SYSTEM 110
Hall Effect Measurement System
User's Manual

Document Number: S1001-9000-00B

Keithley Instruments, Inc.
Systems Division
30500 Bainbridge Road
Solon, OH 44139
(216) 248-4086

September 27, 1990

First Printing, February 1987
Second Printing, July 1988
Third Printing, September 1990

USERS MANUAL for the Keithley Hall Effect Test System 110

Chapter 1

INTRODUCTION

1.1 General Description

The Keithley Hall Effect System is a computer controlled test system for Hall Effect studies and Van der Pauw measurements in Gallium Arsenide and other wide-band gap, semi-insulating, semi conductor materials. It consists of a Keithley instrumentation package coupled with a programmable data acquisition computer and supplied with the software necessary to operate the system. Complete documentation is supplied so that the customer may create his own application-specific test programs.

1.2 Controller

This version of the Keithley Hall Effect Test System is designed to be controlled by an IBM-PC using a Capital Equipment Corporation (CEC) IEEE-488 Instrumentation Interface. The minimum recommended hardware configuration for the IBM-PC is 512Kb memory, 20 MB disk drive, IBM graphics text monitor, monitor interface card, and a printer port/printer with MS-DOS (version 3.0 or later).

1.3 Software

The software furnished with this system is as follows:

- Microsoft ® QuickBASIC 4.5
- Microsoft ® QuickC 2.0
- CEC card software 2.05
- Pro Screen 4.5e
- Geograf 4.0
- Hall Software 2.0

The installation procedure can be found in Appendix B. The software furnished with the system includes the driver object code package which enables the preparation of custom test programs with a minimum of effort. This driver package provides the link between the QuickBASIC operating program and the IEEE-488 Instrumentation Interface software package. All Hall Software is furnished without any protection (except copyright) so the customer may read and modify the programs to suit his specific application, if necessary.

The software distribution includes an operating prototype for Van der Pauw and Hall-Bridge measurements.

The **SYSTST** system test program is supplied to aid in the verification of the functioning and performance of the system.

The **PLOT** program is used to make the following graphs.

- Log R vs (1/T * 1000)
- Log Mob vs (1/T * 1000)
- Log Mob vs Magnetic Field
- Sweep I vs Meas V

This program writes the plot to the screen as well as to a file. The file can be printed out later, if desired.

1.4 High / Low System

The standard S110 High/Low System is able to measure a large range of resistivities. The high terminals are able to measure resistances as high as 100 G Ohms and the low terminals are able to measure resistances as low as 100 m Ohms. Both the high and the low terminals are capable of measuring resistances between 100 and 10 K Ohms. For a complete specifications chart on the measuring capabilities of both the high and low terminals, refer to Appendix F.

1.5 Instrumentation Package

The standard S110 High/Low System instrumentation includes a Keithley programmable electrometer, a four channel buffer amplifier and relay module, a programmable current source, a programmable voltmeter, and a programmable scanner. The standard IEEE-488 bus is used for instrument control.

1.5.1 Model 617 Electrometer

The Keithley Model 617 is a multi-function electrometer. It is used only as a current meter, monitoring the output of the system current source by providing a return path for the current source to system common. Its voltage source is also used to control the strength of the magnetic field if a magnet controller option is being used.

1.5.2 Model 196 Voltmeter

The Keithley Model 196 Voltmeter is used as the system voltmeter. This device is also a multi-function device. It is not only used to measure the voltage across the sample, it is also used to monitor the magnetic field (if a magnet controller is being used).

1.5.3 Quad Electrometer Buffer Amplifier

The electrometer buffer amplifier contains 4 high input impedance amplifiers. These act as unity gain buffers which are connected directly to the sample on the the high terminals. These amplifiers also provide the guard drive for the triaxial sample measuring cables. Provisions are made for zero offset adjustment on these amplifiers. The quad box also contains four low input ports. These ports are wired to low-thermal relays which in turn connect directly to the system voltmeter, bypassing the buffer circuitry used by the high inputs. Additional relays connect these ports to the 220 current source. There are also 2 terminals that are connected directly to the current source. These are "Current Source" and "Current Return". A block diagram of the S110 High/Low system, including a simplified schematic of the Buffer Amplifier/Relay Module is shown in Figure 1.1.

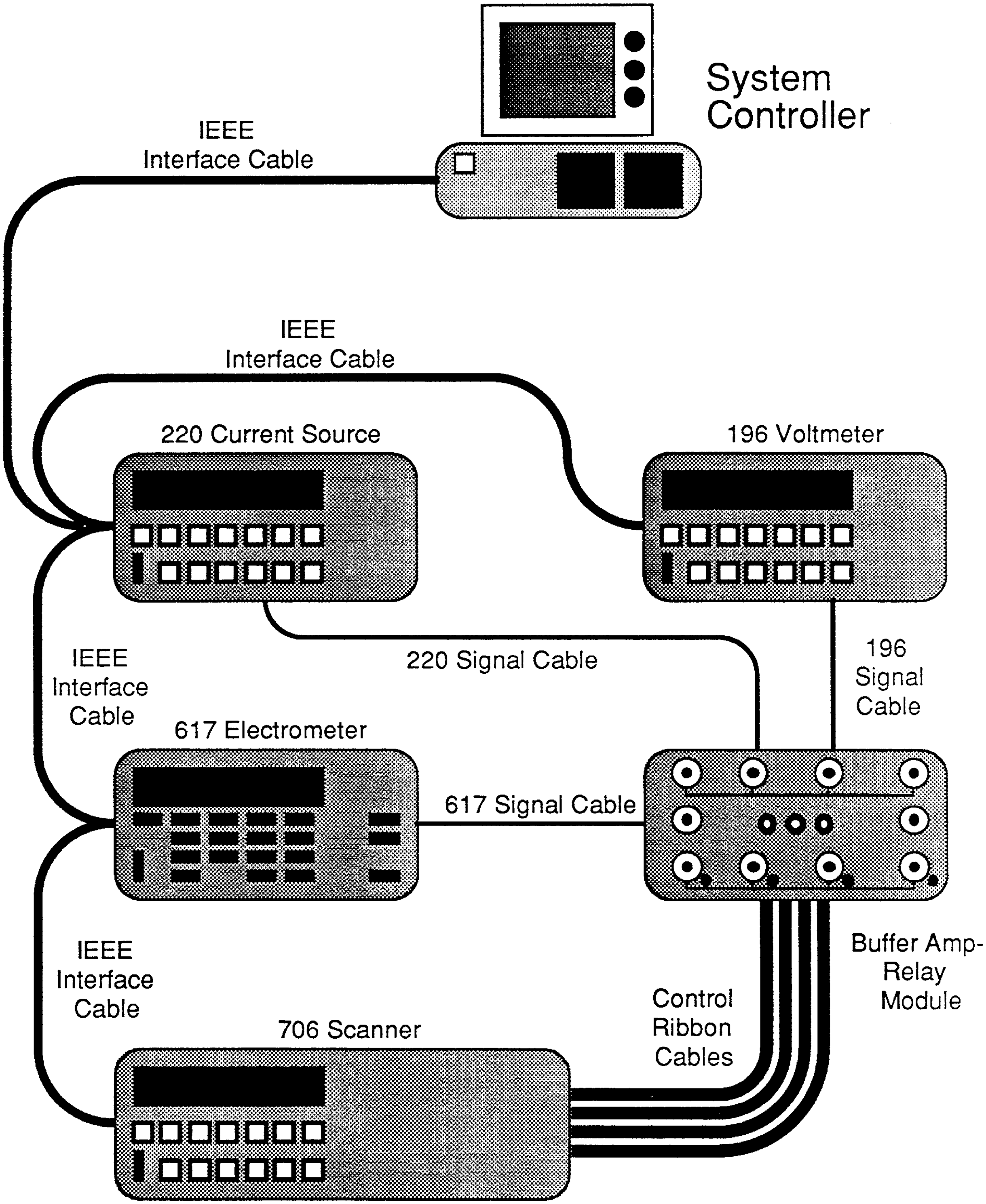


Fig 1.1 System 110 High/Low Instrument Package

1.5.4 Model 706 Scanner

The Keithley Model 706 Scanner is used to interpret the various IEEE commands used to control the relays and generate the drive signals to open and close the relays in the Buffer Amplifier/Relay Module. Ports 1 through 4 of the 706 are used for quad box number one. Port 10 of the 706 is used to control the magnet controller if one is available.

1.5.5 Model 220 Constant Current Source

The Keithley Model 220 current source features constant current output extending from 100mA full scale with 50µA resolution, to 1nA full scale with 500fA resolution. The voltage compliance limit can be set from 1 volt to 105 volts in 1 volt increments. In normal Hall System usage, the output is software restricted to 20mA maximum current and 10 volts maximum compliance voltage.

1.5.6 Power Controller

The system power controller provides the means for all instruments in the system to be controlled from a central unit. It is mounted behind the top front panel and consists of a circuit breaker, multiple contactors, 24-volt step-down transformer, and the ON and OFF push buttons used to actuate the contactor and switch system power. The 24-volt AC supply is used to control the contactor through the ON / OFF push buttons and to illuminate the ON pilot light. Main voltage is not present at the front panel. The power controller can service up to eight instruments with a maximum load not to exceed 15 Amperes. The unit comes equipped to service those instruments configured into the system.

This page is left blank intentionally

Chapter 2

SYSTEM FUNCTION

2.1 Introduction

The Keithley Hall Effect System is designed to switch the inputs and outputs of the current source, voltmeter, and electrometer to any of the high /low input ports.

NOTE: The 617 is **NOT** connected to the low terminal ports. Refer to the block diagram in Appendix SYSTST.

On the front panel of the Buffer Amplifier/Relay Module are triax connectors marked "Current Source" and "Current Return". These provide direct access to the output of the 220 and the analog common respectively. These connections are required for the Hall-Bridge measurements.

It is important to note that all connections and cables carrying signals associated with the test device are triaxial to provide guarding throughout the system.

2.2 Device Switching

Switching for all devices in the S110 system is accomplished with the Buffer Amplifier/Relay Module. Low leakage relays are used in the control of the 220 Current Source and Current Return nodes, and low-thermal-offset relays are used for switching the high/low inputs to the 617 Electrometer. The low-thermal-offset relays are used for switching the high and low inputs to the 617 Electrometer. The low leakage relays are used in the current section to minimize current losses, particularly when forcing low currents into the test sample. The low-thermal-offset relays are necessary to minimize the effects of thermal EMF contact potentials in the relays, which could have an adverse effect on low voltage measurements. The 196 is also connected through the relays.

The 706 Scanner supplies control and power for all the relays in the Buffer Amplifier/Relay Module. Ribbon cables are used to run from the scanner to the amplifier module. The 10-foot cable harness supplied with the buffer amplifier module allows removal of the module from the instrument rack to a remote location in close proximity to the sample if desired.

While the system is ordinarily meant to be operated automatically from the system controller, manual operation from the instruments' front panels is also possible. Refer to the individual instrument manuals for detailed information on the instrument. The block diagram in Appendix SYSTST shows the channel numbers associated with the relays in the Buffer Amplifier/Relay Module.

2.3 Guarding

Guarding signals on the high resistance terminals is extremely useful. Its purpose is to minimize leakage from the signal conductor to low or ground. It also reduces analog settling time which can be a problem in high-impedance circuits.

In the Keithley Hall Effect Test System, guarding of the high resistance signal lines to the device under test is accomplished by connecting the unity-gain buffer amplifier outputs to the inner shield of the triax cables associated with their respective inputs. Since the buffers are always operating as unity-gain, non-inverting amplifiers, their low impedance output is equal to the signal at their input. Thus, the potential difference between the signal conductor and the inner shield is essentially zero. This guard potential is carried to the Buffer Amplifier/Relay Module connector. In actual test operation, the triax cable should be terminated as close to the test point as practicable, to insure maximum benefit from the available guarding.

The low resistance terminals are guarded through the analog ground.

Chapter 3

SYSTEM OPERATION

3.1 Set-Up

Shortly after delivery of the Hall System, a Keithley Systems Field Service Engineer will call to unpack and install the system. The system will be set-up in the desired location and its operation thoroughly tested. During the visit, adequate time will be allowed to train customer personnel in the use of the system's hardware and software, assuming that the customer personnel are familiar with the operation of the controller and its programming language.

If, for any reason, installation by Keithley Field Service is not practical, detailed installation instructions are supplied in Appendix B.

3.2 System Start-Up

3.2.1 Power-On

The power for the system is switched from the power distribution panel on the front of the system rack cabinet. Ordinarily, the 617 Electrometer, 220 Current Source, 196 Voltmeter, and the 706 Scanner should be left with their front panel switches in the ON position. This ensures that power-on for these instruments is accomplished by the system's power switch. When powered up, these instruments each go through their own built-in test routine and come back to a "ready" state upon successful completion of their tests. The buffer module is powered from the 706 Scanner and will be active whenever the 706 is on and the buffer module interface cards are installed.

Generally, the instrument rack should remain powered up continuously, unless it is not to be used for an extended period. By leaving the instruments ON, the system will always be temperature stabilized and there is no need for a warm-up period. There is no problem with allowing the instruments to operate continuously due to the low power dissipation in the system rack.

There are no provisions made for powering the System Controller from the instrumentation package. The design of the Buffer Amplifier/Relay Module makes the possibility of a ground loop from a mains ground mismatch, extremely unlikely. It is recommended, however, that both units be powered from the same outlet to avoid any potential problems.

3.2.2 Stabilization and Offset Adjustment

After powering the system up and allowing time for the instruments to stabilize (about one hour), the zero offset of the buffer amplifiers should be checked. This is done by running the **SYSTST** program and selecting the **ZERO OPAMP TEST**. See chapter 6 for further information on how to run this test. This test does not need to be run that often.

3.2.3 System Evaluation Test

For the first several weeks, the system's **SYSTST** should be run whenever the system is started up. Thereafter, **SYSTST** should be run at least once a month. Retain a printout log reference or disc file for reference and comparison to prior runs. Refer to chapter 6 for further details.

3.3 Connecting to the Test Device

All connections from the sample or test fixture to the Buffer Amplifier/Relay Module's connectors should be made with triax cable and connectors. If the test fixture is furnished with triax connectors, then the triax cables furnished with the system can be utilized.

Wherever possible, the guarding potential on the inner shield of the triax cable from the system's Buffer Amplifier/Relay Module should be carried as close to the sample as practicable. This guarding minimizes leakage and improves measurement speed. (See section 2.3 of this manual for more information.)

Connections will vary according to the application measurement being made. For Van der Pauw and Hall-Bridge measurements, the connections are covered in Section 5 of this manual.

3.4 Operating the System

Although designed primarily for operation from the System Controller, the system's instrument rack may be used directly from the manual front panel controls of the instruments. In some cases, especially where a new set up is to be tried, this may be advantageous since it can verify quickly a new test or procedural step without the necessity of writing and debugging software. Once the procedure is established and verified under manual control, a software program will be much easier to write and debug. In order to use this manual control capability, the user should have a thorough understanding of each of the instruments' operation and capabilities. The separate instruction manuals supplied for the instruments should be studied carefully before manual control operation is attempted.

The system's normal mode of operation is from the System Controller. Chapter 4 of this manual details the system's software distribution and programming.

Chapter 4

VAN DER PAUW AND HALL-BRIDGE PROGRAMS

4.1 Introduction

There are two programs in the system's software distribution for making Hall measurements. The first is a program named VDPHAL which is used for a normal Van der Pauw configured sample, illustrated in Figure 4.1. The second, named HALBRDG, is written for a six contact sample as illustrated in figure 4.6.

These programs are intended to serve as prototype programs which the user may modify as desired to fit individual samples and particular environments. However, you should first become familiar with the operation of the programs as supplied, before attempting any modifications. These programs are written in QuickBASIC.

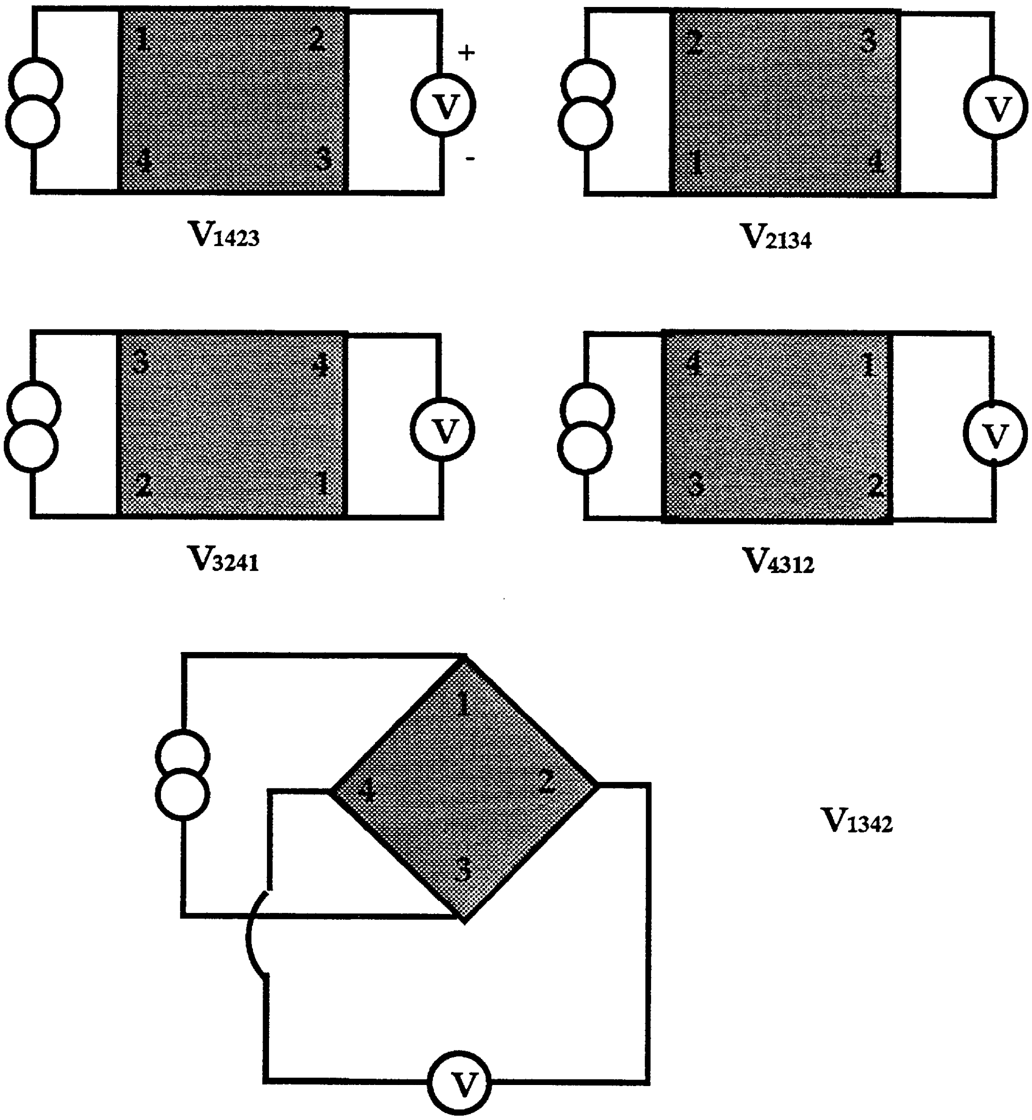
4.2 Van der Pauw - Hall Technique

4.2.1 Measurement Theory

This program makes a sample resistivity measurement by using the classic Van der Pauw technique. The sample connections are illustrated in figure 4.1.

The contacts on the periphery of the sample are connected to the current source and the resulting voltage is measured by the voltmeter. An illustration of the type of measurements being made can be seen in Figure 4.1.

The resistivity of the sample is then calculated as shown in the equations in Figure 4.2 .



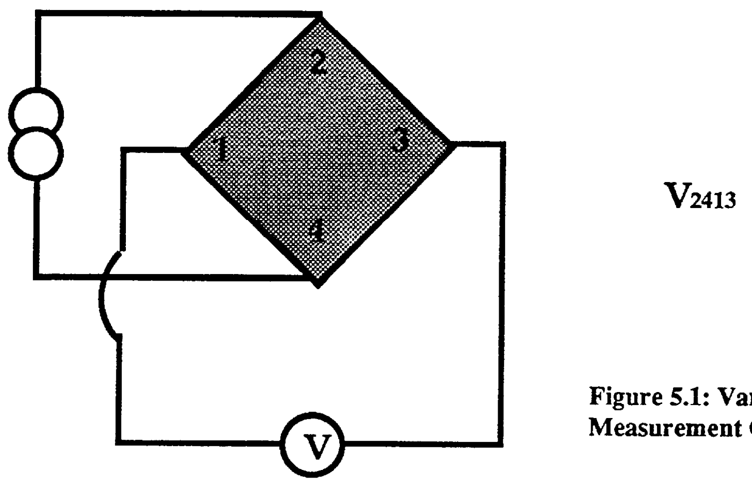


Figure 5.1: Van der Pauw - Hall Measurement Circuits

$$\rho_a = \frac{\pi t}{4I\ln 2} f_a\,(Q_a)\,[V21{,}34 - V12{,}34 + V32{,}41 - V23{,}41]$$

$$\rho_b = \frac{\pi t}{4I\ln 2} f_b\,(Q_b)\,[V21{,}34 - V12{,}34 + V32{,}41 - V23{,}41]$$

$$Q_a = \frac{V21{,}34 - V12{,}34}{V32{,}14 - V23{,}14}$$

$$Q_b = \frac{V43{,}12 - V34{,}12}{V14{,}23 - V41{,}23}$$

$$f_a = 1 - Q_a^2\left[\frac{\ln 2}{2}\right] - Q_a^4\left[\frac{(\ln 2)^2}{4} - \frac{(\ln 2)^3}{12}\right]$$

$$f_b = 1 - Q_b^2\left[\frac{\ln 2}{2}\right] - Q_b^4\left[\frac{(\ln 2)}{4} - \frac{(\ln 2)}{12}\right]$$

$$\rho = \frac{\rho_a + \rho_b}{2}$$

VAB,CD refers to the voltage measured between terminals "C" and "D" when a current is forced from terminals "A" to "B".

Figure 4.2 Van der Pauw Resistivity (ρ) Equations

$$RHA = \frac{2.5 \times 10^{7} t}{BI} [V31,42(+B) - V13,42(+B) + V13,42(-B) - V31,42(-B)]$$

$$RHB = \frac{2.5 \times 10^{7} t}{BI} [V42,13(+B) - V24,13(+B) + V24,13(-B) - V42,13(-B)]$$

$$RH = \frac{RHA + RHB}{2}$$

$$\mu = \frac{RH}{\rho}$$

$$N = \frac{1}{RH \times 1.6 \times 10^{-19}}$$

VAB,CD refers to the voltage measured between terminals "C" and "D" when a current is forced from terminals "A" and "B".

Figure 4.3 Hall Coefficient, Mobility, and Carrier Concentration Equations for a Van der Pauw Sample.

Measurement of mobility and carrier concentration are also made in this program. The system automatically configures the measurement as shown in Figure 4.1. This measurement requires the presence of a magnetic field applied to the Z axis (perpendicular to the sample surface). The results of this measurements are values for mobility and carrier concentration as calculated in the equations shown.

4.2.2 VDPHAL Program Execution

VDPHAL.BAS is the main program. This program can be run from the **S110 SOURCES** directory. A menu will prompt the user to enter:

- Name
- Sample ID
- Sample Thickness (cm)
- Temp (K)
- High or low impedance terminals
- Magnetic field desired (KG)
- Delay period desired between the **FORCEI** and the **MEASV** or **MEASI**
- Quad box number

NOTE: use either the enter key or the tab key to enter the values in the menu. **DO NOT** use the arrow keys.

Once all the entries are completed, testing will begin. The program first checks to see if a cryo is available. If there is and you did not set the temp value to "0", the system will set the temperature.The system then adjusts for either high or low resistance terminals. The program then runs either **VDP** or **VDPLOW** on all the samples being tested. Please consult the manual for more details on **VDP** or **VDPLOW.**

VDP and **VDPLOW** check the current source when making a measurement to see if the current source is in voltage limit. If it is, the current is divided by 10 and then another measurement is taken. This routine can be additionally modified by replacing the **MEASV** and **MEASI** routines with the **SSMEASV** and **SSMEASI** routines. The routine would then return steady state measurements. See the documentation on the **SSMEASV** and **SSMEASI** routines for more details.

After the **VDP** or **VDPLOW** measurements are made, calculations are done with the returned measurements to calculate the resistivity. At this point in the program, all the data acquired from the menus and the measurements for each individual sample are saved in a file DATA.RD1.

At this point, if the user had set the magnetic Field to "0" in the first menu, the results of the tests would be displayed. If the user set the magnetic field to any other value, the program would set the field. This is accomplished by using the **FORCEMAG** routine. Once the field has been set, the program would execute either the **HALL** or **HALLOW** routine. This routine would make the measurements needed to calculate the Hall coefficient, sample type, mobility, and the carrier concentration. Once the measurements are made, all the data gathered is appended to the file DATA.RD1, the same file mentioned above.

When all the measurements have been made in a positive field, the magnet then reverses the field direction. It then runs either the **HALL** or **HALLOW** routine on all the samples. The program opens up the file DATA.RD1 which contains all the measurements made in the positive field. Calculations are made for the Hall coefficient, mobility, carrier concentration, and the sample type. All of this data, along with the negative magnetic field measurements are placed in the file DATA.RD2.

At this point the instruments are no longer needed, so the program calls **DEVINT**. The data recovered from the sample tested is displayed. There are two screens of data for each sample, if no magnet is used. There will be three screens of data for each sample if a magnet is being used. The first screen displays the resistivity , Hall coefficient, sample type, mobility, and carrier concentration. If the F3 function key is pressed, the second menu will appear. If you press the F7 function key, you go to the final menu. The second menu displays all the measurements made without the magnetic field. Again by striking F3, the third menu will appear. If you strike F7 as before the final menu will appear. This third menu will only appear if the magnet was used. This menu displays all the measurements made with the positive and negative magnetic field. By again pressing F3 the final menu will appear. This final menu gives the choices of:

F3 = test again
F5 = show results again
F7 = quit

F3 will return you to the main menu at the beginning of the program.

The data file that is put together at the end of the testing, is a listing of all the results that were gathered from the testing. At the top of the page is a header that lists the operator's name, sample id, sample size, temperature, and magnetic field setting. Below the header is the resistivity, Hall coefficient, sample type, mobility, and the carrier concentration of the sample. The second page of the data file, lists all the currents and voltages that were measured with and without the magnetic field.

MODIFICATIONS THAT CAN BE ADDED

- The program does not print out the "SAMPLE ID".DAT file. A location is indicated near the end of **VDPHAL.BAS** where the user can alter the program to send the data to the printer. If you modify **VDPHAL.BAS** you will have to rebuild it by using the **MAKEIT.BAT** command, as explained in chapter 7.

- **SSMEASV** and/or **SSMEASI** can be added to **VDP**, **VDPLOW**, **HALL** or **HALLOW** by replacing the **MEASV** and/or **MEASI** routines with the **SSMEASV**, **SSMEASI**. This could add extra time to the tests being run due to the way the **SSMEASV** and the **SSMEASI** routines function. Please refer to the description of these routines for more information.

VDPHAL.EXE is the executable file created from **VDPHAL.BAS**. This file is contained in the S110 Sources directory. You get to this directory by typing:

```
C: cd\ <CR>
C: cd s110 <CR>
C: cd sources <CR>
```

To run this program, type:

```
C: vdphal
```

4.3 Hall-Bridge Technique

4.3.1 Measurement Theory

The Hall-Bridge measurement Technique utilizes a different method then the Van der Pauw measurement to arrive at the values of resistivity, mobility, and carrier concentration.

The sample configuration and connections for a Hall-Bridge measurement are shown in Figure 4.4. The equations for the resistivity, Hall Coefficient, mobility, and carrier concentration are shown in Figure 4.5 and Figure 4.6. This measurement requires separate connection to the 220 and analog common connectors on the front panel to provide the current flow along the "X" axis of the sample.

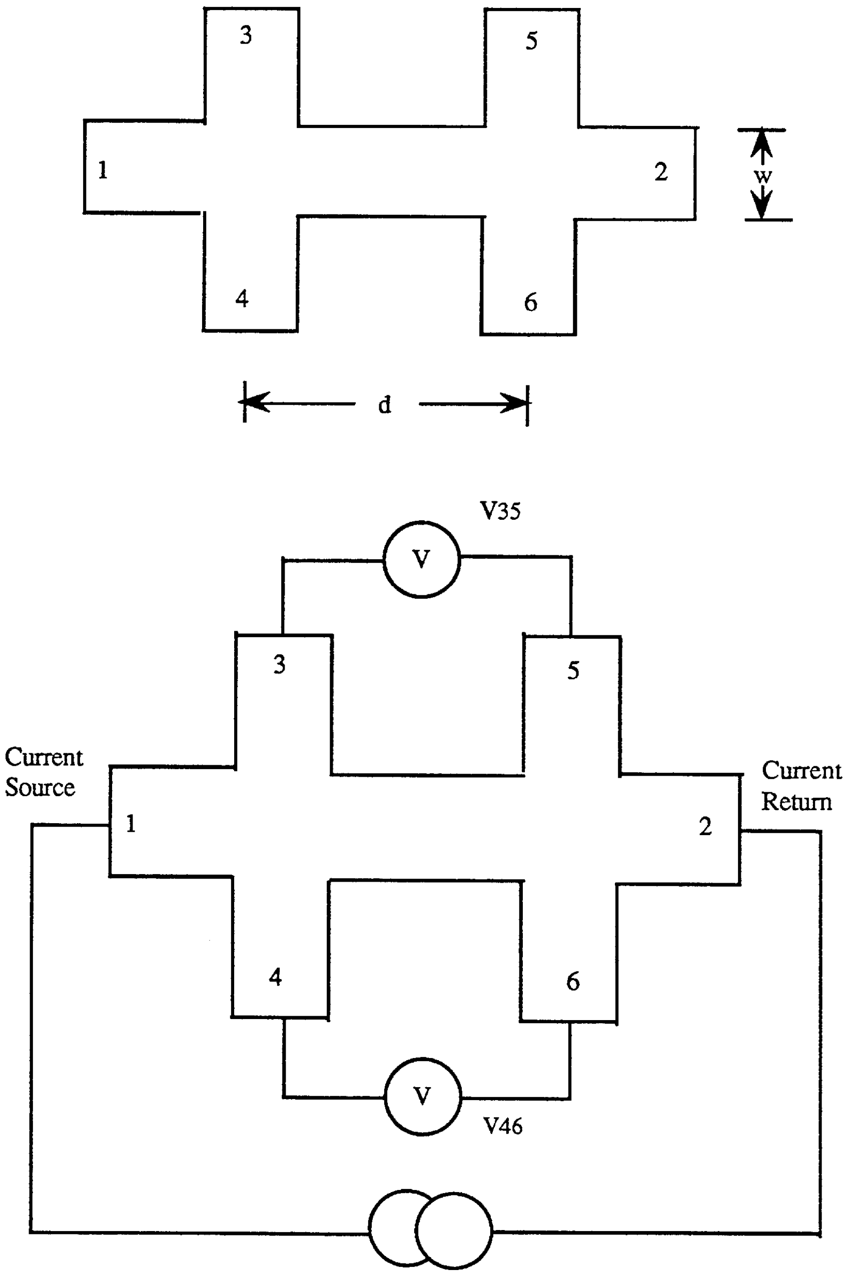


Figure 4.4: Hall-Bridge Configuration

$$\rho a = \frac{V12{,}46 - V21{,}46}{2I} \times \frac{wt}{d}$$

$$\rho b = \frac{V12{,}35 - V21{,}35}{2I} \times \frac{wt}{d}$$

$$\rho = \frac{\rho a + \rho b}{2}$$

VAB,CD refers to the voltage measured between terminals "C" and "D" when a current is forced from terminals "A" and "B".

Figure 4.5: Hall-Bridge resistivity (r) Equations

$$RHA = \frac{2.5 \times 10^{7} t}{BI} [V12{,}65(+B) - V21{,}65(+B) + V21{,}65(-B) - V12{,}65(-B)]$$

$$RHB = \frac{2.5 \times 10^{7} t}{BI} [V12{,}43(+B) - V21{,}43(+B) + V21{,}43(-B) - V12{,}43(-B)]$$

$$RH = \frac{RHA + RHB}{2}$$

$$\mu = \frac{RH}{\rho}$$

$$N = \frac{1}{RH \times 1.6 \times 10^{-19}}$$

VAB,CD refers to the voltage measured between terminals "C" and "D" when a current is forced from terminals "A" and "B".

Figure 4.6: Hall Coefficient, Mobility, and Carrier Concentration Equations for a Hall-Bridge Sample

Care must be taken in sample preparation to ensure that the dimensions "d" and "w" are well defined, since they enter directly into the calculations as the dimensions of the "X" and "Y" axes, respectively. The overall physical length of the sample is not used. Therefore, when responding to the program's request for the "d" dimension, it is the distance between the contacts that is entered (see Figure 4.4).

4.3.2 HALBRDG Program Execution

HALBRDG.BAS is the main program. A menu will prompt the user to enter:

- Name
- Sample ID
- Sample thickness (cm)
- Sample width (cm)
- Sample spacing (cm)
- Magnetic Field (KG)
- Current
- Delay (ms)
- Quad box number

After all data has been entered, the program will connect to either the high or low resistance terminals. The program will then set the temperature if a controller is present and the value is not "0". Next it runs **HLB.** Refer to the section on **HLB** for more details.

HLB is made to check the current source to see if it is in voltage limit. If it is, then the current will be divided by ten until the current source is out of voltage limit.

After the **HLB** routine is done making the required measurements, the resistivity is calculated and all the data is saved in a file "DATA.RD1".

At this point if the user has set the magnetic field to "0", the results will be displayed. If the user has set the magnetic field to something besides "0", the **FORCEMAG** routine will now set the magnetic field. Refer to the **FORCEMAG** routine for details. Once the magnetic field has been set, the **HLBHALL** routine will be executed. The **HALBRDG** routine will return the positive magnetic field measurements needed to calculate the Hall coefficient, mobility, sample type, and the carrier concentration. Once these values have been measured, all the data is appended to "DATA.RD1".

The magnetic field is now reversed by using the **FORCEMAG** routine. **HLBHALL** will again be executed to get the negative magnetic field measurements needed to calculate the Hall coefficient, mobility, sample type, and the carrier concentration. Once these values are measured, the Hall coefficient, mobility, sample type, and the carrier concentration are calculated. All these values are now saved in "DATA.RD2".

At this point the instruments are no longer used. The program calls the **DEVINT** routine. The data for the sample is now displayed. There are two results screens if no magnet is being used. There are three results screens if a magnet is used. The first screen shows the resistivity, mobility, Hall coefficient, sample type, and the carrier concentration. If F3 is pressed, the program will go to the second menu. If F7 is pressed, the final screen will be shown. The second menu shows the measurements used to calculate the resistivity (no magnetic field). Again if F3 is pressed the third menu will appear (if a magnet is being used), otherwise the final menu will appear. If F7 is pressed the final menu will appear. The third menu consists of all the measurements made in both the positive and negative field. These are the measurements used to calculate the Hall coefficient, mobility, sample type, and the carrier concentration. If F3 is pressed at this time, the final menu will appear. When this final menu appears, all the data results have been stored in "Sample ID.DAT". This final menu gives you the choices of:

F3 = test again
F5 = show results again
F7 = quit

F3 will return you to the main menu at the beginning of the program.

MODIFICATIONS THAT CAN BE ADDED

- The program does not print out the "SAMPLE ID".DAT file. A location is indicated near the end of **HALBRDG.BAS** where the user can alter the program to send the data to the printer. If you modify **HALBRDG.BAS** you will have to rebuild it by using the **MAKEIT.BAT** command, as explained in Chapter 7.

- **SSMEASV** and/or **SSMEASI** can be added to **HLB** or **HLBHALL** by replacing the **MEASV** and/or **MEASI** routines with **SSMEASV** or **SSMEASI**. This could add extra time to the tests being run due to the way the **SSMEASV** and the **SSMEASI** routines function. Please refer to the description of these routines for more information.

HALBRDG.EXE is the executable file created from **HALBRDG.BAS**. This file is contained in the S110 Sources directory. You get to this directory by typing:

```
C: cd\ <CR>
C: cd s110 <CR>
C: cd sources <CR>
```

To run this program, type:

```
C: halbrdg
```

This page is left blank intentionally

Chapter 5

SOFTWARE DISTRIBUTION

5.1 Introduction

This chapter gives an indepth description of each of the routines that are part of the Hall system software. The routine names themselves are in **BOLD CAPITAL** letters, the variable names in the parameter lists are in CAPITAL letters.

5.2 Device Driver Interfaces

The S110 software consists of two levels, a high level and a low level. The high level is the level implemented in Quick Basic. This level includes all of the test programs and test diagnostics. The low level is made up of the drivers to the instruments. High level programming is all that the user needs in order to make custom software. The following sections show how to use each of the commands of the S110 test system.

5.2.1 CONNECT

The **CONNECT** routine is used to connect the devices to the appropriate pins through the matrix. This routine can be used at anytime after both the **TSTSEL** and the **HLSEL** routines have been called. This routine first checks to see if the **FORCEI** routine has been used prior to this **CONNECT**. If so, the routine turns off the current source. The routine will then re-initialize all the instruments and open all the relays. If the **FORCEI** routine has not been called, the instruments are not re-initialized and the relays are left alone. The **CONNECT** routine works by sending both the device name and pin numbers in the indicated parameter list to the system. Legal device names can be found in Appendix A. Legal pin numbers are 1, 2, 3 or 4. The routine will send back an error if any illegal connection is attempted. These illegal connections are:

- The high terminal of the current source cannot be connected to the high terminal of the current meter.

- No two devices can be connected to the same pin number at the same time.

The calling sequence for this routine is:

CALL CONNECT(DEVICE, PIN)

Where:

Name	Type	Operation	Description
Device	Integer	Input	Name of device to be connected.
Pin	Integer	Input	Number of pin to be connect to (1-4).

Programming Considerations

- PIN must be 1, 2, 3 or 4.

- Watch for illegal connections

- The relays will not clear until the devices are re-initialized or a **FORCEI** is called prior to the **CONNECT** routine.

- Make sure **TSTSEL** is the first routine called.

ERROR Displays

- If there is an illegal connection attempted.

- If the strings sent to the instruments are not received properly.

5.2.2 DELAY

The **DELAY** routine is used to add a delay into a program. The delay is used mainly between the **FORCEI** and **MEASI** or **MEASV**. **DELAY** is also used when setting the magnet or reversing the magnet. The TIME is in milliseconds. It should be noted that the delay is run from the internal clock of the computer. Therefore the delay is only as accurate as the internal clock. This should be taken into account when using the **DELAY** routine. **DELAY** should be used to allow the instruments to settle once a current has been forced. This routine can be called at anytime.

NOTE: This routine has been built into the magnet controller routines.

The calling sequence for this routine is:

CALL DELAY (TIME)

Where:

Name	Type	Operation	Description
Time	Single	Input	Delay time desired (ms).

Programming Considerations

- Use the **DELAY** routine between the **FORCEI** routine and any **MEASI** or **MEASV** routine. This will allow the instruments to settle before a reading is taken.

ERROR Displays

- No errors can be returned from this routine.

5.2.3 DEVINT

The **DEVINT** routine should be used as the last routine called in a program. This routine will re-initialize all the instruments to their initial states. Refer to Appendix A for a listing of the instrument's initial states.

The calling sequence for this routine is:

Call **Devint**

Programming considerations

- Should be the last routine called in a program.

ERROR Displays

- If the strings are received improperly.

5.2.4 FORCEI

The **FORCEI** routine is used to force a current through the sample. This routine is used by passing both the device and the amount of current desired in the parameter list. The range for the current is +101 mA to -101 mA. This routine can be called at anytime after the **TSTSEL** routine has been called. The current source will automatically turn off when the **CONNECT** routine is called, so another **FORCEI** must be called in order to turn the current source back on. Appendix A lists all the legal devices that can be used in this routine.

The calling sequence for this routine is:

CALL FORCEI (DEVICE, AMPS)

Where:

Name	Type	Operation	Description
Device	Integer	Input	Name of device that will force the current.
Amps	Single	Input	Amount of current to be forced (amps).

Programming Considerations

- The current source will automatically be turned off when the devices are re-initialized or the **CONNECT** routine is called.

- Make sure **TSTSEL** is the first routine called.

ERROR Displays

- If the current is out of range.

- If an illegal device is used.

- If the strings sent to the instruments are improperly received.

5.2.5 FORCEV

The **FORCEV** routine is used to force a voltage. The **FORCEV** routine is only used for controlling the magnet. There is no need to use the **FORCEV** routine if there is no magnet to control. This routine can be called at anytime after the **TSTSEL** routine is called. Unlike the **FORCEI** routine, the voltage source will not be turned off when the **CONNECT** routine is called. The voltage source will only turn off when the **DEVINT** routine is called. The voltage source can also be turned off by forcing zero volts. The legal devices that can be used in the **FORCEV** routine can be found in Appendix A.

The calling sequence for this routine is:

CALL FORCEV (DEVICE, VOLTS)

Where:

Name	Type	Operation	Description
Device	Integer	Input	The name of the device used to force the voltage.
Volts	Single	Input	Voltage to be forced (volts).

Programming Considerations

- There is no need to call **FORCEV** if there is no magnet.
- The voltage source will not turn off until the **DEVINT** routine is called.
- Make sure **TSTSEL** is the first routine called.

ERROR Displays

- If an illegal device is used.
- If the voltage is out of range.
- If the strings sent to the instruments are improperly received.

5.2.6 HLSEL

The **HLSEL** routine is used to define which test terminals are to be used for testing high or low resistance. The **HLSEL** routine should be the next routine following the **TSTSEL** routine. The reason for this is that an error could occur while making connections if the **HLSEL** routine is not called before the **CONNECT** routine. This routine only has to be called once at the beginning of the program. The only other time it would be called is if the user changes from high resistance terminals to low resistance terminals or vice versa.

The proper calling sequence for this routine is:

CALL HLSEL (TERMINAL)

Where:

Name	Type	Operation	Description
Terminal	Integer	Input	Must be HIGH or LOW.

Programming Considerations

- Should be the routine following **TSTSEL**, to insure that the connections are made properly.

ERROR Displays

- Returns an error if the strings sent to close the relays for the high or low test system are received improperly.

Example

```
CALL tstsel(1)
CALL hlsel(HIGH)
```

The example above tests sample 1 on the high resistance terminals.

5.2.7 IEEESEND

The **IEEESEND** routine is used to send an IEEE command string string to any of the instruments. This routine can be used to open and close relays or to set the instruments into different modes that are not offered by the System 110 software. Refer to the appendix S110 for a list of DEVICE names.

The calling sequenced for this routine is:

Call **IEEESEND**(DEVICE, STRING)

Where:

Name	Type	Operation	Description
Device	Integer	Input	The device that the command should be sent to.
String	Integer	Input	The address of the command string . See example.

Programming Considerations

- You **CANNOT** send the command string as is. You must first add the null character to the end of the string, then you must take the **SADD** of the string (**SADD** is a QuickBASIC command that returns the address of the specified string expression). Refer to the example for details.

