AN ABSTRACT OF THE THESIS OF

Steven C. F. Lam for the degree of Master of Science in Electrical and Computer Engineering presented on April 29, 1988.

Title: Analytical Model of GaAs MESFET Output Conductance with Frequency and Temperature Dependent Parameters.

Redacted for Privacy

Abstract approved: David J. Allstot

The output characteristics of a conventional GaAs depletion-mode MESFET device have been investigated. One of the important parameters of the small-signal GaAs MESFET model, g_{ds} (output conductance), is shown to be frequency and temperature dependent. Variations in g_{ds} are a serious problem in many analog and digital circuits since gain and propagation delay are strong functions of g_{ds}. An analytical model which incorporates frequency and temperature dependent parameters has been developed and gives satisfactory results as compared with experimental data.

Analytical Model of GaAs MESFET Output Conductance
with
Frequency and Temperature Dependent Parameters

by

Steven C. F. Lam

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirement for the
degree of

Master of Science

Completed April 29, 1988

Commencement June 1988

APPROVED:

Redacted for Privacy

Professor of Electrical & Computer Engineering
in charge of major

Redacted for Privacy

Head of department of Electrical & Computer Engineering

Redacted for Privacy

Dean of Graduate School

Date thesis is presented April 29, 1988

Typed by Steven C. F. Lam for Steven C. F. Lam

To my parents

Bor Lam

Kwok W. Lau

"Trust in the Lord with all your heart and lean not on your own understanding; in all your ways acknowledge Him, and He will make your paths straight."

Prov. 3:5,6

ACKNOWLEDGEMENT

I wish to express my sincere gratitude to my major professor, Dr. David J. Allstot, for his guidance, advice, and encouragement. He has also been a fine example for me as a professional engineer. I am further grateful to professor Roy C. Rathja, professor Stephen M. Goodnick, and professor Curtis R. Cook for serving on my graduate committee.

I am immeasurably indebted to P. Canfield for his great assistance in this research work. Without his support, this work could not have been completed. A special thanks is extended to TriQuint Semiconductor for providing the measurement devices.

I would like to take this opportunity to give my most sincere thanks to Mr. & Mrs. Jing R. Wang and their family for their continued encouragement and prayer.

I am deeply grateful to my brother and sister-in-law Mr. & Mrs. Dennis Lam. Their advice and encouragement were invaluable to me throughout all my college years.

Finally, my deepest appreciation goes to my parents. Their never ending love and support have sustained me through all these years.

TABLE OF CONTENTS

LIST OF FIGURES

ANALYTICAL MODEL OF GaAs MESFET OUTPUT CONDUCTANCE WITH FREQUENCY AND TEMPERATURE DEPENDENT PARAMETERS

I. INTRODUCTION

The GaAs MESFET has become the workhorse of microwave and high-speed digital integrated circuits because of its excellent microwave gain, high packing density, and high carrier mobility. Many workers in the GaAs area have noticed that the output conductance (or drain conductance), g_{ds}, of a GaAs MESFET is a function of frequency and temperature [1,2]. The changes in g_{ds} have caused serious problems in designing analog and digital circuits. Therefore, an accurate model of the GaAs MESFET output conductance is in great demand for circuit simulation applications. The following discussion illustrates the importance of modelling the output conductance for analog and digital circuits.

A. Analog circuit

Fig. 1 depicts a common source amplifier circuit. With proper dc bias V_{GS} (gate-source voltage) and I_{ds} (drain current), a small low frequency signal applied at the gate input will be amplified at the drain output with 180 degrees phase shift. The small-signal low frequency voltage gain is defined as:

$$A_v = \frac{v_o}{v_{in}} \simeq \frac{-g_m}{g_{ds}} \tag{1}$$

where the transconductance is

$$g_m = \left.\frac{\partial I_{ds}}{\partial V_{gs}}\right|_{V_{ds} = \text{const.}} \tag{2}$$

and the drain conductance is

$$g_{ds} = \left.\frac{\partial I_{ds}}{\partial V_{ds}}\right|_{V_{gs} = \text{const.}} \tag{3}$$

Fig. 2 gives a graphical illustration defining g_m and g_{ds} in the saturation region. To achieve high voltage gain, the device must be operated in the saturation region where gm is large and g_{ds} is small. Since g_{ds} plays an important role in small-signal voltage gain, changes with frequency during operation would severely affect the circuit performance.

B. Digital circuit

Fig. 3 depicts a depletion-load inverter wherein C_T represents the total capacitance lumped at the output node. One way to determine the performance of a logic inverter is to analyze its switching characteristic to determine the propagation delay time. Fig. 4 depicts a typical transient response of a logic inverter. The average propagation delay time is defined as:

$$t_P = \frac{t_{PLH} + t_{PHL}}{2} \tag{4}$$

The shorter the delay time, t_P, the faster the switching frequency of an inverter. The delay time from low to high, t_{PLH}, is approximated as:

$$t_{PLH} = \frac{C_T * (V_{OH} - V_{OL}) / 2}{I_{ave}} \tag{5}$$

where I_{ave} is the average load current. If the output conductance of the load is decreased (or the load current is decreased) due to an increase of the channel temperature for example, a longer delay time t_{PLH} will be observed. From the above two examples, it is clear that the output conductance plays an important role in analog and digital circuit design.

Fig. 5 shows typical curves of the frequency effect on the output conductance at different temperatures. At a relatively low temperature, T_1, the output conductance remains constant at low frequencies but starts to increase at a frequency of approximately 1 Hz, and reaches its maximum at a frequency of approximately 100 Hz. This phenomenon is believed to be related to the time constant of the traps in the substrate near the channel-substrate interface. Since the time constant is known to be temperature dependent, a shift of the curve is expected at a higher temperature, T_2, as shown.

Fig. 6 shows a typical curve of the small-signal output conductance vs. gate-source voltage for a GaAs MESFET biased at V_{ds} = 2.5 V (L_g = 1 µm, L_z = 50 µm). The output conductance of the Curtice model [3] follows closely at gate-source voltages near the pinch-off voltage of -1 volt. When V_{gs} is greater than 0.0 V, the output conductance predicted by the Curtice model deviates exponentially from the measured data which obviously introduces a very large error in circuit simulation. The decrease in the measured output conductance for $V_{gs} > 0$ is believed to be related to the degradation of saturation velocity, V_s (applicable to a short channel device), due to self-heating in the channel region of the MESFET. An increase in the drain current causes an increase in the channel temperature. Since V_s is inversely proportional to temperature[11], an increase in drain current causes a decrease in saturation velocity, and this results in lower output conductance.

II. EXPERIMENTAL SETUP

The devices used in the experiments were standard depletion-mode MESFETs fabricated by TriQuint Semiconductor on high-purity undoped LEC GaAs ($\sim 10^6$ Ω-cm) <100> wafers. The gate length and gate width are 1 µm and 50 µm respectively. A non-self aligned recessed-gate technology was used. The channel was formed by ion implantation of n-type Si with a peak doping concentration of approximately 2.0×10^{17} cm^{-3}. The measurement set up is illustrated in Fig. 7.

Since the output conductance is defined as the incremental change of drain current with respect to an incremental change of drain voltage, a function generator was employed at the drain for this measurement. The drain and gate terminals were biased using two separate dc power supplies. A 50 ohm sense resistor on the drain side was used to monitor the dc current. Finally, the small-signal measurements were accomplished using a lock-in amplifier.

The measurements were performed at higher temperatures by connecting a heating filament on the back side of the device. A BASIC program was written (by P. Canfield) on a Tektronix 4054 for controlling all of the measurement procedures.

III. DEVICE PHYSICS

The basic operation of a short channel GaAs MESFET is discussed in this section. A cross section of a GaAs MESFET is shown in Fig. 8.

When a positive potential V_{ds} is applied at the drain, electrons at the source end are attracted towards the drain end, resulting in a current I_{ds} from drain to source. The current I_{ds} is not only a function of V_{ds}, but also a function of gate-to-source voltage V_{gs}. At small V_{ds} (Fig. 9(a)), the depletion layer under the gate is symmetric. At $V_{gs} = 0$, the depletion width is determined by the barrier height of the Schottky barrier. The barrier height for a metal/GaAs system is found to be about 0.8 eV and is virtually independent of the metal work function. The barrier height is thought to be due to "Fermi level pinning" at the semiconductor surface which occurs when the surface state density is 10^{13} cm^{-2} or more [4].

As V_{ds} increases from 0.2 V, the depletion width under the gate is increased near the drain end of the channel (Fig. 9(b)). This arises because the gate is more reverse biased with respect to the channel at the drain end and the potential drop along the channel is greatest at the drain end. As V_{ds} increases even further, the electric field (E_{sat}) at the drain end of the gate gives rise to electron velocity saturation.

For a 1 um gate length, electron velocity saturation occurs at approximately V_{ds} = 1.0 V. Further increases in V_{ds} do not increase the drain current substantially (Fig. 9(c)).

To prevent significant forward current from flowing into the gate, the maximum gate-source voltage is limited to approximately 0.6 V for a barrier height of 0.8 eV.

A. Frequency effect on output conductance

It is believed that the frequency dependence of the output conductance of a GaAs MESFET is intimately related to the midgap or deep level trap, referred to as EL2 [5]-[8]. The EL2 center is the predominant deep donor level in undoped LEC wafers. It has been shown that the EL2 center in GaAs is associated with the antisite defect As_{Ga}, i.e., an arsenic on gallium site [5]. Fig. 10 shows the energy band diagram of a metal - n-type GaAs - SI GaAs system.

Referring to the energy band diagram of Fig. 10, traps are inactive in the channel because most of traps are annihilated by doping with n-type dopants above approximately 10^{17} cm^{-3} [5], and the remaining traps are filled with electrons (the Fermi level in this region is above the energy trap level, E_t). On the other hand, traps in the substrate are partially ionized at room temperature since the Fermi level is near the midgap in the SI GaAs material, and the trapped level is 0.69 eV below the conduction band edge [7]. Due to the nature of positively charged traps after ionization, EL2 is also known as a deep donor. The concentration of deep donors, N_{DD}, in semi-insulating GaAs is approximately 2×10^{16} cm^{-3} [7]. Using Fermi-Dirac statistics, it can be shown that there are approximately only 2% ionized deep donors at room temperature.

When the field is high in the channel, some of the electrons on the channel side of the interface have gained enough kinetic energy to be scattered into the substrate side when electrons overcome the barrier at the interface. Consequently, some electrons are trapped in the EL2 deep levels (Fig. 11).

When electrons are trapped by the ionized deep donors, these centers become electrically neutral ($N_{DD}^+ \rightarrow N_{DD}$), which in effect increases the negative charge on the substrate side of the interface. By charge balance, this increases the depletion width on the channel side of the interface and assists in pinching off the channel (Fig. 12), and as a result, the drain current is decreased. This phenomenon is known as "self-backgating" [7]. Since an incremental change in drain-source voltage results in less change in the drain current due to self-backgating, lower output conductance is observed.

McCament [8] recently observed the transient responses of drain current for positive and negative voltage steps at the drain. With a positive step from 1 to 2 volts (Fig. 13), the drain current increases almost instantaneously but then exhibits an exponential decay to its steady-state value. The time constant of this transient as measured was in the order of microseconds. For a negative step from 2 to 1 volts (Fig. 14), the drain current first decreases almost instantaneously and then rises to its final value with a time constant in the order of milliseconds. The time constant associated with the positive step is believed to be related to the capture rate of electrons in the EL2 traps while the time constant for the negative step is related to the emission rate of electrons from the traps.

An increase in small signal output conductance at higher frequencies is believed to be related to the time constants of the EL2 traps. With reference to Fig. 14a, the key point to understanding the frequency dependent output conductance is to determine when the drain current reaches its equilibrium value during a positive or negative voltage step (small signal) at the drain. At low frequencies, wherein the period of the signal is much greater than the capture/emission time constants, traps can follow the applied signal. This result in drain current equilibrium for both positive and negative steps, and results in a change in the drain current, ΔI_{ds1}. At higher frequencies, traps cannot follow the signal, resulting in a larger change of the drain current, ΔI_{ds2}. Since ΔI_{ds2} is greater than ΔI_{ds1}, the output conductance (Eqn.(3)), increases at higher frequencies. Because the time constant of emission is much longer than the time constant of capture, only the time constant of emission will be taken into account in the first order model.

The well-known semi-empirical formula for time constant of EL2 is given as [9]:

$$\tau = \frac{3.53 \times 10^{-8}}{T^2} \exp\left(\frac{9450}{T}\right) \text{ sec} \tag{6}$$

where T is the absolute temperature in degrees Kelvin. From the above expression, it is clear that the time constant is shorter at a higher temperature. The shorter time constant corresponds to a trap that can follow the signal at a higher frequency. Therefore, at a higher temperature, the output conductance curve is shifted to the right with frequency. This explanation agrees well with the measured curves shown in Fig. 5.

The frequency effect on the output conductance can also be observed on a curve tracer displaying the drain current - voltage characteristic curves. Fig. 15 shows typical drain I-V characteristics at 0.1 and 100 kHz. At 100 kHz, the curve exhibits a larger slope in the saturation region ($\Delta I_{ds2} > \Delta I_{ds1}$), and hence a larger output conductance because the traps cannot follow the signal at this high frequency.

B. Temperature effect on output conductance

The thermal effect on output conductance is believed to be related to the degradation of saturation velocity due to a temperature increase in the channel. Fig. 16 shows the relationship between electron drift velocity and electric field in GaAs. Before the electric field reaches E_p where the electron gains its maximum velocity, the proportionality constant relating the velocity - field curve is known as the low-field mobility, μ_n. Mobility is a measure of the ease of carrier motion within a semiconductor crystal. For a long channel device, low field mobility serves as the proportionality constant in the drain current model. At room temperature, the low-field mobility for n-type GaAs of doping n = 1.2×10^{17} cm^{-3} is approximately 4100 cm^2/v-s [10]. Beyond E_p (high field region) is the region where the electron enters the velocity saturation (high field effect) region. In this high field region, the saturation velocity instead of the mobility determines the drain current value. A 1 µm gate length device generally lies in the transition region between a long channel and a short channel device. From the data of Jacoboni et al. [11], the saturation velocity with a doping concentration of approximately 10^{17} cm^{-3} is found to be inversely proportional to temperature, $V_s \propto T^{-1.0}$. The drain current expression of the Curtice model in the next section is developed for a long channel device. Since the mobility is also inversely proportional to the temperature for n-type GaAs at $n = 1.2 \times 10^{17}$ cm^{-3} [12], the temperature dependent mobility can serve as the effective mobility accounting for the average drift velocity for the 1 μm gate length device. This assumption is sufficiently accurate for first-order analysis.

Fig. 17 shows the temperature distribution in a GaAs MESFET. Due to the current flowing between the drain and the source, heat is generated in the channel. As a matter of fact, most of the heat is generated in a small region under the gate near the drain side since most of the potential drop from V_{ds} appears in this region. This process is known as "self-heating" which creates a temperature difference as much as twenty celsius degrees or more between the channel and the substrate. As V_{ds} increases further, more power will be dissipated in the form of heat in the channel which makes the channel even hotter. As a result, electron mobility decreases, and lower output conductance is observed. A significant self-heating effect is often observed on a curve tracer as a "negative resistance" on the drain I-V curve. In a negative feedback application, this phenomenon is absolutely unacceptable because it can make a stable system unstable by changing negative feedback to positive feedback.

The new mobility value of electrons in the channel can be calculated by knowing the new temperature of the channel. By approximating the region of heat generation in the channel a simple half-cylinder and using heat flow analysis, Hughes et al. [9] derived an equation for the temperature difference, ΔT, between the channel and the substrate:

$$\Delta T = \left(\frac{1}{\pi K}\right)\left(\frac{P}{L_z}\right)\ln\left(\frac{2t_{sub}}{L_g}\right) \tag{7}$$

$\frac{P}{L_z}$ --- dc power dissipation per mm of gate width

L_g --- gate length

t_{sub} --- substrate thickness

K --- temperature dependent thermal conductivity where $K = 5.6T^{-0.87}$ W/mm/ C

This simple equation gives an adequate estimate of the temperature gradient for our purposes. A full 2-D model is necessary to give a more accurate temperature profile in the channel.

