A MOS PRAMETER EXTRACTION PROGRAM

FOR THE BSIM MODEL

by

Joseph R. Pierret

Memorandum No. UCB/ERL M84/99

21 November 1984

(cover)

A MOS PARAMETER EXTRACTION PROGRAM FOR THE BSIM MODEL

by

Joseph R. Pierret

Memorandum No. UCB/ERL M84/99

21 November 1984

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

A MOS Parameter Extraction Program For the BSIM Model

Joseph R. Pierret

Department of Electrical Engineering
and Computer Sciences
University of California Berkeley
Berkeley, California

ACKNOWLEDGEMENTS

I would like to sincerely thank Professors David A. Hodges and Ping K. Ko for their inspiration and knowledge while I completed this project, working towards the Masters of Engineering Degree at the University of California, Berkeley. I would also like to thank Anne K. Manaut for her support, patience, and understanding.

Research sponsored by the Defense Advanced Research Projects Agency
Contract No. N00039-84-C-0107.

A MOS Parameter Extraction Program For the BSIM Model

Joseph R. Pierret

Department of Electrical Engineering
and Computer Sciences
University of California Berkeley
Berkeley, California

ABSTRACT

A modified MOS transistor parameter extraction program has been developed using the modified BSIM model (Berkeley Short channel Igfet Model) and version CSIM3.2 of the CSIM (Compact Short channel Igfet Model) parameter extraction program. New features include manual probe station capability, BSIM parameter vs. W or L graphics, I-V graphics capabilities using electrical parameter generation from the output process file, verification with the SPICE model, and continuous error checking during operation of the program. After process file generation, circuit simulation with the size-independent parameters may proceed using SPICE.

November 21, 1984

Table of Contents

CHAPTER 1. Description of the Project

1.1. Introduction

This project involved modifying the existing CSIM3.2 parameter extraction program to make it more useful and reliable. It is assumed that the reader has already read the papers describing the CSIM model [1], the implementation of the model into the circuit simulation program SPICE [3], and the previous extraction program [2]. However, a quick summary is given here. The CSIM model (Compact Short-channel Igfet Model), modified to the BSIM model (Berkeley Short-channel Igfet Model) at Berkeley[1], presently has 17 parameters which describe both NMOS and PMOS transistor electrical action. These parameters include 5 describing the threshold voltage V_{th}, 5 describing the gain β, 1 describing mobility reduction due to large transverse electric fields caused by high gate voltage, 1 describing velocity saturation due to high electric fields in short channels, and 5 describing second order effects on these first 12 parameters. An extraction program run on an HP9836 mini-computer was developed which extracts these 17 chanel width and length dependent parameters from devices on a wafer[2]. After the extraction of parameters from many devices, a size-independent process file containing 54 parameters is developed. These parameters are then transported to a main frame computer where they are used in the circuit simulation program SPICE[3].

The work described in this paper improves the previous extraction program in a number of ways. The next section describes the major goals of the project.

1.2. Goals of Project

The primary goals of this project included:

(1) Allow the extraction program to operate not only with an automatic probe station, but also to operate with a manual probe station. This will allow more potential users of the program.

(2) Allow playback in graphic form of any BSIM parameter vs. $\frac{1}{W}$ or $\frac{1}{L}$. A comparison between the 17 size-dependent electrical parameters and the 17 simulated electrical parameters generated from the 54-parameter, size-independent process file should be made. This comparison enables one to see how good a fit any parameter has, and thus expedite the decision making for possible future changes to any parameter.

(3) Allow playback in graphics form of I-V curves using the 54-parameter process file. Since the process file is what the circuit designer will use and what SPICE will receive as it's input, this step allows verification of the I-V curves before running SPICE, and hence savings of valuable testing time.

(4) Implementation of continuous error-checking throughout the program. This should include checks to assure that measured data is within acceptable ranges, that extracted parameters are reasonable, and that playback of curves, both I-V and BSIM parameters, are meaningful.

(5) Verification of the extraction program with the model that is implemented in SPICE.

Each of these 5 goals are the main topics for chapter titles.

Throughout this report, responses from the computer are shown in **bold** type, while procedure, function, and variable names from the program itself are shown in *italic* type.

CHAPTER 2. Manual Probe Station Capability

2.1. General Description

A manual probe station may now be used when running the BSIM parameter extraction program. This added feature allows essentially any MOS structure to have parameters extracted from it, not only those which have been designed with an automatic probe station and associated probe card in mind.

The primary changes in the code were accomplished by simply not calling the automatic prober procedures. Instead of automatically moving to the next device on the wafer, one of the following messages appear on the screen after each device has been tested.

Move the probes to the next device and Press "ENTER" >

Move probes to 1st device of next die. Then Hit "ENTER" >

One probing procedure, *step_to_next_die*, is still called because although automatic stepping does not actually occur with a manual probe station, bookkeeping of location on the die is performed in this procedure.

2.2. Details of Manual Probe Station

When using a manual probe station in the Semi-Automatic mode, it is recommended to take advantage of the program pausing for movement, and to manually test the device with the HP4145 Parameter Analyzer. This extra step will help to assure proper connections, functionality, etc., and therefore help to minimize the time involved and maximize efficiency. If the previous device tested has the same SMU connections, then all this involves is changing the HP4145 to LOCAL mode, and pushing the SINGLE MEASUREMENT button.

Since the size of the die is irrelevant to a manual prober, the first two lines of the prober file, listing the x and y die size, *must* be omitted from the prober file when operating in the Semi-Automatic with Manual Prober mode.

The largest device of *each* device type *must* be placed first in the prober file, and consequently be tested first. For example, if both NMOS depletion and NMOS enhancement devices are to be tested, the largest device of each type must be placed before any other devices of the same device type. It is fine to test smaller NMOS depletion devices before the largest NMOS enhancement device as long as the largest NMOS enhancement device is placed before all other NMOS enhancement devices and the first NMOS depletion device tested is the largest one. This is necessary because the largest device is used to extract **PHIF2**. If the largest device is not placed first in the prober file, a smaller, less optimal size device will be used for the extraction of **PHIF2**.

CHAPTER 3. BSIM Parameter vs. 1/L or 1/W Graphing

3.1. Introduction

A graphing facility to view any BSIM parameter vs. 1/W or 1/L has been implemented. Currently, 16 of 17 parameters are modeled with respect to $\frac{1}{W}$ or $\frac{1}{L}$. Future work might result in different equations to model the parameters, perhaps each parameter having it's own equation. Having the facility to view any electrical parameter and compare it to it's simulated parameter can be a very helpful tool in the evolution of parameter equation development.

There are two *types* of BSIM parameters and both are plotted:

(1) the 17 size-dependent, electrical parameters which are extracted for each device during the course of running the BSIM extraction program. For example, **VFB**, **PHIF2**, **X2ETA** , and in general denoted as $\mathbf{P}_i$ where i stands for the i'th device tested on a particular die, and

(2) the 54 size-independent parameters which make up the process file. These parameters, denoted by $\mathbf{P}_0$, $\mathbf{P}_L$, and $\mathbf{P}_W$ in Equation 3.1. have been generated through a least squares in three variables algorithm using the $\mathbf{P}_i$ parameters from (1) above, along with the corresponding W_i and L_i, from every device on the die.

$$\mathbf{P}_i = \mathbf{P}_0 + \frac{\mathbf{P}_L}{L_i + \Delta L} + \frac{\mathbf{P}_W}{W_i + \Delta W} \qquad 3.1$$

For playback of I-V curves, a set of 17 simulated, electrical parameters for any size device can then be generated by substituting *W* and *L* for that device back into Equation 3.1 to obtain $\mathbf{P}_i$.

A separate section, **BETA0 Scaling,** follows which describes the plotting of **BETA0**, which does not follow the above description.

3.2. User Inputs

At the beginning of the program during the entry of necessary information, the user is asked whether or not he wants to enter the BSIM parameter graphing procedures. If the decision to enter the procedures is yes, the graphing procedure will be entered after *each* die has been tested and parameters exist for that die.

A few pieces of information are required before graphing can proceed. Figures 3.1 through 3.3 show the three pages which are displayed to the user.

```
***BSIM PARAMETER vs. W or L GRAPH***

This graphics mode allows one to compare extracted, size-DEPENDENT parameters
from the 17-parameter ELECTRICAL file, to size-INDEPENDENT values, approximated
from the 54-parameter PROCESS file.

If you plot W on the x-axis, then L becomes the 3rd variable, and vice versa.
You may choose to plot only one third-variable value, or you may plot all of
them.  Choosing only one allows finer details to be analyzed.  The x-axis
values are scaled linear with respect to 1/EFFECTIVE SIZE.

You will choose:  1) the type of device to plot
                  2) the BSIM parameter to plot on the y-axis
                  3) whether W or L will be plotted on the x-axis
                  4) and whether all sizes or one size device will
                     be plotted for the third parameter

SELECT THE DEVICE TYPE YOU WANT TO PLOT=  >
          [1] NMOS enhancement
```

Figure 3.1

1st Page of BSIM Parameter User Inputs

W/L ratios of devices successfully tested are listed here:

W	50.0	50.0	50.0	50.0	50.0	50.0	50.0	25.0	15.0	10.0
L	25.0	15.0	10.0	8.0	6.5	5.0	3.0	10.0	10.0	10.0
W	8.0	6.5	5.0	4.0	3.0	2.5	2.0			
L	10.0	10.0	10.0	10.0	10.0	10.0	10.0			

```
SELECT DESIRED GRAPH=? >
          [1] BSIM PARAMETER vs. W --- for all values of L
          [2] BSIM PARAMETER vs. L --- for all values of W
          [3] BSIM PARAMETER vs. W --- for single value of L.  L=? >
          [4] BSIM PARAMETER vs. L --- for single value of W.  W=? >
```

Figure 3.2

2nd Page of BSIM Parameter User Inputs

```
SELECT THE PARAMETER TO BE GRAPHED=1
          [1] VFB
          [2] 2PHIF
          [3] K1
          [4] K2
          [5] ETA
          [6] BETA0
          [7] U0
          [8] U1
          [9] X2MU0
         [10] X2ETA
         [11] X3ETA
         [12] X2U0
         [13] X2U1
         [14] MU0SAT
         [15] X2MU0SAT
         [16] X3MU0SAT
         [17] X3U1

Press a "c" to make any changes to your choices, or press "ENTER" to begin >
```

Figure 3.3

3rd Page of BSIM Parameter User Inputs

Page 1 explains the BSIM Parameter vs $\frac{1}{L}$ or $\frac{1}{W}$ graphing procedures, and asks for the device type to be plotted. Only those device types which had at least one device of it's kind *successfully* tested on the current die will be listed. This reduces the chance that the user will ask for a device type which does not exist for the current die.

Page 2 lists the drawn $\frac{W}{L}$ values for all devices which were *successfully* tested, and asks the user to choose the type of graph he would like to view. A single value of W or L as the third variable may be chosen to provide a clearer graph.

Page 3 asks for the BSIM parameter to be graphed. Only values between 1 and 17 are accepted as a valid selection.

Checks are made to insure that the user has entered meaningful data. For example, if $\Delta L = -0.6$ and a graph for $L = 0.5$ is desired, then $L_{eff} = -0.1$ which will yield a useless graph. A complete list of errors which are checked is given in **CHAPTER 5**. After all necessary information is known, the graph chosen will be calculated and displayed.

3.3. X-Axis Parameter and Scale

The X axis parameter was chosen to be L_{eff} or W_{eff} , and is scaled linear with respect to $\frac{1}{L_{eff}}$ or $\frac{1}{W_{eff}}$ since the 54-parameter process file is based on Equation 3.1.

$$P_i = P_0 + \frac{P_L}{L_i + \Delta L} + \frac{P_W}{W_i + \Delta W} \qquad 3.1$$

Both L and L_{eff} or W and W_{eff} are displayed for convenience. Due to the choice of scaling, a straight line for the parameters generated from the process file always should be observed, with the size-dependent electrical parameters displayed as individual points on the graph.

3.4. BETA0 Scaling

The **BETA0** parameter is slightly different from the other 16 parameters. During the extraction process, **BETA0** is extracted for each device, denoted here as **BETA0**$_i$. The development of the process file for all **BETA0**$_i$ values results in values of **MUO**$_0$, ΔL, and ΔW. This is discussed in more detail in **APPENDIX 1**. In order to obtain a straight line graph from the **MUO**$_0$ term and still have the X axis scale be $\frac{1}{L_{eff}}$ or $\frac{1}{W_{eff}}$, it was necessary to graph **BETA0** vs $\frac{1}{L_{eff}}$ and $\frac{1}{\text{BETA0}}$ vs $\frac{1}{W_{eff}}$ as described in Equation

3.2.

$$\text{BETA0} = \text{MU0}_0\, C_{ox} \frac{(W + \Delta W)}{(L + \Delta L)} \qquad 3.2$$

3.5. Details of BSIM Parameter Graph

Either BSIM Parameter vs L or BSIM Parameter vs W may be viewed. As mentioned above, the size-dependent electrical parameters are plotted as individual points. If BSIM Parameter vs L(W) is being plotted then along the right hand side of the graph, the drawn width(length) values, along with their plotting character are displayed. Ten different plotting characters are used, (lower case letters **a c e n o s u v x z**), corresponding to 10 different drawn widths(lengths). If more than 10 widths(lengths) are to be plotted, the characters are repeated. Each solid line, corresponding to a constant width(length), is noted by it's size which is labeled at one end of the line. ΔL and ΔW are displayed under the title of the graph.

CHAPTER 4. I-V Graphing

4.1. Introduction

The I-V graphing capability that previously existed has been expanded. I-V curves can now be generated using the 54-parameter process file as the input, whereas I-V curves for single devices using the 17 size-dependent parameters was all that existed before. This feature allows playback of any size device using the process file, which contains size-independent data.

4.2. User Inputs

After completion of extracting parameters for *the entire wafer*, the user is asked, prior to entering the I-V Graphics mode, whether he wants to view I-V curves. Upon answering yes, two pages of information are displayed. They are shown in Figures 4.1 and 4.2.

```
                    ***PREPARATION FOR I-V GRAPHICS***

ENTER X DIE POSITION OF DEVICE TO BE GRAPHED=  >

ENTER Y DIE POSITION OF DEVICE TO BE GRAPHED=  >
                                                  [1] NMOS enhancement
SELECT THE NUMBER CORRESPONDING                   [2] NMOS depletion
TO THE DEVICE TYPE WHICH                          [3] NMOS zero-threshold
YOU WOULD LIKE TO GRAPH=   >                      [4] PMOS enhancement
                                                  [5] PMOS depletion
                                                  [6] PMOS zero-threshold

DEVICE WIDTH (microns) =  >

DEVICE LENGTH (microns) =  >
```

Figure 4.1

1st Page of I-V Graphics User Inputs

```
                    ***BSIM I-V GRAPHICS MENU***

The BSIM I-V graphics routines will draw measured and/or simulated I-V data.
If the program is operating in the "SINGLE" mode, the 17 ELECTRICAL parameters
just extracted will be used. In the "AUTOMATIC" or "SEMI-AUTOMATIC" mode, the
17 ELECTRICAL parameters will be generated from the 54 parameter process file.

            SELECT A NUMBER FOR A GIVEN DISPLAY MODE=3
                 1)Measured Data Only
                 2)Simulated Data Only
                 3)Measured and Simulated Data

             SELECT A NUMBER FOR A GIVEN GRAPH TYPE=1
                 1)IDS versus VDS              VBS=?  >0
                 2)IDS versus VGS              VDS=?  >
                 3)ln(IDS) versus VDS          VBS=?  >

                    NEW SMU CONNECTIONS?  (Y/N)  >
```

Figure 4.2

2nd Page of I-V Graphics User Inputs

On the first page, if new SMU connections are needed, the user will be prompted to input them. If new connections are not needed, the connections from *the last device tested* will be displayed and used for the current device. This is shown in Figure 4.3.

```
                    SMU connected to DRAIN=1
                    SMU connected to GATE=2
                    SMU connected to SOURCE=3
                    SMU connected to BODY=4
Place probes on device and Press "ENTER"  >
```

Figure 4.3

SMU Connections Are Displayed

If any information is not acceptable, appropriate error messages are displayed and are discussed in further detail in **CHAPTER 5** If the program is being run in the SINGLE DEVICE extraction mode, only the 2nd page will be displayed, (without asking for SMU connections), since all information on the first page and SMU connections are known.

4.3. Calculation of I-V Values Using BSIM Electrical Parameters

The I-V equations have changed slightly from the previous model, due to **U1**, **X2U1**, and **X3U1** parameters changing and also due to the **BETA** terms being changed to **MU** terms. (see **APPENDIX 1.** for more detail) In summary, for playback of I-V curves, a set of 17 simulated, electrical parameters is calculated from the 54-parameter process file for a given value of L and W. Following this, the 17 parameters are substituted into the I-V equation. The following 3 sections describe these processes.

4.3.1. Conversion of Process File Parameters to 17 Simulated Electrical Parameters

Starting with the 54 parameter process file, **BETA0** is calculated using Equation 4.2.

$$\mathbf{BETA0} = \mathbf{MU0}_0 \, C_{ox} \frac{(W + \Delta W)}{(L + \Delta L)} \qquad 4.2$$

The remaining 16 BSIM parameters are calculated using Equation 4.3, where all values, except W and L, are taken from the process file.

$$\mathbf{P} = \mathbf{P}_0 + \frac{\mathbf{P}_L}{L + \Delta L} + \frac{\mathbf{P}_W}{W + \Delta W} \qquad 4.3$$

Four of these 16 terms, the 4 remaining **MU** terms, **X2MU0**, **MU0SAT**, **X2MU0SAT**, and **X3MU0SAT** are converted to **BETA** terms using Equation 4.4.

$$\mathbf{BETA} = \mathbf{MU} \, C_{ox} \frac{(W + \Delta W)}{(L + \Delta L)} \qquad 4.4$$

At this point, all 17 simulated electrical parameters have been generated.

4.3.2. Calculation of β

The calculation of β in the $I_{ds_{linear}}$ and $I_{ds_{saturation}}$ equations is performed by creating a parabolic fit, using the 5 simulated **BETA** terms. Figure 4.4 shows the approximation, where β_1, β_2, and β_3 are calculated in Equations 4.5, 4.6, and 4.7, and the value of β is calculated for any value of V_{ds} with Equation 4.8.

$$\beta_1 = \text{BETA0} + \text{X2BETA0}(V_{bs}) \qquad 4.5$$

$$\beta_2 = \text{BETA0SAT} + \text{X2BETA0SAT}(V_{bs}) \qquad 4.6$$

$$\beta_3 = \text{X3BETA0SAT} \qquad 4.7$$

$$\beta = \beta_1\left[\frac{V_{ds}}{V_{dd}} - 1\right]^2 + \beta_2\frac{V_{ds}}{V_{dd}}\left[2 - \frac{V_{ds}}{V_{dd}}\right] + \beta_3 V_{ds}\left[\frac{V_{ds}}{V_{dd}} - 1\right] \qquad 4.8$$

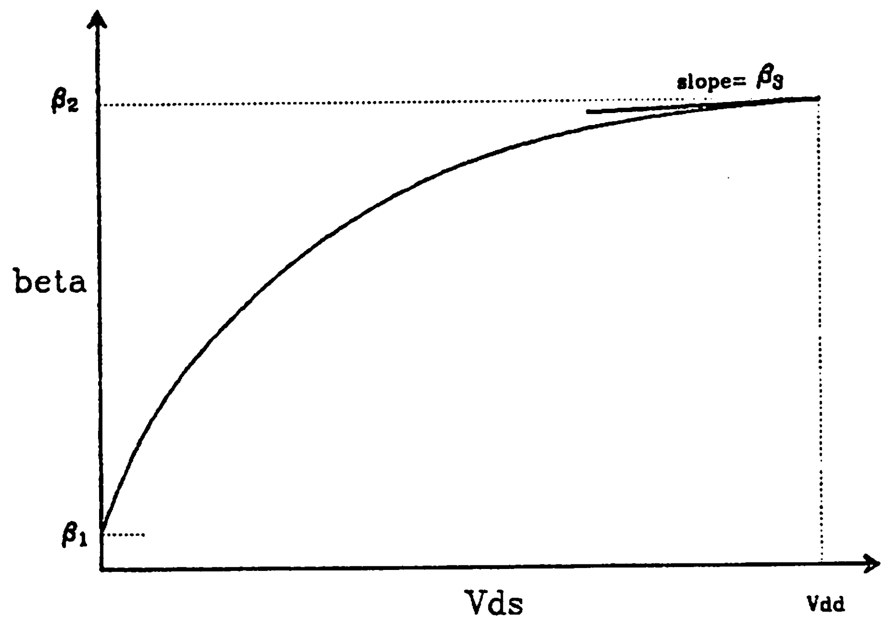


Figure 4.4

β Calculation and Parabolic Approximation

4.3.3. Final Substitution into I-V Equation

The results from above and the remaining simulated electrical parameters are substituted into Equations 4.9 through 4.16.

$$\eta = \text{ETA} + \text{X2ETA}(\text{PHIF2} - V_{bs}) + \text{X3ETA}(V_{ds} - V_{dd}) \qquad 4.9$$

$$\mu 0 = \text{U0} + \text{X2U0}(V_{bs}) \quad 4.10$$

$$\mu 1 = \text{U1} + \text{X2U1}(V_{bs}) + \text{X3U1}(V_{ds} - V_{dd}) \quad 4.11$$

If η , $\mu 1$, *or* $\mu 0 < 0.0$, then they are set = 0.

$$V_{th} = \text{VFB} + \text{PHIF2} + \text{K1}\sqrt{(\text{PHIF2} - V_{bs})} + \text{K2}(\text{PHIF2} - V_{bs}) - \eta(V_{ds}) \quad 4.12$$

$$a = 1.0 + \left[1.0 - \frac{1.0}{1.744 + 0.8364(\text{PHIF2} - V_{bs})}\right] \frac{\text{K1}}{2} \frac{1}{\sqrt{\text{PHIF2} - V_{bs}}} \quad 4.13$$

$$V_{ds_{sat}} = \frac{\sqrt{2}(V_{gs} - V_{th})}{a\left[1 + \frac{\mu 1(V_{gs} - V_{th})}{a \times L_{eff}} + \left(\frac{\mu 1(V_{gs} - V_{th})}{a \times L_{eff}}\right)^{½}\right]} \quad 4.14$$

If $V_{ds_{sat}} < V_{ds} - V_{th}$ then $I_{ds} = I_{ds_{linear}}$

$$I_{ds_{linear}} = \frac{\beta}{1 + \mu 0(V_{gs} - V_{th})}(V_{gs} - V_{th} - \frac{a}{2}V_{ds})V_{ds} \quad 4.15$$

If $V_{ds_{sat}} \geq V_{ds} - V_{th}$ then $I_{ds} = I_{ds_{saturation}}$

$$I_{ds_{saturation}} = \frac{\beta(V_{gs} - V_{th})^2}{(1 + \mu 0(V_{gs} - V_{th}))a\left[1.0 + \frac{\mu 1(V_{gs} - V_{th})}{a \times L_{eff}} + \left(1.0 + \frac{2 \times \mu 1(V_{gs} - V_{th})}{a \times L_{eff}}\right)^{½}\right]} \quad 4.16$$

CHAPTER 5. Error Checking

5.1. Introduction

As with any real time program, error checking should be performed continuously to assure not only accurate results, but also proper completion of the program. Continuous error checking minimizes the risk that valuable data, which has accumulated through successful extraction earlier in the program, will be lost due to an error later on in the program. Error checking in the BSIM program is performed at many points in the program, including during

1) input of necessary data
2) measurement of devices
3) extraction of parameters
4) process file development
5) playback of BSIM parameter curves, and
6) playback of I-V curves.

The flow chart in Figure 5.1 shows this global error checking with a detailed explanation for each section following.

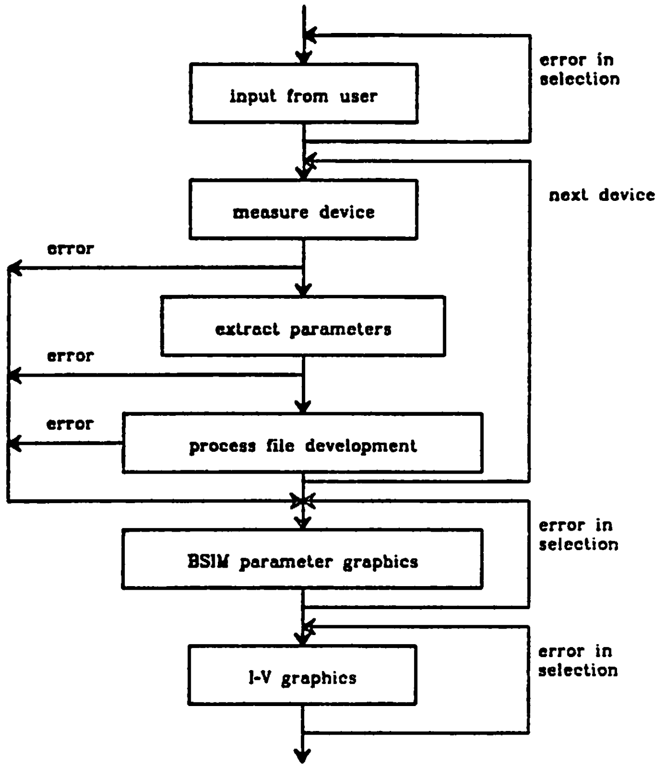


Figure 5.1
Flow chart of global error checking

5.2. User Input Error Checking

The second input menu page requires information from the user. If the user inputs 0 for **TOX** the following message will be displayed:

PLEASE ENTER NON-ZERO VALUE

As in the previous version of this program, after entering all required values, the user has a chance to change any value by pressing a *'c'* for *change*. This causes the entire page or pages to be redrawn and all inputs must be entered again.

5.3. Measurement Error Checking

Existing error messages for determining device functionality remain intact. These include:

****GATE SHORT****

SHORTED JUNCTION

****NO JUNCTION****

OPEN DRAIN-SOURCE

SHORTED DRAIN-SRC

New measurement error checks have also been added. After determining the device type and the proper functionality of the device, measurement proceeds. Four sets of data are measured: 2 in the linear region of operation and 2 in the saturation region. After *each* of the four measurements, the data is checked. In order to properly extract BSIM parameters, the device must have at least 5 *acceptable* values of I_{ds} for each V_{ds} and V_{bs} value, corresponding to 5 different V_{gs} values. *Acceptable* is defined simply as *above a certain threshold* $(0.1\times\frac{W}{L}\mu A)$. If 4 or less values exist, the measurement stops and the following error message is displayed:

NO PARAMETERS HAVE BEEN GENERATED OR SAVED: IDS-VGS ARRAY HAS A ZERO IN IT

5.4. Extraction Error Checking

The parameter extraction is performed first in the linear region of operation, and then in the saturation region. After the linear region parameters have been extracted, a validity check is made to verify that the parameters have reasonable values. They are checked to assure that they are within a predetermined minimum and maximum value range, shown in Table 5.1.

Parameter	minimum	maximum
VFB	-5.0	1.0
PHIF2	0.2	1.5
K1	0.0	5.0
K2	-1.0	1.0
U0	-1.0	1.0
X2U0	-1.0	1.0

Table 5.1
Minimum and Maximum Acceptable Parameter
Ranges for Linear Region Parameters

VFB can take on a large range of values because this variable will contain any shift in V_{th} due to implants, impurities, etc. If all range restrictions are not met, the following error message is displayed:

THESE PARAMETERS ARE NOT WITHIN ACCEPTABLE LIMITS & ARE NOT BEING SAVED

During the saturation region extraction, an estimate for V_{th} is made. Together with a measured value of V_{gs} a value of $V_{gs} - V_{th}$ is calculated. If $V_{gs} - V_{th} \geq 0.0$ then the device is found to be *on*. Otherwise, the device is *off*. If at least 3 sets of data are not found to be *on*, the 3-variable least square procedure can not be called and the following error message is displayed:

THESE PARAMETERS ARE NOT BEING SAVED: ONLY 2 SETS OF DATA FOR LEASTSQ3 PROC

At the successful completion of all 17 parameters being extracted, and before the parameters are stored away to later be used in the development of a process file, another validity check is made for all parameters. Table 5.2 lists the predetermined, acceptable minimum and maximum values for all parameters.

Parameter	minimum	maximum
VFB	-5.0	1.0
PHIF2	0.2	1.5
K1	0.0	5.0
K2	-1.0	1.0
ETA	-0.1	1.0
BETA0	0.0	1.0
U0	-0.1	1.0
U1	-0.1	1.0
X2BETA0	-1.0	1.0
X2ETA	-1.0	1.0
X3ETA	-1.0	1.0
X2U0	-1.0	1.0
X2U1	-1.0	1.0
BETA0SAT	0.0	1.0
X2BETA0SAT	-1.0	1.0
X3BETA0SAT	-1.0	1.0
X3U1	-1.0	1.0

Table 5.2
Minimum and Maximum Acceptable Parameter
Ranges for All Parameters

Again if any one parameter is not within a pre-determined acceptable range, the following error message is displayed:

THESE PARAMETERS ARE NOT WITHIN ACCEPTABLE LIMITS & ARE NOT BEING SAVED

5.5. Process File Development Error Checking

Once all devices on a die have had BSIM parameters extracted, a process file is developed. The 3-variable least square procedure is again used to give a straight line fit for each parameter versus $\frac{1}{L_{eff}}$ and $\frac{1}{W_{eff}}$. The least squares procedure must have at least 3 sets of data to function properly. It is possible that even after testing many devices on a die, that only 2 devices have yielded acceptable parameters. While the process file is being developed, this is checked, and if only 2 good devices exist, the following error message is displayed:

ONLY 2 GOOD DEVICES FOR NMOS-ENHANCEMENT. NO PROCESS FILE DEVELOPED

(The device type **NMOS-ENHANCEMENT** is changed, corresponding to the correct device type.)

5.6. BSIM Parameter vs. 1/W or 1/L Error Checking

After successful extraction, measurement, and process file development, the BSIM Parameter vs 1/W or 1/L Graphing mode will be entered, *if the user chose to do so at the beginning of the program.*

During entry of information on the first page, if a device type is chosen but no devices of that device type have been successfully tested, the following error message is displayed:

ERROR IN YOUR SELECTION.

PLEASE SELECT AGAIN

After entering necessary information for the graphing to begin, many checks are made to insure a meaningful graph. A process file might not have been developed due to a lack of enough good devices tested, entered data might have been entered incorrectly, or the extracted ΔL *or* ΔW might cause a small device to have an effective channel size of less than zero. Below are listed the possible error messages which will be displayed in the event of these errors:

GRAPHICS MENU CAN NOT BE ACCESSED BECAUSE OUTPUT FILE IS EMPTY.

GRAPHICS MENU CAN NOT BE ACCESSED BECAUSE A PROCESS FILE FOR THE TYPE OF DEVICE YOU REQUESTED WAS NOT CREATED AT THE CURRENT DIE POSITION.

DELTA L = ΔL . YOU HAVE SELECTED A DEVICE WITH AN EFFECTIVE CHANNEL LENGTH LESS THAN ZERO.

DELTA W = ΔW . YOU HAVE SELECTED A DEVICE WITH AN EFFECTIVE CHANNEL WIDTH LESS THAN ZERO.

Each of these warnings is followed by the question:

Do you want to select another graph? (Y/N) >

5.7. I-V Graphing Error Checking

Similar to the error checking in the BSIM parameter curves playback, error checking occurs during the playback of I-V curves. During the selection of SMU connections to the HP4145 Parameter Analyzer, if a value is selected which is not within the range of 1 to 4, the following error message is displayed:

SMU VALUES MUST BE BETWEEN 1 AND 4. PLEASE TRY AGAIN

During the selection of the X and Y die location, only values between 1 and 20 are acceptable, corresponding to the prober file die location map. If this condition is not met, the following error message is displayed:

***** VALUE MUST BE BETWEEN 1 AND 20 *****

After entering all necessary values, it is possible that the user has selected an incorrect die location, or that the output file might be empty. In the event this occurs, one of the following messages will be displayed:

GRAPHICS MENU CAN NOT BE ACCESSED BECAUSE THE OUTPUT FILE IS EMPTY. NO PROCESS FILES WERE CREATED AT ANY DIE LOCATION.

GRAPHICS MENU CAN NOT BE ACCESSED BECAUSE A PROCESS FILE FOR THE SELECTED DEVICE TYPE WAS NOT CREATED AT THE X AND Y DIE POSITIONS SHOWN ABOVE.

Each of these warnings is followed by the question:

Do you want to select another graph? (Y/N) >

5.8. General Error Checking Throughout the Program

During the extraction process and the process file development, the 3-variable least square routine is called repeatedly. If a divide-by-zero error is about to occur while in the routine, extraction and development ceases and an error message is displayed:

THESE PARAMETERS ARE NOT BEING SAVED: DIVIDE BY 0 IN LEASTSQ PROCEDURE

CHAPTER 6. Results of Testing

6.1. Introduction

This chapter will show some of the Parameter vs L or W curves and I-V curves which were generated during the normal operation of the program on the HP9836 mini-computer. A separate section will compare the I-V results with those which SPICE generated. Two wafers were used for the extraction: a CMOS wafer supplied by Signetics Corporation and an NMOS wafer supplied by National Semiconductor Corporation.

6.2. BSIM Parameter vs 1/L or 1/W Curves

Figures 6.1 through 6.6 show BSIM parameters, both extracted electrical parameters (individual points) and simulated electrical parameters (solid lines). **CHAPTER 3.** has further information on the actual generation of the graph. When comparing extracted to simulated values, always keep in mind the scale of the y-axis. It may appear that much discrepency lies between them, when in fact, the difference is slight considering that it would create little, if any, change in the I-V equations.

