Automated MOSFET Parameter Extraction

by

Jerome C. Lui

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degrees of

Bachelor of Science in Electrical Science and Engineering and Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1995

Author ....
Department of Electrical Engineering and Computer Science
May 26, 1995.

Certified by ....
Professor James E. Chung
Department of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by ....
Professor F. R. Morgenthaler

Automated MOSFET Parameter Extraction

by

Jerome C. Lui

Submitted to the Department of Electrical Engineering and Computer Science on

May 26, 1995

In partial fulfillment of the requirements for the degrees of Bachelor of Science in Electrical Science and Engineering and Master of Engineering in Electrical Engineering and Computer Science

Abstract

The goal of this Thesis project is to set up an analysis system which will analyze the I-V measurement data obtained from an automatic probe system which performs measurements on CMOS transistors. The purpose of the analysis is to extract parameters to characterize a process and provide a qualitative basis for CMOS device-design issues. Parameters extracted in this system include: the effective channel length and width of devices, the threshold voltage with the use of different algorithms, the subthreshold slope, the peak transconductance, the substrate doping concentration and the flat-band voltage.

Thesis Supervisor: Professor James E. Chung

Acknowledgments

I am greatly indebted to Professor James E. Chung for providing me an opportunity to participate in this research. Without his guidance and support this thesis would not exist. It has been an honor and pleasure to have been supervised by Professor Chung.

I would also like to express my gratitude to Eric Chang, Rajesh Divecha, Robert Ha, Seok-Won Kim, Wenjie Jiang, Huy Le, Daniel Maung, Beniyam Menberu, Jocelyn Nee, Jee-Hoon Yap, and Jung Yoon. Their help and suggestions went beyond all expectations, and have greatly contributed to this work.

I am also indebted to the following people, whose company has immensely enriched my experience at MIT:

• To all members of my family, for their love and support throughout my life. All the members of my family have greatly sacrificed for the sake of this thesis. I hope that they will view this thesis as their achievement.

• To Esther, for being a special partner and friend for 3.5 years. You have been a great source of support, friendship, encouragement and advice. Best of luck in the future years.

• To Albert, Yuk, Andy, Bernard, Richard, Felix and Tony "Ben-Chow" Wong for being my best friends in MIT. You have been a constant source of support, friendship, advice and help during my years at MIT. Sorry for making fun on you most of the time.

• To Jocelyn, Vinci, Jenny, Vivian, Phoebe, Christina, Susan, Annie and Mary for being my best female friends in MIT. Thanks for all your care and help. Also thanks for listening to my boring words when I was in trouble.

• To King, Katherine, Sun-Man, and Leo for being my project partners for many different classes. You have made projects and classes more enjoyable.

• To WMBR for giving me a chance to be the producer of the show "Touch of Hong Kong". This is certainly a life-time experience, and I would not forget all the greatest moment in the radio station. Also thanks to the staffs of "Touch of Hong Kong".

• To HKSS for providing many of my unforgettable activities in MIT including IM sports. Also thanks everyone in HKSS for helping me out in "Anthony Wong plus others Concert", an event of Hong Kong Week 1992.

• To my friends overseas, especially Philip, Randy, Ceypo, Eddie, Cacin, Andrew, Angelina, Kelly, Amy, Christine and Leon, for their help and support in many different ways.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Probing is the process of measuring current and voltage data from a device by making electrical contact with that device using a prober. Prior to this project, manual probe system has been used to acquire device measurement data. This was done using the *iv_main* program; analysis of the measurements was done using the *iv_anal* program. Both of these programs are written in HT BASIC.

When probing is performed using the manual probe system, the user has to adjust the prober manually for each device. In order to extract appropriate I-V data needed to characterize the device, the user will then use the prober to step through the specified devices with the use of the *iv_main* program. The *iv_anal* program is used to extract useful parameters that will characterize a process from the data acquired through *iv_main* program. However, only one set of device measurements can be entered and analyzed at one time when using the *iv_anal* program. If large amount of data is needed for statistical study, running the *iv_anal* program can be very slow.

Last year, Robert Ha implemented a more efficient automatic probe system in his Advanced Undergraduate project. When using the automatic prober, once the user has defined the desired devices to be measured in the instrumentation control program, the system will probe the transistor data automatically. This can save a lot of time in probing when compare to the manual probe system. (More descriptions about the Automatic Probe System will be discussed in the Section 1.2.)

This Thesis project is a continued project of Robert Ha's project. An analysis system has been developed to analyze the measurement obtained from the automatic probe system. This analysis system is similar to the *iv_anal* program, but it can handle a large amount of device data efficiently. It is also able to group devices according to their die locations and their drawn lengths or widths for analysis. Parameters such as the effective channel length and width, the peak transconductance, the threshold voltage, the subthreshold slope, the flat-band voltage and the substrate concentration can be extracted from this analysis system.

1.2 Automatic Probing System Description

This section is intended to give a brief description of the automatic probe system. For a more detailed description, please refer to Robert Ha's report on his Advanced Undergraduate project, "Automatic Probing on Standard Transistor Wafers".

The automatic probe system can:

- acquire transistor measurement data from the test system;
- efficiently store the measured data, and
- transfer the data to a PC environment for data analysis.

1.2.1 Hardware Components

In order to use the system effectively, a person is assumed to have knowledge of HP BASIC 6.0 Operating System, HP BASIC Programming Language, 4062B System Manual, and the interconnections among the instruments.

The automatic probe system contains four hardware components:

1. 4062B Semiconductor Parametric Test System (SPTS)
2. R&K 1032 Prober
3. HP 9000 Series 300 Desktop Computer
4. External Hard Disks

(Both the 1032 and the 4062B are connected to the HP computer by HP-IB cables.)

An instrumentation control program called *probtrans* is written to control the complete operations of the probe system. At the beginning of the program, the user has to define the wafer type and the probe card type if the default one is not used.

1.2.2 Data Acquirement

The *probtrans* program is very flexible in letting the user pick devices on the wafer to be measured. In *define-wafer* section, a list of DATA statements tells the program which devices to probe. When the user wants to make a change to the devices to be measured, these DATA statements have to be modified. The user also needs to specify the total number of devices to be measured. Future program expansion can be implemented in the *define-probe card* section when several probe cards are used.

Following these sections is the main menu; here, the user can select the type of measurement. The program allows 4 types of measurements:

1. Fix V_s, V_d, V_b, sweep V_g and measure I_d.
2. Fix V_s, V_b, vary V_d, sweep V_g and measure I_d.

3. Fix V_s, V_d, vary V_b, sweep V_g and measure I_d.

4. Fix V_s, V_b, vary V_g, sweep V_d and measure I_d.

(where V_s=source voltage, V_d=drain voltage, V_b=substrate voltage, V_g=gate voltage, and I_d=drain current.)

After the type of measurement is selected, the user is asked to specify the necessary parameters of measurement such as the total number of steps for the sweep, source voltage, starting gate voltage for the sweep, ending gate voltage for the sweep, and so on. The program will then call a subroutine to position the wafer to the desired device, and the measurement will then be taken.

1.2.3 Data Storage

For each device, the program will create an output file to store the data in HP LIF BDAT format. The first fifteen rows of the file is called the Header, which carries all of the parameters of the measurement; while the rest of the file contains the measured data.

The files are organized in a way such that the user has to input the wafer's name and the total number of devices to be measured. For example, if the wafer's name is *test* and the total number of devices to be measured are *4*, then the program will create the files *test1*, *test2*, *test3* and *test4*. *test1* will store the measurements and information of the first device indicated in the DATA statement, while *test4* will store the measurements and information of the fourth device indicated in the DATA statement.

The files will be stored in a floppy disk such that they can be transferred to a PC for analysis. The reason that the analysis is done on the PC is because the HP is very slow. Besides, PC's have a much larger RAM size and hard disk space as well as a faster processor. (In fact, a RAM size of 20M bytes size is required to do the analysis because of the amount of data it has to handle.) Therefore, large amount of data can be handled easily and sophisticated computations can be performed more time efficiently on PC's.

Since the data will be transferred to a PC environment for analysis through the floppy disk, a manipulation has been done to minimize the memory storage space. A data point is broken into two integers in the output file as shown below:

1.023456E-7 breaks into *1023* and *-7*

In this manipulation, the data is stored as an integer instead of real numbers. While sacrificing a little in the accuracy of the data, the memory space can be reduced by a factor of two.

1.2.4 Procedure to Use the System

The following is the procedure to use the automatic probing system:

1. Set up the prober with the probe card pins on top of die 1, device 1.
2. Place an empty floppy disk in the right hand floppy disk drive.
3. Load the *probtrans* program.
4. Make sure that the desired devices are listed in the DATA statements in the *define-wafer* section.
5. Run the program.
6. Take the floppy disk to a PC.
7. Convert the HP LIF formatted BDAT files into DOS files by using HT Basic's command HPCOPY.

After all these procedures, the output files are ready to be analyzed.

Chapter 2

Description of the Analysis System

2.1 NEWANAL Program

The analysis system that has been developed in this Thesis project is a program called *newanal*. It provides the user with the following options:

- Input new files
- Enter extra files
- Calculation of effective channel length or width
- Peak transconductance extraction
- Threshold voltage extraction using constant current method
- Threshold voltage extraction using mobility degradation method
- Subthreshold slope extraction

Before using the program, the user has to be sure that all the files to be analyzed, together with the *newanal* file, are in the same directory. The listing of the program is included in Appendix A.

2.2 Procedure to Use the Program

Once the program has started, it will ask the user to select either the screen output or printer output. It will also ask the user whether the devices are N-type or P-type before reading the test files. Again, each test file consists of data probed from one device. The main menu will then come up, and the user can exit the program anytime by pressing F8 button in the main menu.

2.2.1 Files Insertion

First of all, the user has to select "NEW FILE" to input new files by pressing the F1 button in the main menu. The program will then ask the user to enter the input wafer's name, and the total number of tests that have been made. The user has the option of analyzing all the tests or just some selected tests. If the user chooses to analyze some selected tests instead of all tests, then the user has to enter the tests in which he/she wants to be analyzed by typing the test numbers of the tests.

The user can add in extra files for analysis if he/she has chosen to analyze some selected tests at the beginning. For example, if the user has selected to analyze 5 tests out of a total of 20 tests at the beginning, he/she can add in extra files to analyze by pressing the F2 button in the main menu.

Every time after F1 or F2 button has been pressed, the program will read in the data stored in the test files. After the files have been read, the measured resistance, threshold voltage using maximum slope method, and the beta of each device will be calculated and displayed. These parameters are needed for calculating effective channel length or width, which in turn is needed to calculate the subthreshold slope, threshold voltage using constant current method and mobility degradation method, and transconductance of each device.

An example output at this stage is shown in Figure 2.1.

```
Die:     1     Length:  1     Wdrawn:   10
Vth =    .6742   .6599   .6459   .6307   .6099   .5652
Rmeas =   1080    599     444     368     322
1/beta= 470      Slope= 125      2nd= 3.4

Die:     1     Length:  2     Wdrawn:   10
Vth =    .8536   .8374   .8175   .7948   .7641   .6046
Rmeas =   3150    1640    1150    912     772
1/beta= 1500     Slope= 127      2nd= 18.7

Die:     1     Length:  5     Wdrawn:   10
Vth =    .9073   .887    .8642   .836    .794    .6025
Rmeas =   9080    4650    3210    2510    2100
1/beta= 4400     Slope= 187      2nd= 60.5
```

Figure 2.1: Sample output after all the data from selected test file have been read

2.2.2 Effective Channel Length or Width

After all the test files have been read, the user should select "L/W EFFECTIVE" to calculate the effective channel length or width by pressing the F3 button in the main menu, before extracting other parameters. This is because the extraction of other parameters, such as the threshold voltage using the constant current method, requires the effective channel dimension to be calculated. The program will then ask the user to select either calculating effective channel length or width.

After the user has selected the appropriate option, the program will group the devices. If the user has selected to calculate effective channel length, the program will then group the devices first by their die locations, and then the drawn width. If the user has selected to calculate effective channel width, the program will then group the devices first by their die locations, and then the drawn length.

If the user selects the effective channel length calculation, the output will display results using the β_0 method and the R-measured method. If the user chooses effective channel width, then only the β_0 method will be used. This is because the R-measured method is effective for calculating the effective channel length only. More details on this will be discussed in chapter 3.

There are two stages of output no matter whether the effective channel length or width calculation is selected. The first stage of output is shown in Figure 2.2. It can be seen that besides the delta dimension of a die, the drawn dimension and external resistance are also shown. The coefficient of determination (r^2) is also included to indicate the correctness of the delta dimension calculation. r^2 ranges from 0 to 1, with 0 meaning the result is not correct at all, and 1 meaning the result is very reliable. Therefore, the higher the value of r^2, the more accurate is the result.

Die#	Dim.(W)	Beta0-dL	Rsd	Beta0-R^2	Rmeas-dL	Rext	Rmeas-R^2
1	10	.5009	110.667	.9999	.4949	101.045	1.0000

Figure 2.2: First stage output when effective dimension calculation option is selected

If the user is not satisfied with the result (*i.e.* having a low r^2 value) of a certain group of data, he/she can select to look at the graph by first entering the die number, and then the drawn dimension of that group of data. According to the example in Figure 2.2, the user will first enter 1 (for die number) and then enter 10 (for drawn dimension) if he/she wishes to look at the graph. After looking at the graph, the user should have an idea of where the bad data point is, and he/she can choose to delete that data point. After deleting the bad data point, a new calculation will be performed with the bad data point not included in the calculation. The user also has the option of restoring all the data points after some data points are deleted.

When the user is satisfied with the result, he/she can type a '0' to quit, and the second stage of output will be shown. Figure 2.3 is an example of the second stage output. As seen in the figure, the drawn dimensions together with the effective dimensions of all the devices are shown here. Besides the drawn and effective dimensions of devices, other die parameters such as the substrate concentration, flat-band voltage, and *phi*-s are also shown. The user also has the option of saving the result in a file of ASCII format.

Die number: 1 Nsub= 1.03E+15 Phis= .576 Vfb=-.0558

Drawn Dim.	Eff. Dim.(Beta0)	Eff. Dim.(Rmeas)
1.00	.4797	.5051
2.00	1.5250	1.5051
5.00	4.4926	4.5051

Figure 2.3: Second stage output when effective dimension calculation option is selected

2.2.3 Transconductance

After the user has finished the effective channel dimension calculation, he/she can select "TRANSCONDUCTANCE" to calculate the peak transconductances of all the devices by pressing the F4 button in the main menu.