IV. ANALYTICAL MODEL

A. The basic model

The analytical model of GaAs MESFET output conductance is based on the Curtice model [3]. Recall the drain current of the Curtice model:

$$I_{ds} = \beta\,(V_{gs}-V_t)^2(1+\lambda V_{ds})\tanh(\alpha V_{ds}) \qquad (8)$$

In this equation, I_{ds} is the drain current, beta (β) is the device parameter equal to $L_z\,\epsilon_s\,\mu/(2aL_g)$ where a is the effective channel thickness, Vgs is the gate-source voltage, V_t is the threshold voltage, Vds is the drain-source voltage, lambda (λ) is a parameter related to drain conductance (lambda $\uparrow$ ==> output conductance $\uparrow$), and alpha (α) is a parameter related to drain current saturation (alpha $\uparrow$ ==> early drain current saturation).

Note that the drain current saturates independently of V_{gs} but at approximately the same V_{ds}. This is because the carriers in the channel reach velocity saturation at $V_{ds} = E_{sat}L$ where E_{sat} is the critical field, and L is the channel length. Since the hyperbolic tangent term reaches 98% of its maximum value at alpha * V_{ds} = 2.2, alpha can be expressed as:

$$\alpha = 2.2/V_{ds} = 2.2/(E_{sat}L) \qquad (9)$$

In the simulation, $\alpha = 2.2$ is used where $E_{sat} = 3.3 \times 10^3$ V/cm and L = 3 μm (for 1 μm gate length). The alpha expression in (9) is valid only for a short gate length device. For a long gate length device

(> 10 µm), the channel pinches off before the critical field E_{sat} is reached. In this case, equation (9) is no longer valid; it should be changed to :

$$\alpha = 2.2/(V_{po} + V_{gs} + V_{bi}) \qquad (10)$$

where V_{po} is the pinch-off voltage and V_{bi} is the built-in voltage of the Schottky barrier (negative value). To pinch off the channel, the gate-source voltage is not the only factor, but V_{ds} also plays a role. The pinch-off voltage V_{po} is defined as:

$$V_{po} = V_{ds} - V_{gs} - V_{bi} \qquad (11)$$

Since it requires $V_{ds} = V_{po} + V_{gs} + V_{bi}$ for the channel to be pinched off, equation (10) is a valid approximation for alpha.

A recent paper by Statz et al. [13] points out the inadequacy of the Curtice model [3]. The drain current as a function of gate-source voltage is poorly modeled, especially when the pinch-off voltage is large. They explained further that I_{ds} should be in quadratic form for small (V_{gs} - V_t), and in linear form for large (V_{gs} - V_t). To better model the drain current as a function of V_{gs}, an additional empirical term is introduced into the denominator of the Curtice model:

$$I_{ds} = \frac{\beta (V_{gs} - V_t)^2}{1 + b(V_{gs}-V_t)} (1 + \lambda V_{ds}) \tanh(\alpha V_{ds}) \qquad (12)$$

The value of b is dependent on the fabrication process. The more gradual the doping profile, the lower the value of b. Since the parameter b determines the $I_{ds} - V_{gs}$ characteristic, it is not an important parameter for small-signal output conductance, but indeed is an important parameter for small-signal transconductance. The parameter b is used in the simulation program for future purposes.

B. Model with Temperature dependent parameter

For n-type GaAs with peak doping concentration $N_d = 2.0 \times 10^{17}$ cm^{-3}, the channel mobility is inversely proportional to temperature. The new mobility at a new temperature can be calculated by using the following relationship:

$$\mu = \mu_o \left(\frac{T_o}{T_o + \Delta T} \right) \tag{13}$$

wherein the subscript "o" represents the value at the substrate temperature in degrees Kelvin. Rewriting Eqn. (7) as:

$$\Delta T = \frac{I_{dso} V_{ds}}{\pi K L_z} \ln \left(\frac{2t_{sub}}{L_g} \right) \tag{14}$$

where I_{dso} represents the drain current evaluated at the substrate temperature, and the other parameters are as defined for Eqn. (7). For a first-order approximation, the temperature difference between the channel and the substrate is calculated by evaluating the drain current without the heating effect at the new V_{ds}.

The parameter beta in the drain current equation can be written as k * μ. Using simple arithmetic, the units for k are coulombs per volt per square meter, $C/V\text{-}m^2$. I_{dso} is thus expressed as:

$$I_{dso} = \frac{k\ \mu_o (V_{gs} - V_t)^2}{1 + b(V_{gs} - V_t)} (1 + \lambda V_{ds}) \tanh(\alpha V_{ds}) \qquad (15)$$

where μ_o is the mobility at the substrate temperature. To determine the temperature dependent output conductance, a new expression for drain current with the heating effect must first be known. This is determined by substituting the temperature dependent mobility $\mu(T)$ into (15):

$$I_{ds} = \frac{k\ \mu(T) (V_{gs} - V_t)^2}{1 + b(V_{gs} - V_t)} (1 + \lambda V_{ds}) \tanh(\alpha V_{ds}) \qquad (16)$$

Combining (13), (15) and (16), I_{ds} is expressed in a simpler form as:

$$I_{ds} = \frac{I_{dso}\ T_o}{T_o + \Delta T} \qquad (17)$$

The temperature dependent output conductance is thus the derivative of the drain current (Eqn. (17)) with respect to the drain-source voltage. The following derivation gives the temperature dependent output conductance expression:

$$g_{ds} = \left.\frac{\partial I_{ds}}{\partial V_{ds}}\right|_{V_{gs} = \text{constant}}$$

$$= \frac{(T_o + \Delta T)\ T_o\left(\dfrac{\partial I_{dso}}{\partial V_{ds}}\right) - I_{dso}T_o\left(\dfrac{\Delta T}{I_{dso}V_{ds}}\right)\left(\dfrac{\partial (I_{dso}V_{ds})}{\partial V_{ds}}\right)}{(T_o + \Delta T)^2}$$

$$= \frac{(T_o + \Delta T)(g_{dso}T_o) - (T_o \Delta T\ g_{dso}) - (T_o \Delta T\ I_{dso}/V_{ds})}{(T_o + \Delta T)^2}$$

$$= g_{dso}\left(\frac{T_o}{T_o + \Delta T}\right)^2 - \left(\frac{T_o \Delta T}{(T_o + \Delta T)^2}\right)\frac{I_{dso}}{V_{ds}}$$

$$\therefore g_{ds} = g_{dso}\left(\frac{T_{sub}}{T_{ch}}\right)^2 - \left(\frac{T_{sub}\,\Delta T}{T_{ch}^2}\right)\frac{I_{dso}}{V_{ds}} \qquad (18)$$

where

g_{dso} =

$$\frac{k\ \mu_o\ (V_{gs}-V_t)^2}{1 + b(V_{gs}-V_t)}\{(1 + \lambda V_{ds})\alpha\ \mathrm{sech}^2\alpha V_{ds} + \lambda \tanh \alpha V_{ds}\},$$

and T_{sub} and T_{ch} are the substrate and channel temperatures, respectively. Therefore, the new output conductance at the new V_{ds} is easily calculated by knowing the new temperature in the channel, the substrate (ambient) temperature, the drain current at the substrate temperature, and the output conductance at the substrate temperature.

C. Model with Frequency dependent parameter

In Fig. 5, the output conductance - frequency curve exhibits a one pole and one zero characteristic. At the zero frequency, g_{ds} rises, and at the pole frequency, g_{ds} becomes constant. From this observation, a frequency term F is expected to be present in the numerator and in the denominator of the equation. Furthermore, as the time constant of the traps play an important role in the frequency dependent output conductance, a term $\tau * F$ (dimensionless) is also expected to be present in the equation. Finally, the model should be able to predict the low constant g_{ds} at low frequencies, and the transition of g_{ds} from low to high values at higher frequencies.

With a little mathematical analysis, a first-order model for frequency-dependent output conductance is developed:

$$g_{ds}(f,T) = \frac{CF\ g_{ds}(dc)}{1 + ZF/(F \cdot \tau)} + g_{ds}(dc)$$

or

$$g_{ds}(f,T) = g_{ds}(dc) \left(\frac{CF\ (F \cdot \tau)}{F \cdot \tau + ZF} + 1 \right) \tag{19}$$

There are two empirical parameters in (14). CF is called the change factor. It determines how much g_{ds} changes from its minimum value to reach $g_{ds}(max)$. ZF is called the zero factor. It determines the frequency where g_{ds} starts the transition towards its maximum value. With larger CF, the transition will be longer. With larger ZF, a higher frequency is required for g_{ds} to begin the transition. The model assumes a constant transition width in terms of frequency.

At low frequencies, the first term in the brackets of eqn. (14) will approach zero which gives $g_{ds}(f,T) = g_{ds}(dc)$. At high frequencies, the first term will approach the change factor CF, and gives $g_{ds}(f,T) = g_{ds}(dc)\{CF+1\}$. By choosing the proper values for CF and ZF, the model is able to match very closely with the measured data.

The time constant in the model is accounted for at different temperatures. Since a higher temperature results in a shorter time constant, the model predicts a higher frequency for g_{ds} to enter the transition region. This is in agreement with the theory presented in section IV.

V. RESULTS

A program was written in PASCAL for doing the simulations, and it is written to be compiled in Turbo Pascal only. Attempts in using other software will lead to compilation failures. Users are required to enter all the necessary information before the program runs properly. To get a hard copy of the graph, simply use the screen dump command - <shift PrtSc>. All the data entered for the simulation will be printed if item 5 is selected in the main menu. The entered parameters values, and the simulation program are listed in the appendices A and B, respectively.

Figs. 18 to 21 show the measured output conductance vs. gate bias at different values of V_{ds}. The output conductance was also measured over a different frequency range (1 Hz - 100 kHz). From these figures, the frequency effect on g_{ds} is noticeable only at $V_{ds} \geq 1$ V where the device is operating in saturation. At this voltage, the electric field in the channel is high enough to cause the electrons to be scattered into the substrate. The temperature effect is also observed for $V_{ds} \geq 1.5$ V.

Fig. 22 shows the measured data at a higher temperature, T = 360 K. The output conductance at T = 360 K is lower than the output conductance at T = 300 K. A better illustration of the frequency effect on g_{ds} is the g_{ds} plot with frequency on the x-axis. The results are depicted in Figs. 23 and 24 at T = 300 K and 360 K respectively. The major difference between these two figures (T = 300 K and T = 360 K) is the shift of the transition region. In Fig. 24 (T = 360 K), g_{ds} starts to rise at a higher frequency. The shift of the transition region demonstrates the presence of the traps on the substrate side of the interface as described earlier.

The results of the theoretical model are compared with the measured data in Figs. 25 and 26. Strictly speaking, output impedance is given by the following expression:

$$Z_d \simeq R_d + \left(\frac{g_m}{g_{ds}}\right) R_s + r_{ds} \tag{20}$$

where R_d and R_s are series drain and source resistances, and r_{ds} is the small signal-signal channel resistance. When the device is operated in the saturation region, the channel resistance r_{ds} becomes the dominant term in eqn. (15). Therefore, the value of g_{ds} of the new model does represent the measured output conductance in the saturation region.

Fig. 25 shows the temperature effect on g_{ds} at V_{ds} = 2.5 V. The new model matches very well with the measured data. It is qualitatively and quantitatively an improvement to the Curtice model as it predicts the temperature effect on g_{ds} very well.

Fig. 26 shows the frequency dependent output conductance at V_{ds} = 2.5 V and V_{gs} = 0.2 V. The theoretical results match excellently with the experimental results. The values of the change factor CF, and the zero factor ZF used in the simulation were 2.3 and 20 respectively.

VI. CONCLUSION

An analytical model of GaAs MESFET output conductance with frequency and temperature dependent parameters was developed. A program was written (Pascal) to simulate the model. The accuracy of the model was tested by comparing the simulated results with the actual data. The results were very satisfactory compared with previous models. For practical applications, this model could be implemented into a circuit simulation program (e.g. SPICE) where the heating and frequency effects could be taken into account.

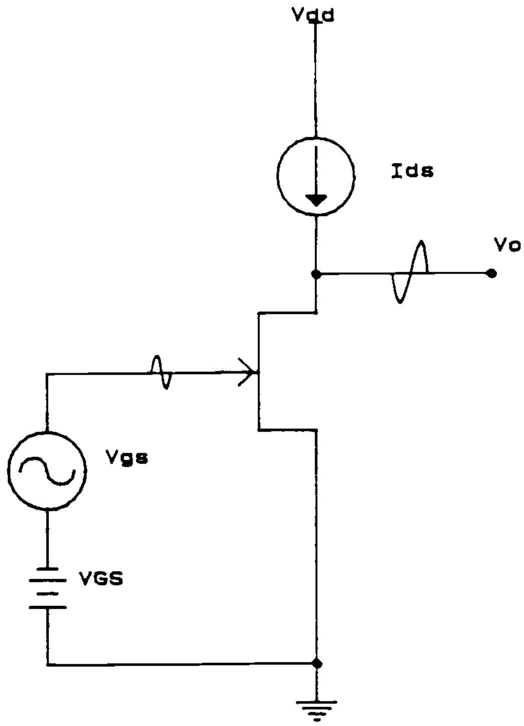

Fig. 1. Common source depletion-mode GaAs MESFET amplifier.

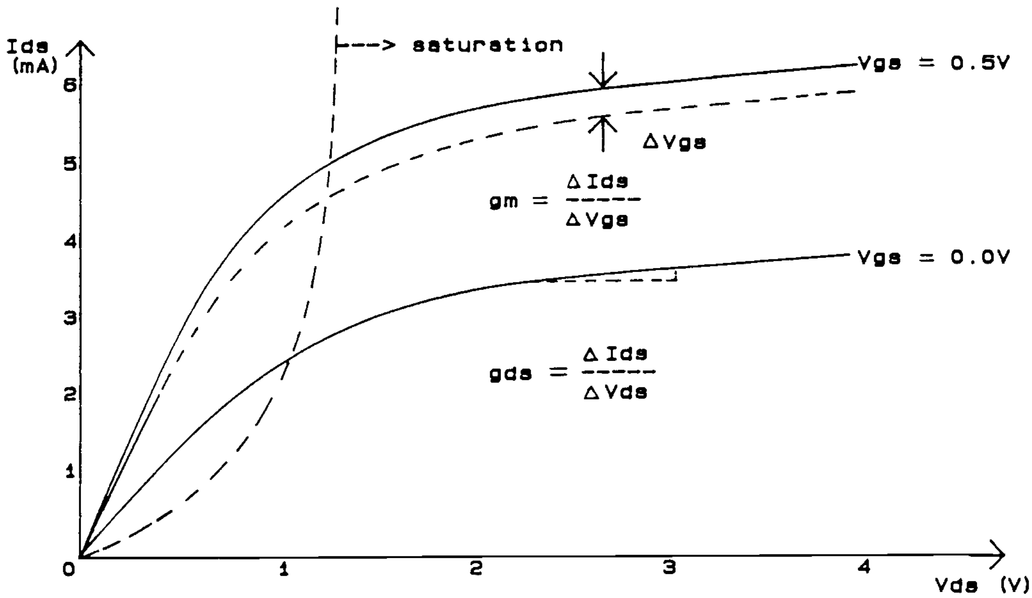

Fig. 2. GaAs MESFET dc current-voltage characteristic curves.

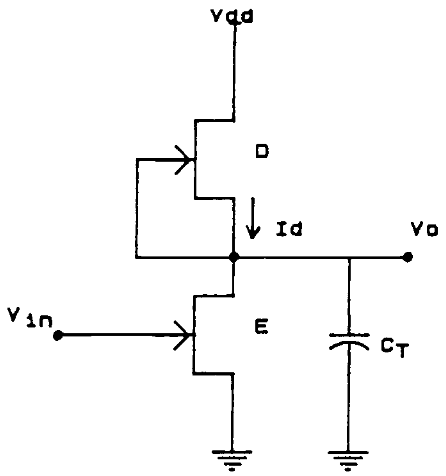

Fig. 3. GaAs MESFET depletion-load inverter.