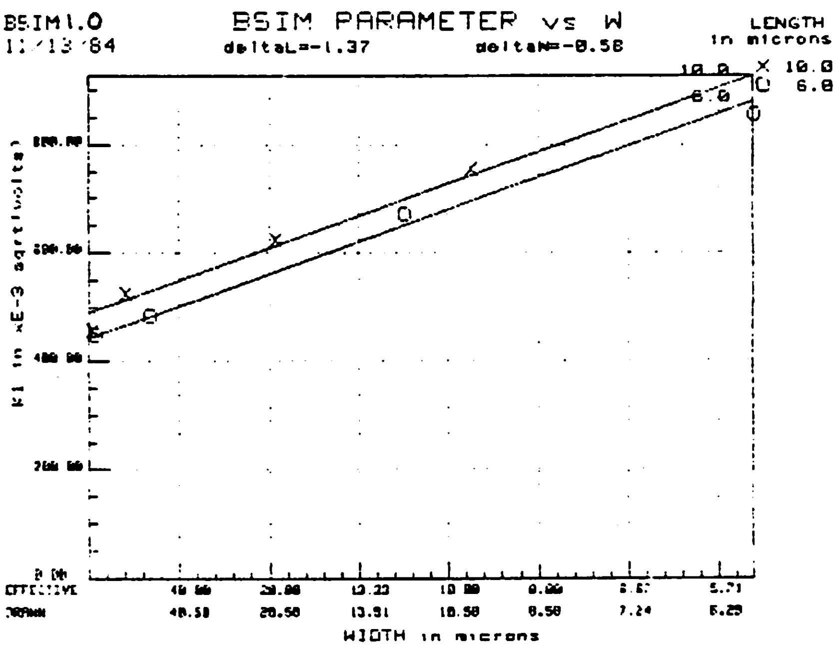


Figure 6.1
Signetics NMOS K1 vs W

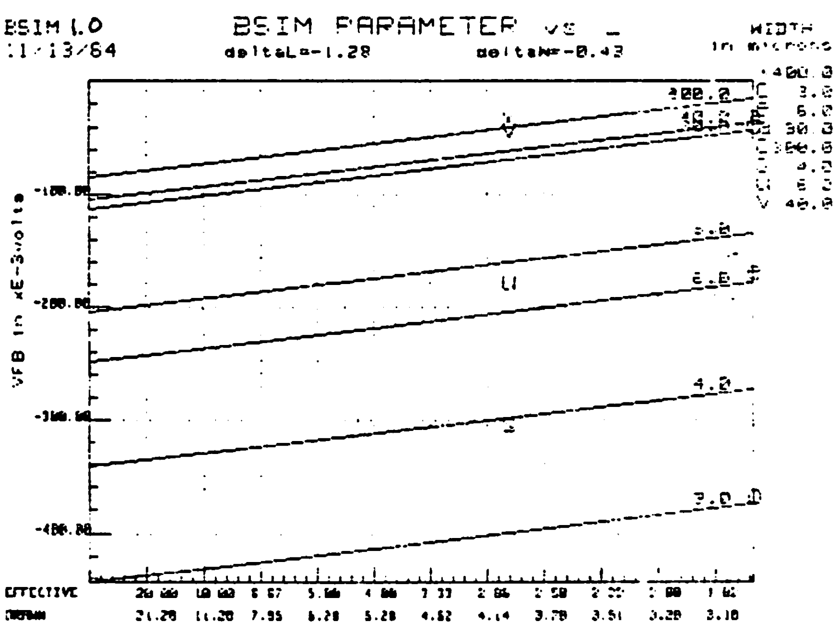


Figure 6.2
Signetics NMOS VFB vs L

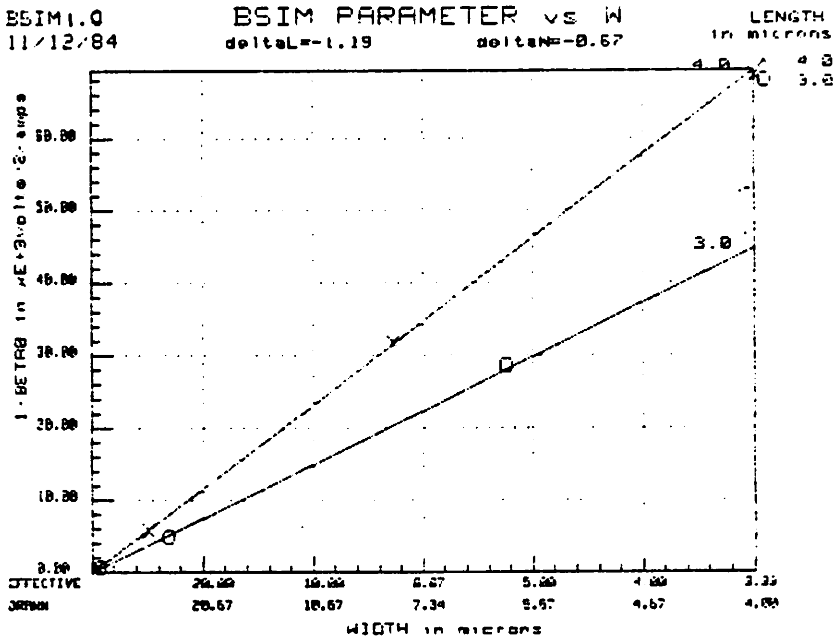


Figure 6.3
Signetics PMOS 1/BETA0 vs W

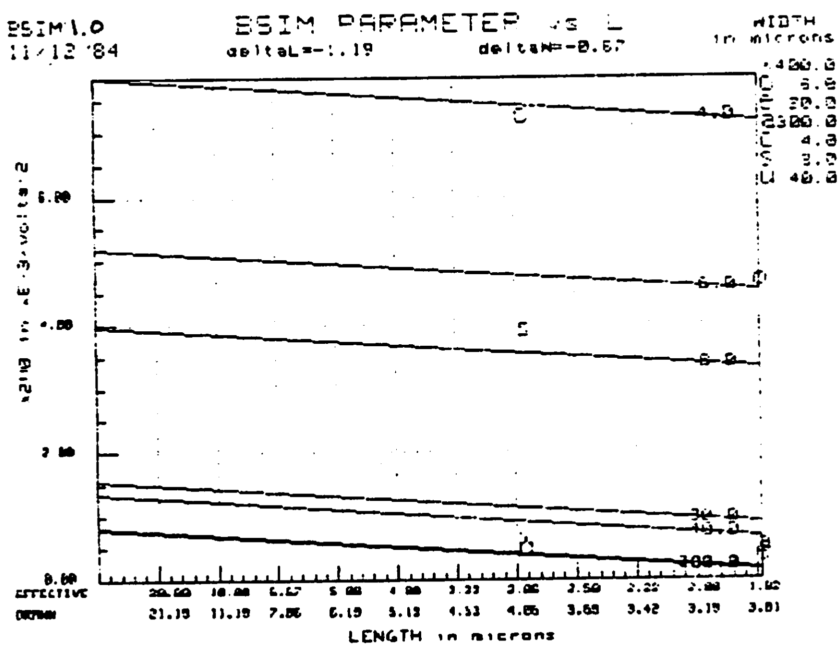


Figure 6.4
Signetics PMOS X2U0 vs L

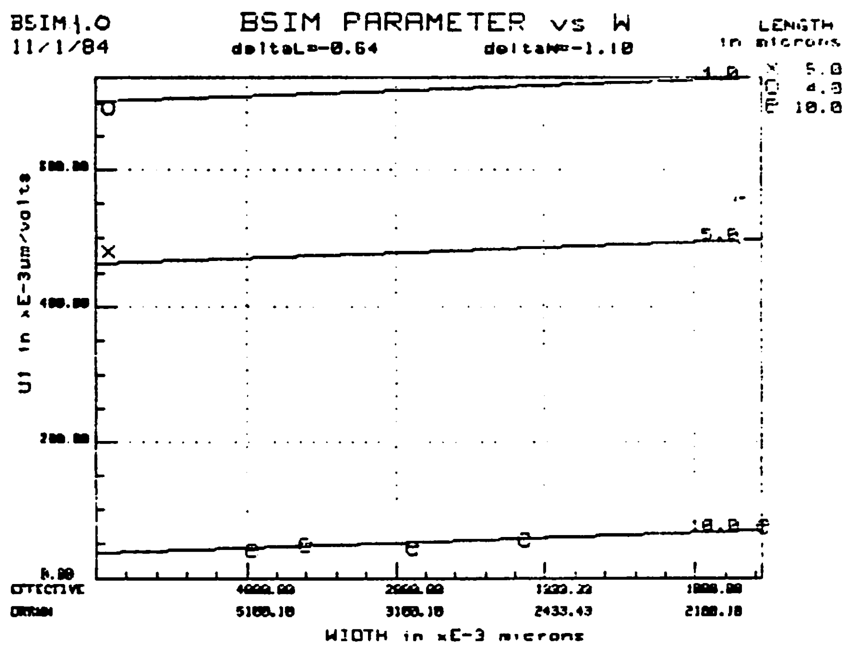


Figure 6.5
National NMOS U1 vs W

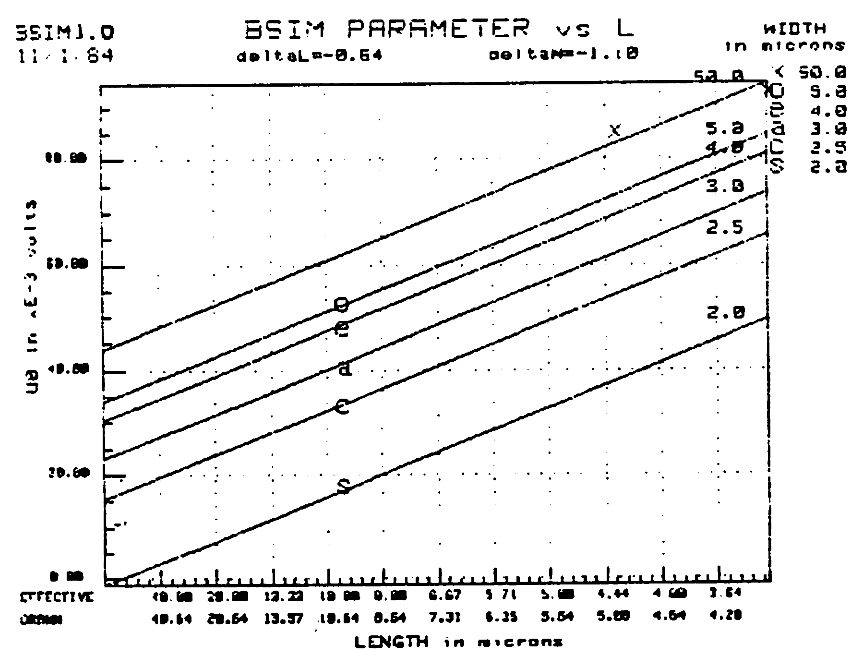


Figure 6.6
National NMOS U0 vs L

6.3. I-V Curves Generated from 54 Parameter Process File

Figures 6.7 through 6.12 show I-V curves generated from the extraction program. CHAPTER 4. describes the actual generation in more detail. The ×'s are measured data from the device and the solid lines are simulated using the process file as the input.

The accuracy of playback is in part determined by the choice of sizes of and the number of devices used for the parameter extraction. As a general rule, the more devices used to generate the process file, the more accurate the playback. There should be some sort of partitioning when choosing devices, and creation of more than one process file is advised. For example, if both short and long channel devices will be used in a design, then at least 2 seperate process files should be generated: one for the short devices and one for the long devices. This is evident when comparing Figure 6.7 and Figure 6.8. The measured curves are both from the same device, and both were generated with a process file that was developed with $\frac{W}{L} = \frac{50}{X}$ and $\frac{X}{10}$, but for Figure 6.7, X varied from 2 to 25, while for Figure 6.8 X varied from only 2 to 5. Figure 6.8 shows better simulation, and hence a better choice of devices for generating the process file.

Figure 6.9 shows playback for a PMOS device that was included in the development of the process file, while Figure 6.10 shows playback for a PMOS device that was not included in the development of the process file. The playback for both is quite good. However, caution is advised when using a process file for simulation of a device not within the size limits of the devices used to generate the process file. This type of playback will not occur for all size devices.

Figures 6.11 and 6.12 show a similar situation for NMOS devices.

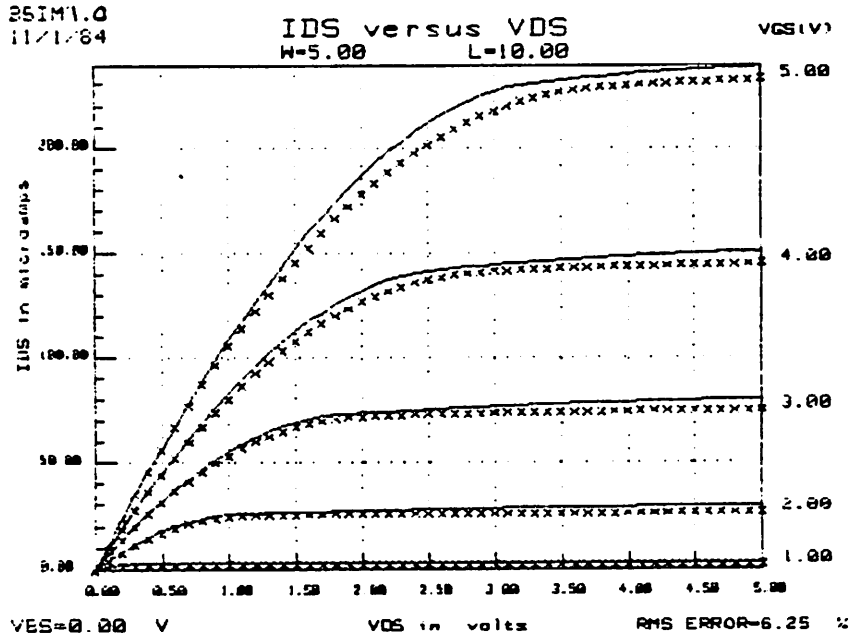


Figure 6.7
National NMOS Device

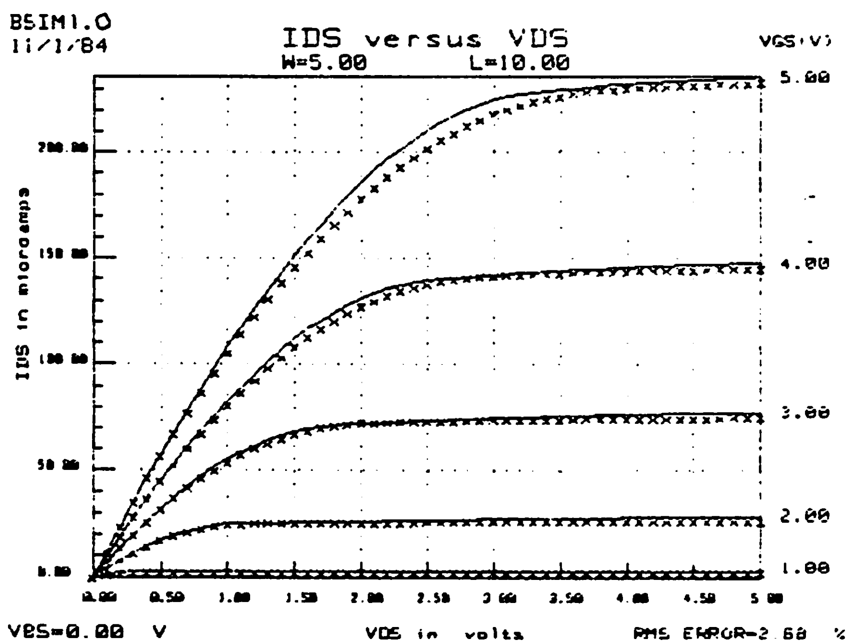


Figure 6.8
National NMOS Device

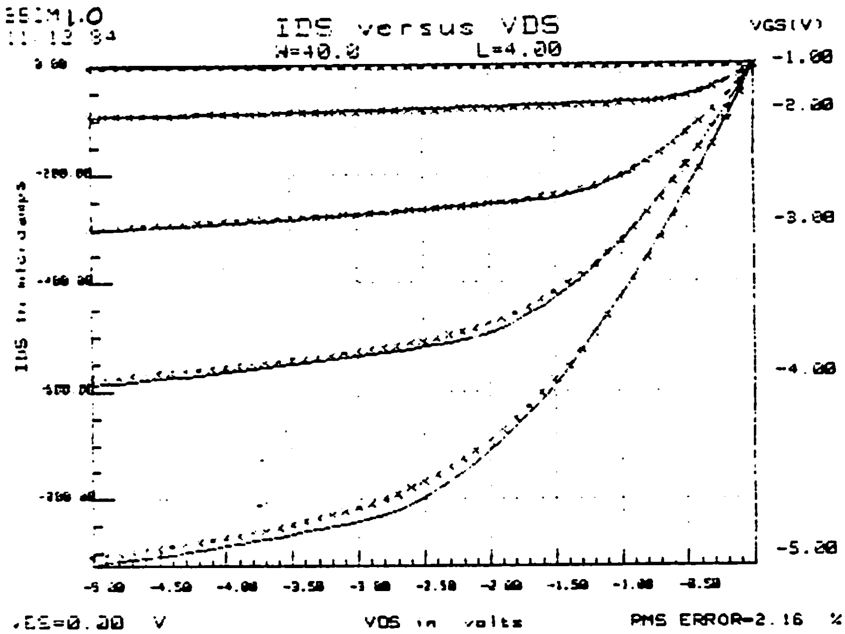


Figure 6.9
Signetics PMOS Device
Process File Devices $\frac{W}{L} = \frac{30}{3}, \frac{300}{3}, \frac{40}{4}, \frac{400}{4}, \frac{60}{6}, \frac{600}{6}, \frac{100}{10}$, and $\frac{1000}{10}$

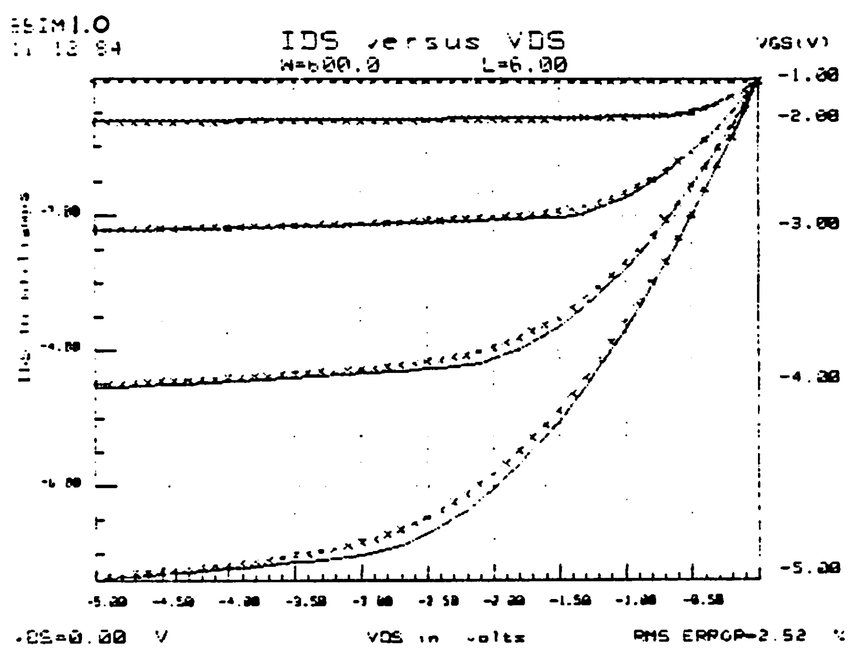


Figure 6.10
Signetics PMOS Device
Process File Devices $\frac{W}{L} = \frac{3}{3}, \frac{6}{3}, \frac{30}{3}, \frac{300}{3}, \frac{4}{4}, \frac{8}{4}, \frac{40}{4}$, and $\frac{400}{4}$

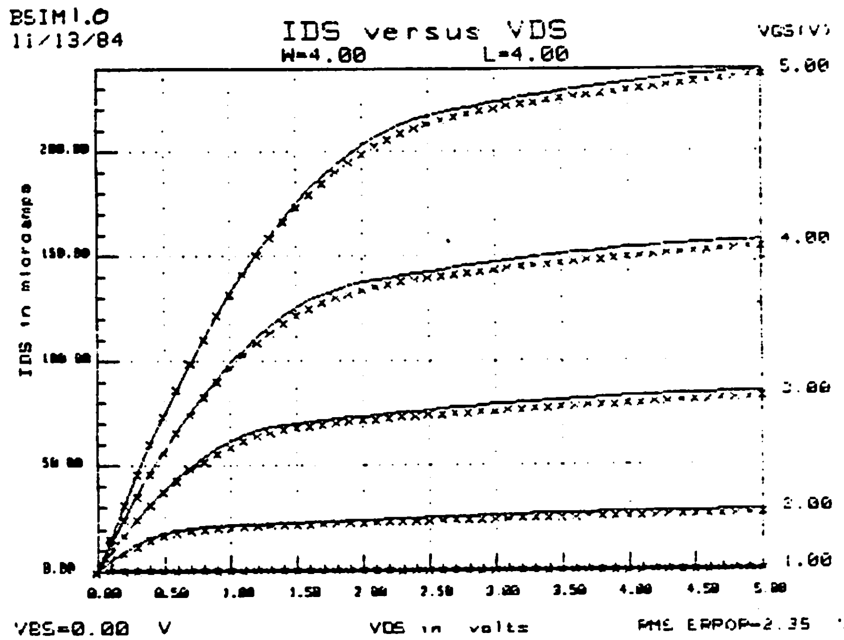


Figure 6.11
Signetics NMOS Device
Process File Devices $\frac{W}{L} = \frac{3}{3}, \frac{6}{3}, \frac{30}{3}, \frac{300}{3}, \frac{4}{4}, \frac{8}{4}, \frac{40}{4}$, and $\frac{400}{4}$

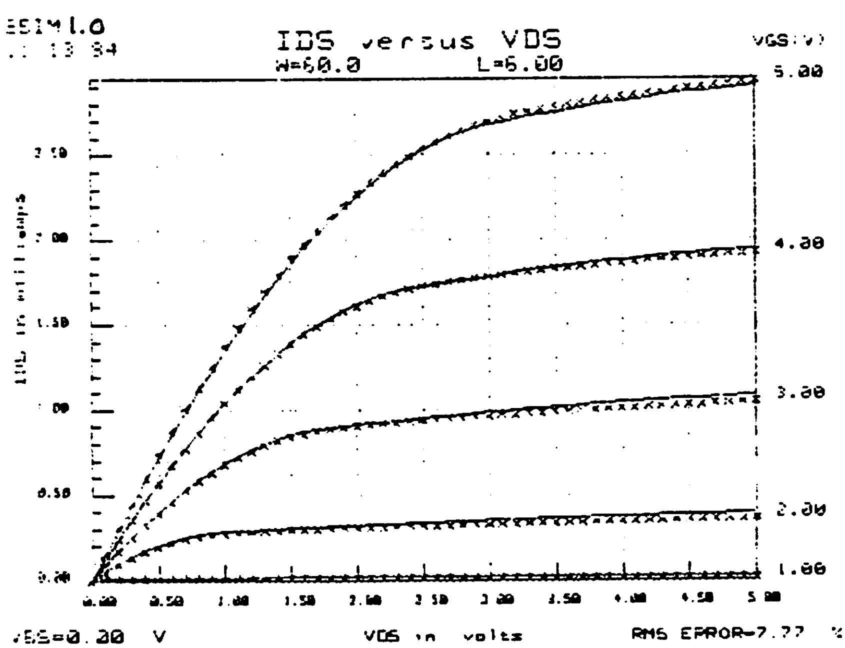


Figure 6.12
Signetics NMOS Device
Process File Devices $\frac{W}{L} = \frac{3}{3}, \frac{6}{3}, \frac{30}{3}, \frac{300}{3}, \frac{4}{4}, \frac{8}{4}, \frac{40}{4}$, and $\frac{400}{4}$

1.4. Verification With SPICE

The I-V curves generated on the HP9836 mini-computer were compared to those generated by SPICE on a VAX 11/780 computer. The results were identical for all graphs compared. This step ensures that the extraction program model and the SPICE model are identical. Figures 6.13 through Figure 6.16 compare the program results to SPICE results.

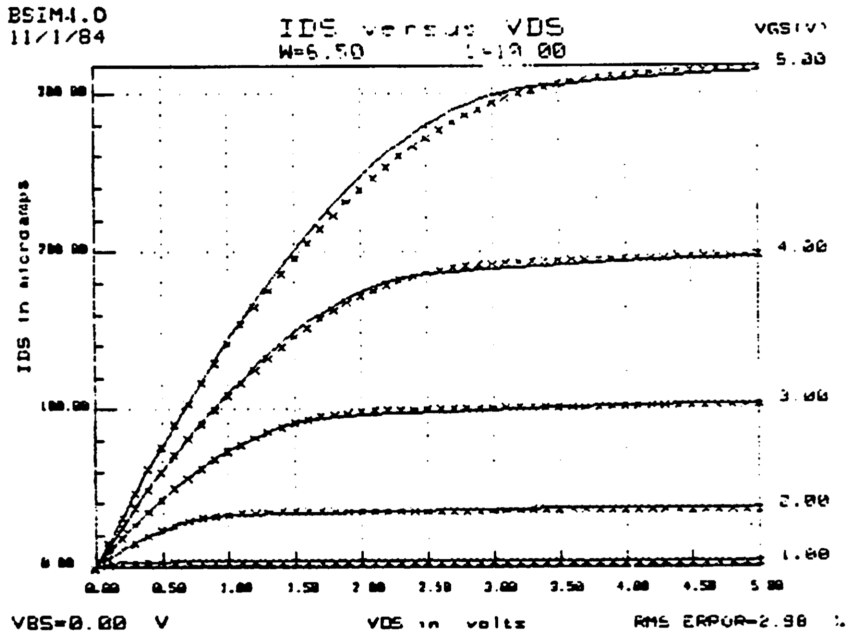


Figure 6.13
NMOS Generated from Program

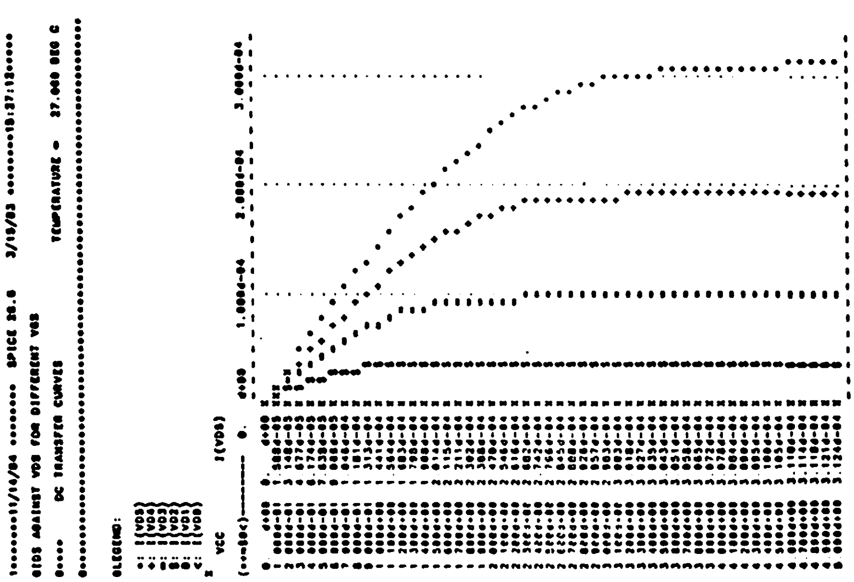


Figure 6.14
NMOS Generated from SPICE

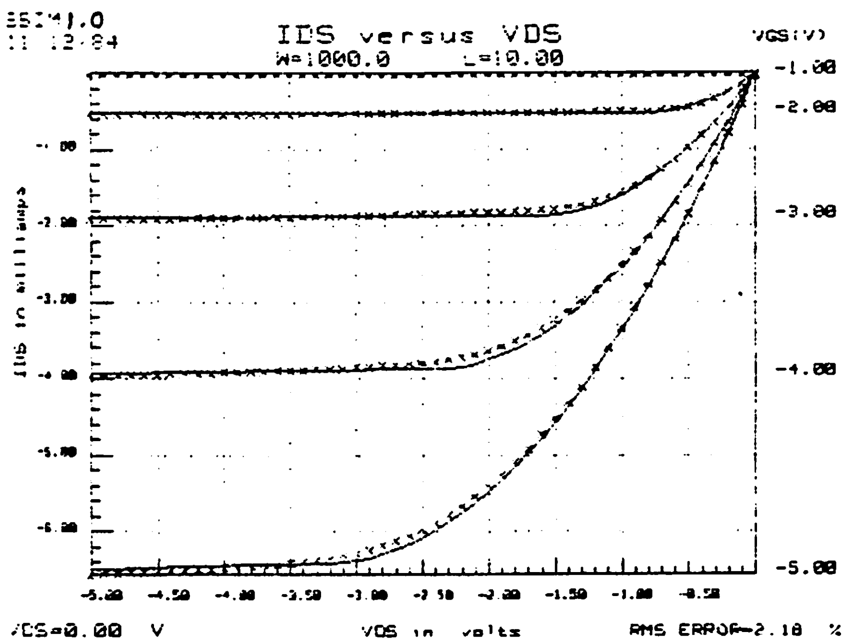


Figure 6.15
PMOS Generated from Program

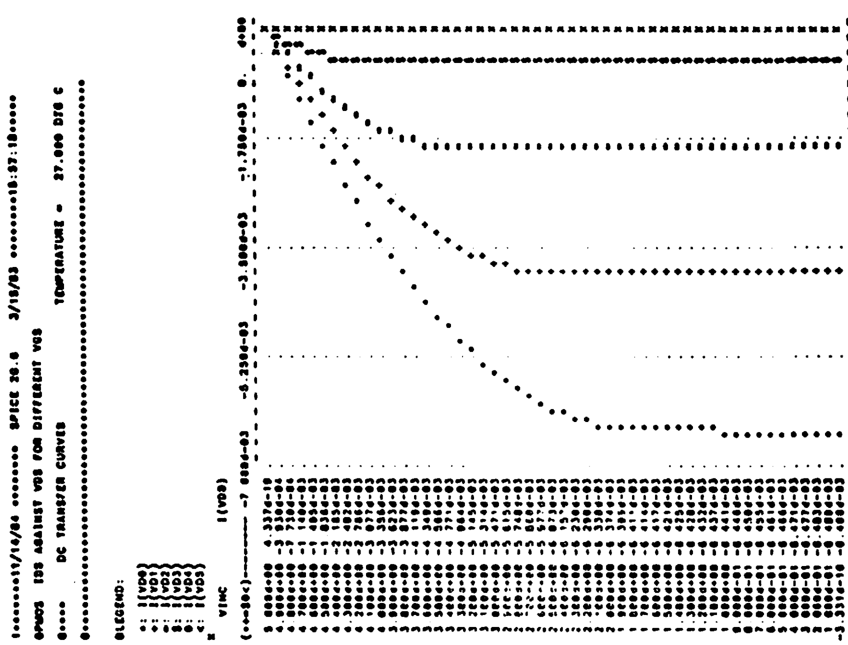


Figure 6.16
PMOS Generated from SPICE

REFERENCES

[1] Sheu, B. J., Scharfetter, D. L., and Poon, H. C., *Compact Short-Channel IGFET Model*, University of California, Berkeley, February 1984. Electronics Research Laboratory memo # M84/20.

[2] Messenger, B. S., *A Fully Automated MOS Device Characterization System For Process-Oriented Integrated Circuit Design*, University of California, Berkeley Master's Report, January 1984. Electronics Research Laboratory memo # M84/18.

[3] Liu, R. C. L., *An IC Process-Oriented Circuit Simulator Developed From SPICE2G.6 Using The CSIM MOSFET Model*, University of California, Berkeley Master's Report, January 1984. Electronics Research Laboratory memo # M84/9.

APPENDIX 1 Description and Units of All Parameters

A.1.1 Introduction

This appendix will describe all parameters, including the 17 electrical parameters and the 54 process file parameters. Generally speaking, the 17 electrical parameters from one die will be noted by **P**, the 17 electrical parameters from the i'th die as $\mathbf{P}_i$, and the 54 process file parameters as $\mathbf{P}_0$, $\mathbf{P}_L$, and $\mathbf{P}_W$. Eight of the BSIM parameters have undergone modifications from the previous program. These changes will be discussed here.

A.1.2 17 Electrical Parameters For Each Device

After the measurement and extraction process for a device has been completed, the 17 electrical parameters shown in Table A1.1 with it's corresponding units are displayed on the screen.

Parameter	Units
VFB	*Volts*
PHIF2	*Volts*
K1	$Volts^{-1/2}$
K2	$Volts^{-1}$
ETA	$Volts^{-1}$
BETA0	$\frac{Amps}{Volts^2}$
U0	$Volts^{-1}$
U1	$Volts^{-1}$
X2BETA0	$\frac{Amps}{Volts^3}$
X2ETA	$Volts^{-2}$
X3ETA	$Volts^{-2}$
X2U0	$Volts^{-2}$
X2U1	$Volts^{-2}$
BETA0SAT	$\frac{Amps}{Volts^2}$
X2BETA0SAT	$\frac{Amps}{Volts^3}$
X3BETA0SAT	$\frac{Amps}{Volts^3}$
X3U1	$Volts^{-2}$

Table A1.1
17 Electrical Parameters for Each Device

The measurement and extraction process continues for all devices on the current die, after which we have a set of 17 electrical, size-dependent parameters for each device. Figure A1.1 shows this.

Parameter	Parameter		Parameter
VFB_1	VFB_2		VFB_i
$PHIF2_1$	$PHIF2_2$		$PHIF2_i$
$K1_1$	$K1_2$		$K1_i$
$K2_1$	$K2_2$		$K2_i$
ETA_1	ETA_2		ETA_i
$BETA0_1$	$BETA0_2$		$BETA0_i$
$U0_1$	$U0_2$		$U0_i$
$U1_1$	$U1_2$	. . .	$U1_i$
$X2BETA0_1$	$X2BETA0_2$		$X2BETA0_i$
$X2ETA_1$	$X2ETA_2$		$X2ETA_i$
$X3ETA_1$	$X3ETA_2$		$X3ETA_i$
$X2U0_1$	$X2U0_2$		$X2U0_i$
$X2U1_1$	$X2U1_2$		$X2U1_i$
$BETA0SAT_1$	$BETA0SAT_2$		$BETA0SAT_i$
$X2BETA0SAT_1$	$X2BETA0SAT_2$		$X2BETA0SAT_i$
$X3BETA0SAT_1$	$X3BETA0SAT_2$		$X3BETA0SAT_i$
$X3U1_1$	$X3U1_2$		$X3U1_i$

Figure A1.1
17 Electrical Parameters
For Each Device on a Die

A.1.3 Calculation of ΔL and ΔW

The first step in the development of the process file is to calculate a value for ΔL and ΔW. After all 17 parameters for all devices on a die have been extracted, all **BETA0** terms, noted here as $\mathbf{BETA0}_i$, along with their respective drawn length (L) and drawn width (W) values, are sent to the least squares in three variables procedure. Equation A1.1 is rearranged to Equation A1.2 to accommodate the 3 variable fit.

$$\mathbf{BETA0}_i = \mathbf{MU0}_0 C_{ox} \frac{(W_i + \Delta W)}{(L_i + \Delta L)} \qquad \text{A1.2}$$

$$L_i = (-\Delta L) + \frac{W_i}{\mathbf{BETA0}_i}(\mathbf{MU0}_0 C_{ox}) + \frac{1}{\mathbf{BETA0}_i}(\mathbf{MU0}_0 C_{ox} \Delta W) \qquad \text{A1.2}$$

The values in parenthesis in Equation A1.2 are returned from the procedure. Since the value for C_{ox} is known, these values, after simple mathematics, result in $\mathbf{MUO}_0$, ΔL, and ΔW.

