IMPLEMENTATION OF THE BSIM SUBSTRATE CURRENT AND DEGRADATION MODELS IN SCALP

by

M. M. Kuo, K. Seki, P. M. Lee,
P. K. Ko, and C. Hu

Memorandum No. UCB/ERL M87/39

6 June 1987

COVER PAGE

IMPLEMENTATION OF THE BSIM SUBSTRATE CURRENT AND DEGRADATION MODELS IN SCALP

by

M. M. Kuo, K. Seki, P. M. Lee, P. K. Ko, and C. Hu

Memorandum No. UCB/ERL M87/39

6 June 1987

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

TITLE PAGE

IMPLEMENTATION OF THE BSIM SUBSTRATE CURRENT AND DEGRADATION MODELS IN SCALP

by

M. M. Kuo, K. Seki, P. M. Lee, P. K. Ko, and C. Hu

Memorandum No. UCB/ERL M87/39

6 June 1987

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

ABSTRACT

The BSIM substrate current and device degradation models have been implemented in the Substrate Current and Lifetime Processors (SCALP). SCALP is linked to SPICE externally in a pre- and post- processors fashion to form an independent simulator. The preprocessor reads in the input deck. SPICE simulates the transient voltage at the four terminal nodes of user-selected devices, and the post-processor calculates the substrate current and makes lifetime prediction based on the transient substrate current. The results show that the quasi-static device lifetime model used in SCALP for AC simulation can provide a good lifetime estimate except for devices which experience a fast V_G transients from high to low while V_D remains high.

ACKNOWLEDGEMENTS

I am grateful to Min-Chie Jeng and Benedict Liu for their help with the BSIM extraction program, Hong-June Park for his assistance with the C programming language and discussion of SPICE, Jeong Y. Choi for providing his experimental data on AC stressing, and Yupin Fong for his encouragement.

Finally, I would like to thank everyone in my family for being there whenever I needed them. It's their love and encouragement that carried me this far.

This research was supported by the IBM Faculty Development Award and the California MICRO program.

Table of Contents

I. INTRODUCTION 1

II. THE DEVICE DEGRADATION MODEL 3

2.1 Basic Equations 3

2.2 Theoretical Analysis 5

2.3 Process File 6

III. SYSTEM STRUCTURE AND IMPLEMENTATION 10

3.1 Structure of the Pre-Processor 12

3.2 Structure of the Post-Processor 15

3.3 Programming Limitations 18

IV. USER'S GUIDE FOR THE PROCESSORS 20

V. RESULTS 25

5.1 A Sample SCALP Run 25

5.2 Effect of .TRAN Control Line on Lifetime Simulation 25

5.3 Validity and Limitation of the Device Degradation Model 31

VI. FUTURE ENHANCEMENT AND CONCLUSION 36

6.1 Future Extension 36

6.2 Conclusion 39

I. INTRODUCTION

Hot-electron-induced MOSFET degradation is a major reliability issue in today's VLSI circuits. A good monitor of device degradation and lifetime has been shown to be the substrate current [1]. Also, excess substrate current can overload on-chip substrate bias generators and cause latch-up in CMOS circuits when the substrate potential variation due to the flow of the substrate current forward biases certain p-n junctions. Thus, the VLSI circuit designers have an urgent need for an effective hot-electron reliability simulator that can accurately monitor the substrate current in each device or in the entire circuit and provide a good estimate of the individual device's lifetime. The Substrate Current And Lifetime Processors (SCALP) is a tool that answers such a need.

An accurate parametric substrate current model and a device degradation model, both developed by Peter M. Lee, are the two models implemented in SCALP. SCALP contains a pre- and a post- processor which when used in conjunction with SPICE allows for the simulation of substrate current and device lifetime in analog and digital circuits. No modification to the original SPICE code is necessary for this implementation: consequently the SCALP simulator can be used with minimal setup time and works independently of the specific models implemented in SPICE. Using SCALP and SPICE together as a simulator system, the circuit designer can easily isolate areas within his circuit susceptible to adverse hot-electron and degradation effects.

The substrate current and the degradation models installed in SCALP are two recent additions of the BSIM (Berkeley Short-Channel IGFET Model) family [2]. All BSIM models are device physics based but contain empirically determined fitting parameters to offer accuracy and computational efficiency in circuit simulation. Extraction of the model parameters are done by using the BSIM Automated Parameter Extraction Program.

This report will discuss the degradation model, summarize the basic operation of this new simulator system, describe the implementation and the structure of the simulator, and present some experimental and simulation results. A discussion of the substrate current model used in SCALP can be found in [3].

In SCALP, the simulation of in-circuit device reliability is based on a DC stressing model. Despite concerns that DC and AC stressings produce different reliability results [4], recent experimental data [5] suggest that DC based models can be used to predict AC stressing lifetime in normal inverter circuits. In the section on results, we will show how our predicted AC stressing lifetime compares with the experimental data.

II. THE DEVICE DEGRADATION MODEL

2.1 Basic Equations

One parameter used to monitor device degradation is the change in threshold voltage (ΔV_{th}). Other parameters, e.g. $\Delta I_d / I_{do}$, may be used to characterize device degradation. In that case, ΔV_{th} in the equations below should be replaced by, for example, $\Delta I_d / I_{do}$. Under static stressing condition, the amount of threshold voltage shift as a function of time is [1]:

$$\Delta V_{th} = At^n \tag{1}$$

where n is dependent on the processing technology and bias condition. If ΔV_{thf} is the amount of shift in threshold voltage defined at device failure, and τ is the device lifetime, then

$$\Delta V_{thf} = A\tau^n$$

The lifetime can be written as [1]:

$$\tau = WBI_{bs}{}^{-m} I_{ds}{}^{m-1} \tag{2}$$

$$m = \frac{\phi_{it}}{\phi_i}$$

$$B = H \, \Delta V_{thf}{}^{\frac{1}{n}} \tag{3}$$

where H is dependent on device processing technology, W is the effective device width, and $q\phi_i$ and $q\phi_{it}$ are the critical energies required for impact ionization and the creation of interface traps, respectively. A typical log - log plot of τ versus I_{bs} is shown in Fig. 1. The above equations can then be combined with numerical calculations to find the device lifetime. The parameters from above which need to be extracted for SCALP are (n, H, and m) or (B and m). These two options will be discussed in section 2.3.

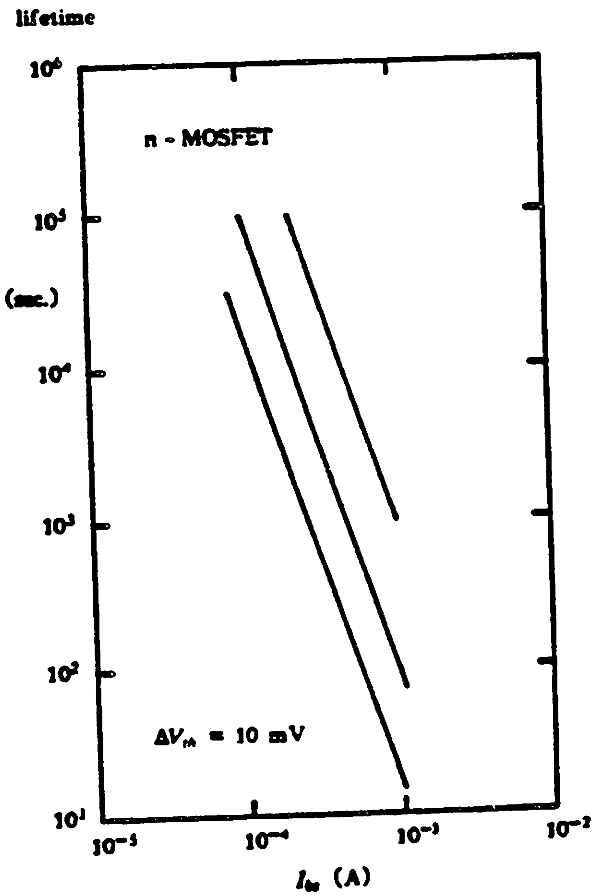


Fig. 1 Typical log - log plots of device lifetime τ versus I_{bs} for various device technologies (from Hu et al [1]).

2.2 Theoretical Analysis

Using a quasi-static approach. equations (1) and (2) can be extended to model the dynamic stressing behavior in a circuit for which the substrate current of a device is a function of time. Let $\Delta V_{thf} = \Delta V_{th} |_{t=\tau}$ be the threshold voltage shift defined at device failure. By solving for A in equation (1) using equation (2). we get:

$$\Delta V_{thf} = A\tau^{n} = A[WBI_{bs}{}^{-m} I_{ds}{}^{m-1}]^{n}$$
$$A = \Delta V_{thf}[WBI_{bs}{}^{-m} I_{ds}{}^{m-1}]^{-n}$$

Thus equation (1) becomes

$$\Delta V_{th} = \Delta V_{thf}(WB)^{-n} I_{bs}{}^{mn} I_{ds}{}^{n(1-m)} t^{n}$$

or

$$\Delta V_{th}{}^{\frac{1}{n}} = \Delta V_{thf}{}^{\frac{1}{n}}(WB)^{-1} I_{bs}{}^{m} I_{ds}{}^{(1-m)} t$$
$$= (WH)^{-1} I_{bs}{}^{m} I_{ds}{}^{(1-m)} t \qquad (4)$$

Since $\Delta V_{th}{}^{\frac{1}{n}}$ is a linear function of time. this quantity can simply be summed over the time period of the SPICE analysis. If $t_0, \ldots, t_P$ are the individual time points of SPICE. then

$$[\Delta V_{th\,(cycle)}]^{\frac{1}{n}} = [\Delta V_{th}(t_1 - t_0)]^{\frac{1}{n}} + \cdots + [\Delta V_{th}(t_P - t_{P-1})]^{\frac{1}{n}}$$

Moreover. to find the device lifetime assuming a periodic signal. a simple linear extrapolation in terms of $\Delta V_{th}{}^{\frac{1}{n}}$ is all that is necessary. The number of time intervals of the SPICE analysis needed so that $\Delta V_{th} = \Delta V_{thf}$ is simply

$$N = [\frac{\Delta V_{thf}}{\Delta V_{th\,(cycle)}}]^{\frac{1}{n}}$$

Thus. if the length of the SPICE analysis is $t_P = T$ and is equal to the period of the signal. the lifetime is found from

$$\tau = NT$$
$$= T[\frac{\Delta V_{thf}}{\Delta V_{th\,(cycle)}}]^{\frac{1}{n}}$$

or

$$\tau = \frac{TWH\,\Delta V_{thf}^{\;\frac{1}{n}}}{\sum_{h=1}^{p} [I_{bs}(t_h)]^m [I_{ds}(t_h)]^{1-m} (t_h - t_{h-1})} \tag{5}$$

$$= \frac{TWB}{\sum_{h=1}^{p} [I_{bs}(t_h)]^m [I_{ds}(t_h)]^{1-m} (t_h - t_{h-1})} \tag{6}$$

Both equations (5) and (6) are implemented in SCALP. If parameters H, n, and m are given in the process file, SCALP will use equation (5) to calculate the device lifetime. If parameters B and m are given, equation (6) will be used.

It should be pointed out that H and n are extraeted from equation (3) from B's measured at several ΔV_{thf}'s. However, the determination of the value of n is difficult and inaccurate because n is a function of processing technology and bias condition. At the present time, no empirical model for n has been developed. On the other hand, B can be more easily and accurately determined from stressing data for only one ΔV_{thf}. Thus, using B and m instead of H, n, and m offers more accuracy in the lifetime simulation. The user should note that the lifetime calculated by using equation (6) is based on the same ΔV_{thf} used to extract B.

2.3. Process File:

The degradation model parameters needed for the process file are (m and B) or (m, n, and H) and should be experimentally obtained by the user before he runs SCALP. The following paragraphs will briefly describe how these parameters are determined.

A lifetime versus I_{BS} experiment should be performed for at least one ΔV_{thf}. A typical result is shown in Fig. 1.

(1) m: The ratio of the trap energy and impact ionization energy $\frac{\phi_{it}}{\phi_i}$ which is approximately the slope of the log τ versus log I_{bs} curve in Fig. 1 (or the slope of log $\tau^* I_d$ versus log $\frac{I_{bs}}{I_d}$). A typical value for m is about 3 [1].

