Electrical Analysis of Low Energy Argon Ion Bombarded GaAs

by

Eric D. Cole

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering

APPROVED:

Dr. L. C. Burton, Chairman

Dr. F. W. Stephenson

Dr. S. Onishi

Dr. R. Zallen

Dr. J. Dillard

June 1988
Blacksburg, Virginia

Electrical Analysis of Low Energy Argon Ion Bombarded GaAs

by

Eric D. Cole

Dr. L. C. Burton, Chairman

Electrical Engineering

(ABSTRACT)

An electrical analysis was done on Al and Au Schottky diodes fabricated on n-type (100) GaAs which had been bombarded with low energy Ar ions. The purpose of this study was to quantify electrically damage caused by the Ion Beam Etching (IBE) as functions of energy and fluence.

Electrical studies included Deep Level Transient Spectroscopy (DLTS), Current-Voltage (I-V), Capacitance-Voltage (C-V), Conductance-Voltage (G-V), Capacitance-Temperature (C-T), and Activation Energy Analysis. These electrical measurements were carried out on GaAs which had been exposed to a variety of treatments after IBE (such as chemical etch removal) to determine damage depth.

At the lowest energy studied, 0.5keV, Schottky reverse saturation currents (I_{sat}) increased by over 4 orders of magnitude from the virgin case. The ideality factor, n, increased slightly while the breakdown voltage decreased. The most prominent changes occurred in the DLTS spectrum where it was observed that the native arsenic defect EL2 peak disappeared completely after ion etching. Concurrently a sharp increase in the diode conductivity with temperature was seen. It was found that chemical removal of 100Å of GaAs by chemical means could restore most of the diode parameters and the EL2 peak. It is proposed that the loss of EL2 is not related to a true physical reduction (i.e. an arsenic deple-

tion) since calculations showed that the As loss would have extended beyond 3000Å for detectable DLTS changes. Also, the EL2 peak could be made to artificially disappear on a virgin sample with an external diode shunting resistor. The loss of the EL2 peak is, rather, attributed to a thin low resistivity surface layer having a partly amorphous non-stoichiometric crystal structure which can desensitize or mask the DLTS measurement. Surface chemical etch studies over the top of the Schottky diodes recovered 25% of the EL2 peak supporting this conclusion. Lower fluences had no effect at 0.5keV.

Increasing ion bombardment energy showed a steady degradation in diode ideality factors. The reverse breakdown voltage increased past the unetched value and the DLTS spectrum began to show a very slight return of EL2. At 3keV the ideality factor was large, indicating the presence of a somewhat thicker high resistance layer. In fact recovery of diode parameters and EL2 did not occur until after 1000Å removal. This was much deeper than expected at this energy, according to theory.

Physical and lumped R-C electrical models are reported with an accompanying computer simulation of experimental DLTS results. The simulation used both thin low resistance and thick high resistance top layers to show that EL2 could be removed artificially. The models were also somewhat successful in explaining previously reported capacitance dispersion found in IBE GaAs.

ACKNOWLEDGEMENTS

The author would like to thank Dr. L. C. Burton, who served as his committee chairman, for providing him with research funding and facilities as well as invaluable information concerning research and teaching that will serve well in the author's future professional career. Thanks also to Dr. R. Zallen and Dr. J. Dillard for serving on the author's committee, and their helpful suggestions and assistance during the course of this research, including the use of the Surface Analysis Lab and communication of their own results which helped guide this study. The author would also like to thank Dr. F. W. Stephenson not only for serving on the committee but also for use of the Hybrid Microelectronics Lab facilities; and Dr. S. Onishi for being on the committee and for many interesting, insightful conversations.

Much gratitude to for the many hours he spent helping to develop the MEDUSA software and his patience for my continuously changing designs. Thanks to for her many hours spent preparing the GaAs samples. Thanks to for his helpful assistance in making "good" diodes, and his and efforts in bringing the big evaporator up to par. The author appreciates the help and skill of for his contributions to the entire project.

The author would also like to thank his lab and research colleagues

and many others for their friendship, advice and good humor.

Lastly I would like to thank my wife for her hard work, good management and constant support throughout my life with her. Also to my family and my wife's family who provided us an oasis in the desert. And finally to my daughter whose smiling face melts through the tension of the day like the sun in winter.

This research was funded by Texas Instruments and the Virginia Center for Innovative Technology.

This work is dedicated to my father-in-law . I am truly privileged and fortunate to have encountered such an individual in my lifetime. And though the sun still rises and the pages of books still turn, these things are forever different for me. I weep at his passing but my heart rejoices that I knew and loved him and the course of my life is and ever shall be enriched.

Table of Contents

Page

List of Figures

Page

Page

List of Tables

Chapter 1: INTRODUCTION

GaAs is not a new semiconductor (it was first made in the 1920's by Goldschmidt) but it is beginning to emerge from the laboratory into the market place [1]. Once touted as a replacement for Si, GaAs is now seen as a material that fills the gaps left by Si in microwave and high speed digital component, electro-optic (lasers, photodetectors) and power component applications. Presently the major consumer of GaAs, due to its high cost and specialized uses, is the military. This will change as material quality and device processing yield improves. The aim of this thesis is to report in detail on one processing step, Ion-Beam Etching (IBE), and its effects on Schottky device quality. First, however, it will be instructive to review some of the basic properties of GaAs and processing steps involved in device fabrication.

1.1 Basic Properties of GaAs

GaAs is a III-V compound semiconductor having a zincblende crystal structure (see Fig. 1), with a lattice constant a = 5.6533Å and the two sublattices separated by 2.44793Å along the body diagonal of the unit cube.

Figure 2 shows the band structure of GaAs from which several important properties can be interpreted. GaAs has a direct bandgap with E_G = 1.423eV, making it ideally suited for electro-optic applications, since electron transitions from the valence band to the conduction band and vice versa at k = 0 occur through pure photon interaction (i.e. no phonons necessary) [4]. These properties are exploited in GaAs lasers and photodetectors. As a result of the wider bandgap and the presence

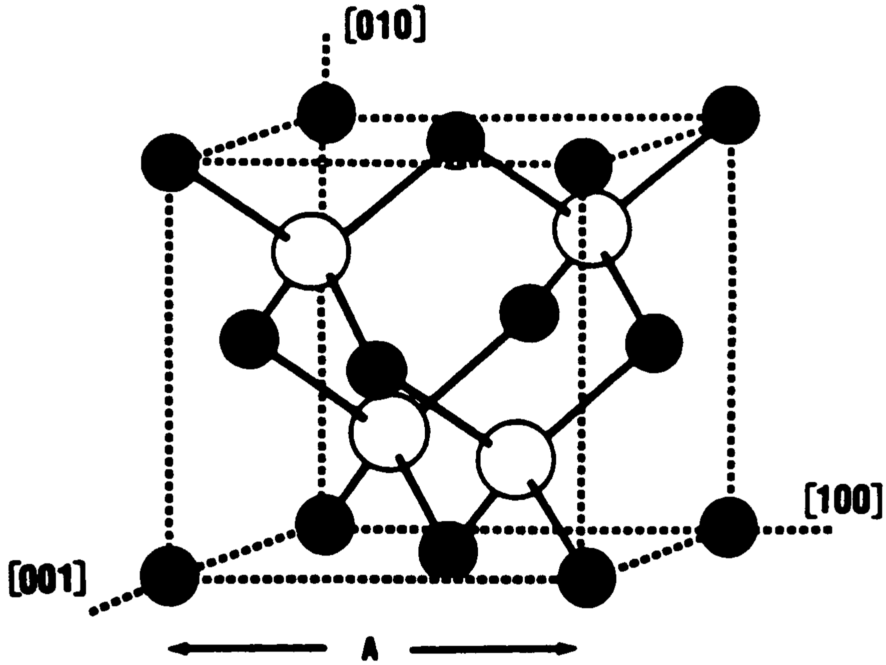


Figure 1. GaAs crystal structure (black = Ga, white = As) [taken from Ref. 2].

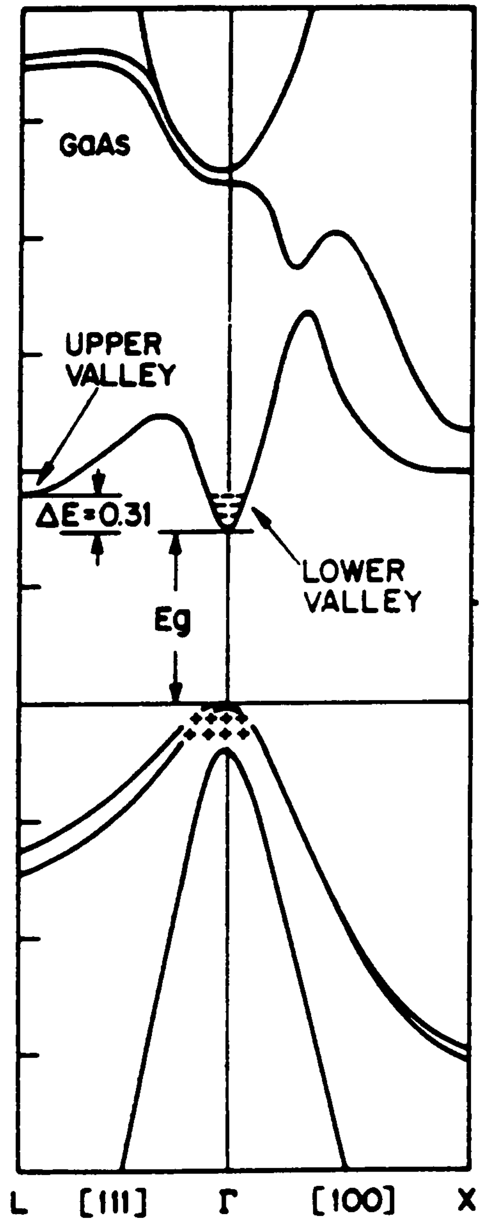


Figure 2. Electronic band structure of GaAs [taken from Ref. 3].

of a native arsenic defect, named EL2, GaAs is semi-insulating in its undoped state, giving excellent device isolation compared to Si, thus eliminating the need for guard ring structures. (EL2 is further explained later in this section.)

Another salient feature of the band structure can be derived from the sharpness of the conduction band at k = 0 where the electron effective mass is smallest [5]. This results in an electron effective mass to rest mass ratio (m_e^*/m_o) of .067 for GaAs compared to .98 for Si. Thus for a given applied electric field the electron will attain a higher saturated velocity in GaAs than in Si, thus making GaAs an electrically faster material. This is usually seen in a comparison of the electron mobilities: μ(GaAs) = 8500cm^2/V-s, μ(Si) = 1500cm^2/V-s. The hole mobilities are nearly equal ($\simeq$ 400cm^2/V-s) indicating that n-type unipolar devices are the most advantageous structures in GaAs [6]. Being a faster material allows devices to be fabricated that can operate at higher frequencies (>100 GHz) and switch in shorter times (<100ps) [7].

The band structure of GaAs also indicates the possibility of intervalley scattering via phonon interaction (i.e. transferred electron effect). This property is exploited in microwave devices such as the Transferred Electron Device (TED). A summary of these and other GaAs properties is shown in Table 1.

It is appropriate at this point to discuss the traps and defects native to or typically found in GaAs in view of the study to be reported later in this paper. An extensive literature search was performed, with the results given in Table 2 [9-33]. As can be seen the most commonly reported electron trap in all types of GaAs is EL2.

Table 1. GaAs physical and electrical properties [taken from Ref. 8].

Properties	GaAs
Atoms/cm^3	4.42×10^{22}
Atomic weight	144.63
Breakdown field(V/cm)	$\sim 4 \times 10^{5}$
Crystal structure	Zincblende
Density (g/cm^3)	5.32
Dielectric constant	13.1
Effective density of states in conduction band, N_C (cm^{-3})	4.7×10^{17}
Effective density of states in valence band, N_V (cm^{-3})	7.0×10^{18}
Effective Mass, m^*/m_0	
Electrons	0.067
Holes	$m^*_{lh} = 0.082$ $m^*_{hh} = 0.45$
Electron affinity, χ(V)	4.07
Energy gap (eV) at 300 K	1.424
Intrinsic carrier concentration (cm^{-3})	1.79×10^{6}
Intrinsic Debye length (μm)	2250
Intrinsic resistivity (Ω-cm)	10^{8}
Lattice constant (Å)	5.6533
Linear coefficient of thermal expansion, $\Delta L / L \Delta T$ (°C^{-1})	6.86×10^{-6}
Melting point (°C)	1238
Minority carrier lifetime (s)	$\sim 10^{-8}$
Mobility (drift) (cm^2/V-s)	
	8500
	400
Optical-phonon energy (eV)	0.035
Phonon mean free path λ_0 (Å)	58
Specific heat (J/g-°C)	0.35
Thermal conductivity at 300 K (W/cm-°C)	0.46
Thermal diffusivity (cm^2/s)	0.24
Vapor pressure (Pa)	100 at 1050°C 1 at 900°C

Table 2. Common defects found in GaAs [9-33].

Common Name	E_T (eV)	N_T (cm^{-3})	x-sect (cm^2)	GaAs	Source
EL2	.72-.82	10^{15}	10^{-12}	LEC, VPE, MBE, S. I., MOCVD	As_{Ga}, cluster
EL0	.82	10^{15}	10^{-14}	LEC, S. I.	oxygen
EL12	.72	10^{14}	10^{-14}	LEC	EL2?
Cr?	.62	--	10^{-15}	LEC, VPE	chromium
EL3	.58	10^{15}	10^{-15}	LEC, HB, HGF	
EL4	.52	10^{13}	10^{-14}	LEC	
EL5	.42	--	10^{-14}	LEC, HB	
EL6	.34	10^{14}	10^{-13}	LEC, HB	
EL8	.27	--	10^{-16}	LEC, HB, MOCVD	
EL14	.22	10^{15}	10^{-15}	LEC, HGF	oxygen

EL2 is a trap which is generally attributed to an As_{Ga} antisite or As cluster defect, though much controversy and research have surrounded EL2 and its source. Early reports showed oxygen impurities to be a possibility since its energy was very close, but much data supports arsenic defects, including crystal stoichiometry- EL2 density relationships [34-40].

Many factors affect the density of EL2 in wafers and epi-layers including melt stoichiometry (As inbalances), anneal temperature and even saw speed (refer to Fig. 3) [41]. The radial distribution of EL2 across LEC wafers is of prime importance in device fabrication and is usually measured using optical scanning techniques, though a chemical etch with KOH gives an etch pit density (EPD) which often correlates with the EL2 distribution (refer to Fig. 4) [42]. EL2 is generally considered responsible for giving GaAs its semi-insulating characteristic in the undoped state since it is located approximately .75eV below the conduction band, near mid-gap, effectively pinning the bulk Fermi level at midgap for S.I. behavior. EL2 as measured by DLTS will be seen later to be a very important monitor of device quality.

Many other physical and electrical properties can be seen in the next section on device processing.

1.2 GaAs Device Fabrication

Fabrication of GaAs devices is a long, involved process from ingot growth to finished device. A typical processing flowchart is given in Fig. 5. Each fabrication step could and has been the subject of much research activity, and a few are briefly described below.

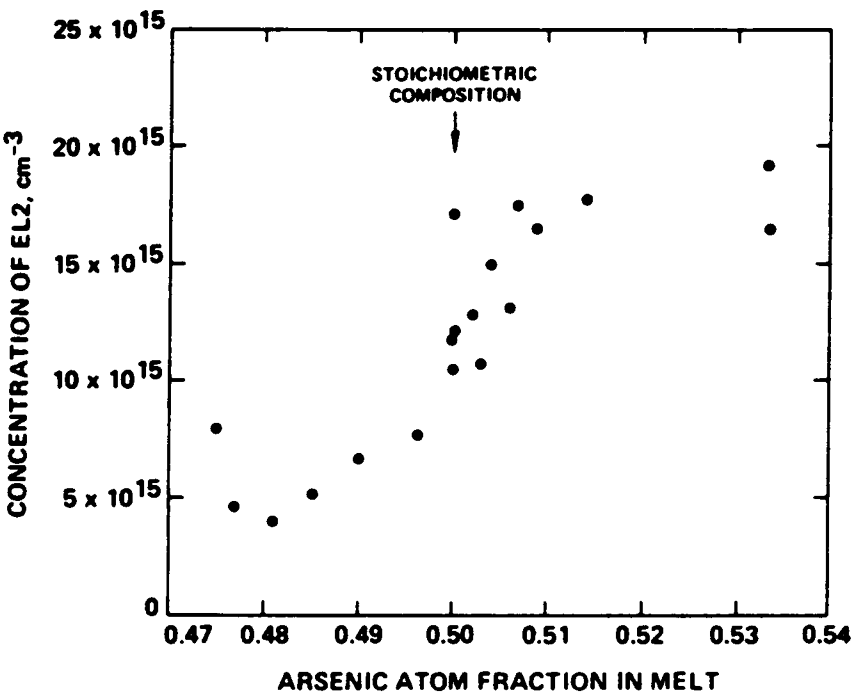


Figure 3. Dependency of EL2 density on GaAs melt stoichiometry [taken from Ref. 41].

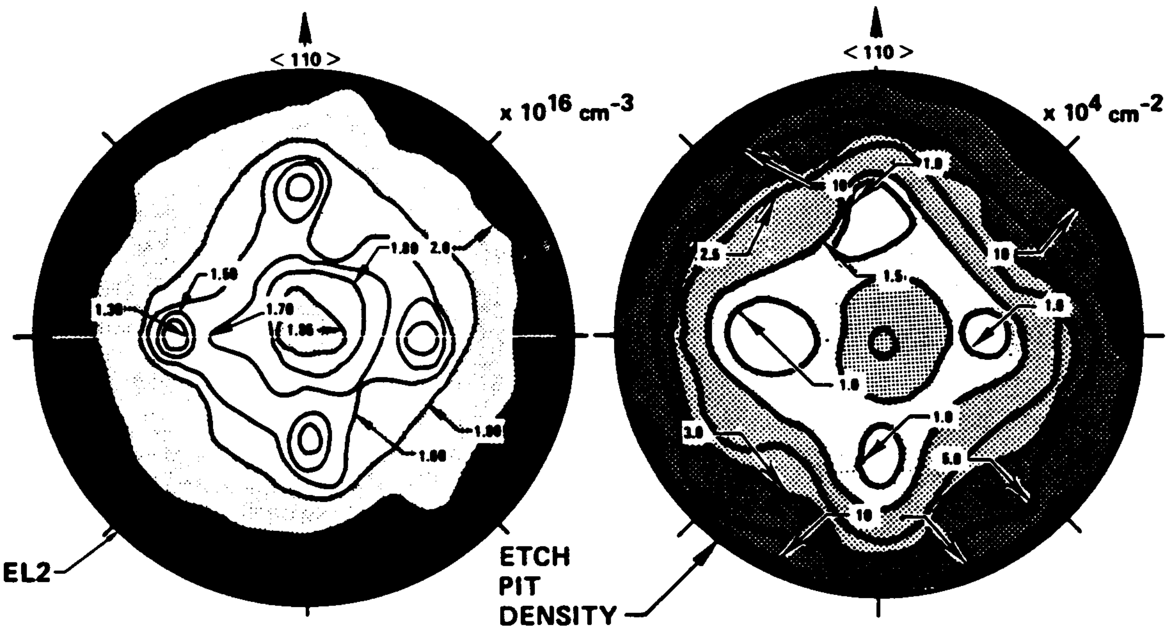


Figure 4. Radial distribution of EL2 across wafer with EPD correlation [taken from Ref. 42].

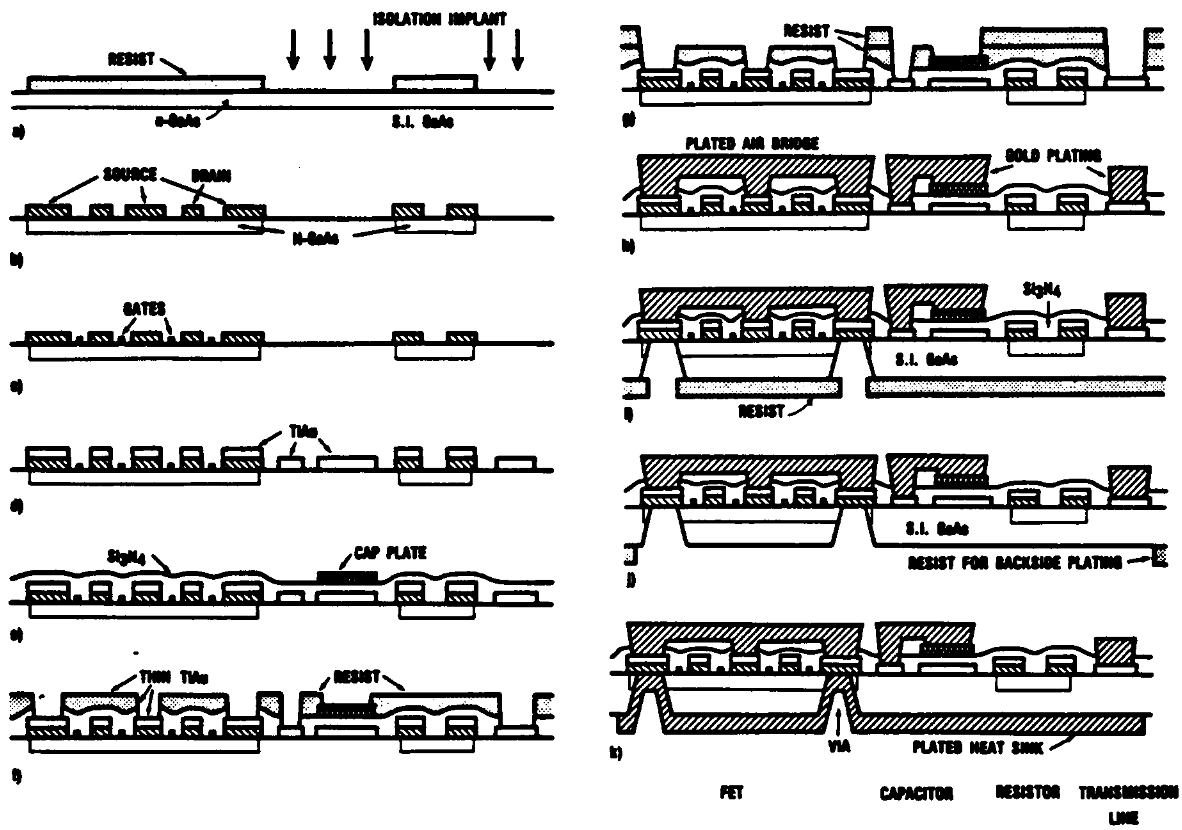


Figure 5. Flowchart of a typical GaAs integrated circuit fabrication process [taken from Ref. 1, pp. 14-15].

Industrial wafers are typically 2" in diameter (3" wafers are presently gaining popularity) grown using Liquid Encapsulated Czochralski (LEC) techniques, though Horizontal Bridgman growth is also used. Epitaxial layers for device fabrication can be made using ion-implantation, MBE, LEC or MOCVD techniques, depending on the device to be produced. The wafer and epi or active layer quality are critical, since defects can lower electron mobility drastically, effectively destroying devices. These defects may be introduced by anything from ingot pull rate to implant energy, as was seen in the previous section

Wafer processing also includes some wet cleaning and etching steps for surface preparation and structural definition. Typical organic solvents used for degreasing and general purpose cleaning include: ethanol, acetone, benzene and propanol. Etchants for GaAs along with their etch rates can be seen in Table 3. The H_2SO_4:H_2O_2:H_2O and Br_2:CH_3OH systems are the most popular with the second being useful for polishing. Due to its crystal structure, GaAs wet etches at different rates along different planes, with variables such as solution temperature and stir speed also affecting etch rates. This creates a problem in resolution, design and controllability during processing which can affect yield and cost (refer to Fig. 6).

To circumvent some of the drawbacks of wet etching, dry processing techniques can be used and are gaining popularity. These techniques include plasma assisted etching (PE), Reactive Ion Etching (RIE) and Ion Beam Etching (IBE). Plasma etching occurs in a similar fashion to metal deposition using Rf sputtering, except that the target becomes the GaAs wafer itself. The etch rate is the highest of the three and plasma

Table 3. Common wet etchants used on GaAs along with their etch rates [taken from Ref. 43].

(proportions are equal volumes unless otherwise noted)

Etchant	Etch Rate (μm/min)
$HCl/CH_3COOH/H_2O_2$	2.0
$HCl/CH_3COOH/(1N\text{-}K_2Cr_2O_7)$	0.40
$HCl/H_3PO_4/H_2O_2$	0.75
$HCl/H_3PO_4/(1N\text{-}K_2Cr_2O_7)$	0.04
HCl/HNO_3 (1:1)	0.50
(1:2)	0.75
(2:1)	0.25
$HCl/HNO_3/H_2O$	0.83
$HCl/HNO_3/H_2O_2$	1.0
$HCl/HNO_3/CH_3COOH$	1.3
HNO_3/H_2O_2	7.0
HNO_3/CH_3COOH	-
$HNO_3/CH_3COOH/H_2O_2$	4.5
HNO_3/H_3PO_4	10.0
$HNO_3/H_3PO_4/H_2O_2$	3.5
$H_2SO_4/H_2O_2/H_2O$	5.0
$H_2SO_4/CH_3COOH/H_2O$	2.5
$H_2SO_4/H_3PO_4/H_2O$	3.0
$H_2SO_4/HCl/(1N\text{-}K_2Cr_2O_7)$	0.75
$H_3PO_4/H_2O_2/H_2O$	4.0
$H_3PO_4/CH_3COOH/H_2O_2$	2.0
$H_3PO_4/CH_3OH/H_2O_2$	2.5
$H_3PO_4/C_2H_5OH/H_2O_2$	2.0
$HF/HNO_3/H_2O$	10.0
$HF/HNO_3/H_2O_2$	8.0
$HF/HNO_3/CH_3COOH$	80.0
$HF/HNO_3/H_3PO_4$	50.0
$HF/H_2SO_4/H_2O_2$	21.0
HBr/HNO_3	0.75
$HBr/NHO_3/H_2O$	0.05
$HBr/CH_3COOH/(1N\text{-}K_2Cr_2O_7)$	1.5
$HBr/H_3PO_4/(1N\text{-}K_2Cr_2O_7)$	1.0
4% Br_2/CH_3OH	6.0
1% Br_2/CH_3OH	0.67
(1% Br_2/CH_3OH)/CH_3COOH	0.20
(1% Br_2/CH_3OH)/H_3PO_4	0.02
NaOCl	1.8
NaOCl/HCl (5:1)	0.75
(1N-NaOH)/H_2O_2/H_2O (1:1:10)	0.38
(1N-NaOH)/H_2O_2/NH_4OH (5:1:1)	1.1
$NH_4OH/H_2O_2/H_2O$ (1:1:5)	1.8
(1N-KOH)/H_2O_2/H_2O (1:1:10)	0.5
(1N-KOH)/H_2O_2/NH_4OH (5:1:1)	1.4

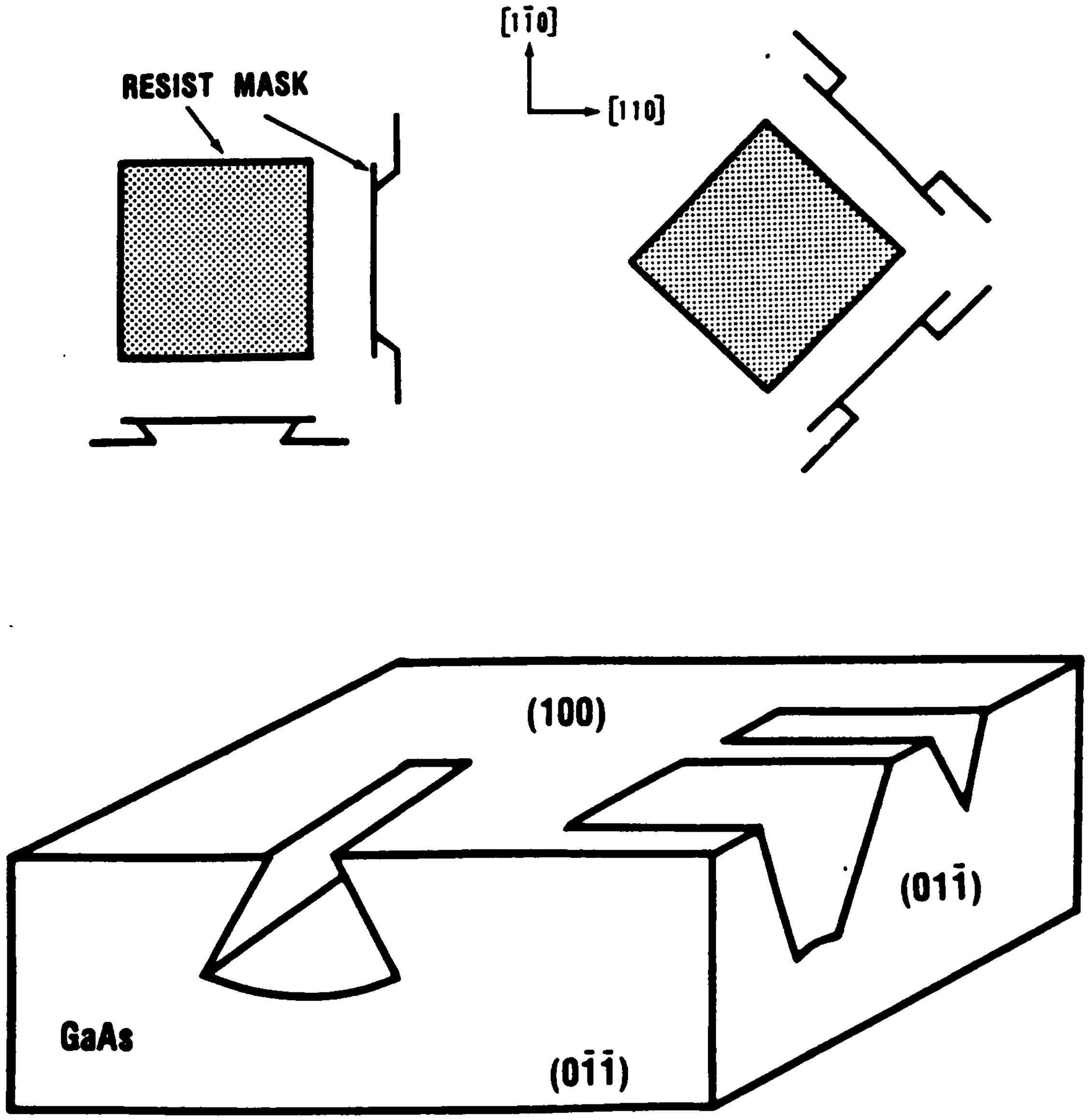


Figure 6. Cutaway of GaAs wafers showing the effects of different etch rates on structure definition [taken from Ref. 44].

etching is often used for via hole fabrication. RIE is similar to PE except in addition to the inert plasma ion a reactive species such as CF_4 is included to improve the anisotropic nature of the etch. A survey of the effects of these two techniques is given in Table 4. Processing variables in PE and RIE included ion energy, vacuum quality and wafer angle which can affect etch rates and crystal stoichiometry, degrading device quality. IBE is somewhat different from the above, and since it is the main subject of this research, its description is taken up in the next section.

This section briefly touched on a few major points involved in GaAs device processing. It was not extensive but was included to give the reader a flavor of the complicated nature of this subject and to show that device quality depends on a near infinity of variables, including those involved in IBE.

1.2.1 Ion Beam Etching (IBE)

Ion Beam Etching (IBE), or ion milling as it is sometimes called, is an important and useful processing technique for GaAs. A typical system is shown in Fig. 7 where it can be seen that IBE is simply the bombardment of a given surface with gas ions of various species, energies and fluences [46]. At low energies (100eV), inert gas ions, (usually Ar^+ for this study) are used to clean wafer surfaces of organic, oxide and other contaminations which could impede device fabrication. At higher energies (>500eV) the ion beam can be used to define structures for devices such as isolation mesas and, more importantly, recessed channels for gate metalization of MESFETs (Fig.

Table 4. Effects of Plasma Etching and Reactuve Ion Etching on GaAs and GaAs devices [taken from Ref. 45, table entry references are given there].

Gas used	Type of system used	Etch conditions	Observations, etch rates, surface and electrical effects
CCl_2F_2, CCl_4, PCl_3, HCl	parallel plate, PE	power supplied to both electrodes; 200–500 mτ	etched both GaAs and oxides; GaAs etch rate much higher than oxide etch rate
Cl_2,$COCl_2$	same	same	GaAs was etched, oxide was not; etch rate ~ 5 μm min^{-1} at 5 W cm^{-2} for Cl_2, very rough surface
CCl_2F_2/Ar/O_2 (:1:1)	RF–RIE	0.5 W cm^{-2}; 5 mτ, 500 V self-bias; 10 mτ, 650 self-bias	at 5 mτ, 3800 Å min^{-1} etch rate; at 10 mτ, (0 Å min^{-1} etch rate; very sharp angle etch walls
Ar	DC–RIE	−3 kV on bottom; 40 mτ	400 Å min^{-1} etch rate
Ar/CCl_2F_2 (9:1)	same	25–60 mτ	500–2000 Å min^{-1} etch rate
CCl_2F_2	same	10–40 mτ	2000–8000 Å min^{-1} etch rate; surface carbon build-up observed and ion bombardment helps remove it; rate-limiting factor is GaF_3 sputter removal
CCl_2F_2/He (1:1)	RF–RIE	40 mτ, 0.18 W cm^{-2}	GaAs to AlGaAs etch rate = 200:1, with ~ 20 Å min^{-1} for AlGaAs; more surface degradation observed for lower He
CCl_2F_2/O_2 (1:1)	RF–RIE	0.1 mτ, 50 W (area unknown)	7 μm min^{-1} GaAs etch rate
CCl_3F/O_2	55 kHz -PE	300°C, 160 mτ	addition of O_2 binds carbon and releases Cl and F ions for GaAs etching; high O_2 concn lowers etch rate due to GaAs surface oxidation
Br_2	0.1–14 MHz	300 mτ, 30 SCCM Br_2, 100°C, ≲ 0.5 W cm^{-2}	for < 0.3 W cm^{-2}, etch rates {111} Ga < {110} < {100} < {111} As; first demonstration of preferential crystallographic plasma etching of GaAs
CF_4 CHF_3	RF-RIE-	500 V self-bias	ϕ_B and V_{BR} increase, but new deep levels appear

mτ = mTorr.

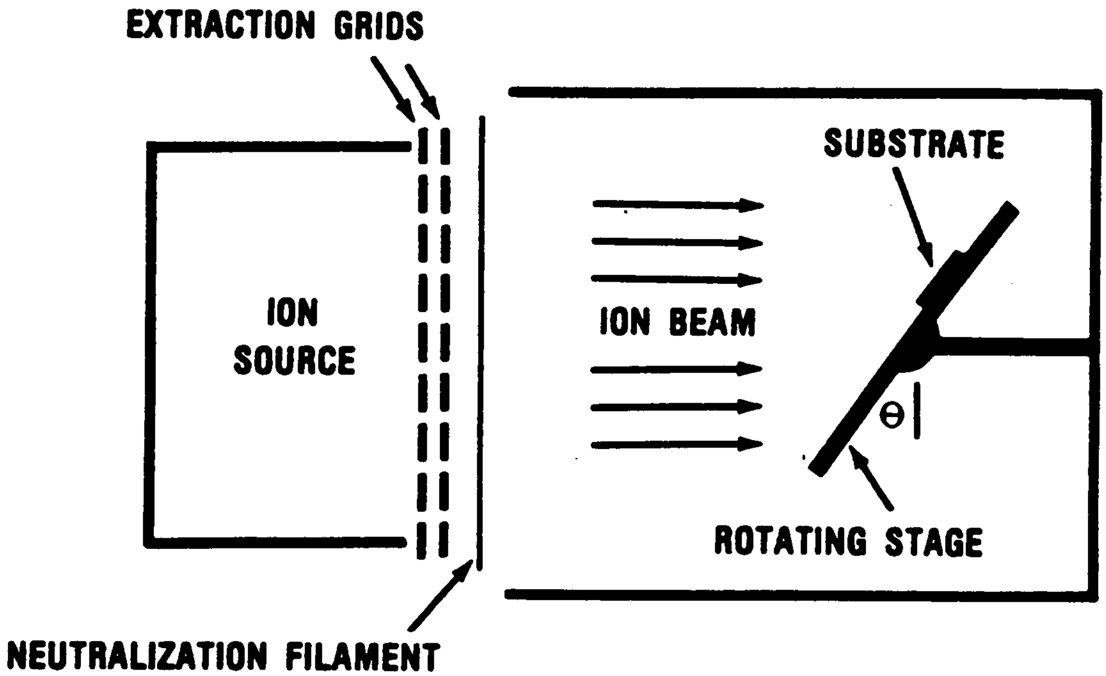


Figure 7. Typical system used for ion-beam etching [taken from Ref. 46].

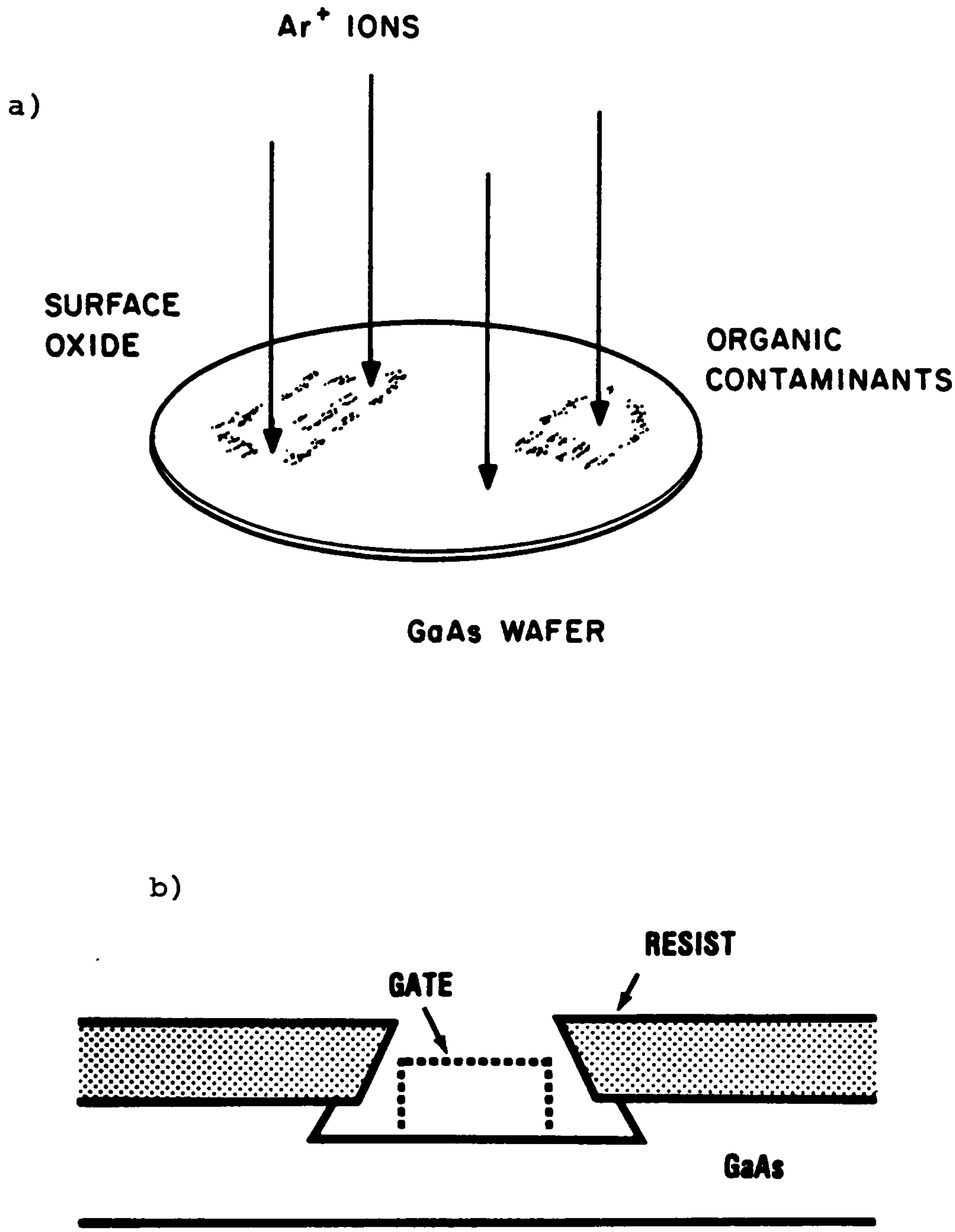


Figure 8. Typical uses of low energy IBE including (a) wafer cleaning and (b) MESFET gate recessing.

8). IBE processing is very useful in the latter application due to the extreme anisotropic nature of the beam. This minimizes mask undercutting, thus improving resolution of device features.

IBE, however, is not without its detriments. Since energetic ions are incident on the surface, radiation damage in the semiconductor is inevitable. Further, in GaAs it is found that arsenic is preferentially etched, making crystal stoichiometry with associated defects a problem [47]. This damage in general reduces the quality and reproducibility of devices made on their surfaces.

Ar^+ IBE can also be viewed as low energy ion-implantation with an unreactive ion species. Therefore implant theory and tables may be used for prediction of depth of penetration and straggle. Table 5 lists values for Ar^+ in GaAs assuming a Gaussian distribution. Further use of this table will be made in subsequent chapters.

1.3 Literature Review and Research Objectives

Although there is a vast amount of literature on GaAs in general, the amount which includes studies of electrical effects of IBE is somewhat small. However a review and summary of these results will be helpful in outlining further research.

Most of these studies have been done on either Al or Au Schottky diodes fabricated on (100) n-GaAs [49,50]. Of course these parameters are dependent upon processing techniques. The barrier height has been shown to be more or less independent of the metal due to surface state pinning of the Fermi level [51]. Al is more thermally stable than Au up to temperatures of 450°C, while Au tends to react with the substrate and getter Ga at 200°C [52].

Table 5. Theoretical ion implantation parameters for argon in GaAs (RP = range, DRP = straggle [taken from ref. 48].

E	RP	DRP	DY	SK
1	25	20	17	0.43
3	45	36	31	0.42
5	60	47	41	0.42
10	93	70	61	0.40
20	153	108	95	0.37
30	213	142	125	0.34
40	273	175	155	0.30
50	335	207	184	0.27
60	397	238	212	0.24
70	461	268	241	0.22
80	525	298	269	0.19
100	657	358	325	0.14
120	792	415	381	0.11
140	930	471	436	0.07
160	1070	526	491	0.03
180	1212	579	545	-0.01
200	1356	632	599	-0.04
220	1501	683	652	-0.07
240	1647	733	705	-0.10
260	1794	782	757	-0.13
280	1943	830	808	-0.16
300	2091	877	858	-0.19
350	2465	989	982	-0.25
400	2841	1096	1101	-0.31
500	3593	1292	1328	-0.41
600	4341	1470	1539	-0.49
700	5081	1632	1737	-0.57
800	5811	1779	1923	-0.64
900	6530	1914	1098	-0.70
1000	7237	2039	2262	-0.76

Some of the major papers on studies of IBE are tabulated in Table 6 [53-60]. It must be noted that only two studies have exceeded 1keV in energy, and the Kawabe article [53] did not discuss any electrical studies. The Holloway study [60] also is somewhat specialized in that the Schottky metals were deposited in situ after IBE. Thus there was no oxide present between the substrate and the metal; but the oxide is a very real and important part of Schottky devices. (Nicollian [61] reports that the oxide serves to change I-V's, and can even be conductive.)

In general, the studies given in Table 6 [55-60] showed that ϕ_b dropped while J_{sat} and n increased with IBE. Also, arsenic appeared to be preferentially etched at the surface since XPS showed an increase in the Ga/As ratio [60]. Holloway, however, showed this trend to change somewhat in that the 5keV sample appeared arsenic rich. All papers agree that a partly amorphous layer of less than 100Å in thickness is developed, while defects can diffuse into the bulk at room temperature. Pang [57] showed that for a 1000eV ion etch I-V characteristics recovered after a 250Å wet etch, but C-V curves did not recover until 750Å of material was removed. The depth of the diffused damage can be quite deep. Ghandhi reports that an Ar^+ IBE at ~100eV produced damage observable at 900Å from the surface. Detectable damage was found as deep as 2000Å by Kawabe using He backscatter and enhanced chemical activity.

Holloway has performed DLTS on his samples and found peaks at .3eV and .6eV below the conduction band edge. He failed to note, however, what traps were present before IBE. This will be shown to be an important distinction in the results of this study.

Table 6. Summary of useful literature of Ar^+ IBE on GaAs.

Energy (eV)	Experiments performed	ref #
50 W plasma	Rutherford backscatter, Raman	53
100-2000	He backscatter, photoluminescence, chemical activity	54
500	I-V	55
100-1000	I-V, C-V	56
250, 500, 1000	I-V, C-V, XPS, DLTS on RIE only	57
100	C-V	58
50W Sputter	I-V	59
1000-5000	DLTS, I-V, C-V, XPS (UHV Schottky deposition after IBE in situ)	60

J. Feng et al. [62] reports that 4keV bombarded surfaces showed a change in the UV reflectivity toward an amorphous crystal structure. It was also shown by J. Epp using ESCA that low energy ions (0.5keV) did not completely remove the native oxide, while higher energies were As deficient and contained a somewhat distorted oxide (different composition compared to virgin) when exposed to atmosphere [62]. Sen reports capacitance-frequency dispersion with 3keV IBE which indicated deep traps and a possible amorphous layer near the surface. He showed that weak reverse bias and reduced temperatures could eliminate the effect [63].

It can be seen from this brief review that there is a general lack of data concerning electrical properties of IBE samples over 1keV. Trap analysis using DLTS is practically nonexistent. Electrical analysis and modelling of low energy ion-etching, and its effects on GaAs surfaces, are necessary if this technology is to be used effectively for device fabrication [64-67]. It is the intent of this thesis to provide this data along with physical and electrical models quantifying IBE induced damage.

In order to be contributory, this study focused on higher energies (>.5keV) and low currents (<100μA) which brought the IBE studied here into the realm of ion cleaning. This does not reduce the importance of this process since dry cleaning techniques are gaining increased popularity and little has been reported about their damaging effects.

The following questions were addressed in this study:

1. How does ion cleaning affect Schottky diode device parameters as

functions of ion energy and fluence?

2. To what depth does electrically detectable damage occur?
3. What is the physical nature of that damage and how does it relate to the electrical measurements?
4. How can the induced damage be modelled?
5. What can be done to repair the residual damage to improve device quality?

Other objectives related to, but not directly associated with, these questions included:

1. Design and construction of an automated electrical test station to increase through-put and reduce the analysis burden.
2. Production of "good" working Schottky diodes starting from the polished wafer.

The methods used to answer these and other questions can be found in Chapter 3.

Chapter 2: Schottky Diodes on GaAs

The Schottky diode was the device chosen for probing crystal and electrical damage due to its simplicity, sensitivity and versatility in detailing electrically related materials parameters. The diode itself consists of two basic contacts, the Schottky or rectifying contact and the ohmic or linear contact. Each is detailed below.

2.1 Schottky Contacts

2.1.1 Ideal

A Schottky or rectifying contact is formed by contacting a metal of a given work function ϕ_M to a semiconductor of a given work function ϕ_S. The resulting band-diagram is given in Fig. 9 for an n-type semiconductor with $\phi_M > \phi_S$. The resulting barrier height looking from the metal to the semiconductor is given as

$$\phi_B = \phi_M - \phi_S \tag{1}$$

Looking from the semiconductor to the metal gives a barrier

$$V_{do} = \phi_M - x_S \tag{2}$$

where x_s is the electron affinity of the semiconductor [68,69]. Table 7 gives a listing of measured barrier heights for various semiconductors and metals.

Unfortunately the picture for real Schottky diodes is not quite as simple as the ideal Schottky contact shown in Fig. 9. Problems such as surface states and oxide layers can completely change Schottky

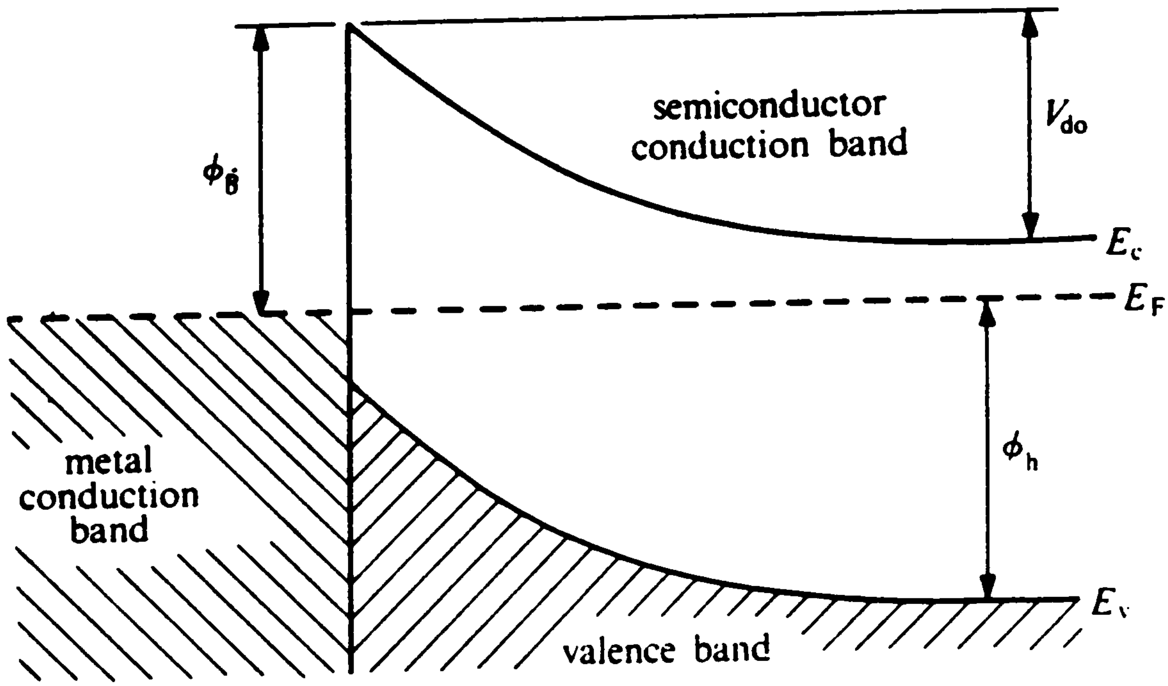


Figure 9. Band diagram for an ideal Schottky contact to an n-type semiconductor.

Table 7. Common Schottky metals to various semiconductors [taken from Ref. 70].

Semiconductor	Type	E_g (eV)	Ag	Al	Au	Cr	Cu	Hf	In	Mg	Mo	Ni	Pb	Pd	Pt	Ta	Ti	W
Diamond	*p*	5.47			1.71													
Ge	*n*	0.66	0.54	0.48	0.59		0.52		0.64			0.49	0.38					0.48
Ge	*p*		0.50		0.30				0.55									
Si	*n*	1.12	0.78	0.72	0.80	0.61	0.58	0.58		0.40	0.68	0.61		0.81	0.90		0.50	0.67
Si	*p*		0.54	0.58	0.34	0.50	0.46				0.42	0.51	0.55				0.61	0.45
SiC	*n*	3.00		2.00	1.95													
AlAs	*n*	2.16			1.20										1.00			
AlSb	*p*	1.63			0.55													
BN	*p*	7.50			3.10													
BP	*p*	6.00			0.87													
GaSb	*n*	0.67			0.60													
GaAs	*n*	1.42	0.88	0.80	0.90		0.82	0.72							0.84	0.85		0.80
GaAs	*p*		0.63		0.42			0.68										
GaP	*n*	2.24	1.20	1.07	1.30	1.06	1.20	1.84		1.04	1.13	1.27			1.45		1.12	
GaP	*p*				0.72													
InSb	*n*	0.16	0.18*		0.17*													
InAs	*p*	0.33			0.47*													
InP	*n*	1.29	0.54		0.52													
InP	*p*				0.76													
CdS	*n*	2.43	0.56	Ohmic	0.78		0.50					0.45	0.59	0.62	1.10		0.84	
CdSe	*n*	1.70	0.43		0.49		0.33								0.37			
CdTe	*n*		0.81	0.76	0.71										0.76			
ZnO	*n*	3.20		0.68	0.65		0.45		0.30					0.68	0.75	0.30		
ZnS	*n*	3.60	1.65	0.80	2.00		1.75		1.50	0.82				1.87	1.84	1.10		
ZnSe	*n*		1.21	0.76	1.36		1.10		0.91				1.16		1.40			
PbO	*n*		0.95						0.93			0.96	0.95					

*77 K.

characteristics. This is especially true with GaAs. It is therefore necessary that these effects be better understood so that a more complete model can be developed for the Schottky contact, and experimental results better understood.

2.1.2 Surface States

All materials have a boundary between themselves and their surroundings which defines the surface of the material. At the surface there are atoms which will have no neighbor to one side and the resulting dangling bond, along with other crystalline defects and surface contaminants, may contribute to the overall surface state distribution. To illustrate the electrical effect of surface states at semiconductor interfaces two cases will be considered, one where the semiconductor is in contact with a metal and the other where the surface is exposed to free space.

Consider the first case as shown in Fig. 10. E_o marks the surface state neutral level, i.e., if E_f is above E_o, the surface is acceptor-like (-) while if E_f is below E_o, the surface is donor-like (+) [71]. Consider an n-type semiconductor with E_f slightly below E_o. The surface will be positively charged, partially compensating the metal, reducing the space charge in the semiconductor. This reduces the band bending and thus the barrier height. With a high density of states any aberration of E_f from E_o would mean massive surface charge build up. The bands will therefore only bend enough to keep E_o close to E_f, otherwise this large uncompensatable surface charge would result. With the band bending so restricted, the barrier height is effectively

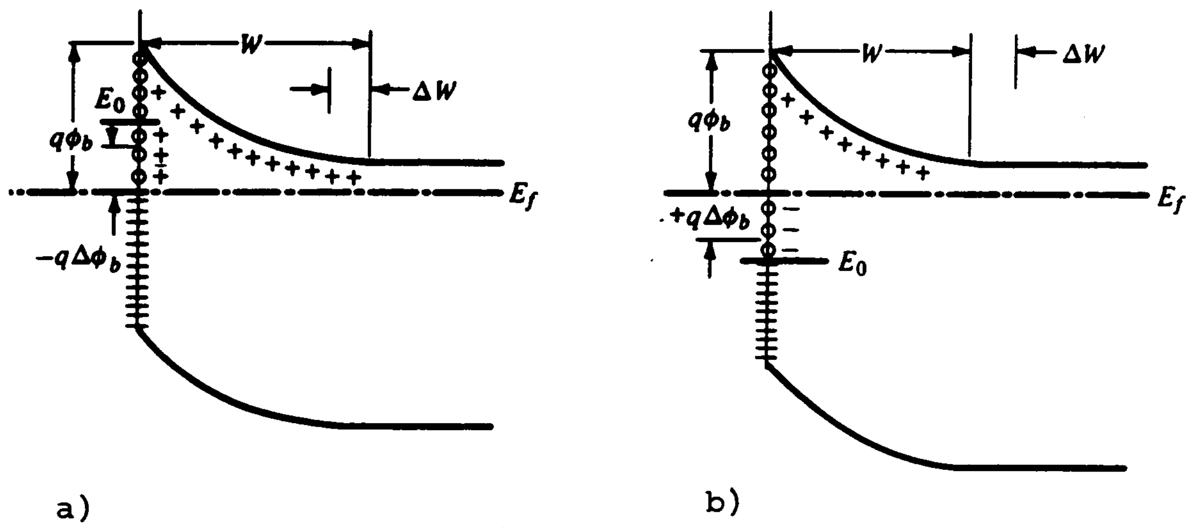


Figure 10. Band diagram of a semiconductor containing surface states (a) donor-like, (b) accetor-like [taken from Ref. 71].

"pinned" by E_o so that

$$\phi_B \simeq E_g - E_o \tag{3}$$

Similarly for the free surface, E_f must be maintained near E_o or else uncompensatable surface charge again results. This forces E_f to be pinned at E_o for the free surface [72].

Clearly, with large surface state densities the barrier height is almost independent of ϕ_M, completely deleting the ideal model results. Unfortunately GaAs has large enough surface states to have a pinned barrier height regardless of metal type. This is one of the major problems with GaAs and much research is presently dedicated to passivating the GaAs surface in an attempt to unpin the Fermi level [73,74].

2.1.3 **Non-ideal**

With regard to the surface states and the presence of interfacial oxides on GaAs (~20 Å) (whose electrical effects will be seen shortly) a non-ideal model has been given in Fig. 11. This is the Bardeen model of the real metal-semiconductor interface. From this diagram ϕ_B may have some bias dependence due to voltage dropped across the interfatial oxide layer given by

$$\phi_b = \phi_B - \alpha\, E_{max} \tag{4}$$

with

$$\alpha = \frac{\delta \varepsilon_S}{\varepsilon_i + q\delta D_S} \tag{5}$$

where ε_S is the semiconductor permitivity, ε_i is the insulator permitivity,

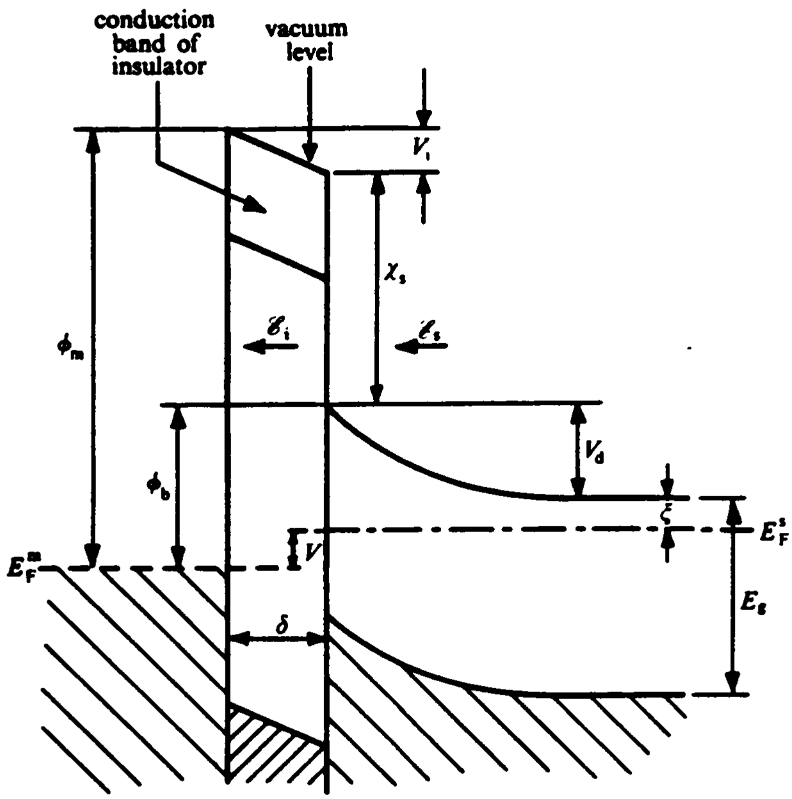


Figure 11. Bardeen model of a non-ideal Schottky contact [taken from Ref. 68].

δ is the insulator thickness and D_S is the density of surface states [75]. This bias dependent barrier height effect is usually small however.

Image force lowering can also decrease the barrier height. The effect is to lower the barrier by

$$\Delta\phi = \left(\frac{qE}{4\pi\varepsilon_S} \right)^{1/2} \tag{6}$$

where E is the field maximum. The barrier peak occurs at

$$x_m = \left(\frac{9}{16\pi\varepsilon_S E} \right)^{1/2} \tag{7}$$

where X_m is the location from the surface in cm of the barrier height maximum. The lowering is on the order of a few tenths of a volt for moderate fields [76].

2.1.4 Current-Voltage

Figure 12 shows a schematic of the Schottky contact with the forward biased current mechanisms numbered. The numbers correspond to:

1. Thermionic emission
2. Tunneling
3. Recombination in the space charge layer
4. Hole injection or recombinaton in the neutral region

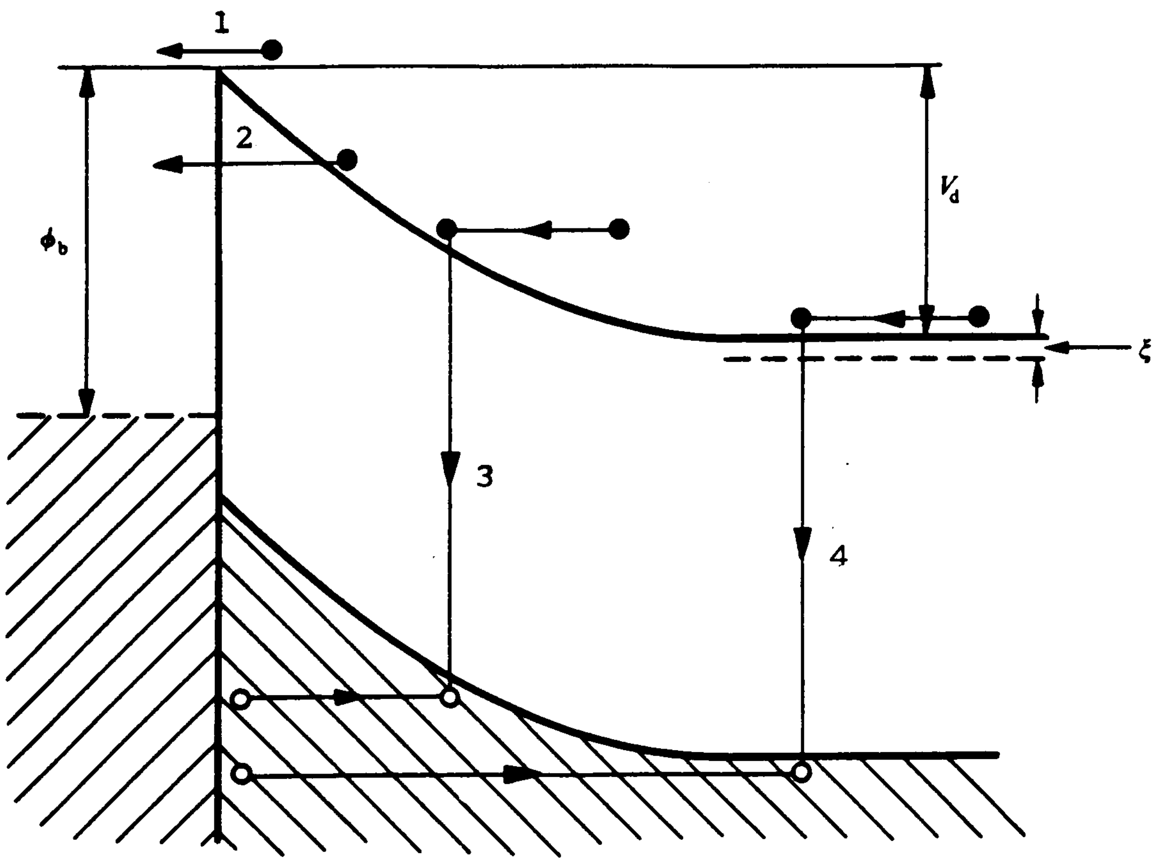


Figure 12. Schematic indicating current components in forward bias [taken from Ref. 68].

For forward and "near" reverse bias, thermionic emission dominates the other components and is considered the most correct theory of current transport. Sze [3] has developed a combination of diffusion and thermionic emission which will be discussed later.

Thermionic emission currents originate from electrons which gain or have enough thermal energy to surmount the contact barrier. From kinetic theory the number of electrons incident on the barrier is

$$J_{SM} \text{ or } J_{MS} = \frac{qn\bar{v}}{4} \tag{8}$$

where n is the electron density and $\bar{v}$ is their average velocity. For electrons in the semiconductor

$$n = N_c \exp\left[-q(\phi_B - V)/kT\right] \tag{9}$$

giving

$$J_{SM} = q\,\frac{Nc\bar{v}}{4}\exp\left(-q(\phi_B - V)/kT\right) \tag{10}$$

At equilibrium and V = 0 the number of electrons incident from the metal side must be equal to those incident from the semiconductor side. Thus

$$J_{MS} = \frac{qNc\bar{v}}{4}\exp\left(-q\phi_B/kT\right) \tag{11}$$

The total current, $J = J_{SM} - J_{MS}$, is given by

$$J = \frac{qNc\bar{V}}{4}\exp\left(-q\phi_B/kT\right)\left[\exp(qV/kT) - 1\right] \tag{12}$$

Considering all of the electron's energy to be kinetic in one direction for a Maxwellian distribution gives

$$\bar{v} = \left(\frac{8kT}{\pi m^*} \right)^{1/2} \tag{13}$$

and substituting for Nc, the effective density of states in the semiconductor gives, after simplification,

$$J = A^* T^2 \exp\left(-q\phi_B/kT\right)\left[\exp(qV/kT) - 1\right] \tag{14}$$

where A* is the Richardson constant given as

$$A^* = 4\pi m^* q k^2/h^3 \tag{15}$$

where m* is the electron effective mass, q the electron charge, k is Boltzmann's constant and h is Planck's constant. With $V > 3kT/q$ the current reduces to

$$J = J_{sat} \exp(qV/kT) \tag{16}$$

This is very similar to the p-n diode equation with the only difference being that the Schottky diodes have a different and often larger reverse saturation current J_{sat}.

If the effect of image force and bias dependency of the barrier height is considered as

$$\phi_{BE} = \phi_B - \Delta\phi + \beta V \tag{17}$$

where β is the coefficient determining the strength of bias dependency, then after substituting in Eqn. 14 with $1/n = 1 - \beta$ and reducing gives

$$J = J_{sat} \exp(qV/nkT) \tag{18}$$

where n is called the diode ideality factor and is often used as a measure of diode quality [77].

Sze [3] has developed a hybrid model which considers both thermionic emission and diffusion theory. The resulting Current-Voltage relation is

$$J = \frac{qN_c v_R}{1 + v_R/v_D} \exp\left(\frac{-q\phi_B}{kT}\right)\left[\exp\left(\frac{-qV}{kT}\right) - 1\right] \tag{19}$$

where v_R is the surface recombination velocity at the interface and v_D is the effective diffusion velocity for electrons transported from the edge of the space charge layer. If $v_D \gg v_R$ than thermionic emission applies, if $v_R \gg v_D$ the process is diffusion limited. This paper will consider thermionic emission to dominate since it has been shown to be a more correct description of Schottky diode current flow [68].

Further modificiation exists for A*. Considering the possibility of electrons emitted from the semiconductor being scattered back, the Richardson constant may be modified to

$$A^{**} = \frac{f_p f_q A^*}{1 + f_p f_q v_R/v_D} \tag{20}$$

But since it is assumed that thermionic emission applies ($v_D \gg v_R$), we obtain

$$A^{**} = f_p f_q A^* \tag{21}$$

where f_p represents the fraction of electrons scattered back by phonon interaction in the metal and f_q the fraction quantum mechanically reflected from the barrier [78].

Reverse breakdown generally occurs via an avalanche process and is usually expressed for a one-sided abrupt junction, (a Schottky can be modeled as a p^+n), as

$$V_B = \frac{\varepsilon_S E^2_{MAX}}{2qN_D} \tag{22}$$

where V_B is the reverse breakdown voltage, E_{max} the peak field at breakdown and N_D the dopant density. The breakdown voltage is also dependent upon the carrier mobility in that the velocity the electron attains under the applied field must be large enough for impact ionization to occur. For lightly doped n-type semiconductors with $N_D \approx 5 \times 10^{16} cm^{-3}$, V_B, as calculated from the above equation, is approximately 11 V [79].

A thin interfacial layer can have many adverse effects on the Current-Voltage characteristics in both forward and reverse bias (see Fig. 13). Since electrons must tunnel through this layer, the current varies with layer thickness for a given applied bias. This leads to a

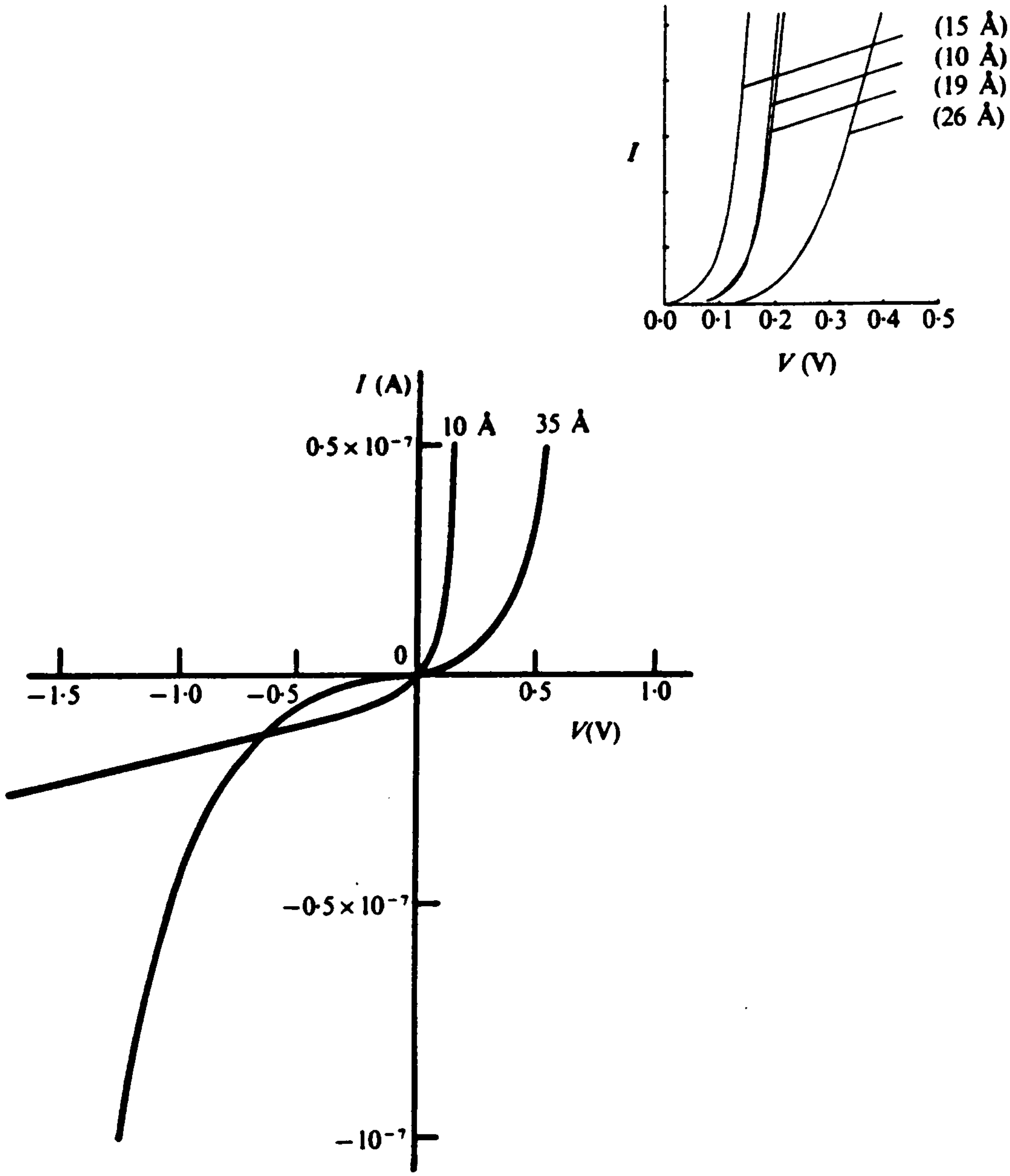


Figure 13. Effect of thin interfacial layer on I-V characteristics of Schottky diodes [taken from Ref. 80].

softening of the forward and reverse characteristics as shown. Also, part of the voltage is dropped across this layer, which causes a bias dependent barrier height as discussed earlier. This results in non-saturation of the reverse bias current and affects the ideality factor.

Surface states can also affect Current-Voltage characteristics by introducing tunneling states at the interface and by thinning the barrier which also aids in tunneling in both directions. Field crowding can also occur because of uneven charge distribution near the surface causing an early breakdown.

2.1.5 **Capacitance-Voltage**

For all intensive purposes the Schottky diode may be treated as a one-sided abrupt junction with a space charge layer on the n-side as shown in Fig. 14. Solving the Poisson equation across the space-charge layer results in

$$C = \left(\frac{q\varepsilon_S N_D}{2}\right)^{1/2} (V_{bi} + V - kT/q)^{-1/2} \tag{23}$$

This may be related to the parallel plate approximation as

$$C = \frac{\varepsilon_S}{W} \tag{24}$$

where W is the space-charge width. Solving for W using the above two equations gives [81]

$$W \simeq \left(\frac{2\varepsilon_s(V_{bi} + V)}{qN_D} \right)^{1/2} \tag{25}$$

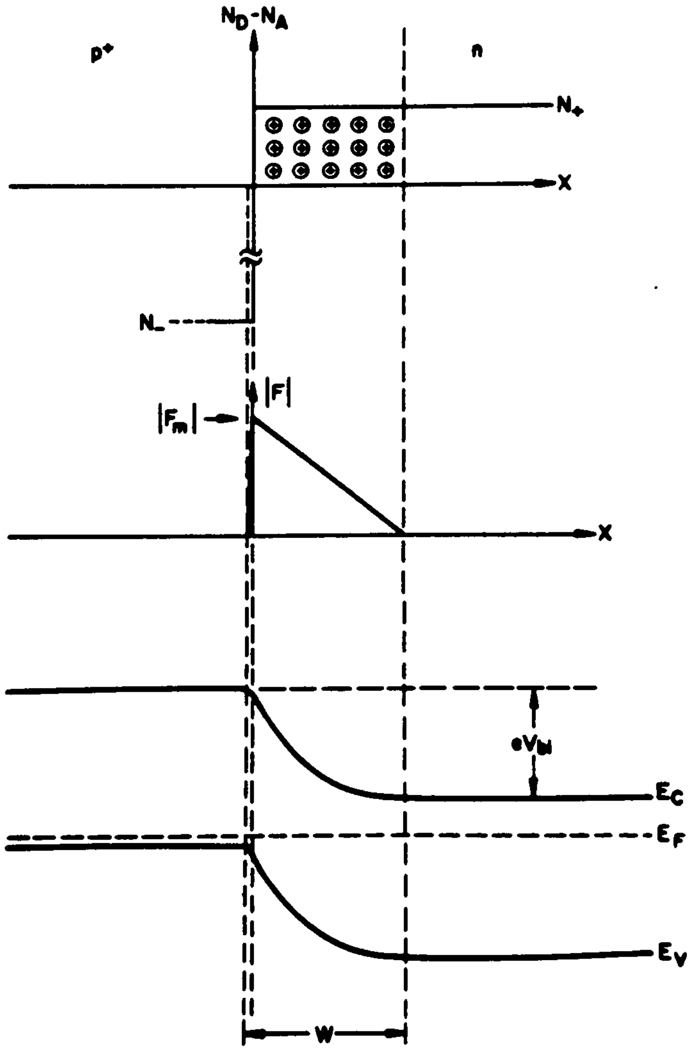


Figure 14. Schottky contact indicating space charge region [taken from Ref. 81].

