THE OPTIMIZATION OF SPICE MODELING

PARAMETERS UTILIZING THE TAGUCHI METHODOLOGY

by

John F. Naber

Dissertation submitted to the faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Doctorate of Philosophy

in

Electrical Engineering

APPROVED:

Aicha Elshabini-Riad

A.Elshabini-Riad, Chairperson

Douglas J. Nelson

D.J.Nelson

S.Onishi

Sedki M. Riad

S.M.Riad

Joseph G. Tront

J.G.Tront

April 27, 1992

Blacksburg, Virginia

THE OPTIMIZATION OF SPICE MODELING

PARAMETERS UTILIZING THE TAGUCHI METHODOLOGY

by

John F. Naber

Aicha A. Elshabini-Riad, Chairperson

Electrical Engineering

(ABSTRACT)

A new optimization technique for SPICE modeling parameters has been developed in this dissertation to increase the accuracy of the circuit simulation. The importance of having accurate circuit simulation models is to prevent the very costly redesign of an Integrated Circuit (IC). This radically new optimization technique utilizes the Taguchi method to improve the fit between measured and simulated I-V curves for GaAs MESFETs. The Taguchi method consists of developing a Signal-to-Noise Ratio (SNR) equation that will find the optimum combination of controllable signal levels in a design or process to make it robust or as insensitive to noise as possible. In this dissertation, the control factors are considered the circuit model curve fitting parameters and the noise is considered the variation in the simulated I-V curves from the measured I-V curves. This is the first known application of the Taguchi method to the optimization of IC curve fitting model parameters. In addition, this method is not technology or device

dependent and can be applied to silicon devices as well. Improvements in the accuracy of the simulated I-V curve fit reaching 80% has been achieved between DC test extracted parameters and the Taguchi optimized parameters. Moreover, the computer CPU execution time of the optimization process is 96% less than a commercial optimizer utilizing the Levenberg-Marquardt algorithm (optimizing 31 FETs). This technique does a least square fit on the data comparing measured currents versus simulated currents for various combinations of SPICE parameters. The mean and standard deviation of this least squares fit is incorporated in determining the SNR, providing the best combination of parameters within the evaluated range. Furthermore, the optimum values of the parameters are found without additional simulation by fitting the response curves to a quadratic equation and finding the local maximum. This technique can easily be implemented with any simulator that utilizes simulation modeling parameters extracted from measured DC test data. In addition, two methods are evaluated to obtain the worst case modeling parameters. One method looks at the correlation coefficients between modeling parameters and the second looks at the actual device parameters that define the +/- 3σ limits of the process. Lastly, an example is given that describes the applicability of the Taguchi methodology in the design of a differential amplifier, that accounts for the effect of offset voltage.

ACKNOWLEDGEMENTS

I would like to sincerely thank Dr. A.E.Riad for her help, understanding and friendship not only on this research, but throughout my five years of study at Virginia Tech. I would like to thank my graduate committee and Loretta Estes for their help and cooperation. I would also like to thank Dr. H.P.Singh and Dr. D.Fisher at the ITT Gallium Arsenide Technology Center for their encouragement and making this research work possible. In addition, I would like to thank Tomas Zlebek of the ITT Statistical Process Group and Dr. M.Phadke of Phadke Associates whose help, guidance and suggestions started this research.

I would like to thank my entire family for their enduring love and especially Mom and Dad, whose encouragement and moral support throughout my academic years will be forever appreciated. Lastly, I would like to dedicate this work to my wife Cara and newborn daughter Julia. I thank them for their unconditional love and understanding that has brought abundant joy and happiness into my life.

Table of Contents

CHAPTER I 1

INTRODUCTION 1

CHAPTER II 5

LITERATURE REVIEW 5

2.1 INTRODUCTION 5

2.2 GaAs SPICE MODELS 6

2.3 I-V CURVE OPTIMIZATION 8

2.4 CONCLUSION 9

CHAPTER III 11

EXTRACTION OF DC TEST PARAMETERS 11

3.1 INTRODUCTION 11

3.2 EXTRACTION PROCEDURE 12

CHAPTER IV 17

EVALUATION OF PRECISE I-V CURVE OPTIMIZER 17

4.1 INTRODUCTION 17

4.2 JFET MODEL 18

4.3 CURTICE MODEL 26

CHAPTER V **31**

EVALUATION OF TAGUCHI I-V CURVE OPTIMIZER 31

5.1 INTRODUCTION 31

5.2 SIGNAL-TO-NOISE RATIO 34

5.3 TEST RESULTS 41

CHAPTER VI **55**

IMPLEMENTATION OF THE TAGUCHI OPTIMIZER 55

6.1 INTRODUCTION 55

6.2 DESCRIPTION AND RESULTS OF PROGRAM FIT1.FOR 55

6.3 STATISTICAL MODELING AND WORST CASE PARAMETERS 69

6.4 APPLICATION OF OPTIMIZER TO OTHER MODELS 78

CHAPTER VII **87**

DIFFERENTIAL AMPLIFIER DESIGN 87

7.1 INTRODUCTION 87

7.2 OFFSET VOLTAGE CHARACTERIZATION 89

7.3 TAGUCHI METHOD APPLIED TO A DIFF-AMP DESIGN 96

CHAPTER VIII **107**

CONCLUSION 107

APPENDIX I 110

APPENDIX II 125

BIBLIOGRAPHY 139

VITA 146

CHAPTER I

INTRODUCTION

In the last ten years the commercial GaAs IC industry has not lived up to expectations in competition with high-speed silicon. However, recent projections for the GaAs IC industry predict up to a billion market by 1995 [1],[2]. The digital and microwave (analog circuits operating over 1 Ghz) GaAs IC markets are proliferating, but the analog or linear market is relatively non-existent. For example, no A/D converters or op-amps are commercially available. There are a multitude of reasons for the shortcomings of linear GaAs circuits, some of which are: material dependent anomalies which affect performance, direct competition with high-performance silicon processes (speed/power/cost tradeoffs), lack of experienced analog/linear designers, and lack of available models which can accurately simulate analog circuitry. The goal of this research work is to develop a generic circuit optimizer that can quickly optimize all devices on the wafer to enable a better match between measured and simulated DC I-V curves for analog as well as digital circuit design.

The impetus for this research was the poor agreement between measured and simulated circuits from the ITT Gallium

Arsenide Technology Center's (GTC) mask set GTC-305 and GTC-282. The GTC-305 design contained various analog circuits, such as sample and holds, op-amps, differential amplifiers and comparators and the GTC-282 design contained an op-amp. The data from GTC-305 and GTC-282 showed typical DC output levels that varied up to 100%. In addition, dynamic characteristics from GTC-305, such as gain and common mode rejection, had average variations of 40% between measured and simulated results. Discrepancies between measured and simulated results of this magnitude will render the simulations ineffective for small signal analog design. This in turn will prevent customers (both captive as well as commercial) from using GTC's Multi-function Self-Aligned Gate (MSAG) GaAs IC process. ITT-GTC's MSAG process is a paradigm of GaAs IC processes because of its state-of-the art results and unmatched versatility [3].

The prior approach at GTC to simulating these types of circuits involved using the program: Simulation Program for Integrated Circuit Emulation (SPICE), with a Junction Field Effect Transistor (JFET) model and no parameter optimization. Berkeley SPICE [4] is a widely used time domain simulator developed sixteen years ago and is commercially available (cost is approximately two-hundred dollars). However, SPICE has demonstrated the large aforementioned inaccuracies due primarily to poor agreement between measured and simulated DC I-V curves for the MSAG FETs. Furthermore, SPICE does not

include the capability to optimize the model parameters or model specific problems inherent to GaAs such as, backgating and frequency dispersion [5],[6]. This research work will focus on optimizing DC I-V curves and not improving frequency dependent properties of the SPICE model because much research has been done in this area [7],[8],[9]. The newest version of SPICE (version 3e.2) is written in "C" and contains silicon BJT, MOSFET and GaAs MESFET models. The principal GaAs models used in most commercial SPICE simulators are the JFET, Curtice, Statz or Raytheon, and TriQuint (TOM) models [10],[11],[12],[9]. This research will consist of using the existing physics based models and Taguchi methods [13],[14] for I-V curve optimization to improve the correlation between measured and simulated results for both digital as well as analog GaAs integrated circuits. This is the first known use of Taguchi methods to the optimization of parameters in an IC model.

This dissertation consists of eight different chapters. This chapter, Chapter I, is a general introduction. Chapter II presents a literature survey on SPICE models for GaAs and IC model parameter optimization. Chapter III consists of the methodology used to extract the IC model parameters from DC test. Chapter IV presents an evaluation of a commercial optimizer using JFET and Curtice models. Chapter V presents the results of the initial investigation into utilizing the Taguchi methodology. Chapter VI takes the results in Chapter

V and develops a practical implementation of the Taguchi method into the current manufacturing process as well as develop a method to obtain worst case model parameters. Chapter VII uses the results of the optimized model parameters from the previous chapters to design a differential amplifier using Taguchi methods. Chapter VIII presents the over all conclusions and appendices I and II lists the computer programs used in this research.

CHAPTER II

LITERATURE REVIEW

2.1 INTRODUCTION

The literature review will consist of summarizing the SPICE models used in simulating GaAs MESFET analog and digital circuits over the last twenty years. This review is by no means inclusive of all the GaAs SPICE models developed over the years, but the purpose is to highlight the models that had the most significant impact on the industry. The review will start with the first SPICE model used to simulate GaAs circuits in the early 1970's up to the latest TOM model published in 1990. In addition, there will be a short review on model parameter optimization. The Levenberg-Marquardt algorithm was first applied to nonlinear least squares approximation in the early 1970's. However, it wasn't utilized in the optimization of IC model parameters until ten years later. IC model parameter optimization is a very limited and relatively new topic, with the initial paper on the subject published as recently as 1983.

2.2 REVIEW OF SPICE MODELS FOR GaAs

The Shichman and Hodges JFET model [10] is the first primarily used SPICE model to simulate GaAs MESFET devices through the first decade of GaAs IC simulation (1970-1980). The model was developed in 1968 to describe the behavior of silicon P-N junction FETs of either N-type or P-type channel. However, the model typically showed a poor fit for currents below saturation [11].

In 1980, the Curtice square law model [11] was introduced and included two significant attributes. First was the inclusion of a hyperbolic tangent term with an additional SPICE curve fitting parameter. The method was first proposed by Van Tuyl and Liechti, but not widely publicized. This eliminated two equations to model the linear and saturation regions, as well as gate voltage independence of saturation. Secondly, a variable was introduced to account for the transit time delay of the MESFET.

In 1983 and 1985, two modeling papers by Materka and Kacprzak [15],[16] were published which were the first to introduce Idss or saturation current into the drain source equation. In addition, these authors were the first to incorporate threshold voltage as a function of drain to source voltage.

The Curtice cubic model was introduced in 1985 [17]. This model improves the DC curve fit of the first Curtice model [11] using a third order polynomial. Model parameter identification is more difficult with this model than the others. Furthermore, non-physical behavior can result for gate voltages below the threshold voltage [18].

The Statz or Raytheon model was reported in 1987 [12]. The three primary distinctions of this model are: 1) a polynomial expression to approximate the hyperbolic tangent, 2) uses the square law for small currents and square root law for large currents, 3) a charge conservation approach used for nodal capacitances.

A paper by Larson in 1987 reference [19] points out deficiencies in models [10] and [11]. In [10], the limitation is the square law relationship of drain current to gate voltage. In [11], the limitation is due to a poorer agreement for enhancement mode FETs than depletion mode FETs. The new model incorporates two new parameters that models a variable exponential expression between drain current and gate voltage, as well as accounting for early saturation.

The TriQuint or TOM model [9] was published in 1990 which is a modification of the Statz model [12]. The fit around the threshold voltage (Vt) was improved by making Vt a function of the Vds voltage. The non-square law term was used as reported

in [19]. Furthermore a term was added to account for a negative slope in the I-V curves at high currents and voltages.

2.3 REVIEW OF I-V CURVE OPTIMIZATION

The following is a brief synopsis of the optimization of SPICE model parameters to improve the accuracy of the simulation. These techniques and methods are not particular to a single model and can be used with any SPICE model. In 1972 the Levenberg-Marquardt algorithm was described for use in nonlinear least squares approximation [20]. The Levenberg-Marquardt algorithm combines two numerical techniques resulting in a synergistic algorithm for the parameter optimization. For instance, the algorithm numerically combines the Gauss-Newton method with the steepest descent method to give the largest stability in the minimization of the error away from the minimum and fast convergence near the minimum. This results in an easy to use and computationally fast algorithm that can optimize all model parameters concurrently in a single operation.

In 1982, the SUXES FET parameter extraction program for silicon devices was introduced [21]. This program became publicly available through Stanford University, but contained

no parameter optimization capability to improve the accuracy of the extracted parameters in the model.

The following year saw the Levenberg-Marquardt algorithm introduced into the SUXES program [22] to optimize the extracted model parameters and significantly improve the accuracy of the model. This program solved all model parameters simultaneously to limit interactions between the parameters and could be used for any generic SPICE model.

From 1985 through 1991, commercial software companies started to introduce optimization capability into their simulators based on the Levenberg-Marquardt algorithm. For example, Silvaco introduced Utmost in 1988, and Electrical Engineering Software introduced its optimizer in 1990. However, these optimizers are quite expensive (excess of $ 40,000) and can be an unrealistic expense for most individuals and universities.

2.4 CONCLUSION

Hence, SPICE models for GaAs have evolved from the JFET model, which was applied directly from silicon experience to the Statz and TOM models. These latter models were developed just for GaAs MESFET devices and addressed some of the particular anomalies associated with GaAs. The accuracy of these as well as any SPICE like models can be significantly

improved by optimizing the model parameters. The Levenberg-Marquardt algorithm was utilized to give a computationally fast method that can concurrently optimize all the model parameters to any SPICE like model.

The next chapter will explain the method to extract the model parameters from the actual measured devices. It will be these extracted parameters that shall undergo the optimization procedure to improve the accuracy between measured performance and simulated performance.

CHAPTER III

EXTRACTION OF DC TEST PARAMETERS

3.1 INTRODUCTION

A description of the extraction of SPICE modeling parameters is in order prior to doing a comparison between measured and simulated I-V curves. The JFET model described in the previous chapter requires nine parameters to be used incorporated into the model (Vt, Rs, Rd, Cgs, Cgd, Vbi, Ileak, beta, and lambda). The threshold voltage or Vt is the value of gate voltage which causes the FET to cease conducting current. Rs and Rd are the parasitic source and drain resistances seen by the FET. Cgs and Cgd are the parasitic diode capacitances seen from gate to source and gate to drain. Vbi is built in schottky barrier of the FET and Ileak is the leakage current between the gate and the channel. These first seven parameters are actual physical quantities, whereas the last two are curve fitting parameters. All nine parameters can be extracted by methods that physically measure the FETs as shown in references [23,24]. However, only five of the nine parameters (Vt, Rs, Rd, beta, lambda) have any effect on the DC I-V curves.

There is a serious problem with these extraction techniques. These extraction techniques are only exact for an

ideal set of I-V curves, which will rarely be seen in actuality. Hence, there will be a significant difference between measured I-V curves and simulated I-V curves. Furthermore, the discrepancy between extracted parameter I-V curves and simulated I-V curves are exacerbated by variations due to: different technologies, different processes, short channel effects and so on. Therefore, a feedback mechanism must exist to compare the measured and simulated I-V curves against each other and optimize the measured model parameters to give an improved fit. The I-V curve optimizer acts as this feedback mechanism. Chapter V will give further insight into extraction with respect to the over all design flow.

3.2 EXTRACTION PROCEDURE

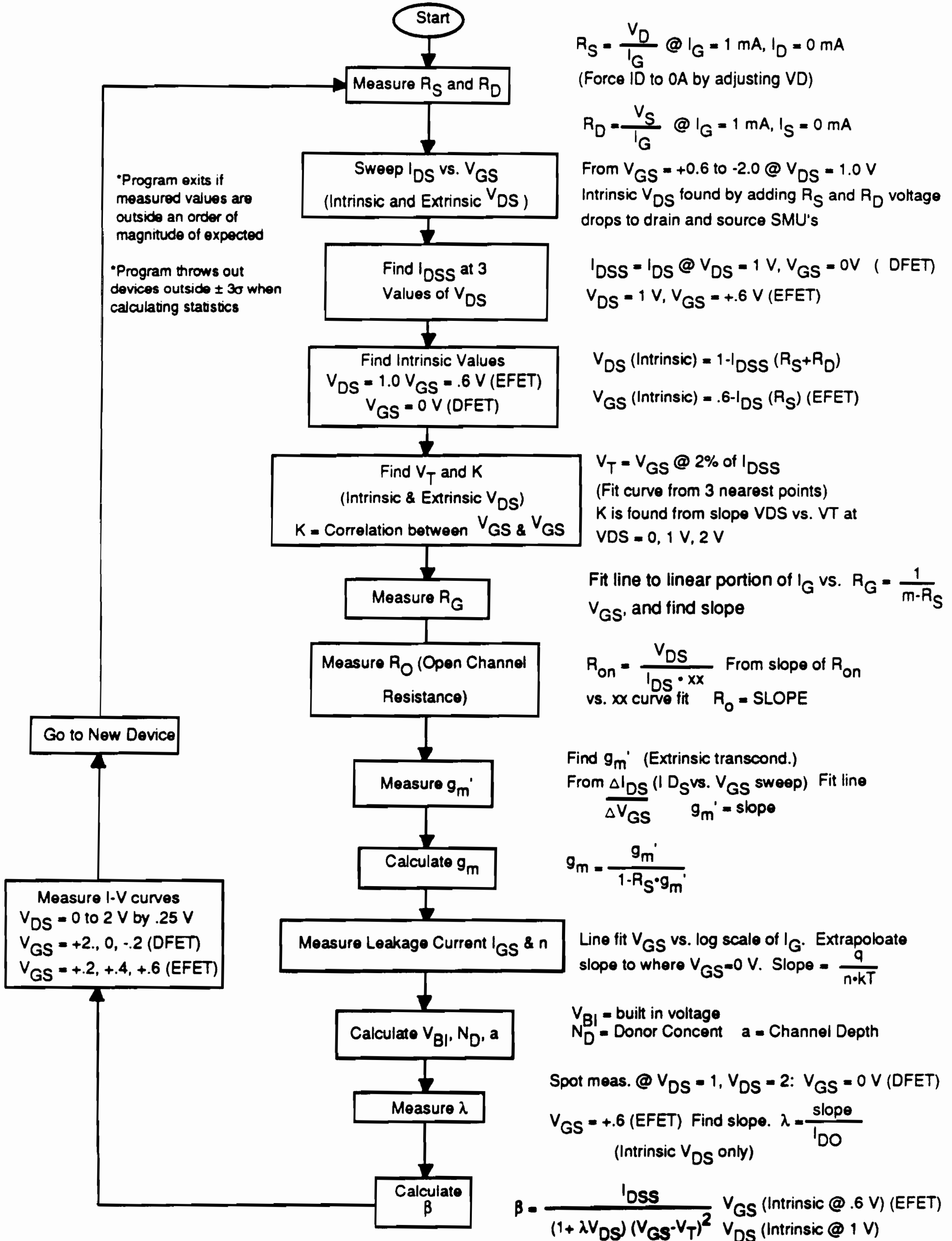
The measured DC data is attained by making on-wafer measurements of a 3-inch wafer that typically contains up to 50 Process Control Monitor (PCM) FETs. A rectangle encompassing a significant section of the 3-inch diameter wafer is chosen and all PCM FETs within this rectangle are evaluated for their measured DC parameters (threshold voltage, saturation current, transconductance, source resistance, drain resistance) and JFET SPICE parameters (lambda, beta) using techniques explained in [23],[24]. The mean and standard deviation for all SPICE modeling parameters as well as some additional parameters are determined with built-in hard limits

to eliminate bad data. The measured data was obtained using an HP-4142 parameter analyzer by sweeping the gate voltage from 0 to -1 V in 100 mV steps and the drain voltage from 0 to 2 V in 100 mV steps.

Figure (3.1) shows a flow chart of the currently used program at ITT-GTC for extracting the various FET and SPICE parameters. A modification to the aforementioned references is in the extraction of Vt. For instance, Vt is not found from the slope of the square-root of Ids versus gate voltage as indicated in the references. Instead, Vt is found from the gate voltage that coincides with 2% of Idss. This is the same method used by the fabrication engineers at ITT and allows direct correlation of their measurements. Finding Vt using both methods on numerous wafers showed the results to be quite close (within 20 mV). Another deviation from references [23,24] is in the calculation of beta. Instead of measuring beta from the slope of Ids versus Vgs, beta is calculated as shown in Figure (3.1). This method guarantees very close agreement at the Vds voltage Idss is measured, which results in an overall improvement in the fit from the beta extraction technique in references [23,24].

The local and global variation in the threshold voltage was evaluated to determine if the FET characteristics in the PCM of a mask reticle or die could be applied to the circuit in that same reticle. However, an average wafer showed the

Figure (3.1). DC FET parameter extraction flow chart.



standard deviation of the threshold voltage (σVt) within a 416 mm2 area containing 8 FETs is 43 mV and the σVt within a 1156 mm2 area containing 23 FETs is 37 mV. This result shows the intradie local parameter variation is less than 416 mm2 and it is useless trying to localize the FET parameters of a particular die within the circuit of that same die. Hence, the FET data and circuit data must be averaged over the wafer for comparison purposes. The FET extraction program has the capability to filter outliers in the database to obtain a meaningful representation of the averaged FET parameters for an entire wafer without corruption from outliers. The program can eliminate data outside one, two or three standard deviations and recalculates the data set without the outliers. For example, one average wafer contained a total of 49 FETs, which were averaged together giving a σVt=62 mV. All FETs outside +/- 3σ were thrown out (1 device) and the devices re-averaged. The new σVt became 29 mV because that one bad device was throwing the σVt off considerably. Next, all devices within +/- 1σ of this new average were averaged together to give an even smaller population of 36 FETs with a σVt=17 mV. However, unless otherwise stated the standard outlier filter is set-up to eliminate data outside +/- 3σ.

The next chapter will graphically show the large discrepancy between measured and simulated I-V curves based on the extraction of DC parameters descibed in this chapter. Furthermore, a commercial optimizer will be evaluated using

two different IC models.

CHAPTER IV

PRECISE OPTIMIZER EVALUATION

4.1 INTRODUCTION

This chapter will show the significant benefits and tradeoffs to modeling I-V curves with a commercial SPICE simulator and optimizer from Electrical Engineering Software. The simulator uses a modified version of Berkeley SPICE called PRECISE. The software contains various models for simulating GaAs FETs, such as the JFET, Curtice and Statz models. This work will only evaluate the JFET and Curtice models due to convergence problems found using the Statz model. The PRECISE optimizer uses the Levenberg-Marquardt algorithm as described in Chapter II to do the optimization. The goal of this chapter is to examine the many optimization routines offered to the user and determine if one is superior for the particular measured data. For instance, different parameters in the model will be optimized and an additional parameter (Rds) will be added to determine its effect on the fit. Furthermore, the extracted parameters from DC test will be done at different Vds values giving a significantly different set of simulated I-V curves. The last option to be evaluated in the PRECISE optimizer will be the curve weighting capability in which the

priority to fit each curve can be varied or weighted.

4.2 JFET MODEL

The Shichman and Hodges JFET model found in SPICE has been used extensively to simulate the I-V characteristics of GaAs MESFETs because of the relatively good agreement, widely available model, and simplicity of parameter extraction [25]. Now that the DC and SPICE parameters have been obtained, a plot can be generated showing a comparison between measured and simulated results. Figures (4.1) and (4.2) show plots of the measured versus the simulated I-V curves for Vgs=0.0, -0.1, and -0.2 V. The SPICE parameters are extracted at Vds=1.0 V in (4.1) and Vds=2.0 V in (4.2) and are listed along with the DC parameters. This is a graphical portrayal of the need for curve optimization. The large variation between measured and simulated I-V curves has been the principal reason for the poor agreement between measured and simulated analog designs. These figures show the measured and simulated curves agree quite well around the set Vds voltage, but deviate significantly elsewhere. This close agreement around the set Vds voltage is due to a modification in the calculation of beta from the referenced FET parameter extraction procedure. If the reference was followed verbatim (calculating beta from the slope of Ids vs. Vgs) the fit would be significantly worse. Furthermore, the values for the threshold voltage when plotted at other Vds values as well

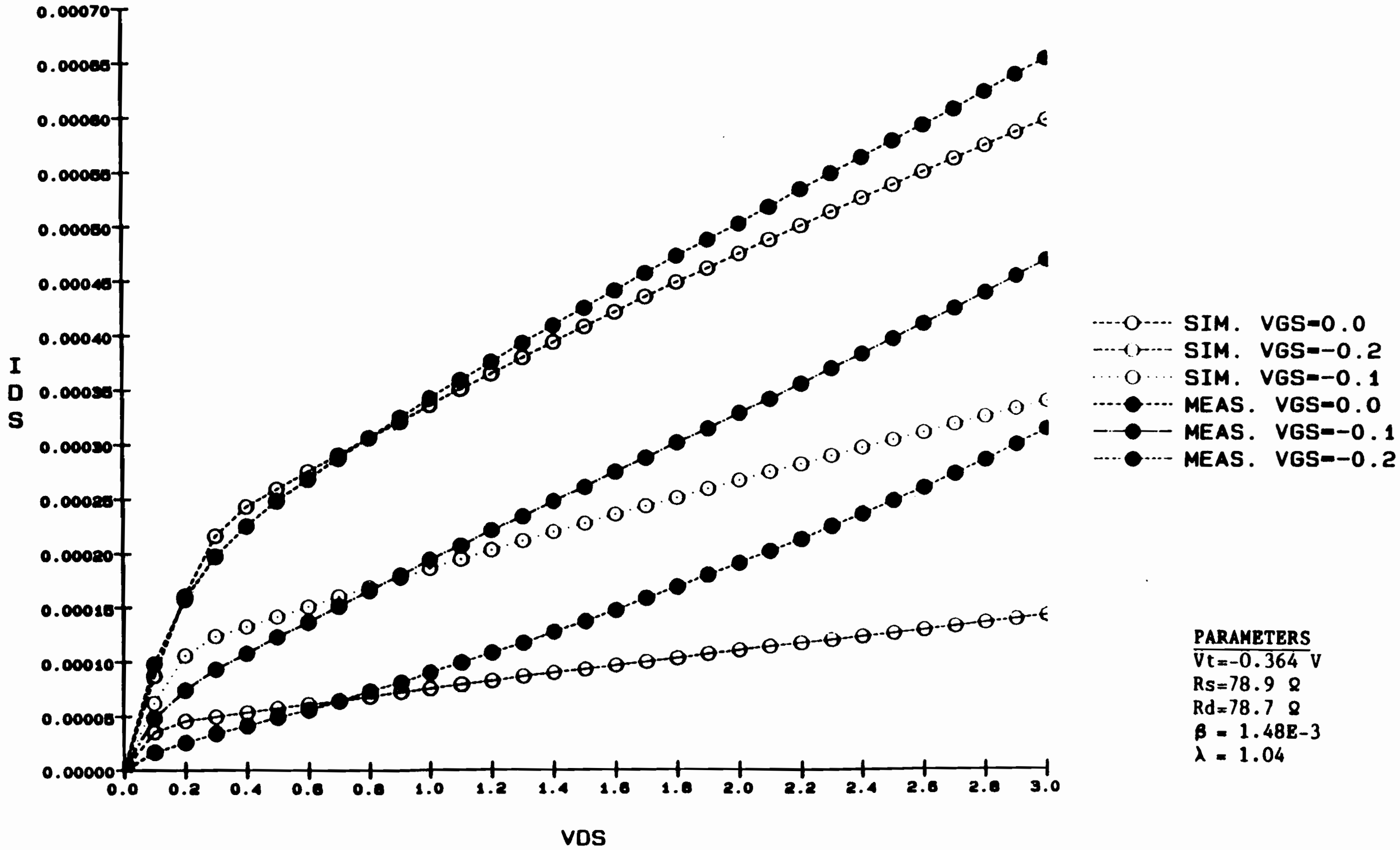


Figure (4.1). Measured versus simulated I-V curves at Vds=1.0 V.

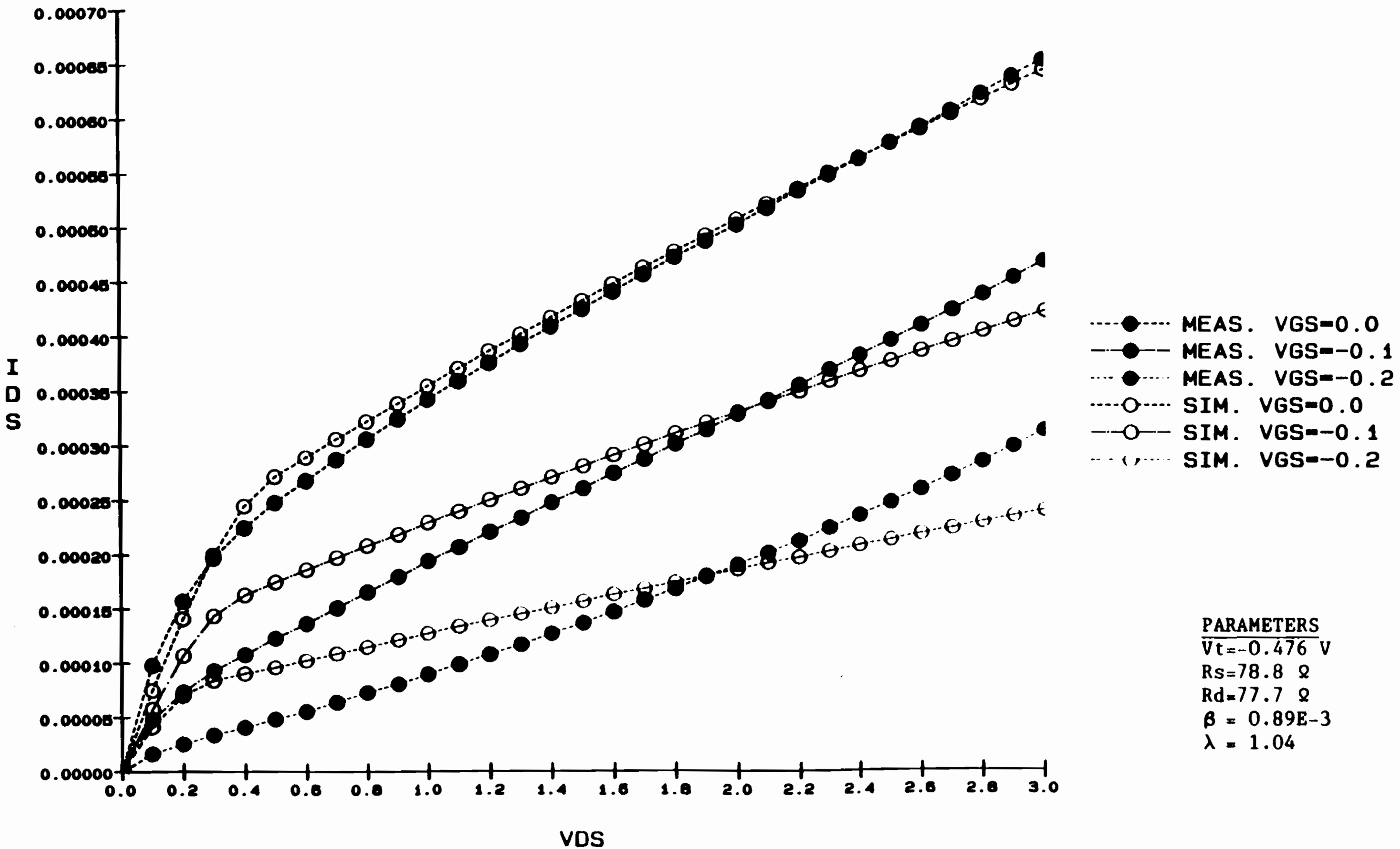


Figure (4.2). Measured versus simulated I-V curves at Vds=2.0 V.

show a distinct relationship as a function of Vds. Reference [9] incorporates this behavior into the TOM model.

