

Computer Service Bulletin # XII-501

**Monrobot XII
Computer
SERVICE MANUAL**

MONROE 

Monroe International Inc., Orange, New Jersey . . . Technical Publications Department

C O N T E N T S

| TITLE | SECTION |
|---|---------|
| System Description | I |
| Theory of Operation | II |
| Circuits | III |
| General Programming and Operating Rules | IV |
| The Loader | V |
| Adjustment and Tests | VI |
| Location of Assemblies and Running Sheets | VII |
| Parts Catalog | VIII |

COMPUTER SERVICE BULLETIN #X11-501

- N O T I C E -

The information contained herein is the confidential property of Monroe International, Inc. It must not be disclosed to others, reproduced, or used for any purpose except as authorized in writing by Monroe International, Inc.

PRINTED IN U. S. A.

July, 1966

BOOK NUMBER _____

SECTION I

SYSTEM DESCRIPTION

SECTION I

SYSTEM DESCRIPTION

The Monrobot XII "MicroMonrobot" is a small general purpose digital computer which uses a stored program and manual input of data. The central processing unit of the system consists of logic circuits, input/output circuits, magnetic drum storage and power supplies. In operation, it requires an input and an output device, which might typically be a keyboard and a typewriter respectively, although any standard Monroe input or output device can be used. The system has been designed to handle as many as three input and three output devices at once.

A general block diagram of the system appears in figure I-1.

The Monrobot XII is a completely solid-state computer, utilizing diode logic and transistor amplifiers and signal shapers. It operates in the binary number system; its number size or sector capacity is 36 binary digits or bits (representing the serial binary equivalent of ten decimal digits). Thus, the maximum decimal number that can be stored in a drum storage sector is $2^{36}-1$, or:

$$111111111111111111111111111111111111 = 68,719,476,735.$$

The actual length of a storage sector is 37 bits. However, the extra bit, which is called the flag bit, is not available for the storage of information.

The 36 information bits of a sector, or word, are processed serially -- that is, one bit at a time -- starting with the low-order bit. It is the low-order bit of any word which arrives first at any given point. The Monrobot XII is a common storage computer, in that the same sectors are used both for the storage of numbers or information and for the storage of program words or instructions. The two are not distinguished in any way in the computer memory.

Sixteen four-bit instruction codes are used in the Monrobot XII; by defining additional conditions in conjunction with some of the codes, these are expanded into twenty-one basic commands. A program instruction word, therefore, has a capacity of nine such four-bit commands, plus a low-order flag bit.

The 21 command codes are used to implement programs in the Monrobot XII by means of a concept called "microprogramming". Complex commands -- such as Multiply, Divide, Decimal Shift, etc. -- are simulated by stored subroutines of simple micro-commands -- the 21 basic commands referred to above. Thereby, a variety of applications can be programmed with a minimum of hardware. It should be noted that no addresses accompany the micro-commands.

Information in the Monrobot XII is stored on a magnetic drum which rotates at approximately 3500 rpm. This drum has eight tracks of data: five General Storage tracks, one fast-access Accumulator and Orders Channel track, one Clock

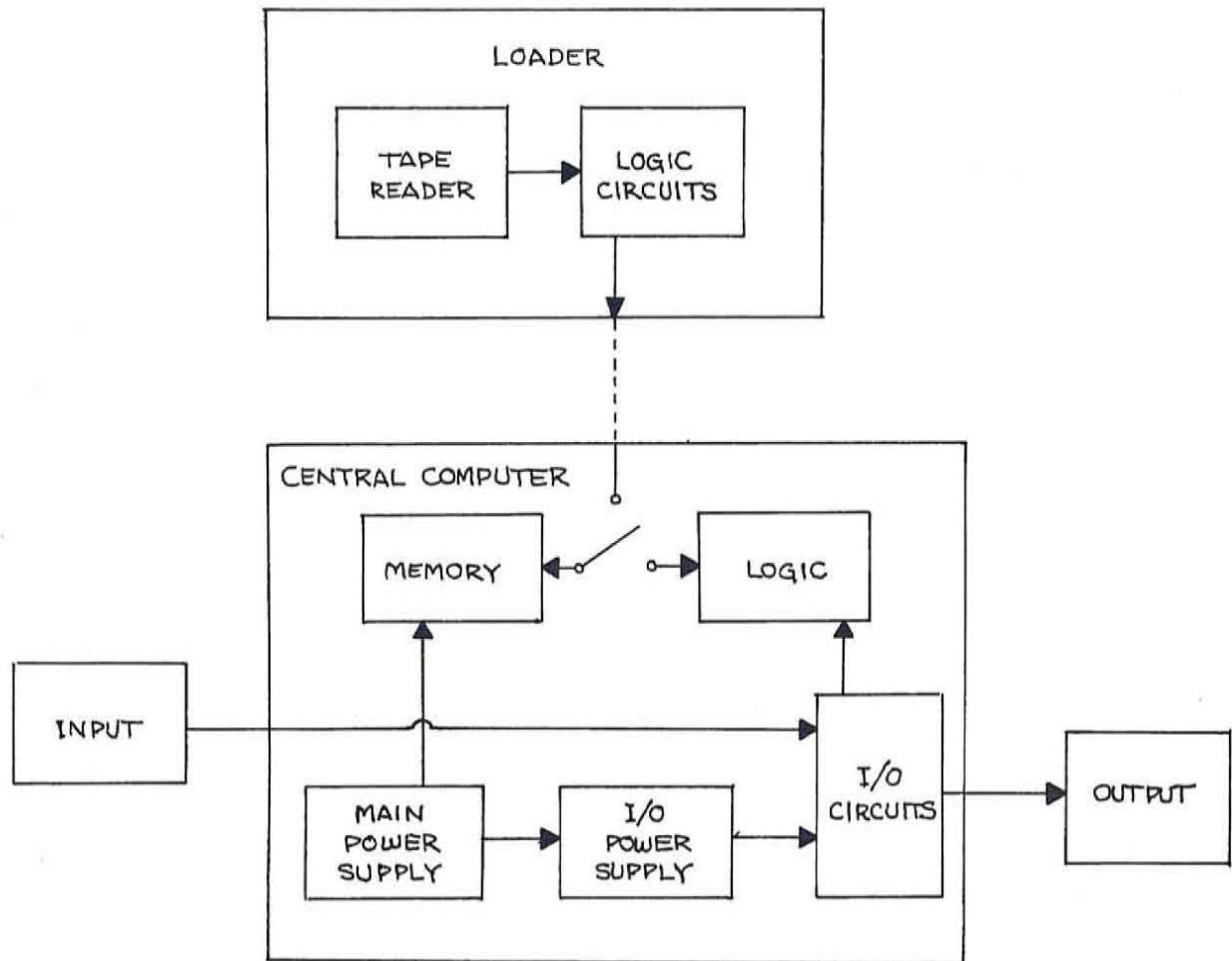


Figure I-1. Monrobot XII System Block Diagram

SYSTEM DESCRIPTION

track and one Sector track. In practice, however, only four of the five General Storage tracks are used at any given time. The fifth track is only an alternate to one of the other four; its purpose is to store special routines. There are separate playback and record heads for each track except the Clock and Sector tracks, which have only playback heads.

The information is stored on the drum in pulses of two different types, having the binary values 0 or 1. Each track is divided into 75 sectors, each sector containing space for one word of data, and each word consisting of 37 bits. Thus, there are 2775 bits on a track.

Assuming a nominal drum speed of 3500 rpm, some basic parameters of the computer become evident:

| | | |
|---------------------|---|-------------------|
| 1 drum revolution | = | 17.1 milliseconds |
| 1 word time | = | 228 microseconds |
| 1 bit time | = | 6.16 microseconds |
| bit pulse frequency | = | 161 kc per second |

The Monrobot XII has no static or rest mode of operation; once started, it is always under the control of the program. Hence, the computer is not operable unless a program has been loaded into General Storage. To accomplish this, a separate loader unit is temporarily connected to the system by a cable. This loading device reads a micro-program from punched paper tape and stores it in the computer's storage. During this operation, the computer's logic circuits are inoperative.

SECTION II

THEORY OF OPERATION

| | Page |
|--|-------|
| II-1. TIMING | II-1 |
| II-1-1. The Clock Track | II-1 |
| II-1-2. The Sector Track | II-1 |
| II-1-3. The F8 Flip-Flop | II-1 |
| II-1-3-a. The Precess-Execute Sequence | II-3 |
| II-2. MEMORY | II-4 |
| II-2-1. General Storage | II-4 |
| II-2-2. The Accumulator and Orders Channel | II-4 |
| II-2-3. The Condition Register K | II-6 |
| II-3. CONTROL | II-7 |
| II-3-1. Orders Precession | II-7 |
| II-3-2. Order Word Transfer | II-7 |
| II-3-3. Track Selection | II-8 |
| II-3-3-a. Alternate Track Selection | II-11 |
| II-3-4. On/Off Control | II-11 |
| II-3-4-a. Control Switches | II-11 |
| II-3-4-b. Sequencing | II-11 |
| II-4. THE PRIMARY COMMANDS | II-15 |
| II-4-1. Skip | II-17 |
| II-4-2. Quadruple Left Shift | II-18 |
| II-4-3. Leap | II-21 |
| II-4-4. Input | II-23 |
| II-4-5. Output | II-28 |
| II-4-6. Complement | II-33 |
| II-4-7. Change Track | II-34 |
| II-4-8. Left Shift | II-35 |
| II-4-9. Transfer | II-37 |
| II-4-10. Add | II-38 |
| II-4-11. Jump | II-39 |
| II-4-12. Multiply By Five | II-41 |
| II-4-13. Partial Divide By Five | II-43 |
| II-4-14. Slide | II-45 |
| II-4-15. Jump Conditional | II-47 |
| II-4-16. Delay | II-49 |
| II-4-17. Write | II-53 |
| II-4-18. Right Shift | II-54 |
| II-4-19. Add Conditional | II-55 |
| II-4-20. Extract - Clear | II-58 |
| II-5. LOGIC DIAGRAMS | II-59 |

THEORY OF OPERATION

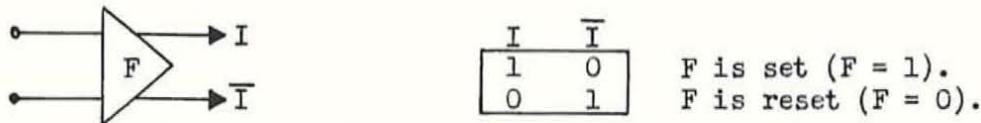
SECTION II

THEORY OF OPERATION

II-1. TIMING

The basic timing in the Monrobot XII is controlled by two tracks on the storage drum -- the clock track and the sector track -- and by the state of the F8 flip-flop. These timing relationships are shown in figure II-1.

It should be noted that a flip-flop has two outputs -- one identified by a "real" label (I) and the other by a "primed" label (\bar{I}). The system operates on two binary logic levels -- 0 volts or "high" (corresponding to a binary 1) and -6 volts or "low" (corresponding to a binary 0). When the real output of a flip-flop is high, the flip-flop is spoken of as "set" or "on"; in this state, the primed output is low. When the primed output is high (and thus the real output low), the flip-flop is "reset" or "off". These conditions are summarized below:



In addition, the time during which a flip-flop is in a given state may frequently be called by the name of that flip-flop and that state. For example, F8-time means the time during which F8 is set, while $\bar{F}8$ -time indicates the time during which F8 is reset.

II-1-1. The Clock Track.

The clock track contains only a sequence of 1's permanently recorded at the factory. There are 2775 of these pulses on the track. The time between each of them defines the basic bit-time for the system. The bit pulse, properly shaped, is taken from the clock track playback.

II-1-2. The Sector Track.

The information on the sector track is used to define the system's word-time. Each word on this track contains thirty 0's and seven 1's. The playback is then modified into the T29 pulse, which lasts for the duration of the seven 1's. The T29 pulse is used to define the T36 pulse, which is the sector or word pulse. The latter occurs once every 37 bit pulses. It is the interval between these sector pulses that defines the basic word-time of the computer. There are 75 sectors in a drum track and thus 75 such word-times during a drum revolution.

The 37 bit-times of a word are denoted t_0, t_1, \dots, t_{36} , where t_0 is the earliest bit-time and t_{36} is the last. The sector pulse occurs at t_{36} time.

II-1-3. The F8 Flip-Flop.

The F8 flip-flop changes state with each word-time, unless instructed to do otherwise by the program. F8 is thus alternately high and low for each succeeding word-time.

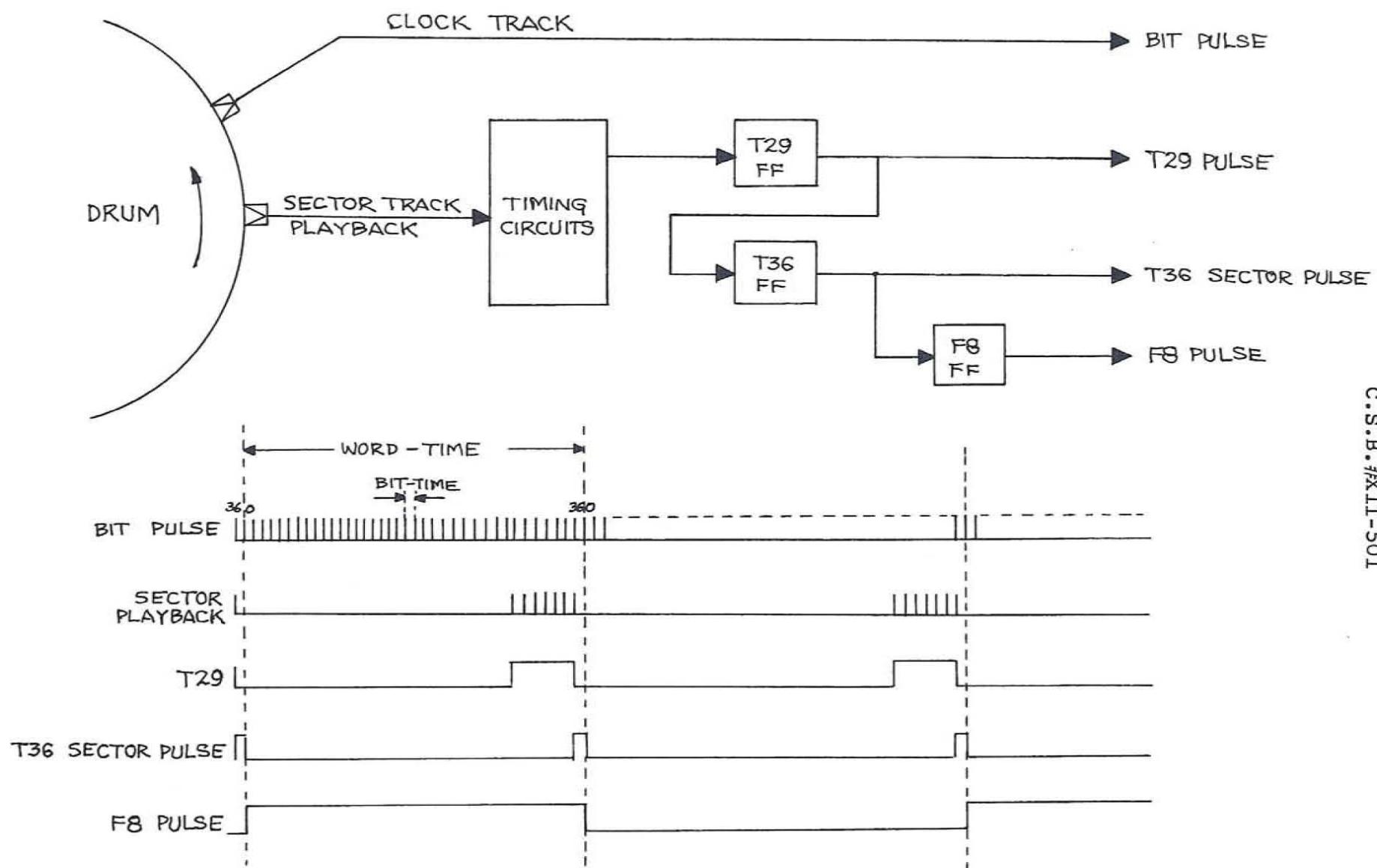


Figure II-1. Monrobot XII Basic Timing Diagram

II-1-3-a. The Precess-Execute Sequence. The essence of the computer's operation can be best understood if the word-times are considered as occurring in pairs. The two word-times in each pair are defined by the conditions F8 and F8, described above. The information and instruction words which are actually being used at any given time are located in a fast-access channel called the Accumulator and Orders Channel, which is described in chapter II-2-2. During the first word-time of each pair, a new command code is brought into a Command Register from the Orders Channel, while during the second word-time, the command is carried out on the contents of the Accumulator. These two paired word-times are called precess times and execute times, respectively. All operations on data are carried out during execute times, while precess times are used to set up the computer for a next operation.

The flip-flop F8 is set during precess times and reset during execute times, producing the F8 pulse sequence shown in figure II-1. Thus, F8 defines a precess word-time, while F8 defines an execute word-time.

Since there is an odd number of registers or sectors on each drum track (75) while the precess-execute times occur in pairs, two drum revolutions are required to bring any sector into the same position again during an execute time. A special Leap command is provided, however, by which it is possible to view a register again in the following drum revolution.

II-2. MEMORY

II-2-1. General Storage.

The General Storage in the Monrobot XII consists of five tracks on the magnetic drum. The computer's active General Storage occupies four tracks and thus has a capacity of 4×75 or 300 words -- or 11,100 bits. The playback and record for each track is located eight words (four precess and four execute) apart, as shown in figure II-2. The latter is an overall data flow diagram of the entire computer.

It can be seen from the figure that the eight word spacing combined with the direction of rotation of the drum allows the computer to playback a storage sector or register into fast access, perform three operations on the data during the three execute word-times between and then record the information back into the same General Storage register during the fourth execute cycle.

The information contained in General Storage can be played back an indefinite number of times; when the power is shut off, it is preserved. However, the recording of any new information into any sector destroys the previous contents of that sector.

The 37 bits in each General Storage register are denoted $g_0, g_1, g_2, \dots, g_{36}$, with bit g_0 arriving at the record head at time t_0 , g_1 arriving at time t_1 , and so on. In short, g_0 corresponds to t_0 , etc. The flag bits g_0 have functions in the initial loading of the program into General Storage, in the execution of the Slide command, and in defining the beginning of the program in the start-up of the machine. When loading, a 0 is recorded in one flag-bit position on each track and 1's in all the rest. During system operation, however, nothing can be recorded in the flag-bit positions.

II-2-2. The Accumulator and Orders Channel.

The Accumulator and Orders Channel is a fast access channel. Its playback and record is spaced two words apart on the drum in such a manner that information from the playback head can be recorded again at the record head, forming a circulating loop, as shown in figure II-2. This process can be repeated indefinitely. Points A and B in the figure are the reference points in this loop. Between these two points, any bit of information is delayed two word-times, or 74 bit-times, before being recorded back on the drum. Thus, the Accumulator and Orders Channel contains 74 bits of information -- 72 on the drum, 73 including L2 and the full 74 to point B. Information recorded at L1 (point A) is available 74 bit-times later at point B (the output of the flip-flop F3).

The Accumulator is that part of the two-word-time loop which arrives at point B during an execute time (F8), while the part of the loop which arrives at the reference point during precess time (F8) is the Orders Channel. Thus, in succeeding word-times, the Accumulator and the Orders Channel are alternately available at the playback head.

Both the Accumulator and the Orders Channel have capacities of 37 bits

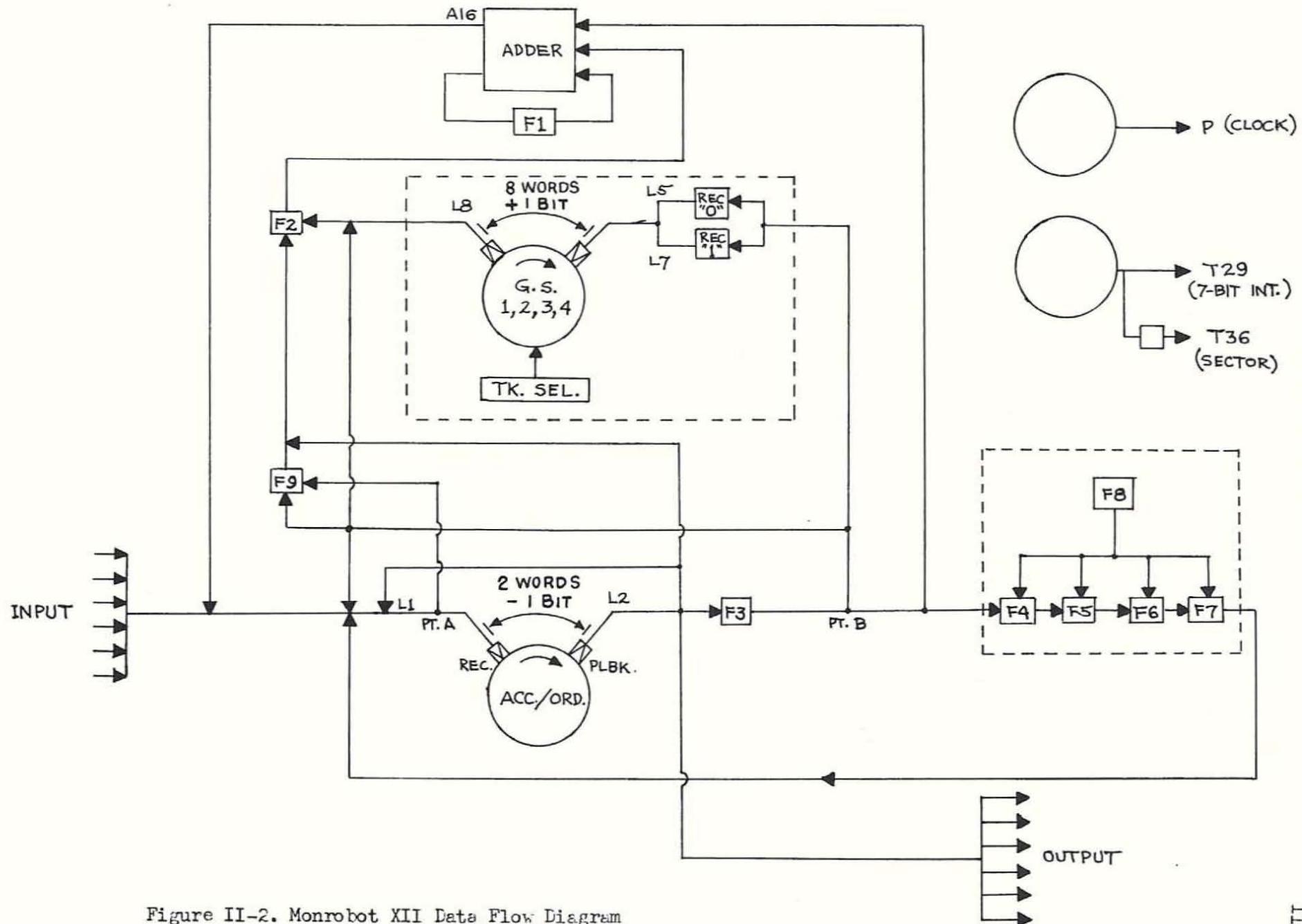


Figure II-2. Monrobot XII Data Flow Diagram

each. The 37 bits in the Accumulator are denoted $a_0, a_1, a_2, \dots, a_{36}$, with a_0 arriving at the reference point at t_0 , a_1 arriving at t_1 , and so on.

The 37 bits of the Orders Channel are denoted $c_0, c_1, c_2, \dots, c_{36}$. Again, c_0 arrives at the reference point at t_0 , and so on.

II-2-3. The Condition Register K.

The K Register is physically a flip-flop (F1) and thus can store only one bit of information. In effect, it carries information forward from one execute word-time to the next one. There are a number of program commands used in the computer whose executions are affected by this "remembered" state of F1. These are referred to as "conditional" commands -- that is, they are conditional on K.

II-3. CONTROL

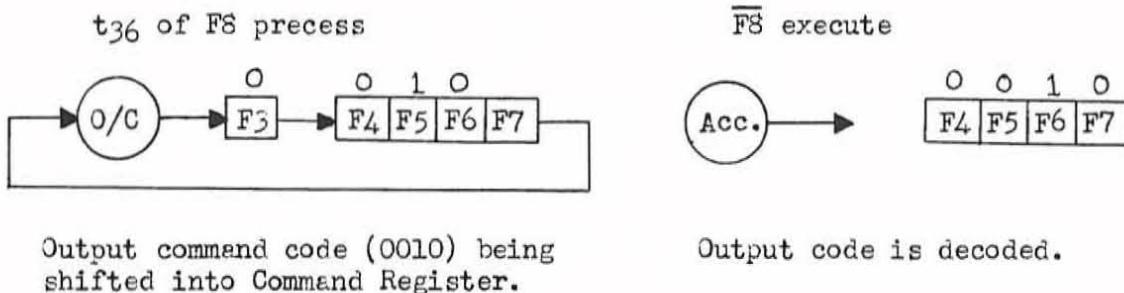
II-3-1. Orders Precession.

The program operation which is actually carried out at any given time is dictated by the command code which is in the Command Register at that time. This register consists of four flip-flops -- F4, F5, F6 and F7. During precess time, when the Orders Channel is being processed, the Command Register becomes part of the circulating loop. Thus, four additional bits are inserted into the loop. The result is that the information in the Orders Channel is shifted four places up toward the high-order end such that the information from the four high-order positions (c_{36} , c_{35} , c_{34} and c_{33}) is left in the Command Register when the precess time ends (see figure II-4). This code tetrad or group of four bits remains in the Command Register and acts as the instruction during the next execute time. The operation determined by this instruction is carried out on the contents of the Accumulator, which is always preserved during orders-precess time.

In effect, the Orders Channel is a buffer for the transfer of instructions from General Storage into the Command Register.

The instructions remaining in the Orders Channel are thus shifted up one tetrad each time. When each four-place shift occurs, the four vacated lowest positions of the Orders Channel (c_3 , c_2 , c_1 and c_0) become filled with whatever may have been left in the Command Register at the end of the previous execute time.

As stated above, normal decoding of the Command Register occurs during \bar{F}_8 execute periods. Note, however, as illustrated in the diagram below, that the instruction code is also available for decoding in flip-flops F3 through F6 during t_{36} of the precess word just before it is shifted into the Command Register.



Several commands are, in fact, decoded in this manner. Wherever this technique is used, it is called early decoding.

II-3-2. Order Word Transfer.

As an order word is precessed through the Orders Channel and the Command Register, the code tetrads in the word will eventually be used up. A new word must then be brought into the Orders Channel from General Storage. The standard method by which this is done is by the use of a Jump command. The Jump command can be programmed as one of the nine command codes in an instruction word or it can be supplied automatically as a tenth command by the hardware.

Consider the sequence of events when a Jump command is precessed into the Command Register. This sequence is shown diagrammatically in figure II-3.

(1). F8-time.

The Jump code is precessed into the Command Register.

(2). $\overline{F8}$ -time.

The Jump command is not executed during this period. However, at the end of this execute time, logical circuitry resets the Jump code in the Command Register to 0000, and sets a special memory.

(3). F8-time.

This is a special non-precess F8-time, as defined by the special memory which was set in the last execute time. It is during this F8-time that the new instruction word is brought into the Orders Channel from General Storage. When a new word is transferred from General Storage, only the 36 bits from g_1 through g_{36} go into the Orders Channel; the flag bit g_0 is not transferred. However, when the transfer is effected, a 1 is automatically recorded in the c_0 bit position in the Orders Channel.

(4). $\overline{F8}$ -time.

Since the Jump code originally present was reset to 0000 at the end of the previous execute time, the Command Register now contains a Skip code. Thus, nothing is done during this execute time.

(5). F8-time.

The normal precession of command codes now resumes during this period, with the precession into the Command Register of the first (high-order) code tetrad of the new word.

As the precession of the new word tetrads proceeds, the 0000 code in the Command Register follows the low-order 1's-bit, which was recorded into the c_0 bit position when the word was transferred, through the Orders Channel toward the high-order end in four-bit shifts, as shown in figure II-4. The tenth precession then brings into the Command Register the low-order 1's-bit followed by three 0's of the converted Skip code, as shown in the figure. In this way, a 1000 code -- which is a Jump code -- is automatically brought into the Command Register at the end of a word, causing a new word to be transferred from General Storage. There are, in effect, therefore, ten tetrads available in each word, the last one always being a Jump command.

II-3-3. Track Selection.

A four-state ring counter monitors the General Storage tracks and determines which track is selected. At any given moment, only the heads of the selected track are connected to the rest of the computer. The other tracks remain disconnected until a Change-Track command changes the state of the track selection counter. The sequence of track changes in this counter progresses cyclically from 1 to 4, each Change-Track command adding 1 to the track number. If the machine is on track 4, a Change-Track command moves it to track 1. Thus, for example, three Change-Track instructions are necessary to move from track 3

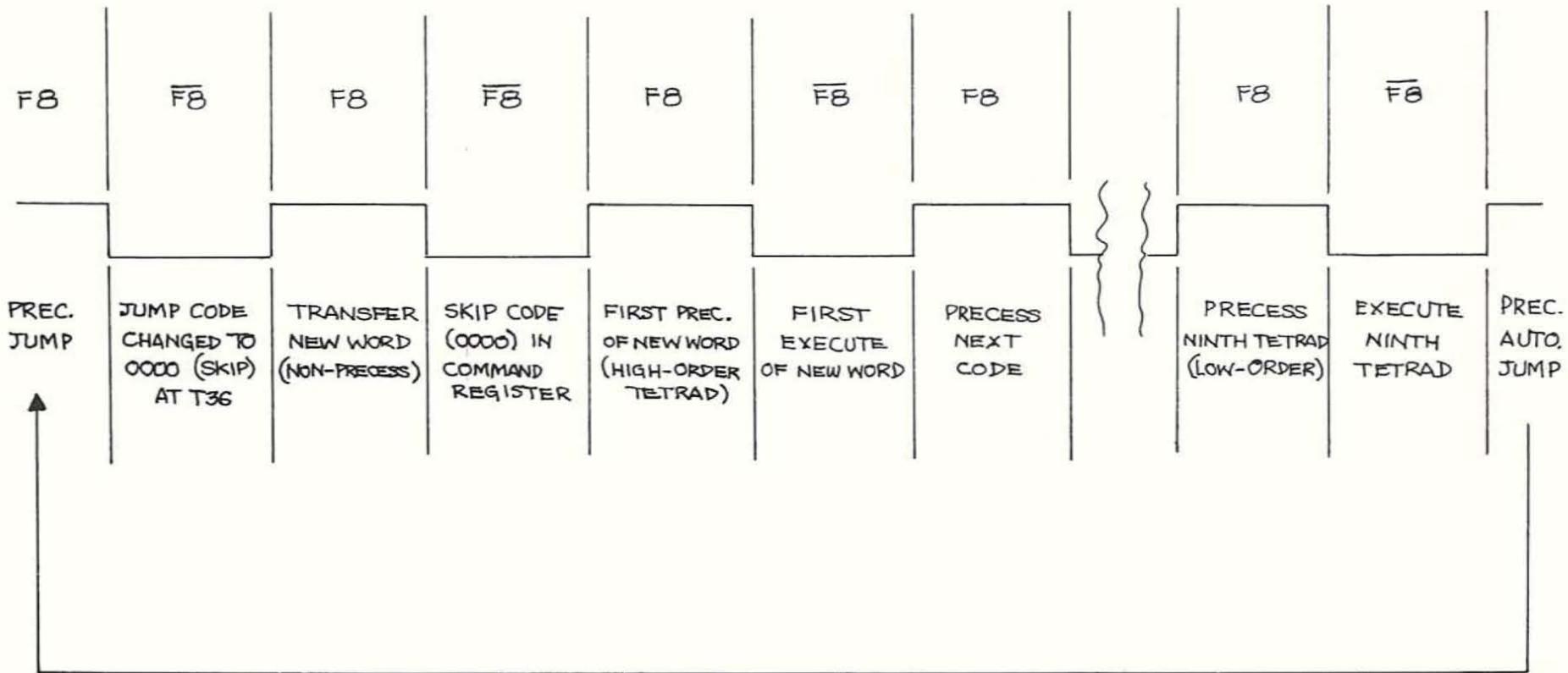


Figure II-3. Order Word Transfer and Precession Sequence

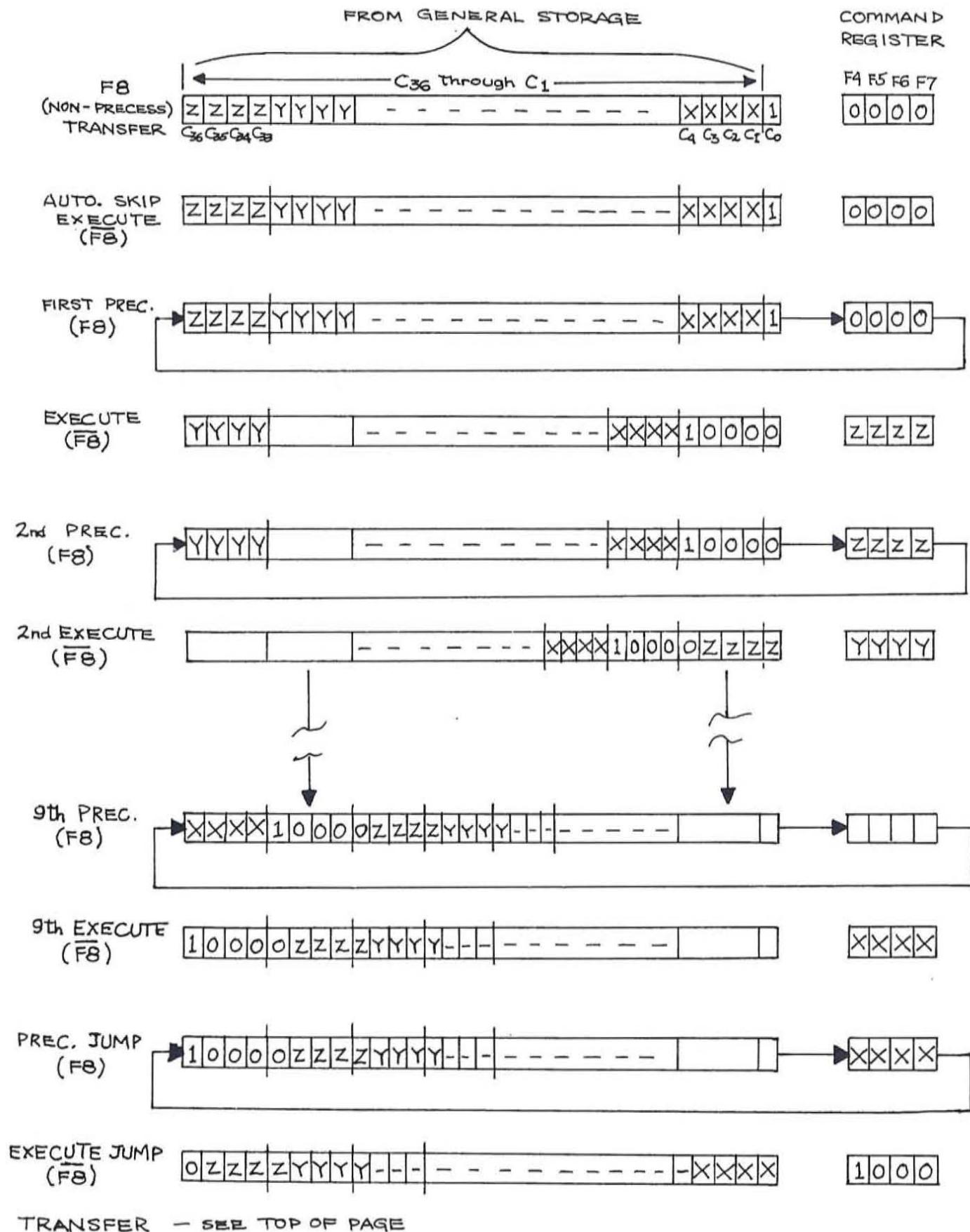


Figure II-4. Code Precession and Word Transfer Sequence

to track 2. When power is first turned on to start the computer, the track selection is forced to return to track 1 and the system starts its operations there.

Instructions which involve General Storage operate on the track determined by the track selected and on the track sector which happens to be under the playback or record head at that time. It must be remembered that no addresses are included with the micro-instructions. It is up to the programmer to arrange the sequence of instructions such that they are executed when the desired information is accessible from General Storage.

II-3-3-a. Alternate Track Selection. It will be recalled that there are five tracks in General Storage, one of which is used as an alternate to one of the other four. The alternate track (2B) is used for special programs. The PGM (Program) switch on the control panel is used to choose this alternate track. Normally, the Program switch is off; in this state, track 2A is in use. Depression of the PGM switch disconnects track 2A and introduces the alternate track 2B in its place.

The real track selection sequence is therefore either 1, 2A, 3, 4 or 1, 2B, 3, 4.

II-3-4. On/Off Control.

Once operational, the Monrobot XII is continually running and is under the control of the program. When the machine is first turned on, however, or when it is being shut down or restarted, its operational requirements (drum speed, etc.) are not up to their correct levels. During these times, the system is under the control of an on/off sequencing circuit and the control switches.

II-3-4-a. Control Switches. There are three control switches on the computer control panel -- PWR (Power), RST (Restart) and PGM (Program). The purpose of the Power switch is simply to turn the computer on. The Restart switch allows the operator to go back to the start of the program if this should become desirable or necessary, without first turning the computer off. The Program switch provides access to the alternate storage track.

II-3-4-b. Sequencing. The on/off sequencing circuit in the system assumes control of the system during start-up, shut-down and restart; it also provides protection to the system during normal operation in the event of a line failure.

Start-up

Depression of the PWR (Power) switch at turn-on connects AC line voltage to the system, energizing the drum motor, the fan, the input-output power supplies, the input-output motors and the main power supply. It also causes the on/off control circuit output signals $\overline{W4}$ and $\overline{W6}$ to go high. $\overline{W4}$ high blocks General Storage record and forces the logic flip-flops F2 and F3 into the reset state; $\overline{W6}$ high blocks the Accumulator and Orders Channel record circuit. While $\overline{W4}$ is high, $\overline{W4}$ is low; the latter condition pulls the track selection circuit to track 1.

When the drum has reached operating speed, the P1 clock signal, which is used in the circuit as an indication of drum speed, acts internally in the circuit to turn off W6. The Accumulator and Orders Channel is then no longer blocked. However, since F3 is still held low, nothing but zeros can be recorded in the channel. Thus, after two word-times, the Accumulator and Orders Channel will be cleared.

Now the sequencing circuit must find the program start-word, which is indicated by the single 0 flag bit on the track. Gate N41 in the set logic of F4 looks for the 0 flag bit in t_{36} of an F8-time. When it finds it, F4 is set, defining a Jump command; F4 remains set for the execute time which follows and then goes low again. This occurs in every second drum revolution. However, in order to bring in the start-word, F2 must be set via N21 to define the special non-precess F8-time of the Jump command sequence, but this cannot be done while W4 is high. Every T29, a W4 circuit samples the Accumulator to determine if it has been cleared and looks for F4 high (indicating a 0 flag bit). When it finds that both of these conditions exist at the same time, $\overline{W4}$ goes low. Since T29 does not include t_{36} therefore T9 goes low before t_{36} and thus, at t_{36} , F2 can be set. This condition defines the transfer period of a Jump command and the start-word is brought into the Orders Channel.

When $\overline{W4}$ goes low, W4 goes high, the RST light goes on (indicating that the machine is ready for operation) and the input-output circuits are freed.

Figure II-5 is a timing diagram of the principal sequencing events.

Shut-down

The sequence of events which takes place when the computer is shut down is essentially the inverse of the start-up events. Almost immediately when power is removed, $\overline{W4}$ and W6 both go high again. Thus, General Storage record, the Accumulator and Orders Channel record and the input-output devices are all blocked. It should be noted, however, that $\overline{W4}$ will only be pulled high during an F8-time, in order to prevent possible interruption of a Write command into General Storage during an execute word.

Restart

The restart sequence of events is basically the same as that of an initial start. The Accumulator and Orders Channel and General Storage record circuits and the input-output devices are all blocked. The system is pulled again to track 1 and the sequencing circuit must seek the program start-word again. When the required conditions have been fulfilled -- although there is usually no loss of drum speed in the case of a restart -- W6 again goes low and then $\overline{W4}$, turning on the RST light and freeing the input-output devices.

Program Change

Operation of the Program switch also creates the sequencing circuit blocking conditions -- W6 high and $\overline{W4}$ high. In the Program-switch case, however, the system stalls in this condition and the Restart switch must then be depressed in order for the sequencing cycle to start.

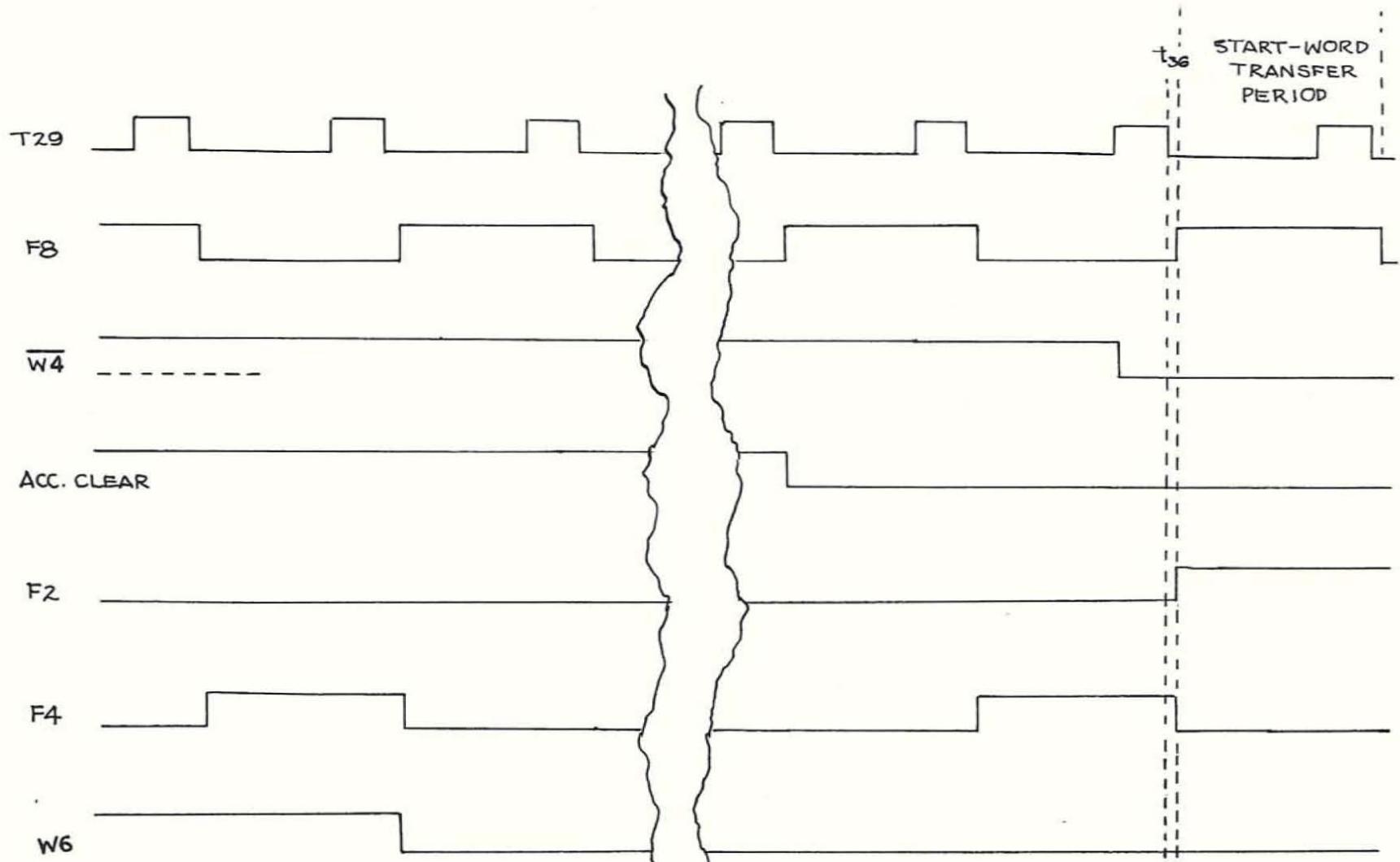


Figure II-5. Sequencing Circuit Timing Diagram

System Protection

In the event of a line failure, or a loss of clock speed, the above sequence again takes place: W6 and $\overline{W4}$ both go high. Assuming that the cause of the malfunction is removed, the system can be placed back in operation by depressing the Restart switch.

II-4. THE PRIMARY COMMANDS

The principles of the Monrobot XII are implemented by a number of coded programming commands. Sixteen binary operation codes are utilized for this purpose; by adding the state of the K Register and various code-track combinations, the following primary operation commands result:

| Command | Code | General Function |
|------------------------------|----------------------|--|
| Skip | 0000 | Does nothing, all information unchanged. |
| Quadruple Left Shift | 0001 (Track 1, 3) | Shifts the Accumulator contents four bits to the left. |
| Leap | 0001 (Track 2, 4) | "Redefines" Accumulator and Orders Channel and interchanges precess and execute times. |
| Input | 0010 (Track 1, 3) | Puts information into computer. |
| Output | 0010 (Track 2, 4) | Takes information out of computer. |
| Complement | 0011 | Replaces Accumulator number with its binary 2's complement. |
| Change Track | 0100 | Advances track counter 1 position. |
| Left Shift | 0101 | Shifts Accumulator contents one bit to the left. |
| Transfer | 0110 | Records General Storage information into Accumulator. |
| Add | 0111 | Adds General Storage information to Accumulator. |
| Jump | 1000 | Moves new instruction word from storage to Orders Channel. |
| Times Five | 1001 (Track 1) | Multiplies Accumulator contents by five. |
| Partial Divide by Five | 1001 (Track 3) | Performs initial step in dividing the Accumulator contents by five. |
| Slide | 1001 (Track 2, 4) | Stops Orders precession, does nothing until a 0-flag bit is found. |
| Jump Conditional (Branch) | 1010 | Same as Jump <u>if K equals 1</u> . |
| Delay | 1011 | Maintains status quo for a delay period, followed by a Jump Conditional |
| Write | 1100 | Records Accumulator contents ($a_1 - a_{36}$) into General Storage. |
| Right Shift | 1101 | Shifts Accumulator information one bit to the right. |
| Add Conditional | 1110 | Adds General Storage to Accumulator contents, if K equals 1. |
| Extract | 1111 (K = 0) | Logically multiplies General Storage by Accumulator. |
| Clear | 1111 (K = 1) | Clears Accumulator. |

The logic by which these commands are carried out in the computer will now be described in detail. Since they are concerned with the execution of the commands, the logical sequences presented describe execute times. The general sequence of a process time is described in the preceding chapter.

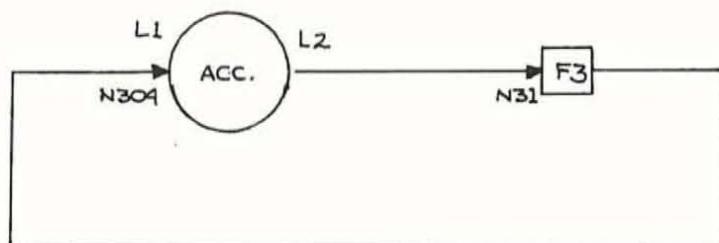
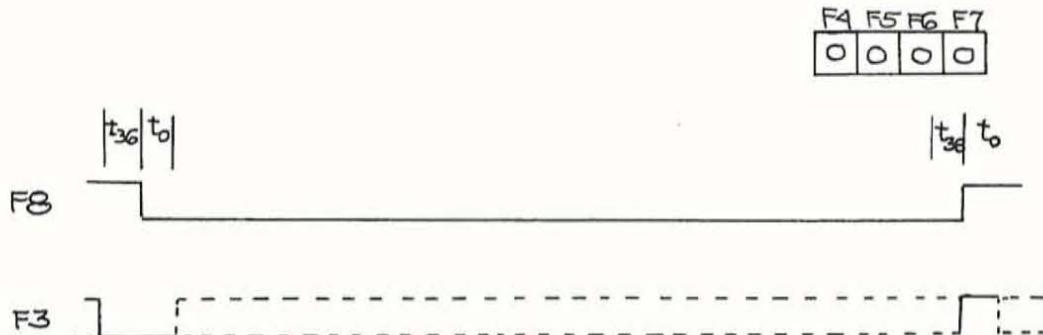
II-4-1. Skip (0000).

The Skip command is a "do-nothing" command; it allows the Accumulator word to regenerate. In the execution of the Skip, K is preserved and a_0 is lost.

SEQUENCE:

F8 (Skip-execute).

- t_0 (1). F3 is low because there was no set logic during the previous bit time. Therefore, record 0 in the Accumulator via N304, L1.
- $t_1 - t_{36}$ (1). Since the Command Register, F4 - F7, is low throughout the execute time, the Accumulator, $a_1 - a_{36}$, is regenerated via N31, F3, N304, L1.



II-4-2. Quadruple Left Shift (0001, Track 1 or Track 3).

The purpose of this command is to shift the contents of the Accumulator $a_1 - a_{36}$ four bit positions toward the high-order end. The a_0 is set to 1, $a_1 - a_4$ are reset to zero and the four high-order bits a_{37} to a_{36} are lost. General Storage and K remain unchanged.

The Quadruple Left Shift (QLS) command is decoded early -- that is, at t_{36} of F8. As a consequence, the F3 flip-flop is not allowed to reset as it normally would. Thus, the word-time which follows is defined as a precess time. Note, however, that the information being precessed in this F8 time is the contents of the Accumulator. The QLS command is therefore implemented by precessing the information in the Accumulator via the Command Register as though it were an Order word.

Memory must be provided to ensure that this special Accumulator precess time is succeeded by a regular Orders Channel precess time in order to preserve the normal precess-execute sequence. To do this, F8 must again be prevented from resetting. This is done by setting flip-flop F9 when the early decoding of the QLS occurs and preserving its state until the end of the QLS execute period.

SEQUENCE:

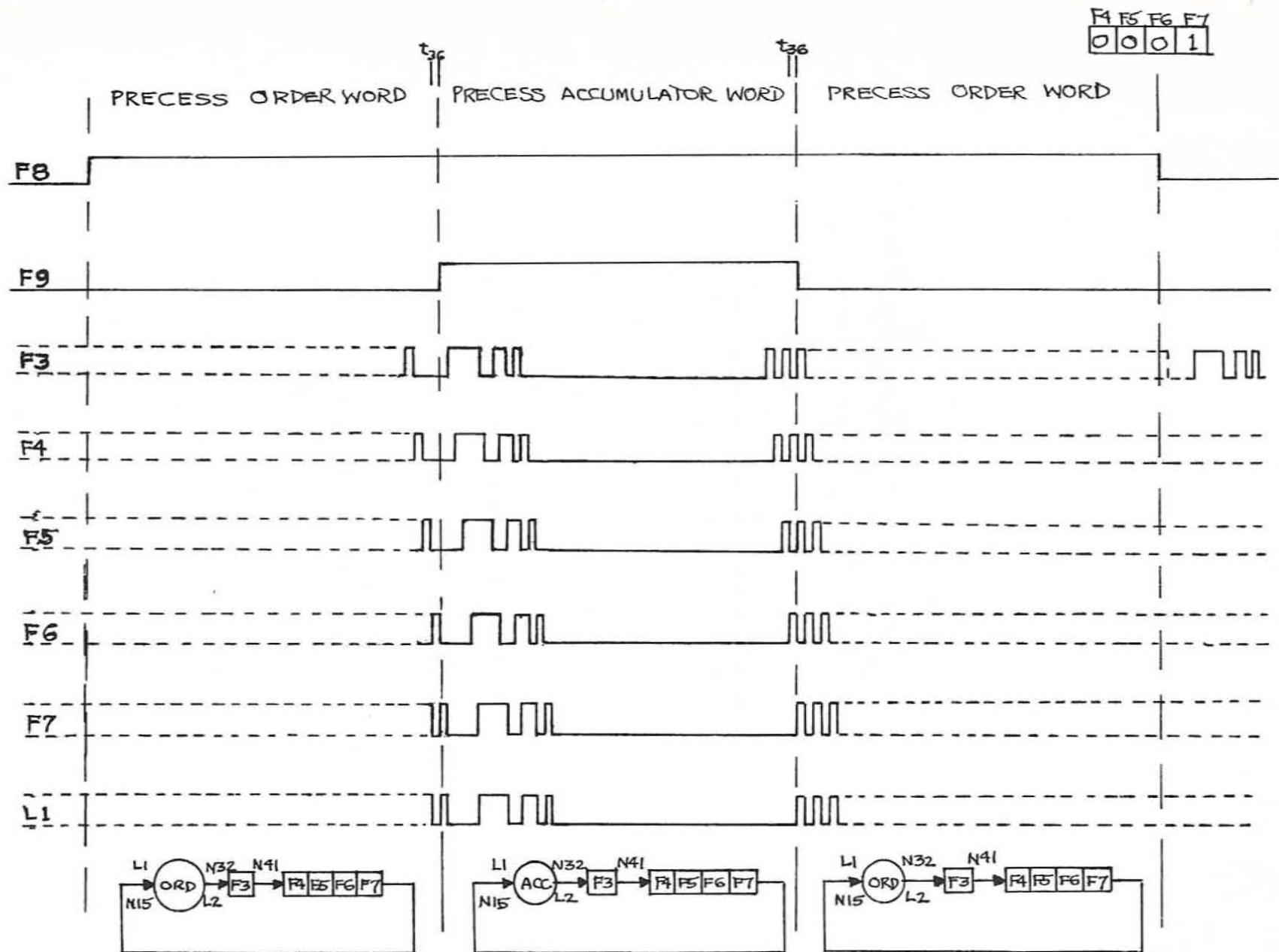
F8 (precess Order word).

- t_{36} (1). The QLS code is shifted into the Command Register.
 (2). It is decoded early from F3 - F6 by N82 to inhibit the reset of F3 and by N93 to set F9.
 (3). At this time, no set trigger for F3 is generated.

F8 (precess Accumulator).

- $t_0 - t_{36}$ (1). As a result of the above, $F3 = 0$ at t_0 and the QLS code is in the Command Register. The Accumulator information now flows through the Command Register flip-flops F4 - F7 in the normal precess manner is recorded back into the Accumulator. The complete path followed in this case is: L2, A26, N32, F3, F4, F5, F6, F7, N151, A15, L1. The flow is such that the QLS code in the Command Register precedes both the 0 in F3 and the Accumulator information in the order of bits to be recorded back in the Accumulator. The QLS code is recorded in $a_0 - a_3$ and the 0 in F3 is recorded in a_4 and the Accumulator information, delayed four bit-times, is recorded in $a_5 - a_{36}$ -- with the exception of its high-order tetrad, which is left in the Command Register.

- t_{36} (1). If L2 -- the low-order bit c_0 of the Order word seen one bit early -- and A19 are both high, N32 will be high to set F3. A19 decodes F4.F5; it is high if a_{34} (stored in F5) and a_{35} (stored in F4) are 0 and 1 respectively. These bits are part of the Accumulator



high-order tetrad left in the Command Register at the end of the end of the word-time because of the four-bit shifting process.

- (2). Because F9 equals 1, A17, and therefore N82, is low and F8 is again not permitted to reset.
- (3). N94 is high, resetting F9. Thus, the following word-time is a regular process (F8) time during which a new order code is shifted into the Command Register.

THEORY OF OPERATION

II-4-3. Leap (0001, Track 2 or Track 4).

The Leap command "redefines" the Accumulator and Orders Channel -- what had been the Accumulator information prior to the Leap becomes an Order word, and the Order word which produced the Leap code in the first place becomes the new Accumulator data.

As a result of the Leap, a Jump code is automatically inserted in the low-order tetrad position ($c_1 - c_4$) of the new Order word. The new Accumulator contains the remnants of the programmed portion of the old Order word, plus any debris which may have been inherited from the Command Register during precess times prior to the Leap operation.

The Leap command permits the programmer to "see" a given register again in the following drum revolution. Ordinarily, since each drum-track has an odd number of word registers (75) while the precess-execute times form an even pair, two drum revolutions are required before a particular register appears again in the same place.

The Leap command is decoded early -- that is, at t_{36} of F8. As a consequence, F8 is not allowed to reset as it normally would. Thus, the next word-time becomes a precess again. This precess time acts on information which had been Accumulator information, redefining it as Order word information. The four high-order bits are precessed into the Command Register to become the next Order code. It is during this precess operation that the low-order end of the new Order word automatically inherits a Jump code.

At the end of this second precess time, F3 is allowed to reset. Thus, the next word-time becomes, by definition, an execute-time (F3) during which the new Order code is executed. The information available at the fast access playback during this execute period is the new Accumulator word.

In this manner, the sequences of precess-times and execute-times have been interchanged.

SEQUENCE:

F3 (precess).

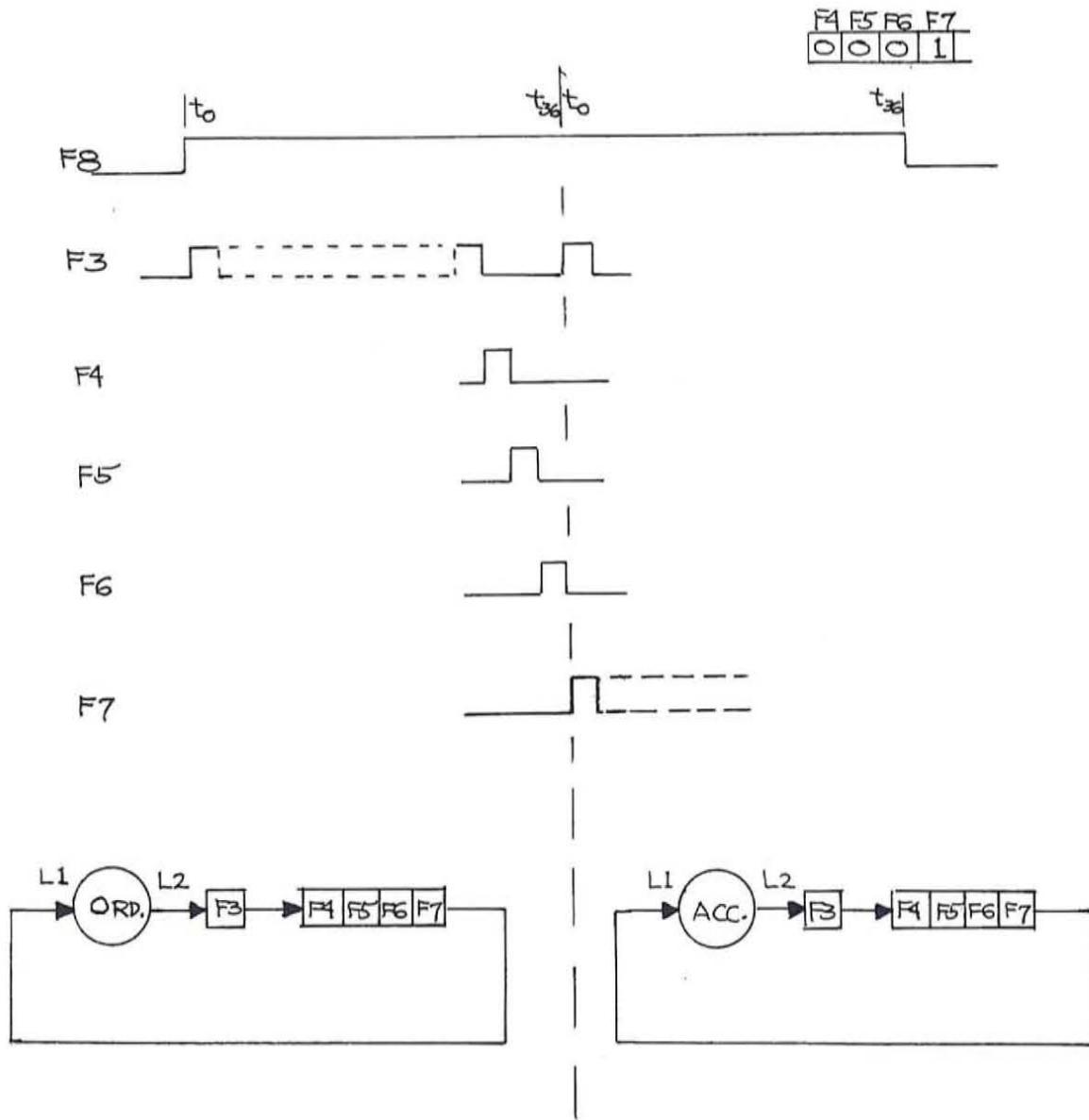
- t_{36} (1). The Leap code is precessed into the Command Register.
 (2). The command is decoded in F3 - F6 by N32 to inhibit the resetting of F3 and by N35 to set F3.

F3 (precess).

- $t_0 - t_{36}$ (1). It can be seen that, at t_0 , F3 is high and the Leap code is in the Command Register. The new Order word information now flows through the Command Register flip-flops F4 - F7 in the normal precess manner and is recorded back into the fast access channel. The flow is such that the code in the Command Register precedes both the 1 in F3 and the new Order word data in the order of bits to be

recorded. The Leap code is recorded in $c_0 - c_3$ and the 1 in F3 is recorded in c_4 of the new Orders Channel. The new Order word is recorded in $c_5 - c_{36}$ -- with the exception of its high-order tetrad, which is shifted into the Command register to become the next Order code.

t_{36} (1). F8 is allowed to reset.



II-4-4. Input (0010, Track 1 or Track 3).

The Input command controls the input of information to the system. The computer will execute a successful input only if the selected input device(s) has a character for the computer at t_1 of the Input execute word. General Storage remains unchanged, but K is set to 1 at t_3 if the Input instruction is successful.

Input devices normally send their coded information to the computer in parallel form. Thus, each incoming coded character must pass through a parallel-to-serial converter in the computer before it can be recorded in the Accumulator. To accomplish this, a bit-time counter generates eight bit-time pulses $t_0 - t_7$ (beginning at t_0 of the Input execute word). These are used in the converter logic to scan the input code wires. The outputs of the sequentially sampled code wires are recorded in $a_0 - a_7$.

The K flip-flop F1 is an integral part of the bit-time counter. Since it defines the last four time pulses, $t_4 - t_7$, it is set to 1 with the t_3 pulse generated by the counter.

In a successful Input, the read-in process is completed at the end of the t_7 pulse. At this point, the selected input device is signaled that the computer has accepted the character. Simultaneously, the Command Register, which has thus far contained the Input instruction code, is reset to a Skip code (0000) by the t_7 pulse. Starting with t_8 , the computer then executes a Skip operation for the balance of that word-time.

In every Input-execute, the computer samples the selected input source at t_1 for a "busy" signal (character not ready). If the device is busy at that time, the Command Register is reset to a Skip code, which terminates the read-in operation. Since F1 decodes the counter and the counter ceases to operate after t_1 , K cannot be set and the character-acceptance-signal is not sent to the input device. This operation constitutes an unsuccessful Input-execute.

During the read-in process, a parity check is made on the input character code in the following way. The sequential output of the parallel-to-serial converter is gated to a flip-flop (F9) in such a manner that the flip-flop changes state whenever a 1 appears in the character code. When it enters the read-in process, it is in the reset (0) state; thus, if the character contains an odd number of 1's, it will end up in the set (1) state at t_7 when the scanning is terminated. The state of this flip-flop is then preserved until the end of the word and recorded in a_{36} .

The Accumulator information bits will be modified in each case such that the following conditions prevail:

Successful Input:

- $a_0 - a_7$Contain Input character code.
- a_80.
- $a_9 - a_{36}$Regenerated.
- a_{36}1 if parity flip-flop is set.

Unsuccessful Input:

- $a_0 - a_1$Contain meaningless information.
- a_21.
- $a_3 - a_{36}$Regenerated.
- a_{36}1 if the parity flip-flop is set.

From the above considerations, it should be obvious that, if a 1-bit had existed in a_{36} just prior to the Input-execute, it would be regenerated and would mask the parity check. Thus a programming rule requires the clearing of a_{36} before attempting an Input-execute when a valid parity check is desired.

Selection of the desired input device is not a function of the Input operation. Instead, a "device select" code is sent to a special device memory. Since this is done only by the Output command, an Output must be executed for this purpose before an Input command in which a change of input devices is desired (see "Output").

Ordinarily, only one input device should be addressed at a time. The computer's hardware allows use of all of the inputs at once, but if more than one device had a character ready their codes would superimpose themselves on each other and the result would be meaningless. It is possible that this use of all the inputs may be done in special cases, but it is not normal procedure.

SEQUENCE:

F8 (precess).

- t_{36} (1). The Input instruction is decoded at F3 - F6, resetting F1 via N18.

 $\overline{F8}$ (Successful Input-execute).

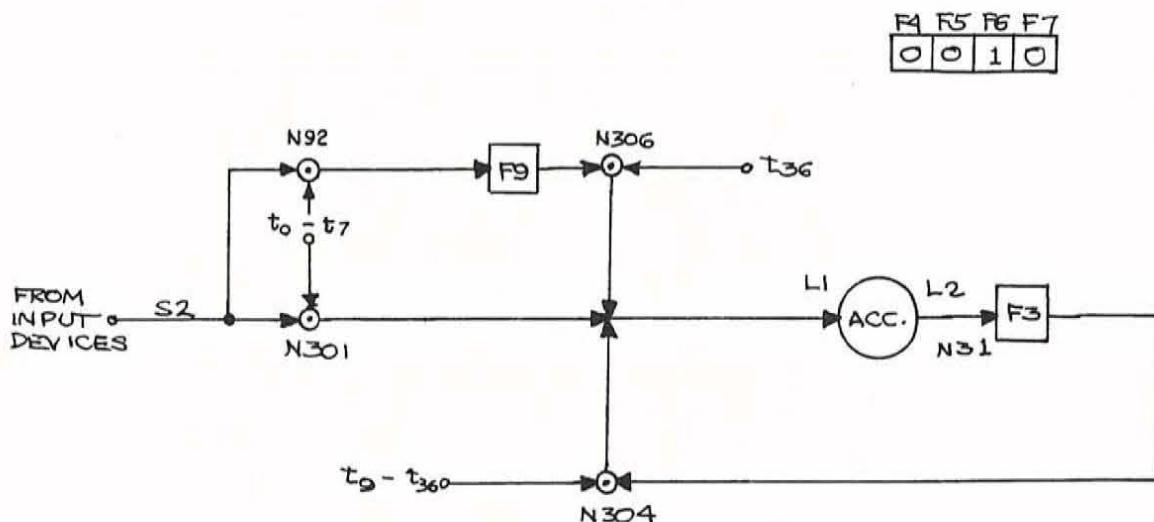
- $t_0 - t_7$ (1). The 8-state bit counter (F1, F2, F3) is started. The code wires from the selected input device are sampled and read sequentially via the parallel-to-serial converter logic net S2 into the Accumulator record net L1 and via N92 and N95 into the parity flip-flop F9. The 8 states of the F1, F2, F3 counter are decoded by the 8 gates N421 - N428 of the converter logic net S2. Each of these gates then samples an assigned code wire from the input device during a specific counter state. Code wires from any unselected input devices automatically appear low.
- t_7 (1). At this time, A10 is high by N103, resetting F6 in the Command Register and changing the operation code to a Skip.
- (2). Signal S0 is high. It will set S1 at the end of this bit-time.
- t_8 (1). The signal S1 goes high, causing the selected input device to clear. S1 remains high until t_{29} .
- $t_8 - t_{36}$ (1). The Skip operation is decoded and the Accumulator regenerated via F3, N304, L1. Because F3 is low at t_8 as a result of its previous function in the counter, a zero is automatically recorded in a36.
- t_{29} (1). S1 is reset by gated supply VG.
- t_{36} (1). If the parity flip-flop F9 is high, a 1 is recorded in a36 via N306, L1. Note that the Accumulator is also being regenerated via N304, L1, and may already contain a 1 in a36.

(2). F9 is reset via N94.

(3). F3 is set via N31.

F8 (Unsuccessful Input-execute).

- $t_0 - t_1$ (1). The bit counter (F1, F2, F3) is started and the first two wires from the selected input device are sampled, the data being read sequentially into the Accumulator record net L1 and the parity flip-flop F9 via the converter logic net S2.
- t_1 (1). Since S4 is high when the selected input device is busy (that is, when it does not have a code ready for the computer -- or it is not plugged in), F6 is reset and the operation code is changed to a Skip.
- $t_2 - t_{36}$ (1). The Accumulator is regenerated via F3. The latter is high at t_2 as a result of its previous function in the counter, and therefore a 1 is automatically recorded in a_2 .
- t_{36} (1). If the parity flip-flop F9 is high, a 1 is recorded in a_{36} (although parity check is meaningless in this case), and F9 is reset.



MULTIPLE INPUT:

Input Common.

Selection of input devices is achieved by making the Input Common wire of the desired device high. The Input Device Memory circuits (ID1, ID2, ID3) are tied to the Common wires through inverter circuits (IB), hence the Device Memory circuit with a low output corresponds to the selected device.

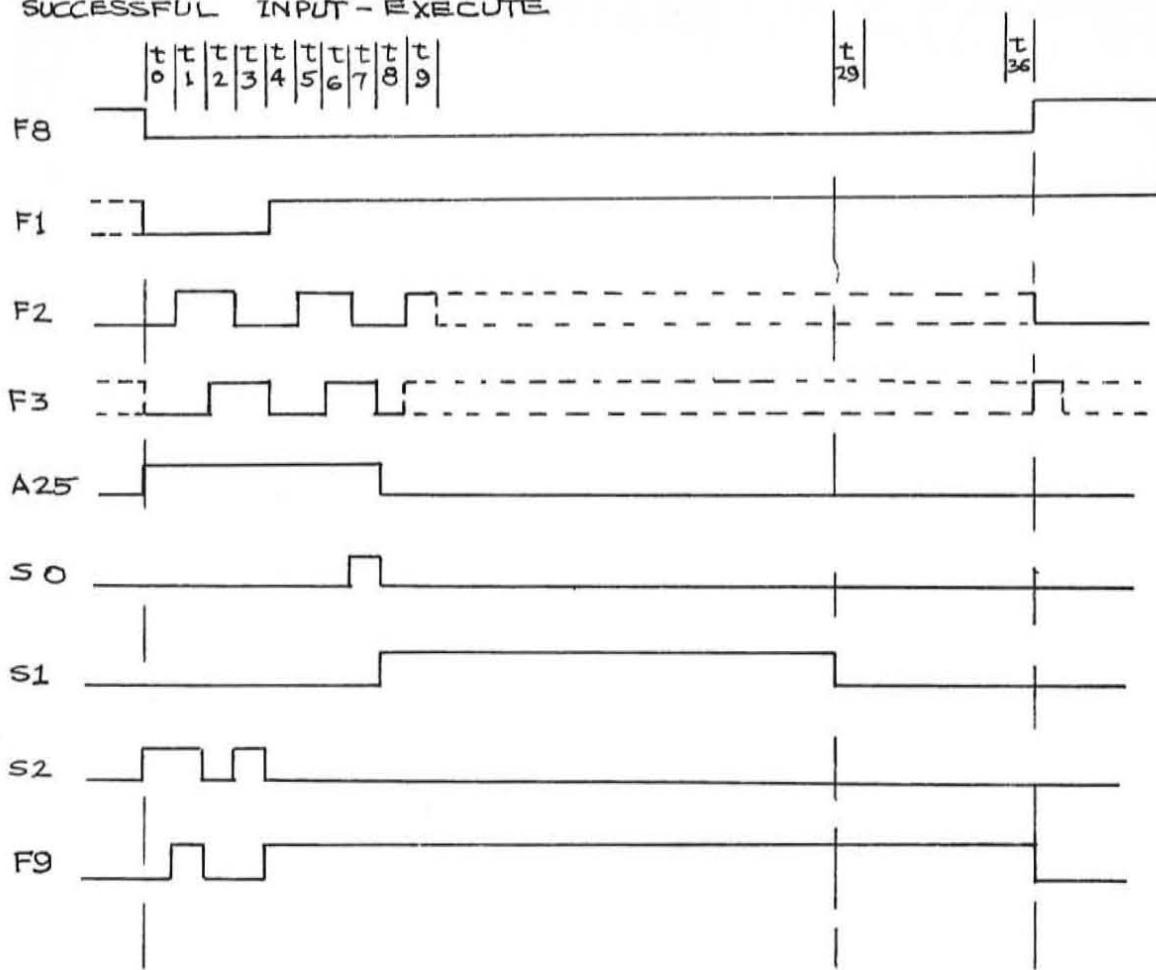
Input Clear.

Routing of the S1 signal to the selected device is accomplished by the three special Input Clear Select circuits, S11, S21 and S31 which represent respectively input devices 1, 2 and 3. Each of these circuits is under control of a Device Memory circuit, ID1, ID2, or ID3. The Device Memory circuit which corresponds to the selected device will have a low output, therefore the Input Clear Select circuit which has the low Device Memory signal input will route S1 to the device to which it is wired. The Input Clear Select circuit performs the And-Not logic function, i.e., S1 must be high and the Device Memory input signal must be low for its output to be high.

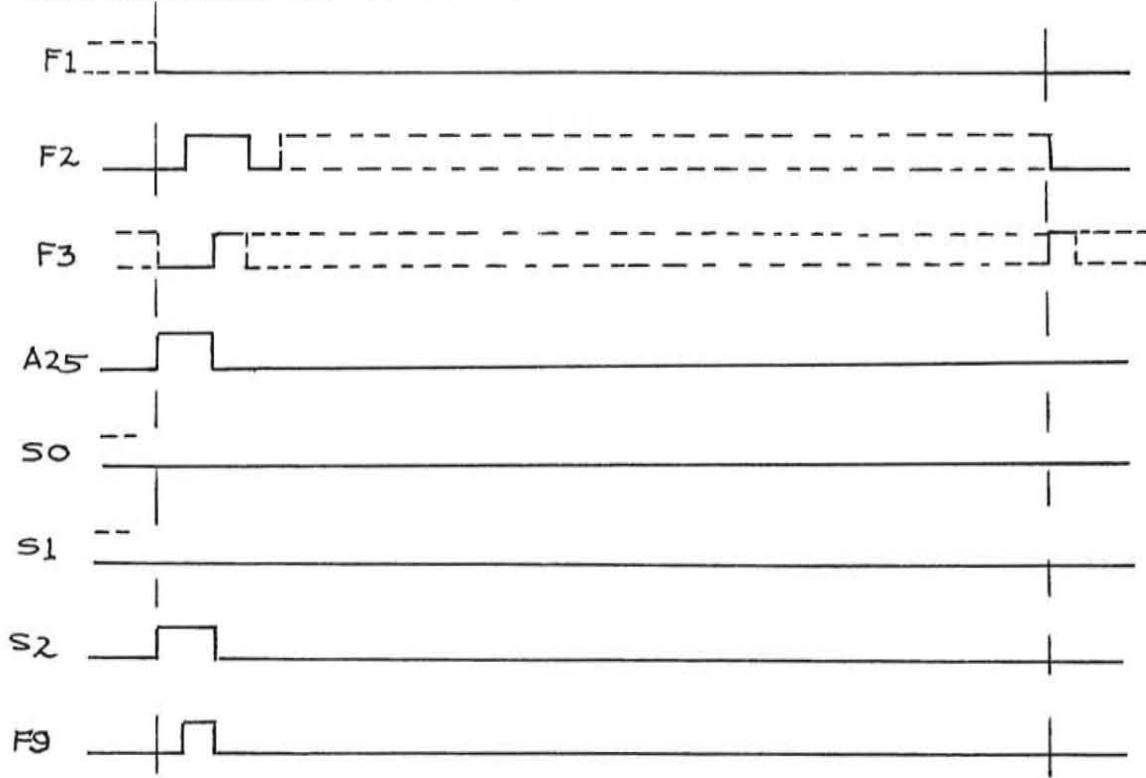
Input Busy.

The Input Busy net, $\overline{S4}$, monitors the busy signals of all the input devices. This net performs a straight-forward or-and logic function. The 'VV' gate is a gingle input 'or' gate which is paired with the busy signal from a device to form a two-input 'or' gate. The 'AA' gate performs the 'and' function. Thus, for example, the output of an 'or' pair is high if the Device Memory Circuit signal (ID1, ID2 or ID3) is high or the busy signal (S14, S24, or S34) is high. All three inputs to the 'AA' gate must be high for its output S4 to be high. Thus S4 will be high if a selected device is busy or if no device is selected.

SUCCESSFUL INPUT-EXECUTE



UNSUCCESSFUL INPUT-EXECUTE



II-4-5. Output (0010, Track 2 or Track 4).

The Output command is responsible for the flow of information from the computer to an output device. For an Output instruction to be successful, the output device addressed must be "not busy" -- that is, it must be plugged into the computer and not in the process of outputting a character by a previous Output instruction. To determine this, the computer samples the output device busy signal wire as the Output instruction code is being precessed into the Command Register. If the device is busy, the code is blocked from entering the Command Register, and a Skip code (all zeros) is entered in its place. During the Skip code which follows, the Accumulator information is kept intact.

If the device is not busy during the precess word, the Output instruction code is allowed to enter the Command Register and an Output-execute follows.

Generally speaking, this command outputs a character code from the Accumulator ($a_1 - a_7$) to an outside device. For multiple device applications, the Output instruction code can also be used to output a "device select" code to the device memory. The latter, as shown in the accompanying table, contains a code that specifies which device(s) will receive subsequent character outputs from the computer. It can be seen from the table that input devices are also specified in this way.

| <u>Device Select Code</u> | | | | | | |
|--|---|---|-------|---|--------------|----|
| An "1" in a_1 selects output device #1 | | | | | | |
| " | " | " | a_2 | " | " | " |
| " | " | " | a_3 | " | " | " |
| " | " | " | a_4 | " | input device | #1 |
| " | " | " | a_5 | " | " | " |
| " | " | " | a_6 | " | " | " |

The significance of the code present in $a_1 - a_7$ is defined by the a_8 bit. The presence of a 1 in a_8 means that $a_1 - a_7$ is a device-select code. The presence of a 0 in a_8 defines a character code in $a_1 - a_7$.

Initially, the Output command causes $a_1 - a_7$ to be output to a buffer register. Beginning at t_0 , a bit-counter generates the bit-time pulses which are used to sample and distribute the code bits into the buffer register as they appear at the Accumulator playback flip-flop output. Once entered into the buffer, the code is available either to the output device if a_8 contains a 0 or to the device memory if a_8 contains a 1.

The output character code bit assignments are listed in the following table:

| Acc. Location | Output Time | Buffer Register Pos. | Code Function | |
|-----------------|-------------|----------------------|---------------|-------|
| | | | IBM | ASCII |
| a_1 | t_0 | N1 | 1 | b1 |
| a_2 | t_1 | N2 | 2 | b2 |
| a_3 | t_2 | N3 | 4 | b3 |
| a_4 | t_3 | N4 | 8 | b4 |
| a_5 | t_4 | N5 | 0 | b5 |
| a_6 | t_5 | N6 | X | b6 |
| a_7 | t_6 | N7 | EL | b7 |
| (F9, parity FF) | t_7 | N8 | P | b8 |

THEORY OF OPERATION

Parity is generated during the read-out of the character into the buffer register. The sequential Accumulator playback output is gated to flip-flop F9 in such a way that the flip-flop changes state whenever a 1 appears in the character code. Since this flip-flop enters the Output-execute time in the reset state, its state after a character has been scanned will indicate whether that character contained an odd or an even number of bits. This scanning process occurs in every Output command, but actual output of parity to a device can occur only if a_8 is 0. A technician adjustment allows choice of either even or odd parity output by connecting either F9 or $\overline{F9}$ to the N8 input.

The outputting of the information stored in $a_1 - a_7$ is completed by t_7 and the Command Register, which has so far contained the Output instruction code, is reset to a Skip code. Beginning with t_8 , the computer then executes a Skip operation for the balance of the word-time.

An Output-execute will modify the Accumulator information in the following manner:

$a_0 - a_7$ 0's.
 a_8 Regenerated.
 a_{36} If the parity FF (F9) is high a 1 is recorded in a_{36} .

The recording of the state of the parity flip-flop in a_{36} has no significance to the programmer and occurs as a byproduct of design minimization.

SEQUENCE:

F3 (precess).

- t_{36}
- (1). The Output instruction code is decoded from F3 - F6, resetting F1 via N18.
 - (2). F6 is set if $\overline{E6}$ is high -- that is, if the output device is not busy. $\overline{E6}$ will be low if any selected device is busy or is not plugged in.

F3 (Output-execute).

- $t_0 - t_{36}$
- (1). Beginning at t_0 , a bit counter (F1, F2, F3) generates time pulses which read the information from the Accumulator playback flip-flop L2 sequentially via E3 into the output buffer register (N1 through N7). Since the information at L2 is one bit early, the bits sampled during $t_0 - t_6$ are $a_1 - a_7$.
 - (2). L2 is gated via N92 and N95 to F9, the parity flip-flop, such that F9 changes state whenever a 1 appears in the character code. F9 is initially in the reset or 0 state.
 - (3). If a_8 is 1 (L2 high), the device select signal DS is high, transferring the code in the buffer to the device memory (\overline{OD}_1 to \overline{OD}_3 , \overline{ID}_1 to \overline{ID}_3).

(2). If a_8 is 0 (indicating a character output) the output common signal E7 is high to set NC. If the state of F9 indicates that a parity bit should be output, the parity signal N8 is set high.

(1). A10 is high, resetting F6 and changing the instruction code in the Command Register to a Skip.

t_8 (1). NC is high as a result of E7 high during t_7 . During this time, the code in the buffer register is transferred to the output device. NC is a 1-shot with a period of 40 microsec.

$t_8 - t_{36}$ (1). F3 regenerates the rest of the Accumulator. It should be noted that a_8 automatically inherits a zero because F3 is low at t_8 as a result of its previous function in the bit counter.

$t_{29} - t_{35}$ (1). The gated supply VG goes high, resetting the buffer register.

t_{36} (1). The state of the parity flip-flop F9 is recorded in a_{36} . This is a consequence of design minimization and is of no significance to the programmer here.

(1). F3 is set via N31.

Unsuccessful Output.

F8 (precess).

t_{36} (1). The Output instruction code is decoded from F3 - F6, resetting F1 via N18.

(2). The set logic of F6 is inhibited by $\overline{E6}$ being low. $\overline{E6}$ will be low if any selected device is busy or not plugged in.

$\overline{F8}$ (Skip-execute).

$t_0 - t_{36}$ (1). The code in the Command Register now decodes as a Skip (see Skip description).

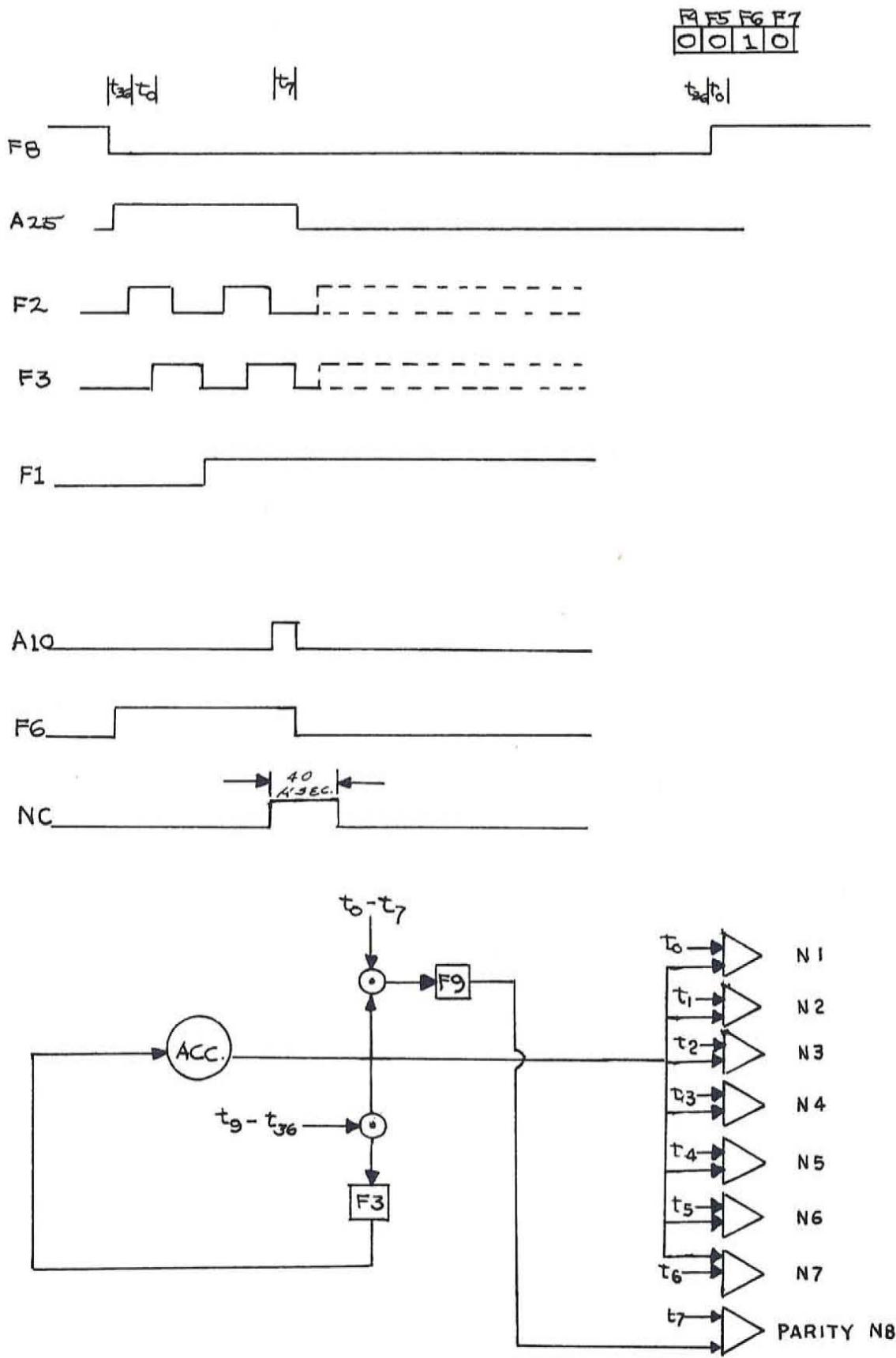
MULTIPLE OUTPUT:

Device Select.

