 12174
06177 36201 12000 54134 41016 61142 36206 66131
37777 00002 36206 45001 44001 33016 64131 36201
10000 46002 61155 36206 66147 36206 45001 44001
36201 46001 41002 61166 36206 66160 65067 06200
24001 61066 64070 00000 77776 77777 00002 00004
00001 64161 74140 76141 22140 64201 64156 12140
06141 64206
```

STOPS P=70 OK STOP

P=71 A≠0 (A = 00000) (Y = 77777) A+Y ≠ 0

P=75 A≠0 (A = 77777) (Y = 00000) A ≠ 0

104 A= VALUE Q= VALUE A≠ZERO END OF SHIFT

COMMAND 66-457

```
66067 04010
76455 06442 60074 61075 66075 57000 62100 66100
12441 63103 66103 61105 60106 66106 55000 63111
64112 66112 04001 62115 60116 66116 65120 64122
67122 66122 45016 62125 66125 45001 60130 66130
42001 60134 63134 66134 40016 63137 66137 46001
60142 66142 44016 63147 40016 45016 62150 66150
41016 36427 66153 46017 61156 66156 63160 66160
46016 62163 63164 66164 42016 61167 66167 63171
66171 44017 63174 66174 42017 63177 64200 66200
30002 31001 64204 66204 31001 66206 30677 35443
30000 32443 33700 64215 66215 33700 66217 30007
12441 04777 46006 36427 66225 04000 05770 46006
36427 66232 04000 07432 36427 66236 10000 04000
10077 46017 11770 44006 46011 36427 66247 04000
13432 46017 36427 66254 04000 55000 00077 42006
01770 46006 36427 66264 04000 12441 02441 36427
66271 04000 03432 36427 66275 04000 10000 14077
42006 15770 46006 36427 66305 04000 16441 36427
66311 04000 17432 36427 66315 06442 61320 66320
20077 45011 62324 66324 21770 61327 66327 22444
41010 23436 61334 66334 57000 26445 60340 66340
25000 61343 66343 57000 27436 60347 66347 24007
61352 66352 12446 43447 26450 61357 66357 46017
26451 61363 66363 06450 12451 47452 26453 61371
66371 46017 26454 61375 66375 12441 74456 06456
36427 66402 57000 76456 12456 46017 61410 66410
55000 75447 06456 36427 66415 57000 77447 12456
46017 61423 66423 06455 24001 61457 64070 64415
57000 60437 76456 06427 20001 76427 06456 57000
64427 77777 00000 00677 03400 77770 55372 36527
25630 76366 35555 04456 56750 00001 00000 66066
```

STARTS AT 67

ENDURANCE A = 77777

P = 70

4 min 15 sec.

CYCLE 10 TIMES

DEMONSTRATION 66-777

```
64303 10020
53000 30661 35055 07000 13001 30000 46006 71077
51000 33004 64076 32055 33776 33720 64072 64066
64322 71111 64113 10040 53000 64110 36561 00007
04347 56071 04404 56105 36066 36110 04007 76116
76117 06116 20001 76116 61157 70132 50000 00077
24042 61145 44006 70143 64136 40006 00077 24024
61164 04405 56071 04446 56105 36066 64120 06117
24001 76117 61120 64131 04447 56071 04504 56105
36066 36110 30517 35775 00000 76776 76777 30001
70200 12776 06777 46006 50000 74776 76777 00077
24042 61224 33005 64200 12776 06777 32775 77000
75001 33777 33532 64173 06777 42006 32775 77000
75001 33777 35775 04505 56071 06775 56105 36066
04533 56071 04566 56105 36066 36320 04567 56071
04576 56105 36066 36320 04577 56071 04614 56105
36066 36320 04615 56071 04624 56105 36066 36320
04625 56071 04660 56105 36066 36320 04661 56071
04720 56105 36066 67120 00037 00052 00074 00070
00064 00062 00066 00072 00060 00033 00522 00000
64276 36110 70322 50000 00077 24042 61320 30304
00077 27000 61336 33315 64330 64322 36340 64321
64337 04721 56071 04774 56105 36066 64340 45471
45704 16300 60406 03010 42420 20043 00401 05140
61304 31140 10503 34010 40725 04131 13024 24202
44204 04471 45724 04300 62503 06200 42401 30062
21406 13042 32504 07204 24577 45472 65734 06062
50426 34060 62504 26340 60625 42040 44725 57033
40405 30172 00407 25042 01120 16011 20306 24043
01111 04300 40131 14010 12012 42454 70757 03172
00403 06041 60307 20221 43006 42457 77777 45473
15720 11110 40520 11110 34204 04470 75725 04063
00720 04142 40447 34061 41730 16042 21413 14013
01104 01123 01406 20125 74204 47315 70005 30010
41424 04250 33412 24424 57777 45471 35711 30220
40103 04072 02001 04250 33404 47027 77777 20071
41125 04300 62204 00323 01636 57470 55730 13132
01201 00000 00025 57420 24547 13570 30104 24030
72001 05140 61304 01030 42405 03310 42503 34420
44701 57051 40636 04032 60430 06250 40634 07232
01204 11202 42404 01053 00604 52373 74245 45470
75734 11011 41511 25042 32504 74424 57777 45473
05722 22040 30504 11200 12404 24302 50477 77777
77777 77777 77466 42457 77777 45472 25714 17142
22004 23250 47442 45777 77777 45472 45734 23011
23016 01040 10520 04063 40723 20120 42503 34040
10503 34130 50104 03260 42612 03070 40105 20040
63407 23201 20425 03340 40530 17200 40603 31424
57777 45471 45726 04250 33404 22030 40603 01040
53017 20045 27004 24202 00401 05200 43001 01200
62230 06014 20404 47055 72004 31141 11104 24050
33104 25033 40425 03341 20420 12120 31242 45777
77777 45471 55711 20302 42004 06033 14204 47145
70424 30142 20447 01051 40636 04570 60301 04012
51520 42044 73257 34240 10401 25152 00404 03364
20404 04310 52006 04250 33404 05301 72004 22030
62045 31053 00104 47145 70401 03112 20425 03340
40103 04220 34245 00000 00000 00524 63642 32301
```

STARTS AT 120

WAIT FOR LIGHT (INPUT)

TYPE PERIOD AT END OF STATEMENT

FLICK 066- 377

```
66067 30001
05000 77400 33077 64070 30000 04004 13364 54100
76000 33012 64076 10040 53000 30001 70106 50000
55000 04000 03400 60174 33074 64106 66117 10020
53000 06203 76204 56130 20001 56131 20001 76204
06000 12000 61143 30000 46006 71135 51000 33004