EXAMPLE

This example will display channel 7 on the 706 instrument and then close channel 7.

```
CALL TSTSEL(1)
        .
        .
        .
        .
A$ = "B7C7X" + CHR$(0)
CALL IEEESEND(MATRIX, SADD(A$))
```

The command string **"B7C7X"** is the command string for the 706 that tell the instrument to display channel 7 and then close channel 7. Refer to the instrument manuals for a listing of all the command strings. The **CHR$(0)** is the null character that is required at the end of any command string sent to the QuickC subroutine.

ERROR Displays

- There are no error messages in this routine. If you do not use the null character at the end of the command string, there is the possibility that your computer will lock up and you will have to re-boot your system.

5.2.8 INIT

The **INIT** routine is used to initialize each device individually. The device names that can be used by this routine can be found in Appendix A. Unlike the other routines, this routine will initialize the bus if it is the first routine called. This routine is not needed if the **TSTSEL** routine is used, since **TSTSEL** initializes each instruments.

The calling sequence for this routine is:

Call INIT (DEVICE)

Where:

Name	Type	Operation	Description
Device	Integer	Input	The device that is to be initialized

Programming Considerations

- Only needed if **TSTSEL** is **NOT** being used.

ERROR Displays

- If the string is received improperly.

5.2.9 LIMITV

The **LIMITV** routine is used to change the voltage limit on the current source. This routine can be called anytime after **TSTSEL** and **HLSEL.** This routine can be placed in the Van Der Pauw routines or the Hall routines if so desired.

The calling sequence for this routine is:

CALL **LIMITV** (DEVICE, VLIMIT)

Where:

Name	Type	Operation	Description
Device	Integer	Input	The device who's voltage limit should be changed.
Vlimit	Single	Input	The desired voltage limit for DEVICE (volts).

Programming Considerations

- By default the voltage limit on the current source is set to 10 volts.

ERROR Displays

- If string is received improperly by the DEVICE.

5.2.10 MEASI

The **MEASI** routine is used to measure the current using the given device. This routine returns the measured value in the parameter list. The **MEASI** routine checks to see if the value is overranged. If it is, **MEASI** will return a value of **1.0E+22.** The **MEASI** routine also checks to see if the current source is in voltage limit. If this is the case, a value of **2.0E+22** will be returned. The legal devices that can be used in this routine can be found in Appendix A. Be sure that the instruments have settled before the **MEASI** routine is called or the received reading will be inaccurate.

The calling sequence for this routine is:

CALL MEASI (DEVICE, AMPS)

Where:

Name	Type	Operator	Description
Device	Integer	Input	Device name being used to measure the current.
Amps	Single	Output	The current read from the device (amps).

Programming Considerations

- Watch for returned values that are overranged, **1.0E+22**.

- Watch for values that show that the current source is in voltage limit, **2.0E+22**.

- **MEASI** cannot be used if the current meter is connected to the same pin as the current source.

- Make sure that the instruments have settled before **MEASI** is called or the received reading will be inaccurate.

- Make sure **TSTSEL** is the first routine called.

- The measured value **CANNOT** be returned directly into an array. You must first return the value to a variable and then place the variable value into an array. See example below.

EXAMPLE

```
call MEASI(IMTR1,AMPS)
arr(1) = AMPS
```

The example above places the measured value into an array **arr**.

ERROR Displays

- If MEASI is used while the current meter is connected to the same pin as the current source.

- If the strings sent to the instruments are received improperly.

- If an illegal device is used in the **MEASI** routine.

5.2.11 MEASV

The **MEASV** routine is used to measure the voltage using the given device. The measured voltage will then be returned in the parameter list. The **MEASV** routine can be used at anytime after the **TSTSEL** routine. A list of the legal devices that can be used by MEASV can be found in Appendix A. **MEASV** will return a value of **1.0E+22** if the reading is overranged. **MEASV** will return a value of **2.0E+22** if the current source is in voltage limit. Make sure you have allowed time for the instruments to settle before taking your measurement.

The calling sequence for this routine is:

CALL MEASV (DEVICE, VOLTS)

Where:

Name	Type	Operation	Description
Device	Integer	Input	Device name used to measure the voltage.
Volts	Single	Output	Measured voltage (volts).

Programming Considerations

- Watch for returned values that are overranged, **1.0E+22**.

- Watch for returned values that show the current is in voltage limit, **2.0E+22**.

- Make sure that the instruments have settled before **MEASV** is called or the received reading will be inaccurate.

- Make sure **TSTSEL** is the first routine called.

- The measured value **CANNOT** be returned directly into an array. You must first return the value to a variable and then place the variable value into an array. See example in **MEASI** routine.

ERROR Displays

- If an illegal device is being used in the **MEASV** routine.

- If the strings sent to the instrument are received improperly.

5.2.12 RANGEI

The **RANGEI** routine is used to set the range on the current meter. This routine can be used at anytime after the **TSTSEL** routine. **RANGEI** is best used immediately before the **MEASI** routine associated with it for program clarity. **RANGEI** allows you to select the range (in amps) which is desired. If the value selected is not a supported range value, then the next higher range will be selected. **RANGEI** can also be used to place the device into autorange mode. This is done by setting RANGE to 0. This is not normally necessary since the current meter is set to autorange by default. The range of the current meter will be reset to autorange when a **CONNECT** routine is called after a **FORCEI**. See the **CONNECT** routine for details. A list of the legal devices that **RANGEI** can use can be found in Appendix A.

The calling sequence for this routine is:

CALL **RANGEI** (DEVICE, RANGE)

Where:

Name	Type	Operation	Description
Device	Integer	Input	Device name for which range is to be set.
Range	Single	Input	The selected range for the current meter (amps).

Programming Considerations

- Do not have to use **RANGEI** if auto range is desired, unless the range has been changed and a FORCEI and CONNECT sequence has not been used since. See **CONNECT** routine for details.

- You should call **RANGEI** before each **MEASI** to insure that the measurement is being made on the appropriate range. See **CONNECT** routine for details.

- Make sure RANGE is a valid range for the DEVICE. See device manual for details.

ERROR Displays

- If an illegal DEVICE is being used in the **RANGEI** routine.

- If the strings sent to the instruments are received improperly.

5.2.13 RANGEV

The **RANGEV** routine is used to set the range on the voltmeter. This routine can be used at anytime after the **TSTSEL** routine. **RANGEV** is best used immediately before the **MEASV** routine for clarity. **RANGEV** allows you to select the range (in volts) which is desired. If the value selected is not a supported range value, then the next highest range supported will be selected. **RANGEV** can be used to place the device into autorange mode. This is done by setting RANGE to zero. This is not normally necessary since the voltmeter is set to autorange by default. The range of the voltmeter will be reset to autorange when a **CONNECT** routine is called after a **FORCEI**. See the **CONNECT** routine for details. A list of the legal devices for **RANGEV** can be found in Appendix A.

The calling sequence for this routine is:

CALL **RANGEV** (DEVICE, RANGE)

Where:

Name	Type	Operation	Description
Device	Integer	Input	Device name for which range is to be set.
Range	Single	Input	The selected range for the voltmeter (volts).

Programming Considerations

- Do not have to use **RANGEV** if auto range is desired, unless the range has been changed and a **FORCEI** and **CONNECT** sequence has not been used since. See **CONNECT** routine for details.

- You should call **RANGEV** before each **MEASV** to insure that the measurement is made on the appropriate range. See **CONNECT** routine for details.

- Make sure RANGE is a valid range for the DEVICE. See device manual for details.

- Make sure **TSTSEL** was called first.

ERROR Displays

- If an illegal DEVICE is being used in the **RANGEV** routine.

- If the strings sent to the instruments are received improperly.

5.2.14 SSMEASI

The **SSMEASI** routine is a steady state **MEASI** routine. This routine will continue taking measurements until either two measurements are within a specified percentage of each other, or the routine has reached the maximum number of readings allowed. This routine can replace any **MEASI** routine with minimal changes. This routine will return a value of **3.0E+22** if no steady state value is found within the number of ATTEMPTS allowed. A list of legal device names can be found in Appendix A.

The Calling Sequence for this routine is:

CALL SSMEASI (DEVICE, AMPS, PERC, ATTEMPTS, DLY)

Where:

Name	Type	Operation	Description
Device	Integer	Input	The device that should be used to make the measurement.
Amps	Single	Output	The amount of current read after the steady state conditions have been reached.
Perc	Single	Input	The percentage difference required between two readings before the routine will return a value. 1% should be entered as a 1.
Attempts	Integer	Input	The number of readings to be taken before returning a value of 3.0E+22.
Dly	Single	Input	The delay time required in (ms) between readings.

Programming Considerations

- This routine can be used anytime after the **TSTSEL** and **HLSEL** routines have been called.

- Watch for returned values of **3.0E+22**, which means no steady state value was found.

- Allow enough ATTEMPTS for the device to reach a steady state value.

- Allow enough DLY for the device to reach a steady state value.

- The measured value **CANNOT** be returned directly into an array. You must first return the value to a variable and then place the variable value into an array. See example in **MEASI** routine.

ERROR Displays

- If illegal device was used.

- If any strings sent to the devices are received improperly.

5.2.15 SSMEASV

The **SSMEASV** routine is a steady state **MEASV** routine. This routine will continue taking measurements until either two measurements are within a specified percentage of each other, or the routine has taken a specified number of readings. This routine can replace any **MEASV** routine with minimal changes. This routine will return a value of **3.0E+22** if no steady state value is found within the number of ATTEMPTS allowed. A list of legal device names can be found in Appendix A.

The calling sequence for this routine is:

CALL SSMEASV (DEVICE, VOLTS, PERC, ATTEMPTS, DLY)

Where:

Name	Type	Operation	Description
Device	Integer	Input	The device that should be used to make the measurement.
Volts	Single	Output	The voltage read after the steady state has been achieved
Perc	Single	Input	The percentage difference required between two readings before the routine will return a value. 1% should be entered as a 1.
Attempts	Integer	Input	The number of readings to be taken before returning a value of 3.0E+22.
Dly	Single	Input	The delay time required in (ms) between readings.

Programming Considerations

- This routine can be used anytime after **TSTSEL** and **HLSEL** routine have been called.

- Watch for returned values of **3.0E+22**, which means no steady state value was found.

- Allow enough ATTEMPTS for the device to reach a steady state value.

- Allow enough DLY for the device to reach a steady state value.

- The measured value **CANNOT** be returned directly into an array. You must first return the value to a variable and then place the variable value into an array. See example in MEASI routine.

ERROR Displays

- If illegal device was used.

- If any strings sent to the devices are received improperly.

5.2.16 TSTSEL

TSTSEL must be the first routine called in any program. This routine does two things. The first is that **TSTSEL** initializes the Bus and the instruments. Secondly it tells the instruments which quad box is to be tested. If the user's test system has only one quad box, then SAMPNUM should be set to 1. If the user has multiple quad box test stations then SAMPNUM should be set to the number of the quad box on which the user wishes to test. This routine must be called once at the beginning of the program. The only other time that this routine needs to be called is if the user wishes to change the quad box being used.

The following conditions are in effect once **TSTSEL** is called:

- All relays are opened and all pathways are cleared.

- All current and voltage sources are taken out of operate mode.

- All meters are set to either current meter or voltage meter mode.

Appendix A lists the devices and their initialized states.

The calling sequence for the **TSTSEL** subroutine is:

CALL TSTSEL(SAMPNUM)

Where :

Name	Type	Operation	Description
Sampnum	Integer	Input	The quad box number to be tested.

Programming Considerations

- This routine must be the first routine called in order to initialize the Bus and the instruments.

- SAMPNUM is set to 1 for single quad box test systems, otherwise SAMPNUM should be set to the quad box number selected.

ERROR Displays

- Returns an error if the Bus is not initialized properly.

- Returns an error if the individual strings to initialize each device are not received properly.

5.3 Van Der Pauw Measurement

The next two functions perform the Van Der Pauw measurements. The only difference between the two is that the **VDP** function is used on the high resistance terminals, and the **VDPLOW** function is used on the low resistance terminals.

5.3.1 VDP

The **VDP** function is used to perform the Van Der Pauw measurements on the high resistance terminals. **VDP** is a function because inside the routine, calculations are done to find the resistivity. This is the returned function value. This value is only accurate if the sample is **1.0** cm thick. This is mainly used when testing the **1 G OHM** test box or the **1 M OHM** test box. Otherwise if the sample thickness is not **1.0 cm**, the calculations for the resistivity must be done after the **VDP** function is called. **VDP** returns, in the parameter list all the measured voltages and currents. The sample must be connected in a 1 - 2 - 3 - 4 clockwise order. If the connections are made in a 1 - 3 - 2 - 4 clockwise order or any other order, the readings will be of no use as far as Van Der Pauw calculations are concerned. This function is made up of **CONNECT, FORCEI, DELAY, MEASI, MEASV** routines. All of the restraints of these routines are in effect. This routine can be called anytime after the **TSTSEL** and **HLSEL** routines have been called. Be aware of returned voltages and currents of **1.0 E+22** and **2.0 E+22.** These values mean that the value is overranged or that the current source is in voltage limit, respectively. If a **Divide By Zero** error occurs in the calculations, it could be because both the voltages and the currents were overranged or the current source was in voltage limit. This can be seen better by analyzing the calculation section of the program.

The calling sequence for this function is:

RES = **VDP** (P1, P2, P3, P4, AMPS, DLY, VLIMIT, V1, V2, V3, V4, V5, V6,V7,V8,
A1,A2,A3,A4, A5,A6,A7, A8, FAIL.FLAG)

Where:

Name	Type	Operation	Description
Res	Single	Output	Resistance found by using 1 cm as default of sample thickness.
P1	Integer	Input	Sample pin # that is connected to pin 1 of the test system.
P2	Integer	Input	Sample pin # that is connected to pin 2 of the test system.
P3	Integer	Input	Sample pin # that is connected to pin 3 of the test system.
P4	Integer	Input	Sample pin # that is connected to pin 4 of the test system.

Amps	**Single**	**Input**	**The current that will be forced through the sample (amps).**
Dly	**Single**	**Input**	**The settling time desired after the FORCEI has been called and before any measurement is taken (ms).**
Vlimit	**Single**	**Input**	**Voltage limit (volts). This is not used in the VDP routine. The program automatically sets the voltage limit to 10 volts. Modifications must be made to the VDP routine in order to use this variable.**
V1	**Single**	**Output**	**Voltage reading of V12,34 (volts).**
V2	**Single**	**Output**	**Voltage reading of V21,34 (volts).**
V3	**Single**	**Output**	**Voltage reading of V23,41 (volts).**
V4	**Single**	**Output**	**Voltage reading of V32,41 (volts).**
V5	**Single**	**Output**	**Voltage reading of V34,12 (volts).**
V6	**Single**	**Output**	**Voltage reading of V43,12 (volts).**
V7	**Single**	**Output**	**Voltage reading of V41,23 (volts).**
V8	**Single**	**Output**	**Voltage reading of V14,23 (volts).**
A1	**Single**	**Output**	**Current reading of A12,34 (amps).**
A2	**Single**	**Output**	**Current reading of A21,34 (amps).**
A3	**Single**	**Output**	**Current reading of A23,41 (amps).**
A4	**Single**	**Output**	**Current reading of A32,41 (amps).**
A5	**Single**	**Output**	**Current reading of A34,12 (amps).**
A6	**Single**	**Output**	**Current reading of A43,12 (amps).**
A7	**Single**	**Output**	**Current reading of A41,23 (amps).**
A8	**Single**	**Output**	**Current reading of A14,23 (amps).**
Fail.flag	**Integer**	**Output**	**Flag = 1 if one or more of the measurements is overranged.**

Programming Considerations

- Make sure that the sample is connected in a 1 - 2 - 3 - 4 clockwise configuration.

- Make sure that the delay is long enough to let the instruments settle after a current is forced and before measurements are made.

- The returned value of RES is calculated assuming that the sample thickness is **1.0** cm.

- P1, P2, P3, P4 must be integer values of 1, 2, 3, 4 in any order.

- Watch for the returned measured values being overranged or in voltage limit. This could result in a **Divide By Zero** error when making the calculations.

- Calculations to find the resistivity using the actual sample thickness must be done outside of the function call using the returned values.

- The measured value **CANNOT** be returned directly into an array. You must first return the value to a variable and then place the variable value into an array. See example in **MEASI** routine.

ERROR Displays

- See the **CONNECT, FORCEI, MEASV, MEASI** routines for individual routine error displays. No other error statements are possible.

5.3.2 VDPLOW

The **VDPLOW** function is used to perform the Van Der Pauw measurements on the low resistance terminals. Since it is impossible to measure the current on the low terminals, it is assumed that the forced current is the measured current. **VDPLOW** is a function because inside the routine calculations are made to determine the resistivity. This is the returned function value. This value is only accurate if the sample is **1.0 cm** thick. This is mainly used when testing the **1 G OHM** test box or the **1 M OHM** test box. Otherwise if the sample thickness is not **1.0** cm, then the calculations for the resistivity must be done after the **VDPLOW** function is called. The **VDPLOW** returns, in the parameter list all the measured voltages and currents. The sample must be connected in a 1 - 2 - 3 - 4 clockwise order. If the connections are made in a 1 - 3 - 2 - 4 clockwise order or any other order, then the readings will be of no use as far as the Van Der Pauw calculations are concerned. This function is made up of **CONNECT, FORCEI, DELAY, MEASI, MEASV** routines. So all of the restraints of these routines are in effect. This routine can be called anytime after the **TSTSEL** and **HLSEL** routines have been called. Be aware of returned voltages and currents of **1.0 E+22** and **2.0 E+22.** These values mean that the value is overranged or that the current source is in voltage limit, respectively. If a **Divide By Zero** error occurs in the calculations, it could be because both the voltages and the currents were overranged or the current source was in voltage limit. This can be seen better by analyzing the calculation section of the program.

The calling sequence for this function is:

RES = **VDPLOW** (P1, P2, P3, P4, AMPS, DLY, VLIMIT, V1, V2, V3, V4, V5, V6,V7,V8, A1,A2,A3,A4, A5,A6,A7, A8, FAIL.FLAG)

Where:

Name	Type	Operation	Description
Res	Single	Output	The resistivity assuming 1.0 cm as the thickness of the sample.
P1	Integer	Input	Sample pin # that is connected to pin 1 of the test system.
P2	Integer	Input	Sample pin # that is connected to pin 2 of the test system.
P3	Integer	Input	Sample pin # that is connected to pin 3 of the test system.
P4	Integer	Input	Sample pin # that is connected to pin 4 of the test system.
Amps	Single	Input	The current that will be forced through the sample (amps).
Dly	Single	Input	The settling time desired after the FORCEI has been called and before any measurement is taken (ms).
Vlimit	Single	Input	Voltage limit (volts). This is not used in the VDPLOW routine. The program automatically sets the voltage limit to 10 volts. Modifications must be made in the VDPLOW routine in order to use this variable.
V1	Single	Output	Voltage reading of V12,34 (volts).
V2	Single	Output	Voltage reading of V21,34 (volts).
V3	Single	Output	Voltage reading of V23,41 (volts).
V4	Single	Output	Voltage reading of V32,41 (volts).
V5	Single	Output	Voltage reading of V34,12 (volts).
V6	Single	Output	Voltage reading of V43,12 (volts).
V7	Single	Output	Voltage reading of V41,23 (volts).
V8	Single	Output	Voltage reading of V14,23 (volts).
A1	Single	Output	A12,34 = Amps (amps).

A2	Single	Output	A21,34 = -Amps (amps).
A3	Single	Output	A23,41 = Amps (amps).
A4	Single	Output	A32,41 = -Amps (amps)
A5	Single	Output	A34,12 = Amps (amps).
A6	Single	Output	A43,12 = -Amps (amps).
A7	Single	Output	A41,23 = Amps (amps).
A8	Single	Output	A14,23 = -Amps (amps).
Fail.flag	Integer	Output	Flag = 1 if one or more of the measurements is overranged.

Programming Considerations

- Make sure that the sample is connected in a 1 - 2 - 3 - 4 clockwise configuration.

- Make sure that the delay is long enough to allow the instruments to settle after a current is forced and before measurements are made.

- The returned value of RES is calculated assuming that the sample thickness is **1.0** cm.

- P1, P2, P3, P4 must be integer values of 1, 2, 3, 4 in any order.

- Watch for the returned measured values being overranged or in voltage limit. This could result in a **Divide By Zero** error when making the calculations.

- Calculations to find the resistivity using the actual sample thickness must be done outside of the function call using the returned values.

- The current returned from **VDPLOW** is **NOT** a measured current. It is the equal to the current forced into the sample.

- The measured value **CANNOT** be returned directly into an array. You must first return the value to a variable and then place the variable value into an array. See example in **MEASI** routine.

ERROR Displays

- See the **CONNECT, FORCEI, MEASV, MEASI** routines for individual routine error displays. No other new error statements are possible.

5.4 HALL Measurement

The **HALL** routine is called when making measurements with the magnet. This routine makes the measurements that are needed to calculate the mobility, carrier concentration, Hall coefficients, and the sample type (N or P). There are two different Hall routines. The first, called **HALL,** is used for the HIGH resistance terminals. The other called **HALLOW** is the same routine except that it uses the low resistance terminals.

5.4.1 HALL Routine

This routine, as mentioned above makes the required measurements needed to calculate the mobility, carrier concentration, Hall coefficients, and the sample type. This routine can be called at anytime after the **TSTSEL** and the **HLSEL** routines. Following this routine should be the equations used to calculate the above values. This routine should be used only on the high resistance terminals. This routine uses the **FORCEI, MEASI, MEASV,** and **DELAY** routines. Therefore all the programming considerations for these routines should be taken into consideration when using this routine.

The calling sequence for this routine is:

CALL **HALL** (P1, P2, P3, P4, AMPS, DLY, VLIMIT, V1, V2, V3, V4, I1, I2, I3, I4,
FAIL.FLAG)

where:

Name	Type	Operation	Description
P1	Integer	Input	The sample pin number that is connected to pin #1 of the test system.
P2	Integer	Input	The sample pin number that is connected to pin #2 of the test system.
P3	Integer	Input	The sample pin number that is connected to pin #3 of the test system.
P4	Integer	Input	The sample pin number that is connected to pin #4 of the test system.
AMPS	Single	Input	The current that will be forced through the sample (amps).
DLY	Single	Input	The delay in ms, that should be placed between the FORCEI and the MEASV or MEASI routines.
VLIMIT	Single	Input	Voltage limit (volts). This is not used in the HALL routine. The program automatically sets the voltage limit to 10 volts. Modifications must be made in the HALL routine in order to use this variable.

V1	Single	Output	V13,42 (volts).
V2	Single	Output	V31,42 (volts).
V3	Single	Output	V24,13 (volts).
V4	Single	Output	V42,13 (volts).
I1	Single	Output	I13,42 (amps).
I2	Single	Output	I31,42 (amps).
I3	Single	Output	I24,13 (amps).
I4	Single	Output	I42,13 (amps).
Fail.flag	Integer	Output	Flag = 1 if one or more of the measurements is overranged.

Programming Considerations

- This routine must follow the **TSTSEL** and **HLSEL** routines.
- The sample **MUST** be connected in a 1-2-3-4 clockwise configuration
- The **FORCEMAG** routine should be used before this routine in order to set the magnet.
- Equations for the mobility, etc. should follow this routine.
- Make sure the delay is long enough so that the instruments have time to settle before taking a measurement.
- Look at **FORCEI, DELAY, MEASV, MEASI** for further programming considerations
- The measured value **CANNOT** be returned directly into an array. You must first return the value to a variable and then place the variable value into an array. See example in MEASI routine.

ERROR Displays

- Consult the **FORCEI, DELAY, MEASV, MEASI** routines for error displays.

5.4.2 HALLOW Routine

This routine makes the measurements needed to calculate the mobility, carrier concentration, Hall coefficients, and sample type. This routine can be used anytime after the **TSTSEL** and the **HLSEL** routines have been called. Following this routine should be the equations used to calculate the above values. This routine should be used only on the low resistance terminals. This routine uses the **FORCEI, MEASI, MEASV,** and **DELAY** routines. Therefore all the programming considerations for these routines should be taken into consideration when using this routine. The sample **MUST** be connected in a 1-2-3-4 clockwise configuration.

The calling sequence for this routine is:

CALL **HALLOW** (P1, P2, P3, P4, AMPS, DLY, VLIMIT, V1, V2, V3, V4, I1, I2, I3, I4, FAIL.FLAG)

Where:

Name	Type	Operation	Description
P1	Integer	Input	The sample pin number that is connected to pin #1 of the test system.
P2	Integer	Input	The sample pin number that is connected to pin #2 of the test system.
P3	Integer	Input	The sample pin number that is connected to pin #3 of the test system.
P4	Integer	Input	The sample pin number that is connected to pin #4 of the test system.
AMPS	Single	Input	The current that will be forced through the sample.
DLY	Single	Input	The delay in ms, between the FORCEI and the MEASV or MEASI routines.
VLIMIT	Single	Input	Voltage limit (volts). This is not used in the HALLOW routine. The program automatically sets the voltage limit to 10 volts. Modifications must be made in the HALLOW routine in order to use this variable.
V1	Single	Output	V13,42 (volts).
V2	Single	Output	V31,42 (volts).
V3	Single	Output	V24,13 (volts).
V4	Single	Output	V42,13 (volts).
I1	Single	Output	I13,42 (amps).
I2	Single	Output	I31,42 (amps).
I3	Single	Output	I24,13 (amps).
I4	Single	Output	I42,13 (amps).
Fail.flag	Integer	Output	Flag = 1 if one or more of the measurements is overranged.

Programming Considerations

- This routine must follow the **TSTSEL** and **HLSEL** routines.

- The sample **MUST** be connected in a 1-2-3-4 clockwise configuration

- The **FORCEMAG** routine should be used before this routine in order to set the magnet.

- Equations for the mobility, etc. should follow this routine.

- Make sure the delay is long enough so that the instruments have time to settle.

- Look at **FORCEI, DELAY, MEASV, MEASI** for further programming considerations

- The measured value **CANNOT** be returned directly into an array. You must first return the value to a variable and then place the variable value into an array. See example in **MEASI** routine.

ERROR Displays

- Consult the **FORCEI, DELAY, MEASV, MEASI** routines for error displays.

5.5 HLB Routine (Hall-Bridge Measurement)

The HLB routine is used in the **HALBRDG.BAS** program to measure resistivity. This routine is used in the same way as the **VDP** routine. There are two differences between **HLB** and **HALL**. The first is that **HLB** does not measure the current being forced on either the high or low terminals. Therefore it does not need two separate routines like the **VDP** and **VDPLOW** . The second difference is that the **HLB** makes different measurements than the **VDP** routine. Both routines are used to calculate the resistivity.

5.5.1 HLB

The **HLB** function is used to perform the resistivity measurements for the HALL-BRIDGE TEST. **HLB** is a function because inside the routine, calculations are done to find the resistivity. This is the returned function value. This value is only accurate if the sample is **1.0 cm** thick, wide, and the spacing is **1.0 cm**. Otherwise if the sample thickness is not **1.0 cm**, the calculations for the resistivity must be done after the **HLB** function is called. **HLB** returns, in the parameter list all the measured voltages and currents. The sample must be connected in a 1 - 2 - 3 - 4 clockwise order. If the connections are made in a 1 - 3 - 2 - 4 clockwise order or any other order, the readings will be of no use as far as Hall-Bridge calculations are concerned. This function is made up of **CONNECT, FORCEI, DELAY, MEASI, MEASV** routines. All of the restraints of these routines are in effect. This routine can be called anytime after the **TSTSEL** and **HLSEL** routines have been called. Be aware of returned voltages and currents of **1.0E+22** and **2.0E+22**. These values mean that the value is overranged or that the current source is in voltage limit, respectively. If a **Divide By Zero** error occurs in the calculations, it could be because both the voltages and the currents were overranged or the current source was in voltage limit. This can be seen better by analyzing the calculation section of the program.

The calling sequence for this function is:

RES = **HLB** (P1, P2, P3, P4, AMPS, DLY, VLIMIT, V1, V2, V3, V4, A1,A2,A3,A4,
FAIL.FLAG)

Where:

Name	Type	Operation	Description
Res	Single	Output	Resistance found by using 1 cm as default of sample thickness, sample width, and the sample spacing.
P1	Integer	Input	Sample pin # that is connected to pin 1 of the test system.
P2	Integer	Input	Sample pin # that is connected to pin 2 of the test system.
P3	Integer	Input	Sample pin # that is connected to pin 3 of the test system.
P4	Integer	Input	Sample pin # that is connected to pin 4 of the test system.
Amps	Single	Input	The current that will be forced through the sample (amps).
Dly	Single	Input	The settling time desired after the FORCEI has been called and before any measurement is taken (ms).
Vlimit	Single	Input	Voltage limit (volts). This is not used in the HLB routine. The program automatically sets the voltage limit to 10 volts. Modifications must be made to the HLB routine in order to use this variable.
V1	Single	Output	Voltage reading of V12,46(volts).
V2	Single	Output	Voltage reading of V21,46(volts).
V3	Single	Output	Voltage reading of V23,35(volts).
V4	Single	Output	Voltage reading of V32,35(volts).
A1	Single	Output	Current reading of A12,46(amps).
A2	Single	Output	Current reading of A21,46(amps).

A3	**Single**	**Output**	**Current reading of A23,35(amps).**
A4	**Single**	**Output**	**Current reading of A32,35(amps).**
Fail.flag	**Integer**	**Output**	**Flag = 1 if one or more of the measurements is overranged.**

Programming Considerations

- Make sure that the sample is connected in a 1 - 2 - 3 - 4 clockwise configuration.

- Make sure that the delay is long enough to let the instruments settle after a current is forced and before measurements are made.

- The returned value of RES is calculated assuming that the sample thickness, sample width, and the sample spacing is **1.0 cm.**
- P1, P2, P3, P4 must be integer values of 1, 2, 3, 4 in any order.

- Watch for the returned measured values being overranged or in voltage limit. This could result in a **Divide By Zero** error when making the calculations.

- Calculations to find the resistivity using the actual sample thickness must be done outside of the function call using the returned values.

- The measured value **CANNOT** be returned directly into an array. You must first return the value to a variable and then place the variable value into an array. See example in MEASI routine.

ERROR Displays

- See the **CONNECT, FORCEI, MEASV, MEASI** routines for individual routine error displays. No other error statements are possible.

5.6 Hall - Bridge Routine

The **HLBHALL** routine is used in the **HALBRDG.BAS** program. This routine is used to make Hall measurements much like the **HALL** routine. There are only two difference between the two routines. The first is that **HLBHALL** does not measure the current being forced for either the high or low terminals, therefore there is no need to have two separate routines like the **HALL** and **HALLOW** routines. The second is that **HLBHALL** makes different measurements then the **HALL** routine.

5.6.1 HLBHALL Routine

This routine makes the measurements needed to calculate the mobility, carrier concentration, Hall coefficients, and sample type. This routine can be used anytime after the **TSTSEL** and the **HLSEL** routines have been called. Following this routine should be the equations used to calculate the above values. This routine should be used only on the low resistance terminals. This routine uses the **FORCEI, MEASI, MEASV,** and **DELAY** routines. Therefore all the programming considerations for these routines should be taken into consideration when using this routine. The sample **MUST** be connected in a 1-2-3-4 clockwise configuration.

The calling sequence for this routine is:

CALL **HLBHALL** (P1, P2, P3, P4, AMPS, DLY, VLIMIT, V1, V2, V3, V4, I1, I2, I3, I4, FAIL.FLAG)

Where:

Name	Type	Operation	Description
P1	Integer	Input	The sample pin number that is connected to pin #1 of the test system.
P2	Integer	Input	The sample pin number that is connected to pin #2 of the test system.
P3	Integer	Input	The sample pin number that is connected to pin #3 of the test system.
P4	Integer	Input	The sample pin number that is connected to pin #4 of the test system.
AMPS	Single	Input	The current that will be forced through the sample (Amps).
DLY	Single	Input	The delay in ms, between the FORCEI and the MEASV or MEASI routines.
VLIMIT	Single	Input	Voltage limit (volts). This is not used in the HLBHALL routine. The program automatically sets the voltage limit to 10 volts. Modifications must be made in the HLBHALL routine in order to use this variable.
V1	Single	Output	V12,65 (volts).
V2	Single	Output	V21,65 (volts).
V3	Single	Output	V12,43 (volts).
V4	Single	Output	V21,43 (volts).
I1	Single	Output	I12,65 (amps).

I2	Single	Output	I21,65 (amps).
I3	Single	Output	I12,43 (amps).
I4	Single	Output	I21,43 (amps).
Fail.flag	Integer	Output	Flag = 1 if one or more of the measurements is overranged.

Programming Considerations

- This routine must follow the **TSTSEL** and **HLSEL** routines.

- The sample **MUST** be connected in a 1-2-3-4 clockwise configuration

- The **FORCEMAG** routine should be used before this routine is called in order to set the magnet.

- Equations for the mobility, etc. should follow this routine.

- Make sure the delay is long enough so that the instruments have time to settle.

- Look at **FORCEI, DELAY, MEASV, MEASI** for further programming considerations

- The measured value **CANNOT** be returned directly into an array. You must first return the value to a variable and then place the variable value into an array. See example in **MEASI** routine.

ERROR Displays

- Consult the **FORCEI, DELAY, MEASV, MEASI** routines for error displays.

Chapter 6

SYSTEM TESTS

6.1 Introduction

This chapter gives a description on each of the tests that can be run on the system. Each test was created to assure that the system is functional. The test program can be found in the S110 directory. The test can be started by typing:

C: cd\ <CR>
C: cd s110 <CR>
C: systst

At this point a menu will appear to allow you to choose the test that you wish to run. Below are the names of the tests that can be run, along with a description of each test.

NOTE: You CAN only run one test at a time. After the test is done, the program WILL exit. If you wish to save the data recovered from the test, either rename the **TEST.DAT** file or print out **TEST.DAT** before you run **SYSTST** again.

6.2 Relay Test

Relay Test is a test used to check the relays in the buffer amplifier box. This test will test relays 1-38. This test is performed to test the continuity of each of the relays when they are open and when they are closed. Detailed diagrams on how each of the test is run on each relay, can be found in Appendix SYSTST. When this test is done, the screen will displays the continuity measured (relay closed), highest continuity allowed, ("*" will appear if the relay failed), measured continuity (relay open), lowest continuity allowed ("*" will appear if the relay failed). The results of the test will be stored in the file **TEST.DAT.**

6.3 Current Source

This test is designed to test the 220 current source. The test makes a connection between the current source and the 617 current meter. The current source is tested on the 2 na through the 20 ma ranges. The current meter is set to measure on the 2na through the 20 ma ranges. The 617 current ranges are staggered so that it is possible to tell which device is failing. Refer to the print out of the test results to see exactly how the ranges are staggered. Each range of the current source is tested with 18 different currents. The results from this test will be saved in the file **TEST.DAT**.

6.4 Current Meter

This test is set up just like the current source test. The 617 is tested between the ranges of 2na and 20ma. This time the current source is the one that has the staggered ranges. This test also forces 18 different currents per range. This test along with the **Current Source** test will show if the current source or the current meter is failing. A "*" will appear on the screen if the test has failed. The results of this test will be saved in the file **TEST.DAT**.

6.5 Voltmeter

The **Voltmeter** test tests the 196. This test connects the 196 to the 220. This test is performed by setting the compliance on the 220 from +-1 through +-10 and forcing a current of 1E-5. The results then must be checked by the user to see if the voltage read is close to the compliance voltage.

NOTE: The voltage read will **NOT** equal the compliance voltage. The voltage could vary by +-1 volt. There is no "*" to tell the user if the device has failed.

This test should be used along with the **Voltmeter & Op Amp Test w/ Sample** to see if the voltmeter has failed. The results from this test are saved in a file **TEST.DAT**.

6.6 Op Amps

The **Op Amp** test connects the current source with each of the op amps (separately) and the 196 (voltmeter) This test is done by setting the compliance from +-1 to +-10 volts and reading the voltage through the op amp. The results from this test must be checked by the user, there is no flag to say if the device has failed.

NOTE: The voltage read will **NOT** equal the compliance voltage set. The voltage could vary by +-1 volt.

This test should be used along with the **Voltmeter & Op Amp Test w/ Sample** to see if the op amps have failed. The results from this test are saved in a file **TEST.DAT**.

6.7 Zero Op Amps

Zero Op Amps is used to set the op amps to zero. The user adjusts the op amps from the front panel. The user can either read the voltage of the op amp on the 196 or can read it from the screen. Once the first op amp is set to zero, the user can strike <CR> and the second op amp will be connected. This test will run until all four op amps have been set.

6.8 Voltmeter & Op Amp Test w/ Sample

When this test is selected, the computer will prompt to connect the **1G OHM** test box to station number 1. The test will check to see if both the 196 voltmeter and the op amps are functioning properly. The test will first connect pins 1 and 3 of the sample and force a current. If the voltage measured is not within 1.75% of the expected value, then a "*" will appear. This test will force 10 currents per connection. The routine makes the following connections: 1-3, 2-4, 3-1,4-2. The results from this test are saved in the file **TEST.DAT**.

6.9 Test All

If **Test All** is selected, all the tests listed above **EXCEPT ZERO OP AMPS** will be run. All of the test results will be sent to the file **TEST.DAT**. A **Test All** should be run when the system is received and the results should be compared to the results in the **TEST1.DAT** file.

This Page Left Blank Intentionally

Chapter 7

PREPARING CUSTOM PROGRAMS

7.1 Introduction

The programs supplied with the system are good prototype programs to follow in preparing your own custom programs. These programs use all of the calls to the driver library (**S110.LIB**) in the proper fashion. These programs are supplied in both executable form (.EXE extension) and source form (.BAS extension) and are written in Microsoft ® QuickBASIC. Creating an executable program requires three steps: Modifying the source code, compiling the source code, and linking the object modules together.

7.2 Modifying the Source Code

The source code of the Hall software can be modified very easily within the QuickBASIC environment. Please refer to the QuickBASIC manuals for details on how to work within the QuickBASIC environment.

7.3 Compiling the Source Code & Linking the Program

If for any reason you have changed the Hall software or you have created your own Hall software, you must compile the new program. This is done by using **MAKEIT.BAT** batch file. This file is in the S110 sources directory. **MAKEIT.BAT** is executed by typing:

```
C: cd\<CR>
C: cd s110\sources <CR>
C: makeit filename
```

Where **filename** is any Hall QuickBASIC program. For example:

```
C: cd\ <CR>
C: cd s110\sources <CR>
C: makeit vdphal
```

This example will compile the **VDPHAL.BAS** program and create **VDPHAL.EXE. DO NOT** use the ".BAS" extension in the filename. Also **DO NOT** attempt to make any Hall software executable files from within the QuickBASIC environment.

NOTE: You can use the **MAKEALL.BAT** file that will make all the executable programs of all the Keithley basic programs provided with the system.

7.4 Using the Quick Library

The Hall software includes a quick library called **S110.QLB**. This file can be found in the S110 LIB directory. This quick library makes it possible to run the Hall software within the Quick Basic environment. This is done by getting into the S110 sources directory and loading Quick Basic with the **S110.QLB** quick library. Example:

```
C: cd\ <CR>
C: cd s110\sources <CR>
C: qb vdphal /l s110.qlb
```

This example loads the program **VDPHAL.BAS** with the **S110.QLB** quick library into the Quick Basic environment. The **VDPHAL.BAS** program can now be run by pressing the SHIFT KEY & the F5 key at the same time.

The **S110.QLB** quick library includes the files:

- 617.OBJ
- 196.OBJ
- 220.OBJ
- 706.OBJ
- WALKER.OBJ
- HI.OBJ
- S110DIS.OBJ
- IEEE488.LIB
- SC.LIB

7.5 Creating a Quick Library

To create the **S110.QLB** quick library for Hall software, the **MAKEQLB.BAT** batch file must be used. In order to use this batch file, all the object files from the S110 OBJ directory must be copied into the S110 LIB directory. This is done by typing:

C: copy c:\s110\obj*.obj c:\s110\lib*.obj <CR>

We now go to the S110 LIB directory and make the quick library. Example:

C: cd\ <CR>
C: cd s110\lib<CR>
C: makeqlb <CR>

The example above just created two files, The first file is **S110.LIB**, which includes all of the object files. The second is **S110.QLB**, which includes the objects and the two libraries.

You can now erase the object files in the S110 LIB directory by typing:

C: del *.obj <CR>

This Page Left Blank Intentionally

CHAPTER 8

PLOT PROGRAM

8.1 Introduction

This chapter gives a description on the plot program. This chapter will also explain how to run **PLOT**. A listing of the program can be seen in the **S110 SOURCES** directory in the files called **PLOT.BAS** and **PLOT2.BAS**.

The plot program can be used to make the following graphs:

- Log R vs (1/T) * 1000
- Log Mobility vs (1/T) * 1000
- Log Mobility vs Magnetic Field
- Sweep I vs Measure V

For all of the plots listed above, the plots are: 1) Printed to the screen. 2) The plot is saved to a file specified by the user so that it can be printed out later by using the **FPLOT** utility in GEOGRAF. 3) The data used to make the plot is also written to a file.

8.2 PLOT Program

The **PLOT** program is created by making its executable file from **PLOT.BAS** and **PLOT2.BAS** using the command procedure **MAKEALL.BAT**. Refer to Chapter 7 for details. Plot can also be run from within the QuickBASIC by typing, within the **S110 SOURCES** directory:

C: qb plot /l s110.qlb <CR>

Then press SHIFT-F5 to run.

The first menu will appear once the program has been executed. This menu allows the user to choose which plot(s) you would like by typing a "Y" next to the choices. You will also be allowed to name the file in which the graph will be saved.

Once you are done with your selection, press F3. The following menu will be the description menu for each of the plot(s) that you have selected. If you have chosen to make all four plots, then there will be four description menus to fill in. These menus will be described in detail later in this chapter. Once you have entered all the data into the menus, the program will start its calculations. The program will gather the data required for the particular plot that is to be made, it will then display the plot to the screen. This display will stay on the screen until return is pressed. Once return is pressed, it will save the plot in a file that the user entered in the main menu, "FILENAME.PLT". At this point the program will either gather data for another plot or if there are no other plots to be made, it will return to the main menu.

8.3 PLOT Descriptions

This section gives a detailed description about each of the plots. This includes how to use the menus for each of them and what calculations are made for each plot.

8.3.1 Log R vs (1/T) * 1000

This graph will be created if a "Y" is entered in the main menu next to:

1) Log R vs (1/T * 1000)

when the **PLOT** program is executed. The description menu that will appear for this graph has the user enter the following information.

- **Current in amps:** This is the amount of current that you would like to force when making this graph.

- **Delay in milliseconds:** This is the amount of delay that you would like between the **FORCEI** and the **MEASI, MEASV** routines.

- **HIGH or LOW terminals:** Enter either HIGH or LOW depending on which terminals you are using on the quad box.

- **Quad Box #:** Unless you have a multi-quad box system, the user would enter 1.

- **Sample Thickness in cm:** This is the thickness of the sample that is to be measured in cm.

- **Enter "Y" to enter your own temperature settings:** Enter "Y" here if you wish to enter your own temperature settings. Enter "N" here if you wish to enter a starting temperature, ending temperature, and the number of increments that should be taken to get from the starting temperature to the ending temperature.

The next three entrees are only used if a "N" was entered above

- **Starting Temperature:** This is the temperature at which the plot will start with in degrees Kelvin.

- **Ending Temperature:** This is the final temperature at which this plot will take a reading .

- **Number of increments:** This is the number of increments that the program should take from the starting temperature to the ending temperature.

The next section is only used if a "Y" was entered above

- **Enter number of settings:** this is the number of temperature settings that the user wishes to enter manually.

