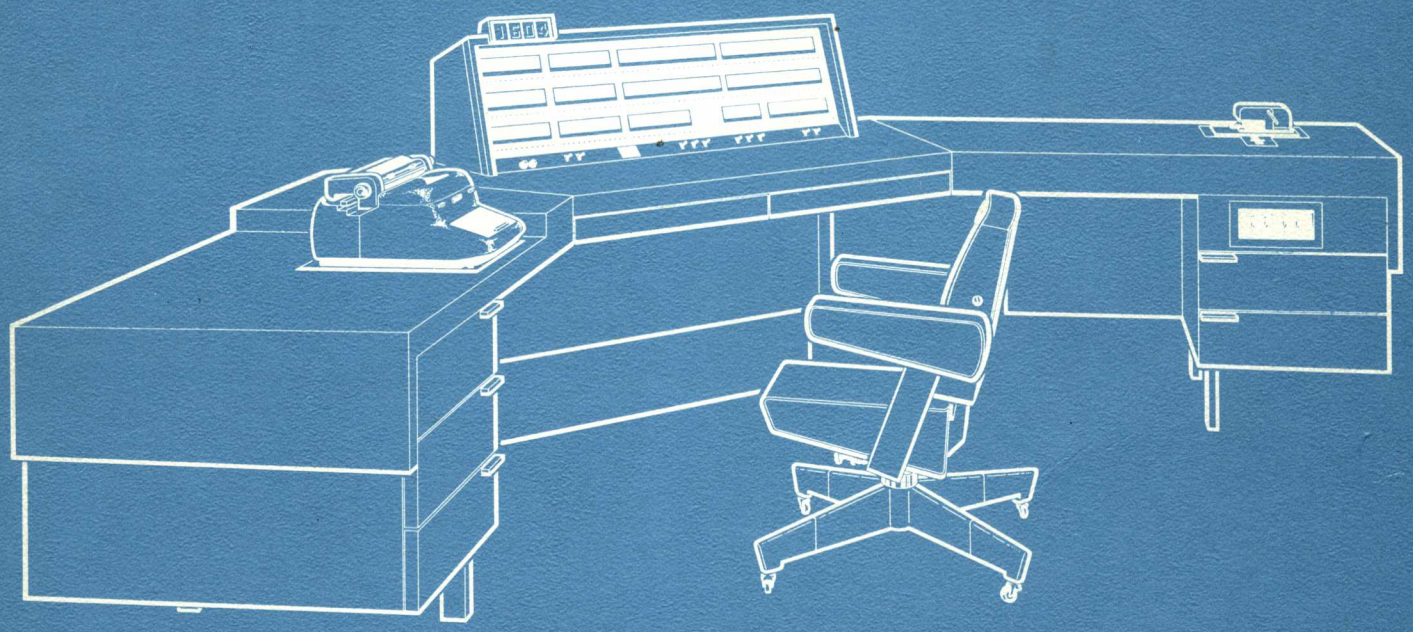
1604 COMPUTER

Volume 3: MAINTENANCE



INSTRUCTION BOOK

CONTROL DATA CORPORATION
MINNEAPOLIS, MINNESOTA

1604 COMPUTER

Volume 3:
MAINTENANCE

INSTRUCTION BOOK

PUBLICATION 033a

CONTROL DATA CORPORATION
MINNEAPOLIS, MINNESOTA

Revised 12/60

GENERAL TABLE OF CONTENTS

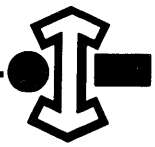
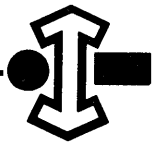


TABLE OF CONTENTS

1604 Computer Cabinet



CHAPTER 1

INTRODUCTION

This volume of the instruction book presents general information for maintaining the basic 1604 system. Its scope does not include all the intricacies of maintaining the computer, as it is assumed this knowledge has been acquired at Control Data training courses. Maintenance of external equipments such as the 1607 magnetic tape system and the 1605 adaptor is contained in the instruction manuals for these equipments.

Computer maintenance falls into the categories of preventive and corrective maintenance. Preventive maintenance is aimed at preventing failures during operation and consists of such procedures as lubricating, cleaning, running test programs, and checking for worn or marginal mechanical parts. Corrective maintenance consists of diagnosing, locating, and remedying the cause of a failure after it has occurred. This manual is mainly concerned with diagnosis and location of the cause of failure.

Of first importance in maintenance is a complete and thorough knowledge of the equipment. The primary sources of information about the logic of the computer are: Principles of Operation (volume 2), File of Equations (volume 4) and Logic Diagrams (volume 5). The File of Equations is the ultimate source of such information. In addition, the following aids to maintenance are provided:

Command Timing Charts (chapter 2 of this volume). The commands that execute each instruction are listed in sequential order (according to the relative computer time at which they occur).

Diagrams (volume 5 and appendix B of this volume). The logic and circuit diagrams in volume 5 show the logic of the computer according to functional areas. Schematic diagrams for the printed circuit cards are in appendix B.

Preventive Maintenance Schedule (appendix C of this volume). This schedule tabulates periodic preventive maintenance procedures.

Parts List. The Parts List provides information necessary for replacing defective parts and components. The units parts list section includes all components for a particular unit (cards, chassis, cabinet); the component parts list section

is a composite list of all components in the equipment.

Card Tester Manual. The card tester built by Control Data is a special purpose unit of test equipment for checking the performance of printed circuit cards. Test procedures and typical waveforms for each card type are provided in the manual.

TEST EQUIPMENT

Other test equipment necessary for servicing the computer consists of an ordinary volt-ohmeter, vacuum tube voltmeter (Hewlett-Packard HP-400D or equivalent) and a good oscilloscope (Tektronix 543 or equivalent). In addition to the ordinary hand tools commonly employed in electrical and mechanical maintenance a taper pin insertion tool and a crimping tool are needed.

COMPUTER IDENTIFICATION NUMBERING SYSTEM

A coordinate numbering system is used throughout the computer installation to locate exactly all items. Familiarization with this system is essential to maintenance. The principles of the system are tabulated in the following pages.

CABINET NUMBERING

Cabinet	Chassis	Component
0	0 0	0 0
1604 Main Computer		1 0 0 0 0
1604 Console		2 0 0 0 0
1607 Magnetic Tape System		3 0 0 0 0
1605 Adaptor		4 0 0 0 0
1606 Printer Control		5 0 0 0 0
1608 Adaptor		6 0 0 0 0
1609 Control Unit		7 0 0 0 0

When chassis or component numbers are not applicable, zeros are used instead.

CHASSIS NUMBERING

Cabinet	Chassis	Component
0	0 0	0 0

10000 Cabinet (main computer)

The 8 chassis of the main computer, as viewed from the top, are numbered as illustrated at the right. Fuses for each chassis are considered as mounted on the cabinet rather than the chassis.

10500	10400
10600	10300
10700	10200
10800	10100

Front of Main Cabinet

20000 Cabinet (console)

Relay chassis	20100	Paper tape switch panel	20500
Relay chassis	20200	Electric typewriter	20600
Connector panel	20300	Paper tape reader	20700
Control panel (switches and indicators)	20400	Paper tape punch	20800

Transformers, E-strips and the loudspeaker are considered as mounted on a cabinet rather than a chassis.

COMPONENT NUMBERING

Basic component numbering format:

Component Type	Cabinet	Chassis	Component
X X	0	0 0	0 0

Components on a Standard Chassis

Components on a standard chassis are numbered consecutively on the unit schematic diagram. The alphabetic designation of the component type is prefixed to the component identification number. Alphabetic designations are:

T - transformer
CR - rectifier
Q - transistor
R - resistor

Connectors on a Standard Chassis

The method for numbering each group of two cable connectors on the periphery of a standard chassis is shown in figure 1-1.

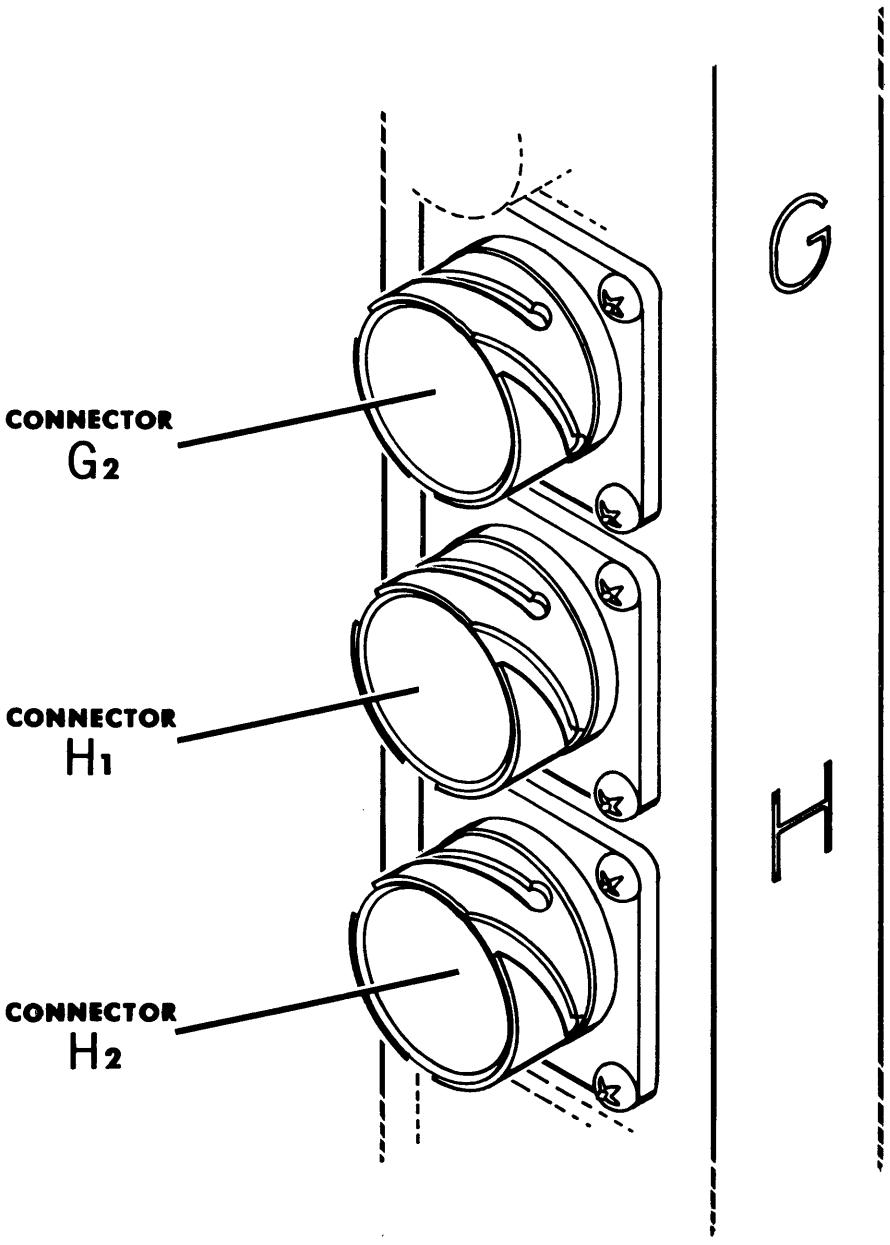
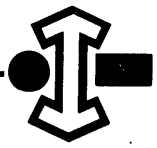


Figure 1-1. Cable Connector Identification.



COMPONENT NUMBERING

Printed Circuit Cards on a Standard Chassis

The coordinate system used to designate printed circuit card locations on a standard chassis is illustrated in figure 1-2. The letters and numbers which appear on the chassis are combined in the following format:

Chassis	Ordinate (row)	Abscissa (column)	Test Point
0 0	X	0 0	X

Cabinet numbers are omitted from the printed circuit card locations because equations and card placement are individual to each cabinet. Test point locations are identified by letter (A - top, B - middle, C - bottom) as they are viewed from the wiring side of the card.

Components in a Cabinet but not on a Chassis

All components located within a cabinet but not on a chassis (e.g. fuses) are numbered consecutively according to the basic component numbering format. A special case exists in the 10000 (main computer) cabinet where fuses are numbered with respect to the chassis they protect even though they are not physically located on the chassis.

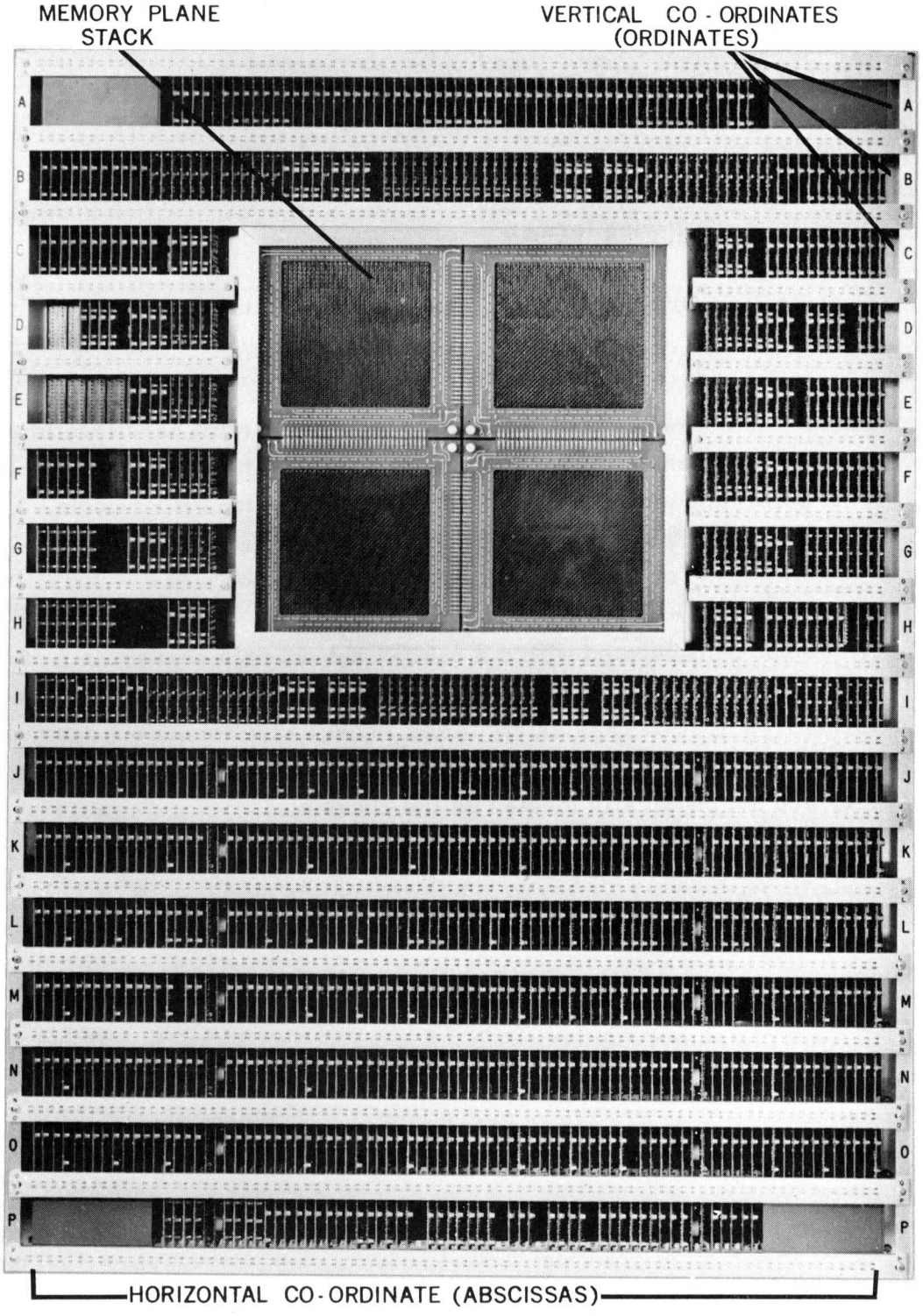
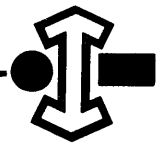


Figure 1-2. Card Side of a Typical Chassis.



CHAPTER 2

DIAGNOSTIC MAINTENANCE

Diagnosis of failure symptoms and location of their causes is one of the chief maintenance activities. Actual correction of a failure usually consists of the simple replacement of a card. The procedures of diagnostic maintenance are dictated by the prominence of logical structure in the computer and the variety of possible causes of initial symptoms. Analysis of symptoms, based on a thorough understanding of computer logic, is required.

TEST PROGRAMS

The functioning of a given part of the computer is checked by a test program; execution of the program causes operations to be performed in the part under test. The results of the operations are checked to determine if they are proper; an improper result produces one of several indications of a malfunction.

The test programs are available in a separate packet. Some of the programs are briefly described in the following paragraphs.

COMMAND TEST

The command test is the most comprehensive of the test programs. It checks all but three of the individual instructions. Included in the check are most subinstructions which provide options to main instructions. It does not check the transfer instructions (62 and 63) or some subinstructions of the external function instruction (74).

The entire test or an individual instruction may be selected for execution. The test, or selected part of it, may be repeated an optional number of minutes. Occurrence of an error stops the test and causes type out of information pertinent to the point of failure.

ARITHMETIC TEST

The arithmetic test checks the various parts of the computer which perform arithmetic operations, the A accumulator, the U^2 accumulator, and iterative sequence. It does not necessarily provide the comprehensive check of arithmetic instructions included in the command test. The test stops at the point of failure as indicated at the console by the content of registers displayed after stopping. The test is made up of the following parts:

1) Index Registers - checks index and R registers
2) Add-Subtract - checks the accumulator pyramid primarily
3) Integer Multiply-Divide - checks accumulator pyramid and parts of the iterative sequence
4) Fractional Multiply-Divide - checks accumulator pyramid and parts of the iterative sequence
5) U^2 Register - checks U^2 accumulator pyramid and associated circuits
6) X Register - checks the numerous uses of this register
7) Floating-Fix - checks floating-point and fixed-point instructions by comparing the results obtained from executing a floating-point instruction with those obtained from executing the corresponding fixed-point instruction. The same quantities are used as operands.

STORAGE TEST

This test checks the circuits employed in referencing each address in storage. Although it is intended to check marginal circuits, it will also reveal malfunctions that occur in normal conditions. It consists of the following parts:

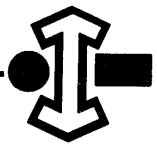
1) Changing noise pattern - while operating with low margins, changing patterns of bits are written and read. The reading and writing operations are checked for interference resulting from noise generated by the changing patterns.
2) Fixed noise pattern - similar to part 1 except that fixed patterns are used.
3) Diverter check - while operating with high margins, the writing and reading of bits are checked for errors caused by slow diverter circuits.

PAPER TAPE TEST

This test checks the performance of both the reader and punch during long continuous runs of tape and during short runs involving many starts and stops. A test tape is read and stored and the information is punched out. The new tape is read and compared with the original tape.

TYPEWRITER TEST

The operation of the typewriter is checked by typing in data which is subsequently typed out. The operator must make a visual comparison of output data with input data.



MAGNETIC TAPE TEST

The performance of the 1607 or the IBM tape units used with the 1605 adaptor are checked by this test. For either case the test causes the unit to perform all operations that can be required of it. The results of the operations are then examined for errors.

DIAGNOSIS FROM CONSOLE

The console with its display of register contents, background lights and operating controls provides for the first level of diagnosis. A test program reveals the presence of a malfunction and the general area of computer logic causing it. The first steps in localizing the failure to a more specific area, for example, a given register or instruction are accomplished by use of the console. For a description of the operating controls and background indicators see chapter 1 Operation, in volume 1.

Suppose, for example, the original symptom of the malfunction was improper results for instruction 14 Add. Since there are several possible causes of such a malfunction, the first step is to eliminate some of these possibilities. The basic procedure at the console is to execute in the step mode several of the other instructions (11 Increase A, 45 Add Logical etc.) which involve the adding operation.

After stepping through each of these instructions the actual result displayed in A is compared with the correct result. If instruction 11 also fails the malfunction must be in an area common to 11 and 14. Thus certain of the potential causes have been eliminated as possibilities. If, on the other hand, instruction 11 does not fail, the malfunction must be in an area not common to 11 and 14. This also eliminates certain other potential causes as possibilities. This procedure is continued, using more instructions, until the number of possibilities is greatly reduced. At this point the methods of the next section can be employed for complete identification of the cause of the malfunction.

LOGICAL CIRCUIT MAINTENANCE

After console diagnosis has indicated the circuits which may be causing the malfunction the operation of these circuits is examined by means of an oscilloscope.

In some cases observation of circuits in a static condition is sufficient; however examination of dynamic circuit conditions is often required. This is done by repeated execution of an instruction that uses the circuit. The UP position of the Storage Mode switch provides a convenient way of making such repetitions.

Information relevant to localizing the cause to a group of circuits and then to an individual circuit is contained in:

1) the file of equations (volume 4)
2) the command timing charts (at end of this chapter)
3) logic diagrams (volume 5)

The jack location and test point information required in taking waveforms for each circuit are provided by equations and diagrams.

The operation of a circuit card is examined by means of waveforms taken at its test points. The test points are on card output. Since the cards are basically inverters, waveforms are the inversion of the card inputs. The common ground connection for the oscilloscope is made at the outer chassis edge. A synchronizing signal for the oscilloscope can be obtained from the test point of another circuit. Typically the synchronizing source is chosen to produce a signal just in advance of the time when a circuit is to be examined.

Occasionally it is necessary to look at signals on the individual pins of a card. This is done by removing the adjacent bars which hold the cards in position, removing the card, inserting the card extender, and plugging the card into the extender. On the extender the pins of the card under test are easily accessible.

TEST MODE

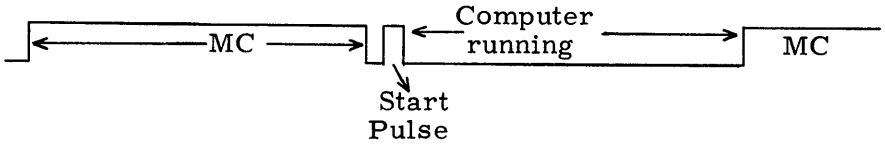
There are some situations for which the simple repetition of an instruction does not yield satisfactory dynamic waveforms. Examples of such situations are:

1) deep end - a sequence fails to exit
2) clean start is required, that is, observations are to be made after master clear.

The test mode is established by simultaneously depressing the Clear switch and raising the Start-Step switch. Raising the Clear switch (external master clear) terminates the test mode.

In the test mode the 60-cycle line frequency is employed as a low-speed oscillator to produce alternate master clears and start pulses (see below). During one cycle (16.6 milliseconds) the internal master clear is held on. The following cycle produces a

start pulse at the beginning. The computer is allowed to run until the next master clear (16.6 milliseconds later).



The start pulse initiates execution of instructions beginning with address 00000. Operation continues until the master clear occurs, or until a malfunction or stop occurs. Typically, an instruction is entered in the upper position of address 00000 such that it acts on the circuit to be checked.

STORAGE MAINTENANCE

Normally maintenance for the storage section involves running a test either to find marginal failures, as in preventive maintenance, or to locate the cause of an actual failure. The tests reveal the addresses or bits where the failure occurs. When the location has been determined the circuits are examined with the oscilloscope.

Much of storage maintenance is accomplished at the console by the use of the two storage test switches. The Mode switch in the UP positiion provides for repeatedly reading and executing the same pair of instructions which causes repeated references to the storage locations involved. The Mode switch in DOWN position provides for sweeping through (successively) all the addresses in storage.

The Margin switch in UP position raises the reference voltage on sense amplifiers, making them less sensitive to weak signals. In DOWN position this switch lowers the reference voltage to make the sense amplifiers more sensitive.

Storage maintenance requires a thorough knowledge of the storage section (chapter 4, volume 2). Pertinent diagrams are located in volume 5.

STORAGE TESTS

Storage testing for preventive maintenance is usually done by means of the test programs, however, a second method may be used to quickly enter the test from the console.

First, a test word (all "1's" or all "0's") is loaded into the A register and from there into every location. Second, the content of each location is read into the A register and a zero test is then made on A. Since the zero test checks both positive zero (all "0's") and negative zero (all "1's") it should always detect a zero. If a non-zero value is detected, a fault has occurred and operation stops. After stopping, the address of the fault is given by the content of the specified index register while the faulty bit (or bits) are indicated by the content of the A register.

A test word of all "1's" is used with high margins which reduces sensitivity and tends to cause weak signals to be dropped. The all "0's" test word is used with low margins which increases sensitivity and tends to cause spurious signals to be picked up.

Loading Storage With Test Word

1) Load address 00000 with the instructions 55 1 00000 (Index Jump) 20 1 00000 (Store A). Execution of this loop will load the test word to all addresses.
2) Master Clear
3) Enter test word in A
4) Enter 77777 in B^1
5) Raise Start-Step switch. Computer will stop when all addresses have been loaded with test word.

Testing Storage with All "1's" or "0's"

One storage is loaded with all "1's" or "0's" a zero test is made on each location. A 4-instruction loop loads A with the content of each location and then makes a zero test on A.

Address	Upper Instruction	Lower Instruction
00000	51 1 00001	12 1 00000
00001	22 0 00000	76 0 00000

(A is tested for both negative and positive zero by the 22.0 instruction.) Repetition of the loop stops when a non-zero value is detected. If no fault occurs it stops when the content of address 00000 is read into A. The test is initiated by the following procedure:

1) Load addresses 00000 and 00001 with the above instructions.
2) Master clear.
3) Raise Start-Step switch. Computer will stop immediately. This is not due to a failure; it results from reading the content of 00001 into A.

4) Raise Start-Step switch again. The test will run now until a failure occurs or the test ends at address 00000.

STORAGE WAVEFORMS

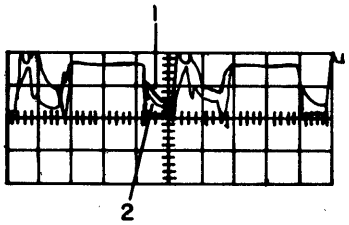
The preceding section describes techniques for determining whether there is a malfunction in the operation of storage. These techniques also reveal the address and bit of the malfunction. Further isolation to a specific card is accomplished by means of waveform analysis. Observed waveforms from pertinent cards are compared with normal waveforms from cards of the same type.

The normal waveforms from the various types of storage cards are included here. Card type 53 is omitted due to the similarity to the standard inverter circuit. In general, these waveforms were taken from circuits in even storage with the computer operating in the sweep mode.

Most of the waveforms are composite because of the sweep mode. For example the waveform for the 52 card shows both the working time of the diverter (rectangular portion) and also the period when it is not in use (base line).

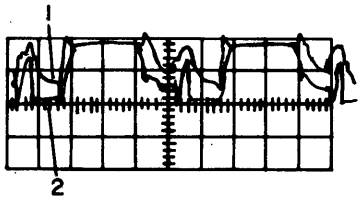
For all waveforms the oscilloscope is connected so that negative voltages produce upward deflection.

R/W DRIVER, 51 CARD



Read Side, Test Point A

1) Rounded pulse is a reflected read pulse from another driver that is turned on when this one is off.



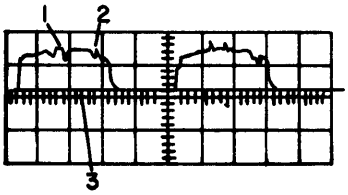
Write Side, Test Point C

2) Squared off pulse shows when this driver is turned on.

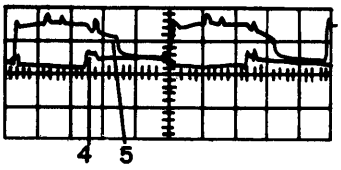
Vertical Sensitivity: 10 volts/cm

Sweep: 2 μsec/cm

DIVERTER, 52 CARD



Good diverter

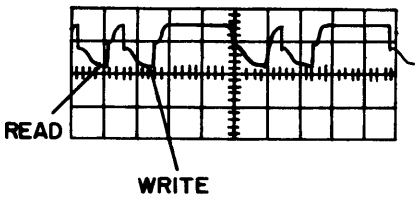


Bad diverter

1) End of read pulse
2) End of write pulse
3) Straight base line (a sign of a good diverter) shows time when diverter is on.
4) Step in base line indicates bad diverter due to faulty output transistor.
5) Slow drop off indicates marginal card.

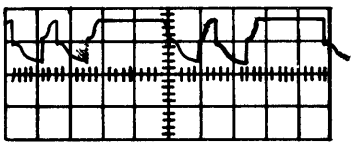
Vertical Sensitivity: 5 volts/cm
Sweep: 2 μsec/cm

CURRENT SOURCE, 54 CARD



Vertical R/W source

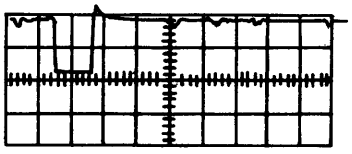
Vertical and horizontal sources should be very similar



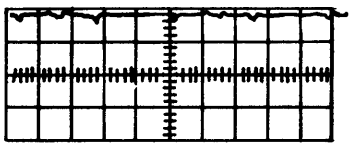
Horizontal R/W source

Vertical Sensitivity: 1 volt/cm
Sweep: 2 μsec/cm

INHIBIT GENERATOR, 55 CARD



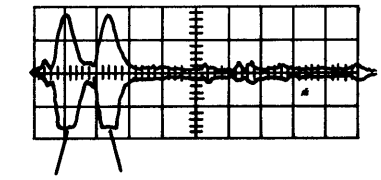
all "0's"



all "1's"

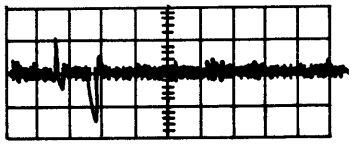
Vertical Sensitivity: 10 volts/cm
Sweep: 2 μsec/cm

SENSE AMPLIFIER, 56 CARD

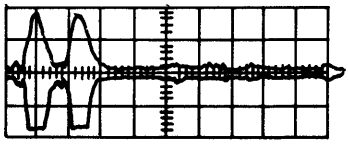


all "1's"

Test Point A



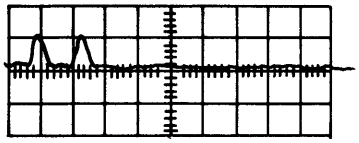
all "0's"



all "1's"

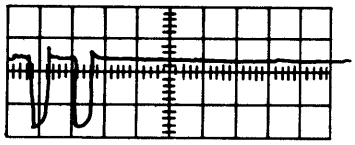
Test Point B

Test points A and B should yield essentially the same waveforms.



all "1's"

Test Point C



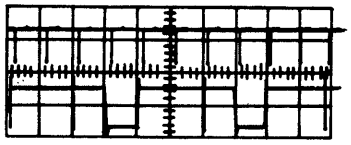
all "1's"

Test Point D

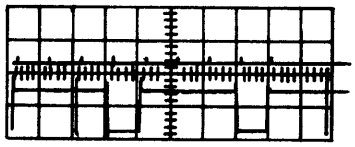
Vertical Sensitivity: 1 volt/cm

Sweep: 2 μsec/cm

Z REGISTER OUTPUTS



Upper trace: Set side of Z with all "1's"
Lower trace: Quadrant selection



Upper trace: Set side of Z with all "0's"
Lower trace: Quadrant selection

Vertical Sensitivity: 2 volts/cm

Sweep: 10 μsec/cm

COMMAND TIMING CHARTS

INTRODUCTION

The computer successively executes instructions from internally-stored programs by a sequence of commands. A command accomplishes one act, for example, transmitting data from one register to another or clearing a register. The operation code of the instruction to be executed selects one of the control sequences. This sequence is then initiated to generate the appropriate commands as determined by the operation code.

All commands involved in the execution of an instruction are listed in the order of occurrence in the command timing charts.* The instruction sequence used to generate the commands is specified under the heading Sequence.

Entries in the Time column indicate the phase time (0.2 μsec in duration) at which the associated command signal occurs. These phase times are related to the phase times at which the sequence is initiated. Initiate time is always considered as time 00. For command signals rising from control flip-flops (FFs) rather than control delays, the entry in the Time column indicates the last time the signal is clocked. Usually this is the time when the control FF is set. The resulting command does not actually take effect until approximately two phase times later.

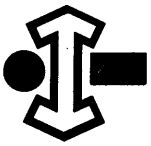
The three entries given under Execution Times take account of the time for three cases of instruction use. Variations in execution time are caused by such factors as:

1) Upper or lower position in instruction word
2) Consecutive references to the same storage unit
3) Storage reference at end of preceding instruction

All three time entries are determined by averaging the times for a long list of the same instruction. Minimum time is an average of a list arranged so that the factors above have minimum values; maximum time is an average of a list in which these factors have maximum values; and average time applies to a list arranged for typical values of the factors.

Comments in the Remarks column describe the function of the command in the execution of the instruction.

*It should be noted that those commands which are generated but are not pertinent to the execution of the instruction have been omitted from the charts.



GLOSSARY OF ABBREVIATIONS

A	arithmetic register
Adv Clk	advance clock
AQ	the double-length register comprised of A and Q
B^b	the designated index register
Buf	buffer
Comp	complement
Exp	exponent
FF	flip-flop
Init	initiate
Inst	instruction
Int	interrupt
I^2I^3	the inverter rank preceding R
I^5I^6	the inverter rank between the storage circuits and the arithmetic and control circuits
LQX	the logical (bit-by-bit) product of Q and X
m	the base execution address
M	the modified execution address
Neg	negative
P	program address register
Part	partial
Pos	positive
Q	auxiliary arithmetic register
R	address buffer register
Red	reduce
SR	sign record
U^1	program control register
U^2	auxiliary program control register
X	exchange register
Z	storage restoration register
→	(arrow) transmit the contents
( )	(parentheses) contents of a register
subscript f	final contents of a register
subscript i	initial contents of a register
subscript L	lower half of a register
subscript LA	the address portion (lowest 15 bits) of the lower instruction
subscript U	upper half of a register
subscript UA	the address portion (lowest 15 bits) of the upper instruction

CODE	INSTRUCTION	FUNCTION
RNI	Read Next Instruction	Prepare computer for receipt of instruction word from storage and for execution of next instruction.

SEQUENCE: Read Next Instruction

EXECUTION TIME:

TIME	COMMAND	CONDITION	REMARKS
00	Adv P^2 to P^1	Full Exit	Add 1 to (P_i)
00	Initiate Storage	Full Exit	Reference address $P_i + 1$
00	Wait Storage	Full Exit	
08	Set Exit FF	Full Exit	Establish mode for concluding the instruction
08	Clear Exit FF	Half Exit	
09	Clear U	Full Exit	
09	Clear U^1_U	Half Exit	Set up current instruction in U^1_U
10	$B^b \rightarrow I^2 I^3$		
11	Set Stop II FF		Step or stop or breakpoint
11	$I^5 I^6 \rightarrow U^1$	Full Exit	
11	$U^1_L \rightarrow U^1_U$	Half Exit	
11	Clear R^1		Prepare R^1 for receipt of (B^b)
11	Clear Interrupt Lockout FF	Interrupt Complete	P=00007 terminates the interrupt instruction routine
12	Wait Step		RNI stops to await subsequent start or step pulse
14	$I^2 I^3 \rightarrow R^1$	$b \neq 0$	Transfer (B^b) to R^1
14	$U^1 \rightarrow U^2$		
15	Clear X^1		

CODE	INSTRUCTION	FUNCTION
01 ARS	A Right Shift	Shift (A) right M places

SEQUENCE: Zero Address (H^2-- V^2--)

EXECUTION TIME: 4.0 us. min. (Lower Inst.) 5.6 us. min. (Upper Inst.) 2.8 us. + .4 us./shift avg., 54.4 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer M to U^2
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
06	$U^2 \rightarrow R^2$		Place shift count in R^1
07	$R^2 \rightarrow R^1$		
09	Set Shift Fault FF	Shift Count $> 127_{10}$	
10	Set A Right FF	$R \neq 0$	
10	Init. Shift		
11	Set Exit Control FF		
12	Red. R^1 to R^2	$R \neq 0$	Reduce shift count; shift
12	Shift 1 Place	$R \neq 0$	
13	Half Exit	$R = 0$	
13	Full Exit	$R = 0$	

CODE	INSTRUCTION	FUNCTION
02 QRS	Q Right Shift	Shift (Q) right M places

SEQUENCE: Zero Address (H^2-- V^2--)

EXECUTION TIME: 4.0 us. min. (Lower Inst.)
5.6 us. min. (Upper Inst.) 2.8 us. + .4 us./shift avg., 54.4 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
06	$U^2 \rightarrow R^2$		Place shift count in R^1
07	$R^2 \rightarrow R^1$		
09	Set Shift Fault FF	Shift Count $> 127_{10}$	
10	Set Q Right FF	$R \neq 0$	
10	Init. Shift		
11	Set Exit Control FF		
12	Red. R^1 to R^2	$R \neq 0$	Reduce shift count; shift
12	Shift 1 Place	$R \neq 0$	
13	Half Exit	$R = 0$	
13	Full Exit	$R = 0$	

CODE	INSTRUCTION	FUNCTION
03 LRS	AQ Right Shift	Shift (AQ) right M places

SEQUENCE: Zero Address

EXECUTION TIME: 4.0 us. min. (Lower Inst.)
5.6 us. min. (Upper Inst.)
2.8 us. + .4 us./shift avg., 54.4 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
06	$U^2 \rightarrow R^2$		Place shift count in R^1
07	$R^2 \rightarrow R^1$		
09	Set Shift Fault FF	Shift Count $> 127_{10}$	
10	Set A and Q Right FFS	$R \neq 0$	
11	Set Exit Control FF		
12	Shift 1 Place	$R \neq 0$	Reduce shift count; shift
12	Red. R^1 to R^2	$R \neq 0$	
13	Half Exit	$R = 0$	
13	Full Exit	$R = 0$	

CODE	INSTRUCTION	FUNCTION
04 ENQ	Enter Q	Transfer M to Q^1, extend the sign

SEQUENCE: Zero Address (H^2-- V^2--)

EXECUTION TIME: 2.8 us. min., 3.0 us. avg., 3.2 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Prepare X for use as exchange register
03	$A^2 \rightarrow Q^1$		Store (A_i) temporarily in Q^1
03	Clear A^1		Prepare A to receive M
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
07	Set f = 04 FF		
07	Set f = 04,10,11 FF		Conditions later commands
07	$U^2 \rightarrow X^1$ (with ^{LA} extension)		Place M in lower 15 stages of X^1, extend the 15th bit through X
08	$X^1 \rightarrow X^2$		Place M in X^2 for transfer to A
09	Half Exit		
09	Full Exit		
13	Part. Add X^2 to A^1		Transfer M to A^1
14	$Q^1 \rightarrow Q^2$		Store (A_i) in Q^2
14	$A^1 \rightarrow A^2$	unconditional	Transfer M to A^2
15	$A^2 \rightarrow Q^1$		Transfer M to Q
15	$Q^2 \rightarrow A^1$		Restore (A_i) to A

CODE	INSTRUCTION	FUNCTION
05 ALS	A Left Shift	Shift (A) left M places

SEQUENCE: Zero Address (H^2-- V^2--)

EXECUTION TIME: 4.0 us. min. (Lower Inst.)
5.6 us. min. (Upper Inst.)
2.8 us. + .4 us./shift avg., 54.4 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
06	$U^2 \rightarrow R^2$		Place shift count in R^1
07	$R^2 \rightarrow R^1$		
09	Set Shift Fault FF	Shift Count $> 127_{10}$	
10	Set A Left FF	$R \neq 0$	
10	Init. Shift		
11	Set Exit Control FF		
12	Red. R^1 to R^2	$R \neq 0$	Reduce shift count; shift
12	Shift 1 Place	$R \neq 0$	
13	Half Exit	$R = 0$	
13	Full Exit	$R = 0$	

CODE	INSTRUCTION	FUNCTION
06 QLS	Q Left Shift	Shift (Q) left M places

SEQUENCE: Zero Address (H^2-- V^2--)

EXECUTION TIME: 4.0 us. min. (Lower Inst.)
5.6 us. min. (Upper Inst.) 2.8 us. + .4 us./shift avg., 54.4 us. max.

TIME	COMMAND	CONDITION	REMARKS
01	$U^1 \rightarrow U^2$		Transfer m to U^2
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
06	$U^2 \rightarrow R^2$		Place shift count in R^1
07	$R^2 \rightarrow R^1$		
09	Set Shift Fault FF	Shift Count 127_{10}	
10	Set Q Left FF	$R \neq 0$	
10	Init. Shift		
11	Set Exit Control FF		
12	Red. R^1 to R^2	$R \neq 0$	Reduce shift count; shift
12	Shift 1 Place	$R \neq 0$	
13	Half Exit	$R = 0$	
13	Full Exit	$R = 0$	

CODE	INSTRUCTION	FUNCTION
07 LLS	AQ Left Shift	Shift AQ left M places

SEQUENCE: Zero Address (H^2-- V^2--)

EXECUTION TIME: 4.0 us. min. (Lower Inst.)
5.6 us. min. (Upper Inst.) 2.8 us. + .4 us./shift avg., 54.4 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
06	$U^2 \rightarrow R^2$		Place shift count in R^1
07	$R^2 \rightarrow R^1$		
09	Set Shift Fault FF	Shift Count $> 127_{10}$	
10	Set A and Q Left FF's	$R \neq 0$	
10	Init. Shift		
11	Set Exit Control FF		
12	Shift 1 Place	$R \neq 0$	Reduce shift count; shift
12	Red. R^1 to R^2	$R \neq 0$	
13	Half Exit	R=0	
13	Full Exit	R=0	

CODE	INSTRUCTION	FUNCTION
10 ENA	Enter A	Transfer M to A^1, extend the sign

SEQUENCE: Zero Address (H^2-- V^2--)

EXECUTION TIME: 2.8 us. min., 3.0 us. avg., 3.2 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Prepare X for use as exchange register
03	Clear A^1		Prepare A to receive M
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
07	Set F = 04,10,11 FF		Conditions later commands
07	$U^2 \rightarrow X^1_{LA}$ (with extension)		Place M in lower 15 stages of X, extend the 15th bit through X.
08	$X^1 \rightarrow X^2$		Place M in X^2 for transfer to A
09	Half Exit		
09	Full Exit		
13	Part. Add X^2 to A^1		Transfer M to A.

CODE	INSTRUCTION	FUNCTION
11 INA	Increase A	Add M to (A), store the result in A

SEQUENCE: Zero Address (H^2-- V^2--)

EXECUTION TIME: 2.8 us. min., 3.0 us. avg., 3.2 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Prepare X for use as exchange register
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
07	Set f = 04,10,11 FF		Conditions later commands
07	$U^2 \rightarrow X^1_{LA}$ (with extension)		Place M in lower 15 stages of X, extend the 15th bit through X.
08	$X^1 \rightarrow X^2$		Position M in X^2 for addition to A
09	Full Exit		
09	Half Exit		
13	Add X^2 to A^1		Add M to A

CODE	INSTRUCTION	FUNCTION
12 LDA	Load A	Transfer (M) to A

SEQUENCE: Read Operand (H^3-- V^3--)

EXECUTION TIME: 4.8 us. min., 7.2 us. avg., 9.6 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Prepare X for use as exchange register
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
04	Init. Storage		
06	Clear A^1		Prepare A^1 to receive M
10	Wait Storage		
15	$I^5I^6 \rightarrow X^1$		Transfer (M) to X^1
16	$X^1 \rightarrow X^2$		Place (M) in X^2 for transfer to A
17	Half Exit		
17	Full Exit		
21	Part Add X^2 to A^1		Transfer (M) to A

CODE	INSTRUCTION	FUNCTION
13 LAC	Load A, Complement (Negative A)	Transfer the complement (M) to A

SEQUENCE: Read Operand (H^3-- V^3--)

EXECUTION TIME: 4.8 us. min., 7.2 us. avg., 9.6 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Prepare X for use as an exchange register
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
04	Initiate Storage		
06	Clear A^1		Prepare A for receipt of (M)
10	Wait Storage		
15	$I^5I^6 \rightarrow X^1$		Transfer (M) to X^1
16	Comp. $X^1 \rightarrow X^2$		Complement (M)
17	Half Exit		
17	Full Exit		
21	Part Add X^2 to A^1		Transfer complement (M) to A

CODE	INSTRUCTION	FUNCTION
14 ADD	Add	Add (A) and (M), store the sum in A

SEQUENCE: Read Operand (H^3-- V^3--)

EXECUTION TIME: 4.8 us. min., 7.2 us. avg., 9.6 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Prepare X for use as an exchange register
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
04	Init. Storage		
10	Wait Storage		
15	$I^5I^6 \rightarrow X^1$		Transfer (M) to X^1
16	$X^1 \rightarrow X^2$		Place (M) in X^2 for addition to A
17	Half Exit		
17	Full Exit		
21	Add X^2 to A^1		Add (M) to A^1

CODE	INSTRUCTION	FUNCTION
15 SUB	Subtract	Subtract (M) from (A), store the difference in A

SEQUENCE: Read Operand (H^3-- V^3--)

EXECUTION TIME: 4.8 us. min., 7.2 us. avg., 9.6 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Prepare X for use as an exchange register
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
04	Init. Storage		
10	Wait Storage		
15	$I^5I^6 \rightarrow X^1$		Transfer (M) to X^1
16	Comp. $X^1 \rightarrow X^2$		Complement (M)
17	Full Exit		
17	Half Exit		
21	Add X^2 to A^1		Add complement (M) to A

CODE	INSTRUCTION	FUNCTION
16 LDQ	Load Q	Transfer (M) to Q

SEQUENCE: Read Operand (H^{3}-- V^{3}--)

EXECUTION TIME: 4.8 us. min., 7.2 us. avg., 9.6 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Prepare X for use as an exchange register
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
04	Init. Storage		
06	Clear A^1		Prepares A^1 for receipt of (M)
07	$A^2 \rightarrow Q^1$		Store (A_i) temporarily in Q
10	Wait Storage		
15	$I^5 I^6 \rightarrow X^1$		Transfer (M) to X
15	Set f=16, 17 FF		Conditions later commands
16	$X^1 \rightarrow X^2$		Place (M) in X^2 for transfer to A
17	Half Exit		
17	Full Exit		
21	Part. Add X^2 to A^1		Add (M) to A^1
22	$Q^1 \rightarrow Q^2$		Place (A_i) in Q^2 for transfer back to A
22	$A^1 \rightarrow A^2$		Unconditional transfer of (M) to A^2
23	$A^2 \rightarrow Q^1$		Transfer (M) to Q^1
23	$Q^2 \rightarrow A^1$		Restore (A_i) to A

CODE	INSTRUCTION	FUNCTION
17 LQC	Load Q, Complement (Negative Q)	Transfer the complement (M) to Q

SEQUENCE: Read Operand (H^3-- V^3--)

EXECUTION TIME: 4.8 us. min., 7.2 us. avg., 9.6 us. max.