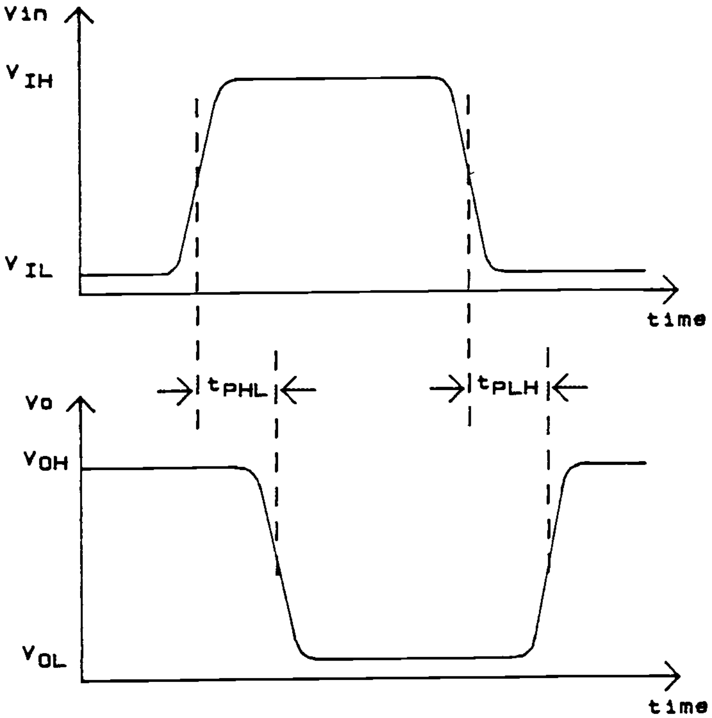

Fig. 4. Transient response of a GaAs MESFET inverter.

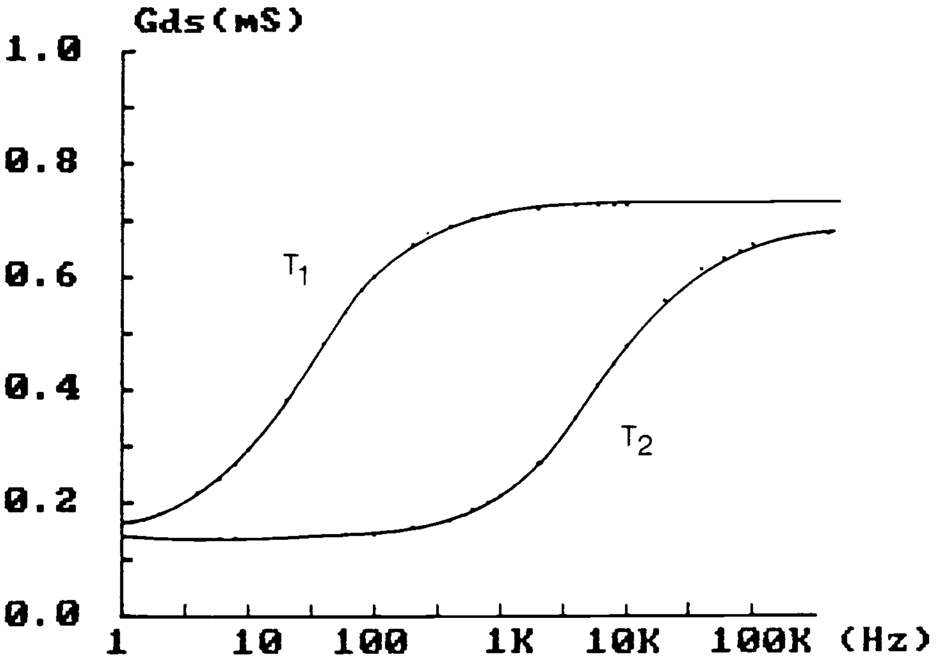

Fig. 5. Typical frequency effect on output conductance of a conventional GaAs MESFET at two different temperatures. $T_2 > T_1$.

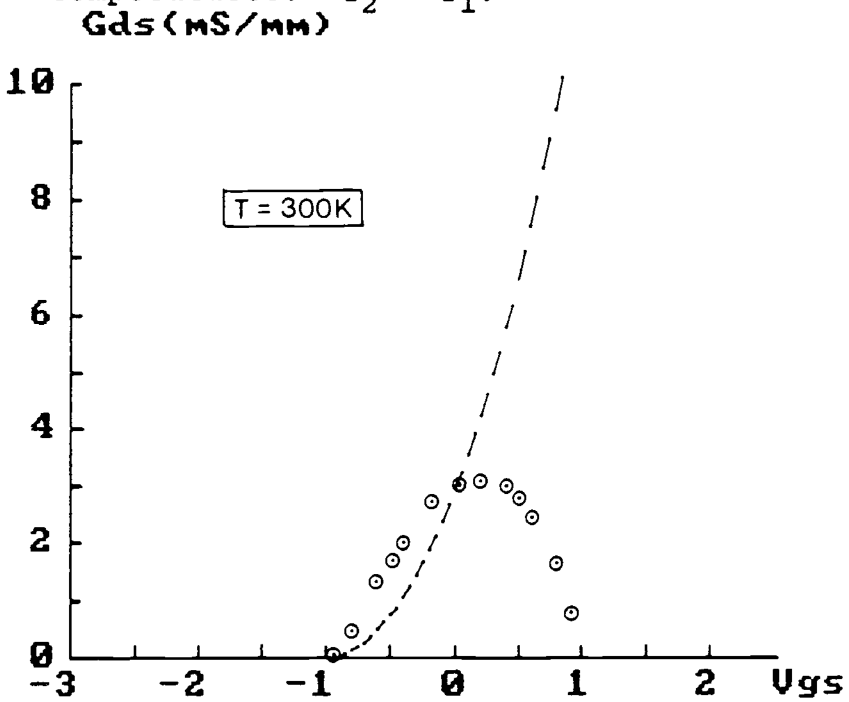

Fig. 6. Temperature effect on output conductance of a conventional GaAs MESFET at Vds = 2.5 V. Curtice model - - - ; Measured data ⊙ ⊙ ⊙ .

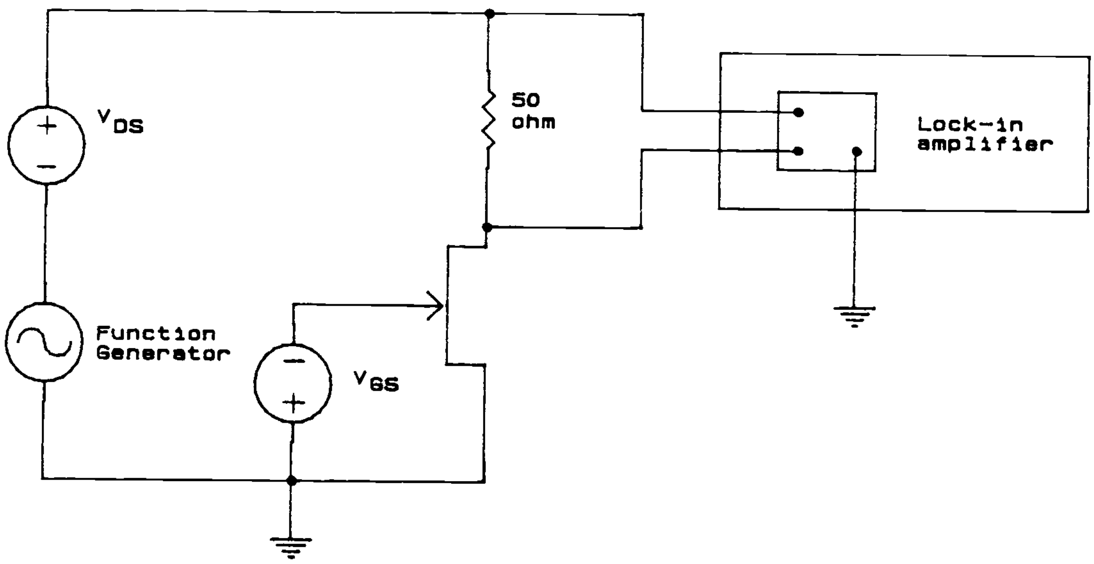

Fig. 7. Output conductance measurement system.

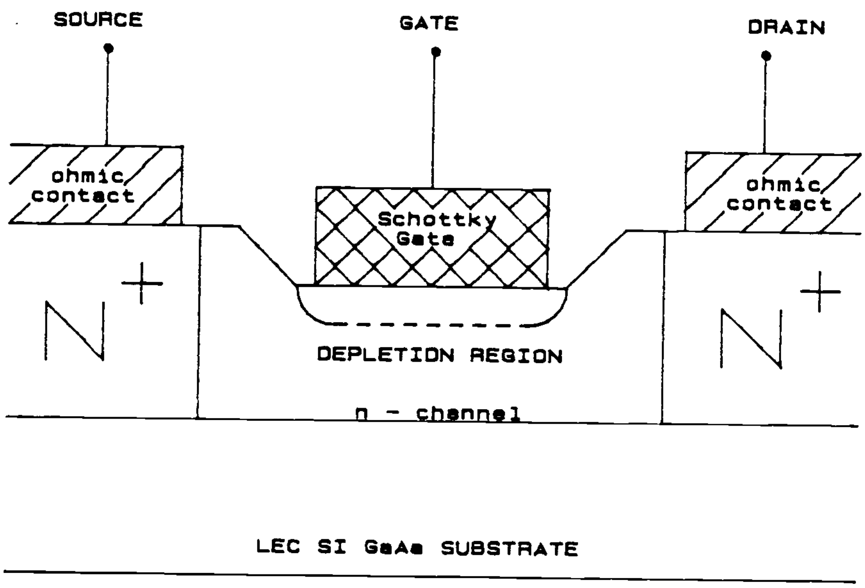

Fig. 8. Cross section of a recessed-gate GaAs MESFET.

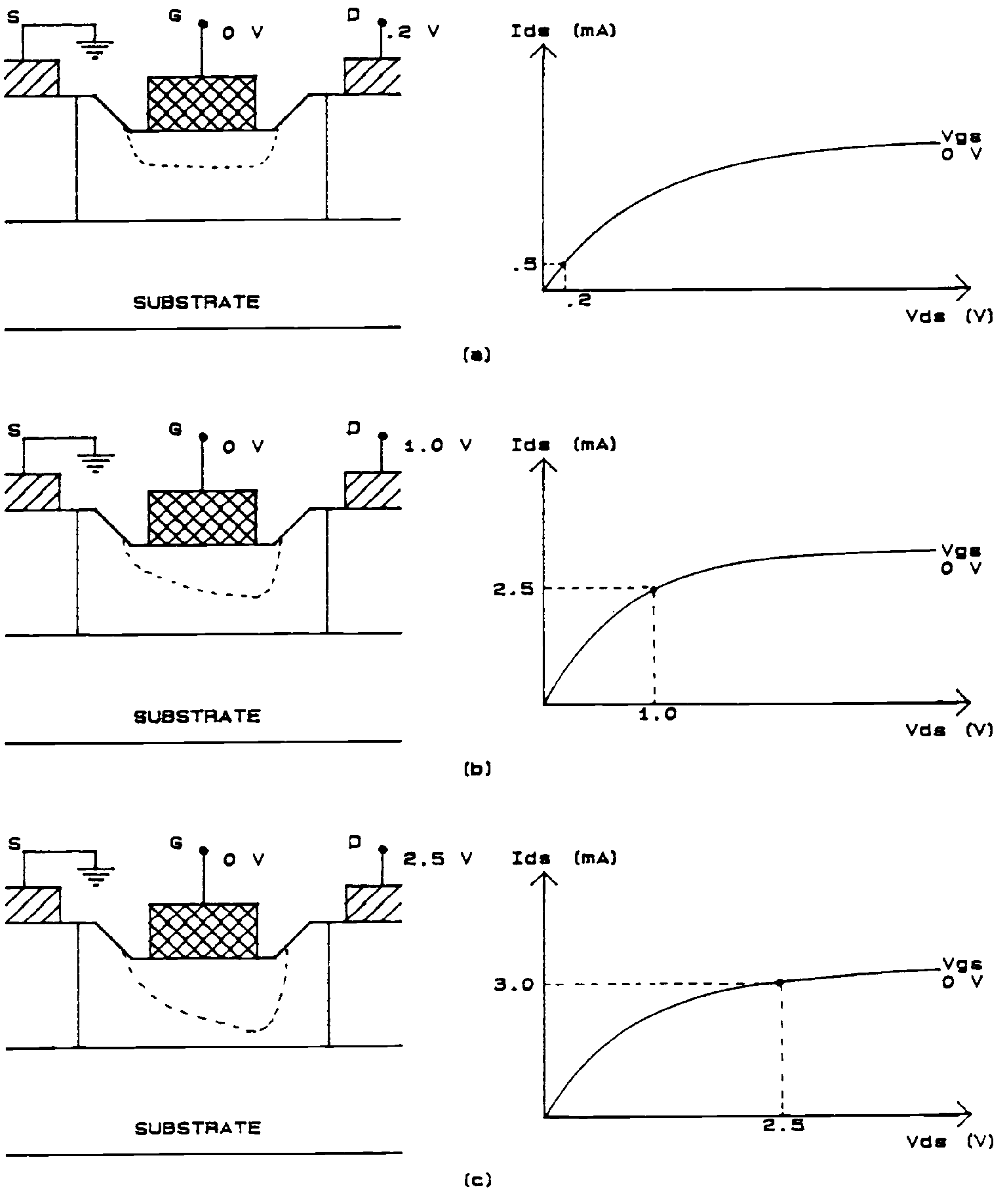

Fig. 9. A GaAs MESFET in (a) linear, (b) edge of saturation, and (c) saturation regions of operation.

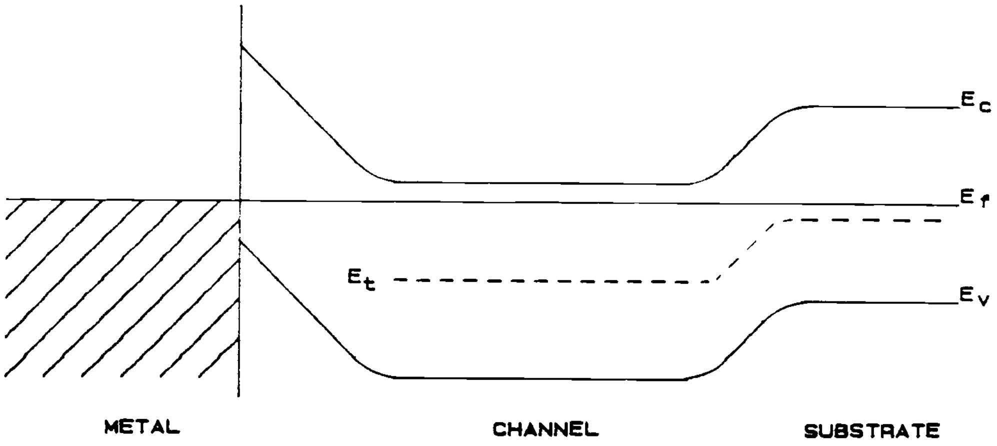

Fig. 10. Energy band diagram of the gate-channel-substrate regions of a n-channel MESFET.