64134 06204 64123 66144 06203 76204 56160 10040
53000 30000 46006 70153 50000 33004 64152 30000
26000 60177 06204 20001 56160 76204 24363 61116
46017 33001 64160 64151 55000 07400 67114 67162
00000 00000 00000 00205 00205 57605 45254 72543
74433 54665 47456 52454 70105 14240 41424 04300
41203 34011 40620 04073 02220 04010 30414 06243
41220 45010 53001 04300 42630 34110 10414 06040
10520 04261 12027 04311 41111 04060 30104 03161
63412 45140 10416 05201 63624 04301 11104 01052
00411 20010 12012 24041 11436 20043 25746 47355
74647 04300 62204 21453 01124 03040 10520 04063
40723 20122 40411 14362 00462 57466 44647 04300
62204 57704 54714 06041 63024 20042 50334 04240
50334 11220 40603 01200 40105 30010 40105 20041
50320 07040 12006 22240 40103 04173 01225 45123
40604 14010 43013 30140 65746 47043 02404 01051
42404 01202 40104 14240 40530 14122 56161 61616
15050 50505 05742 00000 00410 00440 00441 00443
00453 00455 00463 00465 00467 00471 00473 00000
```

1) PUNCH ON
2) LEADER
3) ENABLE RUN
4) STOP CODE
5) LEADER

STOPS AT 144

LOAD TAPE

ENABLE RUN

RUN FROM 067

STOPS AT 117

TO HNM LRGIPCVEZDBSYFXAWJ^{9}UQK^{O})
(_ ^{1}/ -5 4 6 3 7 2

MONITOR

```
66 00000 00451
70 00443 00074 00000 00451 66443 00000 00000 00000
100 76066 74067 06066 64443 56073 56066 56111 26067
110 61301 12451 74074 40011 00077 76254 24031 61226
120 06254 24033 61226 06254 24036 61255 64327 26254
130 62204 06074 76136 06075 12076 32077 76452 64360
140 74076 04001 35077 56103 10077 04102 56203 04073
150 56000 54201 10020 53000 10045 71155 51000 13000
160 35071 30000 44003 74254 00007 20056 56167 12061
170 71170 51000 12254 33004 64162 10004 71176 51000
200 32071 33074 64157 64242 06254 24073 62210 64131
210 20226 56212 64221 64267 64267 64267 64267 64250
220 64244 64235 64267 64352 64352 64267 06074 76231
230 32077 33005 64141 04002 64142 36305 04242 56203
240 10074 64147 06070 66102 67250 04001 56103 64251
250 36305 10074 64145 00000 66443 06066 20001 74254
260 36305 56265 20001 56103 06254 76617 64251 06074
270 76275 04277 56275 06075 12076 61277 64131 36305
300 64144 06304 76103 64111 64317 00236 06305 56316
310 04000 12074 76066 00777 56103 76070 64236 06256
320 76103 06072 76075 76076 76077 56103 66100 06254
330 24051 61337 06254 24050 61344 04057 64127 12076
340 71340 51000 71342 64366 10040 53000 70346 12076
350 50000 64370 06074 56071 06071 26066 61366 64267
360 76075 06365 24001 60362 64140 07777 04001 64143
370 74076 64366 00777
```

LOADS 66-372

STARTS AT 317

STOPS AT 100

ENT A W/ ADDRESS

PLAYER ROUTINE + 3 SONGS

070 - 7L0

64071 54074 56134 30000 13000 00177 24050 61102
06126 64103 06145 76123 00177 56124 02141 10000
46002 45006 56115 43142 54116 04054 42000 46011
20002 35143 30000 24001 33000 64124 24001 31001
64127 60123 32143 65074 33711 64074 70140 64205
67073 77400 00003 00004 33050 64144 24002 30000
60144 64132 00022 64205 00006 03400 00013 24007
45517 43477 44065 47057 76477 51457 42450 53447
46443 44065 22047 42450 51457 45447 44457 44065
44457 45052 45447 44070 44457 46443 45447 45052
44065 43473 43107 45517 43477 44065 47057 76477
51457 42450 53447 46443 44065 22047 42450 51457
45447 44457 44065 44457 45052 45447 42437 45050
70465 42070 42065 42043 45050 70465 42070 42065
22047 44047 00012 00144 07400 00000 00037 00024
43454 44047 44445 51443 50447 47057 45473 44507
45473 47057 52437 66450 25037 44443 44047 43062
42473 47107 44443 50447 46465 22047 44052 45037
44052 43065 52477 50050 43457 44047 43457 42473
43065 42477 42473 51507 45077 42107 46465 65050
46465 65050 55065 50050 44507 42477 43065 42107
42503 43065 52477 56457 22457 52437 65050 45037
44443 44047 43457 45077 46070 44052 42107 42113
44117 47057 42107 46465 65050 46465 43457 44443
42107 43457 22065 50447 51443 50447 47057 45473
44507 45473 47057 52437 65050 25037 44443 44047
43062 42473 47107 44443 50447 46465 22047 44052
45037 44052 43065 52477 50050 43457 44047 43457
42473 43065 42477 42473 51507 45077 42107 46465
46070 22473 42117 42107 45077 46465 50447 52437
23052 50447 44443 44047 43457 42473 56457 43000
43457 42473 43065 42477 42473 42107 22117 76050
45054 52452 50050 70465 44052 70465 43477 75503
75107 43107 45054 52452 50050 70465 44052 70465
43477 75503 75107 43107 45052 53447 50050 71457
44047 71457 44047 64050 73447 42057 42070 44070
45052 53447 50050 71457 44047 71457 45054 42057
42065 44065 45054 52452 50050 70465 44052 70465
43477 75503 75107 43107 45054 52452 50050 70465
44052 70465 43477 75503 75107 43107 45052 53447
50050 71457 44047 71457 44047 64050 73447 42057
42070 44070 45052 53447 50050 71457 44047 71457
45054 42057 42065 50465 44457 44065 44457 44065
42057 45052 42057 47057 42065 44457 44065 42057
42057 64050 42057 42447 42057 22057 45052 44457
45052 44457 42452 46045 42452 50052 42057 43043
64050 43043 43045 42447 45447 42062 45052 64050
42452 42454 42057 44457 45054 52452 50050 70465
42452 70465 43477 75503 75107 43107 45054 52452
45050 70465 42452 70465 43477 75503 75107 43107
45052 53447 50050 71457 42447 71457 44047 64050
73447 42057 42070 44070 42452 53447 50050 71457
44047 71457 45054 42057 54465 25050 44047 64050
73447

STARTS AT 070

070 - 157 PLAYER

ANCHORS AWEIGH A=070
Q=160