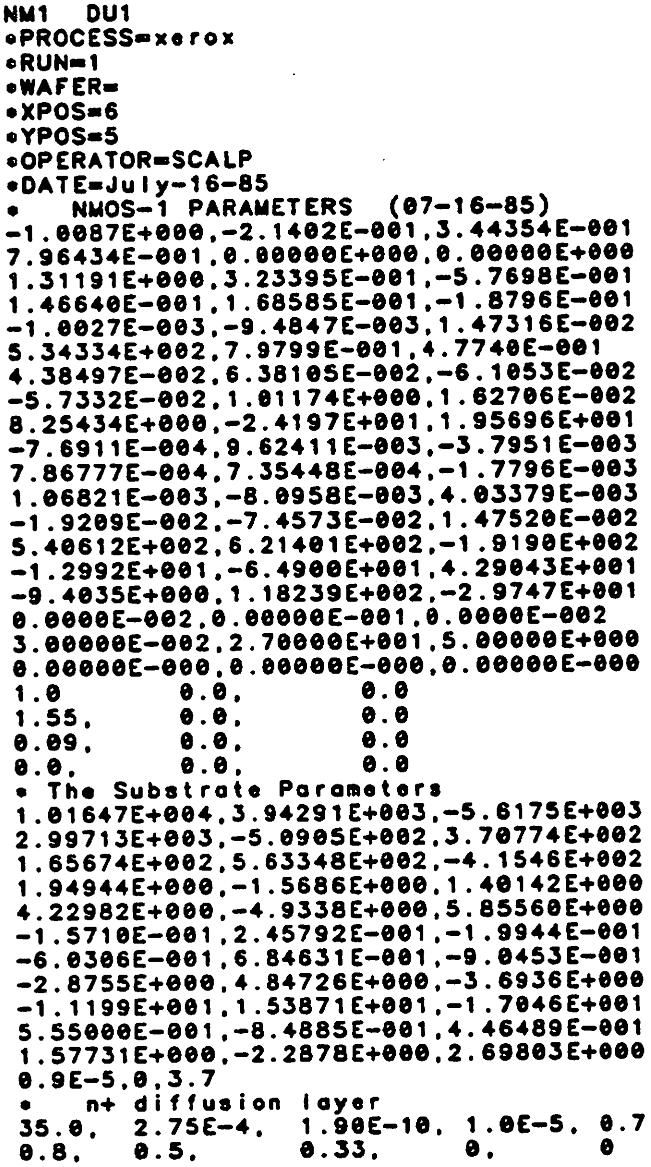
(2) B or (H and n): B is approximately the intercept of the curve in Fig. 1, and the unit used for B in this program is A-sec/μm. For each ΔV_{thf}, there is a value of B corresponding to it. The values of n and H can be obtained by fitting B values at various ΔV_{thf}'s to equation (3). H is dependent on the device processing technology. Typical values for n range from 0.5 to 0.75.

SCALP provides the flexibility to use either (B and m) or (H, n, and m) as the degradation parameters in the process file. To activate the first option, the users have to set the value of parameter n to zero and replace H by B. For example, if the last line of the process file is "1e-2, 0.5, 3.5", SCALP assumes parameters H, n, and m are used. If the last line of the process file is "1e-6, 0, 3.5", SCALP assumes that parameters B and m are used, because the value of n is zero. The user should note that the lifetime calculated by using the first option is based on the same ΔV_{thf} used to extract B.

The degradation parameters should be appended to the process file directly after the substrate current parameters in one row and in the order of H, n, and m, (see Fig. 2). Users who do not need any device lifetime information should enter 0,0,0 for H, n, and m respectively and use the appropriate commands given in section 4. The location of all BSIM parameters in the process file for proper SPICE read-in can also be found in Fig. 2. A sample process file is shown in Fig. 3.

	Name	L sens. factor	W sens. factor	Units of basic parameter
1	V_{FB} (VFB)	V_{FBl} (LVFB)	V_{FBw} (WVFB)	V
2	ϕ_S (PHI)	ϕ_{Sl} (LPHI)	ϕ_{Sw} (WPHI)	V
3	K_1 (K1)	K_{1l} (LK1)	K_{1w} (WK1)	$V^{1/2}$
4	K_2 (K2)	K_{2l} (LK2)	K_{2w} (WK2)	-
5	$\vec{\eta}_0$ (ETA)	η_{0l} (LETA)	η_{0w} (WETA)	-
6	μ_Z (MUZ)	δ_l (DL)	δ_w (DW)	cm^2/V-s. μm. μm
7	U_{0Z} (U0)	U_{0Zl} (LU0)	U_{0Zw} (WU0)	V^{-1}
8	U_{1Z} (U1)	U_{1Zl} (LU1)	U_{1Zw} (WU1)	μm V^{-1}
9	μ_{ZB} (X2MZ)	μ_{ZBl} (LX2MZ)	μ_{ZBw} (WX2MZ)	cm^2/V^2-s
10	η_B (X2E)	η_{Bl} (LX2E)	η_{Bw} (WX2E)	V^{-1}
11	η_D (X3E)	η_{Dl} (LX3E)	η_{Dw} (WX3E)	V^{-1}
12	U_{0B} (X2U0)	U_{0Bl} (LX2U0)	U_{0Bw} (WX2U0)	V^{-2}
13	U_{1B} (X2U1)	U_{1Bl} (LX2U1)	U_{1Bw} (WX2U1)	μm V^{-2}
14	μ_S (MUS)	μ_{Sl} (LMS)	μ_{Sw} (WMS)	cm^2/V^2-s
15	μ_{SB} (X2MS)	μ_{SBl} (LX2MS)	μ_{SBw} (WX2MS)	cm^2/V^2-s
16	μ_{SD} (X3MS)	μ_{SDl} (LX3MS)	μ_{SDw} (WX3MS)	cm^2/V^2-s
17	U_{1D} (X3U1)	U_{1Dl} (LX3U1)	U_{1Dw} (WX3U1)	μm V^{-2}
18	T_{ox} (TOX)	T_{emp} (TEMP)	V_{dd} (VDD)	μm. °C. V
19	CGDO	CGSO	CGBO	F/m
20	XPART	DUM1	DUM2	-
21	N0	LN0	WN0	-
22	NB	LNB	WNB	-
23	ND	LND	WND	-
24	E_{crit0} (ECRIT0)	E_{crit0l} (LECRIT0)	E_{crit0w} (WECRIT0)	V/cm
25	E_{critg} (ECRITG)	E_{critgl} (LECRITG)	E_{critgw} (WECRITG)	1/cm
26	E_{critb} (ECRITB)	E_{critbl} (LECRITB)	E_{critbw} (WECRITB)	1/cm
27	l_{c0} (LC0)	l_{c0l} (LLC0)	l_{c0w} (WLC0)	$\mu m^{1/2}$
28	l_{c1} (LC1)	l_{c1l} (LLC1)	l_{c1w} (WLC1)	$\mu m^{1/2}-V$
29	l_{c2} (LC2)	l_{c2l} (LLC2)	l_{c2w} (WLC2)	$\mu m^{1/2}-V^{-1}$
30	l_{c3} (LC3)	l_{c3l} (LLC3)	l_{c3w} (WLC3)	$\mu m^{1/2}$
31	l_{c4} (LC4)	l_{c4l} (LLC4)	l_{c4w} (WLC4)	$\mu m^{1/2}-V$
32	l_{c5} (LC5)	l_{c5l} (LLC5)	l_{c5w} (WLC5)	$\mu m^{1/2}-V^2$
33	l_{c6} (LC6)	l_{c6l} (LLC6)	l_{c6w} (WLC6)	$\mu m^{1/2}$
34	l_{c7} (LC7)	l_{c7l} (LLC7)	l_{c7w} (WLC7)	$\mu m^{1/2}-V$
35	H	n	m	-,-,-

Fig. 2 BSIM process file format.



```
NM1   DU1
•PROCESS=xerox
•RUN=1
•WAFER=
•XPOS=6
•YPOS=5
•OPERATOR=SCALP
•DATE=July-16-85
•    NMOS-1 PARAMETERS  (07-16-85)
-1.0087E+000,-2.1402E-001,3.44354E-001
7.96434E-001,0.00000E+000,0.00000E+000
1.31191E+000,3.23395E-001,-5.7698E-001
1.46640E-001,1.68585E-001,-1.8796E-001
-1.0027E-003,-9.4847E-003,1.47316E-002
5.34334E+002,7.9799E-001,4.7740E-001
4.38497E-002,6.38105E-002,-6.1053E-002
-5.7332E-002,1.01174E+000,1.62706E-002
8.25434E+000,-2.4197E+001,1.95696E+001
-7.6911E-004,9.62411E-003,-3.7951E-003
7.86777E-004,7.35448E-004,-1.7796E-003
1.06821E-003,-8.0958E-003,4.03379E-003
-1.9209E-002,-7.4573E-002,1.47520E-002
5.40612E+002,6.21401E+002,-1.9190E+002
-1.2992E+001,-6.4900E+001,4.29043E+001
-9.4035E+000,1.18239E+002,-2.9747E+001
0.0000E-002,0.00000E-001,0.0000E-002
3.00000E-002,2.70000E+001,5.00000E+000
0.00000E-000,0.00000E-000,0.00000E-000
1.0        0.0,       0.0
1.55,      0.0,       0.0
0.09,      0.0,       0.0
0.0,       0.0,       0.0
• The Substrate Parameters
1.01647E+004,3.94291E+003,-5.6175E+003
2.99713E+003,-5.0905E+002,3.70774E+002
1.65674E+002,5.63348E+002,-4.1546E+002
1.94944E+000,-1.5686E+000,1.40142E+000
4.22982E+000,-4.9338E+000,5.85560E+000
-1.5710E-001,2.45792E-001,-1.9944E-001
-6.0306E-001,6.84631E-001,-9.0453E-001
-2.8755E+000,4.84726E+000,-3.6936E+000
-1.1199E+001,1.53871E+001,-1.7046E+001
5.55000E-001,-8.4885E-001,4.46489E-001
1.57731E+000,-2.2878E+000,2.69803E+000
0.9E-5,0,3.7
•    n+ diffusion layer
35.0,  2.75E-4,  1.90E-10,  1.0E-5,  0.7
0.8,   0.5,      0.33,      0,       0
```

Fig. 3 Sample BSIM process file

III. SYSTEM STRUCTURE AND IMPLEMENTATION

This system contains three parts: a pre-processor, the SPICE program, and a post-processor. The two models are installed in the post-processor and are external to SPICE.

The pre-processor takes an input deck which contains commands given in section four and also commands prepared according to the SPICE syntax. SPICE is used in this system to simulate the circuit to determine the transient voltage waveforms at the drain, gate, source, and substrate of all user-selected devices. The post-processor calculates the transient substrate currents by using the node voltages provided by SPICE.

For the SPICE analysis, the user is free to choose any one of the models available in SPICE for the circuit simulation. To avoid confusion, from here on we shall refer to the SPICE2 BSIM model as "BSIM2", other SPICE2 models as "SPICE2", the SPICE3 BSIM model as "BSIM3", and other SPICE3 models as "SPICE3".

Besides the basic tasks, the operations of the pre- and post- processors differ somewhat for the BSIM2, BSIM3, SPICE2, and SPICE3 users. This was necessary because BSIM2 reads in the process file created by the BSIM extraction program directly whereas BSIM3, SPICE2, and SPICE3 need .MODEL commands with all parameters appended to the input deck. The system configuration was designed to retain as much commonality as possible between the two schemes (see Figs. 4 and 5). The *rawsub* file is created for communication between the two processors, and the *RAWMODEL* files are for storing model parameters. The *RAWPROC* files are the new process files created when BSIM2 is used. All these intermediate files are removed at the end of the simulation. More detailed discussion will be given in sections 3.1 to 3.3.

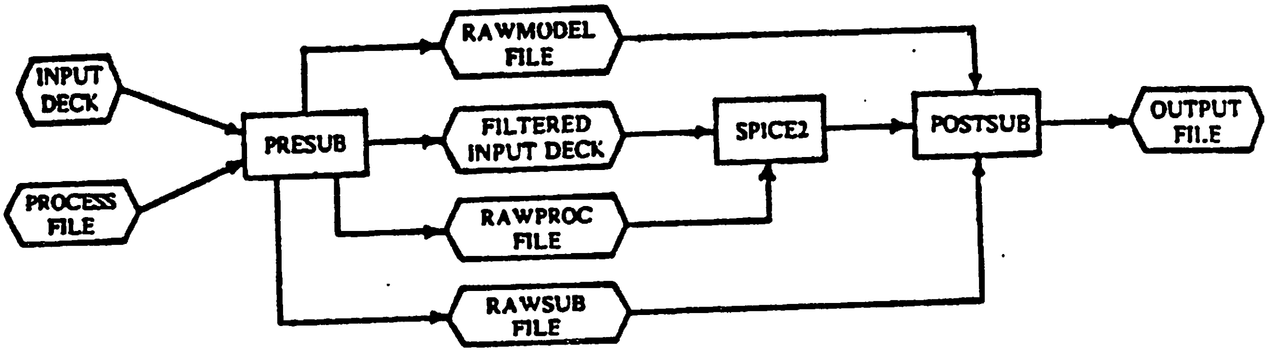


Fig. 4 BSIM2 processor configuration.

