AN ABSTRACT OF THE THESIS OF

Pee-Keong Or for the degree of Master of Science in Electrical and Computer Engineering presented on December 10, 1987.

Title: Modeling of Subthreshold Current in GaAs MESFET and the Design of Voltage Reference Circuit.

Abstract approved: Redacted for Privacy
David J. Allstot

The characteristic of the subthreshold current in a GaAs MESFET exhibits a negative exponential function with Vgs. After studying the behavior of this current in this region over a range of temperature and drain bias voltages, a subthreshold current model was developed. The model was implemented in a circuit simulation program called VREFSIM. An arbitrary reference voltage is obtained by a simple selection of different component values. In this project, 2.56 volt and 1.28 volt references were designed and simulated. The simulated temperature coefficients of these two voltage references over a temperature range of -55 to 125 degrees Celsius were 7 and 26 parts-per-million (PPM)/Kelvin, respectively.

Modeling of Subthreshold Current in GaAs MESFET and the Design of Voltage Reference Circuit

by

Pee-Keong Or

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirment for the
degree of

Master of Science

Completed December 10, 1987

Commencement June 1988

APPROVED:

Redacted for Privacy

Associate Professor of Electrical and Computer Engineering
in charge of major

Redacted for Privacy

Head of department of Electrical and Computer Engineering

Redacted for Privacy

Dean of Graduate School

Date thesis is presented December 10, 1987

ACKNOWLEDGMENTS

I would like to express my appreciation and special thanks to Philip Canfield. Phil has generously provided the work station, controller and plotter programs, and his expertise in buried-channel GaAs MESFETs. I would like to thank Dr. L. Forbes for providing the laboratory facility, and Triquint Semiconductor for fabricating and supplying test devices. Special thanks to Howard Yang for assisting me in the circuit design. I would like to especially thank Dr. Allstot for his dedication, guidance and patience during the course of this work. This work was supported in part by the National Science Foundation under research grant number MIP-8709158.

TABLE OF CONTENTS

LIST OF FIGURES

Modeling of Subthreshold Current in GaAs MESFET and the Design of Voltage Reference Circuit

1. INTRODUCTION

As gallium arsenide (GaAs) integrated circuit (IC) technology matures, high-speed logic and memory applications of GaAs circuits are receiving more attention [1,2]. The inherent high electron mobility makes GaAs Metal Semiconductor Field Effect Transistors (MESFETs) an attractive technology over silicon based technologies. However, leakage currents in GaAs MESFETs due to backgating [3,4] and subthreshold conduction have impeded large scale integration of GaAs circuits.

In section 2 of this report, subthreshold current for a quarter micron gate length depletion-mode buried-channel GaAs MESFET is characterized. Two of the subthreshold models that were introduced in recent publications are discussed. Section 3 explores different possibilities of generating a temperature independent voltage reference. Finally, section 4 describes the circuit design implementation, simulation program VREFSIM, and gives design equations for generating other reference voltages using the same topology.

2. SUBTHRESHOLD CURRENT MODELING

2.1 Subthreshold Current Models

There are several approaches to model drain current behavior in MESFETs. These approaches vary from the physical models which involve a two-dimensional simulation [5,6] to a heuristic model [7]. Though it is insightful to do a comprehensive physical model, such a model obscures the necessary I-V relationship for circuit design and simulation. Therefore, the approach taken herein to model the subthreshold current is a compromise between understanding the first-order device physics, that is, relating device parameters to different drain voltage biases and temperature variations, and at the same time having an I-V relationship that is easily applied to circuit design and simulation.

Even though there are several sophisticated above-threshold MESFET models for circuit simulation [8,9], little work has been done in subthreshold modeling. The MESFET model that is presently implemented in SPICE2 is the Curtice MESFET model [10]. The Curtice model, as represented in Equation (1), is fairly accurate for Vgs values above the pinch-off voltage:

$$Ids = \beta(Vgs - Vt)^2\ (1 + \alpha Vds)\ \mathrm{Tanh}(\sigma Vds), \qquad (1)$$

where,

β is the transconductance,
Vt is the threshold voltage,
α is an empirical parameter that accounts for the output conductance, and
σ is the channel conductance modeled at low Vds.

Unfortunately, this model only represents the current in the square-law region, that is, above threshold voltage. When the gate to source voltage, Vgs, is equal to the threshold voltage, Vt, the model erroneously predicts zero current flow. The understanding of the drain current in the subthreshold region is critical to circuit design. One limitation to large-scale circuit integration in GaAs technology is the subthreshold leakage current. However, in this project the subthreshold current, which exhibits a linear exponential function is used to advantage in the design of a temperature insensitive voltage reference circuit.

Several subthreshold models have been introduced in recent publications. The McKinley model [11] is similar to the Curtice model with an added subthreshold term. The Chang model [12] specifically models the subthreshold conduction region. In the following, we discuss the McKinley and Chang models.

2.1.1 McKinley Drain Current Model [11]

The DC model Equation (2) for the drain to source current is implemented by McKinley as :

$$Ids = \beta (Vgs - Vto)^2 (1 + \alpha Vds) \, Tanh(\sigma Vds) + Vds/Rsh, \quad (2)$$

and

$$Rsh = Rsho/2 \; [exp \; (Vto - Vgs/STF * kT/q) + 1], \quad (3)$$

where,

Rsho	is the parasitic drain to source shunt resistance at $Vgs = Vto$, and
STF	is an empirical parameter.

Equation (2) is similar to Curtice model except that McKinley uses Rsh in Equation (2) to empirically model the subthreshold current. The subthreshold current is given as :

$$Isub = 2Vds/Rsho*[Exp(Vto - Vgs/STF*kT/q) + 1]. \quad (4)$$

This equation was not implemented in this project because it lacks a description of the subthreshold current for Vds at different temperatures. As was observed from the experimental data, drain voltage bias at different temperatures plays a significant role in the subthreshold current. Furthermore, the coefficient of the exponential term, $Io = 2Vds/Rsho$, is not a constant. As shown in section 2.2, a device is defined to enter the subthreshold region when $Isub = Io$, which is a constant.

2.1.2 Chang Subthreshold Current Model [12]

Chang specifically looks at the subthreshold region of MESFET drain current. He observed a linear exponential drain current in the subthreshold region and a monotonic increase of this current at a more negative Vgs due to leakage current through the gate. Figure 1 represents a typical subthreshold current characteristic. The semi-empirical expression is:

$$Isub = WJexp[(q/kTNs)(1 - \Gamma Vds)(Vgs - Vto + \Phi Vds)], \tag{5}$$

for $Vgs < Vto$,

where,

J	current for low drain voltage at Vgs = Vto in ampere/unit width,
Ns	Schottky ideality factor,
Γ	related mainly to the lowering of the Schottky barrier in $volt^{-1}$,
Φ	revealed mainly in the change of the subthreshold current slope, and
W	device width in microns.

This closed form of I-V relationship is relatively simple to implement in a voltage reference circuit design and simulation. The Chang model is therefore more appropriate to develop for this project than the McKinley model. The Chang model is implemented for a buried-channel depletion-mode GaAs MESFET in the next section.

2.2 Device for Model Implementation

The model developed in this project uses several definitions that differ from those of Chang. These modified definitions are necessary in order to relate first-order physics to the model and to set a convention for terms used in the rest of this thesis.

2.2.1 Buried-Channel GaAs MESFET

The device that is characterized and used in the circuit design has a 0.25 micron gate length and 50 micron gate width. It is a depletion-mode buried-channel GaAs MESFET. This device is well described in [13]. Figure 2 shows a side profile and energy band diagram and Figure 3 shows a impurity doping profile of the device.

The device is fabricated on a semi-insulating GaAs substrate. The active channel is silicon doped to a concentration of about 8E17 dopant/cm^3. This channel is "buried" between two p-type layers doped with beryllium. The top p-layer is shallow and highly doped, while the bottom p-layer is deep and lightly doped. The thickness of the active channel is defined as the distance from the gate depletion region to the channel-substrate depletion layer at zero gate bias. From Figure 3, the depth is about 6.6E-2 microns.

2.2.2 Subthreshold Leakage Current Mechanism

For above threshold voltage analysis, the channel under the gate region provides a conduit for free carriers to drift from the source to the drain under the influence of an electric field. The number of available free carriers is assumed to be proportional to the constant doping profile, Nc. The free carriers at the fringing edges is assumed to have little contribution to the drain current conduction as shown in Figure 5. The fringing edge effect is due to the probability of finding carriers a distance away from the edges. The assumption of a constant doping profile is valid only when Vgs > Vt since the total number of free carriers in the constant doping profile dominates over the total number of free carriers in the edges.

However, the carriers at the edges will play a significant role in drain current conduction when Vgs is equal or less than Vt. Under this condition, the doping profile concentration became a exponential term :

$$\text{Doping Concentration} \propto Nc \exp[q(Vgs - Vt)/kT]. \tag{6}$$

The edge effect is modeled as :

$$\text{Edge Effect} \propto \exp[-t/L_D], \tag{7}$$

where,

L_D is the Debye length, and
t channel thickness.

Therefore, the drain current density, Jn, and the total number of free carriers in the profile shown in Figure 5 is represented as:

$$Jn \propto \text{Number of free carriers},$$

$$Jn \propto 2 \int Nc \exp[q(Vgs - Vt)/kT] \exp[-t/L_D]\, dt, \tag{8}$$

for $Vgs <= Vt$.

This first order analysis shows the exponential characteristic of the subthreshold current.

2.2.3 Development of Subthreshold Current Model in Buried-Channel GaAs MESFET

In this project the subthreshold current is modeled as :

$$Isub = Io \exp[(q/kTNs)(1 - \Gamma Vds)(Vgs - Vto + \Phi Vds)], \quad (9)$$

where,

Io	current when the device enters subthreshold region,
Ns, Γ	parameters that model changes in the slope, and
Φ	is related to barrier lowering associated with changes in Vds.

The approach to this model is semi-empirical. For this particular technology a set of empirical values for Ns, Γ and Φ have been determined from experimental data. Ns and Γ are obtained from a curve fitting program (see Appendix 7.4).

Because of the steep slope of the subthreshold characteristic, it is more appropriate to define a constant current when the device enters subthreshold region rather than a constant voltage. When the drain bias voltage, Vds, and/or the temperature is increased, at constant Vgs, the active channel thickness is increased from t to t'[14] as shown in Figure 6a. Thus, more current flow through the channel. Graphical representation of the experimental data in Figure 6b illustrates the physical behavior of the drain current in Figure 6a. A more negative Vgs is needed to deplete the channel from t' back to t. A constant Io provides a consistent measure of the subthreshold region of operation. Io is derived from Equation (2) :

$$Ids = \beta(Vgs - Vt)^2 (1 + \alpha Vds) Tanh(\sigma Vds).$$

When Vgs is equal to the subthreshold voltage, Vts (see Equation (11)), Ids = Io. Since Io is independent of Vds (see Figure 7), Io is expressed as :

$$Io = \beta(Vts - Vt)^2, \tag{10}$$

For the device under study, Io is taken as 0.1 milliampere for all Vds values and at all temperatures. Figure 7 shows Io as defined on the data plots, and Figure 8 shows that Io as expressed in Equation (10) is constant for the temperature range of interest.

At the threshold voltage, the channel is not fully depleted as predicted in the abrupt transition case. The voltage at which the device enters the subthreshold region, Vts, is defined as the gate-source voltage for which the gate-channel depletion region and the channel-substrate depletion region are Υ Debye lengths apart [15] :

$$Vts = Vt + (2Vp\Upsilon L_D/t)(1 - \Upsilon L_D/2t), \tag{11}$$

From Sze [16], the definitions of transconductance, β , pinch-off current, Ip, and threshold voltage with Vds = 0, Vto, are used:

$$Ip = (W/L)Uq^2Nc^2t^3/6\epsilon s, \tag{12}$$

$$Vto = Vbi - Vp, \tag{13}$$

$$Vbi = kT/q\ LN[Nc/ni], \tag{14}$$

$$Vp = \frac{qNct}{2\epsilon s}\,\frac{T}{292}, \tag{15}$$

where,

Nc is the impurity doping concentration in dopant/cm^3,

U bulk carrier mobility in volt/cm^2-second.

From equations (12) and (15), β is simplified to :

$$\beta = \frac{2}{3}\,\frac{W}{L}\,\frac{Uc\epsilon s}{t}\,\frac{292^2}{T^2}. \tag{16}$$

It is observed that β is similar to the k' in a MOSFET, except it has an explicit temperature dependent term.

2.3 Experimental Procedure

A work station based on the Tektronix 4054A controller was set up to take measurements. A controller program was written in BASIC to interface various measuring instruments with the controller. A plotting program was also written to plot the experimental data. The large signals Vds, Vgs, Vp and Ids, for the device under investigation are first examined using a Tektronix 576 curve tracer. The data is stored on a magnetic tape for later analysis. An example of the measured data for the buried-channel MESFET is presented in Appendix 7.3.

2.3.1 Work Station SetUp

The equipment used in this project is listed in

Appendix 7.2. Figure 9 shows a schematic of the work station setup. Communication between instruments was based on HPIB protocol. A closed system helium refrigeration unit and cold head allowed low temperature measurements to be taken. Also listed in Appendices 7.4 and 7.5 are the program listings for the controller and plotter.

2.3.2 Initial Determination of Large Signal I-V Behavior

Before the measurement sequence began, it was imperative that the large signal characteristics of the device be examined. This is to assure that the data gathered is valid within the range of useful Vds, Vgs and Ids values.

Two parasitic effects in some MESFETs are kinking [17] and looping [18]. To avoid kinking effect, Vds is ramped up to a maximum of 3.0 volts. And to avoid looping, measurements were taken at DC. The pinch-off voltage is also determined experimentally using the curve tracer.

2.4 Subthreshold Current Model

The subthreshold current model developed in the previous three sections is implemented in this section. The implementation starts with the interpretation of data. The two empirical parameters Ns and Γ are extracted through a least-square method of curve-fitting.

2.4.1 Data Interpretation

The data acquired for the buried-channel MESFET is plotted as drain current, Ids, versus gate-source voltage, Vgs. To observe the influence of the drain bias voltage, Vds, a set of plots are made with constant temperature but varying Vds as shown in Appendix 7.3.1. Conversely, the effect of temperature is studied by keeping the Vds constant and varying the temperature as in Appendix 7.3.2.

The pinch-off voltage for this device at room temperature is approximately -0.75 volt. In the subthreshold region, the drain current varies exponentially in the region of -0.75 V < Vgs < -1.0 V. Also observed was a gradual monotonic increase of drain current for Vgs < -1.0 V. This increase was principally due to the gate conduction through the reverse-biased gate-to-drain diode.

Since Io is constant, the voltage in which the device enters subthreshold region will vary with temperature and Vds as modeled by the Φ parameter. From both representations of data in Appendix 7.3, the slope of the exponential curve decreases with increasing temperature and Vds. The changes in the slope are modeled by the Ns and Γ parameters.

2.4.2 Parameters Extraction

Two of the three parameters (Ns and Γ) that modeled the changes in temperature and Vds are extracted from a curve fitting program (Appendix 7.5). From the data plots, the changes in Vgs(Isub = Io), Φ, are consistent, and can be measured directly from the plot as shown in Figure 7.

The parameter values extracted by the methods described above for the device characterized are :

$Ns = 1.671$

$\Gamma = 0.046\ \mathrm{Volt}^{-1}$, and

$\Phi = 0.07$.

Implementing these parameters, and letting Io equal 0.1 milliampere in Equation (10), the current model agrees with the subthreshold current data with the experimental error as shown in Figure 10.

3. VOLTAGE REFERENCE

3.1 Generation of Vref

Voltage references find useful applications both in analog and digital circuits. In analog ICs, voltage references are found in operational amplifiers and analog-to-digital converters. In digital ICs, a temperature insensitive voltage reference is required for biasing circuits such as ECL gates. In all applications, the sensitivity of the voltage reference to temperature is a critical issue.

On a GaAs IC, the two devices that are available to generate a DC voltage reference are a thin-film resistor and MESFET a transistor. These two devices have one or more temperature dependent term in their current expressions. Therefore, the strategy was to find a positive and a negative temperature drift component. To obtain a minimum temperature dependence of the voltage reference, the scaled sum of these two components must give an overall change in current over temperature to be approximately equal to zero. Figure 11 shows a schematic of a voltage reference design by the technique of drain current summation. The next two sections examine the available components and how to make use of their temperature variation properties.

3.2 Negative Temperature Coefficient Component

The first-order temperature dependence of resistance of a resistor is represented by :

$$R(T) = Ro + \alpha_t(T - T_{273}), \quad (17)$$

where,

Ro resistance at 273 Kelvin,
α_t resistance coefficient of resistor in ohm/Kelvin,
T operational temperature, and
T_{273} is 273 Kelvin.

The current through the resistor and the derivative of the current with respect to the temperature are given by :

$$I_R = V/R(T), \quad (18)$$

$$dI_R/dT = -(\alpha_t V)/[Ro + \alpha_t(T - T_{273})]^2, \quad (19)$$

where, V is assumed to be a constant with temperature.

Assuming α_t is positive, the dI_R/dT is a negative quantity. Thus, resistors provide a negative temperature drift component of current.

The MESFET operating in the square-law saturation region also exhibits a negative temperature drift coefficient of current as the shown below :

$$Ids = \beta(Vgs - Vt)^2 \ (1 + \alpha Vds) \ Tanh(\delta Vds), \quad (20)$$

From Appendix 7.1.1, Equation (18) is recast as :

$$Ids = (A/T^3)(C - BT)^2 \ D, \quad (20a)$$

where,

A, B, C, D are positive temperature independent variables, and $(C - BT) > 0$.

Therefore, the derivative of Equation (20a) with respect to temperature is :

$$dIds/dT = -(2AD/T^3)(C - BT)(B + 2/T*(C - BT)) \quad (21)$$
$$= \text{Negative Quantity.}$$

Thus, the resistor and the MESFET operating in the square-law saturation region are both used in the design to provide current component with negative temperature drift.

Next, a positive temperature drift component of current with predicable behavior must be implemented in the circuit. When these two temperature components are appropriately combined, the resulting dI/dT is ideally zero.

3.3 Positive Temperature Coefficient Component

The other region of MESFET operation, the subthreshold region is now examined closely.

Though the subthreshold current is usually undesirable in a circuit design, its predicable behavior is advantageously made use of in this design. Interestingly, when the device operates in the subthreshold region, its subthreshold current increases with temperature, as shown in the following equations and derivative:

$$Idsub = Io \exp[(q/kTNs)(1 - \Gamma Vds)(Vgs - Vto + \Phi Vds)]. \quad (22)$$

From Appendix 7.1.2, Equation (22) is recast as :

$$Isub = Io \, Exp[-AB] \, Exp[AC/T], \quad (22a)$$

where,

A and B are positive temperature independent variables.
C is negative temperature independent variable.

Therefore, the derivative of Equation (23a) with respect to temperature is :

$$dIsub/dT = -Io\ EXP[-AB]\ (AC/T^2)\ exp[AC/T] \quad (23)$$
$$= \text{Positive Quantity.}$$

This current provides the positive temperature drift component for the voltage reference circuit design. This positive quantity is predicable by the current model. These quantities are further emphasized by the design curves generated from VREFSIM program in the next section.

4. VOLTAGE REFERENCE CIRCUIT DESIGN

4.1 Circuit Implementation

The voltage reference implementation technique with GaAs MESFETs deviates from that of the bipolar voltage reference [20]. As the name implies, the bipolar bandgap reference circuit is only referenced at one voltage, that is at the silicon bandgap voltage of 1.12 volt at room temperature. The technique deployed is a voltage summation [19] as shown in Figure 12.

Conversely, using GaAs MESFETs and current summation, any reference voltage is realizable so long as the reference voltage is within the power supply limits and all devices operate in saturation. The technique of current summation is illustrated in Figure 13.

From section 3, the schematic in Figure 11 has been realized with two MESFETs, one operating in the square-law saturation region and the other operating in the subthreshold saturation region. Figure 13 shows the transistor-level schematic of the circuit. The idea behind drain current summation is that the negative temperature coefficient of the drain current of device M2 will cancel the positive temperature coefficient of the drain current of the subthreshold device M1. Ideally, the current through the resistor Ro is constant with temperature. Thus, a voltage is generated by the constant current flowing through the resistor Ro. This is almost the case in the circuit simulation except that the thin-film resistor has a small positive temperature coefficient. For the worst case, the temperature coefficient of the resistance is 100 PPM/Kelvin. The resistor R1 biases M1 in the subthreshold region, and R3 holds the node voltage V1 to a variation of only q0.1 V. Without R1, V1 will vary in a range of q0.3V. This is undesirable because the device M1 may exit the the subthreshold region. The resistor R2 biases M2 in the square-law saturation region.

4.2 Square-Law Device in Saturation

In Section 2, the first-order subthreshold device physics were discussed. In this section, the first-order physics of the device operating in square-law saturation region are analyzed.

The doping concentration in the channel, Nc, is assumed to have a uniform profile. The electron mobility is inversely proportional to temperature and thus, mobility at different temperatures is calculated as :

$$\frac{U_{300}}{T_{300}} = \frac{U_T}{T}, \tag{24}$$

where,

U_{300}	bulk mobility at room temperature,
T_{300}	room temperature = 300 Kelvin,
U_T	mobility at temperature T, and
T	temperature of interest.

The device operating in square-law saturation region has a gate length of 0.25 micron and a gate width of 5 micron. The I-V equation for this region is given from the Curtice model as:

$$Ids = \beta(Vgs - Vt)^2 (1 + \alpha Vds) \, Tanh(\sigma Vds),$$

where,

$\alpha = 0.05 \text{ volt}^{-1}$,
$\sigma = 2.2 \text{ volt}^{-1}$, and
$Vgs(T=300) = 0.0$ volt.

The variables that change with temperature are β and Vt. These variations are analyzed in the next two sections.

4.2.1 Vp Variations with Temperature

The pinch-off voltage here assumes a channel pinch-off rather than saturated velocity pinch-off. The pinch-off voltage term used in this project has a slightly different definition from that of Sze. From Equation (15) :

$$Vp = \frac{qNct^2}{2\epsilon s}\,\frac{T}{292}.$$

With the assumption that the doping profile is a constant, Vp varies linearly with temperature. The derivative of $(Vp)^2$ with temperature is a positive term :

$$\frac{dVp}{dT} = \frac{2T}{292^2}\,\frac{qNct^2}{2\epsilon s}. \tag{25}$$

4.2.2 Beta Variations with Temperature

From Equations (16), the transconductance, , is represented as :

$$\beta = \frac{2}{3}\,\frac{W}{L}\,\frac{Uc\epsilon s}{tc}\,\frac{292^2}{T^2}.$$

The implicit temperature dependent term is mobility. The bulk mobility varies as T^{-1}. The derivative of β with respect to temperature is a negative term :

$$\frac{d\beta}{dT} = -\left(\frac{4}{3}\,\frac{W}{L}\,\frac{Uc\epsilon s}{tc}\,\frac{292^2}{T^3}\right). \tag{26}$$

Therefore,

$$\frac{dIds}{dT} = \frac{d\beta}{dT} * \frac{dVp}{dT} * C, \tag{27}$$

where,

$$C = (1 + \alpha Vds)\ \mathrm{Tanh}(\delta Vds).$$

4.3 VREFSIM Circuit Simulation Program

The MESFET model that is available in SPICE2 does not model the subthreshold behavior. A simulation program that implements the DC behavior of the MESFET, including the subthreshold region was implemented in Pascal. In the following sections, the basic programming technique, user interface, and generation of design curves are discussed.

4.3.1 Programming Technique

The VREFSIM program consists of four subroutines. Three of these subroutines generate design curves and one simulates the circuit. The purpose of these different subroutines is explained fully in section 4.3.3. The required algorithms are simplified since only the DC analysis of the circuit is of interest for DC voltage references. The programming techniques for these programs are similar, with the basic idea being to establish a continuity of current at a node.

A node is the point where two or more devices are connected together as illustrated in Figure 14. An initial guess of the node voltage, Vn, is made. For the example shown in the Figure 14, the resistor current, Ir, is calculated using Ohm's law and the current for the MESFET, Im, is found using Equation (2). The relative ratio of Im/Ir indicates the continuity of current through the node. The ratio is recast as (Im/Ir - 1). For the implemented program, the error tolerance was set at 0.5 percent. When the ratio is less than or equal to the tolerance, the continuity of current at that node is established. Otherwise the node voltage is either increased or decreased so that the ratio will approach the tolerance level.

4.3.2 Generating Design Curves

There are three subroutines in VREFSIM that generate design curves for a particular subthreshold device and a square-law device. These subroutines generate the design curves that give the designer a choice of different square-law device gate width for a given gate length, and the subthreshold biasing current for a given Vgs. With this information the designer is able to optimize the design. The generation and explanation of these design curves are given in the following sub-sections.

4.3.2.1 Square-Law Device

The square-law device is simulated by the subroutine called NORMAL. For a given gate width, the current-temperature slope is calculated and plotted in Figure 15.

4.3.2.2 Subthreshold Device

There are two subroutines that simulate the device in subthreshold saturation region. One subroutine, SUB_DEV, generates a plot of dI/dT versus the bias current, Ibias, for a given Vgs, as shown in Figure 16. The purpose of this design curve is to give the designer a view of different dI/dT values that are obtainable with different Vgs and bias current values.

The second subroutine, SUB_R1R3 calculates the values of R1 and R3 for the given bias current. Since nickel-chrome thin-film resistors are used in this GaAs process, the values of resistance have a resolution of 1 ohm. Therefore, R1 and R3 are accurately adjusted to give the desired bias current and dI/dT to compensate the dI/dT of the square-law device.

4.3.3 Design Procedure

Since the generated voltage is based on the principal of current summation, the dI/dT's of the square-law device and the subthreshold device must add up to approximately zero. By using the design curves, one can determine the range of dI/dT's that are suitable for these devices to compensate each other.

The program NORMAL provides the appropriate width W for the square-law device and SUB_R1R3 gives the values of R1 and R3. These data are then input into the full circuit simulation subroutine called CKTSIM. The output gives the voltage reference value over the desired range of temperatures. The temperature coefficient of the reference is determined as :

$$\mathrm{PPM} = \frac{(\mathrm{Vhi} - \mathrm{Vnom}) * 1\mathrm{E}6}{\mathrm{Vhi} * (\mathrm{TempHi} - \mathrm{Templo})}. \qquad (28)$$

4.2.6 Simulated Output

Two voltage reference circuits were designed. Both used the same number of circuit devices. The circuits were simulated over the temperature range of -55 to 125 degrees Celsius. Figure 17 shows the negative temperature drift component of the square-law device, and Figure 18 shows the positive temperature drift component of the subthreshold device. One reference is at 2.56 volts with a temperature variation of 6.5 PPM/Kelvin. The second reference is at 1.28 volt with a temperature variation of 26 PPM/Kelvin. Figures 19 and 20 show the simulated output of these two circuits.

5. CONCLUSION

The advantages of this voltage reference over a bipolar bandgap reference [20] are very significant. A typical temperature sensitivity for a bipolar bandgap circuit is in a range of 30 - 60 PPM/Kelvin, whereas, for this MESFET circuit, the voltage variation over temperature is about 6.5 PPM/Kelvin for the 2.56 volt reference.

The differences in performance are due to the basic technique of generating the voltage reference. In bipolar technology, the reference voltage is generated by summing Vbe and delta Vbe. The derivative of this sum with temperature determines the performance of the circuit with temperature. In GaAs MESFET technology, the reference voltage is generated by summing the derivative of current components with temperature :

$$dI/dT = dIsub/dT + dIds/dT. \qquad (28)$$

The dI/dT is typically in the range of 1E-6 ampere/Kelvin, whereas, dV/dT is typically in the range of 1E-3 volt/Kelvin. The resolution of the current summation technique is three orders of magnitude better than the voltage summation technique.

One important advantage of this MESFET circuit is that it requires no amplification. In this project, 2.56 volt and 1.28 volt references are realized using the same circuit without any additional components. In other words, any voltage reference is realizable using this circuit as long as the devices are kept in saturation. This is also an important advantage over other voltage reference techniques. For example, in a bipolar bandgap reference, the 1.12 volt reference is not generally very useful. This voltage must be amplified by an external gain circuit to a more practical voltage. The additional devices in the amplification circuit not only increase the cost, but also degrade the performance. Figure 21 shows a comparison between the two voltage reference schemes.

Finally, it was observed that the subthreshold behavior in a MESFET is similar to a MOSFET. Therefore, it may be possible to transfer the design technique from GaAs MESFETs to silicon MOSFETs.