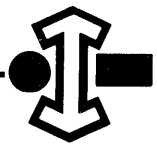
TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Prepare X for use as an exchange register
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
04	Init. Storage		
06	Clear A^1		Prepare A^1 for receipt of complement (M)
07	$A^2 \rightarrow Q^1$		Store (A) temporarily in Q
10	Wait Storage		
15	$I^5I^6 \rightarrow X^1$		Transfer (M) to X
15	Set F=16,17 FF		Conditions later commands
16	Comp. $X^1 \rightarrow X^2$		Complement (M)
17	Half Exit		
17	Full Exit		
21	Part. Add X^2 to A^1		Add complement (M) to A^2
22	$Q^1 \rightarrow Q^2$		Place (A_1) in Q^2 for transfer back to A
22	$A^1 \rightarrow A^2$		Unconditional transfer of (M) to A^2
23	$A^2 \rightarrow Q^1$		Transfer complement (M) to Q
23	$Q^2 \rightarrow A^1$		Restore (A_1) to A

CODE	INSTRUCTION	FUNCTION
20 STA	Store A	Transfer (A) to M

SEQUENCE: Write Operand (H^4-- V^4--)

EXECUTION TIME: 4.8 us. min., 7.2 us. avg., 9.6 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Prepare X for use as an exchange register
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
04	Init. Storage		
07	$A^1 \rightarrow X^1$		
07	Wait Storage		Transfer (A) to storage via X
08	Enable Full Write		
15	$X^1 \rightarrow Z^1\ Z^2$		
15	Half Exit		
15	Full Exit		



CODE	INSTRUCTION	FUNCTION
21 STQ	Store Q	Transfer (Q) to M

SEQUENCE: Write Operand (H^4-- V^4--)

EXECUTION TIME: 4.4 us. min., 7.2 us. avg., 9.6 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Prepare X for use as an exchange register
02	$Q^1 \rightarrow Q^2$		Place Q_i in Q^2 for transfer to A
03	$A^2 \rightarrow Q^1$		Store A_i in Q temporarily
03	$Q^2 \rightarrow A^1$		Store Q_i in A temporarily
04	$A^1 \rightarrow A^2$		Unconditional transmission
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
04	Init. Storage		
06	$Q^1 \rightarrow Q^2$		Place A_i in Q^2 for return to A
07	$A^1 \rightarrow X^1$		Transfer Q_i to X
07	Wait Storage		
08	Enable Full Write		Prepare to transfer Q_i to storage
11	$Q^2 \rightarrow A^1$		Restore A_i to A
11	$A^2 \rightarrow Q^1$		Restore Q_i to Q
15	$X^1 \rightarrow Z^1\ Z^2$		Transfer Q_i to storage
15	Half Exit		
15	Full Exit		

CODE	INSTRUCTION	FUNCTION
22 AJP	A Jump	b = 0,1,2 or 3:Normal jump on specified condition of (A) b = 4,5,6 or 7:Return jump on specified condition of (A)

SEQUENCE: Normal Jump (b = 0, 1, 2 or 3)
Write Operand (b = 4, 5, 6 or 7)
EXECUTION TIME: 4.0 us. min., 7.2 us, avg., 11.6 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Place m in U^2 for transfer to P
01	Clear X^1		Prepare X for use as exchange register for return jump.
Normal Jump Sequence			
03	Jump Exit	b=0,(A)=0 b=1,(A)≠0 b=2, A pos b=3, A neg	Jump
03	Half Exit	No Jump	Conclude instruction
03	Full Exit	No Jump	Conclude instruction
03	$U^2 \rightarrow P^1$	Jump	Place next instruction address in P
Write Operand Sequence			
		b=4, A=0 b=5, A≠0 b=6, A pos b=7, A neg	Jump
04	Initiate Storage	Jump	
06	Adv. P^1 to P^2	Jump	Next address of current routine
07	Half Exit	No Jump	
07	Full Exit	No Jump	
07	Wait Storage	Jump	
08	$P^1 \rightarrow X^2_{LA}$	Jump	Transfer next address of main routine to X^2_{LA}
08	$U^2 \rightarrow P^1$	Jump	Transfer m to P to select 1st instruction word of subroutine
08	Enable Partial Write Upper	Jump	Prepare to write next address of main routine into storage

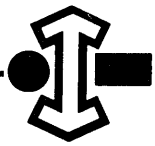
22 AJP

TIME	COMMAND	CONDITION	REMARKS
08	Set Return Jump FF	Jump	Conditions later commands
08	Enable Partial Write Upper	Jump	Prepare to store next address of main program (P_i)
09	$X^2 \longrightarrow X^1$	Jump	Position P_i in X^1_{LA} for transfer to X^1_{UA}
11	Clear U^1	Jump	Prepare U^1 for next instruction
13	$X^1_L \longrightarrow X^1_U$	Jump	Place P_i in X^1_U for transfer to storage
15	$X^1_U \longrightarrow Z^1Z^2$	Jump	Transfer next address of main program to storage
15	Half Exit	Jump	
15	$I^5I^6 \longrightarrow U^1$	Jump	Transfer first instruction of subroutine to U^1

CODE	INSTRUCTION	FUNCTION
23 QJP	Q Jump	b = 0,1,2 or 3:Normal jump on specified condition of (Q) b = 4,5,6 or 7:Return jump on specified condition of (Q)

SEQUENCE: Normal Jump (b = 0, 1, 2 or 3)
Write Operand (b = 4, 5, 6 or 7)
EXECUTION TIME: 4.0 us. min., 7.2 us, avg., 11.6 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Place m in U^2 for transfer to P
01	Clear X^1		Prepare X for use as exchange register for return jump
NORMAL JUMP SEQUENCE			
03	Jump Exit	b=0, (Q)=0 b=1, (Q)≠0 b=2, Q pos b=3, Q neg	} Jump
03	Half Exit	} No Jump	Conclude instruction
03	Full Exit		
03	$U^2 \rightarrow P^1$	Jump	Place next instruction address in P
WRITE OPERAND SEQUENCE			
		b=4, Q=0 b=5, Q≠0 b=6, Q pos b=7, Q neg	} Jump
04	Initiate Storage	Jump	
06	Adv. P^1 to P^2	Jump	Next address of current routine
07	Half Exit	} No Jump	
07	Full Exit		
07	Wait Storage	Jump	
08	$P^1 \rightarrow X^2_{LA}$	Jump	Transfer next address of current routine to X^2_{LA}
08	$U^2 \rightarrow P^1$	Jump	Transfer m to P to select 1st instruction word of subroutine



23 QJP

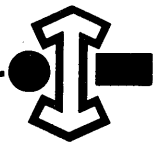
TIME	COMMAND	CONDITION	REMARKS
08	Enable Part. Write Upper (X^1_{LA})	Jump	Prepare to write next address of main routine into storage
08	Set Return Jump FF	Jump	Conditions later commands
09	$X^2 \longrightarrow X^1$	Jump	Position P_i in X^1_{LA} for transfer to X^1_{UA}
11	Clear U^1	Jump	Prepare U^1 for next instruction
13	$X^1_L \longrightarrow X^1_U$	Jump	Place P in X^1_{UA} for transfer to storage
15	Half Exit	Jump	
15	$I^5I^6 \longrightarrow U^1$	Jump	Transfer first instruction of subroutine to U^1
15	$X^1_U \longrightarrow Z^1Z^2$	Jump	Transfer next address of main program to storage

CODE	INSTRUCTION	FUNCTION
24 MUI	Multiply Integer	Multiply (M) by A; store the 96-bit product in QA

SEQUENCE: Iterative (H^6-- V^6--)

EXECUTION TIME: 25.2 us.min., 25.2 us. + .8 us./'1' in Q avg., 66.4 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Set X^1 to zeros
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
04	Comp. $X^1 \rightarrow X^2$		Set X^2 to 'all ones'
04	Init. Storage		Select M
04	Wait Storage		
05	Set Sign Record FF	A neg.	Register the sign of the multiplier
07	Clear R^1		Prepare R to hold the step control count
08	Set I^2 to 48		Generate step control count
12	I^2 $I^3 \rightarrow R^1$		Load count in R
13	Clear X^1		Prepare X^1 to receive multiplicand (M)
13	Part. Add X^2 to A^1	A neg.	Complement A if negative
14	$Q^1 \rightarrow Q^2$		
14	$A^1 \rightarrow A^2$	unconditional	
14	Clear A^1		Transfer multiplier (A_1) to Q
15	$Q^2 \rightarrow A^1$		
15	$A^2 \rightarrow Q^1$		
15	Exit to Mult. Step		
15	I^5 $I^6 \rightarrow X^1$		Position (M) in X^2 for generation of partial products
16	$X^1 \rightarrow X^2$		



24 MUI

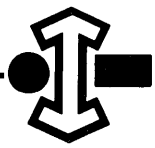
TIME	COMMAND	CONDITION	REMARKS
16	$R^1 \rightarrow R^2$		Set R^2 to 48
17	Comp. Sign Record	X Neg.	Establish the sign of the product
18	Comp. $X^1 \rightarrow X^2$	X neg.	Complement (M) if negative
EXECUTE MULTIPLY STEP			
00	Reduce R^1 to R^2		Perform the actual multiplication
00	Shift AQ Right		Shift to position multiplier bit in sensing position. Add (X) to A if multiplier bit is 1, then shift (AQ) right: if the multiplier bit is 0; shift AQ right. Reduce the step control count once each shift. Exit when the step control count is 0.
01	$R^2 \rightarrow R^1$	Short Loop	
01	$A^2 \rightarrow A^1$		
01	$Q^2 \rightarrow A^1$		
01	Exit to 0	$R \neq 0$	
		$Q_{00}=0$	Long
	End Correction	$R = 0$	Loop
05	Add X^2 to A^1	$Q_{00}=1$	
05	Exit to 0	$R \neq 0$	
05	Exit to End Correction	$R = 0$	
EXECUTE END CORRECTION			
01	Set Part. Add in A FF		Exit immediately if the product is positive. If the Sign Record flip-flop indicates a negative product in A, Q is complemented before concluding the routine.
03	Clear X^1		
04	Comp. $X^1 \rightarrow X^2$		
05	Exit		
05	Half Exit		
05	Part. Add X^2 to A^1		
06	$Q^1 \rightarrow Q^2$		
07	$A^2 \rightarrow Q^1$		
07	$Q^2 \rightarrow A^1$		
09	Part. Add X^2 to A^1	Sign Record = 1	

CODE	INSTRUCTION	FUNCTION
25 DVI	Divide Integer	Divide (QA) by (M). Store the quotient in A, and the remainder in Q.

SEQUENCE: Iterative (H^6-- V^6--)

EXECUTION TIME: 63.6 us. min., 65.2 us. avg., 66.4 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Set X^1 to zeros
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
04	Init. Storage		Select M
04	Wait Storage		
04	Comp. $X^1 \rightarrow X^2$		Set X^2 to 'all ones'
05	Set Dividend Sign Record FF	Q neg.	Record the sign of the dividend
05	Set Sign Record FF	Q Neg.	
06	Part. Add X^2 to A^1	Q Neg.	Complement A if AQ is neg.
07	Clear R^1		Prepare R for divide step count
08	Set I^2 to 48		Select divide step control count
10	$Q^1 \rightarrow Q^2$		
10	$A^1 \rightarrow A^2$	unconditional	Interchange (A) and (Q)
11	$Q^2 \rightarrow A^1$		
11	$A^2 \rightarrow Q^1$		
12	I^2 $I^3 \rightarrow R^1$		Place step count (48) in R^1
13	Clear X^1		Prepare X^1 for receipt of M (divisor)
13	Part. Add X^2 to A^1	A neg.	Complement Q_i if AQ is neg.



25 DVI

TIME	COMMAND	CONDITION	REMARKS
14	$Q^1 \rightarrow Q^2$		
15	$I^5\ I^6 \rightarrow X^1$		Transfer Divisor (M) to X
16	$X^1 \rightarrow X^2$		Position (M) in X^2 for generating partial dividends
16	$R^1 \rightarrow R^2$		Set R^2 = 48
17	Comp. Sign Record FF	X neg.	Establish sign of quotient
17	Exit to Divide Step		
18	Comp. $X^1 \rightarrow X^2$	X pos.	Complement M if neg.

25 DVI

TIME	COMMAND	CONDITION	REMARKS
EXECUTE	DIVIDE STEP		
00	Red. R^1 to R^2		Perform the division. Set least significant bit in Q to '1' if $X \leq A$; to '0' if $X > A$. Shift AQ left once after comparing X to A. Reduce R one count for each shift. Conclude the division when R = 0.
00	Shift AQ Left		
01	$R^2 \rightarrow R^1$		
01	$A^2 \rightarrow A^1$		
01	$Q^2 \rightarrow Q^1$		
01	Exit to 00	$R \neq 0$, $A < X$	
01	Exit to End Correction	R = 0	
05	Add X^2 to A^1	$A \geq X$	
05	Set Q_{00} to 1	$A \geq X$	
05	Exit to 00	$R \neq 0$	
05	Exit to End Correction	R = 0	
EXECUTE	END CORRECTION		
00	Set Divide Fault	Q neg.	The quotient is initially determined as a pos. quantity; if a '1' is present in Q_{47}, a fault exists.
03	Clear X^1		
04	Comp. $X^1 \rightarrow X^2$		
05	Part. Add X^2 to A^1	Div. Sign = 1	Complement remainder if dividend negative
06	$Q^1 \rightarrow Q^2$		Place quotient in A, remainder in Q
07	$A^2 \rightarrow Q^1$		
07	$Q^2 \rightarrow A^1$		
09	Part. Add X^2 to A^1	Sign record=1	Complement quotient

CODE	INSTRUCTION	FUNCTION
26 MUF	Multiply Fractional	Multiply the fractional quantity in M by the fractional quantity in A, store the 96-bit product in AQ

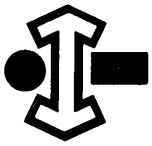
SEQUENCE: Iterative (H^{600} V^{600})

EXECUTION TIME: 25.2 us. min., 25.2 us. + .8 us./ '1' in Q avg., 66.4 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Set X to zero
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
04	Init Storage		Select M
04	Wait Storage		
04	Comp. $X^1 \rightarrow X^2$		Set X to 'all ones'
05	Set Sign Record FF	A neg	Register the sign of the multiplier
07	Clear R^1		Prepare R to hold the multiplication step count
07	Partial Add X^2 to A^1	A neg	Complement the multiplier if it is negative
08	Set I^2 to 47		Select the multiply step control count
10	$Q^1 \rightarrow Q^2$		Transfer the multiplier to Q
11	$A^2 \rightarrow Q^1$		
11	$Q^2 \rightarrow A^1$		
12	$I^2I^3 \rightarrow R^1$		Place the division step control count in R^1
13	Clear X^1		Prepare X^1 to receive multiplicand
14	Clear A^1		Clear A to receive the partial product
14	$Q^1 \rightarrow Q^2$		
15	$I^5I^6 \rightarrow X^1$		Transfer the multiplicand to X
15	Exit to Multiply Step		

26 MUF

TIME	COMMAND	CONDITION	REMARKS
16	$R^1 \rightarrow R^2$		Set $R^2 = R^1$
16	$X^1 \rightarrow X^2$		
16	Clear Part. Add in A FF		
17	Comp. Sign Record	X neg	Establish the sign of the product
18	Comp. $X^1 \rightarrow X^2$	X neg	Position (M) in X^2 for generating the partial products; complement if negative.
EXECUTE MULTIPLY STEP			
00	Reduce R^1 to R^2		Perform the actual multiplication. Shift to position mult. bit in sensing position.
00	Shift AQ Right		Add (X) to S if multiplier bit is 1, then shift (AQ) right; if the multiplier bit is 0, shift AQ right. Reduce the step control count once each shift. Exit when the step control count is 0.
01	$R^2 \rightarrow R^1$	Short Loop	
01	$A^2 \rightarrow A^1$		
01	$Q^2 \rightarrow Q^1$		
01	Exit to 0	$R \neq 0$	
01	Exit to End Correction	$Q_{00}=0$ $R=0$	Long Loop
05	Add X^2 to A^1	$Q_{00}=1$	
05	Exit to 0	$R \neq 0$	
05	Exit to End Correction	$R = 0$	



26 MUF

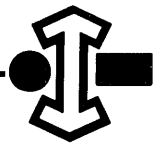
TIME	COMMAND	CONDITION	REMARKS
	EXECUTE END CORRECTION		
01	Set Part. Add in A FF		
02	$Q^1 \rightarrow Q^2$		Exit immediately if the product is positive. If the Sign Record flip-flop indicates a negative product in A, Q is complemented before concluding the routine.
03	$A^2 \rightarrow Q^1$		
03	$Q^2 \rightarrow A^1$		
03	Clear X^1		
04	Comp. $X^1 \rightarrow X^2$		
05	Exit		
05	Half Exit		
05	Part. Add X^2 to A^1	Sign Record=1	
06	$Q^1 \rightarrow Q^2$		
07	$A^2 \rightarrow Q^1$		
07	$Q^2 \rightarrow A^1$		
09	Part. Add X^2 to A^1	Sign Record=1	

CODE	INSTRUCTION	FUNCTION
27 DVF	Divide Fractional	Divide a fractional quantity in AQ by a fractional quantity at M; store the quotient in A and the remainder in Q.

SEQUENCE: Iterative (H^6-- V^6--)

EXECUTION TIME: 63.6 us. min., 65.2 us. avg., 66.4 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Set X^1 to zeros
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
04	Init. Storage		Select M
04	Wait Storage		
04	Comp $X^1 \rightarrow X^2$		Set X^2 to 'all ones'
05	Set Sign Record FF	A neg	Record the sign of the dividend
05	Set Div. Sign Record FF	A neg.	
07	Clear R^1		Prepare R for receipt of divide step count
07	Partial Add X^2 to A^1	A neg	Complement A if AQ is negative
08	Set I^2 to 48		Select the divide step control count
10	$Q^1 \rightarrow Q^2$		
11	$A^2 \rightarrow Q^1$		Switch A and Q
11	$Q^2 \rightarrow A^1$		
12	$I^2I^3 \rightarrow R^1$		Place 48 in R
13	Clear X^1		
13	Partial Add X^2 to A^1	Sign Record=1	
14	$Q^1 \rightarrow Q^2$		
15	$A^2 \rightarrow Q^1$		Complement Q if AQ is negative
15	$Q^2 \rightarrow A^1$		



27 DVF

TIME	COMMAND	CONDITION	REMARKS
15	$I^5I^6 \rightarrow X^1$		Transfer the divisor (M) to X
16	$R^1 \rightarrow R^2$		Set $R^2 = R^1(48)$
16	$X^1 \rightarrow X^2$		Position (M) in X^2 for generating partial dividends, complement if negative.
16	Exit to Multiply Step		
17	Comp. Sign Record FF	X neg	Establish the sign of the quotient
17	Exit to Divide Step		
18	Comp. $X^1 \rightarrow X^2$	X pos.	
EXECUTE	DIVIDE STEP		
00	Red. R^1 to R^2		Perform the division. Set least significant bit in Q to '1' if $X \leq A$; to 0 if $X > A$.
00	Shift AQ Left		Shift AQ left once after comparing X to A. Reduce R one count for each shift. Conclude the division when R=0.
01	$R^2 \rightarrow R^1$		
01	$A^2 \rightarrow A^1$		
01	$Q^2 \rightarrow Q^1$		
01	Exit to 00	$R \neq 0$, $A < X$	
01	Exit to End Correction	R = 0	
05	Add X^2 to A^1	$A \geq X$	
05	Set Q_{00} to 1	$A \geq X$	
05	Exit to 00	$R \neq 0$	
05	Exit to End Correction	R = 0	

27 DVF

TIME	COMMAND	CONDITION	REMARKS
EXECUTE END CORRECTION			
00	Set Divide Fault FF	Q neg.	The quotient is initially determined as a pos. quantity; if a '1' is present in Q_{47}, a fault exists.
01	Set Part. Add in A FF		
03	Clear X^1		
04	Comp. X^1 to X^2		
05	Part. Add X^2 to A^1	Div. Sign=1	Complement remainder if dividend negative
05	Exit		
05	Half Exit		
06	$Q^1 \rightarrow Q^2$		Place quotient in A
07	$A^2 \rightarrow Q^1$		
07	$Q^2 \rightarrow A^1$		
09	Part. Add X^2 to A^1	Sign Record = 1	Complement quotient

CODE	INSTRUCTION	FUNCTION
30 FAD	Floating Add	Add two quantities packed in floating point format, one in A, one in M. Store the result in A, the residue in Q.

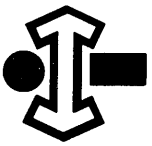
SEQUENCE: Iterative (H^6-- V^6--)

EXECUTION TIME: 11.2 us min., 18.8 us avg., 26-8 us max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \longrightarrow U^2$		Transfer m to U^2
01	Clear X^1		Set X to all zeros
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
04	Init. Storage		Select M
04	Wait Storage		
04	Comp. $X^1 \longrightarrow X^2$		
05	Part. Add X^2 to A^1	A neg	Complement A if negative, record the sign
05	Set SR FF	A neg	
07	$A^1 \longrightarrow X^1$		
08	$X^1_u \longrightarrow X^2_u$		
09	Clear A^1		
10	$A^1 \longrightarrow A^2$		
10	$X^1_{EXP.} \longrightarrow U^2$ (Extend Exp Sign)	Unconditional	Transfer the augend (A) to X, extract the exponent(X^{36}-X^{48}) and place in U^2. Clear A.
11	$U^2 \longrightarrow U^1$		
11	Clear X^1 Exp		
12	Comp. $X^1 \longrightarrow X^2$	Sign Record=1	Restore A (less exponent) to original, non-complement condition.
12	$X^1 \longrightarrow X^2$	Sign Record=0	
12	Clear SR FF		
13	Clear X^1		
13	Part. Add X^2 to A^1		

30 FAD

TIME	COMMAND	CONDITION	REMARKS
13	$A^2 \rightarrow Q^1$		Clear Q^1 and Q^2
14	$Q^1 \rightarrow Q^2$		
14	$U^2 \rightarrow R^2$		Transfer augend exponent to R^2 and comp. to R^1
15	Comp. $R^2 \rightarrow R^1$		
15	$I^5\ I^6 \rightarrow X^1$		Transfer the addend from M to X, register the sign and complement if negative
16	$X^1 \rightarrow X^2$		
16	$R^1 \rightarrow R^2$		
17	Set SR FF	X Neg	
18	Comp. $X^1 \rightarrow X^2$	X Neg	
EXECUTE FLOATING POINT			
00	Set Inhibit A^1 A^2		
01	$X^2 \rightarrow X^1$		
02	$X^1 \rightarrow U^2$ (Extend Exp)		Transfer the exponent of the addend to U^2. Clear out the exponent portion of the addend.
03	$U^2 \rightarrow U^1$		
05	Clear X^1 Exp		
06	Comp. $X^1 \rightarrow X^2$	Sign Record=1	Complement the addend if negative. Store addend in X^2
06	$X^1 \rightarrow X^2$	Sign Record=0	
07	Clear SR FF		Clear sign record
08	Add R^1 to U^2		Compare augend exponent to addend exponent
09	Clear X^1		Clear X to receive the augend
09	Set U^2 SR	U^2 Neg	
09	Set Part. Add in A FF		
10	$U^2 \rightarrow R^2$		Store exponent difference in R^2
11	Clear A^1	U^2 neg at 09	Prepare A for reversal of operands
11	$A^1 \rightarrow X^1$		Transfer augend to X^1
11	Clear U^1_{UA}	U^2 negative 08	Set U^1_{UA} to all zeros
12	$U^1 \rightarrow U^2$		Transfer addend exp. to U^2 Clear U^2 if U^2 Neg at 09



30 FAD

TIME	COMMAND	CONDITION	REMARKS
13	$X^2 \rightarrow X^1$	U^2 pos at 09	Place addend in X^1 if augend exponent < addend exponent
14	Part. Add R^1 to U^2	U^2 neg at 09	Store exponent of augend in U^2 if augend exponent > addend exponent
15	Add X^2 to A^1	U^2 neg at 09	Place addend in A if augend exponent > addend exponent
15	$R^2 \rightarrow R^1$	U^2 pos at 09	Set $R^2 = R^1$ to control the shift
15	Comp $R^2 \rightarrow R^1$	U^2 neg at 09	
16	$R^1 \rightarrow R^2$		
16	$X^1 \rightarrow X^2$		Position operand in X^2 for generation of coefficient of result
19	$A^2 \rightarrow Q^1$		
20	Init. Shift		
	$R^2 \rightarrow R^1$ Reduce R^1 to R^2 Shift one Exit to 21	 $R^2 \neq 0$ $R = 0$	Shift the coefficient in AQ right, reducing R until R = 0. This establishes two quantities with equal exponents
23	$U^2 \rightarrow U^1$		
23	Clear R^1		Set R^1 to all ones
23	$U^2 \rightarrow U^1$		
24	$R^1 \rightarrow R^2$		
25	Clear X^1		
25	Comp. $R^2 \rightarrow R^1$		
25	Add X^2 to A^1		Generate the coefficient of the result
26	Part. Add R^1 to U^2	U^2 neg at 09	Complement (U^2) if augend exponent > addend exponent

30 FAD

TIME	COMMAND	CONDITION	REMARKS
EXECUTE ROUND			
23	Set Execute Round FF	$A^{47} \neq Q^{47}$	
25	Clear X^1		Set X^2 to one or complement one.
25	Set X^2 to 1		
26	$X^1 \rightarrow X^2$	A pos.	
26	Comp. $X^1 \rightarrow X^2$	A neg.	
28	$R^1 \rightarrow R^2$		Set $R^2 = R^1$ for shift control
31	Add X^2 to A^1	$A^{47} \neq Q^{47}$	Perform round off if Q^{47} contains a one
30	Exit to Time 38	A = 0	
EXECUTE NORMALIZE			
33	Shift A Left	$A^{37}=A^{36}=A^{35}$	Position the most sig. 1 bit of the coefficient in position A^{35}. If the shift is left reduce R by one each shift and continue to shift until $A^{35}=1$. If the shift is right increase R by one each shift and comp. R.
34	Inhibit $A^1 \rightarrow A^2$	$A \neq 0$ $A^{37} \neq A^{36}$	
35	Right Shift	$A \neq 0$	
37	Comp. $R^2 \rightarrow R^1$	$A^{37} \neq A^{36}$ $A \neq 0$	
37	Clear X	$A \neq 0$	Prepare X for use as assembly register

30 FAD

TIME	COMMAND	CONDITION	REMARKS
	EXECUTE FINAL ASSEMBLY		
41	$A^1 \rightarrow X^1$		Transfer the coefficient to X
41	Set X^1 S.R. FF	FF A neg.	Record the sign of X.
41	Set Part. Add in A FF		
42	$X^1 \rightarrow X^2$	A pos.	Place the coefficient in non-complement notation.
42	Comp. $X^1 \rightarrow X^2$	A neg.	
43	$X^2 \rightarrow X^1$		
43	Full Exit		
43	Half Exit		
43	Clear A^1		Prepare A to receive the result
44	Add $R^1 \rightarrow U^2$	AQ≠0	Insert the exponent into the proper range of X.
45	$U^2 \rightarrow X^1$ exp	AQ≠0	
46	$X^1 \rightarrow X^2$	X pos. AQ≠0	Position result in X^2, complement if sign of X was neg. at time 41.
46	Comp. $X^1 \rightarrow X^2$	X Neg. AQ≠0	
47	Part. Add X^2 to A^1	AQ≠0	Place Result in A

CODE	INSTRUCTION	FUNCTION
31 FSB	Floating Subtract	Subtract two quantities packed in floating point format, one in A, one in M. Store the results in A, the residue in Q

SEQUENCE: Iterative (H^6-- V^6--)

EXECUTION TIME: 11.2 us min., 18.8 us avg., 26.8 us max.

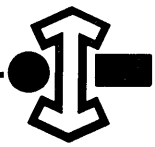
TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Set X to all zeros
04	Add R to U^2	$b \neq 0$	Modify m to M
04	Init. Storage		Select M
04	Wait Storage		
04	Comp. $X^1 \rightarrow X^2$		
05	Part. Add X^2 to A^1	A neg	Complement (A) if negative, record the sign.
05	Set SR FF	A neg	
07	$A^1 \rightarrow X^1$		
08	$X^1_u \rightarrow X^2_u$		
09	Clear A^1		
10	$A^1 \rightarrow A^2$	Unconditional	
10	$X^1 \rightarrow U^2$ (Extend Exp)		Transfer the minuend (A) to X, extract the exponent (X^{36}-X^{46}) and place in U^2. Clear A
11	$U^2 \rightarrow U^1$		
11	Clear X^1 Exp		
12	Comp. $X^1 \rightarrow X^2$	Sign Record=1	
12	$X^1 \rightarrow X^2$	Sign Record=0	Restore A (less exponent) to orignal, non-complement condition
12	Clear SR FF		
13	Clear X^1		
13	Part. Add X^2 to A^1		

31 FSB

TIME	COMMAND	CONDITION	REMARKS
13	$A^2 \rightarrow Q^1$		
14	$Q^1 \rightarrow Q^2$		Clear Q^1 and Q^2
14	$U^2 \rightarrow R^2$		Transfer minuend exponent to R^2 and comp. to R^1
15	Comp $R^2 \rightarrow R^1$		
15	I^5 $I^6 \rightarrow X^1$		Transfer the subtrahend from M to X, register the sign and complement if negative
16	$X^1 \rightarrow X^2$		
17	Set SR FF	X Neg	
18	Comp. $X^1 \rightarrow X^2$	X Neg	
EXECUTE FLOATING POINT			
01	$X^2 \rightarrow X^1$		
02	$X^1 \rightarrow U^2$ (Extend Exp)		Transfer the exponent of the subtrahend to U^2. Clear out the exponent portion of the augend.
03	$U^2 \rightarrow U^1$		
05	Clear X^1 Exp		
06	Comp. $X^1 \rightarrow X^2$	Sign Record=1	Complement the subtrahend if positive Store augend in X^2
06	$X^1 \rightarrow X^2$	Sign Record=0	
07	$X^2 \rightarrow X^1$		Set $X^1 = X^2$
07	Clear SR FF		Clear sign record
08	Comp. $X^1 \rightarrow X^2$		Set up subtraction
08	Add R^1 to U^2		Compare addend exponent to subtrahend exponent
09	Set U^2 SR	U^2 Neg.	
09	Clear X^1		Clear X to receive the minuend
09	Set Part. Add in A FF		
10	$U^2 \rightarrow R^2$		Store exponent difference in R^2 minuend
11	Clear A^1	U^2 neg at 09	Prepare A for reversal of operands
11	$A^1 \rightarrow X^1$		Transfer minuend to X^1
11	Clear U^1_{UA}	U^2 neg at 09	Set U^1_{UA} to all zeros

31 FSB

TIME	COMMAND	CONDITION	REMARKS
12	$U^1 \longrightarrow U^2$		Transfer subtrahend exp. to U^2 Clear U^2 if U^2 neg. at 09
13	$X^2 \longrightarrow X^1$	U^2 pos at 09	Place subtrahend in X^1 if minuend exponent < subtrahend exponent
14	Part. Add R^1 to U^2	U^2 neg at 09	Store exponent of addend in U^2 if minuend exponent > subtrahend exponent
15	Part. Add X^2 to A^1	U^2 neg at 09	Place subtrahend in A if minuend exponent > subtrahend exponent
15	$R^2 \longrightarrow R^1$	U^2 pos at 09	Set $R^2 = R^1$ to control the shift
15	Comp. $R^2 \longrightarrow R^1$	U^2 neg at 09	
16	$R^1 \longrightarrow R^2$		
16	$X^1 \longrightarrow X^2$		Position operand in X^2 for generation of coeffecient of result
19	$A^2 \longrightarrow Q^1$		
20	Init. Shift		
	$R^2 \longrightarrow R^1$ Reduce R^1 to R^2 Shift one Exit to 21	 $R^2 \neq 0$ $R = 0$	Shift the coefficient in A left, reducing R until R = 0. This establishes two quantities with equal exponents
23	Clear R^1		Set R^1 to all ones
23	$U^2 \longrightarrow U^1$		
24	$R^1 \longrightarrow R^2$		
25	Comp $R^2 \longrightarrow R^1$		
25	Clear X^1		
25	Add X^2 to A^1		Generate the coefficient of the result
25	Clear X^1		
26	Part. Add R^1 to U^2	U^2 neg at 09	Complement (U^2) if addend exponent > augend exponent



31 FSB

TIME	COMMAND	CONDITION	REMARKS
	EXECUTE ROUND		
23	Set Execute Round FF	$A^{47} \neq Q^{47}$	
25	Clear X^1		
25	Set X^2 to 1		
26	$X^1 \rightarrow X^2$	A pos.	Set X^2 to one or complement one
26	Comp. $X^1 \rightarrow X^2$	A neg.	
28	$R^1 \rightarrow R^2$		Set $R^2 = R^1$ for shift control
31	Add X^2 to A^1	$A^{47} \neq Q^{47}$	Perform round off if Q^{47} contains a one
30	Exit to Time 38	A=0	
	EXECUTE NORMALIZE		
33	Shift A Left	$A^{37}=A^{36}=A^{35}$	Position the most sig. 1 bit of the coefficient in position A^{35}. If the shift is left reduce R by one each shift and continue to shift until $A^{35}=1$. If the shift is right increase R by one each shift and Comp. R.
34	Inhibit $A^1 \rightarrow A^2$	$A \neq 0$	
35	Right Shift	$A^{37} \neq A^{36}$	
37	Comp. $R^2 \rightarrow R^1$	$A \neq 0$ $A^{37} \neq A^{36}$ $A \neq 0$	
37	Clear X	$A \neq 0$	Prepare X for use as assembly register

31 FSB

TIME	COMMAND	CONDITION	REMARKS
	EXECUTE FINAL ASSEMBLY		
41	$A^1 \rightarrow X^1$		Transfer the coefficient to X
41	Set X^1 S.R. FF	FF A neg.	Record the sign of X
41	Set Part. Add in A FF		
42	$X^1 \rightarrow X^2$	A pos.	Place the coefficient in non-complement notation
42	Comp. $X^1 \rightarrow X^2$	A neg.	
43	$X^2 \rightarrow X^1$		
43	Full Exit		
43	Half Exit		
43	Clear A^1		Prepare A to receive the result
44	Add R^1 to U^2	AQ≠0	Insert the exponent into the proper range of X.
45	$U^2 \rightarrow X^1$ exp	AQ≠0	
46	$X^1 \rightarrow X^2$	X pos. AQ ≠ 0	Position result in X^2, complement if sign of X was neg. at time 41.
46	Comp. $X^1 \rightarrow X^2$	X neg. AQ≠0	
47	Part. Add X^2 to A^1	AQ≠0	Place result in A

CODE	INSTRUCTION	FUNCTION
32 FMU	Floating Multiply	Multiply a number packed in floating point in A with a number, also in floating point, in M. Store the product in A, the residue in Q.

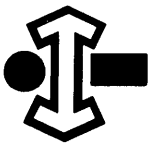
SEQUENCE: Iterative (H^6-- V^6--)

EXECUTION TIME: 3-2 us min., 36. 0 us abg., 57.2.us max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Set X to all zeros
07	Half Exit	A = 0	Leave the sequence if the multiplicand = 0
07	Exit	A = 0	
04	Add R^1 to U^2	b ≠ 0	Modify m to M
04	Init. Storage	A ≠ 0	Select M
04	Wait Storage	A ≠ 0	
04	Comp. $X^1 \rightarrow X^2$		Set X to all ones
05	Part. Add X^2 to A^1	A neg	Complement the multiplicand if negative
05	Set SR FF	A neg	Register the sign of A
07	$A^1 \rightarrow X^1$		Transfer multiplicand to X
08	$X^1_U \rightarrow X^2_U$		
08	Set I^2 to 36		Set multiply step control to 36
09	Clear A^1		Set A to all zeros
10	$X^1 \rightarrow U^2$ (Extend Exp)		Extract the exponent from the multiplicand, store the exponent in U^2 and return the multiplicand minus the exponent to A.
11	$U^2 \rightarrow U^1$		
11	Clear X^1 Exp		
12	$X^1 \rightarrow X^2$		
13	Part. Add X^2 to A^1		

32 FMU

TIME	COMMAND	CONDITION	REMARKS
13	Clear X^1		Clear X to receive the multiplier
13	$A^2 \rightarrow Q^1$		
14	$U^2 \rightarrow R^2$		Place multiplicand exponent in R^2
14	$Q^1 \rightarrow Q^2$		Transfer the multiplicand to Q^1
14	Clear A^1		
15	$A^2 \rightarrow Q^1$		
15	$R^2 \rightarrow R^1$		Set $R^1 = R^2$
15	$I^5\ I^6 \rightarrow X^1$		Transfer the multiplier from storage to X^2
16	$X^1 \rightarrow X^2$		
17	Comp. SR FF	X neg	Record the sign of the multiplier
EXECUTE FLOATING POINT			
04	Comp. $X^1 \rightarrow X^2$	X Neg	Complement the multiplier if it is negative
05	$X^2 \rightarrow X^1$		
06	$X^1 \rightarrow U^2$ (Extend Exp.)		Extract the exponent of the multiplier, Store the exponent in U^2. Retain the multiplier, less the exponent, in X^1
07	$U^2 \rightarrow U^1$		
07	Clear X^1 Exp.		
08	$X^1 \rightarrow X^2$		
12	Add R^1 to U^2		Determine the exponent of the product
13	Clear R^1		Place mult. step control quantity in R.
14	$I^2 I^3 \rightarrow R^1$		
15	Execute Mult. Step		
16	$R^1 \rightarrow R^2$		Set $R^2 = R^1$



32 FMU

TIME	COMMAND	CONDITION	REMARKS
EXECUTE MULTIPLY STEP			
00	Reduce R^1 to R^2	Short Loop	Perform the actual multiplication
00	Shift AQ right	Short Loop	Shift once to position bit in sensing position. Add (X) to A if multiplier bit is 1, then shift (AQ) right; if the multiplier bit is 0, shift AQ right. Reduce the step control count once each shift. Exit when the step control count is 0.
01	$R^2 \rightarrow R^1$	Short Loop	
01	$A^2 \rightarrow A^1$	Short Loop	
01	$Q^2 \rightarrow A^1$	Short Loop	
01	Exit to 0	$R \neq 0$ $Q = 0$ 00 $R = 0$	Long Loop
	End Correction	$R = 0$	
05	Add X^2 to A^1	$Q = 1$ 00	
05	Exit to 0	$R \neq 0$	
05	Exit to End Correction	$R = 0$	
EXECUTE END CORRECTION			
01	Set Part. Add in A FF		If the sign record flip-flop indicates a negative product in A, Q is complemented before concluding the routine.
02	$Q^1 \rightarrow Q^2$		
03	$A^2 \rightarrow Q^1$		
03	$Q^2 \rightarrow A^1$		
03	Clear X^1		
04	Comp. $X^1 \rightarrow X^2$		
05	Part. Add X^2 to A^1		
06	$Q^1 \rightarrow Q^2$		
07	$A^2 \rightarrow Q^1$		
07	$Q^2 \rightarrow A^1$		
09	Part. Add X^2 to A^1	Sign Record = 1	

32 FMU

TIME	COMMAND	CONDITION	REMARKS
EXECUTE	ROUND		
22	Clear R^1		
23	Set Execute Round FF	$A^{47} \neq Q^{47}$	
23	$U^2 \rightarrow U^1$		
24	$R^1 \rightarrow R^2$		
25	Clear X^1		
25	Comp. $R^2 \rightarrow R^1$		
25	Set X^2 to 1		
26	$X^1 \rightarrow X^2$	A pos.	Set X^2 to one or complement one
26	Comp. $X^1 \rightarrow X^2$	A neg.	
27	$U^2 \rightarrow U^1$		
28	$R^1 \rightarrow R^2$		Set $R^2 = R^1$ for shift control
31	Add X^2 to A^1	$A^{47} \neq Q^{47}$	Perform round off if Q^{47} contains a one
30	Exit to Time 38	A = 0	
EXECUTE	NORMALIZE		
33	Shift A Left	$A^{37}=A^{36}=A^{35}$	Position the most sig. 1 bit of the coefficient in position A^{35}. If the shift is left reduce R by one each shift and continue to shift until $A^{35}=1$. If the shift is right increase R by one each shift and comp. R.
		$A \neq 0$	
34	Inhibit $A^1 \rightarrow A^2$	$A^{37} \neq A^{36}$	
35	Right Shift		
37	Comp. $R^2 \rightarrow R^1$	$A^{37} \neq A^{36}$	
		$A \neq 0$	
37	Clear X	$A \neq 0$	Prepare X for use as assembly register

32 FMU

TIME	COMMAND	CONDITION	REMARKS
	EXECUTE FINAL ASSEMBLY		
41	$A^1 \rightarrow X^1$		Transfer the coefficient to X
41	Set X^1 S.R. FF	FF A Neg.	Record the sign of X
41	Set Part. Add in A FF		
42	$X^1 \rightarrow X^2$	A pos.	Place the coefficient in non-complement notation.
42	Comp. $X^1 \rightarrow X^2$	A Neg.	
43	$X^2 \rightarrow X^1$		
43	Full Exit		
43	Half Exit		
43	Clear A^1		Prepare A to receive the result
44	Add R^1 to U^2	$AQ_7 \neq 0$	Insert the exponent into the proper range of X
45	$U^2 \rightarrow X^1$ exp	$AQ_7 \neq 0$	
46	$X^1 \rightarrow X^2$	X pos. $AQ_7 \neq 0$	Position result in X^2, complement if sign of X was neg. at time 41.
46	Comp. $X^1 \rightarrow X^2$	X Neg. $AQ_7 \neq 0$	
47	Part. Add X^2 to A^1	$AQ_7 \neq 0$	Place result in A

CODE	INSTRUCTION	FUNCTION
33 FDV	Floating Divide	Divide a number packed, in floating point in A, by a number also in floating point from memory. Store the quotient in A, the residue in Q.

SEQUENCE: Iterative (H^6--V^6--)

EXECUTION TIME: 3.2 us min., 56.0 us avg., 57.2 us max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Set X to all zeros
07	Half Exit	A = 0	Leave the sequence if dividend = 0
07	Exit	A = 0	
04	Add R^1 to U^2	b ≠ 0	Modify m to M
04	Init.Storage	A ≠ 0	Select M
04	Wait Storage	A ≠ 0	
04	Comp. $X^1 \rightarrow X^2$		Set X to all ones
05	Part. Add X^2 to A^1	A neg	Complement the dividend if negative
05	Set SR FF	A neg	Register the sign of A
07	$A^1 \rightarrow X^1$		Transfer dividend to X
08	$X^1_U \rightarrow X^2_U$		
08	Set I^2 to 36		Set divide step control to 36
09	Clear A^1		Set A to all zeros
10	$X^1 \rightarrow U^2$ (Extend Exp)		
11	$U^2 \rightarrow U^1$		Extract the exponent from the dividend, store the exponent in U^2 and return the dividend less th exponent to A.
11	Clear X Exp		
12	$X^1 \rightarrow X^2$		
13	Part. Add X^2 to A^1		
13	$A^2 \rightarrow Q^1$		

33 FDV

TIME	COMMAND	CONDITION	REMARKS
13	Clear X^1		Clear X to receive the divisor
14	$U^2 \rightarrow R^2$		Place dividend exponent in R^2
14	$Q^1 \rightarrow Q^2$		
15	Comp. $R^2 \rightarrow R^1$		Prepare R for subtraction
15	$I^5\ I^6 \rightarrow X^1$		Transfer the divisor from storage to X^2
16	$X^1 \rightarrow X^2$		
16	Inhibit $A^1 \rightarrow A^2$		Prevent the normal unconditional transfer of $A^1 \rightarrow A^2$
17	Comp. SR FF	X neg	Record the sign of the divisor
	EXECUTE FLOATING POINT		
02	Shift Right		
04	Comp. $X^1 \rightarrow X^2$	X neg	Complement the divisor if it is negative
05	$X^2 \rightarrow X^1$		
06	$X^1 \rightarrow U^2$ (Extend Exp)		Extract the exponent of the divisor, store the exponent in U^2, retain the divisor, less the exponent in X^1.
07	$U^2 \rightarrow U^1$		
07	Clear X^1 Exp		
08	Comp. $X^1 \rightarrow X^2$		
12	Add R^1 to U^2		Determine the exponent of the quotient
13	Clear R^1		Place the divide step control quantity in R
14	$U^2 \rightarrow R^2$		
14	$I^2\ I^3 \rightarrow R^1$		
15	Execute Divide		
16	$R^1 \rightarrow R^2$		Set $R^2 = R^1$
18	Set U^2 SR FF		

33 FDV

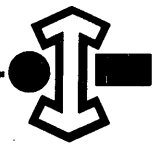
TIME	COMMAND	CONDITION	REMARKS
	EXECUTE DIVIDE STEP		
00	Red. R^1 to R^2		Perform the division. Set least significant bit in Q to '1' if $X \leq A$, to '0' if $X > A$. Shift AQ left once after comparing X to A. Reduce R one count for each shift. Conclude the division when R = 0.
00	Shift AQ Left		
01	$R^2 \rightarrow R^1$		
01	$A^2 \rightarrow A^1$		
01	$Q^2 \rightarrow Q^1$		
01	Exit to 00	$R \neq 0$, $A < X$	
01	Exit to End Correction	R = 0	
05	Add X^2 to A^1	$A \geq X$	
05	Set Q_{00} to 1	$A \geq X$	
05	Exit to 00	$R \neq 0$	
05	Exit to End Correction	R = 0	
	EXECUTE END CORRECTION		
00	Set Divide Fault	Q neg.	The quotient is initially determined as a pos. quantity; if a '1' is present in Q_{47}, a fault exists.
01	Set Part. Add in A FF		
03	Clear X^1		
04	Comp. X^1 to X^2		
05	Part. Add X^2 to A^1	Div. Sign = 1	Complement remainder if dividend negative
06	$Q^1 \rightarrow Q^2$		Place quotient in A remainder in Q
07	$A^2 \rightarrow Q^1$		
07	$Q^2 \rightarrow A^1$		
09	Part. Add X^2 to A^1	Sign record = 1	Complement quotient

33 FDV

TIME	COMMAND	CONDITION	REMARKS
EXECUTE ROUND			
22	Clear R^1		
23	Set Execute Round FF	$A^{47} \neq Q^{47}$	
23	$U^2 \rightarrow U^1$		
24	$R^1 \rightarrow R^2$		
25	Clear X^1		
25	Comp. $R^2 \rightarrow R^1$		
25	Set X^2 to 1		
26	Add R^1 to U^2		Makes exponent positive
26	$X^1 \rightarrow X^2$	A pos.	Set X^2 to one or complement one
26	Comp. $X^1 \rightarrow X^2$	A neg.	
27	$U^2 \rightarrow U^1$		
28	$R^1 \rightarrow R^2$		Set $R^2 = R^1$ for shift control
31	Add X^2 to A^1	$A^{47} \neq Q^{47}$	Perform round off if Q^{47} contains a one
30	Exit to Time 38	A=0	
EXECUTE NORMALIZE			
33	Shift A Left	$A^{37}=A^{36}=A^{35}$	Position the most sig. 1 bit of the coefficient in position A^{35}. If the shift is left, reduce R by one each shift and continue to shift until $A^{35}=1$. If the shift is right, increase R by one each shift and complement R.
34	Inhibit $A^1 \rightarrow A^2$	$A \neq 0$	
35	Right Shift	$A^{37} \neq A^{36}$	
37	Comp. $R^2 \rightarrow R^1$	$A \neq 0$	
37	Clear X	$A^{37} \neq A^{36}$ $A \neq 0$	Prepare X for use as assembly register

33 FDV

TIME	COMMAND	CONDITION	REMARKS
EXECUTE FINAL ASSEMBLY			
41	$A^1 \rightarrow X^1$		Transfer the coefficient to X
41	Set X^1 S.R. FF	FF A neg.	Record the sign of X
41	Set Part. Add in A FF		
42	$X^1 \rightarrow X^2$	A pos.	Place the coefficient in non-complement notation
42	Comp. $X^1 \rightarrow X^2$	A neg.	
43	$X^2 \rightarrow X^1$		
43	Full Exit		
43	Half Exit		
43	Clear A^1		Prepare A to receive the result
44	Add R^1 to U^2	AQ≠0	Insert the exponent into the proper range of X
45	$U^2 \rightarrow X^1$ exp	AQ≠0	
46	$X^1 \rightarrow X^2$	X pos. AQ≠0	Position result in X^2, complement if sign of X was neg. at time 41
46	Comp. $X^1 \rightarrow X^2$	X neg. AQ≠0	
47	Part. Add X^2 to A^1	AQ≠0	Place result in A



CODE	INSTRUCTION	FUNCTION
34 SCA	Scale A	Shift (A) left until the bit position to the right of sign bit contains a '1'. Store M_f in B^b.

SEQUENCE: Zero Address (H^2-- V^2--)

EXECUTION TIME: 2.8 us. min., 2.8 us. + .4 us./shift avg., 22 us. max.

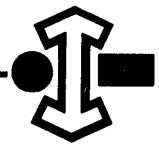
TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2 (m = shift count)
05	Clear B^b		Clear B^b to receive R_f
06	$U^2 \rightarrow R^2$		Load shift count (m) in R^2 and R^1
07	$R^2 \rightarrow R^1$	$R \neq 0$	
10	Init. Shift	$A_{47} = A_{46}$ $A \neq 0$	
10	$R^2 \rightarrow B^b$	$A_{47} \neq A_{46}$ or $A = 0$	Store M in B^b
11	Half Exit	$A_{47} \neq A_{46}$ or $A = 0$	Exit if quantity is expressed in scaled format or quantity is equal to 0
11	Full Exit		
11	Shift	$A_{47} = A_{46}$	
		$A \neq 0$	Shift (A) left until $A_{47} \neq A_{46}$, ie., until A_{46} hold most significant bit.
12	Red. R^1 to R^2		
13	$R^2 \rightarrow B^b$		Store m minus number of shifts performed in B^b
13	Half Exit	$A_{47} \neq A_{46}$ $A \neq 0$ or $R = 0$	Exit when $A_{47} \neq A_{46}$, $A \neq 0$ or when $R = 0$.
13	Full Exit		

CODE	INSTRUCTION	FUNCTION
35 SCQ	Scale AQ	Shift (AQ) left until the bit position to the right of the sign bit contains a '1'. Store M in B^b.

SEQUENCE: Zero Address (H^2-- V^2--)

EXECUTION TIME: 2.8 us. min., 2.8 us. + .4 us./ shift avg., 41.2 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2 (m = shift count)
05	Clear B^b		Clear B^b to receive R_f
06	$U^2 \rightarrow R^2$		Load shift count (m) in R^2
07	$R^2 \rightarrow R^1$	$R \neq 0$ $AQ \neq 0$	
10	Init. Shift	$A_{47} = A_{46}$	
10	$R^2 \rightarrow B^b$	$A_{47} \neq A_{46}$ or $AQ = 0$	Store M in B^b
11	Half Exit	$A_{47} \neq A_{46}$ or $AQ = 0$	Exit if quantity is already in scaled format or if quantity is equal to 0
11	Full Exit		
11	Shift	$A_{47} = A_{46}$ $AQ \neq 0$	Shift (AQ) left until $A_{46} \neq A_{47}$, ie., until A_{46} holds most significant bit.
12	Red. R^1 to R^2		
13	$R^2 \rightarrow B^b$		Store m minus number of shifts performed in B^b
13	Half Exit	$A_{47} \neq A_{46}$ $AQ \neq 0$ or	Exit when $A_{47} \neq A_{46}$, $A \neq 0$ or when $R = 0$
13	Full Exit	$R = 0$	



CODE	INSTRUCTION	FUNCTION
36 SSK	Storage Skip	Skip next instruction if (M) is negative.