Selection of output devices is achieved by making the Inhibit wire of the desired device low. The output Device Memory circuits (OD1, OD2, OD3) are tied directly to the Inhibit wires, hence the Device Memory circuit with a low output corresponds to the selected device.

Output Busy.

The output busy net, E6, monitors the busy signals of the output devices. Its output will be high (not busy) only if all inputs to the 'DD' gate in the net are high. Each of these inputs comes from a NO circuit. The NO output can only be low if its Device Memory input is low and its busy signal input is high. For all other combinations its output will be high. Note that if no device is selected, E6 will be high to indicate not busy.



II-4-6. Complement (0011).

The Complement instruction causes the number in the Accumulator to be replaced by its 2's-complement. General Storage remains unchanged and K is set to 1 only if a_{36} is changed from 1 to 0 as a result of the Complement-execute.

As shown in the accompanying Complement-execute flow diagram, the number in $a_1 - a_{36}$ is fed directly into the adder and is designated input (A). The adder output (C) is fed back via flip-flop F2 as the second input to the adder and is designated input (B). This introduces a delay of one binary bit which, in effect, multiplies the output (C) of the adder by two. The relationships of (A), (B), and (C) can be seen from the following simple equations:

$$(C) = (A) + (B) \quad (1)$$

$$(B) = 2(C) \quad (2)$$

Substituting,

$$(B) = 2(A + B) \quad (3)$$

or,

$$(B) = -2(A) \quad (4)$$

Substituting expression (4) into expression (1),

$$(C) = (A) + (-2A) = -(A) \quad (5)$$

The output (C) of the adder is therefore the negative value of the input (A) -- or, in binary language, the 2's-complement of A.

SEQUENCE:

F3 (precess).

- t_{36} (1). The Complement instruction code is decoded early via N13, re-setting F1.

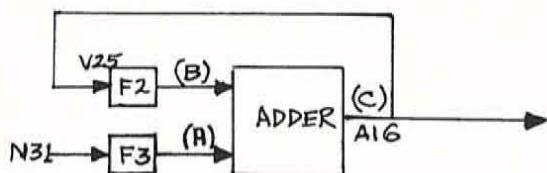
F8 (Complement-execute).

- t_0 (1). F3 is initially in the reset (0) state because no set logic was generated during the preceding word-time. Therefore, a 0 is recorded in a_0 .

- $t_1 - t_{36}$ (1). The information in $a_1 - a_{36}$ is gated to the adder via N31, F3.
 (2). The adder output A16 is gated back as an adder input via N22, F2. Output A16 is the sum output of F2 and F3.
 (3). If a carry is called for while the sum A16 is being generated, F1 is set via N11.

- t_{36} (4). The sum A16 is recorded in the Accumulator via N302, L1.
 (1). F7 in the Command Register is reset via N75.
 (2). F3 is set via N31.

| | | | |
|----|----|----|----|
| F4 | F5 | F6 | F7 |
| 0 | 0 | 1 | 1 |



II-4-7. Change Track (0100).

This command causes the computer to change storage tracks. The instruction code remains in the Command Register for only one bit time -- the first bit time (t_0) of the execute word. The track selection ring counter is triggered once during this time, causing it to move to the next track count configuration. At the end of t_0 , the Command Register is reset to a Skip code, which is executed for the remainder of the word-time $t_1 - t_{36}$.

The Accumulator contents, $a_0 - a_{36}$, are regenerated. The state of K is preserved.

SEQUENCE:

F8 (precess).

t_{36} (1). If a_0 equals 1, F3 is set via N32 and F2 is set via N24.

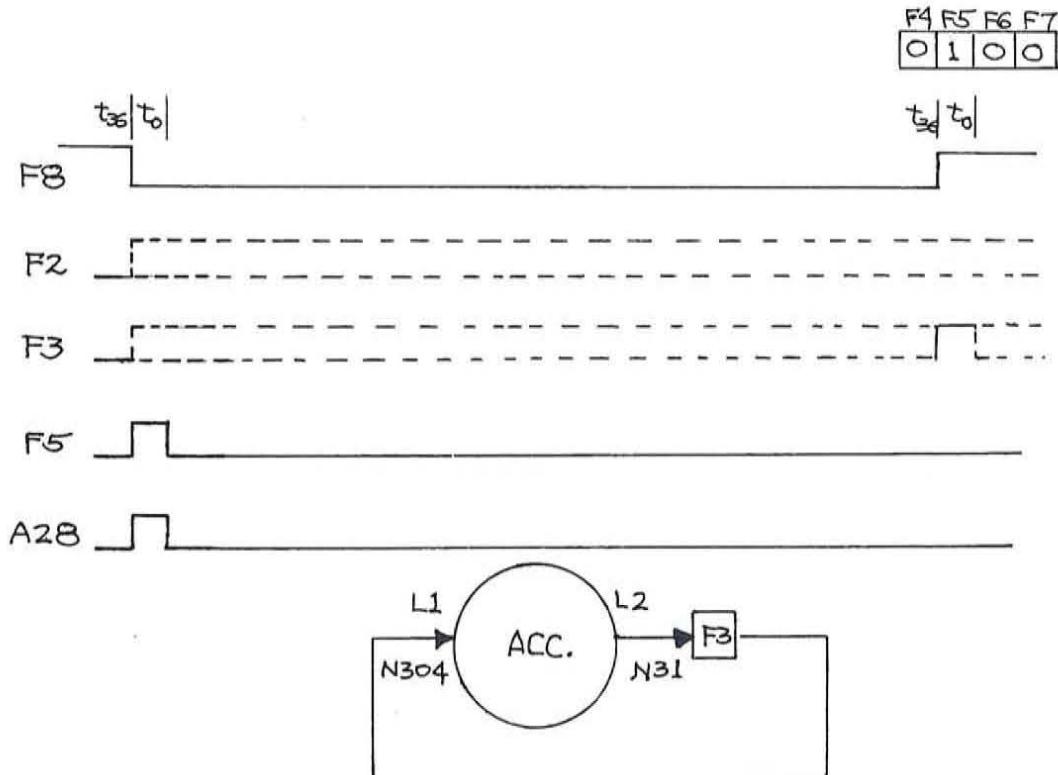
F8 (Change Track-execute).

t_0 (1). The Command Register is decoded by A28, which is high, thus directly triggering the track selection counter once and re-setting F5 in the Command Register via A10.

(2). The information in a_0 is regenerated via N304 in the L1 net.

$t_1 - t_{36}$ (1). The information in $a_1 - a_{36}$ is also regenerated via N304 in the L1 net.

t_{36} (1). F3 is set via N31.



II-4-8. Left Shift (0101).

The Left Shift instruction causes the contents of $a_0 - a_{35}$ to be shifted one bit position toward the high-order end of the Accumulator -- that is, into $a_1 - a_{36}$. The state of K at t_0 is recorded in a_0 . The final state of K is dependent on the initial contents of a_{36} . The shift is accomplished by adding the Accumulator to itself and recording the sum back in the Accumulator, the result being a one-bit shift of the original information in the high-order direction.

SEQUENCE:

F8 (precess).

t_{36} (1). L2, the early Accumulator playback, contains the a_0 bit at this time. If $a_0 = 1$, F3 is set via N32 and F2 is set via N24.

F8 (Left Shift-execute).

t_0 (1). The state of K is recorded in a_0 , as explained by the following tables (note that F2 and F3 represent a_0 as determined in t_{36} of F8):

$t_0 (a_0 = 0)$

| | | |
|----------|---|---------------------|
| $F2 = 0$ | } | Record 0 in a_0 . |
| $F3 = 0$ | | |
| $K = 0$ | | |

| | | |
|----------|---|--|
| $F2 = 0$ | } | Record 1 in a_0 via N161, A16, N302, L1. |
| $F3 = 0$ | | |
| $K = 1$ | | |

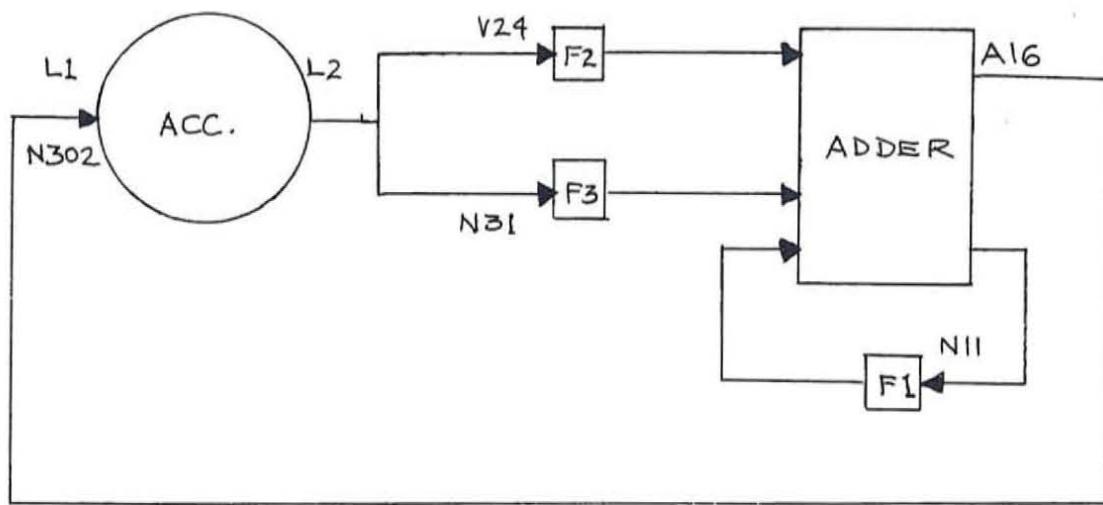
$t_0 (a_0 = 1)$

| | | |
|----------|---|---------------------|
| $F2 = 1$ | } | Record 0 in a_0 . |
| $F3 = 1$ | | |
| $K = 0$ | | |

| | | |
|----------|---|--|
| $F2 = 1$ | } | Record 1 in a_0 via N162, A16, N302, L1. |
| $F3 = 1$ | | |
| $K = 1$ | | |

$t_0 - t_{36}$ (1). The Accumulator playback L2 is gated to the adder A16 via inputs F2 and F3. The gates to F2 and F3 are V24 and N31 respectively.
 (2). The output sum of the adder is recorded in the Accumulator via N302 and L1.
 (3). If the carry logic gate N11 is high, F1 is set; if gate N15 is high, F1 is reset.
 t_{36} (1). F3 is set via N31.

| | | | |
|----|----|----|----|
| F4 | F5 | F6 | F7 |
| 0 | 1 | 0 | 1 |



II-4-9. Transfer (0110).

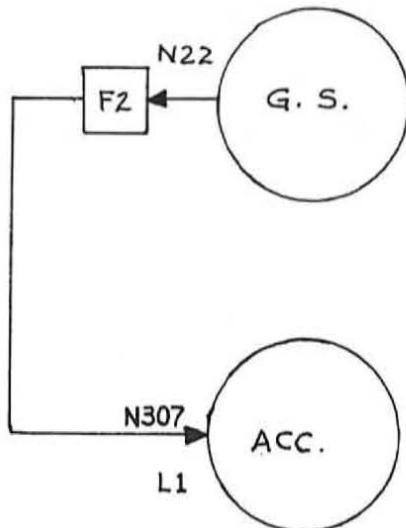
The Transfer instruction causes the General Storage information ($g_1 - g_{36}$) available at the playback during the execute word-time to be recorded in the Accumulator ($a_1 - a_{36}$). A zero is recorded in a_0 , General Storage is unchanged and K is preserved.

SEQUENCE:

F8 (Transfer-execute).

- t_0 (1). F2 is initially in the reset (0) state because there was no set logic during the previous bit time. Therefore, a zero is recorded in a_0 .
- $t_1 - t_{36}$ (1). General Storage playback, L8, is recorded in the Accumulator via N22, F2, N307, L1.
- t_{36} (1). A10 is high, resetting F5 and F6 in the Command Register.
(2). F3 is set via N31.

| F4 | F5 | F6 | F7 |
|----|----|----|----|
| 0 | 1 | 1 | 0 |



II-4-10. Add (0111).

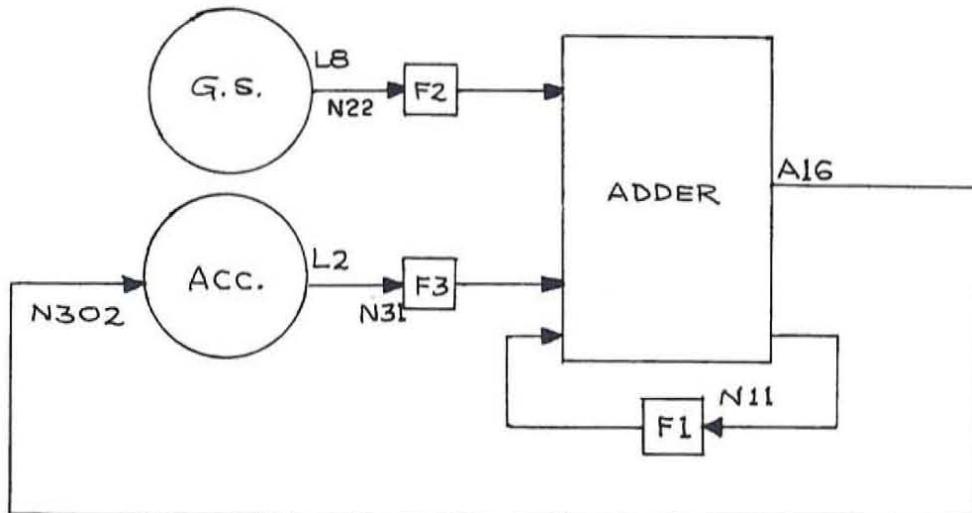
General Storage information ($g_1 - g_{36}$) available at the playback during the Add-execute is added to the Accumulator ($a_1 - a_{36}$) by this instruction and the sum is recorded in the Accumulator. The overflow goes into K; the initial state of K is recorded in a_0 . General Storage remains unchanged.

SEQUENCE:

F8 (Add-execute).

- t_0 (1). The adder gate A16 output is recorded in the Accumulator via N302, L1. F2 and F3 are both low during this bit time because there was no set logic during the previous bit time. Therefore, A16 is equivalent to the state of F1.
 (2). F1 is reset via N15 since F2 and F3 are both low.
- $t_1 - t_{36}$ (1). General Storage L8 is gated to F2 via N22; the playback of the Accumulator, L2, is gated to F3 via N31.
 (2). Adder output A16 is recorded in the Accumulator via N302 and L1.
 (3). If the carry logic gate N11 is high, F1 is set; it is reset if gate N15 is high.
- t_{36} (1). F7 in the Command Register is reset via N75.
 (2). F3 is set via N31.

| F4 | F5 | F6 | F7 |
|----|----|----|----|
| 0 | 1 | 1 | 1 |



II-4-11. Jump (1000).

The Jump command is designed to bring a new word from General Storage into the Orders Channel. The a_0 bit is set to 0; $a_1 - a_{36}$ and K are preserved. As a result of the Jump, c_0 is set to 1 and $g_1 - g_{36}$ is written into $c_1 - c_{36}$.

The Jump departs from the usual two word-time precess-execute pattern which is characteristic of the majority of the micro-commands. Briefly, the sequence is as follows:

(1). The Jump code is precessed into the Command Register in the usual way during F8 time.

(2). The Jump-execute, which follows, preserves the Accumulator and the K memory and sets up a special memory.

(3). The next word-time is a F8 and is defined, by virtue of the special memory set up during the execute-time, as a transfer word-time during which a new Order word from General Storage is written into the Orders Channel. No pre-cession occurs.

(4). An automatic Skip-execute follows which serves the purpose of pre-serving the Accumulator and K until the first command code is precessed into the Command Register from the new Order word, which, of course, will occur during the next F8.

SEQUENCE:

F8 (precess).

t_{36} (1). The Jump command is precessed into the Command Register.

F8 (execute).

t_0 (1). F3 is low because there was no set logic during the previous bit-time. Therefore, a zero is recorded in a_0 via N304, L1.

$t_1 - t_{36}$ (1). $a_1 - a_{36}$ is regenerated via N31, F3, N304, L1.

t_{36} (1). F4 in the Command Register is reset via N101, A10, N45.

(2). F2 is set via N21.

(3). F3 is set via N31.

F3 (Non precess).

t_0 (1). F3 is high, therefore a 1 is recorded in c_0 via N153.

$t_0 - t_{36}$ (1). F2 is maintained high via N23.

$t_1 - t_{36}$ (1). The General Storage playback, L_0 , is gated into L1 via N34, F3, N153, L301, L1.

F8 (Automatic Skip-execute).

t_0 (1). F2 is reset because there is no set logic.

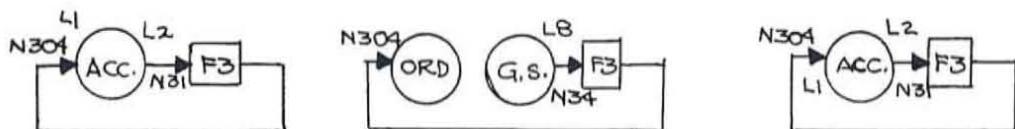
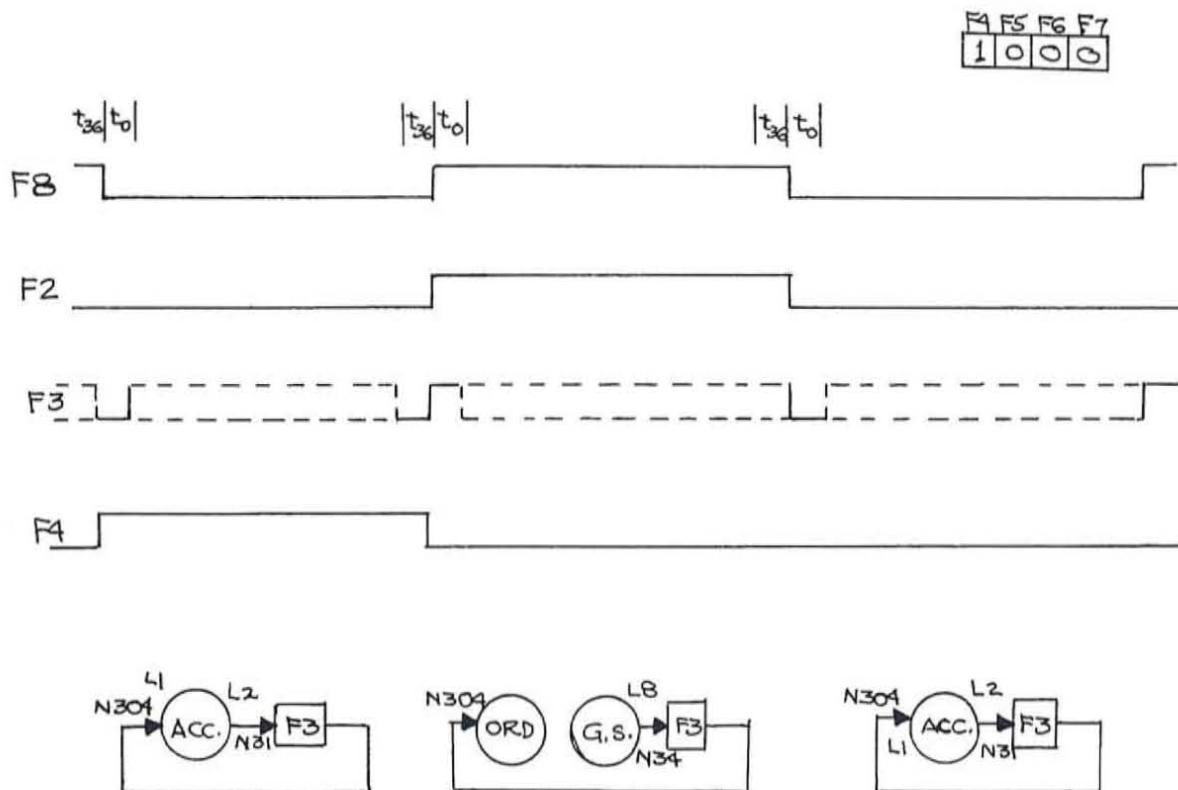
$t_1 - t_{36}$ (1). The Command Register decodes a Skip; therefore, $a_1 - a_{36}$ is regenerated via N304, L1.

t_{36} (1). F3 is set via N31.

Note that at t_0 of the following F8 precess (which is not part of the Jump operation), F3 will equal 1. As a consequence of the precession, this 1 will be preceded by the 0's (of the automatic Skip code) in the Command Register in the order of bits recorded back into the Orders Channel. The low-order end of the Orders Channel following this precess will look thus:

| | | | | | | | | |
|-----------------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| c ₃₆ | c ₇ | c ₆ | c ₅ | c ₄ | c ₃ | c ₂ | c ₁ | c ₀ |
| x..... | x | x | x | 1 | 0 | 0 | 0 | 0 |

The four bracketed bits, it will be noted, decode as a Jump. This code will be moved toward the high-order end with each subsequent precess and, if not preceded by a programmed Jump or a Leap, will eventually appear in the Command Register



II-4-12. Multiply By Five (1001, Track 1).

The contents of the Accumulator $a_1 - a_{36}$ is multiplied by five. The a_0 bit is reset to zero. K is reset if there is no overflow, and may be set if there is an overflow. General Storage remains unchanged.

The Multiply-By-Five (x5) instruction is effected by simultaneously providing two separate paths for the flow of information from the Accumulator to the adder -- one direct, the other with a delay of two bit-times. The latter effectively multiplies the contents of the Accumulator by four. The sum output of the adder is:

$$N + 4N = 5N$$

where N is the number in the Accumulator prior to the multiplication by five.

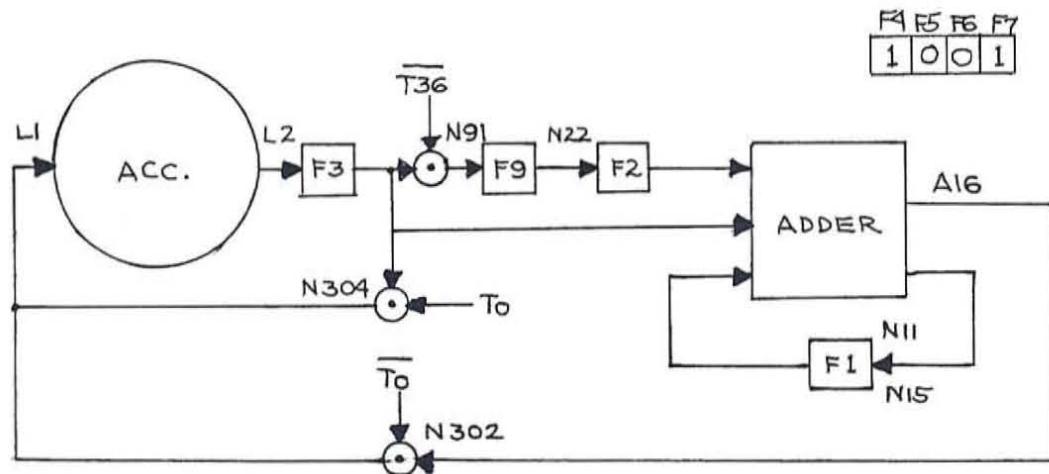
SEQUENCE:

F8 (precess).

- t_{36} (1). The 1001-code is shifted into the Command Register.
 (2). N17 is high, resetting F1.
 (3). No set trigger is generated for F3.

F8 (X5-execute).

- t_0 (1). At this time, F1 and F3 both equal zero. The zero in F3 is recorded in a_0 via N304, L1. After t_0 , N304 will no longer be operative, since the code in the Command Register will have been altered, as will be demonstrated.
 (2). N44 decodes the Command Register and the track counter and resets F4. The result of this is an alteration of the code configuration in the Command Register from 1001 to 0001, in order to distinguish this command from a Slide, the code for which is also 1001.
- $t_1 - t_{36}$ (1). The early Accumulator playback L2 is gated to F3 via N31. Since F3 is an input to the adder A16, a direct Accumulator information path is thus established.
 (2). F3 is gated, via set and reset nets N91 and N96, to F9, which in turn is gated by N22 to F2. The latter is a second input to the adder (A16) logic. F2 and F9 together establish the two-bit delay connecting link between F3 and the adder.
 (3). N11 and N15 are set and reset gates for the carry flip-flop F1.
 (4). The adder output is gated to the Accumulator record circuits via N302, L1.
- t_{36} (1). F9 is reset by N94.
 (2). F3 is set via N31.



THEORY OF OPERATION

II-4-13. Partial Divide By Five (1001, Track 3).

The Partial-Divide-By-Five (PDF) is executed on $a_1 - a_{36}$. A zero is recorded in a_0 . Any overflow resulting from the operation is recorded in K.

The division of a number by five is accomplished by a command circuit followed by a programmed subroutine of further commands. Thus, the command itself is called Partial Divide By Five. It produces a scrambled quotient in one word-time; the subsequent subroutine unscrambles this quotient. Separate unscrambling routines are necessary for positive and negative numbers.

Basically, this partial division by five is effected by the following steps:

1. 00 is added to the two low-order bits of the number.
2. The resulting two bits are inverted and added to the next two higher bits of the original number.
3. The resulting two bits of this operation are then inverted and added to the two next higher bits of the original number, taking into account the carry from the preceding step.
4. The above procedure is repeated until the high-order end of the number is reached.

The logic which carries out the above steps and produces the scrambled quotient is shown in the accompanying figure. At the beginning of the word-time in which this circuit is used, the carry flip-flop F1 and the two-bit delay flip-flops, F9 and F2, are reset to zero. An overflow bit will remain in F1 at the end of the word-time.

SEQUENCE:

F8 (precess).

- t₃₆ (1). The 1001 code is shifted into the Command Register.
 (2). N17 is high and resets F1.
 (3). No set trigger is generated for F3 at this time.

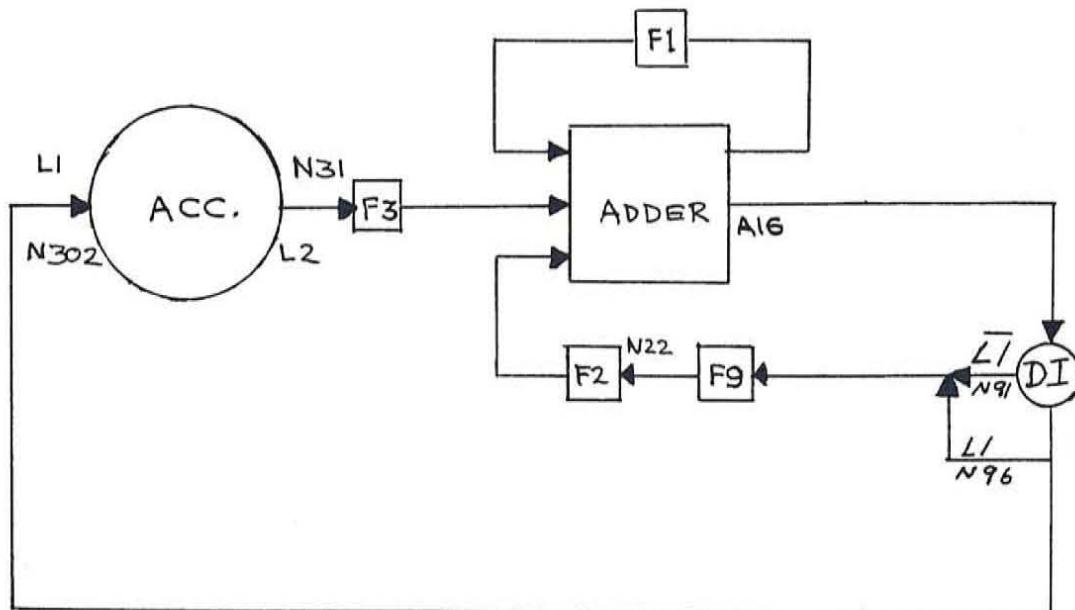
F8 (PDF-execute).

- t₀ (1). F1 and F3 are both low. The zero in F3 is recorded in a_0 via N304, L1. After t₀, N304 will no longer be operative, since the code in the Command Register will have been altered, as will be demonstrated.
 (2). Gate N44 decodes the Command Register and the track counter, resetting F4. As a result, the code in the Command Register is altered from 1001 to 0001, to distinguish this command from a Slide which also has instruction code 1001.
- t₁ - t₃₅ (1). The early Accumulator playback L2 is gated to F3 via N31. F3 is an input to the adder.

(2). $\overline{L1}$ (the inverted Accumulator record gate output) is gated via N22 to F2 and via N91 to F9. L1 is gated via N96 to F9. The latter forms a second input to the adder logic. F2 and F9 together establish a two-bit delay.

- t1 - t36 (1). N11 and N15 are set and reset gates for the carry flip-flop F1.
 (2). The adder output A16 is gated to the Accumulator record circuit via N302, L1.
- t36 (1). F9 is reset via N94.
 (2). F3 is set via N31.

| | | | |
|----|----|----|----|
| F4 | F5 | F6 | F7 |
| 1 | 0 | 0 | 1 |



II-4-14. Slide (1001, Track 2 or Track 4).

Basically, the Slide instruction is a "do-nothing" operation during which the Accumulator information is unaffected and the precession of the order word is suspended. K is reset. Accumulator bits $a_1 - a_{36}$ are regenerated and a_0 is set to 1 if the Slide is of minimum duration (two word-times) or to 0 if it is longer (up to 150 word-times). The F8 periods during this instruction are non-precess. The order word is kept intact. General Storage is unchanged.

A Slide is terminated when a zero flag bit (one per track) is located co-incident with F8-time. At that point, the precession of the order word is immediately resumed and the next order code is shifted into the Command Register.

SEQUENCE:

F8 (precess).

- t_{36} (1). The Slide instruction code is shifted into the Command Register.
(2). The instruction code is early-decoded from F3 - F6 by N17, re-setting F1 and by N35, setting F3.

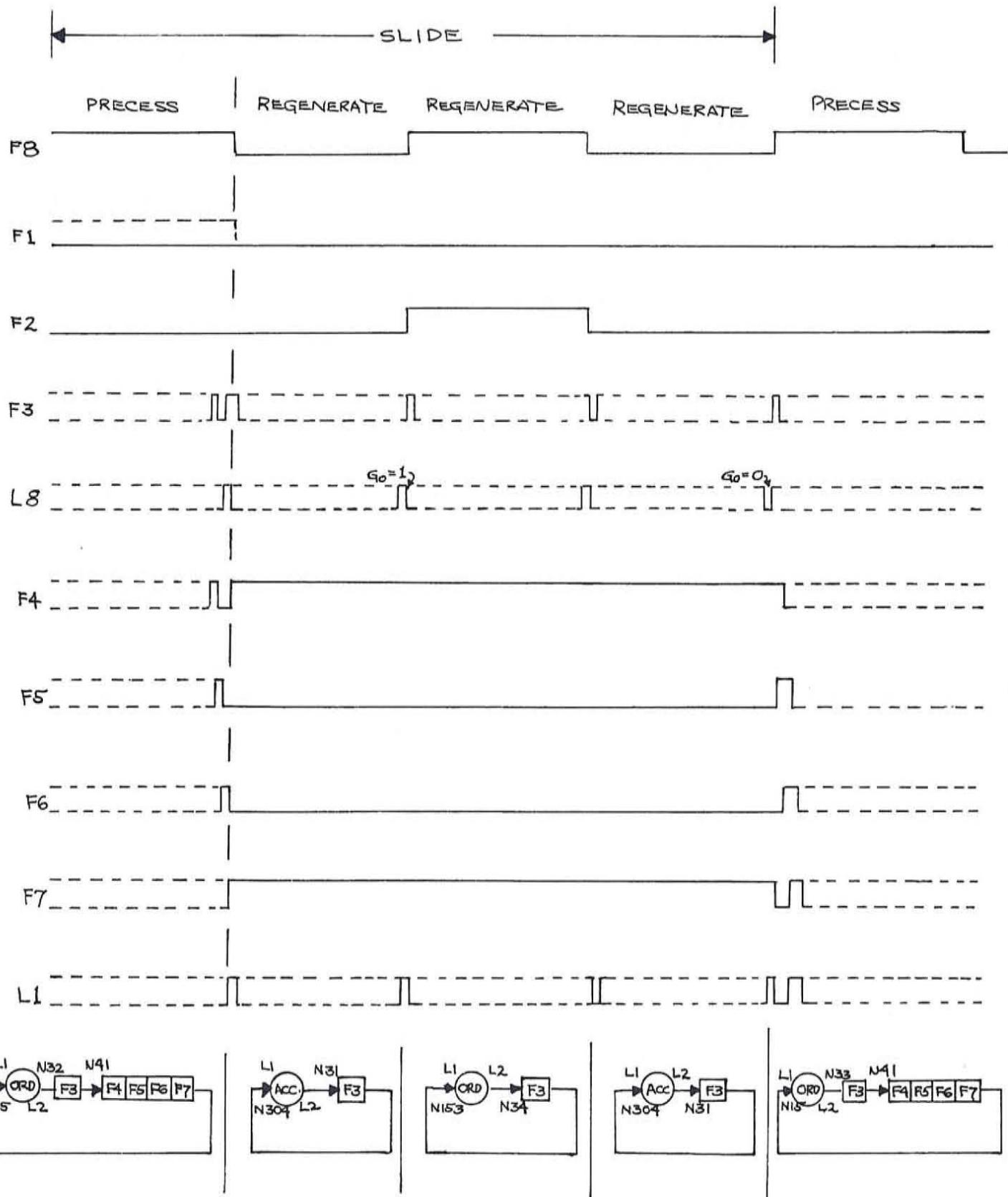
 $\overline{F3}$ (Slide-execute).

- t_0 (1). The 1 in F3 is recorded in a_0 via N304, L1.
 $t_1 - t_{36}$ (1). The Accumulator, $a_1 - a_{36}$, is regenerated via L2, N31, F3, N304, L1.
 t_{36} (1). N31 is high, setting F3.
(2). If the General Storage early playback, L8, is high -- indicating that the General Storage flag bit g_0 equals 1 -- F2 is set via N21. Should this occur, the following word-time will be defined as a non-precess F8 (as described below). If it does not occur (that is, $g_0 = 0$), the Slide command is terminated and the word-time which follows will be a regular precess F8, which will bring in another word to be executed.
(3). The Command Register is not reset.

F8 (non-precess).

- $t_0 - t_{36}$ (1). F2 is maintained high for this word-time by N23. The condition F2.F8 defines a non-precess F8-time. The 1 in F3 at t_0 is recorded in c_0 by N153, A15, L1; the Orders Channel $c_1 - c_{36}$ is regenerated via L2, N34, F3, N153, A15, L1.
 t_{36} (1). No set trigger is generated for F3 at this time.

The word-time which follows is an $\overline{F3}$ (execute) time identical to the one which preceded the above F8, except that $\overline{F3}$ equals 0 at t_0 and F2 becomes reset, with no set logic generated, at t_0 . This $\overline{F3}$ -time is typical of all $\overline{F3}$ -times which may follow during this command.



II-4-15. Jump Conditional (1010).

The Jump Conditional operates in exactly the same manner as the Jump if K equals 1 initially. If K equals 0, the code in the Command Register is converted to a Skip code. If the command is successful (that is, if a Jump is executed), a_0 is reset to zero, $a_1 - a_{36}$ is regenerated, c_0 is set to 1 and $g_1 - g_{36}$ is recorded in $c_1 - c_{36}$. If the command is unsuccessful, a_0 is reset to zero, $a_1 - a_{36}$ is regenerated, c_0 is set to 1 and $c_1 - c_{36}$ is normally precessed. In both cases, the state of K and General Storage are preserved throughout.

SEQUENCE:

Case I: Successful Jump. K = 1.F8 (execute).

$t_0 - t_{36}$ (1). $a_1 - a_{36}$ is regenerated via N31, N304, L1.
 t_{36} (1). F4 and F6 in the Command Register are both reset by A10.
(2). F2 is set via N21.

F8 (non-precess).

$t_0 - t_{36}$ (1). F2 is maintained high via N23.
 $t_1 - t_{36}$ (1). General Storage playback L8 is gated into L1 via N34, N153, F3.

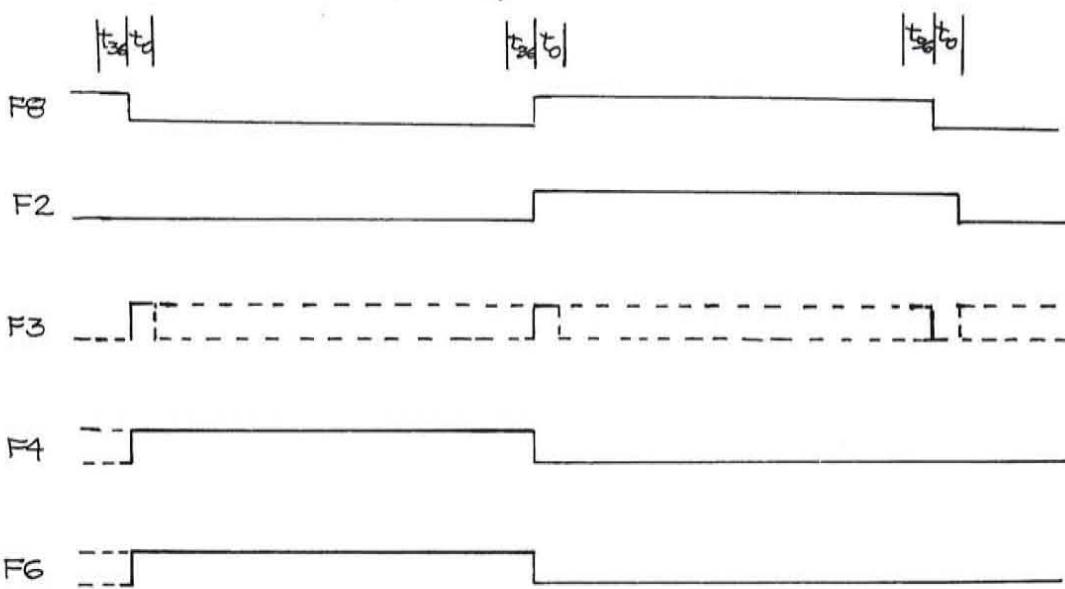
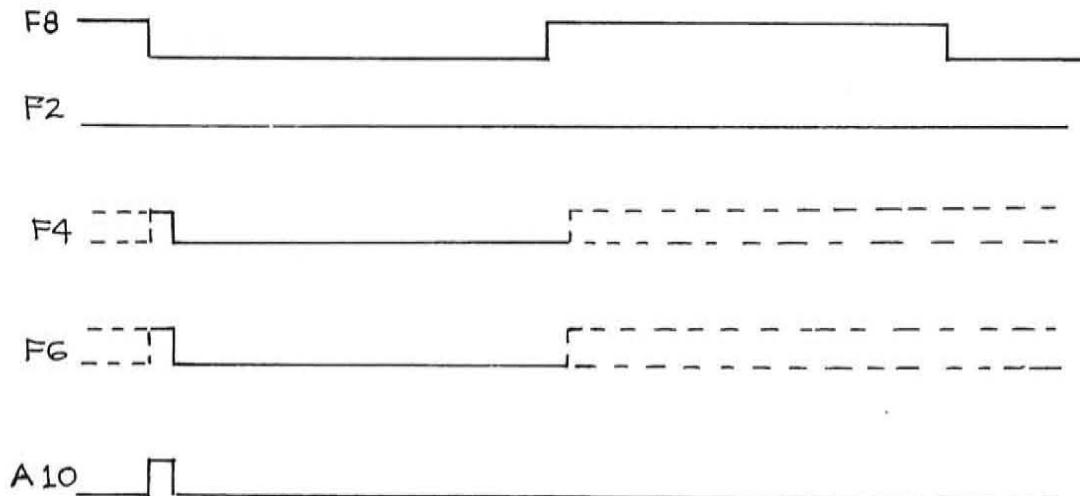
 $\overline{F8}$ (execute).

t_0 (1). F2 is reset because there is no set logic.
 $t_1 - t_{36}$ (1). The Command register decodes a Skip; therefore, $a_1 - a_{36}$ is regenerated via N304, L1.
 t_{36} (1). F3 is set via N31.

Case II: Unsuccessful Jump. K = 0. $\overline{F8}$ (execute).

t_0 (1). F4 and F6 are reset by A27 and A10.
 $t_1 - t_{36}$ (1). The Command Register now decodes a Skip operation and the Accumulator regenerates (see Skip command description).

| | | | |
|----|----|----|----|
| F4 | F5 | F6 | F7 |
| 1 | 0 | 1 | 0 |

SUCCESSFUL ($F_1 = 1$)UNSUCCESSFUL ($F_1 = 0$)

II-4-16. Delay (1011).

Basically, the Delay instruction is a "do-nothing" command during which the normal precess-execute sequence is suspended, which lasts a specified number of word-times (covering both F8 and F8 word-times), and automatically terminates in a Jump Conditional operation. The success of the Jump Conditional depends entirely on the state of K, which must be established before execution of the Delay. If K = 1 initially, it will be successful; if K = 0, it will be unsuccessful.

The duration of the delay period of the command is determined by the magnitude of the programmed delay number, a seven-bit number which partially occupies the two tetrads adjacent to the Delay code in the Orders Channel. This number is increased by 1 at every F8-time -- during the Delay, they are all non-precess times -- until it contains all 1's, its maximum magnitude. At this point, the Delay code in the Command Register is converted to a Jump Conditional code, which is then executed in the same manner as a programmed Jump Conditional.

As in the normally programmed case, a successful Jump Conditional results in the transfer of a new Order word from General Storage to the Orders Channel. An unsuccessful one results in the resumption of the precession of the Order word currently in the Orders Channel.

During the Delay command, General Storage and K remain unchanged. The a_0 bit is reset to zero and $a_1 - a_{36}$ is preserved.

The Delay code, with the delay number and a spare bit, use up three adjoining tetrads in the Orders Channel. The code itself occupies the high-order tetrad of the three, while the seven lower-order bits of the remaining two tetrads are used for the delay number; the single high-order bit of these two tetrads contains the spare bit. The latter is programmed adjacent to the Delay code.

When the Delay code has been precessed into the Command Register to be executed, the spare bit and the seven delay number bits are shifted into the two high-order tetrads of the Orders Channel ($c_{29} - c_{36}$). The spare bit, which occupies c_{36} , remains undisturbed for the duration of the Delay command.

During each F8-time, the delay number, which is in $c_{29} - c_{35}$, is increased by 1. This is the "count-up" procedure. When the delay number consists of all 1's (indicating maximum magnitude), the Delay code in the Command Register is converted to a Jump Conditional code. During this same period, the delay number is further increased by 1 and the result -- all zeros -- is recorded back into $c_{29} - c_{35}$. When the Delay code in the Command Register has been converted, the delay segment of the operation is concluded and the execution of the Jump Conditional segment follows immediately (see the Jump Conditional description).

It has been pointed out that, when K = 0, the Jump Conditional will be unsuccessful. Following execution of this unsuccessful Jump Conditional, the computer resumes its normal routine of precess-execute sequencing. The first two code-tetrads which become available for precessing into the Command Register are a by-product of the Delay command in that they represented the seven delay number bits and the spare bit during the Delay operation. The seven bits which pre-

viously contained the delay number ($c_{29} - c_{35}$) are all zeros (as determined during the final count-up F8-time), while the spare bit may be zero or one, as programmed. The high-order tetrad ($c_{33} - c_{36}$) is the first to be processed; it will decode either as a Skip (0000) or a Jump (1000), depending on the spare bit (c_{36}). The second code-tetrad (0000) will be precessed and executed as a Skip, but only if the first code-tetrad was a Skip.

If $K = 1$, of course, the Jump Conditional segment of the Delay is successful, and a new Order word is transferred from General Storage to the Orders Channel. Obviously, in this case, the old delay number bits and the spare bit lose significance, since they're replaced along with the entire previous Order word by a new Order word.

The duration of the Delay, if $K = 1$, is $2d+4$ word-times. If $K = 0$, the duration is $2d+6$ word-times. The "d" here is the delay number. These computed values include the precess word-time which brought in the Delay code initially. If $K = 1$, the formula also includes the entire Jump Conditional time; if $K = 0$, it includes the time required to precess and execute the codes represented by the delay number and the spare bit.

SEQUENCE:

F8 (precess).

- t_{36} (1). The Delay code is shifted into the Command Register and the delay number bits and the spare bit are shifted into $c_{29} - c_{36}$. At this time, F_3 is not set.

F8 (Delay-execute).

- $t_0 - t_{36}$ (1). The zero in F_3 is recorded in a_0 , and $a_1 - a_{36}$ is regenerated via N_{31} , F_3 , N_{304} , L_1 .

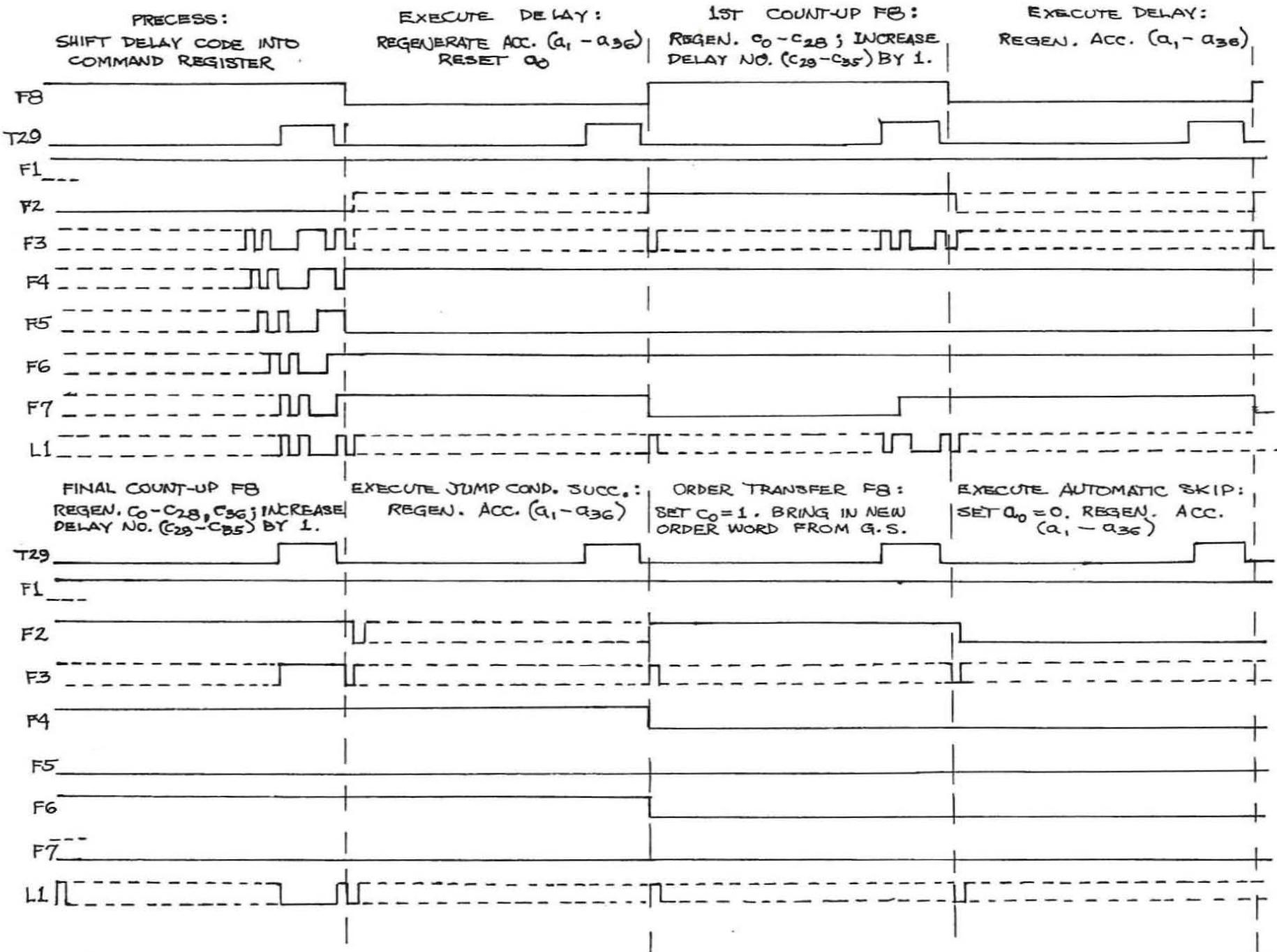
- t_{36} (1). F_3 is now set by N_{31} .
 (2). The Command Register low-order flip-flop F_7 is reset by N_{75} . F_4 and F_6 remain set.
 (3). F_2 is set by N_{21} .

F8 (Count-up).

- $t_0 - t_{28}$ (1). F_2 remains high during this word-time by N_{23} .

- (2). The Order word information ($c_1 - c_{28}$) is regenerated via N_{34} , F_3 , N_{153} , A_{15} , L_1 .

- $t_{29} - t_{35}$ (1). F_7 , via N_{152} , A_{15} , N_{73} , scans the delay number ($c_{29} - c_{35}$) for a zero bit, low-order first. The first zero to appear, if any, will set F_7 . It should be noted that the spare bit, c_{36} , is not included in this scan.
 (2). During the scanning process, if F_7 is low (0) -- indicating that a zero has not yet been found -- the delay number bits are inverted



before they are recorded back in $c_{29} - c_{35}$ via N34, F3, N152, A15, L1. If a zero has been found and F7 is high, the delay number bits are simply regenerated via N34, F3, N153, A15, L1. The result of this is that the delay number is increased by 1, as illustrated in the following example:

| | c_{35} | c_{34} | c_{33} | c_{32} | c_{31} | c_{30} | c_{29} | |
|------------------|----------|----------|----------|----------|----------|----------|----------|-----|
| delay number.... | 0 | 0 | 0 | 0 | 1 | 0 | 1 | (5) |
| F7 flip-flop.... | 1 | 1 | 1 | 1 | 1 | 0 | 0 | |
| L1..... | 0 | 0 | 0 | 0 | 1 | 1 | 0 | (6) |

Thus, when the delay number finally contains all 1's, F7 will not set at all, and the 1's will all be inverted and recorded back as zeros.

- t_{36}
- (1). The spare bit is regenerated via N153.
 - (2). If F7 is high, the Command Register will still contain a Delay code as the computer goes into an execute (F8) word-time identical to the one which preceded this F8 count-up time. If F7 is low, the Command Register will contain a 1010 code and the succeeding word-time will be a Jump Conditional-execute.

II-4-17. Write (1100).

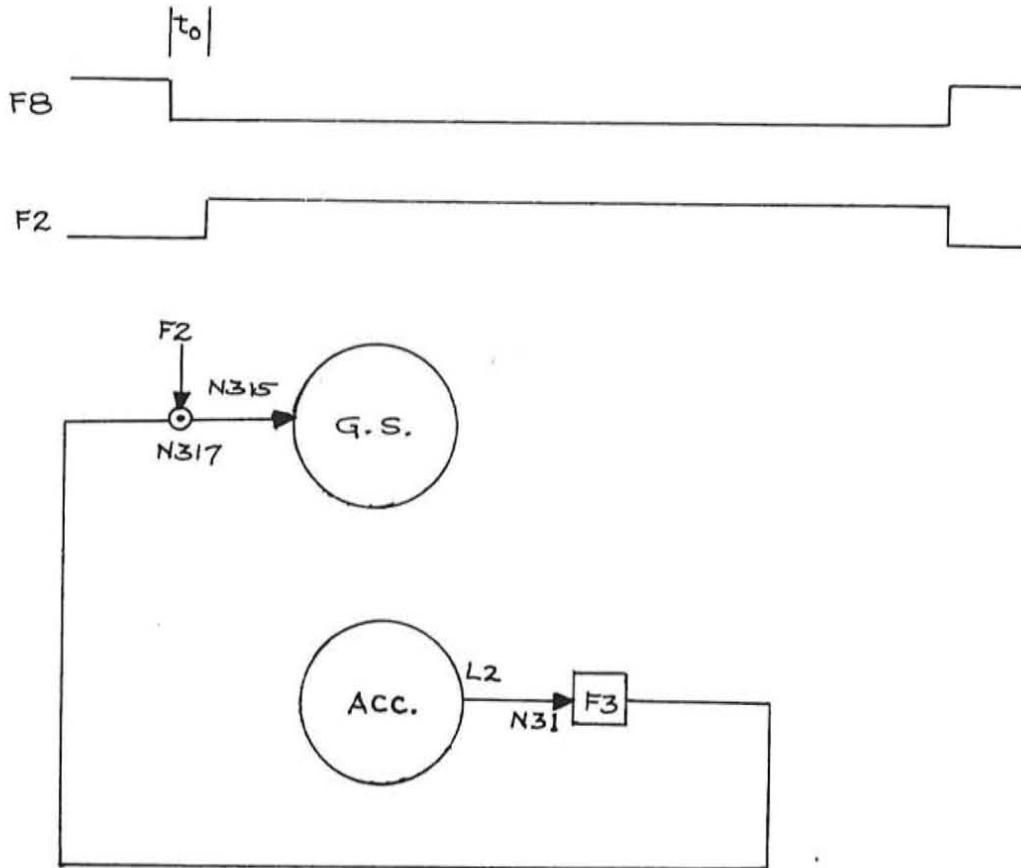
The Write command causes the contents of the Accumulator, $a_1 - a_{36}$, to be recorded in the General Storage location which passes under the General Storage record head during the Write-execute time. The flag bit g_0 is not modified and K is preserved. The entire Accumulator, $a_0 - a_{36}$ is preserved.

SEQUENCE:

F8 (Write-execute).

- t_0 (1). F2 is set via N22.
 (2). F3 is low because there was no set logic during the previous bit time. Therefore record a zero in a_0 via N304, L1.
- $t_1 - t_{36}$ (1). The Accumulator is regenerated via N31, F3, N303, L1.
 (2). The Accumulator playback L2 is gated via N31, F3 to the General Storage record gates L5 (record zeros) and L7 (record ones).
- t_{36} (1). A10 is high, resetting F4 and F5 in the Command Register.
 (2). F3 is set via N31.

| F4 | F5 | F6 | F7 |
|----|----|----|----|
| 1 | 1 | 0 | 0 |



II-4-18. Right Shift (1101).

The Right Shift instruction causes the Accumulator information to be shifted one bit position in the low-order direction. Thus, $a_1 - a_{36}$ is shifted to $a_0 - a_{35}$. The original information in a_0 is lost, while a zero is recorded in a_{36} .

During a Right Shift-execute, the Accumulator playback L2 is gated directly to the Accumulator record net. Since L2 represents the Accumulator one bit-time early with respect to normal record time, information thus recorded is shifted one bit toward the low-order end.

General Storage remains unchanged and a_1 is shifted into K.

SEQUENCE:

F8 (precess).

t_{36} (1). The Right Shift code is decoded early by gate N17, resetting F1.

F8 (Right Shift-execute).

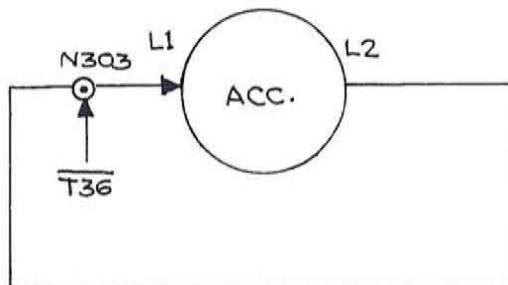
t_0 (1). At this time, the Accumulator playback L2 represents a_1 . If the latter contains a 1, F1 is set via N12. Set F2 via N22.

$t_0 - t_{35}$ (1). L2 is gated via N303, L1, to the Accumulator record.

t_{36} (1). The T36 variable in N303 of L1 blocks recording of a 1 in a_{36} , so that a zero is automatically recorded in that position.

(2). F3 is set via N31.

| F4 | F5 | F6 | F7 |
|----|----|----|----|
| 1 | 1 | 0 | 1 |



II-4-19. Add Conditional (1110).

The purpose of this command is to add the contents of General Storage to the Accumulator if K is initially 1. If this conditiona exists, the Add Conditional-execute is converted to an Add-execute at the end of t_0 , and General Storage, $g_1 - g_{36}$, is added to the Accumulator. If there is an overflow, K is set to 1 again.

If $K = 0$ initially, the Add Conditional is converted to a Skip operation and the Accumulator is simply regenerated.

In either case ($K = 1$ or $K = 0$ initially), a_0 is reset to zero, and General Storage is unchanged.

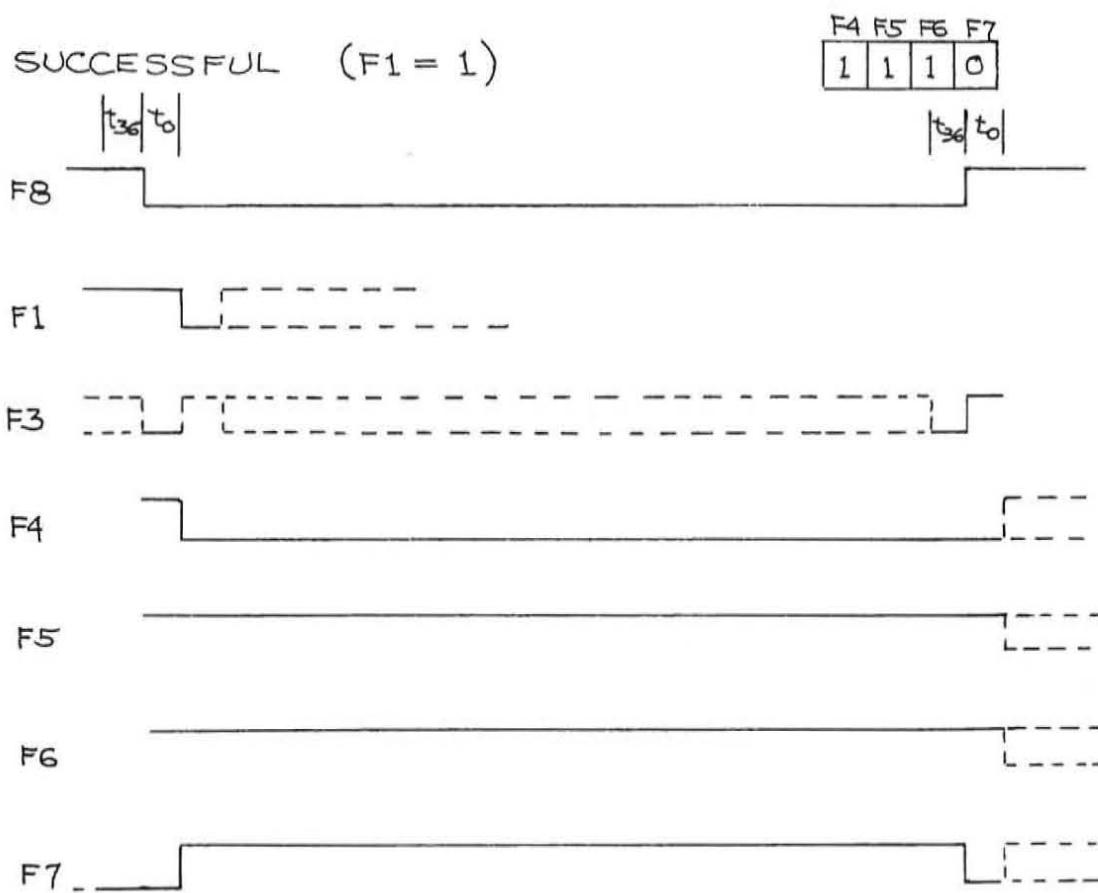
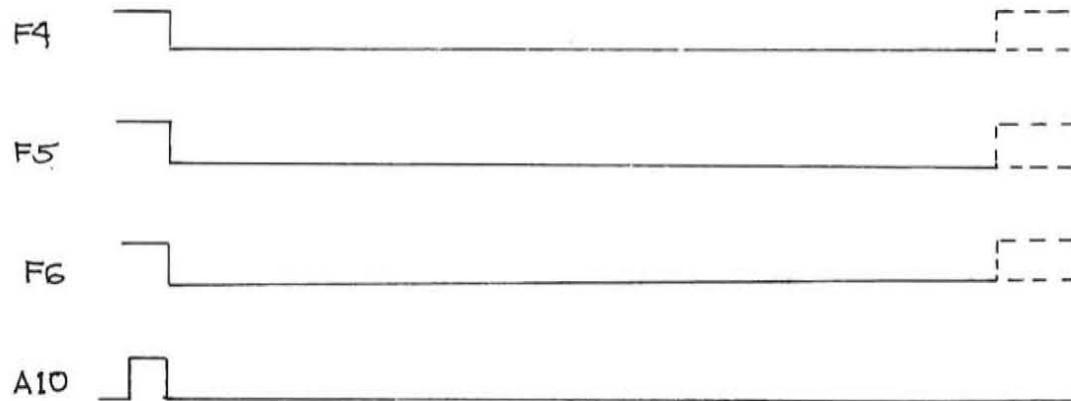
SEQUENCE:

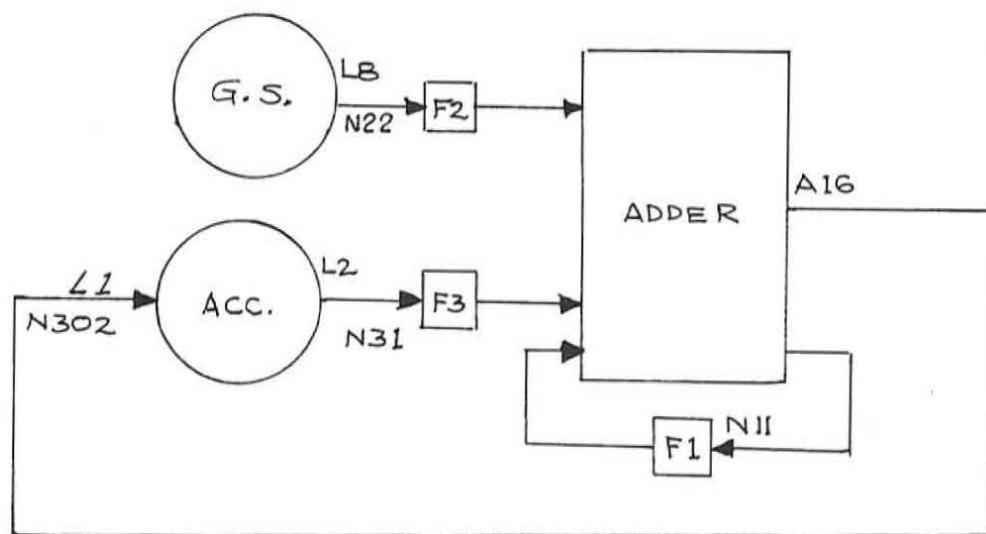
F8 (Successful Add Conditional-execute, $K = 1$).

- t_0
 - (1). F_4 in the Command Register is reset via N_{46} .
 - (2). Since F_1 is in the set state, F_7 in the Command Register is set via N_{72} .
 - (3). F_1 is reset via N_{16} .
- $t_1 - t_{36}$
 - (1). The code in the Command Register is now that of an Add instruction. The General Storage playback, L_8 , is gated via N_{22} to F_2 and the Accumulator playback, L_2 , is gated via N_{31} to F_3 . F_2 and F_3 form indirect inputs to the adder A_{16} (see Add).
 - (2). The adder output A_{16} is recorded in the Accumulator via N_{302} , L_1 .
 - (3). The carry logic set and reset gates for F_1 are N_{11} and N_{15} respectively.
- t_{36}
 - (1). F_7 in the Command Register is reset via N_{75} .
 - (2). F_3 is set via N_{31} .

F8 (Unsuccessful Add Conditional-execute).

- t_0
 - (1). Since F_1 is low, F_4 , F_5 , and F_6 in the Command Register are reset by N_{102} , A_{10} .
- $t_1 - t_{36}$
 - (1). The code in the Command Register is now that of a Skip instruction, and the Accumulator, $a_1 - a_{36}$, is regenerated via N_{31} , F_3 , N_{304} , L_1 (see Skip).
- t_{36}
 - (1). F_3 is set via N_{31} .

SUCCESSFUL ($F_1 = 1$)UNSUCCESSFUL ($F_1 = 0$)



II-4-20. Extract - Clear (1111).

The Extract and Clear commands, which utilize the same instruction code, are differentiated by the state of the K flip-flop F1. If K initially contains a zero, the number in the Accumulator, $a_1 - a_{36}$, is logically multiplied by the number in General Storage, $g_1 - g_{36}$, which is available during the execute word-time, and the result is recorded in the Accumulator. However, if K initially contains a 1, the Accumulator is cleared -- that is, the number is replaced by zeros. In either case, a zero is recorded in a_0 . General Storage remains unchanged and K is set to 1 at t_{36} .

The logical multiplication which takes place during the Extract execute time is a serial "anding" together of the Accumulator and General Storage playback outputs. The resultant AND gate output is recorded in the Accumulator. Since the inputs to an AND gate must be high simultaneously to produce a high output, the AND gate under consideration will have a high or 1 output only when a 1 occurs simultaneously in the corresponding bit positions of both the Accumulator and General Storage. For example:

| | | |
|-----------------|---|----------|
| General Storage | = | 11111010 |
| Accumulator | = | 10111101 |
| Extract Result | = | 10111000 |

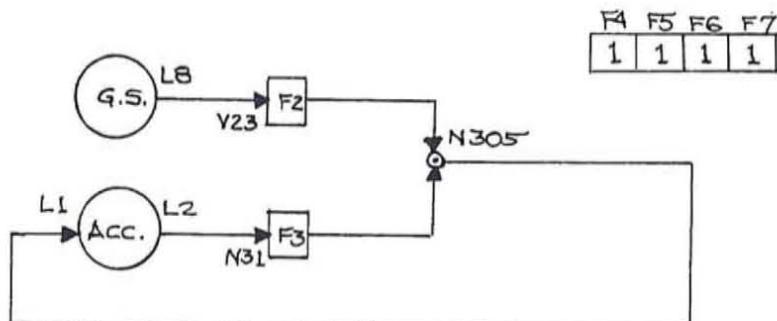
SEQUENCE:

F8 (Extract-execute, K = 0).

- t_0 (1). Since no logic is written for the Accumulator record, a zero is recorded in a_0 .
- $t_1 - t_{36}$ (1). General Storage playback is gated via V23, N22, F2 and Accumulator playback is gated via V31, N31, F3.
 - (2). F2 and F3 are "anded" together via A22 and N305 and the result is recorded in the Accumulator via L1.
- t_{36} (1). F1 is set via N14.
 - (2). F7 in the Command Register is reset via N75.
 - (3). F3 is set via N31.

F8 (Clear-execute, K = 1).

- $t_0 - t_{36}$ (1). Since K is high in this case, no logic is written for the entire Accumulator record and zeros are automatically recorded in $a_0 - a_{36}$.
- t_{36} (1). F1 is set via N14.



SIGNAL GLOSSARY

| <u>SIGNAL</u> | <u>SOURCE</u> | <u>PAGE</u> | <u>FUNCTION</u> | <u>EQUIVALENCE</u> | <u>LOCATION</u> |
|---------------|---------------|-------------|-----------------|--------------------|-----------------|
| A1 | NI | II-63 | | F1 | A7-TR |
| A1 | NI | II-63 | | F1 | A7-TR |
| A3 | NI | II-65 | | F3 | A7-TR |
| A4 | NI | II-66 | | F4 | A7-MR |
| A4 | NI | II-66 | | F4 | A7-MR |
| A5 | NI | II-67 | | F5 | A7-MR |
| A5 | NI | II-67 | | F5 | A7-MR |
| A6 | NI | II-67 | | F6 | A7-BR |
| A7 | NI | II-68 | | F7 | A7-BR |
| A8 | NI | II-68 | | F8 | A7-TR |
| A8 | NI | II-68 | | F8 | A7-TR |
| A10 | NI | II-70 | Clear | | A3-BL |
| A11 | NI | II-70 | | F6, F7 | A3-BR |
| A12 | NI | II-70 | | F5, F6 | A3-BR |
| A13 | NI | II-70 | | A8, T36 | A3-BR |
| A14 | NI | II-71 | | F4, F9, A8 | A3-BL |
| A15 | NI | II-71 | | | A2-TL |
| A16 | NI | II-71 | | | A2-BL |
| A17 | NI | II-72 | | F9, F8, T36 | A2-BL |
| A18 | NI | II-72 | | F2, F8 | A3-BL |
| A19 | NI | II-72 | | A4, A5 | A2-ML |
| A20 | NI | II-72 | | F3, F2 | A7-MR |
| A21 | NI | II-72 | | F3, F2 | A7-BR |
| A22 | NI | II-72 | | F2, A3 | A7-BR |
| A23 | NI | II-72 | | F2, A3 | A7-BR |
| A24 | NI | II-72 | | A4, A5, F6, F8 | A2-ML |
| A25 | NI | II-73 | Input/Output | F8, A4, F5, F7, F6 | A4-BL |
| A26 | NI | II-73 | | L2, F2 | A3-TR |
| A27 | NI | II-73 | Change Track | A5, F6, F7, A8, A4 | A5-TL |

SIGNAL GLOSSARY

| <u>SIGNAL</u> | <u>SOURCE</u> | <u>PAGE</u> | <u>FUNCTION</u> | <u>EQUIVALENCE</u> | <u>LOCATION</u> |
|---------------|---------------|-------------|---------------------|--------------------|-----------------|
| A28 | NI | II-73 | Change Track | A5,F6,F7,A8,A4 | A5-TL |
| A29 | NI | II-83 | | T29 | B-TL |
| C40 | | | | | |
| D.S. | DIA | II-81 | Device Select | E3, A10 | C2-BR |
| E3 | NI | II-74 | Output | M24,V31,A25 | A1-MR |
| E6 | NIN | II-81 | Output Busy | | C2-TR |
| E7 | NI | II-74 | Output | L2,A25,A10,M24 | A1-TR |
| F1 | FF | II-63 | | | A7-TL |
| F1 | FF | II-63 | | | A7-TL |
| F2 | FF | II-64 | | | A7-ML |
| F2 | FF | II-64 | | | A7-ML |
| F3 | FF | II-65 | Retiming | | A7-TL |
| F3 | FF | II-65 | " | | A7-TL |
| F4 | FF | II-66 | Shift Register | | A7-ML |
| F4 | FF | II-66 | " " | | A7-ML |
| F5 | FF | II-67 | " " | | A7-ML |
| F5 | FF | II-67 | " " | | A7-ML |
| F6 | FF | II-67 | " " | | A7-BL |
| F6 | FF | II-67 | " " | | A7-BL |
| F7 | FF | II-68 | " " | | A7-BL |
| F7 | FF | II-68 | " " | | A7-BL |
| F8 | FF | II-68 | | | A7-TL |
| F8 | FF | II-68 | | | A7-TL |
| F9 | FF | II-69 | | | A7-BR |
| F9 | FF | II-69 | | | A7-BR |
| F120 | DI | II-64 | Input to F2 (set) | | A3-TL |
| F120 | DI | II-64 | Input to F2 (reset) | | A3-TL |
| F130 | DI | II-65 | Input to F3 (set) | | A3-ML |
| F130 | DI | II-65 | Input to F3 (reset) | | A3-ML |

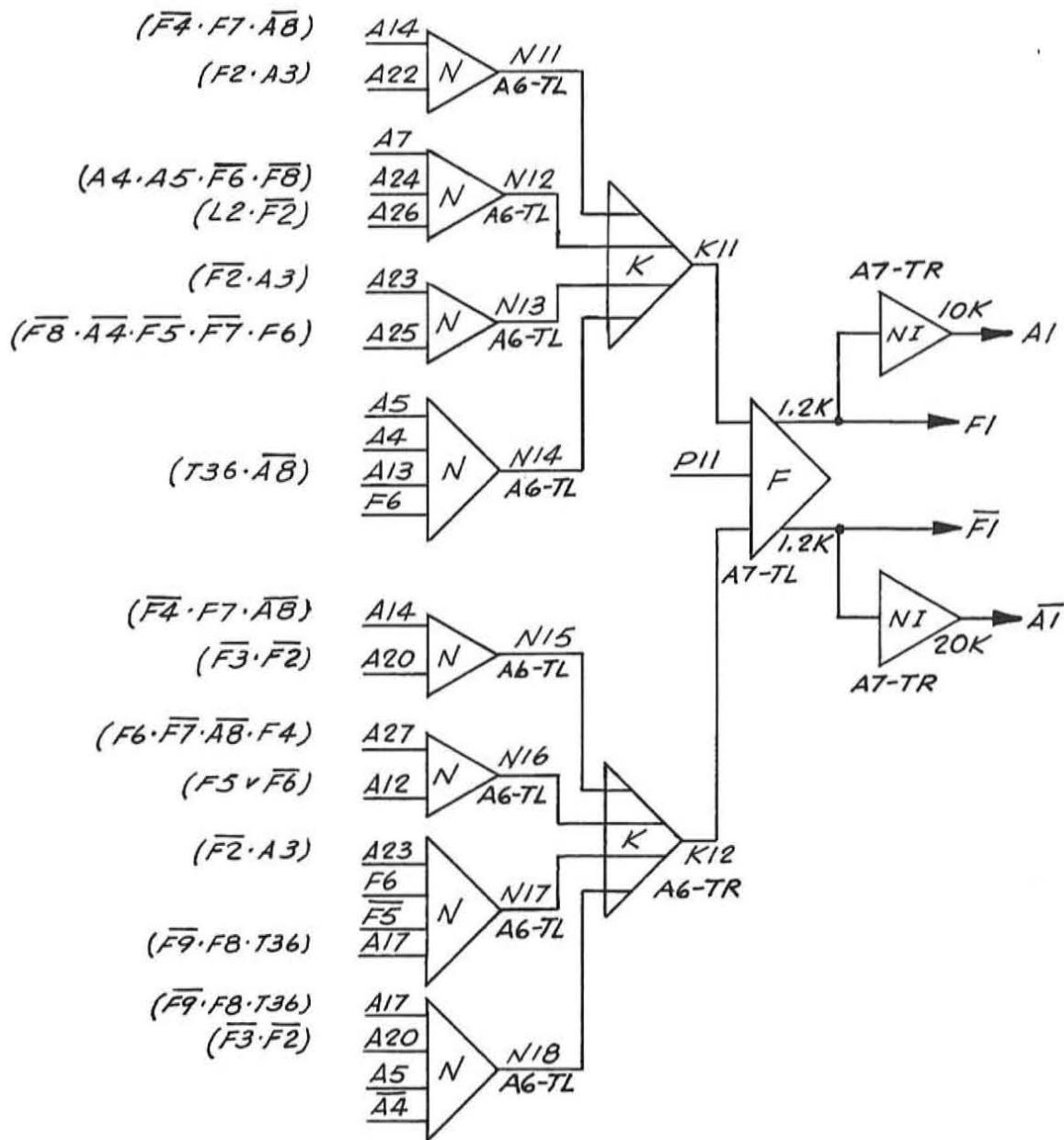
SIGNAL GLOSSARY

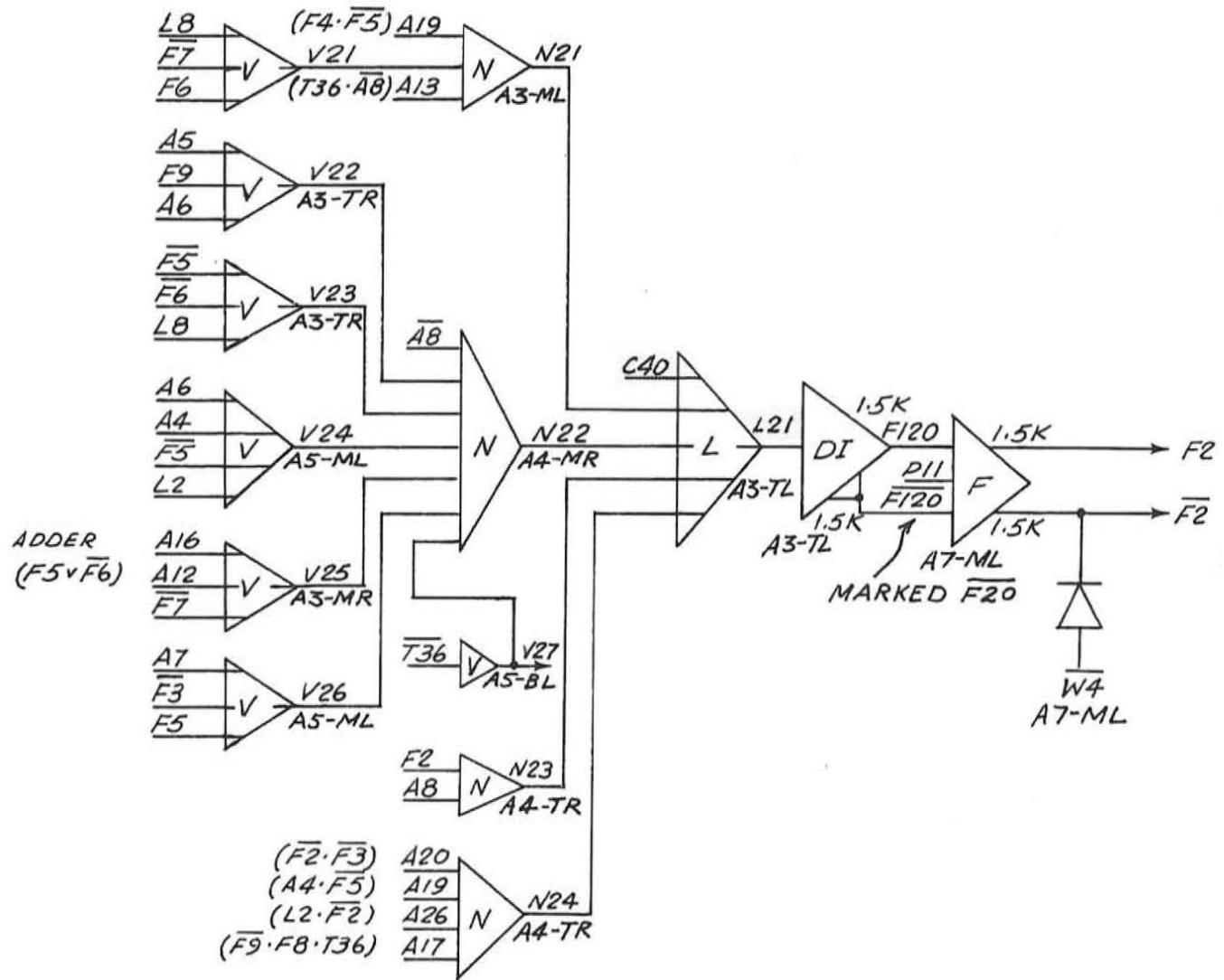
II-61

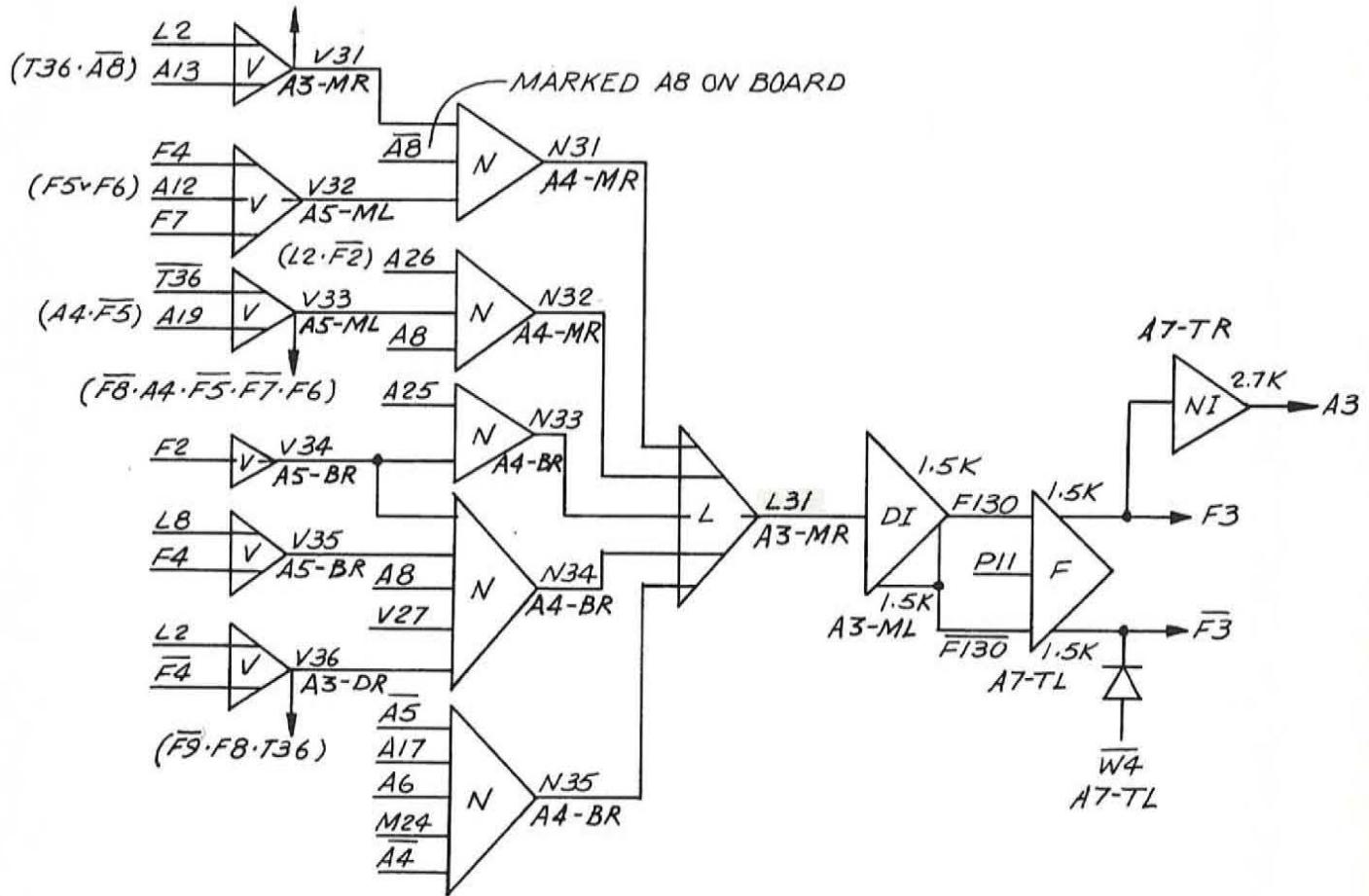
| <u>SIGNAL</u> | <u>SOURCE</u> | <u>PAGE</u> | <u>FUNCTION</u> | <u>EQUIVALENCE</u> | <u>LOCATION</u> |
|---------------|---------------|-------------|--------------------------------|--------------------|-----------------|
| FPB1 | | II-82 | Force Track Sel. | | Loader |
| ID1 | IB | II-81 | Input Device Sel. #1 | | |
| ID1 | FC | II-81 | " " " #1 | | |
| ID2 | IB | II-81 | " " " #2 | | |
| ID2 | FC | II-81 | " " " #2 | | |
| ID3 | IB | II-81 | " " " #3 | | |
| ID3 | FC | II-81 | " " " #3 | | |
| L1 | DI | II-75 | Fast Access Record Logic | | |
| L1 | DI | II-75 | " " " | | |
| L2 | FP | II-83 | Fast Access Playbk. | | |
| L2 | FP | II-83 | " " " | | |
| L5 | AND | II-76 | Gen'l Storage Record Logic "0" | | A2-TL |
| L7 | AND | II-76 | " " "1" | | A2-TL |
| L8 | FP | II-82 | Gen'l Storage Playbk. | | B - |
| L8 | FP | II-82 | " " " | | B - |
| M1 | | II-76 | Track #1 Selection | | B -TR |
| M2 | | II-76 | Track #2 Selection | | B -TR |
| M3 | | II-76 | Track #3 Selection | | B -TR |
| M4 | | II-76 | Track #4 Selection | | B -TR |
| M13 | NIA | II-76 | | M1, M3 | A2-TL |
| M14 | NIA | II-76 | | M2, M4 | A3-TL |
| NI | FB | II-77 | Output Buffer FF | | C2-BR |
| N2 | FB | II-77 | " " " | | C2-BR |
| N3 | FB | II-77 | " " " | | C2-BR |
| N4 | FB | II-77 | " " " | | C2-BR |
| N5 | FB | II-77 | " " " | | C2-MR |
| N6 | FB | II-78 | " " " | | C2-MR |
| N7 | FB | II-78 | " " " | | C2-MR |
| N8 | FB | II-78 | " " " | | C2-MR |
| NC | NC | II-78 | Output Common | E7 | C2-BR |