This non-ideal fit leads us to the optimization issue, which is the next step to improve the fit between measured and simulated data. Optimization is a very powerful technique that modifies the various FET parameters and compares them to a given set of curves to improve the fit between measured and simulated results [22]. Chapter V will cover optimization using the Taguchi approach. PRECISE contains optimization routines that have a wide range of capabilities. Figures (4.3) through (4.5) shows results for a progression of optimization techniques. In Figure (4.3) the SPICE parameters lambda and beta were optimized only from measured parameters at Vds=1.0 V and a resistor was added to the JFET model between source and drain Rds. In Figure (4.4) again only lambda and beta were optimized, but a weighting function was used. The weighting function weighted the importance of each curve on a 4-2-1 ratio, which reflects the relative magnitude of current between the curves. In other words, the Vgs=0.0 V curve carried 4 times as much significance in the optimization routine as the Vgs=-0.2 V curve. Lastly, Figure (4.5) shows the curves being weighted and optimized with lambda, beta, and Rds. Table (4.1) lists the absolute error (ABSERR) for the various techniques used to obtain the SPICE parameters and the percent improvement from directly incorporating the measured SPICE parameters into the model. The ABSERR is the RMS value

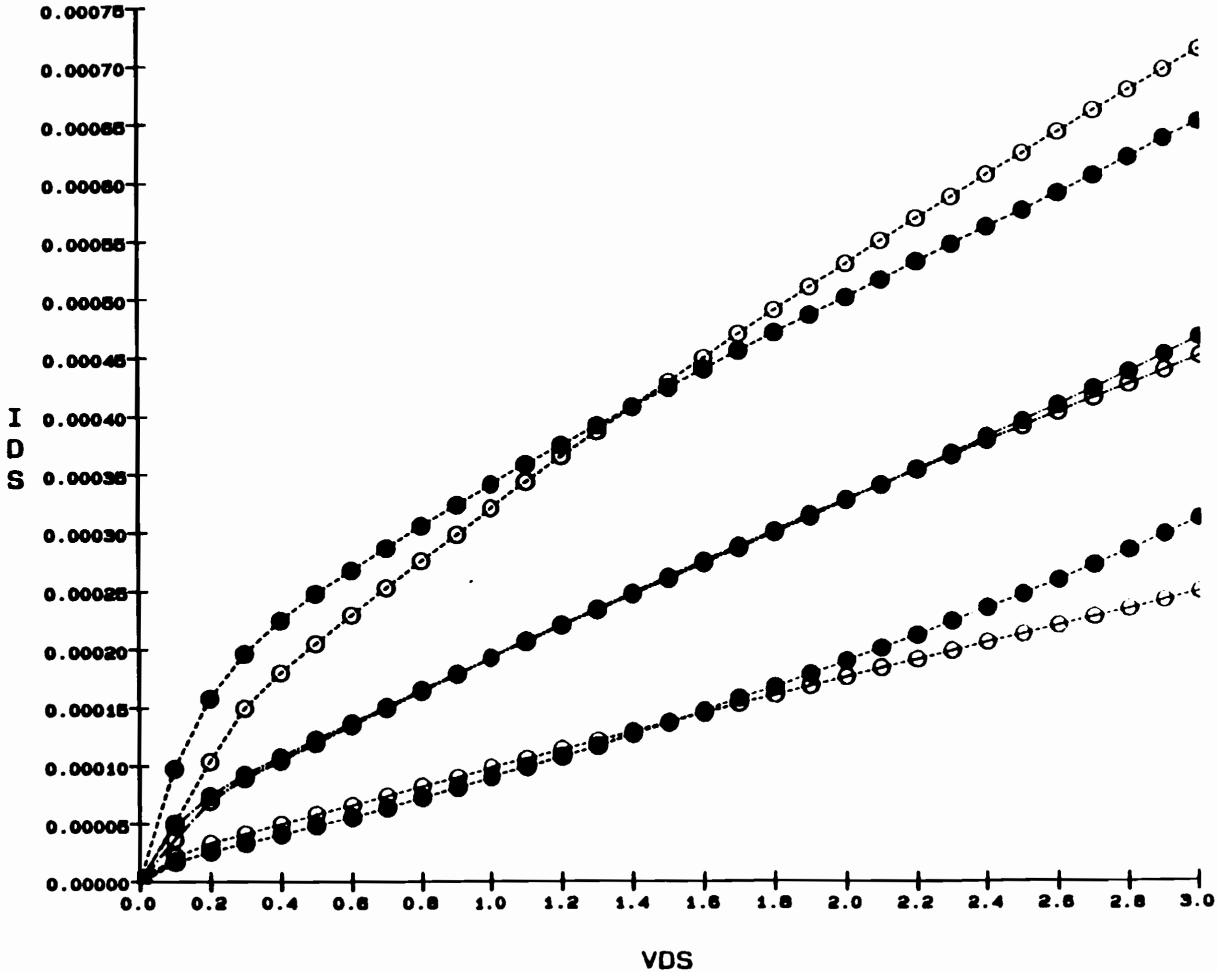


Figure (4.3). Measured versus simulated I-V curves with λ, β, and Rds optimized.

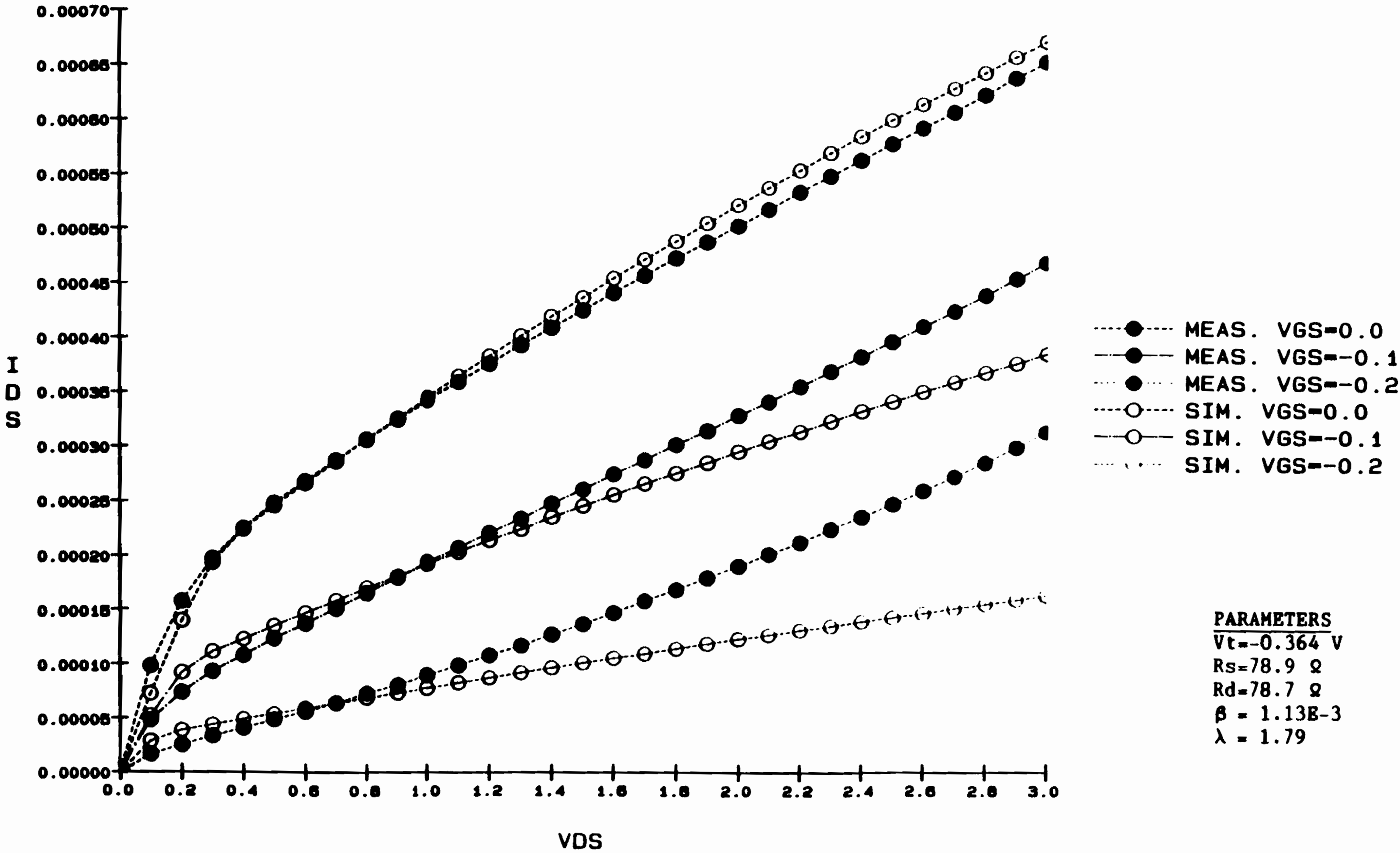


Figure (4.4). Measured vesus simulated I-V curves using curve weighting and optimizing λ and β.

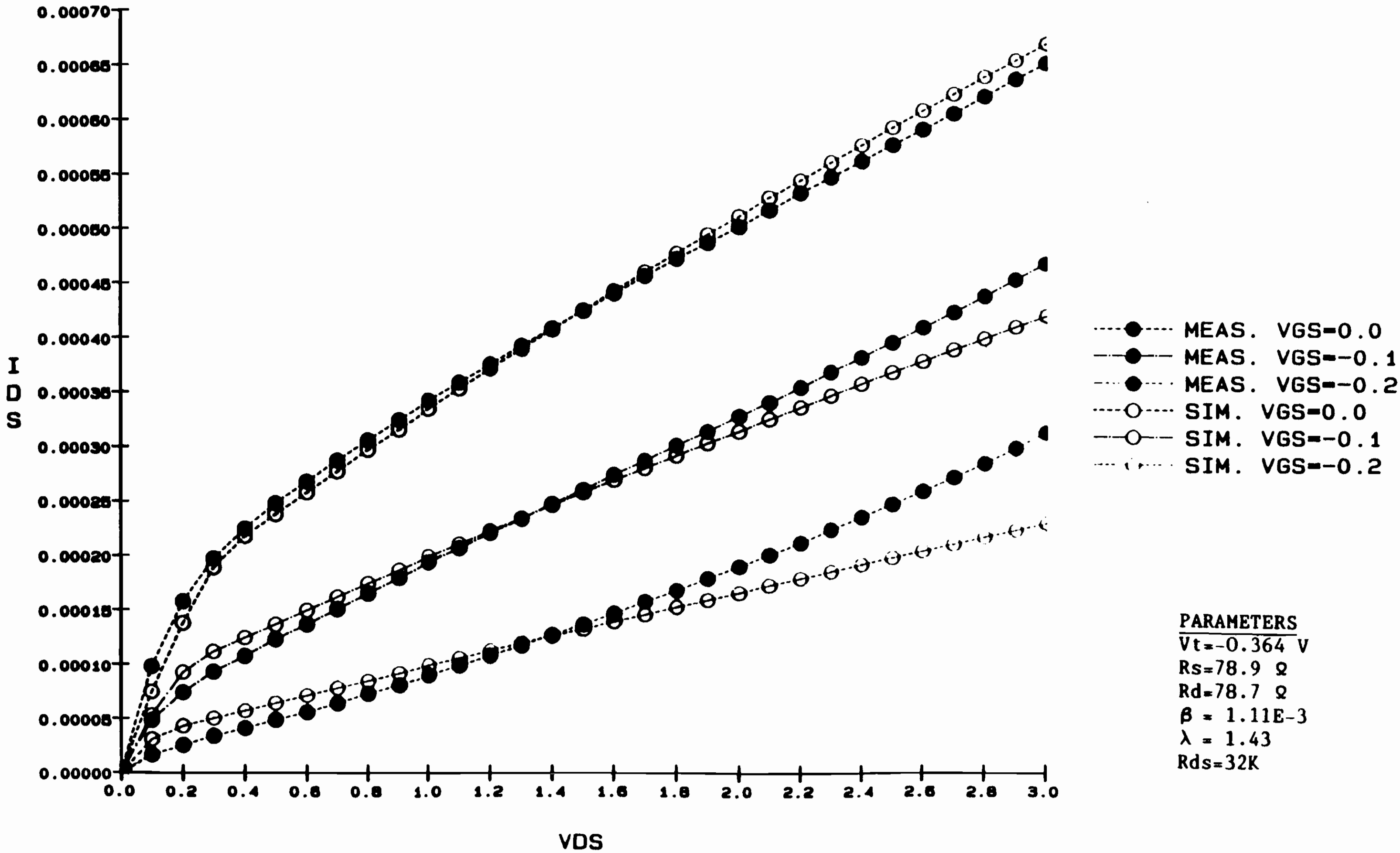


Figure (4.5). Measured vesus simulated I-V curves using curve weighting and optimizing λ, β, and Rds.

Table (4.1). Absolute error and percent improvement from technique (1) using various simulation methods for the JFET model.

TECHNIQUE	ABSERR	% IMPROVEMENT
(1) Parameters Measured; λ and B only at $V_{DS} = 1V$	6.08E-3	—
(2) λ and B optimized	5.20E-3	15%
(3) λ, B and R_{DS} optimized	2.58E-3	58%
(4) λ and B optimized using weighting	4.60E-3	24%
(5) λ, B and R_{DS} optimized using weighting	2.37E-3	61%
(6) Parameters measured; λ and B only at $V_{DS} = 2V$	2.87E-3	53%
(7) λ and B optimized	2.18E-3	64%
(8) λ, B and R_{DS} optimized	2.06E-3	66%
(9) λ, B optimized with weighting	2.26E-3	63%
(10) λ, B and R_{DS} optimized with weighting	2.29E-3	62%

of all the differences between the measured and the simulated curves. Therefore, for these curves, ABSERR is the RMS error in Amps of the Ids difference between measured and simulated currents. This number allows the quantification of various techniques to improve the correlation between the measured and the simulated curves. The relative error function in PRECISE proved erroneous because it was dependent upon whether the simulated curve was above or below the measured curve (apparently not true RMS).

These results show the dependence upon the value of Vds when extracting and optimizing the various parameters. For instance, at Vds=1 V the initial fit is poor and basically remains that way until the Rds term is incorporated into the optimization. On the other hand, at Vds=2 V the initial fit is much better and all optimization methods give similarly small improvements. Therefore, if this trend is true of giving a better initial fit using the JFET model at Vds=2, then for simplicity, no weighting or Rds term will be required when optimizing the fit.

4.3 Curtice Model

The primary difference of the Curtice model from the JFET model is the inclusion of single continuous current equation for the linear and saturation regions that uses a hyperbolic tangent term. Also, a tau term is included for the initial

channel current delay when modulating the gate. The hyperbolic tangent term uses the variable alpha, which must be extracted with the lambda and beta SPICE parameters. However, since the extraction program does not extract alpha the procedure used for the JFET model must be modified. This is done by using the same initial ABSERR as used in the JFET model and observing the accuracy improvement in the optimization process using the Curtice model. These results are illustrated in Table (4.2).

The general behavior is identical to the JFET model results with Vds=2 V. However, at Vds=1 V the trend is the same, but the fit for all optimization techniques except one is approximately 15% better. Therefore, if the initial SPICE parameters are obtained at the non-optimum Vds value (for our example Vds=1 V), then the best fit between the measured and the simulated I-V curves will be obtained using the Curtice model and optimizing lambda, beta, alpha, and Rds without weighting the curves. Conversely, if the optimum Vds voltage is chosen (for our example Vds=2 V), then any optimization technique is equally effective. Hence, assuming the optimum Vds value varies from wafer to wafer, to consistently obtain the best fit would require determining the optimum Vds value each time and using Rds. Table (4.3) lists the actual SPICE model parameters used for 6 of the 20 simulation techniques described in Tables (4.1) and (4.2).

In summary, there are various ways and techniques that

Table (4.2). Absolute error and percent improvement from technique (1) using various simulation methods for the Curtice square model.

TECHNIQUE	ABSERR	% IMPROVEMENT
(1A) Parameters Measured; λ, B only at $V_{DS} = 1V$	6.08E-3	—
(2A) λ, B and α optimized	4.40E-3	28%
(3A) λ, B, α and R_{DS} optimized	1.49E-3	76%
(4A) λ, B, α optimized using weighting	4.70E-3	23%
(5A) λ, B, α and R_{DS} optimized using weighting	1.51E-3	75%
(6A) Parameters measured; λ and B only at $V_{DS} = 2V$	2.87E-3	53%
(7A) λ B and α optimized	2.20E-3	64%
(8A) λ, B, α and R_{DS} optimized	1.92E-3	68%
(9A) λ, B and α optimized with weighting	2.21E-3	64%
(10A) λ, B, α and R_{DS} optimized with weighting	2.11E-3	65%

Table (4.3). Summary of the SPICE model parameters used in 6 simulation techniques from Tables (4.1) and (4.2).

SIMULATION TECHNIQUE #	MODEL	ABSERR	V_T	R_S	R_D	λ	B	α	R_{DS}
1	JFET	6.08E-3	-.364	78.9	78.7	1.04	1.48E-3	—	—
3	JFFET	2.06E-3	-.476	78.8	77.7	1.73	.576E-3	—	86K
6	JFET	2.87E-3	-.476	78.8	77.7	1.04	.891E-3	—	—
7	JFET	2.18E-3	-.476	78.8	77.7	1.88	.577E-3	—	—
3A	CURTICE	1.49E-3	-.364	78.9	78.7	1.17	1.11E-3	6.88	20K
8A	CURTICE	1.92E-3	-.476	78.8	77.7	1.86	.546E-3	7.27	88K

CONSTANT PARAMETERS: C_{gs} = 15 FF C_{gd} = 1.5 FF I_S = 1E-15
PB = .72 R_g = 250 C_{ds} = 0

can be effectively used to significantly improve the I-V curve fit using the PRECISE optimizer. However, all of them take significant time to set-up and execute because each optimization run for a single FET requires a SPICE simulation. Typically each FET takes approximately 6 optimization iterations, when multplied by the number of FETs measured on a wafer (30) results in 180 SPICE simulations. Therefore, from a practical standpoint in the amount of time it takes to optimize a given set of curves, all the FETs can't be optimized and only a single set of averaged I-V curves from any particular wafer can be optimized. In Chapter VI it will be illustrated how the parameters from optimizing a wafer-averaged set of I-V curves gives noticeably different results from the parameters obtained by averaging optimized parameters from each FET on the wafer. In the next chapter a the Taguchi method to optimization will be introduced. This is a radically new concept to be applied to I-V curve optimization.

CHAPTER V

TAGUCHI CURVE FITTING EVALUATION

5.1 INTRODUCTION

A basic precept of the Taguchi methodology is to optimize the control variables of a design or process with respect to the noise factors to make that design or process robust. Robustness is a non-quantitative gauge to the insensitivity of noise. Therefore, the less sensitive a design or process to noise, the more robust the design or process. The more typical approach used today for improving a design or process is an attempt to reduce or eliminate the noise factors altogether, which can become very time consuming and expensive. Hence, the Taguchi method makes a design or process less sensitive to noise by optimizing existing adjustable variables inherent to the design or process, without attempting to reduce the noise sources.

The use of the Taguchi methodology is at first hard to comprehend with respect to optimizing curve fitting parameters to an IC model [13]. Hence, an understanding of the IC design flow and simulation techniques should make it easier to understand and appreciate. Figure (5.1) shows where the techniques discussed here fall in place with regard to the

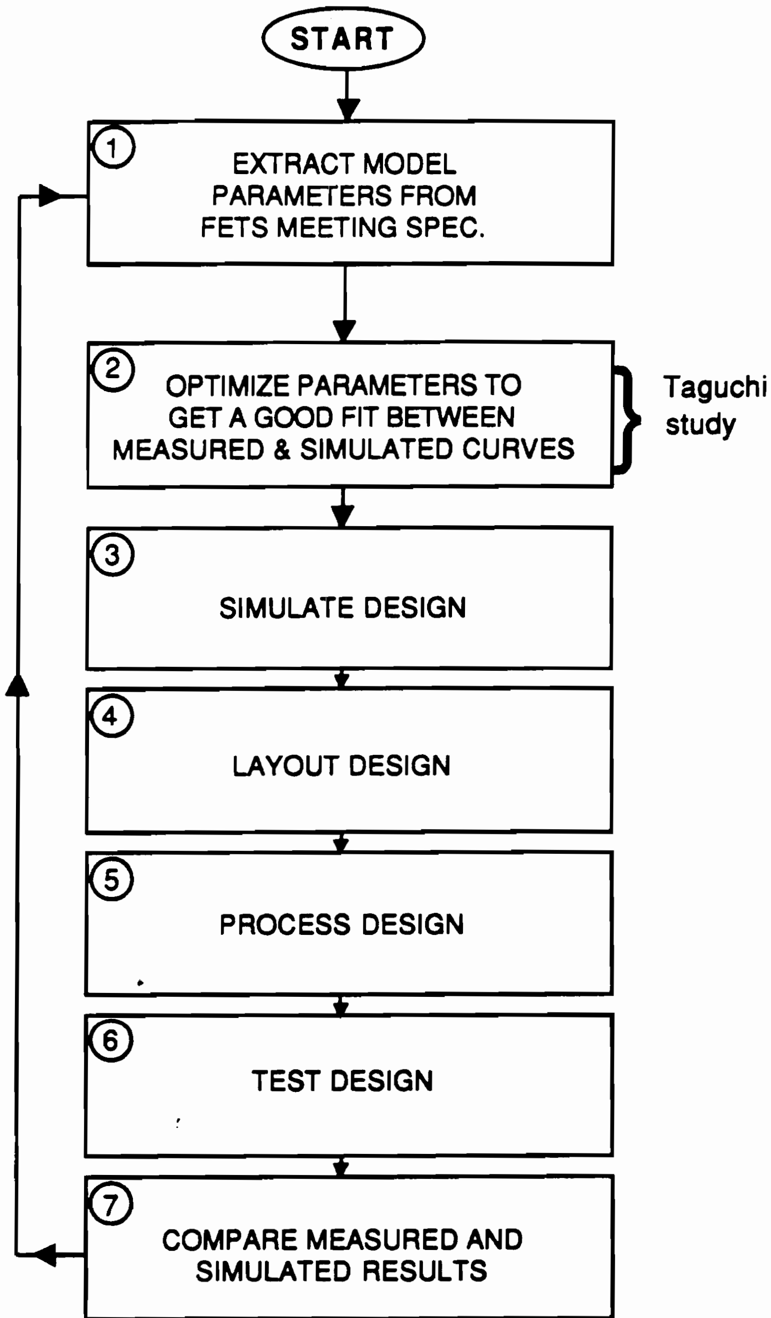


Figure (5.1). Typical integrated circuit design flow.

overall IC design flow. As mentioned previously, a commonly used simulation program for the design of silicon and GaAs ICs is SPICE. This program was developed at the University of California at Berkeley and there currently exists many commercially available derivatives of the original program. SPICE is a time domain circuit simulator that has a variety of mathematically derived current equations to model the behavior of bipolar, MOSFET, JFET, and MESFET transistors. However, to effectively model the behavior of these transistors, the model parameters in the equations must be optimized to attain the best possible correlation between measured and simulated results. These model parameters are optimized by fitting the measured current-voltage or I-V curves to the simulated I-V curves. Commercially available programs can perform this optimization using algorithms discussed in Chapter II, which fit non-linear equations such as, Levenberg-Marquardt. However, some drawbacks to these optimizers are: few commercial simulators use them, the algorithms can get stuck in a local minima, generic optimizers are unsupported, they can cost up to $40,000 and the optimizer can only be used on the models available to a particular simulation package. The commercial optimizer used at ITT-GTC is a SPICE derivative called PRECISE (Electrical Engineering Software). This chapter will evaluate the tradeoffs of simulation accuracy in using the Taguchi method versus using the PRECISE optimizer as well as no optimization at all. In addition, the concept of the Signal-to-Noise Ratio (SNR) as well as the significance of

additivity to limit interactions between the model parameters will be also addressed in this chapter.

5.2 SIGNAL-TO-NOISE RATIO DEVELOPMENT

The objective of this research is to improve the fit between measured and simulated I-V curves by optimizing the parameters which alter the simulation curves. The I-V curves of a transistor determines the behavior of a circuit, which is initially simulated on a computer and consists of solving the Ids current equation as a function of the node voltages. It is imperative that the simulation values are relatively close to the actual measured values because of the large cost and time delay incurred by re-doing a design. For example, it typically takes 6 weeks to process a design and costs approximately $ 30,000 for fabricating 12 wafers not including design or layout time. The computer program uses a variety of nonlinear equations to replicate the physical I-V curves. Hence, by plotting the measured versus the simulated I-V curves as in Figure (4.1) an understanding of the differences is readily seen. The discrepancies between measured and simulated I-V curves in Figure (4.1) can lead to actual circuits which vary up to 100% from simulated results. This discrepancy might be acceptable for digital circuits as long as there is adequate margin around the switching point, but for analog circuits the result is unacceptable. Therefore, the goal of this case study is to improve the accuracy of the analog circuit simulation by

improving the accuracy of the I-V curve fit. This will result in considerable savings in time and money avoiding numerous design iterations.

A listing of some of the primary models and their nonlinear equations for MESFET transistors is shown in Table (5.1). The equations solve the drain current for either the saturation or linear region dependent upon the voltages of the drain, gate, and source. At ITT-GTC the Schichman and Hodges JFET model is primarily used because it requires the extraction of only two SPICE parameters, β and λ. These SPICE parameters are extracted directly from measuring the characteristics of the FET in procedures described in [23],[24]. However, by plugging the measured values of threshold voltage (Vt), β, and λ into the SPICE simulation yields a fit shown in Figure (4.1). This fit gives an RMS error between measured and simulated values of 36 μA, with Vds varied from Vds=0.5,1.0,1.5,2.0 V and Vgs varied from Vgs=0.0, -0.1, -0.2 V (principal operating region of the FET). In order to quantify this error, a subroutine in the FORTRAN program FIT1.FOR (Appendix I) was written to calculate the RMS error between the measured and simulated curves for the given voltages. More importantly though, some points on the curve have up to 200% variation between measured and simulated values, which will result in gross inaccuracies if the FET is operated in those regions. Figure (5.2) is an electrical schematic of the internal JFET model in SPICE as well as the

Table (5.1). Principal GaAs MESFET SPICE model equations.

1) SPICE 2 - Schichman and Hodges (1968)

For $V_{ds} < V_{gs} - V_T$ and $V_{ds} > 0$

$$I_{ds} = \beta V_{ds}\left[2\left(V_{gs} - V_T\right) - V_{ds}\right]\left(1 + \lambda V_{ds}\right)$$

For $V_{ds} > V_{gs} - V_T$ and $V_{ds} > 0$

$$I_{ds} = \beta\left(V_{gs} - V_T\right)^2\left(1 + \lambda V_{ds}\right)$$

2) Van Tuyl and Liechti / Curtice (1974 / 1980)

$$I_{ds} = \beta\left(V_{gs} - V_T\right)^2\left(1 + \lambda V_{ds}\right)\tanh\left(\alpha V_{ds}\right)$$

Curtice includes a time delay: $I_{ds}\left[V_{gs}(t - \tau), V_{ds}\right]$

3) Curtice and Ettenberg (1985)

$$I_{ds} = \left(A_o + A_1V_1 + A_2V_1^2 + A_3V_1^3\right)\tanh\left(\alpha V_{out}\right)$$

$$V_1 = V_{in}(t - \tau)\left[1 + \beta\left(V_{dso} - V_{out}\right)\right]$$

4) Statz, Smith, Pucel and Haus (1987)

$$I_{ds} = \frac{\beta\left(V_{gs} - V_T\right)^2}{1 + b\left(V_{gs} - V_T\right)}\left(1 + \lambda V_{ds}\right)\left(1 - \frac{\alpha V_{ds}}{3}\right)^3$$

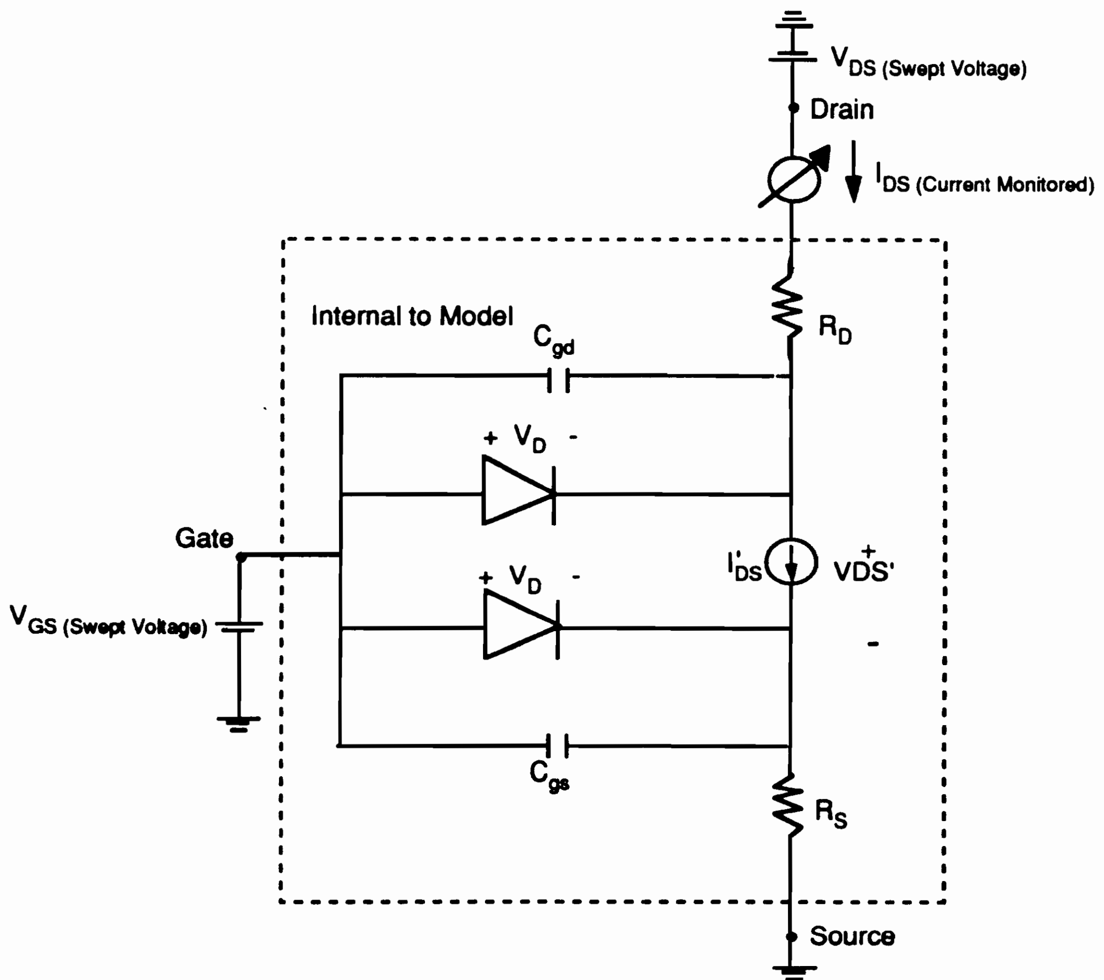


Figure (5.2). Electrically equivalent JFET SPICE model inside dashed line.

external circuitry to sweep Vgs and Vds while monitoring Ids. The Vd, Rd, Rs, Cgd and Cgs terms are set internal to the JFET model (respective values are: 0.72 V, 78 Ω, 78 Ω, 1.5FF, and 15FF). Hence, to compare the simulated Ids with the calculated in Table (5.1), care must be taken to account for the voltage drops due to Rs and Rd. The error in the curve fit due to the limitations of the nonlinear equation is treated as the only noise factor. Three variables: Vt, β, and λ require optimization, but only two (Vt and λ) will undergo the typical Taguchi analysis because β is a multiplication or gain factor (assuming Rs can be set to zero or accounted for). Therefore, to evaluate two variables with three levels each will result in a full factorial of nine combinations or experiments. Since this is such a small number of experiments, orthogonal arrays to limit the number of experiments are not required (Chapter VII will use orthogonal arrays). The challenge now is to optimize Vt, β, and λ to improve the curve fit using the Taguchi methodology. First, the decision has to be made on the number of points to evaluate on the curves. A limited set of data points were chosen to keep the evaluation as simple as possible. These points were chosen to include the principal operating regions of most circuits. The data points, as mentioned previously, consisted of four Vds voltages (Vds=0.5,1.0,1.5,2.0 V) at which each of the three Vgs voltages (Vgs=0.0,-0.1,-0.2 V) were evaluated resulting in a total of twelve data points. The slope and variance can be found (forcing the line through the origin) from equations

(5.1) and (5.2) by plotting the measured Ids versus the calculated Ids as in figure (5.3).