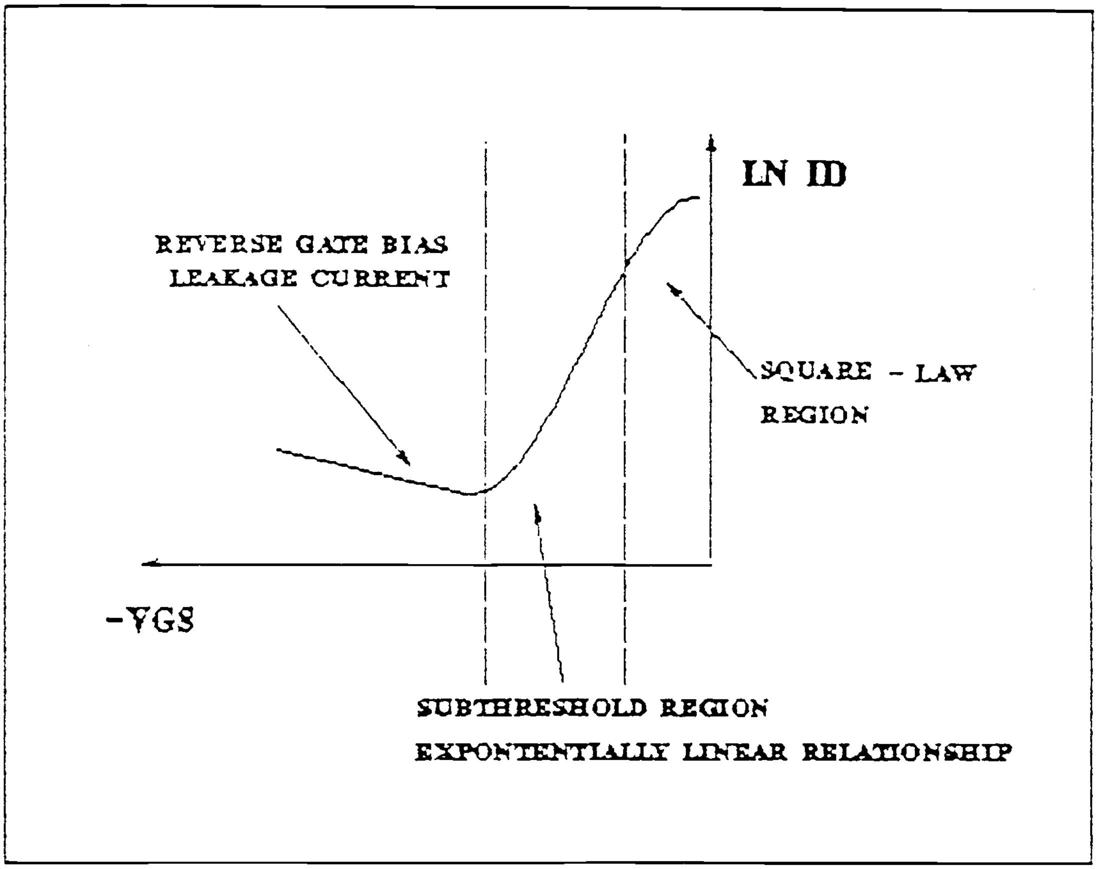


Figure 1. MESFET in Subthreshold Current Region

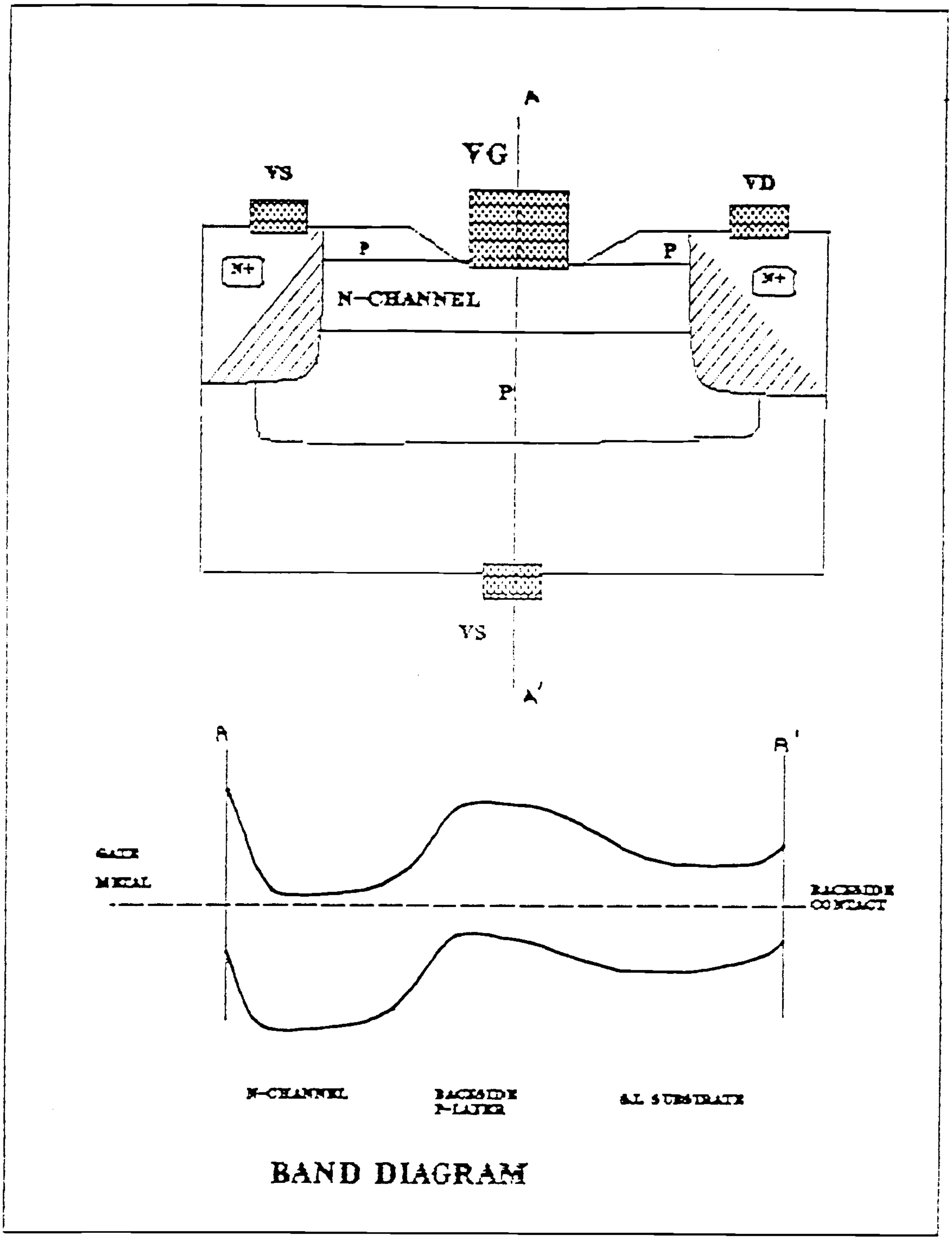


Figure 2. Buried-Channel GaAs MESFET

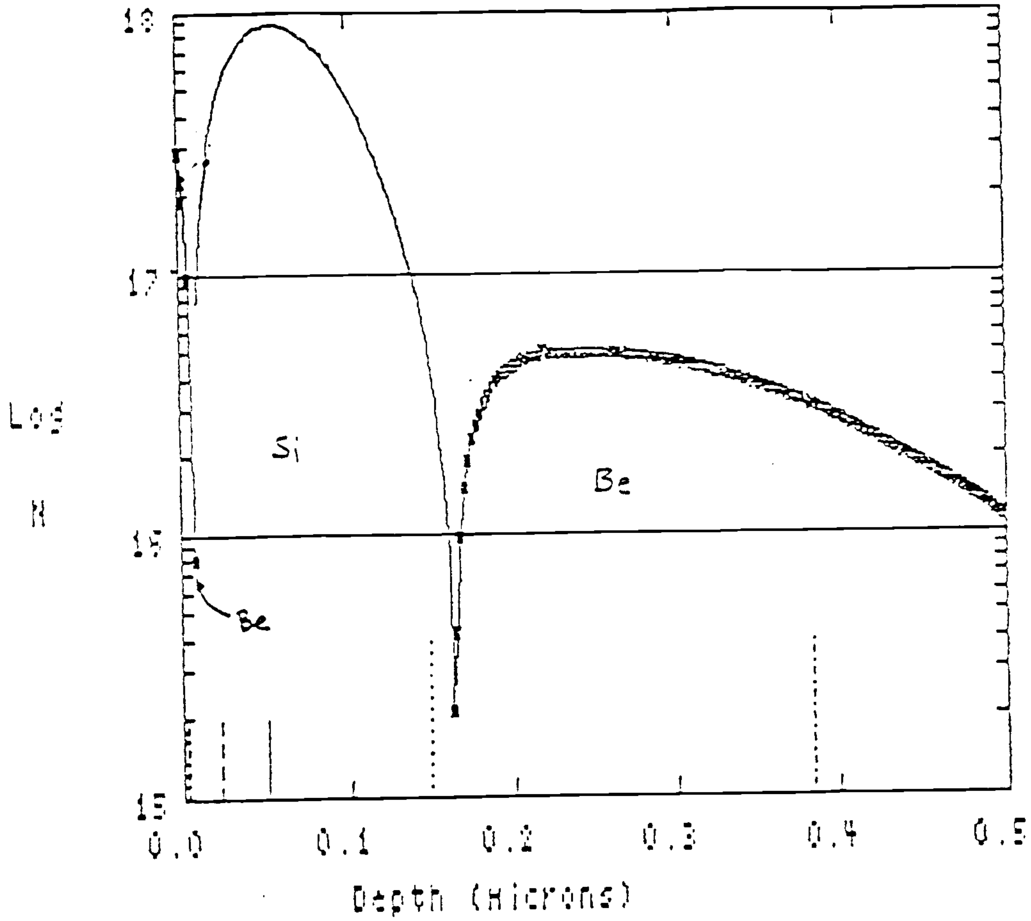


Figure 3. Impurity Doping Profile

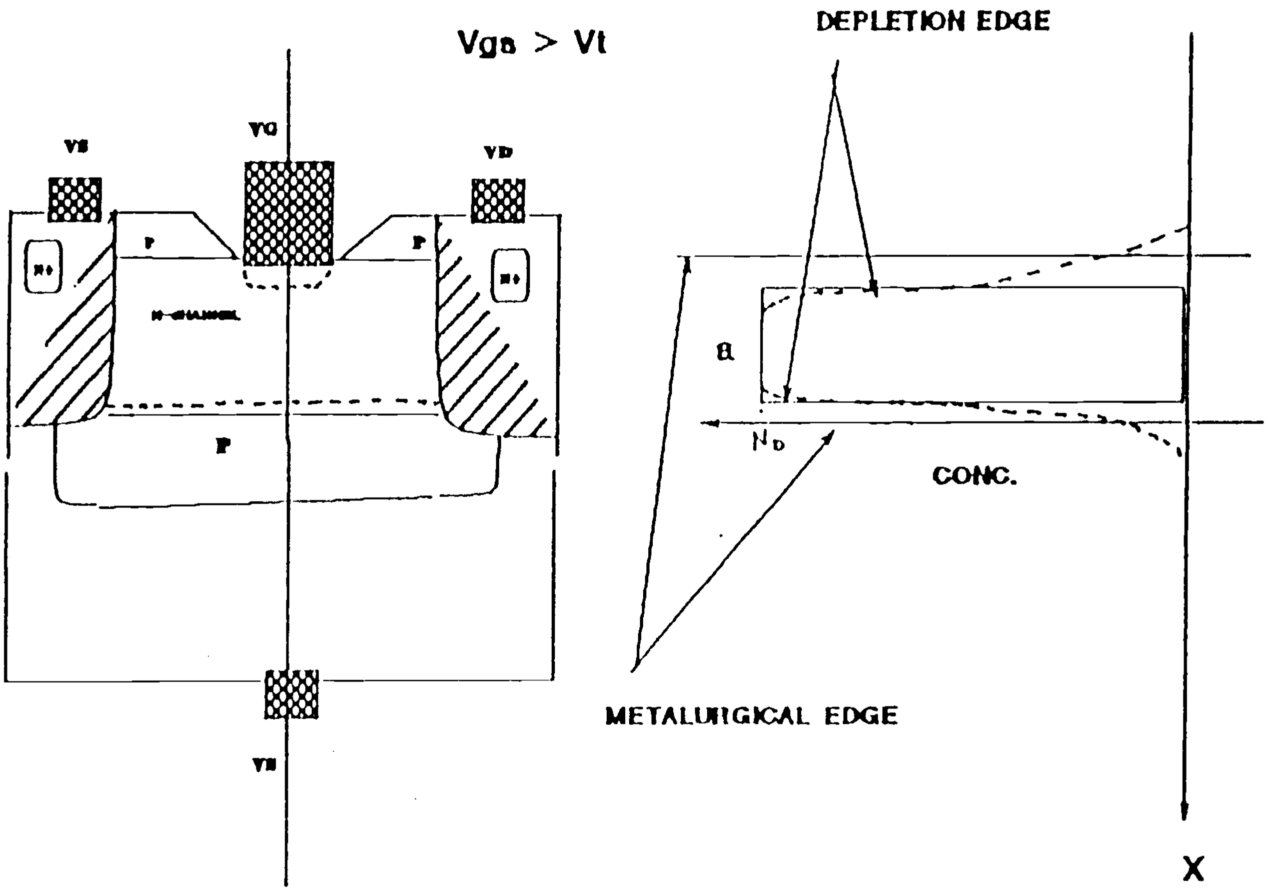
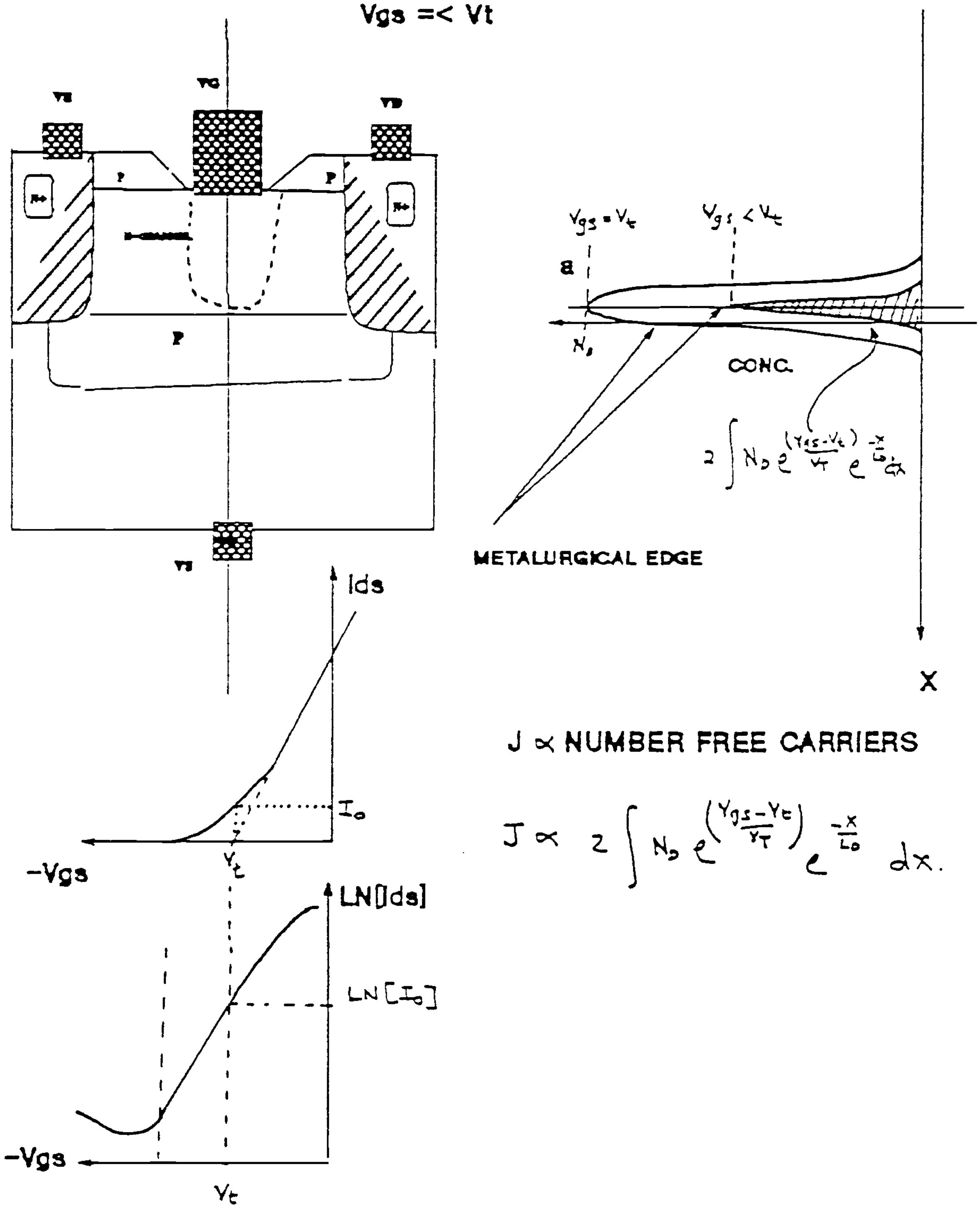


Figure 4. Above Threshold Drain Current Conduction



J ∝ NUMBER FREE CARRIERS

$$J \propto 2\int N_D e^{\left(\frac{V_{gs}-V_t}{V_T}\right)} e^{\frac{-x}{L_0}} dx.$$

Figure 5. Below Threshold Drain Current Conduction

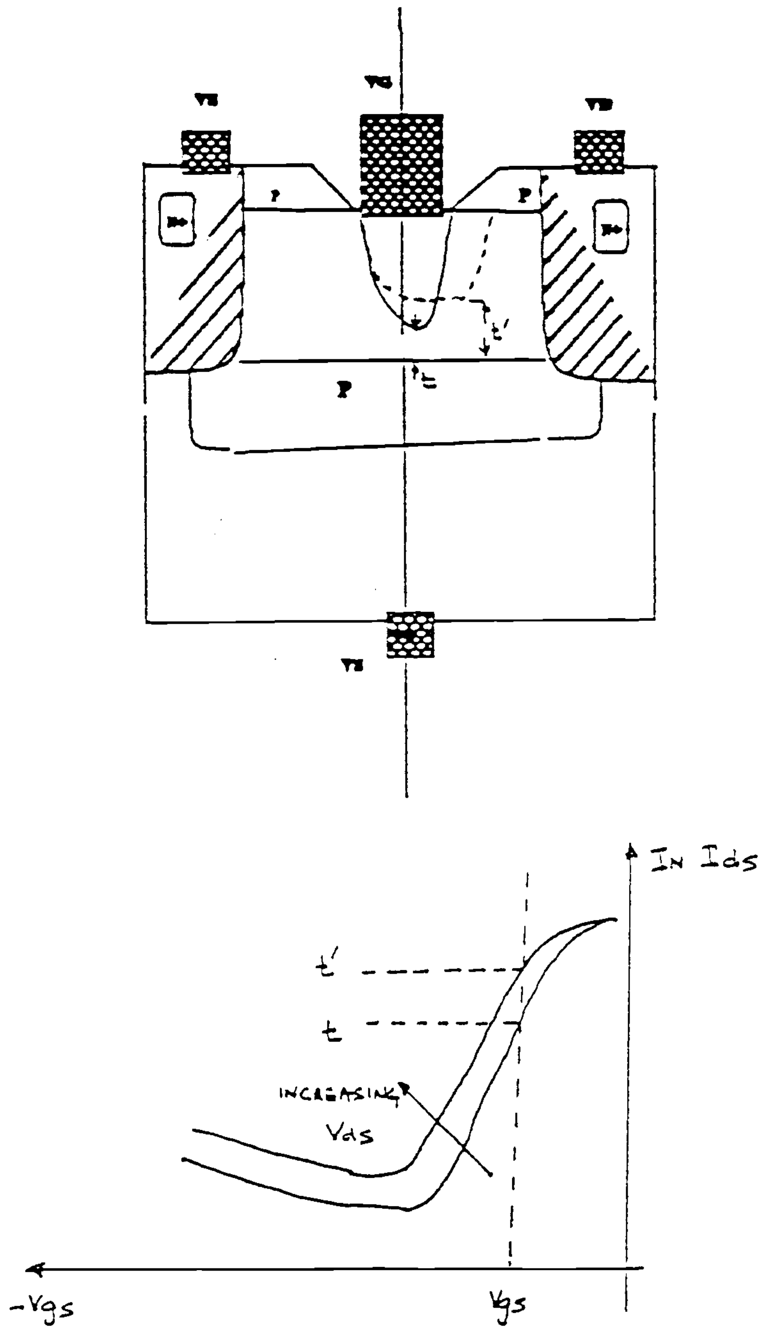


Figure 6. Changes in Channel Thickness Due to Vds change

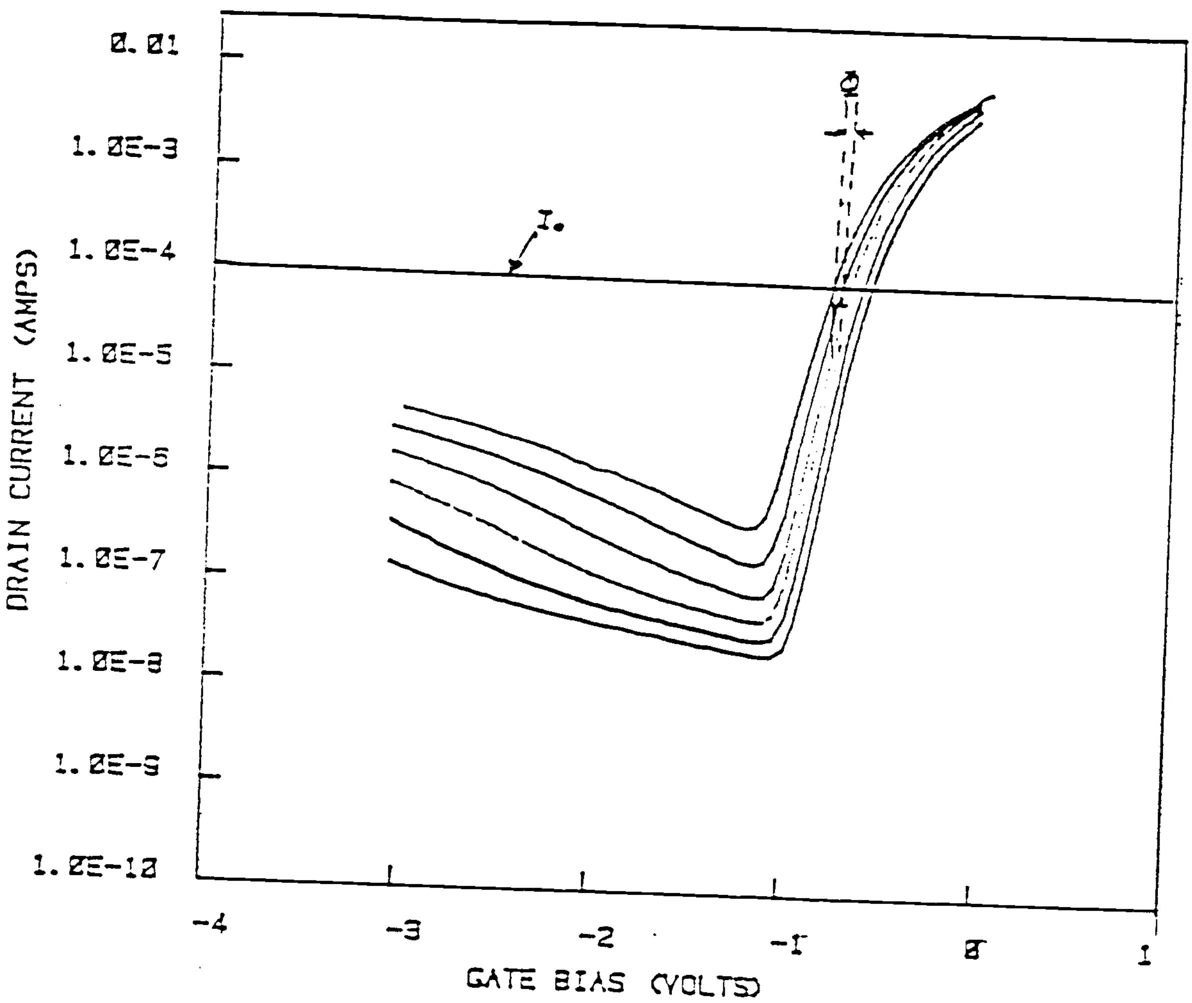


Figure 7. Io as defined on Data Plot

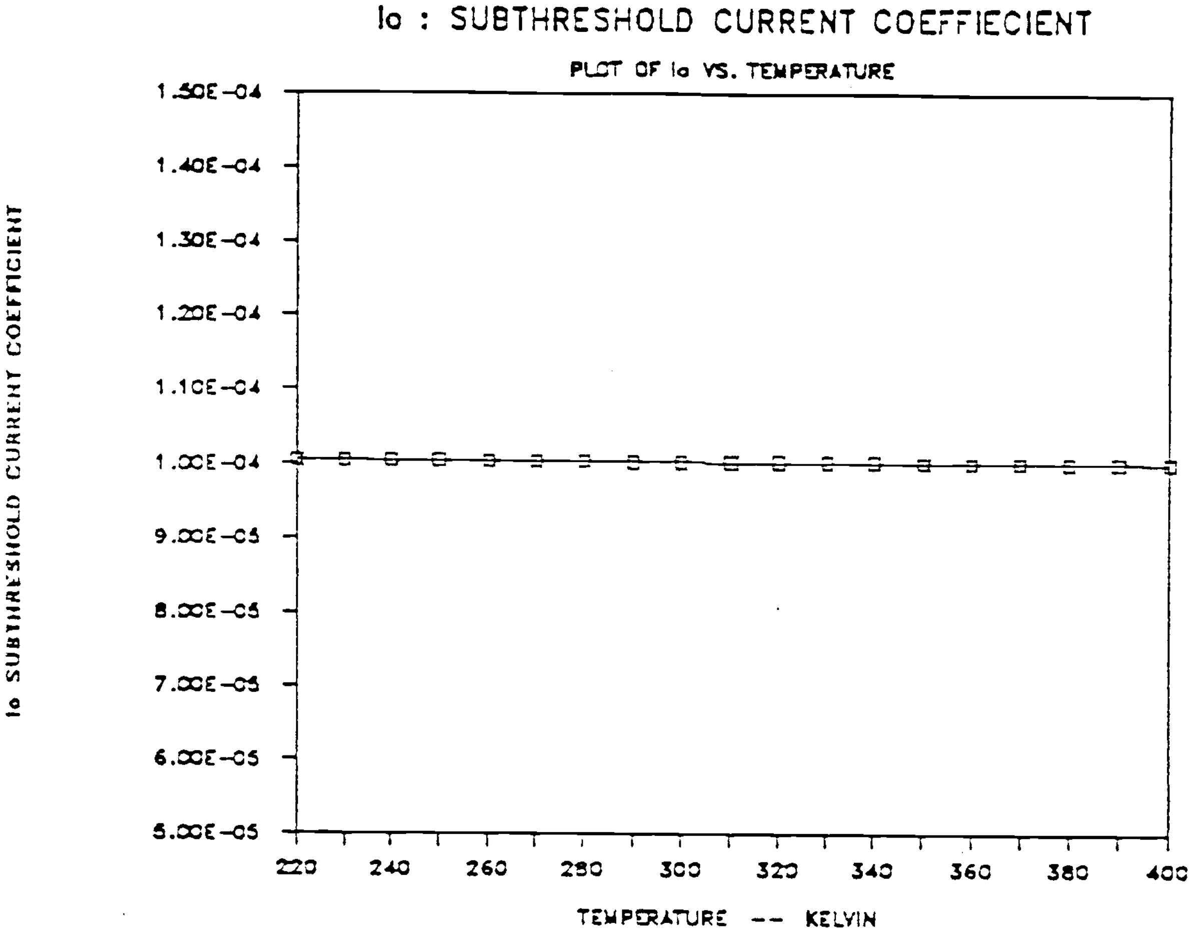


Figure 8. Io as expressed by Eqn. (10)