The result of output is shown in Figure 2.4, and the user can select to see the graph of peak transconductance versus the gate voltage for each device.

```
Die number:      1     Ldrawn:  1     Width:   10
Vb =    -5     -4     -3     -2     -1     0
GM (mS/mm) =      9.19    9.23    9.34    9.49    9.56    9.53
Vg =     .95     .95     .85     .85     .85     .75

Die number:      1     Ldrawn:  2     Width:   10
Vb =    -5     -4     -3     -2     -1     0
GM (mS/mm) =      3.077   3.116   3.14    3.177   3.203   3.189
Vg =     1.15    1.15    1.15    1.05    1.05    .95

Die number:      1     Ldrawn:  5     Width:   10
Vb =    -5     -4     -3     -2     -1     0
GM (mS/mm) =      1.051   1.059   1.069   1.079   1.1     1.102
Vg =     1.35    1.35    1.25    1.15    1.85    .95
```

Figure 2.4: Sample output after the transconductance option is selected

2.2.4 Threshold voltages

The user can select "Vth: Const. I" to calculate the threshold voltages of every devices using constant current method by pressing the F5 button in the main menu. The program will then ask the user whether he/she wants to use the effective channel length generated from β_0 method or R-measured method in the calculation. Figure 2.5 shows an example output of the result.

```
Die number:      1     Length:  .4797   Width:   10
Vb =    -5     -4     -3     -2     -1     0
Idcalc =  2.085E-6
Vth=     .6375   .6272   .6172   .6067   .5845   .5323

Die number:      1     Length:  1.525   Width:   10
Vb =    -5     -4     -3     -2     -1     0
Idcalc =  6.557E-7
```

Vth=	.8179	.8072	.7865	.756	.7299	.5557

Die number: 1 Length: 4.493 Width: 10

Vb =	-5	-4	-3	-2	-1	0
Idcalc =	2.226E-7					
Vth=	.8563	.8384	.8221	.805	.7515	.5477

Figure 2.5: Sample output after the threshold voltage (constant current method) option is selected

Similarly, the user can select "Vth: M. Degrad." to calculate the threshold voltages of every devices using mobility degradation method by pressing F6 button in the main menu. Since the algorithm requires the transconductance of a device to be calculated, this option can only be selected after the transconductances of devices have been calculated. Figure 2.6 shows an example output of the result.

Die: 1 Ldrawn: 1 Wdrawn: 10

Vb =	-5	-4	-3	-2	-1	0
Vth =	.6571	.6528	.5715	.5702	.568	.4725

Die: 1 Ldrawn: 2 Wdrawn: 10

Vb =	-5	-4	-3	-2	-1	0
Vth =	.8486	.8396	.8273	.7667	.7591	.6097

Die: 1 Ldrawn: 5 Wdrawn: 10

Vb =	-5	-4	-3	-2	-1	0
Vth =	.8822	.856	.864	.8376	.7935	.6023

Figure 2.6: Sample output after the threshold voltage (mobility degradation method) option is selected

2.2.5 Subthreshold slope

The user can select "SUBTHRES. SLOPE" to calculate the subthreshold slope of every devices by pressing F7 button in the main menu. In calculating the subthreshold slope, the user can select the set of threshold voltages calculated from either the maximum slope method, constant slope method, or mobility degradation method to be used. An example output is shown in Figure 2.7.

```
Die:    1      Ldrawn:  1      Wdrawn:   10
Vb =    -5       -4       -3       -2       -1       0
Vth =   .6742   .6599   .6459   .6307   .6099   .5652
Subthreshold Slope (mV/decade) =
        73.69   73.96   74.32   74.84   75.69   83.02

Die:    1      Ldrawn:  2      Wdrawn:   10
Vb =    -5       -4       -3       -2       -1       0
Vth =   .8536   .8374   .8175   .7948   .7641   .6046
Subthreshold Slope (mV/decade) =
        64.22   64.56   64.96   63.83   65.79   86.23

Die:    1      Ldrawn:  5      Wdrawn:   10
Vb =    -5       -4       -3       -2       -1       0
Vth =   .9073   .887    .8642   .836    .794    .6025
Subthreshold Slope (mV/decade) =
        64.27   62.5    62.31   63.25   67.23   86.9
```

Figure 2.7: Sample output after the subthreshold slope option is selected

Finally, after all the parameters have been extracted, the user can exit the program by pressing F8 button in the main menu.

Chapter 3

Effective Channel Length or Width

3.1 Description

Technology nowadays has made submicron channel dimensions for MOS transistors possible. However, with smaller channel dimensions (especially channel length), MOS transistor characteristics become highly sensitive to channel dimension variations. A few tenths-of-a-micron decrease in channel length can result in a significant decrease in threshold voltage and a substantial reduction in source-to-drain punch-through voltage.[1] Therefore, accurate channel dimension determination is essential for device analysis and process control in MOS/VLSI technology.

In the *newanal* program, the effective channel length is calculated by both the β_0 method and the R-measured method; while the effective channel width is calculated only by the β_0 method. Least-squares approximation is used in implementing both of these

1. Chern, Chang, Motta, Godinho. "A New Method To Determine MOSFET Channel Length", *IEEE Electron Device Letters*, Vol. Edl-1, No.9, September, 1980, p.170.

algorithms, and an error analysis using the coefficient of determination has been performed to check if the data obtained from the automatic probe system is good.

3.2 Beta0 Method

This algorithm is an adaptation of the method described in the paper, "*Experimental Derivation of the Source and Drain Resistance*", written by Paul I. Suciu and Ralph L. Johnston.

The paper describes a method of extracting source-and-drain resistance from the measurements of two or more transistors that are identical except for their channel lengths.[2] For small drain-to-source voltage the current is approximately:

$$I_{DS} = \beta (V'_{GS} - V_T) V'_{DS} \tag{3.1}$$

where

$$\beta = \frac{\beta_0}{1 + U_0 (V'_{GS} - V_T) + U_1 (V'_{GS} - V_T)^2} \tag{3.2}$$

$$\beta_0 = \mu_0 C_{OX} \frac{W}{L_{eff}} \tag{3.3}$$

$$V'_{GS} = V_{GS} - I_{DS} R_S \tag{3.4}$$

$$V'_{DS} = V_{DS} - I_{DS} R_{SDT} \tag{3.5}$$

2. Suciu, Johnston. "Experimental Derivation of the Source and Drain Resistance", *Transactions on Electron Devices*, Vol. Ed-27, No. 9, September, 1980, p. 1846.

where U_0 is the mobility degradation coefficient, μ_0 is the low-field channel mobility, C_{OX} is the gate oxide capacitance per unit area, W is the channel width, L_{eff} is the effective channel length, R_S is the source resistance, and R_{SD} is the source-and-drain resistance.[3]

Finding the β_0 for each V_{DS} by solving simultaneous equations with the use of least-squares approximation, the *delta-L* can be found by plotting L_{drawn} vs $1/\beta_0$, where the x-intercept would denote the *delta-L.* Similarly, by finding the β_0 for each V_{DS} by solving simultaneous equations stated above, the *delta-W* can be found by plotting W_{drawn} vs β_0, where the y-intercept would denote the *delta-W.*

When the above relationships (equation 3.2 through equation 3.5) are substituted into equation 3.1, the equation can be written in the form:

$$I_{DS} = \frac{\beta_0 (V_{GS} - V_T)}{1 + A (V_{GS} - V_T)} V_{DS} \tag{3.6}$$

where $A \approx U_0 + \beta_0 R_T$. Then, according to Suciu and Johnston,

$$\frac{(V_{GS} - V_T)}{I_{DS}/V_{DS}} = \frac{1 + A (V_{GS} - V_T)}{\beta_0} \equiv E \tag{3.7}$$

$$\frac{dE}{dV_{GS}} = \frac{A}{\beta_0} = \frac{U_0}{\beta_0} + R_{SD} \tag{3.8}$$

By solving the above equations, R_{SD} can be found.

3.3 R-measured Method

3. Suciu, Johnston, ibid.

The algorithm used here is followed from the method described in the paper, "*A New Method To Determine MOSFET Channel Length*", written by John G.J. Chern, Peter Chang, Richard F. Motta, and Norm Godinho.

The I-V characteristics of an MOS transistor operating in the linear region can be expressed as:

$$I_{DS} = \mu_s C_{OX} \frac{W_{eff}}{L_{eff}} \left(V_{GS} - V_T - \frac{1}{2} V_{DS} \right) V_{DS} \qquad (3.9)$$

and

$$R_{channel} = \frac{L_{eff}}{\mu_s C_{OX} W_{eff} \left(V_{GS} - V_T - \frac{1}{2} V_{DS} \right)} \qquad (3.10)$$

where $W_{eff} = W_{drawn} - \Delta W$ and $L_{eff} = L_{drawn} - \Delta L$ represent effective channel width and length, respectively; ΔW accounts for any process bias such as print bias, etch bias, bird's beak and lateral diffusion of channel-stop implant; ΔL accounts for print bias, etch bias and lateral diffusion of source-drain dopant; and $R_{channel}$ is the intrinsic channel resistance of an MOS transistor.[4]

A person can obtain the measured resistance, R_{meas}, in the following way:

$$R_{meas} = \frac{V_{DS}}{I_{DS}} = R_{external} + R_{channel} = R_{external} + A\,(L_{mask} - \Delta L) \qquad (3.11)$$

where

$$A = \mu_s C_{OX} W_{eff} \left(V_{GS} - V_T - \frac{1}{2} V_{DS} \right)^{-1} \qquad (3.12)$$

4. Chern, Chang, Motta, and Godinho, p. 171.

If a set of MOS transistors with different L_{drawn}'s is prepared, then for fixed V_{GS}, a straight line should be obtained by plotting R_{meas} versus L_{drawn} while keeping A in equation 3.11 constant for all the transistors. Assuming each transistor has the same value of $R_{external}$, when several lines with different V_{GS} (*i.e.* different A) are plotted, they would intersect at one $R_{external}$ and ΔL as shown in the Figure 3.1.

The use of uniform width, W_{eff}, transistors helps to maintain the parameter A in equation 3.12 constant. Together with least-squares approximation, this is how effective channel length obtained from this method. However, using uniform length, L_{eff}, transistors will not help in obtaining effective channel width. This is because A in equation 3.12 will not be constant in this case. This is the reason why this method applies to effective channel length extraction only.

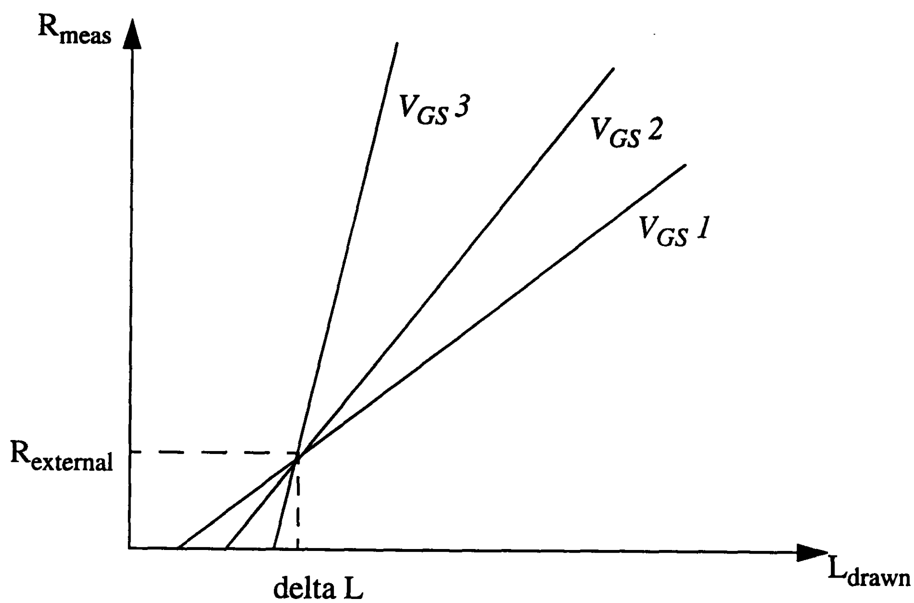

Figure 3.1: Measured resistance, R_{meas}, versus mask level channel length, L_{drawn}.

3.4 Least-Squares Polynomial Approximation

The least-squares polynomial approximation is a basic idea of choosing a function, *p(x)*, to a given function *y(x)* in a way which minimizes the squares of the errors. The polynomial function should be chosen in the following form:

$$p(x) = a_0 + a_1 x + \ldots + a_m x^m \tag{3.13}$$

For a given set of data x_i, y_i and $m < N$, the sum to be minimized is:

$$S = \sum_{i=0}^{N} \left(y_i - a_0 - a_1 x_i - \ldots - a_m x_i^m \right)^2 \tag{3.14}$$

To minimize the sum, standard techniques of calculus then lead to the normal equations, which determine the coefficients a_j. For the case of a linear polynomial, $p(x) = a_0 + a_1 x$, the normal equations are:

$$s_0 a_0 + s_1 a_1 = t_0 \qquad s_1 a_0 + s_2 a_1 = t_1 \tag{3.15}$$

Solving the equations yield:

$$a_0 = \frac{s_2 t_0 - s_1 t_1}{s_0 s_2 - s_1^2} \qquad a_1 = \frac{s_0 t_1 - s_1 t_0}{s_0 s_2 - s_1^2} \tag{3.16}$$

The least squares polynomial approximation for a linear polynomial function will be used in determining the effective channel dimension for both β_0 method and the R-measured method. Details will be explained in Section 3.6.

3.5 Error Analysis

To measure the correlation between variables in a linear equation, the linear correlation coefficient is often used, and is given by the formula:[5]

$$r = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}} \tag{3.17}$$

where $\bar{x}$ is the mean of the x_i's, $\bar{y}$ is the mean of the y_i's.

The value of r lies between -1 and 1. When the data points lie on a perfect straight line with positive slope, r will take on a value of 1. The value of r holds independent of the magnitude of the slope. If the data points lie on a perfect straight line with negative slope, then r has a value of -1. A value of r near zero indicates that the variables x and y are uncorrelated.

Where both x and y are assumed to be built up of simple elements of equal variability, all of which are present in y but some of which are lacking in x, it has been proved that r^2 measures the proportion of all the elements in y which are also present in x. It can be said that r^2, also called the coefficient of determination, measures the percentage to which the variance in y is determined by x, since it measures the proportion of all the elements of variance in y which are also present in x. For example, if $\frac{2}{3}$ of the elements in x correspond to $\frac{2}{3}$ of the elements in y, then the coefficient of determination will be equal to $\frac{4}{9}$.

---

5. Press, Flannery, Teukolsky, Vettering. *Numerical Recipes in C.* Cambridge University Press, New York, 1988, p. 503.

When extracting effective channel dimension using least-squares approximation, the coefficient of determination will be used to measure the accuracy of the data obtained from the automatic probe system.

3.6 Implementation of Algorithm

After the test files are inserted and read by the *newanal* program, the program will first calculate the threshold voltage using maximum slope method (described in section 5.1), the measured resistance by V_{DS}/I_{DS}, and β_0 using equations 3.1 and 3.2. Then, the test files will be grouped by their die locations.

After the user has selected to extract effective channel dimension in the main menu, the program will group the test files within a die. This time, the test files will be grouped according to their widths or lengths depend on which effective channel dimension the user has extracted.

Suppose effective channel length is selected by the user, the program will first use the β_0 method. A least-squares approximation is performed between the lengths of each divided group of devices (with same die location and lengths) and $1/\beta_0$. This is because when equation 3.3 is re-arranged:

$$\frac{1}{\beta_0} = A \times L_{drawn} - A \times deltaL \qquad (3.18)$$

where *A* is the slope generated from the least-squares approximation, and *deltaL* is the constant generated from the approximation divided by the slope. Then, re-arranging equation 3.18:

$$L_{eff} = \frac{\frac{1}{\beta_0} + A \times deltaL}{A} - deltaL \qquad (3.19)$$

After calculating the effective channel length using β_0 method, the program will then use the R-measured method. When least-squares approximation is performed between L_{drawn} and R_{meas} in equation 3.11, the slope will be A and the constant will be $R_{external} - A \times \Delta L$. Then, when another least-squares approximation is performed between the slope and the constant from the first least-squares approximation, *i.e.* between A and $R_{external} - A \times \Delta L$, the slope will be $-\Delta L$ and the constant will be $R_{external}$. Since $L_{eff} = L_{drawn} - \Delta L$, the effective channel length using R-measured method can therefore be calculated by using two least-squares approximations.

