BERT - CIRCUIT AGING SIMULATOR (CAS)

by

Peter M. Lee, Mary M. Kuo, Ping K. Ko, and Chenming Hu

Memorandum No. UCB/ERL M90/2

18 January 1990

BERT - CIRCUIT AGING SIMULATOR (CAS)

by

Peter M. Lee, Mary M. Kuo, Ping K. Ko, and Chenming Hu

Memorandum No. UCB/ERL M90/2

18 January 1990

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

BERT - CIRCUIT AGING SIMULATOR (CAS)

by

Peter M. Lee, Mary M. Kuo, Ping K. Ko, and Chenming Hu

Memorandum No. UCB/ERL M90/2

18 January 1990

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

BERT - CIRCUIT AGING SIMULATOR (CAS)

Peter M. Lee, Mary M. Kuo, Ping K. Ko, and Chenming Hu

Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
Berkeley, CA 94720

Abstract

A Circuit Aging Simulator (CAS) has been developed as part of the BErkeley Reliability Tools (BERT) to predict the effects of hot-electron degradation on circuit behavior. Using the SPICE2 or SPICE3 circuit simulator, CAS simulates circuit behavior at a user-specified future time point using fresh and pre-stressed model parameter files. CAS is configured in a pre- and post-processor configuration so that no modifications to the SPICE code is necessary. An accompanying UNIX shell script has been developed for user-friendliness and automation so that iterative simulation can easily be done to take into account the effects of on-going degradation.

This work was supported by the Semiconductor Research Corporation

Table of Contents

I. Introduction 1
II. NMOSFET Device Degradation Model 4
2.1 Introduction 4
2.2 Model Equations and Implementation 4
2.3 Substrate Current Model 7
2.4 New Degradation Parameters 8
2.5 Device Stressing Methodology 9
2.6 Enhanced AC Degradation 10
2.7 Summary 11
III. NMOSFET Circuit Aging Model 12
3.1 Introduction 12
3.2 Model Formulation 12
3.3 Summary 14
IV. PMOSFET Degradation and Aging Models 16
4.1 Introduction 16
4.2 Gate Current Model 16
4.3 Degradation and Aging Model Based on I_{gate} 17
4.4 Incorporation of I_{sub} and I_{gate} in Predicting Degradation 18
4.5 Parameters Necessary for Simulation 19
4.6 Summary 20
V. BERT Configuration and Operation 21
5.1 Introduction 21
5.2 System Configuration 21
5.3 Installing and Running BERT 21
5.4 CAS Command Summary 22
5.5 Circuit Example: 21-Stage CMOS Inverter Chain 30
5.6 Restrictions 31
5.7 Summary 31
VI. BERT Shell Script Program for UNIX Environments 40
VII. Conclusion 47
VIII. References 48
Appendix: CAS Error Messages 51

I. INTRODUCTION

Hot-electron degradation is becoming an increasingly worrisome issue as device dimensions continue to shrink while the power supply voltage remains constant for compatibility. The high electric fields within these smaller devices accelerate the degradation process, causing device characteristics to change. Commonly-used parameters to quantify the amount of device degradation that has occurred are the drain current degradation $\Delta I_{ds}/I_{ds0}$, transconductance degradation $\Delta g_m/g_{m0}$, and threshold voltage shift ΔV_{th}. Drain current and transconductance are normally measured at V_{gs} = 5 V and V_{ds} = 50 mV or 100 mV, while V_{th} is usually measured at a pre-determined drain current level at the same drain biases. All three of these parameters are directly correlated with the substrate current I_{sub} that is generated. By measuring the substrate current generated by a device in a circuit, $\Delta I_{ds}/I_{ds0}$, $\Delta g_m/g_{m0}$, or ΔV_{th} of the device can be calculated, or conversely, the "lifetime" (the time it takes for the device to degrade a pre-determined amount) can be found.

Thus, the first phase of this project was to implement a parametric substrate current model in the BSIM1 (Berkeley Short-channel Igfet Model, Version 1.0) parameter extraction program. The model was then combined with a device degradation model to relate the substrate current to the device-level degradation parameters $\Delta I_{ds}/I_{ds0}$, $\Delta g_m/g_{m0}$, and ΔV_{th}. This combined model was then implemented for use in circuit simulation in a pre- and post-processor configuration to SPICE and was dubbed the Substrate Current And Lifetime Evaluator (SCALE)*. Details of the models implemented and information of the BSIM1 extraction program are given in [Jen87,Kuo87a].

Although SCALE successfully calculates the lifetime of each device within a circuit based on any of the three degradation parameters, the relationship between individual device degradation and circuit degradation as a whole remains ambiguous. [Aur87] has already demonstrated that not all transistors affect circuit behavior equally. For example, a transistor M1 may degrade much more severely than a transistor M2, but circuit performance may depend more on transistor M2. The sensitivity of this dependence may also change depending on what characteristic of the circuit is studied. Setting a sim-

*Early versions were called SCALP (Substrate Current And Lifetime Processor)

ple guideline, such as setting device lifetime at $\Delta I_{ds}/I_{ds0}$ = 10%, is impossible and often misleading when applied generally. It is thus imperative that a simulator be developed that can predict the degradation of each transistor while operating in a circuit environment for a user-definable length of time, and then simulate the entire circuit using the newly-found degraded behavior of each device in the circuit. Direct and meaningful information can then be obtained since raw circuit behavior is simulated.

Our Circuit Aging Simulator (CAS), as part of the BErkeley Reliability Tools (BERT) system, has been developed specifically to satisfy the need of being able to simulate circuits undergoing dynamic degradation for a user-definable length of time [Lee88]. CAS incorporates the structure and models of SCALE: 1) the system is configured in a pre- and post-processor fashion external to SPICE so that minimal setup time is required; and 2) transient substrate current waveforms and device lifetimes can still be calculated. In fact, SCALE has become a wholly enclosed subset of CAS, so that SCALE commands will also work in CAS. The degradation that each device experiences is still found by calculating the substrate current waveform during normal circuit operation. A new quantity, *Age*, is introduced to quantify the amount of degradation each device suffers. The Age includes extracted degradation parameters, the device width, and the substrate and drain currents. Device parameters corresponding to the user-specified future time point are then calculated by comparing the ages calculated for the circuit devices to that of several pre-stressed devices of which model parameter extractions were done. The newly created "aged" model parameters files are then used to simulate the circuit. In these simulations (as well as in the device lifetime simulations), two assumptions are made : 1) the SPICE analysis must be a transient analysis since aging is based on time; and 2) circuit behavior is assumed to be periodic with the period equal to the length of the SPICE analysis (i.e., the waveforms of the input, output, and all internal voltage nodes are assumed to repeat the pattern simulated in the SPICE run up to the user-defined future time point).

Using the pre- and post-processor directly allows maximum flexibility, but for added convenience and automation, a shell script program has been developed for the UNIX environment. The menu-driven shell program eliminates the use of long UNIX piping and re-direction commands necessary when the three simulators (pre-processor, post-processor, and SPICE) are used together. An added

option enables iterative aging simulation so that ongoing circuit degradation can be taken into account.

CAS has been configured to use SPICE level 1, 2, and 3 models as well as the BSIM1 model (level 4 model in SPICE3)*. Any mixture of the models can be used in the SPICE input deck.

Presently, verification has been obtained for the dominant NMOSFET degradation [Lee90]. Although research is still incomplete concerning PMOSFET degradation models, a preliminary version has been incorporated. More complete models will be included in future versions of CAS once they are available.

Section II and III of this report will outline the NMOSFET device lifetime model and aging models, respectively, while Section IV will describe a preliminary degradation and aging model for PMOSFET devices. Section V includes information on system configuration, program installation, and program usage (including a command summary and simulation example), and Section VI describes the UNIX shell script which enhances the capabilities and convenience of running BERT. Finally, the report concludes with a conclusion, literature references, and a list of CAS error messages that may be encountered during simulations.

The other modules of BERT consist of the Circuit Oxide Reliability Simulator (CORS) and an electromigration simulator. Information concerning these reliability simulators are contained in the companion BERT manuals [Ros90] and [Lie90a] respectively, as well as in the conference papers [Ros89] and [Lie90b].

This report assumes that the reader is familiar with the basic concepts of the SPICE circuit simulator. The reader is encouraged to consult the SPICE user's manuals for more information. [Jen87] and [Lee86] are also recommended for more information concerning the BSIM1 model including substrate current paramcter extraction methodology.

*To use the BSIM1 model with SPICE2 requires a special version of SPICE2G.6.

II. NMOSFET DEVICE DEGRADATION MODEL

2.1 Introduction

This section outlines the models used to calculate dynamic NMOSFET device degradation. This model implementation differs from that in SCALE described in [Jen87] in that a bias-dependent n parameter can now be used. A more detailed background of the model is given in [Hu85]. Because of the various stressing methodologies that exist, the recommended methodology for degradation parameter extraction for CAS is specified. Finally, AC enhanced degradation and how it affects CAS results are discussed.

2.2 Model Equations and Implementation

Device degradation is typically measured by the amount of drain current degradation $\Delta I_{ds}/I_{ds0}$, transconductance degradation $\Delta g_m/g_{m0}$, or threshold voltage shift ΔV_{th} that occurs. They all exhibit the same power law behavior with respect to time. Here we will generalize the degradation by using the symbol ΔD. ΔD may be interchangeably replaced by any of the three degradation parameters in the following equations.

Under DC static stressing conditions, the amount of degradation as a function of time is given by [Hu85]

$$\Delta D = At^n \tag{2.1}$$

where

$$A = C_1 \left[\frac{I_{ds}}{W} \exp(-\phi_{it}/q\lambda E_m) \right]^n \tag{2.2}$$

where ϕ_{it} is the critical energy required for the creation of interface traps, λ is the electron mean free path, E_m is the maximum lateral channel field, W is the device width, and n and C_1 are dependent on the processing technology. Also from [Hu85],

$$\frac{I_{sub}}{I_{ds}} = C_2 \exp(-\phi_i/q\lambda E_m) \tag{2.3}$$

where ϕ_i is the critical energy required for impact ionization and C_2 is a process technology constant. Eq. 2.3 can be re-arranged in the following manner:

$$\exp(-\phi_{it}/q\lambda E_m) = \left[\exp(-\phi_i/q\lambda E_m)\right]^{\frac{\phi_{it}}{\phi_i}} = \left[\frac{I_{sub}}{C_2 I_{ds}}\right]^m \tag{2.4}$$

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

By substituting the exponential term in Eq. 2.2 with Eq. 2.4 and merging all constants into the parameter H, we can obtain

$$A = \left[\frac{I_{ds}}{WH}\left[\frac{I_{sub}}{I_{ds}}\right]^m\right]^n \tag{2.5}$$

where n , m and H are extracted parameters and are dependent on device processing technology. The degradation parameters m and H are also dependent on the gate-drain bias voltage V_{gd} [Kuo87b,Kuo88]. Thus, the expression for device degradation from Eq. 2.1 becomes

$$\Delta D = \left[\frac{I_{ds}}{WH}\left[\frac{I_{sub}}{I_{ds}}\right]^m\right]^n t^n \tag{2.6}$$

From Eq. 2.6, we can obtain the expression for DC device lifetime τ from the fact that $\Delta D_f = A\tau^n$ (ΔD_f is the amount of degradation at which device lifetime is defined):

$$\tau = BWI_{sub}^{-m} I_{ds}^{m-1} \tag{2.7}$$
$$B = H\Delta D_f^{1/n}$$

In CAS, to calculate the device degradation quasi-statically for a device undergoing dynamic operation, we do the following. To calculate the total ΔD that occurs in the SPICE analysis, we need to calculate ΔD during each timestep Δt of the analysis. We assume that all parameters and currents are constant during this timestep and are equal to their values at the beginning of the timestep. Let us number each time period 1,2,... with differing A coefficients A_1, A_2,... and n values n_1, n_2,... because of the variations of the degradation parameters and currents that occur for different times.