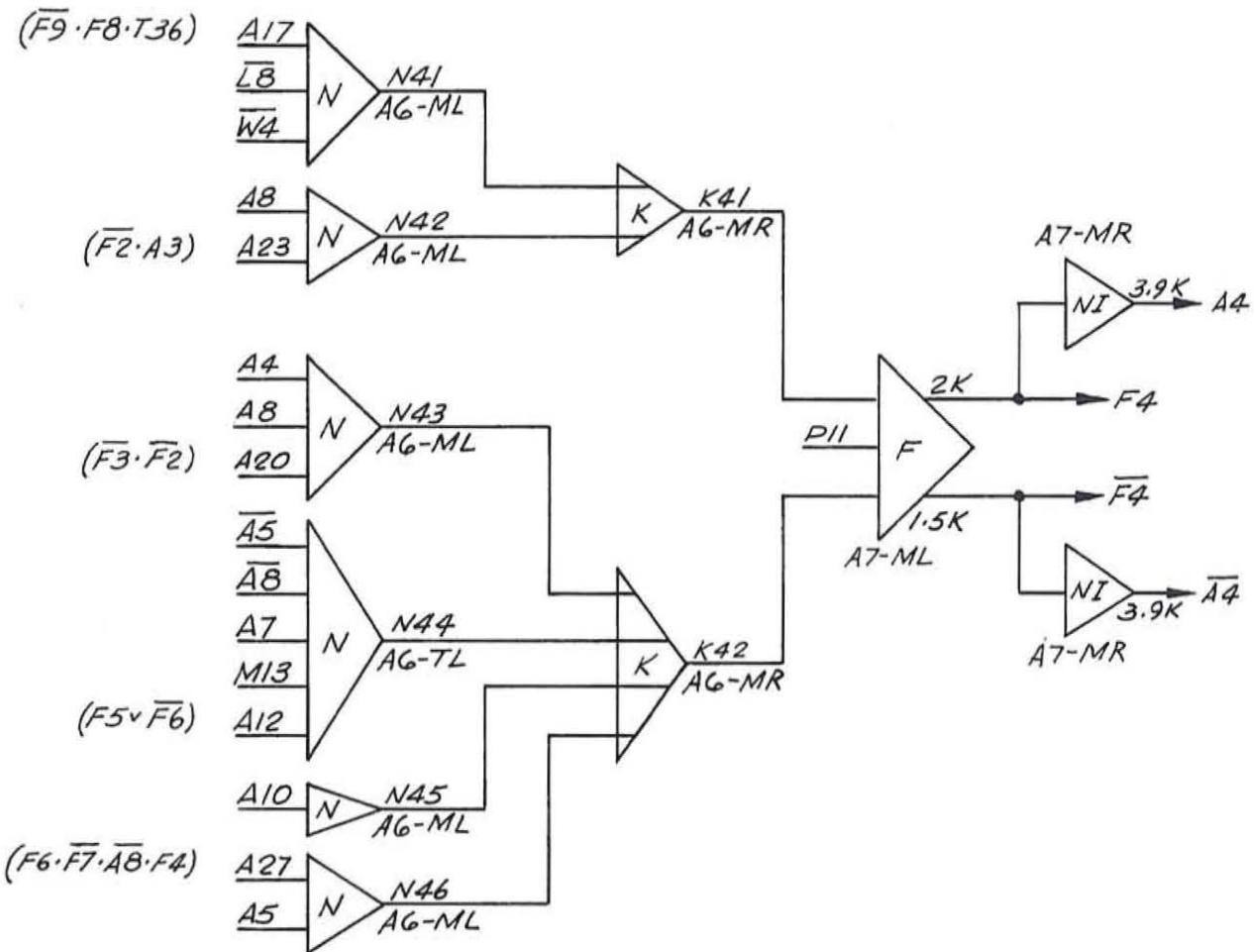
SIGNAL GLOSSARY

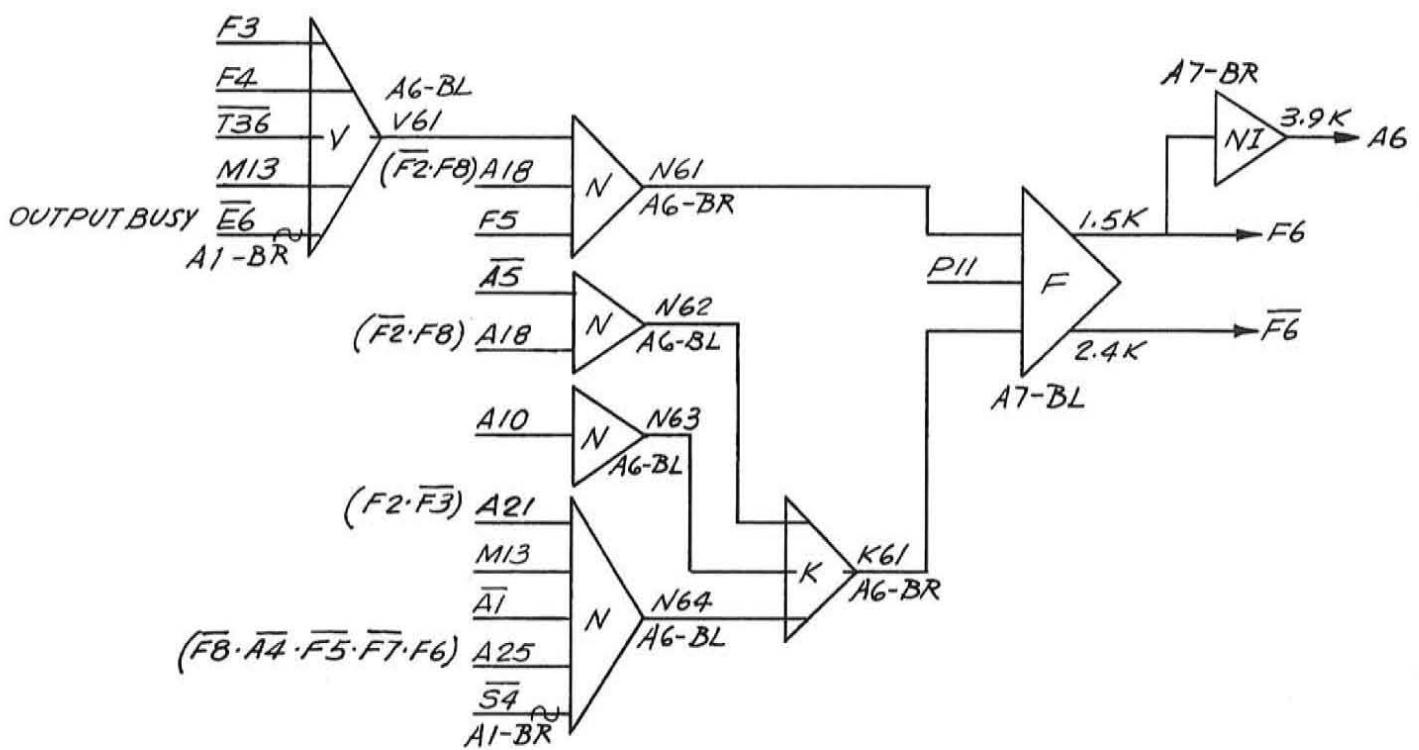
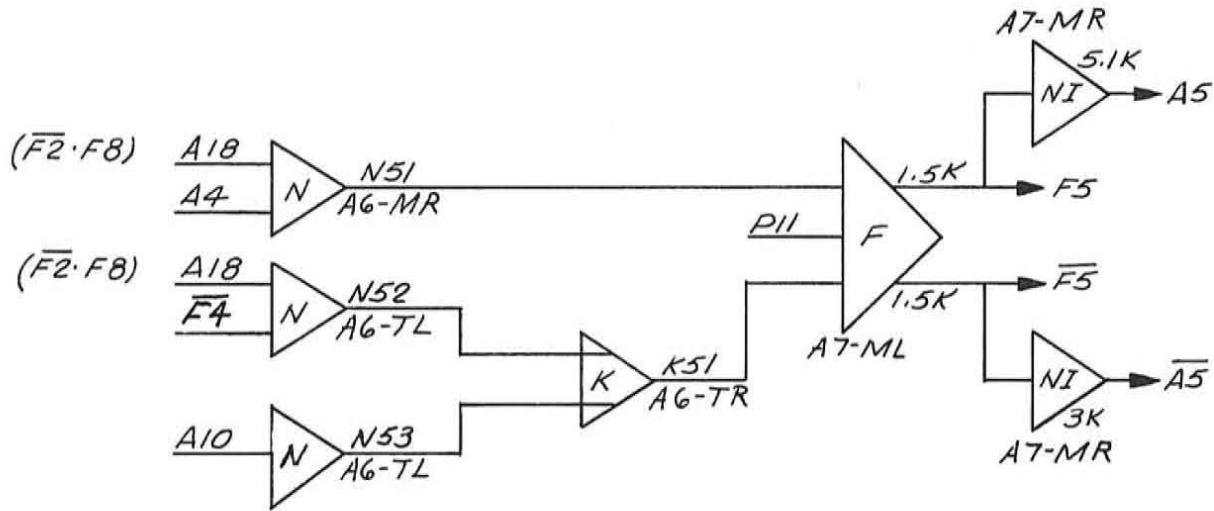
| <u>SIGNAL</u> | <u>SOURCE</u> | <u>PAGE</u> | <u>FUNCTION</u> | <u>EQUIVALENCE</u> | <u>LOCATION</u> |
|---------------|---------------|-------------|-----------------------------------|--------------------|-----------------|
| OD1 | FC | II-81 | Output Device #1 Select Memory | | C2-TL |
| OD2 | FC | II-81 | " " #2 | | C2-TL |
| OD3 | FC | II-81 | " " #3 | | C2-TL |
| P1 | CG | II-83 | Master Clock | | B -ML |
| P1 | CG | II-83 | " " | | B -ML |
| S0 | NIN | II-80 | Input Clear | A25,A10,M13 | A1-BR |
| S1 | SC | II-80 | Input Clear Signal | S0 | C3-BL |
| S2 | DIB | II-79 | Input Net | | C3-BL |
| S4 | AA | II-80 | Input Busy | | C3-BL |
| S11 | NA | II-80 | Input Clear Dev. #1 | | C1-BR |
| S14 | Device | II-80 | Device #1 Busy | | C1-BR |
| S21 | NA | II-80 | Input Clear Dev. #2 | | C1-BR |
| S24 | Device | II-80 | Device #2 Busy | | C1-BR |
| S31 | NA | II-80 | Input Clear Dev. #3 | | C1-BR |
| S34 | Device | II-80 | Device #3 Busy | | C1-BR |
| T2 | | II-76 | Track Selection | | |
| T2A | | II-76 | " " | | |
| T2B | | II-76 | " " | | |
| T29 | FP | II-83 | Sector | | B -ML |
| T29 | FP | II-83 | Sector | | B -ML |
| T36 | FA | II-83 | End of Word | | B -TL |
| T36 | FA | II-83 | End of Word | | B -TL |
| VG | VG | | | | |
| W1 | | II-82 | Monitor (line volt.) | | B -TL |
| W3 | | II-82 | Set Sequencing (Sw) | | B -TM |
| W4 | | II-82 | | | B -TR |
| W4 | | II-82 | Block Gen'l Storage Record | | B -TR |
| W5 | | II-82 | Set Sequencing (Sw) | | B -TL |
| W6 | | II-82 | Block Fast Access Record | | B -TR |
| W40 | | II-81 | Clear Device Memory | | C2-BL |

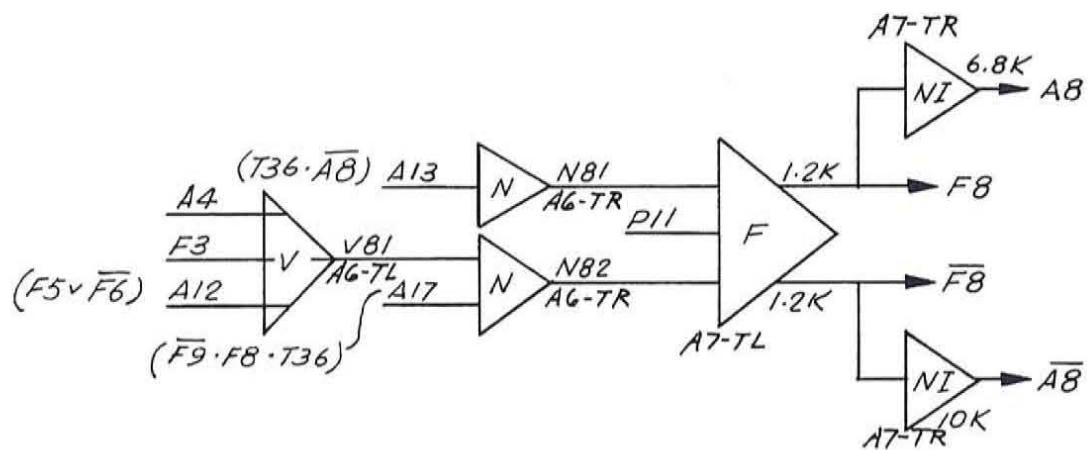
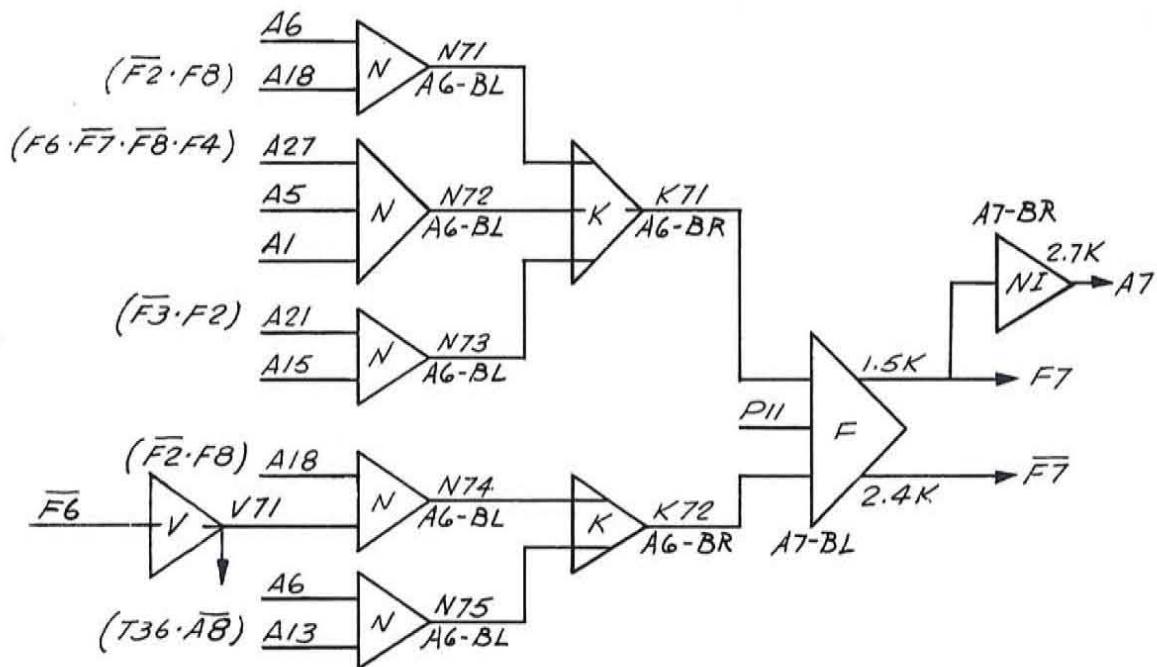


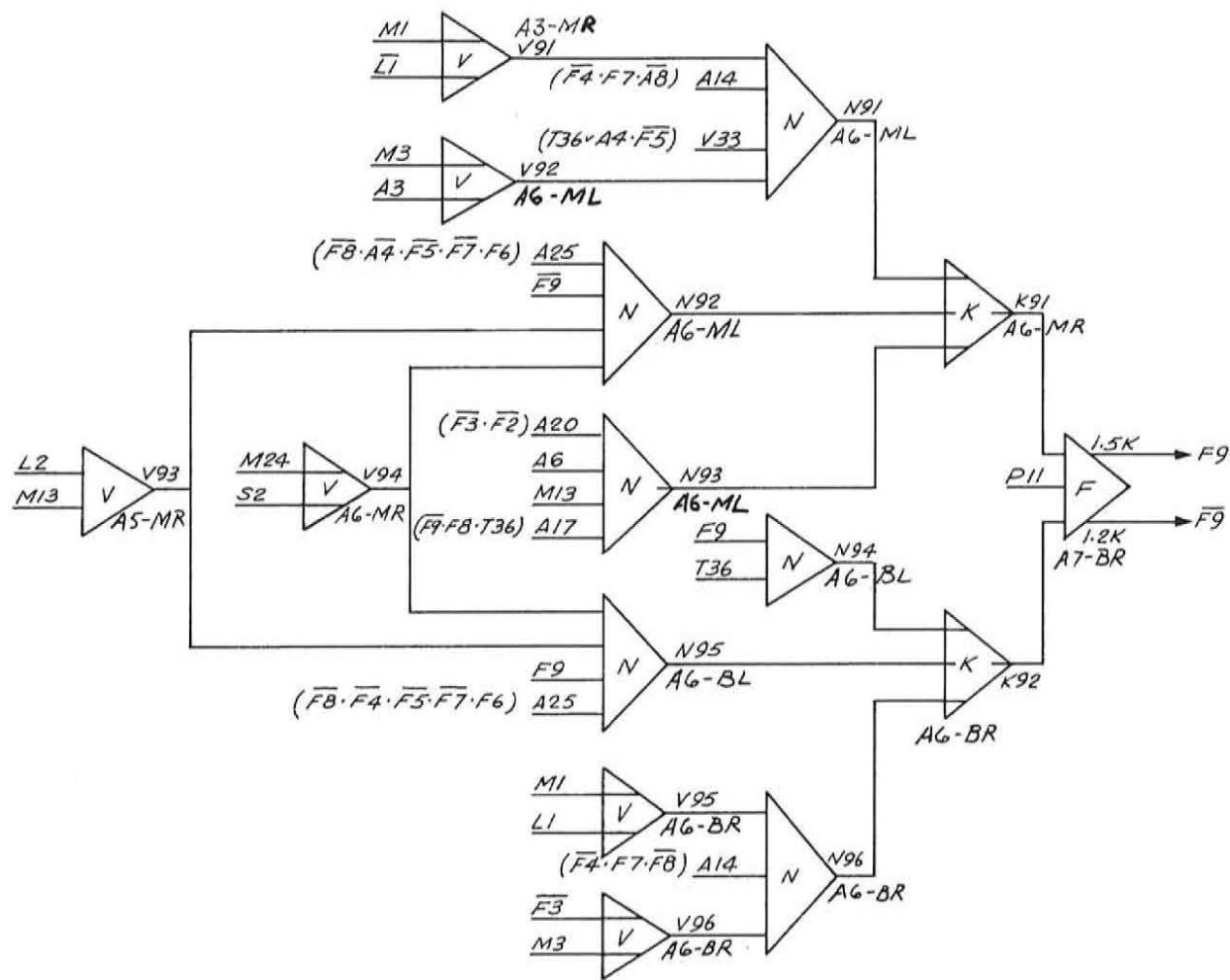


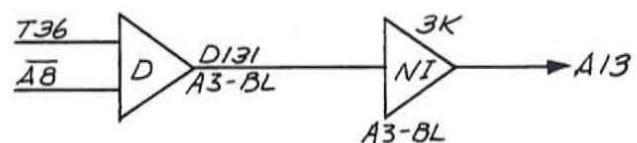
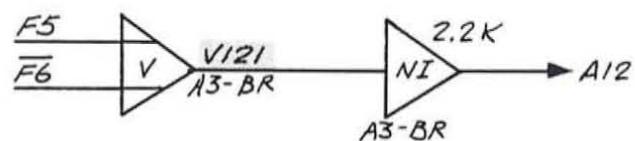
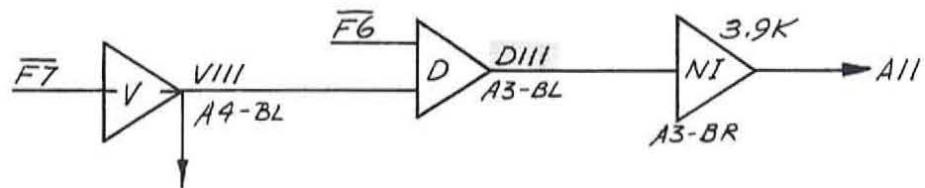
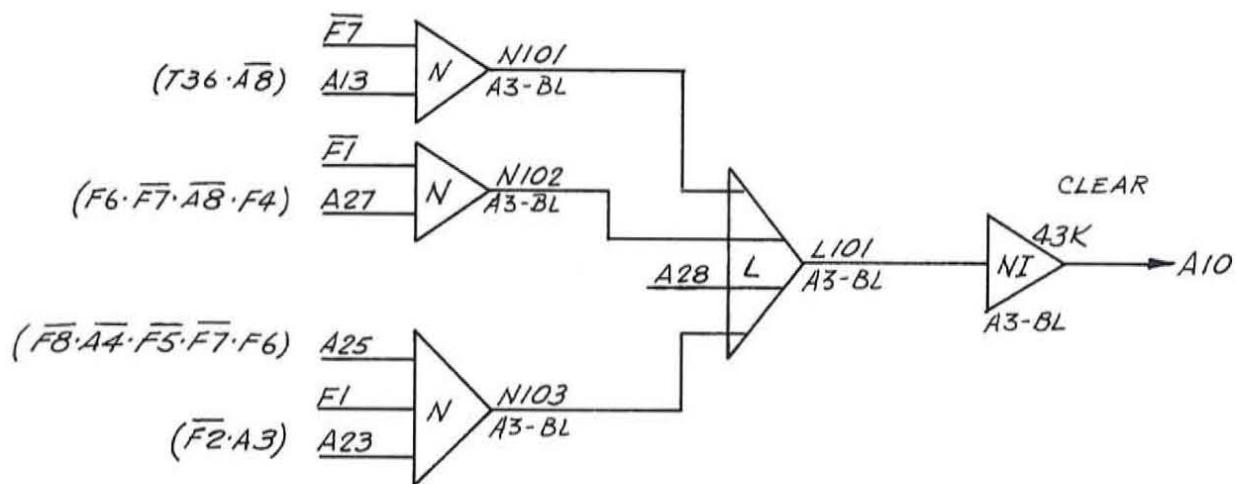


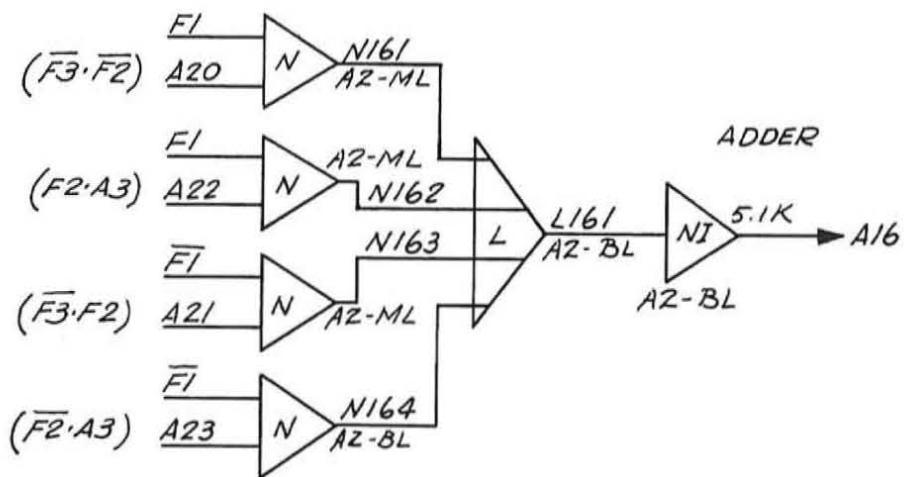
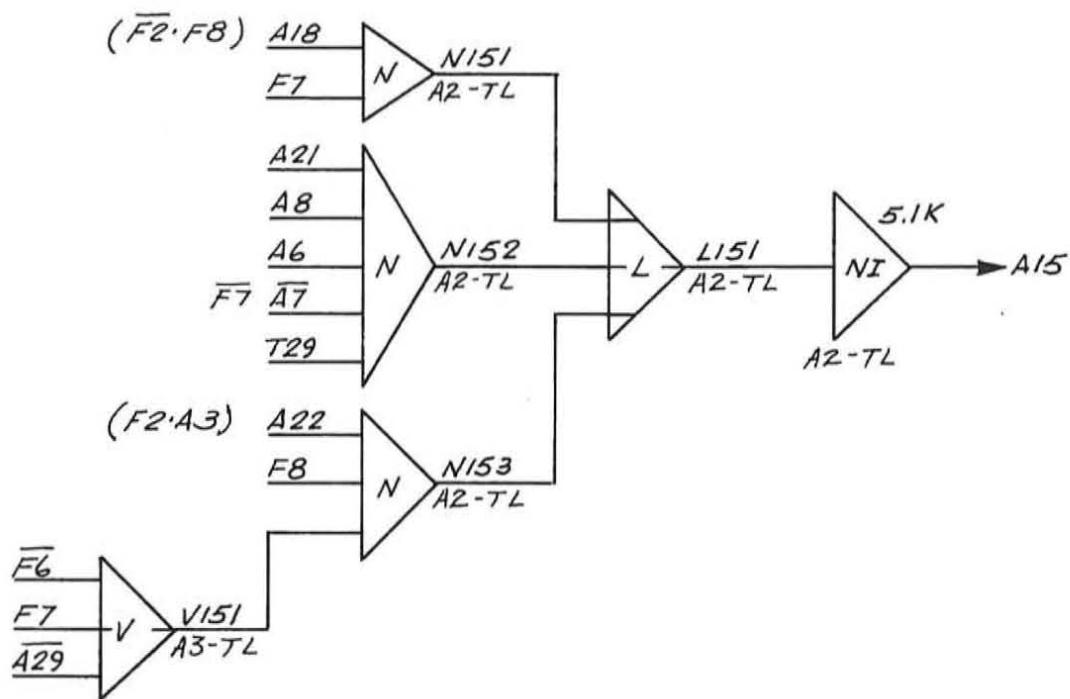


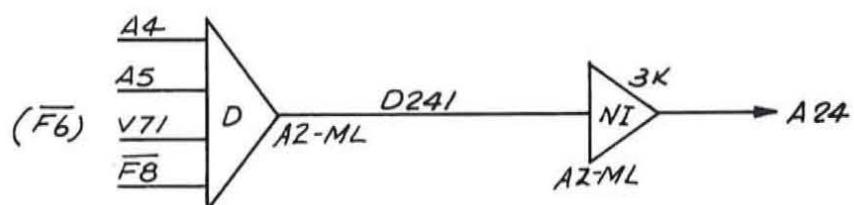
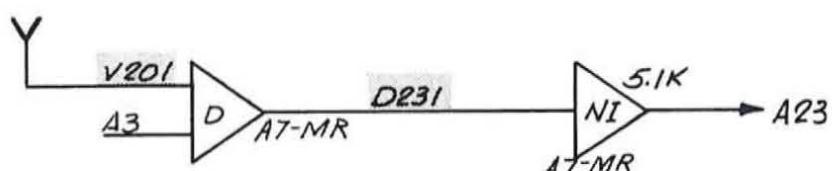
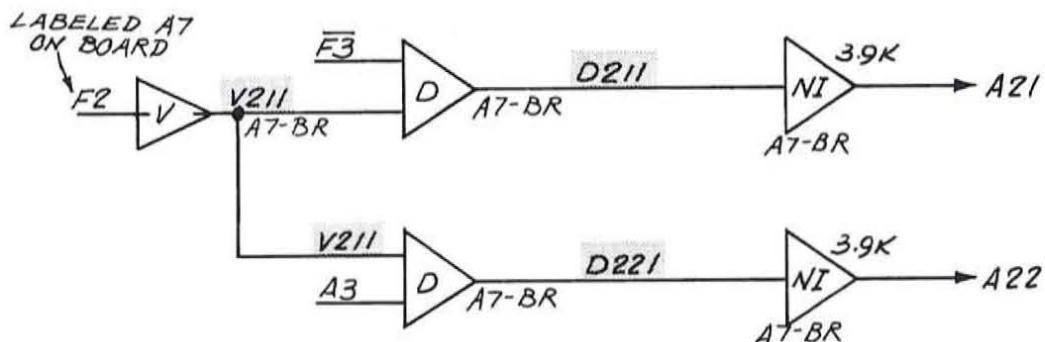
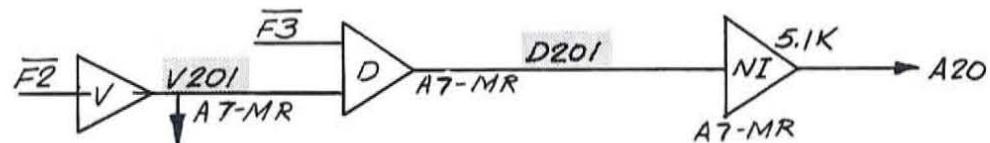
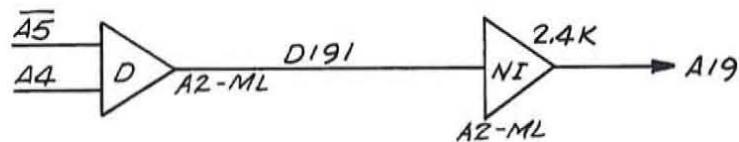
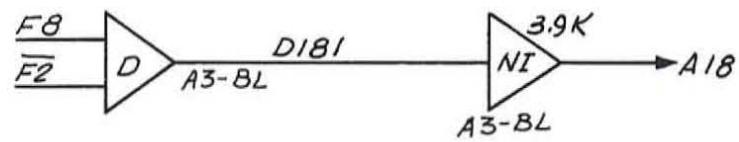
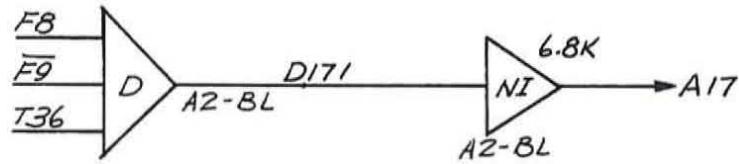


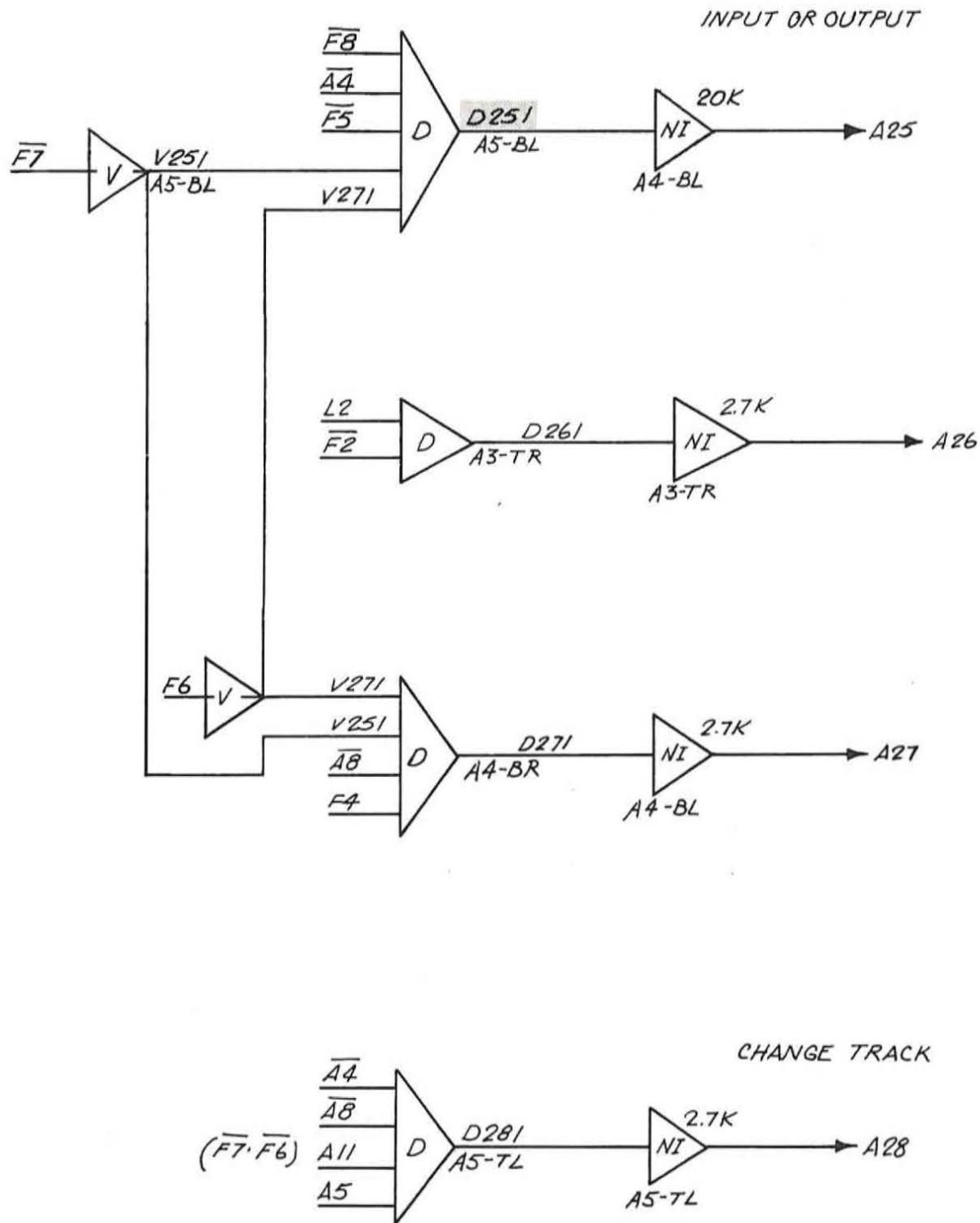


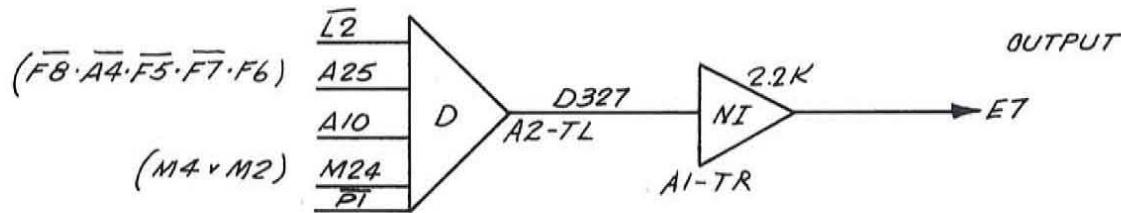
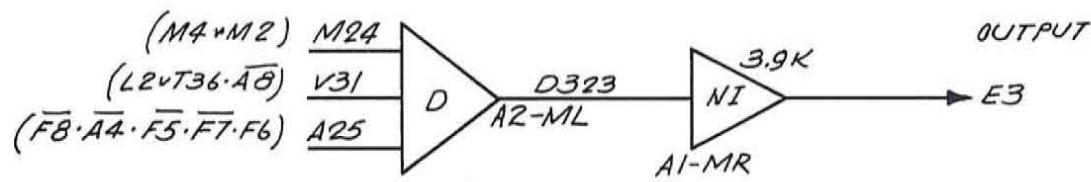




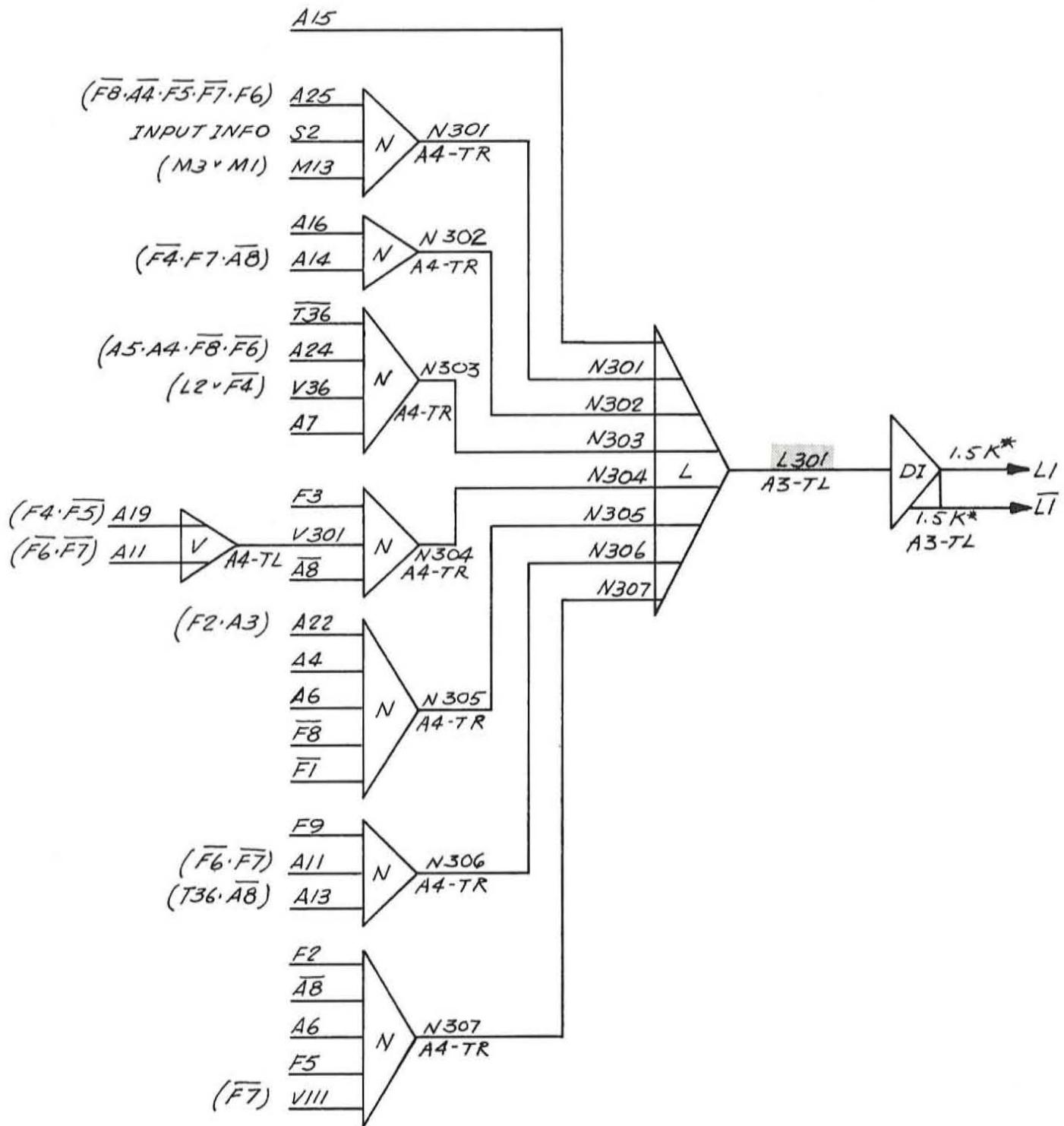




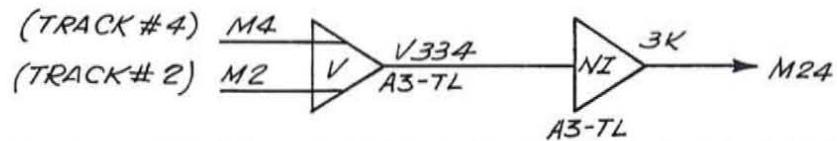
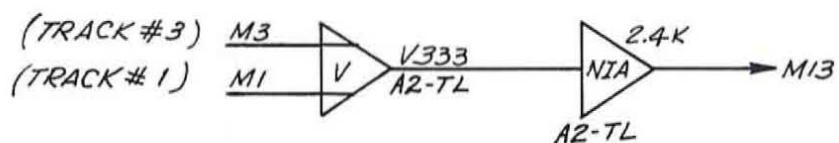
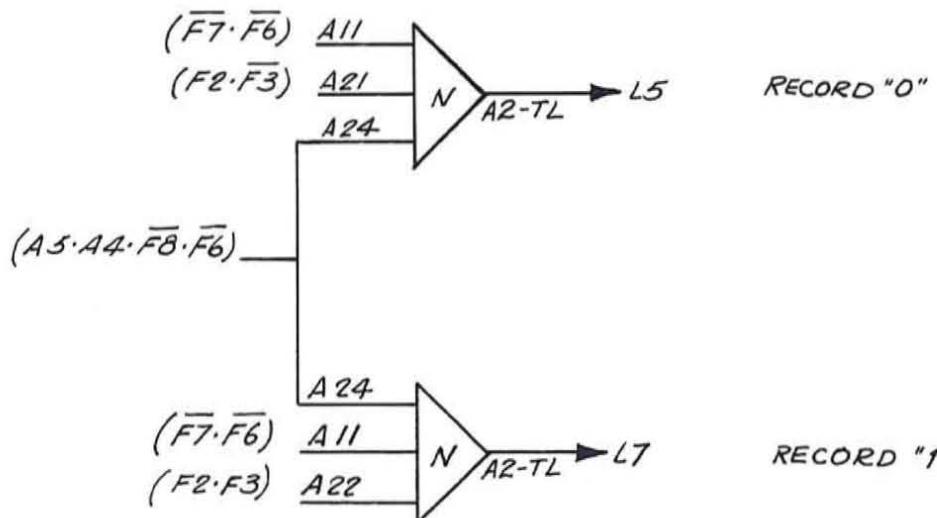




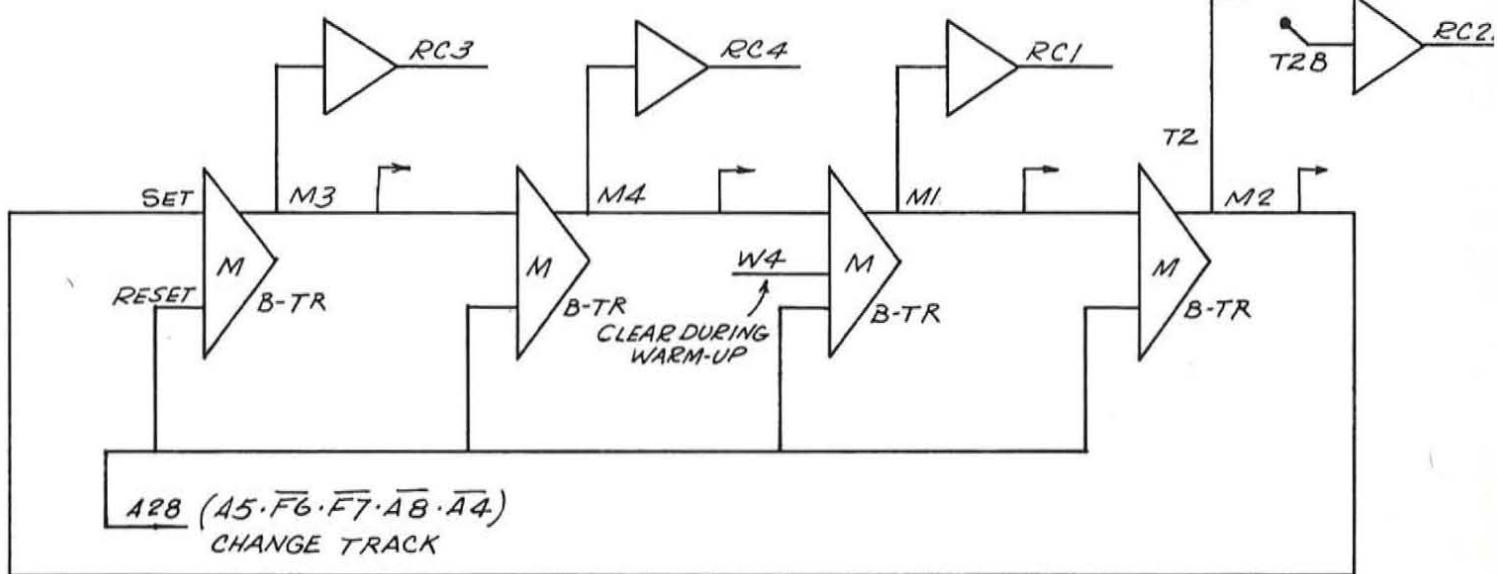
FAST ACCESS RECORD LOGIC



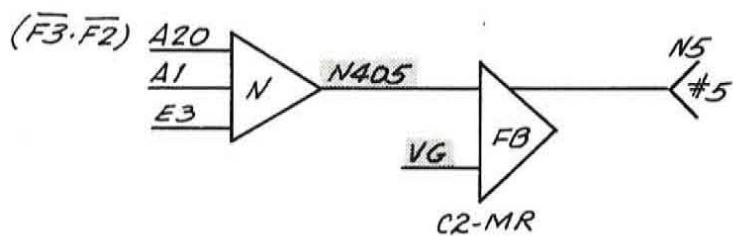
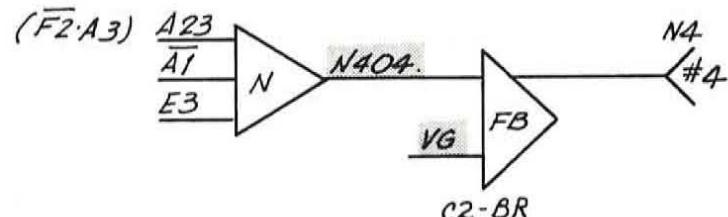
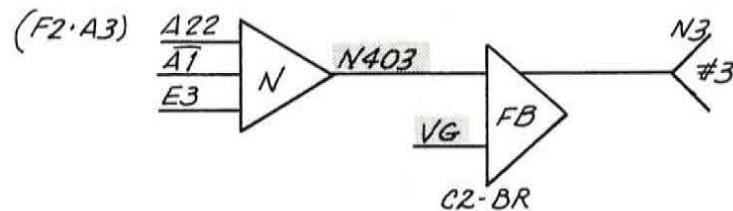
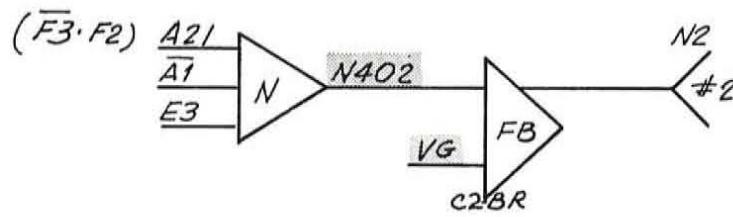
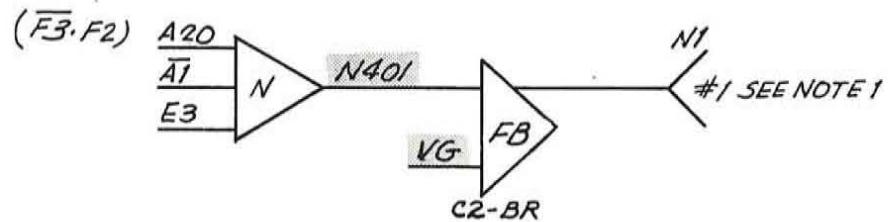
*LOAD RESISTORS LOCATED IN RECORD CIRCUIT ON "B" BOARD



TRACK SELECTION RING COUNTER

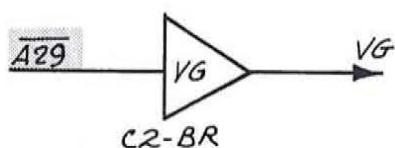
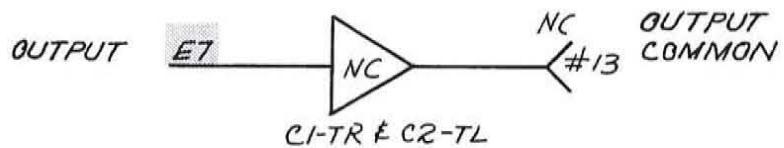
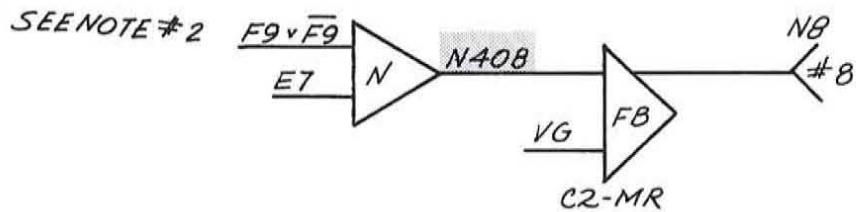
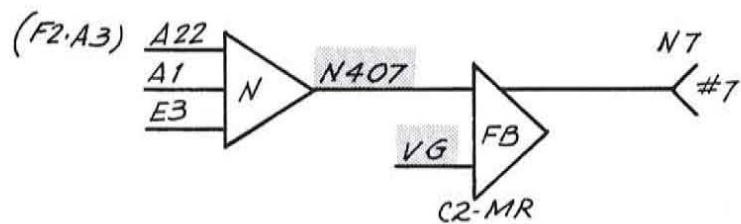
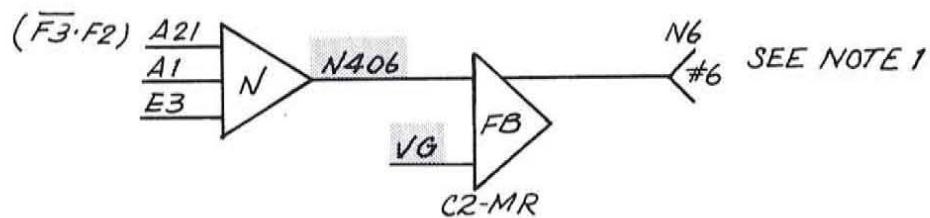


$$E3 = F6 \cdot \overline{F7} \cdot F5 \cdot \overline{A4} \cdot \overline{F8} / (AB \cdot T36) \cdot L2 \cdot M2 \cdot M4$$



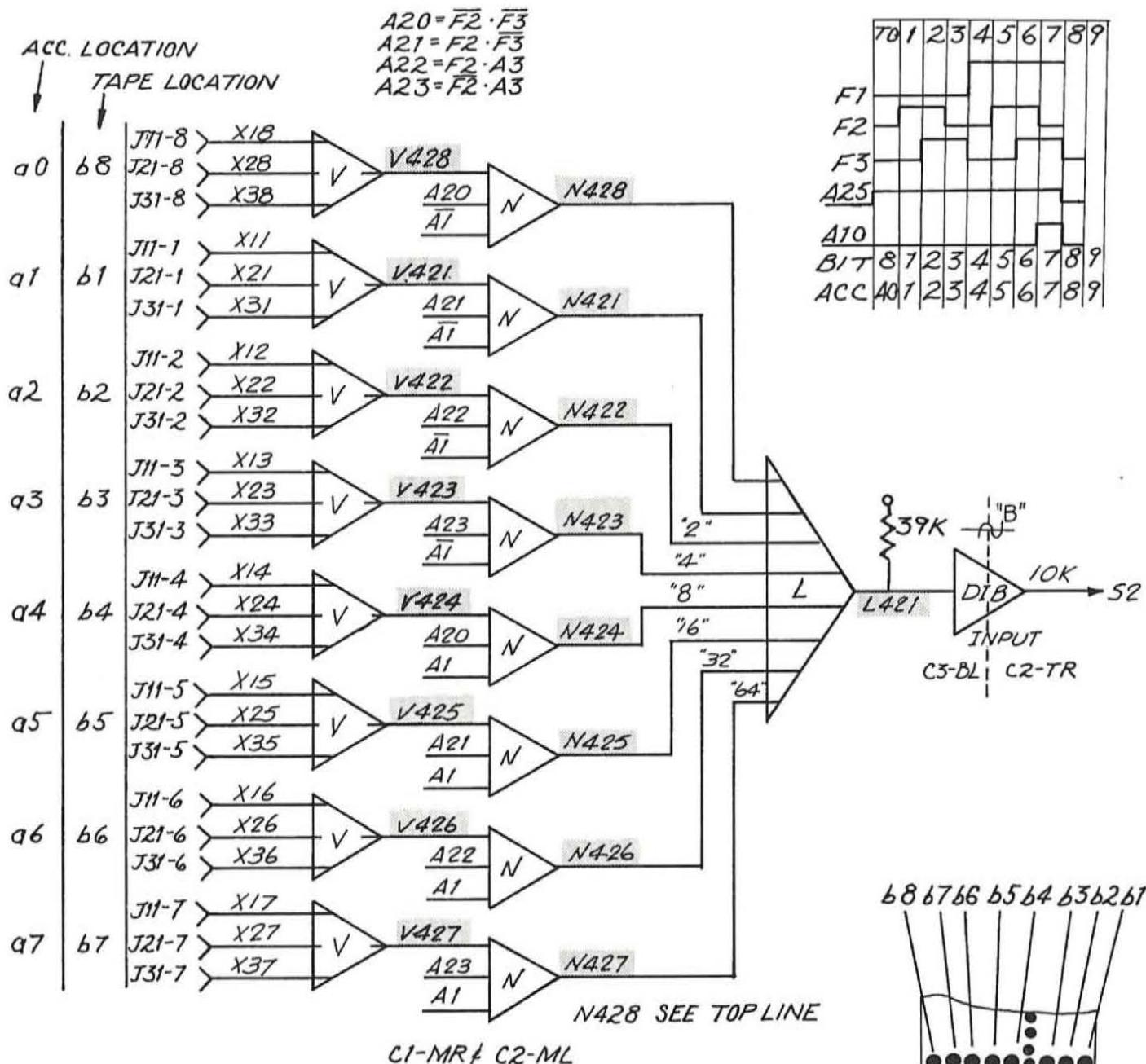
NOTE 1 PIN NUMBER OF OUTPUT CONNECTOR J12, J22, AND J32

C.S.B. #XII-501
 OUTPUT
 $E3 = F6 \cdot \overline{F7} \cdot \overline{F5} \cdot A4 \cdot \overline{F8} \cdot \overline{(A8 \cdot T36) \cdot L2} \cdot M2 \cdot M4$

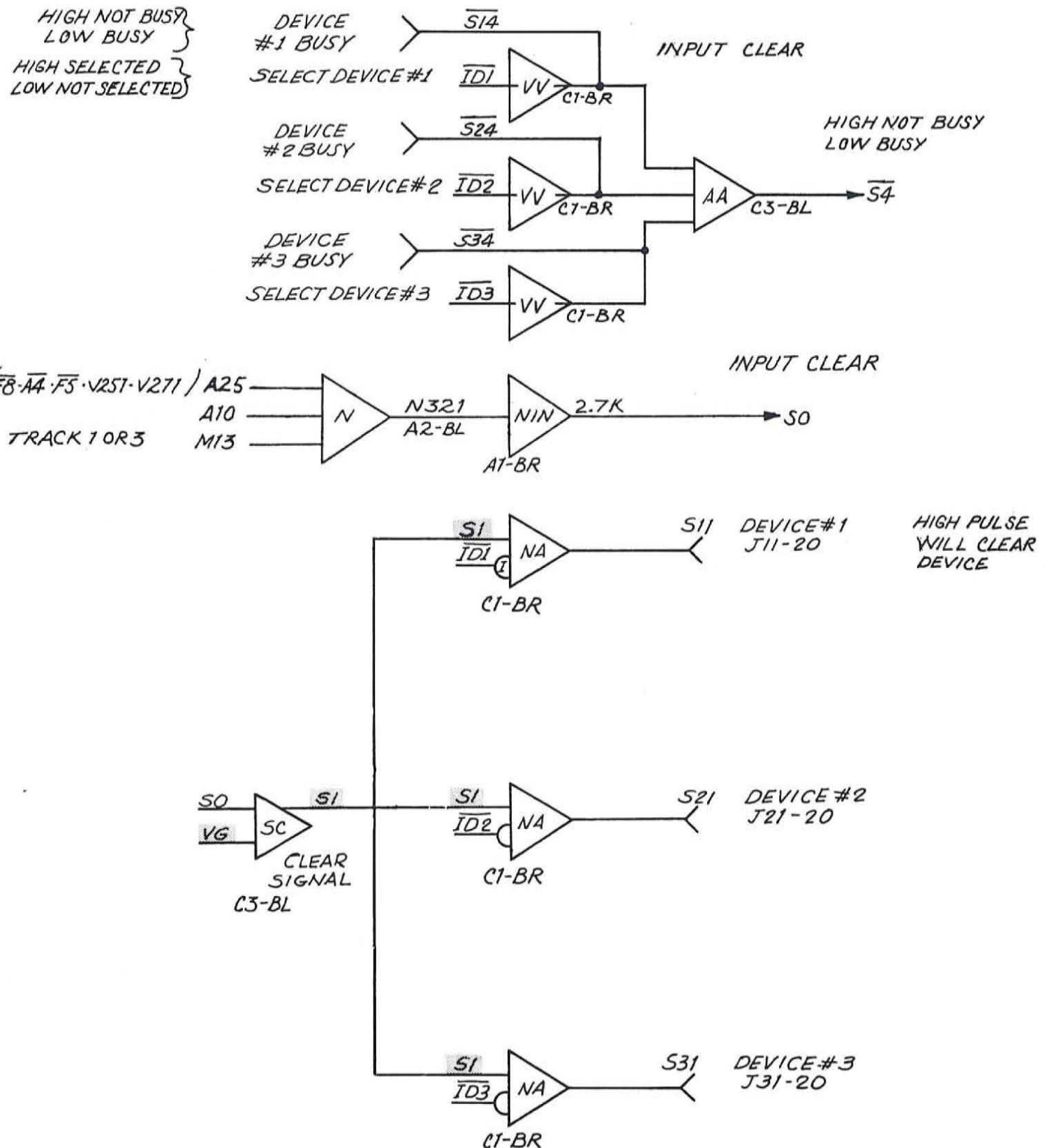


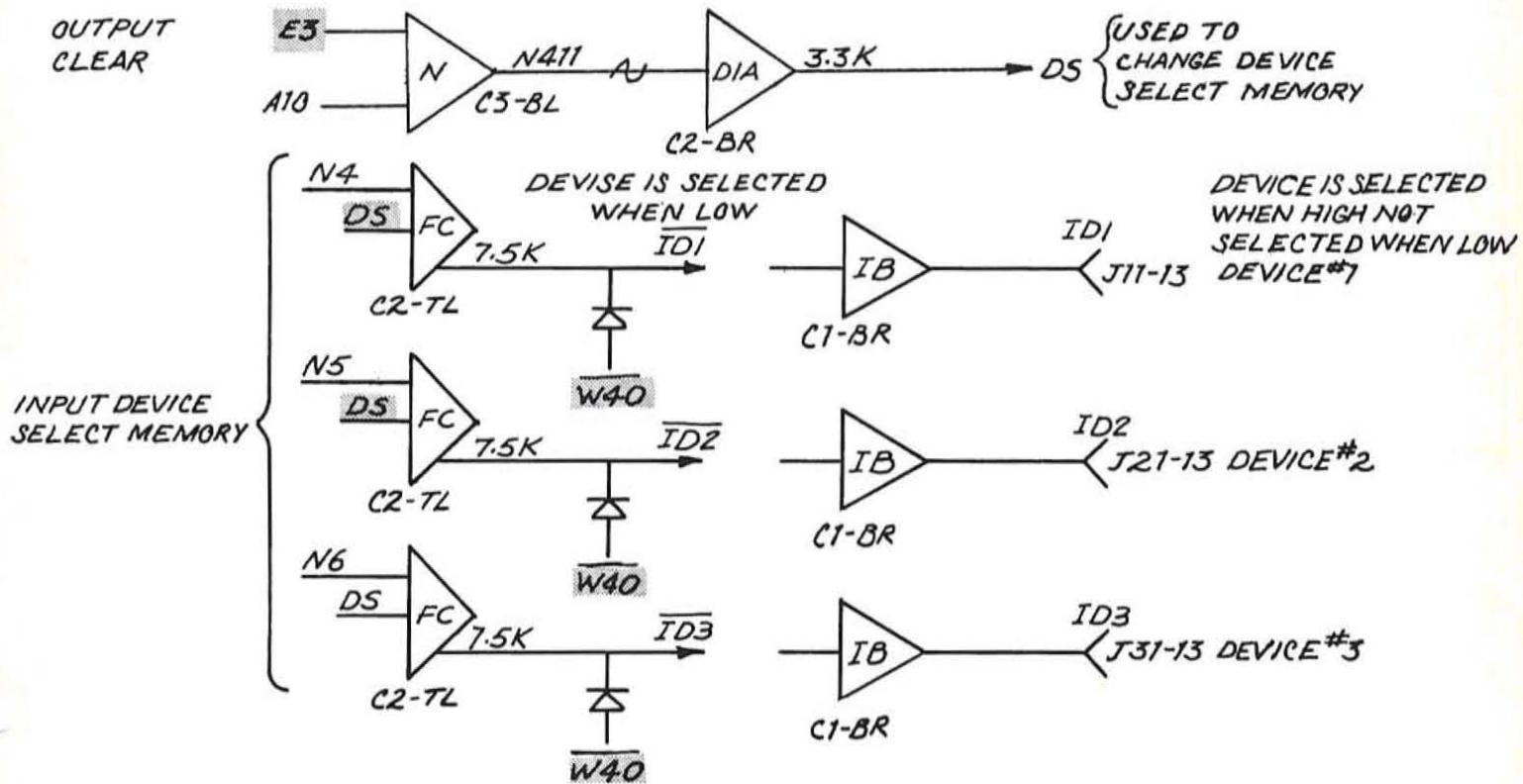
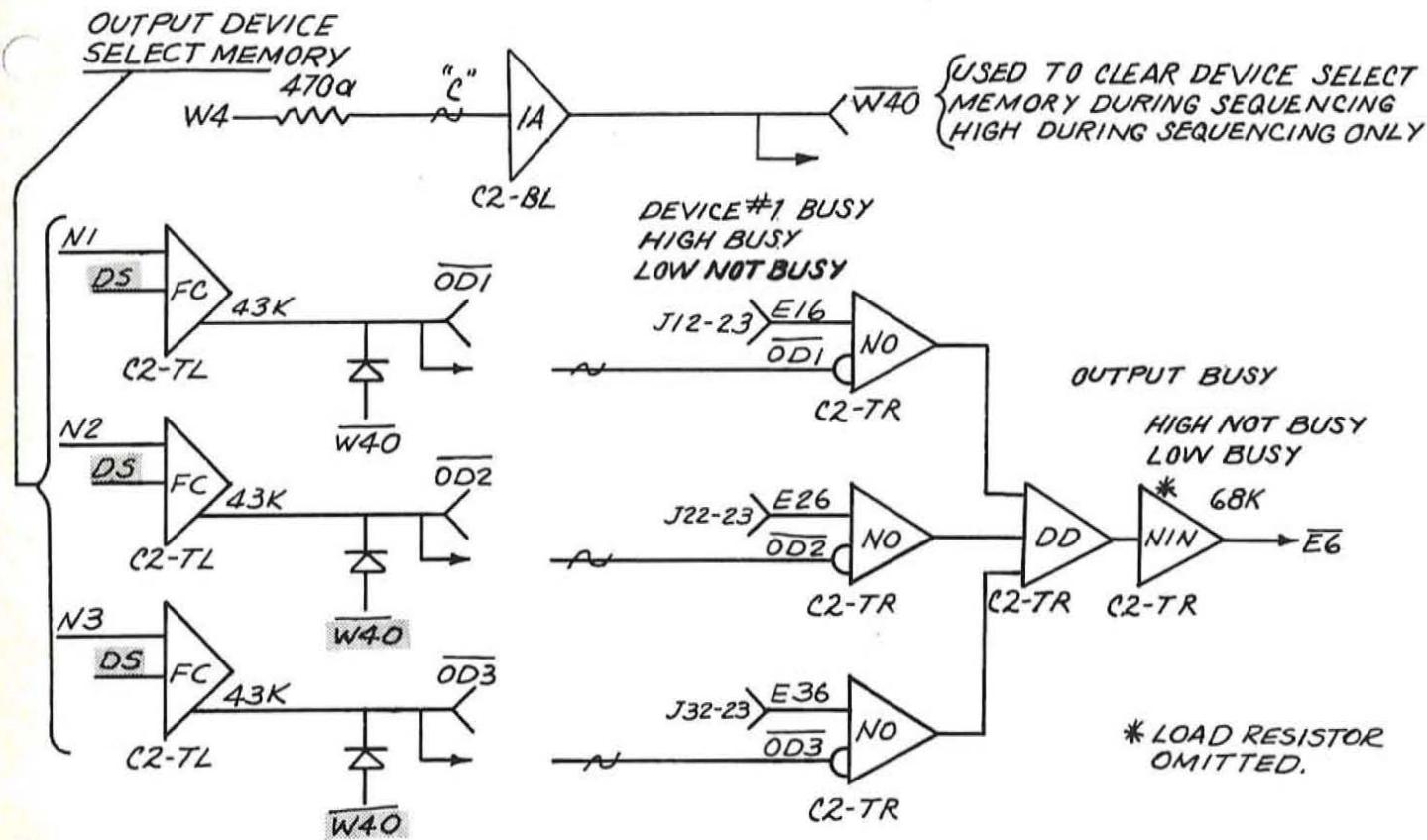
NOTE 1 PIN NUMBER OF OUTPUT CONNECTOR J12, J22, AND J32
 NOTE 2 CONNECT F9 FOR EVEN PARITY AND $\overline{F9}$ FOR ODD PARITY

INPUT NET

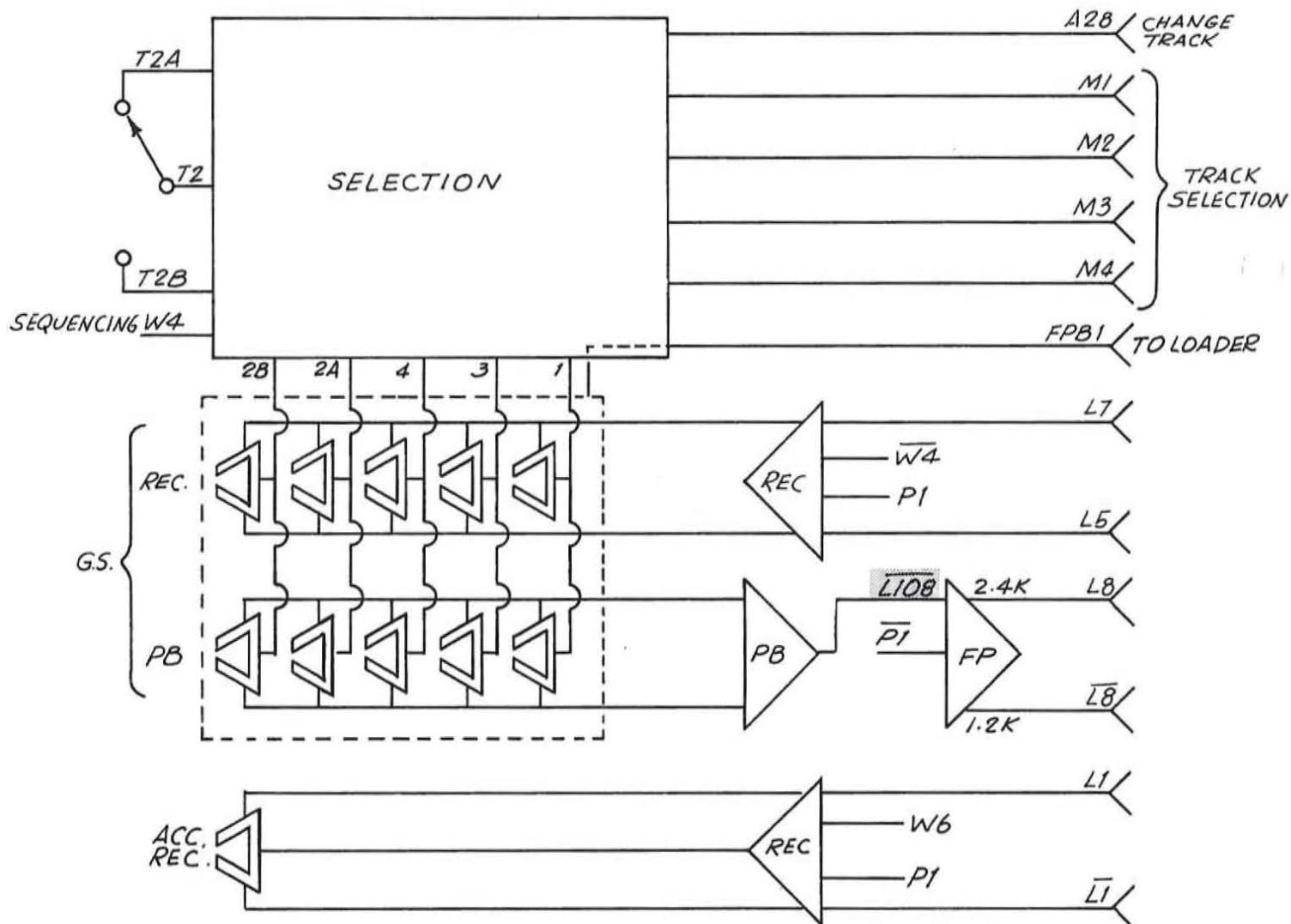
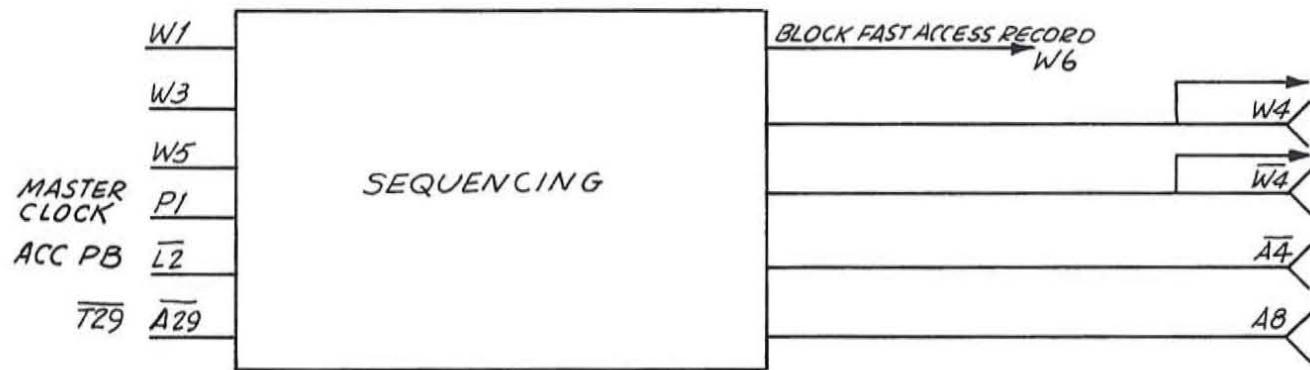


DEVICE #1 (X11 THROUGH X18)
 " #2 (X21 " X28)
 " #3 (X31 " X38)

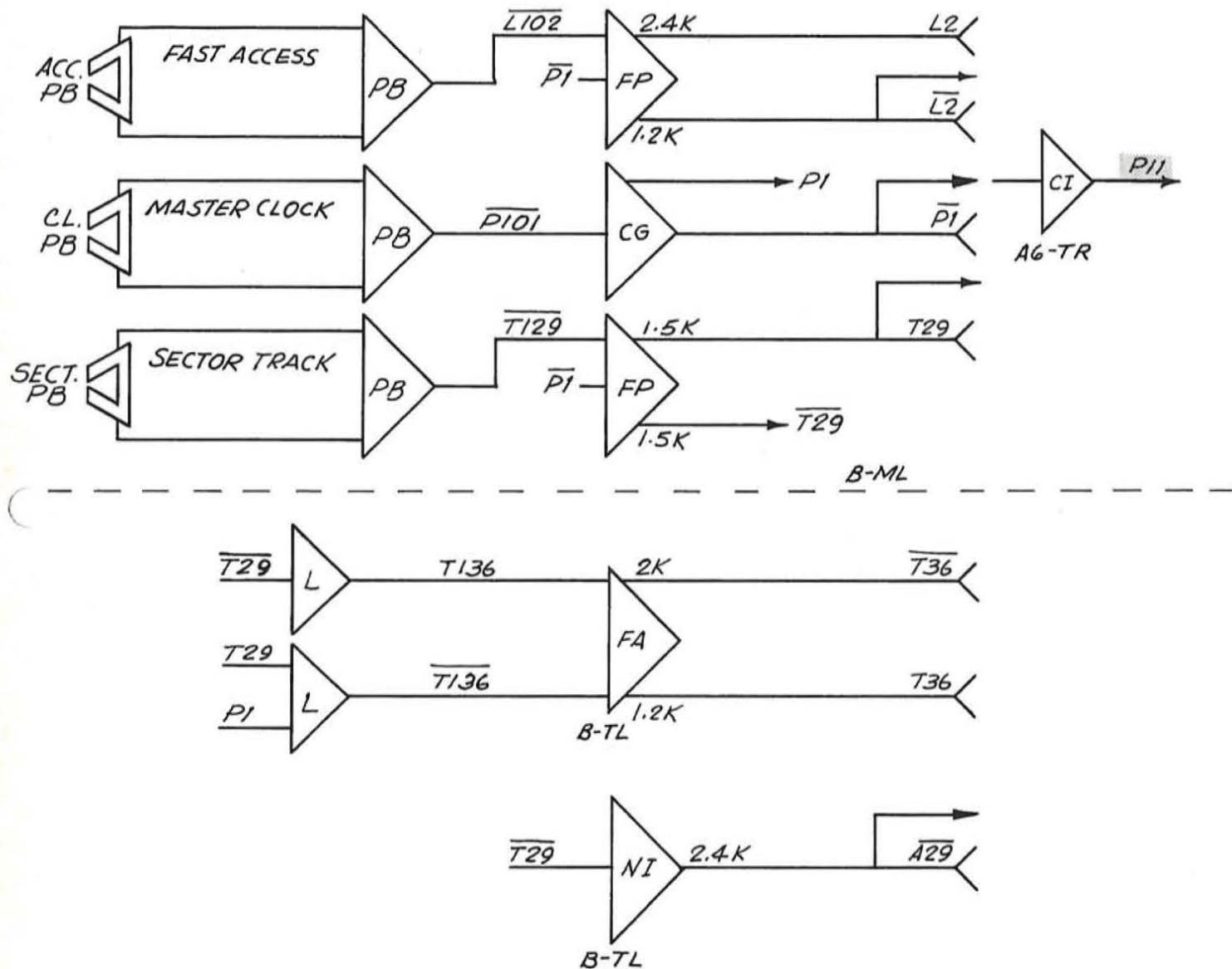


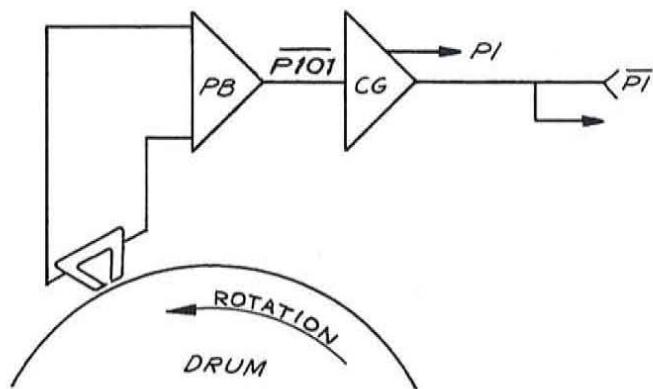
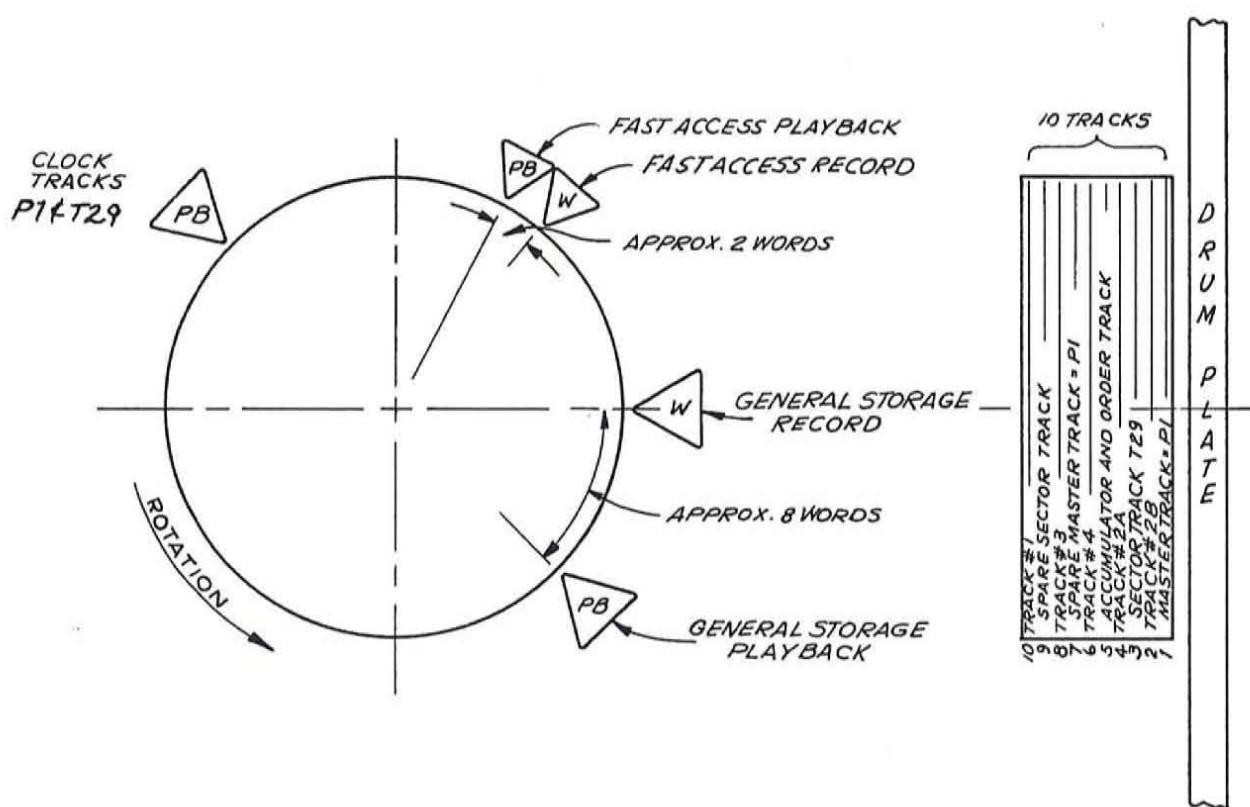


MEMORY



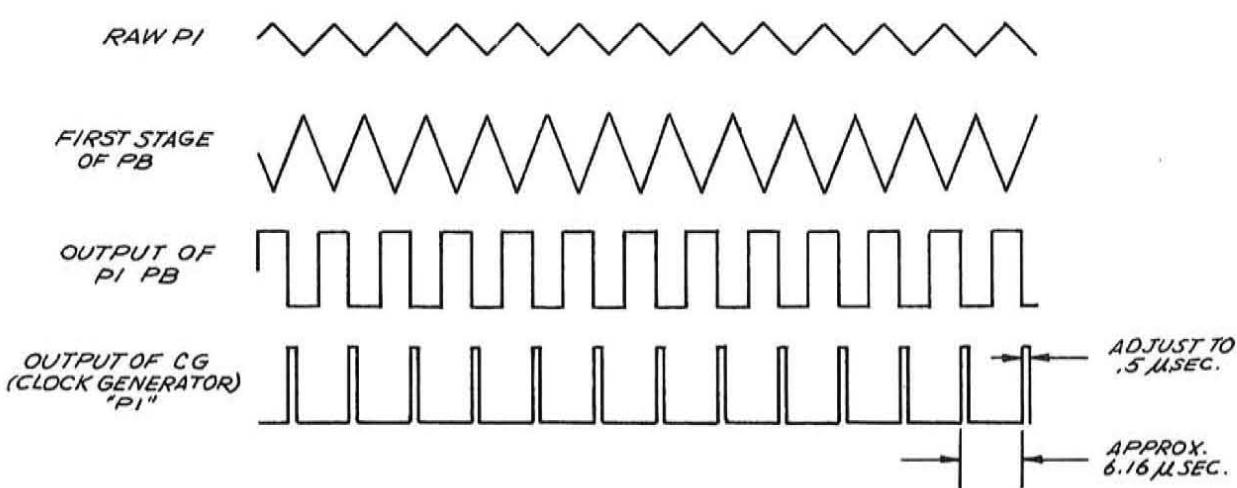
MEMORY

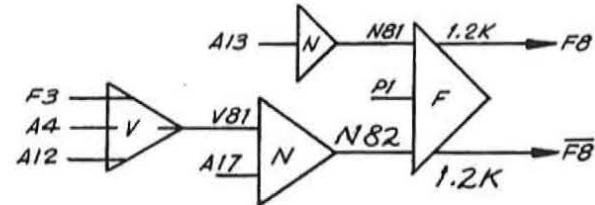
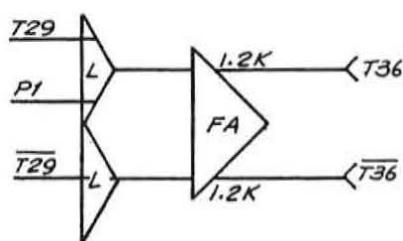
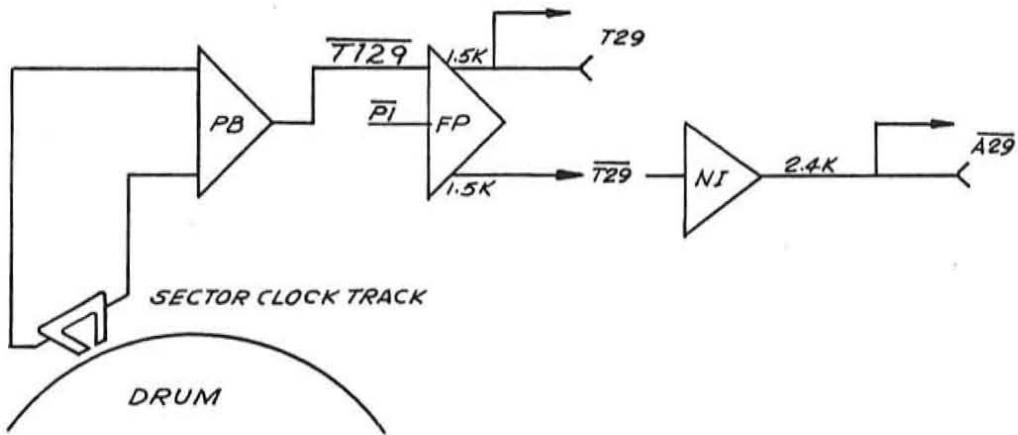
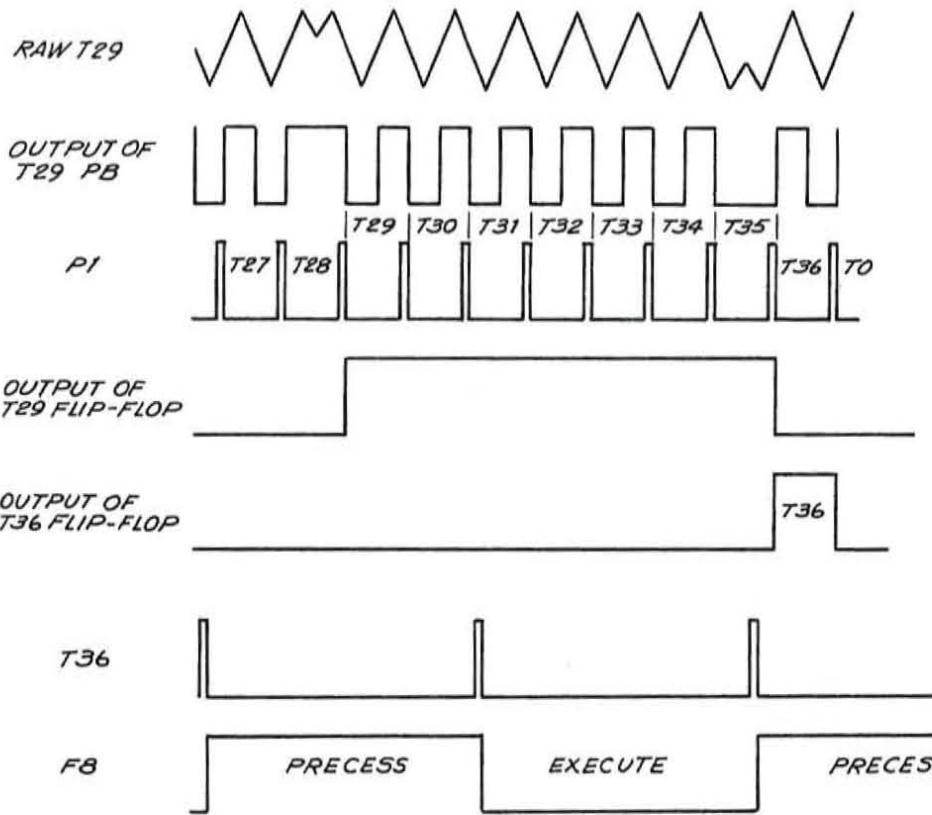




DRUM INFORMATION

1. 75 WORDS = 1 DRUM REVOLUTION
2. 1 WORD = 37 BITS
3. 1 DRUM REVOLUTION = 2775 BITS
4. DRUM SPEED 3,500 R.P.M. - APPROXIMATELY
5. 1 DRUM REVOLUTION = 17.1 M. SEC. "
6. 1 WORD TIME = 228 μ SEC. "
7. 1 BIT TIME = 6.16 μ SEC. "
8. PULSE FREQUENCY = 161 KC PER SEC. "





SECTION III

CIRCUITS

| | Page |
|---|--------|
| III-1. GENERAL | III-1 |
| III-2. UNIVERSAL CIRCUITS | III-2 |
| III-2-1. The AND Gate | III-2 |
| III-2-2. The OR Gate | III-3 |
| III-2-3. The Non-Inverter | III-4 |
| III-3. ARITHMETIC AND MEMORY CIRCUITS | III-5 |
| III-3-1. The Dual Inverter..... | III-5 |
| III-3-2. The Flip-Flop | III-7 |
| III-3-2-a. The 20A Trigger Circuit | III-7 |
| III-3-2-b. The 20B Trigger Circuit | III-7 |
| III-3-2-c. The Basic Flip-Flop | III-7 |
| III-3-3. Playback Amplifier | III-10 |
| III-3-4. Playback Flip-Flop | III-11 |
| III-3-5. Clock Generator | III-12 |
| III-3-6. The Clock Inverter | III-14 |
| III-3-7. The Record Circuits..... | III-15 |
| III-3-7-a. The Record Trigger | III-15 |
| III-3-7-b. The Record One Shot | III-16 |
| III-3-7-c. The Complementary Symmetry Amplifier | III-16 |
| III-3-7-d. The Record Output | III-18 |
| III-3-8. Track Selection | III-20 |
| III-3-9. Sequencing | III-24 |
| III-4. POWER CIRCUITS | III-29 |
| III-4-1. Main Power Supply | III-29 |
| III-4-2. Control Panel | III-31 |
| III-5. INPUT-OUTPUT CIRCUITS | III-32 |
| III-5-1. Single Input-Output Power Supply | III-33 |
| III-5-2. Bias Supplies | III-34 |
| III-5-3. The Output Buffer Circuit | III-35 |
| III-5-4. The Device Memory Circuit | III-36 |
| III-5-5. The Output Common Circuit | III-37 |
| III-5-6. The Input Clear Circuit | III-38 |
| III-5-7. The AND-Not Circuit | III-39 |
| III-5-8. The OR-Not Circuit | III-40 |
| III-5-9. Driver Circuits | III-41 |
| III-5-10. Inverter Circuits | III-42 |
| III-5-11. Logic Gates | III-43 |

CIRCUITS

SECTION III

CIRCUITS

III-1. GENERAL

The logical operations and system functions of the Monrobot XII, which were described in Section II, are implemented by various combinations of "universal" and special circuits. Universal circuits are used throughout the machine. The other circuits are used in one section only.