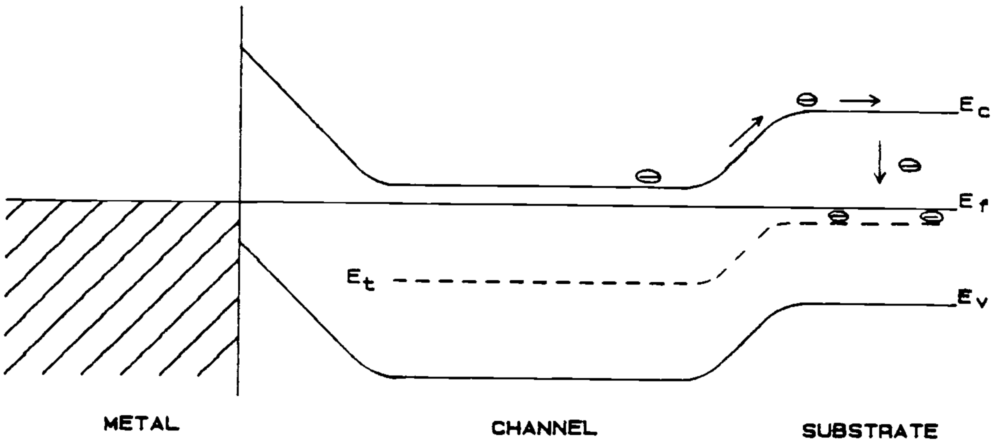

Fig. 11. Energy band diagram showing electrons being scattered and trapped at EL2 centers.

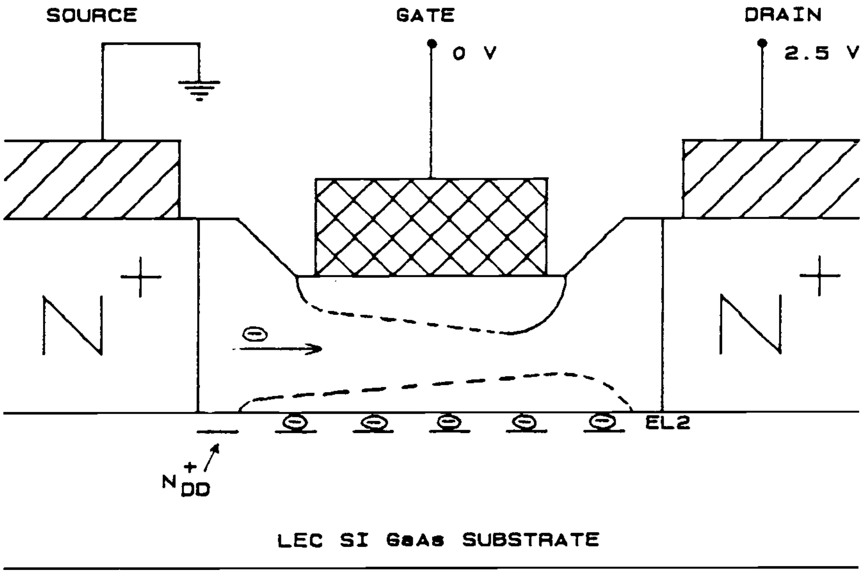

Fig. 12. Self-backgating effect on a GaAs MESFET.

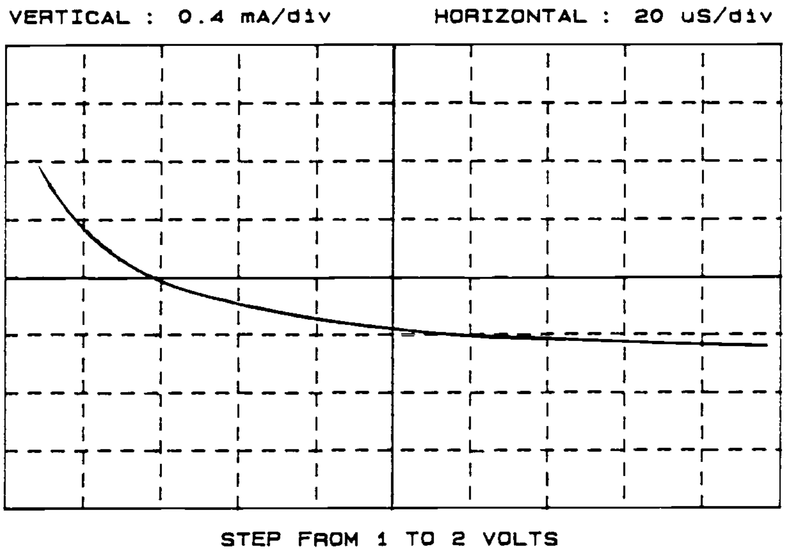

Fig. 13. Drain current transient of a GaAs MESFET after a step from 1 to 2 volts. Idss = 40 mA [8].

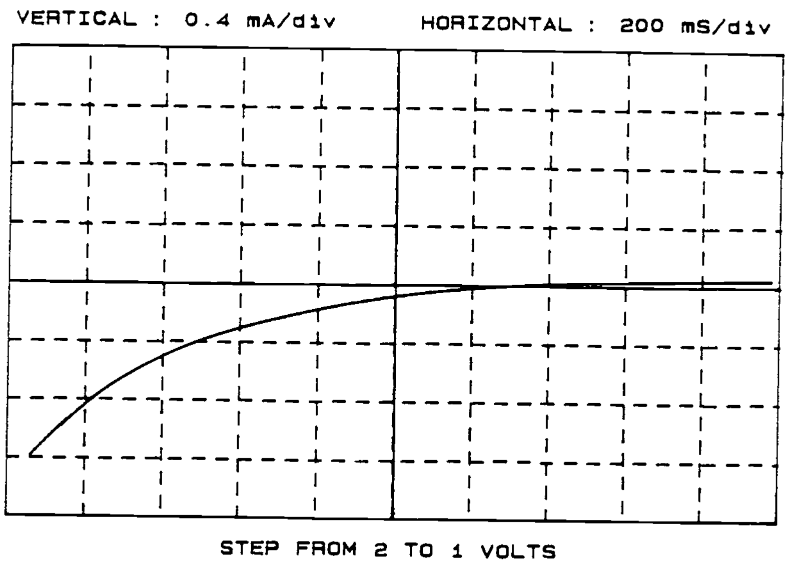

Fig. 14. Drain current transient of a GaAs MESFET after a step from 2 to 1 volts. Idss = 40 mA [8].

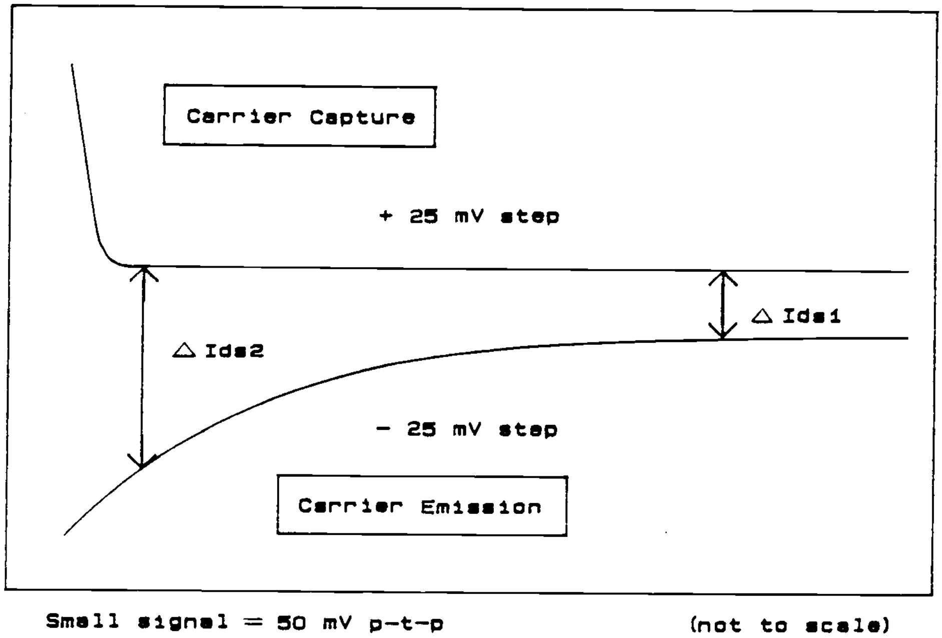

Fig. 14a. Graphical explanation of frequency dependent output conductance.

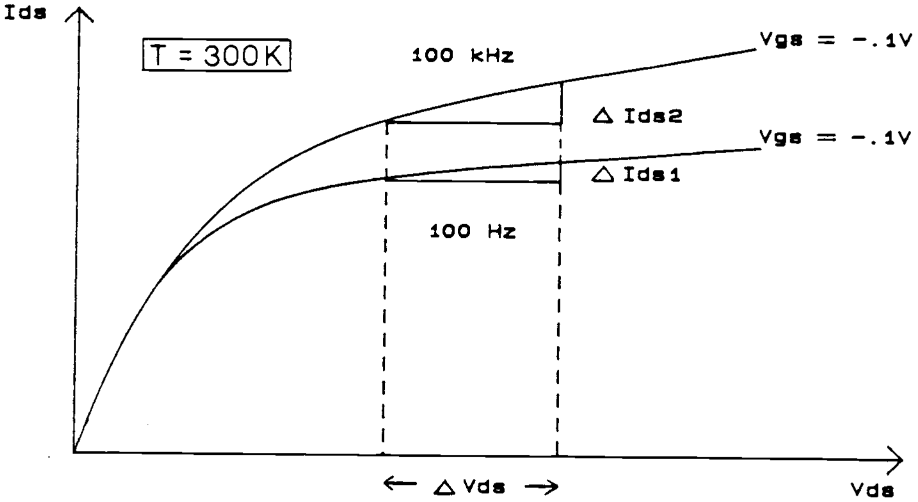

Fig. 15. GaAs MESFET I-V characteristics at .1 and 100 kHz [8].

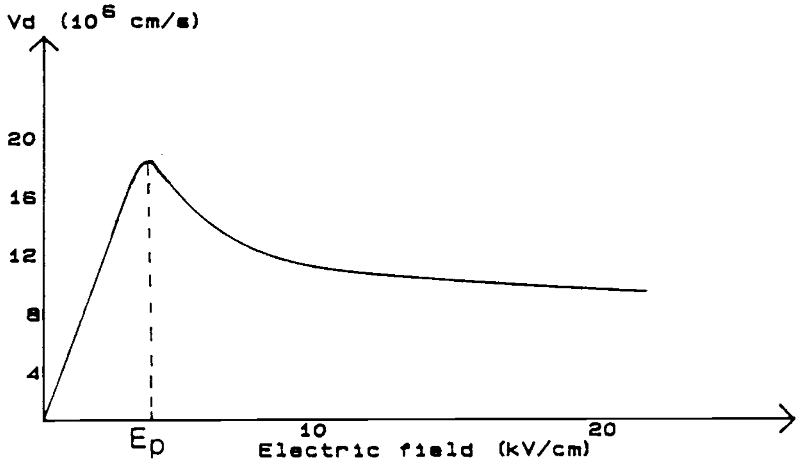

Fig. 16. Drift velocity as a function of electric field in an n-channel GaAs MESFET at 300 K. Nd $\simeq 10^{17}$ cm^{-3}.

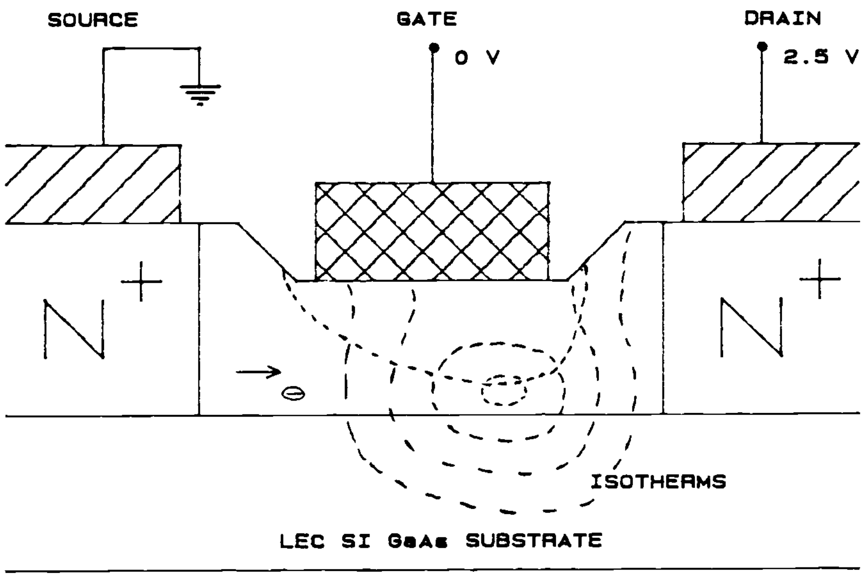

Fig. 17. Cross section of GaAs MESFET indicates the temperature distribution [9].

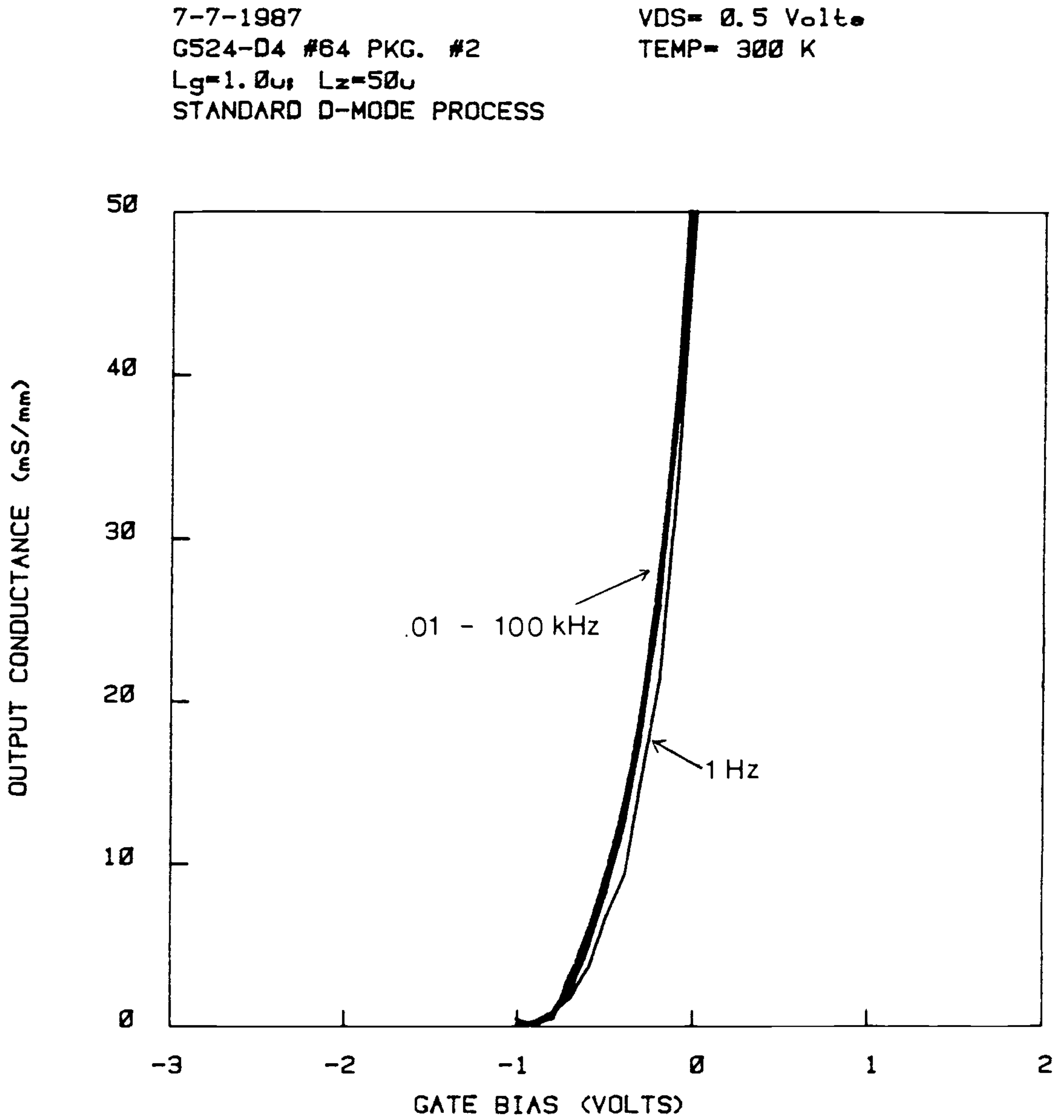

Fig. 18. Measured output conductance vs. gate bias at Vds = 0.5 V. Frequency range is from 1 Hz to 100 kHz. Temperature = 300 K.