STAR DUST A=206
Q=260

JOHNSON RAG A=244
Q=470

ALL THREE A=570
Q=160

SORT

```
00037 00052 00074 00070 00064 00062 00066 00072
00060 00033 00030 00023 00016 00022 00020 00026
00013 00005 00014 00032 00036 00007 00006 00015
00035 64161 74214 10040 53000 12214 64151 64407
36151 70161 50000 64157 64426 13000 36172 63164
33776 64165 64167 44011 36200 44006 36200 64172
64177 36206 71202 44000 51000 64216 64202 74214
10020 53000 12214 64206 44245 64000 71216 64200
53000 12214 64215 64422 24001 56232 30000 00077
27120 61223 33001 64227 64223 64432 76214 04000
43214 76214 36245 22214 64235 64243 00777 21000
42011 56257 06000 25000 45006 56260 00077 20000
20000 42006 64245 64472 74300 30001 04000 43300
61273 35276 64263 33000 75277 64266 00004 00001
00012 00144 01750 23420 00000 00000 00000 00000
00000 00000 00000 00000 00000 00000 64500 06276
76333 20277 56324 04000 47277 36334 46017 06333
61316 24001 64320 00000 64326 46017 20120 56340
06121 46017 36200 64334 64470 24001 56372 04001
63353 45011 64354 55000 54360 56362 41011 56361
30567 07000 27001 62365 64372 07001 13000 75001
77000 64360 33571 64361 64344 00000 00000 67400
76513 74514 54427 10000 74515 74516 36157 00077
24004 61406 24041 61406 24001 61444 20004 61452
06514 36223 61427 36200 30517 36164 64406 04002
12516 36235 61437 30540 36164 10000 64405 63441
64405 06514 63433 64405 32515 12516 75567 33776
35515 64435 06513 56470 65470 10567 41016 61461
55000 06515 61504 24001 61470 20567 56502 36344
10012 36263 10045 36200 36200 30567 13000 36316
10045 36200 33572 64476 30551 36164 30563 36164
12514 06513 67400 40012 40002 00000 00000 04547
00105 03001 02404 00603 00104 03004 00634 00723
02012 00404 00112 02504 03013 03014 00657 44245
04724 00312 01225 05746 00401 00303 00423 01413
04204 04724 00130 01201 00406 02031 00406 03407
02320 01257 44245 04720 00622 05750 45045 17777
12777
```

LOADS 120 - 570

STARTS 377

1. Set Q to "in" Radix
2. Set A to "out" Radix
3. Set Q_{14} - avoids neg numbers.
4. A_{14} set large → small
   A_{14} clear small → large
5. Jump select ~~sets~~, no sort
6. Stop selected. stop after sort, modify and/or resort.
7. Comma after number.
8. Period after comma end input
9. Max number 100

TRIG

LOADS 100 – 776
STARTS 433

STOPS IN INPUT MODE

TAN
SIN SPACE ANGLE PERIOD
COS

TAN 90. HOLY MACK.

00037 00052 00074 00070 00064 00062 00066 00072
00060 00033 00001 64150 35112 30000 70116 50000
00077 27100 61130 33011 64120 00077 32112 64113
11000 64126 00003 00132 00000 64564 35132 30000
35133 74134 10012 43133 46017 22134 76133 36113
60154 74134 33004 64142 12133 32132 64135 00000
00016 00003 00024 00000 00132 00001 00000 57777
37777 20000 00000 00000 00000 00000 00000 00000
64601 63203 64204 55000 04000 47267 55000 74300
44016 63214 57000 20132 61702 46017 26273 45016
76272 74271 43271 74271 12270 43271 22271 67460
76274 76276 57000 20001 76277 30002 33776 11001
43000 42001 74301 33776 12276 47301 74275 12274
43275 67474 76276 12300 63256 57000 55000 74300
22277 76277 64236 06277 12272 64200 00000 00132
17665 00000 57777 40000 00000 00000 00000 00001
77777 00006 00000 00000 00000 00000 00000 00000
00000 00000 00000 00000 00000 00000 00000 00000
74421 64514 57000 62325 76422 10020 53000 10004
71330 51000 10057 71333 51000 62341 10056 71337
51000 10004 54403 12422 30004 04000 47423 77423
31001 64345 46017 60440 30001 07423 60366 06403
20001 56403 33004 64355 31001 07423 36410 33004
64365 12377 00777 24005 61430 71375 10042 51000
12421 43423 36410 33010 64401 64430 00000 00000
64403 20100 56413 06100 71414 46017 51000 46017
64410 00000 00000 00012 00000 00000 00000 00000
71430 10045 51000 10040 53000 64550 64550 00000
30000 07452 10000 71443 46005 51000 60442 33004
64441 64543 10243 23244 17416 75012 40444 00000
46001 76472 62465 46001 64230 76274 76276 26273
57000 64234 00000 00000 74473 12472 63501 61263
64252 12473 46001 61505 64252 06277 45001 64264
63533 74642 64750 00000 46017 61521 46017 76422
64322 46017 20000 64517 00000 64521 00000 00000
63440 64511 00000 10020 53000 10047 71536 51000
64440 00000 00000 71543 10057 51000 64430 00000
30000 35163 70552 50000 75160 33002 64552 36113
61563 74163 36113 36135 74164 36200 76166 74170
10132 06163 24056 61575 55000 46017 22164 46017
36200 76165 74167 06170 62606 55000 74171 06160
24016 61671 62740 06161 24014 61655 12166 63625
64623 36643 64640 04001 10000 47166 74642 10000
47166 06170 62635 06642 64320 06642 57000 64320
12273 04000 00006 64745 12165 04000 42001 74165
12166 04000 42001 74166 64643 12167 04000 63661
57000 12165 64320 12170 04000 63667 57000 12166
64320 06162 24016 61772 62663 64663 12166 04000
64601 00000 06200 26700 60722 06161 24014 60720
46017 26273 42001 04001 63716 57000 10000 64320
04000 64215 06162 24024 61710 64720 46017 64710
00000 00000 06167 76170 64616 00000 00000 00000
12166 04000 63744 64745 36643 12165 47166 64510
10000 47166 06171 62757 06642 57000 64320 06642
64320 06165 12166 76166 74165 64740 12170 76170
74167 64740 06165 76166 06167 76170 64616

PX 2552

VOLUME II

TECHNICAL MANUAL

for

UNIVAC DIGITAL TRAINER

SECTION 5

UNIVAC®
DIVISION OF SPERRY RAND CORPORATION
MILITARY OPERATIONS · UNIVAC PARK · ST. PAUL, MINN.