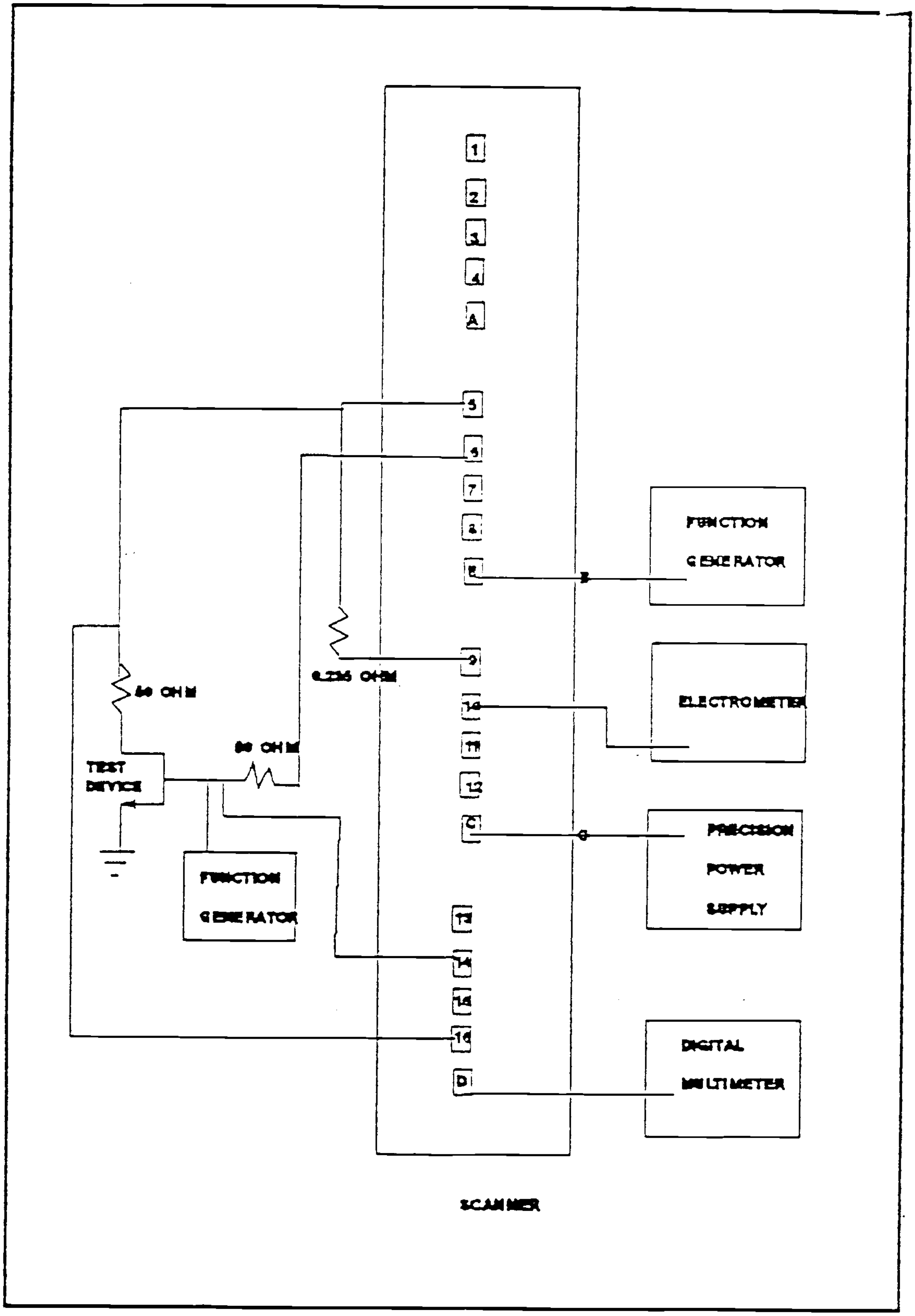


Figure 9. Schematic of Work Station Setup

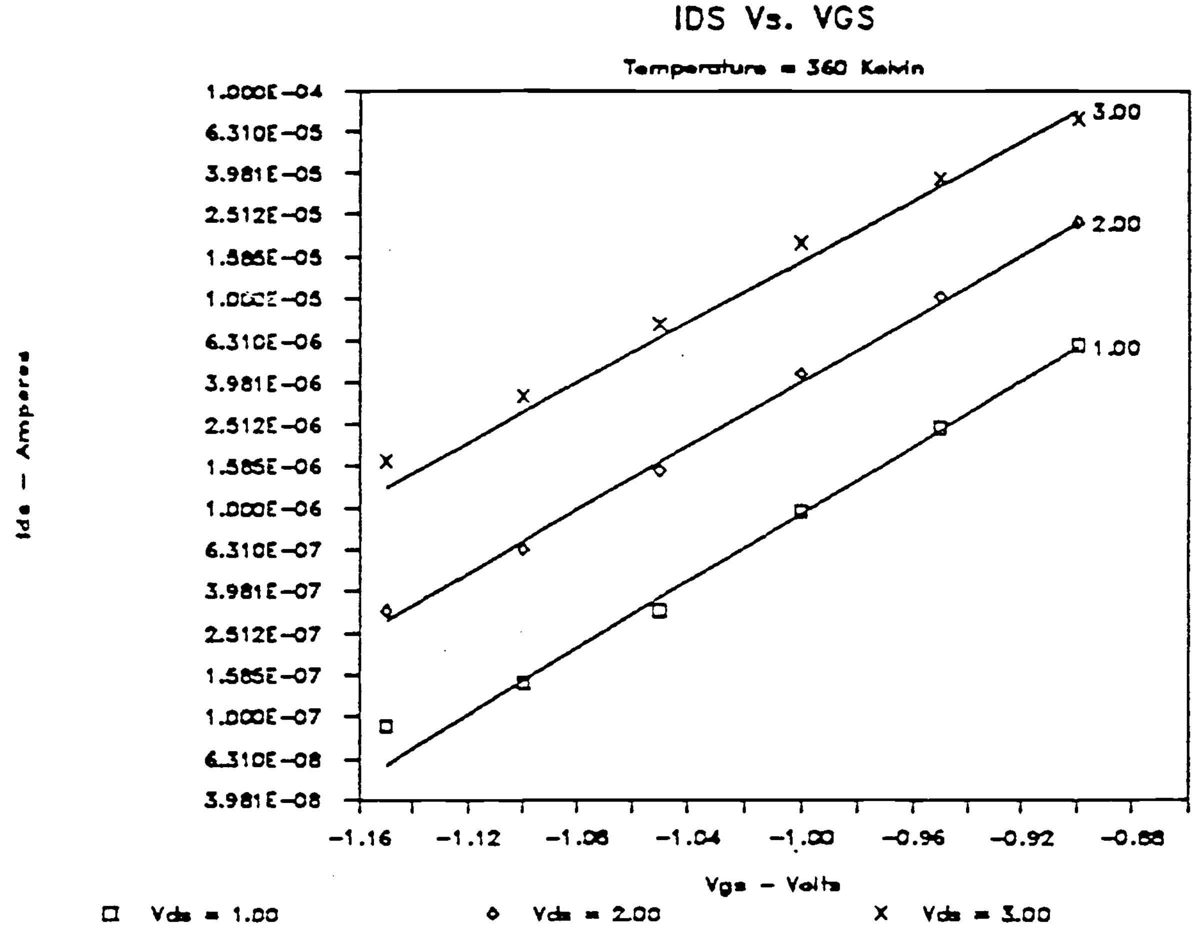


Figure 10a. Theoretical Data and Experemental Data Plots

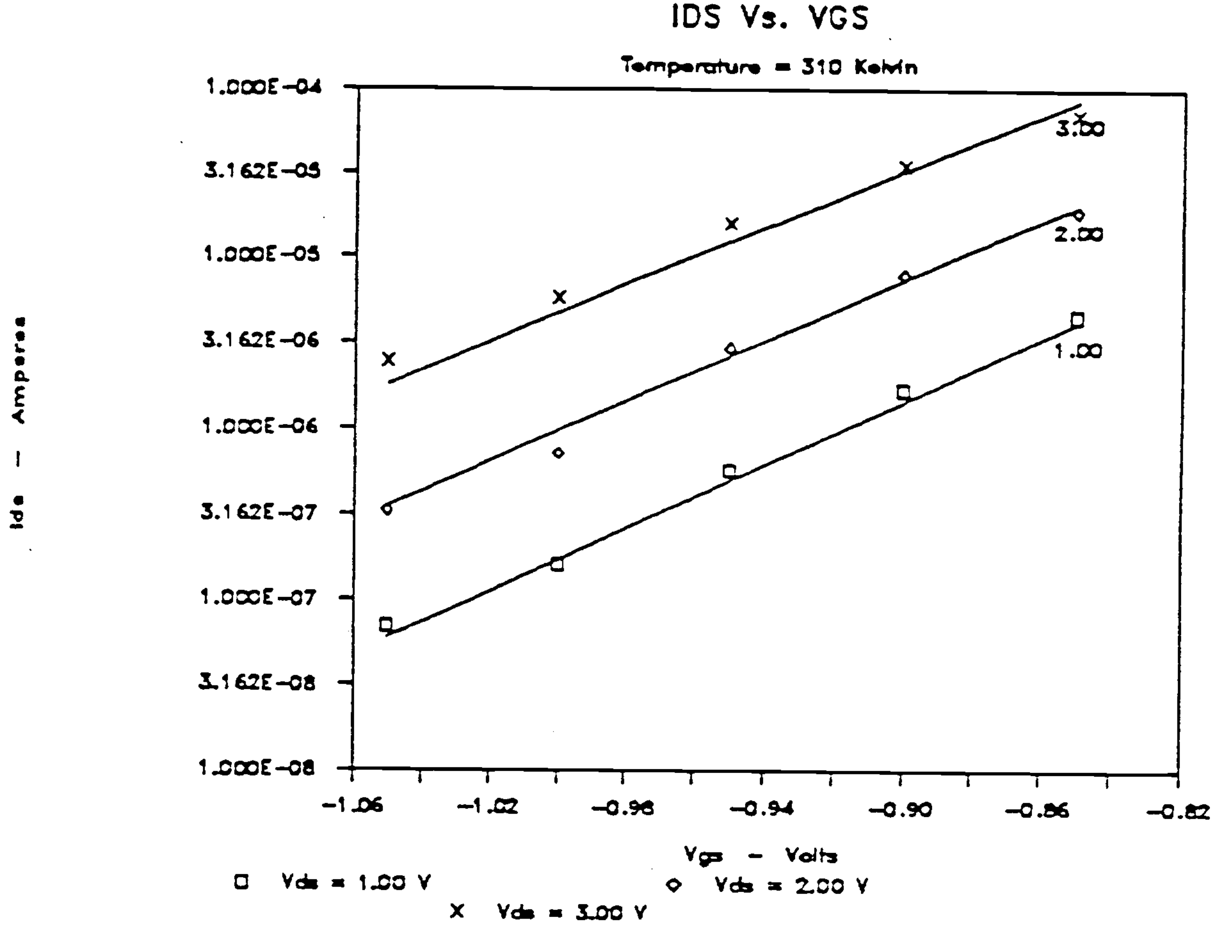


Figure 10b. Theoretical Data and Experemental Data Plots

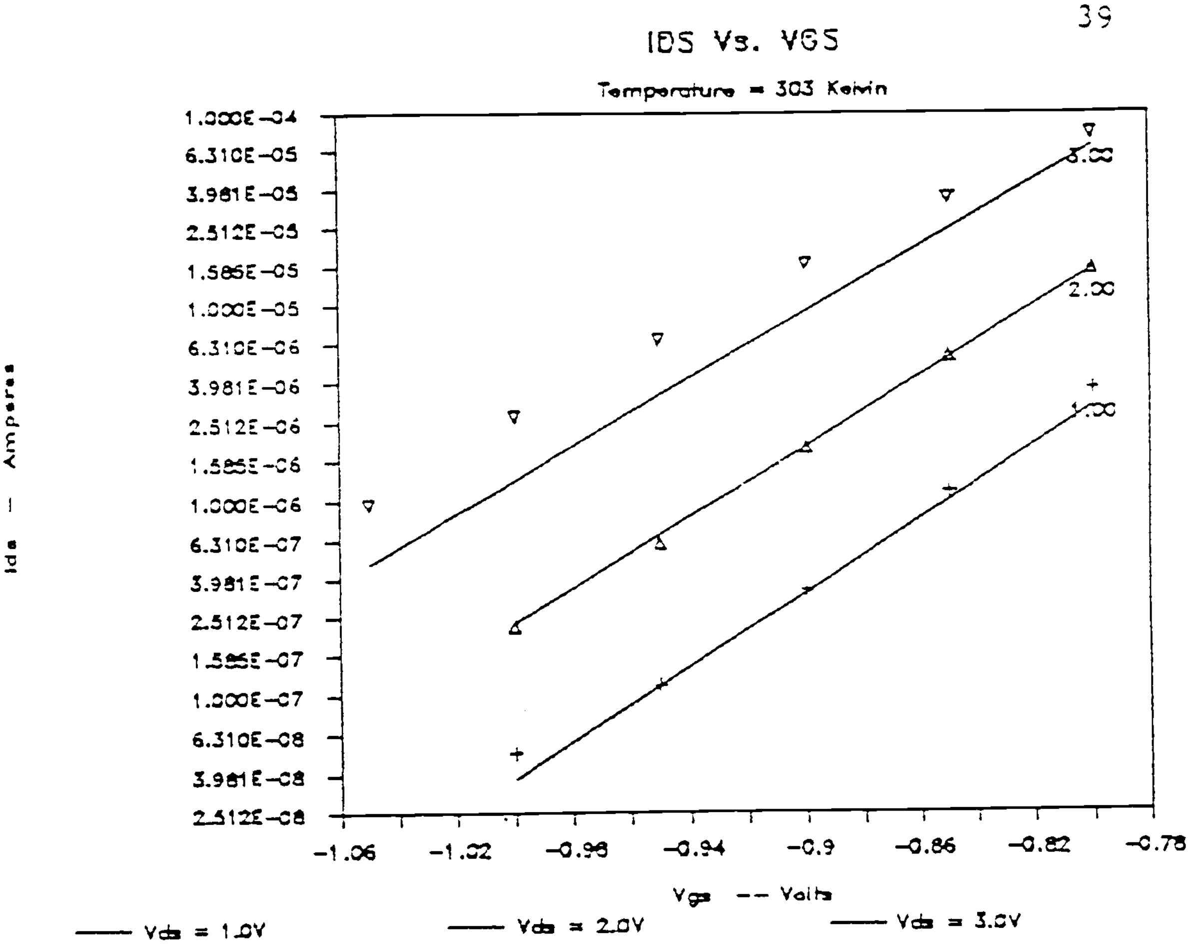


Figure 10c. Theoretical Data and Experimental Data Plots

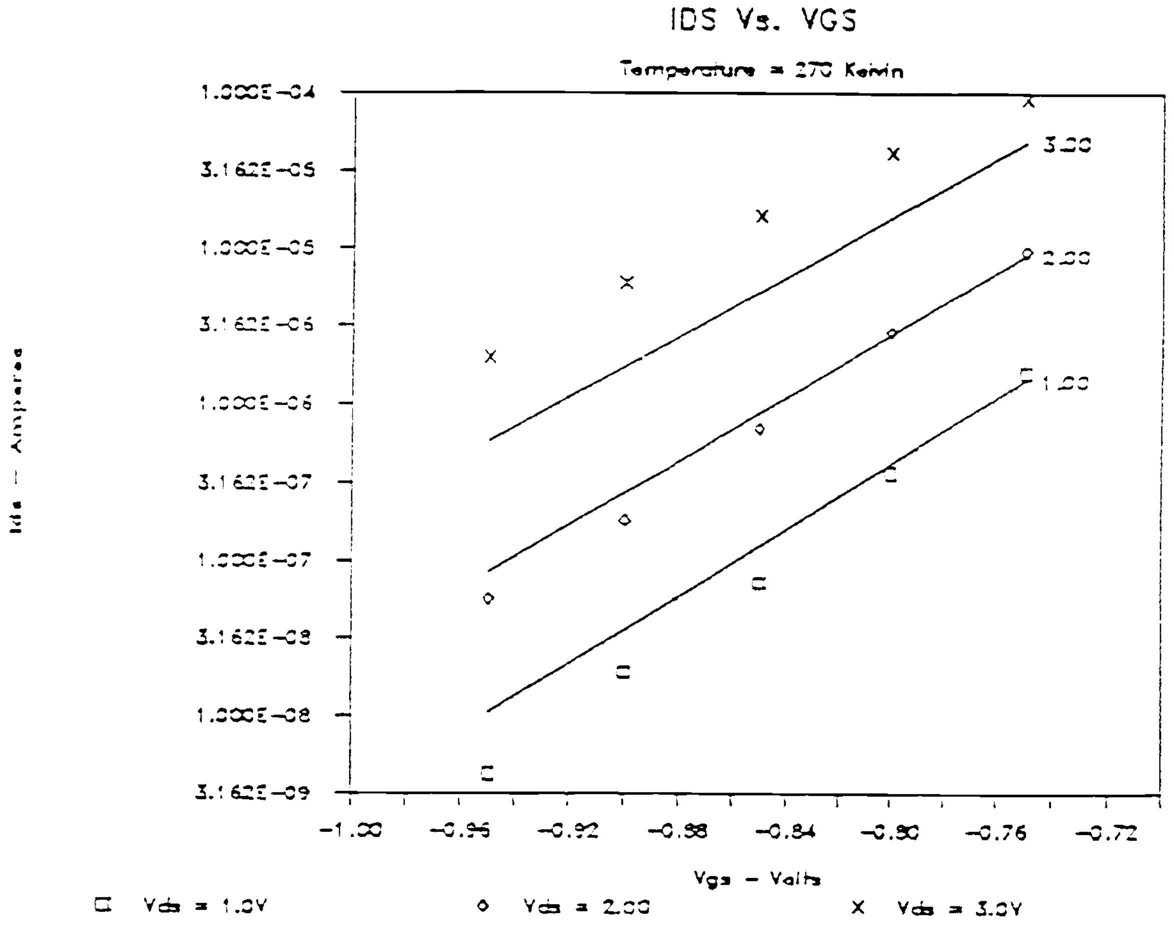


Figure 10d. Theoretical Data and Experimental Data Plots

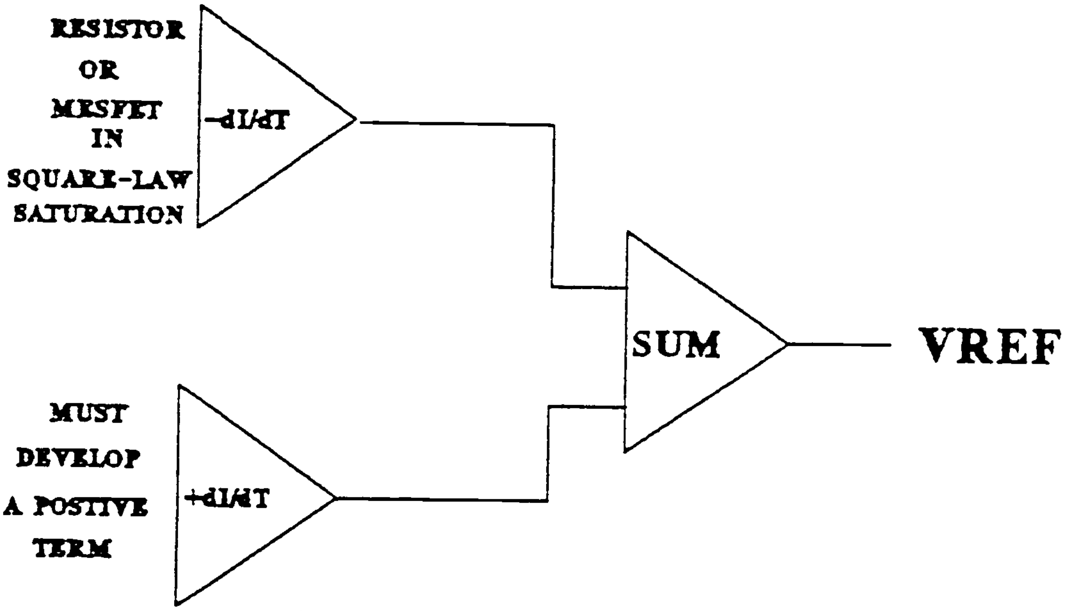


Figure 11. MESFET Voltage Reference Schematic

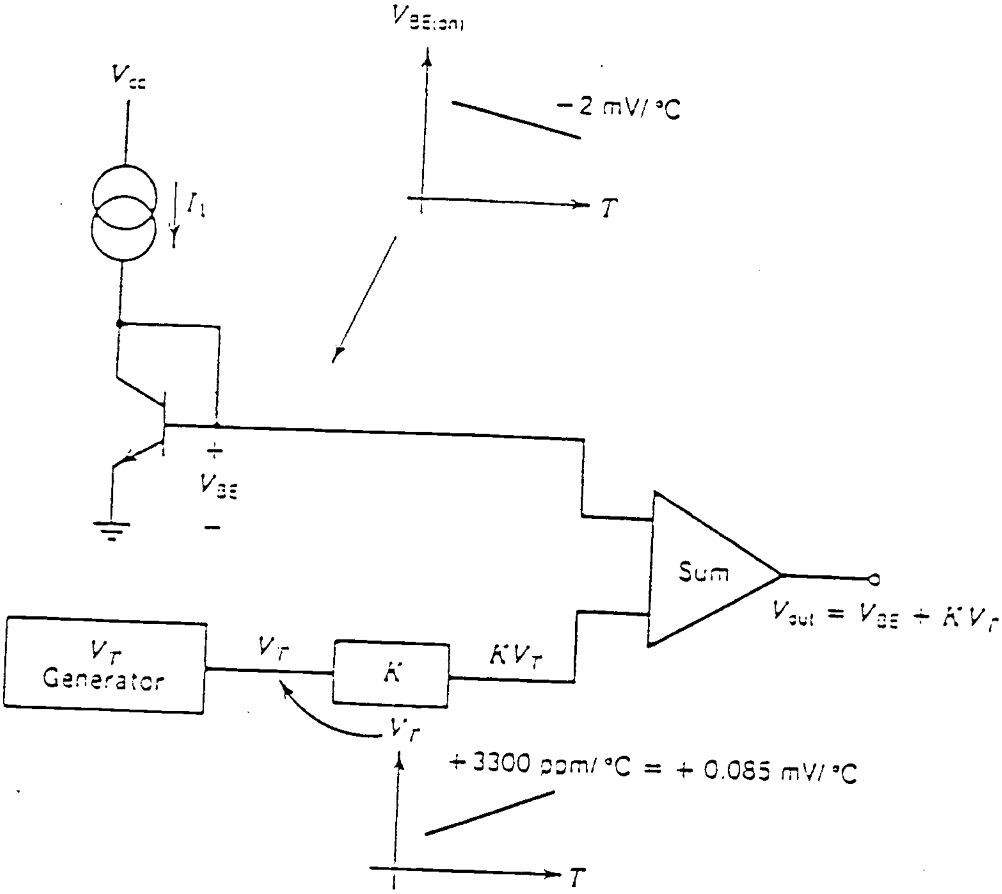


Figure 12. Bipolar Bandgap Voltage Reference Schematic

[After A. L. Grebene, pp 206-209]