When the calculations are complete, the user can select to look at the graphs of L_{drawn} vs $1/\beta_0$ (β_0 method) or L_{drawn} vs R_{meas} (R-measured method). By looking at the graphs, the user can detect whether there are bad data points; and if there is one, the user can select to delete that data point from calculation. This is done by removing the device from the group of devices and recalculating the effective channel length.

If effective channel width is selected instead of effective channel length, then calculation using only β_0 method will be performed. Equations 3.18 and 3.19 will then be modified with $1/\beta_0$ changed to β_0, and L's changed to W's.

After the effective channel dimension has been calculated, the program will calculate the N_{sub} (substrate concentration), ϕ_s, and V_{fb} (flat-band voltage) for each die. The algorithms and calculations are directly implemented from the *iv_anal* program, and the calculations are performed on the largest device within each die.

When the program has finished all the calculations, the user has the option of saving the data to an ASCII file. The first line of the ASCII file will consist of the die number,

N_{sub}, ϕ_s, and V_{fb} results. Then, from second line on, the drawn channel dimension, the effective channel dimension using β_0 method, and the effective channel length using R-measured method (if choosing to calculate effective channel length) will be printed respectively.

3.7 Test and Result

To test the accuracy of the calculations in the *newanal* program, the results from the program are compared to the results from *iv_anal* program. Five devices have been probed for testing. The results show that the calculations from *newanal* program are similar to those of *iv_anal* program:

Parameters extracted	Result from *newanal* program	Result from *iv_anal* program
V_{th} (maximum slope method)	.6742, .6599, .6459, .6307, .6099, .5652	.6741, .6598, .6461, .6306, .6101, .5651
R_{meas}	1080, 599, 444, 368, 322	1080, 599, 445, 368, 322
1/beta	470	470
L_{eff}*(beta0 method; Rmeas method)*	0.4797; 0.5051 1.5250; 1.5051 4.4926; 4.5051	0.48; 0.5067 1.53; 1.5067 4.49; 4.5067
W_{eff}	3.9462 8.7771 48.8766	3.95 8.77 48.9

Table 3.1: Comparison of results (V_{th}, R_{meas}, *1/beta*, L_{eff}, and W_{eff}) between *newanal* program and *iv_anal* program

As seen in the table above, there are differences in some of the results between the two programs. This is due to the fact that *newanal* program inputs data up to 4 significant digits in integer value only, while *iv_anal* program inputs data in floating point value. This is the place where trade-off for reducing memory spaces appears. However, since the percentage difference between the results is less than 1% in any cases, the result from *newanal* program can be concluded to be very reliable.

Chapter 4

Transconductance

4.1 Transconductance Extraction

Transconductance is important since it is a measure of the activity of the transistor, which has a direct effect on its minimum noise, driving capability and bandwidth for a given capacitive load.[6] The transconductance is given by:

$$G_M = \frac{dI_{DS}}{dV_{GS}} \tag{4.1}$$

If the device is operated in strong inversion, its transconductance is also proportional to the ratio of channel width to length, the drain current, and the oxide capacitance per unit area because an increase in any of these terms increases the output current per unit change in gate-to-source voltage:

6. Vittoz, Eric A. "Future of Analog in the VLSI Environment", *BiCMOS Integrated Circuit Design*. IEEE Press, New York, 1994, p. 374.

$$G_M = \sqrt{\left(I_D C_{OX} \frac{W}{L}\right)} \tag{4.2}$$

However, if *W/L* is increased, G_M stops increasing when the device starts operating in weak inversion. [7] This maximum value of transconductance only depends on the drain current:

$$G_{M_{MAX}} \sim I_D \tag{4.3}$$

4.2 Implementation, Test and Result

In the *newanal* program, the transconductance is evaluated at every point with the following formula, which is a variation of equation 4.1:

$$G_M(n) = \frac{I_{DS}(n) - I_{DS}(n-1)}{V_{GS}(n) - V_{GS}(n-1)} \tag{4.4}$$

Then, for every V_d or V_b bias, the peak transconductance is determined by finding the maximum G_m. In the program, since the transconductance is represented in *mS/mm*, the above G_m is actually divided by the width of the device and scaled to get the correct unit.

A comparison between the results obtained from the *newanal* program and the *iv_anal* program is shown in the table below:

7. ibid.

Device (W_{drawn}=10μm)	Transconductance (*mS/mm*) calculated from *newanal* program	Transconductance (*mS/mm*) calculated from *iv_anal* program
L_{drawn}=5μm	1.051, 1.059, 1.069, 1.079, 1.1, 1.102	1.052, 1.06, 1.069, 1.079, 1.098, 1.102
L_{drawn}=2μm	3.077, 3.116, 3.14, 3.177, 3.203, 3.189	3.072, 3.111, 3.14, 3.177, 3.203, 3.184
L_{drawn}=1μm	9.19, 9.23, 9.34, 9.49, 9.56, 9.53	9.185, 9.226, 9.345, 9.485, 9.565, 9.525

Table 4.1: Comparison of transconductance calculations obtained from *newanal* program and *iv_anal* program

The above comparison uses the result of probing three devices of the same width and within the same die location, with six different V_b biases. It shows that the result obtained from *newanal* program is very similar to the result obtained from *iv_anal* program.

The result of probing these same three devices will be used in later chapters to compare other parameters extracted between *newanal* program and *iv_anal* program.

Chapter 5

Threshold Voltage

5.1 Maximum Slope Method

5.1.1 Theory

When deducing the threshold voltage from measurements, one useful approach is to plot the drain current as a function of gate-to-source voltage, as shown in the figure below:

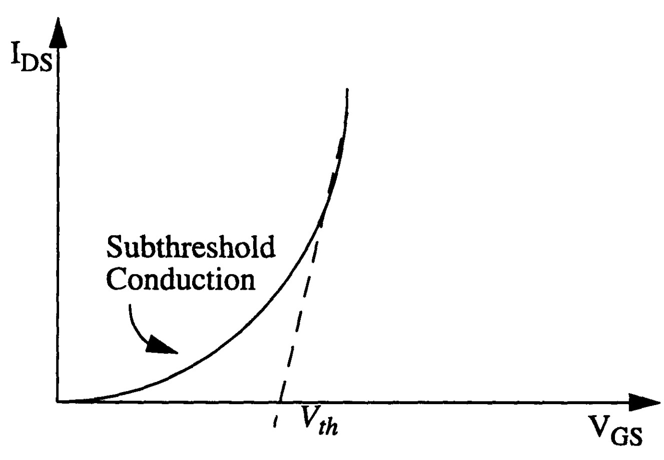

Figure 5.1: Extrapolation of threshold voltage using maximum slope method

The threshold voltage can be determined as the extrapolation of the portion of the curve with maximum slope to zero current. The measured curve deviates from a straight line at low currents because of subthreshold conduction and at high currents because of mobility fall-off in the channel as the carriers approach scattering-limited velocity.[8] (More about subthreshold conduction will be explained in the next chapter.)

5.1.2 Implementation, Test and Result

In the *newanal* program, the slope at each point along the I_{DS} vs V_{GS} curve is evaluated for every V_d or V_b bias. After the slope with the maximum value is found, the threshold voltage is determined by:

$$V_{th} = x(n) - \frac{y(n)}{Slope_{max}} \qquad (5.1)$$

where n is the point of maximum slope.

The results of Vth calculations between the two programs can be seen from table 3.1. The results are similar with a small percentage difference.

5.2 Constant Current Method

The constant current method is another method of extracting threshold voltage. It uses the following equation to calculate a reference current, I_{ref}:

8. Gray, Paul & Meyer, Robert G. *Analysis and Design of Analog Integrated Circuits*. John Wiley & Sons, Inc. New York, 1993, p. 155.

$$I_{ref} = 0.1\mu A \times \frac{W_{eff}}{L_{eff}} \tag{5.2}$$

If W_{eff} or L_{eff} is not available, the program will use W_{drawn} or L_{drawn} instead. For L_{eff}, the user has the option of choosing the result calculated from β_0 method or the R-measured method. Then, for every V_d or V_b bias, the program looks for the point where the measured current is equal to the reference current. It will interpolate that point to get the corresponding V_g at that value. This V_g is the threshold voltage.

The following is the results obtained from both *newanal* program and *iv_anal* program using the same three devices described in section 4.2:

Device (W_{drawn}=10μm)	V_{th} (V) calculated from *newanal* program using constant current method	V_{th} (V) calculated from *iv_anal* program using constant current method
L_{drawn}=5μm	.5477, .7515, .805, .8221, .8384, .8563 (I_{ref} = 0.2226μA)	.5477, .7516, .805, .821, .834, .8563 (I_{ref} = 0.2227μA)
L_{drawn}=2μm	.5557, .7299, .756, .7865, .8072, .8179 (I_{ref} = 0.6557μA)	.5555,.7297, .7559,.7862, .8071, .8178 (I_{ref} = 0.6536μA)
L_{drawn}=1μm	.5323, .5845, .6067, .6172, .6272, .6375 (I_{ref} = 2.085μA)	.5323, .5845, .6067, .6171, .6272, .6375 (I_{ref} = 2.083μA)

Table 5.1: Comparison of threshold voltage obtained from *newanal* program and *iv_anal* program using constant current method

Again, the results obtained from these two programs are very similar with small percentage of difference.

5.3 Mobility Degradation Method

5.3.1 Background

The maximum slope method will usually give good results for Vth, but it can sometimes give bad results, too. For example, a narrow P-channel device might have a very short region where its I_d vs V_{GS} curve is linear. That is, it can go from subthreshold to mobility degradation very quickly. In this case, it is very difficult to pick the linear region and extrapolate the slope accurately.[9] The biggest problem with the maximum slope method, therefore, is its total neglect of mobility degradation. This will cause inaccuracies in β_0 which can result in large inaccuracies in calculating effective channel dimensions. It can also give incorrect values for the source-to-drain resistance.

5.3.2 Implementation, Test and Result

The mobility degradation method uses the equation:

$$I_d = \frac{\beta (V_{GS} - V_{th}) V_{DS}}{1 + \Theta (V_{GS} - V_{th})} \tag{5.3}$$

and a non-linear least squares fitting algorithm is usually used to obtain β_0, V_{th} and Θ. However, in a typical testing environment where many dies per wafer are being tested, this would be too slow.[10] Therefore, a closed form solution would be advantageous to speed up the calculation.

---

9. Bendix, Peter. "Mobility Degradation Method for Obtaining V_{th}", *The Reedholm Report*, April 1994, p. 1.
10. ibid, p. 2.

To obtain a closed form solution, equation 5.3 can be re-arranged in the form:

$$I_d(n) = a \times \frac{V_{GS}(n) - b}{V_{GS}(n) - c} \tag{5.4}$$

where $a = \frac{\beta V_{DS}}{\Theta}$, $b = V_{th} + \frac{V_{DS}}{2}$, $c = V_{th} - \frac{1}{\Theta}$, and $n = 1, 2, 3$. The solution to this problem is demonstrated in the January, 1995 edition of *The Reedholm Report*, and explicit expressions can be written for parameters *a*, *b*, and *c*:[11]

$$a = \frac{I_1 I_2 (V_2 - V_1) - I_1 I_3 (V_3 - V_1) + I_2 I_3 (V_3 - V_2)}{I_1 (V_2 - V_3) + I_2 (V_3 - V_1) - I_3 (V_2 - V_1)} \tag{5.5}$$

$$b = \frac{I_1 I_2 V_3 (V_2 - V_1) - I_1 I_3 V_2 (V_3 - V_1) + I_2 I_3 V_1 (V_3 - V_2)}{I_1 I_2 (V_2 - V_1) - I_1 I_3 (V_3 - V_1) + I_2 I_3 (V_3 - V_2)} \tag{5.6}$$

$$c = \frac{I_1 V_1 (V_2 - V_3) + I_2 V_2 (V_3 - V_1) + I_3 V_3 (V_2 - V_1)}{I_1 (V_2 - V_3) + I_2 (V_3 - V_1) + I_3 (V_2 - V_1)} \tag{5.7}$$

By substituting into the formulas above, V_{th} can be calculated as:

$$V_{th} = b - \frac{V_{DS}}{2} \tag{5.8}$$

Selection of the three measurement points influences the accuracy of the final solution. According to Svoboda, two suitable points are at each side of point of maximum transconductance, and the third point is selected where influence of Θ is greatest, which for a 5V process would be at gate voltage equals 5V.[12]

Since this method of extracting threshold voltage is not implemented in the *iv_anal* program, the results cannot be compared. The following table includes the threshold volt-

11. Svoboda, Vladimer. "Obtaining V_{th}", *The Reedholm Report*, January, 1995, p. 2.
12. ibid.

age extracted from the three different methods using *newanal* program that have been discussed in this chapter:

Device (W_{drawn}=10μm)	V_{th} (V), Maximum slope method	V_{th} (V), Constant current method	V_{th} (V), Mobility degradation method
L_{drawn}=5μm	.6025, .794, .836, .8642, .887, .9073	.5477, .7515, .805, .8221, .8384, .8563	.6023, .7935, .8376, .864, .856, .8822
L_{drawn}=2μm	.6046, .7641, .7948, .8175, .8374, .8536	.5557, .7299, .756, .7865, .8072, .8179	.6097, .7591, .7667, .8273, .8396, .8486
L_{drawn}=1μm	.5652, .6099, .6307, .6459, .6599, .6742	.5323, .5845, .6067, .6172, .6272, .6375	.4725, .568, .5702, .5715, .6528, .6571

Table 5.2: Results of threshold voltage obtained from *newanal* program using three different algorithms

Chapter 6

Subthreshold Slope

6.1 Subthreshold Slope Extraction

The parameters extracted from the *newanal* program so far focused on the normal region of operation where there is a well-defined conducting channel under the gate. Changes in the gate voltage are assumed to cause changes in the channel charge only, and not in the depletion region. However, for gate voltages less than the threshold voltage, the applied gate potential still affects the depletion region charge and the channel charge slightly (which is very small but not zero). The device can therefore conduct small currents for $V_{GS} < V_{th}$.

The operation of devices in this region is called the subthreshold operation, and its major application is for very low power applications at relatively low signal frequencies. In the *newanal* program, an algorithm for extracting the subthreshold slope has been implemented.

6.2 Implementation, Test and Result

The formula used will be:

$$Slope = \frac{1}{\left(\frac{\log I_d 2 - \log I_d 1}{V_{GS} 2 - V_{GS} 1}\right)} \tag{6.1}$$

This is used to extract slopes at every point for the V_{GS} values below threshold. Then for the five points right below threshold, the program looks for a set of two points whose difference is minimum, and the average of the slope values at those two points is taken to get the subthreshold slope. If there are less than five points to begin with, the program proceeds to take the average of all the points available. This algorithm is directly implemented from the *iv_anal* program, but is modified such that the user can choose to use the threshold voltage calculated from either maximum slope method, constant current method or mobility degradation method.