$$SLOPE=(\Sigma Yi*Xi)/(\Sigma Xi*Xi) \quad i=1,12 \qquad (5.1)$$

$$VARIANCE=(1/(n-1))*\Sigma(Yi-SLOPE*Xi)**2 \quad i=1,12 \qquad (5.2)$$

A slope of 1 will give an exact fit between measured and simulated Ids current. The slope will give an indication of how far the mean varies from ideal and the variance will show how far the data varies from the mean. In other words, the slope term will include the variation from target or ideal and the variance term will include the effect of noise factors. Hence, the optimized scaling factor can be found from the deviation of the slope from ideal. These terms can now be incorporated to define a SNR term, which is shown in equation (5.3).

$$SNR=10log((SLOPE**2)/(VARIANCE)) \qquad (5.3)$$

As the variance gets smaller (less effect of noise factors) and the slope gets larger the SNR gets larger. However, the figure of merit isn't just the slope or the variance, but the ratio of the slope to the variance and the slope is squared to eliminate any sign effects. Therefore, the larger the ratio of the slope to the variance the larger the SNR, which means a better setting for the particular variables. The log function

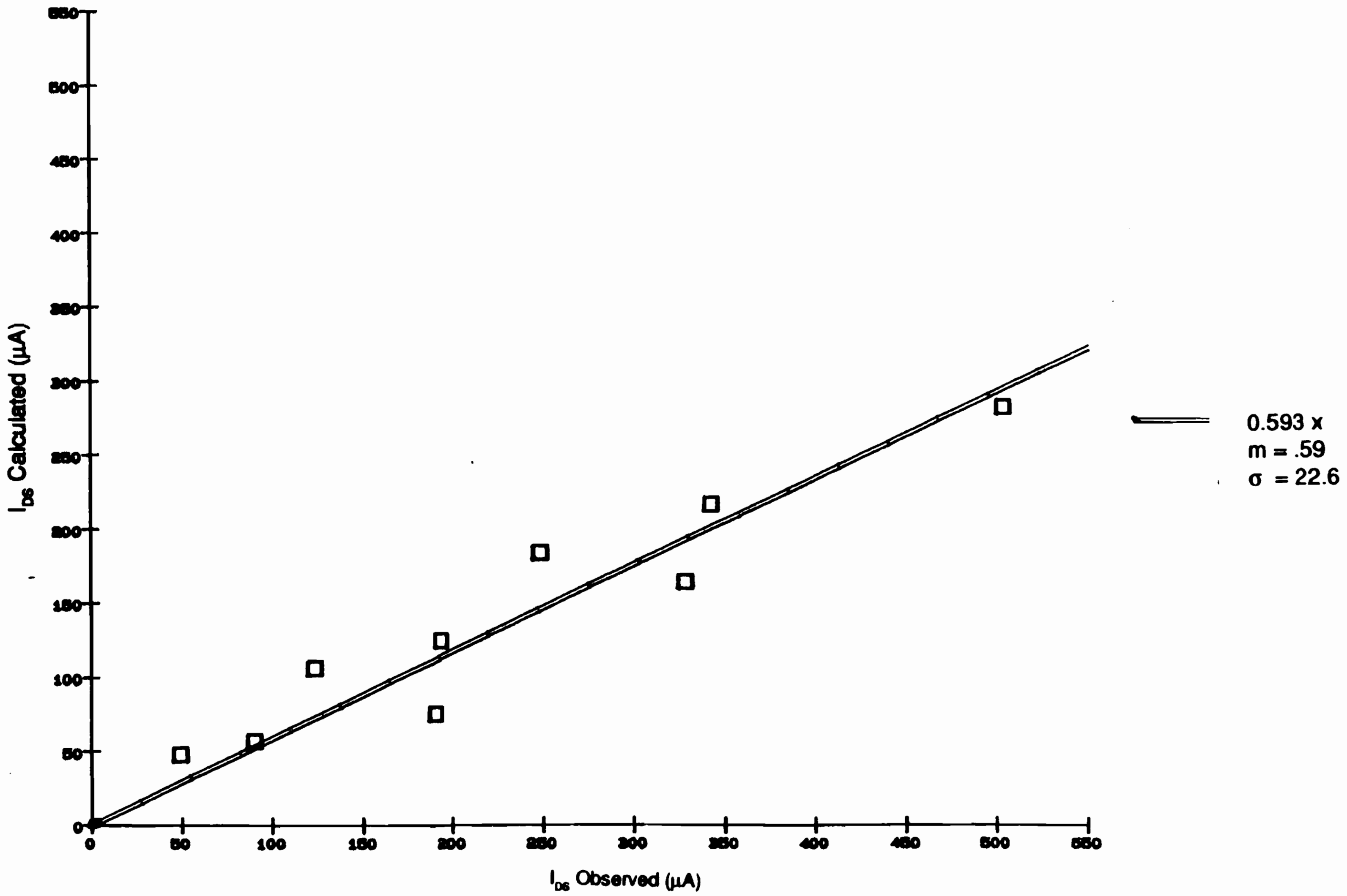


Figure (5.3). Least-squares fit to measured versus simulated Ids for a single experiment.

was used because it gives a larger range of possible values (negative as well as positive). It is critical that the SNR is chosen so the effects of noise are isolated. If this is not done there will be strong interactions among the variables causing the conclusions to be non-optimum. This will be covered in more detail later. A FORTRAN program FIT1.FOR, in appendix I, was developed to automatically calculate the RMS error, slope (mean), sigma (standard deviation), and SNR from the nine experiments of simulated Ids data. This program avoids the time intensive effort of loading the data into a curve fitting program for each experiment to determine the mean and standard deviation. The program reads in the data directly from a simulator output file and a file with the measured data.

5.3 TEST RESULTS

Table (5.2) shows the results of running FIT1.FOR on the first nine simulations or experiments by sweeping Vt from -0.35, -0.40, -0.45 V and λ from 1.0, 1.5 and 2.0. Table (5.3) shows the results of a second pass of running FIT1.FOR with a different set of sweep ranges. Each simulation swept Vds from 0.5 V to 2.0 V in 500 mV steps and Vgs from -0.2 V to 0.0 V in 100 mV steps. The value for β was kept at a nominal setting of 0.001 for all of the simulations. Once the largest SNR was determined (experiment #7 for this case) the mean (1.467) was used to scale all the output currents. It would be more

Table (5.2). Initial pass SNR results for nine experiments.

Experiment #	RMS Error	Mean	Sigma	SNR	λ	V_T
1	22.553	1.011	23.326	-27.258	1.0	-.45
2	78.381	0.789	19.703	-27.945	1.0	-.40
3	113.947	0.593	22.630	-31.638	1.0	-.35
4	71.420	1.249	21.954	-24.902	1.5	-.45
5	18.923	0.979	18.836	-25.683	1.5	-.40
6	75.796	0.738	25.315	-30.701	1.5	-.35
7	130.196	1.467	23.839	-24.219	2.0	-.45
8	46.575	1.155	19.681	-24.629	2.0	-.40
9	43.880	0.875	28.528	-24.266	2.0	-.35

convenient to scale β directly, but β varies non-linearly when the Rs and Rd terms are used in the model of Figure (5.2). In other words, β becomes a function of Vds and Vgs when Rs and Rd are non-zero. As mentioned previously, this method eliminates one variable or factor from the optimization, which geometrically reduces the number of simulations from 27 to 9 (using 3 settings per variable).

The next step in analyzing the SNR is to quantify the effect of λ and Vt with respect to the SNR. This is accomplished by evaluating response curves as shown in Figure (5.4). The λ response curve (leftmost) in Figure (5.4) shows the average SNR values for the three different setting of λ. For instance, the first data point at λ=1 includes the average of SNR=-27.258 dB, -27.945 dB, and -31.638 dB. The purpose is to determine the settings for λ and Vt with the highest SNR or smallest sensitivity to noise, which in our curve fitting example is the variation in the curves from the measured values. Therefore, from the data in Figure (5.4) you would choose λ=2 and Vt=-0.45. However, a maximum is not shown on the curves, which means the optimum setting of the variables λ and Vt could lie outside the evaluated range unless there is a maximum between the evaluated points. Logical choices for the next group of nine experiments are: λ=2.0, 2.5, 3.0 and Vt=-.45, -.50, -.55 V. One variable setting should be consistent with the previous nine groups of experiments to assure continuity of the results. However, prior to expending any

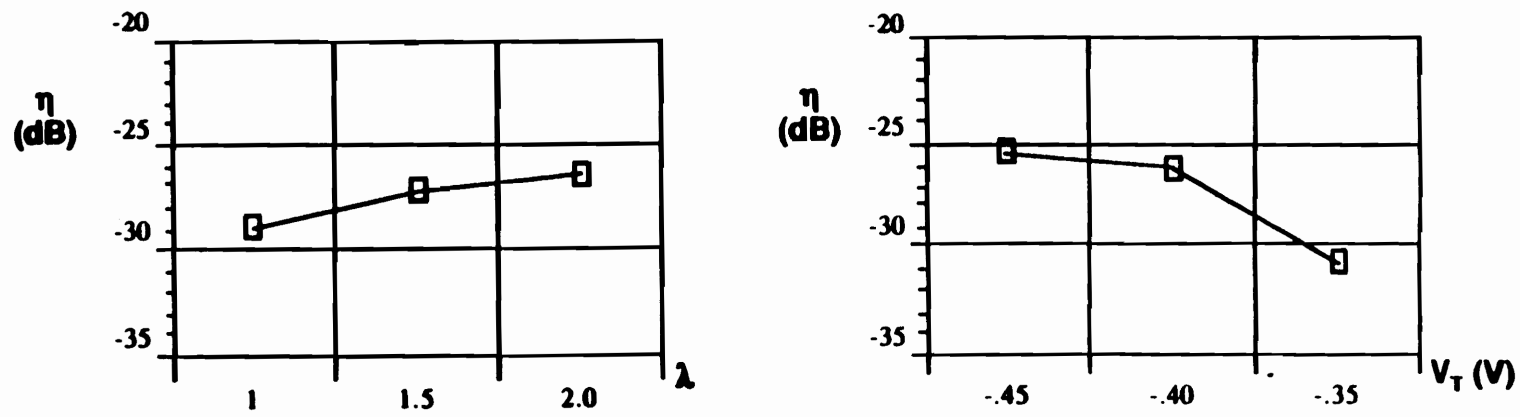
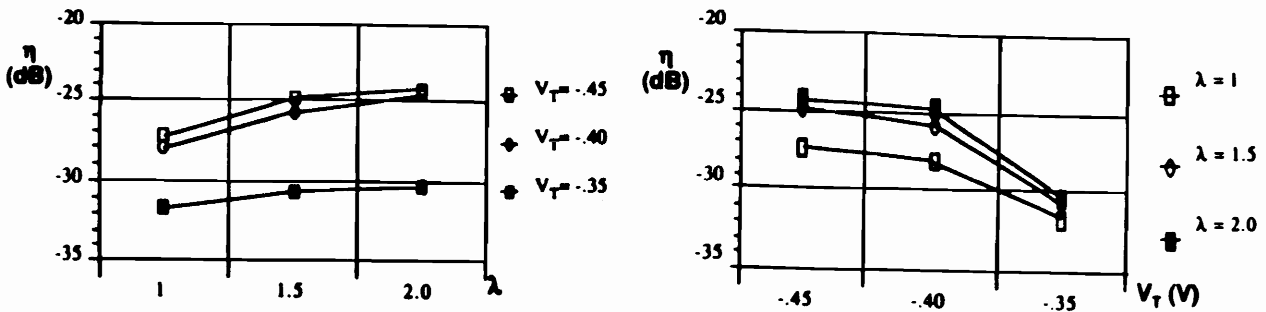


Figure (5.4). Response curves of λ and β for initial pass using SNR.

more time on finding the optimum variable setting, it would be astute to check the assumptions about the SNR equation and verify there are no antisynergistic interactions [14].

Figure (5.5) shows the interaction graphs for the two variables λ and Vt. The λ interaction graph (left most) depicts the variation of λ for the three different settings of Vt and vice-versa for the Vt interaction graph. If the lines are all parallel the variables show no interaction whatsoever, but if the lines intersect they are called antisynergistic and interactions among the variables are present. In the top set of curves in Figure (5.5) the curves are called synergistic interactions in which the curves track in the same direction, indicating the variables are additive and hence the SNR equation is valid to determine the optimum settings of the variables. The concept of additivity means the behavior of the fit is due to the superposition of the individual levels of the variables. In other words, the behavior for the fit can be predicted for any value of the variables; whereas if interactions were present the cross products of the variables would need to be analyzed to understand the results and attain an optimum solution.

The bottom set of curves in Figure (5.5) shows that antisynergistic interactions are formed by using the RMS error instead of the above SNR equation to determine the levels of the parameters. The intersection of the lines shows a clear



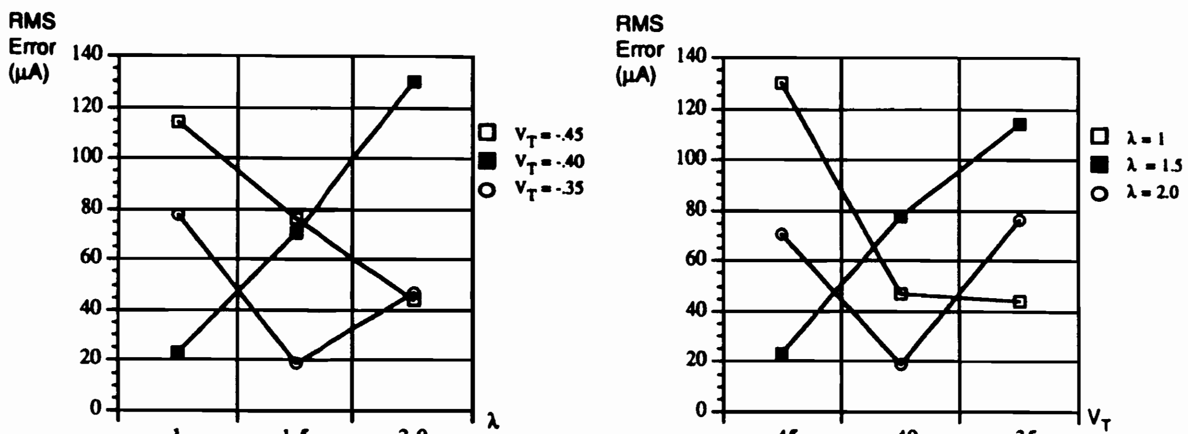


Figure (5.5). Interaction curves of SNR versus RMS error method.

interaction between the λ and Vt parameters which would lead to erroneous results from the response curves as shown in Figure (5.6). In Figure (5.6) the average value of the RMS error is shown for the three different values of the variables λ and Vt. The smaller the value the better for the average of the RMS error because it represents less variability or less sensitivity to noise. However, based on these curves, $\lambda=1.5$ and Vt=-0.40 would be chosen, which is clearly in disagreement with the earlier conclusions. In order to be confident of these original results, both techniques can be compared by determining the RMS error of the fit using the respective set of modeling parameters. In other words, the SNR method will use Vt=-0.45 V and $\lambda=2$ and for the RMS error method will use Vt=-0.40 V and $\lambda=1.5$. Table (5.4) is a summation of the error between measured and simulated I-V curves using five different methods for simulation. Methods (3) and (4) are the results of the aforementioned comparison. From Table (5.4) using the Taguchi SNR from method (3) gives a 56.5% improvement over the parameters extracted directly from the test program with no optimization. More significantly though, using the average of the RMS error to determine the parameters as in method (4) gives a fit which is 8% worse than the fit from using the SNR in method (3). Therefore, the importance of using the SNR in the Taguchi method to eliminate interactions has been shown in the above analysis.

Method (1) in Table (5.4) shows the largest inaccuracy between measured and simulated results is attained by using

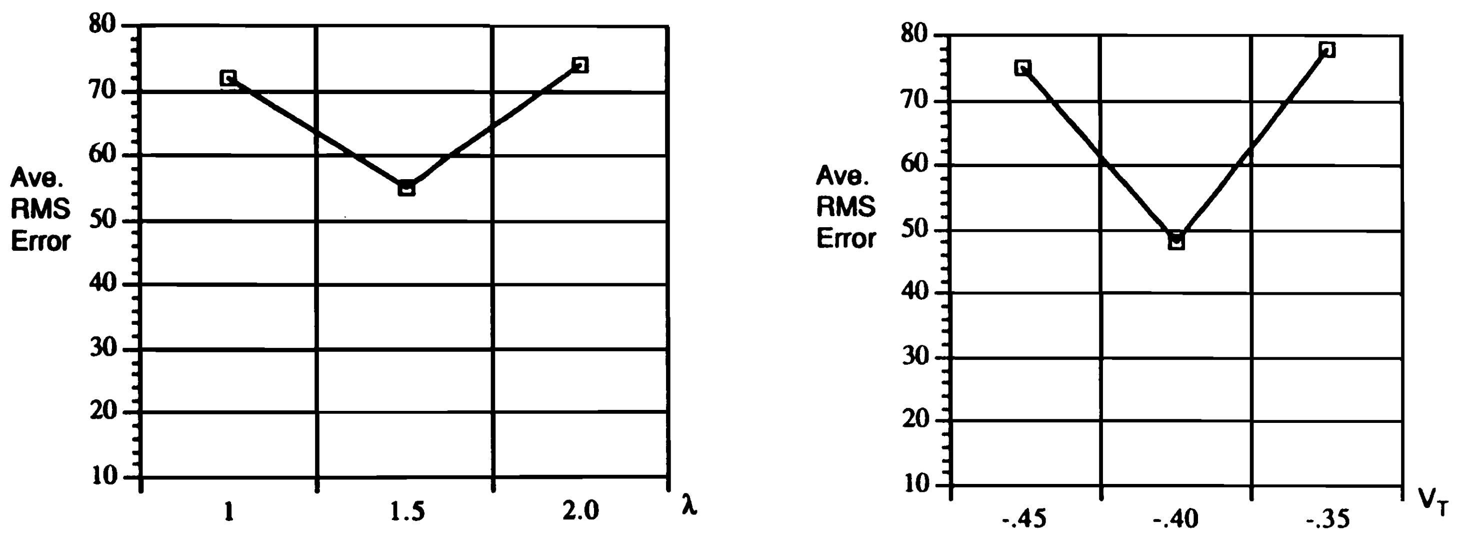


Figure (5.6). Response curves of λ and β for initial pass using RMS error.

Table (5.4). Summary of RMS error between measured and simulated I-V curves using five different simulation methods.

Method	V_T	λ	β	RMS Error From Fit	% Improvement from (1)
(1) Actual measured parameters	-.364	1.04	1.48E-3	35.74 μA	——
(2) Optimized λ & V_T & β using PRECISE optimizer	-.420	1.66	0.840E-3	15.08 μA	57.8%
(3) Optimized λ & V_T & β for initial pass Taguchi method using SNR response curves	-.45	2	1.0E-3* Scale by: 1.467	15.56 μA	56.5%
(4) Optimized λ & V_T & β for initial pass Taguchi method without using SNR	-.40	1.5	1.0E-3* Scale by: 0.979	18.42 μA	48.5%
(5) Optimized λ & V_T & β for Taguchi method using SNR regression of response curves	-.433	2.07	1.0E-3* Scale by: 1.385	13.96 μA	60.9%

12 points on the Curves evaluated: V_{DS} = .5 V, 1 V, 1.5 V, 2.0 V
V_{GS} = 0.0 V, -0.1 V, -0.2 V

*Scale I_{DS} currents by mean from fit.for

the extracted parameters from direct testing of the devices with no optimization. Method (2) is our currently used optimization method at GTC, which gives a 57.8% improvement from method (1). The program required eight iterations to arrive at these optimized values. Method (3) and (4) were explained above and method (5) was determined by solving a second order equation to fit a curve to the measured data points in Figure (5.4) and determining the largest SNR by taking the derivative of the equation. The Vt value will have a maximum value between -.40 and -.45 V if the fit is quadratic. However, in order to verify the second order curve fit is accurate based on the three data points it is good practice to take two additional data points on either side of the maximum value and re-confirm. Furthermore, if the response curve is strictly linear as in Figure (5.4) for λ, additional points should be taken outside this range before fitting a second order equation to the data points and solving for the maximum values. This additional data is shown in Table (5.3) and Figure (5.7). Therefore, by using method (5), a second order curve fit using the Taguchi approach, has given the best improvement in SNR of 60.9% with respect to four other approaches. Figure (5.8) allows a visualization of the degree of improvement between methods 1) and 5).

In summary, five different methods for attaining SPICE parameters to use in analog IC circuit simulation have been evaluated. All four optimization methods significantly

Table (5.3). Second pass SNR results for nine experiments.

Experiment #	Mean	Sigma	SNR	λ	V_T
1	1.467	23.839	-24.219	2.0	-.45
2	1.803	46.092	-28.154	2.0	-.50
3	2.164	76.657	-30.985	2.0	-.55
4	1.669	27.502	-24.338	2.5	-.45
5	2.043	54.079	-28.457	2.5	-.50
6	2.444	89.252	-31.250	2.5	-.55
7	1.857	31.936	-24.707	3.0	-.45
8	2.265	62.587	-28.830	3.0	-.50
9	2.701	102.040	-31.545	3.0	-.55

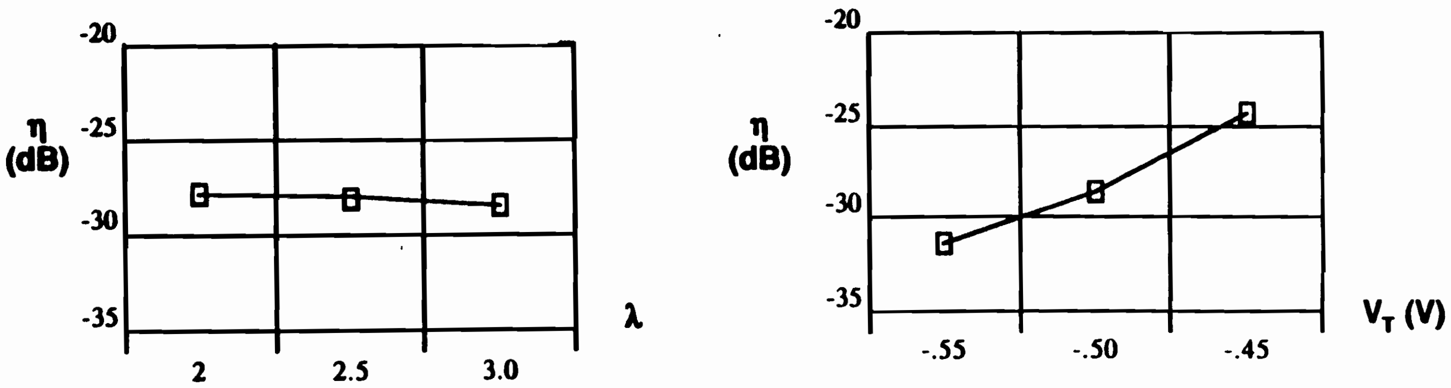
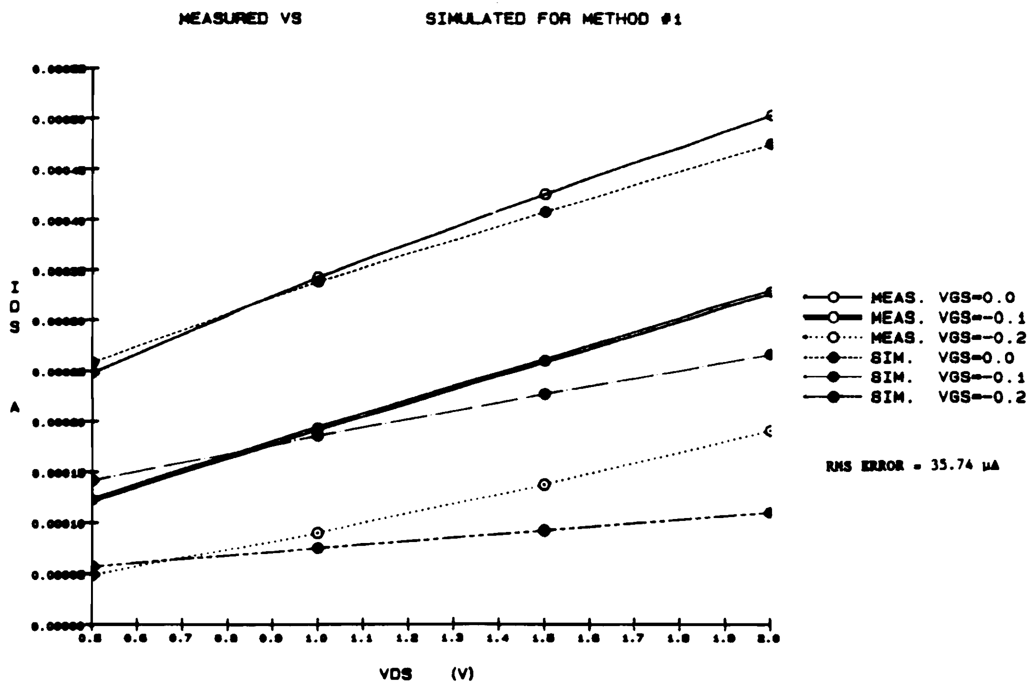


Figure (5.7). Response curves of λ and β for second pass using SNR.



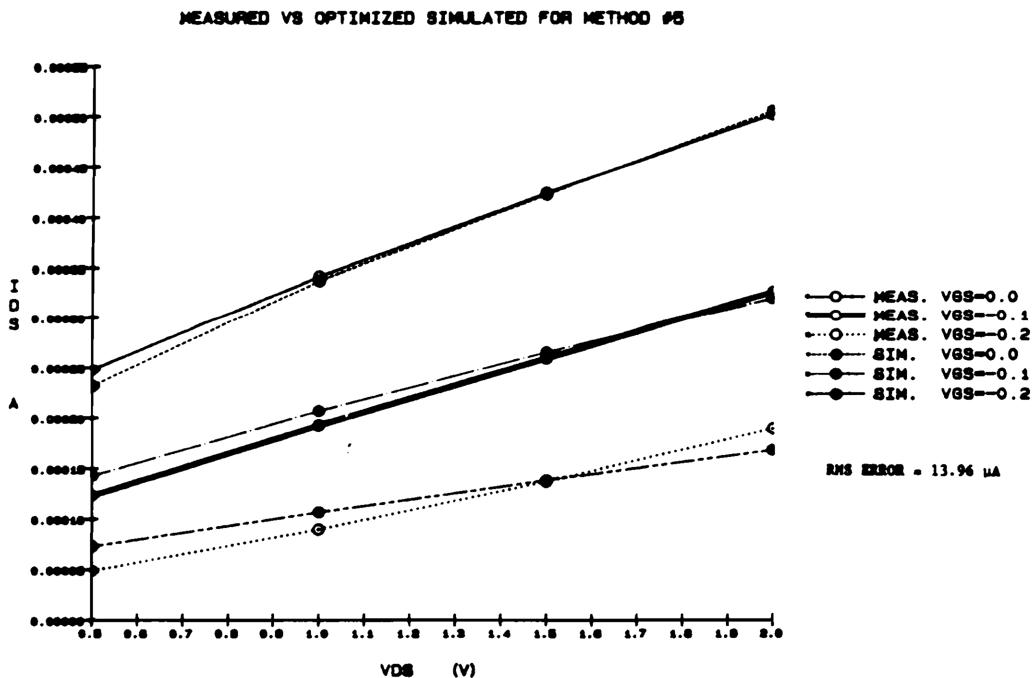


Figure (5.8). Actual comparison of plotted I-V curves method between measured and simulated curves using methods #1 and #5.

improved the accuracy of the simulation versus using the parameters extracted from actual measurements as described in Chapter III. The benefit of using the SNR to limit interactions among the variables to give optimum results was clearly demonstrated. The Taguchi method with no curve fitting to the maximum values gave a SNR similar to the existing commercial optimizer used at GTC. However, by using the Taguchi method and finding the maximum values of the variables through a second order curve fit resulted in the best fit between measured and simulated curves. This is the first known case of applying Taguchi methods to model fitting. Furthermore, a new embellishment to the Taguchi method was introduced by applying a quadratic curve fit to the response curves in order to find the optimum value of the variables. The next chapter will describe the incorporation and modification of these techniques and methods to be useable in an actual manufacturing environment.

CHAPTER VI

TAGUCHI METHODOLOGY IMPLEMENTATION

6.1 INTRODUCTION

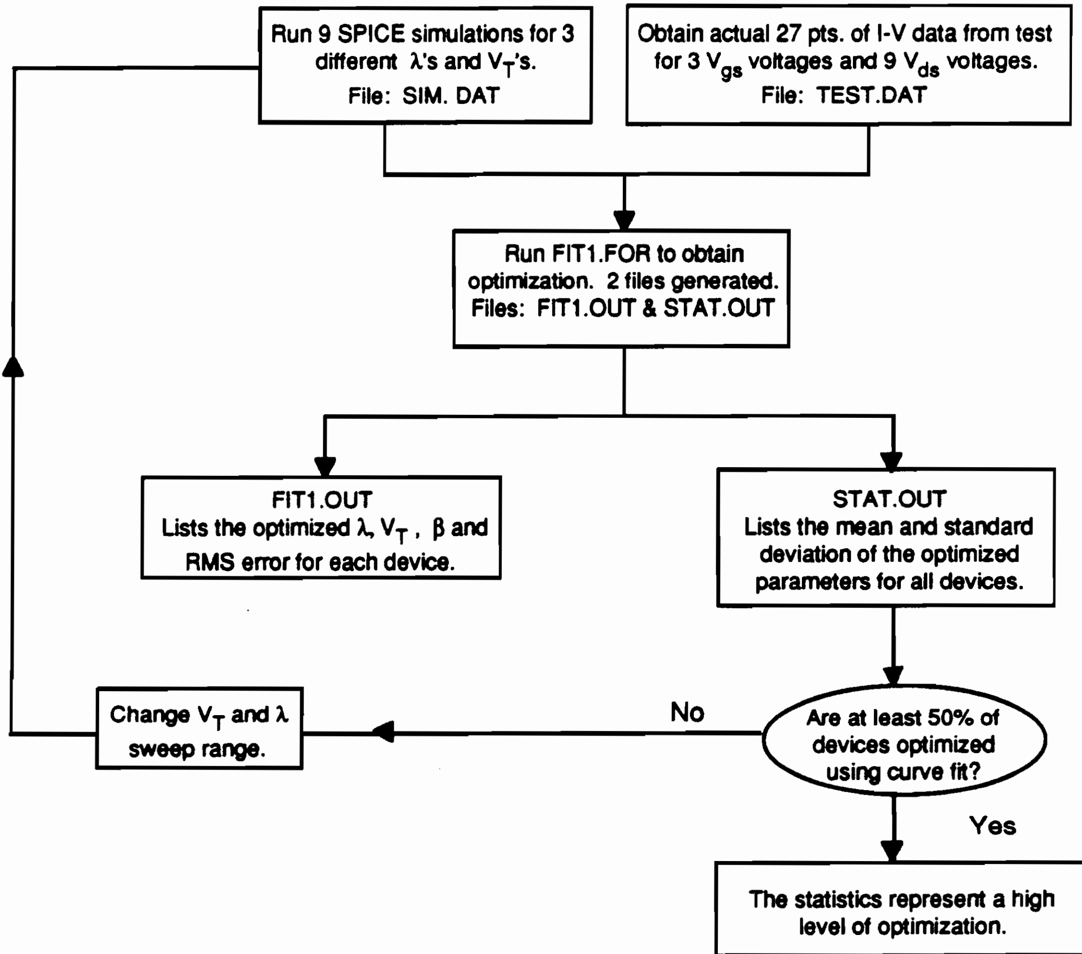
In the previous chapter the Taguchi optimization technique was proven to be an effective way to optimize the GaAs MESFET I-V curves using the JFET model. However, the method was not useable for practical implementation in a manufacturing environment. In order for this program to be used in designing integrated circuits some modifications must be made, such as: using a larger range of I-V curve values, direct uploading of test data into the program, user selectivity of FETs with different threshold voltages, and a methodology to obtain nominal as well as worst case parameters to be used in the circuit simulation. Hence, this chapter will describe the program implementations to make this method a practical tool for any user, which dictates the program and method must be as user friendly or user transparent as much as possible.

6.2 DESCRIPTION OF THE OPTIMIZER PROGRAM

The modified program from the previous chapter is called FIT1.FOR and the FORTRAN code is listed in Appendix I.