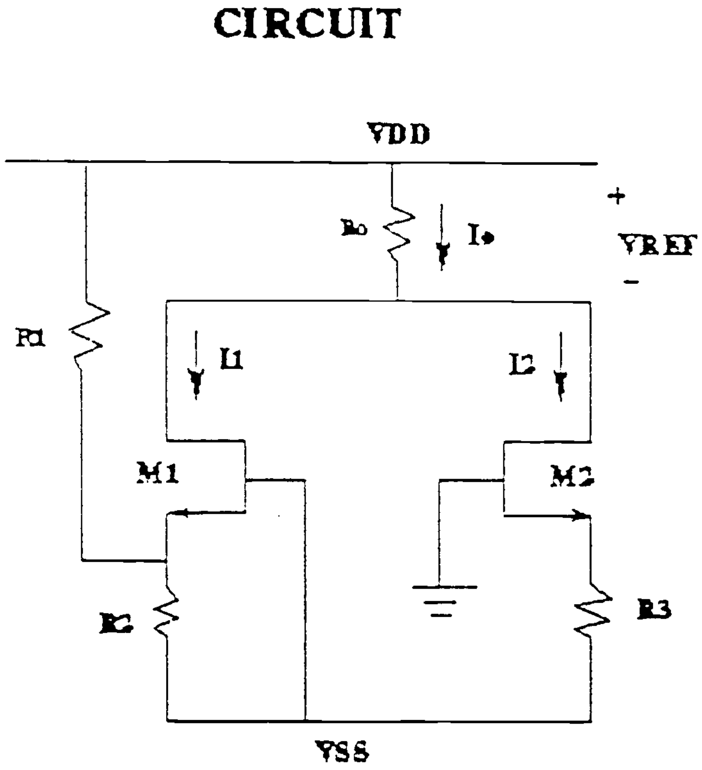


Figure 13. Circuit Implementation

VREFSIM

SIMPLE EXAMPLE

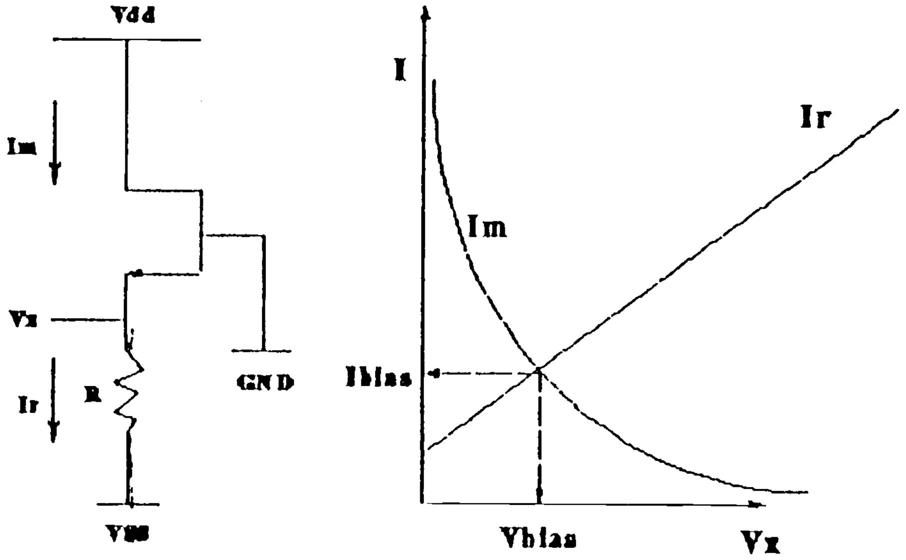


Figure 14. VREFSIM Iteration Technique

Square-Law Scaling by Width

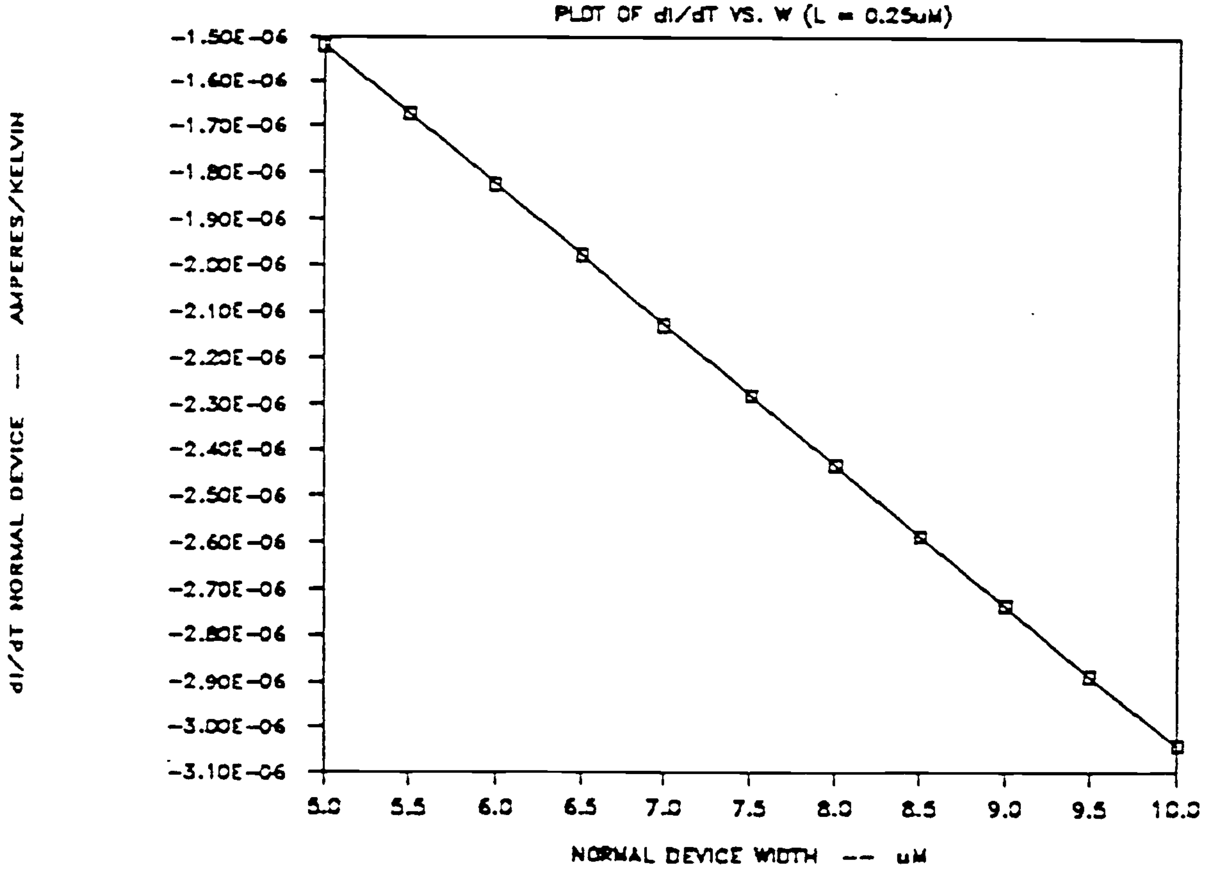


Figure 15. Square-Law Device Design Curve

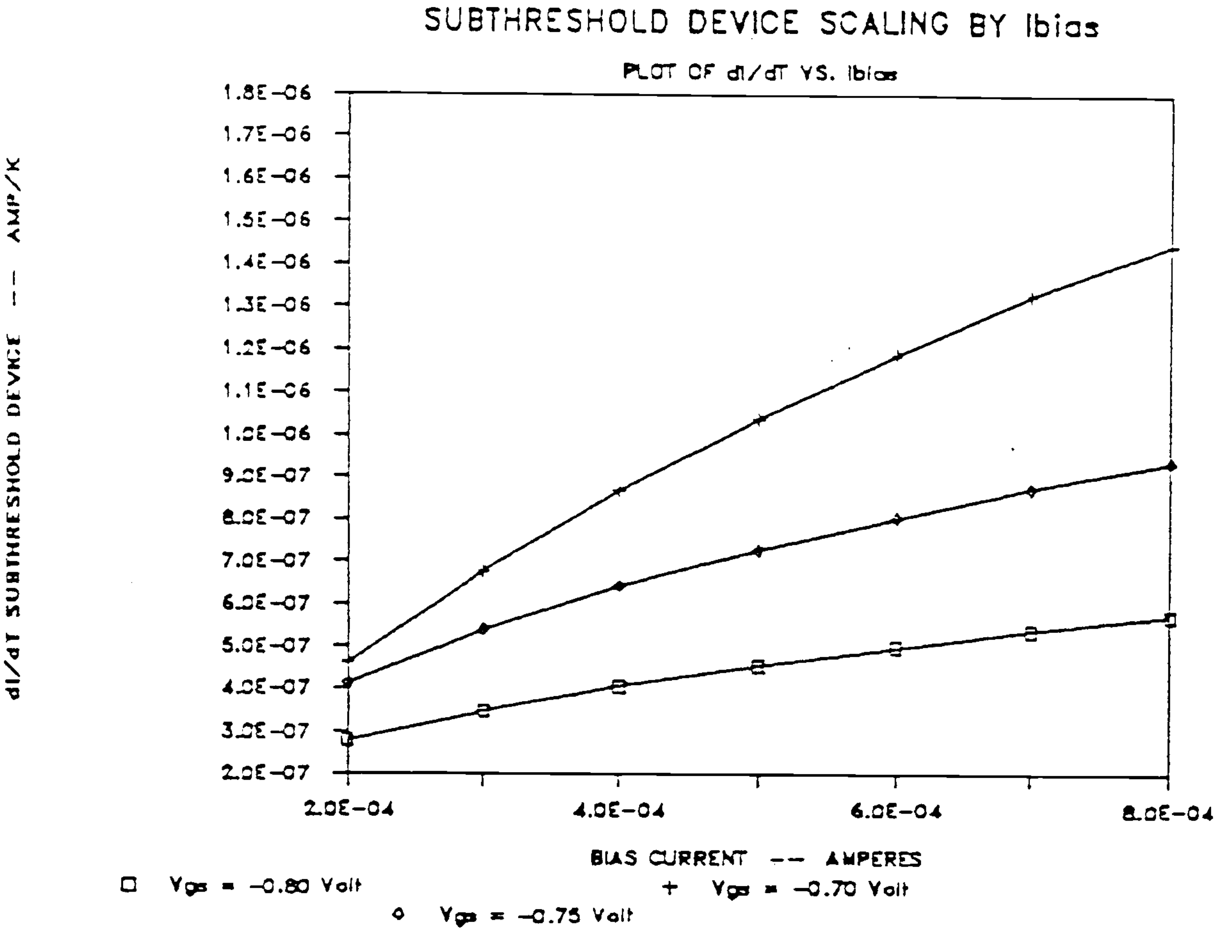


Figure 16. Subthreshold Device Design Curve

dI / dT OF SQUARE-LAW DEVICE

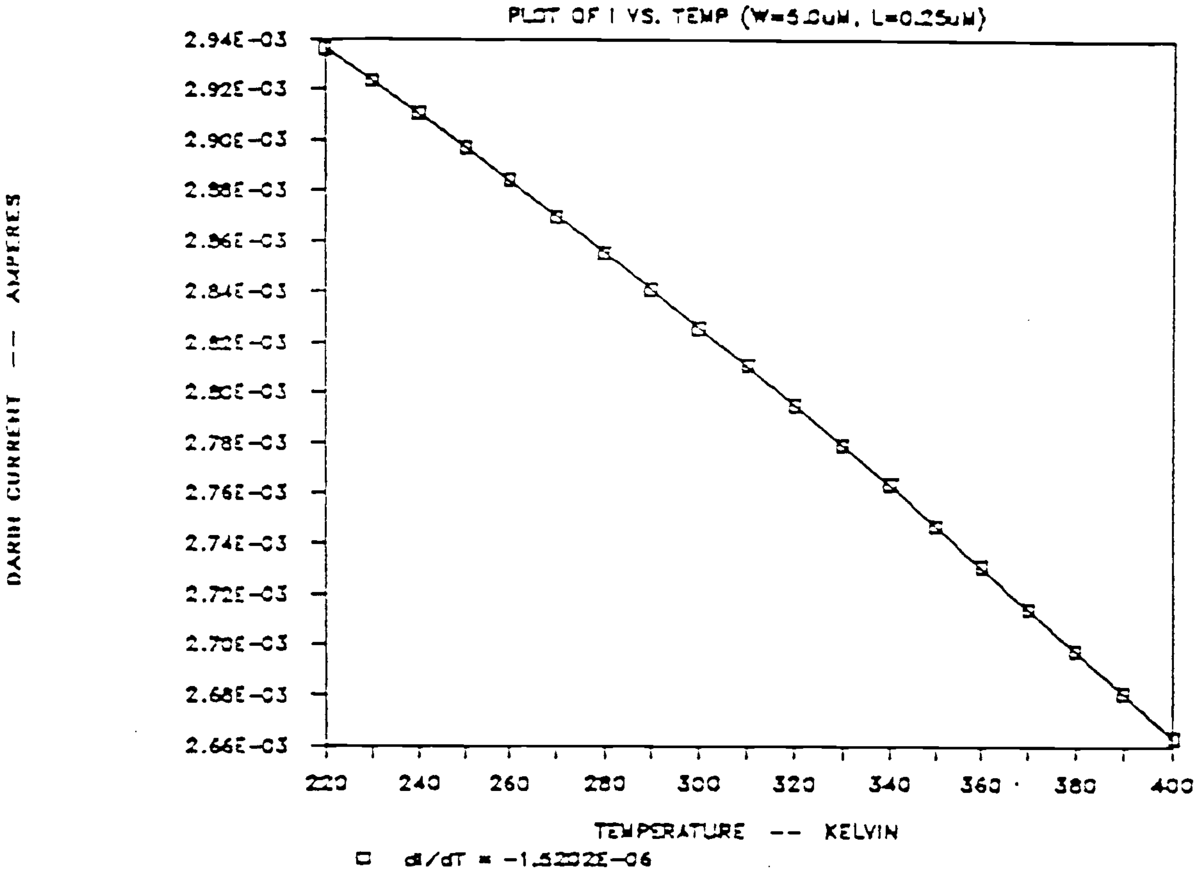


Figure 17. Square-Law Device Negative Temperature Drift Component

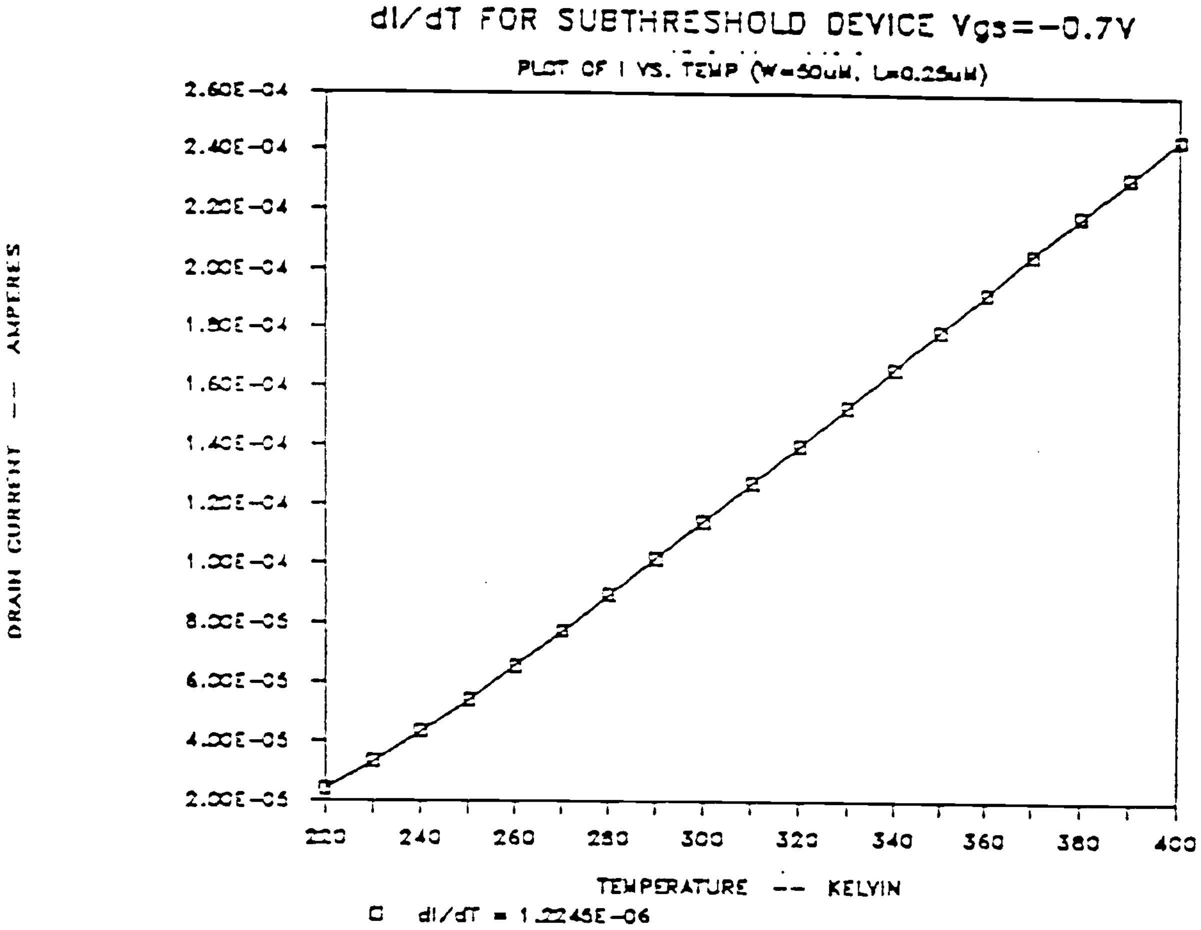


Figure 18. Subthreshold Device Positve Temperature Drift Component

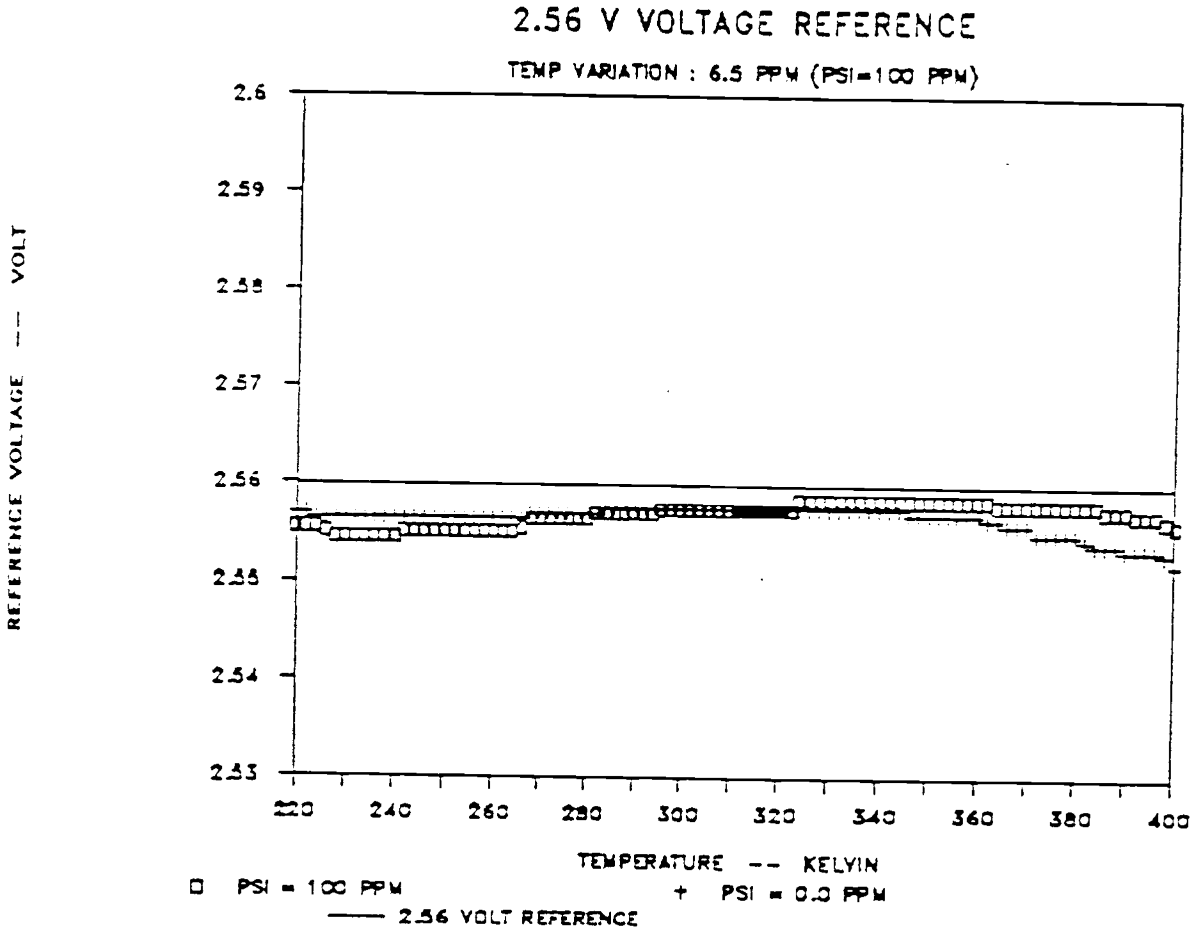


Figure 19. A 2.56 Volt Simulated Output

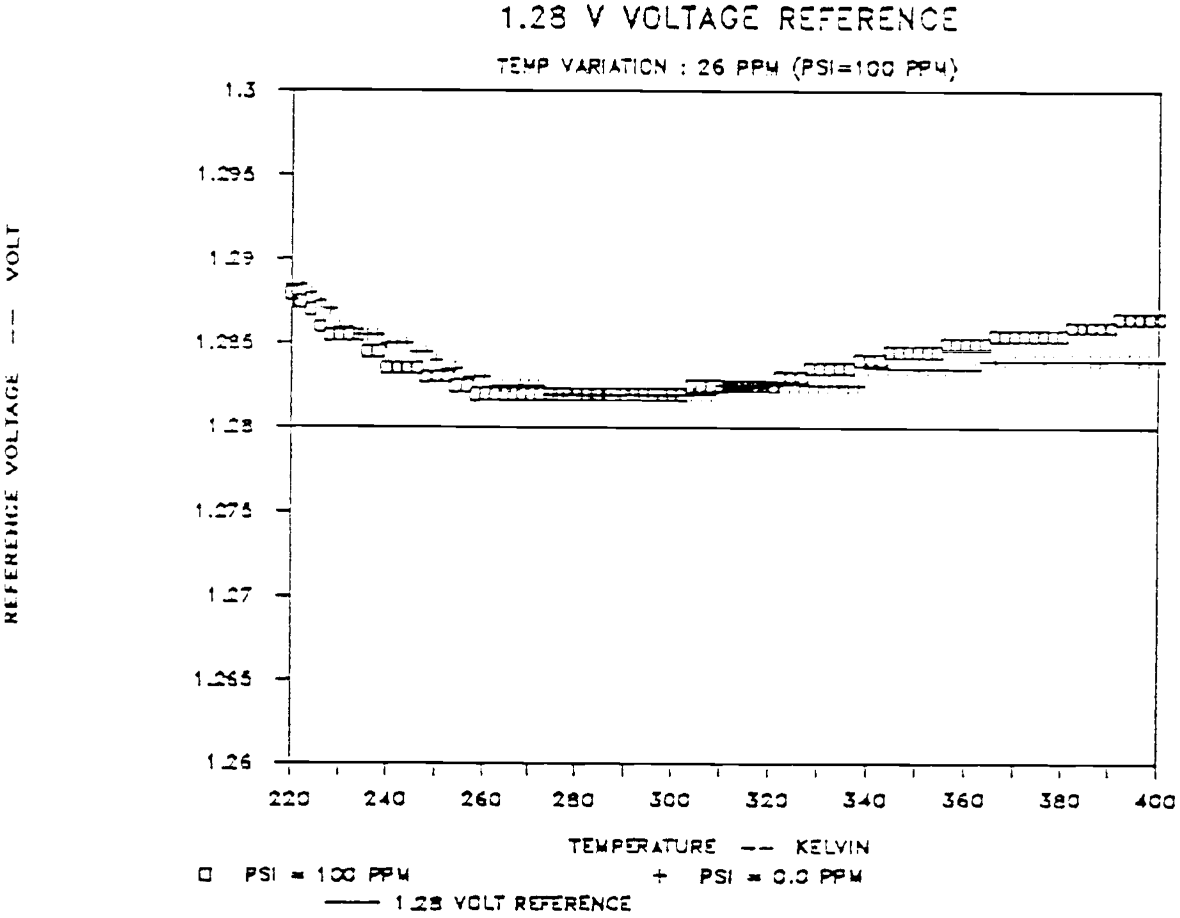
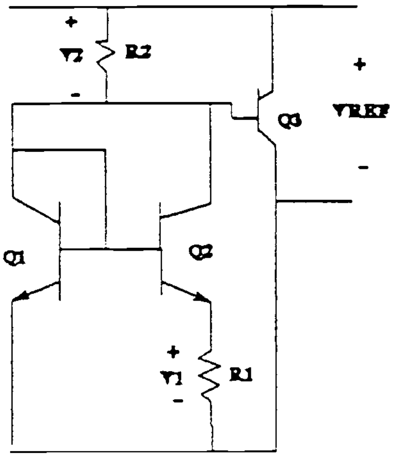


Figure 20. A 1.28 Volt Simulated Output

A. BIPOLAR TECHNOLOGY



TECHNIQUE : VOLTAGE SUMATION

PERFORMANCE : FOR TEMP : -66K TO 125K

VOLATGE VARIATION = 0.0000036 / KELVIN

OR 36 PPM / KELVIN

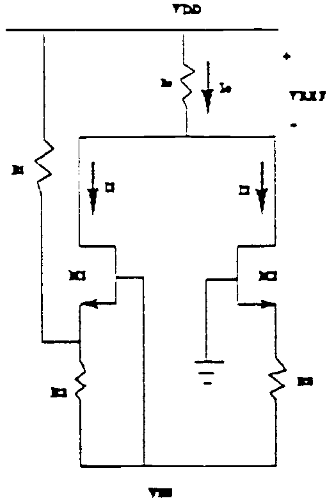
LIMITATIONS : FIXED REFERENCE AT 1.14V

NEED AMPLIFICATION CIRCUITRY,

DEGRADE PEFORMANCE

VOLTAGE DIVISION ON EXTERNAL RESISTOR STRING

B. MESFET TECHNOLOGY



TECHNIQUE : CURRENT SUMATION

PERFORMANCE : FOR TEMP = -66K TO 125K

VOLTAGE VARIATION = 0.0000065 / KELVIN

OR 6.5 PPM / KELVIN

AT 2.56 VOLTS

ADVANTAGES : VREF CAN BE CHOSEN WITHOUT

ANY ADDITIONAL COMPONENT COST

BUILT-IN AMPLIFICATION

INTERNAL VOLTAGE DIVISION - SAVE CHIP SPACE

Figure 21. Bipolar Versus MESFET Technology

6. BIBLIOGRAPHY

1. F. Lee, " A high-speed LSI 8 X 8 bit parallel multiplier", IEEE Journal of Solid State Circuits, Vol. **SC-17**, pp. 638-647, 1982.

2. N. Yokoyama, H. Onodera, T. Shiroki, H. Ohnishi and H. Nishi, " A 3-ns GaAs $K X 1 stsiac RAM", IEEE Transactions on Electron Devices, Vol. **ED-32**, pp. 1797-1801, 1985.
3. C. Kocot and C. A. Stolte, "Backgating in GaAs MESFET's", IEEE Transactions on Electron Devives, Vol. **ED-29**, No. &, pp. 1059-1064, July, 1982.

4. C. P. Lee and M. F. Chang, " Temperature Dependence of Backgating Effect in GaAs Integrared Circuits", IEEE Electron Device Letters, Vol. **ED-6**, No. 8, pp. 428-430, Aug. 1985.

5. P. Andrzej, C. Chen, M Shur, S. Baier, " Modeling and Characterization of Ion-Implanted GaAs MESFET's", IEEE Transaction on Electron Devices, Vol. **ED-34**, No. 4, pp.726-732, April 1987.

6. C. M. Snowden, D. Loret, "Two-Dimensional Hot-Electron Models for Short-Gate Length GaAs MESFET's", IEEE Transaction on Electron Devices, Vol. **ED-34**, No. 2, pp.212-223, Feb. 1987.

7. P. George, K. Hui, P. Ko and C. Hu,"A GaAs MESFET Model for Circuit Simulation", IEEE 1987 Custom Integrated Circuits Conference, pp. 409-412, 1987.

8. H. Statz, P. Newman, I. W. Smith, R. A. Pucel and H. A. Haus, "GaAs FET Device and Circuit simulation in SPICE", IEEE Transaction on Electron Devices, Vol. **ED-34**, No.2, pp. 160-169, Feb. 1987.

9. L. E. Larson, "An Improved GaAs MESFET Equivalent Circuit Model for Analog Integrated Circuit Applications", IEEE Journal of Solid State Circuits, Vol. **SC-22**, NO. 4, May 1980.

10. W.R. Curtice, " A MESFET Model for use in the design of GaAs Integrated Circuits", IEEE Transactions on Microwave Theory and Techniques, Vol. **MTT-28**, No. 5, pp. 448-456, May 1980.

11. W. R. McKinley, "GASSIM: A Circuit Simulator for Large Scale GaAs Circuits", IEEE 1986 Custom Integrated Circuits Conference, pp. 628-631, 1986.

12. C. Chang, T. Vrotos, M. T. Frizzell and R. Caroll, " A Substhreshold Current Model for MESFET's", IEEE

Electron Device Letters, Vol. EDL-8, No. 2, pp. 69-72, Feb. 1987.

13. P. C. Canfield, J. Medinger, D. J. Allstot, L.Forbes, A. J. McCamant, B. A. Vetanen, B. Odekrik, E. P. Finchem and K. R. Gleason, "High Speed Quarter Micron Buried-Channel MESFETs with Improved Output Characteristics for Analog Applications", 11th IEEE/Cornell Conference on Advanced Concepts in High Speed Semiconductor Devices and Circuits, 1987 (in pree).

14. R. R. Troutman, "VLSI Limitations from Drain-Induced Barrier Lowering", IEEE Transactions on Electron Devices, Vol. ED-26, No. 4, pp. 461-468, April 1979.

15. C. D. Hartgring, "An Accurate JFET/MESFET Model for Circuit Analysis", Solid State Electronics, Vol. 25, No. 3, pp. 233-240, 1982.

16. S. M. Sze, Physics of Semiconductor Devices, pp. 314-340, John Wiley & Sons, 1984.

17. M. Rocchi, "Surface and Bulk Parasitic Effects in GaAs ICs", Physica 129B, 1985.

18. M. Rocchi, " Statue of Parasitic Effects in GaAs IC", Physica 129B, 1985.

19. A. L. Grebene, Bipolar and MOS Analog Integrated Circuit Design, pp. 206-209, John Wiley & Sond, 1984.

20. R. J. Widlar, "New Developments in IC Voltage Regulators", IEEE Journal of Solid State Circuits, Vol. SC-6, pp. 2-7, Feb. 1971.

APPENDICES

7. APPENDICES

7.1 Equation Derivations

7.1.1 Equation (20a)

The Curtice model was used to implement the square-law device in saturation region. From Equation (21),

$$Ids = \beta(Vgs - Vt)^2 (1 + \alpha Vds) \tanh(\sigma Vds). \qquad (20.0)$$

From Equation (16),

$$\beta = \left(\frac{2\ W\ Uc\epsilon s\ 292^2}{3\ L\ t\ T^2}\right) \qquad (20.1)$$

The mobility, Uc, as expressed in Equation (24) is inserted into the above equation. Beta is expressed as,

$$\beta = \left(\frac{2\ U_{300}{*}T_{300}\epsilon s\ 292^2}{3\ T\ t\ T^2}\right) = \frac{A}{T^3}, \qquad (20.2)$$

where,

$$A = \left(\frac{2\ U_{300}{*}T_{300}\epsilon s\ 292^2}{3\ t}\right). \qquad (20.3)$$

The threshold voltage is :

$$Vt = Vto - \Phi Vds, \qquad (20.4)$$

and from Equation (13),

$$Vto = Vbi - Vp. \qquad (20.5)$$

Equation (15) gives Vp as:

$$Vp = \frac{qNct}{2\ \epsilon s}\ \frac{T}{292}\ . \qquad (20.6)$$

Combining Equation (20.4), (20.5), and (20.6) :

$$\begin{aligned} Vt &= Vbi - Vp - \Phi Vds \\ &= kT/q{*}Ln(Nc/ni) - (qNct^2/2\ s)(T/292) - \Phi Vds \\ &= T[(k/q{*}Ln(Nc/ni) - (qNct^2/2\ s292)] - \Phi Vds \\ &= BT - \Phi Vds, \qquad (20.7) \end{aligned}$$

where,

$$B = T[(k/q{*}Ln(Nc/ni) - (qNct^2/2\epsilon s292)]. \qquad (20.8)$$

Finally, inserting Equation (20.7) :

$$\begin{aligned} (Vgs - Vt)^2 &= (Vgs - BT + \Phi Vds) \\ &= (C - BT), \qquad (20.9) \end{aligned}$$

where,

$$C = (Vgs - \Phi Vds) \qquad (20.10)$$

$$(1 + \alpha Vds)\ Tanh(\sigma Vds) = D \qquad (20.11)$$

Expressing Equation (21) in terms of the temperature independent variables A, B, and C we have,

$$Ids = (A/T^3)(C - BT)^2\ D. \qquad (20a)$$

7.1.2 Equation (23a)

For the subthreshold device, the current model is :

$$Idsub = Io\ \exp[(q/kTNs)(1 - \Gamma Vds)(Vgs - Vto + \Phi Vds)] \quad (23.0)$$

The exponential term is expressed as,

$$(q/kTNs)(1 - \Gamma Vds) = A/T, \qquad (23.1)$$

and,

$$(Vgs - Vto + \Phi Vds) = (B - Vto). \qquad (23.2)$$

Where,

$$A = (q/kNs)(1 - \Gamma Vds), \qquad (23.3)$$

$$B = (Vgs + \Phi Vds). \qquad (23.4)$$

Inserting Equations (14) and (15) into Vto, we have an explicit temperature term:

$$Vto = T(k/qLn[Nc/ni] + qNct^2/584\epsilon s). \qquad (23.5)$$

Letting the term in Equation (23.5) equal C, Equation (23.0) is expressed in terms of the temperature independent variables A, B, and C as,

$$Idsub = Io\ Exp[A/T^*(B - CT)]. \qquad (23.6)$$

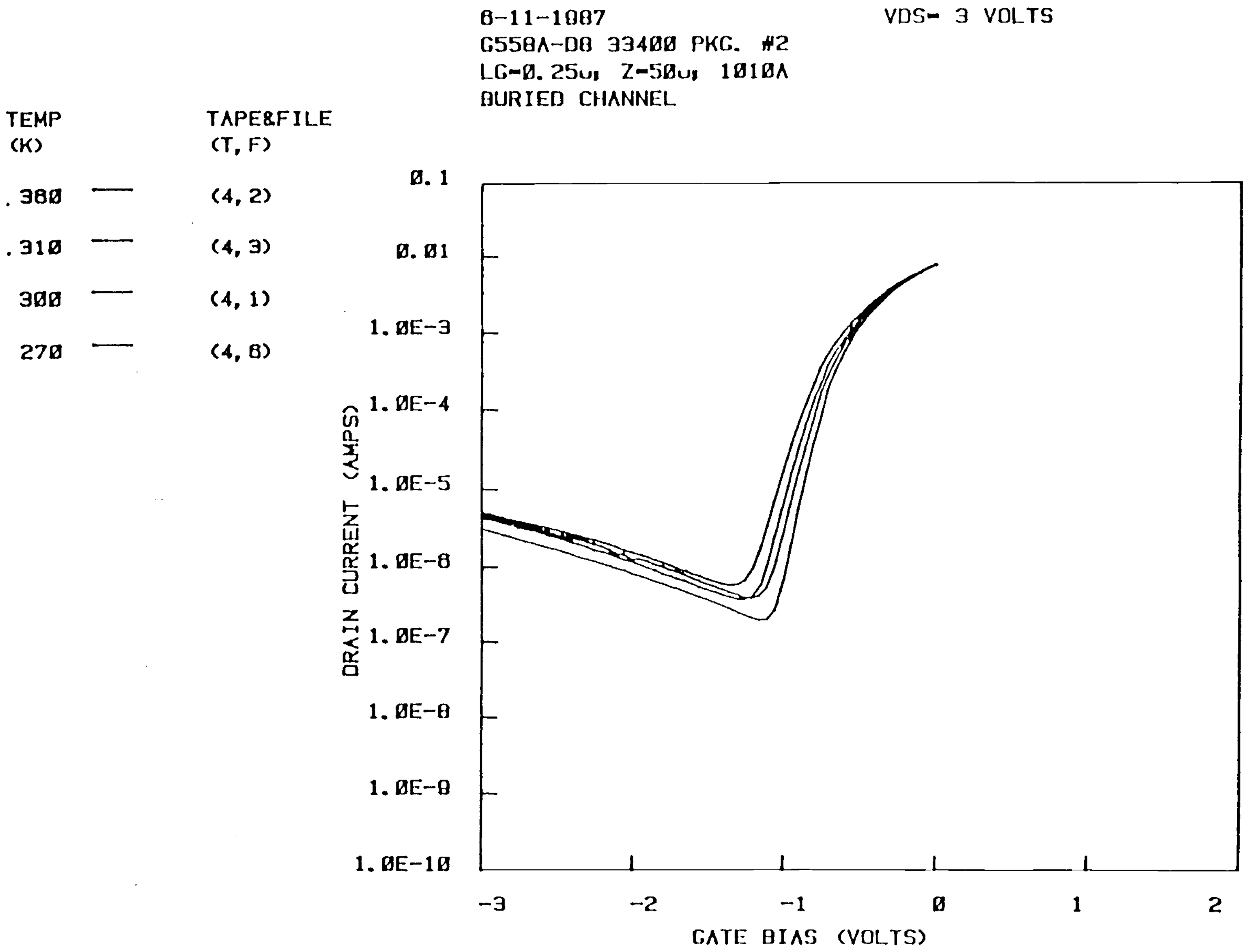
Finally, rearranging the exponential terms in Equation (23.6) :