A 1.4 Parameter Modifications

Before the development of the process file continues, the remaining 4 sets of **BETA** terms which have not been modified yet, shown in Figure A1.1, are converted to **MU** (mobility) terms, and the 3 sets of **U1** terms are converted to **U1×LENGTH** terms.

A 1.4.1 BETA→MU

With the values for ΔL and ΔW known, the remaining 4 **BETA** terms, **X2BETA0**, **BETA0SAT**, **X2BETA0SAT**, and **X3BETA0SAT** can be modified to **MU** terms. As an example, consider **BETA0SAT**. Each of the **BETA0SAT** values, noted here as $\mathbf{BETA0SAT}_i$, have a specific W and L associated with it. Using the known values of ΔL and ΔW, all values of $\mathbf{BETA0SAT}_i$ are converted to $\mathbf{MU0SAT}_i$ using Equation A1.3.

$$\mathbf{MU0SAT}_i = \frac{\mathbf{BETA0SAT}_i}{C_{ox}} \frac{(L_i + \Delta L)}{(W_i + \Delta W)} \qquad \text{A1.3}$$

A similar transformation, shown in Equations A1.4 through A1.6 is made for the parameters **X2BETA0**, **X2BETA0SAT**, and **X3BETA0SAT** to obtain the new parameters **X2MU0**, **X2MU0SAT**, and **X3MU0SAT**.

$$\mathbf{X2MU0}_i = \frac{\mathbf{X2BETA0}_i}{C_{ox}} \frac{(L_i + \Delta L)}{(W_i + \Delta W)} \qquad \text{A1.4}$$

$$\mathbf{X2MU0SAT}_i = \frac{\mathbf{X2BETA0SAT}_i}{C_{ox}} \frac{(L_i + \Delta L)}{(W_i + \Delta W)} \qquad \text{A1.5}$$

$$\mathbf{X3MU0SAT}_i = \frac{\mathbf{X3BETA0SAT}_i}{C_{ox}} \frac{(L_i + \Delta L)}{(W_i + \Delta W)} \qquad \text{A1.6}$$

The need for this conversion of **BETA** terms to **MU** terms can be explained by looking at Figures A1.2 and A1.3.

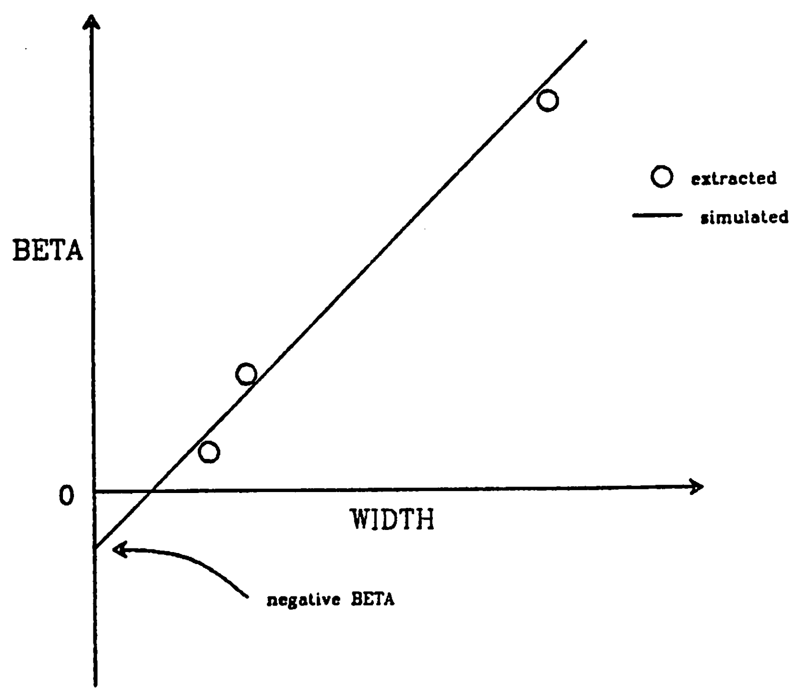


Figure A1.2
BETA vs. W

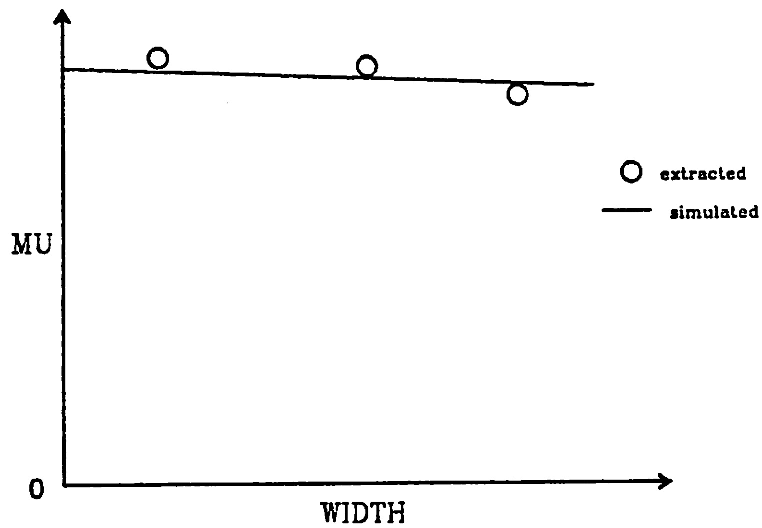


Figure A1.3
MU vs. W

The graph of **BETA** vs. W shows individual extracted values of **BETA0** plotted, and a least square fit of all the **BETA0** values. Notice that as W approaches 0, **BETA** is negative, and has been estimated incorrectly. This is caused by limitations in the least square routine.

Now convert each **BETA** term to a **MU** term and look at the graph of **MU** vs W. Since **MU** is nearly a constant, and should never approach 0, more accurate modeling results.

A.1.4.2 U1→U1×LENGTH

A conversion is also made on the 3 velocity saturation parameters **U1**, **X2U1**, and **X3U1**. For this transformation, each value is multiplied by it's respective effective length shown in Equation A1.7 through Equation A1.9.

$$\mathbf{U1}_{i-new} = \mathbf{U1}_{i-old} \times (L_i + \Delta L) \quad \text{A1.7}$$

$$\mathbf{X2U1}_{i-new} = \mathbf{X2U1}_{i-old} \times (L_i + \Delta L) \quad \text{A1.8}$$

$$\mathbf{X3U1}_{i-new} = \mathbf{X3U1}_{i-old} \times (L_i + \Delta L) \quad \text{A1.9}$$

Table A1.2 shows the parameters upon completion of these 8 transformations while Table A1.3 shows the units of the parameters.

Parameter	Parameter		Parameter
VFB_1	VFB_2		VFB_i
$PHIF2_1$	$PHIF2_2$		$PHIF2_i$
$K1_1$	$K1_2$		$K1_i$
$K2_1$	$K2_2$		$K2_i$
ETA_1	ETA_2		ETA_i
$BETA0_1$	$BETA0_2$		$BETA0_i$
$U0_1$	$U0_2$		$U0_i$
$U1_{1-new}$	$U1_{2-new}$	. . .	$U1_{i-new}$
$X2MU0_1$	$X2MU0_2$		$X2MU0_i$
$X2ETA_1$	$X2ETA_2$		$X2ETA_i$
$X3ETA_1$	$X3ETA_2$		$X3ETA_i$
$X2U0_1$	$X2U0_2$		$X2U0_i$
$X2U1_{1-new}$	$X2U1_{2-new}$		$X2U1_{i-new}$
$MU0SAT_1$	$MU0SAT_2$		$MU0SAT_i$
$MU0SAT_1$	$MU0SAT_2$		$MU0SAT_i$
$MU0SAT_1$	$MU0SAT_2$		$MU0SAT_i$
$X3U1_{1-new}$	$X3U1_{2-new}$		$X3U1_{i-new}$

Table A1.2
Electrical Parameters After Transformations

Parameter	Units
VFB	*Volts*
PHIF2	*Volts*
K1	$Volts^{-1/2}$
K2	$Volts^{-1}$
ETA	$Volts^{-1}$
BETA0	$\frac{Amps}{Volts^2}$
U0	$Volts^{-1}$
U1	$\frac{\mu m}{Volts}$
X2MU0	$\frac{cm^2}{Volt^2 \times sec}$
X2ETA	$Volts^{-2}$
X3ETA	$Volts^{-2}$
X2U0	$Volts^{-2}$
X2U1	$\frac{\mu m}{Volts^2}$
MU0SAT	$\frac{cm^2}{Volt \times sec}$
X2MU0SAT	$\frac{cm^2}{Volt^2 \times sec}$
X3MU0SAT	$\frac{cm^2}{Volt^2 \times sec}$
X3U1	$\frac{\mu m}{Volts^2}$

Table A1.3
Units for Transformed Electrical Parameters

A.1.5. Final Process File Development

The final step in developing the process file is to call the least squares in 3 variables procedure for each of the 17 parameters, *except* **BETA0** . (**BETA0** has previously had $MU0_0$, ΔL , and ΔW extracted from it) All values of the parameter (P_i) and their corresponding length (L_i) and width (W_i), are converted to process file parameters P_0 , P_L , and P_W , shown in Equation 1.1.

$$P_i = P_0 + \frac{P_W}{W_i + \Delta W} + \frac{P_L}{L_i + \Delta L} \qquad 1.1$$

The result is the process file, shown in Table A1.4. The corresponding units of the parameters are shown in Table A1.5.

Parameter	Parameter	Parameter
VFB_0	VFB_L	VFB_W
$PHIF2_0$	$PHIF2_L$	$PHIF2_W$
$K1_0$	$K1_L$	$K1_W$
$K2_0$	$K2_L$	$K2_W$
ETA_0	ETA_L	ETA_W
MUO_0	ΔL	ΔW
$U0_0$	$U0_L$	$U0_W$
$U1_0$	$U1_L$	$U1_W$
$X2MU0_0$	$X2MU0_L$	$X2MU0_W$
$X2ETA_0$	$X2ETA_L$	$X2ETA_W$
$X3ETA_0$	$X3ETA_L$	$X3ETA_W$
$X2U0_0$	$X2U0_L$	$X2U0_W$
$X2U1_0$	$X2U1_L$	$X2U1_W$
$MUOSAT_0$	$MUOSAT_L$	$MUOSAT_W$
$X2MUOSAT_0$	$X2MUOSAT_L$	$X2MUOSAT_W$
$X3MUOSAT_0$	$X3MUOSAT_L$	$X3MUOSAT_W$
$X3U1_0$	$X3U1_L$	$X3U1_W$
T_{ox}	$Temp$	V_{dd}

Table A1.4
Contents of Process File

Units	Units	Units
Volts	none	none
Volts	none	none
$Volts^{-1/2}$	$Volts^{-3/2}$	$Volts^{-3/2}$
$Volts^{-1}$	$Volts^{-2}$	$Volts^{-2}$
$Volts^{-1}$	$Volts^{-2}$	$Volts^{-2}$
$\frac{cm^2}{Volts \times sec}$	μm	μm
$Volts^{-1}$	$Volts^{-2}$	$Volts^{-2}$
$\frac{\mu m}{Volts}$	$\frac{\mu m}{Volts^2}$	$\frac{\mu m}{Volts^2}$
$\frac{cm^2}{Volt^2 \times sec}$	$\frac{cm^2}{Volt^3 \times sec}$	$\frac{cm^2}{Volt^3 \times sec}$
$Volts^{-2}$	$Volts^{-3}$	$Volts^{-3}$
$Volts^{-2}$	$Volts^{-3}$	$Volts^{-3}$
$Volts^{-2}$	$Volts^{-3}$	$Volts^{-3}$
$\frac{\mu m}{Volts^2}$	$\frac{\mu m}{Volts^3}$	$\frac{\mu m}{Volts^3}$
$\frac{cm^2}{Volts \times sec}$	$\frac{cm^2}{Volts^2 \times sec}$	$\frac{cm^2}{Volts^2 \times sec}$
$\frac{cm^2}{Volts^2 \times sec}$	$\frac{cm^2}{Volts^3 \times sec}$	$\frac{cm^2}{Volts^3 \times sec}$
$\frac{cm^2}{Volts^2 \times sec}$	$\frac{cm^2}{Volts^3 \times sec}$	$\frac{cm^2}{Volts^3 \times sec}$
$\frac{\mu m}{Volts^2}$	$\frac{\mu m}{Volts^3}$	$\frac{\mu m}{Volts^3}$
μm	$^\circ C$	Volts

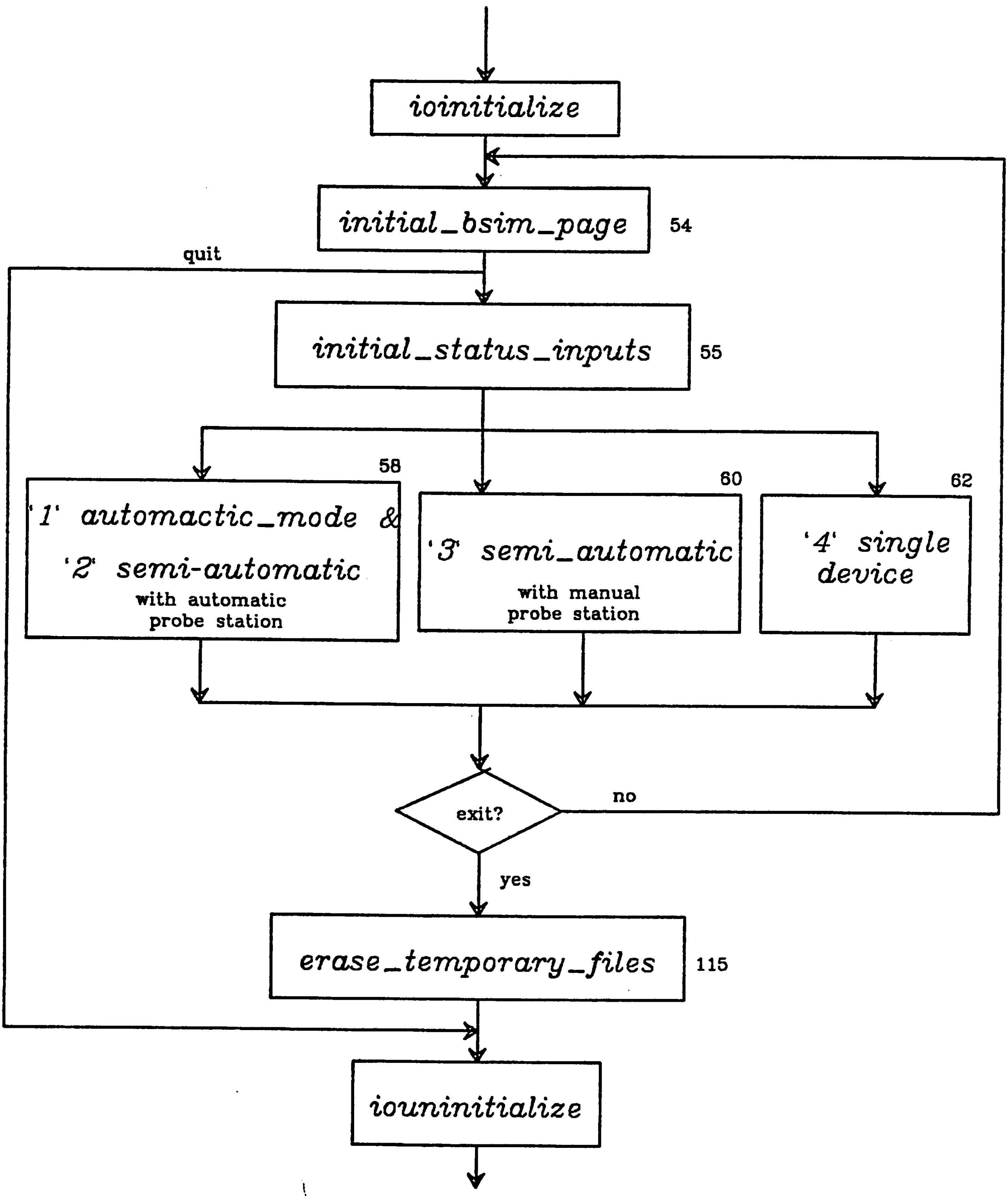
Table A1.5
Units of Parameters in the Process File

APPENDIX 2. Flow Charts and Description of All Procedures in Program

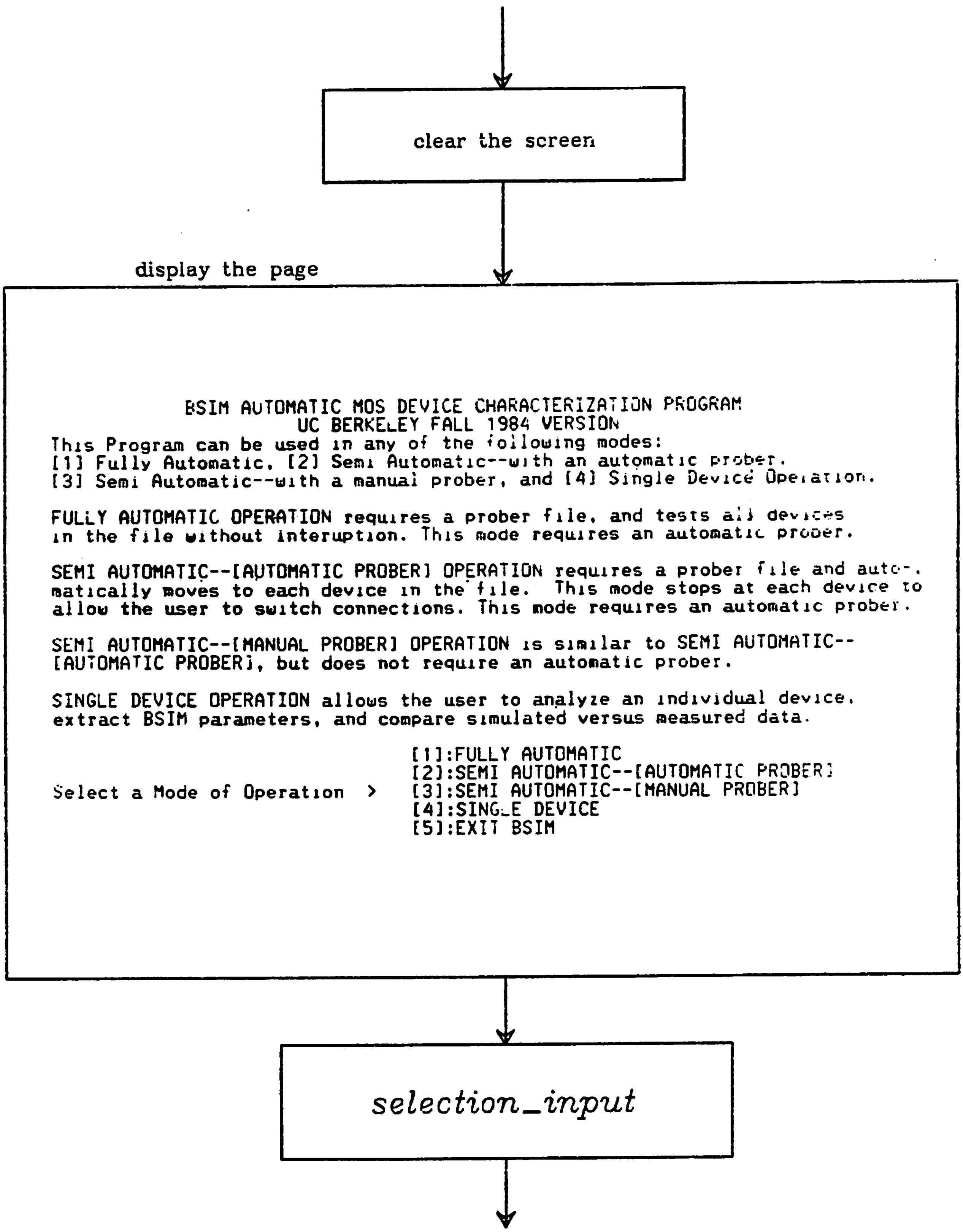
A.2.1 The Flow Charts

This appendix includes flow charts for the BSIM3.3 Parameter Extraction Program. The flow charts do not contain every detail of the program, but have enough information for someone to be able to understand the main flow of the program, which is necessary before continued work on this project is attempted. Procedure, function, and variable names are in *italic* print. Next to each procedure name is a page number where more information about the procedure may be found. This additional information may be in the form of another flow chart, a written description of the procedure, or in the case of the parameter extraction routines, a more detailed description with text, equations, and graphs.

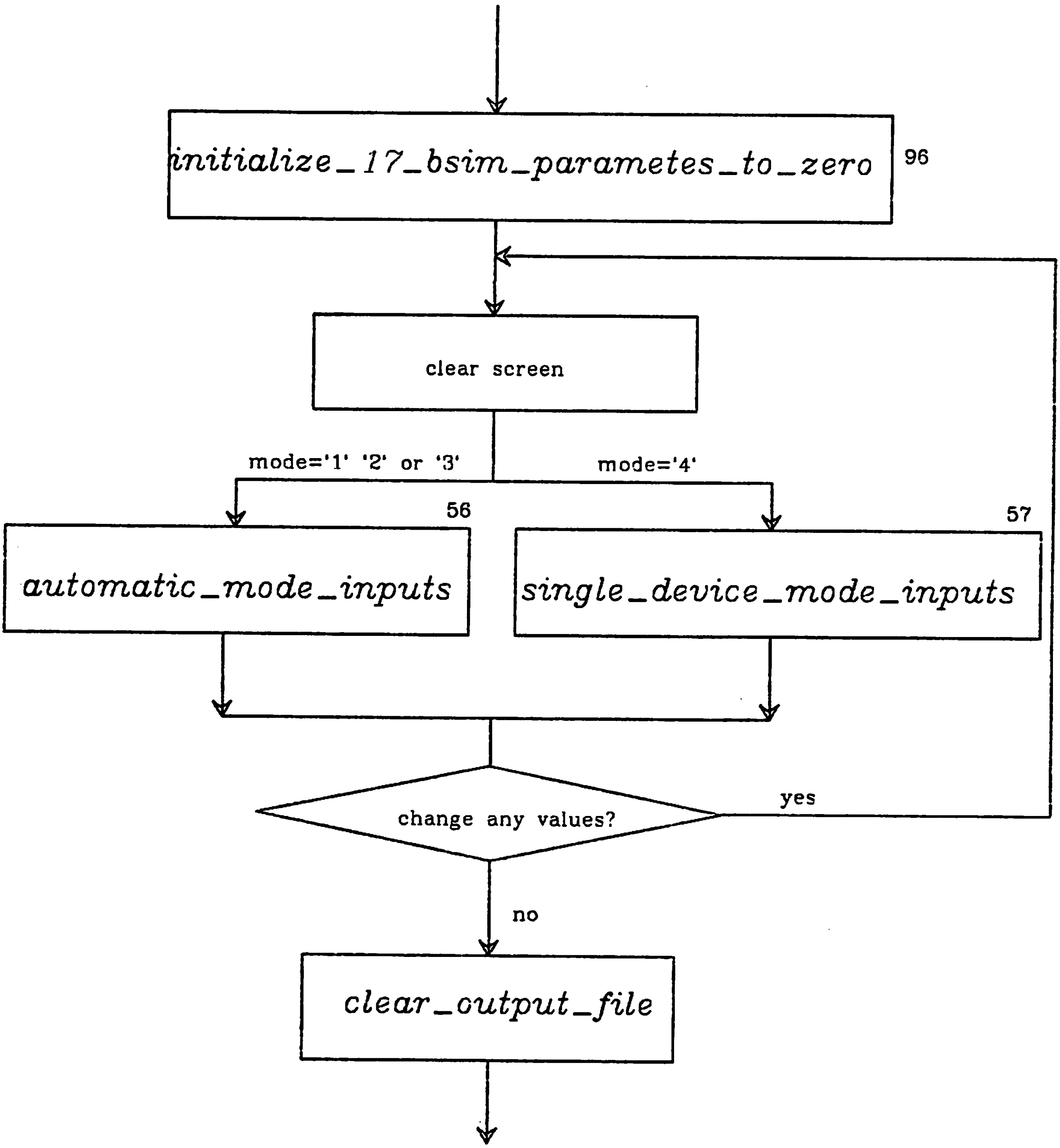
MAIN BSIM PROGRAM



initial_bsim_page



initial_status_inputs



automatic_mode_inputs

```
***AUTOMATIC OR SEMI-AUTOMATIC OPERATION***
```

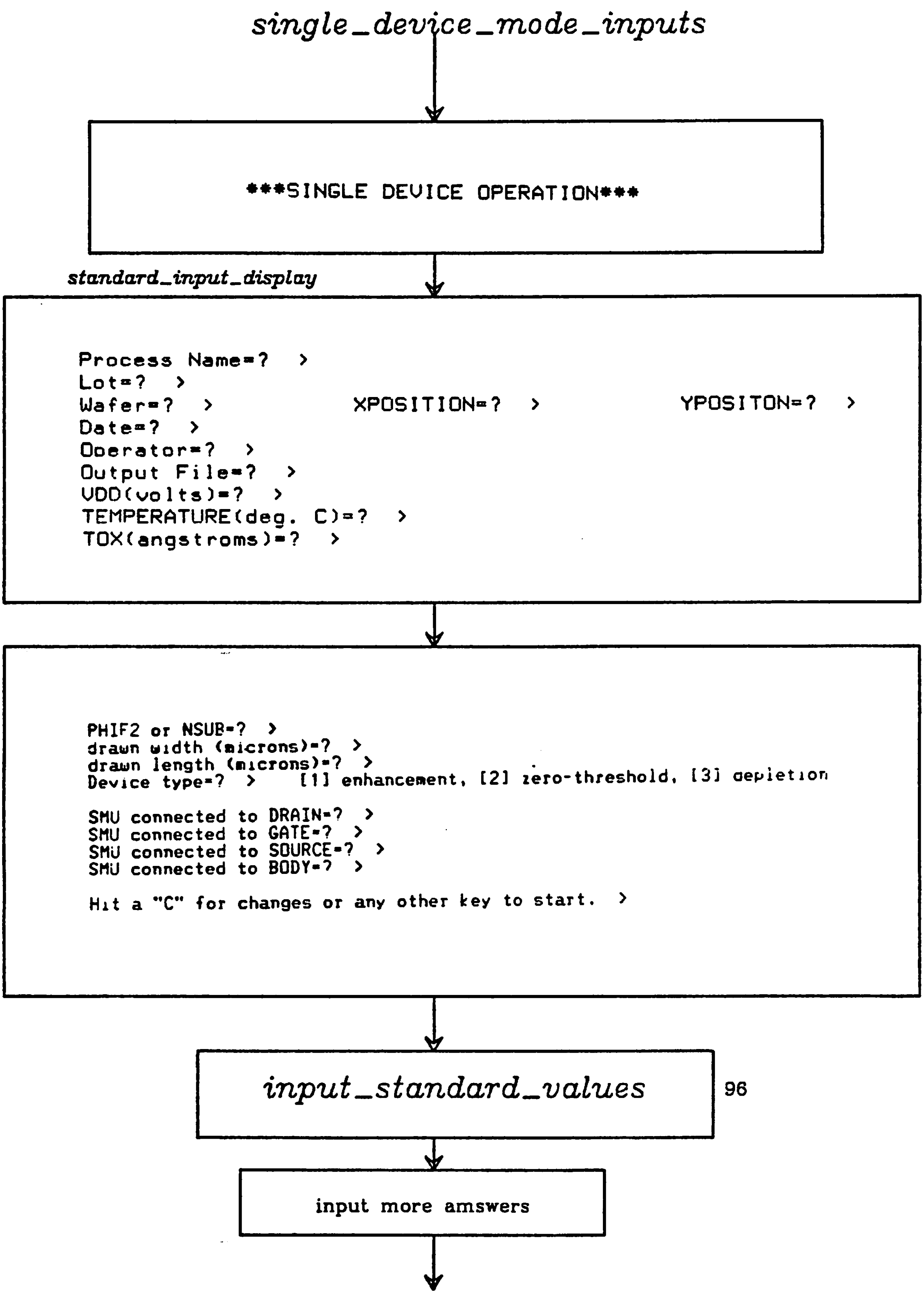
standard_input_display

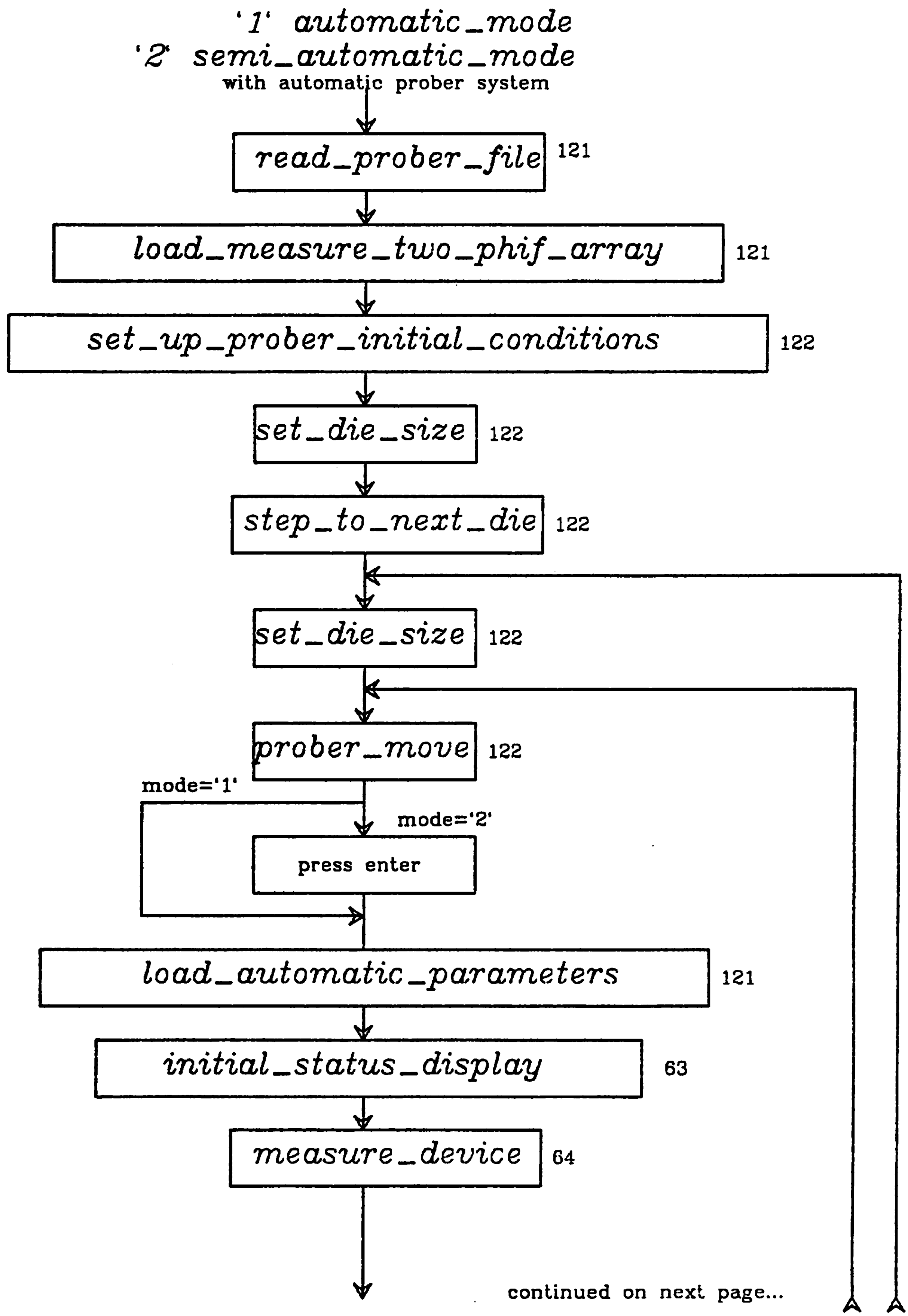
```
Process Name=?  >
Lot=?  >
Wafer=?  >
Date=?  >
Operator=?  >
Output File=?  >
VDD(volts)=?  >
TEMPERATURE(deg. C)=?  >
TOX(angstroms)=?  >
```

```
Prober File=?  >

At the end of EACH DIE, would you like to view plots of
BSIM PARAMETER vs W or L?  (Y/N)  >

Probing Instructions
     The prober should be on, and the probes should be down
on the starting die, starting position.  (see prober instructions)
HIT a "C" for changes, or any other key to start.  >
```