The following is a table of results obtained from *newanal* program and *iv_anal* program using the same three devices described in section 4.2:

Device (W_{drawn}=10μm)	Subthreshold slope calculated from *newanal* program	Subthreshold slope calculated from *iv_anal* program
L_{drawn}=5μm	64.27, 62.5, 62.31, 63.25, 67.23, 86.9	64.27, 62.5, 62.31, 63.25, 67.23, 86.9
L_{drawn}=2μm	64.22, 64.56, 64.96, 63.83, 65.79, 86.23	64.22, 64.56, 64.96, 63.83, 65.79, 86.23

Table 6.1: Comparison of subthreshold slope obtained from *newanal* program and *iv_anal* program

Device (W_{drawn}=10μm)	Subthreshold slope calculated from *newanal* program	Subthreshold slope calculated from *iv_anal* program
L_{drawn}=1μm	73.69, 73.96, 74.32, 74.84, 75.69, 83.02	73.7, 73.96, 74.32, 74.84, 75.69, 83.02

Table 6.1: Comparison of subthreshold slope obtained from *newanal* program and *iv_anal* program

Chapter 7

Conclusion

The *newanal* program has been successfully set up to analyze the measurements obtained from the automatic probe system on standard CMOS transistor wafers. It can extract MOSFET parameters such as the effective channel dimension of devices, the threshold voltage (with the use of different algorithms), the subthreshold slope, the peak transconductance, the substrate doping concentration and the flat-band voltage. The results are similar to that of the *iv_anal* program with only a slight percentage difference (less than 1% in any case). However, the *newanal* program can read a larger amount of data and is able to group devices according to their die locations and drawn lengths or widths when compared to *iv_anal* program.

In the future, more features can be added to the *newanal* program. One example is wafer mapping. Since the *newanal* program is capable of plotting graphs, a person can plot a graph of die number vs effective channel dimension. There is a subsection in the program called *linlin* that is responsible for plotting graphs in the *newanal* program. With a knowledge of HT BASIC, the code in the subsection can be easily understood, and wafer mapping can be easily implemented.

References

Bendix, Peter. "Mobility Degradation Method for Obtaining V_{th}", *The Reedholm Report*, April, 1994.

Chern, Chang, Motta, Godinho. "A New Method To Determine MOSFET Channel Length", *IEEE Electron Device Letters*, Vol. Edl-1, No.9, September, 1980.

Gray, Paul & Meyer, Robert G. *Analysis and Design of Analog Integrated Circuits*. John Wiley &Sons, Inc. New york, 1993.

Ha, Robert. "Automatic Probing on Standard Transistor Wafers", M.I.T. Dept. of E.E.C.S. Advanced Undergraduate project, May, 1994.

Press, Flannery, Teukolsky, Vettering. *Numerical Recipes in C*. Cambridge University Press, New York, 1988.

Suciu, Johnston. "Experimental Derivation of the Source and Drain Resistance", *Transactions on Electron Devices*, Vol. Ed-27, No. 9, September, 1980.

Svoboda, Vladimer. "Obtaining V_{th}", *The Reedholm Report*, January, 1995.

Vittoz, Eric A. "Future of Analog in the VLSI Environment", *BiCMOS Integrated Circuit Design*. IEEE Press, New York, 1994.