The following nomenclature is to be understood unless otherwise stated:

- a. All resistor values are in ohms.
- b. All resistors are rated at $\frac{1}{4}$ -watt, 5%.
- c. All capacitor values are in uuf (pf).
- d. All diodes are general purpose germanium diodes (Monroe No. 580029).

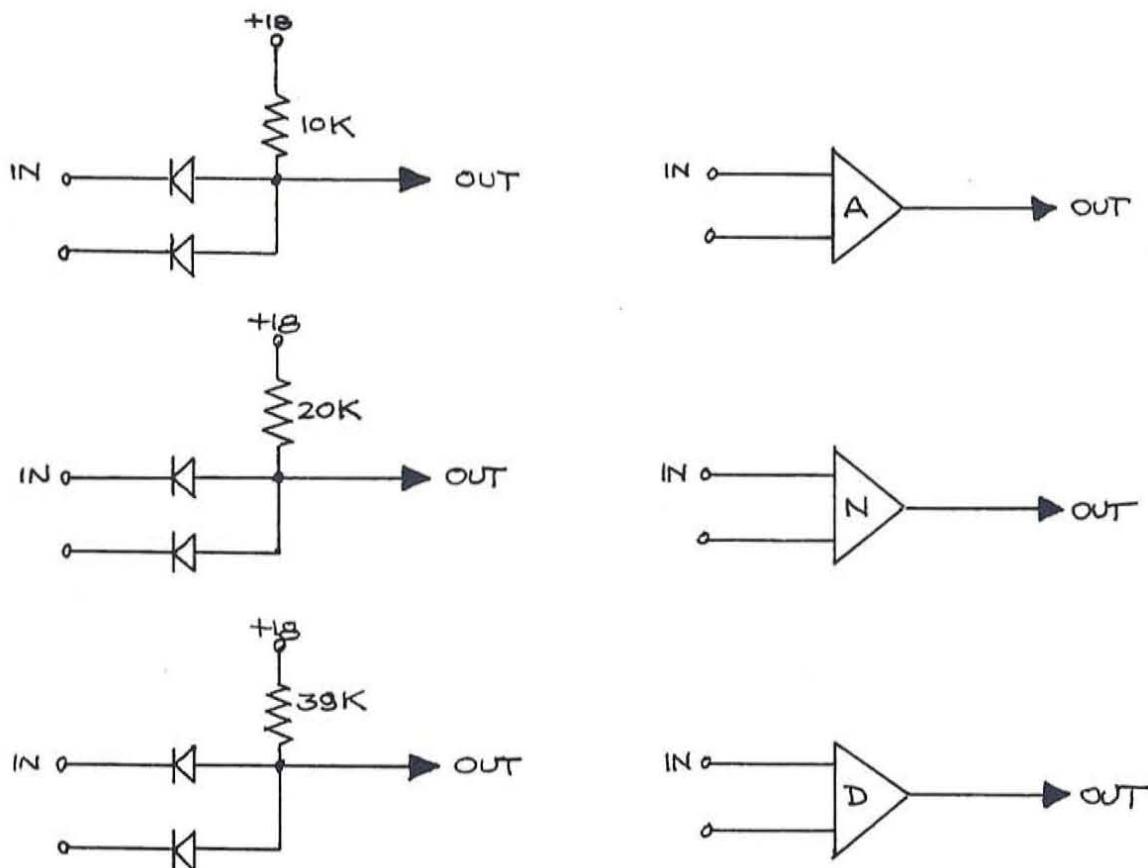
III-2. UNIVERSAL CIRCUITS

III-2-1. The AND Gate.

The AND gate is a logic circuit in which all of the inputs must be high in order to obtain a high output. Several variations, consisting of different values for the load resistors, are used in the computer. In addition, the number of inputs may vary.

The output is pulled high by the +18 volt supply, but the maximum output voltage is controlled by the lowest input voltage. If any of the inputs are low or -6 volts, the output must also be low. If all of the inputs are high or 0 volts, the output is high also.

The circuit schematics and the corresponding logic symbols are shown below.

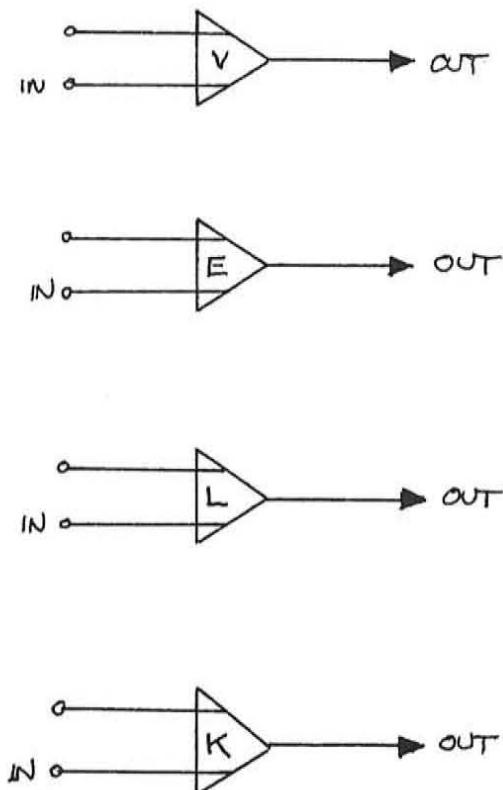
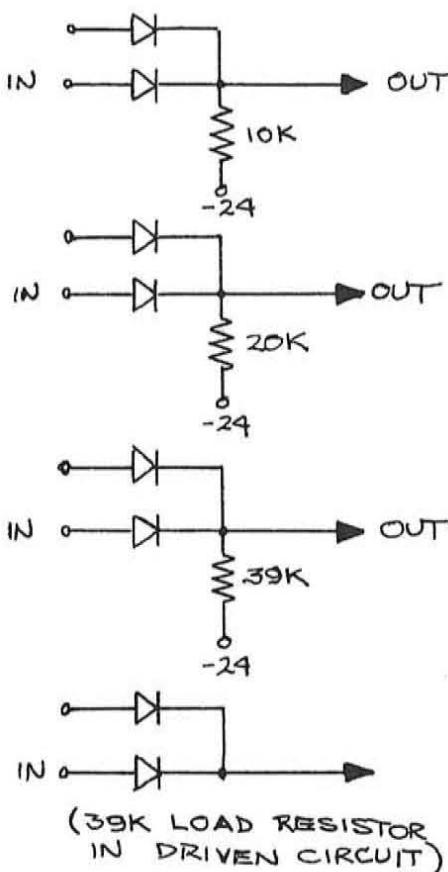


III-2-2. The OR Gate.

This is a logical circuit in which a high at any of the inputs will produce a high output. As in the case of the AND gate, the principal points of variation in the several OR gate configurations lie in the different values for the load resistor.

The output of an OR gate is controlled by the voltage level of the inputs -- it will be as positive as the most positive input. If all of the inputs are low or -6 volts, the output will also be -6 volts.

The OR gate schematic diagrams and their corresponding logic symbols are shown below.



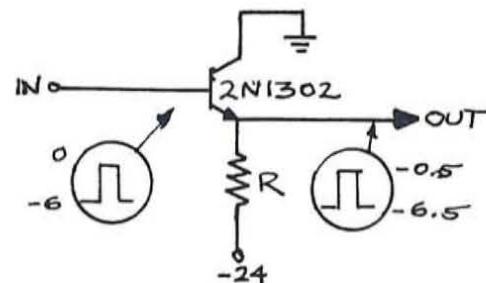
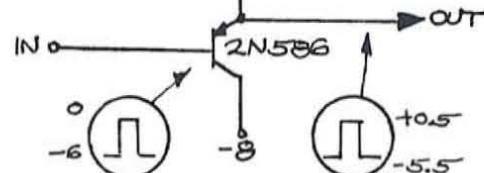
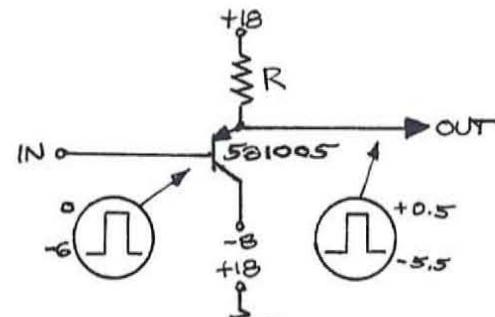
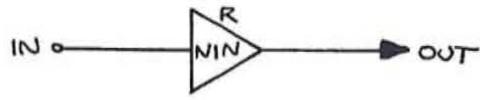
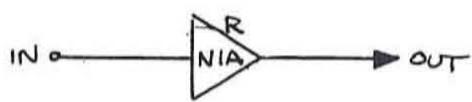
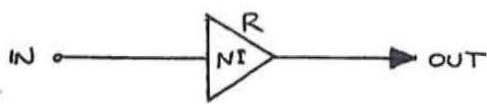
III-2-3. The Non-Inverter.

The non-inverter is essentially a buffer amplifier. It is used in the computer whenever it is desired to increase the power level of a signal without inverting it. Thus, a high at the input to the non-inverter will produce a high at the output and a low input will produce a low output. To accomplish this, a conventional common collector or emitter follower transistor circuit is used.

Since the voltage gain for a non-inverter is very close to 1, in practice the output signal is the same as the input signal, except for a level shift. As shown in the figure, three different types of non-inverters are used in the Monrobot XII -- the basic NI non-inverter, the NIA non-inverter and the NIN non-inverter. The first two both use p-n-p transistors, that used in the NIA non-inverter being a somewhat higher power transistor.

The positive level shift induced in the NI and NIA non-inverters is indicated in the first two sets of waveforms. This level shift limits the number of non-inverters that can be connected in series or cascade. To counteract this, an n-p-n non-inverter called an NIN is also used in logic chains. It can be seen that, in this case, a negative level shift takes place.

The values of the load resistor, R , depend on the number of circuits driven and the driving circuit.



III-3. ARITHMETIC AND MEMORY CIRCUITS

III-3-1. The Dual Inverter.

The Dual Inverter is a logical circuit which produces both a high and a low output from a high input. It returns the signal voltage to the logic levels and produces both a real and a primed form of the signal. A "1" at the input will produce a "0" at the inverted output \bar{I} and a "1" at the real output I .

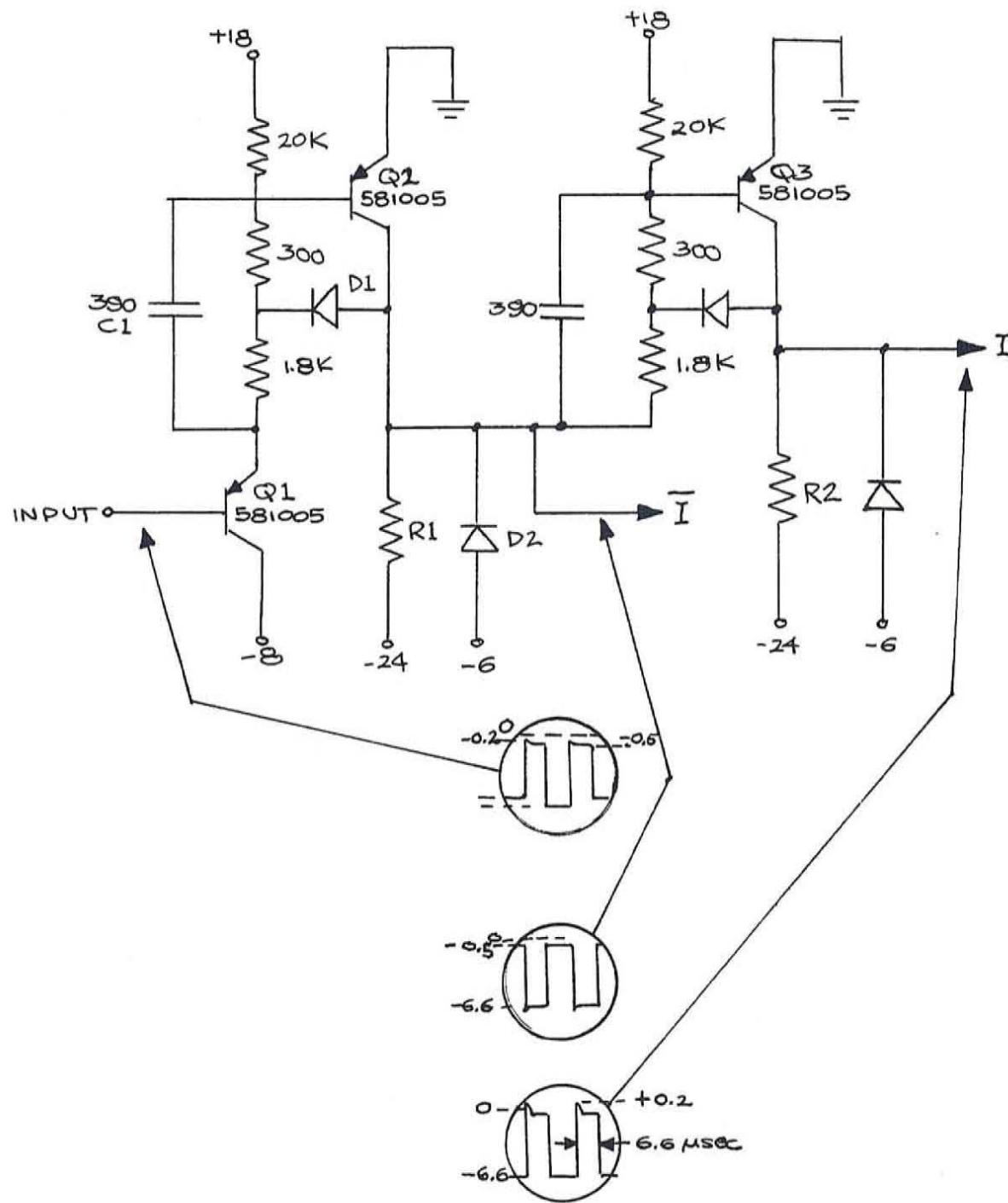
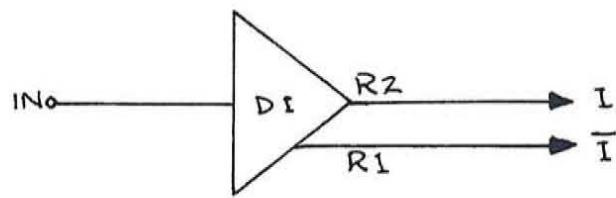
The circuit actually performs two inversions in sequence. Physically, it consists of a two-transistor inverter circuit, driving a grounded emitter amplifier, as shown in the figure. The primary purpose of the latter amplifier is to re-invert the signal.

The first inverter circuit is simply an emitter-follower feeding a grounded emitter amplifier. When the input is high, transistor Q_1 holds the base of Q_2 above ground through the resistor divider, keeping Q_2 off. Diode D_2 clamps the inverter output \bar{I} to -6 volts. With a low input, the Q_1 emitter is low or near -6 volts. The base of Q_2 is pulled negative, turning Q_2 on. C_1 is a speed-up capacitor.

If Q_2 is allowed to saturate, it will take a long time to turn off. To speed up operation, Q_2 must therefore be prevented from saturating. This is effected by preventing too much current from flowing in the Q_2 base, as follows: When the collector goes positive enough, diode D_1 conducts, drawing away some base drive. Typically, the Q_2 collector will be at -0.5 volts. With very high Beta transistors, although the base drive is greatly reduced, the transistor may still be slightly in saturation and the Q_2 collector will be at -0.2 volts.

Diode D_2 prevents Q_1 from pulling the output \bar{I} too negative.

The second inverter in the dual inverter circuit operates in exactly the same fashion. In this case, the output from the first forms the input to the second. When the first output is low, Q_3 turns on; when it is high, Q_3 turns off. Thus, the state of the real output I is the opposite of the first inverter output \bar{I} .



III-3-2. The Flip-Flop.

The flip-flop is one of the most important and fundamental component circuits in the computer. It is a bistable multivibrator which can be made to assume one of its two stable states when the required input conditions are met. It will maintain that state until a different set of input conditions cause it to assume the other state. It can maintain either state indefinitely, even though the conditions which caused it to assume that state were met only briefly.

There are two basic forms of flip-flop used in the Monrobot XII — the 20A and the 20B. They are identical except for their input trigger circuits. The 20A flip-flop requires the clock to be part of the logic driving it, while the 20B makes use of the clock as a separate input to its trigger circuit. The information input to the 20A trigger circuit must be low for some time before a positive-going clock pulse provide the trigger for the circuit. For the 20B, the information input must remain high for a short period before a positive-going clock can trigger it.

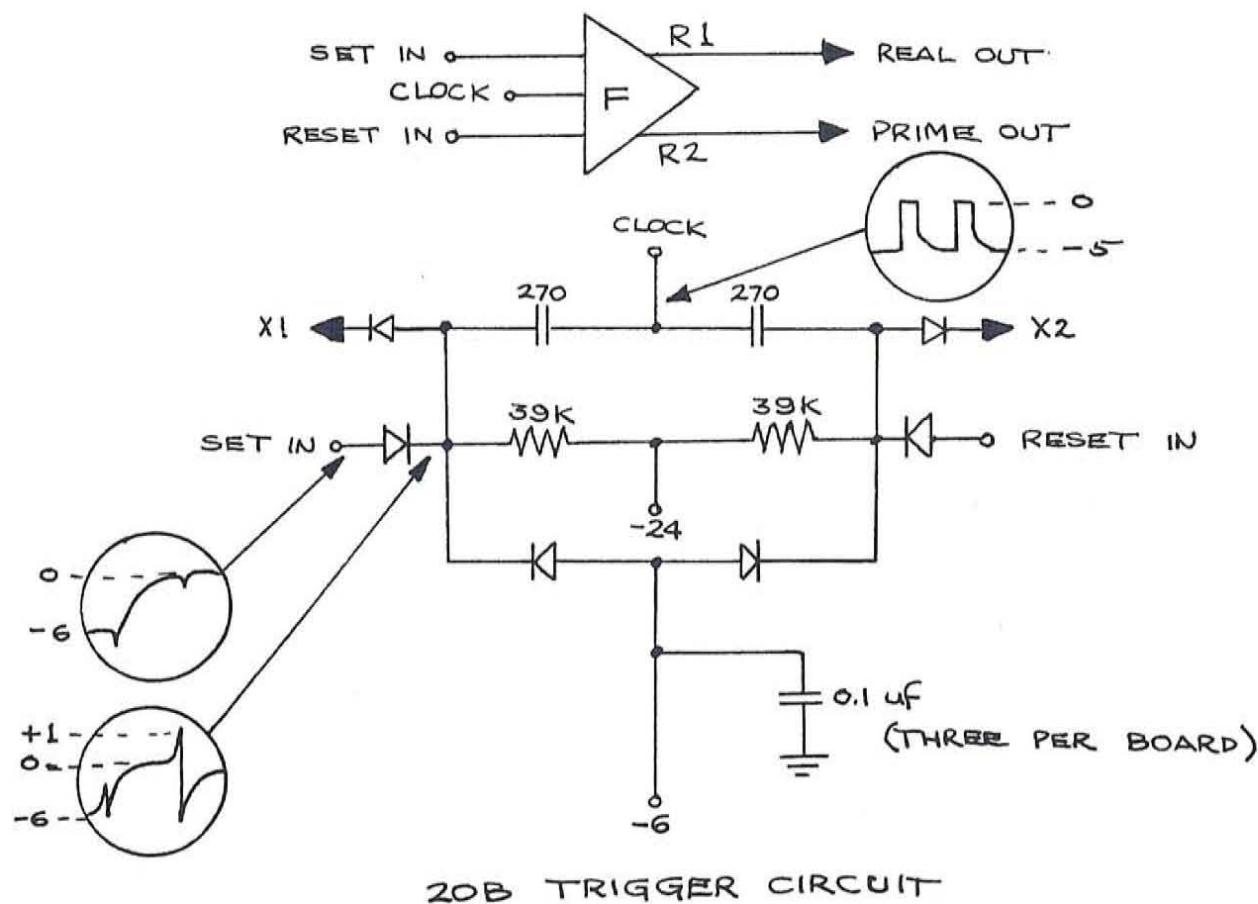
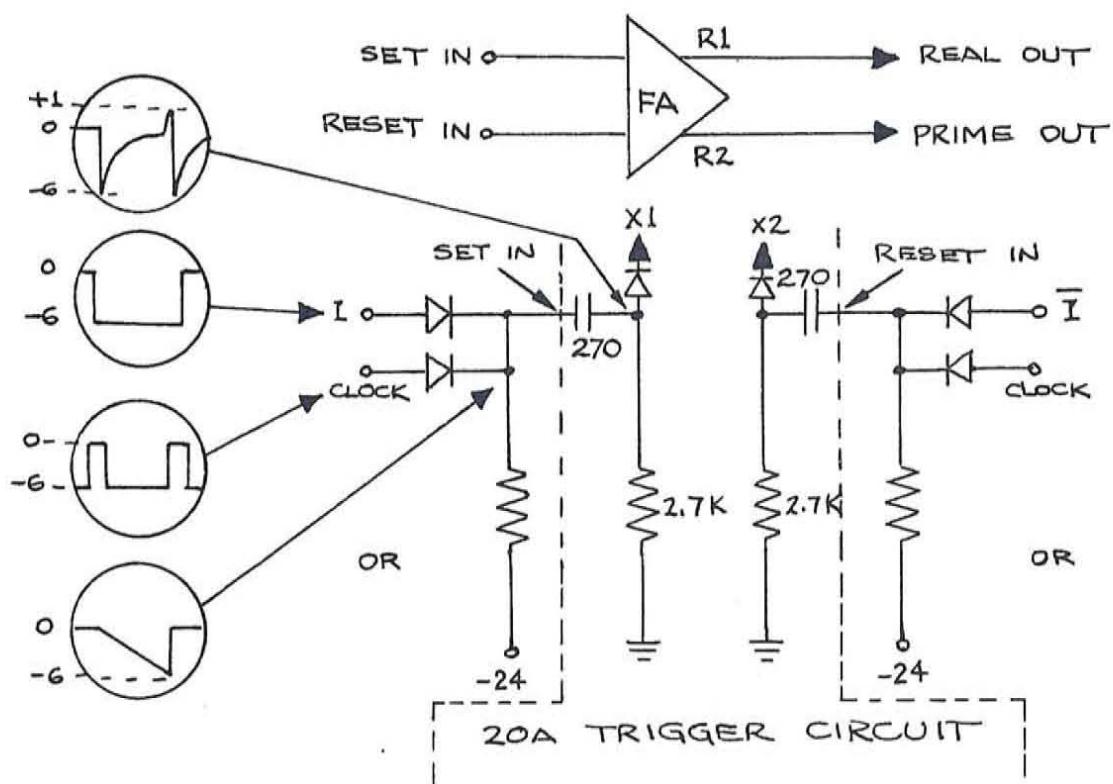
III-3-2-a. The 20A Trigger Circuit. Consider that the I input to the OR gate is initially high and is going low. When the input voltage goes low, the capacitor charges through the 2.7K resistor and the output impedance of the driving circuit. After a short period, the capacitor is fully charged and the circuit can be triggered by a positive-going clock at the OR gate. The capacitor charge plus the positive-going clock produce a voltage which is sufficiently above ground to turn the on-transistor off, changing the state of the flip-flop.

Therefore, to trigger a 20A flip-flop, the signal must first go low for a long enough period to charge the capacitor and must then be followed by a positive-going clock, which performs the triggering action.

III-3-2-b. The 20B Trigger Circuit. Initially, the input signal and the clock will be considered low. There is no charge on the capacitor, since both sides are held to -6 volts. If the signal goes high and the clock remains low, the capacitor charges. It takes a short time to do this, since it must charge through the output impedance of the driving circuit. Then, when the clock goes positive, the voltage spike that appears on the output is equal to the capacitor charge plus the positive rise in the clock voltage. Should a clock pulse occur without the input first being high, the capacitor will not be charged, and since the trigger signal thus cannot rise above ground, it cannot trigger the flip-flop.

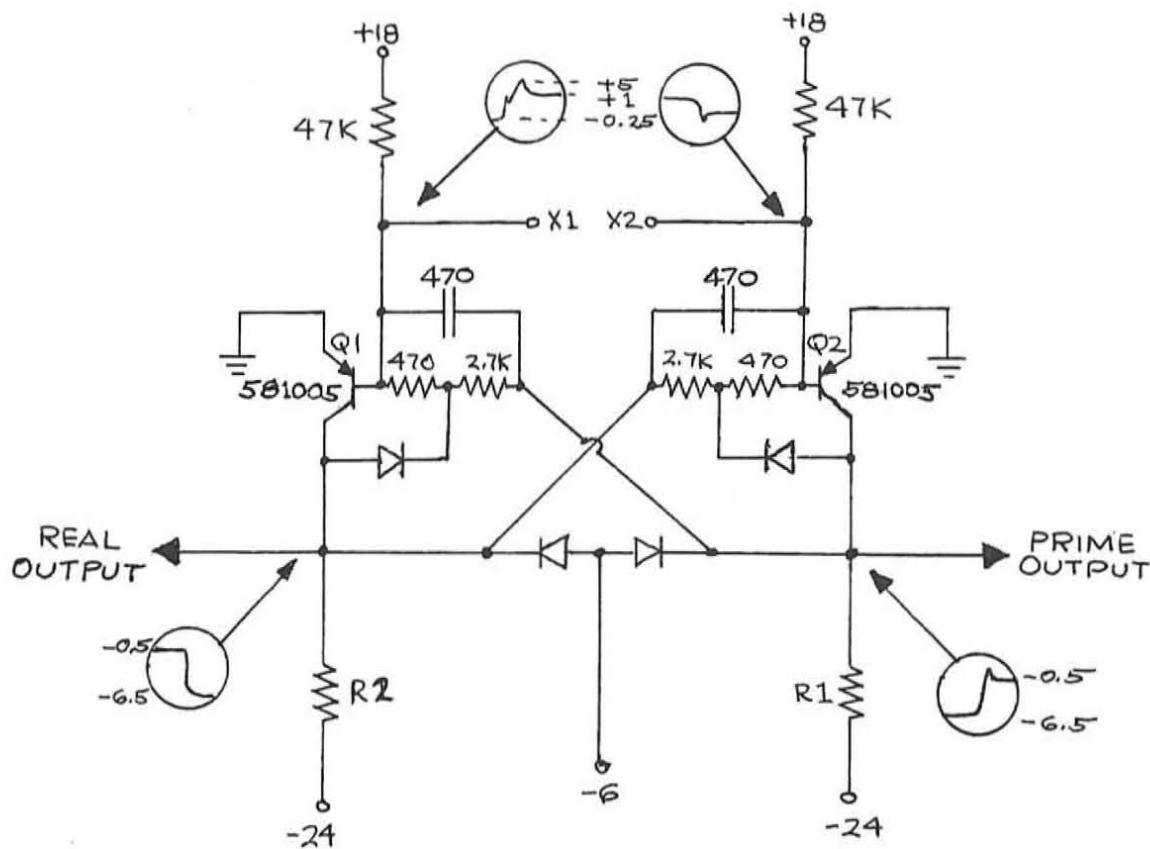
Thus, the input signal to a 20B trigger circuit must go high long enough for the capacitor to charge, which allows a positive-going clock pulse to trigger the flip-flop.

III-3-2-c. The Basic Flip-Flop. Initially, the flip-flop will be assumed to be in a reset state with Q1 turned on and Q2 turned off. In this state, the collector voltage of Q2 will be clamped at -6 volts. This same voltage, when applied to Q1 through the voltage divider, effects a slightly negative voltage on the base of Q1, keeping Q1 turned on. Since Q1 is on, its collector will be almost at ground. The collector of Q1 is coupled to Q2 through a voltage divider, putting



a slightly positive voltage on the base of Q_2 , keeping it turned off. It can be seen that this is a stable condition that will remain so until something happens to upset the circuit. The real output, which is taken from the collector of Q_2 , is at -6 volts and the prime output, from the Q_1 collector, is at ground.

To change the state of the flip-flop, a positive pulse must be applied to X_1 . This pulse must go positive enough to make the base of Q_1 more positive than the emitter, which causes Q_1 to turn off and clamp its collector voltage at -6 volts. When this voltage is applied to the Q_2 voltage divider, the base of Q_2 becomes more negative than its emitter, turning Q_2 on and clamping its collector voltage slightly below ground. This voltage, applied through the Q_1 divider, keeps Q_1 turned off after the trigger pulse has terminated. Thus, the flip-flop has assumed its second stable state, which it will not change until a positive trigger pulse is again applied to the reset input.



III-3-3. Playback Amplifier.

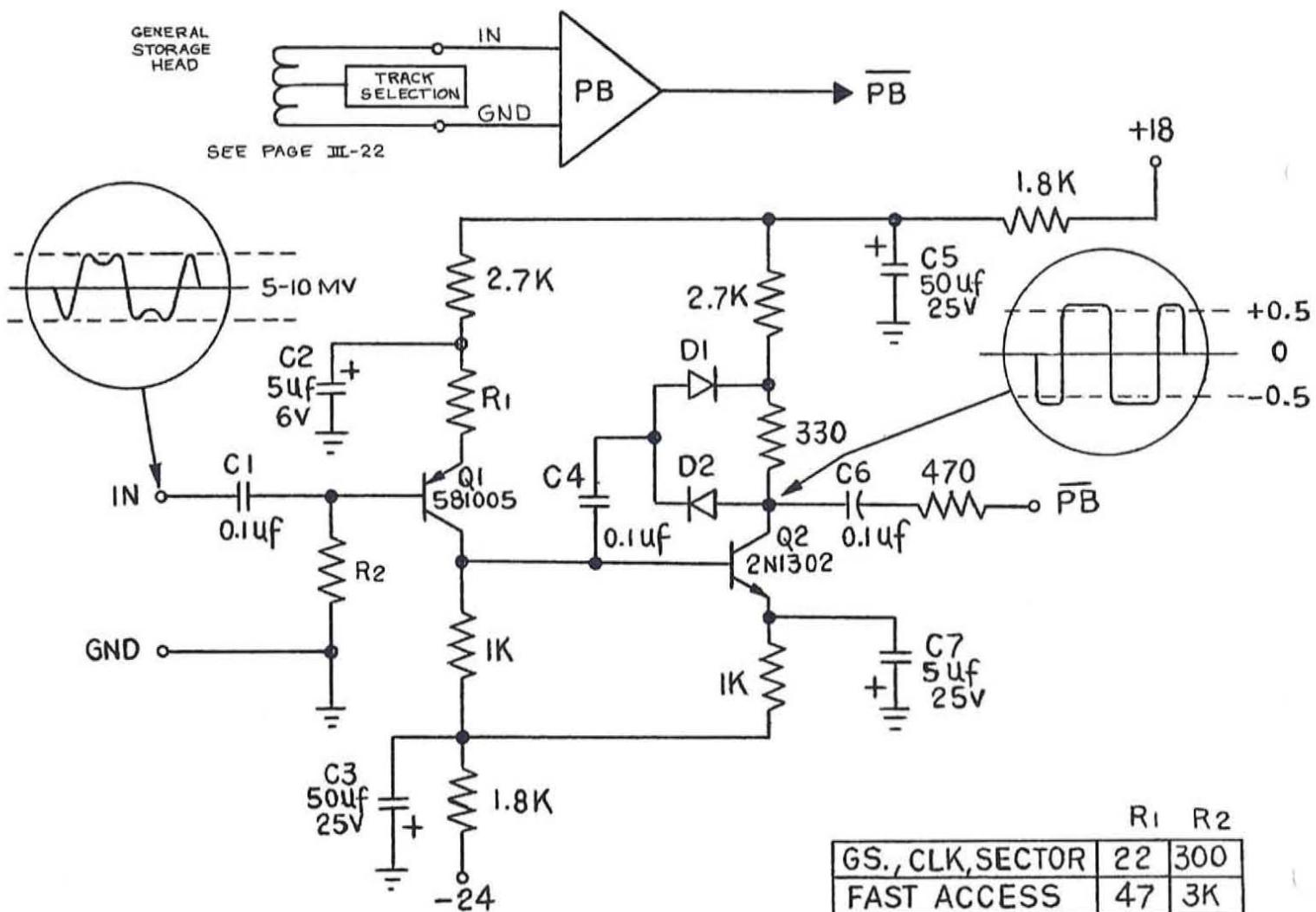
The playback circuit is used to amplify the millivolt signals from a playback head. It is a two-stage amplifier which operates class A, with the output of transistor Q1 DC-coupled to Q2.

Diodes D1 and D2 provide a negative non-linear feedback which acts to limit the output to approximately ± 0.6 volts. The output of Q2 is an amplified version of the input to Q1.

The ground wiring for the playback amplifier is kept separate from the ground runs for the succeeding driver and flip-flop, because of the millivolt signal involved.

C5 and C3 are bypass capacitors to prevent noise on the +18 and -24 volt supplies from affecting the playback circuit.

C2 and C7 are emitter bypass capacitors.



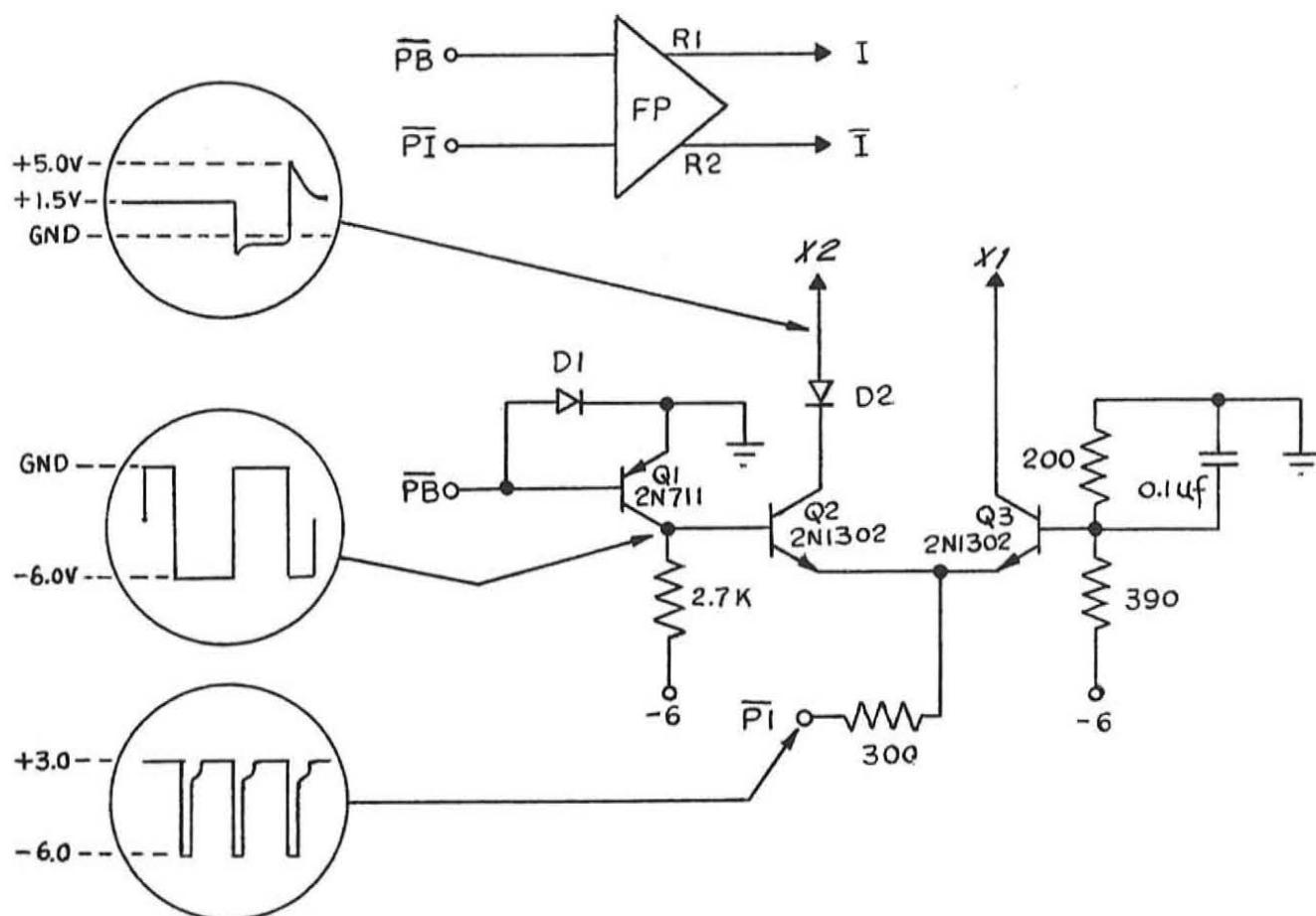
III-3-4 Playback Flip-Flop

A special trigger circuit which gates together the output of a playback amplifier and $\bar{P}1$ is used to trigger the playback flip-flop. The basic flip-flop is a standard series 20 circuit as described in Section III-3-2-C.

The output of the playback amplifier ($\bar{P}B$) is amplified by $Q1$. A high input turns off $Q1$, a low turns it on. When $Q1$ is on it loads capacitor $C6$ in the playback circuit discharging it somewhat. When $Q1$ is off $D1$ serves to charge up $C6$ again.

The clock signal $\bar{P}1$ is fed to the junction of $Q2$ and $Q3$ emitters and the triggering of the flip-flop occurs when $P1$ goes low.

Assume that the flip-flop is reset. When $Q1$ turns on, the base of $Q2$ goes high and when $\bar{P}1$ goes low $Q2$ will turn on. The flip-flop off-transistor is now forward biased and turns on setting the flip-flop. If the next bit from playback is a zero, then $Q1$ and $Q2$ remain off when $P1$ goes low. However, $Q3$ turns on and forward biases the flip-flop off-transistor turning it on and resetting the flip-flop. Thus the above action performed the following, when the output of the playback amplifier was a one the flip-flop was set, and when it was a zero the flip-flop was reset.



III-3-5. Clock Generator.

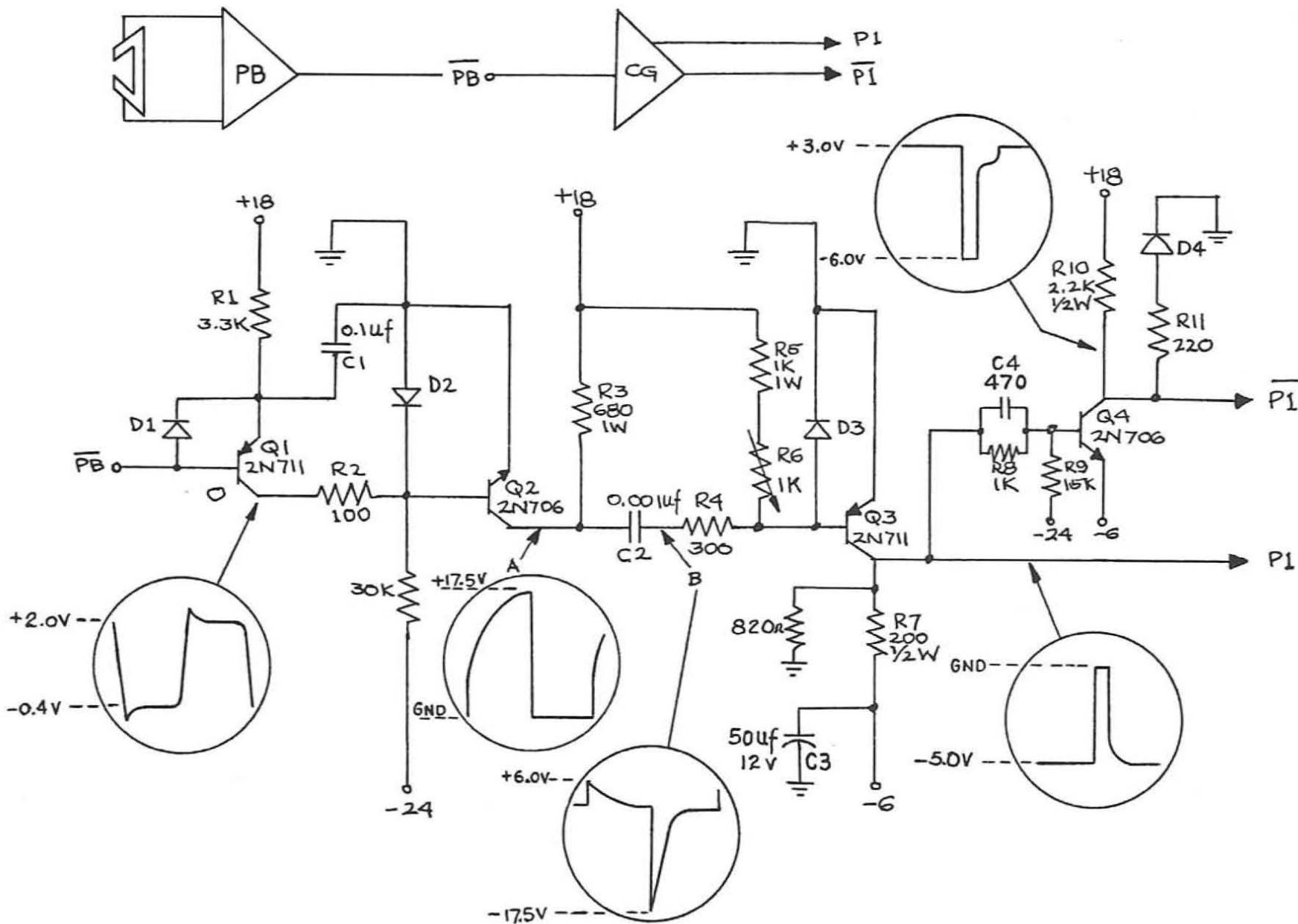
The Clock Generator (CG) circuit produces both a P_1 and a $\overline{P_1}$ clock, which are used on the memory and arithmetic boards. The input to the clock generator is the output ($\overline{P_B}$) from a standard playback amplifier.

The input signal swings cause Q_1 to be alternately on and off. Q_2 goes alternately on and off as Q_1 does. Diode D_2 is a protective diode which prevents the base of Q_2 from going too negative.

In the Q_3 stage, capacitor C_2 and the resistors R_4 , R_5 and R_6 form an RC time constant which determines the pulse width of P_1 . When Q_2 is off, point A charges to +18.0 volts, point B is at +0.5 volts. When Q_2 turns on, point A is discharged to ground, by capacitor action, point B drops low to -17.5 volts turning Q_3 on-- and then gradually returns to its +0.5 level. During its return, a point is reached where the current supplied by the C_2 is not sufficient to keep Q_3 on. Then the R_6 - R_5 - D_3 network turn Q_3 off again. The length of time required for this to happen is controlled by the variable resistor R_6 . The design time constant for the system is $\frac{1}{2}$ microsecond. When Q_3 is on, output P_1 is high. Capacitor C_3 prevents the pulses generated by the circuit from affecting the -6 volt supply. Diode D_3 prevents the base of Q_3 from being driven too positive.

When Q_3 is on and P_1 is high, Q_4 turns on and its output $\overline{P_1}$ goes low; thus P_1 is low when P_1 is high. When Q_4 is off, R_{11} and D_4 form a clamping circuit so that P_1 is clamped at +3.0 volts. The purpose of this clamp is to help produce a clean P_1 signal at the output of the clock inverter circuit located on the logic board (see description of the clock inverter). Capacitor C_4 is a speedup capacitor.

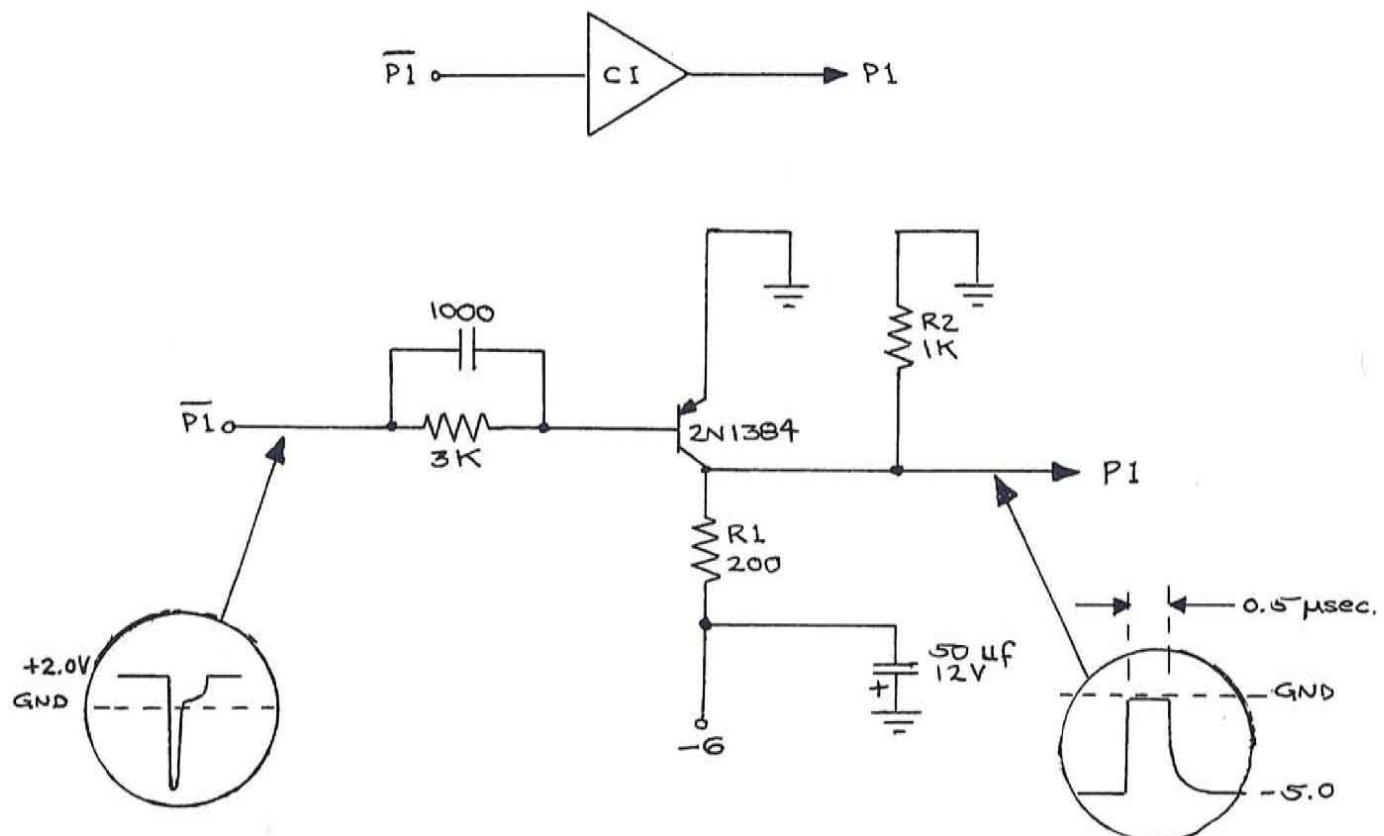
When power is turned on, before any pulses appear from the playback circuit, the base and the emitter of Q_1 are both charged to about +18 volts. As pulses begin to appear from playback, C_1 and the capacitor output discharge and the Q_1 stage gradually seeks its operating level such that the Q_1 emitter is at approximately +2volts and the Q_1 base is about -.2 volts less.



III-3-6. The Clock Inverter

The clock inverter is a special inverter circuit which produces a clock signal P_1 from its inverse \bar{P}_1 . While it is true that there is also a clock generator which produces both P_1 and \bar{P}_1 , this circuit is located on the memory board; the clock inverter is used on the arithmetic board. Any attempts to bring P_1 directly from the memory clock generator would introduce unwanted noise and inductive effects -- ringing, etc. Hence, the \bar{P}_1 signal is fed to an inverter on the arithmetic board in order to produce a local P_1 signal.

When the input \bar{P}_1 signal is low, the transistor turns on, producing the high output P_1 . R_1 and R_2 form a voltage divider to produce a clock amplitude of five volts. The 50 μ f capacitor prevents modulation of the -6 volt supply by the clock inverter.



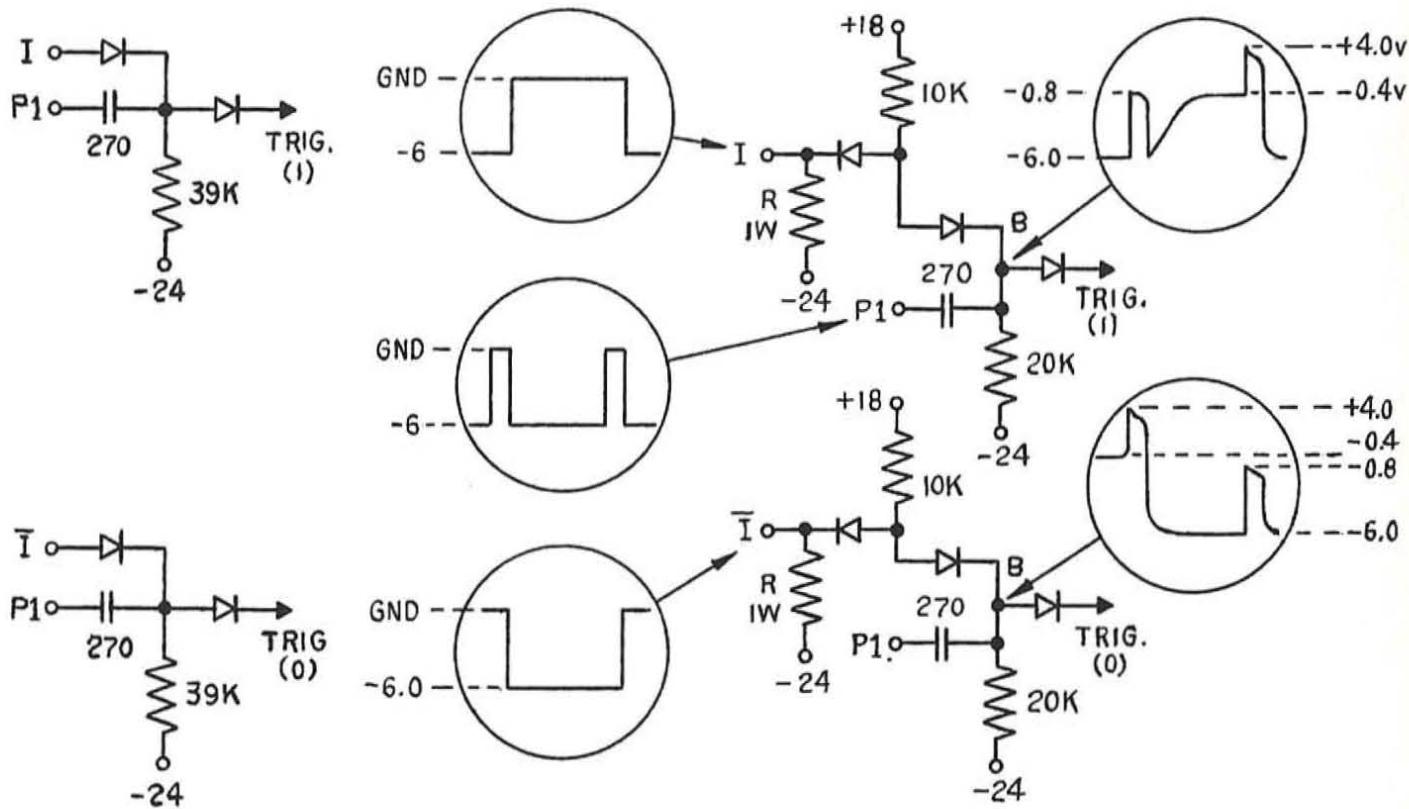
III-3-7. The Record Circuits.

The circuits which are used in the Monrobot XII to record information into General Storage and the Fast Access channel are almost identical. These circuits furnish the proper drive to the recording heads under the control of a clock and the information to be recorded.

The Fast Access and the General Storage Record circuits are each constructed of two identical record amplifiers; each amplifier drives half of the recording head. Since the windings of the record head halves are 180° out of phase, one of the individual record amplifiers will record 1's and the other will record 0's. Each individual record amplifier consists of four parts: a trigger circuit, a record one-shot, a complementary symmetry amplifier and an output stage.

III-3-7-a. The Record Trigger. There are two inputs to each record amplifier -- an information signal and a clock signal. When the information input is low, point B and the trigger to the one-shot will also be low. When the information is high, B rises to ground, charging the capacitor. During this time, it is assumed that the clock input, P_1 , is low.

When a high clock does arrive, its voltage swing to ground, discharges the capacitor and the voltage on the capacitor is added to the clock voltage; point B rises above ground to about +4 volts. Since the one-shot circuit requires a pulse which is a few volts positive, this rise is sufficient to trigger it.



III-3-7-b. The Record One-Shot. This circuit, which generates the actual record pulse, is a monostable multivibrator. In its stable state, Q1 is held on by the negative voltage applied through R* and R2 to its base. This clamps its collector almost to ground. Q2, meanwhile is kept turned off by the positive voltage at its base derived by the 3.9K and 33K voltage divider. Its output is thus clamped to -6 volts.

When the trigger pulse occurs, Q1 is turned off. As a result, its collector goes negative and is clamped at -6 volts. This negative voltage, applied to the base of Q2, turns it on and causes its collector to be clamped to ground.

This positive rise in the Q2 collector voltage, fed back to the base of Q1, keeps it turned off and maintains this condition until the 220 pf capacitor has discharged enough to allow Q1 to turn on again (a period of about 1.2 microseconds). At that time, the circuit returns to its stable state and holds it until another trigger occurs. The 10K potentiometer in the RC network allows pulse width adjustment to balance record currents through both halves of the record head. The General Storage system requires a slightly longer one-shot period for proper recording of information. The value of R* is used to vary the range of the RC network.

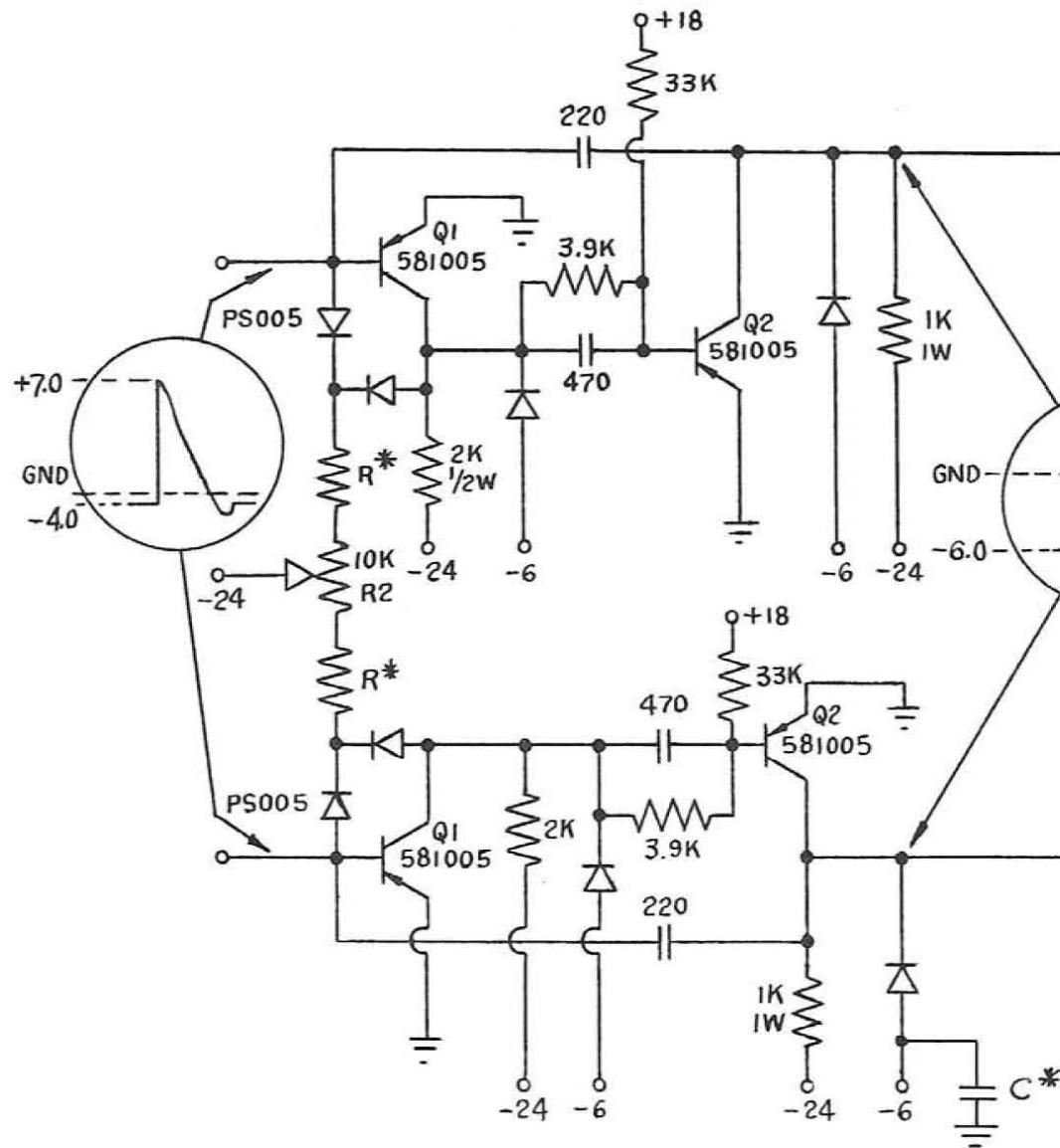
The output from the one-shot circuit is taken from the collector of Q2. The collector voltage rises to ground after a trigger and remains there for the 1.5 microsecond time that Q2 is on. This is the output pulse. After that interval, the voltage drops back to -6 volts and remains there until a trigger appears again.

III-3-7-c. The Complementary Symmetry Amplifier. When the input to this stage is low -- as it will be when there is no trigger -- Q4 is kept turned off by the 1K and 20K voltage divider. The emitter voltage of Q3 is maintained at +3 volts by the 120 ohm and 1K voltage divider and Q3 is biased off by the voltage drop of the base to emitter diode. When the one-shot pulse appears, the voltage at the input rises to 0 volts and Q4 turns on. Q4 then clamps the output of the circuit to -6 volts. After a short period, the one-shot goes low, turning off Q4 and turning on Q3, which brings the output voltage to +3 volts. When the coupling capacitor in the base of Q3 has been charged, Q3 turns off. The output, however, remains at +3 volts because of the connection to the Q3 emitter through the 1K resistor.

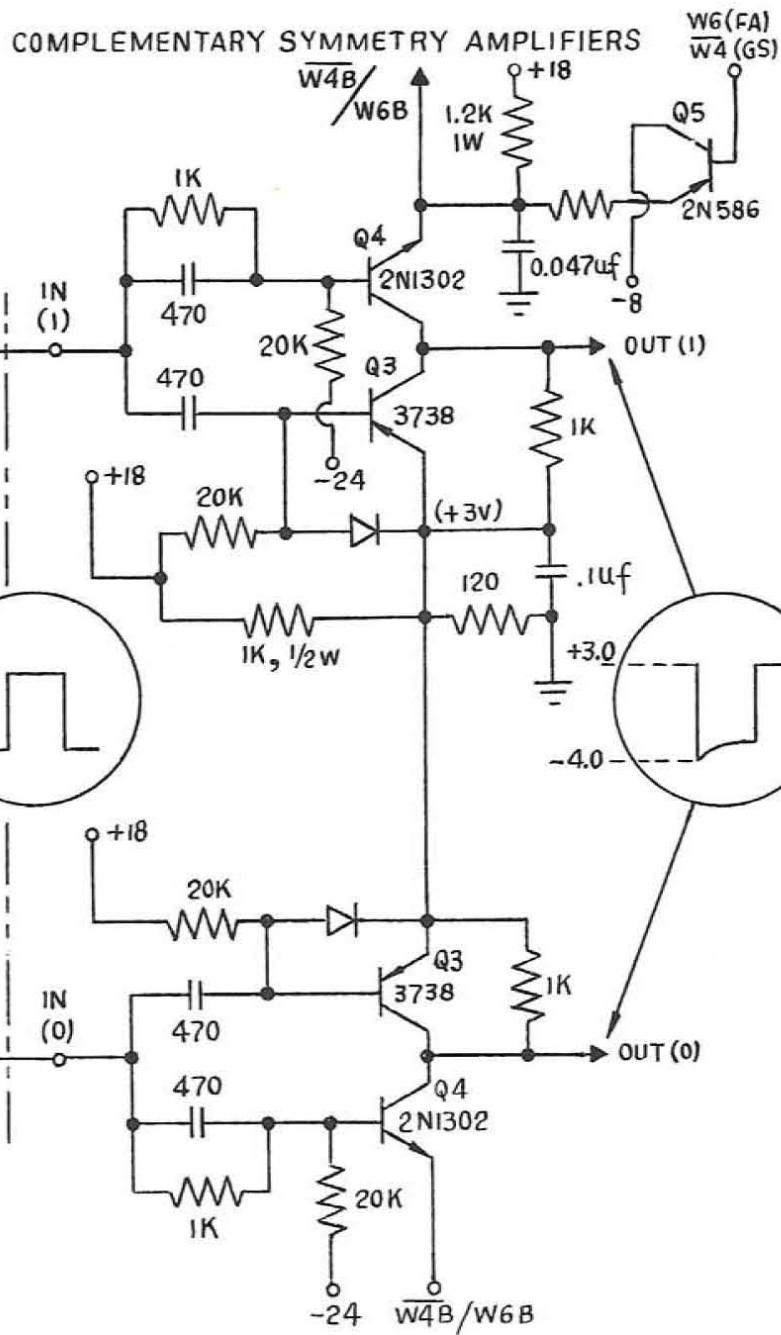
The signals W6 in the case of Fast Access record, and $\overline{W4}$ in the case of General Storage, form an input to the base of emitter-follower Q5. Thus, when either input signal is high -- as they are during initial start-up or restart -- the voltage at the emitter of Q4 is high and Q4 is prevented from turning on. Therefore, recording in either the Fast Access or General Storage is effectively blocked unless the W signal is low.

ONE SHOT CIRCUITS

| | R^* | C^* |
|-----------------|-------|-------------|
| FAST ACCESS | 5.1K | — |
| GENERAL STORAGE | 8.2K | 0.1 μ f |



COMPLEMENTARY SYMMETRY AMPLIFIERS

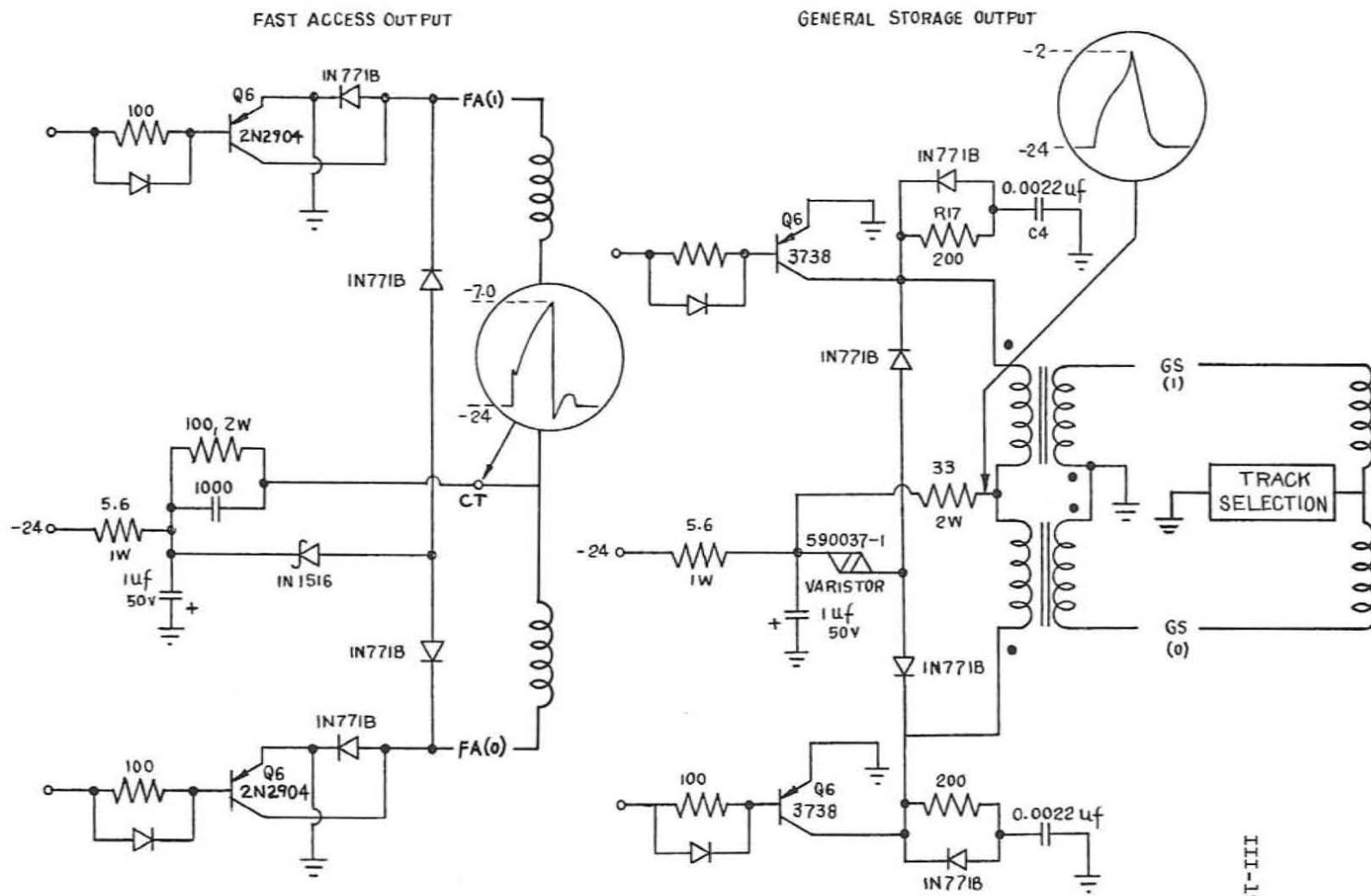


III-3-7-d. The Record Output. After a trigger has occurred, the input to Q6 goes low, turning the transistor on and causing the collector to rise from -24 volts to ground. Current then flows through one-half of the record head and through the current limiting resistor to the -24 volt supply. This is the record current. When the interval is over, the input to the base of Q6 goes to +3 volts and the transistor turns off, terminating the record pulse.

The resistors and diodes in the output circuit provide damping and thus control the peak transistor power dissipation. The ground return wires that carry heavy record currents are kept separate from the remainder of the ground wiring.

Fast Access Record: The record pulse is approximately 1.2 microseconds long with a peak of 0.175 ampere. The head is connected directly across the output of the record circuit. The type of head used has close coupling between its two winding halves. This causes a large backswing voltage to appear at the output transistor of the inactive amplifier. Because of this, an output transistor with a collector-base breakdown of 60 volts is used.

General Storage Record: A 0.5 ampere, 1.5 microsecond pulse is needed for recording in General Storage. The record circuit output is connected to a record transformer. The primary purpose of the transformer is to supply inverted record signals required for the track selection circuit. The transformer consists of separate pot core assemblies for the ones and zeros record signals. This minimizes the coupling from one record amplifier to the other. The collector of the record transistor will reach a peak voltage of about -35 volts at the completion of the record pulse, which is caused by stored energy being released by the record head. The varistor limits and lengthens this peak voltage to lower the power dissipation of the output transistors.



III-111

III-3-8. Track Selection.

This circuit controls which General Storage heads are to be connected to the record and playback circuits. The memory element is a ring counter, Q1-4, which selects the tracks consecutively. The counter advance signal is transmitted through inverter Q5. The counter selects heads through the signals RC1 - RC4 which are produced by control transistors Q6 - Q10. A record head is selected by a high RC signal which allows the diodes at that head to conduct. The RC signal also turns on one of the playback transistors Q11 - Q15 connecting one of the playback heads.

Assume that the Program Switch is selecting track 2A. Operation of the counter is initiated by the signal W4 from the sequencing circuit. When W4 is low, as it is before the computer has reached its operable state, it causes the base of Q1 to go negative and to turn on. The emitter of Q1 is slightly negative and turns on control transistor Q8, grounding the center-tap of the record head for track 1 and turning on the playback transistor Q13 for track 1.

When Q1 is on, the M1 signal at its collector is near ground. This signal is fed through diodes to the base of Q2, Q3 and Q4 keeping these transistors off. The emitters of Q2, Q3 and Q4 are positive keeping Q6, Q7, and Q9 off. The RC2, RC3 and RC4 signals are therefore low preventing the diodes at record heads 2, 3 and 4 from conducting. The low RC signals also bias playback transistors Q11, Q12, and Q14 off keeping the playback heads for tracks 2, 3 and 4 disconnected.

Note that the signals M2, M3 and M4 are low and they will serve to keep Q1 on and the counter in the same state when the W4 signal goes high.

When a high Change Track signal (A28) appears, Q5 turns off. Capacitors C1, C3 and C4 have become charged to -15 volts because they are connected to the collector of the three counter stages Q2, Q3 and Q4 which are off. Capacitor C2 only has a charge of +3 volts. Therefore when the collector of Q5 tries to go to -24 through its load resistor R1, diode D2 turns on and charges C2 through R1. The Q2 base side of the capacitor is pulled negative and turns Q2 on. Q2 collector signal M2 goes to ground and this signal is fed back to inputs of the other three stages. At this time M1 and M2 are both high. M2 at ground turns Q1 off through the M2 diode getting clamped to ground. Q2 remains on because its drive comes from C2 charging through R1 to -24, which lasts for the duration of the one bit time Change Track signal. The time it takes for M2 to turn Q1 off is 2 microseconds. Q2 turns Q9 on and Q9 biases Q14 on which selects track 2A record and playback heads respectively. When the Change Track signal ceases the counter will remain in its new state due to its feedback circuits as described previously.

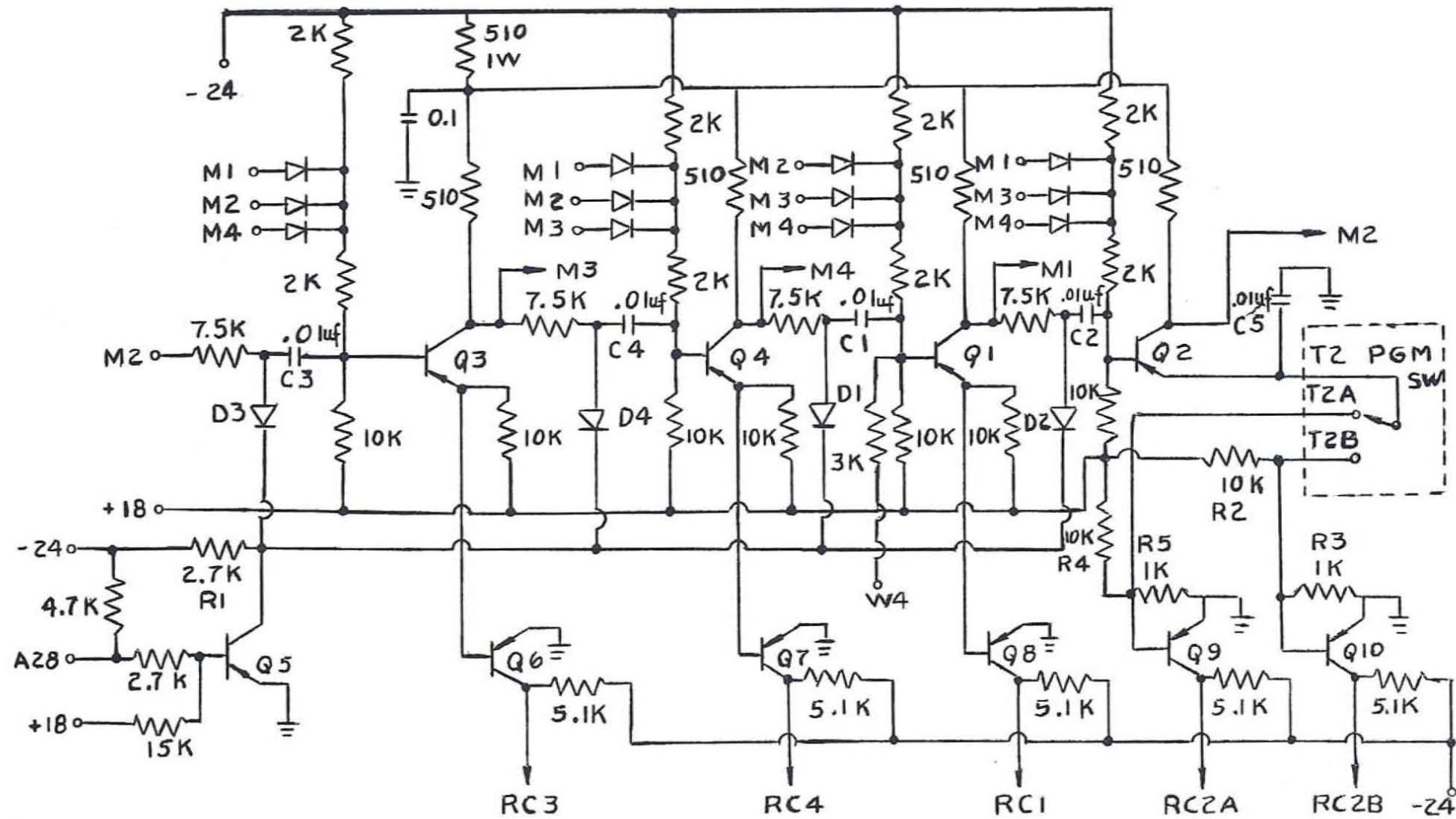
The Program Switch connects tracks 2A or 2B to the ring counter. When selecting 2A, voltage divider R2 and R3 to +18 volts prevents the unused circuit from turning on. When selecting track 2B, divider R4 and R5 keeps 2A off. C5 is a bypass capacitor for any noise that appears on the line connecting to the program switch.

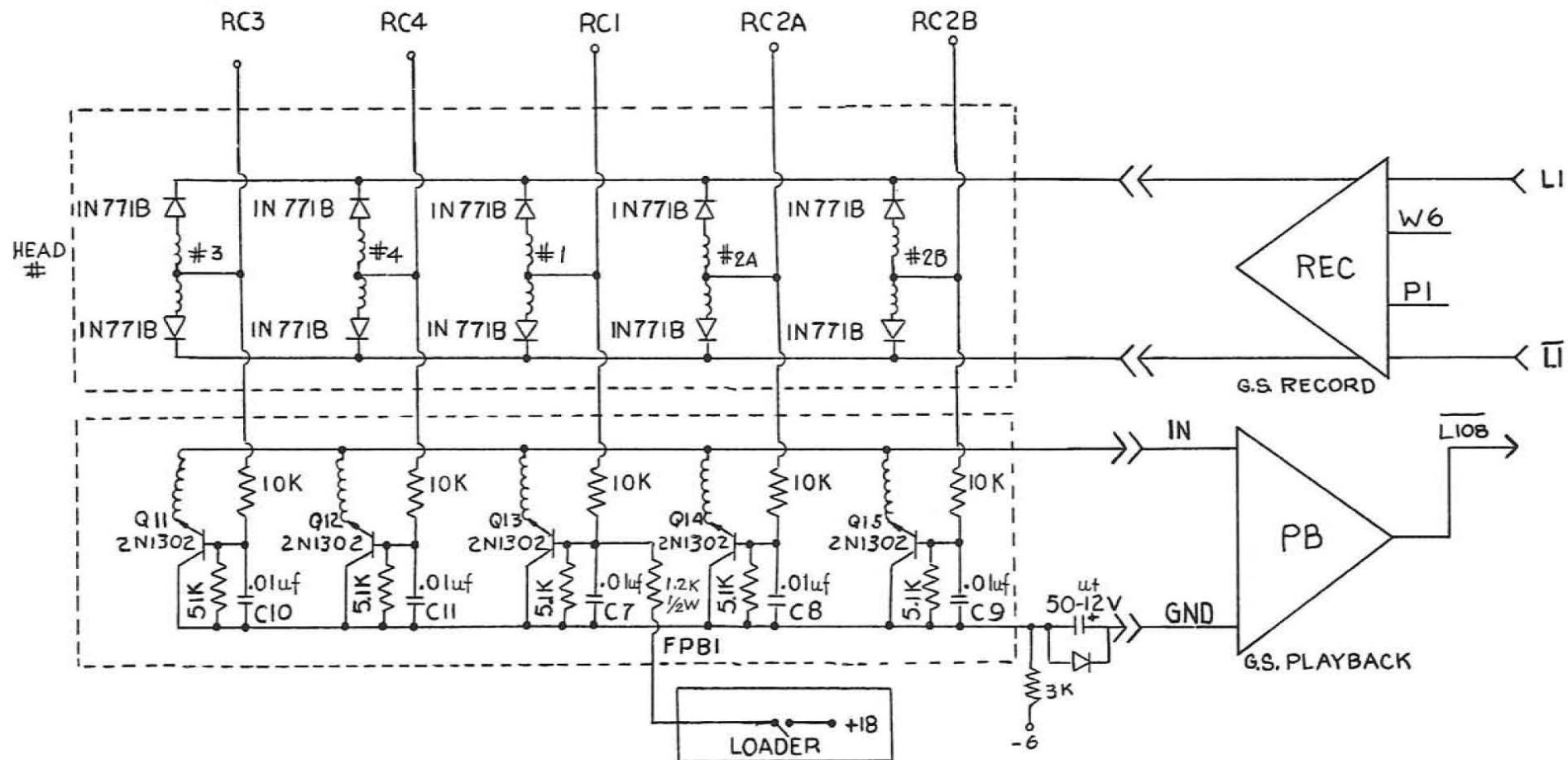
The playback transistors Q11 - Q15 are used with their connections inverted (collector and emitter interchanged). This is done to reduce

leakage current and to keep the voltage drop from base to collector to a minimum. Capacitors C7 - C11 serve as bypasses for record pulses.

The FPB1 line is used when loading a program on the magnetic drum. After track 1 is loaded, a position on the loader track select switch selects Playback track 1 and Record track 2. In this position the ring counter has Q2 on selecting track 2 for recording. It also tries to turn on Q14 but the FPB1 wire makes the base of Q13 more positive than the base of Q14. Q13 acts as an emitter follower. Its emitter voltage is applied through playback heads 1 and 2 to the emitter of Q14 biasing it off. Therefore playback head 1 is forced on instead of head 2.

2 N1303 = Q1, Q2, Q3, Q4, Q5
2 N527 = Q6, Q7, Q8, Q9





III-3-9. Sequencing

The sequencing circuit is designed to fulfill the computer's housekeeping (warmup) requirements and bring it to an operable state without operator assistance. It provides an automatic system whose basic functions are:

- (1) to achieve known starting and controlled stopping conditions in the computer,
- (2) to protect the computer program from damage due to power line failures,
- (3) to force the operator's manipulation of the control panel switches in an orderly manner,
- (4) to clear and prevent operation of input and output devices whenever internal conditions make the program inoperable,
- (5) to monitor the drum speed and use it as a condition for allowing the computer to operate,
- (6) to signal the logic to look for the program start word.

The output signals from the sequencing circuit are:

W6 which blocks Fast Access record.

$\overline{W4}$ which blocks General Storage record, and forces the recording of zeros in Fast Access.

W4 which blocks the computer input-output system and pulls track selection to track 1.

These are produced and removed at various points in the circuit's operation, as the sequencing requirements are fulfilled.

It will simplify the description of this circuit if the functions of the twelve transistors which it contains are outlined separately.

Q1 and Q2 form a flip-flop which is normally off or reset. This fault flip-flop becomes set if any of the following occurs:

- (a) a momentary line failure,
- (b) the drum loses speed and thus the clock is lost,
- (c) power is turned off,
- (d) the program switch is operated.

It is reset either by operation of the restart switch or by removing power completely.

Q4 assumes an on-state when power is applied, and holds the output line W6 high. Q3 in conjunction with the large capacitor on its base delays the turn off of Q4 and, therefore, the resetting of W6. This delay is approximately one second. Q4 will also come on immediately if, for any reason, the fault flip-flop (Q1 - Q2) sets.

Q5 and Q6 monitor the clock and thus measure the drum speed. The connection of Q5 into the delay circuit in effect extends the delay indefinitely until the drum is up to satisfactory speed.

Q7 and Q8 turn on when power is applied. They are connected to form a minimum frequency detector, with Q8 remaining on as long as any negative pulses exist on L2 (indicating that one's exist in the Fast Access channel). When Fast Access is clear, Q8 will turn off. When computation begins, ones will again appear in Fast Access, causing Q8 to turn on again, but this is of no consequence once the operate-state has been reached.

The Q11 - Q12 flip-flop assumes the reset condition ($\overline{W4}$ high) when power is applied. It will be set (W4 high, meaning computer-ready) as soon as the above electronics have ensured that the proper conditions exist and the program start-line has been found. Setting of this ready flip-flop is the final step in the turn-on sequence. Q9 is an emitter follower and Q10 is an amplifier serving the ready flip-flop inputs.

III-3-9-a. Cold Start. When power is applied, the fault flip-flop (Q1 - Q2) comes on reset. Input W1 senses the -24 volt winding of the main supply power transformer and is a step-down representation of the line. Transient contact bounce from the power switch may cause W1 to temporarily appear as a low line to the fault flip-flop. The flip-flop is referenced to the zener voltage -6Z. A large capacitor in the power supply delays -6Z until after W1 stabilizes. This makes the circuit come on in the proper state.

The program and power switch contacts are both closed, providing -6Z at W3. Resistor R5 provides an off-bias for Q1, the Q1 emitter being more positive than -6Z by the drop through silicon diode D5. Q2 is biased off by D4 and R6.

Transistor Q4 comes on when power is applied while C6 charges through R15 to -24. There are no clock pulses at the start and thus Q6 is held on and this in turn forces Q5 on. Q5 then takes over supplying the drive to keep Q4 on.

As the drum approaches operating speed, clock pulses will begin to appear at the base of Q6. Each pulse discharges C7 and turns off Q6 and Q5 for as long as it takes R16 to recharge C7 back to the emitter reference voltage. When Q5 is off, capacitor C5 can start to charge via R11 and the Q3 base-to-emitter diode. The resultant base current turns Q3 on and its collector holds the common collector node at or near -6Z. The result is a continuous base drive to Q4, which holds W6 high blocking Fast Access record.