Moreover, a flow chart showing a basic functionality and internal file generation is illustrated in Figure (6.1). The first modification to this new program was to extend the number of points on the I-V curve from 12 to 27 to allow a more accurate representation of the curves. In the case of the DFET, the gate-to-source voltage varies from +0.2 V, 0.0 V and -0.2 V (typical Vt is -0.4 V). On the other hand, the EFETs were varied from 0.6 V, 0.4 V and 0.2 V (typical Vt is +0.2 V). The Vds voltage was varied from 0.0 V to 2.0 V in 0.25 V increments to give a total of 27 data points per MESFET. Next, the program was modified to allow the reading of the actual I-V data for up to 32 MESFETS from any given wafer of EFETs or DFETs (the program prompts the user for EFET or DFET as well as the total number of devices to evaluate). The program calculates the SNR for 9 Taguchi experiments as before and prints out the maximum SNR experiment as well as the given values for Vt, λ, and β. Furthermore, the program calculates the RMS error between the actual measured data and the new simulated data based on the given SPICE parameters. The program avoids doing a SPICE simulation by calculating the Ids current from equation (5.1). However, in order to use the results of these equations in an actual SPICE simulation, the Rs and Rd values must be set to zero (some circuit simulations might require a small non-zero number that will not effect the current such as 0.000001-ohm). This is done to allow the new β to be used directly in the equation (found from the slope of the measured Ids vs. simulated value). If the actual Rs and

Figure (6.1). Flow chart for the optimization of I-V data.



Rd values are used, then β becomes non-linear with respect to Ids and can't be used to compensate for the variation of the measured Ids versus simulated. This is due to Vgs and Vds becoming a function of Ids if Rs and Rd are non-zero.

Hence, Rs and Rd are not required in the simulation process, even though from a device physics viewpoint they indeed exist. However, the program FIT1.FOR has been modified to address the cases where Rs and Rd values are required in the simulation to replicate non-symmetrical effects between the source and drain. This modified version of FIT1.FOR is called FITINT.FOR and is listed in Appendix II. This program calculates the voltage drop across both Rs and Rd and uses these values to derive the intrinsic Vds and Vgs values. These intrinsic values are then used to calculate the Ids current in Table (5.1). Therefore, the actual Rs and Rd values for each measured FET must be uploaded into the program with the measured Ids currents.

In the preliminary writing of program FIT1.FOR, the RMS error was only defined between the parameters using the maximum SNR from the experiments and the actual measured data. Now, the same is done in FIT1.FOR using the quadratic curve fit of the response graphs to find the optimum values of Vt, λ, and β, as described in Chapter V. Some additional enhancements in the curve fit include checks to make sure the optimum point on the curve is a maximum and not a minimum for both λ

and β. Furthermore, the optimum point is verified to fall between the sweep points of the response curves. Erroneous results will occur if these checks are not made. After the maximum point on the quadratic curve is found for both λ and Vt, a new value for β is found and the RMS error is calculated. All this data, including the coefficients of the quadratic equation are printed for each device, as shown in Table (6.1). In addition, a file is created that calculates the mean and standard deviation of the SNR, Vt, λ, β, and RMS error for all devices, as shown in Table (6.2) using the best values from the actual SNR as well as the quadratic curve fit. This data can be utilized to determine nominal and worst case parameters for the wafer tested. Merging this data to find the best values from the actual experiments and the curve fit isn't useful because the curve fit gave a smaller or equal RMS error in 39 out of 40 devices tested.

Table (6.3) shows the RMS error from a wafer of DFETs on GTC 312-1-6 using four methods to simulate the I-V curves and compare directly with the measured data. The first column shows the results of 6 devices using the extracted DC parameters from test. The next column includes the best results of the actual Taguchi experiments. The third column of data uses the Taguchi curve fit approach and the last column uses the PRECISE optimizer from Electrical Engineering Software. Table 6.4 shows the RMS error as above for EFETs from wafer 312-1-6. Also, the tables include the overall wafer average for both

Table (6.1). Optimization output data for one device in file FIT.OUT.

```
--------------------------------------------------------------------------

DEVICE# 02-01

EXPERIMENT #   LAMBDA    Vt      RMS       MEAN     SIGMA     SNR
      1         0.10    0.05     89.254   0.693    28.539   -32.294
      2         0.10    0.15    150.334   0.456     8.827   -25.730
      3         0.10    0.25    201.788   0.270     8.573   -30.044
      4         0.20    0.05     70.603   0.769    30.682   -32.023
      5         0.20    0.15    136.616   0.506     7.292   -23.177
      6         0.20    0.25    193.779   0.299     8.148   -28.716
      7         0.30    0.05     54.881   0.844    34.785   -32.297
      8         0.30    0.15    123.101   0.555     9.016   -24.210
      9         0.30    0.25    185.808   0.328     8.849   -28.627

MAX.SNR=-23.177  VT= 0.15  LAMBDA=0.200E+00  BETA=0.198E-02
THE RMS ERROR USING THESE VALUES IS (uA):   15.75603


THE COEFFICIENTS OF A QUADRACTIC EQUATION Ax**2+Bx+C=y
TO THE RESPONSE CURVES WHERE x=step size and
dx=-B/2A (MAXIMUM POINT ON CURVE FROM AVE. x) ARE:

    PARAMETER      A          B          C          dx
       VT      -0.63E+03   0.15E+02  -0.24E+02   0.12E-01     LOCAL MAX
     LAMBDA    -0.90E+02   0.49E+01  -0.28E+02   0.27E-01     LOCAL MAX


QUADRACTIC CURVE FIT: VT= 0.162 LAMBDA=0.227 BETA=0.204E-02 SNR=-23.615
THE RMS ERROR USING THESE VALUES IS (uA):   13.77793
THE SNR USING THE NEW BETA IS:  -22.94759

--------------------------------------------------------------------------
```

Table (6.2). Wafer averaged results for 31 devices in STAT.OUT.

312-2-6 EFET RUN #3

THE AVERAGE AND SIGMA FROM THE ACTUAL EXPERIMENTS:

	SNR	VT	LAMBDA	BETA	RMS	#DEVICES
AVERAGE	-24.13	0.14	0.19	0.24E-02	17.78	31
SIGMA	2.27	0.30E-01	0.25E-01	0.29E-03	4.05	

THE AVERAGE AND SIGMA FROM THE QUAD. CURVE FIT:

	SNR	VT	LAMBDA	BETA	RMS	#DEVICES
AVERAGE	-23.38	0.136	0.194	0.23E-02	14.53	30
SIGMA	0.73	0.21E-01	0.22E-01	0.14E-03	1.21	

Table (6.3). Curve fit results for DFETs from wafer 312-1-6.

Device #	RMS Error Between Measured and Simulated			
	Parameters from DC Test	Best Taguchi Exp.	Taguchi Curve Fit	Precise Optimizer*
2-1	39.7	43.7	29.3	29.9
3-1	46.0	49.2	36.4	33.6
4-1	59.8	39.8	28.0	26.6
5-1	104	37.2	30.9	27.9
2-2	32.6	40.0	29.5	28.1
3-2	40.9	44.0	30.9	31.6
Average	53.8 μA (6)	42.3 μA (6)	30.8 μA (6)	29.6 μA (6)
Standard Deviation	26.2 μA	4.24 μA	2.94 μA	2.62 μA
Average	53.7 μA (18)	43.2 μA (31)	31.2 μA (18)	———
Standard Deviation	20.1 μA	5.99 μA	4.80 μA	———

( ) = Number of Devices included in Statistics

* Electrical Engineering Software

Table (6.4). Curve fit results for EFETs from wafer 312-1-6.

Device #	RMS Error Between Measured and Simulated			
	Parameters from DC Test	Best Taguchi Exp.	Taguchi Curve Fit	Precise Optimizer*
2-1	153	19.3	16.7	22.5
3-1	113	15.0	13.6	18.1
4-1	132	16.8	16.9	22.6
1.2	118	14/9	12/7	17.5
2-2	131	14.8	14.8	20.4
3-2	160	21.6	18.5	25.4
Average	134.5 µA (6)	17.1 µA (6)	15.5 µA (6)	21.1 µA (6)
Standard Deviation	26.2 µA	4.24 µA	2.94 µA	2.62 µA
Average	——	16.2 µA (28)	15.2 µA (22)	——
Standard Deviation	——	2.5 µA	1.98 µA	——

( ) = Number of Devices included in Statistics

* Electrical Engineering Software

Taguchi techniques because the program automatically calculates the mean and standard deviation. However, to include this data for the DC test extracted parameters and Precise optimizer would be very time intensive because a simulation needs to be run interactively for each device and multiple files opened and edited. Hence, a smaller subset of the data was chosen and the mean and standard deviation of this subset were compared with the data from the entire wafer using the two Taguchi techniques to make sure this subset is indicative of the behavior of the larger sample. This assumption appears justified for the PRECISE optimizer data since the coefficient of variance (standard deviation divided by the mean) as well as the absolute value of the mean and standard deviation are similar using the Taguchi methods and PRECISE optimizer. On the other hand, this is not true for the DC extracted parameters because of the much larger coefficient of variance and standard deviation. Therefore, the DC extracted parameters were evaluated for all 18 devices of DFETs. The results in Table (6.3) show the mean to be within 0.2% for the 6 device sample versus the 18 device sample. Hence, the 6 device sample is indicative of the larger sample, which make the results shown in Table (6.3) justifiable.

Table (6.5) shows the improvement in the curve fit using the two Taguchi methods and PRECISE optimizer compared with the fit using the extracted DC test parameters. These results show an average improvement of 66% using the Taguchi quadratic curve fit method over the directly extracted parameters from

Table (6.5). Improvement in time and accuracy using Taguchi optimizer from parameters extracted in DC test.

Wafer	FET Type	Best Taguchi Experiment	Taguchi Curve Fit	Precise Optimizer
312-1-6	DFET	21%	43%	45%
312-1-6	EFET	87%	89%	84%
Average	———	54%	66%	65 %
CPU Time for 31 Devices	———	———	31.6 s*	753.6 s

* Using four iterations of λ and V_T sweep ranges to maximize number of devices used in curve FIT.

DC test. The PRECISE optimizer showed an average improvement of 65% over the directly extracted parameters from DC test. Therefore, significant improvement in the fit between measured and simulated I-V curves can be achieved by utilizing different optimization methods. The accuracy of the Taguchi quadratic curve fit method is comparable to the PRECISE optimizer, which uses the Levenberg-Marquardt algorithm. However, the simulation time is twenty-four times less for 31 devices using the Taguchi approach as shown in Table (6.5) and using a Digital Equipment Corporation VAX 6300 with no other users or batch programs executing. For example, using the Taguchi method, the CPU time to obtain the SPICE simulated values for the nine experiments is 5.62 s and the CPU run time for the FORTRAN program is 2.29 s. Moreover, the CPU time doesn't change much by adding additional devices because only one SPICE simulation is required per wafer and the FORTRAN program has no time intensive algorithms. The PRECISE optimizer takes 24.3 s for 5 iterations of 1 device and the time is additive for each device. The much larger simulation time is due to running a SPICE simulation for each iteration of a device, whereas the Taguchi method requires only nine simulations for an infinite number of devices per wafer. The last significant point with regard to the Taguchi quadratic curve optimizer is the number of required changes in the sweep ranges of Vt and λ to obtain the largest number of devices for the curve fit. The above results were obtained using 4 iterations for EFETs and DFETs. In the case of the DFET 18 of

31 devices (58%) were fitted to the quadratic curve fit and for the EFET 22 of 27 devices (79%). The output file (FIT1.O-UT) also lists the reasons for a device failing to be fitted to the quadratic curve fit. For instance, Both Vt and λ might not be local maximums, or the maximum point might be out of the response curves sweep range. The second output file (STAT.OUT) lists the total number of devices which were fitted to the quadratic curve. Experience has shown if the curve fit on the majority of devices is out of the sweep range, then the sweep range settings are to small. Conversely, if the majority of "dx's" (minimum or maximum point on the quadratic curve) are a global minimum for a particular variable, then the sweep range for the variable is probably too large and should be reduced. Another possibility is to offset the sweep window so it is in the center of the majority of devices that were found from the actual experiments in FIT1.OUT. Table 6.6 shows the progressive optimization of three successive runs from the EFET data of wafer 312-2-6. Initially, only 45% of the total devices were fitted to the curve for the given Vt sweep ranges. Since the majority of outliers to the curve fit were due to Vt being a local minima, the window to the sweep range was offset by +50 mV (this brings the center of the window in line with the mean of Vt from the actual data). Hence, in offsetting the Vt window by a +50 mV resulted in an additional 23% of devices being fitted to the quadratic curve as shown by EFET run #2 in Table (6.6) The majority of the 10 outliers were due to Vt being outside the sweep range. Hence, the sweep

Table (6.6). Typical optimization progression.

312-2-6 EFET RUN #1

THE AVERAGE AND SIGMA FROM THE ACTUAL EXPERIMENTS:

	SNR	VT	LAMBDA	BETA	RMS	#DEVICES
AVERAGE	-23.12	0.13	0.19	0.23E-02	15.73	31
SIGMA	1.20	0.25E-01	0.25E-01	0.17E-03	1.73	

THE AVERAGE AND SIGMA FROM THE QUAD. CURVE FIT:

	SNR	VT	LAMBDA	BETA	RMS	#DEVICES
AVERAGE	-23.65	0.112	0.172	0.23E-02	14.97	14
SIGMA	0.62	0.11E-01	0.19E-01	0.99E-04	1.02	

* VT SWEEP RANGE = 0.05/0.10/0.15
* MAJORITY OF OUTLIERS DUE TO VT LOCAL MINIMUM
* 45% OF DEVICES FITTED TO THE QUADRATIC CURVE
* NEED TO CHANGE SWEEP WINDOW

312-2-6 EFET RUN #2

THE AVERAGE AND SIGMA FROM THE ACTUAL EXPERIMENTS:

	SNR	VT	LAMBDA	BETA	RMS	#DEVICES
AVERAGE	-22.91	0.13	0.19	0.23E-02	15.46	31
SIGMA	1.25	0.32E-01	0.25E-01	0.19E-03	1.72	

THE AVERAGE AND SIGMA FROM THE QUAD. CURVE FIT:

	SNR	VT	LAMBDA	BETA	RMS	#DEVICES
AVERAGE	-22.96	0.146	0.201	0.23E-02	13.83	21
SIGMA	0.61	0.18E-01	0.16E-01	0.96E-04	0.97	

* VT SWEEP RANGE = 0.10/0.15/0.20
* MAJORITY OF OUTLIERS DUE TO VT OUT OF SWEEP RANGE
* 68% OF DEVICES FITTED TO THE QUADRATIC CURVE
* NEED TO INCREASE SWEEP RANGE

312-2-6 EFET RUN #3

THE AVERAGE AND SIGMA FROM THE ACTUAL EXPERIMENTS:

	SNR	VT	LAMBDA	BETA	RMS	#DEVICES
AVERAGE	-24.13	0.14	0.19	0.24E-02	17.78	31
SIGMA	2.27	0.30E-01	0.25E-01	0.29E-03	4.05	

THE AVERAGE AND SIGMA FROM THE QUAD. CURVE FIT:

	SNR	VT	LAMBDA	BETA	RMS	#DEVICES
AVERAGE	-23.38	0.136	0.194	0.23E-02	14.53	30
SIGMA	0.73	0.21E-01	0.22E-01	0.14E-03	1.21	

* VT SWEEP RANGE = 0.05/0.15/0.25
* 97% OF DEVICES FITTED TO THE QUADRATIC CURVE

range was increased 100 mV in run #3 resulting in an additional 29% of devices being fitted from run #2. The overall total devices being fitted to the quadratic curve was 30 out of 31 (97%) after three optimization runs. These general rules are useful as a guideline when becoming familiar with the optimization method.

6.3 STATISTICAL MODELING

Much research has been done regarding statistical modeling of IC processes and simulations for digital and analog circuits [26-33]. However, incorporating these techniques into our current manufacturing process is beyond the scope of this work. The traditional approach for evaluating digital VLSI circuit performance as it relates to process is to combine the process and design parameter sets into two distinct groups of fast and slow performance sets [31]. This method does not take into account the correlation between the parameters and can result in significant performance variations. Gradient analysis overcomes these shortcomings by accounting for the correlation between design or process parameters and it gives a more accurate distribution of worst-case and best-case performance. Furthermore, gradient analysis avoids the very complex approach sometimes used in analog IC yield methods, which evaluates statistical distributions of all the performance parameters.

The previous optimization method at GTC was to average all the I-V curve data and optimize that single set of data using the PRECISE optimizer. This was done because it was impractical to optimize each device due to the amount of time it would take. However, the Taguchi optimizer approach is to independently optimize each device and calculate the average and standard deviation for the sum of all optimized devices. Table (6.7) shows the three columns of data for two DFET wafers processed from different lots. The actual DC test data is shown in column A followed by the average maximum SNR calculated for each device. Column C shows the maximum SNR calculated from the average I-V curves from all devices. The maximum SNR from the actual Taguchi experiments was used instead of the quadratic curve fit values so the sample size could remain identical to make the comparison fair. No standard deviation for column C is shown because only a single set of I-V data is used. The data from wafer 312-1-6 shows relatively close agreement between columns B and C with an average variation to all three parameters being 22%. However, to get an idea of the variation of the data you are constrained to using the standard deviation of the DC test data from column A (column B would be unavailable). The difference between standard deviations of the three parameters from columns A and B of wafer 312-1-6 is 47%. The variations from wafer 312-2-6 is more significant, with a variation in the averages of the three parameters between columns B and C to be 36% and the variation between standard deviations to be 64%.

Table (6.7). Optimizing each device on wafer versus optimizing the average of all devices.

Parameter	Wafer 312-1-6 DFET			Wafer 312-4-6 DFET		
	DC Test Data (A)	Ave. Max. SNR FOR @ Device (B)	Max. SNR Using Wafer Ave. Data (C)	DC Test Data (A)	Ave. max. SNR For @ Device (B)	Max. SNR Using Wafer Ave Data (C)
VT	-.40 (.032)	-.42 (.037)	-.40	-.43 (.032)	-.47 (.053)	-.50
λ	.50 (.09)	0.20 (.20)	0.1	.36 (.06)	.20 (.24)	.01
β (E-3)	1.95 (.10)	1.9 (.35)	2.11	1.80 (.07)	1.7 (.32)	1.83
RMS Error (μA)	——	43.18 (5.99)	43.22	——	51.6 (6.9)	53.6
# Devices	31	31	31	32	32	32

( ) = Standard Deviation

Hence, a statistical nominal and worst-case parameter extraction method based on the average and standard deviations of the model parameters will give significant differences between optimizing each device of the wafer or averaging all the raw data and optimizing that single set of data. Moreover, the previous method used at GTC of optimizing a single set of data averaged over the entire wafer will give a much more optimistic set of worst-case SPICE parameters because standard deviations in the parameters are so much smaller.

Now that the model parameters for each device are optimized and the gaussian statistics calculated; how do you calculate the worst-case parameters for that wafer? One method would be to take all possible combinations or a full factorial of all the three optimized parameters (Vt,Lambda,Beta) and vary them each an equal amount of standard deviations. This would result in eight sets of parameters and none of them would be realistic because the parameters are not totally independent. The approach would result in very pessimistic results as pointed out in [34]. Hence, a method of incorporating the correlations between parameters as in [31] was evaluated. First, each group of optimized parameters are plotted against each other. Figure (6.2) shows Lambda vs. Beta, Figure (6.3) shows Vt vs. Beta, and Figure (6.4)shows Vt vs. Lambda. The correlation coefficients are found from the slope of the line of the least-squares fit of the data. The data represents a total of 79 depletion-mode FETs from three

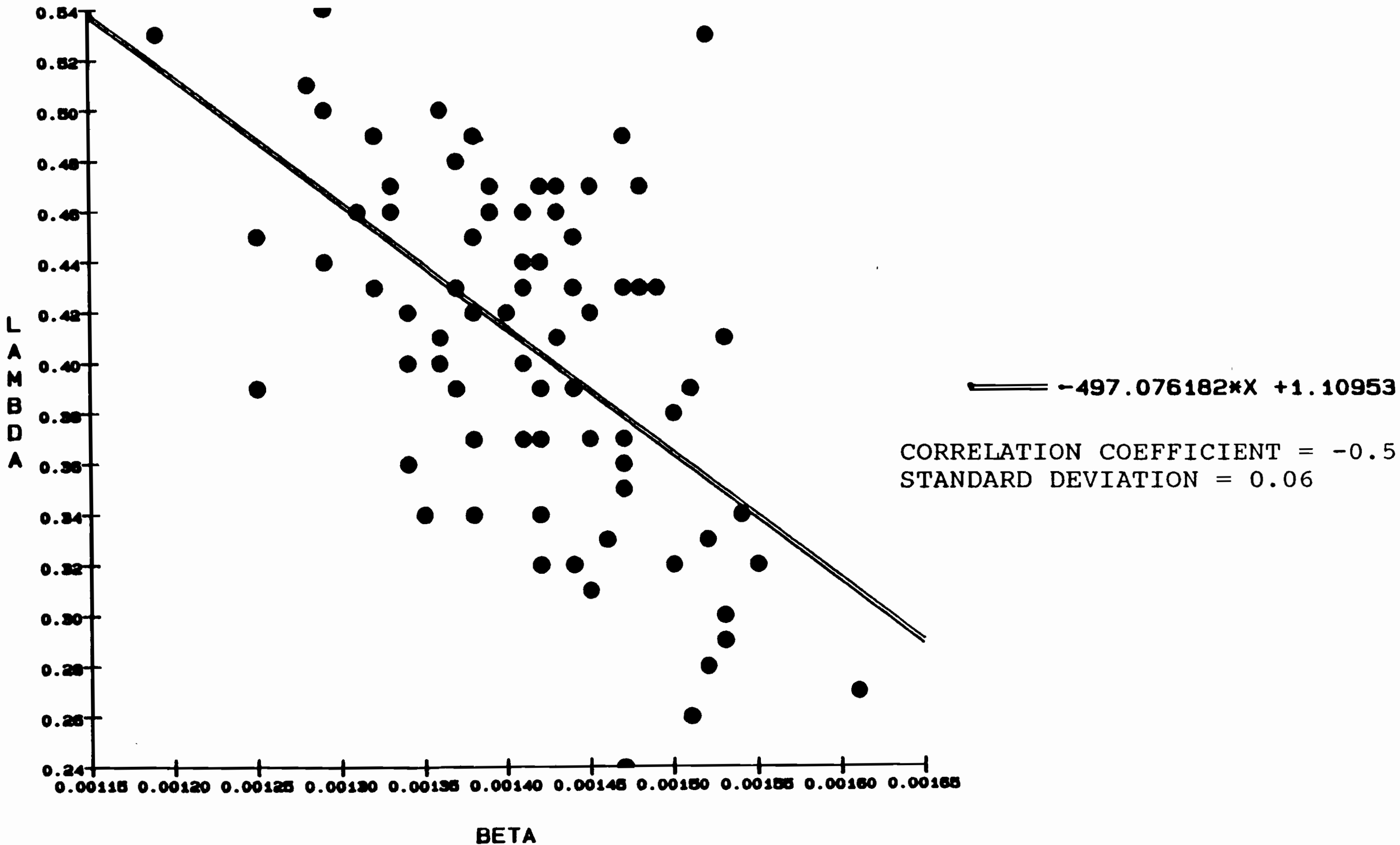


Figure (6.2). Correlating λ and β on 79 devices from 3 wafers.

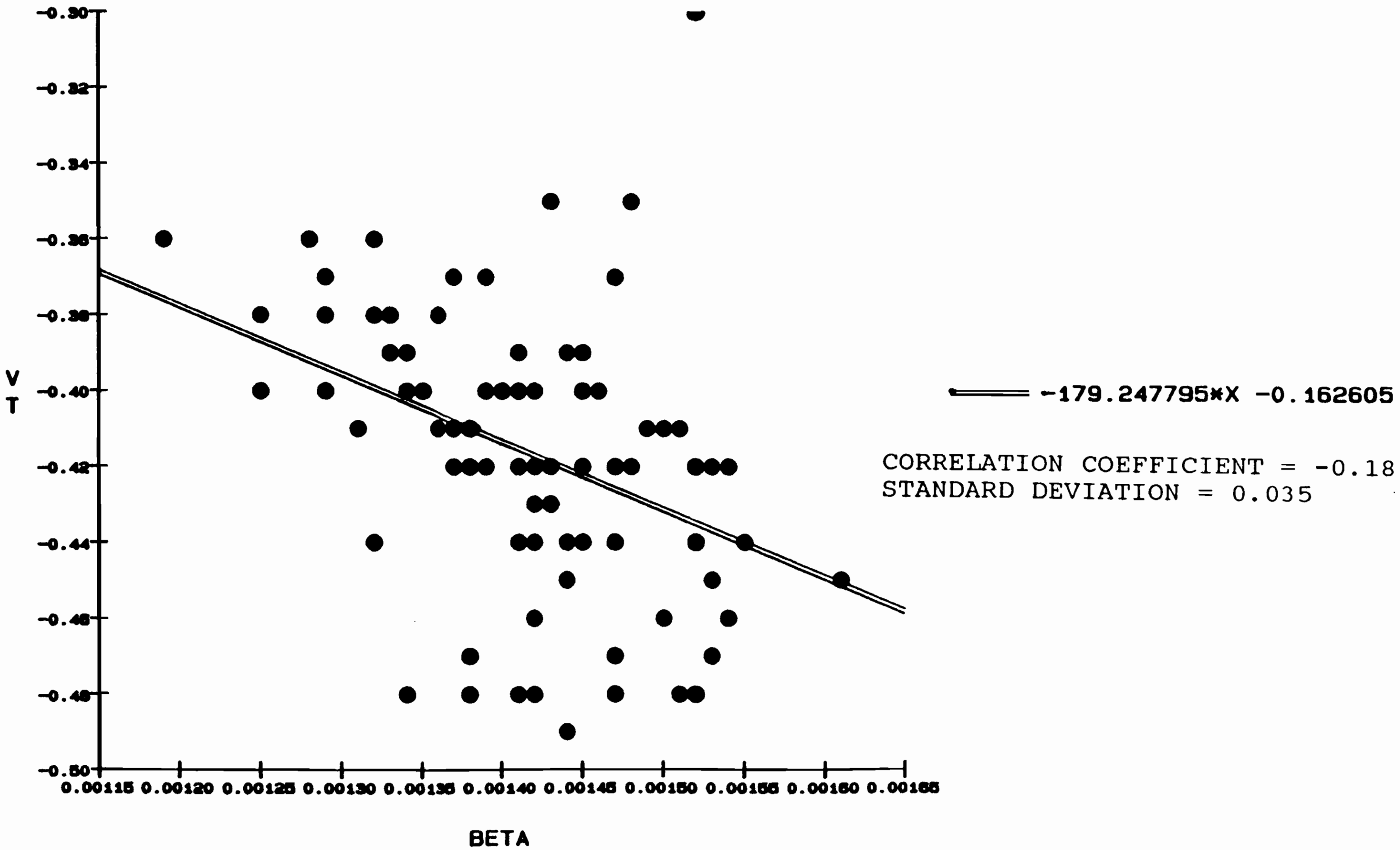


Figure (6.3). Correlating Vt and β on 79 devices from 3 wafers.

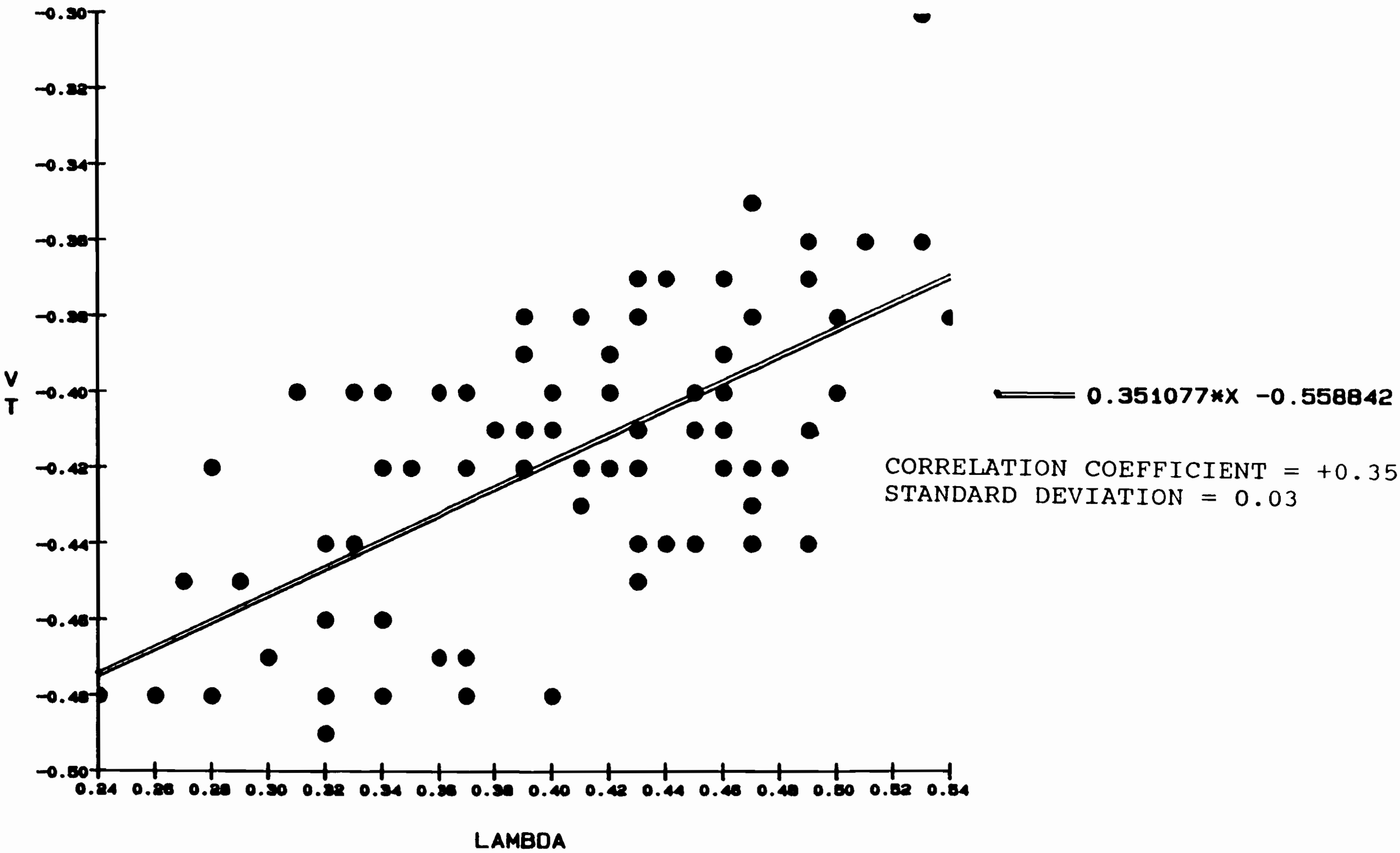


Figure (6.4). Correlating Vt and λ on 79 devices from 3 wafers.

different wafers.

Now that the correlation between parameters are known a single parameter can be selected and varied, with the remaining two parameters correlated. The threshold voltage was the parameter chosen because it is a physically derived and well understood parameter. Hence, Vt was varied +/- one, two, and three standard deviations and Lambda and Beta were correlated accordingly. It will be up to the designer to tradeoff the amount of variation in the parameters they want to design around. A synopsis of these results as presented to a designer are shown in Table (6.8).

The drawback to the previously demonstrated method is the low value of the correlation coefficients. Anything less than 0.1 is considered uncorrelated, while anything above 0.5 is considered correlated and anything between 0.1 and 0.5 quasi-correlated [31]. Therefore, the data presented above would be in the quasi-correlated domain and because of the low correlation will give less than optimum results.

Hence, another method was evaluated to account for the realistic variations between the variables. This technique selects the actual devices from the curve fit optimization that are closest to the +/- 3σ limits of the wafer average. The curve fitting parameters are 100% correlated because they come from an actual device. The I-V curves are then plotted

Table (6.8). Summary of worst-case DFET parameters using the correlation coefficients and gaussian distribution.

GTC JFET SPICE PARAMETERS FOR 10 μm X 0.4 μm MESFETS

Date: January 31,1992

Averaged from 312-1-6, 312-2-6, 312-4-6 using Taguchi optimizer

FET TYPE	VT	BETA	LAMBDA	RS	RD	IS	PB	CGS	CGD
DFET									
mean	-0.417	1.43E-3	0.40	1E-6	1E-6	1E-15	0.72	15FF	1.5FF
ST.DEV.	29mV	0.07E-3	0.05	-	-	-	-	-	-
EFET									
mean	+0.115	2.30E-3	0.19	1E-6	1E-6	1E-15	0.72	6FF	6FF
ST.DEV.	20mV	0.04E-3	0.02	-	-	-	-	-	-

Note:

1) Eventhough the Rs and Rd values are approximately 100 Ω in a physical sense they were set to 1 μΩ so the optimizer could compare the drain and gate voltages measured in test directly with a simulated model. If the measured Rs and Rd values are used then the measured drain and gate voltages must be adjusted to represent the internal Vds and Vgs nodes, which accounts for the Rs and Rs voltage drops.