Starting from the beginning of the analysis, ΔD occurring in the first timestep is merely $\Delta D(t_1) = A_1\Delta t^{n_1}$, since no degradation has occurred before this timestep. To calculate ΔD of the next timestep,

however, we need to consider the amount of degradation that occurred before it, in this case equal to $A_1\Delta t^{n_1}$. Since ΔD of the present timestep depends only on the magnitude of the previous current degradation and not on the stressing history, we can introduce another variable, t', that represents the time it would take the device to experience this previous degradation but at the present current level and parameter value. In other words, in the present example, we can introduce a t' such that

$$\Delta D(t_1) = A_1\Delta t^{n_1} = A_2 t'^{n_2} \tag{2.8}$$

We can then directly add the times to get the total degradation up to the present timestep:

$$\Delta D(t_2) = A_2(t'+\Delta t)^{n_2} \tag{2.9}$$

Note that we cannot directly add the degradation of the two time periods together ($A_1\Delta t^{n_1} + A_2\Delta t^{n_2}$) because Eq. 2.1 only applies to DC stressing where A and n are constant, and a perfectly fresh device is assumed at $t = 0$.

Using (2.8) to solve for t', we can substitute t' in (2.9) and obtain

$$\Delta D(t_2) = A_2\left[\left[\frac{\Delta D(t_1)}{A_2}\right]^{\frac{1}{n_2}} + \Delta t\right]^{n_2} = \left[\Delta D(t_1)^{\frac{1}{n_2}} + A_2^{\frac{1}{n_2}}\,\Delta t\right]^{n_2} \tag{2.10}$$

Eq. 2.10 thus states that the total degradation up to the present timestep can be found from that of the previous timestep and from the present currents and degradation parameters. In CAS, Eq. 2.10 is applied successively to each timestep to find the total device degradation of the SPICE analysis.

To calculate the device lifetime τ, we now need to calculate the length of time needed for the degradation to equal a user-defined value, ΔD_f. As mentioned in the introduction, it is assumed that all waveforms simulated in the SPICE analysis is repeated until ΔD_f is reached. If the length of the SPICE analysis is denoted by T, then we need to find N such that

$$N = \frac{\tau}{T} \tag{2.11}$$

where N equals the number of repetitions of the SPICE analysis necessary for $\Delta D = \Delta D_f$. For each SPICE analysis, we can assign an effective $A = A_{eff}$ and an effective $n = n_{eff}$ such that

$$\Delta D(T) = A_{eff} T^{n_{eff}} \tag{2.12}$$

Since each time period of length T is now identical (since the waveforms are now periodic), finding ΔD reverts back to the DC stressing case where A_{eff} and n_{eff} are constant. Thus, by substituting τ and T in Eq. 2.11 with Eq. 2.12, we get

$$N = \frac{\frac{\Delta D_f^{\frac{1}{n_{eff}}}}{A_{eff}}}{\frac{\Delta D(T)^{\frac{1}{n_{eff}}}}{A_{eff}}} = \left[\frac{\Delta D_f}{\Delta D(T)}\right]^{\frac{1}{n_{eff}}}$$

or

$$\tau = TN = T\left[\frac{\Delta D_f}{\Delta D(T)}\right]^{\frac{1}{n_{eff}}} \tag{2.13}$$

Conversely, to find the amount of degradation at a certain specified future time t_{sp}, Eq. 2.13 can be re-arranged so that

$$\Delta D(t_{sp}) = \Delta D(T)\left[\frac{t_{sp}}{T}\right]^{n_{eff}} \tag{2.14}$$

To find A_{eff} and n_{eff}, we need to use Eq. 2.12 with two different ΔD values because of the two unknown parameters involved. The ΔD calculation of Eq. 2.10 is thus extended to double the length of the original SPICE analysis so that $\Delta D(T)$ and $\Delta D(2T)$ can then be used to find A_{eff} and n_{eff}.

2.3 Substrate Current Model

The substrate current model is an empirical model developed for the BSIM1 parameter extraction program. This model is used for all SPICE models. The following is a summary of the equations and parameters used. Refer to [Jen87] and [Lee86] for more detailed information,

$$I_{sub} = \frac{A_i}{B_i} I_{ds}(V_{ds} - V_{dsat}) \exp\left[-\frac{B_i l_c}{V_{ds} - V_{dsat}}\right]$$

where

$$V_{dsat} = \frac{E_{crit}L(V_{gs} - V_{th})}{E_{crit}L + V_{gs} - V_{th}}$$

$$E_{crit} = E_{crit0} + E_{critg}V_{gs} + E_{critb}V_{bs}$$

and

$$l_c = \sqrt{t_{ox}}\left[l_1 + l_2\left[\frac{1}{V_{gs} + 2}\right]\right]$$

$$l_1 = l_{c0} + l_{c1}\left[\frac{1}{V_{bs} - 4}\right] + \left[l_{c2} + l_{c3}\left[\frac{1}{V_{bs} - 4}\right]\right]V_{ds}$$

$$l_2 = l_{c4} + l_{c5}\left[\frac{1}{V_{bs} - 4}\right] + \left[l_{c6} + l_{c7}\left[\frac{1}{V_{bs} - 4}\right]\right]V_{ds}$$

$$A_i = \begin{cases} 2\times10^2 \text{ 1/µm for NMOS devices} \\ 1\times10^3 \text{ 1/µm for PMOS devices} \end{cases}$$

$$B_i = \begin{cases} 1.7\times10^2 \text{ V/µm for NMOS devices} \\ 3.7\times10^2 \text{ V/µm for PMOS devices} \end{cases}$$

Thus, there are 11 additional parameters (E_{crit0}, E_{critg}, E_{critb}, l_{c0}, l_{c1}, l_{c2}, l_{c3}, l_{c4}, l_{c5}, l_{c6}, l_{c7}). In its simplest form, however, only l_{c0} and E_{crit0} need to be specified, in which case the model simplifies to the physical I_{sub} model [Cha84,Sod84]:

$$I_{sub} = \frac{A_i}{B_i}I_{ds}(V_{ds} - V_{dsat})\exp\left[-\frac{B_i l_{c0}\sqrt{t_{ox}}}{V_{ds} - V_{dsat}}\right]$$

where

$$V_{dsat} = \frac{E_{crit0}L(V_{gs} - V_{th})}{E_{crit0}L + V_{gs} - V_{th}}$$

See section 5.4 for default values.

2.4 New Degradation Parameters

The implementation of the degradation and device lifetime model introduces three parameters, each with two coefficients to model their behavior with respect to V_{gd}. The three parameters, H, m, and n, are implemented as follows:

$$H = H_0 + H_{gd}V_{gd}$$
$$m = m_0 + m_{gd}V_{gd}$$
$$n = n_0 + n_{gd}V_{gd}$$

The bias dependence of H and m on V_{gd} implemented here correlate with the results found in [Kuo87b,Kuo88], while the bias dependence of n on V_{gd} has not been experimentally verified and is thus implemented only as an approximation to what it may be in reality.

It should be mentioned that the these degradation parameters must be extracted by separate device stressing measurements and added manually to the model parameter file when using any SPICE model (BSIM included). More information on creating the modified model parameter file is given in Section V.

2.5 Device Stressing Methodology

There are several possible techniques in doing device stressing to extract the device degradation parameters listed in the previous section. The variations concern both the quantity that is monitored, such as I_{sub} or I_{sub}/I_{ds}, and what type of device stressing is used, such as constant voltage, constant I_{sub}, or constant field (constant I_{sub}/I_{ds}).

Eq. 2.6 of Section 2.2 suggests that constant field stressing (constant I_{sub}/I_{ds}) should be used to extract consistent degradation parameters. This condition implies that the rate of degradation is minimally affected by the degradation of device behavior as stressing proceeds. For instance, for the constant voltage case, as the device degrades, the current levels flowing in the device will change with time. Thus, the actual stressing condition, which depends on the electric field in the device, will also change with time.

Eq. 2.7 implies that to extract m and H, device lifetime should be plotted with the current ratio I_{sub}/I_{ds} rather than I_{sub} alone. By re-arranging Eq. 2.7, we can obtain the following expression:

$$\tau\left[\frac{I_{ds}}{W}\right] = H\Delta D_f^{1/n}\left[\frac{I_{sub}}{I_{ds}}\right]^{-m} \tag{2.15}$$

If we plot Eq. 2.15 in log-log format, we can find m from the slope and H from the intercept. This method is preferred since it corresponds directly with theory [Hu85,Wer86], correlates well with device degradation for a wider range of device sizes and stressing biases [Cha88], and relates directly to the amount of interface traps formed [Bel89].

Because the parameters m and H are V_{gd} dependent as mentioned in the previous section, devices used to extract one m and H pair should be stressed at the same V_{gd} value. Separate sets of devices should then be stressed at different V_{gd} biases if the V_{gd}-sensitivity terms are desired.

One difficulty with doing device stressing based on I_{sub}/I_{ds} is obtaining a wide range of I_{sub}/I_{ds} values to create the plot described by Eq. 2.15, since I_{sub} and I_{ds} tend to track one another as the stressing condition is changed. A wider spread of data points can be obtained by using the more traditional τ versus I_{sub} plot. This method is equivalent to the method based on I_{sub}/I_{ds} if the stressing gate voltage is unchanged for the stressed set of devices (constant V_{gs} implies constant I_{ds} since the device is biased in the saturation region, hence constant I_{sub} implies constant I_{sub}/I_{ds}). However, the V_{gd} sensitivity terms cannot be extracted since in keeping the stressing gate voltage constant, V_{gd} must be necessarily varied to change the stressing conditions. Thus, this method can be somewhat easier to implement at a slight cost in extraction accuracy of the degradation parameters.

2.6 Enhanced AC Degradation

There have been several publications describing enhanced hot-carrier degradation from AC effects [Wer86,Cho87,Aur89]. These effects can be catagorized into two waveform cases, the "good" and the "bad" case. The "bad" waveform, known to cause the greatest enhanced AC degradation, corresponds to the case where there is a sudden and deep fall of V_{gs} in the presence of high V_{ds} [Cho87], while the "good" waveform corresponds to all other cases. Although certain published reports show enhanced degradation at the device level even in the "good" case [Wer86,Aur89], we have not seen this enhancement on the circuit level for inverter-class waveforms which are classified as the "good" case [Lee90]. We thus believe the quasi-static model presented here is valid for this "good" class of waveforms (which represent the majority of waveforms encountered in circuits).

In case the "bad" waveform is encountered, a warning is issued when the following criteria are met:

1) V_{gs} fall > 3 V

2) $V_{ds} - V_{dsat} > 4$ V during the V_{gs} fall.

3) $\frac{\partial V_{gs}}{\partial t} > 10$ V/μs

If this situation is encountered, the models presented in this section (as well as the aging model described in the next section) are susceptible to underestimation of the hot-carrier degradation. A model to take this enhanced degradation into account will be implemented in a future version of CAS.

2.7 Summary

A physically-based device degradation model has been presented in this section. Because of their same power law behavior with respect to time, current degradation, transconductance degradation, and threshold voltage shift can be calculated by directly replacing the ΔD term in the preceding equations with the appropriate parameter. Eq. 2.10, 2.13 and 2.14 are the actual equations implemented in CAS.

The next section will describe the implementation of circuit aging for NMOSFETS in CAS.

III. NMOSFET CIRCUIT AGING MODEL

3.1 Introduction

This section describes the models and formulations used to generate the aged model parameters at the user-specified future time point. A new parameter, Age, is introduced to quantify the amount of degradation each device experiences in a circuit environment. This Age parameter is then used as the basis in finding the aged model parameters.

3.2 Model Formulation

To determine the amount of degradation that occurs in a device, we must look back at the degradation equations of Section II. Since the amount of degradation depends on the stressing condition as well as on time, an Age parameter solely based on time cannot be used. From Eq. 2.6 of Section II, we can describe this degradation by the form

$$\Delta D = f(At^n) = f\left[\left[\frac{I_{ds}}{WH}\left[\frac{I_{bs}}{I_{ds}}\right]^m\right]^n t^n\right] \tag{3.1}$$

where in this case we have generalized the relationship of the degradation to At^n by some monotonic function f (the aging concept does not require an explicit form for f, as we shall see shortly). We can thus introduce an Age variable that is related to this degradation as well as being linearly dependent on time:

$$\Delta D = f(Age^n)$$
$$Age = \frac{I_{ds}}{WH}\left[\frac{I_{bs}}{I_{ds}}\right]^m t \tag{3.2}$$

Eq. 3.2 has all the information necessary - degradation parameters, currents, and time - and is geometry-independent. During circuit simulation, the Age is calculated for each device at each timestep, then integrated to obtain the total Age of the SPICE analysis,

$$Age(T) = \int_{t=0}^{T} \frac{I_{ds}}{WH}\left[\frac{I_{bs}}{I_{ds}}\right]^m dt \tag{3.3}$$

where T is, as before, the length of the SPICE analysis. The age that each device would have at the user-specified time T_{age} is just

$$Age(T_{age}) = Age(T)\left[\frac{T_{age}}{T}\right]$$

The list of ages for every device in the circuit is stored in an external file called "agetable" to be used for the creation of aged model parameters.