10 SEPTEMBER 1962

LIST OF ILLUSTRATIONS

SECTION 5 - FUNCTIONAL SCHEMATICS

SECTION 5 - FUNCTIONAL SCHEMATICS (Cont.)

SECTION 5
FUNCTIONAL SCHEMATICS

Note

The functional schematics contained in this volume are referenced in, and should be used in conjunction with, Section 4, Volume I, of the UNIVAC Digital Trainer manual.

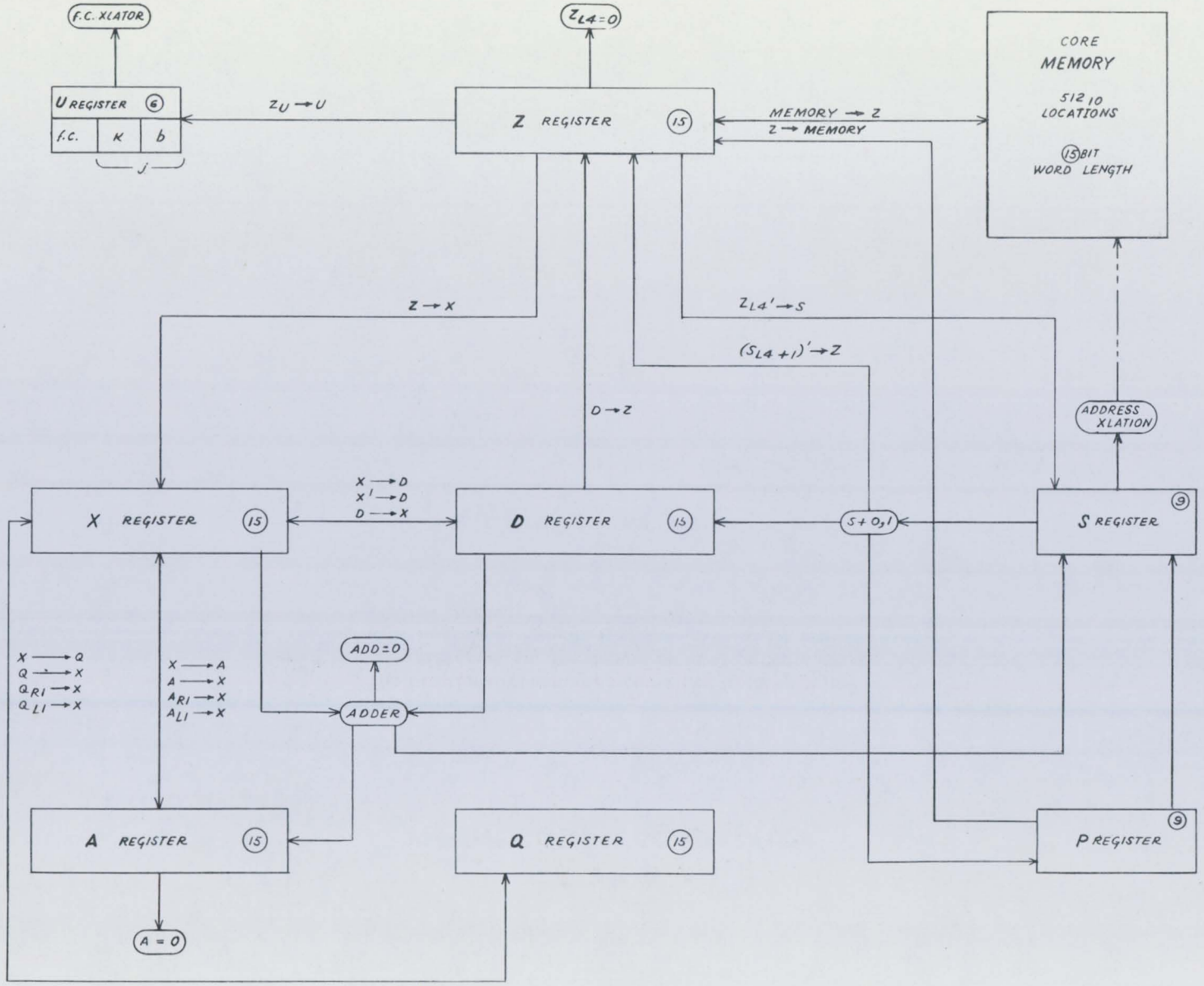


Figure 5-1. Block Diagram

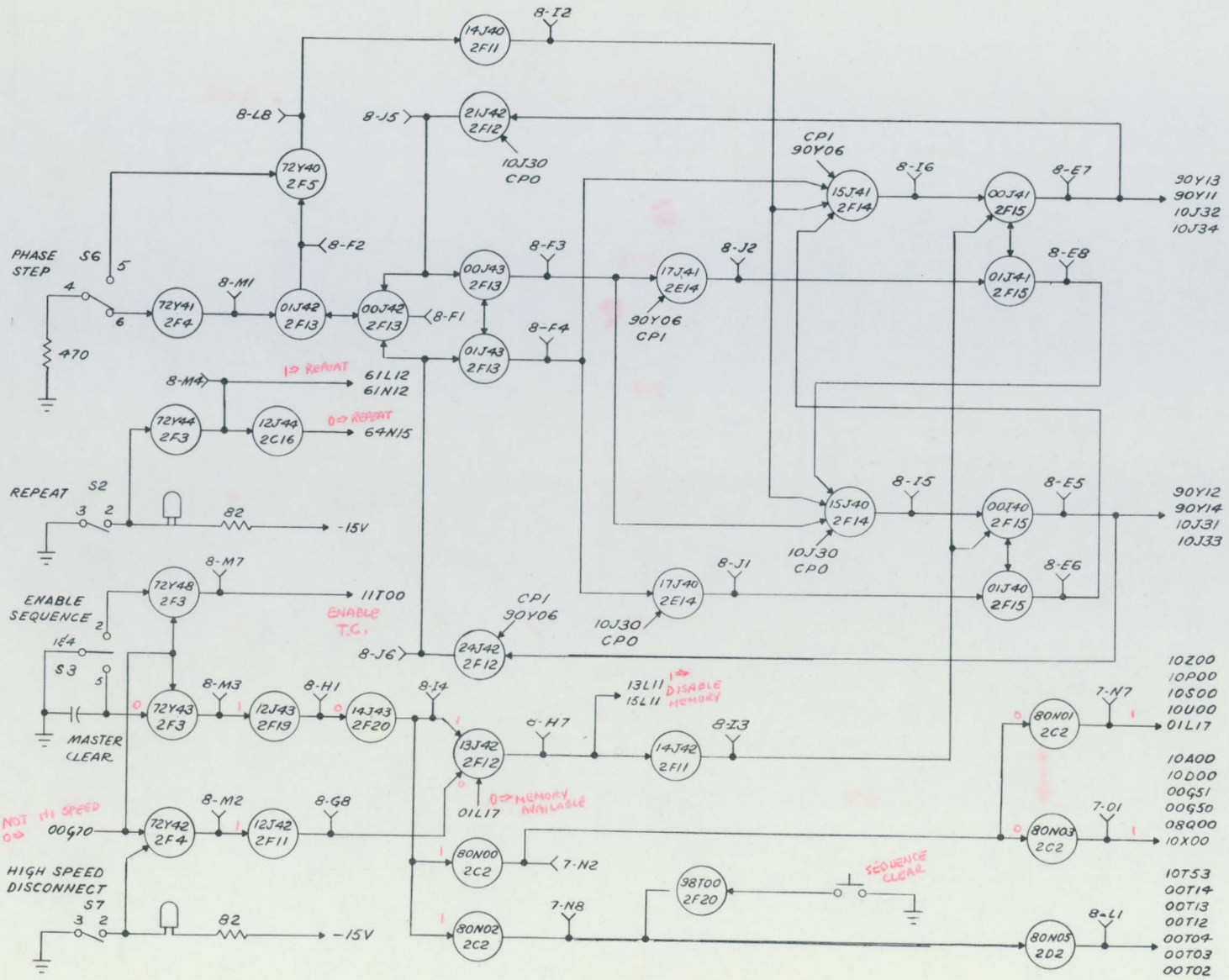


Figure 5-2. Control Console - I

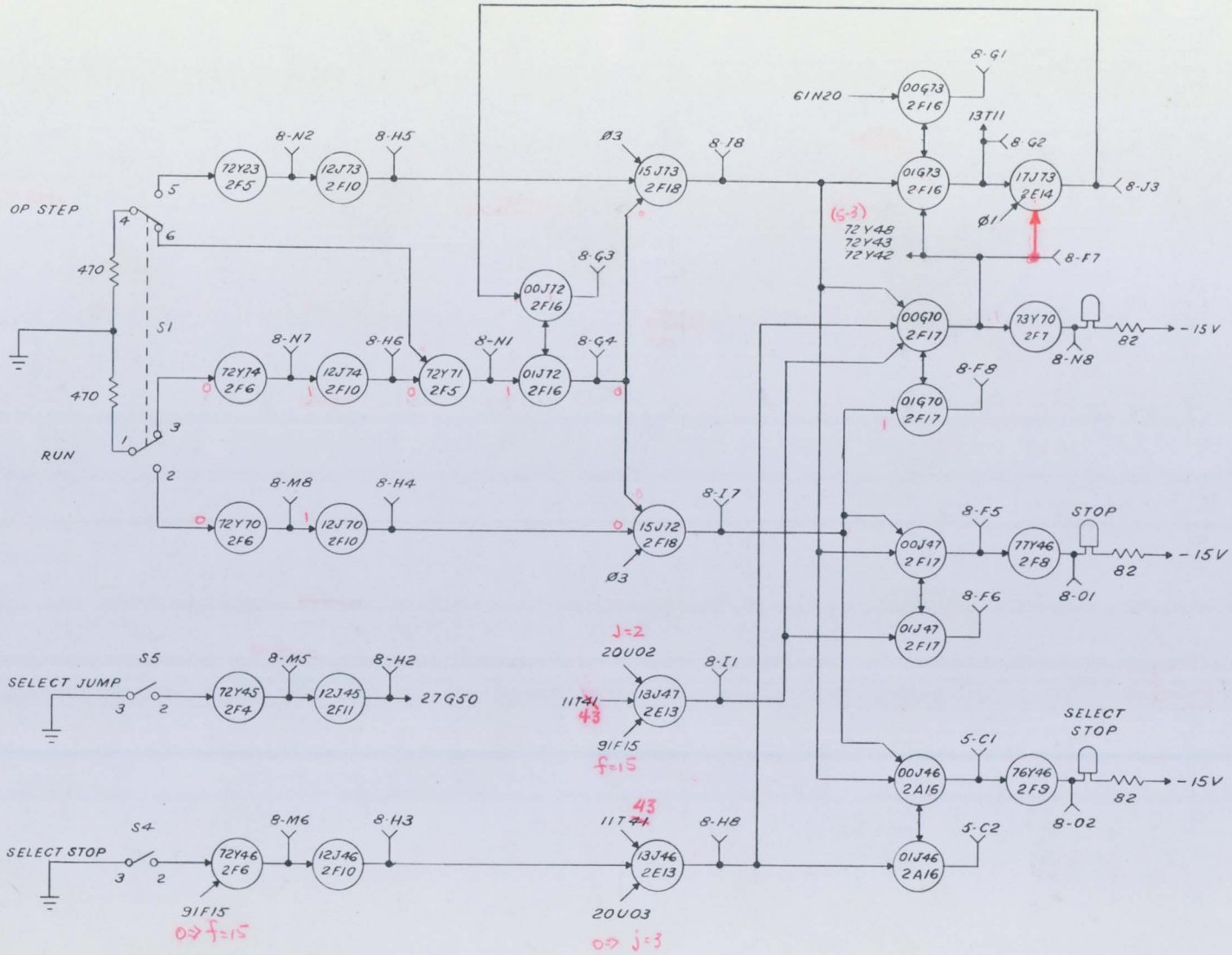


Figure 5-3. Control Console - II

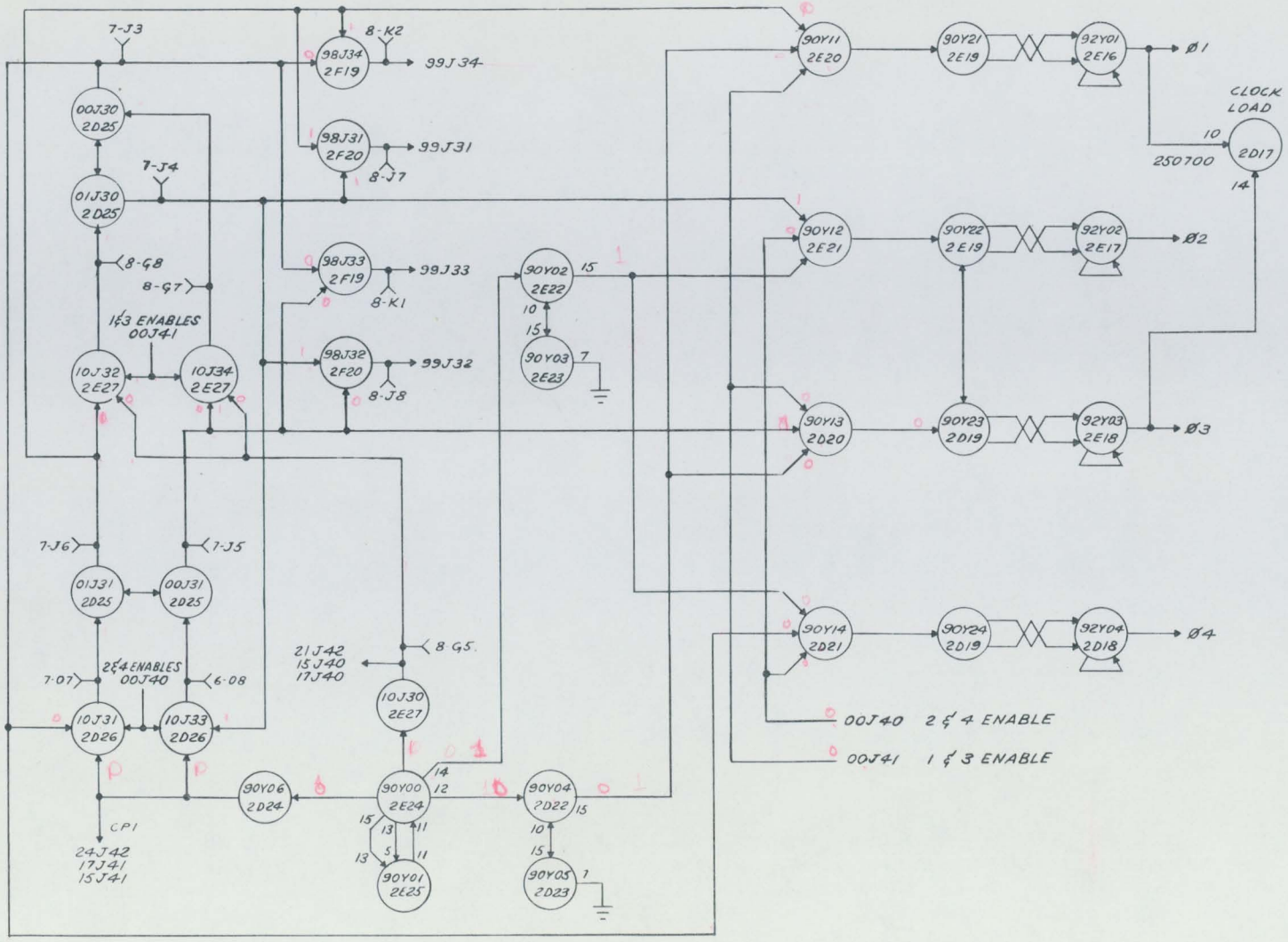


Figure 5-4. Master Clock

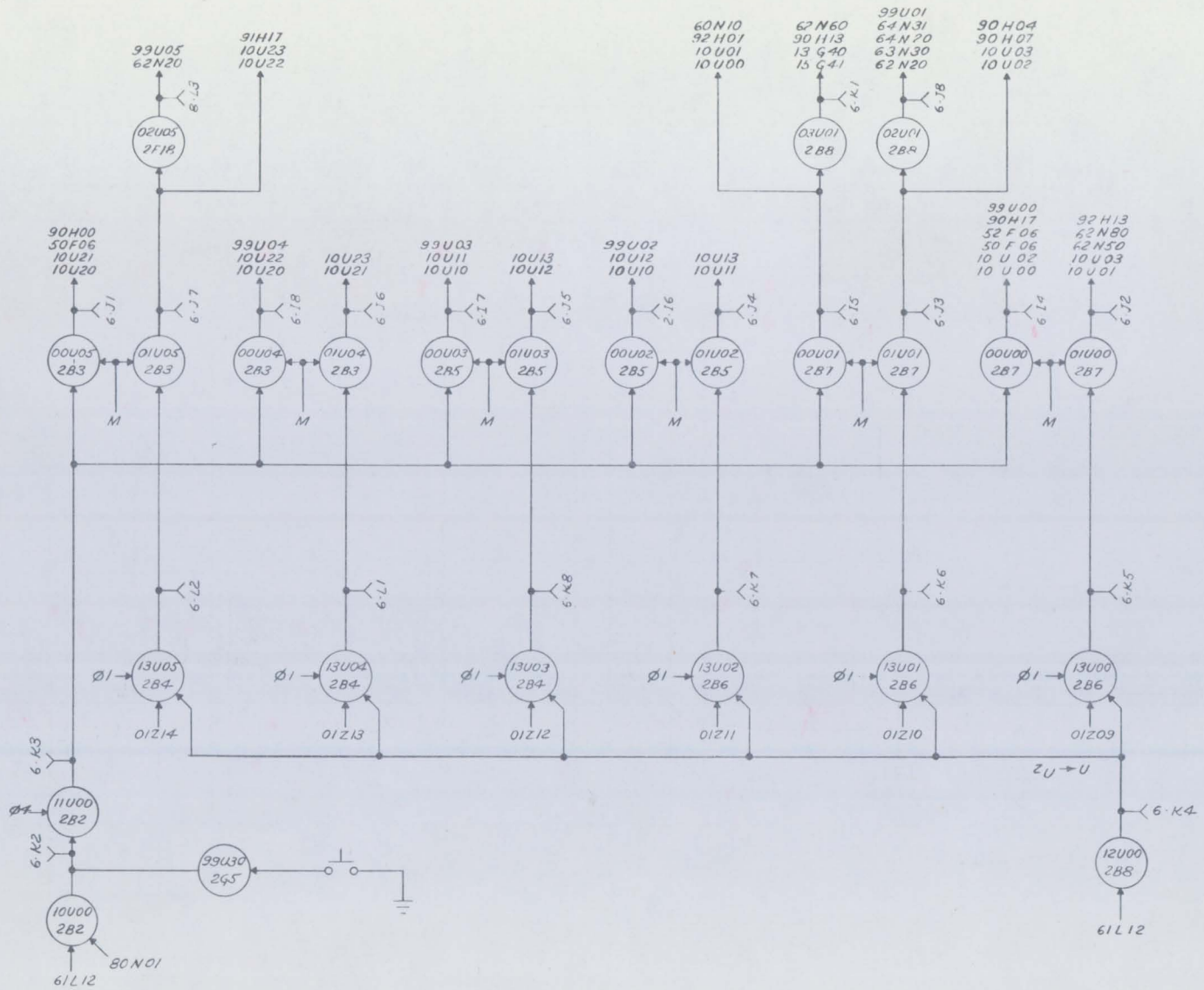


Figure 5-5. U Register

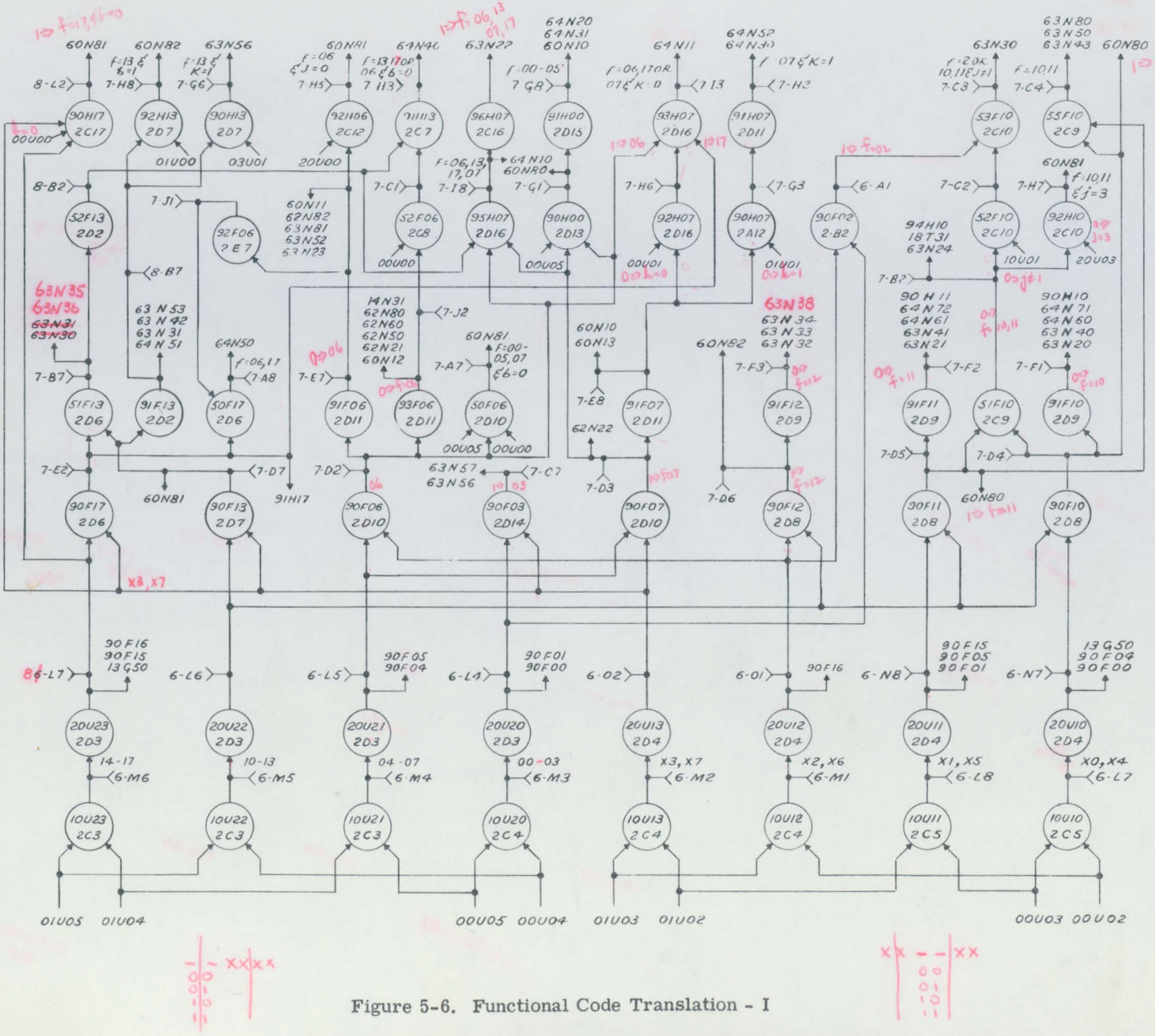


Figure 5-6. Functional Code Translation - I

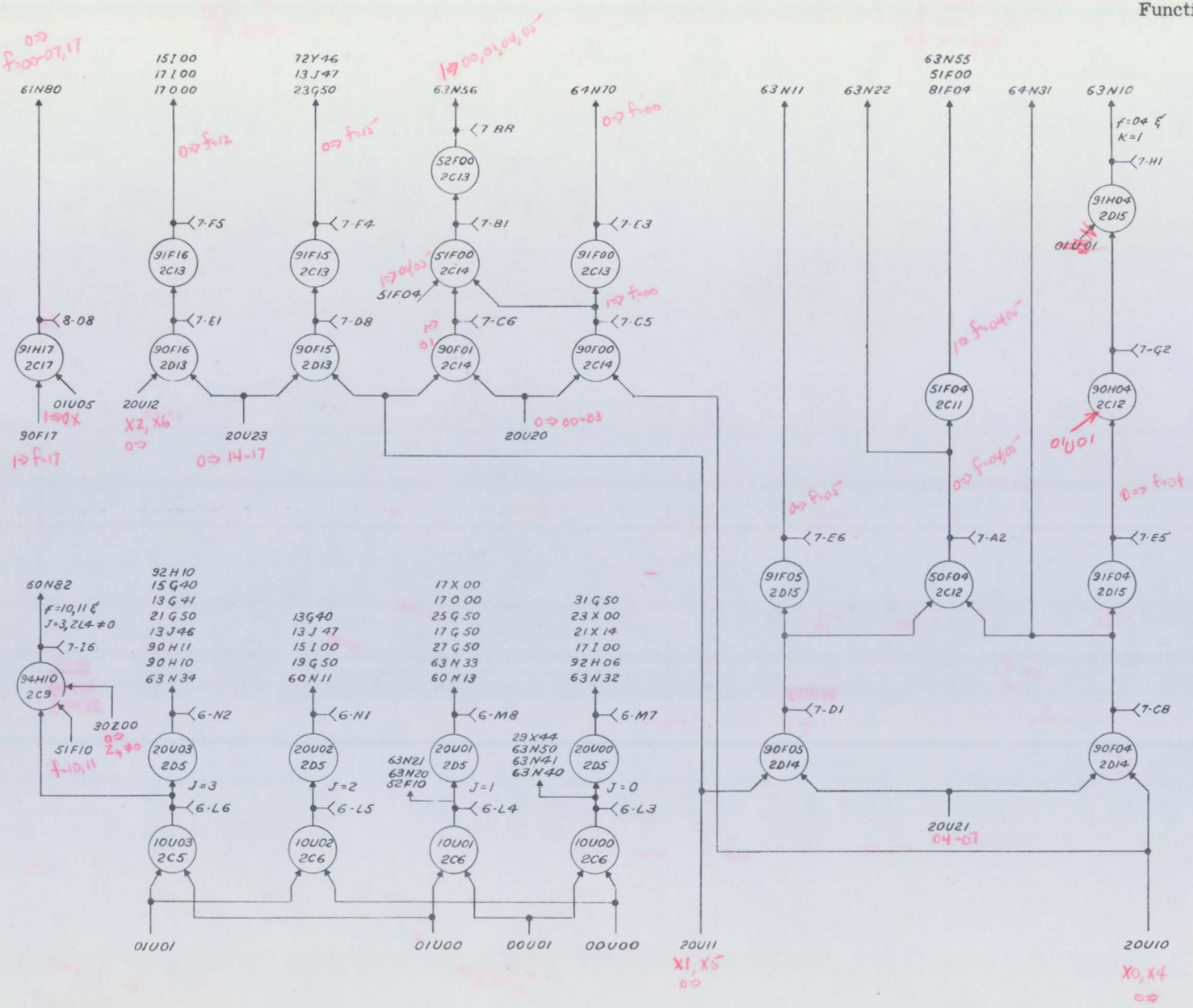


Figure 5-7. Functional Code Translation - II

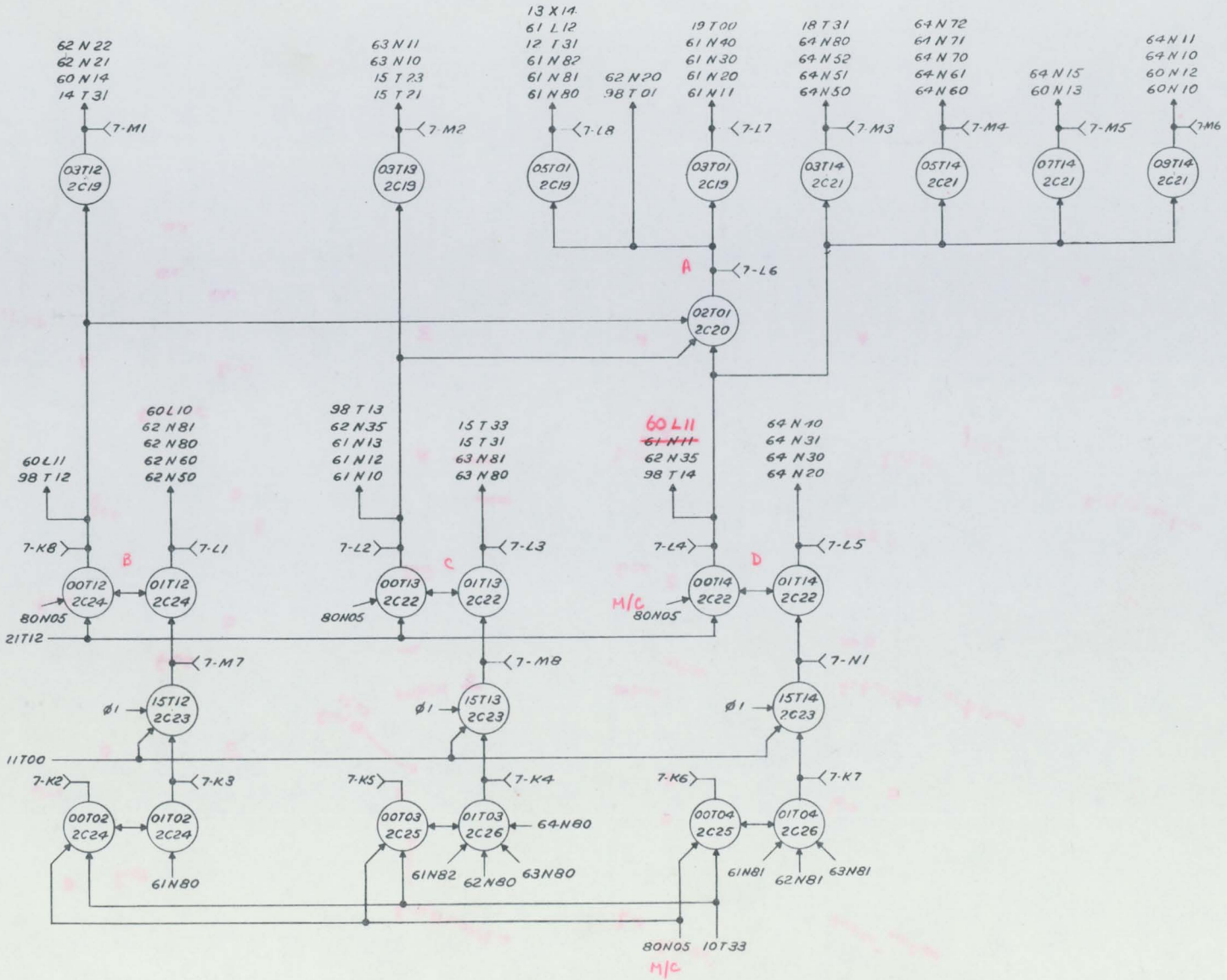


Figure 5-8. Sequence Enables

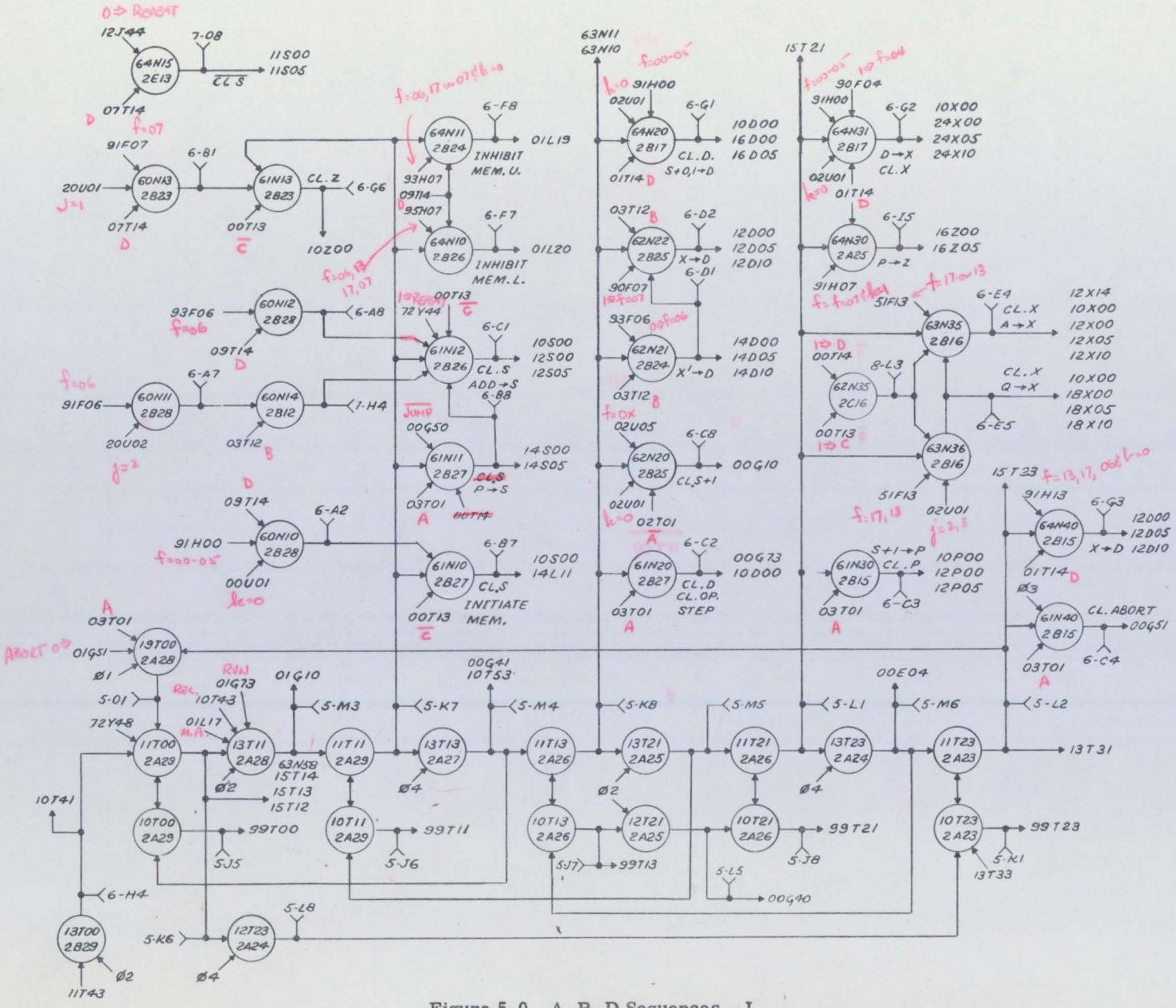


Figure 5-9. A, B, D Sequences - I

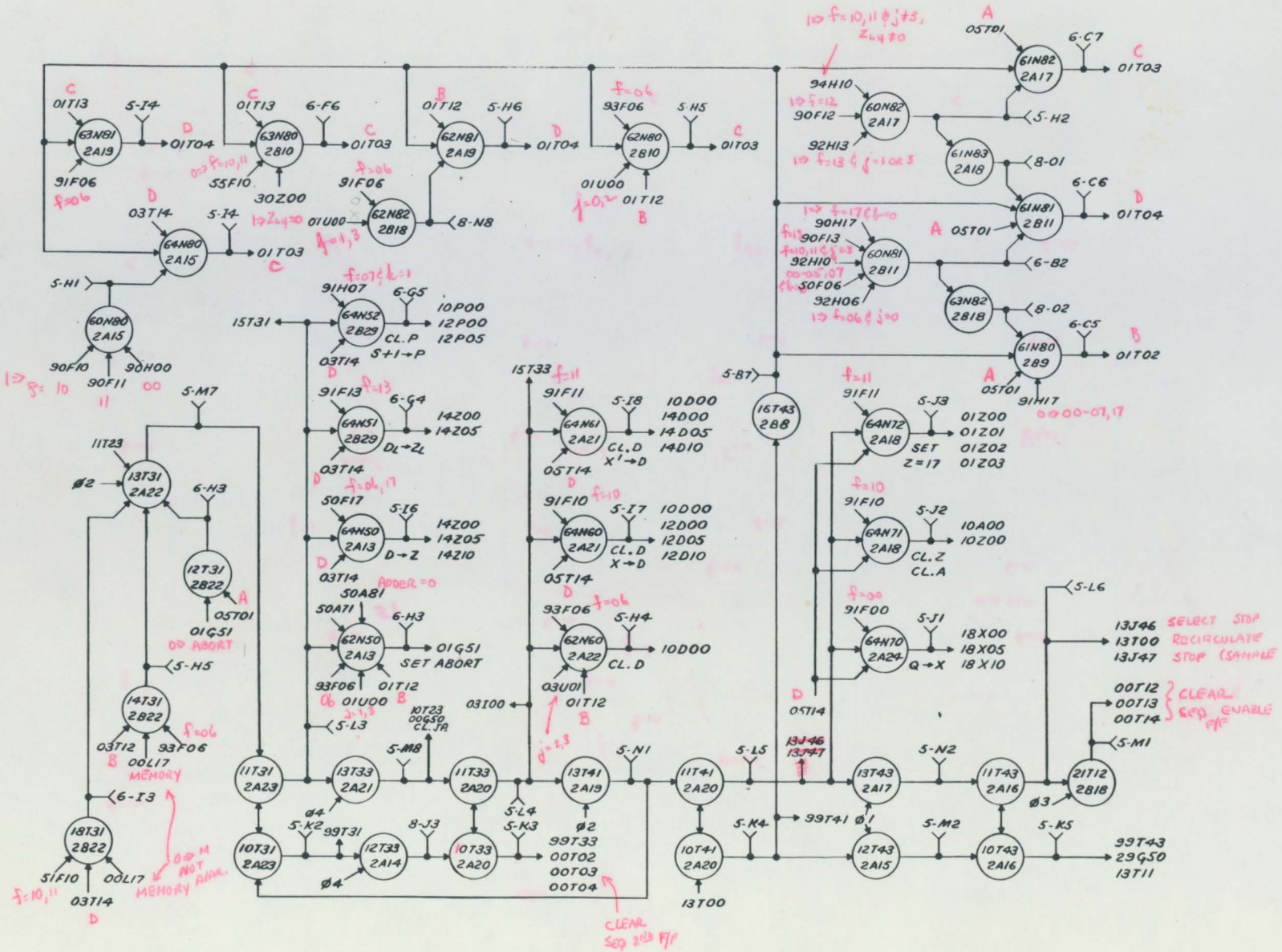


Figure 5-10. A, B, D Sequences - II

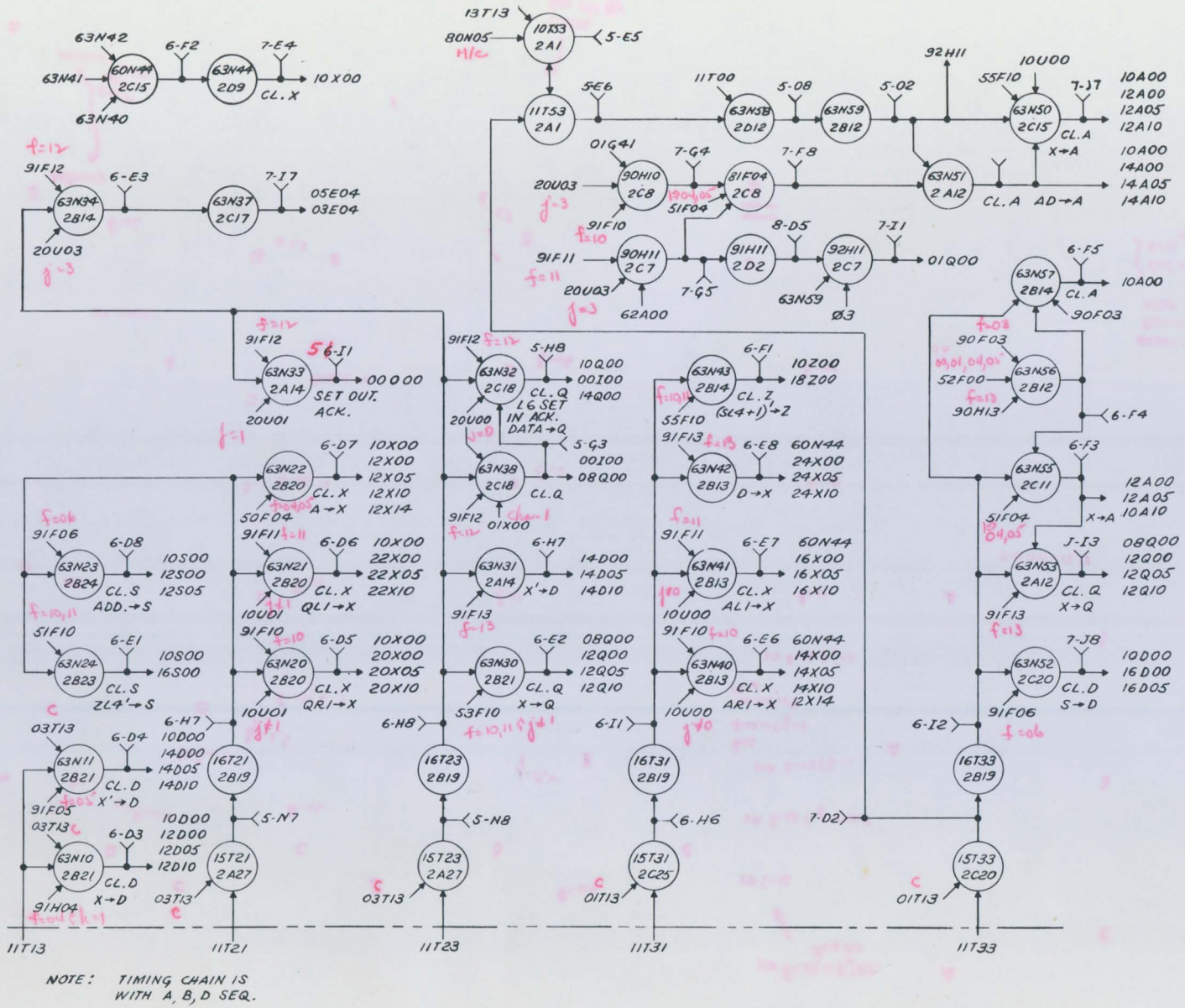


Figure 5-11. C Sequence

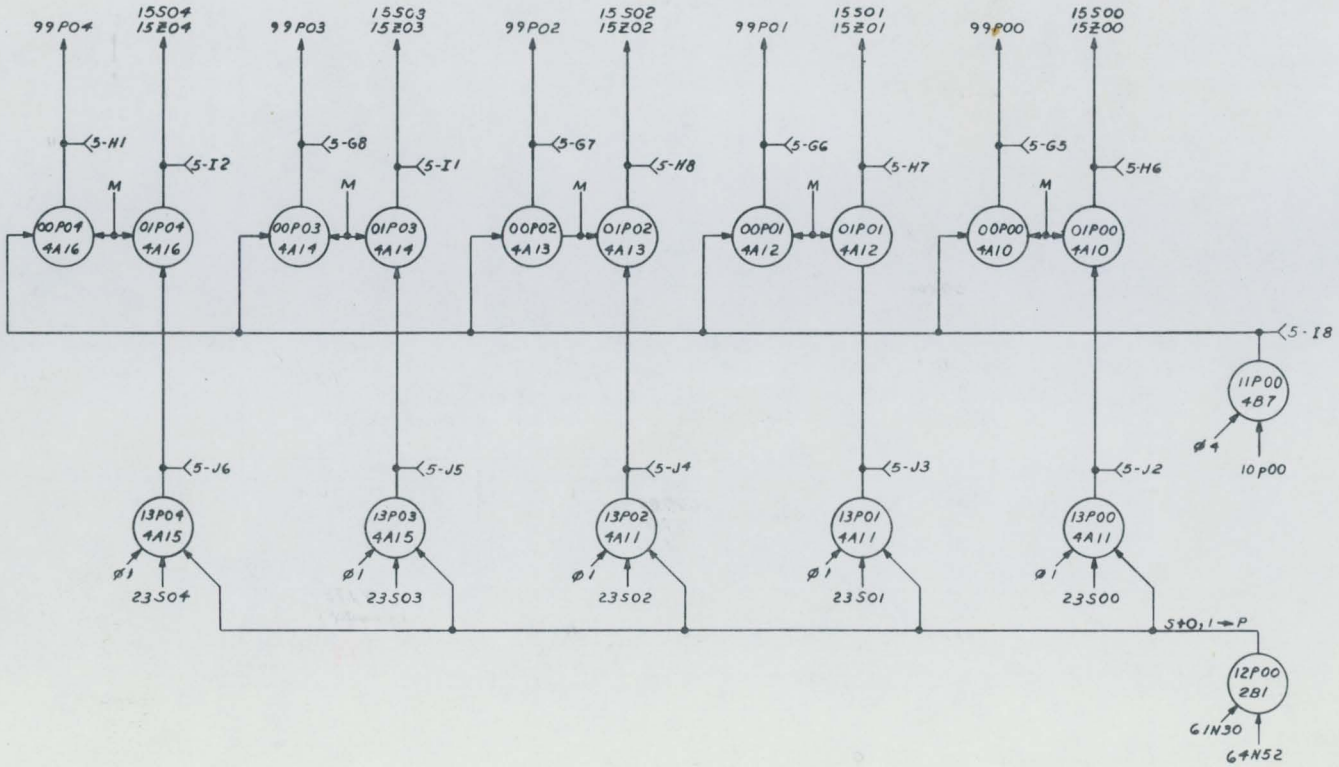


Figure 5-12. P Register, Bits 00-04

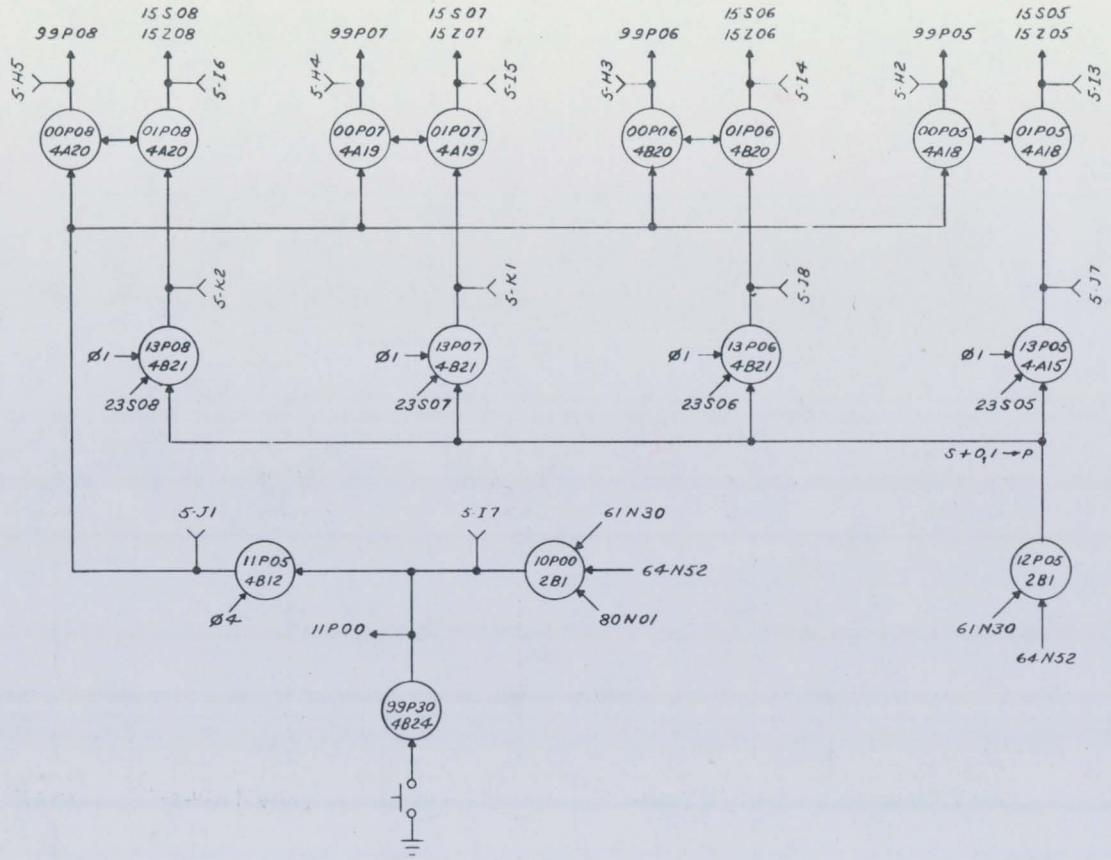


Figure 5-13. P Register, Bits 05-08

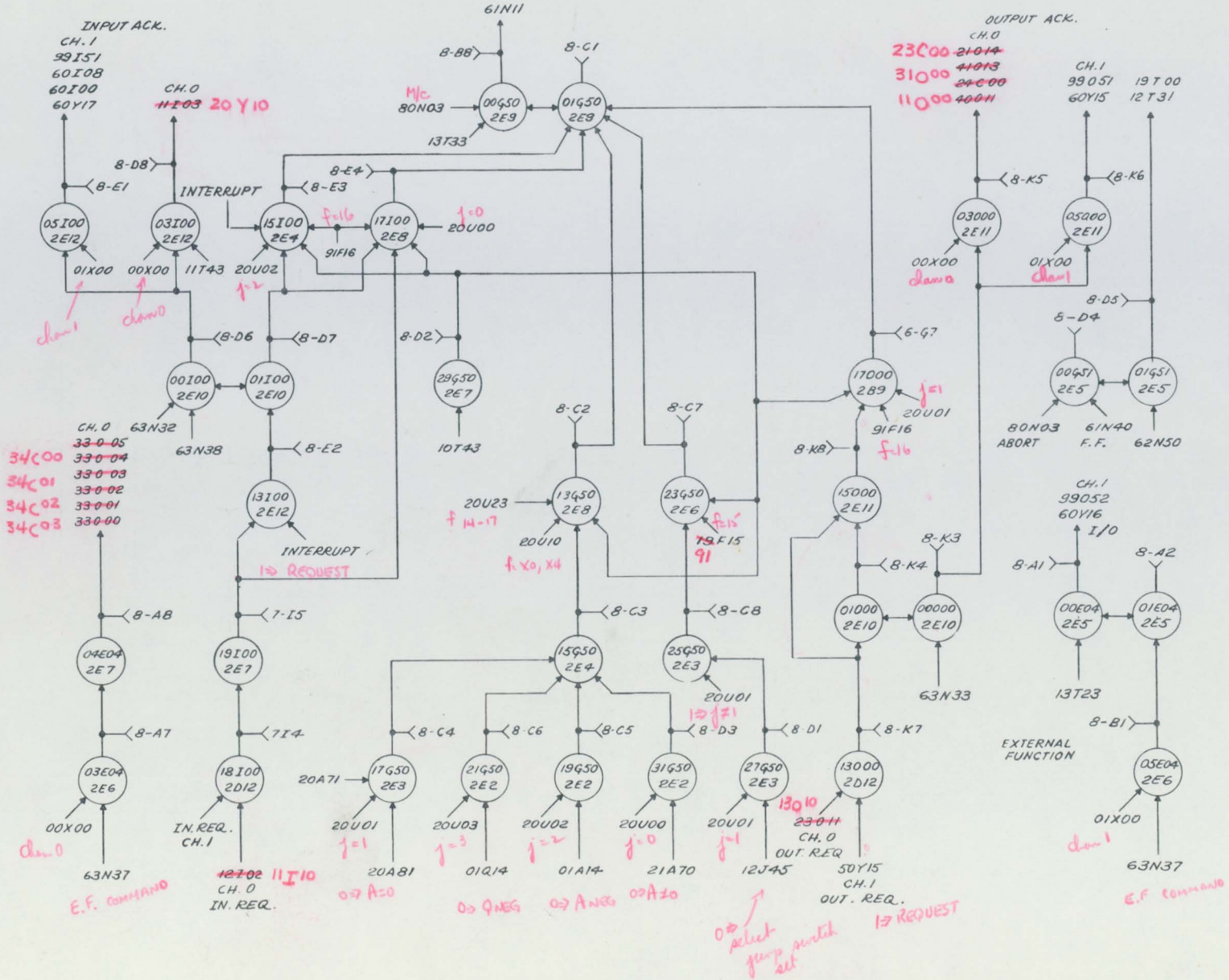


Figure 5-14. Input/Output Jump-Skip

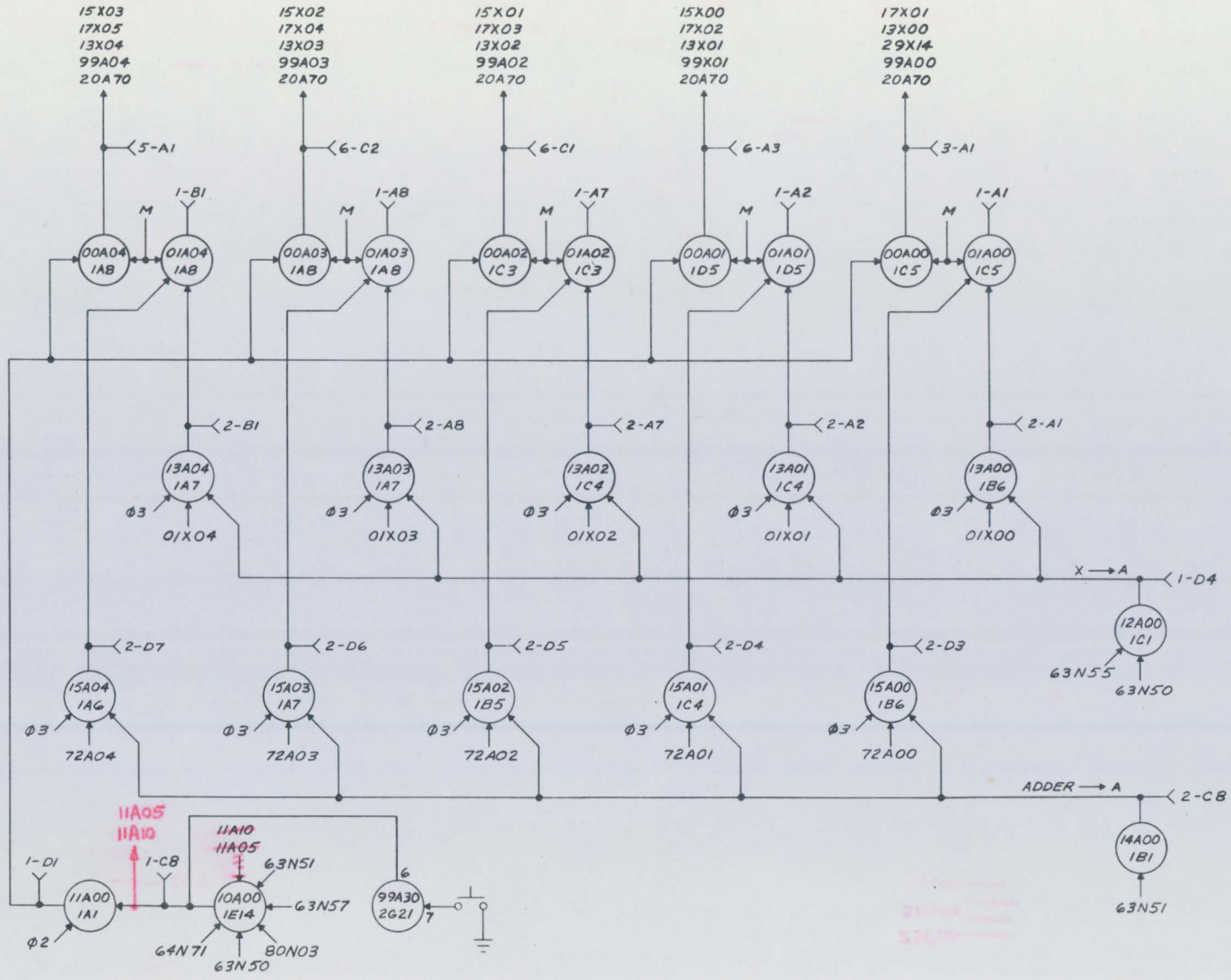


Figure 5-15. A Register, Bits 00-04

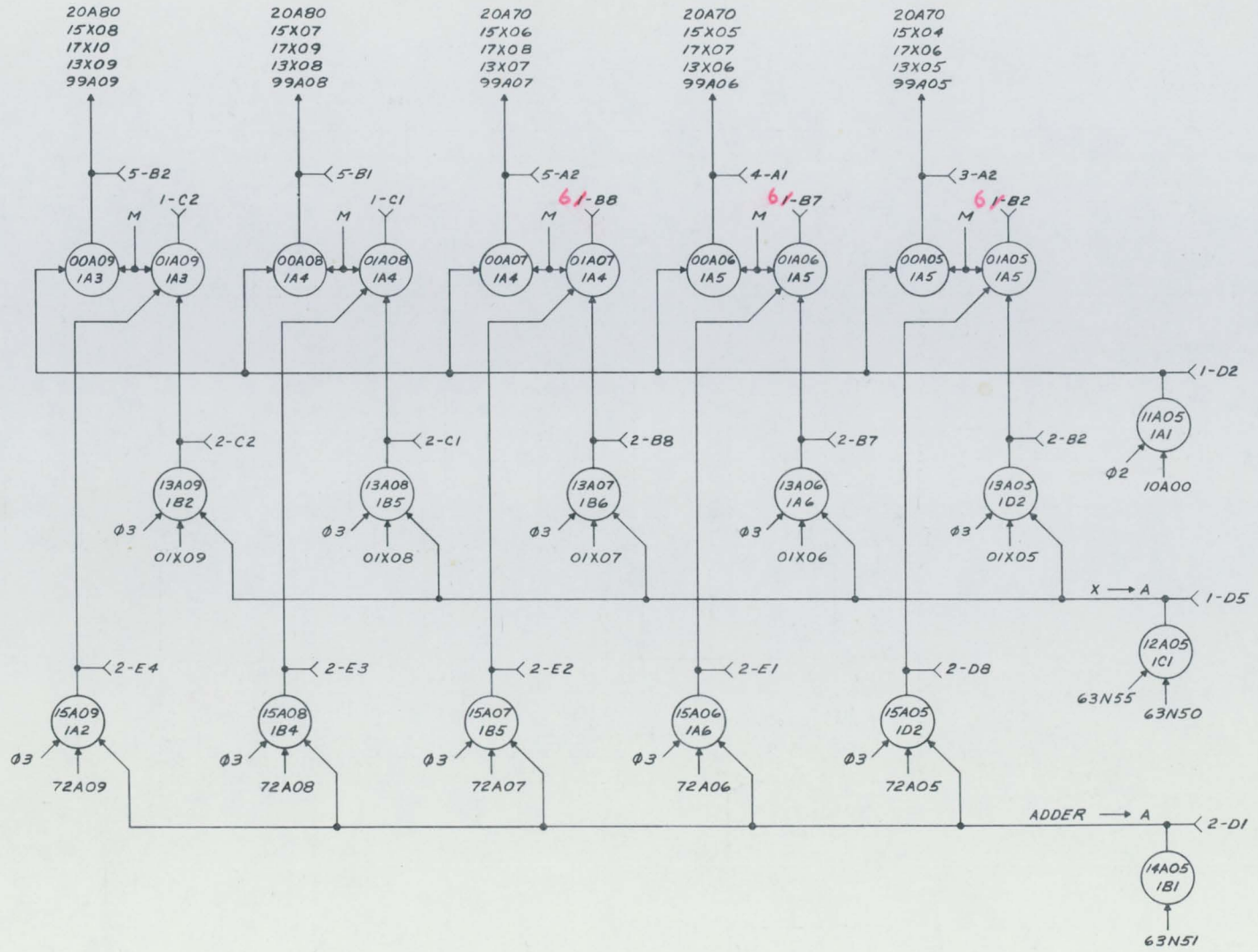


Figure 5-16. A Register, Bits 05-09

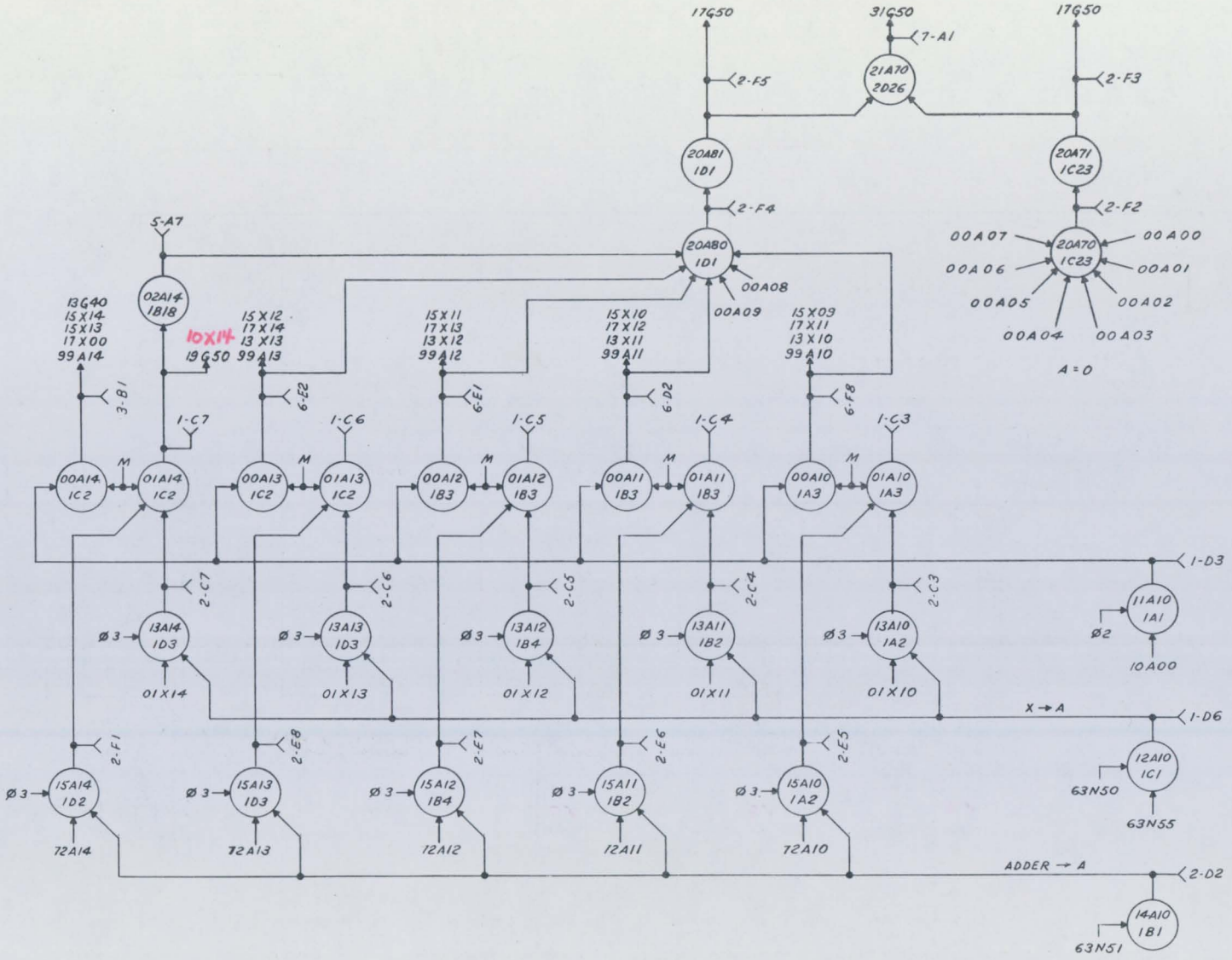


Figure 5-17. A Register, Bits 10-14

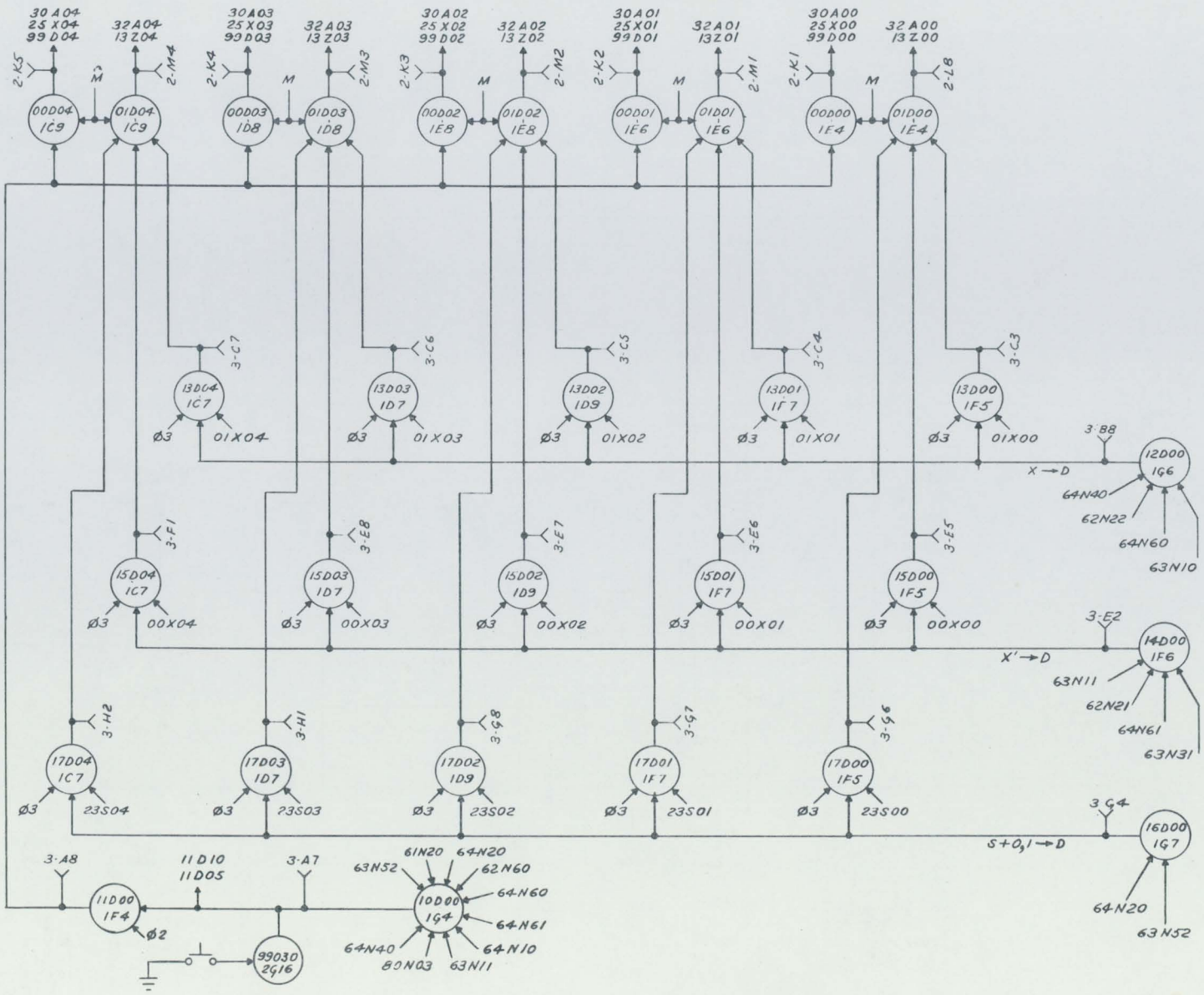


Figure 5-18. D Register, Bits 00-04

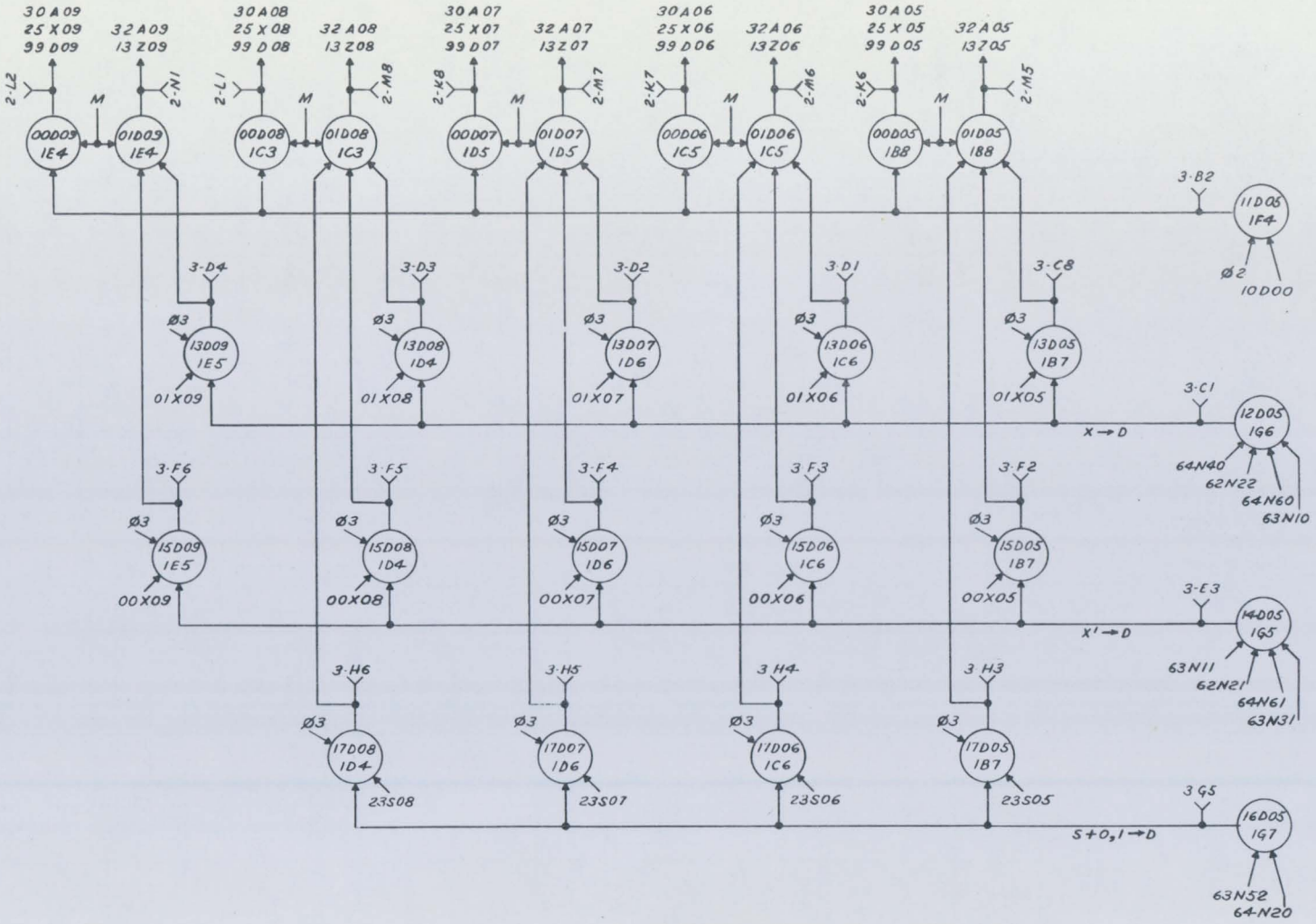


Figure 5-19. D Register, Bits 05-09

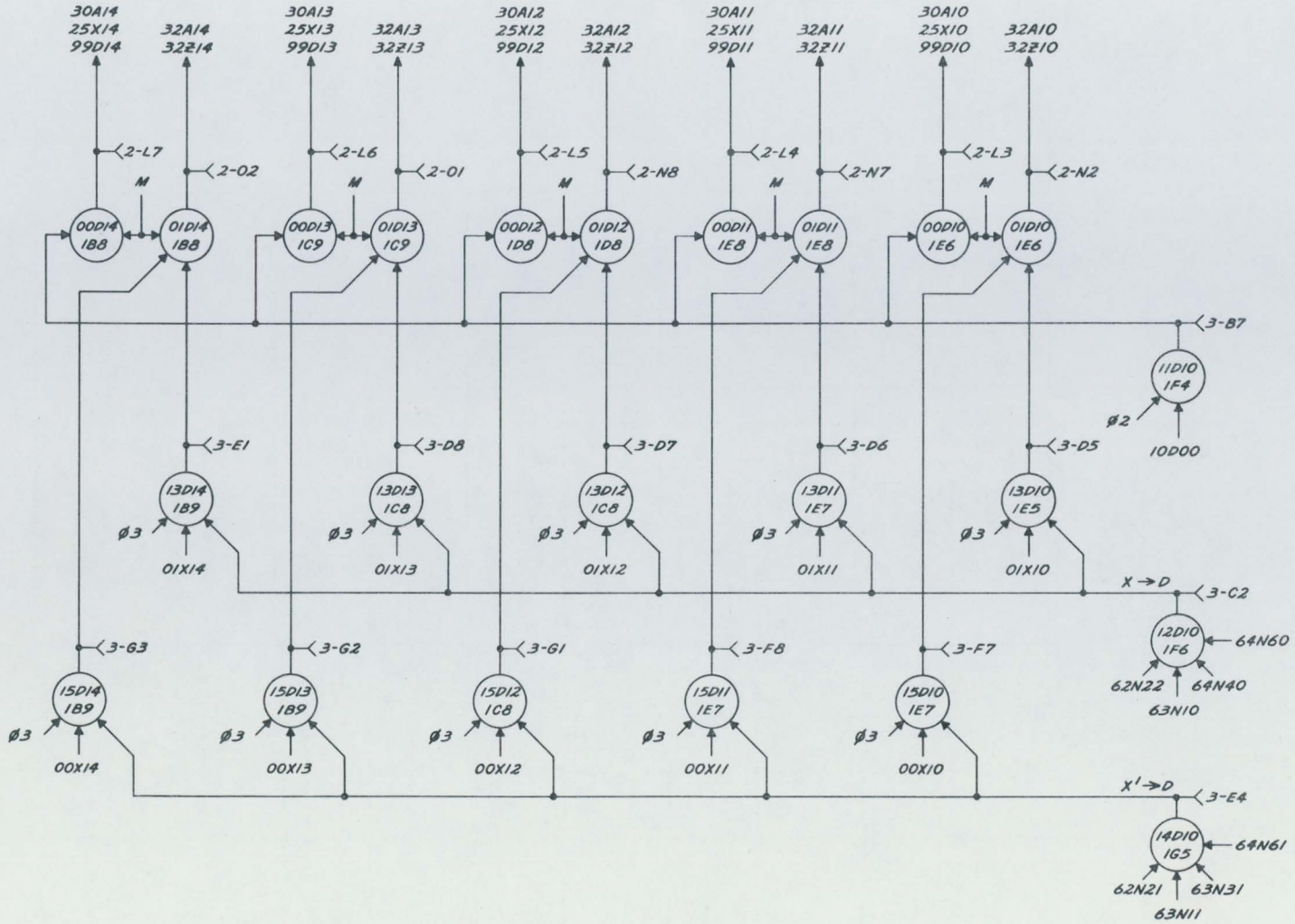


Figure 5-20. D Register, Bits 10-14

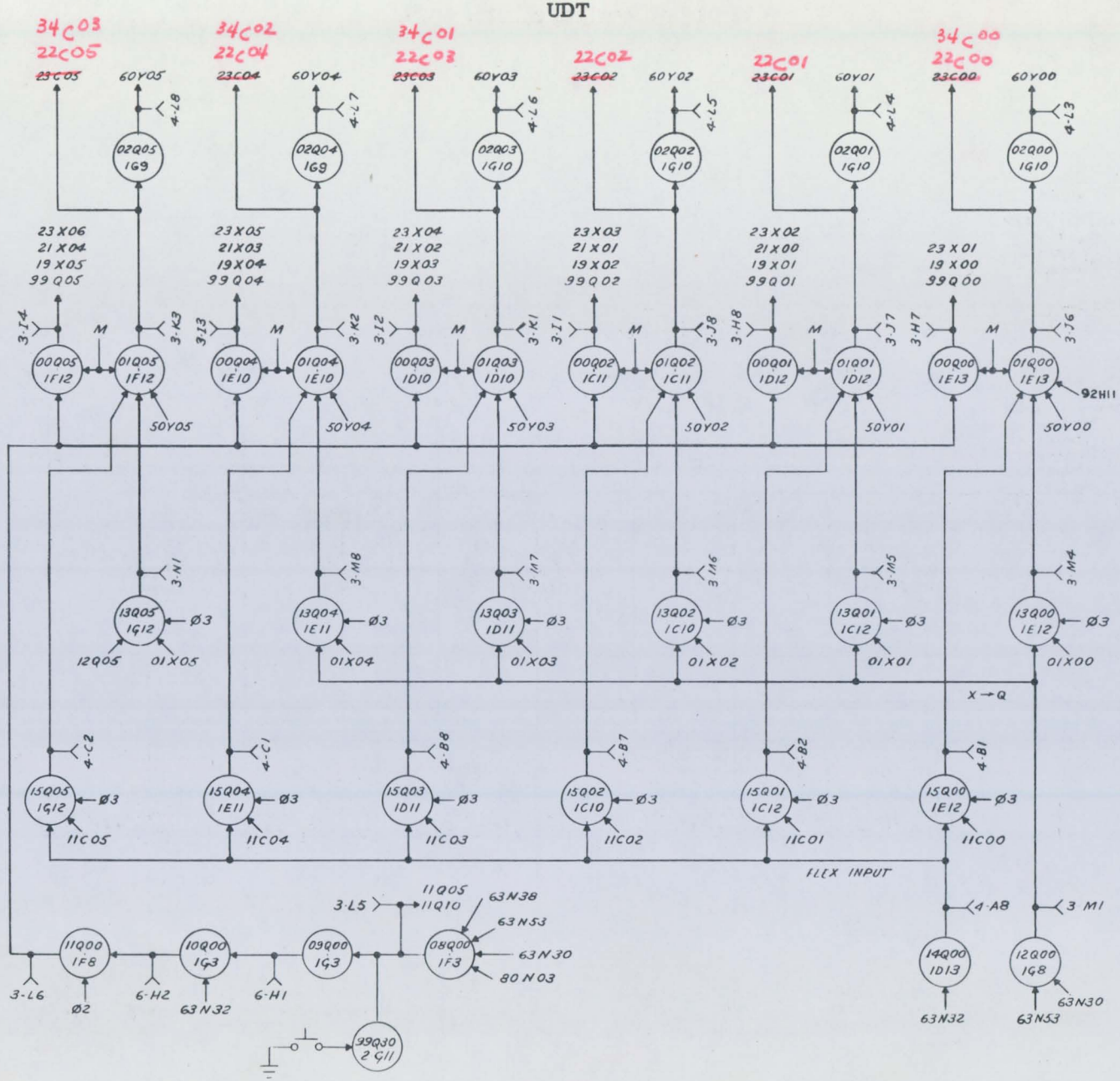


Figure 5-21. Q Register, Bits 00-05

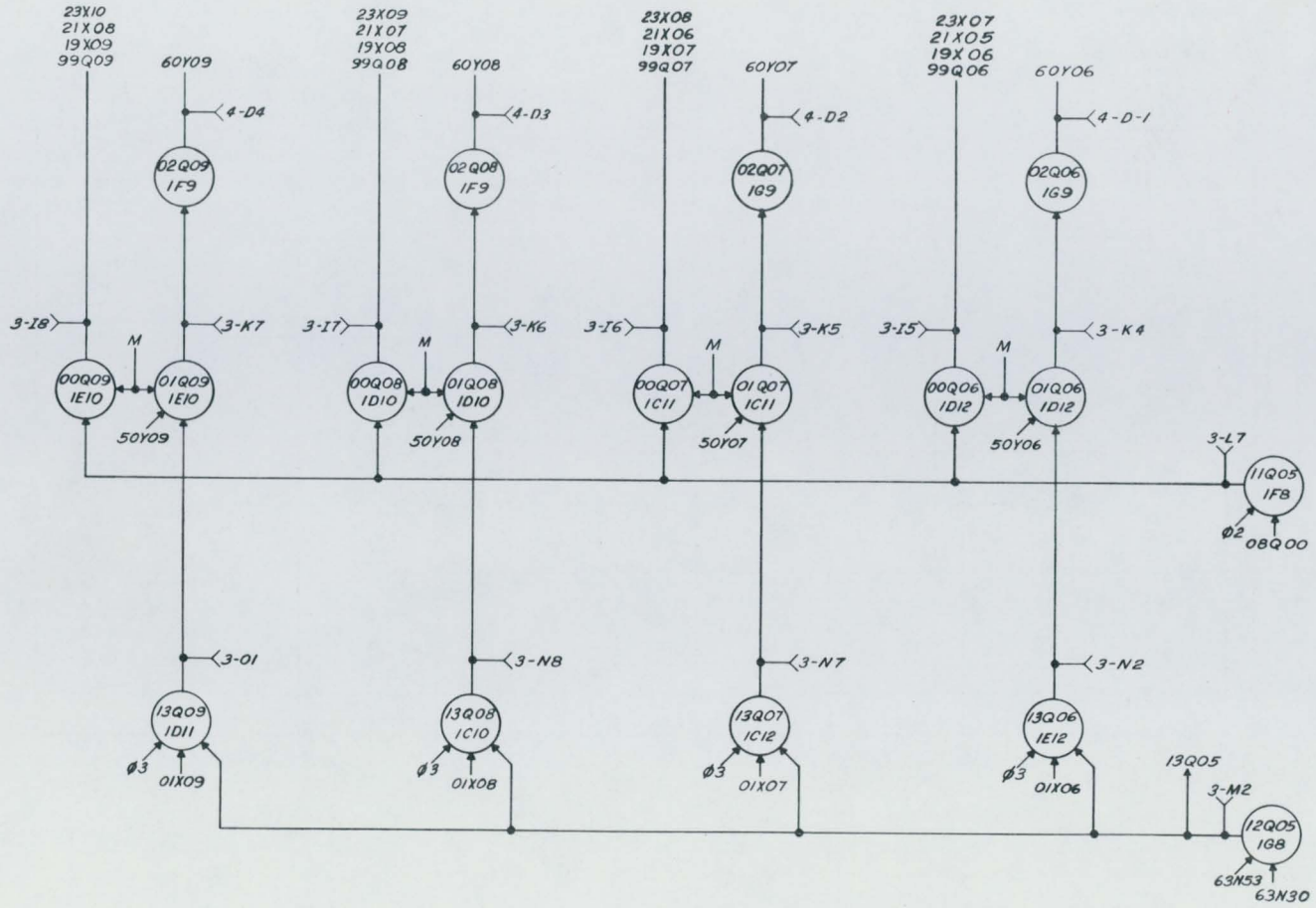


Figure 5-22. Q Register, Bits 06-09

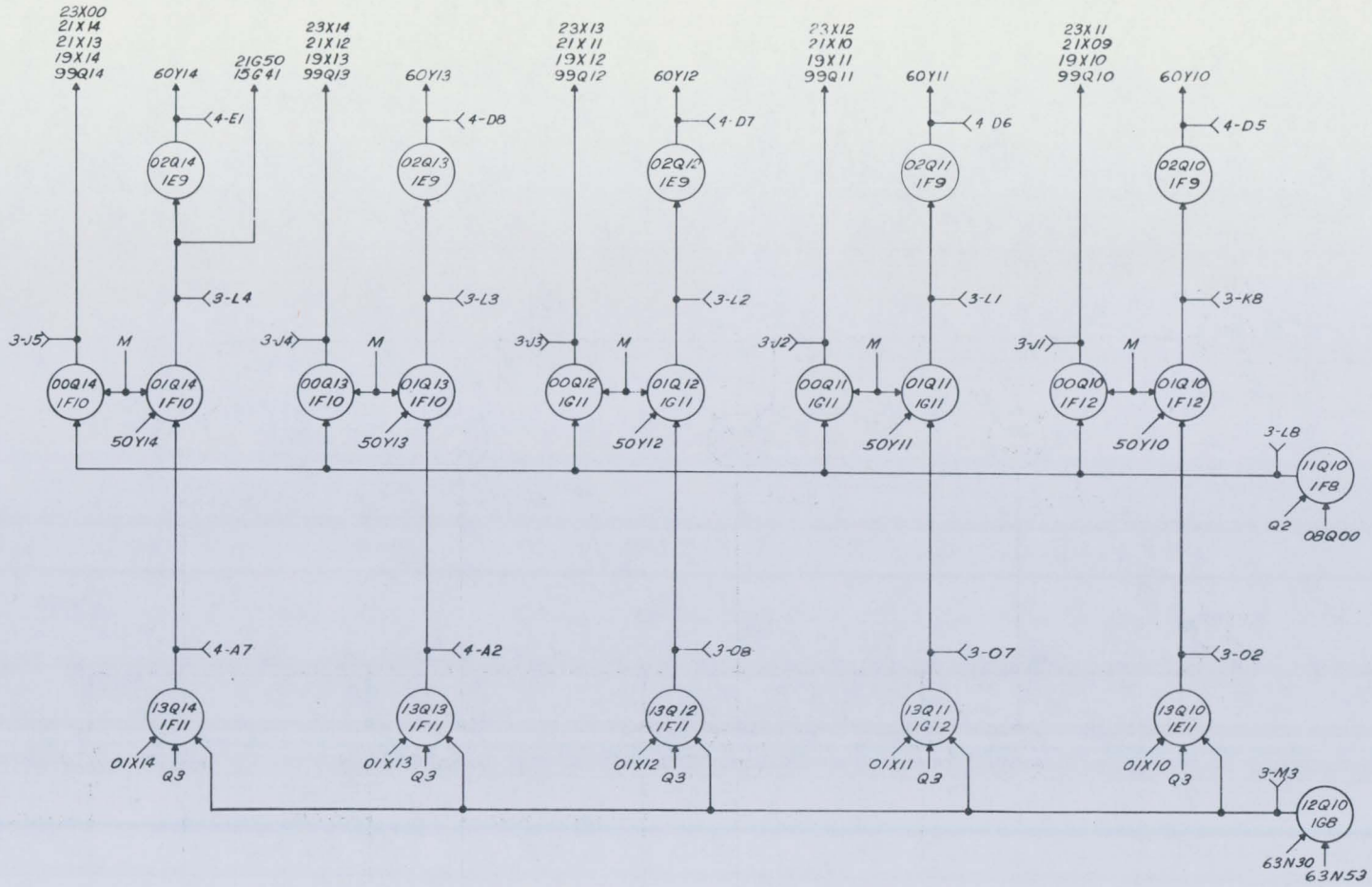


Figure 5-23. Q Register, Bits 10-14

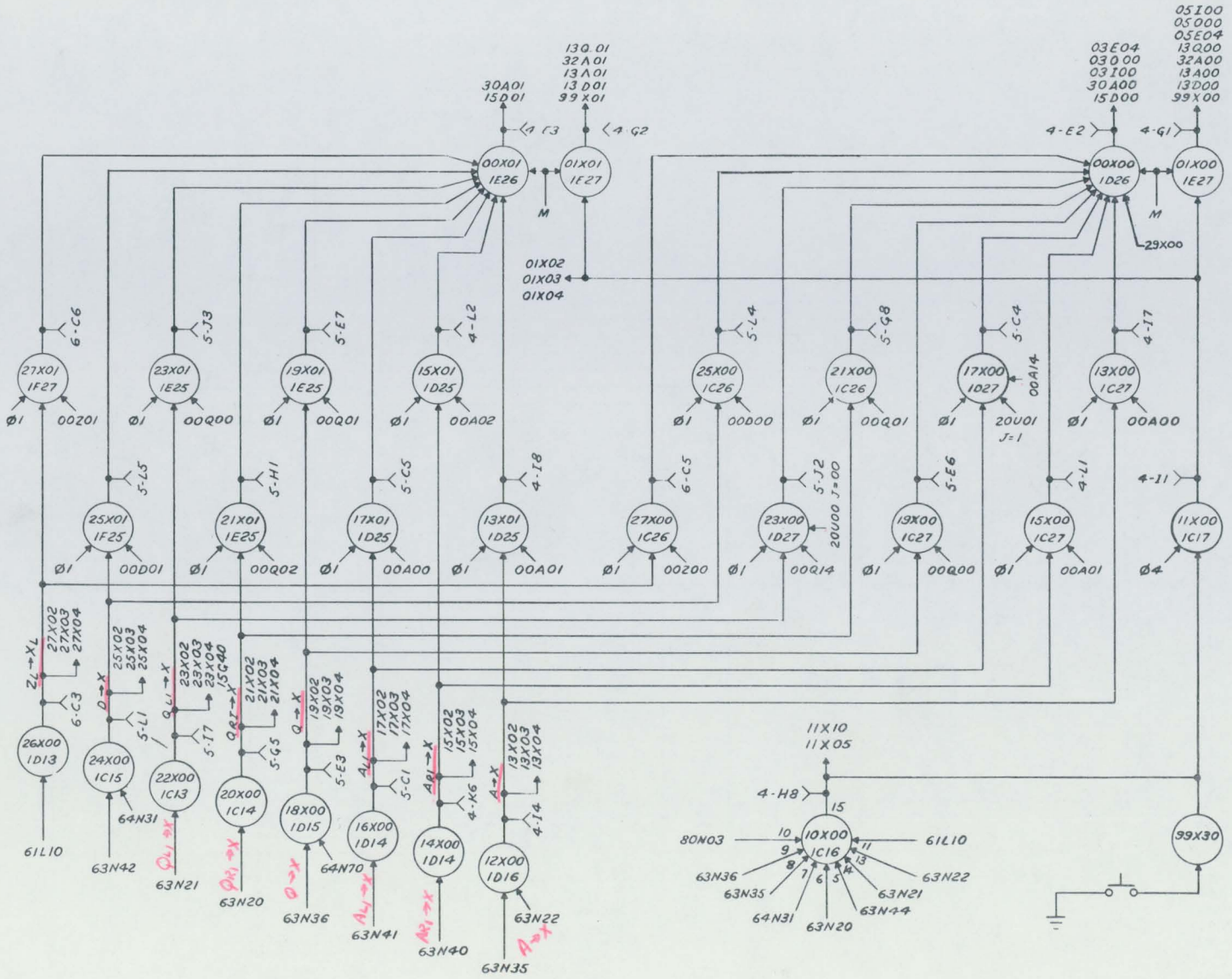


Figure 5-24. X Register, Bits 00,01

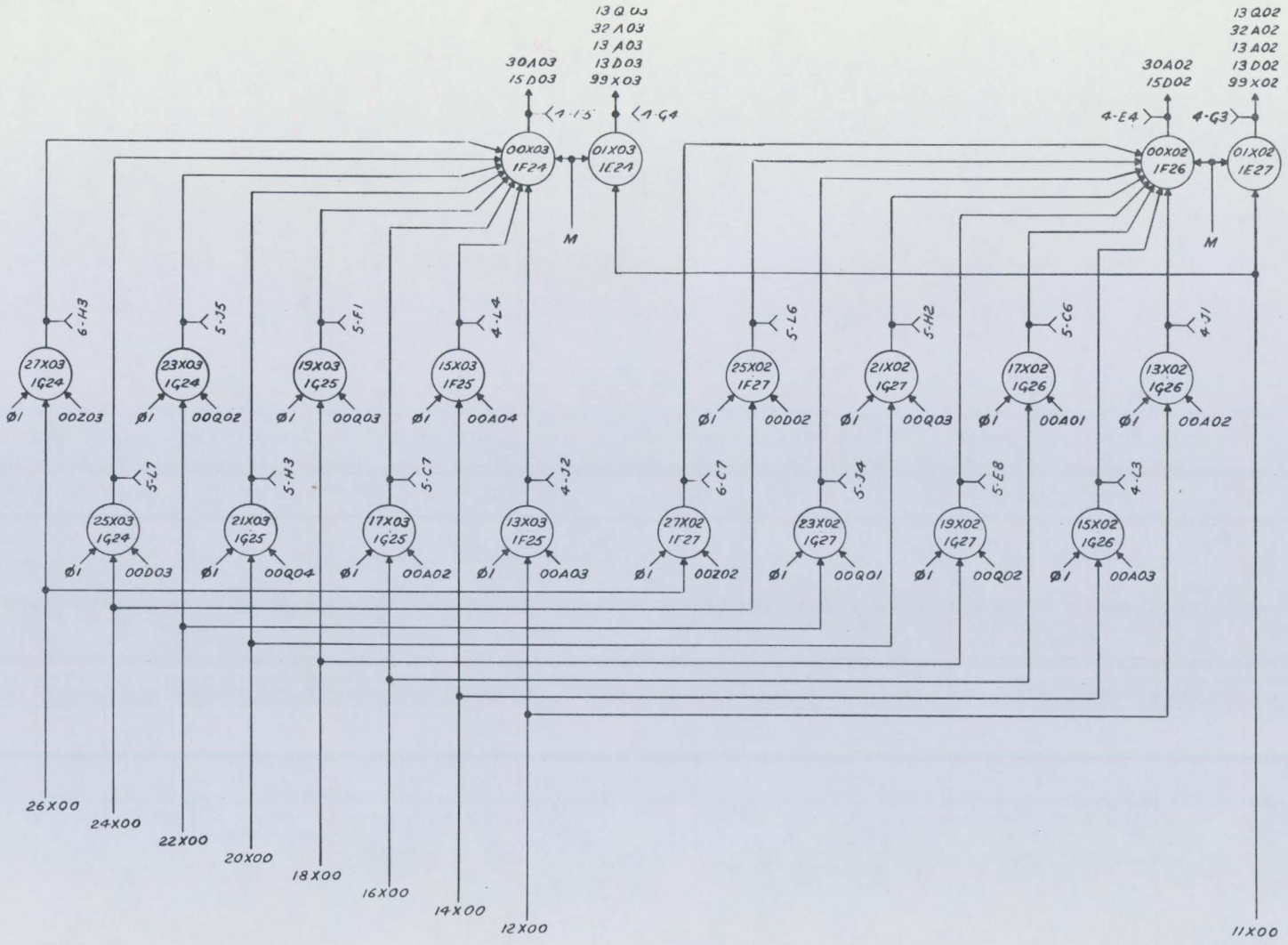


Figure 5-25. X Register, Bits 02,03

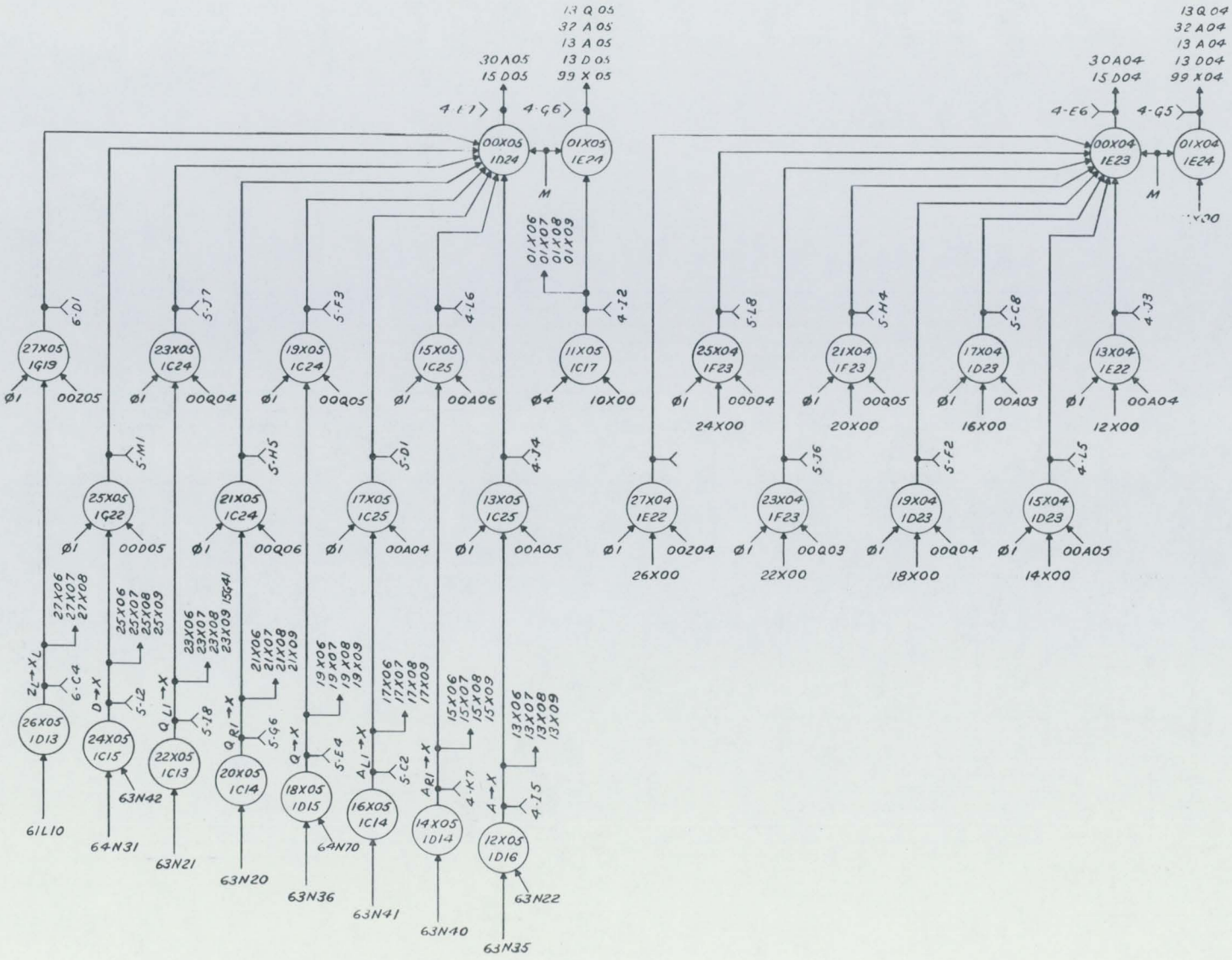


Figure 5-26. X Register, Bits 04,05

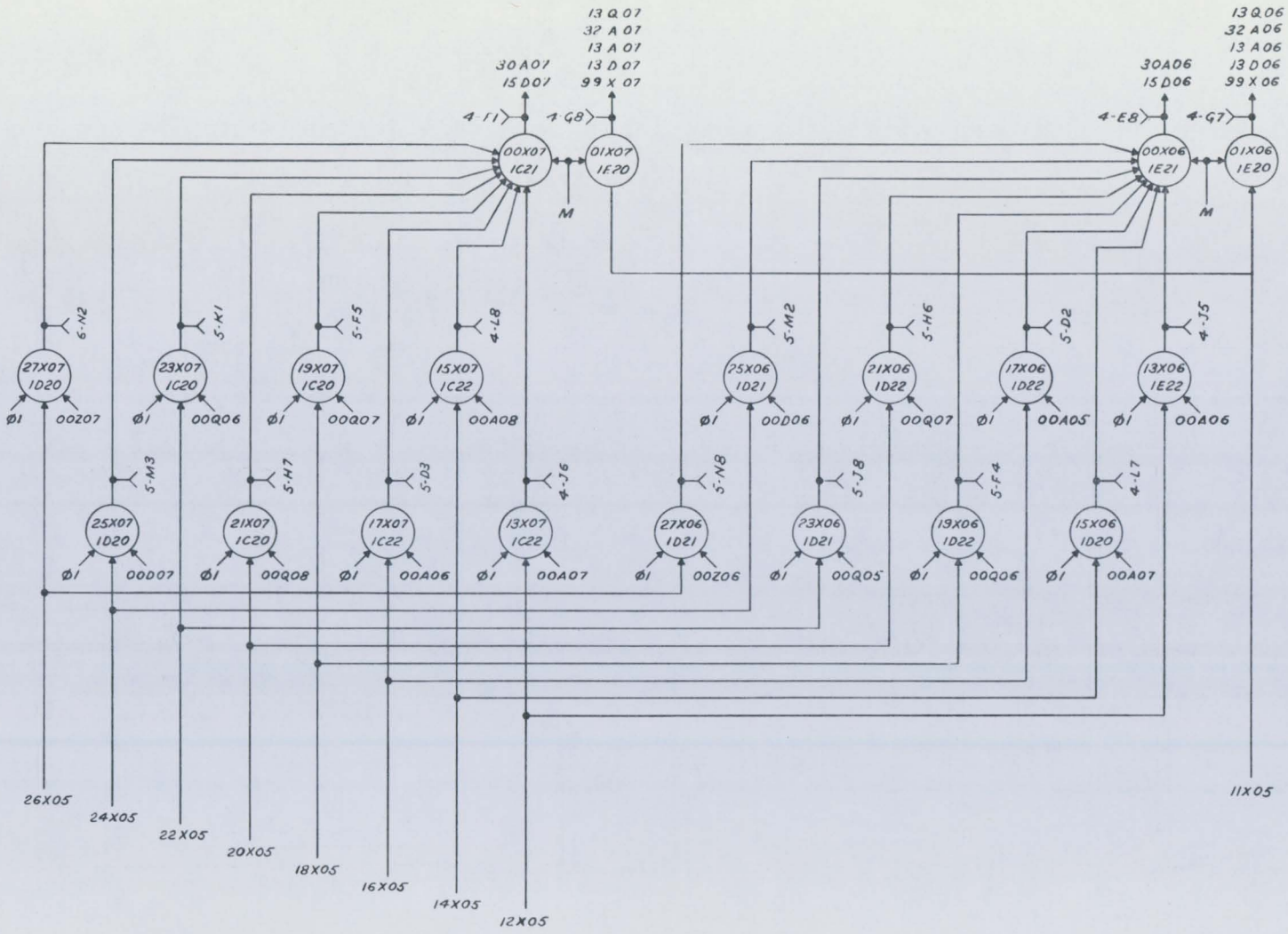


Figure 5-27. X Register, Bits 06,07

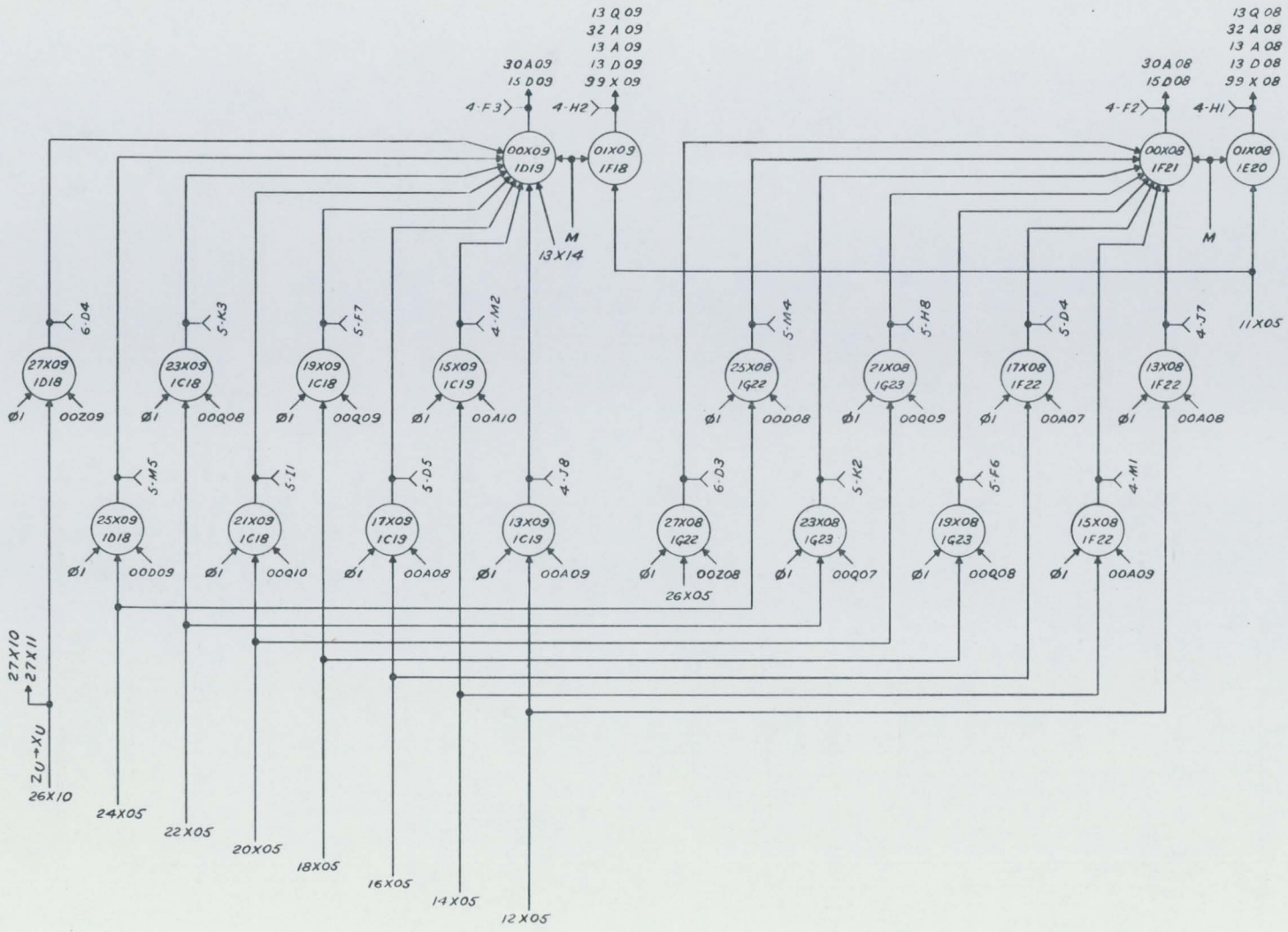


Figure 5-28. X Register, Bits 08,09

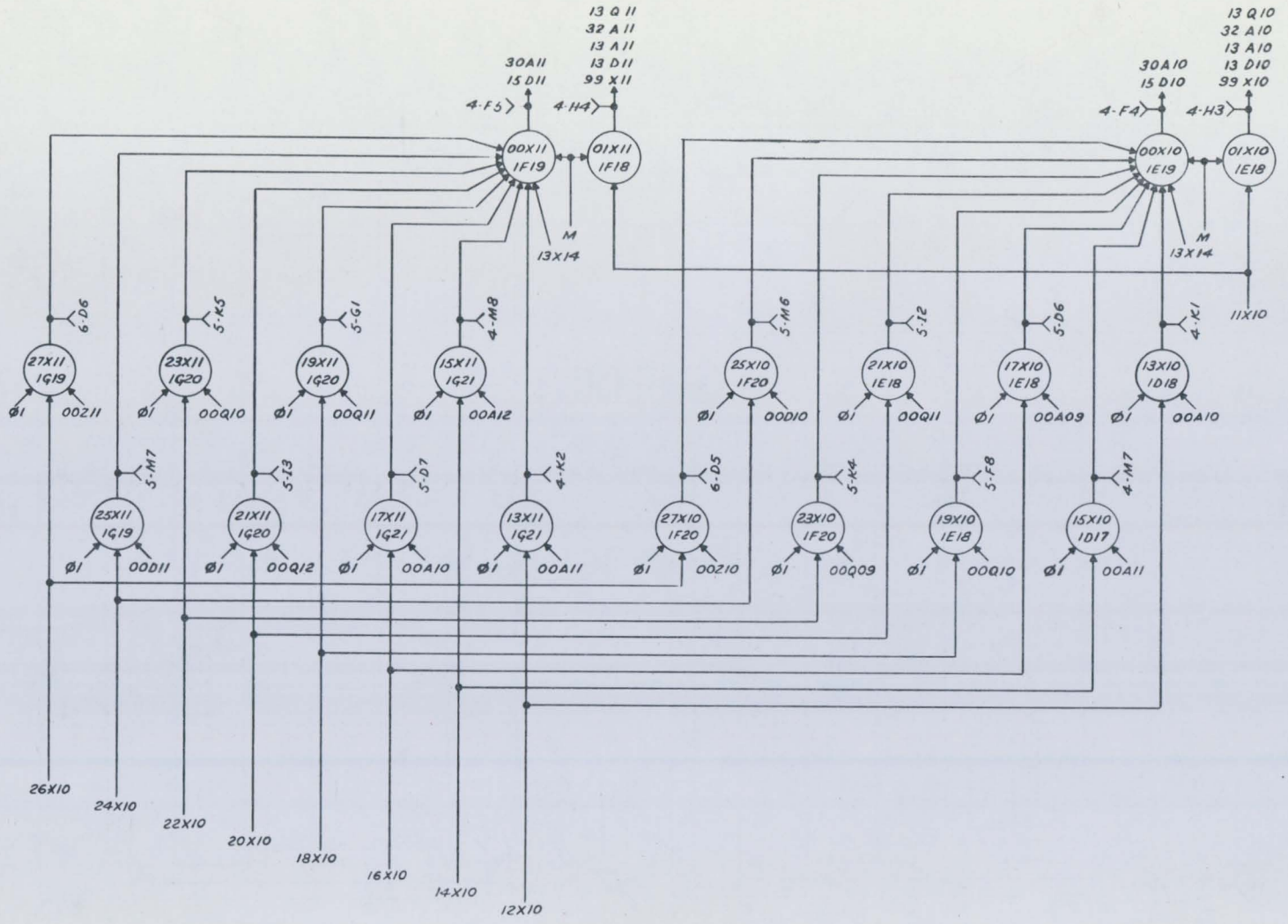


Figure 5-29. X Register, Bits 10,11

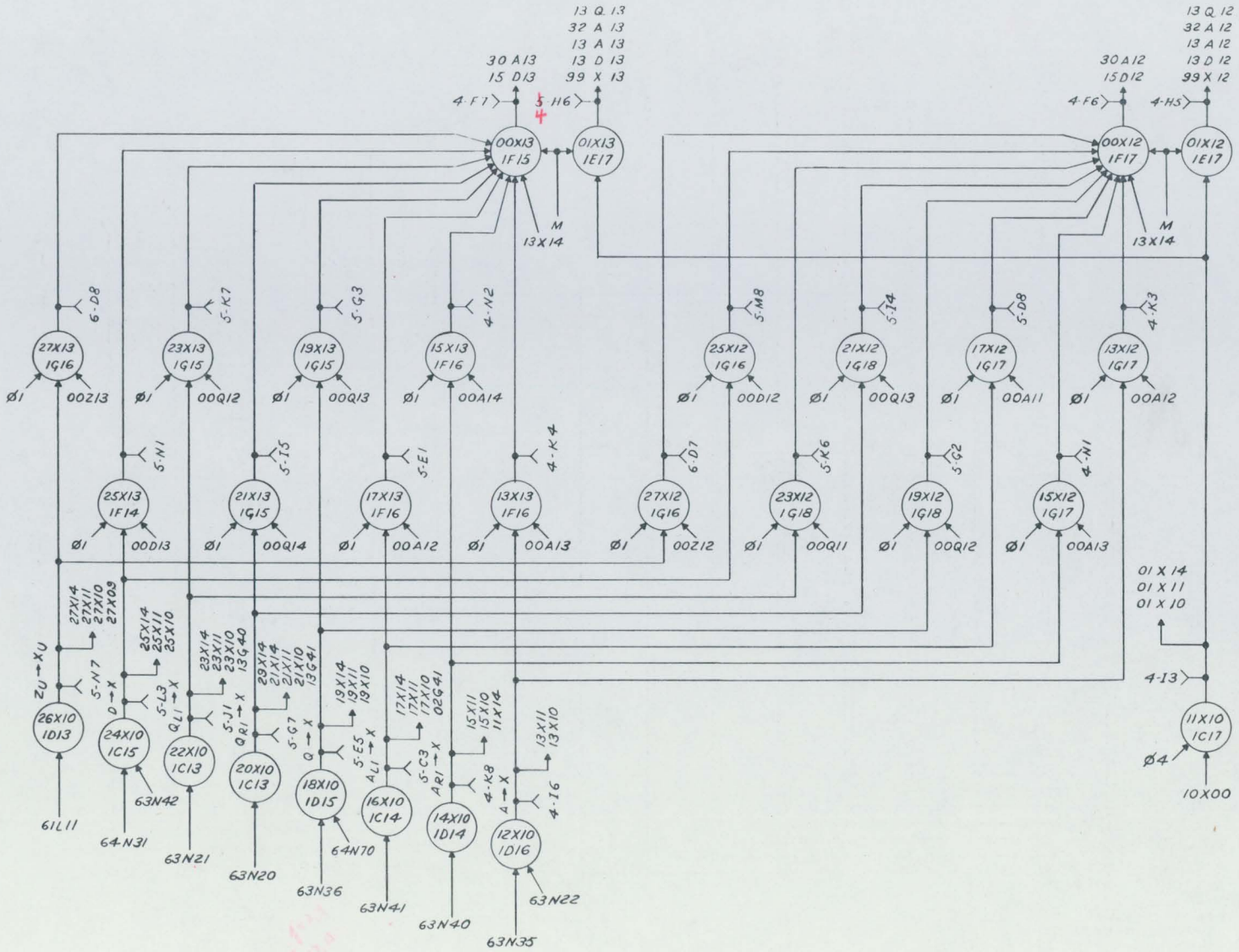


Figure 5-30. X Register, Bits 12,13

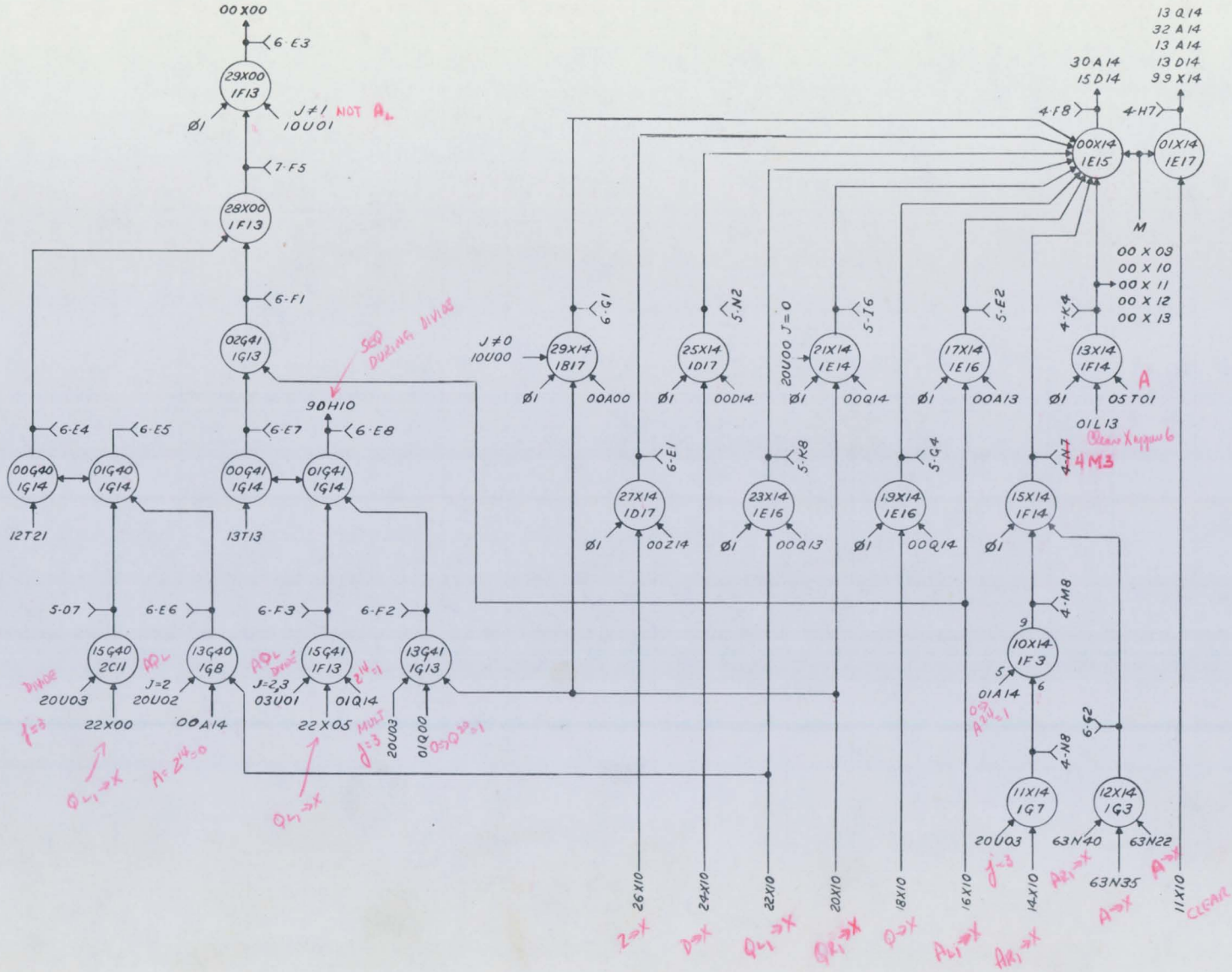


Figure 5-31. X Register, Bit 14

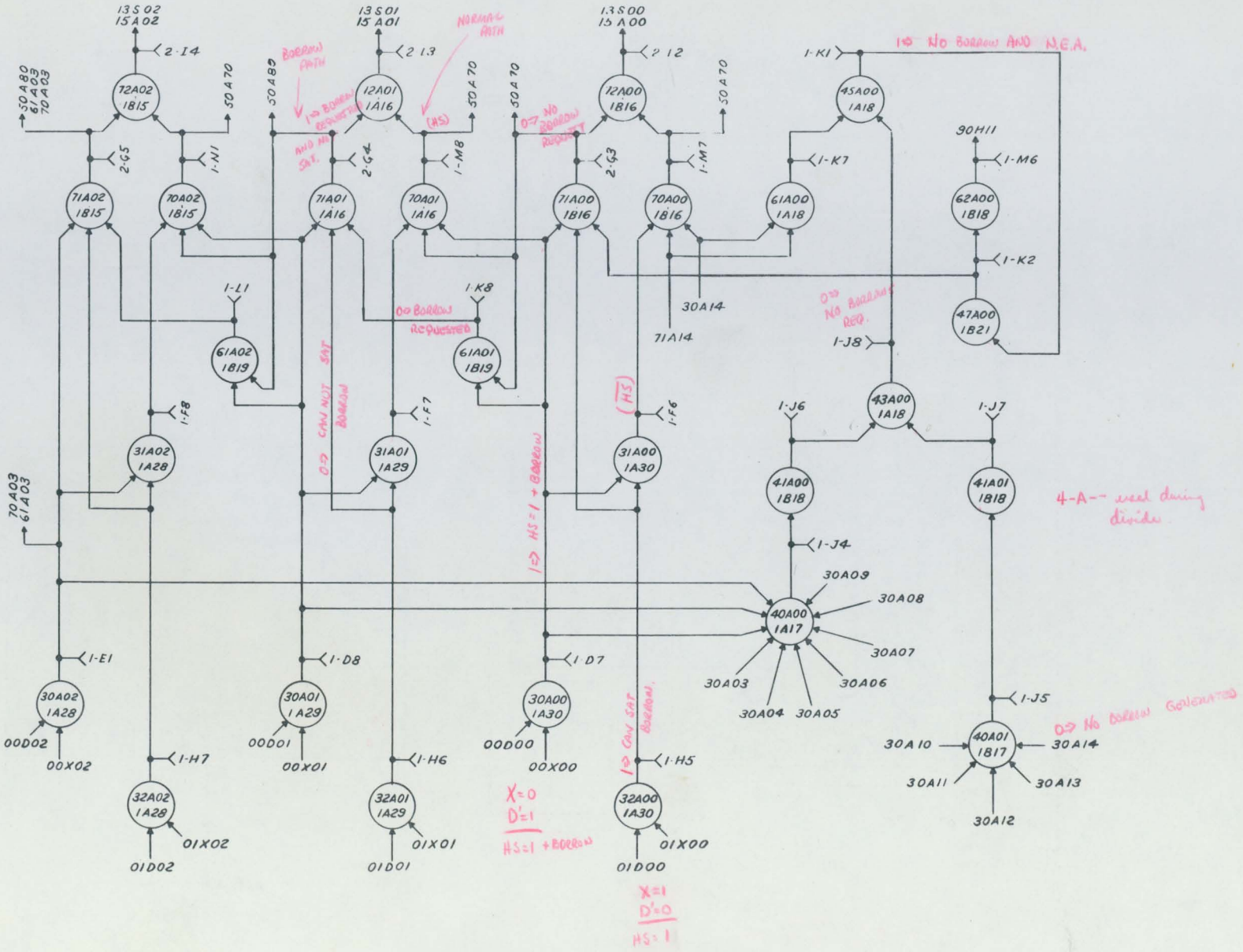


Figure 5-3z. Arithmetic Adder, Stages 00-02

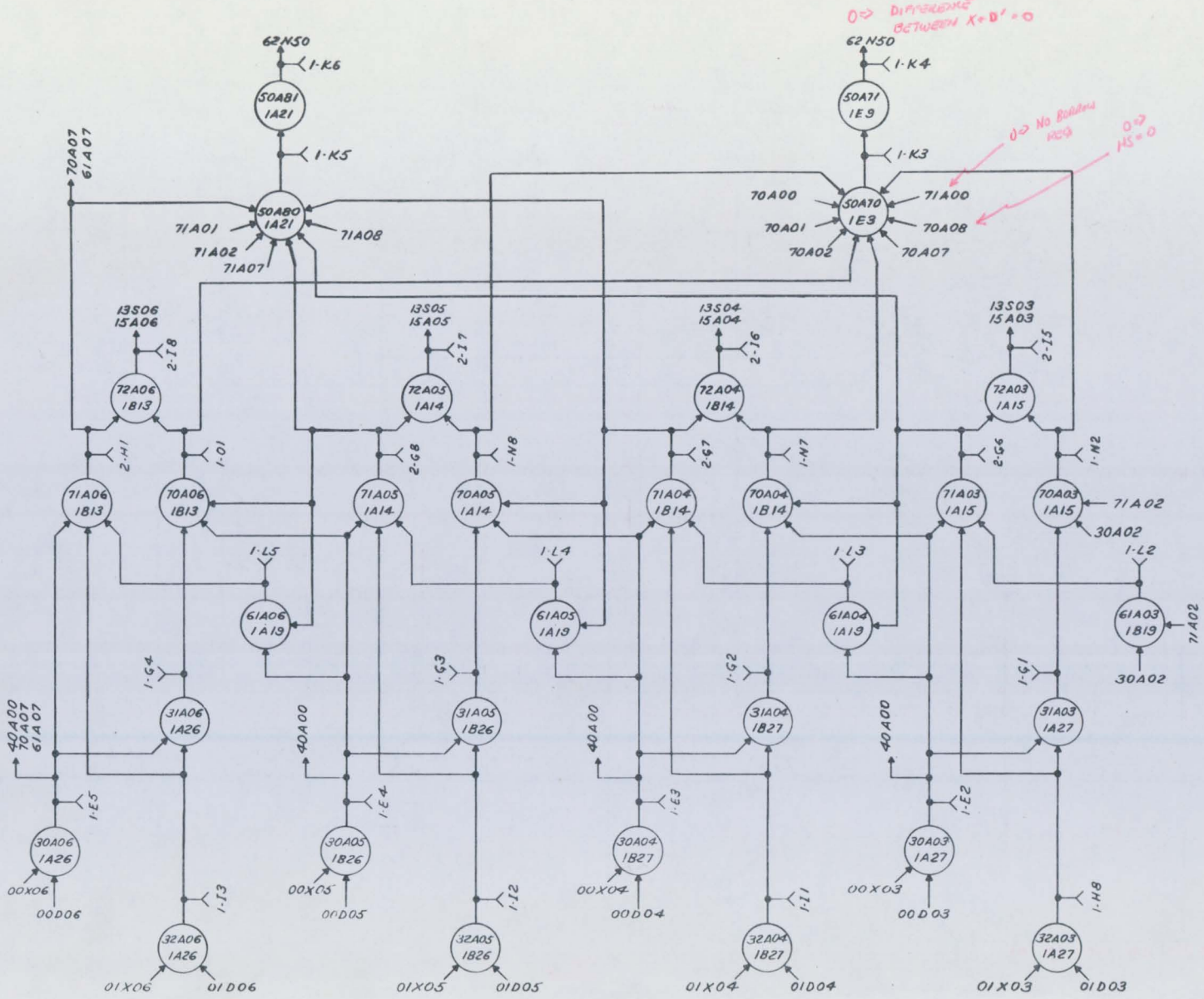


Figure 5-33. Arithmetic Adder, Stages 03-06

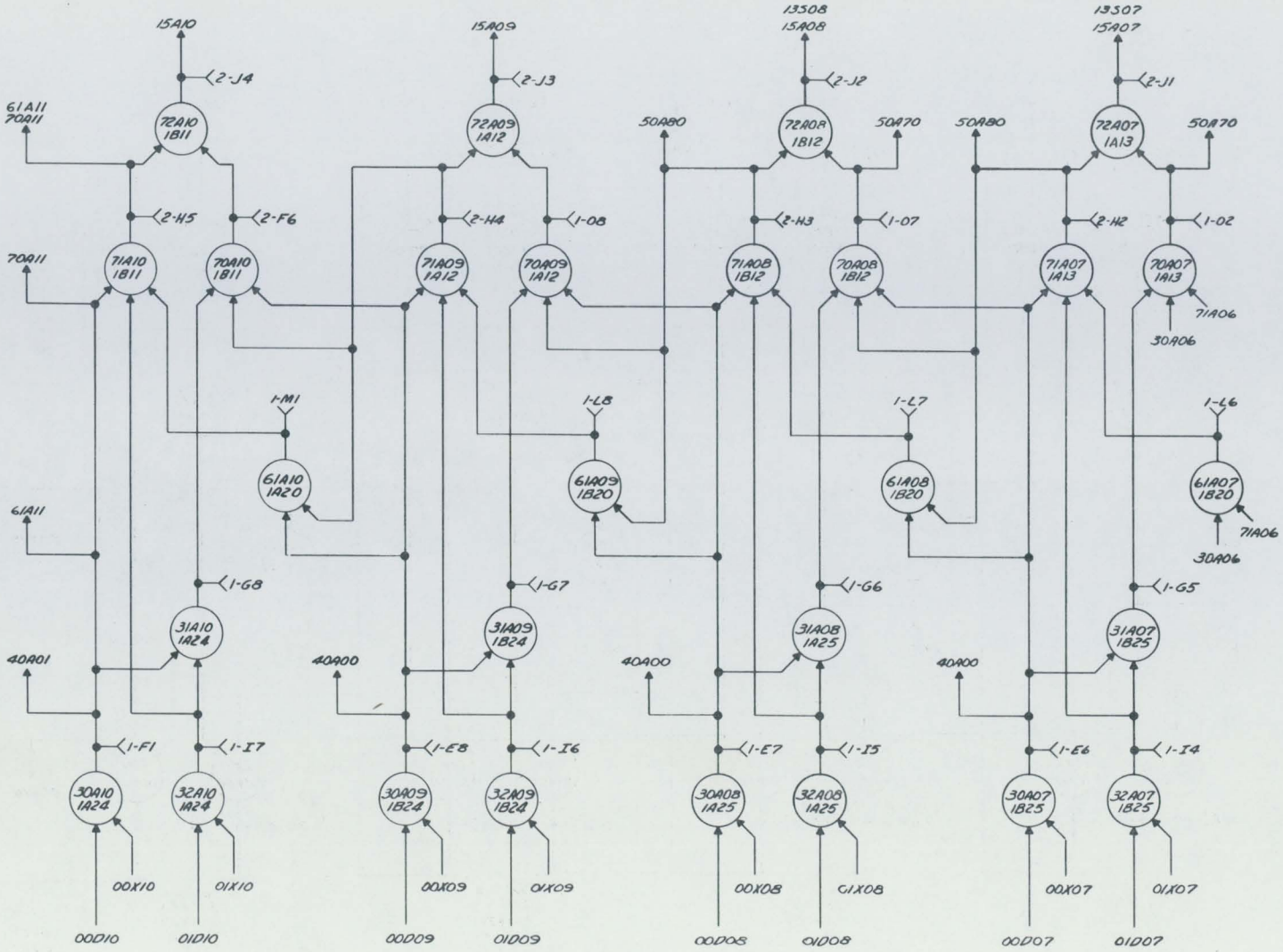


Figure 5-34. Arithmetic Adder, Stages 07-10

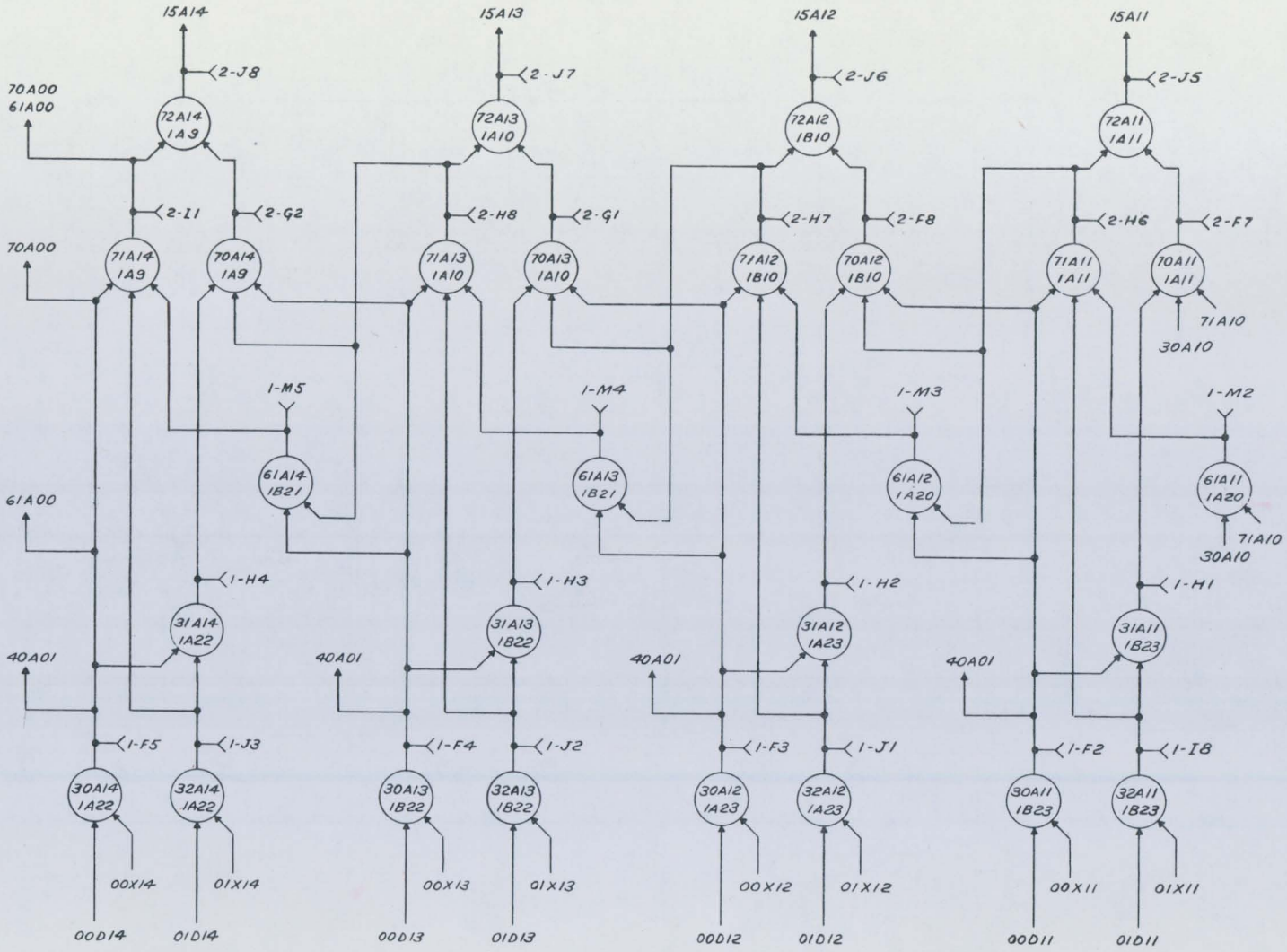


Figure 5-35. Arithmetic Adder, Stages 11-14

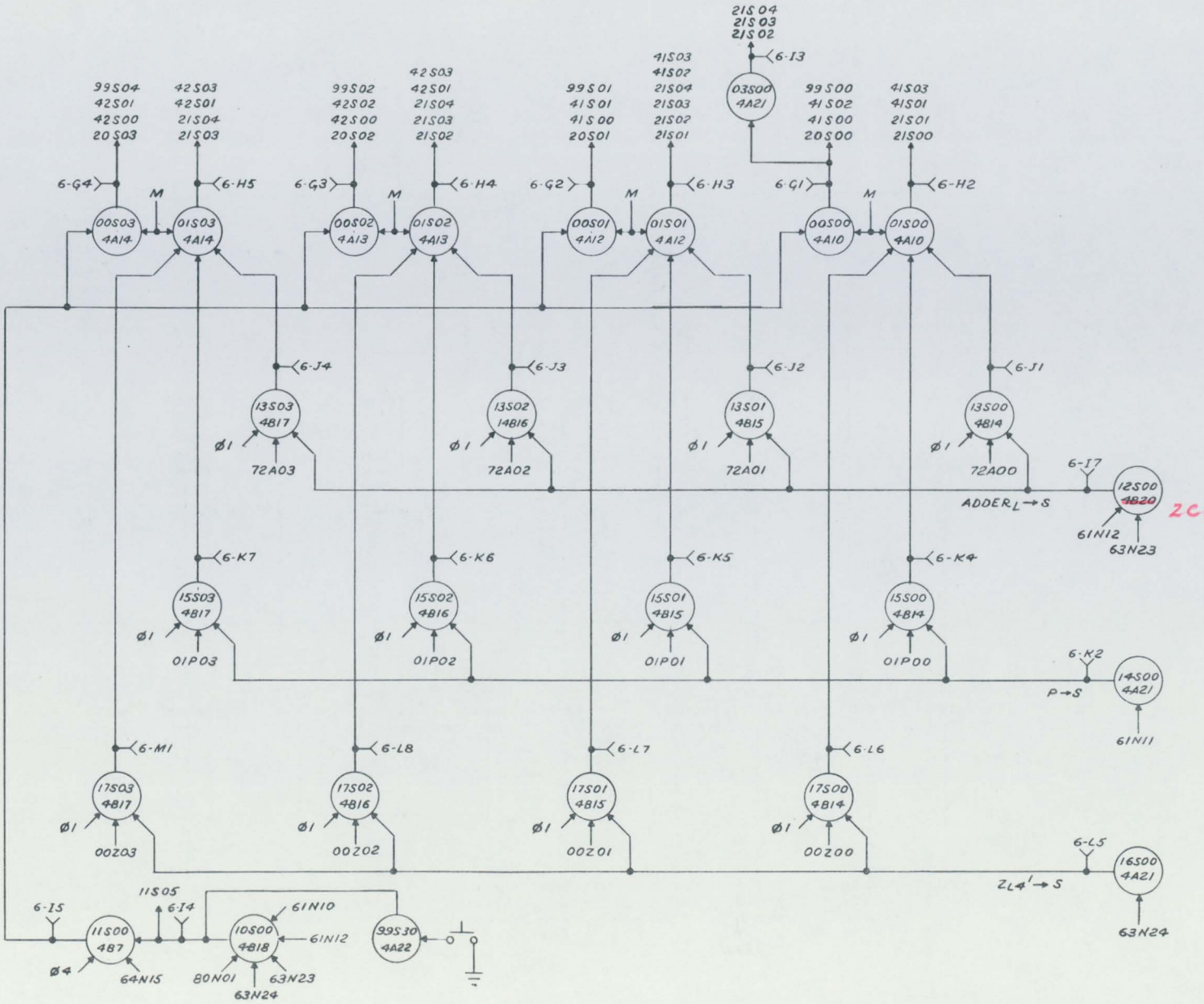


Figure 5-36. S Register, Bits 00-03

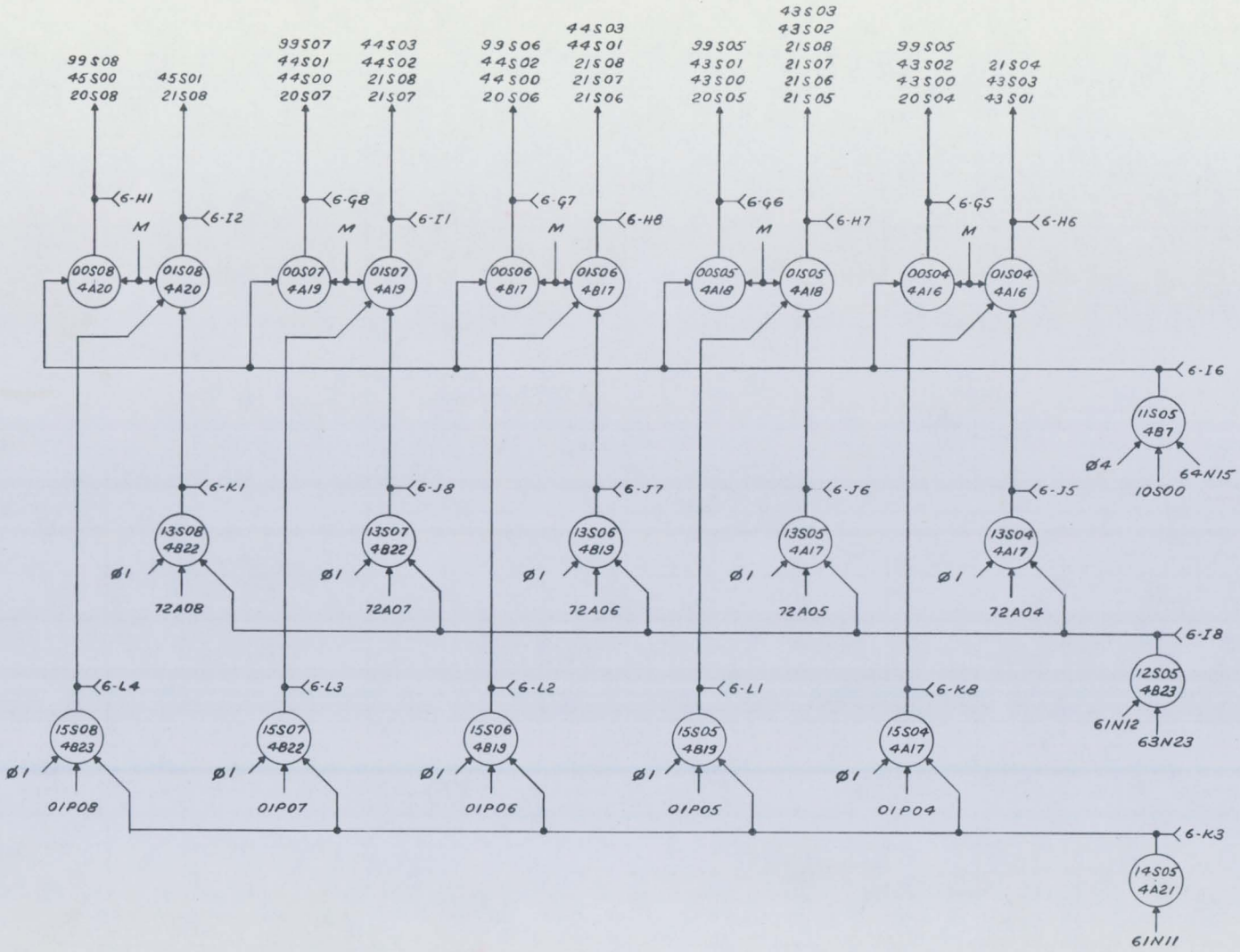


Figure 5-37. S Register, Bits 04-08

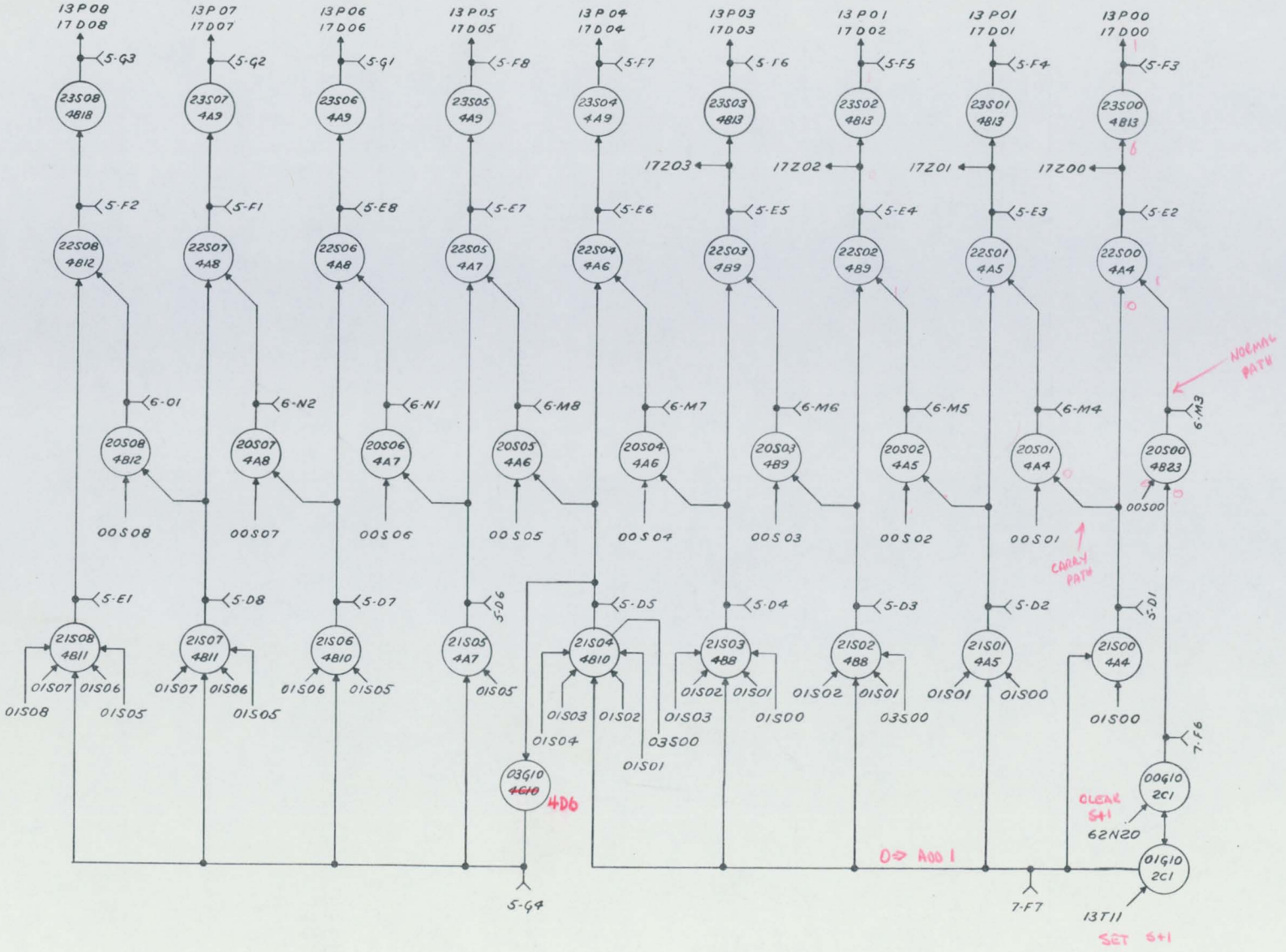


Figure 5-38. S + (0, 1)

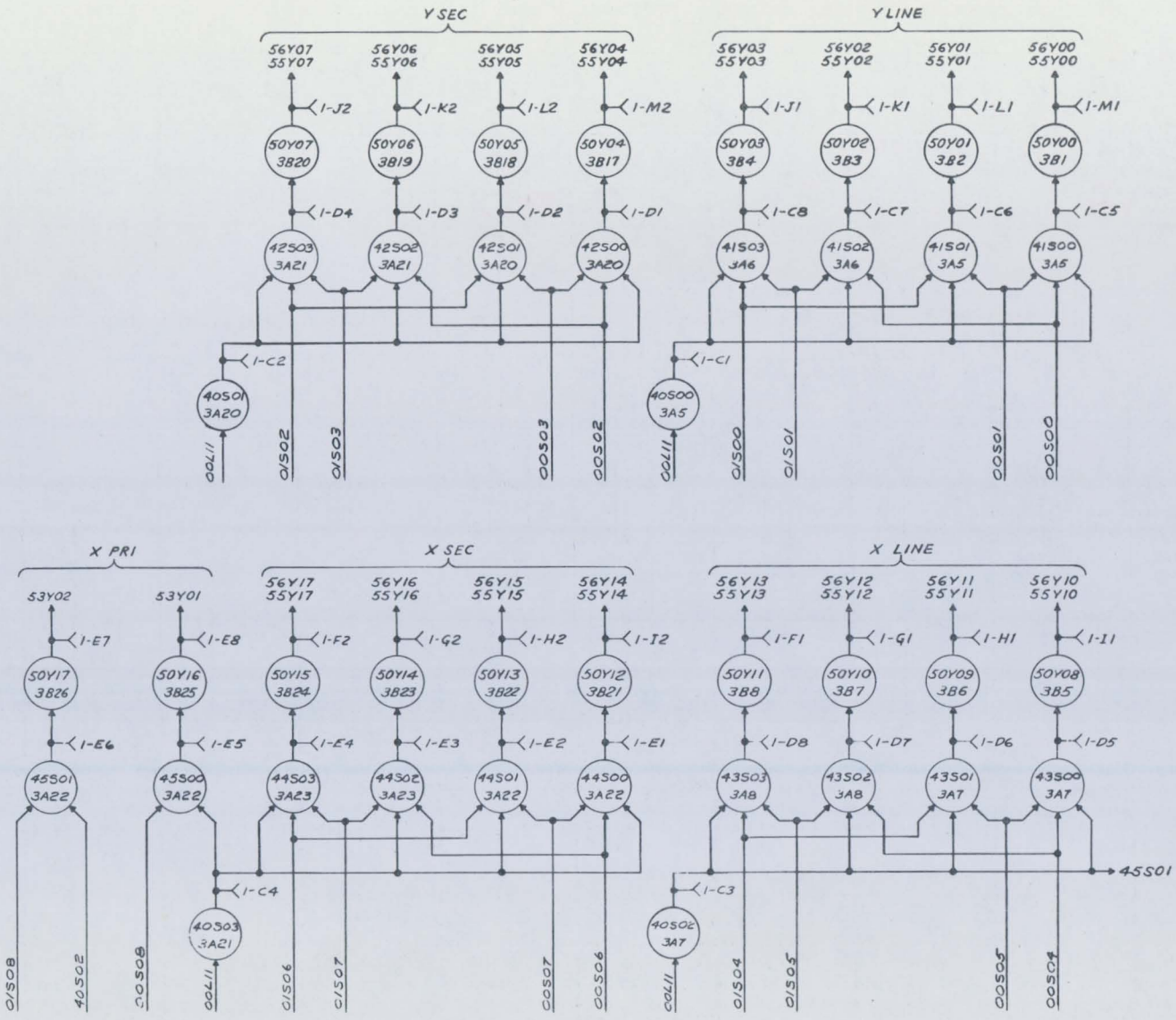


Figure 5-39. S Translator

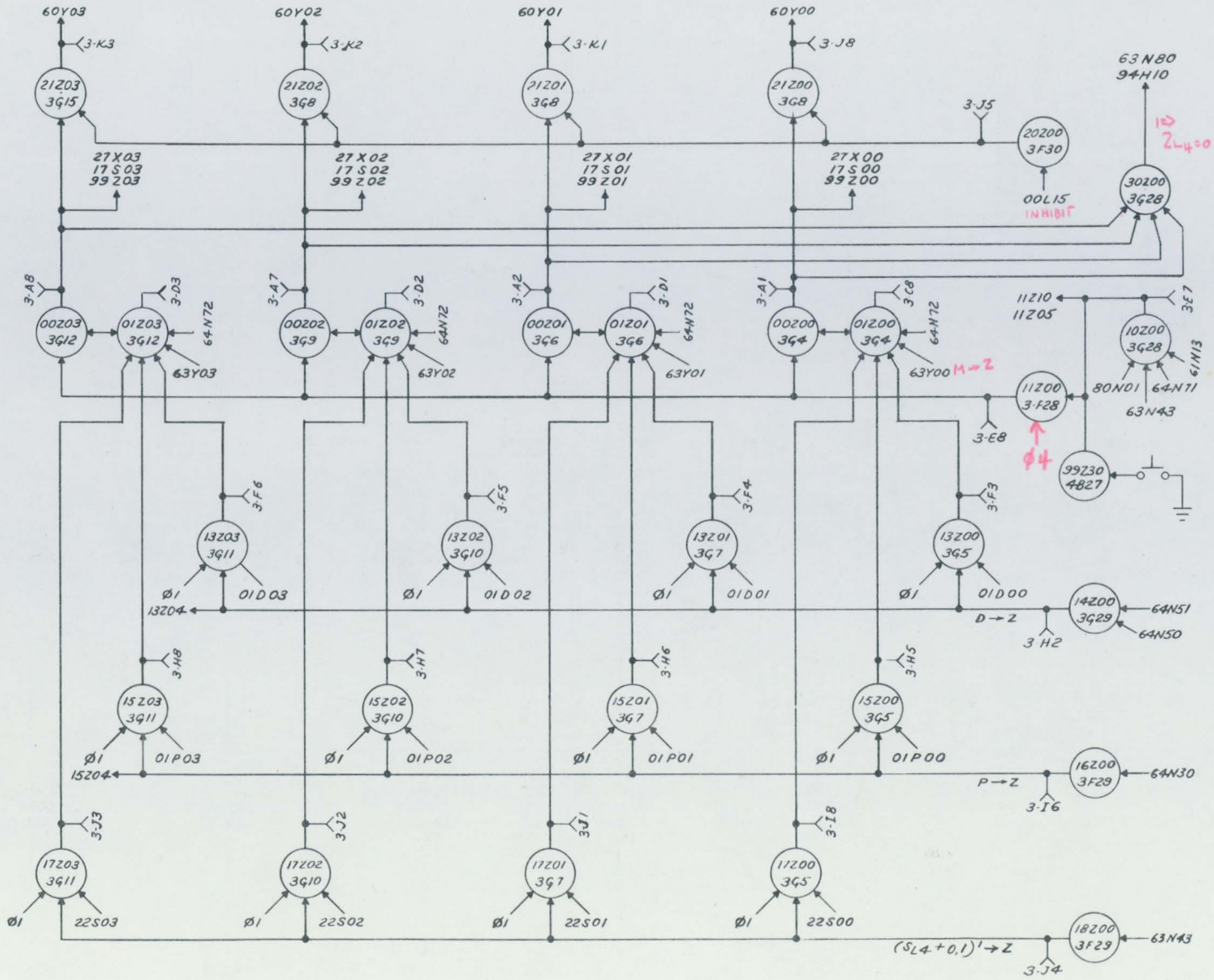


Figure 5-40. Z Register, Bits 00-03

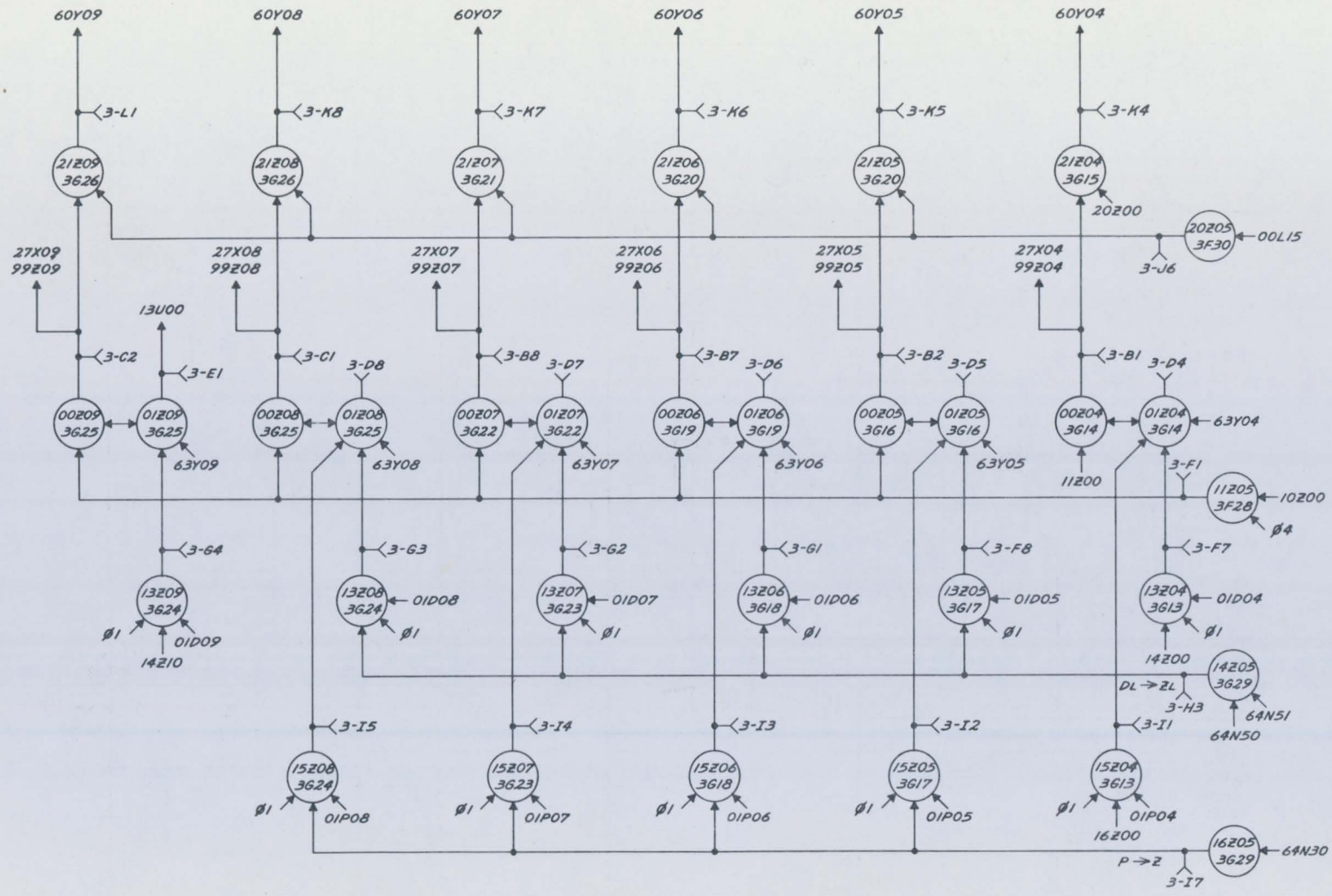


Figure 5-41. Z Register, Bits 04-09

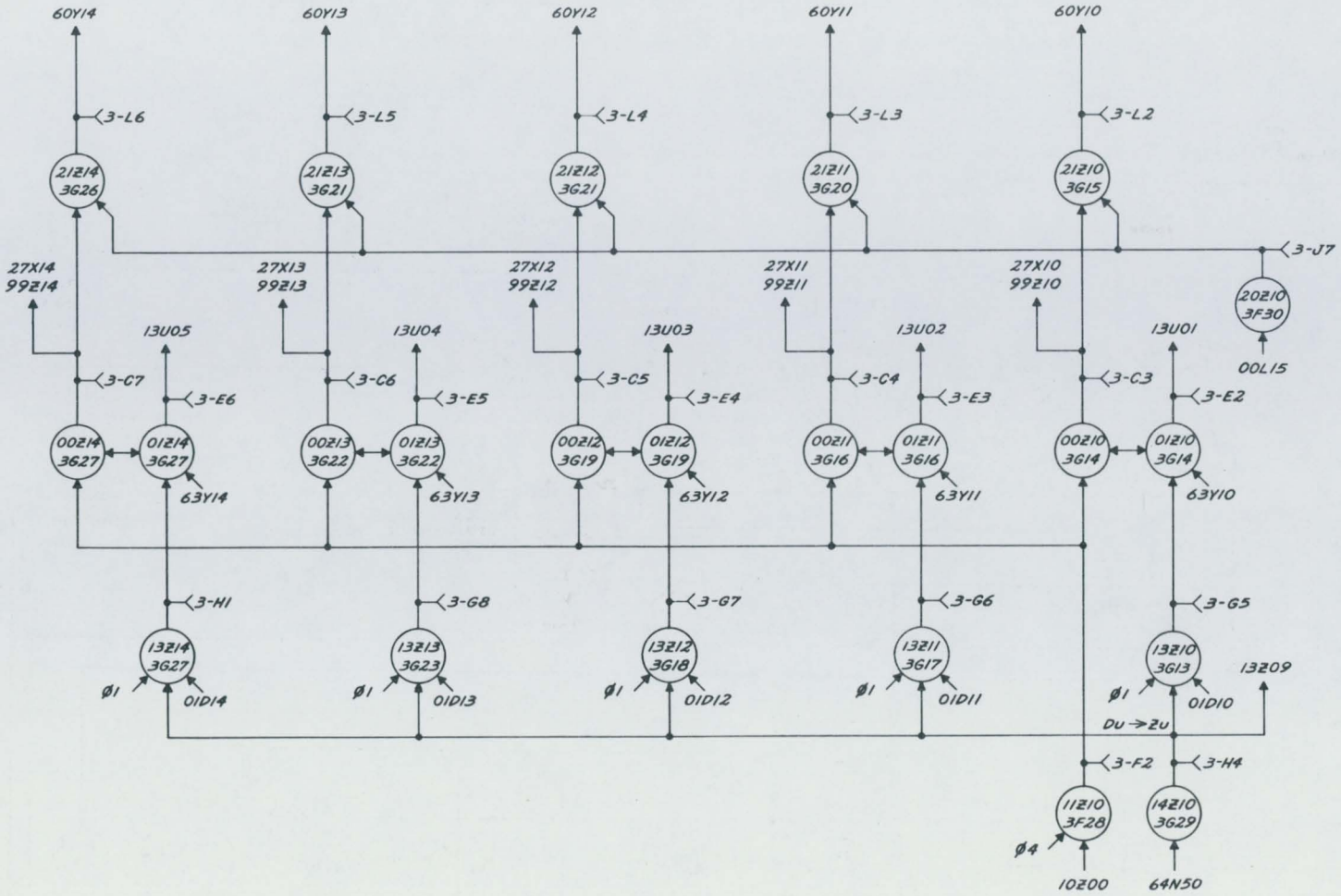


Figure 5-42. Z Register, Bits 10-14

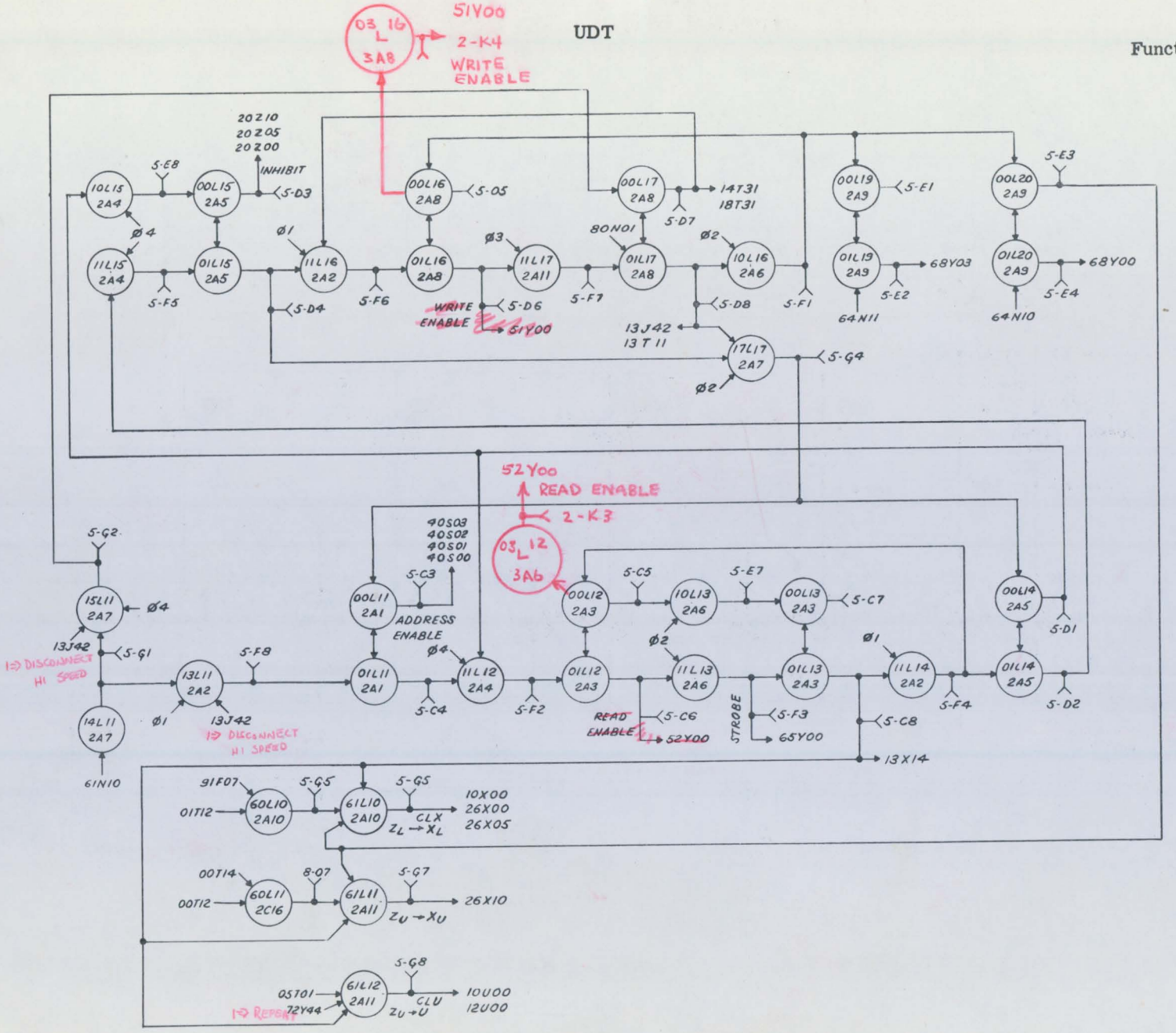


Figure 5-43. Memory Timing Chain

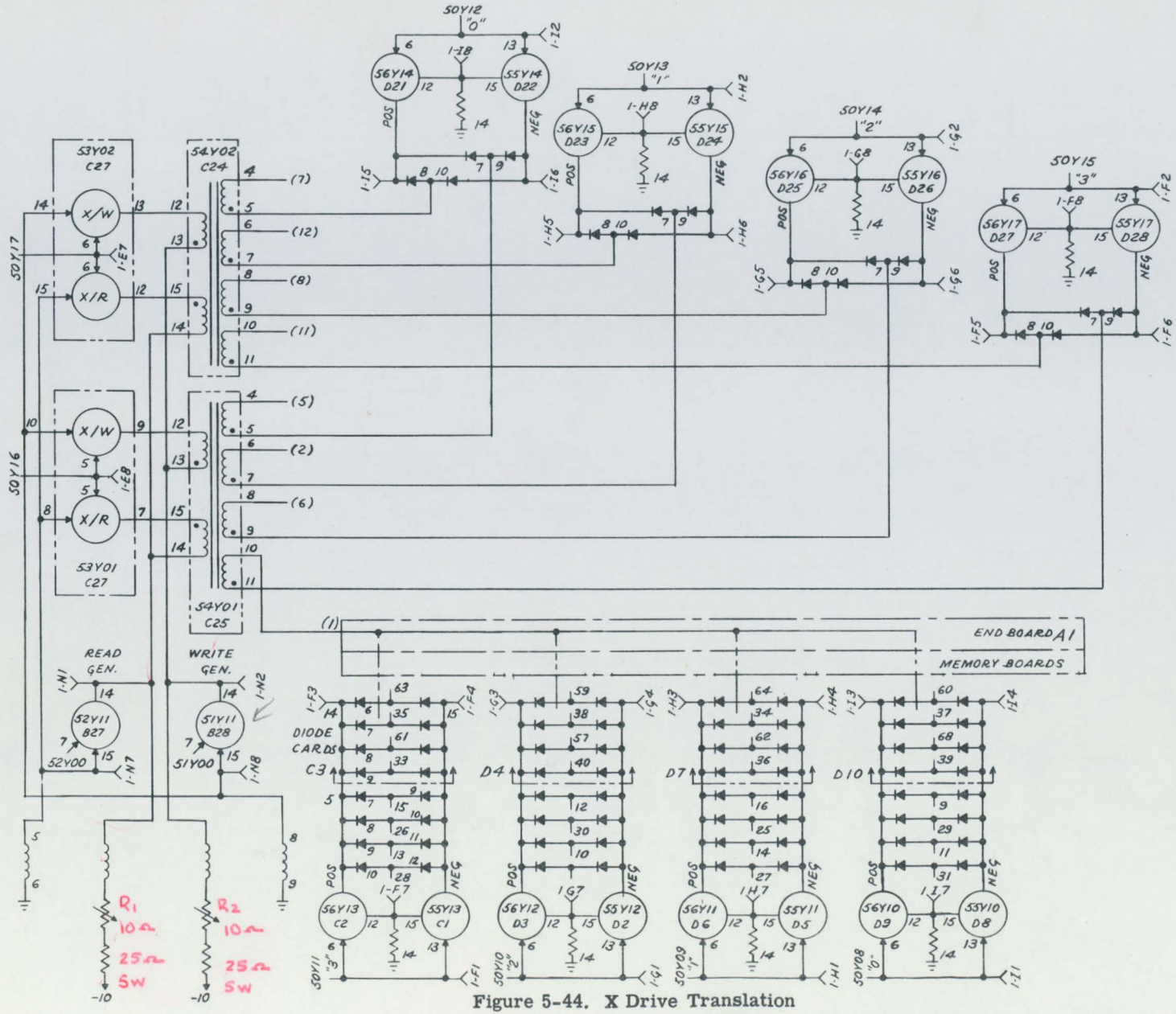


Figure 5-44. X Drive Translation

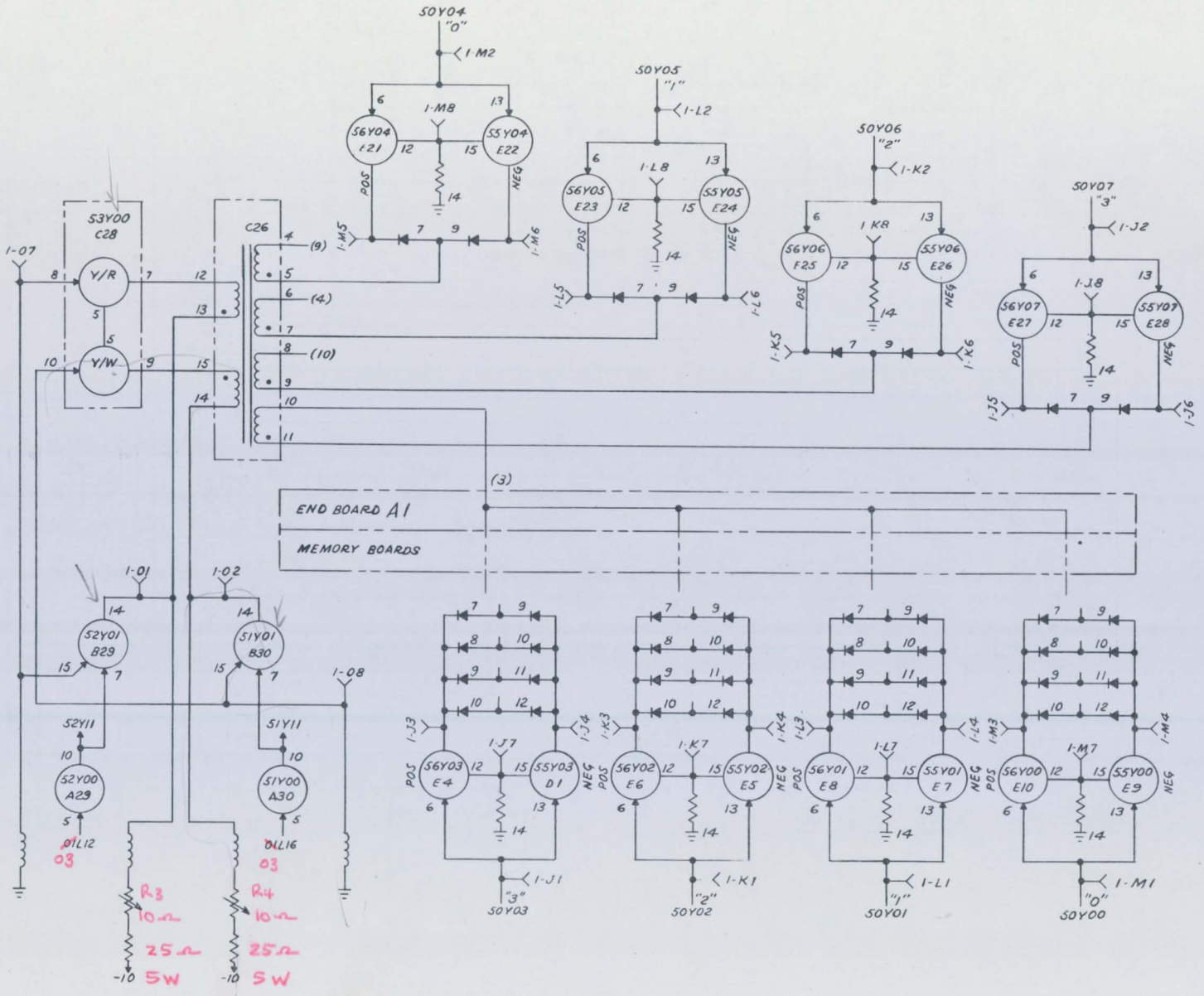


Figure 5-45. Y Drive Translation

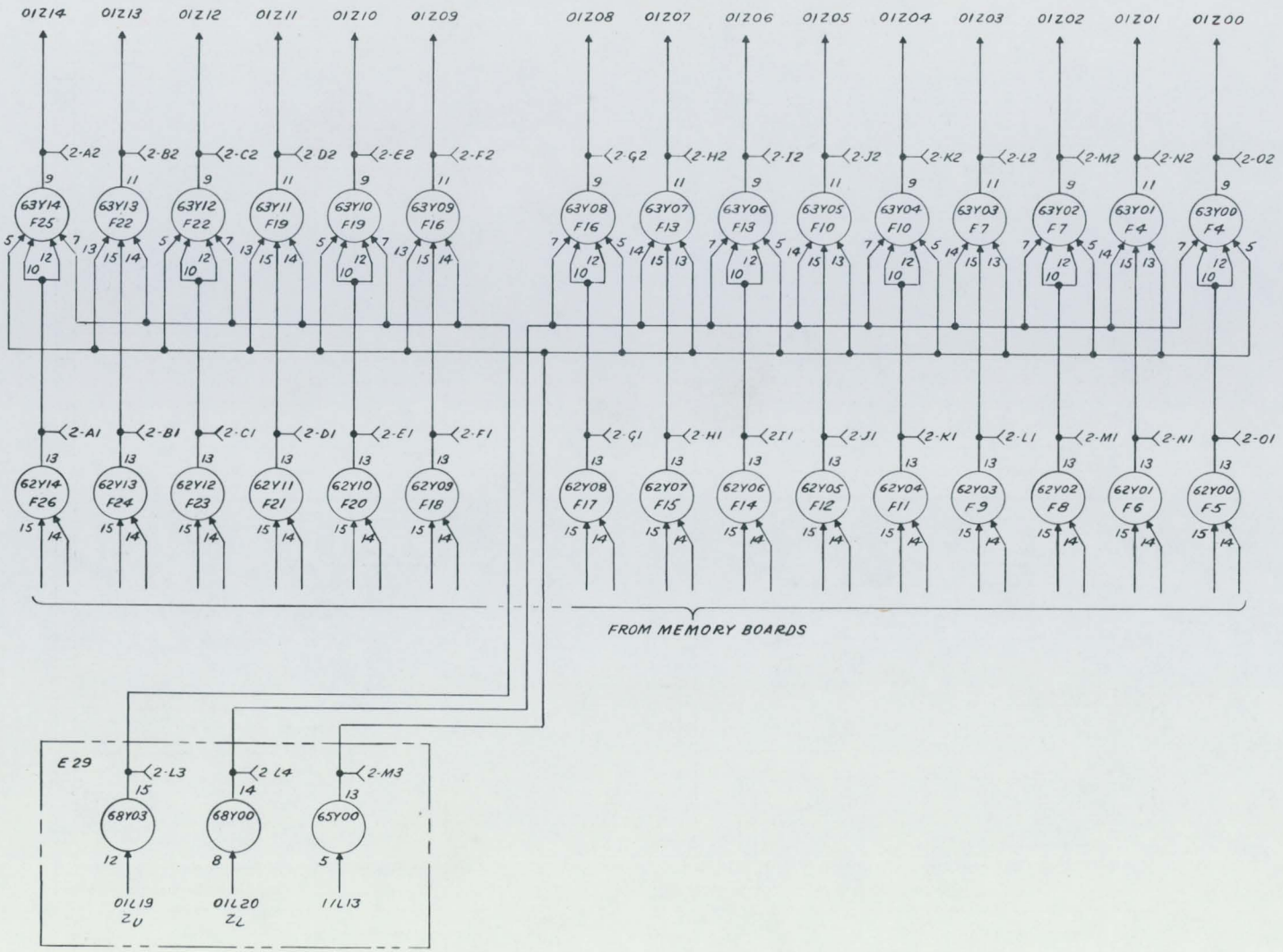


Figure 5-46. Sense Amplifier

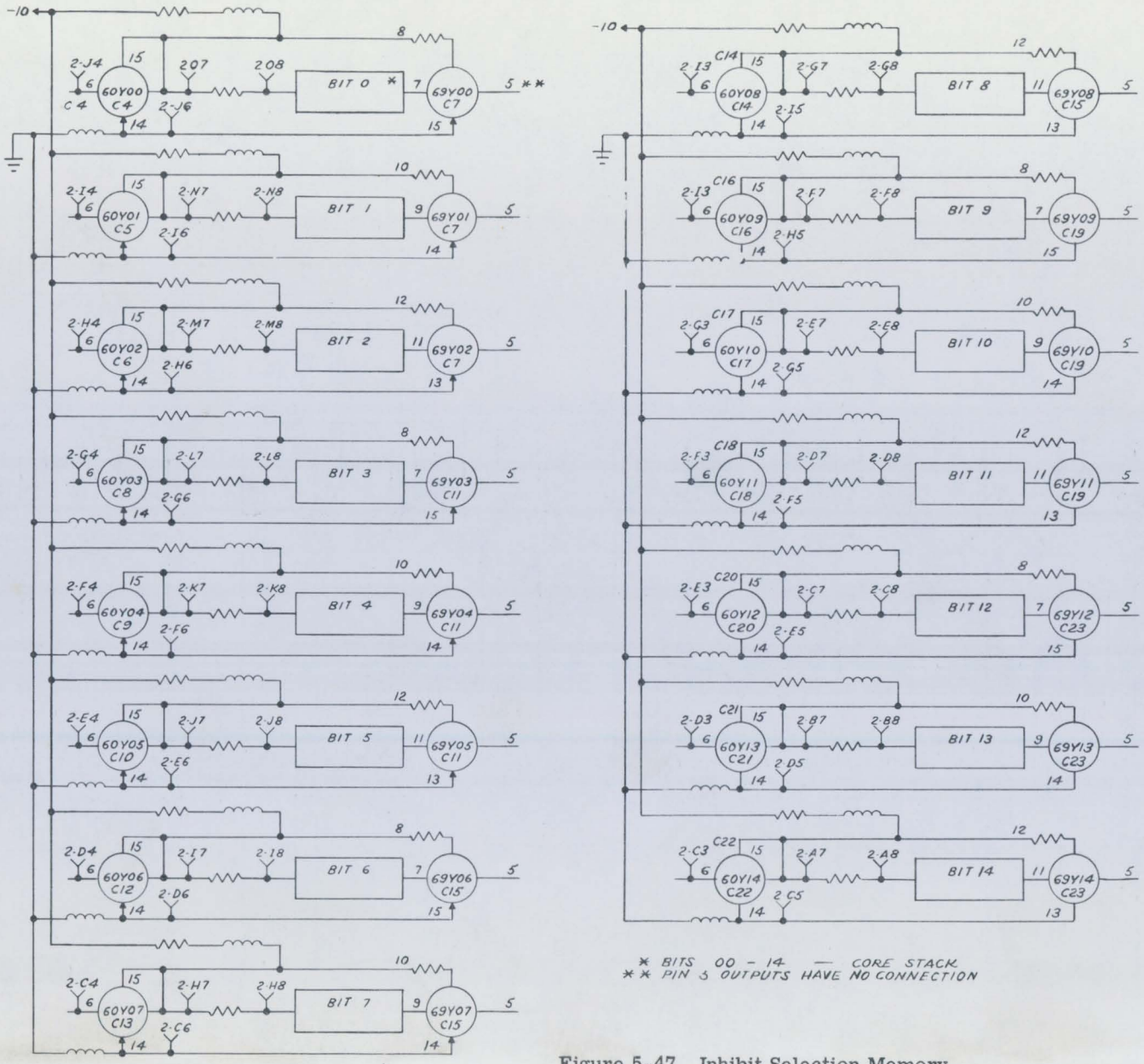


Figure 5-47. Inhibit Selection Memory

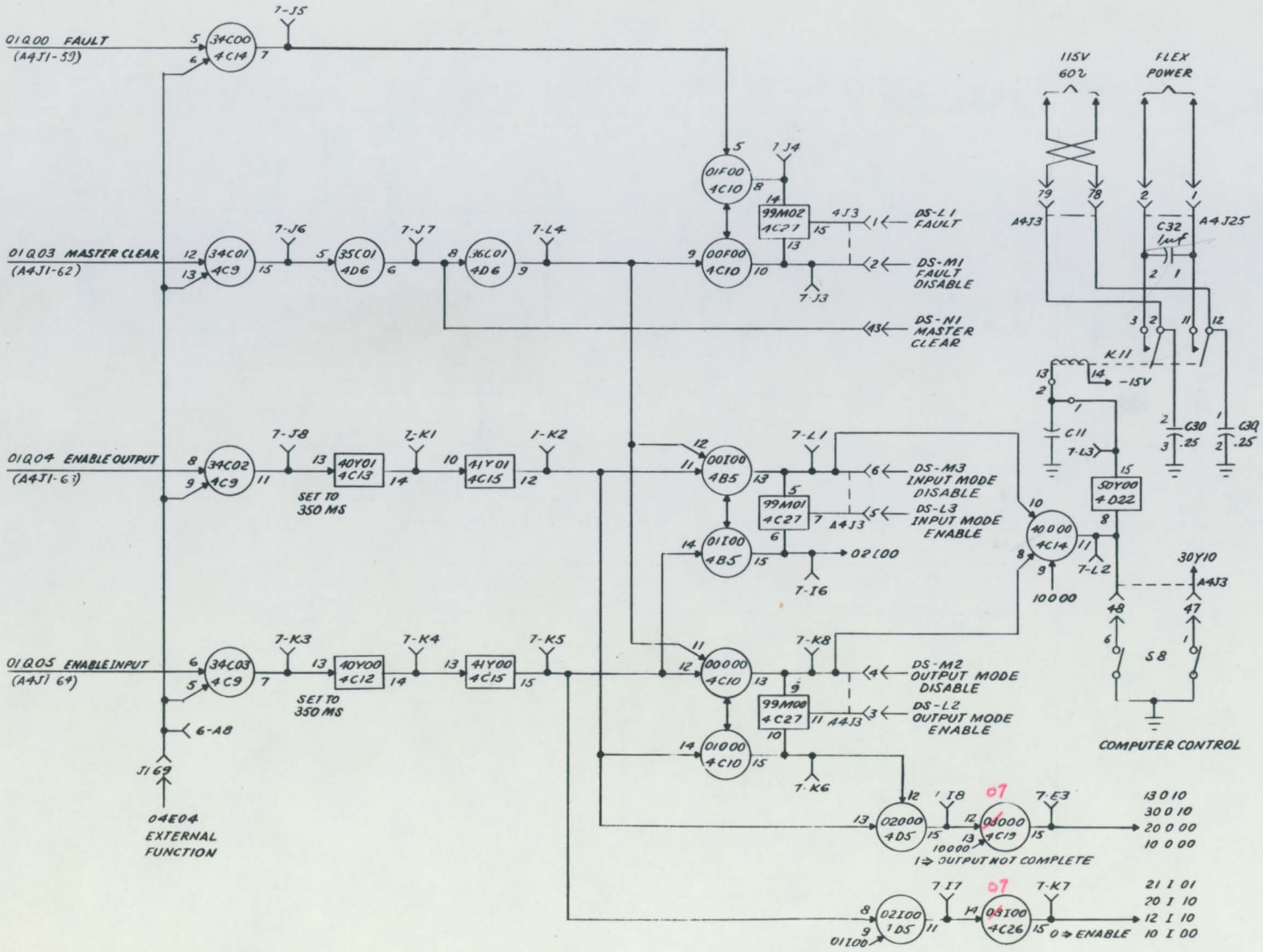


Figure 5-48. Flexowriter Control

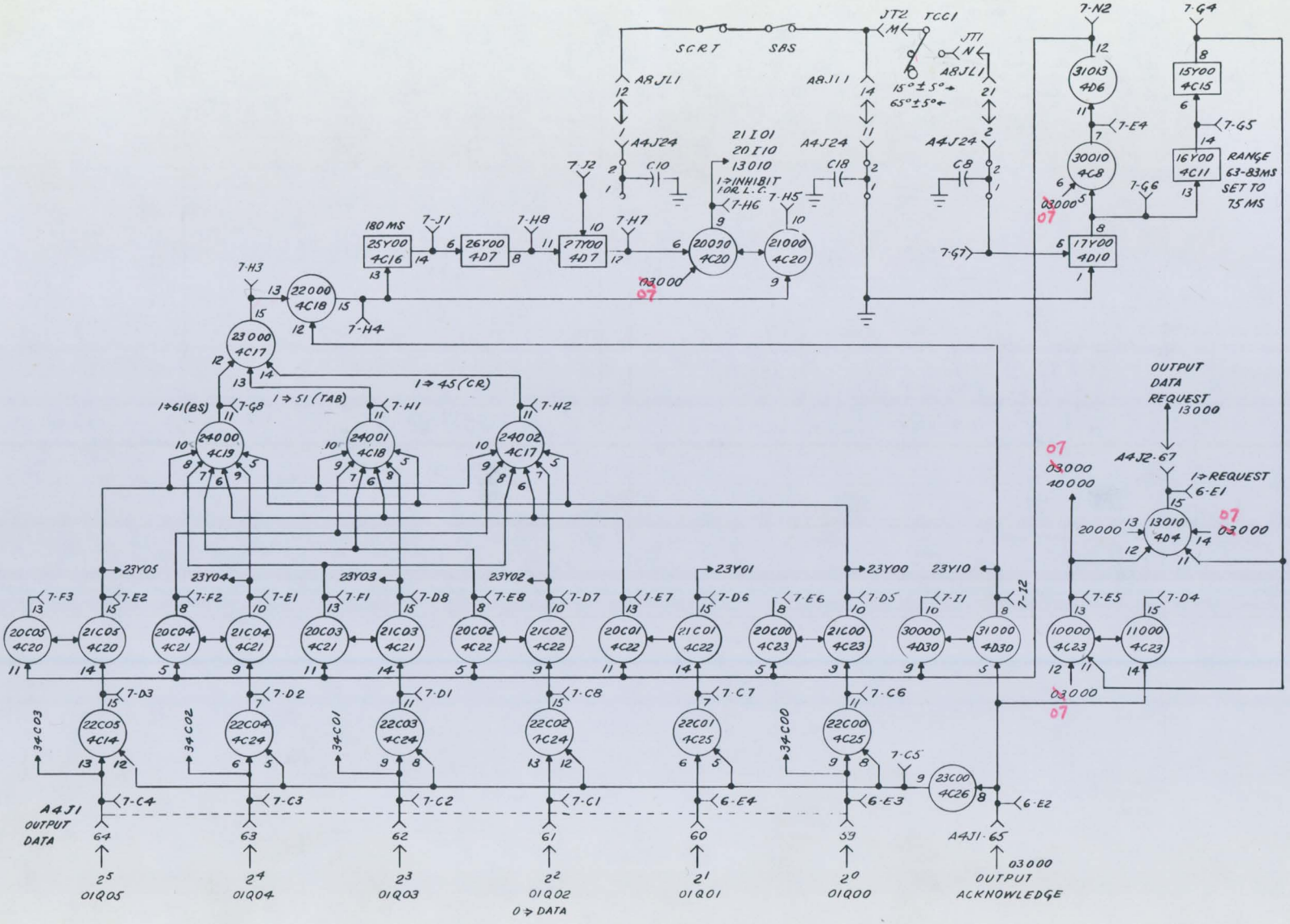


Figure 5-49. Flexowriter Output - I

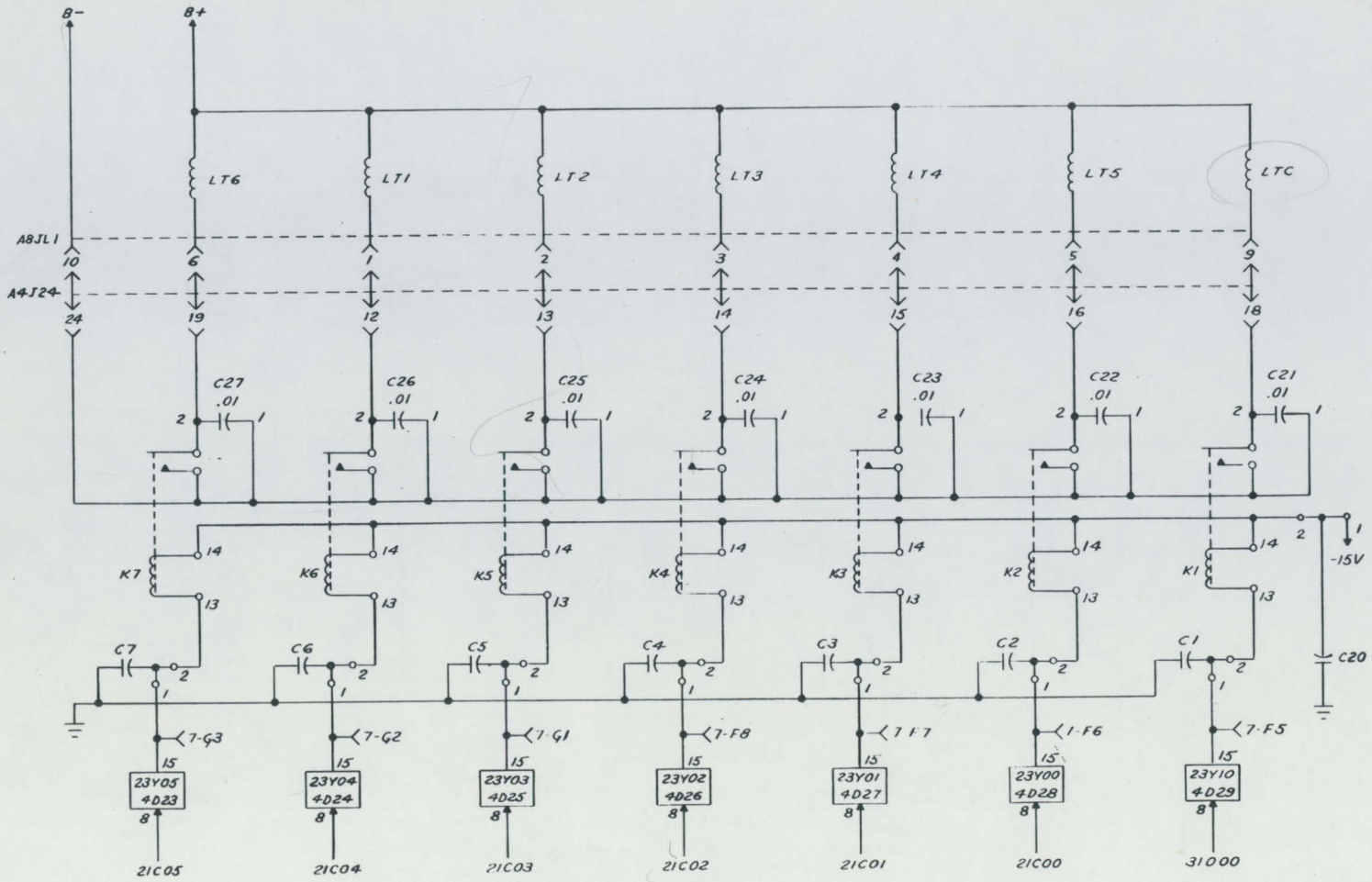


Figure 5-50. Flexowriter Output - II

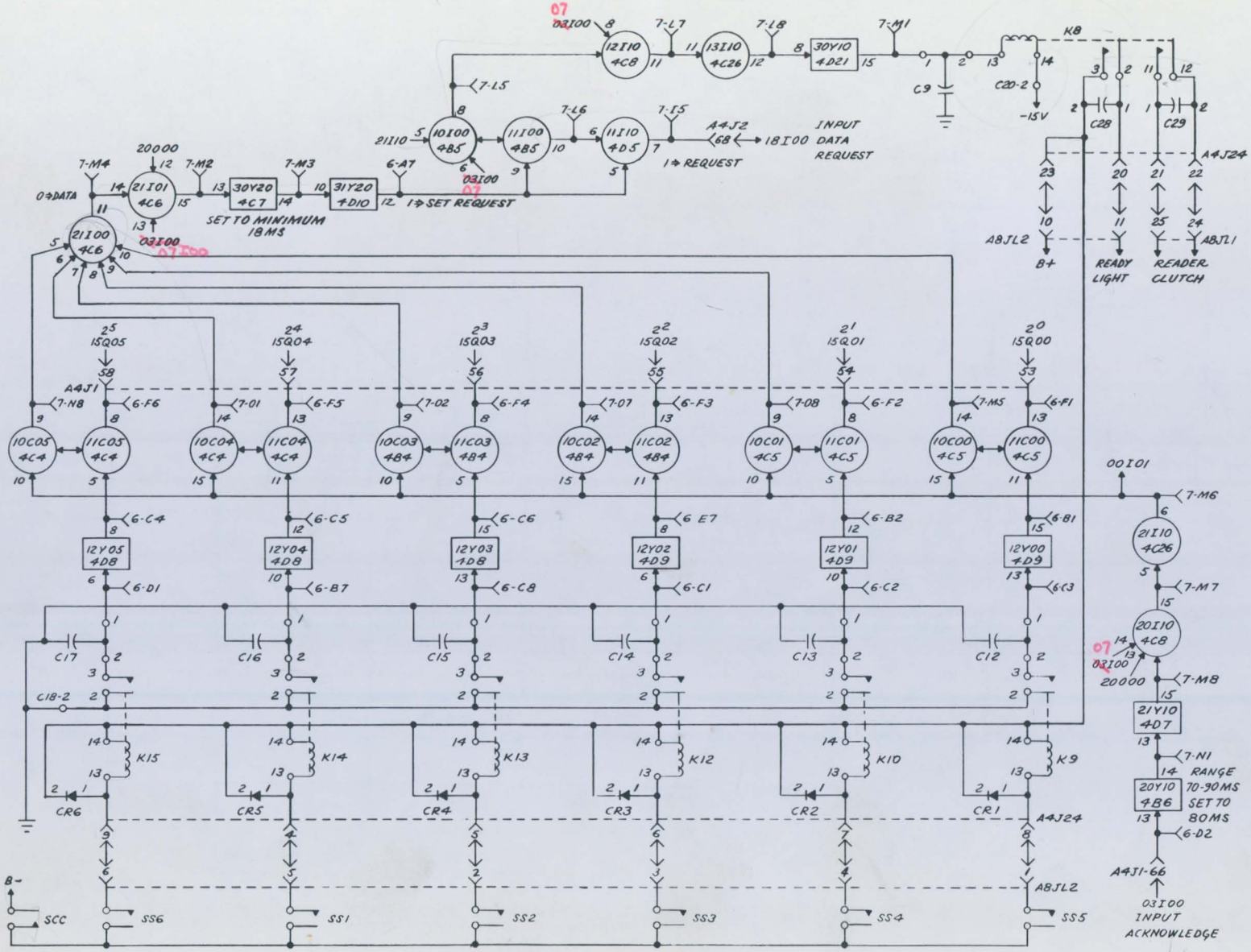


Figure 5-51. Flexowriter Input

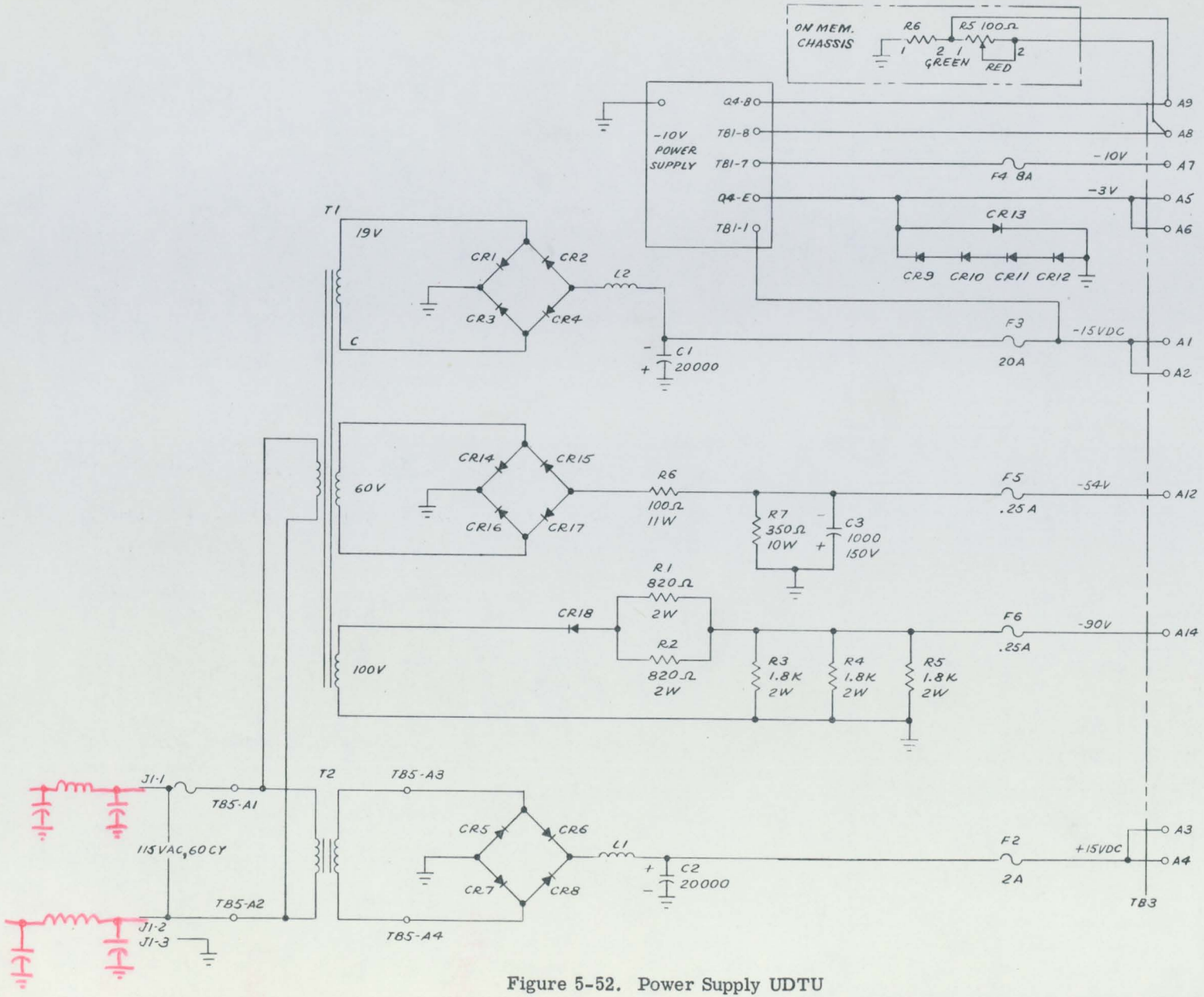
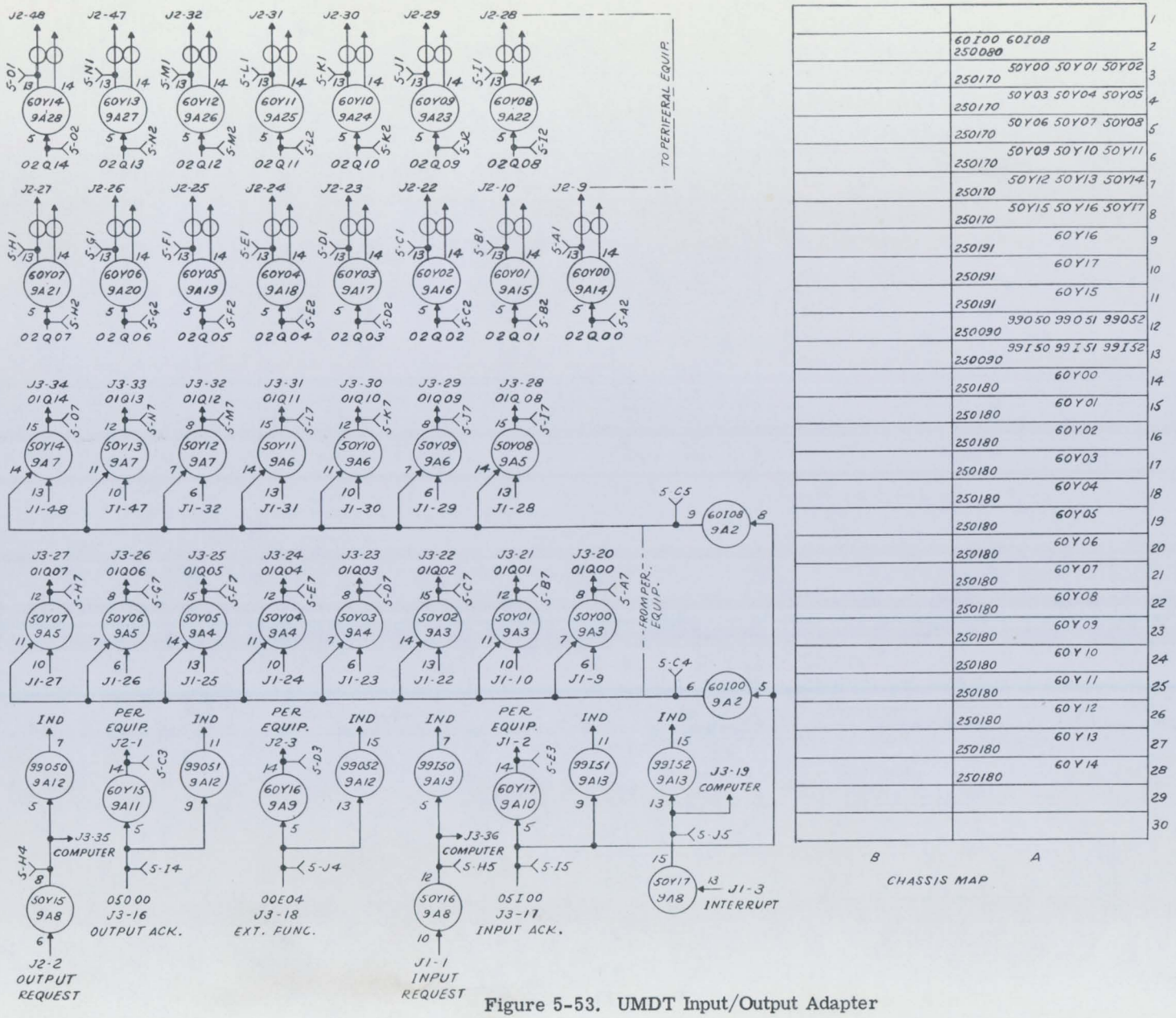


Figure 5-52. Power Supply UDTU



		1
	60100 60108 250080	2
	50Y00 50Y01 50Y02 250170	3
	50Y03 50Y04 50Y05 250170	4
	50Y06 50Y07 50Y08 250170	5
	50Y09 50Y10 50Y11 250170	6
	50Y12 50Y13 50Y14 250170	7
	50Y15 50Y16 50Y17 250170	8
	60Y16 250191	9
	60Y17 250191	10
	60Y15 250191	11
	99050 99051 99052 250090	12
	99150 99151 99152 250090	13
	60Y00 250180	14
	60Y01 250180	15
	60Y02 250180	16
	60Y03 250180	17
	60Y04 250180	18
	60Y05 250180	19
	60Y06 250180	20
	60Y07 250180	21
	60Y08 250180	22
	60Y09 250180	23
	60Y10 250180	24
	60Y11 250180	25
	60Y12 250180	26
	60Y13 250180	27
	60Y14 250180	28
		29
		30

B A

CHASSIS MAP

Figure 5-53. UMDT Input/Output Adapter

	G	F	E	D	C	B	A	
1				20A81 20AB0 250020	12A00 12A05 12A10 250070	14A00 14A05 14A10 250070	11A00 11A05 11A10 250410	1