Appendix A

Program listing

```
100 OPTION BASE 1
190 CLEAR SCREEN
200 PRINT “PLEASE NOTE THAT THIS ANALYSIS IS NOT VALID FOR MEASUREMENT #4 YET.”
210 PRINT “(That is: Fix Vs,Vb, vary Vg, sweep Vd and measure Id)”
220 PRINT
230 PRINT “SELECT OUTPUT OPTION:”
240 INPUT “‘0’ for screen output and ‘1’ for printer & screen output”,Pt
250 IF Pt<>0 AND Pt<>1 THEN GOTO 190
260 CLEAR SCREEN
270 INPUT “N-type (0) or P-type (1) devices?”,Np
271 IF Np<>0 AND Np<>1 THEN GOTO 260
280 CLEAR SCREEN
310 Start:   !
320   OFF KEY
330   ON KEY 1 LABEL “NEW FILE” GOTO Newfile
340   ON KEY 2 LABEL “EXTRA DEVICE” GOTO Add_device
350   ON KEY 3 LABEL “L/W EFFECTIVE” GOTO Leff
360   ON KEY 4 LABEL “TRANSCONDUCTANCE” GOTO Trcond
370   ON KEY 5 LABEL “Vth: Const. I” GOTO Vthvd
380   ON KEY 6 LABEL “Vth: M. Degrad.” GOTO Vthmd
390   ON KEY 7 LABEL “SUBTHRES. SLOPE” GOTO Subvth
400   ON KEY 8 LABEL “EXIT” GOTO Exit
405 DISP “MAIN MENU”
407   PRINT “=========”
410   PRINT “(SELECT OPTIONS)”
420   GOTO 410
485 REM ***********************************************************
490 Newfile: !
495 REM ***********************************************************
496 DIM File(520)
498 CLEAR SCREEN
500 PRINT “Please enter the input wafer’s name”
510 INPUT Wafername$
520 PRINT “Please enter the number of tests made”
530 INPUT Testnumber
534 CLEAR SCREEN
535 REDIM File(Testnumber)
540 PRINT “Want to analyze all “;Testnumber;” devices?”
541 INPUT “(Type ‘1’ if YES)”,Want
550 IF Want=1 THEN
```

```
560   FOR I=1 TO Testnumber
570     File(I)=I
580   NEXT I
585   Tamount=Testnumber
590   GOTO 890
600 ELSE
610   CLEAR SCREEN
620   PRINT "Among the ";Testnumber;" devices' tests, please enter the amount of tests that you want to analyze"
630   INPUT Tamount
635   FOR I=1 TO Tamount
640     CLEAR SCREEN
650     PRINT "Enter the test number one by one"
655     PRINT "Enter '0' to re-start"
661     IF I<>1 THEN
662       PRINT "(Test numbers entered:"
663       FOR J=1 TO I-1
664         PRINT File(J)
665       NEXT J
667       PRINT ")"
668     END IF
670     INPUT File(I)
675     IF File(I)=0 THEN GOTO Newfile
680   NEXT I
690 END IF
890 CLEAR SCREEN
895 DISP "PLEASE WAIT!"
901 REM ************************************************************
902 REM     This part of the program takes in the information
903 REM     from the Header
904 REM ************************************************************
920 INTEGER Head(15,2)
931 INTEGER Opt(520),Die(520),Variation(520),Num_vb(520),Num_vd(520),Numstep(520),Num_vg(520)
932 DIM Length(520),Wdrawn(520),Dmselse(520)
933 DIM Xcoord(520),Ycoord(520),Thold(520),Tdelay(520),I_comp(520)
934 DIM Vd(520),Vs(520),Vgstart(520),Vgend(520),Dms(520),Dmseffb(520),Dmseffr(520),Dmseffbw(520)
935 DIM Vb1(520,10),Vdstart(520),Vdend(520),Vg(520)
940 REDIM Opt(Testnumber),Die(Testnumber),Vgstart(Testnumber),Vgend(Testnumber)
950 REDIM Vd(Testnumber),Vs(Testnumber),Numstep(Testnumber),Wdrawn(Testnumber),Length(Testnumber)
960 REDIM Variation(Testnumber),Num_vb(Testnumber),Num_vd(Testnumber),Vdstart(Testnumber),Vdend(Testnumber),Vg(Testnumber),Num_vg(Testnumber)
970 REDIM Xcoord(Testnumber),Ycoord(Testnumber),Thold(Testnumber),Tdelay(Testnumber),I_comp(Testnumber)
```

```
980 COM /Leastsq2/S0,S1,S2,T0,T1,Slope,Constant
990 COM /Leastsq3/
Sumx1,Sumx2,Sumx12,Sumx22,Sumx1x2,Sumx1y,Sumx2y,Sumy,Nm,Const,Lin1,Lin2
1000 FOR Dev=1 TO Tamount
1010 ASSIGN @Path TO Wafername$&VAL$(File(Dev))
1020 ENTER @Path;Head(*)
1080 Opt(Dev)=Head(1,1)        !STORES THE TYPE OF MEASUREMENT
1085 IF Opt(Dev)=4 THEN GOTO 190
1090 Die(Dev)=Head(1,2)        !STORES THE DIE NUMBER
1100 Xcoord(Dev)=Head(2,1)     !STORES THE X COORDINATE OF THE DEVICE
1110 Ycoord(Dev)=Head(2,2)     !STORES THE Y COORDINATE OF THE DEVICE
1120 Vgstart(Dev)=Head(3,1)/1000 !STORES THE STARTING Vg
1130 Vgend(Dev)=Head(3,2)/1000  !STORES THE ENDING Vg
1140 Vd(Dev)=Head(4,1)/1000     !STORES THE DRAIN VOLTAGE
1150 Vs(Dev)=Head(4,2)/1000     !STORES THE SOURCE VOLTAGE
1160 Numstep(Dev)=Head(5,2)     !STORES THE NUMBER OF STEPS OF THE MEASUREMENT
1170 Thold(Dev)=Head(6,1)/10    !STORES THE HOLD TIME OF THE MEASUREMENT
1180 Tdelay(Dev)=Head(6,2)/10   !STORES THE DELAY TIME OF THE MEASUREMENT
1190 I_comp(Dev)=Head(7,1)/1000 !STORES THE COMPLIANCE CURRENT
1191 IF Head(7,2)<>0 THEN
1192   Wdrawn(Dev)=Head(7,2)   !STORES THE DEVICE WIDTH
1193 ELSE
1194   CLEAR SCREEN
1195   PRINT "Enter the W-drawn (in um) of ";Wafername$&VAL$(File(Dev))
1196   INPUT Wdrawn(Dev)
1197 END IF
1198 IF Head(8,2)<>0 THEN
1199   Length(Dev)=Head(8,2)   !STORES THE DEVICE LENGTH
1200 ELSE
1201   CLEAR SCREEN
1202   PRINT "Enter the L-drawn (in um) of ";Wafername$&VAL$(File(Dev))
1203   INPUT Length(Dev)
1204   CLEAR SCREEN
1205   DISP "PLEASE WAIT!"
1207 END IF
1208 SELECT Opt(Dev)
1210 CASE 1
1220   Vb1(Dev,1)=Head(5,1)/1000  !STORES THE SUBSTRATE VOLTAGE
1222   Num_vb(Dev)=1
1225   Variation(Dev)=1
1230 CASE 3
1240   Num_vb(Dev)=Head(8,1)   !STORES THE NUMBER OF VARYING Vb
1245   Variation(Dev)=Num_vb(Dev)
1280   M=Num_vb(Dev) MODULO 2
1290   IF M=0 THEN
1300   FOR J=1 TO Num_vb(Dev) STEP 2
```

```
1310     M=M+1
1320     K=M+9
1330     Vb1(Dev,J)=Head(K,1)/1000
1340     Vb1(Dev,J+1)=Head(K,2)/1000
1350   NEXT J
1360   ELSE
1370   N=Num_vb(Dev)-1
1380   Y=0
1390   FOR J=1 TO N STEP 2
1400     Y=Y+1
1410     K=Y+9
1420     Vb1(Dev,J)=Head(K,1)/1000
1430     Vb1(Dev,J+1)=Head(K,2)/1000
1440   NEXT J
1450   Vb1(Dev,Num_vb(Dev))=Head(K+1,1)/1000  !LAST ELEMENT OF THE Vb ARRAY
1460   END IF
1470 CASE 2
1480   Vb1(Dev,1)=Head(5,1)/1000   !STORES THE SUBSTRATE VOLTAGE
1490   Num_vb(Dev)=Head(8,1)   !STORES THE NUMBER OF VARYING Vb
1500   Num_vd(Dev)=Head(9,1)   !STORES THE NUMBER OF VARYING Vd
1505   Variation(Dev)=Num_vd(Dev)
1510   DIM Vd1(520,10)
1515   IF Dev=1 THEN REDIM Vd1(Testnumber,Num_vb(Dev))
1520   M=Num_vd(Dev) MODULO 2
1530   IF M=0 THEN
1540   FOR J=1 TO Num_vd(Dev) STEP 2
1551     M=M+1
1560     K=M+9
1570     Vd1(Dev,J)=Head(K,1)/1000
1580     Vd1(Dev,J+1)=Head(K,2)/1000
1590   NEXT J
1600   ELSE
1610   N=Num_vd(Dev)-1
1620   Y=0
1630   FOR J=1 TO N STEP 2
1640     Y=Y+1
1650     K=Y+9
1660     Vd1(Dev,J)=Head(K,1)/1000
1670     Vd1(Dev,J+1)=Head(K,2)/1000
1680   NEXT J
1690   Vd1(Dev,Num_vd(Dev))=Head(K+1,1)/1000   !LAST ELEMENT OF THE Vd ARRAY
1700   END IF
1710 CASE 4
1720   Vdstart(Dev)=Vgstart(Dev)
1730   Vdend(Dev)=Vgend(Dev)
1740   Vg(Dev)=Vd(Dev)
```

```
1750   Vb1(Dev,1)=Head(5,1)/1000
1760   Num_vb(Dev)=Head(8,1)
1770   Num_vd(Dev)=Head(9,1)
1780   Num_vg(Dev)=Head(9,2)
1785   Variation(Dev)=Num_vg(Dev)
1790   M=Num_vg(Dev) MODULO 2
1800   DIM Vg1(520,10)
1810   IF Dev=1 THEN REDIM Vg1(Testnumber,Num_vg(Dev))
1820   IF M=0 THEN
1830   FOR J=1 TO Num_vg(Dev) STEP 2
1840     M=M+1
1850     K=M+9
1860     Vg1(Dev,J)=Head(K,1)/1000
1870     Vg1(Dev,J+1)=Head(K,2)/1000
1880   NEXT J
1890   ELSE
1900   N=Num_vg(Dev)-1
1910   Y=0
1920   FOR J=1 TO N STEP 2
1930     Y=Y+1
1940     K=Y+9
1950     Vg1(Dev,J)=Head(K,1)/1000
1960     Vg1(Dev,J+1)=Head(K,2)/1000
1970   NEXT J
1980   Vg1(Dev,Num_vg(Dev))=Head(K+1,1)/1000   !LAST ELEMENT OF THE Vg ARRAY
1990   END IF
2000 END SELECT
2010 REM **********************************************************
2020 REM     This part of the program stores the measured Id
2030 REM **********************************************************
2040 INTEGER Array1(1000,2)
2050 IF Dev=1 THEN REDIM Array1(Numstep(Dev)*Variation(Dev),2)
2060 DIM Id(520,10,100)
2070 IF Dev=1 THEN REDIM Id(Testnumber,Variation(Dev),Numstep(Dev))
2080 ENTER @Path;Array1(*)
2090 ASSIGN @Path TO *
2100 FOR J=1 TO Variation(Dev)
2110   Begin=J*Numstep(Dev)-Numstep(Dev)+1
2120   Finish=J*Numstep(Dev)
2130   FOR I=Begin TO Finish
2140     E=Array1(I,1)
2150     K=Array1(I,2)
2160     Id(Dev,J,I-Begin+1)=E*(EXP(2.302585093*(K-3)))
2170   NEXT I
2180 NEXT J
2190 IF Eftag=1 THEN GOTO 6080
```

```
2200 NEXT Dev
2220 ASSIGN @Path TO *
2245 REM *************************************
2250 REM      Vth, Rmeas, Beta calculation
2260 REM *************************************
2280 DIM Vgate(100),Vth(520,10),Rmeas(520,5)
2281 DIM Beta(520),Slp(520),Beta1(520),Rmeas1(520,5)
2290 REDIM Vgate(Numstep(1)),Vth(Testnumber,Variation(1))
2291 REDIM Rmeas(Testnumber,5)
2292 REDIM Beta(Testnumber),Slp(Testnumber),Beta1(Testnumber),Rmeas1(Testnumber,5)
3000 REM *****    Vth calculation    *****
3005 FOR Dev=1 TO Tamount
3010   FOR N1=1 TO Variation(Dev)
3020     Last_y=Id(Dev,N1,1)
3030     Last_x=Vgstart(Dev)
3040     Maxslope=0
3050     Ind=0
3060     Vgate(1)=Vgstart(Dev)
3070     FOR N2=2 TO Numstep(Dev)
3080       Vgate(N2)=Vgstart(Dev)+(N2-1)*(Vgend(Dev)-Vgstart(Dev))/(Numstep(Dev)-1)
3090       Y_con=Id(Dev,N1,N2)
3100       Slope=(Y_con-Last_y)/(Vgate(N2)-Last_x)
3102       IF Np=0 THEN
3104         IF Id(Dev,N1,N2)<1.0E-8 THEN GOTO Ignore
3106       ELSE
3108         IF Id(Dev,N1,N2)>(-1.0E-8) THEN GOTO Ignore
3110       END IF
3120       IF (Slope>Maxslope) AND (Ind=0) THEN
3130         Maxslope=Slope
3140         Vth(Dev,N1)=Last_x-Last_y/Maxslope
3150       ELSE
3160         Ind=1
3170       END IF
3180 Ignore:  Last_x=Vgate(N2)
3190       Last_y=Y_con
3200     NEXT N2
3205   NEXT N1
3210   IF Pt=1 THEN PRINTER IS 10
3211   PRINT "Die: ",Die(Dev),"Length: ",Length(Dev),"Wdrawn: ",Wdrawn(Dev)
3220   PRINT "Vth =",
3230   FOR N1=1 TO Variation(Dev)
3240     PRINT DROUND(Vth(Dev,N1),4),
3249   NEXT N1
3250   PRINT
3450 REM *****    Rmeas calculation    *****
3500   IF Opt(Dev)=3 THEN
```

```
3501    FOR N1=1 TO Num_vb(Dev)
3502      IF Vb1(Dev,N1)=0 THEN Z=N1
3503    NEXT N1
3504  ELSE
3560    Z=1
3606  END IF
3607  PRINT "Rmeas = ",
3610  FOR N1=1 TO 5 !Rmeas uses Vb=0
3612    IF Np=0 THEN
3614            Temp=INT((Vth(Dev,Z)+N1*.5-Vgstart(Dev))/((Vgend(Dev)-Vgstart(Dev))/(Numstep(Dev)-1)))+1
3616    ELSE
3618            Temp=INT((Vth(Dev,Z)-N1*.5-Vgstart(Dev))/((Vgend(Dev)-Vgstart(Dev))/(Numstep(Dev)-1)))+1
3620    END IF
3630    IF Temp<Numstep(Dev)-1 THEN
3632      IF Np=0 THEN
3634            Itmp=Id(Dev,Z,Temp)+(Id(Dev,Z,Temp+1)-Id(Dev,Z,Temp))*(Vth(Dev,Z)+N1*.5-Vgate(Temp))/((Vgend(Dev)-Vgstart(Dev))/(Numstep(Dev)-1))
3636      ELSE
3638            Itmp=Id(Dev,Z,Temp)+(Id(Dev,Z,Temp+1)-Id(Dev,Z,Temp))*(Vth(Dev,Z)-N1*.5-Vgate(Temp))/((Vgend(Dev)-Vgstart(Dev))/(Numstep(Dev)-1))
3640      END IF
3645      IF Opt(Dev)<>2 THEN
3650        Rmeas(Dev,N1)=ABS((Vd(Dev)-Vs(Dev))/Itmp)
3651      ELSE
3652        Rmeas(Dev,N1)=ABS((Vd1(Dev,Z)-Vs(Dev))/Itmp)
3653      END IF
3655      PRINT DROUND(Rmeas(Dev,N1),3),
3660    ELSE
3662      PRINT "0.0";
3668    END IF
3670  NEXT N1
3680  PRINT
3700  REM *****    Beta calculation    *****
3710  FOR N2=1 TO Numstep(Dev)
3712    IF Np=0 THEN
3714      Vgsminvth=Vgate(N2)-Vth(Dev,Z)
3716    ELSE
3718      Vgsminvth=Vth(Dev,Z)-Vgate(N2)
3720    END IF
3730    IF Vgsminvth>.7 THEN
3740      Idx=N2
3750      Maxind=Numstep(Dev)-1
3760      GOTO 3800
3770    END IF
3780    NEXT N2
```

```
3800   FOR N2=Idx TO Maxind
3802     IF Np=0 THEN
3804       Vgsminvth=Vgate(N2)-Vth(Dev,Z)
3806     ELSE
3808       Vgsminvth=Vth(Dev,Z)-Vgate(N2)
3810     END IF
3812     IF Opt(Dev)=2 THEN
3814       Dumb=.05-Vs(Dev)
3816     ELSE
3818       Dumb=Vd(Dev)-Vs(Dev)
3820     END IF
3830     CALL Leastsq3(N2,Idx,Maxind,Vgsminvth,Vgsminvth^2,Vgsminvth*Dumb/Id(Dev,Z,N2))
3840   NEXT N2
3850   Beta(Dev)=Const
3860   Slp(Dev)=Lin1
3870   PRINT “1/beta=”;DROUND(Const,3),”Slope=”;DROUND(Lin1,3),”2nd=”;DROUND(Lin2,3)
3990   PRINT
4000   PRINT
5000   PRINTER IS 1
5005   IF Eftag=1 THEN GOTO 6100
5010 NEXT Dev
5020 PRINT
5030 REM ***** Die Separartion *****
5040 INTEGER Diesep(52,10),Numdie(52)