To create these aged model parameters, CAS needs a set of model parameter files extracted from the same device but at different levels of degradation. The principle behind the system is as follows:

(1) The user extracts model parameters from a fresh device, followed by extractions of the same device after it has been DC-stressed for different lengths of time.

(2) The user calculates the Age of each of the extracted set of model parameters by using Eq. 3.2. This is relatively straightforward since the stressing conditions are known.

(3) CAS simulates the desired circuit and calculates the Age that each device in the circuit would have if the SPICE analysis is repeated up to the user-specified future time point.

(4) CAS compares the Age of each device in the circuit with that of the stressed model parameter files of step (1), and calculates the new aged model parameters of the devices in the circuit by interpolation or regression.

The concept of calculating the aged parameter set is graphically given in Fig. 3.1. The barrels represent the fresh and pre-stressed model parameter files with ages Age_1, Age_2, etc., with the age of the circuit device (calculated by CAS) denoted by Age. Typically the age of the circuit device will lie between two of the pre-stressed model parameter sets. The user has the choice to specify whether interpolation is used (as shown in Fig. 3.1), or whether regression is desired. In both cases, the user also has a choice of whether to perform the analysis in the linear-linear, linear-log, or log-log domain. Generally, log-log is not recommended if the devices in the circuit have very small ages.

The reason we have generalized the relationship between degradation and At^n in Eq. 3.1 by a monotonic function f now becomes clear. Because we did not assume any functional form for the

model parameters versus Age, no explicit function f for Eq. 3.1 is assumed.

For BSIM parameter extractions, if multiple device extractions are done to create size-independent process files, every device that will be used to construct one size-independent process file must be subject to the same amount of degradation when doing stressing so that the extraction of each set of stressed process files will consist of devices with the same Age. Since it is difficult to set the same current level for each device, it is recommended that once a current level is set, the stressing time for each device should be varied to obtain the same Age. In general, this method is not recommended because of its complexity.

As a final note, some precautions. First, in its present implementation, the Age expression does not support a bias-dependent n parameter. A constant n ($n_{gd} = 0$) value should be used for all aging simulations. Secondly, the susceptibility to enhanced AC degradation discussed in Section 2.6 equally applies to circuit aging.

3.3 Summary

In this section we have introduced the concept of Age to generate degraded model parameters for NMOSFETs for the simulation of circuits at a user-specified future time point. The next section will describe a preliminary degradation and aging model for PMOS devices.

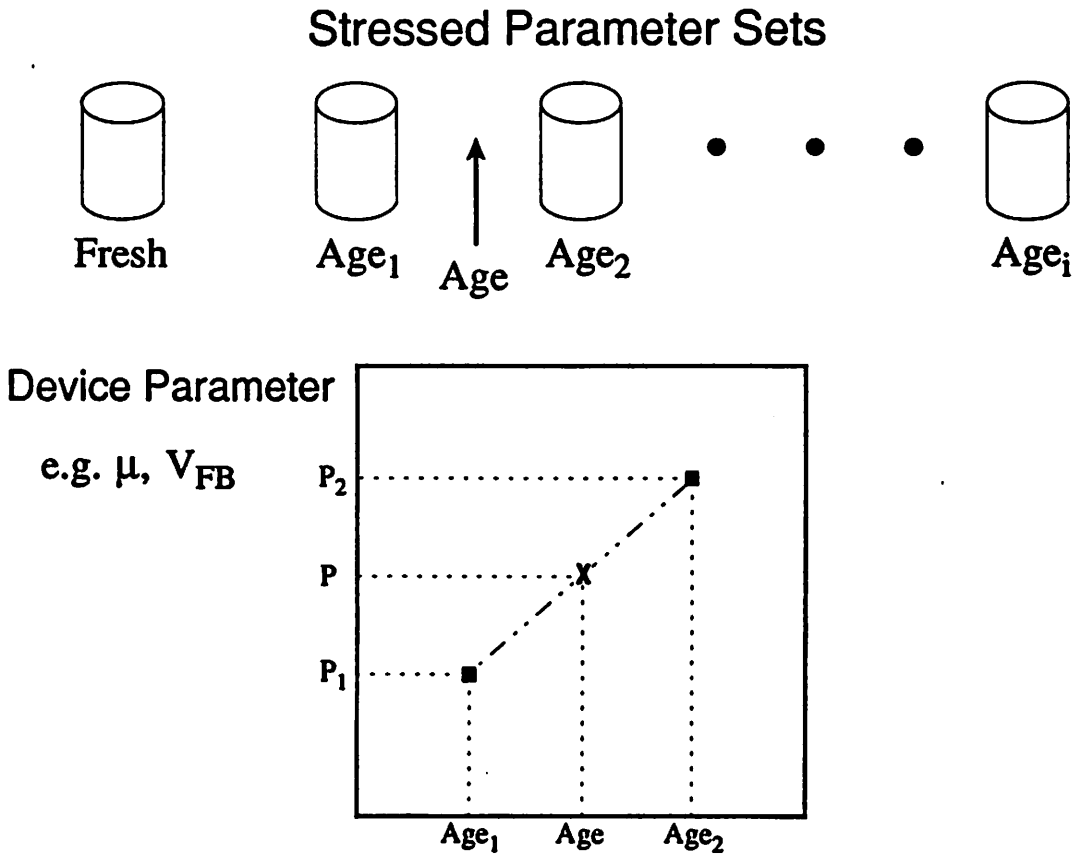

Fig. 3.1 Calculation of the aged parameter from pre-stressed model parameter sets. The barrels represent the various model parameter sets with different ages Age_1, Age_2, etc., while the in-circuit device suffers degradation represented by Age.

IV. PMOSFET DEGRADATION AND AGING MODELS

4.1 Introduction

This section presents a preliminary PMOS degradation and aging model that is implemented in similar fashion to that of the NMOS device. The difference occurs in that now the gate current I_{gate} enters the picture as I_{sub} has for the NMOSFET. Because of the disagreement in the literature on whether I_{gate} or I_{sub} correlates better with degradation, we have incorporated both currents through a weighting coefficient that can be specified by the user.

4.2 Gate Current Model

The following PMOS gate current model developed by [Tam84] and [Ong89] is used in CAS:

$$I_{gate} = G_1 \frac{I_{sub} t_{ox}}{\lambda_r} \left[\frac{\lambda E_m}{\phi_b} \right]^2 P(E_{ox}) \exp\left[-\frac{\phi_b}{E_m \lambda} \right] \tag{4.1}$$

where

$$P(E_{ox}) = \left[\frac{5.66\times10^{-6} E_{ox}}{\left[1+\frac{E_{ox}}{1.45\times10^{5}}\right]} \times \frac{1}{\left[1+\frac{2\times10^{-3}}{L_{eff}} \exp(-E_{ox} t_{ox}/1.5)\right]} + 2.5\times10^{-2} \right] \exp(-300/\sqrt{E_{ox}}) \tag{4.2}$$

for $E_{ox} \geq 0$, and

$$P(E_{ox}) = 2.5\times10^{-2} \exp(-X_{ox}/\lambda_{ox}) \tag{4.3}$$

for $E_{ox} \leq 0$. $P(E_{ox})$ is essentially the probability that a scattered electron will surmount the oxide barrier and flow to the gate. $G_1 = 0.5$, $\lambda_{ox} = 320$ Å, $\lambda_r = 616$ Å is the re-direction scattering mean free path, and $\lambda = 105$ Å is the scattering mean free path of the electron [Ong89]. The oxide barrier height ϕ_b can be expressed by

$$\phi_b = 3.2 - 2.6\times10^{-4}\sqrt{E_{ox}} - \upsilon E_{ox}^{2/3} \tag{4.4}$$

where $\upsilon = 4\times10^{-5}$ in [Tam84].

At present, we have only made G_1 and υ parameters that can be specified by the user. Once developed, a more accurate model will be included in a future version of CAS.

4.3 Degradation and Aging Model Based on I_{gate}

The PMOS degradation and aging models closely parallel that of the NMOS case. By slightly modifying the expression from [Ong89] and paralleling Eq. 2.7,

$$\tau = B\left[\frac{I_{gate}}{W}\right]^{-m} \tag{4.5}$$

where the gate current is normalized by the device width W. Let us assume that the PMOS degradation follows the same power-law behavior as for the NMOS device. Then,

$$\Delta D = At^n \tag{4.6}$$

Denoting ΔD_f as the degradation level defining the device lifetime τ as before, we obtain

$$\Delta D_f = A\tau^n \tag{4.7}$$

Solving for the coefficient A using Eqs. 4.5 and 4.7, we get

$$A = \frac{\Delta D_f}{B^n}\left[\frac{I_{gate}}{W}\right]^{mn} t^n \tag{4.8}$$

Substituting

$$H = \frac{B}{\Delta D_f^{1/n}} \tag{4.9}$$

we finally get

$$\Delta D = \left[\frac{1}{H}\left[\frac{I_{gate}}{W}\right]^m\right]^n t^n \tag{4.10}$$

The conversion of the parameter B to H in Eq. 4.9 is necessary to remove the dependency of the parameter set to the level of degradation that the lifetime is defined at, which, in this case, is ΔD_f.

As for the Age expression, by looking at Eq. 4.10, we can parallel the NMOS analysis and propose the following expression for Age,

$$\Delta D = (\text{Age})^n$$

Then,

$$\text{Age} = \frac{1}{H}\left[\frac{I_{gate}}{W}\right]^m t \tag{4.11}$$

4.4 Incorporation of I_{sub} and I_{gate} in Predicting Degradation

To conglomerate the substrate current and gate current degradation models, we can sum the contributions from each component linearly through weighting coefficients (W_g,W_b = 1 - W_g) that can be specified by the user,

$$\text{Age} = W_b \times \left[\text{Age from } I_{sub}\right] + W_g \times \left[\text{Age from } I_{gate}\right] \tag{4.12}$$

For the following, H_b and m_b denote the H and m parameter for I_{sub}, while H_g and m_g denote the H and m parameter associated with I_{gate}. Note that the n parameter is the same for both cases (since n depends only on the degradation behavior with time and not on what currents are used as a basis for degradation). Then, the following equation can be derived for the age:

$$\text{Age} = \left\{W_b\left[\frac{I_{ds}}{WH_b}\left[\frac{I_{sub}}{I_{ds}}\right]^{m_b}\right] + W_g\left[\frac{1}{H_g}\left[\frac{I_{gate}}{W}\right]^{m_g}\right]\right\} t \tag{4.13}$$

To calculate the degradation expression, to conform with

$$\Delta D = (\text{Age})^n$$

as it is for the NMOS case, the following expression results:

$$\Delta D = \left\{W_b\left[\frac{I_{ds}}{WH_b}\left[\frac{I_{sub}}{I_{ds}}\right]^{m_b}\right] + W_g\left[\frac{1}{H_g}\left[\frac{I_{gate}}{W}\right]^{m_g}\right]\right\}^n t^n \tag{4.14}$$

Eq. 4.13 and 4.14 are then used in the degradation and aging calculations discussed in Sections II and III.

4.5 Parameters Necessary for Simulation

Because the I_{gate} model involves I_{sub} and E_m, all substrate current parameters must be extracted. In addition, the parameters G_1 from Eq. 4.1 and υ from Eq. 4.4 need to be extracted. Future additional research and model development should indicate other parameters that need to be extracted to model I_{gate}.

As for the degradation parameters, H_g, m_g, and n should be extracted in similar manner as in the NMOS case. n is the slope when device degradation is plotted against time in log-log format (Eq. 4.6). $-m_g$ and B_g are the slope and intercept respectively when device lifetime τ is plotted against I_{gate} in log-log format (Eq. 4.5). B_g must then be converted to H_g to remove the dependency of the parameter set to the level of degradation defined at the device lifetime (Eq. 4.9). H_g and m_g are further divided into a constant and V_{gd}-sensitivity term as in the I_{sub} case:

$$H_g = H_{g0} + H_{ggd}V_{gd}$$
$$m_g = m_{g0} + m_{ggd}V_{gd}$$

Thus, to summarize, the following parameters must be added to the model parameter set to simulate PMOS degradation:

1) G_1: G1: constant coefficient for I_{gate} (default = 0.5)[Tam84].