An interfacial layer modifies dependence of the space charge in the depletion region on the bias voltage through barrier variations and voltage drops across the layer. These are usually nonlinear effects. Surface states, because of their charged nature, vary the zero bias capacitance and C-V characteristic of a surface-state-free or ideal diode, since space-charge compensation is either reduced or increased depending on the surface charge state and applied voltage affecting W.

Deep traps in the bulk of the semiconductor can also have effects on the capacitance of the Schottky diode depending on the trap depth, characteristics and location with respect to the Fermi level. Assume a deep donor in two separate cases. If the trap emission frequency $\omega_T (= 1/\tau_T)$ is less than the measuring signal frequency ω_s the traps will not respond fast enough and the measured capacitance will be due to N_D only. If however $\omega_T < \omega_s$ the traps can respond and

$$C = \frac{\varepsilon_S (N_D + N_T)}{N_D W + N_T Y} \qquad (26)$$

where Y is the trap width in the space charge layer [82].

Deep traps also vary their occupation and emission rate with temperature, affecting the transient and thermal capacitance behavior of the diode. This is exploited in the Deep Level Transient Spectroscopy (DLTS) measurement which will be explained more fully in the next chapter.

2.2 Ohmic Contacts

Ohmic contacts are linear in that they generally obey Ohm's Law and may, in general, be formed in two ways to semiconductors. One way, for n-type semiconductors, is to choose a metal with a work function less than the semiconductor work function ($\phi_M < \phi_S$). The resulting band diagram when the two are joined is shown in Fig. 15 where it can be seen that no effective barrier exists looking from either the semiconductor to the metal or from the metal to the semiconductor. This results in ohmic behavior [83]. However, as was seen earlier, GaAs tends to have a somewhat uncontrollable oxide layer and surface state distribution, effectively pinning the semiconductor Fermi level. Thus regardless of the metal work function a rectifying contact will most likely result, making this method of ohmic contact formation to GaAs ineffective.

Another method is to form a so-called tunnel contact as seen in the band diagram given in Fig. 16, which has been shown to give ohmic behavior. This is indeed the common contact method used in the GaAs industry, and in this research. Table 8 illustrates a variety of metal systems used to form ohmic contacts in this manner. Au-Ge-Ni is by far the most popular system used to date and is the ohmic contact system that was used in this study. (This technique was also used to replicate the Texas Instruments diodes.)

Fabrication takes place in several steps [85], as follows

1. An 88:12 Au:Ge alloy is deposited.
2. A Ni cap is deposited on top of the Au:Ge.
3. The contact is annealed in forming gas (95% N_2:5% H_2) for several minutes at ~440°C.

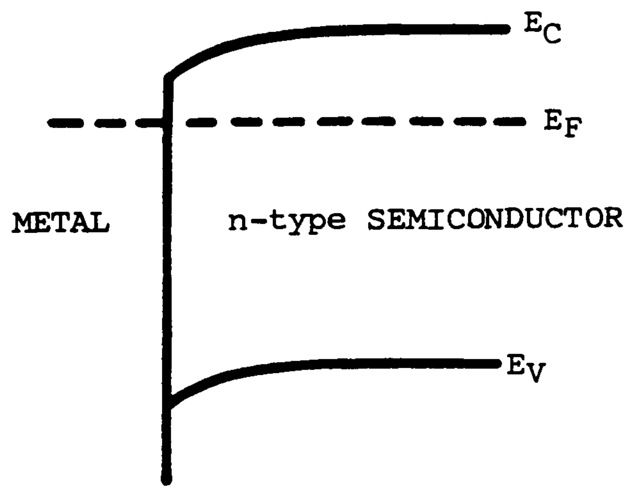


Figure 15. Typical ohmic contact for $\phi_M < \phi_S$ for an n-type semiconductor.

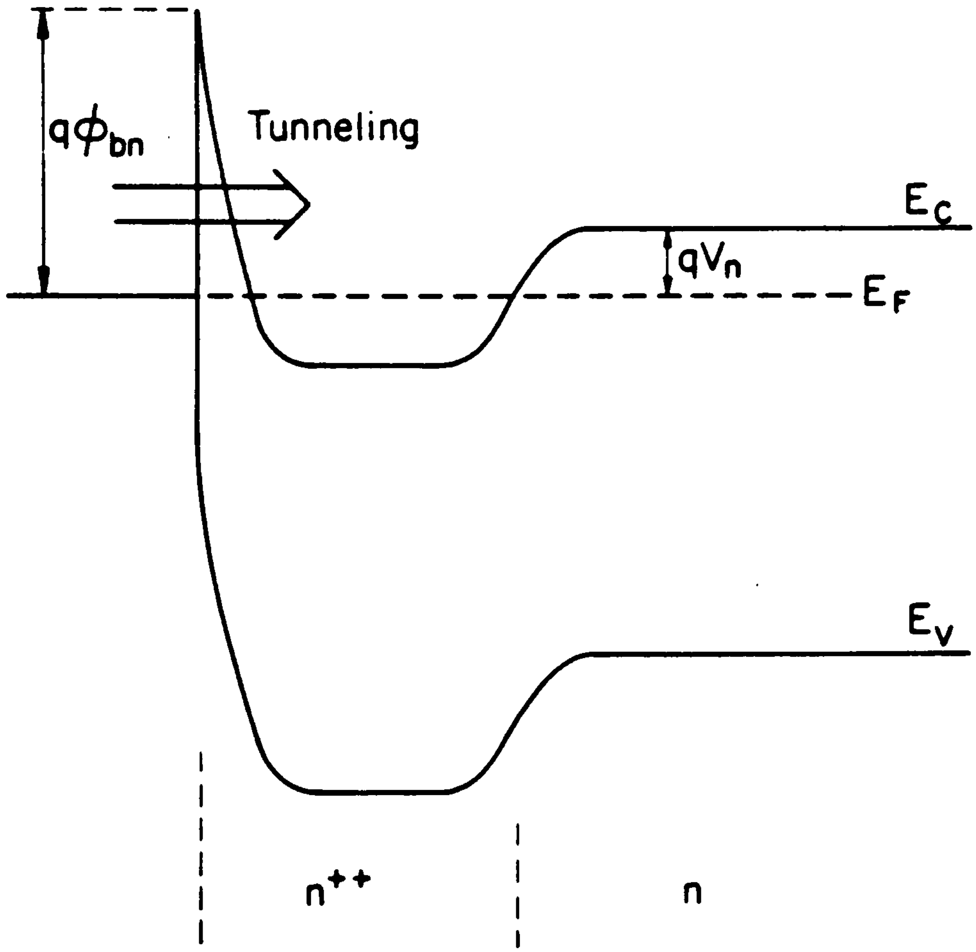


Figure 16. Tunnel type ohmic contact [taken from Ref. 83].

Table 8. Common ohmic metal systems used on GaAs [taken from Ref. 84].

n-Type GaAs	Metal System
Ag-In	75% Ag, 25% In; sinter 500°C
Ag-In-Ge	90% Ag, 5% In, 5% Ge; alloy 600°C
Ag-Sn	98% Ag, 2% Sn; alloy 550–650°C
	33% Ag, 67% Sn; alloy
Au-Ge	12% Ge (alloy); alloy 450°C
	Au overlayer
Au-Ge-Ni	12% Ge (alloy), Ni; alloy 480°C
Au-Ge-In	
Au-Si	alloy 425°C
Au-Sn	20% Sn; alloy 450°C
Au-Sn-Ni-Au	alloy 300°C
Au-Te	10% Te; laser alloy
	2% Te; alloy 500°C
In	300°C melt
In-Al	alloy 320°C
In-Au	90% In; alloy 550°C
In-Ni	Ni plated to In
Pd-Ge	sinter 500°C for two hours
Sn-Ni	Ni plated to Sn; alloyed
Sn-Sb	4% Sb; alloy 300-350°C
p-Type GaAs	
Ag-In	25% In; alloy 500°C
Ag-In-Zn	80% Ag, 10% In, 10% Zn; alloy 600°C
Ag-Zn	90% Ag, 10% Zn; alloy 450°C
Au-Be	1% Be;
Au-Zn	
In-Zn	(to S.I. GaAs)

The anneal step drives the Ge into the near surface of the GaAs, heavily doping it. This forms the narrow, easily tunneled-through barrier, as illustrated earlier. The role of the Ni is less clear however, but it is known that it acts as a wetting agent for the Au during the anneal, so that the Au contact does not ball-up and loose adhesion. It also appears that the Ni can act as a diffusion barrier to Ga, which is gettered by the Au during the anneal. The contact resistance is very sensitive to alloy temperature and time, and some success has been made improving the contact resistance and stability by depositing a thin (~10Å) Ni layer before Au:Ge [86].

Contact resistance, tunneling probability, reproducibility and reliability of ohmic contacts are subjects of much research and analysis which will not be taken up or presented here. It will be assumed that a small change in contact resistance will not affect the measured results of IBE damage and in fact can be neglected since ohmic contacts were made before IBE was done. The actual processing of ohmic and Schottky contacts for this research is discussed in the next chapter.

Chapter 3: Experimental Methods

This chapter details the methods used to study the effects of IBE. The fabrication process is discussed in detail along with the measurement system and the parameters that are derivable from electrical measurement. Also described are the experimental groups into which the samples were divided for study. DLTS is described in some detail as well.

3.1 Schottky Diode Fabrication

As mentioned earlier, the Schottky diode was chosen in this study due to its sensitivity to surface effects, and its relatively simple fabrication. What follows is a detailed description of the fabrication process used to make diodes for experiment.

Wafers were initially purchased from Airton Inc. but were later obtained from Morgan Semiconductor, since Texas Instruments uses Morgan as a primary wafer vendor. They were standard 2" diameter LEC GaAs wafers 15mils thick, doped n-type in the melt to $\sim 5 \times 10^{16} cm^{-3}$ with Si, and cut 2^{o} off (100). Electron mobility was on the order of $4200 cm^{2}/V \cdot s$ as given in the manufacturer's data sheet provided by Morgan. Wafers were received chemically and mechanically polished on one side for device fabrication. The samples were initially cut from the wafer to .5" x .5" squares and marked for crystal orientation as shown in Fig. 17, using a diamond scribe. After cutting, the samples were subjected to the cleaning, metalization and packaging process given in Fig. 18 [87].

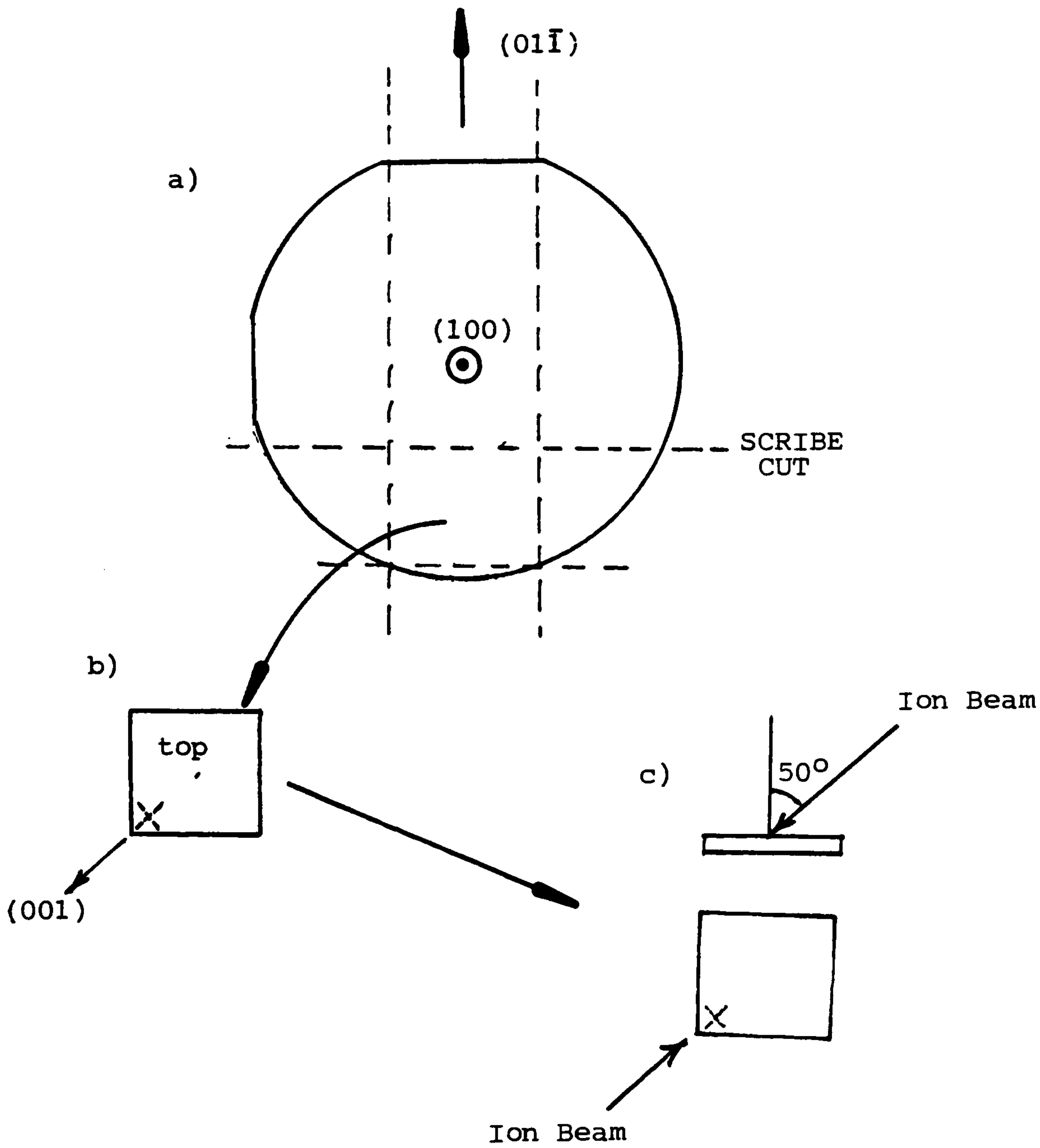


Figure 17. Diagram of as received wafer indicating (a) crystal orientations, (b) cut pieces and (c) IBE angles.

Name:

Date:

SAMPLE #:

WAFER #:

PROCESSING(etch, cleaning, anneal, metals, special, ..., etc./date):

1. (/): 10 min. in hot tetrachloroethylene

2. (/): 10 min. in hot methanol

3. (/): hot HCl dip : D.I. rinse

4. (/): 4 min. in 8:1:1 (H_2SO_4:H_2O_2:H_2O) : D.I. rinse : dry

5. (/): Au/Ge(88%/12%) evaporation about 2000Å at slow rate

6. (/): Ni evaporation about 700Å

7. (/): ohmic anneal 2 min. at 440°C in forming gas

8. (/): 4 min. 8:1:1 or until white haze subsides : D.I. rinse and dry

9. (/): 10 min. HCl : D.I. rinse and dry

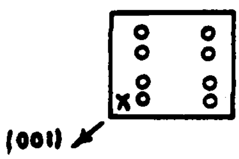
10. (/): Ar^+ ion etch at x keV, x Amps/cm^2, x min. : store in argon atmosphere when complete

11. (/): Al Schottky evaporation about 2000Å at slow rate

12. (/): scribe cut diodes : mount in TO-8 package with 1 mil Au wire and silver epoxy : cure in oven at 65°C for 3hrs

DEVICE/pinout: pin 1=Al Schottky, pin 2=Au/Ge ohmic

COMMENTS: wafer marked for orientation



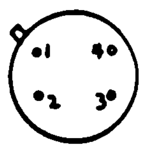


Figure 18. Schottky diode processing flowchart.

The first step involved a cleaning and degreasing process to remove any fingerprints and dirt present from shipping and handling. This involved 10 minutes in a hot trichloroethylene bath followed by a 10 minute hot methyl alcohol wash.

Samples were then rinsed in deionized water and dipped briefly in hot (~40^{o}C) 10% HCl solution. This was done to clean and remove any oxides (Ga_2O_3, As_2O_3, etc.) present; a DI rinse followed.

As suggested by Texas Instruments, the cleaned as-received surface was not used for fabrication. Instead this surface was exposed for 4 minutes to an 8:1:1 H_2SO_4:H_2O_2:H_2O chemical etch which removes GaAs at approximately 1μm/min. This yielded clean surfaces with ~4μm removed which were then DI rinsed and surface-tension-pulled dry.

Four samples, processed in parallel, were then placed in an Al sample jig which was fitted with two stainless steel masks consisting of 4 evaporation holes and 2 alignment holes (Fig. 19) The masks were arranged so that ohmic contacts were evaporated using holes 1/16" in diameter. Schottky contacts were evaporated using the bottom alignment holes on the 1/32" dot diameter mask. This produced 4 diodes per piece or 16 diodes per run.

The ohmic contacts were produced by placing the above jig into a Denton Vacuum evaporator, approximately 9" above two tungsten filaments containing an 88:12 Au:Ge slug of approximately 5 grams in one boat, and 99.999% pure Ni in the other. (Both metals were purchased from Materials Research Corporation.) After achieving a vacuum of 6μtorr, approximately 2000Å of Au:Ge was deposited, as measured by a Kronos Model QM-311 thickness monitor. Under the same vacuum, 700Å of Ni was next

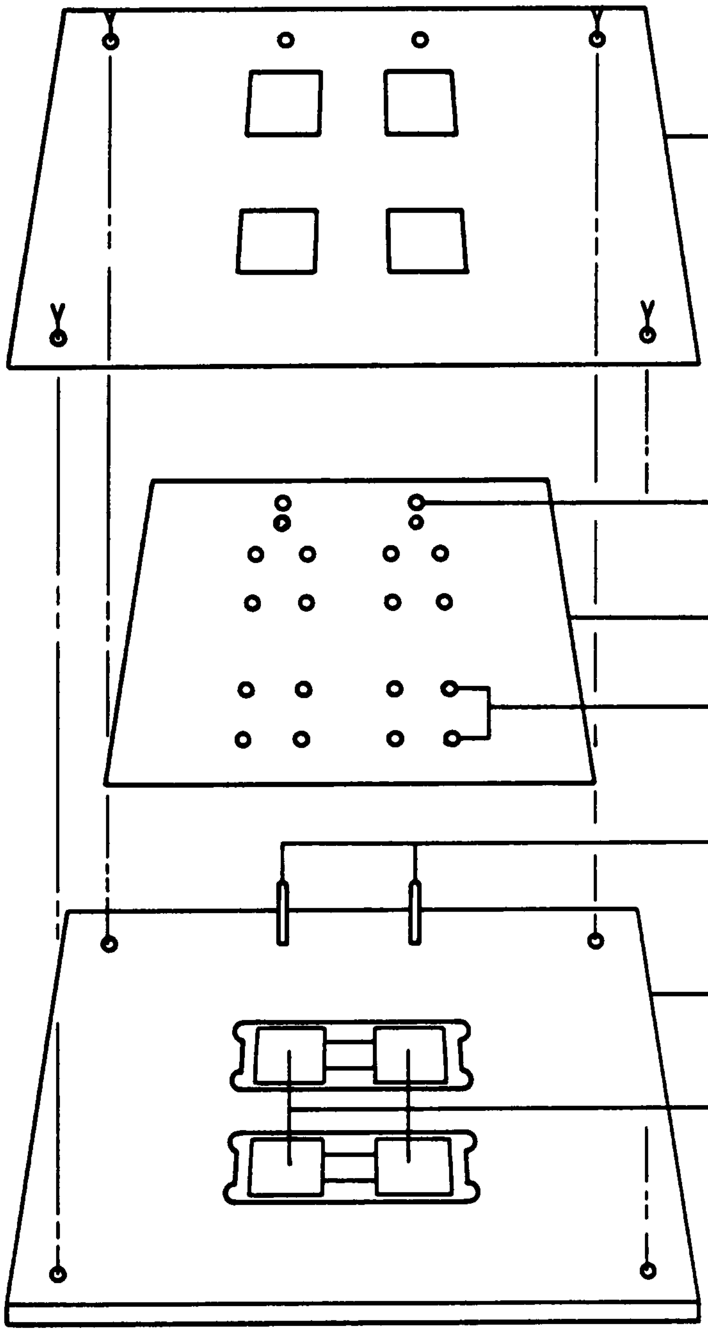


Figure 19. Evaporation jig and mask arrangements for depositing ohmic and Schottky contacts.

was next deposited on top of the Au:Ge. The chamber was then vented with dry N_2 and the samples annealed at 440^oC for 2 minutes in forming gas (95% N_2:5% H_2) (This is the Ge drive-in step explained in Chapter 2).

It was found that Schottky contacts deposited on these annealed surfaces produced inferior, unreproducible diodes. Therefore a second 4 min. 8:1:1 H_2SO_4:H_2O_2:H_2O etch was incorporated to remove any possible surface damage introduced by the anneal. Upon exposure to the etch a white, "milky" coating appeared on the sample surfaces. This is called the "white haze" in industrial circles and is possibly the result of micro-surface cracks exposed by the etch [88]. After sufficient time (~5 min.) all surfaces recovered to their near optically clean appearance.

The samples were next cleaned in 1:1 HCl for 10min. and placed in a Perkin-Elmer 5300 XPS system for Ar^+ etching. Energies between .5keV and 3keV were used with current densities correspondingly between $0.5\mu A/cm^2$ and $60\mu A/cm^2$. Ion etching took place along the (001) direction at 50^o off normal, as oriented by an "X" on the sample back (refer to Fig. 17). Fluences of $10^{14}cm^{-2}$ and $10^{16}cm^{-2}$ were used, with fluence defined as fluence = current density • etch time. ESCA measurements sometimes followed the IBE, and such results will be presented where appropriate.

Al or Au Schottky contacts were then deposited using the 1/32" dot mask and jig described earlier. After approximately 2000Å of Schottky metal deposition, the samples were electrically tested for quality using a Tektronix 575 Transistor-Curve Tracer after which good individual diodes were scribed and cut from the larger pieces and mounted into a TO-8 package with Si based thermal grease. Electrical contact was made using 1mil Au wire and room temperature cured thermal epoxy (ACME E-Solder). A resulting device is illustrated in Fig. 20. The Schottky diodes were then ready for electrical test.

3.2 Automated Electrical Analysis (MEDUSA)

Due to the large number of samples and measurements necessary to study this subject and the requirement that data be reproducible, an automated electrical test station was designed and constructed. This station was given the name MEDUSA for Materials and Electronic Device Unified System Analyzer.

3.2.1 Hardware

The hardware layout is given in Fig. 21. All instruments including the magnet (through an optical power relay) are connected to the IEEE 488 bus controlled by the IBM-AT using a Scientific Solutions board. Each piece of equipment performed several tasks:

HP 4280A: (Capacitance Bridge)	C,G; C,G-V; C,G-t; DLTS (coupled with HP 8112A pulse generator)
HP 4140B: (Current Meter and Voltage Source)	I-V, C-V
Keithley 195A DVM:	Van de Pauw (coupled with 4140B, 4280A, Varian Magnet), 4-pt. Resistivity (coupled with 4140B)

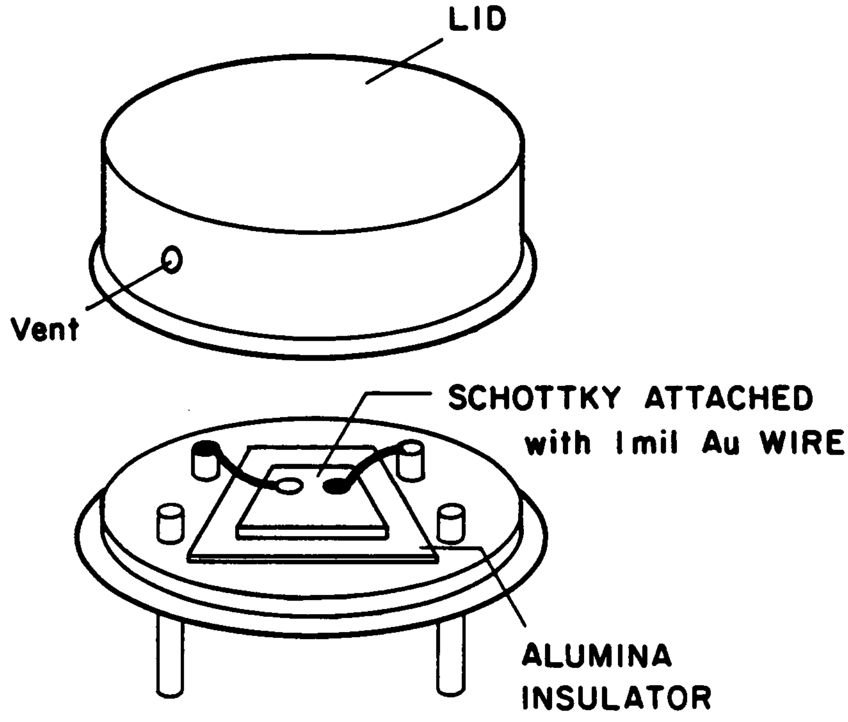


Figure 20. TO-8 package and attached diode.

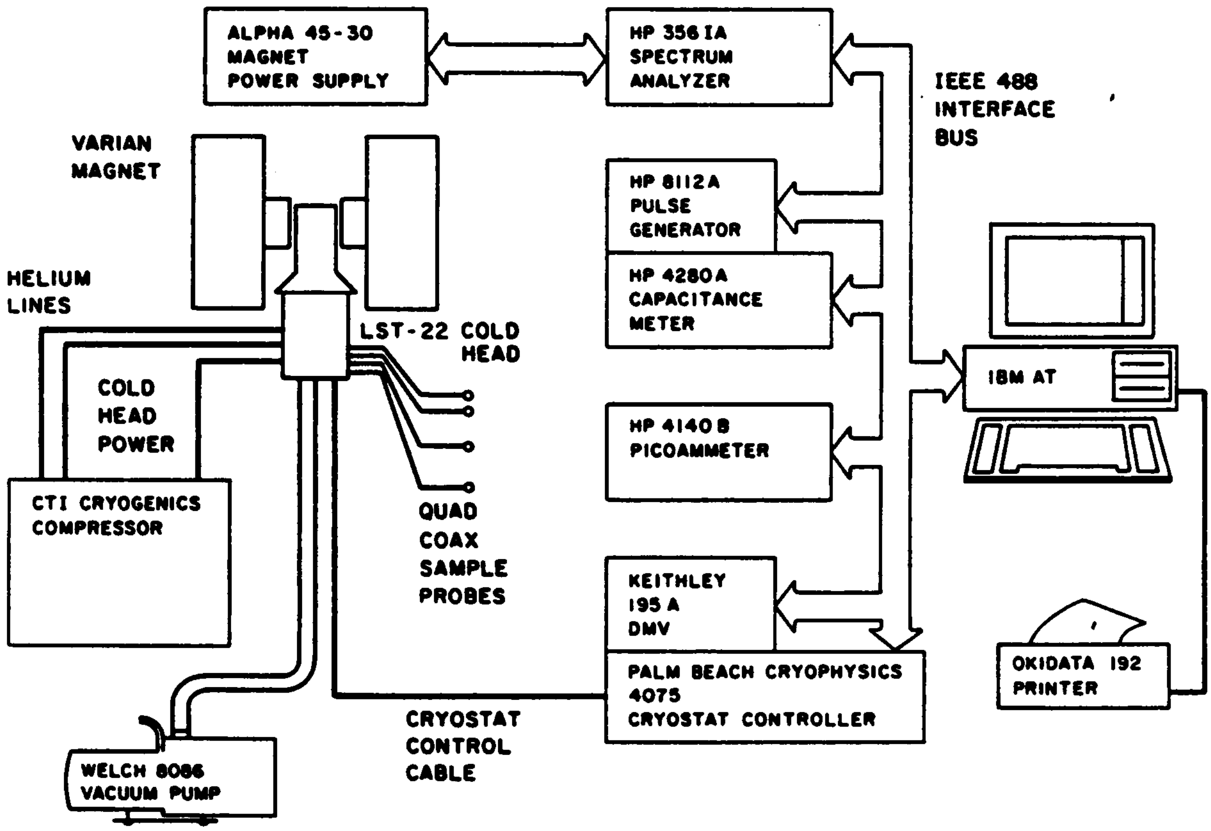


Figure 21. MEDUSA hardware layout.

3.2.2 Software

The software consists of three separate levels: (1) Parameter entry and error check, (2) Experimental run and data collection, (3) Data processing and graphical display. (Flowcharts and the actual software are given in Appendix A.) Graphs derivable from the system include:

1. Hand Graph Entry
2. Capacitance vs. Time
3. Conductance vs. Time
4. Capacitance vs. Temperature
5. Conductance vs. Voltage
6. Capacitance vs. Temperature (TSCAP)
7. Conductance vs. Temperature
8. $1/C^2$ VS. Voltage
9. Dopant Profile
10. Barrier vs. Temperature
11. DLTS Spectrum
12. Current vs. Voltage
13. Log (I) vs. Voltage
14. Ln(I) vs. qV/kT
15. $Ln(I_f/T^2)$ vs. q/kT
16. I_{sat} vs. Temperature
17. Ideality Factor vs. Temperature
18. Resistivity vs. Temperature
19. Mobility vs. Temperature
20. Carrier Concentration vs. Temperature

21. Activation % vs. Temperature

22. $1/C^3$ vs. Voltage

23. 4-point Resistivity vs. Temperature

24. $\ln(I_{sat})$ vs. Temperature

3.3 Electrical Measurements

It is appropriate at this point to discuss how some of the above graphs produce vital data characterizing the diode. The major parameters to be considered for the diodes were ϕ_b, n, I_{sat}, $N_D(x)$, $E_{Traps}(E_T)$, $N_{Traps}(N_T)$, $V_{Turn-on}(V_T)$, $V_{breakdown}(V_B)$, G and C. Table 9 gives the parameter and graphs from which they are derived [89]. Mobility measurements were not done on these samples since any surface damage would be shunted by the low resistivity bulk. The trap parameters were derived from the DLTS measurement, which deserves further explanation (along with TSCAP).

DLTS (Deep Level Transient Spectroscopy) was a major part of this research, and the principles by which it works need further clarification. Essentially DLTS is performed by reverse biasing a diode and pulsing it to zero bias, after which the capacitance is monitored as a function of time at a given temperature. During the pulse, the traps in the reduced space charge fill, changing their charge state, thus effecting the capacitance. These trapped electrons are thermally released exponentially at a rate dependent upon the sample temperature after the bias pulse has subsided. The thermal release rate for electrons is given by

Table 9. Schottky diode parameters and their graphical derivations.

Parameter	Graphical Derivation
Φ_b	x-intercept of $1/C^2$ vs. V, slope of $\ln(I_f/T^2)$ vs. q/kT (activation energy, $\Phi_b = V_f/n$-slope)
n	slope of ln(I) vs. qV/kT (n=1/slope)
I_{sat}	y-intercept of ln(I) vs. qV/kT
V_B	45° slope point of reverse I-V
V_T	extrapolated x-intercept of I-V
$N_D(x)$	$2/q\epsilon[-1/\text{slope of } 1/C^2]$ with W calculated from depletion width
E_T	$23.7kT_{peak}$ of DLTS spectrum
N_T	$2[\delta C/C]N_D$ where δC comes from DLTS peaks in spectrum

$$\alpha = \nu \exp[-(E_C - E_T)/kT] \quad (27)$$

where ν is the release frequency. By sweeping the sample temperature, measuring the capacitance vs. time after pulse biasing, and looking for a specific rate of decay, (rate window), a DLTS spectrum is obtained. This process is illustrated in Fig. 22.

The spectrum is a plot of the magnitude of the capacitive transient versus temperature, where majority carrier traps are indicated by negative peaks in the spectrum. The magnitude of the peaks may be used to determine the trap density through the relation

$$\frac{\Delta C}{C} = -\frac{N_T}{2N_D}\left[1 - \frac{2\lambda}{W(V)}\left(1 - \frac{C(V)}{C(0)} - \frac{C(V)}{C(0)}\right)^2\right] \quad (28)$$

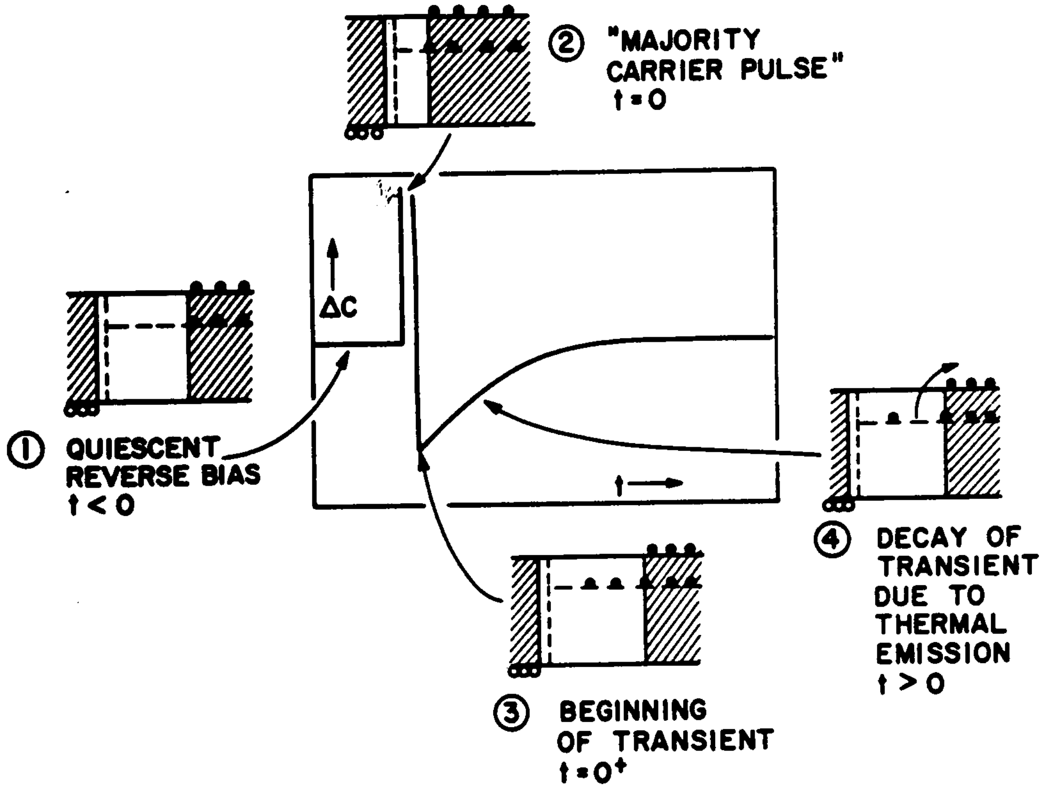
where λ is the so-called edge region thickness, W the space charge thickness, N_T the trap density, C the reverse biased capacitance and ΔC the transient capacitance given by

$$\Delta C = C(t_1) - C(t_2) \quad (29)$$

where t_1 and t_2 are the boxcar integrator times. If the edge region is neglected then the trap density N_T can be found from the relation

$$N_T = 2N_D \left| \frac{\Delta C}{C} \right| \quad (30)$$

assuming $\Delta C < C$. DLTS is much more sensitive to traps located near the



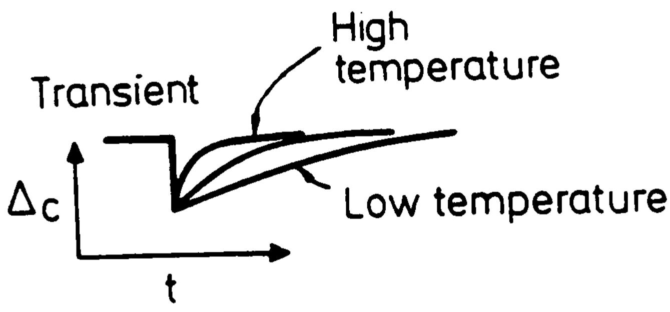


Figure 22. DLTS pulse biasing and electron trapping process [taken from Ref. 90].

edge of the space charge layer, as can be seen in the relation

$$\left| \frac{\Delta C}{C} \right|_x = - \frac{n(x)}{N_D \omega^2} x \Delta x \tag{31}$$

The trap energy may be determined using an Arrhenius plot of log of the transient decay rate versus 1/T. This is implemented in DLTS by scanning the rate window which in turn changes the peak temperature accordingly. A simpler approximate method is to fix the rate window at $50 s^{-1}$, giving the relation

$$E_T = 23.7\ kT_M \tag{32}$$

where $\nu = 10^{12} s^{-1}$ and T_M is the peak temperature. The above calculated energies are accurate to ±10% of their actual values.

The rate window is implemented in the computer by simulating a double boxcar integrator set-up where a capacitance sample is taken at time t_1 and again later at t_2. The DLTS spectrum will then peak when

$$\tau_M \simeq (t_1 - t_2)/\ln(t_1/t_2) \tag{33}$$

This relation was used by fixing t_1 and t_2 so that $1/\tau_M = 50 s^{-1}$, as required for Eqn. 32.

Trap scattering crossections, σ_e and σ_h, may be found by observing peak magnitude changes due to pulse bias width variations, but were not done in this study.

The thermally scanned capacitance, TSCAP, was done in parallel with the DLTS as a cross-check. During a thermal scan, a step can be observed in the capacitance when a trap goes from being mostly full to mostly empty. The trap energy may be calculated using the relation

$$E_T = 30.7\ kT_{step} \qquad (34)$$

where 30.7 has been linearly interpolated from Lang [90] and T_{step} is the mid-point of the capacitance transition. However TSCAP is not as sensitive to deep traps as DLTS since the trap step may be superimposed on a large slope, and since it is not a differential measurement, the capacitance change may be difficult to detect.

3.4 Sample Grouping

Having discussed sample preparation and the electrical test procedures it is necessary to explain how these were applied in this study to deduce electrical effects of IBE. The samples were broken up into the following groups:

1. Test to indicate differences in wafer vendors, and metals used for Schottky contact.
2. IBE with energy between 0.5keV and 3keV at constant fluence.
3. IBE at 0.5keV and 3keV at two different fluences.
4. Chemical etch after IBE and before and after Schottky deposition at 0.5keV and 3keV constant fluence.
5. Anneal 0.5keV and 3keV samples before Schottky deposition.

Group 1 was a catch all for other sundry variables which might influence the main diode studies. Things such as wafer and Schottky metal variations were looked at briefly.

Group 2 was meant to determine the effects of energy on the diodes. All diodes were fabricated directly on top of the IBE surface with fluence held constant during bombardment so that only the effects of beam energy would be seen.

Group 3 was done to discriminate the effects of ion fluence at both the lowest and highest ion energies used in Group 1. It must be noted here that the fluence was studied, meaning beam currents and exposure times varied between the 0.5keV and 3keV samples.

Group 4 was used for two major studies. The first was to determine the depth of the IBE damage by successively building diodes on top of chemically reduced surfaces. The second study was meant to decouple the role of surface damage between contact damage by chemically removing the region between the ohmic and leaving damage under the Schottky contact only, neglecting undercutting.

Group 5 was done to see if damage could be repaired by a brief, low temperature (350^{o}C) thermal anneal. Recovery could indicate the degree of amorphicity and the role of any surface oxides.

As experiments proceeded, a few side tests were performed as an aid in understanding the data. These tests and their results will be mentioned in the next Chapter where appropriate.

Chapter 4: Results and Discussion

The results of the Group 1 experiments are reported first, since chronologically they were performed first. Also the results of this group helped to standardize the rest of the measurements. Each table and curve represents the most typical results obtained for each sample treatment described with samples starting from the whole wafer. At least two pieces were tested for each experiment (reproducibility of results confirmed) ranging up to 8 pieces for some studies.

4.1 Vendor and Metal

As mentioned earlier, wafers from Airtron were used initially to fabricate Schottkys. Later, Morgan wafers were used in an attempt to standardize with Texas Instruments. Differences were seen immediately as the diodes were processed, in that the Airtron wafers did not require the post anneal etch, as did the Morgan. After adjusting the process, both wafers showed the same basic diode parameters n and I_{sat} while the breakdown voltage was less for the Morgan, due to approximately an order of magnitude higher dopant density (Airtron ~3 x $10^{15}cm^{-3}$, Morgan ~5 x $10^{16}cm^{-3}$).

The most striking difference appeared in the DLTS spectra comparison shown in Fig. 23. The Airtron wafers contained 5 distinct peaks, indicating 5 deep traps. The Morgan wafers, however, contained only one trap, EL2. (TSCAP confirmed the results of the DLTS.) The energies and densities are given in Table 10. Clearly the Morgan wafers were the better choice for study since any change in the spectrum due to IBE would be immediately detectable as well as exclusively attributable to IBE damage, and the sensitivity would be highest, since any other peaks

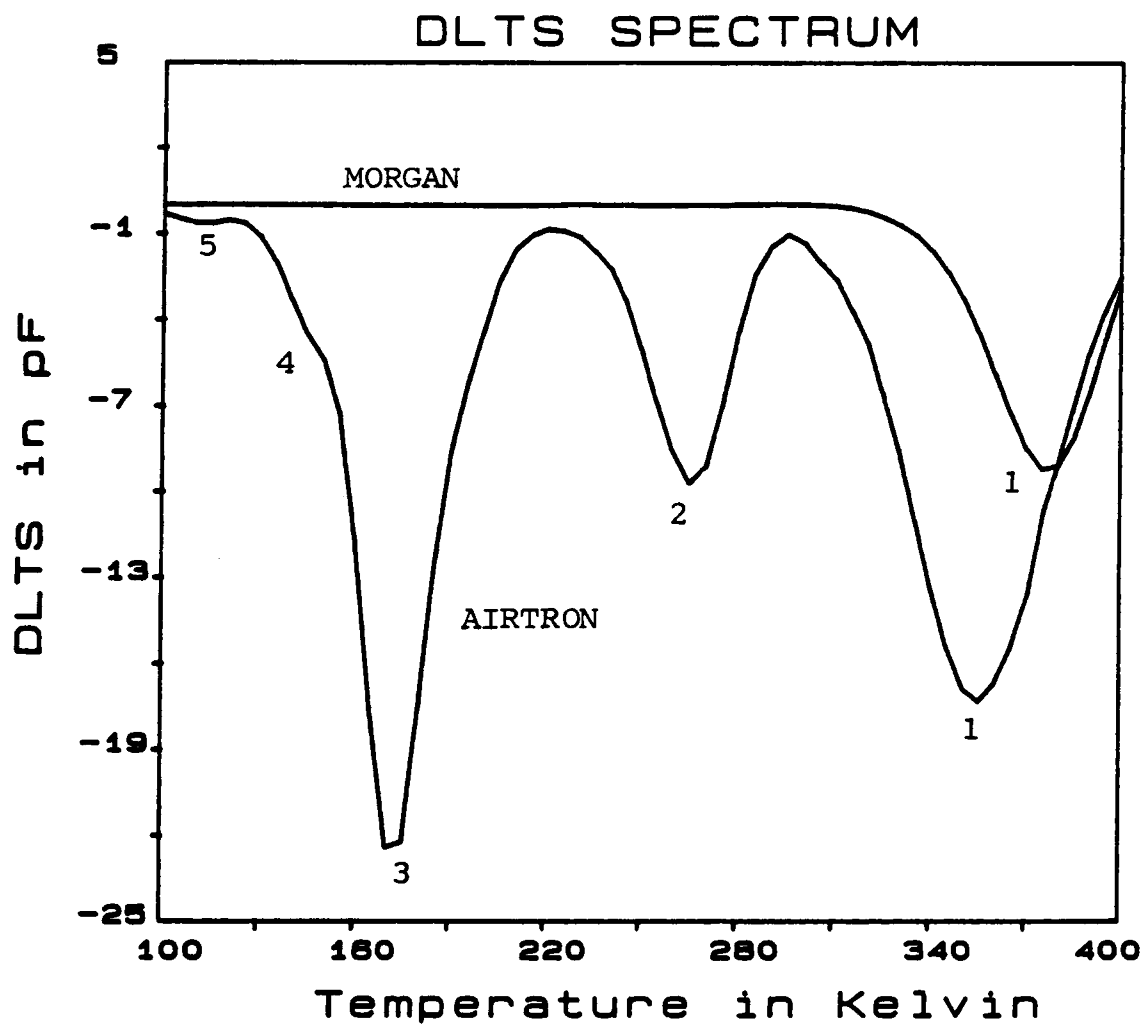


Figure 23. Comparison of DLTS spectra done on diodes made on wafers from Airtron and Morgan.

Table 10. DLTS trap parameters for Airtron and Morgan wafers.

Wafer	Trap#	Name	E_T(eV)	N_T(cm^{-3})
Morgan	1	EL2	.77	$2*10^{15}$
Airtron	1	EL2	.72	$5*10^{15}$
	2	EL3	.54	$3*10^{14}$
	3	EL6	.34	$1*10^{15}$
	4	EL8	.30	$3*10^{14}$
	5	EL14	.23	$6*10^{13}$

would not be buried in those found in the Airtron. The Airtron wafers were, however, used briefly for some IBE damage studies, and their results are given in the next section.

Both Al and Au Schottky contacts were used initially, and both yielded identical Current-Voltage characteristics and DLTS spectra. Stability was good through 400K and no apparent metal GaAs reactions were evident since I-V characteristics were linear in $\ell n(I)$ vs. V plots, even though some workers report AlAs formation and Au diffusion. Both the Au and Al gave similar results on the IBE treated samples. For these reasons and project economics, Al was the Schottky metal chosen for further studies.

Other sundry trials were performed on silver epoxies and Schottky deposition rates. Clean DLTS spectra and I-V characteristics showed that the ACME E-solder room temperature cure epoxy worked best, along with a slow initial Schottky deposition rate (~1Å/sec) for the first 100Å.

4.2 **Energy**

Al Schottky contacts were deposited on GaAs surfaces ion beam etched according to the schedule given in Table 11. The Current-Voltage characteristics are given in Fig. 24 with the accompanying diode parameters in Table 12. Several effects are immediately evident.

The diode barrier height, ϕ_B, dropped at 0.5keV ion energy, and increased slightly with IBE energy. This is also seen in the literature for plasma etching and RIE [91]. Since it is assumed that the barriers are pinned by the surface states in GaAs, this may indicate a shift in

Table 11. Energy and fluence ion etching schedule.

Fluence (cm^{-2})	Energy (keV)	Current density ($\mu A/cm^2$)	time
10^{14}	0.5	.28	57s
	3.0	1.00	16s
10^{16}	0.5	.30	89m
	1.0	1.20	22m
	2.0	14.00	115s
	3.0	50.00	32s

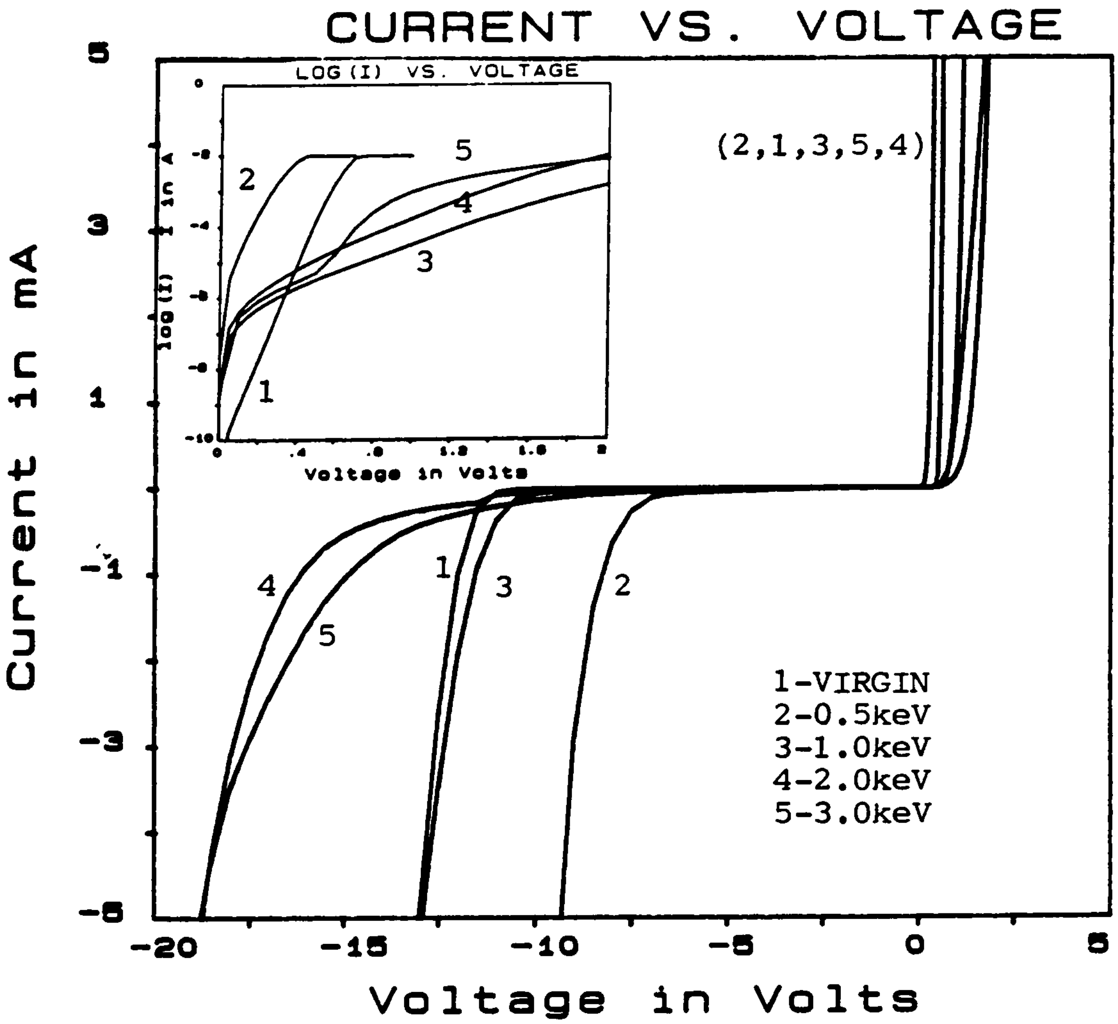


Figure 24. Current-Voltage curves for ion energy comparison. (Inset shows ℓn(I) vs. V of the forward biased diodes.)

Table 12. Diode parameters for different ion energies.

Ion Energy(keV)	Φ_b (eV)	n	I_{sat} (A)	V_B
0	.92	1.3	1.8×10^{-10}	11.7
0.5	.62	1.8	3.5×10^{-6}	8.0
1.0	.70	5.1	2.8×10^{-7}	11.1
2.0	.71	6.8	1.6×10^{-7}	15.6
3.0	.71	6.8	7.5×10^{-7}	14.3

the surface state distribution. ESCA showed the 0.5keV etched surface to be slightly As rich, while at higher energies it became Ga rich or As depleted [92]. Clearly a lack of stoichiometry near the surface is an indicator that surface and near surface defects may exist.

The presence of these near surface defects was also evidenced by capacitance dispersion with frequency for IBE treated samples, as reported by S. Sen. He reports that as IBE energy increased the value of C increased to the limit of the bridge, while frequency decreased, indicating deep, slow defects in the near surface region [93].

The reverse saturation current, I_{sat}, increased dramatically with IBE. At 0.5keV there were approximately 4 orders of magnitude increase from the virgin diodes, but as ion energy increased I_{sat} decreased slightly. This indicates that the diodes had become extremely leaky, also reported in the literature, and this may be related to surface defects which could be providing tunneling/hopping states through the barrier [94]. Surface shunting is another possibility through a conductive layer between the contacts since ESCA shows that at 0.5keV the surface oxide is not removed completely, and Nicollian reports that the oxide can be conductive [95].

Ideality factors increased with IBE energy indicating the presence of a possible insulating layer between the metal and bulk GaAs. This layer could take the form of an oxide or perhaps partly amorphous layer, which has been reported at higher ion energies using UV reflectivity [96]. For the 0.5keV diodes, there was a slight decrease in the turn-on voltage, while at ion energies greater than 1keV, forward turn-on increased until at 3keV no real turn-on could be observed. This is also indicative of an insulating layer which could be of higher resistance, as indicated in Chapter 2.

Similarly the breakdown voltage, V_B, first decreased at 0.5keV, then increased again at higher ion energies. This is contradictory to published results which show almost a total loss of V_B. In fact it was proposed that the introduction of tunneling states mentioned earlier could be used to form ohmic contacts by reducing V_B to zero [97]. The increased V_B at higher energies could be explained by the presence of a lower mobility region which increases V_B by increasing the field maximum necessary for impact ionization velocity to be attained, while at lower IBE energies (<0.5keV) near virgin mobility, coupled with increased carrier density, could decrease V_B.

Capacitance-voltage measurements showed good linearity for $1/C^2$ vs. V plots, but the ϕ_B values were unusually high. The dopant profiles remained flat through the measurable thickness. An example is given in Fig. 25. The average dopant density was approximately $4 \times 10^{16} cm^{-3}$, which is within 10% of the manufacturer's data sheet.

At higher energies the room temperature capacitance decreased slightly at both 0V and -4V bias, but was not always consistent. As temperature was swept the capacitance showed a typical increase, with only the virgin sample showing the definite presence of a trap step, i.e. TSCAP showed no trap structure for all IBE diodes.

The conductance vs. temperature is plotted in Fig. 26. The virgin piece is relatively flat through 400K while the 0.5keV sample showed a sharp increase after 300K. As energy increased past 1keV the conductance again dropped. The 0.5keV increased so much that it appeared as a

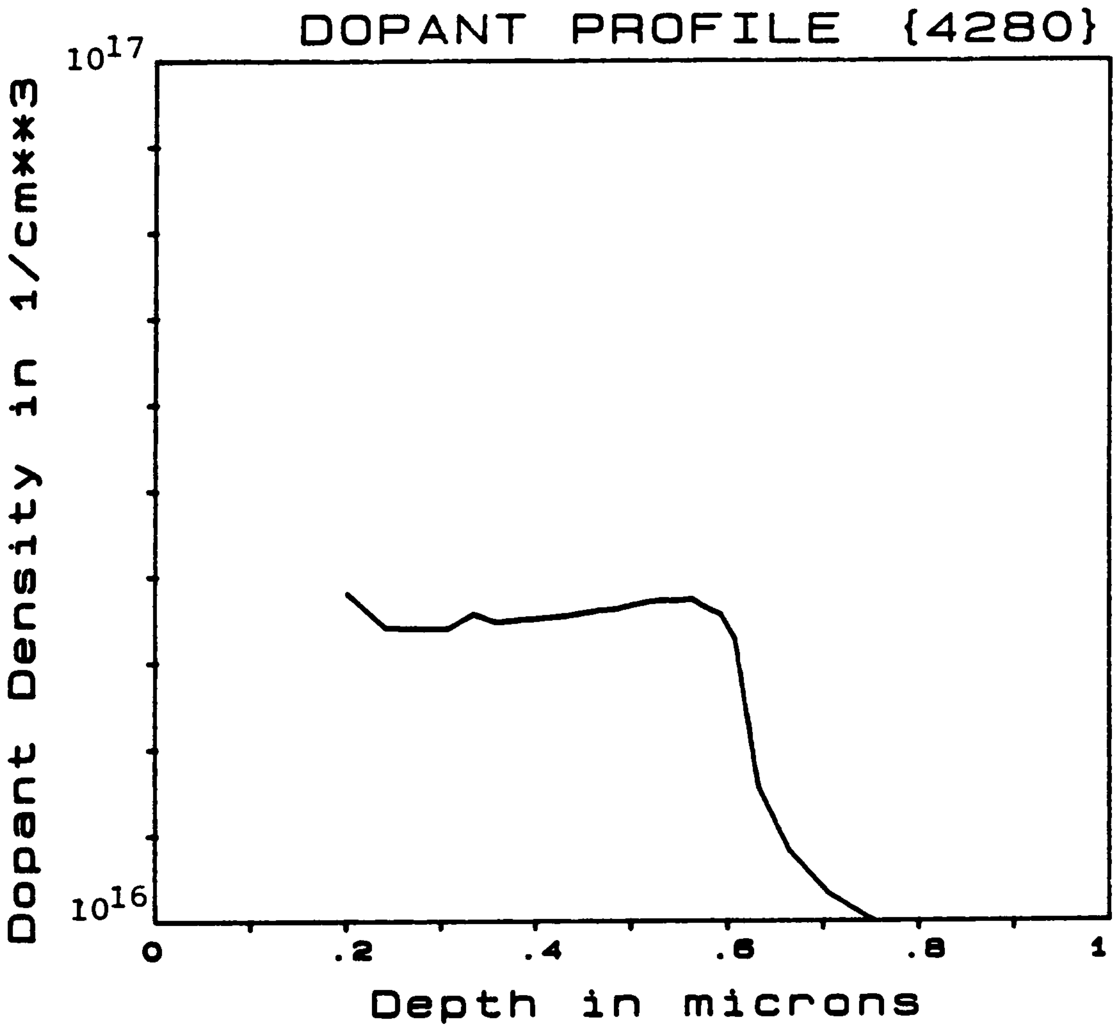


Figure 25. Typical C-V dopant profile for IBE samples.

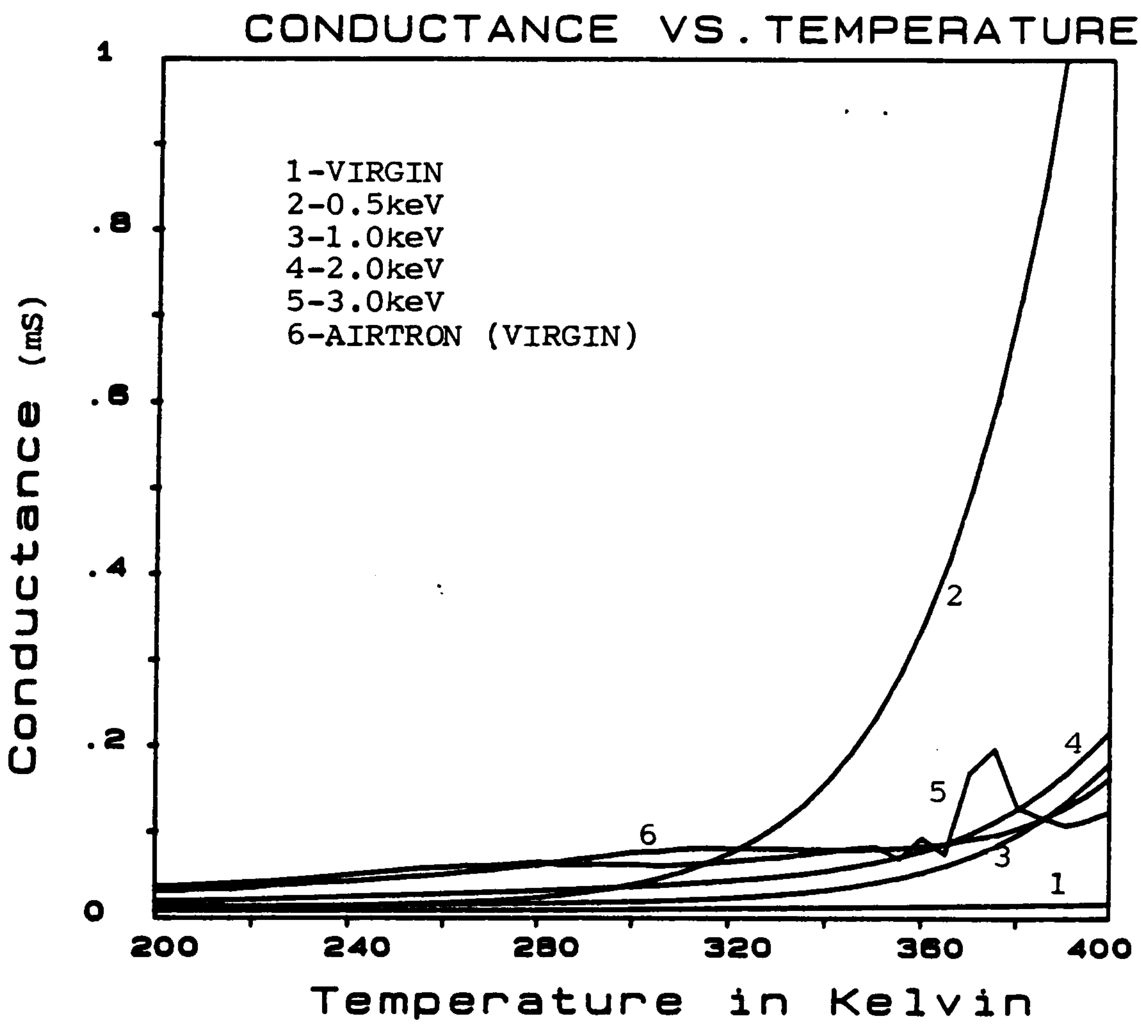


Figure 26. Conductance-Temperature comparison of different IBE energies with the Airtron sample included for further comparison.

short to the HP 4280A, thus killing the G and C measurement. This is possibly attributable to the surface states acting as generation-recombination centers, increasing carrier concentration with temperature and creating a shunt path. At higher energies the mobilities may be sufficiently decreased to negate the effects of the increased carrier concentration to some extent.

The most interesting results were seen in the DLTS measurements shown in Fig. 27. (All spectra had a tendency of being somewhat noisy but not unreadable.) The virgin sample showed an EL2 peak at E_T = .77eV and N_T = 4 x $10^{15}cm^{-3}$. At 0.5keV the EL2 peak was completely removed and a trap was seen at .53eV and 1.5 x $10^{14}cm^{-3}$ density. For IBE at 1keV, EL2 was still missing and traps were seen at .55eV and .53eV with densities of 2.7 x $10^{14}cm^{-3}$ and 1.7 x $10^{14}cm^{-3}$. At 2keV, EL2 was seen slightly (~2.9 x 10^{14}) with other traps given by .52eV, .46eV, .59eV and .54eV, with densities averaging 5 x 10^{14}. Finally at 3keV, there was again some EL2 present but still small compared with other traps at .58eV and .49eV and densities of 8 x $10^{14}cm^{-3}$. This loss of EL2 was seen in the Airtron samples as well, and is shown in Fig. 28. Thus the loss of EL2 was not unique to the Morgan wafers, and can be considered a general result of IBE. This does not indicate that EL2 is physically depleted, loss of As_{Ga}, since the the 0.5keV samples showed As richness with no EL2. However, the DLTS and TSCAP loss of EL2 is a real effect. If it is assumed that the diode capacitance obeys a parallel plate capacitor model, the depths of DLTS study were approximately: 1790Å-4140Å for the virgin, 1530Å-4010Å for 0.5keV, 1540Å-4040Å for 2keV, and 3500Å-5410Å for 3keV.

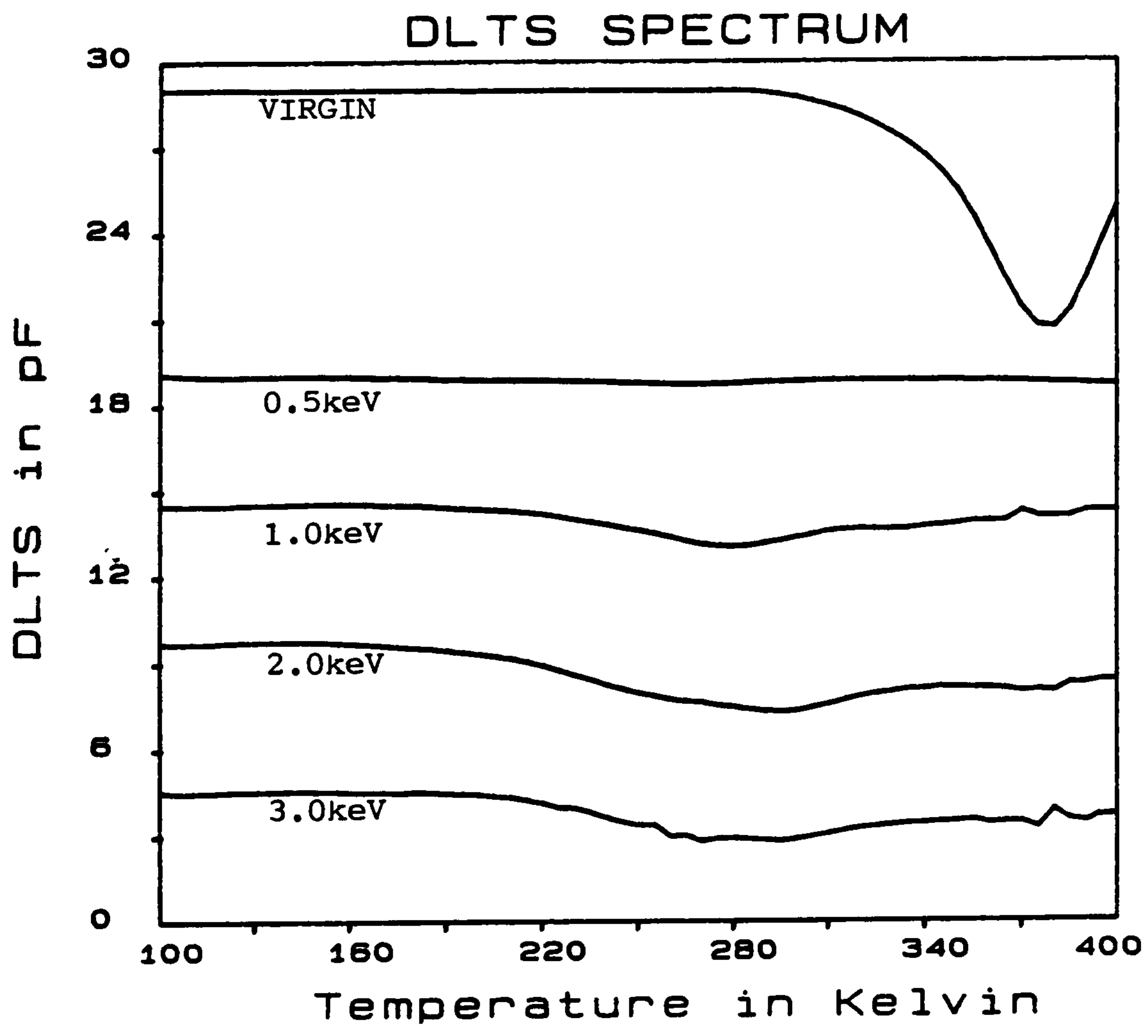


Figure 27. DLTS spectra comparing different IBE energies.

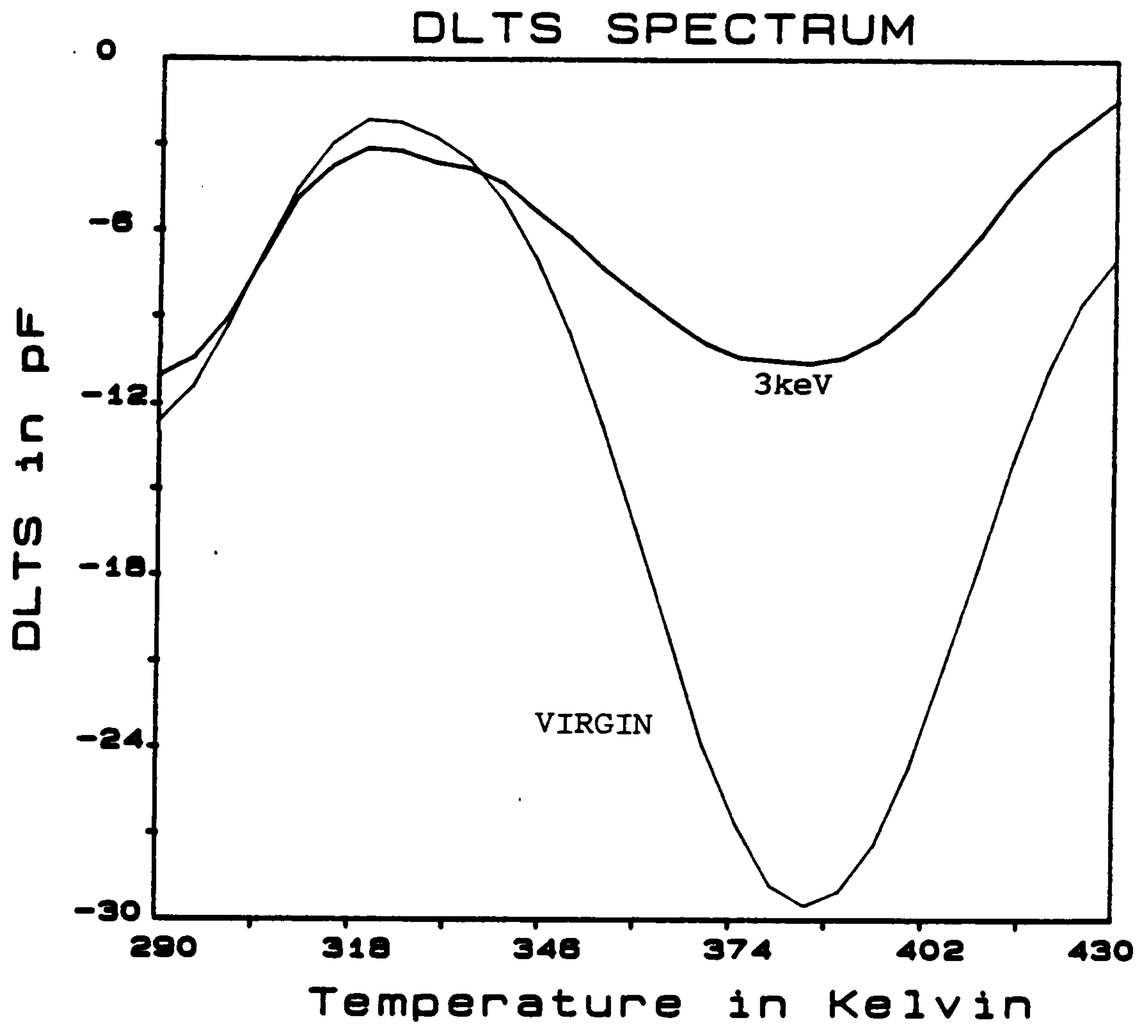


Figure 28. DLTS spectra showing loss of EL2 in 3keV IBE energy Airtron sample as compared to virgin material.

From these depths it might be interpreted that the damage and EL2 loss were reaching as deep as 1500Å into the sample, even at 0.5keV. This, however, is highly unlikely when reference is made to the ion implant tables (Table 5) and was explored further in the chemical etch study (discussed later).

Another factor which could be affecting the DLTS is the increased diode conductivity with temperature. It has already been seen that the HP 4280A is effectively shorted for the 0.5keV bombarded sample, and the capacitance measurement drops accordingly. This question will be further pursued in a later section.

The following is a summary of the important results of the energy study:

1. Diode parameters degraded rapidly with IBE. The 0.5keV GaAs diode appeared leakiest, with decreased turn-on and breakdown voltages. The 3keV samples had slightly lower (less than 0.5keV) leakage (I_{sat}) with increased turn-on and breakdown voltages. Ideality factors increased and saturated above 2keV while barrier heights, ϕ_B, decreased, in agreement with reports.
2. Dopant profiles were unaffected.
3. Zero and reversed bias capacitance decreased slightly with IBE.
4. Diode conductance showed a strong temperature increase for 0.5keV diodes, while the effect was less strong for >1keV IBE GaAs diodes.
5. TSCAP and DLTS indicated a total loss of EL2 for 0.5keV and 1keV samples, while some EL2 signal (almost unmeasurable) was seen in 2keV and 3keV diodes. Other electron traps were introduced into the spectra, but the densities were small compared to virgin EL2

6. Depth analysis, assuming a parallel plate capacitor model, indicated that IBE was affecting the GaAs down to at least 1500Å, even for the 0.5keV IBE GaAs samples.

Possible reasons for the above results have been discussed and further analysis in the thesis (below) will help clarify them.

4.3 Fluence

To study the effects of ion fluence on IBE damage, samples at the low (0.5keV) and high (3keV) energies were subjected to two different fluences, 10^{14}ions•cm^{2} and 10^{16}ions•cm^{-2} before Al Schottky deposition. Diode parameters for this IBE schedule are given in Table 13.

At 0.5keV there was a clear threshold where damage had occurred. A fluence of 10^{14}ions•cm^{-2} appeared to do no real damage to current-voltage characteristics in that ϕ_B, I_{sat}, n, V_B, V_T showed no degradation. In fact, the I-V's given in Fig. 29 appear to be sharpened or improved over the virgin case. However, after 10^{16}ions•cm^{-2}, as reported earlier, I-V damage was evident in the extreme increase in I_{sat} and the decrease in both V_B and V_T. Clearly, the fluence is an important factor in IBE at this energy for I-V characteristics.

Conductance versus temperature characteristics are given in Fig. 30. Again at 10^{14}ion•cm^{-2} fluence there was no perceptable change, indicating, together with the I-V parameters, that the supposed surface state changes had not as yet occurred. Thus there was no sharp temperature dependence in the conductivity.

Table 13. Diode parameters for different fluences.