5041 MAT Diesep=(0)
5042 MAT Numdie=(0)
5050 FOR I=1 TO Tamount
5060   Numdie(Die(I))=Numdie(Die(I))+1
5070   Diesep(Die(I),Numdie(Die(I)))=I
5080 NEXT I
5100 GOTO Start
5990  REM **********************************************
6000 Add_device: !
6005  REM **********************************************
6010 Eftag=1
6020 CLEAR SCREEN
6030 INPUT “Enter the test number of the extra test that you would like to analyze:”,Extra
6040 Tamount=Tamount+1
6050 File(Tamount)=Extra
6060 Dev=Tamount
6062 CLEAR SCREEN
6065 DISP “PLEASE WAIT!”
6070 GOTO 1010
6080 ASSIGN @Path TO *
6090 GOTO 3010
6100 Eftag=0
```

```
6110 Numdie(Die(Tamount))=Numdie(Die(Tamount))+1
6120 Diesep(Die(Tamount),Numdie(Die(Tamount)))=Tamount
6200 GOTO Start
8000 REM ************************************************
8001 Leff: !
8002 REM ************************************************
8003 DIM Nvg1(52,5),Xmaxa(52,5),Xmina(52,5),Ymaxa(52,5),R2(52,5),Rsd(52,5),Bslope(52,5),Bdeltal(52,5),Ymina(52,5)
8005 CLEAR SCREEN
8006 GRAPHICS ON
8008 INPUT "Type '0' for L-effective or type '1' for W-effective calculation",Chs
8015 MAT Nvg1=(5)
8030 FOR I=1 TO Tamount
8040   IF Chs=0 THEN
8050     Dms(I)=Length(I)
8055     Dmselse(I)=Wdrawn(I)
8060   ELSE
8070     Dms(I)=Wdrawn(I)
8075     Dmselse(I)=Length(I)
8080   END IF
8092   IF Chs=0 THEN
8094     Beta1(I)=Beta(I)
8096   ELSE
8098     Beta1(I)=1/Beta(I)
8100   END IF
8110   FOR J=1 TO 5
8120     Rmeas1(I,J)=Rmeas(I,J)
8130   NEXT J
8140 NEXT I
8150 INTEGER Lwsep(52,5,10),Lwsep1(52,5,10),Maxdevice(52)
8151 DIM Vgt(52,5,5),Vgt1(52,5,5),Matchlw(52,5),Matchlw1(52,5)
8155 MAT Matchlw=(0)
8160 MAT Lwsep=(0)
8170 FOR Dien=1 TO 52
8171   IF Numdie(Dien)=0 THEN GOTO 8351
8175   FOR I=1 TO Numdie(Dien)
8180     FOR J=1 TO 5
8190       IF Matchlw(Dien,J)=0 THEN
8200         Matchlw(Dien,J)=Matchlw(Dien,J)+1
8202         Matchlw1(Dien,J)=Matchlw(Dien,J)
8210         Lwsep(Dien,J,Matchlw(Dien,J))=Diesep(Dien,I)
8212         Lwsep1(Dien,J,Matchlw1(Dien,J))=Lwsep(Dien,J,Matchlw1(Dien,J))
8220         GOTO 8350
8230       END IF
8240       IF Dmselse(Lwsep(Dien,J,Matchlw(Dien,J)))=Dmselse(Diesep(Dien,I)) THEN
```

```
8250          Matchlw(Dien,J)=Matchlw(Dien,J)+1
8252          Matchlw1(Dien,J)=Matchlw(Dien,J)
8260          Lwsep(Dien,J,Matchlw(Dien,J))=Diesep(Dien,I)
8262          Lwsep1(Dien,J,Matchlw1(Dien,J))=Lwsep(Dien,J,Matchlw1(Dien,J))
8270          GOTO 8350
8280        END IF
8340      NEXT J
8350    NEXT I
8351  NEXT Dien
8352  FOR Dien=1 TO 52
8353    IF Numdie(Dien)=0 THEN GOTO 8410
8355    FOR K=1 TO 5
8360      FOR J=1 TO Nvg1(Dien,K)
8370        Vgt(Dien,K,J)=.5*J
8380        Vgt1(Dien,K,J)=Vgt(Dien,K,J)
8390      NEXT J
8400    NEXT K
8410  NEXT Dien
9000  REM **********************************************
9010  REM     Leff calculation using beta0 method
9020  REM **********************************************
9021  CLEAR SCREEN
9028  IF Chs=0 THEN
9030    PRINT "Die#","Dim.(W)","Beta0-dL","Rsd","Beta0-R^2","Rmeas-dL","Rext","Rmeas-R^2"
9031    PRINT "----","-------","--------","---","---------","--------","----","---------"
9032  ELSE
9034    PRINT "Die#","Dim.(L)","Beta0-dW","Rsd","Beta-R^2"
9035    PRINT "----","-------","--------","---","--------"
9036  END IF
9038  FOR Dien=1 TO 52
9039    FOR K=1 TO 5
9040      IF Matchlw1(Dien,K)=0 THEN GOTO 10710
9050      IF Matchlw1(Dien,K)=1 THEN GOTO Next_test
9060      Avgx=0
9070      Avgy=0
9080      Xmaxa(Dien,K)=0
9090      Xmina(Dien,K)=0
9100      Ymina(Dien,K)=0
9110      Ymaxa(Dien,K)=0
9138      FOR I=1 TO Matchlw1(Dien,K)
9139        A=Dms(Lwsep1(Dien,K,I))
9140        CALL Leastsq2(I,1,Matchlw1(Dien,K),Dms(Lwsep1(Dien,K,I)),Beta1(Lwsep1(Dien,K,I)))
9141        Avgx=Avgx+(Dms(Lwsep1(Dien,K,I))/Matchlw1(Dien,K))
9142        Avgy=Avgy+(Beta1(Lwsep1(Dien,K,I))/Matchlw1(Dien,K))
9150      NEXT I
9155      Bslope(Dien,K)=Slope
```

```
9160    Bdeltal(Dien,K)=-Constant/Slope
9165    PRINT
9168    IMAGE DD,8X,DD,6X,DD.DDDD,5X,DDD.DDD,2X,D.DDDD,4X,D.DDDD,5X,DDD.DDD,4X,D.DDDD
9169    IMAGE DD,7X,DD,8X,DD.DDDD,2X,DDD.DDD,5X,D.DDDD
9170    FOR I=1 TO Matchlw1(Dien,K)
9171        IF Chs=1 THEN Dmseffbw(Lwsep1(Dien,K,I))=(Beta1(Lwsep1(Dien,K,I))-Constant)/Slope-Bdeltal(Dien,K)
9172      IF Chs=0 THEN Dmseffb(Lwsep1(Dien,K,I))=(Beta1(Lwsep1(Dien,K,I))-Constant)/Slope-Bdeltal(Dien,K)
9173    NEXT I
9180    Den1=0
9190    Num1=0
9200    Num2=0
9210    FOR I=1 TO Matchlw1(Dien,K)
9220      Den1=Den1+(Dms(Lwsep1(Dien,K,I))-Avgx)*(Beta1(Lwsep1(Dien,K,I))-Avgy)
9230      Num1=Num1+(Dms(Lwsep1(Dien,K,I))-Avgx)^2
9240      Num2=Num2+(Beta1(Lwsep1(Dien,K,I))-Avgy)^2
9250    NEXT I
9290    R2(Dien,K)=Den1^2/(Num1*Num2)
9300    FOR I=1 TO Matchlw1(Dien,K)
9305      IF Chs=0 THEN
9310        CALL Leastsq2(I,1,Matchlw1(Dien,K),Beta1(Lwsep1(Dien,K,I)),Slp(Lwsep1(Dien,K,I)))
9312    ELSE
9314    CALL Leastsq2(I,1,Matchlw1(Dien,K),1/Beta1(Lwsep1(Dien,K,I)),Slp(Lwsep1(Dien,K,I)))
9316      END IF
9320    NEXT I
9330    Rsd(Dien,K)=Constant
9335    IF Np=1 THEN Rsd(Dien,K)=-Constant
9340    IF Chs=1 THEN PRINT USING 9169;Dien,Dmselse(Lwsep1(Dien,K,1)),Bdeltal(Dien,K),Rsd(Dien,K),R2(Dien,K)
10000 REM ********************************************
10010 REM    Leff calculation using Rmeas method
10020 REM ********************************************
10030   IF Chs=1 THEN
10050     GOTO Next_test
10060   END IF
10100 Calleff: DIM Consarray(52,5,5),Slopearray(52,5,5),Deltal(52,5),Rext(52,5),R22(52,5)
10180     R22(Dien,K)=0
10320     FOR I=1 TO Nvg1(Dien,K)
10330       Avgx1=0
10340       Avgy1=0
10350       FOR N=1 TO Matchlw1(Dien,K)
10360 CALL Leastsq2(N,1,Matchlw1(Dien,K),Dms(Lwsep1(Dien,K,N)),Rmeas1(Lwsep1(Dien,K,N),I))
10370         Avgx1=Avgx1+(Dms(Lwsep1(Dien,K,N))/Matchlw1(Dien,K))
```

```
10380        Avgy1=Avgy1+Rmeas1(Lwsep1(Dien,K,N),I)/Matchlw1(Dien,K)
10390      NEXT N
10400      Slopearray(Dien,K,I)=Slope
10410      Consarray(Dien,K,I)=Constant
10420      Den1=0
10430      Num1=0
10440      Num2=0
10450      FOR J=1 TO Matchlw1(Dien,K)
10460        Den1=Den1+(Dms(Lwsep1(Dien,K,J))-Avgx1)*(Rmeas1(Lwsep1(Dien,K,J),I)-Avgy1)
10470        Num1=Num1+(Dms(Lwsep1(Dien,K,J))-Avgx1)^2
10480        Num2=Num2+(Rmeas1(Lwsep1(Dien,K,J),I)-Avgy1)^2
10490      NEXT J
10500      R22(Dien,K)=R22(Dien,K)+(Den1^2/(Num1*Num2))/Nvg1(Dien,K)
10520    NEXT I
10530    FOR J=1 TO Nvg1(Dien,K)
10540      CALL Leastsq2(J,1,Nvg1(Dien,K),Slopearray(Dien,K,J),Consarray(Dien,K,J))
10550    NEXT J
10560    Deltal(Dien,K)=-Slope
10570    Rext(Dien,K)=Constant
10580    PRINT USING 9168;Dien,Dmselse(Lwsep1(Dien,K,1)),Bdeltal(Dien,K),Rsd(Dien,K),R2(Dien,K),Deltal(Dien,K),Rext(Dien,K),R22(Dien,K)
10650    !
10700 Next_test: NEXT K
10710  NEXT Dien
10900 INPUT "Choose a die to further analyze or '0' to quit",Dies
10905 IF Dies<>0 THEN INPUT "Choose a dimension to further analyze or '0' to quit",Dims1
10910 IF Dies=0 OR Dims1=0 THEN GOTO 13000
10990 IF Chs=0 THEN
11000   INPUT "TYPE '0' TO QUIT, '1' FOR BETA0 GRAPH, '2' FOR RMEAS GRAPH",Choice
11002   IF Choice<>0 AND Choice<>1 AND Choice<>2 THEN GOTO 11000
11005 ELSE
11006   INPUT "TYPE '0' TO QUIT OR '1' TO SEE THE GRAPH",Choice
11007   IF Choice<>0 AND Choice<>1 THEN GOTO 11006
11009 END IF
11010 IF Choice=0 THEN GOTO 13000
11011 FOR K=1 TO 5
11012   IF Dmselse(Lwsep1(Dies,K,1))=Dims1 THEN
11013     Dims=K
11014     GOTO 11018
11016   END IF
11017 NEXT K
11018 IF Dims=0 THEN GOTO 10905
11020 IF Choice=1 THEN
11030   FOR I=1 TO Matchlw1(Dies,Dims)
11040     IF Beta1(Lwsep1(Dies,Dims,I))>Ymaxa(Dies,Dims) THEN Ymaxa(Dies,Dims)=Beta1(Lwsep1(Dies,Dims,I))
```

```
11042 IF Dms(Lwsep1(Dies,Dims,I))>Xmaxa(Dies,Dims) THEN Xmaxa(Dies,Dims)=Dms(Lwsep1(Dies,Dims,I))
11050 NEXT I
11060 REM ***** Plotting section for the beta0 calculation result *****
11120 Xlab$="L(W) drawn"
11130 Ylab$="Beta"
11132 CLEAR SCREEN
11135 Xmin=Xmina(Dies,Dims)
11136 Xmax=1.05*Xmaxa(Dies,Dims)
11137 Ymin=Ymina(Dies,Dims)
11138 Ymax=1.05*Ymaxa(Dies,Dims)
11145 GOSUB Linlin
11150 LORG 5
11160 FOR I=1 TO Matchlw1(Dies,Dims)
11170   MOVE Dms(Lwsep1(Dies,Dims,I)),Beta1(Lwsep1(Dies,Dims,I))
11180   LABEL "*"
11190 NEXT I
11200 MOVE Bdeltal(Dies,Dims),Ymina(Dies,Dims)
11210 DRAW Xmaxa(Dies,Dims),Bslope(Dies,Dims)*(Xmaxa(Dies,Dims)-Bdeltal(Dies,Dims))
11220 !
11230 OFF KEY
11240 ON KEY 1 LABEL "YES" GOTO 11320
11250 ON KEY 2 LABEL "NO" GOTO 11350
11260 FOR X=3 TO 8
11270   ON KEY X LABEL "" GOTO 11290
11280 NEXT X
11290 DISP "WANT TO CHANGE SCALE? (SELECT OPTION)"
11300 GOTO 11290
11310 !
11320 INPUT "XMIN,XMAX,YMIN,YMAX",Xmina(Dies,Dims),Xmaxa(Dies,Dims),Ymina(Dies,Dims),Ymaxa(Dies,Dims)
11330 GOTO 11120
11340 !
11350 OFF KEY
11360 ON KEY 1 LABEL "YES" GOTO 11440
11370 ON KEY 2 LABEL "NO" GOTO 9021
11380 FOR X=3 TO 8
11390   ON KEY X LABEL "" GOTO 11410
11400 NEXT X
11410 DISP "WANT TO RE-OPTIMIZE? (SELECT OPTION)"
11420 GOTO 11410
11430 !
11440 OUTPUT 2 USING "#,B";255,75
11450 PRINT "No.","L(W) drawn"
11460 FOR I=1 TO Matchlw1(Dies,Dims)
11470   PRINT I,Dms(Lwsep1(Dies,Dims,I))
```

```
11480  NEXT I
11490  INPUT "WHICH ONE TO DELETE? ENTER '0' TO QUIT, '1000' TO RE-STORE ALL DIMENSIONS",Ng
11500  IF Ng=1000 THEN GOTO 8015
11510  IF Ng<>0 THEN
11520    Matchlw1(Dies,Dims)=Matchlw1(Dies,Dims)-1
11530    FOR I=Ng+1 TO Matchlw1(Dies,Dims)+1
11540      Lwsep1(Dies,Dims,I-1)=Lwsep1(Dies,Dims,I)
11560    NEXT I
11570    OUTPUT 2 USING "#,B";255,75
11580    PRINT "No.","L(W) drawn"
11590    FOR I=1 TO Matchlw1(Dies,Dims)
11600      PRINT I,Dms(Lwsep1(Dies,Dims,I))
11610    NEXT I
11670    GOTO 11490
11680  ELSE
11690    GOTO 9021
11740  END IF
11750 ELSE
11758  Xlab$="Ldrawn"
11760  Ylab$="Rmeas"
11770  FOR I=1 TO Matchlw1(Dies,Dims)
11780  IF Dms(Lwsep1(Dies,Dims,I))>Xmaxa(Dies,Dims) THEN Xmaxa(Dies,Dims)=Dms(Lwsep1(Dies,Dims,I))
11790    FOR J=1 TO Nvg1(Dies,Dims)
11800    IF Rmeas1(Lwsep1(Dies,Dims,I),J)>Ymaxa(Dies,Dims) THEN Ymaxa(Dies,Dims)=Rmeas1(Lwsep1(Dies,Dims,I),J)
11810    NEXT J
11820  NEXT I
11825  CLEAR SCREEN
11826  Xmin=Xmina(Dies,Dims)
11827  Xmax=1.05*Xmaxa(Dies,Dims)
11828  Ymin=Ymina(Dies,Dims)
11829  Ymax=1.05*Ymaxa(Dies,Dims)
11835  GOSUB Linlin
11840  LORG 5
11850  FOR I=1 TO Nvg1(Dies,Dims)
11855    PEN I
11860    FOR K=1 TO Matchlw1(Dies,Dims)
11870      MOVE Dms(Lwsep1(Dies,Dims,K)),Rmeas1(Lwsep1(Dies,Dims,K),I)
11880      LABEL "*"
11890    NEXT K
11900    MOVE -Consarray(Dies,Dims,I)/Slopearray(Dies,Dims,I),Ymina(Dies,Dims)
11910    DRAW Xmaxa(Dies,Dims),Consarray(Dies,Dims,I)+Slopearray(Dies,Dims,I)*Xmaxa(Dies,Dims)
11920  NEXT I
11930  !
```

```
11940 OFF KEY
11950 ON KEY 1 LABEL "YES" GOTO 12030
11960 ON KEY 2 LABEL "NO" GOTO 12180
11970 FOR X=3 TO 8
11980   ON KEY X LABEL "" GOTO 12000
11990 NEXT X
12000 DISP "WANT TO RE-SCALE? (SELECT OPTION)"
12010 GOTO 12000
12020 !
12030 INPUT "XMIN,XMAX,YMIN,YMAX",Xmina(Dies,Dims),Xmaxa(Dies,Dims),Ymina(Dies,Dims),Ymaxa(Dies,Dims)
12035 CLEAR SCREEN
12037 Xmin=Xmina(Dies,Dims)
12038 Xmax=1.05*Xmaxa(Dies,Dims)
12039 Ymin=Ymina(Dies,Dims)
12040 Ymax=1.05*Ymaxa(Dies,Dims)
12045 GOSUB Linlin
12050 LORG 5
12060 FOR J=1 TO Nvg1(Dies,Dims)
12070   PEN J
12080   FOR I=1 TO Matchlw1(Dies,Dims)
12090     MOVE Dms(Lwsep1(Dies,Dims,I)),Rmeas1(Lwsep1(Dies,Dims,I),J)
12100     LABEL "*"
12110   NEXT I
12120   Xlow=(Ymina(Dies,Dims)-Consarray(Dies,Dims,J))/Slopearray(Dies,Dims,J)
12130   MOVE Xlow,Ymina(Dies,Dims)
12140   DRAW Xmaxa(Dies,Dims),Consarray(Dies,Dims,J)+Slopearray(Dies,Dims,J)*Xmaxa(Dies,Dims)
12150 NEXT J
12160 GOTO 12000
12170 !
12180 OFF KEY
12190 ON KEY 1 LABEL "YES" GOTO 12270
12200 ON KEY 2 LABEL "NO" GOTO 9021
12210 FOR X=3 TO 8
12220   ON KEY X LABEL "" GOTO 12240
12230 NEXT X