2) υ: UPS: sensitivity of ϕ_b to the $E_{ox}^{2/3}$ term (default = 4×10^{-5} $V^{1/3}cm^{2/3}$)[Tam85].

3) H_{g0}: HG0: intercept parameter of the lifetime versus I_{gate} plot (default = 10^4).

4) H_{ggd}: HGGD: V_{gd}-sensitivity term for H_g (default = 0).

5) m_{g0}: MG0: slope parameter of the lifetime versus I_{gate} plot (default = 1.5)[Ong89].

6) m_{ggd}: MGGD: V_{gd}-sensitivity term for m_g (default = 0).

7) W_g: WG: weighting coefficient for I_{gate}-based degradation (default = 0 or 1).

Unlike the other model and degradation parameters, the PMOS I_{gate} and degradation parameters are declared in the input deck using the '.pmosdeg' command (See Section 5.4). The default value of W_g is 1 if the '.pmosdeg' command is specified; otherwise W_g defaults to 0.

4.6 Summary

This section has introduced a preliminary PMOS degradation and aging model that parallels that of the NMOSFET case. Further development and refinement of the models will be incorporated into future versions of CAS. The next section will describe the system structure, the installation procedure and usage of BERT-CAS, and the special CAS commands that can be used in the SPICE input deck.

V. BERT CONFIGURATION AND OPERATION

5.1 Introduction

This section describes the organization and operating procedure of BERT. A description of the system, the steps needed to install and run the program, a summary of the special CAS commands, and a CAS simulation example are included. The special commands for the oxide and electromigration modules are listed in companion BERT manuals [Ros90] and [Lie90a].

5.2 System Configuration

Figs. 5.1 and 5.2 show the system structure of BERT. As in SCALE, BERT consists of a pre- and post-processor linked by SPICE, with several intermediate files for communication between the pre- and post-processor (Fig. 5.1). The pre-processor interprets the special BERT commands, prepares the input deck so that it is SPICE-compatible, and writes information to an intermediate file for communication with the post-processor. In addition, the pre-processor requests SPICE to print out all voltage nodes necessary for the calculation of substrate current. After SPICE calculations are done, the post-processor uses the voltage node printout to calculate the transient substrate current waveform and individual device degradation. If aging is requested, the post-processor creates the file "agetable" listing the ages of all the devices in the circuit (Fig. 5.1). To create the aged model parameters, the pre-processor is run once again with the original input file as its argument (Fig. 5.2). Once the pre-processor detects that an agetable is present, it will create all the aged model parameter files using the pre-stressed model parameter sets (denoted by the barrels in Fig. 5.2). The pre-processor also creates a new input deck with the necessary modifications to run it with the new aged model parameter files. The pre- and post-processor combination is run again to obtain the aged behavior of the circuit.

5.3 Installing and Running BERT

A standard makefile exists for the compilation of both the pre- and post-processors in the bert/ directory. Simply typing 'make' on UNIX systems in bert/ will compile all modules and place all executable codes in the bert/exe/ directory. Under bert/, all CAS-related files (source code, sample input decks and sample BSIM1 model parameter files) are located in CAS/, all oxide reliability files are located in CORS/, and all electromigration simulation files are located in EM/. To remove all object files from these directories, type 'make clean'.

To execute the programs, type

prebert *-x deck* **|** *spice* **| postbert >** *outfile*

where *x* is "2" for SPICE2G6, or "3" for SPICE3B1. The default (if no option is specified) is SPICE3C1.

To use CAS to find the substrate current and device lifetime only, the above execution is the only step required. To simulate circuit aging, the following three lines must be executed in the order shown:

prebert *-x deck* **|** *spice* **| postbert >** *outfile* (to generate the agetable),

prebert *-x deck* (to generate the aged process files),

prebert *-x* **inpdeck |** *spice* **| postbert >** *outfile* (to simulate the aged circuit).

The second step generates an input deck called **inpdeck** containing all the necessary modifications to use the newly created aged model parameter files. **Inpdeck** is otherwise identical to the original input deck except that the ".age" and ".ageproc" aging commands are omitted. Thus, in step 3, **inpdeck**, rather than the original input deck, is used.

5.4 CAS Command Summary

The following new commands are for use specifically with CAS for substrate current, device degradation analysis, and circuit aging. CAS includes a revised SCALE command set that eliminates

some of the redundancy and adds more flexibility to the ones listed in [Jen87]. Note that many commands are similar to SPICE commands.

(1) **.AGE** *time*

Examples:

.AGE 10years

.AGE 5minutes

This command specifies the future time at which to calculate the aged model parameter files for circuit simulation. The units for time can be in "y", "h", "m", or "s", corresponding to years, hours, minutes, and seconds, with no space between the number and the unit. Letters following the above four units of time are ignored. Thus **10years** and **10y** are interpreted identically.

(2) **.AGEDID** *time*

.AGEDGM *time*

.AGEDVT *time*

Examples:

.AGEDID 10years

.AGEDVT 1year

These commands specify the future time at which drain current degradation $\Delta I_{ds}/I_{ds0}$ (**AGEDID**), transconductance degradation $\Delta g_m/g_{m0}$ (**AGEGM**), or threshold voltage shift ΔV_{th} (**AGEDVT**) is desired. This is the converse of the lifetime commands **DELTAID, DELTAGM, DELTAVT**. The format for *time* is identical to that of the AGE command. Note that appropriate H, m, and n values must be given, since parameter values will differ depending upon the actual degradation specified ($\Delta I_{ds}/I_{ds0}$, $\Delta g_m/g_{m0}$, or ΔV_{th}). Setting H_0 and H_{gd} to 0 will disable the calculation for that particular model.

(3) **.AGEMETHOD** *method <domain>*

Examples:

.AGEMETHOD INTERP LINLOG

.AGEMETHOD LINLIN

This command specifies the method of numerical analysis used to calculate the aged parameter set from the pre-stressed model parameters. The first argument specifies the method of the regression analysis (LINLIN, LINLOG, or LOGLOG). The keyword INTERP should be placed in this position if interpolation rather than regression is desired. The keyword INTERP can be followed by the method in which the interpolation will be performed (LINLIN, LINLOG, or LOGLOG). The default is linear-log interpolation if no **AGEMETHOD** command is present.

(4) **.AGEPROC** *mname* **FILENAMES=***fname1, fname2, fname3 <fname4,...>*

Example:

.AGEPROC PC1 FILENAMES=DE0, DE1, DE2, DE3

This command specifies the names of the pre-stressed model parameter files *fname* associated with the model *mname*. The filenames should be ordered by increasing ages, with the fresh file first. At least one fresh and one aged model parameter file must be present for linear-linear analysis, while two aged model parameter files must be present for linear-log or log-log analysis. Note that unlike the **.PROCESS** statement, "**FILENAMES**" appears in plural form. The **.PROCESS** command is still needed. The format of the aged model parameter files is identical to the fresh model parameter files used in the **.PROCESS** command.

(5) **.DEGPRINT** *trnname1 <trnname2 ...>*

Example:

.DEGPRINT M1 M4 M6

This command restricts degradation information printout (such as that shown in Fig. 5.5) to occur only for the specified transistors. Without this command, degradation information for all the transistors in the circuit will be printed out.

(6) **.DEGSORT**

Example:

.DEGSORT

This command requests a printout in tabular form all the transistors in the circuit listed from the most degraded to least degraded. The corresponding device lifetime is given if one of the **.DELTA** commands (e.g. **.DELTAID**) is present, the amount of device degradation is given if one of the **.AGE** commands (e.g. **.AGEDID**) is present, and the age of each transistor is given if the **.AGE** command is present.

(7) **.DELTAID** *value*

.DELTAGM *value*

.DELTAVT *value*

Examples:

.DELTAID 0.05

.DELTAGM 0.1

.DELTAVT 10mV

These commands specify either drain current degradation $\Delta I_{ds}/I_{ds0}$, transconductance degradation $\Delta g_m/g_{m0}$, or the threshold voltage shift ΔV_{th}, at which the device lifetime is defined. Like the **AGEDID, AGEDGM,** and **AGEDVT** commands, appropriate values of H, m, and n must be specified

depending on which of the three criteria is used to determine device lifetime. Again, setting $H_0 = 0$ and $H_{gd} = 0$ will disable the calculation for that particular model.

(8) **.ISUBWIDTH** = *colwidth*

Example:

.ISUBWIDTH = 90

This command controls the width of the substrate current output printout in SPICE2. This is independent of the usual **.WIDTH** command. Permissible values for *colwidth* range from 80 to 200. The default value is 80.

(9) **.PMOSDEG** *mname <keyword1=value> <keyword2=value> ...*

Example:

.PMOSDEG PMOSMODEL G1=0.6 UPS=1E-5 HG0=2E3 MG0=1.6 WG=0.9

This command specifies the gate current degradation parameters for the PMOS devices. *mname* is the model name that this parameter set is associated with. The following parameter keywords are recognized:

1) G1: constant coefficient for I_{gate} (default = 0.5).

2) UPS: sensitivity of ϕ_b to the $E_{ox}^{2/3}$ term (default = 4×10^{-5} $V^{1/3}cm^{2/3}$).

3) HG0: intercept parameter of the lifetime versus I_{gate} plot (default = 10^4).

4) HGGD: V_{gd}-sensitivity term for H_g (default = 0).

5) MG0: slope parameter of the lifetime versus I_{gate} plot (default = 1.5).

6) MGGD: V_{gd}-sensitivity term for m_g (default = 0).

7) WG: weighting coefficient for I_{gate}-based degradation (default = 0 or 1).

The default value for WG is 1 if the PMOSDEG command is present, 0 if not. See Chapter 4 for the model description.

(10) .PRINTIGATE or .PLOTIGATE

.PRINTIGATE MXXXX <MYYYY ... MZZZZ> <ALL>

.PRINTIGATE SXXXX <SYYYY ... SZZZZ> <ALL>

.PLOTIGATE MXXXX <MYYYY ... MZZZZ> <ALL> <(MIN,MAX)>

.PLOTIGATE SXXXX <SYYYY ... SZZZZ> <ALL> <(MIN,MAX)>

Examples:

.PLOTISUB S1 S4 (0,7E-6)

.PRINTISUB M1 M4 ALL

These commands are used to either print or plot out the gate current of the specified PMOS transistors. SYYY is the transistor denotation for the BSIM1 model in SPICE2, while MYYY is that for non-BSIM1 models in SPICE2 and all models in SPICE3. Note that the format is similar to the normal .PRINT and .PLOT commands in SPICE, except that the TRAN keyword is unnecessary. MIN and MAX specify the minimum and maximum values for the plot. The keyword ALL is used if a printout or plotout of the total gate current of all the PMOS transistors in the circuit is desired.

(11) .PRINTISUB or .PLOTISUB

.PRINTISUB MXXXX <MYYYY ... MZZZZ> <ALL>

.PRINTISUB SXXXX <SYYYY ... SZZZZ> <ALL>

.PLOTISUB MXXXX <MYYYY ... MZZZZ> <ALL> <(MIN,MAX)>

.PLOTISUB SXXXX <SYYYY ... SZZZZ> <ALL> <(MIN,MAX)>

Examples:

.PLOTISUB S1 S4 (0,7E-6)

.PRINTISUB M1 M4 ALL

These commands are used to either print or plot out the substrate current of the specified transistors. SYYY is the transistor denotation for the BSIM1 model in SPICE2, while MYYY is that for non-BSIM1 models in SPICE2 and all models in SPICE3. Note that the format is similar to the normal **.PRINT** and **.PLOT** commands in SPICE, except that the TRAN keyword is unnecessary. MIN and MAX specify the minimum and maximum values for the plot. The keyword is used if a printout or plotout of the total substrate current of all the NMOS and PMOS transistors in the circuit is desired. This is useful to determine whether, for instance, the substrate bias generator used is adequate for the circuit.