SEQUENCE: Read Operand (H^3-- V^3--)

EXECUTION TIME: 6.8 us. min., (Upper Inst.), 8.8 us. avg., 16 us. max.

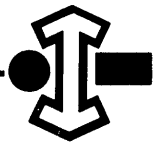
TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Prepare X^1 for receipt of (M)
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
04	Init. Storage		
10	Wait Storage		
15	$I^5I^6 \rightarrow X^1$		Load (M) to X^1
19	Half Exit	(X) pos	Perform the next instruction
19	Full Exit	(X) neg	Skip the next instruction

CODE	INSTRUCTION	FUNCTION
37 SSH	Storage Shift	Skip next instruction if (M) is negative; in either case, shift (M) left one.

SEQUENCE: Read Operand (H^3-- V^3--)

EXECUTION TIME: 10.8 us. min., 12.8 us. avg., 19.2 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Prepare X for use as exchange register
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
04	Init. Storage		
06	$Q^1 \rightarrow Q^2$		Store A_i in Q^1; clear A for use as operation register.
06	Clear A^1		
07	$A^2 \rightarrow Q^1$		
10	Wait Storage		
15	$I^5I^6 \rightarrow X^1$		Place (M) in X for transfer to A
16	$X^1 \rightarrow X^2$		
19	Wait Storage		
20	Inhibit $A^1 \rightarrow A^2$		
21	Clear X^1		
21	Part. Add X^2 to A^1		Transfer (M) to A
22	Shift A^1 to A^2		Shift (M) left
23	$A^2 \rightarrow A^1$		
25	$A^1 \rightarrow X^1$		Load (M) in X^1
27	$Q^2 \rightarrow A^1$		Place Q_i in A^1
28	$Q^1 \rightarrow Q^2$		Place A_i in Q^2



37 SSH

TIME	COMMAND	CONDITION	REMARKS
28	Init. Storage		
29	$Q^2 \rightarrow A^1$		Restore (A_i) and (Q_i)
29	$A^2 \rightarrow Q^1$		
47	$X^1 \rightarrow Z^1 Z^2$		Return (M) shifted one left to storage
47	Half Exit	No Skip	Perform next instruction (M pos)
47	x Full Exit	Skip	Skip next instruction (M neg)
	x Ordinarily this instruction is limited to the upper instruction position		

CODE	INSTRUCTION	FUNCTION
40 SST	Selective Set	Set bits of (A) to ' 1' s according to '1's of (M)

SEQUENCE: Read Operand (H^3-- V^3--)

EXECUTION TIME: 4.8 us. min., 7.2 us. avg., 9.6 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	U^1 U^2		Transfer m to U^2
01	Clear X^1		Prepare X for use as an exchange register
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
04	Init. Storage		
10	Wait Storage		
15	$I^5I^6 \rightarrow X^1$		Superimpose (M) and (A); '1's in either or both words will cause corresponding bits in the combined word to be '1'.
15	$A^1 \rightarrow X^1$		
15	Clear A^1		Clear A to receive (X)
16	$X^1 \rightarrow X^2$		Set (X^1) in X^2 for transfer to A
17	Half Exit		
17	Exit		
21	Part. Add X^2 to A^1		Transfer (X) to A

CODE	INSTRUCTION	FUNCTION
41 SCL	Selective Clear	Clear bits of (A) according to '1's of (M)

SEQUENCE: Read Operand (H^3-- V^3--)

EXECUTION TIME: 4.8 us. min., 7.2 us. avg., 9.6 us. max.

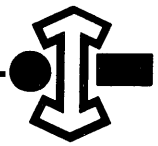
TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
04	Init. Storage		
06	Comp. $X^1 \rightarrow X^2$		Complement (A)
07	Part. Add X^2 to A^1		
10	Wait Storage		
15	$I^5 I^6 \rightarrow X^1$		Superimpose (M) and (A); '1's in either or both words will cause corresponding bits in the combined word to be '1'.
15	$A^1 \rightarrow X^1$		
15	Clear A^1		Prepare A to receive result
16	Comp. $X^1 \rightarrow X^2$		Set result to proper order in X^2; bits corresponding to '1's in (M) are now '0'.
17	Half Exit		
17	Exit		
21	Part. Add X^2 to A^1		Transfer (X^2) to A

CODE	INSTRUCTION	FUNCTION
42 SCM	Selective Complement	Complement bits of (A) according to '1's of (M)

SEQUENCE: Read Operand (H^3-- V^3--)

EXECUTION TIME: 4.8 us. min., 7.2 us. avg., 9.6 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Prepare X for use as an exchange register
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
04	Init. Storage		
10	Wait Storage		
15	$I^5I^6 \rightarrow X^1$		Transfer (M) to X^2
16	$X^1 \rightarrow X^2$		
17	Half Exit		
17	Exit		
21	Part. Add X^2 to A^1		Transfer (M) to A; '1's in M cause corresponding bits in A to be complemented.



CODE	INSTRUCTION	FUNCTION
43 SSU	Selective Substitute	Transfer bits of (M) to corresponding bits in A according to '1's of (Q)

SEQUENCE: Read Operand (H^3-- V^3--)

EXECUTION TIME: 5.2 us. min., 7.4 us. avg., 9.6 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
04	Init. Storage		
06	$Q^1 \rightarrow Q^2$		Place Q^1 in Q^2 for transfer to X
06	Comp. $X^1 \rightarrow X^2$		Complement A and X
07	Part. Add X^2 to A^1		
07	$X^2 \rightarrow X^1$		
07	LQX		Transfer Q to X^1
10	Wait Storage		
11	$A^1 \rightarrow X^1$		Superimposes A on X^1; selectively clears comp. of A_i by forcing bits to '1' if Q = 1.
11	Clear A^1		Prepare A to receive (X)
12	Comp. $X^1 \rightarrow X^2$		Put selectively cleared A_i in normal form
13	Clear X^1		Prepare X for use as an exchange register
15	$I^5I^6 \rightarrow X^1$		Masks M for '0's in Q
15	LQX		
17	Part. Add X^2 to A^1		Enter selectively cleared A_i in A^1
20	$X^1 \rightarrow X^2$		Masked M to X^2
21	Exit		
21	Half Exit		
25	Part. Add X^2 to A^1		Substitutes masked M for cleared bits of A_i

CODE	INSTRUCTION	FUNCTION
44 LDL	Load Logical	Load the logical product of (Q) and (M) in A.

SEQUENCE: Read Operand (H^3-- V^3--)

EXECUTION TIME: 5.2 us. min., 7.4 us. avg., 9.6 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Prepare X for receipt of (M)
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
04	Init. Storage		
06	$Q^1 \rightarrow Q^2$		Position (Q) for logical multiply
06	Clear A^1		Prepare A for receipt of the logical product
10	Wait Storage		
15	$I^5I^6 \rightarrow X^1$		Form the logical product of (M) and (Q)
15	LQX		
20	$X^1 \rightarrow X^2$		Load the logical product in A
21	Full or half Exit		
25	Part. Add X^2 to A^1		

CODE	INSTRUCTION	FUNCTION
45 ADL	Add Logical	Add the logical product of (Q) and (M) to A_i; store the sum in A.

SEQUENCE: Read Operand (H^3-- V^3--)

EXECUTION TIME: 5.4 us. mfn., 7.4 us. avg., 9.6 us. max.

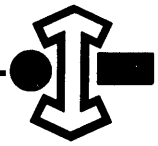
TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Prepare X for use as an exchange register
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
04	Init. Storage		
06	$Q^1 \rightarrow Q^2$		
10	Wait Storage		Form the logical product of (Q) and (M)
15	$I^5I^6 \rightarrow X^1$		
15	LQX		
20	$X^1 \rightarrow X^2$		Add LQM to A
21	Exit		
21	Half Exit		
25	Add X^2 to A^1		

CODE	INSTRUCTION	FUNCTION
46 SBL	Subtract Logical	Subtract the logical product of (Q) and (M) from A_1 store the difference in A.

SEQUENCE: Read Operand (H^3-- V^3--)

EXECUTION TIME: 5.4 us. min., 7.4 us. avg., 9.6 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Prepare X for use as an exchange register
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
04	Init. Storage		
06	$Q^1 \rightarrow Q^2$		
10	Wait Storage		Form the logical product of (Q) and (M)
15	$I^5I^6 \rightarrow X^1$		
15	LQX		
20	Comp. $X^1 \rightarrow X^2$		Subtract LQM from A
21	Exit		
21	Half Exit		
25	Add X^2 to A^1		



CODE	INSTRUCTION	FUNCTION
47 STL	Store Logical	Store the logical product of (Q) and (A) at M

SEQUENCE: Write Operand (H^4-- V^4--)

EXECUTION TIME: 4.8 us. min., 7.2 us. avg., 9.6 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Prepare X for use as an exchange register
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
04	Init. Storage		
06	$Q^1 \rightarrow Q^2$		Form the logical product of (Q) and (A)
07	$A^1 \rightarrow X^1$		
07	LQX		
07	Wait Storage		
08	Enable Full Write		
15	$X^1 \rightarrow Z^1 Z^2$		Store LQA at M
15	Exit		
15	Half Exit		

CODE	INSTRUCTION	FUNCTION
50 ENI	Enter Index *	Enter the base execution address into B^b

SEQUENCE: Zero address (H^2-- V^2--)

EXECUTION TIME: 3.2 us. min., 3.0 us. ag., 3.2 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
05	Clear B^b		Enter m in the designated B register
06	$U^2 \rightarrow R^2$		
08	$R^2 \rightarrow B^b$		
09	Half Exit		
09	Exit		

* With a b designation of 0, this instruction becomes the Pass Instruction

CODE	INSTRUCTION	FUNCTION
51 INI	Increase Index	Add the base execution address to (B^b), store the sum in B^b.

SEQUENCE: Zero Address (H^2-- V^2--)

EXECUTION TIME: 3.2 us. min., 3.0 us. avg., 3.2 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
05	Clear B^b		Store M in B^b
06	$U^2 \rightarrow R^2$		
08	$R^2 \rightarrow B^b$		
09	Half Exit		
09	Exit		

CODE	INSTRUCTION	FUNCTION
52 LIU	Load Index (Upper)	Replace (B^b) with the upper address of the word specified by the base execution address.

SEQUENCE: Read Operand (H^3-- V^3--)

EXECUTION TIME: 4.8 us min., 7.2 us avg., 9.6 us max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2.
01	Clear X^1		Prepare X for use as exchange register
04	Init. Storage		
10	Wait Storage		
14	Clear B^b		Clear B^b to receive $(m)_{UA}$
15	$I^5I^6 \rightarrow X^1$		
18	$X^1 \xrightarrow[UA]{} I^2$		
20	$I^2I^3 \rightarrow R^1$		Transfer $(m)_{UA}$ to B^b
22	$R^1 \rightarrow R^2$		
22	$R^2 \rightarrow B^b$		
23	Half Exit		
23	Exit		

CODE	INSTRUCTION	FUNCTION
53 LIL	Load Index (Lower)	Replace (B^b) with the lower address of the word specified by the base execution address

SEQUENCE: Read Operand (H^3-- V^3--)

EXECUTION TIME: 4.8 us; min., 7.2 us. avg., 9.6 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Prepare X for use as an exchange register
04	Init. Storage		
10	Wait Storage		
14	Clear B^b		Prepare B^b to receive $(m)_{LA}$
15	$I^5 I^6 \Rightarrow X^1$		
16	$X^1_{LA} \rightarrow X^2_{UA}$		Transfer (m_{LA}) to B^b
17	$X^2_{UA} \rightarrow X^1_{UA}$		
18	$X^1_{UA} \rightarrow I^2$		
20	$I^2 I^3 \Rightarrow R^1$		
22	$R^1 \Rightarrow R^2$		
22	$R^2 \rightarrow B^b$		
23	Half Exit		
23	Exit		

CODE	INSTRUCTION	FUNCTION
54 ISK	Index Skip	If (B^b) = the base execution address, skip the next instruction; if $(B^b) \neq$ the base execution address, add one to B^b.

SEQUENCE: Zero Address (H^2-- V^2--)

EXECUTION TIME: 5.6 us. min., 5.6 us. avg., 5.6 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
02	$R^1 \rightarrow R^2$		
04	Part. Add R^1 to U^2	$b \neq 0$	Subtract (B^b) from m, load the difference in R^2.
05	Comp. $R^2 \rightarrow R^1$		
05	Clear B^b		Clear B^b to receive modified (R)
06	$U^2 \rightarrow R^2$		
09	Exit *	R = 0	Proceed to next instruction step if m - (B^b) = 0
10	$R^1 \rightarrow R^2$	$R \neq 0$	
12	Reduce R^1 to R^2	$R \neq 0$	Reduce (R) by one (this increases R_i by one)
13	Comp. $R^2 \rightarrow R^1$	$R \neq 0$	Express (R) in non-complement form
14	$R^1 \rightarrow R^2$	$R \neq 0$	Load (R) in B^b
14	$R^2 \rightarrow B^b$	$R \neq 0$	
15	Half Exit	$R \neq 0$	Leave the sequence

* Ordinarily this instruction is limited to the upper instruction position.

CODE	INSTRUCTION	FUNCTION
55 IJP	Index Jump	If $(B^b) \neq 0$, reduce B^b by one, jump to m If $(B^b) = 0$, continue program

SEQUENCE: Zero Address (H^2-- V^2--)

EXECUTION TIME: 2.8 us. min., 4.4 us. avg., 4.4 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
02	$R^1 \rightarrow R^2$		$R^1 = R^2$
05	$U^2 \rightarrow P^1$	$R \neq 0$	Load m in P
05	Clear B^b		Prepare B^b to receive modified (R)
07	Jump Exit	$R \neq 0$	Leave routine, next program step located at m.
07	Exit	R = 0	Leave the sequence
07	Half Exit	R = 0	
08	Reduce R^1 to R^2	$R \neq 0$	Reduce (R) by one
08	$R^2 \rightarrow B^b$		Store modified (R)

CODE	INSTRUCTION	FUNCTION
56 SIU	Store Index (Upper)	Store (B^b) in upper address of the location specified by the base execution address.

SEQUENCE: Write Operand (H^4-- V^4--)

EXECUTION TIME: 4.8 us. min., 7.2 us. avg., 9.6 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	Init. Storage		Select m
00	Wait Storage		
00	$U^1 \rightarrow U^2$		
01	Clear X^1		Prepare X for use an an exchange register
03	Clear U^1_{UA}		
04	$U^1 \rightarrow U^2$		Clear U^2
06	Part. Add R^1 to U^2		
06	Enable Part. Write Upper		
07	$U^2 \rightarrow X^1_{LA}$		Prepare (B^b) for transfer to upper address portion of word specified by m
08	$X^1_{LA} \rightarrow X^2_{UA}$		
09	$X^2_{UA} \rightarrow X^1_{UA}$		
11	Half Exit		
11	Exit		
11	$X^1_u \rightarrow Z^1Z^2$		Store (B^b) at m

CODE	INSTRUCTION	FUNCTION
57 SIL	Store Index (Lower)	Store (B^b) in the lower address of the location specified by the base execution address

SEQUENCE: Write Operand (H^4-- V^4--)

EXECUTION TIME: 4.8 us. min., 7.2 us. avg., 9.6 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	Init. Storage		Select m
00	Wait Storage		
00	$U^1 \rightarrow U^2$		
01	Clear X^1		Prepare X for use as an exchange register
03	Clear U^1_{UA}		
04	$U^1 \rightarrow U^2$		Clear U^2
06	Part. Add R^1 to U^2		
06	Enable Part. Write Lower		
07	$U^2 \rightarrow X^1_{LA}$		Transfer B^b to lower address portion of word specified by m
11	Half Exit		
11	Exit		
11	$X^1_{LA} \rightarrow Z^1Z^2$		Store B^b at m

CODE	INSTRUCTION	FUNCTION
60 SAU	Substitute Address (Upper)	Replace the upper address of (M) with the lowest order 15-bits of (A)

SEQUENCE: Write Operand (H^4-- V^4--)

EXECUTION TIME: 4.8 us. min., 7.2 us. avg., 9.6 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Prepare X for use as an exchange register
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
04	Init. Storage		
07	$A^1 \rightarrow X^1$		Transfer (A) to X
07	Wait Storage		
08	Enable Part. Write Upper		
12	$X^1 \xrightarrow[LA]{} X^2_{UA}$		Place lowest order 15-bits of A in X^1_{UA} and write into storage
13	$X^2_{UA} \xrightarrow{} X^1_{UA}$		
15	Half Exit		
15	Exit		
15	$X^1 \xrightarrow[U]{} Z^1Z^2$		

CODE	INSTRUCTION	FUNCTION
61 SAL	Substitute Address (Lower)	Replace the lower address of (M) with the lowest order 15-bits of (A)

SEQUENCE: Write Operand (H^4-- V^4--)

EXECUTION TIME: 4.8 us. min., 7.2 us. avg., 9.6 us. max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Prepare X for use as an exchange register
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
04	Init. Storage		
07	$A^1 \rightarrow X^1$		Transfer (A) to X
07	Wait Storage		
08	Enable Part. Write Lower		Store only the lower 15-bits of (A) in storage
15	Half Exit		
15	Exit		
15	$X^1_L \rightarrow Z^1Z^2$		

CODE	INSTRUCTION	FUNCTION
62 INT	Input Transfer	Transfer (B^b) words to storage beginning at M + (B^b - 1)

SEQUENCE: Search and Transfer (H^5-- V^5--)

EXECUTION TIME: 4.8 us. min., 4.0 + 4.8r avg., 6.8 + 4.8r max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Place terminal address in U^2
01	Clear X^1		Prepare X for first word transfer
02	Set R $\neq$ 0 FF		
02	$R^1 \rightarrow R^2$		
04	Reduce R^1 to R^2	b $\neq$ 0	Determine if R = 0 before reduction and exit if condition exists. Prepare first storage address.
05	$R^2 \rightarrow R^1$		
05	Set ST Not Complete FF	R$\neq$0 + b=0	
08	Set Input Trans. Act	ST not complete	Enable input transfer of first word
09	Half Exit	ST Complete	Exit if no input transfer is to be performed
	Full Exit		
09	Clear B^b	(Input Trans. Ready)	Prepare B^b for reduced value
09	Set Wait Storage	(Input Trans. Ready)	
09	Clear Input Transfer Act		Inhibit further input transfer
09	Input Resume		
10	Add R^1 to U^2		Form first storage address
10	Init. Storage	(Input Trans. Ready)	

62 INT

TIME	COMMAND	CONDITION	REMARKS
11	Clear X		Clear X for second and succeeding words
14	$R^2 \rightarrow B^b$		Store reduced value of B
14	Set $R \neq 0$ FF		Prepare to determine R=0 condition
16	$U^1 \rightarrow U^2$		Place terminal address in U^2
16	Comp. $X^1 \rightarrow X^2$		Set X to all '1's and accept input transfer word
17	$X^2 \rightarrow X^1$		
17	$I^0 \rightarrow X^1$		
17	Set ST Not Complete	$R \neq 0$	Determine if R = 0 before reduction and exit if condition exists. Prepare second and succeeding storage addresses.
18	Reduce $R^1 \rightarrow R^2$		
19	$R^2 \rightarrow R^1$		
20	Set Input Trans. Act	(ST Not Complete)	Enable input transfer of second and succeeding words
21	$I^5 I^6 \rightarrow Z^1 Z^2$		Write input word in storage
22	Clear Wait Storage		
24	Add R^1 to U^2		Form second and succeeding storage addresses
33	Init Aux Sequence	Auxiliary Request	Enter AUX if auxiliary request exists. Halt input transfer.
29	Half Exit Full	$(b=0)+\begin{pmatrix} b\neq 0 \\ R=0 \end{pmatrix}$ -Exit=1 -Exit=0	Exit if block transfer is complete
	Return to time 08	on input transfer ready.	

CODE	INSTRUCTION	FUNCTION
63 OUT	Output Transfer	Transfer (B^b) words from storage beginning at m + (B^b - 1)

SEQUENCE: Search and Transfer (H^5-- V^5--)

EXECUTION TIME: 4.8 us min., 4.0 = 4.8r avg., 6.8 + 4.8r max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Place terminal address in U^2
01	Clear X^1		Prepare X for first word transfer
02	Set R $\neq$ 0 FF		
02	$R^1 \rightarrow R^2$		
04	Reduce R^1 to R^2	b $\neq$ 0	Determine if R=0 before reduction
05	$R^2 \rightarrow R^1$		and exit if condition exists. Prepare
05	Set ST Not Complete	(R$\neq$0) + (b=0)	first storage address.
09	Full Exit Half	ST Complete	Exit if no output transfer is to be performed
09	Set Wait Storage		
09	Clear B^b		Prepare B^b for reduced value
10	Add R^1 to U^2		Form first storage address
10	Init. Storage		Read output word from storage
11	Clear X^1		Prepare X for second and succeeding word transfers
14	$R^2 \rightarrow B^b$		Store reduced value of B
14	Set R$\neq$0 FF		Determine if R=0
16	$U^1 \rightarrow U^2$		Place terminal address in U^2

63 OUT

TIME	COMMAND	CONDITION	REMARKS
17	Set ST Not Complete	$R \neq 0$	
18	Reduce R^1 to R^2		Prepare second and succeeding storage addresses
19	$R^2 \rightarrow R^1$		
21	Set Output Trans-Active		Enable output transfer of word
21	$I^5 I^6 \rightarrow X^1$		Place output word in X
22	$X^1 \rightarrow X^2$		Position output word in X
23	$X^2 \rightarrow O^4$		Place output word in O^4
23	Clear Output Trans-Active		Stop transfer operation
24	Add R^1 to U^2		
24	Return to time 08	$(R \neq 0)$ $(b \neq 0)$	Re-enter loop to transfer second and succeeding words
25	Output Ready		
29	Half Exit Full	$(b \neq 0) + \begin{pmatrix} b \neq 0 \text{ -Exit=1} \\ R=0 \text{ -Exit=0} \end{pmatrix}$	Exit to next instruction when block transfer is complete
31	Complement $R^1 \rightarrow R^2$		Prepare quantity in R (to be substituted for B^b) to allow for re-entrance to search and transfer after auxiliary operation is complete
32	Clear B^b		
32	$R^1 \rightarrow R^2$		
33	Init. AUX	Buffer Request	Enter AUX for auxiliary operation
34	Reduce R^1 to R^2		Increase value of R by 1
35	$R^2 \rightarrow R^1$		Prepare R for second reduce operation
36	Reduce R^1 to R^2		Increase value of R by 1
37	Complement $R^2 \rightarrow R^1$		Normalize R
38	$R^1 \rightarrow R^2$		Prepare R for transmission to B^b
38	$R^2 \rightarrow B^b$		Substitute (R) for B^b

CODE	INSTRUCTION	FUNCTION
64 EQS	Equality Search	Search (B^b) words beginning with m + (B^b -1) for (M) = A. Exit.

SEQUENCE: Search and Transfer (H^5-- V^5--)

EXECUTION TIME: 3.6 us min., 4.0 + 3.6r avg., 6.8 + 3.6r max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Place terminal address in U^2
01	Clear X^1		Prepare X for first search word
02	Set R≠0 FF		
02	$R^1 \rightarrow R^2$		
04	Reduce R^1 to R^2	b ≠ 0	Determine if R=0, exit if condition exists. Prepare first storage address.
05	$R^2 \rightarrow R^1$		
05	Set ST Not Complete	(R≠0) + (b=0)	
09	Half Full Exit	ST Complete (R=0)	Exit if no search is to be made
09	Set Wait Storage		
09	Clear B^b		Prepare B^b for reduced value
10	Add R^1 to U^2		Form first storage address
10	Init. Storage		Read word to be searched
11	Clear X^1		Prepare X for second and succeeding words
14	$R^2 \rightarrow B^b$		Store reduced value of B^b
14	Set R≠0 FF		Determine if R=0 before it is reduced
16	$U^1 \rightarrow U^2$		Place terminal address in U^2

64 EQS

TIME	COMMAND	CONDITION	REMARKS
17	Set ST Not Complete	R≠0	
18	Reduce R^1 R^2		Prepare second and succeeding storage address
19	$R^2 \rightarrow R^1$		
21	$I^5 I^6 \rightarrow X^1$		Place word to be searched in X
22	Comp. $X^1 \rightarrow X^2$		Prepare word from storage for comparison with A
24	Add R^1 to U^2		Form second and succeeding storage addresses
29	Full Exit *	X = A	Search condition satisfied. Skip next instruction.
29	Half Exit	(b=0) (R=0 at T=17) + (b=0)	Search block exhausted
30	Return to time 08	(R≠0) (b≠0)	Return to loop to compare second and succeeding words
33	Init. AUX.	Auxiliary Request	Terminate search, initiate auxiliary operation.

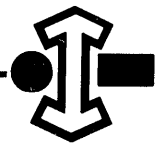
* Ordinarily this instruction is used in upper position.

CODE	INSTRUCTION	FUNCTION
65 THS	Threshold Search	Search (B^b) words, beginning with m + (B^b -1). (M) (A): Exit

SEQUENCE: Search and Transfer (H^5-- V^5--)

EXECUTION TIME: 3.6 us min., 4.0 + 3.6r avg., 6.8 + 3.6r max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Place terminal address in U^2
01	Clear X^1		Prepare X for first search word
02	Set R≠0 FF		
02	$R^1 \rightarrow R^2$		
04	Reduce R^1 to R^2	b ≠ 0	Determine if R=0, exit if condition exists. Prepare first storage address.
05	$R^2 \rightarrow R^1$		
05	Set ST Not Complete	(R≠0)+(b=0)	
09	Half Exit Full	ST Complete (R≠0)	Exit if no search is to be made
09	Set Wait Storage		
09	Clear B^b		Prepare B^b for reduced value
10	Add R^1 to U^2		Form first storage address
10	Init. Storage		Read word to be searched
11	Clear X^1		Prepare X for second and succeeding words
14	$R^2 \rightarrow B^b$		Store reduced value of B^b
14	Set R≠0 FF		Determine if R=0 before it is reduced
16	$U^1 \rightarrow U^2$		Place terminal address in U^2



65 THS

TIME	COMMAND	CONDITION	REMARKS
17	Set ST Not Complete	$R \neq 0$	
18	Reduce R^1 to R^2		Prepare second and succeeding storage address
19	$R^2 \rightarrow R^1$		
21	$I^5 I^6 \rightarrow X^1$		Place word to be searched in X
22	Complement $X^1 \rightarrow X^2$		Prepare word from storage for comparison with A
24	Add R^1 to U^2		Form second and succeeding storage addresses
29	Full Exit *	$X > A$	Search condition satisfied. Skip next instruction.
29	Half Exit	$(b \neq 0)(R=0)+(b=0)$	Search block exhausted
30	Return to time 08	$(R \neq 0)$ $(b \neq 0)$	Return to loop to compare second and succeeding words
33	Init. AUX.	Auxiliary Request	Terminate search, initiate auxiliary operation.

* Ordinarily this instruction is used in upper position.

CODE	INSTRUCTION	FUNCTION
66 MEQ	Masked Equality	Search (B^b) words, beginning with M + (B^b - 1) L(Q)(M) = (A): Exit

SEQUENCE: Search and Transfer (H^5-- V^5--)

EXECUTION TIME: 3.6 us min., 4.0 + 3.6r avg., 6.8 + 3.6r max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Place terminal address in U^2
01	Clear X^1		Prepare X for first search word
02	Set R≠0 FF		
02	$R^1 \rightarrow R^2$		
04	Reduce R^1 to R^2	b ≠ 0	Determine if R=0, exit if condition exists
05	$R^2 \rightarrow R^1$		Prepare first storage address.
05	Set ST Not Complete	(R≠0)+(b=0)	
08	$Q^1 \rightarrow Q^2$	66	Position mask in Q
09	Half / Full Exit	ST Complete (R=0)	Exit if no search is to be made
09	Set Wait Storage		
09	Clear B^b		Prepare B^b for reduced value
10	Add R^1 to U^2		Form first storage address
10	Init. Storage		Read word to be searched
11	Clear X^1		Prepare X for second and succeeding words
14	$R^2 \rightarrow B^b$		Store reduced value of B^b
14	Set R≠0 FF		Determine if R=0 before it is reduced
16	$U^1 \rightarrow U^2$		Place terminal address in U^2

66 MEQ

TIME	COMMAND	CONDITION	REMARKS
17	Set ST Not Complete	R≠0	
18	Reduce $R^1 \rightarrow R^2$		Prepare second and succeeding storage address
19	$R^2 \rightarrow R^1$		
21	$I^5I^6 \rightarrow X^1$		Place word to be searched in X
21	LQX		Logical add Q + X
24	Add R^1 to U^2		Form second and succeeding storage addresses
24	Comp. $X^1 \rightarrow X^2$		Prepare masked word for comparison with A
29	Full Exit *	X > A	Search condition satisfied, skip next instruction.
29	Half Exit	(b≠0)(R=0)+(b=0)	Search block exhausted
30	Return to time 08	(R≠0)(b≠0)	Return to loop to compare second and succeeding words
33	Initiate AUX	Auxiliary Request	Terminate search, initiate auxiliary operation.

* Ordinarily this instruction is limited to the upper position.

CODE	INSTRUCTION	FUNCTION
67 MTH	Masked Threshold	Search (B^b) words, beginning with M + (B^b - 1) L(Q)(M) (A): Exit

SEQUENCE: Search and Transfer (H^{500} V^{500})

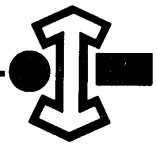
EXECUTION TIME: 3.6 us min., 4.0 + 3.6r avg., 6.8 + 3.6r max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Place terminal address in U^2
01	Clear X^1		Prepare X for first search word
02	Set R≠0 FF		
02	$R^1 \rightarrow R^2$		
04	Reduce R^1 to R^2	b ≠ 0	Determine if R=0, exit if condition exists. Prepare first storage address.
05	$R^2 \rightarrow R^1$		
05	Set ST Not Complete	(R≠0)+(b=0)	
08	$Q^1 \rightarrow Q^2$		Position mask in Q
09	Half Exit Full	ST Complete (R=0)	Exit if no search is to be made
09	Set Wait Storage		
09	Clear B^b		Prepare B^b for reduced value
10	Add R^1 to U^2		Form first storage address
10	Init.Storage		Read word to be searched
11	Clear X^1		Prepare X for second and succeeding words
14	$R^2 \rightarrow B^b$		Store reduced value of B^b
14	Set R≠0 FF		Determine if R=0 before it is reduced
16	$U^1 \rightarrow U^2$		Place terminal address in U^2

67 MTH

TIME	COMMAND	CONDITION	REMARKS
17	Set ST Not Complete	$R \neq 0$	
18	Reduce R^1 to R^2		Prepare second and succeeding storage addresses
19	$R^2 \rightarrow R^1$		
21	$I^5 I^6 \rightarrow X^1$		Place word to be searched in X
21	LQX		Logical add Q + X
24	Add R^1 to U^2		Form second and succeeding storage addresses
24	Comp. $X^1 \rightarrow X^2$		Prepare masked word for comparison with A
29	Half Exit	$(b \neq 0)(R=0)+(b=0)$	Search block exhausted
29	Full Exit *	$X > A$	Search condition satisfied; skip next instruction.
30	Return to time 08	$(R \neq 0)(b \neq 0)$	Return to loop to compare second and succeeding words
33	Initiate AUX	Auxiliary Request	Terminate search, initiate auxiliary operation.

* Ordinarily this instruction is limited to the upper position.



CODE	INSTRUCTION	FUNCTION
70 RAD	Replace Add	Store the sum of (M) and (A) at M and in A.

SEQUENCE: Read Operand (H^3-- V^3--)

EXECUTION TIME: 10.2 us min., 13.2 us avg., 16.0 us max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Replace X for use as exchange register
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
04	Init. Storage		
10	Wait Storage		
15	$I^5I^6 \rightarrow X^1$		
16	$X^1 \rightarrow X^2$		
21	Clear X^1		Add (M) and (A), store the sum in memory. The sum which is generated in A is not destroyed in A.
21	Add X^2 to A^1		
25	$A^1 \rightarrow X^1$		
28	Init. Storage		
19	Wait Storage		
47	$X^1 \rightarrow Z^1Z^2$		
47	Half Exit		
47	Exit		

CODE	INSTRUCTION	FUNCTION
71 RSB	Replace Subtract	Store the difference of $(M)_i$ and A at M and in A

SEQUENCE: Read Operand (H^3-- V^3--)

EXECUTION TIME: 10.2 us min., 13.2 us avg., 16.0 us max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Prepare X for use as exchange register.
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
04	Init. Storage		
06	Comp. $X^1 \rightarrow X^2$		
07	Part.Add X^2 to A^1		
10	Wait Storage		Subtract (A) from $(M)_i$ by complementing (A) and adding. Store the difference at M. The difference in A is not destroyed.
15	$I^5I^6 \rightarrow X^1$		
16	$X^1 \rightarrow X^2$		
21	Clear X^1		
21	Add X^2 to A^1		
25	$A^1 \rightarrow X^1$		
28	Init. Storage (Write)		
19	Wait Storage		
47	$X^1 \rightarrow Z^1Z^2$		
47	Half Exit		
47	Exit		

CODE	INSTRUCTION	FUNCTION
72 RAO	Replace Add One	Store the sum of $(M)_i$ and one at M and in A.

SEQUENCE: Read Operand (H^3-- V^3--)

EXECUTION TIME: 10.2 us min., 13.2 us avg., 16.0 us max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Clear X to all zeros
03	Set X^2 to one		
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
04	Init. Storage		
06	$X^1 \rightarrow X^2$		Place a '1' in A
06	Clear A^1		
10	Wait Storage		
11	Part. Add X^2 to A^1		
13	Clear X^1		Add $(M)_i$ to the '1' in A; store the sum at M. The sum in A is not destroyed.
15	$I^5I^6 \rightarrow X^1$		
16	$X^1 \rightarrow X^2$		
19	Wait Storage		
21	Clear X^1		
21	Add X^2 to A^1		
25	$A^1 \rightarrow X^1$		
28	Init. Storage (Write)		
47	$X^1 \rightarrow Z^1Z^2$		
47	Half Exit		
47	Exit		

CODE	INSTRUCTION	FUNCTION
73 RSO	Replace Subtract One	Store $(M)_i$ less one at M and in A.

SEQUENCE: Read Operand (H^3-- V^3--)

EXECUTION TIME: 10.2 us min., 13.2 us avg., 16.0 us max.

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Clear X to all zeros
03	Set X^2 to one		
04	Add R^1 to U^2	$b \neq 0$	Modify m to M
04	Init. Storage		
06	Comp. $X^1 \rightarrow X^2$		Place a 'complement 1' in A
06	Clear A^1		
10	Wait Storage		
11	Part. Add X^2 to A^1		
13	Clear X^1		Subtract a '1' from $(M)_i$ by adding complement 1. The difference in A is not destroyed.
15	$I^5I^6 \rightarrow X^1$		
16	$X^1 \rightarrow X^2$		
19	Wait Storage		
21	Clear X^1		
21	Add X^2 to A^1		
25	$A^1 \rightarrow X^1$		Store $(M)_i$ less one at M
28	Init. Storage (Write)		
47	$X^1 \rightarrow Z^1Z^2$		
47	Half Exit		
47	Exit		

CODE	INSTRUCTION	FUNCTION
EXF 0 or 7	74.0 or 74.7	Create or Sense Specific Conditions within External Equipments

SEQUENCE: External Function

EXECUTION TIME:

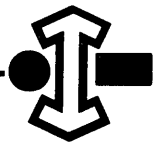
TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Prepare X for external function code
01	b to Aux Ref Desig		Prepare gating conditions for commands
03	Extend U^2 in X		Place external function code in X
04	$X^1 \rightarrow X^2$	74.0 or 74.7	Place external function code in 0^0
05	Set Select FF	74.0	Set external function counter to time
05	Set Sense FF	74.7	
24	Function or sense Ready		
63	Half Full Exit		Exit to next instruction

CODE	INSTRUCTION	FUNCTION
EXF	74.1 - 74.6	Activate Buffer Channel

SEQUENCE: External Function

EXECUTION TIME:

TIME	COMMAND	CONDITION	REMARKS
00	$U^1 \rightarrow U^2$		Transfer m to U^2
01	Clear X^1		Prepare X for initial address
01	Set f=74		Prepare gating conditions for external function commands
02	b to Aux Ref Designator		
02	$P^2 \rightarrow P^1$	74.1 - 74.6	Advance P to next instruction address
02	Initiate Storage	74.1 - 74.6	
03	$U^2 \rightarrow X^1$		Place initial address in X
05	Set Wait Storage	74.1 - 74.6	
06	Comp. Exit FF	74.1 - 74.6	Prepare proper exit condition to next instruction
10	$X^1_L \rightarrow X^2_U$	Not Adv Clk	Position initial address in X^1_U
11	$X^2 \rightarrow X^1$	Not Adv Clk	
13	$X^1 \rightarrow I^2$		Send initial address to I^2
13	$I^5 I^6 \rightarrow X^1$		
14	$X^1_L \rightarrow X^2_U$	74.1 - 74.6	Position terminal address in X^1_U
15	$X^2 \rightarrow X^1$	74.1 - 74.6	
17	Clear R^1		Prepare R register



74.1 - 74.6

TIME	COMMAND	CONDITION	REMARKS
18	$I^2I^3 \rightarrow R^1$		Place initial address in R
20	Part. Add R^1 to U^2		Toggles initial and terminal addresses
22	$U^2 \rightarrow R^2$		Place comparison of initial and terminal address in R
23	Half Jump Exit	74.1 - 74.6	Exit to next instruction
23	Set / Clear } R≠0 FF	R≠0 R = 0	
25	Set / Clear } Buffer Act FF	R ≠ 0 R = 0	Prepare buffer request

CODE	INSTRUCTION	FUNCTION
75	Selective Jump	Cause a jump to occur in the program, the jump is conditioned by the state of b and the setting of the STOP keys on the console.

SEQUENCE: b = 0-3 Normal Jump
b = 4-7 Write Operand

EXECUTION TIME: 3.0 us min., 7.2 us avg., 11.6 us max.

TIME	COMMAND	CONDITION	REMARKS
Normal	Jump		
00	$U^1 \rightarrow U^2$		
03	Jump Exit	Jump Satisfied	
03	Full Exit	Jump Not Satisfied	
03	Half Exit	Jump Not Satisfied	
03	$U^2 \rightarrow P^1$	Jump Satisfied	Insert next instruction address into P
Write	Operand		
00	$U^1 \rightarrow U^2$		
04	Init. Storage		
06	Adv. P^1 to P^2	Jump Satisfied	Determine next address of current routine (P_i)
07	Wait Storage		
07	Full Exit	Jump Not Satisfied	
07	Half Exit		
08	$P^1 \rightarrow X^2_{LA}$		Position P_i in X^2_{LA} for transfer to X^1_{UA}
08	$U^2 \rightarrow P^1$	Jump Satisfied	Transfer m to U^2 to select next instruction word
08	Set Return Jump FF		Conditions later commands
08	Enable Part. Write Upper		

75

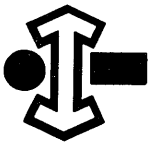
TIME	COMMAND	CONDITION	REMARKS
09	$X^2 \rightarrow X^1$		
11	Clear U^1		
12	$X^1_{LA} \rightarrow X^2_U$		Place P_i in X^1_{UA}
13	$X^2_U \rightarrow X^1_U$		
15	$I^5\ I^6 \rightarrow U^1$		Transfer next instruction to U^1
15	Half Exit		
15	$X^1_u \rightarrow Z^1Z^2$		Write return address (P_i) into storage

CODE	INSTRUCTION	FUNCTION
76	Selective Stop	Cause a stop to occur in the program, the action of the computer when the program is resumed is controlled by the condition of b and the setting of the STOP keys on the console.

SEQUENCE: b = 0-3 Normal Jump
b = 4-7 Write Operand

EXECUTION TIME: 3.0 us min., 7.2 us avg., 11.6 us max.

TIME	COMMAND	CONDITION	REMARKS
01	Set Stop II FF	b = 0,4	
		b = 1-3	With Stop Key 1-3 Set
		b = 5-7	With Stop Key 1-3 Set
00	$U^1 \rightarrow U^2$		
01	Stop operation to await manual intervention	Stop II-FF=1	
	Operator reinitiates instruction with a start or step pulse and Stop II FF is cleared at Time 09 (RNI Sequence)		
	Normal Jump Sequence		Normal Jump
03	Jump Exit		
03	$U^2 \rightarrow P^1$		Place next instruction address in P
	Write Operand Sequence		Return Jump
04	Init. Storage		
06	Adv. P^1 to P^2		Determine next address of current routine (P_1)
07	Wait Storage		
08	$U^2 \rightarrow P^1$		Transfer m to P to select next instruction word
08	$P^1 \rightarrow X^2_L$		
08	Enable Part. Write Upper		



76

TIME	COMMAND	CONDITION	REMARKS
08	Set Return Jump FF		Conditions later commands
09	$X^2 \rightarrow X^1$		Set $X^1 = X^2$
11	Clear U^1		Prepare U^1 for next instruction
12	$X^1_{LA} \rightarrow X^2_{UA}$		Store P_i
13	$X^2_U \rightarrow X^1_U$		
15	$X^1_U \rightarrow Z^1Z^2$		
15	$I^5\ I^6 \Rightarrow U^1$		Transfer next instruction to U^1
15	Half Exit		

CODE	INSTRUCTION	FUNCTION
	Interrupt	Halt main computer program for recognition of action demand by an external equipment.

SEQUENCE: AUX (INTERRUPT)

EXECUTION TIME: 3.2 usec.

TIME	COMMAND	CONDITION	REMARKS
00	Initiate Storage		Enable storage reference
01	Clear Trans. Act. FF		Computer will not recognize transfers
03	Wait Storage		Enable storage reference
04	Enable Partial Write Upper		Delay writing content of P in upper portion of 00007 until interrupt control word is in I^5I^6.
04	$P^1 \rightarrow X^2_L$		
04	Interrupt Exit		Prepare proper exit from sequence
05	Set Interrupt Lockout		Exclude recognition of further interrupt signals
05	Clear U^1_L		Prepare U for interrupt control word
06	Set P to 00007		Set P to interrupt control word address
07	$X^2 \rightarrow X^1$		Move contents of P to X
08	$X^1_L \rightarrow X^2_U$		
09	$X^2_U \rightarrow X^1_U$		Position P in X preparatory to storage in upper portion of 00007.
11	$I^5 I^6 \rightarrow U^1$		Take interrupt control word from storage
11	$X^1 \rightarrow Z^1Z^2$		Store P in upper address portion of 00007
15	Half Exit		Exit to first instruction of interrupt routine

CODE	INSTRUCTION	FUNCTION
	Advance Clock	Advance real time count which is retained in special storage address 00000.

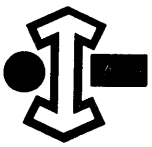
SEQUENCE: AUX (ADVANCE CLOCK)

EXECUTION TIME: 8.8 usec.

TIME	COMMAND	CONDITION	REMARKS
00	Init. Storage		Prepare to read control word 00000 from storage
01	Clear Trans. Act. FF		Computer will not recognize transfer
01	Rank 1 Scanner to Aux. Ref. Desig.		
02	Set S^1 to 0		Prepare storage to read special address
03	Wait Storage		
05	Set X^2 to 1		Prepare the increment to clock value
05	Clear X^1		Prepare X for unincremented clock value
05	Clear R^1		
06	$X^1 \rightarrow X^2$		Position clock increment preparatory to storage
06	$Q^1 \rightarrow Q^2$		Store contents of Q from previous instruction
07	$A^2 \rightarrow Q^1$		Store contents of A from previous instruction
07	Clear A^1		Prepare A for advance clock operation
11	Part. Add X^2 to A^1		Set lowest bit of A to '1'
11	$I^5 I^6 \rightarrow X^1$		Position unincremented clock value in X
12	$X^1 \rightarrow X^2$		Place unincremented clock value in both ranks of X
13	Start Scanner		
17	Clear X^1		Clear one rank of X
17	Add X^2 to A^1		Add unincremented clock value to increment
19	$A^1 \rightarrow X^1$		Prepare incremented clock value for storage
20	Init. Storage		Prepare to write incremented clock value in storage
20	Wait Storage		

Adv. Clk

TIME	COMMAND	CONDITION	REMARKS
20	Clear AUX REQ. FF		Computer can recognize next auxiliary request
21	$Q^2 \rightarrow A^1$		
22	$Q^1 \rightarrow Q^2$		Restore original contents of A and Q registers
23	$A^2 \rightarrow Q^1$		
23	$Q^2 \rightarrow A^1$		
43	Half Jump EXIT		Exit to next instruction in main program



CODE	INSTRUCTION	FUNCTION
	Buffer	Exchange of one input or output word via buffer channels

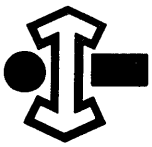
SEQUENCE: AUXILIARY

EXECUTION TIME: 10.8 usec. - 17.2 usec.

TIME	COMMAND	CONDITION	REMARKS
00	Init. Storage		Prepare storage to read control word
01	Set Aux. Desig.		Set storage to control word
01	Clear Tran. Act.		Disable transfer operations
03	Wait Storage-1		
05	Clear X^1		
05	Clear R^1		Prepare R for determination of amended control word
07	Clear U^1_U		Prepare U for initial address
09	Storage Resume		
11	$I^5I^6 \rightarrow U^1_U$		Place initial address in U^1
11	$I^5I^6 \rightarrow X^1$		Place terminal word in X^1
12	$U^1 \rightarrow U^2$		Place initial address in U^2
12	Init. Storage		Prepare storage to: read output word write input word
17	Clear X^1		Set X to all '0's
18	Comp. $X^1 \rightarrow X^2$	INPUT BUF	Set X to all '1's
19	$X^2 \rightarrow X^1$	INPUT BUF	Place '1's in both ranks of X

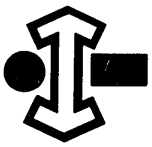
BUFFER

TIME	COMMAND	CONDITION	REMARKS
20	$U^2 \rightarrow R^2$		Place initial address in R
20	$I^0 \rightarrow X^1$	INPUT BUF	Place input word in X
20	Wait Storage -2		
21	Storage Resume		
23	I^5 $I^6 \rightarrow X^1$	OUTPUT BUF	Place output word in X
23	Clear $I^0 \rightarrow X^1$		Disable sample of input buffer lines
23	Set ADV CLK REQ		Prepare to recognize next advance clock request
24	$X^1 \rightarrow X^2$	OUTPUT BUF	Place output word in both ranks of X
25	$X^2 \rightarrow$ 0-		
25	Set $I^1 \rightarrow$ 0-		
25	Comp. $R^2 \rightarrow R^1$		Complement initial address
25	Clear U^1_u		Prepare U^1 for new initial address
26	$R^1 \rightarrow R^2$		Place complement of initial address in both ranks of R
24	Wait Storage -3		
28	Reduce R^1 to R^2		
28	Init. Storage	INPUT BUF OUTPUT BUF	Prepare storage to replace control word
29	Comp. $R^2 \rightarrow R^1$		Normalize initial address
31	Set Part. Add in U^2		Prepare to write terminal address in U^2
32	$U^1 \rightarrow U^2$		Place initial address in U^2
33	Clear X^1		Prepare X for control word



BUFFER

TIME	COMMAND	CONDITION	REMARKS
34	Add R^1 to U^2		Place initial address in U^2
35	$U^2 \rightarrow X^1$ (With extension)		Place initial address in X
39	$X^1_L \rightarrow X^2_U$		
40	$X^2_U \rightarrow X^1_U$		
41	Storage Resume		
42	Clear Ready Resume FF	$f \neq 74$	Prepare for recognition of next buffer request
43	$U^2 \rightarrow U^1$		Position initial address in U^1
43	$I^5I^6 \rightarrow X^1$		
43	Clear Request FF		Prepare to recognize next interrupt request
43	$X^1_U \rightarrow I^2$		
44	Clear Buffer Request FF		Prepare for recognition of next buffer request
44	$X^1_L \rightarrow X^2_U$		Position initial address in X upper
45	$X^2_U \rightarrow X^1_U$		
47	Clear R^1		
48	$I^2 \rightarrow R^1$		Place initial address in R
50	Part. Add $R^1 \rightarrow U^2$		Place initial address in U
52	$U^2 \rightarrow R^2$		Compare initial and terminal addresses
53	Set $R \neq 0$		Determine equality of initial and terminal addresses
53	Half Jump Exit		Return to main program



CHAPTER 3

CONSOLE INPUT-OUTPUT EQUIPMENT

Maintenance is performed on the console input-output equipments (punch, reader and typewriter) and on the control and data circuits associated with each unit. The manufacturers' manuals provide the required maintenance procedures for each unit. These manuals, which are contained in a packet furnished with the computer, are:

punch	Description, Adjustments and Lubrication Teletype Bulletin 215B Parts Catalog, Teletype Bulletin 1154B
reader	Ferranti High-Speed Tape Reader Type TR5 Technical Manual List No. E.P. 9
typewriter	Adjustment and Lubrication Procedures for Decoder and Power Unit of the Computeriter

The section on modifications (immediately following) should also be consulted before performing maintenance.