input_standard_values 96

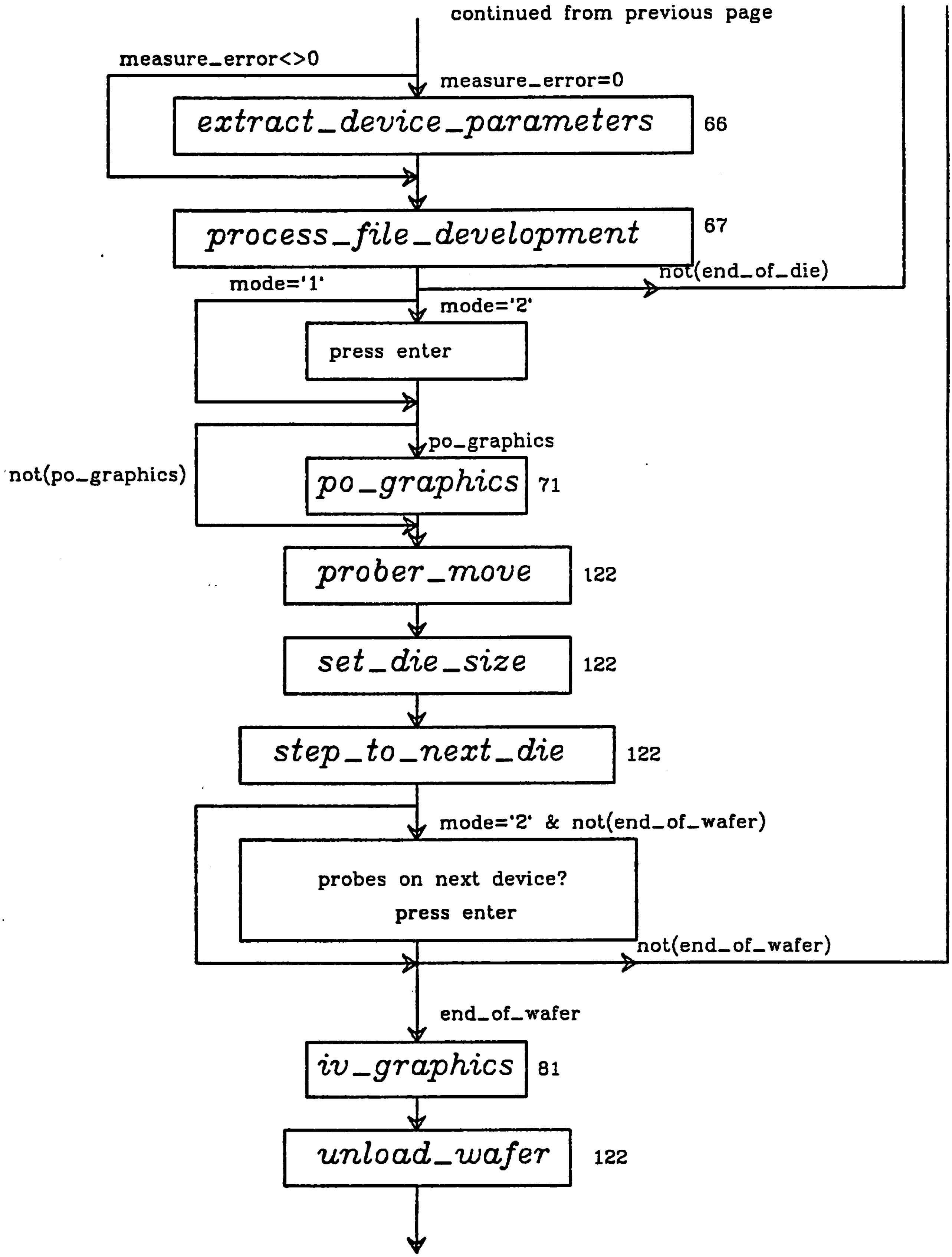
input more amswers

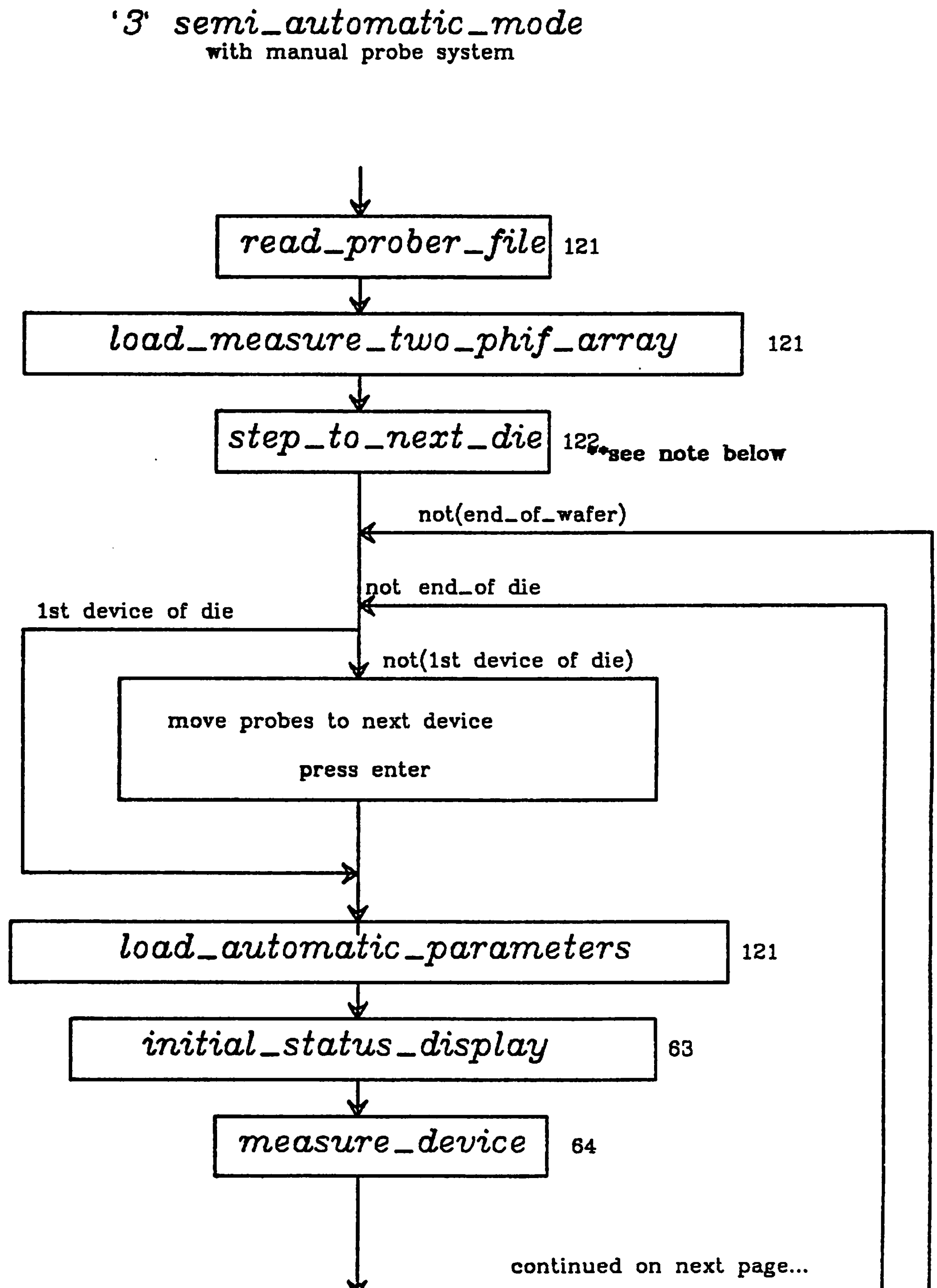

single_device_mode_inputs
SINGLE DEVICE OPERATION
standard_input_display
Process Name=? >
Lot=? >
Wafer=? >
XPOSITION=? >
YPOSITON=? >
Date=? >
Ooerator=? >
Output File=? >
VDD(volts)=? >
TEMPERATURE(deg. C)=? >
TOX(angstroms)=? >
PHIF2 or NSUB=? >
drawn width (microns)=? >
drawn length (microns)=? >
Device type=? > [1] enhancement, [2] zero-threshold, [3] depletion
SMU connected to DRAIN=? >
SMU connected to GATE=? >
SMU connected to SOURCE=? >
SMU connected to BODY=? >
Hit a "C" for changes or any other key to start. >
input_standard_values
96
input more amswers



continued on next page...

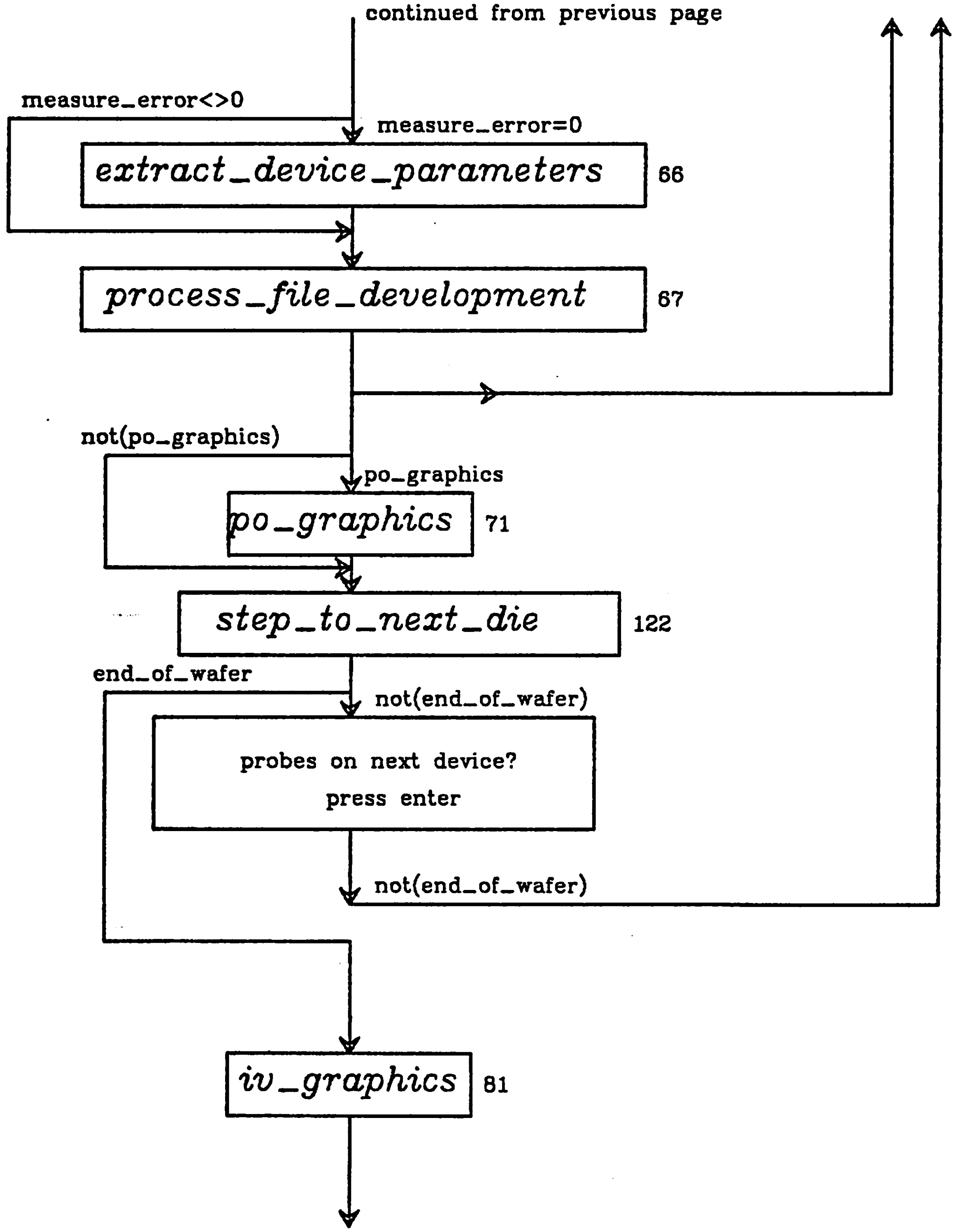
'1' automatic_mode
'2' semi_automatic_mode
with automatic prober system



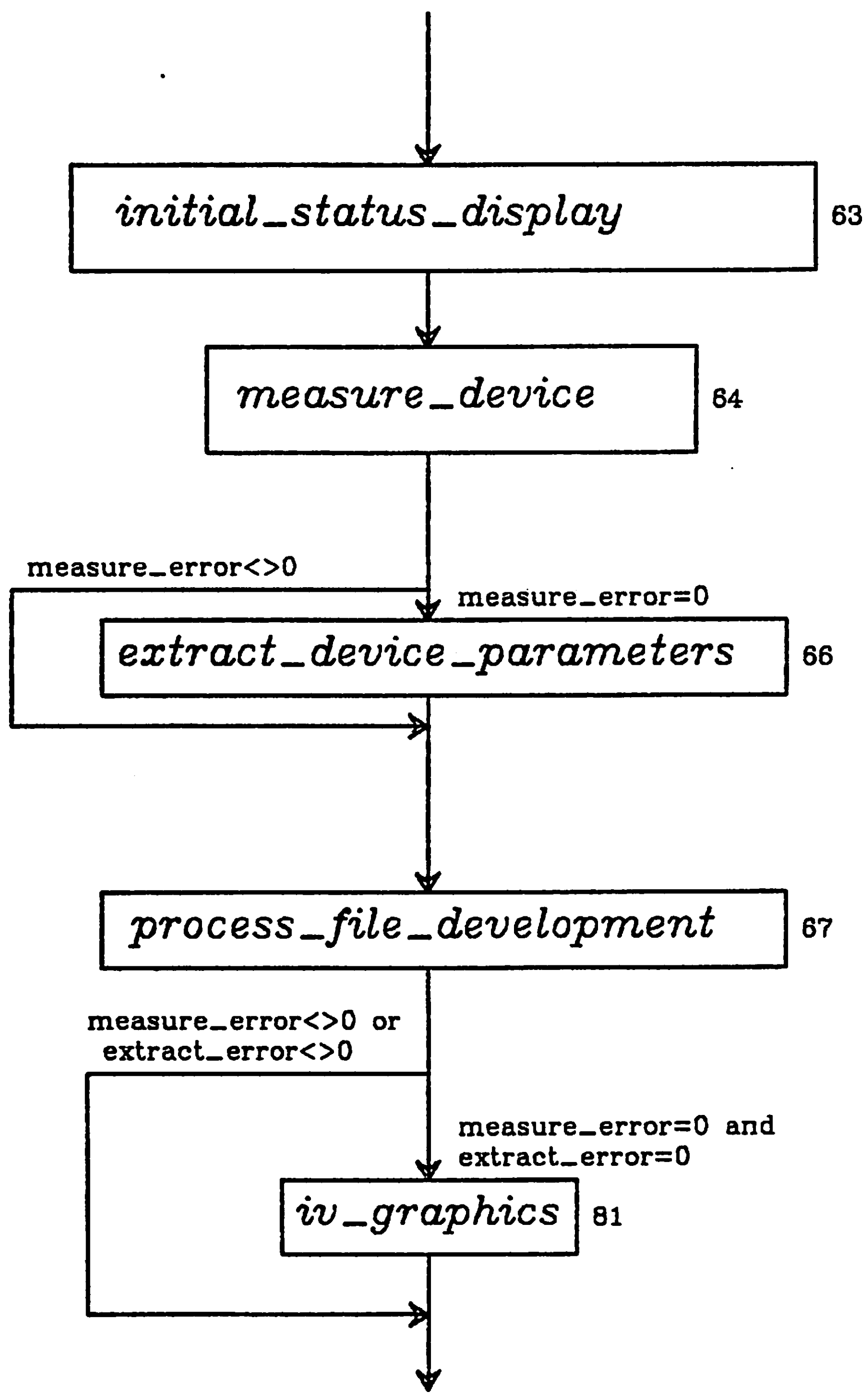


**athough stepping does not actually occur with a manual probe station, bookkeeping of location is performed here.

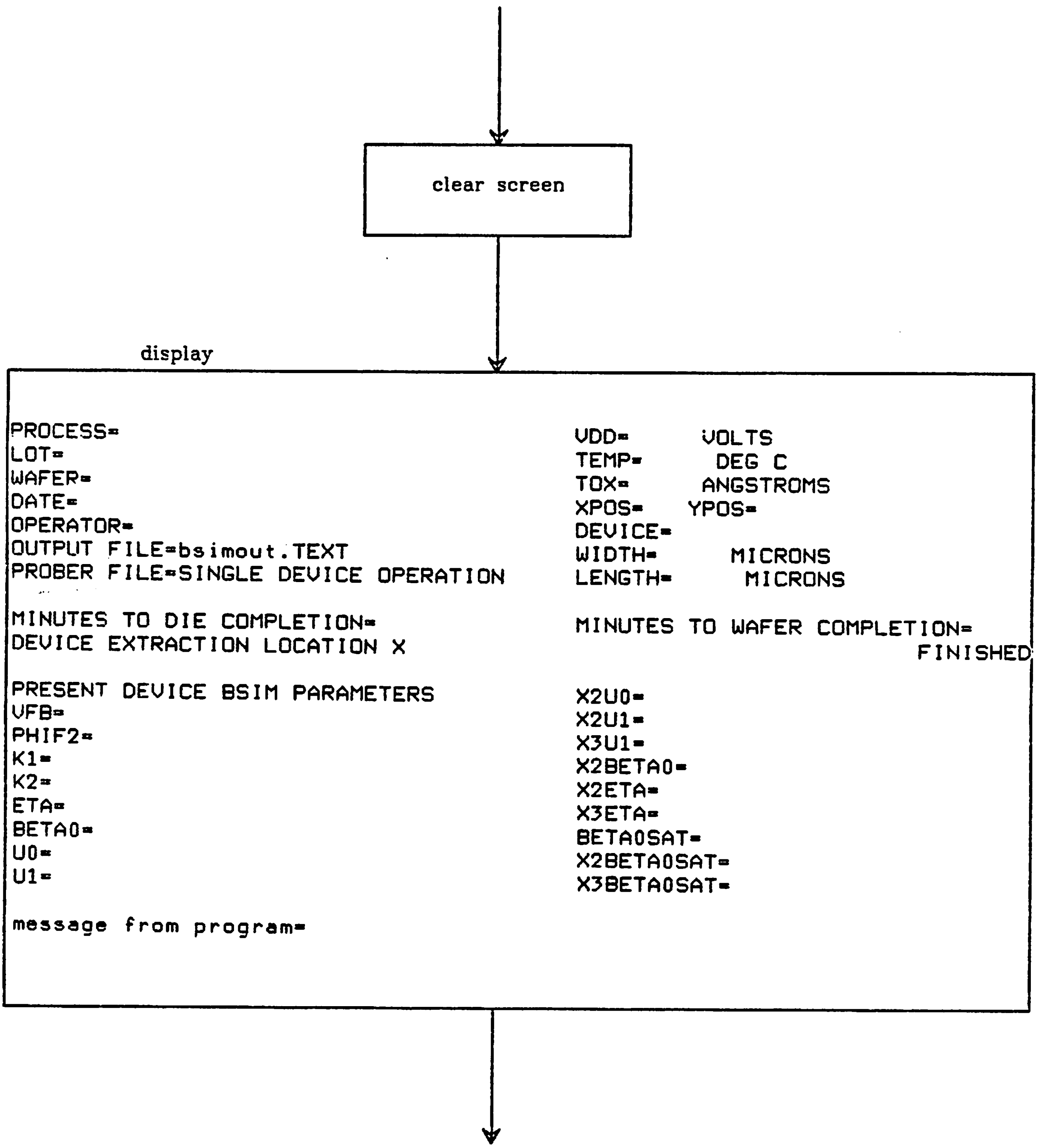
'3' semi_automatic_mode
with manual prober system