(12) **.PROCESS** *mname* **FILENAME=***fname*

Examples:

.PROCESS PC1 FILENAME=TRN

.PROCESS MK1 FILENAME=NMOS5

This command specifies the model name *mname* and the corresponding model parameter filename *fname* which contains all the device parameters. This configuration is identical to that already implemented for the BSIM1 model in SPICE2, but is new for the other models and SPICE3. It is important to realize that **.MODEL** commands are no longer necessary in the input deck, but that a **.PROCESS** command is now mandatory. All model parameter filenames should be in capital letters if SPICE2 used.

For SPICE Level 1, 2, or 3 models, the model parameter file format contains **.MODEL** commands with the model parameters in the usual SPICE **.MODEL** format. The only restrictions are that the I_{sub} and degradation parameters must be on separate lines from the drain current parameters, and only one model per file is allowed. For the SPICE Level 4 (BSIM1) model, the model parameter file is the file created by the BSIM1 extraction program (see [Jen87] and Fig. 5.3).

The following provides information concerning the format of these model parameter files.

Additional SPICE Level 1, 2, 3 Parameters:

The following shows the additional parameters and their keywords that can be added to the .MODEL parameter declarations.

	Name	parameter	units	default
43	ECRIT0	Constant term of E_{crit}	V/cm	1.0E4
44	ECRITG	V_{gs} dependence of E_{crit}	1/cm	0.0
45	ECRITB	V_{bs} dependence of E_{crit}	1/cm	0.0
46	LC0	Constant term of $l_c/\sqrt{t_{ox}}$	$\mu m^{1/2}$	1.0E-7
47	LC1	Bias-sensitivity term of $l_c/\sqrt{t_{ox}}$	$\mu m^{1/2}$–V	0.0
48	LC2	Bias-sensitivity term of $l_c/\sqrt{t_{ox}}$	$\mu m^{1/2}$–V^{-1}	0.0
49	LC3	Bias-sensitivity term of $l_c/\sqrt{t_{ox}}$	$\mu m^{1/2}$	0.0
50	LC4	Bias-sensitivity term of $l_c/\sqrt{t_{ox}}$	$\mu m^{1/2}$–V	0.0
51	LC5	Bias-sensitivity term of $l_c/\sqrt{t_{ox}}$	$\mu m^{1/2}$–V^2	0.0
52	LC6	Bias-sensitivity term of $l_c/\sqrt{t_{ox}}$	$\mu m^{1/2}$	0.0
53	LC7	Bias-sensitivity term of $l_c/\sqrt{t_{ox}}$	$\mu m^{1/2}$–V	0.0
54	H0	Degradation plot intercept (H_0)	A sec / (m $V^{\frac{1}{n}}$)	1.0E4
55	HGD	Degradation plot intercept (H_{gd})	A sec / (m $V^{\frac{n+1}{n}}$)	0.0
56	NN0	Slope of degradation parameter (n_0)		0.5
57	NNGD	Slope of degradation parameter (n_{gd})	V^{-1}	0.0
59	M0	Slope of degradation plot (m_0)		3.5
59	MGD	Slope of degradation plot (m_{gd})	V^{-1}	0.0
60	AGE	Device Age	A sec / m	0.0

BSIM1 Process File Modifications (SPICE3 Level 4):

Fig. 5.3 shows the modified format of the BSIM1 parameter process file. The format is identical to the previous format except five rows have been added below the substrate current parameters. Rows 35 through 37 contain the coefficients of the H, n, and m degradation parameters. The first column of Row 38 is the Age of the process file. This should be set to zero for a fresh process file. Columns two and three of Rows 38 and 39 are the minimum and maximum channel lengths and widths of the devices that were measured. For the single device case, set $L_{min} = L_{max}$ and $W_{min} = W_{max}$. All entries labeled "DUM" are dummy positions used as placeholders by the program.

The BSIM1 parameter extraction program includes a row of zeroes for Row 35, but no other rows are present. The user must add the extra rows manually and enter the appropriate values. As mentioned previously, the BSIM1 extraction program does not do DC stressing measurements; the degradation parameters must be obtained separately.

(13) **.TRAN** *tstep tstop* < *tstart* >

Examples:

.TRAN 1NS 100NS

.TRAN 5NS 1000NS 2NS

Since this simulator system is designed to calculate transient substrate currents, the SPICE .TRAN command should always be included whenever BERT is used. In order for the degradation calculations to be meaningful, the difference between *tstop* and *tstart* should be equal to a multiple of the period of the input signal.

(14) General form for MOSFETs :

SXXXX *nd ng ns nb mname* < **W=***value* > < **L=***value* > ...etc.

MXXXX *nd ng ns nb mname* < **W=***value* > < **L=***value* > ...etc.

Examples:

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

M1 1 2 3 4 PC1_NM1_DU1 W=20U L=1U

M2 1 2 3 4 MODP W=5U L=10U AD=100P AS=100P PD=40U PS=40U
```

To describe a MOSFET, the user should use SXXXX for the BSIM1 model in SPICE2, or MXXXX for all other models in SPICE2 and for all models in SPICE3. *mname* is the model name which should always be given. The format for the model name for the BSIM1 model is *pname_mt_dt*, where *pname* is the process name, *mt* is the MOSFET type, and *dt* is the source/drain junction type. The possible choices for *mt* are NM1 through NM5 for NMOSFETs, and PM1 through PM5 for PMOSFETs. DU1 to DU3 are the three available diffusion types. For users who are not familiar with SPICE commands, please consult the SPICE manual. For users who wish to learn more about the BSIM1 model implemented in SPICE or about the BSIM1 parameter extraction program, please refer to [Jen87].

One other note about transistor names. BERT-CAS treats transistors labeled as M1 and S1 as having identical names. Thus, use transistor names that differ from the second character onwards (e.g. M1 and S2).

5.5 Circuit Example: 21-Stage CMOS Inverter Chain

Fig. 5.4 shows a SPICE3 input deck for a 21-stage CMOS inverter chain circuit with a 100 MHz clocked input and 0.1pF capacitive loading at each inverter output (sample input file located in the bert/CAS/Sample/ directory). For this example circuit, the period of the input waveform is 10 ns. However, to accurately predict circuit degradation, we need to make the SPICE analysis long enough for the signal to propagate through the last stage of the inverter chain. Thus, the SPICE analysis is doubled to 20ns, with no additional signal being inputted during the extra time. This effectively means that, for this particular case, we also need to double the ages we specify for the '.age' and '.agedid'-type of commands if we want to simulate a periodic waveform of 10ns. Thus, in this case, although we want circuit degradation at 10 years in the future, we need to specify 20 years for the commands. Also, the device lifetime results calculated by the simulator will need to be halved to obtain the correct value.

Fig. 5.5 shows various degradation information for the NMOS (M202) and PMOS (M201) transistor of the 20th stage. As an example, to correctly interpret the results, the lifetime of M202 is 2.8 + 2 = 1.4 years, and $\Delta I_{ds}/I_{ds0}$ = 16.5% after 10 years of operation. Fig. 5.6 shows the generated agetable with the Age that all the transistors would have after 10 years of operation. Fig. 5.7 shows the output waveform of the 20th inverter stage comparing the propagation delay difference between the fresh and 10-year aged inverter chain. As expected, with device degradation, propagation delay is longer for the aged case.

5.6 Restrictions

1) BERT does not recognize subcircuits. All transistors must be explicitly declared.

2) Hot-carrier degradation of transistors in which the source and drain are switched regularly in circuit operation (such as transmission gates) cannot be simulated properly in this version.

5.7 Summary

We have described the installation and operating procedure of BERT-CAS in this section. Being able to separate the pre- and post-processing adds flexibility in use, but for convenience, a UNIX shell script program has been developed that automates the simulation process, as well as making iterative aging simulations possible. The shell script is the topic of the next section.

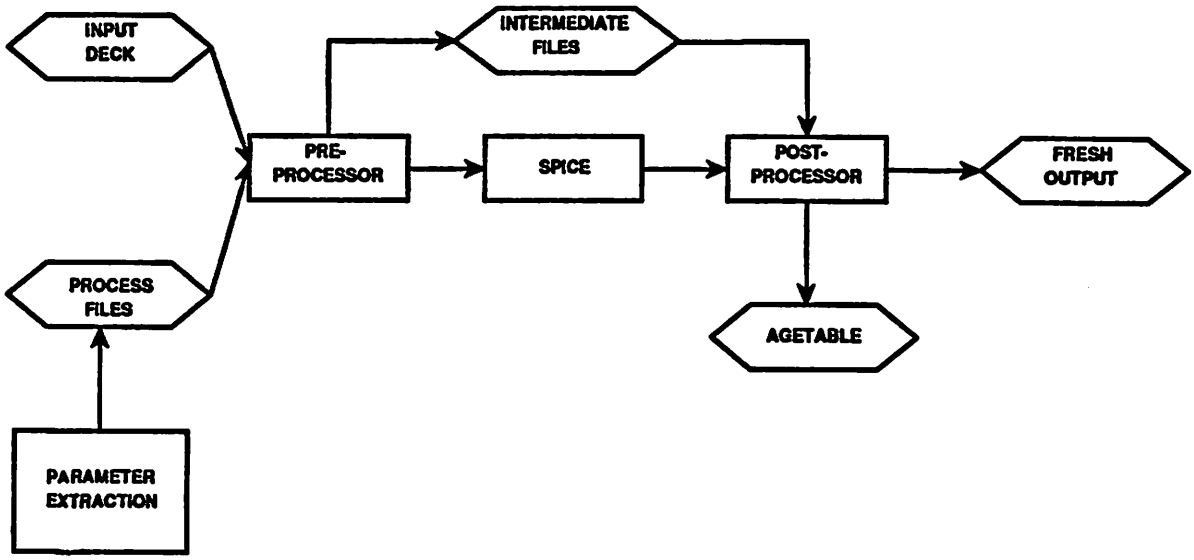

Fig. 5.1 BERT system configuration: First pass is to calculate degradation information (such as device lifetime) and the agetable.

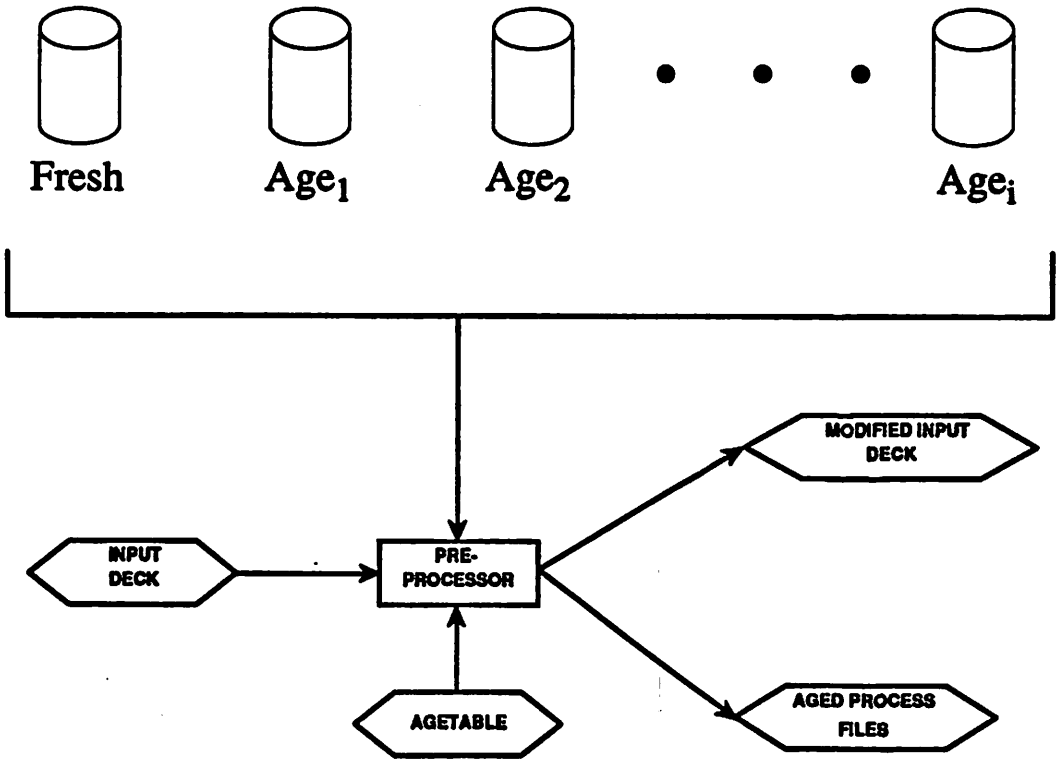

Fig. 5.2 BERT system configuration: Second pass is to generate the aged model parameters at the future time point specified by the '.age' command.

	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	η_0 (ETA)	η_{0l} (LETA)	η_{0w} (WETA)	-
6	μ_Z (MUZ)	δ_l (DL)	δ_w (DW)	$cm^2/V\text{-}s$, 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)	$cm^2/V^2\text{-}s$
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)	$cm^2/V^2\text{-}s$
15	μ_{SB} (X2MS)	μ_{SBl} (LX2MS)	μ_{SBw} (WX2MS)	$cm^2/V^2\text{-}s$
16	μ_{SD} (X3MS)	μ_{SDl} (LX3MS)	μ_{SDw} (WX3MS)	$cm^2/V^2\text{-}s$
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_0 (H0)	H_{gd} (HGD)	DUM3	*
36	n_0 (NN0)	n_{gd} (NNGD)	DUM4	-
37	m_0 (M0)	m_{gd} (MGD)	DUM5	-
38	AGE	L_{min} (LMIN)	W_{min} (WMIN)	**, µm, µm
39	DUM6	L_{max} (LMAX)	W_{max} (WMAX)	-, µm, µm

* : A sec / $(m\ V^{\frac{1}{n}})$
** : A sec / m

Fig. 5.3 The modified BSIM process file format to be used with CAS.