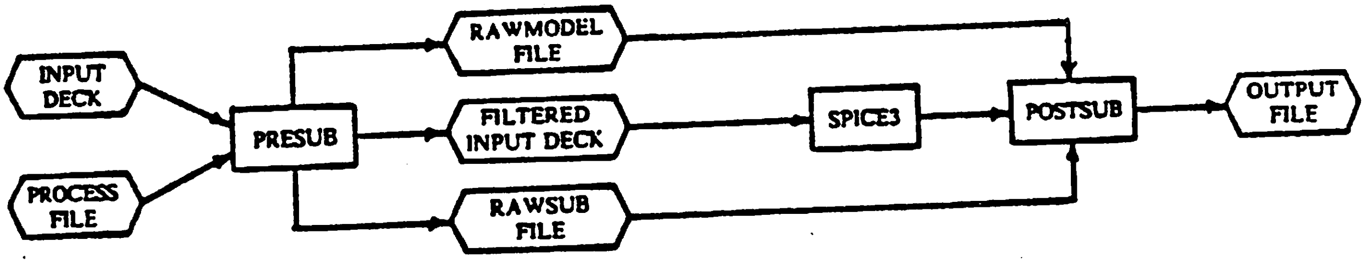


Fig. 5 BSIM3/SPICE2/SPICE3 processor configuration.

3.1 Structure of the Pre-Processor:

The program name for the pre-processor is "prescalp.c". All global variables used by this program are declared in "BSIMpredefs.h". Other associated program names needed for compilation are "ACdefs.h", "CKTdefs.h", "FTEconstant.h", "FTEwritedata.h", "GENdefs.h", "INPdefs.h", "NIdefs.h", "OUTinterface.h", "Proc2ModSub.c", "SMPdefs.h", "SPerror.h", "TRCVdefs.h", "prefix.h", "suffix.h", "util.h", and "version.h". A flowchart showing the structure of the pre-processor is given in Fig. 6. The program "prescalp.c" contains six main routines. The functions of these main routines are described below:

(1) *PreFilter*:

This routine scans the entire input deck to determine the number of transistors in the circuit, the number of process files given, and the number of transistors given in the .PRINT or .PLOT ISUB commands. This is done in order to allocate the proper amount of memory spaces for data storage. SCALP commands, with the exception of .ISUBWIDTH, are also read in from the input deck and stored in *rawsub*.

(2) *FindProc*:

The .PROCESS control line is searched for to obtain the process name and process file name. For each process file given, a corresponding model file is created which contains all the process parameters. The routine named "Proc2ModSub" stored in "Proc2ModSub.c" is called to do this. When BSIM2 is used, a new process file is created which is identical to the original process file except for the substrate and degradation parameters are commented out. This is done by calling "CreateRawprocess" stored in "Proc2ModSub.c". The filenames for the model file and the new process file are *RAWMODEL* 0 and *RAWPROC* 0 respectively. When more than one process file is given, *RAWPROC* 1, *RAWPROC* 2, etc., will be created. Since SPICE only recognizes the first eight characters of a given process filename, the new process filename becomes *RAWPRO* 10 when more than nine process files are

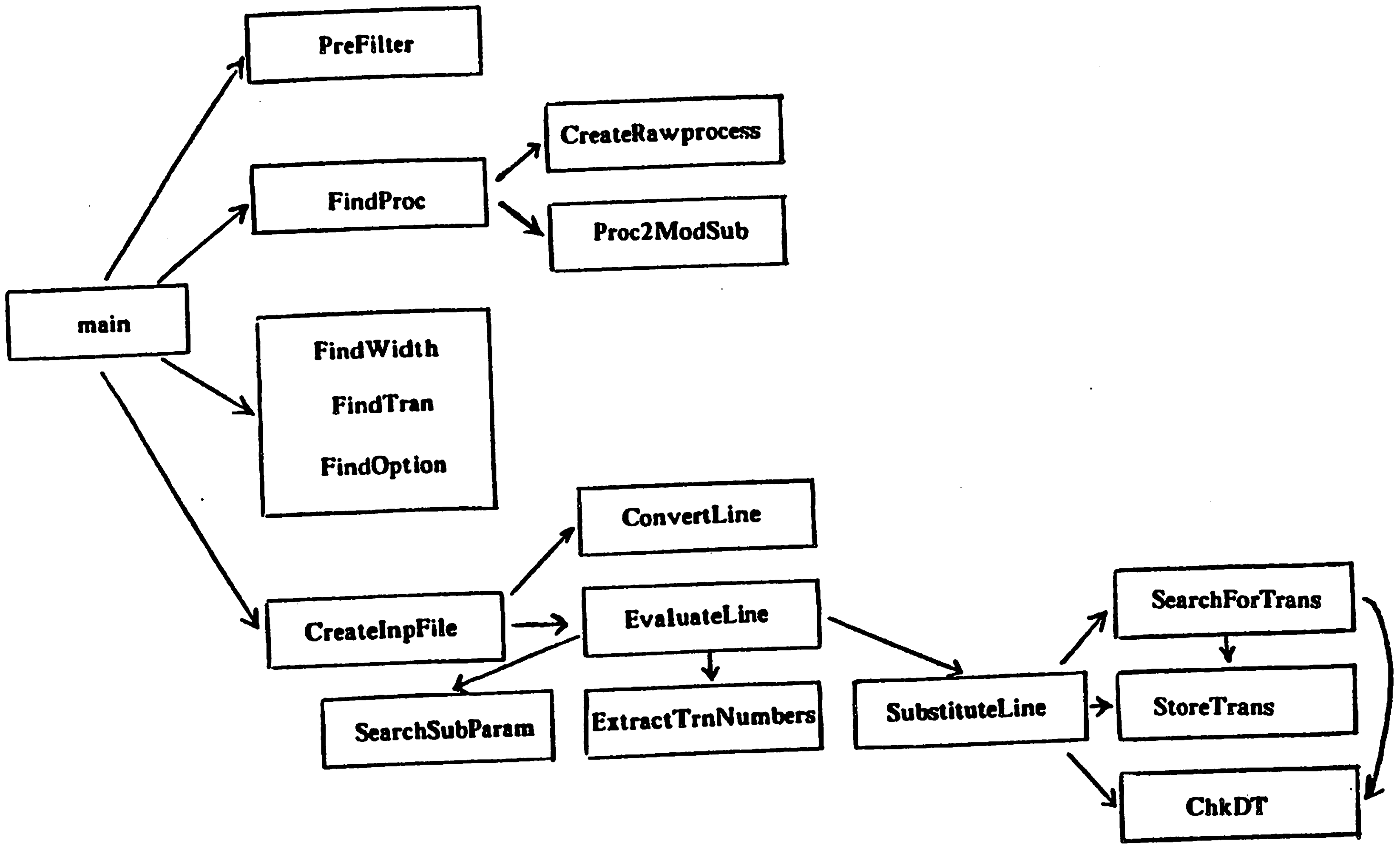


Fig. 6 Basic structure of the SCALP pre-processor.

given. When more than 99 files are given, the new process filename becomes *RAWP*100. Thus, the maximum number of .PROCESS lines that can be given in an input deck is set to be 9999.

(3) *FindWidth* , *FindTran* , *FindOption* :

"FindWidth" and "FindTran" search for the .ISUBWIDTH and .TRAN control lines and store these information in *rawsub* . "FindOption" routine looks for the relevant information (i.e. DEFL, DEFW, NOMOD, etc.) that may appear in the .OPTIONS command.

(4) *CreateInpFile* :

The filtered input deck for SPICE is created in this routine. For BSIM2 users, the filename in the .PROCESS control line is changed to *RAWPROC#*; for BSIM3 users, the *RAWMODEL#* file is appended to the input deck, and the .PROCESS command is deleted by the pre-processor; for SPICE2 and SPICE3 users, the .PROCESS command is deleted. All of these are done to maintain compatibility with the different versions of the SPICE codes. Also, for BSIM/SPICE3 users, the .width SPICE output control line is set to 90. This is done to facilitate the data fetching process of the post-processor.

"CreateInpFile" delegates its job to two routines, "ConvertLine" and "EvaluateLine". "ConvertLine" deletes the non-SPICE compatible parameters contained in the MOSFET information line, and "EvaluateLine" is basically responsible for commenting out all non-SPICE-compatible commands. "EvaluateLine" also calls "ExtractTrnNumbers", "SubstituteLine", and "SearchSubParam". "ExtractTrnNumbers" extracts the transistor names from the .PRINT or .PLOT ISUB commands and stores this information in *rawsub* ; when BSIM3 is used, "SearchSubParam" is called to comment out the substrate current and degradation parameters in *RAWMODEL* files before appending them to the input deck. "SubstituteLine" is mainly responsible for adding to the input deck the .PRINT node voltages commands which correspond to the terminal nodes of the user-selected transistors. It also calls "SearchForTrans", "ChkDT", and "StoreTrans". "SearchForTrans" is very similar to "SubstituteLine"; it is called when the user wishes to know the lifetimes of all transistors in the circuit. "ChkDT" is used to check whether the diffusion type to be used for the source/drain junctions is given in the BSIM MOSFET's information line. If it is not, "ChkDT" gives it the default type, DU1. "StoreTrans" determines the size of the transistor, and stores all relevant information pertaining to the user-selected transistors in *rawsub*.

3.2 Structure of the Post-Processor:

A flowchart showing the structure of the post-processor is given in Fig. 7. The code for the post-processor is stored in "scalp.c", and all global variables are defined in "BSIMpostdefs.h". This program is divided into two main sections. The first part reads in all transistor information from the *rawsub* and the *RAWMODEL* files and stores them. The second part reads in the voltage values from SPICE, calculates the substrate current and device lifetime, and prints the output. The main functions from both parts will be discussed here.

(1) *ErrorCheck*:

This routine checks to see if there has been an error in pre-processing or if *rawsub* cannot be found.

(2) *ReadParameters*

This is one of the main routines in "scalp.c". The function of this routine is to obtain all the parameters from the *RAWMODEL* files. It first calls "FindInfo" to read all the user-given SCALP commands from *rawsub*. "ObtainTrans" is then called to read in all transistor information from *rawsub* and store them in the variable array, TrnArray. Finally, "ObtainModelCards" is called to allocate memory spaces for the model parameters.

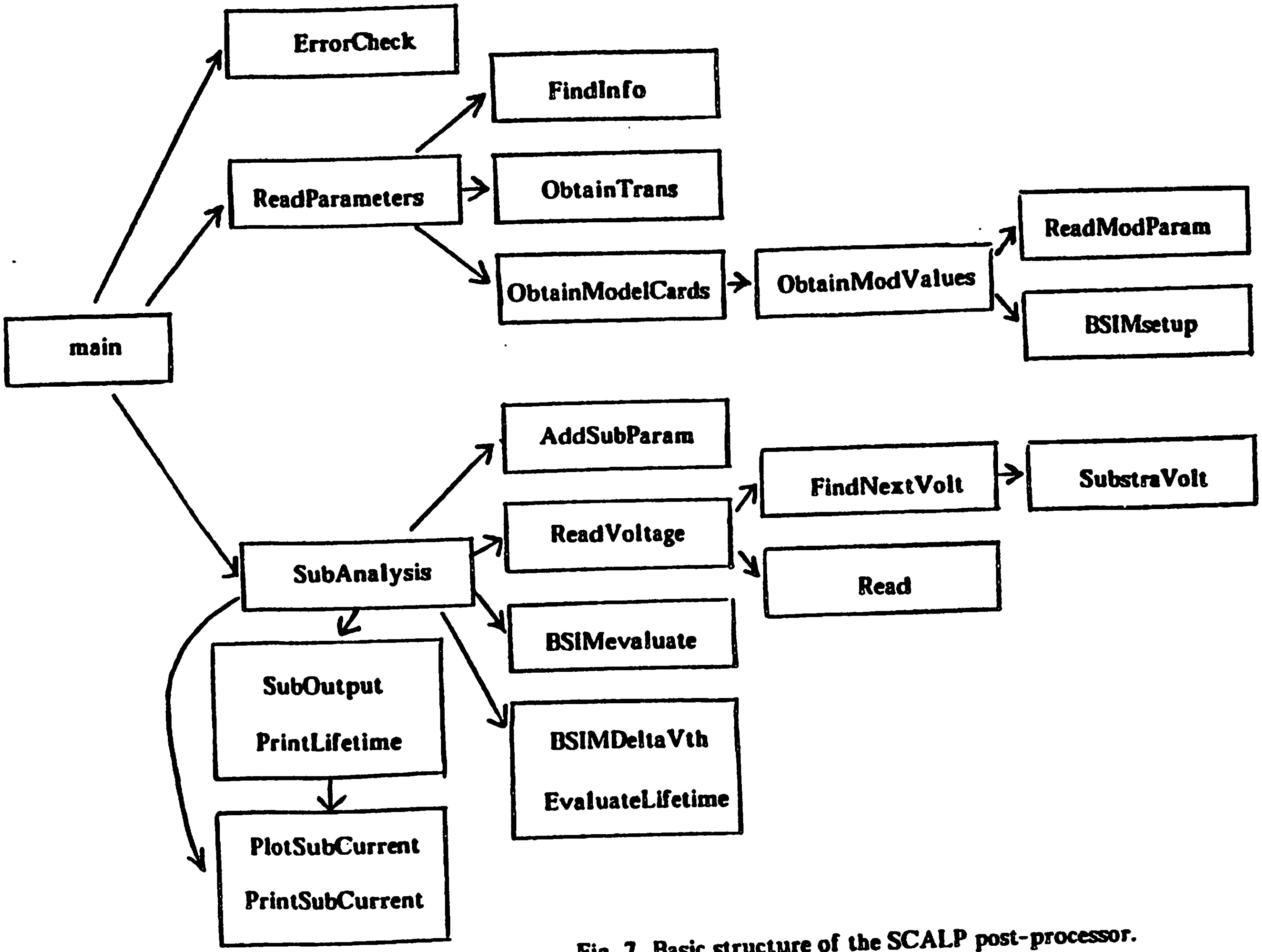


Fig. 7 Basic structure of the SCALP post-processor.