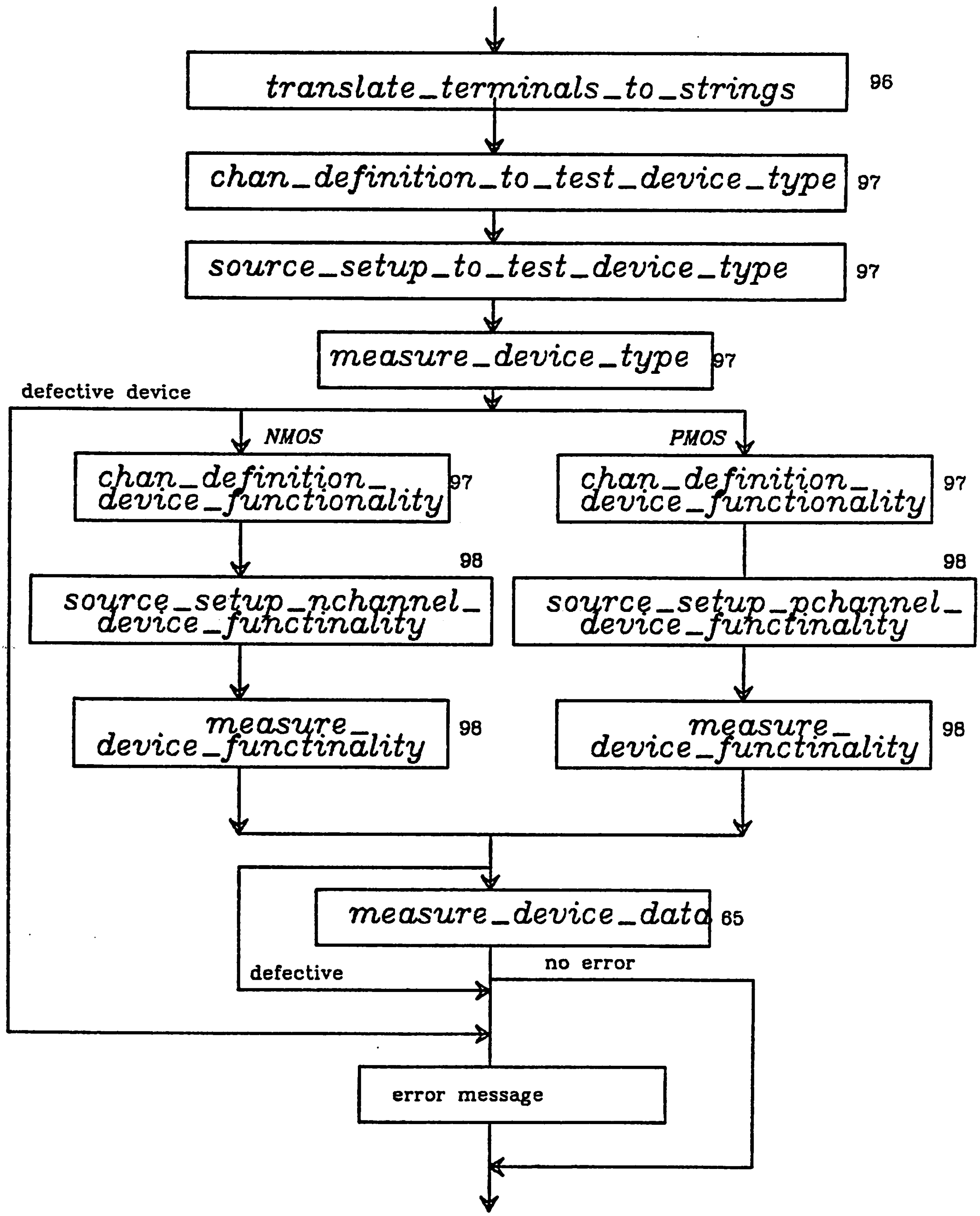
'4' *single_device_mode*

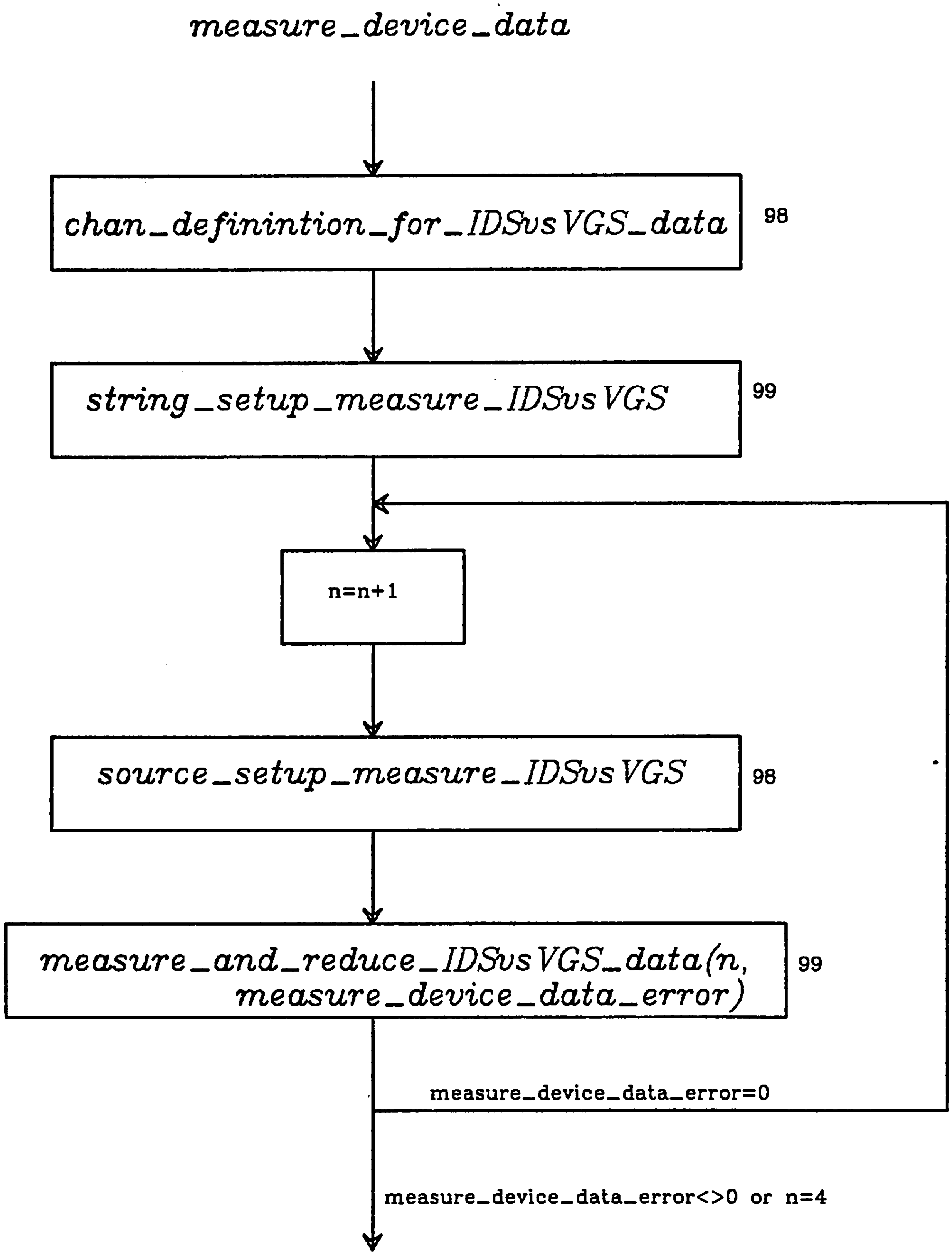


initial_status_display

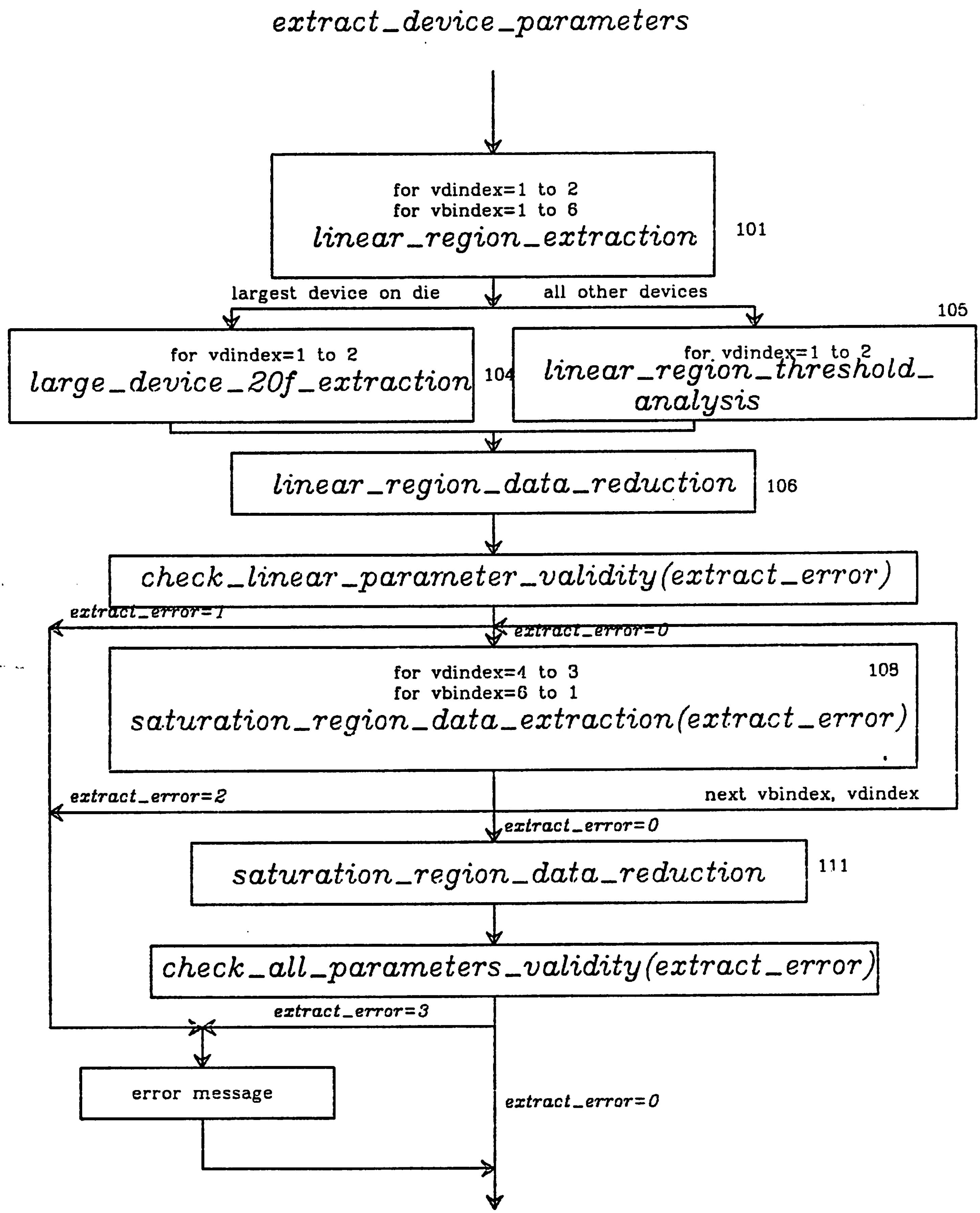


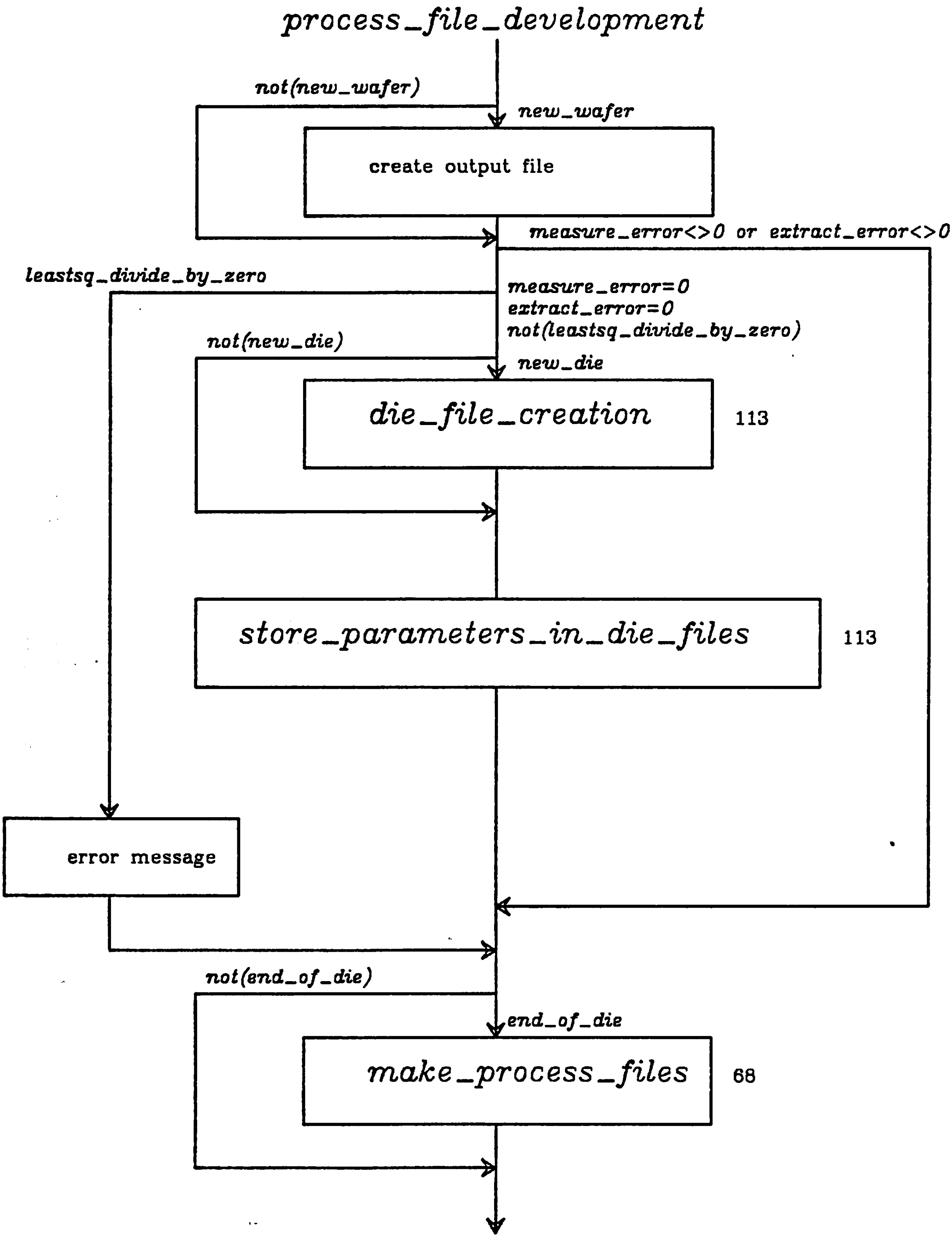
measure_device

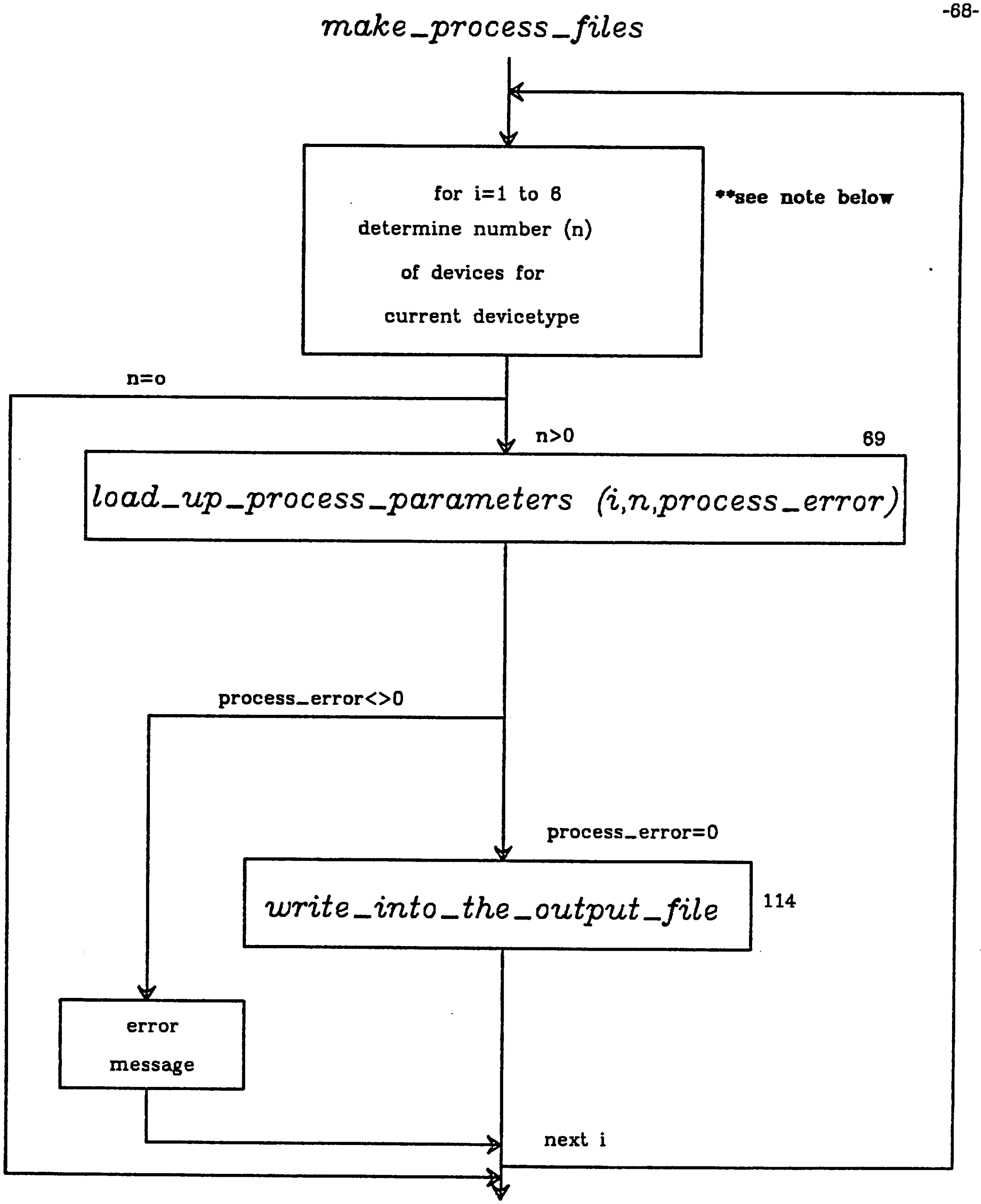




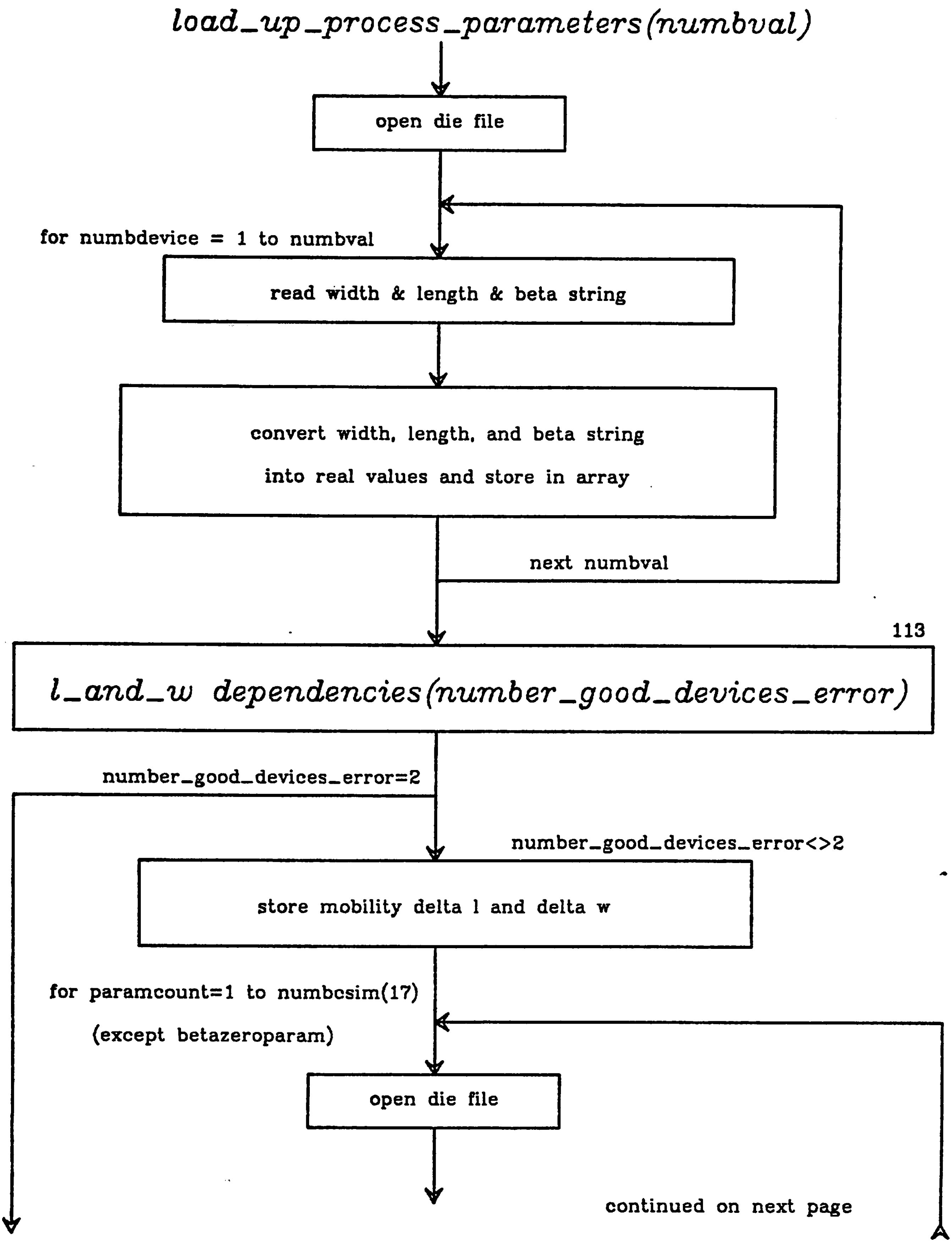
measure_device_data
chan_defininition_for_IDSvsVGS_data
98
string_setup_measure_IDSvsVGS
99
n=n+1
source_setup_measure_IDSvsVGS
98
measure_and_reduce_IDSvsVGS_data(n, measure_device_data_error)
99
measure_device_data_error=0
measure_device_data_error<>0 or n=4


extract_device_parameters
for vdindex=1 to 2
for vbindex=1 to 6
linear_region_extraction
101
largest device on die
all other devices
for vdindex=1 to 2
large_device_20f_extraction
104
105
for vdindex=1 to 2
linear_region_threshold_analysis
linear_region_data_reduction
106
check_linear_parameter_validity(extract_error)
extract_error=1
extract_error=0
for vdindex=4 to 3
for vbindex=6 to 1
saturation_region_data_extraction(extract_error)
108
extract_error=2
next vbindex, vdindex
extract_error=0
saturation_region_data_reduction
111
check_all_parameters_validity(extract_error)
extract_error=3
error message
extract_error=0

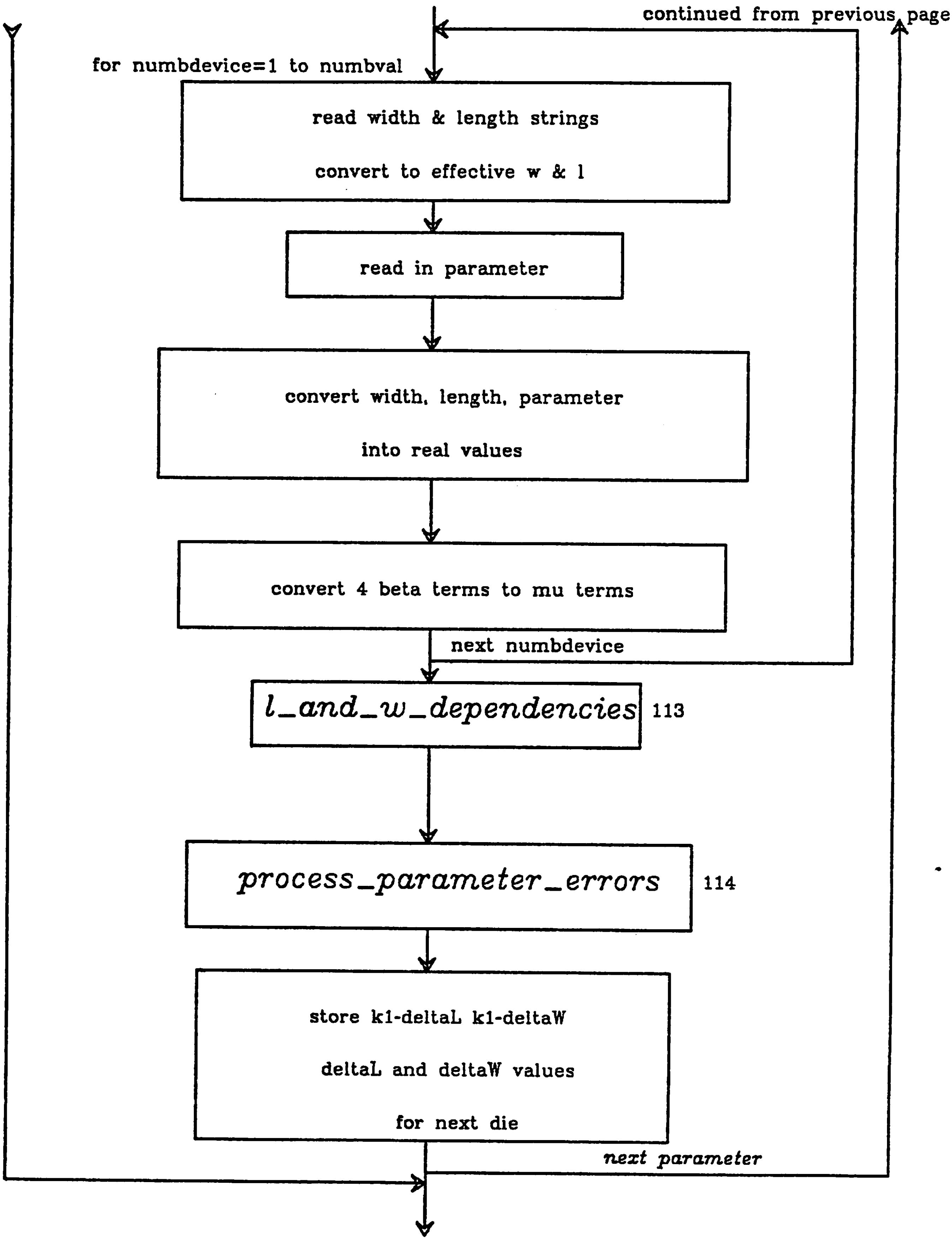

process_file_development
not(new_wafer)
new_wafer
create output file
measure_error<>0 or extract_error<>0
leastsq_divide_by_zero
measure_error=0
extract_error=0
not(leastsq_divide_by_zero)
not(new_die)
new_die
die_file_creation
113
store_parameters_in_die_files
113
error message
not(end_of_die)
end_of_die
make_process_files
68



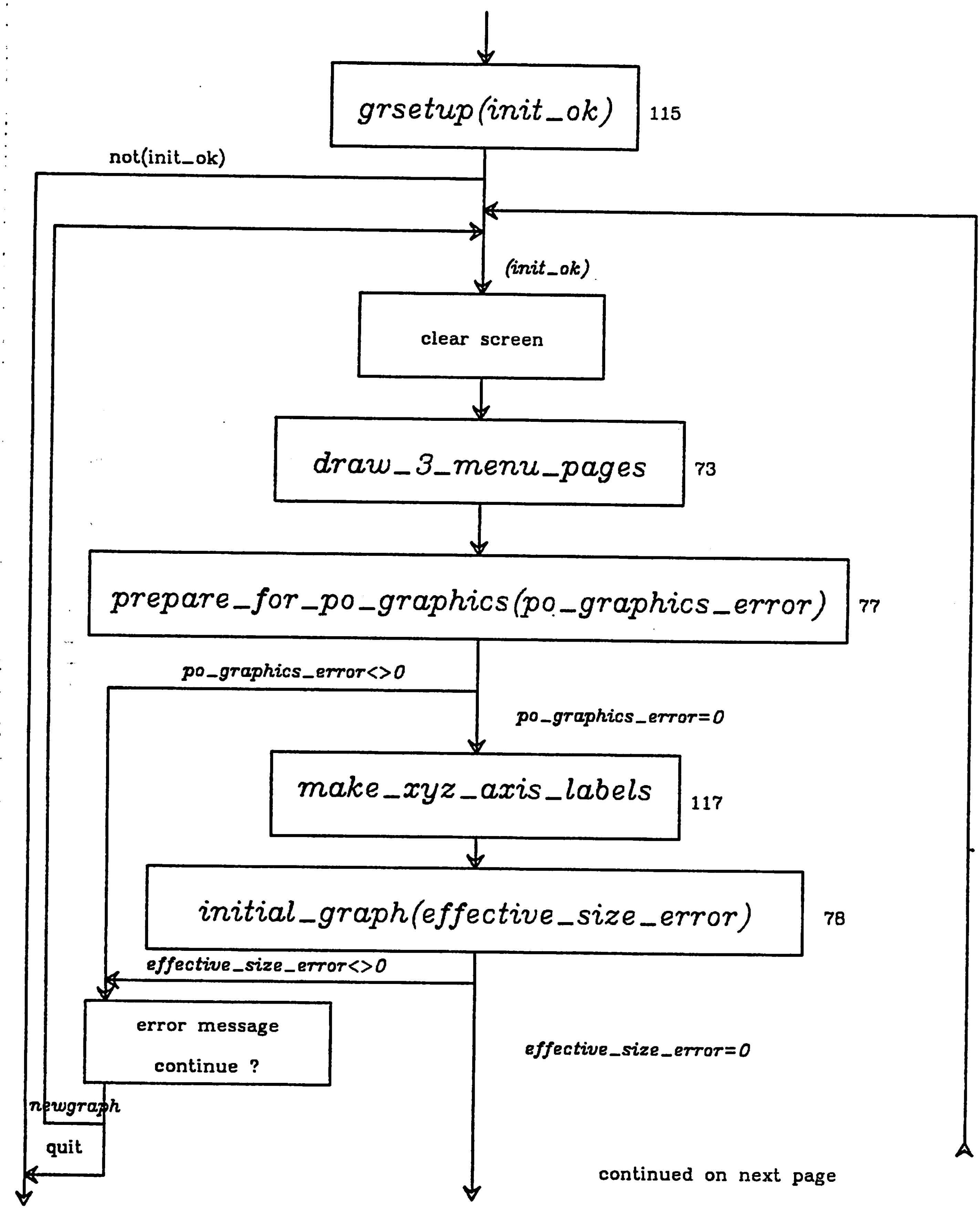
**there are six possible devicetypes: NMOS enhancement, NMOS depletion, NMOS zero-threshold, PMOS enhancement, PMOS depletion, and PMOS zero-threshold.


load_up_process_parameters(numbval)
open die file
for numbdevice = 1 to numbval
read width & length & beta string
convert width, length, and beta string
into real values and store in array
next numbval
113
l_and_w dependencies(number_good_devices_error)
number_good_devices_error=2
number_good_devices_error<>2
store mobility delta l and delta w
for paramcount=1 to numbcsim(17)
(except betazeroparam)
open die file
continued on next page

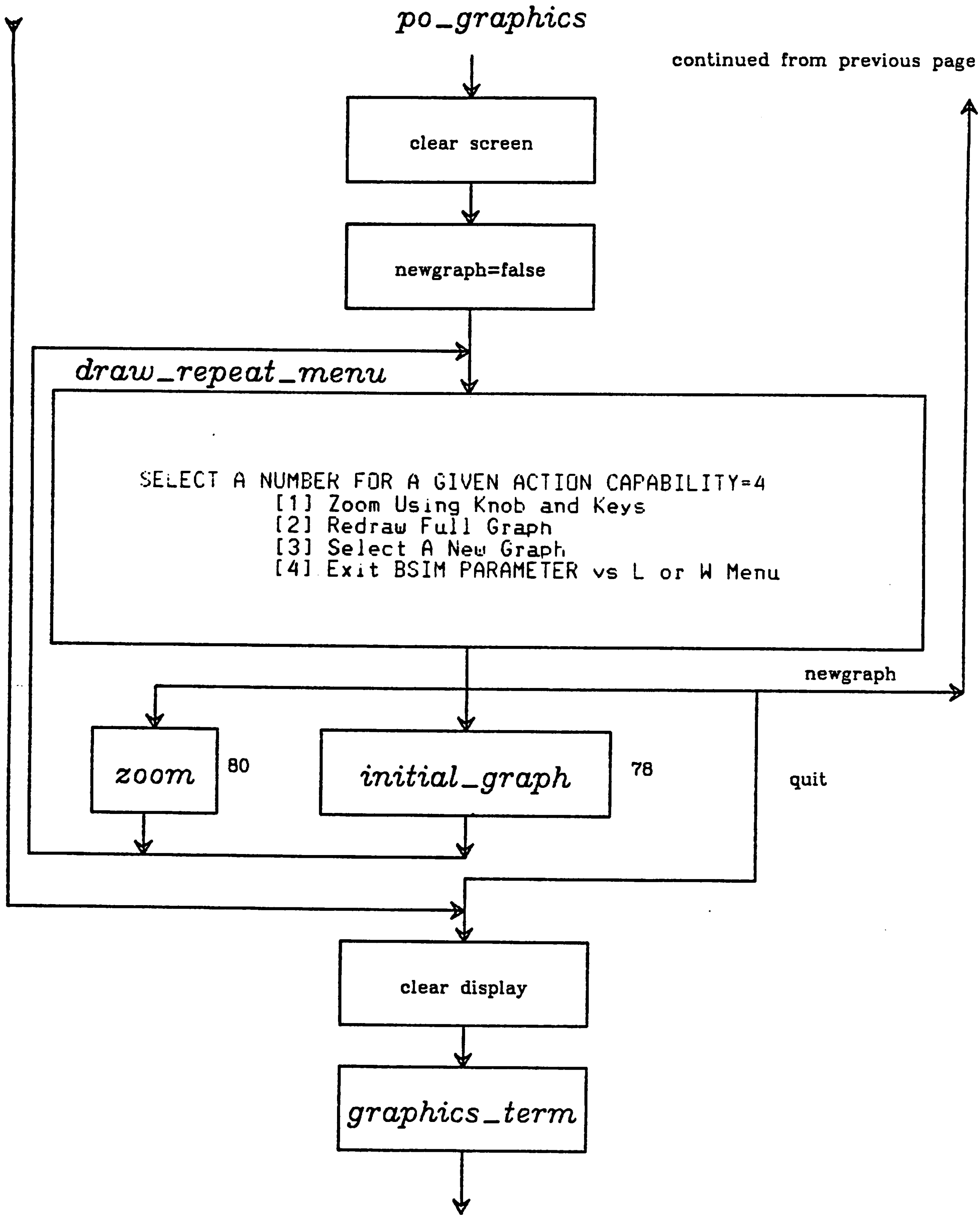
load_up_process_parameters(numbval)

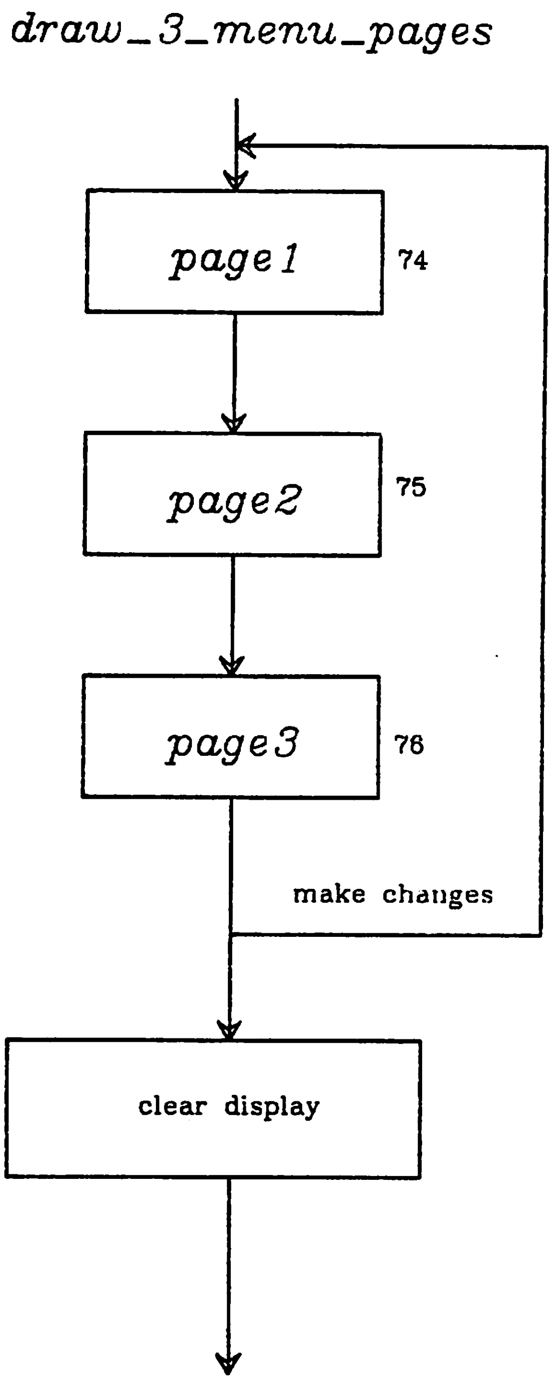


po_graphics

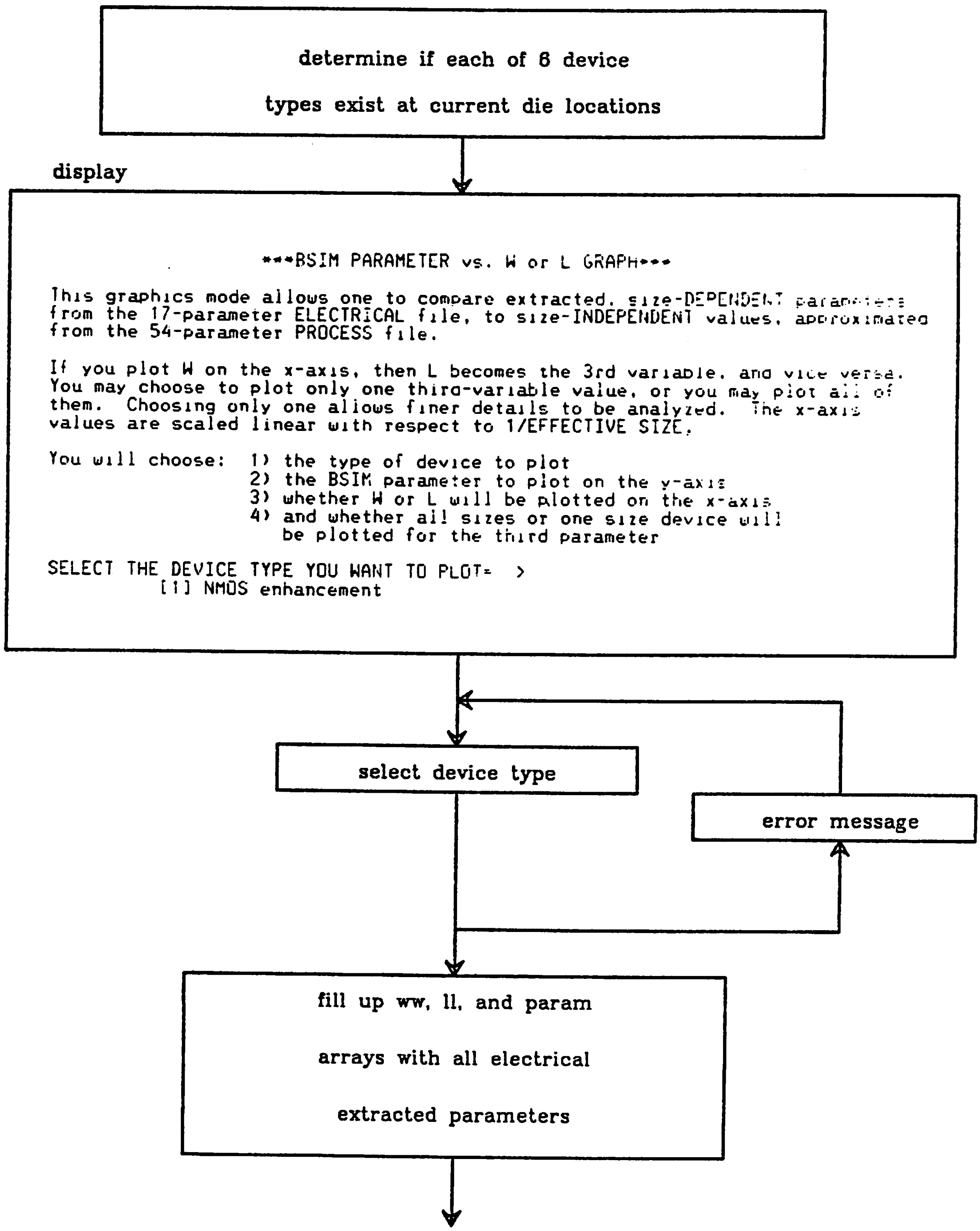


continued on next page

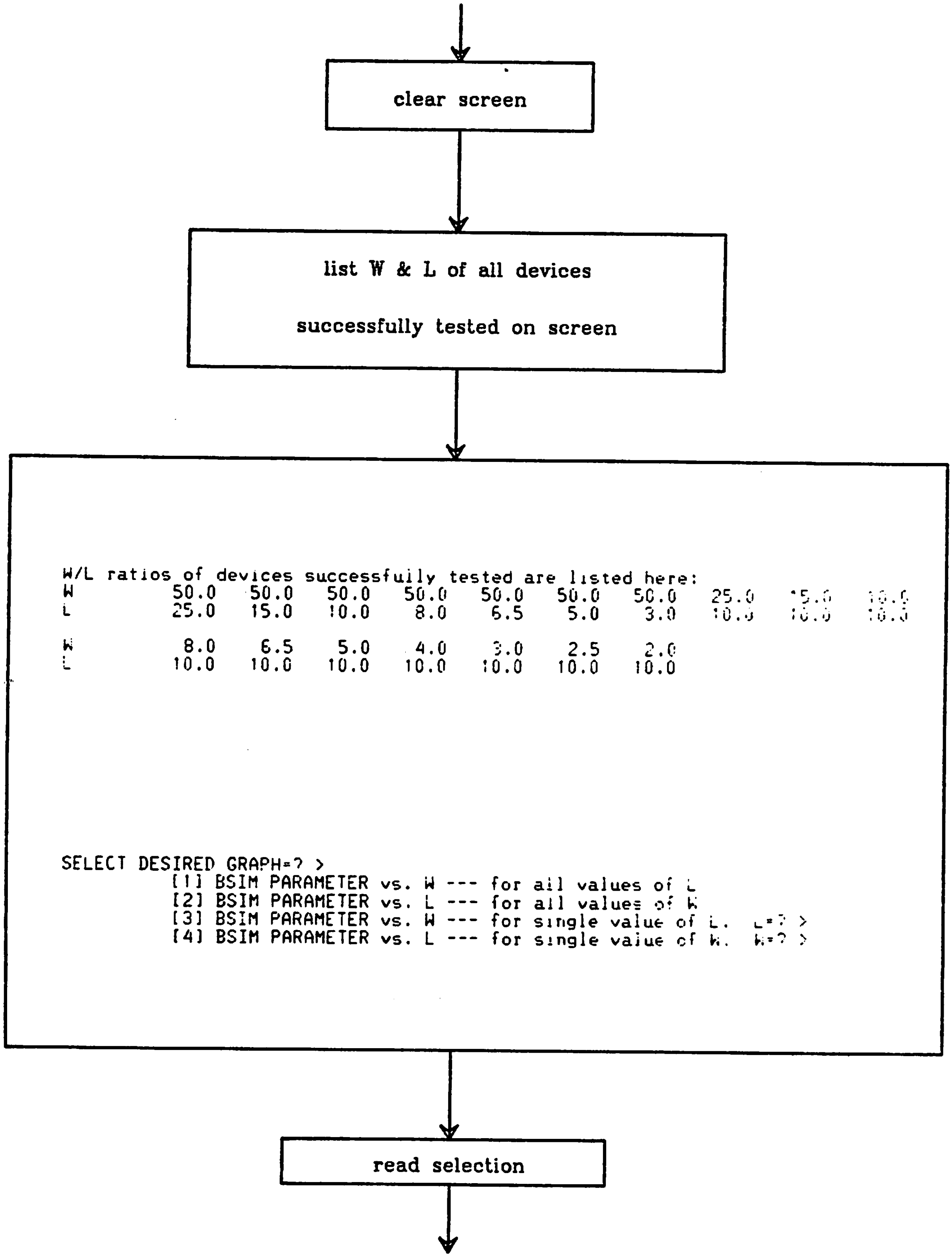

po_graphics
continued from previous page
clear screen
newgraph=false
draw_repeat_menu
SELECT A NUMBER FOR A GIVEN ACTION CAPABILITY=4
[1] Zoom Using Knob and Keys
[2] Redraw Full Graph
[3] Select A New Graph
[4] Exit BSIM PARAMETER vs L or W Menu
newgraph
zoom
80
initial_graph
78
quit
clear display
graphics_term


draw_3_menu_pages
page1
74
page2
75
page3
76
make changes
clear display

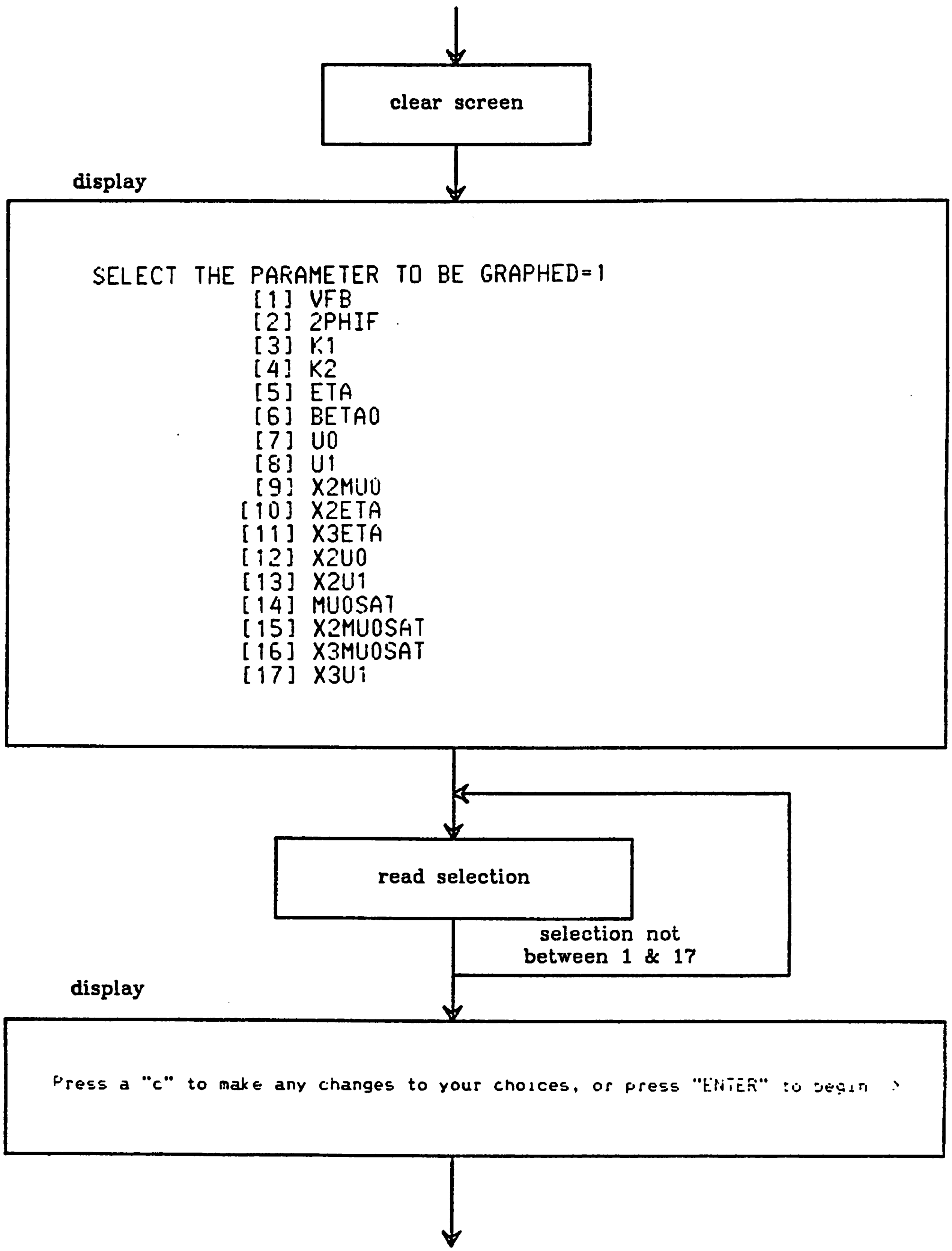
page1

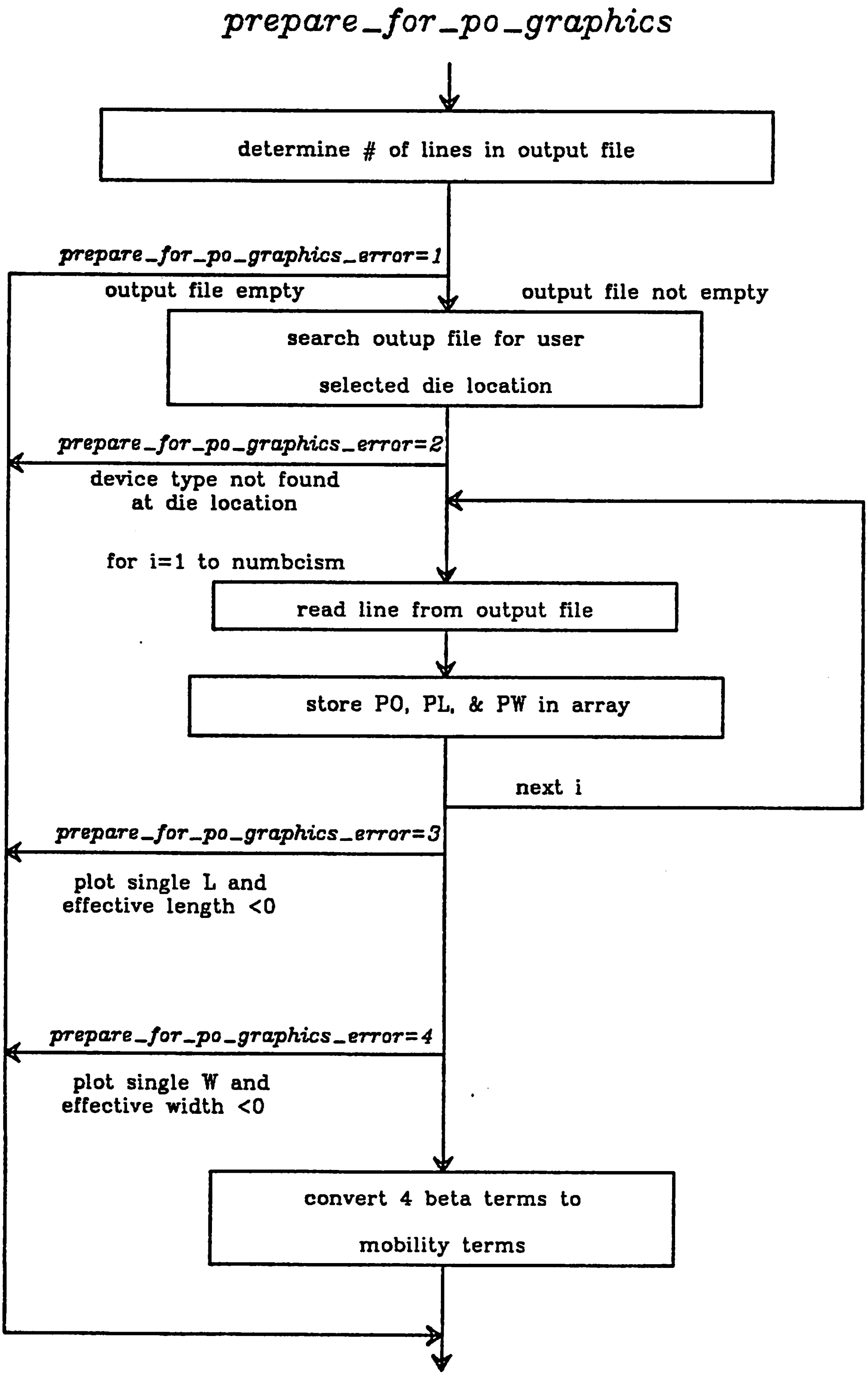

determine if each of 6 device
types exist at current die locations
display
BSIM PARAMETER vs. W or L GRAPH
This graphics mode allows one to compare extracted, size-DEPENDENT parameters
from the 17-parameter ELECTRICAL file, to size-INDEPENDENT values, approximated
from the 54-parameter PROCESS file.
If you plot W on the x-axis, then L becomes the 3rd variable, and vice versa.
You may choose to plot only one third-variable value, or you may plot all of
them. Choosing only one allows finer details to be analyzed. The x-axis
values are scaled linear with respect to 1/EFFECTIVE SIZE.
You will choose: 1) the type of device to plot
2) the BSIM parameter to plot on the y-axis
3) whether W or L will be plotted on the x-axis
4) and whether all sizes or one size device will
be plotted for the third parameter
SELECT THE DEVICE TYPE YOU WANT TO PLOT= >
[1] NMOS enhancement
select device type
error message
fill up ww, ll, and param
arrays with all electrical
extracted parameters