```
CMOS CLOCKED INVERTER CHAIN (21 STAGES)
*
* Power Supplies and Input Pulse.
*
vdd 40 0 dc 5.5
vin 1 0 pwl(0 0 0.02ns 5.5 5ns 5.5 5.2ns 0 )
vmeas 50 0 dc 0
*
* The Inverter Chain
*
m1 2 1 40 40 PC1_pm1_du2 w=60u L=1.4u
m2 2 1 0 0 PC2_nm1_du1 W=20u L=1.4u
m21 3 2 40 40 PC1_pm1_du2 w=60u L=1.4u
m22 3 2 0 0 PC2_nm1_du1 W=20u L=1.4u
m31 4 3 40 40 PC1_pm1_du2 w=60u L=1.4u
m32 4 3 0 0 PC2_nm1_du1 W=20u L=1.4u
m41 5 4 40 40 PC1_pm1_du2 w=60u L=1.4u
m42 5 4 0 0 PC2_nm1_du1 W=20u L=1.4u
m51 6 5 40 40 PC1_pm1_du2 w=60u L=1.4u
m52 6 5 0 0 PC2_nm1_du1 W=20u L=1.4u
m61 7 6 40 40 PC1_pm1_du2 w=60u L=1.4u
m62 7 6 0 0 PC2_nm1_du1 W=20u L=1.4u
m71 8 7 40 40 PC1_pm1_du2 w=60u L=1.4u
m72 8 7 0 0 PC2_nm1_du1 W=20u L=1.4u
m81 9 8 40 40 PC1_pm1_du2 w=60u L=1.4u
m82 9 8 0 0 PC2_nm1_du1 W=20u L=1.4u
m91 10 9 40 40 PC1_pm1_du2 w=60u L=1.4u
m92 10 9 0 0 PC2_nm1_du1 W=20u L=1.4u
m101 11 10 40 40 PC1_pm1_du2 w=60u L=1.4u
m102 11 10 0 0 PC2_nm1_du1 W=20u L=1.4u
m111 12 11 40 40 PC1_pm1_du2 w=60u L=1.4u
m112 12 11 0 0 PC2_nm1_du1 W=20u L=1.4u
m121 13 12 40 40 PC1_pm1_du2 w=60u L=1.4u
m122 13 12 0 0 PC2_nm1_du1 W=20u L=1.4u
m131 14 13 40 40 PC1_pm1_du2 w=60u L=1.4u
m132 14 13 0 0 PC2_nm1_du1 W=20u L=1.4u
m141 15 14 40 40 PC1_pm1_du2 w=60u L=1.4u
m142 15 14 0 0 PC2_nm1_du1 W=20u L=1.4u
m151 16 15 40 40 PC1_pm1_du2 w=60u L=1.4u
m152 16 15 0 0 PC2_nm1_du1 W=20u L=1.4u
m161 17 16 40 40 PC1_pm1_du2 w=60u L=1.4u
m162 17 16 0 0 PC2_nm1_du1 W=20u L=1.4u
m171 18 17 40 40 PC1_pm1_du2 w=60u L=1.4u
m172 18 17 0 0 PC2_nm1_du1 W=20u L=1.4u
m181 19 18 40 40 PC1_pm1_du2 w=60u L=1.4u
m182 19 18 0 0 PC2_nm1_du1 W=20u L=1.4u
m191 20 19 40 40 PC1_pm1_du2 w=60u L=1.4u
m192 20 19 50 0 PC2_nm1_du1 W=20u L=1.4u
m201 21 20 40 40 PC1_pm1_du2 w=60u L=1.4u
m202 21 20 0 0 PC2_nm1_du1 W=20u L=1.4u
m211 22 21 40 40 PC1_pm1_du2 w=60u L=1.4u
m212 22 21 0 0 PC2_nm1_du1 W=20u L=1.4u
*
* Capacitive Loading.
*
c2 2 0 0.1pF
c3 3 0 0.1pF
c4 4 0 0.1pF
c5 5 0 0.1pF
c6 6 0 0.1pF
c7 7 0 0.1pF
c8 8 0 0.1pF
c9 9 0 0.1pF
c10 10 0 0.1pF
```

Fig. 5.4 SPICE3 input deck for a 21-stage CMOS inverter chain with substrate current, gate current, device lifetime, and circuit aging calculations requested (continued on next page).

```
c11 11 0 0.1pF
c12 12 0 0.1pF
c13 13 0 0.1pF
c14 14 0 0.1pF
c15 15 0 0.1pF
c16 16 0 0.1pF
c17 17 0 0.1pF
c18 18 0 0.1pF
c19 19 0 0.1pF
c20 20 0 0.1pF
c21 21 0 0.1pF
c22 22 0 0.1pF
*
* Numerical Control.
*
.nodeset v(1)=0 v(2)=5 v(3)=0 v(4)=5 v(5)=0 v(6)=5
+ v(7)=0 v(8)=5 v(9)=0
+ v(10)=5 v(11)=0  v(12)=5 v(13)=0 v(14)=5 v(15)=0
+ v(16)=5 v(17)=0 v(18)=5 v(19)=0 v(20)=5 v(21)=0
*
* For uniform aging of all transistors, the period is 10ns, but
* the SPICE time window is 20ns long to allow the pulse to clear
* the last transistor.  The age is modified accordingly so that 10years
* of aging is equivalent to 20years in the SPICE input deck.
*
.tran 0.02ns 20ns
*
* Output Control.
*
.print tran v(21)
.width out=80
*
* Model parameter file declarations.
*
.process PC1 filename = PM0OUT
.process PC2 filename = NM0OUT
.pmosdeg PC1_pm1_du2 g1=0.7 ups=5e-5 hg0=1e3 mg0=1.5 wg=1
*
* Isub, Igate and lifetime commands.
*
.isubwidth=80
.deltaid 0.1
.agedid 20years
.plotisub m202 all
.plotigate m201
*
* Aging Commands.
*
.agemethod interp linlog
.age 20years
.ageproc PC2 filenames = NM0OUT, NM1OUT, NM2OUT, NM3OUT, NM4OUT
.end
```

Fig. 5.4 (cont.) SPICE3 input deck for a 21-stage CMOS inverter chain with substrate current, gate current, device lifetime, and circuit aging calculations requested.

```
-------------------------------------------------------------------------
|
|       DEVICE DEGRADATION INFORMATION : TRANSISTOR M201
|
-------------------------------------------------------------------------
|
| AVERAGE IDRAIN                              =    1.8070955e-04 A
| MAXIMUM IDRAIN                              =    5.0016691e-03 A
| AVERAGE ISUB                                =    3.3965421e-09 A
| MAXIMUM ISUB                                =    3.7613535e-07 A
| AVERAGE IGATE                               =    1.8173783e-14 A
| MAXIMUM IGATE                               =    2.5329937e-12 A
| DELTA ID / ID0 IN THE FIRST TIME PERIOD     =    2.6738449e-17
|
-------------------------------------------------------------------------
|  DEVICE LIFETIME AT DELTA ID / ID0 = 0.1:
-------------------------------------------------------------------------
|
|   >>>>>> TAU(m201) = 2.353e+07 YEARS ( 7.42e+14 SEC. ) <<<<<<
|
-------------------------------------------------------------------------
|  DEGRADATION OF M201 AT 6.31152e+08 SEC. (20.0137 YEARS):
-------------------------------------------------------------------------
|
|   >>>>>> DELTA ID0/ID = 6.47915e-06 <<<<<<
|
-------------------------------------------------------------------------