After the user is finished entering all the data above, press F3. If the user selected to enter their own temperature settings, then a screen will be displayed that will ask to enter the desired temperature setting. After each temperature is typed from the keyboard, **F3 MUST be pressed to prompt the user for another setting or to continue.**

This particular plot does the following . The plot first checks to make sure that a cryo is available, if not it will return an error and go to the next plot or return to the main menu. It will then set either the starting temperature or the first temperature entered, depending on if a "Y" or "N" was entered in the description menu. The program will then makes the measurements to calculate the resistivity using the **VDP** or the **VDPHAL** routines, refer to Chapter 4 for details. Once this calculation is made both the temperature and the resistivity is stored in an array. The program then loops back to see if there is another temperature setting. If there is, the program will repeat the same steps until the ending temperature has been reached or the last temperature that the user entered has been set.

Once all the data has been stored, it is sent to a file called "FILENAME1.DAT". The data is then used to set up the x and y axis for the plot and the points are plotted. Once the graph appears on the screen, it will be displayed until the return key is pressed. The program will then write the plot to a file called "FILENAME1.PLT".

8.3.2 Log Mob vs (1/T * 1000)

This graph will be created if a "Y" is entered in the main menu next to:

2) Log Mob vs (1/T * 1000)

when the **PLOT** program is executed. The description menu that will appear for this graph has the user enter the following information.

- **Current in amps:** This is the amount of current that you would like to force when making this graph.

- **Delay in milliseconds:** This is the amount of delay that you would like between the **FORCEI** and the **MEASI, MEASV** routines.

- **HIGH or LOW terminals:** Enter either HIGH or LOW depending on which terminals you are using on the quad box.

- **Quad Box #:** Unless you have a multi-quad box system, the user would enter 1.

- **Magnetic Field Setting (KG):** The magnetic field setting at which you wish to have the mobility calculated.

- **Sample Thickness in cm:** This is the thickness of the sample that is to be measured.

- **Enter "Y" to enter your own temperature settings:** Enter "Y" here if you wish to enter your own temperature settings. Enter "N" here if you wish to enter a starting temperature, ending temperature, and the number of increments that should be taken to get from the starting temperature to the ending temperature.

The next three entrees are only used if a "N" was entered above

- **Starting Temperature:** This is the temperature at which the plot will start with in degrees Kelvin.

- **Ending Temperature:** This is the final temperature at which this plot will make a reading.

- **Number of increments:** This is the number of increments that the program should take to get from the starting temperature to the ending temperature.

The next section is only used if a "Y" was entered above

- **Enter number of settings:** this is the number of temperature settings that the user wishes to enter manually.

After the user is finished entering all the data above, press F3. If the user selected to enter their own temperature settings, then a screen will be displayed that will ask to enter the desired temperature setting. After each temperature is typed from the keyboard, **F3 MUST be pressed to prompt the user for another setting or to continue.**

This particular plot does the following: The plot first checks to make sure that a cryo and the magnet are available, if either is not it will return an error and go to the next plot or return to the main menu. If they are available it will set either the starting temperature or the first temperature entered, depending on if a "Y" or "N" was entered in the description menu. The program then takes the measurements to calculate the resistivity using the **VDP** or the **VDPHAL** routines. Refer to chapter 4 for details. Once this calculation is made the mobility is then calculated by using the **HALL** or the **HALLOW** routines. The program then loops back to see if there is another temperature setting. If there is, the program will repeat the same steps until the ending temperature has been reached or the last temperature that the user entered has been set.

Once all the data has been stored, it is sent to a file called "FILENAME2.DAT". The data is then used to set up the x and y axis for the plot and the points are plotted. Once the graph appears on the screen, it will be displayed until the return key is pressed. the program will then write the plot to a file called "FILENAME2.PLT".

8.3.3 Log Mob vs Magnetic Field

This graph will be created if a "Y" is entered in the main menu next to:

3) Log Mob vs Magnetic Field

when the **PLOT** program is executed. The description menu that will appear for this graph has the user enter the following information.

- **Current in amps:** This is the amount of current that you would like to force when making this graph.

- **Delay in milliseconds:** This is the amount of delay that you would like between the **FORCEI** and the **MEASI, MEASV** routines.

- **HIGH or LOW terminals:** Enter either HIGH or LOW depending on which terminals you are using on the quad box.

- **Quad Box #:** Unless you have a multi-quad box system, the user would enter 1.

- **Sample Thickness in cm:** This is the thickness of the sample that is to be measured.

- **Temperature Setting (K):** This is the temperature at which you wish to have the mobility calculated.

- **Enter "Y" to enter your own magnetic field settings:** Enter "Y" here if you wish to enter your own magnetic field settings. Enter "N" here if you wish to enter a starting magnetic field, ending magnetic field, and the number of increments that should be taken to get from the starting magnetic field to the ending magnetic field.

The next three entrees are only used if a "N" was entered above

- **Starting Magnetic Field:** This is the magnetic field at which the plot will start with in KiloGauss.

- **Ending Temperature:** This is the final magnetic field at which this plot will make a reading .

- **Number of increments:** This is the number of increments that the program should take to get from the starting magnetic field to the ending magnetic field.

The next section is only used if a "Y" was entered above

- **Enter number of settings:** this is the number of magnetic field settings that the user wishes to enter manually.

After the user is finished entering all the data above, press F3. If the user selected to enter their own magnetic field settings, a screen will be displayed that will ask to enter the desired magnetic field setting. After each magnetic field is entered, **F3 MUST be pressed to prompt the user for another setting or to continue.**

This particular plot does the following : It first checks to make sure that a magnet is available, if it is not, it will return an error and go to the next plot or return to the main menu. At this point the temperature will be set. It will set either the starting magnetic field or the first magnetic field entered, depending on if a "Y" or "N" was entered in the description menu. The program then takes the measurements to calculate the resistivity using the **VDP** or the **VDPHAL** routines. Refer to chapter 4 for details. Once this calculation is made the mobility is then calculated using either the **HALL** or the **HALLOW** routine. The program then loops back to see if there is another temperature setting. If there is, the program will repeat the same steps until the ending magnetic field has been reached or the last magnetic field that the user entered has been set. All the data is stored into two separate arrays.

Once all the data has been stored, it is sent to a file called "FILENAME3.DAT". The data is then used to set up the x and y axis for the plot and the points are plotted. Once the graph appears on the screen, it will be displayed until the return key is pressed. the program will then write the plot to a file called "FILENAME3.PLT".

8.3.4 Sweep I vs Meas V

This graph will be created if a "Y" is entered in the main menu next to:

4) Sweep I vs Meas V

when the **PLOT** program is executed. The description menu that will appear for this graph has the user enter the following information.

- **Delay in milliseconds:** This is the amount of delay that you would like between the **FORCEI** and the **MEASI, MEASV** routines.

- **HIGH or LOW terminals:** Enter either HIGH or LOW depending on which terminals you are using on the quad box.

- **Quad Box #:** Unless you have a multi-quad box system, the user would enter 1.

- **Sample Thickness in cm:** This is the thickness of the sample that is to be measured.

- **Enter "Y" to enter your own current settings:** Enter "Y" here if you wish to enter your own temperature settings. Enter "N" here if you wish to enter a starting temperature, ending temperature, and the number of increments that should be taken to get from the starting temperature to the ending temperature.

The next three entrees are only used if a "N" was entered above

- **Starting current:** This is the current at which the plot will start with (Amps).

- **Ending current:** This is the final current at which this plot will take a reading.

- **Number of increments:** This is the number of increments that the program should take to get from the starting temperature to the ending temperature.

The next section is only used if a "Y" was entered above

- **Enter number of settings:** this is the number of current settings that the user wishes to enter manually.

After the user is finished entering all the data above, press F3. If the user selected to enter their own current settings, a screen will be displayed that will ask to enter the desired current setting. After each current is typed from the keyboard, **F3 MUST be pressed to prompt the user for another setting or to continue.**

This particular plot does the following: It will first make the appropriate connection, it will set either set the starting current or the first current entered, depending on if a "Y" or "N" was entered in the description menu. The program will then take the voltage measurement. The program then loops back to see if there is another current setting. If there is, the program will repeat the same steps until the ending current has been reached or the last current that the user entered has been set.

Once all the data has been stored, it is sent to a file called "FILENAME4.DAT". The data is then used to set up the x and y axis for the plot and the points are plotted. Once the graph appears on the screen, it will be displayed until the return key is pressed. The program will then write the plot to a file called "FILENAME4.PLT".

This page is left blank intentionally

APPENDIX A

A.1 Device Names Key

This section is a list of all the device names and their numbers, these will be used in all routines that require a DEVICE. The H that follows some of the device names means the high side of the device. The L that follows some of the device names, indicates the low side of that device.

NOTE: The names with H or L at the end should only be used in the connect routines.

A.1.1 Device 196

196 = VMTR2	
196 = VMTR2H	(high terminal)
196 = VMTR2L	(low terminal)

NOTE: VMTR2 is equal to VMTR2H

A.1.2 Device 220

220 = ISRC1

A.1.3 Device 617

617 = IMTR1
617 = IMTR1H
617 = VMTR1H
617 = GND (on the **HIGH PINS ONLY**)

NOTE: There is no IMTR1L since the low end is tied to ground.

A.1.4 Device 706

706 = MATRIX

A.2 Legal Device Names

A.2.1 CONNECT

DEVICE
- ISRC1
- VMTR1H
- VMTR2H
- VMTR2L
- IMTR1H
- GND

NOTE: Use GND to ground the current source on the high or low pin terminals.

A.2.2 FORCEV

DEVICE
- VSRC1

A.2.3 FORCEI

DEVICE
- ISRC1

A.2.4 INIT

DEVICE
- VMTR1
- IMTR1
- ISRC1
- VSRC1
- MATRIX
- VMTR2
- IMTR2

A.2.5 LIMITV

DEVICE
- ISRC1

A.2.6 MEASI

DEVICE
- IMTR1
- IMTR2

A.2.7 MEASV

DEVICE
- VMTR1
- VMTR2

A.2.8 RANGEI

DEVICE
- IMTR1
- IMTR2
- ISRC1

A.2.9 RANGEV

DEVICE
- VMTR1
- VMTR2

A.2.10 SSMEASI

DEVICE
- IMTR1
- IMTR2

A.2.11 SSMEASV

DEVICE
- VMTR1
- VMTR2

A.2.12 IEEESEND

DEVICE
- IEEE.196
- IEEE.220
- IEEE.617
- IEEE.706
- IEEE.CRYO

A.3 Initialization States

The listings below are the states that each device is set to when the devices are initialized. Refer to the device manuals for details on each setting.

A.3.1 Device196

K0 - Enable EOI and bus hold off on X
R0 - Autorange
F0 - DC volts
T1 - One shot on talk
S2 - 5 1/2 digit display
W0 - delay period zero
Q0 - One shot into buffer
M0 - SRQ disabled
Z0 - Zero disabled
Z1 - Zero enabled

A.3.2 Device 220

K0 - EOI transmitted on last byte out
R0 - Autorange
P1 - Continuous program mode
I0 - Current set to 0 amps
V10 - voltage limit set to 10 volts

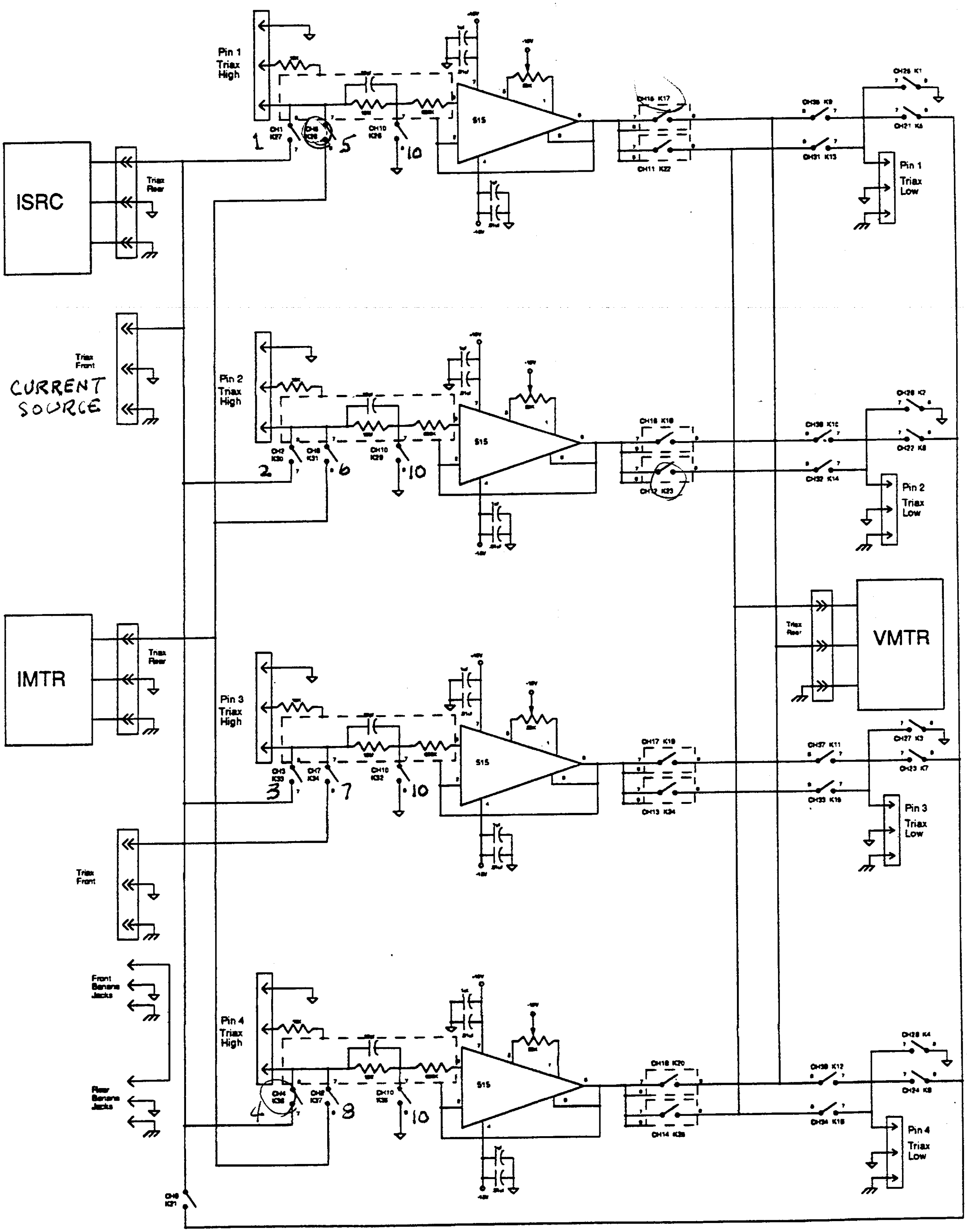
A.3.3 Device 617

V0 - Set voltage to 0 volts
O0 - V source off
D0 - Set to electrometer
F1 - Set to amps
R0 - Autorange
C1 - Zero check on
Z1 - Zero correct on
C0 - Zero check off

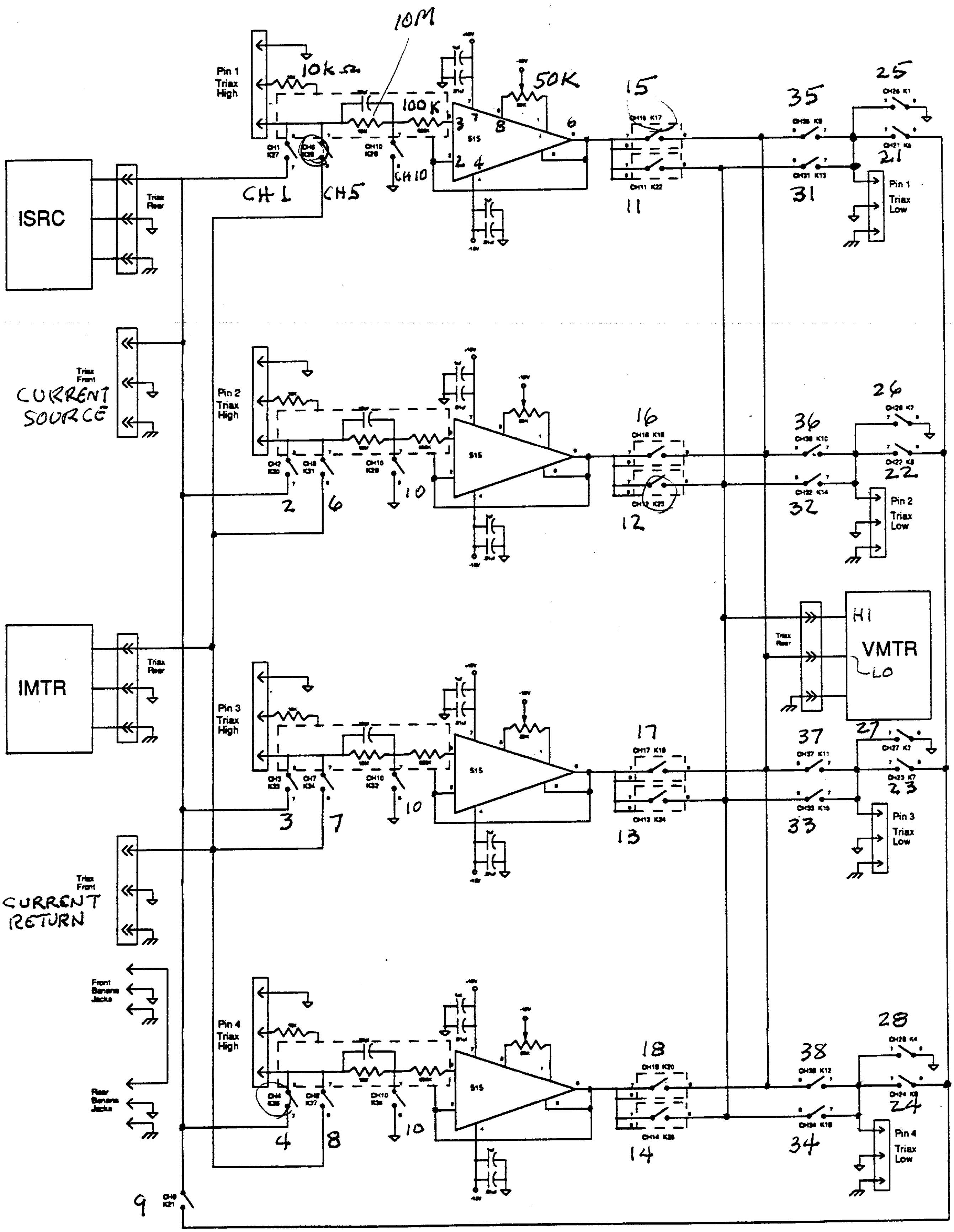
A.3.4 Device 706

K0 - EOI transmitted on last byte
R0 - Open all channels
A2 - 2 pole
D0 - Display channel
T6 - Start on external

This page is left blank intentionally



ISRC
IMTR
VMTR
CURRENT SOURCE
Triax Rear
Triax Front
Front Banana Jacks
Rear Banana Jacks
Pin 1 Triax High
Pin 2 Triax High
Pin 3 Triax High
Pin 4 Triax High
Pin 1 Triax Low
Pin 2 Triax Low
Pin 3 Triax Low
Pin 4 Triax Low
515
1
2
3
4
5
6
7
8
10



10M
10kΩ
100K
50K
Pin 1 Triax High
Pin 2 Triax High
Pin 3 Triax High
Pin 4 Triax High
ISRC
Triax Rear
CURRENT SOURCE
Triax Front
IMTR
CURRENT RETURN
Front Banana Jacks
Rear Banana Jacks
CH1
CH5
CH10
515
VMTR
HI
LO
Pin 1 Triax Low
Pin 2 Triax Low
Pin 3 Triax Low
Pin 4 Triax Low
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
21
22
23
24
25
26
27
28
31
32
33
34
35
36
37
38

TO: GREG SMITH
937/255-3636 x 4564
937/904-8122 FAX

10/25/01
p1 OF 2

FROM: JOHN YEAGE
KEITHLEY

Post-it® Fax Note 7671	Date 3/22/04	# of pages ▶ 1
To CHRIS CAYLOR	From JOHN YEAGER	
Co./Dept. RTI	Co. KEITHLEY	
Phone # -1280	Phone #	
Fax # 919/541-6515	Fax #	

CLOSE CHANNEL	CENTER PIN OF HIGH TRIAX	[illegible]
1	1	LOW R ($<0.3\Omega$)
2	2	
3	3	
4	4	
5	1	TO CENTER PIN OF CURRENT RETURN LOW R
6	2	
7	3	
8	4	
10	CENTER PIN OF HIGH TRIAX 1 THRU 4	TO INSIDE SHIELD OF CURRENT SOURCE

$10M\Omega$

10/26/01 Added 5Ω in series w/ +6V to inverter; now ~ 4.8V in, ∓15V or
All op amps zero except #1
10/29/01 Added 10µF across inverter input. Replaced #1 op amp w/ AD515.
10/31/01 Replaced CH11 relay (RL-77)

937/255 - 1929 x 4855 (lab)
-3636 x 4564 (office)
937/904-8122 FAX

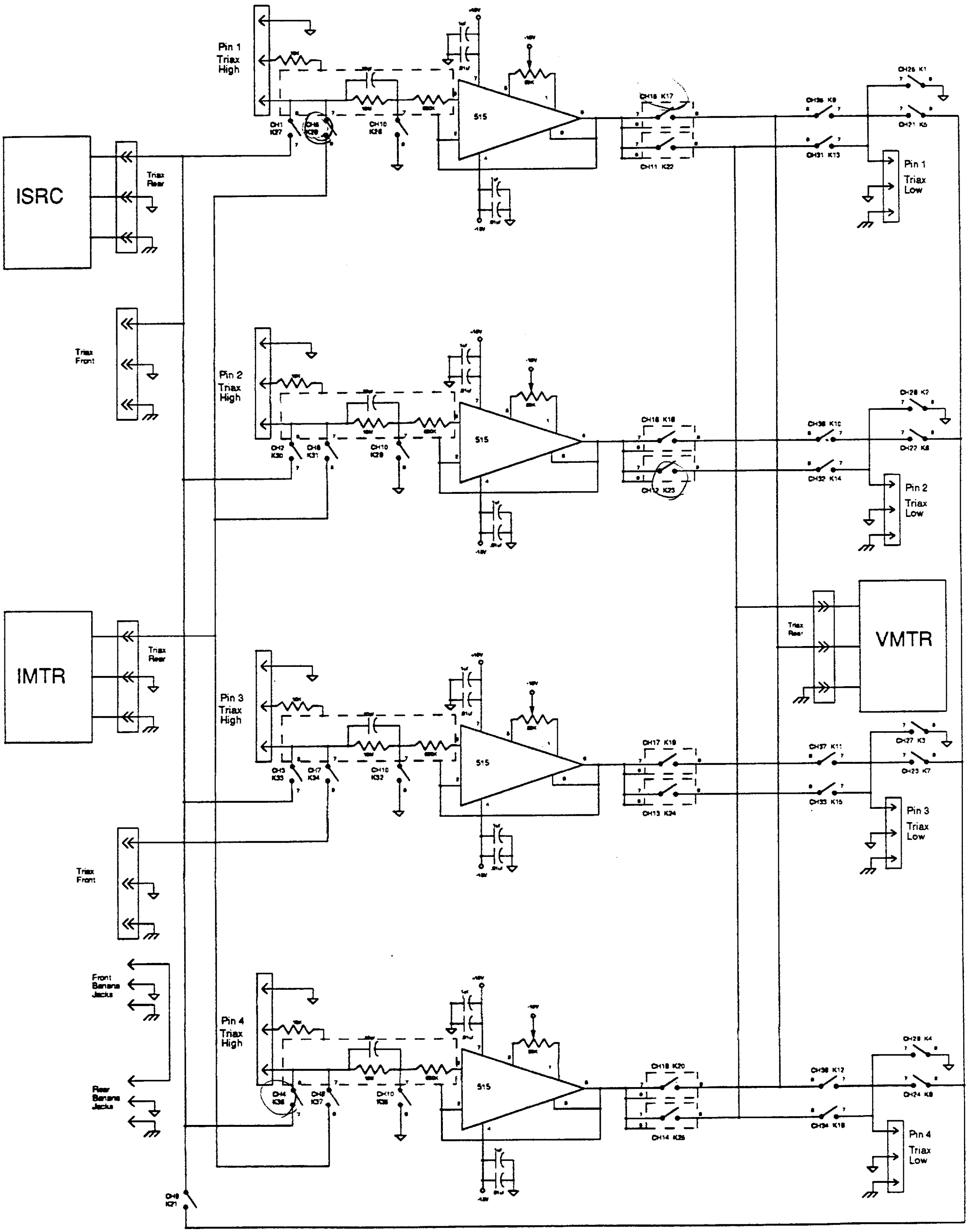
PS-18 5V in / ±15V out Computer Products, Stevens-Arnold Div. BA05D15/250L

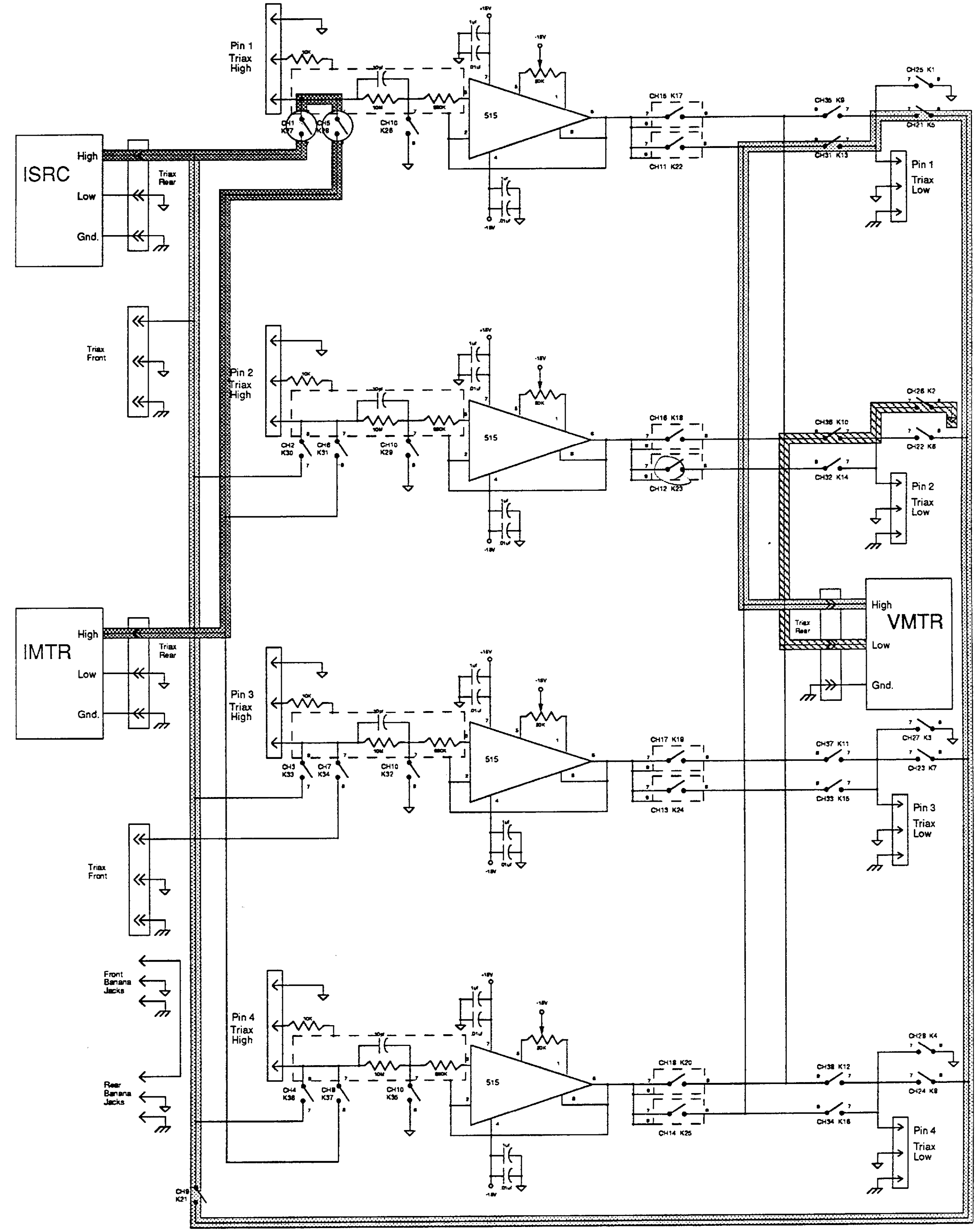
12/6/01 Checked 5110 out w/ quad 1GΩ - looks good, even after putting all shields, cover in place. Ship today. UPS overnite ✓

12/10/01 11^27 LW

APPENDIX B

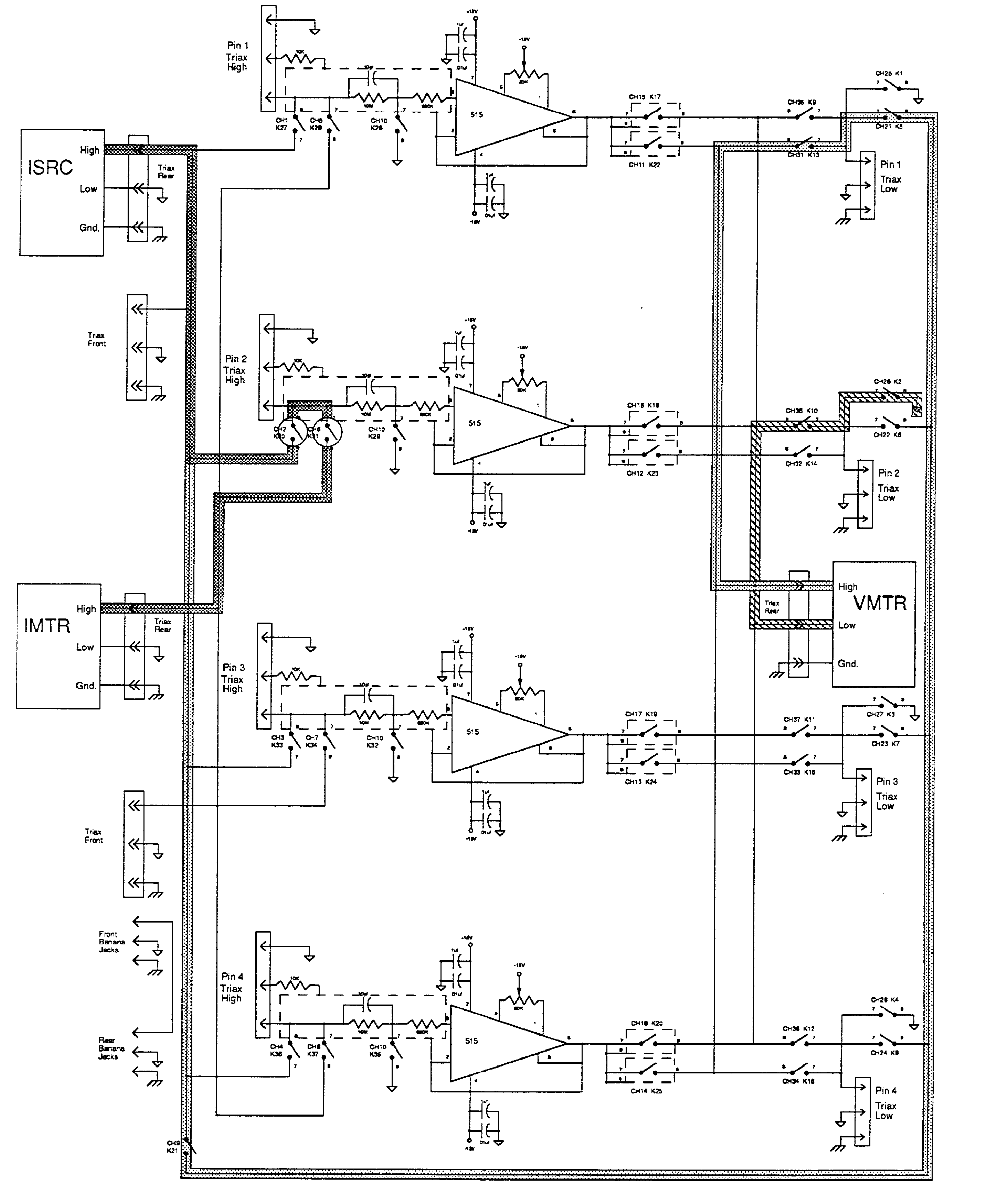
B.1 SYSTST Relay Test Diagrams


ISRC
IMTR
VMTR
Pin 1
Triax
High
Pin 2
Triax
High
Pin 3
Triax
High
Pin 4
Triax
High
Pin 1
Triax
Low
Pin 2
Triax
Low
Pin 3
Triax
Low
Pin 4
Triax
Low
Triax
Rear
Triax
Front
Front
Banana
Jacks
Rear
Banana
Jacks
515



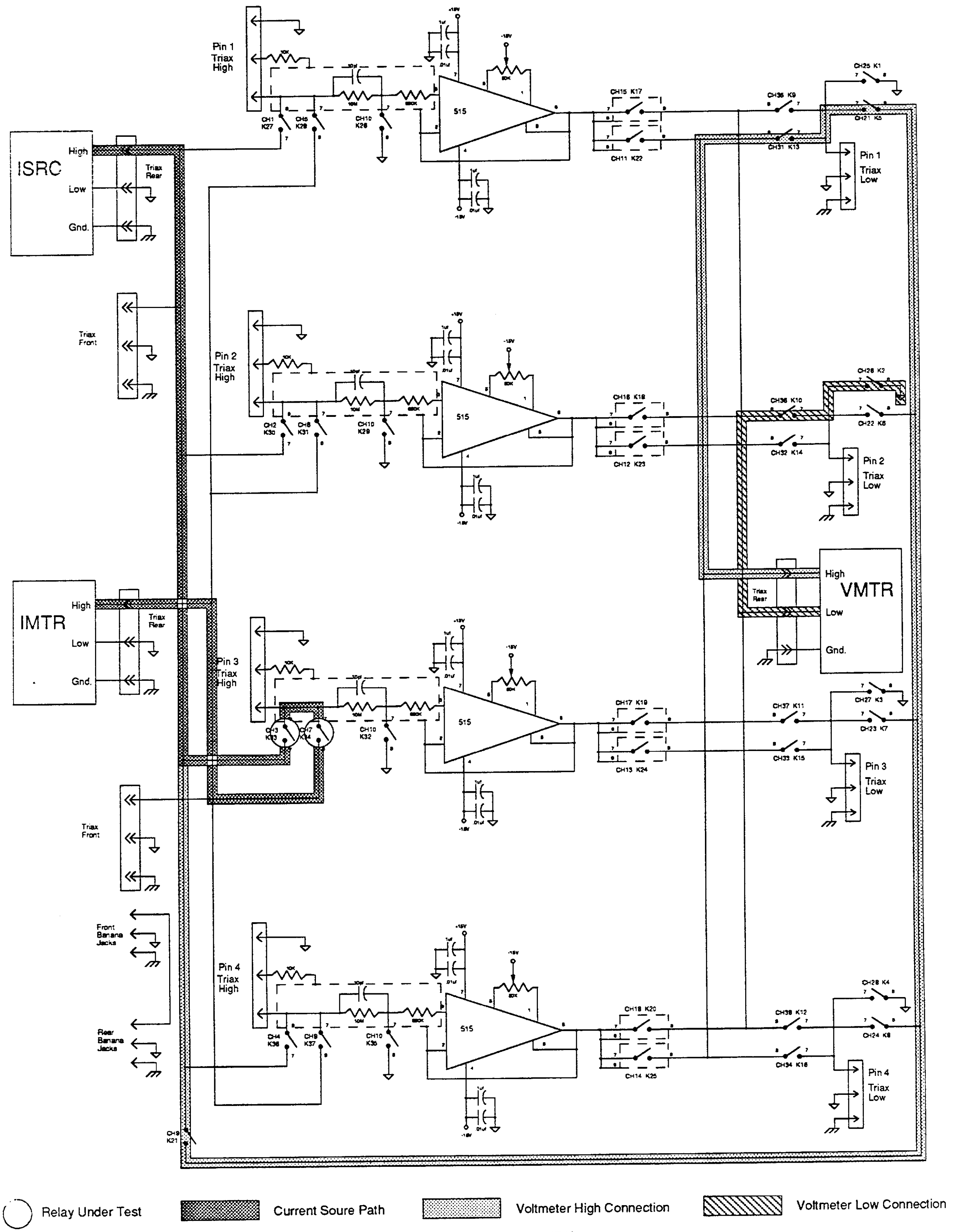
Relay Under Test | Current Soure Path | Voltmeter High Connection | Voltmeter Low Connection

Relays Being Tested = CH1,K27 CH5,K28

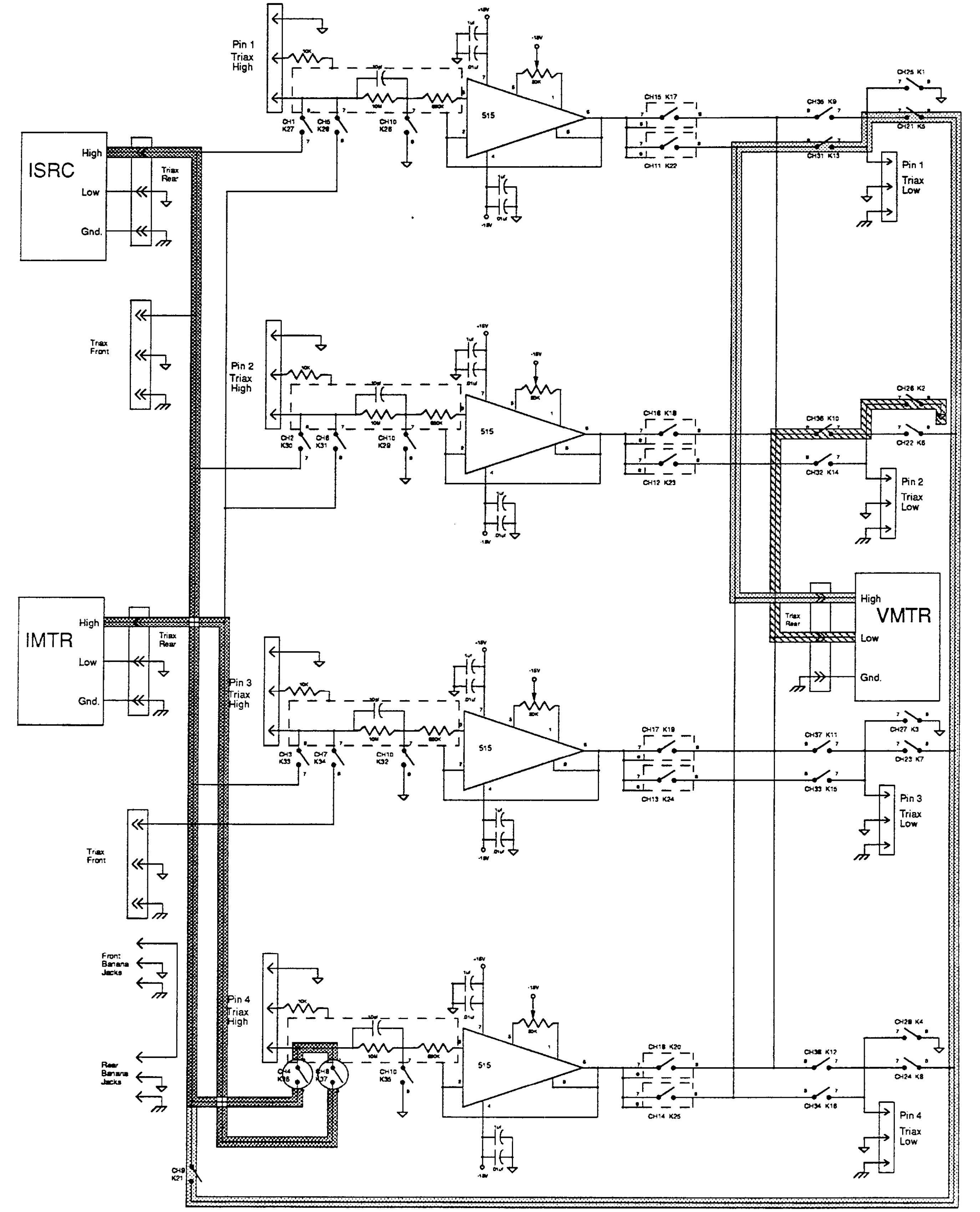


Relay Under Test | Current Soure Path | Voltmeter High Connection | Voltmeter Low Connection