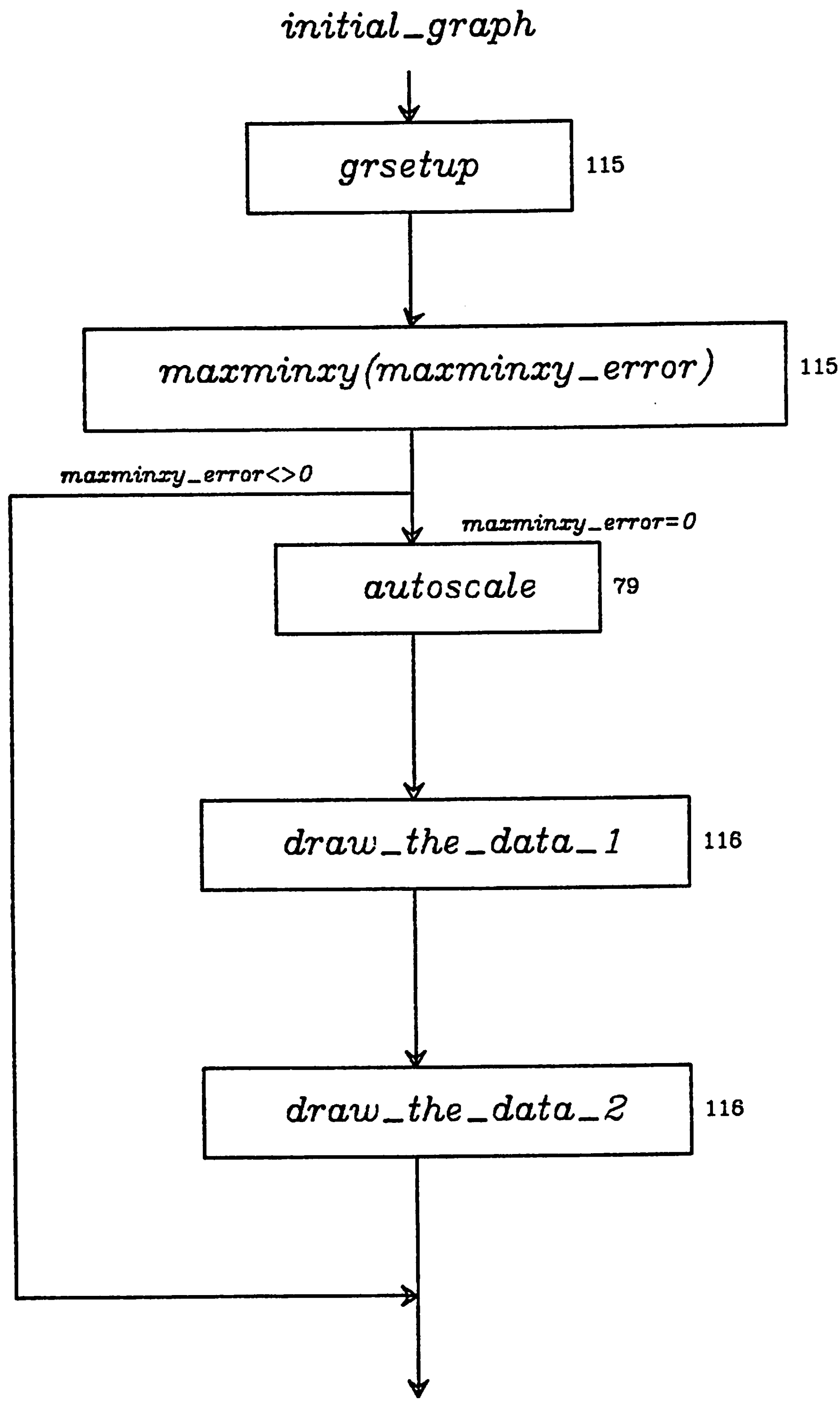
page2

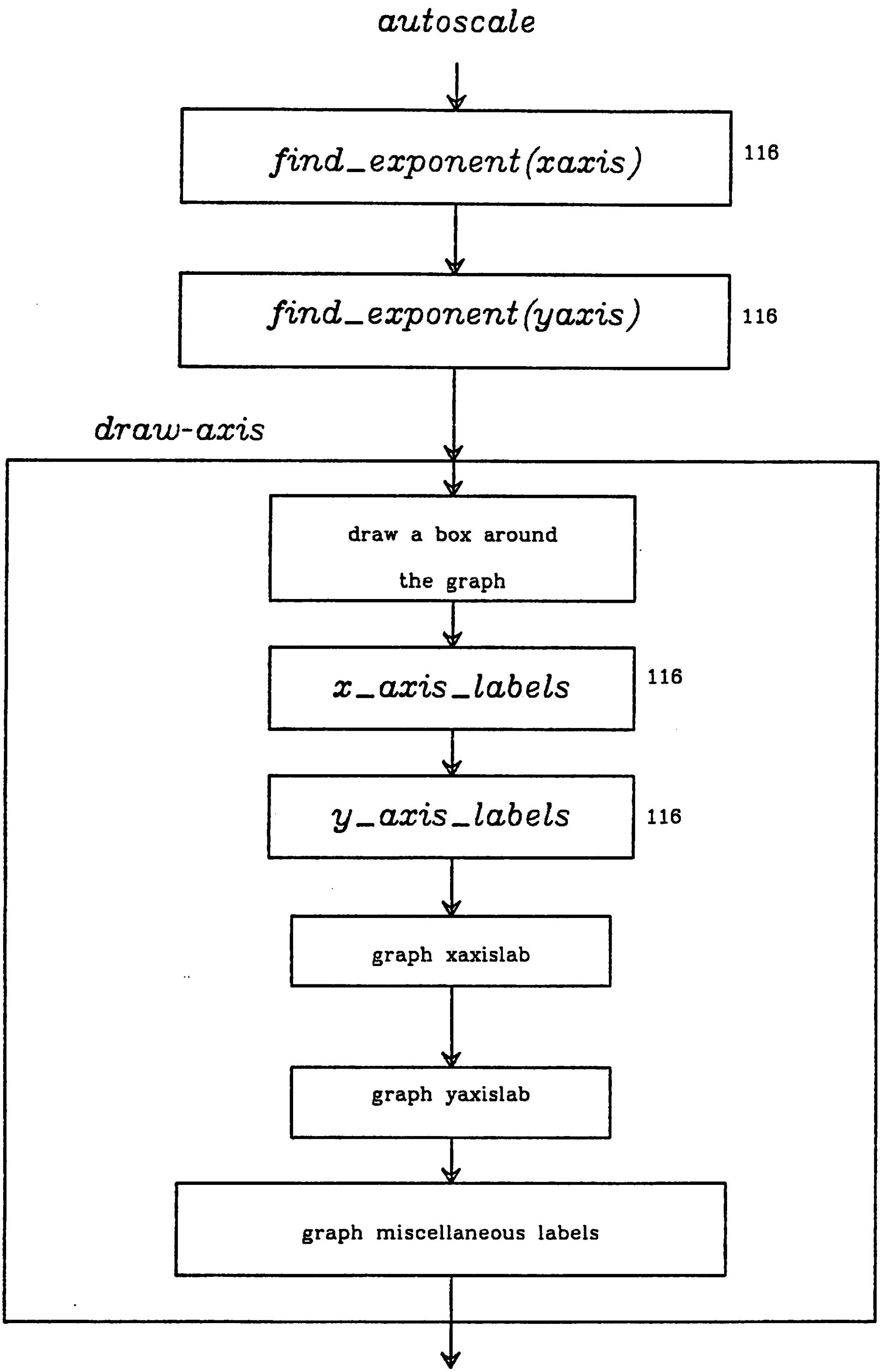

clear screen
list W & L of all devices
successfully tested on screen
W/L ratios of devices successfully tested are listed here:
W 50.0 50.0 50.0 50.0 50.0 50.0 50.0 25.0 15.0 10.0
L 25.0 15.0 10.0 8.0 6.5 5.0 3.0 10.0 10.0 10.0
W 8.0 6.5 5.0 4.0 3.0 2.5 2.0
L 10.0 10.0 10.0 10.0 10.0 10.0 10.0
SELECT DESIRED GRAPH=? >
[1] BSIM PARAMETER vs. W --- for all values of L
[2] BSIM PARAMETER vs. L --- for all values of W
[3] BSIM PARAMETER vs. W --- for single value of L. L=? >
[4] BSIM PARAMETER vs. L --- for single value of W. W=? >
read selection

page3

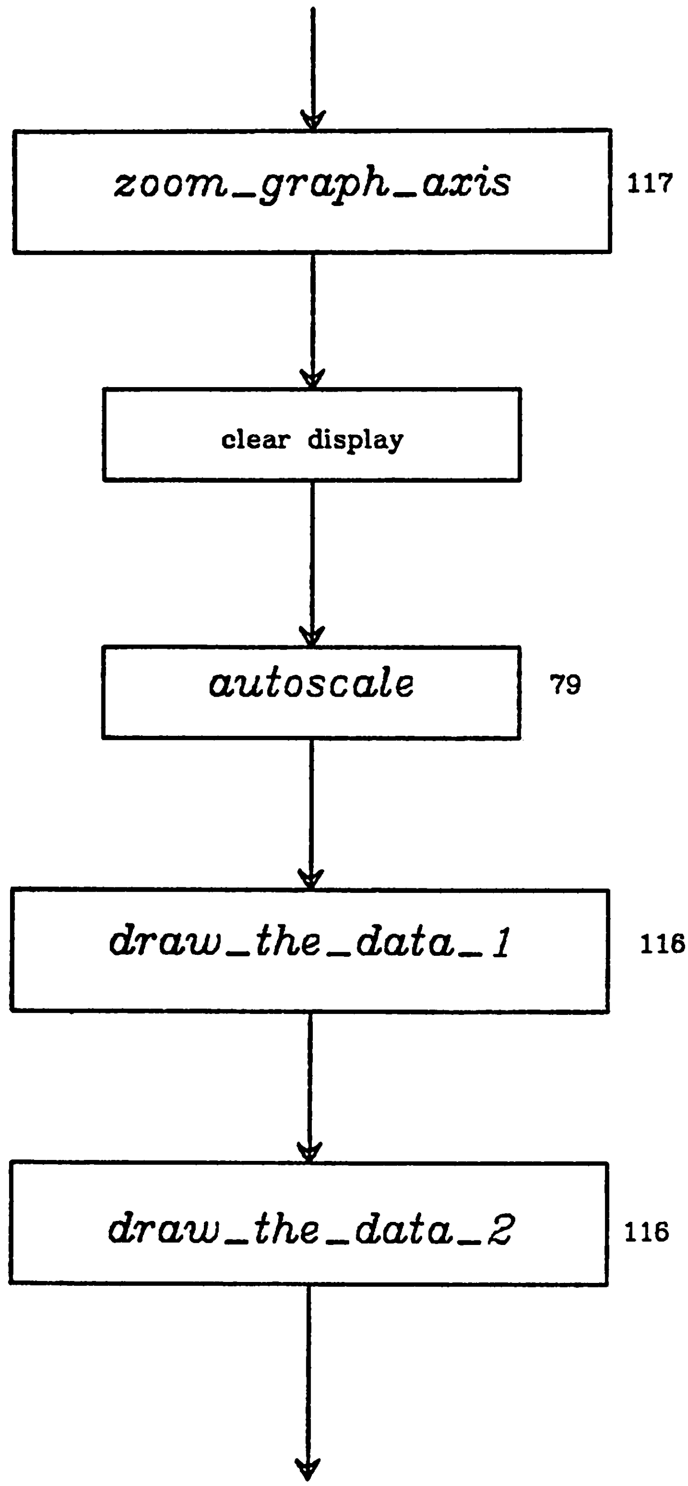

clear screen
display
SELECT THE PARAMETER TO BE GRAPHED=1
[1] VFB
[2] 2PHIF
[3] K1
[4] K2
[5] ETA
[6] BETA0
[7] U0
[8] U1
[9] X2MU0
[10] X2ETA
[11] X3ETA
[12] X2U0
[13] X2U1
[14] MU0SAT
[15] X2MU0SAT
[16] X3MU0SAT
[17] X3U1
read selection
selection not between 1 & 17
display
Press a "c" to make any changes to your choices, or press "ENTER" to begin


prepare_for_po_graphics
determine # of lines in output file
prepare_for_po_graphics_error=1
output file empty
output file not empty
search outup file for user
selected die location
prepare_for_po_graphics_error=2
device type not found
at die location
for i=1 to numbcism
read line from output file
store PO, PL, & PW in array
next i
prepare_for_po_graphics_error=3
plot single L and
effective length <0
prepare_for_po_graphics_error=4
plot single W and
effective width <0
convert 4 beta terms to
mobility terms

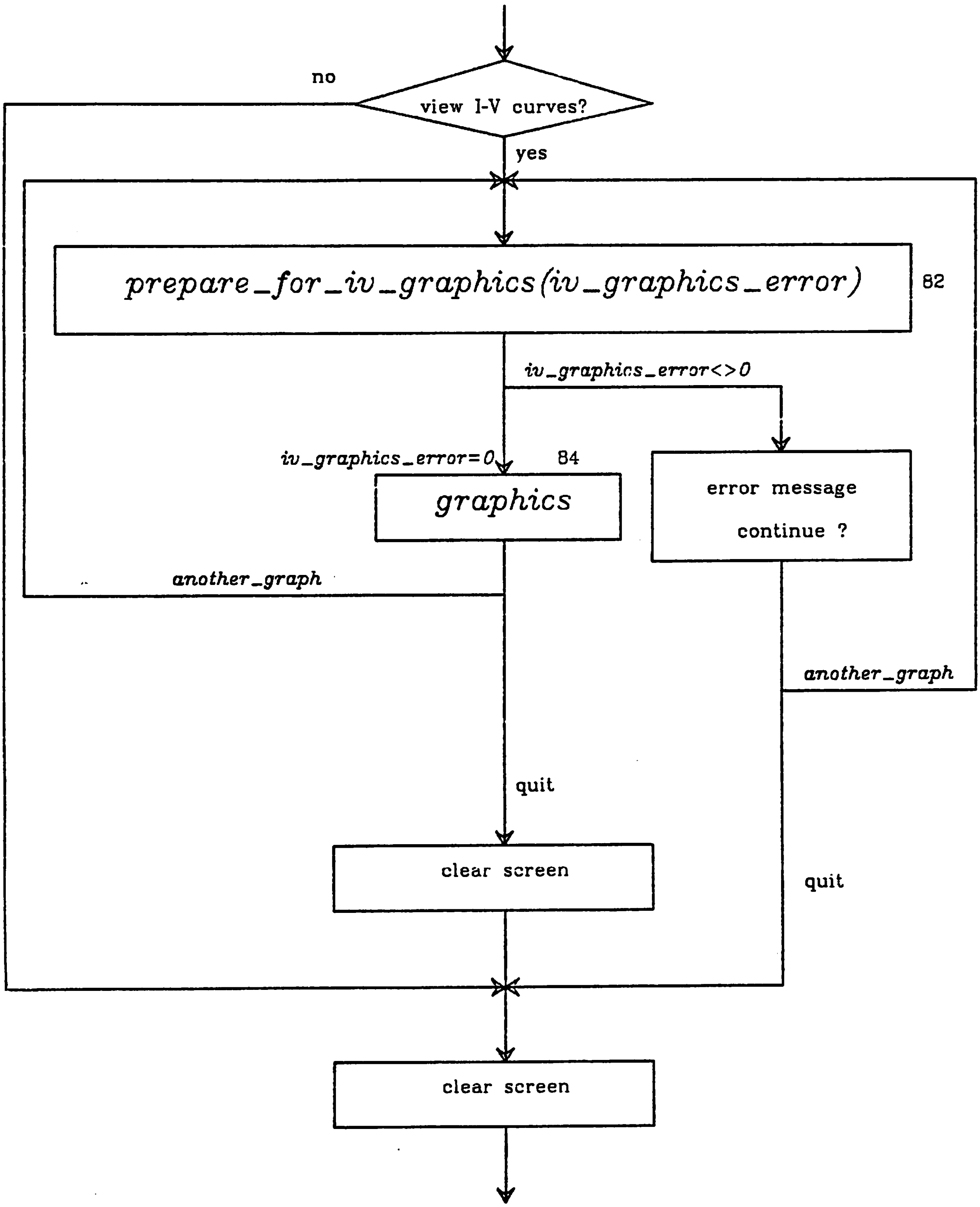

initial_graph
grsetup
115
maxminxy(maxminxy_error)
115
maxminxy_error<>0
maxminxy_error=0
autoscale
79
draw_the_data_1
116
draw_the_data_2
116


autoscale
find_exponent(xaxis)
116
find_exponent(yaxis)
116
draw-axis
draw a box around the graph
x_axis_labels
116
y_axis_labels
116
graph xaxislab
graph yaxislab
graph miscellaneous labels

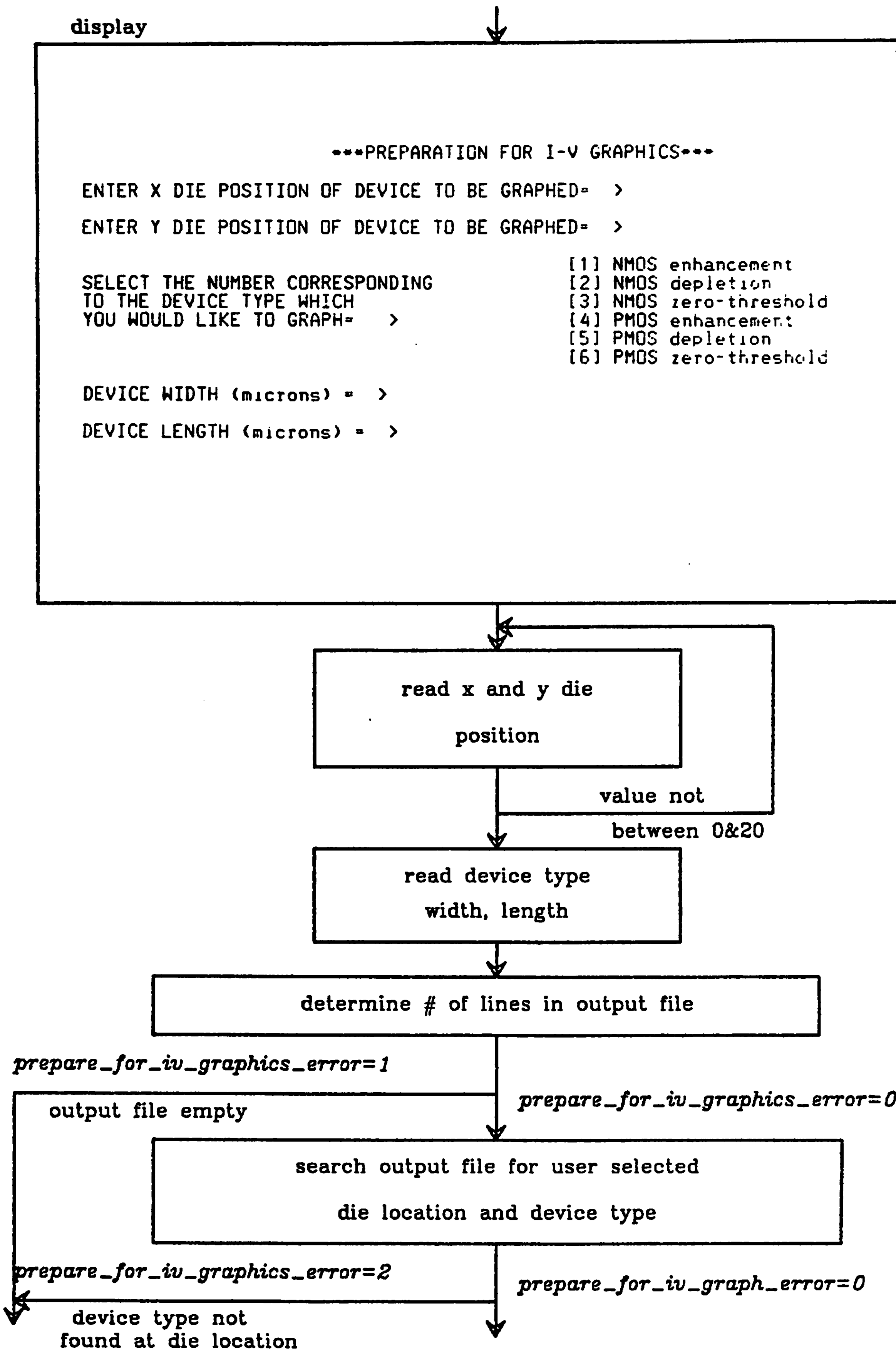
zoom


zoom_graph_axis
117
clear display
autoscale
79
draw_the_data_1
116
draw_the_data_2
116

iv_graphics

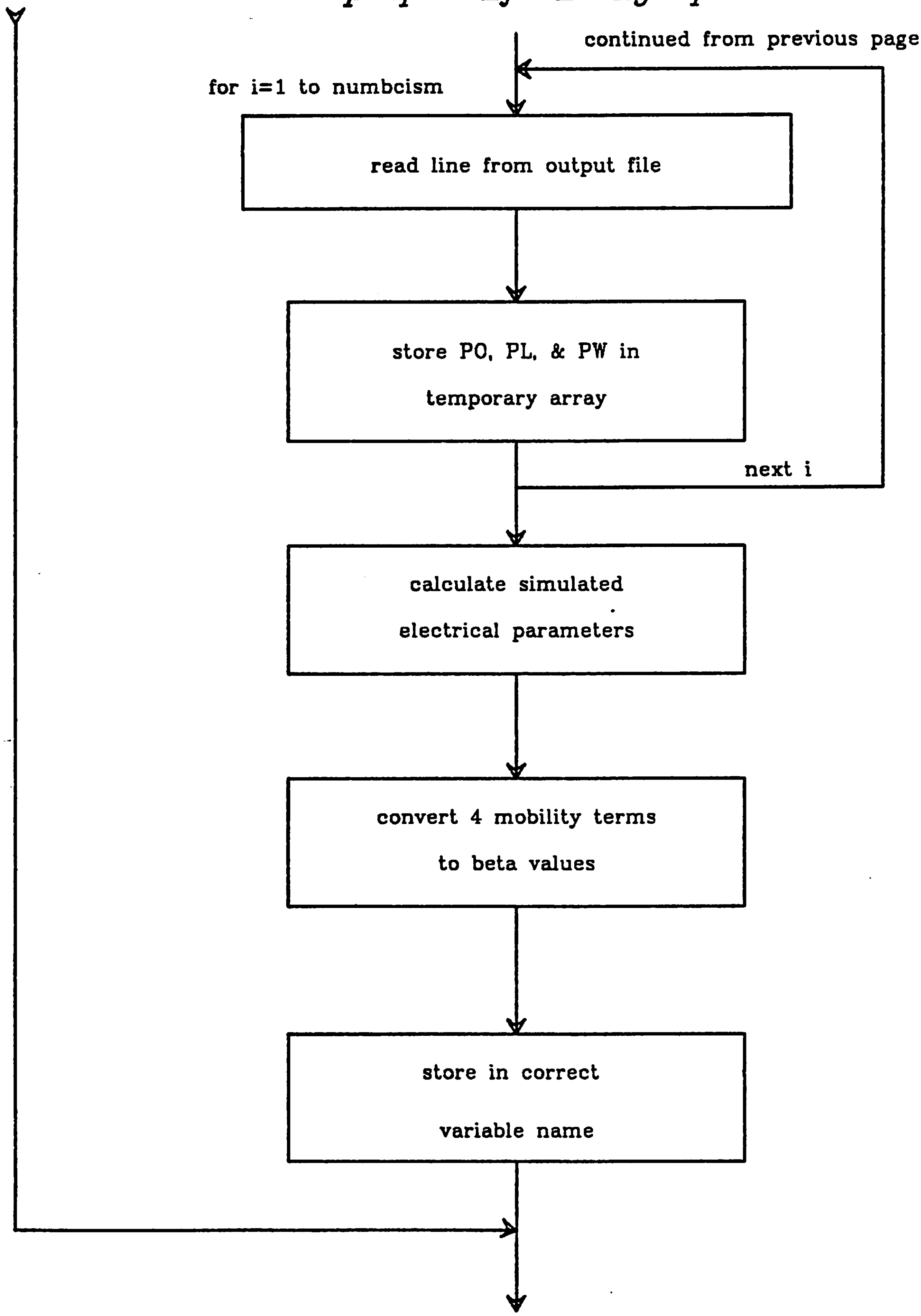

no
view I-V curves?
yes
prepare_for_iv_graphics(iv_graphics_error)
82
iv_graphics_error<>0
iv_graphics_error=0
84
graphics
error message
continue ?
another_graph
another_graph
quit
clear screen
quit
clear screen

prepare_for_iv_graphics

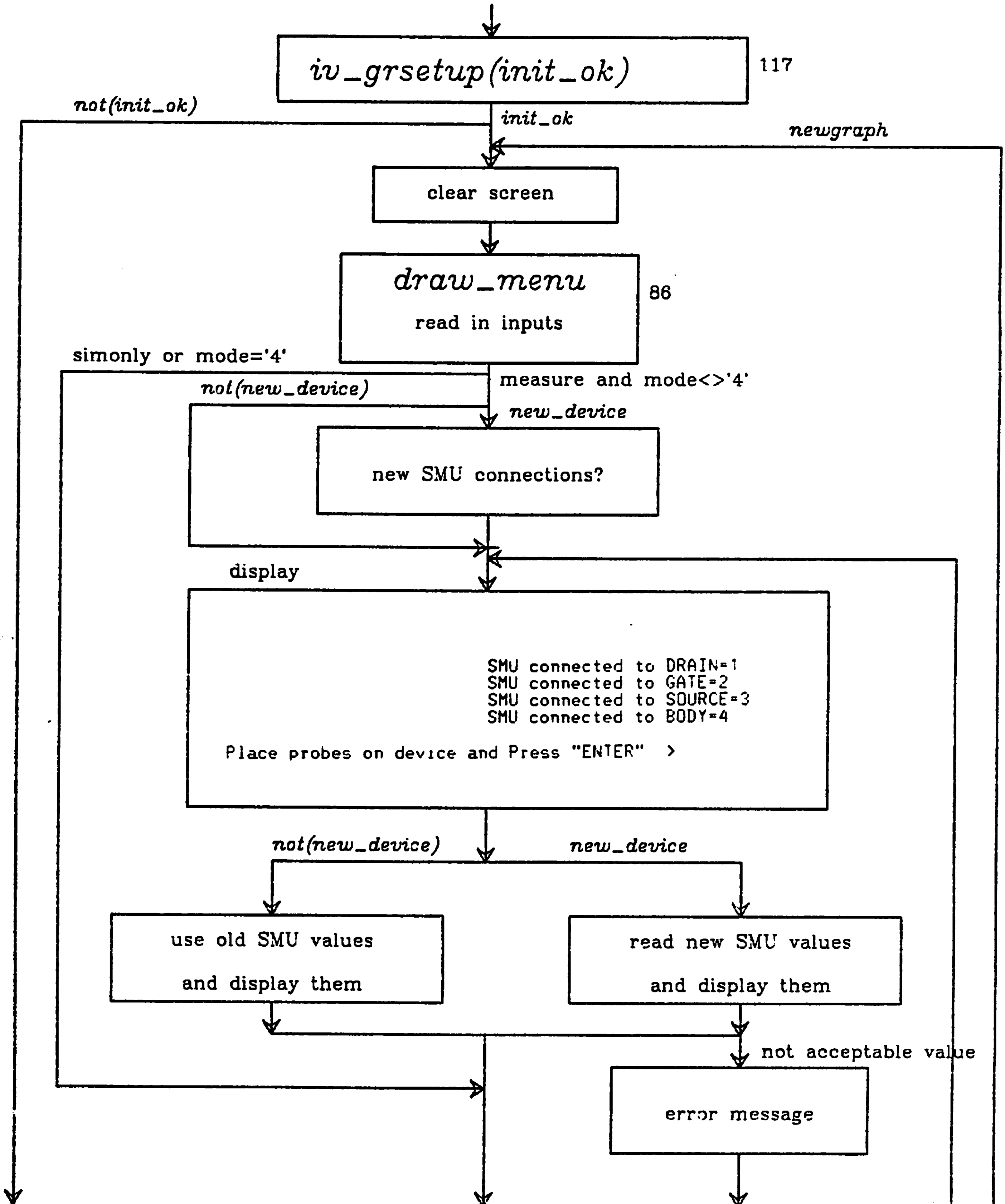


continued on next page

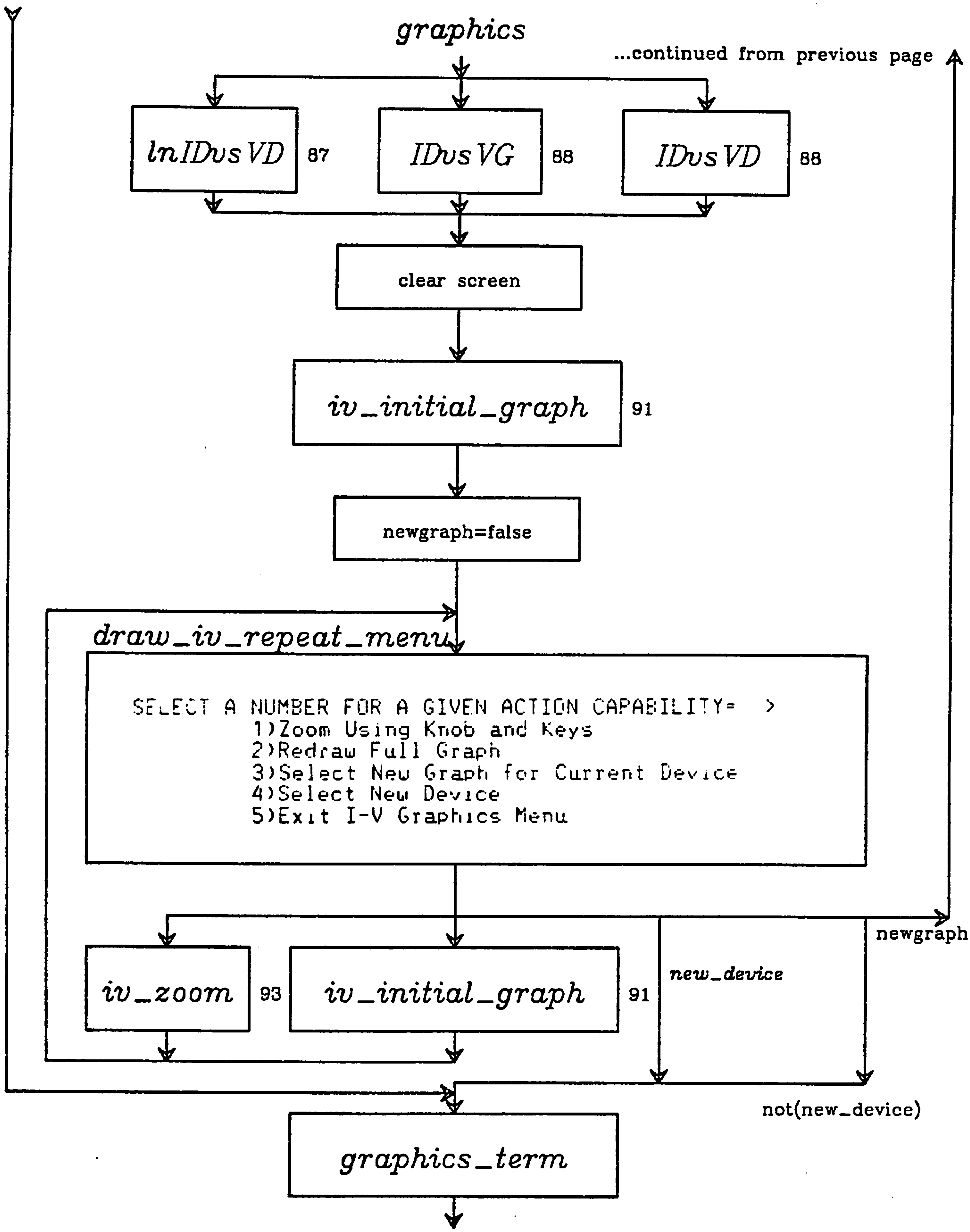
prepare_for_iv_graphics



graphics



continued on next page ...