2				13A05 15A05 15A14 250410	01A13 00A13 01A14 250420 00A14	13A11 15A11 13A09 250410	15A09 15A10 13A10 250410	2
3	09Q00 10Q00 12X14 250070	—— 08Q00 250040	50A70 250010	13A13 15A13 13A14 250410	01D08 00D08 01A02 250420 00A02	01A11 00A11 01A12 250420 00A12	01A09 00A09 01A10 250420 00A10	3
4	10D00 250010	11D00 11D05 11D10 250410	01D00 00D00 01D09 250420 00D09	13D08 15D08 17D08 250410	13A01 15A01 13A02 250410	13A12 15A12 15A08 250410	01A07 00A07 01A08 250420 00A08	4
5	14D05 14D10 250040	13D00 15D00 17D00 250410	13D10 13D09 15D09 250410	01D07 00D07 01A01 250420 00A01	01D06 00D06 01A00 250420 00A00	15A02 15A07 13A08 250410	01A05 00A05 01A06 250420 00A06	5
6	12D00 12D05 250040	12D10 14D00 250040	01D01 00D01 01D10 250420 00D10	13D07 15D07 17D07 250410	13D06 15D06 17D06 250410	13A00 15A00 13A07 250410	13A06 15A06 15A04 250410	6
7	16D00 16D05 —— 250070	13D01 15D01 17D01 250410	13D11 15D11 15D10 250410	13D03 15D03 17D03 250410	13D04 15D04 17D04 250410	13D05 15D05 17D05 250410	13A03 15A03 13A04 250410	7
8	12Q00 12Q05 12Q10 250070	11Q00 11Q05 11Q10 250410	01D02 00D02 01D11 250420 00D11	01D03 00D03 01D12 250420 00D12	13D12 15D12 13D13 250410	01D05 00D05 01D14 250420 00D14	01A03 00A03 01A04 250420 00A04	8
9	02Q04 02Q05 02Q06 250080 02Q07	02Q08 02Q09 02Q10 250080 02Q11	02Q12 02Q13 02Q14 250080 50A71	13D02 15D02 17D02 250410	01D04 00D04 01D13 250420 00D13	15D13 13D14 15D14 250410	72A14 71A14 70A14 250070	9
10	02Q00 02Q01 02Q02 250080 02Q03	01Q13 00Q13 01Q14 250420 00Q14	01Q04 00Q04 01Q09 250420 00Q09	01Q03 00Q03 01Q08 250420 00Q08	13Q08 13Q02 15Q02 250410	72A12 71A12 70A12 250070	72A13 71A13 70A13 250070	10
11	01Q11 00Q11 01Q12 250420 00Q12	13Q12 13Q13 13Q14 250410	13Q10 13Q04 15Q04 250410	13Q09 13Q03 15Q03 250410	01Q02 00Q02 01Q07 250420 00Q07	72A10 71A10 70A10 250070	72A11 71A11 70A11 250070	11
12	13Q11 13Q05 15Q05 250410	01Q05 00Q05 01Q10 250420 00Q10	13Q00 15Q00 13Q06 250410	01Q01 00Q01 01Q06 250420 00Q06	13Q01 15Q01 13Q07 250410	72A08 71A08 70A08 250070	72A09 71A09 70A09 250070	12
13	02G41 13G40 13G41 250070	28X00 29X00 15G41 250070	00Q00 01Q00 250040	14Q00 26X00 26X05 250080 26X10	20X10 22X00 22X05 250080 22X10	72A06 71A06 70A06 250070	72A07 71A07 70A07 250070	13
14	01G41 00G41 01G40 250420 00G40	25X13 13X14 15X14 250410	21X14 10A00 250040	14X00 14X05 14X10 250080 16X00	16X05 16X10 20X00 250080 20X05	72A04 71A04 70A04 250070	72A05 71A05 70A05 250070	14
15	19X13 21X13 23X13 250410	00X13 250010	00X14 250010	18X00 18X05 18X10 250070	24X00 24X05 24X10 250070	72A02 71A02 70A02 250070	72A03 71A03 70A03 250070	15
16	27X13 25X12 27X12 250410	13X13 15X13 17X13 250410	17X14 19X14 23X14 250410	12X00 12X05 12X10 250070	10X00 250010	72A00 71A00 70A00 250070	72A01 71A01 70A01 250070	16
17	13X12 15X12 17X12 250410	00X12 250010	01X12 01X13 01X14 250070	25X14 27X14 15X10 250410	11X00 11X05 11X10 250410	29X14 40A01 250040	40A00 250010	17
18	19X12 21X12 23X12 250410	01X09 01X10 01X11 250070	17X10 19X10 21X10 250410	13X10 25X09 27X09 250410	19X09 21X09 23X09 250410	02A14 62A00 41A00 250080 41A01	43A00 45A00 61A00 250070	18
19	27X05 25X11 27X11 250410	00X11 250010	00X10 250010	00X09 250010	13X09 15X09 17X09 250410	61A01 61A02 61A03 250070	61A04 61A05 61A06 250070	19
20	19X11 21X11 23X11 250410	23X10 25X10 27X10 250410	01X06 01X07 01X08 250070	15X06 25X07 27X07 250410	19X07 21X07 23X07 250410	61A07 61A08 61A09 250070	61A10 61A11 61A12 250070	20
21	13X11 15X11 17X11 250410	00X08 250010	00X06 250010	23X06 25X06 27X06 250410	00X07 250010	61A13 61A14 47A00 250070	50A81 50A80 250020	21
22	25X05 25X08 27X08 250410	13X08 15X08 17X08 250410	27X04 13X06 13X04 250410	17X06 19X06 21X06 250410	13X07 15X07 17X07 250410	30A13 31A13 32A13 250070	30A14 31A14 32A14 250070	22
23	19X08 21X08 23X08 250410	21X04 23X04 25X04 250410	00X04 250010	15X04 17X04 19X04 250410	20A71 20A70 250020	30A11 31A11 32A11 250070	30A12 31A12 32A12 250070	23
24	23X03 25X03 27X03 250410	00X03 250010	01X03 01X04 01X05 250070	00X05 250010	19X05 21X05 23X05 250410	30A09 31A09 32A09 250070	30A10 31A10 32A10 250070	24
25	17X03 19X03 21X03 250410	13X03 15X03 25X01 250410	19X01 21X01 23X01 250410	13X01 15X01 17X01 250410	13X05 15X05 17X05 250410	30A07 31A07 32A07 250070	30A08 31A08 32A08 250070	25
26	13X02 15X02 17X02 250410	00X02 250010	00X01 250010	00X00 250010	21X00 25X00 27X00 250410	30A05 31A05 32A05 250070	30A06 31A06 32A06 250070	26
27	19X02 21X02 23X02 250410	25X02 27X02 27X01 250410	01X02 01X01 01X00 250070	17X00 23X00 250040	13X00 15X00 19X00 250410	30A04 31A04 32A04 250070	30A03 31A03 32A03 250070	27
28							30A02 31A02 32A02 250070	28
29	1J6	1J5	1J4	1J3	1J2	1J1	30A01 31A01 32A01 250070	29
30							30A00 31A00 32A00 250070	30
	G	TB10 +15 / TB9 -15 F	TB8 -3 / TB7 Ø4 E	TB6 Ø3 D	TB5 Ø2 C	TB4 Ø1 / TB3 -3 B	TB2 -15 / TB1 +15 A	

Figure 5-54. Chassis No. 1

G	F	E	D	C	B	A	
				00G10 01G10 12500 250070	12P00 12P05 10P00 250070	00L11 01L11 10T53 250420 11T53	1
		19G50 21G50 31G50 250070	91F13 52F13 91H11 250080 80N05	80N00 80N02 80N04 250080 80N03	10U05 11U00 90F02 250070	13L11 11L14 11L16 250410	2
99U03 99U04 99U05 250090	72Y44 72Y48 72Y43 250170	25G50 27G50 17G50 250070	20U20 20U21 20U22 250080 20U23	10U21 10U22 10U23 250070	00U04 01U04 00U05 250420 01U05	00L12 01L12 00L13 250420 01L13	3