In "ObtainModelCards", "ObtainModValues" is called to determine the relevant data in the *RAWMODEL* files which need to be printed out. "ObtainModValues" calls "ReadModParam" to do the actual model parameters read-in, and calls "BSIMsetup" to convert the size-independent model parameters to the size dependent parameters for each transistor.

The second main routine in "scalp.c" is "SubAnalysis". This routine calls many functions; these functions will be described below.

(3) *AddSubParam*:

If either SPICE2 or BSIM2 is being run and the NOMOD option is not present in the .OPTIONS control line as determined by the pre-processor earlier, then this routine appends the substrate current and the degradation parameters to the model information section of the SPICE output.

(4) *ReadVoltage*:

This routine is used to search the right set of output voltages from SPICE for the user-selected transistors. This routine is dependent on the SPICE output format in its search and therefore may need modification if SPICE output format is changed. Presently, "ReadVoltage" searches for the key words 'TRANSIENT ANALYSIS' if BSIM/SPICE2 is used, and the words 'Transient Analysis' if BSIM/SPICE3 is used. It then calls "FindNextVolt" to search for the word 'TIME' if BSIM/SPICE2 is used, and the word 'Index' if BSIM/SPICE3 is used. "FindNextVolt" calls "SubstraVolt" to determine if the node numbers given in SPICE output match with those of the transistors specified in *rawsub*. If there is a match, "Read" is called from "ReadVoltage" to read in voltage values from SPICE output and to store them in the appropriate variable arrays.

(5) *BSIMevaluate*:

The BSIM substrate current model is implemented is this routine. Given all the process parameters and specified node voltages, this routine calculates the transient substrate current.

(6) *BSIMDeltaVth , EvaluateLifetime*:

Implementation of the degradation model is done in these two routines. Device lifetime is calculated based on the transient substrate current.

(5) *SubOutput , PrintLifetime*:

These two routines are responsible for the printing and plotting of the substrate current and the device lifetime of each individual device. "SubOutput" calls "PrintSubCurrent" and "PlotSubCurrent" when the applicable .PRINT ISUB or .PLOT ISUB command is given. "SubAnalysis" also calls "PrintSubCurrent" and "PlotSubCurrent" to print and plot the total substrate current flowing in the specified devices.

3.3 Programming limitations:

(1) Memory Limitation:

SCALP does not have a limitation on the size of the circuit (i.e. number of transistors in the circuit) to be simulated because it allocates memory spaces based on the circuit information described in the input deck. This means that in his simulation, the user is limited by the amount of available computer memory that he has.

(2) BSIM/SPICE3 Output Size:

The time step specified in the .tran control line used in BSIM/SPICE3 has no effect on the BSIM/SPICE3 output. BSIM/SPICE3 does internal interpolation based on the time interval specified in the .tran control line, and outputs the voltage value at each time point when this value differ from the previous value by a pre-determined amount. Thus SCALP has no way of knowing how much memory to allocate for the voltage data. To solve this problem. SCALP uses the information given in the .tran line, and approximates the amount of data output by BSIM/SPICE3. If too many data points are found in the output, SCALP will signal the user by outputting an error message.

(3) Data Size:

The precision of the data types used is very dependent on the machine used. When a variable exceeds its maximum or minimum size, the machine will stop the execution of its program. To avoid this problem, SCALP has internally set the absolute maximum size of a double-precision floating point variable to 1E+38, and the absolute minimum to 1E-38.

IV. USER'S GUIDE FOR THE PROCESSORS

Since the input format for SPICE3, SPICE2, BSIM2 and BSIM3 are all different, the user needs to pay special attention to the command syntax that is applicable to his use. The following new commands have been added for substrate current and degradation analysis:

(1) .PRINT ISUB, or .PLOT ISUB:

```
.PRINT  ISUB(MXXX)  <ISUB(MXXX) ... ISUB(MXXX) >
.PRINT  ISUB(SXXX)  <ISUB(SXXX) ... ISUB(SXXX) >
.PLOT  ISUB(MXXX)  <ISUB(MXXX) ... ISUB(MXXX) >  <(MIN,MAX) >
.PLOT  ISUB(SXXX)  <ISUB(SXXX) ... ISUB(SXXX) >  <(MIN,MAX) >
```

Examples:

```
.PLOT  ISUB(S1)  ISUB(S4)  (0,7E-6)
.PRINT  ISUB(M1)  ISUB(M4)  ISUB(M5)
```

These control lines are used to either print or plot out the substrate current. The device lifetime is always printed out following the output of the corresponding substrate current unless otherwise specified. The sums of the specified substrate current for the NMOS and PMOS devices are also automatically printed out. This is a good test to determine if the substrate-bias generator will be overloaded by the substrate current. SXXX is a syntax following the BSIM2 convention and should be used for both BSIM2 and SPICE2, while MXXX is used for SPICE3 and BSIM3. MIN and MAX are optional parameters specifying the minimum and maximum substrate current to be plotted.

(2) .ISUBALL

This command will enable the SCALP simulator to provide the sums of the substrate current for all MOS devices in the circuit regardless of transistors specified in the .PRINT or .PLOT ISUB command.

(3) .ISUBONLY

This command should be used if the user is not interested in the device lifetime calculation. When this command is given, only the substrate current information is provided by the SCALP simulator.

(4) .LIFE

When the user wishes to know the lifetimes of all devices in the circuit, this command can be given. This command overrides the .ISUBONLY command and can be used together with the .PRINT or .PLOT ISUB command.

(5) .ISUBWIDTH=COLWIDTH

Examples:

```
.ISUBWIDTH=80
.ISUBWIDTH=100
```

This command controls the width of the substrate current output printout. This is independent of the usual .WIDTH or .OPTIONS WIDTH command used in SPICE. Permissible values for COLWIDTH range from 80 to 200. The default value is 132. In SPICE3 or BSIM3, all non-substrate current analysis printout is outputted in 90 column format, regardless of the .OPTIONS WIDTH command. This was done to insure proper voltage read-in when none of the four nodes of the specified transistor is grounded.

(6) .DELTAVT VALUE

Examples:

```
.DELTAVT 10MV
.DELTAVT 900UV
```

This control line is used for setting the amount of threshold voltage shift defined at device failure. The calculation of device lifetime is based on this value when the parameters H, n, and m are given. If parameters B and m are given, this command is ignored.

(7) .TRAN TSTEP TSTOP <TSTART>

Examples:

```
.TRAN 1NS 100NS

.TRAN 5NS 1000NS 2NS
```

Since this simulator system is designed to calculate the transient substrate currents, the SPICE .TRAN command should always be included whenever this system is used. In order for the calculated device lifetime to be meaningful, the difference between TSTOP and TSTART should equal to a multiple of the period of the input signal.

(8) .PROCESS PNAME FILENAME=FNAME

Examples:

```
.PROCESS PC1 FILENAME=RUN

.PROCESS MK1 FILENAME=MMK
```

This control line specifies the process name and the corresponding process file name which contains all the process parameters. The format is identical to that already implemented in BSIM2, but it is new for SPICE2, SPICE3 and BSIM3. It is important to realize that ".MODEL" commands are no longer necessary for BSIM3, but that a ".PROCESS" command is now mandatory.

(9) General form for MOSFETs:

```
SXXXXXXX ND NG NS NB <MNAME> PNAME_MT_DT <L=VAL> <W=VAL> ... etc.

MXXXXXXX ND NG NS NB <MNAME> PNAME_MT_DT <L=VAL> <W=VAL> ... etc.
```

Examples:

```
S1 1 2 3 4  MODN  PC1_NM1_DU1  L=1U  W=20U

S1 1 2 3 4  PC1_NM1_DU1  L=1U  W=20U

M1 1 2 3 4  MODP  PC3_PM1_DU2  L=10U  W=5U  AD=100P  AS=100P

+ PD=40U  PS=40U
```

To describe a MOSFET, the user should use SXXXXXXX for BSIM2 or SPICE2, MXXXXXXX for BSIM3 or SPICE3. MNAME is the model name which should be given only if SPICE2 or SPICE3 is used. PNAME_MT_DT should be given regardless of the type of model used in the SPICE analysis. PNAME is the process name which must be specified in a .PROCESS command. MT is the mos type. The possible choices are NM1 to NM5, and PM1 to PM5. DT is the source/drain junction diffusion type. DU1 to DU3 ar the three available diffusion types with DU1 used as the default value. For users who are not familiar with SPICE commands, please consult the manual for SPICE. For users who wish to learn more about the BSIM model implemented in SPICE or about the BSIM extraction program, please refer to "SPICE IMPLEMENTATION OF BSIM"[6] or "BSIM PARAMETER EXTRACTION - ALGORITHMS AND USER'S GUIDE"[2].

In an UNIX environment, the commands for compilation are:

```
cc -o prescalp prescalp.c -lm
cc -o scalp scalp.c -lm
```

where "prescalp" and "scalp" are the names of the executable files and "prescalp.c" and "scalp.c" are the names of the programs. To execute the programs, the command is:

```
prescalp -x -y deck | spice | scalp > outfile
```

where x is "b" if the BSIM model is used in the SPICE analysis, or "o" if other model is used, y is either "2" or "3" depending on whether SPICE2/BSIM2 or SPICE3/BSIM3 is used, "deck" is the input deck file, "spice" is the name of the executable version of the

SPICE program the user is using, and "outfile" is the output file desired. If no "-x" option is specified, the BSIM model is assumed to be used; if no "-y" option is given, the simulator defaults to SPICE2/BSIM2.

V. RESULTS

5.1 A Sample SCALP Run

Figs. 8 and 9 illustrate sample BSIM2 and BSIM3 input decks respectively for a CMOS EEPROM sense amplifier shown in Fig. 10. M1 and M7 are EEPROM reference and memory cells, respectively, and M3 and M5 are the sense amplifier's NMOS devices of which substrate current simulations are requested. The main difference between the two input decks is the BSIM transistor designations (SXXX for BSIM2, MXXX for BSIM3). Figs. 11 and 12 show the resulting output of M3 and M5 for BSIM2 with a 5 volt input pulse of 4ns rise and fall time, and 25ns pulse width applied at $V_{in} = V(6)$ at $t = 0$ns. Output for BSIM3 is virtually identical to the one shown here. Fig. 13 shows the total amount of substrate current flowing in M3 and M5 as a function of the specified time interval. As one can see from Fig. 12, an improperly designed MOSFET device under normal circuit operating condition can be degraded in such a way as to reduce its lifetime to less than 10 years.