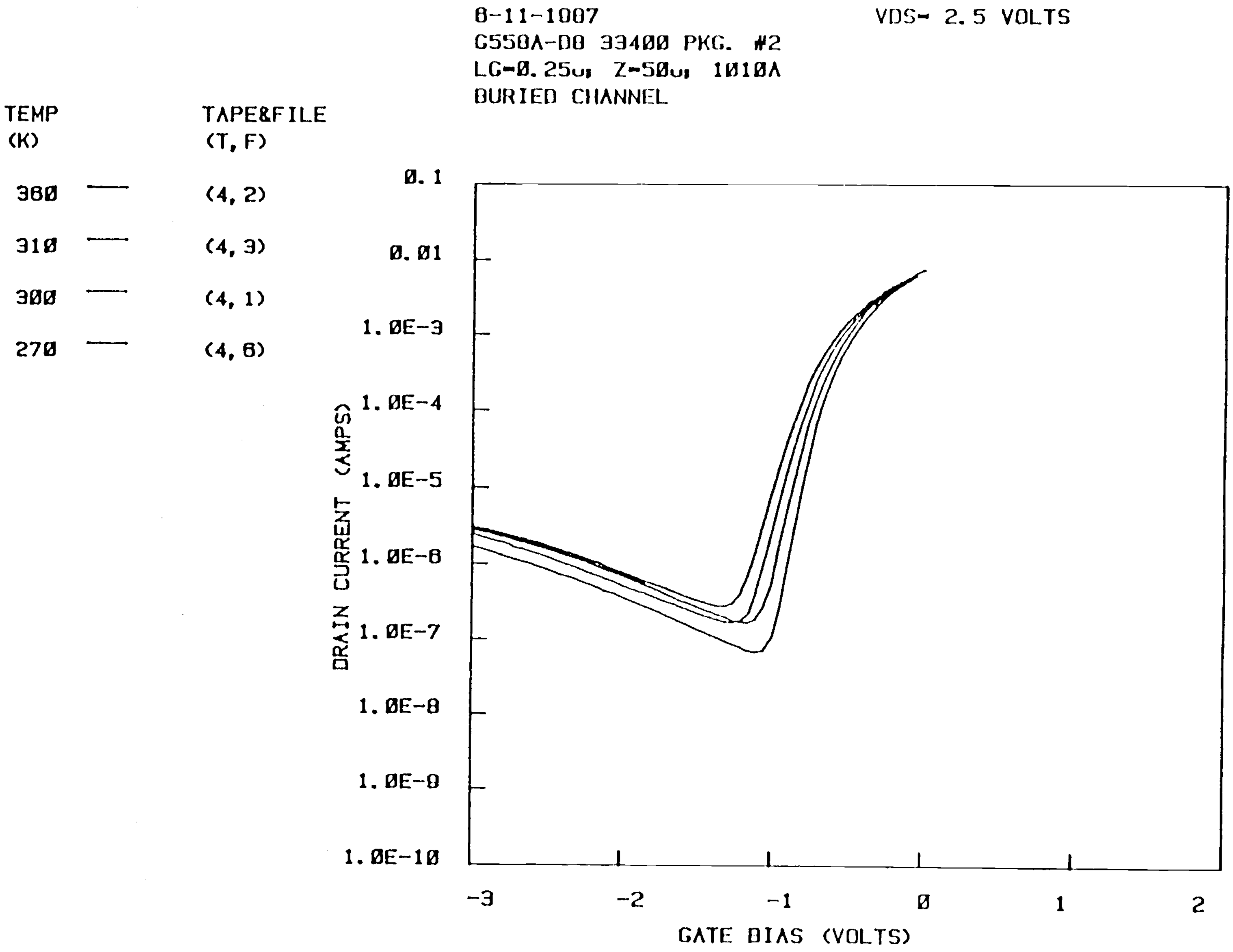
$$Idsub = Io\ Exp[-AB]\ Exp[AC/T] \qquad (23a)$$

7.2 List of Equipment

The work station consists of a Tektronix 4054A computer, Keithley 192 electrometer, two Tektronix PS 5004 programmable precision power supply, a Tektronix FG 5010 Programmable 20MHz Function Generator, a Tektronix DM 5010 Programmable Digital Multimeter, a Tektronix SI 5010 Programmer Scanner, a Air Product Temperature controller, and a Helium closed cycle refrigeration unit. Figure 9 shows the experimental work station set up in the laboratory.

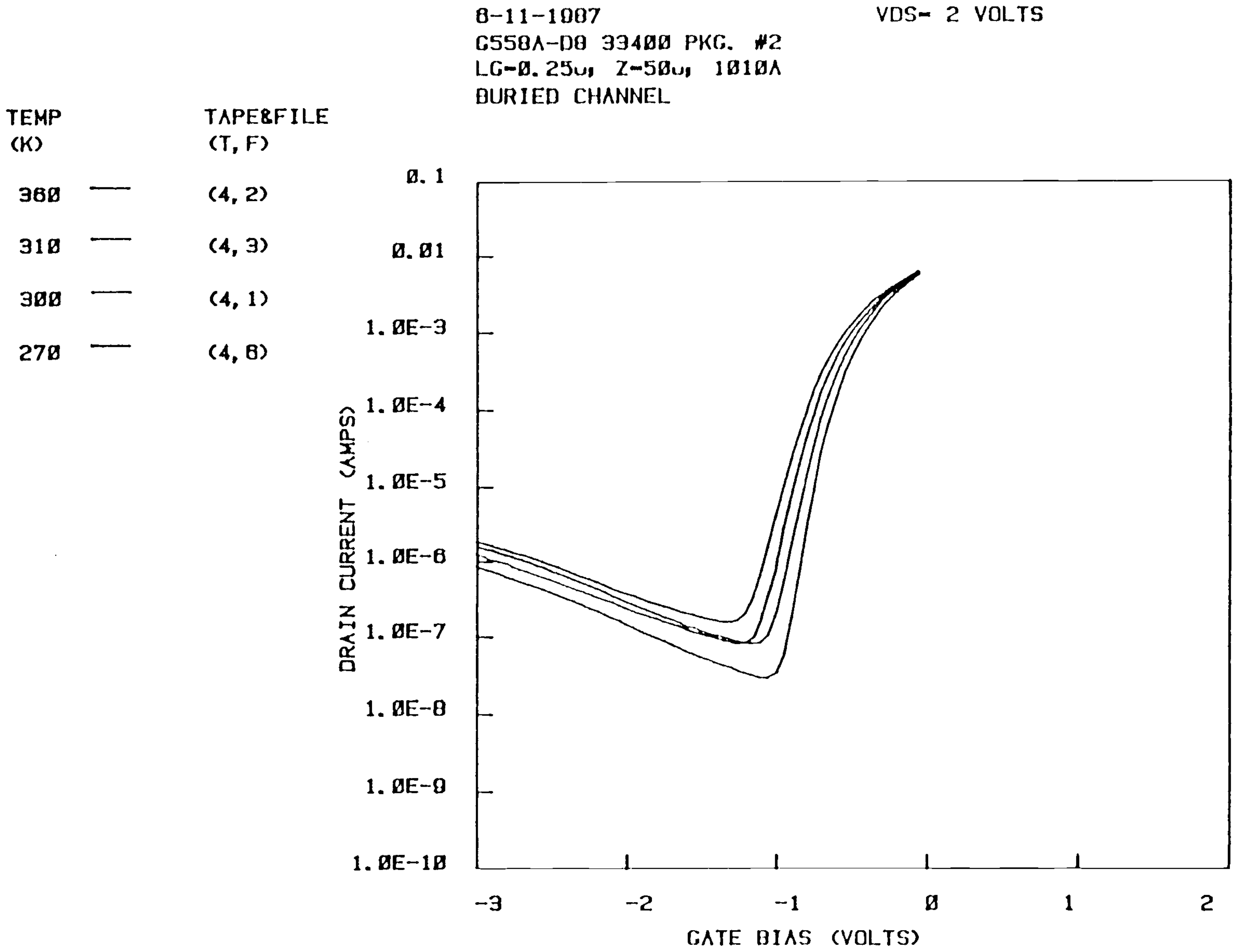
7.3 List of Experimental data

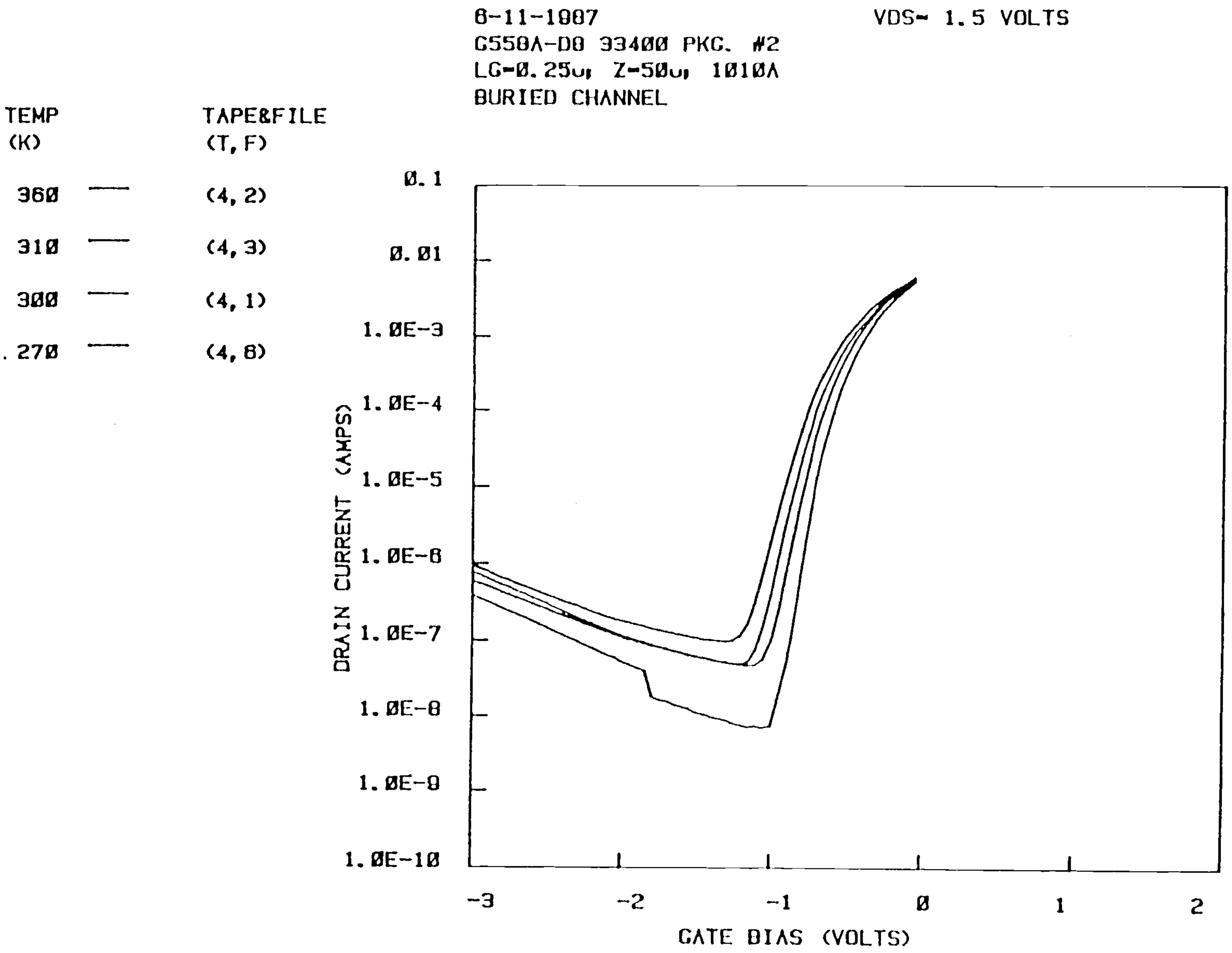




8-11-1987
G550A-D8 33400 PKG. #2
LG=0.25u, Z=50u, 1010A
BURIED CHANNEL

VDS= 2.5 VOLTS


8-11-1987
G558A-D8 33400 PKG. #2
LG=0.25u, Z=50u, 1010A
BURIED CHANNEL
VDS= 2 VOLTS
TEMP (K)
TAPE&FILE (T, F)
360 (4, 2)
310 (4, 3)
300 (4, 1)
270 (4, 6)
0.1
0.01
1.0E-3
1.0E-4
1.0E-5
1.0E-6
1.0E-7
1.0E-8
1.0E-9
1.0E-10
DRAIN CURRENT (AMPS)
-3
-2
-1
0
1
2
GATE BIAS (VOLTS)



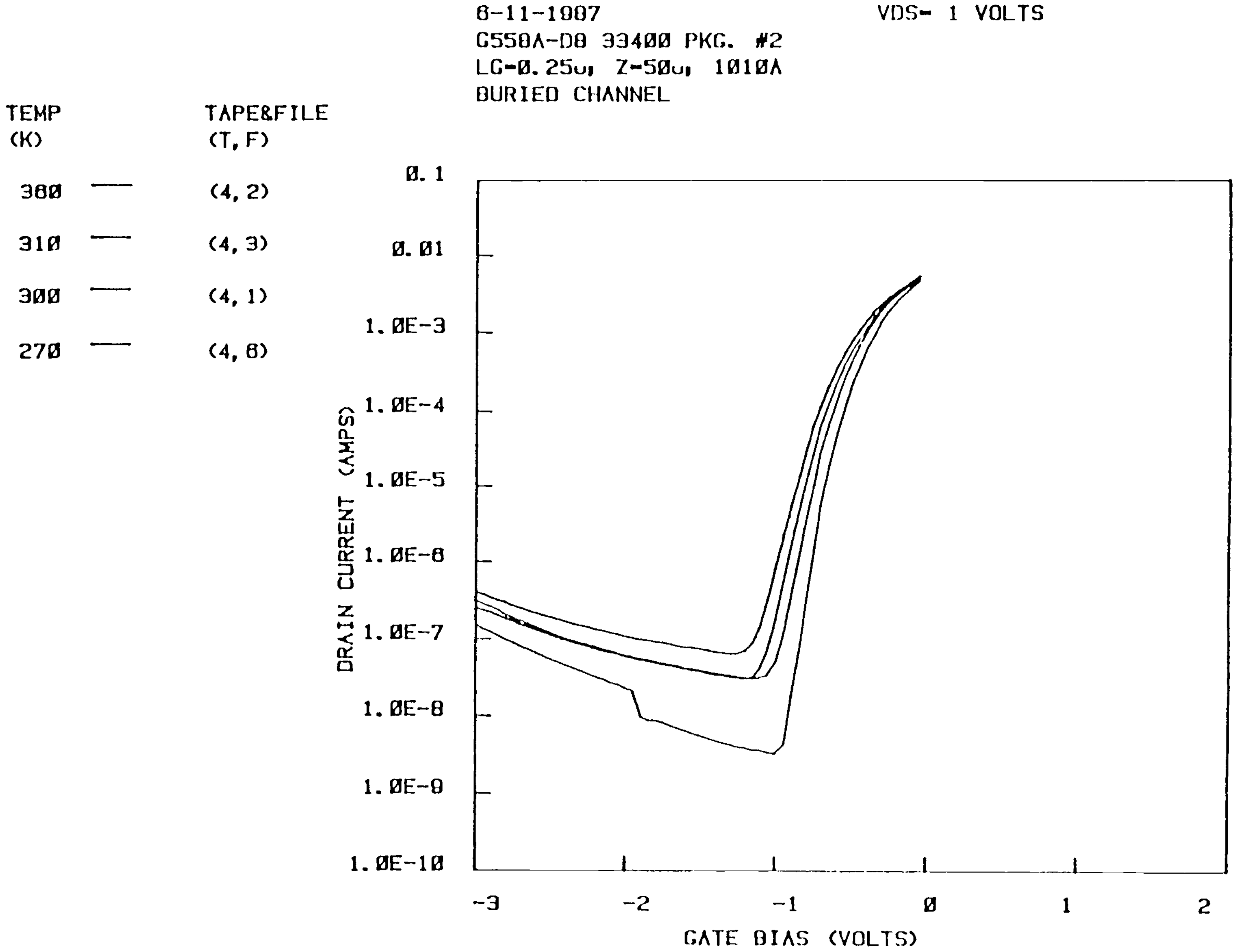
6-11-1987

G558A-D6 33400 PKG. #2

LG-0.25u, Z-50u, 1010A

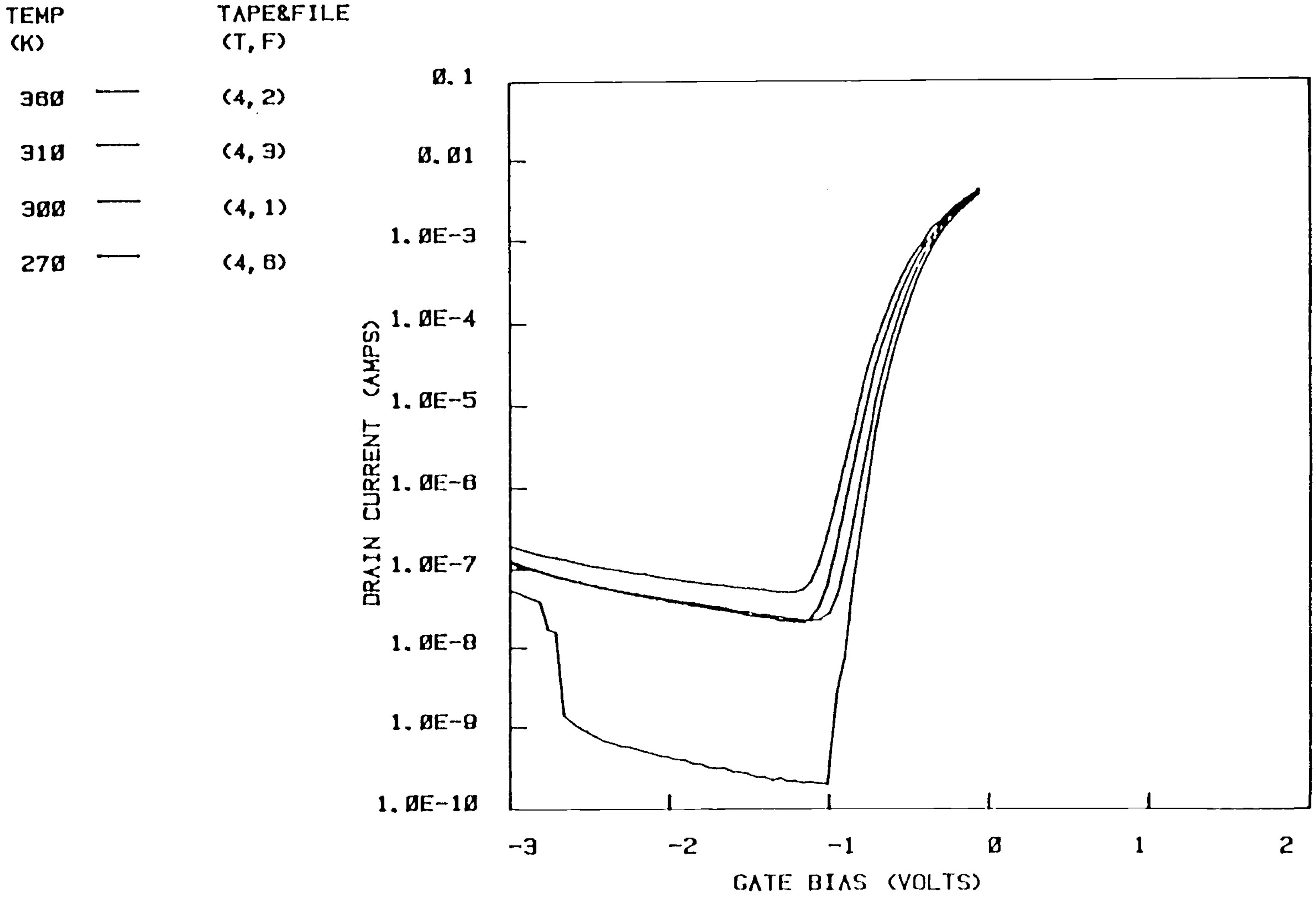
BURIED CHANNEL

VDS- 1.5 VOLTS

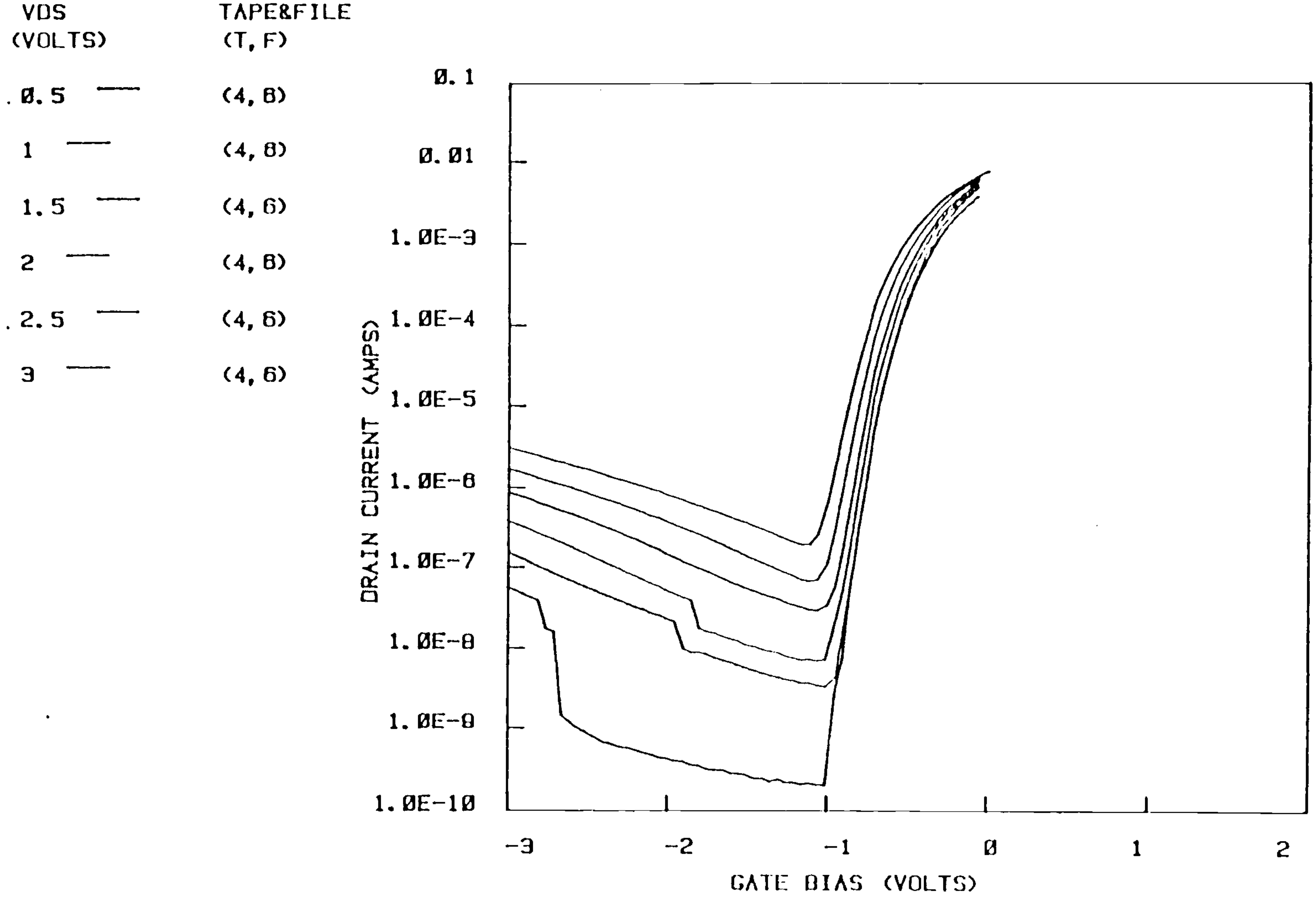

8-11-1987
VDS= 1 VOLTS
G558A-D8 33400 PKG. #2
LG=0.25u, Z=50u, 1010A
BURIED CHANNEL
TEMP (K)
TAPE&FILE (T,F)
380 (4,2)
310 (4,3)
300 (4,1)
270 (4,8)
0.1
0.01
1.0E-3
1.0E-4
1.0E-5
1.0E-6
1.0E-7
1.0E-8
1.0E-9
1.0E-10
DRAIN CURRENT (AMPS)
-3
-2
-1
0
1
2
GATE BIAS (VOLTS)