-------------------------------------------------------------------------
|
|       DEVICE DEGRADATION INFORMATION : TRANSISTOR M202
|
-------------------------------------------------------------------------
|
| AVERAGE IDRAIN                              =    1.6224191e-04 A
| MAXIMUM IDRAIN                              =    4.1909208e-03 A
| AVERAGE ISUB                                =    1.4233360e-07 A
| MAXIMUM ISUB                                =    1.4238605e-05 A
| DELTA ID / ID0 IN THE FIRST TIME PERIOD     =    9.8720072e-06
|
-------------------------------------------------------------------------
|  DEVICE LIFETIME AT DELTA ID / ID0 = 0.1:
-------------------------------------------------------------------------
|
|   >>>>>> TAU(m202) = 2.812 YEARS ( 8.869e+07 SEC. ) <<<<<<
|
-------------------------------------------------------------------------
|  DEGRADATION OF M202 AT 6.31152e+08 SEC. (20.0137 YEARS):
-------------------------------------------------------------------------
|
|   >>>>>> DELTA ID0/ID = 0.165263 <<<<<<
|
-------------------------------------------------------------------------
```

Fig. 5.5 Degradation information of the NMOS and PMOS devices of the 20th stage of the 21-stage inverter chain.

Device Name	Model Name	Age
m1	pc1 pm1 du2	2.746207e-08
m2	pc2 nm1 du1	3.197151e-05
m21	pc1 pm1 du2	3.879195e-08
m22	pc2 nm1 du1	7.242949e-04
m31	pc1 pm1 du2	3.667386e-08
m32	pc2 nm1 du1	7.449220e-04
m41	pc1 pm1 du2	3.576101e-08
m42	pc2 nm1 du1	8.931114e-04
m51	pc1 pm1 du2	3.646031e-08
m52	pc2 nm1 du1	1.062909e-03
m61	pc1 pm1 du2	4.441195e-08
m62	pc2 nm1 du1	9.446778e-04
m71	pc1 pm1 du2	3.620795e-08
m72	pc2 nm1 du1	1.241809e-03
m81	pc1 pm1 du2	3.997718e-08
m82	pc2 nm1 du1	1.011967e-03
m91	pc1 pm1 du2	4.236287e-08
m92	pc2 nm1 du1	6.358872e-04
m101	pc1 pm1 du2	3.904291e-08
m102	pc2 nm1 du1	8.521347e-04
m111	pc1 pm1 du2	3.346978e-08
m112	pc2 nm1 du1	7.083170e-04
m121	pc1 pm1 du2	4.031306e-08
m122	pc2 nm1 du1	1.074169e-03
m131	pc1 pm1 du2	3.771089e-08
m132	pc2 nm1 du1	1.099252e-03
m141	pc1 pm1 du2	3.947127e-08
m142	pc2 nm1 du1	1.103625e-03
m151	pc1 pm1 du2	3.421533e-08
m152	pc2 nm1 du1	8.793844e-04
m161	pc1 pm1 du2	2.859180e-08
m162	pc2 nm1 du1	1.116071e-03
m171	pc1 pm1 du2	4.571724e-08
m172	pc2 nm1 du1	9.507527e-04
m181	pc1 pm1 du2	3.002879e-08
m182	pc2 nm1 du1	1.294368e-03
m191	pc1 pm1 du2	3.450917e-08
m192	pc2 nm1 du1	9.508127e-04
m201	pc1 pm1 du2	3.023158e-08
m202	pc2 nm1 du1	8.830805e-04
m211	pc1 pm1 du2	4.229153e-08
m212	pc2 nm1 du1	1.319330e-03

Fig. 5.6 The agetable generated by CAS of all the transistors in the circuit.

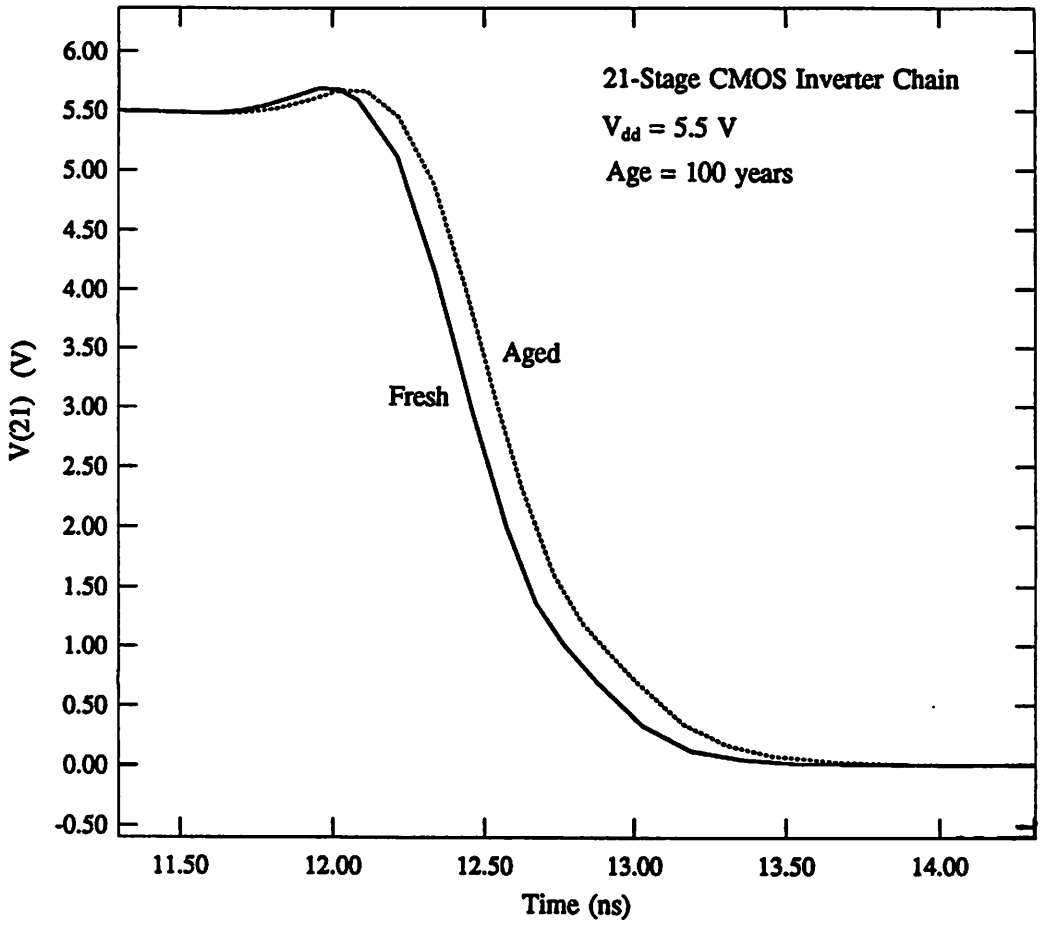

Fig. 5.7 The voltage waveform at the output of the 20th stage showing the propagation delay difference between the fresh and aged inverter chain.

VI. BERT SHELL SCRIPT PROGRAM FOR UNIX ENVIRONMENTS

A shell script program for BERT has been developed for use in a UNIX environment. A menu-driven system enables the user to choose the desired simulation without having to enter the lengthy piping commands. All operations are automated for convenience and speed. In addition, an option is added to iteratively simulate the circuit so that ongoing degradation can be taken into account.

To call the shell, simply type

bert < *input file* > < *output file* >

Specifying the input and output file in the command line is optional; the shell will prompt the user to enter them if they are not specified.

Fig. 6.1 shows the main menu. Seven different options are available depending on whether a one-pass simulation (such as calculating the degradation information of Fig. 5.5, or doing CORS or electromigration simulation) or a CAS-type simulation (multiple-pass circuit aging) is desired, as well as whether SPICE2, SPICE3B1, or SPICE3C1 is used. The menus and options for SPICE2 and the two SPICE3 versions are identical and are no different in operation. The transistor declarations for the BSIM1 model however, is different (as described in the previous section and in the SPICE manuals), so that the input file must be altered when switching between SPICE2 and SPICE3 versions. Finally, the seventh option enables the user to exit the program.

Fig. 6.2 shows the menu when the one-pass option is selected. The first option allows the user to alter the input file by entering the UNIX "vi" editor, while the second option permits the user to use entirely different input and output files. Option (3) makes it possible to call and use model parameter files from a different directory than the one in which the simulation is done. This allows the user to store all his model parameter files in one directory while switching from directory to directory for his simulations. Option (4) starts the actual simulation, option (5) returns the user to the main menu (Fig. 6.1), and option (6) exits the shell.

Fig. 6.3 shows the screen format when option (3) (selecting a new path for the parameter files) is

chosen. Presently, the user can customize his shell by writing in four different often-used paths in the shell code. Path (1) is the default path that is active whenever the shell program is started. The user can also enter an entirely new path (option (5)). This path, however, will not be stored when the shell is exited. Option (6) allows the user to stay with the present path listed at the top of the screen.

After doing all the necessary adjustments, the user can select option (4) in the one-pass menu (Fig. 6.2) to start the simulation. While the programs are running, the present status of the execution is successively displayed until the END OF SIMULATION menu appears (Fig. 6.4). Here, the user has the choice of viewing the newly created output file, going back to the main menu (Fig. 6.1), or exiting the shell altogether.

When one of the CAS options is selected from the main menu, a menu similar to the one-pass menu is displayed (Fig. 6.5). All options are identical, except for option (2). This option enables iterative simulation so that ongoing degradation can be taken into account. For instance, the user may want to simulate his circuit 10 years in the future. He may iterate only once so that the aged process files created by CAS are directly based on the degradation that occurred in the fresh circuit, or he may subdivide the 10 years into, for example, 10 intervals equally spaced in log time, so that each CAS simulation will generate model parameter files that have aged for an intermediate length of time. The aged model parameters of the first simulation is used by the next CAS simulation to produce the next set of aged model parameters files. This cycle is continued progressively until the 10 years is reached. In this way, the change in circuit degradation from continually changing device characteristics can be taken into account. Greater accuracy can undoubtably be achieved with a larger number of iterations, but with a sacrifice in speed and CPU time. In the present version, iterative simulation is only permitted with the BSIM1 (SPICE3 Level 4) model.

Once the simulation is started by selection option (5) from the CAS menu (Fig. 6.5), diagnostics similar to the one-pass case are displayed showing the present status of the simulation, with an END OF SIMULATION menu again appearing when program execution is completed (Fig. 6.6). The same options as in the one-pass case are present, except that the user can now view the output files of both the fresh and aged circuit.

Once the shell script is exited, all temporary files used by the shell and the pre- and post-processors are erased. The input file, the fresh and aged output files, the agetable of each iteration, and the aged model parameter files remain. The fresh output file can be identified by a ".fr" suffix added to the name of the output file specified by the user. A word of caution. The BERT system uses temporary files beginning with "raw" and "age", both in lower and upper cases. The user should avoid naming his personal files matching this pattern, as these files will be overwritten and erased when BERT is exited.

Finally, a note on installing the shell script program. The shell script itself is a normal text file and can be copied directly into the desired directory for use. Two additional items, however, must be taken care of by the user. These involve manually modifying the shell script itself. Both modifications are at the beginning of the program, and directions are contained in the listing (Fig. 6.7). The first is to specify the location of the various programs required to run BERT. The relevant paths are entered in the third column of text in the "alias" statements. The various programs include the BERT pre- and post-processor and the SPICE circuit simulator, as well as a collection of programs that are used exclusively by the BERT shell script. All necessary executable files except SPICE are placed in the bert/exe directory once the makefile is executed (Section 5.3). The second modification is to set the paths for the location of the process files that will appear in the path selection menu (Fig. 6.3). The text after the equal sign in the "set PfDirx = " statements should appropriately be replaced by the desired paths. Note that double quotes must surround the path listing.

Once these additional items are done, BERT can be used immediately.

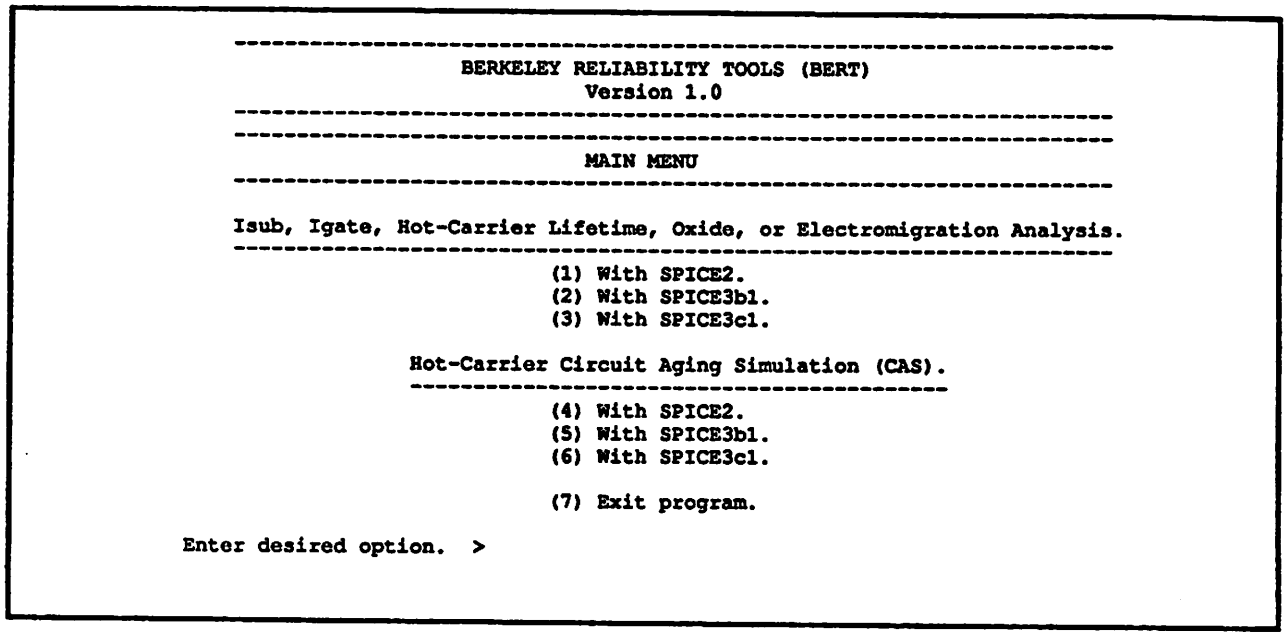

Fig. 6.1 The initial main menu of the shell script program.

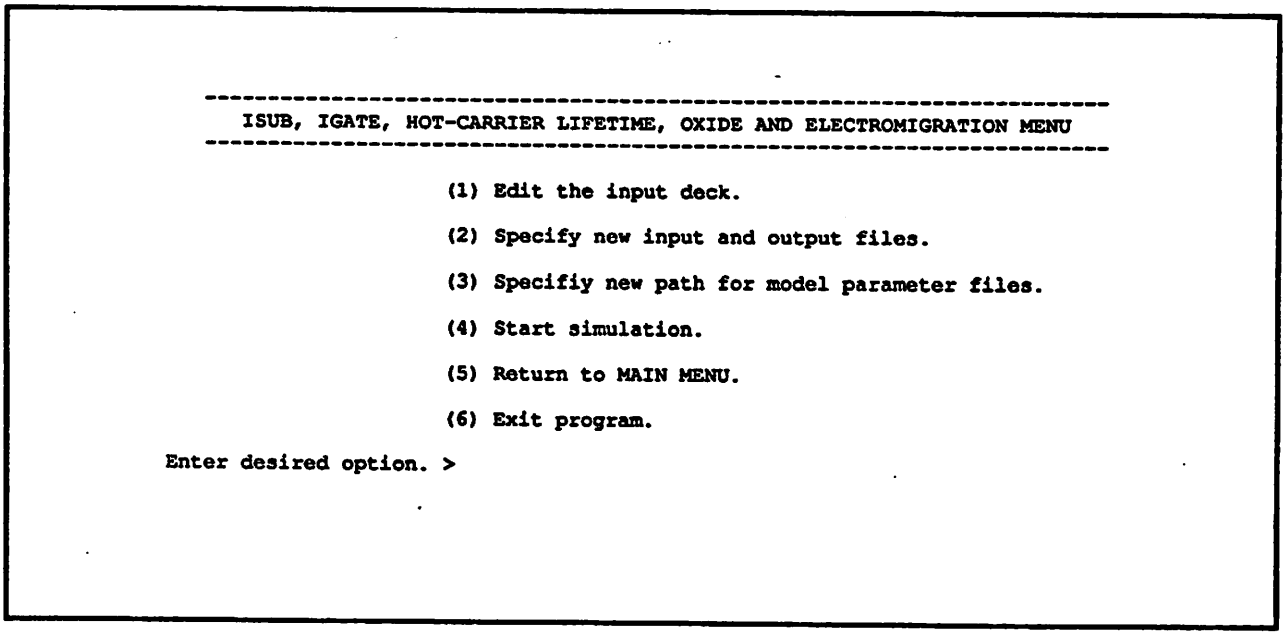

Fig. 6.2 The one-pass menu (option (1), (2), or (3) of the main menu).