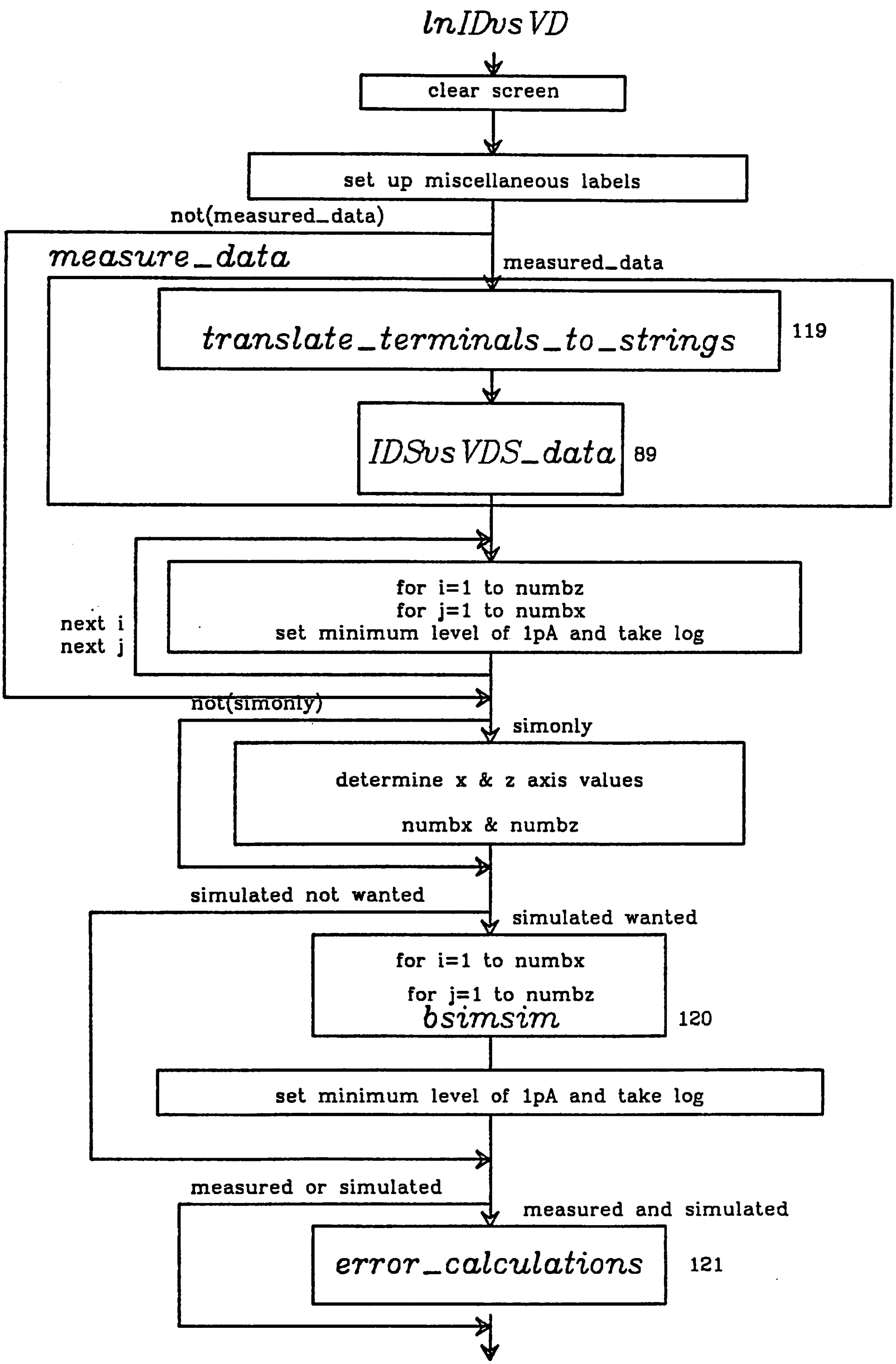

graphics
...continued from previous page
lnIDvsVD
87
IDvsVG
88
IDvsVD
88
clear screen
iv_initial_graph
91
newgraph=false
draw_iv_repeat_menu
SELECT A NUMBER FOR A GIVEN ACTION CAPABILITY= >
1)Zoom Using Knob and Keys
2)Redraw Full Graph
3)Select New Graph for Current Device
4)Select New Device
5)Exit I-V Graphics Menu
iv_zoom
93
iv_initial_graph
91
new_device
newgraph
not(new_device)
graphics_term

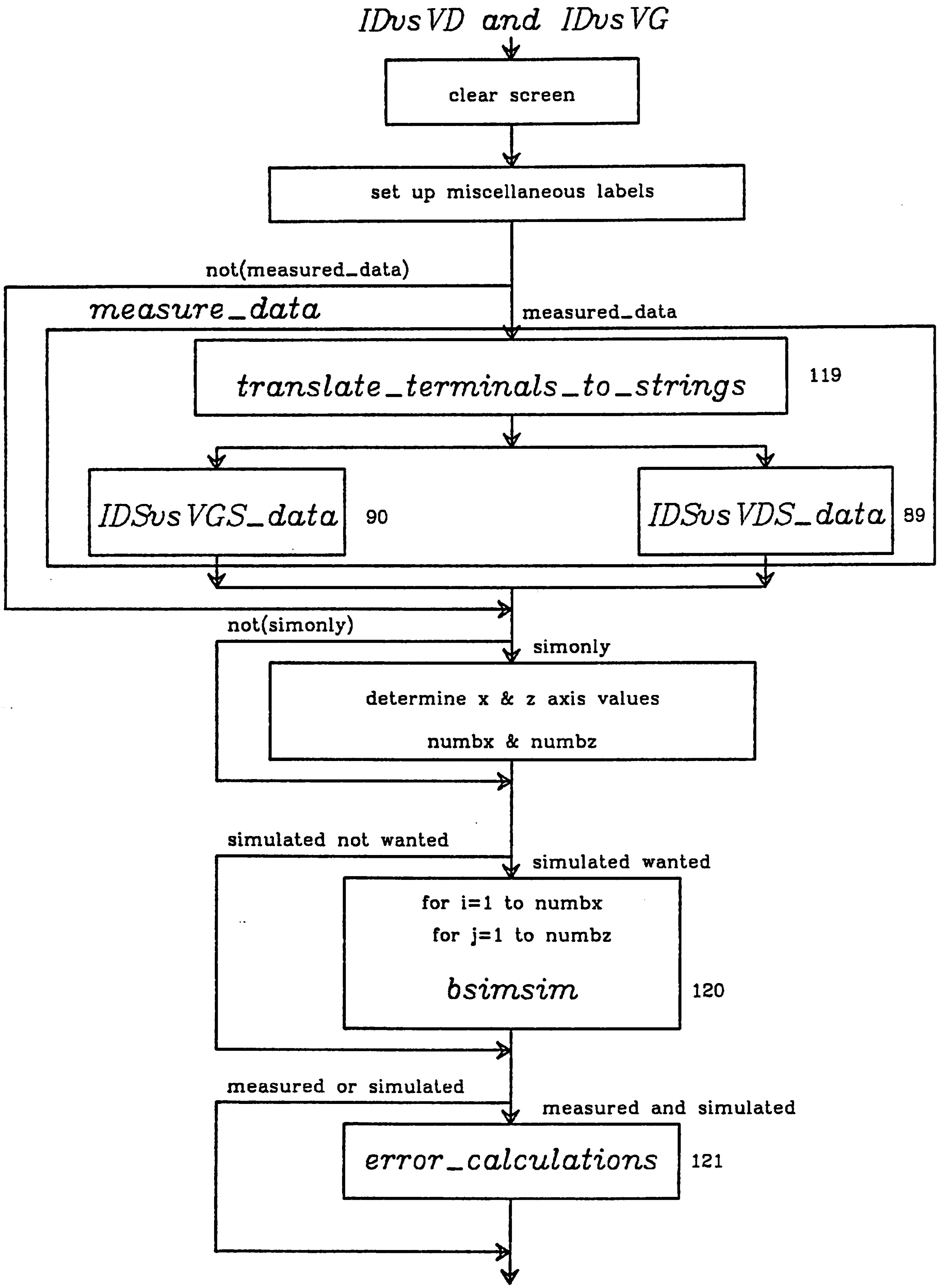
draw_menu

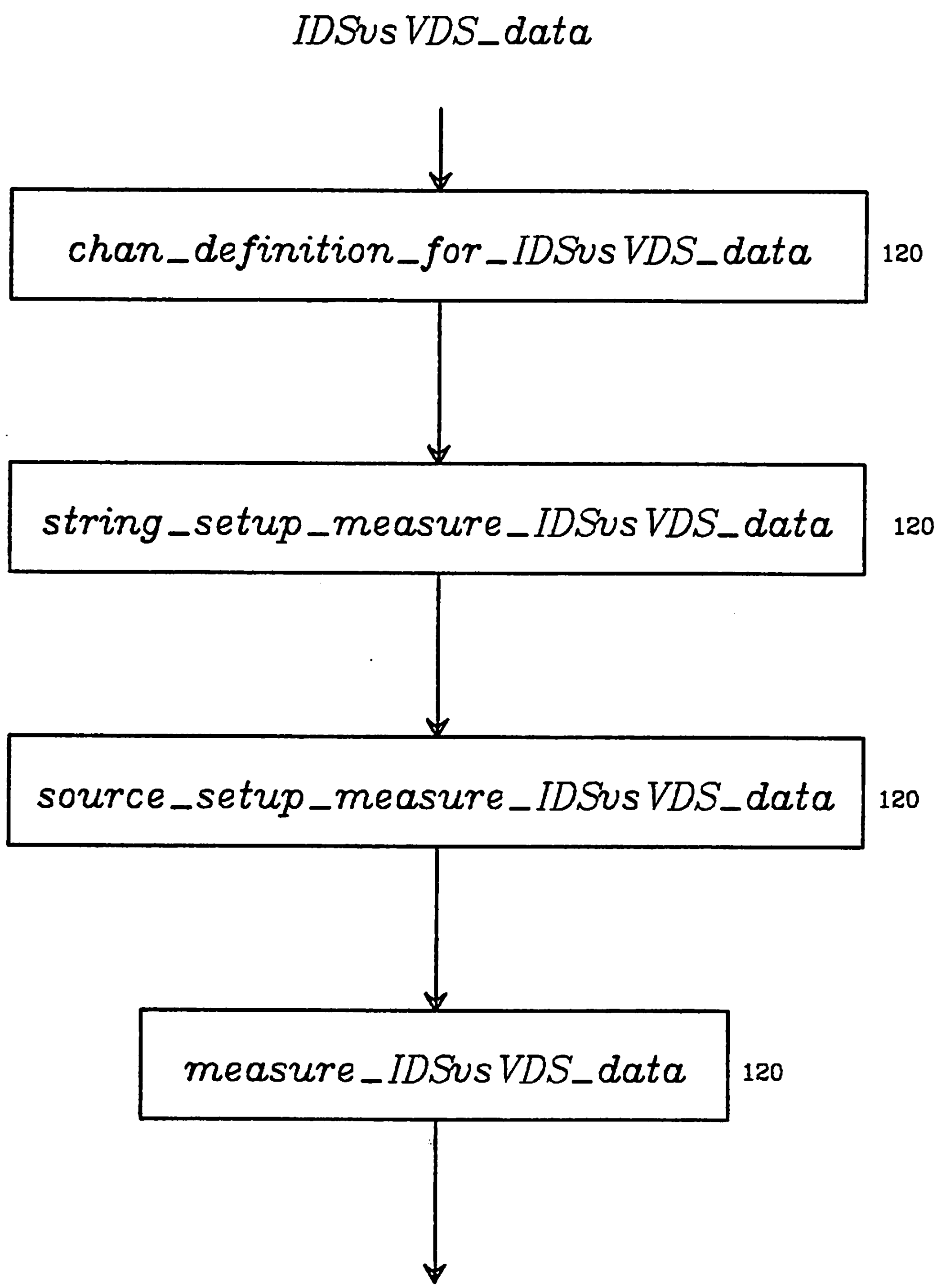
clear screen

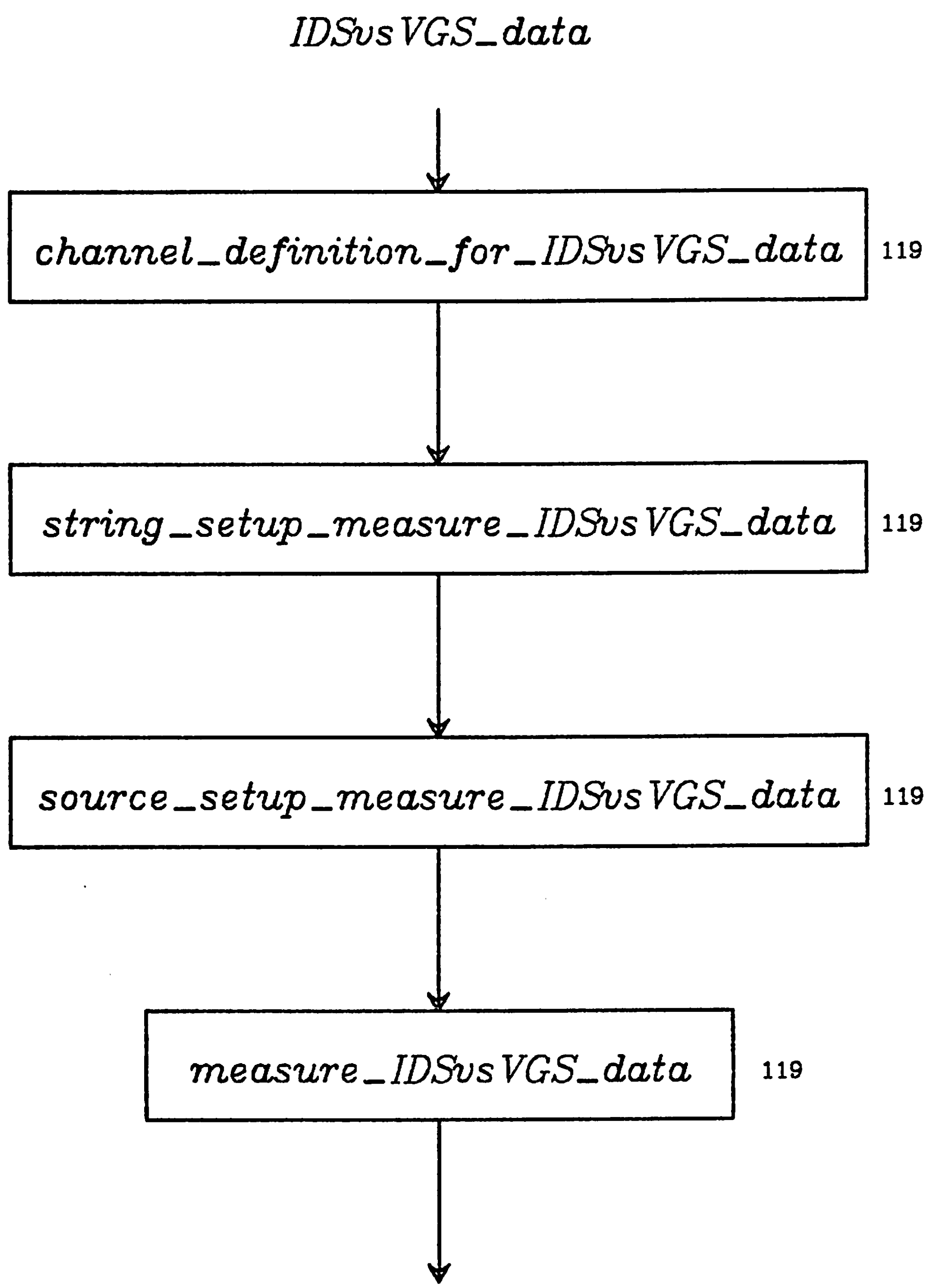
display

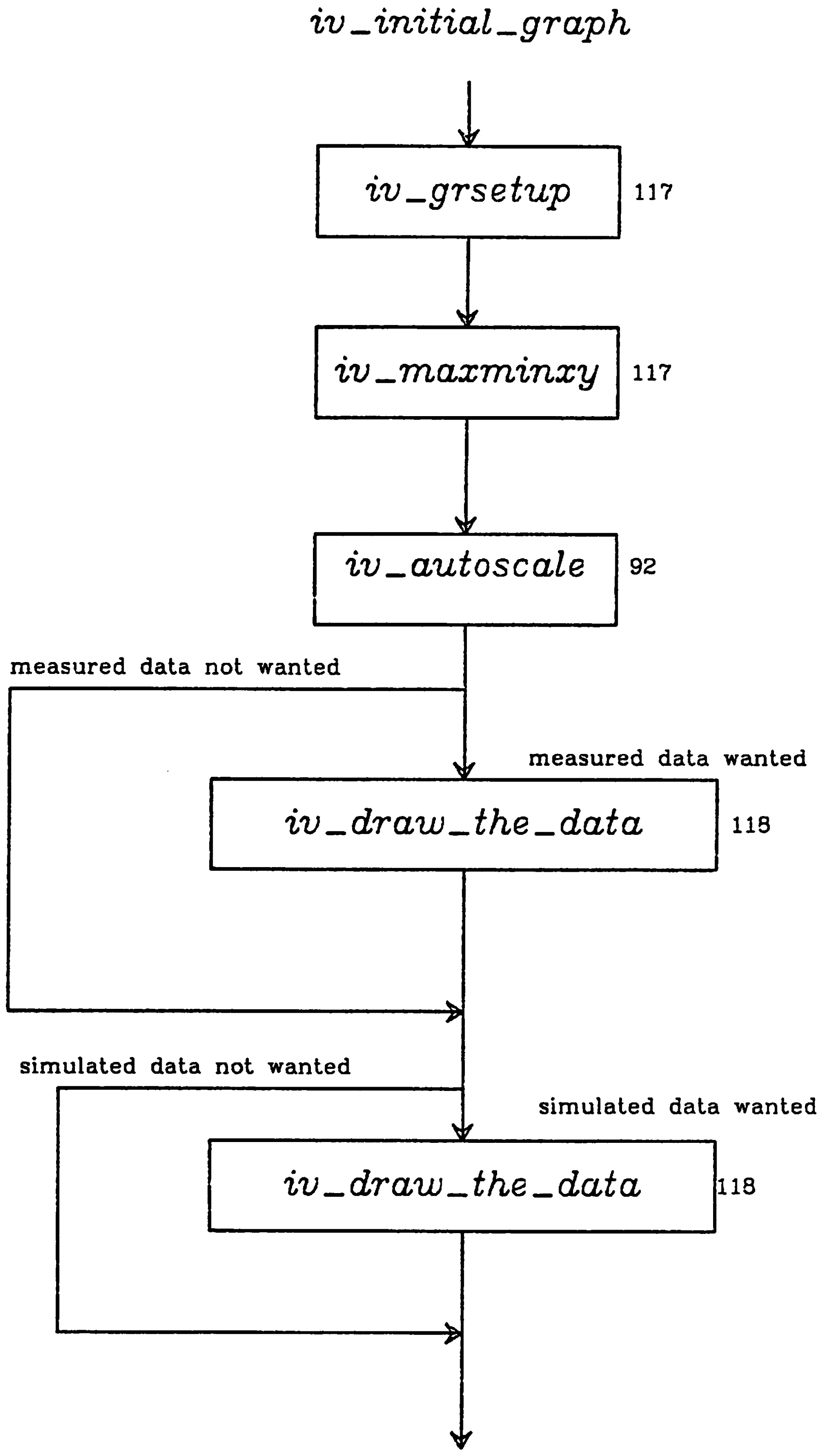
```
***BSIM I-V GRAPHICS MENU***

The BSIM I-V graphics routines will draw measured and/or simulated I-V data.
If the program is operating in the "SINGLE" mode, the 17 ELECTRICAL parameters
just extracted will be used. In the "AUTOMATIC" or "SEMI-AUTOMATIC" mode, the
17 ELECTRICAL parameters will be generated from the 54 parameter process file.

        SELECT A NUMBER FOR A GIVEN DISPLAY MODE=3
              1)Measured Data Only
              2)Simulated Data Only
              3)Measured and Simulated Data

         SELECT A NUMBER FOR A GIVEN GRAPH TYPE=1
              1)IDS versus VDS           VBS=?  >0
              2)IDS versus VGS           VDS=?  >
              3)ln(IDS) versus VDS       VBS=?  >

                 NEW SMU CONNECTIONS?  (Y/N)  >
```

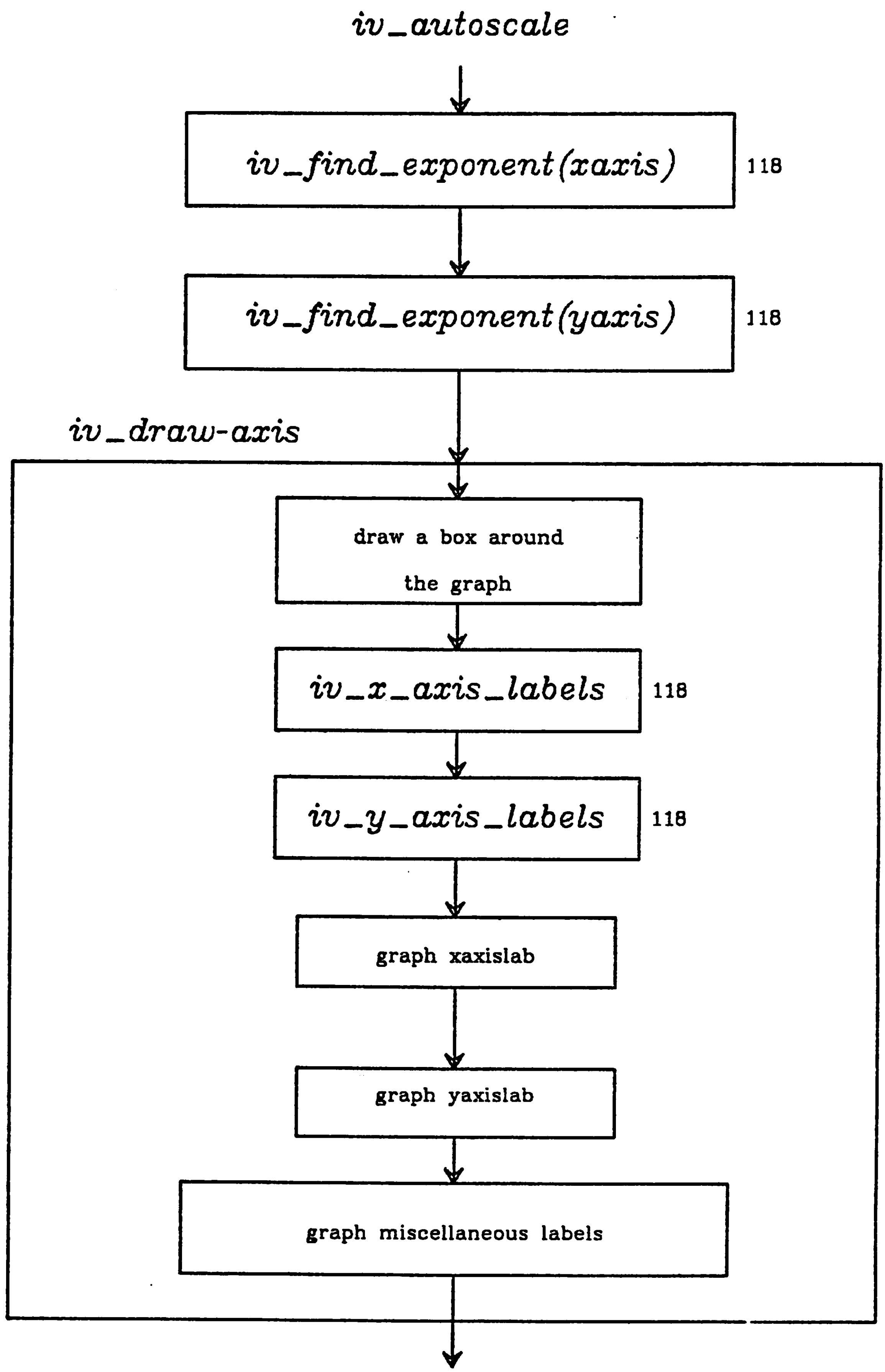

lnIDvsVD
clear screen
set up miscellaneous labels
not(measured_data)
measure_data
measured_data
translate_terminals_to_strings
119
IDSvsVDS_data
89
for i=1 to numbz
for j=1 to numbx
set minimum level of 1pA and take log
next i
next j
not(simonly)
simonly
determine x & z axis values
numbx & numbz
simulated not wanted
simulated wanted
for i=1 to numbx
for j=1 to numbz
bsimsim
120
set minimum level of 1pA and take log
measured or simulated
measured and simulated
error_calculations
121

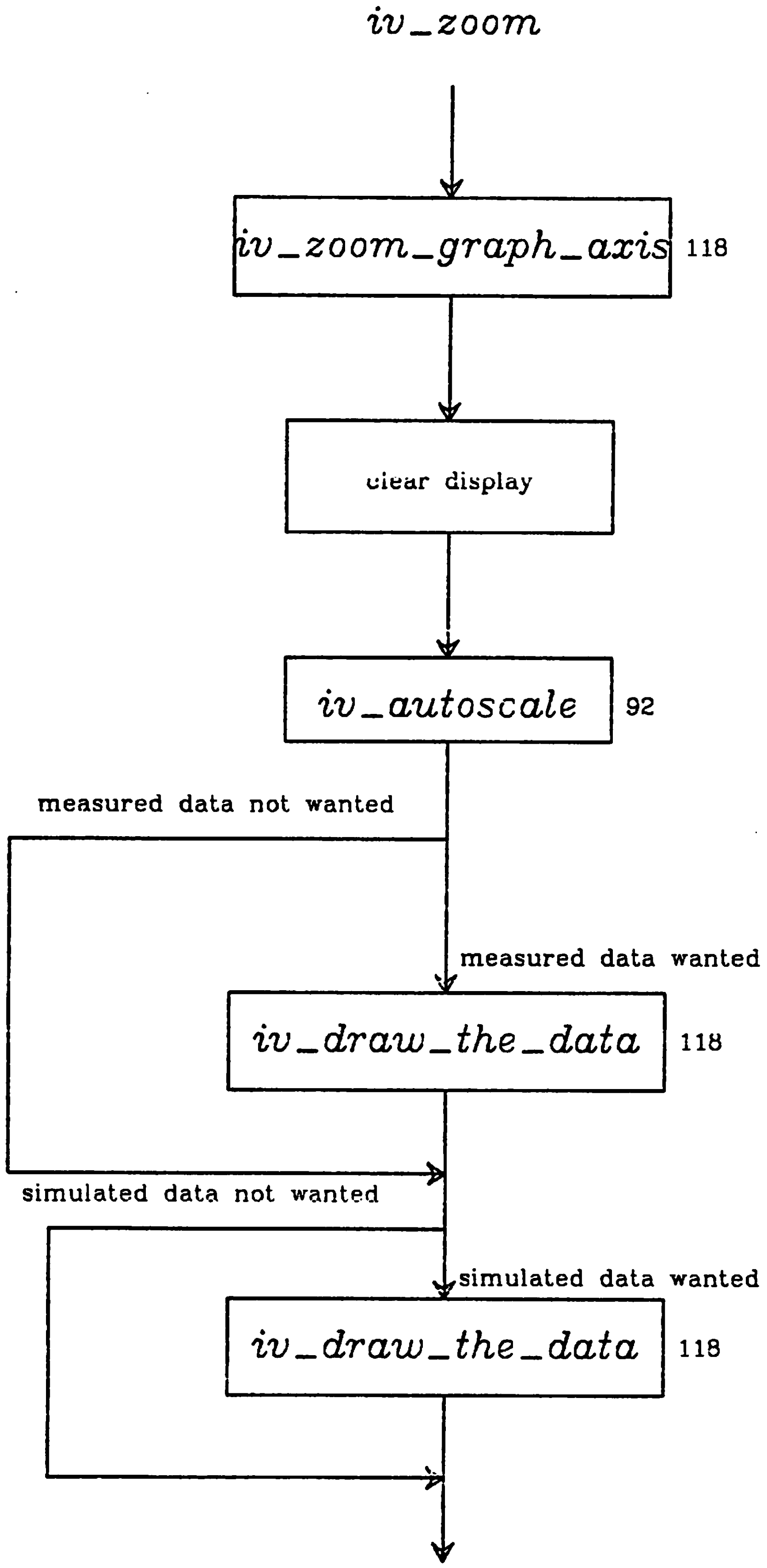

IDvsVD and IDvsVG
clear screen
set up miscellaneous labels
not(measured_data)
measure_data
measured_data
translate_terminals_to_strings
119
IDSvsVGS_data
90
IDSvsVDS_data
89
not(simonly)
simonly
determine x & z axis values
numbx & numbz
simulated not wanted
simulated wanted
for i=1 to numbx
for j=1 to numbz
bsimsim
120
measured or simulated
measured and simulated
error_calculations
121


IDSvsVDS_data
chan_definition_for_IDSvsVDS_data
120
string_setup_measure_IDSvsVDS_data
120
source_setup_measure_IDSvsVDS_data
120
measure_IDSvsVDS_data
120


IDSvsVGS_data
channel_definition_for_IDSvsVGS_data
119
string_setup_measure_IDSvsVGS_data
119
source_setup_measure_IDSvsVGS_data
119
measure_IDSvsVGS_data
119


iv_initial_graph
iv_grsetup
117
iv_maxminxy
117
iv_autoscale
92
measured data not wanted
measured data wanted
iv_draw_the_data
118
simulated data not wanted
simulated data wanted
iv_draw_the_data
118


iv_autoscale
iv_find_exponent(xaxis)
118
iv_find_exponent(yaxis)
118
iv_draw-axis
draw a box around the graph
iv_x_axis_labels
118
iv_y_axis_labels
118
graph xaxislab
graph yaxislab
graph miscellaneous labels


iv_zoom
iv_zoom_graph_axis
118
clear display
iv_autoscale
92
measured data not wanted
measured data wanted
iv_draw_the_data
118
simulated data not wanted
simulated data wanted
iv_draw_the_data
118

A.2.2. Number Manipulation Functions

realtostr

This function reads in a number from the user and converts it to a string. It accepts all 10 digits, E, e, + (plus), . (period), and - (minus).

digit

This function returns an integer given a character.

strtoreal

This function takes a real number string and converts it to the corresponding real number.

A.2.3. HPIB Procedures

talk_to_hpib

This procedure sets up the HP9836 computer to talk on the HPIB and a designated HPIB device to listen.

listen_to_hpib

This procedure sets up the HP9836 computer to listen on the HPIB and a designated HPIB device to talk.

wait_till_hpib_ready

This procedure performs a serial poll on the HPIB.

wait_till_bit7_IOSTATUS_set

This procedure polls bit 7 of the IOSTATUS register until it is set to a '1'. This is needed for the Electroglas prober, due to old software which controls the prober.

A.2.4. Graphing Functions

test_point_on_screen1

This function tests whether a pre-calculated point is within the -0.8 and +0.8 screen limits in the y-direction.

test_point_on_screen2

This function tests whether a pre-calculated point is within the -0.8 and +0.8 screen limits in the x-direction.

A.2.5. User Input Procedures

yes_no_selection_input

This procedure accepts moves to a location on the screen and accepts only a Y or y (yes), or N or n (no) character as inputs. Subsequent steps in the program are based on the decision.

selection_input

This procedure moves to a location on the screen and accepts a range of character values, given in the procedure call.

A.2.6. Least Squares Procedures

leastsq2

This procedure performs a least square fit for an equation of the form $Y = C + L(X)$, Sets of X and Y values are sent to the procedure, and the result is the values for C and L .

leastsq3

This procedure performs a least square fit for an equation of the form Y = C + L1(X1) + L2(X2) , Sets of X1, X2, and Y values are sent to the procedure, and the result is the values for C, L1, and L2 . If only 2 sets of data are sent, an error flag *(leastsq_divide_by_zero)* is set.

A.2.7. Miscellaneous Procedures

clear_output_file

This procedure simply clears out the output file.

initialize_17_bsim_parameters_to_zero

This procedure sets all 17 BSIM parameters to zero. This procedure assures that any values that remain from a previous run of the program are not accidently used for the current run.

input_standard_values

This procedure reads in all the values that are asked for in the procedure *standard_input_display*.

bsim_timer

This procedure is called throughout the program. It is the heart of the timer that shows itself as a series of "X's" on the screen as the program is running. Every time this procedure is called, another "X" is displayed on the screen, and the time remaining, both per wafer and per die, are modified.

A.2.8. Measurement Procedures

translate_terminals_to_strings

This procedure takes the SMU and V_{dd} inputs and creates strings out of them that the HP4145 Analyzer can understand.

chan_definition_to_test_device_type

This procedure sends values (e.g. CH1 'VDRAIN' 'IDRAIN' 1,3) on the HPIB to the HP4145 Analyzer for the Channel Definition Page, for determination of the device type.

source_setup_to_test_device_type

This procedure sends values (e.g. VC1 0.0, 0.01) on the HPIB to the HP4145 Analyzer for the Source Setup Page, for determination of the device type.

measure_device_type(var devtype,error:integer)

This procedure sends values on the HP4145 Analyzer for the Measurement Page. It reads back *IBODY* and *IGATE* from the HP4145 to determine the device type. The current levels are checked for a defective device. Possibilities include

1) **GATE SHORT**
2) **SHORTED JUNCTION**
3) **NO JUNCTION**

If an error exists, an error flag *(error)* is set. If no error exists, **NCHANNEL** or **PCHANNEL** is displayed on the screen.

chan_definition_device_functionality

This procedure sends values (e.g. CH1 'VDRAIN' 'IDRAIN' 1,3) on the HPIB to the HP4145 Analyzer for the Channel Definition Page, for determination of device functionality.

source_setup_nchannel_device_functionality
source_setup_pchannel_device_functionality

Each procedure sends values (e.g. VC1 5.0, 0.01) on the HPIB to the HP4145 Analyzer for the Source Setup Page, for determination of the device functionality.

measure_device_functionality(var devtype,error:integer)

This procedure sends values on the HP4145 Analyzer for the Measurement Page. It reads back *IDRAIN off* and *on* conditions, from the HP4145 to determine the device functionality. The current levels are checked for a defective device. Possibilities include

1) **OPEN DRAIN-SOURCE**
2) **SHORTED DRAIN-SRC**

If an error exists, an error flag *(error)* is set. If no error exists, the program continues.

chan_definition_for_IDSvsVGS_data

This procedure sends values (e.g. CH1 'VDRAIN' 'IDRAIN' 1,3) on the HPIB to the HP4145 Analyzer for the Channel Definition Page, for measuring the I_{ds} *vs* V_{gs} data.

source_setup_measure_IDSvsVGS

This procedure sends values (e.g. VR1 0.0, 5.0 0.01) on the HPIB to the HP4145 Analyzer for the Source Setup Page, for measuring the I_{ds} *vs* V_{gs} data.

string_setup_measure_IDSvsVGS

This procedure uses the determined device type and the entered value of V_{dd} and creates strings out of them that the HP4145 Analyzer can understand. It also fills up the V_{ds} and V_{bs} arrays.

measure_and_reduce_IDSvsVGS_data

First, this procedure sends values on the HP4145 Analyzer for the Measurement Page. The measurement is a graphics display with I_{ds} on the Y-axis and V_{gs} on the X-axis. Upon completion of measurement, it reads back *IDRAIN* from the HP4145.