Relays Being Tested = CH2,K30 CH6,K31

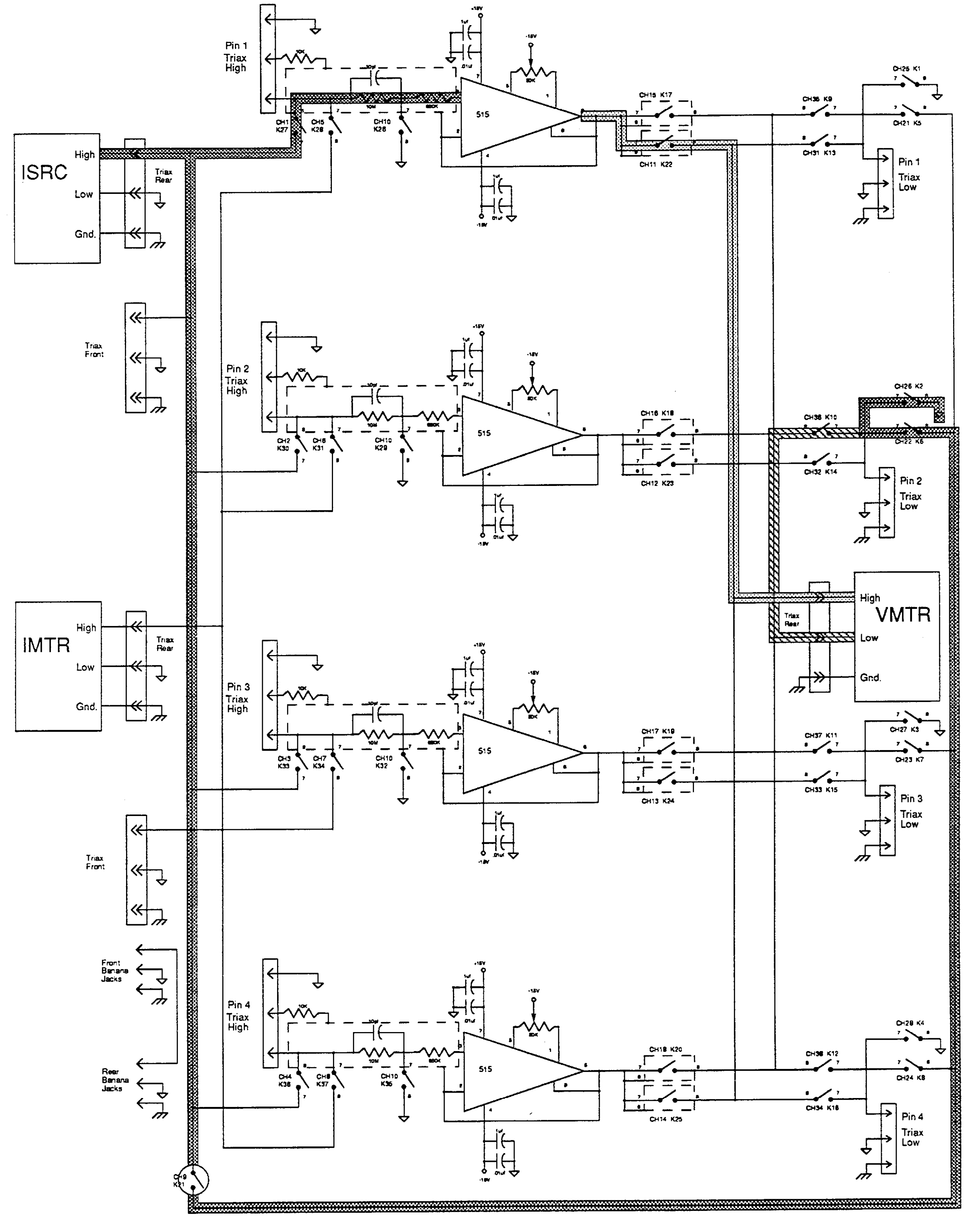


Relays Being Tested = CH3,K33 CH7,K34



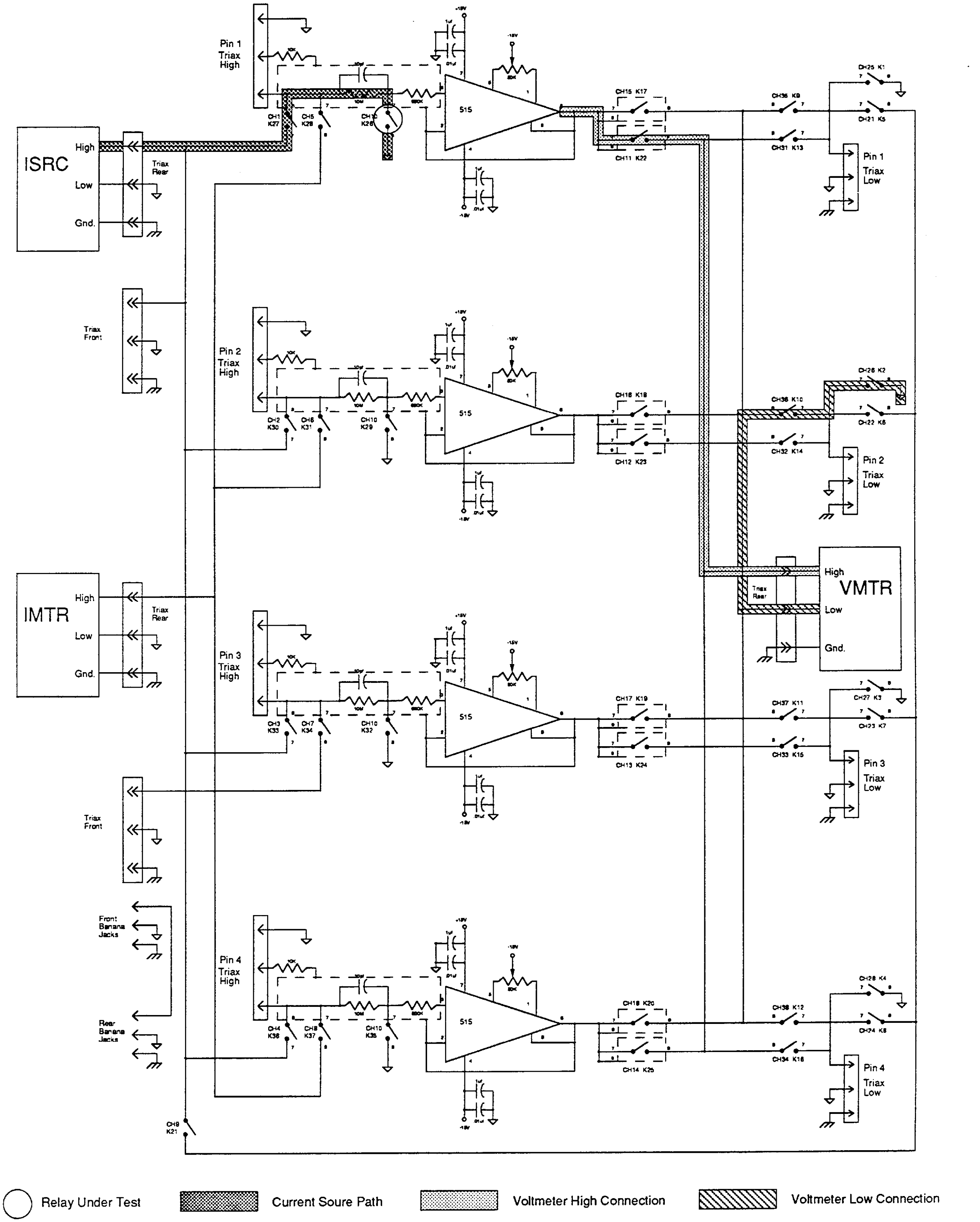
Relay Under Test | Current Soure Path | Voltmeter High Connection | Voltmeter Low Connection

Relays Being Tested = CH4,K36 CH8,K37



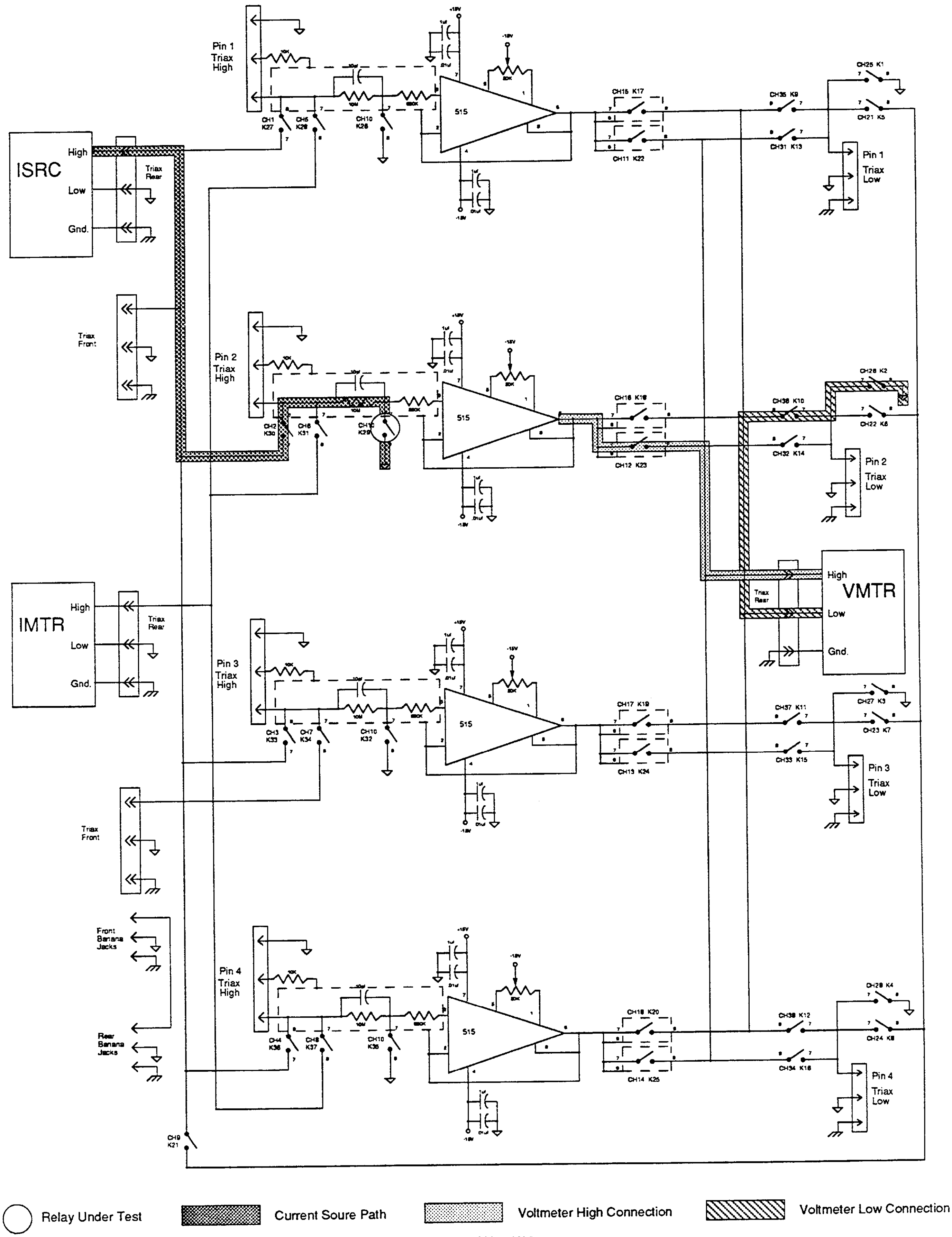
Relay Under Test   Current Soure Path   Voltmeter High Connection   Voltmeter Low Connection

Relay Being Tested = CH9,K21

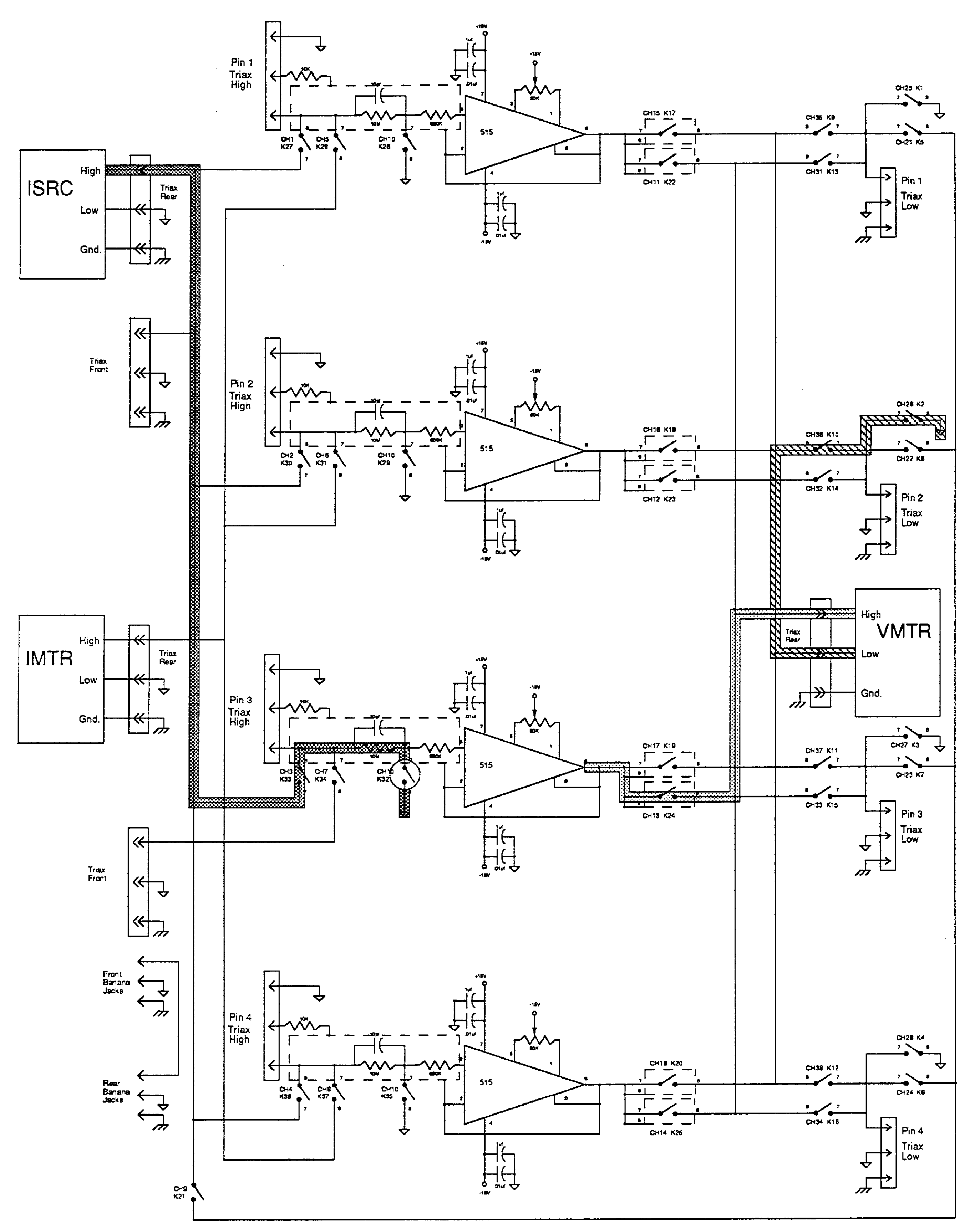


Relay Under Test   Current Soure Path   Voltmeter High Connection   Voltmeter Low Connection

Relay Being Tested = CH10,K26

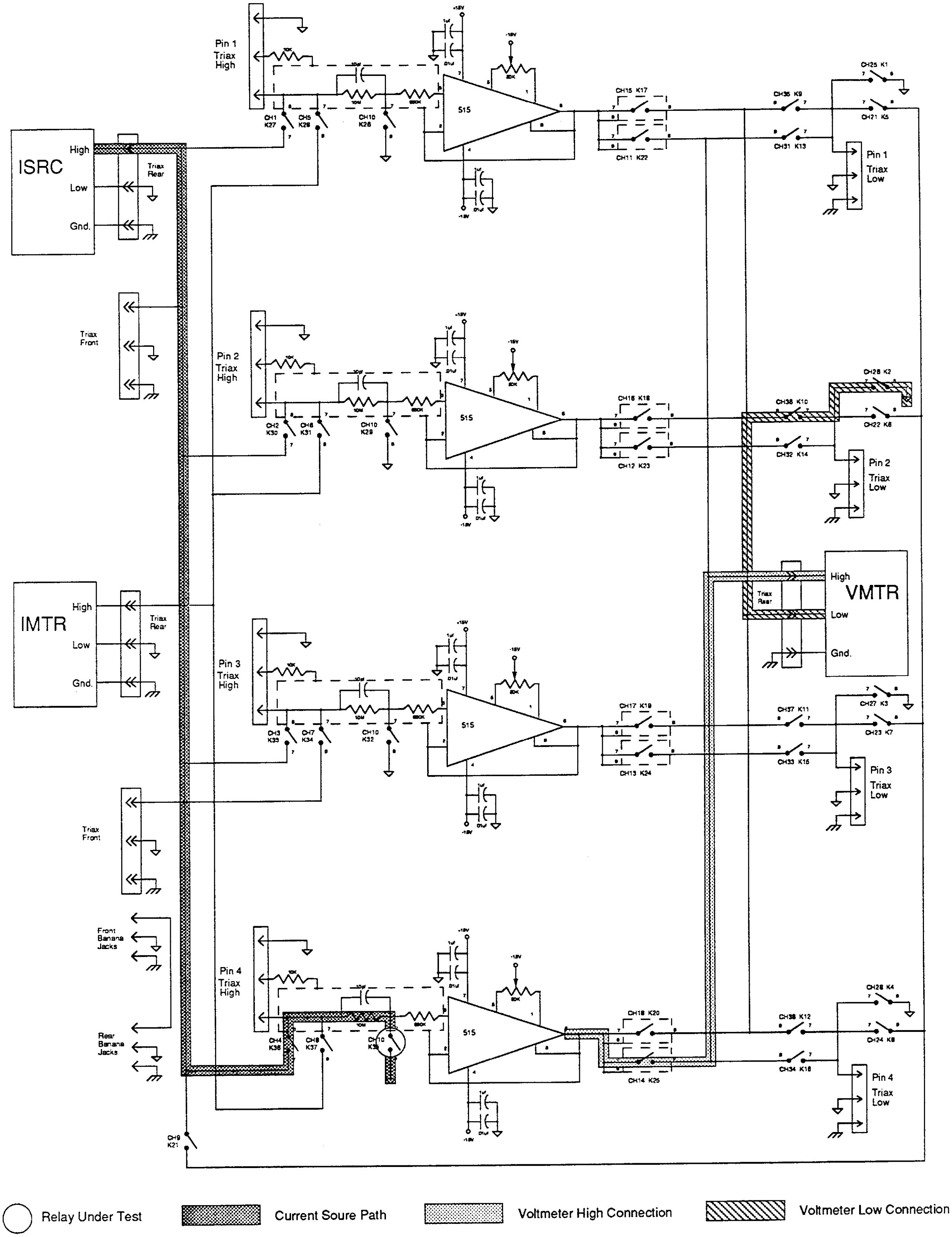


Relay Being Tested = CH10,K29

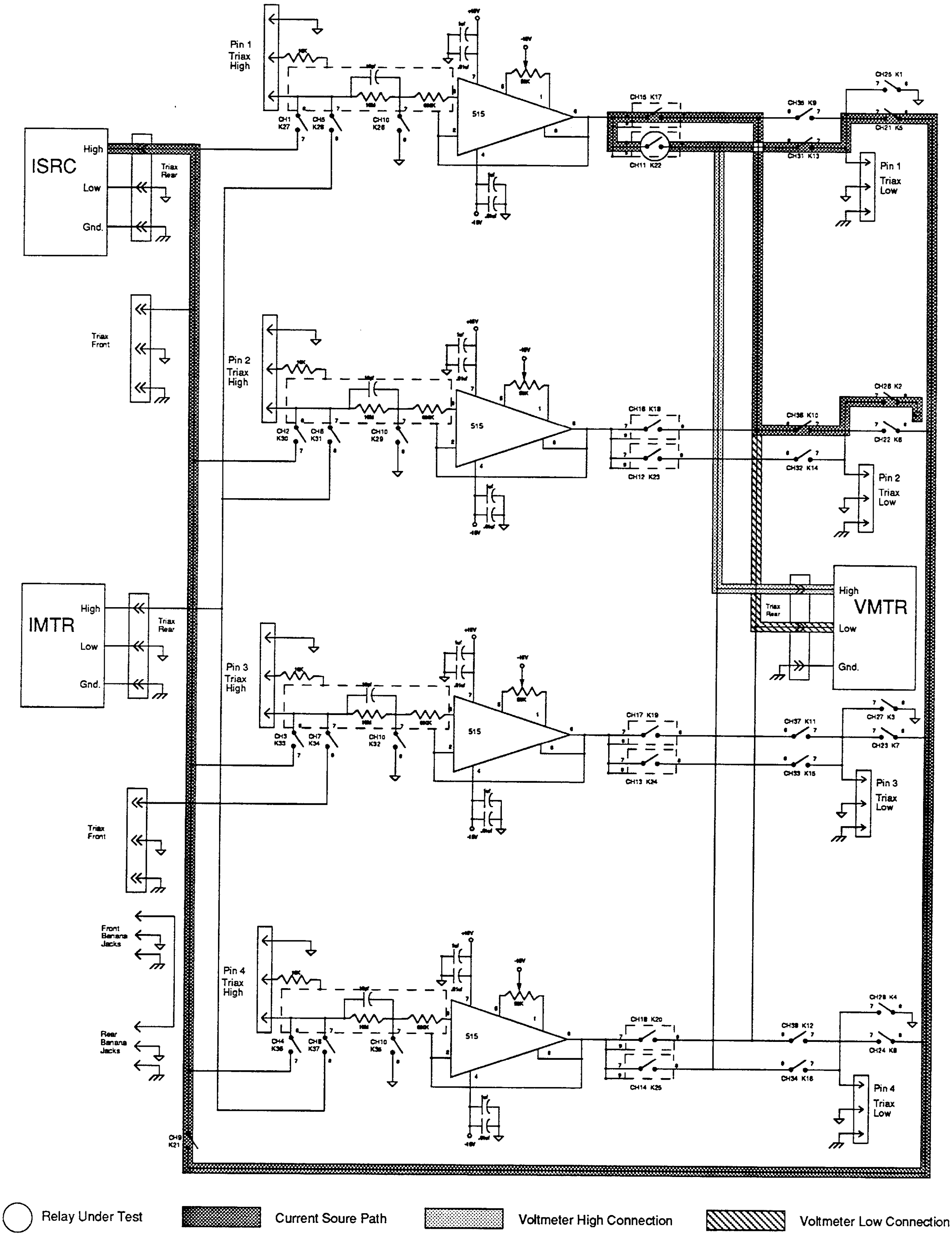


Relay Under Test | Current Soure Path | Voltmeter High Connection | Voltmeter Low Connection

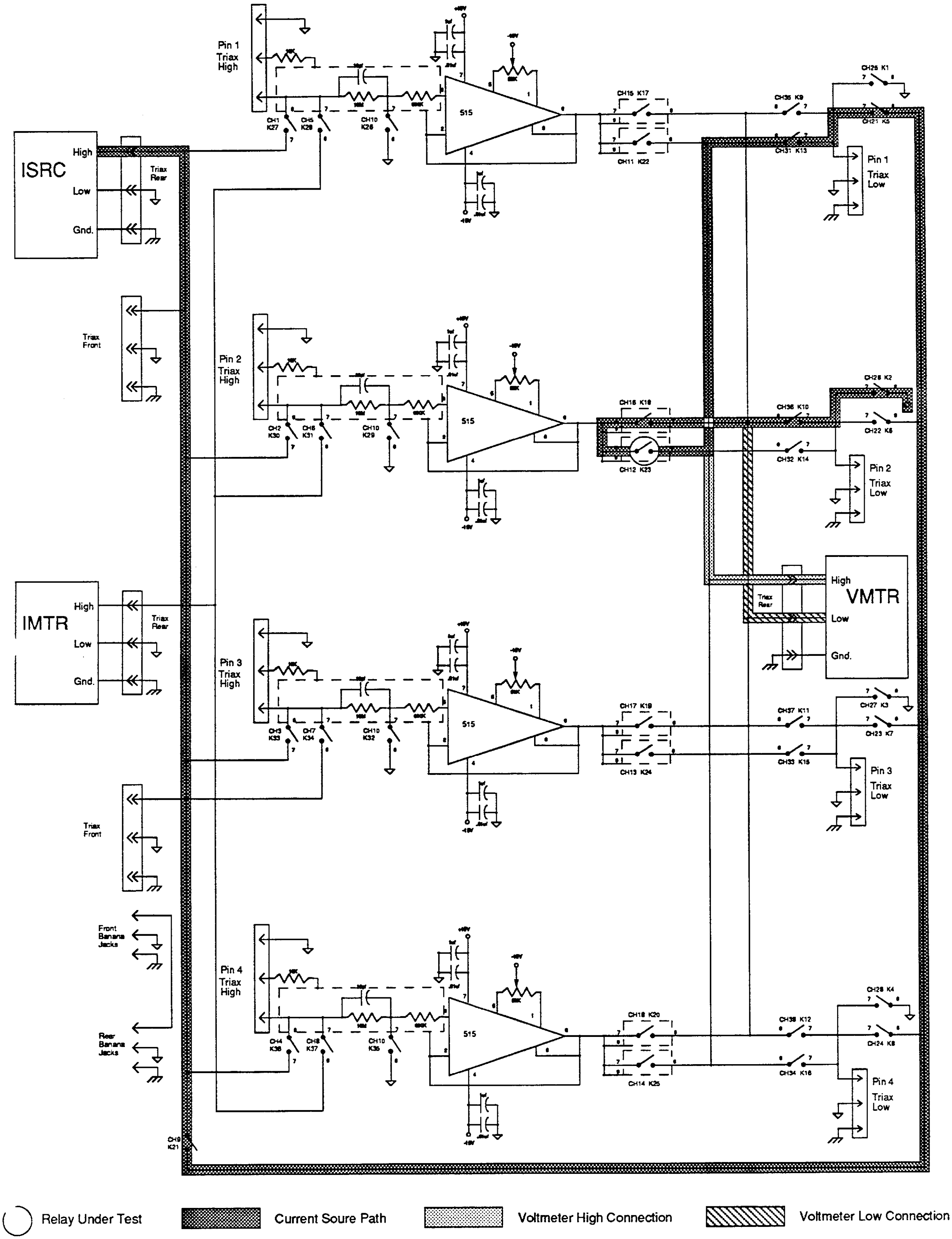
Relay Being Tested = CH10,K32



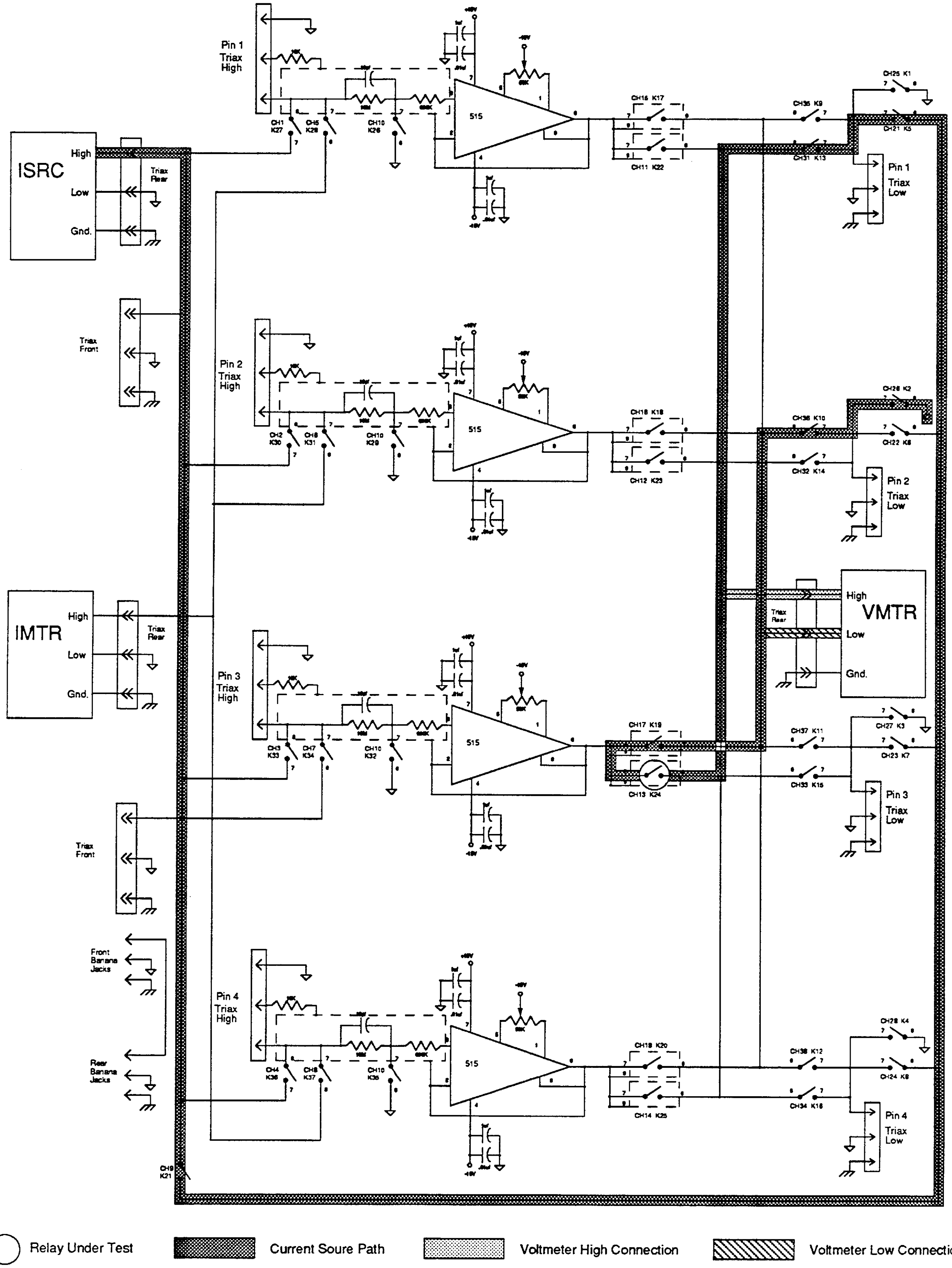
Relay Being Tested = CH10,K35



Relay Being Tested = CH11,K22

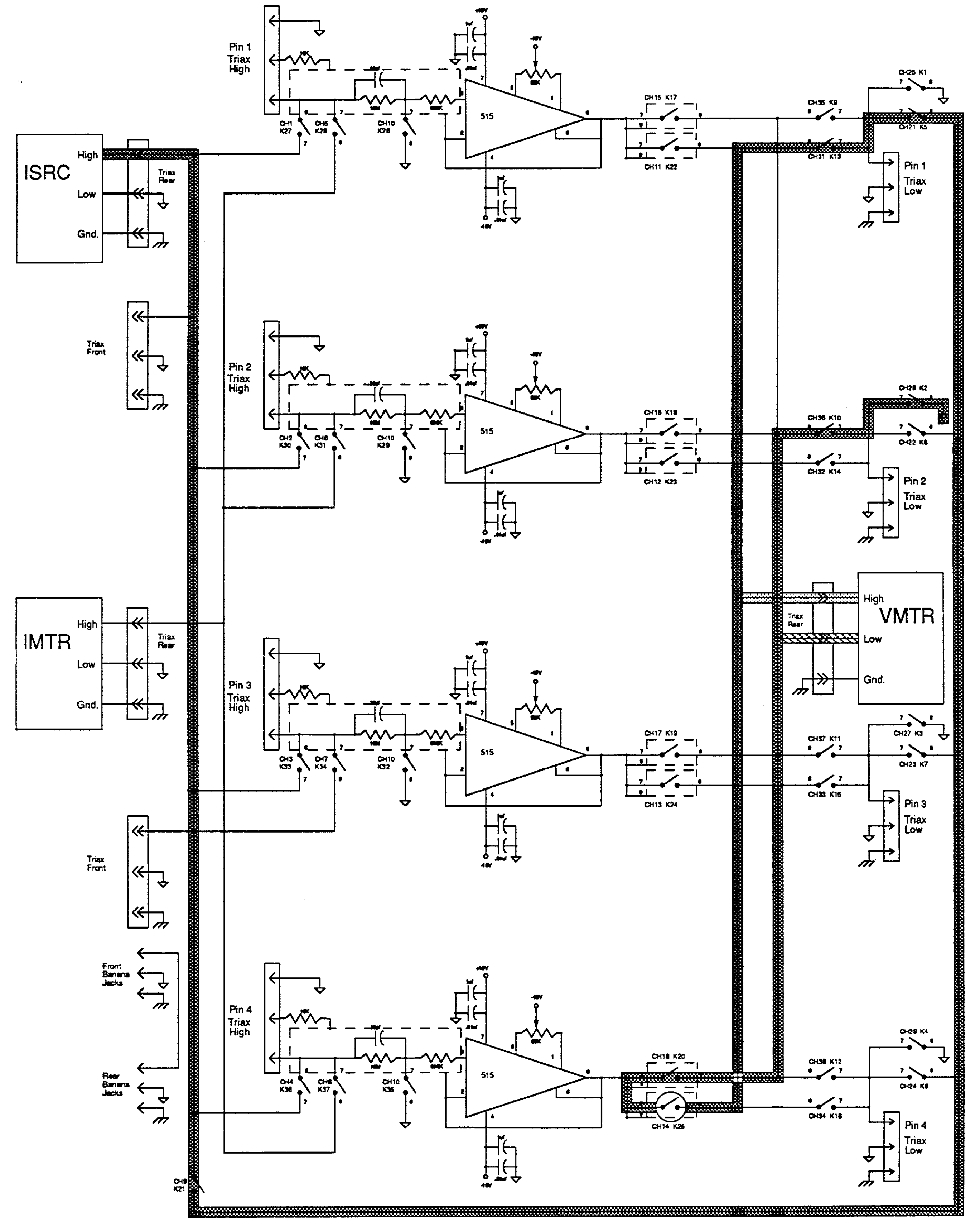


Relay Being Tested = CH12,K23



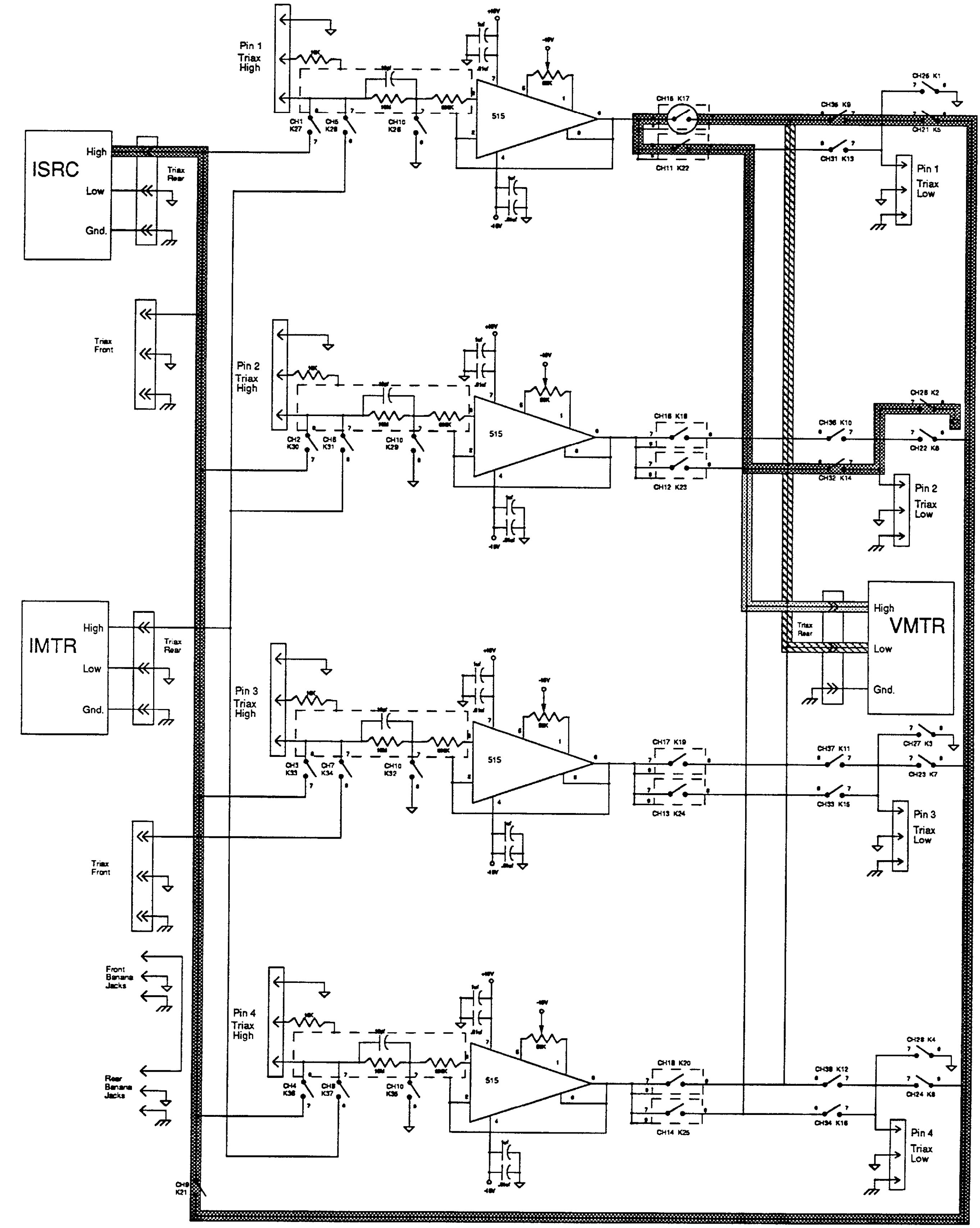
Relay Under Test | Current Soure Path | Voltmeter High Connection | Voltmeter Low Connection

Relay Being Tested = CH13,K24



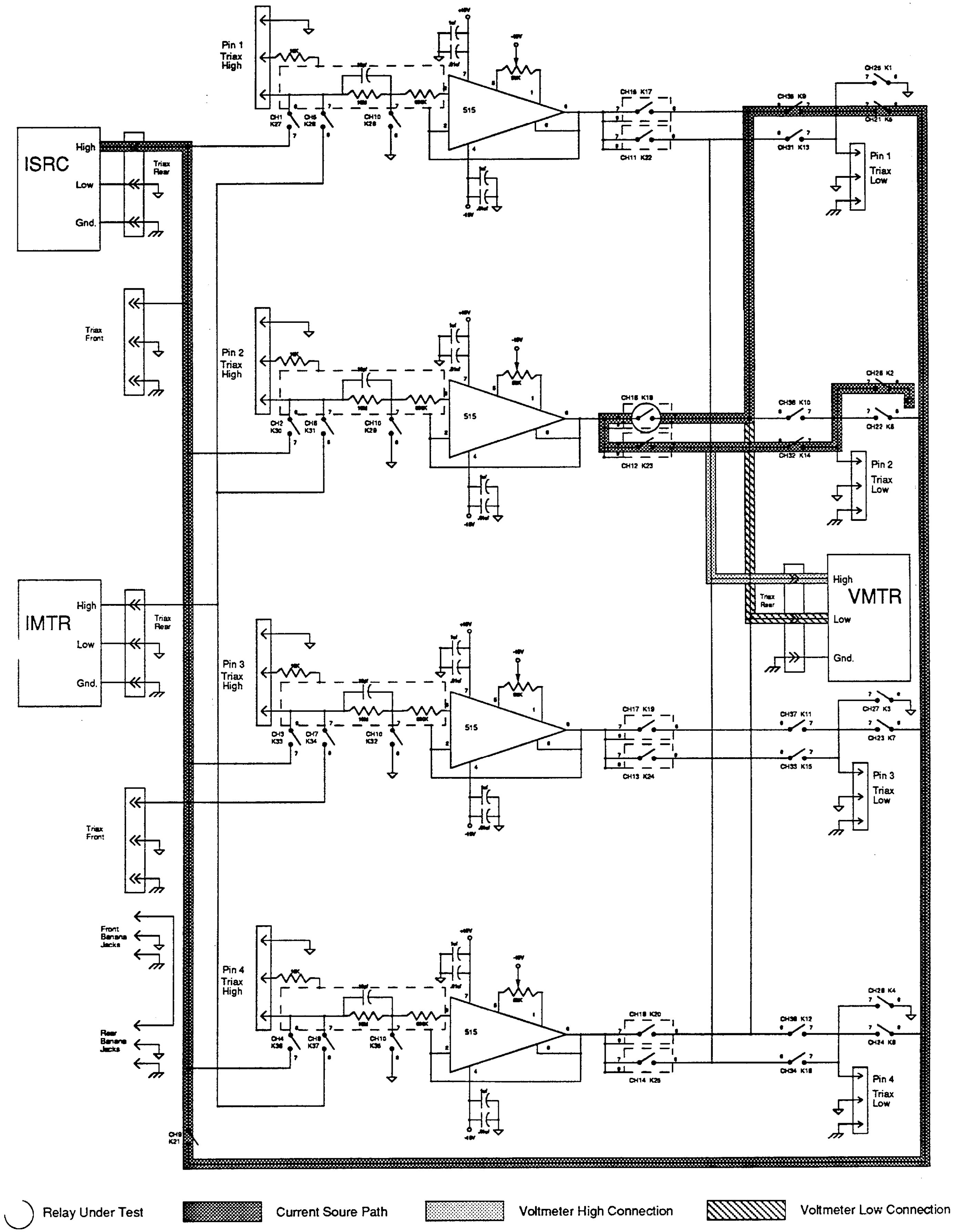
Relay Under Test | Current Soure Path | Voltmeter High Connection | Voltmeter Low Connection

Relay Being Tested = CH14,K25

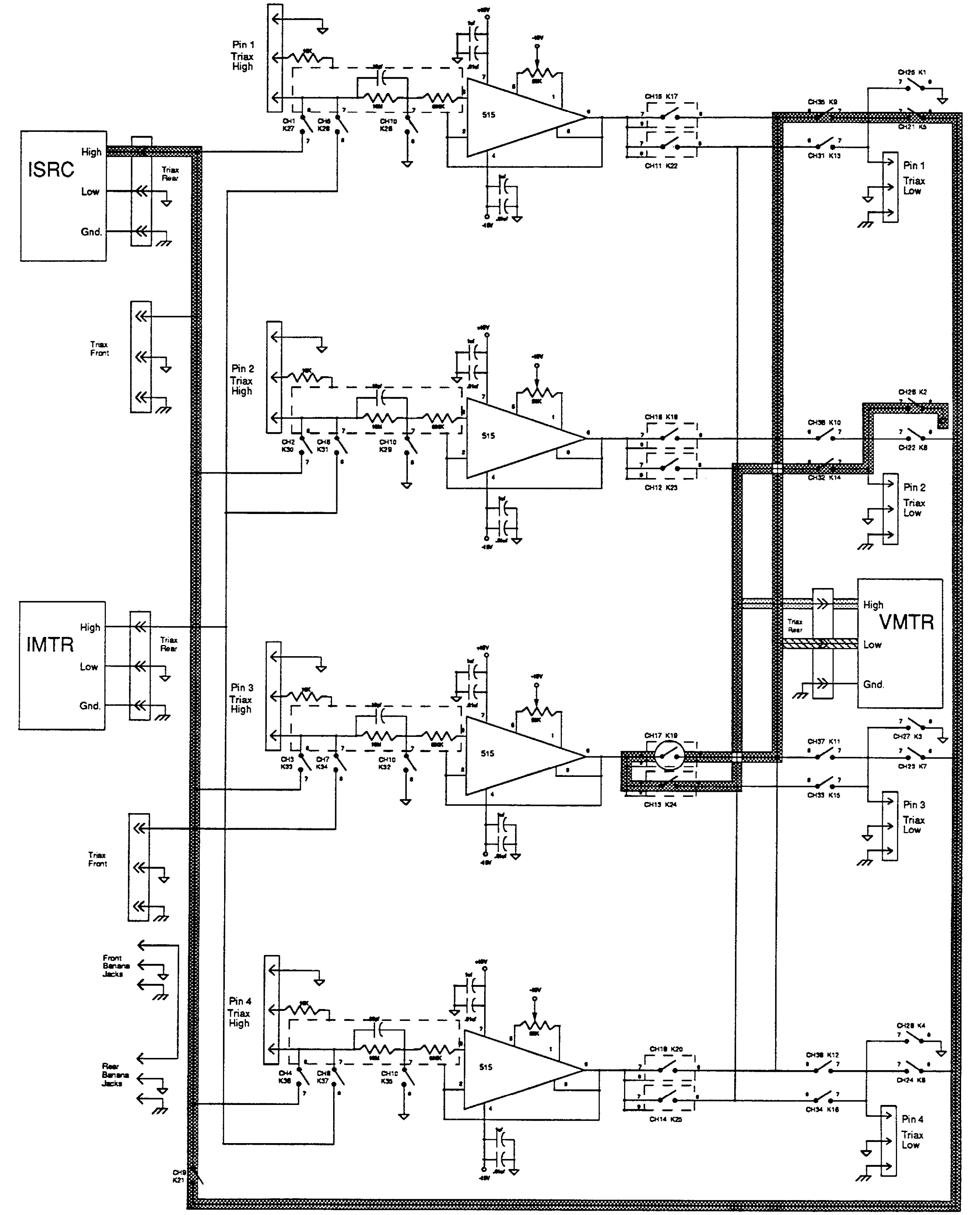


Relay Under Test | Current Soure Path | Voltmeter High Connection | Voltmeter Low Connection