Ion Energy	Fluence (ions • cm^{-2})	ϕ_B (eV)	n	I_{Sat} (A)	V_B (V)
virgin		92	1.3	1.8×10^{-10}	11.7
.5 keV	10^{14}	1.05	1.2	2.5×10^{-12}	11.3
	10^{16}	.62	1.8	3.5×10^{-6}	8.0
3 keV	10^{14}	.67	1.6	1.5×10^{-8}	12.0
	10^{16}	.70	5.4	7.5×10^{-7}	14.3

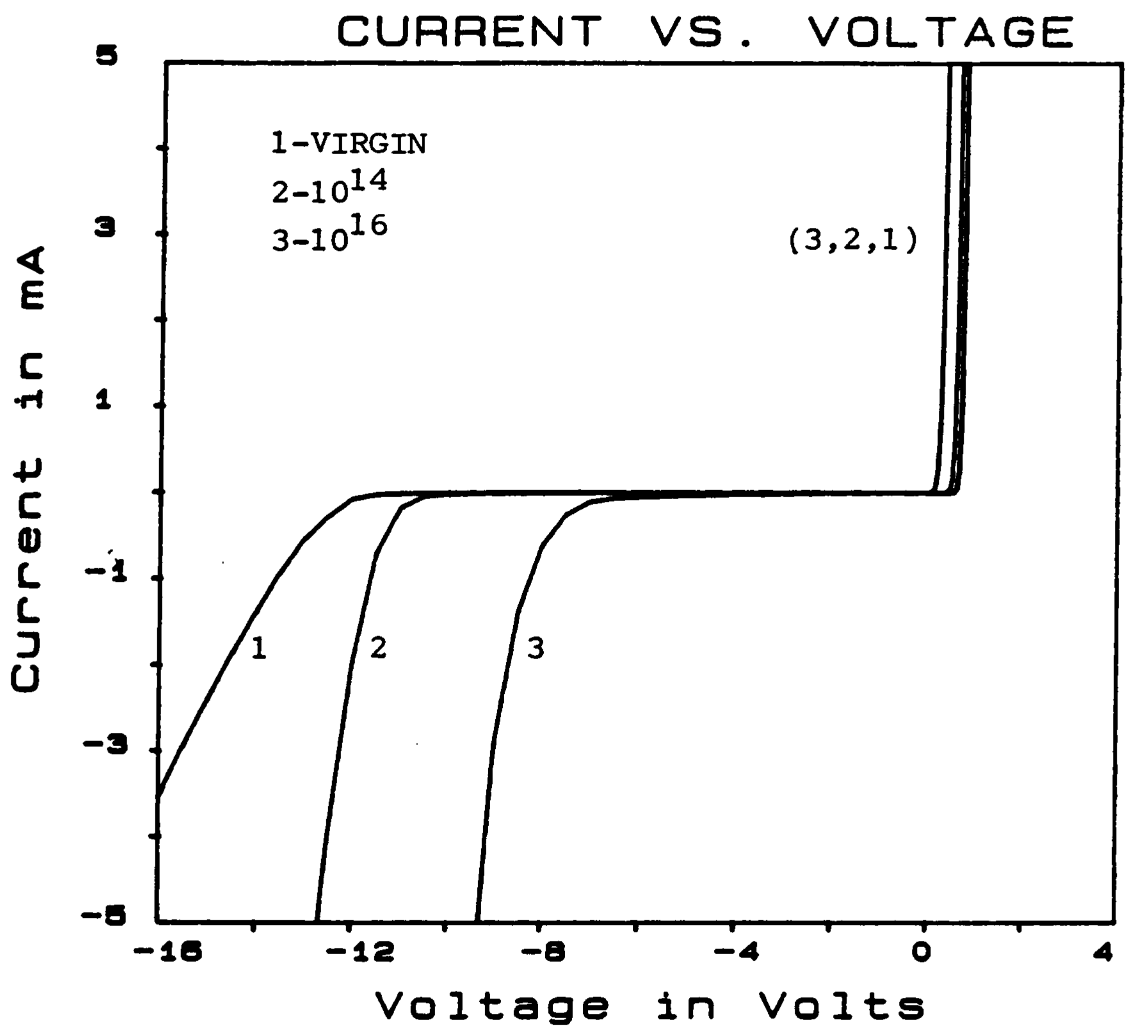


Figure 29. Current-Voltage characteristics for 0.5keV energy with variation of fluence.

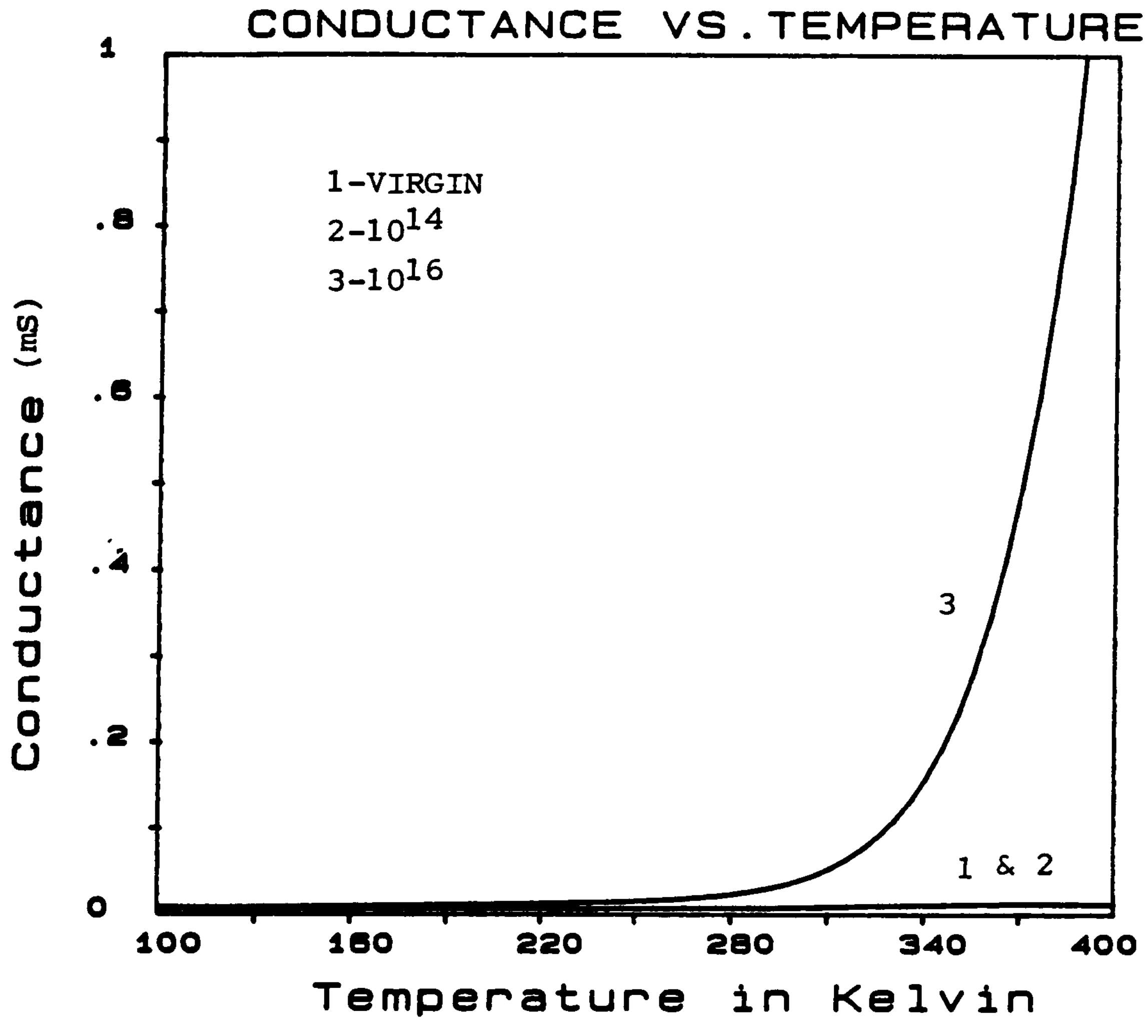


Figure 30. Conductance-Temperature characteristics for 0.5keV energy with variation of fluence.

The above is borne out in the DLTS spectra shown in Fig. 31. Note that at 10^{14}ions•cm^{-2} there was no change in the EL2 peak. Therefore it appears that at low fluence and low energy there is no perceptable damage.

This, however, was not the case for the 3keV samples; refer to Table 12 and Fig. 32. The diode parameters indicate clear damage at 10^{14}ions•cm^{-2}, and a considerably decreased ϕ_B. In fact, the barrier was slightly lower than for the 10^{16}ions•cm^{-2} samples. I_{sat} showed a 3-order of magnitude increase at 10^{14} over virgin samples, and continued to increase another order of magnitude at 10^{16}, while n increased only slightly at first until it saturated at 10^{16}. V_B and V_T remained relatively unaffected at 10^{14} ions•cm^{-2}. Conductivities increased with fluence and dopant profiles remained flat.

The DLTS spectra are given in Fig. 33. The virgin spectrum is used as a gauge since EL2 is plainly visible. At 10^{14} fluence there did appear to be an EL2 peak with E_T = .79eV and N_T = 6.2 x 10^{14}cm^{-3}, with the 10^{16}ion•cm^{-2} fluence having a very similar spectrum. Thus it appears that all detectable DLTS damage had occurred by 10^{14}cm^{-2} fluence at 3keV ion energy.

The results of the fluence study thus indicate that:

1. Low fluence (10^{14}) and low energy (.5keV) had no perceptable effect on diodes. However, the effective surface cleaning performed by such a beam was questionable since ESCA indicated the presence of an oxide after IBE. Thus, at low energies, fluence is an important parameter for damage effects.

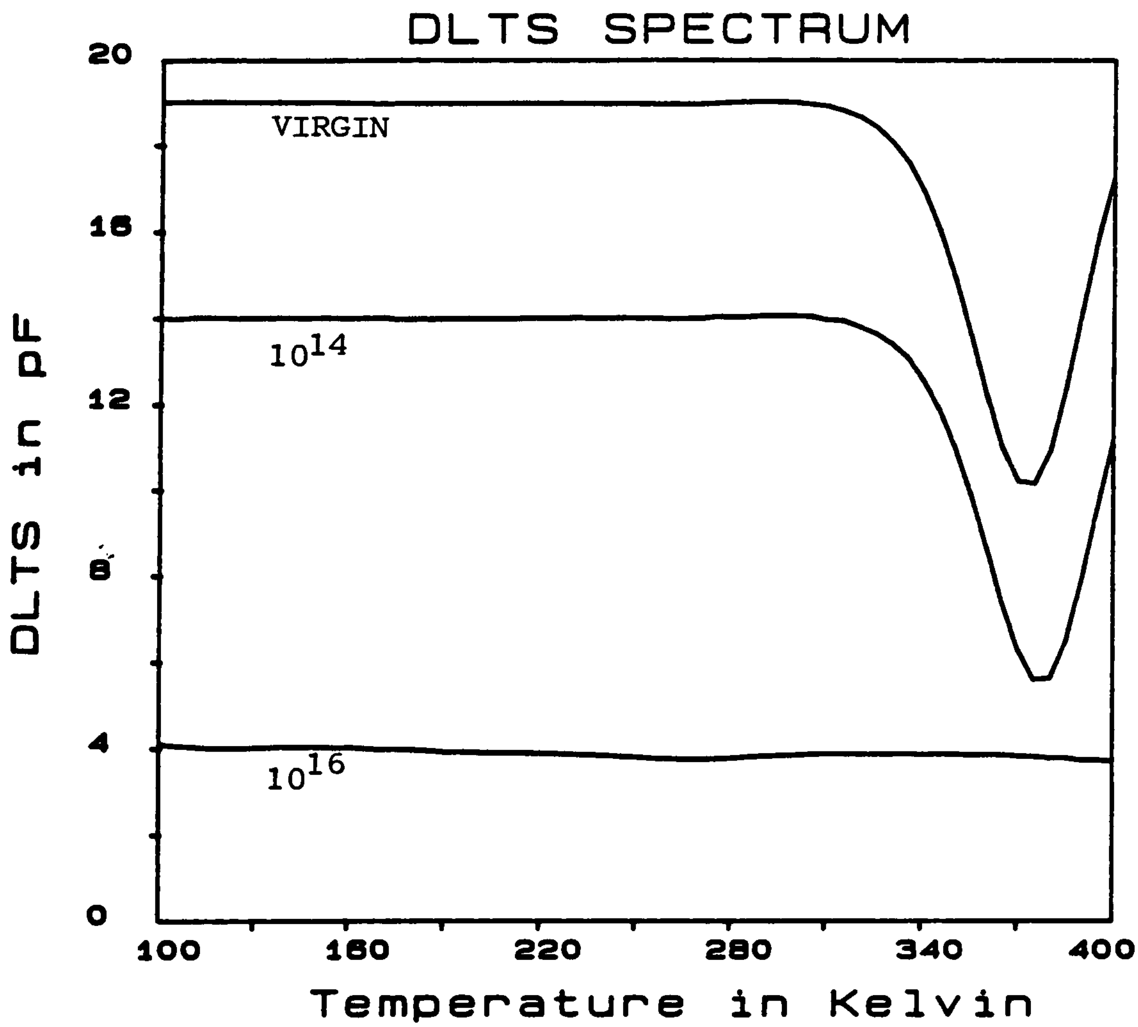


Figure 31. DLTS spectra for 0.5keV energy, at different ion fluences.

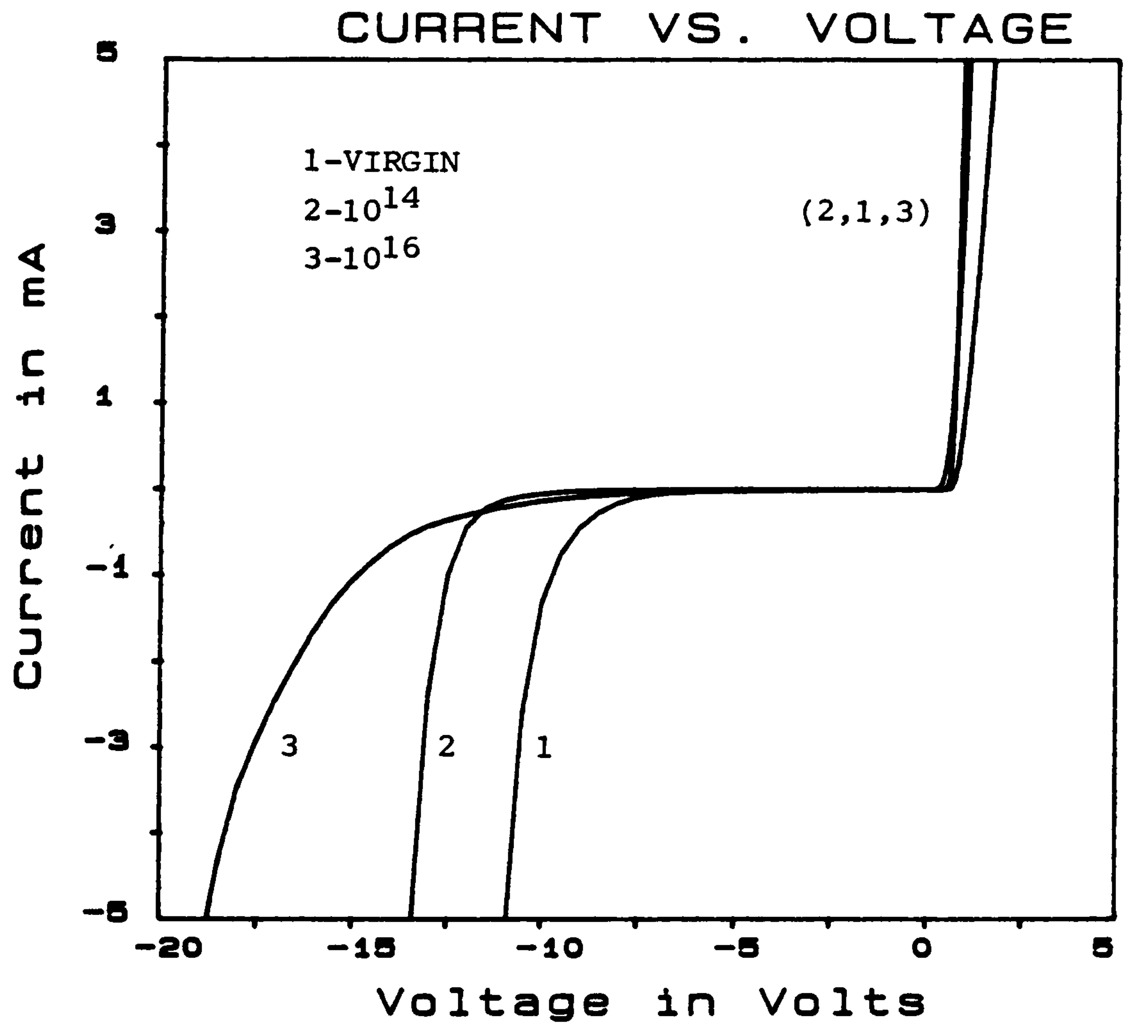


Figure 32. Current-Voltage characteristics for 3keV energy with variation of fluence.

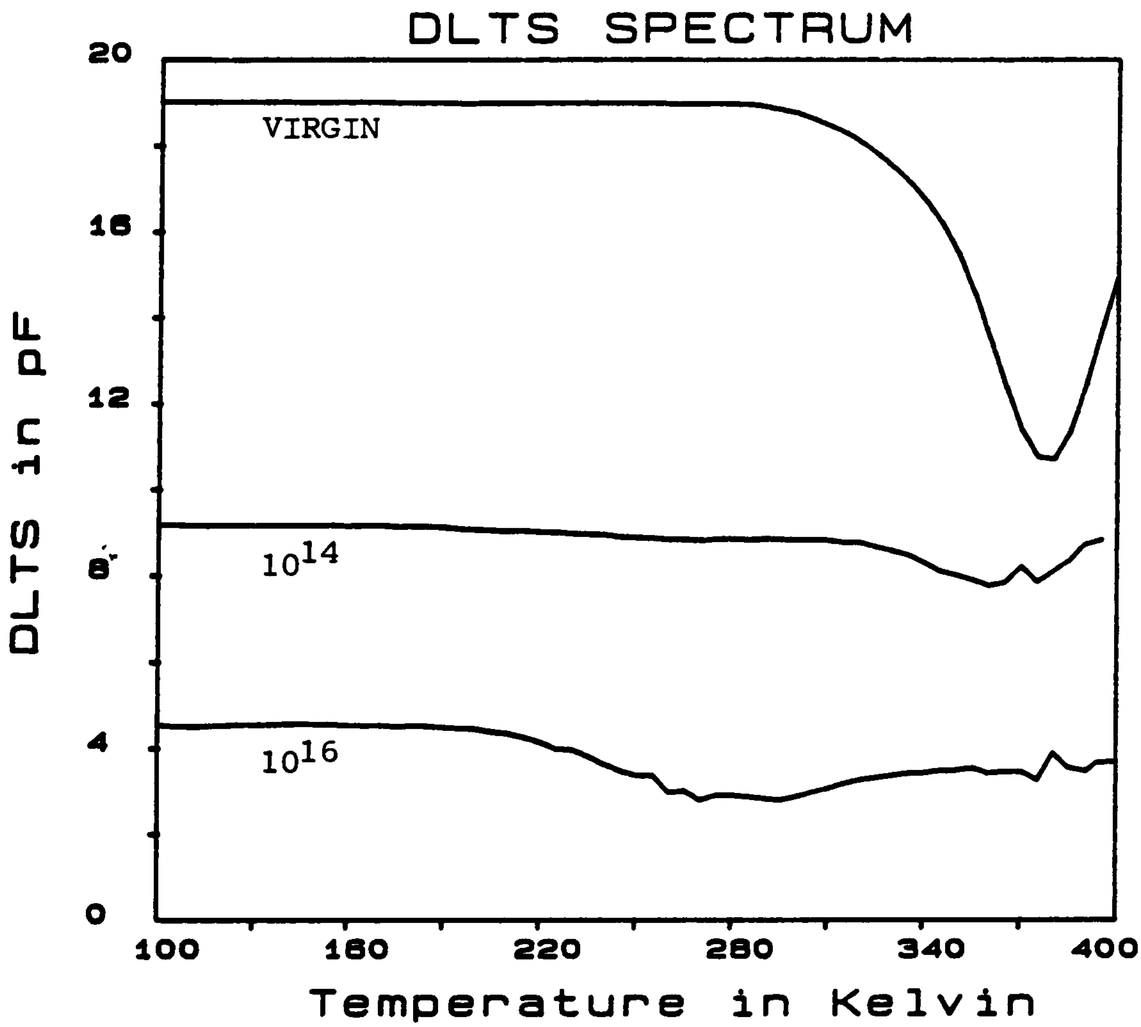


Figure 33. DLTS for spectra for 3keV energy, at different ion fluences.

2. Higher energy (3keV) bombarded GaAs materials were much less sensitive to IBE fluence, since damage was similar at 10^{14} and 10^{16} fluences.

4.4 Chemical Etch

Previous results have shown the effective depth of the damage induced by IBE to be uncertain since it appeared from DLTS that EL2, a bulk trap, could be completely removed from the crystal with a low energy (0.5keV) surface etch. This appears quite unlikely since damage would have to extend past 1500Å at 0.5keV for this to occur, the ion-implant tables indicating penetration to no more than 45Å at 1keV [98]. Since, however, the IBE surface treatment is affecting the spectra, an electrical depth profiling DLTS was not advised. Instead, a chemical etch removal of the surface was chosen to perform depth profiling of the IBE damage.

To etch the surfaces a weak H_2SO_4:H_2O_2:H_2O 1:1:100 (1:1 H_2SO_4, 30% H_2O_2) solution was used which has been found to etch crystalline GaAs at 530Å/min, and amorphous GaAs at 660Å/min [99]. Since the nature of the damaged layer was somewhat questionable, it was assumed that the damage in general consisted of a 50% crystalline and 50% amorphous composition, which was dependent on energy. Therefore, the 1:1:100 solution was assumed to etch the IBE surfaces at approximately 600Å/min.

The etch solution was checked by suspending a piece of S.I. GaAs over the etchant so that half of the sample extended into the solution. Etching was allowed to proceed for three hours, after which time the sample profile was taken using a Sigmascan stylus profilometer.

Etching was found to occur at approxiamtely 500Å/min, in good agreement with reported results. Thus, the assumed 600Å/min rate represented a maximum depth or worst case scenario.

Samples were prepared using 0.5keV and 3keV IBE energies, with 10^{16} ions•cm^{-2} fluence to maximize damage. The surfaces were chemically removed according to the schedule in Table 14, after which Al Schottky contacts were deposited, giving a damage depth analysis for the IBE surfaces.

Table 15 indicates the diode parameters for the 0.5keV I-V characteristics given in Fig. 34. As indicated in the parameters and the curves, the characteristics appeared to recover somewhat after 100Å removal, and appeared fully recovered after 500Å. This places the damage depth relating to for the I-V's at about 100Å. Recovery was also seen in the conductance-temperature curve given in Fig. 35. The shunting effect also disappeared after 100Å of removal.

Surprisingly, the DLTS spectra also recovered after only 100Å removal as can be seen by the 60% recovered EL2 peak in Fig. 36. DLTS recovery was complete by 500Å . Therefore it appears that at this low energy the DLTS measurement, which is typically seen as a bulk measurement technique, was being affected by a near surface damaged region. This effect has not been reported in the literature, and appears to be an important one.

The diode parameters and I-V's for the 3keV diodes are given in Table 16 and Fig. 37. The diode and the conductance-temperature characteristics did not appear to recover until removal of 1000Å of damaged GaAs. This is far deeper than implant theory predicts (the

Table 14. Chemical etch schedule used in depth profile.

Depth in Å	Etch time (600Å/min)*
100	10s
500	50s
1000	1m 40s
5000	8m 20s

*All solutions were stirred during etch

Table 15. Diode parameters for 0.5keV depth profile.

Depth (Å)	ϕ_B (eV)	n	I_{Sat} (A)	V_B (V)
0	.62	1.80	3.5×10^{-6}	8.0
100	.80	2.60	1.2×10^{-6}	11.5
500	.94	1.40	9.0×10^{-12}	11.2
1000	1.05	1.10	6.0×10^{-14}	11.3
5000	.99	1.05	1.0×10^{-12}	11.3

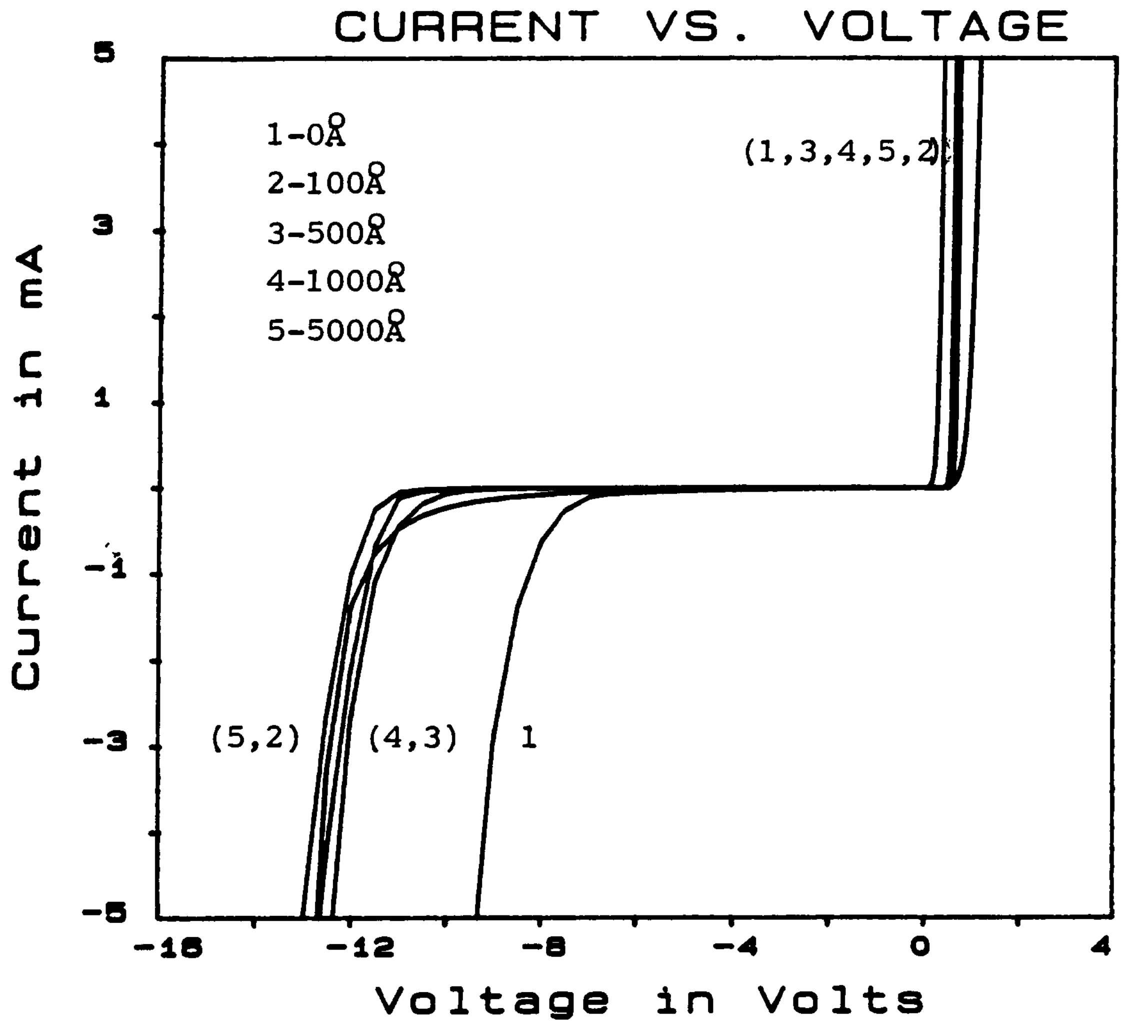


Figure 34. Current-Voltage characteristics for 0.5keV energy depth profile.

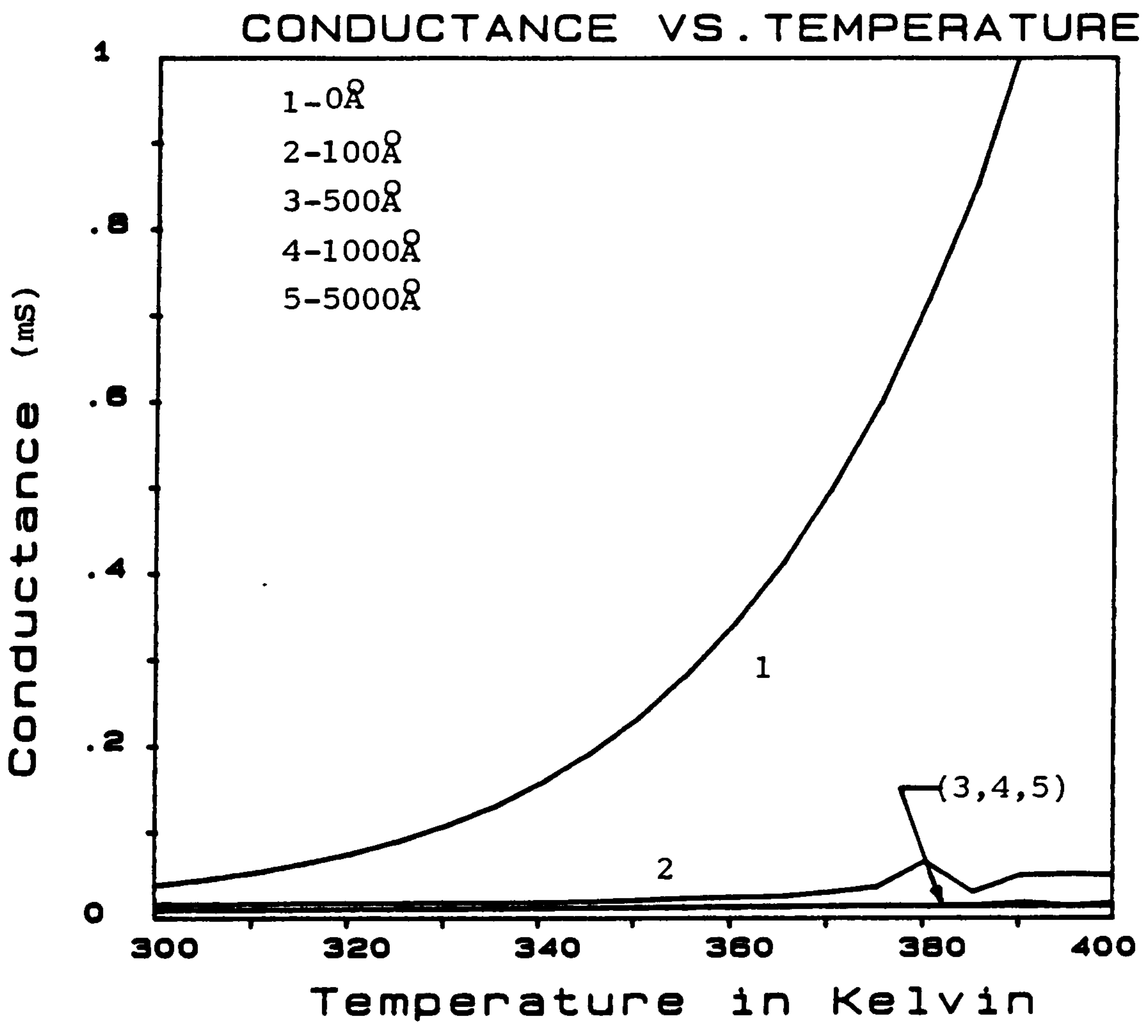


Figure 35. Conductance-Temperature characteristics for 0.5keV energy depth profile.

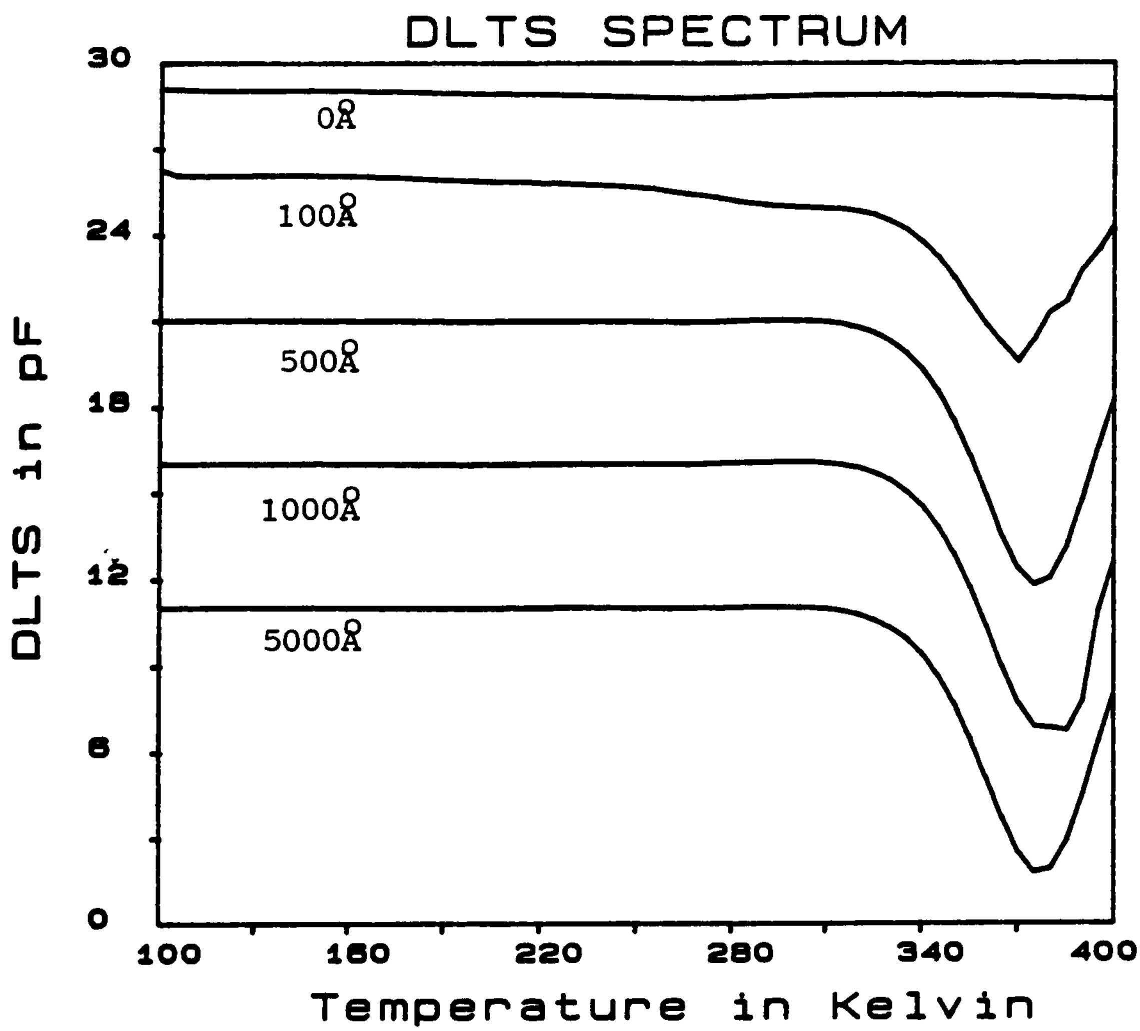


Figure 36. DLTS spectra for 0.5keV energy depth profile.

Table 16. Diode parameters for 3keV depth profile.

Depth (Å)	ϕ_B (eV)	n	I_{sat} (A)	V_B (V)
0	.70	5.4	7.5×10^{-7}	14.3
100	.71	5.0	1.7×10^{-7}	15.5
500	.78	5.4	1.5×10^{-8}	13.5
1000	.84	2.7	2.1×10^{-9}	12.5
5000	.89	2.6	2.9×10^{-10}	10.5

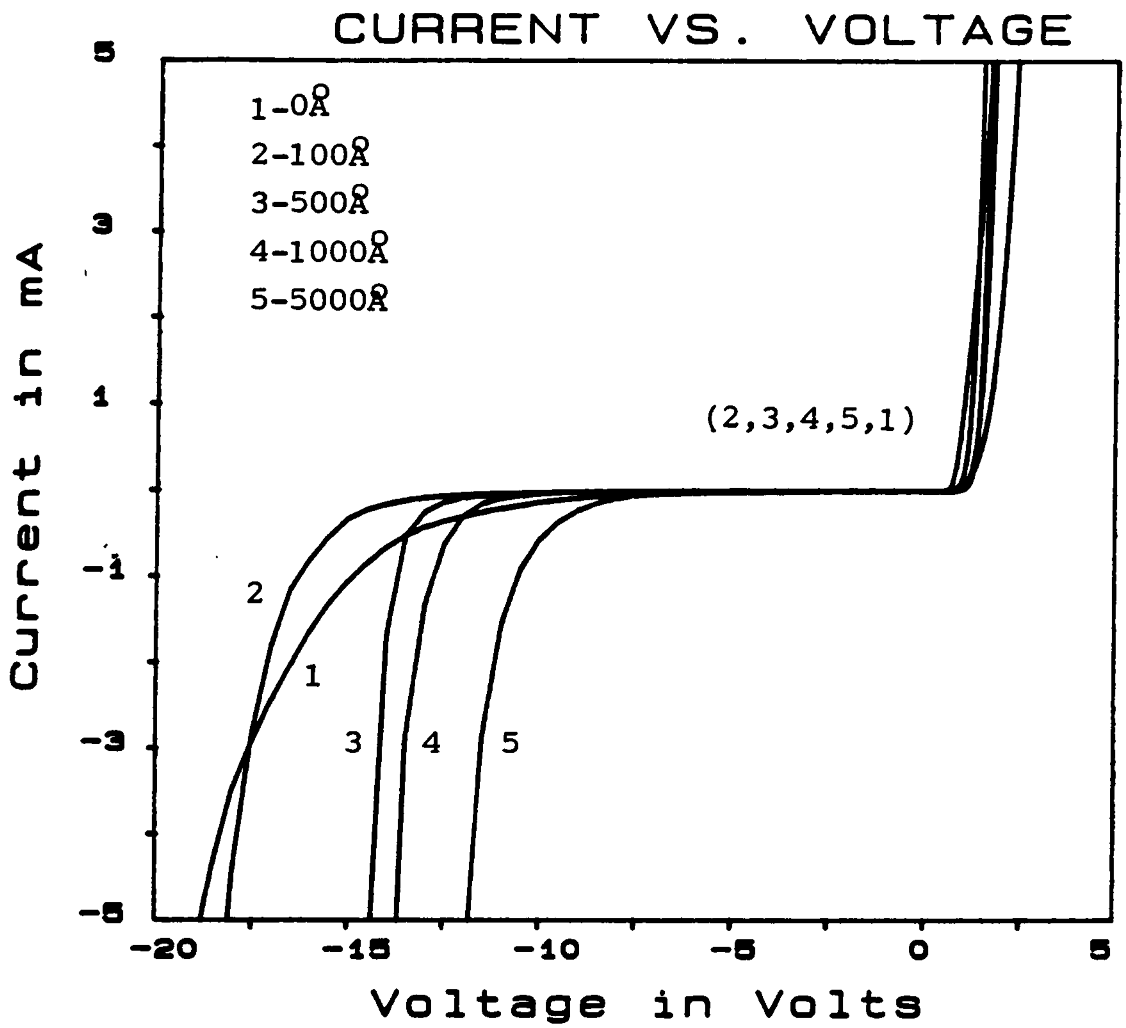


Figure 37. Current-Voltage characteristics of 3keV energy depth profile.

tables give ≈ 80Å). This depth is not completely unexpected, however, since several workers have seen detectable damage down to 2000Å. [100]. Pang, in a similar study of RIE on VPE GaAs showed wet etch recovery of I-V's done at 1keV after removal of 250Å [101]. All papers tend to agree that a partly amorphous layer ≈100Å thick is generated during ion treatment while defects can diffuse into the bulk at room temperature.

Similarly, after removal of 1000Å by sequential wet etch showed recovery in the DLTS spectra, as shown in Fig. 38. It must be noted that no studies showed diffused bulk damage at this energy any deeper than 2000Å, which would be necessary to generate a true EL2 loss. Thus it appears that the EL2 recovery is tightly coupled to the DLTS measurement and its interaction with the damaged surface. Another interesting result of the DLTS measurement is the appearance, after 500Å removal, of two prominant electron trap peaks at .58eV, 3.8 x $10^{15}cm^{-3}$ and .51eV, 2.7 x $10^{15}cm^{-3}$. These may be attributable possibly to buried or shallow depth traps introduced by IBE, and previously hidden from the DLTS due to the damaged surface.

Further evidence that EL2 was not truly being removed at any energy is given by the electrical depth of study assuming a parallel plate model, where it was seen that the actual depth studied did not actually change while the EL2 peak recovered. Clearly this is an indicator of a desensitized DLTS measurement which has been distorted due to surface damage.

The major results of this section are:

1. At IBE of 0.5keV, 10^{16}ions-cm^{-2}, diode parameters, conductivity and EL2 recovered after 100Å of surface removal. Damage layer thickness is approximately 100Å.

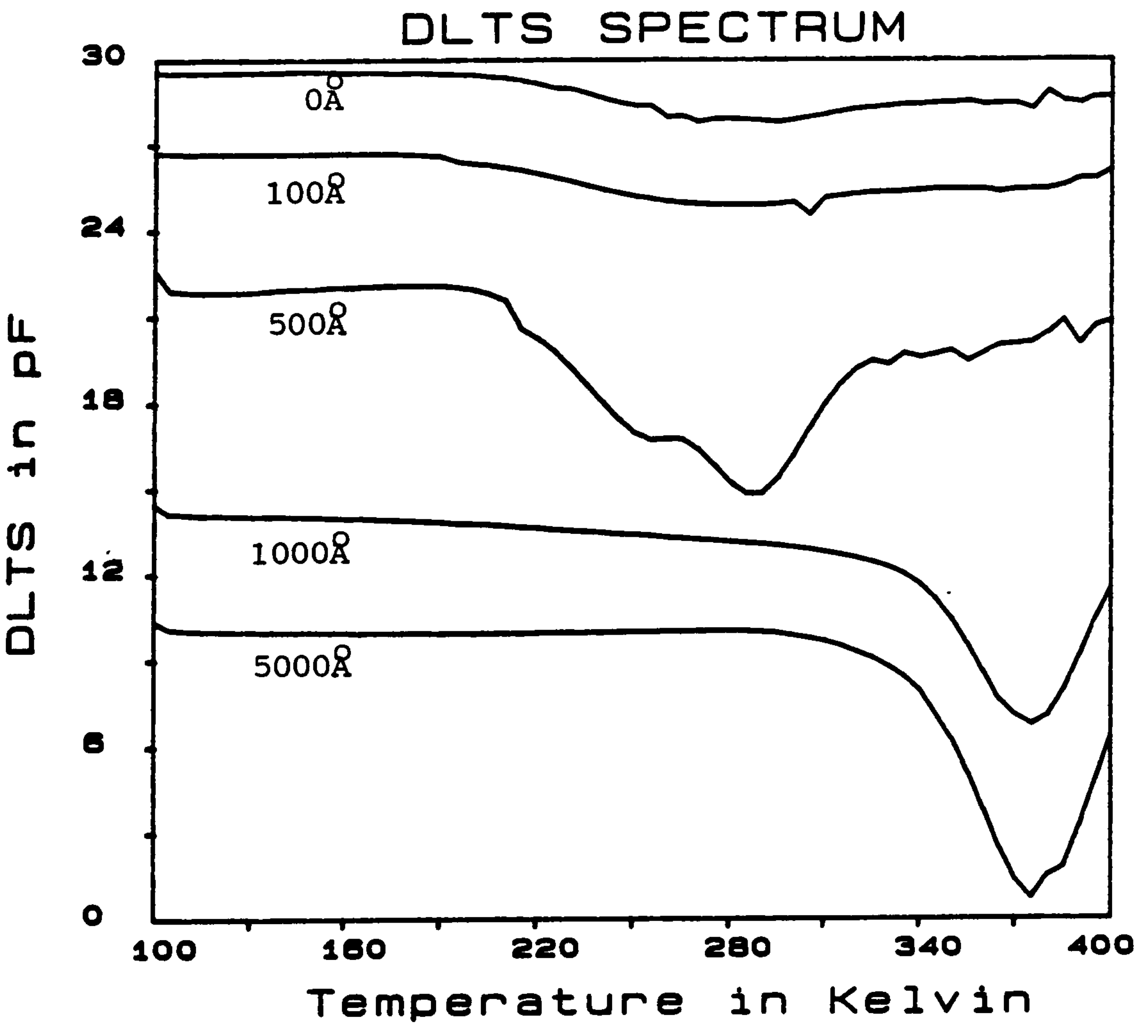


Figure 38. DLTS spectra for 3keV energy depth profile.

2. At 3keV IBE, diode parameters, conductivity and EL2 recovered after 1000Å, of surface removal. Damage layer thickness is thus estimated at 1000Å for 3keV IBE material.
3. Depth of analysis shows that the loss of EL2 is most probably not real, but rather a result of a desensitized DLTS measurement due to the top damaged layer.
4. The same damage responsible for diode degradation is desensitizing DLTS measurements.

At this point, however, the nature of the damaged layers appears somewhat different at 0.5keV than 3keV in that the 0.5keV IBE seems to result in a thin somewhat conductive layer, while the 3keV causes a thicker and much less conductive layer. The electrical role of the bulk and surface damage may be somewhat different also. In order to further delineate their effects and clarify the damaging results of IBE two other tests were performed. In one test the surface layer between the contacts was chemically removed. In a second test, a variable resistor was placed in shunt with a virgin diode. The procedures and results are given below.

4.4.1 **Surface Removal**

In order to further delineate the effects of bulk or under contact damage from the surface or between contact damage both 0.5keV and 3keV (10^{16}ions$\cdot$cm^{-2}) diodes were chemically etched with the 1:1:100 etchant so that 2000Å were removed from the surface between the contacts. The damage beneath the contacts remained the same, so that only the effect of damage under the Schottky survived.

No real improvement occurred in the I-V characteristics for the 0.5keV diodes, while the conductivity-temperature characteristic improved toward the virgin material, as shown in Fig. 39. This indicates that the surface shunt path may have been removed and the remaining conductivity was a result of damage beneath the contact. More surprisingly, however, was the DLTS spectra given in Fig. 40, in which it can be seen that EL2 was 25% recovered. This clearly indicates that the loss of EL2 is a result of some desensitized DLTS measurement in that the true spectrum is being mased by a surface effect.

At 3keV, 2000Å surface removal caused no improvement in the diode parameters and only slight improvement in the conductance temperature (G-T) characteristic. The DLTS spectrum was identical to that of unetched 3keV sample. It appears that the 3keV ion etch damage shows itself not through the surface between the contacts but, exclusively by damage under the contacts.

The results of these tests are:

1. For 0.5keV samples with 2000Å removal showed improvement in G-T and DLTS measurements. Therefore, damage at low energies occurs in both the surface and bulk. Further, the reduction in shunting conductance lends itself to the notion that there exists a low resistance layer.
2. For 3keV samples, removal of 2000Å around the contacts showed no perceptable recovery. This indicates that a higher resistance region may exist under the Schottky metal.
3. The nature of the damaged layer (low resistance, high resistance) is somewhat different for low and high energy IBE.

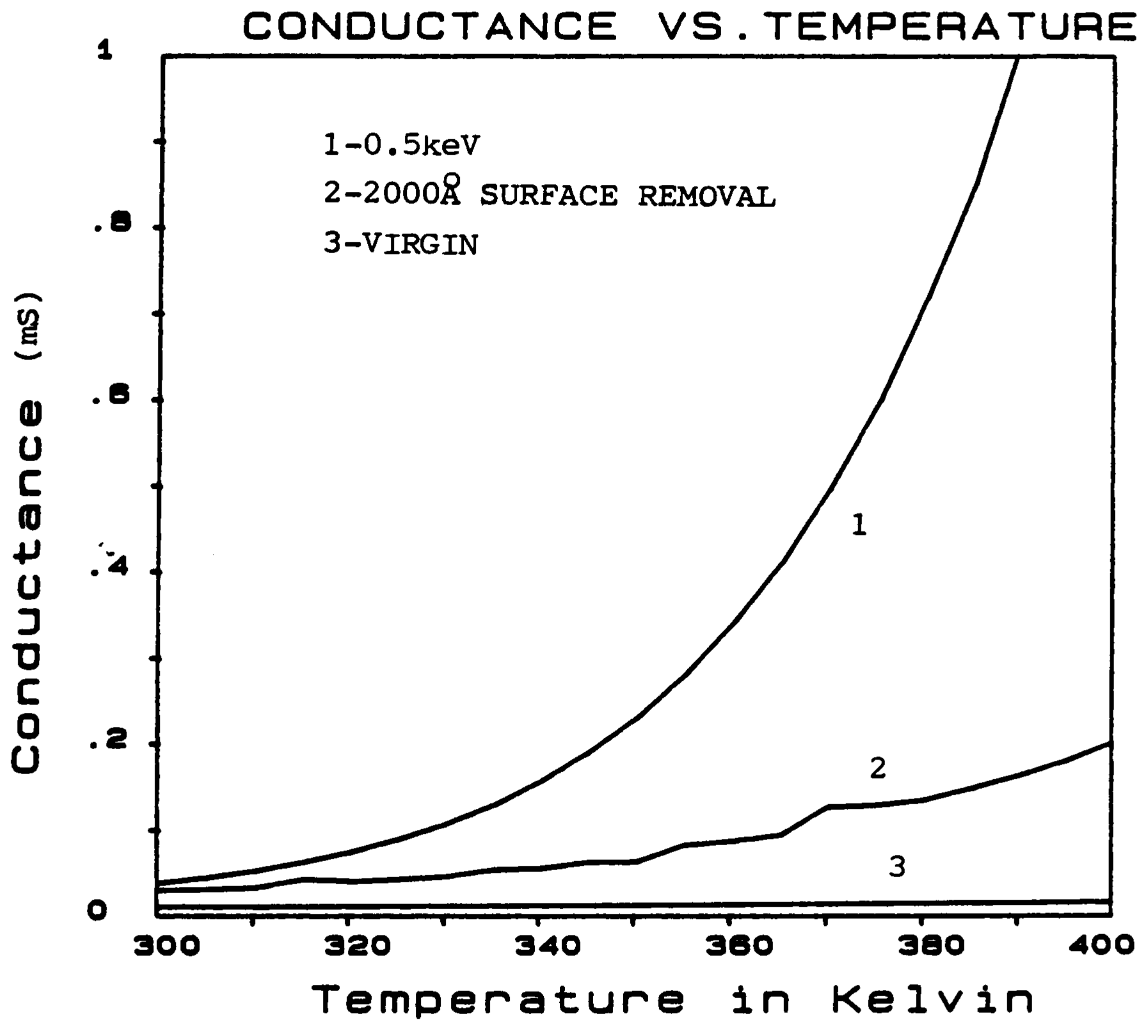


Figure 39. Conductance-Temperature virgin, 0.5keV and 2000Å surface removal.

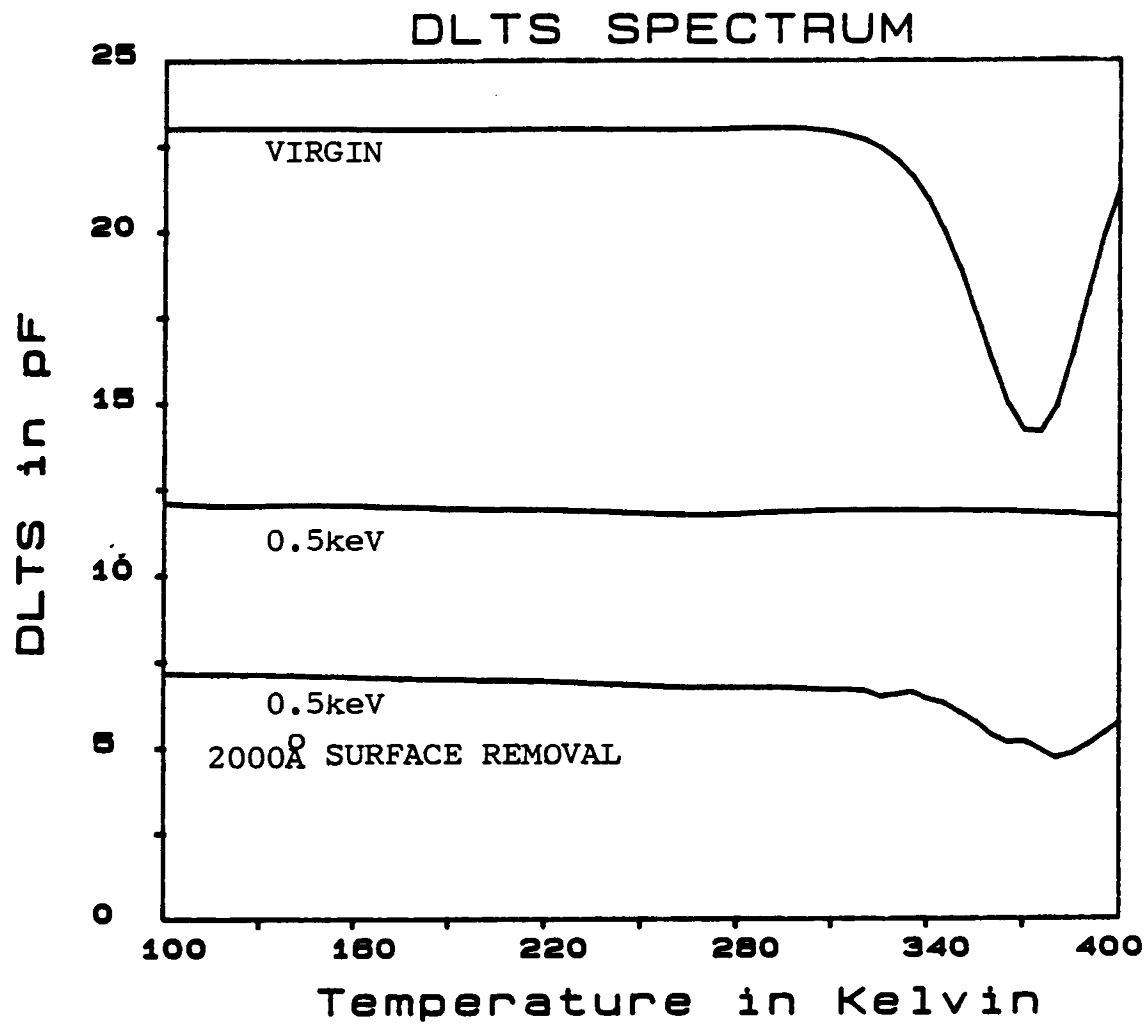


Figure 40. DLTS spectra comparing virgin, 0.5keV and 2000Å surface removal.

4.4.2 Shunt Resistor

As further proof of the effects of a shunt conductance, a virgin diode was placed in the cryostat with an external shunting potentiometer connected as in Fig. 41. As the resistance was reduced from approximately 9kΩ to 240Ω, the EL2 peak reduced as shown in Fig. 42, and the measured conductance increased from .1mS to 4.2mS, which is similar to values for the actual 0.5keV sample.

Therefore, the 0.5keV model of a low resistance layer shunting the diode appears correct, since the reduction of EL2 was nearly the same as the recovery of EL2 in the 0.5keV sample in which 2000Å of GaAs between the contacts had been removed. This will be seen further in section 4.6.

4.5 Anneal

Having partially understood the nature of the damage as being confined to a layer containing traps, with a partly amorphous crystal structure and variable depth according to IBE energy, an attempt was made to repair the damage. To do this, 0.5keV and 3keV (10^{16}ions$\cdot$cm^{-2}) samples were thermally annealed in forming gas at approixmately 350°C for 15min. before Schottky metal deposition. This temperature was chosen since it was believed that it may be high enough to repair damage without being hot enough to change the ohmic contact characteristics [102]. This was not an extensive study, but merely a brief attempt based upon previous knowledge of IBE damage found in this study to improve diode characteristics through a low temperature anneal.

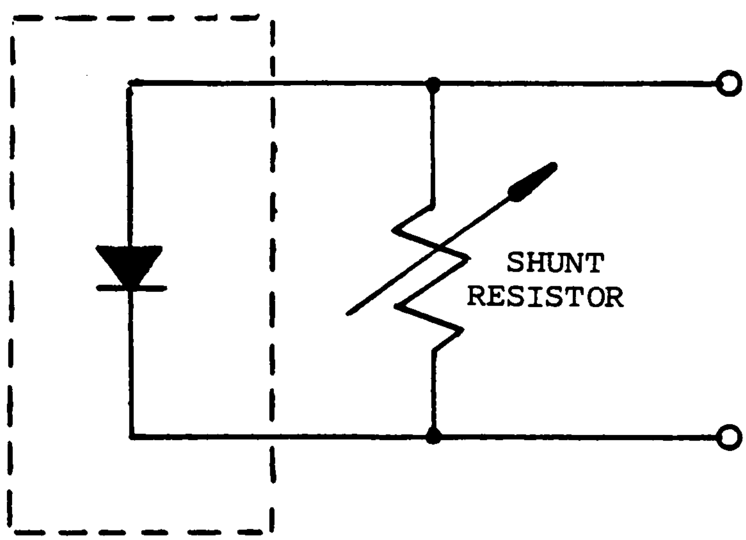


Figure 41. Experimental setup for shunt resistor test.

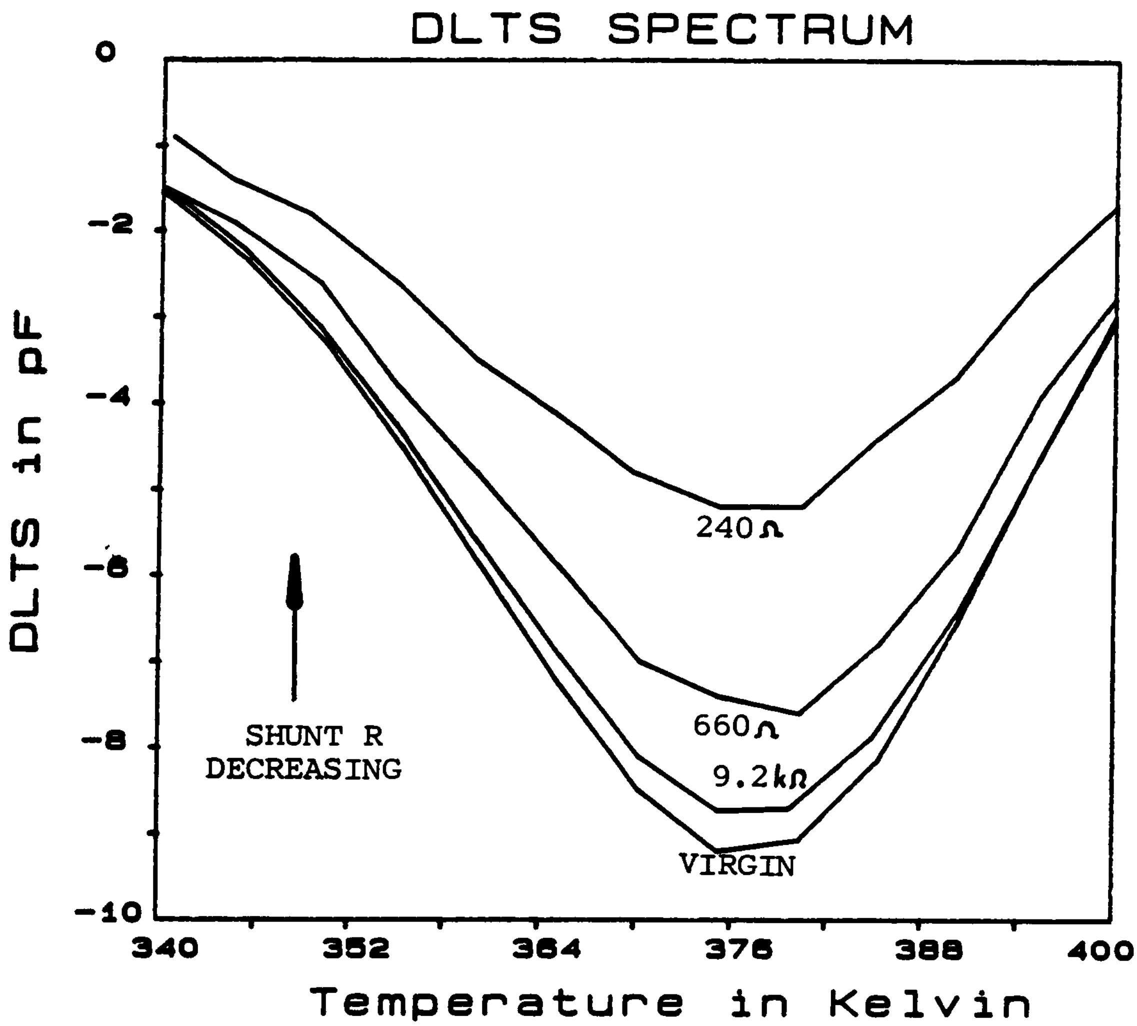


Figure 42. DLTS spectra for shunt resistor test.

For the 0.5keV samples, no improvement was seen in the DLTS spectra while the G-T curve did flatten somewhat toward a virgin characteristic. This was accompanied by an order of magnitude decrease in I_{sat}, with an increase of V_B to 11.0 V. Thus it appears that the brief anneal did slightly improve the I-V characteristics and conductivity, which could indicate a slight change in the defect density of the damaged layer toward the virgin state.

At 3keV there was again no improvement in the DLTS spectrum, while the conductivity vs. temperature characteristic took the shape of the conductivity vs. temperature characteristic of the 0.5keV IBE sample. This could indicate that the mobility of the damaged layer was somewhat improved, while defect densities were maintained, which would tend to resemble the 0.5keV case. This was supported in the I-V's in that V_B and n dropped while I_{sat} increased.

Chapter 5: **Modelling and Computer Analysis**

From experimental results based on the electrical study taken up here, along with ESCA, UV reflectivity and capacitance-frequency dispersion, physical and electrical models were developed, and are given below.

5.1 **Physical Model**

Shown in Fig. 43 is a simple physical model based on the literature and previous results. The thicknesses given are variable, depending on the IBE parameters and total surface treatment. The first layer (D1) consists of an oxide of approximately 20Å in thickness and of variable composition, since ESCA (done by June Epp) showed that 0.5keV samples never completely lost their pre-etch oxide, while 3keV samples showed several different As and Ga oxides present after atmospheric exposure. The second layer (D2) is considered to be a partly amorphous layer of thickness less than 100Å to 1000Å (depending on beam energy), as seen by UV reflectivity (done by Jeff Feng) and the chemical etch profile. The degree of amorphicity is dependent upon energy since it seems that the lower energy IBE still retains most of its crystallinity, while higher energies indicate a marked loss in crystal structure. The third layer (D3) is somewhat ambiguous in thickness since it represents room temperature diffused damage which has been indicated in the literature. In this study, the damage appears to vary in depth from 100Å to 1000Å, depending on Ar^{+} energy. This is a somewhat generalized and simplified model, though electrical and computer analyses will be shown to help validate it.

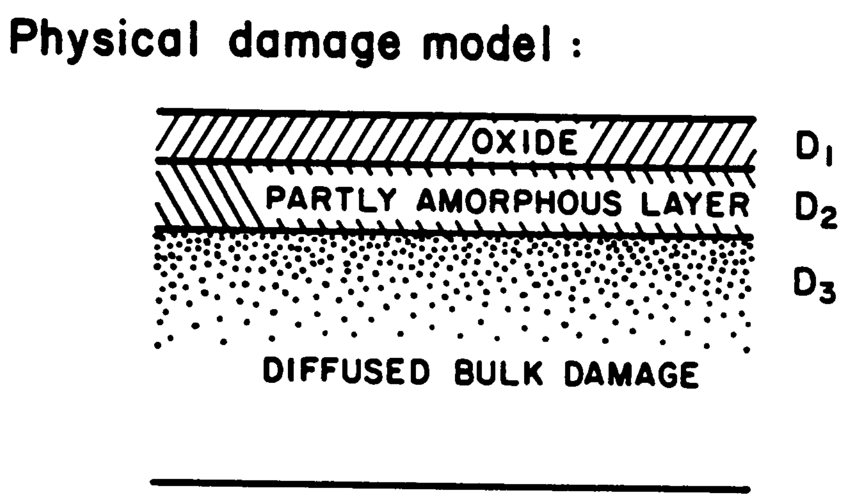


Figure 43. Physical model of IBE damaged GaAs.

5.2 Electrical Model

A reverse bias electrical model was developed from the above physical structure by assignment of lumped R-C components to corresponding damaged layers. The result is given in Fig. 44, with the components listed. This model was used in computer analysis to simulate DLTS and capacitance-frequency measurements. As will be seen, the simulations are in good agreement with experimental results.

5.2.1 DLTS Simulation

In order to simulate the DLTS measurement so that results corresponding to experiment could be directly compared the simulation took measurements in the same fashion as in the actual experiment. Any capacitance bridge analyzes a network's capacitance from a measurement of its impedance at a given frequency (fixed at 1 MHz for HP 4280A). The network capacitance is defined as

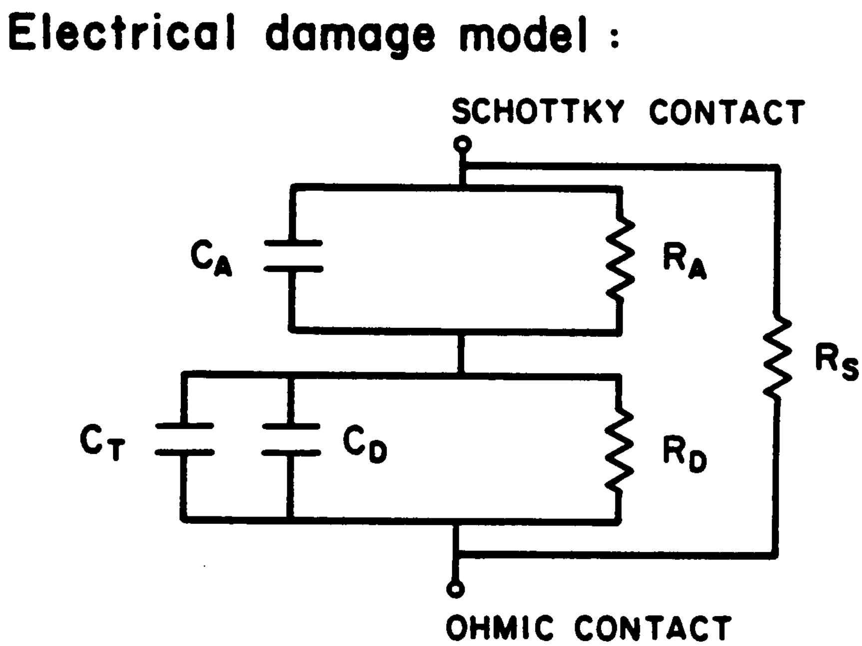
$$C = -\frac{1}{\omega X} \tag{35}$$

where X is the imaginary part of the impedance $Z = R + jX$, [103]. The impedance of the electrical model is

$$Z_M = \frac{(R_S EG + R_S F^2)}{G^2 + F^2} + j\,\frac{(R_S FE - R_S FG)}{G^2 + F^2} \tag{36}$$

where

$$E = \frac{R_A}{1 + \omega^2 C_A^2 R_A^2} + \frac{R_D}{1 + \omega^2 C^2 R_D^2} \tag{37}$$



R_A, C_A - amorphized series layer

R_S - surface shunt layer

R_D, C_D - ideal diode

C_T - trap capacitor(EL2)

Figure 44. Electrical model corresponding to the physical model.

and

$$F = \frac{C_A R_A^2}{1 + \omega^2 C_A^2 R_A^2} + \frac{C R_D^2}{1 + \omega^2 C^2 R_D^2} \tag{38}$$

with $G = E + R_S$ and $C = C_D - C_T$, where the negative sign has been added to indicate the electron trap nature of EL2. The measured capacitance from the above impedance is given as

$$C_M = \frac{1}{\omega F} + \frac{2E}{\omega F R_S} + \frac{E^2}{\omega F R^2} + \frac{F}{\omega R_S^2} \tag{39}$$

Clearly as $R_S \rightarrow \infty$ and $R_A \rightarrow 0$, $C_M \rightarrow C_D$, (ideal diode) with C_T as a transient capacitance. Since C_T represents EL2, it is to be treated as a transient capacitance whose value will be given as

$$C_T = C_o \exp(-r_T t) \tag{40}$$

where C_o is the capacitance change amplitude and r_T is the trap decay rate given by

$$r_T = \nu \exp(-E_T/kT) \tag{41}$$

with $\nu = 10^{12} s^{-1}$ (the thermal release frequency) and $E_T = .77eV$ (the EL2 trap energy).

These equations were programmed as detailed in Appendix B, and DLTS spectra simulated.

Using the above equations, the EL2 peak magnitude was plotted as a set of curves (Fig. 45) developed from variation of R_A and R_S with all other components held constant (C_A = 2pF, C_D = 150pF, R_D = 10MΩ). C_A was held constant since it was believed that the resistivity of these layers played a more important role, as seen in the resistor shunt experiment. Two regions of EL2 loss can be seen, even though no true loss has occurred. These regions may be directly related to the IBE results (loss of EL2) and confirm that the DLTS was being desensitized by the surface damage layer.

The first loss region relates the 0.5keV samples to a low resistance shunt layer. Figure 46 indicates the progression of decreasing shunt and the corresponding EL2 magnitude. It is also interesting to note that these results correspond directly to both the shunt resistor and surface removal experimental results. The second region of loss (higher resistances) can be related to the 3keV samples, in which it is believed that a higher resistance, lower mobility layer was generated by the high energy IBE. This progression can be seen in Fig. 47.

The strong resemblance of the simulated DLTS spectra to experiment suggests the validity of both the physical and corresponding electrical models.

5.2.2 Capacitance-frequency Simulation

As mentioned earlier S. Sen reported capacitance-frequency dispersion for 3keV IBE diodes. The same electrical model was used to simulate his results and are given in Fig. 48 assuming high resistance layers corresponding to the virgin and 3 keV cases (network values corresponded to those used in the 3keV DLTS simulations that reduced

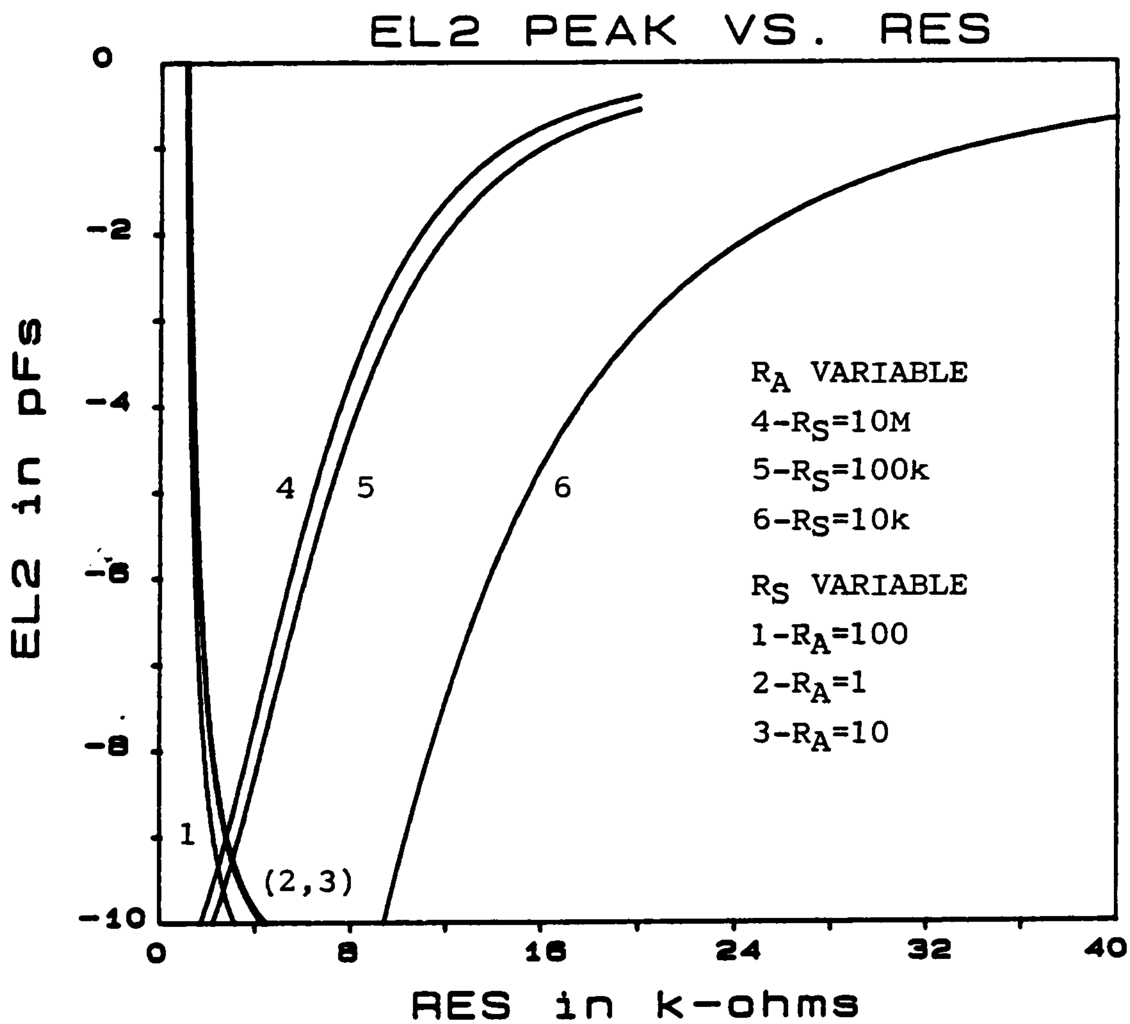


Figure 45. Curves showing dependency of EL2 peak magnitude on R_L and R_S variation.