Maintenance of the control and data circuits for these units is similar to that for circuits of the main computer.

Test routines are available for checking the operation of the console equipments.

ADDITIONAL TYPEWRITER PROCEDURES

Maintenance procedures in the manual for the decoder and power unit are to be supplemented with those listed below. It is recommended that adjustments be made on-line unless an off-line checker is available to simulate actual operating conditions.

LUBRICATION

Normally lubrication should take place after 100 hours of operation. Apply a heavy gear grease to all points where metal rubs on metal, for example, at the permutation bars where they are pulled by the arms of the rotary solenoids. Apply a light oil to all springs and pivot points.

POWER CAM UNIT

If acceleration of power cam (Soroban diagram D-5022) is sluggish, replace accelerator spring. If cam is hanging up on anti-repeat lug or trip lug, remove the power unit from the typewriter and manually energize the magnet (TCM), watch complete cycle of operation.

TRANSLATOR BAIL ASSEMBLY

If translator bail is not setting properly, check drive crank spring; if it is worn replace it. Shock is present each time the carriage is returned. After continued use the bail assembly and translator may need readjusting. Check adjustment every 100 hours or when a malfunction occurs.

MODIFICATIONS

READER

Modifications of the reader consist chiefly of removing plugs A and B as well as the four printed circuit boards. In their stead connector J20701 and the heavy-lined circuits of figure 3-1 are installed. For detailed diagrams of the reader circuits after modification see pages 73 and 74, volume 5.

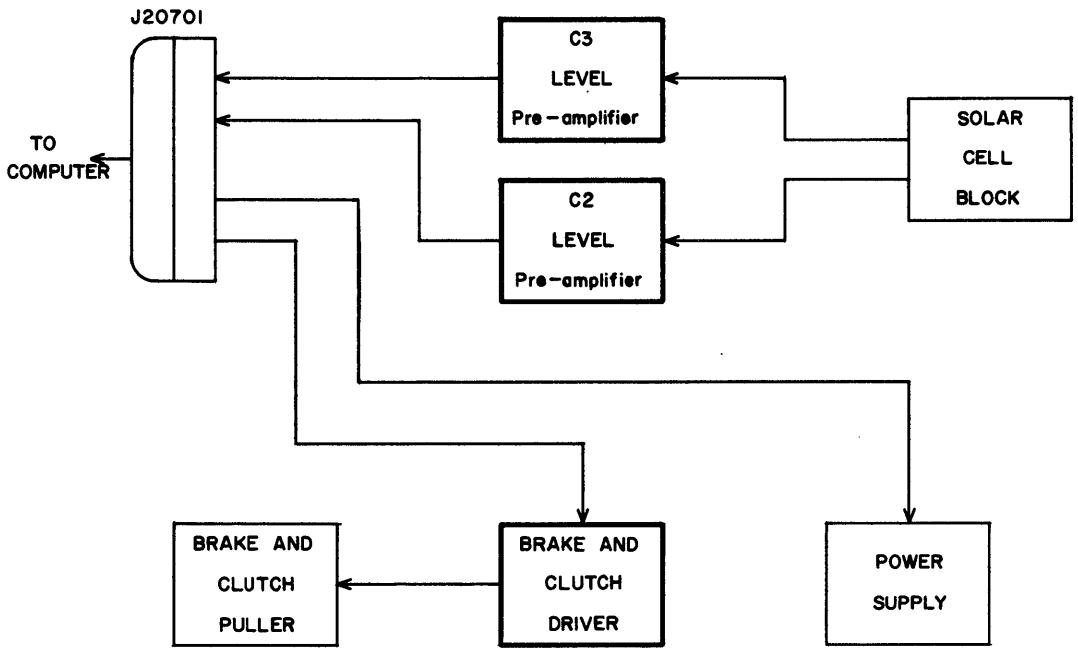
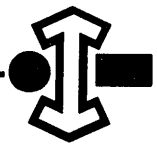


Figure 3-1. PT Reader Modifications.



PUNCH

Physical modifications of the punch consist of:

1) Removal of the On-Off switch - The Punch Motor switch on the reader-punch control panel replaces the On-Off switch on the punch itself.
2) Addition of the Out-of-Tape microswitch - The Out-of-Tape switch provides a means of monitoring the tape supply reel.
3) Removal of the chad drawer - The built-in chad bin in the console replaces the chad drawer originally supplied with the punch.

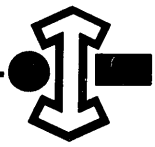
Electrical modifications to the punch consist of:

1) The punch magnet coils are rewound to allow energizing with -15 volts instead of the -90 volts normally required.
2) The -15 volt and the ground connections at the connector are interchanged so that pin R carries -15 volts and pin S carries ground.

The electrical modifications are indicated on page 71, volume 5.

TYPEWRITER

A decoder and a coder enable the IBM electric typewriter to communicate with the computer. The necessary modifications are described in the Soroban manual included in the packet of manufacturers' manuals.



CHAPTER 4

POWER SYSTEM

Maintenance of the power system involves checking for proper output levels and occasional replacement of fuses. The system and associated protective circuits are described in chapter 7 of volume 2. Fuse locations in the various cabinets are listed in table 4-1.

MOTOR GENERATOR SET

The 400-cycle power for the computer system is furnished by a brushless motor generator (MG) set. A manual provided by the manufacturer (Electric Machinery Mfg. Co.) is included in a separate packet.

The manual motor switch on the MG control cabinet remains on as normally the MG set is turned on and off remotely by the Power switch at the console.

Preventive maintenance steps:

1) Check the voltage output at the control cabinet for a value of approximately 208 volts. The voltage adjust control should be used to obtain proper output level only when it is certain that improper output is not due to a malfunction.
2) Check the current output for a value of 13.5 - 15.0 amperes.
3) Check frequency of output for indication of 410 - 420 cps.
4) Replace the 2 pre-lubricated bearings on the MG set once a year.

A sharp jolt can occasionally cause the exciter field of the MG set to lose its residual magnetism. As a result the MG set fails to develop output voltage. The required residual magnetism can be restored by flashing the exciter field. This is done by connecting a 3-volt battery (two 1 1/2-volt dry cells) to the exciter field.

The exciter field is most conveniently accessible at terminals 4 and 5 of the terminal board on the regulator panel in the rear of the control cabinet (figure 4-1). The battery is connected as shown in figure 4-2. Remove the lead from terminal 5 and connect it to the positive battery terminal. Connect the negative battery terminal to terminal 5. Now run the MG set to develop voltage. If voltage fails to develop, stop the MG and reverse leads to the battery.

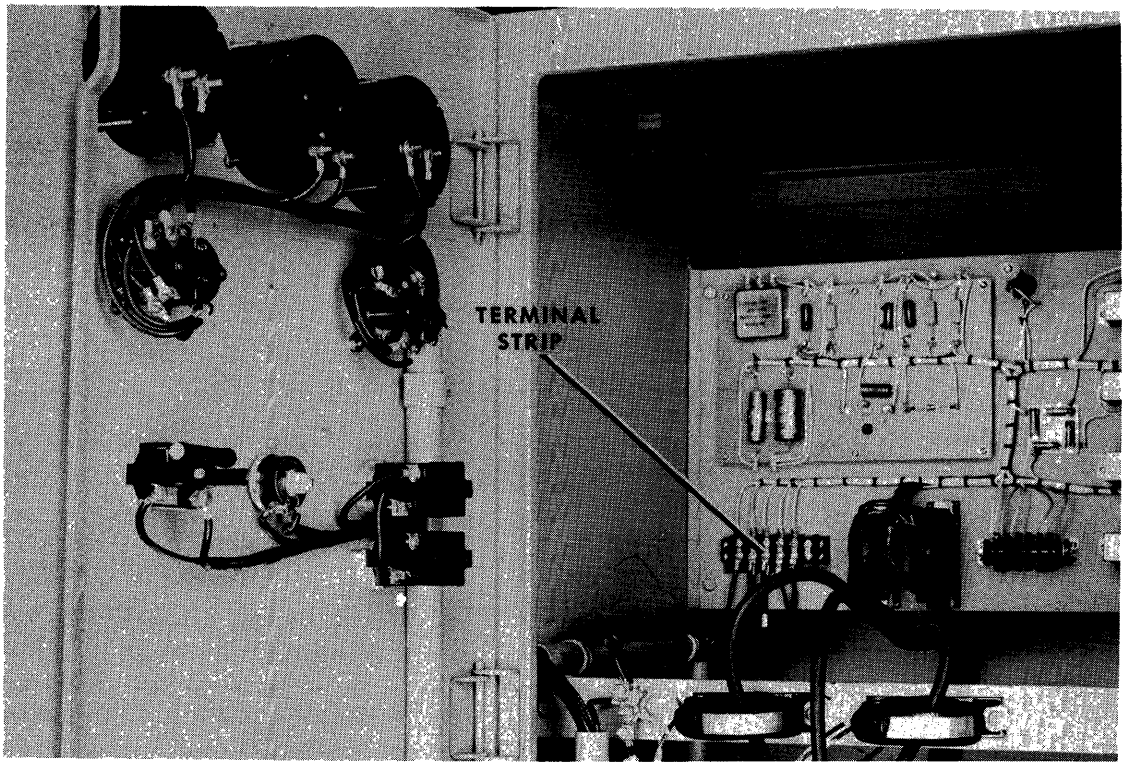


Figure 4-1. Regulator Panel

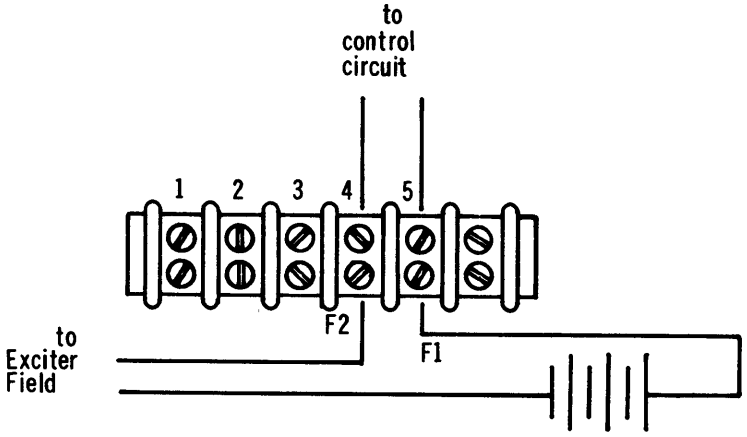


Figure 4-2. Circuit for Flashing MG Exciter

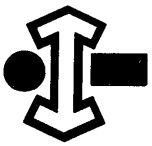


TABLE 4-1. CIRCUIT BREAKER AND FUSE LOCATION

MAIN CABINET			
Number	Protects	Number	Protects
F11 F12 F13	chassis 1	F51 F52 F53	chassis 5
F21 F22 F23	chassis 2	F61 F62 F63	chassis 6
F31 F32 F33	chassis 3	F71 F72 F73	chassis 7
F41 F42 F43	chassis 4	F81 F82 F83	chassis 8
Rating 1.5A 208-vac, 3-phase input			

CONSOLE		
Number	Rating	Protects
F01 F02 F03	2.0A 2.0A 2.0A	208-vac, 400-cps, 3-phase input (light modules, relays)
F04	8.0A	120-vac, 60-cps (punch, reader, typewriter, outlets)

400 CYCLE SWITCH PANEL		
Number	Rating	Protects
CB101 CB102	20A 5A	208-vac, 400-cps, 3-phase input to all computer chassis

60 CYCLE SWITCH PANEL		
Number	Rating	Protects
CB202	20A	computer (outlets, fans, etc.)
CB203	15A	console (outlets, punch, reader typewriter)

APPENDIX A

INSTALLATION

The standard 1604 Installation manual is included in the following pages to provide supplementary maintenance information.

Typical 1604 Computer Installation

CONTENTS

FIGURES

TABLES

INSTALLATION OF THE 1604 COMPUTER SYSTEM

The Control Data 1604 and 160 systems are designed to be used with a minimum of environmental restrictions. This manual, which will be furnished to the customer well in advance of shipment of the computer system, provides electrical and physical information to aid in the preparation of a suitable site for the system. Detailed data on equipment sizes, power requirements and cables are included.

Two months before the system is shipped, a detailed floor layout should be submitted to Control Data Corporation so that cable requirements may be determined. One month before shipment, the Control Data Corporation engineer responsible for delivery and installation of equipment will visit the site to discuss unloading of the equipment from the carrier and placing it in the computer area. The general area requirements will be reviewed at this time and any final modifications agreed upon.

GENERAL REQUIREMENTS

FLOOR

The weight of the cabinet is distributed over its entire base, causing a load no greater than 150 pounds per square foot. The leveling pads in each cabinet are not normally used to support the cabinet, but are provided to level the cabinet on an uneven floor. If leveling pads are used, the floor must be able to withstand the concentrated load thus created.

Cables connecting the cabinets in the computer system are run beneath the floor and enter the cabinets through openings in the bottom of each cabinet. To permit passage of the cables, raceways may be built into the floor, or a false floor may be laid above the room floor (figure 1). The false floor permits considerable freedom in equipment layout, as cables may be routed without restriction. A false floor is sometimes used to provide an underfloor plenum blower system instead of individual cabinet blowers.

TEMPERATURE

Blowers at the bottom of the cabinets or an underfloor plenum blower system cool the equipment by circulating room air through reusable air filters up through the cabinet and out at the top. Room air should not exceed a temperature of 70^{o} F. Heat generated by the equipment should be quickly removed from the vicinity of the cabinets by circulation of the room air. The amount of heat generated by each equipment is listed in table 1; the additional heat load caused by the equipment can be dissipated through increased air conditioning capacity.

Recommended humidity limits are 40% low and 60% high. The low limit protects against static build-up on magnetic tape. The high limit protects punch card operation.

AREA CLEANLINESS

Clean the computer site regularly to avoid dust accumulation. Dust and cigarette ashes may collect on the magnetic tape, and cause errors in operation. Avoid smoking when handling magnetic tapes.

FIRE PRECAUTIONS

Locate fire extinguishing equipment throughout the room, and observe normal fire precautions in the area.

SPACE AND LAYOUT REQUIREMENTS

Positioning of the equipment cabinets will be partially determined by the size and shape of the area available for the computer installation. The operator seated at the console should be able to view the tape handlers and any other equipment with moving parts. It is not necessary for the computer cabinet to be in direct view of the console, although this is desirable for maintenance purposes. Cabinets should be arranged to permit ease of access both for the operator and for maintenance personnel and their equipment. Sample layouts of computer installations, which allow sufficient area between cabinets while remaining within the cable limitations are shown in figure 2. Installation information including dimensions, door swings, floor cutouts, connector data, and weights are given for each item of equipment on following figures. Physical dimensions and weights are summarized in table 1.

As an aid to planning, plastic templets of the equipments (figure 3) may be obtained from Control Data Corporation, 1604 Product Department. The templets are scaled 1/4 inch equals one foot.

TABLE 1. SPECIFICATIONS OF CONTROL DATA 1604 SYSTEM

Equipment	Length (ins.)	Width (ins.)	Height (ins.)	Weight (lbs.)	BTU/ Hr	400 (Breeker	60 Spec.)
1604	89 1/8	27 1/2	67 3/4	2650	24, 000	20A	20A
1605	47 3/4	20 1/2	43	575	4, 000	5A	15A
1607	88 1/2	27 1/2	67 3/4	2580	30, 000	5A	40A
1608	47 3/4	20 1/2	43	575	4, 000	5A	15A
1609	47 3/4	20 1/2	43	575	4, 000	5A	15A
1610	47 3/4	20 1/2	43	575	4, 000	5A	15A
1612	72	31	56	890	6, 400	---	16A
1604 Console	158	27 1/2	43 5/8	800	6, 800	5A	15A
M/G Control	30	22	76	575	---	---	---
M/G Set	39 1/2	19 1/8	18 1/2	610	---	---	---

POWER REQUIREMENTS

The Control Data 1604 and 160 systems operate from 208-volt, 400-cycle, 4-wire service and from 208-volt, 60-cycle, 4-wire service (figure 20). The 400-cycle service is obtained from the motor-generator furnished with the computer system. The motor-generator, utility outlets and equipment blowers are operated from the 208-volt, 60-cycle service. The motor -generator and control cabinet should be located at a ventilated site remote from the computer area (figure 19). The motor-generator and control cabinet may also be located in separate areas.

The motor-generator and control unit (figures 17, 18 and 19) will be installed and wired by Control Data Corporation at the time of computer delivery. The spare motor-generator will also be installed to provide for a minimum of interruption due to generator failure (control and switch-over gear for spare unit included in single control unit).

Two control wires and four power wires from the motor-generator set to the computer area breaker panel are to be installed by the customer prior to shipment of the computer system. These wires may be routed in the same raceway. The motor-alternator output is 7.5-KW, 208-volt, 400-cycle. The four wires carrying the 400-cycle power should be sized to allow no more than a two per cent voltage drop over the length of the run. The two control wires should be sized, in accordance with the code for control circuits, to handle a pushbutton station operating a magnetic contactor.

Two circuit breaker panels provided by the customer (figure 19 shows a sample arrangement) are to be mounted side by side on a wall in the computer room and are to have a common wire raceway across the bottom. One panel handles the 208-volt, 400-cycle, 4-wire power from the motor-generator. This panel needs no main breaker, but one 3-phase breaker for each piece of equipment in the system must be provided.

The other panel handles the 208-volt, 60-cycle, 4-wire power for the various equipments in the computer system. It requires a magnetic contactor for the main disconnect; the size of this contactor will depend upon the amount of power used in the system. This panel should contain one 3-phase breaker for each equipment in the system. Breaker specifications are listed in table 1.

The output side of the breakers will be wired at the time of installation by Control Data Corporation. Space should be left in both panels for the addition of other breakers.

To ease routing and connection of power cables, locate the breaker panels in line with available floor raceways and in an area central to all equipment in the system.

CABLES

The information cables which connect the various elements in the computer system will be delivered at the time of installation. Prior to delivery, the customer can determine the length of the cables to be used by referring to figure 4. Equipment layout can then be revised if any of the cables exceed the maximum of 50 feet.

Cables supplying power to the cabinets (figure 5) originate at the breaker panel where they are permanently installed. Sufficient spare cable should be allowed to accommodate minor changes in location of the equipment. The power cable should not exceed 100 feet in length.

At the time the customer submits the final equipment configuration, Control Data Corporation should be advised of any unusual cabling requirements or obstructions beneath the floor that will interfere with the cables. This should be done no later than two months prior to shipment.

TABLE 2. CABLE CONNECTIONS, 1604 CONSOLE

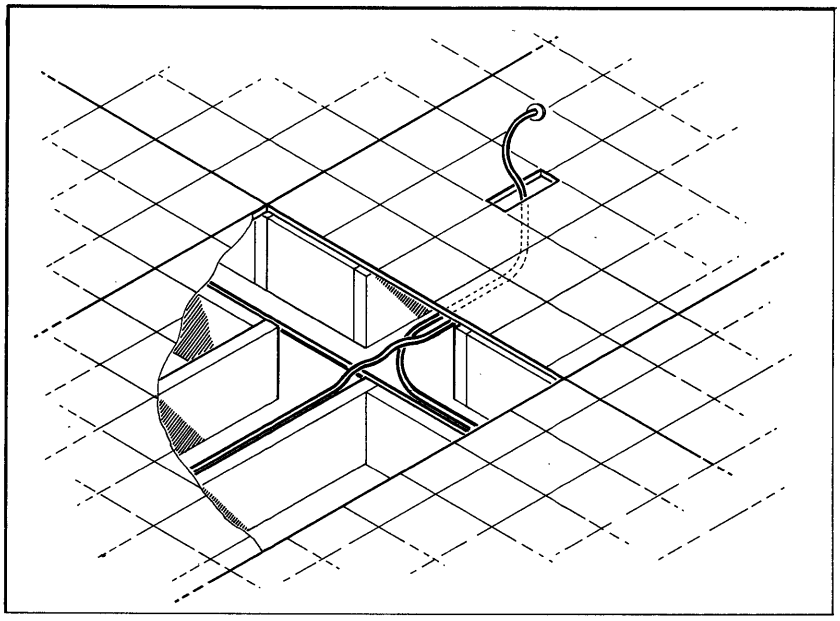
1604 Console	1604 Computer		1604 Console	1604 Computer	
J20301	7H2	Chassis 7	J20318	5L1	
J20302	7I1		J20319	5L2	
J20303	1M2		J20320	5M1	
J20304	1N1		J20321	5M2	
J20305	1N2	Chassis 1	J20322	5N1	Chassis 5
J20306	1O1		J20323	5N2	
J20307	1O2		J20324	5O1	
J20308	1P1		J20325	5O2	
J20309	2M2		J20326	5P1	
J20310	2N1		J20327	6N2	
J20311	2N2	Chassis 2	J20328	6O1	Chassis 6
J20312	2O1		J20329	6O2	
J20313	2O2		J20330	7I2	Chassis 7
J20314	2P1				
J20315	---	400 Power			
J20316	---	60 Power			
J20317	---	Power Control			

TABLE 3. CABLE CONNECTIONS, INPUT-OUTPUT EQUIPMENT

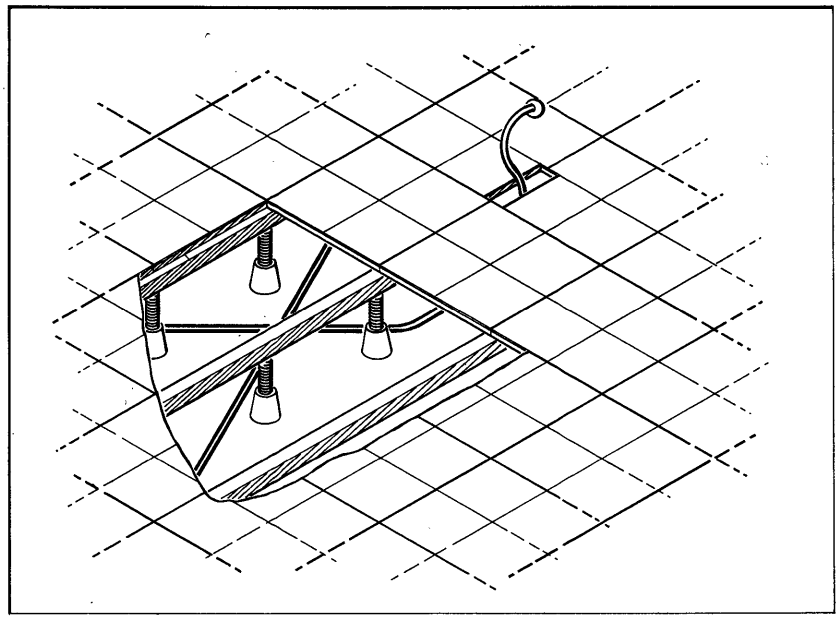
Type Designation	Cable Group 1	Cable Group 2	Cable Group 3	Cable Group 4
Input Channel Cable A	7J2	7L1	7M2	7O1
Input Channel Cable B	7K1	7L2	7M1	7O2
Input Channel Cable C	7K2	7M1	7N2	7P1
Output Channel Cable D	8J2	8L1	8M2	8O1
Output Channel Cable E	8K1	8L2	8N1	8O2
Output Channel Cable F	8K2	8M1	8N2	8P1

Except for variation in length all information cables used in the systems, including 1605, 1607 and other equipment, are identical. Detailed cable makeup and inter-connection data are found in the maintenance volume.

All cables used in the 1604 system are supplied by Control Data Corporation at the time of delivery.



A. False Floor - Raceway Type



B. False Floor - Pedestal Type

Figure 1. False Floors

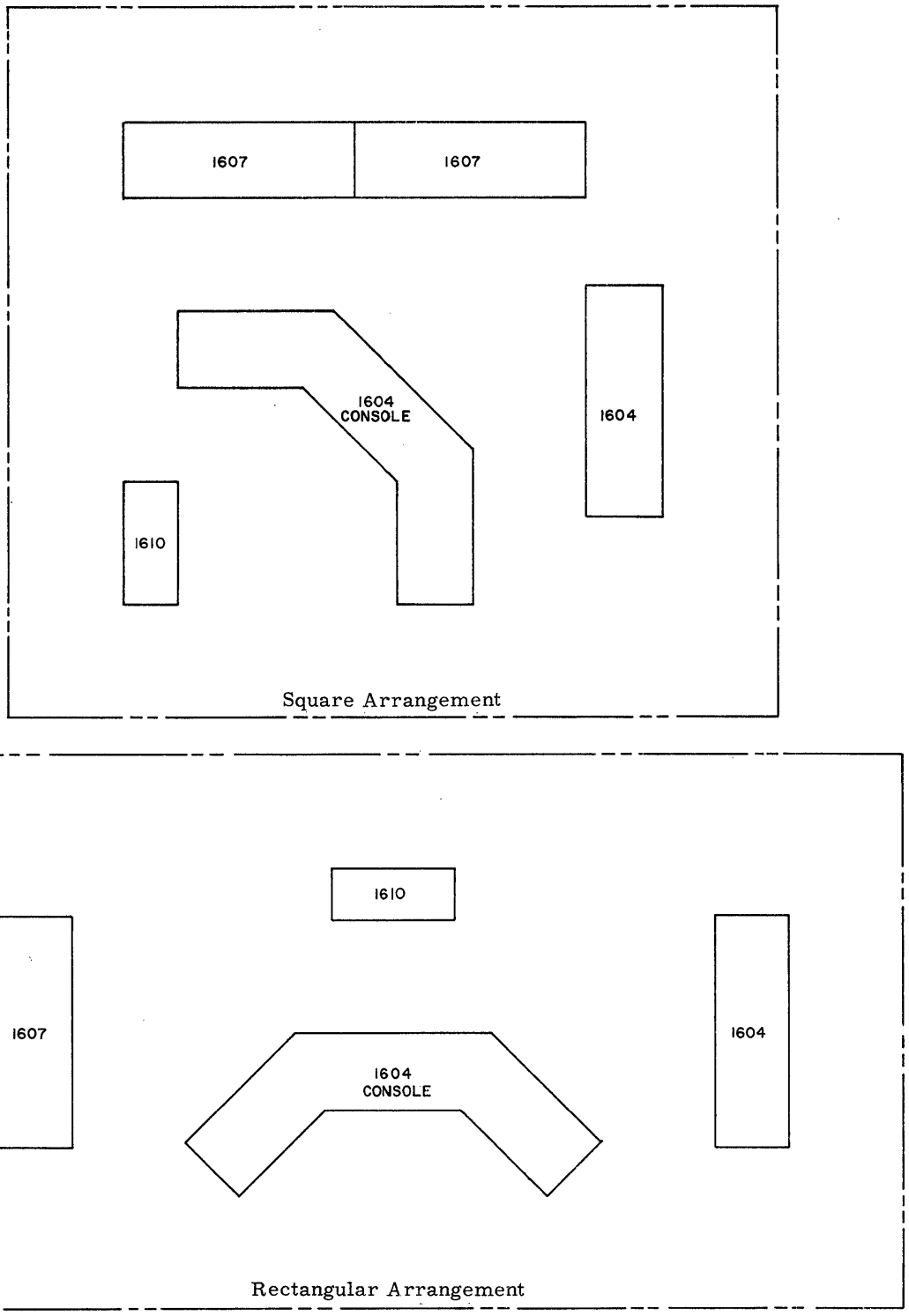


Figure 2. Sample Layouts of Computer Installation

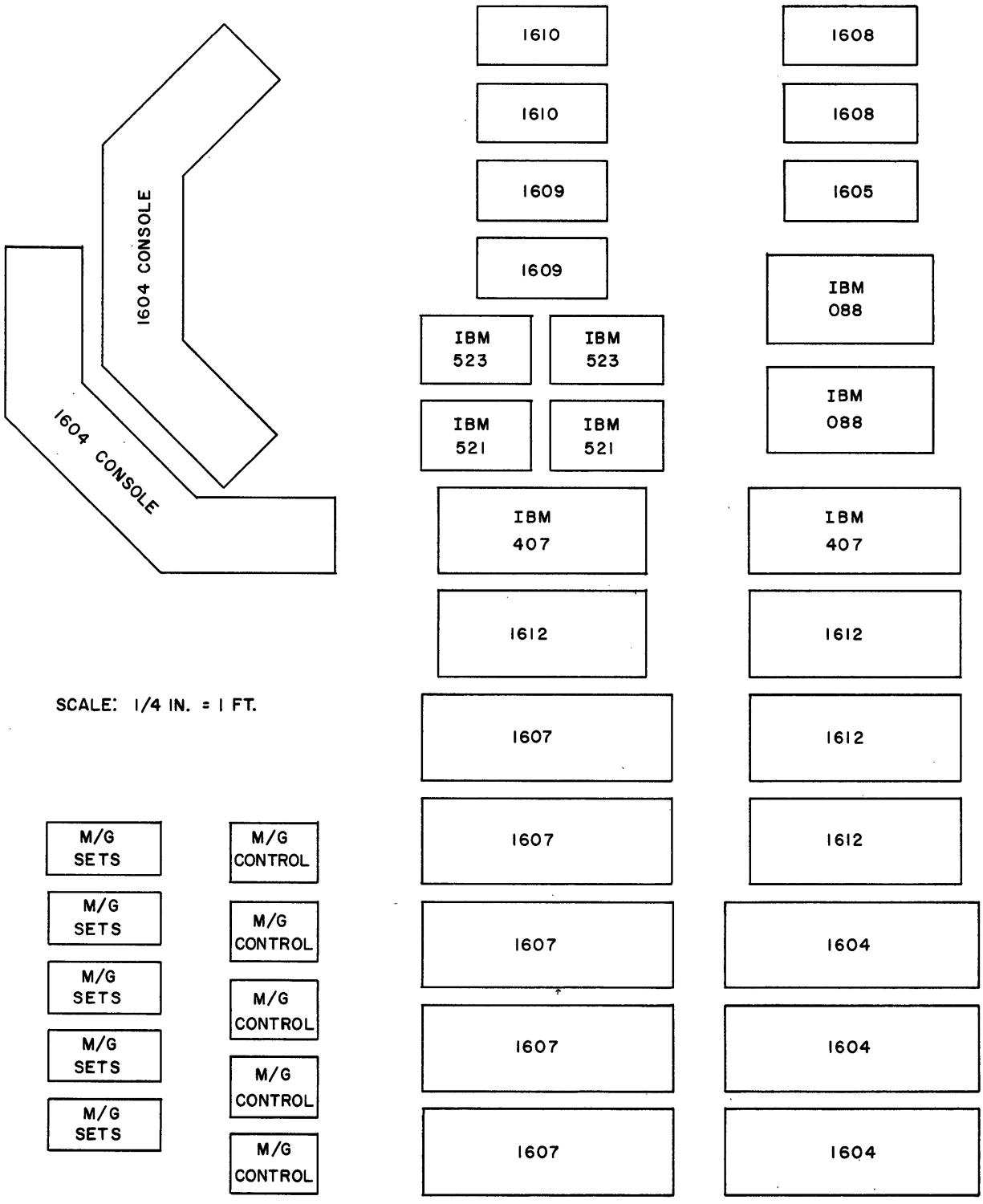
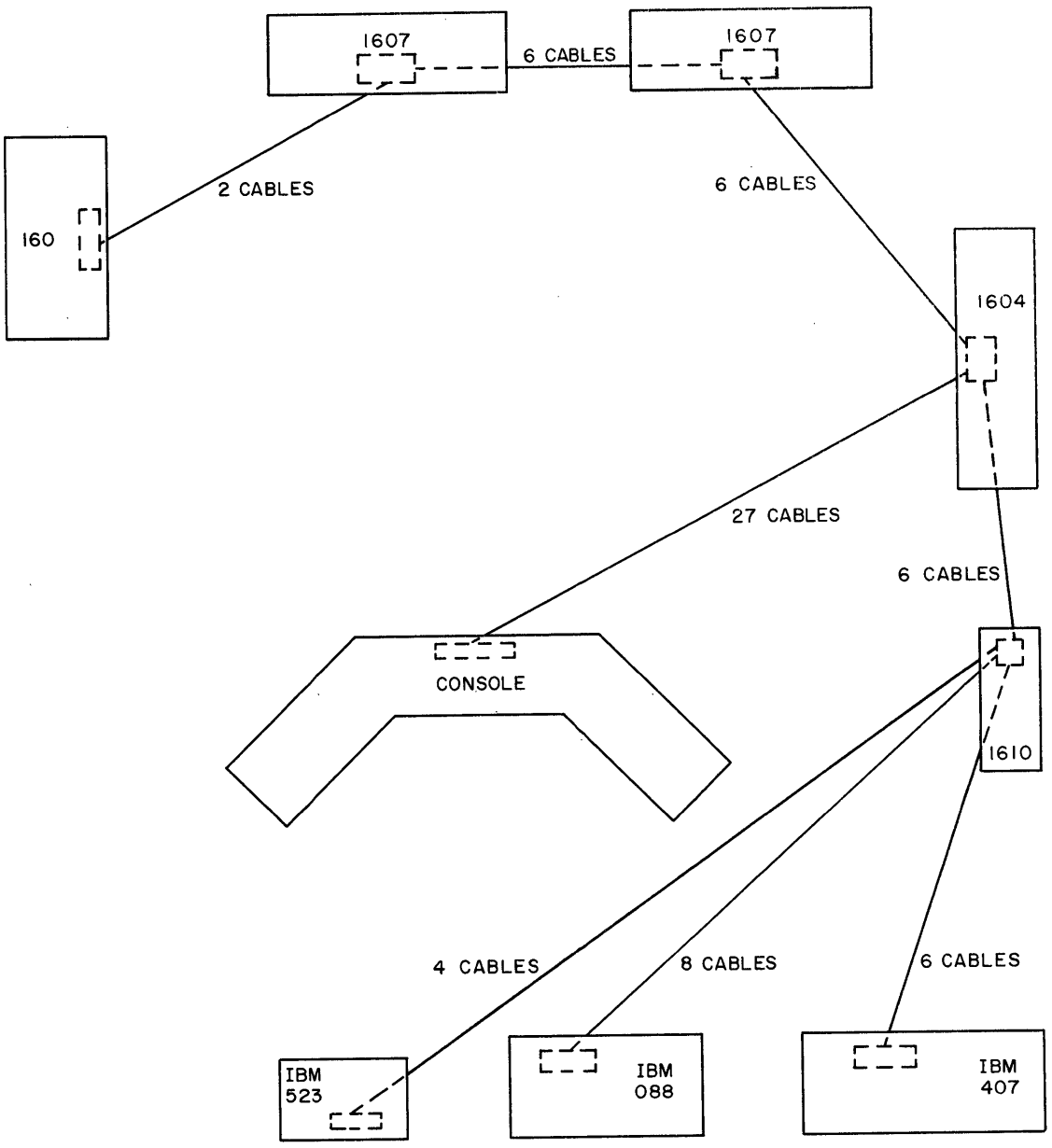


Figure 3. 1604 Computer System Layout Templets



Cable length is determined by the distance between the cable cutouts of two equipments plus 10 feet which allows sufficient cable for internal conditions.

Figure 4. Information Cable Lengths

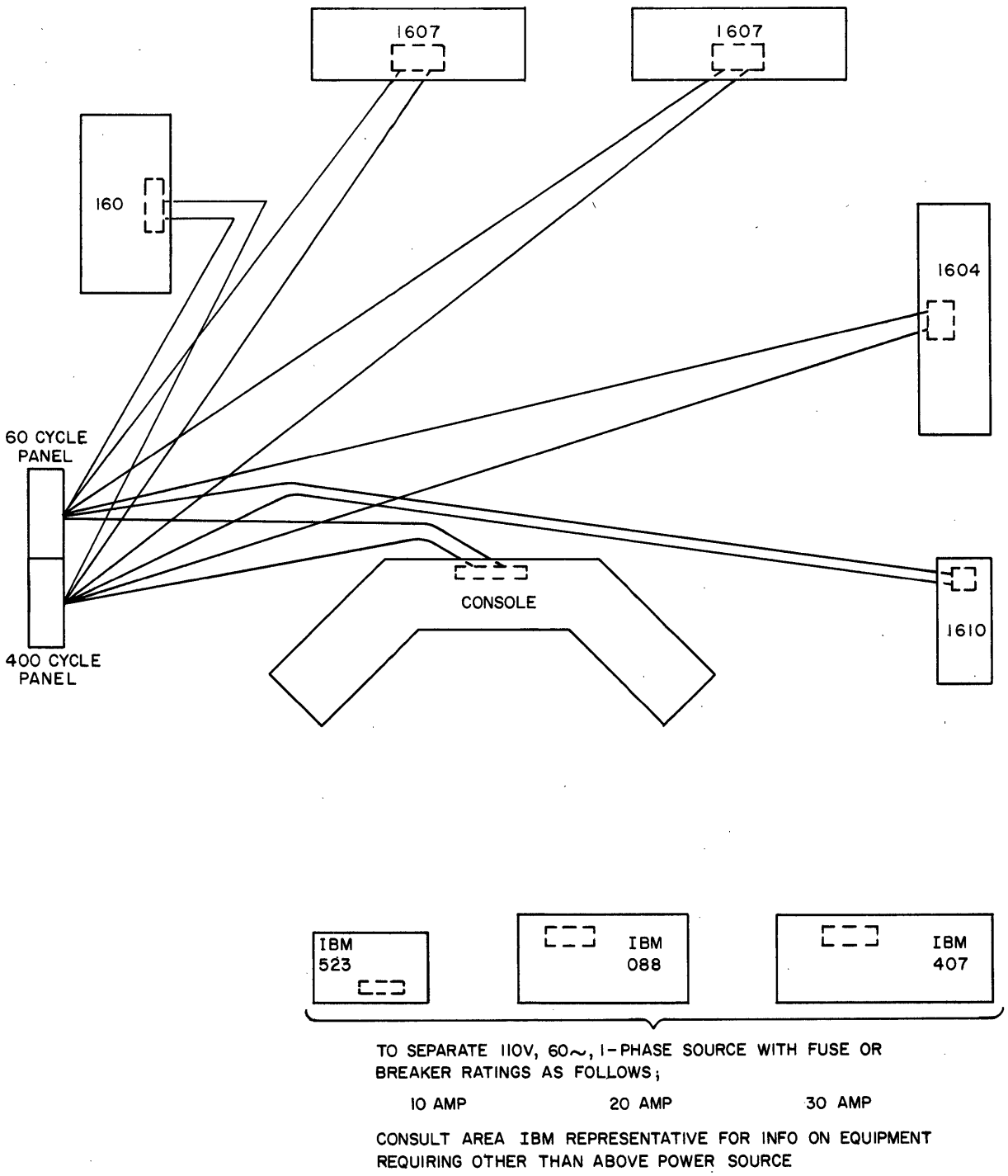
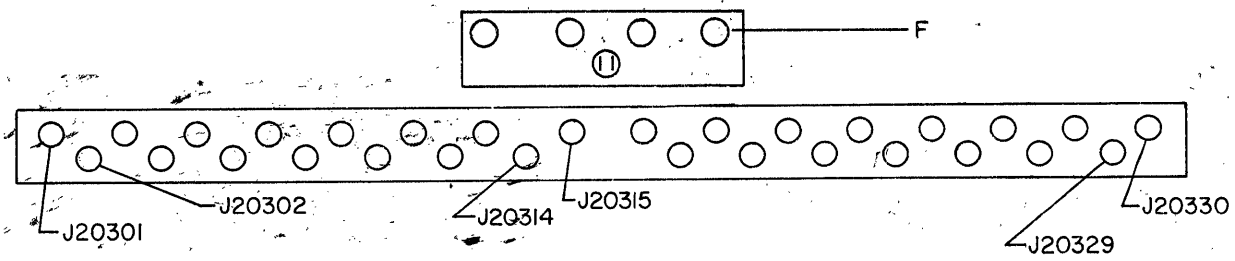


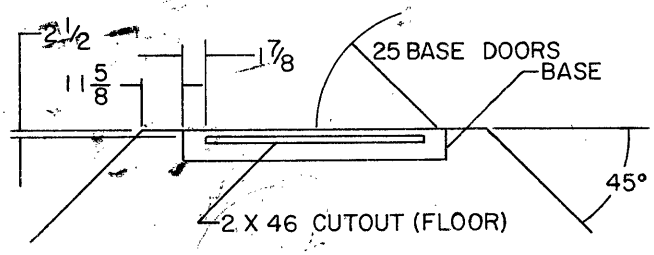
Figure 5. Power Cable Lengths

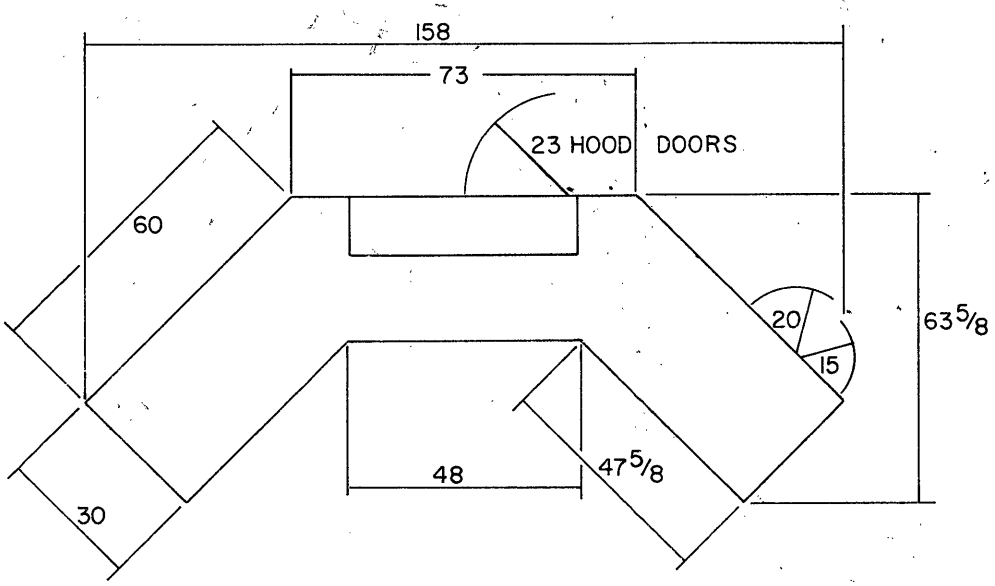


VIEW FROM BOTTOM OF CONNECTOR PANEL

J20315 = 400~ POWER
J20316 = 60~ POWER
J20317 = CONTROL

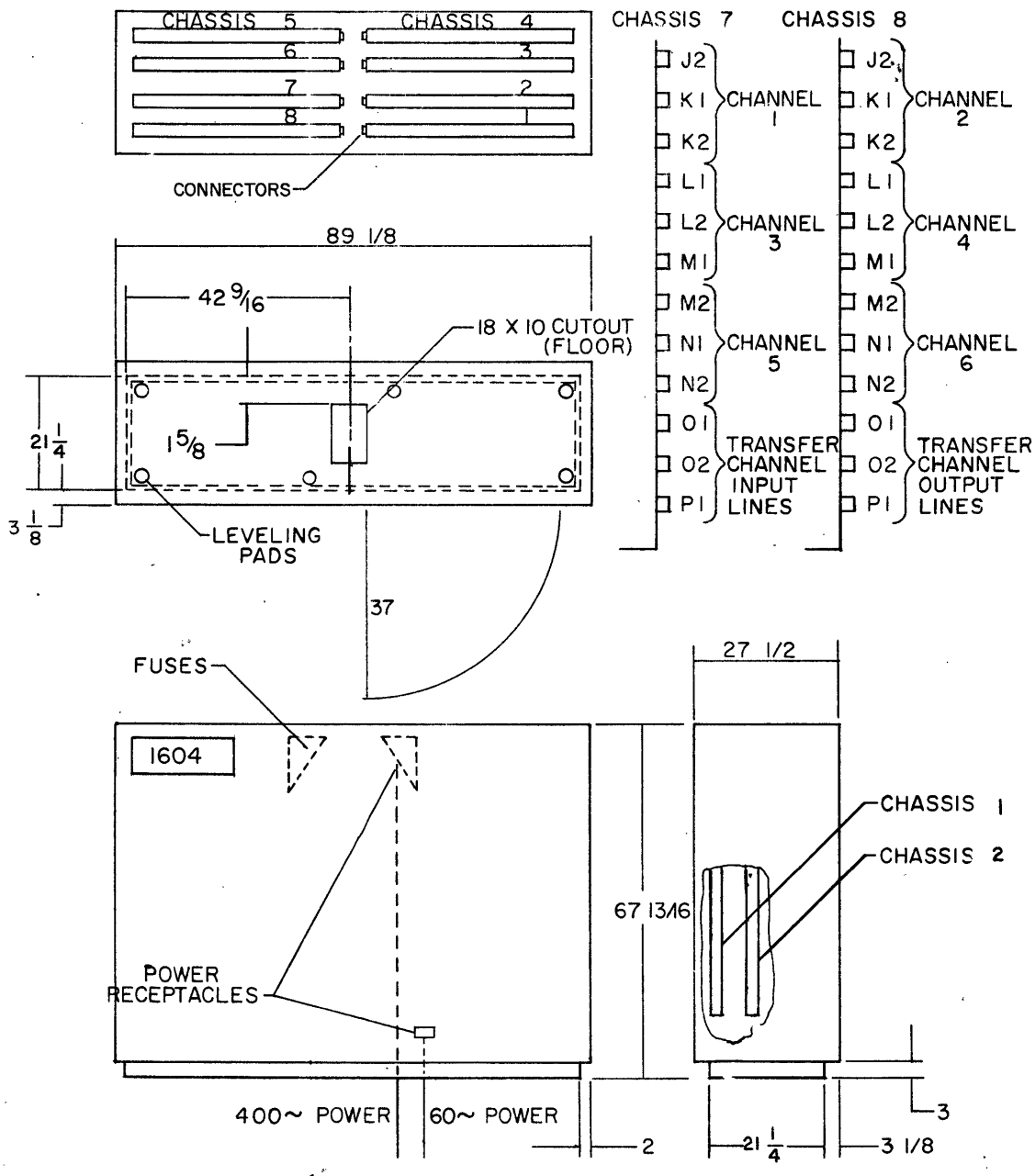
Note: Cable connections for the 1604 are listed in table 2.





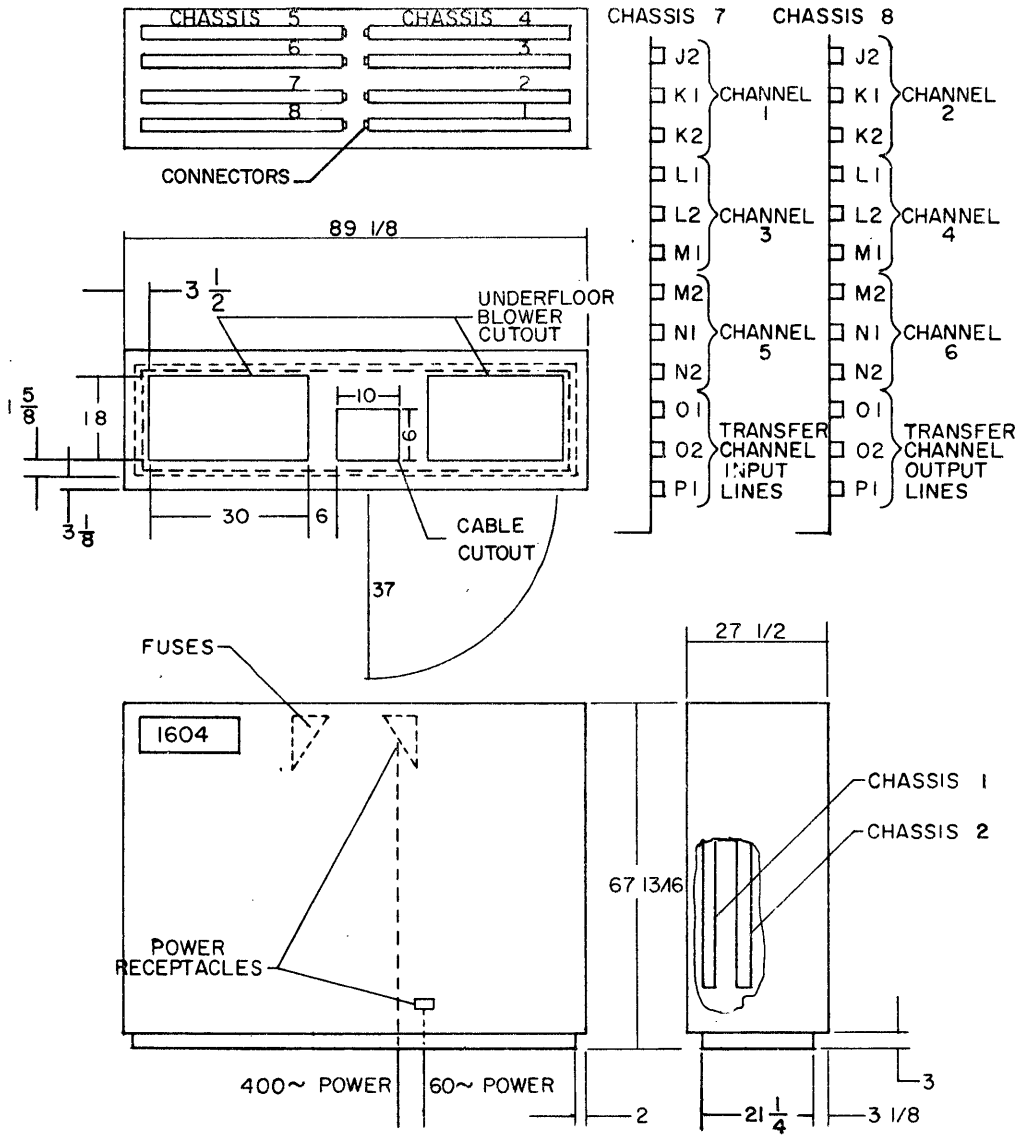
Length	158"	Width	27 1/2"
Height	43 5/8"	Weight	800 lbs.