Relay Being Tested = CH15,K17

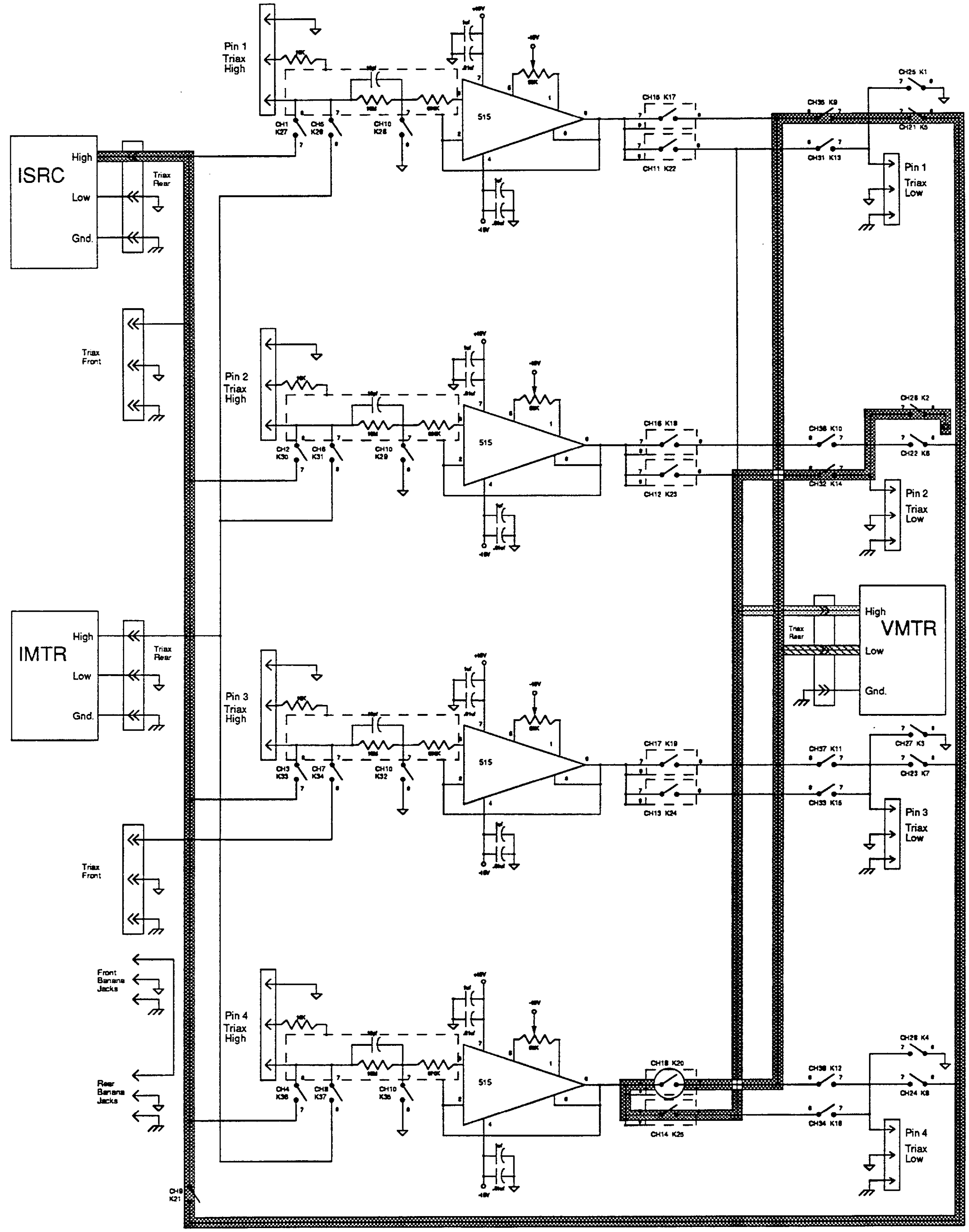


Relay Being Tested = CH16,K18



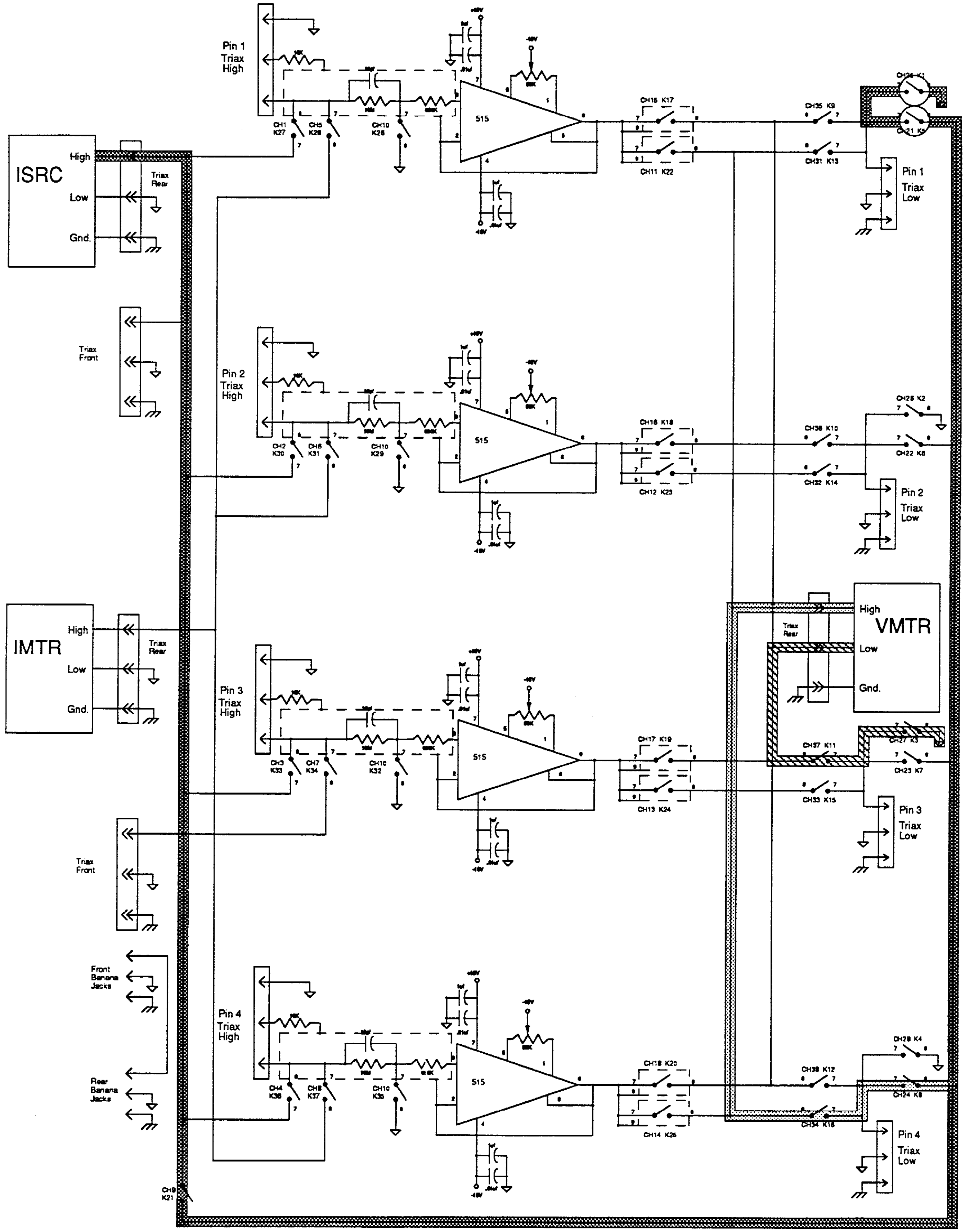
Relay Under Test | Current Soure Path | Voltmeter High Connection | Voltmeter Low Connection

Relay Being Tested = CH17,K19



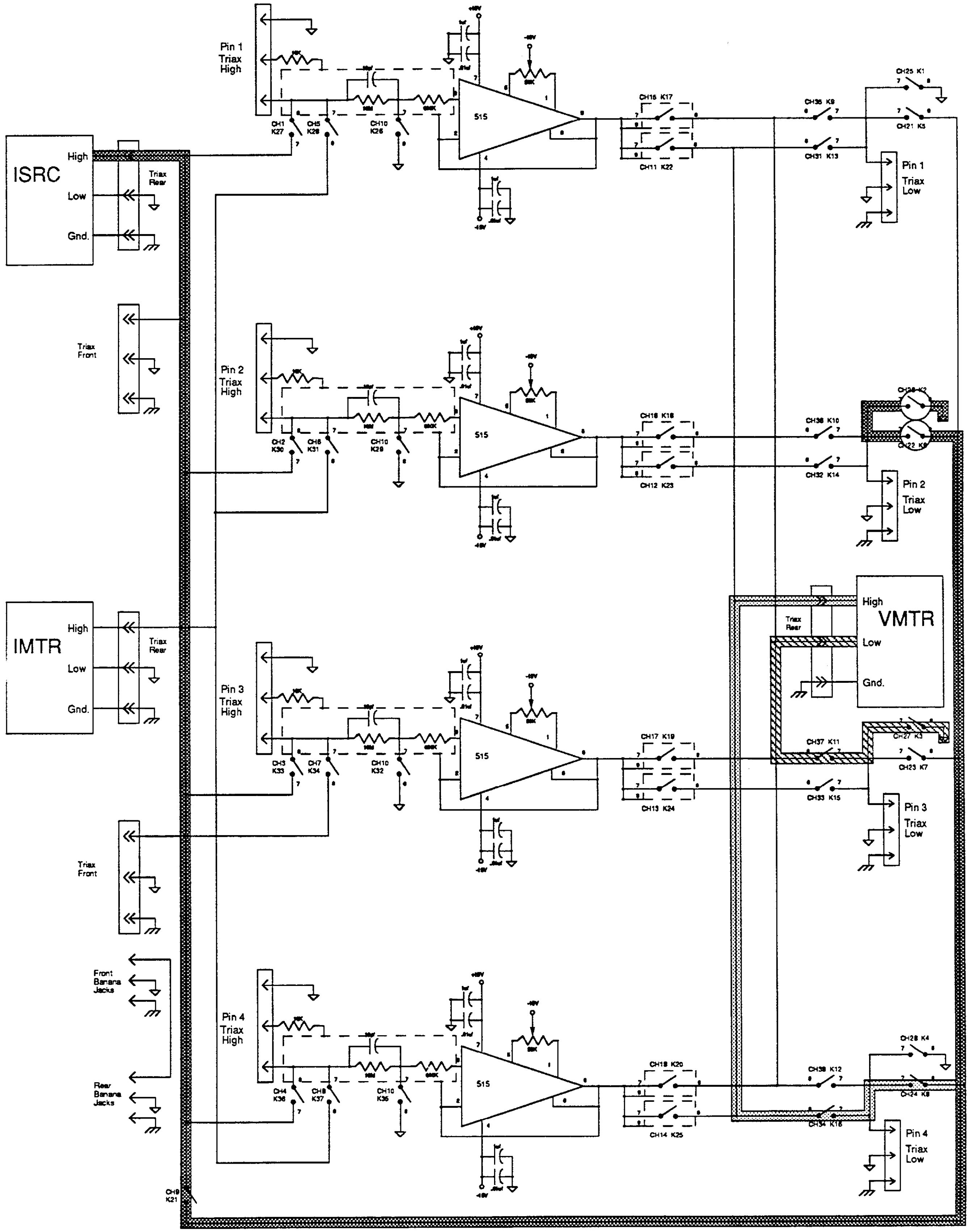
Relay Under Test | Current Soure Path | Voltmeter High Connection | Voltmeter Low Connection

Relay Being Tested = CH18,K20



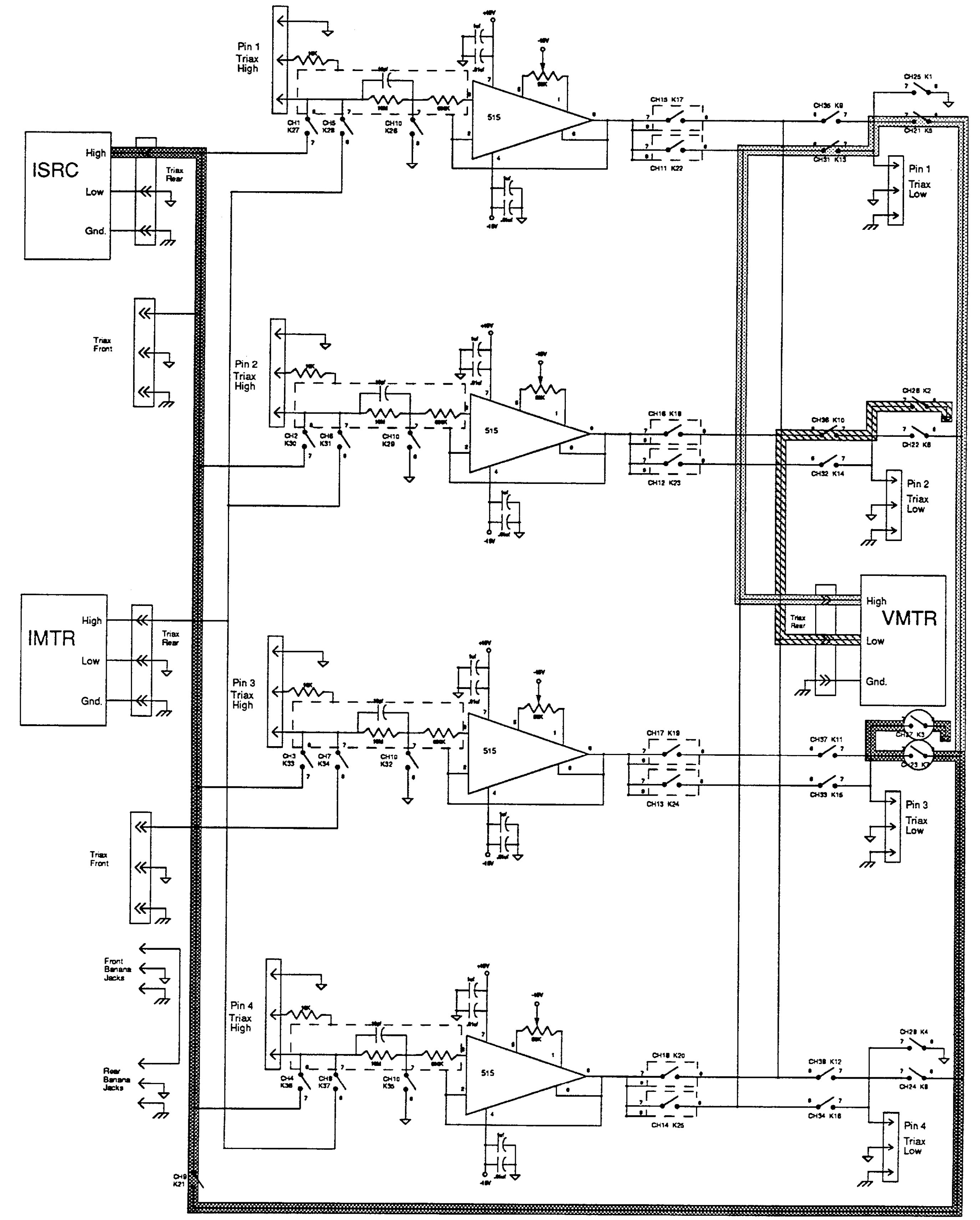
Relay Under Test | Current Soure Path | Voltmeter High Connection | Voltmeter Low Connection

Relays Being Tested = CH21,K5 CH25,K1



Relay Under Test   Current Soure Path   Voltmeter High Connection   Voltmeter Low Connection

Relays Being Tested = CH22,K6 CH26,K2



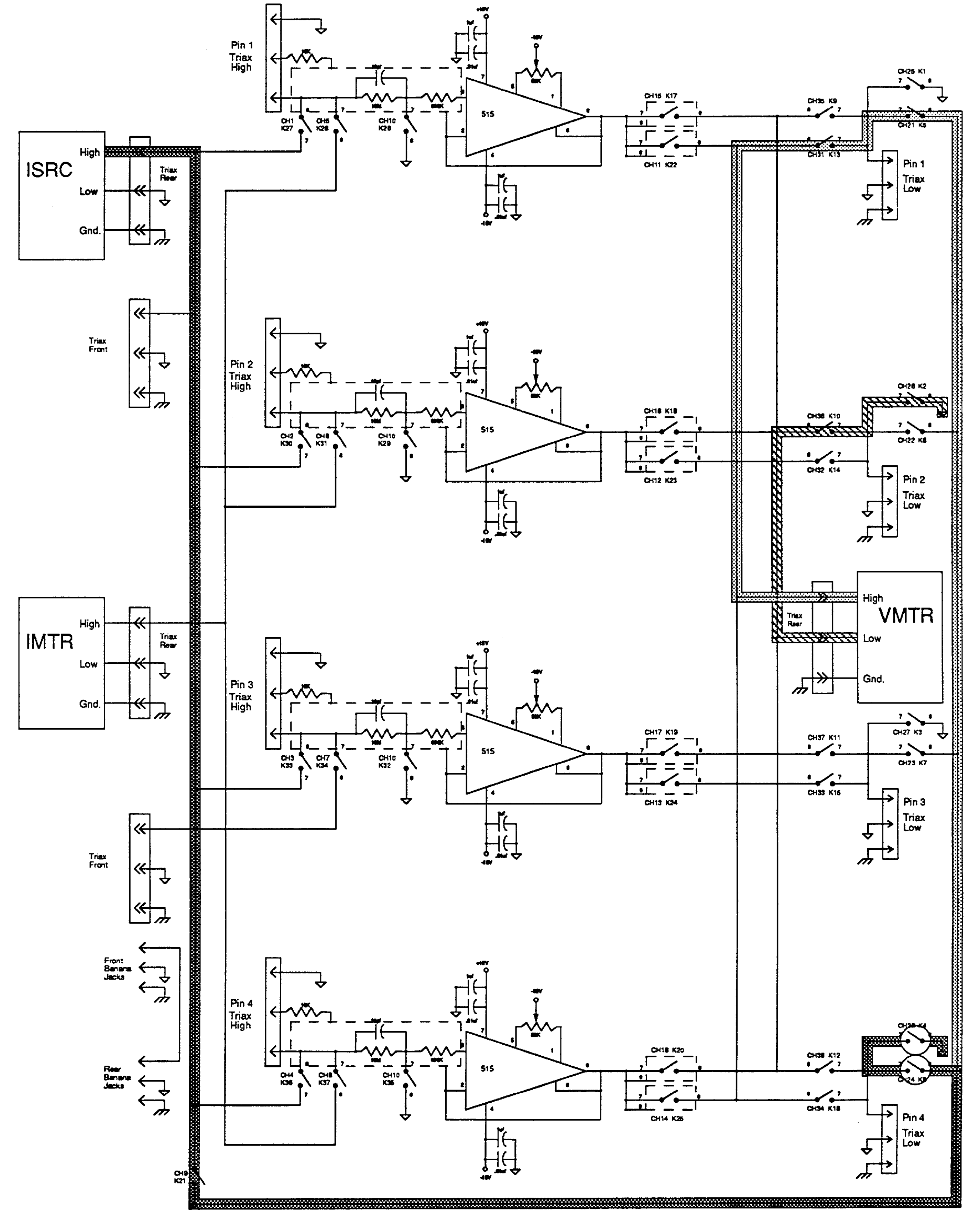
Relay Under Test

Current Soure Path

Voltmeter High Connection

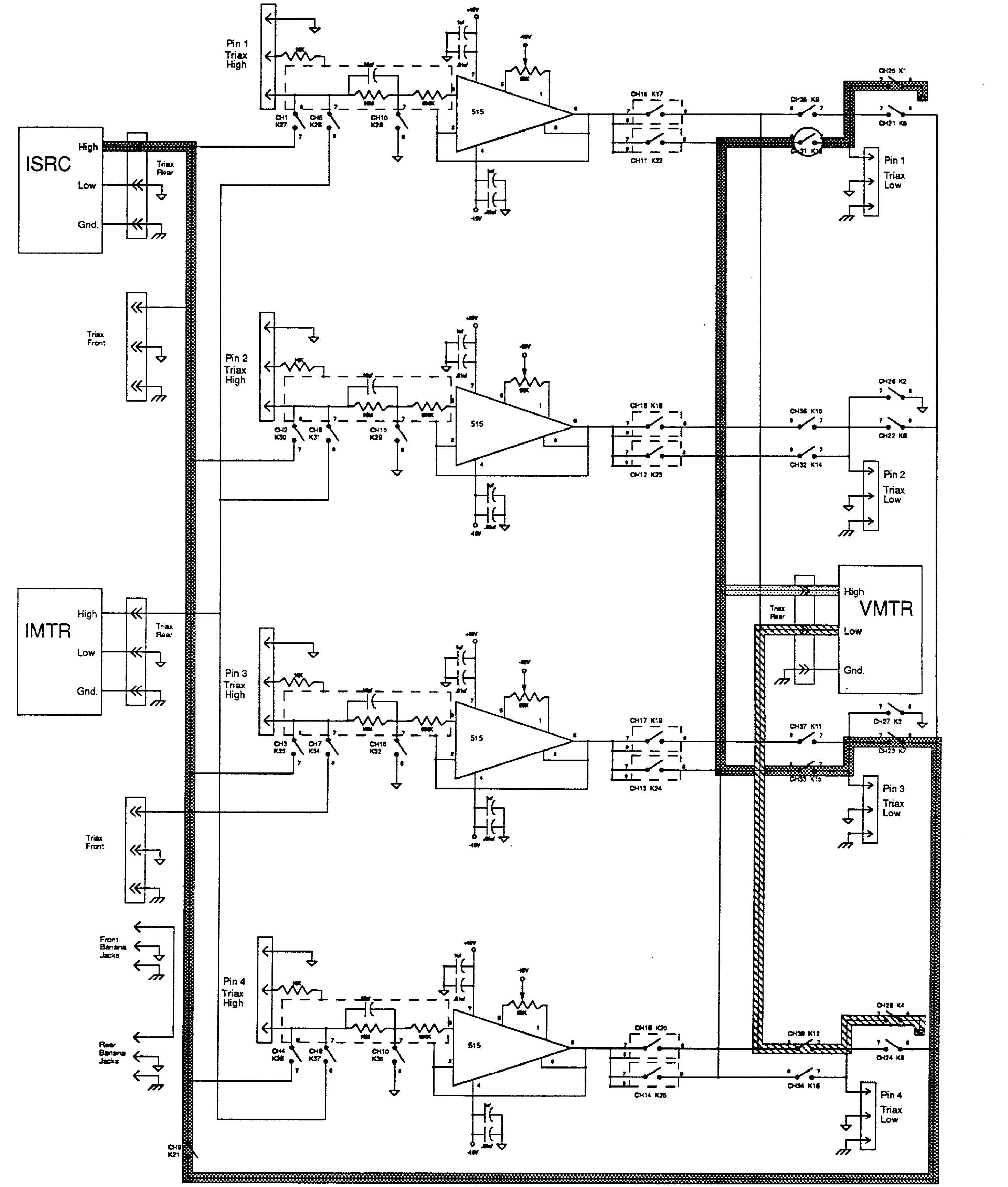
Voltmeter Low Connection

Relays Being Tested = CH23,K7 CH27,K3



Relay Under Test   Current Soure Path   Voltmeter High Connection   Voltmeter Low Connection

Relays Being Tested = CH24,K8 CH28,K4



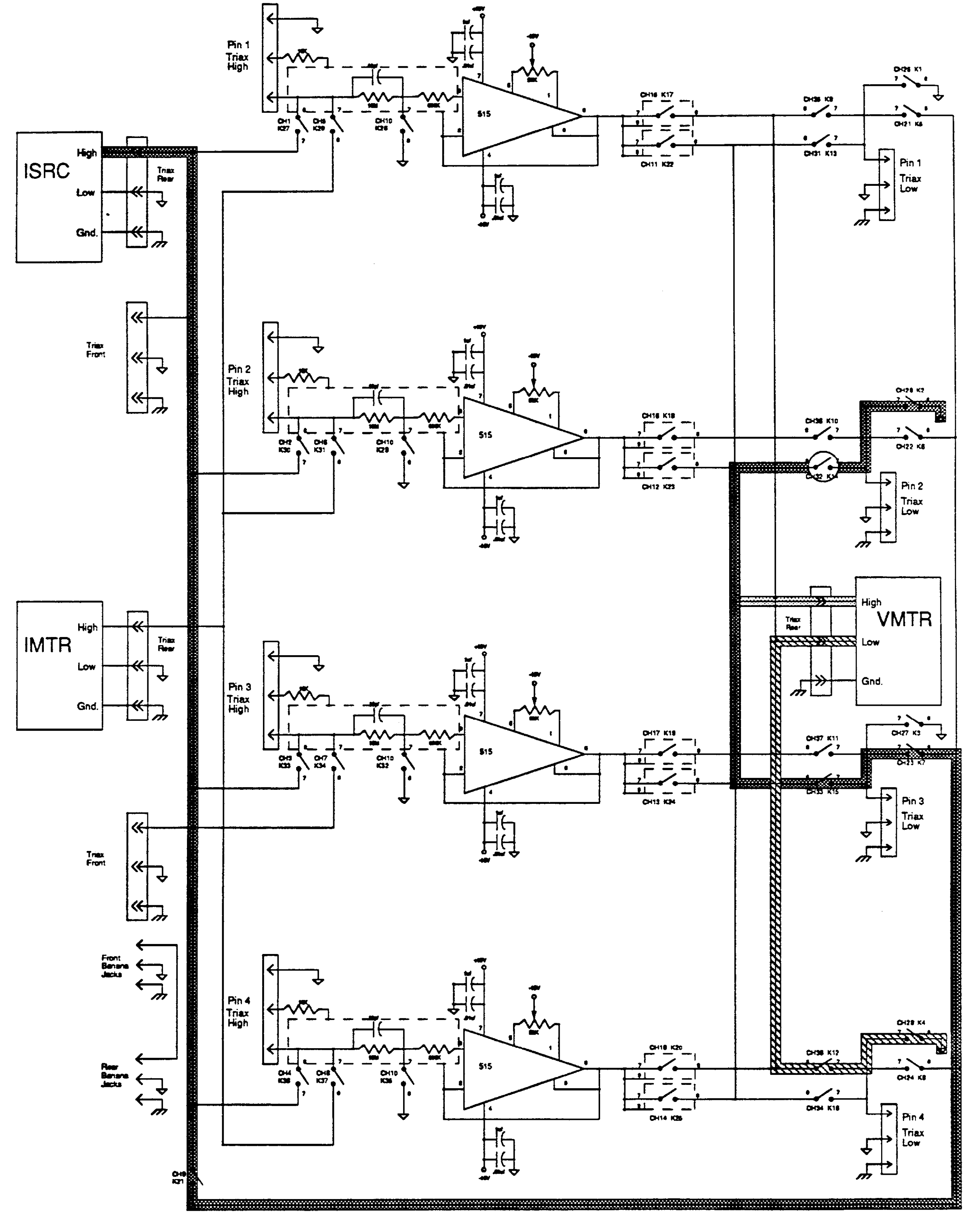
Relay Under Test

Current Soure Path

Voltmeter High Connection

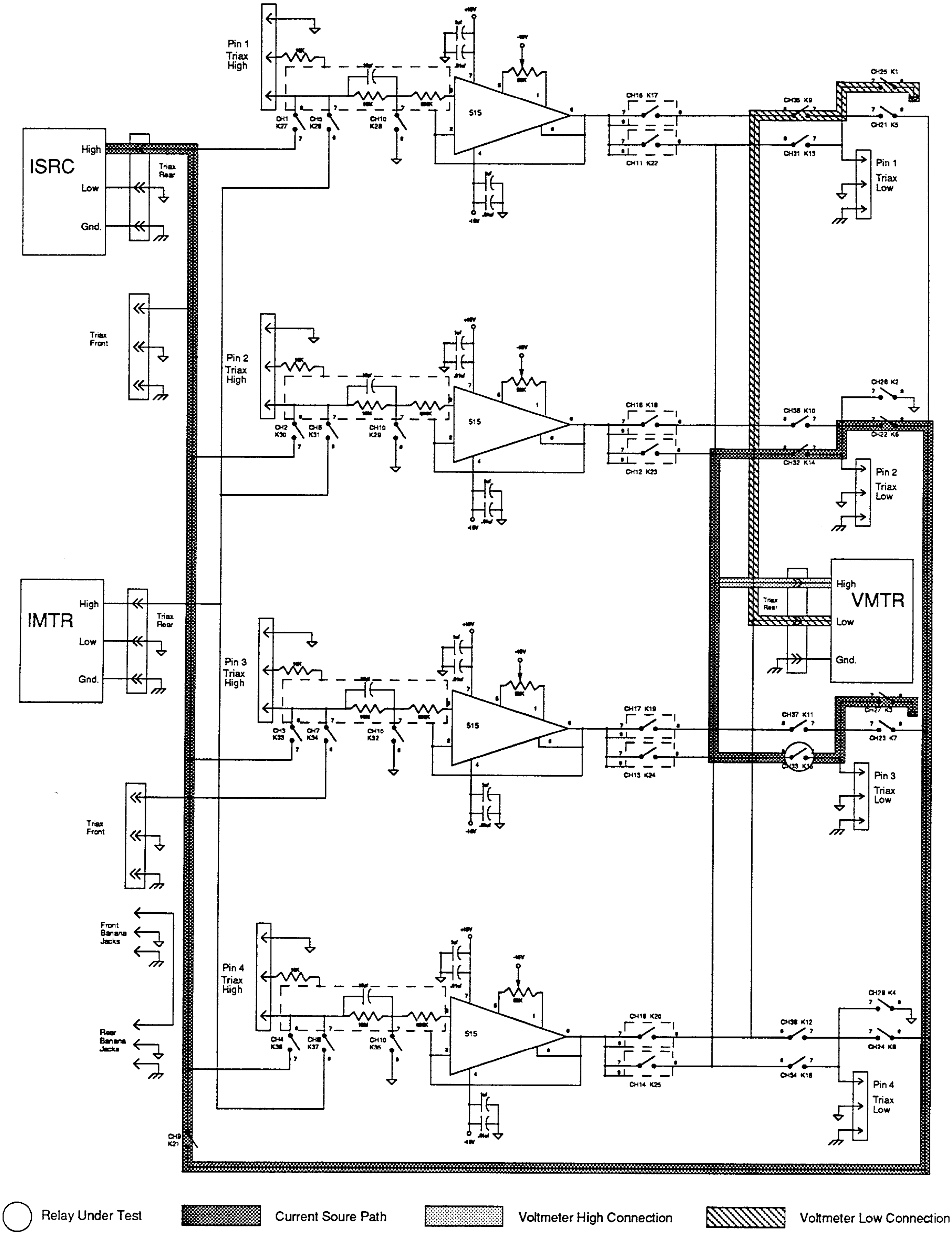
Voltmeter Low Connection

Relays Being Tested = CH31,K13

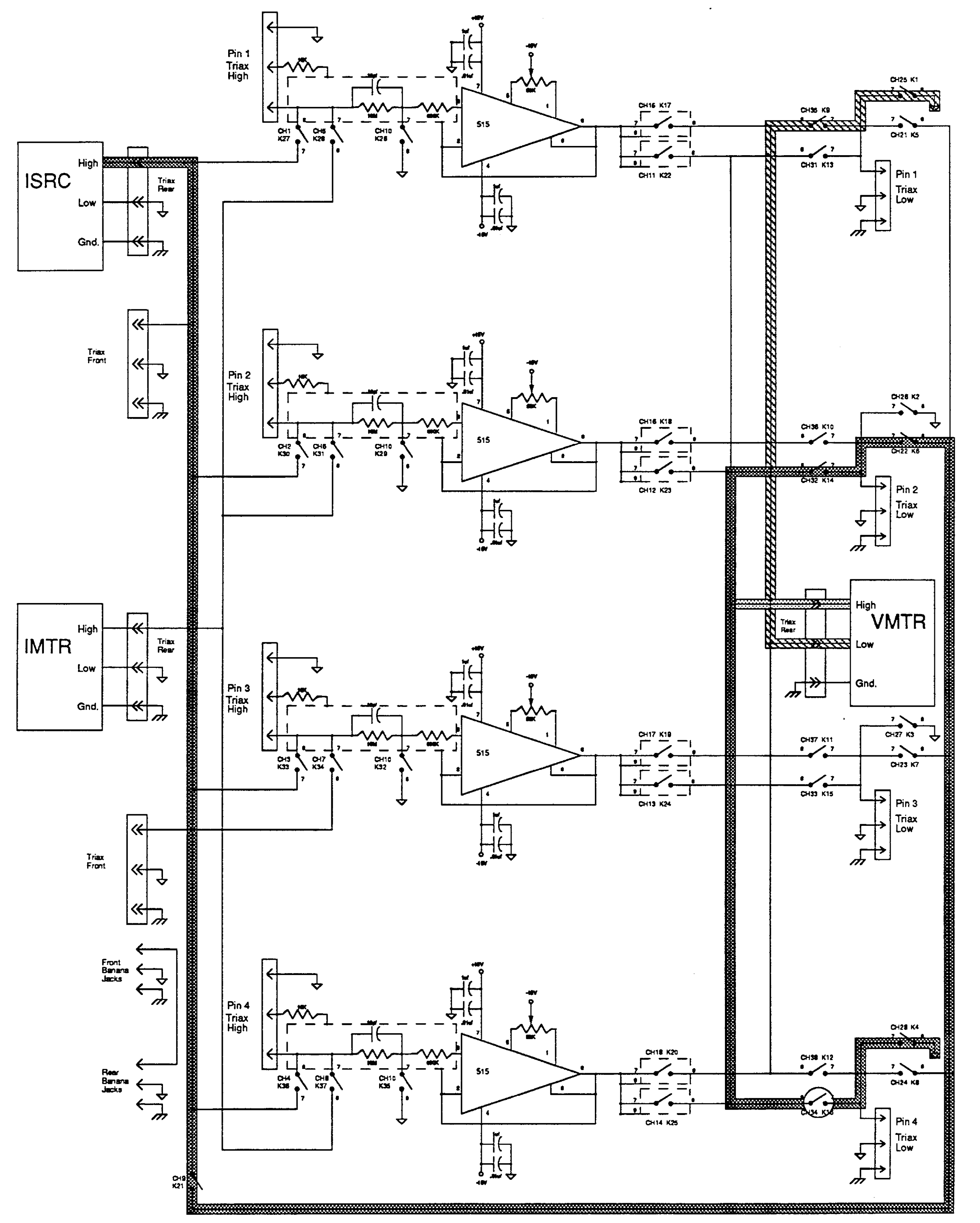


Relay Under Test Current Soure Path Voltmeter High Connection Voltmeter Low Connection

Relays Being Tested = CH32,K14



Relays Being Tested = CH33,K15



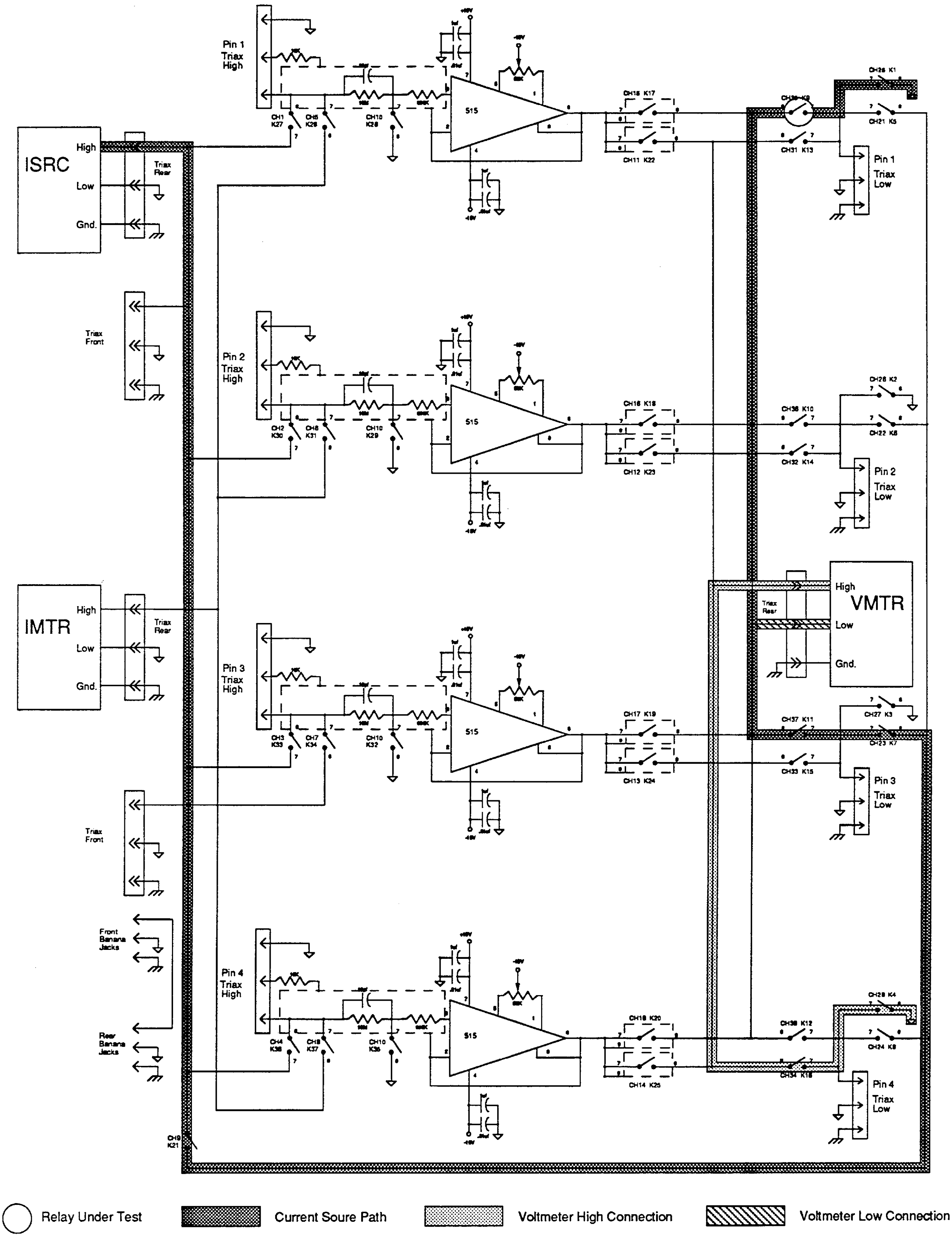
Relay Under Test

Current Soure Path

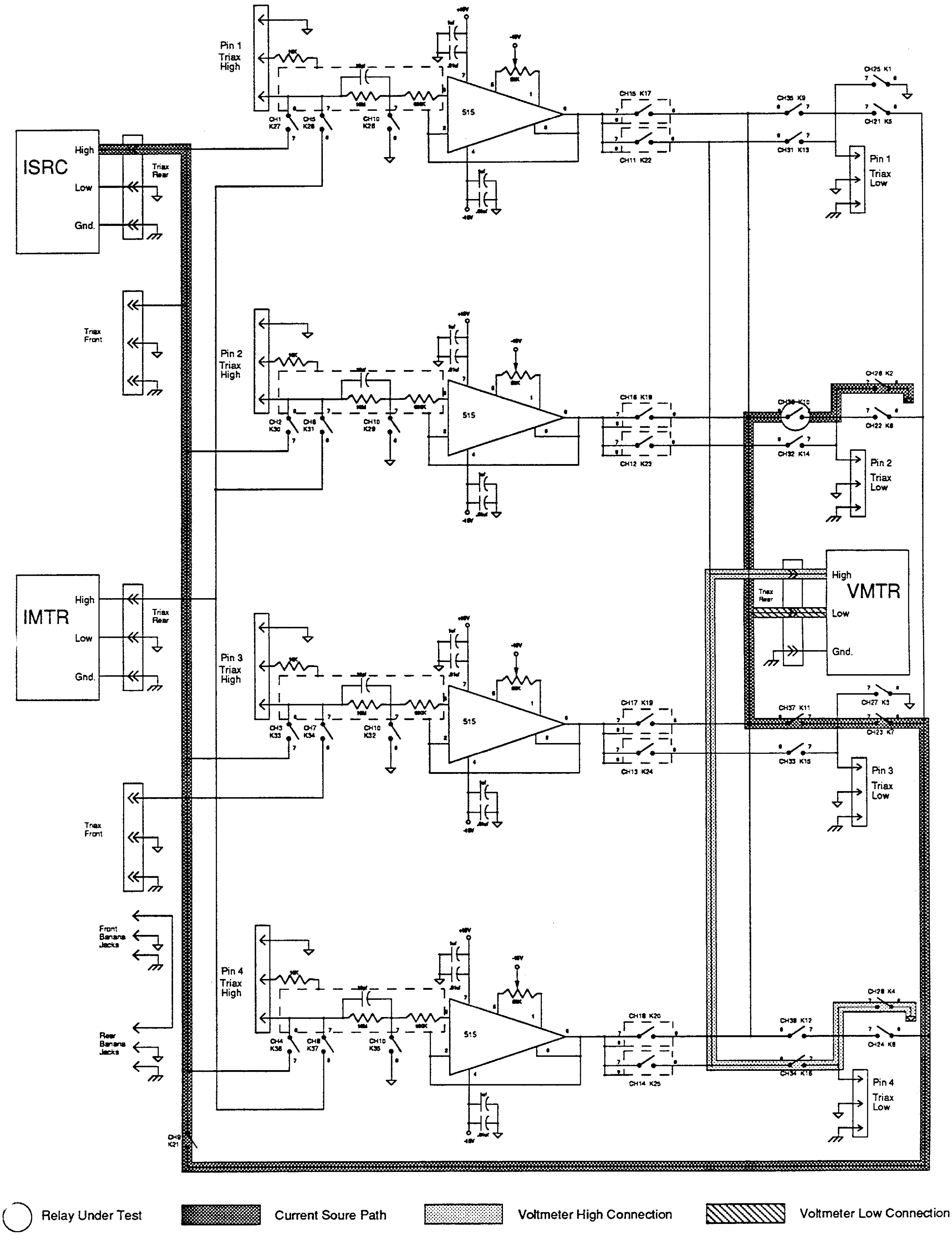
Voltmeter High Connection

Voltmeter Low Connection

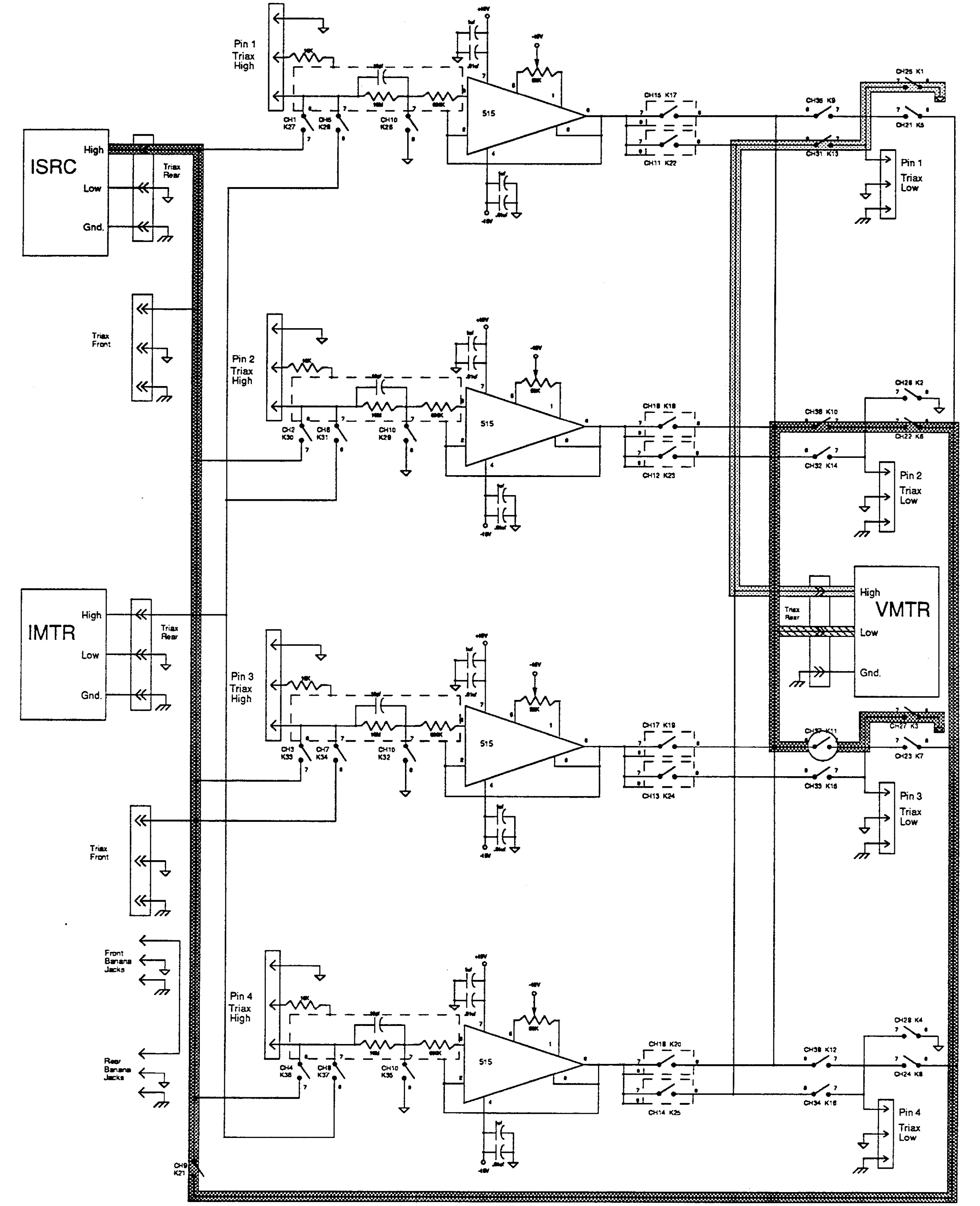
Relays Being Tested = CH34,K16



Relays Being Tested = CH35,K9

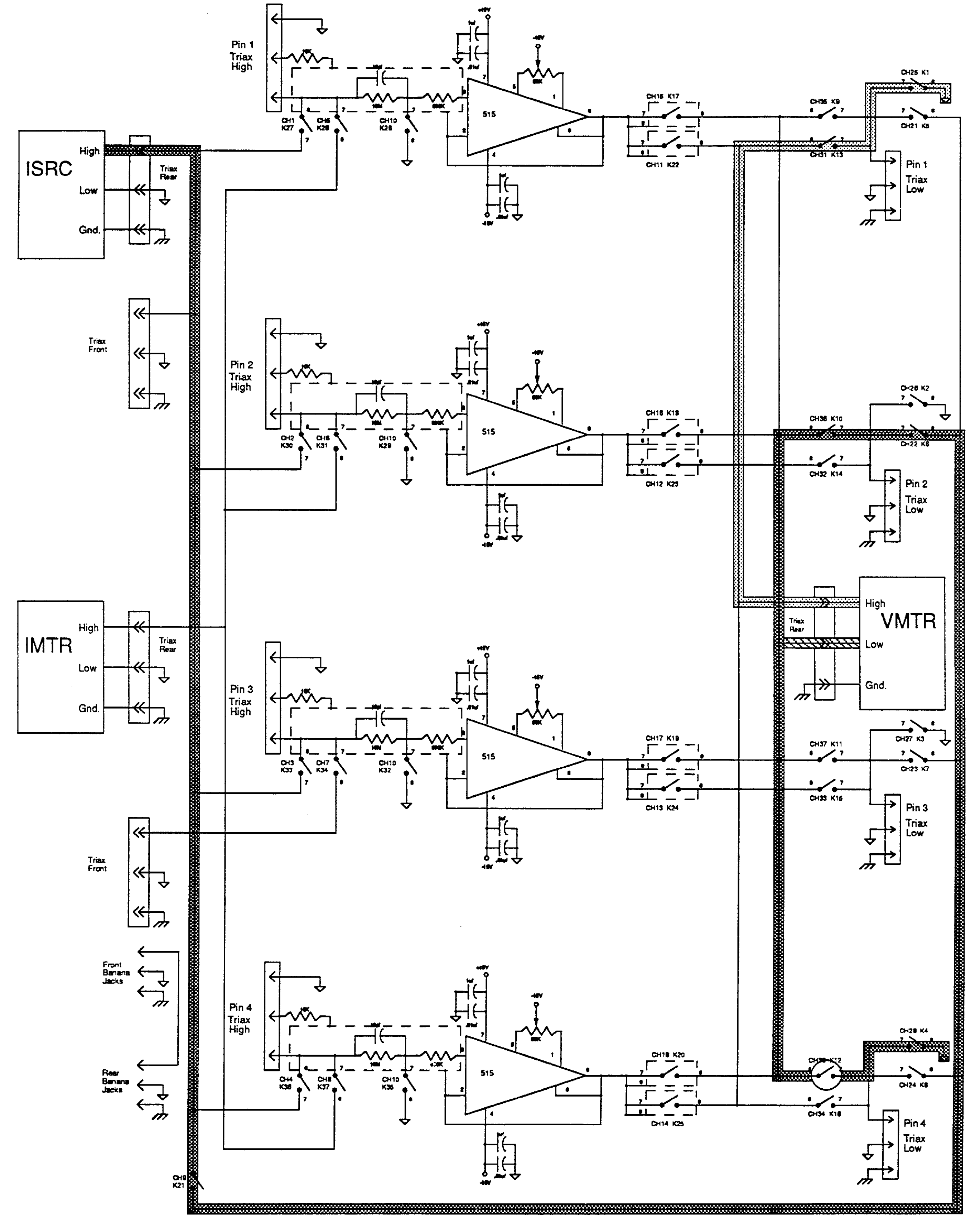


Relays Being Tested = CH36,K10



Relay Under Test | Current Soure Path | Voltmeter High Connection | Voltmeter Low Connection