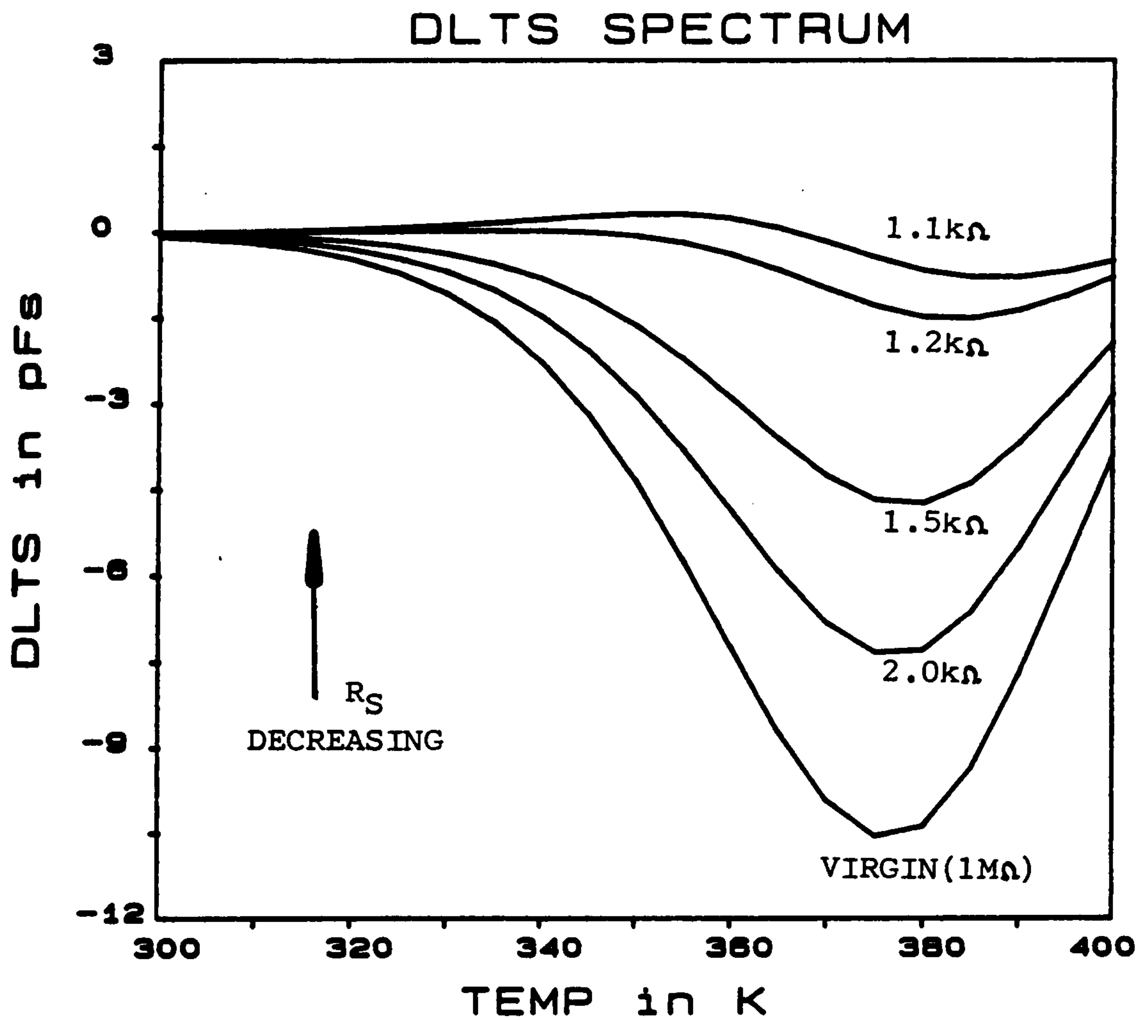


Figure 46. Simulated DLTS for 0.5keV layer model.

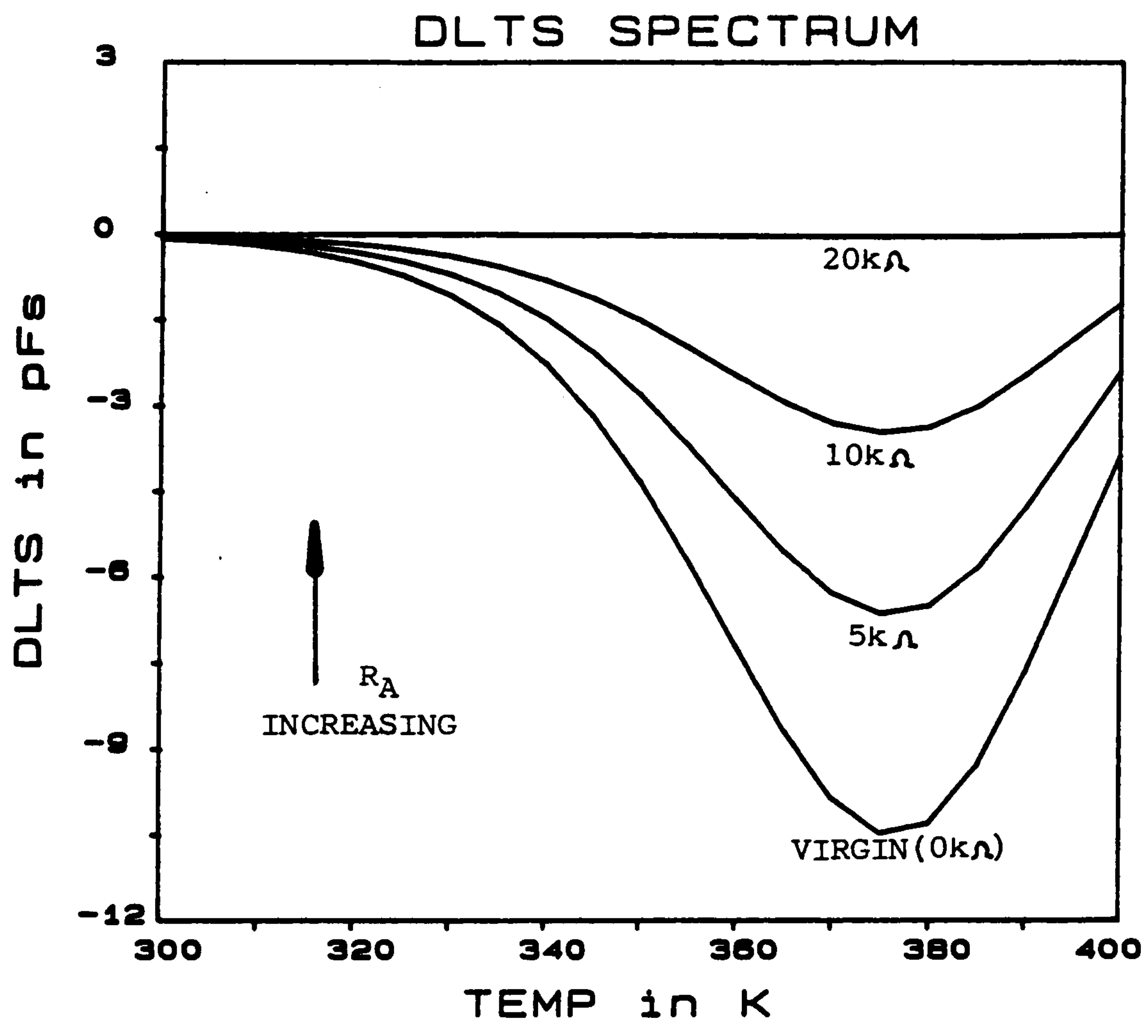


Figure 47. Simulated DLTS for 3keV layer model.

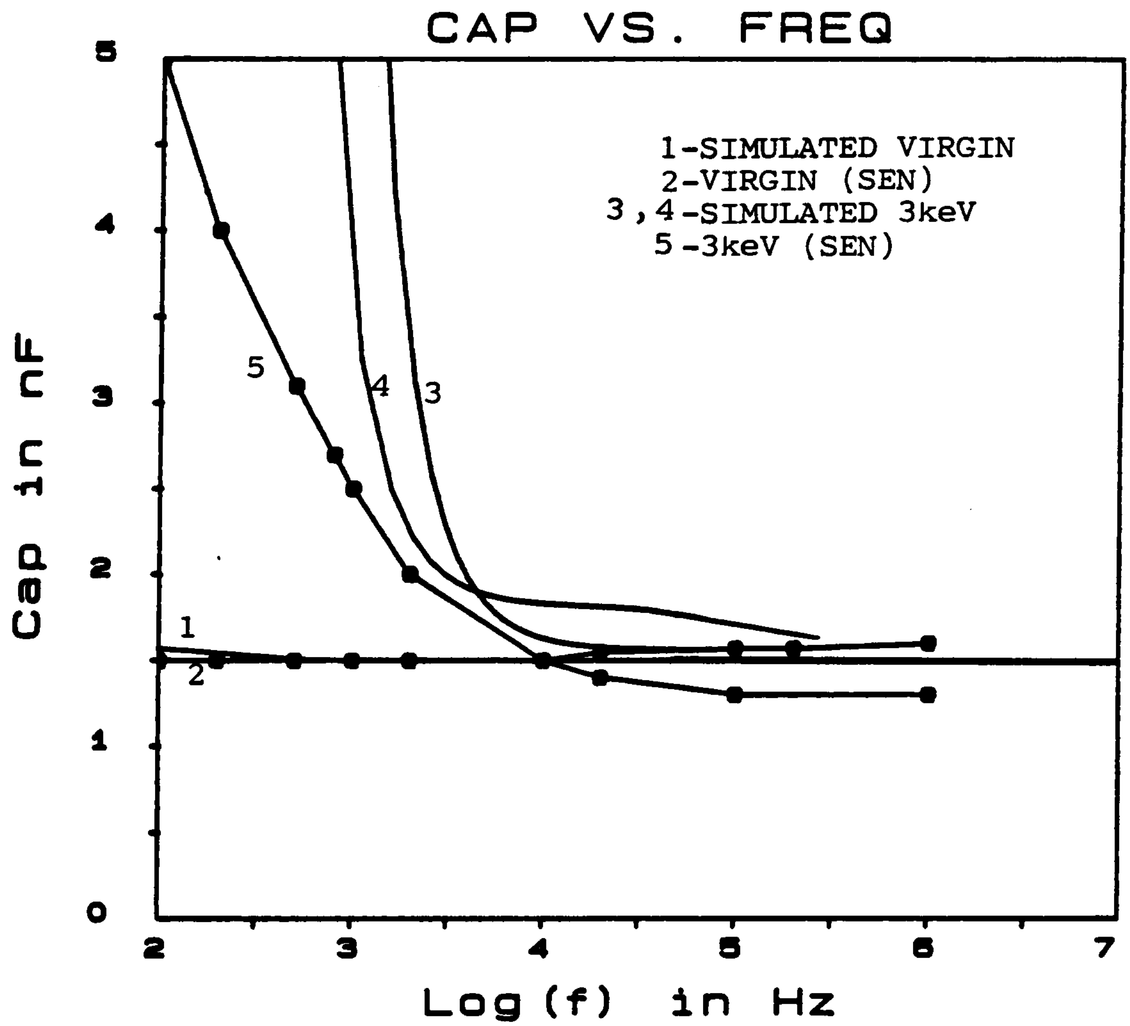


Figure 48. Capacitance-Frequency dispersion simulated using electrical model as compared to Sen's results.

EL2). As can be seen, the computer results accurately predict the dispersion, and compare well with his results. Therefore, this electrical model helps to explain both.

(It must be noted that the models used were first developed for Sen's work. It is therefore even more gratifying that with only the addition of C_T for DLTS simulation, the results could be simulated with the same model.)

Chapter 6: Summary and Conclusions

Several important and useful results were obtained from this research, and they are summarized below with recommendations for further analysis.

6.1 Summary

First, the fabrication process for "good" reproducible diodes was developed, starting from vendor wafers. This took approximately three months to perfect, using trial-and-error, changing such things as evaporation rates, chemical etch times and silver bonding epoxies. The result was discussed in section 3.1.

Second was the development of a fully automated test station, which allows the user to connect a sample and run experiments via a computer. The computer runs the electrical tests, collects the data and graphically displays it on either the screen, printer or plotter. The graphs in this dissertation are examples of that output. This system has not only capabilities for Schottky diodes but is adaptable to many other devices including the new high T_c superconductors.

Third was the analysis of the electrical effects of low energy argon ion etching of GaAs through the use of the electrical testing of Schottky diodes fabricated on these surfaces. A more complete understanding of this process was achieved, with the major results showing that:

1. At low energies (0.5keV) and moderate fluences IBE seems to create a thin (≈100Å), low resistance layer (higher resistance with respect to virgin samples) with possible near surface traps. This damaged layer causes leaky diodes with reduced reverse breakdown

voltages. The EL2 peak is hidden by the desensitizing effects of this layer and the corresponding electrical network (resulting from IBE) on DLTS. Low energies at low fluences produce no discernable damage to the crystal.

2. At higher energies (≈3keV) the IBE seems to generate a somewhat thicker (≈1000Å), higher resistance (due possibly to lowered mobility), partly amorphous layer with associated traps and defects which tend to increase V_B and V_T as well as making diodes leaky. These effects are essentially independent of fluence. Being of higher resistance, the conductivity, and thus the effect of the surface path, is all but eliminated from influencing the measurements as in the 0.5keV samples. The DLTS is also desensitized by this higher resistance series layer, which is sufficient to create an apparent loss of EL2.

3. Annealing at low temperatures (350°C) for 15min. resulted in no improvement in DLTS characteristics for both 0.5keV and 3keV samples. However, I-V and G-T characteristics showed movement for the 0.5keV sample towards virgin results, while the 3keV sample moved towards 0.5keV energy sample results. Though diode recovery was not achieved, sample changes could be seen.

4. The DLTS (usually considered a bulk measurement) appears to be quite sensitive to ion surface treatment through the correspondingly modified layers generated by IBE.

5. The EL2 DLTS peak appears to be useful as a sensitive monitor of surface layer changes due to IBE.

6. Computer analysis of the proposed physical and electrical models

successfully reproduced several experimental results, including DLTS spectra and previously reported capacitance-frequency dispersion. This indicates that some validity exists in these models.

6.2 Future Recommendations

A somewhat clearer picture of IBE damage and its electrical effects on Schottky diodes has been presented. However, there is room for improvement in several areas which would clarify the nature of the damage.

- Study IBE effects in MBE grown layers, since carrier mobility can be measured without the shunting effects of the bulk.
- Study IBE effects on MESFET operation.
- Extend the annealing study to optimize the process for temperature and time as functions of IBE energy and fluence to detail damage sensitivity to heat treatments.
- Perform photo-electric measurements on IBE surfaces such as spectral response and quantum efficiency to study surface related defects as could be seen in photocapacitance measurements.
- In situ deposition of Schottky metals directly after IBE to eliminate oxide effects caused by exposure to atmosphere.
- Extension of experiments to RIE and plasma etch for correlation with these results.
- More detailed modelling and computer analysis to include possible temperature and voltage dependence of the damage layer and network components to aid in understanding of damage and possible repair mechanisms.

This study, coupled with the above recommendations, could extend understanding, optimization and use of this most effective device processing technique.

REFERENCES

1. Ralph E. Williams, Gallium Arsenide Processing Techniques, ARTECH House, Inc. MA, p. 17, 1984.

2. M. J. Howes, D. V. Morgan, Gallium Arsenide Materials, Devices and Circuits, John Wiley & Sons, Chichester, p. 219, 1985.

3. S. M. Sze, Physics of Semiconductor Devices, 2nd Edition, John Wiley & Sons, New York, p. 13, 1981.

4. Same as [1], p. 22.

5. C. Kittel, Introduction to Solid State Physics, 5th ed. John Wiley & Sons, Inc., New York, p. 219, 1976.

6. Same as [3], p. 850.

7. Same as [2], p. 528.

8. Same as [3], p. 850.

9. Y. Yuba, M. Matsuo, K. Gamo and S. Namba, "Deep Levels in GaAs Induced by Electron, Proton and Heavy Ion Irradiations and their Annealing behavior at High Temperature," Thirteenth Int. Conf. on Defects in Semiconductors, Coranado, CA, pp. 973-979, Aug. 1984.

10. A. J. Grant, A. M. White and B. Day, "Low Frequency Noise and Deep Traps in Schottky Barrier Diodes," Proc. Int. Conf. on Noise, 5th, Nauheim, Ger., pp. 175-180, 1978.

11. J. Sikula, P. Vasina, V. Musilova, Z. Chobola and M. Rothbauer, "1/f Noise in GaAs Schottky Diodes," Phys. Stat. Sol., S4 693, pp. 693-696, 1984.

12. B. L. Sharma and R. K. Purohit, "The Variation of Schottky Barier Height Due to an Interfacial Layer," Surface Technology, 11, pp. 411-414, 1980.

13. S. Namba, M. Kawabe, N. Kanzaki and K. Masuda, "Photoluminescence measurement of ion-etched GaAs surface," J. Vac. Sci. Technol., Vol. 12, No. 6, pp. 1348-1351 Nov/Dec 1975.

14. S. Subramanian, S. Guha and B. M. Arora, "Effect of Heat Treatment on Electron Traps in n-Type GaAs," Solid-State Electronics, Vol. 20, pp. 799-802, 1977.

15. J. M. Borrego, R. J. Gutmann and S. Ashok, "Interface State Density in Au-nGaAs Schottky Diodes," Solid-State Elec., Vol. 20, pp. 125-132, 1977.

16. G. H. Glover, "Determination of Deep Levels in Semiconductors from C-V Measurements," IEEE Trans. on Elec. Dev., Vol. ED-19, No. 2, pp. 138-143, Feb. 1972.

17. Kazuo Sakai and Toshiaki Ikoma, "Deep Levels in Gallium Arsenide by Capacitive Methods," Appl. Phys. 5, pp. 165-171, 1974.

18. Michio Tajima, "Deep level photoluminescence commonly present in undoped Czochralski grown GaAs," Appl. Phys. Lett. 46(5), pp. 484-486, Mar. 1985.

19. S. Dhar, K. Seo and P. Bhattacharya, "Nature and distribtuion of electrically active defects in Si-implanted and lamp annealed GaAs," J. Appl. Phys. 58(11), pp. 4216-4220, Dec. 1985.

20. T. Nguyen, K. Wang, and G. Li, "Study of electric field enhanced emission of deep levels using a new emission spectroscopic technique," Appl. Phys. Lett. 44(2), pp. 211-213, Jan. 15, 1985.

21. E. Gombia, C. Ghezzi and R. Mosca, "Use of spatially dependent electron capture to profile deep-level densities in Schottky barriers," J. Appl. Phys. 58(3), pp. 1285-1291, 1 Aug. 1985.

22. R. Magno, et al., "Transient Capacitance Measurements on Neutron Irradiated Gallium Arsenide," Thirteenth Int. Conf. on Defects in Semiconductors, Coranado, CA, pp. 981-987, Aug. 1985.

23. F. Auret, W. Leitch and J. Vermaak, "A deep level transient spectroscopy analysis of electron and hole traps in bulk-grown GaAs," J. Appl. Phys. 59(1), pp. 158-163, 1 Jan. 1986.

24. S. Dindo, I. Abdel-Motaleb, K. Lowe, W. Tang and L. Young, "Deep Levels in Semi-Insulating LEC GaAs Before and After Silicon Implantation," J. Electrchem. Soc. Solid-State Sci. and Technol., pp. 2673-2677, Nov. 1985.

25. G. Baraff and M. Schluter, "Gallium Vacancy and EL2 in GaAs," Phys. Rev. Lett., Vol. 56, No. 11, pp. 1213-1214, 17 Mar. 1986.

26. M. Kaminska, M. Skowronski and W. Kuszko, "Identification of the 0.82eV Electron Trap, EL2 in GaAs, as an Isolated Arsenic Defect," Phys. Rev. Lett., Vol. 55, No. 20, pp. 2204-2207, 11 Nov. 1985.

27. K. Wada and N. Inove, "Effects of heat treatments on the near-surface distribution of EL2 defects," Appl. Phys. Lett. 47(9), pp. 945-947, 1 Nov. 1985.

28. M. Skowronski, J. Lagowski and H. Gatos, "Metastability of the mid-gap level EL2 in GaAs: Relationship with the As antisite defect," Phys. Rev. B, Vol. 32, No. 6, pp. 4264-4267, 15 Sept. 1985.

29. D. Sitrland, M. Brozel and I. Grant, "Direct observation of an enhanced concentration of the principal deep level EL2 at single dislocations," Appl. Phys. Lett 46(11), pp. 1066-1068, 1 June 1985.

30. S. Dhar, P. Bhattacharya, F. Juang, W. Houg and R. Sadler, "Dependence of Deep-Level Parameters in Ion-Implanted GaAs MESFET's on Material Preparation," IEEE Trans. Elec. Dev., Vol. ED-33, No. 1, pp. 111-117, 1 Jan. 1986.

31. P. Nayar and C. Penchina, "Deep Level Studies of Oxygen Doped GaAs-Au Schottky Barrier Diodes," J. Elec. Mat., Vol. 11, No. 2, pp. 321-333, 1982.

32. D. Stievenard and J. Bourgoin, "Impurity Defect Interactions in GaAs," J. Appl. Phys., 59(3), pp. 743-747, 1 Feb. 1986.

33. F. Auret and M. Nel, "Deep level transient spectroscopy of hole defects in bulk-grown p-GaAs using Schottky Barrier diodes," Appl. Phys. Lett. 48(2), pp. 130-132, 13 Jan. 1986.

34. Same as [2], p. 62.

35. J. C. Bourgoina nd D. Stievenard, "Comment on 'Identification of oxygen-related midgap level in GaAs,'" Appl. Phys. Lett. 46(3), pp. 311-312, Feb. 1, 1985.

36. G. Baraff and M. Schluter, "Gallium Vacancy and EL2 in GaAs," Phys. Rev. Lett., Vol. 56, No. 11, pp. 1213-1214, 17 Mar. 1986.

37. Same as [27].

38. Same as [28].

39. Same as [29].

40. Same as [31].

41. Conversation with Keith Ritala of Cominco.

42. Same as [2], p. 75.

43. Same as [1], p. 120.

44. Same as [1], p. 104.

45. Same as [2], p. 151, 156.

46. Same as [1], p. 199.

47. G. F. Feng, M. Holtz, R. Zallen, J. Epp, J. Dillard, E. Cole, P. Johnson, S. Sen, L. Burton, "Optical Chemical and Electrical Characterization of Ion-Etched Gallium Arsenide Surfaces," Proceedings of the Spring Meeting of the MRS, Vol. 93, p. 381, 1987.

48. A. Burenkov, and S. Amoretty, Tables of Ion Implantation Spatial Distributions, Gordona nd Breach, New York, p. 371, 1980.

49. S. P. Svenson, G. Landgrena nd T. G. Anderson, "Al-GaAs (001) Schottky barrier formation," J. Appl. Phys. 54(8), pp. 4474-4481, Aug. 1983.

50. N. Newman, W. Spicer, T. Kendelewicz, K. Chin and I. Lindau, "Electrical study of Schottky barriers on atomically clean GaAs (110) surfaces," Phys. Rev. B., Vol. 3, No. 2, pp. 1146-1158, 1986.

51. S. Offsey, J. Woodall, A. Warren, P. Kirchner, T. Chappell and G. Pettit, "Unpinned (100) GaAs surfaces in air using photochemistry," Appl. Phys. Lett. 48(7), pp. 475-477, 17 Feb. 1986.

52. Same as [2], p. 207.

53. Y. Chung, D. W. Langer, R. Becker and D. Look, "Modification of Surface Characteristics in GaAs with Dry Processing," IEEE Trans. Elec. Dev., Vol. ED-32, No. 1, pp. 40-44, Jan. 1984.

54. M. Kawabe, N. Kanzaki, K. Masuda and S. Namba, "Effects of Ion Etching on the Properties of GaAs," Applied Optics, Vol. 17, No. 16, pp. 2556-2561, Aug. 1978.

55. C. S. Wu, D. Scott, Wei-Xi Chen and S. Lau, "The Effects of Ion beam Etching on Si, Ge, GaAs, InP Schottky Barrier Diodes," J. Electrochem. Soc. Solid State Sci. and Tech., Vol. 132, No. 4, pp. 918-922, April 1985.

56. P. Kwan, K. Bhat, J. Borrego and S. Ghandhi, "Ion-cleaning Damage in (100) GaAs and its Effect on Schottky Diodes," Solid State Electronics, Vol. 26, No. 2, pp. 125-129, 1983.

57. S. W. Pang, "Dry Etching Induced Damage in Si and GaAs," Solid State Technology, pp. 249-256# April 1984.

58. S. K. Ghandi, P. Kwan, K. Bhat and J. Borrego, "Ion Beam Damage Effects During the Low Energy Cleaning of GaAs," IEEE Elec. Dev. Letters, Vol. EDL-3, No. 2, pp. 48-50, Feb. 1982.

59. K. Yamasaki, K. Asai, K. Shimada and T. Makimura, "Sputter Etching Effects on GaAs Schottky Junctions," J. Elctrochem. Soc. Solid State Sci. and Tech., Vol. 129, No. 12, pp. 2760-2764, Dec. 1982.

60. Y. X. Yang, P. H. Holloway, "Effect of Ion Sputtering on Interface Chemistry and Electrical Properties of Au GaAs (100) Schottky Contacts," J. Vac. Sci. B 2 (4), Oct-Dec 1984.

61. E. H. Nicollian, "Influence of a Thin Oxide Layer Between Metala nd Semiconductor on Schottky Diode Behavior," J. Vac. Sci. Tech., Vol. 13, No. 5, pp. 1047-1055, Oct. 1976.

62. Same as [47].

63. S. Sen, E. Cole and L. C. Burton, "Capacitance Spectroscopy of Schottky Diodes Formed on Ion-Etched GaAs, Mat. Res. Soc. Symp. Proc., Vol. 100, pp. 273-276, Dec. 1987.

64. G. I. Roberts and C. R. Crowell, Sol. St. Elec., Vol. 16, No. 1, pp. 29-32, 1973.

65. J. R. Morante, A. Lousa, J. Carceller and J. Morenza, Sol. St. Elec., Vol. 26, No. 6, pp 537-538, 1983.

66. F. Chenkir and C. Barret, Appl. Phys. Lett., Vol. 45, No. 11, pp. 1212-1214, 1984.

67. K. Hattori, T. Ohtani and T. Fugii, J. Appl. Phys., Vol. 59, No. 7, pp. 2507-2510, 1986.

68. E. H. Rhoderick, Metal-semiconductor Contacts, Clarendon Press, Oxford, p. 21. 1978.

69. Same as [3].

70. Same as [3], p. 291.

71. E. Yang, Fundamentals of Semiconductor Devices, McGraw-Hill, New York, 1978.

72. Same as [68], p. 27.

73. S. Pearton, et al., "Hydrogeneration of shallow-donor levels in GaAs," J. Appl. Phys. 59(8), pp. 2821-2827, 15 April 1986.

74. Same as [51].

75. Same as [68], p. 33.

76. Same as [3], p. 250.

77. Same as [68], p. 77-91.

78. Same as [37], p. 259-263.

79. Same as [3], p. 100.

80. Same as [68], p. 125.

81. Same as [3], p. 248.

82. Same as [68], p. 147-152.

83. Same as [2], p. 200.

84. Same as [1], p. 233.

85. Conversations with Rick Hudgens of Texas Instruments.

86. Seminar by Dr. Wu of IBM Thomas J. Watson Research Center.

87. Same as [85].

88. Same as [41].

89. Same as [3], p. 245-295.

90. D. V. Lang, "Space Charge Spectroscopy in Semiconductors," Thermally Stimulated Relaxation in Solids, Editor: P. Braunlich, Springer-Verlag, Berlin, pp. 93-133, 1979.

91. K. Yamasaki, K. Asai, K. Shimada and T. Makimura, "Sputter Etching Effects on GaAs Schottky Junctions," J. Elctrochem. Soc. Solid State Sci. and Tech., Vol. 129, No. 12, pp. 2760-2764, Dec. 1982.

92. Same as [47].

93. Same as [61].

94. Same as [2], p. 153.

95. Same as [61].

96. Same as [47].

97. Same as [2], p. 201.

98. Same as [48].

99. H. Holtz, Raman-Scattering Studies of the Structure of Ion-Implanted GaAs, Thesis, Va. Tech Dept. of Physics, August 1987.

100. Same as [54].

101. Same as [57].

102. Same as [85].

103. HP 4280A Operations Manual.

Appendix A: MEDUSA Software

The software is divided into three separate sections as mentioned earlier in 3.2 and the flowcharts are given in Fig. 49.

The system allows for 3 device environments to be used. These include:

1. Temperature sweep from 30K to 600K in increments of 1K.
2. Time sweep at a fixed holding temperature with experiments run every 5min. for an indefinite period.
3. Room temperature, no cryostat use.

Within these environments the sample may be exposed to either of 4 experimental groups as required by instrumentation. These include:

1. C-t, G-t, C-V, G-V, C, G
2. I-V, C-V
3. Mobility
4. 4-pt. Resistivity

After experimental completion the software automatically resets all instruments and cycles into a graphical display routine. These graphics allow for device analysis from the computer screen as mentioned in Section 3.2.

The complete documented software written in Better Basic follows given in the same four discrete sections as it was written and operates:

1. MEDUSA Batch: ties and runs all software together as a unit.
2. Parameter: experimental parameter set and error check.
3. Runnit: runs experiments and collects data.
4. Graphics: processes and displays data.