Figure 6. 1604 Computer Console



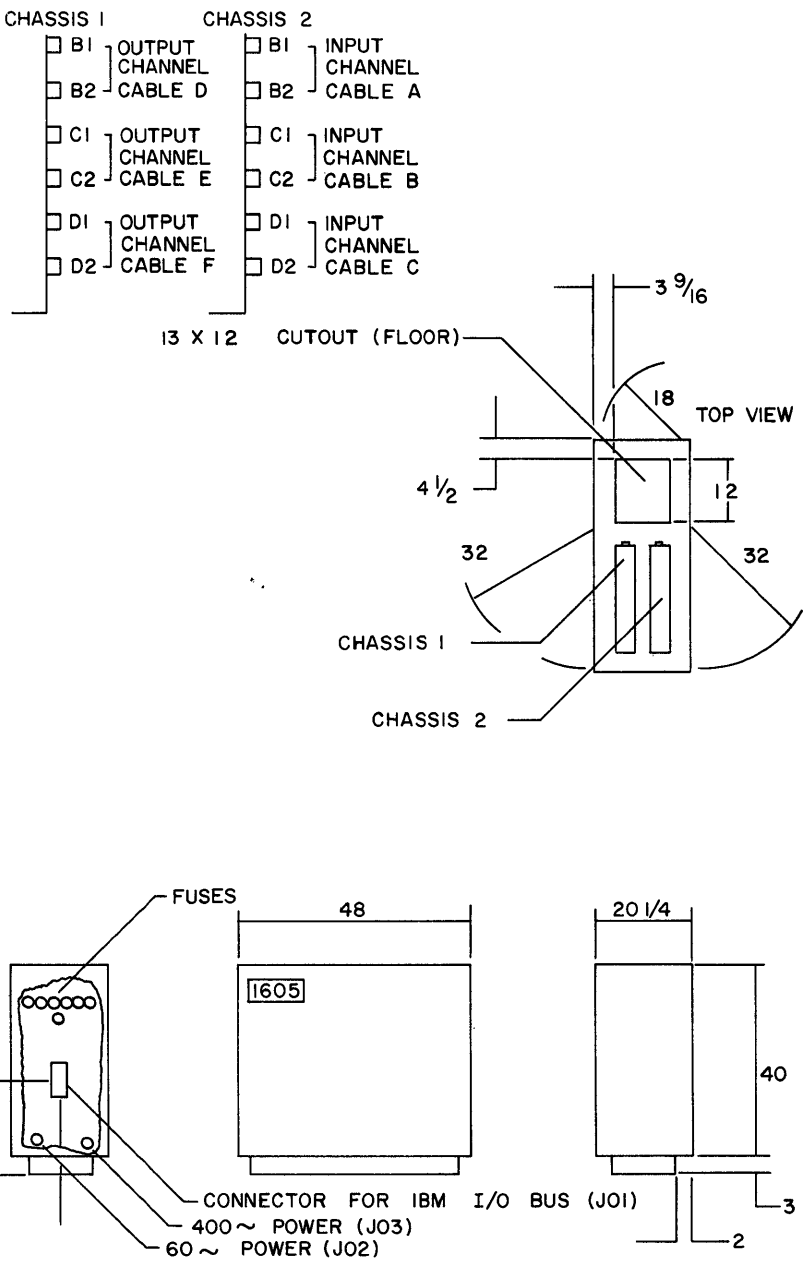
Length	89 1/2"	Width	27 1/2"
Height	67 13/16"	Weight	2650 lbs.

Figure 7. 1604 Computer Cabinet



Length	89 1/2"	Width	27 1/2"
Height	67 13/16"	Weight	2650 lbs.

Figure 8. 1604 Computer Cabinet (Underfloor Plenum Blower System)



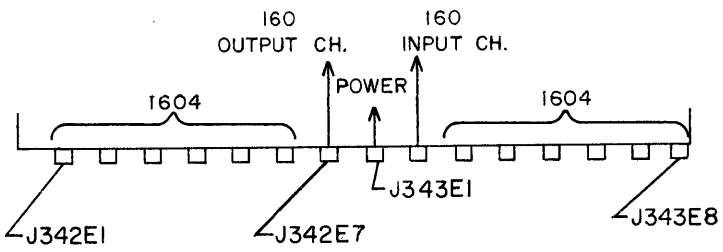
Length 48

Width 20 1/4"

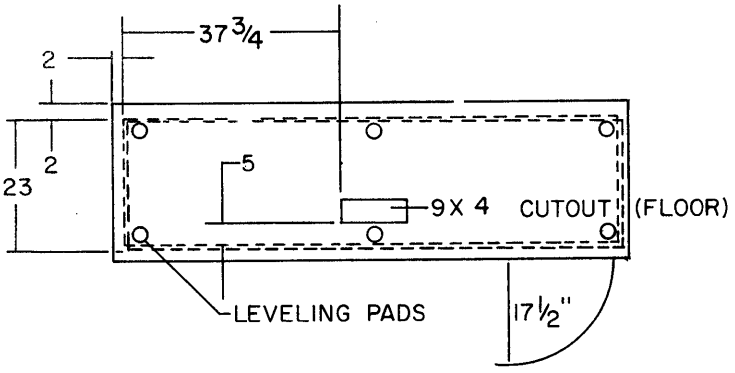
Height 43"

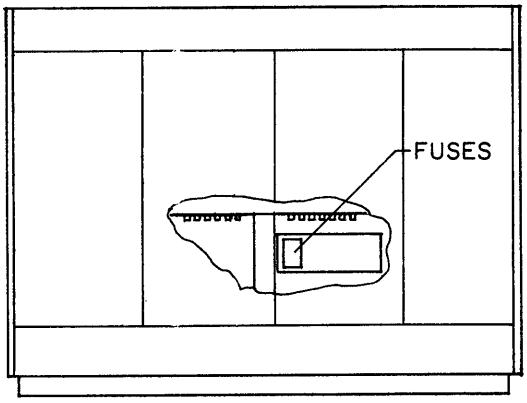
Weight 575 lbs.

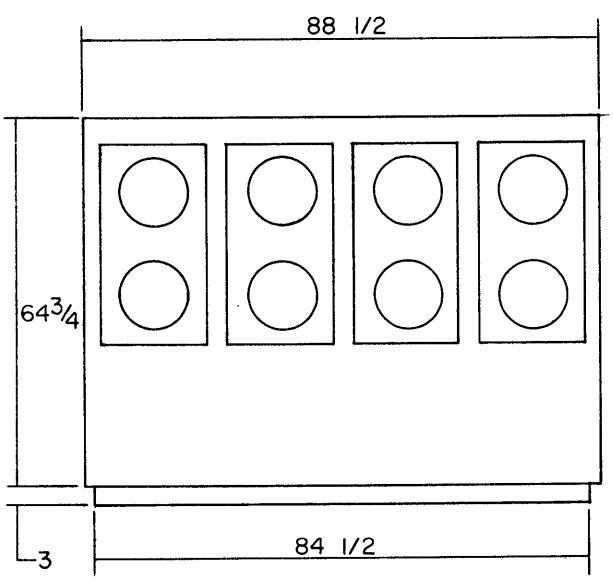
Figure 9. 1605 Adaptor Cabinet

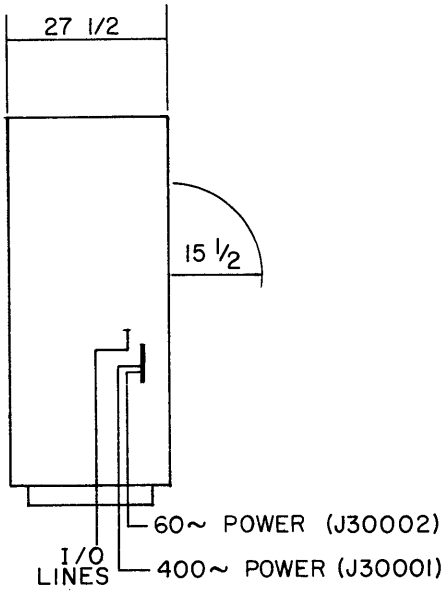


Note: Cable connections for the 1607 are listed in table 3.



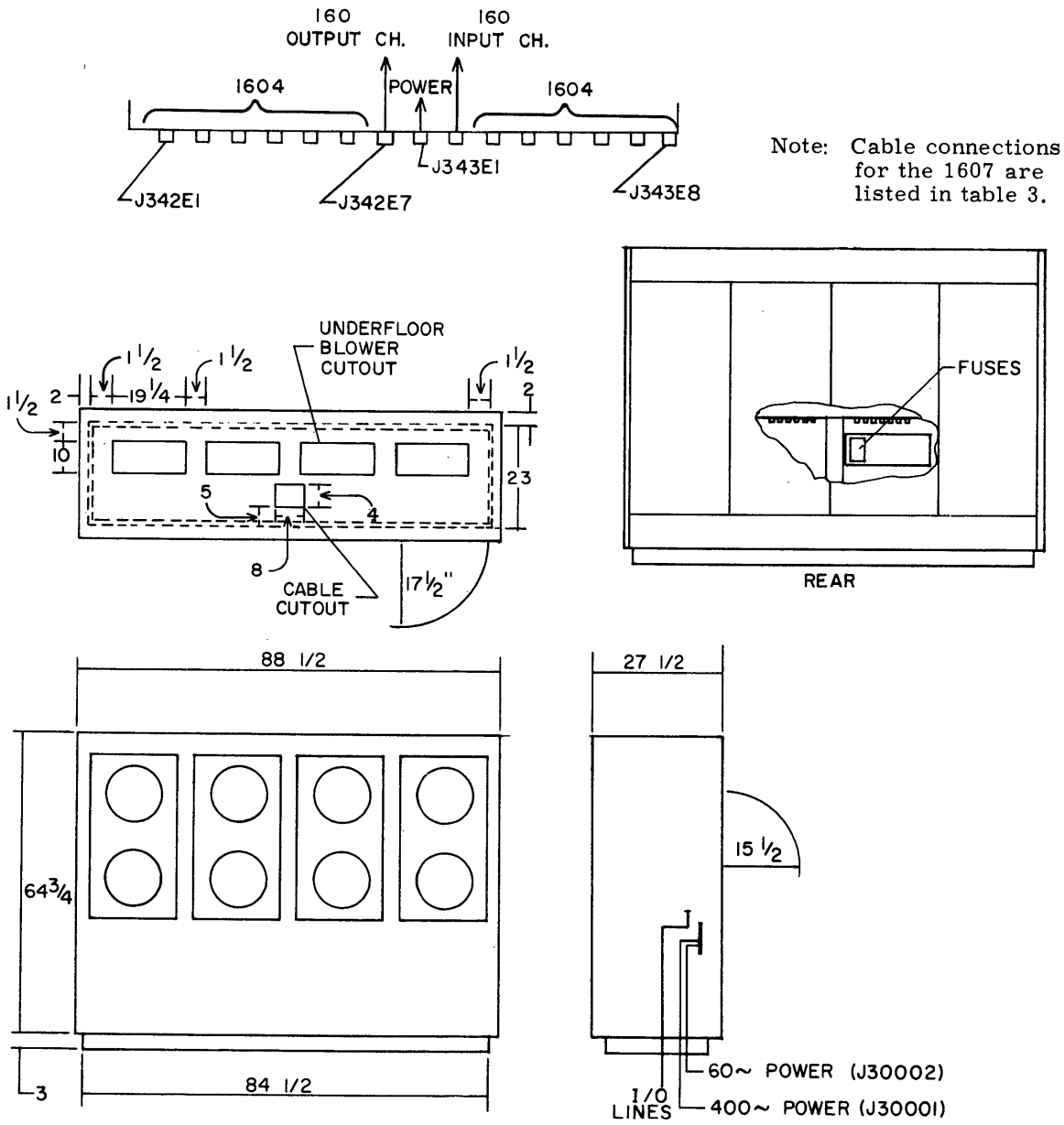






Length	88 1/2"	Width	27 1/2"
Height	67 3/4"	Weight	2580 lbs.

Figure 10. 1607 Tape System



Length	88 1/2"	Width	27 1/2"
Height	67 3/4"	Weight	2580 lbs.

Figure 11. 1607 Tape System (Underfloor Plenum Blower System)

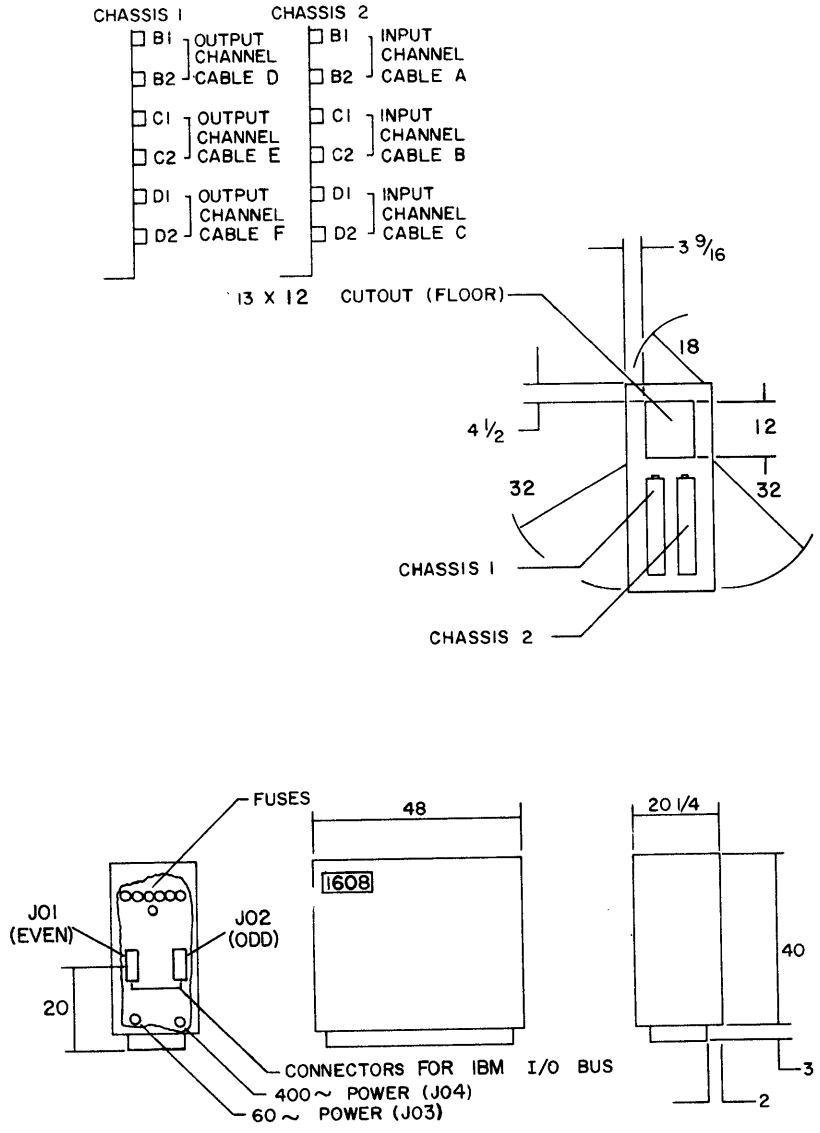


Figure 12. 1608 Tape Control Unit Cabinet

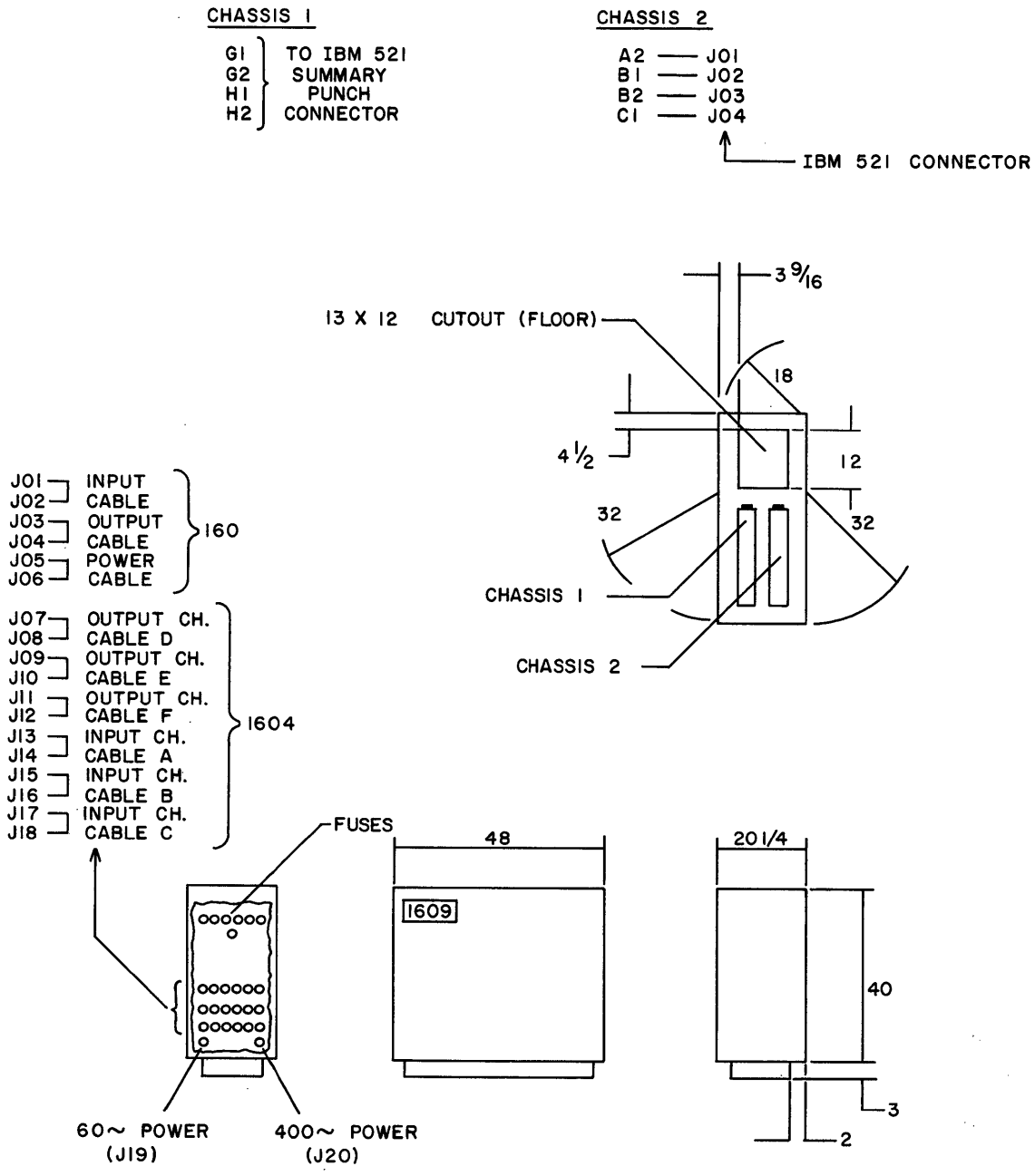


Figure 13. 1609 Adaptor Cabinet

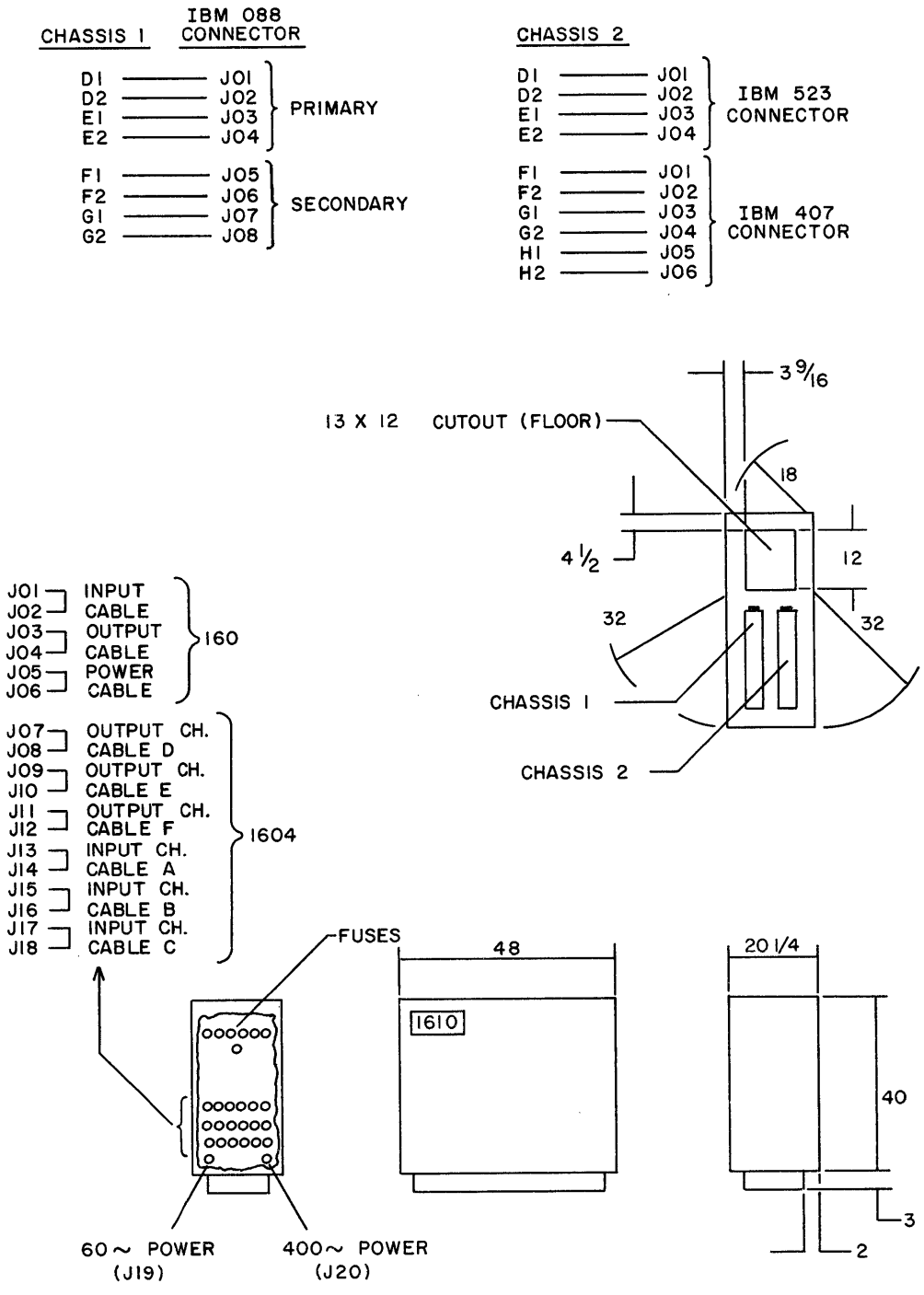


Figure 14. 1610 Adaptor Cabinet

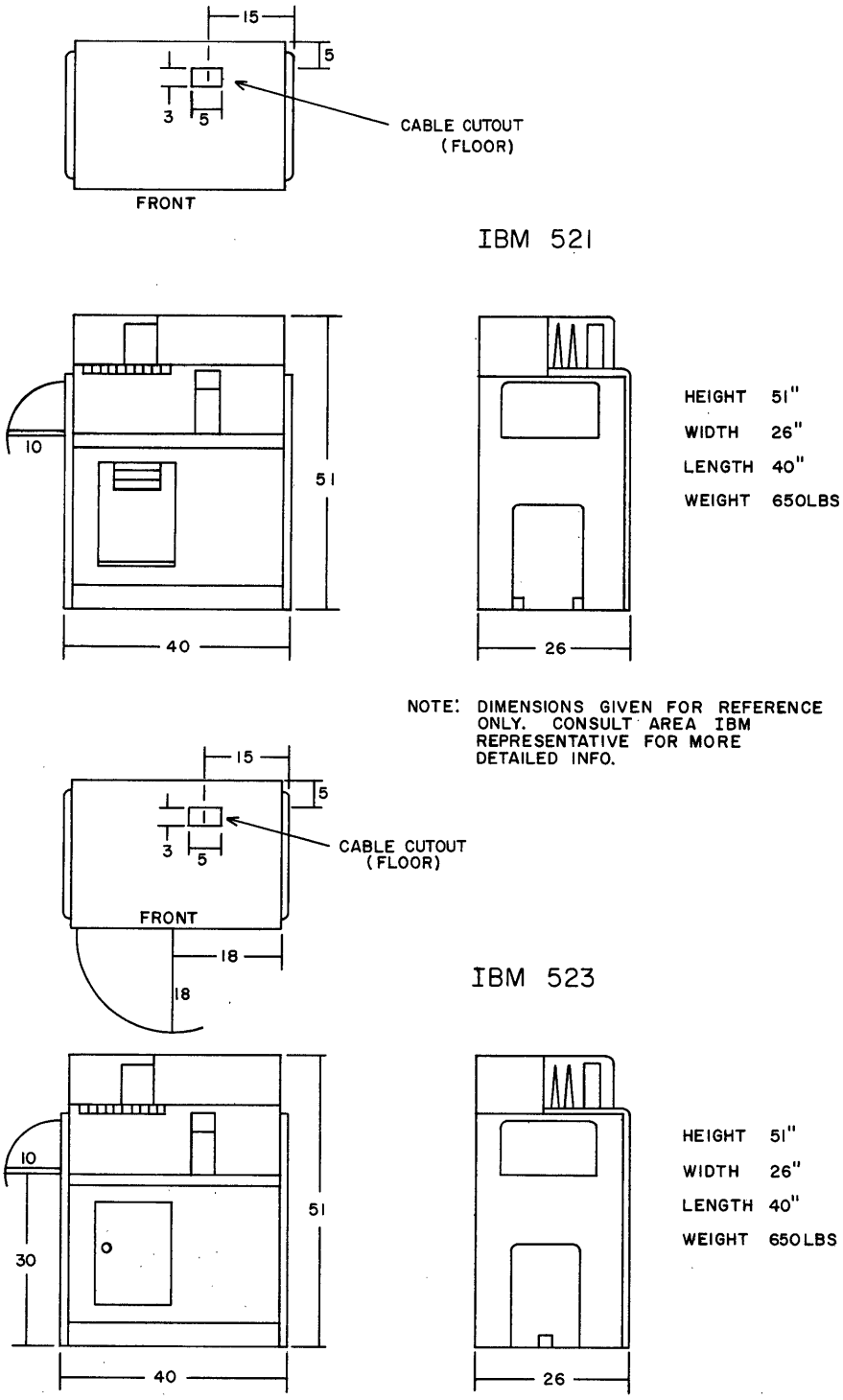


Figure 15. IBM 521 and 523 Cabinets

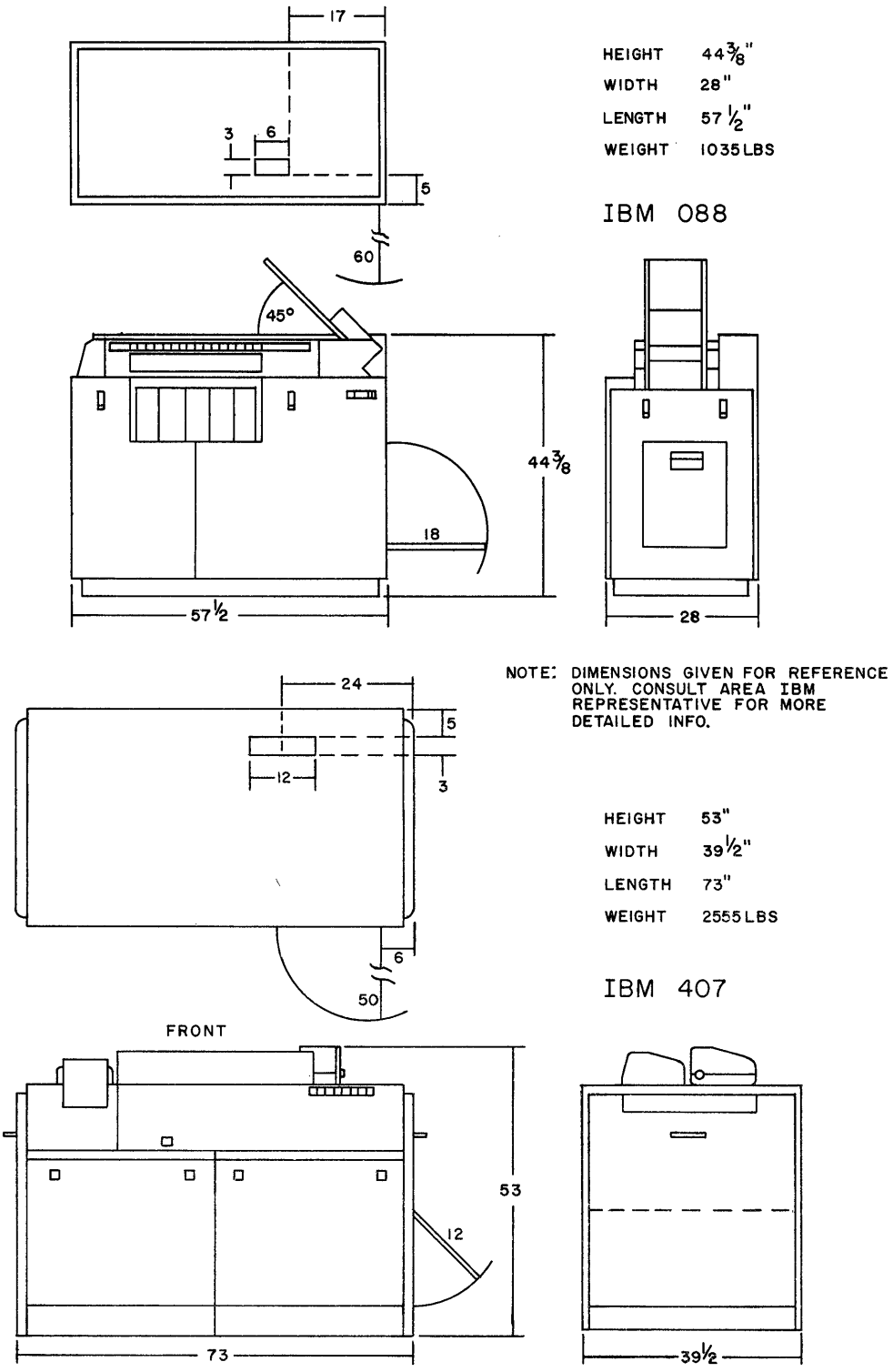


Figure 16. IBM 088 and 407 Cabinets

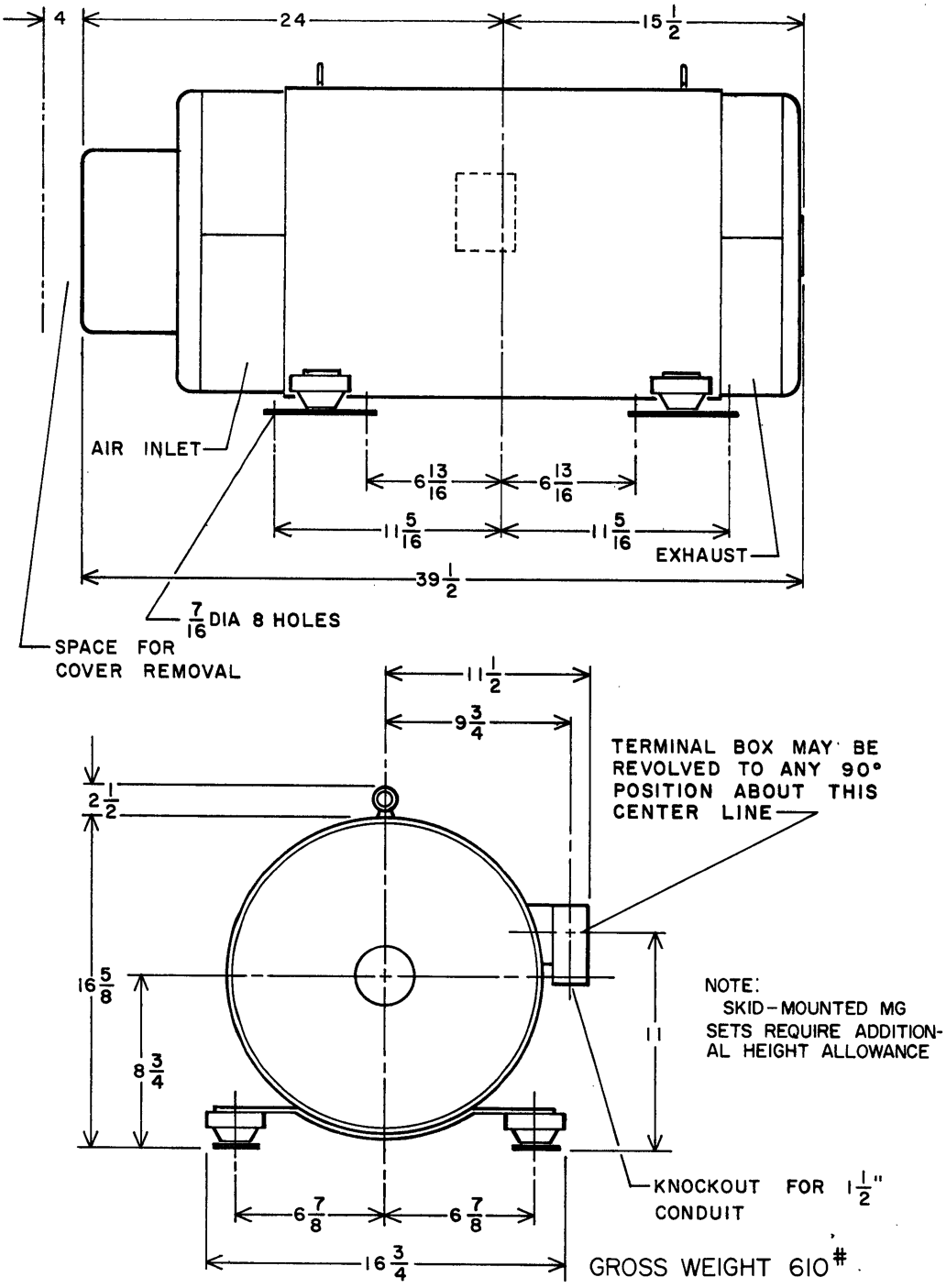
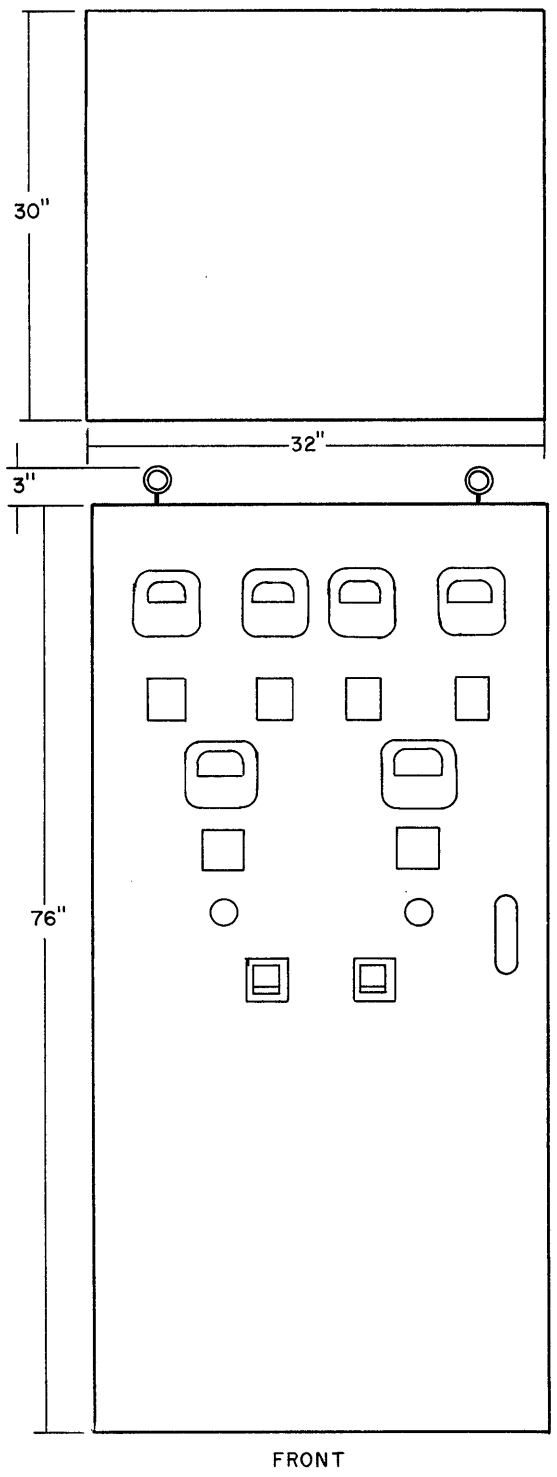
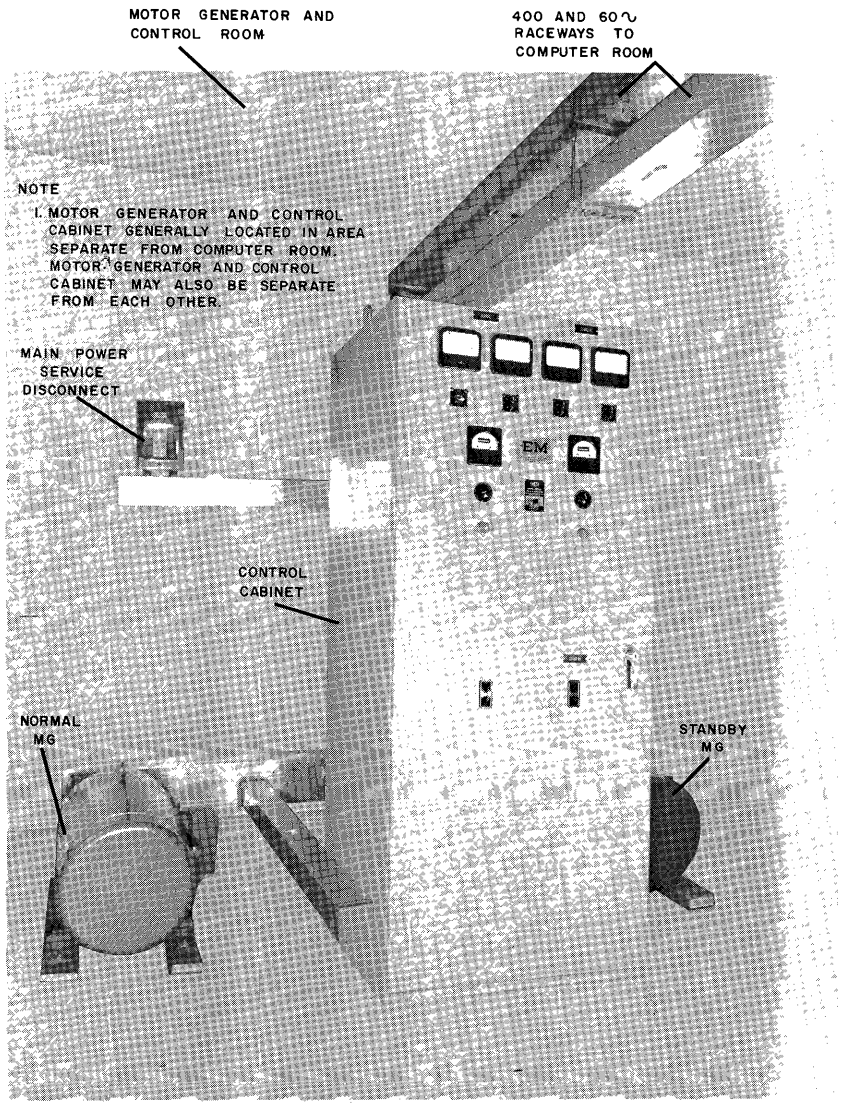


Figure 17. Motor-Generator Set



GROSS WEIGHT 800 LBS

Figure 18. Motor-Generator Control Cabinet



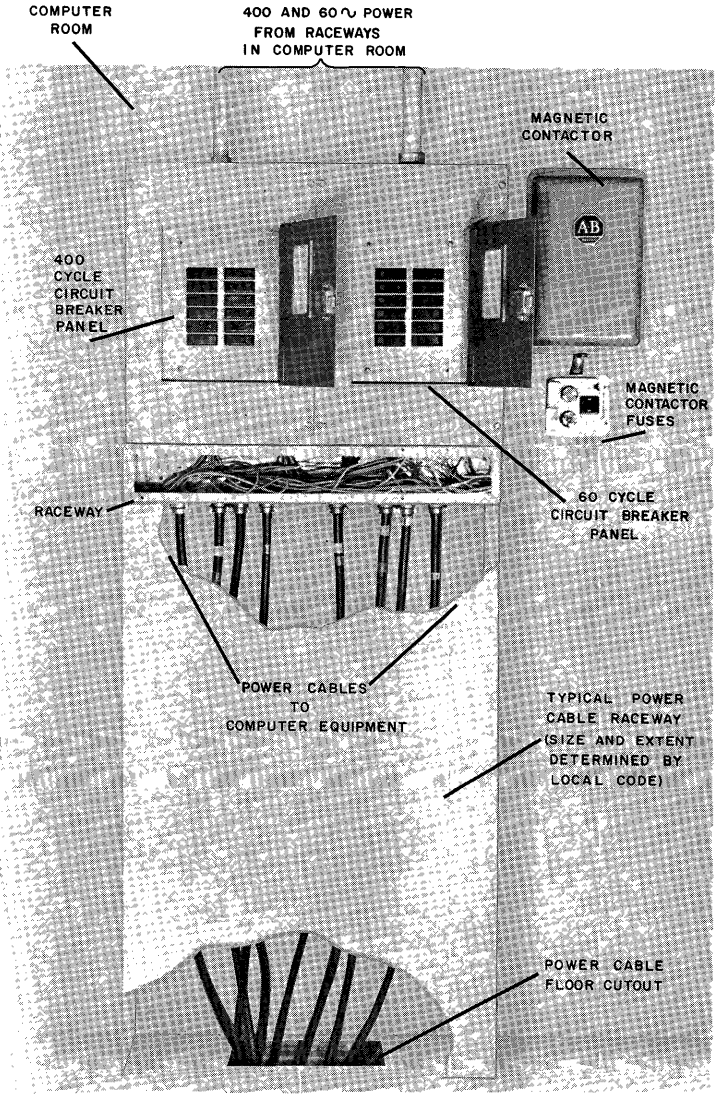
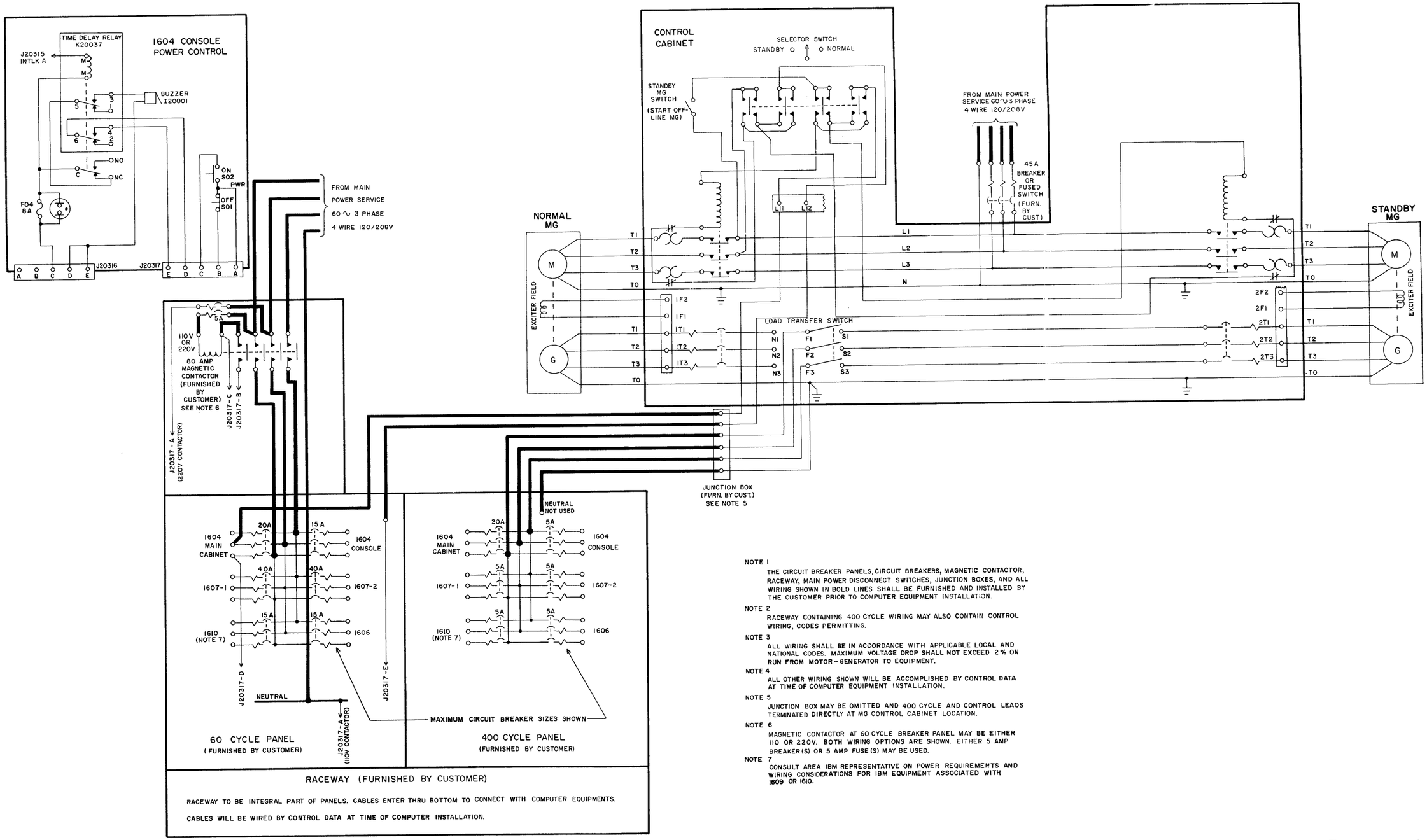


Figure 19. Typical Power Installation



NOTE 1
THE CIRCUIT BREAKER PANELS, CIRCUIT BREAKERS, MAGNETIC CONTACTOR, RACEWAY, MAIN POWER DISCONNECT SWITCHES, JUNCTION BOXES, AND ALL WIRING SHOWN IN BOLD LINES SHALL BE FURNISHED AND INSTALLED BY THE CUSTOMER PRIOR TO COMPUTER EQUIPMENT INSTALLATION.

NOTE 2
RACEWAY CONTAINING 400 CYCLE WIRING MAY ALSO CONTAIN CONTROL WIRING, CODES PERMITTING.

NOTE 3
ALL WIRING SHALL BE IN ACCORDANCE WITH APPLICABLE LOCAL AND NATIONAL CODES. MAXIMUM VOLTAGE DROP SHALL NOT EXCEED 2% ON RUN FROM MOTOR-GENERATOR TO EQUIPMENT.

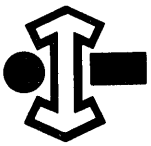
NOTE 4
ALL OTHER WIRING SHOWN WILL BE ACCOMPLISHED BY CONTROL DATA AT TIME OF COMPUTER EQUIPMENT INSTALLATION.

NOTE 5
JUNCTION BOX MAY BE OMITTED AND 400 CYCLE AND CONTROL LEADS TERMINATED DIRECTLY AT MG CONTROL CABINET LOCATION.

NOTE 6
MAGNETIC CONTACTOR AT 60 CYCLE BREAKER PANEL MAY BE EITHER 110 OR 220V. BOTH WIRING OPTIONS ARE SHOWN. EITHER 5 AMP BREAKER(S) OR 5 AMP FUSE(S) MAY BE USED.

NOTE 7
CONSULT AREA IBM REPRESENTATIVE ON POWER REQUIREMENTS AND WIRING CONSIDERATIONS FOR IBM EQUIPMENT ASSOCIATED WITH 1609 OR 1610.