If $\overline{L2}$ arrives low when power is applied, then Q7 and Q8 will both

be on. If it arrives high, Q8 will be on because C9 will be discharged and R21 will provide a charging current thru the Q8 base.

The base drives and collector loads of the ready flip-flop are unbalanced when power arrives, forcing Q12 on and thus W4 high. Although the computer logic and the signals A29, A4 and A8 are unpredictable at this point, W6 is guaranteed high and therefore D10 will hold the AND-gate at the set input high, maintaining the initial conditions.

As the clock pulses appear more frequently, Q6 will be off for a larger portion of the time and the drive to Q4 will gradually become the total responsibility of Q3. Without assistance from Q5, R11 will ultimately gain charge on C5. Both the collector of Q3 and the R11 end of C5 will slowly rise toward the ground. When the Q3 collector reaches a point at which it no longer supplies sufficient drive to Q4, the latter will turn off, dropping W6 to the level of -6Z. W6 low allows recording to begin in Fast Access and W4, which is still high, guarantees that any such recording will be zeros.

In order to set the ready flip-flop, a low is required at the junction of D9 and D13, W6 is low already and the combination of A29 and A4 will go low each time the zero flag bit is decoded by the logic. However, the junction will remain high until Fast Access is cleared and Q8 goes off.

The capacitor C9 discharges with each negative pulse arriving at L2; between pulses it recharges. During both parts of this cycle Q8 remains on. However, when the ones on L2 cease for a period of time sufficient to charge C9 to the voltage of divider R23 and R24, Q8 will turn off. Then, when the zero flag bit appears again, the AND-gate will fall negative, applying drive to Q9 and Q11. Thus the ready flip-flop will set, making W4 high and W4 low.

The computer is now in its operate mode and the turn on sequence is complete. W4 is high, illuminating the ready lights, releasing selection and allowing output devices to operate, and W4 is low, allowing recording in General Storage.

III-3-9-b. Restart. If the operator should press the restart button, the junction of R9 and R14 is pulled to the -6Z reference voltage. C5 discharges rapidly, and Q4 is turned on immediately, pulling W6 high again. With W6 high, Fast Access recording is immediately blocked. R11 begins to charge C5, but the long delay described above again takes place. If the computer happens to be in an execute cycle, which might involve recording into General Storage, A8 will be low and there will be no immediate effect on the ready flip-flop. A8 will go high at the start of the next precess cycle, R28 will supply base drive to Q10, turning on Q12, and the ready flip-flop will reset. Thus W6 and W4 are once again high with W4 low, and the turn-on sequence begins automatically once more.

Had the computer been in a precess cycle when the restart switch was first closed, there would be no delay in resetting the ready flip-flop.

III-3-9-c. Reset. As indicated earlier, the fault flip-flop (Q1 - Q2) may set at any time in the event that low line voltage is detected, the

CIRCUITS

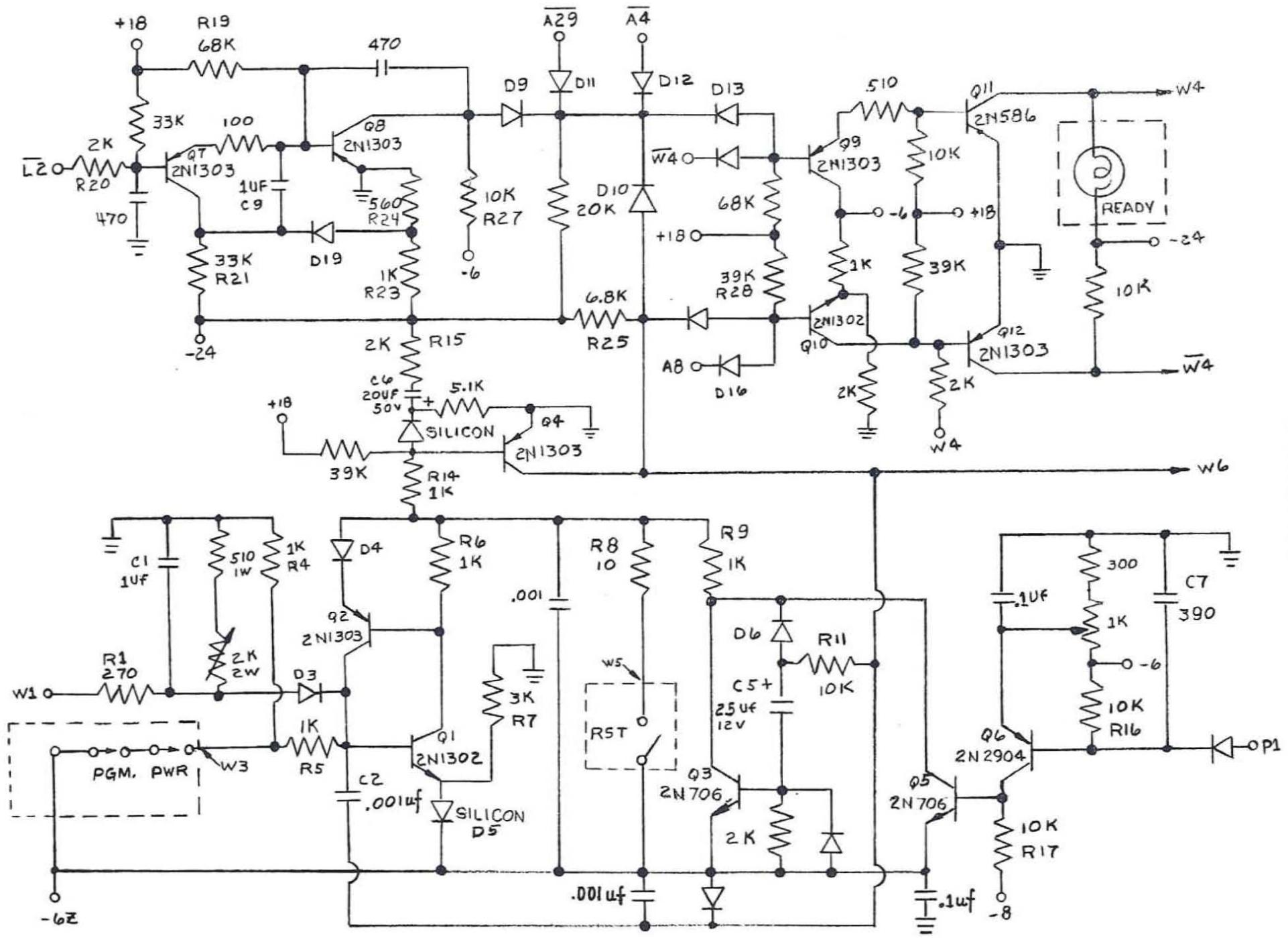
power switch is turned off, the program is switched or the clock frequency falls (loss of drum speed).

In the case of a normal turn-off or program change, the contacts serving W3 open early in the switch motion. Before any switching of programs or breaking of power lines actually occurs, the W3 circuit opens. The ground through R4 and R5 supplies turn-on drive for Q1, Q1 and Q2 regenerate on and, except for diode drops, the intersection of R9 and R14 is connected to -6Z. Thus Q4 is turned on and the shut-down sequence follows. Again, as above, the ready flip-flop resets with the first precess cycle.

The same sequence occurs in the event of a line failure, except that the set current for the fault flip-flop arrives via D3.

The setting of the fault flip-flop in the event of loss of clock frequency is less direct. When the clock pulses stop, or become too widely spaced, R16 charges C7 to the emitter reference voltage of Q6, with the result that both Q6 and Q5 turn on for at least a pulse. Q5 discharges C5 to a level which supplies base drive to Q4. The latter turns on, and, with or without further aid from Q5, Q3 and Q4 regenerate on. W6 rapidly goes high, initiating the turn-off sequence. Since Q3 and Q4 are not stable in the on condition and the turn-on sequence could be re-initiated if clock pulses were to re-appear, the rise of W6 is coupled into the fault flip-flop via C2, setting it as described before.

If this sequence was begun as a result of a normal turn-off, the DC power would ultimately be removed. If, however, the shutdown occurred for any other reason, the condition can be rectified by pressing the restart button, which shorts out the fault flip-flop and reinstates the condition prevailing when power was first applied. Should the cause of the turn-off persist (for example, a line failure or a clock failure), the computer will constantly be forced back into the shutdown state.



CIRCUITS

III-4 POWER CIRCUITS

III-4-1 Main Power Supply

The main power supply of the computer contains two conventional DC supplies (-24 and +18 volts) and a regulator circuit which supplies DC voltages -6, -6Z and -8.

When the POWER switch on the control panel is operated, the power relay is energized, closing the relay contacts in the input line and activating the drum motor, the I/O power supply, the fan and the main supply power transformer. The latter has primary taps for 105, 115 and 125 volts AC. Each step down secondary connects to a full-wave bridge rectifier followed by LC filtering. These produce the -24 and +18 volt computer DC voltages. Each transformer secondary has a high tap, No. 7 for +18 and No. 10 for -24, for use in margin testing.

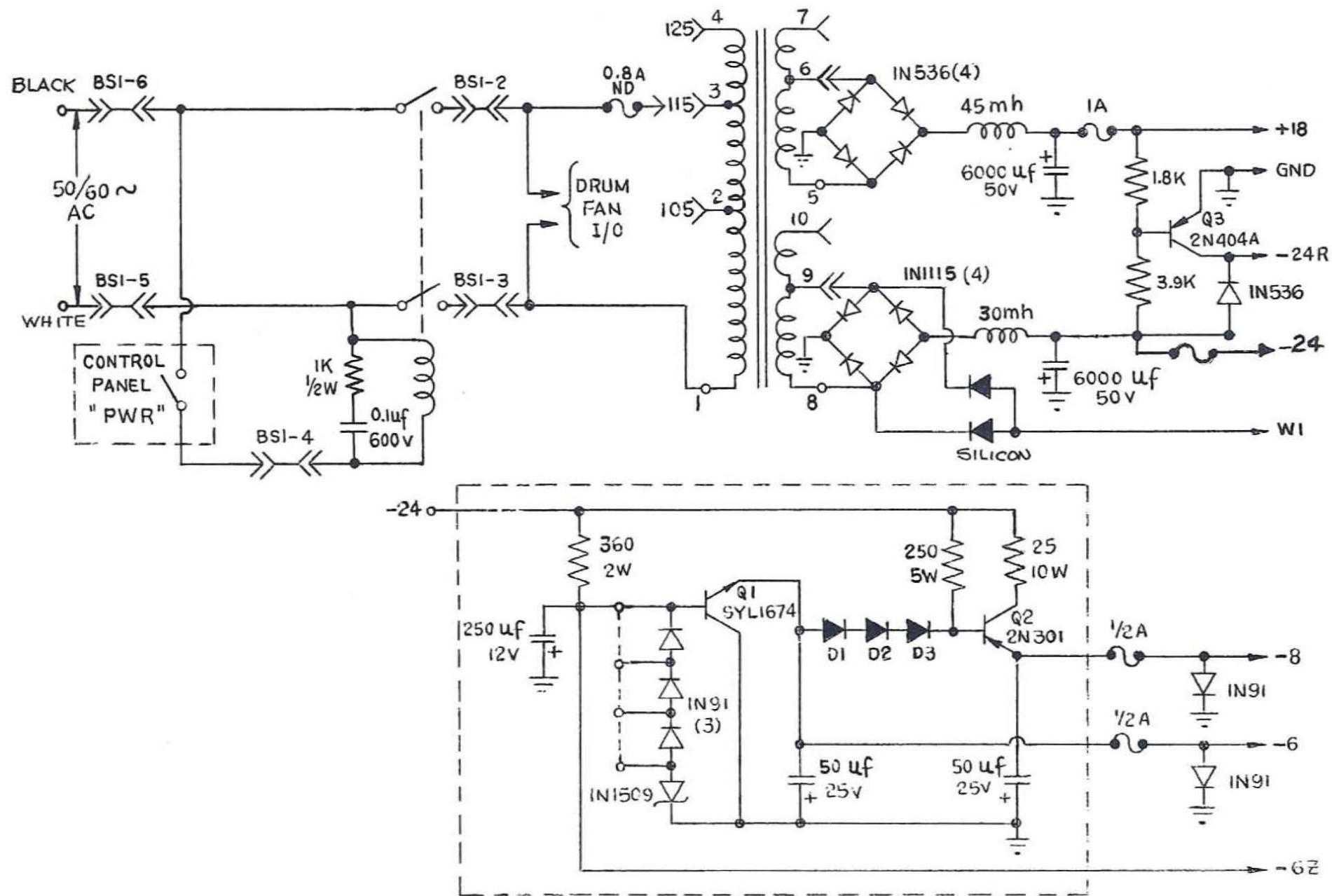
The regulator circuit produces constant voltages regardless of normal fluctuations of the AC line. The zener diode is reverse-biased and has a zener breakdown voltage of approximately 5.6 volts, while each of the three 1N91 diodes has a forward voltage drop of 0.3 volts. At the factory 1N91 diodes are unshunted as required to adjust the Q1 base voltage as close to -6 volts as possible. This base voltage is used as the reference voltage -6Z. The 250uf capacitor delays the application of this voltage, when power is turned on, until the other supplies have had a chance to build up.

Since the transistor Q1 is connected as an emitter follower, the emitter voltage is also regulated. The -6 output will be a few tenths of a volt more negative than the base of Q1 because of the drop through the base to emitter diode of the power transistor and through the fuse.

An important characteristic of the -6 volt supply is that it is not a true power supply. The source of energy for the "supply" comes from the circuits that it clamps. A circuit that is clamped at -6 volts is seeking to go more negative toward the -24 volt supply; the -6 volt clamp "supply" furnishes enough current to keep it from doing so. In effect, the -6 volt supply must act as a sink rather than a source of energy.

A resistor and diode arrangement is also used to apply the regulated voltage to the base of Q2. Silicon diodes D1-3 have a combined drop of approximately 2 volts which added to the voltage at the emitter of Q1 establishes the base of Q2 at approximately -8 volts. Q2 is also an emitter follower whose output will follow the input.

The -24R output provides a -24 volt reference to the Input-Output power supply through diode D4. Should the +18 volt supply fail, Q3 will be biased on, grounding the output at -24R. The signal W1 monitors the AC line through a full-wave rectifier, the inputs to which are taken from the -24V transformer secondary. Any momentary drop of AC voltage will be transmitted by W1 to sequencing.



III-4-2. Control Panel.

The control panel of the Monrobot XII consists of three switch assemblies -- PWR (Power), RST (Restart) and PGM (Program). The first turns on the power, the second allows the operator to return to the program start line without shutting the computer off and the third permits the operator to change from one of the alternate program tracks (T2A or T2B) to the other.

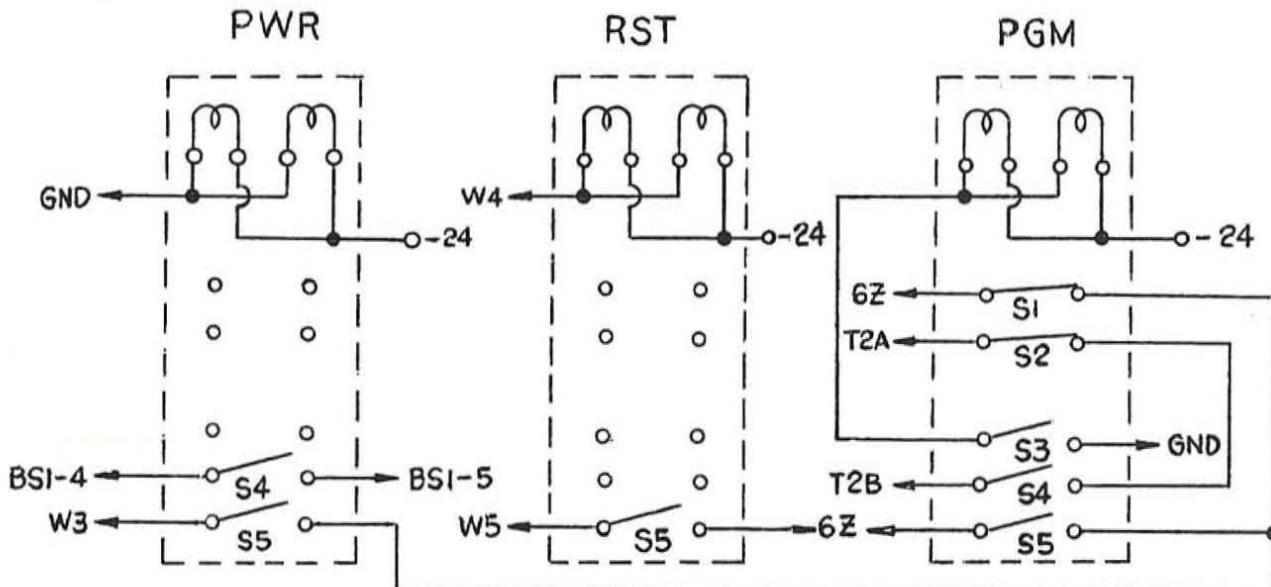
The panel switch assemblies contain form A (normally open) reed switches. They are actuated by a magnet which is attached to the switch button and moves when the switch button is operated. In the normal or "off" position, the magnet is over S1 and S2. In the depressed or "on" position, the magnet is over S3, S4 and S5.

It can be seen from the figure that depressing the PWR button closes S4 and S5 on that assembly. Switch S5 -- in conjunction with S1 or S5 on the PGM switch, one of which is normally closed -- produces signal W3 to the sequencing circuit. S4 on the PWR switch assembly completes the power relay circuit on the main power supply.

The lamps in the RST switch light when the W4 signal goes high, which indicates that the computer is ready for operation. Operating this switch produces the W5 signal, which resets the sequencing circuit to its initial state. The RST switch is of the momentary type. The two parallel RST lamps act as the load resistor for the sequencing ready flip-flop output W4. The ready flip-flop will perform properly if only one lamp is operating but the flip-flop will fail if both lamps burn out.

The PGM switch is of the push-push type. In the normal position, S1 and S2 are closed. S2 connects track 2A into the track selection system. When the switch button is operated, the first switch action is to open S1. This opens the W3 line and resets sequencing so that no recording can take place while tracks are being changed. As the magnet continues to move, S2 is opened. Then in sequence, S3, S4 and S5 are closed. S3 applies voltage to the PGM switch lamps and S4 connects track 2B to selection. Note that S5 closes last; when this happens, W3 is reconnected to sequencing.

When the PGM switch is moved from the depressed position, a similar sequence takes place. The first action is to open S5 to reset sequencing and the last action is the closing of S1, which again supplies signal W3 to sequencing.



III-5. INPUT-OUTPUT CIRCUITS

As described in earlier chapters, the Monrobot XII can utilize as many as three input and three output devices in the same program. The computer electronics for all input and output channels is contained on one input-output circuits board. The input-output (buffer) power supplies are mounted separately from the electronics. The number 1 input-output power supply, which is located under the input-output board, supplies -24 volts for the input-output electronics, for channel 1 and in some cases channel 2. The number 2 input-output power supply, if installed, provides separate -24 volts for channel 2. Channel 3 is reserved for devices which have their own power supplies and no corresponding supply is provided in the computer.

The devices are connected into the computer by input and output connectors located on the computer rear wall. Input and output connectors are polarized differently to prevent improper connections. The top pair of connectors is always considered channel 1, the second pair channel 2 and the lowest pair, if installed, channel 3. The signals available at the pins of the input and output connectors are tabulated below.

| Pin | INPUT CHANNEL | | | OUTPUT CHANNEL | | |
|-----|---------------|------------|------------|----------------|------------|------------|
| # | 1 | 2 | 3 | 1 | 2 | 3 |
| 1 | X11 | X21 | X31 | N1 | N1 | N1 |
| 2 | X12 | X22 | X32 | N2 | N2 | N2 |
| 3 | X13 | X23 | X33 | N3 | N3 | N3 |
| 4 | X14 | X24 | X34 | N4 | N4 | N4 |
| 5 | X15 | X25 | X25 | N5 | N5 | N5 |
| 6 | X16 | X26 | X36 | N6 | N6 | N6 |
| 7 | X17 | X27 | X37 | N7 | N7 | N7 |
| 8 | X18 | X28 | X38 | N8 | N8 | N8 |
| 9 | | | | | | |
| 10 | | | | | | |
| 11 | | | | | | |
| 12 | | | | | | |
| 13 | IDI | ID2 | ID3 | NC | NC | NC |
| 14 | | | | | | |
| 15 | | | | | | |
| 16 | GND | GND | GND | GND | GND | GND |
| 17 | SHLD | SHLD | SHLD | SHLD | SHLD | SHLD |
| 18 | | | | | | |
| 19 | | | | | | |
| 20 | <u>S11</u> | <u>S21</u> | <u>S31</u> | <u>OD1</u> | <u>OD2</u> | <u>OD3</u> |
| 21 | W40 | W40 | W40 | | | |
| 22 | | | | | | |
| 23 | <u>S14</u> | <u>S24</u> | <u>S34</u> | E16 | E26 | E36 |
| 24 | +18 | +18 | +18 | +18 | +18 | +18 |
| 25 | B.GND | B.GND | | B.GND | B.GND | |
| 26 | -24B | -24B | | -24B | -24B | |
| 27 | FR.G. | FR.G. | FR.G. | FR.G. | FR.G. | FR.G. |

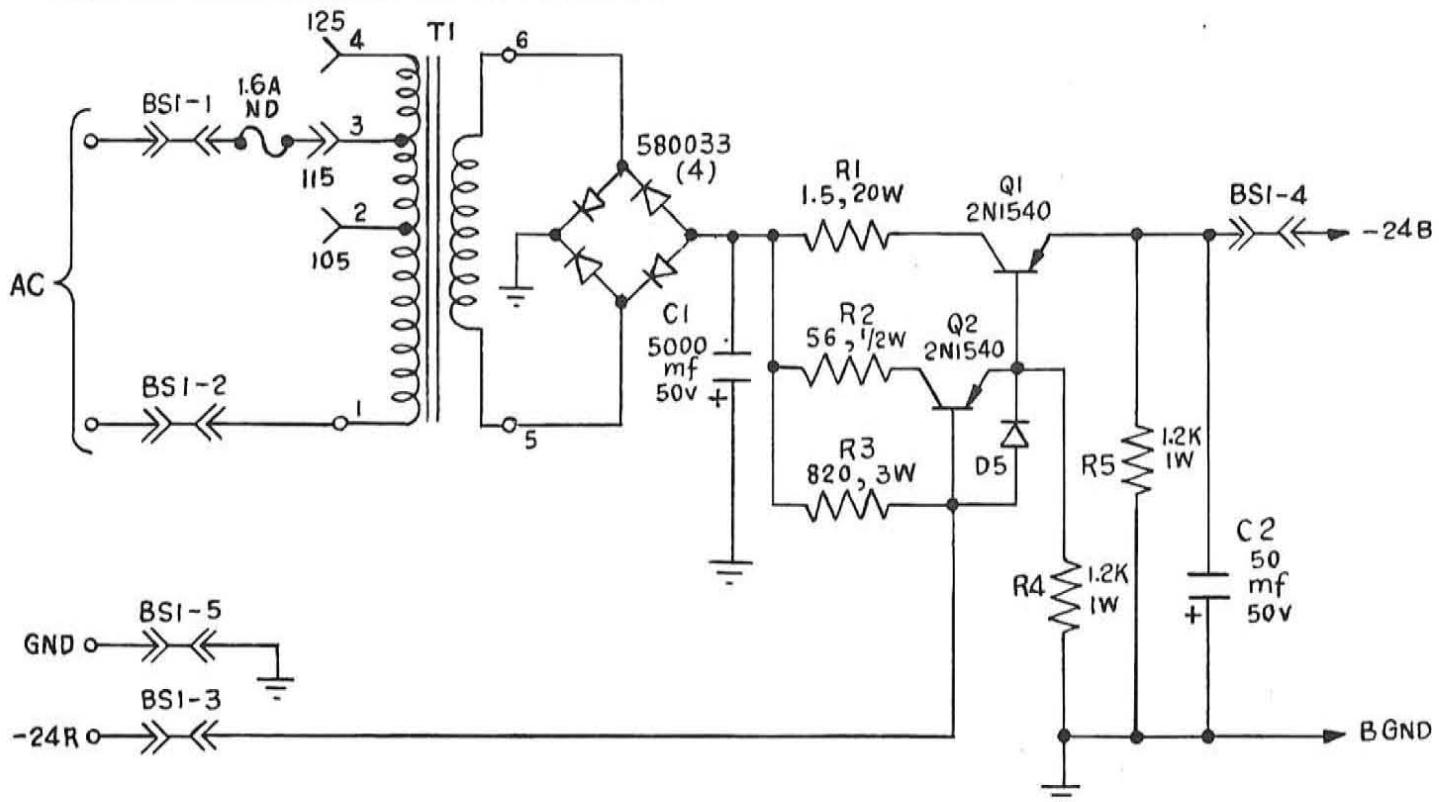
III-5-1. Single Input-Output Power Supply.

This regulated supply will provide up to 3 amps at -24 volts to an input-output load.

The transformer steps down the 115 VAC line to 25.5 VAC (rms). The diode bridge rectifier produces a negative pulsating DC voltage from this. Filter capacitor C1 reduces the ripple of the negative voltage.

This negative voltage (-30.6 VDC, ripple 3.5 volts p.p.) is applied to a two-transistor (Q1 and Q2) series regulator, which uses the -24 VDC from the computer supply (-24R) as its reference. Q1 and Q2 operate as class A amplifiers and under normal conditions should never saturate. The output of this regulator is -24 VDC with ripple content approximately 100 mv p.p. The output load can vary from 0 to 3.0 amps and the voltage output should remain stable.

If the output should be shorted to ground, resistor R1 serves as a current limiting resistor for Q1 and sufficient current will flow through R3 to reduce the voltage across Q1 and Q2, until the fuse blows. Diode D5 is part of the protective circuit which grounds the output of this supply if the +18 volt fuse blows. The state of Q2 is unpredictable under these conditions. The diode will guarantee that ground is applied to the base of Q1. The emitter output must therefore also be near ground. Resistor R5 and capacitor C2 form an output filter which is designed to prevent oscillations in the supply.

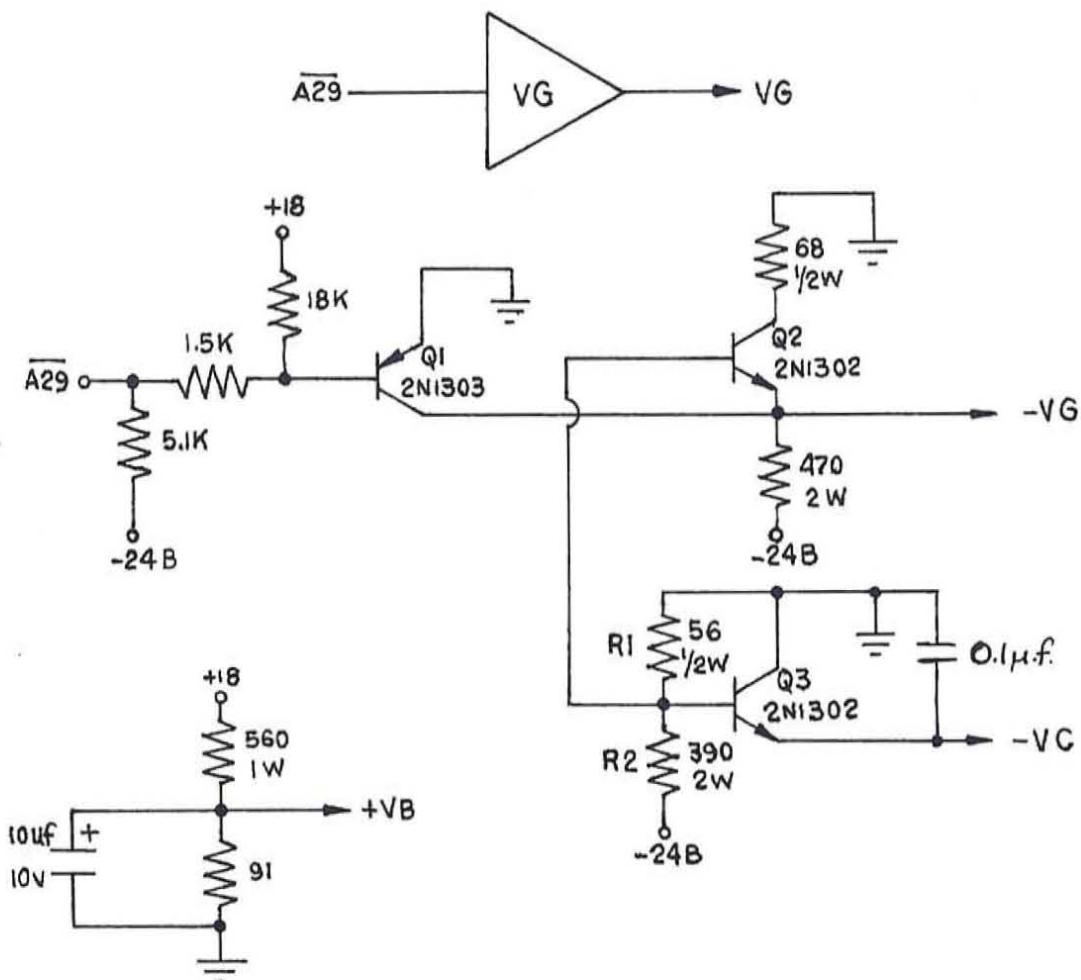


III-5-2. Bias Supplies.

Three local bias voltages are used in the Input-Output circuits. The bias supplies are:

$+VB$: +2.4 volts
 $-VC$: -3.0 volts
 $-VG$: -3.0 volts (t₃₆ - t₂₈)
 GROUND (t₂₉ - t₃₅)

The $+VB$ supply is produced by a voltage divider between +18 and ground. The VG circuit supplies both a $-VC$ and a $-VG$ bias supply. Q₂ and Q₃ are emitter followers and are under control of the voltage divider R₁ & R₂, which supplies -3.0 volts to the base of these two transistors. The output of Q₂ (-VG) is under control of the Q₁ amplifier. When A₂₉ is at ground, Q₁ is off and $-VG = -3.0$ volts. When A₂₉ goes low (from t₂₉ to t₃₅) Q₁ turns on grounding VG for that period.

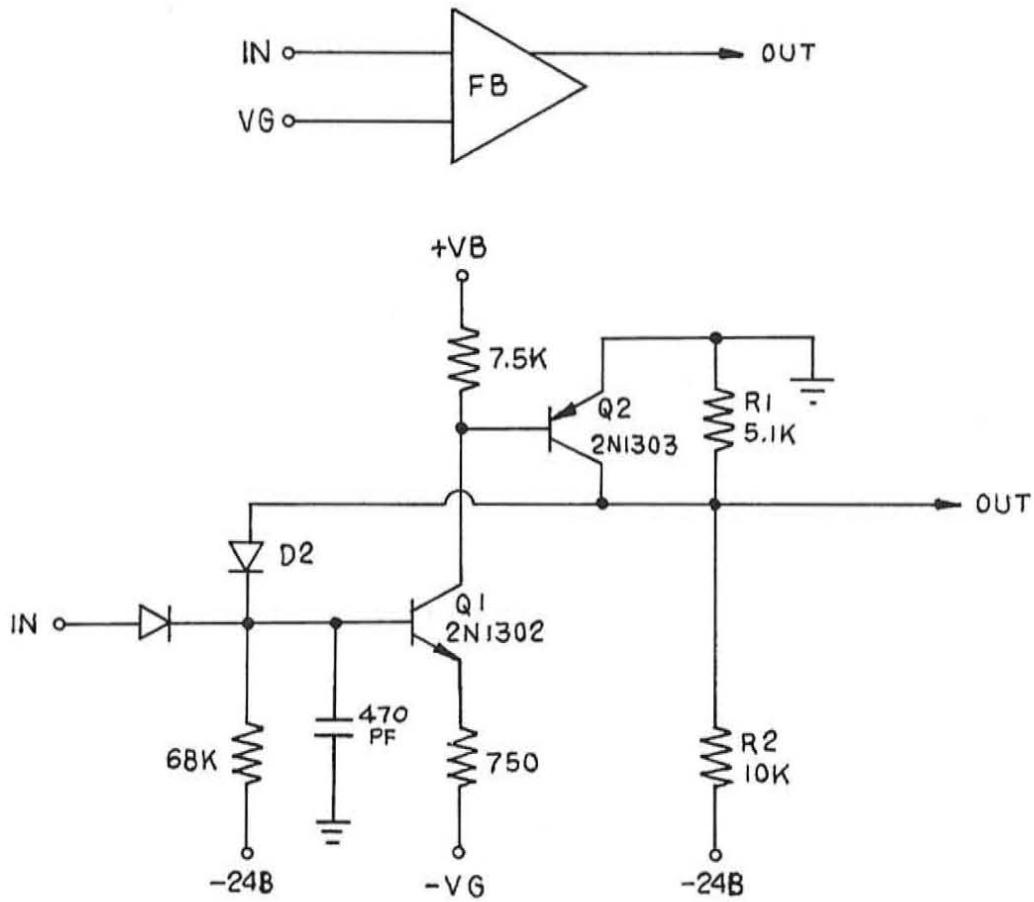


III-5-3. The Output Buffer Circuit.

The Output Buffer (FB) circuit is a memory circuit used in a register to assemble output information until the computer indicates whether the information should be sent to an output device or to the device memory.

Initially, the input to the circuit is low, keeping Q1 and Q2 turned off. The output is then held low by the R1, R2 voltage divider. A high input causes the base of Q1 to go to ground, turning Q1 on and, in turn, pulling Q2 on and producing a high output (N1 through N7) from the circuit. Feedback diode D2 holds the circuit on until, at time t_{29} , -VG goes to ground, shutting off Q1 and dropping the output to low again.

Resistors R1 and R2 are omitted on N1 through N6 because the loads supply equivalent resistances.



III-5-4. The Device Memory Circuit

The Device Memory (FC) circuit is used to control which input and output devices are to be operable. The circuit has two inputs. One represents the information from the output buffer (N1 - N6); the other -- the DS input -- represents the command to select devices.

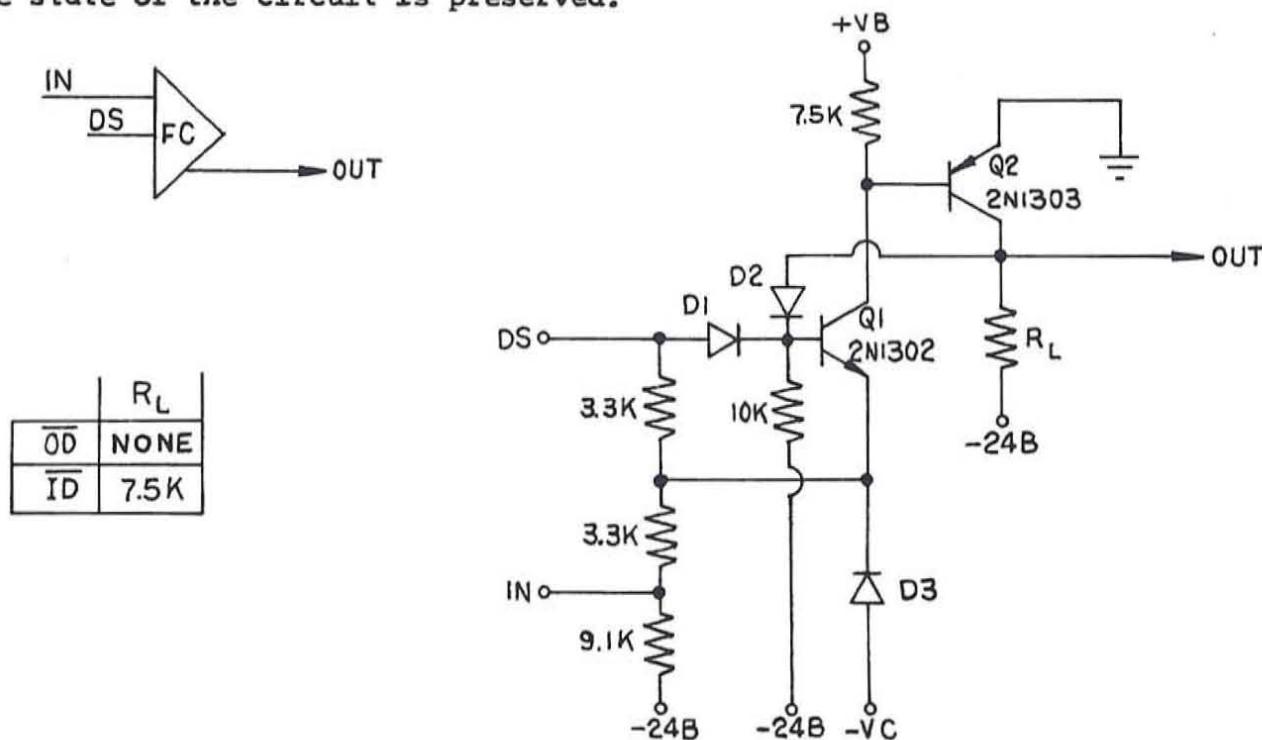
There are three sets of input conditions of interest in this circuit, as follows:

Case 1. When the DS signal goes high and a buffer input is high, diode D3 will be back-biased, making the emitter of Q1 high. Q1 will turn off, removing the drive to the base of Q2. Q2 will then turn off and the output will go low. The D2 feedback diode will then be back-biased. The DS input then goes low, followed by the buffer input going low, and diode D3 turns on, clamping the emitter of Q1 to -VC. However, the base of Q1 is more negative than its emitter, keeping Q1 off. The low output indicates that the device is selected.

Case 2. When the DS signal goes high and the buffer input is low, the emitter of Q1 being at -3 volts turns Q1 on, supplying drive to the Q2 base. This turns Q2 on. When the DS signal returns to low, Q2 being on turns on diode D2 and this will keep the output high. The high output keeps the device unselected.

This case shows that, if a device had previously been selected and the programmer wants to program a device select operation, he must always re-select devices to prevent resetting.

Case 3. If the DS signal is low and the buffer input is high or low, the state of the circuit is preserved.

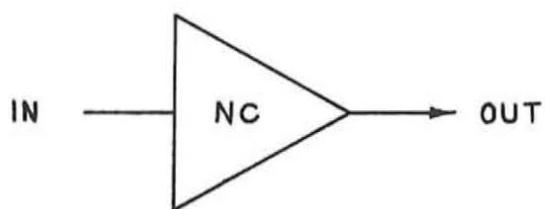
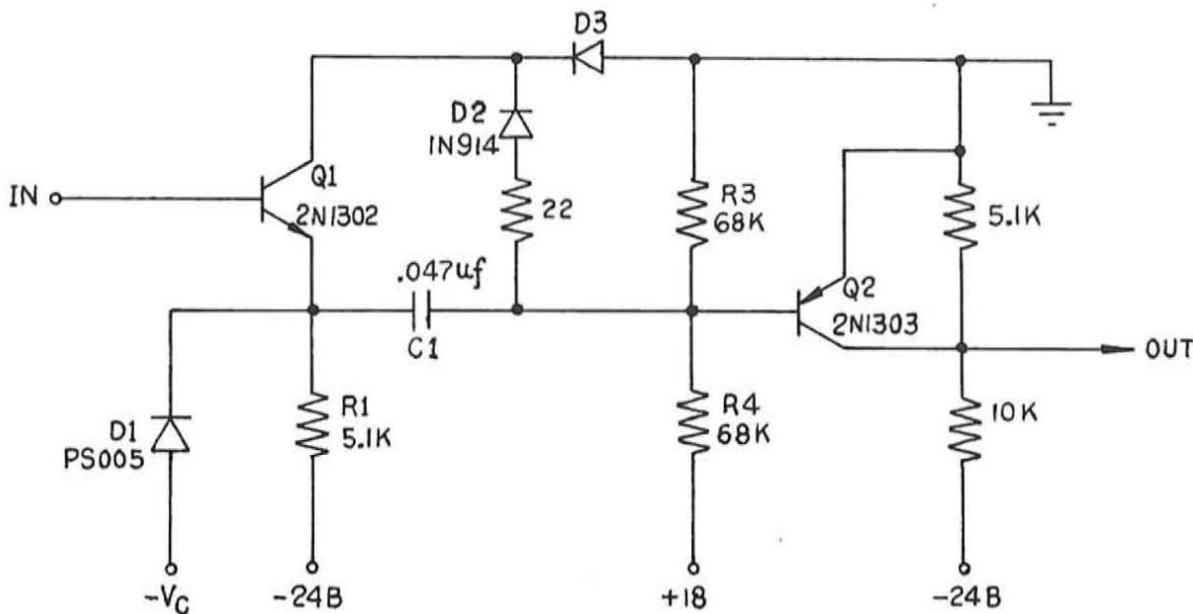


III-5-5. The Output Common Circuit.

The Output Common (NC) circuit consists of an emitter follower and an inverter. In its initial state Q1 and Q2 are off and the output of Q2 is held low by the voltage divider between ground and -24 volts. C1 is initially charged, so that the base of Q2 is at +9 volts and the emitter of Q1 is at -4 volts.

At time t_7 the input goes to ground and C1 discharges through Q1 via D2. When the input goes low, Q1 turns off. A negative pulse is produced at the base of Q2 due to the coupling action of C1 and Q2 turns on. Q2 remains on until C1 charges sufficiently negative to turn D1 on. At this point C1 begins to charge through R4 making the base of Q2 positive and turning Q2 off.

The NC output is therefore delayed one bit time and stretched to approximately 40 microseconds.

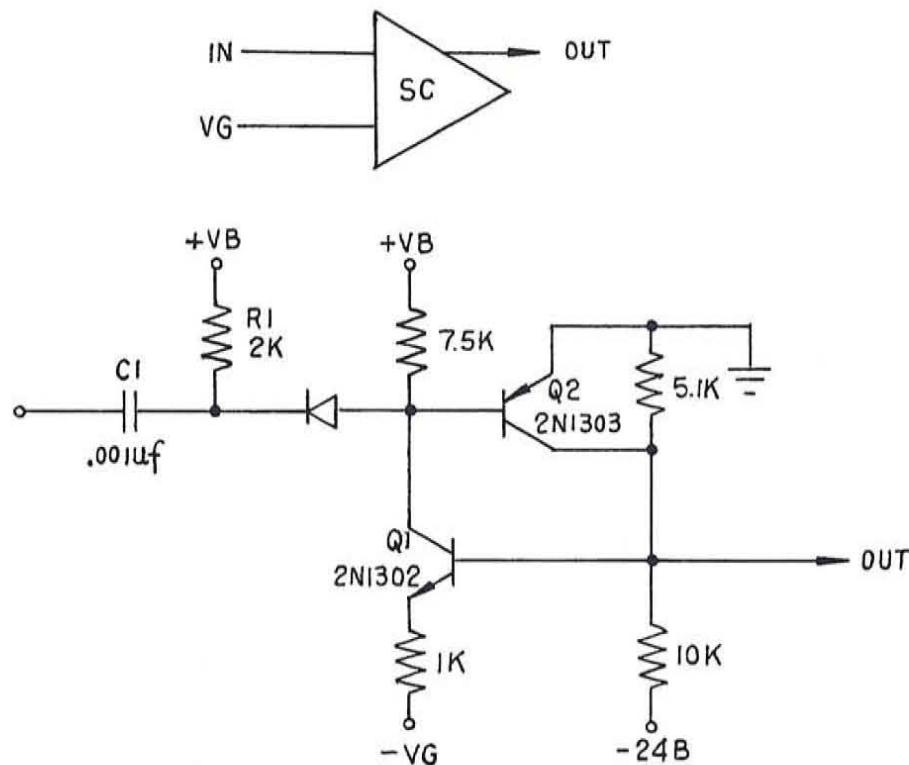


III-5-6. The Input Clear Circuit

When the character read-in process has been completed, at time t_7 of a successful input, the Input Clear (SC) circuit is used to signal the selected input device that the character code has been accepted by the computer, and that the device may set up another character. To insure that the read-in of a character is completed, this circuit action is delayed one bit-time. The output signal is stretched so that sufficient time is available for the proper operation of any driven circuits.

Transistors Q1 and Q2 are initially off. The input goes high at t_7 and remains high for one bit-time. When it goes low again, the negative voltage spike produced by the differentiating action of C_1 and R_1 pulls the base of Q2 negative, turning Q2 on. This results in a high output and also turns on Q1. Q1 acts to keep Q2 on even though the input signal has gone low.

At t_{29} , V_G goes to ground, turning off Q1. This allows the $+V_B$ bias supply to pull the base of Q2 high and turn off Q2, resetting the output.

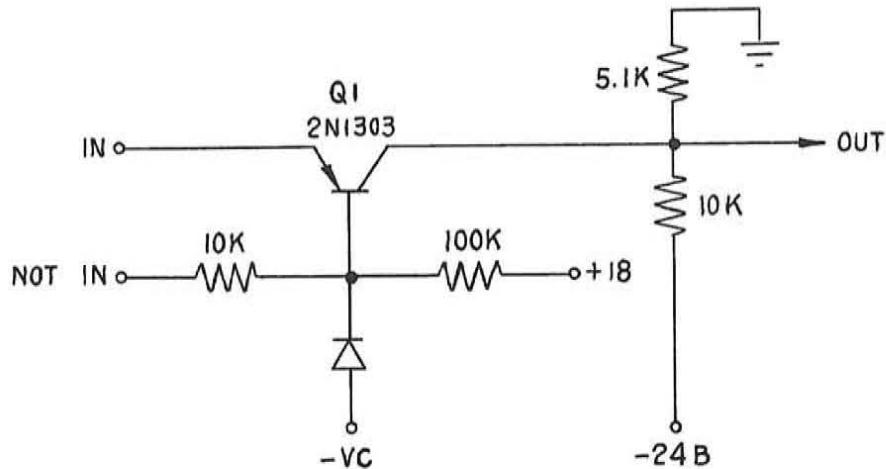
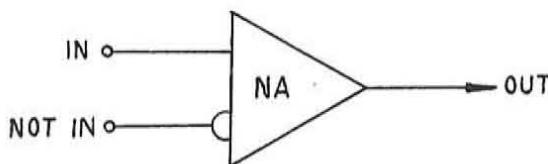


III-5-7. The AND-Not Circuit

The AND-Not (NA) logic circuit produces a high output only when its normal input is high and its not-input is low.

Whenever the not-input is high the base of the transistor is held positive and the transistor cannot be turned on regardless of the state of the normal input. When the transistor is off the output of the circuit is held low by the voltage divider from ground to -24 volts.

When the not-input is low, the base of the transistor is clamped at approximately -3 volts. A high at the normal input can now turn the transistor on and produce a high output. Note that when the normal input is high, and the emitter of the transistor goes to ground, the base of the transistor will follow the emitter and be slightly below ground.

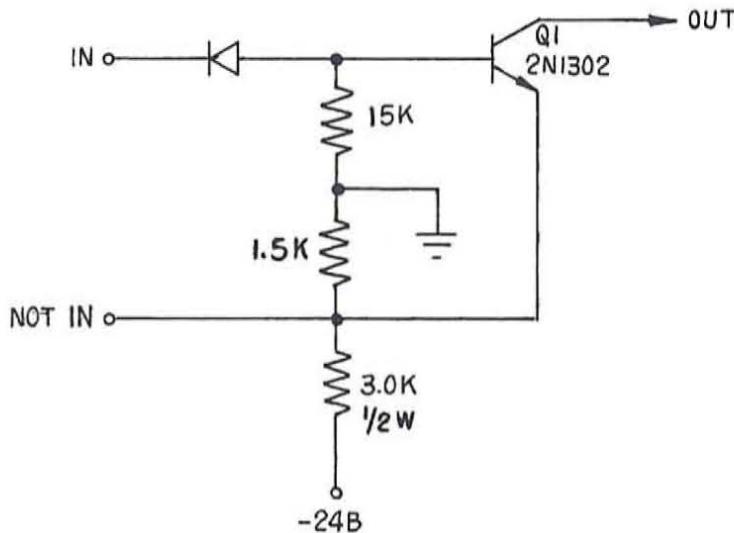
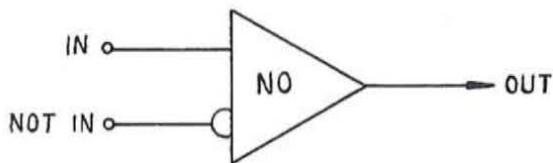


III-5-8. The OR-Not Circuit

The OR-Not (NO) logic produces a low output only when its normal input is high and its not-input is low. For all other input combinations the transistor remains off.

When the not-input is high, the emitter of the transistor is at ground and it cannot be turned on regardless of the state of the input signal.

When the not-input is low, a high at the normal input will turn on the transistor and produce a low output.

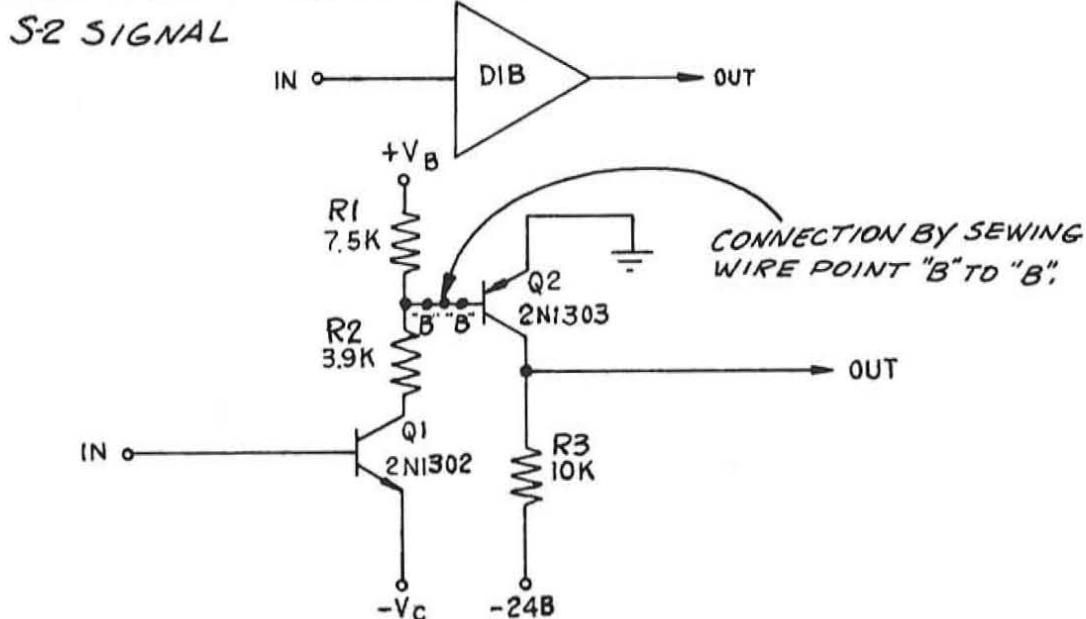
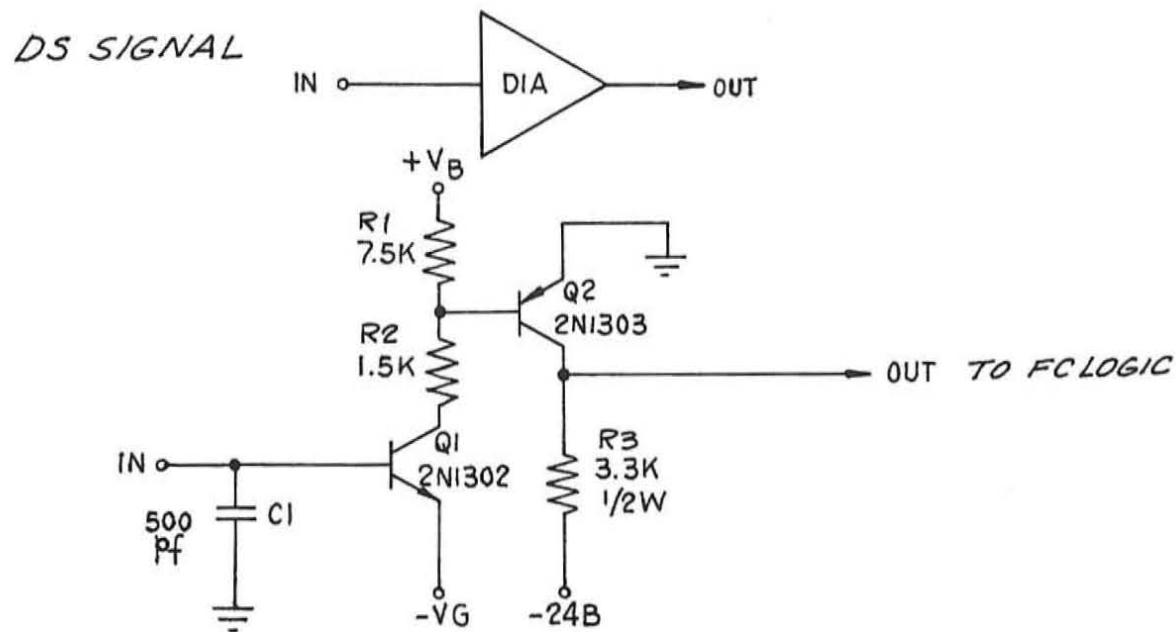


III-5-9. Driver Circuits

The DIA and DIB circuits are simple double inverters which are used to repower signals. Only the real output is available from these circuits.

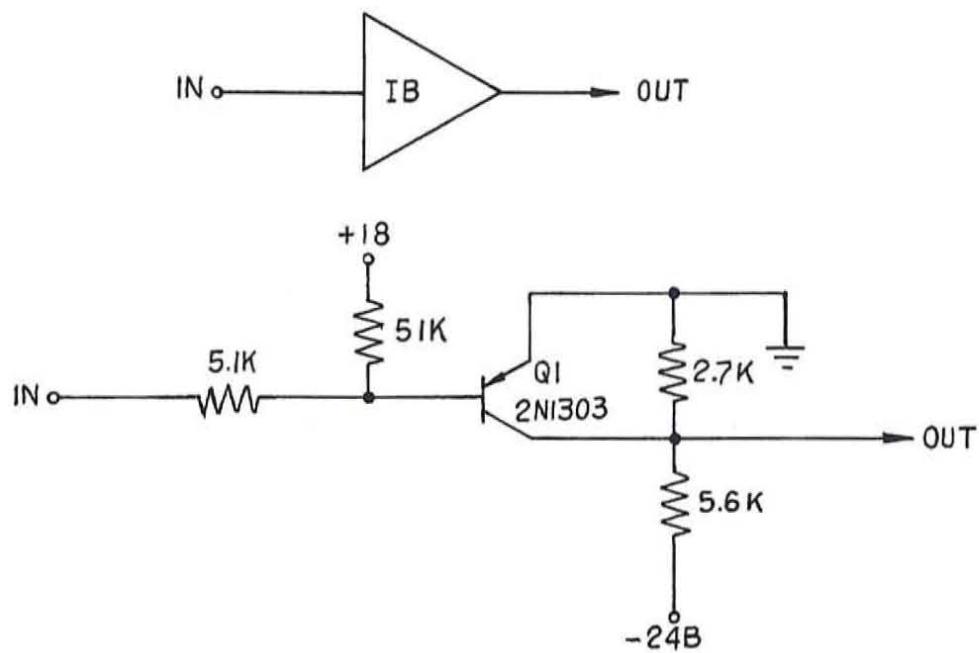
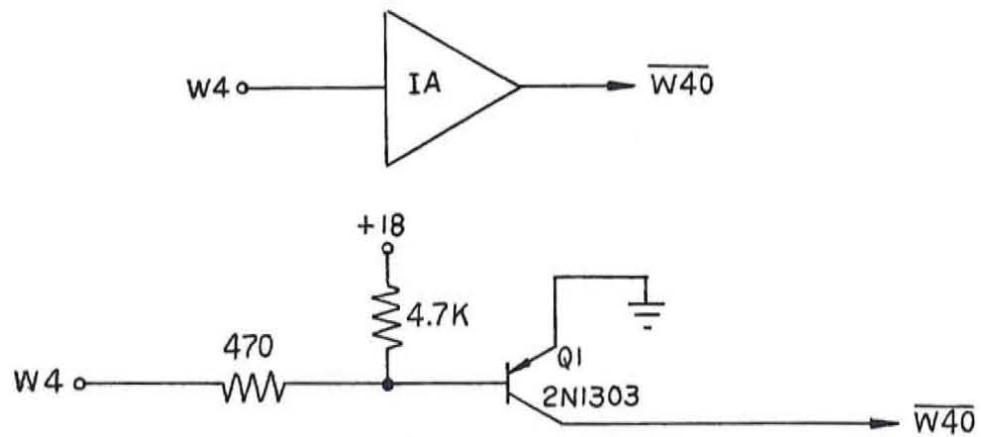
A high input turns on Q1. The base of Q2 is pulled negative and Q2 will turn on producing a high output. When the input is low, the output will be low.

Capacitor C1 prevents false outputs due to crossovers.



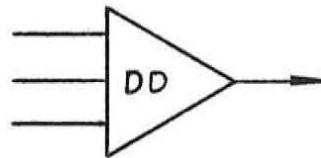
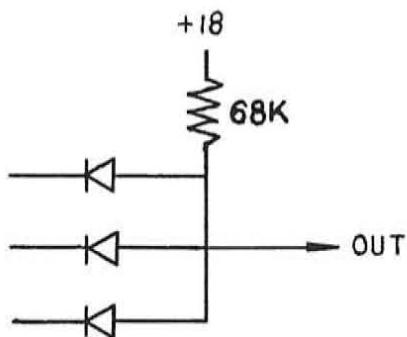
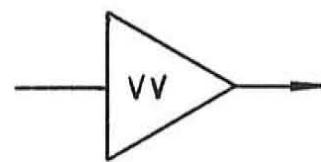
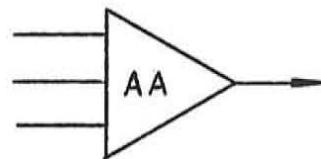
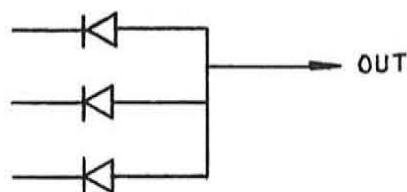
III-5-10. Inverter Circuits

The IA and IB circuits are simple inverters. The IA circuit is used to invert the sequencing signal W4 and is called the warm-up inverter. The IB circuit is used to invert input device memory (ID) signals and form the input device common.



III-5-11. Logic Gates

The special logic gates shown below are used in the input-output circuits. Where load resistors are omitted equivalent loads are supplied by the driven circuits.



SECTION IV

GENERAL PROGRAMMING & OPERATING RULES

| | Page |
|------------------------------------|-------|
| IV-1. PROGRAMMING | IV-1 |
| IV-1-1. Applications | IV-1 |
| IV-1-2. Procedures | IV-1 |
| IV-1-3. Delay | IV-8 |
| IV-1-4. Leap | IV-9 |
| IV-1-5. Slide | IV-10 |
| IV-1-6. Divide By Five | IV-11 |
| IV-1-7. Example | IV-12 |
| IV-2. STARTING AND OPERATING | IV-15 |

SECTION IV

GENERAL PROGRAMMING AND OPERATING RULES

A basic understanding of the rules by which the Monrobot XII is programmed and operated is necessary to anyone working with the computer.

IV-1. PROGRAMMING

IV-1-1. Applications.

As indicated briefly in Section I, the Monrobot XII is designed around the "micro-programming" concept. This technique utilizes routines of stored basic micro-commands to create a number of more involved commands, such as Multiply, Compare, Bring Total, and so on. Thus, the command Multiply might be created by a micro-routine using Add Conditional and Right Shift commands. In many computers, these more involved commands, such as Multiply, would be wired into the machine; the micro-programming technique allows the implementation of commands for many types of programs with a minimum of hardware. A series of micro-routines -- coordinated to fit into the available storage -- is called a micro-program.

Customer programs are written by calling for these micro-routines by name. A customer application consisting of a sequence of these micro-routines is called a macro-program and the micro-routines can be called macro-commands.

In essence, the function of the micro-program is to simulate those operations which are not built into the computer's hardware. The function of the macro-program is to determine which micro-routines are to be performed and in which sequence they are to be carried out.

The macro- and micro-technique has a wide flexibility in application. The computer may be put to a number of uses -- for example, invoicing, scientific calculation, etc. -- and each of these may vary considerably from one specific case to another as to types of data used, output content, and so on. The use of the macro-micro concept permits general programs to be developed for each type of usage. Each of these micro-programs can then be used in various ways via different macro-programs, depending on individual customer requirements.

A frequent use of General Storage track 2B is to contain a micro-routine which allows the loading of a macro-program in the field. A given customer can then change macro-programs and make one Monrobot XII perform several jobs.

IV-1-2. Procedures.

All of the micro-commands are listed in figure IV-1, with their programming symbols or letter-codes and the influence of each command on aq and K . The programmer uses the symbols shown here to devise a program flow-chart, which presents in direct graphic form the program sequence to be executed.

The flow-chart form has the execute words arranged in vertical blocks. When this chart is completed, it is transferred to a set of coding-chart forms which list the drum registers by track and in numerical order. Information

| Command | Binary Code | Program Symbol | Influence on K | Influence on a_0 |
|--------------------------------------|-------------|----------------|---|-------------------------------------|
| Skip | 0000 | - | Preserved | 0 |
| Quadruple Left Shift (Track 1, 3) | 0001 | QLS | Preserved | 1 |
| Leap (Track 2, 4) | 0001 | L | Preserved | $c_0 \rightarrow a_0$ |
| Input: Successful (Track 1, 3) | 0010 | IN | 1 | Parity bit of input character |
| Unsuccessful | | | 0 | Debris |
| Output: Successful (Track 2, 4) | 0010 | P | 1 | 0 |
| Unsuccessful | | | 0 | 0 |
| Complement | 0011 | CM | 1 if a_{36} changes from 1 to 0, 0 if not. | 0 |
| Change Track | 0100 | CT | Preserved | Preserved |
| Left Shift | 0101 | LS | $a_{36} \rightarrow K$ | $K \rightarrow a_0$ |
| Transfer | 0110 | T | Preserved | 0 |
| Add | 0111 | A | 0 if no overflow, 1 if overflow. | $K \rightarrow a_0$ |
| Jump | 1000 | J | Preserved | 0 |
| Times Five (Track 1) | 1001 | XF | Number dependent | 0 |
| Partial Divide by Five (Track 3) | 1001 | PD | Number dependent | 0 |
| Slide (Track 2, 4) | 1001 | SL | 0 | 1 if min. duration, 0 otherwise. |
| Jump Conditional (Branch) | 1010 | B | Preserved | 0 |
| Delay Branch | 10110 | DB | Preserved | 0 |
| Delay Jump | 10111 | DJ | Preserved | 0 |
| Write | 1100 | W | Preserved | Preserved |
| Right Shift | 1101 | RS | $a_1 \rightarrow K$ | $a_1 \rightarrow a_0$ |
| Add Conditional: Successful | 1110 | AC | 1 if overflow, 0 if no overflow. | 0 |
| Unsuccessful | | | 0 | 0 |
| Extract | 1111 | E | 1 | 0 |
| Clear | 1111 | CL | 1 | 0 |

Figure IV-1. The Command and Their Influence on K and a_0 .

GENERAL PROGRAMMING AND OPERATING RULES

from this coding-chart form is used to punch the micro-loading tape. The special control codes needed on the micro-tape are discussed in the section on the Loader.

The programming symbols of figure IV-1 are also used on the coding-chart. Constants and special numbers are represented in binary or hexadecimal notations, as follows:

| <u>Decimal Number</u> | <u>Binary</u> | <u>Hexadecimal</u> |
|-----------------------|---------------|--------------------|
| 0 | 0000 | 0/- |
| 1 | 0001 | 1 |
| 2 | 0010 | 2 |
| 3 | 0011 | 3 |
| 4 | 0100 | 4 |
| 5 | 0101 | 5 |
| 6 | 0110 | 6 |
| 7 | 0111 | 7 |
| 8 | 1000 | 8 |
| 9 | 1001 | 9 |
| 10 | 1010 | S |
| 11 | 1011 | T |
| 12 | 1100 | U |
| 13 | 1101 | V |
| 14 | 1110 | W |
| 15 | 1111 | X |

The 75 registers on each track are numbered in the arrangement shown in figure IV-2. Note that this results in the following sequence:

.....1, 39, 2, 40, 3, 41,.....36, 74, 37, 75, 38, 1, 39,.....

This is done because of the alternate precess-execute word operation sequence of the computer. The numbering of the consecutive execute words, which is when the programmer does his data processing, is simply:

.....1, 2, 3,.....36, 37, 38, 39, 40, 41,.....74, 75, 1,.....

Note again that it requires two drum revolutions to gain access to the same drum register during an execute word (unless the Leap command is used).

On all the programming forms, three-digit register numbers are used. The first or left-hand digit represents the track, while the second and third represent the register on that track. Normally, this number specifies the location of the word with which it is associated. This location is the address of the word, although it is not an address to which the computer can simply be directed in the usual manner. This is because, in the Monrobot XII, addresses are not included with micro-instructions.

The programmer must make certain that one of the commands in each instruction word addresses the program to a new instruction word. Figure IV-3 shows the new instruction words accessible from any drum sector using a Jump or Jump Conditional command.

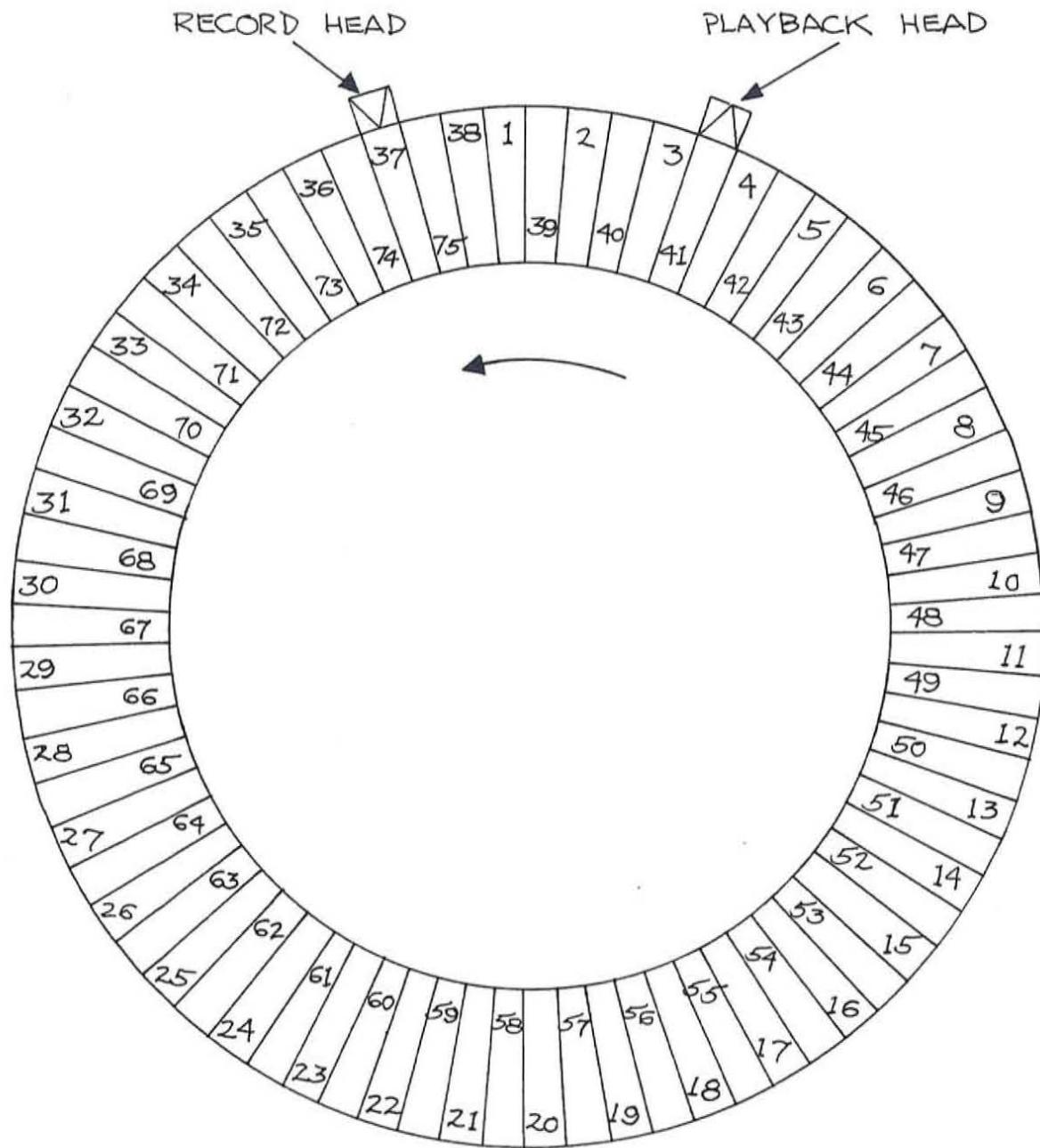


Figure IV-2. The Drum Register Sequence

| | AUT | J | AUT | J |
|----|---|---|-----|---|
| 1 | J 3 J 4 J 5 J 6 J 7 J 8 J 9 J10 J11 J12 | | 39 | J41 J42 J43 J44 J45 J46 J47 J48 J49 J50 |
| 2 | J 4 J 5 J 6 J 7 J 8 J 9 J10 J11 J12 J13 | | 40 | J42 J43 J44 J45 J46 J47 J48 J49 J50 J51 |
| 3 | J 5 J 6 J 7 J 8 J 9 J10 J11 J12 J13 J14 | | 41 | J43 J44 J45 J46 J47 J48 J49 J50 J51 J52 |
| 4 | JJ6 J 7 J 8 J 9 J10 J11 J12 J13 J14 J15 | | 42 | J44 J45 J46 J47 J48 J49 J50 J51 J52 J53 |
| 5 | JJ7 J 8 J 9 J10 J11 J12 J13 J14 J15 J16 | | 43 | J45 J46 J47 J48 J49 J50 J51 J52 J53 J54 |
| 6 | J 8 J 9 J10 J11 J12 J13 J14 J15 J16 J17 | | 44 | J46 J47 J48 J49 J50 J51 J52 J53 J54 J55 |
| 7 | J 9 J10 J11 J12 J13 J14 J15 J16 J17 J18 | | 45 | J47 J48 J49 J50 J51 J52 J53 J54 J55 J56 |
| 8 | J10 J11 J12 J13 J14 J15 J16 J17 J18 J19 | | 46 | J48 J49 J50 J51 J52 J53 J54 J55 J56 J57 |
| 9 | J11 J12 J13 J14 J15 J16 J17 J18 J19 J20 | | 47 | J49 J50 J51 J52 J53 J54 J55 J56 J57 J58 |
| 10 | J12 J13 J14 J15 J16 J17 J18 J19 J20 J21 | | 48 | J50 J51 J52 J53 J54 J55 J56 J57 J58 J59 |
| 11 | J13 J14 J15 J16 J17 J18 J19 J20 J21 J22 | | 49 | J51 J52 J53 J54 J55 J56 J57 J58 J59 J60 |
| 12 | J14 J15 J16 J17 J18 J19 J20 J21 J22 J23 | | 50 | J52 J53 J54 J55 J56 J57 J58 J59 J60 J61 |
| 13 | J15 J16 J17 J18 J19 J20 J21 J22 J23 J24 | | 51 | J53 J54 J55 J56 J57 J58 J59 J60 J61 J62 |
| 14 | J16 J17 J18 J19 J20 J21 J22 J23 J24 J25 | | 52 | J54 J55 J56 J57 J58 J59 J60 J61 J62 J63 |
| 15 | J17 J18 J19 J20 J21 J22 J23 J24 J25 J26 | | 53 | J55 J56 J57 J58 J59 J60 J61 J62 J63 J64 |
| 16 | J18 J19 J20 J21 J22 J23 J24 J25 J26 J27 | | 54 | J56 J57 J58 J59 J60 J61 J62 J63 J64 J65 |
| 17 | J19 J20 J21 J22 J23 J24 J25 J26 J27 J28 | | 55 | J57 J58 J59 J60 J61 J62 J63 J64 J65 J66 |
| 18 | J20 J21 J22 J23 J24 J25 J26 J27 J28 J29 | | 56 | J58 J59 J60 J61 J62 J63 J64 J65 J66 J67 |
| 19 | J21 J22 J23 J24 J25 J26 J27 J28 J29 J30 | | 57 | J59 J60 J61 J62 J63 J64 J65 J66 J67 J68 |
| 20 | J22 J23 J24 J25 J26 J27 J28 J29 J30 J31 | | 58 | J60 J61 J62 J63 J64 J65 J66 J67 J68 J69 |
| 21 | J23 J24 J25 J26 J27 J28 J29 J30 J31 J32 | | 59 | J61 J62 J63 J64 J65 J66 J67 J68 J69 J70 |
| 22 | J24 J25 J26 J27 J28 J29 J30 J31 J32 J33 | | 60 | J62 J63 J64 J65 J66 J67 J68 J69 J70 J71 |
| 23 | J25 J26 J27 J28 J29 J30 J31 J32 J33 J34 | | 61 | J63 J64 J65 J66 J67 J68 J69 J70 J71 J72 |
| 24 | J26 J27 J28 J29 J30 J31 J32 J33 J34 J35 | | 62 | J64 J65 J66 J67 J68 J69 J70 J71 J72 J73 |
| 25 | J27 J28 J29 J30 J31 J32 J33 J34 J35 J36 | | 63 | J65 J66 J67 J68 J69 J70 J71 J72 J73 J74 |
| 26 | J28 J29 J30 J31 J32 J33 J34 J35 J36 J37 | | 64 | J66 J67 J68 J69 J70 J71 J72 J73 J74 J75 |
| 27 | J29 J30 J31 J32 J33 J34 J35 J36 J37 J38 | | 65 | J67 J68 J69 J70 J71 J72 J73 J74 J75 J 1 |
| 28 | J30 J31 J32 J33 J34 J35 J36 J37 J38 J39 | | 66 | J68 J69 J70 J71 J72 J73 J74 J75 J 1 J 2 |
| 29 | J31 J32 J33 J34 J35 J36 J37 J38 J39 J40 | | 67 | J69 J70 J71 J72 J73 J74 J75 J 1 J 2 J 3 |
| 30 | J32 J33 J34 J35 J36 J37 J38 J39 J40 J41 | | 68 | J70 J71 J72 J73 J74 J75 J 1 J 2 J 3 J 4 |
| 31 | J33 J34 J35 J36 J37 J38 J39 J40 J41 J42 | | 69 | J71 J72 J73 J74 J75 JJ1 J 2 J 3 J 4 J 5 |
| 32 | J34 J35 J36 J37 J38 J39 J40 J41 J42 J43 | | 70 | J72 J73 J74 J75 J 1 J 2 J 3 J 4 J 5 J 6 |
| 33 | J35 J36 J37 J38 J39 J40 J41 J42 J43 J44 | | 71 | J73 J74 J75 J 1 J 2 J 3 J 4 J 5 J 6 J 7 |
| 34 | J36 J37 J38 J39 J40 J41 J42 J43 J44 J45 | | 72 | J74 J75 J 1 J 2 J 3 J 4 J 5 J 6 J 7 J 8 |
| 35 | J37 J38 J39 J40 J41 J42 J43 J44 J45 J46 | | 73 | J75 J 1 J 2 J 3 J 4 J 5 J 6 J 7 J 8 J 9 |
| 36 | J38 J39 J40 J41 J42 J43 J44 J45 J46 J47 | | 74 | J 1 J 2 J 3 J 4 J 5 J 6 J 7 J 8 J 9 J10 |
| 37 | J39 J40 J41 J42 J43 J44 J45 J46 J47 J48 | | 75 | J 2 J 3 J 4 J 5 J 6 J 7 J 8 J 9 J10 J11 |
| 38 | J40 J41 J42 J43 J44 J45 J46 J47 J48 J49 | | | |

Figure IV-3. Jump address (Jump, Jump Conditional)

| | | | | | | | | | | | | | | | | | | | |
|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1 | T40 | T41 | T42 | T43 | T44 | T45 | T46 | T47 | T48 | 39 | T 3 | T 4 | T 5 | T 6 | T 7 | T 8 | T 9 | T10 | T11 |
| 2 | T41 | T42 | T43 | T44 | T45 | T46 | T47 | T48 | T49 | 40 | T 4 | T 5 | T 6 | T 7 | T 8 | T 9 | T10 | T11 | T12 |
| 3 | T42 | T43 | T44 | T45 | T46 | T47 | T48 | T49 | T50 | 41 | T 5 | T 6 | T 7 | T 8 | T 9 | T10 | T11 | T12 | T13 |
| 4 | T43 | T44 | T45 | T46 | T47 | T48 | T49 | T50 | T51 | 42 | T 6 | T 7 | T 8 | T 9 | T10 | T11 | T12 | T13 | T14 |
| 5 | T44 | T45 | T46 | T47 | T48 | T49 | T50 | T51 | T52 | 43 | T 7 | T 8 | T 9 | T10 | T11 | T12 | T13 | T14 | T15 |
| 6 | T45 | T46 | T47 | T48 | T49 | T50 | T51 | T52 | T53 | 44 | T 8 | T 9 | T10 | T11 | T12 | T13 | T14 | T15 | T16 |
| 7 | T46 | T47 | T48 | T49 | T50 | T51 | T52 | T53 | T54 | 45 | T 9 | T10 | T11 | T12 | T13 | T14 | T15 | T16 | T17 |
| 8 | T47 | T48 | T49 | T50 | T51 | T52 | T53 | T54 | T55 | 46 | T10 | T11 | T12 | T13 | T14 | T15 | T16 | T17 | T18 |
| 9 | T48 | T49 | T50 | T51 | T52 | T53 | T54 | T55 | T56 | 47 | T11 | T12 | T13 | T14 | T15 | T16 | T17 | T18 | T19 |
| 10 | T49 | T50 | T51 | T52 | T53 | T54 | T55 | T56 | T57 | 48 | T12 | T13 | T14 | T15 | T16 | T17 | T18 | T19 | T20 |
| 11 | T50 | T51 | T52 | T53 | T54 | T55 | T56 | T57 | T58 | 49 | T13 | T14 | T15 | T16 | T17 | T18 | T19 | T20 | T21 |
| 12 | T51 | T52 | T53 | T54 | T55 | T56 | T57 | T58 | T59 | 50 | T14 | T15 | T16 | T17 | T18 | T19 | T20 | T21 | T22 |
| 13 | T52 | T53 | T54 | T55 | T56 | T57 | T58 | T59 | T60 | 51 | T15 | T16 | T17 | T18 | T19 | T20 | T21 | T22 | T23 |
| 14 | T53 | T54 | T55 | T56 | T57 | T58 | T59 | T60 | T61 | 52 | T16 | T17 | T18 | T19 | T20 | T21 | T22 | T23 | T24 |
| 15 | T54 | T55 | T56 | T57 | T58 | T59 | T60 | T61 | T62 | 53 | T17 | T18 | T19 | T20 | T21 | T22 | T23 | T24 | T25 |
| 16 | T55 | T56 | T57 | T58 | T59 | T60 | T61 | T62 | T63 | 54 | T18 | T19 | T20 | T21 | T22 | T23 | T24 | T25 | T26 |
| 17 | T56 | T57 | T58 | T59 | T60 | T61 | T62 | T63 | T64 | 55 | T19 | T20 | T21 | T22 | T23 | T24 | T25 | T26 | T27 |
| 18 | T57 | T58 | T59 | T60 | T61 | T62 | T63 | T64 | T65 | 56 | T20 | T21 | T22 | T23 | T24 | T25 | T26 | T27 | T28 |
| 19 | T58 | T59 | T60 | T61 | T62 | T63 | T64 | T65 | T66 | 57 | T21 | T22 | T23 | T24 | T25 | T26 | T27 | T28 | T29 |
| 20 | T59 | T60 | T61 | T62 | T63 | T64 | T65 | T66 | T67 | 58 | T22 | T23 | T24 | T25 | T26 | T27 | T28 | T29 | T30 |
| 21 | T60 | T61 | T62 | T63 | T64 | T65 | T66 | T67 | T68 | 59 | T23 | T24 | T25 | T26 | T27 | T28 | T29 | T30 | T31 |
| 22 | T61 | T62 | T63 | T64 | T65 | T66 | T67 | T68 | T69 | 60 | T24 | T25 | T26 | T27 | T28 | T29 | T30 | T31 | T32 |
| 23 | T62 | T63 | T64 | T65 | T66 | T67 | T68 | T69 | T70 | 61 | T25 | T26 | T27 | T28 | T29 | T30 | T31 | T32 | T33 |
| 24 | T63 | T64 | T65 | T66 | T67 | T68 | T69 | T70 | T71 | 62 | T26 | T27 | T28 | T29 | T30 | T31 | T32 | T33 | T34 |
| 25 | T64 | T65 | T66 | T67 | T68 | T69 | T70 | T71 | T72 | 63 | T27 | T28 | T29 | T30 | T31 | T32 | T33 | T34 | T35 |
| 26 | T65 | T66 | T67 | T68 | T69 | T70 | T71 | T72 | T73 | 64 | T28 | T29 | T30 | T31 | T32 | T33 | T34 | T35 | T36 |
| 27 | T66 | T67 | T68 | T69 | T70 | T71 | T72 | T73 | T74 | 65 | T29 | T30 | T31 | T32 | T33 | T34 | T35 | T36 | T37 |
| 28 | T67 | T68 | T69 | T70 | T71 | T72 | T73 | T74 | T75 | 66 | T30 | T31 | T32 | T33 | T34 | T35 | T36 | T37 | T38 |
| 29 | T68 | T69 | T70 | T71 | T72 | T73 | T74 | T75 | T 1 | 67 | T31 | T32 | T33 | T34 | T35 | T36 | T37 | T38 | T39 |
| 30 | T69 | T70 | T71 | T72 | T73 | T74 | T75 | T 1 | T 2 | 68 | T32 | T33 | T34 | T35 | T36 | T37 | T38 | T39 | T40 |
| 31 | T70 | T71 | T72 | T73 | T74 | T75 | T 1 | T 2 | T 3 | 69 | T33 | T34 | T35 | T36 | T37 | T38 | T39 | T40 | T41 |
| 32 | T71 | T72 | T73 | T74 | T75 | T 1 | T 2 | T 3 | T 4 | 70 | T34 | T35 | T36 | T37 | T38 | T39 | T40 | T41 | T42 |
| 33 | T72 | T73 | T74 | T75 | T 1 | T 2 | T 3 | T 4 | T 5 | 71 | T35 | T36 | T37 | T38 | T39 | T40 | T41 | T42 | T43 |
| 34 | T73 | T74 | T75 | T 1 | T 2 | T 3 | T 4 | T 5 | T 6 | 72 | T36 | T37 | T38 | T39 | T40 | T41 | T42 | T43 | T44 |
| 35 | T74 | T75 | T 1 | T 2 | T 3 | T 4 | T 5 | T 6 | T 7 | 73 | T37 | T38 | T39 | T40 | T41 | T42 | T43 | T44 | T45 |
| 36 | T75 | T 1 | T22 | T 3 | T 4 | T 5 | T 6 | T 7 | T 8 | 74 | T38 | T39 | T40 | T41 | T42 | T43 | T44 | T45 | T46 |
| 37 | T 1 | T 2 | T 3 | T 4 | T 5 | T 6 | T 7 | T 8 | T 9 | 75 | T39 | T40 | T41 | T42 | T43 | T44 | T45 | T46 | T47 |
| 38 | T 2 | T 3 | T 4 | T 5 | T 6 | T 7 | T 8 | T 9 | T10 | | | | | | | | | | |

Figure IV-4. Transfer Addresses
(Transfer, Extract, And Conditional).

| | | | |
|----|-------------------------------------|----|-------------------------------------|
| 1 | w36 w37 w38 w39 w40 w41 w42 w43 w44 | 39 | w74 w75 w 1 w 2 w 3 w 4 w 5 w 6 w 7 |
| 2 | w37 w38 w39 w40 w41 w42 w43 w44 w45 | 40 | w75 w 1 w 2 w 3 w 4 w 5 w 6 w 7 w 8 |
| 3 | w38 w39 w40 w41 w42 w43 w44 w45 w46 | 41 | w 1 w 2 w 3 w 4 w 5 w 6 w 7 w 8 w 9 |
| 4 | w39 w40 w41 w42 w43 w44 w45 w46 w47 | 42 | w 2 w 3 w 4 w 5 w 6 w 7 w 8 w 9 w10 |
| 5 | w40 w41 w42 w43 w44 w45 w46 w47 w48 | 43 | w 3 w 4 w 5 w 6 w 7 w 8 w 9 w10 w11 |
| 6 | w41 w42 w43 w44 w45 w46 w47 w48 w49 | 44 | w 4 w 5 w 6 w 7 w 8 w 9 w10 w11 w12 |
| 7 | w42 w43 w44 w45 w46 w47 w48 w49 w50 | 45 | w 5 w 6 w 7 w 8 w 9 w10 w11 w12 w13 |
| 8 | w43 w44 w45 w46 w47 w48 w49 w50 w51 | 46 | w 6 w 7 w 8 w 9 w10 w11 w12 w13 w14 |
| 9 | w44 w45 w46 w47 w48 w49 w50 w51 w52 | 47 | w 7 w 8 w 9 w10 w11 w12 w13 w14 w15 |
| 10 | w45 w46 w47 w48 w49 w50 w51 w52 w53 | 48 | w 8 w 9 w10 w11 w12 w13 w14 w15 w16 |
| 11 | w46 w47 w48 w49 w50 w51 w52 w53 w54 | 49 | w 9 w10 w11 w12 w13 w14 w15 w16 w17 |
| 12 | w47 w48 w49 w50 w51 w52 w53 w54 w55 | 50 | w10 w11 w12 w13 w14 w15 w16 w17 w18 |
| 13 | w48 w49 w50 w51 w52 w53 w54 w55 w56 | 51 | w11 w12 w13 w14 w15 w16 w17 w18 w19 |
| 14 | w49 w50 w51 w52 w53 w54 w55 w56 w57 | 52 | w12 w13 w14 w15 w16 w17 w18 w19 w20 |
| 15 | w50 w51 w52 w53 w54 w55 w56 w57 w58 | 53 | w13 w14 w15 w16 w17 w18 w19 w20 w21 |
| 16 | w51 w52 w53 w54 w55 w56 w57 w58 w59 | 54 | w14 w15 w16 w17 w18 w19 w20 w21 w22 |
| 17 | w52 w53 w54 w55 w56 w57 w58 w59 w60 | 55 | w15 w16 w17 w18 w19 w20 w21 w22 w23 |
| 18 | w53 w54 w55 w56 w57 w58 w59 w60 w61 | 56 | w16 w17 w18 w19 w20 w21 w22 w23 w24 |
| 19 | w54 w55 w56 w57 w58 w59 w60 w61 w62 | 57 | w17 w18 w19 w20 w21 w22 w23 w24 w25 |
| 20 | w55 w56 w57 w58 w59 w60 w61 w62 w63 | 58 | w18 w19 w20 w21 w22 w23 w24 w25 w26 |
| 21 | w56 w57 w58 w59 w60 w61 w62 w63 w64 | 59 | w19 w20 w21 w22 w23 w24 w25 w26 w27 |
| 22 | w57 w58 w59 w60 w61 w62 w63 w64 w65 | 60 | w20 w21 w22 w23 w24 w25 w26 w27 w28 |
| 23 | w58 w59 w60 w61 w62 w63 w64 w65 w66 | 61 | w21 w22 w23 w24 w25 w26 w27 w28 w29 |
| 24 | w59 w60 w61 w62 w63 w64 w65 w66 w67 | 62 | w22 w23 w24 w25 w26 w27 w28 w29 w30 |
| 25 | w60 w61 w62 w63 w64 w65 w66 w67 w68 | 63 | w23 w24 w25 w26 w27 w28 w29 w30 w31 |
| 26 | w61 w62 w63 w64 w65 w66 w67 w68 w69 | 64 | w24 w25 w26 w27 w28 w29 w30 w31 w32 |
| 27 | w62 w63 w64 w65 w66 w67 w68 w69 w70 | 65 | w25 w26 w27 w28 w29 w30 w31 w32 w33 |
| 28 | w63 w64 w65 w66 w67 w68 w69 w70 w71 | 66 | w26 w27 w28 w29 w30 w31 w32 w33 w34 |
| 29 | w64 w65 w66 w67 w68 w69 w70 w71 w72 | 67 | w27 w28 w29 w30 w31 w32 w33 w34 w35 |
| 30 | w65 w66 w67 w68 w69 w70 w71 w72 w73 | 68 | w28 w29 w30 w31 w32 w33 w34 w35 w36 |
| 31 | w66 w67 w68 w69 w70 w71 w72 w73 w74 | 69 | w29 w30 w31 w32 w33 w34 w35 w36 w37 |
| 32 | w67 w68 w69 w70 w71 w72 w73 w74 w75 | 70 | w30 w31 w32 w33 w34 w35 w36 w37 w38 |
| 33 | w68 w69 w70 w71 w72 w73 w74 w75 w 1 | 71 | w31 w32 w33 w34 w35 w36 w37 w38 w39 |
| 34 | w69 w70 w71 w72 w73 w74 w75 w 1 w 2 | 72 | w32 w33 w34 w35 w36 w37 w38 w39 w40 |
| 35 | w70 w71 w72 w73 w74 w75 w 1 w 2 w 3 | 73 | w33 w34 w35 w36 w37 w38 w39 w40 w41 |
| 36 | w71 w72 w73 w74 w75 w 1 w 2 w 3 w 4 | 74 | w34 w35 w36 w37 w38 w39 w40 w41 w42 |
| 37 | w72 w73 w74 w75 w 1 w 2 w 3 w 4 w 5 | 75 | w35 w36 w37 w38 w39 w40 w41 w42 w43 |
| 38 | w73 w74 w75 w76 w 2 w 3 w 4 w 5 w 6 | | |

Figure IV-5. Write Addresses

In each instruction word, the programmer can only schedule Write or Transfer type commands for a limited number of registers. The Transfer, Extract, Add and Add Conditional commands all involve the playing-back of a register from General Storage. Figure IV-4 shows which General Storage registers can be played back with each instruction word. Figure IV-5 shows the registers accessible for recording in General Storage for each instruction word.