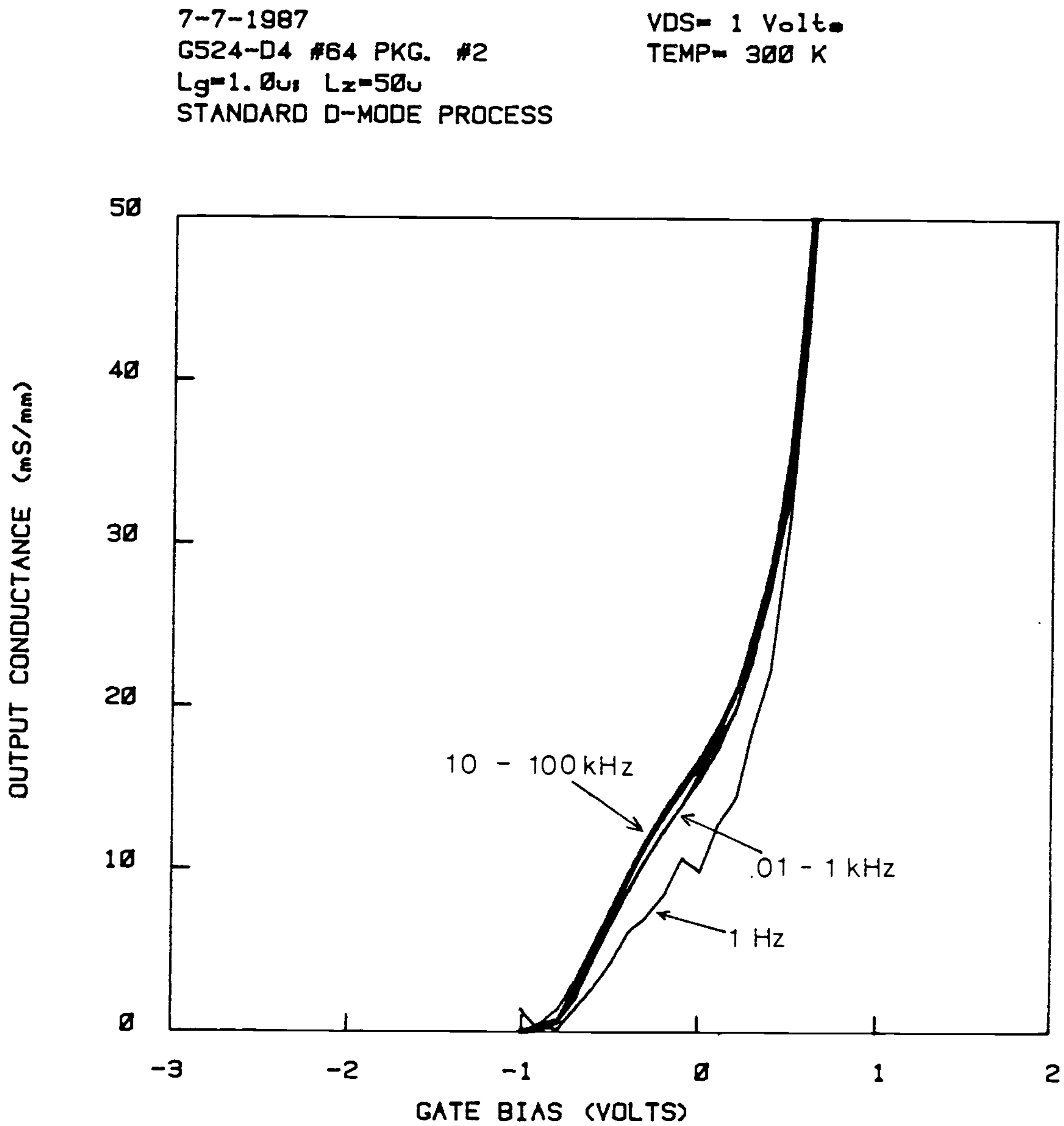

Fig. 19. Measured output conductance vs. gate bias at Vds = 1.0 V.

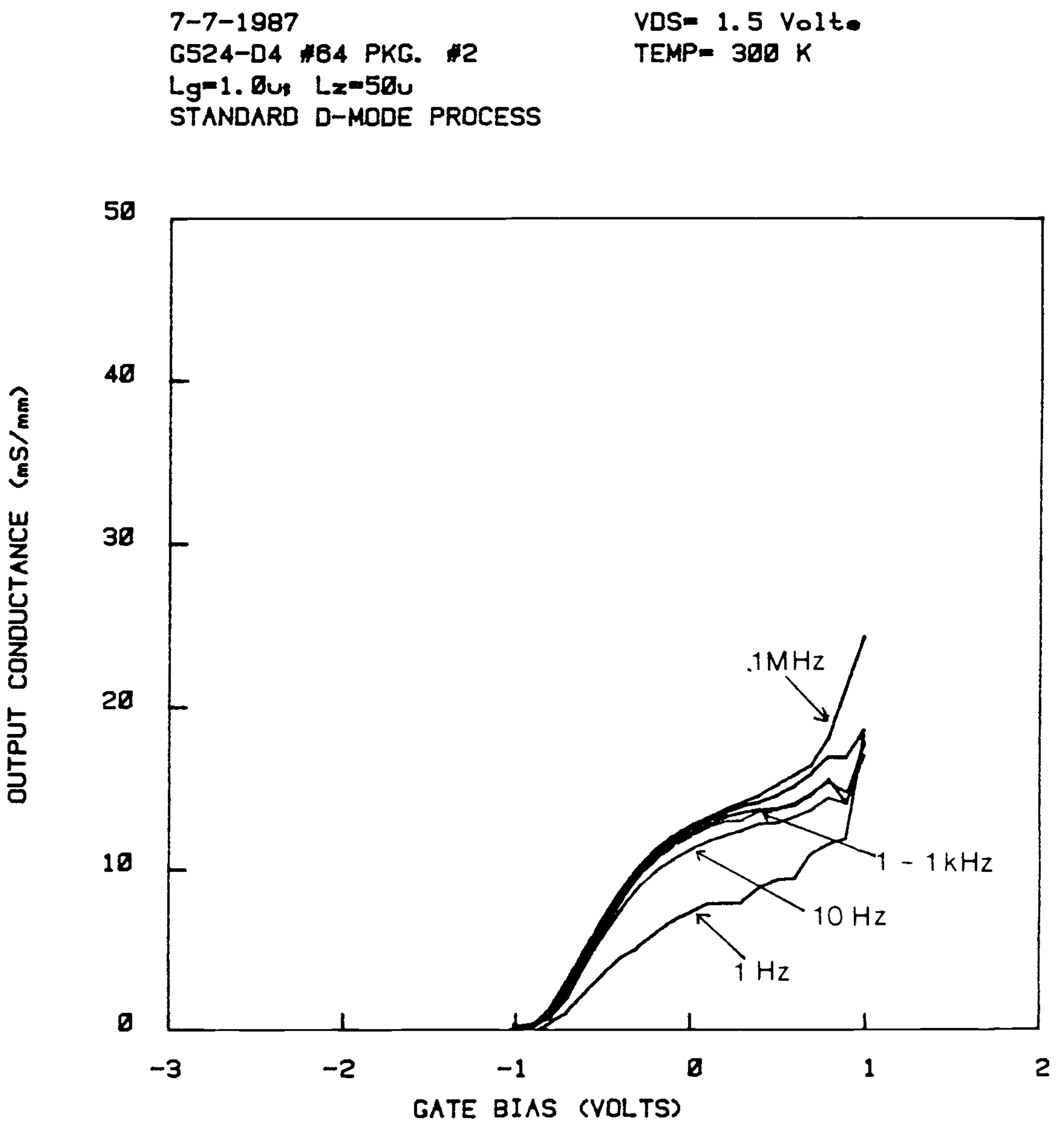

Fig. 20. Measured output conductance vs. gate bias at Vds = 1.5 V.

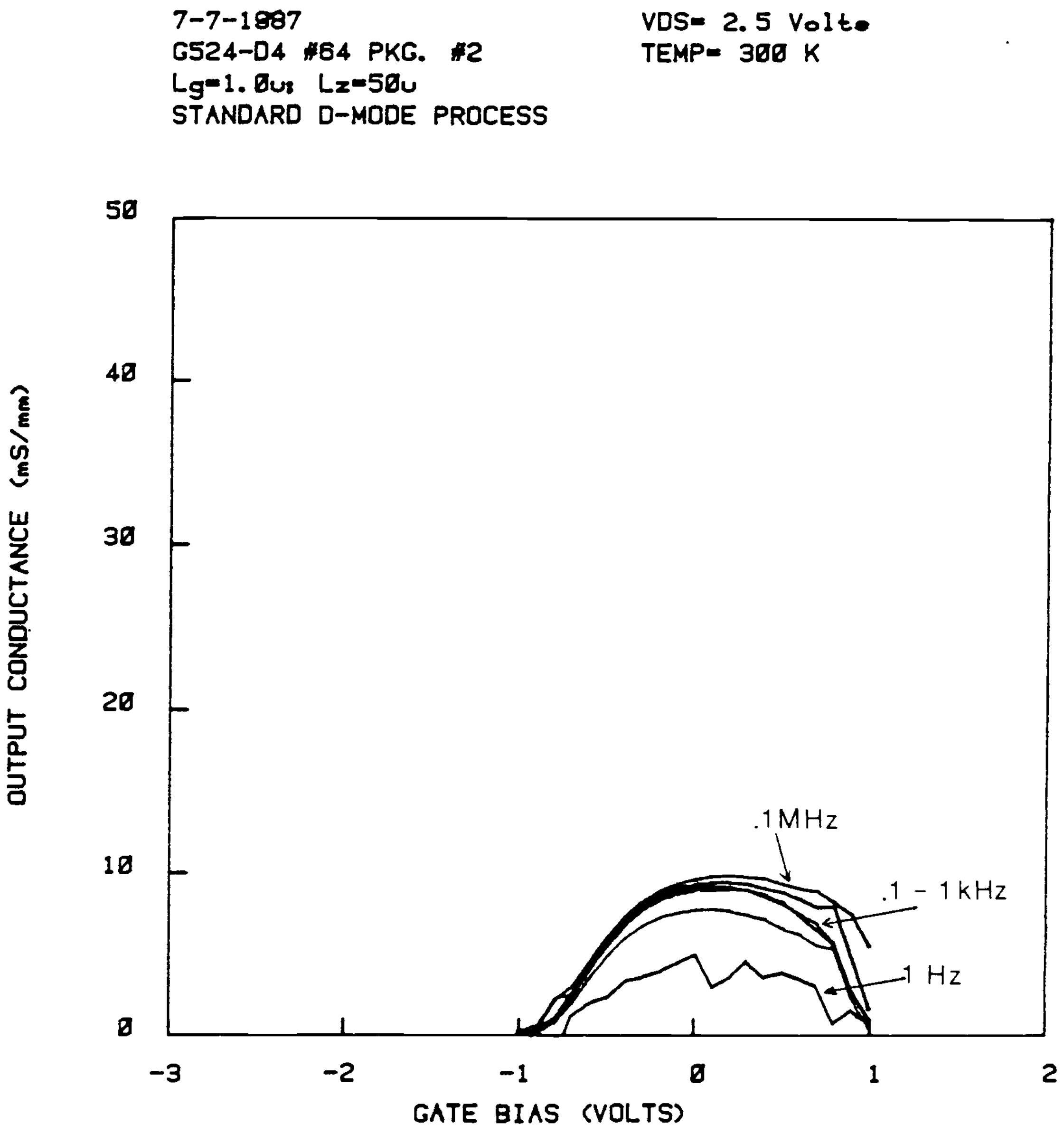

Fig. 21. Measured output conductance vs. gate bias at Vds = 2.5 V.

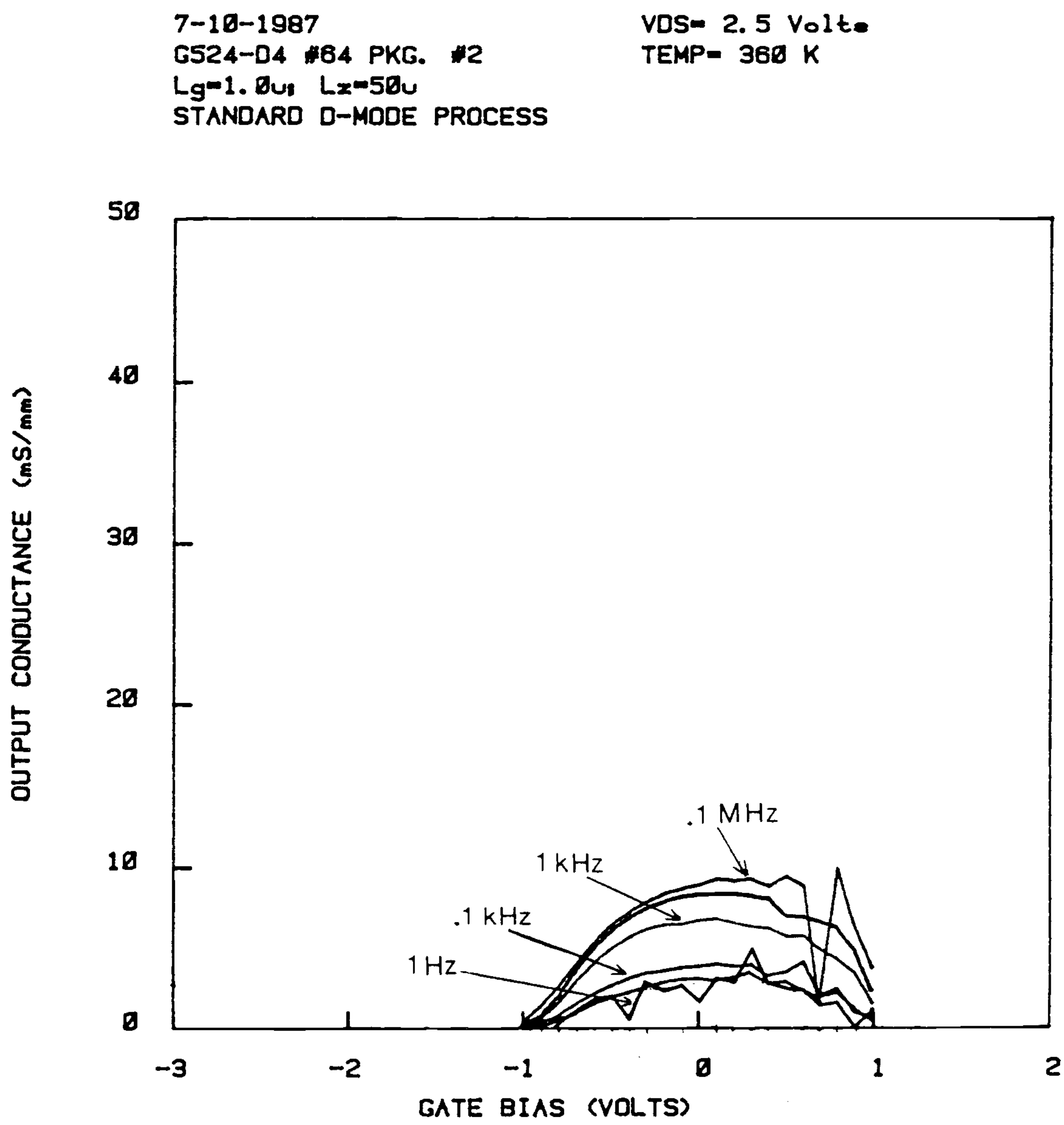

Fig. 22. Measured output conductance vs. gate bias at Vds = 2.5 V. Temperature = 360 K.