Figure 20. Installation Power Requirements



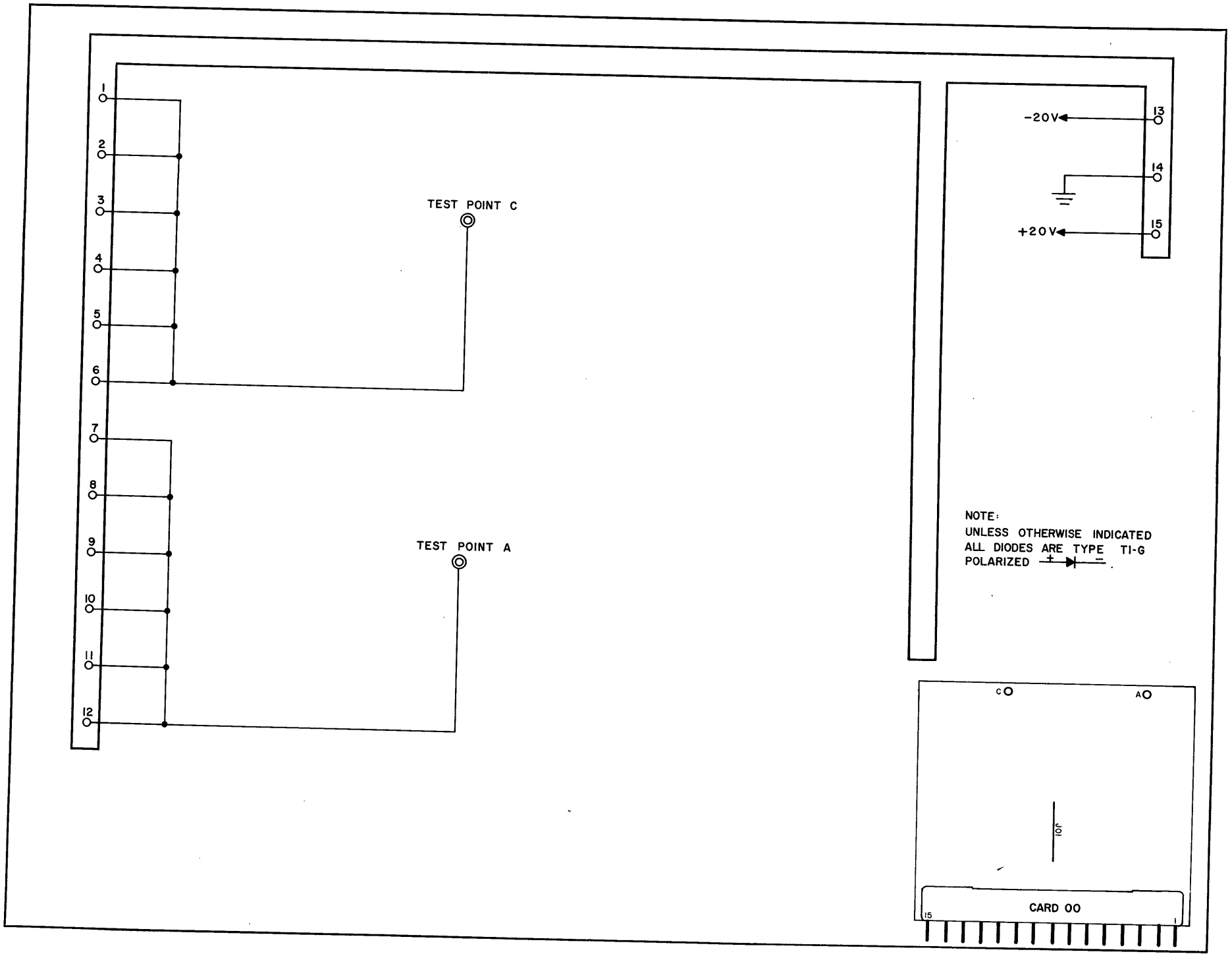
APPENDIX B

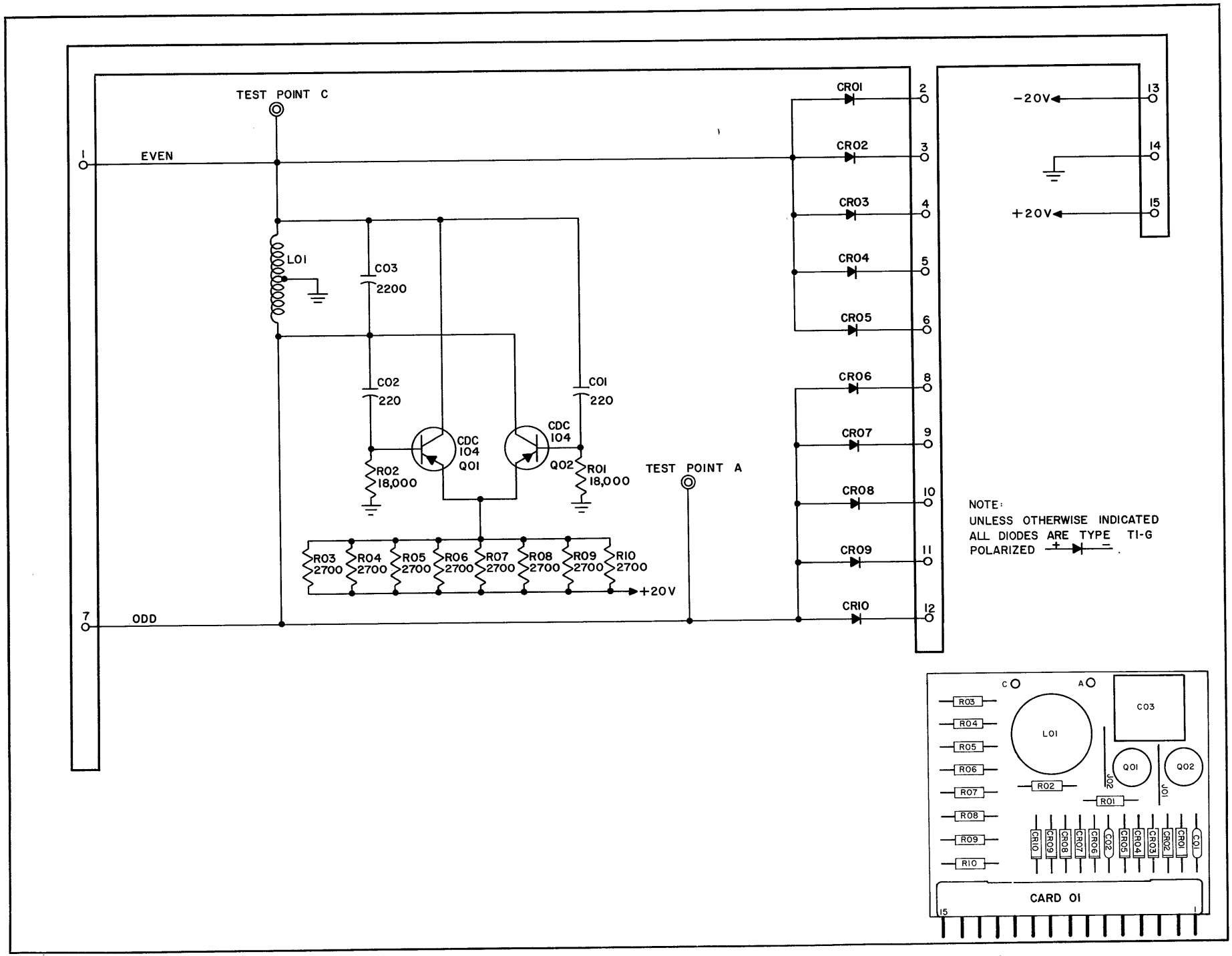
Card Schematics

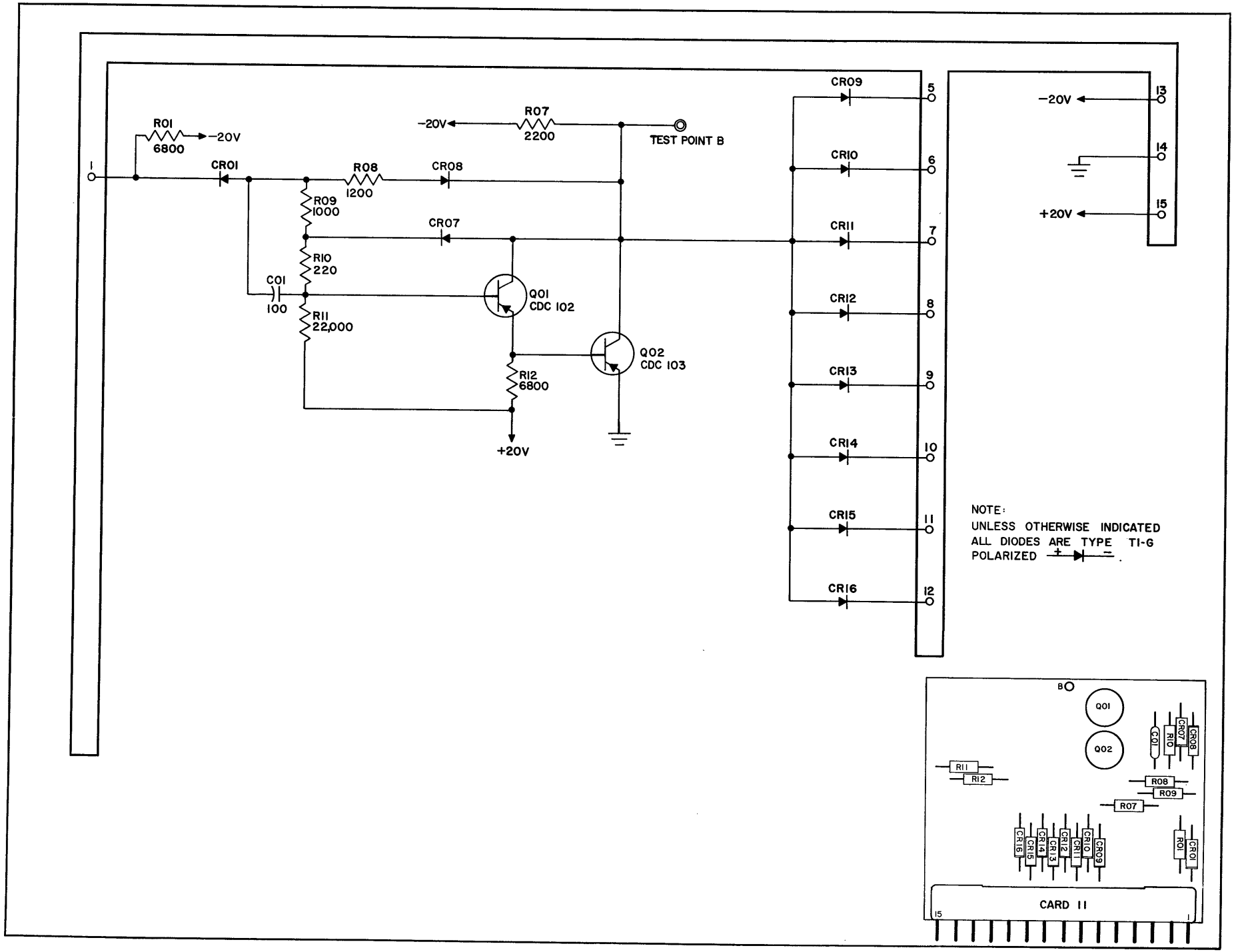
This appendix contains schematic diagrams for all printed circuit cards used in the 1604 computer. Schematics for special cards used only in a given external equipment appear in the instruction book for that equipment.

The lower right-hand corner of the schematic shows the physical layout of components on the printed circuit board.

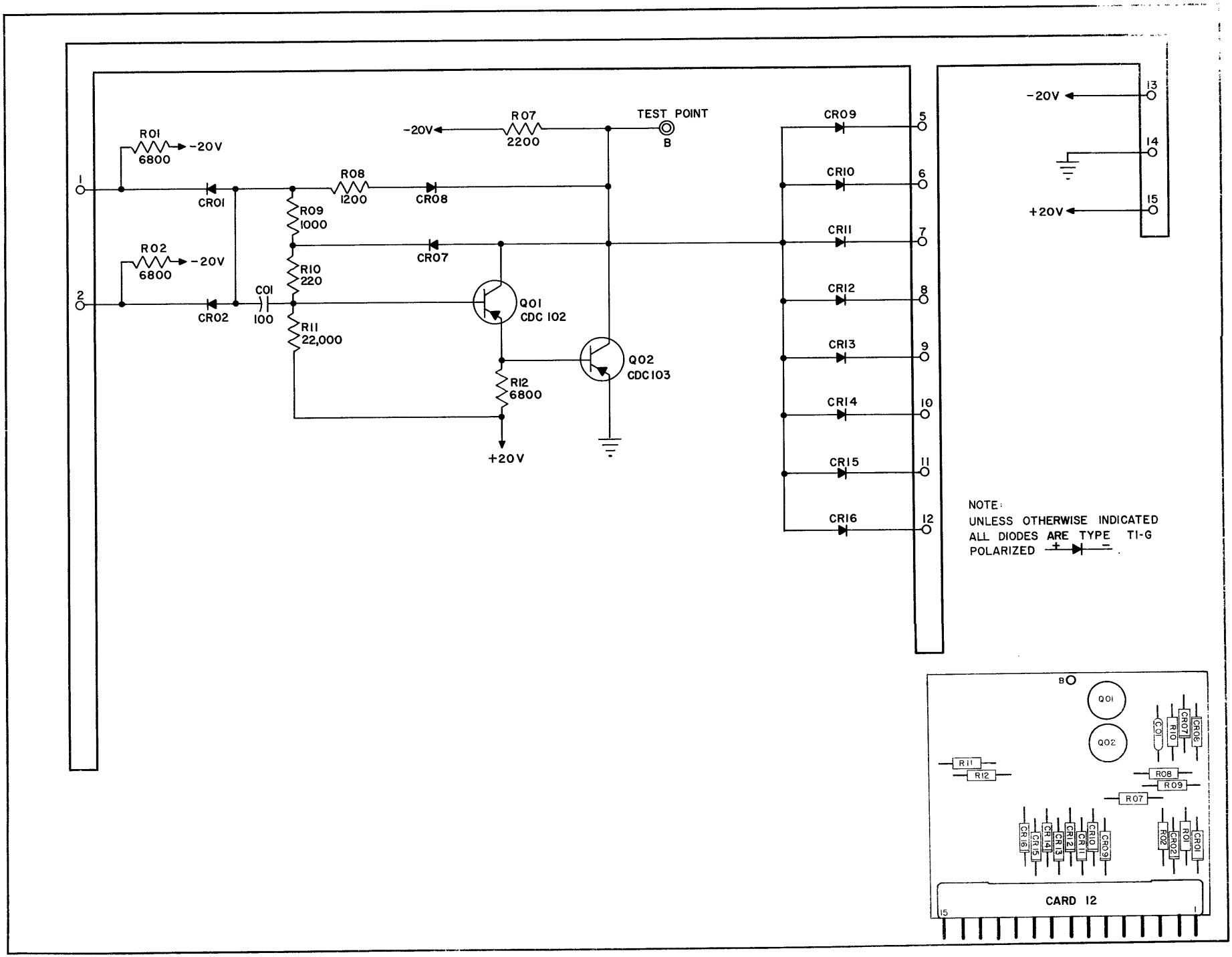
The schematics are arranged in ascending order of card type numbers.


1
2
3
4
5
6
7
8
9
10
11
12
TEST POINT C
TEST POINT A
-20V
+20V
13
14
15
NOTE:
UNLESS OTHERWISE INDICATED
ALL DIODES ARE TYPE TI-G
POLARIZED
C
A
10K
CARD 00
15
1

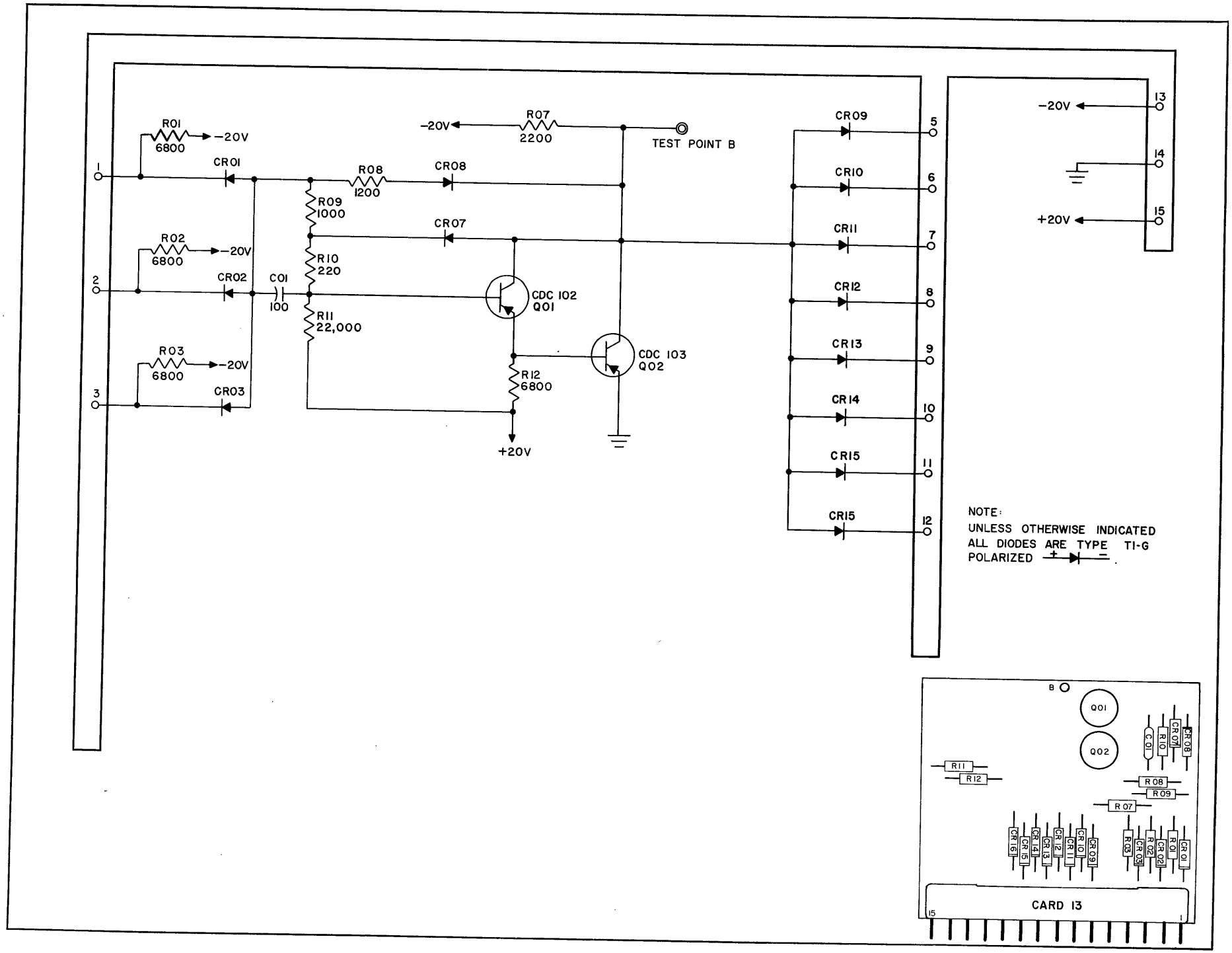

TEST POINT C
EVEN
ODD
L01
C03
2200
C02
220
C01
220
CDC
104
Q01
CDC
104
Q02
R02
18,000
R01
18,000
TEST POINT A
R03
2700
R04
2700
R05
2700
R06
2700
R07
2700
R08
2700
R09
2700
R10
2700
+20V
CR01
CR02
CR03
CR04
CR05
CR06
CR07
CR08
CR09
CR10
-20V
+20V
NOTE:
UNLESS OTHERWISE INDICATED
ALL DIODES ARE TYPE TI-G
POLARIZED
CARD 01



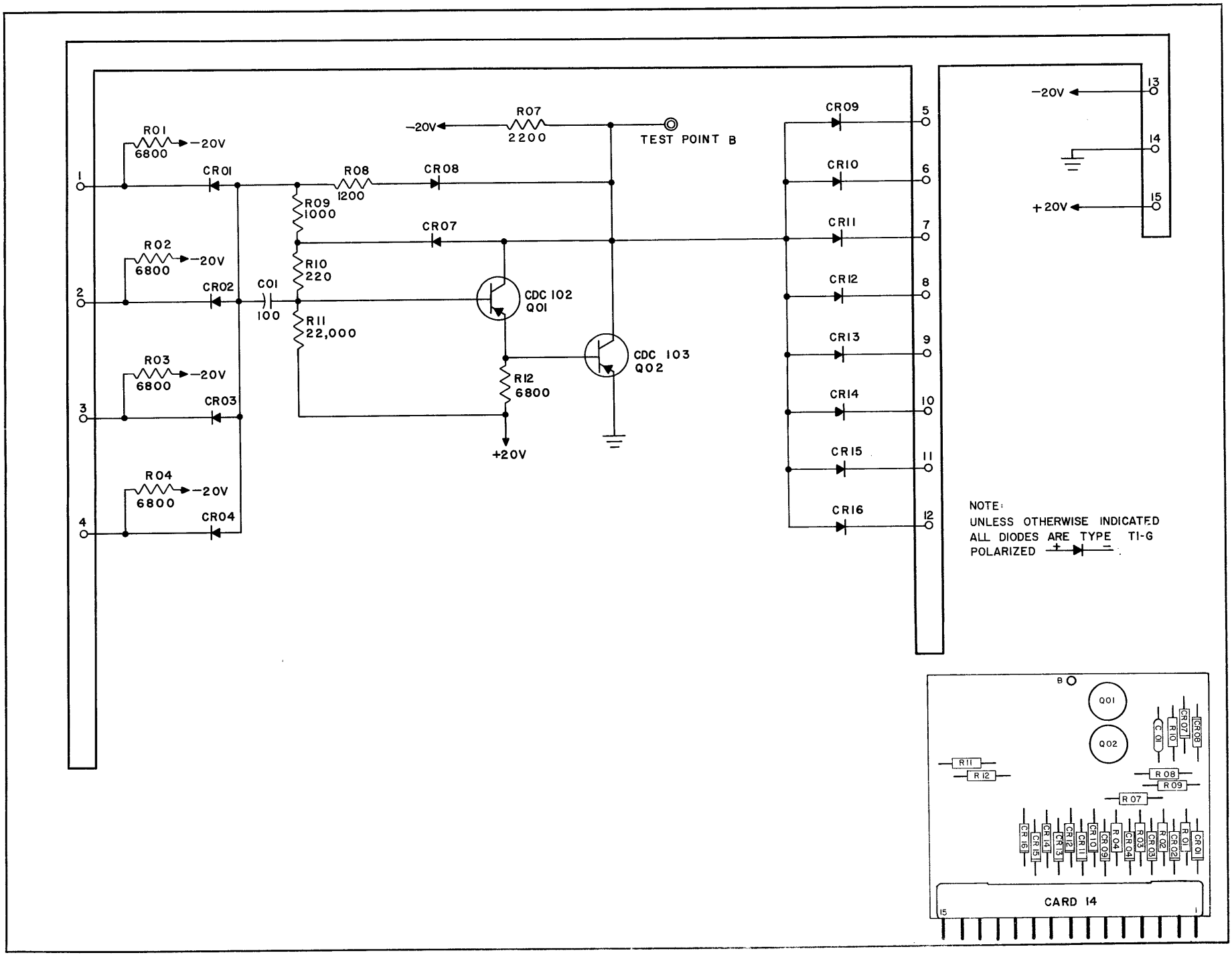
Single Inverter Card 11

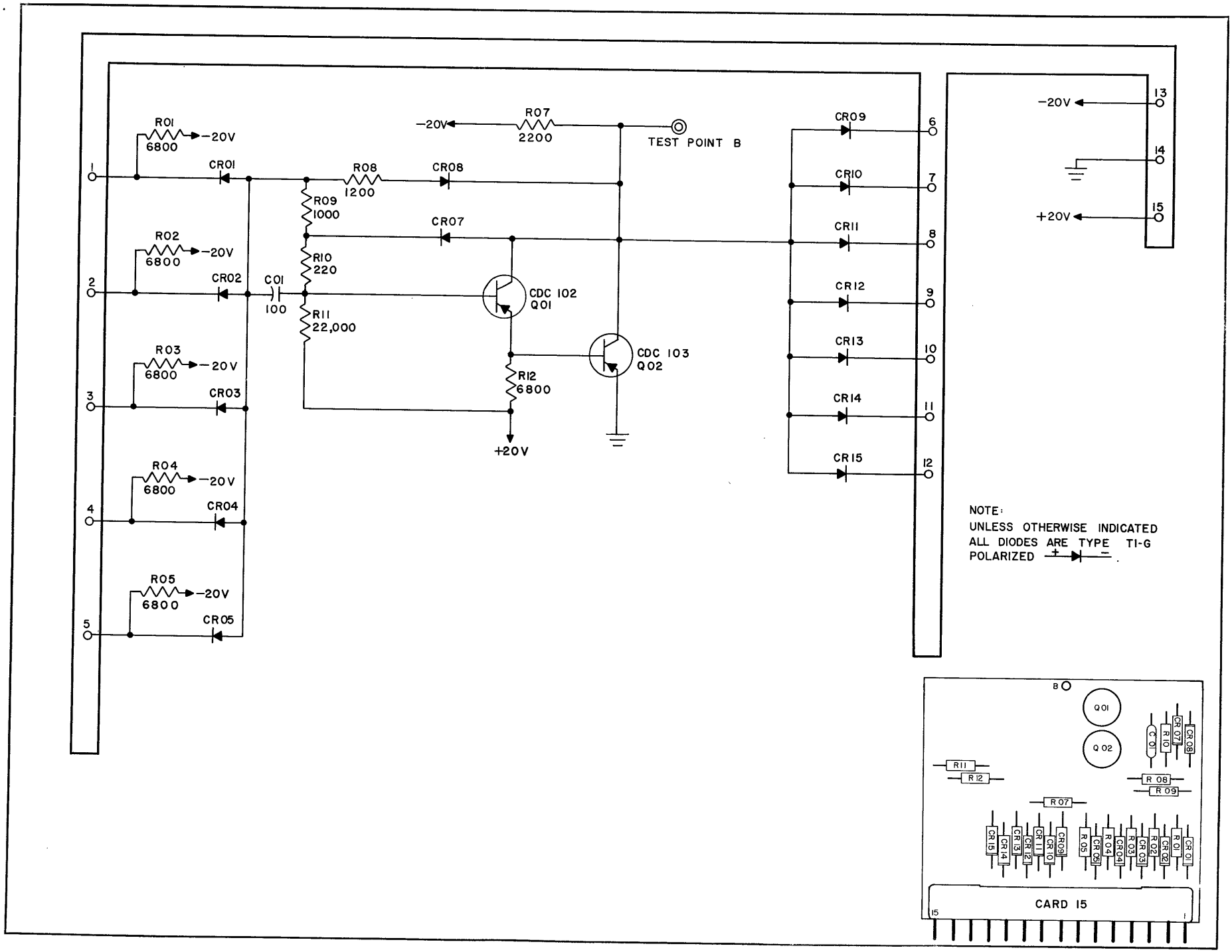


Single Inverter Card 12

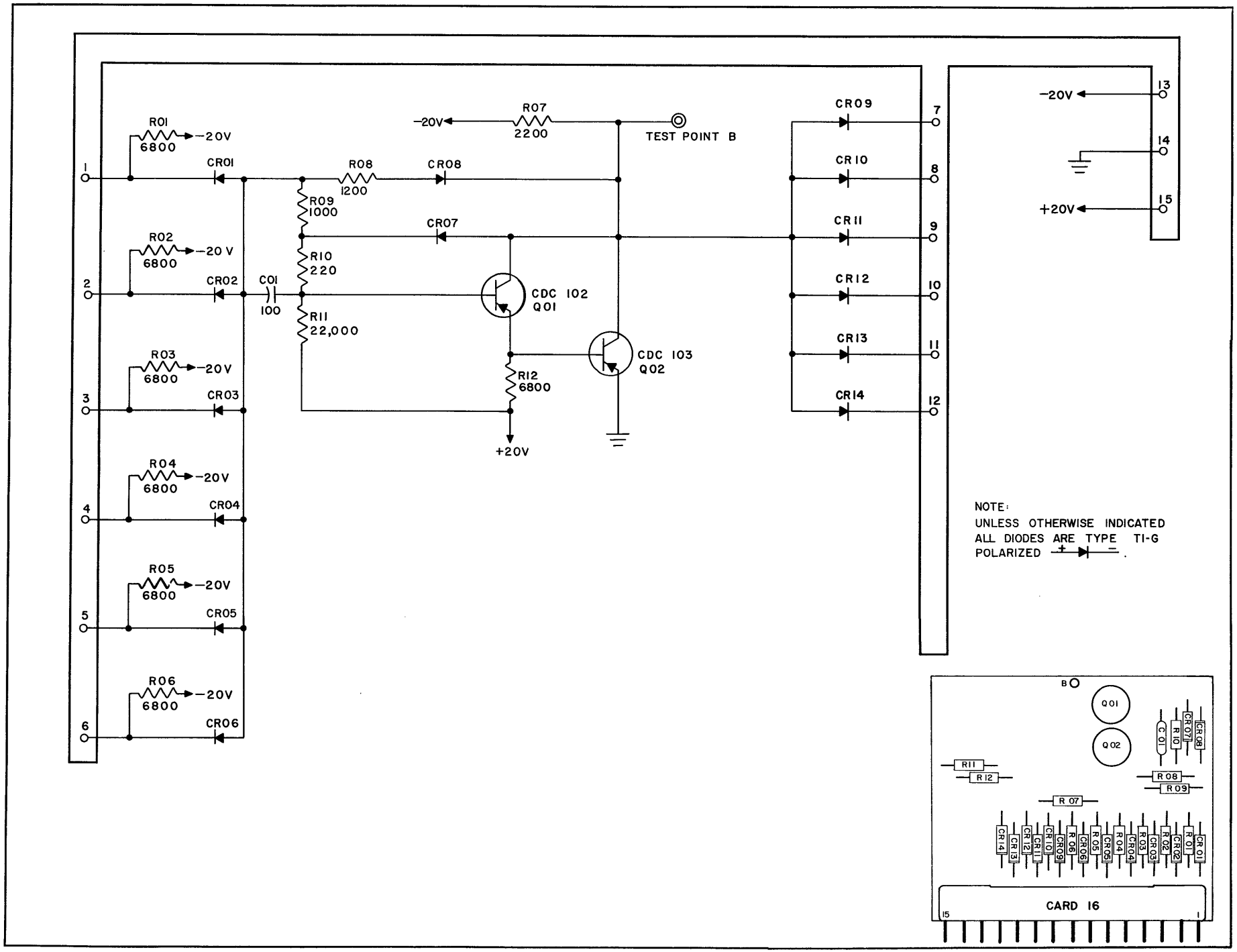


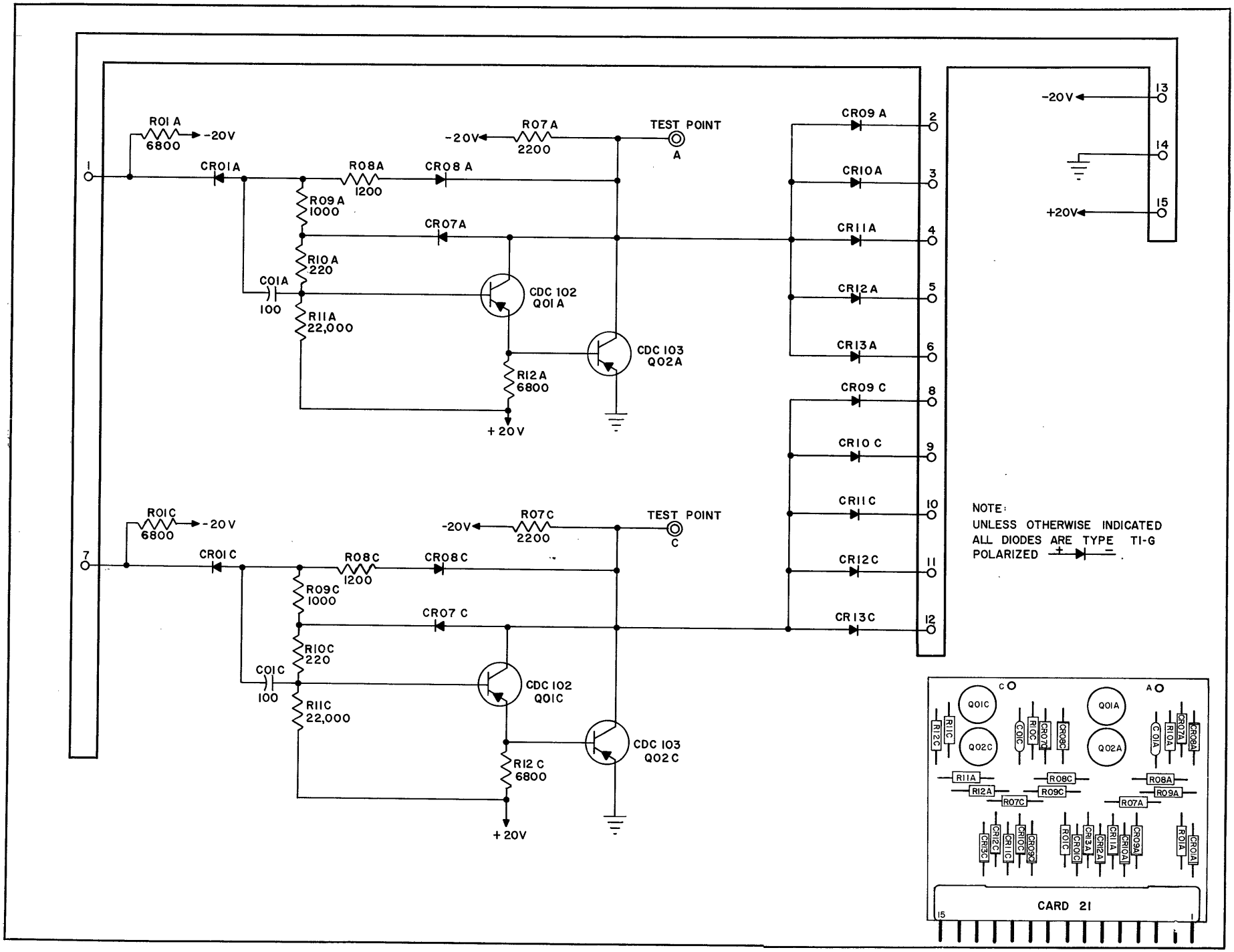
Single Inverter Card 13

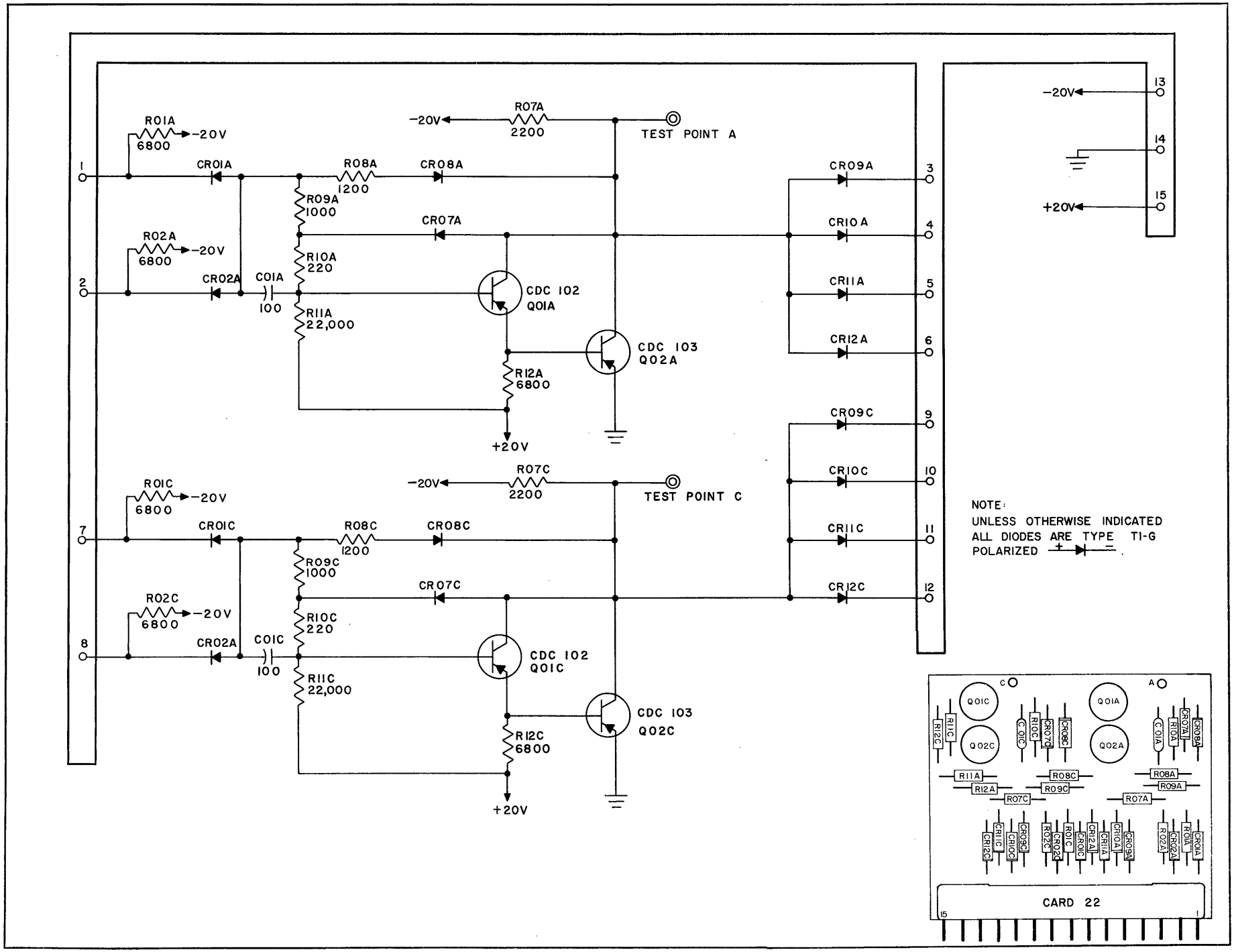

R01
6800
-20V
CR01
R02
6800
-20V
CR02
R03
6800
-20V
CR03
R04
6800
-20V
CR04
1
2
3
4
C01
100
R09
1000
R08
1200
CR08
R07
2200
-20V
TEST POINT B
CR07
R10
220
R11
22,000
CDC 102
Q01
CDC 103
Q02
R12
6800
+20V
CR09
CR10
CR11
CR12
CR13
CR14
CR15
CR16
5
6
7
8
9
10
11
12
13
14
15
-20V
+20V
NOTE:
UNLESS OTHERWISE INDICATED
ALL DIODES ARE TYPE TI-G
POLARIZED
CARD 14

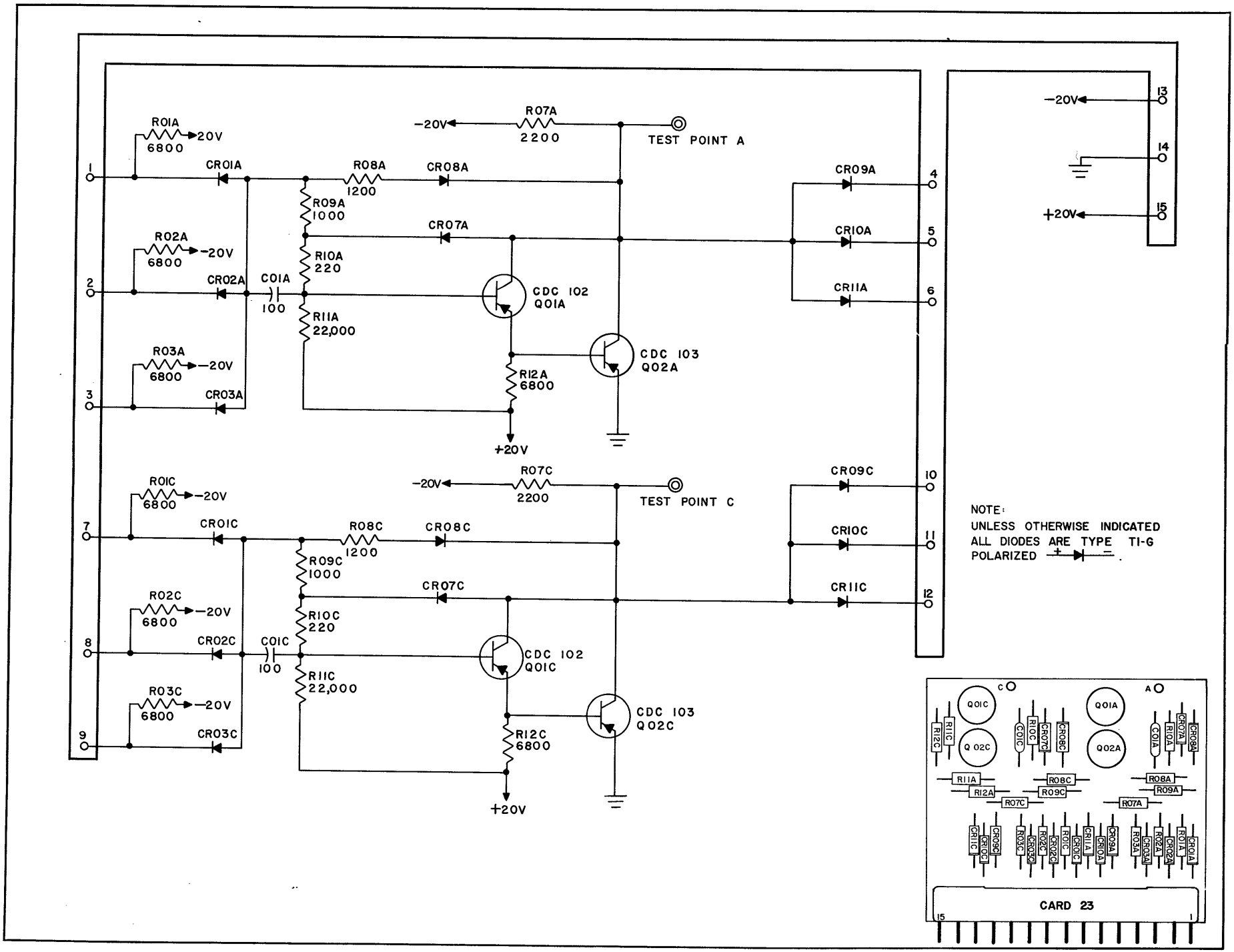


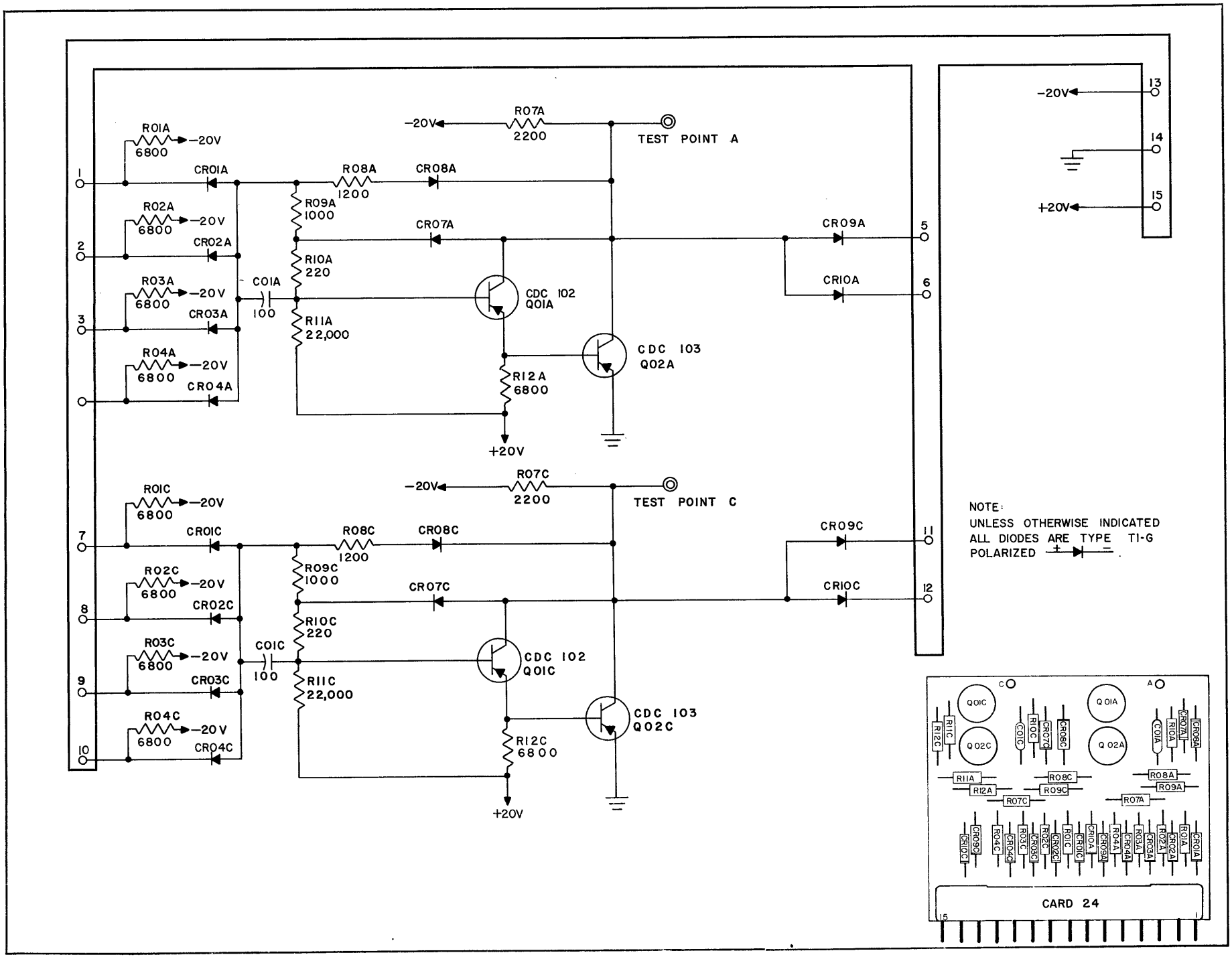
Single Inverter Card 15

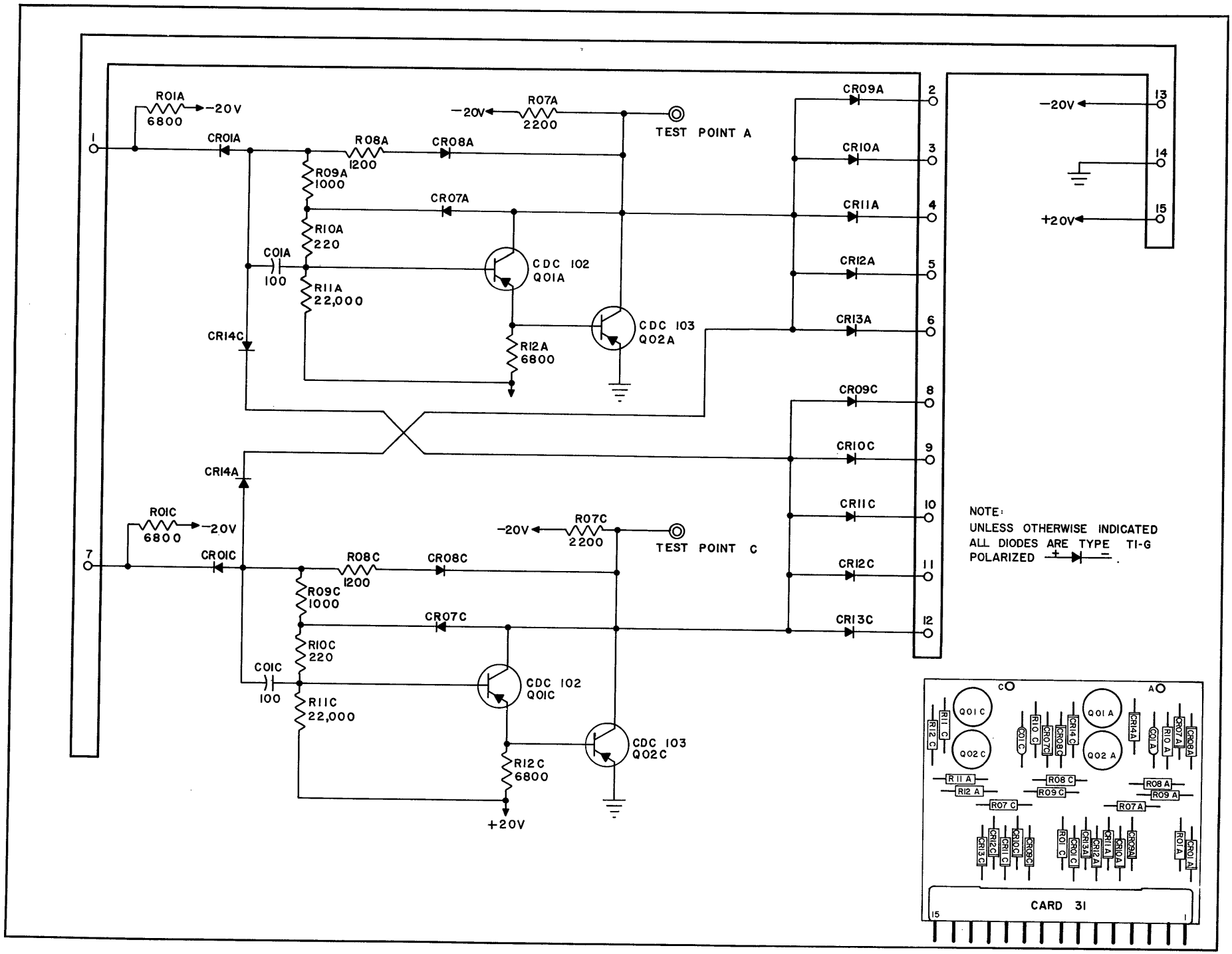

R01
6800
-20V
CR01
R08
1200
CR08
R07
2200
-20V
TEST POINT B
CR09
CR10
CR11
CR12
CR13
CR14
-20V
+20V
R09
1000
CR07
R02
6800
-20 V
R10
220
CR02
C01
100
CDC 102
Q01
R11
22,000
CDC 103
Q02
R12
6800
+20V
R03
6800
-20V
CR03
R04
6800
-20V
CR04
R05
6800
-20V
CR05
R06
6800
-20V
CR06
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
NOTE:
UNLESS OTHERWISE INDICATED
ALL DIODES ARE TYPE TI-G
POLARIZED
B
Q01
Q02
C01
R10
CR07
CR08
R11
R12
R08
R09
R07
CR14
CR13
CR12
CR11
CR10
CR09
R06
CR06
R05
CR05
R04
CR04
R03
CR03
R02
CR02
R01
CR01
CARD 16
15
1