8-11-1987
G558A-D8 33400 PKG. #2
LG=0.25μ, Z=50μ, 1010A
BURIED CHANNEL

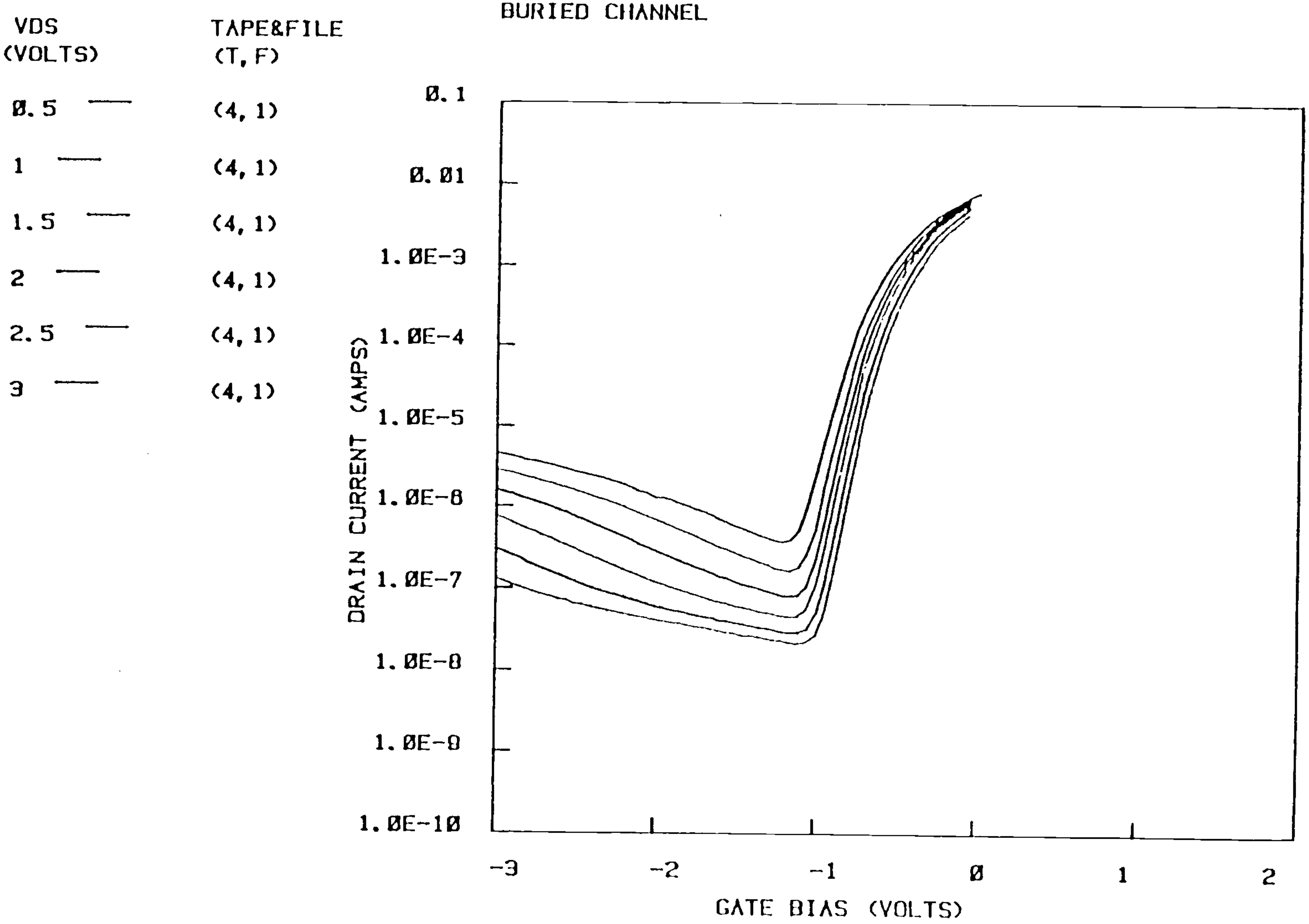
VDS= 0.5 VOLTS

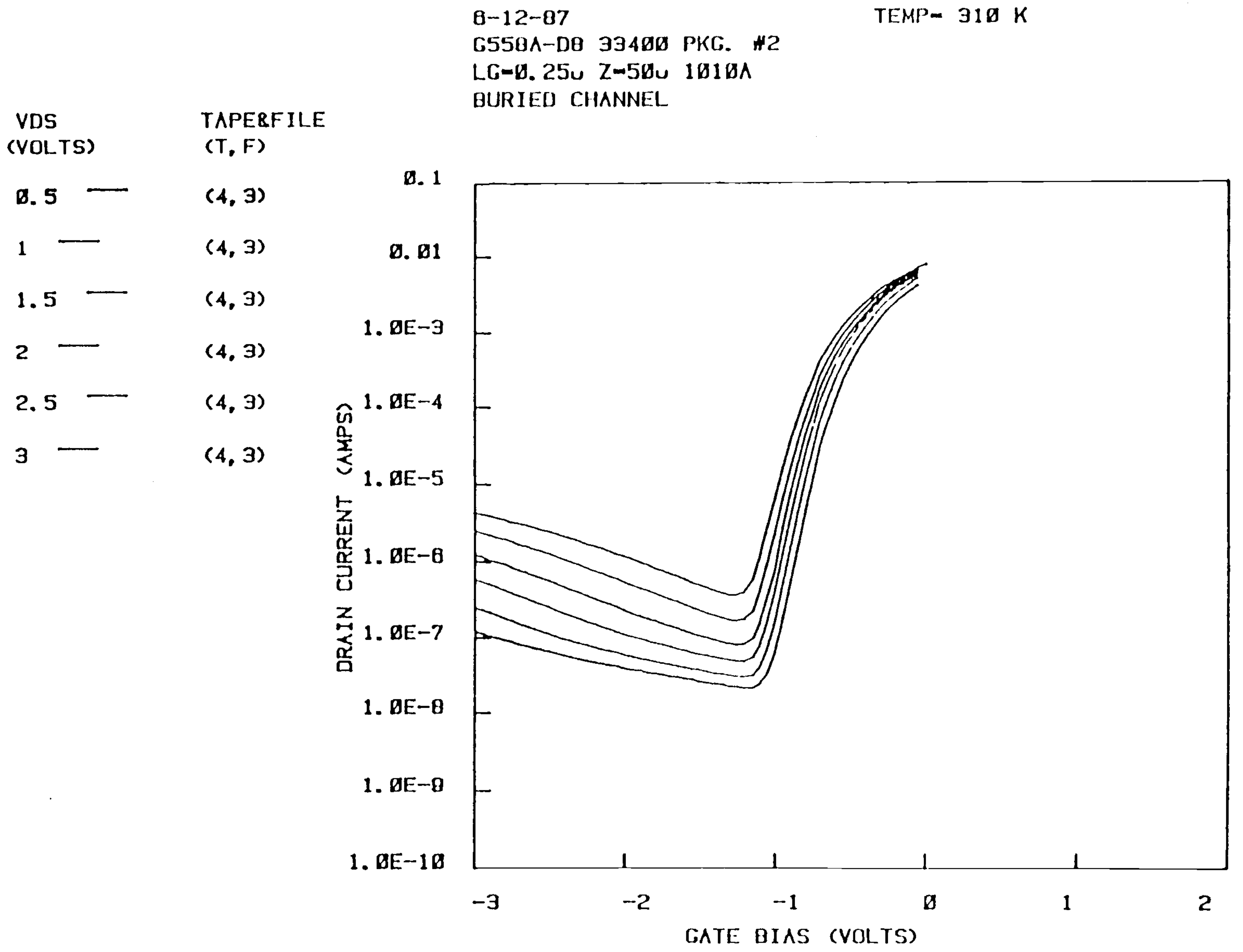


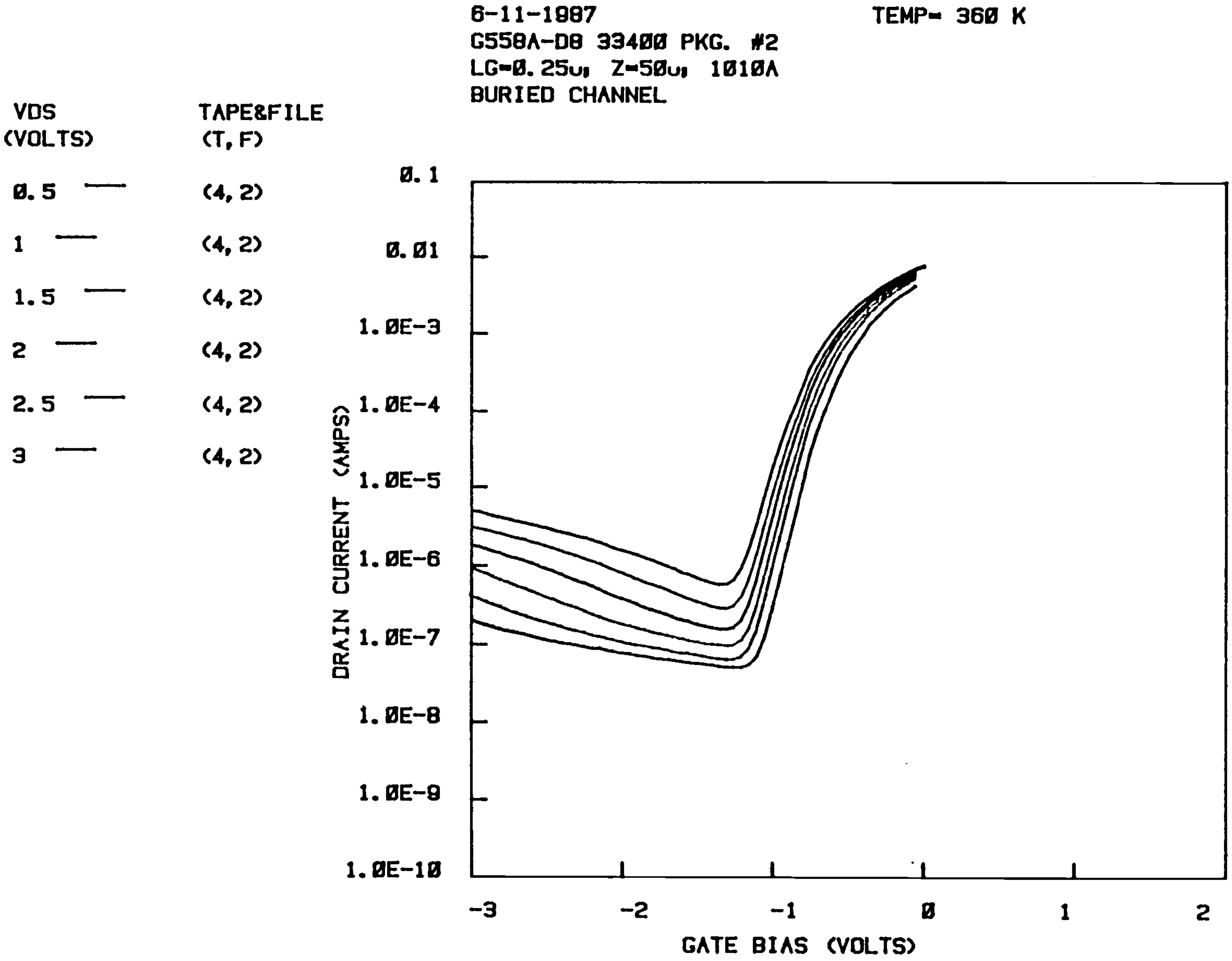
8-14-1987 TEMP= 270 K
G558A-D8 33400 PKG. #2
LG=0.25u, Z=50u, 1010A
BURIED CHANNEL



6-11-1987 TEMP= 300 K
G558A-D8 33400 PKG. #2
LG=0.25u, Z=50u, 1010A
BURIED CHANNEL




8-12-87
G550A-D8 33400 PKG. #2
LG=0.25u Z=50u 1010A
BURIED CHANNEL
TEMP= 310 K
VDS
(VOLTS)
TAPE&FILE
(T, F)
0.5 (4, 3)
1 (4, 3)
1.5 (4, 3)
2 (4, 3)
2.5 (4, 3)
3 (4, 3)
0.1
0.01
1.0E-3
1.0E-4
1.0E-5
1.0E-6
1.0E-7
1.0E-8
1.0E-9
1.0E-10
DRAIN CURRENT (AMPS)
-3
-2
-1
0
1
2
GATE BIAS (VOLTS)


6-11-1987
TEMP= 360 K
G558A-D8 33400 PKG. #2
LG=0.25u, Z=50u, 1010A
BURIED CHANNEL
VDS (VOLTS)
TAPE&FILE (T, F)
0.5 (4, 2)
1 (4, 2)
1.5 (4, 2)
2 (4, 2)
2.5 (4, 2)
3 (4, 2)
DRAIN CURRENT (AMPS)
0.1
0.01
1.0E-3
1.0E-4
1.0E-5
1.0E-6
1.0E-7
1.0E-8
1.0E-9
1.0E-10
-3
-2
-1
0
1
2
GATE BIAS (VOLTS)

7.4 Controller and Plotter Program Listings

```
100 REM *******************5-13-1987 ******** FILE 6 *****************
110 REM *
120 REM *  Author    :  Philip C. Canfield
130 REM *  Update    :  P. K. Or
140 REM *  Date      :  July 15, 1987  (Last Update)
150 REM *  Implement :  BASIC on Tektronix 4054A
160 REM *
170 REM *  Purpose   :  To issue communication protocol to various
180 REM *               measuring equipments, sequence the measurement
190 REM *               procedure and store data in tape.
200 REM *
210 REM ***************************************************************1
220 REM ****************************************************************
    **
230 CALL Start
240 END
250 CALL Ndev
260 END
270 CALL Tempset
280 END
290 CALL Bias
300 END
310 CALL Idsvgs
320 END
330 CALL Vdsswp
340 END
350 CALL Sbthshd
360 END
370 REM ****************************************************************
380 REM ***                                                          ***
390 REM ***  PROGRAM: MEASURES Ids vs Vgs                ***
400 REM ***                                                          ***
410 REM ****************************************************************
420   !
430   !
440 REM
450 REM
460 REM INITIALIZE THE SYSTEM AND IDENTIFY THE DEVICE TO BE MEASURED
470 REM
480 REM
490 REM ****************************************************************
500 REM
510 REM
520 SUB Start
530    PAGE
540    INIT
550    PRINT @26:"init"
560    PRINT @16:"init"
570    PRINT @24:"init"
580    PRINT @20:"IN"
590    PRINT @25:"INit"
600    PRINT @25:"freq 20E6;ampl 0.0;offs 0.0"
610    Rsen=50
620    Ids=" "
```

```
630    Devcom$=" "
640    Com$=" "
650    Dte$=" "
660    Temp=300
670    Lz=300
680    Crntlm$="0.1"
690    J=1
700    Relec=0.235
710    PRINT @26:"clo 9,10"
720    PRINT "Check the resistance used to measure the current by the"
730    PRINT "electrometer when the drain current is out of range."
740    R$=STR(Relec)
750    ALTER R$,R$
760    Relec=VAL(R$)
770    PRINT @26:"ope 9,10"
780    PRINT @37,0:10,255,13
790    PRINT "IF THE ELECTROMETER 'ZERO CHECK' LIGHT IS ON, TURN IT OFF"
800    PRINT "AND PRESS 'RETURN' TO CONTINUE"
810    INPUT C$
820    File=1
830    Tape=1
840 END SUB
850   !
860   !
870   ! ===========================================================
880   !
890   ! RECORD NEW DEVICE IDENTIFICATION
900   !
910   ! =======================================================
920   !
930   !
940 SUB Ndev
950    Thrshld$="N"
960    Ovrrng$="N"
970    PRINT "ENTER DEVICE I.D. NUMBER"
980    ALTER Id$,Id$
990    PRINT "ENTER DEVICE SPECIFICS AND/OR PECULIARITIES"
1000    ALTER Devcom$,Devcom$
1010    Lz$=STR(Lz)
1020    PRINT "Enter the device gate width."
1030    ALTER Lz$,Lz$
1040    Lz=VAL(Lz$)
1050    Rsen$=STR(Rsen)
1060    PRINT "THE VALUE OF THE DRAIN SENSE RESISTOR=?"
1070    ALTER Rsen$,Rsen$
1080    Rsen=VAL(Rsen$)
1090    PRINT "CURRENT LIMIT FOR THE DRAIN CURRENT SOURCE=? (AMPS)"
1100    ALTER Crntlm$,Crntlm$
1110    Cl$="CU "
1120    Cl$=Cl$&Crntlm$
1130    PRINT @23:Cl$
1140    ALTER Dte$,Dte$
1150 END SUB
1160   !
```

```
1170   !
1180   ! ======================================================
1190   !
1200   ! RECORD TEMPERATURE OF THE MEASUREMENT
1210   !
1220   ! ======================================================
1230   !
1240   !
1250 SUB Tempset
1260     Temp$=STR(Temp)
1270     PRINT "Enter temperature of the measurement."
1280     ALTER Temp$,Temp$
1290     Temp=VAL(Temp$)
1300 END SUB
1310   !
1320   !
1330 REM
1340 REM
1350 REM ****************************************************
1360 REM
1370 REM
1380 REM SET VARIOUS BIASES AND DIMENSION ARRAYS
1390 REM
1400 REM
1410 REM ****************************************************
1420 REM
1430 REM
1440 SUB Bias
1450     PRINT "ENTER MOST NEGATIVE GATE BIAS"
1460     INPUT Negvgs
1470     PRINT "ENTER MOST POSITIVE GATE BIAS"
1480     INPUT Plsvgs
1490     PRINT "ENTER GATE BIAS INCREMENT"
1500     INPUT Vgsincr
1510     M=INT(1+(Plsvgs-Negvgs)/Vgsincr)
1520     PRINT "ENTER ADDITIONAL IMPORTANT INFORMATION"
1530     ALTER Com$,Com$
1540     IF Thrshld$="Y" THEN 1560
1550     DIM Gds(M),Vgsgo(M+1),Gm(M),Vgsgm(M+1)
1560 END SUB
1570 REM
1580 REM
1590 REM ****************************************************
1600 REM
1610 REM
1620 REM MEASURE DRAIN CURRENT VS VGS
1630 REM
1640 REM
1650 REM ****************************************************
1660 REM
1670 REM
1680 SUB Idsvgs
1690     Vgs=Plsvgs
1700     Vds4=Vds
```

```
1710    Vgs4=Vgs
1720    PRINT @16:"dcv"
1730    Ovrrng$="N"
1740    Xd$="VO "
1750    Xg$="offs "
1760    Y$=STR(Vds)
1770    Zd$=Xd$&Y$
1780    PRINT @20:Zd$
1790    PRINT @37,26:1
1800    PRINT @6,1:"F1R9T4X"  !put electormeter on the 2A scale.
1810    PRINT @37,26:0
1820    CALL "wait",4
1830    PRINT @20:"OUT ON"  !turn on the drain power supply
1840    Vgg=Vgs
1850    Gps=25
1860    Y$=STR(Vgs)
1870    Zg$=Xg$&Y$
1880    PRINT @Gps:Zg$
1890    PRINT @Gps:"OUT ON"  !turn on the gate power supply
1900    Dps=20
1910    PRINT @26:"clo 14,10"  !measure Vgs DVM; connect the elect.
1920    CALL "WAIT",2
1930    CALL Adjvgs
1940    PRINT @26:"ope 14;clo 15"  !measure Vds with the DVM
1950    CALL "wait",2
1960    CALL Adjvds
1970    PRINT @26:"ope 15;clo 14"  !measure Vgs with the DVM
1980    IF Ovrrng$="Y" THEN 2050
1990    CALL "wait",2
2000    PRINT @16:"SEN"
2010    INPUT @16:Vgsid(K,1)
2020    Vdif=Vgsid(K,1)-Vgs
2030    IF ABS(Vdif)>0.01 THEN 1930
2040    PRINT @26:"ope 14;clo 15"  !measure Vds with the DVM
2050    PRINT "Check the electrometerGGGGGG"
2060    P=1
2070    CALL "wait",4
2080    PRINT @37,25:1
2090    INPUT %5,1:Ids(K,P)
2100    PRINT @37,26:0
2110    IF ABS(Ids(K,P))<0.01 THEN 2130
2120    GO TO 2470
2130    PRINT @37,26:1
2140    PRINT @6,1:"F1R8T4X"  !set the electrometer to the 20mA scale
2150    CALL "wait",2
2160    PRINT @6,1:"F1R0T4X"  !set the electrometer to autoranging
2170    PRINT @37,26:0
2180    Vgs=Vgs-Vgsincr
2190    Y$=STR(Vgs)
2200    Zg$=Xg$&Y$
2210    P1=P
2220    FOR P=P1 TO M
2230       PRINT @26:"ope 15;clo 14"  !measure Vgs with the DVM
2240       CALL "wait",2
```

```
2250        CALL Adjvgs
2260        PRINT @25:"ope 14;clo 15"  !measure Vds with the DVM
2270        CALL "wait",2
2280        CALL Adjvds
2290        PRINT @26:"ope 15;clo 14"  !measure Vgs with the DVM
2300        CALL "wait",2
2310        PRINT @16:"SEN"
2320        INPUT @16:Vgsid(K,P)
2330        Vdif=Vgsid(K,P)-Vgs
2340        IF ABS(Vdif)>0.01 THEN 2250
2350        PRINT @26:"ope 14;clo 15"  !measure Vds with the DVM
2360        CALL "WAIT",2
2370        PRINT @37,26:1
2380        INPUT %6,1:Ids(K,P)
2390        PRINT @37,26:0
2400        Vgs=Vgs-Vgsincr
2410        Y$=STR(Vgs)
2420        Zg$=Xg$&Y$
2430        PRINT @Gps:Zg$
2440        CALL "wait",1
2450     NEXT P
2460     GO TO 2830
2470     PRINT @26:"clo 9"  !connect resistor across inputs of electr.
2480     PRINT @37,26:1
2490     PRINT @6,1:"F0R0T4X"  !SET ELECTROMETER TO MEASURE VOLTAGE
2500     PRINT @37,26:0
2510     PRINT @26:"ope 15;clo 14"  !MEASURE Vgs with the DVM
2520     CALL "Wait",2
2530     CALL Adjvgs
2540     PRINT @26:"ope 14;clo 15"  !measure Vds with the DVM
2550     CALL "Wait",2
2560     CALL Adjvds
2570     PRINT @26:"ope 15;clo 14"  !measure Vds with the DVM
2580     CALL "wait",2
2590     PRINT @16:"SEN"
2600     INPUT @16:Vgsid(K,P)
2610     Vdif=Vgsid(K,P)-Vgs
2620     IF ABS(Vdif)>0.01 THEN 2520
2630     PRINT @26:"ope 14;clo 15"  !measur Vds with the DVM
2640     CALL "WAIT",3
2650     PRINT @37,26:1
2660     INPUT %6,1:V8
2670     PRINT @37,26:0
2680     Ids(K,P)=V8/Relec
2690     P=P+1
2700     Vgs=Vgs-Vgsincr
2710     Y$=STR(Vgs)
2720     Zg$=Xg$&Y$
2730     PRINT @Gps:Zg$
2740     CALL "wait",1
2750     IF P>M THEN 2830
2760     IF ABS(Ids(K,P-1))=>0.01 THEN 2510
2770     P1=P
2780     PRINT @26:"ope 9"
```

```
2790     PRINT @37,26:1
2800     PRINT @6,1:"F1R0T4X"  !set the electrometer to autorange current
2810     PRINT @37,26:0
2820     GO TO 2220
2830     PRINT @Gps:"OUT OFF"
2840     PRINT @20:"OUT OFF"
2850     PRINT @26:"ope all"
2860 END SUB
2870   !
2880   !
2890   ! ==========================================================
2900   !
2910   ! ADJUST VDS TO THE DESIRED VALUE
2920   !
2930   ! ==========================================================
2940   !
2950   !
2960 SUB Adjvds
2970     Vdd=Vds4
2980     PRINT @16:"sen"
2990     INPUT @16:Vds4
3000     Vdif=Vds-Vds4
3010     IF ABS(Vdif)<0.01 THEN 3130
3020     Vdd=Vdd+Vdif
3030     IF Dps<>24 THEN 3050
3040     IF Vdd>7.5 THEN 3120
3050     IF Vdd<0 THEN 2970
3060     IF Vdd>10 THEN 3130
3070     Y$=STR(Vdd)
3080     Zd$=Xd$&Y$
3090     PRINT @Dps:Zd$
3100     CALL "wait",2
3110     GO TO 2980
3120     Ovrrng$="Y"
3130 END SUB
3140   !
3150   !
3160   ! ==========================================================
3170   !
3180   ! ADJUST THE GATE BIAS TO THE DESIRED VALUE
3190   !
3200   ! ==========================================================
3210   !
3220   !
3230 SUB Adjvgs
3240     Vgg=Vgs
3250     PRINT @16:"sen"
3260     INPUT @16:Vgs4
3270     Vdif=Vgs-Vgs4
3280     IF ABS(Vdif)<0.01 THEN 3380
3290     Vgg=Vgg+Vdif
3300     IF Vgg<-7.5 THEN
3310        Vgg=Vgs
3320     END IF
```

```
3330     Y$=STR(Vgg)
3340     Zg$=Xg$&Y$
3350     PRINT @Gps:Zg$
3360     CALL "wait",2
3370     GO TO 3250
3380 END SUB
3390   !
3400   !
3410   ! ===================================================
3420   !
3430   ! SUBROUTINE FOR MEASUREING SUBTHRESHOLD CURRENTS
3440   !
3450   ! ===================================================
3460   !
3470   !
3480 SUB Sbthshd
3490     PRINT @26:"OPE ALL"
3500     Thrshld$="Y"
3510     PRINT "Enter Vds max."
3520     INPUT Vdsmax
3530     PRINT "Enter Vds min."
3540     INPUT Vdsmin
3550     PRINT "Enter the step size for Vds."
3560     INPUT Vdsstp
3570     J=INT((Vdsmax-Vdsmin)/Vdsstp+1.001)
3580     DIM Sbvds(J+1)
3590     PRINT "Enter the data tape number and be sure that"
3600     PRINT "it is in the computer.GGG"
3610     INPUT Tape
3620     PRINT "Enter the starting data file #."
3630     INPUT File
3640     CALL Bias
3650     CALL Tempset
3660     Sbvds(1)=Vdsmin
3670     DIM Ids(J,M),Vgsid(J,M)
3680     FOR K=1 TO J
3690        Vds=Sbvds(K)
3700        CALL Idsvgs
3710        IF K=>J THEN 3730
3720        Sbvds(K+1)=Vdsmin+Vdsstp*K
3730     NEXT K
3740     CALL Sbthstr
3750     J=1
3760     Thrshld$="N"
3770 END SUB
3780   !
3790   !
3800   ! ===================================================
3810   !
3820   ! STORE THE SUBTHRESHOLD CURRENT DATA
3830   !
3840   ! ===================================================
3850   !
3860   !
```

```
3870 SUB Sbthstr
3880    FIND File
3890    WRITE @33:Id$,Devcom$,Dte$,Com$
3900    WRITE @33:Tape,File,J,M,Temp,Negvgs,Plsvgs,Vdsmax,Vdsmin
3910    FOR Ks=1 TO J
3920       WRITE @33:Sbvds(Ks)
3930       FOR Ps=1 TO M
3940          WRITE @33:Ids(Ks,Ps),Vgsid(Ks,Ps)
3950       NEXT Ps
3960    NEXT Ks
3970 END SUB
```

```
100   !**************6-29-1987**********FILE 7************
110 REM *
120 REM *  Author    :  P. K. Or,  Philip C. Canfield
130 REM *  Update    :  P. K. Or
140 REM *  Date      :  July 30, 1987  (Last update)
150 REM *  Implement:  BASIC on Tektronix 4054A
160 REM *  Plotter   :  HP 7225A
170 REM *
180 REM *  Purpose   :  To plot data stored on magnetic tapes. Four
190 REM *               forms of data : Ids vs Vgs, Igs vs Vgs, Iss vs
200 REM *               Vgs and Isb vs Vgs for different temperature
210 REM *               ranges. The data are presented in linear or
220 REM *               log current plot with normalized or non-
230 REM *               normalized Y-axis. These choices are presented in
240 REM *               the main menu.
250 REM *
260 REM *
270   !**************BACKUP FILE 6************************
280 CALL Main
290   !
300   !
310   !=================================================
320   !
330   ! SUBROUTINE TO DRAW AND LABLE THE AXIS
340   !
350   !=================================================
360   !
370   !
380 SUB Pltaxis
390    Prt=5
400    PRINT @37,26:1
410    PRINT @Prt:"IN"
420    H$=CHR(3)
430    PRINT @Prt:"PA 9800,6500;PD;"
440    PRINT @Prt:"PA 9800,900;"
450    PRINT @Prt:"PA 3800,900;"
460    PRINT @Prt:"PA 3800,6500;"
470    PRINT @Prt:"PA 9800,6500;PU;"
480    PRINT @Prt:"PA 3800,6500;"
490    PRINT @Prt:"TL 2.0;"
500    PRINT @Prt:"IP 3800,900,9800,6500;"
510    PRINT @Prt:"SC";Xlo;",";Xhi;",";Ylo;",";Yhi;";"
520    FOR I=1 TO Nydiv-1
530       PRINT @Prt:"PR 0,";-Ty;";YT;"
540    NEXT I
550    PRINT @Prt:"PA";Xlo;",";Ylo;";"
560    FOR I=1 TO Nxdiv-1
570       PRINT @Prt:"PR";Tx;",0;XT;"
580    NEXT I
590    PRINT @Prt:"PA";Xlo;",";Ylo;";"
600    PRINT @Prt:"SI;"
610    Xdvlb$=STR(Xdivlb(1))
620    PRINT @Prt:"CP -1,-1.5;LB";Xdvlb$;H$;";"
630    FOR I=2 TO Nxdiv+1
```

```
640         Xdvlb$=STR(Xdivlb(I))
650         PRINT @Prt:"PR";Tx;",0;CP-3,0;LB";Xdvlb$;H$;";"
660      NEXT I
670      PRINT @Prt:"PA";Xlo;",";Ylo;";CP -2,0;"
680      FOR I=1 TO Nydiv+1
690         Ydvlb$=STR(Ydivlb(I))
700         PRINT @Prt:"CP";-LEN(Ydvlb$);",0;"
710         PRINT @Prt:"LB";Ydvlb$;H$;";PR 0,";Ty;";"
720      NEXT I
730      PRINT @Prt:"IP;SC;"
740      PRINT @Prt:"PA 5500,300;"
750      PRINT @Prt:"LB";Xlable$;H$;";"
760      IF Nrmal$="Y" THEN
770         PRINT @Prt:"PA 2700,1700;"
780      ELSE
790         PRINT @Prt:"PA 2700,2500;"
800      END IF
810      PRINT @Prt:"DI 0,1;"
820      PRINT @Prt:"LB";Ylable$;H$;";"
830      PRINT @Prt:"DI 1,0;"
840      PRINT @Prt:"PA 3800,7800;"
850      PRINT @Prt:"LB";Header1$;H$;";CP;"
860      PRINT @Prt:"LB";Header2$;H$;";CP;"
870      PRINT @Prt:"LB";Header3$;H$;";CP;"
880      PRINT @Prt:"LB";Header4$;H$;";"
890      PRINT @Prt:"PA 7000,7800;"
900      PRINT @Prt:"LB";Var1$;H$;";CP;"
910      PRINT @Prt:"LB";Var2$;H$;";"
920      PRINT @Prt:"PA 50,7000;LB";Var3$;H$;";"
930      PRINT @Prt:"PA 1500,7000;LB TAPE&FILE";H$;";"
940      PRINT @Prt:"PA 50,7000;CP;"
950      PRINT @Prt:"OA;"
960      INPUT @Prt:Lx,Ly,St
970      PRINT @Prt:"LB";Unt$;H$;";"
980      PRINT @Prt:"PA 1500,";Ly;";"
990      PRINT @Prt:"LB (T,F)";H$;";"
1000     PRINT @Prt:"PA 50,";Ly;";CP 0,-2;OA;"
1010     INPUT @Prt:Lx,Ly,St
1020     PRINT @37,26:0
1030  END SUB
1040     !
1050     !
1060     !==========================================
1070     !
1080     ! SUBROUTINE TO PLOT THE DATA AT THE PLOTTER
1090     !
1100     !==========================================
1110     !
1120     !
1130  SUB Pltdta
1140     PRINT @37,26:1
1150     PRINT @Prt:"IP 3800,900,9800,6500;"
1160     PRINT @Prt:"IW 3800,900,9800,6500;"
1170     PRINT @Prt:"SC";Xlo;",";Xhi;",";Ylo;",";Yhi;";"
```

```
1180     PRINT @Prt:"PA";X(1);",";Y(1);";PD;"
1190     FOR I=1 TO Numpts
1200        PRINT @Prt:"PA";X(I);",";Y(I);";"
1210     NEXT I
1220     PRINT @Prt:"IP;IW;SC;"
1230     PRINT @Prt:"PU;PA";Lx;",";Ly;";"
1240     PRINT @Prt:"LB";Crntdta$;H$;";"
1250     PRINT @Prt:"CP 2,0.5;PD;PR 300,0;PU;"
1260     PRINT @Prt:"PA 1500,";Ly;";LB (";Tape;",";File;")";H$;";"
1270     PRINT @Prt:"PA";Lx;",";Ly;";CP;CP;OA;"
1280     INPUT @Prt:Lx,Ly,St
1290     PRINT @37,26:0
1300 END SUB
1310   !
1320   !
1330   !==================================================
1340   !
1350   !SUBROUTIND TO READ THE SUBTHRESHOLD CURRENT DATA
1360   !
1370   !==================================================
1380   !
1390   !
1400 SUB Ridsdta
1410    PRINT "Check to be sure that data tape #";Tape;" is instaled"
1420    INPUT C$
1430    FIND File
1440    READ @33:Id$,Devcom$,Dte$,Com$
1450    READ @33:Tape,File,J,M,Temp,Negvgs,Plsvgs,Vdsmax,Vdsmin,Lg,Lz
1460    DIM Sbvds(J),Ids(J,M),Vgsid(J,M),Igs(J,M),Iss(J,M),Isb(J,M)
1470    FOR Ks=1 TO J
1480       READ @33:Sbvds(Ks)
1490       FOR Ps=1 TO M
1500          READ @33:Ids(Ks,Ps),Vgsid(Ks,Ps),Igs(Ks,Ps),Iss(Ks,Ps),Isb
          (Ks,Ps)
1510       NEXT Ps
1520    NEXT Ks
1530 END SUB
1540   !
1550   !
1560   !=======================================================
1570   !
1580   ! SUBROUTINE TO PUT THE SUBTHRESHOLD DATA IN THE PROPER
1590   ! FORMAT FOR PLOTTING AND THEN TO PLOT IT.
1600   !
1610   !=======================================================
1620   !
1630   !
1640 SUB Indcrnt
1650    Pass$="N"
1660    PAGE
1670    PRINT "CHOOSE CURRENT VERSUS Vgs PLOT WANTED::"
1680    PRINT
1690    PRINT "1.....DRAIN CURRENT"
1700    PRINT "2.....GATE CURRENT"
```

```
1710    PRINT "3.....SOURCE CURRENT"
1720    PRINT "4.....SUBSTRATE CURRENT"
1730    PRINT "5.....RETURN TO MAIN MENU"
1740    INPUT Ipt
1750    IF Pass$="Y" THEN 1830
1760    PRINT "SELECT THE DATA TAPE AND FILE NUMBER TO BE PLOTTED."
1770    PRINT "TAPE?    ";
1780    INPUT Tape
1790    PRINT "FILE?   ";
1800    INPUT File
1810    CALL Ridsdta
1820    DIM Sbipt(J,M)
1830    IF Ipt=1 THEN
1840       Crnt$="DRAIN"
1850       FOR Ks=1 TO J
1860          FOR Ps=1 TO M
1870             Sbipt(Ks,Ps)=Ids(Ks,Ps)
1880          NEXT Ps
1890       NEXT Ks
1900    ELSE
1910       IF Ipt=2 THEN
1920          Crnt$="GATE"
1930          FOR Ks=1 TO J
1940             FOR Ps=1 TO M
1950                Sbipt(Ks,Ps)=Igs(Ks,Ps)
1960             NEXT Ps
1970          NEXT Ks
1980       ELSE
1990          IF Ipt=3 THEN
2000             Crnt$="SOURCE"
2010             FOR Ks=1 TO J
2020                FOR Ps=1 TO M
2030                   Sbipt(Ks,Ps)=Iss(Ks,Ps)
2040                NEXT Ps
2050             NEXT Ks
2060          ELSE
2070             IF Ipt=4 THEN
2080                Crnt$="SUBSTRATE"
2090                FOR Ks=1 TO J
2100                   FOR Ps=1 TO M
2110                      Sbipt(Ks,Ps)=Isb(Ks,Ps)
2120                   NEXT Ps
2130                NEXT Ks
2140             ELSE
2150                IF Ipt=5 THEN 2410
2160                PRINT "INVALID SELECTION. TRY AGAIN";
2170                GO TO 1740
2180             END IF
2190          END IF
2200       END IF
2210    END IF
2220    Scle$="O"
2230    IF Ipt=1 THEN
2240       PRINT "PLOT DRAIN CURRENT ON LOG(O) OR LINEAR(I) SCALE?   ";
```

```
2250         ALTER Scle$,Scle$
2260      ELSE
2270         PRINT "CURRENT IS PLOTTED ON LOG SCALE"
2280      END IF
2290      PRINT "CHOOSE TEMPERATURE(T) OR Vds(V).   ";
2300      Choice$="V"
2310      ALTER Choice$,Choice$
2320      PRINT "NORMALIZE PLOT WITH GATE WIDTH (Y/N)?  ";
2330      Nrmal$="N"
2340      ALTER Nrmal$,Nrmal$
2350      CALL Label
2360      CALL Xyaxis
2370      CALL Pltaxis
2380      CALL Frmdta
2390      Pass$="Y"
2400      GO TO 1660
2410 END SUB
2420   !
2430 SUB Label
2440     Header1$=Dte$
2450     Header2$=Id$
2460     Header3$=Devcom$
2470     Header4$=Com$
2480     Var2$=" "
2490     IF Ipt=1 THEN
2500        IF Scle$="O" THEN
2510           IF Nrmal$="Y" THEN
2520              Ylable$="DRAIN CURRENT (AMPS) / GATE WIDTH (uM)"
2530           ELSE
2540              Ylable$="DRAIN CURRENT (AMPS)"
2550           END IF
2560        ELSE
2570           IF Nrmal$="Y" THEN
2580              Ylable$="DRAIN CURRENT (MILLIAMPS) / GATE WIDTH (uM)"
2590           ELSE
2600              Ylable$="DRAIN CURRENT (MILLIAMPS)"
2610           END IF
2620        END IF
2630     ELSE
2640        IF Nrmal$="Y" THEN
2650           Ylable$=Crnt$&" CURRENT (AMPS) / GATE WIDTH (uM)"
2660        ELSE
2670           Ylable$=Crnt$&" CURRENT (AMPS)"
2680        END IF
2690     END IF
2700     Xlable$="GATE BIAS (VOLTS)"
2710     IF Choice$="T" THEN
2720        Var1$="VDS="
2730        FOR K=1 TO J
2740           PRINT K;"........VDS(";K;")=";Sbvds(K)
2750        NEXT K
2760        PRINT "CHOOSE THE VDS TO PLOT BY #"
2770        INPUT K
2780        IF K<1 THEN
```

```
2790          PRINT "INVALID CHOICE TRY AGAIN"
2800          GO TO 2770
2810       END IF
2820       IF K>J THEN
2830          PRINT "INVALID CHOICE TRY AGAIN"
2840          GO TO 2770
2850       END IF
2860       Vds$=STR(Sbvds(K))
2870       Vdsunt$=" VOLTS"
2880       Var1$=Var1$&Vds$&Vdsunt$
2890       Var3$="TEMP"
2900       Unt$="(K)"
2910    ELSE
2920       Var1$="TEMP="
2930       Temp$=STR(Temp)
2940       Tunt$=" K"
2950       Var1$=Var1$&Temp$&Tunt$
2960       Var3$=" VDS"
2970       Unt$="(VOLTS)"
2980    END IF
2990 END SUB
3000   !
3010   !
3020 SUB Xyaxis
3030      !
3040      ! X-AXIS DIVISIONS
3050      !
3060   Xlo=-4
3070   Xhi=1
3080   Nxdiv=Xhi-Xlo
3090   DIM Xdivlb(Nxdiv+1)
3100   Xdivlb(1)=Xlo
3110   FOR Ix=2 TO Nxdiv+1
3120      Xdivlb(Ix)=INT(Xdivlb(Ix-1)+1.001)
3130   NEXT Ix
3140   Xlo=INT(100*Xlo+1.0E-3)
3150   Xhi=INT(100*Xhi+1.0E-3)
3160   Tx=100
3170     !
3180     ! Y-AXIS DIVISIONS
3190     !
3200   IF Scle$="I" THEN
3210      Ylo=0
3220      IF Ids(J,1)<0.025 THEN
3230         Yhi=25
3240         Nydiv=5
3250      ELSE
3260         IF Ids(J,1)<0.05 THEN
3270            Yhi=50
3280            Nydiv=5
3290         ELSE
3300            IF Ids(J,1)<0.1 THEN
3310               Yhi=100
3320               Nydiv=5
```

```
3330             ELSE
3340                Yhi=150
3350                Nydiv=6
3360             END IF
3370          END IF
3380       END IF
3390       Ty=Yhi/Nydiv
3400       DIM Ydivlb(Nydiv+1)
3410       Ydivlb(1)=0
3420       FOR Iy=2 TO Nydiv+1
3430          Ydivlb(Iy)=INT(Ydivlb(Iy-1)+Ty+1.0E-3)
3440       NEXT Iy
3450       Yhi=INT(Yhi*10+1.0E-3)
3460       Ty=INT(Ty*10+1.0E-3)
3470    ELSE
3480       IF Crnt$="DRAIN" OR Crnt$="SOURCE" THEN
3490          Ylo=-10
3500          Yhi=-1
3510       ELSE
3520          Ylo=-13
3530          Yhi=-4
3540       END IF
3550       Nydiv=Yhi-Ylo
3560       DIM Ydivlb(Nydiv+1)
3570       Ydivlb(1)=10^Ylo
3580       FOR Iy=2 TO Nydiv+1
3590          Ydivlb(Iy)=10*Ydivlb(Iy-1)
3600       NEXT Iy
3610       Ty=100
3620       Ylo=INT(Ylo*100+1.0E-3)
3630       Yhi=INT(Yhi*100+1.0E-3)
3640    END IF
3650 END SUB
3660   !
3670   !
3680 SUB Frmdta
3690    DIM X(M),Y(M)
3700      !
3710      ! FORMAT DATA
3720      !
3730    IF Choice$="T" THEN
3740       Crntdta$=STR(Temp)
3750       Numpts=M
3760       FOR P=1 TO M
3770          X(P)=INT(100*Vgsid(K,P)+1.0E-3)
3780       NEXT P
3790       IF Scle$="I" THEN
3800          IF Nrmal$="Y" THEN
3810             FOR P=1 TO M
3820                Y(P)=INT(10000*Sbipt(K,P)/Lz+1.0E-3)
3830             NEXT P
3840          ELSE
3850             FOR P=1 TO M
3860                Y(P)=INT(10000*Sbipt(K,P)+1.0E-3)
```

```
4410            IF Nrmal$="Y" THEN
4420               FOR P=1 TO M
4430                  Y(P)=INT(10000*Sbipt(Vdv(K),P)/Lz+1.0E-3)
4440               NEXT P
4450            ELSE
4460               FOR P=1 TO M
4470                  CALL Pdata
4480               NEXT P
4490            END IF
4500         ELSE
4510            IF Nrmal$="Y" THEN
4520               FOR P=1 TO M
4530                  Y(P)=INT(LGT(ABS(Sbipt(Vdv(K),P)/Lz))*100+1.0E-33
                      )
4540               NEXT P
4550            ELSE
4560               FOR P=1 TO M
4570                  Y(P)=INT(LGT(ABS(Sbipt(Vdv(K),P)))*100+1.0E-3)
4580               NEXT P
4590            END IF
4600         END IF
4610         PRINT "INSERT A DIFFERENT COLORED PEN IN THE PLOTTER"
4620         PRINT "AND PRESS RETURN."
4630         INPUT C$
4640         CALL Pltdta
4650      NEXT K
4660   END IF
4670 END SUB
4680   !
4690   !
4700 SUB Pdata
4710   PAGE
4720   Pnt=1
4730   PRINT "SELECT DATA TAPE & FILE TO BE PRINTED"
4740   PRINT "TAPE=  ";
4750   INPUT Tape
4760   PRINT "FILE=  ";
4770   INPUT File
4780   CALL Ridsdata
4790   PRINT @37,26:1
4800   PRINT @Pnt:Id$
4810   PRINT @Pnt:Devcom$
4820   PRINT @Pnt:Com$
4830   PRINT @Pnt:"TAPE: ",Tape,"      FILE:",File
4840   PRINT @Pnt:"TEMP=",Temp
4850   PRINT @Pnt:"LG= ",Lg,"        LZ= ",Lz
4860   FOR Ks=1 TO J
4870      PRINT @Pnt:"Vds=",Sbvds(Ks)
4880      FOR Ps=1 TO M
4890         PRINT @Pnt:"Vgs=",Vgsid(Ks,Ps),"    Ids=",Ids(Ks,Ps)
4900      NEXT Ps
4910   NEXT Ks
4920   PRINT @37,26:0
4930 END SUB
```

```
3870           NEXT P
3880        END IF
3890     ELSE
3900        IF Nrmal$="Y" THEN
3910           FOR P=1 TO M
3920              Y(P)=INT(LGT(Sbipt(K,P)/Lz)*100+1.0E-3)
3930           NEXT P
3940        ELSE
3950           FOR P=1 TO M
3960           END IF
3970           PRINT "CHOOSE A DIFFERENT COLORED PEN FOR THE"
3980           PRINT "PLOTTER AND PRESS RETURN ."
3990           INPUT C$
4000           CALL Pltdta
4010           PRINT "PLOT ANOTHER TEMPERATURE? (Y/N)"
4020           Y(P)=INT(LGT(Sbipt(K,P))*100+1.0E-3)
4030        NEXT P
4040     END IF
4050     INPUT Anthrt$
4060     IF Anthrt$="N" THEN 4670
4070     PRINT "DATA TAPE #?"
4080     INPUT Tape
4090     PRINT "FILE #?"
4100     INPUT File
4110     CALL Ridsdta
4120     GO TO 3690
4130  ELSE
4140     PAGE
4150     PRINT "DO YOU WISH TO PLOT THE DATA FOR ALL OF THE"
4160     PRINT "VDS VALUES LISTED BELOW? (Y/N)"
4170     Slvdvlu$="Y"
4180     DIM Vdv(J+1)
4190     FOR Ki=1 TO J
4200        PRINT Ki;"......VDS(";Ki;")=";Sbvds(Ki)
4210        Vdv(Ki)=Ki
4220     NEXT Ki
4230     J1=J
4240     ALTER Slvdvlu$,Slvdvlu$
4250     IF Slvdvlu$="Y" THEN 4340
4260     PRINT "SELECT THE #'S CORRESPONDING TO THE VDS VALUES"
4270     PRINT "YOU WANT TO PLOT ONE AT A TIME FOLLOWED BY A <CR>."
4280     PRINT "ENTER A 0 TO QUIT"
4290     Kc=0
4300     Kc=Kc+1
4310     INPUT Vdv(Kc)
4320     IF Vdv(Kc)<>0 THEN 4300
4330     J1=Kc-1
4340     Numpts=M
4350     FOR K=1 TO J1
4360        Crntdta$=STR(Sbvds(Vdv(K)))
4370        FOR P=1 TO M
4380           X(P)=INT(100*Vgsid(Vdv(K),P)+1.0E-3)
4390        NEXT P
4400        IF Scle$="I" THEN
```

```
4940   !
4950   !
4960   !
4970 SUB Main
4980    Tape$="1"
4990    File$="1"
5000    PAGE
5010    CHARSIZE 3
5020    PRINT "******************************************************"
5030    PRINT
5040    PRINT
5050    PRINT "                  MAIN MENU "
5060    PRINT
5070    PRINT
5080    PRINT "******************************************************"
5090    PRINT
5100    PRINT
5110    PRINT "1.....INDIVIDUAL COMPONENT OF CURRENT"
5120    PRINT "2.....PRINT DATA FROM DATA TAPE"
5130    PRINT "3.....QUIT"
5140    INPUT Slct
5150    IF Slct=1 THEN
5160       CALL Indcrnt
5170    ELSE
5180       IF Slct=2 THEN
5190          CALL Pdata
5200       ELSE
5210          IF Slct=3 THEN 5270
5220          PRINT "INVALID CHOICE. CHOOSE AGAIN.";
5230          GO TO 5140
5240       END IF
5250    END IF
5260    GO TO 5000
5270 END SUB
```