5.2 Effect of .TRAN Control Line on Lifetime Simulation

Since the device lifetime is simulated based on a quasi-static model, one of the factors that affects the accuracy of the lifetime simulation is the number of time points implied in the .TRAN line. A 5 volt input pulse of 4ns rise and fall time and a 4ns pulse width was applied at the gate of an NMOS transistor while the drain voltage remains at 5V, and two separate runs with different time steps were simulated. The results in Fig. 14 showed that the lifetime can differ by 3 times or more when insufficient transient data points were used in the simulation of the lifetime.

```
SIMPLE CMOS SENSE AMPLIFIER
s1 2 5 0 0 PC1_nm1_du1 w=3u l=2u
s2 2 2 5 5 PC1_pm1_du2 w=140u l=2u
s3 3 2 1 0 PC1_nm1_du1 w=20u l=2u
s4 3 2 5 5 PC1_pm1_du2 w=60u l=2u
s5 4 1 0 0 PC1_nm1_du1 w=5u l=2u
s6 4 3 5 5 PC1_pm1_du2 w=10u l=2u
s7 1 6 0 0 PC1_nm1_du1 w=3u l=2u
vdd 5 0 5 dc
vin 6 0 pulse(0.5.0ns.4ns.4ns.25ns.58ns)
.process PC1 filename=PF.SUB
.plot isub(s3) isub(s5)
.deltavt 10mV
c1 1 0 2pf
c2 4 0 0.2pf
.tran 1ns 58ns
.isubwidth=80
.width in=80 out=80
.end
```

Fig. 8 Sample BSIM2 input deck.

```
SIMPLE CMOS SENSE AMPLIFIER
m1 2 5 0 0 PC1_nm1_du1 w=3u l=2u
m2 2 2 5 5 PC1_pm1_du2 w=140u l=2u
m3 3 2 1 0 PC1_nm1_du1 w=20u l=2u
m4 3 2 5 5 PC1_pm1_du2 w=60u l=2u
m5 4 1 0 0 PC1_nm1_du1 w=5u l=2u
m6 4 3 5 5 PC1_pm1_du2 w=10u l=2u
m7 1 6 0 0 PC1_nm1_du1 w=3u l=2u
vdd 5 0 5 dc
vin 6 0 pulse(0.5.0ns.4ns.4ns.25ns.58ns)
.process PC1 filename=PF.SUB
.plot isub(m3) isub(m5)
.deltavt 10mV
c1 1 0 2pf
c2 4 0 0.2pf
.tran 1ns 58ns
.isubwidth=80
.end
```

Fig. 9 Sample BSIM3 input deck.

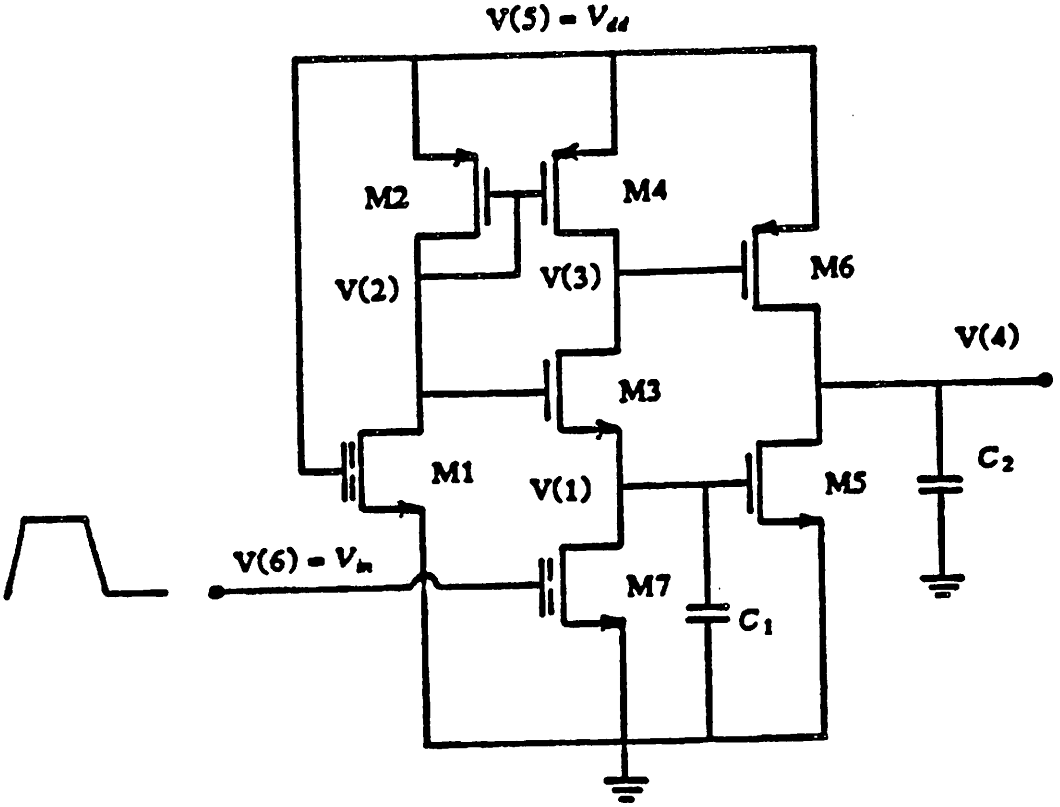
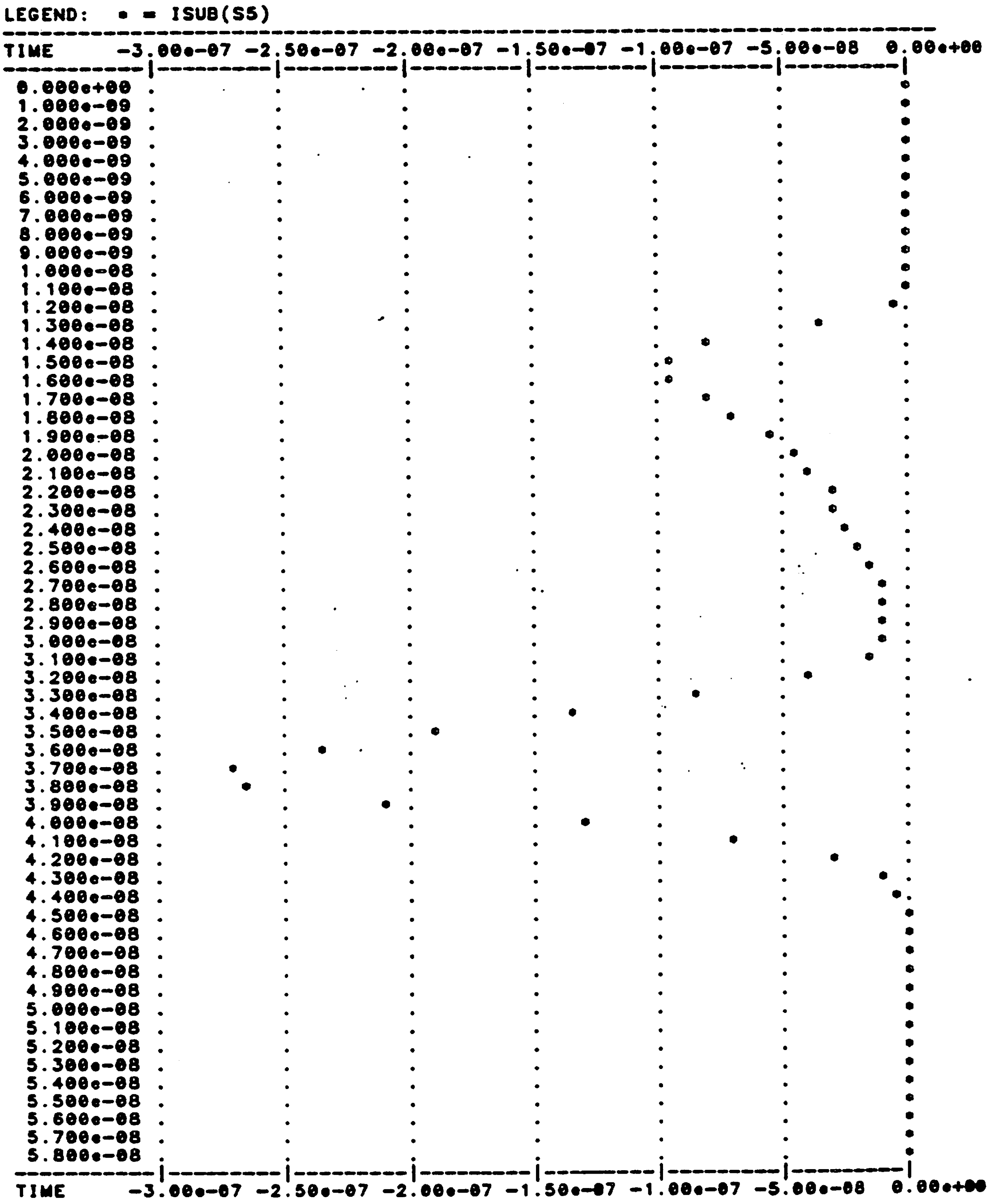