The three tables have the addresses of the 75 instruction words or registers in a track listed vertically at the left, while the ten specific command-tetrad addresses are listed horizontally in each register starting with the high-order tetrad on the left. For those commands preceded by a Delay or a Slide, the addresses in the tables are not valid.

IV-1-3. Delay.

The Delay command entails a "do-nothing" delay followed by a successful Jump Conditional if $K = 1$ or simply by a resumption of normal processing if $K = 0$. For writing brevity, the programmer calls the Jump Conditional command a Branch. A Branch command allows a programmer to move to another storage address, the decision about whether or not to move depending on the results of some previous operation.

The programmer uses the Delay in several ways. For example, he recognizes a Delay Jump (DJ) and a Delay Branch (DB) command, each having the same code (1011) and operating in exactly the same way, except that, for a Delay Branch to be successful, K must be 1. An unsuccessful Delay Branch (i.e., $K = 0$), the programmer calls a Delay Skip. The difference between the Delay Branch and the Delay Jump in the computer is determined by the state of the high-order spare bit which immediately follows the code and precedes the seven-bit delay number (see the Delay command description). From the programmer's viewpoint, there are conceptual differences between the use of the Delay Jump and the Delay Branch, but the rules for determining delay addresses can be laid down without going into these differences.

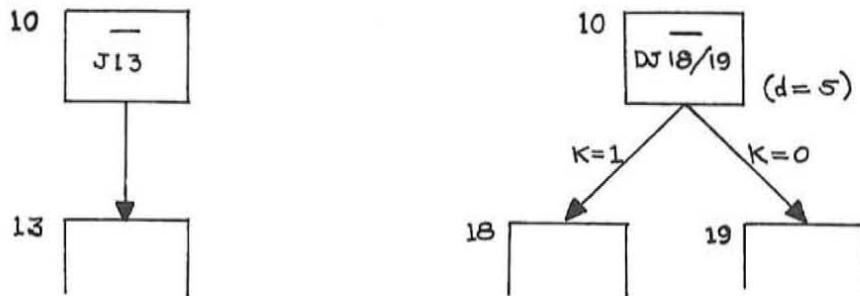
Normally, Delay commands are used in the first seven tetrads of a program word. They may be followed by other information in the same word. The minimum delay number is 1, the maximum 128. Delay numbers are coded in the program by their 128-complements:

| <u>Delay No.</u> | <u>Code</u> |
|------------------|-------------|
| 1..... | 1111111 |
| 2..... | 1111110 |
| 3..... | 1111101 |
| 4..... | 1111100 |
| | ↓ |
| 126..... | 0000010 |
| 127..... | 0000001 |
| 128..... | 0000000 |

For all Delay commands, the delay number is given by:

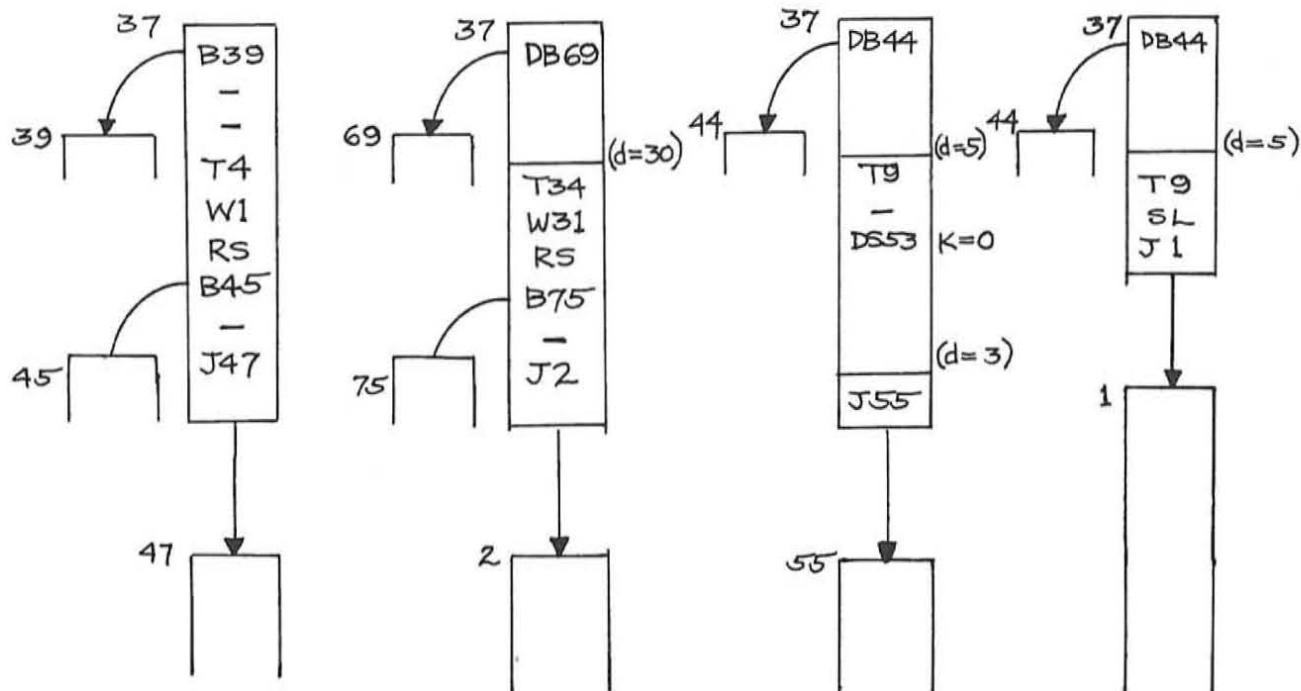
$$d = A - J - (1 - K)$$

where A is the address to which the Jump is made and J is the Jump address of the tetrad in which the Delay code is located. It will be noted that the delay is shorter when K = 1 than when K = 0. For example:



A successful Delay Branch ($K = 1$) operates in exactly the same manner as a Delay Jump with K equal to 1.

An unsuccessful Delay Branch ($K = 0$), which is also called a Delay Skip, is a pure delay, after which the commands following the delay number in the program are executed. The delay number is added to the addresses of the commands following a Delay Skip, remembering that 75 is the maximum number which can be added in a track. However, the influence of a Delay Skip on the addresses never extends beyond a Slide. Furthermore, if there are two Delay Skips in a word, both delay numbers are added to the addresses of the commands following the second delay. For example:



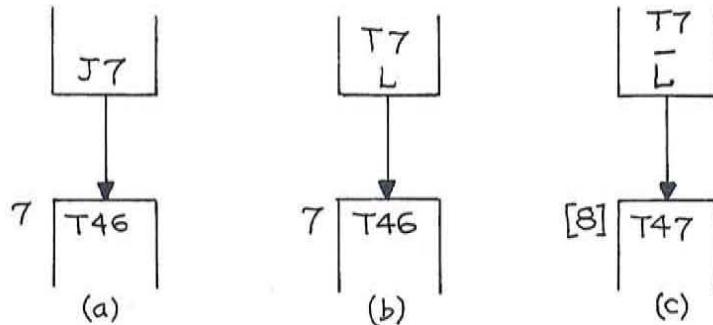
IV-1-4. Leap.

In order to facilitate the formulation of special addressing rules, the concept of key number was devised. Each program word has a key number associated with it, which usually -- that is, except for some uses of the Leap command --

is also its address in General Storage.

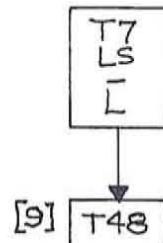
Note that the address of a specific command depends on both the key number of the program word and the location of the command within the word.

If a key number differs from the address at which the program word is actually stored, it is written in brackets on the flow-chart.



If a program word is entered into by a Leap, its key number is equal to the Transfer address of the tetrad which immediately precedes the Leap code -- that is, the register of the Transfer command, if such a Transfer command were present in that tetrad. Thus, in example (c) above, the address of the word is 7 (in a program flow-chart, this would be preceded by two other digits -- a track number on the extreme left and a zero) and its key number is 8.

If a program word is the result of an arithmetic operation before a Leap, it has a key number but no address. For example:



In this case, the information in the second program word is not taken directly from General Storage but undergoes a modification. Thus the word has no General Storage address but its key number is 9, which is the register address for a Transfer command for the tetrad just before the Leap.

IV-1-5. Slide.

The first tetrad after a Slide has the following addresses associated with it, regardless of whatever went before the Slide:



The duration of a Slide is a measure of its address. For example, if the addresses above coincide with the addresses associated with the same tetrad in the absence of a Slide, then the Slide is of minimum duration. Thus:

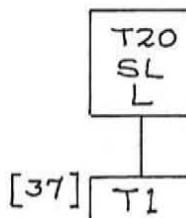
73 **SL** this Slide is of minimum duration (2 word-times),
J1

73 — SL J1 while this one is of maximum duration (148 word-times).

The tetrads which follow the first tetrad after the Slide are then addressed consecutively, as follows:

56 T20
SL
B1
T39
W36
J4

If a Slide is directly followed by a Leap, the key number after the Leap is 37:



IV-1-6. Divide By Five.

It has been pointed out that the Partial Divide By Five command produces only a scrambled quotient. It must be followed by a subroutine or sequence of commands in order to complete the division process.

The full subroutine for the division of a negative number by five is as follows:

PDF
AC (4z+1)
AC (4z+1)
AC (4z+1)
AC (4z+1)

Here, the constant z has the value:

and $4z+1 = 110011001100110011001100110011001101$.

The subroutine for the division by five of positive numbers is as follows:

```
CM
PDF
AC (4z+1)
AC (4z+1)
AC (4z+1)
AC (4z+1)
CM
```

where the expression $(4z+1)$ has the same value as in the case of negative numbers.

There is an automatic roundup in the process. All answers with remainders are rounded up to the next whole number.

IV-1-7. Example.

A flow-chart for a "three-factor" Multiply routine is shown in figure IV-6; excerpts from the coding chart are shown in figure IV-7. The routine computes $A \times B \times 1/C$, with the factors A, B and $1/C$ located where shown on the flow-chart. Briefly, the sequence is to right-shift Multiplier A, which tests the low-order bit by shifting a_1 into K. The shifted Multiplier is written back in storage. Each time that a_1 is 1, the Multiplicand is added into the Intermediate Product. This is equivalent to the procedure followed in multiplying two numbers with pencil and paper. The second half of the routine now treats the Intermediate Product as the Multiplier and multiplies it by $1/C$. Therefore, the Intermediate Product is also right-shifted before it is written back into storage. Each time the a_1 of the Intermediate Product is 1, the factor $1/C$ is added to the Final Product.

A count number is loaded into register 237 at the time the Multiply factors are entered. In each cycle of the routine, the count number is incremented by +1. When register 237 overflows, the Branch to register 203 is successful, which terminates the routine. Note that the routine is short enough so that a Leap can be used to restart the calculation after one drum revolution.

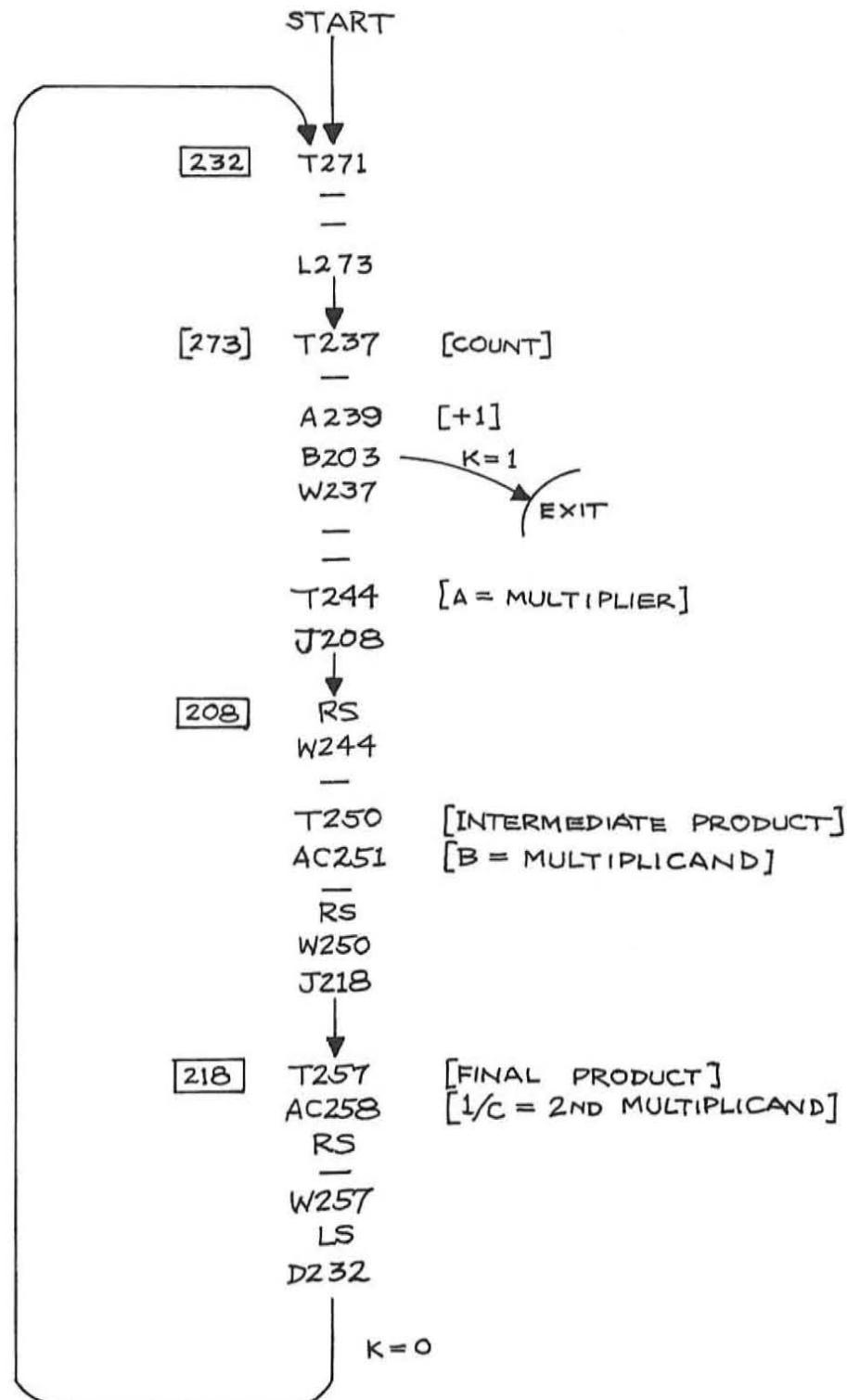


Figure IV-6. Program Flow Chart

TRACK 2

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|-------|----|-------|-------|-------|-------|-------|-------|-------|-----------|
| MP | 1 | RS | CM | CM | DB222 | 7 | — | DJ231 | 15 |
| | 2 | DS | 0100 | 0000 | CT | T | CT | SL | — |
| | 3 | LS | DS210 | 7 | 12 | T250 | D241 | 14 | 6 |
| | 4 | DS222 | 7 | — | W25B | T263 | A264 | B228 | J229 |
| | 5 | — | T245 | CT | CT | DS428 | G | 16 | L467 |
| | 6 | — | — | — | W244 | E | CL | — | J215 |
| | 7 | CM | — | W244 | E | CL | LS | J215 | |
| | 8 | RS | W244 | — | T250 | AC251 | — | RS | W250 |
| | 9 | A248 | P | B213 | — | T252 | DS231 | 7 | J21B |
| | 10 | T249 | P | — | — | — | — | — | B219 |
| <hr/> | | | | | | | | | |
| MP | 16 | W251 | CL | — | J221 | | | | |
| | 17 | — | — | T25B | W25B | LS | LS | W25B | J226 |
| | 18 | T257 | AC252 | RS | — | W257 | LS | P232 | 15 |
| | 19 | T25B | LS | B223 | LS | W25B | CT | CT | RS |
| | 20 | DS245 | 6 | 9 | T210 | L210 | | | — |
| <hr/> | | | | | | | | | |
| MP | 28 | W263 | LS | CT | RS | RS | RS | RS | J338 |
| | 29 | CT | CT | T470 | T470 | | | | |
| | 30 | — | T270 | LS | A272 | J236 | | | |
| | 31 | T270 | — | A272 | J236 | | | | |
| | 32 | T271 | — | — | L273 | | | | |
| | 33 | CT | — | — | — | — | — | J341 | |
| | 34 | — | T274 | — | W272 | — | CT | CT | T405 L405 |
| | 35 | CT | CT | — | — | T403 | L403 | | |
| | 36 | E275 | W272 | A202 | L202 | | | | |
| | 37 | COUNT | | | | | | | |
| MP | 38 | — | W274 | CT | CT | — | — | T408 | L408 |
| | 39 | — | — | — | — | — | — | — | 0001 |
| <hr/> | | | | | | | | | |
| MP | 40 | | | | | | | | |
| | 41 | CM | CM | DB248 | 7 | 13 | — | T214 | L214 |
| | 42 | RS | LS | — | — | — | — | T213 | L213 |
| | 43 | CT | CT | T135 | W132 | T137 | — | W135 | J103 |
| | 44 | — | — | — | — | — | — | | |
| | 45 | CT | T132 | CT | CM | DB208 | 15 | 8 | |
| | 46 | — | — | — | — | — | — | 0010 | 0000 |
| | 47 | T211 | — | — | — | — | — | L216 | |
| | 48 | CL | CT | CT | CT | DI75 | 14 | 11 | 0011 |
| | 49 | — | — | — | — | SP | — | | |
| MP | 50 | — | — | — | — | LP | — | | |
| | 51 | — | — | — | — | MC | — | | |
| MP | 52 | CT | CT | CT | T137 | J101 | — | 0011 | 0000 |
| | 53 | 2 | 5 | 4 | — | 11 | 14 | 4 | — |
| | 54 | 17 | 5 | 7 | 14 | 8 | 3 | 8 | — |
| | 55 | — | — | — | — | — | — | | |
| | 56 | — | — | — | — | — | — | | |
| | 57 | — | — | — | — | FP | — | | |
| | 58 | — | — | — | — | 1/C | — | | |
| | 59 | — | — | — | — | — | — | — | 0001 |
| | 60 | — | — | — | — | — | — | | |
| | 61 | — | — | — | — | — | — | | |
| <hr/> | | | | | | | | | |
| MP | 68 | CT | CT | CT | T135 | DJ110 | 15 | 6 | |
| | 69 | P | DB233 | 5 | 12 | DJ269 | 14 | — | |
| | 70 | — | — | 0001 | — | — | — | — | — |
| | 71 | T237 | — | A239 | B203 | W237 | — | — | T244 |
| | 72 | — | — | — | — | — | — | — | J208 |
| | 73 | CM | B201 | CT | CT | CT | — | — | — |
| | 74 | — | JM/JR | — | — | — | — | — | — |
| | 75 | — | 0011 | 1111 | — | — | — | — | — |
| | | | | | | | | | AJ109 |

Figure IV-7. Program Coding Chart

IV-2. STARTING AND OPERATING

When the computer is started or recycled, its on/off sequencing circuitry causes it to look for a zero flag bit on track 1 and then perform an automatic Jump into the start-word, which appears in register 138. From this point on, the operation of the machine is under program control.

As described in "On/Off Control", the control panel of the Monrobot XII has three switches -- PWR (Power), RST (Restart) and PGM (Program). PWR simply connects AC power to the computer. RST allows the programmer to return to the starting point in the program without shutting off the computer. The PGM switch is used to select either track 2A or 2B. In the course of operating the PGM switch, the on/off sequencing circuit is reset, and the RST switch must then be depressed to restart the program.

When turning on the computer, the operator simply depresses the PWR switch. As soon as the transitional turn-on requirements have been satisfied, the RST switch lights, indicating that the computer has found the starting point and is ready to operate. At the same time, any ready lights that may be on the input devices are turned on. The operator can then punch the input keyboards as necessary. When the input devices are busy, they are again locked by the lack of an input-clear signal. The device being used is released when a successful input is executed. Thus, the operator need use only the PWR switch and the controls of the input devices, unless it becomes necessary to restart or change programs.

SECTION V

THE LOADER

| | Page |
|---|------|
| V-1. GENERAL DESCRIPTION | V-1 |
| V-2. TAPE CODING | V-4 |
| V-3. THE LOADING PROCESS | V-6 |
| V-4. DETAILED DESCRIPTION OF LOGICAL SEQUENCING | V-14 |
| V-4-1. Reader Control Logic | V-14 |
| V-4-2. Program Data Assembly Logic | V-17 |
| V-4-3. Transfer Without Sync Logic | V-19 |
| V-4-4. Transfer With Sync Logic (8 Word-Times) | V-23 |
| V-4-5. Transfer With Sync Logic (10 Word-Times) | V-25 |
| V-4-6. Stop Logic | V-27 |
| V-4-7. Ignore Logic | V-29 |
| V-4-8. Turn-On Logic | V-31 |
| V-5. LOGIC DIAGRAMS | V-32 |
| V-6. CIRCUITS | V-40 |
| V-6-1. Device Clear and Device Common Circuits | V-41 |
| V-6-2. Device Busy Inverter Circuit | V-42 |
| V-6-3. Level Shift Circuit | V-43 |
| V-6-4. Clamping Circuits | V-44 |
| V-6-5. Noise Suppression Circuits | V-45 |
| V-6-6. Control Panel Switches | V-46 |
| V-7. SIGNAL GLOSSARY | V-48 |

SECTION V
THE LOADER

V-1 GENERAL DESCRIPTION

A micro program must be loaded into the storage of Monrobot XII before the computer can be operated. It is the loader which performs this function; the loader transfers the coded program data from punched paper tape to the General Storage tracks.

The loader system consists of logic circuits, a tape reader and panel-mounted operator controls. It has neither a permanent memory of its own nor a power supply.

A cable with a two-sided printed circuit connector (PLL) is provided which plugs into the computer. This cable:

- a) brings power to the loader circuits from the computer power supplies, and
- b) connects the loader logic to the computer drum memory and its associated playback and record circuits.

Pin assignments of connector PLL are as follows:

| <u>Side</u> A | <u>Pin</u> 1 | <u>Signal</u> M1 | <u>Side</u> B | <u>Pin</u> 1 | <u>Signal</u> FPB1 |
|------------------|-----------------|---------------------|------------------|-----------------|-----------------------|
| | 2 | M2 | | 2 | A28 |
| | 3 | <u>L1</u> | | 3 | F2 |
| | 4 | <u>L1</u> | | 4 | F3 |
| | 5 | L2 | | 5 | GND |
| | 6 | <u>L2</u> | | 6 | L7 |
| | 7 | <u>L8</u> | | 7 | L5 |
| | 8 | L8 | | 8 | <u>A4</u> |
| | 9 | <u>P1</u> | | 9 | A8 |
| | 10 | M3 | | 10 | C40 |

| <u>Side</u> | <u>Pin</u> | <u>Signal</u> | <u>Side</u> | <u>Pin</u> | <u>Signal</u> |
|-------------|------------|------------------|-------------|------------|------------------|
| A | 11 | M4 | B | 11 | $\overline{W4}$ |
| | 12 | T36 | | 12 | W4 |
| | 13 | $\overline{T36}$ | | 13 | T29 |
| | 14 | GND | | 14 | $\overline{A29}$ |
| | 15 | + 18 | | 15 | - 24 |
| | 16 | - 8 | | 16 | - 6 |
| | 17 | B.GND | | 17 | FR.G. |
| | 18 | - 24 B | | 18 | |

The above cable not only makes the prescribed connections between the computer and the loader, it also disconnects the computer's memory circuits from its logic unit. Thus, the computer logic is inactivated whenever the loader is plugged in.

The operation of the loader is controlled manually by the switches mounted on the front control panel. (See Fig. 1) The controls are listed:

- (a) The Track Selection switch, the purpose of which is self-evident, is a seven-position rotary switch. The positions are labelled: RC1/PB1, RC2/PB1, RC2/PB2, RC3/PB1, RC3/PB3, RC4/PB1, RC4/PB4. Some positions allow the operator to record in one track while playing back from track 1. The significance of this will become clear later.
- (b) The Clear/Rec T36 switch, when operated in the momentary up position, causes zeros to be recorded in Fast Access, and in the General Storage track indicated by the setting of the Track Selection switch. When operated in the down position, it causes a 1 to be recorded in every t_{36} position of the selected General Storage track.
- (c) The Tape Start/Stop switch, when operated in the momentary

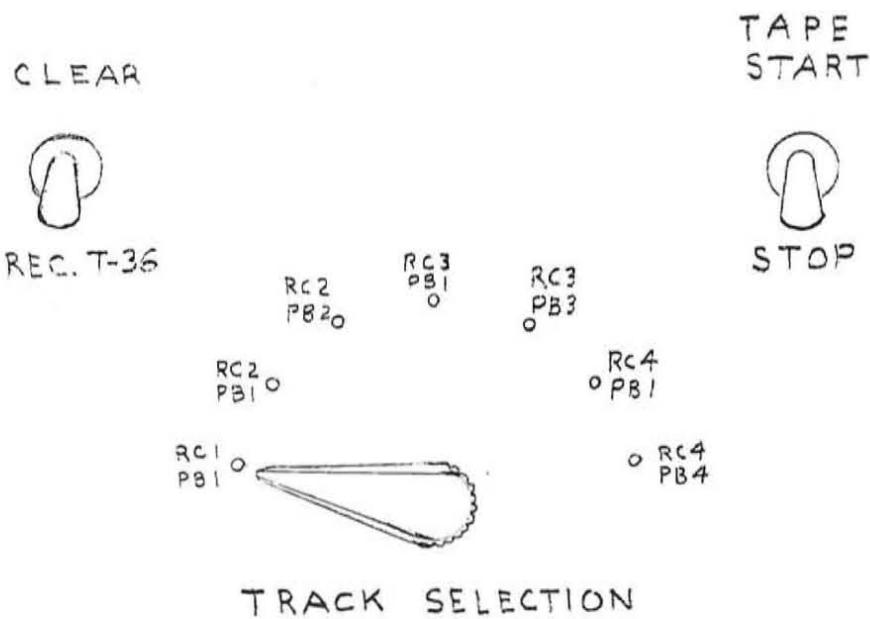


FIG. 1
FRONT CONTROL PANEL

up position, sets the tape driver in motion. When operated in the down position, the switch causes the reader to stop when in motion, or prevents it from starting.

The tape reader used by the loader is a standard Monroe input device. Details on these devices are available in the Input/Output section of the manual.

V-2 Tape Coding

The program data is input to the system on paper tape using a five-bit code. Each program word on the tape consists of eleven such five-bit characters---a flag-bit character, nine program data characters and a loader control character. The layout of a tape word is shown in Figure 2. Tapes prepared on a Teletype will have additional bits punched so that the Teletype can read & edit the tapes. Only the five information bits (b_1-b_4 , b_7) are read by the loader.

The flag-bit character, which is the first in a program word in the direction of tape travel, may be either a zero or a one. The nine succeeding data characters are each assigned any value from 0 - 15.

The eleventh or last character in a tape word is a transfer character. Note that the transfer character contains a 1 in the b_7 bit position. There are two transfer characters which may be used.

- (1) Transfer Without Synchronization
- (2) Transfer With Synchronization

In addition to the above mentioned characters there are two "house-keeping" characters which appear on the tape and contain a 1 in the b_7 bit position. They are the "Stop" character and "Ignore" character. The meaning of the "Stop" character is self-explanatory; it simply causes the loader to stop reading tape. The "Ignore" character is, in effect, a non-

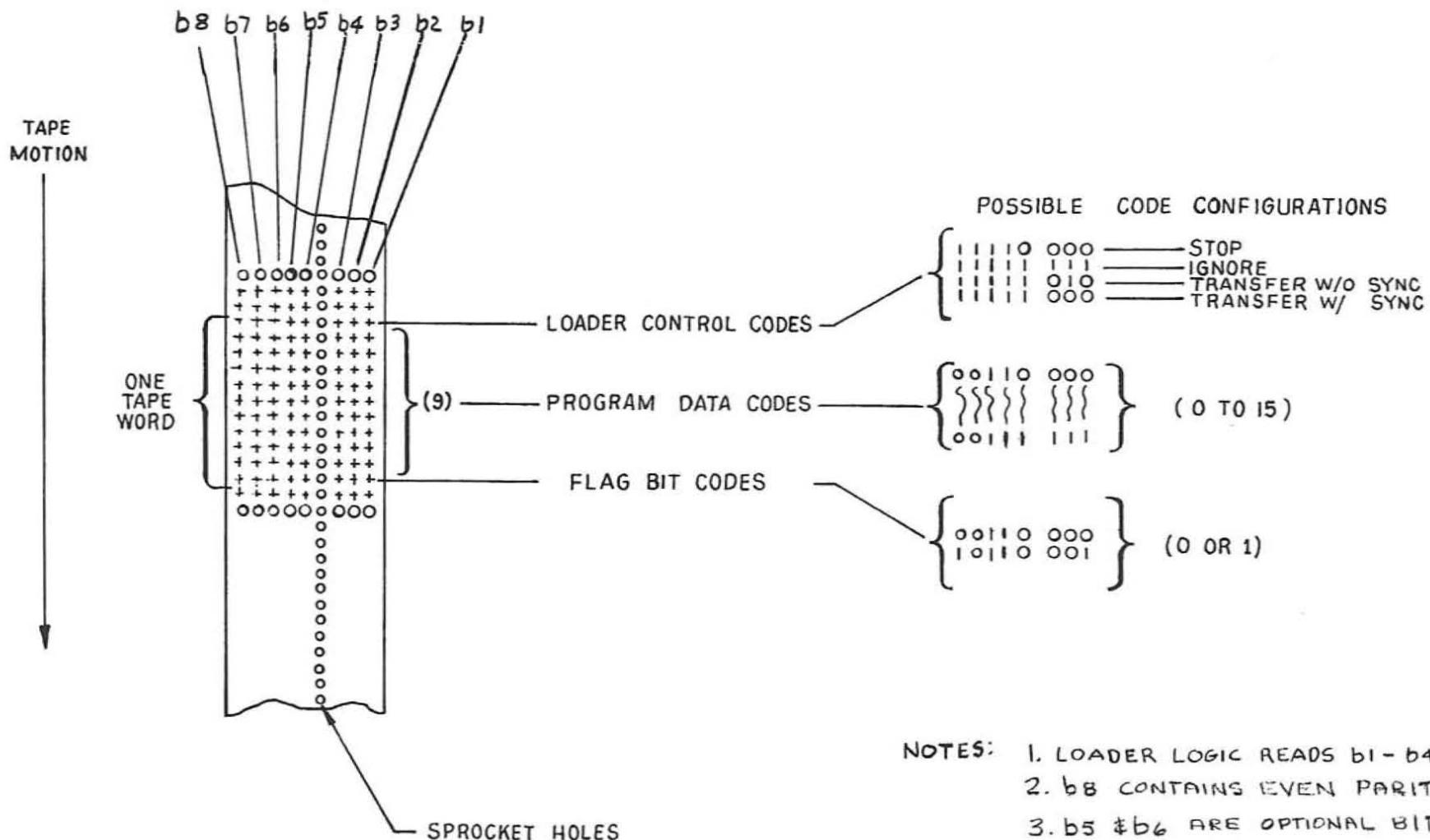


FIG. 2
TAPE CODE FORMAT

entry. It permits the elimination of a wrong character punched on the tape; the programmer need only punch all the holes to convert the wrong character entry to an "Ignore" or non-entry.

It should be noted from the foregoing discussion and Fig. 2 that all loader control characters are characterized by a 1 in the b_7 hole position.

V-3 The Loading Process

Prior to starting the loading process the track to be loaded is cleared to zeros by first positioning the Track Selector switch to the desired track and then operating the Clear switch. If the entire General Storage is to be loaded, this process must be repeated for each track. After this clearing operation is complete the loading process may be started by placing the tape in the reader and operating the Tape switch to the Start position. Once tape motion is started the loader will not stop until it reads a "Stop" character. Note that the tape motion may also be stopped by manual operation of the Tape switch to the Stop position, or if the Tight Tape/No Tape switch on the reader is automatically actuated, but these are not normal occurrences in a loading procedure.

The actual loading process may be said to occur in two phases:

(1) Data Word Assembly. Reference to Fig. 3 will help in the following discussion. Note that the fast access loop has two information paths, a regeneration path with no delay and a precess path consisting of four flip-flops. Note also that a fifth flip-flop F9 is shown which is not part of the information flow path. This flip-flop along with the four in the precess path comprise the input register, i.e., the register into which each tape character is read, one at a time, in parallel form. The lone flip-flop F9 stores the b_7 bit of the character. The other flip-flops F4-F7 store the

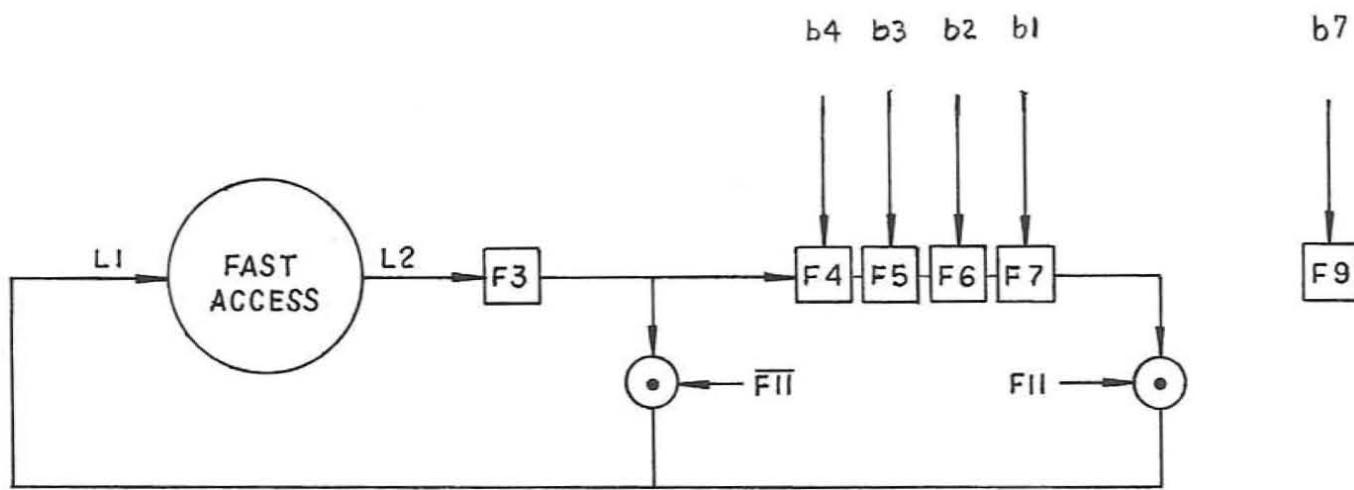


FIG. 3

DATA WORD ASSEMBLY FLOW DIAGRAM

b_4, b_3, b_2 and b_1 data in the order shown.

Fast access is normally regenerating so that the four precess-path flip-flops F4-F7 are normally in a static state. When a new character is deposited in the input register the logic examines the state of the b_7 bit flip-flop F9. If it contains a zero the information in flip-flops F4-F7 is program data: Flip-flops F4-F7 now form a shift register through which the fast access loop information is precessed for one word-time. As a result of this precession, the character in the register is read into the low order end of the fast-access word, and the information is delayed toward the high order end by four bit times. Thus, each new data character is read into the low order end of the word and the previously entered characters are shifted up until the program word is complete.

Flip-flop F11 is used to gate the fast access flow. The regenerate condition is defined by $\overline{F11}$. Precess is defined by F11. The precess time is always one word-time, therefore F11 is never set for longer than a word-time. It should be pointed out that only one word space is used in the two word-time fast access loop, therefore, the set logic of F11 is in synchronism with a word counter flip-flop, F8. F8, in the loader, is the exact counterpart of F8 in the computer logic, i.e., it defines the two word-spaces in the fast-access channel.

Note that ten four-bit characters (nine data characters plus a flag tetrad) or forty bits, are precessed into the fast access in the manner described, whereas the fast access register itself is only 37 bits long (nine tetrads plus a single flag-

bit). However, in the flag-bit tetrad of the tape word, it is only the low-order bit which contains any information and thus is of any interest; if the other three bits of this tetrad are ignored or discarded, a total of 37 bits remain in the tape word. This, in fact, is precisely what happens. The flag tetrad, which is read in first, is precessed up by the nine data tetrad entries to the "top" of the word and its three high-order bits are spilled off the high-order end and lost. All that remains is the single low-order bit of the tetrad, which is the flag bit.

(2) Transfer to General Storage. If the character read into the input register contains a by bit, F9 will be set indicating to the loader logic that the character is an instruction or command to the loader. Normally, the tenth data character is followed by a control character, specifically, a "Transfer" command. As previously indicated, there are two "Transfer" characters:

a) The "Transfer Without Sync" character causes the assembled word in the fast-access loop to be transferred to a random location on the selected General Storage drum track. It is used only once in any given program - in the first word-entry -- and in effect establishes a reference point on the storage drum for further loading.

b) The "Transfer With Sync" character also causes transfer of the information in fast access to General Storage, but in this case, the transfer is not random. A specified sequential pattern is followed in transferring a word to its proper location (as determined by the program) on its selected

track.

When the loading process begins, there is no reference index for determining the location of the first word entry. Therefore, since all the registers are empty, the first word is entered at random. This is the reason for the use of the special "Transfer Without Sync" character with the first word-entry. The word is recorded in any random register location and its flag-bit, which is always a "1", becomes an index for the next word-entry. The latter contains a "Transfer With Sync" character, as do all the rest of the words to be recorded on that track.

The loader control logic, when it senses the "Transfer With Sync" character, looks for the first 1's-flag-bit on the selected drum track to appear under the General Storage playback head -- which, in this case, is the flag-bit for the first word-entry. Having sensed this, it then looks for the first 0-flag-bit to follow, which indicates an empty register. As soon as this has been found, the control logic counts eight word-times (the playback and record heads of General Storage are eight word-times apart) and then executes the transfer. Flip-flops F5, F6, F7 are used to perform the counting since they do not contain program data during the transfer operation and are therefore available.

Thus, starting with the first word transferred, each word contains a 1 in its flag bit character, except for the last word (the 75th). This word contains a 0-flag-bit. There are two reasons for this.

- (1) If the numerical sequence of registers on the other tracks is to be aligned "side by side" with that on track 1, (that is, if the first word on each track is to be adjacent to that on every other track, and so on for all of the 75 words) a reference index on track 1 is required. The zero flag-bit provides such an index.

(2) When the computer is started, under program control, it must find its start-word on track 1, which it does by seeking out the single 0-flag-bit on the track. This 0-flag marks the program start-line; the following word is the first word in the computer program.

When the loading of track 1 has been completed, it becomes necessary to stop the loading procedure and switch the track selector manually. Thus a "Stop" character must appear on the tape immediately following the 75th or last transfer character. When the "Stop" character has been received, the loading on track 1 is complete. The track now contains 75 words of program data -- the first 74 with a 1's flag-bit and the last with a 0-flag-bit. The operator can now turn the Track Selector switch to the next detent position, which is labelled RC2/PB1. In this position, the loader will record or transfer data into track 2 but will force playback of information from track 1. By this means the first word-entry on track 2 is indexed by the information on track 1. However, only the first word need be so indexed; all subsequent entries are indexed by the first entry on the same track, as in the word-entry procedure described for track 1.

The first word-entry, then, is transferred in synchronization with the single 0-flag-bit on track 1, using the same procedure as described above. The loader logic looks for a 1's flag-bit; when it finds it, it then proceeds to look for a 0-flag, after which the loader logic begins the count procedure to allow for the spacing between the record and playback heads. The count in this case, however, is not eight word-times, even though the head spacing has not changed. If the first words on the tracks are to be aligned, the count at this point must be ten word-times. A count of eight, as before, would transfer this first word into track 2 in alignment with the last word in track 1. The two extra counts (to allow for the alternating sequence of register numbers on the drum) transfer the first word into

track 2 in alignment with word number 1 on track 1.

Following this first track 2 transfer, another "Stop" character is provided. This stop allows the operator to turn the Track Selector switch to the next detent position, labelled RC2/PB2. The loading process can then be resumed, by operating the Tape Start switch, and the next 74 words are transferred exactly as described for track 1.

The procedure described here for track 2 is then repeated for tracks 3 and 4, synchronizing the first word of each in turn with the 0-flag-bit on track 1, then allowing the remaining 74 words on each to be transferred automatically. It will be recalled that there is a fifth storage track in the computer -- track 2B. This can be loaded at any time in the loading sequence after loading track 1 by stopping the loading procedure, pressing the PGM (Program) switch on the computer, and loading in exactly the same manner as described for track 2 above, using the track 2 detent positions on the Track Selector switch.

To summarize the transfer process briefly: the first word entry on track 1 is transferred from the fast access channel into General Storage without synchronization, and the remaining 74 are transferred with synchronization in "one after the other" fashion. All words contain a 1's flag-bit, except the last.

The first word to be transferred into tracks 2, 3 and 4 is synchronized using a 0-flag-bit of the last register on track 1. However, the loader logic aligns it with register 1 of track 1 by augmenting the regular 8-count, which allows for the head spacing, by 2 counts. The remaining 74 are then transferred with synchronization, as on track 1, in sequence.

Register 75 is loaded with a 0-flag-bit on all tracks so that this pulse will be available as a scope synchronizing signal for test purposes.

The order of register loading on each track, the states of the flag-bits and the control characters used are listed in the following table:

| <u>Track</u> | <u>Register</u> | <u>Flag Bit</u> | <u>Control Character</u> |
|--------------|-----------------|-----------------|--------------------------|
| 1 | 1 | 1 | Without Sync |
| 1 | 2-74 | 1 | With Sync |
| 1 | 75 | 0 | With Sync |
| 2, 3 & 4 | 1-74 | 1 | With Sync |
| 2, 3 & 4 | 75 | 0 | With Sync |

It will be recalled that the flag-bit, which is left in the high-order end of the word in fast access at the end of the data assembly process, must be in the lowest order bit position when the word is finally recorded in General Storage. This condition is rectified in the following manner: before any actual transfer from fast access to General Storage is performed the data word is first read into a flip-flop (F2). Since this device can "store" only one bit of information, its state after the word has been read into it reflects that of the last bit to pass through it - which is, of course, the high-order or flag-bit. When the actual transfer count occurs, the data word is read into General Storage via a path which includes F2 with its stored flag-bit. In this way, the flag-bit is the first bit to be read into General Storage (g_0) and the balance of the word is appropriately delayed one bit-time toward the high-order end. Thus, the flag-bit has, in effect, been moved to the opposite end of the data word.

V-4 Detailed Description of Logical Sequencing

Following is a description of the logical sequencing of the loader.

V-4-1 Reader Control Logic

The reader used by the loader is a pins-up device. The loader accepts the character from the reader and then causes the reader to cycle and advance the tape to the next character.

Sequence

(1) F10 is set as a result of:

- a) F11 being high indicating a program character has just been precessed into fast access, or
- b) C67 being high indicating a transfer operation has just been completed, or
- c) W12 which is generated when the Tape Start switch is operated.

(2) $\overline{A21}$ goes low with the setting of F10 and remains low throughout the entire read cycle which includes the busy period of the reader. $\overline{A21}$ low prevents the loader logic from acting upon the character in the input register.

(3) A22 is generated by $F10 \cdot \overline{F12}$ to reset the input register flip-flops, F4-F7.

(4) If E1 is high (parity error on the character about to be input), A22 sets F9. In this situation, a "Stop" code has been forced into the input register and sequencing stops.

If $\overline{E1}$ is high (parity correct), A22 resets F9. The input register has been completely cleared to receive a new character.

(5) F2 is set by F10 (note that F2 may already be high).

(6) F12 is set with the first t_{36} pulse after F10 is set.

(7) S2, the read "common" signal, is high to cause the reader to

transmit the tape character via its code wires (b_4, b_3, b_2, b_1, b_7) to the input register flip-flops F4-F7 and F9.

(8) F10 is reset the first t_{36} pulse after F12 is set.

(9) S1, the reader "clear" signal goes high, causing the reader to clear or reset the code wires to zero and begin its mechanical advance to the next position. As a result of this action, the reader feeds back a "busy" signal, S3, to the loader logic.

(10) F2 is reset the first t_{36} pulse following the reset of F10.

(11) No further logical sequencing occurs until the reader mechanical cycling is complete as evidenced by the busy signal S3 going low to reset F12. As a result $\overline{A21}$ goes high and the logic decodes the input register to start the next operation.

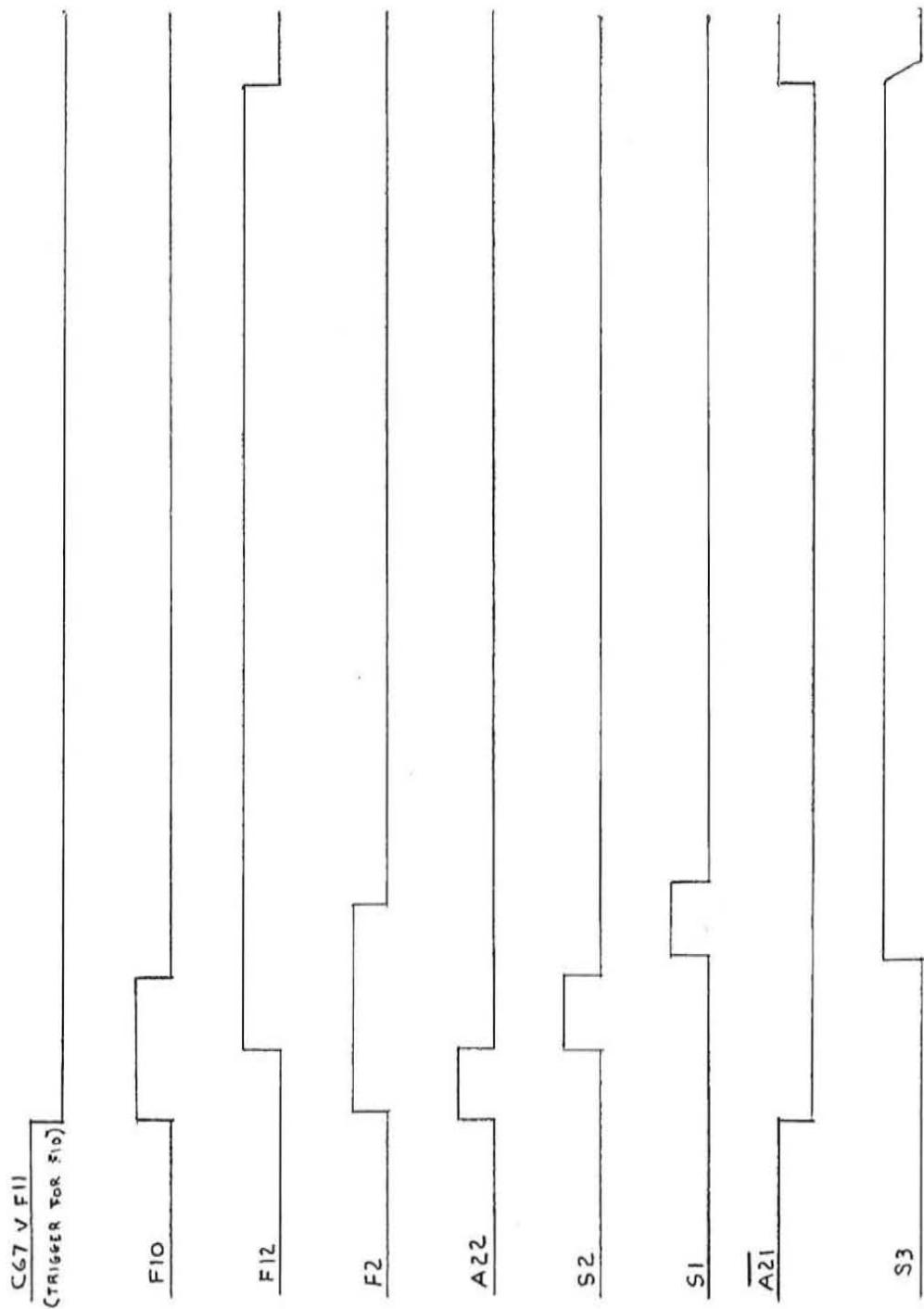


FIG. 4
READER CONTROL TIMING

V-4-2 Program Data Assembly Logic

The following description assumes that a read cycle has just been completed as described under "Reader Control Logic".

Sequence

- (1) $\overline{A21}$ has just gone high to indicate the end of the read cycle. The logic now decodes the input register, F4-F7 and F9. F9 is low, therefore, the code in F4-F7 is program data.
- (2) F11 is set with the first t_{36} that occurs coincident with $\overline{F8}$ after $\overline{A21}$ goes high. F11 stops regeneration of the fast access information and causes fast access data to flow through the input register flip-flops, F4-F7, before being recorded back in fast access. Thus, the code in F4-F7 is recorded in the low order of fast access and all previously recorded data is delayed four-bit-times toward the high order.
- (3) F2 is set one bit-time after F11 is set. F2 will function as an anti-repeat memory for F11.
- (4) F11 is reset one word-time after it is set. Thus the data character previously read has been recorded in fast access and the logic is now prepared to accept the new character which is waiting in the read station of the reader.

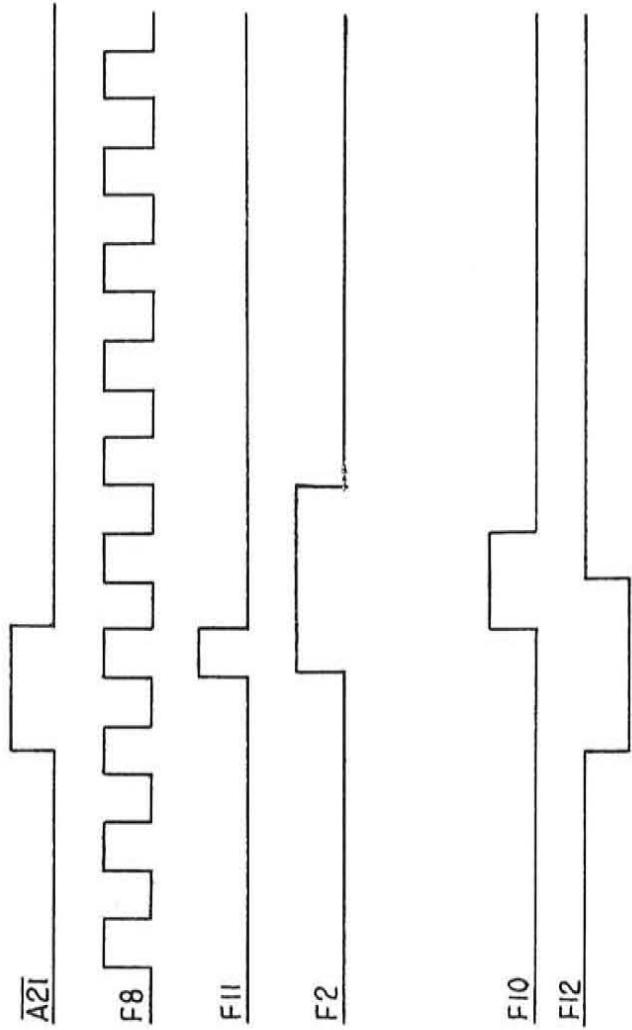


FIG. 5
PROGRAM WORD ASSEMBLY TIMING

V-4-3 Transfer Without Sync Logic

The following description assumes the read cycle, as described under "Reader Control Logic", is just terminating. It should be noted that fast access is continually regenerating, and at this time contains a completely assembled program data word. It is the purpose of this operation to transfer this data word to a random location in General Storage.

Sequence

- (1) $\overline{A21}$ goes high to indicate the read cycle is complete. The logic decodes the "Transfer Without Sync" character in the input register flip-flops F4-F7 and F9.
- (2) F5, F6 and F7 form a binary counter. (They are available for use as a counter since in this operation they do not contain a data character. Their binary weights are: $F5 = 4, F6 = 2, F7 = 1.$) The counter begins to count with the first t_{36} that occurs after $\overline{A21}$ has gone high and continues to count every subsequent t_{36} . Note that F6 is set initially as part of this transfer code, therefore the count begins at binary 2.
- (3) When the count reaches 4, the fast access data (F3) is read into the F2 flip-flop for one word-time via N21. F2 is used here to "capture" the last bit read into it from fast access, which is the flag bit (it will be remembered that the flag bit was stored in the high order (t_{36}) bit position of the completely assembled word in fast access).
- (4) During count 6, the program word is transferred from fast access (F3) to General Storage via a one bit delay path, F2. F2 contains the flag bit at the beginning of count 6

that it "captured" from fast access at the end of count 4. Therefore, the first bit recorded in General Storage is the flag bit. The remainder of the program word is, of course, delayed one bit-time and recorded in the t_1 - t_{36} bit positions of the General Storage word-space. The record path is F3 - N21 - F2 - L5, L7. L5 is the record zeroes gate and L7 is the record ones gate for General Storage. (Note that fast access continues to regenerate, i.e., the program word is not destroyed as a result of the transfer.)

- (5) F10 is set at the end of count 6 (t_{36}).

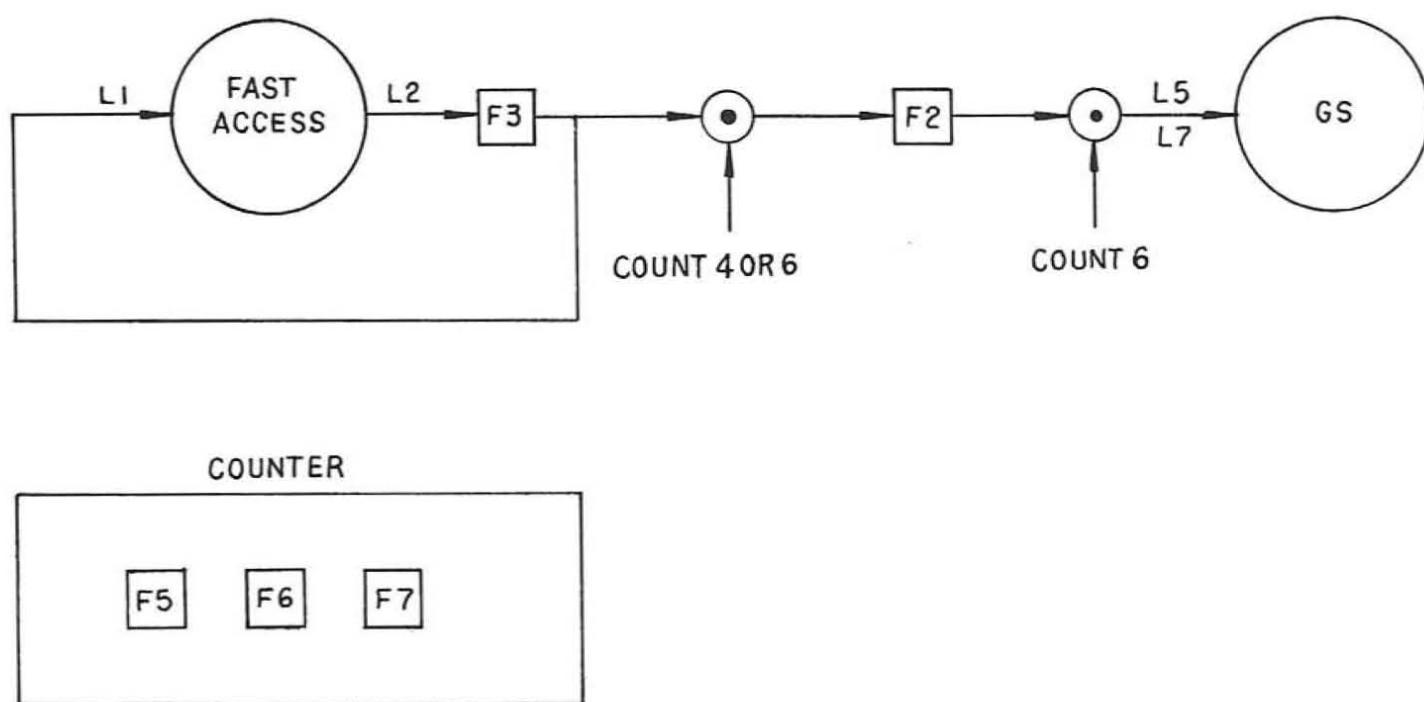


FIG. 6
TRANSFER TO GENERAL STORAGE FLOW DIAGRAM

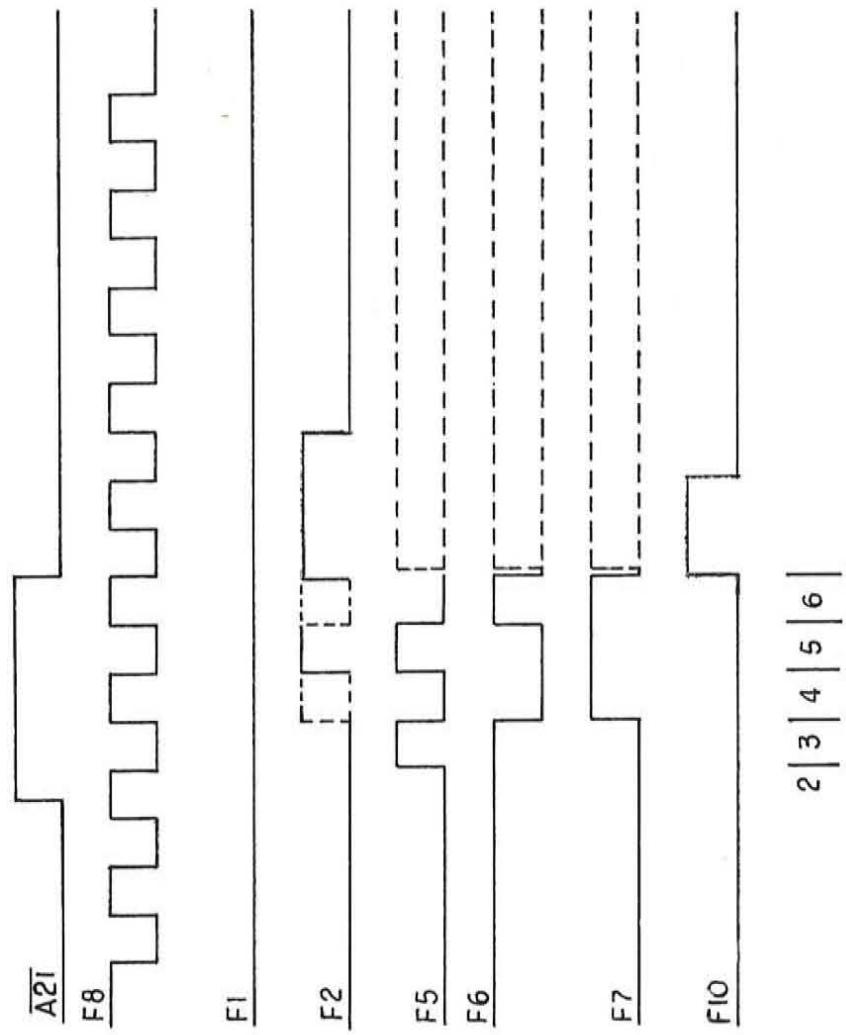


FIG.7

TRANSFER WITHOUT SYNC TIMING

V-4-4 Transfer With Sync Logic (8 word-times)

The following description assumes the read cycle, as described under "Reader Control Logic" is in the process of terminating.

The purpose of this loader operation is to transfer the assembled data word in fast access to a specific register in General Storage relative to the index (first) word. Note that the index word was stored in a random location with a "Transfer Without Sync" character. A specific pattern of "one after the other" is followed in properly locating each data word transferred subsequent to the index word.

Sequence

- (1) $\overline{A21}$ goes high to indicate the read cycle is complete. The logic decodes the "Transfer With Sync" character in the input register flip-flops F4-F7 and F9.
- (2) Flip-flop F1 is set by the first 1's flag bit to pass under the selected General Storage playback head coincident with t_{36} of $\overline{F8}$ after $\overline{A21}$ goes high.
- (3) F2 is set and F4 is reset one bit-time later via N24 and N45 respectively.
- (4) F1 is reset by the first 0's flag bit to appear coincident with t_{36} of $\overline{F8}$ at the playback of the selected track.
- (5) F4 is set at the next t_{36} of $\overline{F8}$ via N41.
- (6) F5, F6, and F7 form a binary counter. The count begins with the first t_{36} to occur after F4 is set. Note that the count begins at zero since these flip-flops were initially reset as part of the input character.
- (7) See items (3) - (5) of "Transfer Without Sync" description.

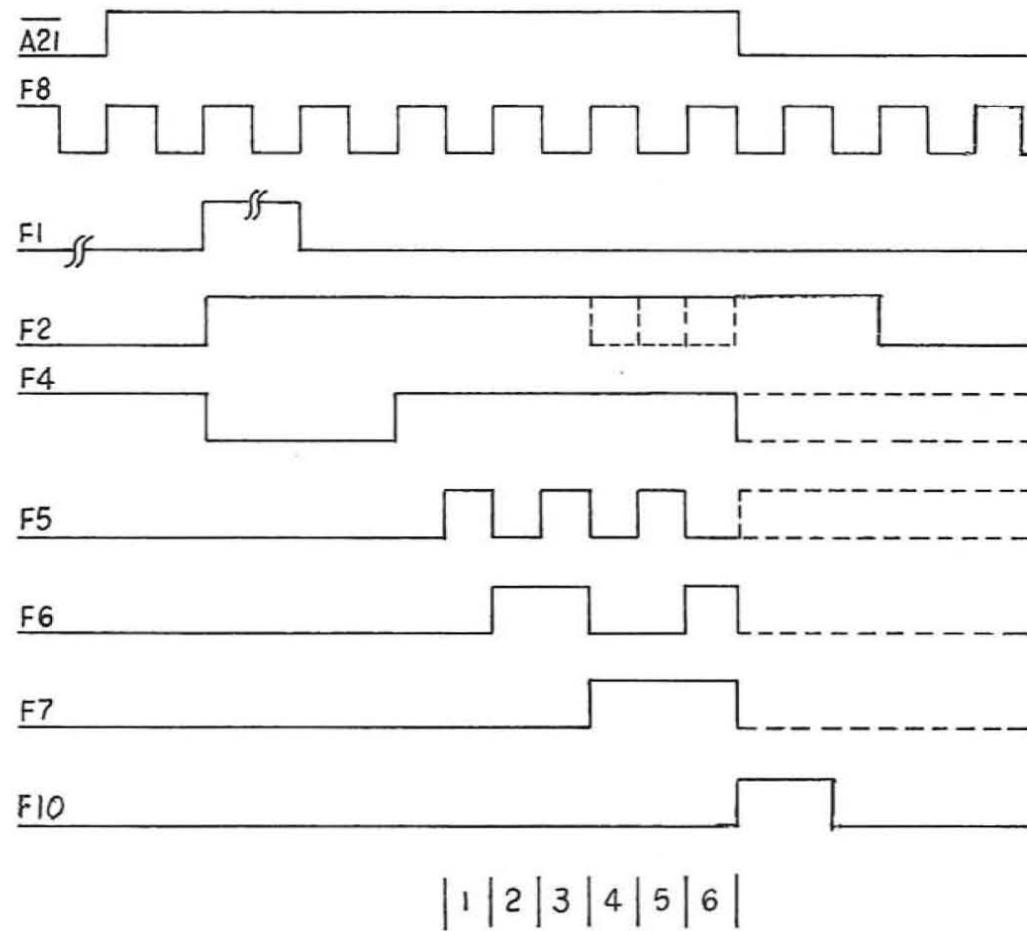


FIG. 8

TRANSFER WITH SYNC (8 WORD TIMES) TIMING

V-4-5 Transfer With Sync Logic (10 word-times)

The character for this operation is identical to that of "Transfer with Sync (8 word-time)" and its purpose is to index the first data word transfer to General Storage tracks 2, 3, and 4 with the single zero flag bit in Track 1. The only difference in operation is that the count, which is begun after the zero flag is found in track 1 of General Storage, is extended to ten instead of eight word-times. Thus the operation is defined separately from the standard "Transfer With Sync" by the position of the Track Selector switch, i.e., those positions which force playback from track 1 while selecting a different track for record.

Sequence

The sequence is identical to "Transfer With Sync (8 word-times)" except for item 6 which should read as follows:

(6) F5, F6 and F7 form a binary counter. The count begins with the first t_{36} to occur after F4 is set. Note that the count begins at zero since these flip-flops were initially reset as part of the input character. At the end (t_{36}) of count 3, F4 is reset via N44 because W11 is high. W11 is high whenever the Track Selector switch is in a position which forces playback 1 while recording in another track. F4 is set again two word-times later via N41. During the two-word time reset period of F4 the count is held up at 4. The count resumes with the first t_{36} after F4 goes high again. Thus, the count has been extended by two word-times.

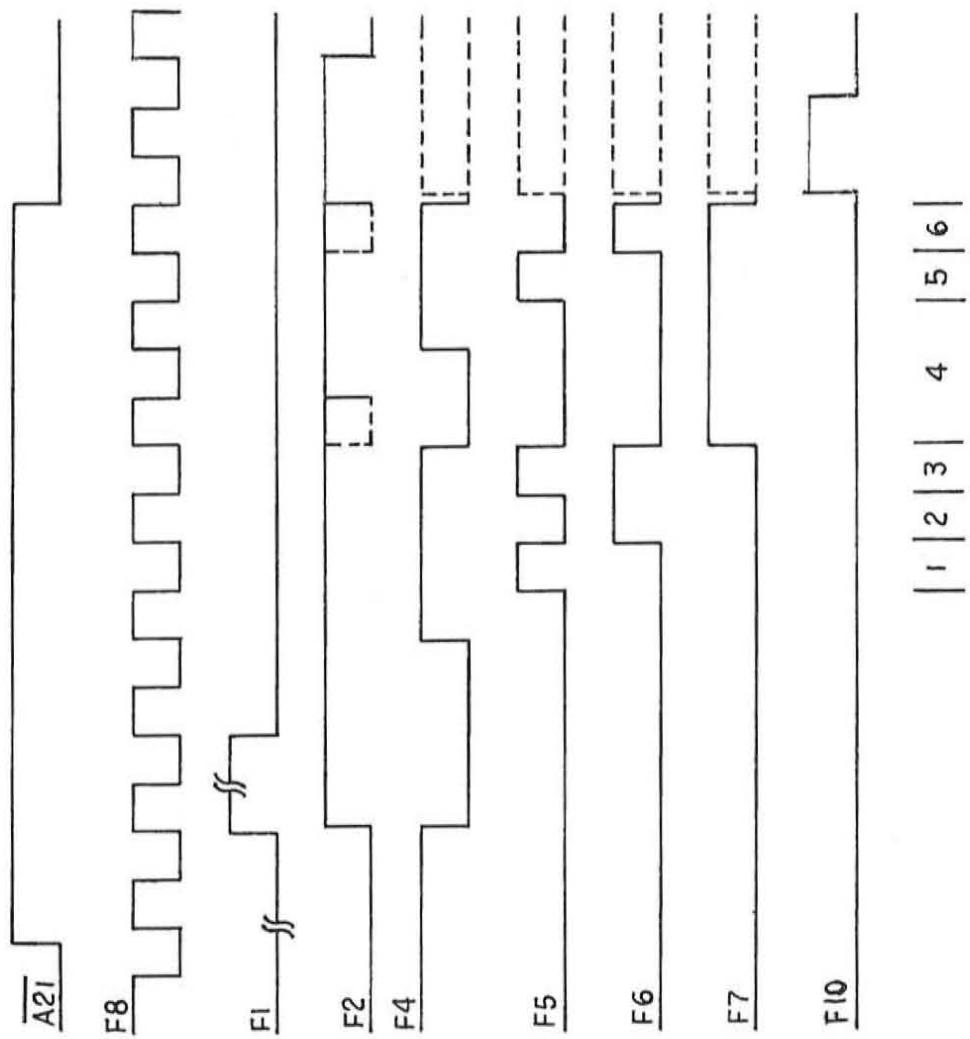


FIG. 9
TRANSFER WITH SYNC (10 WORD TIMES) TIMING

V-4-6 Stop Logic

The following description assumes the read cycle is in the process of terminating.

Sequence

- (1) $\overline{A21}$ goes high to indicate the read cycle is complete.

There is no logic in the loader which will respond to the Stop character, therefore, all sequencing ceases.

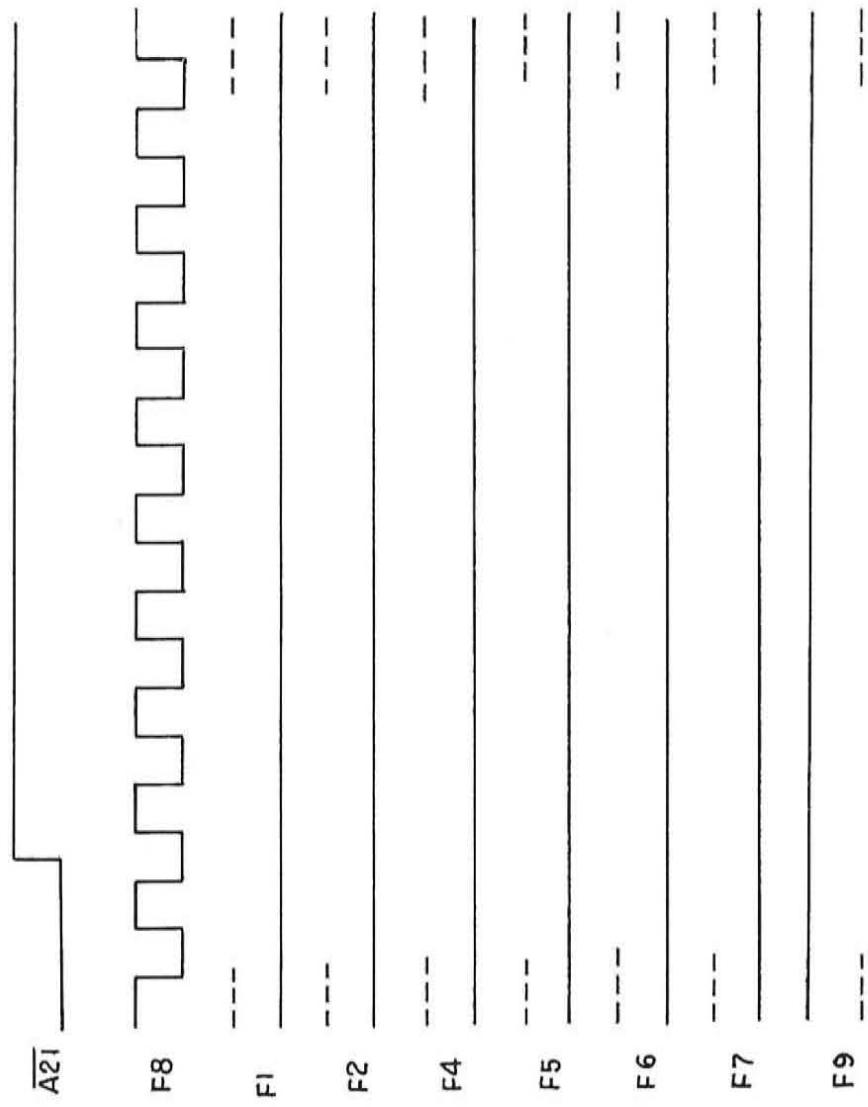


FIG. 10
STOP TIMING

V-4-7 Ignore Logic

The following description assumes the read cycle is in the process of terminating.

Sequence

- (1) $\overline{A21}$ goes high to indicate the read cycle is complete.

The logic decodes a "Transfer" character in the input register, however, flip-flops F5-F7, which form a binary counter in "Transfer" operations, are initially set as part of the character. The count, therefore, is already at maximum and the logic interprets this as the end of a transfer which actually never took place.

- (2) F10 is set by C67.

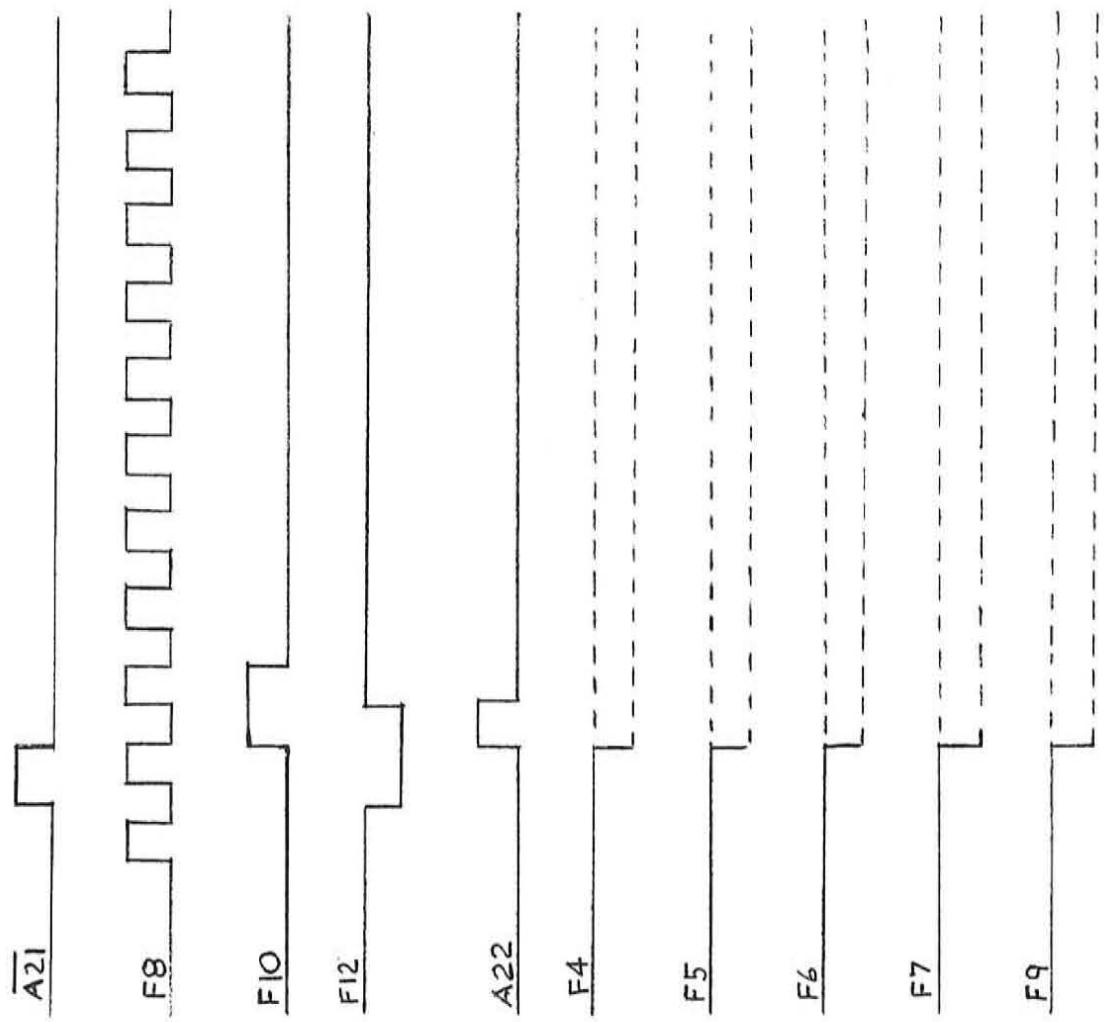
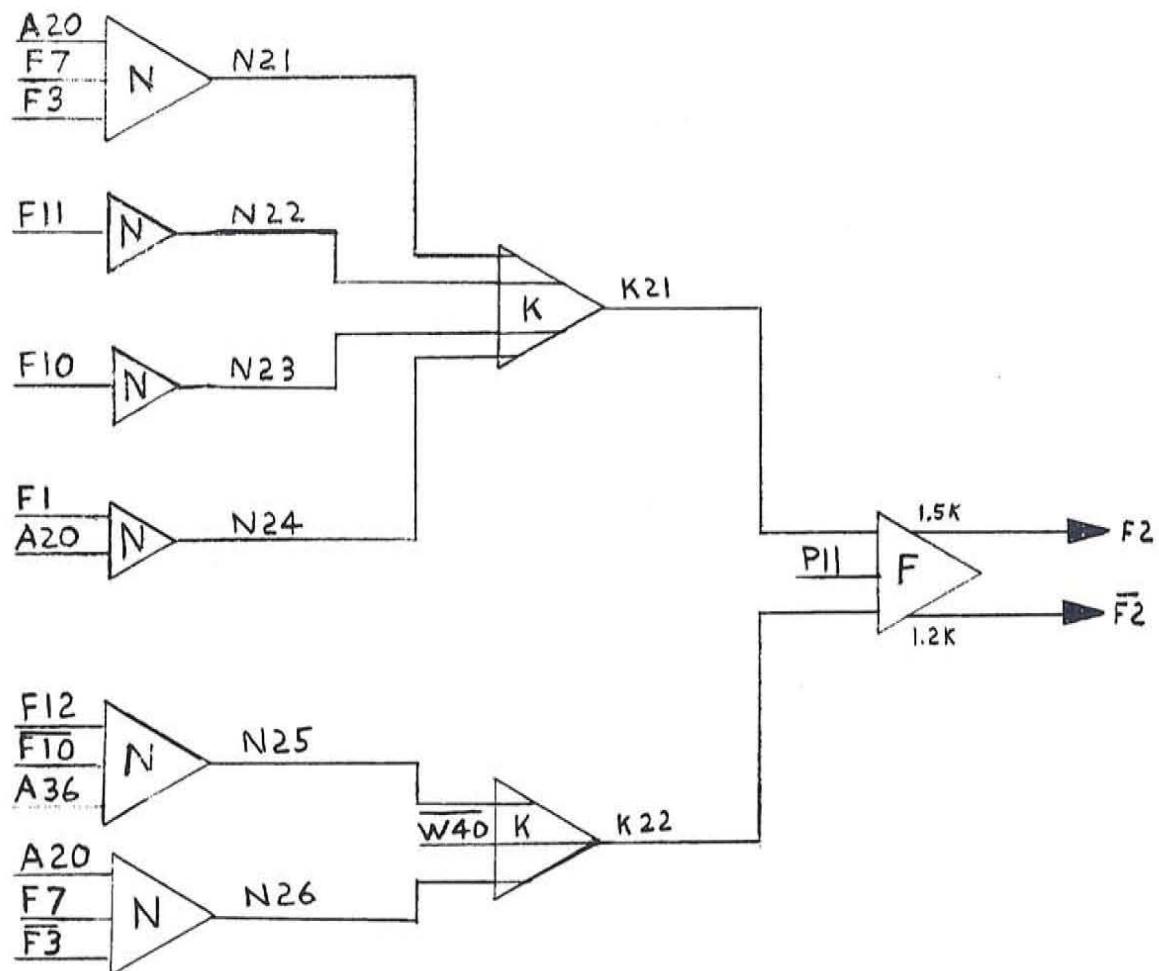
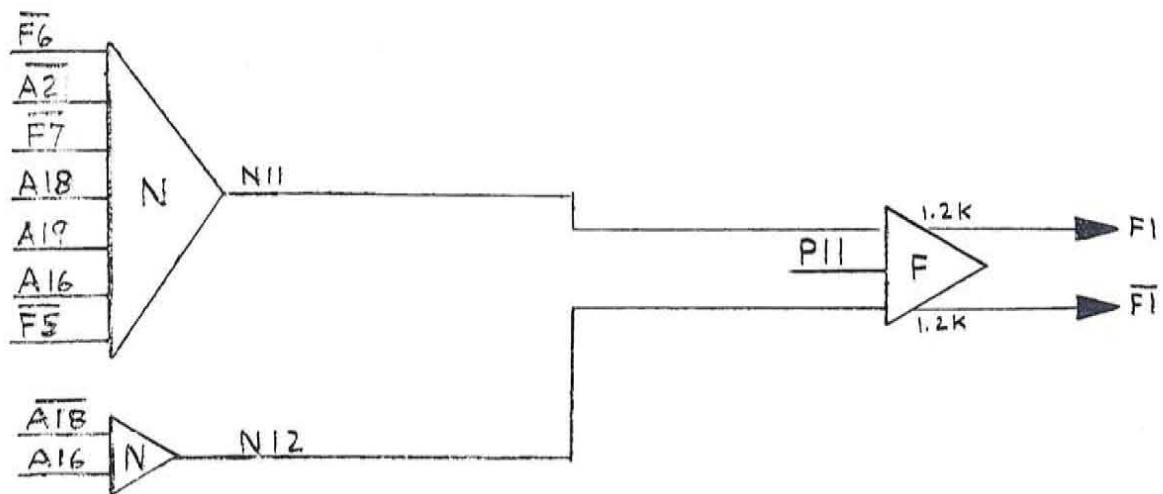


FIG. 11
IGNORE TIMING

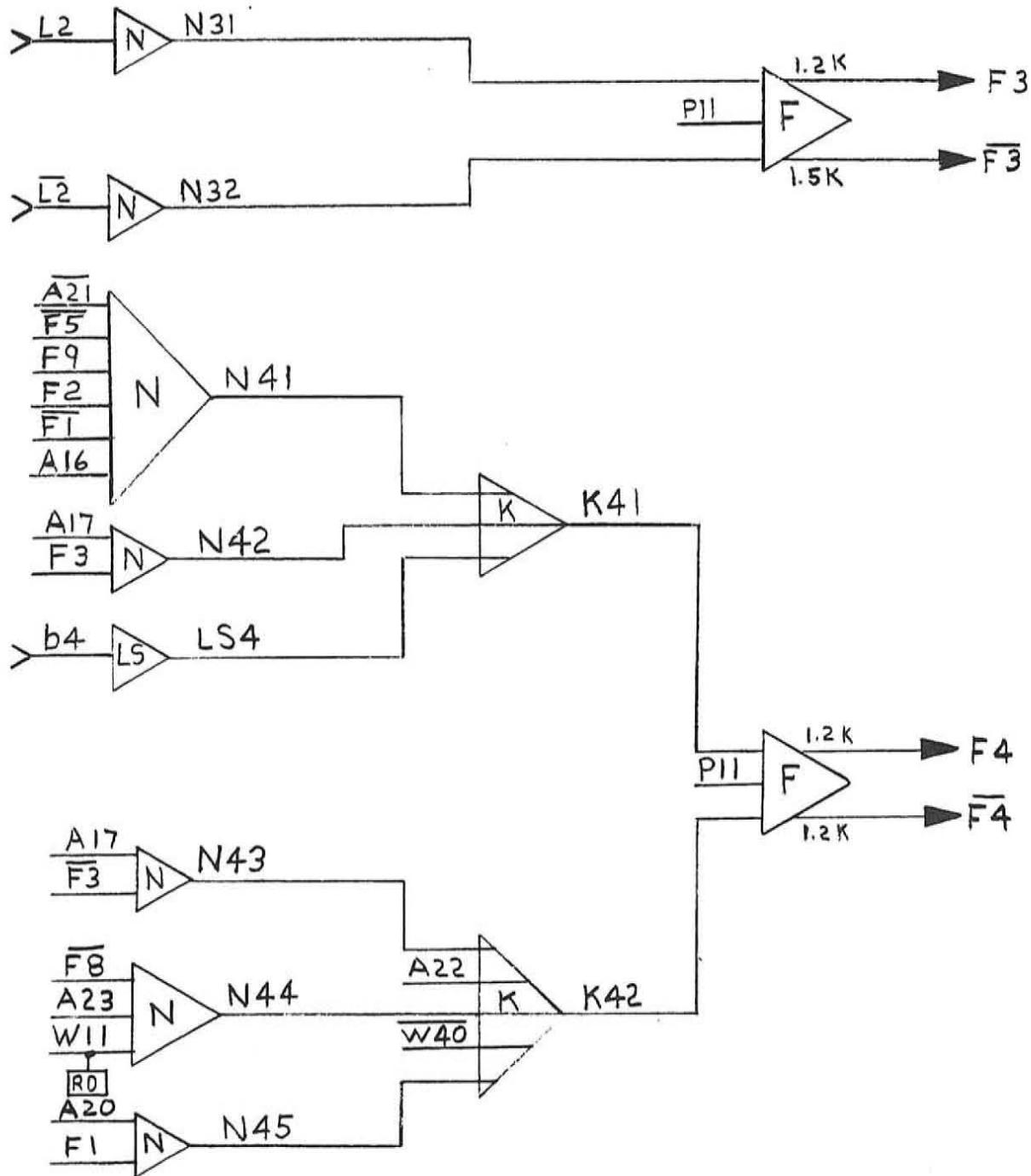
V-4-8 Turn-on Logic

When the loader is first turned on, it is necessary to force certain flip-flops into specific states in order to prevent the logic from sequencing out of control. This is accomplished by setting the five Input Register flip-flops F4-F7 and F9 to a "Stop" character via $\overline{W40}$. $\overline{W40}$ is derived from $\overline{W4}$ which is generated by the computer's on-off control logic as described in section II-3-4b. $\overline{W4}$ is wired via the connecting loader cable to the to the Clear/Rec T36 switch. $\overline{W40}$ is wired from the switch to the reset nets of F4-F7, the set net of F9, the Fast Access record gate L1, and the primed outputs of F10 and F12. With the Clear/Rec T36 switch in the normal or middle position, $\overline{W4}$ is connected to $\overline{W40}$ via a normally closed contact and the two are, therefore, logically equivalent. Thus, when the system is turned on, $\overline{W40}$ is generated, forcing the Stop code into the Input Register flip-flops, clearing the Fast Access to zero, and force resetting the reader sequencing control flip-flops F10, F12.