7.5 Curve Fitting Program Listing

```
(************************************************************
*****************)
(*****
*****)
(*****   FILE       : SUBTRESHOLD_CURRENT.PAS
*****)
(*****   VERSION    : 1.02
*****)
(*****   AURTHOR    : P. K. OR
*****)
(*****   PURPOSE    : SUBTHRESHOLD VOLTAGE vs. Vgs & Vds
*****)
(*****   IMPLEMENT : TURBO_PAS
*****)
(*****
*****)
(************************************************************
*****************)


(:::::::::::::::::::::::::::::  EDITING HISTORY
::::::::::::::::::::::::::::::::)
(
)
( 1.02   Create file handling procedures
09-17-1987  )
( 1.01   Working program for Nc constant
09-16-1987  )
( 1.00   Create Procedures and Function necessary for math
09-14-1987  )
(
)
(:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
::::::::::::::::::)

($F20)

Program Subthres ( input, output);

Const
   MaxR = 40;      ( data Points )
   MaxC = 4;       ( polynomial terms )

Const
   Es    = 1.15989E-12;       (Farad/cm - GaAs dielectric
constant)
   Q     = 1.60218E-19;       (Coulomb  - Electronic charge)
   Nc    = 4.022E17;          (cmΔ-3    - Uniform doping)
   t     = 6.00E-6;           (cm       - Active layer
thickness)
   Uc    = 3500;              (cm/V.s   - Channel mobility)
   K     = 1.38066E-23;       (J/K      - Boltzmann constant)
```

```
Type
   Index    = 1..MaxR;
   Ary1R    = Array[Index] of Real;
   Ary1C    = Array[1..MaxC] of Real;

Var
   (**************    Variables used for experimental data
****************)

   Vgs                     : Ary1R;
   Vds                     : Ary1R;
   Isub                    : Ary1R;
   Vto                     : Real;
   Vthv                    : Real;
   Temp                    : Real;
   Lg, Lz                  : Real;
   DeviceID                : String[80];
   Gamma                   : Real;
   Alpha                   : Real;
   Ns                      : Real;
   Data                    : Text;
   DataFile                : String[6];
   StoDataFile             : String[10];
   Vpts,DataPts            : Integer;
   Vgstep                  : Real;

   (**************     Variables used in calculation
********************)

   X, Y                    : Ary1R;
   Y_Calc                  : Ary1R;
   Coef                    : Ary1C;   (solution vector)
   Ex                      : Ary1R;
   Process                 : Text;
   ProcessFile             : String[6];
   StoProcessFile          : String[12];
   Sequence                : Integer;
   SequenceNum             : String[2];
   DDrive                  : String[2];
   Select                  : Integer;
   Vdsub, VthvP            : Real;
   Nrow, Ncol              : Integer;
   A, B                    : Real;
   SRS, ASRS               : Real;
   Done, Error             : Boolean;
   I                       : Integer;
   Store                   : Char;

(ΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔ
ΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔ)
(
)
(                        PROCEDURE GET_DATA
```

```
)
(
)
(    Read Vgs and Isub values from data file.
)
(    X = (Vgs - Vto - Gamma*Vds)    : independent variable
)
(    Y = (q/KT) * (1 - Alpha*Vds)   : dependent variable
)
(
)
(____________________________________________________________
_________________)

Procedure Get_Data ( Var X, Y : Ary1R;
                     Nrow : Integer;
                     Vdsub : Real);

Var
  I : Integer;

Begin
   For I := 1 to Nrow Do
   Begin
      Readln (Data, Isub[I], Vgs[I]);
      X[I] := (Vgs[I] - Vto)*(1 - Alpha*Vdsub)/Ns;
      Y[I] := Isub[I];
   End;
End;


(ΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔ
ΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔ)
(
)
(                          PROCEDURE SET_Vthv
)
(
)
(____________________________________________________________
_________________)

Procedure Set_Vthv (Var Vthv : Real;
                        Temp : Real);

Begin
   Vthv := (K/Q) * Temp;
End;


(ΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔ
ΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔ)
(
```

```
(                         PROCEDURE PRINT_DATA
)
(
)
(   Display Isub values for each set of Vds.
)
(   SRS  :  Sum of Residual square.
)
(   A = Coef[1] : Idso, drain current at low bias
)
(
)
(____________________________________________________________
_______________)

Procedure Print_Data (Vdsub : Real; Nrow : Integer);

Var
   I : Integer;

Begin
   Writeln;
   Writeln ('Vds = ',Vdsub:4:3);
   Writeln ('  I    Vgs   Isub      ','     Isub Calc');
   For I := 1 To Nrow Do
      Writeln (I:3, Vgs[I]:8:2,'  ',Isub[I]:12,'
',Y_Calc[I]:12);
   Writeln;
   Writeln ('Coefficients');
   Writeln (Coef[1]:12,'     Constant Term');
   For I := 2 To Ncol Do
      Writeln (Coef[I]:12);    (other terms)
   Writeln;
   Writeln (' Idso     = ', A:12,' Amps');
   Writeln (' 1/B      = ', 1/B:6:5,' Volt',' Vthv =
',Vthv:6:5);
   Writeln;
   VthvP := ABS(Vthv*B-1);
   Writeln (' SRS      = ',SRS:8:4);
   Writeln (' VthvP    = ',VthvP:8:4);
   Writeln;
End;


(ΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔ
ΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔ)
(
)
(                       PROCEDURE STORE_DATA
)
(
(
)
(    Store calculate data into process file.
)
```

```
(____________________________________________________________
_________________)

Procedure Store_Data( Vdsub          : Real;
                      Nrow           : Integer);
Var
   I  :  Integer;

Begin
   StoProcessFile := ProcessFile + SequenceNum + '.Pro';
   Assign(Process, StoProcessFile);
   Rewrite(Process);
   Writeln(Process, 'DeviceID = ',DeviceID);
   Writeln(Process, 'Lg = ',Lg:4:3, '  Lz = ',Lz:4:3, '
Temp = ',Temp:4:3, '  Vto = ',Vto:4:3);
   Writeln(Process, 'Vthv = ',Vthv:7:6,'  Alpha =
',Alpha:6:5, '  Ns = ',Ns:4:3);
   Writeln(Process);
   Writeln (Process, 'Idso = ',Coef[1]:12,'  1/B =
',1/B:7:6,'  SRS = ',SRS:5:4, '  VthvP = ',VthvP:5:4);
   Writeln (Process,'Vds = ',Vdsub:4:3);
   Writeln (Process,'Vgs     Isub          Isub Calc  ');
   For I := 1 to Nrow Do
    Begin
      Writeln (Process, Vgs[I]:5:3,'  ',Y[I]:12,'
',Y_Calc[I]:12);
    End;
   Flush (Process);
   Sequence := Sequence + 1;
   Str(Sequence, SequenceNum);
End;


(ΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔ
ΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔ)
(
)
(                          PROCEDURE DISKSTORAGE
)
(
)
(   Create data file from experimental data, and store file
in DATA        )
(____________________________________________________________
_________________)


Procedure DiskStorage;
Var
   I, J : Integer;

Begin
   Write ('Device ID          = ');        Readln
(DeviceID);
```

```
   Write ('Gate Length(um)      = ');          Readln (Lg);
   Write ('Gate Width(um)       = ');          Readln (Lz);
   Write ('Temperature(K)       = ');          Readln
(Temp);
   Write ('Threshold Voltage(V) = ');          Readln (Vto);
   Write ('# of Vds Points      = ');          Readln
(Vpts);
   StoDataFile := DataFile + '.Dat';
   Assign (Data, StoDataFile);
   Rewrite (Data);
   Writeln (Data, DeviceID);
   Writeln (Data, Lg:8:3, Lz:8:3, Temp:8:3, Vto:8:3,
Vpts:3);
   For I:= 1 to Vpts Do
   Begin
      Writeln;
      Write ('Vds                = ');          Readln
(Vds[I]);
      Write ('# of Data points = ');            Readln
(DataPts);
      Write ('Starting Vgs       = ');          Readln
(Vgs[1]);
      Write ('Vgs step           = ');          Readln
(Vgstep);
      Writeln (Data, Vds[I]:4:3, DataPts:3);
      Writeln;
      Writeln ('Enter Isub <CR> ');
      For J := 1 to DataPts Do
      Begin
         Readln (Isub[J]);
         Writeln (Data, Isub[J]:12, Vgs[J]:8:3);
         Vgs[J+1]:= Vgs[J] - Vgstep;
      End;
   End;
   Close (Data);
End;


(ΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔ
ΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔ)
(
)
(                             PROCEDURE FUNC
)
(
)
(____________________________________________________________
_________________)
_________________

Procedure Func( B: Real;
                Var Fb, Dfb: Real);

Var
```

```
   I: Integer;
   S1, S2, S3, S4, S5, S6,
   Ex1, EX2, Xi, X2, Yi, Y2 : Real;

Begin
   S1 := 0.0;
   S2 := 0.0;
   S3 := 0.0;
   S4 := 0.0;
   S5 := 0.0;
   S6 := 0.0;
   For I := 1 to DataPts Do
   Begin
      Xi := X[I];
      X2 :=Xi*Xi;
      Yi := Y[I];
      Y2 := Yi*Yi;
      Ex1 := Exp(B*Xi);
      Ex[I] := Ex1;
      Ex2 := Ex1*Ex1;
      S1 := S1 + Xi*Ex2/Y2;
      S2 := S2 + Ex1/Yi;
      S3 := S3 + Xi*Ex1/Yi;
      S4 := S4 + Ex2/Y2;
      S5 := S5 + 2.0*X2*Ex2/Y2;
      S6 := S6 + X2*Ex1/Yi;
   End;
   Fb := S1*S2 - S3*S4;
   Dfb := S2*S5 - S1*S3 - S4*S6;
   A := S2/S4;
End;   (Func)

(ΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔ
ΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔ)
(
)
(                            PROCEDURE NEWTON
)
(___________________________________________________________
_________________)

Procedure Newton(Var X : Real);
Const
   Tol = 1.0E-6;
   Max = 20;

Var
   I : Integer;
   Fx, Dfx, Dx, X1 : Real;

Begin
   Error := False;
   I := 0;
   Repeat
```

```
      I := I + 1;
      X1 := X;
      Func(X, Fx, Dfx);
      If Dfx = 0.0 then
       Begin
         Error := True;
         X := 1.0;
         Writeln ('ERROR: slope zero');
       End
      Else
       Begin
         Dx := Fx/Dfx;
         X := X1 - Dx;
       End;
   Until Error OR (I > Max) OR (Abs(Dx) <= Abs(Tol*X));
   If I > Max Then
    Begin
       Writeln ('ERROR: no convergence in ',max,' loops');
       Error := True;
    End;
End;   (NEWTON)


(ΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔ
ΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔ)
(
)
(                            PROCEDURE NLIN
)
(____________________________________________________________
_________________)

Procedure Nlin(X, Y : Ary1R;
               Var Y_Calc : Ary1R;
                   Nrow : Integer);
  (fit the subthreshold equation through N stes of X and Y
pairs of points)

Var
   Resid : Ary1R;
   I : Integer;
   Xi, Yi, Sum_X, Sum_Y, Sum_Xy, Sum_X2 : Real;

Begin
   Ncol := 2;   (two terms)
   Sum_X := 0.0;
   Sum_Y := 0.0;
   Sum_Xy := 0.0;
   Sum_X2 := 0.0;
   For I := 1 To Nrow Do
    Begin
       Xi := X[I];
       Yi := Ln(Y[I]);
       Sum_X := Sum_X + Xi;
```

```
        Sum_Y := Sum_Y + Yi;
        Sum_Xy := Sum_Xy + Xi*Yi;
        Sum_X2 := Sum_X2 + Xi*Xi;
     End;
    B := (Sum_Xy - Sum_X*Sum_Y/Nrow) /
         (Sum_X2 - Sqr(Sum_X)/Nrow);
    Newton(B);
    Coef[1] := A;
    Coef[2] := B;
    SRS := 0.0;
    For I := 1 To Nrow do
     Begin
        Y_Calc[I] := A*Ex[I];
        IF Y[I] <> 0.0 Then
           Resid[I] := Y_Calc[I]/Y[I] - 1.0
        Else
           Resid[I] := Y[I]/Y_Calc[I] - 1.0;
        SRS := SRS + Sqr(Resid[I]);
     End;
     ASRS := SRS/Nrow;
End;   (NLIN)


(ΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔ
ΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔΔ)
 (
)
 (                             PROCEDURE SETUP
)

(____________________________________________________________
________________)


Procedure SetUp (Var Select : Integer);
Var
   OK  :  Boolean;

Begin
   ClrScr;
   Writeln ('         M E N U          ');
   Writeln;
   Writeln ('1. CREATE DATA FILE.');
   Writeln ('2. RUN CURVE FITTING PROGRAM.');
   Writeln ('3. END.');
   Writeln;
   Readln (Select);
   DDrive := ':B';
   Case Select Of
      1   : Begin
               ClrScr;
               Write ('Data File name (max of 6 characters)
= ');
```

```
               Readln (DataFile);
            End;
      2   : Begin
               ClrScr;
               Repeat
                  Write ('Which Data File to be processed =
? ');
                  Readln (DataFile);
                  StoDataFile := DataFile + '.Dat';
                  Assign (Data, StoDataFile);
                  ($I-) Reset (Data) ($I+);
                  OK := (IOresult = 0);
                  If Not OK then
                     Writeln ('Cannot find file
',StoDataFile);
               Until OK;
               ProcessFile := DataFile + '_';
               Readln (Data, DeviceID);
               Readln (Data, Lg, Lz, Temp, Vto, Vpts);
               Writeln;
               Write ('Alpha = ? ');           Readln
(Alpha);
               Write ('Ns    = ? ');           Readln (Ns);
               Sequence := 0;
               SequenceNum := '0';
            End;
      3   : Exit;
   End;
End;


(***********************************************************
****************)
(
)
(                         M A I N     P R O G R A M
)
(
)
(***********************************************************
****************)


Begin
   Select := 1;
   Repeat
      SetUp (Select);
      ClrScr;
      Case Select Of
         1    : DiskStorage;
         2    : Begin
                  Set_Vthv (Vthv, Temp);
                  For_I := 1 To Vpts Do
```

```
                  Begin
                     Readln (Data, Vds[I], DataPts);
                     Get_Data(X, Y, DataPts, Vds[I]);
                     Nlin (X, Y, Y_Calc, DataPts);
                     Print_Data(Vds[I], DataPts);
                     Write ('Store Data on Disk? (y/n) ');
                     Readln (Store);
                     If Store = 'y' then
                        Store_Data(Vds[I], DataPts);
                  End;
                  Close (Data);
                End;
         3    : ;
      End;
   Until Select = 3;
End.
```