Fig. 10 CMOS sense amplifier circuit used for sample SPICE simulations.

```
                    SUBSTRATE CURRENT TRANSIENT ANALYSIS
                               TRANSISTOR S3

LEGEND:  • = ISUB(S3)
---------------------------------------------------------------------------
TIME     -3.00e-08 -2.50e-08 -2.00e-08 -1.50e-08 -1.00e-08 -5.00e-09  0.00e+00
-----------|---------|---------|---------|---------|---------|---------|
 0.000e+00 .         .         .         .         .         .   •     .
 1.000e-09 .         .         .         .         .         .   •     .
 2.000e-09 .         .         .         .         .         . •       .
 3.000e-09 .         .         .         .         . •       .         .
 4.000e-09 .         .         .  •      .         .         .         .
 5.000e-09 .        •.         .         .         .         .         .
 6.000e-09 .         .         . •       .         .         .         .
 7.000e-09 .         .         .         .         .         .       • .
 8.000e-09 .         .         .         .         .         .         •
 9.000e-09 .         .         .         .         .         .         •
 1.000e-08 .         .         .         .         .         .         •
 1.100e-08 .         .         .         .         .         .         •
 1.200e-08 .         .         .         .         .         .         •
 1.300e-08 .         .         .         .         .         .         •
 1.400e-08 .         .         .         .         .         .         •
 1.500e-08 .         .         .         .         .         .         •
 1.600e-08 .         .         .         .         .         .         •
 1.700e-08 .         .         .         .         .         .         •
 1.800e-08 .         .         .         .         .         .         •
 1.900e-08 .         .         .         .         .         .         •
 2.000e-08 .         .         .         .         .         .         •
 2.100e-08 .         .         .         .         .         .         •
 2.200e-08 .         .         .         .         .         .         •
 2.300e-08 .         .         .         .         .         .         •
 2.400e-08 .         .         .         .         .         .         •
 2.500e-08 .         .         .         .         .         .         •
 2.600e-08 .         .         .         .         .         .         •
 2.700e-08 .         .         .         .         .         .         •
 2.800e-08 .         .         .         .         .         .         •
 2.900e-08 .         .         .         .         .         .         •
 3.000e-08 .         .         .         .         .         .         •
 3.100e-08 .         .         .         .         .         .         •
 3.200e-08 .         .         .         .         .         .         •
 3.300e-08 .         .         .         .         .         .         •
 3.400e-08 .         .         .         .         .         .         •
 3.500e-08 .         .         .         .         .         .         •
 3.600e-08 .         .         .         .         .         .         •
 3.700e-08 .         .         .         .         .         .         •
 3.800e-08 .         .         .         .         .         .         •
 3.900e-08 .         .         .     •   .         .         .         .
 4.000e-08 .         .         •         .         .         .         .
 4.100e-08 .         .        •.         .         .         .         .
 4.200e-08 .         .         .•        .         .         .         .
 4.300e-08 .         .         .  •      .         .         .         .
 4.400e-08 .         .         .    •    .         .         .         .
 4.500e-08 .         .         .       • .         .         .         .
 4.600e-08 .         .         .         •         .         .         .
 4.700e-08 .         .         .         . •       .         .         .
 4.800e-08 .         .         .         .   •     .         .         .
 4.900e-08 .         .         .         .     •   .         .         .
 5.000e-08 .         .         .         .      •  .         .         .
 5.100e-08 .         .         .         .       • .         .         .
 5.200e-08 .         .         .         .         •         .         .
 5.300e-08 .         .         .         .         .•        .         .
 5.400e-08 .         .         .         .         . •       .         .
 5.500e-08 .         .         .         .         . •       .         .
 5.600e-08 .         .         .         .         .  •      .         .
 5.700e-08 .         .         .         .         .   •     .         .
 5.800e-08 .         .         .         .         .   •     .         .
-----------|---------|---------|---------|---------|---------|---------|
TIME     -3.00e-08 -2.50e-08 -2.00e-08 -1.50e-08 -1.00e-08 -5.00e-09  0.00e+00

---------------------------------------------------------------------------
   DEVICE LIFETIME AT DELTA VTH = 0.01 VOLTS
---------------------------------------------------------------------------
   >>>>>> TAU(S3) = 3.42072e+10 HRS.  (1.23146e+14 SEC.) <<<<<<

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

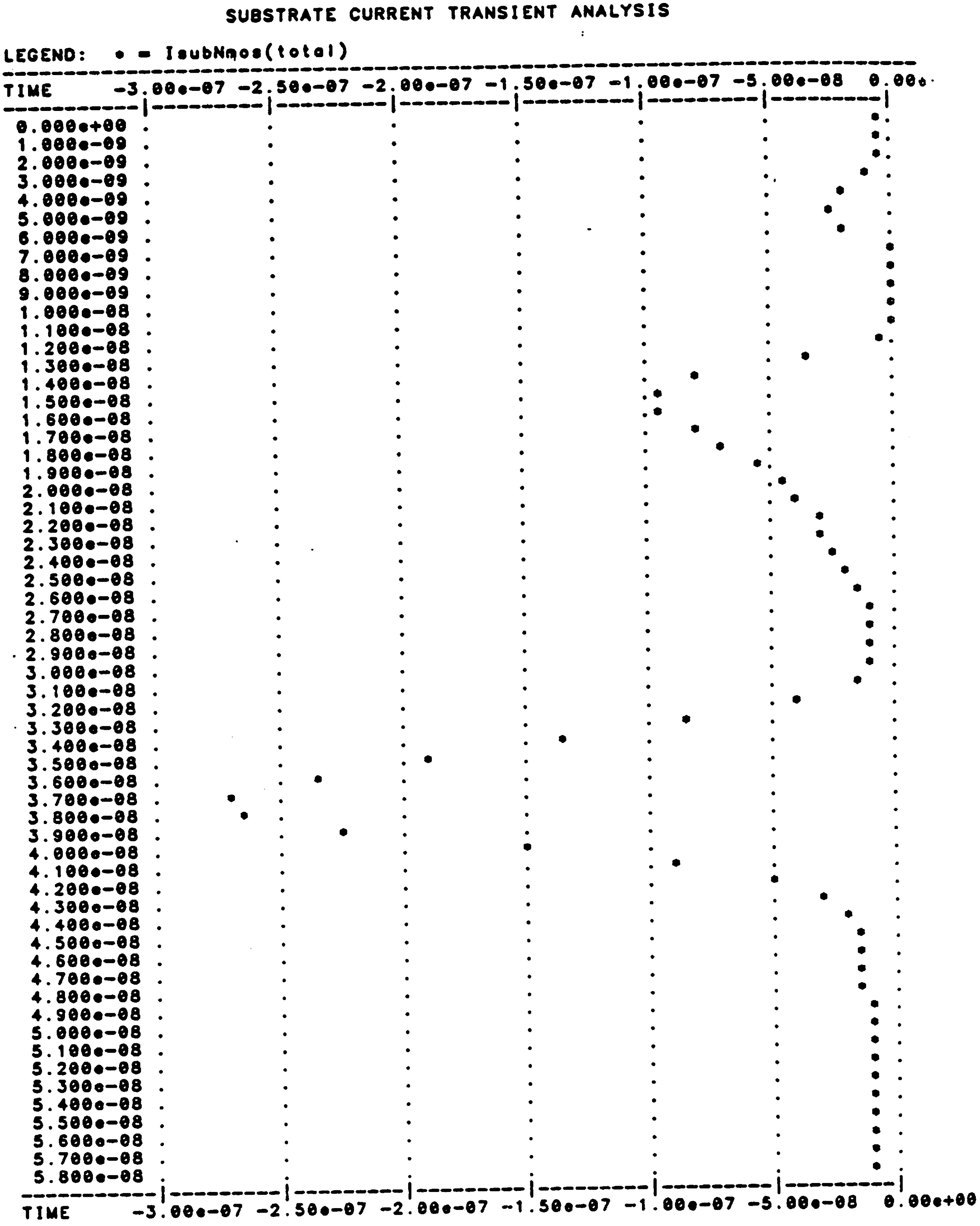
Fig. 11 BSIM2 output file

SUBSTRATE CURRENT TRANSIENT ANALYSIS
TRANSISTOR S5



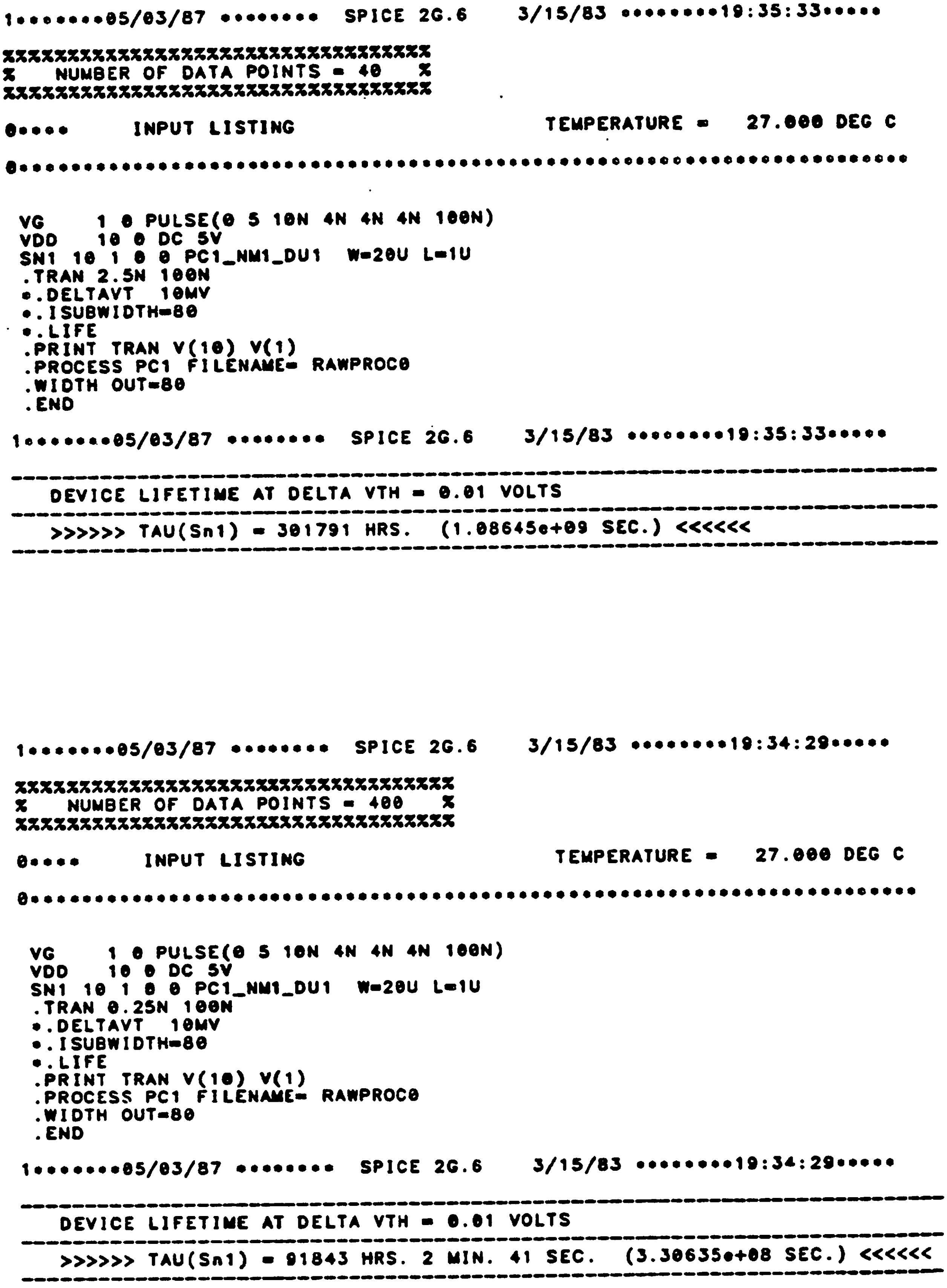
```
------------------------------------------------------------------------------
   DEVICE LIFETIME AT DELTA VTH = 0.01 VOLTS
------------------------------------------------------------------------------
   >>>>>> TAU(S5) = 84523 HRS. 56 MIN. 14 SEC.  (3.04286e+08 SEC.) <<<<<<
------------------------------------------------------------------------------
```

Fig. 12 BSIM2 output file



```
                 SUBSTRATE CURRENT TRANSIENT ANALYSIS

LEGEND:   * = IsubNmos(total)
------------------------------------------------------------------------------
TIME     -3.00e-07  -2.50e-07  -2.00e-07  -1.50e-07  -1.00e-07  -5.00e-08   0.00e
-----------|----------|----------|----------|----------|----------|----------|
 0.000e+00 .          .          .          .          .          .         *.
 1.000e-09 .          .          .          .          .          .         *.
 2.000e-09 .          .          .          .          .          .         *.
 3.000e-09 .          .          .          .          .          .        * .
 4.000e-09 .          .          .          .          .          .      *   .
 5.000e-09 .          .          .          .          .          .     *    .
 6.000e-09 .          .          .          .          .          .      *   .
 7.000e-09 .          .          .          .          .          .          *
 8.000e-09 .          .          .          .          .          .          *
 9.000e-09 .          .          .          .          .          .          *
 1.000e-08 .          .          .          .          .          .          *
 1.100e-08 .          .          .          .          .          .          *
 1.200e-08 .          .          .          .          .          .         *.
 1.300e-08 .          .          .          .          .          .  *       .
 1.400e-08 .          .          .          .          .   *      .          .
 1.500e-08 .          .          .          .          .*         .          .
 1.600e-08 .          .          .          .          .*         .          .
 1.700e-08 .          .          .          .          .   *      .          .
 1.800e-08 .          .          .          .          .     *    .          .
 1.900e-08 .          .          .          .          .         *.          .
 2.000e-08 .          .          .          .          .          .*         .
 2.100e-08 .          .          .          .          .          . *        .
 2.200e-08 .          .          .          .          .          .   *      .
 2.300e-08 .          .          .          .          .          .   *      .
 2.400e-08 .          .          .          .          .          .    *     .
 2.500e-08 .          .          .          .          .          .     *    .
 2.600e-08 .          .          .          .          .          .       *  .
 2.700e-08 .          .          .          .          .          .        * .
 2.800e-08 .          .          .          .          .          .        * .
 2.900e-08 .          .          .          .          .          .        * .
 3.000e-08 .          .          .          .          .          .        * .
 3.100e-08 .          .          .          .          .          .       *  .
 3.200e-08 .          .          .          .          .          . *        .
 3.300e-08 .          .          .          .          .  *       .          .
 3.400e-08 .          .          .          .  *       .          .          .
 3.500e-08 .          .          . *        .          .          .          .
 3.600e-08 .          .  *       .          .          .          .          .
 3.700e-08 .      *   .          .          .          .          .          .
 3.800e-08 .       *  .          .          .          .          .          .
 3.900e-08 .          .     *    .          .          .          .          .
 4.000e-08 .          .          .          *          .          .          .
 4.100e-08 .          .          .          .          . *        .          .
 4.200e-08 .          .          .          .          .          *          .
 4.300e-08 .          .          .          .          .          .   *      .
 4.400e-08 .          .          .          .          .          .      *   .
 4.500e-08 .          .          .          .          .          .       *  .
 4.600e-08 .          .          .          .          .          .       *  .
 4.700e-08 .          .          .          .          .          .       *  .
 4.800e-08 .          .          .          .          .          .       *  .
 4.900e-08 .          .          .          .          .          .        * .
 5.000e-08 .          .          .          .          .          .        * .
 5.100e-08 .          .          .          .          .          .        * .
 5.200e-08 .          .          .          .          .          .        * .
 5.300e-08 .          .          .          .          .          .        * .
 5.400e-08 .          .          .          .          .          .        * .
 5.500e-08 .          .          .          .          .          .        * .
 5.600e-08 .          .          .          .          .          .        * .
 5.700e-08 .          .          .          .          .          .        * .
 5.800e-08 .          .          .          .          .          .        * .
-----------|----------|----------|----------|----------|----------|----------|
TIME     -3.00e-07  -2.50e-07  -2.00e-07  -1.50e-07  -1.00e-07  -5.00e-08   0.00e+00
```

Fig. 13 BSIM2 output file



```
1•••••••05/03/87 ••••••••  SPICE 2G.6    3/15/83 ••••••••19:35:33•••••

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   NUMBER OF DATA POINTS = 40    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

0••••     INPUT LISTING                      TEMPERATURE =   27.000 DEG C

0••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••


 VG     1 0 PULSE(0 5 10N 4N 4N 4N 100N)
 VDD    10 0 DC 5V
 SN1 10 1 0 0 PC1_NM1_DU1  W=20U L=1U
 .TRAN 2.5N 100N
 •.DELTAVT  10MV
 •.ISUBWIDTH=80
 •.LIFE
 .PRINT TRAN V(10) V(1)
 .PROCESS PC1 FILENAME= RAWPROC0
 .WIDTH OUT=80
 .END
1•••••••05/03/87 ••••••••  SPICE 2G.6    3/15/83 ••••••••19:35:33•••••