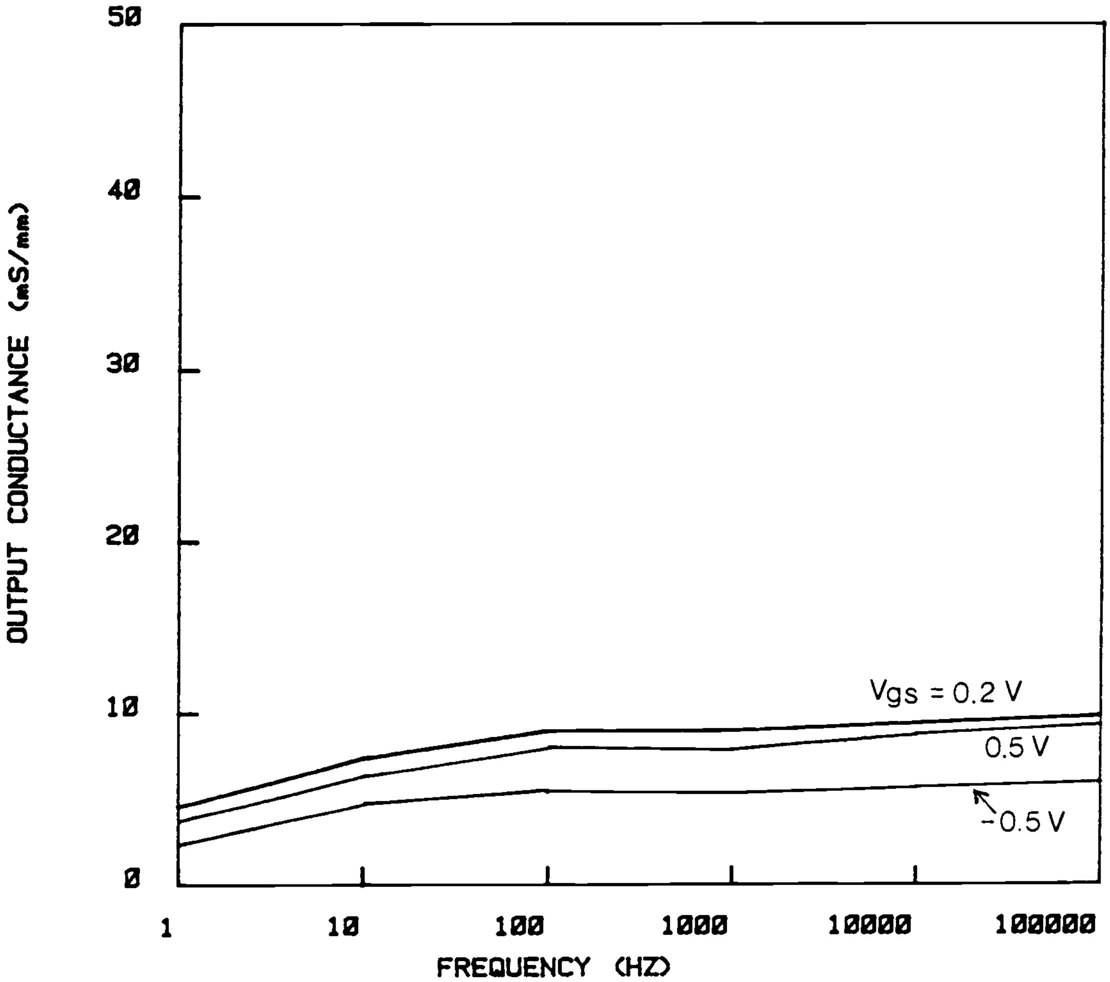

Fig. 23. Measured output conductance vs. frequency at Vds = 2.5 V. Temperature = 300 K.

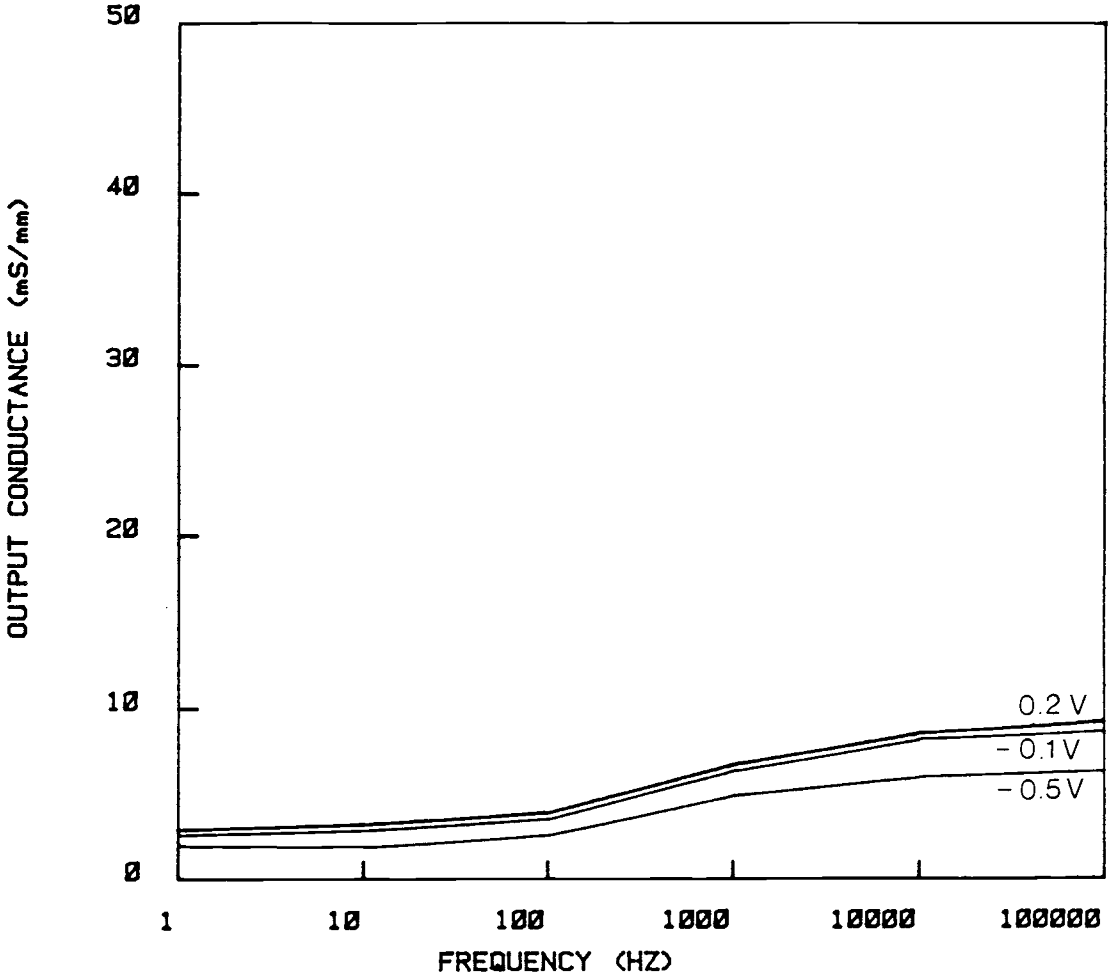

Fig. 24. Measured output conductance vs. frequency at Vds = 2.5 V. Temperature = 360 K.

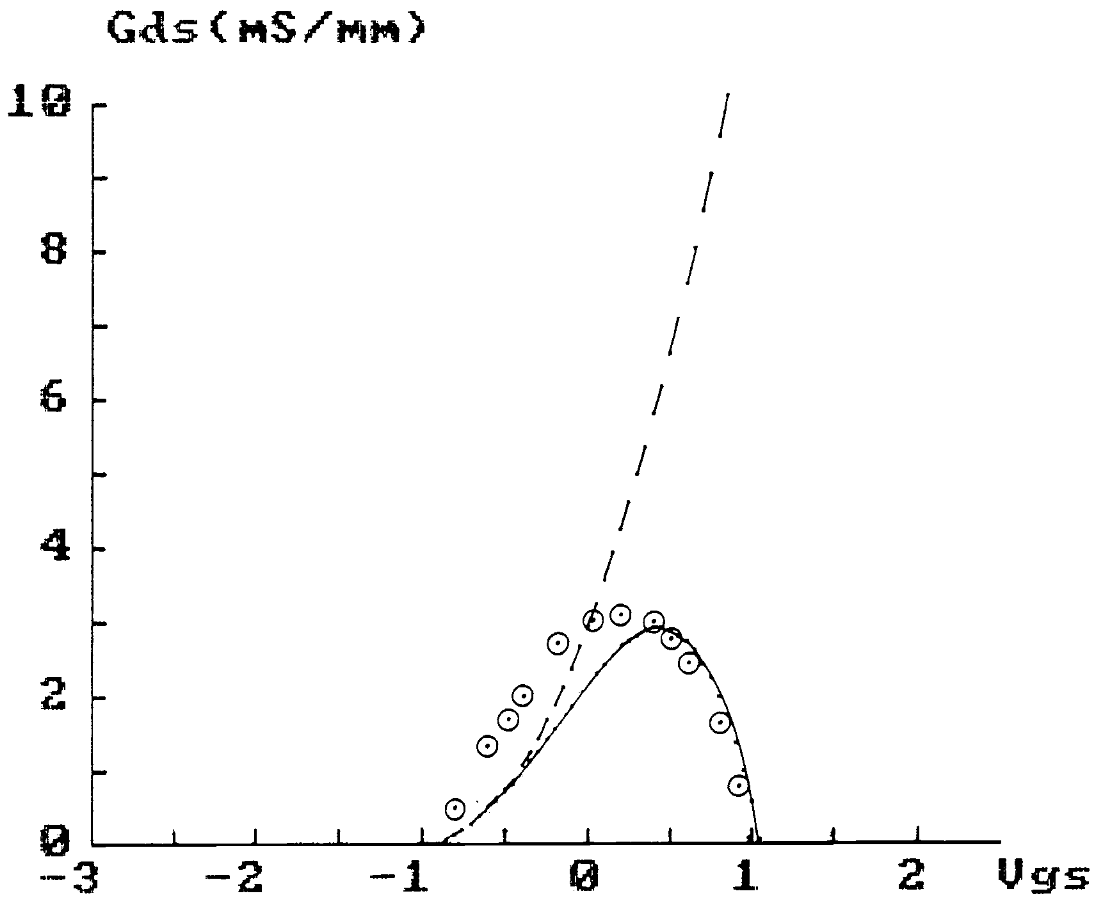

Fig. 25. Output conductance vs. gate bias at Vds = 2.5 V
- - - Curtice model, ——— analytical model,
⊙ ⊙ ⊙ measured data. Temperature = 300 K.

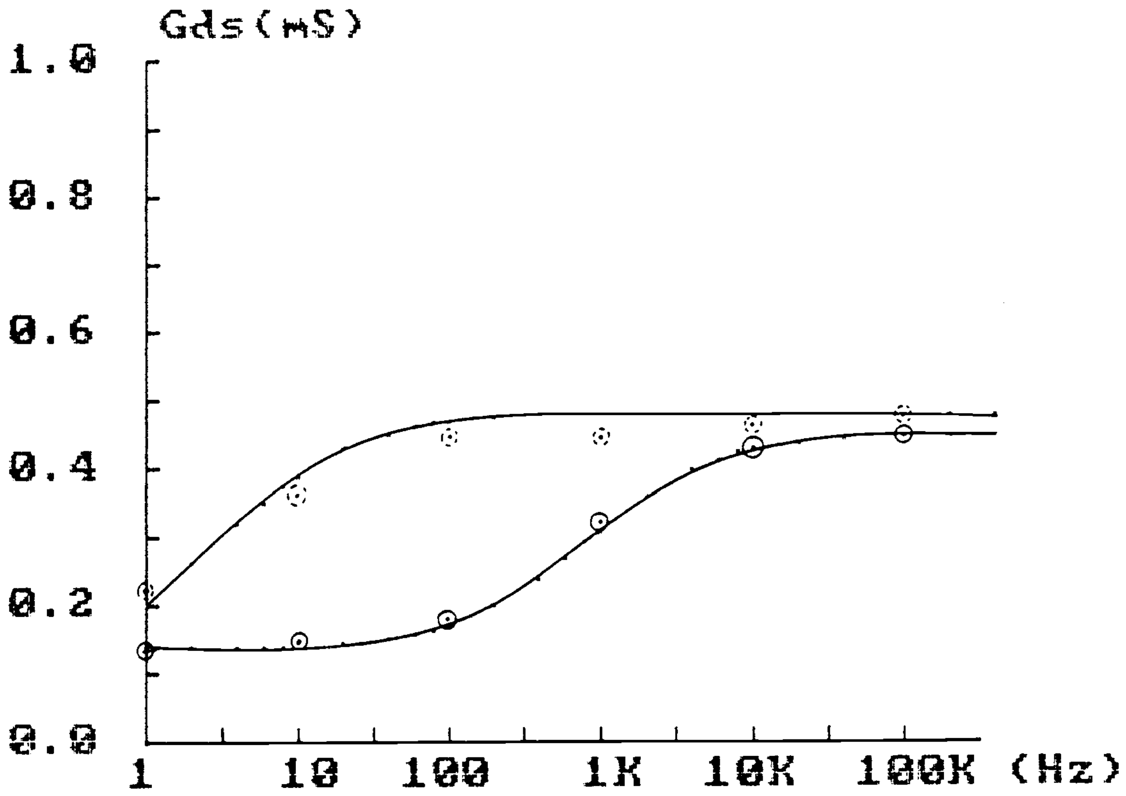

Fig. 26. Output conductance observed in frequency range 1 Hz - 100 kHz. Vds = 2.5 V, Vgs = 0.2 V, ZF = 20, CF = 2.3, and Ids = 4 mA.
Measured data : ⊙ - 300 K, and ⊙ - 360 K.
Analytical model : ——— .

VII. REFERENCES

[1] C. Camacho-Penalosa, and C. S. Aitchison, "Modelling frequency dependent of output impedance of a microwave MESFET at low frequencies," Electron. Lett., vol. 21, no. 12, pp. 528-529, 1985.

[2] P. Canfield, J. Medinger, and L. Forbes, "Buried-channel GaAs MESFET's with frequency-independent output conductance," IEEE Electron Device Lett., vol. EDL-8, no. 3, pp. 88-89, 1987.

[3] W. R. Curtice, "A MESFET model for use in the design of GaAs integrated circuits," IEEE Trans. Microwave Theory Tech., vol. MITT-28, no. 5, pp. 448-456 1980.

[4] M. J. Howes, and D. V. Morgan, Gallium arsenide material, device, and circuits. Wiley, 1985.

[5] J. Lagowski, H. C. Gatos, J. M. Parsey, K. Wada, M. Kaminska, and W. Walukiewicz, "Origin of the 0.82-eV electron trap in GaAs and its annihilation by shallow donors," Applied Physics Lett., vol. 40, no. 4, pp. 342-344, 1982.

[6] M. B. Das, and P. K. Ghosh, "Low frequency emissions from deep levels in GaAs MESFETs," Electron. Lett., vol. 18, pp. 207-208, 1982.

[7] J. F. Wager, and A. J. McCamant, "GaAs MESFET interface considerations," IEEE Trans. Electron Devices, vol. ED-34, no. 5, pp. 1001-1007, 1987.

[8] A. McCamant, TriQuint internal report, 1987.

[9] B. Hughes, N. G. Fernandez, and J. M. Gladstone, "GaAs FET's with a flicker-noise corner below 1 MHz," IEEE Trans. Electron Devices, vol. ED-34, no. 4, pp. 733-741, 1987.

[10] J. S. Blakemore, "Semiconducting and other major properties of gallium arsenide," J. Appl. Phys., vol. 53, no. 10, pp. R123-R181, 1982.

[11] C. Jacoboni, C. Canali, G. Ottaviani, and A. A. Quaranta, "A review of some charge transport properties of silicon," Solid State Electron., vol. 20, pp. 77, 1977.

[12] K. H. Nichols, Camellis M. L. Yee, and C. M. Wolfe, "High-temperature carrier transport in n-type epitaxial GaAs," Solid State Electron., vol. 23, pp. 109-116, 1980.

[13] H. Statz, P. Newman, I. W. Smith, R. A. Pucel, and H. A. Haus, "GaAs FET device and circuit simulation in SPICE," IEEE Trans. Electron Devices, vol. ED-34, no. 2, pp. 160-169, 1987.