2) In order to model a dual gate EFET used in AOI gates etc.; use two EFETs connected together in a subcircuit.

DFET Correlation Equations:
VT=0.35*LAMBDA-0.56
VT=-179*BETA-0.16
LAMBDA=-497*BETA+1.10

Worst Case DFET parameters by Varying VT and Correlating LAMBDA and BETA:

+1σ:	VT=-0.388	BETA=1.26E-3	LAMBDA=0.487
-1σ:	VT=-0.446	BETA=1.59E-3	LAMBDA=0.322
+2σ:	VT=-0.359	BETA=1.10E-3	LAMBDA=0.570
-2σ:	VT=-0.475	BETA=1.75E-3	LAMBDA=0.240
+3σ:	VT=-0.330	BETA=0.94E-3	LAMBDA=0.650
-3σ:	VT=-0.504	BETA=1.91E-3	LAMBDA=0.157

using the curve fitting parameters of the actual measured devices from three wafers. These optimized, simulated I-V curves for each wafer are shown in Figures (6.5) to (6.7). These plots are compared against each other and the wafer with the largest spread in I-V curves is chosen to represent the +/- 3σ model parameters. The parameters from Figure (6.6) were chosen because it contained five of the six possible curve limits for the three gate biases. In other words, each Vgs or gate bias has an upper and a lower limit and Figure (6.6) had 83% of these upper and lower limits. Table (6.9) lists the actual +/- 3σ model parameters as well as the nominal parameters. In this case, it is apparent which wafer to choose in order to obtain the worst case FET parameters. However, if each wafer had two upper or lower limits, then the choice of parameters becomes more difficult. This uncertainty can be rectified with a larger sample of wafers.

6.4 APPLICATION TO OTHER MODELS

Some of the principal GaAs model equations were listed in Table (5.1). In order to use these other models, which have inherent improvements associated with GaAs as pointed out in Chapter II, the existing Taguchi optimizer for a JFET model must be modified. The JFET model contains 3 parameters being optimized: Vt, β, and λ, which requires 9 (3 raised to the n-1 power where n=number of optimized parameters) simulations or experiments to do all combinations or a full factorial. The

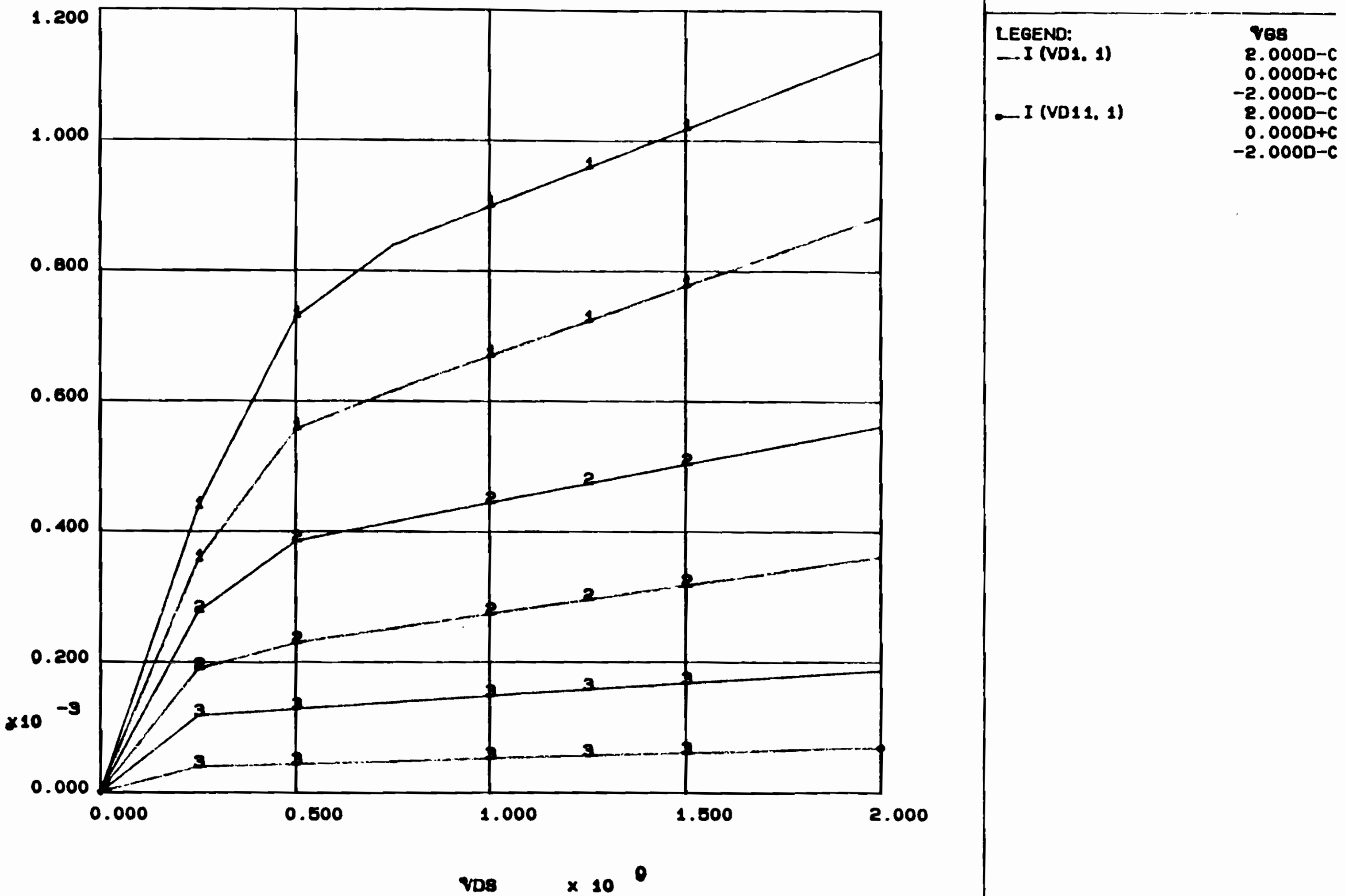


Figure (6.5). The I-V curves of actual devices at +/- 3σ from wafer 312-1-6.

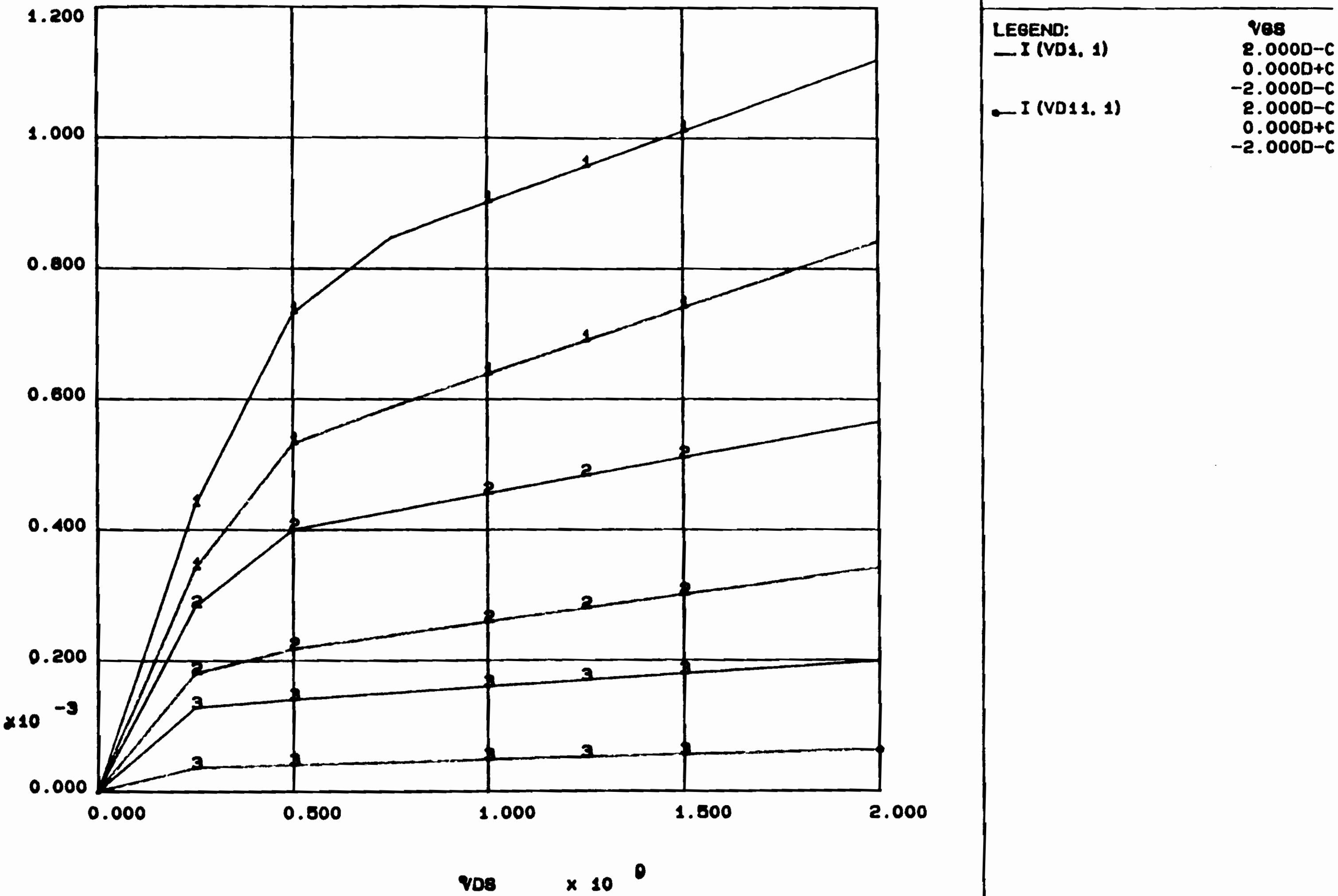


Figure (6.6). The I-V curves of actual devices at +/- 3σ from wafer 312-4-6.

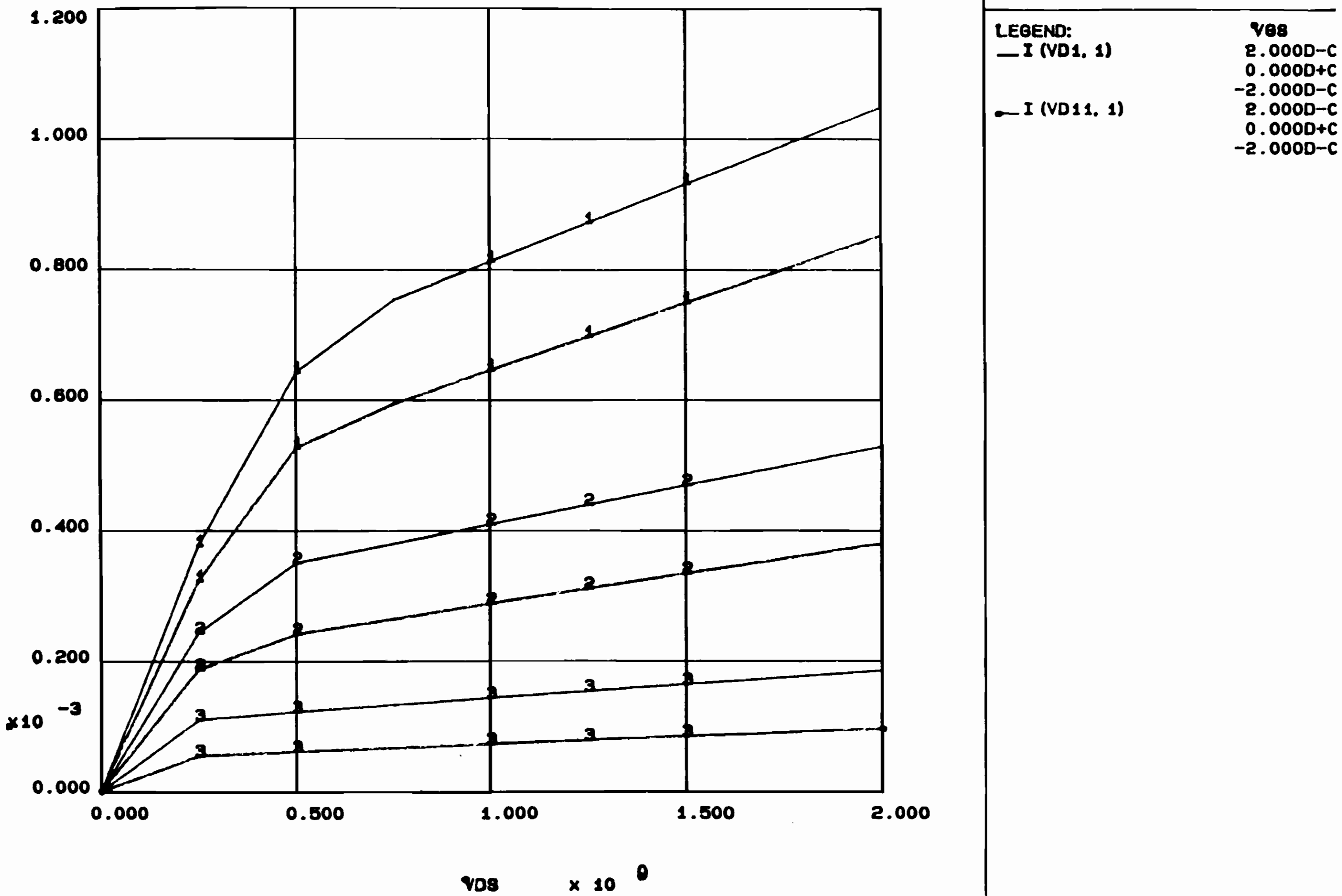


Figure (6.7). The I-V curves of actual devices at +/- 3σ from wafer 444-1-5.

Table (6.9). Actual nominal and worst case model parameters.

Date: April 3,1992

Averaged from 312-1-6, 312-2-6, 312-4-6 using Taguchi optimizer

FET TYPE	VT	BETA	LAMBDA	RS	RD	IS	PB
DFET							
mean	-0.417	1.43E-3	0.40	1E-6	1E-6	1E-15	0.72
ST.DEV.	29mV	0.07E-3	0.05	-	-	-	-
EFET							
mean	+0.115	2.30E-3	0.19	1E-6	1E-6	1E-15	0.72
ST.DEV.	20mV	0.04E-3	0.02	-	-	-	-

Note:

1) Eventhough the Rs and Rd values are approximately 100 Ω in a physical sense they were set to 1 $\mu\Omega$ so the optimizer could compare the drain and gate voltages measured in test directly with a simulated model. If the measured Rs and Rd values are used then the measured drain and gate voltages must be adjusted to represent the internal Vds and Vgs nodes, which accounts for the Rs and Rs voltage drops.

2) In order to model a dual gate EFET used in AOI gates etc.; use two EFETs connected together in a subcircuit.

Instead of using correlation coefficients because the coefficients are so low, actual FET parameters of the devices +/- 3σ on the wafer are chosen. The wafers evaluated were: 312-1-6, 312-4-6, and 444-1-5. The I-V curves were then compared for each wafer and the wafer with the largest spread was chosen (312-4-6).

+3σ: VT=-0.354 BETA=1.48E-3 LAMBDA=0.473

AVE: VT=-0.448 BETA=1.50E-3 LAMBDA=0.344

-3σ: VT=-0.473 BETA=1.46E-3 LAMBDA=0.359

Curtice square law model [11] requires 1 additional parameter to be optimized (α), that translates into 27 experiments to do all combinations of parameters instead of 9 for the JFET model. This is a straight forward modification of the existing program, with the only drawback being a tripling in execution time. However, the Statz [12] and TOM [9] models require 4 and 5 parameters respectively to be optimized resulting in 81 and 243 experiments. The Statz model would require 9 times the CPU time of the JFET model to do a full combination optimization. Moreover, the TOM model would require 27 times the CPU time of the JFET model to simulate all combinations of parameters. This geometric increase in CPU time to do the optimization becomes unacceptable.

However, orthogonal arrays can be implemented to significantly reduce the number of experiments required, yet still get the same amount of information of a full factorial set of experiments. This method of using orthogonal arrays will be demonstrated in the following chapter in the design and simulation of a differential amplifier. Therefore, by incorporating the use of orthogonal arrays into optimizing the Statz model parameters will reduce the number of simulations from 81 to 18 (78% reduction) using a L18 array described in references [13,14]. An L18 array describes the 18 combination of up to seven 3-level parameters and one 2-level parameter that must be run to obtain the same information from a full factorial of all the parameters. In other words, 4374 possible

combinations of parameters can be reduced to only 18. The use of the array will be demonstrated in the next chapter. Therefore, the reduction in simulations using the L18 for the TOM model will be from 243 simulations to 18 simulations for a 93% reduction in the number of simulations.

The L18 array can optimize up to 7 model parameters. However, some models require up to 17 modeling parameters. For example, microwave FET models typically contain on the order of 12 parameters requiring optimization. Another example is the popular BSIM (Berkeley Short-channel IGFET Model) for simulating silicon MOSFET transistors. The BSIM model contains 62 parameters that can be controlled. Fortunately, less than half of these parameters typically have a significant impact on the lot to lot variation in a given process. It should be noted the orthogonal arrays are not limited to the seven parameters in a L18 array. For instance, an L54 array can optimize 25 parameters with 54 simulations or an L81 can be used to optimize 40 model parameters with 81 simulations. It is up to the designer to determine the number of parameters that should be optimized for a given model and choose the corresponding orthogonal array.

Lastly, the user needs to keep in mind some of the limitations involved in using orthogonal arrays. A presumption in using the orthogonal arrays are the parameters are additive or independent of each other. If the parameters are not

independent, they will interact with each other, causing non-ideal settings for the parameters. In other words, there will be a larger RMS error between measured and simulated results. However, there are methods that can be used to analyze the orthogonal array results for interactions. The first method is to do an ANOVA (ANalysis Of VAriance) table of the results from the orthogonal array and look at the contribution of each parameter to the over all variation in the results. This is useful for two reasons. First, the ANOVA table will help eliminate parameters that are insignificant in the optimization. Second, the ANOVA table will show the contribution of error in the results, which is due to interactions between parameters or limitations in the fitting capabilities of the model. Therefore, if the error is small the interactions have no effect on the optimized results and the model fit is good, but if it is significant then interaction graphs need to be evaluated to observe which parameters are interacting. Hence, any type of model in GaAs or silicon be can optimized using the Taguchi method. In order to keep CPU times small for models with a large number of parameters requires using orthogonal arrays. An analysis of variance is done on the results of the orthogonal array to evaluate the effect each parameter has on the over all optimization as well as the amount of error in the optimization. This error can be caused by the inaccuracy of the model and/or interactions between the model parameters.

In summary, the Taguchi optimization method has been characterized and modified to be implemented in a manufacturing environment. A program was developed (FIT1.FOR) to take data directly from DC test and run the Taguchi optimization using the JFET model for various types of FETs. The importance of optimizing each FET on the wafer versus optimizing a single set of average I-V curves was demonstrated. Furthermore, a novel methodology was developed to obtain the worst case FET parameters. This approach used data from actual FETs to determine the +/- 3σ limits of the model parameters instead of using the more common approach of correlating the behavior of one model parameter with the remaining model parameters.

The next chapter will utilize these optimized JFET model parameters in the design of a differential amplifier. The design approach will also use the Taguchi methodology to find the optimum values of the various FETs.

CHAPTER VII

DIFFERENTIAL AMPLIFIER DESIGN

7.1 INTRODUCTION

The ultimate goal of this modeling effort is to design analog circuits with a higher degree of confidence in the simulated results. Therefore, the next step is to incorporate these improved SPICE parameters into the JFET model and simulate an actual circuit. Moreover, the same Taguchi methodology introduced previously can also be applied to the design of circuits as well [13,14]. A SNR equation will be developed as before, but the concept of utilizing orthogonal arrays to substantially reduce the number of simulations will be explained.

The circuit used for this evaluation will be the differential amplifier (diff-amp) shown in Figure (7.1). The differential amplifier was chosen because it is an essential cell in most digital and analog circuits. In digital circuits, differential amplifiers are used in level translators to convert signals from ECL levels to GaAs levels. In analog circuits, differential amplifiers are the heart of comparators and operational amplifiers. This particular diff-amp uses all

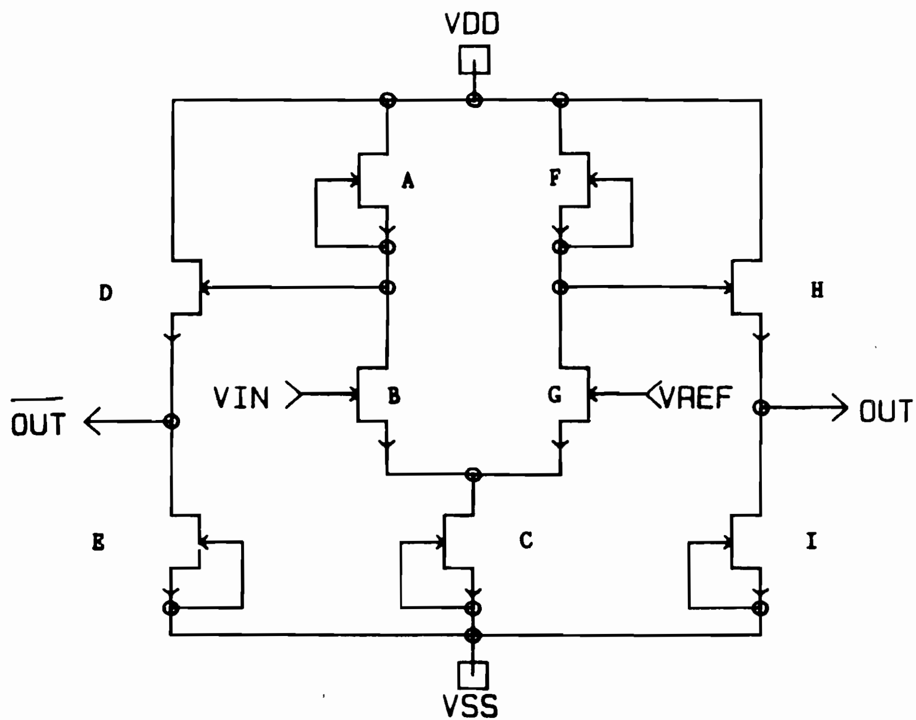


Figure (7.1). Differential amplifier using all DFETs.

depletion-mode FETs (D-FETS) and was chosen because of its simplicity. However, prior to elaborating on the design method it is imperative to understand and address the effects of offset voltage in the diff-amp.

7.2 EFFECT OF OFFSET VOLTAGE IN THE SIMULATION

Offset voltage is a term used to describe differences in the outputs of a differential amplifier when both inputs are at the same DC bias. This effect is caused by the variations in the threshold voltage of the input FETs of the diff-amp. Offset voltages are found in all silicon and GaAs FET transistors and are typically less than 50 millivolts [35]. Conversely, bipolar transistors of both technologies typically have offset voltages less than 1 mV because of the uniform control of the base-emitter voltage. This is one of the principal reasons for using bipolar transistors in high resolution flash A/D converters because of the inherently low offset voltage. By not including the effect of offset voltage in the simulation will result in very optimistic results in the small signal input resolution as well as identical output voltages for both outputs of the diff-amp. Therefore, this section will characterize the relationship between measured and simulated results to accurately account for offset voltage.

The initial step is to obtain measured data on an entire wafer describing the offset voltage. Table (7.1) shows the raw data of 33 differential amplifiers from wafer 305-3-5. The average and standard deviation at the bottom is for all diff-amps within +/- 1 σ of the original 49 devices. This represents the majority of the device population and eliminates outliers and misleading data. The right-most column shows the difference between the two outputs with both inputs equal, which is the offset voltage. Hence, as can be seen from these values, offset does indeed have a significant impact on the outputs of the diff-amp. The offset voltages depicted here can be directly attributable to the variation in the threshold voltage of the input FETs. The standard deviation of the threshold voltage of all FETs measured was 30 mV. In other words, 68% of the measured FETs had a threshold voltage variation within 60 mV of each other (+/-1 σ). Moreover, Table (7.1) shows 68% of the diff-amps had an offset voltage that varied within 66 mV of each other. Therefore, the variation of the threshold voltage is comparable to the variation of the offset in the diffamps.

The data in Table (7.1) shows an offset term must be included in the simulation to improve the accuracy between measured and simulated results because of its significance (an average of 45 mV variation between OUT and OUTBAR). However, the data from Table (7.1) shows the voltage offset has an inherent polarity. In other words, OUT is not always greater

Table (7.1). Raw DC data of 33 diff-amps on a single wafer.

Device #	OUT (V)	$\overline{OUT}$ (V)	OFFSET $\Delta V = \lvert OUT - \overline{OUT} \rvert$ (mV)
02-02	.439	.478	-39
04-02	.467	.383	84
07-02	.595	.514	81
08-02	.451	.517	-66
04-03	.467	.543	-24
05-03	.532	.560	-28
07-03	.567	.525	42
08-03	.482	.523	-41
03-04	.511	.486	25
04-04	.481	.476	5
05-04	.494	.371	123
08-04	.510	.462	48
02-05	.341	.321	20
03-05	.483	.368	115
04-05	.525	.554	-29
05-05	.537	.450	87
06-05	.510	.468	42
07-05	.424	.439	-15
08-05	.504	.483	21
02-06	.594	.674	-80
03-06	.567	.630	-53
04-06	.445	.444	1
07-06	.402	.361	41
08-06	.581	.600	-19
04-07	.600	.660	-60
05-07	.585	.688	-103
06-07	.617	.552	65
07-07	.344	.351	-7
02-08	.375	.339	36
03-08	.401	.398	3
05-08	.470	.430	40
07-08	.516	.554	-38
08-08	.585	.587	-2
Average	.497	.491	45 mV
Standard Deviation	.075	.099	33 mV

than or less than OUTBAR, but varies per device. Therefore, the data in Table (7.1) must be adjusted to show the average of positive as well as negative offset because a simulation can't account for both simultaneously. This is done in Table (7.2), which shows the data from Table (7.1) broken down into positive offset and negative offset. The distribution is surprisingly evenly divided, with 15 diff-amps demonstrating negative offset and 18 diff-amps demonstrating positive offset. The mean and standard deviations are relatively close, which eliminates extreme single sided behavior from offset polarity. However, Table (7.2) does show a 9 mV difference in the mean and an 11 mV difference in the standard deviation between the negative and positive offset. The problem now becomes how to simulate the effect of offset with respect to the polarity involved.

In order to compare the effect of incorporating offset into the simulation to improve accuracy, both the positive and negative offset need to be evaluated separately. Hence, the average measured value of the output for the respective positive and negative offset in Table (7.2) was used as the target value (0.557 V for negative offset and 0.484 V for positive offset). Next, the value for offset to use in the simulations as well as the interactive effect of offset between simulation methods had to be determined. The mean value for offset was used from Table (7.2) (40 mV for negative offset and 49 mV for positive offset). However, in directly

Table (7.2). Table (7.1) broken down into positive and negative offset voltages.

DIE NUMBER	NEGATIVE OFFSET (mV)	$\overline{OUT}$ (V)	POSITIVE OFFSET (mV)	$\overline{OUT}$ (V)
2-2	39	.478		
4-2			84	.467
7-2			81	.595
8-2	66	.517		
4-3	24	.543		
5-3	28	.560		
7-3			42	.567
8-3	41	.523		
3-4			25	.511
4-4			5	.481
5-4			123	.494
8-4			48	.510
2-5			20	.341
5-8			40	.470
7-8	38	.554		
8-8	2	.587		
3-5			115	.483
4-5	29	.554		
5-5			87	.537
6-5			42	.510
7-5	15	.439		
8-5			21	.504
2-6	80	.674		
3-6	53	.630		
4-6			1	.445
7-6			41	.402
8-6	19	.600		
4-7	60	.660		
5-7	103	.688		
6-7			65	.617
7-7	7	.351		.375
2-8			36	.401
3-8			3	
DEVICES CHOSEN = 33/49	μ = 40 mV σ = 28 mV	μ = .557 V	μ = 49 mV σ = 37 mV	μ = .484 V

inserting the 40 mV of offset into the simulation by giving FET "B" a different model statement resulted in output voltages with over 100 mV of offset. Moreover each simulation method varied significantly from each other. Therefore, the amount of threshold variation for each simulation method was found to give approximately 40 mV of offset for the negative case and 49 mV for the positive case. Table (7.3) shows the required offset in the simulation to replicate the measured positive and negative offset. In other words, in order to duplicate the measured 50 mV of positive offset required over 10 mV of cffset in the simulation. Therefore, in the following section a conservative offset value of +/- 20 mV is used in the simulations to account for the voltage offset effect in designing the differential amplifier.

Table (7.3). Offset required in the simulation to reproduce the actual measured offset.

Positive Offset

Simulated Output	VT Offset For $\Delta V_o = 50$ mV
.492	10.2 mV

Negative Offset

Simulated Output	VT Offset For $\Delta V_o = 39$ mV
.468	8.7 mV

7.3 TAGUCHI METHOD USED IN THE DESIGN OF A DIFF-AMP

In the previous section a basic characterization of offset voltage was investigated along with accurately replicating the phenomenon in the simulation. This understanding of the offset voltage will be incorporated in the design of the differential amplifier utilizing the Taguchi methodolcgy. The ideal function of a diff-amp is shown in Figure (7.2) exhibiting linear gain and no offset voltage. Moreover, the primary goal of this design is to obtain the optimum combination of FET widths for the diff-amp to give high gain, linear response and small offset voltage. Offset voltage is the difference in the two outputs of a diff-amp for identical inputs and was described in greater detail in the previous section. The y-axis of Figure (7.2) is the difference between the two outputs of the diff-amp and the x-axis is the input voltage, with the reference voltage set at 0 V. Therefore, a SNR equation can be developed based on this function using the slope and standard deviation of this line.

$$SNR=10*LOG(m**2/\sigma**2) \qquad (7.1)$$

Where "m" is the slope of the line and "σ" is the standard deviation. This equation applies equal weight to both design objectives, which are high gain and a linear response. As "m"

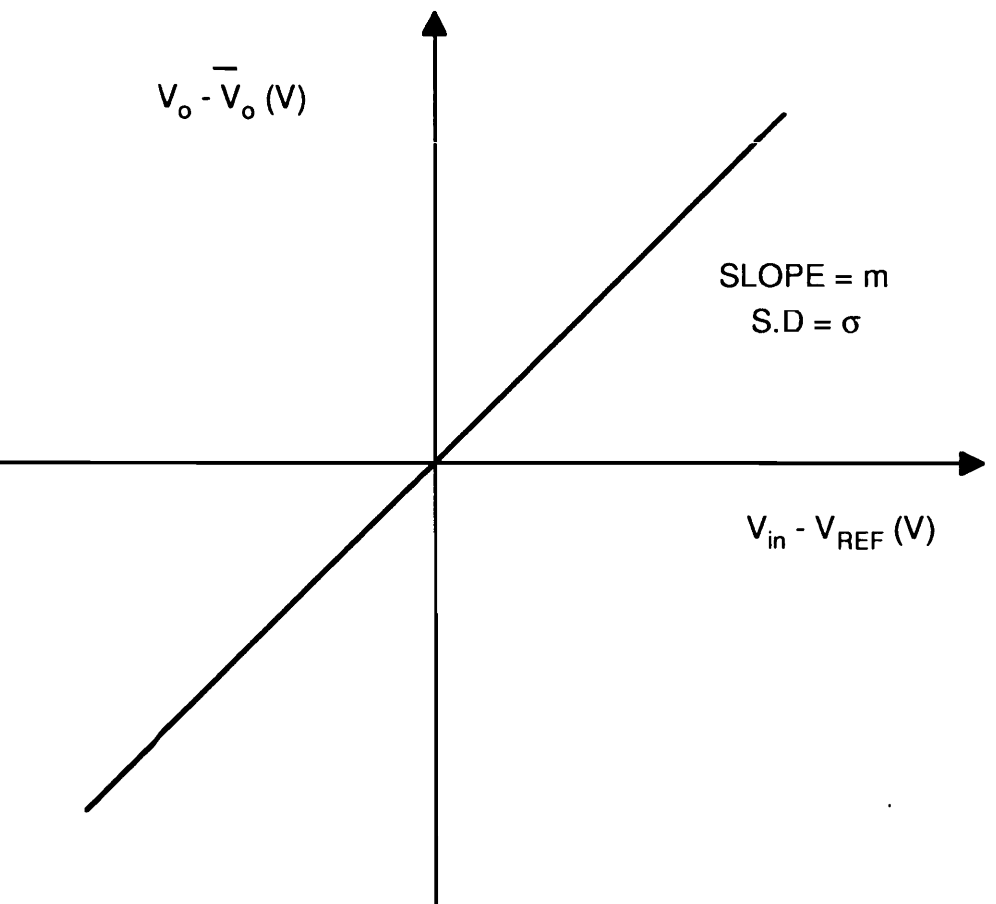


Figure (7.2). The ideal function of a differential amplifier showing no offset and constant linear gain.

gets larger the gain also gets larger and the smaller the "σ", the better the linearity. Hence, the larger the SNR the higher the gain and the more linear the response.