------------------------------------------------------------------------
   DEVICE LIFETIME AT DELTA VTH = 0.01 VOLTS
------------------------------------------------------------------------
   >>>>>> TAU(Sn1) = 301791 HRS.  (1.08645e+09 SEC.) <<<<<<
------------------------------------------------------------------------
```

```
1•••••••05/03/87 ••••••••  SPICE 2G.6    3/15/83 ••••••••19:34:29•••••

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   NUMBER OF DATA POINTS = 400   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

0••••     INPUT LISTING                      TEMPERATURE =   27.000 DEG C

0••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••


 VG     1 0 PULSE(0 5 10N 4N 4N 4N 100N)
 VDD    10 0 DC 5V
 SN1 10 1 0 0 PC1_NM1_DU1  W=20U L=1U
 .TRAN 0.25N 100N
 •.DELTAVT  10MV
 •.ISUBWIDTH=80
 •.LIFE
 .PRINT TRAN V(10) V(1)
 .PROCESS PC1 FILENAME= RAWPROC0
 .WIDTH OUT=80
 .END
1•••••••05/03/87 ••••••••  SPICE 2G.6    3/15/83 ••••••••19:34:29•••••

------------------------------------------------------------------------
   DEVICE LIFETIME AT DELTA VTH = 0.01 VOLTS
------------------------------------------------------------------------
   >>>>>> TAU(Sn1) = 91843 HRS. 2 MIN. 41 SEC.  (3.30635e+08 SEC.) <<<<<<
------------------------------------------------------------------------
```

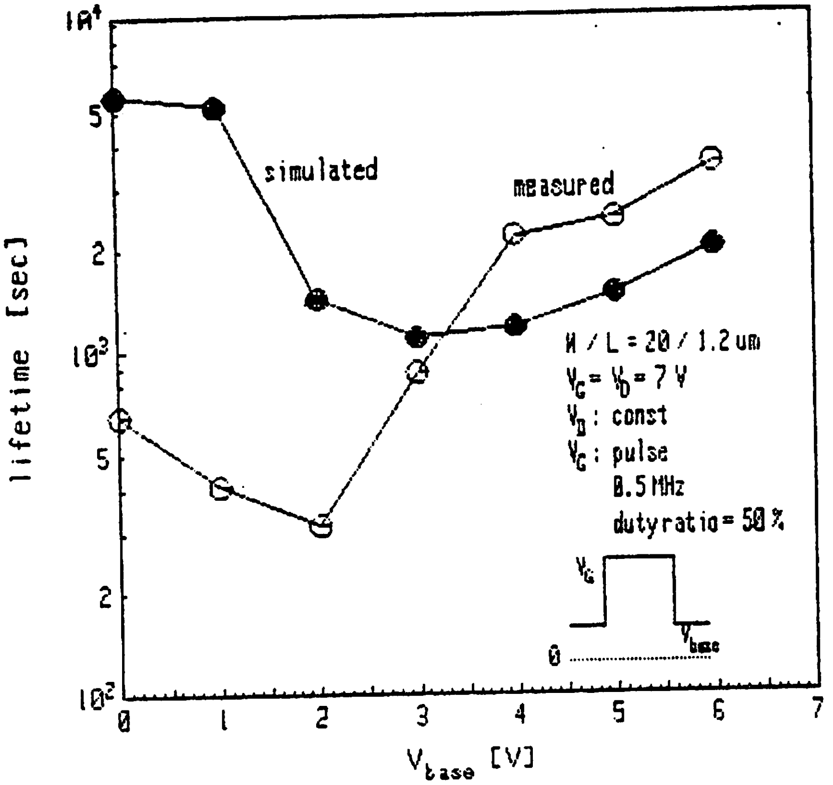
Fig. 14 Circuit simulation showing device lifetime using 40 and 400 transient data points.

5.3 Validity and Limitation of the Device Degradation Model

A major concern is the applicability of our quasi-static model and the use of DC test data for AC stressing lifetime prediction. It has been previously reported that AC stress can increase the degradation rate over DC stress by an order of magnitude [7,8]. However, recent results demonstrate the dependability of the ratio of AC and DC degradation rate on the waveforms used in the stressing experiment [5]. It was shown that the AC enhanced degradation occurs in the case of fast V_G transients from high to low while V_D remains high. This is attributed to an additional component of I_{SUB} during the falling edge of V_G which encourages hole injection and electron trap generation [5]. Under other pulsed conditions, the device lifetime can be predictable from the DC test data. Thus, our quasi-static model should be a good predictor of lifetime for normal inverter circuits.

In Fig. 15, we show the simulated and measured lifetimes plotted as a function of V_{base}, the base voltage. The device was stressed at a constant drain voltage of 7V, and a pulsed gate voltage of .5 MHz with a 4ns rise and fall time and a duty ratio of 50%. The gate voltage was pulsed from V_{base} to 7V. V_{base} was varied from 0 to 7V, and the life time at $\Delta V_{th} = 10$ mV was measured. The result shows that the largest error is introduced at small V_{base} when the enhanced AC degradation discussed above takes place. As V_{base} is increased, the effect of enhanced AC degradation diminishes, and thus the measured and our simulated results show good agreement.

A normal inverter operating condition is also simulated and compared to the measured data. In Fig. 16, only the rising gate pulse of an NMOS takes place when the drain voltage is high. This is similar to the operation of an inverter pull-down device because the output (drain) responds after the input (gate) changes. A gate voltage rise time of 4 ns and 100 ns are used, and the gate-drain voltage overlap time is varied. Fig. 16 shows the plot of lifetime versus t_{ov}. For $t_{ov} = 0$, the simulated lifetime for the gate voltage with a 100ns rise time is more than an order of magnitude less than that with a 4ns rise time. This indicates that a pulsed gate voltage with a fast rise time causes negligible degradation.



Vbase	measured	simulated	error(%)
3.0	870.60	1099.08	+26.24
4.0	2176.20	1176.86	-45.92
5.0	2444.40	1476.00	-39.62
6.0	3508.80	2014.00	-42.60

Fig. 15 Lifetime as a function of the V_{base} applied at the gate. The gate voltage is pulsed from V_{base} to 7V. Effect of enhanced AC degradation can be seen for small V_{base}.

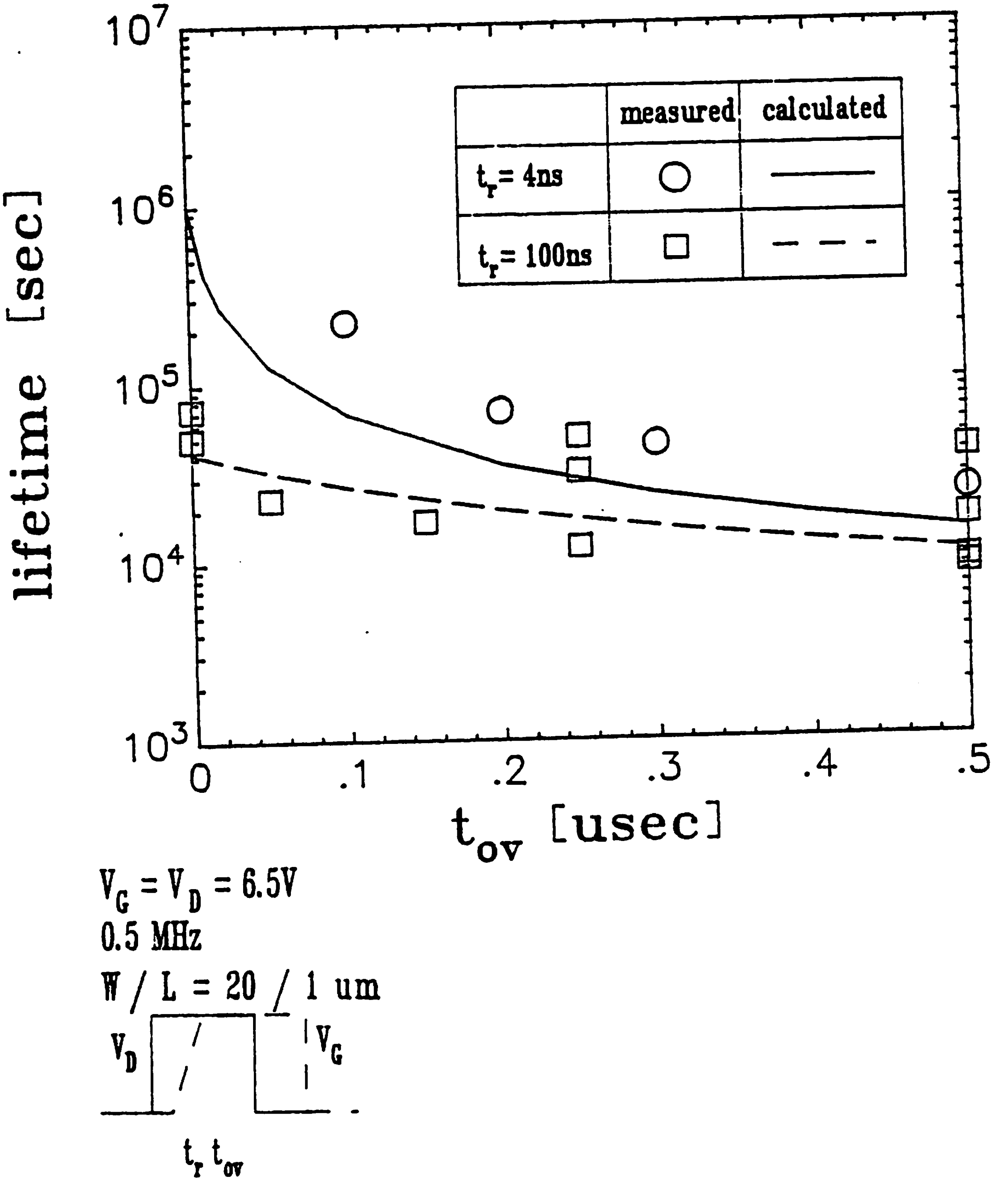


Fig. 16 Lifetime as a function of t_{ov}. The agreement between the AC stress test and our quasi-static model is very good.

Our simulated and measured data show very good agreement. In all cases, the difference between the measurement and simulation is less than a factor of five. This shows that our quasi-static model and the DC test data are indeed a good predictor of device lifetime under AC stressing.