7.6 VREFSIM Program Listing

```
½                           )
(   N O R M A L             )
(                           )
(Circuit1 is the program to simulate the normal device
operation. It runs      )
(from a temperature of 400K to 220K. The user has to change
the lambda and    )
(and Alpha manually in the constant declaration. The (W/L)
scale the Tempe    )
(coefficient of the device. The mimimum current must be
greater than 1mA.     )
(     INPUT  : TEMP HIGH, TEMP LOW, TEMP STEP, RS
)
(     OUTPUT : VOUT, IM
)


Program Circuit1;

Const
   Es     = 1.1599E-12;
   Q      = 1.60218E-19;
   K      = 1.3806E-23;
   t      = 6.6E-6;
   Nc     = 4.0E17;
   ni     = 2.0E6;
   U300   = 3500;
   T300   = 300;
   Vdd    = 0.74;
   Vss    = -2.0;
   Lambda = 0.05;
   Alpha  = 2.2;

Type
   Element = Array[1..25] of Real;

Var
   Vds, Vgs                       : Real;
   Vthv, Vto, Vp                  : Real;
   Vout, Vt, Vr, Vdif             : Real;
   Temp, Vg                       : Real;
   W, L, Psi                      : Real;
   Beta                           : Real;
   Rs, Rsi                        : Real;
   Im, Ir, Ilast                  : Real;
   TempHi, TempLow, TempStep      : Real;
   I, P, LastElem                 : Integer;
   Okay, FirstPass                : Boolean;
   Data                           : Text;
   DataFile                       : String[12];
   StoDataFile                    : String[16];
```

```
   IBias                           : Element;

Procedure SetUp;

Begin
   ClrScr;
  ( Write ('High Temperature = ? ');        Readln (TempHi);
   Write ('Low  Temperatute = ? ');        Readln (TempLow);
   Write ('Temperature Step = ? ');        Readln
(TempStep);)
   TempHi := 400;    TempLow := 220;  TempStep :=10;
   Write ('DataFile = ? ');                Readln
(DataFile);

   Vout := 0.0;
   Vg := 0.0;
   Write ('W = ?');   Readln(W);
   L := 0.25;
   Rs := 2/(1.4129E-4*(W/L));        Writeln('Rs =
',Rs:1:0);
   Psi := 0.0001;
   Rsi := Rs/(1+Psi*(T300-273));
End;


Function Tanh (X : Real): Real;

Var
   A, B  :  Real;

Begin
   A := Exp(X);
   B := Exp(-X);
   Tanh := (A-B)/(A+B);
End;


Procedure  Im_Solve;

Begin
   Beta  := (2*W*U300*T300*Es*Sqr(292/Temp))/(3*L*Temp*t);
   Vds  := Vdd - Vout;
   Vgs  := Vg - Vout;
   Vp   := (Q*Nc*Sqr(t)/(2*Es))*(Temp/292);
   Vthv := (K/Q)*Temp;
   Vto  := Vthv*Ln(Nc/ni)-Vp;
   Vt   := Vto-0.07*Vds;
   Im   := Beta*Sqr(Vgs-Vt)*(1+Lambda*Vds)*Tanh(Alpha*Vds);
   (writeln('Beta=',Beta:6:5,' Vp=',Vp:4:3,'
Vthv=',Vthv:4:3,' Vto=',Vto:4:3,' Vt=',Vt:4:3);)
End;


Procedure Ir_Solve;
```

```
Begin
   Vr := Vout - Vss;
   Rs := Rsi*(1+Psi*(Temp-273));
   Ir := (Vr)/Rs;
End;


Procedure Test_Bias (Var OKAY : Boolean);

Const
   Tol  = 0.0001;

Var
   Itest  :  Real;

Begin
   Itest := Im/Ir - 1;
   If ABS(Itest) <= Tol Then
     OKAY := True
   Else
     Begin  (ABS)
       If FirstPass Then
          Begin  (FirstPass)
             Vdif := 0.0005;
             If Itest > 0 then
                Vout := Vout + Vdif
             Else
                Vout := Vout - Vdif;
             FirstPass := False;
          End  (FirstPass)
       Else
         Begin  (Subsequent Passes)
           If Itest > 0 then
             Begin
               If Ilast < 0 then Vdif := Vdif*0.5;
               Vout := Vout + Vdif;
             End
           Else
             Begin
               If Ilast > 0 then Vdif := Vdif*0.5;
               Vout := Vout - Vdif;
             End;
         End;  (Subsequent Passes)
     End;  (ABS)
     Ilast := Itest;
End;

Procedure Store_Data (Ibias : Element; LastElem: integer);
Var
  J  : Integer;

Begin
  StoDataFile := DataFile+'.Dat';
```

```
  Rs := Rsi*(1 + Psi*(300-273));
  Assign (Data, StoDataFile);
  Rewrite (Data);
  Writeln (Data, 'W :',W:1:1);
  Writeln (Data, 'Rs :',Rs:1:0);
  For J := 1 to LastElem Do
    Writeln (Data, Ibias[J]);
  Flush (Data);
  Close (Data);
End;


Begin
   SetUp;
   Temp := TempLow;
   P := 0;
   Writeln ('Temp   Vg      Vout     Vgs      Im
Ir');
   Repeat
      OKAY := False;
      P := P + 1;
      FirstPass := True;
      I := 0;
      Repeat
         I := I + 1;
         Im_Solve;
         Ir_Solve;
         Test_Bias (OKAY);
         If Keypressed then exit;
      Until (OKAY = True) OR (I > 1000);
      If I >= 1000 then
       Begin
        Writeln ('No convergence. Program terminated.');
        Exit
       End;
      Ibias[P] := Im;
      If (Vg-Vout)<0 then
        Writeln (Temp:0:0,'   ',Vg:5:3,'    ', Vout:5:3,'
',(Vg-Vout):3:3,'   ',Im:10,'   ',Ir:10)
      Else
        Writeln (Temp:0:0,'   ',Vg:5:3,'   ', Vout:5:3,'
',(Vg-Vout):3:3,'   ',Im:10,'   ',Ir:10);
      Temp := Temp + TempStep;
   Until Temp >= TempHi + Tempstep;
   LastElem := P;
   Store_Data(Ibias, LastElem);
End.
```

```
(                           )
(  S U B _ D E V            )
(                           )
(THE PROGRAM SIMULATE THE POSITIVE TEMPERATURE DRIFT
COMPONENT PARTOF THE        )
(CIRCUIT. IT CONSISTS OF A TRANSISTOR IN SUBTRESHOLD REGION,
AND TWO RESISTOR.)
(   INPUT   :   temp high, Temp low, Rs, R2
)
(   OUTPUT  :   IM
)


Program SUB_DEV;

Const
   Es      = 1.1599E-12;
   Q       = 1.60218E-19;
   K       = 1.3806E-23;
   t       = 6.6E-6;
   Nc      = 4.0E17;
   ni      = 2.0E6;
   U300    = 3500;
   T300    = 300;
   Vdd     = 0.74;
   Vss     = -2.0;
   Gamma = 0.07;
   Alpha   = 0.046;
   Ns      = 1.671;

Var
   Vthv, Vto, Vp                    : Real;
   Vout, Vdif                       : Real;
   Temp, Vg                         : Real;
   W, L, Rs, Rsi, Psi, R2, R2i      : Real;
   Im, Ir, Ir2, Ilast, Io           : Real;
   TempHi, TempLow, TempStep        : Real;
   I                                : Integer;
   Okay, FirstPass                  : Boolean;


Procedure SetUp;

Begin
   ClrScr;
   Write ('High Temperature = ? ');        Readln (TempHi);
   Write ('Low  Temperatute = ? ');        Readln (TempLow);
   Write ('Temperature Step = ? ');        Readln
(TempStep);
   Write ('Rs = ? ');                       Readln (Rs);
   Write ('R2 = ? ');                       Readln (R2);
   Vout := -1.25;
   Vg := -2.0;
   W := 50;
```

```
   L := 0.25;
   Psi := 0.0001;
   Rsi := Rs/(1+Psi*(T300-273));
   R2i := R2/(1+Psi*(T300-273));
   Io := 1.0E-4;
End;



Procedure Calc_TempVar;

Begin
   Vp   := (Q*Nc*Sqr(t)/(2*Es))*(Temp/292);
   Vthv := (K/Q)*Temp;
   Vto  := Vthv*Ln(Nc/ni)-Vp;
   Rs   := Rsi*(1+Psi*(Temp-273));
   R2   := R2i*(1+Psi*(Temp-273));
End;

Procedure  Im_Solve;
Var
  Vds, Vgs, Vt :Real;

Begin
   Vds  := Vdd - Vout;
   Vgs  := Vss - Vout;
   Im   := Io*Exp(((1-Alpha*Vds)/(Ns*Vthv))*(Vgs-
Vto+Gamma*Vds));

End;


Procedure Ir_Solve;

Var
   Vr, Vr2 : Real;
Begin
   Vr := Vout-Vss;
   Ir := Vr/Rs;
   Vr2 := 3.3 - Vout;
   Ir2 := Vr2/R2
End;


Procedure Test_Bias (Var OKAY : Boolean);

Const
   Tol  = 0.0001;

Var
   Itest  :  Real;

Begin
   Itest := Im/(Ir - Ir2) - 1;
```

```
   If ABS(Itest) <= Tol Then
     OKAY := True
   Else
     Begin  (ABS)
       If FirstPass Then
          Begin  (FirstPass)
             Vdif := 0.0005;
             If Itest > 0 then
                Vout := Vout + Vdif
             Else
                Vout := Vout - Vdif;
             FirstPass := False;
          End  (FirstPass)
       Else
         Begin  (Subsequent Passes)
           If Itest > 0 then
             Begin
               If Ilast < 0 then Vdif := Vdif*0.5;
               Vout := Vout + Vdif;
             End
           Else
             Begin
               If Ilast > 0 then Vdif := Vdif*0.5;
               Vout := Vout - Vdif;
             End;
         End;  (Subsequent Passes)
     End;  (ABS)
     Ilast := Itest;
End;


Begin
   SetUp;
   Temp := TempLow;
   Writeln ('Temp   Vref    Vout      Vgs      Im
Ir            Ir2');
   Repeat
      OKAY := False;
      FirstPass := True;
      I := 0;
      Calc_TempVar;
      Repeat
         I := I + 1;
         Im_Solve;
         Ir_Solve;
   (      Writeln('Vout=',Vout:5:4,' Im=',Im:8,' Id=',(Ir-
Ir2):8,' Ir=',Ir:8,' Ir2=',Ir2:8);
         If I mod 20 = 0 Then Repeat Until KeyPressed;)
         Test_Bias (OKAY);
         If KeyPressed then Exit;
      Until (OKAY = True) OR (I > 10000);
      If I > 10000 then
       Begin
        Writeln ('No convergence. Program terminated.');
```

```
        Exit
       End;
      Writeln (Temp:0:0,'   ',(Vdd-Vout):5:3,'
',Vout:5:3,'    ',(Vg-Vout):4:3,
               '   ',Im:10,'   ',Ir:10,'  ',Ir2:10);
      Temp := Temp+TempStep;
   Until Temp >= TempHi + Tempstep
End.
```

```
(                          )
(    S U B _ R 1 R 3       )
(                          )
(                          )
(simulate the subthreshold device, the positive temperature
drift portion)
(of the vref circuit. The program step through a given range
of desire    )
(bias current. Temp high, temp low and temp Step are preset
)
(   input :  Desire Vgs
)
(   output:  R1, R3, Icoeff, Im
)

($F20)

Program SUB_R1R3;

Const
   Es      = 1.1599E-12;
   Q       = 1.60218E-19;
   K       = 1.3806E-23;
   t       = 6.6E-6;
   Nc      = 4.0E17;
   ni      = 2.0E6;
   U300    = 3500;
   T300    = 300;
   Vdd     = 0.74;
   Vss     = -2.0;
   Gamma = 0.07;
   Alpha   = 0.046;
   Ns      = 1.671;

Var
   Vthv, Vto, Vp                    : Real;
   Vout, Vdif                       : Real;
   Temp, Vgs                        : Real;
   W, L, R1, R1i, Psi, R2, R2i      : Real;
   IMin, IMax, IStep                : Real;
   Im, I1, I2, Ilast, Io,I          : Real;
   TempHi, TempLow, TempStep        : Real;
   J, Last, P                       : Integer;
   Okay, FirstPass, Init, Skip      : Boolean;
   Ij                               : Array[1..20] of real;
   Data                             : Text;
   DataFile                         : String[14];
   StoDataFile                      : String[14];
   Sequence                         : Integer;
   SequenceNum                      : String[2];
Label
  Next;
```

```
Procedure SetUp;

Begin
   ClrScr;
   TempHi := 400;    TempLow := 220;   TempStep := 10;
   IMin   := 1.6E-3;   Imax    := 2.1E-3;  IStep    := 5E-5;
   Write ('Vgs (must be -ve number) = ?');      Readln
(Vgs);
   Write ('Data File Name = ? ');              Readln
(DataFile);
   Sequence :=0;
   SequenceNum := '0';
   W := 50;
   L := 0.25;
   Psi := 0.0001;
   Io := 1.0E-4;
End;


Procedure Temp_300Var;

Var
   Vds : Real;

Begin
   Vout := Vss - Vgs;
   Vds := Vdd - Vout;
   R1 := (Vout-Vss)/I1;
   R1i := R1/(1 + Psi*(300-273));
   Im  := Io*Exp((1-Alpha*Vds)/(Ns*Vthv)*(Vgs - Vto +
Gamma*Vds));
   I2 := (I1-Im);
   R2 := (3.3-Vout)/I2;
   R2i := R2/(1+Psi*(300-273));
End;


Procedure Calc_TempVar;

Begin
   Vp   := (Q*Nc*Sqr(t)/(2*Es))*(Temp/292);
   Vthv := (K/Q)*Temp;
   Vto  := Vthv*Ln(Nc/ni)-Vp;
   If Not Init Then
   Begin
     R2   := R2i*(1+Psi*(Temp-273));
     R1   := R1i*(1+Psi*(Temp-273));
   End
End;

Procedure  Im_Solve;
Var
  Vds, Vgs, Vt :Real;
```

```
Begin
   Vds  := Vdd - Vout;
   Vgs  := Vss - Vout;
   Im   := Io*Exp(((1-Alpha*Vds)/(Ns*Vthv))*(Vgs-
Vto+Gamma*Vds));

End;


Procedure Ir_Solve;

Var
   V1, V2 : Real;
Begin
   V1 := Vout - Vss;
   I1  := V1/R1;
   V2 := 3.3 - Vout;
   I2  := V2/R2;
End;


(*Procedure Test_Bias (Var OKAY : Boolean);

Const
   Tol = 0.001;

Var
   Itest : Real;

Begin
   Itest := Im/(I1 - I2) -1;
   If ABS(Itest) <= Tol Then OKAY := True
   Else
     Begin
        OKAY := False;
        If (ABS(Itest) > 1)                             Then
Vdif := 0.05;
        If (ABS(Itest) < 1) And (ABS(Itest) >= 0.9)     Then
Vdif := 0.01;
        If (ABS(Itest) < 0.9) And (ABS(Itest) >= 0.8)   Then
Vdif := 0.005;
        If (ABS(Itest) < 0.8) And (ABS(Itest) >= 0.7)   Then
Vdif := 0.001;
        If (ABS(Itest) < 0.7) And (ABS(Itest) >= 0.6)   Then
Vdif := 5e-4;
        If (ABS(Itest) < 0.6) And (ABS(Itest) >= 0.5)   Then
Vdif := 1e-4;
        If (ABS(Itest) < 0.5) And (ABS(Itest) >= 0.4)   Then
Vdif := 5e-5;
        If (ABS(Itest) < 0.4) And (ABS(Itest) >= 0.3)   Then
Vdif := 1e-5;
        If (ABS(Itest) < 0.3) And (ABS(Itest) >= 0.2)   Then
Vdif := 5e-6;
```

```
        If (ABS(Itest) < 0.2) And (ABS(Itest) >= 0.1)   Then
Vdif := 1e-6;
        If (ABS(Itest) < 0.1) And (ABS(Itest) >= 0.05)  Then
Vdif := 5e-7;
        If (ABS(Itest) < 0.05) And (ABS(Itest) >= 0.01) Then
Vdif := 1e-7;
        If (ABS(Itest) < 0.01)                          Then
Vdif := 5e-8;
        If Itest > 0 then
           Vout := Vout + Vdif
        Else
           Vout := Vout - Vdif;
      End;
   Writeln ('Itest=',Itest:10,' Im= ',Im:10,' I1-I2=',(I1-
I2):10,
           ' Vdif= ',Vdif:10,' Vout=',Vout:4:3);
End;*)



Procedure Test_Bias (Var OKAY : Boolean);

Const
   Tol  = 0.001;

Var
   Itest  :  Real;

Begin
   Itest := Im/(I1 - I2) - 1;
   If ABS(Itest) <= Tol Then
     OKAY := True
   Else
     Begin  (ABS)
       If FirstPass Then
          Begin  (FirstPass)
             Vdif := 0.0001;
             If Itest > 0 then
                Vout := Vout + Vdif
             Else
                Vout := Vout - Vdif;
             FirstPass := False;
          End  (FirstPass)
       Else
         Begin  (Subsequent Passes)

            If Itest > 0 then
              Begin
                If Ilast < 0 then Vdif := Vdif*0.5;
                  Vout := Vout + Vdif
              End
            Else
              Begin
                If Ilast > 0 then Vdif := Vdif*0.5;
```

```
                 Vout := Vout - Vdif
             End;
         End;  (Subsequent Passes)
     End;  (ABS)
     Ilast := Itest;
   (  Writeln ('Itest= ',Itest:10,' Im= ',Im:10,' I1-I2=
',(I1-I2):10,' I= ',I:4);)
End;

Procedure Store_Data;

Var
   J  : Integer;

Begin;
   StoDataFile := DataFile+'_'+SequenceNum+'.Dat';
   Assign (Data, StoDataFile);
   Rewrite (Data);
   For J := 1 to Last do
     Writeln(Data, Ij[J]);
   Flush (Data);
   Close (Data);
   Sequence := Sequence + 1;
   Str(Sequence, SequenceNum);
End;

Begin
   SetUp;
   I := Imin;
   Repeat
     Temp := 300;
     I1   := I;
     Init := True;
     Calc_TempVar;
     Init := False;
     Temp_300Var;
     Skip := False;
     Temp := TempLow;
     P := 0;
     Write ('I= ',I:8,' Im= ',Im:10,' Vgs= ',Vgs:1:3,' R1=
',R1:4:0,' R2= ',R2:5:0);
(     Writeln ('Temp   Vref    Vout      Vgs      Im
I1-I2');)
     Repeat
        P := P+1;
        OKAY := False;
        FirstPass := True;
        J := 0;
        Calc_TempVar;
        Repeat
           J := J + 1;
           Im_Solve;
           Ir_Solve;
           Test_Bias (OKAY);
```

```
           If KeyPressed then exit;
        Until (OKAY = True) OR (J > 2000);
        If J > 2000 then
          Begin
            Writeln (' No convergence.');
            Temp := TempHi + TempStep;
            Skip := true;
            Goto Next;
          End;
      (  Writeln (Temp:0:0,'    ',(Vdd-Vout):5:3,'
',Vout:5:3,'    ',(Vss-Vout):4:3,
                 '   ',Im:10,'   ',(I1-I2):10);)
          Ij[P] := Im;
 Next:    Temp := Temp+TempStep;
     Until Temp >= TempHi + Tempstep;
     Last := P;
    ( Store_Data;)
     If not¯Skip then
     Writeln ( ' Icoef = ',(Abs(Ij[1]-Ij[Last])/180):10);
     I := I + Istep;
  Until I >= Imax + Istep;
End.
```

```
(                             )
(         C K T S I M         )
(                             )
(                             )
Program Circuit3;

Const
   K   = 1.3806E-23;
   Q   = 1.60218E-19;
   Es  = 1.1599E-12;
   Nc  = 4.0E17;
   ni  = 2.0E6;
   t   = 6.6E-6;
   Ido = 1.0E-4;
   Vdd = 3.3;
   Vss = -2.0;
   W1   = 50;
   L1   = 0.25;    L2 = 0.25;
   Alpha  = 0.046;
   Ns     = 1.671;
   Gamma  = 0.07;            (Vds Coefficient of Isub)
   Lambda = 0.05;
   Theta  = 2.2;
   Psi    = 0.0001;          (Temperature Coefficient of
resistor)
   T300   = 300;
   U300   = 3500;

Var
   Vo, V1, V2                            :  Real;
   Io, I1, I2, I3                        :  Real;
   Ro, R1, R2, R3                        :  Real;
   Vthv, Vbi, Vp, Vto, Vt                :  Real;
   Roi, R1i, R2i, R3i                    :  Real;
   Ilast, Ibias, Vbias                   :  Real;
   Vdif, Idummy                          :  Real;
   Temp, TempHi, TempLo, TempStep        :  Real;
   Beta, W2                              :  Real;
   FirstPass, Okay                       :  Boolean;
   Pass, Dummy, Converge                 :  Boolean;
   C, Last                               :  Integer;
   Data                                  :  Text;
   DataFile                              :  String[12];
   StoDataFile                           :  String[16];
   Vref                                  :  Array [1..200] of
Real;

Procedure SetUp;

Begin
   ClrScr;
(   Write ('High Temperature = ? ');          Readln
(TempHi);
```

```
   Write ('Low  Temperature = ? ');          Readln
(TempLo);
   Write ('Temperature Step = ? ');          Readln
(TempStep);)
   TempHi := 400;  TempLo := 220;  TempStep := 10;
   Write ('Ro     = ? ');                        Readln
(Ro);
   Write ('R1     = ? ');                        Readln
(R1);
   Write ('R2     = ? ');                        Readln
(R2);
   Write ('R3     = ? ');                        Readln
(R3);
   Write ('W2     = ? ');                        Readln
(W2);
   Write ('DataFile = ? ');                      Readln
(DataFile);

   Roi := Ro/(1 + Psi*(T300-273));
   R1i := R1/(1 + Psi*(T300-273));
   R2i := R2/(1 + Psi*(T300-273));
   R3i := R3/(1 + Psi*(T300-273));
End;


Procedure Initial_Guess;

Begin
   Vo := 0.74;
   V1 := -1.25;
   V2 := 0.0;
End;


Procedure Calc_TempVar (Var Vthv, Vto      : Real;
                        Var Ro, R1, R2, R3 : Real);

Begin
   Vthv := (K/Q)*Temp;
   Vbi  := Vthv*Ln(Nc/ni);
   Vp   := ((Q*Nc*Sqr(t)/(2*Es))*(Temp/292));
   Vto  := Vbi - Vp;

   Ro   := Roi*(1 + Psi*(Temp-273));
   R1   := R1i*(1 + Psi*(Temp-273));
   R2   := R2i*(1 + Psi*(Temp-273));
   R3   := R3i*(1 + Psi*(Temp-273));
End;


Function Tanh (X : Real) : Real;

Var
   A, B  : Real;
```

```
Begin
   A    := Exp(X);
   B    := Exp(-X);
   Tanh := (A-B)/(A+B);
End;


Procedure Calc_Ibias (    Branch      : Integer;
                          Var Ir, Im  : Real;
                          V           : Real);

Var
  Vds, Vgs : Real;
  Ip, I1, I3  : Real;

Begin
   Case Branch of
     0 :   Begin
              Ir := (Vdd - V)/Ro;
              Im := 0.0;
           End;
     1 :   Begin
              Vds := Vo - V;
              Vgs := Vss - V;
              Im  := Ido*Exp((1-Alpha*Vds)/(Ns*Vthv)*(Vgs-
Vto+Gamma*Vds));
              I1  := (V - Vss)/R1;
              I3  := (Vdd - V)/R3;
              Ir  := (I1 - I3);
           End;
     2 :   Begin
              Vgs := 0.0 - V;
              Vds := Vo - V;
              Ip  :=
(W2/L2)*(Sqr(Q*Nc*t)*t*U300*T300/Temp)/(6*Es);
              Beta := Ip/Sqr(Vp);
              Vt  := Vto - Gamma*Vds;
              Im := Beta*Sqr(Vgs-
Vt)*(1+Lambda*Vds)*Tanh(Theta*Vds);
              Ir := (V-Vss)/R2;
           End;
  End;
End;


Procedure Test_Bias ( Ir, Im      : Real;
                      Var Vbias   : Real;
                      Var Okay    : Boolean);

Const
   Tol = 0.0005;
```

```
Var
   Itest  : Real;

Begin
   Itest := Abs(Ir/Im) - 1;
   If Abs(Itest) <= Tol Then
      Okay := True
   Else
      Begin
         If FirstPass then
            Begin   (FirstPass)
               Vdif := 0.0005;
               If Itest > 0 then
                  Vbias := Vbias - Vdif
               Else
                  Vbias := Vbias + Vdif;
               FirstPass := False;
            End   (FirstPass)
         Else
            Begin   (Subsequent Passes)
               If Itest > 0 then
                  Begin
                     If Ilast < 0 then Vdif := Vdif*0.5;
                     Vbias := Vbias - Vdif;
                  End
               Else
                  Begin
                     If Ilast > 0 then Vdif := Vdif*0.5;
                     Vbias := Vbias + Vdif;
                  End;
            End;   (Subsequent Passes)
      End;  (Abs(Itest)
   Ilast := Itest;
End;


Procedure Set_Bias (Branch       : Integer;
                    Var I, V     : Real;
                    Var Converge : Boolean;
                    Var Flag     : Boolean);

Var
   Im, Ir, Io       : Real;
   J                : Integer;

Begin
   Vbias := V;
   Ibias := I;
   Okay := False;
   FirstPass := True;
   J := 0;
   Repeat
      J := J + 1;
```

```
      Case Branch of
         0 :    Begin
                   Calc_IBias (0, Io, Idummy, Vbias);
                   Test_Bias ((I1+I2), Io, Vbias, Okay);
                   Ibias := Io;
                End;
         1 :    Begin
                   Calc_IBias (Branch, Ir, Im, Vbias);
                   Test_Bias (Ir, Im, Vbias, Okay);
                   Ibias := Im;
                End;
         2 :    Begin
                   Calc_IBias (2, Ir, Im, Vbias);
                   Test_Bias (Ir, Im, Vbias, Okay);
                   Ibias := Im;
                End;
      End;
   Until (Okay) Or (J >10000);
   If J > 10000 then
      Begin
          Writeln ('No Convergence in branch ',Branch:2,'.
Program Terminated');
          Converge := False;
       End;
    If J = 1 then Flag := True;
    I := Ibias;
    V := Vbias;
End;

Procedure Store_Data;
Var
  I : Integer;

Begin
  StoDataFile := DataFile+'.Dat';
  Assign(Data, StoDataFile);
  Rewrite(Data);
  For I := 1 to Last Do
    Writeln (Data, Vref[I]);
  Flush (Data);
  Close (Data);
End;


Begin
   SetUp;
   Initial_Guess;
   Writeln ('Temp  Vref    Vo      V1      V2      Io
I1          I2');
   Temp := TempLo;
   Pass := False;
   Converge := True;
   C := 0;
   Repeat  (Temperature loop)
```

```
      Calc_TempVar (Vthv, Vto, Ro, R1, R2, R3);
      C:=C + 1;
      Repeat     (Bias Setting Loop)
         Set_Bias (1, I1, V1, Converge, Dummy);  If Not
Converge then Exit;
         Set_Bias (2, I2, V2, Converge, Dummy);  If Not
Converge then Exit;
         Set_Bias (0, Io, Vo, Converge, Pass);   If Not
Converge then Exit;
         If KeyPressed then exit;
      Until Pass = True;
      Vref[C] := (Vdd-Vo);
      If V2 > 0 Then
         Writeln(Temp:0:0,'  ', (Vdd-Vo):5:4,'  ', Vo:4:3,'
',
                 V1:4:3,'   ', V2:4:3,'  ', Io:10,'  ',
I1:10,'  ', I2:10)
      Else
         Writeln(Temp:0:0,'  ', (Vdd-Vo):5:4,'  ', Vo:4:3,'
',
                 V1:4:3,'  ', V2:4:3,'  ',Io:10,'  ',
I1:10,'  ', I2:10);
      Temp := Temp + TempStep;
   Until Temp >= TempHi + TempStep;
   Last := C;
   Store_Data;

End.
```