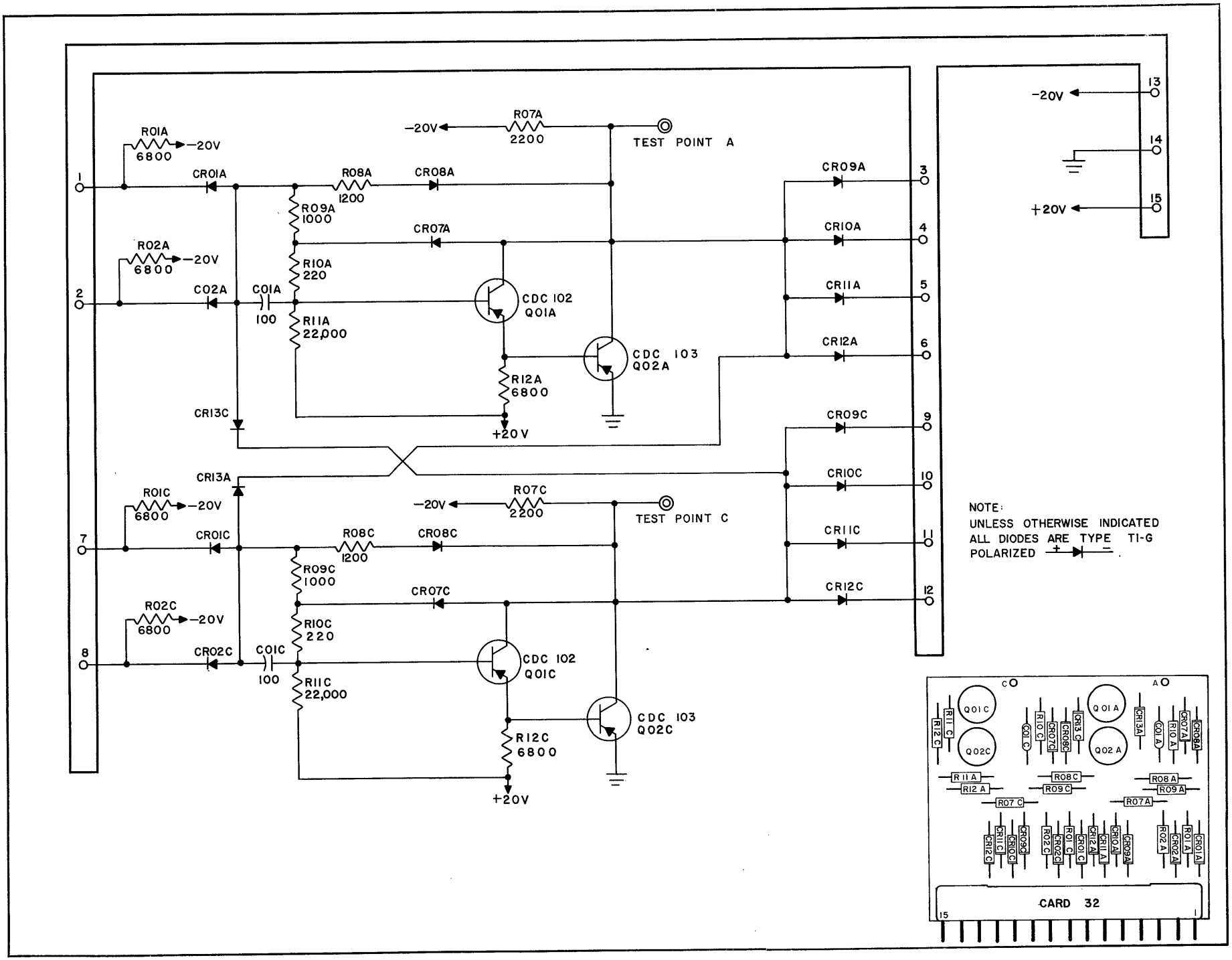

R01A
6800
-20V
CR01A
R08A
1200
CR08A
R07A
2200
TEST POINT
A
R09A
1000
CR07A
R10A
220
C01A
100
CDC 102
Q01A
R11A
22,000
CDC 103
Q02A
R12A
6800
+20V
CR09A
CR10A
CR11A
CR12A
CR13A
R01C
6800
-20V
CR01C
R08C
1200
CR08C
R07C
2200
TEST POINT
C
R09C
1000
CR07C
R10C
220
C01C
100
CDC 102
Q01C
R11C
22,000
CDC 103
Q02C
R12C
6800
+20V
CR09C
CR10C
CR11C
CR12C
CR13C
-20V
+20V
NOTE:
UNLESS OTHERWISE INDICATED
ALL DIODES ARE TYPE TI-G
POLARIZED
CARD 21

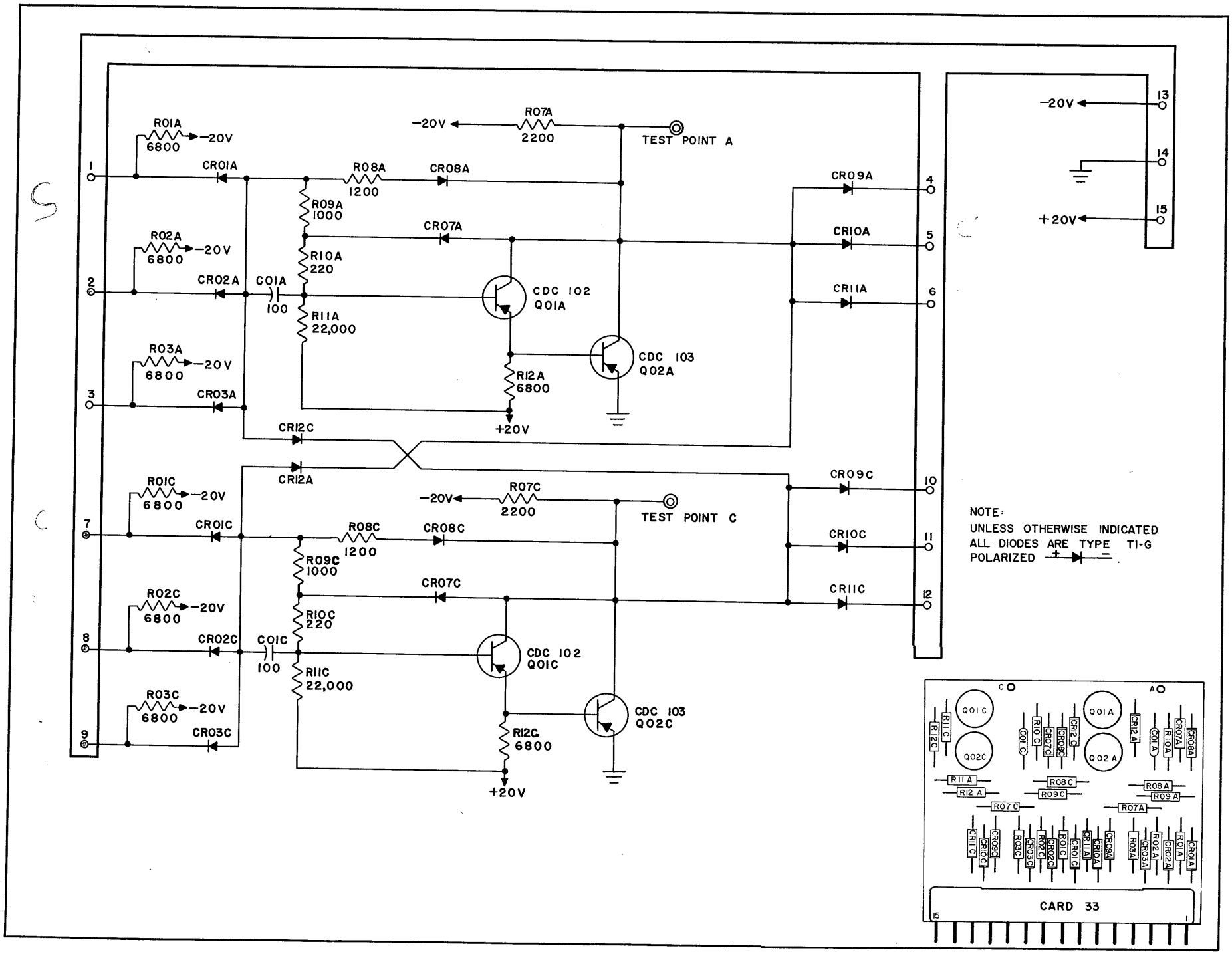

R01A
6800
-20V
CR01A
R08A
1200
CR08A
R09A
1000
R07A
2200
TEST POINT A
CR07A
R02A
6800
-20V
R10A
220
CR02A
C01A
100
R11A
22,000
CDC 102
Q01A
CDC 103
Q02A
R12A
6800
+20V
CR09A
CR10A
CR11A
CR12A
R01C
6800
-20V
CR01C
R08C
1200
CR08C
R09C
1000
R07C
2200
TEST POINT C
CR07C
R02C
6800
-20V
R10C
220
CR02A
C01C
100
R11C
22,000
CDC 102
Q01C
CDC 103
Q02C
R12C
6800
+20V
CR09C
CR10C
CR11C
CR12C
-20V
+20V
NOTE:
UNLESS OTHERWISE INDICATED
ALL DIODES ARE TYPE TI-G
POLARIZED
CARD 22

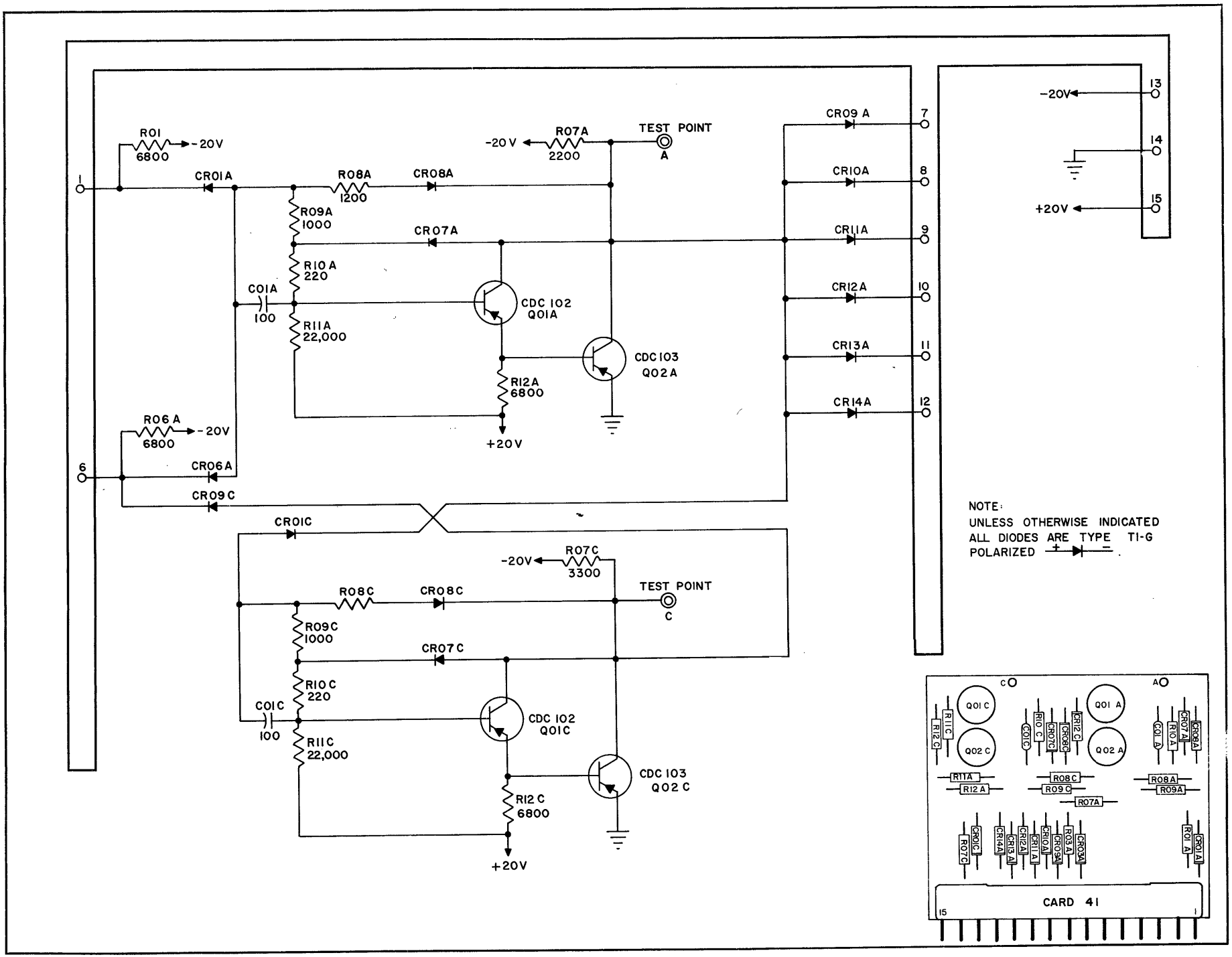

R01A
6800
20V
CR01A
R08A
1200
CR08A
R07A
2200
-20V
TEST POINT A
R09A
1000
CR07A
CR09A
CR10A
CR11A
R02A
6800
-20V
CR02A
C01A
100
R10A
220
CDC 102
Q01A
R11A
22,000
CDC 103
Q02A
R03A
6800
-20V
CR03A
R12A
6800
+20V
R01C
6800
-20V
CR01C
R08C
1200
CR08C
R07C
2200
-20V
TEST POINT C
CR09C
CR10C
CR11C
R09C
1000
CR07C
R02C
6800
-20V
CR02C
C01C
100
R10C
220
CDC 102
Q01C
R11C
22,000
CDC 103
Q02C
R03C
6800
-20V
CR03C
R12C
6800
+20V
-20V
+20V
NOTE:
UNLESS OTHERWISE INDICATED
ALL DIODES ARE TYPE TI-G
POLARIZED
CARD 23

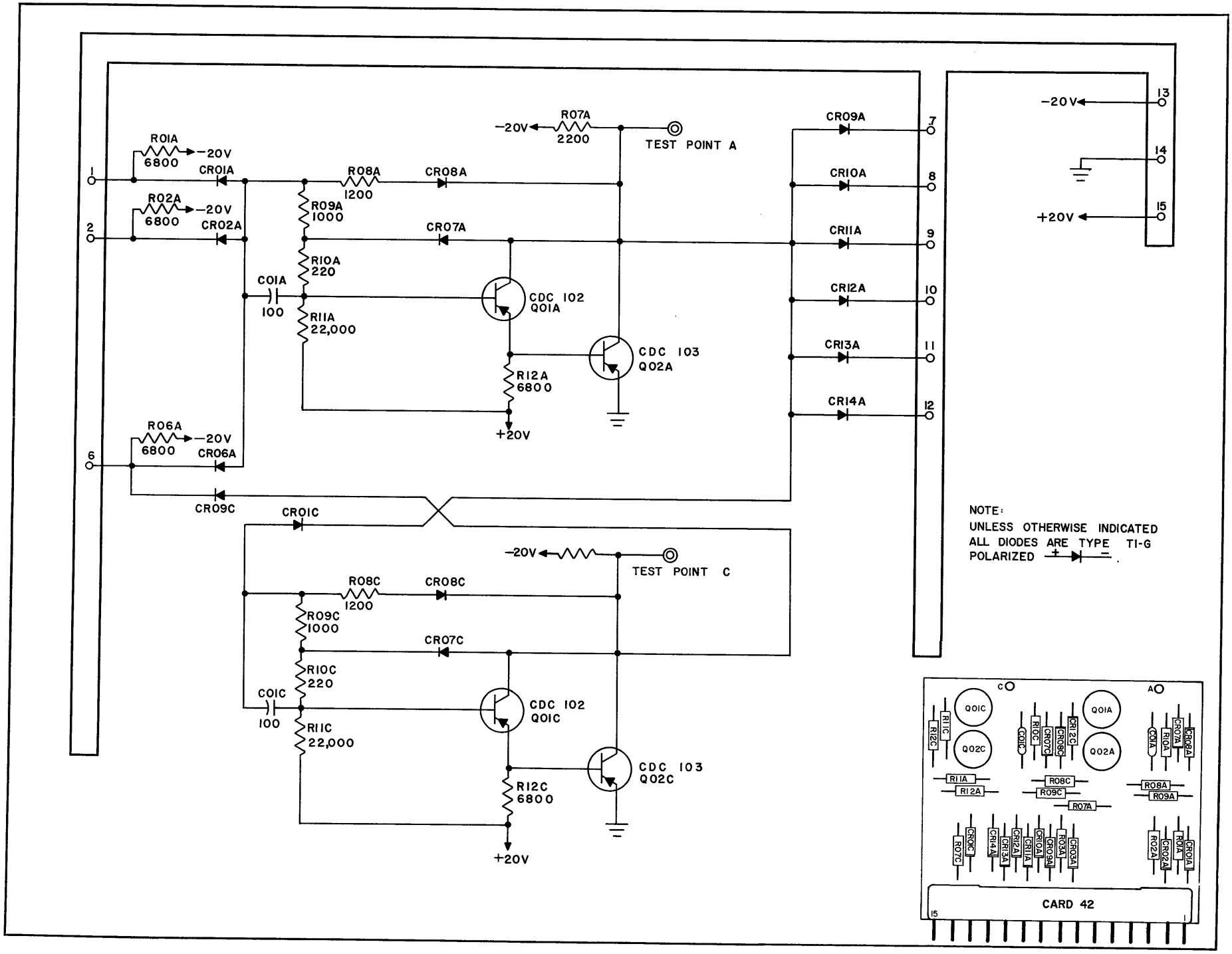

R01A
6800
-20V
CR01A
R02A
6800
-20V
CR02A
R03A
6800
-20V
CR03A
R04A
6800
-20V
CR04A
C01A
100
R09A
1000
R08A
1200
CR08A
-20V
R07A
2200
TEST POINT A
CR07A
R10A
220
R11A
22,000
CDC 102
Q01A
CDC 103
Q02A
R12A
6800
+20V
CR09A
CR10A
R01C
6800
-20V
CR01C
R02C
6800
-20V
CR02C
R03C
6800
-20V
CR03C
R04C
6800
-20V
CR04C
C01C
100
R09C
1000
R08C
1200
CR08C
-20V
R07C
2200
TEST POINT C
CR07C
R10C
220
R11C
22,000
CDC 102
Q01C
CDC 103
Q02C
R12C
6800
+20V
CR09C
CR10C
-20V
+20V
NOTE:
UNLESS OTHERWISE INDICATED
ALL DIODES ARE TYPE TI-G
POLARIZED
CARD 24

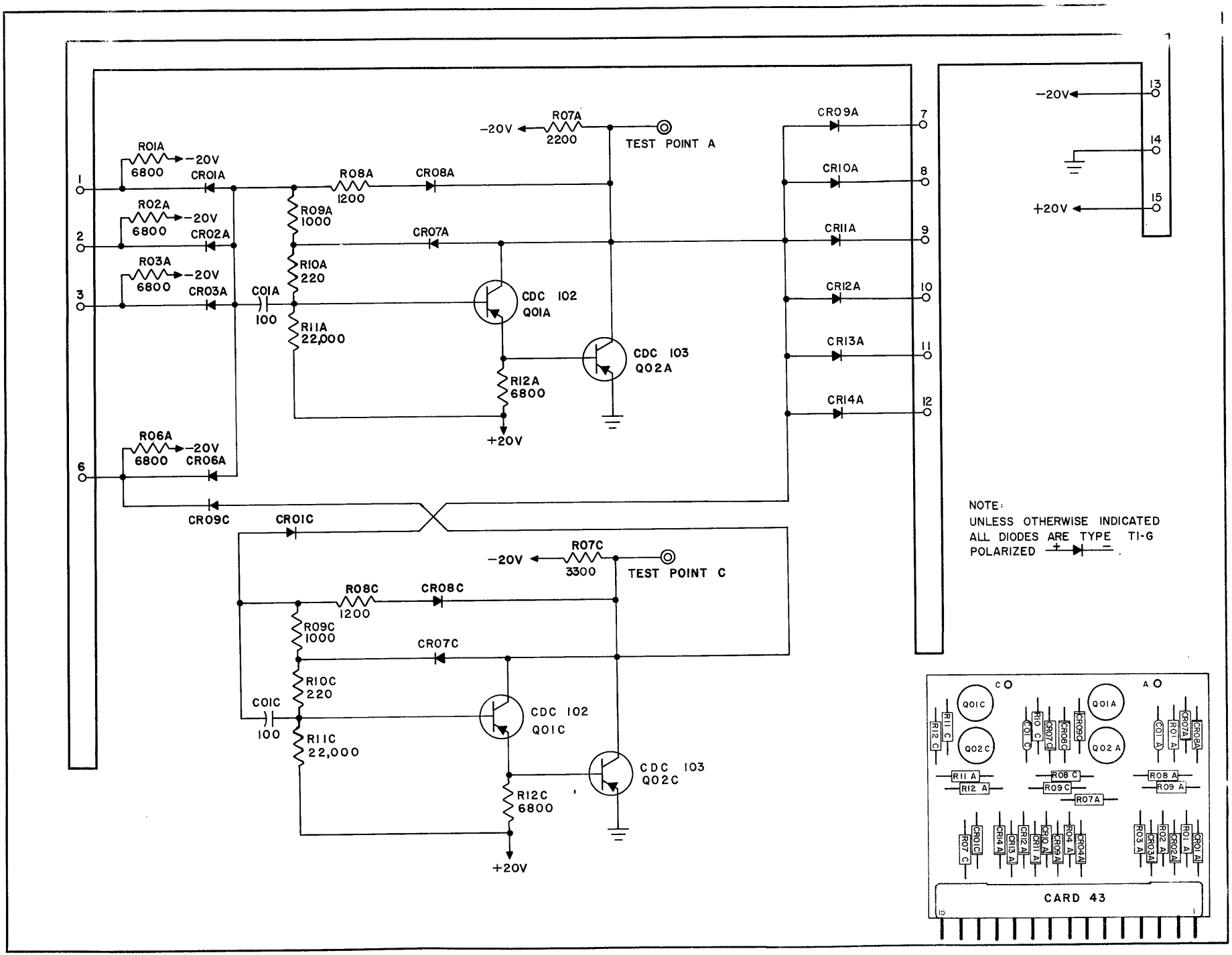

R01A
6800
-20V
CR01A
R08A
1200
CR08A
-20V
R07A
2200
TEST POINT A
CR09A
CR10A
CR11A
CR12A
CR13A
R09A
1000
CR07A
R10A
220
C01A
100
R11A
22,000
CDC 102
Q01A
CDC 103
Q02A
CR14C
R12A
6800
CR09C
CR10C
CR11C
CR12C
CR13C
CR14A
R01C
6800
-20V
CR01C
R08C
1200
CR08C
-20V
R07C
2200
TEST POINT C
R09C
1000
CR07C
R10C
220
C01C
100
R11C
22,000
CDC 102
Q01C
CDC 103
Q02C
R12C
6800
+20V
-20V
+20V
NOTE:
UNLESS OTHERWISE INDICATED
ALL DIODES ARE TYPE TI-G
POLARIZED
CARD 31

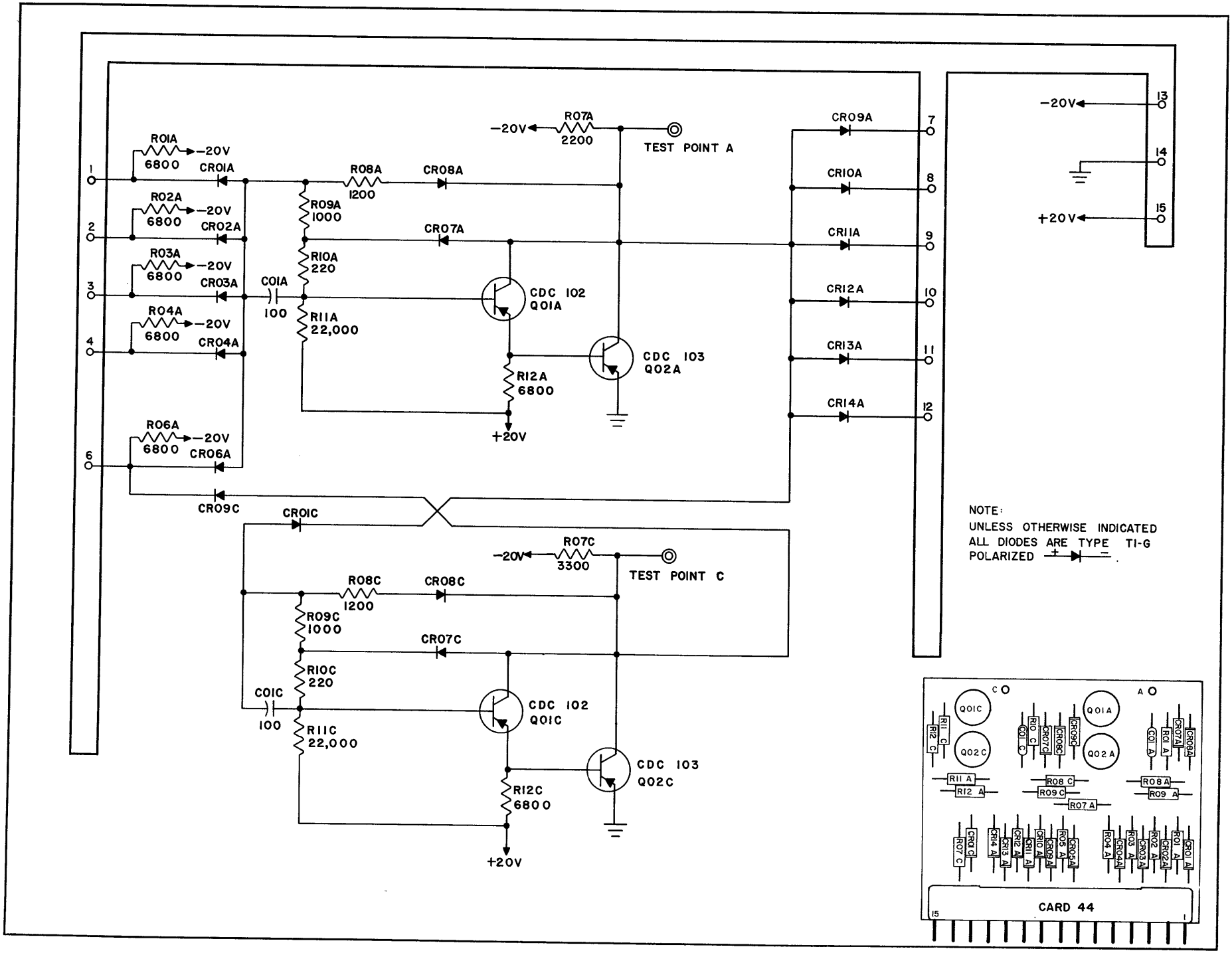

R01A
6800
-20V
CR01A
R08A
1200
CR08A
R09A
1000
R07A
2200
TEST POINT A
CR09A
CR07A
CR10A
R02A
6800
R10A
220
C02A
C01A
100
R11A
22,000
CDC 102
Q01A
CR11A
CDC 103
Q02A
CR12A
R12A
6800
+20V
CR13C
CR13A
R01C
6800
CR01C
R08C
1200
CR08C
R09C
1000
R07C
2200
TEST POINT C
CR09C
CR10C
CR11C
CR12C
CR07C
R02C
6800
R10C
220
CR02C
C01C
100
R11C
22,000
CDC 102
Q01C
CDC 103
Q02C
R12C
6800
+20V
-20V
+20V
NOTE:
UNLESS OTHERWISE INDICATED
ALL DIODES ARE TYPE TI-G
POLARIZED
CARD 32

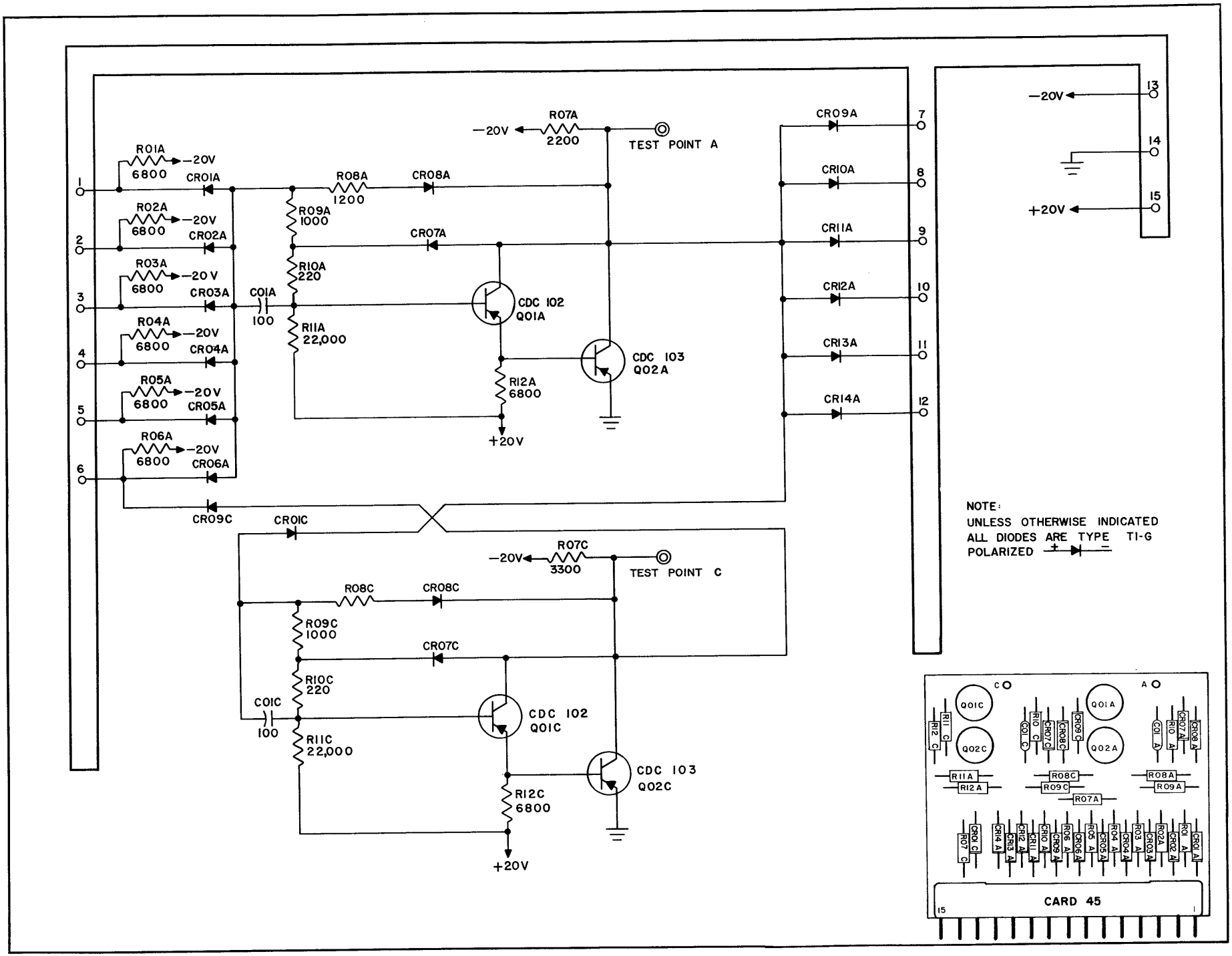

R01A
6800
-20V
CR01A
R08A
1200
CR08A
-20V
R07A
2200
TEST POINT A
R09A
1000
CR07A
R02A
6800
-20V
R10A
220
CR02A
C01A
100
CDC 102
Q01A
CR09A
CR10A
CR11A
R11A
22,000
R03A
6800
-20V
CDC 103
Q02A
CR03A
R12A
6800
+20V
CR12C
CR12A
R01C
6800
-20V
CR01C
R08C
1200
CR08C
-20V
R07C
2200
TEST POINT C
R09C
1000
CR07C
CR09C
CR10C
CR11C
R02C
6800
-20V
R10C
220
CR02C
C01C
100
CDC 102
Q01C
R11C
22,000
R03C
6800
-20V
CDC 103
Q02C
CR03C
R12C
6800
+20V
-20V
+20V
NOTE:
UNLESS OTHERWISE INDICATED
ALL DIODES ARE TYPE TI-G
POLARIZED
CARD 33

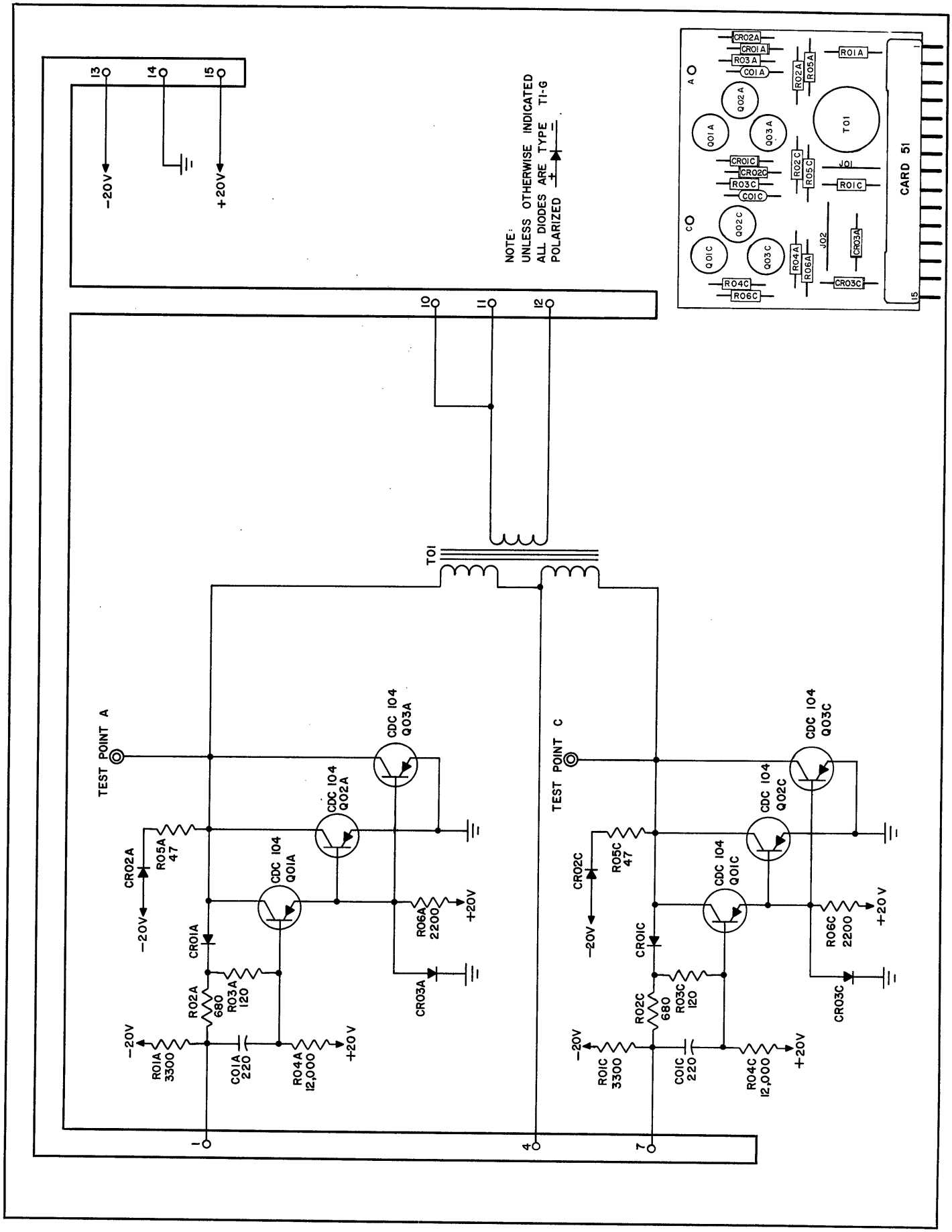

R01
6800
-20V
CR01A
R08A
1200
CR08A
R09A
1000
CR07A
R10A
220
C01A
100
CDC 102
Q01A
R11A
22,000
CDC 103
Q02A
R12A
6800
+20V
R07A
2200
TEST POINT
A
CR09A
CR10A
CR11A
CR12A
CR13A
CR14A
7
8
9
10
11
12
13
14
15
-20V
+20V
R06A
6800
-20V
CR06A
CR09C
6
1
CR01C
R07C
3300
-20V
TEST POINT
C
R08C
CR08C
R09C
1000
CR07C
R10C
220
C01C
100
CDC 102
Q01C
R11C
22,000
CDC 103
Q02C
R12C
6800
+20V
NOTE:
UNLESS OTHERWISE INDICATED
ALL DIODES ARE TYPE TI-G
POLARIZED
CO
AO
Q01C
Q02C
Q01A
Q02A
CARD 41
15
1


R01A
6800
-20V
CR01A
R02A
6800
-20V
CR02A
R06A
6800
-20V
CR06A
CR09C
R07A
2200
-20V
TEST POINT A
R08A
1200
CR08A
R09A
1000
CR07A
R10A
220
C01A
100
R11A
22,000
CDC 102
Q01A
CDC 103
Q02A
R12A
6800
+20V
CR09A
CR10A
CR11A
CR12A
CR13A
CR14A
1
2
6
7
8
9
10
11
12
13
14
15
-20V
+20V
CR01C
-20V
TEST POINT C
R08C
1200
CR08C
R09C
1000
CR07C
R10C
220
C01C
100
R11C
22,000
CDC 102
Q01C
CDC 103
Q02C
R12C
6800
+20V
NOTE:
UNLESS OTHERWISE INDICATED
ALL DIODES ARE TYPE TI-G
POLARIZED
C
A
Q01C
Q02C
Q01A
Q02A
CARD 42
15

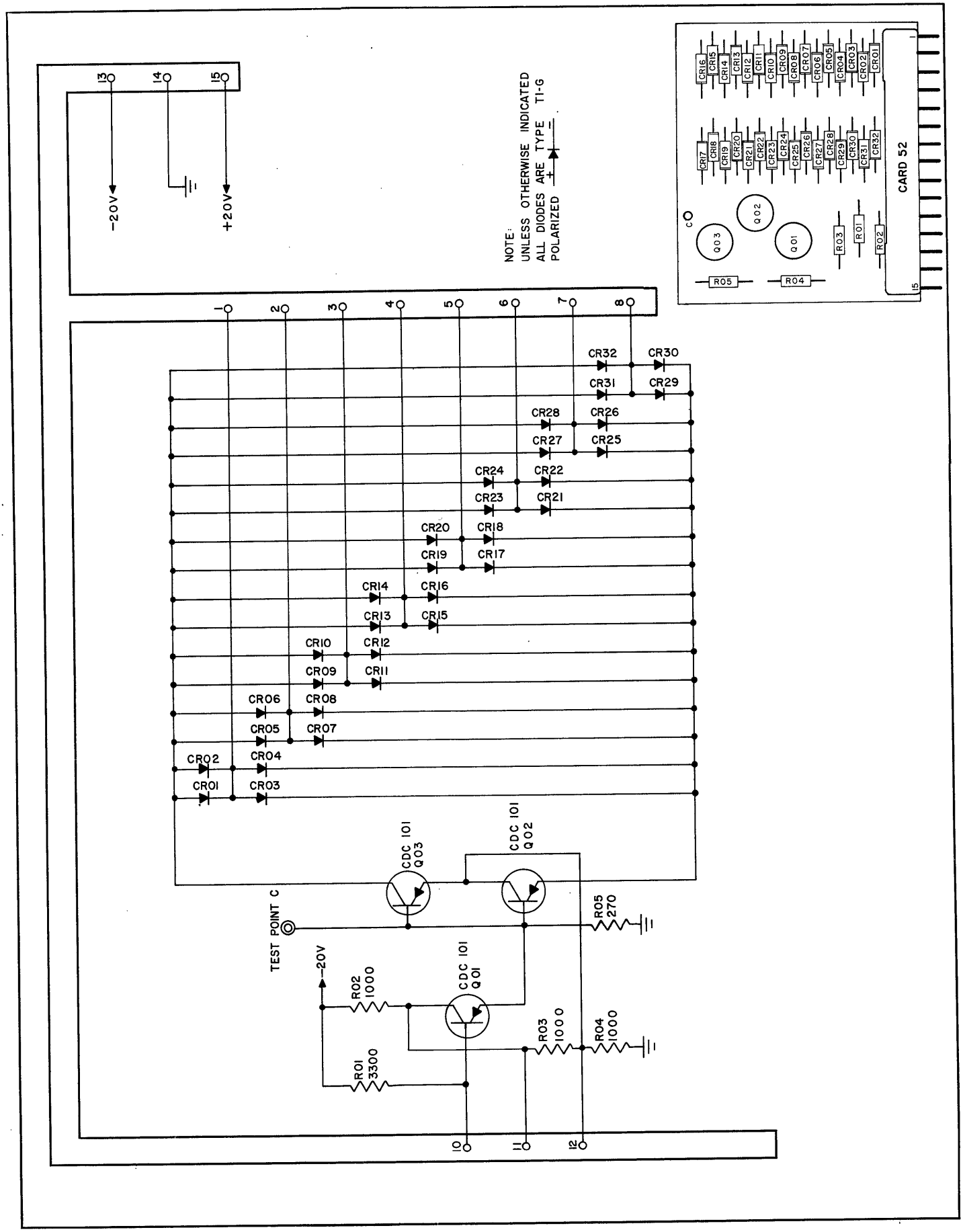

R01A
6800
-20V
CR01A
R02A
6800
-20V
CR02A
R03A
6800
-20V
CR03A
R06A
6800
-20V
CR06A
C01A
100
R07A
2200
-20V
TEST POINT A
R08A
1200
CR08A
R09A
1000
CR07A
R10A
220
R11A
22,000
CDC 102
Q01A
CDC 103
Q02A
R12A
6800
+20V
CR09A
CR10A
CR11A
CR12A
CR13A
CR14A
1
2
3
6
7
8
9
10
11
12
13
14
15
-20V
+20V
CR09C
CR01C
R07C
3300
-20V
TEST POINT C
R08C
1200
CR08C
R09C
1000
CR07C
R10C
220
C01C
100
R11C
22,000
CDC 102
Q01C
CDC 103
Q02C
R12C
6800
+20V
NOTE:
UNLESS OTHERWISE INDICATED
ALL DIODES ARE TYPE TI-G
POLARIZED
C
A
Q01C
Q02C
Q01A
Q02A
CARD 43


R07A
2200
-20V
TEST POINT A
R01A
6800
CR01A
R08A
1200
CR08A
R09A
1000
R02A
6800
CR02A
CR07A
R10A
220
R03A
6800
CR03A
C01A
100
CDC 102
Q01A
R11A
22,000
R04A
6800
CR04A
CDC 103
Q02A
R12A
6800
+20V
R06A
6800
CR06A
CR09C
CR01C
R07C
3300
TEST POINT C
R08C
1200
CR08C
R09C
1000
CR07C
R10C
220
C01C
100
CDC 102
Q01C
R11C
22,000
CDC 103
Q02C
R12C
6800
+20V
CR09A
CR10A
CR11A
CR12A
CR13A
CR14A
-20V
+20V
NOTE:
UNLESS OTHERWISE INDICATED
ALL DIODES ARE TYPE TI-G
POLARIZED
CARD 44


R01A
6800
-20V
CR01A
R02A
6800
-20V
CR02A
R03A
6800
-20V
CR03A
R04A
6800
-20V
CR04A
R05A
6800
-20V
CR05A
R06A
6800
-20V
CR06A
CR09C
R08A
1200
CR08A
R09A
1000
CR07A
R10A
220
C01A
100
R11A
22,000
CDC 102
Q01A
CDC 103
Q02A
R12A
6800
+20V
-20V
R07A
2200
TEST POINT A
CR09A
CR10A
CR11A
CR12A
CR13A
CR14A
7
8
9
10
11
12
13
14
15
-20V
+20V
CR01C
R07C
3300
TEST POINT C
R08C
CR08C
R09C
1000
CR07C
R10C
220
C01C
100
R11C
22,000
CDC 102
Q01C
CDC 103
Q02C
R12C
6800
+20V
NOTE:
UNLESS OTHERWISE INDICATED
ALL DIODES ARE TYPE TI-G
POLARIZED
CARD 45