12240 DISP "WANT TO RE-OPTIMIZE? (SELECT OPTION)"
12250 GOTO 12240
12260 !
12270 CLEAR SCREEN
12271 PRINT "No.","VGS-VTH="
12280 FOR J=1 TO Nvg1(Dies,Dims)
12290   PRINT J,Vgt1(Dies,Dims,J)
12300 NEXT J
```

```
12310   INPUT "DELETE WHICH VG? ENTER '0' TO QUIT AND '1000' TO RE-STORE ALL
DIMENSIONS",Ng
12320   IF Ng=1000 THEN GOTO 8015
12330   IF Ng<>0 THEN
12340     Nvg1(Dies,Dims)=Nvg1(Dies,Dims)-1
12350     FOR J=Ng+1 TO Nvg1(Dies,Dims)+1
12360       Vgt1(Dies,Dims,J-1)=Vgt1(Dies,Dims,J)
12370       FOR I=1 TO Matchlw1(Dies,Dims)
12380         Rmeas1(Lwsep1(Dies,Dims,I),J-1)=Rmeas1(Lwsep1(Dies,Dims,I),J)
12390       NEXT I
12400     NEXT J
12410   END IF
12420   OUTPUT 2 USING "#,B";255,75
12430   PRINT "No.","DIMENSION"
12440   FOR I=1 TO Matchlw1(Dies,Dims)
12450     PRINT I,Dms(Lwsep1(Dies,Dims,I))
12460   NEXT I
12470   INPUT "DELETE WHICH L(W)? ENTER '0' TO QUIT",Ng
12480   IF Ng<>0 THEN
12490     Matchlw1(Dies,Dims)=Matchlw1(Dies,Dims)-1
12500     FOR I=Ng+1 TO Matchlw1(Dies,Dims)+1
12510       Lwsep1(Dies,Dims,I-1)=Lwsep1(Dies,Dims,I)
12520       FOR J=1 TO Nvg1(Dies,Dims)
12530         Rmeas1(Lwsep1(Dies,Dims,I-1),J)=Rmeas1(Lwsep1(Dies,Dims,I),J)
12540       NEXT J
12550     NEXT I
12560   END IF
12570   GOTO 9021
12580 END IF
13000 MAT Maxdevice=(1)
13001 FOR D=1 TO 52
13002   IF Numdie(D)=0 THEN GOTO 13172
13003   FOR A=1 TO 5
13005     FOR B=1 TO Matchlw1(D,A)
13010       IF Lwsep1(D,A,B)<>0 THEN
13013         IF Dms(Lwsep1(D,A,B))>Dms(Maxdevice(D)) THEN Maxdevice(D)=Lwsep1(D,A,B)
13014       END IF
13015     NEXT B
13020   NEXT A
13030   REM *****    Nsub, Phis, and Vfb calculations    *****
13032   IF Opt(1)<>2 THEN
13035   DIM Nsub(52),Phis(52),Vfb(52)
13040   Phis(D)=.8
13050   Eox=3.9*8.854E-14
13060   Esi=11.7*8.854E-14
13070   Vtm=.0258
```

```
13075   Ni=1.45E+10
13080   Cox=Eox/(330*1.E-8)
13090   FOR I=1 TO 4
13100     FOR N1=1 TO Num_vb(Maxdevice(D))
13110       Tempo=SQR(ABS(Phis(D))+ABS(Vb1(Maxdevice(D),N1)))
13115       Tump=Num_vb(Maxdevice(D))
13120       IF Tump<>1 THEN CALL Leastsq2(N1,1,Tump,Tempo,Vth(Maxdevice(D),N1))
13130     NEXT N1
13140     IF Tump<>1 THEN Nsub(D)=(Slope*Cox)^2/(2*1.6E-19*Esi)
13150     IF Tump<>1 THEN Phis(D)=2*Vtm*LOG(Nsub(D)/Ni)
13160   NEXT I
13170   IF Tump<>1 THEN Vfb(D)=Constant-Phis(D)
13171   END IF
13172 NEXT D
13173 CLEAR SCREEN
13174 IF Pt=1 THEN PRINTER IS 10
13175 Counter2=0
13176 Counter1=0
13177 FOR D=1 TO 52
13179   IF Numdie(D)=0 THEN GOTO 13290
13180   IF Tump<>1 AND Opt(1)=3 THEN
13181     PRINT "Die number:
";D,"Nsub=";DROUND(Nsub(D),3),"Phis=";DROUND(Phis(D),3),"Vfb=";DROUND(Vfb(D),3)
13182   Counter1=Counter1+1
13183   ELSE
13184     PRINT "Die number: ";D
13185   END IF
13186   PRINT
13187   IMAGE K,5X,K,5X,K
13188   IMAGE 5X,DD.DD,15X,DD.DDDD,15X,DD.DDDD
13189   PRINT USING 13187;"Drawn Dim.","Eff. Dim.(Beta0)","Eff. Dim.(Rmeas)"
13190   PRINT USING 13187;"==========","================","================"
13200   FOR I=1 TO 5
13210     FOR J=1 TO Matchlw1(D,I)
13215       IF Matchlw1(D,I)=1 THEN GOTO 13280
13220       IF Lwsep1(D,I,J)<>0 THEN
13240         IF Chs=0 THEN
13242           Dmseffr(Lwsep1(D,I,J))=Dms(Lwsep1(D,I,J))-Deltal(D,I)
13243                                                   PRINT     USING
13188;Dms(Lwsep1(D,I,J)),Dmseffb(Lwsep1(D,I,J)),Dmseffr(Lwsep1(D,I,J))
13245         ELSE
13250           PRINT USING 13188;Dms(Lwsep1(D,I,J)),Dmseffbw(Lwsep1(D,I,J))
13255         END IF
13256         Counter2=Counter2+1
13260       END IF
13270     NEXT J
```

```
13280  NEXT I
13285  PRINT
13290 NEXT D
13300 PRINTER IS 1
13310 INPUT "Want to store results in an ASCII file? (Type '1' if YES)",Sfile
13320 IF Sfile=1 THEN
13330  INPUT "Enter output file name: ",Sfilename$
13335  CREATE Sfilename$,7*Counter2+Counter1*4
13340  ASSIGN @Path TO Sfilename$;FORMAT ON
13350  FOR D=1 TO 52
13360   IF Numdie(D)=0 THEN GOTO 13560
13370   IF Tump<>1 THEN
13380    OUTPUT @Path;D,DROUND(Nsub(D),4),DROUND(Phis(D),4),DROUND(Vfb(D),4)
13381   ELSE
13382    OUTPUT @Path;D
13383   END IF
13390   FOR I=1 TO 5
13400    FOR J=1 TO Matchlw1(D,I)
13410     IF Matchlw1(D,I)=1 THEN GOTO 13550
13420     IF Chs=0 THEN
13430 OUTPUT @Path;DROUND(Dms(Lwsep1(D,I,J)),4),DROUND(Dmseffb(Lwsep1(D,I,J)),4),DROUND(Dms(Lwsep1(D,I,J))-Dmseffb(Lwsep1(D,I,J)),4),
13435 OUTPUT @Path;DROUND(Rsd(D,I),4),DROUND(Dmseffr(Lwsep1(D,I,J)),4),DROUND(Dms(Lwsep1(D,I,J))-Dmseffr(Lwsep1(D,I,J)),4),DROUND(Rext(D,I),4)
13450     ELSE
13460      OUTPUT @Path;Dms(Lwsep1(D,I,J)),Dmseffbw(Lwsep1(D,I,J)),Dms(Lwsep1(D,I,J))-Dmseffbw(Lwsep1(D,I,J)),Rsd(D,I)
13470     END IF
13490    NEXT J
13550   NEXT I
13560  NEXT D
13570  ASSIGN @Path TO *
13580 END IF
13590 GOTO Start
13600 REM *******************************************************
13610 Trcond: !
13620 REM *******************************************************
13630 CLEAR SCREEN
13640 GCLEAR
13650 GRAPHICS ON
13660 DIM Gm(520,10,100),Gmindex(520,10,100),Peak(10),Tindex(520,10),Trans(100),Shifting(520,10)
13680 REDIM Gm(Testnumber,Variation(1),Numstep(1)),Gmindex(Testnumber,Variation(1),Numstep(1)),Peak(Variation(1)),Tindex(Testnumber,Variation(1))
```

```
13685 REDIM Trans(Numstep(1)),Shifting(Testnumber,Variation(1))
13688 IF Pt=1 THEN PRINTER IS 10
13690 FOR Dev=1 TO Tamount
13695   MAT Peak=(0)
13700   IF Dmseffw(Dev)=0 THEN
13710     Wtemp=Wdrawn(Dev)
13720   ELSE
13730     Wtemp=Dmseffw(Dev)
13740   END IF
13750   FOR N1=1 TO Variation(Dev)
13760     FOR N2=2 TO Numstep(Dev)-1
13770             Gm(Dev,N1,N2-1)=(Id(Dev,N1,N2)-Id(Dev,N1,N2-1))/((Vgend(Dev)-Vgstart(Dev))/
(Numstep(Dev)-1))
13780       Gm(Dev,N1,N2-1)=1000*Gm(Dev,N1,N2-1)/(Wtemp*.001)
13790             Gmindex(Dev,N1,N2-1)=Vgstart(Dev)+(N2-1.5)*(Vgend(Dev)-Vgstart(Dev))/(Num-
step(Dev)-1)
13800     NEXT N2
13810   NEXT N1
13820   FOR N1=1 TO Variation(Dev)
13830     FOR N2=2 TO Numstep(Dev)-1
13840       Trans(N2-1)=Gm(Dev,N1,N2-1)
13850     NEXT N2
13860     Peak(N1)=MAX(Trans(*))
13870     FOR N2=1 TO Numstep(Dev)-2
13880       IF Gm(Dev,N1,N2)=Peak(N1) THEN
13890         Tindex(Dev,N1)=N2
13900         IF Gm(Dev,N1,N2)=Gm(Dev,N1,N2+1) THEN
13910           Shifting(Dev,N1)=.5*(Vgend(Dev)-Vgstart(Dev))/(Numstep(Dev)-1)
13920         ELSE
13930           Shifting(Dev,N1)=0
13940         END IF
13950         GOTO 14000
13960       ELSE
13970         Tindex(Dev,N1)=0
13980       END IF
13990     NEXT N2
14000   NEXT N1
14017   PRINT "Die number: ",Die(Dev),"Ldrawn: ",Length(Dev),"Width: ",Wdrawn(Dev)
14027   IF Opt(Dev)=2 THEN
14037     PRINT "Vd = ",
14047     FOR N1=1 TO Num_vd(Dev)
14057       PRINT Vd1(Dev,N1),
14067     NEXT N1
14077   ELSE
14087     PRINT "Vb = ",
14097     FOR N1=1 TO Num_vb(Dev)
```

```
14107       PRINT Vb1(Dev,N1),
14117     NEXT N1
14127   END IF
14137   PRINT
14147   PRINT “GM (mS/mm) = “,
14157   FOR N1=1 TO Variation(Dev)
14167     IF Tindex(Dev,N1)=0 THEN
14177       PRINT “N/A”,
14187     ELSE
14197       PRINT DROUND(Gm(Dev,N1,Tindex(Dev,N1)),4),
14207     END IF
14217   NEXT N1
14227   PRINT
14237   PRINT “Vg = “,
14247   FOR N1=1 TO Variation(Dev)
14257     PRINT Vgstart(Dev)+(Tindex(Dev,N1)-.5)*(Vgend(Dev)-Vgstart(Dev))/(Numstep(Dev)-1),
14267   NEXT N1
14277   PRINT
14287   PRINT
14357 NEXT Dev
14360 PRINTER IS 1
14370 INPUT “Type ‘1’ to see the graphs or ‘0’ to quit”,Choice
14380 IF Choice<>1 AND Choice<>0 THEN GOTO 14370
14390 IF Choice=0 THEN GOTO 15162
14440   !
14470 FOR Dev=1 TO Tamount
14480   CLEAR SCREEN
14482   PRINT “Die: “,Die(Dev),”Length: “,Length(Dev),”Width: “,Wdrawn(Dev)
14530   PRINT “GM (mS/mm) = “,
14540   FOR N1=1 TO Variation(Dev)
14550     IF Tindex(Dev,N1)=0 THEN
14560       PRINT “N/A”,
14570     ELSE
14580       PRINT DROUND(Gm(Dev,N1,Tindex(Dev,N1)),4),
14600     END IF
14610   NEXT N1
14620   PRINT
14630   PRINT “Vg = “,
14640   FOR N1=1 TO Variation(Dev)
14650     PRINT Vgstart(Dev)+(Tindex(Dev,N1)-.5)*(Vgend(Dev)-Vgstart(Dev))/(Numstep(Dev)-1),
14660   NEXT N1
14670   PRINT
14680   PRINT
14710   Xmin=Vgstart(Dev)
14720    Xmax=1.15*(Vgstart(Dev)+(Numstep(Dev)-2)*(Vgend(Dev)-Vgstart(Dev))/(Numstep(Dev)-1))
```

```
14730  Ymin=0
14740  Ymax=0
14741  FOR N1=1 TO Variation(Dev)
14742    IF Gm(Dev,N1,Tindex(Dev,N1))>Ymax THEN Ymax=Gm(Dev,N1,Tindex(Dev,N1))
14743  NEXT N1
14744  Ymax=1.05*Ymax
14750  Xlab$="Vg    (V)"
14760  Ylab$="Gm    (mS/mm)"
14770  GOSUB Linlin
14780  LORG 5
14790  FOR N1=1 TO Variation(Dev)
14800    PEN N1
14810    MOVE Gmindex(Dev,N1,1),Gm(Dev,N1,1)
14820    FOR N2=3 TO Numstep(Dev)-1
14830      DRAW Gmindex(Dev,N1,N2-1),Gm(Dev,N1,N2-1)
14840    NEXT N2
14850    LORG 1
14860    CLIP OFF
14870    IF Opt(Dev)=2 THEN
14880      LABEL "Vd = "&VAL$(Vd1(Dev,N1))&""
14890    ELSE
14900      LABEL "Vb = "&VAL$(Vb1(Dev,N1))&""
14910    END IF
14920    CLIP ON
14930    LORG 5
14940    IF Tindex(Dev,N1)=0 THEN GOTO 14970
14950    MOVE Gmindex(Dev,N1,Tindex(Dev,N1))+Shifting(Dev,N1),Gm(Dev,N1,Tindex(Dev,N1))
14960    LABEL "*"
14970  NEXT N1
14980  OFF KEY
14990  ON KEY 1 LABEL "YES" GOTO 15070
15000  ON KEY 2 LABEL "NO" GOTO 15095
15010  FOR X=3 TO 8
15020    ON KEY X LABEL "" GOTO 15040
15030  NEXT X
15040  DISP "WANT TO CHANGE SCALE?  (SELECT OPTION)"
15050  GOTO 15040
15060  !
15070  INPUT "XMIN, XMAX, YMIN, YMAX",Xmin,Xmax,Ymin,Ymax
15080  GOTO 14750
15090  !
15095  IF Dev<>Tamount THEN
15100    ON KEY 1 LABEL "YES" GOTO 15161
15110    ON KEY 2 LABEL "NO" GOTO 15162
15120    FOR X=3 TO 8
15130      ON KEY X LABEL "" GOTO 15150
```

```
15140     NEXT X
15150     DISP "WANT TO ANALYZE ANOTHER FILE? (SELECT OPTION)"
15155     GOTO 15150
15160   END IF
15161 NEXT Dev
15162 GCLEAR
15170 CLEAR SCREEN
15220 GOTO Start
15300 REM ******************************************************
15301 Subvth: !
15302 REM ******************************************************
15310 DIM Vtg(520,10),Sslope(10),Nval(10),Subindex2(10),Subindex(10,101)
15315 DIM Subreal(10,101),Subdiff(10,101)
15320 REDIM Vtg(Testnumber,Variation(1)),Sslope(Variation(1))
15330 REDIM Nval(Variation(1)),Subindex2(Variation(1)),Subindex(Variation(1),Numstep(1))
15340 CLEAR SCREEN
15350 GCLEAR
15360 GRAPHICS ON
15361 PRINT "Want the set of Vth from the result of:"
15370 PRINT "1. Maximum Slope Method"
15371 IF Tagvthvd=1 THEN PRINT "2. Constant Slope Method"
15372 IF Tagvthmd=1 THEN PRINT "3. Mobility Degradation Method"
15375 INPUT "Your choice : ",Choice
15376 IF Choice<>1 AND Choice<>2 AND Choice<>3 THEN GOTO 15340
15378 CLEAR SCREEN
15379 IF Pt=1 THEN PRINTER IS 10
15380 FOR Dev=1 TO Tamount
15390   FOR N1=1 TO Variation(Dev)
15400     IF Choice=1 THEN
15410       Vtg(Dev,N1)=Vth(Dev,N1)
15420     ELSE
15430       IF Choice=2 THEN
15435         IF Tagvthvd=0 THEN GOTO 15340
15440         Vtg(Dev,N1)=Vtd(Dev,N1)
15450       ELSE
15455         IF Tagvthmd=0 THEN GOTO 15340
15460         Vtg(Dev,N1)=Vthmd(Dev,N1)
15470       END IF
15480     END IF
15490     Vgstep=(Vgend(Dev)-Vgstart(Dev))/(Numstep(Dev)-1)
15495     IF Np=0 THEN
15500       IF Vtg(Dev,N1)<=Vgstart(Dev) THEN
15510         Nval(N1)=0
15515       ELSE
15520         Nval(N1)=INT((Vtg(Dev,N1)-Vgstart(Dev))/Vgstep)+2
15530       END IF
```

```
15540     IF Id(Dev,N1,1)<=0 THEN Id(Dev,N1,1)=1.E-12
15542   ELSE
15544     IF Vtg(Dev,N1)>=Vgstart(Dev) THEN
15546       Nval(N1)=0
15548     ELSE
15550       Nval(N1)=INT((Vtg(Dev,N1)-Vgstart(Dev))/Vgstep)+2
15552     END IF
15554     IF Id(Dev,N1,1)>=0 THEN Id(Dev,N1,1)=-1.E-12
15556   END IF
15560 NEXT N1
15570 MAT Subdiff=(0)
15580 FOR N1=1 TO Variation(Dev)
15590   IF Nval(N1)=0 THEN GOTO 15690
15600   FOR N2=2 TO Nval(N1)
15602     IF Np=0 THEN
15604       IF Id(Dev,N1,N2)<=0 THEN Id(Dev,N1,N2)=1.E-12