APPENDICES

APPENDIX A. List of parameter values

Gate length (Lg)	-	1 um
Gate width (Lz)	-	50 um
Active layer thickness (ALT)	-	0.2 um
Substrate thickness (Tsub)	-	500 um
Threshold voltage (Vt)	-	-1.0 V
Parameter k (beta1)	-	6.2×10^{-7} C/V-m^2
Alpha (alpha)	-	2.2 V^{-1}
Lambda (lambda)	-	0.058 V^{-1}
Mobility at 300 K (Uo)	-	4100 cm^2/V-s
Relative permittivity (Er)	-	12.6
Ambient temperature (Temp0)	-	300 K
Thermal conductivity (k)	-	$56T^{-.87}$ W/cm-°C
Parameter b (b)	-	0.01
Change factor (CF)	-	2.3 (Vds = 2.5 V, Vgs = 0.2 V)
Zero factor (ZF)	-	20 (Vds = 2.5 V, Vgs = 0.2 V)

APPENDIX B. List of simulation program

```
{
This program  simulates the characteristics of a GaAs
MESFET.  User has to enter all the required parameters
in order for this program to run properly. The program
will simulate the following characteristics:
          1. I-V characteristics (dc)
               a. Ids vs. Vds
               b. Ids vs. Vgs
          2. Output conductance characteristics
               a. Gds vs. Vgs
               b. Gds vs. frequency
          3. Transconductance characteristic
               a. Gm vs. Vgs
}

PROGRAM MESFET_SIMULATION;

const
   q  = 1.6E-19;
   Eo = 8.85E-14;
   StepVds = 0.05;  {step size of Vds}
   StepVgs = 0.05;  {step size of Vgs}
   NumStep1 = 10;
   NumVds1 = 100;
   NumVgs1 = 100;
   X1 = 25; Y1 = 10; X2 = 25; Y2 = 170; X3 = 245;
   Y3 = 170;    {coordinates in definning graphic
display}
   Xscale = 10;
   Yscale = 16;
   Pi = 3.14159;

type
   IdsArray = array[1..NumStep1,1..NumVds1] of real;
   Array10 = array[1..10] of real;
   Array2dim = array[1..NumStep1,1..NumVgs1] of real;
   Heading = array[1..10] of char;
   Array1dim = array[1..100] of real;
var
   Vgs, Vds, Vgsmax, VgsStep, Vdsmax, Vt, Beta1, Beta2,
   Alpha, Lambda, Ip, Vp, Uo, U1, UTo, Nd, Lg, Lz, ALT,
   Er, alv, tanh, sech, arcosh, temp1, temp2, temp3,
   temp4, factor, b, Ids, Power, Tem1, Tem2, MobPower,
   lambda1, gamma, Tcon, Tem0, A1, B1, C1, D1, E1, F1,
   G1, J1, K1, M1, N1, U2, V2, U3, V3, Vs, Fs, Vis, Lx,
   dFs, dL1Lx, dVis, dLx, dL1Vds, dL1Vgs, dLxVgs, dU,
   dV, L1, Kd, Vpo, YmaxR, CF, GdsL, ChGds, Ten, ZF,
```

```
    MaxFreq, Freq, FreqStep, NumR, StepR, Tsub, k,
    Esat : real;

    choose, NumStep, NumVds, NumVgs, X2A, Y2A, TempY,
    NumFreq, TempF, select, VdsN, StartVgs, mm, dd, yy,
    Ymax, num, Step, count, TempN : integer;

    ch : char;
    name1,nameA,nameB,nameC : Heading;
    VdsArr,VgsArr,TempArr : Array10;
    IdsArr,GdsArr,GmArr,GdsFArr : Array2dim;

label 1, 2, 11, 12;

Procedure InputData;
var
  I:integer;
  begin
    writeln('Please input current date.');
    write('mm dd yy = ',mm:2,' ',dd:2,' ',yy:2,' ');
    readln(mm,dd,yy);
    writeln('Please enter the value of device size:Gate
    length (Lg),');
    writeln('gate width (Lz),  active layer thickness
    (ALT), and substrate thickness.');
    write('Lg = ',Lg:1,' ');
    readln(Lg);
    write('Lz = ',Lz:1,' ');
    readln(Lz);
    write('ALT = ',ALT:1,' ');
    readln(ALT);
    write('Tsub = ',Tsub:1,' ');
    readln(Tsub);

    writeln('Please enter the value of threshold
    voltage, beta(exclude mobility),');
    write('Vt = ',Vt:1:1,' ');
    readln(Vt);
    write('beta2 = ',beta2:1,' ');
    readln(beta2);
    write('alpha  = ',alpha:1:2,' ');
    readln(alpha);
    write('lambda = ',lambda:1:2,' ');
    readln(lambda);

    writeln('Please enter mobility at substrate temp.');
    write('mobility = ',Uo:1,' ');
    readln(Uo);
    writeln('Please enter the relative permittivity of
    the device.');
    write('Er = ',Er:1:1,' ');
```

```
    readln(Er);
    writeln('Please enter the substrate temperature.');
    write('Temp0 = ',Tem0:3,' ');
    readln(Tem0);

    {Ip :=
    Lz*mobility*sqr(q*Nd)*exp(3*ln(ALT))/(3*Eo*Er*Lg);
    Vp := q*Nd*sqr(ALT)/(2*Eo*Er);
    beta := Ip/sqr(Vp);
    beta1 := 4/3*Lz*Er*Eo/ALT/Lg;}
  end;    {end of InputData}

Procedure ScaleCh1;
  begin
    writeln('Please enter the maximun scale on Y-axis');
    write('Ymax = ',Ymax:3,' ');
    readln(Ymax);
    factor := 160/Ymax/(Lz*10); {adjust factor for
                        different scale in plotting routine}
    step := Ymax div 5; {compute the step for different
                             scale}
  end;    {end of ScaleCh1}

Procedure ScaleCh2;
  begin
    writeln('Please enter the maximum scale on Y-axis');
    write('Ymax = ',Ymax:3,' ');
    readln(Ymax);
    factor := 160/Ymax;
    step := Ymax div 5;
  end;    {end of ScaleCh2}

Procedure ScaleCh3;
  begin
    writeln('Please enter the maximun scale on Y-axis');
    write('Ymax = ',YmaxR:1:1,' ');
    readln(YmaxR);
    factor:= 160/YmaxR;
    StepR:= YmaxR/5;
  end;    {end of ScaleCh3}

Procedure AjMob(Vds1, Vgs1:real);
   begin
    {Beta2:= 4/3*Lz*Uo*Er*Eo/ALT/Lg}
    Ids:= beta2*Uo*sqr(Vgs1-Vt)*(1+lambda*Vds1)
          *tanh/(1+b*(Vgs1-Vt));
    Power:= Ids*Vds1;
    {Tem1:= Power*gamma*ln(4*5E-2/Lg+sqrt(sqr(4*5E-2/
            Lg)-1))/0.46/Pi/Lz;}
    k:= 56*exp(-0.87*ln(Tem0+Tem1));
    Tem1:= Power*ln(2*Tsub/Lg)/Pi/k/Lz;
```

```
      U1:= Uo*exp(MobPower*ln(Tem0))/
           exp(MobPower*ln(Tem1+Tem0));
      {Vs:= (12E6) - gamma*(Tem0+Tem1);}
  end;    {end of AjMob}

Procedure data;          {print data}
  var I:integer;
  begin
    for I:= 1 to 10 do
      writeln(lst,' ');
      write(lst,'Lg = ',Lg:2,'cm , ','Lz = ',Lz:2,
      'cm , ','Vt = ',Vt:1:1,'v , ');
      writeln(lst,'beta2 = ',beta2:2,' , ','alpha =  '
      ,alpha:1:2,' , ');
      write(lst,'Uo = ',Uo:2,'cm**2/v-s , ','Temp0 = '
      ,Tem0:1:1,'K , ');
      writeln(lst,'b = ',b:1:2,' , ','Tsub = ',Tsub:2,
      'cm , ','delta = ',delta:1:2,' , ');
      write(lst,'pole = ',pole:1:1,' , ','Vpo = ' ,
      Vpo:1:2,' , Er = ',Er:1:2,' , ALT = ',ALT:1,
      ' , ');
      writeln(lst,'lambda = ',lambda:1:3,' ,);
      for I:= 1 to 2 do
        writeln(lst,' ');
      for I:= 1 to 60 do
        write(lst,' ');
      writeln(lst,mm:2,'/',dd:2,'/',yy:2);
  end;    {end of data}

Procedure beta(Vds1,Vgs1:real);
  begin
    Fs:= Vs/U1;
    Vis:= Fs*Lg*(Vgs1-Vt)/(Fs*Lg+Vgs1-Vt);
    Lx:= Pi*Kd*(Vds1-Vis)/2/ALT/Fs;
    L1:= Lg - 2*ALT/Pi*ln(Lx + sqrt(sqr(Lx)+1));
    beta1:= 2*Er*Eo*Vs*Lz/ALT/(Vpo+3*Fs*L1);
  end;    {end of beta}

Procedure Getdata;
  var j :integer;
  begin
    writeln('How many different drain bias are
    interested ?');
    write('Vds# = ',VdsN:1,' ');
    readln(VdsN);
    for j:= 1 to VdsN do
      begin
        write('Vds[',j:1,'] = ',VdsArr[j]:1:1,' ');
        readln(VdsArr[j]);
      end;
    writeln('Please enter the value of "b"');
```

```
    write('b = ',b:1:2,' ');
    readln(b);
    writeln('Please enter the power of mobility');
    write('MobPower = ',MobPower:1:1,' ');
    readln(MobPower);

    writeln('Please enter values of gamma, Kd, and pinch
    off voltage.');
    write('gamma = ',gamma:1,' ');
    readln(gamma);
    write('Kd = ',Kd:1:1,' ');
    readln(Kd);
    write('Vpo = ',Vpo:1:2,' ');
    readln(Vpo);

    writeln(' ');
    writeln('"P" --- plot using new data');
    writeln('"R" --- replot using old data');
    writeln('"M" --- return to main menu');
    writeln('"C" --- go back to change the bias');
  end;    {end of Getdata}

Procedure Graph(var name:heading;var Arr:Array2dim);
  var j, loop1, loop2:integer;
  label 4,12,14;
  begin
12: GraphColorMode;
    GraphWindow(35,10,285,199);  {define boundary}
    Palette(0);
    Draw(X1,Y1,X2,Y2,1); {draw Y-axis}
    Draw(X2,Y2,X3,Y3,1); {draw X-axis}
    X2A:=X2;
    Y2A:=Y2;
    for j:= 1 to Xscale do    {draw division on axes}
      begin
        X2A:=X2A + 2*Xscale;
        Draw(X2A,Y2,X2A,Y2-3,1);
        Y2A:=Y2A - Yscale;
        Draw(X2,Y2A,X2+3,Y2A,1);
      end;
    write('Vds= ',VdsArr[1]:1:1);  {display Vds biases}
    for j:= 2 to VdsN do
      write(', ',VdsArr[j]:1:1);
    writeln(' V');
    writeln('        ',name:10);   {name the Y-axis}
    window(1,10,40,25);
    writeln(' ');
    writeln('   ',Ymax:4);          {print the desired}
    num:=Ymax;                     {scale on Y-axis}
    for j:= 1 to 5 do
```

```
      begin
        writeln(' ');
        writeln(' ');
        writeln(' ');
        num:=num-step;
        writeln('   ',num:4);
    end;
    writeln('      -3   -2   -1    0    1    2  Vgs');
    StartVgs:= 200 - NumVgs*2;
    {plot data}
    for loop1:= 1 to VdsN do
      begin
        X2A:= X2 + StartVgs-2;
        for loop2:= 1 to NumVgs+1 do
          begin
            TempY:= round(factor*Arr[loop1,loop2]);
            X2A:= X2A + 2;
            if TempY < 0 then
              TempY:=0;
            Plot(X2A,Y2-TempY,2);
          end;
      end;
    readln(ch);
    case ch of
      'a' : goto 4;
    end;
4:  clrscr;
    write('Do you want to plot in different scale
    (Y/N) ? ');
    readln(ch);
    case ch of
      'Y','y' : begin
                  case select of
                    1 :begin
                         ScaleCh2;
                         goto 12;
                       end;
                   2,3:begin
                         ScaleCh1;
                         goto 12;
                       end;
                   else goto 4;
                  end;
                end;
      'N','n' : goto 14;
      else goto 4;
    end;
14:end;   {end fo Graph}
```

```
Procedure IVcurve;
  var
    Ids:IdsArray;
    loop1,loop2,I:integer;
  label 1,2,3,4,5,8,9,10,11,13;
  begin
  clrscr;
  writeln('                         1 ----------- Ids vs.
  Vds');
  writeln('                         2 ----------- Ids vs.
  Vgs');
1:readln(ch);
  case ch of
    '1' : goto 9;
    '2' : goto 2;
    else  goto 1;
  end;
  9:ScaleCh2;
    writeln('Please enter the maximun value of the
    drain-source bias.');
    write('Vds(max) = ',Vdsmax:1:1,' ');
    readln(Vdsmax);
    writeln('Please enter # of steps generated.');
    write('NumStep = ',NumStep:1,' ');
    readln(NumStep);   {number of Vgs steps}
    for I:= 1 to NumStep do
      begin
        write('Vgs[',I:1,'] = ',VgsArr[I]:1:1,' ');
        readln(VgsArr[I]);
      end;
    writeln('Please enter the value of "b"');
    write('b = ',b:1:2,' ');
    readln(b);
    writeln('Please enter the power of mobility');
    write('MobPower = ',MobPower:1:1,' ');
    readln(MobPower);

    write('Gamma = ',Gamma:1,' ');
    readln(gamma);
    write('Kd = ',Kd:1:1,' ');
    readln(Kd);
    write('Vpo = ',Vpo:1:2,' ');
    readln(Vpo);

    writeln(' ');
    writeln('"P" --- plot using new data');
    writeln('"R" --- replot using old data');
    writeln('"M" --- return to main menu');
    writeln('"C" --- go back to change the bias');
 10:readln(ch);
    case ch of
```

```
        'p' : begin
                clrscr;
                goto 8;
              end;
        'r' : begin
                clrscr;
                goto 11;
              end;
        'm' : begin
                clrscr;
                goto 13;
              end;
        'c' : begin
                clrscr;
                goto 9;
              end;
        else goto 10;
      end;

8:  NumVds:=round(Vdsmax/StepVds) + 1;  {compute # of
    points for Vds}

    for loop1 := 1 to NumStep do begin
      Vds :=0.0;
      count:=0;
      tem1:=0;
      for loop2 := 1 to NumVds do begin
        alv := alpha*Vds;
        tanh := (exp(alv)-exp(-alv))/(exp(alv)+
                  exp(-alv));
        temp1 := 1+lambda*Vds;
        temp2 := 1+b*(VgsArr[loop1]-Vt);
        AjMob(Vds,VgsArr[loop1]);
        {beta(Vds,VgsArr[loop1]);}
        Ids[loop1,loop2] :=1000*(beta2*U1*
               sqr(VgsArr[loop1]-Vt)*temp1*tanh)/temp2;
        Vds := Vds + StepVds;  {increment of Vds}
        write(Ids[loop1,loop2]:2,' ');
        write(U1:2,' ',k:1:2,' ',tem1:1:2,' ');
      end;
    end;
    readln(ch);
11: GraphColorMode;
    GraphWindow(35,10,285,199);  {define the graphic
                                   window}
    Palette(0);   {choose different color}
    Draw(X1,Y1,X2,Y2,1);  {draw the x-axis}
    Draw(X2,Y2,X3,Y3,1);  {draw the y-axis}
    X2A:=X2;
    Y2A:=Y2;
    for I := 1 to Xscale do
```

```
    begin
      X2A:=X2A + 2*Xscale;
      Draw(X2A,Y2,X2A,Y2-3,1);  {draw the X partition}
      Y2A:=Y2A - Yscale;
      Draw(X2,Y2A,X2+3,Y2A,1);  {draw the Y partition}
    end;
  write('Vgs = ',VgsArr[1]:1:1);  {write the value for
                                    different Vgs}
  for I:= 2 to NumStep do
  write(', ',VgsArr[I]:1:1);
  writeln(' V');
  window(1,10,40,25);
  writeln('        Ids(mA)');