99U00 99U01 99U02 250090	72Y41 72Y42 72Y45 250170	15G50 15I00 250040	20U10 20U11 20U12 250080 20U13	10U12 10U13 10U20 250070	13U03 13U04 13U05 250410	11L12 10L15 11L15 250410	4
99Y30 99U30 250090	72Y40 72Y71 72Y73 250170	00E04 01E04 00G51 250420 01G51	20U00 20U01 20U02 250080 20U03	10U03 10U10 10U11 250070	00U02 01U02 00U03 250420	00L14 01L14 00L15 250420 01L15	5
99X12 99X13 99X14 250090	72Y74 72Y70 72Y46 250170	03E04 05E04 23G50 250070	90F17 50F17 51F13 250070	10U00 10U01 10U02 250070	13U00 13U01 13U02 250410	11L13 10L13 10L16 250410	6
99X09 99X10 99X11 250090	73Y70 250200	04E04 29G50 19I00 250080 92F06	90F13 90H13 92H13 250070	91H13 90H11 92H11 250070	00U00 01U00 00U01 250420 01U01	14L11 15L11 17L17 250070	7
99X06 99X07 99X08 250090	77Y46 250200	13G50 17I00 250040	90F10 90F11 90F12 250070	52F06 90H10 81F04 250070	16T43 12U00 02U01 250080 03U01	01L16 00L16 00L17 250420 00L17	8
99X03 99X04 99X05 250090	76Y46 250200	01G50 00G50 250050	91F10 91F11 91F12 250080 63N44	51F10 55F10 94H10 250070	61N80 17000 250040	00L19 01L19 00L20 250420 01L20	9
99X00 99X01 99X02 250090	12J46 12J70 12J74 250080 12J73	00Q00 01Q00 00I00 250420 01I00	90F07 90F06 50F06 250070	52F10 53F10 92H10 250070	62N80 63N80 250040	60L10 17L12 61L10 250070	10
99Q13 99Q14 99Q30 250090	14J40 12J45 12J42 250080 14J42	15Q00 03Q00 05Q00 250070	91F06 91F07 93F06 250080 91H07	51F04 63N55 15G40 250070	61N81 60N81 250040	11L17 61L11 61L12 250070	11
99Q10 99Q11 99Q12 250090	21J42 24J42 13J42 250070	13I00 03I00 05I00 250070	18I00 13000 63N58 250070	92H06 50F04 90H04 250070	63N59 63N14 63N56 250070	63N51 63N53 90H07 250070	12
99Q07 99Q08 99Q09 250090	00J42 01J42 00J43 250420 01J43	64N15 13J47 13J46 250070	90F15 90F16 90H00 250070	91F15 91F16 91F00 250080 52F00	63N42 63N41 63N40 250070	62N50 64N50 250050	13
99Q04 99Q05 99Q06 250090	15J40 15J41 250040	17J40 17J41 17J73 250070	90F03 90F04 90F05 250070	90F00 90F01 51F00 250070	63N43 63N34 63N57 250070	63N31 63N33 250070	14
99Q01 99Q02 99Q03 250090	00J41 01J41 00J40 250420 01J40		91H00 91F04 91F05 250080 91H04	60N44 63N50 250040	61N30 64N40 61N40 250070	12T43 60N80 64N80 250070	15
99Q31 99Q32 99Q00 250090	00J72 01J72 01G73 250420 00G73	92Y01 250160	92H07 93H07 95H07 250070	12N44 60L11 62N35 250070	63N36 63N35 250040	00J46 01J46 10T43 250420 11T43	16
99D11 99D12 99D13 250090	00670 01G70 00J47 250420 01J47	92Y02 250160	CLOCK LOAD 250700	63N37 90H17 91H17 250070	64N20 64N31 250040	13T43 60N82 61N82 250070	17
99D08 99D09 99D10 250090	02J05 15J73 15J72 250070	92Y03 250160	92Y04 250160	63N32 63N38 250040	63N82 21T12 62N82 250070	61N83 64N71 64N72 250070	18
99D05 99D06 99D07 250090	12J43 98J34 98J33 250070	90Y21 90Y22 250150	90Y23 90Y24 250150	03T20 3T13 05T01 250080 03T01	16T21 16T23 16T31 250080 16T33	13T41 62N81 63N81 250070	19
99D02 99D03 99D04 250090	14J43 98J32 98J31 250070	90Y11 250140	90Y13 250140	15T33 63N52 02T01 250070	63N22 63N21 63N20 250070	10T33 11T33 10T41 250420 11T41	20
99A30 99A00 99D01 250090	99J32 99J33 99J34 250090	90Y12 250140	90Y14 250140	03T14 05T14 07T14 250080 09T14	63N30 63N10 63N11 250070	13T33 64N60 64N61 250070	21
99A12 99A13 99A14 250090	99T41 99T43 99J31 250090	90Y02 250690	90Y04 250690	00T13 01T13 00T14 250420 01T14	12T31 14T31 18T31 250070	62N60 13T31 250040	22
99A09 99A10 99A11 250090	99T23 99T31 99T33 250090	90Y03 250250	90Y05 250250	15T12 15T13 15T14 250410	63N24 60N13 61N13 250070	10T23 11T23 10T31 250420 11T31	23
99A06 99A07 99A08 250090	99T11 99T13 99T21 250090	90Y00 250100	90Y06 250690	00T02 01T02 00T12 250420 01T12	63N23 62N21 64N11 250070	12T23 13T23 64N70 250070	24
99A03 99A04 99A05 250090	98T13 98T14 99T00 250090	90Y01 250240	00J30 01J30 00J31 250420 01J31	15T31 00T04 00T03 250070	62N20 62N22 250040	12T21 13T21 64N30 250070	25
99A00 99A01 99A02 250090	98T00 98T01 98T12 250090		21A70 10J33 10J31 250070	01T04 01T03 250040	61N12 64N10 250050	11T13 10T13 11T21 250420 10T21	26
		10J30 10J34 10J32 250070			61N20 61N10 61N11 250070	13T13 15T21 15T23 250070	27
2J8	2J7	2J6	2J5		60N12 60N11 60N10 250070	19T00 13T11 250040	28
					13T00 64N51 64N52 250070	11T00 10T00 11T11 250420 10T11	29
							30
TB10 +15	TB9 -15	TB8 -3	TB7 Ø4	TB6 Ø3	TB5 Ø2	TB4 Ø1	TB3 -3	TB2 -15	TB1 +15	

Figure 5-55. Chassis No. 2

03L12

03L16

	G	F	E	D	C	B	A	
				55Y03 265565	55Y13 265565	50Y00 250430	R1 ○ XR R2 ○ XW R3 ○ YR R4 ○ YW R5 ○ -10	1
	J3	J2	J1	55Y12 265565	56Y13 265562	50Y01 250430		2
				56Y12 265562	DIODE CARD 265568	50Y02 250430		3
	01Z00 00Z00 250050	63Y00 63Y01 250340	56Y03 265562	DIODE CARD 265568	60Y00 265682	50Y03 250430		4
	13Z00 15Z00 17Z00 250410	62Y00 250330	55Y02 265565	55Y11 265565	60Y01 265682	50Y08 250430	40S00 41S00 41S01 250070	5
	01Z01 00Z01 250050	62Y01 250330	56Y02 265562	56Y11 265562	60Y02 265682	50Y09 250430	41S02 41S03 250070	6
	13Z01 15Z01 17Z01 250410	63Y02 63Y03 250340	55Y01 265565	DIODE CARD 265568	69Y00 69Y01 69Y02 250270	50Y10 250430	40S02 43S00 43S01 250070	7
	21Z00 21Z01 21Z02 250070	62Y02 250330	56Y01 265562	55Y10 265565	60Y03 265682	50Y11 250430	43S02 43S03 250070	8
	01Z02 00Z02 250050	62Y03 250330	55Y00 265565	56Y10 265562	60Y04 265682	CHOKE 265680		9