The next requirement is to define the control or signal factors and the noise factors. The signal factors are variables that can be controlled and adjusted, whereas noise factors are considered uncontrollable. The signal factors will consist of various FET widths and power supply levels with three levels each as shown in Table (7.4). The relationship between the FET sizes were based on the required symmetry for the design. In other words, the left hand side must mirror the right hand side of the diff-amp for symmetrical behavior. A basic understanding of the behavior of the diff-amp is used to arrive at the variations of the signal factors. Hence, a total of six signal factors were identified containing three levels each. A full factorial or simulating all possible combinations would require three to the sixth power or 729 simulations. However, the power of utilizing orthogonal arrays becomes apparent now because only 18 simulations will be required and the same amount of information will be attained. This is a reduction of 98% in the number of required simulations.

Before the orthogonal array can be set-up the noise factors must be defined. As was previously mentioned, the noise factors are not controllable and the primary uncontrollable factor relating to offset and linearity is the

Table (7.4). The noise and signal variables with their corresponding levels.

Signal Factor	Level 1	Level 2	Level 3
FET B = FET G (μm)	10	20	40
FET A = FET F (μm)	5	10	20
FET C/FET A (μm)	1.0	1.5	2.0
FET H/FET G (μm)	1.0	1.5	2.0
FET I/FET H (μm)	0.5	1.5	2.0
$V_{DD} = (-V_{SS})$ (V)	1	2	5

Noise Factor	Level 1	Level 2	Level 3
Offset Voltage of FET G	$V_tG=V_tB$	V_tG-V_tB+20 mV	$V_tG=V_tB-20$ mV

variation in threshold voltage. A maximum range of threshold voltage variation in the simulation will be set at +/- 20 mV to duplicate the measured offset as shown in the previous section. This is accomplished by applying to FET "G" a variable threshold voltage by using a different model statement in the SPICE simulation. Hence, there will be three noise levels: Vt of FET G (VtG) = Vt of FET B VtB, VtG = VtB + 20 mV, and VtG = VtB - 20 mV.

Since there are six signal factors with 3 levels each an L18 orthogonal array is the best choice. The L18 array is shown in Table (7.5), which was taken from Appendix C in reference [14]. The array consists of experiments or simulations in this case and can accommodate seven 3 level factors and one 2 level factor, with each level repeated the same number of times in any given column. Therefore, Table (7.2) tells us the exact 18 combinations of signal levels to use that will give us the same amount of information as if all 729 possible simulations (full factorial) had been performed. Table (7.6) combines Table (7.4) and Table (7.5) showing the actual FET sizes and power supply setting used in each of the 18 simulations. Furthermore, the three right-most columns show the slope of the line, the standard deviation and SNR results of each simulation. An actual plot of the simulated results for experiment #1 is shown in Figure (7.3). The top two sets of curves are the two outputs of the diff-amp V(6),V(7), with each set containing a different noise level (numbered as 1-2-

Table (7.5). A L18 orthogonal array containing 7 three level factors and 1 two level factor.

$L_{18}\ (2^1 \times 3^7)$

$L_{18}\ (2^1 \times 3^7)$ Orthogonal Array

Expt. No.	Column 1	2	3	4	5	6	7	8
1	1	1	1	1	1	1	1	1
2	1	1	2	2	2	2	2	2
3	1	1	3	3	3	3	3	3
4	1	2	1	1	2	2	3	3
5	1	2	2	2	3	3	1	1
6	1	2	3	3	1	1	2	2
7	1	3	1	2	1	3	2	3
8	1	3	2	3	2	1	3	1
9	1	3	3	1	3	2	1	2
10	2	1	1	3	3	2	2	1
11	2	1	2	1	1	3	3	2
12	2	1	3	2	2	1	1	3
13	2	2	1	2	3	1	3	2
14	2	2	2	3	1	2	1	3
15	2	2	3	1	2	3	2	1
16	2	3	1	3	2	3	1	2
17	2	3	2	1	3	1	2	3
18	2	3	3	2	1	2	3	1

Table (7.6). The SNR results of the L18 array for the differential amplifier.

Experiment #	Col 2 FET B=G	Col 3 FET A=F	Col 4 FET C	Col 5 FET H=D	Col 6 FET I=E	Col 7 V_{DD}/V_{SS}	m	σ	η (dB)
1	10 μm	5 μm	5 μm	10 μm	5 μm	±1	1.27	.048	28.4
2	10 μm	10 μm	15 μm	15 μm	15 μm	±2	7.11	.145	33.8
3	10 μm	20 μm	40 μm	20 μm	30 μm	±5	3.46	.117	29.4
4	20 μm	5 μm	5 μm	30 μm	30 μm	±5	19.6	.391	34.0
5	20 μm	10 μm	15 μm	40 μm	60 μm	±1	2.85	.0917	29.8
6	20 μm	20 μm	40 μm	20 μm	10 μm	±2	1.24	.081	23.7
7	40 μm	5 μm	7.5 μm	80 μm	120 μm	±2	8.53	.379	27.0
8	40 μm	10 μm	20 μm	60 μm	30 μm	±5	.458	.028	24.3
9	40 μm	20 μm	20 μm	80 μm	80 μm	±1	1.40	.047	29.5
10	10 μm	5 μm	10 μm	20 μm	20 μm	±2	.554	.036	23.8
11	10 μm	10 μm	10 μm	10 μm	15 μm	±5	11.9	.209	35.1
12	10 μm	20 μm	30 μm	15 μm	7.5 μm	±1	1.80	.045	31.9
13	20 μm	5 μm	7.5 μm	40 μm	20 μm	±5	17.2	.373	33.3
14	20 μm	10 μm	20 μm	20 μm	20 μm	±1	.401	.024	24.6
15	20 μm	20 μm	20 μm	30 μm	45 μm	±2	2.66	.091	29.3
16	40 μm	5 μm	10 μm	60 μm	90 μm	±1	.089	.0046	25.8
17	40 μm	10 μm	10 μm	80 μm	40 μm	±2	6.86	.142	33.7
18	40 μm	20 μm	30 μm	40 μm	40 μm	±5	1.41	.236	35.5

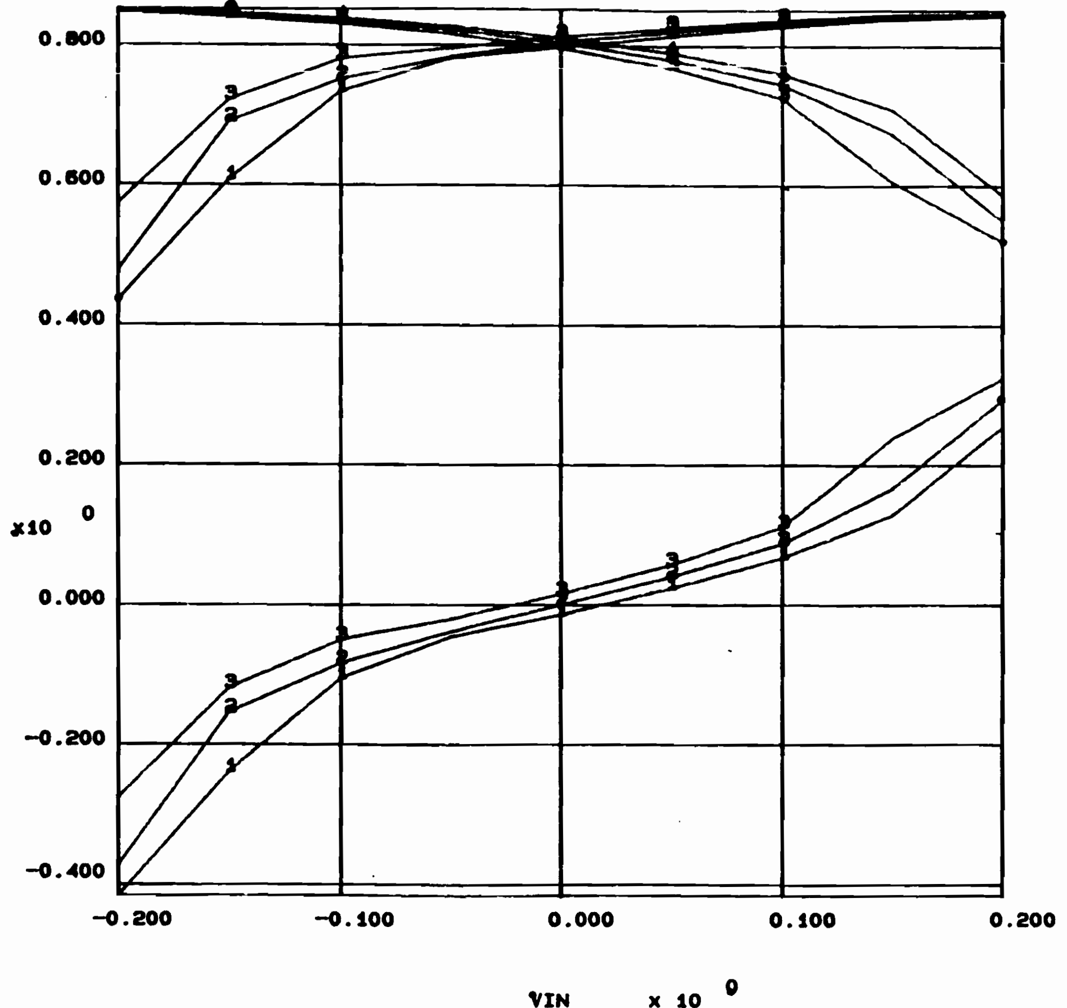


Figure (7.3). Simulation results for experiment #1; V(6) and V(7) are the outputs of the diff-amp.

3). The bottom set of curves shows the difference between V(6)-V(7). It is this bottom set of curves that are used to determine the SNR as shown in Figure (7.4). A least squares fit to the data is calculated with the line forced through the origin.

This procedure is used on all 18 experiments to solve for all the SNRs in Table (7.6). The best SNR (larger the better) is simulation #18, with a SNR=35.5 dB. However, keep in mind this experiment is not the optimum design because only 18 of the 729 possible combinations have been evaluated. In order to derive the optimum combination (largest gain with highest linearity) for the FET widths of the diff-amp, the response curves for each variable must be evaluated as in the previous chapters. However, for simplification the response curves are shown in a tabulated format in Table (7.7). From this table the highest SNR for each variable is chosen and shown in the rightmost column of Table (7.7). Therefore, using these values in the design is expected to give the highest gain and linearity of all 729 possible circuit configurations.

The goal of this section was to show the basic procedure of using the Taguchi method in design. There are additional steps that go beyond the scope of this example. For instance, the results can be quantified by solving an ANOVA (ANalysis OF VAriance) table and finding the process estimate equation within a given confidence interval.

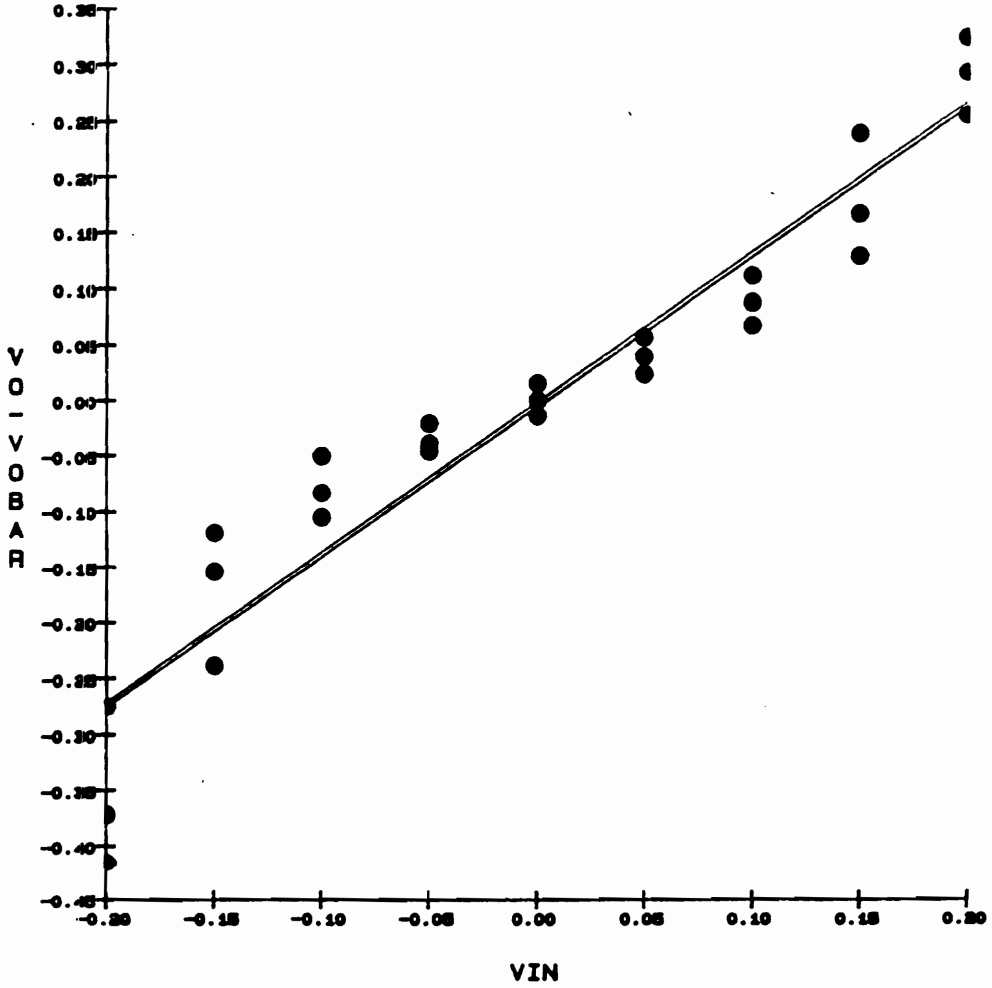


Figure (7.4). The least squares of the data in experiment #1 to determine the SNR.

Table (7.7). Response table for all the variables in Table (7.6) and the resulting optimum design.

Variable	Average SNR (dB)			Optimum Design
	Level 1	Level 2	Level 3	
FET B=G	30.4	29.12	29.3	10 μm
FET A=F	28.7	30.2	29.9	10 μm
FET C	31.7	31.9	25.2	15 μm
FET H=D	29.1	29.85	29.92	20 μm
FET I=E	29.2	30.2	29.4	20 μm
V_{DD}/V_{SS}	28.3	28.6	31.9	±1-5 V

CHAPTER VIII

CONCLUSIONS

The Taguchi methodology has been successfully implemented for the first time in the optimization of I-V curves. Moreover, this Taguchi methodology has been implemented in the design of a differential amplfier as well. This research has demonstrated just a small set of examples in which these powerful methods can be applied. These techniques develop a SNR equation which is used to improve additivity of the signal factors for optimization of modeling parameters as well as the optimization of an IC design. The Taguchi method utilizes orthogonal arrays to significantly reduce the number of simulations or experiments required.

Improvements in the accuracy of the simulated I-V curve fit of over 80% has been achieved between DC test extracted parameters and the Taguchi optimized parameters. Moreover, the CPU execution time for the optimization is 96% less than a commercial optimizer utilizing the Levenberg-Marquardt algorithm (optimizing 31 FETs). These techniques are very generic and can easily be implemented with any simulator that utilizes simulation modeling parameters extracted from measured DC test data. There is very limited commercial

software available for optimizing IC model parameters and it is quite expensive. Furthermore, some of the commercial software is designed to optimize only a single set of I-V curves at a time, which means from a practical standpoint of only being able to optimize average I-V data from an entire wafer. On the contrary, it was shown in Chapter VI that there is a significant difference between optimizing each device on a wafer and optimizing the average of all devices. The Taguchi method optimizes all devices in a very time efficient manner. In addition, two methods have been evaluated to obtain the worst case modeling parameters. One method looked at the correlation coefficients between modeling parameters and the second looked at the actual device parameters that define the +/- 3σ limits of the process. The second method has been implemented because of the low correlation coefficients between the parameters.

The universal applicability of the Taguchi optimization method to other GaAs as well as silicon models was evaluated. It was determined that other esoteric models having significantly more parameters requires using orthogonal arrays to limit the number of required simulations. Lastly, an example is given that describes using the Taguchi methodology in the design of a differential amplifier. The goal is to make the design as least sensitive as possible to offset while maintaining linear gain. A SNR equation was derived that accounted for offset and linearity in the simulation.

Moreover, an L18 orthogonal array was used to reduce the number of simulations required to do a full factorial from 729 to only 18 (98% reduction). Future work could address the large task of evaluating the Taguchi curve optimizer for more complicated GaAs and silicon models. This effort would require integrating the analysis of variance capability into the optimization program.