  writeln('   ',Ymax:4);
  num:=Ymax;
  for I:= 1 to 5 do   {write the # on Y-axis}
    begin
      writeln(' ');
      writeln(' ');
      writeln(' ');
      num:=num - step;
      writeln('   ',num:4);
    end;
  writeln('       0    1    2    3    4    5  Vds');
  for loop1:= 1 to NumStep do
    begin
      X2A:= X2 - 2;
      for loop2:= 1 to NumVds do
        begin
          TempY:= round(factor*Ids[loop1,loop2]);
          X2A:= X2A + 2;
          Plot(X2A,Y2-TempY,2);    {plot Ids}
        end;
    end ;

  readln(ch);
  case ch of
   'a' : goto 3;
    else goto 3;
  end;
3: clrscr;
  write('Do you want to plot in different scale (Y/N)
   ');
  readln(ch);
  case ch of
   'Y','y' : begin
               ScaleCh2;
               goto 11;
             end;
   'N','n' : goto 13;
    else goto 3;
```

```
    end;
2:  ScaleCh2;
    Getdata;
4:  readln(ch);
    case ch of
     'p' : begin
             clrscr;
             goto 5;
           end;
     'r' : begin
             clrscr;
             Graph(name1,IdsArr);
             goto 13;
           end;
     'm' : begin
             clrscr;
             goto 13;
           end;
     'c' : begin
             clrscr;
             ScaleCh2;
             Getdata;
           end;
     else  goto 4;
    end;
 5: NumVgs:=round((2-Vt)/StepVgs);
    tem1:=0;
    for loop1:= 1 to VdsN do
      begin
        Vgs:= 2-(NumVgs * 0.05);
        for loop2:= 1 to NumVgs+1 do
          begin
            alv:= alpha*VdsArr[loop1];
            tanh:= (exp(alv)-exp(-alv))/(exp(alv)+
                    exp(-alv));
            temp1:= 1+lambda*VdsArr[loop1];
            temp2:= 1+b*(Vgs-Vt);
            AjMob(VdsArr[loop1],Vgs);
            {beta(VdsArr[loop1],Vgs);}
            IdsArr[loop1,loop2]:= 1000*(beta2*sqr(Vgs-
              Vt)*temp1*tanh)/temp2;
            Vgs:= Vgs+StepVgs;
            write(IdsArr[loop1,loop2]:2,' ');
            write(U1:2,' ');
          end;
      end;
      writeln(' ');
      readln(ch);    {pause after the data is displayed}
      select:= 1;
      name1:= 'Ids(mA)    ';
      Graph(name1,IdsArr);    {call the graphic routine}
```

```
13:end;   {end of IVcurve}

Procedure OutCond;
var
  j, loop1, loop2:integer;
  label  5, 7, 14;
  begin
    ScaleCh1;
    Getdata;
7:  readln(ch);
    case ch of
      'p' : begin
              clrscr;
              goto 5;
            end;
      'r' : begin
              clrscr;
              Graph(name1,GdsArr);
              goto 14;
            end;
      'm' : begin
              clrscr;
              goto 14;
            end;
      'c' : begin
              clrscr;
              ScaleCh1;
              Getdata;
            end;

       else goto 7;
     end;

5:  NumVgs:=round((2-Vt)/StepVgs);  {compute # of points
                                     for Vgs}
    for loop1:= 1 to VdsN do
      begin
        Vgs:= 2-(NumVgs * 0.05);     {start plotting at
                                      threshold voltage}
        tem1:=0;
        for loop2:= 1 to NumVgs+1 do
          begin
            alv:=alpha*VdsArr[loop1];
            sech:=2/(exp(alv)+exp(-alv));
            tanh:=(exp(alv)-exp(-alv))/(exp(alv)+
                   exp(-alv));
            temp3:=(1+lambda*VdsArr[loop1])*
                    alpha*sqr(sech);
            temp4:=1+b*(Vgs-Vt);
            {arcosh:= ln(5E-2/Lg+sqrt(sqr(5E-2/Lg)-1));}
            arcosh:= ln(2*Tsub/Lg);
```

```
              AjMob(VdsArr[loop1],Vgs);
              A1  := Tem0*(1+b*(Vgs-Vt));
              B1  := Uo*Beta2*sqr(Vgs-Vt)*arcosh/k/Pi/Lz;
              C1  := Tem0*Uo*beta2*sqr(Vgs-Vt);
              U1  := C1*(1+lambda*VdsArr[loop1])*tanh;
              V1  := A1+B1*(VdsArr[loop1]+lambda*
                     sqr(VdsArr[loop1]))*tanh;
              dU1:= C1*(temp3+lambda*tanh);
              dV1:= B1*((1+2*lambda*VdsArr[loop1])*
                     tanh+VdsArr[loop1]*temp3);
              GdsArr[loop1,loop2]:=1000*(V1*dU1 - U1*dV1)/
                                                    sqr(V2);
              Vgs:=Vgs+StepVgs;
              write(GdsArr[loop1,loop2]:2,' ',k:1:2,' ');
            end;
    end;
    writeln(' ');
    readln(ch);     {pause after data is displayed}
    select:= 2;
    name1:= 'Gds(mS/mm)';
    graph(name1,GdsArr);
14:end;   {end of OutCond}

Procedure Transcon;
  label 5, 7, 14;
  var loop1,loop2:integer;
  begin
    ScaleCh1;
    Getdata;
7:  readln(ch);
    case ch of
      'p' : begin
              clrscr;
              goto 5;
            end;
      'r' : begin
              clrscr;
              Graph(name1,GmArr);
              goto 14;
            end;
      'm' : begin
              clrscr;
              goto 14;
            end;
      'c' : begin
              clrscr;
              Getdata;
            end;
     else goto 7;
    end;
5:  NumVgs:=round((2-Vt)/StepVgs);
```

```
    for loop1:= 1 to VdsN do
      begin
        Vgs:= 2-(NumVgs * 0.05);
        for loop2:= 1 to NumVgs+1 do
          begin
            alv:= alpha*VdsArr[loop1];
            tanh:=(exp(alv)-exp(-alv))/(exp(alv)+
                   exp(-alv));
            arcosh:= ln(2*Tsub/Lg);
            AjMob(VdsArr[loop1],Vgs);
            A1 := Tem0*(1+lambda*VdsArr[loop1];
            B1 := VdsArr[loop1]*Uo*Beta2*arcosh*tanh/
                 k/Pi/Lz;
            C1 := Tem0*Beta2*tanh;
            U1 := C1*sqr(Vgs-Vt);
            V1 := A1*(1+b*(Vgs-Vt))+B1*sqr(Vgs-Vt);
            dU1:= 2*C1*(Vgs-Vt);
            dV1:= A1*b+2*B1*(Vgs-Vt);
            GmArr[loop1,loop2]:=1000*(V1*dU1-U1*dV1)/
                                    sqr(V1);
            Vgs:= Vgs+StepVgs;
            write(GmArr[loop1,loop2]:2,' ');
          end;
      end;
    writeln(' ');
    readln(ch);      {pause after the data is displayed}
    select:= 3;
    name1:= 'Gm(mS/mm) ';
    graph(name1,GmArr);
14:end;   {end of Trancon}

Procedure GdsFreq;
  Var I, J, count:integer;
  label 1,2,3,4,5;
  begin
    ScaleCh3;
    {writeln('Please enter the low frequency Output
    Conductance.');
    write('Gds[FreqL] = ',GdsL:1,' ');
    readln(GdsL);
    writeln('Please enter the change factor of Gds.');
    write('Change factor = ',CF:1,' ');
    readln(CF);
    writeln('Please enter the time constant of trap.');
    write('Tcon = ',Tcon:1,' ');
    readln(Tcon);}
    writeln('How many different temperature bias ?');
    write('# of Temp = ',TempN:1,' ');
    readln(TempN);
    for j:= 1 to TempN do
```

```
  begin
    write('Temp[',j:1,'] = ',TempArr[j]:1:1,' ');
    readln(TempArr[j]);
  end;
write('b= ',b:1:1,' ');
readln(b);
writeln('please enter the drain and the gate
bias.');
write('Vds = ',Vds:1:1,' ');
readln(Vds);
write('Vgs = ',Vgs:1:1,' ');
readln(Vgs);
write('MobPower = ',MobPower:1:1,' ');
readln(MobPower);
write('Change factor = ',CF:1:1,' ');
readln(CF);
write('Templ = ',Tem1:1,' ');
readln(Tem1);
writeln('Please enter the pole factor.');
write('pole factor = ',ZF:2,' ');
readln(ZF);
writeln('Please enter the maximun frequency.');
write('maxFreq = ',MaxFreq:1,' ');
readln(maxFreq);
NumFreq:=round(5*(ln(maxFreq)/ln(10)));  {generate #
                                           of points}
{Freq:= 10;
FreqStep:= 20;
GdsFArr[1]:= ChGds/(1+ZF/Freq/Tcap) + GdsL;
write(GdsFArr[1]:2:3,' ');}
for I:= 1 to TempN do
begin
  UTo:=300*Uo/TempArr[i];{compute the right mobility
                          at the specific temperature}
  Tem0:= TempArr[i];
  alv:=alpha*Vds;
  sech:=2/(exp(alv)+exp(-alv));
  tanh:=(exp(alv)-exp(-alv))/(exp(alv)+exp(-alv));
  {lambda:=Tcap*ln(Freq)/ln(10)/100;}
  {temp1:= ZF*Tcap*Freq;}
  {lambda:= 0.05*(1+(exp(temp1)-exp(-temp1))/
            (exp(temp1)+exp(-temp1)));}
  {lambda:= 0.05*exp(Freq*Ten);}
  temp3:=(1+lambda*Vds)*alpha*sqr(sech);
  temp4:=1+b*(Vgs-Vt);
  arcosh:= ln(2*Tsub/Lg);
  AjMob(Vds,Vgs);
  {arcosh:= ln(5E-2/Lg+sqrt(sqr(5E-2/Lg)-1));}
  A1:= TempArr[i]*(1+b*(Vgs-Vt));
  B1:= UTo*Beta2*sqr(Vgs-Vt)*arcosh/k/Pi/Lz;
  C1:= TempArr[i]*UTo*beta2*sqr(Vgs-Vt);
```

```
U1:= C1*(1+lambda*Vds)*tanh;
V1:= A1+B1*(Vds+lambda*sqr(Vds))*tanh;
dU1:= C1*(temp3+lambda*tanh);
dV1:= B1*((1+2*lambda*Vds)*tanh+Vds*temp3);
GdsL:=(V1*dU1 - U1*dV1)/sqr(V1);
GdsFArr[I,1]:= 0.0;
Freq:= 2.0E0; {start freq. measurement at 2 Hz}
FreqStep:=2.0E0;  {initial freq. step = 2 Hz}
count:=0;       {use count to change freq. step}
writeln(Tem0:1);
{AjMob(Vds,Vgs);}
writeln('GdsL = ',GdsL:1,' ',Tem1:1:1,' ');
for J:= 2 to NumFreq+1 do
  begin
    {k:= 0.37;}
    Ten:= 3.53E-8/sqr(TempArr[I]+Tem1)*
          exp(9450/(TempArr[I]+Tem1));
    GdsFArr[I,J]:=1000*
              (CF*GdsL/(1+ZF/Freq/Ten)+GdsL);
    Freq:= Freq + FreqStep;
    write(Ten:1,' ',GdsFArr[I,J]:2:3,' ',Freq:1,
    ' ',FreqStep:1,' ');
    count:= count + 1;
    if Freq > 1.1E1 then
      begin
        FreqStep:= 2.0E1;
      end;
    if Freq > 1.1E2 then
      begin
        FreqStep:= 2.0E2;
      end;
    if Freq > 1.1E3 then
      begin
        FreqStep:= 2.0E3;
      end;
    if Freq > 1.1E4 then
      begin
        FreqStep:= 2.0E4;
      end;
    if Freq > 1.1E5 then
      begin
        FreqStep:= 2.0E5;
      end;
    case count of
       5 : Freq:= 2E1;
      10 : Freq:= 2E2;
      15 : Freq:= 2E3;
      20 : Freq:= 2E4;
      25 : Freq:= 2E5;
      else  goto 5;
    end;
```

```
5:      end;     { do loop J }
    end;         {do loop I}
    writeln(' ');
    write('Tem1 = ',Tem1:1:1,' ','GdsL = ',GdsL:1);
    read(ch);  {pause after the data are displayed}
2:  GraphColorMode;
    GraphWindow(35,10,285,199);
    Palette(0);
    Draw(X1,Y1,X2,Y2,1);
    Draw(X2,Y2,X3,Y3,1);
    X2A:=X2;
    Y2A:=Y2;
    for I:= 1 to XScale do
      begin
        X2A:=X2A + 2*Xscale;
        Draw(X2A,Y2,X2A,Y2-3,1);
        Y2A:=Y2A - Yscale;
        Draw(X2,Y2A,X2+3,Y2A,1);
      end;
    writeln(' ');
    window(1,10,40,25);
    writeln('        Gds(mS)');
    writeln('    ',YmaxR:1:1);
    numR:=YmaxR;
    for I:= 1 to 5 do
      begin
        writeln(' ');
        writeln(' ');
        writeln(' ');
        numR:= numR - StepR;
        writeln('    ',NumR:1:1);
      end;
    writeln('       1    10   100   1K   10K   100K   Hz');
    for I:= 1 to TempN do
      begin
        X2A:=X2;
        TempY:= round(factor*GdsFArr[I,1]);
        Plot(X2A,Y2-TempY,2);
        Freq:=2;
        for J:= 2 to NumFreq+1 do
          begin
            TempY:= round(factor*GdsFArr[I,J]);
            tempF:=round(40*ln(Freq)/ln(10));
            Plot(X2A+TempF,Y2-TempY,2);
            Freq:=Freq+2;
            if Freq = 12 then
              Begin
                Freq:=2;
                X2A:=X2A+40;
              end;  {if loop}
          end; {do loop j}
```

```
      end;  {do loop i}

    readln(ch);
    case ch of
      'a' : goto 3;
      else goto 3;
    end;  {end of case}
3:  clrscr;
    write('Do you want the plot in different scale
           (Y/N) ? ');
    readln(ch);
    case ch of
      'Y','y' : begin
                  ScaleCh3;
                  goto 2;
                end;
      'N','n' : goto 4;
      else goto 3;
    end;  {end of case}
4:end;    {end of Gds vs. Freq}

begin     {beginning of main program}
   nameA:= 'Gds(mS/mm)';
   nameB:= 'Ids(mA)    ';
   nameC:= 'Gm(mS/mm) ';
   clrscr;
1: writeln('                          Main Menu');
   writeln('                          ~~~~~~~~~');
   writeln('                      0 : Input Data');
   writeln('                      1 : I-V characteristic');
   writeln('                      2 : Gds vs. Vgs');
   writeln('                      3 : Gm vs. Vgs');
   writeln('                      4 : Gds vs. Frequency');
   writeln('                      5 : Print data');
   writeln('                      6 : Quit');

   readln(ch);
   case ch of
     '0' : begin
             Inputdata;
             clrscr;
             goto 1;
           end;
     '1' : begin
             IVcurve;
             clrscr;
             goto 1;
           end;
     '2' : begin
             OutCond;
             clrscr;
```

```
            goto 1;
          end;
      '3' : begin
            Transcon;
            clrscr;
            goto 1;
          end;
      '4' : begin
            GdsFreq;
            clrscr;
            goto 1;
          end;
      '5' : begin
            data;
            clrscr;
            goto 1;
          end;
      '6' : goto 2;
      else  begin
            clrscr;
            goto 1;
          end;
    end;    {end of case}
2:end.      {end of main program}
```