---

*Note assistance of Phillip Johnson

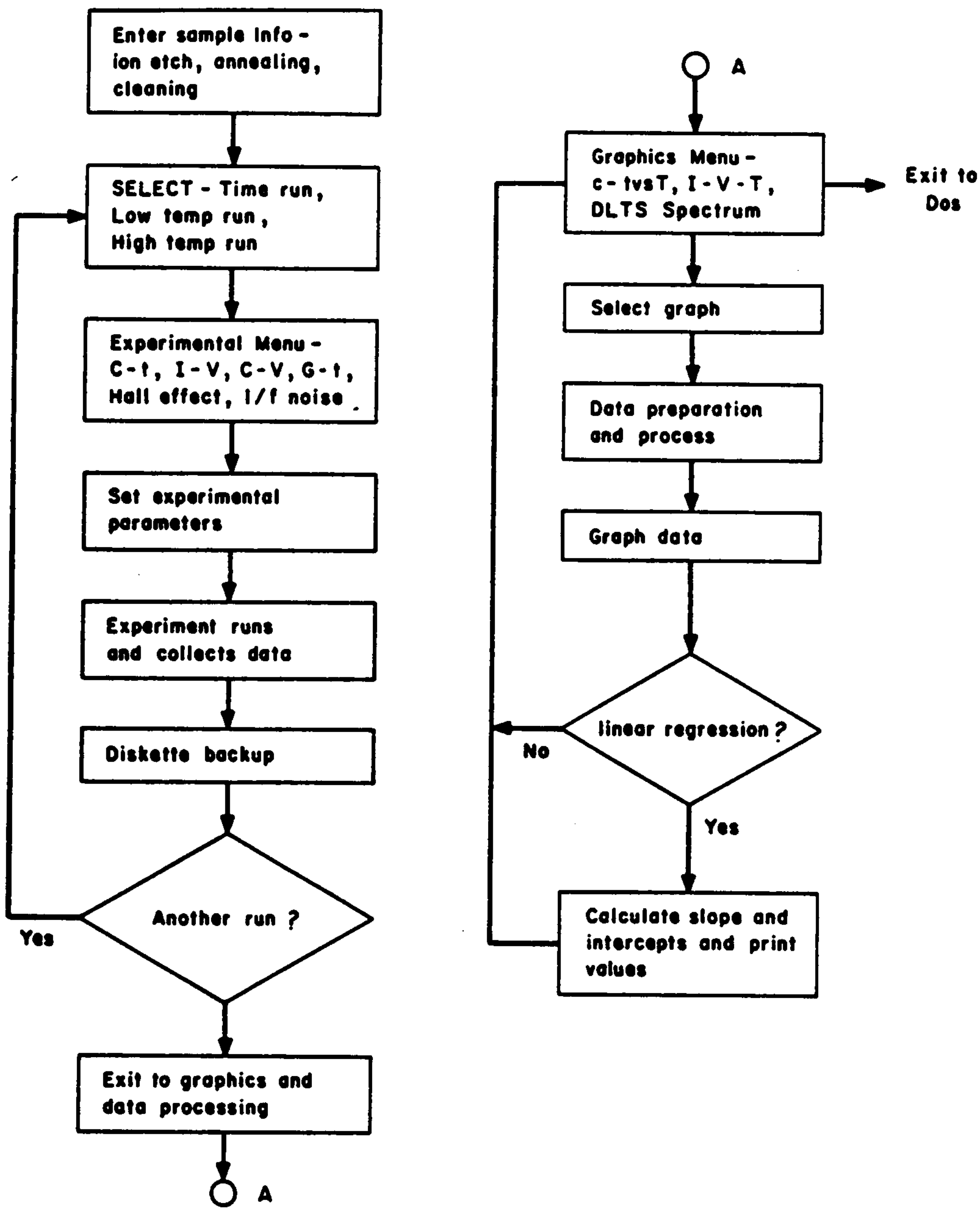


Figure 49. Flowchart for MEDUSA software.

```
ECHO OFF
:REPEAT
CD \BB
BB.COM \MEDUSA\NEW.CNF/C run "\MEDUSA\PARAMETER.CMP"
IF EXIST C:\DATA\END GOTO GRAPH
CD \BB
BB.COM \MEDUSA\NEW.CNF/C run "\MEDUSA\RUNIT.CMP"
IF EXIST C:\DATA\REDO GOTO REPEAT
:GRAPH
IF EXIST C:\DATA\END ERASE C:\DATA\END
CD \BB
BB.COM \MEDUSA\NEW.CNF/C run "\MEDUSA\GRAPHICS.CMP"
CLS
CD\
```

```
PROCEDURE: Border
   INTEGER: M%
       9 REM
      10 REM *********************************************
****************
      11 REM *
                 *
      12 REM *   The Procedure BORDER prints '*' around th
e menu.          *
      13 REM *
                 *
      14 REM *   to call type BORDER (ROWNUMBER%) (RETURN)
                 *
      15 REM *
                 *
      16 REM *   INTEGER:  M% - used in a FOR-NEXT loop
                 *
      17 REM *
                 *
      18 REM *   INTEGER ARGUMENT:  Rownumber%/VAR - used
to delineate     *
      19 REM *                      the maximum row for th
e '*'            *
      20 REM *
                 *
      21 REM *********************************************
****************
      22 REM
     100 COLOR 2,0
     110 LOCATE 3,1
     120 PRINT "*******************************************
*****************";
     130 PRINT "*********************"
     140 Rownumber% = Rownumber% + 1
     150 FOR M% = 4 TO Rownumber%
     160 LOCATE M%,1
     170 PRINT "*"
     180 LOCATE M%,79
     190 PRINT "*"
     200 NEXT M%
     210 Rownumber% = Rownumber% + 1
     220 LOCATE Rownumber%,1
     230 PRINT "*******************************************
*****************";
     240 PRINT "*********************"
     250 Rownumber% = 0
     260 COLOR 7,0
     270 EXIT
```

```
END PROCEDURE

PROCEDURE: Title
   INTEGER: Column%
      9 REM
     10 REM ***********************************************
****************
     11 REM *
                             *
     12 REM *    The Procedure TITLE prints the title of t
he menu                 *
     13 REM *    transferred from the main program by TITL
E$ in the               *
     14 REM *    location 1,col%
                             *
     15 REM *
                             *
     16 REM *    to call:  TITLE (COL%,TITLE$)
                             *
     17 REM *
                             *
     18 REM *    INTEGER ARGUMENT:  Col% - gives the 1-80
column number           *
     19 REM *
                             *
     20 REM *    STRING ARGUMENT:  Title$ - gives the titl
e string                *
     21 REM *
                             *
     22 REM ***********************************************
****************
     23 REM
    100 Column% = CINT((79 - LEN(Title$))/2)
    110 COLOR 5,0:CLS
    120 LOCATE 1,Column%
    130 PRINT Title$
    140 COLOR 7,0
    150 EXIT
END PROCEDURE

PROCEDURE: Menu
   INTEGER: Col%
   STRING: Number$[4]
   INTEGER: Row1%
      9 REM
     10 REM ***********************************************
****************
     11 REM *
                             *
     12 REM *    The Procedure MENU takes the names transf
erred by the            *
```

```
      13 REM *    calling program and prints them as a scre
en.              *
      14 REM *
                 *
      15 REM *    to call:  MENU (NUMBER%,ROWNUMBER%,NAME$,
ROW%)            *
      16 REM *
                 *
      17 REM *    INTEGER:  Col%
                 *
      18 REM *
                 *
      30 REM *    INTEGER ARGUMENT:  Number%/VAR,Rownumber%
/VAR,Row%/VAR    *
      32 REM *
                 *
      40 REM *    STRING:  Number$[4]
                 *
      43 REM *
                 *
      50 REM *    STRING ARGUMENT:  Name$
                 *
      59 REM *
                 *
      60 REM ******************************************
****************
      61 REM
     100 COLOR 3,0
     110 Number% = Number% + 1
     120 Number$ = STR$(Number%)
     130 DEL$(Number$,1,1)
     140 IF Number% < 13 THEN Col% = 5 ELSE Col% = 42
     150 IF Number% = 13 THEN Row% = 1 ELSE Row% = Row% +
1
     160 LOCATE Row%+4,Col%
     170 PRINT "(";Number$;") ";Name$
     180 IF Rownumber% < CSRLIN THEN Rownumber% = CSRLIN
     190 COLOR 7,0
     200 EXIT
END PROCEDURE

PROCEDURE: Finish
   STRING: Number$[4]
   INTEGER: Column%
       9 REM
      10 REM ******************************************
****************
      11 REM *
                 *
      12 REM *    The Procedure FINISH prints the message w
hich allows      *
```

```
       13 REM *   the user to exit the menu.
                *
       14 REM *
                *
       15 REM *   to call:  FINISH (ROWNUMBER%,NUMBER%,COL%
,NAME$)         *
       16 REM *
                *
       20 REM *   INTEGER ARGUMENT:  Rownumber%/VAR,Number%
/VAR,col%       *
       22 REM *
                *
       30 REM *   STRING:  Number$[4]
                *
       32 REM *
                *
       40 REM *   STRING ARGUMENT:  Name$
                *
       43 REM *
                *
       50 REM ******************************************
****************
       51 REM
      100 Column% = CINT((77 - LEN(Name$))/2)
      110 COLOR 3,0
      120 Rownumber% = Rownumber% + 1
      130 Number% = Number% + 1
      140 Number$ = STR$(Number%)
      150 DEL$(Number$,1,1)
      160 LOCATE Rownumber%,Column%
      170 PRINT "(";Number$;")  ";Name$
      180 COLOR 7,0
      190 EXIT
END PROCEDURE

PROCEDURE: Filecheck
   STRING: Chk$[16]
   INTEGER: T%,I%
   STRING: A$[?]
   INTEGER: Maxloop%
        9 REM
       10 REM ******************************************
****************
       11 REM *
                *
       12 REM *   The Procedure FILECHECK sees if the direc
tory specified *
       13 REM *   has been previously used.  If it has then
 it asks the    *
       14 REM *   user if they want to erase all previous f
iles.  If the   *
```

```
       15 REM *   directory has not been used, it then crea
tes the asked   *
       16 REM *   for directory.
                *
       17 REM *   to call:  FILECHECK (FILE$,REDO%)
                *
       18 REM *
                *
       20 REM *   INTEGER ARGUMENT:  Rownumber%/VAR, Number
%/VAR, col%     *
       23 REM *
                *
       30 REM *   STRING:  Number$[4]
                *
       32 REM *
                *
       40 REM *   STRING ARGUMENT:  Name$
                *
       43 REM *
                *
       50 REM ********************************************
****************
       51 REM
       99 CLOSE
      100 ON ERROR GOTO 10000
      110 OPEN "\DATA\" + File$ + "\INFO" FOR INPUT AS #1
      120 IF ERR = 1001 OR ERR=1007 THEN GOTO 180  'If no f
ile goto erase routine
      130 CLOSE #1
      140 LOCATE 20,5:COLOR 7,0
      150 PRINT "THE FILE NUMBER YOU CHOSE HAS ALREADY BEEN
 USED."
      160 INPUT "    DO YOU WANT TO OVERWRITE THE EXISTING
FILES (Defaults to NO)";Chk$
      170 IF INSTR(Chk$,"Y") <> 0 OR INSTR(Chk$,"y") <> 0 T
HEN GOTO 180 ELSE GOTO 1000
      180 RESTORE,50000
      190 READ Maxloop%
      210 I% = I% + 1
      220 READ A$
      230 KILL "\DATA\" + File$ + "\" + A$
      265 IF I% < Maxloop% THEN GOTO 210
      270 RMDIR "\data\" + File$
      280 MKDIR "\data\" + File$
      290 Redo% = 0
      300 GOTO 1010
     1000 Redo% = 1
     1010 ON ERROR 0
     1020 COLOR 7,0
     1030 LOCATE 19,5:PRINT SPC(79):PRINT SPC(79):PRINT SPC
(79):PRINT SPC(79)
```

```
  1040 CLEAR
  1050 EXIT
 10000 REM
 10001 REM *********************************************
****************
 10002 REM *   These are the ERROR handlers which allows
 the Procedure *
 10003 REM *   to check to see if the files or directory
 have been      *
 10004 REM *   used or exist.
               *
 10005 REM *
               *
 10006 REM *********************************************
****************
 10007 REM
 10010 IF ERR = 1001 THEN RESUME NEXT
 10020 IF ERR=18 THEN RESUME,270
 10030 IF ERR = 1007 THEN RESUME NEXT
 10040 IF ERR=1022 THEN RESUME,280
 10050 PRINT "I'm sorry but the FILECHECK procedure stil
l doesn't work."
 10060 PRINT "This is error number ";ERR;" from line "ER
L
 10070 END
 50000 REM
 50001 REM *********************************************
****************
 50002 REM *
               *
 50003 REM *   The DATA statements are for erasing the p
ossible files  *
 50004 REM *   from the hard disk.
               *
 50005 REM *
               *
 50006 REM *********************************************
****************
 50007 REM
 50010 DATA 9
 50020 DATA info,C-t,CGV,CG,G-t,IV,CV,Mob,Res
END PROCEDURE

PROCEDURE: Placeampersand
 INTEGER: Col%,Pos%
     9 REM
    10 REM *********************************************
****************
    11 REM *
               *
    12 REM *   The Procedure PLACEAMPERSAND takes the nu
```

```
mbers passed    *
      13 REM *   by CHOICE% and NUMBER% and arranges them
so that an      *
      14 REM *   "@" is placed in the appropriate place to
 show that      *
      15 REM *   the item chosen has actually been chosen.
                *
      16 REM *
                *
      17 REM *   to call:  PLACEAMPERSAND (NUMBER%,CHOICE%
)               *
      18 REM *
                *
      20 REM *   INTEGER:  Pos%, Col%
                *
      21 REM *
                *
      30 REM *   INTEGER ARGUMENT:  NUMBER%, CHOICE%
                *
      33 REM *
                *
      40 REM *******************************************
****************
      41 REM
     100 IF Number% < 12 THEN Col% = 3
     110 IF Number% > 12 THEN Col% = 40
     120 IF Choice% > 12 THEN Pos% = Choice% - 12 ELSE Pos
% = Choice%
     130 LOCATE Pos%+4,Col%
     140 PRINT "@"
     150 EXIT
END PROCEDURE

PROCEDURE: Removeampersand
   INTEGER: Col%,Pos%
     100 IF Number% < 12 THEN Col% = 3 ELSE Col% = 40
     110 IF Choice% > 12 THEN Pos% = 12 - Choice% ELSE Pos
% = Choice%
     120 LOCATE Pos%+4,Col%
     130 PRINT " "
     140 EXIT
END PROCEDURE

PROCEDURE: Settempparam
   REAL: Initialtemp!,Finaltemp!,Deltatemp!,Chk!,Chk1!,Min
!,Max!,Test!
   INTEGER: Error%,Line%
   STRING: Chk$[4]
   REAL: Test
   INTEGER: Delay%
   REAL: Tyme
```

```
    STRING: Sign$[16]
    REAL: Increment!,Incrementtemp!,Maxamimumtemp!
        9 REM
       10 REM ***************************************************
****************
       12 REM *   The Procedure SETTEMPPARAM allows the use
r to 1) set      *
       13 REM *   the initial,final and increment temperatu
re for each      *
       14 REM *   experiment, 2) set the temperature for a
TIME run, or     *
       15 REM *   3) choose a room temperature run.  This p
rocedure also    *
       16 REM *   has a variety of error checks to make sur
e the chosen     *
       17 REM *   temperatures are possible.
                 *
       18 REM *
                 *
       19 REM *
                 *
       20 REM *   to call:  SETTEMPPARAM (TIME%,COL%,INITIA
LTEMP,           *
       21 REM *                               FINALTEMP,DELTATEM
P,EXPT$,NAME$)   *
       22 REM *
                 *
       30 REM *   INTEGER:  error%,line%
                 *
       32 REM *
                 *
       40 REM *   INTEGER ARG:  time%,col%
                 *
       41 REM *         min!,max!,test!
                 *
       42 REM *
                 *
       50 REM *   REAL:  initialtemp!,finaltemp!,deltatemp!
,chk!,chk1!      *
       51 REM *         min!,max!,test!
                 *
       52 REM *
                 *
       60 REM *   REAL ARG:  minimumtemp!/VAR,maximumtemp/V
AR,              *
       61 REM *              incrementemp!/VAR
                 *
       62 REM *
                 *
       70 REM *   STRING:  chk$
                 *
```

```
 72 REM *
       *
 80 REM *   STRING ARG:  expt$,name$
       *
 82 REM *
       *
 85 REM ********************************************
****************
 86 REM
 90 ON ERROR GOTO 30000
 95 IF Time% = 3 THEN GOTO 800
100 IF Time% = 1 THEN GOTO 600
105 IF Expt$ <> "Cryostat" THEN GOTO 130
110 COLOR 2,0:CLS:LOCATE 1,Col%
120 PRINT "This sets the temperature for the ";Expt$
130 COLOR 4,0:LOCATE 3,5
140 PRINT "These are the Cryostat temperature ";Name$
;":"
150 COLOR 3,0:LOCATE 5,16
160 PRINT "Initial";SPC(15);"Final";SPC(15);"Incremen
t"
170 COLOR 9,0:LOCATE 6,17
180 PRINT Minimumtemp!;SPC(17);Maximumtemp!;SPC(18);I
ncrementemp!
190 COLOR 5,0:LOCATE 10,5
250 LOCATE 10,1:PRINT SPC(79)
260 LOCATE 10,5:INPUT "Enter the initial temperature"
;Initialtemp!
270 IF Initialtemp! >= Minimumtemp! THEN GOTO 290
280 Error% = 1:Line% = 1:GOTO 2000
290 IF Initialtemp! <= Maximumtemp! THEN GOTO 310
300 Error% = 2:Line% = 1:GOTO 2000
310 LOCATE 12,5:INPUT "Enter the final temperature";F
inaltemp!
320 IF Finaltemp! <= Maximumtemp! THEN GOTO 340
330 Error% = 2:Line% = 2:GOTO 2000
340 IF Finaltemp! >= Initialtemp! THEN GOTO 360
350 Error% = 8:Line% = 2:GOTO 2000
360 LOCATE 14,5:INPUT "Enter the temperature incremen
t";Deltatemp!
370 IF Deltatemp! > Incrementemp! THEN GOTO 410
380 IF Deltatemp! = 0 AND Initialtemp! = Finaltemp! T
HEN GOTO 410
390 IF Deltatemp! >= Incrementemp! THEN GOTO 410
400 Error% = 3:Line% = 3:GOTO 2000
410 GOSUB 1000
420 COLOR 7,0:LOCATE 20,1
430 INPUT "Do you want to change any of the above (De
faults to No)";Chk$
440 Chk$ = MID$(Chk$,1,1)
450 IF Chk$ = "Y" OR Chk$ = "y" THEN GOTO 110
```

```
     460 IF Expt$ = "Cryostat" THEN GOTO 560      'If for c
ryostat goto var. set
     470 IF SGN(Incrementemp!) = SGN(Deltatemp!) THEN GOTO
 560
     480 IF SGN(Incrementemp!) = SGN(Deltatemp!) THEN GOTO
 560
     490 Minimumtemp! = Finaltemp!
     500 Maximumtemp! = Initialtemp!
     510 Incrementemp! = SGN(Incrementemp!) * ABS(Deltatem
p!)
     520 EXIT
     560 Minimumtemp! = Initialtemp!
     570 Maximumtemp! = Finaltemp!
     580 Incrementemp! = Deltatemp!
     590 EXIT
     599 REM
     600 REM *********************************************
****************
     601 REM *
                *
     602 REM *   This section is used for a time run.  It
allows the      *
     603 REM *   user to enter the temperature which will
be used.        *
     604 REM *
                *
     605 REM *********************************************
****************
     606 REM
     610 COLOR 2,0:CLS:LOCATE 12,5
     620 PRINT "At what temperature do you want to do the
time run (must be ";
     630 PRINT " between ";Minimumtemp!;" and ";Maximumtem
p!;
     640 INPUT " K)";Min!
     650 IF Min! >= Minimumtemp! THEN GOTO 670
     660 Min! = 10:Error% = 1:Line% = 4:GOTO 2000
     670 IF Min! <= Maximumtemp! THEN GOTO 690
     680 Max! = 600:Error% = 2:Line% = 4:GOTO 2000
     690 Minimumtemp! = Min!
     700 Maximumtemp! = Min!
     710 Incrementemp! = 0
     720 EXIT
     799 REM
     800 REM
     810 CLS:COLOR 2,0:LOCATE 12,5
     820 INPUT "Enter The Room's Temperature(defaults to 2
90K)";Minimumtemp!
     825 IF Minimumtemp!=0 THEN Minimumtemp!=290
     830 Maximumtemp! = Minimumtemp!
     840 Incrementemp! = 400.0
```

```
 850 EXIT
 899 STOP
 999 REM
1000 REM *********************************************
****************
1001 REM *
               *
1002 REM *   This section checks to see if the tempera
tures set      *
1003 REM *   intersect those of the cryostat.
               *
1004 REM *
               *
1005 REM *********************************************
****************
1006 REM
1009 GOTO 1030
1010 IF Initialtemp! + ABS(Deltatemp!) < Finaltemp! TH
EN GOTO 1030
1020 Error% = 5:Line% = 1:GOTO 2000
1030 Chk! = (Initialtemp! - Minimumtemp!)/Incrementemp
!
1040 Chk1! = FIX(Chk!)
1050 Test! = Chk1! - Chk!
1060 IF Test < 0.0001 OR Test > 0.0001 THEN GOTO 1080
1070 Error% = 6:Line% = 1:GOTO 2000
1080 Chk! = Deltatemp!/Incrementemp!
1090 Chk1! = FIX(Chk!)
1100 Test! = Chk1! - Chk!
1110 IF Test < 0.0001 OR Test > 0.0001 THEN GOTO 1130
1120 Error% = 7:Line% = 3:GOTO 2000
1130 RETURN
1999 REM
2000 REM *********************************************
****************
2001 REM *
               *
2002 REM *   This is the section which prints up the e
rror messages  *
2003 REM *   for temperature settings which are not al
lowed.         *
2004 REM *
               *
2005 REM *********************************************
****************
2006 REM
2010 COLOR 7,0:LOCATE 20,5
2020 ON Error% GOSUB 2100,2150,2200,2250,2300,2350,240
0,2450
2030 Delay% = 3:GOSUB 20000
2040 COLOR 5,0:LOCATE 20,1:PRINT SPC(79):PRINT SPC(79)
```

```
2050 ON Line% GOTO 2500,2550,2600,2650
2099 REM
2100 REM ---------This is for a temperature less than
the minimum--------
2101 REM          -----------cryostat temperature------
-----
2102 REM
2110 PRINT "The temperature must be greater than ";Min
imumtemp!;"!"
2120 RETURN
2149 REM
2150 REM ---------This is for a temperature greater th
an the maximum---------
2151 REM              ----------cryostat temperature---
---------
2152 REM
2160 PRINT "The temperature must be less than ";Maximu
mtemp!;"!"
2170 RETURN
2199 REM
2200 REM ---------This is for an increment less than t
he cryostat----------
2201 REM              -----------temperature-----------
---
2202 REM
2210 PRINT "The temperature increment must be greater
than";Incrementemp!;"!"
2220 RETURN
2249 REM
2250 REM ------------This is for an increment which is
 going in-------------
2251 REM            ------------the wrong direction---
-----------
2252 REM
2255 IF SGN(Finaltemp! - Initialtemp!) = 1 THEN Sign$
= "positive"
2257 IF SGN(Finaltemp! - Initialtemp!) = -1 THEN Sign$
 = "negative"
2258 IF SGN(Finaltemp! - Initialtemp!) = 0 THEN Sign$
= "no slope"
2260 PRINT "The temperature increment must be ";Sign$;
"!"
2270 RETURN
2299 REM
2300 REM --------This is for an intersection of less t
han 2-------------
2301 REM
2310 PRINT "The parameters you have set has only 1 exp
erimental reading in"
2320 PRINT "it. It must have at least 2. ";
2340 RETURN
```

```
    2349 REM
    2350 REM ------------This is for a wrong initial tempe
rature--------------
    2351 REM
    2360 PRINT "The cryostat temperatures will never inter
sect with the initial";
    2370 PRINT "temperature set"
    2380 RETURN
    2399 REM
    2400 REM ----------For an impossible temperature incre
ment-----------
    2401 REM
    2410 PRINT "The increment you want is not possible!"
    2420 RETURN
    2450 REM
    2460 PRINT "The Final Temperature must be greater than
 ";Initialtemp!
    2470 RETURN
    2499 REM
    2500 REM ----------This sends the program back to the
initial temp----------
    2501 REM
    2510 GOTO 250
    2549 REM
    2550 REM -----This takes the experiment to the final t
emperature----
    2551 REM
    2560 LOCATE 12,1:PRINT SPC(79)
    2570 GOTO 310
    2599 REM
    2600 REM ---------This takes the experiment back to th
e increment----------
    2601 REM
    2610 LOCATE 14,1:PRINT SPC(79)
    2620 GOTO 360
    2649 REM
    2650 REM ------This takes the experiment back to the t
ime segment------
    2651 REM
    2660 GOTO 610
   19999 REM
   20000 REM *********************************************
****************
   20001 REM *
                *
   20002 REM *   This allows a timed delay for error messa
ges             *
   20003 REM *
                *
   20004 REM *********************************************
****************
```

```
    20005 REM
    20010 Tyme = TIMER + Delay%
    20020 IF Tyme > TIMER THEN GOTO 20020
    20030 RETURN
    30000 PRINT "The error is ";ERR;" and the line number i
s ";ERL
END PROCEDURE

PROCEDURE: Timeparam
    STRING: Chk$[4]
        9 REM
       10 REM ***********************************************
****************
       11 REM *
                 *
       12 REM *   The Procedure TIMEPARAM sets the final an
d incremental    *
       13 REM *   times for a constant temperature time run
.                *
       14 REM *
                 *
       15 REM *   to call:  TIMEPARAM (FINALTIME!,DELTATIME
!)               *
       16 REM *
                 *
       20 REM *   REAL ARG:  Finaltime!/VAR, Deltatime!/VAR
                 *
       22 REM *
                 *
       30 REM *   STRING:  Chk$[4]
                 *
       32 REM *
                 *
       40 REM ***********************************************
****************
       42 REM
      100 ON ERROR GOTO 10000
      110 COLOR 2,0:CLS:LOCATE 1,25
      120 PRINT "This sets the TIME parameters"
      130 COLOR 5,0
      140 LOCATE 4,3:INPUT "Enter the amount of time this p
rogram is to run (> 5 minutes)";Finaltime!
      150 LOCATE 6,3:PRINT "Enter the length of time betwee
n experiment";
      160 COLOR 3,0:PRINT " set ";
      170 COLOR 5,0:INPUT "runs ( > 5 minutes)";Deltatime!
      180 COLOR 7,0:LOCATE 20,5
      190 INPUT "Do you want to change any of the above (De
faults to No)";Chk$
      200 Chk$ = MID$(Chk$,1,1)
      210 IF Chk$ = "Y" OR Chk$ = "y" THEN GOTO 110
```

```
     220 Finaltime! = 60 * Finaltime!
     230 Deltatime! = 60 * Deltatime!
     240 EXIT
   10000 CLS:LOCATE 15,5:PRINT "The error is ";ERR;" in li
ne";ERL
   10010 STOP
END PROCEDURE

PROCEDURE: Svolt
   STRING: Chk$[3]
       9 REM
      10 REM ********************************************
****************
      11 REM *
             *
      12 REM *   The Procedure SVOLT allows the user to en
ter the      *
      13 REM *   voltages (including a break voltage and t
wo different  *
      14 REM *   increments.  It does not include any erro
r checking   *
      15 REM *   for entering incorrect voltages.
             *
      16 REM *
             *
      17 REM *   to call SVOLT (START!,STOP1!,STEP1!,STOP2
!,STEP2!)    *
      18 REM *
             *
      30 REM *   REAL ARGUMENTS:  Start!/VAR, Stop1!/VAR,
Stop2!/VAR    *
      32 REM *                    Step1!/VAR, Step2!/VAR
             *
      40 REM *
             *
      41 REM ********************************************
****************
      42 REM
     100 ON ERROR GOTO 10000
     110 LOCATE 3,1
     120 DO 20 TIMES
     130 PRINT SPC(79)
     140 REPEAT
     145 LOCATE 4,24:COLOR 4,0
     147 PRINT "This sets the Voltage settings"
     150 COLOR 5,0
     160 LOCATE 6,5:INPUT "Enter the Start Voltage (-100 t
o 100 V)";Start!
     170 LOCATE 8,5:INPUT "Enter the first Stop Voltage (-
100 to 100 V)";Stop1!
     180 LOCATE 10,5:INPUT "Enter the first Step Voltage (
```

```
> abs[0.01] V)";Step1!
     190 LOCATE 12,5:INPUT "Enter the second Stop Voltage
(-100 to 100 V])";Stop2!
     200 LOCATE 14,5:INPUT "Enter the second Step Voltage
(> abs[0.01] V)";Step2!
     210 COLOR 7,0:LOCATE 20,1
     220 INPUT "Do you want to change any of the above (De
faults to No)";Chk$
     230 Chk$ = MID$(Chk$,1,1)
     250 IF Chk$ = "Y" OR Chk$ = "y" THEN GOTO 110
     260 ON ERROR 0
     270 Step1! = VAL(STR$(Step1!))
     280 Step2! = VAL(STR$(Step2!))
     290 EXIT
   10000 REM
   10010 IF ERR = 1008 THEN RESUME
   10020 COLOR 7,0:LOCATE 18,3:PRINT "Sorry the procedure
";
   10030 COLOR 20,0:PRINT "SVOLT ";
   10040 COLOR 7,0:PRINT "is bombing.  This is error ";ERR
;" and line number";ERL
END PROCEDURE

PROCEDURE: DVdt
   STRING: Chk$[3]
       9 REM
      10 REM ************************************************
****************
      11 REM *
              *
      12 REM *   The Procedure dVdt allows the user the se
t the slope     *
      13 REM *   for the C-V experiment in group 2.
              *
      14 REM *
              *
      15 REM *   to call:  DVDT (Dvdt!,start!,stop!,step!)
              *
      16 REM *
             .*
      30 REM *   real arg:  dVdt!/VAR
              *
      32 REM *
              *
      40 REM ************************************************
****************
      41 REM
      47 REM *   real arg:start!/var,stop!/var,step!/var
     100 ON ERROR GOTO 10000
     110 LOCATE 3,1
     120 DO 20 TIMES
```

```
      130 PRINT SPC(79)
      140 REPEAT
      150 COLOR 4,0:LOCATE 4,23
      160 PRINT "This sets the dV/dt slope"
      170 COLOR 5,0
      180 LOCATE 6,5:INPUT "Enter the dV/dt setting (0.001
to 1 V/s)";DVdt!
      185 LOCATE 8,5:INPUT "Enter the Start Voltage (-100 t
o 100 V)";Start!
      187 LOCATE 10,5:INPUT "Enter the Stop Voltage (-100 t
o 100 V)";Stop!
      188 LOCATE 12,5:INPUT "Enter the Step Voltage (-10 to
 10 V)";Step!
      190 LOCATE 20,5:COLOR 7,0
      200 INPUT "Do you want to change any of the above (De
faults to No)";Chk$
      210 Chk$ = MID$(Chk$,1,1)
      220 IF Chk$ = "Y" OR Chk$ = "y" THEN GOTO 110
      230 EXIT
    10000 REM
    10001 REM ********************************************
****************
    10002 REM *
                *
    10003 REM *   This is the error procedures.  It correct
s the expected *
    10004 REM *   errors and kills the program and writes w
hich error     *
    10005 REM *   and line number for an unexpected error
                *
    10006 REM *
                *
    10007 REM ********************************************
****************
    10008 REM
    10010 IF ERR = 1008 THEN RESUME
    10020 COLOR 7,0:LOCATE 20,5:PRINT "Sorry the procedure
";
    10030 COLOR 20,0:PRINT "dVdt ";
    10040 COLOR 7,0:PRINT "bombing.  The error is ";ERR;" i
n line number";ERL
    10050 STOP
END PROCEDURE

PROCEDURE: Settime
    STRING: Chk$[3]
    REAL: Tyme!
    INTEGER: M%
       10 REM
       11 REM ********************************************
****************
```

```
12 REM *
                    *
13 REM *   The Procedure SETTIME allows the user to
enter the           *
14 REM *   Step Delay Time, Initial Hold Time, Final
 Hold Time          *
15 REM *   and Delta Hold Time in seconds.
                    *
16 REM *
                    *
17 REM *   to call:  SETTIME (STEPDELAY!,INITIALHOLD
!,FINALHOLD!,       *
18 REM *             DELTAHOLD!,MINHOLDTIME!,MINDELA
YTIME!,TYPE%)       *
19 REM *
                    *
20 REM *   REAL ARG: Stepdelay!/VAR, Initialhold!/VA
R,                  *
21 REM *             Deltahold!/VAR, Finalhold!/VAR
                    *
22 REM *
                    *
23 REM *********************************************
****************
24 REM
100 ON ERROR GOTO 10000
110 LOCATE 3,1
120 DO 21 TIMES
130 PRINT SPC(79)
140 REPEAT
150 LOCATE 4,25:COLOR 4,0
160 PRINT "This sets the Time Parameters"
170 COLOR 5,0
180 IF Type% = 3 THEN GOTO 210
190 LOCATE 6,5:PRINT "Enter the Step Delay Time, in s
econds (t > ";Mindelaytime!;
200 INPUT " seconds)";Stepdelay!
210 LOCATE 8,5:PRINT "Enter the Initial Hold Time, in
 seconds (t > ";Minholdtime!;
220 INPUT " seconds)";Initialhold!
230 IF Type% = 1 THEN GOTO 290
240 IF Type% = 3 THEN GOTO 290
250 LOCATE 10,5:PRINT "Enter the Final Hold Time, in
seconds (t > ";Minholdtime!;
260 INPUT " seconds)";Finalhold!
270 LOCATE 12,5:PRINT "Enter the Delta Hold Time, in
seconds (t > ";Minholdtime!;
280 INPUT " seconds)";Deltahold!
290 IF Stepdelay! < Mindelaytime! OR Initialhold! < M
inholdtime! THEN GOSUB 10100
300 LOCATE 20,5:COLOR 7,0
```

```
    310 INPUT "Do you need to change any of the above (De
faults to No)";Chk$
    320 Chk$ = MID$(Chk$,1,1)
    330 IF Chk$ = "Y" OR Chk$ = "y" THEN GOTO 110
    340 EXIT
   9999 STOP
  10000 REM
  10001 REM ***********************************************
****************
  10002 REM *
            *
  10003 REM *   This error routine takes care of any expe
cted errors.    *
  10004 REM *   If an unexpected error occurs, the progra
m prints the    *
  10005 REM *   procedure name, error number and line num
ber to screen   *
  10006 REM *   then ends the program.
            *
  10007 REM *
            *
  10008 REM ***********************************************
****************
  10009 REM
  10010 IF ERR = 1008 THEN RESUME
  10020 LOCATE 23,5:COLOR 7,0
  10030 PRINT "Sorry the Procedure ";
  10040 COLOR 20,0:PRINT "settime";
  10050 COLOR 7,0:PRINT " has bombed.  The error number i
s ";ERR;" from line ";ERL
  10099 STOP
  10100 REM
  10101 REM ***********************************************
****************
  10102 REM *
            *
  10103 REM *   This error is given if either the initial
 hold time or   *
  10104 REM *   the step delay time is less than the aske
d for values.   *
  10105 REM *   It then returns them to the beginning of
this routine.   *
  10106 REM *
            *
  10107 REM ***********************************************
****************
  10108 REM
  10110 LOCATE 20,5:COLOR 7,0
  10120 PRINT "The times you have chosen are not within t
he meter's resolution."
  10130 LOCATE 21,5:PRINT "Please choose different parame
```

```
ters."
   10140 Tyme! = TIMER + 5
   10150 IF TIMER < Tyme! THEN GOTO 10150
   10160 RETURN,100
   10170 STOP
END PROCEDURE

PROCEDURE: Setbias
   STRING: Chk$[3]
        9 REM
       10 REM *********************************************
****************
       11 REM *
                *
       12 REM *   The Procedure SETBIAS sets the different
bias voltages   *
       13 REM *
                *
       14 REM *   to call:  SETBIAS (INITIALBIAS!,FINALBIAS
!,DELTABIAS!    *
       15 REM *
                *
       30 REM *   real arg:initialbias!/VAR,deltabias!/VAR,
finalbias!/VAR *
       32 REM *
                *
       40 REM *********************************************
****************
       41 REM
      100 ON ERROR GOTO 10000
      110 LOCATE 3,1
      120 DO 21 TIMES
      130 PRINT SPC(79)
      140 REPEAT
      150 LOCATE 4,25:COLOR 4,0
      160 PRINT "This sets the Bias Voltages"
      170 COLOR 5,0
      180 LOCATE 6,5:INPUT "Enter the Initial Bias Voltage
(-99 to 99 V)";Initialbias!
      190 LOCATE 8,5:INPUT "Enter the Final Bias Voltage (-
99 to 99 V)";Finalbias!
      200 LOCATE 10,5:INPUT"Enter the Delta Bias Voltage (
> abs(0.01) V";Deltabias!
      210 LOCATE 20,5:COLOR 7,0
      220 INPUT "Do you need to change any of the above (De
faults to No)";Chk$
      230 Chk$ = MID$(Chk$,1,1)
      240 IF Chk$ = "Y" OR Chk$ = "y" THEN GOTO 110
      250 EXIT
    10000 REM
    10001 REM *********************************************
```

```
****************
   10002 REM *
               *
   10003 REM *   These are the error messages which take c
are of the     *
   10004 REM *   expected errors.  If an unexpected error
takes place    *
   10005 REM *   an error messages is given along with the
 line number   *
   10006 REM *   and the program stops.
               *
   10007 REM *
               *
   10008 REM *******************************************
****************
   10009 REM
   10010 IF ERR = 10008 THEN RESUME
   10020 LOCATE 23,5:COLOR 7,0
   10030 PRINT "The procedure ";
   10040 COLOR 20,1:PRINT "SETBIAS";
   10050 COLOR 7,0:PRINT "has bombed.  The error number is
 ";ERR;" and the line number ";ERL
   10060 STOP
END PROCEDURE

PROCEDURE: Setsamples
   STRING: Chk$[3]
      15 REM *   to call:  SETSAMPLES (samples!)
      30 REM *   REAL ARG:  SAMPLES!/var
               *
     100 ON ERROR GOTO 10000
     110 LOCATE 3,1
     120 DO 21 TIMES
     130 PRINT SPC(79)
     140 REPEAT
     150 LOCATE 4,25:COLOR 4,0
     160 PRINT "This sets the Number of Samples"
     170 COLOR 5,0
     180 LOCATE 6,5:INPUT "Enter the number of Samples nee
ded ( > 1)";SAMPLES!
     190 LOCATE 20,5:COLOR 7,0
     200 INPUT "Do you want to change any of the above (De
faults to No)";Chk$
     210 Chk$ = MID$(Chk$,1,1)
     220 IF Chk$ = "Y" OR Chk$ = "y" THEN GOTO 110
     230 SAMPLES! = INT(SAMPLES!)
     240 EXIT
   10000 STOP
   10001 REM *******************************************
****************
   10002 REM *
```

```
                *
   10003 REM *   This corrects any expected errors.  If an
 error is       *
   10004 REM *   unexpected the program bombs and an error
 message with   *
   10005 REM *   its line number.
                *
   10006 REM *
                *
   10007 REM ********************************************
****************
   10008 REM
   10010 IF ERR = 1008 THEN RESUME
   10020 LOCATE 1,5:COLOR 7,0
   10030 PRINT "The procedure ";
   10040 COLOR 20,0:PRINT "SETSAMPLES ";
   10050 COLOR 7,0:PRINT "has bombed.  The error is ";ERR;
" and line number ";ERL
   10060 STOP
END PROCEDURE

PROCEDURE: Setpulse
   STRING: Chk$[3]
       9 REM
      10 REM ********************************************
****************
      11 REM *
                *
      12 REM *   The Procedure SETPULSE allows the user to
 enter the      *
      13 REM *   Initial High Pulse, Final High Pulse, Del
ta High Pulse   *
      14 REM *   and Low Pulses for the voltages in the C-
t program       *
      15 REM *
                *
      16 REM *   to call:  SETPULSE (INITIALPULSE!,FINALPU
LSE!,           *
      17 REM *                   DELTAPULSE!,LOWPULSE!)
                *
      18 REM *
                *
      20 REM *   REAL ARG:  Initialpulse!/VAR, Finalpulse!
/VAR,           *
      21 REM *              Deltapulse!/VAR
                *
      22 REM *
                *
      30 REM *   STRING:  Chk$[3]
                *
      32 REM *
```

```
            *
      40 REM ***********************************************
****************
      42 REM
     110 LOCATE 3,1
     120 DO 21 TIMES
     130 PRINT SPC(79)
     140 REPEAT
     150 LOCATE 4,25:COLOR 4,0
     160 PRINT "This sets the Pulse Voltages"
     165 COLOR 5,0
     170 LOCATE 6,5:INPUT "Enter the Initial High Pulse (-
7 to +7 V)";Initialpulse!
     180 LOCATE 8,5:INPUT "Enter the Final High Pulse (-7
to +7 V)";Finalpulse!
     190 LOCATE 10,5:INPUT "Enter the Delta High Pulse ( )
 .01 V)";Deltapulse!
     200 LOCATE 12,5:INPUT "Enter the Low Pulse (-7 to +7
V)";Lowpulse!
     210 LOCATE 20,5:COLOR 7,0
     220 INPUT "Do you want to change any of the above (De
faults to No)";Chk$
     230 Chk$ = MID$(Chk$,1,1)
     240 IF Chk$ = "Y" OR Chk$ = "y" THEN GOTO 110
     250 IF Deltapulse! = 0 THEN Finalpulse! = 0
     260 EXIT
END PROCEDURE

PROCEDURE: Copytodisk
   INTEGER: Drive%
   STRING: Chk$[3],A$[?]
      20 REM *   integer:drive%
      30 REM *   string:chk$[4]
      40 REM *   STRING ARG:FILE$
     100 Drive% = 1
     110 CLS:COLOR 3,0:LOCATE 12,5
     120 PRINT "Do you want to copy the files to a ";
     130 COLOR 26,0
     140 ON Drive%  GOSUB 1000,2000
     150 LOCATE 12,44:COLOR 3,0:INPUT " floppy diskette (D
efaults to No)";Chk$
     160 Chk$ = MID$(Chk$,1,1)
     170 IF Chk$ = "Y" OR Chk$ = "y" THEN GOTO 180 ELSE GO
TO 280
     180 LOCATE 15,5:COLOR 6,0
     190 PRINT "Please insert a BLANK FORMATTED ";
     200 COLOR 24,0
     210 ON Drive% GOSUB 1000,2000
     220 LOCATE 15,7:COLOR 6,0:PRINT " floppy diskette int
o drive";
     230 ON Drive% GOSUB 1050,2050
```

```
     240 PRINT " and press any key to begin copying."
     250 A$ = INKEY$:IF A$ = "" THEN GOTO 250
     260 CLS
     270 ON Drive% GOSUB 1100,2100
     280 Drive% = Drive% + 1
     290 IF Drive% < 3 THEN GOTO 110 ELSE EXIT
    1000 REM
    1010 PRINT "1.2 Mbyte";
    1020 RETURN
    1050 PRINT "A";
    1060 RETURN
    1100 SHELL "\SEMI\COPYCTOA.BAT " + FILE$
    1110 RETURN
    2000 REM
    2010 PRINT "360 Kbyte";
    2020 RETURN
    2050 PRINT "B";
    2060 RETURN
    2100 SHELL "\SEMI\COPYCTOB.BAT " + FILE$
    2110 RETURN
END PROCEDURE

PROCEDURE: Timedelay
   REAL: Tyme!
      22 REM *   integer arg:delay%/opt = 5
      32 REM *   real:tyme!
     100 Tyme! = TIMER
     110 IF TIMER < Tyme! + Delay% THEN GOTO 110
     120 EXIT
END PROCEDURE

PROCEDURE: Clearscreen
   INTEGER: M%
       1 REM
       2 REM *********************************************
****************
       3 REM *
            *
       5 REM *   to call:  CLEARSCREEN (ROW%)
            *
      19 REM *
            *
      20 REM *   INTEGER ARG:row%
            *
      21 REM *
            *
      31 REM *********************************************
****************
      32 REM
     100 FOR M% = Row% TO 23
     110 LOCATE M%,1:PRINT SPC(79)
```

```
     120 NEXT M%
     130 EXIT
END PROCEDURE

'MAIN Program:


  200 REM
  201 REM *************************************************
*************
  202 REM *
           *
  203 REM *   This section prints a welcome message to the
 screen,   *
  204 REM *   clears all variables, changes the drive and
directory   *
  205 REM *   to "C:", sets the appropriate devices to rem
ote and     *
  206 REM *   initializes the error hander.
           *
  207 REM *
           *
  208 REM *************************************************
*************
  209 REM
  210 CLS:STATUSLINE OFF
  220 COLOR 2,0,8:LOCATE 12,30
  230 PRINT "MEDUSA welcomes you"
  240 LOCATE 14,12:PRINT "Materials and Electronic Device,
 Ultimate System Analyzer"
  250 Timedelay (5)
  260 CLEAR
  270 ON ERROR GOSUB 9500
  280 DRIVE$ = "C:":DIR$ = "\"
  290 PARAM$ = "INIT/1/&H310/P/":GOSUB 50000
  300 PARAM$ = "SDR/5,12,16,17,8/":GOSUB 50000
  310 REM
  311 REM *************************************************
*************
  312 REM *
           *
  313 REM *   This checks to see if the batch file has com
e from      *
  314 REM *   the program "RUNIT" and if it has it branche
s to        *
  315 REM *   see if the user wants to use the same inform
ation       *
  316 REM *   block.  If the batch file hasn't come from "
RUNIT" then *
  317 REM *   a message about turning on the printer is se
nt to       *
```

```
318 REM *   screen and the printer is sent a code to con
figure it.  *
319 REM *
        *
320 REM ************************************************
*************
321 REM
330 OPEN "\DATA\REDO" FOR INPUT AS #1
340 GOSUB 9000
350 COLOR 2,0,0:CLS:LOCATE 12,25
360 PRINT "Please turn on the printer"
390 Timedelay(3)
400 GOTO 1210
401 REM ************************************************
*************
402 REM *
        *
404 REM *   information to correctly identify the sample
.       *
405 REM *
        *
406 REM ************************************************
*************
407 REM
410 Redo% = 0
420 COLOR 2,0:CLS:LOCATE 3,5
430 INPUT "Enter the Directory number (between 1 and 999
)"File$
440 Information$(1) = File$
450 IF VAL(File$) < 1 OR VAL(File$) > 999 THEN GOTO 420
460 Filecheck (File$,Redo%)
470 IF Redo% = 1 THEN GOTO 410
480 Information$(2) = DATE$
490 COLOR 2,0
500 LOCATE 5,5
505 LINE INPUT "Enter Experimenter 's Name. ";Informatio
n$(3)
510 LOCATE 7,5
515 LINE INPUT "Enter the sample number. ";Information$(
4)
520 LOCATE 9,5
525 LINE INPUT "Enter the Cryostat chronometer reading.
";Information$(5)
530 LOCATE 11,5
535 LINE INPUT "Enter any comments. ";Information$(6)
760 LOCATE 20,5:COLOR 7,0
770 INPUT "Do you want to change any of the above (Defau
lts to No)";Chk$
780 Chk$ = MID$(Chk$,1,1)
790 IF Chk$ = "Y" OR Chk$ = "y" THEN GOTO 800 ELSE GOTO
1510
```

```
  800 REM
  801 REM ************************************************
*************
  802 REM *
          *
  803 REM *   This allows the user to change any part of t
he            *
  804 REM *   information block, using the various Procedu
res.          *
  805 REM *
          *
  806 REM ************************************************
*************
  807 REM
  810 Number% = 0:Row% = 0
  820 Title$ = "INFORMATION CHANGES"
  830 Title (Title$)
  840 RESTORE,60010
  850 READ Maxloop%
  860 DO Maxloop% TIMES
  870 READ Name$
  880 Menu (Number%, Rownumber%, Name$, Row%)
  890 REPEAT
  900 Name$ = "Finished changing information"
  910 Finish (Rownumber%,Number%,Name$)
  920 Border (Rownumber%)
  930 DO
  940 Clearscreen (20)
  950 COLOR 6,0:LOCATE 20,5
  960 INPUT "Enter which number you want to change";Choice
%
  970 IF Choice% < 1 OR Choice% > Number% THEN GOTO 940
  980 IF Choice% = Number% THEN EXIT TO,1510
  990 Placeampersand (Number%,Choice%)
  995 Clearscreen (20)
 1000 LOCATE 20,3:COLOR 4,0
 1010 PRINT "The current information is:  ";Information$(C
hoice%)
 1020 LOCATE 22,1:LINE INPUT "  What is the new informatio
n? ";Information$(Choice%)
 1030 IF Choice% <> 1 THEN GOTO 1100
 1040 Redo% = 0
 1050 Clearscreen (20)
 1060 Filecheck (Information$(1),Redo%)
 1070 IF Redo% = 1 THEN GOTO 1090
 1080 File$ = Information$(1)
 1090 Information$(1) = File$
 1100 Removeampersand (Number%,Choice%)
 1110 REPEAT
 1200 REM
 1201 REM ************************************************
```

```
*************
 1202 REM *
          *
 1203 REM *   This section allows the user to choose which
 type of    *
 1204 REM *   temperature run he wishes.  The choses are a
 1)Time run *
 1205 REM *   2)Temp run; 3)Room Temp; 4)Exit to graphs   *
 1206 REM *
          *
 1207 REM **********************************************
*************
 1208 REM
 1210 Number% = 0:Row% = 0
 1220 Title$ = "Menu for choosing TIME/TEMPERATURE Run"
 1230 Title (Title$)
 1240 RESTORE,60410
 1250 READ Maxloop%
 1260 DO Maxloop% TIMES
 1270 READ Name$
 1280 Menu (Number%,Rownumber%,Name$,Row%)
 1290 REPEAT
 1300 Name$ = "EXIT to Graphing Routines"
 1310 Finish (Rownumber%,Number%,Name$)
 1320 Border (Rownumber%)
 1330 Clearscreen (20)
 1340 COLOR 6,0:LOCATE 20,5
 1350 INPUT "Choose which type of run";Choice%
 1360 IF Choice% < 1 OR Choice% > Number% THEN GOTO 1330
 1370 IF Choice% = Number% THEN GOTO 8910
 1380 Expt%(26) = Choice%
 1390 RESTORE,60010
 1400 READ M%
 1410 M% = M% + 1
 1420 RESTORE,60600
 1430 DO Choice% TIMES
 1440 READ Information$(M%)
 1450 REPEAT
 1455 IF INSTR(UPPER$(Chk$),"Y")=0 THEN GOTO 410
 1500 REM
 1501 REM **********************************************
*************
 1502 REM *
          *
 1503 REM *   This section allows the user to choose betwe
en groups.  *
 1504 REM *
          *
 1505 REM **********************************************
*************
 1506 REM
```

```
 1510 Number% = 0:Row% = 0
 1520 RESTORE,60100
 1530 Title$ = "GROUP TYPES"
 1540 Title (Title$)
 1550 READ Maxloop%
 1560 DO Maxloop% TIMES
 1570 READ Name$
 1580 Menu (Number%,Rownumber%,Name$,Row%)
 1590 REPEAT
 1600 Border (Rownumber%)
 1610 Clearscreen (20)
 1620 COLOR 6,0:LOCATE 20,5
 1630 INPUT "Which group of experiments do you want to per
form";Choice%
 1640 IF Choice% < 1 OR Choice% > Number%  THEN GOTO 1610
 1650 Expt%(25) = Choice%
 1700 REM
 1701 REM ***********************************************
*************
 1702 REM *
           *
 1703 REM *   This section allows the user to pick which e
xperiments  *
 1704 REM *   he wants to run from the group chosen previo
usly        *
 1705 REM *
           *
 1706 REM ***********************************************
*************
 1707 REM
 1710 Number% = 0:Row% = 0
 1720 RESTORE,60200
 1730 Title$ = "EXPERIMENTAL MENU"
 1740 Title (Title$)
 1750 READ B%
 1760 FOR M% = 1 TO B%
 1770 READ Group%(M%)
 1780 NEXT M%
 1790 GOSUB 41000                        'Used to set Read
 Pointer
 1800 DO Group%(Expt%(25)) TIMES
 1810 READ Name$
 1820 Menu (Number%,Rownumber%,Name$,Row%)
 1830 REPEAT
 1840 Name$ = "FInished choosing experiments"
 1850 Finish (Rownumber%,Number%,Name$)
 1860 Border (Rownumber%)
 1870 DO
 1880 Clearscreen (20)
 1890 LOCATE 20,5:COLOR 6,0
 1900 INPUT "Which experiment do you want to run";Choice%
```

```
1910 IF Choice% < 1 OR Choice% > Number% THEN GOTO 1880
1920 IF Choice% = Number% THEN EXIT
1930 Placeampersand (Number%,Choice%)
1940 Expt%(Choice%) = Choice%
1950 REPEAT
1960 REM
1961 REM -------This asks if any experiments are to be de
leted-------
1962 REM
1970 DO
1980 Clearscreen (20)
1990 LOCATE 20,1:COLOR 7,0
2000 INPUT "Do you need to delete any of the above (Defau
lts to No)";Chk$
2010 Chk$ = MID$(Chk$,1,1)
2020 IF Chk$ = "Y" OR Chk$ = "y" THEN GOTO 2030 ELSE EXIT
2030 Clearscreen (20)
2040 LOCATE 20,1:COLOR 6,0
2050 INPUT "Which of the above do you need to delete";Cho
ice%
2060 IF Choice% < 1 OR Choice% > Number% - 1 THEN GOTO 20
90
2070 Removeampersand (Number%,Choice%)
2080 Expt%(Choice%) = 0
2090 REPEAT
2100 REM
2102 REM      -------This asks if any experiments are to
be added-------
2103 REM
2110 DO
2120 Clearscreen (20)
2130 LOCATE 20,1:COLOR 7,0
2140 INPUT "Do you want to add any to the above (Defaults
 to NO)";Chk$
2150 Chk$ = MID$(Chk$,1,1)
2160 IF Chk$ = "Y" OR Chk$ = "y" THEN GOTO 2170 ELSE EXIT
2170 Clearscreen (20)
2180 COLOR 6,0:LOCATE 20,5:INPUT "Which experiment do you
 want to add";Choice%
2190 IF Choice% < 1 OR Choice% > Number% - 1 THEN GOTO 22
20
2200 Expt%(Choice%) = Choice%
2210 Placeampersand(Number%,Choice%)
2220 REPEAT
2230 REM
2231 REM *********************************************
*************
2232 REM *
           *
2233 REM *   This section returns the program to the Temp
erature     *
```

```
2234 REM *   run type choosing section if no experiments
were chosen *
2235 REM *
         *
2236 REM ***********************************************
*************
2237 REM
2240 FOR M% = 1 TO 24
2250 IF Expt%(M%) <> 0 THEN EXIT TO,2500
2260 NEXT M%
2270 CLS:COLOR 7,0:LOCATE 12,30
2280 PRINT "NO experiments were chosen"
2290 Timedelay (4)
2300 GOTO 1210
2500 REM
2501 REM ***********************************************
*************
2502 REM *
         *
2503 REM *   This allows the user to set the Cryostat tem
perature(s) *
2504 REM *   and the times if it is a TIME run.
         *
2505 REM *
         *
2506 REM ***********************************************
*************
2507 REM
2510 ON Expt%(26) GOSUB 2600,2700,2900
2599 STOP
2600 REM
2601 REM ***********************************************
*************
2602 REM *
         *
2603 REM *   This subroutine allows the user to set the t
ime         *
2604 REM *   variables and the 1 temperature for a time r
un.         *
2605 REM *
         *
2606 REM ***********************************************
*************
2607 REM
2610 Parameter!(25,1) = 30.0
2620 Parameter!(25,2) = 600.0
2630 Parameter!(25,3) = 1.0
2640 GOSUB 3500
2650 Timeparam(Parameter!(25,5),Parameter!(25,6))
2660 RETURN,4000
2699 STOP
```

```
 2700 REM
 2701 REM *********************************************
*************
 2702 REM *
          *
 2703 REM *   This subroutine allows the user to set the t
emperatures *
 2704 REM *   for a temp run.
          *
 2705 REM *
          *
 2706 REM *********************************************
*************
 2707 REM
 2710 Parameter!(25,1) = 30.0
 2720 Parameter!(25,2) = 600.0
 2730 Parameter!(25,3) = 1.0
 2740 GOSUB 3500
 2750 RETURN,4000
 2799 STOP
 2900 REM
 2901 REM *********************************************
*************
 2902 REM *
          *
 2903 REM *   This subroutine allows the user to set the r
oom       *
 2904 REM *   temperature.
          *
 2905 REM *
          *
 2906 REM *********************************************
*************
 2907 REM
 2910 GOSUB 3500
 2930 RETURN,4000
 3499 STOP
 3500 REM
 3501 REM *********************************************
*************
 3502 REM *
          *
 3503 REM *   This section goes to the procedure SETTEMPPA
RAM to    *
 3504 REM *   allow the user to set what temperatures for
the       *
 3505 REM *   cryostat.
          *
 3506 REM *
          *
 3507 REM *********************************************
```

```
*************
 3508 REM
 3510 Col% = 20
 3520 Expt$ = "Cryostat"
 3530 Name$ = "limits"
 3540 Settempparam (Expt%(26),Col%,Parameter!(25,1),Parame
ter!(25,2),Parameter!(25,3),Expt$,Name$)
 3550 Parameter!(25,4) = Parameter!(25,1)
 3560 RETURN
 3999 STOP
 4000 REM
 4001 REM ************************************************
*************
 4002 REM *
         *
 4003 REM *   This section branches to the parameter setti
ng          *
 4004 REM *   subroutines selected by the user.
         *
 4005 REM *
         *
 4006 REM ************************************************
*************
 4007 REM
 4010 ON Expt%(25) GOTO 4100,4200,4300,4400
 4020 STOP
 4100 REM
 4101 REM ************************************************
*************
 4102 REM *
         *
 4103 REM *   This is to branch to the set-up routine for
the         *
 4104 REM *   experiments in group I.
         *
 4105 REM *
         *
 4106 REM ************************************************
*************
 4107 REM
 4110 L% = 1
 4120 ON Expt%(L%) GOSUB 10000,10000,12000,13000
 4130 L% = L% + 1
 4140 IF L% <= Group%(Expt%(25)) THEN GOTO 4120
 4150 GOTO 8010
 4199 STOP
 4200 REM
 4201 REM ************************************************
*************
 4202 REM *
         *
```

```
 4203 REM *   This is to branch to the set-up routine for
the          *
 4204 REM *   experiments in group II.
             *
 4205 REM *
             *
 4206 REM **************************************************
*************
 4207 REM
 4210 L% = 1
 4220 ON Expt%(L%) GOSUB 14000,15000
 4230 L% = L% + 1
 4240 IF L% <= Group%(Expt%(25)) THEN GOTO 4220
 4250 GOTO 8010
 4299 STOP
 4300 REM
 4301 REM **************************************************
*************
 4302 REM *
             *
 4303 REM *   This is the branch to Group III.  Since no p
arameters    *
 4304 REM *   need to be set it goes immediately to the lp
rint         *
 4305 REM *   statements.
             *
 4306 REM *
             *
 4307 REM **************************************************
*************
 4308 REM
 4310 CLS:LOCATE 1,20:COLOR 2,0
 4320 PRINT "This sets the";
 4330 COLOR 18,0:PRINT " MOBILITY";
 4340 COLOR 2,0:PRINT " Parameter Settings"
 4350 COLOR 14,0 : SET CURSOR 13,6 : INPUT "Enter the Curr
ent Bias in mA ( <9 mA ). ",Parameter!(1,17)
 4360 IF Parameter!(1,17) <= 0 OR Parameter!(1,17) > 9 THE
N GOTO 4310
 4370 Parameter!(1,17) = Parameter!(1,17) * 1E-03
 4380 GOTO 8010
 4400 REM
 4401 REM **************************************************
*************
 4402 REM *
             *
 4403 REM *   This is the branch to Group IV.  Since no pa
rameters     *
 4404 REM *   need to be set it goes immediately to the lp
rint         *
 4405 REM *   statements.
```

```
         *
 4406 REM *
         *
 4407 REM ****************************************************
*************
 4408 REM
 4410 CLS:LOCATE 1,15:COLOR 2,0
 4420 PRINT "This sets the";
 4430 COLOR 18,0:PRINT " 4-point Resistivity";
 4440 COLOR 2,0:PRINT " Parameter Settings"
 4450 COLOR 14,0 : SET CURSOR 13,6 : INPUT "Enter the Curr
ent Bias in mA ( <100 mA ). ",Parameter!(1,17)
 4460 IF Parameter!(1,17) <= 0 OR Parameter!(1,17) > 100 T
HEN GOTO 4410
 4480 GOTO 8010
 8000 REM
 8001 REM ****************************************************
*************
 8002 REM *
         *
 8003 REM *   This section saves all the information neces
sary to      *
 8004 REM *   running the experiments.  It also prints out
 a list      *
 8005 REM *   of the information block, the cryostat setti
ngs, the     *
 8006 REM *   time settings and all the parameters for eac
h            *
 8007 REM *   experiment chosen.
         *
 8008 REM *
         *
 8009 REM ****************************************************
*************
 8010 REM
 8011 REM -------This prints and saves the information blo
ck-------
 8012 REM
 8014 LPRINT CHR$(24);CHR$(27);CHR$(58)
 8015 LPRINT CHR$(27);CHR$(68);CHR$(35);CHR$(45);CHR$(55);
CHR$(65);CHR$(75);CHR$(85);CHR$(0)
 8020 OPEN "\DATA\" + File$ + "\INFO" FOR OUTPUT AS #1
 8030 RESTORE,60010
 8040 READ M%
 8050 Maxloop% = M% + 1
 8060 LPRINT CHR$(9);CHR$(9);CHR$(9);CHR$(9);CHR$(9);CHR$(
9);DATE$
 8070 FOR M% = 1 TO Maxloop%
 8080 READ Name$
 8090 PRINT #1,Name$
 8100 PRINT #1,Information$(M%)
```

```
8110 IF M% = 2 THEN GOTO 8140
8120 LPRINT SPC(10);Name$
8130 LPRINT SPC(10);Information$(M%)
8140 NEXT M%
8150 CLOSE #1
8160 REM
8161 REM  -------This prints and saves the Cryostat and T
ime Settings--------
8162 REM
8170 OPEN "\DATA\PARAM" FOR OUTPUT AS #1
8180 RESTORE,60320
8190 LPRINT CHR$(10);CHR$(10)
8200 LPRINT SPC(10);"These are the Cryostat and Time sett
ings"
8210 LPRINT
8220 FOR M% = 1 TO 6
8230 READ Name$
8240 PRINT #1,Parameter!(25,M%)
8250 IF M% = 4 THEN GOTO 8270
8260 LPRINT SPC(10);Name$,Parameter!(25,M%)
8270 NEXT M%
8280 CLOSE #1
8290 LPRINT CHR$(10)
8300 REM
8301 REM        -------This prints the parameter settings
-------
8302 REM
8310 LPRINT SPC(10)"These are the parameter settings for
the chosen experiments"
8320 GOSUB 40000
8330 LPRINT
8340 LPRINT SPC(10);"PARAMETER";CHR$(09);
8350 FOR M% = 1 TO Group%(Expt%(25))
8360 READ Name$
8370 IF Expt%(M%) = 0 THEN GOTO 8390
8380 LPRINT Name$;CHR$(09);
8390 NEXT M%
8400 LPRINT CHR$(10)
8410 RESTORE,60310
8420 READ Maxloop%
8430 FOR M% = 1 TO Maxloop%
8440 READ Name$
8450 IF M% > 3 AND M% < 7 THEN GOTO 8520
8460 LPRINT SPC(10);Name$;CHR$(09);
8470 FOR N% = 1 TO Group%(Expt%(25))
8480 IF Expt%(N%) = 0 THEN GOTO 8500
8490 LPRINT Parameter!(N%,M%);CHR$(09);
8500 NEXT N%
8510 LPRINT CHR$(13);
8520 NEXT M%
8530 LPRINT CHR$(12)
```

```
8540 REM
8541 REM -------This saves which experiments are to be ru
n-------
8542 REM
8550 OPEN "\DATA\EXPT" FOR OUTPUT AS #1
8560 PRINT #1,File$
8570 FOR M% = 1 TO 30
8580 PRINT #1,Expt%(M%)
8590 NEXT M%
8600 CLOSE #1
8610 REM
8611 REM -------This saves the parameters for the experim
ents-------
8612 REM
8620 OPEN "\DATA\PARAM" FOR APPEND AS #1
8630 RESTORE,60310
8640 READ Maxloop%
8650 FOR N% = 1 TO Group%(Expt%(25))
8660 IF Expt%(N%) = 0 THEN GOTO 8700
8670 FOR M% = 1 TO Maxloop%
8680 PRINT #1,Parameter!(N%,M%)
8690 NEXT M%
8700 NEXT N%
8710 CLOSE #1
8800 REM
8801 REM ***********************************************
*************
8802 REM *
          *
8803 REM *   This exits the "PARAMETER" program and goes
to the        *
8804 REM *   program "RUNIT".
          *
8805 REM *
          *
8806 REM ***********************************************
*************
8807 REM
8810 CLS
8820 SYSTEM
8899 STOP
8900 REM
8901 REM ***********************************************
*************
8902 REM *
          *
8903 REM *   This allows the user to go directly to the g
raphics       *
8904 REM *   routines if he has chosen to do so in the TE
MP RUN        *
8905 REM *   section.
```

```
          *
 8906 REM *
          *
 8907 REM **********************************************
*************
 8908 REM
 8910 OPEN "\DATA\END" FOR OUTPUT AS #1
 8920 PRINT #1,"Howdy there pardner"
 8930 CLOSE
 8940 CLS
 8950 SYSTEM
 8999 STOP
 9000 REM
 9001 REM **********************************************
*************
 9002 REM *
          *
 9003 REM *   This subroutine allows the user to reuse the
          *
 9004 REM *   information block used in the previous exper
iment.    *
 9005 REM *
          *
 9006 REM **********************************************
*************
 9007 REM
 9010 CLOSE
 9013 LPRINT CHR$(24)
 9020 CLS:COLOR 10,0,0:LOCATE 12,5
 9030 INPUT "Do you want to use the same information block
 (Defaults to No)";Chk$
 9040 Chk$ = MID$(Chk$,1,1)
 9050 IF Chk$ = "Y" OR Chk$ = "y" THEN GOTO 9090 ELSE GOTO
 9260
 9090 OPEN "\DATA\REDO" FOR INPUT AS #1
 9100 INPUT #1,FILE1$
 9110 CLOSE #1
 9160 RESTORE,60010
 9170 READ Maxloop%
 9180 OPEN "\DATA\" + FILE1$ + "\INFO" FOR INPUT AS #1
 9190 INPUT #1,BS$,BS$
 9200 FOR M% = 2 TO Maxloop%
 9210 INPUT #1,BS$,Information$(M%)
 9220 NEXT M%
 9230 CLOSE #1
 9233 Clearscreen (17):COLOR 10,0,0
 9234 LOCATE 17,5:INPUT "Enter the directory number (betwe
en 1 and 999)";File$
 9235 IF VAL(File$) < 1 OR VAL(File$) > 999 THEN GOTO 9233
 9236 Information$(1) = File$
 9237 Redo% = 0
```

```
 9238 Filecheck (File$,Redo%)
 9239 IF Redo% = 1 THEN GOTO 9233
 9240 KILL "\DATA\REDO"
 9250 GOTO 1210
 9260 KILL "\DATA\REDO"
 9270 GOTO 1210
 9499 STOP
 9500 REM ***************************************************
*************
 9501 REM *
          *
 9502 REM *   This is the ERROR handling routines.  It che
cks to see   *
 9503 REM *   what error is and attempts to fix it.
          *
 9504 REM *
          *
 9505 REM ***************************************************
*************
 9506 REM
 9510 IF ERR = 1008 THEN RESUME NEXT
 9520 IF ERR  = 1001 THEN RESUME,350
 9530 IF ERR = 1007 THEN RESUME,350
 9540 CLS:COLOR 7,0:LOCATE 11,5
 9550 PRINT"Sorry the main program is bombing.  This is th
e error ";ERR
 9560 LOCATE 12,5:PRINT "The line number is ";ERL
 9570 STOP
 9999 STOP
10000 REM
10001 REM ***************************************************
*************
10002 REM *
          *
10003 REM *   This is the Capacitance vs. Time and Conduct
ance vs.     *
10004 REM *   Time Subroutine.  It sets the parameters and
 then sends *
10005 REM *   them to the C-V meter and Pulse Generator fo
r checking. *
10006 REM *
          *
10007 REM ***************************************************
*************
10008 REM
10010 CLS:LOCATE 1,19:COLOR 2,0
10020 IF L% = 1 GOTO 10060              'If C-t skip next 2
 instructions
10030 Function$ = "FN6"
10040 Name$ = "G-t"
10050 GOTO 10080                        'Since G-t skip C-t
```

```
set-up
10060 Function$ = "FN5"
10070 Name$ = "C-t"
10080 COLOR 2,0:PRINT "This sets the ";
10090 COLOR 18,0:PRINT Name$;
10100 COLOR 2,0:PRINT " Experiment Parameters"
10110 Parameter!(L%,1) = Parameter!(25,1)
10120 Parameter!(L%,2) = Parameter!(25,2)
10130 Parameter!(L%,3) = Parameter!(25,3)
10140 Parameter!(L%,4) = Parameter!(25,4)
10150 IF Expt%(26) = 3 OR Expt%(26) = 1 THEN GOTO 10220
10160 REM
10161 REM              -----Sets Temperature Parameters-----
10162 REM
10170 Expt$ = Name$ + " Experiment"
10180 Name$ = "Settings"
10190 Col% = 20
10200 Settempparam (Time%,Col%,Parameter!(L%,1),Parameter!
(L%,2),Parameter!(L%,3),Expt$,Name$)
10210 REM
10211 REM                    -----Sets Bias Voltages-----
10212 REM
10220 Setbias (Parameter!(L%,17),Parameter!(L%,18),Paramet
er!(L%,19))
10230 Parameter!(L%,4) = Parameter!(L%,1)
10240 REM
10241 REM               -----Sets Number of Samples-----
10242 REM
10250 Setsamples (Parameter!(L%,20))
10260 REM
10261 REM                 -----Sets Pulse Times-----
10262 REM
10270 Settime (Parameter!(L%,13),Parameter!(L%,14),Paramet
er!(L%,15),Parameter!(L%,16),1.0E-05,1.0E-05,0)
10280 REM
10281 REM                 -----Sets Pulse Voltages-----
10282 REM
10290 Setpulse (Parameter!(L%,21),Parameter!(L%,22),Parame
ter!(L%,23),Parameter!(L%,24))
10300 REM
10301 REM ***********************************************
************
10302 REM *
       *
10304 REM *   This section checks the parameters with the
C-V meter   *
10305 REM *   and Pulse generator.
       *
10306 REM *
       *
10307 REM ***********************************************
```

```
*************
10308 REM
10310 REM----Checks the Initial High Pulse Voltage with th
e Pulse Generator----
10311 REM
10320 DO 2 TIMES
10330 PARAM$ = "SER.POLL/13/":GOSUB 50000
10340 PARAM$ = "SDR/13/":GOSUB 50000
10350 DATA_STRING$ = "M4,CTO,T1,W1,HIL" + STR$(Parameter!(
L%,21)) + "V,LOL" + STR$(Parameter!(L%,24)) + "V"
10360 PARAM$ = "WR.STR/13//EOS/":GOSUB 50000
10370 PARAM$ = "RD.STR/13//EOS/":GOSUB 50000
10380 PARAM$ = "SER.POLL/13/":GOSUB 50000
10390 REPEAT
10400 IF POLL_RESP% AND &H40 = 64 THEN GOSUB 10740
10410 IF Parameter!(L%,23) = 0 THEN GOTO 10520
10420 REM
10421 REM----Checks the Final High Pulse Voltage with the
Pulse Generator----
10422 REM
10430 DO 2 TIMES
10440 PARAM$ = "SER.POLL/13/":GOSUB 50000
10450 DATA_STRING$ = "M4,CTO,T1,W1,HIL" + STR$(Parameter!(
L%,22)) + "V,LOL" + STR$(Parameter!(L%,24)) + "V"
10460 PARAM$ = "WR.STR/13//EOS/":GOSUB 50000
10470 PARAM$ = "RD.STR/13//EOS/":GOSUB 50000
10480 PARAM$ = "SER.POLL/13/":GOSUB 50000
10490 REPEAT
10500 IF POLL_RESP% AND &H40 = 64 THEN GOSUB 10740
10510 PARAM$ = "SDL/13/":GOSUB 50000
10520 REM
10521 REM -----Checks the Hold Times vs. Bias Voltage with
 the C-V meter-----
10522 REM
10530 M% = 14
10540 N% = 17
10550 DATA_STRING$ = Function$ + "CN13TR3LE2SA1PC" + STR$(
Parameter!(L%,N%)) + ";PN" + STR$(Parameter!(L%,20)) + ";P
H" + STR$(Parameter!(L%,M%)) + ";PT" + STR$(Parameter!(L%,
13)) + "SW1"
10560 PARAM$ = "WR.STR/17//EOS/":GOSUB 50000
10565 DATA_STRING$ = "SWO"
10567 PARAM$ = "WR.STR/17//EOS/":GOSUB 50000
10570 PARAM$ = "SER.POLL/17/":GOSUB 50000
10580 IF POLL_RESP% AND &H40 = &H40 THEN GOSUB 10870
10590 N% = N% + 1
10600 IF Parameter!(L%,19) = 0 OR Parameter!(L%,18) = 0 TH
EN GOTO 10620
10610 IF N% = 18 THEN GOTO 10550
10620 IF Parameter!(L%,15) = 0 THEN GOTO 10660
10630 M% = M% + 1
```

```
10640 IF M% = 15 THEN GOTO 10540
10650 REM
10651 REM -------This checks to see if delaytime/holdtime
is > 200-------
10652 REM
10660 IF Parameter!(L%,14) / Parameter!(L%,13) > 200 THEN
GOTO 10680
10670 IF Parameter!(L%,16) / Parameter!(L%,13) <= 200 THEN
 GOTO 10710
10680 COLOR 7,0:CLS:LOCATE 13,5
10690 PRINT "The delay time divided by the pulse times mus
t be less than 200!"
10700 GOTO 10270
10710 PARAM$ = "SDC/13,17/":GOSUB 50000
10720 RETURN
10738 STOP
10739 REM
10740 REM ***********************************************
*************
10741 REM *
           *
10742 REM *   This section prints the error messages if an
y of the     *
10743 REM *   parameters evoke an "illegal" call from the
generator    *
10744 REM *   or C-V meter.
           *
10745 REM *
           *
10746 REM ***********************************************
*************
10747 REM
10748 REM      -----Prints out error message for the Pulse
Generator-----
10749 REM
10750 DATA_STRING$ = "IERR"
10760 PARAM$ = "WR.STR/13//EOS/":GOSUB 50000
10770 PARAM$ = "RD.STR/13//EOS/":GOSUB 50000
10780 Clearscreen (3)
10790 LOCATE 10,5:COLOR 7,0
10800 PRINT "The settings chosen are not viable.  You must
 select new settings."
10810 LOCATE 11,5:PRINT "To aid you the error from the HP8
112 A is "
10820 LOCATE 11,47:COLOR 5,0:PRINT MID$(DATA_STRING$,2,18)
:COLOR 7,0
10830 LOCATE 12,5:PRINT "Please look in the manual on page
s 3-21 to 3-23.)"
10840 PARAM$ = "SDL/13/":GOSUB 50000
10850 Timedelay(6)
10860 RETURN,10280
```

```
10869 STOP
10870 REM
10871 REM     -----Prints out error message for the C-V me
ter-----
10872 REM
10880 DATA_STRING$ = "ERR?"
10890 PARAM$ = "WR.STR/17//EOS/":GOSUB 50000
10900 PARAM$ = "RD.STR/17//EOS/":GOSUB 50000
10910 Startnoerror% = INSTR(DATA_STRING$,"ER00.0")
10920 IF Startnoerror% <> 0 THEN RETURN
10930 Clearscreen (3)
10940 LOCATE 10,5:COLOR 7,0
10950 PRINT "The settings chosen are not viable.  You must
 select new settings."
10960 PRINT "To aid you the error number from the HP4280 A
 is ";
10970 COLOR 5,0:PRINT DATA_STRING$:COLOR 7,0
10980 PRINT "Please look in the manual on pages 3-23 to 3-
30.)"
10990 Timedelay(6)
11000 PARAM$ = "SDC/17/":GOSUB 50000
11010 RETURN,10220
11999 STOP
12000 REM
12001 REM ************************************************
*************
12002 REM *
          *
12003 REM *   This subroutine control the C-G-V experiment
 and allows *
12004 REM *   the user to enter the parameter settings and
 checks for *
12005 REM *   any possible errors.
          *
12006 REM *
          *
12007 REM ************************************************
*************
12008 REM
12010 CLS:LOCATE 1,20:COLOR 2,0
12020 PRINT "This sets the";
12030 COLOR 18,0:PRINT " C-G-V";
12040 COLOR 2,0:PRINT " Parameter Settings"
12050 Parameter!(3,1) = Parameter!(25,1)
12060 Parameter!(3,2) = Parameter!(25,2)
12070 Parameter!(3,3) = Parameter!(25,3)
12080 Parameter!(3,4) = Parameter!(25,4)
12090 IF Expt%(26) = 3 OR Expt%(26) = 1 THEN GOTO 12150
12100 REM
12101 REM         -------Sets the Temperature Parameters--
-----
```

```
12102 REM
12110 Expt$ = "C-G-V Experiment"
12120 Col% = 19
12130 Name$ = "Settings"
12140 Settempparam (Time%,Col%,Parameter!(3,1),Parameter!(
3,2),Parameter!(3,3),Expt$,Name$)
12150 REM
12151 REM       --------Sets the Start, Stop and Delta Vol
tages----------
12152 REM
12160 Svolt (Parameter!(3,7),Parameter!(3,8),Parameter!(3,
9),Parameter!(3,10),Parameter!(3,11))
12170 REM
12171 REM       ---------Sets the Hold and Step Delay Time-
-------
12172 REM
12180 Settime (Parameter!(3,13),Parameter!(3,14),Parameter
!(3,15),Parameter!(3,16),0.1,0.1,1)
12200 REM
12201 REM ***********************************************
***************
12202 REM *
          *
12203 REM *     This checks the parameters entered with th
e C-V meter.  *
12204 REM *
          *
12205 REM ***********************************************
***************
12206 REM
12210 REM  --------This checks the first start, stop and d
elta voltages--------
12211 REM
12220 DATA_STRING$ = "FN1IB2TR3PS"+STR$(Parameter!(3,7))+"
;PP"+STR$(Parameter!(3,8))+";PE"+STR$(Parameter!(3,9))+";P
D"+STR$(Parameter!(3,13))+";PL"+STR$(Parameter!(3,14))
12230 PARAM$ = "WR.STR/17//EOS/":GOSUB 50000
12240 PARAM$ = "SER.POLL/17/":GOSUB 50000
12250 IF POLL_RESP% AND &H40 = 64 THEN GOSUB 12500
12260 REM
12261 REM  --------This checks the second start, stop and
delta voltages--------
12262 REM
12270 DATA_STRING$ = "FN1IB2TR3PS"+STR$(Parameter!(3,8))+"
;PP"+STR$(Parameter!(3,10))+";PE"+STR$(Parameter!(3,11))+"
;PD"+STR$(Parameter!(3,13))+";PL"+STR$(Parameter!(3,14))
12280 PARAM$ = "WR.STR/17//EOS/":GOSUB 50000
12290 PARAM$ = "SER.POLL/17/":GOSUB 50000
12300 IF POLL_RESP% AND &H40 = 64 THEN GOSUB 12500
12310 PARAM$ = "SDC/17/":GOSUB 50000
12320 RETURN
```

```
12499 STOP
12500 REM
12501 REM *************************************************
*************
12502 REM *
          *
12503 REM *   This subroutine prints the error found by th
e C-V     *
12504 REM *   meter, then returns the program to reenter t
he        *
12505 REM *   parameters.
          *
12506 REM *
          *
12507 REM *************************************************
*************
12508 REM
12510 DATA_STRING$ = "ERR?"
12520 PARAM$ = "WR.STR/17//EOS/":GOSUB 50000
12530 PARAM$ = "RD.STR/17//EOS/":GOSUB 50000
12540 Startnoerror% = INSTR(DATA_STRING$,"ER00.0")
12550 IF Startnoerror% <> 0 THEN RETURN
12560 Clearscreen (3)
12570 LOCATE 10,5:COLOR 7,0
12580 PRINT "The settings chosen are not viable.  You must
 select new settings."
12590 PRINT "To aid you the error number from the HP4280 A
 is ";
12600 COLOR 5,0:PRINT DATA_STRING$:COLOR 7,0
12610 PRINT "Please look in the manual on pages 3-23 to 3-
30.)"
12620 Timedelay(6)
12630 PARAM$ = "SDC/17/":GOSUB 50000
12640 RETURN,12150
12999 STOP
13000 REM
13001 REM *************************************************
*************
13002 REM *
          *
13003 REM *   This is the C-G subroutine.  It controls the
 parameter  *
13004 REM *   settings and also the data collecting and st
oring.    *
13005 REM *
          *
13006 REM *************************************************
*************
13007 REM
13010 CLS:LOCATE 1,20:COLOR 2,0
13020 PRINT "This sets the";
```

```
13030 COLOR 18,0:PRINT " C-G";
13040 COLOR 2,0:PRINT " Experiment Parameters"
13050 Parameter!(4,1) = Parameter!(25,1)
13060 Parameter!(4,2) = Parameter!(25,2)
13070 Parameter!(4,3) = Parameter!(25,3)
13080 Parameter!(4,4) = Parameter!(25,4)
13090 IF Expt%(26) = 3 OR Expt%(26) = 1 THEN GOTO 13150
13100 REM
13101 REM       -------Sets the Temperature Parameters---
----
13102 REM
13110 Expt$ = "C-G Experiment"                    'Sets title
and names for the
13120 Col% = 19                                   'Procedure
13130 Name$ = "Settings"
13140 Settempparam (Time%,Col%,Parameter!(4,1),Parameter!(
4,2),Parameter!(4,3),Expt$,Name$)
13150 REM
13151 REM         ---------Sets the Bias Voltages---------
-
13152 REM
13160 Setbias (Parameter!(4,17),Parameter!(4,18),Parameter
!(4,19))
13170 REM
13171 REM ************************************************
*************
13172 REM *
       *
13173 REM *   This section checks to see if the parameters
 set are     *
13174 REM *   allowed by the C-V meter.
       *
13175 REM *
       *
13176 REM ************************************************
*************
13177 REM
13178 REM      -------This checks the start voltage-------
-
13179 REM
13180 DATA_STRING$ = "FN1CN10IB1RA1MS2SL2TR3PV" + STR$(Par
ameter!(4,17))
13190 PARAM$ = "WR.STR/17//EOS/":GOSUB 50000
13200 PARAM$ = "SER.POLL/17/":GOSUB 50000
13210 IF POLL_RESP% AND &H40 = 64 THEN GOSUB 13500
13220 REM
13221 REM      -------This checks the final voltage-------
-
13222 REM
13230 DATA_STRING$ = "FN1CN10IB1RA1MS2SL2TR3PV" + STR$(Par
ameter!(4,18))
```

```
13240 PARAM$ = "WR.STR/17//EOS/":GOSUB 50000
13250 PARAM$ = "SER.POLL/17/":GOSUB 50000
13260 IF POLL_RESP% AND &H40 = 64 THEN GOSUB 13500
13270 PARAM$ = "SDC/17/":GOSUB 50000
13280 RETURN
13499 STOP
13500 REM
13501 REM ***********************************************
*************
13502 REM *
          *
13503 REM *   This subroutine prints the error found by th
e C-V       *
13504 REM *   meter, then returns the program to reenter t
he          *
13505 REM *   parameters.
          *
13506 REM *
          *
13507 REM ***********************************************
*************
13508 REM
13510 DATA_STRING$ = "ERR?"
13520 PARAM$ = "WR.STR/17//EOS/":GOSUB 50000
13530 PARAM$ = "RD.STR/17//EOS/":GOSUB 50000
13540 Startnoerror% = INSTR(DATA_STRING$,"ER00.0")
13550 IF Startnoerror% <> 0 THEN RETURN
13560 Clearscreen (3)
13570 LOCATE 10,5:COLOR 7,0
13580 PRINT "The settings chosen are not viable.  You must
 select new settings."
13590 PRINT "To aid you the error number from the HP4280 A
 is ";
13600 COLOR 5,0:PRINT DATA_STRING$:COLOR 7,0
13610 PRINT "Please look in the manual on pages 3-23 to 3-
30.)"
13620 Timedelay(6)
13630 PARAM$ = "SDC/17/":GOSUB 50000
13640 RETURN,13150
13999 STOP
14000 REM
14001 REM ***********************************************
*************
14002 REM *
          *
14003 REM *   This is the Current vs. Voltage subroutine.
 It sets    *
14004 REM *   up the parameters necessary to run the progr
am.  It     *
14005 REM *   It also checks to see if the parameters are
feasible.   *
```

```
14006 REM *
          *
14007 REM *************************************************
*************
14008 REM
14010 CLS:LOCATE 1,20:COLOR 2,0
14020 PRINT "This sets the ";
14030 COLOR 18,0:PRINT "I-V";
14040 COLOR 2,0:PRINT " Experiment Parameters"
14050 Parameter!(1,1) = Parameter!(25,1)
14060 Parameter!(1,2) = Parameter!(25,2)
14070 Parameter!(1,3) = Parameter!(25,3)
14080 Parameter!(1,4) = Parameter!(25,4)
14090 IF Expt%(26) = 3 OR Expt%(26) = 1 THEN GOTO 14150
14100 REM
14101 REM        -------Sets the Temperature Parameters----
----
14102 REM
14110 Expt$ = "I-V Experiment"
14120 Col% = 20
14130 Name$ = "Settings"
14140 Settempparam (Time%,Col%,Parameter!(1,1),Parameter!(
1,2),Parameter!(1,3),Expt$,Name$)
14150 REM
14151 REM        --------Sets the Start, Stop and Delta Vol
tages----------
14152 REM
14160 Svolt (Parameter!(1,7),Parameter!(1,8),Parameter!(1,
9),Parameter!(1,10),Parameter!(1,11))
14170 REM
14171 REM          --------Sets the Hold and Step Delay Tim
e--------
14172 REM
14180 Settime (Parameter!(1,13),Parameter!(1,14),Parameter
!(1,15),Parameter!(1,16),0.7,0.1,1)
14190 REM
14191 REM *************************************************
*************
14192 REM *
          *
14193 REM *   This checks the parameters set with the I-V
meter.     *
14194 REM *
          *
14195 REM *************************************************
*************
14196 REM
14200 M% = 7
14210 N% = 8
14215 X$ = STR$(VAL(STR$(Parameter!(1,N%+1))))
14220 DATA_STRING$ = "F2I2L3PS" + STR$(Parameter!(1,M%)) +
```

```
";PT" + STR$(Parameter!(1,N%)) + ";PE" + X$ + ";PH" + STR
$(Parameter!(1,14)) + ";PD" + STR$(Parameter!(1,13)) + ";W
1"
14230 PARAM$ = "WR.STR/5//EOS/":GOSUB 50000
14240 DATA_STRING$ = "W7"
14250 PARAM$ = "WR.STR/5//EOS/":GOSUB 50000
14260 PARAM$ = "SER.POLL/5/":GOSUB 50000
14270 IF POLL_RESP% AND &H40 = 64 THEN GOSUB 14500
14280 IF Parameter!(1,11) = 0 THEN GOTO 14320
14290 M% = M% + 1
14300 N% = N% + 2
14310 IF M% = 8 THEN GOTO 14215
14320 PARAM$ = "SDC/5/":GOSUB 50000
14330 RETURN
14499 STOP
14500 REM
14501 REM ************************************************
*************
14502 REM *
          *
14503 REM *   This prints the error message sent in binary
 by the      *
14504 REM *   I-V meter.  It then sends the program back t
o reenter    *
14505 REM *   the parameters.
          *
14506 REM *
          *
14507 REM ************************************************
*************
14508 REM
14510 Clearscreen (3)
14520 COLOR 7,0:LOCATE 10,5
14530 PRINT "The settings chosen are not viable.  You must
 select new settings."
14540 LOCATE 11,5
14550 PRINT "If you are not sure why this occurred please
check the HP 4140B."
14560 LOCATE 12,5:PRINT "The BINARY error code is ";
14570 COLOR 5,0:PRINT MID$(BIN$(POLL_RESP%),8)
14580 Timedelay(6)
14590 PARAM$ = "SDC/5/":GOSUB 50000
14600 RETURN,14160
14999 STOP
15000 REM
15001 REM ************************************************
*************
15002 REM *
          *
15003 REM *   This is the Capacitance vs. Voltage subrouti
ne.  It      *
```

```
15004 REM *   sets up the parameters necessary to run the
program.     *
15005 REM *   It also checks to see if the parameters are
feasible.    *
15006 REM *
             *
15007 REM ************************************************
*************
15008 REM
15010 CLS:LOCATE 1,20:COLOR 2,0
15020 PRINT "This sets the";
15030 COLOR 18,0:PRINT " C-V ";
15040 COLOR 2,0:PRINT "Experiment Parameters"
15050 Parameter!(2,1) = Parameter!(25,1)
15060 Parameter!(2,2) = Parameter!(25,2)
15070 Parameter!(2,3) = Parameter!(25,3)
15080 Parameter!(2,4) = Parameter!(25,4)
15090 IF Expt%(26) = 3 OR Expt%(26) = 1 THEN GOTO 15160
15100 REM
15101 REM       -------Sets the Temperature Parameters----
----
15102 REM
15110 Expt$ = "C-V Experiment"
15120 Col% = 20
15130 Name$ = "Settings"
15140 Settempparam (Time%,Col%,Parameter!(2,1),Parameter!(
2,2),Parameter!(2,3),Expt$,Name$)
15150 REM
15151 REM       --------Sets the Start, Stop and Delta Vol
tages----------
15152 REM
15160 DVdt (Parameter!(2,12),Parameter!(2,7),Parameter!(2,
8),Parameter!(2,9))
15170 REM
15171 REM              ----------Sets the Hold Time ------
----------
15172 REM
15180 Settime (Parameter!(2,13),Parameter!(2,14),Parameter
!(2,15),Parameter!(2,16),0.1,0,3)
15190 REM
15191 REM ************************************************
*************
15192 REM *
             *
15193 REM *  This checks the parameters set with the I-V
meter.       *
15194 REM *
             *
15195 REM ************************************************
*************
15196 REM
```

```
15004 REM *   sets up the parameters necessary to run the
program.     *
15005 REM *   It also checks to see if the parameters are
feasible.    *
15006 REM *
             *
15007 REM *********************************************
*************
15008 REM
15010 CLS:LOCATE 1,20:COLOR 2,0
15020 PRINT "This sets the";
15030 COLOR 18,0:PRINT " C-V ";
15040 COLOR 2,0:PRINT "Experiment Parameters"
15050 Parameter!(2,1) = Parameter!(25,1)
15060 Parameter!(2,2) = Parameter!(25,2)
15070 Parameter!(2,3) = Parameter!(25,3)
15080 Parameter!(2,4) = Parameter!(25,4)
15090 IF Expt%(26) = 3 OR Expt%(26) = 1 THEN GOTO 15160
15100 REM
15101 REM       -------Sets the Temperature Parameters----
----
15102 REM
15110 Expt$ = "C-V Experiment"
15120 Col% = 20
15130 Name$ = "Settings"
15140 Settempparam (Time%,Col%,Parameter!(2,1),Parameter!(
2,2),Parameter!(2,3),Expt$,Name$)
15150 REM
15151 REM       --------Sets the Start, Stop and Delta Vol
tages----------
15152 REM
15160 DVdt (Parameter!(2,12),Parameter!(2,7),Parameter!(2,
8),Parameter!(2,9))
15170 REM
15171 REM             ----------Sets the Hold Time ------
----------
15172 REM
15180 Settime (Parameter!(2,13),Parameter!(2,14),Parameter
!(2,15),Parameter!(2,16),0.1,0,3)
15190 REM
15191 REM *********************************************
*************
15192 REM *
             *
15193 REM *   This checks the parameters set with the I-V
meter.       *
15194 REM *
             *
15195 REM *********************************************
*************
15196 REM
```

```
15200 DATA_STRING$ = "F3I2L3PS" + STR$(Parameter!(2,7)-Par
ameter!(2,9)) +";PT" + STR$(Parameter!(2,8)+Parameter!(2,9
)) + ";PE" + STR$(Parameter!(2,9)) + ";PV" + STR$(Paramete
r!(2,12)) + ";PH" + STR$(Parameter!(2,14)) + ";W1"
15210 PARAM$ = "WR.STR/5//EOS/":GOSUB 50000
15215 DATA_STRING$ = "W7"
15218 PARAM$ = "WR.STR/5//EOS/":GOSUB 50000
15220 PARAM$ = "SER.POLL/5/":GOSUB 50000
15230 IF POLL_RESP% AND &H40 = 64 THEN GOSUB 15500
15240 PARAM$ = "SDC/5/":GOSUB 50000
15250 RETURN
15500 REM
15501 REM ************************************************
*************
15502 REM *
          *
15503 REM *   This prints the error message sent in binary
 by the     *
15504 REM *   I-V meter.  It then sends the program back t
o reenter   *
15505 REM *   the parameters.
          *
15506 REM *
          *
15507 REM ************************************************
*************
15508 REM
15510 Clearscreen (3)
15520 COLOR 7,0:LOCATE 10,5
15530 PRINT "The settings chosen are not viable.  You must
 select new settings."
15540 LOCATE 11,5
15550 PRINT "If you are not sure why this occurred please
check the HP 4140B."
15560 LOCATE 12,5:PRINT "The BINARY error code is ";
15570 COLOR 5,0:PRINT MID$(BIN$(POLL_RESP%),8)
15580 Timedelay(6)
15590 PARAM$ = "SDC/5/":GOSUB 50000
15600 RETURN,15150
39999 STOP
40000 REM
40001 REM ************************************************
*************
40002 REM *
          *
40003 REM *   This subroutine places the data pointer to t
he right    *
40004 REM *   experimental group, for the print out.
          *
40005 REM *
          *
```

```
40006 REM ****************************************************
*************
40007 REM
40010 ON Expt%(25) GOTO 40020,40030,40040,40045
40020 RESTORE,60510:GOTO 40050
40030 RESTORE,60520:GOTO 40050
40040 RESTORE,60530:GOTO 40050
40045 RESTORE,60540:GOTO 40050
40050 RETURN
40999 STOP
41000 REM
41001 REM ****************************************************
*************
41002 REM *
                         *
41003 REM *   This subroutine places the data pointer to t
he right     *
41004 REM *   experimental group, for the experiment menu.
                         *
41005 REM *
                         *
41006 REM ****************************************************
*************
41007 REM
41010 ON Expt%(25) GOTO 41020,41030,41040,41050
41020 RESTORE,60220:GOTO 41060
41030 RESTORE,60230:GOTO 41060
41040 RESTORE,60240:GOTO 41060
41050 RESTORE,60250:GOTO 41060
41060 RETURN
49988 REM            IEEE-488 INTEFACE FOR THE IBM PC V4_2
49989 REM                   WRITTEN IN ADVANCED BASIC
49990 REM AND INCORPORATING ASSEMBLY LANGUAGE ROUTINES TO
IMPLEMENT
49991 REM                  DMA - DRIVEN GPIB TRANSACTIONS
49992 REM THE ASSEMBLY LANGUAGE ROUTINES MUST BE LOADED PR
IOR TO ENTERING
49993 REM BASICA BY TYPING "SUBLIB"_  THEN TYPE "BASICA",
LOAD IEEE488_BAS,
49994 REM AND CALL SUBROUTINES AS DESCRIBED IN THE MANUAL_
49995 REM
49996 REM (C) Copyright Scientific Solutions, Inc_ 1982,19
83,1984,1985
49997 REM
49998 REM ******************** START OF SUBROUTINE ******
*********************
49999 STOP
60000 REM
60001 REM ****************************************************
*************
60002 REM *
```

```
        *
60003 REM *   These DATA statements are used for the infor
mation        *
60004 REM *   block print statements and correction routin
e.            *
60005 REM *
              *
60006 REM *************************************************
*************
60007 REM
60010 DATA 6
60020 DATA "Directory Number","Date ","Experimenters Name"
,"Sample Number","Cryostat Chronometer Reading","Comments"
,"Type of Run"
60099 STOP
60100 REM
60101 REM *************************************************
*************
60102 REM *
              *
60103 REM *   These DATA statements allow the user to choo
se which      *
60104 REM *   group of experiments to run.
              *
60105 REM *
              *
60106 REM *************************************************
*************
60110 DATA 4
60120 DATA "GROUP I   C-t, G-t, C-G-V, C-G"
60130 DATA "GROUP II  I-V, C-V"
60140 DATA "GROUP III  Van der Pauw/Mobility"
60150 DATA "GROUP IV  4-point Resistivity"
60199 STOP
60200 REM
60201 REM *************************************************
*************
60202 REM *
              *
60203 REM *   These DATA statements allow the user to choo
se which      *
60204 REM *   experiments he wants to run from the group c
hoosen        *
60205 REM *   earlier.
              *
60206 REM *
              *
60207 REM *************************************************
*************
60208 REM
60210 DATA 4,4,2,1,1
```

```
60220 DATA "Capacitance (C) vs. Time","Conductance (G) vs.
 Time","C and G vs. Voltage","Capacitance and Conductance"
60230 DATA "Current vs. Voltage","Capacitance vs. Voltage"
60240 DATA "Van der Pauw/Mobility"
60250 DATA "4-point Resistivity"
60299 STOP
60300 REM
60301 REM ************************************************
*************
60302 REM *
          *
60303 REM *   These DATA statements are for the PARAMETER
(x,y)     *
60304 REM *   array.
          *
60305 REM *
          *
60306 REM ************************************************
*************
60307 REM
60310 DATA 24
60320 DATA "Initial Temp","Final Temp","Delta Temp",Curren
t Temp"
60330 DATA "Final Time","Delta Time"