Second, this procedure checks the data. All data below a fixed threshold $(0.1\times\frac{W}{L}\mu A)$ is ignored. Five values of I_{ds} are chosen from the remaining data such that their corresponding V_{gs} values are evenly spaced and the spacing is maximized. Figure A2.1 show an example of this.

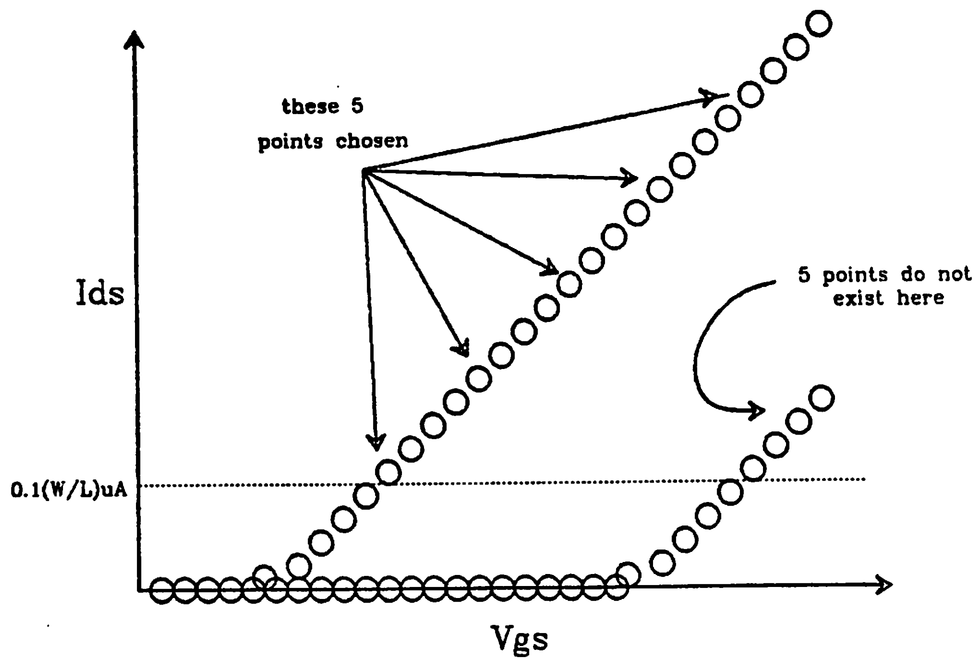


Figure A2.1
Optimum I_{ds} values are chosen

If 5 points do not exist, an error flag *(ids_vgs_array_error)* is set. The data is stored in the I_{ds} array.

Third, it fills up the V_{gs} array.

A.2.9. Extraction Procedures

A.2.9.1. Introduction to Extraction Summary

This section of Appendix 2 describes in some detail the BSIM parameter extraction process. It does not contain every line in the code file, nor does it simply list the procedure titles. It is meant to enlighten the reader on how the actual extraction process occurs.

Each procedure in the extraction process is listed, with a description following. A few abbreviations will be used; among them: lsq3 for calling procedure *leastsq3*, (a least square fit for a 3 variable equation), lsq2 for calling procedure *leastsq2*, (a least square fit for a 2 variable equation), and G for $\frac{I_{ds}}{V_{ds}}$. For the purposes of this example, assume $V_{dd}=5V$. The actual value of V_{dd} will determine the values of V_{ds} , V_{bs} , and V_{gs}. The procedures begin with 4 arrays of data already filled. They are:

1) **vds[vd]**

   vds[1]=0.1V, vds[2]=0.2V, vds[3]=4.5V, vds[4]=5.0V

2) **vbs[vb]**

   vbs[1]=0.0V, vbs[2]=-1.0V, vbs[3]=-2.0V, vbs[4]=-3.0V, vbs[5]=-4.0V, vbs[6]=-5.0V

3) **vgs[vd,vb,vg]**

   A value for V_{gs} corresponds to I_{ds} at the same bias conditions.

   vg = 1, 2, 3, 4, and 5V.

4) **ids[vd,vb,vg]**

   There is a value of I_{ds} for each value of V_{ds} , V_{bs} , and V_{gs}.

vd, vb, and vg are abbreviated from the program's *vdindex*, *vbindex*, and *vgindex* for ease of reading. After all measurement and reduction for a single device is finished, 120 I_{ds} and 120 V_{gs} values exist.

linear_region_extraction

This entire procedure is performed in 2 loops, the outside one is for vb = 1 to 6, and the inside one is for vd = 1 to 2. Figure A2.2 is used as a reference for this procedure.

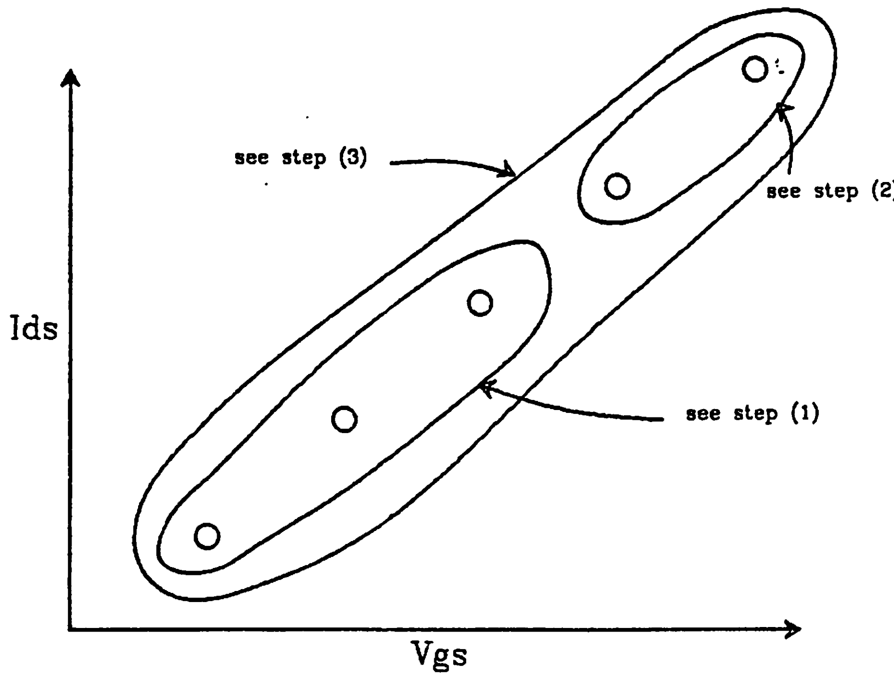


Figure A2.2
Points Remaining After Reduction of Data
Only Single Value of V_{bs} Shown For Clarity

(1) for each point do lsq3 for:

$$G = C_1 + L_1 V_{gs} + Q_1 V_{gs}^2$$

using quadratic fit, make initial estimate for V_{tx} and $Beta0x$:

$$V_{tx} = \frac{-2C_1}{L_1 + L_1 \left(1 - \frac{4C_1 Q_1}{L_1^2} \right)^2}$$

$$Beta0x = L_1 + 2Q_1 V_{tx}$$

(2) make an estimate for $U0x$. for each point do lsq2 for:

$$\frac{V_{gs} - V_{tx}}{G} = C_2 + L_2 \left(V_{gs} - V_{tx} \right)$$

since $C_2 = \frac{1}{Beta0x}$ and $L_2 = \frac{U0x}{Beta0x}$ then

$$U0x = \frac{L_2}{C_2}$$

(3) now do the final fit for

$$G = \frac{\beta(V_{gs} - V_{tx})}{1 + U0x(V_{gs} - V_{tx})}$$

let $U0x = U0x + \Delta\, U0x$, $V_{th} = V_{th} + \Delta\, V_{th}$, and $\beta = \beta + \Delta\,\beta$, then:

$$G = \frac{\beta(V_{gs} - V_{tx}) + \Delta\,\beta(V_{gs} - V_{tx}) - \Delta\,\beta\Delta\, V_{th}}{1 + U0x(V_{gs} - V_{tx}) + \Delta\, U0x(V_{gs} - V_{tx}) - \Delta\, U0x\Delta\, V_{th}}$$

since $\Delta\,\beta\Delta\, V_{th} \approx 0$ and $\Delta\, U0x\Delta\, V_{th} \approx 0$, then after rearranging do lsq3 for all 6 points:

$$\frac{G}{V_{gs} - V_{tx}} + U0x \times G - \beta = C_3 + L_3\left[\frac{U0x \times G - \beta}{V_{gs} - V_{tx}}\right] + L4(-G)$$

where $L_3 = \Delta\, V_{th}$, $L4 = \Delta\, U0x$, and $C_3 = \Delta\,\beta$.

iterate step (3) until $\frac{\Delta\,\beta}{\beta}$, L_3, and $\frac{\Delta\, U0x}{U0x}$ are minimized.

(4) final results:

beta0[vd,vb]

vth[vd,vb]

u0x[vd,vb]

...all for the linear region.

large_device_20f_extraction

this entire procedure is performed within a loop for vd = 1 to 2.

(1)

phif2[vd]=0.6

k2[vd]=0

for 1st die:

s=1

for other die: $s = 1 + \frac{K1_L}{L + \Delta L} + \frac{K1_W}{W + \Delta W}$

Na=f(phif2[vd])

k1=f(s,phif2[vd])

(2)

(2a) for vb = 1 to 6 do calculation of *temp1* and *temp2* and perform lsq3 on 2nd *temp2=*... equation.

$$temp1 = 1 - k2[vd] + \frac{k1[vd]}{2\sqrt{phif2[vd] - vbs[vb]}} + \frac{k1[vd]\sqrt{phif2[vd] - vbs[vb]}}{4ktq}$$

$$+ \frac{V_{ds}}{2}\left\{\frac{-gk1[vd]}{4(phif2[vd] - vbs[vb])\sqrt{phif2[vd] - vbs[vb]}} + \frac{k1[vd](0.8364)}{2\sqrt{phif2[vd] - vbs[vb]}\,t^2}\right\}$$

where $t = 1.744 + 0.836(phif2[vd] - vbs[vb])$.

$$temp2 = vth[vd,vb] - phif2[vd] - k1[vd]\sqrt{phif2[vd] - vbs[vb]} - \frac{a[vd,vb] \times vds[vd]}{2}$$

$$temp2 = vfb[vd] + L \times temp1 - k2[vd](phif2[vd] - vbs[vb])$$

(2b) calculate values

$$phif2[vd] = phif2[vd] + L$$

$$k1[vd] = f(s, phif2[vd])$$

$$Na = f(phif2[vd])$$

(3) iterate (15 times maximum) steps (2a) and (2b) until L is minimized

(4) final results:

k1[vd]

k2[vd]

vfb[vd]

...all for linear region.

linear_region_threshold_analysis

This entire procedure is performed within a loop for vd = 1 to 2.

(1)

phif2[vd]=value from previous device extraction

s=1

Na=f(phif2)

k1=f(s,phif2)

(2) for vb = 1 to 6 do lsq3 on

$$vth[vd,vb] - phif2[vd] - \frac{a[vd,vb] \times vds[vd]}{2} = vfb[vd] +$$

$$k1[vd]\left(\frac{vds[vd] \times g}{4\sqrt{phif2[vd] - vbs[vb]}} + \sqrt{phif2[vd] - vbs[vb]}\right) - k2[vd]\sqrt{phif2[vd] - vbs[vb]}$$

where $g = 1 - \dfrac{1}{1.744 + 0.8364(phif2[vd] - vbs[vb])}$

(3) final results:

k1[*vd*]

k2[*vd*]

vfb[*vd*]

...all for linear region.

and *phif2*[*vd*] is already known.

linear_region_data_reduction

(1) this section is performed within 2 loops, the outside one is for vd = 1 to 2 and the inside one is for vb = 1 to 6:

$$a[vd,vb] = 1 + \left(1 - \frac{1}{1.744 + 0.8364(phif2[vd] - vbs[vb])}\right)\frac{k1[vd]}{2}\frac{1}{\sqrt{phif2[vd] - vbs[vb]}}$$

$$vth[vd,vb] = vth[vd,vb] - \frac{a[vd,vb]vds[vd]}{2}$$

$$u0x[vd,vb] = \frac{u0x[vd,vb]}{\left(1 - u0x[vd,vb]\frac{a[vd,vb]vds[vd]}{2}\right)}$$

(2) do lsq3 on:

$$u0x[vd,vb] = \mathrm{U0} + \mathrm{X2U0}(vbs[vb]) + x3u0(vds[vd])$$

(3) for vb = 1 to 6 do:

$$u0zerovds[vb] = \mathrm{U0} + \mathrm{X2U0}(vbs[vb])$$

(4) for vd = 1 to 2 do lsq2 on:

$$vfb[vd] = \mathrm{VFB} + linear1(vds[vd])$$

(5) for each type, if 1st good device of the die *(20f_meas=TRUE)* then for vd = 1 to 2 do lsq2 on:

$$phif2[vd] = \mathrm{PHIF2} + linear1(vds[vd])$$

(6) for vd = 1 to 2 do lsq2 on:

$$k1[vd] = \mathrm{K1} + linear1(vds[vd])$$

(7) for vd = 1 to 2 do lsq2 on:

$$k2[vd] = \mathrm{K2} + linear1(vds[vd])$$

(8) for vb = 1 to 6 do:

$$vt0[vb] = \mathrm{VFB} + \mathrm{PHIF2} + \mathrm{K1}\sqrt{phif2[vd] - vbs[vb]} + \mathrm{K2}(phif2[vd] - vbs[vb])$$

$$azerovds[vb] = 1 + \left[1 - \frac{1}{1.744 + 0.8364(phif2[vd] - vbs[vb])}\right]\frac{\mathrm{K1}}{2}\frac{1}{\sqrt{phif2[vd] - vbs[vb]}}$$

(9) for vds = 1 to 2, for vbs = 1 to 6 do:

$$eta[vd,vb] = \frac{vth[vd,vb] - vto[vb]}{vds[vd]}$$

(10) final results:

VFB , PHIF2 , K1 , K2 , U0 and X2U0

eta[*vd,vb*]

azerovds[*vb*]

...all for linear region.

saturation_region_data_extraction

this entire procedure is performed within 2 loops, the outside one is for vd = 4 to 3 and the inside one is for vb = 6 to 1.

(1)

u0sat=u0zerovds[*vb*]

asat=azerovds[*vb*]

call procedure *initial_saturation_region_estimates* to get back values for *beta0sat*, *u1sat*, and *vthsat*.

(2)

(2a) for vg = 1 to 5 calculate derivatives for *temp1*, *temp2*, and *temp3* where:

$$temp1 = \frac{I_{ds_{meas}} - I_{ds_{sim}}}{\frac{\partial I_{ds_{sim}}}{\partial beta0sat}}$$

$$temp2 = \frac{\frac{\partial I_{ds_{sim}}}{\partial vthsat}}{\frac{\partial I_{ds_{sim}}}{\partial beta0sat}}$$

$$temp3 = \frac{\frac{\partial I_{ds_{sim}}}{\partial u1sat}}{\frac{\partial I_{ds_{sim}}}{\partial beta0sat}}$$

(2b) for i=1 to *numbon* or for i= *numbsat+1* to *numbon* do lsq3 on:

$$temp3 = \Delta\, beta0sat + temp1(\Delta\, vthsat) + temp2(\Delta\, u1sat)$$

(2c)

$$u1sat = u1sat + \Delta\, u1sat \ \ or\ \ \frac{u1sat}{10} \text{ if } it's\ going\ negative$$

$$beta0sat = beta0sat + \Delta\, beta0sat$$

or use *beta0sat* from vds=5.0V if > *beta0sat* from vds=5.0V.

$$vthsat = vthsat + \nabla\, vthsat$$

(2d) if $\frac{\Delta\, beta0sat}{beta0sat} < E{-}8$ then *iterate* =false

if $u1sat < E{-}7$ then *u1sat* =0.

(2e)

if $u1sat < E{-}7$ and $(nrindex > 6\ or\ iterate = false)$ then

iterate =false

call procedure *zero_u1_saturation_extraction*

(3) iterate entire step (2) 12 times if necessary

(4)

$$eta[vd,vb] = \frac{vto[vb] - vth[vd,vb]}{vds[vd]}$$

(5) final results:

vth[*vd,vb*]

beta0[*vd,vb*]

u1[*vd,vb*]

eta[*vd,vb*]

...all for saturation region.

initial_saturation_region_estimates

(1) if vb=6 and vd=4 then

u1sat=0

vthsat=vt0[6] - eta[1,6] × vds[4]

beta0sat=beta0[1,6]

(2) if vb=6 and vd=3 then

etasat=eta[4,6]

u1sat=u1[4,6]

vthsat=vt0[6] - eta[1,6] × vds[3]

beta0sat=beta0[4,6]

(3) if vb <6 then

etasat=eta[vd,vb+1]

u1sat=u1[vd,vb+1]

vthsat=vt0[vb] - eta[vd,vb+1] × vds[vd]

beta0sat=beta0[vd,vb+1]

zero_u1_saturation_extraction

(1)

(1a) for vg = 1 to 5 calculate derivatives for *temp1* and *temp2* where:

$$temp1 = \frac{I_{ds_{meas}} - I_{ds_{sim}}}{\frac{\partial I_{ds_{sim}}}{\partial beta0sat}}$$

$$temp2 = \frac{\frac{\partial I_{ds_{sim}}}{\partial vthsat}}{\frac{\partial I_{ds_{sim}}}{\partial beta0sat}}$$

(1b) for i=1 to *numbon* do lsq2 on:

$$temp2 = \Delta\, beta0sat + temp1(\Delta\, vthsat)$$

(1c)

$$beta0sat = beta0sat + \Delta\, beta0sat$$

$$vthsat = vthsat + \Delta\, vthsat$$

(1d) if $\frac{\Delta\, beta0sat}{beta0sat} < E{-8}$ then *iterate* =false

(2) iterate entire step (1) 12 times if necessary

(3) final results:

beta0sat and *vthsat* which go into the calling procedure *saturation_region_data_extraction*.

saturation_region_data_reduction

(1) for vb = 1 to 6

for vd = 3 to 4 do lsq3 on:

$$u1[vd,vb] = U1 + X2U1(vbs[vb]) + X3U1(vds[vd]-vdd)$$

(2) for vb = 1 to 6

for vd = 3 to 4 do lsq3 on:

$$eta[vd,vb] = ETA + X2ETA(PHIF2 - vbs[vb]) + X3ETA(vds[vd]-vdd)$$

(3) for vb = 1 to 6 do:

$$etavddvds[vb] = \mathrm{ETA} + \mathrm{X2ETA}(\mathrm{PHIF2} - vbs[vb])$$

(4) for vd = 1 to 2

for vb = 1 to 6

$$u1lin = \mathrm{U1} + \mathrm{X2U1}(vbs[vb]) + \mathrm{X3U1}(vds[vd]-vdd)$$

if $u1lin < 0$ then $u1lin = 0$

$$beta0[vd,vb] = beta0[vd,vb][1 + u1lin(vds[vd])]$$

(5) for vb = 1 to 6

for vd = 1 to 2 do lsq3 on:

$$beta0[vd,vb] = \mathrm{BETA0} + \mathrm{X2BETA0}(vbs[vb]) + X3BETA0(vds[vd])$$

(6) for vb = 1 to 6

for vd = 3 to 4 do lsq3 on:

$$beta0[vd,vb] = \mathrm{BETA0SAT} + \mathrm{X2BETA0SAT}(vbs[vb]) + \mathrm{X3BETA0SAT}(vds[vd]-vdd)$$

(7)

final results:

U1, X2U1, X3U1, ETA, X2ETA, X3ETA,

BETA0, X2BETA0, BETA0SAT, X2BETA0SAT, and X3BETA0SAT

A.2.10. Process File Development Procedures

store_parameters_in_die_files

This procedure first reads any information already in the current die file and writes it into a temporary file, and then writes the current device parameters into the temporary file also. The entire temporary file is then read back into the current, now empty die file.

die_file_creation

This procedure zeroes out the 6 variables which contain the number of good devices for each device type. It also creates the 6 die files, corresponding to the 6 device types.

l_and_w_dependencies

This procedure performs two similar functions, depending on the data supplied to it.

1) If the parameter supplied is **BETA0** , it performs a least square fit in 2 variables on all of the size-dependent **BETA0** parameters for all values of W and L to fit Equation A2.1

$$\mathbf{MUO_0} = \frac{\mathbf{BETA0}_i (L_i + \Delta L)}{C_{ox} (W_i + \Delta W)} \qquad \text{A2.1}$$

2) For all other parameters, this procedure performs a least square fit in 3 variables to fit Equation 1.1

$$\mathbf{P}_i = \mathbf{P}_0 + \frac{\mathbf{P}_L}{L_i + \Delta L} + \frac{\mathbf{P}_W}{W_i + \Delta W} \qquad 1.1$$

where the process file parameters for each of the 17 BSIM parameters will be substituted for $\mathbf{P}_0$, $\mathbf{P}_L$, and $\mathbf{P}_W$ Prior to any processing in this procedure, the data is checked for any duplicate values of W or L, since this would imply dependent equations would be sent to the least square procedure, and only one set of such data is kept.

process_parameter_errors

Using the just determined parameters $\mathbf{P}_0$, $\mathbf{P}_L$, and $\mathbf{P}_W$, this procedure simulates a value **P** using the width W and length L of each device in Equation A2.2.

$$\mathbf{P} = \mathbf{P}_0 + \frac{\mathbf{P}_L}{L + \Delta L} + \frac{\mathbf{P}_W}{W + \Delta W} \qquad \text{A2.2}$$

This is compared to the extracted value for each device. The device with the largest RMS deviation between extracted and simulated parameters is remembered and the device size and RMS error is later stored in the process file.

write_into_the_output_file

This procedure first reads any information already in the current output file and writes it into a temporary file, and then writes the current output file also into the temporary file. The entire temporary file is then read back into the current, now empty output file.

erase_temporary_files

This procedure removes all files used during the execution of the program but not needed as permanent output later.

A.2.11. BSIM Parameter vs W or L Graphing Procedures

grsetup

This procedure initializes the graphics display.

ymin_ymax_check

This procedure checks to see if the current yvalue is larger or smaller than the current *ymax* and *ymin* value. If it is, it keeps it.

maxminxy

This procedure finds the maximum and minimum X and Y values to be graphed. It checks only those values relevant to the current graph. For example, if we are plotting only a single value of L, then only those parameters with that value of L will be checked. First all electrical parameters are checked (e.g. **VFB** for all relevant sizes). Then the process file values are used to generate all relevant combinations of minimum and maximum Y values. (e.g. $\mathbf{VFB}_{sim} = \mathbf{VFB}_0 + \frac{\mathbf{VFB}_L}{L + \Delta L} + \frac{\mathbf{VFB}_W}{W + \Delta W}$). The minimum X value is always 0 and the maximum X value is $\frac{1}{L_{largest}}$ or $\frac{1}{W_{largest}}$, whichever is relevant.

find_exponent

Given the maximum and minimum values for an axis, this procedure determines the exponent for this value. Exponents in factors of 3 are used.

x_axis_labels
y_axis_labels

Using the just determined X-axis or Y-axis exponent, and the minimum and maximum graphing values, these procedures draw out the scales, the grid, and the tick marks.

draw_the_data1

This procedure draws out the electrical size-dependent parameters. A new plotting character is chosen for each new Z-value. A maximum of 10 different plotting characters are used. More than 10 different values to be plotted results in the characters being repeated. After completion of plotting each size, record of the plotting character and the size it represents is made on the right hand size of the graph.

plot_point

This procedure moves to the new X,Y location (offset by half the size of the plotting character) and plots the point with the current plotting character.

draw_the_data2

This procedure draws out the straight line which is generated from the process file parameters. This line is for the equation: $P_{sim} = P_0 + \frac{P_{\ell}}{L + \Delta L} + \frac{P_{\ell}}{W + \Delta W}$. This line is drawn by calculating 2 points, one for maximum X and the other for minimum X and connecting the 2 points with a line.

draw_line_and_zstring

This procedure draws a line to the X,Y line location given to it. Then it makes the current third variable value (L or W) string, moves to the X,Y string location and writes it on the graph.

zoom_graph_axis

This procedure, by using the locator knob on the HP9836, locates the endpoints of a rectangle drawn by the user. These endpoints become the new axis limits.

make_xyz_axis_labels

This procedure generates the X, Y, and Z axis labels. It also creates the labels at the top of the graph.

A.2.12. I-V Graphing Procedures

iv_grsetup

This procedure initializes the graphics display.

iv_maxminxy

This procedure finds the maximum and minimum X and Y values to be graphed. It checks only those values relevant to the current graph. For example, if we are plotting only measured data, then only the measured data will be checked, while both measured and simulated data will be checked if both are being plotted.

iv_find_exponent

Given the maximum and minimum values for an axis, this procedure determines the exponent for this value. Exponents in factors of 3 are used.

iv_x_axis_labels

iv_y_axis_labels

Using the just determined X-axis or Y-axis exponent, and the minimum and maximum graphing values, these procedures draw out the scales, the grid, and the tick marks.

iv_draw_the_data

This procedure draws simulated and/or measured data. An "x" plotting character is used for measured data while a solid line connects all simulated points. When zooming occurs, it is probable that the new X and Y maximum and minimum limits will not be directly at a data point. This procedure will draw the simulated data up to the edge of the graph. No information is lost in this manner.

iv_zoom_graph_axis

This procedure, by using the locator knob on the HP9836, locates the endpoints of a rectangle drawn by the user. These endpoints become the new axis limits.

translate_terminals_to_strings

This procedure takes the SMU and V_{dd} inputs and creates strings out of them that the HP4145 Analyzer can understand.

chan_definition_for_IDSvsVGS_data

This procedure sends values (e.g. CH1 'VGATE' 'IGATE' 1,3) on the HPIB to the HP4145 Analyzer for the Channel Definition Page, for measuring the I_{ds} *vs* V_{gs} data.

string_setup_measure_IDSvsVGS

This procedure uses the determined device type and the entered value of V_{dd} and creates strings out of them that the HP4145 Analyzer can understand.

source_setup_measure_IDSvsVGS_data

This procedure sends values (e.g. VR1 0.0, 5.0 0.01) on the HPIB to the HP4145 Analyzer for the Source Setup Page, for measuring the I_{ds} *vs* V_{gs} data.

measure_IDSvsVGS_data

First, this procedure sends values on the HP4145 Analyzer for the Measurement Page. The measurement is a graphics display with I_{ds} on the Y-axis and V_{gs} on the X-axis. It reads back *IDRAIN* from the HP4145. Then it fills up the V_{gs} and V_{bs} arrays.

chan_definition_for_IDSvsVDS_data

This procedure sends values (e.g. CH1 'VDRAIN' 'IDRAIN' 1,3) on the HPIB to the HP4145 Analyzer for the Channel Definition Page, for measuring the I_{ds} *vs* V_{ds} data.

string_setup_measure_IDSvsVDS

This procedure uses the determined device type and the entered value of V_{dd} and creates strings out of them that the HP4145 Analyzer can understand.

source_setup_measure_IDSvsVDS_data

This procedure sends values (e.g. VR1 0.0, 5.0 0.01) on the HPIB to the HP4145 Analyzer for the Source Setup Page, for measuring the I_{ds} *vs* V_{ds} data.

measure_IDSvsVDS_data

First, this procedure sends values on the HP4145 Analyzer for the Measurement Page. The measurement is a graphics display with I_{ds} on the Y-axis and V_{ds} on the X-axis. It reads back *IDRAIN* from the HP4145. Then it fills up the V_{gs} and V_{ds} arrays.

bsimsim

This procedure takes the 17 extracted electrical values or the 17 simulated parameters and generates a values of I_{ds} for a given value of V_{ds} , V_{bs} , and V_{gs} .

error_calculations

Given the measured and simulated data, this procedure calculates an RMS error for all points. This error value is later displayed on the graph in procedure *iv_draw_axis*.

A.2.13. Automatic Prober Procedures

read_prober_file

This procedure reads the prober file (which the user has created before running the program) and stores all the data in various variable names and arrays.

load_measure_two_phif_array

This procedure simply calls *find_and_switch_best_device* 6 times, once for each type of device.

find_and_switch_best_device This procedure searches the W and L arrays of devices to be tested; which were just read from the prober file, and determines the largest one. Then a switch is made between the first device in each array and the largest device. This ensures that the largest device is tested first. This is necessary for maximum accuracy when extracting the parameter **PHIF2** .

load_automatic_parameters

This procedure gets the next set of probing data (SMU connections, W, L, device type, and largest device?) from the arrays which were created in *read_prober_file*.

prober_move

Given X and Y values, this procedure sends the necessary information on the HPIB to tell the prober to move to the new X and Y device location for the current die.

set_die_size

This procedure sends the necessary information on the HPIB to tell the prober the correct die size.

set_up_prober_initial_conditions

This procedure sends the necessary information on the HPIB to set the prober to understand metric units and gives the direction for positive X and Y movement.

step_to_next_die

Given X and Y values, this procedure sends the necessary information on the HPIB to tell the prober to move to the new X and Y die location.

unload_wafer

This procedure returns the automatic prober head to the unload position. This allows easy access for putting a new wafer on it.

APPENDIX 3. Minor Changes to Previous Program

This Appendix lists some of the minor changes that were made to the program to enhance it. This list does not reflect all of the changes made.

(1) The functions *strtoreal* and *digit* were made more compact and each still perform the same functions.

(2) The HPIB procedures which were called quite often during the program have been made into "macro" procedures, thus simplifying the calling to a single line of code.

(3) During the entry of the output file name during the beginning of the program, the suffix ".TEXT" is no longer appended automatically. This allows the user to name the output file with any name.

(4) The HPIB addresses of the automatic prober and the analyzer are changeable with a single line of code.

(5) The timer procedure, *bsim_timer*, has been placed at different locations throughout the code, causing the timer to function more linearly.

(6) In the SINGLE MODE of operation, the process file data was always incorrectly headed with the name NMOSZ (NMOS Zero-threshold) at the beginning of it, regardless of what type of device was being tested. This has been corrected and the relevant device type now heads each process file.

(7) If the last device on a die was defective, the process file was never developed, because *process_file_development* was never called for a defective device. This has been corrected.

(8) As a device is being measured, both prior to parameter extraction and during I-V curve playback, the data is shown on the HP4145 Parameter Analyzer in the form of I-V curves instead of a matrix listing. This allows the user to easily spot if the device connections are bad, or if the device is bad.

(9) In the procedure *zero_u1_saturation_extraction* the derivative $\frac{\partial I_{ds}}{\partial V_{th}}$ is now calculated exactly and correctly.

(10) In SEMI-AUTOMATIC [WITH AUTOMATIC PROBER] MODE, the program pauses after each device instead of only after each die.

(11) Previously, if a parameter had a zero value, it was not used for development of the process file. Since 0.0 is a valid value for a parameter, it is now included in the development of the process file.

(12) If the first device on a die was bad, the parameter PHIF2 was not calculated and the program would stop. A change has been made such that the program will continue searching for a functional device until it finds one to extract this parameter from.

(13) If η, $\mu 0$, *or* $\mu 1$ are negative, they are set to 0.0 for I-V playback. Previously, only $\mu 1$ was checked.

(14) If the user decides to quit the program immediately after beginning, the program will now end gracefully, unlike before.

APPENDIX 4. Corrections to ERL memo M84/18

Below is a list of typographical errors which appeared in the ERL memo M84/18 *A Fully Automated MOS Device Characterization System For Process-Oriented Integrated Circuit Design* by Brian Scott Messenger.

Equation 2 should read:

$$I_{ds_{sat}} = \frac{Beta0sat\,(Vgs - Vthsat)^2}{(1 + U0(Vgs - Vthsat))\,a\left[1.0 + \frac{U1(Vgs - Vthsat)}{a} + \left(1.0 + \frac{2 \times U1(Vgs - Vthsat)}{a}\right)^{1/2}\right]}$$

Equation 3 should read:

$$Vdssat = \frac{\sqrt{2}(Vgs - Vth)}{a\sqrt{1 + \frac{U1(Vgs - Vth)}{a} + \left(1 + \frac{2 \times U1 \times (Vgs - Vth)}{a}\right)^{1/2}}}$$

Page 8, last paragraph, line 5 should read:

voltage equation (Equation 4) ...

Page 8, last paragraph, line 7 should read:

...three CSIM parameters (Equation 5)

Equation 5 should read:

$$ETA = ETA_0 + X2ETA(2Phif - VBS) + X3ETA(Vds - Vdd)$$

Equation 9 should read:

$$a=1.0+\left[1.0-\frac{1.0}{1.744+0.8364(2phif-Vbs)}\right]\frac{K1}{2}\frac{1}{\sqrt{2phif-Vbs}}$$

Page 12, paragraph 1, line 6 should read:

In multi-device extraction, the low field mobility, DeltaL and DeltaW are stored in the process file and are used by SPICE in determining the device gain. In contrast, BETA0 *only* is stored in single device extraction.

Equation 11 should read:

$$Parameter(W,L)=P0+\frac{PL}{L_{effective}}+\frac{PW}{W_{effective}}$$

Page 18, paragraph 2, line 2 should read:

output-conductance in the linear ...

Equation 12 should read:

$$G=\frac{I_{ds}}{V_{ds}}\approx\frac{Beta0x\times(Vgs-Vtx)}{1+U0x\times(Vgs-Vtx)} \qquad \text{for } V_{ds}^2\approx 0$$

Equation 13 should read:

$$\frac{Vgs-Vtx}{G}=\frac{1}{Beta0x}+\frac{U0x(Vgs-Vtx)}{Beta0x}$$

Equation 15 should read:

$$K1 = \frac{s\sqrt{2q\,\varepsilon_{Si}\,Na}}{Cox}$$

Equation 16 should read:

$$Na = n_i\, e^{\frac{2Phif \times q}{2k \times Temp}}$$

Page 21, paragraph 2, line 2 should read:

hold voltage equation (Equation 18) ...

Page 22, paragraph 2, line 4 should read:

UO and **X2UO** (Equation 22) ...

Equation 29 should read:

$$U1[Vds, Vbs] = U1 + X2U1(Vbs) + X3U1(Vds - Vdd)$$

Equation 32 should read:

$$Beta0[Vds, Vbs] = BETA0 + X2BETA0(Vbs) + (discarded\ term)(Vds)$$