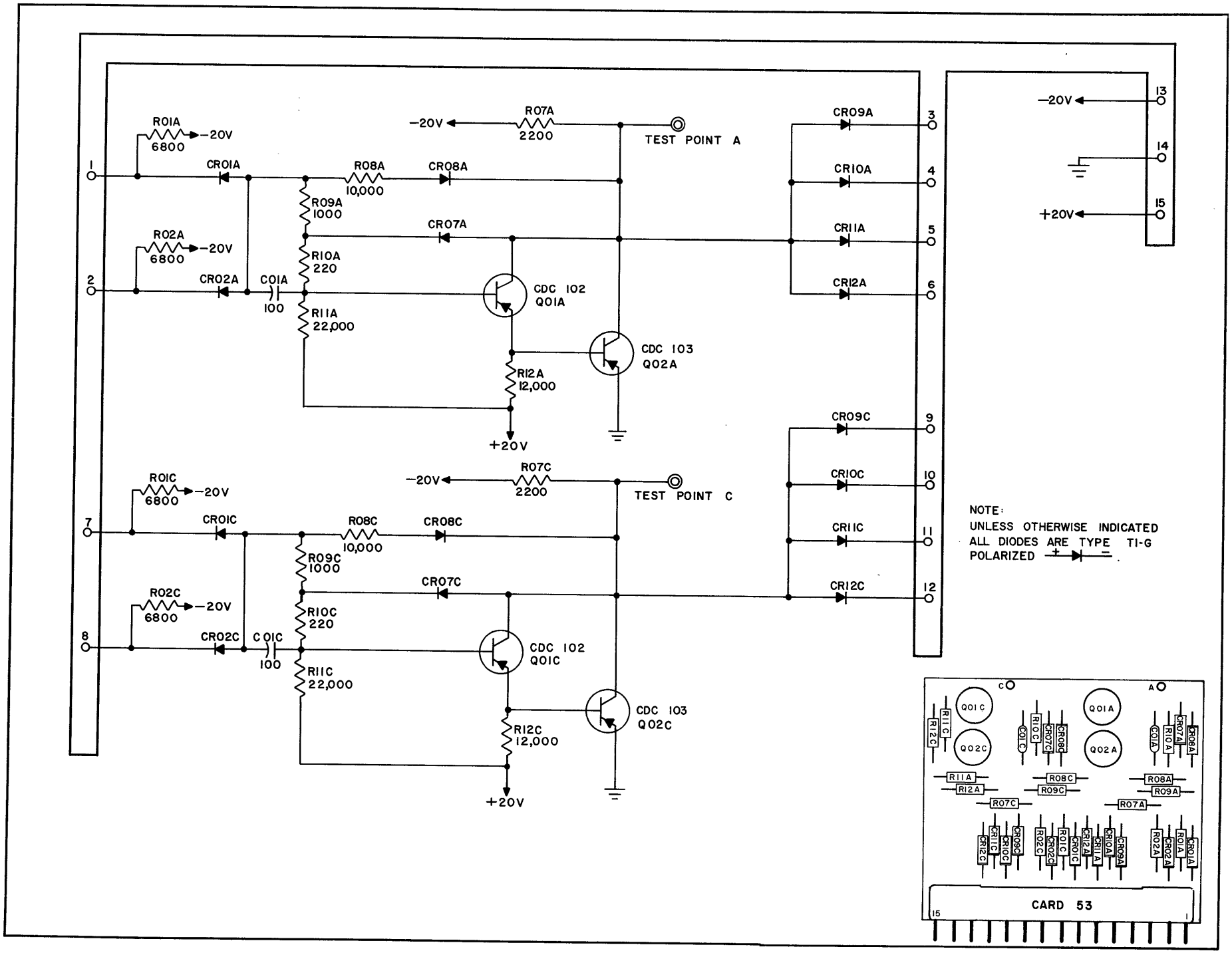


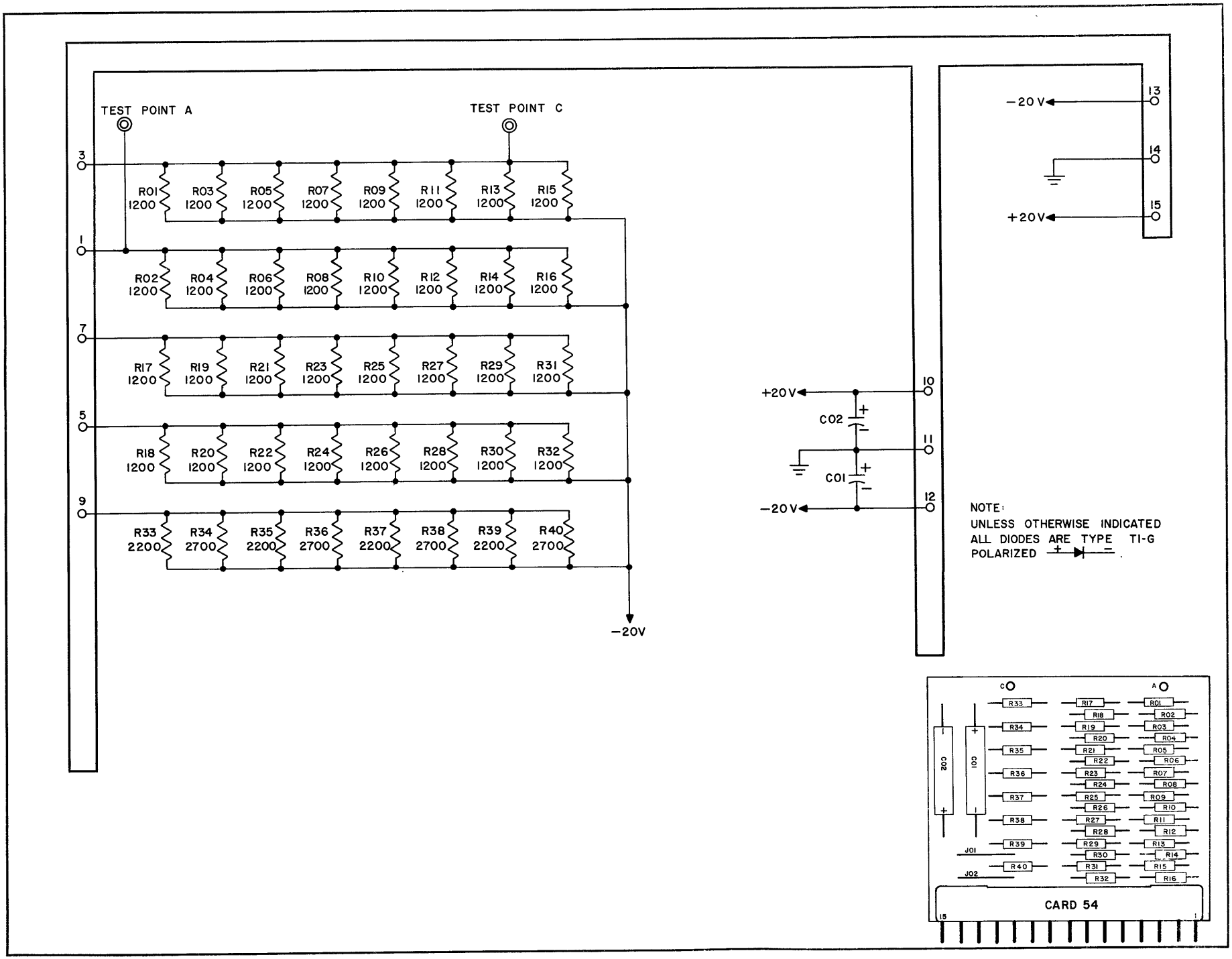
Drive Generator Card 51

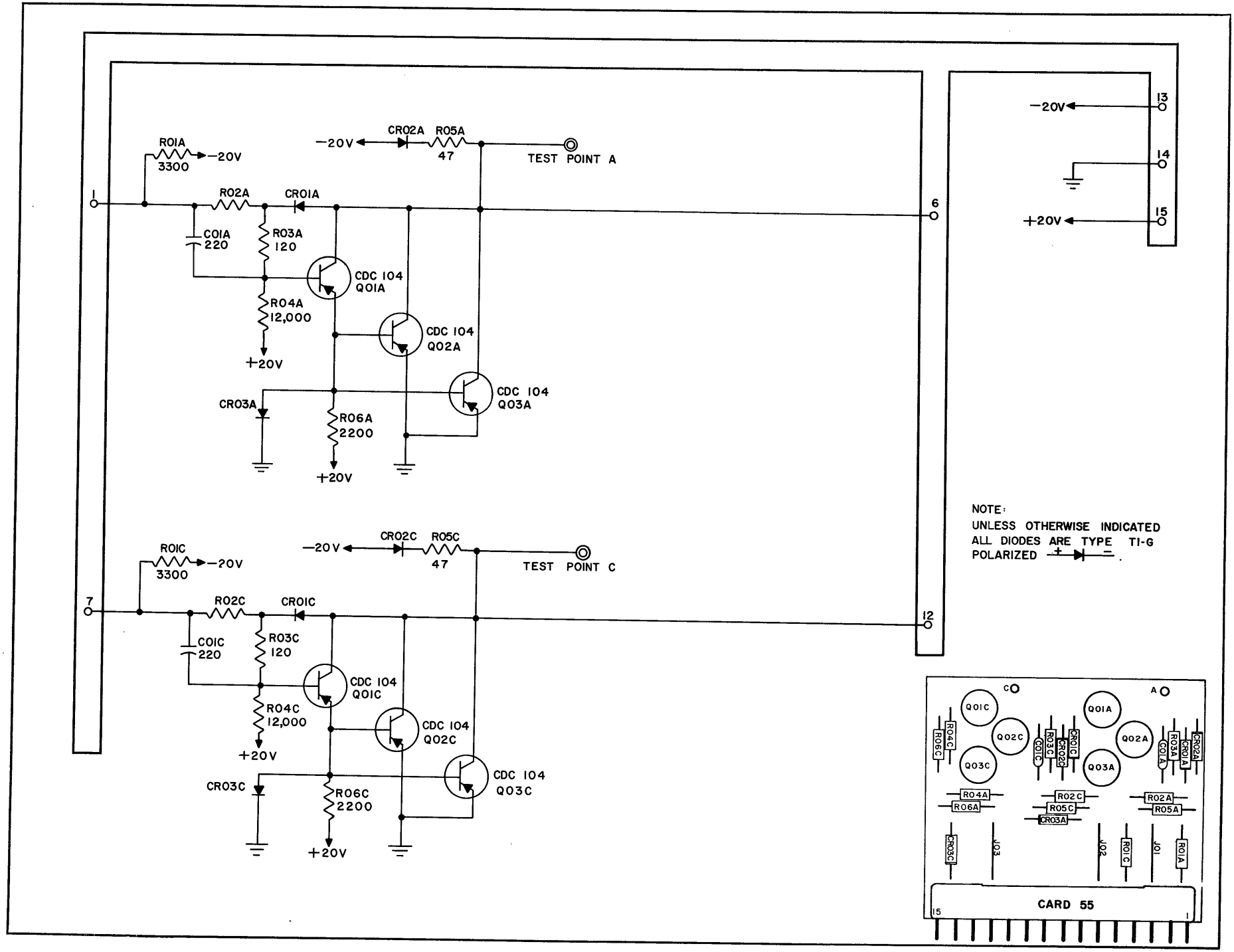


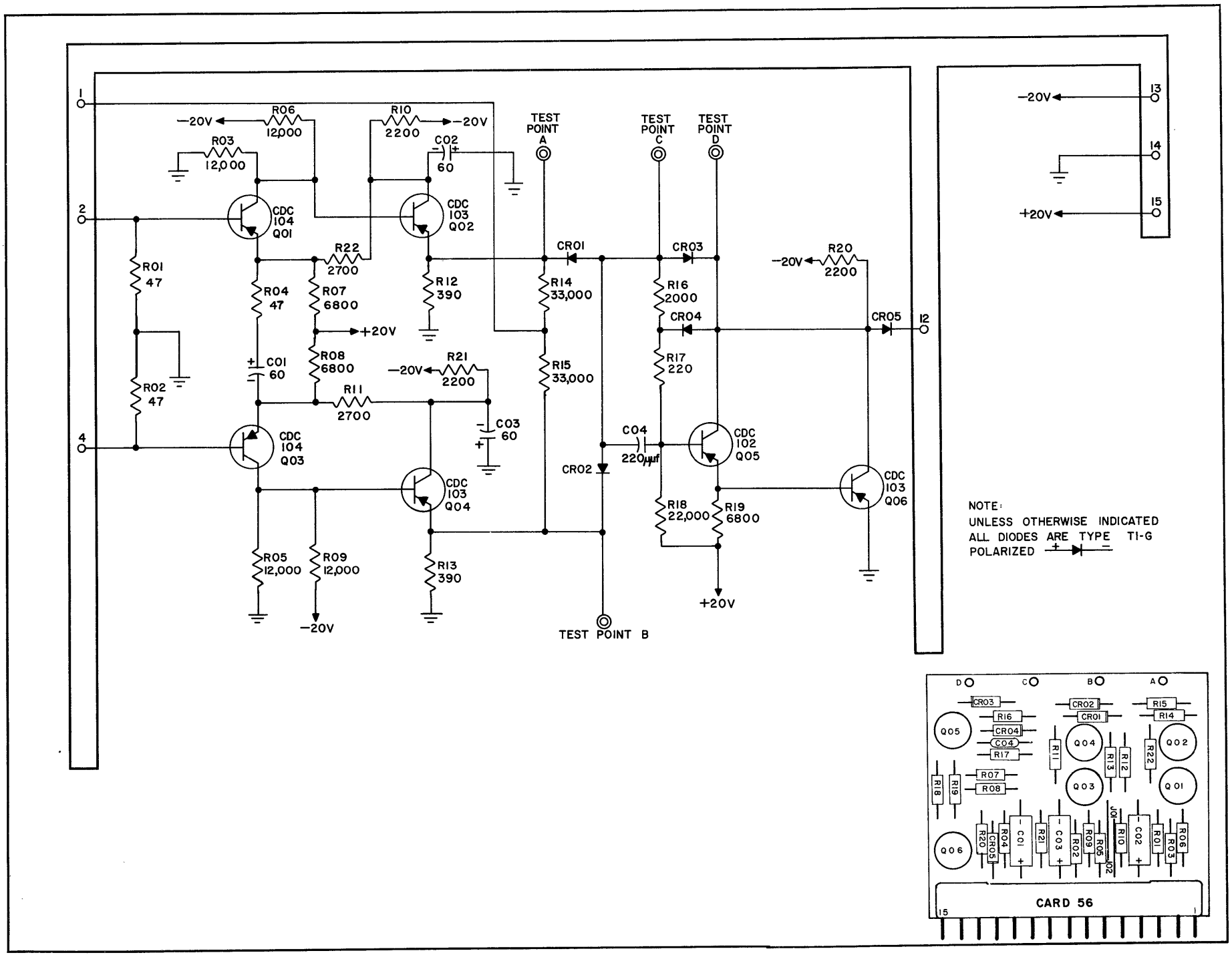
Diverter Card 52

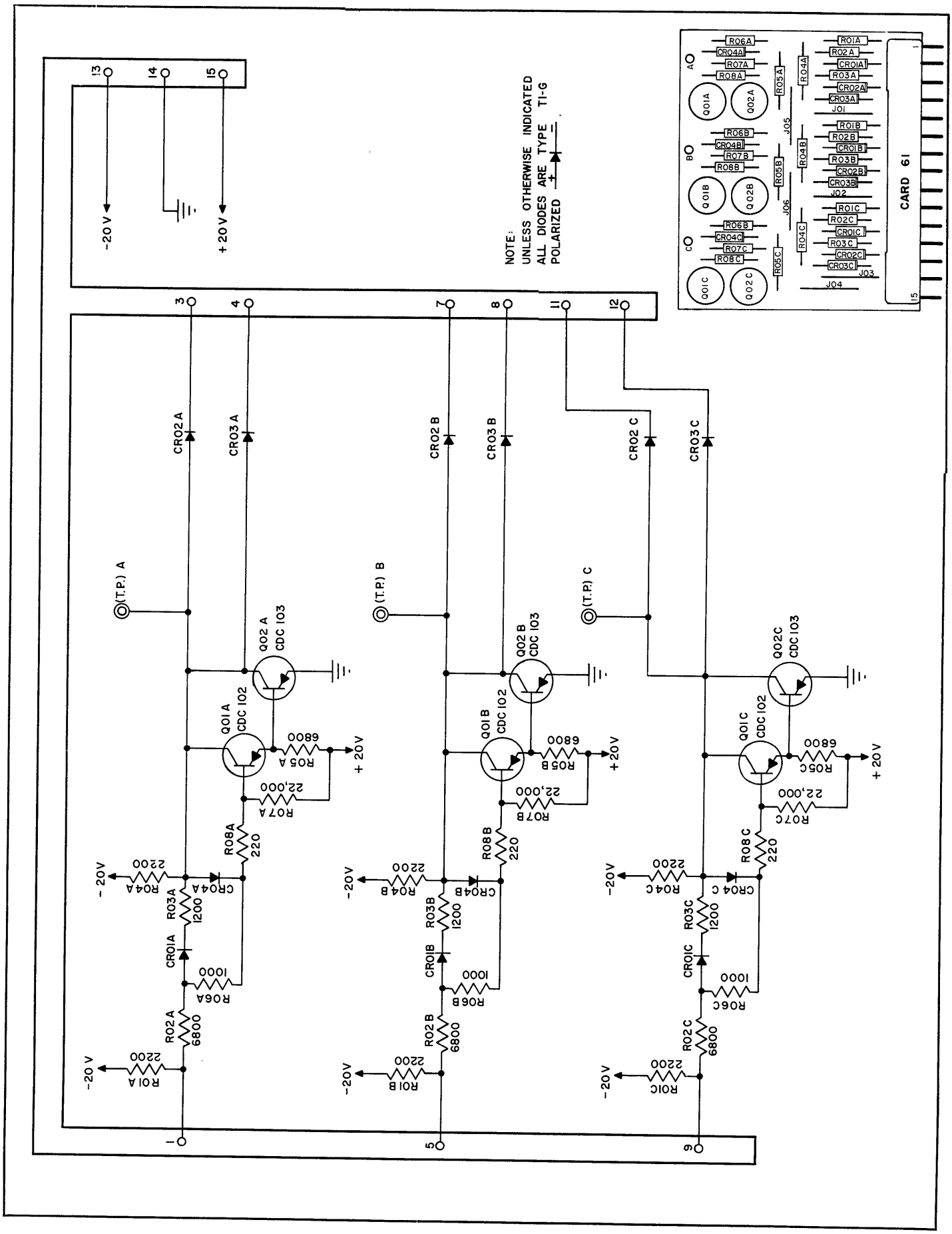
Rev. 12/60


R01A
6800
-20V
CR01A
R08A
10,000
CR08A
R09A
1000
R07A
2200
TEST POINT A
CR09A
CR10A
CR11A
CR12A
R02A
6800
CR02A
C01A
100
R10A
220
CR07A
R11A
22,000
CDC 102
Q01A
CDC 103
Q02A
R12A
12,000
+20V
R01C
6800
CR01C
R08C
10,000
CR08C
R09C
1000
R07C
2200
TEST POINT C
CR09C
CR10C
CR11C
CR12C
R02C
6800
CR02C
C01C
100
R10C
220
CR07C
R11C
22,000
CDC 102
Q01C
CDC 103
Q02C
R12C
12,000
+20V
NOTE:
UNLESS OTHERWISE INDICATED
ALL DIODES ARE TYPE TI-G
POLARIZED
CARD 53

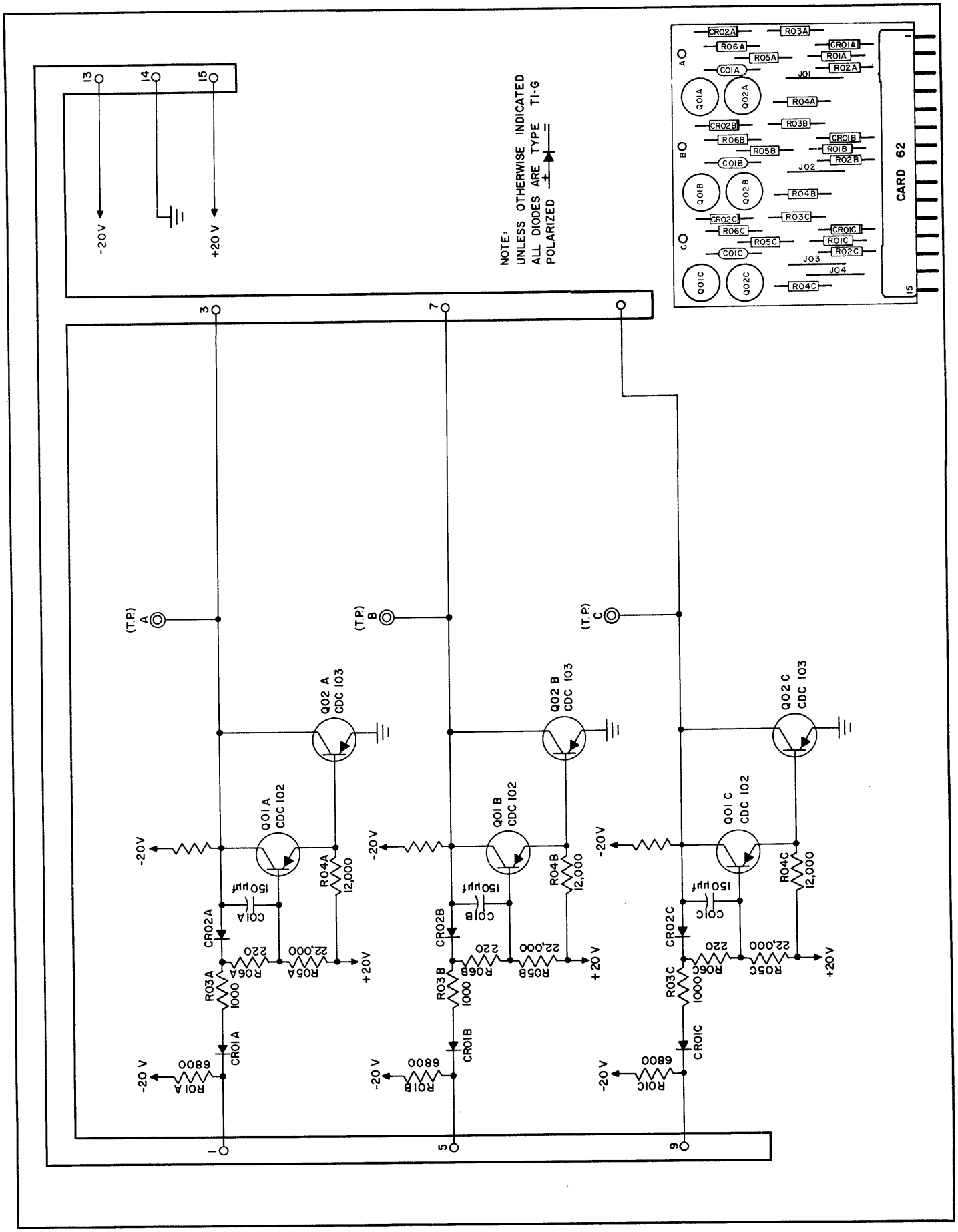

TEST POINT A
TEST POINT C
3
1
7
5
9
R01 1200
R03 1200
R05 1200
R07 1200
R09 1200
R11 1200
R13 1200
R15 1200
R02 1200
R04 1200
R06 1200
R08 1200
R10 1200
R12 1200
R14 1200
R16 1200
R17 1200
R19 1200
R21 1200
R23 1200
R25 1200
R27 1200
R29 1200
R31 1200
R18 1200
R20 1200
R22 1200
R24 1200
R26 1200
R28 1200
R30 1200
R32 1200
R33 2200
R34 2700
R35 2200
R36 2700
R37 2200
R38 2700
R39 2200
R40 2700
-20V
-20V
13
+20V
14
15
+20V
10
C02
11
C01
12
-20V
NOTE:
UNLESS OTHERWISE INDICATED
ALL DIODES ARE TYPE TI-G
POLARIZED
C
A
C02
C01
J01
J02
CARD 54
15


R01A
3300
-20V
CR02A
R05A
47
TEST POINT A
1
R02A
CR01A
C01A
220
R03A
120
CDC 104
Q01A
R04A
12,000
+20V
CDC 104
Q02A
CR03A
R06A
2200
CDC 104
Q03A
6
-20V
13
14
15
+20V
NOTE:
UNLESS OTHERWISE INDICATED
ALL DIODES ARE TYPE TI-G
POLARIZED
R01C
3300
-20V
CR02C
R05C
47
TEST POINT C
7
R02C
CR01C
C01C
220
R03C
120
CDC 104
Q01C
R04C
12,000
+20V
CDC 104
Q02C
CR03C
R06C
2200
CDC 104
Q03C
12
C
A
Q01C
Q01A
Q02C
Q02A
Q03C
Q03A
R04C
R06C
C01C
R03C
CR02C
CR01C
C01A
R03A
CR01A
CR02A
R04A
R06A
R02C
R05C
CR03A
R02A
R05A
CR03C
J03
J02
R01C
J01
R01A
CARD 55
15
1

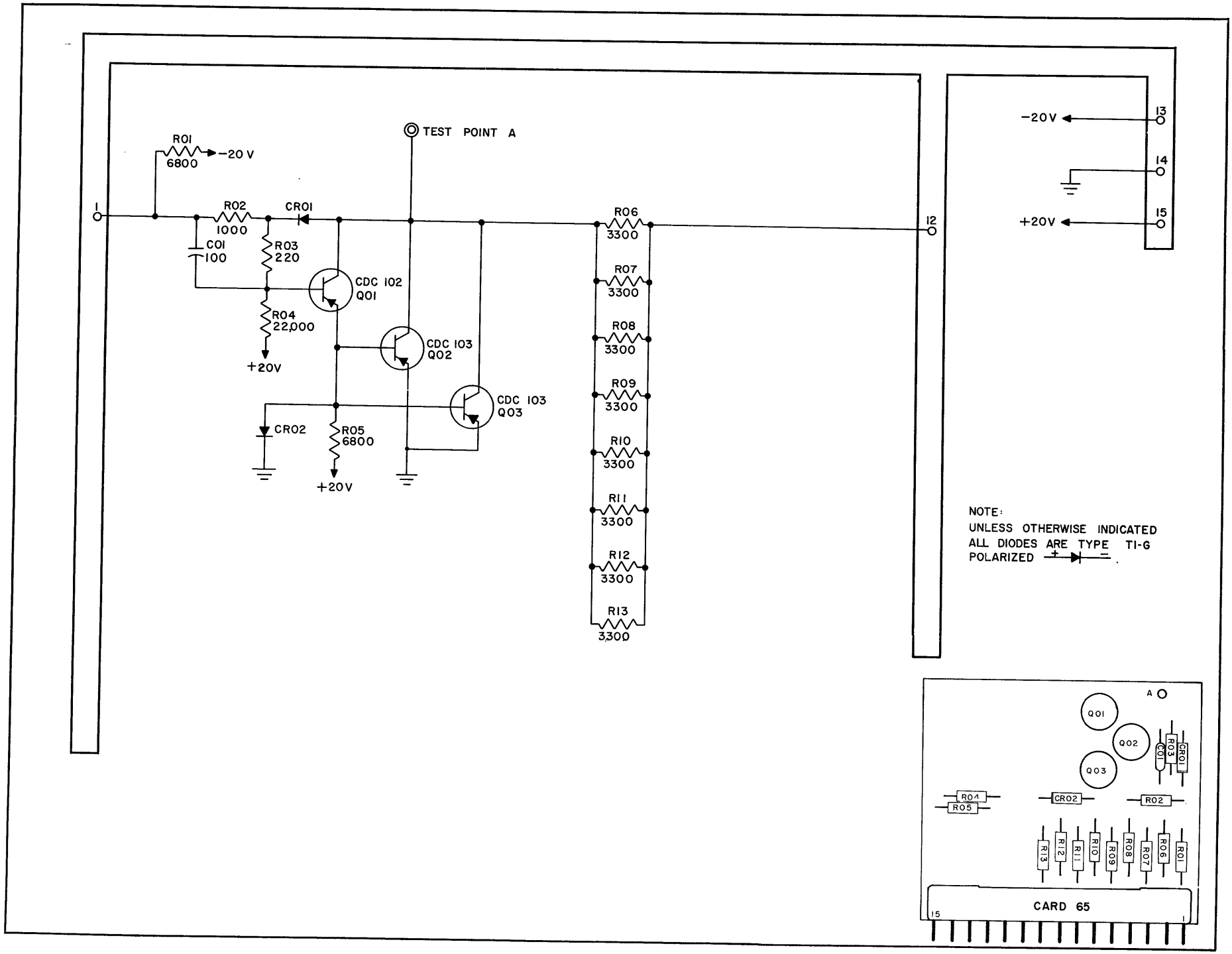

1
2
4
R06
12,000
-20V
R10
2200
-20V
R03
12,000
C02
60
TEST POINT A
TEST POINT C
TEST POINT D
CDC 104 Q01
CDC 103 Q02
R01
47
R22
2700
CR01
CR03
R20
-20V
2200
R04
47
R07
6800
R12
390
R14
33,000
R16
2000
CR04
CR05
12
+20V
R08
6800
C01
60
R21
-20V
2200
R15
33,000
R17
220
R02
47
R11
2700
C03
60
C04
220μμf
CDC 102 Q05
CDC 104 Q03
CR02
CDC 103 Q04
CDC 103 Q06
R18
22,000
R19
6800
R05
12,000
R09
12,000
R13
390
-20V
+20V
TEST POINT B
13
14
15
-20V
+20V
NOTE:
UNLESS OTHERWISE INDICATED
ALL DIODES ARE TYPE T1-G
POLARIZED
D
C
B
A
CR03
R16
CR04
C04
R17
R07
R08
CR02
CR01
R15
R14
Q05
Q04
Q02
Q03
Q01
Q06
R18
R19
R11
R13
R12
R22
J01
J02
R20
CR05
R04
C01
R21
C03
R02
R09
R05
R10
C02
R01
R03
R06
CARD 56
15
1

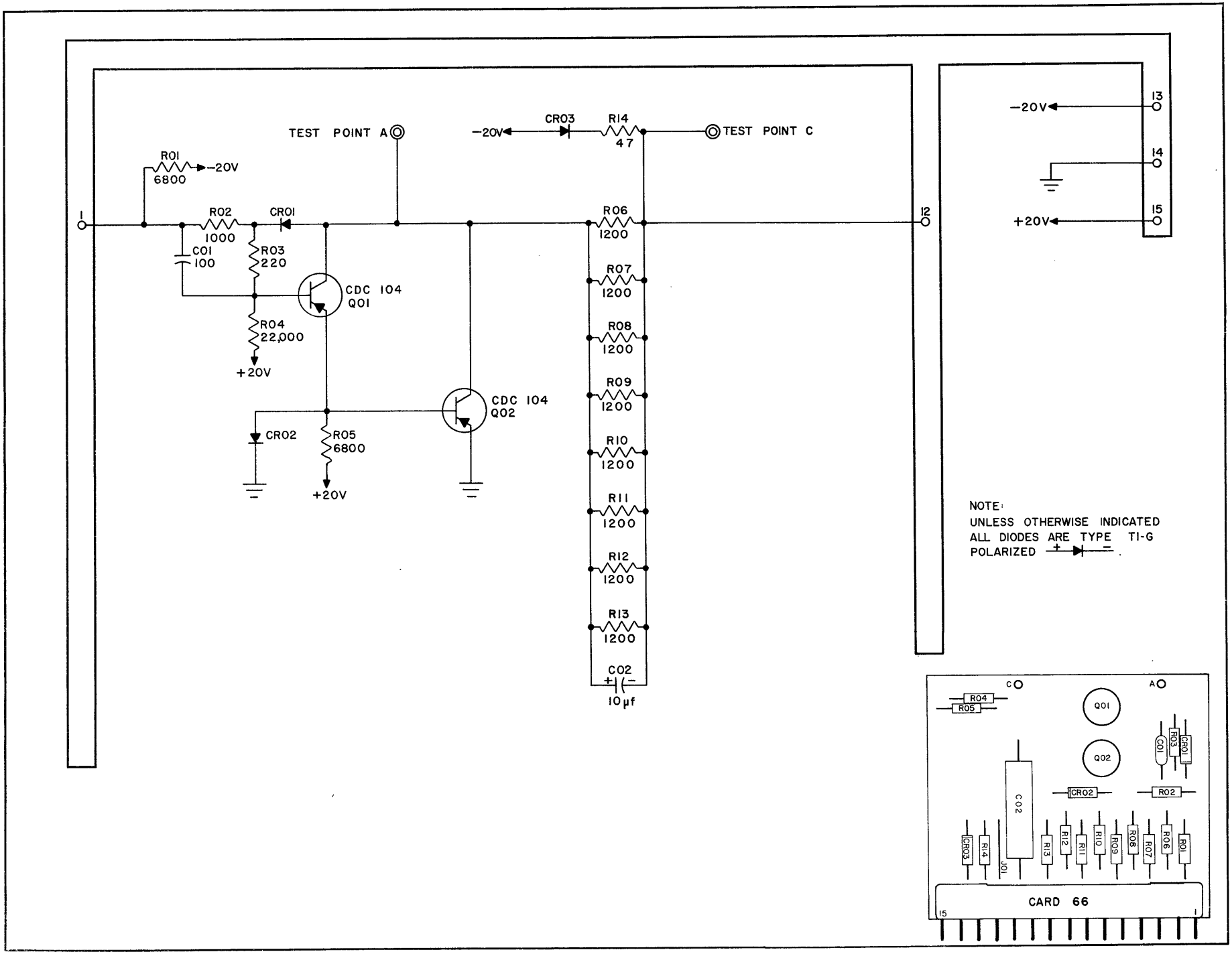


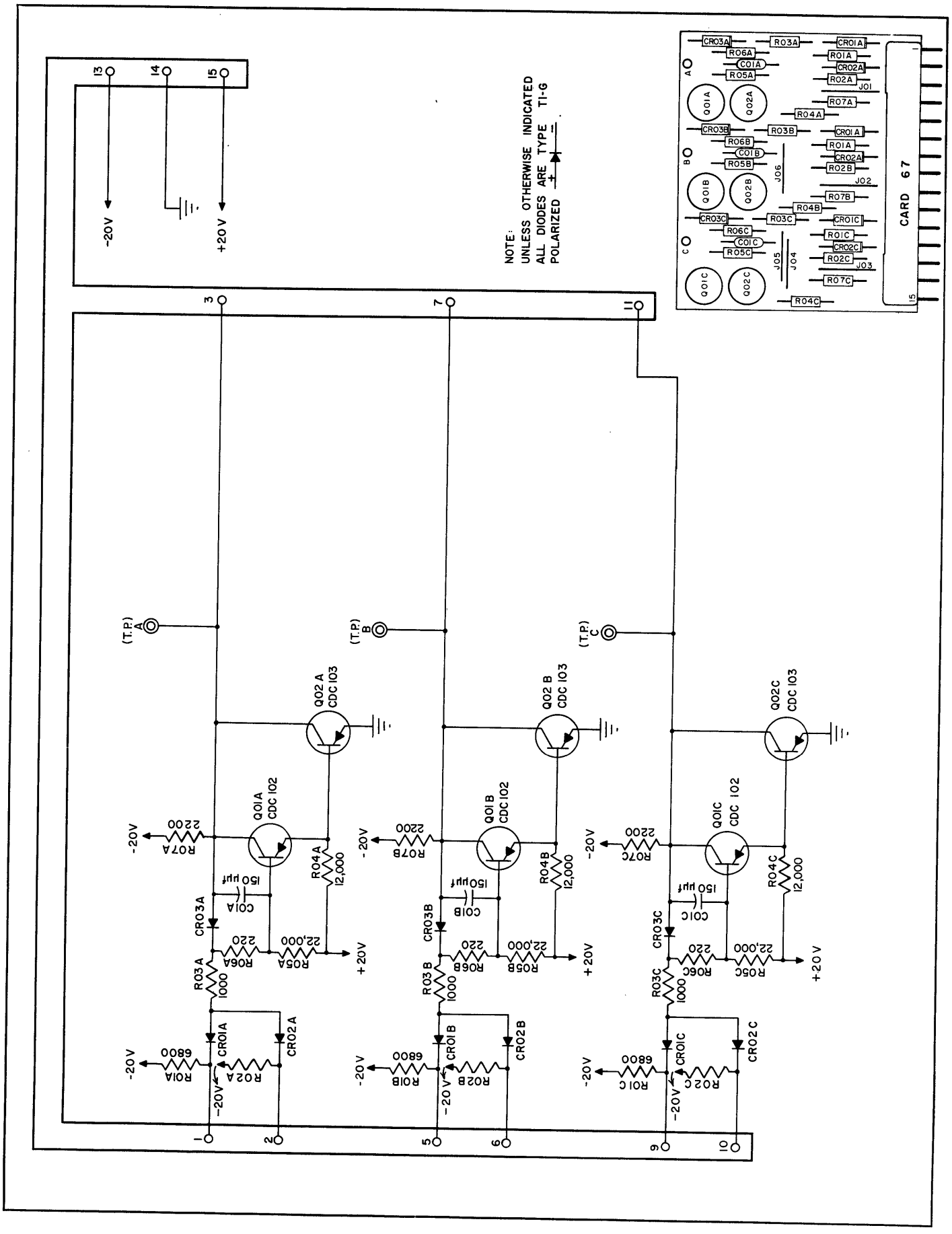
Input Card 61



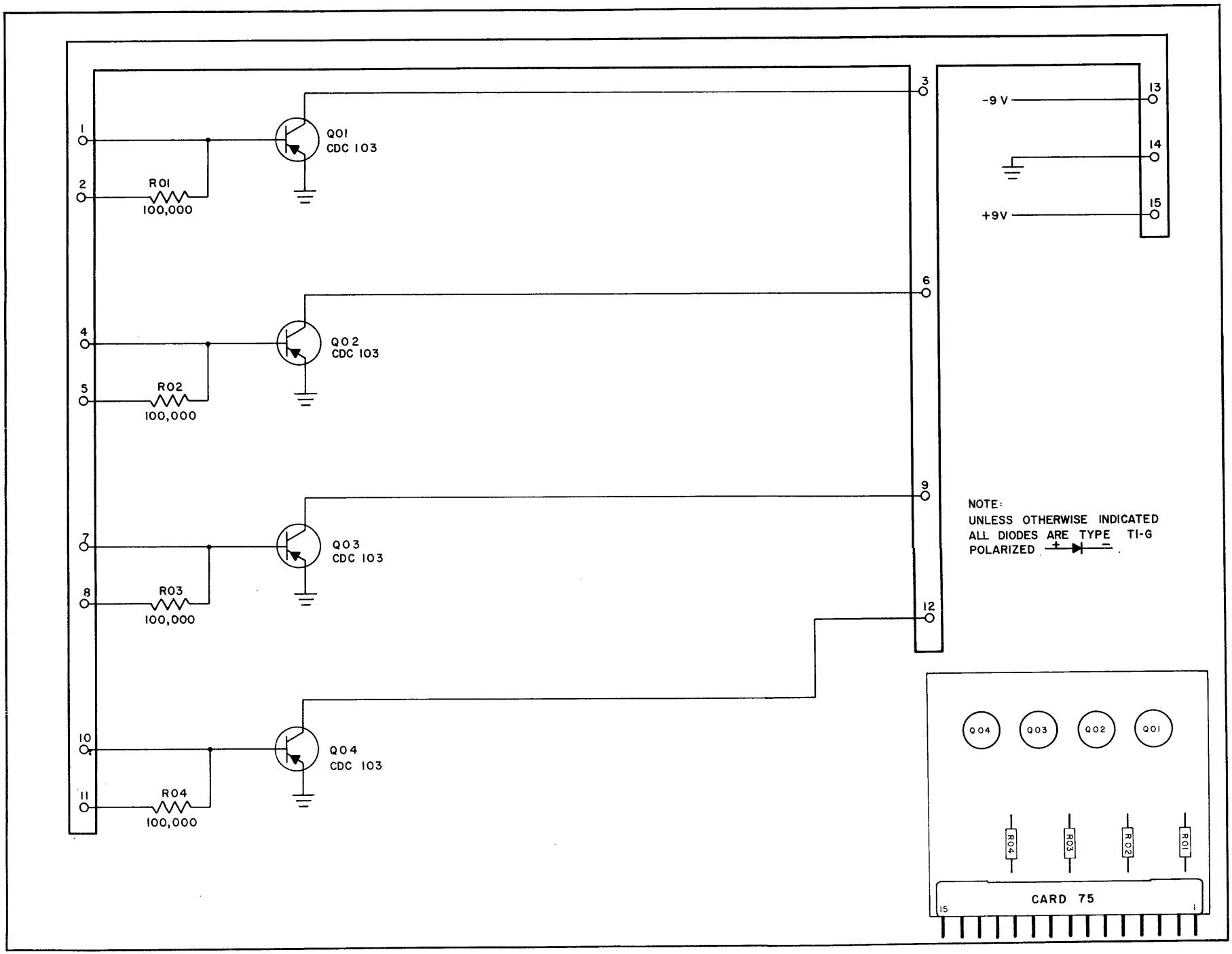
Output Card 62

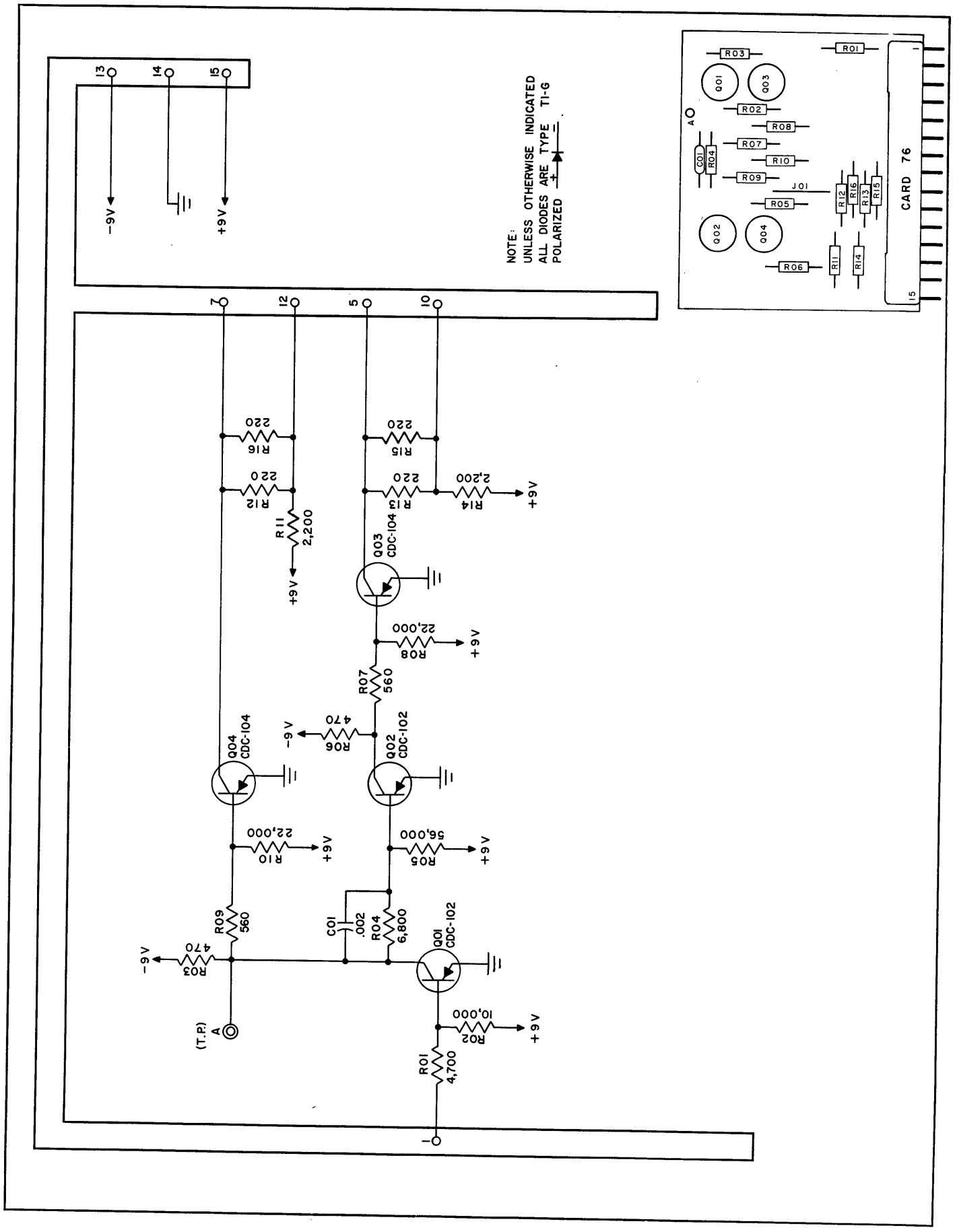

TEST POINT A
R01
6800
-20 V
R02
1000
CR01
C01
100
R03
220
CDC 102
Q01
R04
22,000
+20V
CDC 103
Q02
CDC 103
Q03
CR02
R05
6800
+20V
R06
3300
R07
3300
R08
3300
R09
3300
R10
3300
R11
3300
R12
3300
R13
3,300
1
12
13
14
15
-20V
+20V
NOTE:
UNLESS OTHERWISE INDICATED
ALL DIODES ARE TYPE TI-G
POLARIZED
A
Q01
Q02
Q03
R03
C01
CR01
R04
R05
CR02
R02
R13
R12
R11
R10
R09
R08
R07
R06
R01
CARD 65
15


TEST POINT A
TEST POINT C
CR03
R14
47
-20V
R01
6800
-20V
R02
1000
CR01
C01
100
R03
220
CDC 104
Q01
R04
22,000
+20V
CDC 104
Q02
CR02
R05
6800
+20V
R06
1200
R07
1200
R08
1200
R09
1200
R10
1200
R11
1200
R12
1200
R13
1200
C02
10 µf
1
12
13
14
15
-20V
+20V
NOTE:
UNLESS OTHERWISE INDICATED
ALL DIODES ARE TYPE TI-G
POLARIZED
CARD 66

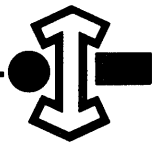


Output Card 67


1
2
R01
100,000
Q01
CDC 103
4
5
R02
100,000
Q02
CDC 103
7
8
R03
100,000
Q03
CDC 103
10
11
R04
100,000
Q04
CDC 103
3
6
9
12
-9 V
13
14
+9V
15
NOTE:
UNLESS OTHERWISE INDICATED
ALL DIODES ARE TYPE TI-G
POLARIZED
Q04
Q03
Q02
Q01
R04
R03
R02
R01
CARD 75
15
1



Reader Brake Clutch Driver Card 76



APPENDIX C

PREVENTIVE MAINTENANCE SCHEDULE

DAILY

Janitorial services: clean computer room, especially console top, tape baskets, and floors

Clean

PT Reader: remove tape setting clip to clean photo cell block
PT Punch: chad and paper lint
1607: -capstans, pinch rollers, and permanent leader
-heads (Ampex manual paragraph 2a)
-tape sensing slots and chambers (paragraph 2c)
-all surfaces over which tape moves

Lubricate

PT Punch: tape reel bearings if required (Teletype manual p.3-1)

Operating Checks

Run Test programs
PT Punch: registration of punches
1607: -Worn connectors on magnetic tape leaders
-Worn or noisy pinch roller or bearings

WEEKLY

Clean

Air filters in cabinets

Lubricate

PT Punch: -Toggle arm shaft, saturate felt washers
-Punch bail shaft, saturate felt washers
-grease tape reel bearings

WEEKLY (cont'd.)

Typewriter: -heavy gear grease on points where metal is moved on metal
-light oil on springs and pivot points

Operating checks

1607: -gaps for brakes (Ampex manual paragraph 3e)
-adjust vacuum for 20 1/2 inches (paragraph 3f)
-gaps on pinch rollers (paragraph 3i)
-adjust servo gain using MT test (paragraph 3m)

Voltage margins using test programs and varying MG output voltages

MONTHLY

Clean

Typewriter: keys, platen and actuator solenoids
PT Reader: clean all surfaces above console top

Lubricate

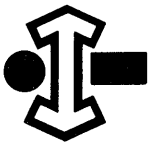
PT Reader
PT Punch: each end of motor, feed wheel ratchet and punch block
1607: check positive pressure blower (paragraph 2i)

Operating Checks

Typewriter: worn ribbon
PT Punch: punch for wearing
PT Reader: check festoon lamp
All Cabinets: check blowers

SEMIANNUALLY AND ANNUALLY

MG Control Cabinet and Relays: clean and check semiannually
Typewriter: clean and lubricate semiannually
MG Bearings: replace annually



APPENDIX D

CABLING INFORMATION

The identification of input-output cables and the information carried on their lines are treated in the following tables. Table D-1 lists the labels on the individual cables of the four groups. Each label indicates the function of the cable in the group by a prefix letter. The expression following the slash gives the computer connector for the cable. Table D-2 lists the information on each line of the six cables in a group.

Other cables in the computer system such as those connecting chassis within the main cabinet or those connecting the main cabinet and console are labelled as required.

TABLE D-1. CABLE IDENTIFICATION

Cable Group 1	A/7J2 CH 1-2	
	B/7K1 CH 1-2	
	C/7K2 CH 1-2	
	D/8J2 CH 1-2	
	E/8K1 CH 1-2	
	F/8K2 CH 1-2	
Cable Group 2	A/7L1 CH 3-4	
	B/7L2 CH 3-4	
	C/7M1 CH 3-4	
	D/8L1 CH 3-4	
	E/8L2 CH 3-4	
	F/8M1 CH 3-4	
Cable Group 3	A/7M2 CH 5-6	
	B/7M1 CH 5-6	
	C/7N2 CH 5-6	
	D/8M2 CH 5-6	
	E/8N1 CH 5-6	
	F/8N2 CH 5-6	
Cable Group 4	A/7O1 CH 7	
	B/7O2 CH 7	
	C/7P1 CH 7	
	D/8O1 CH 7	
	E/8O2 CH 7	
	F/8P1 CH 7	

Group 1
- channel 1 - buffer input
- channel 2 - buffer output

Group 2
- channel 3 - buffer input
- channel 4 - buffer output

Group 3
- channel 5 - buffer input
- channel 6 - buffer output

Group 4
- channel 7 - transfer input and output

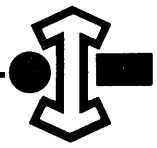
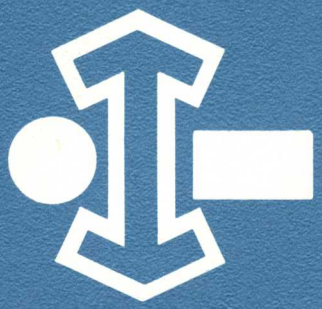


TABLE D-2. CONNECTOR PIN NUMBER ASSIGNMENTS

Pin No.	Input Buffer or Transfer Channel			Output Buffer or Transfer Channel		
	Cable A	Cable B	Cable C	Cable D	Cable E	Cable F
A	bit 47	bit 24	bit 01	bit 00	bit 23	bit 46
B	46	23	00	01	24	47
C	45	22	Input Ready	02	25	Output Ready
D	44	21	Input Resume	03	26	Output Resume
E	43	20	Input Buffer Active*	04	27	Interrupt Function
F	42	19	External Master Clear	05	28	Input Function Ready*
H	41	18	Not Used	06	29	Input Sense Ready*
J	40	17		07	30	Output Function Ready
K	39	16		08	31	Output Sense Ready
L	38	15		09	32	Sense Response
M	37	14		10	33	Output Buffer Active*
N	36	13		11	34	Function Bit 00
P	35	12		12	35	01
R	34	11		13	36	02
S	33	10		14	37	03
T	32	09		15	38	04
U	31	08		16	39	05
V	30	07		17	40	06
W	29	06		18	41	07
X	28	05		19	42	08
Y	27	04		20	43	09
Z	26	03		21	44	10
a	25	02		22	45	11
b	gnd	gnd	gnd	gnd	gnd	gnd

* Buffer cable only, unused in transfer



CONTROL DATA CORPORATION
501 Park Avenue, Minneapolis 15, Minnesota