15606     ELSE
15608       IF Id(Dev,N1,N2)>=0 THEN Id(Dev,N1,N2)=-1.E-12
15610     END IF
15620     IF Id(Dev,N1,N2)=Id(Dev,N1,N2-1) THEN
15630       Subreal(N1,N2-1)=0
15640     ELSE
15650       Subreal(N1,N2-1)=Vgstep/(LGT(ABS(Id(Dev,N1,N2)))-LGT(ABS(Id(Dev,N1,N2-1))))
15660     END IF
15670     Subindex(N1,N2-1)=Vgstart(Dev)+(N2-1.5)*Vgstep
15680   NEXT N2
15690 NEXT N1
15700 FOR N1=1 TO Variation(Dev)
15710   IF Nval(N1)=0 THEN GOTO 15800
15720   IF Nval(N1)=2 THEN
15730     Nval(N1)=3
15735     PRINTER IS 1
15740     PRINT "NVAL("&VAL$(Dev)&","&VAL$(N1)&") CHANGED"
15745     IF Pt=1 THEN PRINTER IS 10
15750     GOTO 15800
15760   END IF
15770   FOR N2=1 TO Nval(N1)-1
15780     IF Subreal(N1,N2)=0 THEN Subindex2(N1)=N2
15790   NEXT N2
15800 NEXT N1
15820 PRINT "Die: ",Die(Dev),"Ldrawn: ",Length(Dev),"Wdrawn: ",Wdrawn(Dev)
15830 IF Opt(Dev)=2 THEN
15840   PRINT "Vd = ",
15850   FOR N1=1 TO Variation(Dev)
15860     PRINT Vd1(Dev,N1),
15870   NEXT N1
```

```
15880   PRINT
15890 ELSE
15900   PRINT "Vb = ",
15910   FOR N1=1 TO Variation(Dev)
15920     PRINT Vb1(Dev,N1),
15930   NEXT N1
15940   PRINT
15950 END IF
15960 PRINT "Vth = ",
15970 FOR N1=1 TO Variation(Dev)
15980   PRINT DROUND(Vtg(Dev,N1),4),
15990 NEXT N1
16000 PRINT
16010 IF Variation(Dev)>4 THEN
16020   PRINT "Subthreshold Slope (mV/decade) = "
16030   PRINT " ",
16040 ELSE
16050   PRINT "Subthreshold Slope (mV/decade) = ",
16060 END IF
16070 FOR N1=1 TO Variation(Dev)
16080   IF Nval(N1)=0 THEN
16081     PRINT "O.O.R.",
16082   ELSE
16085     IF Nval(N1)>=5 THEN
16090       Tempsum=0
16100       Tempind=0
16110       IF Nval(N1)-3>Subindex2(N1) THEN
16120         Tempindex=Nval(N1)-3
16130       ELSE
16140         Tempindex=Subindex2(N1)+1
16150       END IF
16160       Slp1=ABS(Subreal(N1,Nval(N1)-1)-Subreal(N1,Nval(N1)-2))
16170       Tempind=Nval(N1)-1
16180       FOR N2=Tempindex TO Nval(N1)-1
16190         Subdiff(N1,N2)=ABS(Subreal(N1,N2)-Subreal(N1,N2-1))
16200         IF Subdiff(N1,N2)<Slp1 THEN
16210           Tempind=N2
16220           Slp1=Subdiff(N1,N2)
16230         END IF
16240       NEXT N2
16250       Tempsum=Subreal(N1,Tempind)+Subreal(N1,Tempind-1)
16260       IF Tempsum>.6 THEN
16270         Sslope(N1)=1000*(Subreal(N1,Nval(N1)-1)+Subreal(N1,Nval(N1)-2))/2
16280         PRINT DROUND(Sslope(N1),4),
16290       ELSE
16300         Sslope(N1)=1000*Tempsum/2
```

```
16310           PRINT DROUND(Sslope(N1),4),
16320         END IF
16330       ELSE
16340         Tempsum2=0
16350         Tempind2=0
16360         FOR N2=Subindex2(N1)+1 TO Nval(N1)-1
16370           IF Subreal(N1,N2)>0 THEN
16380             Tempind2=Tempind2+1
16390             Tempsum2=Subreal(N1,N2)+Tempsum2
16400           END IF
16410         NEXT N2
16420         IF Tempind2=0 THEN
16430           Sslope(N1)=0
16440           PRINT “0.0”,
16450         ELSE
16460           Sslope(N1)=1000*Tempsum2/Tempind2
16470           PRINT DROUND(Sslope(N1),4),
16480         END IF
16490       END IF
16495     END IF
16500   NEXT N1
16505 PRINT
16507 PRINT
16510 NEXT Dev
16520 PRINTER IS 1
17780 GOTO Start
17790 REM ******************************************************
17800 Vthvd: !
17810 REM ******************************************************
17900 DIM Vtd(520,10)
17910 REDIM Vtd(Testnumber,10)
17920 CLEAR SCREEN
17930 INPUT “Use Leff from Beta0(0) result or Rmeas(1) result?  “,Borr
17940 IF Borr<>0 AND Borr<>1 THEN GOTO 17930
17960 FOR Dev=1 TO Tamount
17970   Wtemp=Dmseffbw(Dev)
17980   IF Wtemp=0 THEN Wtemp=Wdrawn(Dev)
17990   IF Borr=0 THEN
18000     Ltemp=Dmseffb(Dev)
18010   ELSE
18020     Ltemp=Dmseffr(Dev)
18025   END IF
18030   IF Ltemp=0 THEN Ltemp=Length(Dev)
18032   IF Np=0 THEN
18034     Temp=1.0E-7*Wtemp/Ltemp
18036   ELSE
```

```
18037     Temp=-1.E-7*Wtemp/Ltemp
18040   END IF
18050   FOR N1=1 TO Variation(Dev)
18060     Vtd(Dev,N1)=Vgstart(Dev)
18070     IF Opt(Dev)=4 THEN
18071       Numvg=Num_vg(Dev)
18072     ELSE
18073       Numvg=Numstep(Dev)-1
18074     END IF
18080     FOR N2=1 TO Numvg
18090       IF (Id(Dev,N1,N2)<=Temp AND Id(Dev,N1,N2+1)>=Temp) OR (Id(Dev,N1,N2)>=Temp AND Id(Dev,N1,N2+1)<=Temp) THEN
18100         Vgate(N2)=Vgstart(Dev)+(N2-1)*(Vgend(Dev)-Vgstart(Dev))/Numvg
18110         Delta_y=Id(Dev,N1,N2+1)-Id(Dev,N1,N2)
18120         Slope=Delta_y/((Vgend(Dev)-Vgstart(Dev))/Numvg)
18130         Mult=(Temp-Id(Dev,N1,N2))/Delta_y
18140         Vtd(Dev,N1)=Vgate(N2)+Mult*(Vgend(Dev)-Vgstart(Dev))/Numvg
18150         GOTO 18180
18160       END IF
18170     NEXT N2
18180   NEXT N1
18190   IF Pt=1 THEN PRINTER IS 10
18200   PRINT
18210   PRINT “Die number: “,Die(Dev),”Length: “,DROUND(Ltemp,4),”Width: “,DROUND(Wtemp,4)
18220   IF Opt(Dev)=2 THEN
18221     PRINT “Vd = “,
18222   ELSE
18224     PRINT “Vb = “,
18228   END IF
18230   FOR N1=1 TO Variation(Dev)
18240     IF Opt(Dev)=2 THEN
18245       PRINT Vd1(Dev,N1),
18250     ELSE
18260       PRINT Vb1(Dev,N1),
18265     END IF
18270   NEXT N1
18280   PRINT
18290   PRINT “Idcalc = “,DROUND(Temp,4)
18300   PRINT “Vth= “,
18310   FOR N1=1 TO Variation(Dev)
18320     PRINT DROUND(Vtd(Dev,N1),4),
18330   NEXT N1
18340   PRINT
18350   PRINTER IS 1
18400 NEXT Dev
```

```
18405 Tagvthvd=1
18410 GOTO Start
18500 REM ****************************************
18510 Vthmd: !
18511 REM ****************************************
18520 DIM Vthmd(520,10)
18560 REDIM Vthmd(Testnumber,Variation(1))
18570 CLEAR SCREEN
18590 IF Pt=1 THEN PRINTER IS 10
18610 FOR Dev=1 TO Tamount
18630   PRINT "Die: ",Die(Dev),"Ldrawn: ",Length(Dev),"Wdrawn: ",Wdrawn(Dev)
18640   IF Opt(Dev)=2 THEN
18642     PRINT "Vd = ",
18643   ELSE
18645     PRINT "Vb = ",
18647   END IF
18650   FOR N1=1 TO Variation(Dev)
18652     IF Opt(Dev)=2 THEN
18653       PRINT Vd1(Dev,N1),
18654     ELSE
18655       PRINT Vb1(Dev,N1),
18656     END IF
18657   NEXT N1
18659   PRINT
18662   PRINT "Vth =",
18690   FOR N1=1 TO Variation(Dev)
18700     Vgstep=(Vgend(Dev)-Vgstart(Dev))/(Numstep(Dev)-1)
18710     V1=Vgstart(Dev)+Tindex(Dev,N1)*Vgstep-3*Vgstep
18720     V2=Vgstart(Dev)+Tindex(Dev,N1)*Vgstep+2*Vgstep
18730     V3=Vgend(Dev)
18740     I1=Id(Dev,N1,Tindex(Dev,N1)-3)
18750     I2=Id(Dev,N1,Tindex(Dev,N1))
18760     I3=Id(Dev,N1,Numstep(Dev))
18770                 B=(I1*I2*V3*(V2-V1)-I1*I3*V2*(V3-V1)+I2*I3*V1*(V3-V2))/(I1*I2*(V2-
V1)+I1*I3*(V1-V3)+I2*I3*(V3-V2))
18780     IF Opt(Dev)=2 THEN
18790       Vthmd(Dev,N1)=B-(Vd1(Dev,N1)-Vs(Dev))/2
18800     ELSE
18810       Vthmd(Dev,N1)=B-(Vd(Dev)-Vs(Dev))/2
18820     END IF
18821     IF Np=0 THEN
18822       IF Vthmd(Dev,N1)<0 THEN Vthmd(Dev,N1)=0
18823     ELSE
18824       IF Vthmd(Dev,N1)>0 THEN Vthmd(Dev,N1)=0
18825     END IF
18830     PRINT DROUND(Vthmd(Dev,N1),4),
```

```
18840    NEXT N1
18850    PRINT
18860    PRINT
18870 NEXT Dev
18880 PRINTER IS 1
18890 Tagvthmd=1
18900 GOTO Start
19000 Linlin: ! Linear-linear graph plane
19010 GCLEAR
19020 GINIT
19030 LINE TYPE 1
19040 PEN 1
19050 DEG
19060 CSIZE 3.5,.6
19070 MOVE 75,15
19080 LORG 5
19090 LDIR 0
19100 LABEL Xlab$
19110 MOVE 6,60
19120 LDIR 90
19130 LABEL Ylab$
19140 LORG 5
19150 VIEWPORT 25,125,23,97
19160 WINDOW Xmin,Xmax,Ymin,Ymax
19170 PEN 2
19180 Xs=(Xmax-Xmin)/5
19190 Ys=(Ymax-Ymin)/5
19200 IF (Prt_plt=1) THEN 19230
19210 LINE TYPE 4
19220 GRID Xs,Ys,Xmin,Ymin,1,1
19230 LINE TYPE 1
19240 PEN 1
19250 AXES Xs,Ys,Xmin,Ymin,1,1
19260 AXES Xs,Ys,Xmax,Ymax,1,1
19270 FRAME
19280 CLIP OFF
19290 GOSUB Lxaxes_1
19300 GOSUB Lyaxes_1
19310 CLIP ON
19320 RETURN
19340   REM *****  PLOTTING SECTION -- LOG-LINEAR GRAPH PLANE  *****
19350 Loglin:! LOG-LINEAR GRAPH PLANE
19360 GCLEAR
19370 LINE TYPE 1
19380 PEN 1
19390 DEG
```

```
19400 CSIZE 3.5,.6
19410 MOVE 75,15
19420 LORG 5
19430 LDIR 0
19440 LABEL Xlab$
19450 MOVE 6,60
19460 LDIR 90
19470 LABEL Ylab$
19480 LORG 5
19490 VIEWPORT 25,125,23,97
19500 WINDOW Xmin,Xmax,LGT(Ymin),LGT(Ymax)
19510 PEN 2
19520 Xs=(Xmax-Xmin)/5
19530 Ys=1
19540 IF (Prt_plt=1) THEN 19570
19550 LINE TYPE 4
19560 GRID 1,1,Xmin,LGT(Ymin),1,1
19570 LINE TYPE 1
19580 PEN 1
19590 AXES 1,1,Xmin,LGT(Ymin),1,1
19600 AXES 1,1,Xmax,LGT(Ymax),1,1
19610 FRAME
19620 CLIP OFF
19630 GOSUB Lxaxes_2
19640 GOSUB Lyaxes_2
19650 CLIP ON
19660 RETURN
19670   REM *****  PLOTTING SECTION -- LINEAR-LINEAR X-AXIS SETUP  *****
19680 Lxaxes_1:!  LINEAR-LINEAR X-AXIS
19690 FOR Xp=Xmin TO Xmax STEP Xs
19700   MOVE Xp,Ymin-Ymax/50
19710   CSIZE 3,.6
19720   LORG 6
19730   LDIR 0
19740   LABEL USING "K";Xp
19750 NEXT Xp
19760 RETURN
19761   REM *****  PLOTTING SECTION -- LINEAR-LINEAR Y-AXIS SETUP  *****
19770 Lyaxes_1:!  LINEAR-LINEAR Y-AXIS
19780 FOR Yp=Ymin TO Ymax STEP Ys
19790   MOVE Xmin-Xmax/50,Yp
19800   CSIZE 3,.6
19810   LORG 8
19820   LDIR 0
19830   LABEL USING "MD.DE  ";Yp
19840 NEXT Yp
```

```
19850 RETURN
19851   REM ***** PLOTTING SECTION -- LOG-LINEAR X-AXIS SETUP *****
19860 Lxaxes_2:! LOG-LINEAR X-AXIS
19870 FOR Xp=Xmin TO Xmax STEP 1
19880   MOVE Xp,LGT(Ymin)-LGT(Ymax/Ymin)/20
19890   CSIZE 3,.6
19900   LORG 5
19910   LDIR 0
19920   LABEL USING "K";Xp
19930 NEXT Xp
19940 RETURN
19941   REM ***** PLOTTING SECTION -- LOG-LINEAR Y-AXIS SETUP *****
19950 Lyaxes_2:! LOG-LINEAR Y-AXIS
19960 FOR Yp=LGT(Ymin) TO LGT(Ymax) STEP 1
19970   MOVE Xmin-(Xmax-Xmin)/50,Yp
19980   CSIZE 3,.6
19990   LORG 8
20000   LDIR 0
20010 LABEL USING "DE,MDD";10^Yp
20020 NEXT Yp
20030 RETURN
20040 !
21000 Exit: !
21010 END
22100 SUB Leastsq2(Presind,Initind,Maxind,Xval,Yval)
22110 COM /Leastsq2/S0,S1,S2,T0,T1,Slope,Constant
22120 IF Presind<Initind OR Presind>Maxind THEN
22130   PRINT "error in index of leastsq2"
22140 ELSE
22150   IF Presind=Initind THEN
22160     S0=1
22170     S1=Xval
22180     S2=Xval*Xval
22190     T0=Yval
22200     T1=Xval*Yval
22210   ELSE
22220     S0=S0+1
22230     S1=S1+Xval
22240     S2=S2+Xval*Xval
22250     T0=T0+Yval
22260     T1=T1+Xval*Yval
22270   END IF
22280   IF Presind=Maxind THEN
22290     Delta=S0*S2-S1*S1
22300     Temp1=S2*T0-S1*T1
22310     Temp2=S0*T1-S1*T0
```

```
22320     Constant=Temp1/Delta
22330     Slope=Temp2/Delta
22340   END IF
22350 END IF
22360 SUBEND
22400 SUB Leastsq3(Presind,Initind,Maxind,X1val,X2val,Yval)
22410 COM /Leastsq3/ Sumx1,Sumx2,Sumx12,Sumx22,Sumx1x2,Sumx1y,Sumx2y,Sumy,Nm,Const,Lin1,Lin2
22422 IF Presind<Initind OR Presind>Maxind THEN
22430   PRINT "ERROR IN INDEX OF LEASTSQ3"
22440 ELSE
22450   IF Presind=Initind THEN
22460     Sumx1=X1val
22470     Sumx12=X1val^2
22480     Sumx2=X2val
22490     Sumx22=X2val^2
22500     Sumx1x2=X1val*X2val
22510     Sumx1y=X1val*Yval
22520     Sumx2y=X2val*Yval
22530     Sumy=Yval
22540     Nm=1
22550   ELSE
22560     Sumx1=Sumx1+X1val
22570     Sumx12=Sumx12+X1val^2
22580     Sumx2=Sumx2+X2val
22590     Sumx22=Sumx22+X2val^2
22600     Sumx1x2=Sumx1x2+X1val*X2val
22610     Sumx1y=Sumx1y+X1val*Yval
22620     Sumx2y=Sumx2y+X2val*Yval
22630     Sumy=Sumy+Yval
22640     Nm=Nm+1
22700   END IF
22710   IF Presind=Maxind THEN
22720     Delta=Nm*Sumx12*Sumx22+2*Sumx1*Sumx1x2*Sumx2
22730     Delta=Delta-Sumx12*Sumx2^2-Sumx22*Sumx1^2-Nm*Sumx1x2^2
22740     Temp1=Sumy*Sumx12*Sumx22+Sumx1y*Sumx1x2*Sumx2+Sumx2y*Sumx1x2*Sumx1
22750     Temp1=Temp1-Sumx2y*Sumx12*Sumx2-Sumx1*Sumx1y*Sumx22-Sumy*Sumx1x2^2
22760     Temp2=Nm*Sumx1y*Sumx22+Sumx1*Sumx2y*Sumx2+Sumx2*Sumx1x2*Sumy
22770     Temp2=Temp2-Sumx2*Sumx1y*Sumx2-Sumy*Sumx1*Sumx22-Nm*Sumx2y*Sumx1x2
22780     Temp3=Nm*Sumx12*Sumx2y+Sumx1*Sumx1x2*Sumy+Sumx1*Sumx2*Sumx1y
22790     Temp3=Temp3-Sumy*Sumx12*Sumx2-Sumx1*Sumx1*Sumx2y-Nm*Sumx1x2*Sumx1y
22800     Const=Temp1/Delta
22810     Lin1=Temp2/Delta
22820     Lin2=Temp3/Delta
22830   END IF
22840 END IF
```

22850 SUBEND