Relays Being Tested = CH37,K11



Relay Under Test | Current Soure Path | Voltmeter High Connection | Voltmeter Low Connection

Relays Being Tested = CH38,K12

This Page Left Blank Intentionally

APPENDIX C

PROGRAM LISTINGS

C.1 VDPHAL.BAS Program Listing

```
'                          Van der Pauw Program
'   Copyright (C) 1990 by Keithley Instruments, Inc.
'   Cleveland, Ohio
'
'   This software is furnished under a license and may be used and copied
'   only in accordance with the terms of such license, and with the inclusion
'   of the above COPYRIGHT notice.  This software or any other copies thereof
'   may not be provided or otherwise made available to any other person. No
'   title to and ownership of the software is hereby transferred.
'
'   The information in this software is subject to change without notice, and
'   should not be construed as a commitment by KEITHLEY INSTRUMENTS, INC.
'
'   KEITHLEY assumes no responsibility for the use or reliability of its
'   software on equipment which is not supplied by KEITHLEY.
'****************************************************************************
'
'   History
'
'       The purpose of this software is to calculate resistivity and Hall calculations
'
'       V1.0    /8/10/90                 Brian Polaski
'****************************************************************************

' $INCLUDE: 'c:\ps\linkscrn.dim'
' $INCLUDE: 'c:\s110\lib\S110.QB'
DIM I AS INTEGER
DIM ID AS STRING
fail.flag% = 0' this is a flag to detect if an invalid reading was made
'-------
'THIS SETS ALL THE CURRENTS AND VOLTAGES TO 1E+22 FOR ERROR CHECKING
'-------
V1234 = 1E+22
V2134 = 1E+22
V2341 = 1E+22
V3241 = 1E+22
V3412 = 1E+22
V4312 = 1E+22
V4123 = 1E+22
V1423 = 1E+22
A1234 = 1E+22
A2134 = 1E+22
A2341 = 1E+22
A3241 = 1E+22
A3412 = 1E+22
A4312 = 1E+22
A4123 = 1E+22
A1423 = 1E+22
VP1342! = 1E+22
VP3142! = 1E+22
VP2413! = 1E+22
VP4213! = 1E+22
AP1342! = 1E+22
AP3142! = 1E+22
AP2413! = 1E+22
AP4213! = 1E+22
VN1342! = 1E+22
VN3142! = 1E+22
VN2413! = 1E+22
VN4213! = 1E+22
AN1342! = 1E+22
AN3142! = 1E+22
```

```
AN2413! = 1E+22
AN4213! = 1E+22
RHOAV! = 1E+22
RHAV! = 1E+22
MOB! = 1E+22
N! = 1E+22
'-------
'THIS IS THE FIRST MENU
'-------
start: CALL getkey(NAME$, ID$, HL$, TEMP!, THICK!, MG!, AMPS!, DLY!, I%)
'-------
'SETS TEMP OF THE SAMPLE
'-------
CALL tstsel(1)    'must have this command first
IF TEMP! <> 0 THEN CALL restemp   'COMMENT OUT IF NO K20 CRYO
IF (TEMP.TYPE <> 0) THEN
     IF TEMP! <> 0 THEN

            CALL settemp(TEMP!, gettemp!)           'COMMENT OUT IF NO TEMP CONTROLLER
          IF gettemp < 0 THEN
               CALL restemp    'COMMENT OUT IF NO K20
               GOTO start
          END IF
          TEMP! = gettemp!
     END IF
END IF
'-------
'INITIALIZES THE INSTRUMENTS & SETS FOR HIGH OR LOW TERMINALS
'-------

 IF (HL$ = "H") OR (HL$ = "HIGH") THEN hlsel (HIGH)
 IF (HL$ = "L") OR (HL$ = "LOW") THEN hlsel (LOW)

vlim = 10
CLS
'-------
'MAKES THE Van Der Pauw MEASUREMENTS
'-------

LOCATE 10, 10: PRINT "Doing Calculations on Sample #"; I%
LOCATE 11, 10: PRINT "Please Wait"
IF (HL$ = "h") OR (HL$ = "HIGH") THEN res = vdp!(1, 2, 3, 4, AMPS, DLY!, vlim!, V1234,
V2134, V2341, V3241, V3412, V4312, V4123, V1423, A1234, A2134, A2341, A3241, A3412,
A4312, A4123, A1423, fail.flag%)
IF (HL$ = "l") OR (HL$ = "LOW") THEN res = vdplow!(1, 2, 3, 4, AMPS, DLY!, vlim!,
V1234, V2134, V2341, V3241, V3412, V4312, V4123, V1423, A1234, A2134, A2341, A3241,
A3412, A4312, A4123, A1423, fail.flag%)
IF fail.flag% = 1 THEN GOTO breakerror
RHOFCT = (3.14159 * THICK) / LOG(2)
RA = ((V2134 / -A2134) - (V1234 / A1234) + (V3241 / -A3241) - (V2341 / A2341)) * .25
RHOA = RHOFCT * RA
RB = ((V4312 / -A4312) - (V3412 / A3412) + (V1423 / -A1423) - (V4123 / A4123)) * .25
RHOB = RHOFCT * RB
'-------
'CALCULATES THE VALUES OF RATIO
'-------
IF ((V3241 / -A3241) = (V2341 / A2341)) THEN
     QA = 1
     GOTO vqb
END IF
QA = ((V2134 / -A2134) - (V1234 / A1234)) / ((V3241 / -A3241) - (V2341 / A2341))
IF (QA < 1) THEN QA = 1 / QA
vqb: IF ((V1423 / -A1423) = (V4123 / A4123)) THEN
```

```
     QB = 1
     GOTO vafqb
END IF
QB = ((V4312 / -A4312) - (V3412 / A3412)) / ((V1423 / -A1423) - (V4123 / A4123))
IF QB < 1 THEN QB = 1 / QB
'-------
'CALCULATE THE RATIO CORRECTION FACTORS
'-------
vafqb:
FA = 1 - (.347 * ((QA - 1) / (QA + 1)) ^ 2) - (.092 * ((QA - 1) / (QA + 1)) ^ 4)
FB = 1 - (.347 * ((QB - 1) / (QB + 1)) ^ 2) - (.092 * ((QB - 1) / (QB + 1)) ^ 4)
CRHOA = FA * RHOA
CRHOB = FB * RHOB
'-------
'CALCULATE THE AVERAGE RHO
'-------

RHOAV! = (CRHOA + CRHOB) / 2!
CLS
'-------
'WRITES DATA TO FILE "DATA.RD1"
'-------
OPEN "DATA.RD1" FOR OUTPUT AS #4
WRITE #4, THICK, TEMP!, AMPS, MG!, NAME$, ID$, V2134!, V1234!, RHOA!, A2134!, A1234!,
QA!, V3241!, V2341!, FA!, A3241!, A2341!, CRHOA!, V4312!, V3412!, RHOB!, A4312!,
A3412!, QB!, V1423!, V4123!, FB!, A1423!, A4123!, CRHOB!, RHOAV!
CLOSE #4
'-------
'SETS THE POSITIVE MAGNETIC FIELD
'-------
IF MAG.TYPE = 0 THEN GOTO NOMAG
IF MG! <> 0 THEN
     LOCATE 10, 10: PRINT "Setting Magnetic Field To "; MG!; " KG"
'        CALL forcemag(MG!, MG1!)   'comment out if no magnet controller
     MG1! = MG1! * 10000
     CLS
     LOCATE 10, 10: PRINT "Doing Calculations on sample #"; I%; " in "; MG!; " KG
field"
     LOCATE 11, 10: PRINT "Please Wait"
'-------
'MAKES THE HALL MEASUREMENTS
'-------
     IF (HL$ = "HIGH") THEN CALL HALL(1, 2, 3, 4, AMPS, DLY!, vlim, VP1342!, VP3142!,
VP2413!, VP4213!, AP1342!, AP3142!, AP2413!, AP4213!, fail.flag%)
     IF (HL$ = "LOW") THEN CALL hallow(1, 2, 3, 4, AMPS, DLY!, vlim, VP1342!, VP3142!,
VP2413!, VP4213!, AP1342!, AP3142!, AP2413!, AP4213!, fail.flag%)
     IF fail.flag% = 1 THEN GOTO breakerror
'-------
'WRITES THE DATA TO "DATA.RD1"
'-------
     OPEN "DATA.RD1" FOR APPEND AS #4
     WRITE #4, THICK, TEMP!, AMPS, MG1!, NAME$, ID, VP1342!, VP3142!, VP2413!, VP4213,
AP1342!, AP3142!, AP2413!, AP4213!
     CLOSE #4
'-------
'SETS THE NEGATIVE MAGNETIC FIELD
'-------
     CLS
     LOCATE 10, 10: PRINT "Setting Magnetic Field To "; -MG!; " KG"
'      CALL forcemag(-MG!, MG2!)     ' comment out if no magnet controller
     MG2! = MG2! * 10000
     CLS
'-------
'MAKES THE HALL MEASUREMENTS
```

```
'-------
     LOCATE 10, 10: PRINT "Doing Calculations on Sample #"; I%; " in "; -MG!; " KG
field"
     LOCATE 11, 10: PRINT "Please Wait"
     IF (HL$ = "HIGH") THEN CALL HALL(1, 2, 3, 4, AMPS, DLY!, vlim, VN1342!, VN3142!,
VN2413!, VN4213!, AN1342!, AN3142!, AN2413!, AN4213!, flag.fail%)
     IF (HL$ = "LOW") THEN CALL hallow(1, 2, 3, 4, AMPS, DLY!, vlim, VN1342!, VN3142!,
VN2413!, VN4213!, AN1342!, AN3142!, AN2413!, AN4213!, flag.fail%)
     IF fail.flag% = 1 THEN GOTO breakerror
'-------
'CALCULATE THE VALUES FOR THE HALL COEFFICIENT
'-------
     OPEN "DATA.RD1" FOR INPUT AS #4
     INPUT #4, THICK, TEMP!, AMPS, MG!, NAME$, ID$, V2134!, V1234!, RHOA!, A2134!,
A1234!, QA!, V3241!, V2341!, FA!, A3241!, A2341!, CRHOA!, V4312!, V3412!, RHOB!,
A4312!, A3412!, QB!, V1423!, V4123!, FB!, A1423!, A4123!, CRHOB!, RHOAV!
     INPUT #4, THICK, TEMP!, AMPS, MG1!, NAME$, ID$, VP1342!, VP3142!, VP2413!,
VP4213, AP1342!, AP3142!, AP2413!, AP4213!

     HALFCTP = (2.5E+07 * THICK) / MG1!
     HALFCTN = (2.5E+07 * THICK) / ABS(MG2!)
     RHA! = HALFCTP * ((VP3142 / -AP3142) - (VP1342 / AP1342)) + HALFCTN * ((VN1342 /
AN1342) - (VN3142 / -AN3142))
     RHB! = HALFCTP * ((VP4213 / -AP4213) - (VP2413 / AP2413)) + HALFCTN * ((VN2413 /
AN2413) - (VN4213 / -AN4213))
'-------
'CALCULATES THE SAMPLE TYPE
'-------
     IF RHA < 0 THEN TYPE$ = "N" ELSE TYPE$ = "P"
     RHAV! = ABS((RHA + RHB) / 2)
'-------
' CALCULATE THE MOBILITY
'-------
     MOB = RHAV / RHOAV!
'-------
'CALCULATE THE CARRIER CONCENTRTION
'-------
     N = 6.25E+18 / RHAV
     CLS
'-------
'WRITES ALL THE DATA TO A FILE "DATA.RD2"
'-------
     OPEN "DATA.RD2" FOR OUTPUT AS #5
     WRITE #5, MG2!, MOB!, N!, VN3142!, VN1342!, AN3142!, AN1342!, VN4213!, VN2413!,
AN4213!, AN2413!, RHA!, RHB!, TYPE$
     CLOSE #5
     CLOSE #4
END IF
'-------
'RESETS THE INSTRUMENTS
'-------
NOMAG: CALL devint
IF TEMP! <> 0 THEN CALL restemp    'COMMENT OUT IF NO K20
'-------
'DISPLAYS THE RESULTS
'-------
review: IF fail.flag% = 1 THEN GOTO breakerror
OPEN "DATA.RD1" FOR INPUT AS #4
INPUT #4, THICK, TEMP!, AMPS, MG!, NAME$, ID$, V2134!, V1234!, RHOA!, A2134!, A1234!,
QA!, V3241!, V2341!, FA!, A3241!, A2341!, CRHOA!, V4312!, V3412!, RHOB!, A4312!,
A3412!, QB!, V1423!, V4123!, FB!, A1423!, A4123!, CRHOB!, RHOAV!
IF (MG! <> 0 AND MAG.TYPE <> 0) THEN
     INPUT #4, THICK, TEMP!, AMPS, MG1!, NAME$, ID$, VP1342!, VP3142!, VP2413!,
VP4213, AP1342!, AP3142!, AP2413!, AP4213!
```

```
     OPEN "DATA.RD2" FOR INPUT AS #5
     INPUT #5, MG2!, MOB!, N!, VN3142!, VN1342!, AN3142!, AN1342!, VN4213!, VN2413!,
AN4213!, AN2413!, RHA!, RHB!, TYPE$
END IF
CALL resul(NAME$, ID$, THICK, TEMP!, AMPS, MG!, RHOAV!, RHAV!, MOB!, N!, TYPE$,
quit.flag%)
IF (quit.flag% = 1) THEN
     quit.flag% = 0
     CLOSE #4
     IF (MG! <> 0) THEN CLOSE #5
     GOTO filewr
END IF
CALL resu2(NAME$, ID$, THICK, TEMP!, MG!, AMPS, V2134!, A2134!, V1234!, A1234!, RHOA!,
QA!, V3241!, A3241!, V2341!, A2341!, FA!, CRHOA!, V4312!, A4312!, V3412!, A3412!,
RHOB!, QB!, V1423!, A1423!, V4123!, A4123!, FB!, CRHOB!, quit.flag%)
IF (quit.flag% = 1) THEN
     CLOSE #4
     IF (MG! <> 0) THEN CLOSE #5
     quit.flag% = 0
     GOTO filewr
END IF
IF (MG! <> 0 AND MAG.TYPE <> 0) THEN
     CALL resu3(start.flag%, NAME$, ID$, THICK, TEMP!, MG!, AMPS, VP3142!, AP3142!,
VP1342!, AP1342!, VP4213!, AP4213!, VP2413!, AP2413!, VN3142!, AN3142!, VN1342!,
AN1342!, VN4213!, AN4213!, VN2413!, AN2413!, RHA!, RHB!)
     CLOSE #5
END IF
CLOSE #4
'-------
'RESULTS SENT TO OUTPUT FILE ID+".DAT"
'-------
filewr: OPEN "DATA.RD1" FOR INPUT AS #4
INPUT #4, THICK, TEMP!, AMPS, MG!, NAME$, ID$, V2134!, V1234!, RHOA!, A2134!, A1234!,
QA!, V3241!, V2341!, FA!, A3241!, A2341!, CRHOA!, V4312!, V3412!, RHOB!, A4312!,
A3412!, QB!, V1423!, V4123!, FB!, A1423!, A4123!, CRHOB!, RHOAV!
IF (MG! <> 0 AND MAG.TYPE <> 0) THEN
     INPUT #4, THICK, TEMP!, AMPS, MG1!, NAME$, ID$, VP1342!, VP3142!, VP2413!,
VP4213, AP1342!, AP3142!, AP2413!, AP4213!
     OPEN "DATA.RD2" FOR INPUT AS #5
     INPUT #5, MG2!, MOB!, N!, VN3142!, VN1342!, AN3142!, AN1342!, VN4213!, VN2413!,
AN4213!, AN2413!, RHA!, RHB!, TYPE$
END IF
OPEN ID + ".DAT" FOR OUTPUT AS #6
CALL outfile(THICK, TEMP!, AMPS, MG, MG2!, MG1!, NAME$, ID$, RHOAV!, RHAV!, TYPE$,
MOB!, N!, V2134!, V1234!, RHOA!, A2134!, A1234!, QA!, V3241!, V2341!, FA!, A3241!,
A2341!, CRHOA!, V4312!, V3412!, RHOB!, A4312!, A3412!, QB!, V1423!, V4123!, FB!, _
A1423!, A4123!, CRHOB!, VP3142!, VP1342!, AP3142!, AP1342!, VP4213!, VP2413!, AP4213!,
AP2413!, VN3142!, VN1342!, AN3142!, AN1342!, VN4213!, VN2413!, AN4213!, AN2413!, RHA!,
RHB!)
CLOSE #6
CLOSE #4
IF MG! <> 0 THEN CLOSE #5
     '**************************************************
     '**************************************************
     ' PLACE PRINT FILE TO A PRINTER STATEMENT HERE
     ' FILE NAME TO BE PRINTED IS ID$."DAT"
     '**************************************************
     '**************************************************
'-------
'THIS SECTION PRINTS OUT THE RESULTS IF THE FAILFLAG HAS BEEN SET
'-------
breakerror: IF fail.flag% = 1 THEN
     CALL resul(NAME$, ID$, THICK, TEMP!, AMPS, MG!, RHOAV!, RHAV!, MOB!, N!, TYPE$,
quit.flag%)
```

```
     IF (quit.flag% = 1) THEN GOTO outfle
     CALL resu2(NAME$, ID$, THICK, TEMP!, MG!, AMPS, V2134!, A2134!, V1234!, A1234!,
RHOA!, QA!, V3241!, A3241!, V2341!, A2341!, FA!, CRHOA!, V4312!, A4312!, V3412!,
A3412!, RHOB!, QB!, V1423!, A1423!, V4123!, A4123!, FB!, CRHOB!, quit.flag%)
     IF (quit.flag% = 1) THEN GOTO outfle
     IF MG! <> 0 THEN CALL resu3(start.flag%, NAME$, ID$, THICK, TEMP!, MG!, AMPS,
VP3142!, AP3142!, VP1342!, AP1342!, VP4213!, AP4213!, VP2413!, AP2413!, VN3142!,
AN3142!, VN1342!, AN1342!, VN4213!, AN4213!, VN2413!, AN2413!, RHA!, RHB!)
outfle: OPEN ID + ".DAT" FOR OUTPUT AS #6
     CALL outfile(THICK, TEMP!, AMPS, MG, MG2!, MG1!, NAME$, ID$, RHOAV!, RHAV!,
TYPE$, MOB!, N!, V2134!, V1234!, RHOA!, A2134!, A1234!, QA!, V3241!, V2341!, FA!,
A3241!, A2341!, CRHOA!, V4312!, V3412!, RHOB!, A4312!, A3412!, QB!, V1423!, V4123!
, FB!, A1423!, A4123!, CRHOB!, VP3142!, VP1342!, AP3142!, AP1342!, VP4213!, VP2413!,
AP4213!, AP2413!, VN3142!, VN1342!, AN3142!, AN1342!, VN4213!, VN2413!, AN4213!,
AN2413!, RHA!, RHB!)
     CLOSE #6
END IF
CALL final(start.flag%, review.flag%)
IF (start.flag% = 1) THEN
     start.flag% = 0
     fail.flag% = 0
     GOTO start
END IF
IF (review.flag% = 1) THEN
     review.flag% = 0
     GOTO review
END IF

END

SUB final (start.flag%, review.flag%)
' Prototype for screen: LAST
'$INCLUDE: 'c:\s110\ps\LAST.inc'

sc.name$ = "LAST"
CALL sc.init(sc.name$)

Sc.Text% = -1
Sc.Clear% = 0
sc.output% = -1

CLS
CALL sc.msg("", 2, 0)
GOSUB LAST


Sc.Text% = 0

WHILE (NOT ((sc.pf% = 3) OR (sc.pf% = 7) OR (sc.pf% = 5)))
   GOSUB LAST
WEND

IF (sc.pf% = 3) THEN
     start.flag% = 1

END IF
IF (sc.pf% = 5) THEN
     review.flag% = 1

END IF


COLOR 7, 0
CLS
```

```
END SUB

SUB getkey (NAME$, ID$, HL$, TEMP!, THICK!, MG!, AMPS!, DLY!, I%)


'$INCLUDE: 'c:\s110\ps\SCREEN1.inc'
'
'
OPEN "GETKEY.DAT" FOR INPUT AS #1
INPUT #1, NAME$, ID$, HL$, TEMP!, THICK!, MG!, AMPS!, DLY!, I%
CLOSE #1
' Prototype for screen: SCREEN1


sc.name$ = "SCREEN1"
CALL sc.init(sc.name$)

Sc.Text% = -1
Sc.Clear% = 0
sc.output% = -1

CLS
CALL sc.msg("F3=begin test  F7=quit", 2, 0)
GOSUB SCREEN1


Sc.Text% = 0
WHILE (NOT ((sc.pf% = 3) OR (sc.pf% = 7)))

   GOSUB SCREEN1
WEND

COLOR 7, 0
CLS
IF sc.pf% = 7 THEN END

OPEN "GETKEY.DAT" FOR OUTPUT AS #1
WRITE #1, NAME$, ID$, HL$, TEMP!, THICK!, MG!, AMPS!, DLY!, I%
CLOSE #1



END SUB

SUB HALL (p1, p2, p3, p4, ival!, DLY!, vlim!, v1, v2, v3, v4, a1, a2, a3, a4,
fail.flag%)
'v1 = v1342  v2 = v3142 v3 = v2413  v4 = v4213
'i1 = i1342  i2 = i3142 i3 = i2413  i4 = i4213
fail.flag% = 0
'-------
'MAKES CONNECTION 13,42
'-------
ival1 = ival
hstart1: CALL connect(ISRC1, p1)
CALL connect(VMTR2H, p4)
CALL connect(GND, p3)
CALL connect(VMTR2L, p2)
CALL forcei(ISRC1, ival1!)
CALL delay(DLY!)
CALL measv(VMTR2, v1)
CALL measi(IMTR1, a1)
```

```
IF (v1 = 2E+22 OR a1 = 2E+22) THEN
     ival1! = ival1! / 10
     IF ABS(ival1!) <= 1E-11 THEN
          v1 = -2E+22
          v2 = -2E+22
          a1 = -2E+22
          a2 = -2E+22
          fail.flag% = 1
          GOTO hstart2a
     END IF
     GOTO hstart1
END IF
'-------
'MAKES CONNECTION 31,42
'-------
CALL forcei(ISRC1, -ival1!)
CALL delay(DLY!)
CALL measv(VMTR2, v2)
CALL measi(IMTR1, a2)
'-------
'MAKES CONNECTION 24,13
'-------
hstart2a: ival2 = ival
hstart2: CALL connect(ISRC1, p2)
CALL connect(VMTR2H, p1)
CALL connect(GND, p4)
CALL connect(VMTR2L, p3)
CALL forcei(ISRC1, ival2!)
CALL delay(DLY!)
CALL measv(VMTR2, v3)
CALL measi(IMTR1, a3)
IF (v3 = 2E+22 OR a3 = 2E+22) THEN
     ival2! = ival2! / 10
     IF ABS(ival2!) <= 1E-11 THEN
          v3 = -2E+22
          v4 = -2E+22
          a3 = -2E+22
          a4 = -2E+22
          fail.flag% = 1
          GOTO hstart3a
     END IF
     GOTO hstart2
END IF
'-------
'MAKES CONNECTION 42,13
'-------
CALL forcei(ISRC1, -ival2!)
CALL delay(DLY!)
CALL measv(VMTR2, v4)
CALL measi(IMTR1, a4)
hstart3a:
END SUB

SUB hallow (p1, p2, p3, p4, ival!, DLY!, vlim!, v1!, v2!, v3!, v4!, a1!, a2!, a3!,
a4!, fail.flag%)
fail.flag% = 0
'-------
'MAKES CONNECTION 13,42
'-------
ival1 = ival
hlstart1: CALL connect(ISRC1, p1)
CALL connect(VMTR2H, p4)
CALL connect(GND, p3)
CALL connect(VMTR2L, p2)
```

```
CALL forcei(ISRC1, ivall!)
CALL delay(DLY!)
CALL measv(VMTR2, v1)
a1! = ivall!
IF (v1 = 2E+22 OR a1 = 2E+22) THEN
     ivall! = ivall! / 10
     IF ABS(ivall!) <= 1E-11 THEN
          v1 = -2E+22
          v2 = -2E+22
          a1 = -2E+22
          a2 = -2E+22
          fail.flag% = 1
          GOTO hlstart2a
     END IF
     GOTO hlstart1
END IF
'-------
'MAKES CONNECTION 31,42
'-------
CALL forcei(ISRC1, -ivall!)
CALL delay(DLY!)
CALL measv(VMTR2, v2)
a2! = -ivall!
'-------
'MAKES CONNECTION 24,13
'-------
hlstart2a: ival2 = ival
hlstart2: CALL connect(ISRC1, p2)
CALL connect(VMTR2H, p1)
CALL connect(GND, p4)
CALL connect(VMTR2L, p3)
CALL forcei(ISRC1, ival2!)
CALL delay(DLY!)
CALL measv(VMTR2, v3)
a3! = ival2!
IF (v3 = 2E+22 OR a3 = 2E+22) THEN
     ival2! = ival2! / 10
     IF ABS(ival2!) <= 1E-11 THEN
          v3 = -2E+22
          v4 = -2E+22
          a3 = -2E+22
          a4 = -2E+22
          fail.flag% = 1
          GOTO hlstart3a
     END IF
     GOTO hlstart2
END IF
'-------
'MAKES CONNECTION 42,13
'-------
CALL forcei(ISRC1, -ival2!)
CALL delay(DLY!)
CALL measv(VMTR2, v4)
a4! = -ival2!
hlstart3a:
END SUB

SUB outfile (THICK!, TEMP!, AMPS!, MG!, MG2!, MG1!, NAME$, ID$, RHOAV!, RHAV!, TYPE$,
MOB!, N!, V2134!, V1234!, RHOA!, A2134!, A1234!, QA!, V3241!, V2341!, FA!, A3241!,
A2341!, CRHOA!, V4312!, V3412!, RHOB!, A4312!, A3412!, QB!, V1423!, V4123!, FB! _
, A1423!, A4123!, CRHOB!, VP3142!, VP1342!, AP3142!, AP1342!, VP4213!, VP2413!,
AP4213!, AP2413!, VN3142!, VN1342!, AN3142!, AN1342!, VN4213!, VN2413!, AN4213!,
AN2413!, RHA!, RHB!)
```

```
PRINT #6, " "
PRINT #6, TAB(10); "     date: "; DATE$; TAB(40); "Sample Thickness: "; THICK!; " cm"
PRINT #6, TAB(10); "     Time: "; TIME$; TAB(40); "     Temperature: "; TEMP!
PRINT #6, TAB(10); " Operator: "; NAME$; TAB(40); "    Test Current: "; AMPS!
PRINT #6, TAB(10); "Sample ID: "; ID$; TAB(37); " Magnetic Field(KG): "; MG!
PRINT #6, " "
PRINT #6, " "
PRINT #6, TAB(10); "           Resistivity (Ohm-cm): "; USING "##.####^^^^"; RHOAV!
PRINT #6, " "
PRINT #6, TAB(10); "Hall Coefficient (cm**3 - C**-1): "; USING "###.####^^^^"; RHAV!
PRINT #6, " "
PRINT #6, TAB(10); "           Sample Type (N or P):  "; TYPE$
PRINT #6, " "
PRINT #6, TAB(10); "          Mobility (cm**2/V-sec): "; USING "###.####^^^^"; MOB!
PRINT #6, " "
PRINT #6, TAB(10); "  Carrier Concentration (cm**-3): "; USING "###.####^^^^"; N!
PRINT #6, CHR$(12) ' form feed
PRINT #6, " "
PRINT #6, TAB(30); "RESISTIVITY"
PRINT #6, " "
PRINT #6, TAB(5); "V21,34 = "; V2134!; TAB(30); "V12,34 = "; V1234!; TAB(55); "   RHOA
= "; USING "###.####^^^^"; RHOA!
PRINT #6, TAB(5); "I21,34 = "; A2134!; TAB(30); "A12,34 = "; A1234!; TAB(55); "     Qa
= "; USING "###.####^^^^"; QA!
PRINT #6, TAB(5); "V32,41 = "; V3241!; TAB(30); "V23,41 = "; V2341!; TAB(55); "     Fa
= "; USING "###.####^^^^"; FA!
PRINT #6, TAB(5); "I32,41 = "; A3241!; TAB(30); "A23,41 = "; A2341!; TAB(55); " RHOACo
= "; USING "###.####^^^^"; CRHOA!
PRINT #6, TAB(5); "V43,12 = "; V4312!; TAB(30); "V34,12 = "; V3412!; TAB(55); "   RHOB
= "; USING "###.####^^^^"; RHOB!
PRINT #6, TAB(5); "I43,12 = "; A4312!; TAB(30); "A34,12 = "; A3412!; TAB(55); "     Qb
= "; USING "###.####^^^^"; QB!
PRINT #6, TAB(5); "V14,23 = "; V1423!; TAB(30); "V41,23 = "; V4123!; TAB(55); "     Fb
= "; USING "###.####^^^^"; FB!
PRINT #6, TAB(5); "I14,23 = "; A1423!; TAB(30); "A41,23 = "; A4123!; TAB(55); " RHOBCo
= "; USING "###.####^^^^"; CRHOB!
PRINT #6, " "
PRINT #6, TAB(30); "HALL COEFFICIENT"
PRINT #6, " "
PRINT #6, TAB(5); "    +MG = "; MG1!; TAB(30); "V31,42 = "; VP3142!; TAB(55); "V13,42
= "; VP1342
PRINT #6, TAB(30); "I31,42 = "; AP3142!; TAB(55); "I13,42 = "; AP1342
PRINT #6, TAB(30); "V42,13 = "; VP4213!; TAB(55); "V24,13 = "; VP2413
PRINT #6, TAB(30); "I42,13 = "; AP4213!; TAB(55); "I24,13 = "; AP2413
PRINT #6, " "
PRINT #6, TAB(5); "    -MG = "; MG2!; TAB(30); "V31,42 = "; VN3142!; TAB(55); "V13,42
= "; VN1342
PRINT #6, TAB(30); "I31,42 = "; AN3142!; TAB(55); "I13,42 = "; AN1342
PRINT #6, TAB(30); "V42,13 = "; VN4213!; TAB(55); "V24,13 = "; VN2413
PRINT #6, TAB(30); "I42,13 = "; AN4213!; TAB(55); "I24,13 = "; AN2413
PRINT #6, " "
PRINT #6, TAB(30); "   RHA = "; USING "###.####^^^^"; RHA!
PRINT #6, TAB(30); "   RHB = "; USING "###.####^^^^"; RHB!

END SUB

SUB resul (NAME$, ID$, THICK!, TEMP!, AMPS!, MG!, RHOAV!, RHAV!, MOB!, N!, TYPE$,
quit.flag%)
' Prototype for screen: RESULT1
'$INCLUDE: 'c:\s110\ps\RESULT1.inc'

sc.name$ = "RESULT1"
CALL sc.init(sc.name$)
```

```
Sc.Text% = -1
Sc.Clear% = 0
sc.output% = -1

CLS
CALL sc.msg(" F3=goto next menu  F7=quit", 2, 0)
GOSUB RESULT1


Sc.Text% = 0

WHILE (NOT ((sc.pf% = 3) OR (sc.pf% = 7)))
   GOSUB RESULT1
WEND

IF (sc.pf% = 7) THEN
     quit.flag% = 1
END IF


COLOR 7, 0
CLS



END SUB

SUB resu2 (NAME$, ID$, THICK!, TEMP!, MG!, AMPS!, V2134!, I2134!, V1234!, I1234!,
RHOA!, QA!, V3241!, I3241!, V2341!, I2341!, FA!, CRHOA!, V4312!, I4312!, V3412!,
I3412!, RHOB!, QB!, V1423!, I1423!, V4123!, I4123!, FB!, CRHOB!, quit.flag%)
' Prototype for screen: RESULT2
'$INCLUDE: 'c:\s110\ps\RESULT2.inc'

sc.name$ = "RESULT2"
CALL sc.init(sc.name$)

Sc.Text% = -1
Sc.Clear% = 0
sc.output% = -1

CLS
CALL sc.msg(" F3=goto next menu  F7=quit", 2, 0)
GOSUB RESULT2


Sc.Text% = 0


WHILE (NOT ((sc.pf% = 3) OR (sc.pf% = 7)))
   GOSUB RESULT2
WEND

IF (sc.pf% = 7) THEN
     quit.flag% = 1
END IF



COLOR 7, 0
CLS



END SUB
```

```
SUB resu3 (start.flag%, NAME$, ID$, THICK!, TEMP!, MG!, AMPS!, VP3142!, IP3142!,
VP1342!, IP1342!, VP4213!, IP4213!, VP2413!, IP2413!, VN3142!, IN3142!, VN1342!,
IN1342!, VN4213!, IN4213!, VN2413!, IN2413!, RHA!, RHB!)

' Prototype for screen: RESULT3
'$INCLUDE: 'c:\s110\ps\RESULT3.inc'

sc.name$ = "RESULT3"
CALL sc.init(sc.name$)

Sc.Text% = -1
Sc.Clear% = 0
sc.output% = -1

CLS
CALL sc.msg("F3=goto next menu", 2, 0)
GOSUB RESULT3


Sc.Text% = 0

WHILE (NOT ((sc.pf% = 3)))
   GOSUB RESULT3
WEND



COLOR 7, 0
CLS
END SUB

 FUNCTION vdp (p1, p2, p3, p4, AMPS, DLY, vlim, v1, v2, v3, v4, v5, v6, v7, v8, a1,
a2, a3, a4, a5, a6, a7, a8, fail.flag%)
 fail.flag% = 0
'-------
'MAKES CONNECTION 12,34
'-------
AMPS1 = AMPS
start1: CALL connect(ISRC1, p1)
CALL connect(GND, p2)
CALL connect(VMTR2H, p3)
CALL connect(VMTR2L, p4)
CALL forcei(ISRC1, AMPS1!)
CALL delay(DLY!)
CALL measv(VMTR2, v1)
CALL measi(IMTR1, a1)
IF (v1 = 2E+22 OR a1 = 2E+22) THEN
     AMPS1! = AMPS1! / 10
     IF ABS(AMPS1!) <= 1E-11 THEN
          v1 = -2E+22
          a1 = -2E+22
          v2 = -2E+22
          a2 = -2E+22
          fail.flag% = 1
          GOTO start2a
     END IF
     GOTO start1
END IF
'-------
'MAKES CONNECTION 21,34
'-------
CALL forcei(ISRC1, -AMPS1!)
CALL delay(DLY!)
```

```
CALL measv(VMTR2, v2)
CALL measi(IMTR1, a2)
'-------
'MAKES CONNECTION 23,41
'-------
start2a: AMPS2 = AMPS
start2: CALL connect(ISRC1, p2)
CALL connect(VMTR2H, p4)
CALL connect(GND, p3)
CALL connect(VMTR2L, p1)
CALL forcei(ISRC1, AMPS2!)
CALL delay(DLY!)
CALL measv(VMTR2, v3)
CALL measi(IMTR1, a3)
IF (v3 = 2E+22 OR a3 = 2E+22) THEN
     AMPS2! = AMPS2! / 10
     IF ABS(AMPS2!) <= 1E-11 THEN
          v3 = -2E+22
          a3 = -2E+22
          v4 = -2E+22
          a4 = -2E+22
          fail.flag% = 1
          GOTO start3a
     END IF
     GOTO start2
END IF
'-------
'MAKES CONNECTION 32,41
'-------
CALL forcei(ISRC1, -AMPS2!)
CALL delay(DLY!)
CALL measv(VMTR2, v4)
CALL measi(IMTR1, a4)
'-------
'MAKES CONNECTION 34,12
'-------
start3a: AMPS3 = AMPS
start3: CALL connect(ISRC1, p3)
CALL connect(VMTR2H, p1)
CALL connect(GND, p4)
CALL connect(VMTR2L, p2)
CALL forcei(ISRC1, AMPS3!)
CALL delay(DLY!)
CALL measv(VMTR2, v5)
CALL measi(IMTR1, a5)
IF (v5 = 2E+22 OR a5 = 2E+22) THEN
     AMPS3! = AMPS3! / 10
     IF ABS(AMPS3!) <= 1E-11 THEN
          v5 = -2E+22
          a5 = -2E+22
          v6 = -2E+22
          a6 = -2E+22
          fail.flag% = 1
          GOTO start4a
     END IF
     GOTO start3
END IF
'-------
'MAKES CONNECTION 43,12
'-------
CALL forcei(ISRC1, -AMPS3!)
CALL delay(DLY!)
CALL measv(VMTR2, v6)
CALL measi(IMTR1, a6)
```

```
'-------
'MAKES CONNECTION 41,23
'-------
start4a: AMPS4 = AMPS
start4: CALL connect(ISRC1, p4)
CALL connect(VMTR2H, p2)
CALL connect(GND, p1)
CALL connect(VMTR2L, p3)
CALL forcei(ISRC1, AMPS4!)
CALL delay(DLY!)
CALL measv(VMTR2, v7)
CALL measi(IMTR1, a7)
IF (v7 = 2E+22 OR a7 = 2E+22) THEN
     AMPS4! = AMPS4! / 10
     IF ABS(AMPS4!) <= 1E-11 THEN
          v7 = -2E+22
          a7 = -2E+22
          v8 = -2E+22
          a8 = -2E+22
          fail.flag% = 1
          GOTO start5a
     END IF
     GOTO start4
END IF
'-------
'MAKES CONNECTION 14,23
'-------
CALL forcei(ISRC1, -AMPS4!)
CALL delay(DLY!)
CALL measv(VMTR2, v8)
CALL measi(IMTR1, a8)
'-------
'THIS SECTION CALCULATES THE RESISTIVITY IF THE SAMPLE THICKNESS IS 1cm
'-------
start5a: THICK = 1!

RHOFCT = (3.14159 * THICK) / LOG(2)
RA = ((v2 / -a2) - (v1 / a1) + (v4 / -a4) - (v3 / a3)) * .25
RHOA = RHOFCT * RA
RB = ((v6 / -a6) - (v5 / a5) + (v8 / -a8) - (v7 / a7)) * .25
RHOB = RHOFCT * RB
'-------
'CALCULATES THE VALUES OF RATIO
'-------
IF ((v4 / -a4) = (v3 / a3)) THEN
     QA = 1
     GOTO doqb
END IF
QA = ((v2 / -a2) - (v1 / a1)) / ((v4 / -a4) - (v3 / a3))
IF (QA < 1) THEN QA = 1 / QA
doqb: IF ((v8 / -a8) = (v7 / a7)) THEN
     QB = 1
     GOTO afqb
END IF
QB = ((v6 / -a6) - (v5 / a5)) / ((v8 / -a8) - (v7 / a7))
IF QB < 1 THEN QB = 1 / QB
'-------
'CALCULATE THE RATIO CORRECTION FACTORS
'-------
afqb:
FA = 1 - (.347 * ((QA - 1) / (QA + 1)) ^ 2) - (.092 * ((QA - 1) / (QA + 1)) ^ 4)
FB = 1 - (.347 * ((QB - 1) / (QB + 1)) ^ 2) - (.092 * ((QB - 1) / (QB + 1)) ^ 4)
CRHOA = FA * RHOA
CRHOB = FB * RHOB
```

```
'-------
'CALCULATE THE AVERAGE RHO
'-------
RHOAV = (CRHOA + CRHOB) / 2!

vdp = RHOAV

END FUNCTION

FUNCTION vdplow (p1, p2, p3, p4, AMPS, DLY, vlim, v1, v2, v3, v4, v5, v6, v7, v8, a1,
a2, a3, a4, a5, a6, a7, a8, fail.flag%)
'-------
'MAKES CONNECTION 12,34
'-------
fail.flag% = 0
AMPS1 = AMPS
vlstart1: CALL connect(ISRC1, p1)
CALL connect(GND, p2)
CALL connect(VMTR2H, p3)
CALL connect(VMTR2L, p4)
CALL forcei(ISRC1, AMPS1!)
CALL delay(DLY!)
CALL measv(VMTR2, v1)
IF (v1 = 2E+22) THEN
     AMPS1! = AMPS1! / 10
     IF ABS(AMPS1!) <= 1E-11 THEN
          v1 = -2E+22
          a1 = -2E+22
          v2 = -2E+22
          a2 = -2E+22
          fail.flag% = 1
          GOTO vlstart2a
     END IF
     GOTO vlstart1
END IF
'-------
'MAKES CONNECTION 21,34
'-------
a1! = AMPS1!
CALL forcei(ISRC1, -AMPS1!)
CALL delay(DLY!)
CALL measv(VMTR2, v2)
a2! = -AMPS1!
'-------
'MAKES CONNECTION 23,41
'-------
vlstart2a: AMPS2 = AMPS
vlstart2: CALL connect(ISRC1, p2)
CALL connect(VMTR2H, p4)
CALL connect(GND, p3)
CALL connect(VMTR2L, p1)
CALL forcei(ISRC1, AMPS2!)
CALL delay(DLY!)
CALL measv(VMTR2, v3)
IF (v3 = 2E+22) THEN
     AMPS2! = AMPS2! / 10
     IF ABS(AMPS2!) <= 1E-11 THEN
          v3 = -2E+22
          a3 = -2E+22
          v4 = -2E+22
          a4 = -2E+22
          fail.flag% = 1
          GOTO vlstart3a
     END IF
```

```
     GOTO vlstart2
END IF
'-------
'MAKES CONNECTION 32,41
'-------
a3! = AMPS2!
CALL forcei(ISRC1, -AMPS2!)
CALL delay(DLY!)
CALL measv(VMTR2, v4)
a4! = -AMPS2!
'-------
'MAKES CONNECTION 34,12
'-------
vlstart3a: AMPS3 = AMPS
vlstart3: CALL connect(ISRC1, p3)
CALL connect(VMTR2H, p1)
CALL connect(GND, p4)
CALL connect(VMTR2L, p2)
CALL forcei(ISRC1, AMPS3!)
CALL delay(DLY!)
CALL measv(VMTR2, v5)
IF (v5 = 2E+22) THEN
     AMPS3! = AMPS3! / 10
     IF ABS(AMPS3!) <= 1E-11 THEN
          v5 = -2E+22
          a5 = -2E+22
          v6 = -2E+22
          a6 = -2E+22
          fail.flag% = 1
          GOTO vlstart4a
     END IF
     GOTO vlstart3
END IF
'-------
'MAKES CONNECTION 43,12
'-------
a5! = AMPS3!
CALL forcei(ISRC1, -AMPS3!)
CALL delay(DLY!)
CALL measv(VMTR2, v6)
a6! = -AMPS3!
'-------
'MAKES CONNECTION 41,23
'-------
vlstart4a: AMPS4 = AMPS
vlstart4: CALL connect(ISRC1, p4)
CALL connect(VMTR2H, p2)
CALL connect(GND, p1)
CALL connect(VMTR2L, p3)
CALL forcei(ISRC1, AMPS4!)
CALL delay(DLY!)
CALL measv(VMTR2, v7)
IF (v7 = 2E+22) THEN
     AMPS4! = AMPS4! / 10
     IF ABS(AMPS4!) <= 1E-11 THEN
          v7 = -2E+22
          a7 = -2E+22
          v8 = -2E+22
          a8 = -2E+22
          fail.flag% = 1
          GOTO vlstart5a
     END IF
     GOTO vlstart4
END IF
```

```
'-------
'MAKES CONNECTION 14,23
'-------
a7! = AMPS4!
CALL forcei(ISRC1, -AMPS4!)
CALL delay(DLY!)
CALL measv(VMTR2, v8)
a8! = -AMPS4!
'-------
'THIS SECTION CALCULATES THE RESISTIVITY IF THE SAMPLE THICKNESS IS 1cm
'-------

vlstart5a: THICK = 1!
IF fail.flag% = 1 THEN GOTO bottom
RHOFCT = (3.14159 * THICK) / LOG(2)
RA = ((v2 / -a2) - (v1 / a1) + (v4 / -a4) - (v3 / a3)) * .25
RHOA = RHOFCT * RA
RB = ((v6 / -a6) - (v5 / a5) + (v8 / -a8) - (v7 / a7)) * .25
RHOB = RHOFCT * RB
'-------
'CALCULATES THE VALUES OF RATIO
'-------
IF ((v4 / -a4) = (v3 / a3)) THEN
     QA = 1
     GOTO vlqb
END IF
QA = ((v2 / -a2) - (v1 / a1)) / ((v4 / -a4) - (v3 / a3))
IF (QA < 1) THEN QA = 1 / QA
vlqb: IF ((v8 / -a8) = (v7 / a7)) THEN
     GOTO vlafqb
END IF
QB = ((v6 / -a6) - (v5 / a5)) / ((v8 / -a8) - (v7 / a7))
IF QB < 1 THEN QB = 1 / QB
'-------
'CALCULATE THE RATIO CORRECTION FACTORS
'-------
vlafqb:
FA = 1 - (.347 * ((QA - 1) / (QA + 1)) ^ 2) - (.092 * ((QA - 1) / (QA + 1)) ^ 4)
FB = 1 - (.347 * ((QB - 1) / (QB + 1)) ^ 2) - (.092 * ((QB - 1) / (QB + 1)) ^ 4)
CRHOA = FA * RHOA
CRHOB = FB * RHOB
'-------
'CALCULATE THE AVERAGE RHO
'-------
RHOAV = (CRHOA + CRHOB) / 2!

vdplow = RHOAV
bottom:
END FUNCTION
```