60340 DATA "Start Volt","Stop1 Volt","Step1 Volt","Stop2 V
olt","Step2 Volt"
60350 DATA "DV/dt"
60360 DATA "Delay Time","Initial Hold","Final Hold","Delta
 Hold"
60370 DATA "Initial Bias","Final Bias","Delta Bias"
60380 DATA "Number of Samples"
60390 DATA "Start High Pulse","Final High Pulse","Delta Hi
gh Pulse","Low Pulse"
60399 STOP
60400 REM
60401 REM ************************************************
*************
60402 REM *
          *
60403 REM *   This data statement is used for the type of
temperature *
60404 REM *   run section.
          *
60405 REM *
          *
60406 REM ************************************************
*************
60407 REM
60410 DATA 3
60420 DATA "Time Run ( > 5 Minutes; 30-600 K)","Temperatur
e Run (30 - 600 K)","Room Temperature Run (290 K)"
```

```
60499 STOP
60500 REM
60501 REM ************************************************
*************
60502 REM *
            *
60503 REM *   These data statements are for the print out
of the      *
60504 REM *   parameter set.
            *
60505 REM *
            *
60506 REM ************************************************
*************
60507 REM
60510 DATA "C-t","G-t","C-G-V","C-G"
60520 DATA "I-V","C-V"
60530 DATA "Van der Pauw"
60540 DATA "4-point Resistivity"
60599 STOP
60600 REM
60601 REM ************************************************
***************
60602 REM *
               *
60603 REM *     This data statement gives what type of tem
perature run   *
60604 REM *     it is.
               *
60605 REM *
               *
60606 REM ************************************************
***************
60607 REM
60610 DATA "TTEMP","LTEMP","RTEMP"

ENDFILE
```

```
STRING: X$[?]
REAL: TEMP_SET!,TEMP_PEAK!,Temp1!,Temp2!,TIFF!
REAL: Temp3!
INTEGER: Print_temp%,ROW%,P%
REAL: ACTUALTEMP,N

PROCEDURE: TIMEDELAY()
   REAL ARG: DELAY!/OPT=5!
END PROCEDURE

PROCEDURE: CLEARSCREEN()
   INTEGER ARG: ROW%
END PROCEDURE


PROCEDURE: TIMEDELAY
   REAL: Tyme!
   STRING: OVERHEAT$[?],Data_string$[?]
   REAL: TEMP_PEAK!,PEAK_TEMP!
      22 REM *   integer arg:delay%/opt = 5
      32 REM *   real:tyme!
     100 Tyme! = TIMER
     110 IF TIMER < Tyme! + DELAY! THEN GOTO 110
     120 EXIT
END PROCEDURE

PROCEDURE: CLEARSCREEN
   INTEGER: N%
     100 FOR N% = ROW% TO 24
     110   SET CURSOR N%,0
     120   PRINT SPC(80)
     130 NEXT N%
END PROCEDURE

'MAIN Program:


  200 REM
  201 REM ***********************************************
*************
  202 REM *
         *
  203 REM *   This section prints the entrance message to
RUNIT.  It   *
  204 REM *   also initializes the devices on the IEEE bus
.        *
  205 REM *
         *
  206 REM ***********************************************
*************
  207 REM
```

```
  210 CLEAR
  220 ON ERROR GOSUB 42000
  230 CLS : STATUSLINE OFF : SCREEN 0,0,0 : COLOR 2,0 : LO
CATE 12,23
  240 PRINT "Hit any key to begin Experiments"
  250 COLOR 3,0:LOCATE 14,10
  260 PRINT "All instruments must be on and the proper con
nections be made"
  270 TIMEDELAY (2)
  280 A$ = INKEY$: IF A$ = "" THEN GOTO 250
  290 PARAM$ = "INIT/1/&H310/P/":GOSUB 50000
  300 PARAM$ = "SDR/5,7,8,12,13,16,17/":GOSUB 50000
  310 REM
  311 REM ********************************************
*************
  312 REM *
          *
  313 REM *   This section loads the data from the files s
tored by     *
  314 REM *   the program PARAMETER.
          *
  315 REM *
          *
  316 REM ********************************************
*************
  317 REM
  320 CLS:COLOR 2,0,0:LOCATE 12,30
  330 PRINT "Now LOADING data"
  340 DRIVE$ = "C:"
  350 DIR$ = "\"
  360 OPEN " \DATA\EXPT" FOR INPUT AS #1
  370 INPUT #1,File$
  380 FOR M% = 1 TO 30
  390   INPUT #1,Expt%(M%)
  400 NEXT M%
  410 CLOSE #1
  420 OPEN " \DATA\PARAM" FOR INPUT AS #1
  430 FOR M% = 1 TO 6
  440   INPUT #1,Parameter!(25,M%)
  450 NEXT M%
  460 RESTORE,60010
  470 READ Maxloop%
  480 FOR M% = 1 TO Maxloop%
  490   READ Group%(M%)
  500 NEXT M%
  505 RESTORE,60410
  507 READ Maxloop%
  510 FOR N% = 1 TO Group%(Expt%(25))
  520   FOR M% = 1 TO Maxloop%
  530     IF Expt%(N%) = 0 THEN EXIT 1 LEVELS
  540     INPUT #1,Parameter!(N%,M%)
```

```
550   NEXT M%
560 NEXT N%
570 CLOSE #1
575 ACTUALTEMP=Parameter!(25,4)
600 REM
601 REM *************************************************
*************
602 REM *
        *
603 REM *   This section finds which type of temperature
 run was     *
604 REM *   requested, then stores in in the string XTEM
P$          *
605 REM *
        *
606 REM *************************************************
*************
607 REM
610 RESTORE,60110
620 FOR M% = 1 TO 30
630   READ Xtemp$
640   IF M% = Expt%(26) THEN EXIT
650 NEXT M%
700 REM
704 P% = 0 : Print_temp% = 1
705 RESTORE,62000
710 DO
720   IF Expt%(25) <> 3 THEN EXIT TO,1000
730   CLS : COLOR 4,0,0 : SET CURSOR 0,35
740   PRINT "MOBILITY EXPERIMENT"
750 END DO
765 COLOR 3,0,0 : SET CURSOR 7,6 : PRINT "Connect the ca
bles as follows"
766 COLOR 2,0,0
770 FOR N% = 1 TO 5
780   READ Name$
800   SET CURSOR 7 + N%*2,10
810   PRINT Name$
820 NEXT N%
830 COLOR 7,0,0 : SET CURSOR 22,6
840 PRINT "Hit any key to continue"
850 A$ = INKEY$ : IF A$ = "" THEN GOTO 850
860 P% = P% + 1
870 GOSUB 15000
871 IF P% < 3 THEN GOTO 710
872 CLEARSCREEN(2)
873 COLOR 2,0,0 : SET CURSOR 13,6
874 PRINT "Hit any key to begin the experiment run"
875 A$ = INKEY$ : IF A$ = "" THEN GOTO 875
880 Print_temp% = 2:GOTO 1000
1000 REM
```

```
 1001 REM ***************************************************
*************
 1002 REM *
          *
 1003 REM *   This section print up the proper screen for
each type    *
 1004 REM *   of temperature run.  Prints the time if a ti
me run,      *
 1005 REM *   and goes to the proper subroutine for the ex
periments    *
 1006 REM *   chosen.  It also exits to finish routines if
 either      *
 1007 REM *   the final temperature or final time has been
 reached.    *
 1008 REM *
          *
 1009 REM ***************************************************
*************
 1010 REM
 1020 Mintemp! = Parameter!(25,1)
 1030 Maxtemp! = Parameter!(25,2)
 1040 COLOR 5,0:CLS
 1050 IF Expt%(26) = 3 THEN PRINT SPC(60);"Room Temperatur
e"
 1060 COLOR 4,0:IF Expt%(26) = 3 THEN PRINT SPC(63);Parame
ter!(25,4);"K"
 1070 COLOR 3,0:LOCATE 12,28:PRINT "The Current Program is
:"
 1080 IF Expt%(26) = 3 THEN GOTO 1100
 1090 GOSUB 9000
 1100 IF Expt%(26) <> 1 THEN GOTO 1150
 1110 Timerun! = 0.0
 1120 TIME$ = "00:00:00"
 1130 LOCATE 1,8:COLOR 5,0:PRINT "Time"
 1140 LOCATE 2,6:COLOR 4,0:PRINT TIME$
 1150 COLOR 2,0
 1160 ON Expt%(25) GOSUB 2000,2500,3010,4010
 1170 IF Expt%(26) = 1 THEN GOTO 1220
 1180 Parameter!(25,4) = Parameter!(25,4) + Parameter!(25,
3)
 1190 IF Parameter!(25,4) > Maxtemp! THEN GOTO 40000
 1210 GOTO 1040
 1220 IF TIMER > Parameter!(25,5) THEN GOTO 40000
 1230 Timerun! = Timerun! + Parameter!(25,6)
 1240 TIMEDELAY (.5)
 1250 LOCATE 14,35:PRINT "TIME SET"
 1260 LOCATE 2,6:COLOR 4,0:PRINT TIME$
 1270 IF TIMER <= Timerun! THEN GOTO 1250
 1280 GOTO 1150
 1999 STOP
 2000 REM
```

```
 2001 REM *********************************************
*************
 2002 REM *
         *
 2003 REM *   This subroutine branches to the proper exper
iments        *
 2004 REM *   chosen in group I.  It also checks after eac
h             *
 2005 REM *   experiment to see if the COLD HEAD has overh
eated         *
 2006 REM *   (except in the case of a Room Temperature ru
n).           *
 2007 REM *
         *
 2008 REM *********************************************
*************
 2009 REM
 2010 L% = 1
 2020 ON Expt%(L%) GOSUB 10000,10000,11000,12000
 2030 LOCATE 14,1:PRINT SPC(79)
 2040 L% = L% + 1
 2050 IF Expt%(26) <> 3 THEN GOSUB 9500
 2060 IF L% <= Group%(Expt%(25)) THEN GOTO 2020
 2070 RETURN
 2499 STOP
 2500 REM
 2501 REM *********************************************
*************
 2502 REM *
         *
 2503 REM *   This subroutine branches to the proper exper
iments        *
 2504 REM *   chosen in group II.  It also checks after ea
ch            *
 2505 REM *   experiment to see if the COLD HEAD has overh
eated         *
 2506 REM *   (except in the case of a Room Temperature ru
n).           *
 2507 REM *
         *
 2508 REM *********************************************
*************
 2509 REM
 2510 L% = 1
 2520 ON Expt%(L%) GOSUB 13000,14000
 2530 LOCATE 14,1:PRINT SPC(79)
 2540 IF Expt%(26) <> 3 THEN GOSUB 9500
 2550 L% = L% + 1
 2560 IF L% <= Group%(Expt%(25)) THEN GOTO 2520
 2570 RETURN
 3000 REM
```

```
 3001 REM ************************************************
*************
 3002 REM *
          *
 3003 REM *   This subroutine branches to the proper exper
iments        *
 3004 REM *   chosen in group III.  It also checks after e
ach           *
 3005 REM *   experiment to see if the COLD HEAD has overh
eated         *
 3006 REM *   (except in the case of a Room Temperature ru
n).           *
 3007 REM *
          *
 3008 REM ************************************************
*************
 3009 REM
 3010 GOSUB 15000
 3020 IF Expt%(26) <> 3 THEN GOSUB 9500
 3030 RETURN
 4000 REM
 4001 REM ************************************************
*************
 4002 REM *
          *
 4003 REM *   This subroutine branches to the proper exper
iments        *
 4004 REM *   chosen in group IV.  It also checks after ea
ch            *
 4005 REM *   experiment to see if the COLD HEAD has overh
eated         *
 4006 REM *   (except in the case of a Room Temperature ru
n).           *
 4007 REM *
          *
 4008 REM ************************************************
*************
 4009 REM
 4010 GOSUB 16000
 4020 IF Expt%(26) <> 3 THEN GOSUB 9500
 4030 RETURN
 8999 STOP
 9000 REM
 9001 REM ************************************************
*************
 9002 REM *
          *
 9003 REM *   This sets the Cryostat to the correct temper
ature and     *
 9004 REM *   then lets the temperature to stabilize to +/
-1   degree *
```

```
 9005 REM *
          *
 9007 REM ***********************************************
*************
 9008 REM
 9010 COLOR 2,0:LOCATE 14,24
 9020 PRINT "Setting the Cryostat Temperature"
 9030 LOCATE 1,30:PRINT SPC(49)
 9040 COLOR 5,0:LOCATE 1,60:PRINT "Sample Setpoint"
 9050 COLOR 4,0:LOCATE 2,63:PRINT Parameter!(25,4);"K"
 9060 Test_Passed% = 0:Cryostat% = 0
 9065 IF Parameter!(25,2) > 290 THEN Offset! = 0 ELSE Offs
et! = 0
 9070 Temp_wanted! = Parameter!(25,4) + Offset!
 9075 IF Temp_wanted!=>290 THEN LOCATE 22,10:COLOR 30,0:PR
INT "Warning: ":LOCATE 22,19:COLOR 7,0:PRINT"the vacuum sy
stem should be pulling on the coldhead!!"
 9080 IF Temp_wanted! < 15 THEN Heater$ = "7"
 9090 IF Temp_wanted! => 15 AND Temp_wanted! < 20 THEN Hea
ter$ = "8" ELSE Heater$ = "9"
 9100 IF Temp_wanted! > 290 THEN Int_Gain$="00100000400" E
LSE Int_Gain$="00100000500"
 9101 IF Temp_wanted!<100 THEN Int_Gain$="00150000600"
 9102 IF Temp_wanted!< 50 THEN Int_Gain$="00200000700"
 9110 Delta_temp!=1
 9120 Sensor$ = "33"
 9130 DATA_STRING$ = "P"
 9140 PARAM$ = "WR.STR/7//EOS/":GOSUB 50000
 9150 Temp_wanted$ = MID$(STR$(Temp_wanted! + 1000),3)
 9170 DATA_STRING$ = Temp_wanted$ + Int_Gain$ + Sensor$ +
Heater$ + "505"
 9180 PARAM$ = "WR.STR/7//EOS/":GOSUB 50000
 9190 PARAM$ = "RD.STR/7//EOS/":GOSUB 50000
 9200 PARAM$ = "RD.STR/7//EOS/":GOSUB 50000
 9205 TIMEDELAY(2)
 9210 DO 2 TIMES
 9220   PARAM$ = "RD.STR/7//EOS/":GOSUB 50000
 9230   Temp! = VAL(MID$(DATA_STRING$,2))
 9240   IF Temp!=> Temp_wanted! - Delta_temp! AND Temp! <=
Temp_wanted! + Delta_temp! THEN EXIT TO,9300
 9250   TIMEDELAY (1)
 9260 REPEAT
 9270 Cryostat% = 0
 9275 Test_Passed% = 0
 9280 IF Temp!< Temp_wanted!-2 OR Temp!>Temp_wanted!+2 THE
N GOSUB 9500
 9290 GOTO 9130
 9299 STOP
 9300 Test_Passed% = Test_Passed% + 1
 9310 IF Test_Passed% > 1 THEN GOTO 9400
 9320 TIMEDELAY(35)
```

```
 9330 GOTO 9130
 9400 LOCATE 1,30:PRINT SPC(79)
 9410 COLOR 5,0:LOCATE 1,59:PRINT "Sample Temperature"
 9415 COLOR 4,0:LOCATE 2,65:PRINT Parameter!(25,4)
 9417 ACTUALTEMP=Temp!
 9420 COLOR 2,0
 9430 Cryostat% = 1
 9440 RETURN
 9499 STOP
 9500 REM
 9501 REM ************************************************
*************
 9502 REM *
            *
 9503 REM *   This subroutine checks the Cold Head to make
 sure its    *
 9504 REM *   not overheating.
            *
 9505 REM *
            *
 9506 REM ************************************************
*************
 9507 REM
 9510 IF Cryostat% = 0 THEN GOTO 9540
 9520 LOCATE 14,1:PRINT SPC(79)
 9530 COLOR 2,0:LOCATE 14,30:PRINT "Checking Cold Finger"
 9540 DATA_STRING$ = "P"
 9550 PARAM$ = "WR.STR/7//EOS/":GOSUB 50000
 9555 Sensor1$ = "31"
 9560 DATA_STRING$ = Temp_wanted$ + Int_Gain$ + Sensor1$ +
 Heater$ + "505"
 9570 PARAM$ = "WR.STR/7//EOS/":GOSUB 50000
 9580 DATA_STRING$ = "R"
 9590 PARAM$ = "WR.STR/7//EOS/":GOSUB 50000
 9595 TIMEDELAY(3)
 9600 DO 5 TIMES
 9610    PARAM$ = "RD.STR/7//EOS/":GOSUB 50000
 9620    Temp! = VAL(MID$(DATA_STRING$,2))
 9630    IF Temp! > 325 AND Temp! < 900 THEN GOTO 9700
 9640 REPEAT
 9650 IF Cryostat% = 0 THEN GOTO 9670
 9660 LOCATE 14,1:PRINT SPC(79):LOCATE 14,1
 9670 RETURN
 9699 STOP
 9700 REM
 9701 REM ************************************************
*************
 9702 REM *
            *
 9703 REM *   This subroutine is used if the Cold Finger o
verheats.    *
```

```
 9704 REM *   It writes death parameters to the Cryostat a
nd gives     *
 9705 REM *   a warning message.
             *
 9706 REM *
             *
 9707 REM ***********************************************
*************
 9708 REM
 9710 DATA_STRING$ = "P"
 9720 PARAM$ = "WR.STR/7//EOS/":GOSUB 50000
 9730 DATA_STRING$ = "00210000000001117501"
 9740 PARAM$ = "WR.STR/7//EOS/":GOSUB 50000
 9750 DATA_STRING$ = "R"
 9760 PARAM$ = "WR.STR/7//EOS/":GOSUB 50000
 9770 PARAM$ = "RD.STR/7//EOS/":GOSUB 50000
 9780 CLS:COLOR 2,0:LOCATE 11,21
 9790 PRINT "The ";
 9800 COLOR 19,0:PRINT "COLD FINGER";
 9810 COLOR 2,0
 9820 PRINT " has ";
 9830 COLOR 20,0:PRINT "Overheated"
 9840 LOCATE 14,16:COLOR 7,0
 9850 PRINT "PLEASE CHECK FOR DAMAGE AND CONTACT ERIC COLE
"
 9860 LOCATE 16,12:COLOR 1,0:PRINT "ALSO TURN THE HEATER S
WITCH ABOVE THE CONTROLLER OFF"
 9870 CLEAR
 9880 GOTO 9880
 9999 STOP
10000 REM
10001 REM ***********************************************
*************
10002 REM *
             *
10003 REM *   This subroutine run the experiment [either C
-t (1) or    *
10004 REM *   G-t (2)] specified by 1%.  It stores the dat
a in the     *
10005 REM *   file C:\DATA\##\?-t where ? is C or G respec
tively *
10006 REM *
             *
10007 REM ***********************************************
*************
10008 REM
10010 IF Expt%(26) = 1 OR Expt%(26) = 3 THEN GOTO 10030
10020 IF Parameter!(L%,4) <> Parameter!(25,4) GOTO 10500
10030 IF L% = 1 THEN GOTO 10080
10040 Function$ = "FN6"
10050 Name$ = "\G-t"
```

```
10060 Moniker$ = "CONDUCTANCE"
10070 GOTO 10110
10080 Function$ = "FN5"
10090 Name$ = "\C-t"
10100 Moniker$ = "CAPACITANCE"
10110 OPEN "C:\DATA\" + File$ + Name$ FOR APPEND AS #1
10120 LOCATE 14,20:PRINT "Taking ";Moniker$;" vs. TIME Mea
surements"
10130 Pulsevolt! = Parameter!(L%,21)
10140 DATA_STRING$ = "M4CT0T1W1HIL" + STR$(Pulsevolt!) + "
VLOL" + STR$(Parameter!(L%,24)) + "VDO"
10150 PARAM$ = "WR.STR/13//EOS/":GOSUB 50000
10160 PARAM$ = "RD.STR/13//EOS/":GOSUB 50000
10170 Bias! = Parameter!(L%,17)
10180 Time! = Parameter!(L%,14)
10190 IF Time! <= 0.01 THEN GOSUB 10580:GOTO 10220
10200 IF Time! < 10.1 * Parameter!(L%,13) AND Time! < .1 T
HEN GOSUB 10580:GOTO 10220
10210 GOSUB 10650
10220 N% = 1
10230 PRINT #1,Xtemp$;Parameter!(25,4)";"ACTUALTEMP
10240 PRINT #1,"TIME = ";Timerun!;"  (";TIMER;" )"
10250 PRINT #1,"BIAS = ";Bias!
10260 PRINT #1,"HIGH PULSE = ";Pulsevolt!
10270 PRINT #1,"HOLD TYME = ";Time!
10280 PARAM$ = "RD.STR/17//EOS/":GOSUB 50000
10290 NCM% = INSTR(DATA_STRING$,"M") + 1
10300 Cdata! = VAL(MID$(DATA_STRING$,NCM%))
10310 T% = INSTR(DATA_STRING$,"T") + 1
10320 Xdata! = VAL(MID$(DATA_STRING$,T%))
10330 PRINT #1,Cdata!;",";Xdata!
10340 N% = N% + 1
10350 IF N% <= Parameter!(L%,20) THEN GOTO 10280
10360 PRINT #1,"END"
10370 IF Parameter!(L%,15) = 0 OR Parameter!(L%,16) = 0 TH
EN GOTO 10400
10380 Time! = Time! + Parameter!(L%,16)
10390 IF Time! <= Parameter!(L%,15) THEN GOTO 10190
10400 IF Parameter!(L%,18) = 0 OR Parameter!(L%,19) = 0 TH
EN GOTO 10430
10410 Bias! = Bias! + Parameter!(L%,19)
10420 IF Bias! <= Parameter!(L%,18) THEN GOTO 10180
10430 IF Parameter!(L%,23) = 0 THEN GOTO 10460
10440 Pulsevolt! = Pulsevolt! + Parameter!(L%,23)
10450 IF Pulsevolt! <= Parameter!(L%,22) THEN GOTO 10140
10460 Parameter!(L%,4) = Parameter!(L%,4) + Parameter!(L%,
3)
10470 CLOSE #1
10480 DATA_STRING$ = "BLO"
10490 PARAM$ = "WR.STR/17//EOS/":GOSUB 50000
10500 REM
```

```
10540 IF Parameter!(L%,4) < Parameter!(25,4) THEN Paramete
r!(L%,4) = Parameter!(L%,3) + Parameter!(L%,4)
10550 IF Parameter!(L%,4) > Parameter!(L%,2) THEN Paramete
r!(L%,4) = 600
10560 PARAM$ = "SDC/13,17/":GOSUB 50000
10570 RETURN
10579 STOP
10580 REM
10581 REM ************************************************
*************
10582 REM *
           *
10583 REM *   This subroutine is used if the step delay ti
me and     *
10584 REM *   hold times does require the meter to be put
into       *
10585 REM *   block mode.
           *
10586 REM *
           *
10587 REM ************************************************
*************
10588 REM
10590 DATA_STRING$ = Function$ + "CN13TR3LE2SA1PC" + STR$(
Bias!) + ";PN" + STR$(Parameter!(L%,20)) + ";PH" + STR$(Ti
me!) + ";PT" + STR$(Parameter!(L%,13)) + ";BL1;SW1"
10600 PARAM$ = "WR.STR/17//EOS/":GOSUB 50000
10610 TIMEDELAY(10)
10620 DATA_STRING$ = "BD"
10630 PARAM$ = "WR.STR/17//EOS/":GOSUB 50000
10640 RETURN
10649 STOP
10650 REM
10651 REM ************************************************
*************
10652 REM *
           *
10653 REM *   This subroutine is used if the step delay ti
me and     *
10654 REM *   hold times do not require the meter to be pu
t into     *
10655 REM *   block mode.
           *
10656 REM *
           *
10657 REM ************************************************
*************
10658 REM
10660 DATA_STRING$ = Function$ + "CN13TR3LE2SA1PC" + STR$(
Bias!) + ";PN" + STR$(Parameter!(L%,20)) + ";PH" + STR$(Ti
me!) + ";PT" + STR$(Parameter!(L%,13)) + ";SW1"
```

```
10670 PARAM$ = "WR.STR/17//EOS/":GOSUB 50000
10680 RETURN
10999 STOP
11000 REM
11001 REM ***********************************************
*************
11002 REM *
           *
11003 REM *   This subroutine runs the C-G-V experiment an
d stores    *
11004 REM *   the data in the file "\DATA\##\CGV".
           *
11005 REM *
           *
11006 REM ***********************************************
*************
11007 REM
11010 IF Expt%(26) = 1 OR Expt%(26) = 3 THEN GOTO 11030
11020 IF Parameter!(3,4) <> Parameter!(25,4) GOTO 11310
11030 OPEN "C:\DATA\" + File$ + "\CGV" FOR APPEND AS #1
11040 PRINT #1,Xtemp$;Parameter!(25,4)";"ACTUALTEMP
11050 PRINT #1,"TIME = ";Timerun!;"  (";TIMER;")"
11060 LOCATE 14,10:PRINT "taking CAPACITANCE AND CONDUCTAN
CE vs. VOLTAGE measurements"
11070 N% = 6
11080 M% = 6
11090 N% = N% + 1
11100 M% = M% + 2
11110 IF Parameter!(3,M%+1) = 0 THEN GOTO 11270
11120 DATA_STRING$ = "FN1CN10IB2LE2TR3PS" + STR$(Parameter
!(3,N%)) + ";PP" + STR$(Parameter!(3,M%)) + ";PE" + STR$(P
arameter!(3,M%+1)) + ";PL" + STR$(Parameter!(3,14)) + ";PD
" + STR$(Parameter!(3,13)) + ";SW1"
11130 PARAM$ = "WR.STR/17//EOS/":GOSUB 50000
11140 L! = Parameter!(3,N%)
11150 PARAM$ = "RD.STR/17//EOS/":GOSUB 50000
11160 NCM% = INSTR(DATA_STRING$,"M") + 1
11170 Cdata! = VAL(MID$(DATA_STRING$,NCM%))
11180 NGM% = INSTR(DATA_STRING$,"GM") + 2
11190 Gdata! = VAL(MID$(DATA_STRING$,NGM%))
11200 V% = INSTR(DATA_STRING$,"V") + 1
11210 Xdata! = VAL(MID$(DATA_STRING$,V%))
11220 PRINT #1,Cdata!;",";Gdata!;",";Xdata!
11240 IF L! < Parameter!(3,M%) THEN L! = L! + Parameter!(3
,M%+1): GOTO 11150
11250 DATA_STRING$ = "SW0"
11260 PARAM$ = "WR.STR/17//EOS/":GOSUB 50000
11270 IF N% = 7 THEN GOTO 11090
11280 Parameter!(3,4) = Parameter!(3,4) + Parameter!(3,3)
11290 PRINT #1,"END"
11300 CLOSE #1
```

```
11310 REM
11350 IF Parameter!(3,4) < Parameter!(25,4) THEN Parameter
!(3,4) = Parameter!(3,4) + Parameter!(3,3)
11360 IF Parameter!(3,4) > Parameter!(3,2) THEN Parameter!
(3,4) = 600
11370 PARAM$ = "SDC/17/":GOSUB 50000
11380 RETURN
11999 STOP
12000 REM
12001 REM ***********************************************
*************
12002 REM *
           *
12003 REM *   This section takes the C-G readings and puts
 them in     *
12004 REM *   file \DATA\##\CG"
           *
12005 REM *
           *
12006 REM ***********************************************
*************
12007 REM
12010 IF Expt%(26) = 1 OR Expt%(26) = 3 THEN GOTO 12030
12020 IF Parameter!(4,4) <> Parameter!(25,4) THEN GOTO 122
50
12030 OPEN "C:\DATA\" + File$ + "\CG" FOR APPEND AS #1
12040 Bias! = Parameter!(4,17)
12050 LOCATE 14,15:PRINT "taking CAPACITANCE AND CONDUCTAN
CE measurements"
12060 DATA_STRING$ = "FN1CN10IB1RA1MS3SL2LE2TR3PV" + STR$(
Bias!)
12070 PARAM$ = "WR.STR/17//EOS/":GOSUB 50000
12072 DATA_STRING$ ="VO1"
12074 PARAM$ = "WR.STR/17//EOS/":GOSUB 50000
12080 PRINT #1,Xtemp$;Parameter!(25,4)";"ACTUALTEMP
12090 PRINT #1,"TIME = ";Timerun!;"  (";TIMER;" )"
12100 PRINT #1,"BIAS = ";Bias!
12105 TIMEDELAY(5)
12110 DATA_STRING$ ="EX"
12120 PARAM$ = "WR.STR/17//EOS/":GOSUB 50000
12130 PARAM$ = "RD.STR/17//EOS/":GOSUB 50000
12140 NCM% = INSTR(DATA_STRING$,"M") + 1
12150 Cdata! = VAL(MID$(DATA_STRING$,NCM%))
12160 NGM% = INSTR(DATA_STRING$,"GM") + 2
12170 Gdata! = VAL(MID$(DATA_STRING$,NGM%))
12180 PRINT #1,Cdata!","Gdata!","ACTUALTEMP
12185 DATA_STRING$ ="VOO"
12187 PARAM$ = "WR.STR/17//EOS/":GOSUB 50000
12190 IF Parameter!(4,19) = 0 THEN GOTO 12220
12200 Bias! = Bias! + Parameter!(4,19)
12210 IF Bias! <= Parameter!(4,18) THEN GOTO 12060
```

```
12220 Parameter!(4,4) = Parameter!(4,4) + Parameter!(4,3)
12225 TIMEDELAY(10)
12230 CLOSE #1
12240 PARAM$ = "SDC/17/":GOSUB 50000
12250 REM
12290 IF Parameter!(4,4) < Parameter!(25,4) THEN Parameter
!(4,4) = Parameter!(4,4) + Parameter!(4,3)
12300 IF Parameter!(4,4) > Parameter!(4,2) THEN Parameter!
(4,4) = 600
12310 PARAM$ = "SDC/17/":GOSUB 50000
12320 RETURN
12999 STOP
13000 REM
13001 REM ***********************************************
*************
13002 REM *
          *
13003 REM *   This is the subroutine for actually running
the I-V      *
13004 REM *   experiment.  It stores the data in the file
          *
13005 REM *   \DATA\##\IV
          *
13006 REM *
          *
13007 REM ***********************************************
*************
13008 REM
13010 IF Expt%(26) = 1 OR Expt%(26) = 3 THEN GOTO 13030
13020 IF Parameter!(1,4) <> Parameter!(25,4) THEN GOTO 132
90
13030 OPEN "C:\DATA\" + File$ + "\IV" FOR APPEND AS #1
13040 LOCATE 14,20:PRINT "taking CURRENT vs. VOLTAGE measu
rements"
13050 PRINT #1,Xtemp$;Parameter!(25,4)";"ACTUALTEMP
13060 PRINT #1,"TIME = ";Timerun!;"  (";TIMER;" )"
13070 M% = 7
13080 N% = 8
13085 X$ = STR$(VAL(STR$(Parameter!(1,N% + 1))))
13090 DATA_STRING$ = "F2I2L3PS" + STR$(Parameter!(1,M%)) +
 ";PT" + STR$(Parameter!(1,N%)) + ";PE" + X$ + ";PH" + STR
$(Parameter!(1,14)) + ";PD" + STR$(Parameter!(1,13)) + ";W
1"
13100 TIMEDELAY(1)
13110 PARAM$ = "WR.STR/5//EOS/":GOSUB 50000
13120 B% = 0
13130 PARAM$ = "RD.STR/5//EOS/":GOSUB 50000
13140 NCM% = INSTR(DATA_STRING$,"I") + 1
13150 Idata! = VAL(MID$(DATA_STRING$,NCM%))
13160 V% = INSTR(DATA_STRING$,"A") + 1
13170 Xdata! = VAL(MID$(DATA_STRING$,V%))
```

```
13180 PRINT #1,Idata!",",Xdata!
13190 B% = INSTR(DATA_STRING$,"L")
13200 IF B% = 0 THEN GOTO 13130
13210 TIMEDELAY(2)
13215 PARAM$ = "RD.STR/5//EOS/":GOSUB 50000
13220 IF Parameter!(1,11) = 0 THEN GOTO 13260
13230 M% = M% + 1
13240 N% = N% + 2
13250 IF M% = 8 THEN GOTO 13085
13260 Parameter!(1,4) = Parameter!(1,4) + Parameter!(1,3)
13270 PRINT #1,"END"
13280 CLOSE #1
13290 REM
13330 IF Parameter!(1,4) < Parameter!(25,4) THEN Parameter
!(1,4) = Parameter!(1,4) + Parameter!(1,3)
13340 IF Parameter!(1,4) > Parameter!(1,2) THEN Parameter!
(1,4) = 600
13350 PARAM$ = "SDC/5/":GOSUB 50000
13360 RETURN
13999 STOP
14000 REM
14001 REM ***********************************************
*************
14002 REM *
          *
14003 REM *   This actually runs the Capacitance vs. Volta
ge        *
14004 REM *   experiment.
          *
14005 REM *
          *
14006 REM ***********************************************
*************
14007 REM
14010 IF Expt%(26) = 1 OR Expt%(26) = 3 THEN GOTO 14030
14020 IF Parameter!(2,4) <> Parameter!(25,4) THEN GOTO 142
10
14030 OPEN "C:\DATA\" + File$ + "\CV" FOR APPEND AS #1
14040 LOCATE 14,18:PRINT "taking CAPACITANCE vs. VOLTAGE m
easurements"
14050 B% = 0
14060 PRINT #1,Xtemp$;Parameter!(25,4)";"ACTUALTEMP
14070 PRINT #1,"TIME = ";Timerun!;"  (";TIMER;" )"
14080 DATA_STRING$ = "F3I2L3PS" + STR$(Parameter!(2,7)-Par
ameter!(2,9)) +";PT" + STR$(Parameter!(2,8)+Parameter!(2,9
)) + ";PE" + STR$(Parameter!(2,9)) + ";PV" + STR$(Paramete
r!(2,12)) + ";PH" + STR$(Parameter!(2,14)) + ";W1"
14090 PARAM$ = "WR.STR/5//EOS/":GOSUB 50000
14100 PARAM$ = "RD.STR/5//EOS/":GOSUB 50000
14110 NCM% = INSTR(DATA_STRING$,"C") + 1
14120 Cdata! = VAL(MID$(DATA_STRING$,NCM%))
```

```
14130 V% = INSTR(DATA_STRING$,"A") + 1
14140 Xdata! = VAL(MID$(DATA_STRING$,V%))
14150 PRINT #1,Cdata!",",Xdata!
14160 B% = INSTR(DATA_STRING$,"L")
14170 IF B% = 0 THEN GOTO 14100
14180 Parameter!(2,4) = Parameter!(2,4) + Parameter!(2,3)
14190 PRINT #1,"END"
14200 CLOSE #1
14210 REM
14250 IF Parameter!(2,4) < Parameter!(25,4) THEN Parameter
!(2,4) = Parameter!(2,4) + Parameter!(2,3)
14260 IF Parameter!(2,4) > Parameter!(2,2) THEN Parameter!
(2,4) = 600
14270 PARAM$ = "SDC/5/":GOSUB 50000
14280 RETURN
14999 STOP
15000 REM
15001 REM ************************************************
*************
15002 REM *
         *
15003 REM *   This subroutine runs the Van der Pauw/Mobili
ty        *
15004 REM *   experiment and stores it in the file MOB.
         *
15005 REM *
         *
15006 REM ************************************************
*************
15007 REM
15009 IF Print_temp% = 1 THEN CLEARSCREEN(1)
15010 COLOR 2,0:LOCATE 14,20
15020 PRINT "taking Van der Pauw/Mobility Measurements"
15030 OPEN "\DATA\" + File$ + "\MOB" FOR APPEND AS #1
15032 DO
15035   IF Print_temp% <> 2 THEN EXIT 1 LEVELS
15040   PRINT #1,Xtemp$;Parameter!(25,4)";"ACTUALTEMP
15050   PRINT #1,"TIME = ";Timerun!;"  (";TIMER;" )"
15055 END DO
15060 DATA_STRING$ = "FN1CN10IB2TR3PS-10;PP10;PE.01;PL2;PD
2"
15070 PARAM$ = "WR.STR/17//EOS/":GOSUB 50000
15078 DATA_STRING$ = "L3;"
15079 PARAM$ = "WR.STR/5//EOS/":GOSUB 50000
15089 DATA_STRING$ = "R0" + CHR$(13) + "X"
15090 PARAM$ = "WR.STR/16//EOS/":GOSUB 50000
15092 TIMEDELAY(15)
15095 DATA_STRING$ = "Z1" + CHR$(13) + "X"
15096 PARAM$ = "WR.STR/16//EOS/":GOSUB 50000
15100 FOR M% = 1 TO Print_temp%
15110   DATA_STRING$ = "F2RA1I2J0A3B2L3PS0;PT10;PE0.01;PH1
```

```
;PD.1;W2"
15120    PARAM$ = "WR.STR/5//EOS/":GOSUB 50000
15130    DO 1000 TIMES
15140      DATA_STRING$ = "W6"
15150      PARAM$ = "WR.STR/5//EOS/":GOSUB 50000
15160      TIMEDELAY (2)
15170      PARAM$ = "RD.STR/5//EOS/":GOSUB 50000
15180      CURRENT! = VAL(MID$(DATA_STRING$,4))
15190      VOLT! = VAL(MID$(DATA_STRING$,16))
15200      IF CURRENT! => Parameter!(1,17) THEN EXIT
15210    REPEAT
15211    DATA_STRING$ = "S1" + CHR$(13) + "X"
15212    PARAM$ = "WR.STR/16//EOS/":GOSUB 50000
15235    TIMEDELAY(10)
15237    PARAM$ = "RD.STR/16//EOS/":GOSUB 50000
15240    VOLT2! = VAL(MID$(DATA_STRING$,5))
15250    PRINT #1,M%;",";CURRENT!;",";VOLT!;","VOLT2!
15260    IF M% = 1 AND Print_temp%=2 THEN DATA_STRING$ = "S
W1" ELSE DATA_STRING$ = "SW0"
15270    PARAM$ = "WR.STR/17//EOS/":GOSUB 50000
15280    DATA_STRING$ = "W7"
15290    PARAM$ = "WR.STR/5//EOS/":GOSUB 50000
15300    IF M% = 1 AND Print_temp% = 2 THEN TIMEDELAY (20):
DATA_STRING$ = "Z1" + CHR$(13) + "X":PARAM$ = "WR.STR/16//
EOS/":GOSUB 50000:TIMEDELAY(15)
15310 NEXT M%
15320 PRINT #1,"END"
15330 CLOSE #1
15340 PARAM$ = "SDC/5,16,17/":GOSUB 50000
15350 RETURN
16000 REM
16001 REM ***********************************************
*************
16002 REM *
        *
16003 REM *   This section takes the RES readings and puts
 them in    *
16004 REM *   file \DATA\##\RES"
         *
16005 REM *
        *
16006 REM ***********************************************
*************
16007 REM
16030 OPEN "C:\DATA\" + File$ + "\RES" FOR APPEND AS #1
16040 Bias! = Parameter!(1,17)
16050 LOCATE 14,17:PRINT "taking 4-POINT RESISTIVITY measu
rements"
16080 PRINT #1,Xtemp$;Parameter!(25,4)";"ACTUALTEMP
16090 PRINT #1,"TIME = ";Timerun!;"  (";TIMER;" )"
16100 PRINT #1,"BIAS = ";Bias!
```

```
16110 DATA_STRING$ = "RO" + CHR$(13) + "X"
16120 PARAM$ = "WR.STR/16//EOS/":GOSUB 50000
16130 TIMEDELAY(5)
16140 PARAM$ = "RD.STR/16//EOS/":GOSUB 50000
16150 VOLT2! = VAL(MID$(DATA_STRING$,5))
16180 PRINT #1,VOLT2!","VOLT2!","ACTUALTEMP
16225 TIMEDELAY(1)
16230 CLOSE #1
16240 PARAM$ = "SDC/16/":GOSUB 50000
16320 RETURN
16999 STOP
39999 STOP
40000 REM
40001 REM ************************************************
*************
40002 REM *
          *
40003 REM *   This stops the program after a successful ru
n.        *
40004 REM *
          *
40005 REM ************************************************
*************
40006 REM
40010 IF Expt%(26) <> 3 THEN GOSUB 41500
40020 IF Expt%(25) = 1 THEN RESTORE,60210
40030 IF Expt%(25) = 2 THEN RESTORE,60220
40040 IF Expt%(25) = 3 THEN RESTORE,60230
40050 IF Expt%(25) = 4 THEN RESTORE,60240
40060 IF Expt%(25) = 5 THEN RESTORE,60250
40070 IF Expt%(25) = 6 THEN RESTORE,60260
40080 IF Expt%(25) = 7 THEN RESTORE,60270
40090 IF Expt%(25) = 8 THEN RESTORE,60280
40100 IF Expt%(25) = 9 THEN RESTORE,60290
40110 IF Expt%(25) = 10 THEN RESTORE,60300
40200 FOR M% = 1 TO Group%(Expt%(25))
40210   READ Name$
40220   IF Expt%(M%) = 0 THEN GOTO 40260
40230   OPEN "\DATA\" + File$ + "\" + Name$ FOR APPEND AS
#1
40240   PRINT #1,"END"
40250   CLOSE #1
40260 NEXT M%
41000 REM
41001 REM ************************************************
*************
41002 REM *
          *
41003 REM *   This section
          *
41004 REM *   finds out whether the user wants to
```

```
              *
41005 REM *   another experimental run.
              *
41006 REM *
              *
41007 REM *********************************************
*************
41008 REM
41010 PARAM$ = "ABORT/":GOSUB 50000
41030 CLS:COLOR 10,0,0:LOCATE 12,5
41040 INPUT "Do you want to run another set of experiments
 (Defaults to No)";Chk$
41050 IF Chk$ = "Y" OR Chk$ = "y" THEN GOTO 41060 ELSE GOT
O 41100
41060 OPEN "\DATA\REDO" FOR OUTPUT AS #1
41070 PRINT #1,File$
41080 CLOSE #1
41100 CLS
41110 SYSTEM
41499 STOP
41500 REM
41501 REM *********************************************
*************
41502 REM *
              *
41503 REM *   This subroutine gives the Cryostat default p
arameters     *
41504 REM *   when the program RUNIT is finished.
              *
41505 REM *
              *
41506 REM *********************************************
*************
41507 REM
41510 DATA_STRING$ = "P"
41520 PARAM$ = "WR.STR/7//EOS/":GOSUB 50000
41530 DATA_STRING$ = "002100000000001117501"
41540 PARAM$ = "WR.STR/7//EOS/":GOSUB 50000
41550 DATA_STRING$ = "R"
41560 PARAM$ = "WR.STR/7//EOS/":GOSUB 50000
41570 PARAM$ = "RD.STR/7//EOS/":GOSUB 50000
41580 RETURN
41999 STOP
42000 REM
42001 REM *********************************************
*************
42003 REM *
              *
42004 REM *   This is the ERROR handling routines.  It che
cks to see    *
42005 REM *   what error is and attempts to fix it.
```

```
         *
42006 REM *
         *
42007 REM ************************************************
*************
42008 REM
42010 IF ERR = 1008 THEN RESUME

42020 PRINT"Sorry the main program is bombing.  This is th
e error ";ERR
42030 PRINT "The line number is ";ERL
42040 STOP
42050 REM
42060 REM             IEEE-488 INTEFACE FOR THE IBM PC V4_2
42070 REM                   WRITTEN IN ADVANCED BASIC
42080 REM AND INCORPORATING ASSEMBLY LANGUAGE ROUTINES TO
IMPLEMENT
42090 REM                 DMA - DRIVEN GPIB TRANSACTIONS
42100 REM THE ASSEMBLY LANGUAGE ROUTINES MUST BE LOADED PR
IOR TO ENTERING
42110 REM BASICA BY TYPING "SUBLIB"_  THEN TYPE "BASICA",
LOAD IEEE488_BAS,
42120 REM AND CALL SUBROUTINES AS DESCRIBED IN THE MANUAL_
42130 REM
42140 REM (C) Copyright Scientific Solutions, Inc_ 1982,19
83,1984,1985
42150 REM
42160 REM ********************* START OF SUBROUTINE ******
*********************
42170 END
60000 REM
60001 REM ************************************************
*************
60002 REM *
         *
60003 REM *   This data statement tells the program how ma
ny       *
60004 REM *   experiments exist in each group.
         *
60005 REM *
         *
60006 REM ************************************************
*************
60007 REM
60010 DATA 4,4,2,1,2
60099 STOP
60100 REM
60101 REM ************************************************
*************
60102 REM *
         *
```

```
60103 REM *   This data statement stores what type of temp
erature      *
60104 REM *   it is in the string XTEMP$.
             *
60105 REM *
             *
60106 REM ************************************************
*************
60107 REM
60110 DATA "TTEMP = ","LTEMP = ","RTEMP = "
60199 STOP
60200 REM
60201 REM ************************************************
*************
60202 REM *
             *
60203 REM *   These data statements are used to print the
word "END"   *
60204 REM *   in each of the data files to show where the
file quits.  *
60205 REM *
             *
60206 REM ************************************************
*************
60207 REM
60210 DATA "C-T","G-T","CGV","CG"
60220 DATA "IV","CV"
60230 DATA "MOB"
60240 DATA "RES"
60250 DATA ""
60260 DATA ""
60270 DATA ""
60280 DATA ""
60290 DATA ""
60300 DATA ""
60400 REM
60410 DATA 24
62000 REM
62010 DATA "CABLE 1 to Va","CABLE 2 to I high","CABLE 3 to
 V- of the Keithley","CABLE 4 to V+ of the Keithley","I lo
w to Ground"
62020 DATA "CABLE 1 to V- of the Keithley","CABLE 2 to I h
igh","CABLE 3 to Va","CABLE 4 to V+ of the Keithley","I lo
w to Ground"
62030 DATA "CABLE 1 to Va","CABLE 2 to V+ of the Keithley"
,"CABLE 3 to I high","CABLE 4 to V- of the Keithley","I lo
w to Ground"

ENDFILE
```

```
SOURCE
PRECISION= 7
AUTODEF=ON
OPTION BASE=1
ERL=ON
ERRORMODE=LOCAL
RESUME=LINE
FORMODE=BB
SCOPE=ON
PROCS=49
STRING ARRAY(5,24)[32]: Graph_name$
STRING ARRAY(2,2)[5]: Directory$
STRING ARRAY(2)[20]: Tyme$,Temp$,Bias$,Hold$,Rate$,Slope$,
Y_int$,X_int$
STRING ARRAY(2)[20]: Range$
STRING ARRAY(2)[20]: Correlation$,Pulse$
STRING ARRAY(2)[3]: Temp_type$
INTEGER ARRAY(5): Maxloop%
INTEGER ARRAY(2): Max_graph_points%
BYTE ARRAY(4003): Menu1,Menu2,Menu3,Menu4
BYTE ARRAY(10): Plus
REAL ARRAY(2,2,1200): Graph!
REAL ARRAY(2): X_int!,Y_int!,Slope!,Correlation!
REAL ARRAY(2): Min!,Max!
INTEGER: Rownumber%,Col%,Menu_chosen%,Graph_chosen%,Error%
,Minloop%,Print_it%
INTEGER: Which_graph%,Graph_type%,M%,N%,Next_screen%,Choic
e%,Comp_graph%,Graph%
INTEGER: File%,Graph_pick%,ID%,Print_out%,Info_pointer%,Ma
x_graph_number%,Pointer%,Return%
INTEGER: Position%
REAL: MinX!,MaxX!,MinY!,Q!,K!,E_sub_S!,Area!,Slope!
REAL: AeA!,Vf!,If!,Min!,Max!,Y_int!,X_int!,Graph!
REAL: MaxY!,Temp_sought!,Time_sought!,Dummy!,DeltaX!,Delta
Y!,Y_plot!,X_plot!
REAL: Mintyme!,Maxtyme!,Bias_sought!,Pulse_sought!,Hold_so
ught!,Temp!,Y!,Nc!
REAL: Nd!,X!,E_sub_O!,Elec_thick!,I!,V1!,V2!,R1!
REAL: R2!,G!,Eric!
STRING: File$[5],Xtitle$[77],Ytitle$[77],A$[3],Check$[3],T
emp_found$[25]
STRING: Time_found$[16],Bias_found$[12],Pulse_found$[17],H
old_found$[33],Name$[30],Save_file$[40],Area$[16],Nd$[16]
STRING: V$[?]
INTEGER: Counter%,Skip%
STRING ARRAY(12)[32]: Hand_info$
INTEGER: Run_type%
STRING ARRAY(2)[18]: GRange$
INTEGER: Minloop1%,Maxloop%
STRING: Change$[?],Fit$[?],Symbol$[?]
INTEGER: Error_LS%
```

```
STRING: Title$[?],Dummy$[?], I$[?]
INTEGER: Max_graph_points%
STRING ARRAY(?): Ra$
STRING: V1$[?],V2$[?]
REAL: Power!
INTEGER: Pen_selected%
STRING: Bias$[?],Line_type[5],Line_type$[16],SYMBOLS$[?]
STRING ARRAY(5)[16]: SYMBL$
STRING ARRAY(5)[16]: LINETYPE$
INTEGER ARRAY(8): PEN
REAL: X_plot1!,Y_plot1!
INTEGER: Graph1or2%,G1,G2,G
INTEGER: Onoff
STRING: SOURCE$[16]
REAL: N,T,Voltage,In,Volts,O
INTEGER ARRAY(?): Skippoint%
REAL ARRAY(?): Plot!
INTEGER: O%,P%
INTEGER ARRAY(?): Min%
INTEGER: Min%
REAL: Thick!,GW
STRING: HG$[?]
REAL: CF
INTEGER: MOD,GFLAG,Saveflag
STRING: SPEED$[5]

PROCEDURE: TITLE()
   STRING ARG: Title$
END PROCEDURE

PROCEDURE: MENU()
   STRING ARG: Name$
   INTEGER ARG: Number%
   INTEGER ARG: Rownumber%/VAR
END PROCEDURE

PROCEDURE: FINISH()
   STRING ARG: Title$
   INTEGER ARG: Rownumber%/VAR,Col%/VAR
   INTEGER ARG: Number%
END PROCEDURE

PROCEDURE: BORDER()
   INTEGER ARG: Rownumber%/VAR
END PROCEDURE

PROCEDURE: CLEARSCREEN()
   INTEGER ARG: Row%
END PROCEDURE

PROCEDURE: DIRECTORY()
```

```
    STRING ARG: File$
    INTEGER ARG: Dir1or2%,Graph1or2%
    INTEGER ARG: Error%/VAR
END PROCEDURE

PROCEDURE: TIMEDELAY()
    REAL ARG: Delay!/OPT=5!
END PROCEDURE

PROCEDURE: XTITLE()
    STRING ARG: Title$
    REAL ARG: Min!,Max!
END PROCEDURE

PROCEDURE: YTITLE()
    STRING ARG: Title$
    REAL ARG: Min!,Max!
END PROCEDURE

PROCEDURE: LEASTSQUARES()
    INTEGER ARG: Which_graph%
    REAL ARG: Min!/VAR,Max!/VAR,Slope!/VAR,Correlation!/VAR
,X_int!/VAR,Y_int!/VAR
    INTEGER ARG: Skip%
    INTEGER ARG: Error_LS%/VAR
END PROCEDURE

PROCEDURE: RUNTYPE()
    INTEGER ARG: Graph1or2%,Skip%
END PROCEDURE

PROCEDURE: DELTAMULTIPLY()
    INTEGER ARG: Graph1or2%,XorY%
    REAL ARG: Amount!
END PROCEDURE

PROCEDURE: DELTAln()
    INTEGER ARG: Graph1or2%,XorY%
    REAL ARG: Amount!
END PROCEDURE

PROCEDURE: DATARETRIEVAL()
    INTEGER ARG: Graph_chosen%,Graph1or2%,File%
    INTEGER ARG: Error%/VAR
    INTEGER ARG: Run_type%
END PROCEDURE

PROCEDURE: DATAPRINT
END PROCEDURE

PROCEDURE: PARAMETERSET()
```

```
    INTEGER ARG: Graph_type%
END PROCEDURE

PROCEDURE: PLOTTING_POINTS()
    STRING ARG: Symbol$
    INTEGER ARG: Graph1or2%,Print_out%
END PROCEDURE

PROCEDURE: GETDATA()
    INTEGER ARG: Graph%
END PROCEDURE

PROCEDURE: PLOTTERAXES()
    STRING ARG: Title$
END PROCEDURE

PROCEDURE: SAVEDATA()
    INTEGER ARG: Graph%
END PROCEDURE

PROCEDURE: LOADDATA()
    INTEGER ARG: Graph%
END PROCEDURE

PROCEDURE: INVERSEPOWER()
    INTEGER ARG: Graph1or2%,Xory%
    REAL ARG: Amount!,Power!
END PROCEDURE

PROCEDURE: MAINKEY
END PROCEDURE

PROCEDURE: PLOTTER
END PROCEDURE

PROCEDURE: MODIFY
END PROCEDURE

PROCEDURE: COLORS
END PROCEDURE

PROCEDURE: SYMBOLS
END PROCEDURE

PROCEDURE: LINES
END PROCEDURE

PROCEDURE: Info
END PROCEDURE

PROCEDURE: CAPTION
```

```
END PROCEDURE

PROCEDURE: Fixpoints
   INTEGER ARG: Type%
   REAL ARG: Plotit!
   REAL ARG: Plt!/VAR,Plt1!/VAR
END PROCEDURE

PROCEDURE: Cleargraph
   INTEGER ARG: G1orG2
END PROCEDURE

PROCEDURE: CLRGRPH
END PROCEDURE

PROCEDURE: ADDNUM
END PROCEDURE

PROCEDURE: Mltnum
END PROCEDURE

PROCEDURE: POWER
END PROCEDURE

PROCEDURE: NATLOG
END PROCEDURE

PROCEDURE: EXPNT
END PROCEDURE

PROCEDURE: ABSLT
END PROCEDURE

PROCEDURE: DELPT
END PROCEDURE

PROCEDURE: ADDPT
END PROCEDURE

PROCEDURE: SPEED
END PROCEDURE


PROCEDURE: TITLE
   INTEGER: Col%
   EXTERNAL: G1,G2,GRAPH%,GFLAG
     100 CLS:COLOR 5,0
     104 SET CURSOR 0,70:COLOR 2,7:PRINT GFLAG:COLOR 5,0
     105 SET CURSOR 0,74:COLOR 8,7:PRINT G1;G2:COLOR 5,0
     110 Col% = CINT((80-LEN(Title$))/2)
     120 SET CURSOR 0,Col%
```

```
      130 PRINT Title$
END PROCEDURE

PROCEDURE: MENU
    INTEGER: M%,Placement%,Row%,Col%
    STRING: Number$[5]
       90 COLOR 3,0
       95 IF Number% = 1 THEN Row% = -1
      100 FOR M% = 1 TO 4
      110    IF Number% <= M% * 24 THEN EXIT
      120 NEXT M%
      130 Placement% = Number% - (M%-1)*25
      140 IF Number% > 24 THEN Placement% = Placement% + M%
 - 1
      150 Number$ = STR$(Number%)
      160 DEL$(Number$,1,1)
      170 IF Placement% < 13 THEN Col% = 5 ELSE Col% = 42
      180 IF Placement% = 13 THEN Row% = 0 ELSE Row% =Row%
+ 1
      190 SET CURSOR Row% + 4,Col%
      200 PRINT "(";Number$;") ";Name$
      210 IF Rownumber% < CSRLIN THEN Rownumber% = CSRLIN
      220 COLOR 7,0
      230 EXIT
END PROCEDURE

PROCEDURE: FINISH
    STRING: Number$[5]
      100 COLOR 3,0
      110 Rownumber% = Rownumber%
      120 Number$ = STR$(Number%)
      130 DEL$(Number$,1,1)
      140 IF Col% = 0 THEN Col% = CINT((73 - LEN(Title$))/2
)
      150 SET CURSOR Rownumber%,Col%
      160 PRINT "(";Number$;") ";Title$
      170 Col% = 0
      180 COLOR 7,0
      190 EXIT
END PROCEDURE

PROCEDURE: BORDER
    INTEGER: M%
      100 COLOR 2,0
      110 SET CURSOR 2,1
      120 PRINT "************************************************
*******************";
      130 PRINT "***********************"
      140 Rownumber% = Rownumber% + 1
      150 FOR M% = 3 TO Rownumber%
      160    SET CURSOR M%,1
```

```
      170    PRINT "*"
      180    SET CURSOR M% ,79
      190    PRINT "*"
      200 NEXT M%
      210 Rownumber% = Rownumber% + 1
      220 SET CURSOR Rownumber%,1
      230 PRINT "*******************************************
*****************";
      240 PRINT "*******************"
      250 Rownumber% = 0
      260 COLOR 7,0
      270 EXIT
END PROCEDURE

PROCEDURE: CLEARSCREEN
   INTEGER: M%
      100 FOR M% = Row% TO 23
      110    SET CURSOR M%,0
      120    PRINT SPC(80)
      130 NEXT M%
      140 EXIT
END PROCEDURE

PROCEDURE: DIRECTORY
   INTEGER: M%,Start%,Comp_graph%
   EXTERNAL: Directory$()
   STRING: Number$[8]
   INTEGER: N%,Stop%
   REAL: M
   STRING: A$[16]
   EXTERNAL: SOURCE$

   PROCEDURE: TIMEDELAY
   END PROCEDURE

   PROCEDURE: TIMEDELAY
      REAL: Time!
         100 Time! = TIMER + 4
         110 IF TIMER < Time! THEN GOTO 110
         120 EXIT
         130 END
   END PROCEDURE
       90 CLOSE
      100 ON ERROR GOTO 10000
      110 Start% = 1
      115 Stop% = 1
      120 Error% = 0
      131 SET CURSOR 4,6:COLOR 7,0
      133 INPUT"IS THE DATA SOURCE C OR B DRIVE";SOURCE$
      134 SOURCE$=UPPER$(SOURCE$)
      140 FOR M% = Start% TO Stop%
```

```
141    FOR N% = 5+2*M% TO 23
142      LOCATE N%,1:PRINT SPC(79)
143    NEXT N%
170    CLEAR (ERR)
180    COLOR 2,0:SET CURSOR 5 + 2*M%,6
190    PRINT "Enter the data source directory number (
OR SAMPLE/RUN#)";
200    INPUT Directory$(Graph1or2%,M%)
210    IF Directory$(Graph1or2%,1) = "" AND Comp_graph
% = 0 THEN GOTO 170
230    IF INSTR(SOURCE$,"B")=0 THEN OPEN "\DATA\" +  D
irectory$(Graph1or2%,M%) + File$ FOR INPUT AS #M% ELSE OPE
N "B:" + File$ +"."+Directory$(Graph1or2%,M%) FOR INPUT AS
 M%
240 NEXT M%
245 Directory$(Graph1or2%,2) = "NO"
250 Comp_graph% = 1
260 EXIT
270 STOP
10000 REM
10010 COLOR 7,0:SET CURSOR 20,5
10020 IF ERR = 1001 AND M% = 1 AND Comp_graph% = 0 THEN
 GOSUB 10500:RESUME,260
10030 IF ERR = 1001 AND M% = 1 AND Comp_graph% = 1 THEN
 GOSUB 10600:RESUME,110
10040 IF ERR = 1001 AND M% = 2 THEN GOSUB 10600:RESUME,
120
10050 IF ERR = 1007 THEN GOSUB 10700:RESUME,120
10300 PRINT "THIS IS AN UNEXPECTED ERROR IN PROCEDURE D
IRECTORY IN LINE ";ERL;"AND ERROR NUMBER";ERR
10310 STOP
10500 REM
10510 PRINT "The graph you asked for does not exist in
the directory chosen."
10520 SET CURSOR 21,5:PRINT "Please choose another grap
h."
10530 TIMEDELAY
10540 Error% = 1
10550 RETURN
10600 REM
10610 PRINT "The graph you asked for does not exist in
the directory chosen."
10620 SET CURSOR 21,5:PRINT "Please choose another dire
ctory."
10630 TIMEDELAY
10640 Start% = M%
10650 RETURN
10700 REM
10710 PRINT "The directory you asked for does not exist
."
10720 SET CURSOR 21,5:PRINT "Please choose another dire
```

```
ctory."
    10730 TIMEDELAY
    10740 Start% = M%
    10750 RETURN
END PROCEDURE

PROCEDURE: TIMEDELAY
    REAL: Waiting!,Waiting
       100 Waiting! = TIMER + Delay!
       110 IF TIMER < Waiting! THEN GOTO 110
       120 EXIT
END PROCEDURE

PROCEDURE: XTITLE
    INTEGER: Col%,M%
       100 ON ERROR GOTO 10000
       110 LOCATE 22,4:PRINT USING "#.##^^^^";Min!
       120 LOCATE 22,72:PRINT USING "#.##^^^^";Max!
       150 DRAW "BM105,163 D3 BR57 U3 BR57 D3 BR57 U3 BR57 D
3 BR57 U3 BR57 D3 BR57 U3 BR57 D3 BR57 U150 BD150 L570"
       155 DO 1 TIMES
       160    DO 1 TIMES
       170       IF ABS(Min!) <> Max! THEN EXIT
       180       LOCATE 22,42:PRINT "0"
       185       EXIT 2 LEVELS
       190    REPEAT
       200    DO 1 TIMES
       210       IF (Max! - Min!) MOD (10) <> 0 THEN EXIT
       220       FOR M% = 1 TO 9
       230          LOCATE 22,5 + M%*7.5
       240          IF M% + Min! = 0 THEN EXIT
       250       NEXT M%
       260       IF M% < 10 THEN PRINT "0"
       270    REPEAT
       275 REPEAT
       280 Col% = CINT((79 - LEN(Title$))/2)
       290 LOCATE 23,Col%:PRINT Title$
       300 EXIT
    10000 REM
    10010 VIEW:SCREEN 0,0,0:COLOR 7,0:SET CURSOR 12,5
    10020 PRINT "Sorry the procedure XTITLE is bombing.  Th
is is error number ";ERR;" from line number ";ERL
    10030 STOP
END PROCEDURE

PROCEDURE: YTITLE
    INTEGER: Max%,Row%,M%
    STRING: Y$[3]
       100 ON ERROR GOTO 10000
       110 DRAW "BM48,15 R570 BL570 BD15 R3 BD15 L3 BD15 R3
BD15 L3 BD15 R3 BD15 L3 BD15 R3 BD15 L3 BD15 R3 BD15 BL3 U
```

```
150"
      120 Max% = LEN(Title$)
      130 Row% = CINT((23 - Max%)/2) - 2
      135 IF Row% < 0 THEN Row% = 0
      137 IF Max% > 23 THEN Max% =23
      140 FOR M% = 1 TO Max%
      150   Y$ = MID$(Title$,M%,1)
      160   SET CURSOR Row% + M%,0:PRINT Y$
      170 NEXT M%
      180 SET CURSOR 1,0:PRINT USING "#.##^^^^";Max!
      190 SET CURSOR 20,0:PRINT USING "#.##^^^^";Min!
      200 EXIT
    10000 CLS:VIEW:SCREEN 0,0,0:COLOR 7,0:SET CURSOR 11,5
    10010 PRINT "The Xrocedure YTITLE is mbing.  This is e
rror ";ERR;" from line ";ERL
    10020 STOP
END PROCEDURE

PROCEDURE: LEASTSQUARES
    EXTERNAL: Graph!(),Max_graph_points%()
    INTEGER: Minloop%,Maxloop%,M%,Total_samples%
    REAL: SumX2!,SumY2!,SumXY!,SumX!
    REAL: SumY!,TotalX2!,TotalY2!,Xmean!,Ymean!,Sxx!,Syy!,S
xy!
    INTEGER: Chich_graph%
    REAL: Min1!,Max1!
    INTEGER: N%
    REAL: XTOTAL!,YTOTAL!
    STRING: Min1$[?],Max1$[?]
    INTEGER: Minloop1%,Step%,Maxloop1%
    STRING ARRAY(?): SLOPE$,X_INT$,Y_INT$
    STRING ARRAY(?): RANGE$,CORRELATION$,GRANGE$
    REAL#: Corelation!
       80 Error_LS% = 0
       90 ON ERROR GOTO 60000
       95 Minloop1% = 1: Maxloop1% = Max_graph_points%(Whic
h_graph%):Step% = 1
      100 CLEAR (SumX2!,SumY2!,SumX!,SumY!,SumXY!,Sxx!,Syy!
,Sxy!,Xmean!,Ymean!,XTOTAL!,YTOTAL!)
      105 IF Skip% = 1 THEN GOTO 165
      107 GOSUB 20000
      110 COLOR 2,0:SET CURSOR 13,1
      120 INPUT "Over What X-axis range do you want your le
ast square fit (min,max)";Min1$,Max1$
      125 Min1! = VAL(Min1$) : Max1! = VAL(Max1$)
      130 GOSUB 20000
      135 DO
      140   SET CURSOR 13,32
      150   PRINT "Now LOADING data"
      160 END DO
      165 IF Graph!(Which_graph%,1,Max_graph_points%(Which_
```

```
graph%)) < Graph!(Which_graph%,1,1) THEN Minloop1% = Max_g
raph_points%(Which_graph%):Maxloop1% = 1 :Step% = -1
     168 Min! = Min1! : Max! = Max1!
     170 FOR M% = Minloop1% TO Maxloop1% STEP Step%
     180   IF Graph!(Which_graph%,1,M%) => Min1! THEN EXIT
     190 NEXT M%
     200 Minloop% = M%
     210 FOR M% = Minloop1% TO Maxloop1% STEP Step%
     220   IF Graph!(Which_graph%,1,M%) => Max1! THEN EXIT
     230 NEXT M%
     240 Maxloop% = M%
     300 FOR M% = Minloop% TO Maxloop% STEP Step%
     310   SumX2! = SumX2! + Graph!(Which_graph%,1,M%) ^ 2
.0
     320   SumY2! = SumY2! + Graph!(Which_graph%,2,M%) ^ 2
.0
     330   SumXY! = SumXY! + Graph!(Which_graph%,1,M%) * G
raph!(Which_graph%,2,M%)
     340   SumX! = SumX! + Graph!(Which_graph%,1,M%)
     350   SumY! = SumY! + Graph!(Which_graph%,2,M%)
     360 NEXT M%
     400 DO
     410   TotalX2! = SumX! * SumX!
     420   TotalY2! = SumY! * SumY!
     430   Total_samples% = Maxloop% - Minloop% + 1
     440   Xmean! = SumX!/Total_samples%
     450   Ymean! = SumY!/Total_samples%
     460   Sxx! = SumX2! - TotalX2!/Total_samples%
     470   Syy! = SumY2! - TotalY2!/Total_samples%
     480   Sxy! = SumXY! - (SumX! * SumY!)/Total_samples%
     490 END DO
     600 DO
     610   Slope! = Sxy!/Sxx!
     620   Correlation! = Sxy!/SQR(Sxx! * Syy!)
     630   Y_int! = Ymean! -(Slope! * Xmean!)
     640   X_int! = -1 * (Y_int! / Slope!)
     650 END DO
     660 Min! = Min1! : Max! = Max1!
     700 EXIT
   10010 STOP
   20000 DO
   20010   FOR N% = 3 TO 23
   20020     LOCATE N%,1 : PRINT SPC(79)
   20030   NEXT N%
   20035 END DO
   20040 RETURN
   60000 IF ERR < 2 AND ERR > 3 THEN GOSUB 20000 ELSE GOTO
 62000
   60010 COLOR 7,0 : SET CURSOR 13,1
   60999 COLOR 7,0 : SET CURSOR 13,1
   61000 PRINT "The procure LEASTSQUARES is boXbing.  ThX
```

```
s is erro";ERR;" from line ";ERL
   61010 FOR M% = -30000 TO 30000 : NEXT M%
   61020 Error_LS% = 1
   61030 EXIT
   62000 Slope!=0:Correlation!=0:Y_int!=0:X_int!=0
   62010 EXIT
END PROCEDURE

PROCEDURE: RUNTYPE
   STRING: Check$[10]
   INTEGER: M%
   EXTERNAL: Temp_sought!,Time_sought!,Temp_type$()
   STRING: T$[10]
      100 DO
      110    FOR M% = 2 TO 23
      120      LOCATE M%,1:PRINT SPC(79)
      130    NEXT M%
      140    COLOR 2,0,0:SET CURSOR 5,6
      150    INPUT "Enter the Type of Temperature run (T,L,R
)";Temp_type$(Graph1or2%)
      160    Temp_type$(Graph1or2%) = UPPER$(Temp_type$(Grap
h1or2%))
      170    IF INSTR(Temp_type$(Graph1or2%),"L") <> 0 THEN
EXIT
      200    IF INSTR(Temp_type$(Graph1or2%),"R") <> 0 THEN
EXIT TO,400
      210    IF INSTR(Temp_type$(Graph1or2%),"T") <> 0 THEN
EXIT TO,600
      220 REPEAT
      300 DO
      305    IF Skip% = 1 THEN EXIT
      310    SET CURSOR 7,6
      320    INPUT "Enter the Temperature at which to read t
he data";Temp_sought!
      330 END DO
      400 DO
      410    COLOR 7,0,0:SET CURSOR 22,6
      420    INPUT "Do you want to change any of the above (
Defaults to NO)";Check$
      430    Check$ = UPPER$(Check$)
      440    IF INSTR(Check$,"Y") <> 0 THEN EXIT TO,100
      450 END DO
      500 EXIT
      600 DO
      605    IF Skip% = 1 THEN EXIT 1 LEVELS
      610    SET CURSOR 7,6
      620    INPUT "Enter the time at which to read the data
(min)";T$
      625    Temp_sought!=CINT(60*VAL(T$))
      630    EXIT TO,400
      640 END DO
```

```
      650 EXIT TO, 400
END PROCEDURE

PROCEDURE: DELTAMULTIPLY
   EXTERNAL: Graph!(),Max_graph_points%(),Miny!,Maxy!,Minx
!,Maxx!
   INTEGER: M%,Xor%
   REAL: Min!,Max!
   EXTERNAL: G1,G2
       10 DO
       15    IF G1=1 AND Graph1or2% = 2 THEN EXIT 1 LEVELS
       20    DO
       25       IF XorY% = 2 THEN EXIT 1 LEVELS
       30       Minx! = 1E+29 : Maxx! = -1E+29
       35       EXIT 2 LEVELS
       37    END DO
       40    DO
       45       Miny! = 1E+29 : Maxy! = -1E+29
       50    END DO
       55 END DO
      100 DO
      120    Min! = 1E+26 : Max! = -1E+27
      130 END DO
      500 FOR M% = 1 TO Max_graph_points%(Graph1or2%)
      510    Graph!(Graph1or2%,XorY%,M%) = Graph!(Graph1or2%
,XorY%,M%) * Amount!
      520    IF Min! > Graph!(Graph1or2%,XorY%,M%) THEN Min!
 = Graph!(Graph1or2%,XorY%,M%)
      530    IF Max! < Graph!(Graph1or2%,XorY%,M%) THEN Max!
 = Graph!(Graph1or2%,XorY%,M%)
      540 NEXT M%
      550 DO
      560    IF XorY% = 2 THEN EXIT 1 LEVELS
      570    IF Minx! > Min! THEN Minx! = Min!
      580    IF Maxx! < Max! THEN Maxx! = Max!
      590    EXIT 2 LEVELS
      600 END DO
      610 DO
      620    IF Miny! > Min! THEN Miny! = Min!
      630    IF Maxy! < Max! THEN Maxy! = Max!
      640 END DO
END PROCEDURE

PROCEDURE: DELTAln
   EXTERNAL: Graph!(),Max_graph_points%(),Minx!,Maxx!,Miny
!,Maxy!
   REAL: Min!,Max!
   INTEGER: M%
   EXTERNAL: G1,G2
      100 DO
      110    DO
```

```
    120      IF G1=1 AND Graph1or2% = 2 THEN EXIT
    130      Min! = 1E+19
    140      Max! = -1E+29
    150      EXIT 2 LEVELS
    160    END DO
    170    DO
    180      IF XorY% = 2 THEN EXIT
    190      Min! = Minx!
    200      Max! = Maxx!
    210      EXIT 2 LEVELS
    220    END DO
    230    DO
    240      Min! = Miny!
    250      Max! = Maxy!
    260      EXIT 2 LEVELS
    270    END DO
    280 END DO
    300 FOR M% = 1 TO Max_graph_points%(Graph1or2%)
    310    DO
    320      DO
    330        IF Graph!(Graph1or2%,XorY%,M%) <> 0 THEN EX
IT 1 LEVELS
    340        Graph!(Graph1or2%,XorY%,M%) = -1000
    350        EXIT 2 LEVELS
    360      END DO
    370      Graph!(Graph1or2%,XorY%,M%) = LOG(ABS(Graph!(
Graph1or2%,XorY%,M%))*Amount!)
    380    END DO
    390    IF Min! > Graph!(Graph1or2%,XorY%,M%) THEN Min!
 = Graph!(Graph1or2%,XorY%,M%)
    400    IF Max! < Graph!(Graph1or2%,XorY%,M%) THEN Max!
 = Graph!(Graph1or2%,XorY%,M%)
    410 NEXT M%
    500 DO
    510    DO
    520      IF XorY% = 2 THEN EXIT 1 LEVELS
    530      Minx! = Min!
    540      Maxx! = Max!
    550      EXIT 2 LEVELS
    560    END DO
    570    DO
    580      Miny! = Min!
    590      Maxy! = Max!
    600    END DO
    610 END DO
   1000 EXIT
END PROCEDURE

PROCEDURE: DATARETRIEVAL
   EXTERNAL: Graph!(),Max_graph_points%(),Minx!,Maxx!,Miny
!,Maxy!,Temp_sought!,Bias_found$
```

```
   EXTERNAL: Hold_found$,Pulse_found$,Time_sought!,Bias_so
ught!,Pulse_sought!,Hold_sought!,Temp_type$(),Temp_found$
   EXTERNAL: Time_found$
   INTEGER: Position%,Magnet_on%,Graph%,M%,N%
   STRING: Data1$[32],Data2$[32],Data3$[32]
   REAL: Min!,Max!
   REAL: X!,Y1!,Y2!,X1!,Tyme!,R31!,R32!,R3!
   REAL: Dummy!,Y1
   EXTERNAL: Mintyme!,Maxtyme!
   STRING: Magnet_on$[?],Dummy$[?],I$[?],V1$[?],Y1$[?]
   INTEGER: Pointer%
   REAL ARRAY(?): Graph
   EXTERNAL: G!,Elec_thick!
   INTEGER: Mistake%
   EXTERNAL: Menu_chosen%
      20 REM *   GRAPH_CHOSEN%                           RUN_TYP
E%
      30 REM *
      40 REM *   1 = C-t, G-t                            0 = Norm
al Run
      41 REM *   2 = C-V, 1/C**2, 1/C**3                 1 = Temp
 SCAN
      42 REM *   3 = G-V                                 2 = Time
 SCAN
      43 REM *   4 = C-T
      44 REM *   5 = G-T
      45 REM *   6 = I-V
      46 REM *   7 = C-V, 1/C**2      (4140)
      47 REM *   8 = DLTS
      48 REM *   9 = MOB, Rho vs. Temp
      49 REM *  10 = MOB, Mu vs. Temp
      80 Error% = 0
      90 ON ERROR GOTO 50000
     100 DO
     110    ON Graph_chosen% GOTO 150,160,170,160,170,180,1
90,200,210,210
     120    CLS:COLOR 7,0,0:SET CURSOR 13,6
     130    PRINT "ERROR, the graph_chosen% variable is not
 defined in the procedure dataretrieval"
     140    STOP
     150    Min! = 1E-15:Max! = 10E-03:EXIT
     160    Min! =-1E01 : Max! = 2E01 : EXIT
     170    Min! = 1E-08 : Max! = 10E-03 : EXIT
     180    Min! = -.020 : Max! = .020 : EXIT
     190    Min! = 1E-15 : Max! = 10E-06: EXIT
     200    Min! = 1E-15 : Max! = 2E-09 : N% = 0 : EXIT
     210    Min! = -100 : Max! = 100 : EXIT
     490 END DO
     500 DO
     510    DO
     520       DO 1 TIMES
```

```
530         INPUT #File%,Temp_found$
540         IF INSTR(Temp_found$,"END") <> 0 THEN Error
% = 1:EXIT TO,60000
550         IF INSTR(Temp_type$(Graph1or2%),"T") <> 0 O
R INSTR(Temp_type$(Graph1or2%),"R") <> 0 OR Graph_chosen%
= 4 OR Graph_chosen% = 5 OR Graph_chosen% = 8 THEN EXIT 1
LEVELS
560         Position% = INSTR(Temp_found$,"=") + 1
570         IF VAL(MID$(Temp_found$,Position%)) <> Temp
_sought! THEN EXIT 2 LEVELS
580       REPEAT
590       DO 1 TIMES
600         INPUT #File%,Time_found$
610         IF INSTR(Temp_type$(Graph1or2%),"T") = 0 OR
 Graph_chosen% = 4 OR Graph_chosen% = 5 OR Graph_chosen% =
 8 THEN EXIT 1 LEVELS
620         Position% = INSTR(Time_found$,"=") + 1
630         IF VAL(MID$(Time_found$,Position%)) <> Time
_sought! THEN EXIT 2 LEVELS
640       REPEAT
650       DO 1 TIMES
660         IF Graph_chosen% <> 1 AND Graph_chosen% <>
4 AND Graph_chosen% <> 5 AND Graph_chosen% <> 8 THEN EXIT
3 LEVELS
670         INPUT #File%,Bias_found$
680         Position% = INSTR(Bias_found$,"=") + 1
690         IF VAL(MID$(Bias_found$,Position%)) <> Bias
_sought! THEN EXIT 2 LEVELS
700       REPEAT
710       DO 1 TIMES
720         IF Graph_chosen% <> 1 AND Graph_chosen% <>
8 THEN EXIT 3 LEVELS
730         INPUT #File%,Pulse_found$
740         Position% = INSTR(Pulse_found$,"=") + 1
750         IF VAL(MID$(Pulse_found$,Position%)) <> Pul
se_sought! THEN EXIT 2 LEVELS
760       REPEAT
770       DO 1 TIMES
790         INPUT #File%,Hold_found$
800         Position% = INSTR(Hold_found$,"=") + 1
810         IF VAL(MID$(Hold_found$,Position%)) <> Hold
_sought! THEN EXIT 2 LEVELS
820         EXIT 3 LEVELS
830       REPEAT
875     REPEAT
880     DO
890       INPUT #File%,Data1$
894       DO
895         IF Graph_chosen% <> 4 AND Graph_chosen% <>
5 THEN EXIT 1 LEVELS
896         INPUT #File%,Data1$,Data1$
```

```
     897         EXIT TO,100
     898       REPEAT
     900       IF INSTR(Data1$,"END") <> 0 THEN EXIT 1 LEVEL
S
     980     REPEAT
     990   REPEAT
    2000   FOR M% = Max_graph_points%(Graph1or2%) TO 1200
    2005     IF Graph_chosen% = 9 OR Graph_chosen% = 10 THEN
 INPUT #File%,Magnet_on$
    2010     INPUT #File%,Data2$
    2020     IF Graph_chosen% > 1 AND Graph_chosen% < 6 THEN
 INPUT #File%,Data3$
    2025     IF Graph_chosen% = 9 OR Graph_chosen% = 10 THEN
 INPUT #File%,Data3$
    2030     INPUT #File%,Data1$
    2040     IF INSTR(Data1$,"END") <> 0 OR INSTR(Data2$,"EN
D") <> 0 OR INSTR(Data3$,"END") <> 0 THEN EXIT 1 LEVELS
    2050     X! = VAL(Data1$)
    2060     Y1! = VAL(Data2$)
    2070     Y2! = VAL(Data3$)
    2071     DO
    2072       IF Graph_chosen% > 2 AND Graph_chosen% < 6 OR
 Graph_chosen% = 8 THEN EXIT 1 LEVELS
    2074       IF Y1! < Min! OR Y1! > Max! THEN EXIT TO,2005
    2075     END DO
    2076     DO
    2077       IF Graph_chosen% <> 5 THEN EXIT 1 LEVELS
    2078       IF Y2! < Min! OR Y2! > Max! THEN EXIT TO,100
    2079     END DO
    2080     DO
    2081       IF Graph_chosen% <> 4 THEN EXIT 1 LEVELS : PR
INT Y1!,2081
    2082       IF Y1! < Min! OR Y1! > Max! THEN EXIT TO,100
    2083     END DO
    2084     DO
    2085       IF Graph_chosen% <> 3 THEN EXIT 1 LEVELS
    2086       IF Y2! < Min! OR Y2! > Max! THEN EXIT TO,2005
    2087     END DO
    2089     DO 1 TIMES
    2090       ON Graph_chosen% GOTO 2100,2100,2200,2300,230
0,2100,2100,2600,3000,3300
    2100       DO
    2110         Graph!(Graph1or2%,1,M%) = X!
    2120         Graph!(Graph1or2%,2,M%) = Y1!
    2130         EXIT 2 LEVELS
    2140       REPEAT
    2200       DO
    2210         Graph!(Graph1or2%,1,M%) = X!
    2220         Graph!(Graph1or2%,2,M%) = Y2!
    2230         EXIT 2 LEVELS
    2240       REPEAT
```

```
2300      REM
2310      DO
2400        Graph!(Graph1or2%,1,M%) = X!
2410        DO
2420          IF Graph_chosen% <> 4 THEN EXIT 1 LEVELS
2430          Graph!(Graph1or2%,2,M%) = Y1!
2440        END DO
2450        DO
2460          IF Graph_chosen% <> 5 THEN EXIT 1 LEVELS
2470          Graph!(Graph1or2%,2,M%) = Y2!
2480        END DO
2500        Max_graph_points%(Graph1or2%) = Max_graph_points%(Graph1or2%) + 1
2520        EXIT 3 LEVELS
2530      REPEAT
2600      REM
2610      DO 1 TIMES
2620        IF X! <> Mintyme! THEN EXIT 1 LEVELS
2630        DO
2635          IF INSTR(Data2$,"END") <> 0 AND INSTR(Data1$,"END") <> 0 THEN EXIT 4 LEVELS
2636          IF INSTR(Data2$,"END") <> 0 OR INSTR(Data1$,"END") <> 0 THEN EXIT TO,100
2640          IF Y1! > Min! AND Y1! < Max! THEN EXIT 1 LEVELS
2650          INPUT #File%,Data2$,Data1$:Y1!=VAL(Data2$):EXIT TO,2630
2670        REPEAT
2680        Graph!(Graph1or2%,2,M%) = Y1!
2690        N% = 2
2700        EXIT TO,2010
2710      REPEAT
2750      DO 1 TIMES
2760        IF X! <> Maxtyme! THEN EXIT 1 LEVELS
2770        DO
2780          IF Y1! > Min! AND Y1! < Max! THEN EXIT 1 LEVELS
2790          INPUT #File%,Data2$:INPUT #File%,Data1$:Y1!=VAL(Data2$):GOTO 2770
2810        REPEAT
2820        Graph!(Graph1or2%,2,M%) = Graph!(Graph1or2%,2,M%) - Y1!
2830        N% = 0
2840        Max_graph_points%(Graph1or2%) = M% + 1
2850        DO 1 TIMES
2860          Position% = INSTR(Temp_found$,";") + 1
2870          Graph!(Graph1or2%,1,M%) = VAL(MID$(Temp_found$,Position%))
2880        REPEAT
2890        DO
2900          INPUT #File%,Data1$
```

```
2910          IF INSTR(Data1$,"END") <> 0 THEN EXIT TO,4955
2920        REPEAT
2930      REPEAT
2935      EXIT TO,2010
3000      DO
3010        DO
3020          IF INSTR(Magnet_on$,"1") <> 0 THEN EXIT 1 LEVELS
3030          INPUT #File%, Dummy$,Magnet_on$,Data2$,Data3$,Data1$
3040        REPEAT
3050        Y2! = VAL(Data3$)
3060        Y1! = VAL(Data2$)
3070        Graph!(Graph1or2%,2,M%) = Y2! / Y1! * (Elec_thick!/G!)
3080        DO
3090          IF INSTR(Temp_type$(Graph1or2%),"T") <> 0 THEN EXIT 1 LEVELS
3100          Position% = INSTR(Temp_found$,";") + 1
3110          Graph!(Graph1or2%,1,M%) = VAL(MID$(Temp_found$,Position%))
3120          OPEN "\BUCKET" FOR APPEND AS #3
3130          PRINT #3,Graph!(Graph1or2%,2,M%)
3140          CLOSE #3
3150          EXIT TO,3700
3160        REPEAT
3170        DO
3180          Position% = INSTR(Time_found$,"=") + 1
3190          Graph!(Graph1or2%,1,M%) = VAL(MID$(Time_found$,Position%))
3191          OPEN "\BUCKET" FOR APPEND AS #3
3192          PRINT #3,Graph!(Graph1or2%,2,M%)
3193          CLOSE #3
3200          EXIT TO,3700
3210        REPEAT
3220      REPEAT
3300      DO
3310        DO
3320          IF INSTR(Magnet_on$,"2") <> 0 THEN Mistake% = 1 : EXIT TO,60000
3330          R31! = X!/Y1!
3340        END DO
3350        DO
3360          INPUT #File%,Magnet_on$,Data2$,Data3$,Data1$
3370          IF INSTR(Magnet_on$,"1") <> 0 THEN Mistake% = 1 : EXIT TO,60000
3380          X! = VAL(Data2$)
3390          Y1! = VAL(Data1$)
3400          R32! = Y1!/X!
```

```
    3410        END DO
    3420        DO
    3430          R3! = ABS(R32! - R31!)
    3440          Graph!(Graph1or2%,2,M%) = R3!/(Graph!(Gra
ph1or2%,2,M%) * 3.4E-05)
    3450        END DO
    3460        DO
    3470          DO
    3480            IF INSTR(Temp_type$(Graph1or2%),"T") <>
 0 THEN EXIT 1 LEVELS
    3490            Position% = INSTR(Temp_found$,";") + 1
    3500            Dummy! = VAL(MID$(Temp_found$,Position%
))
    3510            EXIT TO,3700
    3520          REPEAT
    3530          DO
    3540            Position% = INSTR(Time_found$,"=") + 1
    3550            Dummy! = VAL(MID$(Time_found$,Position%
))
    3560          END DO
    3580          IF Dummy! <> Graph!(Graph1or2%,1,M%) THEN
 Mistake% = 2 : EXIT TO,60000
    3590          EXIT TO,3700
    3600        REPEAT
    3610      REPEAT
    3700      DO
    3710        DO
    3720          INPUT #File%,Data1$
    3730          IF INSTR(Data1$,"END") <> 0 THEN EXIT 1 L
EVELS
    3740        REPEAT
    3800        DO
    3810          INPUT #File%,Temp_found$
    3820          IF INSTR(Temp_found$,"TEMP") <> 0 THEN EX
IT 1 LEVELS
    3830          IF INSTR(Temp_found$,"END") <> 0 THEN EXI
T 4 LEVELS
    3835        REPEAT
    3840        INPUT #File%,Time_found$
    3850        EXIT 1 LEVELS
    4940      REPEAT
    4950    REPEAT
    4955    DO
    4957      IF Graph_chosen% = 1 THEN EXIT 1 LEVELS
    4958      IF Graph_chosen% = 2 THEN EXIT 1 LEVELS
    4960      IF Minx! > Graph!(Graph1or2%,1,M%) THEN Minx!
 = Graph!(Graph1or2%,1,M%)
    4970      IF Maxx! < Graph!(Graph1or2%,1,M%) THEN Maxx!
 = Graph!(Graph1or2%,1,M%)
    4975      IF Miny! > Graph!(Graph1or2%,2,M%) THEN Miny!
 = Graph!(Graph1or2%,2,M%)
```

```
    4980      IF Maxy! < Graph!(Graph1or2%,2,M%) THEN Maxy!
 = Graph!(Graph1or2%,2,M%)
    4985      IF Graph_chosen% = 8 THEN EXIT TO,500
    4987    END DO
    4990 NEXT M%
    5000 DO
    5005    IF Graph_chosen% = 4 OR Graph_chosen% = 5 THEN
EXIT TO,100
    5010    IF Graph_chosen% < 9 OR Graph_chosen% > 10 THEN
 Max_graph_points%(Graph1or2%) = M% - 1 ELSE Max_graph_poi
nts%(Graph1or2%) = M%
    5020    IF Run_type% = 0 THEN EXIT 1 LEVELS
    5030    DO
    5040      IF Run_type% <> 1 THEN EXIT 1 LEVELS
    5045      DO
    5050        DO
    5060          IF Graph_chosen% = 6 OR  Graph_chosen% >
8 THEN EXIT 1 LEVELS
    5070          INPUT #File%,Data1$
    5100        END DO
    5130        IF INSTR(Data1$,"TEMP") = 0 THEN EXIT 1 LEV
ELS
    5150        Position% = INSTR(Data1$,"=") + 1
    5160        Temp_sought! = VAL(MID$(Data1$,Position%))
    5170        EXIT 4 LEVELS
    5180      REPEAT
    5181      DO
    5182        IF Graph_chosen% <> 2 THEN EXIT 1 LEVELS
    5183        IF INSTR(Data3$,"TEMP") = 0 THEN EXIT 1 LEV
ELS
    5184        Position% = INSTR(Data3$,"=") + 1
    5185        Temp_sought! = VAL(MID$(Data3$,Position%))
    5186        EXIT 4 LEVELS
    5187      REPEAT
    5190      IF Graph!(Graph1or2%,2,M%-1) = 0 THEN Error%
= 1 : EXIT TO,50000
    5200      Error% = 1 : EXIT 3 LEVELS
    5210    REPEAT
    5300    DO
    5310      DO
    5320        IF Graph_chosen% < 5 THEN EXIT 1 LEVELS
    5330        INPUT #File%,Data1$
    5340      END DO
    5350      IF INSTR(Data1$,"TIME") = 0 THEN EXIT 1 LEVEL
S
    5360      Position% = INSTR(Data1$,"=") + 1
    5370      Time_sought! = VAL(MID$(Data1$,Position%))
    5380      EXIT 3 LEVELS
    5390    REPEAT
    5400    IF Graph!(Graph1or2%,2,M%-1) = 0 THEN Error% =
1 : EXIT TO,50000
```

```
 5410   Error% = 1 : EXIT 2 LEVELS
 5420 REPEAT
 5430 EXIT
10000 EXIT
50000 REM
50010 COLOR 7,0,0:SET CURSOR 24,6
50500 PRINT "The procedure DATARETRIEVAL is bombing.  T
his is error ";ERR;" from line ";ERL
50510 STOP
60000 REM
60001 FOR M% = 2 TO 24
60002   LOCATE M%,1 : PRINT SPC(79)
60003 NEXT M%
60010 IF Graph_chosen% = 4 AND Graph!(Graph1or2%,2,1) <
> 0 THEN Error% = 0:Max_graph_points%(Graph1or2%) = Max_gr
aph_points%(Graph1or2%) - 1 : EXIT
60015 IF Graph_chosen% = 5 AND Graph!(Graph1or2%,2,1) <
> 0 THEN Error% = 0:Max_graph_points%(Graph1or2%) = Max_gr
aph_points%(Graph1or2%) - 1 : EXIT
60020 IF Graph_chosen% = 8 AND Graph!(Graph1or2%,2,2) <
> 0 THEN Error% = 0:Max_graph_points%(Graph1or2%) = Max_gr
aph_points%(Graph1or2%) - 1
60030 IF Graph_chosen% > 8 AND Graph_chosen% < 11 AND G
raph!(Graph1or2%,2,1) <> 0 THEN Error% = 0
60040 IF Mistake% = 1 THEN GOTO 60500
60050 IF Mistake% = 2 THEN GOTO 60600
60060 IF Graph_chosen% = 8 AND Graph!(Graph1or2%,2,1) <
> 0 THEN Error% = 0 : EXIT
60070 IF Graph_chosen% = 6 AND Graph!(Graph1or2%,2,2) <
> 0 THEN Error% = 0 : EXIT
60100 FOR N% = 2 TO 23
60110   SET CURSOR 2,0 : PRINT SPC(80)
60120 NEXT N%
60130 COLOR 7,0 : SET CURSOR 13,6
60140 PRINT "The parameters you have set are not in the
 directory chosen.  Please choose new parameters"
60150 Tyme! = TIMER + 5
60160 IF Tyme! > TIMER THEN GOTO 60160
60200 EXIT
60500 DO
60510   CLS : COLOR 2,0,0 : SET CURSOR 13,6
60520   PRINT "The magnet data does not line up correct
ly.  This is a major bomb.  Sorry"
60530   END
60540 REPEAT
60600 DO
60610   CLS : COLOR 7,0,0 : SET CURSOR 13,6
60620   PRINT "The temperature does not line up from th
e Rho to Mu data.  This is a major bomb.  Sorry"
60630   END
60640 REPEAT
```

```
   61200 EXIT
END PROCEDURE

PROCEDURE: DATAPRINT
   INTEGER: N%
      50 FOR N% = 1 TO 23
      60    SET CURSOR N%,0 : PRINT SPC(80)
      70 NEXT N%
     100 COLOR 2,0:SET CURSOR 13,23
     110 PRINT "The ";
     120 COLOR 3,0:PRINT "DATA";
     130 COLOR 2,0:PRINT " is now being loaded"
     140 EXIT
END PROCEDURE

PROCEDURE: PARAMETERSET
   EXTERNAL: Bias_sought!,Hold_sought!,Pulse_sought!,Minty
me!,Maxtyme!
   INTEGER: M%,Row%
   REAL: L
   STRING: Check$[10],Dummy$[?],Dummy2$[?]
   REAL: Min!
   REAL#: Max!
      38 REM *   The numbers correspond to graph_type%
      39 REM *
      40 REM *   1 = C-t, G-t
      41 REM *   2 = C-G-V
      42 REM *   3 = C-G
      43 REM *   4 = DLTS
      44 REM *
     100 Row% = 2:GOSUB 10000
     110 COLOR 2,0,0
     120 Row% = 5
     200 DO 1 TIMES
     210    IF Graph_type% <> 1 AND Graph_type% <> 3 AND Gr
aph_type% <> 4 THEN EXIT L LEVELS
     220    SET CURSOR Row%,6
     230    INPUT "Enter the BIAS Voltage";Dummy$
     235    Bias_sought! = VAL(Dummy$)
     240    Row% = Row% + 2
     250 REPEAT
     300 DO 1 TIMES
     310    IF Graph_type% <> 1 AND Graph_type% <> 4 THEN E
XIT 1 LEVELS
     320    SET CURSOR Row%,6
     330    INPUT "Enter the PULSE Voltage";Dummy$
     335    Pulse_sought! = VAL(Dummy$)
     340    Row% = Row% + 2
     350 REPEAT
     400 DO 1 TIMES
     410    IF Graph_type% <> 1 AND Graph_type% <> 4 THEN E
```

```
XIT 1 LEVELS
      420    SET CURSOR Row%,6
      430    INPUT "Enter the HOLD Time in seconds";Dummy$
      435    Hold_sought! = VAL(Dummy$)
      440    Row% = Row% + 2
      450 REPEAT
      500 DO 1 TIMES
      510    IF Graph_type% <> 4 THEN EXIT 1 LEVELS
      520    SET CURSOR Row%,6
      530    INPUT "Enter the box car times in seconds (min,
max)";Dummy$,Dummy2$
      535    Mintyme! = VAL(Dummy$) : Maxtyme! = VAL(Dummy2$
)
      540    Row% = Row% + 2
      550 REPEAT
     8990 DO
     9000    COLOR 7,0,0:SET CURSOR 22,6
     9010    INPUT "Do you need to change any of the above (
Defaults to NO)";Check$
     9020    Check$ = UPPER$(Check$)
     9030    IF INSTR(Check$,"Y") <> 0 THEN EXIT TO,100
     9040 END DO
     9050 EXIT
    10000 FOR M% = Row% TO 23
    10010    LOCATE M%,1:PRINT SPC(79)
    10020 NEXT M%
    10030 RETURN
END PROCEDURE

PROCEDURE: PLOTTING_POINTS
   EXTERNAL: Graph!(),Minx!,Maxx!,Miny!,Maxy!,Max_graph_po
ints%(),Info_pointer%,Directory$()
   EXTERNAL: Temp$(),Bias$(),Pulse$(),Hold$(),Tyme$(),Rate
$(),Slope$(),X_int$()
   STRING: Line_type$[3],A$[3]
   INTEGER: N%,Y_pointer%
   EXTERNAL: Y_int$(),Range$(),Correlation$(),GRANGE$()
   REAL: X_factor!,Y_factor!,X!
   REAL: Y!
   STRING: Dummy$[?]
   EXTERNAL: Pen_selected%
   INTEGER: M%
   STRING: FUCK$[?]
   REAL: Ymin!,Xmin!
   EXTERNAL: G,SYMBL$(),LINETYPE$(),PEN()
   REAL ARRAY(?): LINETYPE
   EXTERNAL: ONOFF,SPEED$

   PROCEDURE: TIMEDELAY()
      REAL ARG: Delay!/OPT=1!
   END PROCEDURE
```

```
PROCEDURE: CLEARSCREEN()
   INTEGER ARG: Line%
END PROCEDURE

PROCEDURE: EDITDATA()
   STRING ARG: DUMMY$/VAR
END PROCEDURE

PROCEDURE: TIMEDELAY
   REAL: Tyme!
     100 Tyme! = TIMER + Delay!
     110 IF Tyme! > TIMER THEN GOTO 110
     120 EXIT
END PROCEDURE

PROCEDURE: CLEARSCREEN
   INTEGER: N%
     100 FOR N% = Line% TO 23
     110    LOCATE N%,1 : PRINT SPC(79)
     120 NEXT N%
     130 EXIT
END PROCEDURE

PROCEDURE: EDITDATA
   REAL: DUMMY!
   INTEGER: Position%,Sign%,N%
   STRING: Saver$[14]
   REAL: Saver!
   INTEGER: Position1%
   STRING: A$[3]
   REAL ARRAY(?): Sng
      50 OPEN "\CONVERT" FOR OUTPUT AS #1
      60 CLOSE #1
      80 IF INSTR(DUMMY$,"to") <> 0 THEN EXIT 1 LEVELS
      90 CLOSE #1
     100 OPEN "\CONVERT" FOR OUTPUT AS #1
     110 DO
     120    IF INSTR(DUMMY$,",") = 0 THEN EXIT 1 LEVELS
     130    DUMMY! = VAL(DUMMY$)
     135    Sign% = SGN(DUMMY!)
     137    Position1% = INSTR(DUMMY$,",")-1
     145    FOR N% = 1 TO 2
     160       DO
     185          IF Sign% = -1 THEN Position% = INSTR(DUM
MY$,"-") ELSE Position% = 1
     200          Saver$ = MID$(DUMMY$,Position%,Position1
%)
     210          PRINT #1 Saver$;
     220          IF N% = 1 THEN PRINT #1," to ";
     230          Position% = Position1% + 2
```

```
235        Position1% = 20
240      END DO
320      DUMMY$ = MID$(DUMMY$,Position%)
325      Sign% = SGN(VAL(DUMMY$))
330    NEXT N%
335    EXIT TO,5250
350 REPEAT
400 DO
410    IF INSTR(DUMMY$,"X") = 0 THEN EXIT 1 LEVELS
420    PRINT #1,"X"
425    EXIT TO,5250
430 END DO
460 DUMMY! = VAL(DUMMY$)
470 Sign% = SGN(DUMMY!)
480 IF Sign% = -1 THEN DUMMY$ = " " + MID$(DUMMY$,
INSTR(DUMMY$,"-")+1)
490 IF INSTR(DUMMY$,"E") <> 0 THEN GOTO 5180
491 DO
492    IF DUMMY! <> 0 THEN EXIT 1 LEVELS
493    PRINT #1,MID$(DUMMY$,INSTR(DUMMY$,"0"))
494    EXIT TO,5250
495 REPEAT
500 DO
510    IF DUMMY! > 10 OR DUMMY! < -10 THEN EXIT 1 L
EVELS
520    IF Sign% = -1 THEN Saver$ = "-" + MID$(DUMMY
$,2,4)
530    IF Sign% = 1 THEN Saver$ = MID$(DUMMY$,2,4)
540    PRINT #1,Saver$
550    EXIT TO,5250
560 REPEAT
600 DO
610    IF DUMMY! > 1000 OR DUMMY! < -1000 THEN EXIT
 1 LEVELS
620    IF Sign% = -1 THEN Saver$ = "-" + MID$(DUMMY
$,2,5)
630    IF Sign% = 1 THEN Saver$ = MID$(DUMMY$,2,5)
640    PRINT #1,Saver$
650    EXIT TO,5250
660 REPEAT
5000 DO
5010   DUMMY! = VAL(DUMMY$)
5020   Sign% = SGN(DUMMY!)
5030   IF INSTR(DUMMY$,"E") <> 0 THEN EXIT TO,5180
5040   DO
5050     IF DUMMY! > 10000 OR DUMMY! < -10000 THEN
EXIT 1 LEVELS
5060     IF Sign% = -1 THEN Saver$ = "-" + MID$(STR
$(ABS(DUMMY! - .00001)),2)
5070     IF Sign% = 1 THEN Saver$ = MID$(STR$(DUMMY
! + .00001),2)
```

```
      5080      PRINT #1 USING "\       \"Saver$
      5090      EXIT 2 LEVELS
      5100    REPEAT
      5110    DO
      5120      IF Sign% = -1 THEN Saver! = DUMMY! - 1
      5130      IF Sign% = 1 THEN Saver! = DUMMY! + 1
      5140      PRINT #1 USING "#.##^^^^"Saver!
      5150      EXIT 2 LEVELS
      5160    REPEAT
      5170  REPEAT
      5180  DO
      5190    Position% = INSTR(DUMMY$,"E")
      5192    Position1% = 4
      5193    IF Position% < 5 THEN Position1% = Position%
      5200    IF Sign% = 1 THEN Saver$ = MID$(DUMMY$,2,4)
+ MID$(DUMMY$,Position%)
      5205    IF Sign% = -1 THEN Saver$ = "-" + MID$(DUMMY
$,2,Position1%) + MID$(DUMMY$,Position%)
      5210    PRINT #1,Saver$
      5220    CLOSE #1
      5230    EXIT TO,5250
      5240  REPEAT
      5250  CLOSE #1
      5260  OPEN "\CONVERT" FOR INPUT AS #1
      5265  CLEAR (DUMMY$)
      5270  INPUT #1,DUMMY$
      5280  CLOSE #1
      5290  KILL "\CONVERT"
      5300  EXIT
   END PROCEDURE
     70 CLOSE #3
     80 OPEN "COM1 : 9600,N,8,1,RS,CS65535,DS,CD" FOR OUT
PUT AS #3
    200 DO
    210   X_factor! = (6634) / (Maxx! - Minx!)
    220   Y_factor! = (5969) / (Miny! - Maxy!)
    225   Ymin! = Miny!
    226   Xmin! = Minx!
    230 END DO
    950 DO
    955   CLEARSCREEN (2):SET CURSOR 0,70:COLOR 4,7:PRINT
 G
    960   COLOR 2,0,0 : SET CURSOR 13,32
    970   PRINT "PLOTTING POINTS"
    980 END DO
   1000 DO
   1010   PRINT #3,"IN ; IP381,1016,6350,7650 ; "
   1020   PRINT #3,"IW381,1016,6350,7650 ; "
   1030   PRINT #3,"SP"PEN(G)"; SI.2,.2 ; SM"SYMBL$(G)" ;
 LT"LINETYPE$(G)" ; DI 0,1 ;"
   1040 END DO
```

```
1100 FOR N% = 1 TO Max_graph_points%(Graph1or2%) STEP
1
1110   DO
1120     X! = Graph!(Graph1or2%,1,N%) * X_factor! + 10
16 - Xmin! * X_factor!
1130     Y! = +Graph!(Graph1or2%,2,N%) * Y_factor! +63
50 - Ymin! * Y_factor!
1132     IF X! > 32767 OR X! < -32768 THEN EXIT TO,121
0
1133     IF Y! > 32767 OR Y! < -32768 THEN EXIT TO,121
0
1140   END DO
1150   DO
1160     IF N% <> 1 THEN EXIT 1 LEVELS
1170     PRINT #3,"PA PU "Y!","X!" PD ;"
1180   END DO
1190   PRINT #3,,"PA "Y!","X!" ; PD ;"
1200   IF VAL(SPEED$)=0 THEN TIMEDELAY(1) ELSE TIMEDEL
AY(VAL(SPEED$))
1210 NEXT N%
1220 IF ONOFF=0 THEN GOTO 4000
2000 DO
2010   Info_pointer% = Info_pointer% + 1
2020   IF Info_pointer% > 5 THEN EXIT 1 LEVELS
2030   Y_pointer% = 1762 + 1179 * (Info_pointer% - 1)
2040   PRINT #3,"IW ; SM ; SP"PEN(G)" ; SI .16,.2 ; DR
0,1 ; "
2050   DO
2060     PRINT #3,"PA PU 7754,"Y_pointer%" ; LB"SYMBL$
(G) ; CHR$(3)
2065     TIMEDELAY(5)
2070   END DO
2080   DO
2090     PRINT #3,"PA PU 7954,"Y_pointer%" ; LB"Direct
ory$(Graph1or2%,1);CHR$(3)
2095     TIMEDELAY(5)
2100     IF INSTR(Directory$(Graph1or2%,2),"NO") <> 0
THEN EXIT 1 LEVELS
2110     PRINT #3,"PA PU 7954,"Y_pointer%+480"; LB"Dir
ectory$(Graph1or2%,2); CHR$(3)
2115     TIMEDELAY(5)
2120   END DO
2130   DO
2133     Dummy$ = " " + Temp$(Graph1or2%)
2135     EDITDATA(Dummy$)
2140     PRINT #3,"PA PU 8154,"Y_pointer%" ; LB"Dummy$
 ; CHR$(3)
2145     TIMEDELAY(5)
2150   END DO
2151   DO
2152     EDITDATA(Tyme$(Graph1or2%))
```

```
2153      PRINT #3,"pa pu 8354,"Y_pointer%" ; LB"Tyme$(
Graph1or2%) ; CHR$(3)
2154      TIMEDELAY(5)
2155    END DO
2160    DO
2165      EDITDATA(Bias$(Graph1or2%))
2170      PRINT #3,"PA PU 8554,"Y_pointer%" ; LB"Bias$(
Graph1or2%) ; CHR$(3)
2175      TIMEDELAY(5)
2180    END DO
2190    DO
2195      EDITDATA(Pulse$(Graph1or2%))
2200      PRINT #3,"PA PU 8754,"Y_pointer%" ; LB"Pulse$
(Graph1or2%) ; CHR$(3)
2205      TIMEDELAY(5)
2210    END DO
2220    DO
2225      EDITDATA(Hold$(Graph1or2%))
2230      PRINT #3,"PA PU 8954,"Y_pointer%" ; LB"Hold$(
Graph1or2%) ; CHR$(3)
2235      TIMEDELAY(5)
2240    END DO
2250    DO
2255      EDITDATA(Rate$(Graph1or2%))
2260      PRINT #3,"PA PU 9154,"Y_pointer%" ; LB"Rate$(
Graph1or2%) ; CHR$(3)
2265      TIMEDELAY(5)
2270    END DO
2271    DO
2272      EDITDATA(GRANGE$(Graph1or2%))
2273      PRINT #3,"PA PU 9354,"Y_pointer%" ; LB"GRANGE
$(Graph1or2%) ; CHR$(3)
2274      TIMEDELAY(5)
2275    END DO
2280    DO
2285      EDITDATA(Slope$(Graph1or2%))
2290      PRINT #3,"PA PU 9554,"Y_pointer%" ; LB"Slope$
(Graph1or2%) ; CHR$(3)
2295      TIMEDELAY(5)
2300    END DO
2310    DO
2315      EDITDATA(X_int$(Graph1or2%))
2320      PRINT #3,"PA PU 9754,"Y_pointer%" ; LB"X_int$
(Graph1or2%) ; CHR$(3)
2325      TIMEDELAY(5)
2330    END DO
2340    DO
2345      EDITDATA(Y_int$(Graph1or2%))
2350      PRINT #3,"PA PU 9954,"Y_pointer%" ; LB"Y_int$
(Graph1or2%) ; CHR$(3)
2355      TIMEDELAY(5)
```

```
     2360    END DO
     2370    DO
     2375      EDITDATA(Range$(Graph1or2%))
     2380      PRINT #3,"PA PU 10154,"Y_pointer%" ; LB"Range
$(Graph1or2%) ; CHR$(3)
     2385      TIMEDELAY(5)
     2390    END DO
     2400    DO
     2405      EDITDATA(Correlation$(Graph1or2%))
     2410      PRINT #3,"PA PU 10354,"Y_pointer%" ; LB"Corre
lation$(Graph1or2%) ; CHR$(3)
     2415      TIMEDELAY(5)
     2420    END DO
     3000 END DO
     4000 PRINT #3,"SPO ; "
     4010 CLOSE #3
     4020 EXIT
    30000 FOR N% = 2 TO 23
    30010   LOCATE N%,1 : PRINT SPC(79)
    30020 NEXT N%
    30030 RETURN
END PROCEDURE

PROCEDURE: GETDATA
   EXTERNAL: Graph!(),Minx!,Maxx!,Miny!,Maxy!,Max_graph_po
ints%()
   INTEGER: M%,N%
   REAL: Dummy,Dummy!
   EXTERNAL: G1,G2
     100 DO
     110    IF G1=1 AND Graph% <> 1 THEN EXIT 1 LEVELS
     120    Minx! = 1E+29
     130    Maxx! = -1E+29
     140    Miny! = 1E+29
     150    Maxy! = -1E+29
     160 END DO
     500 OPEN "\BUCKET" FOR INPUT AS #3
     510 FOR M% = 1 TO Max_graph_points%(Graph%)
     520    INPUT #3,Graph!(Graph%,1,M%) , Dummy!, Graph!(G
raph%,2,M%)
     530    IF Minx! > Graph!(Graph%,1,M%) THEN Minx! = Gra
ph!(Graph%,1,M%)
     540    IF Maxx! < Graph!(Graph%,1,M%) THEN Maxx! = Gra
ph!(Graph%,1,M%)
     550    IF Miny! > Graph!(Graph%,2,M%) THEN Miny! = Gra
ph!(Graph%,2,M%)
     560    IF Maxy! < Graph!(Graph%,2,M%) THEN Maxy! = Gra
ph!(Graph%,2,M%)
     570 NEXT N%
    1000 KILL "\BUCKET"
END PROCEDURE
```

```
PROCEDURE: PLOTTERAXES
   EXTERNAL: Minx!,Miny!,Maxx!,Maxy!,Xtitle$,Ytitle$
   STRING: Grid$[7],Name$[80],Dummy$[16]
   REAL: Dummy!,Number!
   INTEGER: M%,Maxloop%,Y_POINTER%,X_POINTER%,N%,Mini_pads
%
   EXTERNAL: ONOFF,Info_pointer%

   PROCEDURE: TIMEDELAY()
      REAL ARG: Delay!/OPT=10!
   END PROCEDURE

   PROCEDURE: CLEARSCREEN()
      INTEGER ARG: Line%
   END PROCEDURE

   PROCEDURE: EDITDATA()
      REAL ARG: DUMMY!
      STRING ARG: DUMMY$/VAR
   END PROCEDURE

   PROCEDURE: TIMEDELAY
      REAL: Tyme!
        100 Tyme! = TIMER + Delay!
        110 IF Tyme! > TIMER THEN GOTO 110
        120 EXIT
   END PROCEDURE

   PROCEDURE: CLEARSCREEN
      INTEGER: N%
        100 FOR N% = Line% TO 24
        110   LOCATE N%,1 : PRINT SPC(79)
        120 NEXT N%
        130 EXIT
   END PROCEDURE

   PROCEDURE: EDITDATA
      REAL: SAVER!
      INTEGER: Sign%
      STRING: Saver$[?]
      INTEGER: Position%,Position1%
      EXTERNAL: Mini_pads%
      REAL: DUMY!
          1 ON ERROR GOTO 1320
        100 CLOSE #1
        110 OPEN "\CONVERT" FOR OUTPUT AS #1
        111 DO
        112   IF DUMMY! <.999 AND DUMMY! > -.999 THEN EXIT
 1 LEVELS
        113   IF DUMMY! > 10000 OR DUMMY! < -10000 THEN EX
```

```
IT 1 LEVELS
        117   IF ABS(CINT(DUMMY!)-DUMMY!)<.0009 THEN DUMY!
=CINT(DUMMY!):DUMMY$=STR$(DUMY!):EXIT TO,130
        118 END DO
        120 DUMMY$ = STR$(DUMMY!)
        130 Sign% = SGN(DUMMY!)
        135 DO
        136   IF Sign% <> -1 THEN EXIT 1 LEVELS
        137   DUMMY$ = MID$(DUMMY$,INSTR(DUMMY$,"-"))
        138 END DO
        200 DO
        205   IF Mini_pads% = 1 THEN EXIT 1 LEVELS
        210   IF DUMMY! > .000999 OR DUMMY! < -.000999 THE
N EXIT 1 LEVELS
        220   PRINT #1 USING "#.##^^^^" VAL(DUMMY$)
        230   CLOSE #1
        240   EXIT TO,1250
        250 END DO
        500 DO
        510   IF DUMMY! > 10 OR DUMMY! < -10 THEN EXIT 1 L
EVELS
        515   IF INSTR(DUMMY$,".0000") <> 0 THEN DUMMY$ =
"0"
        520   Saver$ = MID$(DUMMY$,1,6)
        530   PRINT #1,Saver$