	13Z02 15Z02 17Z02 250410	63Y04 63Y05 250340	56Y00 265562	DIODE CARD 265568	60Y05 265682	CHOKE 265680	CHOKE 265680	10
	13Z03 15Z03 17Z03 250410	62Y04 250330	CORE STACK		69Y03 69Y04 69Y05 250270		CHOKE 265680	11
	01Z03 00Z03 250050	62Y05 250330			60Y06 265682	CHOKE 265680		12
	13Z04 15Z04 13Z10 250410	63Y06 63Y07 250340			60Y07 265682	CHOKE 265680		13
	01Z04 00Z04 01Z10 250420 00Z10	62Y06 250330			60Y08 265682			14
	21Z03 21Z04 21Z10 250070	62Y07 250330			69Y06 69Y07 69Y08 250270	CHOKE 265680		15
	01Z05 00Z05 01Z11 250420 00Z11	63Y08 63Y09 250340			60Y09 265682	CHOKE 265680		16
	13Z05 15Z05 13Z11 250410	62Y08 250330			60Y10 265682	50Y04 250430		17
	13Z06 15Z06 13Z12 250410	62Y09 250330			60Y11 265682	50Y05 250430		18
	01Z06 00Z06 01Z12 250420 00Z12	63Y10 63Y11 250340			69Y09 69Y10 69Y11 250270	50Y06 250430		19
	21Z05 21Z06 21Z11 250070	62Y10 250330			60Y12 265682	50Y07 250430	42S01 40S00 42S01 250070	20
	21Z07 21Z12 21Z13 250070	62Y11 250330	56Y04 265562	56Y14 265562	60Y13 265682	50Y12 250430	40S03 42S02 42S03 250070	21
	01Z07 00Z07 01Z13 250420 00Z13	63Y12 63Y13 250340	55Y04 265565	55Y14 265565	60Y14 265682	50Y13 250430	45S00 44S00 44S01 250070	22
	13Z07 15Z07 13Z13 250410	62Y12 250330	56Y05 265562	56Y15 265562	69Y12 69Y13 69Y14 250270	50Y14 250430	45S01 44S02 44S03 250070	23
	13Z08 15Z08 13Z09 250410	62Y13 250330	55Y05 265565	55Y15 265565	54Y02 250370	50Y15 250430		24
	01Z08 00Z08 01Z09 250420 00Z09	63Y14 250340	56Y06 265562	56Y16 265562	54Y01 250370	50Y16 250430		25
	21Z08 21Z09 21Z14 250070	62Y14 250330	55Y06 265565	55Y16 265565	54Y00 250370	50Y17 250430		26
	00Z14 01Z14 13Z14 250070		56Y07 265562	56Y17 265562	53Y01 53Y02 250300	52Y11 250290		27
	10Z00 30Z00 250040	11Z00 11Z05 11Z10 250410	55Y07 265565	55Y17 265565	53Y00 250300	51Y11 250290		28
	14Z00 14Z05 14Z10 250070	16Z00 16Z05 18Z00 250080	65Y00 68Y00 68Y03 265677		CHOKE 265680	52Y01 250290	52Y00 250280	29
		20Z00 20Z05 250080			CHOKE 265680	51Y01 250290	51Y00 250280	30
	TB9 -3	TB8 +5	TB7 Ø1	TB6 +5	TB5	TB4 -10	TB3 +5	TB2 -3	TB1 -15	

Figure 5-56. Chassis No. 3

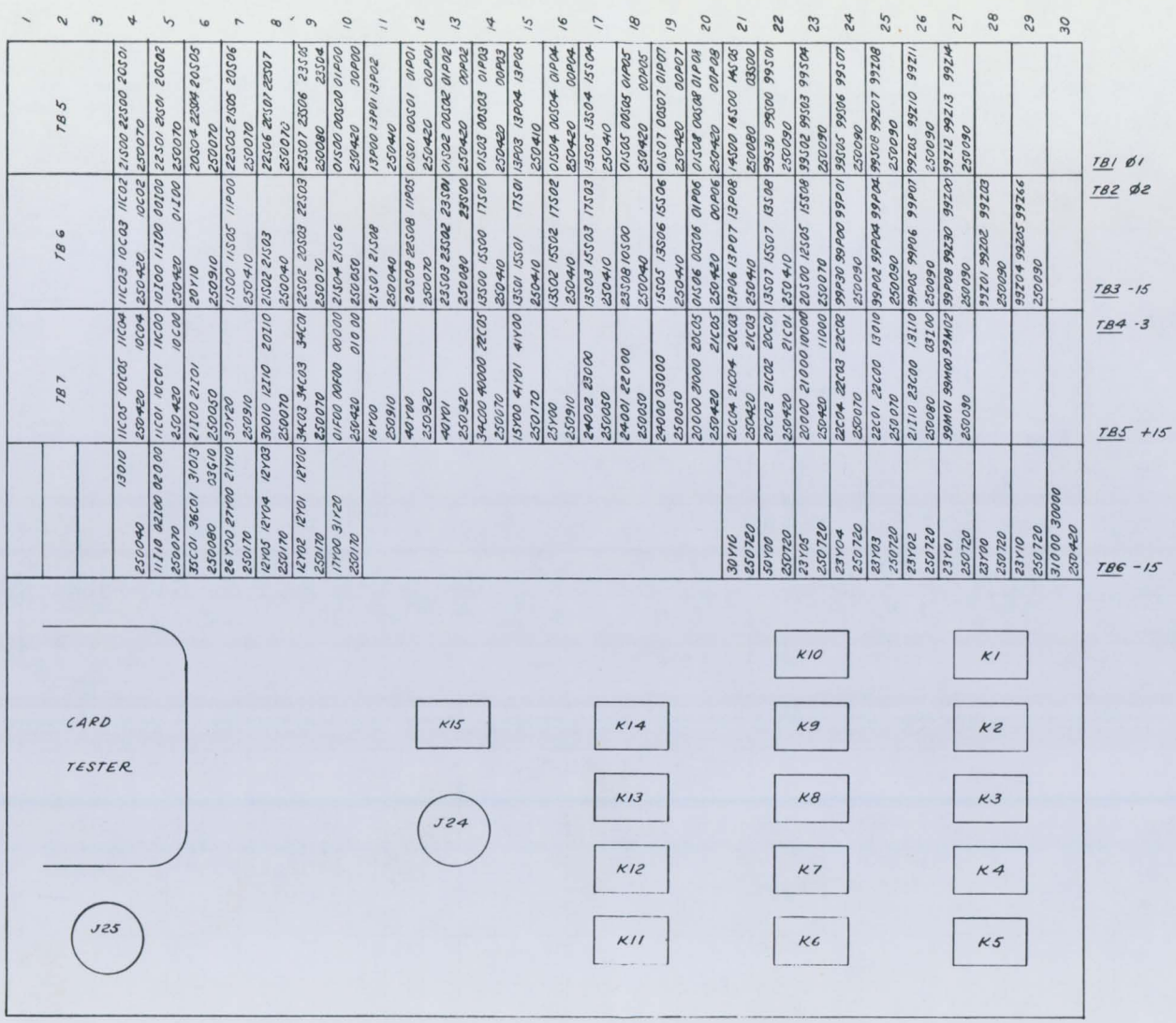


Figure 5-57. Chassis No. 4

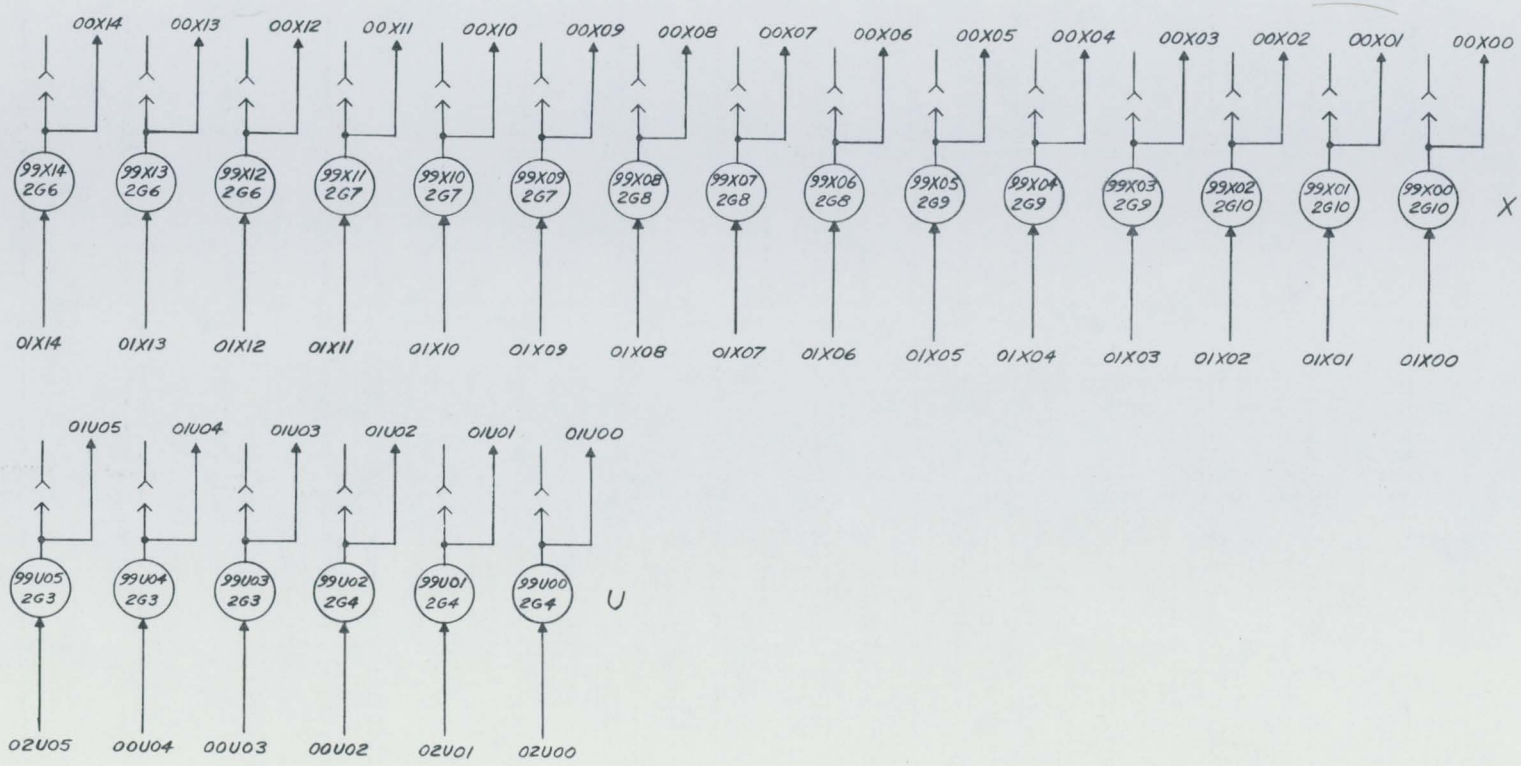


Figure 5-58. Indicator Drivers (Sheet 1)

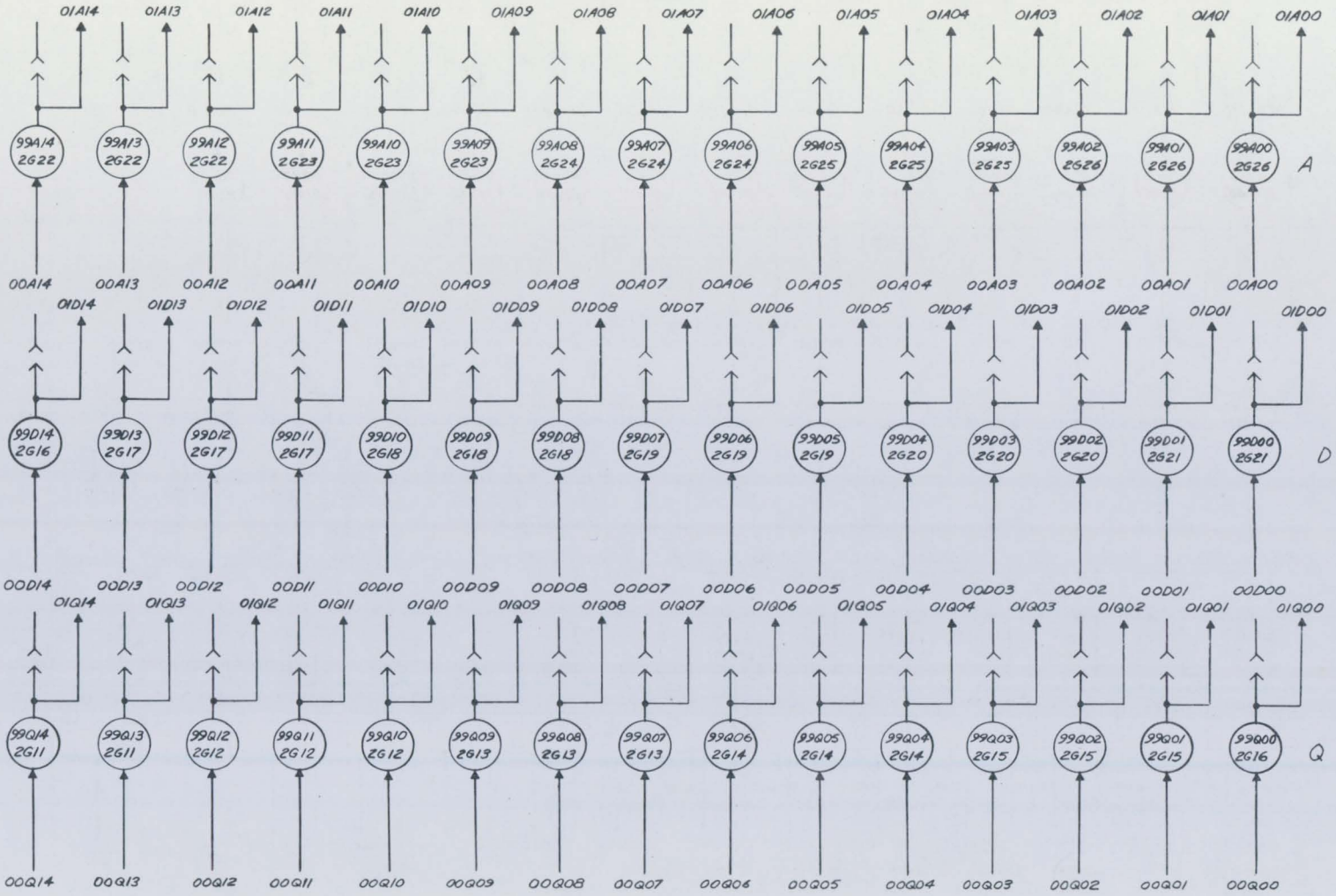


Figure 5-59. Indicator Drivers (Sheet 2)

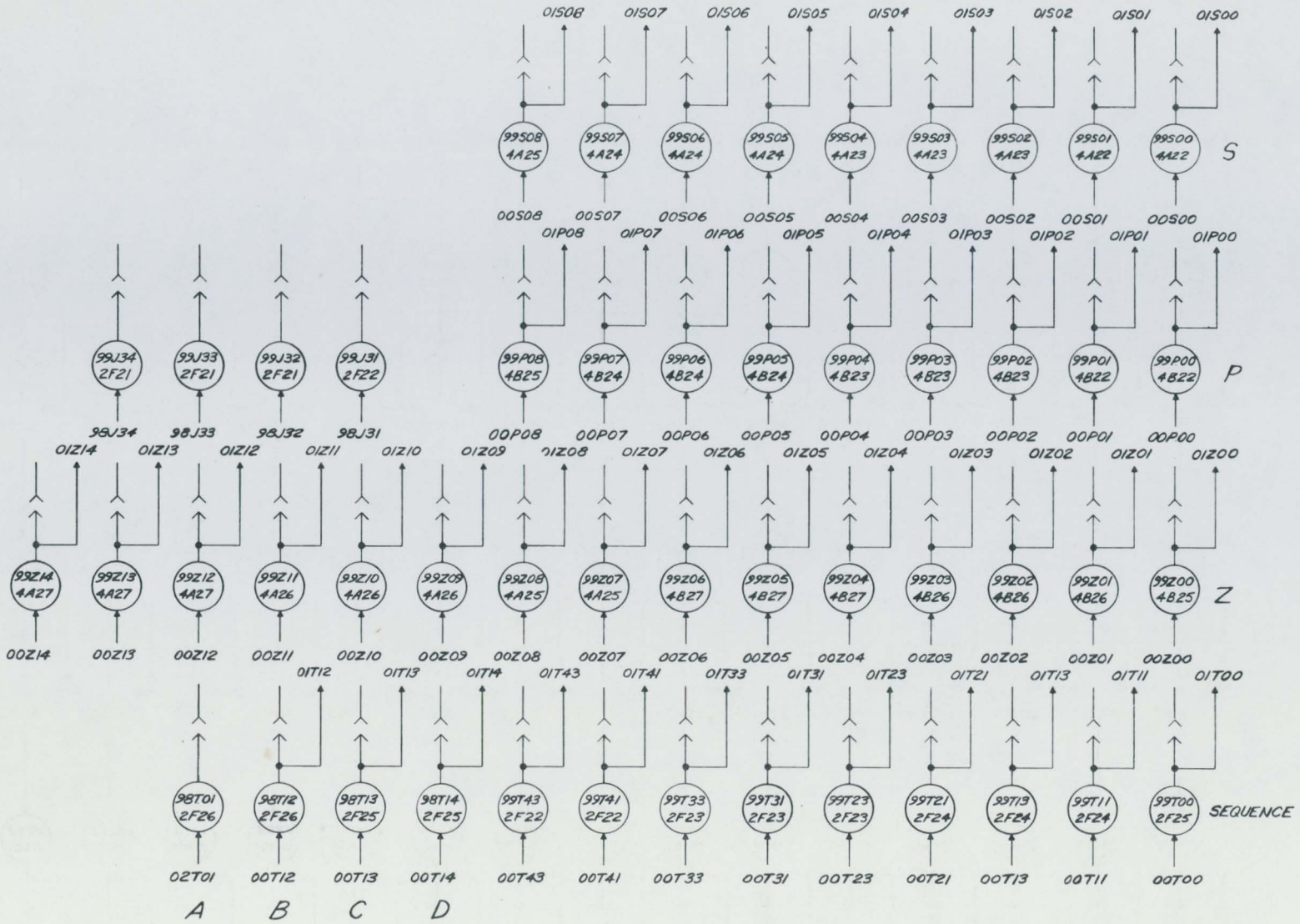


Figure 5-60. Indicator Drivers (Sheet 3)

					T^{11}	T^{10}	T^{21}	T^{23}	T^{31}	T^{33}	T^{41}	T^{43}	T^{00}
MEMORY TIMING CHAIN						CLEAR MEM. AVAILABLE F.F. ADDRESS	ENABLE	READ ENABLE	CL.U.B.(→1) + A SEQ. REPEAT B1→A0+ D SEQ CL.J1J2+K1→OTE SEQ			INHIBIT WRITE	SET MEMORY AVAILABLE F.F. ENABLE
TYPICAL A SEQUENCE					CLEAR S&B P→S, *SET S+1 INIT. MEM.	CLEAR D CLEAR OP. STEP	*CLEAR EXT. FUNC. CLEAR P S+1→P	CLEAR ABORT		*CLEAR JUMP	*CLEAR LOWER SEQUENCE F.F.s		
f	x	b	A										
00	0	0	ENABLE D	B									
				C						CLEAR A X→A		ENABLE A	
				D	CLEAR S&B, ADDER→S	CLEAR S+1 D S→D	CLEAR X D→X				Q→X	ENABLE C	
00	1	0	ENABLE D	B									
				C						CLEAR A X→A		ENABLE A	
				D	CL. S&B ADD→S INIT. MEM.						Q→X	ENABLE C	
00	0	1	ENABLE B	B	CLEAR S&B INIT. MEM.	CLEAR S+1 X→D						ENABLE D	
				C						CLEAR A X→A		ENABLE A	
				D	CLEAR S&B ADDER→S	CLEAR S+1 D S→D	CLEAR X D→X				Q→X	ENABLE C	
00	1	1	ENABLE B	B	CLEAR S&B INIT. MEM.	X→D						ENABLE D	
				C						CLEAR A X→A		ENABLE A	
				D	CL. S&B ADD→S INIT. MEM.						Q→X	ENABLE C	
01	0	0	ENABLE D	B									
				C						CLEAR A X→A		ENABLE A	
				D	CLEAR S&B ADDER→S	CLEAR S+1 D S→D	CLEAR X D→X					ENABLE C	
01	1	0	ENABLE D	B									
				C						CLEAR A X→A		ENABLE A	
				D	CLEAR S&B, ADD→S, INIT. MEM.							ENABLE C	
01	0	1	ENABLE B	B	CLEAR S&B INIT. MEM.	CLEAR S+1 X→D						ENABLE D	
				C						CLEAR A X→A		ENABLE A	
				D	CLEAR S&B ADDER→S	CLEAR S+1 D S→D	CLEAR X D→X					ENABLE C	
01	1	1	ENABLE B	B	CLEAR S&B INIT. MEM.	X→D						ENABLE D	
				C						CLEAR A X→A		ENABLE A	
				D	CL. S&B, ADD→S INIT. MEM.							ENABLE C	
02	0	0	ENABLE D	B									
				C				CLEAR Q X→Q				ENABLE A	
				D	CLEAR S&B, ADD→S	CLEAR S+1 D S→D	CLEAR X D→X					ENABLE C	
02	1	0	ENABLE D	B									
				C				CLEAR Q X→Q				ENABLE A	
				D	CLEAR S&B, ADD→S, INIT. MEM.							ENABLE C	
02	0	1	ENABLE B	B	CLEAR S&B, INIT. MEM.	CLEAR S+1 X→D						ENABLE D	
				C				CLEAR Q X→Q				ENABLE A	
				D	CLEAR S&B, ADD→S	CLEAR S+1 D S→D	CLEAR X D→X					ENABLE C	
02	1	1	ENABLE B	B	CLEAR S&B, INIT. MEM.	X→D						ENABLE D	
				C				CLEAR Q X→Q				ENABLE A	
				D	CLEAR S&B, ADD→S, INIT. MEM.							ENABLE C	
03	0	0	ENABLE D	B									
				C						X→A		ENABLE A	
				D	CLEAR S&B, ADD→S	CLEAR S+1 D S→D	CLEAR X D→X					ENABLE C	
03	1	0	ENABLE D	B									
				C						X→A		ENABLE A	
				D	CLEAR S&B, ADD→S, INIT. MEM.							ENABLE C	

* INDICATES THAT THIS COMMAND WILL TAKE PLACE DURING EACH RUN THRU TIMING CHAIN, REGARDLESS OF THE SEQUENCE.

COMPONENTS (UNLESS OTHERWISE INDICATED)	SIZE	VALUE	TOLERANCE
RESISTORS			
CAPACITORS			
INDUCTORS			

REFERENCE DRAWINGS — DRAWING TITLE	DWG. NO.

REVISIONS

LAYOUT | DATE | TITLE
DRAFTSMAN
CHECKER
WRITER | CONTROL | PROJECT OR PRODUCT
APPROVAL
PAGE | JACK NO.

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION
ST. PAUL
MINNESOTA

DRAWING NUMBER

DWG. NO.

REV.