C.2 HALBRDG.BAS Program Listing

```
'**************************************************************************
'**************************************************************************
'                          Hall Bridge Test
'   Copyright (C) 1990 by Keithley Instruments, Inc.
'   Cleveland, Ohio
'
'   This software is furnished under a license and may be used and copied
'   only in accordance with the terms of such license, and with the inclusion
'   of the above COPYRIGHT notice.  This software or any other copies thereof
'   may not be provided or otherwise made available to any other person. No
'   title to and ownership of the software is hereby transferred.
'
'   The information in this software is subject to change without notice, and
'   should not be construed as a commitment by KEITHLEY INSTRUMENTS, INC.
'
'   KEITHLEY assumes no responsibility for the use or reliablity of its
'   software on equipment which is not supplied by KEITHLEY.
'**************************************************************************
'   History
'
'       The purpose of this program is to calculate resistivity and Hall calculations
'
'   V1.0        8/10/90                 Brian Polaski
'**************************************************************************

' $INCLUDE: 'c:\ps\linkscrn.dim'
' $INCLUDE: 'c:\s110\lib\S110.QB'
DIM I AS INTEGER
DIM ID AS STRING
start: CALL hbgetkey(NAME$, ID$, HL$, TEMP!, THICK!, MG!, AMPS!, DLY!, I%, SPAC!,
WID!)
'-------
'SETS TEMP OF THE SAMPLE
'-------
CALL tstsel(1)    'must have this command first
IF TEMP! <> 0 THEN CALL restemp   'COMMENT OUT IF NO K20 CRYO
IF (TEMP.TYPE <> 0) THEN
     IF TEMP! <> 0 THEN
            CALL settemp(TEMP!, gettemp!)          'COMMENT OUT IF NO TEMP CONTROLLER
          IF gettemp < 0 THEN
                IF TEMP! <> 0 THEN CALL restemp   'COMMENT OUT IF NO K20 CRYO
               GOTO start
          END IF
          TEMP! = gettemp!
     END IF
END IF
'-------
'INITIALIZES THE INSTRUMENTS & SETS FOR HIGH OR LOW TERMINALS
'-------

 IF (HL$ = "H") OR (HL$ = "HIGH") THEN hlsel (HIGH)
 IF (HL$ = "L") OR (HL$ = "LOW") THEN hlsel (LOW)

vlim = 10
CLS
'-------
'DEFAULTS ALL THE RESISTIVITY VALUES TO 1E+22
'-------
fail.flag% = 0
V1246 = 1E+22
V2146 = 1E+22
```

```
V1235 = 1E+22
V2135 = 1E+22
A1246 = 1E+22
A2146 = 1E+22
A1235 = 1E+22
A2135 = 1E+22
RHOAV = 1E+22
'-------
'MAKES THE HALL-BRIDGE MEASUREMENTS
'-------

LOCATE 10, 10: PRINT "Doing Calculations on Sample #"; I%
LOCATE 11, 10: PRINT "Please Wait"
res = HLB!(1, 2, 3, 4, AMPS, DLY!, vlim!, V1246, V2146, V1235, V2135, A1246, A2146,
A1235, A2135, fail.flag%)
IF fail.flag% = 1 THEN GOTO hallp
RHOFCT = (WID! * THICK!) / SPAC!
RA = ((V1246 / A1246) - (V2146 / -A2146)) * .5
RHOA = RHOFCT * RA
RB = ((V1235 / A1235) - (V2135 / -A2135)) * .5
RHOB = RHOFCT * RB
'-------
'CALCULATE THE AVERAGE RHO
'-------

RHOAV! = (RHOA + RHOB) / 2
CLS
'-------
'WRITES DATA TO FILE "DATA.RD1"
'-------
hallp:
OPEN "DATA.RD1" FOR OUTPUT AS #4
WRITE #4, THICK, TEMP!, AMPS, MG!, NAME$, ID$, WID!, SPAC!, V1246!, V2146!, A1246!,
A2146!, V1235!, V2135!, A1235!, A2135!, RHOA!, RHOB!, RHOAV!
CLOSE #4
'-------
'DEFAULTS ALL POSITIVE HALL MEASUREMENTS TO 1E+22
'-------
VP1265! = 1E+22
VP2165! = 1E+22
VP1243! = 1E+22
VP2143! = 1E+22
AP1265! = 1E+22
AP2165! = 1E+22
AP1243! = 1E+22
AP2143! = 1E+22
'-------
'SETS THE POSITIVE MAGNETIC FIELD
'-------

IF MAG.TYPE = 0 THEN GOTO NOMAG
IF fail.flag% = 1 THEN GOTO record
IF MG! <> 0 THEN
     LOCATE 10, 10: PRINT "Setting Magnetic Field To "; MG!; " KG"
     'CALL forcemag(MG!, MG1!)   'comment out if no magnet controller
     MG1! = MG1! * 10000
     CLS
     LOCATE 10, 10: PRINT "Doing Calculations on sample #"; I%; " in "; MG!; " KG
field"
     LOCATE 11, 10: PRINT "Please Wait"
'-------
'MAKES THE HALL MEASUREMENTS
'-------
```

```
CALL HALL(1, 2, 3, 4, AMPS, DLY!, vlim, VP1265!, VP2165!, VP1243!, VP2143!, AP1265!,
AP2165!, AP1243!, AP2143!, fail.flag%)
'-------
'WRITES THE DATA TO "DATA.RD1"
'-------
record:
     OPEN "DATA.RD1" FOR APPEND AS #4
     WRITE #4, AMPS, MG1!, VP1265!, VP2165!, VP1243!, VP2143!, AP1265!, AP2165!,
AP1243!, AP2143!
     CLOSE #4
'-------
'DEFAULTS ALL NEGATIVE VALUES TO 1E+22
'-------
VN1265! = 1E+22
VN2165! = 1E+22
VN1243! = 1E+22
VN2143! = 1E+22
AN1265! = 1E+22
AN2165! = 1E+22
AN1243! = 1E+22
AN2143! = 1E+22
MOB = 1E+22
N = 1E+22
'-------
'SETS THE NEGATIVE MAGNETIC FIELD
'-------
IF fail.flag% = 1 THEN GOTO recordn
     CLS
     LOCATE 10, 10: PRINT "Setting Magnetic Field To "; -MG!; " KG"
     'CALL forcemag(-MG!, MG2!)     ' comment out if no magnet controller
     MG2! = MG2! * 10000
     CLS
'-------
'MAKES THE HALL MEASUREMENTS
'-------
     LOCATE 10, 10: PRINT "Doing Calculations on Sample #"; I%; " in "; -MG!; " KG
field"
     LOCATE 11, 10: PRINT "Please Wait"
CALL HALL(1, 2, 3, 4, AMPS, DLY!, vlim, VN1265!, VN2165!, VN1243!, VN2143!, AN1265!,
AN2165!, AN1243!, AN2143!, fail.flag%)
'-------
'CALCULATE THE VALUES FOR THE HALL COEFFICIENT
'-------
     OPEN "DATA.RD1" FOR INPUT AS #4
     INPUT #4, THICK, TEMP!, AMPS, MG!, NAME$, ID$, WID!, SPAC!, V1246!, V2146!,
A1246!, A2146!, V1235!, V2135!, A1235!, A2135!, RHOA!, RHOB!, RHOAV!
     INPUT #4, AMPS, MG1!, VP1265!, VP2165!, VP1243!, VP2143!, AP1265!, AP2165!,
AP1243!, AP2143!
     HALFCTP = (2.5E+07 * THICK) / MG1!
     HALFCTN = (2.5E+07 * THICK) / ABS(MG2!)
     RHA! = HALFCTP * ((VP1265 / AP1265) - (VP2165 / -AP2165)) + HALFCTN * ((VN2165 /
-AN2165) - (VN1265 / AN1265))
     RHB! = HALFCTP * ((VP1243 / AP1243) - (VP2143 / -AP2143)) + HALFCTN * ((VN2143 /
-AN2143) - (VN1243 / AN1243))
'-------
'CALCULATES THE SAMPLE TYPE
'-------
     IF RHA < 0 THEN TYPE$ = "N" ELSE TYPE$ = "P"
     RHAV! = ABS((RHA + RHB) / 2)
'-------
' CALCULATE THE MOBILITY
'-------
     MOB = RHAV / RHOAV!
'-------
```

```
'CALCULATE THE CARRIER CONCENTRTION
'-------
     N = 6.25E+18 / RHAV
     CLS
'-------
'WRITES ALL THE DATA TO A FILE "DATA.RD2"
'-------
recordn:
     OPEN "DATA.RD2" FOR OUTPUT AS #5
     WRITE #5, MG2!, MOB!, N!, VN1265!, VN2165!, AN1265!, AN2165!, VN1243!, VN2143!,
AN1243!, AN2143!, RHA!, RHB!
     CLOSE #5
     CLOSE #4
END IF
'-------
'RESETS THE INSTRUMENTS
'-------
IF TEMP! <> 0 THEN CALL restemp    'COMMENT OUT IF NO Cryo
NOMAG: CALL devint
'-------
'DISPLAYS THE RESULTS
'-------
review: OPEN "DATA.RD1" FOR INPUT AS #4
INPUT #4, THICK, TEMP!, AMPS, MG!, NAME$, ID$, WID!, SPAC!, V1246!, V2146!, A1246!,
A2146!, V1235!, V2135!, A1235!, A2135!, RHOA!, RHOB!, RHOAV!
IF (MG! <> 0 AND MAG.TYPE <> 0) THEN
     INPUT #4, AMPS, MG1!, VP1265!, VP2165!, VP1243!, VP2143!, AP1265!, AP2165!,
AP1243!, AP2143!
     OPEN "DATA.RD2" FOR INPUT AS #5
     INPUT #5, MG2!, MOB!, N!, VN1265!, VN2165!, AN1265!, AN2165!, VN1243!, VN2143!,
AN1243!, AN2143!, RHA!, RHB!
END IF
CALL hlbresul(NAME$, ID$, THICK, SPAC!, WID!, TEMP!, AMPS, MG!, RHOAV!, RHAV!, MOB!,
N!, quit.flag%, TYPE$)
IF (quit.flag% = 1) THEN
     quit.flag% = 0
     CLOSE #4
     IF (MG! <> 0) THEN CLOSE #5
     GOTO filewr
END IF
CALL hbresu2(NAME$, ID$, THICK!, TEMP!, MG!, AMPS, WID!, SPAC!, V1246!, A1246, V2146!,
A2146!, RHOA!, V1235!, A1235!, V2135!, A2135!, RHOB!, quit.flag%)
IF (quit.flag% = 1) THEN
     CLOSE #4
     IF (MG! <> 0) THEN CLOSE #5
     quit.flag% = 0
     GOTO filewr
END IF
IF (MG! <> 0 AND MAG.TYPE <> 0) THEN
     CALL hbresu3(start.flag%, NAME$, ID$, THICK, TEMP!, MG!, AMPS, WID!, SPAC!,
VP1265!, AP1265!, VP2165!, AP2165!, VP1243!, AP1243!, VP2143!, AP2143!, VN1265!,
AN1265!, VN2165!, AN2165!, VN1243!, AN1243!, VN2143!, AN2143!, RHA!, RHB!)
     CLOSE #5
END IF
CLOSE #4
'-------
'RESULTS SENT TO OUTPUT FILE ID+".DAT"
'-------
filewr: OPEN "DATA.RD1" FOR INPUT AS #4
INPUT #4, THICK, TEMP!, AMPS, MG!, NAME$, ID$, WID!, SPAC!, V1246!, V2146!, A1246!,
A2146!, V1235!, V2135!, A1235!, A2135!, RHOA!, RHOB!, RHOAV!
IF (MG! <> 0 AND MAG.TYPE <> 0) THEN
     INPUT #4, AMPS, MG1!, VP1265!, VP2165!, VP1243!, VP2143!, AP1265!, AP2165!,
AP1243!, AP2143!
```

```
     OPEN "DATA.RD2" FOR INPUT AS #5
     INPUT #5, MG2!, MOB!, N!, VN1265!, VN2165!, AN1265!, AN2165!, VN1243!, VN2143!,
AN1243!, AN2143!, RHA!, RHB!
END IF
OPEN ID + ".DAT" FOR OUTPUT AS #6
CALL hlbfile(THICK, TEMP!, AMPS, MG, MG2!, MG1!, NAME$, ID$, RHOAV!, RHAV!, TYPE$,
MOB!, N!, WID!, SPAC!, V1246!, V2146!, RHOA!, A1246!, A2146!, V1235!, V2135!, A1235!,
A2135!, RHOB!, VP1265!, VP2165!, AP1265, AP2165!, VP1243!, VP2143!, AP1243!,
AP2143!, VN1265!, VN2165, AN1265, AN2165!, VN1243!, VN2143!, AN1243!, AN2143!, RHA!,
RHB!)
CLOSE #6
CLOSE #4
IF MG! <> 0 THEN CLOSE #5
     '************************************************
     '************************************************
     ' PLACE PRINT FILE TO A PRINTER STATEMENT HERE
     ' FILE NAME TO BE PRINTED IS ID$."DAT"
     '************************************************
     '************************************************

CALL final(start.flag%, review.flag%)
IF (start.flag% = 1) THEN
     start.flag% = 0
     GOTO start
END IF
IF (review.flag% = 1) THEN
     review.flag% = 0
     GOTO review
END IF

END

SUB final (start.flag%, review.flag%)
' Prototype for screen: LAST
'$INCLUDE: 'c:\s110\ps\LAST.inc'

sc.name$ = "LAST"
CALL sc.init(sc.name$)

Sc.Text% = -1
Sc.Clear% = 0
sc.output% = -1

CLS
CALL sc.msg("", 2, 0)
GOSUB LAST


Sc.Text% = 0

WHILE (NOT ((Sc.Pf% = 3) OR (Sc.Pf% = 7) OR (Sc.Pf% = 5)))
   GOSUB LAST
WEND

IF (Sc.Pf% = 3) THEN
     start.flag% = 1

END IF
IF (Sc.Pf% = 5) THEN
     review.flag% = 1

END IF
```

```
COLOR 7, 0
CLS


END SUB

SUB HALL (p1, p2, p3, p4, ival!, DLY!, vlim!, v1, v2, v3, v4, a1, a2, a3, a4,
fail.flag%)
'v1 = v1265  v2 = v2165 v3 = v1243  v4 = v2143
'i1 = i1265  i2 = i2165 i3 = i1243  i4 = i2143
fail.flag% = 0
'-------
'MAKES CONNECTION 12,65
'-------
hstart1: CALL connect(VMTR2H, p4)
CALL connect(VMTR2L, p3)
CALL forcei(ISRC1, ival!)
CALL delay(DLY!)
CALL measv(VMTR2, v1)
a1 = ival!
IF (v1 = 2E+22) THEN
     ival! = ival! / 10
     IF ABS(ival!) <= 1E-11 THEN
          v1 = 1E+22
          a1 = 1E+22
          v2 = 1E+22
          a2 = 1E+22
          fail.flag% = 1
          GOTO try2
     END IF
     GOTO hstart1
END IF
'-------
'MAKES CONNECTION 21,65
'-------
CALL forcei(ISRC1, -ival!)
CALL delay(DLY!)
CALL measv(VMTR2, v2)
a2 = -ival!
try2:
'-------
'MAKES CONNECTION 12,43
'-------
hstart2: CALL connect(VMTR2H, p2)
CALL connect(VMTR2L, p1)
CALL forcei(ISRC1, ival!)
CALL delay(DLY!)
CALL measv(VMTR2, v3)
a3 = ival!
IF (v3 = 2E+22) THEN
     ival! = ival! / 10
     IF ABS(ival!) <= 1E-11 THEN
          v3 = 1E+22
          a3 = 1E+22
          v4 = 1E+22
          a4 = 1E+22
          fail.flag% = 1
          GOTO try3
     END IF
     GOTO hstart2
END IF
'-------
'MAKES CONNECTION 21,43
'-------
```

```
CALL forcei(ISRC1, -ival!)
CALL delay(DLY!)
CALL measv(VMTR2, v4)
a4 = -ival!
try3:
END SUB

SUB hbgetkey (NAME$, ID$, HL$, TEMP!, THICK!, MG!, AMPS!, DLY!, QBNUM%, SPAC!, WID!)

' Prototype for screen: HBSCREEN
'$INCLUDE: 'C:\S110\PS\HBSCREEN.inc'
OPEN "HBGETKEY.DAT" FOR INPUT AS #1
INPUT #1, NAME$, ID$, HL$, TEMP!, THICK!, MG!, AMPS!, DLY!, QBNUM%, SPAC!, WID!
CLOSE #1
' Prototype for screen: SCREEN1

sc.name$ = "HBSCREEN"
CALL sc.init(sc.name$)

Sc.Text% = -1
Sc.Clear% = 0
sc.output% = -1

CLS
CALL sc.msg("F3=begin  F7= quit", 2, 0)
GOSUB HBSCREEN


Sc.Text% = 0

WHILE (NOT ((Sc.Pf% <> 3) OR (Sc.Pf% <> 7)))
   GOSUB HBSCREEN
WEND
IF Sc.Pf% = 7 THEN END


OPEN "HBGETKEY.DAT" FOR OUTPUT AS #1
WRITE #1, NAME$, ID$, HL$, TEMP!, THICK!, MG!, AMPS!, DLY!, QBNUM%, SPAC!, WID!
CLOSE #1

COLOR 7, 0
CLS

END SUB

SUB hbresu2 (NAME$, ID$, THICK!, TEMP!, MG!, AMPS, WID!, SPAC!, V1246!, A1246, V2146!,
A2146!, RHOA!, V1235!, A1235!, V2135!, A2135!, RHOB!, quit.flag%)
' Prototype for screen: RESULT2
'$INCLUDE: 'c:\s110\ps\HLBREST2.inc'

sc.name$ = "HLBREST2"
CALL sc.init(sc.name$)

Sc.Text% = -1
Sc.Clear% = 0
sc.output% = -1

CLS
CALL sc.msg(" F3=goto next menu  F7=quit", 2, 0)
GOSUB HLBREST2


Sc.Text% = 0
```

```
WHILE (NOT ((Sc.Pf% = 3) OR (Sc.Pf% = 7)))
   GOSUB HLBREST2
WEND

IF (Sc.Pf% = 7) THEN
     quit.flag% = 1
END IF



COLOR 7, 0
CLS



END SUB

SUB hbresu3 (start.flag%, NAME$, ID$, THICK, TEMP!, MG!, AMPS, WID!, SPAC!, VP1265!,
AP1265!, VP2165!, AP2165!, VP1243!, AP1243!, VP2143!, AP2143!, VN1265!, AN1265!,
VN2165!, AN2165!, VN1243!, AN1243!, VN2143!, AN2143!, RHA!, RHB!)
' Prototype for screen: RESULT3
'$INCLUDE: 'c:\s110\ps\HLBREST3.inc'

sc.name$ = "HLBREST3"
CALL sc.init(sc.name$)

Sc.Text% = -1
Sc.Clear% = 0
sc.output% = -1

CLS
CALL sc.msg("F3=goto next menu", 2, 0)
GOSUB HLBREST3


Sc.Text% = 0

WHILE (NOT ((Sc.Pf% = 3)))
   GOSUB HLBREST3
WEND



COLOR 7, 0
CLS
END SUB

 FUNCTION HLB (p1, p2, p3, p4, AMPS, DLY, vlim, v1, v2, v3, v4, a1, a2, a3, a4,
fail.flag%)
 fail.flag% = 0
'-------
'MAKES CONNECTION 12,46
'-------

start1: CALL connect(VMTR2H, p2)
CALL connect(VMTR2L, p4)
CALL forcei(ISRC1, AMPS!)
CALL delay(DLY!)
CALL measv(VMTR2, v1)
a1 = AMPS!
IF (v1 = 2E+22) THEN
     AMPS! = AMPS! / 10
     IF ABS(AMPS!) <= 1E-11 THEN
```

```
          a1 = 1E+22
          v1 = 1E+22
          a2 = 1E+22
          v2 = 1E+22
          fail.flag% = 1
          GOTO try4
     END IF
     GOTO start1
END IF
'-------
'MAKES CONNECTION 21,46
'-------
CALL forcei(ISRC1, -AMPS!)
CALL delay(DLY!)
CALL measv(VMTR2, v2)
a2 = -AMPS!
try4:
'-------
'MAKES CONNECTION 12,35
'-------
start2: CALL connect(VMTR2H, p1)
CALL connect(VMTR2L, p3)
CALL forcei(ISRC1, AMPS!)
CALL delay(DLY!)
CALL measv(VMTR2, v3)
a3 = AMPS!
IF (v3 = 2E+22) THEN
     AMPS! = AMPS! / 10
     IF ABS(AMPS!) <= 1E-11 THEN
          a1 = 1E+22
          v1 = 1E+22
          a2 = 1E+22
          v2 = 1E+22
          fail.flag% = 1
          GOTO try5
     END IF
     GOTO start2
END IF
'-------
'MAKES CONNECTION 21,35
'-------
CALL forcei(ISRC1, -AMPS!)
CALL delay(DLY!)
CALL measv(VMTR2, v4)
a4 = -AMPS!
'-------
'Resistivity calculations assuming:
'-------
IF fail.flag% = 1 GOTO try5
THICK! = 1!
SPAC! = 1!
WID! = 1!

RHOFCT = (WID! * THICK!) / SPAC!
RA = ((v1 / a1) - (v2 / -a2)) * .5
RHOA = RHOFCT * RA
RB = ((v3 / a3) - (v4 / -v4)) * .5
RHOB = RHOFCT * RB
'-------
'CALCULATE THE AVERAGE RHO
'-------
RHOAV = (RHOA + RHOB)
HLB = RHOAV
try5:
```

```
END FUNCTION

SUB hlbfile (THICK, TEMP!, AMPS, MG, MG2!, MG1!, NAME$, ID$, RHOAV!, RHAV!, TYPE$,
MOB!, N!, WID!, SPAC!, V1246!, V2146!, RHOA!, A1246!, A2146!, V1235!, V2135!, A1235!,
A2135!, RHOB!, VP1265!, VP2165!, AP1265, AP2165!, VP1243!, VP2143!, AP1243!, _
AP2143!, VN1265!, VN2165, AN1265, AN2165!, VN1243!, VN2143!, AN1243!, AN2143!, RHA!,
RHB!)
PRINT #6, " "
PRINT #6, TAB(10); "     date: "; DATE$; TAB(40); "Sample Thickness: "; THICK!; " cm"
PRINT #6, TAB(10); "     Time: "; TIME$; TAB(40); "     Temperature: "; TEMP!
PRINT #6, TAB(10); " Operator: "; NAME$; TAB(40); "    Test Current: "; AMPS!
PRINT #6, TAB(10); "Sample ID: "; ID$; TAB(37); " Magnetic Field(KG): "; MG!
PRINT #6, TAB(7); "Sample Width: "; WID!; " cm"; TAB(40); "Contact Spacing: "; SPAC!;
" cm"
PRINT #6, " "
PRINT #6, " "
PRINT #6, TAB(10); "              Resistivity (Ohm-cm): "; USING "##.####^^^^"; RHOAV!
PRINT #6, " "
PRINT #6, TAB(10); "Hall Coefficient (cm**3 - C**-1): "; USING "###.####^^^^"; RHAV!
PRINT #6, " "
PRINT #6, TAB(10); "              Sample Type (N or P): "; TYPE$
PRINT #6, " "
PRINT #6, TAB(10); "            Mobility (cm**2/V-sec): "; USING "###.####^^^^"; MOB!
PRINT #6, " "
PRINT #6, TAB(10); "  Carrier Concentration (cm**-3): "; USING "###.####^^^^"; N!
PRINT #6, CHR$(12) ' form feed
PRINT #6, " "
PRINT #6, TAB(30); "RESISTIVITY"
PRINT #6, " "
PRINT #6, TAB(5); "V12,46 = "; V1246!; TAB(30); "V21,46 = "; V2146!; TAB(55); "   RHOA
= "; USING "###.####^^^^"; RHOA!
PRINT #6, TAB(5); "I12,46 = "; A1246!; TAB(30); "I21,46 = "; A2146!
PRINT #6, " "
PRINT #6, TAB(5); "V12,35 = "; V1235!; TAB(30); "V21,35 = "; V2135!; TAB(55); "   RHOB
= "; USING "###.####^^^^"; RHOB!
PRINT #6, TAB(5); "I12,35 = "; A1235!; TAB(30); "I21,35 = "; A2135!
PRINT #6, " "
PRINT #6, TAB(30); "HALL COEFFICIENT"
PRINT #6, " "
PRINT #6, TAB(5); "    +MG = "; MG1!; TAB(30); "V12,65 = "; VP1265!; TAB(55); "V21,65
= "; VP2165
PRINT #6, TAB(30); "I12,65 = "; AP1265!; TAB(55); "I21,65 = "; AP2165
PRINT #6, TAB(30); "V12,43 = "; VP1243!; TAB(55); "V21,43 = "; VP2143
PRINT #6, TAB(30); "I12,43 = "; AP1243!; TAB(55); "I21,43 = "; AP2143
PRINT #6, " "
PRINT #6, TAB(5); "    -MG = "; MG2!; TAB(30); "V12,65 = "; VN1265!; TAB(55); "V21,65
= "; VN2165
PRINT #6, TAB(30); "I12,65 = "; AN1265!; TAB(55); "I21,65 = "; AN2165
PRINT #6, TAB(30); "V12,43 = "; VN1243!; TAB(55); "V21,43 = "; VN2143
PRINT #6, TAB(30); "I12,43 = "; AN1243!; TAB(55); "I21,43 = "; AN2143
PRINT #6, " "
PRINT #6, TAB(30); "   RHA = "; USING "###.####^^^^"; RHA!
PRINT #6, TAB(30); "   RHB = "; USING "###.####^^^^"; RHB!

END SUB

SUB hlbresul (NAME$, ID$, THICK!, SPAC!, WID!, TEMP!, AMPS!, MG!, RHOAV!, RHAV!, MOB!,
N!, quit.flag%, TYPE$)
' Prototype for screen: RESULT1
'$INCLUDE: 'c:\s110\ps\HLBREST1.inc'

sc.name$ = "HLBREST1"
CALL sc.init(sc.name$)
```

```
Sc.Text% = -1
Sc.Clear% = 0
sc.output% = -1

CLS
CALL sc.msg(" F3=goto next menu  F7=quit", 2, 0)
GOSUB HLBREST1


Sc.Text% = 0

WHILE (NOT ((Sc.Pf% = 3) OR (Sc.Pf% = 7)))
   GOSUB HLBREST1
WEND

IF (Sc.Pf% = 7) THEN
     quit.flag% = 1
END IF


COLOR 7, 0
CLS


END SUB
```

This page is left blank intentionally

APPENDIX D

D.1 Software Installation Guide

The software required to use the Hall System is:

- Microsoft ® Quick Basic 4.5
- Microsoft ®Quick C 2.0
- CEC card software 2.05
- Pro Screen 4.5e
- Geograf 4.0
- Dos 3.3
- Hall software 2.0

D.2 Installing The Software

The software should be installed onto a hard drive. By default this example will show how to install the Hall System software onto drive **C:**.
The first step is to install DOS onto the computer. Refer to the DOS installation guide for more details.

Now install Quick Basic 4.5. This is done by following the Quick Basic installation instructions. The setup disk should be placed in drive **A:**, then type:

A: setup<CR>

At this point a menu will appear, and the full menu setup should be used. All of the Quick Basic default settings and paths should be used.

After successfully installing Quick Basic there should be a **QB45 <DIR>** on the **C:** disk.

Install the Quick C software by following the Quick C installation manual.

NOTE: When installing Quick C,make sure that the medium compiling module is loaded **WITH** the graphics option. This will create the **MLIBCE.LIB** file. If in doubt, load all the compiling modules.

It is now time to install Pro Screen and GEOGRAF. This is accomplished by making a **PS <DIR>** and a **GEOGRAF <DIR>**. These are made as follows:

C: cd\ <CR>

C: md ps <CR>
C: md geograf <CR>
C: cd ps <CR>

This will create the two directories and place you in the **PS <DIR>**. Now place the first Pro Screen disk into drive **A:**. To load the disk into the hard drive, type:

C: copy a:*.* <CR>

This will load the contents of disk 1 into the **PS** directory. Repeat the same steps for the second disk. To load the Geograf software type:

C: cd\ <CR>
C: cd geograf <CR>

This will put you in the **GEOGRAF** directory. Now place the first disk into drive **A:**, and type:

C: copy a:*.* <CR>

This will copy the contents of the first disk into the **GEOGRAF** directory. Repeat these steps for the second Geograf disk.

It is now time to make the directories for the Hall software. On the **C** drive, create a directory called **S110**. This is done by typing:

C: cd\ <CR>
C: md s110 <CR>
C: cd s110 <CR>

These steps will put you into the **S110** directory. Now four subdirectories must be created. These are created by typing:

C: md sources <CR>
C: md lib <CR>
C: md obj <CR>
C: md ps <PS>

To load the sources for the Hall software place distribution disk1 with the sources on it into drive **A**. Then type:

C: copy a:\s110\sources*.* c:\s110\sources*.* <CR>

To load the object codes for the Hall software place distribution disk1 with the objects on it into drive **A**. Then type:

C: copy a:\s110\obj*.* c:\s110\obj*.* <CR>

To load the libraries for the Hall software place distribution disk2 with the libraries on it into drive **A**. Then type:

C: copy a:\s110\lib*.* c:\s110\lib*.* <CR>

To load the menus for the Hall software place distribution disk2 with the Pro screen on it into drive **A**. Then type:

C: copy a:\s110\ps*.* c:\s110\ps*.* <CR>

Now load the Quick C **MLIBCE.LIB** file. This file should be placed in the S110 **LIB** subdirectory. This is done by copying the Quick C file called **MLIBCE.LIB** from the QC2 LIB directory into the S110 LIB directory. This is accomplished by typing:

C: copy c:\qc2\lib\MLIBCE.LIB c:\s110\lib*.*

The last file needed is from the CEC software. The name of this file is **IEEE488.LIB**. This file should be placed in the S110 **LIB** subdirectory. This is accomplished by placing the CEC disk into drive **A**. Then type:

C: copy a:ieee488.lib c:\s110\lib <CR>

Note: both of the libraries loaded above are also included on the Hall Software distribution diskette.

Now all of the Hall Software has been installed. The **AUTOEXEC.BAT** file must now be changed for the software to run properly. An example of a changed **AUTOEXEC.BAT** is:

```
path=c:\;c:\QB45;c:\QC2\bin;c:\dos;c:\s110\ps;c:\s110\lib;c:\ps;c:\geograf
set lib=c:\qb45;c:\s110\lib
prompt $p$g
vol
ver
```

NOTE: In the **path** of the **AUTOEXEC.BAT** file, QB45 **MUST** be listed **BEFORE** QC2 **AND** the DOS. The reason for this is that the Quick BASIC 4.5 linker **MUST** be used, **NOT** the Quick C linker **OR** the DOS linker.

To compile all the S110 programs into executable files and place them into the S110 SOURCES directory type:

C: cd\s110\sources <CR>
C: MAKEALL.BAT <CR>

This concludes the installation process.

APPENDIX E

E.1 Installing Plot

Before you can use **PLOT** for the first time, you must define what type of graphics card and what type of printer you are using. This is done by getting into the **GEOGRAF** directory by typing:

C: cd / <CR>
C: cd geograf <CR>

typing:

C: drivers <CR>

you will execute the program that will allow you to define which graphics card and which printer you are using. In order for you to use the **PLOT** program provided you **MUST** define **BOTH** the graphics card and the printer even if you do not intend to print your plot.
When you run **DRIVERS** a main menu will appear. Follow the instructions on the screen and choose **GRAPHICS CARD** by moving the cursor onto **GRAPHICS CARDS** and pressing F1 to select.

The next screen will allow you to choose the manufacturer of your GRAPHICS card. Again move the cursor to the correct manufacturer and press F1 to select.

The third screen will allow you to choose which model and mode desired. Again move the cursor to the correct choice and press F1.

NOTE: It is important to choose the correct graphics card since this tells the program how to display the plot.

At this point a color menu will appear on the screen. Choose F4 to quit. If you look at the top of the screen, the display says it will save your choices in a file called "SCREEN.DRV". Do **NOT** change this file name, press F1 to save.

Once again the main menu will appear, this time move the cursor to **PRINTERS** and press F1.

The second screen is a list of all the manufacturers of printers supported by GEOGRAF, move the cursor to the appropriate one and press F1.

NOTE: If you do **NOT** intend to use a printer, you must still create a driver for the printer so that the **PLOT** program will run. Therefore choose any name.

The following screen allows you to choose the model and mode of the printer. Again make the appropriate choice and strike F1.

The third screen is the color table screen, strike F4 to quit. If you look at the top of the screen, the display says it will save your choices in a file called "PRINTER.DRV". Do **NOT** change this file name, press F1 to save.

This concludes the setup procedure needed to use the **PLOT** program. Refer to the GEOGRAF manual for more information on setting up the drivers.

If you run plot and the graph on the screen is distorted, re-run this procedure to make sure that your drivers where entered correctly.

APPENDIX F

System Options

F.1 Magnet Field Level Controller

This section is a listing of the legal calls for the magnet controller and a detail description of their usage.

NOTE: When you are using the S110 software, you **MUST** have the gaussmeter set on the **1K** range in order for the magnet controller to operate properly.

F.1.1 ACON

ACON is called to turn on the ac power to the magnet. This must be the first call made to the magnet. This routine has already been included in the **FORCEMAG** routine, so the user does not have to call it explicitly. This call closes relay 93 on the scanner.

The calling sequence for this routine is:

CALL ACON

Programming Considerations

- This routine must be the first routine sent to the magnet unless the **FORCEMAG** routine is being used.

ERROR Displays

- If the string sent to the instrument is received improperly.

F.1.2 ACOFF

ACOFF should be the last routine sent to the magnet. This routine turns off the AC power to the magnet. This call opens relay 93 on the scanner.

The calling sequence for this routine is:

CALL ACOFF

Programming Considerations

- **ACOFF** should not be called until you are finished using the magnet.

ERROR Displays

- If strings are received improperly

F.1.3 DCON

DCON should be used after the **ACON** routine. This routine turns on the DC power to the magnet. This routine is already included in the **FORCEMAG** routine. This routine pulses relay 92 of the scanner.

The calling sequence for this routine is:

CALL DCON

Programming Considerations

- Must be used after the **ACON** routine.

ERROR Displays

- If the strings are received improperly.

F.1.4 DCOFF

DCOFF should be called before the **ACOFF** routine. This routine turns off the DC power to the magnet. This call pulses relay 91 in the scanner.

The calling sequence for this routine is:

CALL **DCOFF**

Programming Considerations

- Should be used before the **ACOFF** routine

F.1.5 FORCEMAG

The **FORCEMAG** routine sets the magnet to the desired strength in Kilogauss. This routine first turns on the AC power and then the DC power. It then checks to make sure that the polarity desired and the present polarity of the magnet are the same. If it is not, the magnet will reverse the polarity. The routine then sets the magnetic field. The magnetic field is monitored using the 196 voltmeter to assure that the desired magnetic field strength has been achieved. The Walker Gauss Meter **MUST BE SET TO THE 1K range.** If this is not done, the magnet will not be set properly. The magnet will display an error if the user tries to force more than 100% of full scale on the magnet controller, or more than 10 volts on the voltage source which is used to set the magnet.

The calling sequence for this routine is:

CALL **FORCEMAG** (MGWANT, MGSET)

Where

Name	Type	Operation	Description
Mgwant	Single	Input	The setting of the magnet (KG) desired.
Mgset	Single	Output	The setting at which the magnet ended up (KG).