One thing that we need to point out is the values of the degradation parameters we used to do the above simulations. Fig. 17 shows the log - log plot of $\tau * I_d$ versus $\frac{I_{SUB}}{I_d}$. Parameter m is the slope of this curve. The extracted m is 1.4 and B is 1.99E-3 A-sec/μm. Since the impact ionization energy is 1.3eV [1], it follows that the critical energy for device degradation, ϕ_{it}, is 1.82eV which is a lot lower than that physically possible. Thus, in this case, the extracted m seems to be more of an empirical parameter than that with a physical meaning.

Our results show good agreement between the AC stress test and our quasi-static model when parameters m and B are used in the lifetime calculation. However, because of the variability of n on bias voltages, lifetime simulation using parameters n, m, and H will not always give results as good as the ones obtained from parameters m and B.

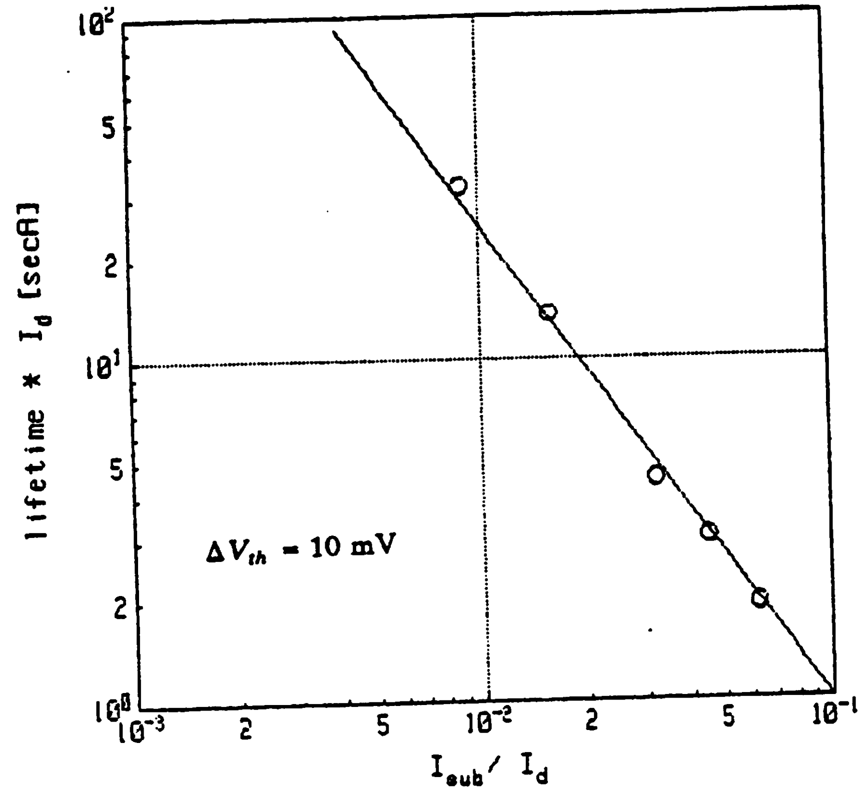


Fig. 17 Stress data used to calculate m and B for the simulation result shown in Fig. 15 and Fig. 16 are plotted in this log - log plot of $\tau^{*}I_d$ versus $\frac{I_{SUB}}{I_d}$.

VI. FUTURE ENHANCEMENT AND CONCLUSION

6.1 Future Extension:

The data used in the log - log plot of $\tau^* I_d$ versus $\frac{I_{SUB}}{I_d}$ of Fig. 17 were obtained by maintaining a constant drain voltage of 7V and varying the gate voltage from 3V to 7V. Evidence has shown that constant-voltage stress and the log - log plot of $\tau^* I_d$ versus $\frac{I_{SUB}}{I_d}$ can be used to extrapolate device lifetime from accelerated lifetime tests for a limited range of $V_G - V_D$ [9]. Since ϕ_{it} is reported to be dependent on the oxide field, this means m is also bias voltage dependent. We are in the process of developing a second degradation model which includes this bias dependence. Fig. 18 shows how the bias dependent m vary as a function of the oxide field, or $V_G - V_D$ in our proposed model. The simulation done in Fig. 16 is repeated by using this model, and the results is shown in Fig. 19. A very good fit is shown for the gate pulse with a 4ns rise time. Although the simulated result for the 100ns rise time case is not as good, the error introduced still is tolerable. Also the parameter m used in this simulation is physically meaningful as compared to the one used in Figs. 15 and 16. More work and tests are still being done on the development of this new model. When it is complete, it will be implemented in a second version of SCALP.

Another extension to SCALP under development at the present time is the prediction of MOS circuit performance degradation as a function of time. The degradation of BSIM process parameters as a function of time under the accelerated stress test will be investigated and the findings will be applied to the environment of normal circuit aging.

A third possible extension of SCALP is gate current modeling. This issue still needs further investigation before it can be carried out.

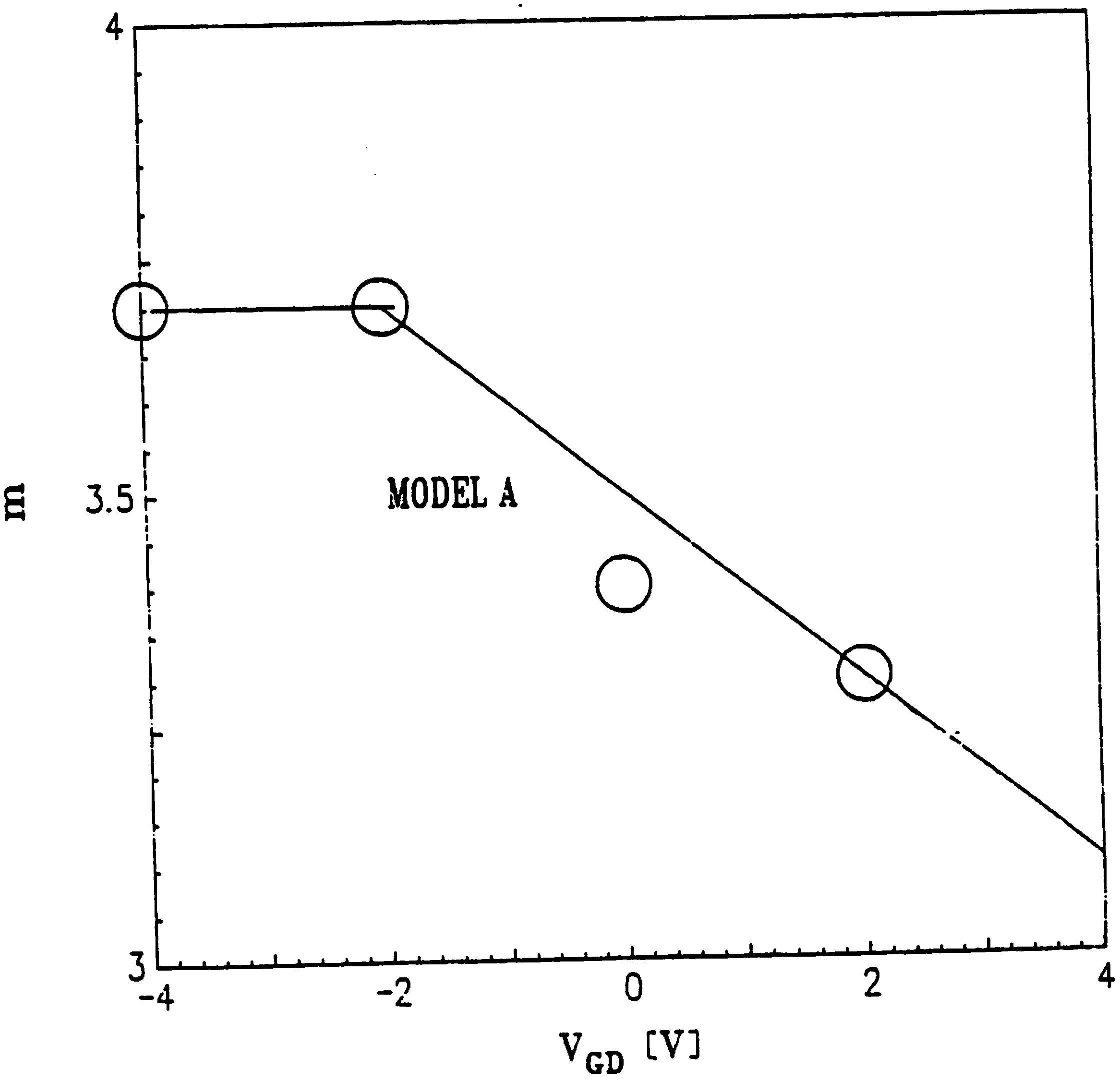


Fig. 18 Proposed model for m.

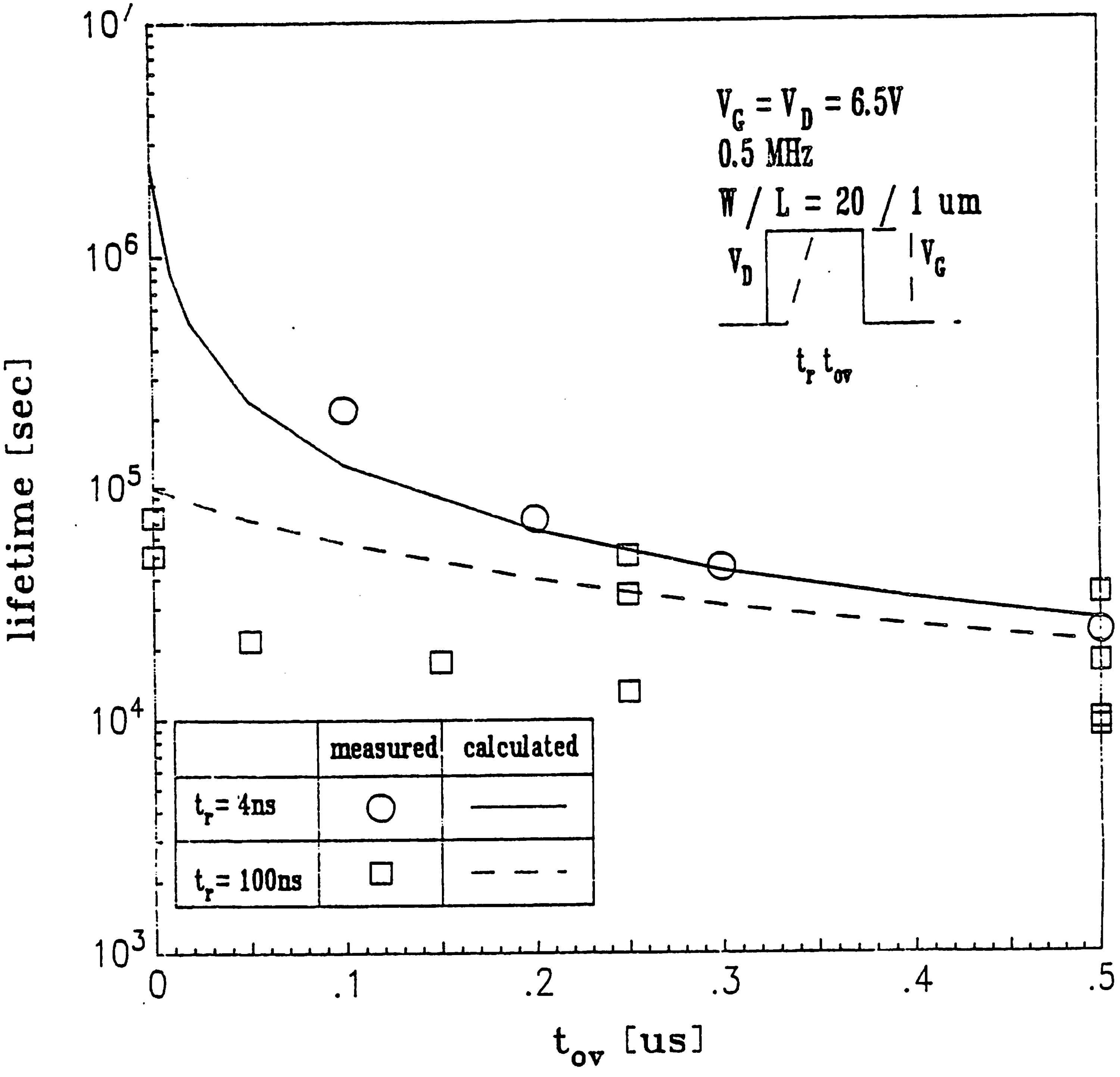


Fig. 19 Lifetime as a function of the overlap time of drain and gate pulses after the rising edge of the gate pulse.

6.2 Conclusion

In this report we have shown a simple way to evaluate substrate current and device lifetime by using SPICE and SCALP. With this tool, the circuit designer can now easily identify transistors that are most likely to suffer from hot-electron effects by comparing individual substrate currents and device lifetimes. Although this system predicts dynamic degradation based on DC data, it does provide a good first-order estimation of transistor reliability in a circuit environment.

REFERENCES

[1] C. Hu, S. Tam, F-C Hsu, P.K. Ko, T.Y. Chan, and K.W. Terrill, "Hot-Electron-Induced MOSFET Degradation - Model, Monitor, and Improvement," IEEE Trans. Elec. Dev., Vol. ED-32, p. 375, Feb. 1985.

[2] M.C. Jeng, B.J. Sheu, and P.K. Ko, "BSIM Parameter Extraction - Algorithms and User's Guide," University of California, Berkeley, ERL Memo. UCB/ERL M85/79.

[3] P.M. Lee, "BSIM - Substrate Current Modeling," University of California, Berkeley, ERL Memo, UCB/ERL M86/49.

[4] K.R. Hoffmann, W. Weber, C. Werner, and G. Dorda, "Hot Carrier Degradation Mechanism in N-MOSFETS," in IEDM Tech. Dig., p. 104, 1984.

[5] J.Y. Choi, P.K. Ko, and C. Hu, "Hot-Carrier-Induced MOSFET Degradation: AC Versus DC Stressing," to be published in Proc. VLSI Tech. Symp., 1987.

[6] B.J. Sheu, D.L. Scharfetter, and P.K. Ko, "SPICE2 Implementation of BSIM," University of California, Berkeley, ERL Memo, UCB/ERL M85/42.

[7] K.-L. Chen, S. Saller, and R. Shah, "The Case of AC Stress in the Hot-Carrier Effect," IEEE Trans. Elec. Dev., Vol. ED-33, p. 424, Mar. 1986.

[8] W. Weber, C. Werner, and G. Dorda, "Degradation of N-MOS-Transistors After

Pulsed Stress." IEEE Elec. Dev. Lett., Vol. EDL-5, p. 518, Dec. 1984.

[9] J.Y. Choi, P.K. Ko, and C. Hu, "Effect of Oxide Field on Hot-Carrier-Induced Degradation of Metal-Oxide-Semiconductor Field-Effect Transistors." Appl. Phys. Lett., Vol. 50 (17), p. 1188, Apr. 1987.