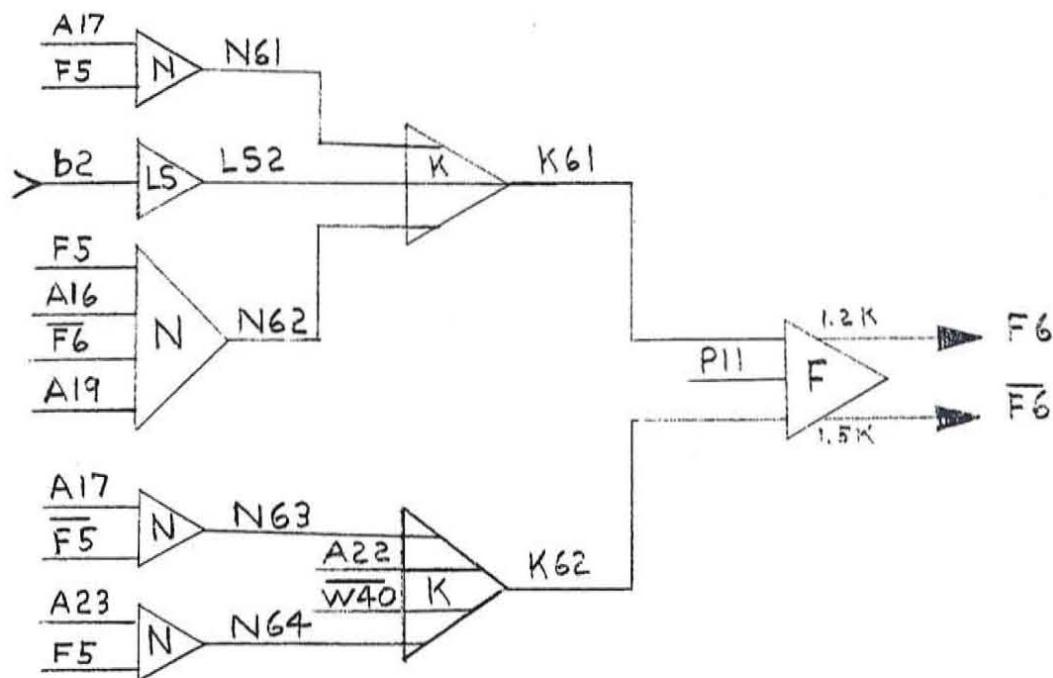
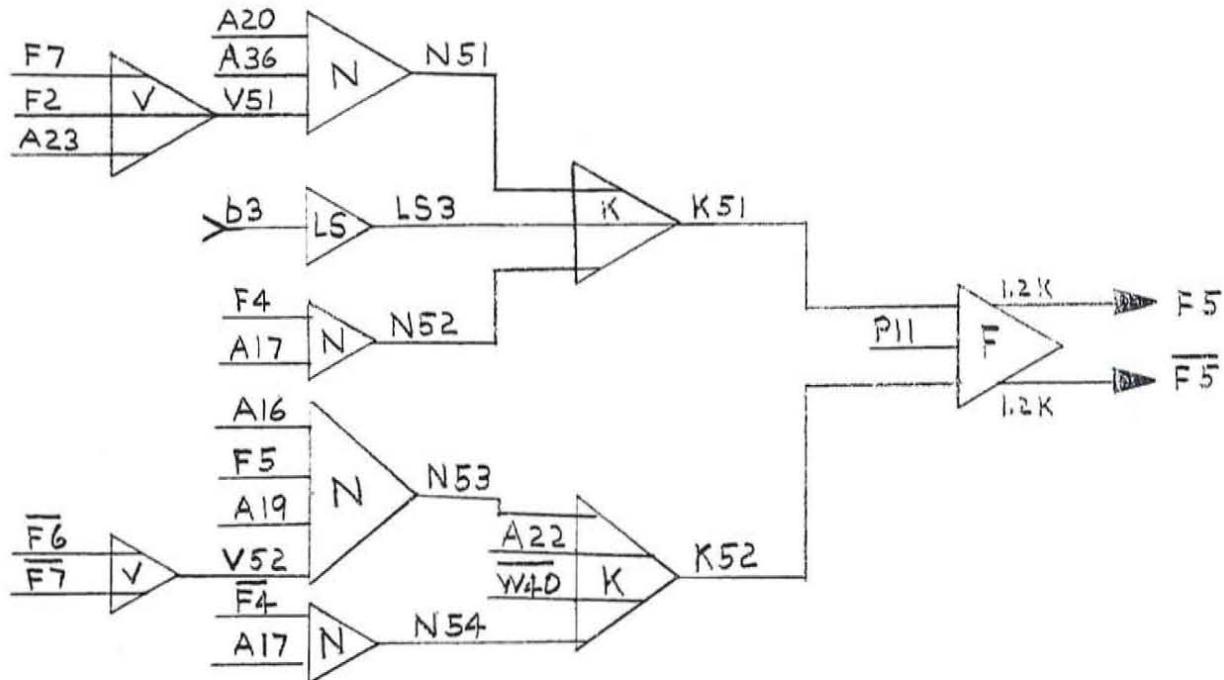
If the Clear/Rec T36 switch is operated in the Clear position, the normally closed contact that connected $\overline{W4}$ to $\overline{W40}$ is opened and $\overline{W40}$ is connected to ground instead. In this situation $\overline{W4}$ and $\overline{W40}$ are not equivalent. The end result, however, is an artificial generation of $\overline{W40}$ and the effects are as stated in the previous paragraph.



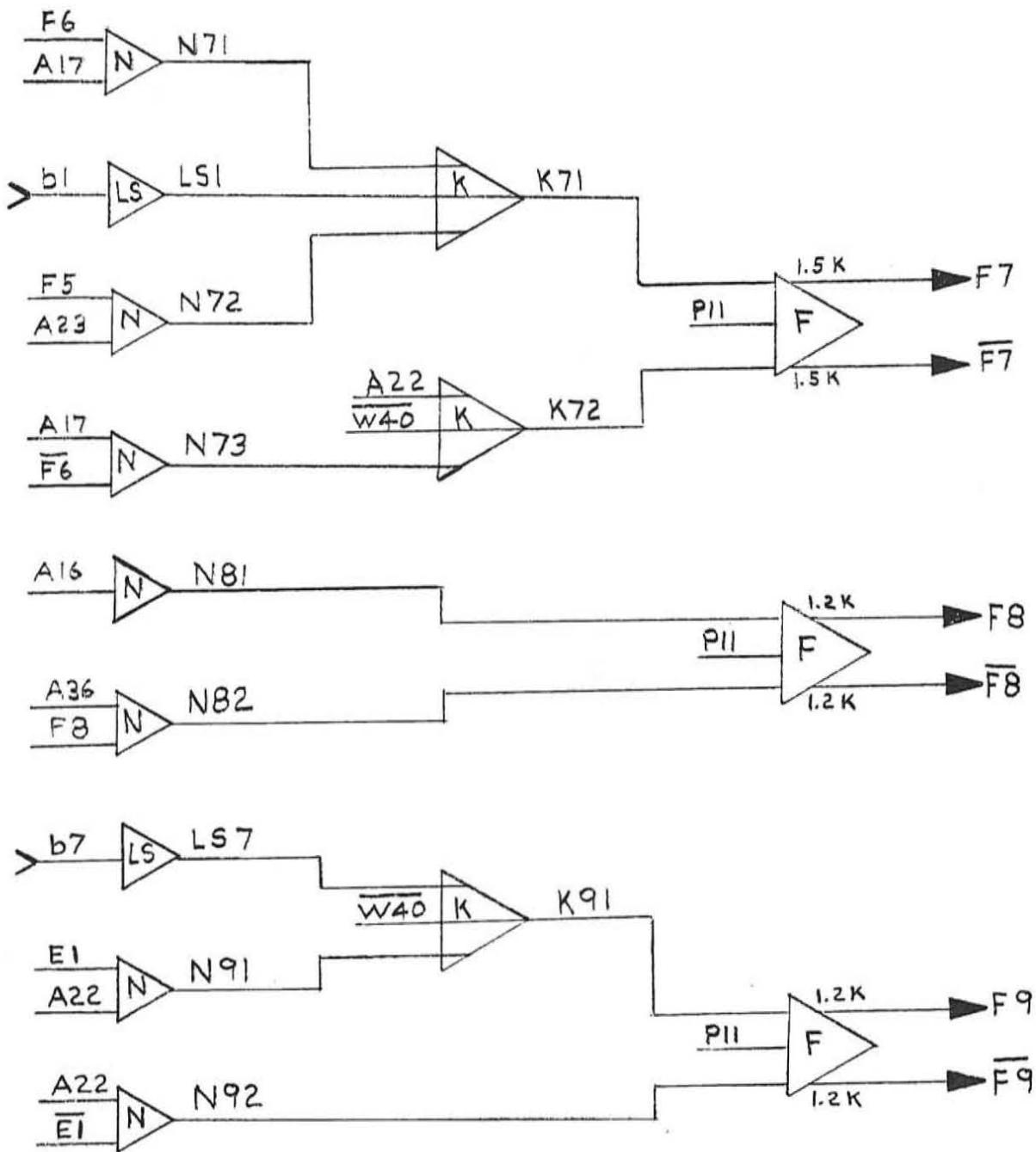
$F_1 \notin F_2$



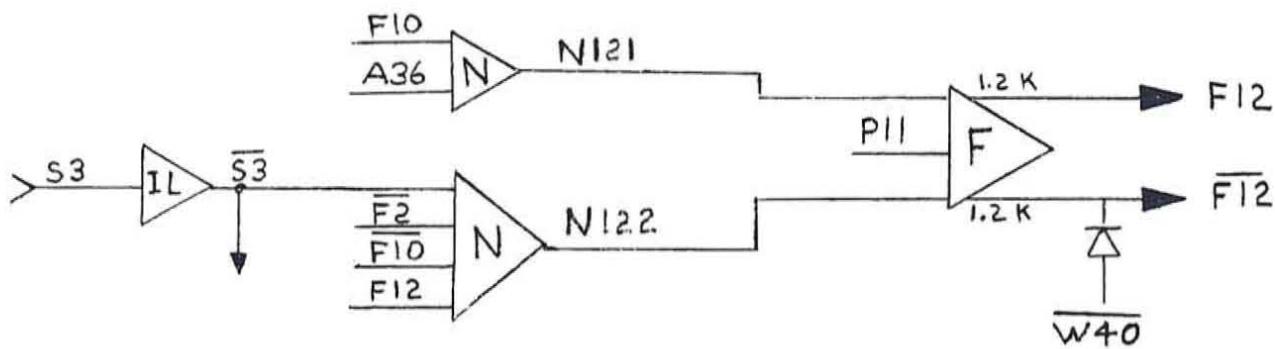
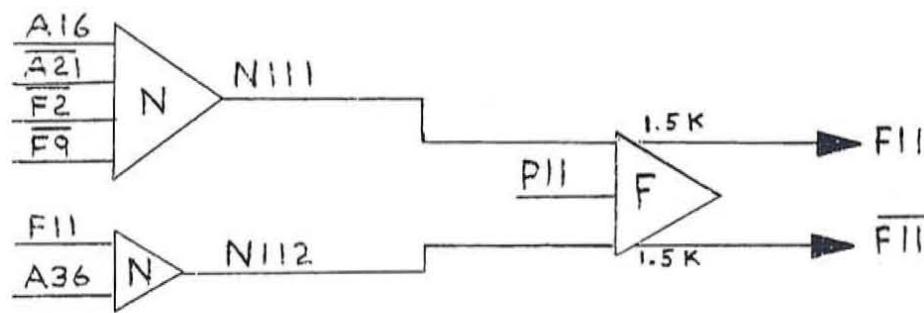
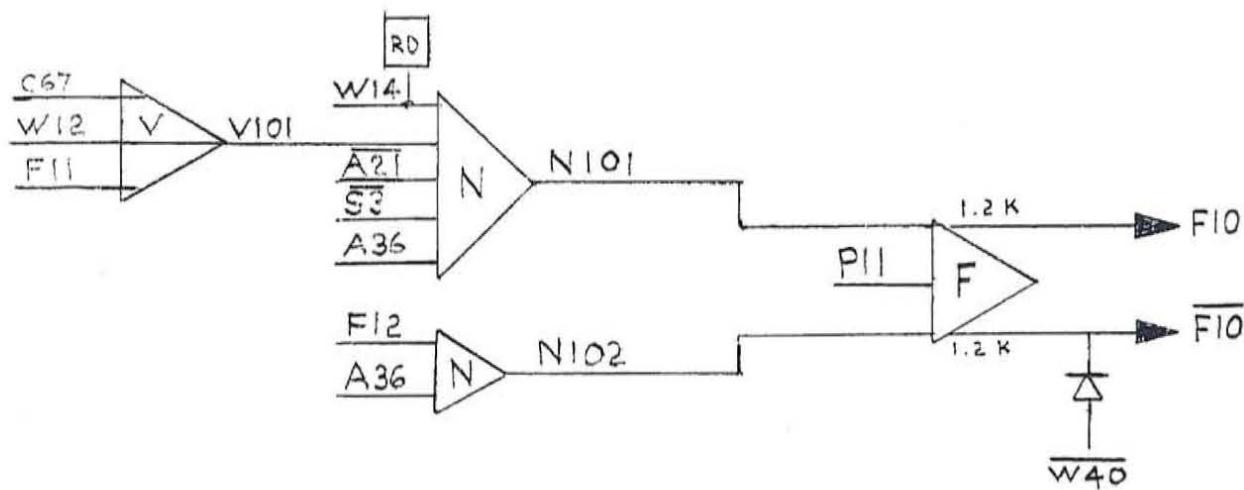
$F_3 \notin F_4$



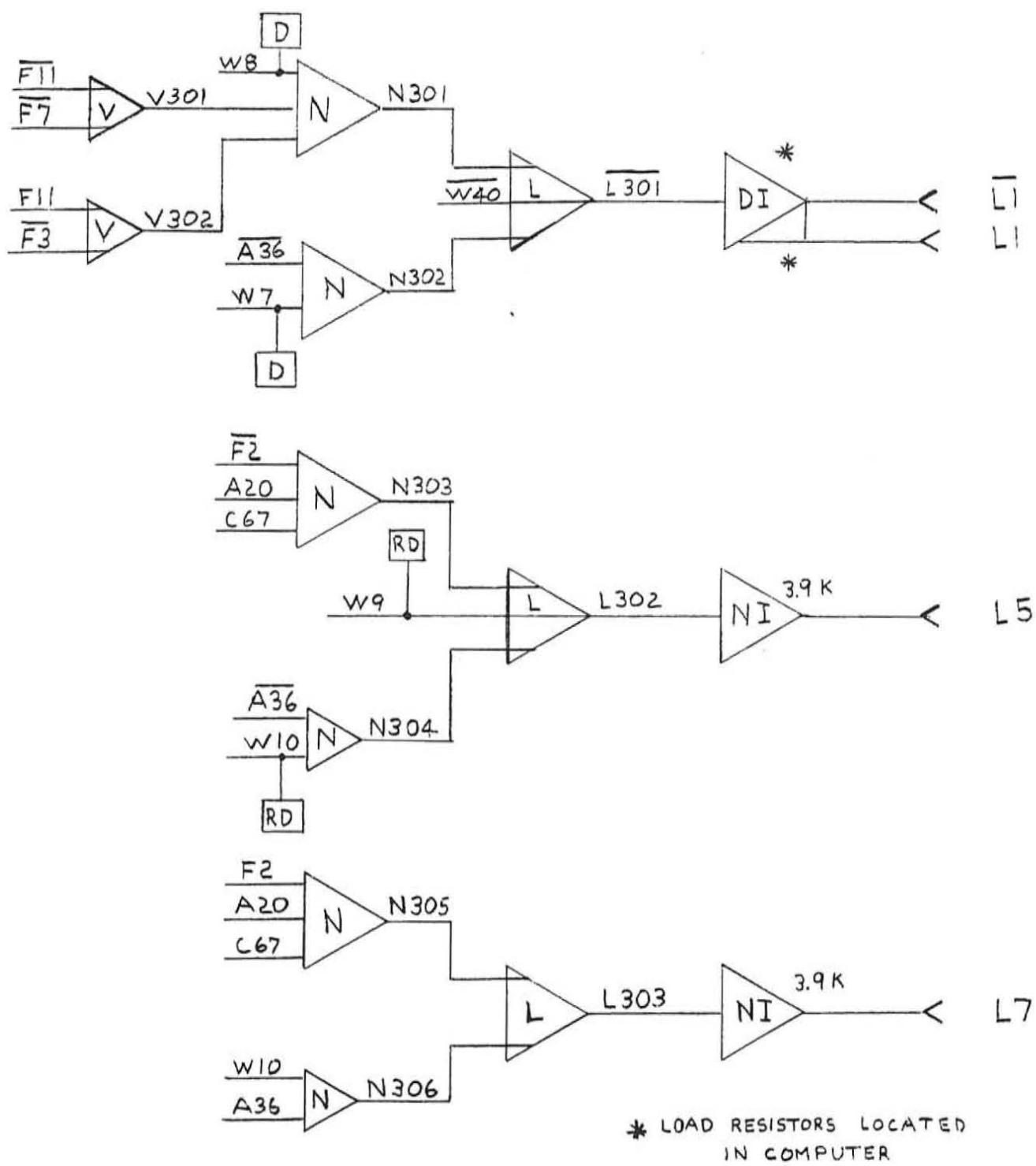
F5 & F6



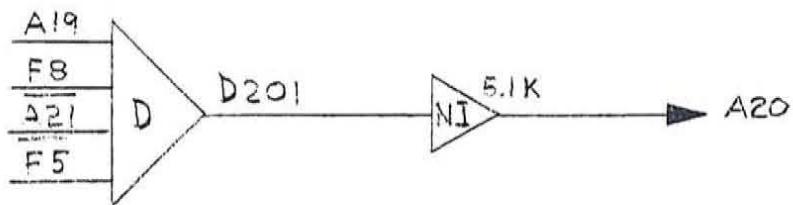
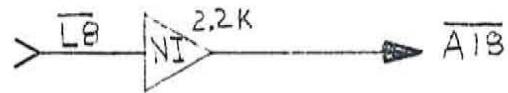
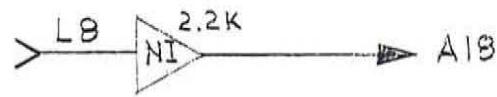
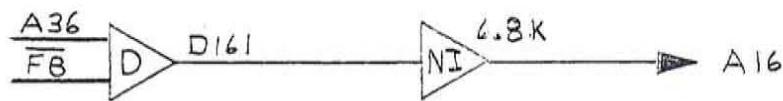
F7 - F9



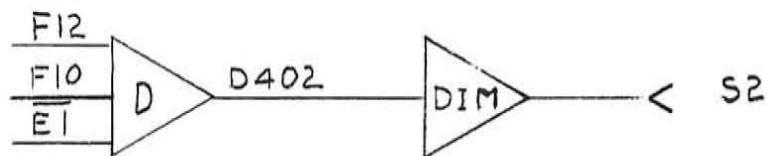
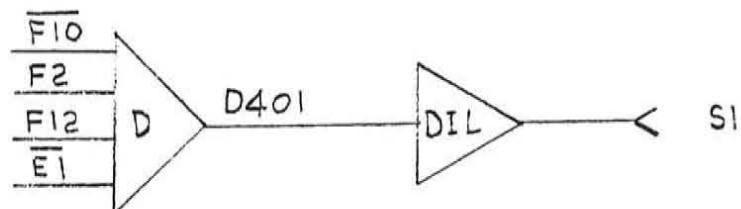
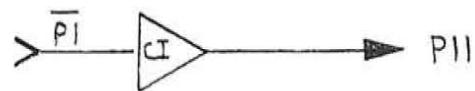
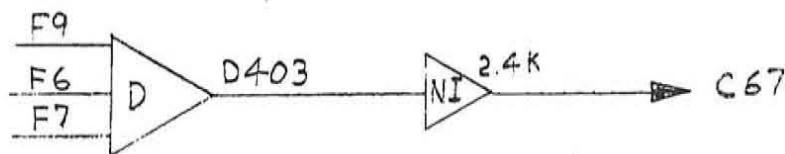
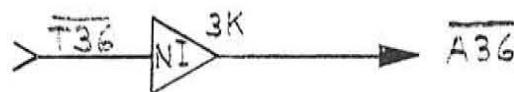
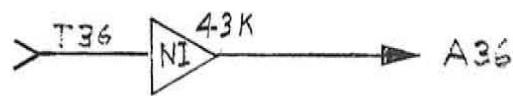
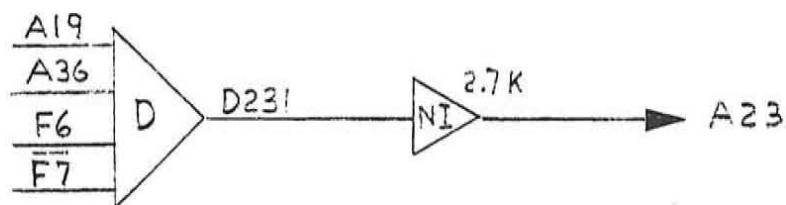
$F10 - F12$



L1, L5, L7



A16 — A22



MISC.

V-6 Circuits

The loader employs standard logical components (flip-flops, gates, non-inverters, dual inverters, etc.) which are described in section III of this manual. There are several circuits used specifically in the loader which will be described here.

NOTE: The electromechanical tape reader used to input information into the loader requires drive and control circuits. These circuits will not be covered in this section.

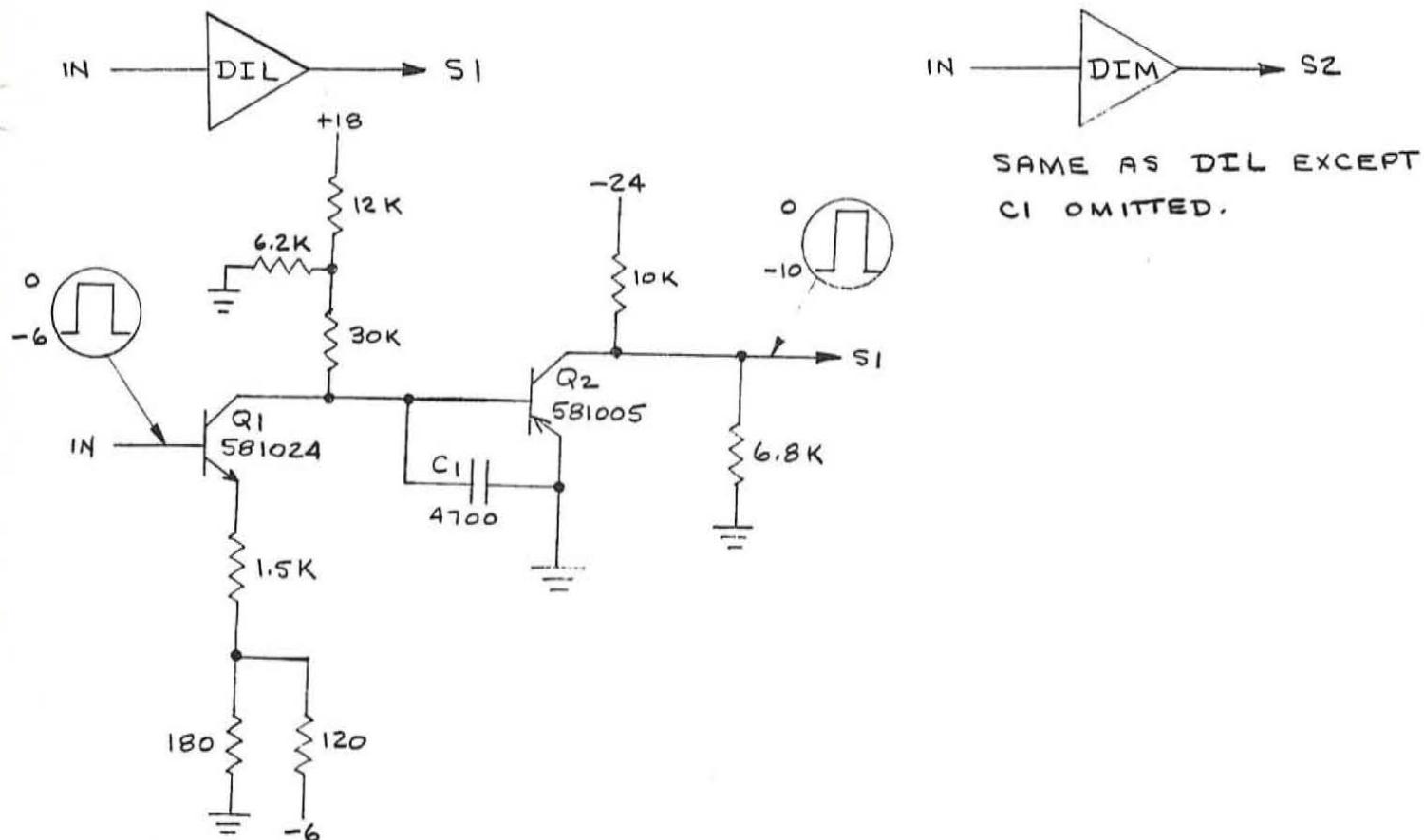
V-6-1 Device Clear and Device Common Circuits

The DIL and DIM circuits are simple double inverters used to repower signals. Only the real output is available from these circuits.

A high input turns on Q1. The base of Q2 is pulled negative turning on Q2 and producing a high output. With a low input, the output will be low.

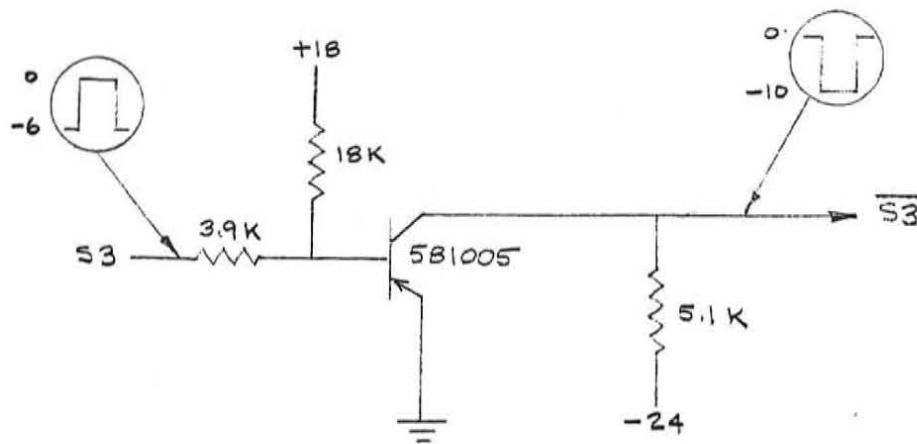
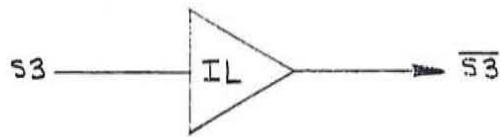
The DIL circuit is used to generate the clear and advance signal. Capacitor C1 delays the turn on of Q2. This prevents false inputs due to crossovers.

The DIM circuit is used to generate the read common signal.



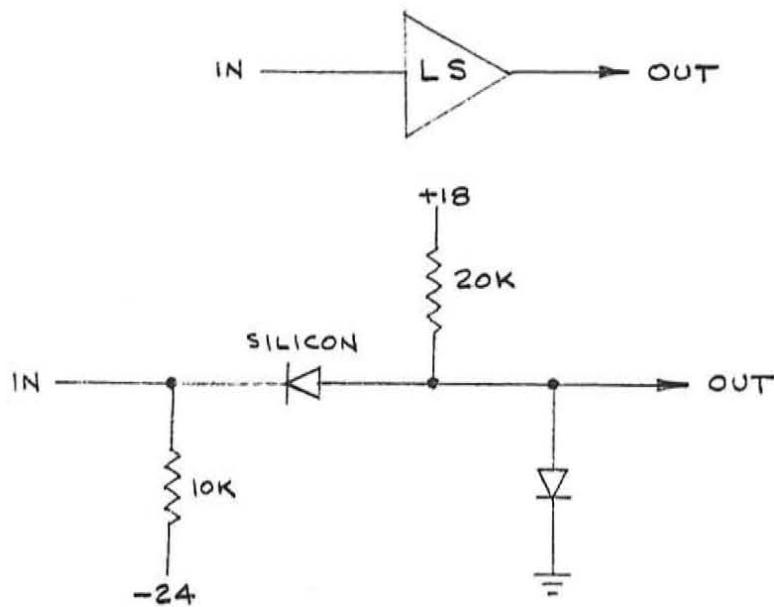
V-6-2 Device Busy Inverter Circuit

The IL circuit is a simple inverter. It is used to invert the reader busy signal S3.



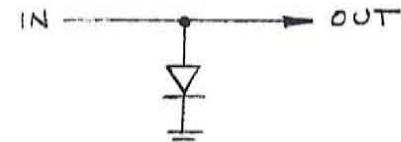
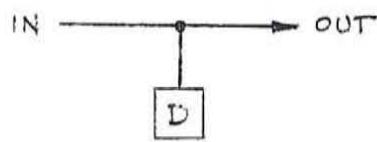
V-6-3 Level Shift Circuit

The LS circuit is used to shift the level of the tape character signals b_1, b_2, b_3, b_4, b_7 . The LS circuit shifts the level of the reader input signal approximately one volt positive.

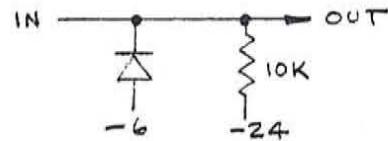
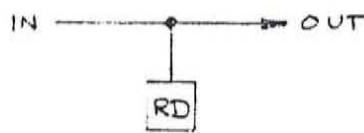


V-6-4 Clamping Circuits

Signals which are connected to the logic through a switch, require clamping circuits to hold the signal at the proper logic level (with the switch open). The D and RD circuits used for this purpose are shown below:



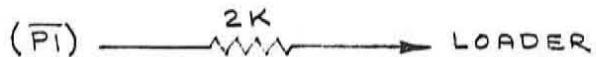
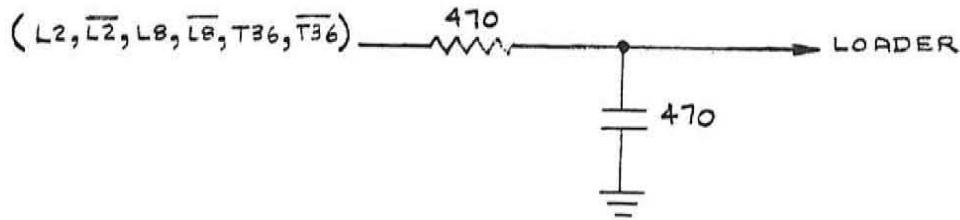
The D circuit clamps the signal to ground.



The RD circuit clamps the signal to -6 volts; the 10K resistor supplies the current required to drive the load.

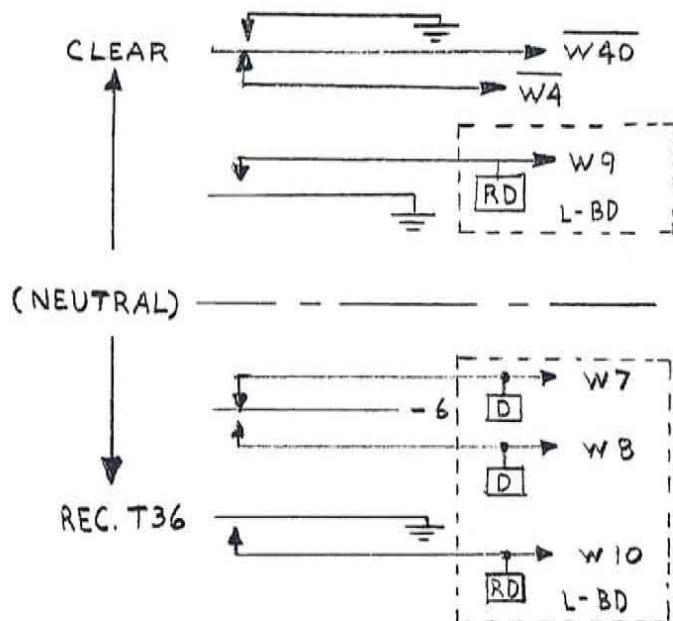
V-6-5 Noise Suppression Circuits

Signals L2, $\overline{L2}$, L8, $\overline{L8}$, T36, $\overline{T36}$, and $\overline{P1}$ require noise suppression. This is accomplished by use of an RC network or series R (at connector PL1) as shown below:

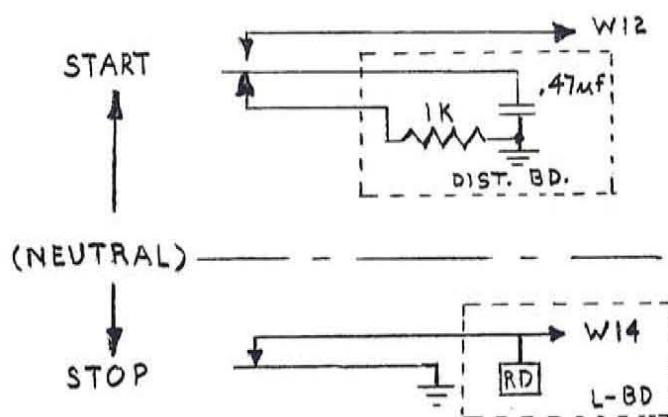


V-6-6

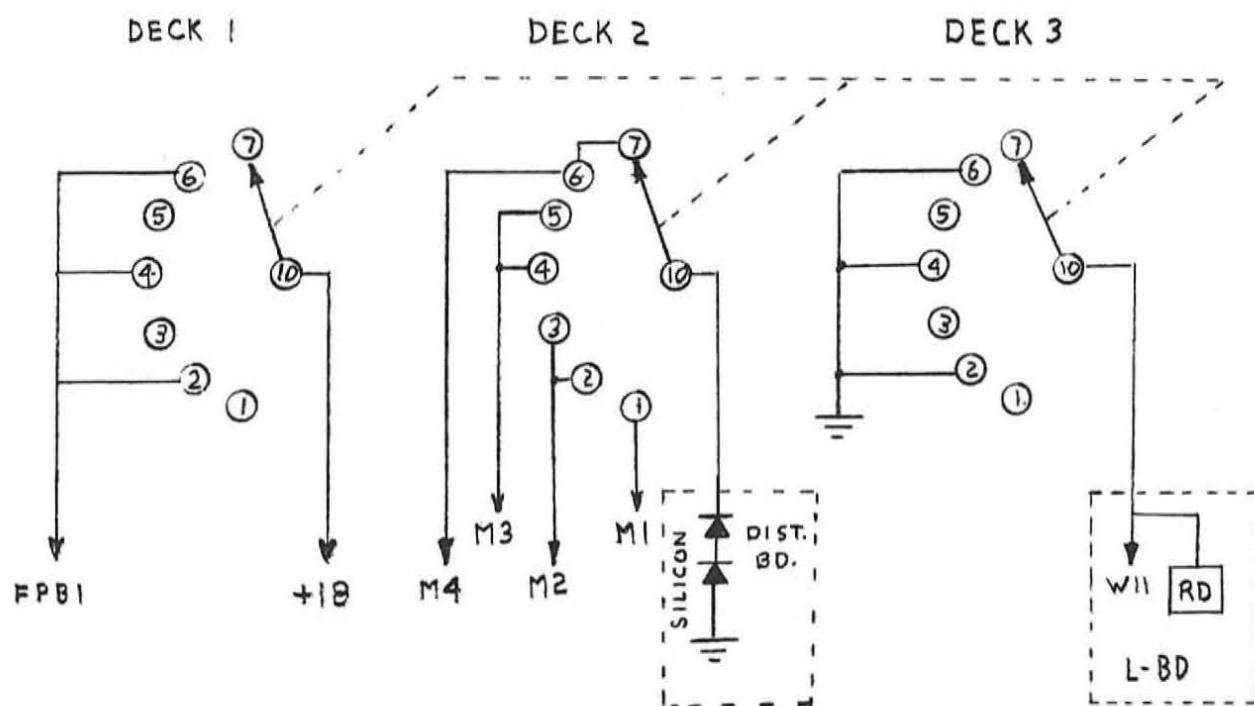
CONTROL PANEL SWITCHES



CLEAR-RECORD T36 SWITCH



TAPE START-STOP SWITCH



TRACK SELECTION SWITCH

V-7 Loader - Signal Glossary

| <u>Signal</u> | <u>Source</u> | <u>Function</u> |
|---------------|---------------|---|
| A16 | NI | Minimization |
| A17 | NI | Minimization |
| A18 | NI | G.S. Playback (repowered) |
| A19 | NI | Transfer or ignore code |
| A20 | NI | Minimization |
| A21 | NI | Character not being used and reader not busy |
| A22 | NI | Reset input register (F4-F7,F9) and start reader cycle |
| A23 | NI | Minimization |
| A36 | NI | T36 repowered |
| C67 | NI | Count six or seven during Transfer operation |
| E1 | Switch | Parity error signal |
| F1 | FF | Flag bit search FF |
| F2 | FF | Anti-repeat FF and path for fast access to G.S. |
| F3 | FF | Fast access playback (L2 delayed one bit time) |
| F4 | FF | Input register FF |
| F5 | FF | Input register FF |
| F6 | FF | Input register FF |
| F7 | FF | Input register FF |
| F8 | FF | Word counter |
| F9 | FF | Input register FF |
| F10 | FF | Reader control FF |
| F11 | FF | Fast access control FF |

| <u>Signal</u> | <u>Source</u> | <u>Function</u> |
|---------------|---------------|------------------------------------|
| F12 | FF | Reader control FF |
| FPB1 | Switch | Forces Selection to playback |
| | | Track 1 |
| L1 | DI | Fast access record |
| L2 | Pb | Fast access playback |
| L5 | NI | G.S. record zeroes |
| L7 | NI | G.S. record ones |
| L8 | Pb | G.S. playback |
| M1 | Switch | Selects Track 1 |
| M2 | Switch | Selects Track 2 |
| M3 | Switch | Selects Track 3 |
| M4 | Switch | Selects Track 4 |
| <u>P1</u> | Computer | Clock pulse |
| P11 | CI | Loader clock pulse |
| S1 | NI | Clear and advance reader |
| S2 | NI | Read common |
| S3 | Reader | Device busy |
| T36 | FF | Word time pulse |
| b1 | Reader | 1st channel code wire |
| b2 | Reader | 2nd channel code wire |
| b3 | Reader | 3rd channel code wire |
| b4 | Reader | 4th channel code wire |
| b7 | Reader | 7th channel bit code wire |
| <u>W4</u> | Computer | Sequencing signal |
| W7 | Switch | Record T36 pulses in accumulator |
| W8 | Switch | Normal record path for accumulator |
| W9 | Switch | Clear G.S. |

| <u>Signal</u> | <u>Source</u> | <u>Function</u> |
|---------------|---------------|---|
| W10 | Switch | Record T36 pulses in G.S. |
| W11 | Switch | Extend count from eight to ten word times during transfer operation |
| W12 | Switch | Tape start signal |
| W14 | Reader | Tape stop signal |
| <u>W40</u> | Switch | Loader Sequencing Signal |

SECTION VI

ADJUSTMENT AND TESTS

| | Page |
|-------------------------------------|-------|
| VI-1-5 Drum Speed Monitor | VI-1 |
| VI-2 Master Track | VI-2 |
| VI-3 Master Track Restoration | VI-7 |
| VI-4 Erasing | VI-9 |
| VI-5 System Test | VI-10 |

VI-1-5. Drum Speed Monitor.

The drum speed monitor potentiometer is located in the Sequencing circuit. It allows the drum speed monitor circuit to be individually adjusted.

The setting of this variable resistor should be attempted prior to loading a Micro program. The value of the variable resistor is 1000 ohms $\frac{1}{2}$ watt and the center arm is connected to the emitter of Q6 (2N1303).

1. Turn pot to its most counter clockwise setting.
2. With loader connected to computer, turn computer on.
3. When drum gets up to running speed, operate RST switch.
4. Turn pot SLOWLY in a clockwise direction until the ready light comes on.
5. Connect scope to the collector of transistor Q6. Continue adjusting pot in a clockwise direction until the peak to peak signal seen on the collector levels off at its minimum amplitude. (Do not rotate the pot more than is necessary to achieve the above.)
6. Lock pot in this position.

VI-2. MASTER TRACK

VI-2-1. General.

This procedure describes a method of master tracking a Monroe 1200 drum using a Monroe 1200 Memory Ass'y. It is assumed that the drum assembly has all heads mounted and adjusted .001" away from the surface of the drum.

VI-2-2. Equipment Set Up.

1. Remove drum cover from base plate.
2. Check for foreign matter on drum.
3. Mount drum in frame.
4. Connect ground straps from Memory Assembly (one on each end) to top of drum base plate. Use $\frac{1}{4}$ " 6-32 screws to secure ground straps.
5. Wire A.C. to drum motor.
6. Clean "Master Disc" mounting screw holes.
7. Mount "Master Disc" to drum.
8. Adjust disc for maximum of $^{+}.0001"$ runout.
9. Mount Master Disc head assembly block to base plate.
10. Adjust pins on head assembly block for concentricity with circumference of drum.
11. Mount Master Disc head assembly using proper shim for disc track alignment.
12. Adjust heads to .001" away from surface of disc.
13. Turn drum on.
14. Using a high gain pre-amplifier, check playback amplitude of master disc heads. SHORT leads of the probes before looking across heads. If sufficient playback is obtained (between 5 & 10 millivolts) proceed to next step. If playback amplitude is insufficient, turn drum off and back off heads from disc surface and remove head assembly. Repeat steps 11 through 14.

15. With scope probe across master disc clock head, tune head for maximum amplitude using capacitor decade box, with ranges from .001 to .01 mfd.
16. Remove scope probe.
17. Turn drum off.
18. Connect master disc clock and sector heads to their proper amplifiers on the Memory Assembly using two wire shielded cables. Wire as per fig. 2-1.

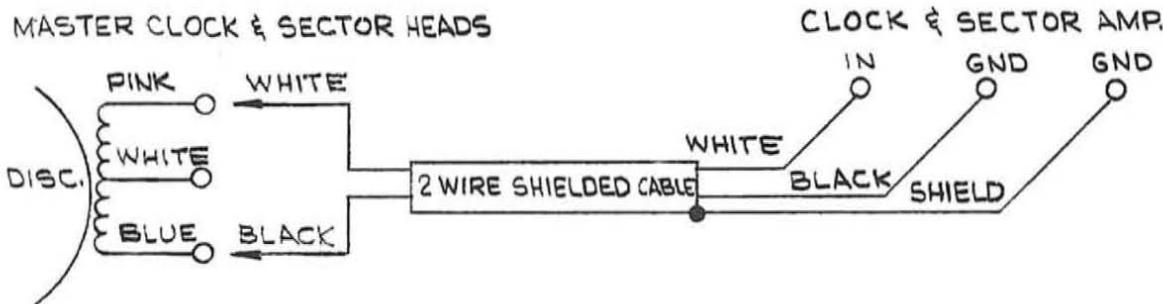


FIG. 2-1

19. Turn drum on.
20. Turn power on for Memory Assembly.
21. Scope pre-amplifier dual trace type. Use 10 to 1 probes. Scope trigger set to internal sync positive. Check for P1 and P1 at the output of clock generator circuit on Memory Assembly.
22. Connect probe to sensing point of sector amplifier. Adjust triggering to sync on rise time of input signal. Compare sweep with fig. 2-2.

NOTE: The input signal consists of binary one's and zero's which will cause superimposed traces on the scope. The input signal will have 4 times as many zero's as it has one's. therefore the brightest of the superimposed traces will represent the zero's on the drum. The background light traces should be ignored at this point.

23. Set up sensing between the two heads as per fig. 2-2. To change position of clock use capacitor decade box with ranges from .001 to .01 mfd across clock head. Select proper capacitor to shift clock to proper position.

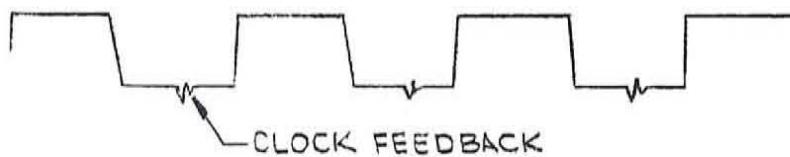


FIG. 2-2

24. Connect T29 to external sync input.
Change sync trigger to external sync positive.
Check result of sensing.
Scope Z modulation to Pl.
Probe A - connect to T29.
Probe B - connect to T36.
Set sweep to 5 us/cm.
Set mode selector switch to alternate position.
Adjust trigger for steady sweep.
Compare sweep with fig. 2-3.

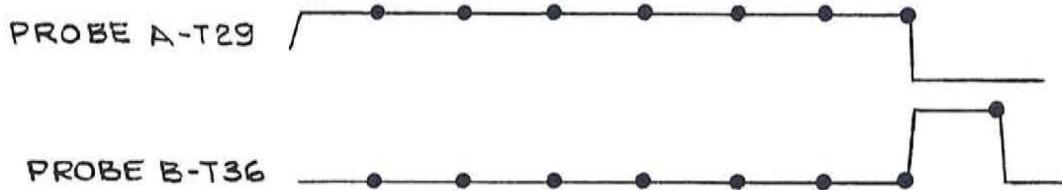


FIG. 2-3

25. Reduce intensity of scope to increase modulation contrast.
Probe A - T29 should be high for seven clock pulses.
Probe B - T36 goes high at the end of T29 and gets reset at the next clock pulse.
26. Remove all probes.
Remove Z modulation wire.

VI-2-3. Recording of Clock Track

Record clock in two places - spare clock head and main clock head.

1. Equipment set up and checked as per VI-2-2 Step 21
2. Disconnect the G.S. record L7 and L5 inputs and connect L5 to ground.

3. Connect record 3 wire shielded cable from G.S. record output to clock head. Wire as per fig. 3-1. Record pot should be set to its maximum resistance position.

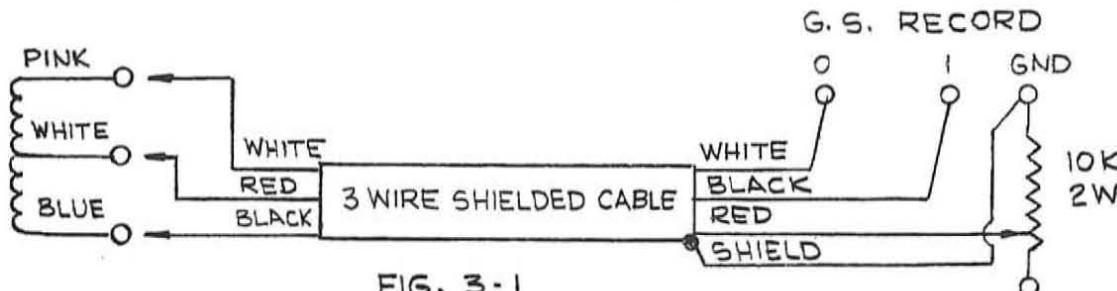


FIG. 3-1

4. Turn drum on.
5. Turn power on for Memory Assembly.
Wait for Ready light to come on.
6. Turn record pot to its minimum resistance setting for one second and return pot to its original setting.
7. Turn off ready light by depressing alternate track switch.
8. Remove record wires from Memory Assembly removing red wire from G.S. record first.
9. Check results of recording.
The pattern should meet all relevant specifications.

VI-2-4. Recording of Sector Track

Record sector in two places spare sector head and main sector head.

1. Equipment set up and checked as per VI-2-2 Step 26
2. Disconnect the G.S. record L7 and L5 inputs and connect T29 and T29 to L7 and L5 respectively.
3. Connect record 3 wire shielded cable from G.S. record output to sector head as shown in fig. 4-1. Record pot should be set to its maximum resistance position.

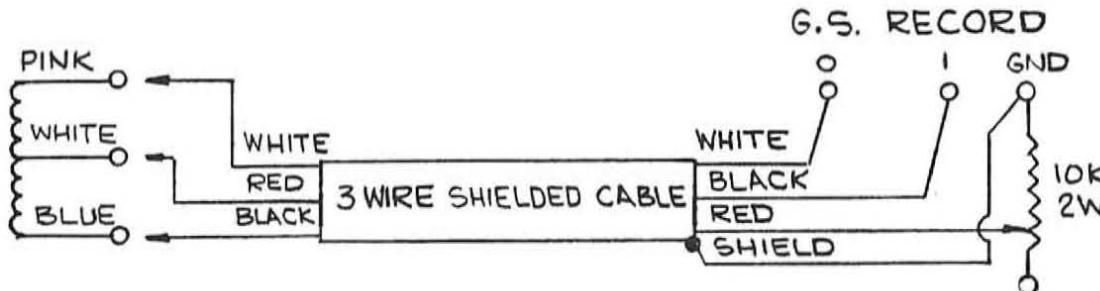


FIG. 4-1

4. Turn drum on.
5. Turn power on for Memory Assembly.
Wait for ready light to come on.
6. Turn record pot to its minimum resistance setting for one second
and return pot to its original setting.
7. Turn off ready light by depressing alternate track switch.
8. Remove wires from Memory Assembly removing red wire from
G.S. record first.
9. Check results of recording.
The pattern should meet all relevant specifications.

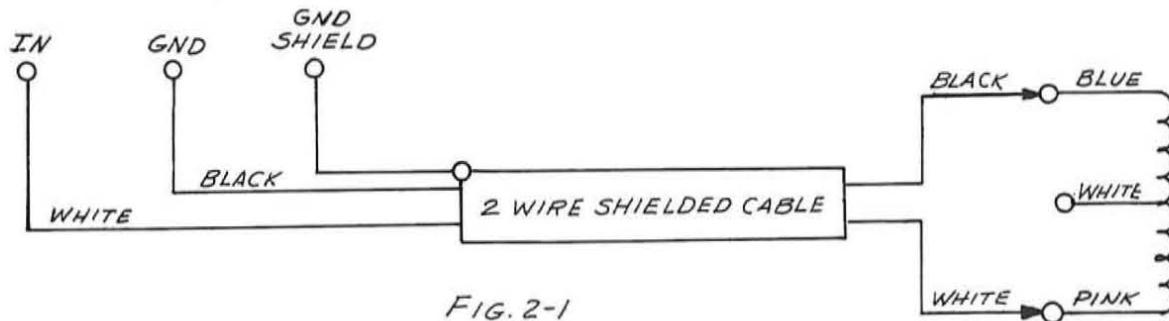
VI-3. MASTER TRACK RESTORATION

VI-3-1. General.

Master tracks can be restored by transferring the proper spare track.

VI-3-2. Clock & Sector Tracks

1. Power off, drum stopped.
 2. Disconnect both ends of the shielded cables to the clock and sector playback amplifiers.
 3. Connect 2 wire shielded cables from the spare clock and sector heads to their respective amplifiers on the Memory Ass'y as shown in Figure 2-1.



4. Turn power on.
 5. Erase defective master track.
 6. Check Pl at output of clock generator circuit.

NOTE: If clock track is the only track to be restored bypass step #7 & step #11(b).

7. Check sensing in sector track playback amplifier. If rise time of clock is not centered on the information, shift sensing by tuning input to clock amplifier with a capacitor decade box with ranges from .001 to .01 mfd.
 8. Turn power and drum off.
 9. Disconnect the G.S. record cable from the G.S. record circuit.
 10. Connect 3 wire shielded cable from G.S. record output to head on track to be restored. Wire as shown in Figure 2-2. Record

pot should be set at its maximum resistance position.

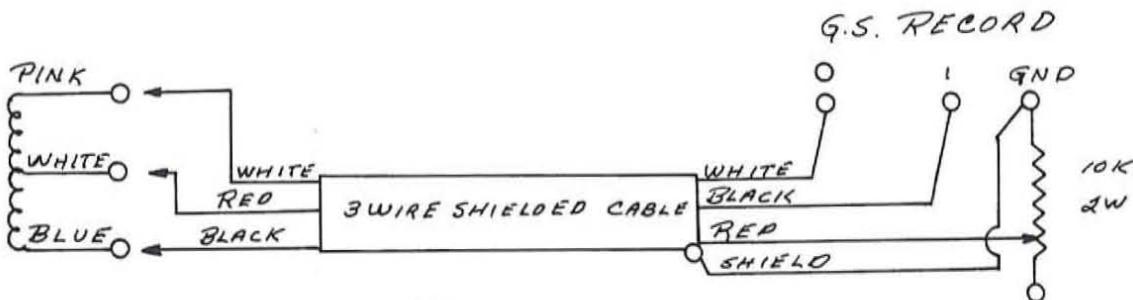


FIG. 2-2

11. (a.) In order to record a new clock track, disconnect the G.S. record inputs and connect L5 to GND.
- (b.) In order to record a new sector track, disconnect the G.S. record inputs and connect T29 to L5 and T29 to L7.
12. Turn Power on and wait for ready light to come on.
13. Turn record pot to its minimum resistance position for one second and return pot to its original setting.
14. Turn power off.
15. Remove record cable.
16. Check results of recording. The pattern should meet all relevant specifications.
17. Power off, drum stopped. Remove cables from spare tracks. Restore computer connections.
18. Check sensing between main clock and sector track. Adjust sector head if needed.

VI-4. ERASING

VI-4-1 General

1. In all cases where there is a record and playback head on the same track, the record head should be used for erasing.
2. 60 Cycle Erase Currents

| <u>Head</u> | <u>Peak Current</u> |
|-----------------|---------------------|
| General Storage | 475 - 750 ma. |
| Fast Access | 200 - 300 ma. |

VI-4-2. Fast Access

In erasing the Fast Access track, all 4 head wires should be disconnected from the record circuit (with power off) prior to erasing. The erase leads should be connected between the center tap and one side of the head. After erasing, power should be turned off prior to reconnecting head leads.

VI-4-3. General Storage

In erasing General Storage tracks, erase leads should be connected to points on the Drum Selection Ass'y. allocated for this purpose. For example: in order to erase track 3 connect erase leads to empty pin labelled 3 and to head wire labelled 3.

VI-5. SYSTEM TEST

VI-5-1. General

The following are the essential steps in factory test of a processor. Field test would normally start at VI-5-4 step 7.

VI-5-2. Pre Checks

1. Machine assembled less drum.

Vibrate machine for $\frac{1}{2}$ hour on a pallet.

Check for any mechanical defects.

2. Complete machine. The following should be performed before the line cord is connected.

3. Set all potentiometers to mid-range position.

4. Check transformer taps:

Select 115 v primary taps.

Select normal secondary taps.

5. Check power points for shorts to ground and to each other.

VI-5-3. Power Checks

1. Connect dummy connector to J1. Connect line cord.

2. Set variable transformer for 115 v at secondary (Export Only).

3. Apply power and check D.C. voltages at load side of fuses.

4. Adjust -6Z.

VI-5-4. Signal Checks

1. Set line monitor (W1) for 85 volts A.C.

2. Set clock width.

3. Set sequencing clock monitor.

4. Connect loader to J1.

5. Check Fast Access by recording and playing back T36 pattern. Balance record currents.

6. Check all General Storage tracks by recording and playing back T36 pattern. Balance record currents.

7. Connect input-output device and trouble shoot using test programs.
8. Take D.C. voltage margins.
9. Take A.C. voltage margins.
10. Check for proper drum acceleration and power relay operation at 90 volts A.C.
11. Load customer program and run test routine.
12. Run machine with customer program with covers on.
13. Set line monitor (W1) for 95 volts A.C.

SECTION VII

LOCATION OF ASSEMBLIES & RUNNING SHEETS

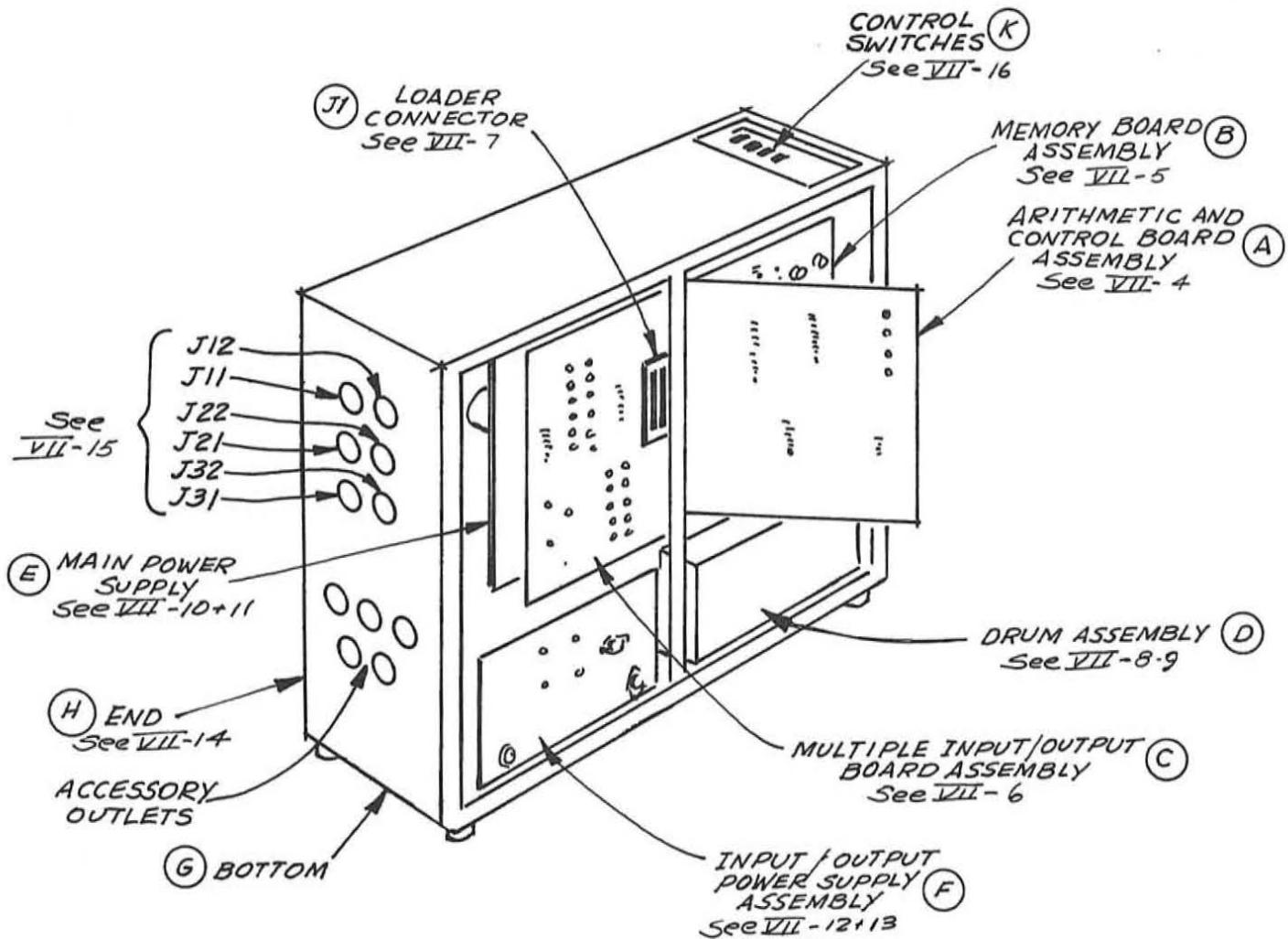
| | Page |
|--|----------------|
| Major Assembly Locations | VII-3 |
| Arithmetic and Control Board (Board A) | VII-4 |
| Memory Board (Board B) | VII-5 |
| Input-Output Board (Board C) | VII-6 |
| Loader Connector | VII-7 |
| Drum Unit Assembly | VII-8, VII-9 |
| Main Power Supply | VII-10, VII-11 |
| Input-Output Power Supply | VII-12, VII-13 |
| AC Wiring | VII-14 |
| Input-Output Connections | VII-15 |
| Control Switch Breakdown | VII-16 |
| Running Sheets | VII-17, VII-51 |

This section is devoted to aiding the serviceman in locating circuits and following their continuity through the computer with the utmost speed and ease.

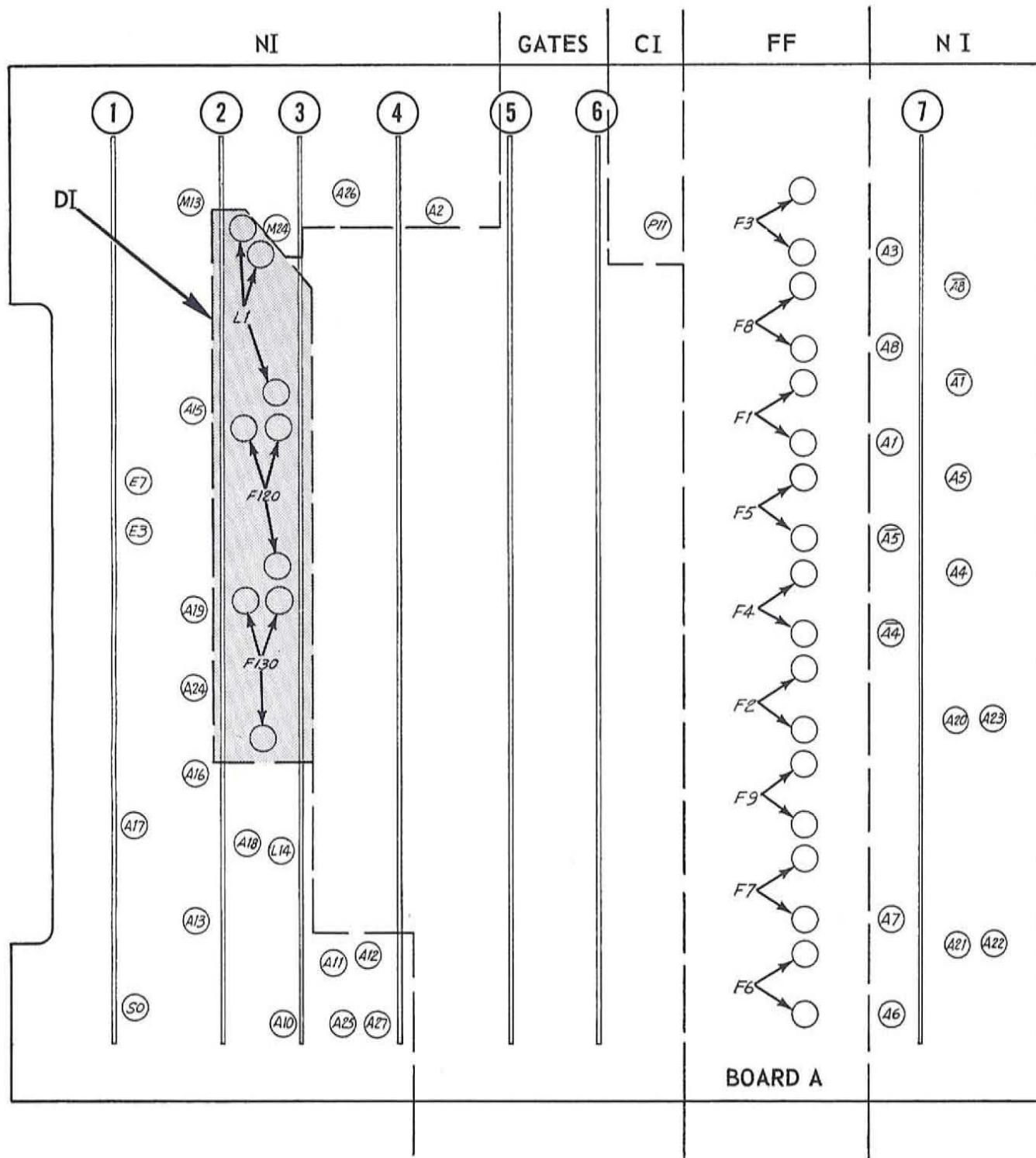
Page VII-3 shows the major assemblies and indicates the page of the detailed breakdown of each.

Logic running sheets start on Page VII-17 and are listed alphabetically.

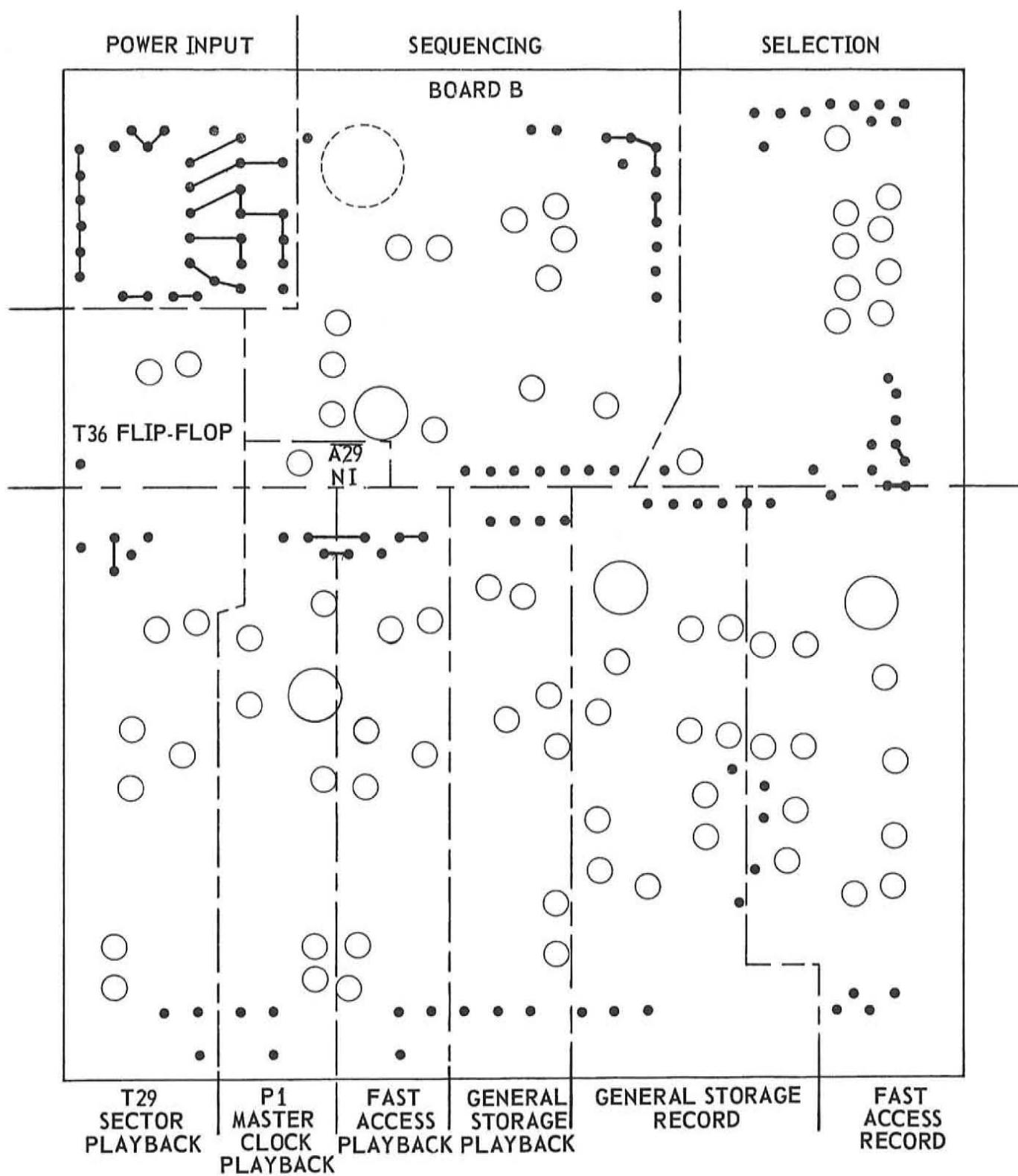
Power running sheets start on Page VII-32 and are listed -6, -8, -24, -24R, +18.