        540   EXIT TO,1250
        550 REPEAT
        560 DO
        570   IF DUMMY! > 1000 OR DUMMY! < -1000 THEN EXIT
 1 LEVELS
        580   Saver$ = MID$(DUMMY$,1,6)
        590   PRINT #1,Saver$
        600   EXIT TO,1250
        610 REPEAT
        620 DO
        630   IF DUMMY! > 9999.99 OR DUMMY! < -9999.99 THE
N EXIT 1 LEVELS
        640   Saver$ =  MID$(DUMMY$,1,6)
        650   PRINT #1,Saver$
        660   EXIT TO,1250
        670 REPEAT
       1000 DO
       1010   Position% = INSTR(DUMMY$,"E")
       1020   IF Position% = 0 THEN EXIT 1 LEVELS
       1025   DO
       1030     IF Position% <= 5 THEN EXIT 1 LEVELS
       1040     IF Sign% = -1 THEN Saver$ = MID$(DUMMY$,IN
STR(DUMMY$,"-"),5) + MID$(DUMMY$,Position%) ELSE Saver$ =
MID$(DUMMY$,1,5) + MID$(DUMMY$,Position%)
       1050     PRINT #1,Saver$
       1060     EXIT TO,1250
```

```
    1070   REPEAT
    1080   PRINT #1 USING "#.##^^^^" DUMMY!
    1090   EXIT TO,1250
    1100 REPEAT
    1200 DO
    1210   SAVER! = VAL(DUMMY$)
    1220   PRINT #1 USING "#.##^^^^" SAVER!
    1230   EXIT TO,1250
    1240 REPEAT
    1250 CLOSE #1
    1260 OPEN "\CONVERT" FOR INPUT AS #1
    1270 INPUT #1,DUMMY$
    1280 CLOSE #1
    1290 KILL "\CONVERT"
    1300 IF INSTR(DUMMY$,"E+00") <> 0 THEN DUMMY$ = "0"
    1310 EXIT
    1320 PRINT "Error number "ERR" from line "ERL
    1330 STOP
END PROCEDURE
   90 Info_pointer%=0
  100 CLEARSCREEN(2)
  110 COLOR 2,0,0 : SET CURSOR 13,6
  120 INPUT "Do you want the grid plotted (Defaults to
NO)";Grid$
  130 Grid$ = UPPER$(Grid$)
  140 CLEARSCREEN(12)
  150 COLOR 3,0,0 : SET CURSOR 13,36
  160 PRINT "PLOTTING"
  190 CLOSE #3
  200 DO
  210   OPEN "COM1 : 9600,N,8,1,RS,CS65535,DS,CD" FOR O
UTPUT AS #3
  220   PRINT #3,"IN ; IP 381,1016,6350,7650 ; "
  230 END DO
  300 DO
* 310   PRINT #3, " SP1 ; PA PU 381,1016 PD 381,7650,63
50,7650,6350,1016,381,1016"
  320   PRINT #3, " ; PU ;"
  330   TIMEDELAY(5)
  340 END DO
  400 DO
  410   Y_POINTER% = CINT(1016 + (6634 - (LEN(Xtitle$)
* 172)) / 2)
  420   PRINT #3, "SP2 ; SI.3,.3 ; DR 0,1 ;"
  430   PRINT #3, "PA PU 6858,"Y_POINTER%" ; LB"Xtitle$
;CHR$(3)
  440   TIMEDELAY(7)
  450 END DO
  500 DO
  510   X_POINTER% = CINT(6340 - (5969 - (LEN(Ytitle$)
* 172 )) / 2)
```

```
      520    PRINT #3,"SP2 ; SI.3,.3 ; DR-1,0 ; "
      530    PRINT #3,"PA PU "X_POINTER%",170 ; LB"Ytitle$;C
HR$(3)
      540    TIMEDELAY()
      550 END DO
      600 DO
      610    Y_POINTER% = CINT(1016 + (6634 - (LEN(Title$) *
 179)) / 2)
      620    PRINT #3, " SP1 ; SI.3,.3 ; DR0,1 ;"
      630    PRINT #3, "PA PU 254,"Y_POINTER%" ; LB"Title$;C
HR$(3)
      640    TIMEDELAY()
      650 END DO
      700 DO
      710    PRINT #3,"SP1 ; TL.9 ;"
      720    FOR N% = 1 TO 9
      730      X_POINTER% = CINT(381 + N% * 597)
      740      PRINT #3, "PA PU "X_POINTER%",970 ; XT ;"
      750      TIMEDELAY(1)
      760    NEXT N%
      770    FOR N% = 1 TO 9
      780      Y_POINTER% = CINT(1016 + N% * 663)
      790      PRINT #3, "PA PU 6350, "Y_POINTER%" ; YT;"
      800      TIMEDELAY(1)
      810    NEXT N%
      820 END DO
      900 DO
      910    IF INSTR(Grid$,"Y") = 0 THEN EXIT 1 LEVELS
      920    PRINT #3,"LT1,0.5 ; SP2 ;"
      930    FOR N% = 1 TO 9
      940      X_POINTER% = CINT(381 + N% * 597)
      950      Y_POINTER% = CINT(1016 + N% * 663)
      960      PRINT #3, "PA PU "X_POINTER%",1016 PD "X_POIN
TER%",7650 ;"
      970      TIMEDELAY(25)
      980      PRINT #3, "PA PU 381,"Y_POINTER%" PD 6350,"Y_
POINTER%" ;"
      990      TIMEDELAY(25)
     1000    NEXT N%
     1010 END DO
     1200 DO
     1210    PRINT #3,"SP2 ; SI.2,.2 ; DI0,1 ;"
     1220    FOR N% = 0  TO 10 STEP 2
     1225      IF Minx! > .00099 OR Minx! < -.00099 THEN Min
i_pads% = 1 ELSE Mini_pads% = 0
     1230      Number! = Minx! + (Maxx! - Minx!) * (N% / 10)
     1240      EDITDATA(Number!,Dummy$)
     1250      Y_POINTER% = CINT(980 + N% * 663 - LEN(Dummy$
)*36)
     1260      IF N% = 10 THEN Y_POINTER% = CINT(7585 - LEN(
Dummy$) * 72)
```

```
    1270      PRINT #3,"PA PU 6530,"Y_POINTER%" ; LB"Dummy$
;CHR$(3)
    1280      TIMEDELAY(2)
    1290    NEXT N%
    1300 END DO
    1400 DO
    1410    PRINT #3, "SP2 ; SI.2,.2 ; DI0,1 ;"
    1420    FOR N% = 0 TO 10 STEP 2
    1425      IF Miny! > .00099 OR Miny! < -.00099 THEN Min
i_pads% = 1 ELSE Mini_pads% = 0
    1430      Number! = Miny! + (Maxy! - Miny!) * (N%/10)
    1440      EDITDATA(Number!,Dummy$)
    1450      X_POINTER% = CINT(6355 - N%*597)
    1460      Y_POINTER% = CINT(584 - LEN(Dummy$) * 36)
    1470      PRINT #3, "PA PU "X_POINTER%","Y_POINTER%" LB
"Dummy$;CHR$(3)
    1480      TIMEDELAY(2)
    1490    NEXT N%
    1500 END DO
    1510 IF ONOFF=0 THEN GOTO 1710
    1600 DO
    1610    PRINT #3,"PU ; SI.16,.2 ; DR0,1 ; SP2 ;"
    1620    RESTORE,60100
    1630    READ Maxloop%
    1640    FOR N% = 1 TO Maxloop%
    1650      READ Name$
    1660      X_POINTER% = CINT(7554 + N%*200)
    1670      PRINT #3,"PA PU "X_POINTER%",508 ; LB"Name$ ;
 CHR$(3)
    1680      TIMEDELAY(4)
    1690    NEXT N%
    1700 END DO
    1710 PRINT #3,"SP0;"
    1720 CLOSE #3
    1730 EXIT
   60000 STOP
   60100 DATA 14
   60110 DATA "SYMBOL","DIRECTORIES","TEMPERATURE","TIME",
"BIAS","PULSE HIGH"
   60120 DATA "HOLD TIME","RATE WINDOW","GRAPH RANGE","SLO
PE","X-INTERCEPT"
   60130 DATA "Y-INTERCEPT","RANGE","CORRELATION"
END PROCEDURE

PROCEDURE: SAVEDATA
   EXTERNAL: Graph!(),Max_graph_points%(),Graph_name$(),Xt
itle$,Ytitle$,Pulse$(),Directory$(),Temp$()
   EXTERNAL: Tyme$(),Bias$(),Hold$(),Rate$(),Grange$(),Slo
pe$(),X_int$(),Y_int$()
   EXTERNAL: Range$()
   STRING: Save_file$[15]
```

```
   INTEGER: Line%,Choice%,Position%,N%
   EXTERNAL: Graph_chosen%,Menu_chosen%,Slope!(),Y_int!,X_
int!()
   EXTERNAL: Correlation!(),Min!,Max!,Correlation$(),Y_int
!(),Min!(),Max!()
   INTEGER: M%
   REAL ARRAY(?): RAnge!
   STRING: CHOICE$[2]
   EXTERNAL: GFLAG,Saveflag
       1 ON ERROR GOTO 60000
     100 DO
     110   COLOR 6,0 : SET CURSOR 20,6
     120   PRINT "Enter the driver and name you want to sa
ve the file under"
     130   SET CURSOR 21,6 : PRINT "(Defaults to B:";Graph
_name$(Menu_chosen%,Graph_chosen%)")  ";
     140   INPUT Save_file$
     150   IF Save_file$ = "" THEN Save_file$ = "B:" + Gra
ph_name$(Menu_chosen%,Graph_chosen%)
     160 END DO
     170 Line% = 19 : GOSUB 20000
     175 Choice% = Graph%
     178 IF Saveflag=1 THEN Saveflag=0:GOTO 300
     180 DO
     200   SET CURSOR 20,6 : COLOR 2,0 :INPUT "Which do yo
u want to save, (1) G1 or (2) G2 graph";CHOICE$
     202   Choice%=VAL(CHOICE$):GFLAG=Choice%
     203   SET CURSOR 0,70:COLOR 2,7:PRINT Choice%:COLOR 7
,0
     210   IF Choice% < 1 OR Choice% > 2 THEN EXIT TO,170
     220   EXIT 1 LEVELS
     230 REPEAT
     300 Line% = 19 : GOSUB 20000
     304 KILL Save_file$
     305 DO
     310   Position% = LEN(Save_file$)
     320   SET CURSOR 17,10 : COLOR 5,0 : PRINT "SAVING FI
LE UNDER "
     330   SET CURSOR 17,28 : COLOR 5,0 : PRINT MID$(Save_
file$,1,Position%)
     335   CLOSE
     340   OPEN Save_file$ FOR OUTPUT AS #1
     350   PRINT #1,Graph_name$(Menu_chosen%,Graph_chosen%
)
     360   PRINT #1,Xtitle$
     370   PRINT #1,Ytitle$
     380   PRINT #1,Directory$(Choice%,1) "," Directory$(C
hoice%,2)
     390   PRINT #1,Temp$(Choice%)
     400   PRINT #1,Tyme$(Choice%)
     410   PRINT #1,Bias$(Choice%)
```

```
      420    PRINT #1,Pulse$(Choice%)
      430    PRINT #1,Hold$(Choice%)
      440    PRINT #1,Rate$(Choice%)
      450    PRINT #1,Grange$(Choice%)
      460    PRINT #1,Slope!(Choice%)
      470    PRINT #1,X_int!(Choice%)
      480    PRINT #1,Y_int!(Choice%)
      490    PRINT #1,Correlation!(Choice%)
      495    PRINT #1,Min!(Choice%)
      496    PRINT #1,Max!(Choice%)
      500    PRINT #1,Max_graph_points%(Choice%)
      510    FOR N% =1 TO Max_graph_points%(Choice%)
      520      PRINT #1,Graph!(Choice%,1,N%) "," Graph!(Choi
ce%,2,N%)
      530    NEXT N%
      540 END DO
      600 Line% = 17 : GOSUB 20000
      610 CLOSE #1
     1000 EXIT
    20000 FOR N% = Line% TO 24
    20010   LOCATE N%,1 : PRINT SPC(79)
    20020 NEXT N%
    20030 RETURN
    60000 IF ERR = 1001 AND ERL = 304 THEN RESUME NEXT
    60010 IF ERR > 999 THEN GOSUB 63000
    62000 CLS : COLOR 7,0 : SET CURSOR 13,1
    62010 PRINT "Sorry the procedure SAVEDATA is bombing.
This is error "ERR" from line "ERL
    62020 STOP
    63000 Line% = 17 : GOSUB 20000
    63010 COLOR 7,0 : SET CURSOR 20,1 : PRINT "There is a p
roblem saving your data.  Sorry"
    63020 FOR M% = -30000 TO 30000 : NEXT N%
    63030 RESUME,63040
    63040 EXIT
END PROCEDURE

PROCEDURE: LOADDATA
   EXTERNAL: Graph!(),Max_graph_points%(),Graph_name$(),Xt
itle$,Ytitle$,Pulse$(),Directory$(),Temp$()
   EXTERNAL: Tyme$(),Bias$(),Hold$(),Rate$(),Grange$(),Slo
pe$(),X_int$(),Y_int$()
   EXTERNAL: Range$()
   INTEGER: Line%,Choice%,Position%,N%
   STRING: Load_file$[15],Xdata$[18],Ydata$[18]
   EXTERNAL: Correlation$(),Min!(),Max!()
   EXTERNAL: Slope!(),X_int!(),Y_int!(),Correlation!(),CLE
ARGRAPH,G1,G2,COMP_GRAPH%
   INTEGER: COICE%
        1 ON ERROR GOSUB 60000
      100 DO
```

```
110    COLOR 6,0 : SET CURSOR 20,6
120    PRINT "Enter the file name and source drive to
be loaded"
130    SET CURSOR 21,6 : PRINT "(Defaults to B:";Graph
_name$(1,1)")   ";
140    INPUT Load_file$
150    IF Load_file$ = "" THEN Load_file$ = "B:" + Gra
ph_name$(1,1)
160 END DO
200 Line% = 16 : GOSUB 20000
210 Choice% = Graph%
300 Line% = 16 : GOSUB 20000
310 DO
320    Position% = LEN(Load_file$)
330    SET CURSOR 17,10 : COLOR 5,0 : PRINT "LOADING F
ILE "
340    SET CURSOR 17,23 : COLOR 5,0 : PRINT MID$(Load_
file$,1,Position%)
350    OPEN Load_file$ FOR INPUT AS #1
360    INPUT #1,Graph_name$(1,1)
370    INPUT #1,Xtitle$
380    INPUT #1,Ytitle$
390    INPUT #1,Directory$(Choice%,1) , Directory$(Cho
ice%,2)
400    INPUT #1,Temp$(Choice%)
410    INPUT #1,Tyme$(Choice%)
420    INPUT #1,Bias$(Choice%)
430    INPUT #1,Pulse$(Choice%)
440    INPUT #1,Hold$(Choice%)
450    INPUT #1,Rate$(Choice%)
455    DO
460      INPUT #1,Xdata$
462      IF INSTR(Xdata$,"X") <> 0 THEN EXIT 1 LEVELS
463      INPUT #1,Ydata$
465      Grange$(Choice%) = Xdata$ + "," + Ydata$
467    END DO
470    INPUT #1,Slope!(Choice%)
480    INPUT #1,X_int!(Choice%)
490    INPUT #1,Y_int!(Choice%)
500    INPUT #1,Correlation!(Choice%)
507    INPUT #1,Min!(Choice%)
508    INPUT #1,Max!(Choice%)
510    INPUT #1,Max_graph_points%(Choice%)
520    FOR N% = 1 TO Max_graph_points%(Choice%)
530      INPUT #1,Xdata$,Ydata$
540      Graph!(Choice%,1,N%) = VAL(Xdata$)
550      Graph!(Choice%,2,N%) = VAL(Ydata$)
560    NEXT N%
570 END DO
600 Line% = 18 : GOSUB 20000
610 CLOSE #1
```

```
     1000 EXIT
    20000 FOR N% = Line% TO 24
    20010   LOCATE N%,1 : PRINT SPC(79)
    20020 NEXT N%
    20030 RETURN
    59999 EXIT
    60000 IF ERR > 999 THEN GOSUB 61000: RESUME,59999
    60005 Line% = 2 : GOSUB 20000
    60010 SET CURSOR 13,6
    60020 PRINT "The procedure LOADDATA is bombing.  This i
s error "ERR" from line "ERL
    60030 STOP
    61000 Line% = 19: GOSUB 20000
    61005 SET CURSOR 19,6 : PRINT "There is a file loading
error. Check you files"
    61010 FOR N% = -30000 TO 30000 : NEXT N%
    61020 Line% = 13: GOSUB 20000
    61030 RETURN
END PROCEDURE

PROCEDURE: INVERSEPOWER
   EXTERNAL: MINX!,MAXX!,MINY!,MAXY!,GRAPH!(),MAX_GRAPH_PO
INTS%()
   INTEGER: M%,Minloop%
      10 ON ERROR GOTO 50000
      20 Minloop% = 1
      84 DO
      85   IF Xory% = 2 THEN EXIT 1 LEVELS
      90   MINX! = 1.0E+37: MAXX! = -1.0E+37
      91 END DO
      92 DO
      93   IF Xory% = 1 THEN EXIT 1 LEVELS
      98   MINY! = 1.0E+37: MAXY! = -1.0E+37
      99 END DO
     100 FOR M% = Minloop% TO MAX_GRAPH_POINTS%(Graph1or2%
)
     110   GRAPH!(Graph1or2%,Xory%,M%) = 1 / (GRAPH!(Graph
1or2%,Xory%,M%) * Amount!)^Power!
     120   DO
     130     DO
     140       IF Xory% <> 1 THEN EXIT 1 LEVELS
     150       IF MINX! > GRAPH!(Graph1or2%,1,M%) THEN MIN
X! = GRAPH!(Graph1or2%,1,M%)
     160       IF MAXX! < GRAPH!(Graph1or2%,1,M%) THEN MAX
X! = GRAPH!(Graph1or2%,1,M%)
     170       EXIT 2 LEVELS
     180     END DO
     190     DO
     200       IF MINY! > GRAPH!(Graph1or2%,2,M%) THEN MIN
Y! = GRAPH!(Graph1or2%,2,M%)
     210       IF MAXY! < GRAPH!(Graph1or2%,2,M%) THEN MAX
```

```
Y! = GRAPH!(Graph1or2%,2,M%)
      220       END DO
      230     END DO
      240 NEXT M%
      250 EXIT
   50000 DO
   50010    IF ERR = 2 AND ERL = 110 THEN Minloop% = M% + 1
   50020    EXIT TO,100
   51000    PRINT "The procedure INVERSEPOWER is bombing.
This is error "ERR" from line "ERL"."
   51010    END
   60000 END DO
END PROCEDURE

PROCEDURE: MAINKEY
   REAL: I,L,N
   STRING ARRAY(10)[16]: KEYFUNC$,FUNC$
   REAL: O
   EXTERNAL: G1,G2,Clearscreen,TITLE,GRAPH_NAME$()
   EXTERNAL: MENU_CHOSEN%,GRAPH_CHOSEN%,GRAPH%,GFLAG
      10 Clearscreen(2):STATUSLINE OFF
      14 SET CURSOR 0,70:COLOR 2,7:PRINT GRAPH%:COLOR 7,0
      15 IF MENU_CHOSEN%=0 OR GRAPH_CHOSEN%=0 THEN GOTO 20
 ELSE TITLE(GRAPH_NAME$(MENU_CHOSEN%,GRAPH_CHOSEN%)):SET C
URSOR 0,70:COLOR 2,7:PRINT GFLAG:COLOR 7,0
      20 FOR I=1 TO 10
      30    IF COS(I*3.14159)<0 THEN L=2*I:N=1 ELSE N=14
      40    READ KEYFUNC$(I)
      50    READ FUNC$(I)
      60    KEY I,FUNC$(I)+CHR$(13)
      70    SET CURSOR L,N:COLOR 2,0:PRINT "F";I;" ":COLOR
2,7:SET CURSOR L,N+4:PRINT KEYFUNC$(I)
      80 NEXT I
      90 COLOR 2,0
      95 STOP
     100 DATA "SYSTEM ",SYSTEM,"SAVEDAT",EXIT TO 8555,"MOD
IFY ",MODIFY,"PLOTTER",EXIT TO 9000,"MENU   ",EXIT TO 200,
"LEAST  ",EXIT TO 6300,"LOAD G1",EXIT TO 1429,"LOAD G2",EX
IT TO 8700,"SCALE  ",EXIT TO 3000,"SCRNPLT",EXIT TO 4000
END PROCEDURE

PROCEDURE: PLOTTER
   REAL: I,L,N
   STRING ARRAY(10)[16]: KEYFUNC$,FUNC$
   EXTERNAL: Onoff,Clearscreen,MENU_CHOSEN%,GRAPH_CHOSEN%,
G1
   EXTERNAL: G2,TITLE,GRAPH_NAME$(),G
      10 Clearscreen(2)
      15 IF MENU_CHOSEN%=0 OR GRAPH_CHOSEN%=0 THEN GOTO 20
 ELSE TITLE(GRAPH_NAME$(MENU_CHOSEN%,GRAPH_CHOSEN%)):SET C
URSOR 0,70:COLOR 4,7:PRINT G:COLOR 7,0
```

```
      20 FOR I=1 TO 10
      30    IF COS(I*3.14159)<0 THEN L=2*I:N=1 ELSE N=14
      40    READ KEYFUNC$(I)
      50    READ FUNC$(I)
      60    SET CURSOR L,N:COLOR 4,0:PRINT "F";I;" ":COLOR
4,7:SET CURSOR L,N+4:PRINT KEYFUNC$(I)

      70    KEY I,FUNC$(I)+CHR$(13)
      80 NEXT I
      81 IF Onoff=1 THEN SET CURSOR 18,5:COLOR 0,7:PRINT K
EYFUNC$(9)
      90 COLOR 4,0
     100 STOP
     110 DATA "MAINKEY",MAINKEY,"SPEED  ",SPEED,"LINES  ",
LINES,"SYMBOLS",SYMBOLS,"COLORS ",COLORS,"PLOTAXS",EXIT TO
 9570,"PLOT G1",EXIT TO 9640,"PLOT G2",EXIT TO 9220,"ON/OF
F ",INFO,"CAPTION",CAPTION
END PROCEDURE

PROCEDURE: MODIFY
   REAL: I,L,N
   STRING ARRAY(10)[16]: KEYFUNC$
   EXTERNAL: Clearscreen,MENU_CHOSEN%,GRAPH_CHOSEN%,G1,G2,
TITLE
   EXTERNAL: GRAPH_NAME$()
   REAL: MD
   EXTERNAL: MOD
      10 Clearscreen(2)
      15 IF MENU_CHOSEN%=0 OR GRAPH_CHOSEN%=0 THEN GOTO 20
 ELSE TITLE(GRAPH_NAME$(MENU_CHOSEN%,GRAPH_CHOSEN%)):SET C
URSOR 0,70:COLOR 1,7:PRINT MOD:COLOR 7,0
      20 FOR I=1 TO 10
      30    IF COS(I*3.14159)<0 THEN L=2*I:N=1 ELSE N=14
      40    READ KEYFUNC$(I)
      50    KEY I,KEYFUNC$(I)+CHR$(13)
      60    SET CURSOR L,N:COLOR 1,0:PRINT "F";I;" ":COLOR
1,7:SET CURSOR L,N+4:PRINT KEYFUNC$(I)
      70 NEXT I
      80 COLOR 1,0
      90 STOP
     100 DATA "MAINKEY","ADDNUM ","MLTNUM ","POWER  ","NAT
LOG ","EXPNT  ","ABSLT  ","DELPT  ","ADDPT  ","CLRGRPH"
END PROCEDURE

PROCEDURE: COLORS
   EXTERNAL: PLOTTER
   REAL: I,J
   STRING: A$[16]
   REAL: N
   INTEGER ARRAY(5): EXIST
   EXTERNAL: Pen_selected%
```

```
    REAL ARRAY(?): PN
    EXTERNAL: PEN(),G1,G2,Clearscreen
    STRING: N$[?],J$[?]
        10 EXIST(1)=G1:EXIST(2)=G2
        11 EXIST(3)=0:EXIST(4)=0:EXIST(5)=0
        20 Clearscreen(2):SET CURSOR 5,1:COLOR 7,0:PRINT "EX
ISTING GRAPHS:"
        30 FOR I=1 TO 4
        40    IF EXIST(I)=0 THEN GOTO 60
        50    IF PEN(I)<>0 THEN SET CURSOR 5,2*I+18:COLOR PEN
(I)-2,7:PRINT EXIST(I) ELSE SET CURSOR 5,2*I+18:COLOR 0,7:
PRINT EXIST(I)
        60 NEXT I
        70 SET CURSOR 10,1:COLOR 7,0:PRINT "COLORS:"
        80 FOR I=1 TO 5
        90    SET CURSOR 10,2*I+8:COLOR I-1,7:PRINT I+1
       100 NEXT I
       110 SET CURSOR 15,1:COLOR 7,0:INPUT "WHICH GRAPH" N$
       111 N=VAL(N$)
       120 SET CURSOR 17,1:INPUT"WHICH COLOR" J$
       121 J=VAL(J$)
       125 IF N<1 OR N>2 THEN GOTO 135
       130 PEN(N)=J
       131 FOR I=1 TO 4
       132    IF EXIST(I)=0 THEN GOTO 134
       133    IF PEN(I)<>0 THEN SET CURSOR 5,2*I+18:COLOR PEN
(I)-2,7:PRINT EXIST(I) ELSE SET CURSOR 5,2*I+18:COLOR 0,7:
PRINT EXIST(I)
       134 NEXT I
       135 COLOR 7,0
       140 SET CURSOR 20,1:INPUT"FINISHED CHOOSING COLORS";A
$
       150 IF INSTR(UPPER$(A$),"Y") THEN GOTO 160 ELSE GOTO
20
       160 PLOTTER
       170 STOP
END PROCEDURE

PROCEDURE: SYMBOLS
    INTEGER ARRAY(10): EXIST,LINES
    EXTERNAL: PLOTTER
    REAL: I,J,N
    STRING: A$[16]
    REAL ARRAY(10): SYMBOLS
    EXTERNAL: Symbol$,SYMBL$()
    STRING ARRAY(5)[16]: SMBL$
    EXTERNAL: G1,G2,CLEARSCREEN
    STRING: N$[?],J$[?]
        10 EXIST(1)=G1:EXIST(2)=G2
        11 EXIST(3)=0:EXIST(4)=0:EXIST(5)=0
        20 CLEARSCREEN(2):SET CURSOR 5,1:COLOR 7,0:PRINT "EX
```

```
ISTING GRAPHS:"
      30 FOR I=1 TO 5
      40   IF EXIST(I)=0 THEN GOTO 60 ELSE SET CURSOR 5,2*
I+18:COLOR 0,7:PRINT EXIST(I)
      50   IF SYMBOLS(I)<>0 THEN SET CURSOR 5,2*I+18:COLOR
 SYMBOLS(I)+10,7:PRINT EXIST(I)
      60 NEXT I
      70 SET CURSOR 10,1:COLOR 7,0:PRINT "SYMBOLS: ":SET C
URSOR 10,10:COLOR 11,0:PRINT "1)NONE":SET CURSOR 10,18:COL
OR 12,0:PRINT "2) +  ":SET CURSOR 10,23:COLOR 13,0:PRINT "
3) *  ":SET CURSOR 10,28:COLOR 14,0:PRINT "4) @  "
      80 SET CURSOR 15,1:COLOR 7,0:INPUT"WHICH GRAPH" N$
      81 N=VAL(N$)
      85 SET CURSOR 17,1:INPUT"WHICH SYMBOL" J$
      86 J=VAL(J$)
      90 IF N<1 OR N>2 THEN GOTO 105
      91 IF J=1 THEN SYMBL$(N)=" "
      92 IF J=2 THEN SYMBL$(N)="+"
      93 IF J=3 THEN SYMBL$(N)="*"
      94 IF J=4 THEN SYMBL$(N)="@"
     100 SYMBOLS(N)=J
     101 FOR I=1 TO 5
     102   IF EXIST(I)=0 THEN GOTO 104 ELSE SET CURSOR 5,2
*I+18:COLOR 0,7:PRINT EXIST(I)
     103   IF SYMBOLS(I)<>0 THEN SET CURSOR 5,2*I+18:COLOR
 SYMBOLS(I)+10,7:PRINT EXIST(I)
     104 NEXT I
     105 COLOR 7,0
     110 SET CURSOR 20,1:INPUT"FINISHED CHOOSING SYMBOLS";
A$
     120 IF INSTR(UPPER$(A$),"Y") THEN GOTO 130 ELSE GOTO
20
     130 PLOTTER
     140 STOP
END PROCEDURE

PROCEDURE: LINES
   INTEGER ARRAY(10): EXIST,LINES
   EXTERNAL: PLOTTER
   REAL: I,J,N
   STRING: A$[16]
   EXTERNAL: Line_type$,LINETYPE$(),G1,G2,CLEARSCREEN
   STRING: N$[?],J$[?]
      10 EXIST(1)=G1:EXIST(2)=G2
      11 EXIST(3)=0:EXIST(4)=0:EXIST(5)=0
      20 CLEARSCREEN(2):SET CURSOR 5,1:COLOR 7,0:PRINT "EX
ISTING GRAPHS:"
      30 FOR I=1 TO 5
      40   IF EXIST(I)=0 THEN GOTO 60 ELSE SET CURSOR 5,2*
I+18:COLOR 0,7:PRINT EXIST(I)
      50   IF LINES(I)<>0 THEN SET CURSOR 5,2*I+18:COLOR L
```

```
INES(I)+1,7:PRINT EXIST(I)
      60 NEXT I
      70 SET CURSOR 10,1:COLOR 7,0:PRINT "LINES:  ":SET CU
RSOR 10,7:COLOR 2,0:PRINT "1)NONE":SET CURSOR 10,14:COLOR
3,0:PRINT "2)----":SET CURSOR 10,21:COLOR 4,0:PRINT "3)...
.":SET CURSOR 10,28:COLOR 5,0:PRINT "4)- - -"
      80 SET CURSOR 15,1:COLOR 7,0:INPUT"WHICH GRAPH" N$
      81 N=VAL(N$)
      85 SET CURSOR 17,1:INPUT"WHICH LINE" J$
      86 J=VAL(J$)
      90 IF N<1 OR N>2 THEN GOTO 105
      91 IF J=1 THEN LINETYPE$(N)="0"
      92 IF J=2 THEN LINETYPE$(N)=" "
      93 IF J=3 THEN LINETYPE$(N)="1"
      94 IF J=4 THEN LINETYPE$(N)="2"
     100 LINES(N)=J
     101 FOR I=1 TO 5
     102   IF EXIST(I)=0 THEN GOTO 104 ELSE SET CURSOR 5,2
*I+18:COLOR 0,7:PRINT EXIST(I)
     103   IF LINES(I)<>0 THEN SET CURSOR 5,2*I+18:COLOR L
INES(I)+1,7:PRINT EXIST(I)
     104 NEXT I
     105 COLOR 7,0
     110 SET CURSOR 20,1:INPUT"FINISHED CHOOSING LINES";A$
     120 IF INSTR(UPPER$(A$),"Y") THEN GOTO 130 ELSE GOTO
20
     130 PLOTTER
     140 STOP
END PROCEDURE

PROCEDURE: Info
   EXTERNAL: Onoff,Plotter
      10 IF Onoff=0 THEN Onoff=1 ELSE Onoff=0
      20 Plotter
      30 STOP
END PROCEDURE

PROCEDURE: CAPTION
   STRING: CPTION$[80]
   INTEGER: YPOINT
   EXTERNAL: PLOTTER,TIMEDELAY
   REAL: HELLO
       5 WIDTH 80:CLS
      10 SET CURSOR 15,5:COLOR 7,0:PRINT "ENTER THE PLOT C
APTION BELOW(80 CHARACTERS OR LESS)"
      20 SET CURSOR 18,1:COLOR 0,7:INPUT">";CPTION$
      30 CLOSE #3
      40 OPEN "COM1 : 9600,N,8,1,RS,CS65535,DS,CD" FOR OUT
PUT AS #3
      50 PRINT #3,"IN; "
      70 PRINT #3, "SP2 ; SI.15,.25 ; DR 0,1 ;"
```

```
      80 PRINT #3, "PA PU 7200,900 ; LB"CPTION$;CHR$(3)
      85 TIMEDELAY(15)
      86 PRINT #3, "SPO ;"
      90 CLOSE #3
      95 COLOR 7,0
     100 PLOTTER
     110 STOP
END PROCEDURE

PROCEDURE: Fixpoints
   INTEGER: O%
      10 ON Type% GOTO 100,200,300,400,500
     100 Plt! = 48
     110 IF Plt1! < 15 THEN Plt1! = 15 : EXIT
     120 IF Plt1! > 166 THEN Plt1! = 166 : EXIT
     198 EXIT
     199 STOP
     200 Plt! = 618
     210 IF Plt1! < 15 THEN Plt1! = 15 : EXIT
     220 IF Plt1! > 166 THEN Plt1! = 166 : EXIT
     298 EXIT
     299 STOP
     300 Plt1! = 166
     310 IF Plt! < 48 THEN Plt! = 48 : EXIT
     320 IF Plt! > 618 THEN Plt! = 618 : EXIT
     398 EXIT
     399 STOP
     400 Plt1! = 15
     410 IF Plt! < 48 THEN Plt! = 48 : EXIT
     420 IF Plt! > 618 THEN Plt! = 618 : EXIT
     500 EXIT
END PROCEDURE

PROCEDURE: Cleargraph
   EXTERNAL: Pulse$(),Hold$(),Rate$(),Slope$(),X_int$(),Y_
int$(),Range$(),Correlation$()
   EXTERNAL: Tyme$(),Temp$(),Grange$(),G1,G2,Comp_graph%
   REAL: I,N
   EXTERNAL: Bias$(),Max_graph_points%()
   EXTERNAL: Graph!()
   INTEGER: J
      10 Bias$(G1orG2) = "X"
      20 Pulse$(G1orG2) = "X"
      30 Hold$(G1orG2) = "X"
      40 Rate$(G1orG2) = "X"
      50 Slope$(G1orG2) = "X"
      60 X_int$(G1orG2) = "X"
      70 Y_int$(G1orG2) = "X"
      80 Range$(G1orG2)= "X"
      90 Correlation$(G1orG2) = "X"
     100 Tyme$(G1orG2) = "X"
```

```
      110 Temp$(G1orG2) = "X"
      120 Grange$(G1orG2) = "X"
      130 IF Max_graph_points%(G1orG2)>300 THEN J=Max_graph
_points%(G1orG2) ELSE J=300
      140 FOR I=1 TO 2
      150   FOR N=1 TO J
      160     Graph!(G1orG2,I,N)=0
      170   NEXT N
      180 NEXT I
      185 Max_graph_points%(G1orG2)=1
      190 IF G1orG2=1 THEN G1=0
      200 IF G1orG2=2 THEN G2=0:Comp_graph%=0
      210 EXIT
      220 STOP
END PROCEDURE

PROCEDURE: CLRGRPH
   EXTERNAL: G2,COMP_GRAPH%,MODIFY,Cleargraph,G1
   REAL ARRAY(10): Exist
   EXTERNAL: Clearscreen
   REAL: I
   STRING: N$[?]
   REAL: N
   STRING: A$[?]
   EXTERNAL: MOD
       10 Exist(1)=G1:Exist(2)=G2
       11 Exist(3)=0:Exist(4)=0:Exist(5)=0
       20 Clearscreen(2):SET CURSOR 5,1:COLOR 7,0:PRINT "EX
ISTING GRAPHS:"
       30 FOR I=1 TO 4
       40   IF Exist(I)=0 THEN GOTO 60
       50   SET CURSOR 5,2*I+18:COLOR 0,7:PRINT Exist(I)
       60 NEXT I
       70 SET CURSOR 15,1:COLOR 7,0:INPUT"CLEAR WHICH GRAPH
" N$
       80 N=VAL(N$):MOD=N
       81 IF N<1 OR N>2 THEN GOTO 100
       90 Cleargraph N
      100 Clearscreen(2):SET CURSOR 5,1:COLOR 7,0:PRINT "EX
ISTING GRAPHS:"
      105 Exist(1)=G1:Exist(2)=G2
      110 FOR I=1 TO 4
      120   IF Exist(I)=0 THEN GOTO 140
      130   SET CURSOR 5,2*I+18:COLOR 0,7:PRINT Exist(I)
      140 NEXT I
      150 SET CURSOR 18,1:COLOR 7,0:INPUT"FINISHED CLEARING
 GRAPHS" A$
      160 A$=UPPER$(A$)
      170 IF INSTR(A$,"Y")=0 THEN GOTO 10
      180 MODIFY
      190 STOP
```

```
END PROCEDURE

PROCEDURE: ADDNUM
   EXTERNAL: GRAPH!(),G1,G2,MAX_GRAPH_POINTS%()
   INTEGER ARRAY(8): EXIST
   STRING: W$[2],X$[2],NUM$[10],A$[2]
   EXTERNAL: CLEARSCREEN
   INTEGER: I,W
   EXTERNAL: MODIFY
   INTEGER: M
   REAL: NUM
   INTEGER: J
   EXTERNAL: MOD
      10 EXIST(1)=G1:EXIST(2)=G2
      11 EXIST(3)=0:EXIST(4)=0:EXIST(5)=0
      20 CLEARSCREEN(2):SET CURSOR 5,1:COLOR 7,0:PRINT "EX
ISTING GRAPHS:"
      30 FOR I=1 TO 4
      40   IF EXIST(I)=0 THEN GOTO 60
      50   SET CURSOR 5,2*I+18:COLOR 0,7:PRINT EXIST(I)
      60 NEXT I
      70 SET CURSOR 15,1:COLOR 7,0:INPUT"ADD TO WHICH GRAP
H" W$
      80 W=VAL(W$):MOD=W
      81 IF W<1 OR W>2 THEN GOTO 100
      82 SET CURSOR 17,1:COLOR 7,0:INPUT"ADD TO X OR Y" X$
      83 X$=UPPER$(X$)
      84 M=1
      85 IF INSTR(X$,"X")=0 THEN M=2
      87 SET CURSOR 19,1:COLOR 7,0:INPUT"ADD WHAT NUMBER"
NUM$
      88 NUM=VAL(NUM$)
      90 FOR J=1 TO MAX_GRAPH_POINTS%(W)
      92   GRAPH!(W,M,J)=GRAPH!(W,M,J)+NUM
      94 NEXT J
     100 REM
     150 SET CURSOR 21,1:COLOR 7,0:INPUT"FINISHED ADDING"
A$
     160 A$=UPPER$(A$)
     170 IF INSTR(A$,"Y")=0 THEN GOTO 10
     180 MODIFY
     190 STOP
END PROCEDURE

PROCEDURE: Mltnum
   EXTERNAL: GRAPH!(),G1,G2,MAX_GRAPH_POINTS%()
   INTEGER ARRAY(8): EXIST
   STRING: W$[2],X$[2],NUM$[10],A$[2]
   EXTERNAL: CLEARSCREEN
   INTEGER: I,W
   EXTERNAL: MODIFY
```

```
    INTEGER: M
    REAL: NUM
    INTEGER: J
    EXTERNAL: MOD
       10 EXIST(1)=G1:EXIST(2)=G2
       11 EXIST(3)=0:EXIST(4)=0:EXIST(5)=0
       20 CLEARSCREEN(2):SET CURSOR 5,1:COLOR 7,0:PRINT "EX
ISTING GRAPHS:"
       30 FOR I=1 TO 4
       40   IF EXIST(I)=0 THEN GOTO 60
       50   SET CURSOR 5,2*I+18:COLOR 0,7:PRINT EXIST(I)
       60 NEXT I
       70 SET CURSOR 15,1:COLOR 7,0:INPUT"MLTNUM WHICH GRAP
H" W$
       80 W=VAL(W$):MOD=W
       81 IF W<1 OR W>2 THEN GOTO 100
       82 SET CURSOR 17,1:COLOR 7,0:INPUT"MLT TO X OR Y" X$
       83 X$=UPPER$(X$)
       84 M=1
       85 IF INSTR(X$,"X")=0 THEN M=2
       87 SET CURSOR 19,1:COLOR 7,0:INPUT"MLT WHAT NUMBER"
NUM$
       88 NUM=VAL(NUM$)
       90 FOR J=1 TO MAX_GRAPH_POINTS%(W)
       92   GRAPH!(W,M,J)=GRAPH!(W,M,J)*NUM
       94 NEXT J
      100 REM
      150 SET CURSOR 21,1:COLOR 7,0:INPUT"FINISHED MULTIPLY
ING" A$
      160 A$=UPPER$(A$)
      170 IF INSTR(A$,"Y")=0 THEN GOTO 10
      180 MODIFY
      190 STOP
END PROCEDURE

PROCEDURE: POWER
    EXTERNAL: GRAPH!(),G1,G2,MAX_GRAPH_POINTS%()
    INTEGER ARRAY(8): EXIST
    STRING: W$[2],X$[2],NUM$[10],A$[2]
    EXTERNAL: CLEARSCREEN
    INTEGER: I,W
    EXTERNAL: MODIFY
    INTEGER: M
    REAL: NUM
    INTEGER: J
    EXTERNAL: MOD
       10 EXIST(1)=G1:EXIST(2)=G2
       11 EXIST(3)=0:EXIST(4)=0:EXIST(5)=0
       20 CLEARSCREEN(2):SET CURSOR 5,1:COLOR 7,0:PRINT "EX
ISTING GRAPHS:"
       30 FOR I=1 TO 4
```

```
      40   IF EXIST(I)=0 THEN GOTO 60
      50   SET CURSOR 5,2*I+18:COLOR 0,7:PRINT EXIST(I)
      60 NEXT I
      70 SET CURSOR 15,1:COLOR 7,0:INPUT"POWER WHICH GRAPH
" W$
      80 W=VAL(W$):MOD=W
      81 IF W<1 OR W>2 THEN GOTO 100
      82 SET CURSOR 17,1:COLOR 7,0:INPUT"POWER X OR Y" X$
      83 X$=UPPER$(X$)
      84 M=1
      85 IF INSTR(X$,"X")=0 THEN M=2
      87 SET CURSOR 19,1:COLOR 7,0:INPUT"WHAT POWER" NUM$
      88 NUM=VAL(NUM$)
      90 FOR J=1 TO MAX_GRAPH_POINTS%(W)
      92   GRAPH!(W,M,J)=GRAPH!(W,M,J)^NUM
      94 NEXT J
     100 REM
     150 SET CURSOR 21,1:COLOR 7,0:INPUT"FINISHED POWERING
" A$
     160 A$=UPPER$(A$)
     170 IF INSTR(A$,"Y")=0 THEN GOTO 10
     180 MODIFY
     190 STOP
END PROCEDURE

PROCEDURE: NATLOG
   EXTERNAL: GRAPH!(),G1,G2,MAX_GRAPH_POINTS%()
   INTEGER ARRAY(8): EXIST
   STRING: W$[2],X$[2],NUM$[10],A$[2]
   EXTERNAL: CLEARSCREEN
   INTEGER: I,W
   EXTERNAL: MODIFY
   INTEGER: M
   REAL: NUM
   INTEGER: J
   EXTERNAL: MOD
      10 EXIST(1)=G1:EXIST(2)=G2
      11 EXIST(3)=0:EXIST(4)=0:EXIST(5)=0
      20 CLEARSCREEN(2):SET CURSOR 5,1:COLOR 7,0:PRINT "EX
ISTING GRAPHS:"
      30 FOR I=1 TO 4
      40   IF EXIST(I)=0 THEN GOTO 60
      50   SET CURSOR 5,2*I+18:COLOR 0,7:PRINT EXIST(I)
      60 NEXT I
      70 SET CURSOR 15,1:COLOR 7,0:INPUT"NATLOG WHICH GRAP
H" W$
      80 W=VAL(W$):MOD=W
      81 IF W<1 OR W>2 THEN GOTO 100
      82 SET CURSOR 17,1:COLOR 7,0:INPUT"NATLOG X OR Y" X$
      83 X$=UPPER$(X$)
      84 M=1
```

```
      85 IF INSTR(X$,"X")=0 THEN M=2
      90 FOR J=1 TO MAX_GRAPH_POINTS%(W)
      92   GRAPH!(W,M,J)=LOG(GRAPH!(W,M,J))
      94 NEXT J
     100 REM
     150 SET CURSOR 21,1:COLOR 7,0:INPUT"FINISHED NATLOG"
A$
     160 A$=UPPER$(A$)
     170 IF INSTR(A$,"Y")=0 THEN GOTO 10
     180 MODIFY
     190 STOP
END PROCEDURE

PROCEDURE: EXPNT
   EXTERNAL: GRAPH!(),G1,G2,MAX_GRAPH_POINTS%()
   INTEGER ARRAY(8): EXIST
   STRING: W$[2],X$[2],NUM$[10],A$[2]
   EXTERNAL: CLEARSCREEN
   INTEGER: I,W
   EXTERNAL: MODIFY
   INTEGER: M
   REAL: NUM
   INTEGER: J
   EXTERNAL: MOD
      10 EXIST(1)=G1:EXIST(2)=G2
      11 EXIST(3)=0:EXIST(4)=0:EXIST(5)=0
      20 CLEARSCREEN(2):SET CURSOR 5,1:COLOR 7,0:PRINT "EX
ISTING GRAPHS:"
      30 FOR I=1 TO 4
      40   IF EXIST(I)=0 THEN GOTO 60
      50   SET CURSOR 5,2*I+18:COLOR 0,7:PRINT EXIST(I)
      60 NEXT I
      70 SET CURSOR 15,1:COLOR 7,0:INPUT"EXPNT WHICH GRAPH
" W$
      80 W=VAL(W$):MOD=W
      81 IF W<1 OR W>2 THEN GOTO 100
      82 SET CURSOR 17,1:COLOR 7,0:INPUT"EXPNT X OR Y" X$
      83 X$=UPPER$(X$)
      84 M=1
      85 IF INSTR(X$,"X")=0 THEN M=2
      90 FOR J=1 TO MAX_GRAPH_POINTS%(W)
      92   GRAPH!(W,M,J)=EXP(GRAPH!(W,M,J))
      94 NEXT J
     100 REM
     150 SET CURSOR 21,1:COLOR 7,0:INPUT"FINISHED EXPNT" A
$
     160 A$=UPPER$(A$)
     170 IF INSTR(A$,"Y")=0 THEN GOTO 10
     180 MODIFY
     190 STOP
END PROCEDURE
```

```
PROCEDURE: ABSLT
   EXTERNAL: GRAPH!(),G1,G2,MAX_GRAPH_POINTS%()
   INTEGER ARRAY(8): EXIST
   STRING: W$[2],X$[2],NUM$[10],A$[2]
   EXTERNAL: CLEARSCREEN
   INTEGER: I,W
   EXTERNAL: MODIFY
   INTEGER: M
   REAL: NUM
   INTEGER: J
   EXTERNAL: MOD
      10 EXIST(1)=G1:EXIST(2)=G2
      11 EXIST(3)=0:EXIST(4)=0:EXIST(5)=0
      20 CLEARSCREEN(2):SET CURSOR 5,1:COLOR 7,0:PRINT "EX
ISTING GRAPHS:"
      30 FOR I=1 TO 4
      40   IF EXIST(I)=0 THEN GOTO 60
      50   SET CURSOR 5,2*I+18:COLOR 0,7:PRINT EXIST(I)
      60 NEXT I
      70 SET CURSOR 15,1:COLOR 7,0:INPUT"ABSLT WHICH GRAPH
" W$
      80 W=VAL(W$):MOD=W
      81 IF W<1 OR W>2 THEN GOTO 100
      82 SET CURSOR 17,1:COLOR 7,0:INPUT"ABSLT X OR Y" X$
      83 X$=UPPER$(X$)
      84 M=1
      85 IF INSTR(X$,"X")=0 THEN M=2
      90 FOR J=1 TO MAX_GRAPH_POINTS%(W)
      92   GRAPH!(W,M,J)=ABS(GRAPH!(W,M,J))
      94 NEXT J
     100 REM
     150 SET CURSOR 21,1:COLOR 7,0:INPUT"FINISHED ABSLT" A
$
     160 A$=UPPER$(A$)
     170 IF INSTR(A$,"Y")=0 THEN GOTO 10
     180 MODIFY
     190 STOP
END PROCEDURE

PROCEDURE: DELPT
   EXTERNAL: GRAPH!(),G1,G2,MAX_GRAPH_POINTS%()
   INTEGER ARRAY(8): EXIST
   STRING: W$[2],X$[2],NUM$[10],A$[2]
   EXTERNAL: CLEARSCREEN
   INTEGER: I,W
   EXTERNAL: MODIFY
   INTEGER: M
   REAL: NUM
   INTEGER: J,X
   STRING: D$[4]
```

```
    INTEGER: D,N,FLAG
    STRING: DP$[?]
    INTEGER: DP
    REAL: MAIN
    STRING: Null$[?]
    EXTERNAL: MOD
       10 EXIST(1)=G1:EXIST(2)=G2
       11 EXIST(3)=0:EXIST(4)=0:EXIST(5)=0
       20 CLEARSCREEN(2):SET CURSOR 5,1:COLOR 7,0:PRINT "EX
ISTING GRAPHS:"
       30 FOR I=1 TO 4
       40    IF EXIST(I)=0 THEN GOTO 60
       50    SET CURSOR 5,2*I+18:COLOR 0,7:PRINT EXIST(I)
       60 NEXT I
       70 SET CURSOR 15,1:COLOR 7,0:INPUT"DELPT FROM WHICH
GRAPH" W$
       80 W=VAL(W$):MOD=W
       81 IF W<1 OR W>2 THEN GOTO 240
       90 N=-2
      100 M=1
      101 FOR J=3 TO 20
      102    SET CURSOR J,40:COLOR 2,0:PRINT "
                          "
      104 NEXT J
      110 FOR J=M TO MAX_GRAPH_POINTS%(W)
      120    SET CURSOR J-N,40:COLOR 2,0:PRINT J;".  ";GRAPH!
(W,1,J);",";GRAPH!(W,2,J)
      130    IF J-N=15 THEN EXIT TO,150
      135    FLAG=J
      140 NEXT J
      150 N=J-3:M=J
      160 SET CURSOR 17,1:COLOR 7,0:INPUT"DELETE WHICH POIN
T";D$
      170 D=VAL(D$)
      180 IF D=0 AND FLAG<>MAX_GRAPH_POINTS%(W) THEN GOTO 1
01
      190 IF D=0 THEN GOTO 240
      210 FOR J=D TO MAX_GRAPH_POINTS%(W)
      220    GRAPH!(W,1,J)=GRAPH!(W,1,J+1)
      221    GRAPH!(W,2,J)=GRAPH!(W,2,J+1)
      230 NEXT J
      231 MAX_GRAPH_POINTS%(W)=MAX_GRAPH_POINTS%(W)-1
      240 REM
      250 SET CURSOR 21,1:COLOR 7,0:INPUT"FINISHED DELPT" A
$
      260 A$=UPPER$(A$)
      270 IF INSTR(A$,"Y")=0 THEN GOTO 10
      280 MODIFY
      290 STOP
END PROCEDURE
```

```
PROCEDURE: ADDPT
   EXTERNAL: GRAPH!(),G1,G2,MAX_GRAPH_POINTS%()
   INTEGER ARRAY(8): EXIST
   STRING: W$[2],X$[2],NUM$[10],A$[2]
   EXTERNAL: CLEARSCREEN
   INTEGER: I,W
   EXTERNAL: MODIFY
   INTEGER: M
   REAL: NUM
   INTEGER: J,X
   STRING: AP$[4]
   INTEGER: AP,N,FLAG
   STRING: XAD$[10],YAD$[10]
   REAL: XAD,YAD
   EXTERNAL: MOD
      10 EXIST(1)=G1:EXIST(2)=G2
      11 EXIST(3)=0:EXIST(4)=0:EXIST(5)=0
      20 CLEARSCREEN(2):SET CURSOR 5,1:COLOR 7,0:PRINT "EX
ISTING GRAPHS:"
      30 FOR I=1 TO 4
      40   IF EXIST(I)=0 THEN GOTO 60
      50   SET CURSOR 5,2*I+18:COLOR 0,7:PRINT EXIST(I)
      60 NEXT I
      70 SET CURSOR 15,1:COLOR 7,0:INPUT"ADDPT TO WHICH GR
APH" W$
      80 W=VAL(W$):MOD=W
      81 IF W<1 OR W>2 THEN GOTO 240
      90 N=-2
     100 M=1
     101 FOR J=3 TO 20
     102   SET CURSOR J,40:COLOR 2,0:PRINT "
                         "
     104 NEXT J
     110 FOR J=M TO MAX_GRAPH_POINTS%(W)
     120   SET CURSOR J-N,40:COLOR 2,0:PRINT J;".  ";GRAPH!
(W,1,J);",";GRAPH!(W,2,J)
     130   IF J-N=15 THEN EXIT TO,150
     135   FLAG=J
     140 NEXT J
     150 N=J-3:M=J
     160 SET CURSOR 17,1:COLOR 7,0:INPUT"ADD BEFORE WHICH
POINT";AP$
     170 AP=VAL(AP$)
     180 IF AP=0 AND FLAG<>MAX_GRAPH_POINTS%(W) THEN GOTO
101
     190 IF AP=0 THEN GOTO 240
     191 SET CURSOR 19,1:COLOR 7,0:INPUT"ENTER NEW X,Y";XA
D$,YAD$
     210 FOR J=MAX_GRAPH_POINTS%(W) TO AP STEP -1
     220   GRAPH!(W,1,J+1)=GRAPH!(W,1,J)
     221   GRAPH!(W,2,J+1)=GRAPH!(W,2,J)
```

```
      230 NEXT J
      231 MAX_GRAPH_POINTS%(W)=MAX_GRAPH_POINTS%(W)+1
      232 GRAPH!(W,1,AP)=VAL(XAD$)
      233 GRAPH!(W,2,AP)=VAL(YAD$)
      240 REM
      250 SET CURSOR 21,1:COLOR 7,0:INPUT"FINISHED ADDPT" A
$
      260 A$=UPPER$(A$)
      270 IF INSTR(A$,"Y")=0 THEN GOTO 10
      280 MODIFY
      290 STOP
END PROCEDURE

PROCEDURE: SPEED
   EXTERNAL: SPEED$,PLOTTER
       10 CLS:COLOR 9,1:SET CURSOR 5,1
       20 INPUT"ENTER PLOTTER PEN SPEED .1(fast) to 2(slow)
";SPEED$
       30 COLOR 7,0:PLOTTER
       40 STOP
END PROCEDURE

'MAIN Program:


  100 DO
  110   SCREEN 0,0,0 : VIEW : CLS : STATUSLINE OFF : TROFF
  120   COLOR 2,1:CLS:SET CURSOR 13,30:PRINT "Now Entering
 Graphics"
  130   TIMEDELAY (3)
  140   COLOR 2,0,0
  145   CLS
  150 END DO
  155 SET CURSOR 0,74:COLOR 8,7:PRINT G1;G2:COLOR 7,0
  160 MAINKEY
  200 CLEAR:WIDTH 80
  210 ON ERROR GOTO 55000
  230 CHDIR "\"
  240 MinX! = 1E+19
  250 MaxX! = -1E+19
  260 MinY! = 1E+19
  270 MaxY! = -1E+19
  275 FOR N% = 1 TO 2
  280   Cleargraph N%
  385 NEXT N%
  390 Q! = 1.60218E-19
  400 K! = 1.38066E-23
  410 E_sub_O! = 8.85418E-14
  490 Minloop% = 1
  500 REM
  510 RESTORE,60200
```

```
 520 READ Max_graph_number%
 530 FOR N% = 1 TO 5
 540   FOR M% = 1 TO 24
 550     READ Graph_name$(N%,M%)
 560     IF (N%-1)*24 + M% => Max_graph_number% THEN EXIT
 2 LEVELS
 570   NEXT M%
 580   Maxloop%(N%) = 24
 590 NEXT N%
 600 Maxloop%(N%) = M% - 1
 605 IF M% = 24 THEN Maxloop%(N%) = 24
 610 IF N% = 6 THEN Next_screen% = 5 ELSE Next_screen% =
N%
 700 REM
 710 FOR N% = 1 TO Next_screen%
 720   TITLE ("Graphics Menu")
 730   FOR M% = 1 TO Maxloop%(N%)
 740     MENU (Graph_name$(N%,M%),M%,Rownumber%)
 750   NEXT M%
 760   DO
 770     IF N% = Next_screen% THEN EXIT
 780     IF N% > 1 THEN Col% = 5
 790     FINISH ("Go to Next Graphics Menu",Rownumber%,Co
l%,25):EXIT
 800   REPEAT
 810   DO
 820     IF N% < 2 THEN EXIT
 830     IF N% < Next_screen% THEN Col% = 42
 835     IF N% = Next_screen% THEN Col% = 5
 840     FINISH ("Go to Previous Graphics Menu",Rownumber
%,Col%,26):EXIT
 850   REPEAT
 860   DO
 870     IF N% < Next_screen% THEN EXIT
 875     Col% = 42
 880     FINISH ("Exit to DOS",Rownumber%,Col%,27):EXIT
 890   REPEAT
 895   BORDER (Rownumber%)
 900   REM
 910   DO
 920     ON N% GOTO 940,950,960,970
 930     CLS:PRINT "ERROR SAVE SCREEN ROUTINE":STOP
 940     SAVE SCREEN 0,0,24,79,MENU1:EXIT
 950     SAVE SCREEN 0,0,24,79,MENU2:EXIT
 960     SAVE SCREEN 0,0,24,79,MENU3:EXIT
 970     SAVE SCREEN 0,0,24,79,MENU4:EXIT
 990   REPEAT
1000   REM
1010   DO
1020     CLEARSCREEN (21)
1030     COLOR 6,0:SET CURSOR 21,5
```

```
 1040      INPUT "Enter the number to be executed";Choice%
 1050      IF Choice% > 0 AND Choice% <=Maxloop%(N%) THEN E
XIT 2 LEVELS
 1060      IF Choice% = 26 AND N% > 1 THEN EXIT TO,1100
 1070      IF Choice% = 25 AND N% < Next_screen% THEN EXIT
TO,1300
 1080      IF Choice% = 27 AND N% = Next_screen% THEN CLS:S
YSTEM
 1090    REPEAT
 1100    REM
 1110    DO
 1120      N% = N% - 1
 1130      ON N% GOTO 1150,1160,1170,1180
 1140      CLS:PRINT "ERROR IN RESTORE ROUTINE":STOP
 1150      RESTORE SCREEN 0,0,24,79,MENU1:EXIT
 1160      RESTORE SCREEN 0,0,24,79,MENU2:EXIT
 1170      RESTORE SCREEN 0,0,24,79,MENU3:EXIT
 1180      RESTORE SCREEN 0,0,24,79,MENU4:EXIT
 1200    REPEAT
 1210    GOTO 1000
 1300    REM
 1310 NEXT N%
 1320 Col% = CINT((80 - LEN(Graph_name$(N%,Choice%)))/2)
 1330 CLS:COLOR 3,0:SET CURSOR 0,Col%
 1340 PRINT Graph_name$(N%,Choice%)
 1350 Max_graph_points%(1) = 1
 1360 Max_graph_points%(2) = 1
 1400 REM
 1410 Menu_chosen% = N%
 1420 Graph_chosen% = Choice%
 1425 GOTO 160
 1429 Cleargraph 1:G1=1:Graph%=1
 1430 WIDTH 80:GFLAG=Graph%:TITLE(Graph_name$(Menu_chosen%
,Graph_chosen%)):ON Menu_chosen% GOTO 1500,1600,1700,1800,
1900
 1440 CLS:COLOR 7,0:PRINT "ERROR IN MENU_CHOSEN ROUTINE (1
420)":STOP
 1500 REM
 1510 ON Graph_chosen% GOSUB 20000,10000,10000,23000,23000
,17000,17000,23000,23000,23000,10000,11000,11000,11000,110
00,11000,11000,11000,11000,11000,15000,15000,15000,15000
 1520 CLS:PRINT "ERROR IN FIRST MENU ROUTINE (1500)":STOP
 1600 REM
 1610 ON Graph_chosen% GOSUB 23000,23000,11000,25000,2000,
2000,2000,2000,2000,2000,2000,2000
 1620 CLS:PRINT "ERROR IN SECOND MENU ROUTINE (1600)":STOP
 1700 REM
 1710 ON Graph_chosen% GOSUB 2000,2000,2000,2000,2000,2000
,2000,2000,2000,2000,2000,2000
 1720 CLS:PRINT "ERROR IN THIRD MENU ROUTINE (1700)":STOP
 1800 REM
```

```
 1810 ON Graph_chosen% GOSUB 2000,2000,2000,2000,2000,2000
,2000,2000,2000,2000,2000,2000
 1820 CLS:PRINT "ERROR IN FOURTH MENU ROUTINE (1800)":STOP
 1900 REM
 1910 ON Graph_chosen% GOSUB 2000,2000,2000,2000,2000,2000
,2000,2000,2000,2000,2000,2000
 1920 CLS:PRINT "ERROR IN FIFTH MENU ROUTINE (1900)":STOP
 2000 CLS:PRINT "THESE GRAPHS ARE NOT INSTALLED."
 2010 TIMEDELAY ()
 2020 GOTO 200
 2425 GRange$(Graph%) = STR$(Min!) + "," + STR$(Max!)
 2430 GOTO 160
 3000 WIDTH 80:Minloop%=1
 3002 IF G1=1 THEN DELTAMULTIPLY(1,1,1.0):DELTAMULTIPLY(1,
2,1.0)
 3003 IF G2=2 THEN DELTAMULTIPLY(2,1,1.0):DELTAMULTIPLY(2,
2,1.0)
 3007 CLOSE
 3010 DO
 3020   TITLE ("Extrema for X and Y axes")
 3030   COLOR 4,0:SET CURSOR 2,5
 3040   PRINT "Minimum X";SPC(10);"Maximum X";SPC(10);"Min
imum Y";SPC(10);"Maximum Y"
 3050   COLOR 3,0
 3060   SET CURSOR 3,7:PRINT MinX!
 3070   SET CURSOR 3,26:PRINT MaxX!
 3080   SET CURSOR 3,45:PRINT MinY!
 3090   SET CURSOR 3,64:PRINT MaxY!
 3100   COLOR 7,0:SET CURSOR 19,5
 3110   INPUT "Do you want to change any of the extrema (D
efaults to NO)";Check$
 3120   Check$ = UPPER$(Check$)
 3130   IF INSTR(Check$,"Y") = 0 THEN EXIT 1 LEVELS
 3140   CLEARSCREEN (19)
 3150   DO 1 TIMES
 3160     COLOR 6,0:SET CURSOR 9,5
 3170     PRINT "Enter the Minimum X value (Defaults to ";
MinX!;")";
 3180     INPUT Change$
 3190     IF Change$ <> "" THEN MinX! = VAL(Change$)
 3200     COLOR 6,0:SET CURSOR 11,5
 3210     PRINT "Enter the Maximum X value (Defaults to ";
MaxX!;")";
 3220     INPUT Change$
 3230     IF Change$ <> "" THEN MaxX! = VAL(Change$)
 3240     COLOR 6,0:SET CURSOR 13,5
 3250     PRINT "Enter the Minimum Y value (Defaults to ";
MinY!;")";
 3260     INPUT Change$
 3270     IF Change$ <> "" THEN MinY! = VAL(Change$)
 3280     COLOR 6,0:SET CURSOR 15,5
```

```
 3290      PRINT "Enter the Maximum Y value (Defaults to ";
MaxY!;")";
 3300      INPUT Change$
 3310      IF Change$ <> "" THEN MaxY! = VAL(Change$)
 3320    REPEAT
 3330 REPEAT
 3340 GOTO 160
 4000 REM
 4010 DO
 4020    CLS:COLOR 2,0:SET CURSOR 11,5
 4030    INPUT "Do you want a dot (1) or a line (2) graph";
Choice%
 4040    IF Choice% > 0 AND Choice% < 3 THEN EXIT
 4050 REPEAT
 4060 Graph_type% = Choice%
 4500 REM
 4510 DO 1 TIMES
 4520    COLOR 2,0
 4530    SCREEN 2
 4540    DRAW "BM200,100 U2 D4 U2 L2 R4"
 4550    GET (198,102) - (202,98),PLUS
 4560    CLS
 4570 REPEAT
 5000 REM
 5010 Col% = CINT((79 - LEN(Graph_name$(Menu_chosen%,Graph
_chosen%)))/2)
 5020 LOCATE 1,Col%:PRINT Graph_name$(Menu_chosen%,Graph_c
hosen%)
 5050 XTITLE (Xtitle$,MinX!,MaxX!)
 5060 YTITLE (Ytitle$,MinY!,MaxY!)
 5500 REM
 5510 DeltaX! = (570)/(MaxX! - MinX!)
 5520 DeltaY! = (150)/(MaxY! - MinY!)
 6000 REM
 6004 ERASE Plot!(),Skippoint%()
 6005 DIM Skippoint%(2,600)
 6010 FOR M% = 1 TO Max_graph_points%(1)
 6015    DO
 6017       IF MinX! > Graph!(1,1,M%) THEN Skippoint%(1,M%)
= 1 : EXIT 1 LEVELS
 6019       IF MaxX! < Graph!(1,1,M%) THEN Skippoint%(1,M%)
= 2 : EXIT 1 LEVELS
 6021       IF MinY! > Graph!(1,2,M%) THEN Skippoint%(1,M%)
= 3 : EXIT 1 LEVELS
 6023       IF MaxY! < Graph!(1,2,M%) THEN Skippoint%(1,M%)
= 4 : EXIT 1 LEVELS
 6025       X_plot! = DeltaX!*(Graph!(1,1,M%) - MinX!) +46
 6026       Y_plot! =-DeltaY!*(Graph!(1,2,M%) - MinY!) +165
 6030       PSET (X_plot!,Y_plot!)
 6031       Skippoint%(1,M%) = 5
 6035    END DO
```

```
 6040 NEXT M%
 6045 DO 1 TIMES
 6050   IF Comp_graph% = 0 THEN EXIT
 6120   FOR M% = Minloop% TO Max_graph_points%(2)
 6130     DO
 6140       IF MinX! > Graph!(2,1,M%) THEN Skippoint%(2,M%
) = 1 : EXIT 1 LEVELS
 6145       IF MaxX! < Graph!(2,1,M%) THEN Skippoint%(2,M%
) = 2 : EXIT 1 LEVELS
 6150       IF MinY! > Graph!(2,2,M%) THEN Skippoint%(2,M%
) = 3 : EXIT 1 LEVELS
 6155       IF MaxY! < Graph!(2,2,M%) THEN Skippoint%(2,M%
) = 4 : EXIT 1 LEVELS
 6160       X_plot! = DeltaX!*(Graph!(2,1,M%) - MinX!) +46
 6170       Y_plot! =-DeltaY!*(Graph!(2,2,M%) - MinY!) +16
5
 6180       PUT (X_plot!,Y_plot!),PLUS,OR
 6190       Skippoint%(2,M%) = 5
 6200     END DO
 6208   NEXT M%
 6209 REPEAT
 6210 DIM Plot!(2,2)
 6212 DO
 6213   IF Graph_type% <> 2 THEN EXIT 1 LEVELS
 6214   FOR N% = 1 TO 2
 6215     IF N% = 2 AND Comp_graph% <> 1 THEN EXIT 2 LEVEL
S
 6216     FOR M% = 1 TO Max_graph_points%(N%) - 1
 6217       DO
 6218         FOR O% = 1 TO 2
 6219           Plot!(O%,1)= DeltaX!*(Graph!(N%,1,M%+O%-1)
 - MinX!) +46
 6220           Plot!(O%,2)=-DeltaY!*(Graph!(N%,2,M%+O%-1)
 - MinY!) +165
 6224           Fixpoints Skippoint%(N%,M%+O%-1),0,Plot!(O
%,1),Plot!(O%,2)
 6233         NEXT O%
 6235         IF N% = 2 AND (Skippoint%(1,M%) = 5 OR Skipp
oint%(2,M%) = 5) THEN Plot!(1,1) = Plot!(1,1)+ 2:Plot!(1,2
)=Plot!(1,2)+2:Plot!(2,1)= Plot!(2,1)+2:Plot!(2,2)=Plot!(2
,2)+2
 6238         LINE (Plot!(1,1),Plot!(1,2)) - (Plot!(2,1),P
lot!(2,2))
 6240       END DO
 6242     NEXT M%
 6244   NEXT N%
 6245 END DO
 6248 DO
 6249   TIMEDELAY(2)
 6250   A$ = INKEY$ : IF INKEY$ = "" THEN GOTO 6250
 6260   SCREEN 0,0,0 : VIEW : CLS
```

```
 6265   ERASE Skippoint%(),Plot!()
 6270 END DO
 6275 GOTO 160
 6300 WIDTH 80:TITLE ( Graph_name$(Menu_chosen%,Graph_chos
en%))
 6310 Fit$=UPPER$("Y")
 7550 Error_LS%=0
 7560 FOR M% = 1 TO 2
 7562   DO 1 TIMES
 7565     DO
 7568       IF INSTR(Fit$,"Y")=0 THEN EXIT 1 LEVELS
 7569       CLEAR (Slope!(M%),Correlation!(M%),X_int!(M%),
Y_int!(M%))
 7570       LEASTSQUARES (M%,Min!(M%),Max!(M%),Slope!(M%),
Correlation!(M%),X_int!(M%),Y_int!(M%),M%-1,Error_LS%)
 7572       IF Error_LS% <> 0 THEN Fit$ = "END"
 7575     END DO
 7640   REPEAT
 7650   DO
 7655     IF Error_LS% = 1 THEN EXIT TO,160
 7660     IF INSTR(Fit$,"Y") = 0 THEN EXIT
 7670     DO 1 TIMES
 7680       IF M% = 2 THEN EXIT
 7690       CLS : TITLE ("Least Squares Fit")
 7694       COLOR 2,0,0
 7700       SET CURSOR 8,0:PRINT "X - Intercept"
 7710       SET CURSOR 10,0:PRINT "Y - Intercept"
 7720       SET CURSOR 12,0:PRINT "Slope"
 7730       SET CURSOR 14,0:PRINT "Correlation"
 7740       SET CURSOR 16,0:PRINT "Range"
 7750     REPEAT
 7760     DO
 7790       IF M% =1 THEN Col% = 20
 7800       IF M% = 2 THEN Col% = 50
 7810       SET CURSOR 6,Col%
 7820       IF M% = 1 THEN PRINT ". CURVE 1" ELSE PRINT "+
 CURVE 2"
 7830       FOR N% = 1 TO 6
 7840         SET CURSOR N%*2 + 6,Col%
 7850         DO
 7860           ON N% GOTO 7870,7880,7890,7900,7910,7920
 7870           Dummy! = X_int!(M%):EXIT 1 LEVELS
 7880           Dummy! = Y_int!(M%):EXIT 1 LEVELS
 7890           Dummy! = Slope!(M%):EXIT 1 LEVELS
 7900           Dummy! = Correlation!(M%):EXIT 1 LEVELS
 7910           Dummy! = Min!(M%):EXIT 1 LEVELS
 7920           SET CURSOR 16,Col% + 8:PRINT ","
 7930           SET CURSOR 16,Col% + 10:Dummy! = Max!(M%):
EXIT 1 LEVELS
 7940         REPEAT
 7945         DO 1 TIMES
```

```
7950              IF Dummy! => .01 AND Dummy! <= 1000 THEN P
RINT USING ".";Dummy!:EXIT
7960              IF Dummy!<= -.01 AND Dummy! >= -1000 THEN
PRINT USING ".";Dummy!:EXIT
7970              IF Dummy! = 0 THEN PRINT "0" ELSE PRINT US
ING "#.###^^^^";Dummy!:EXIT
7975            REPEAT
7980          NEXT N%
7990        END DO
7995      END DO
8000 NEXT M%
8015 DO 1 TIMES
8016   IF INSTR(Fit$,"Y") = 0 THEN EXIT
8020   TIMEDELAY (4)
8030   SET CURSOR 21,5:COLOR 7,0:PRINT "HIT ANY KEY TO CO
NTINUE"
8040   A$ = INKEY$:IF A$ = "" THEN GOTO 8040
8050 REPEAT
8060 IF X_int!(1) <> 0 OR X_int!(2) <> 0 THEN Fit$ = "Y"
8065 GOTO 160
8555 DO
8560   CLEARSCREEN(2) : SET CURSOR 13,6
8590   SAVEDATA(Comp_graph%+1)
8595 END DO
8600 GOTO 160
8700 Cleargraph 2:G2=2:Graph%=2:Comp_graph%=1
8796 CLS:COLOR 3,0:SET CURSOR 0,CINT((80 - LEN(Graph_name
$(Menu_chosen%,Graph_chosen%)))/2)
8797 PRINT Graph_name$(Menu_chosen%,Graph_chosen%)
8800 EXIT TO,1430
9000 REM
9001 TITLE (Graph_name$(Menu_chosen%,Graph_chosen%))
9040 DO 1 TIMES
9050   IF Info_pointer% < 5 THEN EXIT 1 LEVELS
9060   COLOR 7,0 : SET CURSOR 13,6
9070   PRINT "WARNING!  If you plan on plotting the anoth
er graph, the information block"
9075   SET CURSOR 14,6 : PRINT "will not be printed!"
9080   TIMEDELAY (5)
9090 REPEAT
9095 PLOTTER
9220 DO
9230   IF INSTR(Fit$,"Y") = 0 THEN EXIT 1 LEVELS
9240   Slope$(2) = STR$(Slope!(2))
9250   X_int$(2) = STR$(X_int!(2))
9260   Y_int$(2) = STR$(Y_int!(2))
9265   Range$(2) = STR$(Min!(2)) + "," +STR$(Max!(2))
9266   Correlation$(2) = STR$(Correlation!(2))
9267 END DO
9350 G=G2:Print_out%=1
9360 PLOTTING_POINTS(Symbol$,2,1)
```

```
 9370 EXIT TO,9999
 9560 GOTO 9000
 9570 DO
 9580   Title$ = UPPER$(Graph_name$(Menu_chosen%,Graph_cho
sen%))
 9590   PLOTTERAXES (UPPER$(Graph_name$(Menu_chosen%,Graph
_chosen%)))
 9600 END DO
 9601 GOTO 9000
 9640 DO
 9650   IF INSTR(Fit$,"Y") = 0 THEN EXIT 1 LEVELS
 9660   Slope$(1) = STR$(Slope!(1))
 9670   X_int$(1) = STR$(X_int!(1))
 9680   Y_int$(1) = STR$(Y_int!(1))
 9690   Range$(1) = STR$(Min!(1)) + "," + STR$(Max!(1))
 9695   Correlation$(1) = STR$(Correlation!(1))
 9696 END DO
 9700 DO
 9701   Temp$(1) = " " + Temp$(1)
 9702   Tyme$(1) =  " " + Tyme$(1)
 9703   Bias$(1) = " " + Bias$(1)
 9704   Pulse$(1) = " " + Pulse$(1)
 9705   Hold$(1) = " " + Hold$(1)
 9706   Rate$(1) = " " + Rate$(1)
 9707   GRange$(1) = " " + GRange$(1)
 9708   G=G1
 9710   PLOTTING_POINTS(Symbol$,1,0)
 9715   Print_out% = 1
 9720 END DO
 9999 GOTO 9000
10000 REM
10007 Error% = 0
10010 DO
10020   IF Graph_chosen% = 11 THEN EXIT 1 LEVELS
10030   Directory$(Graph%,2) = "NO"
10040 END DO
10400 DO
10410   IF Graph_chosen% = 3 THEN File$ = "\G-T" ELSE File
$ = "\C-T"
10415   IF Graph_chosen% = 11 THEN ID% = 0 ELSE ID% = 1
10420   DIRECTORY (File$,ID%,Graph%,Error%)
10430   IF Error% <> 0 THEN EXIT TO,160
10434   DO 1 TIMES
10435     IF Graph_chosen% = 11 THEN ID% = 1 ELSE ID% = 0
10440     RUNTYPE(Graph%,ID%)
10445     DO
10446       IF Graph_chosen% <> 11 THEN EXIT 1 LEVELS
10447       IF INSTR(Temp_type$(Graph%),"T") = 0 AND INSTR
(Temp_type$(Graph%),"R") = 0 THEN EXIT 1 LEVELS
10449       CLEARSCREEN(2)
10451       SET CURSOR 13,6 : COLOR 7,0
```

```
10453        PRINT "The DLTS Spectrum cannot be done with e
ither a TIME or ROOM TEMPERATURE run."
10455        TIMEDELAY()
10457        EXIT TO,10400
10459      REPEAT
10460    REPEAT
10465    IF Graph_chosen% <> 11 THEN ID% = 1 ELSE ID% = 4
10470    PARAMETERSET (ID%)
10480    IF Graph_chosen% = 11 THEN ID% = 8 ELSE ID% = 1
10485    DATAPRINT
10490    FOR N% = 1 TO 2
10500      DO
10510        IF INSTR(Directory$(Graph%,N%),"NO") <> 0 THEN
 EXIT 1 LEVELS
10520        DATARETRIEVAL (ID%,Graph%,N%,Error%,0)
10525        IF Error% <> 0 THEN EXIT TO,10400
10530      END DO
10540    NEXT N%
10550  END DO
10600  DO
10610    DO
10620      IF Graph_chosen% <> 2 THEN EXIT 1 LEVELS
10630      DELTAMULTIPLY (Graph%,2,1.0E+09)
10640      Ytitle$ = "Capacitance in nF"
10650      DELTAMULTIPLY (Graph%,1,1000.0)
10660      Xtitle$ = "Time in milliSeconds"
10665      IF INSTR(Temp_type$(Graph%),"T") = 0 THEN Temp$(
Graph%) = STR$(Temp_sought!)
10666      IF INSTR(Temp_type$(Graph%),"T") <> 0 THEN Tyme$
(Graph%) = STR$(Time_sought!)
10667      Bias$(Graph%) = MID$(Bias_found$,INSTR(Bias_foun
d$,"=") + 2)
10668      Pulse$(Graph%) = MID$(Pulse_found$,INSTR(Pulse_f
ound$,"=") + 2)
10669      Hold$(Graph%) = MID$(Hold_found$,INSTR(Hold_foun
d$,"=") + 2)
10670      EXIT 2 LEVELS
10680    END DO
10690    DO
10700      IF Graph_chosen% <> 3 THEN EXIT 1 LEVELS
10710      DELTAMULTIPLY (Graph%,2,1000.0)
10720      Ytitle$ = "Conductance in mMhos"
10730      DELTAMULTIPLY (Graph%,1,1000.0)
10740      Xtitle$ = "Time in milliSeconds"
10741      IF INSTR(Temp_type$(Graph%),"T") = 0 THEN Temp$(
Graph%) = STR$(Temp_sought!)
10742      IF INSTR(Temp_type$(Graph%),"T") <> 0 THEN Tyme$
(Graph%) = STR$(Time_sought!)
10743      Bias$(Graph%) = MID$(Bias_found$,INSTR(Bias_foun
d$,"=") + 1)
10744      Pulse$(Graph%) = MID$(Pulse_found$,INSTR(Pulse_f
```

```
ound$,"=") + 1)
10745       Hold$(Graph%) = MID$(Hold_found$,INSTR(Hold_foun
d$,"=") + 1)
10750       EXIT 2 LEVELS
10760     END DO
10770     DO
10780       IF Graph_chosen% <> 11 THEN EXIT 1 LEVELS
10790       DELTAMULTIPLY (Graph%,2,1E+12)
10800       Ytitle$ = "DLTS in pF"
10810       Xtitle$ = "Temperature in Kelvin"
10822       Bias$(Graph%) = MID$(Bias_found$,INSTR(Bias_foun
d$,"=") + 1)
10823       Pulse$(Graph%) = MID$(Pulse_found$,INSTR(Pulse_f
ound$,"=") + 1)
10824       Hold$(Graph%) = MID$(Hold_found$,INSTR(Hold_foun
d$,"=") + 1)
10825       Rate$(Graph%) = STR$(LOG(Mintyme! / Maxtyme!) /
(Mintyme! - Maxtyme!))
10830       EXIT 2 LEVELS
10840     END DO
10980 END DO
10985 CLOSE
10990 RETURN,160
11000 REM
11010 Counter% = 0
11030 File% = 1
11500 DO
11510   IF Graph_chosen% = 13 OR Graph_chosen% = 14 THEN F
ile$ = "\CV" ELSE File$ = "\IV"
11515   IF Graph_chosen% > 16 AND Graph_chosen% < 25 THEN
ID% = 0 ELSE ID% = 1
11517   IF Graph_chosen% = 3 THEN ID% = 0
11520   DIRECTORY(File$,ID%,Graph%,Error%)
11530   IF Error% <> 0 THEN EXIT TO,160
11540 END DO
11550 DO
11560   IF Graph_chosen% > 16 OR Graph_chosen% = 3 THEN ID
% = 1 ELSE ID% = 0
11570   RUNTYPE(Graph%,ID%)
11574   DO
11575     IF INSTR(Temp_type$(Graph%),"R") = 0 OR Graph_ch
osen% <> 17 THEN  EXIT 1 LEVELS
11576     CLEARSCREEN(2)
11577     COLOR 7,0,0 : SET CURSOR 13,6
11578     PRINT "You cannot do the chosen graph with a roo
m temperature run."
11579     TIMEDELAY(5)
11580     EXIT TO,11500
11581   REPEAT
11589 END DO
11590 DO
```

```
11600    IF Graph_chosen% <> 20 THEN EXIT 1 LEVELS
11610    CLEARSCREEN(2)
11620    SET CURSOR 13,6 : COLOR 2,0
11630    INPUT "Enter the AE*A  ",Dummy$
11640    AeA! = VAL(Dummy$)
11650    IF AeA! <= 0 THEN EXIT TO,11590
11660 END DO
11670 DO
11680    IF Graph_chosen% <> 17 THEN EXIT 1 LEVELS
11685    CLEARSCREEN(2)
11690    SET CURSOR 13,6 : COLOR 2,0,0
11700    INPUT "Enter the forward bias.  ",Dummy$
11710    Vf! = VAL(Dummy$)
11730 END DO
11740 DATAPRINT
11885 DO
11886    IF Graph_chosen% < 17 AND Graph_chosen% <> 3 THEN
EXIT 1 LEVELS
11887    IF INSTR(Temp_type$(Graph%),"T") <> 0 THEN Run_typ
e% = 2 ELSE Run_type% = 1
11888    IF INSTR(Temp_type$(Graph%),"T") = 0 THEN Temp_typ
e$(Graph%) = "R"
11889    EXIT TO,11900
11890 REPEAT
11891 Run_type% = 0
11900 DO
11910    IF Graph_chosen% = 13 OR Graph_chosen% = 14 THEN I
D% = 7 ELSE ID% = 6
11930    DATARETRIEVAL (ID%,Graph%,File%,Error%,Run_type%)
11942    IF Error% = 1 AND Graph!(Graph%,2,2) <> 0 AND Grap
h_chosen% = 16 THEN EXIT 1 LEVELS
11943    IF Error% = 1 AND Graph!(Graph%,2,2) <> 0 AND Grap
h_chosen% = 12 THEN EXIT 1 LEVELS
11944    IF Error% = 1 AND Graph!(Graph%,2,2) <> 0 AND Grap
h_chosen% = 15 THEN EXIT 1 LEVELS
11945    IF Error% <> 0 AND Graph_chosen% < 17 AND Graph_ch
osen% <> 3 THEN EXIT TO,11500
11950 END DO
12000 DO
12010    DO
12020      IF Graph_chosen% <> 12 THEN EXIT 1 LEVELS
12030      DELTAMULTIPLY(Graph%,2,1000.0)
12040      Xtitle$ = "Voltage in Volts"
12050      Ytitle$ = "Current in mA"
12060      EXIT 2 LEVELS
12070    REPEAT
12080    DO
12090      DO
12095        DO
12100          IF Graph_chosen% <> 13 THEN EXIT 1 LEVELS
12110          DELTAMULTIPLY(Graph%,2,1E+09)
```

```
12120          Ytitle$ = "Cap in nF"
12130          EXIT 2 LEVELS
12140        REPEAT
12150        DO
12160          IF Graph_chosen% <> 14 THEN EXIT 3 LEVELS
12170          INVERSEPOWER(Graph%,2,1E+09,2.000)
12180          Ytitle$ = "1/C**2  C in nF"
12190        END DO
12195      END DO
12210      Xtitle$ = "Voltage in Volts"
12220      EXIT 2 LEVELS
12230    REPEAT
12240    DO
12250      IF Graph_chosen% > 16 OR Graph_chosen% = 3 THEN
EXIT 1 LEVELS
12260      DELTAln(Graph%,2,1.0)
12270      Ytitle$ = "log(I)  I in A"
12280      DO
12290        IF Graph_chosen% <> 15 THEN EXIT 1 LEVELS
12300        Xtitle$ = "Voltage in Volts"
12305        DELTAMULTIPLY(Graph%,2,1/LOG(10))
12310        EXIT 3 LEVELS
12320      REPEAT
12330      DO
12340        Position% = INSTR(Temp_found$,";") + 1
12350        Temp! = VAL(MID$(Temp_found$,Position%))
12360        DELTAMULTIPLY(Graph%,1,(Q!/(K! * Temp!)))
12370        Xtitle$ = "QV / kT"
12375        IF Graph_chosen% = 17 THEN EXIT 1 LEVELS
12380        EXIT 3 LEVELS
12390      REPEAT
12400    END DO
12410    DO
12420      IF Graph_chosen% <> 17 THEN EXIT 1 LEVELS
12430      DO
12440        Position% = INSTR(Temp_found$,";") + 1
12450        Temp! = VAL(MID$(Temp_found$,Position%))
12460      END DO
12470      Counter% = Counter% + 1
12480      DO
12490        FOR M% = 1 TO Max_graph_points%(Graph%)
12500          IF Graph!(Graph%,1,M%) = Vf! THEN EXIT 2 LEV
ELS
12510        NEXT M%
12515        IF Counter% > 1 THEN EXIT TO,12630
12520        CLEARSCREEN(2)
12530        COLOR 7,0 : SET CURSOR 13,6
12540        PRINT "The Forward bias you specified was not
found"
12550        TIMEDELAY (5)
12560        EXIT TO,11670
```

```
12570       REPEAT
12580       If! = Graph!(Graph%,2,M%)
12590       X! = (Q!) / (K! * Temp!)
12600       Y! = LOG (ABS(If!) / Temp!^2.0)
12610       OPEN "\BUCKET" FOR APPEND AS #3
12620       PRINT #3,X! "," Y!
12630       CLOSE #File% : CLOSE #3
12632       IF INSTR(SOURCE$,"B")=0 THEN OPEN "\DATA\"+ Dire
ctory$(Graph%,File%) + File$ FOR INPUT AS #File% ELSE OPEN
 "B:" + File$ +"."+Directory$(Graph%,File%) FOR INPUT AS #
File%
12635       Max_graph_points%(Graph%) = 1
12637       IF INSTR(Temp_type$(Graph%),"T") = 0 THEN Temp_t
ype$(Graph%) = "L"
12640       IF Error% = 0 THEN EXIT TO,11900
12642       DO
12643         CLOSE #File%
12644         File% = File% + 1
12645         IF File% = 3 OR INSTR(Directory$(Graph%,2),"NO
") <> 0 THEN EXIT 1 LEVELS
12646         IF INSTR(Temp_type$(Graph%),"T") = 0 THEN Temp
_type$(Graph%) = "R"
12647         EXIT TO,11900
12648       REPEAT
12650       DO
12670         Max_graph_points%(Graph%) = Counter%
12680         GETDATA(Graph%)
12710         Ytitle$ = "Ln (If / T**2) If in A"
12720         Xtitle$ = "q / KT"
12725         Graph_name$(1,17) = "Activation Energy"
12730         Bias$(Graph%) = STR$(Vf!)
12740         EXIT 3 LEVELS
12750       REPEAT
12760     REPEAT
12770     DO
12775       DO
12777         IF Graph_chosen% = 3 THEN EXIT 1 LEVELS
12780         IF Graph_chosen% < 18 OR Graph_chosen% > 20 TH
EN EXIT 2 LEVELS
12785       END DO
12790       Counter% = Counter% + 1
12800       DO
12810         Position% = INSTR(Temp_found$,";") + 1
12820         Temp! = VAL(MID$(Temp_found$,Position%))
12830         DELTAln (Graph%,2,1.0)
12840         DELTAMULTIPLY (Graph%,1,(Q!/(K! * Temp!)))
12850         IF Counter% = 1 THEN ID% = 0 ELSE ID% = 1
12860         LEASTSQUARES (Graph%,Min!,Max!,Slope!,Dummy!,D
ummy!,Y_int!,ID%,Error_LS%)
12862         IF Error_LS% = 1 THEN EXIT TO,160
12865         GRange$(Graph%) = STR$(Min!) + "," + STR$(Max!
```

```
)
12870      END DO
12880      DO
12890        DO
12900          IF Graph_chosen% <> 18 AND Graph_chosen% <>
3 THEN EXIT 1 LEVELS
12905          DO
12906            IF Graph_chosen% = 3 THEN EXIT 1 LEVELS
12907            IF Y_int! < -39 OR Y_int! > 1 THEN Counter
% = Counter% - 1 : EXIT TO,13313
12910            Y! = EXP(Y_int!)
12920            Ytitle$ = "Isat in Amps"
12924            Graph_name$(1,18) = "I Saturation"
12925          END DO
12926          DO
12927            IF Graph_chosen% <> 3 THEN EXIT 1 LEVELS
12928            Y! = Y_int!
12929            Ytitle$ = "ln(Isat) in Amps"
12930            Graph_name$(2,3) = "ln(Isat)"
12935          END DO
12936          EXIT 2 LEVELS
12940        REPEAT
12950        DO
12960          IF Graph_chosen% <> 19 THEN EXIT 1 LEVELS
12970          Y! = 1/Slope!
12980          Ytitle$ = "Ideality Factor"
12985          Graph_name$(1,19) = "Ideality Factor n"
12990          EXIT 2 LEVELS
13000        REPEAT
13010        DO
13020          IF Graph_chosen% <> 20 THEN EXIT 1 LEVELS
13025          IF Y_int! < -39 OR Y_int! > 1 THEN Counter%
= Counter% - 1 : EXIT TO,13313
13030          Y! = (K! * Temp!/ Q!) * LOG (AeA! * Temp!^2.
0/EXP(Y_int!))
13040          Ytitle$ = "Barrier in Volts"
13045          Graph_name$(1,20) = "Barrier Height (Current
)"
13050          EXIT 2 LEVELS
13060        REPEAT
13070      END DO
13150      DO
13160        DO
13170          IF INSTR(Temp_type$(Graph%), "T") <> 0 THEN
EXIT 1 LEVELS
13180          X! = Temp!
13190          Xtitle$ = "Temperature in Kelvin"
13200          EXIT 2 LEVELS
13210        REPEAT
13220        DO
13230          Position% = INSTR(Time_found$,"=") + 1
```

```
13240           X! = VAL (MID$(Time_found$,Position%))
13250           Xtitle$ = "Time in Seconds"
13260         END DO
13270       END DO
13280       DO
13290         OPEN "\BUCKET" FOR APPEND AS #3
13300         PRINT #3, X! "," Y!
13310         CLOSE #File% : CLOSE #3
13312       END DO
13313       DO
13315         Max_graph_points%(Graph%) = 1
13316         CLOSE #File%
13317         IF INSTR(SOURCE$,"B")=0 THEN OPEN "\DATA\"+ Di
rectory$(Graph%,File%) + File$ FOR INPUT AS #File% ELSE OP
EN "B:" + File$ +"."+Directory$(Graph%,File%) FOR INPUT AS
 #File%
13318         IF INSTR(Temp_type$(Graph%),"T") = 0 THEN Temp
_type$(Graph%) = "L"
13320         IF Error% = 0 THEN EXIT TO,11900
13330       END DO
13332       DO
13333         CLOSE #File%
13334         File% = File% + 1
13335         IF File% = 3 OR INSTR(Directory$(Graph%,2),"NO
") <> 0 THEN EXIT 1 LEVELS
13336         IF INSTR(Temp_type$(Graph%),"T") = 0 THEN Temp
_type$(Graph%) = "R"
13337         EXIT TO,11900
13338       REPEAT
13340       DO
13350         Max_graph_points%(Graph%) = Counter%
13360         GETDATA(Graph%)
13410       END DO
13420       EXIT 2 LEVELS
13425     END DO
13437 END DO
13440 DO
13442     IF Graph_chosen% > 17 AND Graph_chosen% < 21 OR Gr
aph_chosen% = 3 THEN EXIT 1 LEVELS
13445     DO
13450       IF INSTR(Temp_type$(Graph%),"T") <> 0 THEN EXIT
1 LEVELS
13460       Position% = INSTR(Temp_found$,";") + 1
13470       Temp$(Graph%) = MID$(Temp_found$,Position%)
13480       EXIT 2 LEVELS
13490     REPEAT
13500     DO
13510       Position% = INSTR(Time_found$,"=") + 1
13520       Tyme$(Graph%) = MID$(Time_found$,Position%,14)
13530     END DO
13540 END DO
```

```
13560 RETURN,160
15000 REM
15010 DO
15020   IF Graph_chosen% > 20 AND Graph_chosen% < 25 THEN
EXIT 1 LEVELS
15030   CLEARSCREEN(2)
15040   COLOR 7,0,0 : SET CURSOR 13,6
15050   PRINT "The graph for the mobility experiment chose
n is not installed (oops)"
15060   END
15070 REPEAT
15090 File% = 1
15095 File$="\MOB"
15100 DO
15110   DIRECTORY(File$,0,Graph%,Error%)
15130 END DO
15140 RUNTYPE(Graph%,1)
15200 DO
15210   CLEARSCREEN(2)
15220   COLOR 2,0,0 : SET CURSOR 10,6
15230   INPUT "Enter the samples electrical thickness. ",D
ummy$
15240   Elec_thick! = VAL(Dummy$)
15250   IF Elec_thick! <= 0 THEN EXIT TO,15200
15260 END DO
15300 DO
15310   IF Graph_chosen% <> 24 THEN EXIT 1 LEVELS
15320   CLEARSCREEN(16)
15330   SET CURSOR 16,6
15340   INPUT "Enter the average dopant density. ",Dummy$
15350   Nd! = VAL(Dummy$)
15360   IF Nd! <= 0 THEN EXIT TO,15300
15370 END DO
15380 DATAPRINT
15500 DO
15510   DO
15516     IF INSTR(Temp_type$(Graph%),"T") = 0 THEN Temp_t
ype$(Graph%) = "R"
15517   END DO
15520   IF INSTR(Directory$(Graph%,File%),"NO") <> 0 THEN
EXIT 2 LEVELS
15530   INPUT #File%,Dummy$,I$,V1$,V2$,Dummy$
15540   R1! = VAL(V2$) /VAL(I$)
15550   INPUT #File%,Dummy$,I$,V1$,V2$,Dummy$
15560   R2! = VAL(V2$)/VAL(I$)
15570   INPUT #File%,Dummy$,I$,V1$,V2$,Dummy$
15575   Dummy! = MIN(R2!,R1!)
15576   Eric! = MAX(R2!,R1!)
15580   G! = 1.0 / ((3.141592654 *VAL(I$) / (2 * VAL(V1$)
* .693147181)) * (R1!+R2!)*(1-.138974234 * LOG(ABS(Eric! /
 Dummy!))))
```

```
15590    DATARETRIEVAL(9,Graph%,File%,Error%,0)
15600    IF Error% <> 0 THEN EXIT TO,15200
15610 END DO
15700 DO
15710    DO
15720       IF Graph_chosen% <> 21 THEN EXIT 1 LEVELS
15725       Graph_name$(1,21) = "Resistivity"
15730       Ytitle$ = "Resistivity in ohms*cm"
15731       DO
15732          CLOSE #File%
15733          File% = File% + 1
15734          IF File% = 3 OR INSTR(Directory$(Graph%,2),"NO
") <> 0 THEN EXIT 1 LEVELS
15735          Max_graph_points%(Graph%) = Max_graph_points%(
Graph%) + 1
15736          EXIT TO,15500
15737       REPEAT
15740       EXIT 2 LEVELS
15750    END DO
15760    DO
15770       IF Graph_chosen% < 22 OR Graph_chosen% > 24 THEN
 EXIT 1 LEVELS
15780       CLOSE #File%
15790       DO
15810          IF INSTR(SOURCE$,"B")=0 THEN OPEN "\DATA\"+ Di
rectory$(Graph%,File%) + File$ FOR INPUT AS #File% ELSE OP
EN "B:" + File$ +"."+Directory$(Graph%,File%) FOR INPUT AS
 #File%
15820          DO 15 TIMES
15830             INPUT #File%,Dummy$
15840          REPEAT
15850       END DO
15855       IF File% = 1 THEN Counter% = Max_graph_points%(G
raph%) + 1
15856       IF File% = 1 THEN Max_graph_points%(Graph%) = 1
15857       IF File% = 2 THEN Max_graph_points%(Graph%) = Co
unter%
15860       DO
15870          IF INSTR(Directory$(Graph%,File%),"NO") <> 0 T
HEN EXIT 1 LEVELS
15880          DATARETRIEVAL(10,Graph%,File%,Error%,0)
15890          IF Error% <> 0 THEN EXIT TO,15200
15900       END DO
15912       DO
15913          CLOSE #File%
15914          File% = File% + 1
15915          IF File% = 3 OR INSTR(Directory$(Graph%,2),"NO
") <> 0 THEN EXIT 1 LEVELS
15916          Max_graph_points%(Graph%) = Max_graph_points%(
Graph%) + 1
15917          EXIT TO,15500
```

```
15918       REPEAT
15919       DELTAMULTIPLY(Graph%,2,Elec_thick!)
15920       DO
15930         IF Graph_chosen% <> 22 THEN EXIT 1 LEVELS
15940         Ytitle$ = "Mobility in cm**2/V-s"
15945         Graph_name$(1,22) = "Mobility"
15950         EXIT 3 LEVELS
15960       END DO
15970       DO
15980         IF Graph_chosen% <> 23 AND Graph_chosen% <> 24
 THEN EXIT 1 LEVELS
15990         OPEN "\BUCKET" FOR INPUT AS #3
16000         FOR N% = 1 TO Max_graph_points%(Graph%)
16010           INPUT #3,Y!
16020           Graph!(Graph%,2,N%) = 1/ (Q! * Y! * Graph!(G
raph%,2,N%))
16030         NEXT N%
16040         DO
16050           IF Graph_chosen% <> 23 THEN EXIT 1 LEVELS
16060           Ytitle$ = "Carrier Conc in 1/cm**3"
16062           Graph_name$(1,23) = "Carrier Concentration n
"
16065           EXIT 3 LEVELS
16070           EXIT 4 LEVELS
16080         END DO
16090         DO
16100           DELTAMULTIPLY(Graph%,2,100.0/Nd!)
16110           Ytitle$ = "% Activation"
16115           Graph_name$(1,24) = "Dopant Activation %"
16120           EXIT 3 LEVELS
16130         END DO
16140       REPEAT
16150     REPEAT
16160 END DO
16165 DO
16170     DO
16180       IF INSTR(Temp_type$(Graph%),"T") <> 0 THEN EXIT
1 LEVELS
16190       Xtitle$ = "Temperature in Kelvin"
16200       EXIT 2 LEVELS
16210     END DO
16220     DO
16230       Xtitle$ = "Time in Seconds"
16240     END DO
16245 END DO
16250 KILL "\BUCKET"
16260 RETURN,160
17000 REM
17010 DO
17020     IF Graph_chosen% = 6 OR Graph_chosen% = 7 THEN EXI
T 1 LEVELS
```

```
17030   CLEARSCREEN(2)
17040   SET CURSOR 13,1 : COLOR 7,0
17050   PRINT "The graph chosen has not been implemented (
17000)."
17060   STOP
17070 REPEAT
17210 DO
17220   DIRECTORY("\CG",0,Graph%,Error%)
17230   IF Error% <> 0 THEN EXIT TO,160
17235   RUNTYPE(Graph%,1)
17240 END DO
17250 DO
17260   CLEARSCREEN(2)
17270   SET CURSOR 13,6 : COLOR 2,0,0
17280   INPUT "Enter the Bias Voltage. ",Dummy$
17285   Bias_sought! = VAL(Dummy$)
17290 END DO
17500 FOR N% = 1 TO 2
17510   IF INSTR(Directory$(Graph%,N%),"NO") <> 0 THEN EXI
T 1 LEVELS
17512   DATAPRINT
17515   IF N% = 2 THEN Max_graph_points%(Graph%) = Max_gra
ph_points%(Graph%) + 1
17520   IF Graph_chosen% = 6 THEN ID% = 4 ELSE ID% = 5
17530   DATARETRIEVAL(ID%,Graph%,N%,Error%,0)
17540   IF Error% = 1 THEN EXIT TO,17210
17550 NEXT N%
17700 DO
17710   IF Graph_chosen% <> 6 THEN EXIT 1 LEVELS
17720   DELTAMULTIPLY(Graph%,2,1E+09)
17730   Ytitle$ = "Capacitance in nF"
17735   Graph_name$(1,6) = "Capacitance vs."
17740 END DO
17750 DO
17760   IF Graph_chosen% <> 7 THEN EXIT 1 LEVELS
17770   DELTAMULTIPLY(Graph%,2,1000.0)
17775   Graph_name$(1,7) = "Conductance vs."
17780   Ytitle$ = "Conductance in mMhos"
17790 END DO
17800 DO
17810   Position% = INSTR(Bias_found$,"=") + 1
17820   Bias$(Graph%) = MID$(Bias_found$,Position%)
17830   DO
17840     IF INSTR(Temp_type$(Graph%),"T") <> 0 THEN EXIT
1 LEVELS
17850     Xtitle$ = "Temperature in Kelvin"
17860     Graph_name$(1,Graph_chosen%) = Graph_name$(1,Gra
ph_chosen%) + "Temperature"
17870     EXIT 2 LEVELS
17880   REPEAT
17890   DO
```

```
17900      Xtitle$ = "Time in Seconds"
17910      Graph_name$(1,Graph_chosen%) = Graph_name$(1,Gra
ph_chosen%) + "Time"
17920    END DO
17930 END DO
17940 RETURN,160
20000 REM
20010 DO
20020   CLEARSCREEN(2)
20030   RESTORE,62000
20040   FOR N% = 4 TO 11
20050     READ Hand_info$(N%)
20060     COLOR 11,0 : LOCATE N%,1
20070     PRINT Hand_info$(N%)
20080   NEXT N%
20090 END DO
20200 LOCATE 4,1, : COLOR 0,7 : PRINT Hand_info$(4)
20210 COLOR 5,0 : LOCATE 13,40 : PRINT "MAKE SELECTION"
20215 LOCATE 4,1
20220 A$ = INKEY$ : IF A$ = "" THEN GOTO 20220
20230 IF ASC(RIGHT$(A$,1)) = 13 THEN GOTO 20300
20240 IF ASC(RIGHT$(A$,1)) = 72 THEN GOSUB 20400
20250 IF ASC(RIGHT$(A$,1)) = 80 THEN GOSUB 20500
20260 GOTO 20220
20300 ON CSRLIN -3 GOSUB 20600,20700,20800,20900,21100,220
00,22600,22400
20310 PRINT CSRLIN : STOP
20400 Position% = CSRLIN
20410 COLOR 11,0 : LOCATE Position%,1 : PRINT Hand_info$(P
osition%)
20420 IF Position% < 5 THEN Position% = 12
20430 COLOR 0,7 : LOCATE Position%-1,1 : PRINT Hand_info$(
Position%-1);
20440 RETURN,20220
20500 Position% = CSRLIN
20510 COLOR 11,0 : LOCATE Position%,1 : PRINT Hand_info$(P
osition%)
20520 IF Position% > 10 THEN Position% = 3
20530 COLOR 0,7 : LOCATE Position%+1,1 : PRINT Hand_info$(
Position%+1);
20540 RETURN,20220
20600 GOSUB 22500
20610 DO
20620   LINE INPUT "Enter the TITLE of the graph (up to 60
 letters) ";Graph_name$(1,1)
20625   Position% = INSTR(Graph_name$(1,1),"   ") - 1
20627   Graph_name$(1,1) = MID$(Graph_name$(1,1),1,Positio
n%)
20630   CLEARSCREEN (20)
20640 END DO
20650 RETURN,20200
```

```
20700 GOSUB 22500
20710 DO
20720   LINE INPUT "Enter the X AXIS title (up to 60 lette
rs) ";Xtitle$
20725   Position% = INSTR(Xtitle$,"   ") - 1
20727   Xtitle$ = MID$(Xtitle$,1,Position%)
20730   CLEARSCREEN(20)
20740 END DO
20750 RETURN,20200
20800 GOSUB 22500
20810 DO
20820   LINE INPUT "Enter the Y AXIS title (up to 20 lette
rs) ";Ytitle$
20825   Position% = INSTR(Ytitle$,"   ") - 1
20827   Ytitle$ = MID$(Ytitle$,1,Position%)
20830   CLEARSCREEN(20)
20840 END DO
20850 RETURN,20200
20900 GOSUB 22500
20920 PRINT "Enter the X,Y data point (RETURN)   "
20930 COLOR 7,0
20940 FOR N% = Minloop% TO 1200
20947   SET CURSOR 21,43 : PRINT SPC(34)
20950   SET CURSOR 21,43
20955   DO
20960     LINE INPUT V$
20961     IF V$ = "" THEN EXIT TO,21005
20962     Pointer% = INSTR(V$,",")
20963     IF Pointer% = 0 THEN EXIT TO,21005
20964   END DO
20965   Graph!(Graph%,1,N%) = VAL(MID$(V$,1,Pointer%))
20966   Graph!(Graph%,2,N%) = VAL(MID$(V$,Pointer%+1))
20985   SET CURSOR 21,43 : PRINT SPC(34)
20990   CLEARSCREEN(22)
21000 NEXT N%
21005 Minloop% = N%
21007 Max_graph_points%(Graph%) = N%-1
21010 CLEARSCREEN (21)
21020 RETURN,20200
21050 RETURN
21100 GOSUB 22500
21110 SET CURSOR 3,40 : COLOR 5,0 : PRINT "Changing X-Y da
ta"
21120 Minloop1% = 1
21130 FOR N% = Minloop1% TO Max_graph_points%(Graph%)
21140   COLOR 2,0 : SET CURSOR N% + 5 - Minloop1%,45
21150   PRINT Graph!(Graph%,1,N%) "," Graph!(Graph%,2,N%)
21155   IF N%+5-Minloop1% > 17 THEN EXIT 1 LEVELS
21160 NEXT N%
21170 Maxloop% = N% - 1
21200 FOR N% = Minloop1% TO Maxloop%
```

```
21210   COLOR 0,2 : SET CURSOR N%+5-Minloop1%,45
21220   PRINT Graph!(Graph%,1,N%) "," Graph!(Graph%,2,N%)
21225   COLOR 0,0 :CLEARSCREEN(20)
21230   SET CURSOR N%+5-Minloop1%,45
21240   COLOR 0,2 : LINE INPUT V$
21245   COLOR 0,0 :CLEARSCREEN(20)
21250   COLOR 0,0 : Position% = INSTR(V$,",") + 1
21260   IF Position% = 1 THEN EXIT TO,21400
21270   Graph!(Graph%,1,N%) = VAL(MID$(V$,1,Position%))
21280   Graph!(Graph%,2,N%) = VAL(MID$(V$,Position%))
21290   COLOR 2,0 : SET CURSOR N%+5-Minloop1%,45
21295   PRINT SPC(35)
21297   COLOR 2,0 : SET CURSOR N%+5-Minloop1%,45
21300   PRINT Graph!(Graph%,1,N%) "," Graph!(Graph%,2,N%)
21305   CLEARSCREEN(20)
21310 NEXT N%
21400 DO
21410   IF Maxloop% => Max_graph_points%(Graph%) THEN EXIT
 TO,21500
21411   FOR N% = 4 TO 23
21412     COLOR 7,0 : SET CURSOR N%,30
21413     PRINT SPC(50)
21414   NEXT N%
21420   Minloop1% = Maxloop%
21430   EXIT TO,21130
21440 REPEAT
21500 FOR N% = 2 TO 23
21510   COLOR 7,0 : SET CURSOR N%,30
21520   PRINT SPC(50)
21530 NEXT N%
21540 RETURN,20200
22000 GOSUB 22500
22010 DO
22020   Choice% = 1:Saveflag=1
22030   SAVEDATA(Graph%)
22230 END DO
22240 SET CURSOR 13,40 : PRINT SPC(40)
22250 CLEARSCREEN(16)
22260 CLOSE #1
22270 RETURN,20200
22400 COLOR 7,0
22410 CLEARSCREEN(2)
22420 RETURN,160
22500 REM
22505 COLOR 11,0 : LOCATE CSRLIN,1 : PRINT Hand_info$(CSRL
IN)
22510 DO
22520   COLOR 0,0 : CLEARSCREEN(12)
22530   SET CURSOR 13,40:PRINT SPC(39)
22540   SET CURSOR 21,6 : COLOR 2,0
22580 END DO
```

```
22590 RETURN
22599 STOP
22600 GOSUB 22500
22610 LOADDATA(Graph%)
22870 RETURN,20200
22999 STOP
23000 REM
23020 DO
23030   IF Graph_chosen% <> 10 OR Graph_chosen% <> 2 THEN
Directory$(Graph%,2) = "NO"
23040 END DO
23500 DO
23505   IF Graph_chosen% = 10 OR Graph_chosen% = 2 THEN ID
% = 0 ELSE ID% = 1
23510   DIRECTORY("\CGV",ID%,Graph%,Error%)
23520   IF Error% <> 0 THEN EXIT TO,160
23530 END DO
23540 DO
23555   IF Graph_chosen% = 10 OR Graph_chosen% <> 2  THEN
ID% = 1 ELSE ID% = 0
23560   RUNTYPE(Graph%,ID%)
23570 END DO
23610 DO
23620   IF Graph_chosen% > 2 AND Graph_chosen% < 6 THEN EX
IT 1 LEVELS
23630   CLEARSCREEN(2)
23640   SET CURSOR 13,6 : COLOR 2,0,0
23650   INPUT "Enter the area of the device in cm squared.
  ",Area$
23660   Area! =VAL(Area$)
23665   IF Area! = 0 THEN EXIT TO,23650
23670 END  DO
23671 DO
23672   IF Graph_chosen% <> 10 AND Graph_chosen% <> 9 AND
Graph_chosen% <> 2 THEN EXIT 1 LEVELS
23673   CLEARSCREEN(15)
23674   SET CURSOR 15,6 : INPUT "Enter the type of materia
l you have tested (GaAs, Si)";Name$
23675   Name$ = UPPER$(Name$)
23676   IF INSTR(Name$,"GAAS") = 0 AND INSTR(Name$,"SI") =
 0 THEN EXIT TO,23671
23677   IF INSTR(Name$,"G") <> 0 THEN RESTORE,61000 ELSE R
ESTORE,61500
23678   READ Nc!,Dummy!
23679   E_sub_S! = E_sub_O! * Dummy!
23680   IF Graph_chosen% = 9 THEN EXIT 1 LEVELS
23681   CLEARSCREEN (17)
23682   SET CURSOR 17,6 : INPUT "Enter the donor density (
Nd) ",Nd$
23683   Nd! = VAL(Nd$)
23685   IF Nd! = 0 THEN GOTO 23682
```

```
23687 END DO
23689 DATAPRINT
23695 DO
23696   DO
23697     IF Graph_chosen% < 10 AND Graph_chosen% <> 2 THE
N EXIT 1 LEVELS
23698     IF INSTR(Temp_type$(Graph%),"T") = 0 THEN Temp_t
ype$(Graph%) = "R"
23699     IF INSTR(Temp_type$(Graph%),"T") = 0 THEN Run_ty
pe% = 1 ELSE Run_type% = 2
23700     EXIT 2 LEVELS
23701   REPEAT
23702   Run_type% = 0
23703 END DO
23704 File% = 1
23705 DO
23710   IF Graph_chosen% = 5 THEN ID% = 3 ELSE ID% = 2
23720   DATARETRIEVAL(ID%,Graph%,File%,Error%,Run_type%)
23730   IF Error% <> 0 AND Graph_chosen% = 10 THEN EXIT 1
LEVELS
23740   IF Error% <> 0 AND Graph_chosen% = 2 THEN EXIT 1 L
EVELS
23745   IF Error% <> 0 THEN EXIT TO,23540
23750 END DO
23800 DO
23810   DO
23820     IF Graph_chosen% <> 4 THEN EXIT 1 LEVELS
23830     DELTAMULTIPLY(Graph%,2,1E+09)
23840     Ytitle$ = "Cap in nF"
23850     EXIT 2 LEVELS
23860   REPEAT
23870   DO
23880     IF Graph_chosen% <> 5 THEN EXIT 1 LEVELS
23890     DELTAMULTIPLY(Graph%,2,1E+03)
23900     Ytitle$ = "Cond in mMhos"
23910     EXIT 2 LEVELS
23920   REPEAT
23930   DO
23940     IF Graph_chosen% <> 8 AND Graph_chosen% <> 1 THE
N EXIT 1 LEVELS
23945     IF Graph_chosen% <> 8 THEN Power! = 3.000 ELSE P
ower! = 2.0000
23950     INVERSEPOWER(Graph%,2,1.0E+09/Area!,Power!)
23960     IF Graph_chosen% = 8 THEN Ytitle$ = "1/C**2 C in
 nF" ELSE Ytitle$ = "1/C**3 in nF"
23970     EXIT 2 LEVELS
23980   REPEAT
23990   DO
24000     IF Graph_chosen% <> 9 THEN EXIT 1 LEVELS
24010     INVERSEPOWER(Graph%,2,1.0/Area!,2.000)
24015     MinX! = 1E+37 : MaxX! = -1E+37
```

```
24016      MinY! = 1E+37 : MaxY! = -1E+37
24020      FOR N% = 1 TO Max_graph_points%(Graph%)-1
24030        Dummy! = E_sub_S! * SQR(Graph!(Graph%,2,N%))
24040        Slope! = (Graph!(Graph%,2,N%+1) - Graph!(Graph
%,2,N%)) / (Graph!(Graph%,1,N%+1) - Graph!(Graph%,1,N%))
24050        Graph!(Graph%,1,N%) = Dummy!
24060        IF Slope! = 0 THEN  GOTO 24120
24070        Graph!(Graph%,2,N%) = (-2/(Q! * E_sub_S!)) * 1
/Slope!
24120      NEXT N%
24130      Max_graph_points%(Graph%) = Max_graph_points%(Gr
aph%) - 1
24140      Ytitle$ = "Dopant Density in 1/cm**3 X 1e+10"
24150      Xtitle$ = "Depth in microns"
24154      DELTAMULTIPLY(Graph%,2,1E-10)
24155      DELTAMULTIPLY(Graph%,1,1E+04)
24160      EXIT 2 LEVELS
24170    REPEAT
24180    DO
24190      IF Graph_chosen% <> 10 AND Graph_chosen% <> 2 TH
EN EXIT 1 LEVELS
24200      Counter% = Counter% + 1
24201      IF Graph_chosen% <> 10 THEN Power! = 3.0000 ELSE
 Power! = 2.0000
24205      INVERSEPOWER(Graph%,2,1/Area!,Power!)
24207      DELTAMULTIPLY(Graph%,2,1E-15)
24210      IF Counter% = 1 THEN ID% = 0 ELSE ID% = 1
24220      LEASTSQUARES(Graph%,Min!,Max!,Dummy!,Slope!,X_in
t!,Y_int!,ID%,Error_LS%)
24221      IF Counter% = 1 THEN GRange$(Graph%) = STR$(Min!
) + "," + STR$(Max!)
24225      IF Error_LS% = 1 THEN RUN,200
24230      Position% = INSTR(Temp_found$,";") + 1
24240      Temp! = VAL(MID$(Temp_found$,Position%))
24250      Y! = X_int! + ((K! * Temp!) / Q!) * LOG(Nc!/Nd!)
24260      DO
24270        DO
24280          IF INSTR(Temp_type$(Graph%),"T") <> 0 THEN E
XIT 1 LEVELS
24290          X! = Temp!
24300          EXIT 2 LEVELS
24310        REPEAT
24320        DO
24330          Position% = INSTR(Time_found$,"=") + 1
24340          X! = VAL(MID$(Time_found$,Position%))
24350        END DO
24360      END DO
24370      OPEN "\BUCKET" FOR APPEND AS #3
24380      PRINT #3,X!","Y!
24390      CLOSE #3 : CLOSE #File%
24393      IF INSTR(Directory$(Graph%,File%),"NO") <> 0 THE
```

```
N EXIT 1 LEVELS
24394      OPEN "A:\CGV." + Directory$(Graph%,File%) FOR IN
PUT AS #File%
24395      IF INSTR(Temp_type$(Graph%),"T") = 0 THEN Temp_t
ype$(Graph%) = "LH"
24396      Max_graph_points%(Graph%) = 1
24400      IF Error% = 0 THEN EXIT TO,23705
24401      DO
24402        File% = File% + 1
24403        IF File% = 3 OR INSTR(Directory$(Graph%,2),"NO
") <> 0 THEN EXIT 1 LEVELS
24404        CLOSE #1 : Error% = 0
24406        IF INSTR(Temp_type$(Graph%),"T") = 0 THEN Temp
_type$(Graph%) = "R"
24407        EXIT TO,23705
24408      REPEAT
24410      DO
24430        Max_graph_points%(Graph%) = Counter%
24440        GETDATA(Graph%)
24450        Graph_name$(1,10) = "Barrier Height (Capacitiv
e)"
24470        Ytitle$ = "Barrier in Volts"
24480        IF INSTR(Temp_type$(Graph%),"T") THEN Xtitle$
= "Time in Seconds" ELSE Xtitle$ = "Temperature in Kelvin"
24485        EXIT TO,3000
24490      END DO
24500    END DO
24510 END DO
24520 DO
24530    IF Graph_chosen% = 9 THEN EXIT TO,24560
24555    Xtitle$ = "Voltage in Volts"
24560    DO
24570      IF Temp_type$(Graph%) = "T" THEN EXIT 1 LEVELS
24580      Position% = INSTR(Temp_found$,";") + 1
24590      Temp$(Graph%) = MID$(Temp_found$,Position%)
24600      EXIT 2 LEVELS
24610    REPEAT
24615    DO
24620      Position% = INSTR(Time_found$,"=") + 1
24630      Tyme$(Graph%) = MID$(Time_found$,Position%)
24640    END DO
24650    EXIT TO, 160
24660 END DO
24900 RETURN, 160
25000 REM
25210 DO
25220    DIRECTORY("\RES",0,Graph%,Error%)
25230    IF Error% <> 0 THEN EXIT TO,160
25235    RUNTYPE(Graph%, 1)
25240 END DO
25250 DO
```

```
25260    CLEARSCREEN(2)
25270    SET CURSOR 13,6 : COLOR 2,0,0
25280    INPUT "Enter the Current Bias (mA).",Dummy$
25281    Bias_sought! = VAL(Dummy$)
25282    SET CURSOR 15,6 : COLOR 2,0,0
25283    INPUT "Enter the 4-pt. Correction Factor(CF;pg.31 of Sze).",Dummy$
25284    CF = VAL(Dummy$)
25286    SET CURSOR 17,6 : COLOR 2,0,0
25287    INPUT "Enter the Sample thickness.",Dummy$
25288    Thick! = VAL(Dummy$)
25290 END DO
25500 FOR N% = 1 TO 2
25510    IF INSTR(Directory$(Graph%,N%),"NO") <> 0 THEN EXIT 1 LEVELS
25512    DATAPRINT
25515    IF N% = 2 THEN Max_graph_points%(Graph%) = Max_graph_points%(Graph%) + 1
25520    ID% = 4
25530    DATARETRIEVAL(ID%,Graph%,N%,Error%,0)
25540    IF Error% = 1 THEN EXIT TO,25210
25550 NEXT N%
25700 DO
25720    DELTAMULTIPLY(Graph%,2,CF*Thick!/(.001*Bias_sought!))
25730    Ytitle$ = "Resistivity in ohm*cms"
25735    Graph_name$(1,6) = "Resistivity vs."
25740 END DO
25800 DO
25810    Position% = INSTR(Bias_found$,"=") + 1
25820    Bias$(Graph%) = MID$(Bias_found$,Position%)
25830    DO
25840      IF INSTR(Temp_type$(Graph%),"T") <> 0 THEN EXIT 1 LEVELS
25850      Xtitle$ = "Temperature in Kelvin"
25860      Graph_name$(1,Graph_chosen%) = Graph_name$(1,Graph_chosen%) + "Temperature"
25870      EXIT 2 LEVELS
25880    REPEAT
25890    DO
25900      Xtitle$ = "Time in Seconds"
25910      Graph_name$(1,Graph_chosen%) = Graph_name$(1,Graph_chosen%) + "Time"
25920    END DO
25930 END DO
25940 RETURN,160
44999 STOP
55000 REM
55010 IF ERR = 723 AND ERL=6205 THEN Minloop% = M% + 1:RESUME,6045
55020 IF ERR = 1 AND ERL = 5520 THEN GOSUB 55500: RESUME,3
```

```
000
55030 IF ERR = 2 AND ERL = 5520 THEN GOSUB 55500: RESUME,3
000
55100 REM
55200 PRINT "The main program is bombing.  This is error "
;ERR;" from line ";ERL
55210 STOP
55499 STOP
55500 REM
55510 VIEW : SCREEN 0,0,0 : COLOR 7,0 : SET CURSOR 13,6
55520 PRINT "The Maximum  minus Minimum points are beyond
BetterBasic's Accuracy."
55530 TIMEDELAY(5)
55540 RETURN
60200 REM
60210 DATA 29
60220 DATA "Hand Graph Entry","Capacitance vs. Time","Cond
uctance vs. Time","Capacitance vs. Voltage {4280}","Conduc
tance vs. Voltage","Capacitance vs. Temp. (Time)","Conduct
ance vs. Temp. (Time)"
60230 DATA "1/C**2 vs. Voltage {4280}","Dopant Profile {42
80}","Barrier vs. Temp (Time) 1/C**2","DLTS Spectrum"
60240 DATA "Current vs. Voltage","Capacitance vs. Voltage
{4140}","1/C**2 vs. Voltage {4140}","log(I) vs. Voltage","
ln(I) vs. qV/kT","ln(If/T**2) vs. q/kT"
60250 DATA "Isat vs. Temp (Time)","Ideality Factor vs Temp
 (Time)","Barrier vs Temp. (Time) {4140}"
60260 DATA "Resistivity vs. Temp. (Time)","Mobility vs. Te
mp. (Time)","Carrier Conc. vs. Temp. (Time)","Activation %
 vs. Temp. (Time)"
60270 DATA "1/C**3 vs. Voltage {4280}","Barrier vs. Temp (
Time) 1/C**3","ln(Isat) vs. Temp (Time)","4-pt. Resistivit
y vs. Temp(Time)"
60500 REM
61000 DATA 4.7E+17,13.1
61500 DATA  2.8E+19,11.9
62000 DATA "TITLE","X-AXIS TITLE","Y-AXIS TITLE","X-Y DATA
 INPUT","CHANGE X-Y DATA","SAVE","LOAD","GO TO MAINKEY"

ENDFILE
```