Programming Considerations

- This routine turns on both the AC and DC power. The routine will automatically reverse the magnet if the polarity is wrong.
- The magnet will stay set until the **DCOFF** routine is called.

ERROR Displays

- If more than 10 volts is forced.

- if a string is not received properly

F.1.6 REVERSE

REVERSE is used to reverse the polarity of the magnet. There is a 30 second wait built into this routine so that the magnet has enough time to change its polarity. This routine does not have to be called if the **FORCEMAG** routine is being used, since it is built into that routine. Refer to the **FORCEMAG** routine for more details. This call pulses relay 94 in the scanner.

The calling sequence for this routine is:

CALL REVERSE

Programming Considerations

- Only used to change the polarity of the magnet. Does not have to be used if the **FORCEMAG** routine is being used.

ERROR Displays

- If strings are not received properly.

F.2 Magnet ON/OFF/REVERSE

This magnet controller option is used if you have a magnet without the ability to set the level of the field through the use of the 617 and the 196. Instead this magnet controller will turn on/off the magnet and reverse the magnet. All the routines that are used in the **Magnet Field Level Controller** are still valid. The only difference in the performance of the routines occurs in the **FORCEMAG** routine. This routine **WILL NOT** set the magnet level. Instead this routine will prompt the user to set the desired field and enter the setting that was reached. This is done so that the calculations will be done using the correct value for the strength of the magnetic field. Refer to the **Magnet Field Level Controller** in section F.1 of this appendix for details on the parameters of these routines.

F.3 Fixed Magnet Controller

The routines that are used in the **Magnet Field Level Controller** option are all valid routines in this controller. The only calls that should be used are the **FORCEMAG** and the **REVERSE** routines. **ALL OTHER ROUTINES HAVE NO EFFECTS ON THE SYSTEM OR THE MAGNET.** This controller option should be used if the user has a fixed magnet.

F.3.1 FORCEMAG Routine

The **FORCEMAG** routine will prompt the user to place the sample into the magnetic field. The routine will **NOT** prompt you to enter the magnetic field at this point. It will instead use the value that was entered at the beginning of the S110 program. Refer to the **Magnet Level Controller** in section F.1 of this appendix for details on the parameter list of these routines.

F.4 Temperature Controller

F.4.1 Palm Beach Model 4075

F.4.1.1 Introduction

This temperature controller option allows programming and temperature readback of the Palm Beach Model 4075 temperature controller. The 4075 is interfaced to the system using the IEEE-488 instrument interface bus. This option consists of a 4075 unit, mounting of the unit in the system rack cabinet, and a test program to exercise the temperature controller.

F.4.1.2 Connecting to the Controller

If the customer specifies, the cryo system may be integrated into the system as part of a "turn-key" installation. Otherwise, it is the customer's responsibility to supply the necessary sensors and to connect them to the temperature controller, Selection of the sensors is discussed in the controller's instruction manual. Care should be taken to route the sensor cables away from any high-level signals, such as ac power lines, to avoid introducing noise into the sensor cables.
Depending upon the selection of the sensor, it may be necessary to modify the default programming string of the 4075 or to program the 4075 to use the proper curve-fit routine in each applications program using the controller.

F.4.1.3 Device Driver Interface

The 4075 instrument is initialized to the following state:

Temp	= 10 degrees K
Integral	= 35
Rate	= 0
Gain	= 1000
Amp error	= 9

The user is referred to the 4075 instrument manual for the meaning of these states. The heater is controlled by the **SETTEMP** routine. The heater is set on the medium level when the controller is set to values less than 25 degrees K. The heater is set on the high level for values greater than 25 degrees K.

There are also routines that set the temperature, read the temperature, set the integral, set the rate, set the gain, and set the error amplifier. These are

- SETTEMP
- READTEMP
- SETRESET
- SETRATE
- SETGAIN
- ERRORAMP

The **SETTEMP** routine is used to set the temperature on the cryo controller. This routine will then return the final temperature read. This routine will except values between 0 and 799.9 degrees K. If you enter an invalid temperature the routine will prompt you for a valid temperature.

The calling sequence for this routine is:

Call **SETTEMP** (WANTSET, GETSET);

The temperature setting **MUST** be in degrees Kelvin. Check the cryo manual for legal values. If zero is entered as the temperature then the cryo will not be set.

The calling sequence for this routine is as follows:

CALL **SETTEMP** (WANTSET, GETSET)

Where

Name	Type	Operation	Description
Wantset	Single	Input	Temperature desired, in degrees Kelvin.
Getset	Single	Output	Temperature achieved by the controller, degrees Kelvin.

Programming Considerations

- Watch that the values entered are in degrees Kelvin and between zero and 799.9 degrees Kelvin.

ERROR Displays

- There are no ERROR messages.

The **TEMPERROR** routine is used prior to the **SETTEMP** routine. This routine should not be confused with the **SETERAMP** routine. This routine allows you to set the tolerance error allowed. For example, if the initial temperature of the cryo is 300 degrees Kelvin and you want to set the cryo to 200 degrees Kelvin, with a 3% error in the temperature setting, the calling sequence for this routine would be as follows:

```
CALL  TSTSEL(1)
        .
        .
        .
        .
CALL TEMPERROR(3.0)
CALL SETTEMP(200, GETSET)
```

The example above uses **TSTSEL** to initialize the bus. Refer to the **TSTSEL** routine for details. **TEMPERROR** is used to set the error to 3%.

NOTE: If **TEMPERROR** is **NOT** called, then the error percent is defaulted to 2%.

SETTEMP will set the temp to 200 degrees Kelvin, But it will exit out of the routine when the temperature is within 3% of 200 degrees Kelvin. The cryo controller will continue setting the temperature to 200 degrees Kelvin.

Note: There will be some oscillation of the temperature around the set temperature desired. Refer to the cryo controller manual for details on how the temperature is set by the cryo controller.

The calling sequence for this routine is:

CALL TEMPERROR(Perc)

Where:

Name	Type	Operation	Description
Perc	Single	Input	Error percentage allowed when setting the temperature before the SETTEMP routine will return a value and continue with the rest of the program.

Programming Considerations

- Call this routine before the **SETTEMP** routine.
- If this routine is not called, then the percent error is defaulted to 2%.

ERROR Displays

- There are no error messages built into this routine.

The **SETGAIN** routine is used to set the gain of the Palm Beach controller. Refer to the cryo manual on details on the gain routine. This routine will except values between 1 and 1023. If an invalid number is received by this routine, you will be prompted to enter a valid number.

The calling sequence for this routine is:

CALL SETGAIN(GAIN)

Where

Name	Type	Operation	Description
Gain	Integer	Input	The value the gain should be set to.

Programming Considerations

- The gain has a default setting of 1000. This routine should only be called if you wish to change the default setting.

ERROR Displays

- There are no error messages

The **SETRATE** routine is used to set the gain of the Palm Beach controller. Refer to the cryo manual on details on the rate routine. This routine will except values between 1 and 127. If an invalid number is received by this routine, you will be prompted to enter a valid number.

The calling sequence for this routine is:

CALL SETRATE(RATE)

Where

Name	Type	Operation	Description
Rate	Integer	Input	The value the rate should be set to.

Programming Considerations

- The rate has a default setting of 0. This routine should only be called if you wish to change the default setting.

ERROR Displays

- There are no error messages

The **SETRESET** routine is used to set the reset or integral of the Palm Beach controller. Refer to the cryo manual on details on the reset routine. This routine will except values between 1 and 127. If an invalid number is received by this routine, you will be prompted to enter a valid number.

The calling sequence for this routine is:

CALL **SETRESET**(RESET)

Where

Name	Type	Operation	Description
Reset	Integer	Input	The value the reset should be set to.

Programming Considerations

- The reset has a default setting of 35. This routine should only be called if you wish to change the default setting.

ERROR Displays

- There are no error messages

The **SETERAMP** routine is used to set the error amplifier of the Palm Beach controller. Refer to the cryo manual on details on the error amplifier routine. This routine will except values between 1 and 11. If an invalid number is received by this routine, you will be prompted to enter a valid number.

The calling sequence for this routine is:

CALL **SETERAMP**(ERR)

Where

Name	Type	Operation	Description
Err	Integer	Input	The value the error amplifier should be set to.

Programming Considerations

- The error amplifier has a default setting of 9. This routine should only be called if you wish to change the default setting.

ERROR Displays

- There are no error messages

The **READTEMP** routine is used to read the temperature that the cryo is currently set to.

The calling sequence for this routine is:

CALL **READTEMP**(TEMP)

Where

Name	Type	Operation	Description
Temp	Single	Output	The temperature that the cryo is currently reading.

Programming Considerations

- This routine is used only if you want to read what temperature the cryo is currently reading.

ERROR Displays

- There are no error messages

F.4.1.4 TSTCRYO

TSTCRYO is a test program that will test the 4075 cryo controller. The program can be seen in the **TSTCRYO.BAS** program. The executable program is called **TSTCRYO.EXE**. This program also serves as an example of string programming of the controller.

NOTE: Before using the 4075 cryo controller **READ** the manual provided with the 4075 cryo system for safety precautions.

While the test is in progress, observe the front panel of the 4075 to verify that the programmed values agree with the displayed values on the terminals. The steps of the program are as follows:

1) Devices and bus are initialized
2) Temperature read (degrees Kelvin)
3) Temperature set to 250 degrees, Kelvin
4) Integral set to 15
5) Rate set to 00

6) Gain set to 850
7) Error Amplifier set to range 5
8) Temperature read
9) Devices and bus are cleared.

F.4.2 MMR Temperature Controller K-20

F.4.2.1 Introduction

This temperature controller option allows programming and temperature readback of the MMR K-20 temperature controller. The K-20 is interfaced to the system using the IEEE-488 instrument interface bus. The K-20 option consists of a K-20 unit, mounting of the unit in the system rack cabinet, K-20 software driver, and a test program to exercise the temperature controller.

F.4.2.2 Connecting to the Controller

If the customer so specifies, the cryo system may be integrated into the system as part of a "turn-key" installation. Otherwise, it is the customer's responsibility to supply the necessary sensors and to connect them to the temperature controller. Selection and connection of the sensors is discussed in the controller's instruction manual. Care should be taken to route the sensor cables away from any high-level signals, such as ac power lines, to avoid introducing noise into the sensor cables.

F.4.2.3 Device Driver Interface

These are the routines that can be used by the K-20 driver

- settemp (sets the temperature)
- temperror

The **SETTEMP** routine lets the user input a desired temperature setting. Then it returns the final temperature read. This routine makes use of the **SK NNN.NN (SET TEMP)** and the **TEMP** routines that can be found in the K-20 owner's manual.

The calling sequence for this routine is as follows:

CALL **SETTEMP** (WANTSET, GETSET)

Where

Name	Type	Operation	Description
Wantset	Single	Input	Temperature desired, in degrees Kelvin.
Getset	Single	Output	Temperature achieved by the controller, degrees Kelvin.

Programming Considerations

- Watch that the values entered are in degrees Kelvin and between zero and 409.00 degrees Kelvin.

ERROR Displays

- Refer to the K-20 owner's manual for a description on the Error messages that can be displayed.

The **TEMPERROR** routine is used prior to the **SETTEMP** routine. This routine allows you to set the tolerance error allowed. For example, if the initial temperature of the cryo is 300 degrees Kelvin and you want to set the cryo to 200 degrees Kelvin, with a 3% error in the temperature setting, the calling sequence for this routine would be as follows:

```
CALL  TSTSEL(1)
        .
        .
        .
        .
CALL TEMPERROR(3.0)
CALL SETTEMP(200, GETSET)
```

The example above uses **TSTSEL** to initialize the bus. Refer to the **TSTSEL** routine for details. **TEMPERROR** is used to set the error to 3%.

NOTE: If **TEMPERROR** is **NOT** called, then the error percent is defaulted to 2%.

SETTEMP will set the temp to 200 degrees Kelvin, But it will exit out of the routine when the temperature is within 3% of 200 degrees Kelvin. The cryo controller will continue setting the temperature to 200 degrees Kelvin.

Note: There will be some oscillation of the temperature around the set temperature desired. Refer to the cryo controller manual for details on how the temperature is set by the cryo controller.

The calling sequence for this routine is:

CALL **TEMPERROR**(Perc)

Where:

Name	Type	Operation	Description
Perc	Single	Input	Error percentage allowed when setting the temperature before the SETTEMP routine will return a value and continue with the rest of the program.

Programming Considerations

- Call this routine before the **SETTEMP** routine.
- If this routine is not called, then the percent error is defaulted to 2%.

ERROR Displays

- There are no error messages built into this routine.

F.4.2.4 TSTCRYO

TSTCRYO program tests the 91C cryo system. The listing of the program can be seen in **TSTCRYO.BAS**, the executable program is called **TSTCRYO.EXE**. This program will prompt the user to enter a desired temperature. The program will then set the cryo to the temperature desired and return the temperature reached.

F.4.3 Lakeshore Cryo Controller 91C

F.4.3.1 Introduction

This temperature controller option allows programming and temperature readback of the Lakeshore temperature controller. The 91C is interfaced to the system using the IEEE-488 instrument interface bus. The 91C option consists of a 91C unit, mounting of the unit in the system rack cabinet, modifying the software device driver, and a test program to exercise the temperature controller.

NOTE: If you are using the MMR controller, you will not be able to run your S110 programs from within the QuickBASIC environment. You **MUST** use the **MAKEIT.BAT** file to create an executable program.

F.4.3.2 Connecting to the Controller

If the customer specifies, the cryo system may be integrated into the system as part of a "turn-key" installation. Otherwise, it is the customer's responsibility to supply the necessary sensors and to connect them to the temperature controller. Selection and connection of the sensors is discussed in the controller's instruction manual. Care should be taken to route the sensor cables away from any high-level signals, such as ac power lines, to avoid introducing noise into the sensor cables.

F.4.3.3 Device Driver Interface

These are the routines that can be used by the 91C driver

- settemp (sets the temperature)
- temperror

SETTEMP routine lets the user input a desired temperature setting and a returned value of what the temperature got to is returned. This routine makes use of the **S NNN.NN (SET TEMP)** and the **TEMP** routines that can be found in the 91C owner's manual.

The calling sequence for this routine is as follows:

CALL **SETTEMP** (WANTSET, GETSET)

Where

Name	Type	Operation	Description
Wantset	Single	Input	Temperature desired, in degrees Kelvin.
Getset	Single	Output	Temperature achieved by the controller, degrees Kelvin.

Programming Considerations

- Watch that the values entered are in degrees Kelvin and between zero and 409.00 degrees Kelvin.

ERROR Displays

- Refer to the 91C owner's manual for a description on what Error messages can be displayed.

TEMPERROR routine is used prior to the **SETTEMP** routine. This routine allows you to set the tolerance error allowed. For example if the initial temperature of the cryo is at 300 degrees Kelvin. You want to set the cryo to 200 degrees Kelvin, but you can allow a 3% error in the temperature setting.The calling sequence for this routine would be as follows:

```
CALL TSTSEL(1)
         .
         .
         .
         .
CALL TEMPERROR(3.0)
CALL SETTEMP(200, GETSET)
```

The example above uses **TSTSEL** to initialize the bus. Refer to the **TSTSEL** routine for details. **TEMPERROR** is used to set the error to 3%.

NOTE: If **TEMPERROR** is **NOT** called, then the error percent is defaulted to 2%.

SETTEMP will set the temp to 200 degrees Kelvin, But it will exit out of the routine when the temperature is within 3% of 200 degrees Kelvin. The cryo controller will continue setting the temperature to 200 degrees Kelvin.

Note: There will be some oscillation of the temperature around the set temperature desired. Refer to the cryo controller manual for details on how the temperature is set by the cryo controller.

The calling sequence for this routine is:

CALL **TEMPERROR**(Perc)

Where:

Name	Type	Operation	Description
Perc	Single	Input	Error percentage allowed when setting the temperature before the SETTEMP routine will return a value and continue with the rest of the program.

Programming Considerations

- Use this routine before the **SETTEMP** routine.

- If this routine is not called, then the percent error is defaulted to 2%.

ERROR Displays

- There are no error messages built into this routine.

F.4.2.4 TSTCRYO

TSTCRYO program tests the 91C cryo system. The listing of the program can be seen in **TSTCRYO.BAS**, the executable program is called **TSTCRYO.EXE**. This program will prompt the user to enter a desired temperature. The program will then set the cryo to the temperature desired and return the temperature reached.

APPENDIX G

System Specifications

G.1 General and Environmental Specifications

G.1.1 Controller Specifications

Controller Type:	IBM PC, PC-XT, PC-AT Hard Disc (20 Mb)
Operating System	IBM PC-DOS or Microsoft ® MS-DOS, version 3.3
Memory Requirements:	512 KBytes RAM minimum
Instrument Interface:	Capital Equipment Corporation

G.1.2 Instrument Specifications

Cabinet Dimensions:	68.4"*H* x 22"*W* x 38.2" *D*
Minimum Clear Space:	12 inches from front, top and back unit
Mains Power:	110-125 VAC, 4 Amperes typical, 15 Amperes maximum
Temperature range:	18 to 28 degrees Celsius
Relative humidity:	30% to 60% R.H., non-condensing
Warm up time:	2 hours to rated accuracy
Accuracy:	1 year

G.2 Model 1101-HLA Quad Electrometer Buffer Amplifier

High Resistivity Mode

Instrumentation:	Model 220 ISRC Model 617 IMTR Model 196 VMTR Model 706 Scanner Model 1101-HLA Hall Module
DUT Input Impedance:	Greater than 100 Teraohms in parallel with 4 pF*
Output Resistance: (Voltmeter Port)	Less than 5. ohms (1. milliamp max. current)
Amplifier noise:	Less than 15 mV. P-P, .1 to 10 Hz. Constant ambient temperature, Measured at Voltmeter Port with High Inputs shorted to System Common
Amplifier Gain:	+1.00; +0, -30 ppm
Amplifier Offset:	+-50 mV (Adjustable)
Amplifier CMRR:	80 dB minimum
Input voltage operating range:	+-10. VDC
Maximum input overload:	+-120 Volts DC
Amplifier bias current:	+-1.5E-13 Amps maximum

Low Resistivity Mode

Input Resistance:	1 Gigaohm with System Voltmeter in 0.2, 2.0 or 20 volt ranges. Input resistance governed by System Voltmeter.
Input to Output Resistance:	Less than 5 Ohms
Noise:	Less than 50 nV Peak-to-Peak from 0.1 to 10 Hz at a constant ambient temperature.
Input voltage operating range:	+-10 VDC
Maximum Input Overload:	+-120 VDC

*- Minimum input resistance is 10 Teraohms when used in the S110 System at 28 degrees Celsius and 60% R.H.

G.3 Van der Pauw Measurement Specifications

High Terminals

Range (Ohms)	Resolution (Ohms)	Current (Amps)	V/I Accuracy +-% Reading	V/I Accuracy +-% Range	Delay (sec)
100	0.1 m	10 m	0.2	0.02	0.02
1 K	1 m	1 m	0.2	0.03	0.02
10 K	10 m	100 µ	0.2	0.02	0.02
100 K	100 m	10 µ	0.2	0.02	0.02
1 M	1	1 µ	0.2	0.03	0.02
10 M	10	100 n	0.4	0.02	0.20
100 M	100	10 n	0.4	0.02	0.20
1 G	1 K	1 n	0.4	0.04	2.0
10 G	10 K	100 p	3.0	0.04	20.0
100 G	100 K	10 p	5.0	0.20	40.

Low Terminals

Range (Ohms)	Resolution (Ohms)	Current (Amps)	V/I Accuracy +-% Reading	V/I Accuracy +-% Range	Delay (sec)
100 m *	1 µ	100 m	0.2	0.02	0.02
1 *	10 µ	10 m	0.1	0.02	0.02
10 *	10 µ	10 m	0.1	0.003	0.02
100	100 µ	10 m	0.1	0.002	0.02
1 K	1 m	1 m	0.1	0.002	0.02
10 K	10 m	100 µ	0.1	0.002	0.02

Accuracy based on average reading from positive and negative current.

Ambient temperature variation during measurement: 0.5 degrees Celsius maximum.

Voltmeter: 2 volt range (except * - 0.2 volt range).

Delay: Delay between forcing current and measuring voltage in seconds.

G.4 Hall-Bridge Measurement Specifications

High Terminals

Sample Resistance	Voltage Accuracy +-% Reading	Voltage Accuracy +-% Range	Delay (sec)
100	0.02	0.1	0.02
1 K	0.02	0.1	0.02
10 K	0.02	0.1	0.02
100 K	0.02	0.1	0.02
1 M	0.02	0.1	0.02
10 M	0.02	0.1	0.20
100 M	0.02	0.1	0.20
1 G	0.04	0.2	2.0
10 G	0.30	0.2	20.
100 G	2.1	0.8	40.

Low Terminals

Sample Resistance	Voltage Accuracy +-% Reading	Voltage Accuracy +-% Reading	Delay (sec)
100 m	0.02	0.02	0.02
1	0.02	0.02	0.02
10	0.02	0.002	0.02
100	0.02	0.001	0.02
1 K	0.02	0.001	0.02
10 K	0.02	0.001	0.02

Accuracy based on average differential voltage reading with forced positive and negative current. Measured voltage excludes current source errors.

Ambient temperature variation during measurement: 0.5 degrees Celsius maximum.

Voltmeter: 0.2 volt range.

Delay: Delay between forcing current and measuring voltage in seconds.

Maximum voltage across sample: 1 Volt.

APPENDIX H

H.1 ERROR Messages

This appendix gives a listing of all the error messages that can occur within each routine. There is also a list of possible causes for each error.

NOTE: Not all routines are listed in this appendix. Only the routines that have error messages in them.

H.1.1 CONNECT Routine

- ERROR - CONNECT - Pin number is (#) it must be < or = to 4

One of the connect routines is trying to connect a device to pin (#). Check the **CONNECT** routines.

- ERROR: Illegal connection with IMTR1 at pin (#)

You are trying to connect the IMTR1 to the same pin as the ISRC at pin number (#). Check **CONNECT** routines.

- ERROR - CONNECT - device not found in configuration
device = (#)

You are trying to connect a device that does not exists. Check the **CONNECT** routine's device name.

- ERROR - CONNECTH_196: device 196H already connected

You are attempting to connect the high side of the 196 more than once. Check the **CONNECT** routine to see if you are trying to connect the 196 high side more than once BEFORE the devices are disconnected. Refer to the **CONNECT** routine for details.

- ERROR - CONNECTL_196: device 196L already connected

You are attempting to connect the low side of the 196 more than once. Check the **CONNECT** routine to see if you are trying to connect the 196 low side more than once BEFORE the devices are disconnected. Refer to the **CONNECT** routine for details.

- ERROR - CONNECT_220: Current source already connected.

You are attempting to connect the 220 more than once. Check the **CONNECT** routine to see if you are trying to connect the 220 more than once BEFORE the devices are disconnected. Refer to the **CONNECT** routine for details.

- ERROR - CONNECT_220: Illegal connection for the ISRC and IMTR

You are trying to connect the ISRC and the IMTR to the same pin. Check the **CONNECT** routines.

- ERROR - CONNECT_617: device 617 is already connected.

You are attempting to connect the 617 more than once. Check the **CONNECT** routine to see if you are trying to connect the 617 more than once BEFORE the devices are disconnected. Refer to the **CONNECT** routine for details.

H.1.2 FORCEI Routine

- ERROR: device not found in FORCEI routine
device = (#)

You are trying to use an illegal device to force a current. Check device name in **FORCEI** routine.

- ERROR - FORCEI_220 - current out of range

You are trying to force a current > .1 or < -.1. Check **FORCEI** routine.

H.1.3 FORCEV Routine

- ERROR: device not found in FORCEV routine
device = (#)

You are trying to use an illegal device to force a current. Check device name in **FORCEV** routine.

- ERROR - FORCEV_617 - voltage out of range
voltage = (#)

You are trying to force a voltage > 10 or < -10. Check **FORCEV** routine.

H.1.4 HLSEL Routine

- ERROR - HLSEL - invalid response

You did not enter either HIGH or LOW for the parameter in this routine. Check the **HLSEL** routine for details.

H.1.5 INIT Routine

- ERROR: device not found in initialization

You are trying to initialize an illegal device name. Check **INIT** routine.

H.1.6 LIMITV Routine

- ERROR LIMITV: device not found
device (#)

You are using an illegal device for this routine. Check **LIMITV** routine.

H.1.7 MEASI Routine

- ERROR MEASI: Cannot use IMTR1 when using the low pins

You are trying to use IMTR1 to measure when connected to the low pins. Check **MEASI** routine for details.

- ERROR MEASI: device not found
device = (#)

You are using an illegal device in the **MEASI** routine. Check **MEASI** routine for details.

-ERROR MEASI_617: illegal connection made with the IMTR and the ISRC at pin (#)

You have connected the current meter with the current source this is an invalid connection.

H.1.8 MEASV Routine

- ERROR MEASV: Cannot use VMTR1 when using the Low pins

You are trying to use VMTR1 on the low pins. This is not a legal connection

- ERROR MEASV: device not found
device = (#)

You are using an illegal device in the **MEASV** routine. Check **MEASV** routine for details.

H.1.9 RANGEI Routine

- ERROR RANGEI: device not found
device = (#)

You are using an illegal device in **RANGEI** routine. Check **RANGEI** routine for details.

H.1.10 RANGEV Routine

- ERROR RANGEV: device not found
device = (#)

You are using an illegal device in **RANGEV** routine. Check **RANGEV** routine for details.

H.1.11 SSMEASI Routine

- ERROR SSMEASI: Cannot use IMTR1 when using the low pins

You are trying to use IMTR1 to measure when connected to the low pins. Check **SSMEASI** routine for details.

- ERROR SSMEASI: device not found
device = (#)

You are using an illegal device in the **SSMEASI** routine. Check **SSMEASI** routine for details.

H.1.12 SSMEASV Routine

- ERROR SSMEASV: Cannot use VMTR1 when using the low pins

You are trying to use VMTR1 to measure when connected to the low pins. Check **SSMEASV** routine for details.

- ERROR SSMEASV: device not found
device = (#)

You are using an illegal device in the **SSMEASV** routine. Check **SSMEASV** routine for details.

H.2 IEEE ERROR Messages

- ERROR -INIT BUS: STATUS = (#)

The BUS did not get initialized. Refer to CEC manual for description of status (#). Check cable connections.

- ERROR - IEEESEND: length of string is 0

String length sent out to the instruments was 0.

- ERROR IEEESEND: Status = (#)

String was sent incorrectly. Refer to CEC manual for description of status (#).

- ERROR IEEEREAD: Status = (#)

String was received incorrectly. Refer to CEC manual for description of status (#).

H.3 Magnet Controller ERROR Messages

H.3.1 Walker Magnet Controller

- ERROR - FORCEMAG Voltage reading from 196 is (#)

196 instrument made an incorrect reading.

- ERROR - FORCEMAG Trying to force (#) **Volts**

The magnet can only receive a voltage between 0 and 10 volts. This ERROR will occur only if the system is forcing an invalid voltage. Check to make sure magnet is operating correctly.

H.4 Cryo Controller ERROR Messages

H.4.1 MMR K-20 Cryo

-ERROR Cryo temp: ERROR Message is (STRING)

An ERROR occured in the **SETTEMP** routine. Refer to the MMR manual under the **SK** routine.

- ERROR Cryo temp is not set, ERROR message is (STRING)

An ERROR occured in the **SETTEMP** routine. Refer to the MMR manual under the **SK** routine.

- ERROR Cryo did not reset, error message is (STRING)
Reset the Cryo manually

The cryo did not reset. You must reset it yourself. Refer to MMR manual.

- ERROR read temp: Message is (STRING)

ERROR while trying to read the temperature. Refer to **TE** routine in the MMR manual.

H.4.2 Lakeshore 91C Cryo Controller

- ERROR -SETTEMP- Set temp was not set to #1 it was set to #2

An ERROR occured in the **SETTEMP** routine. The temperature that you wanted (#1) was not set, it was set to (#2). Check to make sure that the temperature that you programmed is a valid temperature. Refer to the Lakeshore owner's manual.

- ERROR -READTEMP- Message is (STRING) **Refer to 'W0' command in cryo manual**

An ERROR occured when the cryo tried to read the temperature. Check the string that was displayed to the string in the manual.

-ERROR -TEMPGAIN- Gain should be set to #1 gain was set to #2

An ERROR occured when setting the gain. Check the cryo manual to make sure that #1 is a valid gain value.

- ERROR -TEMPRATE- Rate should be set to #1 rate was set to #2

An ERROR occured when setting the rate. Check the cryo manual to make sure that #1 is a valid rate value.

- ERROR -TEMPRESET- reset should be set to #1 reset is set to #2.

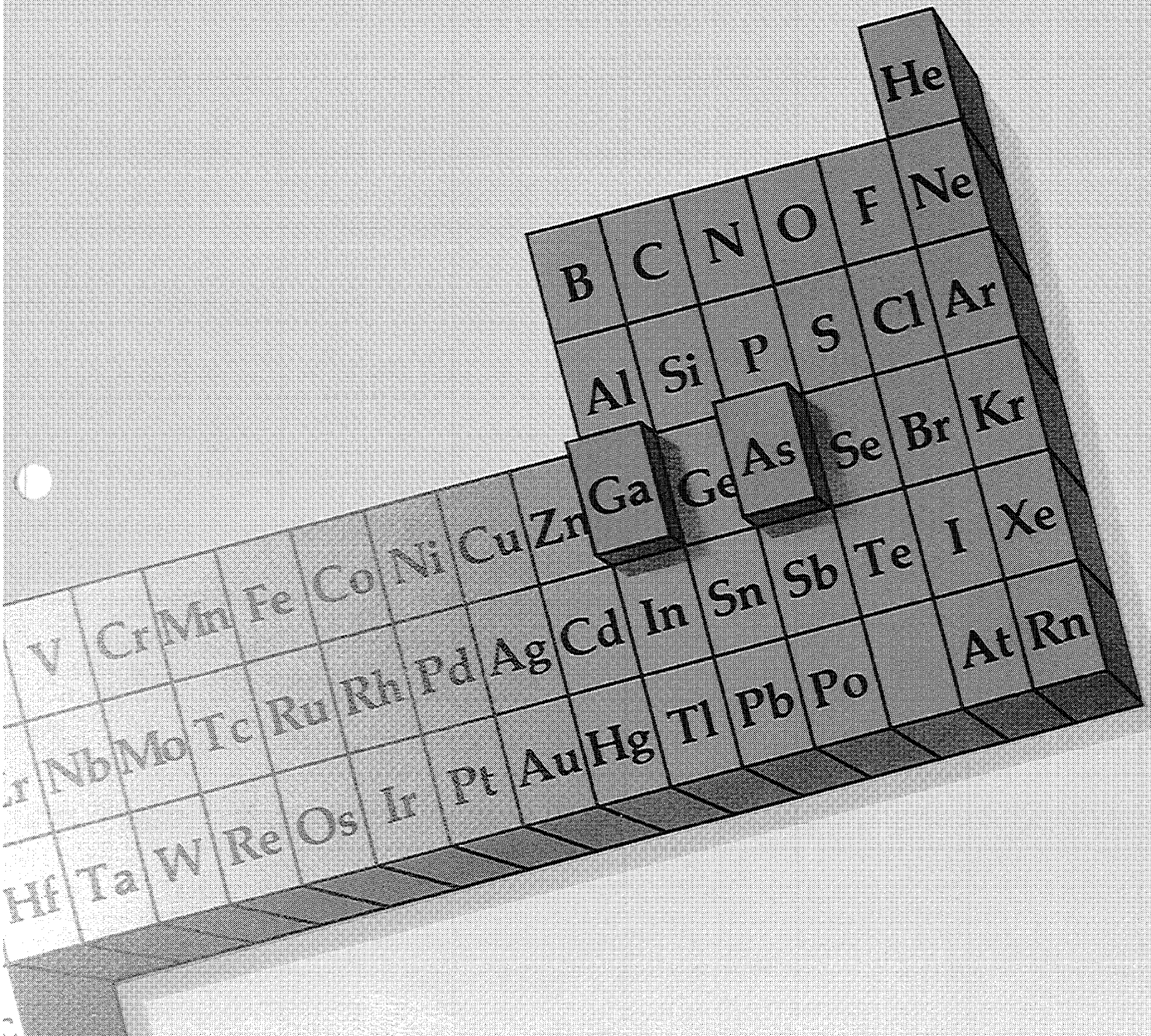
An ERROR occured when setting the reset. Check the cryo manual to make sure that #1 is a valid reset value.

- ERROR -TEMPRANGE- range should be set to #1 range is set to #2

An ERROR occured when setting the range. Check the cryo manual to make sure that #1 is a valid range value.

KEITHLEY

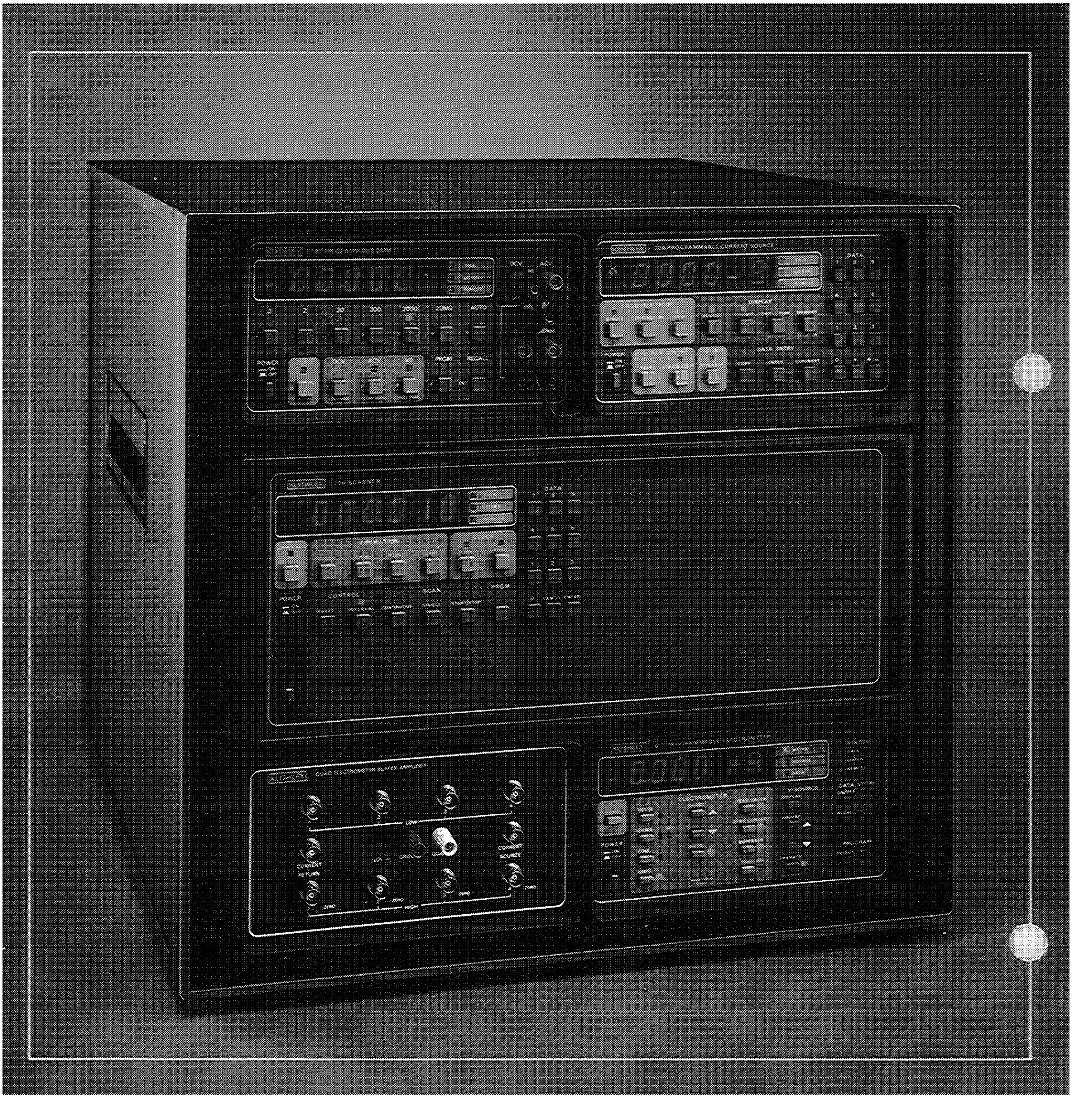
QUESTION: HALL MOBILITY



ANSWER: KEITHLEY!

OUR SYSTEM 110 CAN ANSWER YOUR
QUICKLY, ACCURATELY AND ECONO

The Keithley System 110 is a computerized test system for Hall effect, resistivity, and mobility measurements in bar and wafer samples of gallium arsenide and other compound semiconductors.



IALL EFFECT QUESTIONS...
·CALLY.

The Keithley System 110 provides reliable, accurate performance without requiring the time, expense, and frustration of assembling a Hall test system piecemeal. The System 110 is backed by 15 years of Keithley knowledge and experience in Hall effect measurement.

The System 110 uses the latest generation of digital instrumentation with driven guarding and an improved input switching matrix. The System 110 is fully integrated and tested by Keithley; thus, users can concentrate on testing rather than hardware and software develoment. A test box (dummy sample) is provided to verify system operation. A complete software package is included to perform test measurements and system diagnostics. The measurement programs automatically calculate the Hall coefficient and carrier mobility from resistivity and Hall voltage measurements. The user can select between configurations for Van der Pauw or six contact bar-type samples.

Instrumentation Package

The System 110 meets the requirements of ASTM Standard F76 pertaining to electronic test equipment (Standard Method for Measuring Hall Mobility and Hall Coefficient in Semiconductor Single Crystals, Designation F76). The instrumentation package includes a voltmeter, a current source, and a scanner mainframe and Keithley's new Quad Electrometer, Buffer Amplifier which may be remotely located near your magnet and cryogenics system. High system input impedance permits measurement of samples of semi-insulating materials. System 110 allows verification of sample currents and can detect nonohmic contacts. The low resistivity option can measure samples down to 20 milliohms. Low sample power from one microwatt to one milliwatt limits sample heating.

Controller and Programming Language

Control and programming of the System 110 has been simplified through the use of the IEEE-488 bus standard, and the inclusion of complete software source listings with the system. Program source codes are supplied in BASIC and are designed to run on the IBM Personal Computer or compatible.

The System 110 can also be controlled by other computer/language combinations which are capable of interfacing with the IEEE-488 bus as a talker/controller. In these cases, the standard software can be used as a model for program development. Consult Keithley Systems Division on questions regarding the suitability of a particular computer or language.

A Complete, Packaged System

The Keithley System 110 is delivered as a complete, packaged unit including all cabling, software listings, and comprehensive hardware and software manual. It is backed by applications and field service who can provide phone assistance and on-site service.

Keithley can supply the System 110 as a turnkey system including electromagnet with power supply, controller and Gauss meter, and a closed loop helium cryogenics system with controller.

For more product or applications information, or a demonstration of the Keithley System 110 contact your local Keithley representative or *Keithley Systems Division, 30500 Bainbridge Road, Cleveland, Ohio 44139-2216, (216) 248-6312.*

Hall effect measurements and the Keithley System 110.

In the early stages of semiconductor fabrication, the amount of dopant in a wafer is an important process control parameter. This is determined most often by measuring resistivity of a wafer of the material. The technique most commonly used for such measurements employs a four-probe collinear array, a constant current source, and a voltmeter.

This approach is usable with volume resistivities from 0.001 ohms/cm to 10,000 ohms/cm, easily covering the normal range for silicon. Compound semiconductors such as gallium arsenide, however, may have much higher resistivities; other techniques must be used to determine their characteristics.

The System 110 makes use of Van der Pauw method to measure resistivities greater than 10^{10} ohms/cm. The same system can also be used to measure the Hall coefficient, from which the Hall mobility and carrier mobility can be derived.

High Resistivity Materials

Circuitry for measuring the resistivity of a test sample by the Van der Pauw method is shown in Figure 1.

The Hall coefficient measurement is made with the contacts on the periphery of the sample connected to a current source and a very high input impedance differential electrometer, as shown in Figure 2, the resistivity of the material can be stated as:

$$(1)\quad p = \frac{\pi t}{\ln 2} \cdot \frac{R_a + R_b}{2} \cdot f$$

Where p is resistivity in ohms per cm, t is sample thickness in cm, and f is a correction factor dependent on the degree of symmetry in the sample connections.

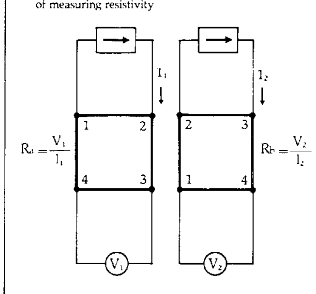


FIGURE 1 Connections for Van der Pauw method of measuring resistivity

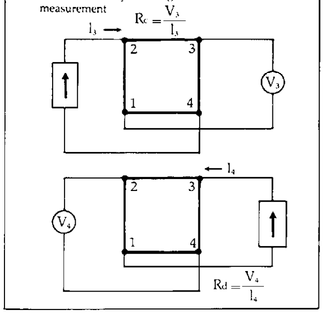


FIGURE 2 Circuitry rearranged for Hall coefficient measurement

For the case of perfect symmetry f = 1.0 and R_a will equal R_b. Equation #1 can then be reduced to:

where $R_m = R_a = R_b$, or

$$p = \frac{\pi t}{\ln 2} \cdot R_m, \qquad p = 4.53 R_{mt}$$

To measure the Hall coefficient, the circuit in Figure 1 is rearranged as shown in Figure 2. This arrangement measures two resistances, Rc and Rd. A magnetic field of known magnitude is then applied perpendicular to the sample, and changes in the two resistances (ΔRc and ΔRd) are noted. The Hall coefficient R_H can then be stated as:

$$R_H = \frac{\Delta R_c + \Delta R_d}{2B}$$

Where B is the magnetic field in gauss. The Hall mobility can be calculated from the equation:

$$\mu_H = R_H / p$$

Some Practical Considerations

Some materials have resistivities of 10^{11} ohms/cm or higher. The measured resistances in Figures 1 and 2 will then be on the order of 10^{12}. The voltmeter shown in these figures is a differential electrometer with input resistance much greater the 10^{12}. This is achieved by using electrometer unity gain buffers plus an isolated voltmeter.

An important advantage of the System 110 is its ability to neutralize cable capacitance. The shield of each sample lead is driven by its buffer output. This guards against cable leakage and reduces the system time constant. In the complete system, relays are used to connect the voltmeter between any two buffer outputs. The entire sequence of measurements is automated by computer control via the IEEE-488 bus.

KEITHLEY
Systems Division

Keithley Instruments, Inc./30500 Bainbridge Road/Cleveland, Ohio 44139-2216/U.S.A./(216) 248-6312/Telex: 4971512
CALIFORNIA: Keithley Instruments, Inc./2540 Mission College Blvd., #103/Santa Clara, California 95054-/U.S.A./(408) 986-8411/Telex: 25172271
ENGLAND: Keithley Instruments, LTD./1-3 Boulton Road/Reading, Berkshire RG2 ONL/(0736) 86 12 87/Telex: (851)172271
WEST GERMANY: Keithley Instruments GmbH/Heiglhofstrasse 5/D-8000 Munchen 70/(089) 710-02-30/Telex: (841)5212160
CHINA: Schmidt & Co. (Hong Kong) LTD./18 Fl. Great Eagle Center/23 Harbour Rd./Wanchai/Hong Kong/(5) 8330222/Telex: (780)74766

5/86 Printed in U.S.A.