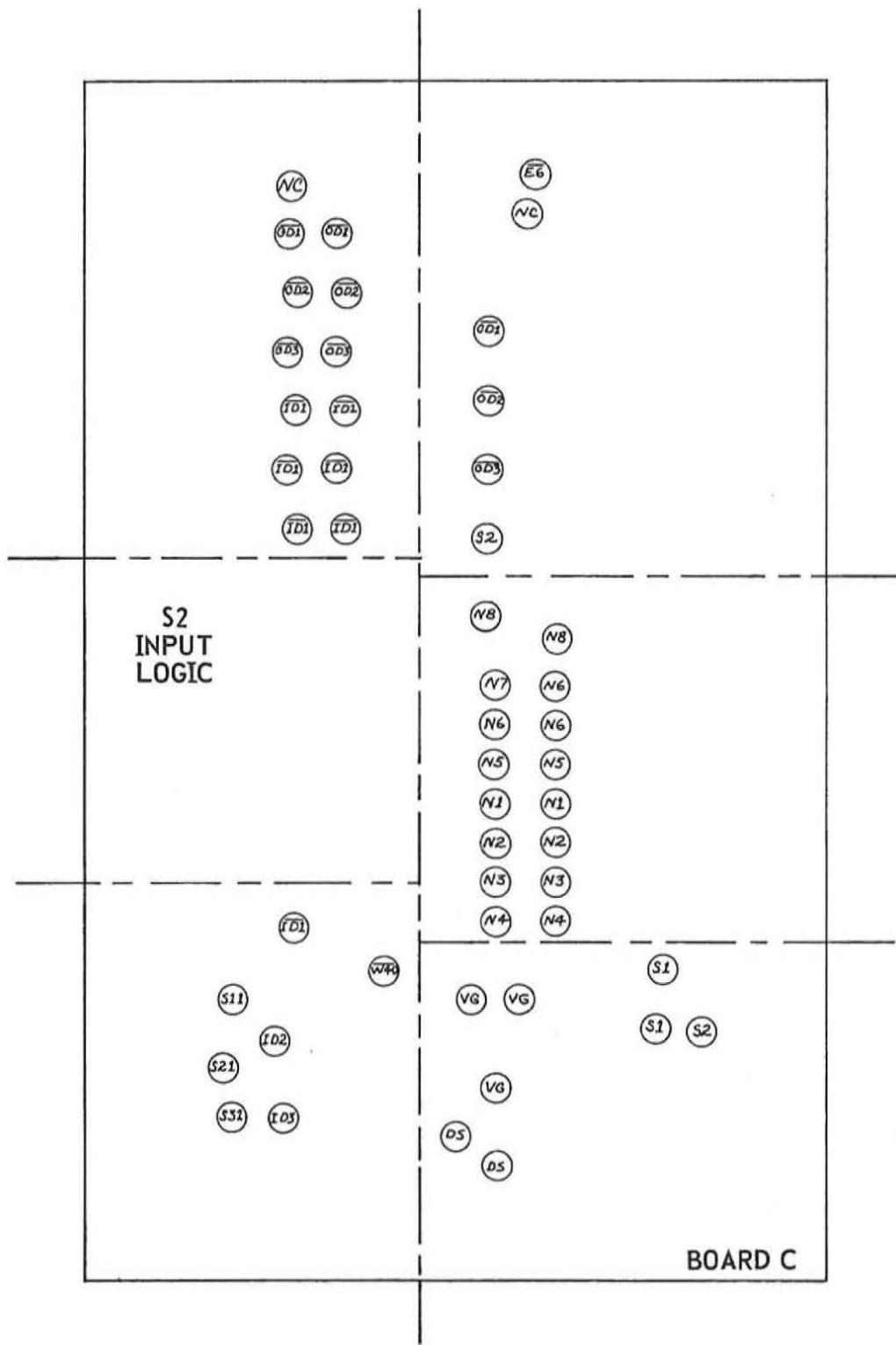
ARITHMETIC & CONTROL BOARD



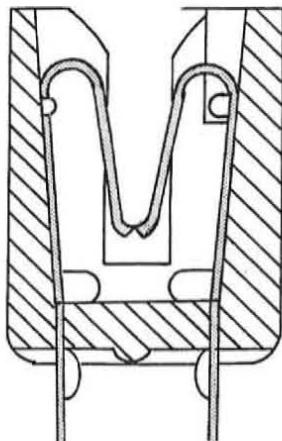
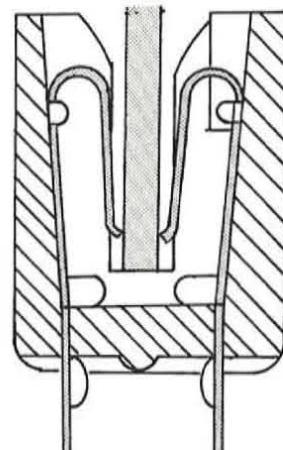
MEMORY BOARD

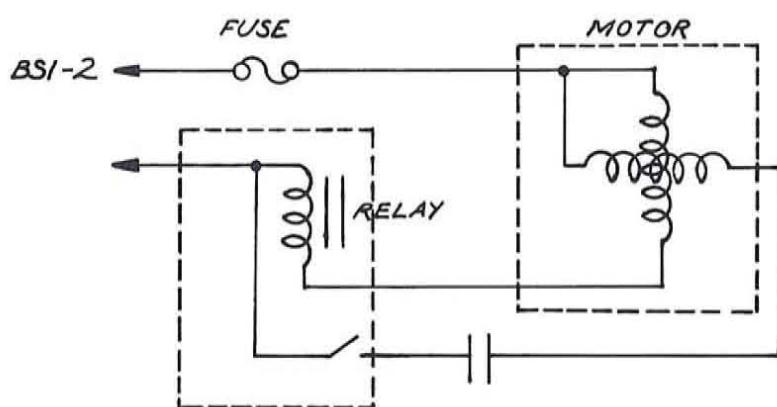
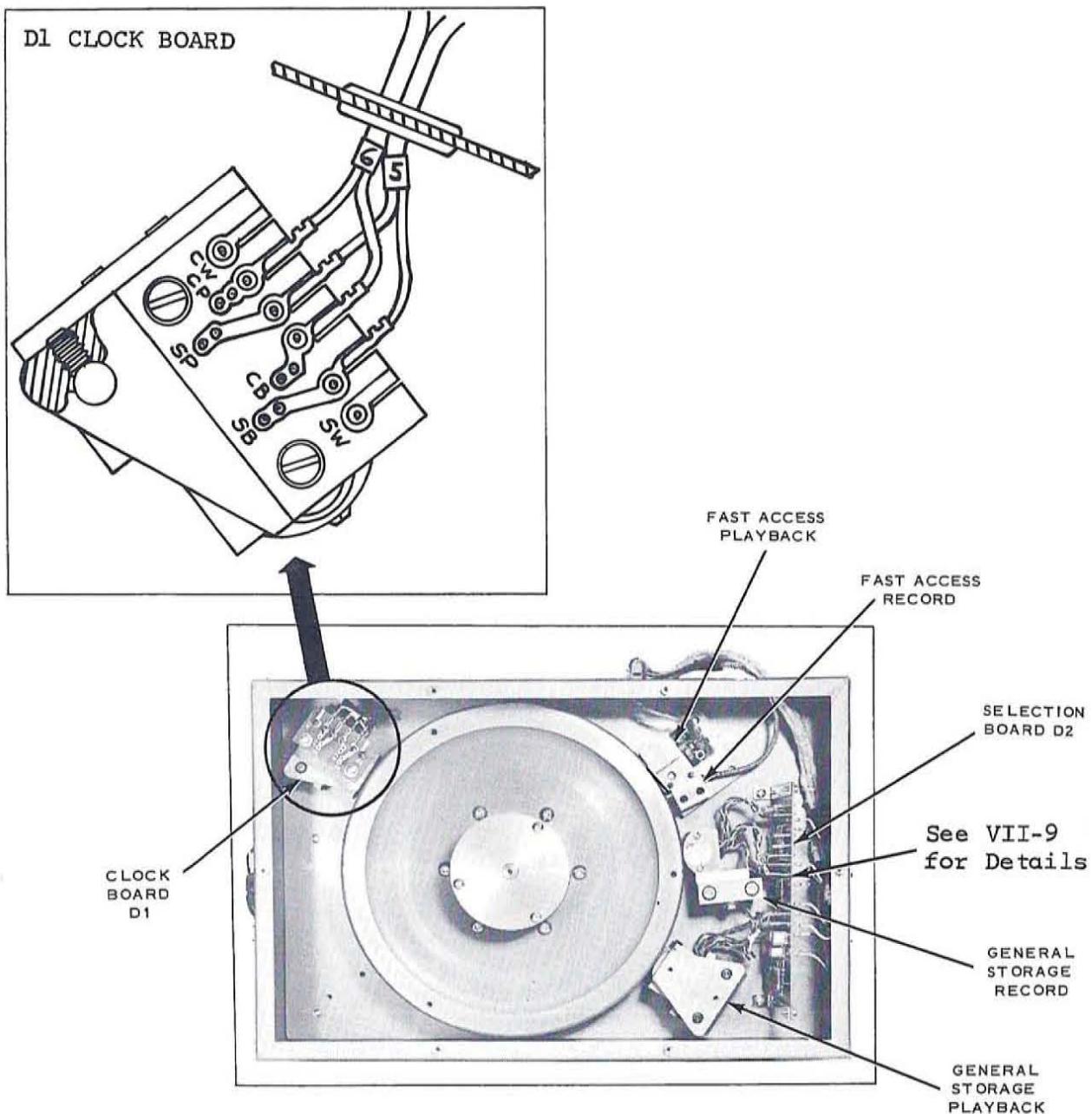


INPUT-OUTPUT BOARD

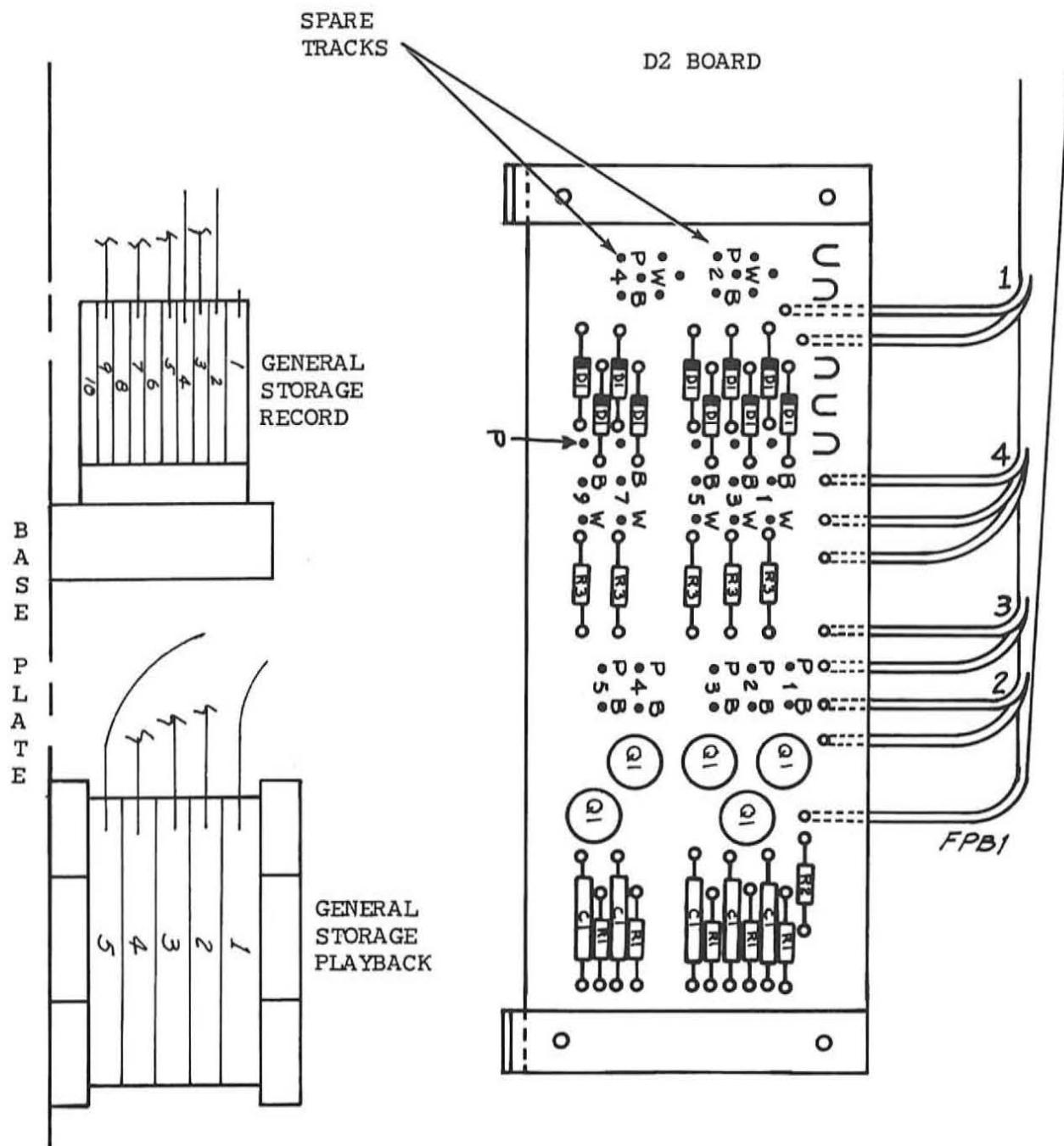


| PIN # | CONNECTOR | |
|----------|------------|------------|
| | "J1-A" | "J1-B" |
| 1 | M1 | FPB1 |
| 2 | M2 | A28 |
| 3 | <u>L1</u> | F2 |
| 4 | L1 | - |
| 5 | L2 | - |
| 6 | <u>L2</u> | L7 |
| 7 | L8 | L5 |
| 8 | L8 | A4 |
| 9 | <u>P1</u> | A8 |
| 10 | T29 | C40 |
| 11 | <u>A29</u> | <u>K14</u> |
| 12 | T36 | K14 |
| 13 | <u>T36</u> | T29 |
| 14 | GND | <u>A29</u> |
| 15 | +18 | -24 |
| 16 | - 8 | - 6 |
| 17 | P.GND | FR.GND |
| 18 | -24B | |

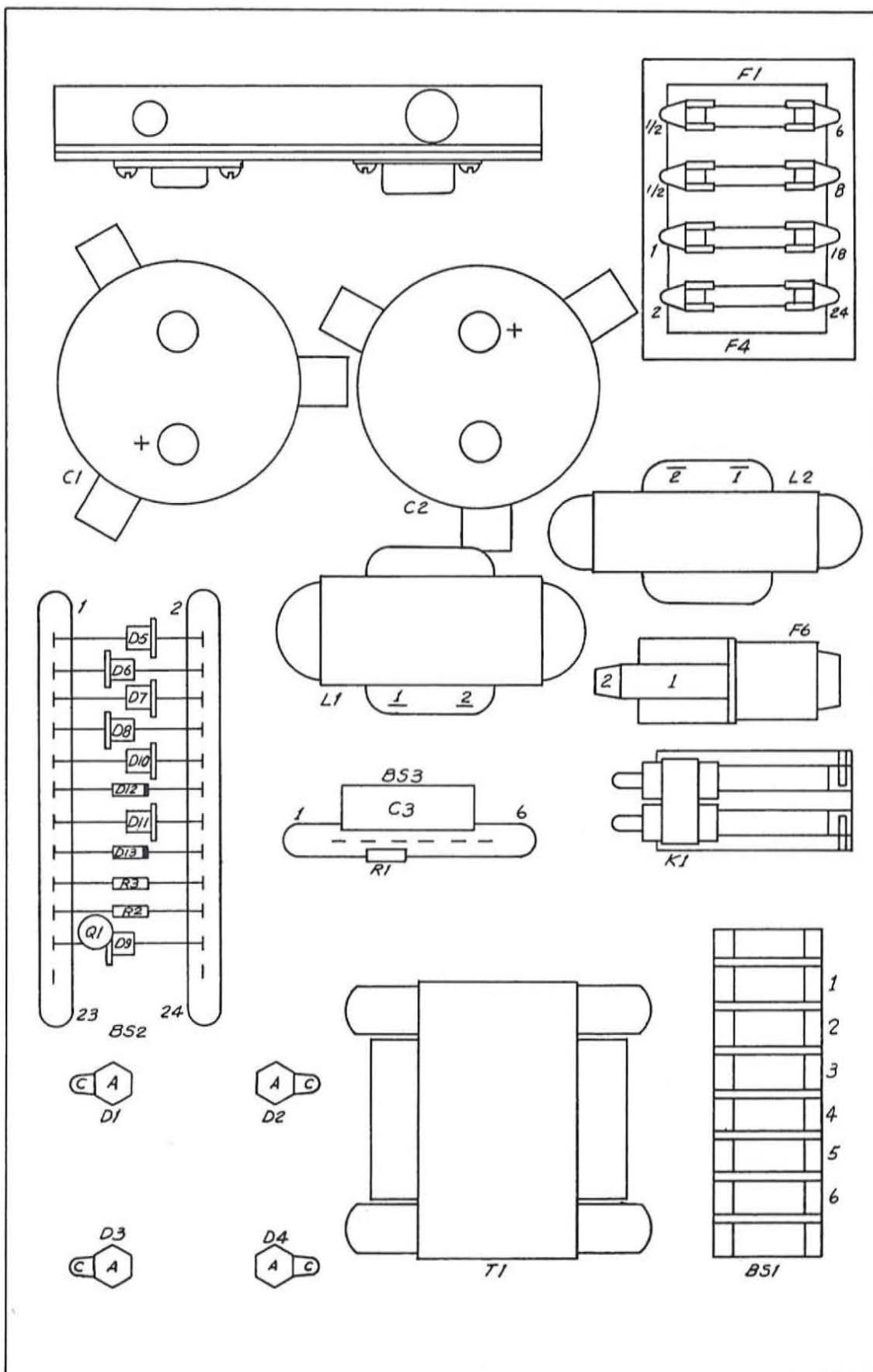
Loader Connector
Without BoardLoader Connector
With Board



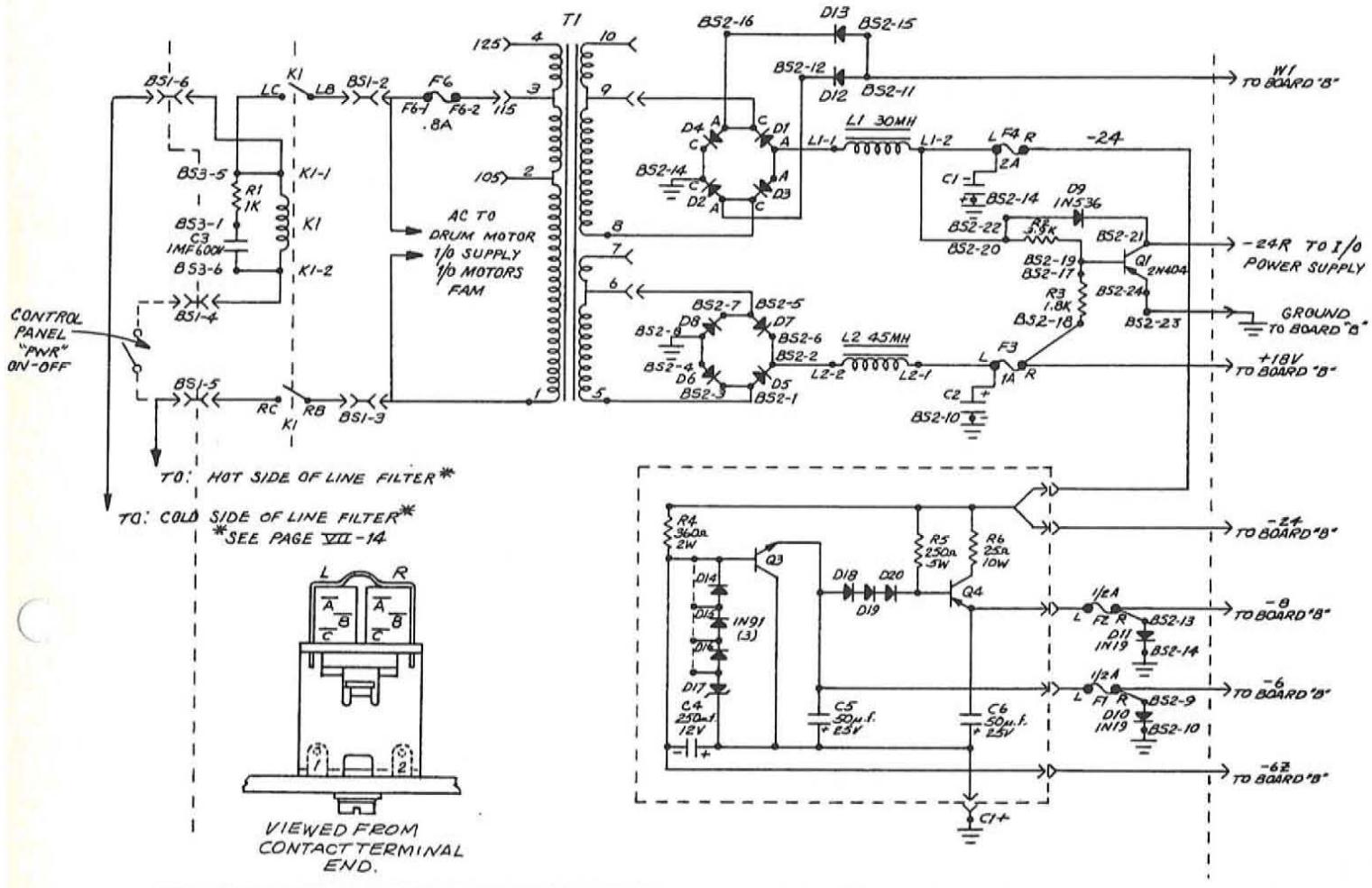
MOTOR CIRCUIT SCHEMATIC



MAIN POWER SUPPLY COMPONENT LOCATION

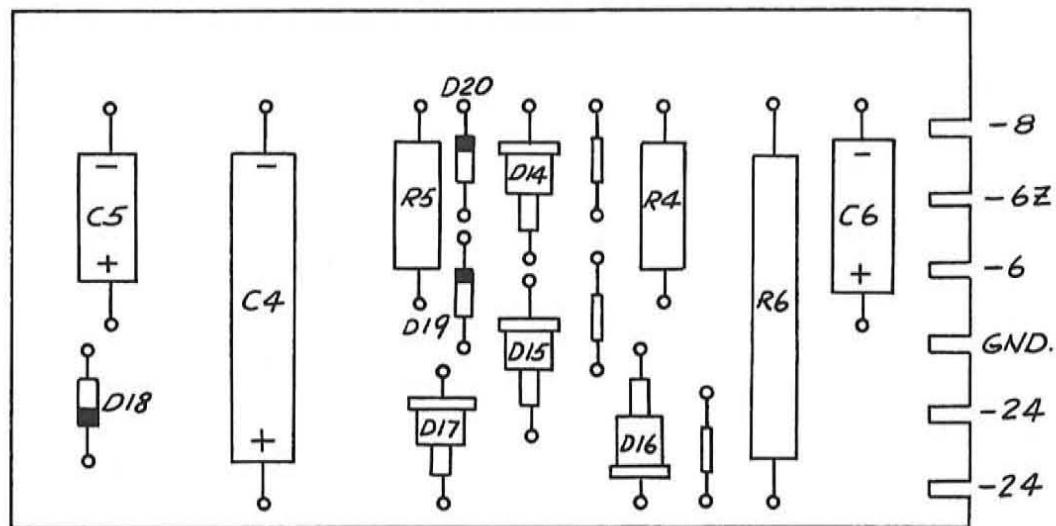


MAIN POWER SUPPLY SCHEMATIC



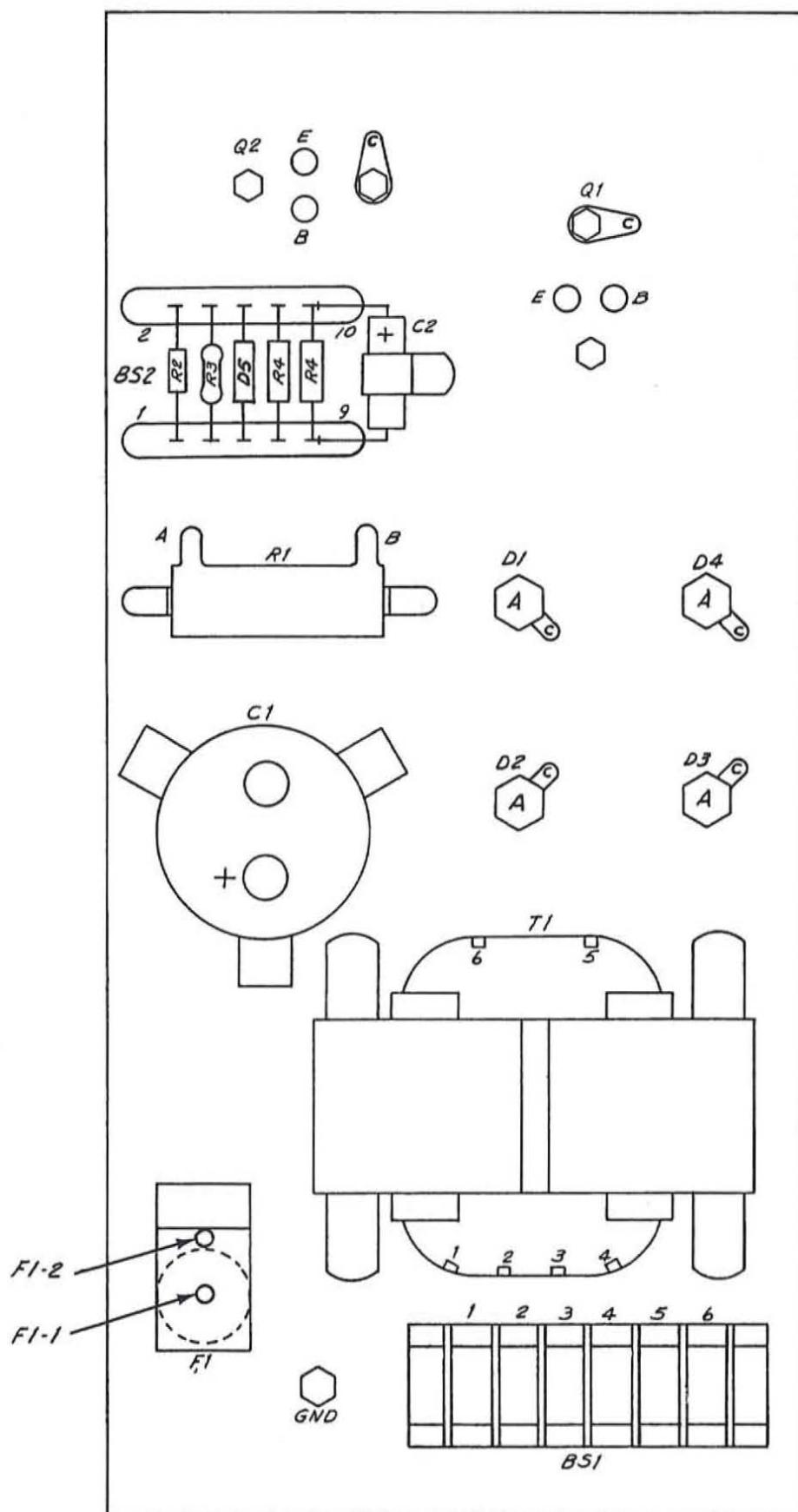
Q3

Q4

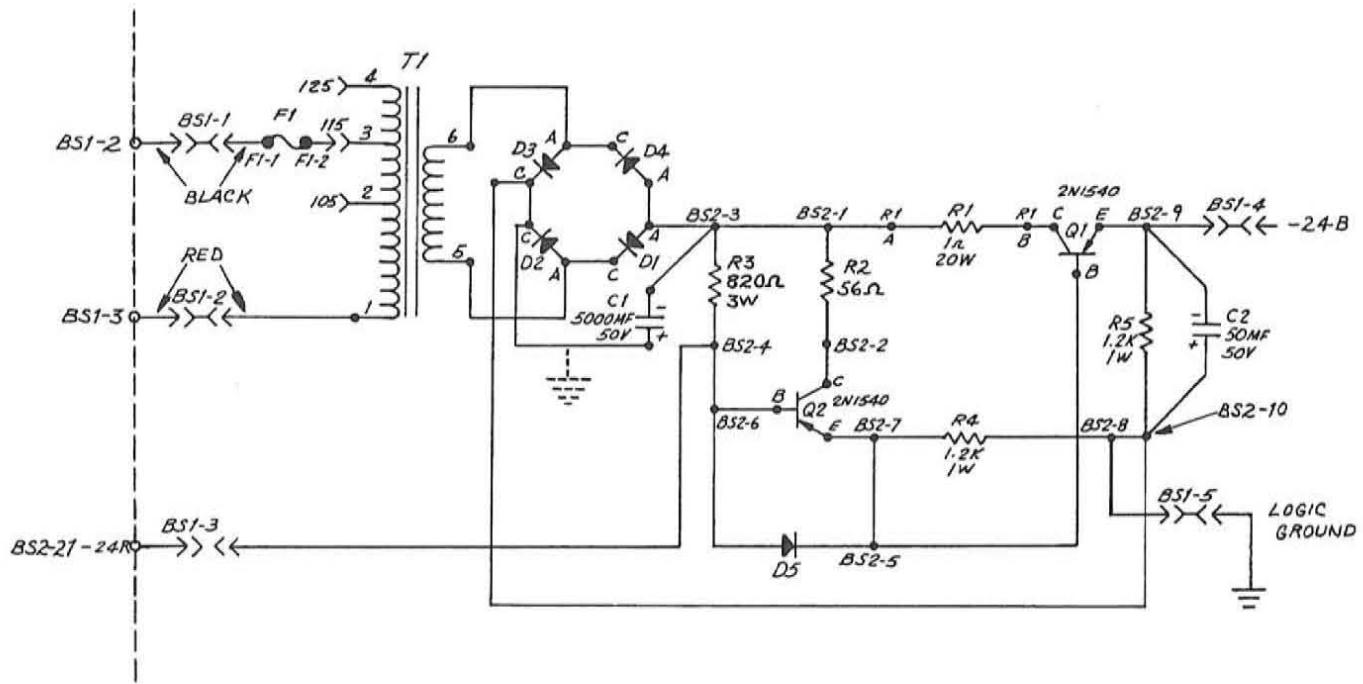


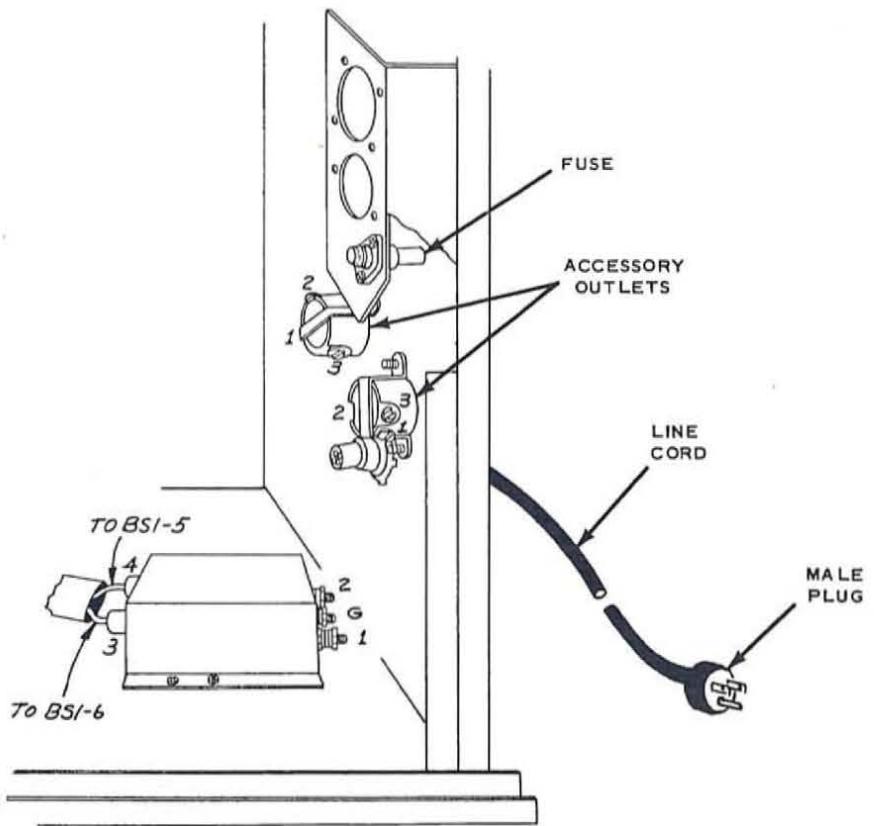
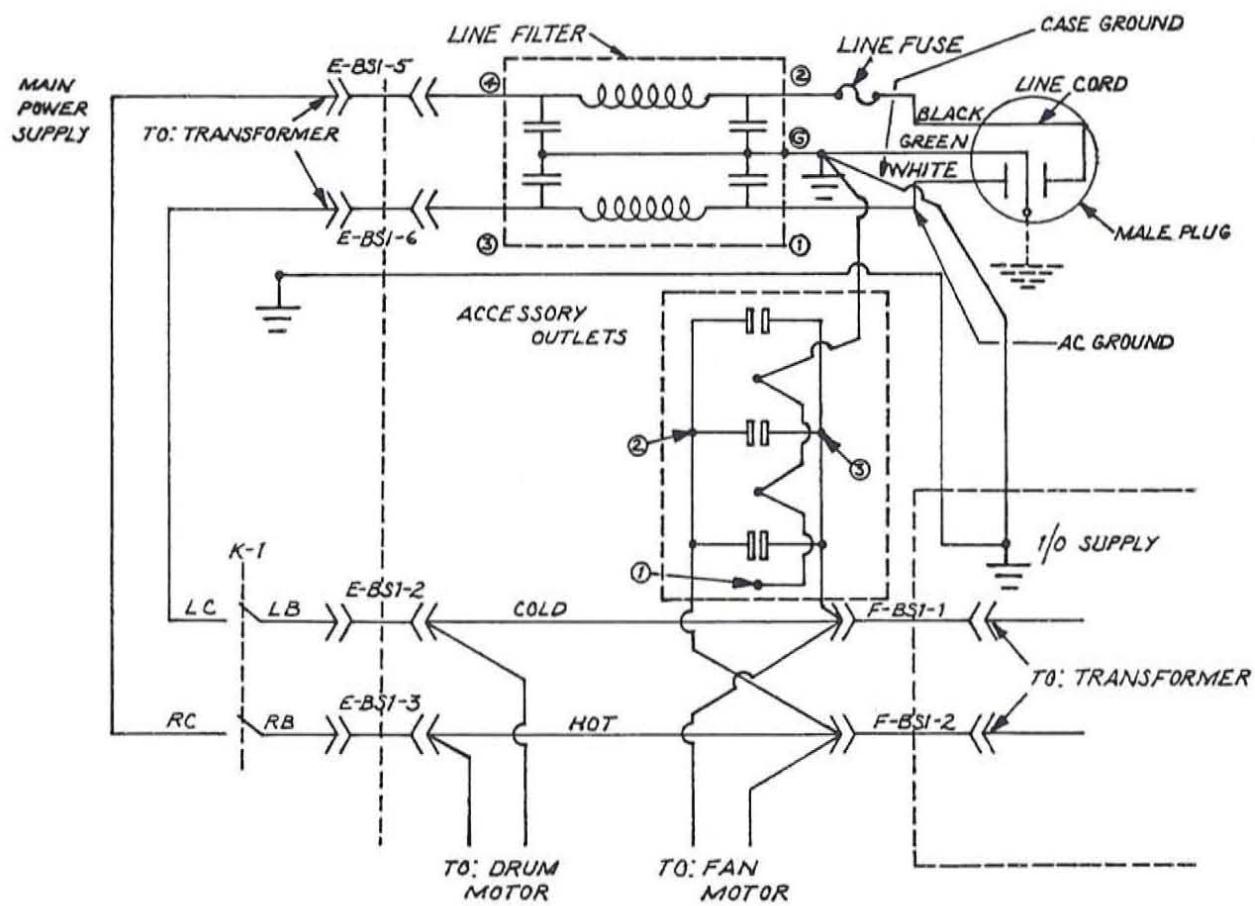
REGULATOR BOARD

INPUT-OUTPUT POWER SUPPLY COMPONENT LOCATION

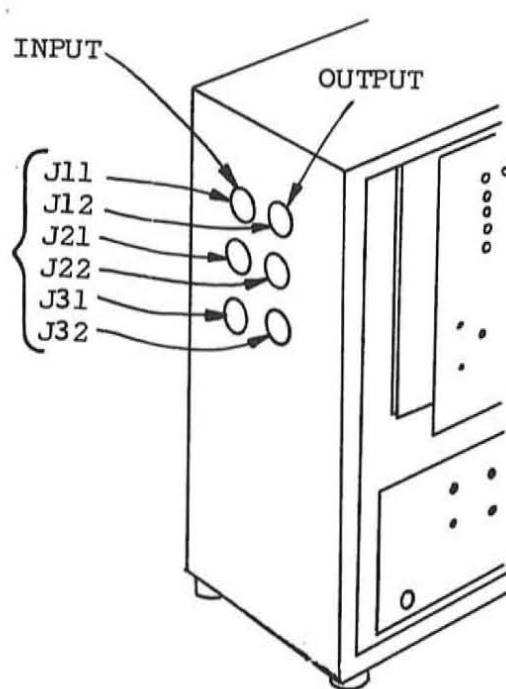


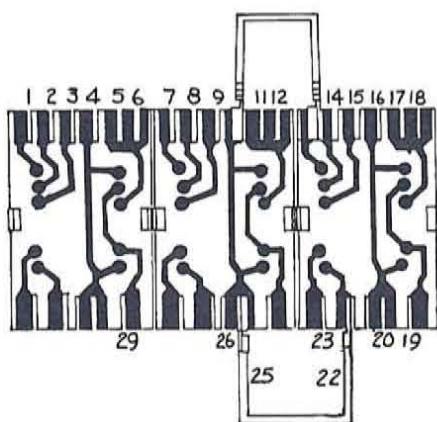
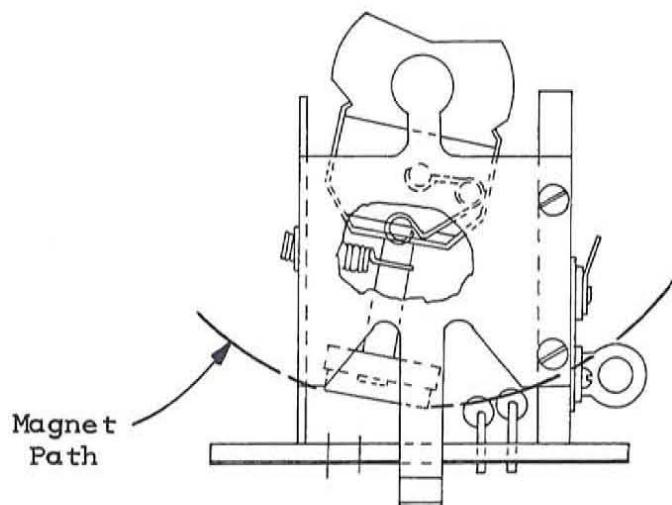
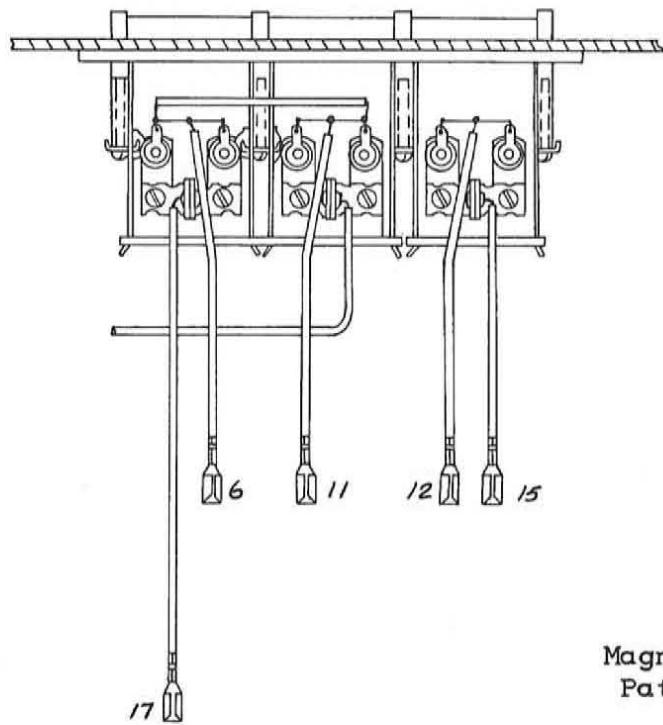
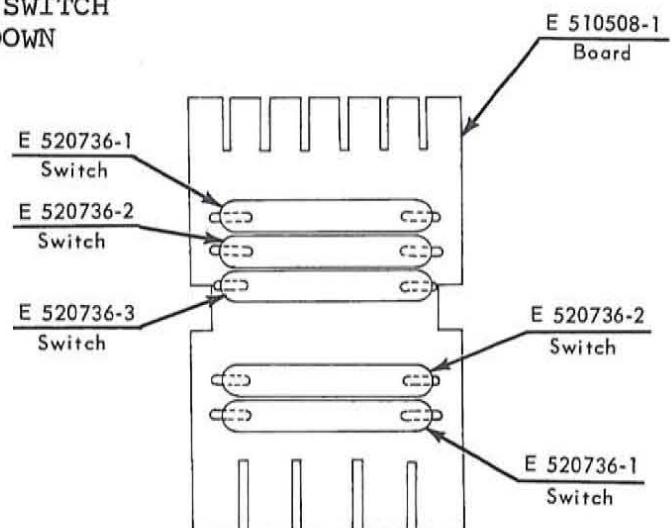
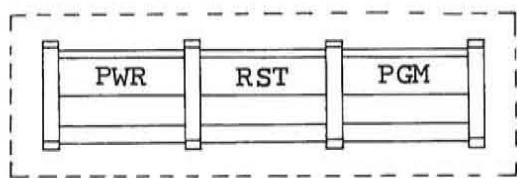
INPUT-OUTPUT POWER SUPPLY SCHEMATIC



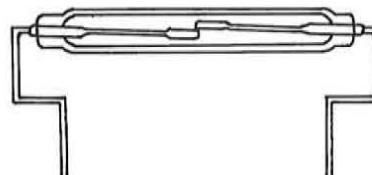


| Pin # | INPUT CHANNEL | | | OUTPUT CHANNEL | | |
|----------|---------------|------------|------------|----------------|----------|----------|
| | J11 1 | J21 2 | J31 3 | J12 1 | J22 2 | J32 3 |
| 1 | X11 | X21 | X31 | N1 | N1 | N1 |
| 2 | X12 | X22 | X32 | N2 | N2 | N2 |
| 3 | X13 | X23 | X33 | N3 | N3 | N3 |
| 4 | X14 | X24 | X34 | N4 | N4 | N4 |
| 5 | X15 | X25 | X35 | N5 | N5 | N5 |
| 6 | X16 | X26 | X36 | N6 | N6 | N6 |
| 7 | X17 | X27 | X37 | N7 | N7 | N7 |
| 8 | X18 | X28 | X38 | N8 | N8 | N8 |
| 9 | | | | | | |
| 10 | | | | | | |
| 11 | | | | | | |
| 12 | | | | | | |
| 13 | ID1 | ID2 | ID3 | NC | NC | NC |
| 14 | | | | | | |
| 15 | | | | | | |
| 16 | GND | GND | GND | GND | GND | GND |
| 17 | | | | | | |
| 18 | | | | | | |
| 19 | | | | | | |
| 20 | Clear | S11 W40 | S21 W40 | S31 W40 | | |
| 21 | | | | OD1 | OD2 | OD3 |
| 22 | | | | | | |
| 23 | S14 | S24 | S34 | E16 | E26 | E36 |
| 24 | +18 | +18 | +18 | +18 | +18 | +18 |
| 25 | B.GND | B.GND | B.GND | B.GND | B.GND | B.GND |
| 26 | -24B | -24B | -24B | -24B | -24B | -24B |
| 27 | FR.G. | FR.G. | FR.G. | FR.G. | FR.G. | FR.G. |



CONTROL SWITCH
BREAKDOWN

REED SWITCH



LOCATION OF ASSEMBLIES & RUNNING SHEETS

VII-17

LOCATION OF ASSEMBLIES & RUNNING SHEETS

VII-19

LOCATION OF ASSEMBLIES & RUNNING SHEETS

VII-21

LOCATION OF ASSEMBLIES & RUNNING SHEETS

VII-23

LOCATION OF ASSEMBLIES & RUNNING SHEETS

VII-25

| SIGNAL | SEWING | PRINTED CIRCUIT | INTERNAL WIRING | PRINTED CIRCUIT | SEWING | GATE | LOGIC PAGE | LOGIC PKG. |
|--------|--------|-----------------|-----------------|-----------------|--------|---------------|------------|------------|
| E6 | C2-TR | | | | | | 11-81 | |
| | C-MT | | | | | | | |
| | | A-BL | | | | | | |
| | | | A1-BR | | | V61 | 11-67 | F6 |
| E7 | A1-TR | | | | | | 11-74 | |
| | A-TL | | | | | | | |
| | | C-MR | | | | | | |
| | | | C2-MR | | | N408 | 11-78 | N8 |
| | | | C2-TR | | | OUTPUT COMMON | 11-78 | NC |
| F1 | A7-TL | | | | | | 11-63 | |
| | A3-BL | | | | | N103 | 11-71 | A10 |
| | A1-MR | | | | | N162 | 11-70 | A16 |
| | A1-MR | | | | | N161 | 11-70 | A16 |
| F1 | AT-TL | | | | | | 11-63 | |
| | A5-TL | | | | | N305 | 11-75 | L1 |
| | A2-BR | | | | | N102 | 11-70 | A10 |
| | A1-BR | | | | | N163 | 11-71 | A16 |
| | A1-BR | | | | | N164 | 11-71 | A16 |

LOCATION OF ASSEMBLIES & RUNNING SHEETS

VII-29

| SIGNAL | SEWING | PRINTED CIRCUIT | INTERNAL WIRING | PRINTED CIRCUIT | SEWING | GATE | LOGIC PAGE | LOGIC PKG. |
|-----------------|---|-----------------|-----------------|-----------------|--------|------|------------|------------|
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| $\overline{F7}$ | A7-BL | | | | | | 11-68 | |
| |  | A7-BR | | | | | | |
| | | | | | | | | |
| | A5-ML | | | | | V21 | 11-64 | E2 |
| | A4-BR | | | | | V251 | 11-73 | A25 |
| | A4-ML | | | | | V25 | 11-64 | F2 |
| | A3-BR | | | | | V111 | 11-70 | A11 |
| | A2-BR | | | | | N101 | 11-70 | A10 |
| | A1-TR | | | | | N152 | 11-71 | A15 |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| $F8$ | A7-TL | | | | | | 11-68 | |
| |  | A7-TR | | | | | | |
| | | | | | | | | |
| | A3-BL | | | | | D181 | 11-72 | A18 |
| | A1-BR | | | | | D171 | 11-72 | A17 |
| | A1-TR | | | | | N153 | 11-71 | A15 |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| $\overline{F8}$ | A7-TL | | | | | | 11-68 | |
| |  | A7-TR | | | | | | |
| | | | | | | | | |
| | A5-ML | | | | | N305 | 11-75 | L1 |
| | A5-BL | | | | | D251 | 11-73 | A25 |
| | A1-MR | | | | | D241 | 11-72 | A24 |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| $F9$ | A7-BL | | | | | | 11-69 | |
| |  | A7-BR | | | | | | |
| | | | | | | | | |
| | A5-BR | | | | | N94 | 11-69 | F9 |
| |  | A5-BR | | | | | | |
| | | | | | | | | |
| | A5-TL | | | | | N95 | 11-69 | F9 |
| | A4-TL | | | | | N306 | 11-75 | L1 |
| | A2-BL | | | | | V22 | 11-64 | F2 |

| SIGNAL | SEWING | PRINTED CIRCUIT | INTERNAL WIRING | PRINTED CIRCUIT | SEWING | GATE | LOGIC PAGE | LOGIC PKG. |
|-------------------|---|-----------------|-----------------|-----------------|--------|------|------------|------------|
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| $\overline{F9}$ | A7-BL | | | | | | 11-69 | |
| |  | A7-BR | | | | N-92 | 11-69 | F9 |
| |  | A5-MR | | | | D171 | 11-72 | A17 |
| | | | | | | | | |
| | | | | | | | | |
| * | JUMPER | | | | | | | |
| | A1-BR | | | | | | | |
| | | A-ML | | | | | | |
| | | | C-MR | | | | | |
| | | | | C2-MR | | N408 | 11-78 | N8 |
| | | | | | | | | |
| | | | | | | | | |
| F120 | A3-TL | | | | | | 11-64 | |
| | A3-TL | | | | | | | |
| |  | A3-TR | | | | | | |
| |  | A6-MR | | | | | 11-64 | F2 |
| | | | | | | | | |
| | | | | | | | | |
| $\overline{F120}$ | A3-TL | | | | | | 11-64 | |
| | A3-TL | | | | | | | |
| |  | A3-TR | | | | | | |
| |  | A6-MR | | | | | 11-64 | F2 |
| | | | | | | | | |
| | | | | | | | | |
| F130 | A3-ML | | | | | | 11-65 | |
| | | | | | | | | |
| |  | A3-MR | | | | | | |
| |  | A6-TR | | | | | 11-65 | F3 |
| | | | | | | | | |
| | | | | | | | | |

*CONNECT F9 FOR EVEN PARITY AND $\overline{F9}$ FOR ODD PARITY.

LOCATION OF ASSEMBLIES & RUNNING SHEETS

VII-33

LOCATION OF ASSEMBLIES & RUNNING SHEETS

VII-35

| SIGNAL | SEWING | PRINTED CIRCUIT | INTERNAL WIRING | PRINTED CIRCUIT | SEWING | GATE | LOGIC PAGE | LOGIC PKG. |
|--------|--------|-----------------|-----------------|-----------------|--------|-----------|------------|------------|
| L8 | B-ML | | | | | G.S. P.B. | 11-82 | |
| | | J1A-8L | | | | | | |
| | | J1A-8R | | | | | | |
| | | | IN A-TM | | | | | |
| | | | A5-ML | | | V21 | 11-64 | F2 |
| | | | | A4-ML | | V23 | 11-64 | F2 |
| | | | | A4-BR | | V35 | 11-65 | F3 |
| L8 | B-ML | | | | | G.S. P.B. | 11-82 | |
| | | J1A-7L | | | | | | |
| | | J1A-7R | | | | | | |
| | | | IN A-TM | | | | | |
| | | | A5-MR | | | N41 | 11-66 | F4 |
| M1 | B-TR | | | | | TR.SEL. | 11-76 | TR. SEL. |
| | | J1A-1L | | | | | | |
| | | J1A-1R | | | | | | |
| | | | IN A-TL | | | | | |
| | | | A1-TR | | | V333 | 11-76 | M13 |
| | | | | A6-ML | | V91 | 11-69 | F9 |
| | | | | A6-BL | | V95 | 11-69 | F9 |
| M2 | B-TR | | | | | | 11-76 | |
| | | J1A-2L | | | | | | |
| | | J1A-2R | | | | | | |
| | | | IN A-TL | | | | | |
| | | | A2-TR | | | V334 | 11-76 | M24 |

LOCATION OF ASSEMBLIES & RUNNING SHEETS

VII-37

LOCATION OF ASSEMBLIES & RUNNING SHEETS

VII-39

| SIGNAL | SEWING | PRINTED CIRCUIT | INTERNAL WIRING | PRINTED CIRCUIT | SEWING | GATE | LOGIC PAGE | LOGIC PKG. |
|--------|--------|-----------------|-----------------|-----------------|--------|------|------------|------------|
| N5 | C3-MR | | | | | | 11-77 | |
| | C1-TR | | | | | | 11-81 | <u>ID2</u> |
| | | OUT C-TL | | | | | | |
| | | | J22-5 | | | | | |
| | | | J12-5 | | | | | |
| N6 | C3-MR | | | | | | 11-78 | |
| | C1-MR | | | | | | 11-81 | <u>ID3</u> |
| | | OUT C-TL | | | | | | |
| | | | J22-6 | | | | | |
| | | | J12-6 | | | | | |
| N7 | C2-MR | | | | | | 11-78 | |
| | | OUT C-TL | | | | | | |
| | | | J22-7 | | | | | |
| | | | J12-7 | | | | | |
| N8 | C2-MR | | | | | | 11-78 | |
| | | OUT C-TL | | | | | | |
| | | | J22-8 | | | | | |
| | | | J12-8 | | | | | |
| N.C. | C1-TR | | | | | | 11-78 | |
| | | OUT C-TL | | | | | | |
| | | | J22-13 | | | | | |
| | | | J12-13 | | | | | |

LOCATION OF ASSEMBLIES & RUNNING SHEETS

VII-41

LOCATION OF ASSEMBLIES & RUNNING SHEETS

VII-43

LOCATION OF ASSEMBLIES & RUNNING SHEETS

VII-45

LOCATION OF ASSEMBLIES & RUNNING SHEETS

VII-47

LOCATION OF ASSEMBLIES & RUNNING SHEETS

VII-49

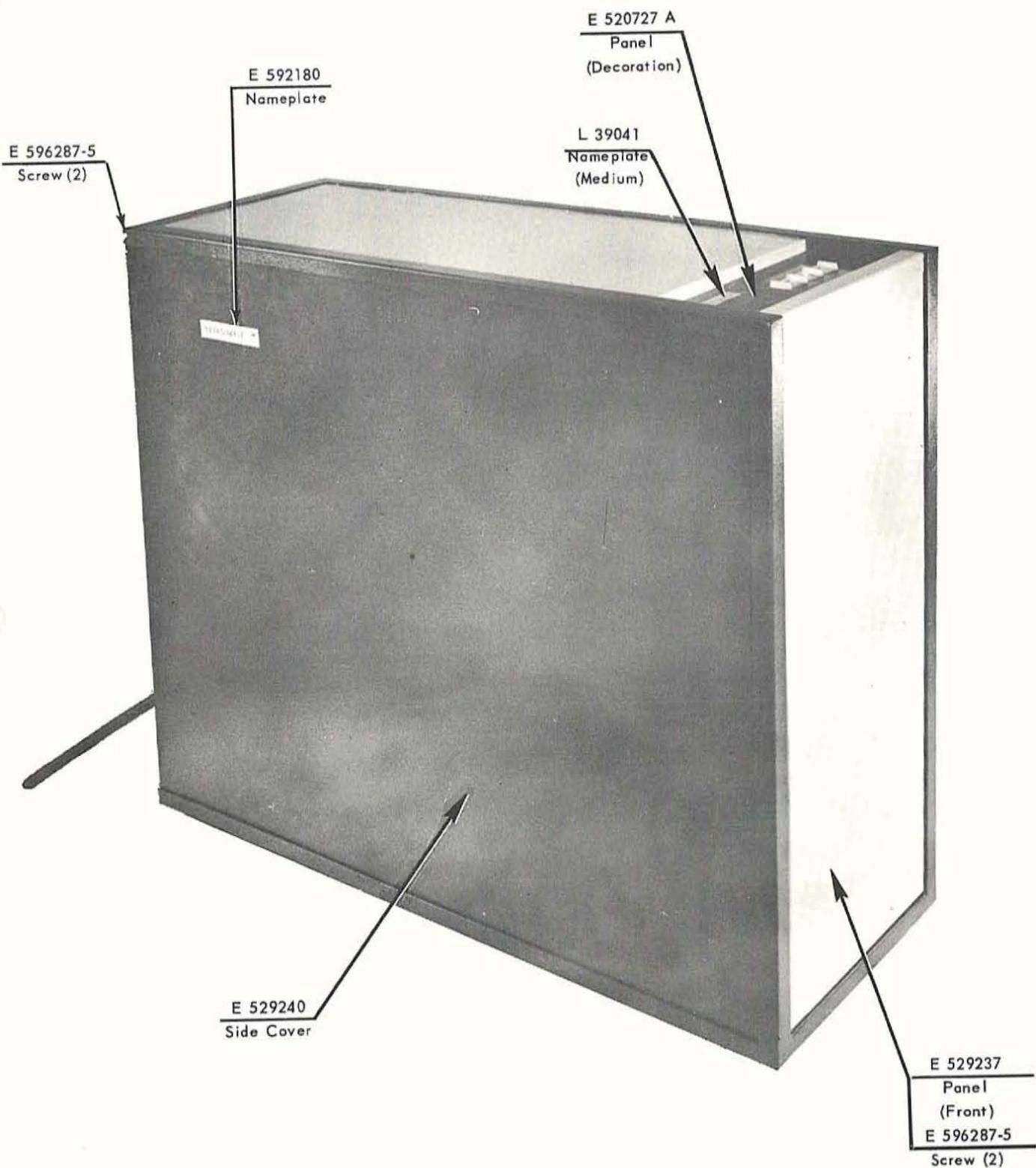
LOCATION OF ASSEMBLIES & RUNNING SHEETS

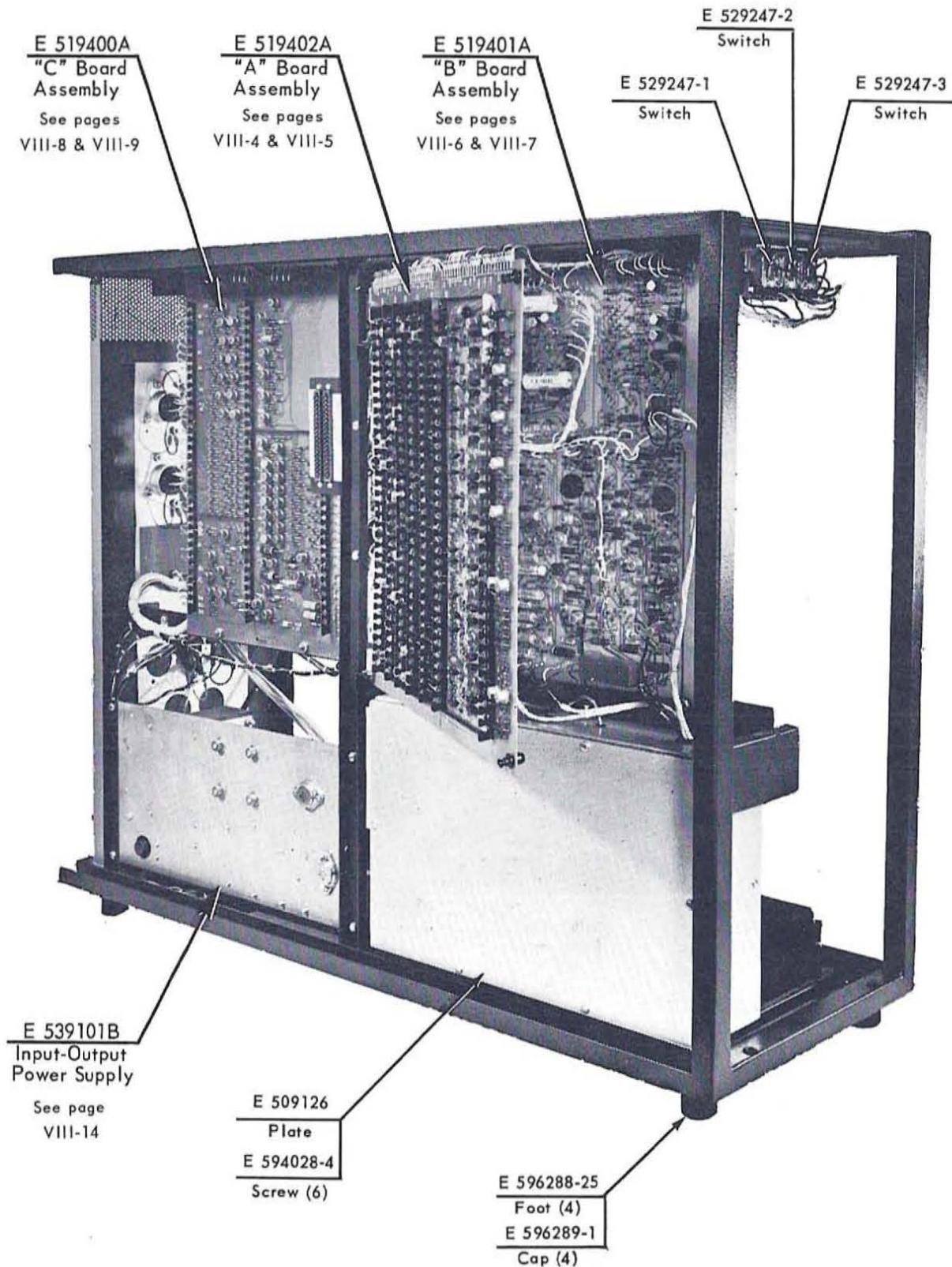
VII-51

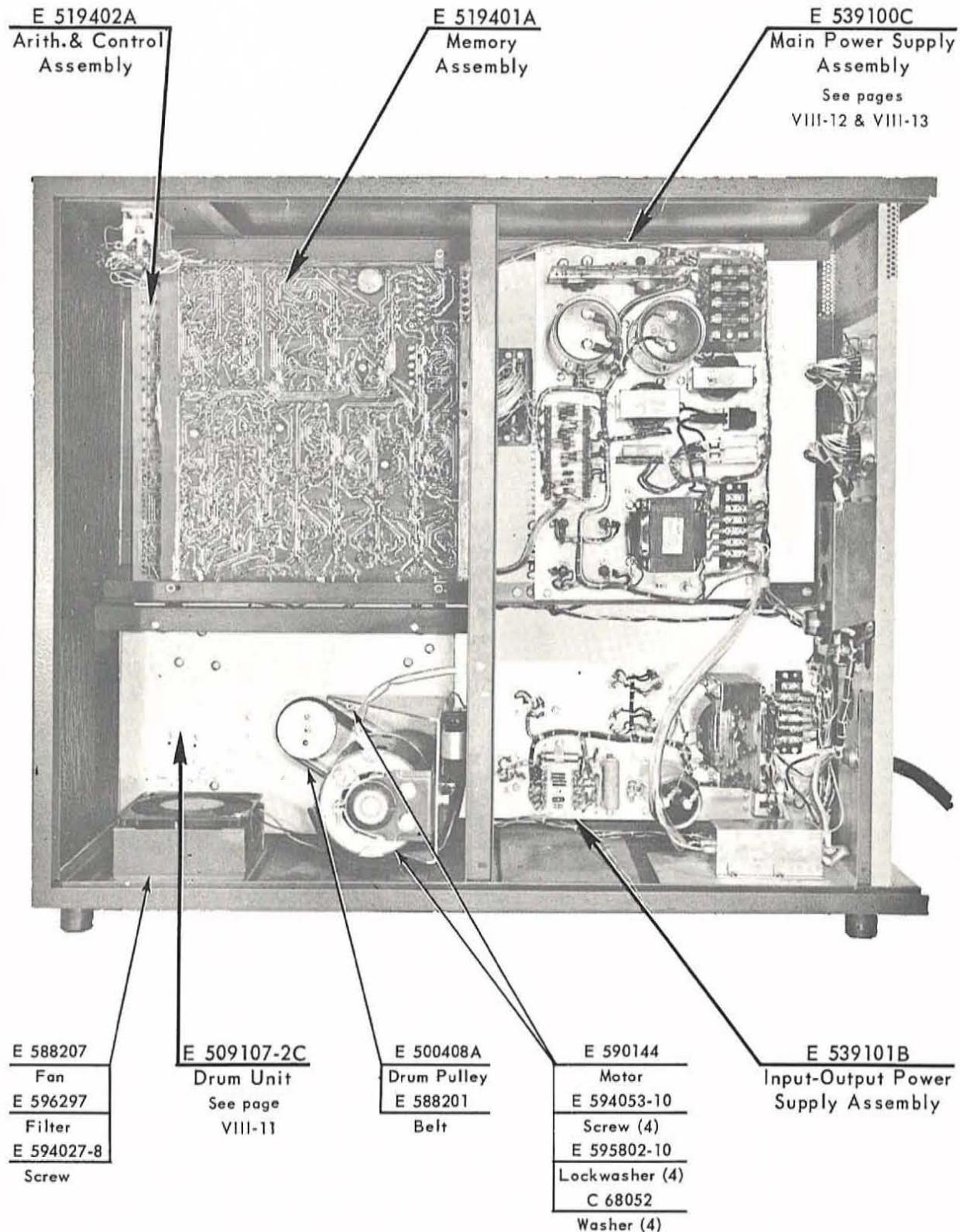
SECTION VIII

PARTS CATALOG

| | Page |
|---------------------------------|------------------|
| Exterior Parts | VIII-1 |
| Major Assemblies | VIII-2, VIII-3 |
| "A" Board Parts Breakdown | VIII-4, VIII-5 |
| "B" Board Parts Breakdown | VIII-6, VIII-7 |
| "C" Board Parts Breakdown | VIII-8, VIII-9 |
| Drum Unit | VIII-11 |
| Main Power Supply | VIII-12, VIII-13 |
| Input/Output Power Supply | VIII-14 |
| Rear Panel | VIII-15 |

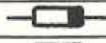


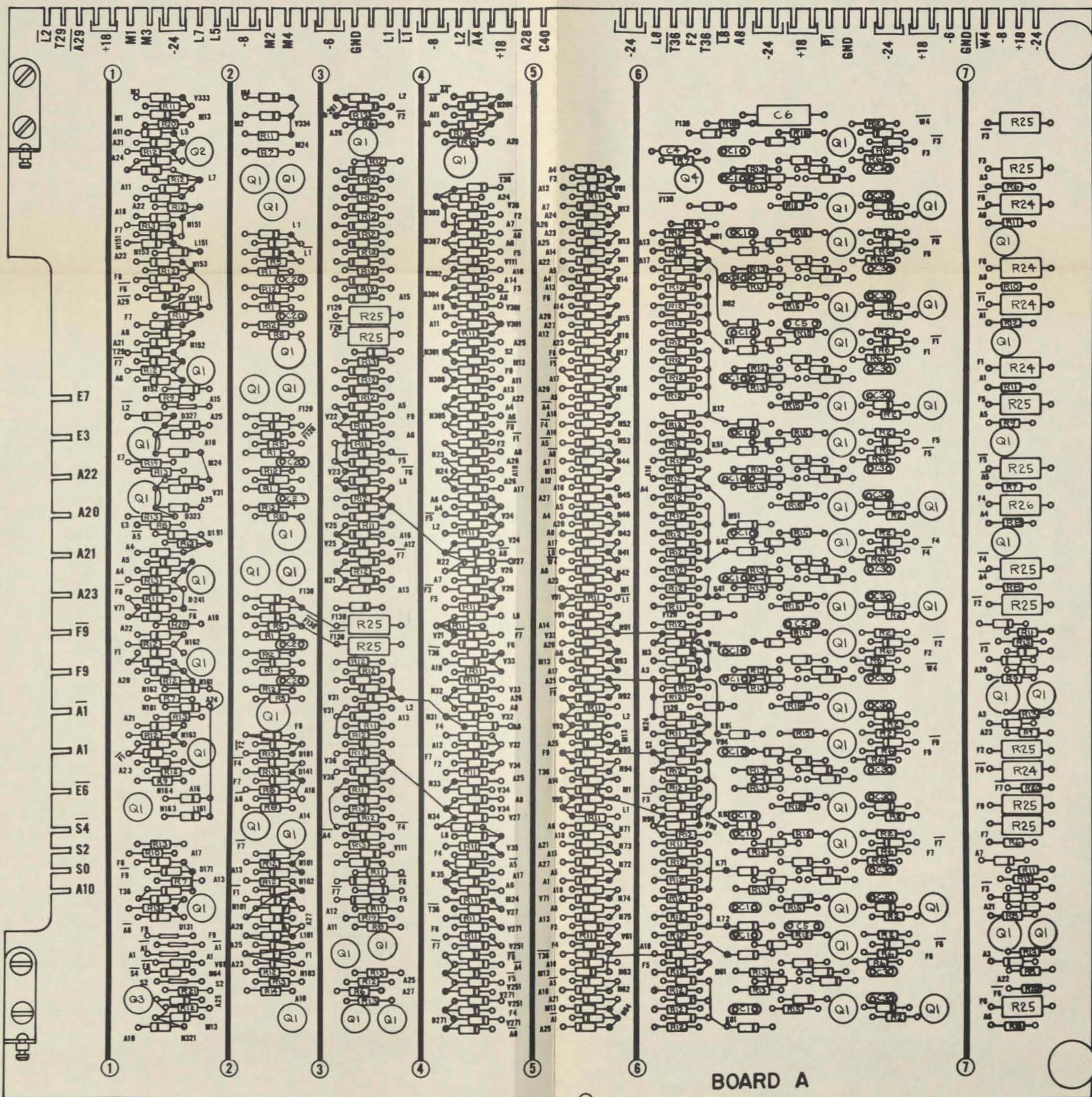




| PARTS LIST | | | |
|------------|--------|------------|------------------------------|
| Symbol | Amount | Part No. | Description |
| | 1 | E 519401E | MEMORY ASSEMBLY |
| C1 | 4 | E 586127 | CAPACITOR, 220PFD-500V-5% |
| C2 | 6 | E 586111 | CAPACITOR, 270PFD-500V-5% |
| C3 | 1 | E 586129 | CAPACITOR, 390PFD-500V-5% |
| C4 | 23 | E 586144 | CAPACITOR, 470PFD-300V-5% |
| C5 | 1 | E 586076 | CAPACITOR, 0.1MFD 50V |
| C7 | 1 | E 586138 | CAPACITOR, .001MFD-200V-5% |
| C8 | 1 | E 586096 | CAPACITOR, .001MFD-200V-10% |
| C9 | 3 | E 586137 | CAPACITOR, .001MFD-200V-20% |
| C10 | 2 | E 586136 | CAPACITOR, .0022MFD-200V-10% |
| C11 | 5 | E 586084 | CAPACITOR, .01MFD-200V-10% |
| C12 | 2 | E 586045 | CAPACITOR, .047MFD-200V-20% |
| C13 | 26 | E 586135 | CAPACITOR, 0.1MFD-200V-20% |
| C15 | 2 | E 586134 | CAPACITOR, 1.0MFD-200V-20% |
| C17 | 2 | E 586133 | CAPACITOR, 1.0MFD-50V |
| C18 | 4 | E 586132 | CAPACITOR, 5MFD-6V |
| C19 | 4 | E 586026 | CAPACITOR, 5MFD-25V |
| C20 | 1 | E 586131 | CAPACITOR, 20MFD-50V |
| C21 | 1 | E 586130 | CAPACITOR, 25MFD-12V |
| C22 | 2 | E 586058 | CAPACITOR, 50MFD-12V |
| C23 | 8 | E 586087 | CAPACITOR, 50MFD-25V |
| Q1 | 5 | E 581013 | TRANSISTOR, PNP Germanium |
| Q2 | 3 | E 581008 | TRANSISTOR, PNP Germanium |
| Q3 | 4 | E 581034 | TRANSISTOR, NPN Silicon |
| Q4 | 5 | E 581037 | TRANSISTOR, PNP Germanium |
| Q5 | 16 | E 581024 | TRANSISTOR, NPN Germanium |
| Q6 | 17 | E 581025 | TRANSISTOR, PNP Germanium |
| Q7 | 15 | E 581005 | TRANSISTOR, PNP Germanium |
| Q8 | 6 | E 581015 | TRANSISTOR, PNP Germanium |
| Q9 | 2 | E 581038 | TRANSISTOR, PNP Silicon |
| Q10 | 1 | E 581045 | TRANSISTOR, PNP Silicon |
| R1 | 1 | E 582010-4 | RESISTOR, 10Ω 1/4W 5% |
| R2 | 3 | E 582022-4 | RESISTOR, 22Ω 1/4W 5% |
| R3 | 1 | E 582047-4 | RESISTOR, 47Ω 1/4W 5% |
| R4 | 8 | E 582110-4 | RESISTOR, 100Ω 1/4W 5% |
| R5 | 2 | E 582112-4 | RESISTOR, 120Ω 1/4W 5% |
| R6 | 5 | E 582120-4 | RESISTOR, 200Ω 1/4W 5% |
| R7 | 1 | E 582122-4 | RESISTOR, 220Ω 1/4W 5% |
| R8 | 8 | E 582130-4 | RESISTOR, 300Ω 1/4W 5% |
| R9 | 4 | E 582133-4 | RESISTOR, 330Ω 1/4W 5% |
| R10 | 3 | E 582139-4 | RESISTOR, 390Ω 1/4W 5% |
| R11 | 12 | E 582147-4 | RESISTOR, 470Ω 1/4W 5% |
| R12 | 5 | E 582151-4 | RESISTOR, 510Ω 1/4W 5% |
| R13 | 25 | E 582210-4 | RESISTOR, 1.0K 1/4W 5% |
| R14 | 8 | E 582218-4 | RESISTOR, 1.8K 1/4W 5% |
| R15 | 12 | E 582220-4 | RESISTOR, 2.0K 1/4W 5% |
| R16 | 1 | E 582182-4 | RESISTOR, 820Ω 1/4W 5% |
| R17 | 22 | E 582227-4 | RESISTOR, 2.7K 1/4W 5% |
| R18 | 4 | E 582230-4 | RESISTOR, 3.0K 1/4W 5% |
| R19 | 1 | E 582233-4 | RESISTOR, 3.3K 1/4W 5% |

| PARTS LIST | | | |
|------------|---------|------------|------------------------------|
| Symbol | Amount | Part No. | Description |
| R20 | 1 | E 582239-4 | RESISTOR, 3.9K 1/4W 5% |
| R21 | 5 | E 582247-4 | RESISTOR, 4.7K 1/4W 5% |
| R22 | 8 | E 582251-4 | RESISTOR, 5.1K 1/4W 5% |
| R23 | 1 | E 582268-4 | RESISTOR, 6.8K 1/4W 5% |
| R24 | 4 | E 582275-4 | RESISTOR, 7.5K 1/4W 5% |
| R25 | 2 | E 582282-4 | RESISTOR, 8.2K 1/4W 5% |
| R26 | 17 | E 582310-4 | RESISTOR, 10K 1/4W 5% |
| R27 | 3 | E 582315-4 | RESISTOR, 15K 1/4W 5% |
| R28 | 11 | E 582320-4 | RESISTOR, 20K 1/4W 5% |
| R29 | 1 | E 582330-4 | RESISTOR, 30K 1/4W 5% |
| R30 | 6 | E 582333-4 | RESISTOR, 33K 1/4W 5% |
| R31 | 8 | E 582339-4 | RESISTOR, 39K 1/4W 5% |
| R32 | 8 | E 582347-4 | RESISTOR, 47K 1/4W 5% |
| R33 | 2 | E 582368-4 | RESISTOR, 68K 1/4W 5% |
| R35 | 1 | E 582120-0 | RESISTOR, 200Ω 1/2W 5% |
| R36 | 1 | E 582156-0 | RESISTOR, 560Ω 1/2W 5% |
| R37 | 3 | E 582210-0 | RESISTOR, 1.0K 1/2W 5% |
| R38 | 5 | E 582220-0 | RESISTOR, 2.0K 1/2W 5% |
| R39 | 1 | E 582222-0 | RESISTOR, 2.2K 1/2W 5% |
| R40 | 3 | E 582224-0 | RESISTOR, 2.4K 1/2W 5% |
| R41 | 1 | E 582227-0 | RESISTOR, 2.7K 1/2W 5% |
| R44 | 2 | E 584004-1 | RESISTOR, 5.6Ω 1W 5% |
| R45 | 1 | E 582127-1 | RESISTOR, 270Ω 1W 5% |
| R46 | 2 | E 582151-1 | RESISTOR, 510Ω 1W 5% |
| R47 | 1 | E 582168-1 | RESISTOR, 680Ω 1W 5% |
| R48 | 5 | E 582210-1 | RESISTOR, 1.0K 1W 5% |
| R49 | 5 | E 582212-1 | RESISTOR, 1.2K 1W 5% |
| R50 | 4 | E 582215-1 | RESISTOR, 1.5K 1W 5% |
| R51 | 1 | E 582220-1 | RESISTOR, 2.0K 1W 5% |
| R54 | 1 | E 582033-2 | RESISTOR, 33Ω 2W 5% |
| R55 | 1 | E 582110-2 | RESISTOR, 100Ω 2W 5% |
| R58 | 1 | E 584075 | POTENTIOMETER, 2W |
| R59 | 2 | E 584060 | POTENTIOMETER, 1.0K 1/2W 20% |
| R60 | 2 | E 584061 | POTENTIOMETER, 10K 1/2W 20% |
| R61 | 1 | E 590037-1 | VARISTOR |
| | 1 | E 580035 | DIODE, Silicon |
| | 8 | E 580039 | DIODE, Silicon |
| | 8 | E 580013 | DIODE, Germanium |
| | 102 | E 580029 | DIODE, Germanium |
| | 5 | E 592252 | JUMPER |
| | 67 | E 596291 | TRANSIPAD |
| | 17 1/2" | E 592016 | WIRE, #18 Stranded Black |
| | 7 | E 509130 | TRANSFORMER, Record |

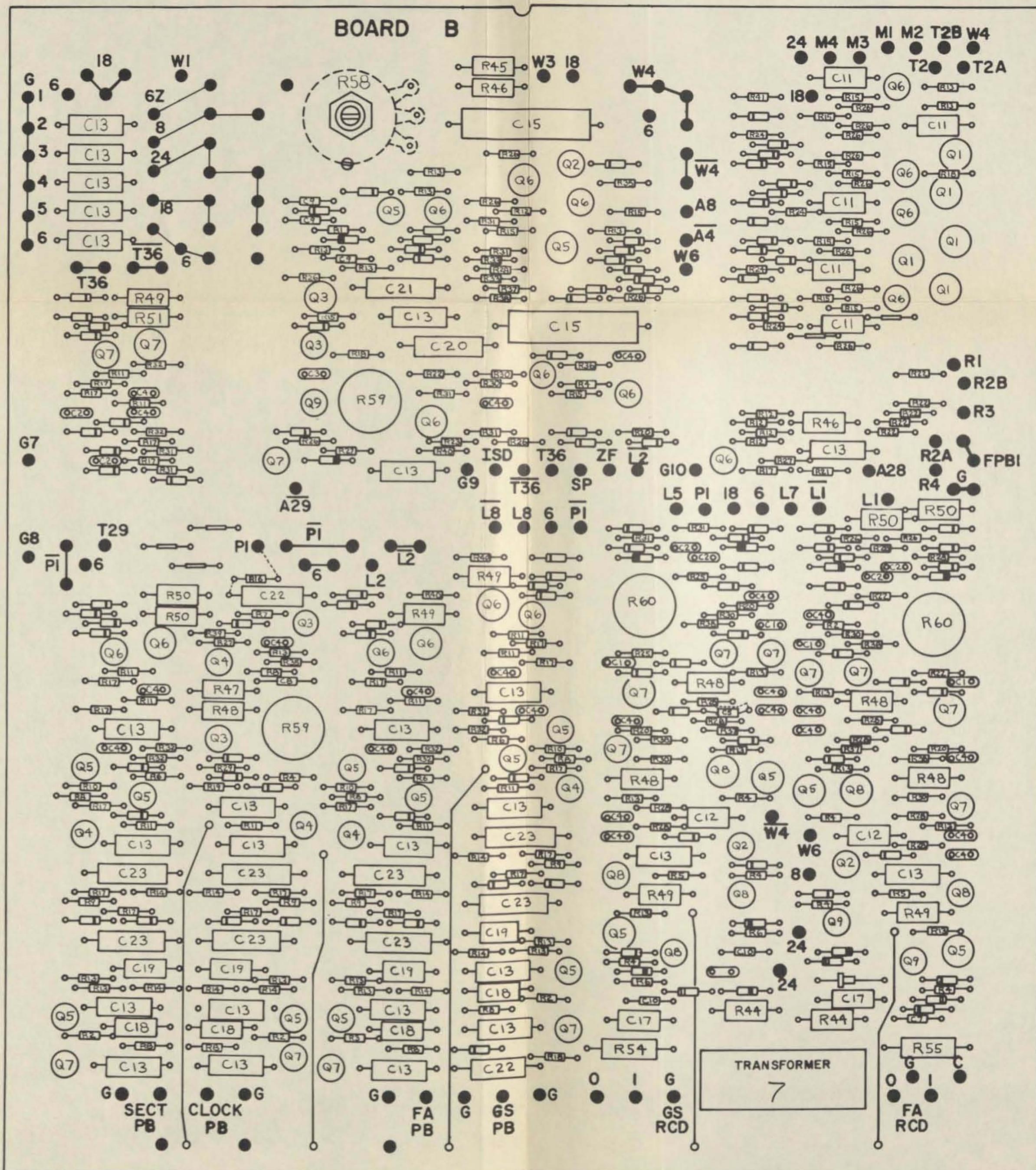
| PARTS LIST | | | |
|---|--------|------------|---|
| Symbol | Amount | Part No. | Description |
| | 1 | E 519400B | MULTIPLE INPUT-OUTPUT ASSEMBLY |
| C1 | 1 | E 586137 | CAPACITOR, .001MFD-200V-20% |
| C2 | 1 | E 586045 | CAPACITOR, .047MFD-200V-20% |
| C3 | 3 | E 586135 | CAPACITOR, 0.1MFD-200V-20% |
| C4 | 1 | E 586092 | CAPACITOR, 10MFD-12V |
| C5 | 9 | E 586144 | CAPACITOR, 470PFD-300V-10% |
| Q1 | 24 | E 581024 | TRANSISTOR, 2N1302 |
| Q2 | 26 | E 581025 | TRANSISTOR, 2N1303 |
| R1 | 1 | E 582091-4 | RESISTOR, 91Ω $\frac{1}{4}W$ 5% |
| R2 | 1 | E 582147-4 | RESISTOR, 470Ω $\frac{1}{4}W$ 5% |
| R3 | 8 | E 582175-4 | RESISTOR, 750Ω $\frac{1}{4}W$ 5% |
| R4 | 1 | E 582210-4 | RESISTOR, 1.0K $\frac{1}{4}W$ 5% |
| R5 | 5 | E 582215-4 | RESISTOR, 1.5K $\frac{1}{4}W$ 5% |
| R6 | 1 | E 582220-4 | RESISTOR, 2.0K $\frac{1}{4}W$ 5% |
| R7 | 12 | E 582233-4 | RESISTOR, 3.3K $\frac{1}{4}W$ 5% |
| R8 | 1 | E 582239-4 | RESISTOR, 3.9K $\frac{1}{4}W$ 5% |
| R9 | 1 | E 582247-4 | RESISTOR, 4.7K $\frac{1}{4}W$ 5% |
| R10 | 12 | E 582251-4 | RESISTOR, 5.1K $\frac{1}{4}W$ 5% |
| R11 | 3 | E 582256-4 | RESISTOR, 5.6K $\frac{1}{4}W$ 5% |
| R12 | 20 | E 582275-4 | RESISTOR, 7.5K $\frac{1}{4}W$ 5% |
| R13 | 6 | E 582291-4 | RESISTOR, 9.1K $\frac{1}{4}W$ 5% |
| R14 | 25 | E 582310-4 | RESISTOR, 10K $\frac{1}{4}W$ 5% |
| R15 | 3 | E 582315-4 | RESISTOR, 15K $\frac{1}{4}W$ 5% |
| R16 | 1 | E 582318-4 | RESISTOR, 18K $\frac{1}{4}W$ 5% |
| R17 | 17 | E 582320-4 | RESISTOR, 20K $\frac{1}{4}W$ 5% |
| R18 | 1 | E 582339-4 | RESISTOR, 39K $\frac{1}{4}W$ 5% |
| R19 | 3 | E 582227-4 | RESISTOR, 2.7K $\frac{1}{4}W$ 5% |
| R20 | 3 | E 582351-4 | RESISTOR, 51K $\frac{1}{4}W$ 5% |
| R21 | 11 | E 582368-4 | RESISTOR, 68K $\frac{1}{4}W$ 5% |
| R22 | 3 | E 582410-4 | RESISTOR, 100K $\frac{1}{4}W$ 5% |
| R23 | 1 | E 582022-4 | RESISTOR, 22Ω $\frac{1}{4}W$ 5% |
| R25 | 1 | E 582056-0 | RESISTOR, 56Ω $\frac{1}{2}W$ 5% |
| R26 | 1 | E 582068-0 | RESISTOR, 68Ω $\frac{1}{2}W$ 5% |
| R27 | 3 | E 582230-0 | RESISTOR, 3.0K $\frac{1}{2}W$ 5% |
| R28 | 1 | E 582233-0 | RESISTOR, 3.3K $\frac{1}{2}W$ 5% |
| R30 | 1 | E 582156-1 | RESISTOR, 560Ω 1W 5% |
| R32 | 1 | E 582139-2 | RESISTOR, 390Ω 2W 5% |
| R33 | 1 | E 582147-2 | RESISTOR, 470Ω 2W 5% |
|  | 138 | E 580029 | DIODE, Germanium |
|  | 1 | E 580005 | RECTIFIER, Silicon |
|  | 1 | E 580039 | DIODE, Silicon |
|  | 10 | E 592252 | JUMPER |
| As Req'd. | | E 592241 | WIRE, Magnet #32 Heavy Soldereze |
| | 50 | E 596291 | TRANSIPAD |



VIII-6

| PARTS LIST | | | |
|------------|--------|------------|------------------------------|
| Symbol | Amount | Part No. | Description |
| | 1 | E 519401E | MEMORY ASSEMBLY |
| C1 | 4 | E 586127 | CAPACITOR, 220PFD-500V-5% |
| C2 | 6 | E 586111 | CAPACITOR, 270PFD-500V-5% |
| C3 | 1 | E 586129 | CAPACITOR, 390PFD-500V-5% |
| C4 | 23 | E 586144 | CAPACITOR, 470PFD-300V-5% |
| C5 | 1 | E 586076 | CAPACITOR, 0.1MFD 50V |
| C7 | 1 | E 586138 | CAPACITOR, .001MFD-200V-5% |
| C8 | 1 | E 586096 | CAPACITOR, .001MFD-200V-10% |
| C9 | 3 | E 586137 | CAPACITOR, .001MFD-200V-20% |
| C10 | 2 | E 586136 | CAPACITOR, .0022MFD-200V-10% |
| C11 | 5 | E 586084 | CAPACITOR, .01MFD-200V-10% |
| C12 | 2 | E 586045 | CAPACITOR, .047MFD-200V-20% |
| C13 | 26 | E 586135 | CAPACITOR, 0.1MFD-200V-20% |
| C15 | 2 | E 586134 | CAPACITOR, 1.0MFD-200V-20% |
| C17 | 2 | E 586133 | CAPACITOR, 1.0MFD-50V |
| C18 | 4 | E 586132 | CAPACITOR, 5MFD-6V |
| C19 | 4 | E 586026 | CAPACITOR, 5MFD-25V |
| C20 | 1 | E 586131 | CAPACITOR, 20MFD-50V |
| C21 | 1 | E 586130 | CAPACITOR, 25MFD-12V |
| C22 | 2 | E 586058 | CAPACITOR, 50MFD-12V |
| C23 | 8 | E 586087 | CAPACITOR, 50MFD-25V |
| Q1 | 5 | E 581013 | TRANSISTOR, PNP Germanium |
| Q2 | 3 | E 581008 | TRANSISTOR, PNP Germanium |
| Q3 | 4 | E 581034 | TRANSISTOR, NPN Silicon |
| Q4 | 5 | E 581037 | TRANSISTOR, PNP Germanium |
| Q5 | 16 | E 581024 | TRANSISTOR, NPN Germanium |
| Q6 | 17 | E 581025 | TRANSISTOR, PNP Germanium |
| Q7 | 15 | E 581005 | TRANSISTOR, PNP Germanium |
| Q8 | 6 | E 581015 | TRANSISTOR, PNP Germanium |
| Q9 | 2 | E 581038 | TRANSISTOR, PNP Silicon |
| Q10 | 1 | E 581045 | TRANSISTOR, PNP Silicon |
| R1 | 1 | E 582010-4 | RESISTOR, 10Ω 1/4W 5% |
| R2 | 3 | E 582022-4 | RESISTOR, 22Ω 1/4W 5% |
| R3 | 1 | E 582047-4 | RESISTOR, 47Ω 1/4W 5% |
| R4 | 8 | E 582110-4 | RESISTOR, 100Ω 1/4W 5% |
| R5 | 2 | E 582112-4 | RESISTOR, 120Ω 1/4W 5% |
| R6 | 5 | E 582120-4 | RESISTOR, 200Ω 1/4W 5% |
| R7 | 1 | E 582122-4 | RESISTOR, 220Ω 1/4W 5% |
| R8 | 8 | E 582130-4 | RESISTOR, 300Ω 1/4W 5% |
| R9 | 4 | E 582133-4 | RESISTOR, 330Ω 1/4W 5% |
| R10 | 3 | E 582139-4 | RESISTOR, 390Ω 1/4W 5% |
| R11 | 12 | E 582147-4 | RESISTOR, 470Ω 1/4W 5% |
| R12 | 5 | E 582151-4 | RESISTOR, 510Ω 1/4W 5% |
| R13 | 25 | E 582210-4 | RESISTOR, 1.0K 1/4W 5% |
| R14 | 8 | E 582218-4 | RESISTOR, 1.8K 1/4W 5% |
| R15 | 12 | E 582220-4 | RESISTOR, 2.0K 1/4W 5% |
| R16 | 1 | E 582182-4 | RESISTOR, 820Ω 1/4W 5% |
| R17 | 22 | E 582227-4 | RESISTOR, 2.7K 1/4W 5% |
| R18 | 4 | E 582230-4 | RESISTOR, 3.0K 1/4W 5% |
| R19 | 1 | E 582233-4 | RESISTOR, 3.3K 1/4W 5% |

| PARTS LIST | | | |
|------------|--------|------------|------------------------------|
| Symbol | Amount | Part No. | Description |
| R20 | 1 | E 582239-4 | RESISTOR, 3.9K 1/4W 5% |
| R21 | 5 | E 582247-4 | RESISTOR, 4.7K 1/4W 5% |
| R22 | 8 | E 582251-4 | RESISTOR, 5.1K 1/4W 5% |
| R23 | 1 | E 582268-4 | RESISTOR, 6.8K 1/4W 5% |
| R24 | 4 | E 582275-4 | RESISTOR, 7.5K 1/4W 5% |
| R25 | 2 | E 582282-4 | RESISTOR, 8.2K 1/4W 5% |
| R26 | 17 | E 582310-4 | RESISTOR, 10K 1/4W 5% |
| R27 | 3 | E 582315-4 | RESISTOR, 15K 1/4W 5% |
| R28 | 11 | E 582320-4 | RESISTOR, 20K 1/4W 5% |
| R29 | 1 | E 582330-4 | RESISTOR, 30K 1/4W 5% |
| R30 | 6 | E 582333-4 | RESISTOR, 33K 1/4W 5% |
| R31 | 8 | E 582339-4 | RESISTOR, 39K 1/4W 5% |
| R32 | 8 | E 582347-4 | RESISTOR, 47K 1/4W 5% |
| R33 | 2 | E 582368-4 | RESISTOR, 68K 1/4W 5% |
| R35 | 1 | E 582120-0 | RESISTOR, 200Ω 1/2W 5% |
| R36 | 1 | E 582156-0 | RESISTOR, 560Ω 1/2W 5% |
| R37 | 3 | E 582210-0 | RESISTOR, 1.0K 1/2W 5% |
| R38 | 5 | E 582220-0 | RESISTOR, 2.0K 1/2W 5% |
| R39 | 1 | E 582222-0 | RESISTOR, 2.2K 1/2W 5% |
| R40 | 3 | E 582224-0 | RESISTOR, 2.4K 1/2W 5% |
| R41 | 1 | E 582227-0 | RESISTOR, 2.7K 1/2W 5% |
| R44 | 2 | E 584004-1 | RESISTOR, 5.6Ω 1W 5% |
| R45 | 1 | E 582127-1 | RESISTOR, 270Ω 1W 5% |
| R46 | 2 | E 582151-1 | RESISTOR, 510Ω 1W 5% |
| R47 | 1 | E 582168-1 | RESISTOR, 680Ω 1W 5% |
| R48 | 5 | E 582210-1 | RESISTOR, 1.0K 1W 5% |
| R49 | 5 | E 582212-1 | RESISTOR, 1.2K 1W 5% |
| R50 | 4 | E 582215-1 | RESISTOR, 1.5K 1W 5% |
| R51 | 1 | E 582220-1 | RESISTOR, 2.0K 1W 5% |
| R54 | 1 | E 582033-2 | RESISTOR, 33Ω 2W 5% |
| R55 | 1 | E 582110-2 | RESISTOR, 100Ω 2W 5% |
| R58 | 1 | E 584075 | POTENTIOMETER, 2W |
| R59 | 2 | E 584060 | POTENTIOMETER, 1.0K 1/2W 20% |
| R60 | 2 | E 584061 | POTENTIOMETER, 10K 1/2W 20% |
| R61 | 1 | E 590037-1 | VARISTOR |
| | 1 | E 580035 | DIODE, Silicon |
| | 8 | E 580039 | DIODE, Silicon |
| | 8 | E 580013 | DIODE, Germanium |
| | 102 | E 580029 | DIODE, Germanium |
| | 5 | E 592252 | JUMPER |
| | 67 | E 596291 | TRANSIPAD |
| | 17½" | E 592016 | WIRE, #18 Stranded Black |
| 7 | 1 | E 509130 | TRANSFORMER, Record |



VIII-8

| PARTS LIST | | | |
|------------|--------|------------|----------------------------------|
| Symbol | Amount | Part No. | Description |
| | 1 | E 519400B | MULTIPLE INPUT-OUTPUT ASSEMBLY |
| C1 | 1 | E 586137 | CAPACITOR, .001MFD-200V-20% |
| C2 | 1 | E 586045 | CAPACITOR, .047MFD-200V-20% |
| C3 | 3 | E 586135 | CAPACITOR, 0.1MFD-200V-20% |
| C4 | 1 | E 586092 | CAPACITOR, 10MFD-12V |
| C5 | 9 | E 586144 | CAPACITOR, 470PFD-300V-10% |
| Q1 | 24 | E 581024 | TRANSISTOR, 2N1302 |
| Q2 | 26 | E 581025 | TRANSISTOR, 2N1303 |
| R1 | 1 | E 582091-4 | RESISTOR, 91Ω 1/4W 5% |
| R2 | 1 | E 582147-4 | RESISTOR, 470Ω 1/4W 5% |
| R3 | 8 | E 582175-4 | RESISTOR, 750Ω 1/4W 5% |
| R4 | 1 | E 582210-4 | RESISTOR, 1.0K 1/4W 5% |
| R5 | 5 | E 582215-4 | RESISTOR, 1.5K 1/4W 5% |
| R6 | 1 | E 582220-4 | RESISTOR, 2.0K 1/4W 5% |
| R7 | 12 | E 582233-4 | RESISTOR, 3.3K 1/4W 5% |
| R8 | 1 | E 582239-4 | RESISTOR, 3.9K 1/4W 5% |
| R9 | 1 | E 582247-4 | RESISTOR, 4.7K 1/4W 5% |
| R10 | 12 | E 582251-4 | RESISTOR, 5.1K 1/4W 5% |
| R11 | 3 | E 582256-4 | RESISTOR, 5.6K 1/4W 5% |
| R12 | 20 | E 582275-4 | RESISTOR, 7.5K 1/4W 5% |
| R13 | 6 | E 582291-4 | RESISTOR, 9.1K 1/4W 5% |
| R14 | 25 | E 582310-4 | RESISTOR, 10K 1/4W 5% |
| R15 | 3 | E 582315-4 | RESISTOR, 15K 1/4W 5% |
| R16 | 1 | E 582318-4 | RESISTOR, 18K 1/4W 5% |
| R17 | 17 | E 582320-4 | RESISTOR, 20K 1/4W 5% |
| R18 | 1 | E 582339-4 | RESISTOR, 39K 1/4W 5% |
| R19 | 3 | E 582227-4 | RESISTOR, 2.7K 1/4W 5% |
| R20 | 3 | E 582351-4 | RESISTOR, 51K 1/4W 5% |
| R21 | 11 | E 582368-4 | RESISTOR, 68K 1/4W 5% |
| R22 | 3 | E 582410-4 | RESISTOR, 100K 1/4W 5% |
| R23 | 1 | E 582022-4 | RESISTOR, 22Ω 1/4W 5% |
| R25 | 1 | E 582056-0 | RESISTOR, 56Ω 1/2W 5% |
| R26 | 1 | E 582068-0 | RESISTOR, 68Ω 1/2W 5% |
| R27 | 3 | E 582230-0 | RESISTOR, 3.0K 1/2W 5% |
| R28 | 1 | E 582233-0 | RESISTOR, 3.3K 1/2W 5% |
| R30 | 1 | E 582156-1 | RESISTOR, 560Ω 1W 5% |
| R32 | 1 | E 582139-2 | RESISTOR, 390Ω 2W 5% |
| R33 | 1 | E 582147-2 | RESISTOR, 470Ω 2W 5% |
| — | 138 | E 580029 | DIODE, Germanium |
| — | 1 | E 580005 | RECTIFIER, Silicon |
| — | 1 | E 580039 | DIODE, Silicon |
| — | 10 | E 592252 | JUMPER |
| As Reg'd. | | E 592241 | WIRE, Magnet #32 Heavy Solderize |
| | 50 | E 596291 | TRANSIPAD |