APPENDIX I

This FORTRAN program (FIT1.FOR) reads in the raw test data for up to 33 devices, calculates the IDS current from the extrinsic Vds and Vgs values. In order to use the extrinsic values Rs and Rd must be set to zero so Beta will be able to be scaled directly (calculated slope between measured and simulated devices). The program then calculates the Taguchi SNR ratio for nine different combinations of Vt and Lambda. The highest value is chosen, which reflects the best fit and a re-adjusted beta is found from the slope of the measured versus simulated Ids values. Lastly, the program calculates an optimum value for Vt and lambda from a quadratic curve fit from the response graphs. The program requires one input file with the measured data and outputs two files: FIT1.OUT contains the optimization progression and values for each individual device, while STAT.OUT contains the overall mean and standard deviation for all three optimized parameters from the entire wafer.

```
C        THIS IS PROGRAM CALCULATES THE TAGUCHI SIGNAL-TO-NOISE RATIO FOR
C        A SET I-V CURVE FITTING PARAMETERS FOR A JFET MODEL. THE PROGRAM
C        EVALUATES 27 I-V POINTS (3 VGS CURVES WITH 9 VDS VALUES PER CURVE)
C        AND CAN HANDLE UP TO 32 DEVICES PER WAFER. THE PROGRAM SHOWS THE
C        STATISTICS FOR EACH DEVICE IN FIT1.OUT AND SHOW THE OVERALL AVERAGES
C        AND STANDARD DEVIATIONS IN STAT.OUT. THE PROGRAM PROMPTS THE USER
C        FOR NUMBER OF DEVICES AND WHETHER THE FET IS ENHANCEMENT OR DEPLETION
C        MODE. THE PROGRAM ALSO CALCULATES THE RMS ERROR FROM THE ACTUAL
C        TAGUCHI SNR EXPERIMENTS VERSUS THE MEASURED DATA AS WELL AS QUADRACTIC
C        CURVE FIT TO THE TAGUCHI RESPONSE CURVES.
C
C        ARRAY A1 AND A ARE THE MEASURED DATA
C        ARRAY B AND BNEW ARE THE CALCULATED DATA IN THIS PROGRAM
C        ARRAY C IS THE SIMULATED DATA FROM PRECISE


         DIMENSION A(900)
         DIMENSION A1(900,3)
         DIMENSION A2(900,3)
         DIMENSION R(33,2)
         DIMENSION B(243)
         DIMENSION BNEW(243)
         DIMENSION B1(27)
         DIMENSION B2(27)
         DIMENSION B3(27)
         DIMENSION B4(27)
         DIMENSION B5(27)
         DIMENSION B6(27)
         DIMENSION B7(27)
         DIMENSION B8(27)
         DIMENSION B9(27)
         DIMENSION SLOPES(9)
         DIMENSION SIGSQS(9)
         DIMENSION SNS(9)
         DIMENSION RMSS(9)
         DIMENSION PAR(9,2)
         DIMENSION DEVICE(33)
         DIMENSION STAT1(33,5)
         DIMENSION STAT2(33,5)
         DIMENSION AVE1(5)
         DIMENSION AVE2(5)
         DIMENSION DEV1(5)
         DIMENSION DEV2(5)

         DOUBLE PRECISION TOTAL,TOTAL1,TOTAL2,SLOPE,SIGSQ,SN,SIGSQ1,SIG,SLOPESQ
         REAL LAMBDA,LAMMIN,LAMBINIT,LAMBSTEP,LAMBAVE,M1VT,M2VT,M1LAMB,M2LAMB
         REAL OLAMMIN,RMS1,SNRRMS
         CHARACTER*27 DEVICE
         CHARACTER*20 HEAD
         CHARACTER*1 TYPE
         INTEGER COUNT1,COUNT2,NUM

         CALL LIB$INIT_TIMER
         IFIT=0

         WRITE(6,*)'ENTER NUMBER OF DEVICES'
         READ(5,FMT='(I2)')NUM
C        INCLUDE WAFER AVERAGED I-V DATA BY ADDING ADDITIONAL 27 POINTS
         NUM=NUM+1
```

```
      WRITE(6,*)'Enter a Capital D if DFET, E for EFET or S for SFET'
      READ(5,FMT='(A1)')TYPE

      INCREM=1

C     THE OPEN STATEMENT READS DATA IN FROM ACTUAL TEST DATA
      OPEN (UNIT=12,FILE='444-1-5EPCM.DAT',CARRIAGECONTROL='LIST',
     a       STATUS='OLD')
C     VDS=A1(I,1)
C     VGS=A1(I,2)
C     IDS=A1(I,3)
      DO 13 J=1,NUM
      READ(12,FMT='(A27)')DEVICE(J)
      READ(12,FMT='(F6.2,F22.2)')R(J,1),R(J,2)
      READ(12,FMT='(A20)')HEAD
C     WRITE(6,FMT='(A27))')DEVICE(J)
C     WRITE(6,*)'RD=',R(J,1)
C     WRITE(6,*)'RS=',R(J,2)
C     WRITE(6,FMT='(A20))')HEAD
      DO 14 I=INCREM,INCREM+26
      READ (12,FMT='(F5.2,F7.2,E14.3)')A1(I,1),A1(I,2),A1(I,3)
C     CONVERT THE MEASURED EXTRINSIC VDS AND VGS TO THEIR INTRINSIC VALUES
C     SO THEY CAN BE DIRECTLY COMPARED WITH THE PROGRAM GENERATED SIMULATED
C     VALUES. DON'T HAVE TO CONVERT IDS BECAUSE IT WILL REMAIN THE SAME.
C     KNOWING RS,RD AND IDS:
C     INTRINSIC VDS = A2(I,1)
C     INTRINSIC VGS = A2(I,2)
C     RD = R(J,1)
C     RS = R(J,2)
      A2(I,1)=A1(I,1)-A1(I,3)*(R(J,1)+R(J,2))
      A2(I,2)=A1(I,2)-A1(I,3)*R(J,2)
C     CONVERT THE IDS CURRENT FROM AMPS TO MICROAMPS
      A2(I,3)=A1(I,3)*1E6
14    CONTINUE
      INCREM=INCREM+27
13    CONTINUE



C     INITIAL VALUE AND STEP SIZE FOR LAMBDA AND VT

      IF(TYPE.EQ.'D')THEN
      VTINIT=-0.50
      VTSTEP=0.1
      LAMBINIT=0.01
      LAMBSTEP=0.5
      ENDIF

      IF(TYPE.EQ.'E')THEN
      VTINIT=0.05
      VTSTEP=0.1
      LAMBINIT=0.1
      LAMBSTEP=0.1
      ENDIF

      IF(TYPE.EQ.'S')THEN
      VTINIT=-4.5
```

```
      VTSTEP=.5
      LAMBINIT=0.01
      LAMBSTEP=0.2
      ENDIF

      VAVE=VTINIT+VTSTEP
      LAMBAVE=LAMBINIT+LAMBSTEP
      VTIN1=VTINIT
      VTIN2=VTINIT
      VTIN3=VTINIT

C     GENERATE PARAMETER ARRAY FOR EACH EXPERIMENT:
      DO 5 I=1,9
      IF(I.LE.3)THEN
      PAR(I,1)=VTIN1
      VTIN1=VTIN1+VTSTEP
      PAR(I,2)=LAMBINIT
      ENDIF
      IF(I.GE.4.AND.I.LE.6)THEN
      PAR(I,1)=VTIN2
      VTIN2=VTIN2+VTSTEP
      PAR(I,2)=LAMBINIT+LAMBSTEP
      ENDIF
      IF(I.GE.7.AND.I.LE.9)THEN
      PAR(I,1)=VTIN3
      VTIN3=VTIN3+VTSTEP
      PAR(I,2)=LAMBINIT+2*LAMBSTEP
      ENDIF
5     CONTINUE

C     CALCULATE SIMULATED IDS VALUES WITHOUT USING SIMULATOR
C     27 POINTS FOR 9 SIMULATIONS = 243 TOTAL SIMULATED IDS VALUES
C     NEED TO USE SAME INTRINSIC VALUES FOR VDS AND VGS AS MEASURED


C     ********START MAIN PROGRAM LOOP:********

      M=1
      K=0
C     THIS LOOP REPRESENTS EACH DEVICE
      DO 150 J=1,NUM

      BETA=0.001



C     THIS LOOP REPRESENTS EACH EXPERIMENT OR SET OF VT AND LAMBDA
      DO 15 N=1,9

C     THIS LOOP REPRESENTS THE ACTUAL 27 INT. VDS AND VGS FOR @ DEVICE
C     THE B ARRAY WILL CONTAIN 27*9=243 PTS PER DEVICE
C     THE A(I) ARRAY LOADS THE 27 MEAS. IDS TO EVALUATE IN THE 9 EXP.

      DO 20 I=M,M+26,1
      A(I)=A2(I,3)
      VDS=A2(I,1)
C     WRITE(6,*)'I=',I,'  VDS=',VDS
      VGS=A2(I,2)
      V1=VGS-PAR(N,1)
```

```
C       ******** PAR(N,1)=VT  PAR(N,2)=LAMBDA ******
        IF(VDS.GE.0.0.AND.VDS.LE.V1)THEN
        B(I+K)=BETA*VDS*(1+PAR(N,2)*VDS)*(2*(VGS-PAR(N,1))-VDS)
        ELSE
        B(I+K)=BETA*(1+PAR(N,2)*VDS)*(VGS-PAR(N,1))**2
        ENDIF
20      CONTINUE
        K=K+27
C       WRITE(6,*)'N=',N
C       WRITE(6,*)'I=',I
15      CONTINUE
        WRITE(6,*)'M=',M
        WRITE(6,*)'K=',K
        WRITE(6,*)'DEVICE#=',J

C       CONVERT THE SIMULATED IDS CURRENT FROM AMPS TO MICROAMPS
        DO 11 I=1,243
        BNEW(I)=B(I)*1E6
11      CONTINUE

C       THE OPEN STATEMENT WRITES DATA TO AN OUTPUT FILE COMPARING
C       MEASURED AND SIMULATED CURRENTS FOR EACH DEVICE
C       THERE SHOULD BE 243 VALUES FOR BNEW AND 27 FOR A(I)
        OPEN (UNIT=61,FILE='COMPARE.OUT',CARRIAGECONTROL='LIST',
     a        STATUS='NEW')
        WRITE(61,*)'MEASURED I   SIMULATED I'
        DO 12 I=1,243
        WRITE(61,16)A(I),BNEW(I)
16      FORMAT(E8.2,6X,E8.2)
12      CONTINUE


C       THIS SUB BREAKS THE ARRAY B INTO 9 INDIVIDUAL ARRAYS FOR CALCULATIONS
C       B1 ARRAY IS EXPERIMENT #1
        DO 30 I=1,27
        B1(I)=BNEW(I)
30      CONTINUE
        CALL CALC(A,B1,RMS,SLOPE,SIG,SN)
        RMSS(1)=RMS
        SLOPES(1)=SLOPE
        SIGSQS(1)=SIG
        SNS(1)=SN

C       B2 ARRAY IS EXPERIMENT #2
        DO 35 I=1,27
        B2(I)=BNEW(I+27)
35      CONTINUE
        CALL CALC(A,B2,RMS,SLOPE,SIG,SN)
        RMSS(2)=RMS
        SLOPES(2)=SLOPE
        SIGSQS(2)=SIG
        SNS(2)=SN

C       B3 ARRAY IS EXPERIMENT #3
        DO 40 I=1,27
        B3(I)=BNEW(I+54)
40      CONTINUE
        CALL CALC(A,B3,RMS,SLOPE,SIG,SN)
        RMSS(3)=RMS
```

```
      SLOPES(3)=SLOPE
      SIGSQS(3)=SIG
      SNS(3)=SN

C     B4 ARRAY IS EXPERIMENT #4
      DO 45 I=1,27
      B4(I)=BNEW(I+81)
45    CONTINUE
      CALL CALC(A,B4,RMS,SLOPE,SIG,SN)
      RMSS(4)=RMS
      SLOPES(4)=SLOPE
      SIGSQS(4)=SIG
      SNS(4)=SN

C     B5 ARRAY IS EXPERIMENT #5
      DO 50 I=1,27
      B5(I)=BNEW(I+108)
50    CONTINUE
      CALL CALC(A,B5,RMS,SLOPE,SIG,SN)
      RMSS(5)=RMS
      SLOPES(5)=SLOPE
      SIGSQS(5)=SIG
      SNS(5)=SN

C     B6 ARRAY IS EXPERIMENT #6
      DO 55 I=1,27
      B6(I)=BNEW(I+135)
55    CONTINUE
      CALL CALC(A,B6,RMS,SLOPE,SIG,SN)
      RMSS(6)=RMS
      SLOPES(6)=SLOPE
      SIGSQS(6)=SIG
      SNS(6)=SN

C     B7 ARRAY IS EXPERIMENT #7
      DO 60 I=1,27
      B7(I)=BNEW(I+162)
60    CONTINUE
      CALL CALC(A,B7,RMS,SLOPE,SIG,SN)
      RMSS(7)=RMS
      SLOPES(7)=SLOPE
      SIGSQS(7)=SIG
      SNS(7)=SN

C     B8 ARRAY IS EXPERIMENT #8
      DO 65 I=1,27
      B8(I)=BNEW(I+189)
65    CONTINUE
      CALL CALC(A,B8,RMS,SLOPE,SIG,SN)
      RMSS(8)=RMS
      SLOPES(8)=SLOPE
      SIGSQS(8)=SIG
      SNS(8)=SN

C     B9 ARRAY IS EXPERIMENT #9
      DO 70 I=1,27
      B9(I)=BNEW(I+216)
70    CONTINUE
      CALL CALC(A,B9,RMS,SLOPE,SIG,SN)
      RMSS(9)=RMS
```

```
      SLOPES(9)=SLOPE
      SIGSQS(9)=SIG
      SNS(9)=SN

C     THE OPEN STATEMENT WRITES DATA TO A FILE
      OPEN (UNIT=60,FILE='TEMP.OUT',CARRIAGECONTROL='LIST',
     a       STATUS='NEW')
      WRITE(60,*)
      WRITE(60,*)
      WRITE(60,*)'---------------------------------------------------
     B------------------'
      WRITE(60,*)
      WRITE(60,FMT='(A27)')DEVICE(J)
      WRITE(60,999)
999   FORMAT(/,12HEXPERIMENT #,3X,6HLAMBDA,4X,2HVt,7X,3HRMS,
     B 7X,5HSLOPE,5X,5HSIGMA,5X,3HSNR)
      DO 1001 I=1,9
      WRITE(60,1000)I,PAR(I,2),PAR(I,1),RMSS(I),SLOPES(I),SIGSQS(I),SNS(I)
1000  FORMAT(I7,8X,F5.2,3X,F6.2,3X,F9.3,1X,F7.3,2X,F9.3,2X,F7.3)
1001  CONTINUE
C     FIND MAXIMUM VALUE OF SNR ARRAY
      SNRMAX=AMAX1(SNS(1),SNS(2),SNS(3),SNS(4),SNS(5),SNS(6),SNS(7),
     E SNS(8),SNS(9))
      WRITE(6,*)'SNRMAX=',SNRMAX
      DO 71 I=1,9
      COUNT=I
      TEMP=SNS(I)
      TEMP1=ABS(TEMP-SNRMAX)
      IF(TEMP1.LE.0.01)GO TO 72
71    CONTINUE
72    I=COUNT
      BETA=1.0E-3/SLOPES(I)

C     STORE THESE VALUES FOR EACH DEVICE TO CALCULATE OVERALL MEAN AND SIGMA
C     VT=PAR(I,1)    LAMBDA=PAR(I,2)
      NUM1=J
      STAT1(NUM1,1)=SNRMAX
      STAT1(NUM1,2)=PAR(I,1)
      STAT1(NUM1,3)=PAR(I,2)
      STAT1(NUM1,4)=BETA

      WRITE(60,73)SNRMAX,PAR(I,1),PAR(I,2),BETA
73    FORMAT(//,1X,8HMAX.SNR=,F7.3,2X,3HVT=,F5.2,2X,7HLAMBDA=,E9.3,
     E 2X,5HBETA=,E9.3)
C     WRITE(60,*)'MAXIMUM SNR=',SNRMAX,'VT=',PAR(I,1),'LAMBDA=',PAR(I,2)

C     CALCULATE THE RMS ERROR BETWEEN THE MEASURED DATA AND THE NEW
C     SPICE PARAMETERS FROM THE MAXIMUM SNR
      LAMBDA=PAR(I,2)
      VT=PAR(I,1)
      CALL RMSSUB(A,LAMBDA,VT,BETA,RMS1,TYPE)
      WRITE(60,*)'THE RMS ERROR USING THESE VALUES IS (uA):',RMS1
      STAT1(NUM1,5)=RMS1

C     SEE IF THE DATA MEETS THE CRITERIA FOR A QUAD. FIT TO THE RESP. CURVES

C     FIND AVERAGE SNR FOR EACH  VT AND COEFFICIENTS OF QUAD. EQ.
      Y1=(SNS(1)+SNS(4)+SNS(7))/3
      Y2=(SNS(2)+SNS(5)+SNS(8))/3
      Y3=(SNS(3)+SNS(6)+SNS(9))/3
```

```
      AVT=(Y3+Y1-2*Y2)/(2*VTSTEP**2)
      BVT=(Y3-Y1)/(2*VTSTEP)
      CVT=Y2

C     FIND IF THE RESPONSE CURVES ARE LOCAL MINIMA OR LOCAL MAXIMA
      M1VT=(Y2-Y1)/VTSTEP
      M2VT=(Y3-Y2)/VTSTEP
      IF(M1VT.GT.M2VT)THEN
      FLAG=1.0
      ELSE
      FLAG=0.0
      ENDIF

      WRITE(60,*)
      WRITE(60,*)
      WRITE(60,*)
      WRITE(60,*)'THE COEFFICIENTS OF A QUADRACTIC EQUATION Ax**2+Bx+C=y'
      WRITE(60,*)'TO THE RESPONSE CURVES WHERE x=step size and'
      WRITE(60,*)'dx=-B/2A (MAXIMUM POINT ON CURVE FROM AVE. x) ARE:'
      VTMIN=-BVT/(2*AVT)
      VTMINO=VTMIN
      WRITE(60,75)
75    FORMAT(/,5X,9HPARAMETER,6X,1HA,9X,1HB,9X,1HC,9X,2Hdx)
      IF(FLAG.EQ.1.0)THEN
      WRITE(60,76)AVT,BVT,CVT,VTMIN
76    FORMAT(8X,2HVT,6X,E9.2,2X,E9.2,2X,E9.2,2X,E9.2,4X,10H LOCAL MAX)
      VTMIN=VAVE+VTMIN
      ELSE
      WRITE(60,760)AVT,BVT,CVT,VTMIN
760   FORMAT(8X,2HVT,6X,E9.2,2X,E9.2,2X,E9.2,2X,E9.2,4X,10H LOCAL MIN)
      VTMIN=0.0
      ENDIF

C     FIND AVERAGE SNR FOR EACH LAMBDA AND COEFFICIENTS OF QUAD. EQ.
      Y1=(SNS(1)+SNS(2)+SNS(3))/3
      Y2=(SNS(4)+SNS(5)+SNS(6))/3
      Y3=(SNS(7)+SNS(8)+SNS(9))/3
      ALAMB=(Y3+Y1-2*Y2)/(2*LAMBSTEP**2)
      BLAMB=(Y3-Y1)/(2*LAMBSTEP)
      CLAMB=Y2
      AVELAMSNR=(Y1+Y2+Y3)/3
C     THE MINIMUM POINT ON THE QUAD. IS THE FIRST DERIVATIVE
      LAMMIN=-BLAMB/(2*ALAMB)
      OLAMMIN=LAMMIN
C     FIND IF THE RESPONSE CURVES ARE LOCAL MINIMA OR LOCAL MAXIMA
      M1LAMB=(Y2-Y1)/LAMBSTEP
      M2LAMB=(Y3-Y2)/LAMBSTEP
      IF(M1LAMB.GT.M2LAMB)THEN
      FLAG=1.0
      ELSE
      FLAG=0.0
      ENDIF

      IF(FLAG.EQ.1.0)THEN
      WRITE(60,77)ALAMB,BLAMB,CLAMB,LAMMIN
77    FORMAT(6X,6HLAMBDA,4X,E9.2,2X,E9.2,2X,E9.2,2X,E9.2,4X,10H LOCAL MAX)
      LAMMIN=LAMMIN+LAMBAVE
      ELSE
      WRITE(60,770)ALAMB,BLAMB,CLAMB,LAMMIN
770   FORMAT(6X,6HLAMBDA,4X,E9.2,2X,E9.2,2X,E9.2,2X,E9.2,4X,10H LOCAL MIN)
```

```
      LAMMIN=0.0
      ENDIF

C     BEFORE PRINTING OUT THE MAXIMUM SNR VALUES FROM THE CURVE FIT
C     BOTH VARIABLES MUST BE LOCAL MAX AND WITHIN THE SWEEP LIMITS

C     TEST THAT BOTH ARE LOCAL MAX
      IF(M1LAMB.GT.M2LAMB.AND.M1VT.GT.M2VT)THEN
      VTMIN1=VTINIT
      VTMAX1=VTINIT+(2*VTSTEP)
      ZLAMMIN=LAMBINIT
      ZLAMMAX=LAMBINIT+(2*LAMBSTEP)
      ELSE
      WRITE(60,*)
      WRITE(60,*)'CURVE FIT VT AND LAMBDA ARE NOT BOTH LOCAL MAX'
      IF(J.EQ.NUM)THEN
      IFIT=IFIT+1
      ENDIF
      GO TO 80
      ENDIF

      IF(VTMIN.GE.VTMIN1.AND.VTMIN.LE.VTMAX1)THEN
      GO TO 78
      ELSE
      WRITE(60,*)'CURVE FIT VT IS OUT OF SWEEP RANGE'
      IF(J.EQ.NUM)THEN
      IFIT=IFIT+1
      ENDIF
      GO TO 80
      ENDIF

78    IF(LAMMIN.GE.ZLAMMIN.AND.LAMMIN.LE.ZLAMMAX)THEN
      IFIT=IFIT+1
C     NOW SOLVE THE SNR FOR THE FITTED VALUES OF VT AND LAMBDA
      SNRVT=AVT*(VTMINO**2)+BVT*VTMINO+CVT
      SNRLAMB=ALAMB*(OLAMMIN**2)+BLAMB*OLAMMIN+CLAMB
      SNRFIT=SNRVT+SNRLAMB-(AVELAMSNR)
      WRITE(60,*)'THE ESTIMATED SNR USING CURVE FIT VT AND LAMB. IS:',SNRFIT


C     SOLVE NEW BETA FOR CURVE FIT:
      CALL BETACALC(A,LAMMIN,VTMIN,TYPE,BETANEW,SNRRMS)
      BETA=BETANEW
      STAT2(IFIT,1)=SNRRMS
      STAT2(IFIT,2)=VTMIN
      STAT2(IFIT,3)=LAMMIN
      STAT2(IFIT,4)=BETA
      WRITE(60,79)VTMIN,LAMMIN,BETA
79    FORMAT(//,1X,21HQUADRACTIC CURVE FIT:,1X,3HVT=,F6.3,1X,
     B 7HLAMBDA=,F5.3,1X,5HBETA=,E9.3)
      WRITE(60,*)
C     CALCULATE THE RMS ERROR OF THE CURVE FIT USING THE NEW BETA VALUE
      CALL RMSSUB(A,LAMMIN,VTMIN,BETA,RMS2,TYPE)
      WRITE(60,*)'THE RMS ERROR USING THESE VALUES IS (uA):',RMS2
      WRITE(60,*)'THE SNR USING THE NEW BETA IS:',SNRRMS
      STAT2(IFIT,5)=RMS2
      ELSE
      WRITE(60,*)'CURVE FIT LAMBDA IS OUT OF SWEEP RANGE'
      IF(J.EQ.NUM)THEN
      IFIT=IFIT+1
```

```
      ENDIF
      ENDIF
80    M=M+27
      K=0
150   CONTINUE


C     CALCULATE THE STATISTICS FOR ALL DEVICES AND PRINT TO A FILE
C     THE OPEN STATEMENT WRITES DATA TO AN OUTPUT FILE
      OPEN (UNIT=60,FILE='LIST.OUT',CARRIAGECONTROL='LIST',
     a       STATUS='NEW')
      DO 120 I=1,IFIT
      WRITE(60,121)STAT2(I,2),STAT2(I,3),STAT2(I,4)
121   FORMAT(E8.2,2X,E8.2,2X,E9.3)
120   CONTINUE

C     CALCULATE THE STATISTICS FOR ALL DEVICES AND PRINT TO A FILE
C     THE OPEN STATEMENT WRITES DATA TO AN OUTPUT FILE
      OPEN (UNIT=60,FILE='STAT.OUT',CARRIAGECONTROL='LIST',
     a       STATUS='NEW')
      CALL STAT(STAT1,NUM,AVE1,DEV1)
      WRITE(60,*)
      WRITE(60,*)
      WRITE(60,*)
      WRITE(60,*)'THE AVERAGE AND SIGMA FROM THE ACTUAL EXPERIMENTS:'
      WRITE(60,85)
85    FORMAT(/,10X,3HSNR,9X,2HVT,6X,6HLAMBDA,6X,4HBETA,6X,3HRMS,6X,
     B 8H#DEVICES)
      WRITE(60,95)AVE1(1),AVE1(2),AVE1(3),AVE1(4),AVE1(5),NUM
95    FORMAT(7HAVERAGE,2X,F6.2,4X,F6.2,4X,F6.2,6X,E8.2,2X,F8.2,6X,I2)
      WRITE(60,100)DEV1(1),DEV1(2),DEV1(3),DEV1(4),DEV1(5)
100   FORMAT(5HSIGMA,4X,F6.2,4X,E8.2,2X,E8.2,4X,E8.2,1X,F8.2)

      CALL STAT(STAT2,IFIT,AVE2,DEV2)
      WRITE(60,*)
      WRITE(60,*)
      WRITE(60,*)
      WRITE(60,*)'THE AVERAGE AND SIGMA FROM THE QUAD. CURVE FIT:'
      WRITE(60,105)
105   FORMAT(/,10X,3HSNR,9X,2HVT,6X,6HLAMBDA,6X,4HBETA,6X,3HRMS,6X,
     B 8H#DEVICES)
      WRITE(60,110)AVE2(1),AVE2(2),AVE2(3),AVE2(4),AVE2(5),IFIT
110   FORMAT(7HAVERAGE,2X,F6.2,4X,F6.3,4X,F6.3,6X,E8.2,2X,F6.2,6X,I2)
      WRITE(60,115)DEV2(1),DEV2(2),DEV2(3),DEV2(4),DEV2(5)
115   FORMAT(5HSIGMA,4X,F6.2,4X,E8.2,2X,E8.2,4X,E8.2,1X,F6.2)
      WRITE(60,*)
      WRITE(60,*)
      WRITE(60,*)


      CALL LIB$SHOW_TIMER
      STOP
      END


C     THIS SUBROUTINE CALCULATES THE SLOPE OF THE LINE THROUGH THE ORIGIN
```

```
C       ARRAY A IS THE MEASURED CURRENT AND ARRAY DAT IS THE CALCULATED
        SUBROUTINE CALC(A,DAT,RMS,SLOPE,SIG,SN)
        DIMENSION DAT(27)
        DIMENSION A(27)
        DOUBLE PRECISION RMS1(27),SUM
        REAL SN,SLOPESQ,SIGSQ
C       THIS PART CALCULATES THE RMS ERROR:
        SUM=0.0
        DO 200 I=1,27
        RMS1(I)=(DAT(I)-A(I))**2
        SUM=SUM+RMS1(I)
200     CONTINUE
        SUM=SUM/27.0
        RMS=SQRT(SUM)

C       THIS PART CALCULATES SLOPE, SIGMA AND SNR
        TOTAL=0.0
        TOTAL1=0.0
        TOTAL2=0.0
        DO 400 I=1,27
C       WRITE(6,*)DAT(I)
        TOTAL=TOTAL+A(I)*DAT(I)
        TOTAL1=TOTAL1+A(I)*A(I)
400     CONTINUE
        IF(TOTAL1.EQ.0)THEN
        TOTAL1=0.001
        WRITE(6,*)'TOTAL1=0 IN SUBROUTINE CALC'
        ENDIF
        SLOPE=TOTAL/TOTAL1
        SLOPESQ=SLOPE**2
        DO 500 I=1,27
        TOTAL2=TOTAL2+(DAT(I)-SLOPE*A(I))**2
C       WRITE(6,*)'TOTAL2=',TOTAL2
500     CONTINUE
        SIGSQ=TOTAL2/26
        SIG=SQRT(SIGSQ)
C       WRITE(6,*)'SLOPESQ=',SLOPESQ
C       WRITE(6,*)'SIGSQ=',SIGSQ
        IF(SIGSQ.EQ.0.0)THEN
        SIGSQ=0.001
        WRITE(6,*)'SIGSQ=0 IN SUBROUTINE CALC'
        ENDIF
        IF(SLOPESQ.EQ.0.0)THEN
        SLOPESQ=0.001
        WRITE(6,*)'SLOPESQ=0 IN SUBROUTINE CALC'
        ENDIF
        SN=10*ALOG10(SLOPESQ/SIGSQ)
C       WRITE(6,*)'SN=',SN
        RETURN
        END

C       THIS SUBROUTINE READS IN TWO DATA FILES WITH 12 ROWS AND TWO COLUMNS
C       AND FINDS THE RMS ERROR OF THE SECOND COLUMN (IDSS).

        SUBROUTINE RMSSUB(A,LAMBDA,VT,BETA,RMS1,TYPE)

        DIMENSION A(27)
        DIMENSION B1(27)
        DOUBLE PRECISION IDS(27)
        REAL SUM,LAMBDA,VT,BETA,RMS1
```

```
      CHARACTER TYPE
C     WRITE(6,*)
C     WRITE(6,*)'LAMB.=',LAMBDA,'VT=',VT,'BETA=',BETA
C     SOLVE FOR IDS VALUES WITH NEW MAX. SNR SPICE PARAM.
C     SPICE CURRENT EQUATIONS:
C     *****MUST CHANGE VGS=0.6 FOR AN EFET*****
      IF(TYPE.EQ.'D')THEN
      VGS=0.2
      ENDIF
      IF(TYPE.EQ.'S')THEN
      VGS=0.0
      ENDIF
      IF(TYPE.EQ.'E')THEN
      VGS=0.6
      ENDIF
      DO 10 J=1,3
      VDS=0.0
      IF(J.EQ.1)COUNT=1
      IF(J.EQ.2)THEN
      COUNT=10
      IF(TYPE.EQ.'S')THEN
      VGS=VGS-1.5
      ENDIF
      VGS=VGS-0.2
      ENDIF
      IF(J.EQ.3)THEN
      COUNT=19
      IF(TYPE.EQ.'S')THEN
      VGS=VGS-1.5
      ENDIF
      VGS=VGS-0.2
      ENDIF
      DO 5 I=COUNT,COUNT+8,1
      V1=VGS-VT
      IF(VDS.GE.0.0.AND.VDS.LE.V1)THEN
      IDS(I)=BETA*VDS*(1+LAMBDA*VDS)*(2*(VGS-VT)-VDS)
      ELSE
      IDS(I)=BETA*(1+LAMBDA*VDS)*(VGS-VT)**2
      ENDIF
C     WRITE(6,*)'VGS-VT=',V1,' I=',I
      VDS=VDS+0.25
5     CONTINUE
10    CONTINUE

C     CONVERT THE CURRENTS FROM AMPS TO MICROAMPS
      DO 15 I=1,27
      IDS(I)=IDS(I)*1E6
15    CONTINUE


      SUM=0.0
      DO 20 I=1,27
      COUNT=I
C     WRITE(6,*)'IDS(I)=',IDS(I),'A(I)=',A(I)
      B1(I)=(IDS(I)-A(I))**2
      SUM=SUM+B1(I)
20    CONTINUE
      SUM=SUM/27.0
      RMS1=SQRT(SUM)
```

```
      RETURN
      END


C     THIS SUBROUTINE CALCULATES A NEW BETA VALUE FOR THE CURVE FIT

      SUBROUTINE BETACALC(A,LAMMIN,VTMIN,TYPE,BETANEW,SNRRMS)

      DIMENSION A(27)
      DOUBLE PRECISION IDS1(27)
      REAL LAMBDA,BETANEW,LAMMIN,VTMIN
      CHARACTER TYPE

      LAMBDA=LAMMIN
      VT=VTMIN
      BETA=0.001

C     CALCULATE THE IDS VALUES FOR THE NEW LAMB. AND VT:
      WRITE(6,*)
      IF(TYPE.EQ.'D')THEN
      VGS=0.2
      ENDIF
      IF(TYPE.EQ.'E')THEN
      VGS=0.6
      ENDIF
      IF(TYPE.EQ.'S')THEN
      VGS=0.0
      ENDIF
      DO 10 J=1,3
      VDS=0.0
      IF(J.EQ.1)COUNT=1
      IF(J.EQ.2)THEN
      COUNT=10
      IF(TYPE.EQ.'S')THEN
      VGS=VGS-1.5
      ENDIF
      VGS=VGS-0.2
      ENDIF
      IF(J.EQ.3)THEN
      COUNT=19
      IF(TYPE.EQ.'S')THEN
      VGS=VGS-1.5
      ENDIF
      VGS=VGS-0.2
      ENDIF
      DO 5 I=COUNT,COUNT+8,1
      V1=VGS-VT
      IF(VDS.GE.0.0.AND.VDS.LE.V1)THEN
      IDS1(I)=BETA*VDS*(1+LAMBDA*VDS)*(2*(VGS-VT)-VDS)
      ELSE
      IDS1(I)=BETA*(1+LAMBDA*VDS)*(VGS-VT)**2
      ENDIF
C     WRITE(6,*)'VGS-VT=',V1,' I=',I
      VDS=VDS+0.25
5     CONTINUE
10    CONTINUE

C     CONVERT THE CURRENTS FROM AMPS TO MICROAMPS
```

```
        DO 15 I=1,27
        IDS1(I)=IDS1(I)*1E6
15      CONTINUE
C       THE OPEN STATEMENT WRITES DATA TO AN OUTPUT FILE
C       OPEN (UNIT=6,FILE='BETA.OUT',CARRIAGECONTROL='LIST', .
C    a        STATUS='NEW')
        WRITE(6,*)' I  CALCULATED   MEASURED'
        DO 16 I=1,27
        WRITE(6,17)I,IDS1(I),A(I)
17      FORMAT(2X,I2,2X,E8.2,2X,E8.2)
16      CONTINUE


C       THIS PART CALCULATES SLOPE, SIGMA, AND SNR
        TOTAL=0.0
        TOTAL1=0.0
        TOTAL2=0.0
        DO 20 I=1,27
        TOTAL=TOTAL+A(I)*IDS1(I)
        TOTAL1=TOTAL1+A(I)*A(I)
20      CONTINUE
        SLOPE=TOTAL/TOTAL1
        BETANEW=BETA/SLOPE


        SLOPESQ=SLOPE**2
        DO 25 I=1,27
        TOTAL2=TOTAL2+(IDS1(I)-SLOPE*A(I))**2
25      CONTINUE
        SIGSQ=TOTAL2/26
        SIG=SQRT(SIGSQ)
        WRITE(6,*)'THE SIGMA BETWEEN MEAS. AND CALC. IS:',SIG
        SNRRMS=10*ALOG10(SLOPESQ/SIGSQ)
        RETURN
        END

C       THIS SUBROUTINE FINDS THE MEAN AND STANDARD DEVIATION
        SUBROUTINE STAT(STATX,NUM,AVE,DEV)
        DIMENSION STATX(33,5)
        DIMENSION AVE(5)
        DIMENSION DEV(5)
        INTEGER NUM
C       DO NOT USE WAFER AVG I-V DATA IN STATISTICS. THEREFORE, DECREMENT
C       NUM COUNTER BY 1:
        NUM=NUM-1
        IF(NUM.EQ.0)THEN
        NUM=2
        WRITE(6,*)'VARIABLE NUM=0 IN SUBROUTINE STAT'
        ENDIF
        DO 2 I=1,5
        COUNT=I
        SUM=0.0
        TOTAL=0.0
        DO 5 T=1,NUM,1
C       WRITE(6,*)'STATX(T,COUNT)=',STATX(T,COUNT),' T=',T
        SUM=SUM+STATX(T,COUNT)
5       CONTINUE
C       AVE EQUALS THE MEAN
        AVE(COUNT)=SUM/NUM
```

```
C         WRITE(6,*)'AVE(COUNT)=',AVE(COUNT),' COUNT=',COUNT
          DO 10 T=1,NUM,1
          STATX(T,COUNT)=(STATX(T,COUNT)-AVE(COUNT))**2
          TOTAL=TOTAL+STATX(T,COUNT)
10        CONTINUE
          TOTAL1=TOTAL/NUM
          DEV(COUNT)=SQRT(TOTAL1)
2         CONTINUE
          RETURN
          END
```