Appendix B: Model Simulation Software

The flowchart for the general simulation software is given in Fig. 50 and is self explanatory. The actual documented software listing follows for both the DLTS and Capacitance-frequency simulations.

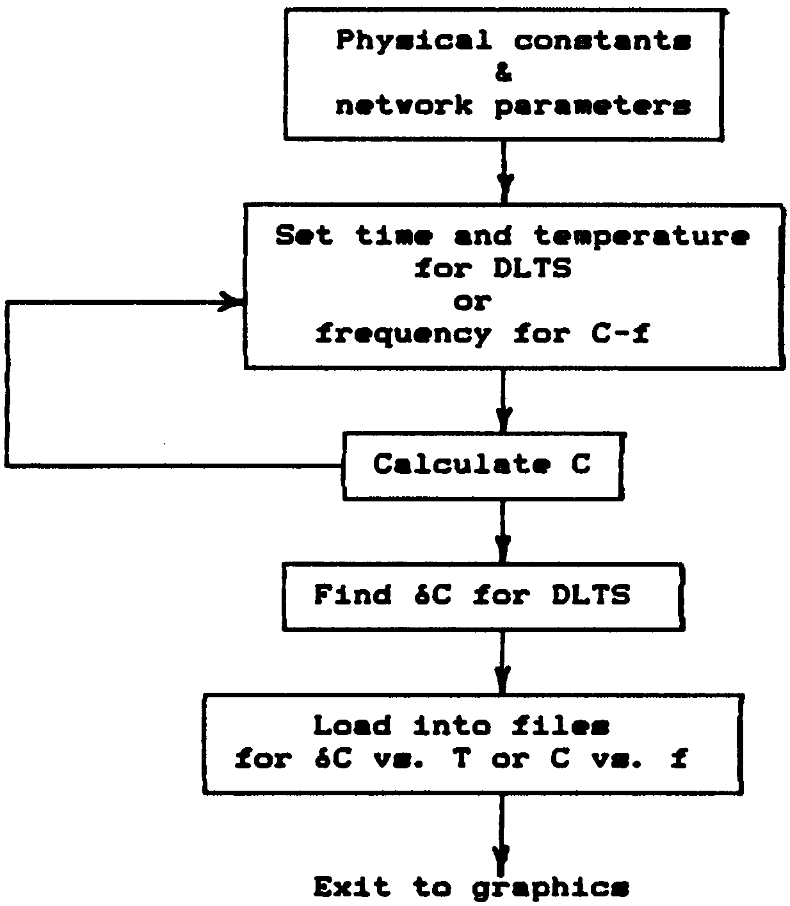


Figure 50. Flowchart for general simulation software.

```
SOURCE
PRECISION= 7
AUTODEF=ON
OPTION BASE=0
ERL=OFF
ERRORMODE=LOCAL
RESUME=LINE
FORMODE=BB
SCOPE=ON
PROCS=0
REAL: A,B,C,D,E,F,G,R1
REAL: R2,RD,C1,CD,W,TP,TM,ET
REAL: V,Q,K,X1,X2,CO,CT1,CT2
REAL: E1,E2
REAL ARRAY(2,5): CM
REAL: M,T
REAL ARRAY(20,20): CZ
REAL: Con,Cap,Peak,N


'MAIN Program:


   10 REM *THIS PROGRAM PRODUCES A DLTS PLOT USING COMPONE
NTS DERIVED FROM ANALYSIS
   20 REM
   30 REM *THESE ARE THE CONSTANTS USED FOR CALCULATION
   40 REM
   50 Q=1.6E-19:V=1E12:K=1.38E-23:W=6.283E6:Peak=0
   60 REM
   70 REM *THESE ARE THE NETWORK COMPONENTS:CO,RD-DIODE;C1
,R1-LAYER;R2-SHUNT
   80 REM
   90 RD=1:CO=2:R1=1:C1=2:R2=1
  100 REM
  110 REM THESE ARE THE EL2 TRAP PARAMETERS
  120 REM
  130 E2=.77:CT2=20E-12
  140 REM
  150 REM *THIS SECTION PUTS THE DATA IN AFORM WHICH CAN B
E USED BY MEDUSA
  160 REM
  170 OPEN "\ERIC\DATA#" FOR OUTPUT AS #1
  180 PRINT #1 "DLTS SPECTRUM"
  190 PRINT #1 "TEMP in K"
  200 PRINT #1 "DLTS in pFs"
  210 PRINT #1 ","
  220 PRINT #1 "X"
  230 PRINT #1 "X"
  240 PRINT #1 "X"
  250 PRINT #1 "X"
```

```
  260 PRINT #1  "X"
  270 PRINT #1  "X"
  280 PRINT #1  "X"
  290 PRINT #1  " 0"
  300 PRINT #1  " 0"
  310 PRINT #1  " 0"
  320 PRINT #1  " 0"
  330 PRINT #1  " 0"
  340 PRINT #1  " 0"
  350 PRINT #1  " 61"
  360 REM
  370 REM *THIS SECTION SIMULATES THE DLTS
  380 REM
  390 FOR TP=100 TO 400 STEP 5
  400   FOR M=1 TO 5
  410     IF Q*E2/K/TP>45 THEN X2=1E-8:GOTO 430
  420     X2=V*EXP(-Q*E2/K/TP)
  430     IF Q*E1/K/TP>45 THEN X1=1E-8:GOTO 450
  440     X1=V*EXP(-Q*E1/K/TP)
  450     TM=.008*M
  460     IF X1*TM>6 THEN X1=1000
  470     IF X2*TM>6 THEN X2=1000
  480     CD=C0-CT1*EXP(-X1*TM)-CT2*EXP(-X2*TM)
  490     A=R1/(1+W^2*C1^2*R1^2):B=W*C1*R1^2/(1+W^2*C1^2*R
1^2)
  500     C=RD/(1+W^2*CD^2*RD^2):D=W*CD*RD^2/(1+W^2*CD^2*R
D^2)
  510     E=A+C:F=B+D
  520     CZ(M,1)=1/W/F+2*E/W/F/R2+E^2/W/F/R2^2+F/W/R2^2
  530   NEXT M
  540   PRINT #1 " ";TP;" , ";1E12*(CZ(1,1)-CZ(5,1))
  550 NEXT TP
  560 CLOSE #1
  570 STOP

ENDFILE
```

```
SOURCE
PRECISION= 7
AUTODEF=ON
OPTION BASE=0
ERL=OFF
ERRORMODE=LOCAL
RESUME=LINE
FORMODE=BB
SCOPE=ON
PROCS=0
REAL: A,B,C,D,E,F,G,R1
REAL: R2,RD,C1,CD,W,TP,TM,ET
REAL: V,Q,K,X1,X2,CO,CT1,CT2
REAL: E1,E2
REAL ARRAY(2,5): CM
REAL: M,T
REAL ARRAY(20,20): CZ
REAL: Con,Cap,Peak,N


'MAIN Program:


   10 REM *THIS PROGRAM CALCULATES CAPACITANCE FREQUENCY D
ISPERSION
   20 REM
   30 REM
   40 REM *THESE ARE THE PHYSICAL CONSTANTS
   50 REM
   60 Q=1.6E-19:V=1E12:K=1.38E-23:Peak=0
   70 REM
   80 REM *THESE ARE THE NETWORK VALUES:DIODE-CO,RD;LAYER-
C1,R1;SHUNT-R2
   90 REM
  100 CO=150E-12:C1=2E-12:R1=0:R2=1E7:RD=1E7
  110 REM
  120 REM *THESE ARE THE EL2 TRAP PARAMETERS
  130 REM
  140 E2=.77:CT2=20E-12
  150 REM
  160 REM *THIS SETS THE DATA FILE IN A FORM FOR MEDUSA
  170 REM
  180 OPEN "\ERIC\DATD1" FOR OUTPUT AS #1
  190 PRINT #1 "CAP vs. FREQ"
  200 PRINT #1 "FREQ in Hz"
  210 PRINT #1 "CAP in pFs"
  220 PRINT #1 ","
  230 PRINT #1 "X"
  240 PRINT #1 "X"
  250 PRINT #1 "X"
  260 PRINT #1 "X"
```

```
  270 PRINT #1  "X"
  280 PRINT #1  "X"
  290 PRINT #1  "X"
  300 PRINT #1  "  O"
  310 PRINT #1  "  O"
  320 PRINT #1  "  O"
  330 PRINT #1  "  O"
  340 PRINT #1  "  O"
  350 PRINT #1  "  O"
  360 PRINT #1  "  101"
  370 REM
  380 REM *THIS FINDS CAPACITANCE DISPERSION
  390 REM
  400 FOR N=1 TO 501 STEP 5
  410    W=2*3.14159*N*100
  420    CD=CO
  430    A=R1/(1+W^2*C1^2*R1^2):B=W*C1*R1^2/(1+W^2*C1^2*R1^
2)
  440    C=RD/(1+W^2*CD^2*RD^2):D=W*CD*RD^2/(1+W^2*CD^2*RD^
2)
  450    E=A+C:F=B+D
  460    Con=((E+R2)^2+F^2)/(R2*E^2+E*R2^2+R2*F^2)
  470    Cap=1/W/F+2*E/W/F/R2+E^2/W/F/R2^2+F/W/R2^2
  480    PRINT #1 " ";W/2/3.14159;" , ";1E12*Cap
  490 NEXT N
  500 CLOSE #1
  510 STOP

ENDFILE
```

The vita has been removed from the scanned document