t	X R	X D	A		T11	T10	T21	T12	T31	T32	T41	T42
03	0	1	ENABLE B	B	CLEAR SEZ INITMEM	CLEAR S+1 X→D						ENABLE D
				C						X→A		ENABLE A
				D	CLEAR SEZ ADD→S	CLEAR S+1 D S→D	CLEAR X D→X					ENABLE C
03	1	1	ENABLE B	B	CLEAR SEZ INIT MEM	X→D						ENABLE D
				C						X→A		ENABLE A
				D	CLEAR SEZ ADD→S INIT MEM							ENABLE C
04	0	0	ENABLE D	A								
				C			CLEAR X A→X			CLEAR A ADD→A		ENABLE A
				D	CLEAR SEZ ADD→S	CLEAR S+1 D S→D						ENABLE C
04	1	0	ENABLE D	E								
				C		CLEAR D X→D	CLEAR X A→X			CLEAR A ADD→A		ENABLE A
				D	CLEAR SEZ ADD→S INIT MEM							ENABLE C
04	0	1	ENABLE B	B	CLEAR SEZ INIT MEM	CLEAR S+1 X→D						ENABLE D
				C			CLEAR X A→X			CLEAR A ADD→A		ENABLE A
				D	CLEAR SEZ ADD→S	CLEAR S+1 S→D						ENABLE C
04	1	1	ENABLE B	B	CLEAR SEZ INIT MEM	X→D						ENABLE D
				C		CLEAR D X→D	CLEAR X A→X			CLEAR A ADD→A		ENABLE A
				D	CLEAR SEZ ADD→S INIT MEM							ENABLE C
05	0	0	ENABLE D	A								
				C		CLEAR D X'→D	CLEAR X A→X			CLEAR A ADD→A		ENABLE A
				D	CLEAR SEZ ADD→S	CLEAR S+1 D S→D	CLEAR X D→X					ENABLE C
05	1	0	ENABLE D	B								
				C		CLEAR D X→D	CLEAR X A→X			CLEAR A ADD→A		ENABLE A
				D	CLEAR SEZ ADD→S INIT MEM							ENABLE C
05	0	1	ENABLE B	B	CLEAR SEZ INIT MEM	CLEAR S+1 X→D						ENABLE D
				C		CLEAR D X'→D	CLEAR X A→X			CLEAR A ADD→A		ENABLE A
				D	CLEAR SEZ ADD→S	CLEAR S+1 S→D	CLEAR X D→X					ENABLE C
05	1	1	ENABLE B	B	CLEAR SEZ INIT MEM	X→D						ENABLE D
				C		CLEAR D X→D	CLEAR X A→X			CLEAR A ADD→A		ENABLE A
				D	CLEAR SEZ ADD→S INIT MEM							ENABLE C
06	0	0	ENABLE D	E								
				C					S→Z			ENABLE A
				D	CLEAR SEZ, INIT MEM, INH MEM, INH MEM	CLEAR S+1		X→C WAIT FOR MEM. ENABLE	ADD+D, SET ABORT			ENABLE C
06	0	1	ENABLE B	B	CLEAR SEZ INIT. MEM	CLEAR S+1 X'→D				CLEAR D S→D		ENABLE D
				C			CLEAR S ADD→S		D→Z			ENABLE A
				D	CLEAR SEZ INIT MEM INH MEM	CLEAR S+1				CLEAR D		ENABLE D
06	1	0	ENABLE B	B	CLEAR SEZ ADD→S INIT MEM			WAIT FOR MEM. ENABLE				
				C					D→Z			ENABLE A
				D	CLEAR SEZ, INIT MEM, INH MEM			X→D WAIT FOR MEM. ENABLE	ADD+D, SET ABORT	CLEAR D		ENABLE C
06	1	1	ENABLE B	B	CLEAR SEZ INIT MEM	X'→D				CLEAR D S+1→D		ENABLE D
				C			CLEAR S ADD→S		D→Z			ENABLE A
				D	CLEAR SEZ, INIT MEM, INH MEM							ENABLE D
07	0	1		B	CLEAR SEZ INIT MEM INH	CLEAR S+1						
				C								ENABLE A
				D	CLEAR S, ADD→S, INIT MEM, INH MEM,	CLEAR S+1						
07	1	0		B								
				D								
				J	CLEAR SEZ ADD→S, INIT MEM, INH M		P→Z		CLEAR D S+1→D			ENABLE A

* DURING THIS MEMORY CYCLE
Z→0 WILL BE INHIBITED

DWG. NO.

REV.

COMPONENTS (UNLESS OTHERWISE INDICATED)				REFERENCE DRAWINGS	
	SIZE	VALUE	TOLERANCE	DRAWING TITLE	DWG. NO.
RESISTORS					
CAPACITORS					
INDUCTORS					

REVISIONS	DATE	APPROVAL

LAYOUT
DRAFTSMAN
CHECKER
WRITER
APPROVAL

TITLE

CONTROL | PROJECT OR PRODUCT

PAGE | JACK NO.

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION
ST. PAUL
MINNESOTA

ERA FORM 115
DRAWING NUMBER

F	J		A		T11	T13	T21	T23	T31	T33	T41	T43
10	0	0	ENABLE A ENABLE C	A B C D			CL.S ENA→S; CL.X GR→X	CLEAR Q X→S	CLEAR X (SLAVE)'→Z			ENABLE A; ENABLE C
10	0	1	ENABLE A ENABLE C	A B C D			CLEAR S ENA→S		CL.Z (SLAVE)→Z CL.X ADD→X	CL. A X→A		ENABLE A; ENABLE C
10	1	0	ENABLE A ENABLE C	A B C D			CL.S ENA→S, CL.X GR→X	CLEAR Q X→S	CL.Z (SLAVE)'→Z, CL.X ADD→X	CLEAR A X→A		ENABLE A; ENABLE C
10	1	1	ENABLE D	A B C D	CLEAR SEL ADD→S, WAIT MEM.		CL.S ENA→S, CL.X GR→X	CL. Q X→S WAIT FOR MEM. ACK.	CL.Z (SLAVE)'→Z, CL.X ADD→X	CL. A SET ADD→A CLEAR D X→D	CLEAR Z & A	ENABLE C
11	0	0	ENABLE A ENABLE C	A B C D			CL.S ENA→S; CL.X Q→X	CLEAR Q X→Q	CL.Z (SLAVE)'→Z			ENABLE A; ENABLE C
11	0	1	ENABLE A ENABLE C	A B C D			CL.S ENA→S		CL.Z (SLAVE)'→Z, CL.X ADD→X	CL. A X→A		ENABLE A; ENABLE C
11	1	0	ENABLE A ENABLE C	A B C D			CL.S ENA→S; CL.X GR→X	CLEAR Q X→S	CL.Z (SLAVE)'→Z, CL.X ADD→X	CL. A X→A		ENABLE A; ENABLE C
11	1	1	ENABLE D	A B C D	CLEAR SEL ADD→S, WAIT MEM.		CL.S ENA→S; CL.X Q→X	CLEAR Q X→Q WAIT FOR MEM. ACK.	CL.Z (SLAVE)'→Z, CL.X ADD→X	CL. A SET ADD→A, SET Q=1 CLEAR D X→D	CL. Z & A SET Z=1	ENABLE A; ENABLE C ENABLE C
12	0	0	ENABLE C	A B C D				SET Q=1 ACK. SET Q=0				ENABLE A
12	0	1	ENABLE C	A B C D				SET Q=1 ACK.				ENABLE A
12	1	1	ENABLE C	A B C D				SET Q=1 ACK.				ENABLE A
13	0	0	ENABLE C	A B C D	CL. SEL REG=S, WAIT MEM. DATA		CL. S Q→S	X→D	D→ E→			ENABLE A
13	0	1	ENABLE C	A B C D			CLEAR X Q→X	X'→D	CLEAR X D→X	CLEAR Q X→Q		ENABLE A
13	1	0	ENABLE D	A B C D	CLEAR, ADD→L, WAIT MEM. DATA		CL. A A→X	X→D	D→ R→			ENABLE A
13	1	1	ENABLE C	A B C D			CL. X A→X	X'→D	CLEAR X D→X	CLEAR A X→A		ENABLE A

COMPONENTS (UNLESS OTHERWISE INDICATED)				REFERENCE DRAWINGS	
	SIZE	VALUE	TOLERANCE	DRAWING TITLE	DWG. NO.
RESISTORS					
CAPACITORS					
INDUCTORS					

REVISIONS			
			APPROVAL
			DATE
			CHKD. BY
			DRAWN BY
			CHG. NO.

LAYOUT	DATE	TITLE
DRAFTSMAN		
CHECKER		
WRITER		CONTROL PROJECT OR PRODUCT
APPROVAL		PAGE DWG. NO.

Remington Rand Univac
DIVISION OF SPERRY RAND CORPORATION
ST. PAUL
MINNESOTA
DRAWING NUMBER

#8

F	J	A	T11	T12	T21	T28	T31	T32	T41	T42
14	0	0	A CL SEL P=S, INIT MEM A P=S → JUMP ADD=S → JUMP & CONTINUE THRU NORMAL A SEQ.	CL D & OP STEP	CL F SWAP	CL AREA		CL JUMP RT	SET JUMP → A+D	ENABLE A
14	0	1	A CL SEL P=S, INIT MEM A P=S → JUMP ADD=S → JUMP & CONTINUE THRU NORMAL A SEQ.	CL D & OP STEP	CL F SWAP	CL AREA		CL JUMP FF	SET JUMP → A+D	ENABLE A
14	1	0	A CL SEL P=S, INIT MEM A P=S → JUMP ADD=S → JUMP & CONTINUE THRU NORMAL A SEQ.	CL D & OP STEP	CL F SWAP	CL AREA		CL JUMP FT	SET JUMP → A+D	ENABLE A
14	1	1	A CL SEL P=S, INIT MEM A P=S → JUMP ADD=S → JUMP & CONTINUE THRU NORMAL A SEQ.	CL D & OP STEP	CL F SWAP	CL AREA		CL JUMP TF	SET JUMP → A+D	ENABLE A
15	0	0	A CL SEL P=S, INIT MEM A P=S → JUMP ADD=S → JUMP & CONTINUE THRU NORMAL A SEQ.	CL D & OP STEP	CL F SWAP	CL AREA		CL JUMP TT	SET JUMP → A+D	ENABLE A
15	0	1	A CL SEL P=S, INIT MEM A P=S → JUMP ADD=S → JUMP & CONTINUE THRU NORMAL A SEQ.	CL D & OP STEP	CL F SWAP	CL AREA		CL JUMP FF	SET JUMP → A+D	ENABLE A
15	1	0	A CL SEL P=S, INIT MEM	CL D & OP STEP	CL F SWAP	CL AREA		CL JUMP FT	SET JUMP → A+D	ENABLE A
15	1	1	A CL SEL P=S, INIT MEM	CL D & OP STEP	CL F SWAP	CL AREA		CL JUMP TF	SET JUMP → A+D	ENABLE A
16	0	0	A CL SEL P=S, INIT MEM A P=S → JUMP ADD=S → JUMP & CONTINUE THRU NORMAL A SEQ.	CL D & OP STEP	CL F SWAP	CL AREA		CL JUMP TT	SET JUMP → A+D	ENABLE A
16	0	1	A CL SEL P=S, INIT MEM A P=S → JUMP ADD=S → JUMP & CONTINUE THRU NORMAL A SEQ.	CL D & OP STEP	CL F SWAP	CL AREA		CL JUMP FF	SET JUMP → A+D	ENABLE A
16	1	0	A CL SEL P=S, INIT MEM A P=S → JUMP ADD=S → JUMP & CONTINUE THRU NORMAL A SEQ.	CL D & OP STEP	CL F SWAP	CL AREA		CL JUMP FT	SET JUMP → A+D	ENABLE A
17	0	0	ENABLE D B C D CL SEL CONS, INIT MEM, WM	CL A D → X		X → D	C = B			ENABLE A
17	1	0	ENABLE D B C D CL SEL B, ADD=S, INIT MEM, WM	CL A AM		X → D	D = B			ENABLE A

COMPONENTS UNLESS OTHERWISE INDICATED:				REFERENCE DRAWINGS	
	SIZE	MASK	TOLERANCE	DRAWING TITLE	DRAWING NO.
RESISTORS					
CAPACITORS					
INDUCTORS					

REVISIONS					

LAYOUT	DATE	TITLE	Remington Rand Univac DIVISION OF SPERRY RAND CORPORATION ST. PAUL MINNESOTA
DRAFTSMAN			
CHECKER			
WRITER		CONTRACT	PROJECT OR PRODUCT
APPROVAL		SIZE	DRAWING NUMBER

49

RECOMMENDED REPAIR PARTS LIST
PROVISIONING FORMAT

FX 6832-1

SYMBOL NO. PREFIX OR UNIT NOMENCLATURE				PRIME CONTRACTORS		QTY. PER ASSY.	QTY. PER COMP.	QTY PER END ITEM	SHELF LIFE	TOTAL RECM/ ORDERED	UNIT PRICE	EXTENDED UNIT PRICE
ITEM OR SEQUENCE NUMBER	INDENT	REFERENCE SYMBOL NO.	ITEM NAME	PART NUMBER / LONG PART NO CODE							DOLLARS CTS M	DOLLARS CTS
SMR CODE		FEDERAL STOCK NUMBER	ITEM AND LOT NO. / FEDERAL MFR. CODE	MANUFACTURERS PART NUMBER / LONG PART NO CODE		RECM MAINT QTY/ FACT	RECM OVHL QTY/ FACT	USEABLE ON CODE			SPARES ALLOCATION	
1.		1260-593-5400 N	Bd. Assy. Inv. 90536	250010 250010				19		3	10.26	30.78
2.		1260-600-0821 N	Bd. Assy. Inv. 90536	250020 250020				3		3	12.38	37.14
3.		1260-600-0822 N	Bd. Assy. Inv. 90536	250040 250040				27		3	12.38	37.14
4.		1260-593-5402 N	Bd. Assy. Inv. 90536	250050 250050				11		3	12.38	37.14
5.		1260-600-0823 N	Bd. Assy. Inv. 90536	250070 250070				147		10	14.58	145.80
6.		1260-600-0826 N	Bd. Assy. 90536	250150 250150				2		3	18.75	56.25
7.		1260-600-0827 N	Bd. Assy. Ampl. 90536	250160 250160				4		3	18.22	54.66
8.		1260-593-5433 N	Bd. Assy. Inv. 90536	250170 250170				9		3	17.84	53.52
9.		6110-860-9643 N	Bd. Assy. Ampl. 90536	250200 250200				3		3	13.07	39.21
10.		1260-593-5443 N	Bd. Assy. Net. 90536	250240 250240				1		3	37.71	113.13
11.		1260-593-5447 N	Bd. Assy. Net. 90536	250250 250250				2		3	30.77	92.31
12.		7440-860-9644 N	Bd. Assy. Ampl. 90536	250270 250270				5		3	42.57	127.71
13.		7440-860-9645 N	Bd. Assy. Ampl. 90536	250280 250280				2		3	28.91	86.73
14.		7440-860-9646 N	Bd. Assy. Inv. 90536	250290 250290				4		3	14.62	43.86
15.		7440-855-9344 N	Bd. Assy. Ampl. 90536	250300 250300				2		3	43.86	131.58

CONTRACT NUMBER: NObsr 87204

NOMENCLATURE: COMMERCIAL REPAIR PARTS LISTING FOR UNIVAC DIGITAL TRAINER AND ADAPTER AT INSTALLATION WITHOUT EXISTING NTDS EQUIPMENT

MODEL/TYPE NO.:

CONTRACTOR: UNIVAC Div. of Sperry Rand Corp.

DATE OF LIST: 28 Nov. 1962

REVISION:

A-1454

RECOMMENDED REPAIR PARTS LIST
PROVISIONING FORMAT

FX 6832-1

SYMBOL NO. PREFIX OR UNIT NOMENCLATURE

ITEM OR SEQUENCE NUMBER	SMR CODE	REFERENCE SYMBOL NO. / FEDERAL STOCK NUMBER	ITEM NAME / ITEM AND LOT NO. / FEDERAL MFR. CODE	PRIME CONTRACTOR PART NUMBER / LONG PART NO. CODE / MANUFACTURERS PART NUMBER / LONG PART NO. CODE	QTY. PER ASSY. / RECM MAINT QTY/FACT	QTY. PER COMP. / RECM OVHL QTY/FACT	QTY. PER END ITEM / USEABLE ON CODE	SHELF LIFE	TOTAL RECM ORDERED	UNIT PRICE DOLLARS CTS	EXTENDED UNIT PRICE DOLLARS CTS	SPARES ALLOCATION
16.		7440-860-9649 N	Bd. Assy. Ampl. 90536	250330 250330			15		3	29.22	87.66	
17.		7440-860-9650 N	Bd. Assy. Ampl. 90536	250341 250341			8		3	30.94	92.82	
18.		7440-860-9653 N	Bd. Assy. Xfmr. 90536	250370 250370			3		3	15.13	45.39	
19.		1260-600-0828 N	Bd. Assy. Inv. 90536	250410 250410			100		7	31.21	218.47	
20.		1260-600-0829 N	Bd. Assy. FF 90536	250420 250420			70		5	37.26	186.30	
21.		6110-860-9657 N	Bd. Assy. Ampl. 90536	250430 250430			18		3	31.06	93.18	
22.		1260-593-5453 N	Bd. Assy. Ampl. 90536	250690 250690			3		3	20.50	61.50	
23.		7440-860-9630 N	Bd. Assy. Ampl. 90536	250720 250720			9		3	27.49	82.47	
24.			DELETE									
25.		5895-971-1995 N	Bd. Assy. Ampl. 90536	265562 265562			16		3	29.94	89.82	
26.		5895-971-1996 N	Bd. Assy. Ampl. 90536	265565 265565			16		3	37.75	113.25	
27.		5915-971-1997 N	Bd. Assy. Isol. 90536	265568 265568			4		3	18.26	54.78	
28.		7440-971-1998 N	Bd. Assy. Inv. 90536	265677 265677			1		3	23.37	70.11	
29.		7440-971-1999 N	Bd. Assy. Reactor 90536	265680 265680			10		3	34.51	103.53	
30.		7440-971-2000 N	Bd. Assy. Inv. 90536	265682 265682			15		3	25.06	75.18	

CONTRACT NUMBER: NObsr 87204

NOMENCLATURE: COMMERCIAL REPAIR PARTS LISTING FOR UNIVAC DIGITAL TRAINER AND ADAPTER AT INSTALLATION WITHOUT EXISTING NTDS EQUIPMENT

MODEL TYPE NO.

CONTRACTOR: UNIVAC Div. of Sperry Rand Corp.

DATE OF LIST: 28 Nov. 1962

REVISION

RECOMMENDED REPAIR PARTS LIST
PROVISIONING FORMAT

PX 6832-1

ITEM OR SEQUENCE NUMBER	SYMBOL NO. PREFIX OR UNIT NOMENCLATURE / REFERENCE SYMBOL NO.	ITEM NAME / FEDERAL MFR. CODE	PRIME CONTRACTORS / MANUFACTURERS — PART NUMBER / LONG PART NO. CODE	QTY. PER ASSY. / RECM MAINT QTY/FACT	QTY. PER COMP. / RECM OVHL QTY/FACT	QTY. PER END ITEM / USABLE ON CODE	SHELF LIFE	TOTAL RECM/ ORDERED	UNIT PRICE DOLLARS CTS M	EXTENDED UNIT PRICE DOLLARS CTS
SMR CODE	FEDERAL STOCK NUMBER	ITEM AND LOT NO.	PART NUMBER / LONG PART NO. CODE						SPARES ALLOCATION	
31.	5910-971-2001 P N	Capacitor 53021	902867-01 DCM20M/30			2		2	26.35	52.70
32.	5910-112-7406 N	Capacitor 81349	903703-01 CP53B1FF504K			1		2	4.31	8.62
33.	5910-542-7372 N	Capacitor 56289	903878-01 150D106X0020B2			2		2	2.65	5.30
34.	5910-893-3785 N	Capacitor 00656	906378-11 MC80V104AM			15		3	5.15	15.45
35.	5910-971-2002 N	Capacitor 81350	910109-02 CL25BH320UP1			6		2	8.62	17.24
36.	5950	Choke 80023	4913200 S3683			1		1	21.45	21.45
37.	5950	Choke 80023	4913203 S3684			1		1	15.62	15.62
38.	5925-971-2005 N	Circuit Breaker 73803	4911933 C6363-1-6			1		1	31.25	31.25
39.		DELETE								
40.	5935-971-2007 N	Connector 90002	904379 5284			2		1	2.37	2.37
41.	5935-971-2008 N	Connector 90002	904380 5278			1		1	1.62	1.62
42.	5935-841-6206 N	Connector 71468	906013 DPD4500-1387			8		1	28.18	28.18
43.	5935-844-5223 N	Connector 02660	906155 143-825			711		5	4.77	23.85
44.	5935-841-6205 N	Connector 71468	906489 DPD4500-1388			11		1	24.73	24.73
45.	5935-898-9330 N	Connector 71468	908220 DPD4500-4301			2		1	45.28	45.28

CONTRACT NUMBER: NObsr 87204

NOMENCLATURE: COMMERCIAL REPAIR PARTS LISTING FOR UNIVAC DIGITAL TRAINER AND ADAPTER AT INSTALLATION WITHOUT EXISTING NTDS EQUIPMENT

MODEL/TYPE NO.

CONTRACTOR: UNIVAC Div. of Sperry Rand Corp.

DATE OF LIST: 28 Nov. 1962

REVISION

SYMBOL NO. PREFIX OR UNIT NOMENCLATURE

ITEM OR SEQUENCE NUMBER	A S T	REFERENCE SYMBOL NO.	ITEM NAME	PRIME CONTRACTORS PART NUMBER	LONG PART NO. CODE	QTY PER ASSY	QTY PER COMP	QTY PER END ITEM	SHELF LIFE	TOTAL RECM ORDERED	UNIT PRICE DOLLARS CTS	EXTENDED UNIT PRICE DOLLARS CTS
SMR CODE		FEDERAL STOCK NUMBER	ITEM AND LOT NO / FEDERAL MFR. CODE	MANUFACTURERS PART NUMBER	LONG PART NO. CODE	RECM MAINT QTY/ FACT	RECM OVHL QTY/ FACT	USABLE ON CODE			SPARE ALLOCATION	
46.		5935-971-5606 N	Connector 75173	911510 P308CCT				2		1	.94	.94
47.		5920-280-5031 N	Fuse 81350	907783-05 F02GR250A				2		10	.08	.80
48.		5920-280-4960 N	Fuse 81350	907783-11 F02G2R00A				1		10	.08	.80
49.			DELETE									
50.		5920-281-0225 N	Fuse 81350	907783-15 F02G6R00A				1		10	.10	1.00
51.		5920-280-3562 N	Fuse 81350	907783-20 F03D20R0A				1		10	.08	.80
52.		5920-850-4986 N	Fuseholder 71400	908830 HKPELQRWZJ				6		2	.71	1.42
53.		1260-733-9339 N	Connector 14028	908029 99011				17		2	2.98	5.96
54.		5935-257-7044 N	Jack Tip 78947	903136 118930B				15		5	5.75	28.75
55.		5945-897-8608 N	Relay 71482	905246 B226068				6		1	14.94	14.94
56.		5945-899-2446 N	Relay 71482	908130 A282581				9		1	14.31	14.31
57.			DELETE									
58.		5905-279-1734 N	Resistor 81350	900486-27 RC32GF330J				1		2	.17	.34
59.		5905-299-2051 N	Resistor 81350	900486-55 RC32GF471J				2		2	.18	.36
60.		5905-279-1894 N	Resistor 81350	900489-23 RC20GF820J				6		2	.13	.26

CONTRACT NUMBER: NObsr 87204

NOMENCLATURE: COMMERCIAL REPAIR PARTS LISTING FOR UNIVAC DIGITAL TRAINER AND ADAPTER AT INSTALLATION WITHOUT EXISTING NTDS EQUIPMENT

MODEL TYPE NO.

CONTRACTOR: UNIVAC Div. of Sperry Rand Corp.

DATE OF LIST: 28 Nov. 1962

REVISION

A-1654

ITEM OR SEQUENCE NUMBER	SMR CODE	REFERENCE SYMBOL NO. / FEDERAL STOCK NUMBER	ITEM NAME / ITEM AND LOT NO.	FEDERAL MFR. CODE	PRIME CONTRACTORS: PART NUMBER / LONG PART NO. CODE	MANUFACTURERS: PART NUMBER / LONG PART NO. CODE	QTY. PER ASSY.	QTY. PER COMP.	QTY. PER END ITEM / USABLE ON CODE	SHELF LIFE	TOTAL RECM'D ORDERED	UNIT PRICE DOLLARS CTS	EXTENDED UNIT PRICE DOLLARS CTS	SPARES ALLOCATION: RECM MAINT QTY/FACT / RECM OVHL QTY/FACT
61.	N	5905-192-3973	Resistor	81350	900489-41	RC20GF471J			3		2	.50	1.00	
62.	N	5905-195-6806	Resistor	81350	900489-49	RC20GF102J			1		2	.13	.26	
63.	N	5905-270-7084	Resistor	81350	900507-10	RW55G250			4		2	.94	1.88	
64.	N	5905-191-7668	Resistor	81350	900507-13	RW55G450			15		4	.94	3.76	
65.	N	5905-899-8695	Resistor	09145	910625-04	W10L100			5		2	16.06	32.12	
66.	N	5905-821-7063	Resistor	06228	908302-01	TM1/4-100 Ohm			1		2	8.33	16.66	
67.	N	5960-577-6214	Diode	03984	907186	1N538			11		4	3.06	12.24	
68.	N	5960-811-5799	Diode	81483	908293	1N1202			16		4	5.73	22.92	
69.	N	5930-050-2707	Switch	81350	908106	ST52N			1		2	2.65	5.30	
70.	N	5930-844-6513	Switch	91929	908832	13AT403T2			4		2	19.75	39.50	
71.	N	5930-844-6514	Switch	91929	908833	13AT401T2			3		1	19.66	19.66	
72.	N	5999-856-7202	Switch	07137	911842-04	MBL-S-773A			1		2	5.10	10.20	
73.	N	5999-856-7201	Switch	07137	911842-08	MBLAA1-000-2102			5		5	5.10	25.50	
74.	N	5999-856-7204	Switch	07137	911873-01				96		20	6.60	132.00	
75.	N	5950	Transformer	80023	4913198	S3682			1		1	45.41	45.41	

CONTRACT NUMBER: NObsr 87204

NOMENCLATURE: COMMERCIAL REPAIR PARTS LISTING FOR UNIVAC DIGITAL TRAINER AND ADAPTER AT INSTALLATION WITHOUT EXISTING NTDS EQUIPMENT

MODEL/TYPE NO.

CONTRACTOR: UNIVAC Div. of Sperry Rand Corp.

DATE OF LIST: 28 Nov. 1963

REVISION

A-1494

RECOMMENDED REPAIR PARTS LIST
PROVISIONING FORMAT

SYMBOL NO. PREFIX OR UNIT NOMENCLATURE / ITEM OR SEQUENCE NUMBER	SMR CODE	REFERENCE SYMBOL NO. / FEDERAL STOCK NUMBER	ITEM NAME / ITEM AND LOT NO.	FEDERAL MFR. CODE	PRIME CONTRACTORS: PART NUMBER / MANUFACTURERS: PART NUMBER	LONG PART NO. CODE	QTY PER ASSY. / RECM MAINT QTY-FACT	QTY PER COMP. / RECM OVHL QTY-FACT	QTY PER END ITEM / USABLE ON CODE	SHELF LIFE	TOTAL RECM ORDERED	UNIT PRICE DOLLARS CTS	EXTENDED UNIT PRICE DOLLARS CTS	SPARES ALLOCATION
76.	N	5960-752-5897	Transistor	40931	908152 2N575				1		3	62.59	187.77	
77.	N	5960-583-0652	Transistor	40931	908153 2N539				3		3	18.85	56.55	
78.	N	6145-971-2010	Cable	70903	904134-03 17425				1		1	1.52	1.52	
79.	N	5910-971-2011	Capacitor	80183	905909-01 32D326				1		2	13.14	26.28	
80.	N	5910-688-7211	Capacitor	80183	4912108-08 150D157X0015S2				2		2	18.18	36.36	
81.	N	5935-971-2012	Connector	74545	904381 5270				1		1	1.87	1.87	
82.	N	5935-665-4680	Connector	75173	911509 P308CCT				2		1	1.35	1.35	
83.	N	5930-971-2013	Switch	91812	909927-09 1900-1E10				1		1	16.18	16.18	
84.	N	5930-971-2015	Switch	07137	911873-00 MBL-S-391				42		15	6.00	90.00	
85.		5950	Transformer	80023	4913199 S2928A				1		1	13.75	13.75	
86.			DELETE											
87.			DELETE											
88.	N	7440-860-9598	Bd. Assy. VoltProt.	90536	250700 250700				1		3	29.47	88.41	
89.	Z	1260-593-5457	Bd. Assy. Delay	90536	250910 250910				3		3	52.00	156.00	
90.	N	1260-600-0824	Bd. Assy. Inv.	90536	250080 250080				34		3	31.77	95.31	

CONTRACT NUMBER: NObsr 87204

NOMENCLATURE: COMMERCIAL REPAIR PARTS LISTING FOR UNIVAC DIGITAL TRAINER AND ADAPTER AT INSTALLATION WITHOUT EXISTING NTDS EQUIPMENT

MODEL/TYPE NO.

CONTRACTOR: UNIVAC Div. of Sperry Rand Corp.

DATE OF LIST: 28 Nov. 1962

REVISION

RECOMMENDED REPAIR PARTS LIST
PROVISIONING FORMAT

SYMBOL NO. PREFIX OR UNIT NOMENCLATURE / ITEM OR SEQUENCE NUMBER	CHANGE CODE	REFERENCE SYMBOL NO. / SMR CODE	FEDERAL STOCK NUMBER	ITEM NAME / ITEM AND LOT NO.	FEDERAL MFR. CODE	PRIME CONTRACTORS: PART NUMBER	LONG. PART NO. CODE	MANUFACTURERS: PART NUMBER	LONG PART NO. CODE	QTY. PER ASSY. / RECM MAINT. QTY./ FACT	QTY. PER COMP. / RECM OVHL. QTY./ FACT	QTY PER END ITEM / USEABLE ON CODE	SHELF LIFE	TOTAL RECM/ ORDERED	UNIT PRICE DOLLARS CTS M	EXTENDED UNIT PRICE DOLLARS CTS
															SPARES ALLOCATION	
91.		1260-600-0825	N	Bd. Assy. Inv.	90536	250090		250090				47		4	31.70	126.80
92.		1260-593-5403	N	Bd. Assy. Osc.	90536	250100		250100				1		3	27.57	82.71
93.		1260-593-5405	N	Bd. Assy. Ampl.	90536	250140		250140				4		3	28.09	84.27
94.				DELETE												
95.				DELETE												
96.		5910-971-2014	N	Capacitor	80183	906678-01						8		3	.35	1.65
97.		1260-716-0566	N	Bd. Assy.	90536	250920		250920				2		3	39.82	119.46
98.		5910		Capacitor	14655	900261-01		CP53B4FF254V				1		2	1.42	2.84
99.		5910		Capacitor	71590	911694-04		DA717 100uuf				21		3	.91	2.73
100.		5910		Capacitor	14655	4913616		PM-4W1				1		2	1.05	2.10
101.		5935		Connector	71468	4913604		RNK22-31SL				1		1	9.10	9.10
102.		5935		Connector	71468	4913605		RNK22-27C1-2				1		1	9.10	9.10
103.		5915		Filter	56289	4913603		10JX52				2		2	18.20	36.40
104.		5355		Knob	37942	901193		366-1				1		1	.75	.75

CONTRACT NUMBER: NObsr 87204

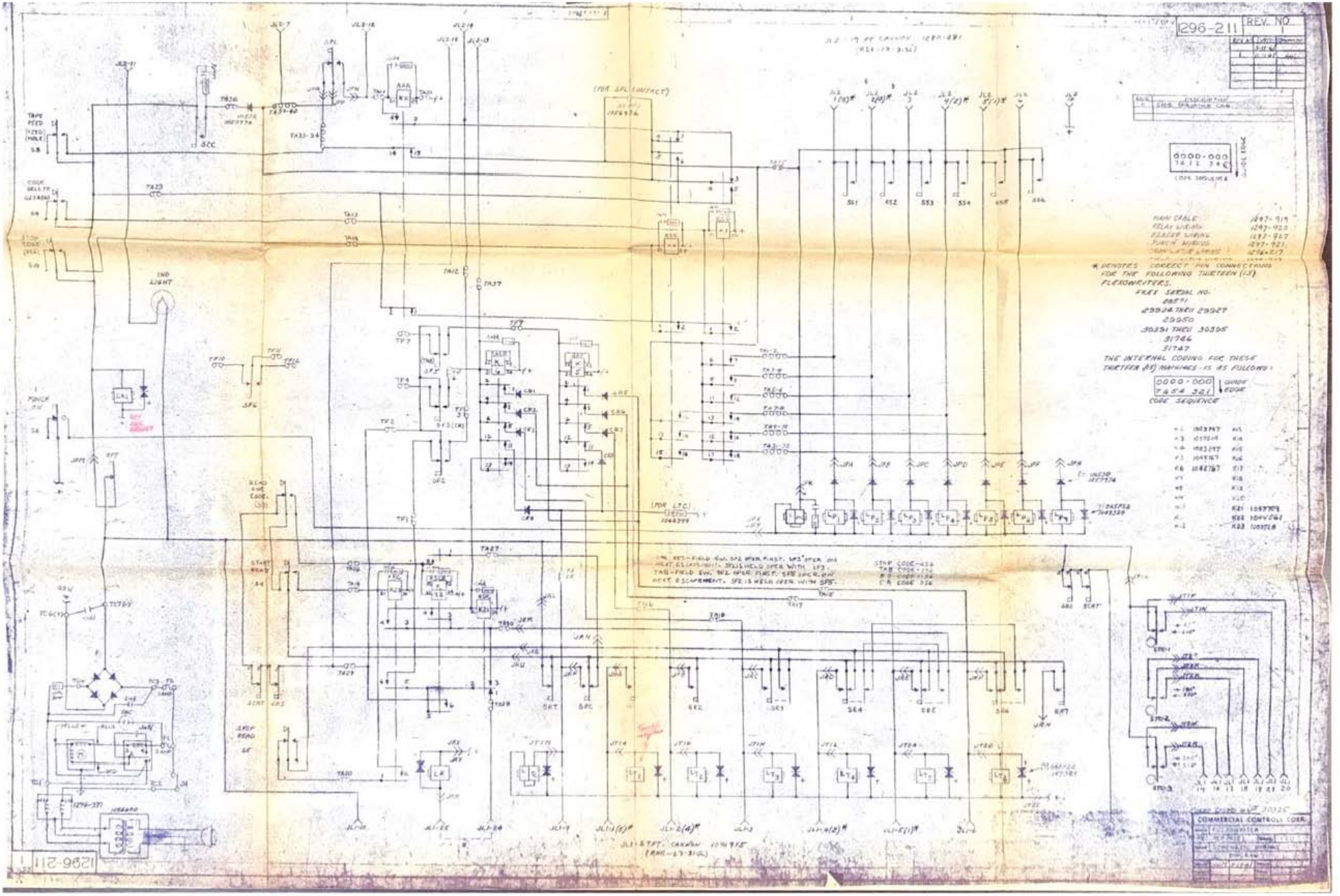
NOMENCLATURE: COMMERCIAL REPAIR PARTS LISTING FOR UNIVAC DIGITAL TRAINER AND ADAPTER AT INSTALLATION WITHOUT EXISTING NTDS EQUIPMENT

MODEL/TYPE NO.

CONTRACTOR: UNIVAC Div. of Sperry Rand Corp.

DATE OF LIST: 28 Nov. 1962

REVISION


1296-211
REV. NO
COMMERCIAL CONTROLS CORP.
1296-211

SEMI CONDUCTOR

CROSS REFERENCE DATA

UNIVAC DIGITAL TRAINER

907186 Diode

Silicon Power Diode — Type JAN 1N538M
Vendor: General Electric — Used On:
Motorola — 250720 Card
Raytheon
Transitron

907801 Diode

Same as 908290
Parts are interchangeable

907806 Diode

Silicon Switching Diode — Type S231
Vendor: Sperry Semiconductor — Used On:
250641 Card
Main Frame

908013 Transistor

PNP Germanium Switching — Type T1876
Vendor: Philco — Used On:
Logic Cards

908014 Transistor

PNP Germanium Switching — Type 2N1729
Vendor: Texas Instrument — Used On:
Gensral Electric
250191 Card
250321 Card
250850 Card

908015 Transistor

NPN Germanium Switching — Type 2N1730
Vendor: Texas Instrument — Used On:
General Electric
250270 Card
250300 Card
250850 Card

908152 Transistor

PNP Germanium Power — Type USA 2N575
Vendor: Mpls. Honeywell — Used On:
Main Frame

908153 Transistor

PNP Germanium Power Type JAN 2N539M
Vendor: Mpls. Honeywell Used On:
Main Frame

908181 Transistor

PNP Germanium Switching Type B1779
Vendor: Bendix SemiConductor Used On:
250720 Card

908287 Transistor

PNP Germanium Switching Type 2N1731
Vendor: Texas Instrument Used On:
250191 Card
250200 Card
250720 Card

908290 Diode

Germanium Switching Diode Type 1N3592
Vendor: National Transistor Used On:
Clevite
Logic Cards

908293 Diode

Silicon Power Diode Type USAF 1N1202
Vendor: General Electric Used On:
Westinghouse
Main Frame

908298-01 Diode

Silicon Voltage Regulator Type USN 1N753AM
Vendor: Continental Device Used On:
Corporation
250910 Card
250920 Card

908298-03 Diode

Silicon Voltage Regulator Type USN 1N758AM
Vendor: Continental Device Used On:
Corporation
250910 Card
250920 Card

908328 Transistor

PNP Germanium High Voltage Type 2N398
Vendor: RCA Used On:
Motorola
250090 Card

908380-01 Diode

Silicon Voltage Regulator Type 1N969B
Vendor: Motorola Used On:

250700 Card

911890 Transistor

Interchangeable with 908013 or 908014.

4911929 Transistor

PNP Germanium Switching Type T2347
Vendor: Philco Used On:

250100 Card
250170 Card
250200 Card
250330 Card