APPENDIX II

This FORTRAN program (FITINT.FOR) is similar to the program in Appendix I except it allows the user to use Rs and Rd in the SPICE model statement instead of setting them to zero or a very small number. FITINT.FOR reads in the raw test data for up to 33 devices, calculates the IDS current from the intrinsic Vds and Vgs values. The intrinsic values can be calculated knowing the drain and source resistance and are used in all calculations involving Vds and Vgs. The program then calculates the Taguchi SNR ratio for nine different combinations of Vt and Lambda. The highest value is chosen, which reflects the best fit and a re-adjusted beta is found from the slope of the measured versus simulated Ids values. Lastly, the program calculates an optimum value for Vt and lambda from a quadratic curve fit from the response graphs. The program requires one input file with the measured data and outputs two files: FITINT.OUT contains the optimization progression and values for each individual device, while STATINT.OUT contains the over all mean and standard deviation for all three optimized parameters from the entire wafer.

```
C                ************* PROGRAM FITINT.FOR ***********

C        THIS PROGRAM IS SIMILAR TO FIT1.FOR EXCEPT IT USES RS AND RD TO FIND
C        THE INTRINSIC VALUES OF VGS AND VDS. THIS ALLOWS THE USER TO
C        INCORPORATE RS AND RD VALUES INTO THE SPICE SIMULATION.
C        THIS PROGRAM USES THE MEASURED RS AND RD VALUES AND CALCULATES THE
C        ACTUAL INTRINSIC VDS AND VGS VALUES WHERE VGSINT=VGSEXT-RS*IDS
C        THESE VALUES ARE THEN USED TO CALCULATE THE INTRINSIC IDS WHICH IS
C
C        ARRAY A1 AND A ARE THE MEASURED DATA
C        ARRAY B AND BNEW ARE THE CALCULATED DATA IN THIS PROGRAM
C        ARRAY C IS THE SIMULATED DATA FROM PRECISE


         DIMENSION A(27)
         DIMENSION A1(900,3)
         DIMENSION A2(900,3)
         DIMENSION R(33,2)
         DIMENSION B(243)
         DIMENSION BNEW(243)
         DIMENSION B1(27)
         DIMENSION B2(27)
         DIMENSION B3(27)
         DIMENSION B4(27)
         DIMENSION B5(27)
         DIMENSION B6(27)
         DIMENSION B7(27)
         DIMENSION B8(27)
         DIMENSION B9(27)
         DIMENSION SLOPES(9)
         DIMENSION SIGSQS(9)
         DIMENSION SNS(9)
         DIMENSION RMSS(9)
         DIMENSION PAR(9,2)
         DIMENSION DEVICE(33)
         DIMENSION STAT1(33,5)
         DIMENSION STAT2(33,5)
         DIMENSION AVE1(5)
         DIMENSION AVE2(5)
         DIMENSION DEV1(5)
         DIMENSION DEV2(5)
         DIMENSION VDS(27)
         DIMENSION VGS(27)

         DOUBLE PRECISION TOTAL,TOTAL1,TOTAL2,SLOPE,SIGSQ,SN,SIGSQ1,SIG,SLOPESQ
         REAL LAMBDA,LAMMIN,LAMBINIT,LAMBSTEP,LAMBAVE,M1VT,M2VT,M1LAMB,M2LAMB
         REAL OLAMMIN,RMS1,SNRRMS
         CHARACTER*27 DEVICE
         CHARACTER*20 HEAD
         CHARACTER*1 TYPE
         INTEGER COUNT1,COUNT2,NUM,P

         CALL LIB$INIT_TIMER
         IFIT=0

         WRITE(6,*)'ENTER NUMBER OF DEVICES'
         READ(5,FMT='(I2)')NUM
C        INCLUDE WAFER AVERAGED I-V DATA BY ADDING ADDITIONAL 27 POINTS
         NUM=NUM+1
```

```
      WRITE(6,*)'Enter a Capital D if DFET, E for EFET or S for SFET'
      READ(5,FMT='(A1)')TYPE

      INCREM=1


C     THE OPEN STATEMENT READS DATA IN FROM ACTUAL TEST DATA
      OPEN (UNIT=12,FILE='444-1-5EPCM.DAT',CARRIAGECONTROL='LIST',
C     OPEN (UNIT=12,FILE='TEST1.DAT',CARRIAGECONTROL='LIST',
     a       STATUS='OLD')
C     VDS=A1(I,1)
C     VGS=A1(I,2)
C     IDS=A1(I,3)
      DO 13 J=1,NUM
      READ(12,FMT='(A27)')DEVICE(J)
      READ(12,FMT='(F6.2,F22.2)')R(J,1),R(J,2)
      READ(12,FMT='(A20)')HEAD
      DO 14 I=INCREM,INCREM+26
      READ (12,FMT='(F5.2,F7.2,E14.3)')A1(I,1),A1(I,2),A1(I,3)
C     CONVERT THE MEASURED EXTRINSIC VDS AND VGS TO THEIR INTRINSIC VALUES
C     SO THEY CAN BE DIRECTLY COMPARED WITH THE PROGRAM GENERATED SIMULATED
C     VALUES. DON'T HAVE TO CONVERT IDS BECAUSE IT WILL REMAIN THE SAME.
C     KNOWING RS,RD AND IDS:
C     INTRINSIC VDS = A2(I,1)
C     INTRINSIC VGS = A2(I,2)
C     EXTRINSIC VDS = A1(I,1)
C     EXTRINSIC VGS = A1(I,2)
C     RD = R(J,1)
C     RS = R(J,2)
      IF(A1(I,3).LT.0.0)A1(I,3)=0.0
      A2(I,1)=A1(I,1)-A1(I,3)*(R(J,1)+R(J,2))
      A2(I,2)=A1(I,2)-A1(I,3)*R(J,2)
C     CONVERT THE IDS CURRENT FROM AMPS TO MICROAMPS
      A2(I,3)=A1(I,3)*1E6
14    CONTINUE
      INCREM=INCREM+27
13    CONTINUE




C     INITIAL VALUE AND STEP SIZE FOR LAMBDA AND VT

      IF(TYPE.EQ.'D')THEN
      VTINIT=-0.50
      VTSTEP=0.1
      LAMBINIT=0.01
      LAMBSTEP=0.5
      ENDIF

      IF(TYPE.EQ.'E')THEN
      VTINIT=0.05
      VTSTEP=0.1
      LAMBINIT=0.1
      LAMBSTEP=0.2
      ENDIF

      IF(TYPE.EQ.'S')THEN
      VTINIT=-4.5
```

```
      VTSTEP=.5
      LAMBINIT=0.01
      LAMBSTEP=0.2
      ENDIF

      VAVE=VTINIT+VTSTEP
      LAMBAVE=LAMBINIT+LAMBSTEP
      VTIN1=VTINIT
      VTIN2=VTINIT
      VTIN3=VTINIT

C     GENERATE PARAMETER ARRAY FOR EACH EXPERIMENT:
      DO 5 I=1,9
      IF(I.LE.3)THEN
      PAR(I,1)=VTIN1
      VTIN1=VTIN1+VTSTEP
      PAR(I,2)=LAMBINIT
      ENDIF
      IF(I.GE.4.AND.I.LE.6)THEN
      PAR(I,1)=VTIN2
      VTIN2=VTIN2+VTSTEP
      PAR(I,2)=LAMBINIT+LAMBSTEP
      ENDIF
      IF(I.GE.7.AND.I.LE.9)THEN
      PAR(I,1)=VTIN3
      VTIN3=VTIN3+VTSTEP
      PAR(I,2)=LAMBINIT+2*LAMBSTEP
      ENDIF
5     CONTINUE

C     CALCULATE SIMULATED IDS VALUES WITHOUT USING SIMULATOR
C     27 POINTS FOR 9 SIMULATIONS = 243 TOTAL SIMULATED IDS VALUES
C     NEED TO USE SAME INTRINSIC VALUES FOR VDS AND VGS AS MEASURED


C     ********START MAIN PROGRAM LOOP:********

      M=1
      P=0
C     THIS LOOP REPRESENTS EACH DEVICE
      DO 150 J=1,NUM
      BETA=0.001

C     THIS LOOP LOADS THE MEASURED IDS,VDS,VGS ARRAY FOR EACH DEVICE WHICH
C     IS AN ARRAY OF UP TO 900 ROWS AND CONVERTS IT TO AN ARRAY NUMBERED
C     ROW#1 TO ROW#27 FOR EACH DEVICE

      DO 20 I=M,M+26,1
      A(I-P)=A2(I,3)
C     PP=I-P
C     WRITE(6,*)'A(I-P);I-P=',PP
C     WRITE(6,*)'A(I)=',A(I),'  I=',I
      VDS(I-P)=A2(I,1)
      VGS(I-P)=A2(I,2)
20    CONTINUE


      K=1
      L=0
```

```
C       THIS LOOP CREATES THE 9 SETS OF EXPERIMENTS FOR EACH VT AND LAMBDA

        DO 15 N=1,9

C       THIS LOOP REPRESENTS THE ACTUAL 27 INT. VDS AND VGS FOR @ DEVICE
C       THE SAME MEASURED DATA FOR VGSINT AND VDSINT MUST BE USED
C       FOR ALL 9 EXPERIMENTS
C       THE B(I) ARRAY IS THE CALCULATED IDS USING INTRINSIC VDS AND VGS
C       FOR ALL 9 EXPERIMENTS. B(I) WILL HAVE 243 VALUES
C       ********  PAR(N,1)=VT  PAR(N,2)=LAMBDA ******

        DO 19 I=K,K+26
        V1=VGS(I-L)-PAR(N,1)
        IF(VDS(I-L).GE.0.0.AND.VDS(I-L).LT.V1)THEN
        B(I)=BETA*VDS(I-L)*(1+PAR(N,2)*VDS(I-L))
     a  *(2*(VGS(I-L)-PAR(N,1))-VDS(I-L))
        ELSE
        B(I)=BETA*(1+PAR(N,2)*VDS(I-L))*(VGS(I-L)-PAR(N,1))**2
        ENDIF
19      CONTINUE
        K=K+27
        L=27+27*(N-1)
15      CONTINUE

C       CONVERT THE SIMULATED IDS CURRENT FROM AMPS TO MICROAMPS
        DO 11 I=1,243
        BNEW(I)=B(I)*1E6
11      CONTINUE



C       THIS SUB BREAKS THE ARRAY B INTO 9 INDIVIDUAL ARRAYS FOR CALCULATIONS
C       B1 ARRAY IS EXPERIMENT #1
        DO 30 I=1,27
        B1(I)=BNEW(I)
30      CONTINUE
        CALL CALC(A,B1,RMS,SLOPE,SIG,SN)
        RMSS(1)=RMS
        SLOPES(1)=SLOPE
        SIGSQS(1)=SIG
        SNS(1)=SN

C       B2 ARRAY IS EXPERIMENT #2
        DO 35 I=1,27
        B2(I)=BNEW(I+27)
35      CONTINUE
        CALL CALC(A,B2,RMS,SLOPE,SIG,SN)
        RMSS(2)=RMS
        SLOPES(2)=SLOPE
        SIGSQS(2)=SIG
        SNS(2)=SN

C       B3 ARRAY IS EXPERIMENT #3
        DO 40 I=1,27
        B3(I)=BNEW(I+54)
40      CONTINUE
        CALL CALC(A,B3,RMS,SLOPE,SIG,SN)
        RMSS(3)=RMS
        SLOPES(3)=SLOPE
```

```
      SIGSQS(3)=SIG
      SNS(3)=SN
C     B4 ARRAY IS EXPERIMENT #4
      DO 45 I=1,27
      B4(I)=BNEW(I+81)
45    CONTINUE
      CALL CALC(A,B4,RMS,SLOPE,SIG,SN)
      RMSS(4)=RMS
      SLOPES(4)=SLOPE
      SIGSQS(4)=SIG
      SNS(4)=SN
C     B5 ARRAY IS EXPERIMENT #5
      DO 50 I=1,27
      B5(I)=BNEW(I+108)
50    CONTINUE
      CALL CALC(A,B5,RMS,SLOPE,SIG,SN)
      RMSS(5)=RMS
      SLOPES(5)=SLOPE
      SIGSQS(5)=SIG
      SNS(5)=SN
C     B6 ARRAY IS EXPERIMENT #6
      DO 55 I=1,27
      B6(I)=BNEW(I+135)
55    CONTINUE
      CALL CALC(A,B6,RMS,SLOPE,SIG,SN)
      RMSS(6)=RMS
      SLOPES(6)=SLOPE
      SIGSQS(6)=SIG
      SNS(6)=SN
C     B7 ARRAY IS EXPERIMENT #7
      DO 60 I=1,27
      B7(I)=BNEW(I+162)
60    CONTINUE
      CALL CALC(A,B7,RMS,SLOPE,SIG,SN)
      RMSS(7)=RMS
      SLOPES(7)=SLOPE
      SIGSQS(7)=SIG
      SNS(7)=SN
C     B8 ARRAY IS EXPERIMENT #8
      DO 65 I=1,27
      B8(I)=BNEW(I+189)
65    CONTINUE
      CALL CALC(A,B8,RMS,SLOPE,SIG,SN)
      RMSS(8)=RMS
      SLOPES(8)=SLOPE
      SIGSQS(8)=SIG
      SNS(8)=SN
C     B9 ARRAY IS EXPERIMENT #9
      DO 70 I=1,27
      B9(I)=BNEW(I+216)
70    CONTINUE
      CALL CALC(A,B9,RMS,SLOPE,SIG,SN)
      RMSS(9)=RMS
      SLOPES(9)=SLOPE
```

```
      SIGSQS(9)=SIG
      SNS(9)=SN

C     THE OPEN STATEMENT WRITES DATA TO A FILE
      OPEN (UNIT=60,FILE='TEMP.OUT',CARRIAGECONTROL='LIST',
     a      STATUS='NEW')
      WRITE(60,*)
      WRITE(60,*)
      WRITE(60,*)'-----------------------------------------------------
     B-----------------'
      WRITE(60,*)
      WRITE(60,FMT='(A27)')DEVICE(J)
      WRITE(60,999)
999   FORMAT(/,12HEXPERIMENT #,3X,6HLAMBDA,4X,2HVt,7X,3HRMS,
     B 7X,5HSLOPE,5X,5HSIGMA,5X,3HSNR)
      DO 1001 I=1,9
      WRITE(60,1000)I,PAR(I,2),PAR(I,1),RMSS(I),SLOPES(I),SIGSQS(I),SNS(I)
1000  FORMAT(I7,8X,F5.2,3X,F6.2,3X,F9.3,1X,F7.3,2X,F9.3,2X,F7.3)
1001  CONTINUE
C     FIND MAXIMUM VALUE OF SNR ARRAY
      SNRMAX=AMAX1(SNS(1),SNS(2),SNS(3),SNS(4),SNS(5),SNS(6),SNS(7),
     E SNS(8),SNS(9))
C     WRITE(6,*)'SNRMAX=',SNRMAX
      DO 71 I=1,9
      COUNT=I
      TEMP=SNS(I)
      TEMP1=ABS(TEMP-SNRMAX)
      IF(TEMP1.LE.0.01)GO TO 72
71    CONTINUE
72    I=COUNT
      BETA=1.0E-3/SLOPES(I)

C     STORE THESE VALUES FOR EACH DEVICE TO CALCULATE OVERALL MEAN AND SIGMA
C     VT=PAR(I,1)    LAMBDA=PAR(I,2)
      NUM1=J
      STAT1(NUM1,1)=SNRMAX
      STAT1(NUM1,2)=PAR(I,1)
      STAT1(NUM1,3)=PAR(I,2)
      STAT1(NUM1,4)=BETA

      WRITE(60,73)SNRMAX,PAR(I,1),PAR(I,2),BETA
73    FORMAT(//,1X,8HMAX.SNR=,F7.3,2X,3HVT=,F5.2,2X,7HLAMBDA=,E9.3,
     E 2X,5HBETA=,E9.3)
C     WRITE(60,*)'MAXIMUM SNR=',SNRMAX,'VT=',PAR(I,1),'LAMBDA=',PAR(I,2)

C     CALCULATE THE RMS ERROR BETWEEN THE MEASURED DATA AND THE NEW
C     SPICE PARAMETERS FROM THE MAXIMUM SNR
      LAMBDA=PAR(I,2)
      VT=PAR(I,1)
      CALL RMSSUB1(A,VDS,VGS,LAMBDA,VT,BETA,RMS1)
      WRITE(60,*)'THE RMS ERROR USING THESE VALUES IS (uA):',RMS1
      STAT1(NUM1,5)=RMS1

C     SEE IF THE DATA MEETS THE CRITERIA FOR A QUAD. FIT TO THE RESP. CURVES

C     FIND AVERAGE SNR FOR EACH  VT AND COEFFICIENTS OF QUAD. EQ.
      Y1=(SNS(1)+SNS(4)+SNS(7))/3
      Y2=(SNS(2)+SNS(5)+SNS(8))/3
      Y3=(SNS(3)+SNS(6)+SNS(9))/3
      AVT=(Y3+Y1-2*Y2)/(2*VTSTEP**2)
```

```
      BVT=(Y3-Y1)/(2*VTSTEP)
      CVT=Y2
C     FIND IF THE RESPONSE CURVES ARE LOCAL MINIMA OR LOCAL MAXIMA
      M1VT=(Y2-Y1)/VTSTEP
      M2VT=(Y3-Y2)/VTSTEP
      IF(M1VT.GT.M2VT)THEN
      FLAG=1.0
      ELSE
      FLAG=0.0
      ENDIF

      WRITE(60,*)
      WRITE(60,*)
      WRITE(60,*)
      WRITE(60,*)'THE COEFFICIENTS OF A QUADRACTIC EQUATION Ax**2+Bx+C=y'
      WRITE(60,*)'TO THE RESPONSE CURVES WHERE x=step size and'
      WRITE(60,*)'dx=-B/2A (MAXIMUM POINT ON CURVE FROM AVE. x) ARE:'
      VTMIN=-BVT/(2*AVT)
      VTMINO=VTMIN
      WRITE(60,75)
75    FORMAT(/,5X,9HPARAMETER,6X,1HA,9X,1HB,9X,1HC,9X,2Hdx)
      IF(FLAG.EQ.1.0)THEN
      WRITE(60,76)AVT,BVT,CVT,VTMIN
76    FORMAT(8X,2HVT,6X,E9.2,2X,E9.2,2X,E9.2,2X,E9.2,4X,10H LOCAL MAX)
      VTMIN=VAVE+VTMIN
      ELSE
      WRITE(60,760)AVT,BVT,CVT,VTMIN
760   FORMAT(8X,2HVT,6X,E9.2,2X,E9.2,2X,E9.2,2X,E9.2,4X,10H LOCAL MIN)
      VTMIN=0.0
      ENDIF

C     FIND AVERAGE SNR FOR EACH LAMBDA AND COEFFICIENTS OF QUAD. EQ.
      Y1=(SNS(1)+SNS(2)+SNS(3))/3
      Y2=(SNS(4)+SNS(5)+SNS(6))/3
      Y3=(SNS(7)+SNS(8)+SNS(9))/3
      ALAMB=(Y3+Y1-2*Y2)/(2*LAMBSTEP**2)
      BLAMB=(Y3-Y1)/(2*LAMBSTEP)
      CLAMB=Y2
      AVELAMSNR=(Y1+Y2+Y3)/3
C     THE MINIMUM POINT ON THE QUAD. IS THE FIRST DERIVATIVE
      LAMMIN=-BLAMB/(2*ALAMB)
      OLAMMIN=LAMMIN
C     FIND IF THE RESPONSE CURVES ARE LOCAL MINIMA OR LOCAL MAXIMA
      M1LAMB=(Y2-Y1)/LAMBSTEP
      M2LAMB=(Y3-Y2)/LAMBSTEP
      IF(M1LAMB.GT.M2LAMB)THEN
      FLAG=1.0
      ELSE
      FLAG=0.0
      ENDIF

      IF(FLAG.EQ.1.0)THEN
      WRITE(60,77)ALAMB,BLAMB,CLAMB,LAMMIN
77    FORMAT(6X,6HLAMBDA,4X,E9.2,2X,E9.2,2X,E9.2,2X,E9.2,4X,10H LOCAL MAX)
      LAMMIN=LAMMIN+LAMBAVE
      ELSE
      WRITE(60,770)ALAMB,BLAMB,CLAMB,LAMMIN
770   FORMAT(6X,6HLAMBDA,4X,E9.2,2X,E9.2,2X,E9.2,2X,E9.2,4X,10H LOCAL MIN)
      LAMMIN=0.0
```

```
      ENDIF
C     BEFORE PRINTING OUT THE MAXIMUM SNR VALUES FROM THE CURVE FIT
C     BOTH VARIABLES MUST BE LOCAL MAX AND WITHIN THE SWEEP LIMITS

C     TEST THAT BOTH ARE LOCAL MAX
      IF(M1LAMB.GT.M2LAMB.AND.M1VT.GT.M2VT)THEN
      VTMIN1=VTINIT
      VTMAX1=VTINIT+(2*VTSTEP)
      ZLAMMIN=LAMBINIT
      ZLAMMAX=LAMBINIT+(2*LAMBSTEP)
      ELSE
      WRITE(60,*)
      WRITE(60,*)'CURVE FIT VT AND LAMBDA ARE NOT BOTH LOCAL MAX'
      IF(J.EQ.NUM)THEN
      IFIT=IFIT+1
      ENDIF
      GO TO 80
      ENDIF

      IF(VTMIN.GE.VTMIN1.AND.VTMIN.LE.VTMAX1)THEN
      GO TO 78
      ELSE
      WRITE(60,*)'CURVE FIT VT IS OUT OF SWEEP RANGE'
      IF(J.EQ.NUM)THEN
      IFIT=IFIT+1
      ENDIF
      GO TO 80
      ENDIF

78    IF(LAMMIN.GE.ZLAMMIN.AND.LAMMIN.LE.ZLAMMAX)THEN
      IFIT=IFIT+1
C     NOW SOLVE THE SNR FOR THE FITTED VALUES OF VT AND LAMBDA
      SNRVT=AVT*(VTMINO**2)+BVT*VTMINO+CVT
      SNRLAMB=ALAMB*(OLAMMIN**2)+BLAMB*OLAMMIN+CLAMB
      SNRFIT=SNRVT+SNRLAMB-(AVELAMSNR)
      WRITE(60,*)'THE ESTIMATED SNR USING CURVE FIT VT AND LAMB. IS:',SNRFIT


C     SOLVE NEW BETA FOR CURVE FIT:
      CALL BETACALC(A,VGS,VDS,LAMMIN,VTMIN,BETANEW,SNRRMS)
      BETA=BETANEW
      STAT2(IFIT,1)=SNRRMS
      STAT2(IFIT,2)=VTMIN
      STAT2(IFIT,3)=LAMMIN
      STAT2(IFIT,4)=BETA
      WRITE(60,79)VTMIN,LAMMIN,BETA
79    FORMAT(//,1X,21HQUADRACTIC CURVE FIT:,1X,3HVT=,F6.3,1X,
     B 7HLAMBDA=,F5.3,1X,5HBETA=,E9.3)
      WRITE(60,*)
C     CALCULATE THE RMS ERROR OF THE CURVE FIT USING THE NEW BETA VALUE
      CALL RMSSUB1(A,VDS,VGS,LAMMIN,VTMIN,BETA,RMS2)
      WRITE(60,*)'THE RMS ERROR USING THESE VALUES IS (uA):',RMS2
      WRITE(60,*)'THE SNR USING THE NEW BETA IS:',SNRRMS
      STAT2(IFIT,5)=RMS2
      ELSE
      WRITE(60,*)'CURVE FIT LAMBDA IS OUT OF SWEEP RANGE'
      IF(J.EQ.NUM)THEN
      IFIT=IFIT+1
      ENDIF
```

```
      ENDIF
80    M=M+27
      P=27+27*(J-1)
150   CONTINUE


C     CALCULATE THE STATISTICS FOR ALL DEVICES AND PRINT TO A FILE
C     THE OPEN STATEMENT WRITES DATA TO AN OUTPUT FILE
      OPEN (UNIT=60,FILE='LIST.OUT',CARRIAGECONTROL='LIST',
     a      STATUS='NEW')
      DO 120 I=1,IFIT
      WRITE(60,121)STAT2(I,2),STAT2(I,3),STAT2(I,4)
121   FORMAT(E8.2,2X,E8.2,2X,E9.3)
120   CONTINUE

C     CALCULATE THE STATISTICS FOR ALL DEVICES AND PRINT TO A FILE
C     THE OPEN STATEMENT WRITES DATA TO AN OUTPUT FILE
      OPEN (UNIT=60,FILE='STAT.OUT',CARRIAGECONTROL='LIST',
     a      STATUS='NEW')
      CALL STAT(STAT1,NUM,AVE1,DEV1)
      WRITE(60,*)
      WRITE(60,*)
      WRITE(60,*)
      WRITE(60,*)'THE AVERAGE AND SIGMA FROM THE ACTUAL EXPERIMENTS:'
      WRITE(60,85)
85    FORMAT(/,10X,3HSNR,9X,2HVT,6X,6HLAMBDA,6X,4HBETA,6X,3HRMS,6X,
     B 8H#DEVICES)
      WRITE(60,95)AVE1(1),AVE1(2),AVE1(3),AVE1(4),AVE1(5),NUM
95    FORMAT(7HAVERAGE,2X,F6.2,4X,F6.2,4X,F6.2,6X,E8.2,2X,F8.2,6X,I2)
      WRITE(60,100)DEV1(1),DEV1(2),DEV1(3),DEV1(4),DEV1(5)
100   FORMAT(5HSIGMA,4X,F6.2,4X,E8.2,2X,E8.2,4X,E8.2,1X,F8.2)

      CALL STAT(STAT2,IFIT,AVE2,DEV2)
      WRITE(60,*)
      WRITE(60,*)
      WRITE(60,*)
      WRITE(60,*)'THE AVERAGE AND SIGMA FROM THE QUAD. CURVE FIT:'
      WRITE(60,105)
105   FORMAT(/,10X,3HSNR,9X,2HVT,6X,6HLAMBDA,6X,4HBETA,6X,3HRMS,6X,
     B 8H#DEVICES)
      WRITE(60,110)AVE2(1),AVE2(2),AVE2(3),AVE2(4),AVE2(5),IFIT
110   FORMAT(7HAVERAGE,2X,F6.2,4X,F6.3,4X,F6.3,6X,E8.2,2X,F6.2,6X,I2)
      WRITE(60,115)DEV2(1),DEV2(2),DEV2(3),DEV2(4),DEV2(5)
115   FORMAT(5HSIGMA,4X,F6.2,4X,E8.2,2X,E8.2,4X,E8.2,1X,F6.2)
      WRITE(60,*)
      WRITE(60,*)
      WRITE(60,*)


      CALL LIB$SHOW_TIMER
      STOP
      END


C     THIS SUBROUTINE CALCULATES THE SLOPE OF THE LINE THROUGH THE ORIGIN
C     ARRAY A IS THE MEASURED CURRENT AND ARRAY DAT IS THE CALCULATED
```

```
      SUBROUTINE CALC(A,SIM,RMS,SLOPE,SIG,SN)
      DIMENSION SIM(27)
      DIMENSION A(27)
      DOUBLE PRECISION RMS1(27),SUM
      REAL SN,SLOPESQ,SIGSQ
C     THIS PART CALCULATES THE RMS ERROR:
      SUM=0.0
      DO 200 I=1,27
      RMS1(I)=(SIM(I)-A(I))**2
      SUM=SUM+RMS1(I)
200   CONTINUE
      SUM=SUM/27.0
      RMS=SQRT(SUM)

C     THIS PART CALCULATES SLOPE, SIGMA AND SNR
      TOTAL=0.0
      TOTAL1=0.0
      TOTAL2=0.0
      DO 400 I=1,27
C     WRITE(6,*)SIM(I)
      TOTAL=TOTAL+A(I)*SIM(I)
      TOTAL1=TOTAL1+A(I)*A(I)
400   CONTINUE
      IF(TOTAL1.EQ.0)THEN
      TOTAL1=0.001
      WRITE(6,*)'TOTAL1=0 IN SUBROUTINE CALC'
      ENDIF
      SLOPE=TOTAL/TOTAL1
      SLOPESQ=SLOPE**2
      DO 500 I=1,27
      TOTAL2=TOTAL2+(SIM(I)-SLOPE*A(I))**2
500   CONTINUE
      SIGSQ=TOTAL2/26
      SIG=SQRT(SIGSQ)
      IF(SIGSQ.EQ.0.0)THEN
      SIGSQ=0.001
      WRITE(6,*)'SIGSQ=0 IN SUBROUTINE CALC'
      ENDIF
      IF(SLOPESQ.EQ.0.0)THEN
      SLOPESQ=0.001
      WRITE(6,*)'SLOPESQ=0 IN SUBROUTINE CALC'
      ENDIF
      SN=10*ALOG10(SLOPESQ/SIGSQ)
      RETURN
      END

C     THIS SUBROUTINE READS IN TWO DATA FILES WITH 12 ROWS AND TWO COLUMNS
C     AND FINDS THE RMS ERROR OF THE SECOND COLUMN (IDSS).

      SUBROUTINE RMSSUB1(A,VDS,VGS,LAMBDA,VT,BETA,RMS1)
      DIMENSION A(27)
      DIMENSION VDS(27)
      DIMENSION VGS(27)
      DIMENSION B1(27)
      DOUBLE PRECISION IDS(27)
      REAL SUM,LAMBDA,VT,BETA,RMS1
C     WRITE(6,*)'VT=',VT,' LAMB.=',LAMBDA,' BETA=',BETA
      DO 19 I=1,27
      V1=VGS(I)-VT
C     WRITE(6,*)'V1=',V1,'   I=',I
```

```
        IF(VDS(I).GE.0.0.AND.VDS(I).LT.V1)THEN
        IDS(I)=BETA*VDS(I)*(1+LAMBDA*VDS(I))*(2*(VGS(I)-VT)-VDS(I))
        ELSE
        IDS(I)=BETA*(1+LAMBDA*VDS(I))*(VGS(I)-VT)**2
        ENDIF
C       CONVERT THE SIMULATED IDS CURRENT FROM AMPS TO MICROAMPS
        IDS(I)=IDS(I)*1E6
C       WRITE(6,*)'IDS(I)=',IDS(I),'    A(I)=',A(I)
C       WRITE(6,*)
19      CONTINUE


C       THE OPEN STATEMENT WRITES DATA TO AN OUTPUT FILE COMPARING
C       MEASURED AND SIMULATED CURRENTS FOR EACH DEVICE
C       THERE SHOULD BE 243 VALUES FOR BNEW AND 27 FOR A(I)
        OPEN (UNIT=61,FILE='COMPARE.OUT',CARRIAGECONTROL='LIST',
     a        STATUS='NEW')
c       WRITE(61,*)'DEVICE#=',J
        WRITE(61,*)'VDS  MEASURED I   CALCULATED I'
        V=0.0
        DO 12 I=1,27
        WRITE(61,16)V,A(I),IDS(I)
16      FORMAT(F5.2,E12.4,6X,E12.4)
        V=V+0.25
        IF(V.GT.2.00)V=0.0
12      CONTINUE


        SUM=0.0
        DO 20 I=1,27
        B1(I)=(IDS(I)-A(I))**2
        SUM=SUM+B1(I)
20      CONTINUE
        SUM=SUM/27.0
        RMS1=SQRT(SUM)

        RETURN
        END



C       THIS SUBROUTINE CALCULATES A NEW BETA VALUE FOR THE CURVE FIT

        SUBROUTINE BETACALC(A,VGS,VDS,LAMMIN,VTMIN,BETANEW,SNRRMS)

        DIMENSION A(27)
        DIMENSION VGS(27)
        DIMENSION VDS(27)
        DOUBLE PRECISION IDS1(27)
        REAL LAMBDA,BETANEW,LAMMIN,VTMIN


        LAMBDA=LAMMIN
        VT=VTMIN
        BETA=0.001

C       CALCULATE THE IDS VALUES FOR THE NEW LAMB. AND VT:
        WRITE(6,*)
```

```
C       THE OPEN STATEMENT WRITES DATA TO AN OUTPUT FILE
C       OPEN (UNIT=6,FILE='BETA.OUT',CARRIAGECONTROL='LIST',
C     a       STATUS='NEW')
C       WRITE(6,*)' I  CALCULATED   MEASURED'
C       DO 16 I=1,27
C       WRITE(6,17)I,IDS1(I),A(I)
C17     FORMAT(2X,I2,2X,E8.2,2X,E8.2)
C16     CONTINUE
        DO 19 I=1,27
        V1=VGS(I)-VT
        IF(VDS(I).GE.0.0.AND.VDS(I).LT.V1)THEN
        IDS1(I)=BETA*VDS(I)*(1+LAMBDA*VDS(I))*(2*(VGS(I)-VT)-VDS(I))
        ELSE
        IDS1(I)=BETA*(1+LAMBDA*VDS(I))*(VGS(I)-VT)**2
        ENDIF
C       CONVERT THE SIMULATED IDS CURRENT FROM AMPS TO MICROAMPS
        IDS1(I)=IDS1(I)*1E6
19      CONTINUE


C       THIS PART CALCULATES SLOPE, SIGMA, AND SNR
        TOTAL=0.0
        TOTAL1=0.0
        TOTAL2=0.0
        DO 20 I=1,27
        TOTAL=TOTAL+A(I)*IDS1(I)
        TOTAL1=TOTAL1+A(I)*A(I)
20      CONTINUE
        SLOPE=TOTAL/TOTAL1
        BETANEW=BETA/SLOPE


        SLOPESQ=SLOPE**2
        DO 25 I=1,27
        TOTAL2=TOTAL2+(IDS1(I)-SLOPE*A(I))**2
25      CONTINUE
        SIGSQ=TOTAL2/26
        SIG=SQRT(SIGSQ)
C       WRITE(6,*)'THE SIGMA BETWEEN MEAS. AND CALC. IS:',SIG
        SNRRMS=10*ALOG10(SLOPESQ/SIGSQ)
        RETURN
        END

C       THIS SUBROUTINE FINDS THE MEAN AND STANDARD DEVIATION
        SUBROUTINE STAT(STATX,NUM,AVE,DEV)
        DIMENSION STATX(33,5)
        DIMENSION AVE(5)
        DIMENSION DEV(5)
        INTEGER NUM
C       DO NOT USE WAFER AVG I-V DATA IN STATISTICS. THEREFORE, DECREMENT
C       NUM COUNTER BY 1:
        NUM=NUM-1
        IF(NUM.EQ.0)THEN
        NUM=2
        WRITE(6,*)'VARIABLE NUM=0 IN SUBROUTINE STAT'
        ENDIF
        DO 2 I=1,5
        COUNT=I
        SUM=0.0
```

```
      TOTAL=0.0
      DO 5 T=1,NUM,1
      SUM=SUM+STATX(T,COUNT)
5     CONTINUE
C     AVE EQUALS THE MEAN
      AVE(COUNT)=SUM/NUM
      DO 10 T=1,NUM,1
      STATX(T,COUNT)=(STATX(T,COUNT)-AVE(COUNT))**2
      TOTAL=TOTAL+STATX(T,COUNT)
10    CONTINUE
      TOTAL1=TOTAL/NUM
      DEV(COUNT)=SQRT(TOTAL1)
2     CONTINUE
      RETURN
      END
```

REFERENCES

[1] L.Waller,"Is GaAs Ready for the Mass Markets", ASIC Technology and News, Vol. 2, No. 11, p1, March 1991.

[2] G.Watson,"Gigahertz Nets Spur Gallium Arsenide ICs", IEEE Institute, Vol. 15, No. 6, pp. 1-3, July/August 1991.

[3] R.Sadler, et. al.,"A Manufacturable 0.4 μm Process for High-Performance LSI Circuits", IEEE GaAs IC Symposium, pp.63-66, October 1990.

[4] L.W.Nagel,"SPICE2: A Computer Program to Simulate Semiconductor Circuits", Electronics Research Laboratory, University of California, Berkeley, Me. No. EPL-M520, May 9, 1975.

[5] C. Kocot and C.A.Stolte, "Backgating in GaAs MESFET's", IEEE Transactions on Electron Devices, Vol. ED-29, pp. 1059-1064, 1982.

[6] P. Canfield, J. Medinger, and L. Forbes, "Buried-channel GaAs MESFET's with frequency dependent output conductance", IEEE Electron Device Letters, Vol. EDL-8, p. 88, 1987.

[7] M.Lee, et. al., "An Analytical Self-Backgating GaAs MESFET Model Including Deep-Level Traps", IEEE International Electron Device Meeting, pp. 316-318, December 1989.

[8] P.Canfield, S.Lam, D. Allstot, "Modeling of Frequency and Temperature Effects in GaAs MESFET's", IEEE Journal of Solid-State Circuits, Vol. 25, No. 1, pp. 299-306, February 1990.

[9] A.McCamant, G.McCormack, D.Smith, "An Improved GaAs MESFET Model for SPICE", IEEE Transactions on Microwave Theory and Techniques, Vol. 38, No. 6, pp. 822-824, June 1990.

[10] H.Shichman and D.A.Hodges,"Modeling and Simulation of Insulated-Gate Field-Effect Transistor Switching Circuits", IEEE Journal of Solid-State Circuits, Vol. SC-3, pp. 285-289, September 1968.

[11] 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.

[12] H.Statz, et. al. ,"GaAs FET Device and Circuit Simulation in SPICE", IEEE Transactions on Electron Devices, Vol. ED-34, No. 2, pp. 160-169, February 1987.

[13] G.Taguchi,"System of Experimental Design", Kraus International Publications, White Plains, New York, 1987.

[14] M.Phadke,"Quality Engineering Using Robust Design", Prentice Hall, Englewood Cliffs, New Jersey, 1989.

[15] T.Kacprzak and A.Materka,"Compact DC Model of GaAs FET's for Large Signal Computer Calculation", IEEE Journal of Solid-State Circuits, Vol. 18, pp. 211-213, April 1983.

[16] A.Materka and T.Kacprzak,"Computer Calculation of Large-Signal GaAs FET Amplifier Characteristics", IEEE Transactions on Microwave Theory and Techniques, Vol. MTT-33, No. 2, pp. 129-134, February 1985.

[17] W.R.Curtice and M.Ettenberg,"A nonlinear GaAs FET Model for use in the Design of Output Circuits for Power Amplifiers", IEEE Transactions on Microwave Theory and Techniques, Vol. MTT-33, pp. 1383-1394, December 1985.

[18] W.R.Curtice,"GaAs MESFET Modeling and Nonlinear CAD", IEEE Transactions on Microwave Theory and Techniques, Vol. MTT-36, No. 2, pp. 220-230, February 1988.

[19] L.Larson, "An Improved CaAs MESFET Equivalent Circuit Model for Analog Integrated Circuit Applications", IEEE Journal of Solid-State Circuits, Vol. 22, No. 4, pp. 567-574, August 1987.

[20] K.M.Brown and J.E.Dennis,"Derivative Free Analogues of Levenberg-Marquardt and Gauss Algorithms for Nonlinear Least Squares Approximation", Numer. Math., Vol. 18, pp. 289-297, 1972.

[21] K.Doganis and R.Dutton, "SUXES-Stanford University Extractor of Model Parameters (user's manual)", Stanford Electronics Laboratories, Stanford University, Stanford Calf., November 1982.

[22] K.Doganis, D.L.Scharfetter,"General Optimization and Extraction of IC Device Model Parameters", IEEE Transactions on Electron Devices, Vol. ED-30, No. 9, September 1983.

[23] H.Fukui, "Determination of the Basic Device Parameters of a GaAs MESFET", Bell System Technical Journal, Vol. 58, pp.771-797, March 1979.

[24] Hewelet Packard Application note No. 356 pp. 16-19.

[25] W.A.White and M.R.Namordi,"GaAs MESFET model adds life to SPICE", Microwaves and RF, pp.197-200, September 1984.

[26] P.Cox, et. al., "Statistical Modeling for Efficient Parametric Yield Estimation of MOS VLSI Circuits", IEEE Transactions on Electron Devices, Vol. ED-32, No. 2, pp. 471-478, February 1985.

[27] S.Inohira, et. al., "A Statistical Model Including Parameter Matching for Analog Integrated Circuits Simulation", IEEE Transactions on Electron Devices, Vol. ED-32, No. 10, pp. 2177-2184, October 1985.

[28] W.Maly, A.Strojwas, S.Director, "VLSI Yield Prediction and Estimation: A Unified Framework", IEEE Transactions on Computer-Aided Design, Vol. CAD-5, No. 1, pp. 114-130, January 1986.

[29] L.Sanders, "Combining Analog Simulation Techniques Optimizes Designs", EDN, pp. 237-242, June 21, 1990.

[30] T.Mukherjee, and R.Carley, "Rapid Yield Estimation as a Computer Aid for Analog Circuit Design", IEEE Journal of Solid-State Circuits, Vol. 26, No. 3, pp. 291-299, March 1991.

[31] M.Bolt, M.Rocchi, J.Engel, "Realistic Statistical Worst-Case Simulations of VLSI Circuits", IEEE Transactions on Semiconductor Manufacturing, Vol. 4, No. 3, pp. 193-198, August 1991.

[32] S.Nassif, A.Strojwas, S.Director, "A Methodology for Worst-case Analysis of Integrated Circuits", IEEE Transactions on Computer-Aided Design, Vol. CAD-5, No. 1, pp. 104-113, January 1986.

[33] C.Michael and M.Ismail, "Statistical Modeling of Device Mismatch for Analog MOS Integrated Circuits", IEEE Journal of Solid-State Circuits, Vol. 27, No. 2, pp. 154-165, February 1992.

[34] D.Divekar, "DC Statistical Circuit Analysis for Bipolar IC's Using Parameter Correlations-An Experimental Example", IEEE Transactions on Computer-Aided Design, Vol. CAD-3, January 1984.

[35] D.Haigh and J. Everard,"GaAs Technology and its Impact on Circuits and Systems" Peter Peregrinus Ltd., London, England,p 225, 1989.

VITA

The author was born in Philadelphia Pennsylvania in 1961. He received his B.S. degree in Electrical Engineering in 1983 and the M.E.E in 1985, both from the University of Louisville. The author worked at General Electric's Electronics Laboratory in Syracuse New York from 1985 to 1987 as a digital GaAs IC design engineer. In 1987 he began work at ITT-GTC as a GaAs digital/analog IC design engineer. ITT-GTC is located in Roanoke Virginia and consists of approximately 100 employees. The charter of ITT-GTC is to develop GaAs ICs and modules for the various units of ITT Defense, which will be required for next generation military systems. The author is presently a member of the technical staff at ITT-GTC and has published over a dozen professional papers on the design of a variety of GaAs ICs and has received one patent. He is a member of the IEEE and is Vice-chairman of the IEEE Virginia Mountain Section.