```
Present Path =  /usr/tmp/bert/CAS/Pfiles

Choose new path from the following :

(1)  /usr/tmp/bert/CAS/Pfiles
(2)  /users1/users/pml/CAS/Pfiles/General
(3)  /users1/users/pml/CAS/Pfiles
(4)  /users1/users/pml/CAS/Pfiles/NMOS5
(5) Set new path.
(6) Remain with present path.

Enter desired option. >
```

Fig. 6.3 Changing the path of the location of the model parameter files (option (3) of the one-pass menu).

```
<<<<<< SIMULATION IN PROGRESS >>>>>>>

>>>> Pre-processing finished. Executing SPICE3..... <<<<

>>>> SPICE3 finished. Executing the post processor..... <<<<

------------------------------------------------------------------------
                          END OF SIMULATION
------------------------------------------------------------------------

                      (1) Edit the output file.

                      (2) Return to Main Menu.

                      (3) Exit program.

Enter desired option. >
```

Fig. 6.4 One-pass simulation diagnostics and END OF SIMULATION menu.

```
-------------------------------------------------------------------------
                              CAS MENU
-------------------------------------------------------------------------

                (1) Edit the input deck.

                (2) Select the number of iterations desired for
                    intermediate aged process file calculations.
                    *** For BSIM1 (SPICE3 Level 4) model only. ***
                    (Present Value = 1)

                (3) Specify new input and output files.

                (4) Specify new path for model parameter files.

                (5) Start simulation.

                (6) Return to MAIN MENU.

                (7) Exit program.

Enter desired option. >
```

Fig. 6.5 The CAS menu (option (4), (5) or (6) of the main menu).

```
                <<<<<< CAS SIMULATION IN PROGRESS >>>>>>>

**** Length of aging for Simulation No. 1 is 3.16228 years(s) ****

                >>> Simulation No.  1 . <<<
          Pre-processing finished.  Executing SPICE3....
          SPICE3 finished.  Executing the post-processor....
          Post-processing finished. Creating aged process files....

**** Length of aging for Simulation No. 2 is 6.83772 years(s) ****

                >>> Simulation No.  2 . <<<
          Pre-processing finished.  Executing SPICE3....
          SPICE3 finished.  Executing the post-processor....
          Post-processing finished. Creating aged process files....

                SIMULATING AGED INPUT DECK....
            Pre-processing finished.  Executing SPICE3....
            SPICE3 finished.  Executing the post processor....

-------------------------------------------------------------------------
                          END OF SIMULATION
-------------------------------------------------------------------------

                (1) Edit the aged output file.

                (2) Edit the fresh output file.

                (3) Return to Main Menu.

                (4) Exit program.

Enter desired option. >
```

Fig. 6.6 CAS simulation diagnostics and END OF SIMULATION menu.

```
#!/bin/csh -f
################################################################################
#                                                                              #
#                        BERKELEY RELIABILITY TOOLS (BERT)                     #
#                             SHELL SCRIPT PROGRAM                             #
#                                                                              #
#                                 Version 1.0                                  #
#                               By Peter M. Lee                                #
#          Department of Electrical Engineering and Computer Sciences          #
#                      University of California, Berkeley                      #
#                               January 8, 1990                                #
#                                                                              #
#     This program runs a shell script for use with the pre- and post-         #
# processing system of the BErkeley Reliability Tools (BERT).  With this       #
# shell script, automatic execution of the single-pass Isub, Igate, hot-       #
# carrier lifetime, oxide (CORS), and/or electromigration simulations can be   #
# done, as well as the muliple pass Circuit Aging Simulator (CAS) type         #
# simulations.                                                                 #
#                                                                              #
# Copyright (c) 1988, 1989, 1990   Peter M. Lee  All rights reserved.          #
#                                                                              #
################################################################################
#
# Edit the following nine lines to set the correct path for the various
# simulation programs.
#
alias prebert    /usr/tmp/bert/exe/prebert
alias spice2     /usr/cad/spice2
alias spice3b1   /usr/cad/spice3b
alias spice3c1   /usr/cad/spice3
alias postbert   /usr/tmp/bert/exe/postbert
alias CopyProc   /usr/tmp/bert/exe/copyproc
alias DelProc    /usr/tmp/bert/exe/delproc
alias AgeFilter /usr/tmp/bert/exe/agefilt
alias AgeConv    /usr/tmp/bert/exe/ageconv
alias ConvInp    /usr/tmp/bert/exe/convinp
#
# Put the paths of the location of your process files equal to the
# variables PfDir1 through PfDir4.
#
set PfDir1 = "/usr/tmp/bert/CAS/Pfiles"
set PfDir2 = "/users1/users/pml/CAS/Pfiles/General"
set PfDir3 = "/users1/users/pml/CAS/Pfiles"
set PfDir4 = "/users1/users/pml/CAS/Pfiles/NMOS5"
#
################################################################################
# Do not modify below this line.                                               #
################################################################################
clear
set PfDir = $PfDir1
```

Fig. 6.7 The first several lines of the shell script program. To customize the shell, the user must modify the "alias" statements and the "set PfDirx" statements as described in the text.

VII. CONCLUSION

We have presented a hot-electron reliability simulator CAS which is a part of the BERT reliability simulator system. Used in conjunction with the SPICE circuit simulator, CAS can calculate various degradation information for individual devices in a circuit undergoing dynamic operation. For instance, by using the device lifetime option, hot spots in the circuit can be easily pin-pointed. More importantly, CAS can predict the behavior of circuits that have undergone hot-carrier degradation for a user-specified length of time. With this tool, VLSI design engineers will be able to better understand the degradation and reliability performance of their circuits.

VIII. REFERENCES

[Cha84] T.Y. Chan, P.K. Ko, and C. Hu, "A simple method to characterize substrate current in MOSFET's," *IEEE Electron Device Letters*, Vol. EDL-5, No. 12, pp. 505-507, December 1984.

[Sod84] Charles Sodini, Ping-Keung Ko, and John L. Moll, "The effect of high fields on MOS device and circuit performance," *IEEE Trans. Electron Devices*, Vol. ED-31, No. 10, pp. 1386-1393, October 1984.

[Tam84] Simon Tam, Ping-Keung Ko, and Chenming Hu, "Lucky-electron model of channel hot-electron injection in MOSFET's," *IEEE Trans. Electron Devices*, Vol. ED-31, No. 9, pp. 1116-1125, September 1984.

[Hu85] 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. Electron Devices, Vol. ED-32, pp. 375-385, February 1985.*

[Lee86] Peter M. Lee, "BSIM - Substrate current modeling," University of California, Berkeley, Electronics Research Laboratory Memorandum UCB/ERL M86/49, June 1986.

[Wer86] W. Weber, C. Werner, and A.V. Schwerin, "Lifetimes and substrate currents in static and dynamic hot-carrier degradation," in *IEDM Tech. Digest*, pp. 390-393, December 1986.

[Aur87] Shian Aur, Dale E. Hocevar, and Ping Yang, "Circuit hot electron effect simulation," *IEDM Tech. Digest*, pp. 498-501, December 1987.

[Jen87] M.-C. Jeng, P.M. Lee, M.M. Kuo, P.K. Ko, and C. Hu, "Theory, algorithms, and user's guide for BSIM and SCALP," University of California, Berkeley, Electronics Research Laboratory Memorandum UCB/ERL M87/35, May 1987.

[Kuo87a] Mary Mei-Lin Kuo, "Implementation of the BSIM substrate current and degradation models in SCALP," Master's Thesis, May 1987.

[Kuo87b] M.M. Kuo, K. Seki, P.M. Lee, J.Y. Choi, P.K. Ko, and C. Hu, "Quasi-static simulation of hot-electron-induced MOSFET degradation under AC (pulse) stress," *IEDM Tech. Digest*, pp.

47-50, December 1987.

[Kuo88] M.M. Kuo, K. Seki, P.M. Lee, J.Y. Choi, P.K. Ko, and C. Hu, "Simulation of MOSFET lifetime under AC hot-electron stress," *IEEE Trans. Electron Devices*, Vol. ED-35, No. 7, pp. 1004-1011, July 1988.

[Lee88] Peter M. Lee, Mary M. Kuo, Ping K. Ko, and Chenming Hu, "Circuit Aging Simulator (CAS)," *IEDM Tech. Digest*, pp. 134-137, December 1988.

[Aur89] S. Aur, "Kinetics of hot carrier effects for circuit simulation," in *Proc. IEEE Rel. Phys. Symp.*, pp. 88-91, April 1989.

[Bel89] R. Bellens, P. Heremans, G. Groeseneken, and H.E. Maes, "On the channel-length dependence of the hot-carrier degradation of n-channel MOSFETs," *IEEE Electron Device Letters*, Vol. 10, No. 12, pp. 553-555, December 1989.

[Ong89] Tong-Chern Ong, Koichi Seki, Ping K. Ko, and Chenming Hu, "P-MOSFET gate current and device degradation", *Proc. IEEE Rel. Phys. Symp.*, pp. 178-182, March 1989.

[Ros89] Elyse Rosenbaum, Peter M. Lee, Reza Moazzami, P.K. Ko, and C. Hu, "Circuit reliability simulator - oxide breakdown module," *IEDM Tech. Digest*, pp. 331-334, December 1989.

[Lee90] Peter M. Lee, Paul G.Y. Tsui, Ping K. Ko, Chenming Hu, Frank K. Baker, and Jim D. Hayden, "Simulation of hot carrier degradation in CMOS inverter-based VLSI circuits," submitted to the *IEEE Symposium on VLSI Technology*, June 1990.

[Lie90a] B.K. Liew, Peng Fang, N.W. Cheung, and C. Hu, "BERT - Circuit Electromigration Reliability Simulator", University of California, Berkeley, Electronics Research Laboratory Memorandum UCB/ERL M90/3, January 1990.

[Lie90b] B.K. Liew, N.W. Cheung, and C. Hu, "Reliability simulator for interconnect and intermetallic contact electromigration," to be presented at the *Proc. IEEE Rel. Phys. Symp.*, March 1990.

[Ros90] Elyse Rosenbaum, Peter M. Lee, Reza Moazzami, Ping K. Ko, and Chenming Hu, "BERT - Circuit Oxide Reliability Simulator (CORS)", University of California, Berkeley, Electronics Research Laboratory Memorandum UCB/ERL M90/4, January 1990.

[Tsu90] Paul G.Y. Tsui, Lee Howington, Peter M. Lee, Tom Tiwald, Brian Mowry, Frank K. Baker, Jim D. Hayden, Bill B. Feaster, and Bob Garbs, "An integrated system for circuit level hot carrier evaluation," to be presented at the *Custom Integrated Circuits Conference*, May 1990.

APPENDIX: CAS Error Messages

The following list contains the error messages of the pre- and post-processor of BERT-CAS and the BERT shell script including the routine name in which they occur. Error codes in the 'Cxx:' format are CAS errors, those in the 'Bxx:' format are BERT shell script errors. Furthermore, for CAS, two-digit codes represent pre-processing errors while three-digit codes represent post-processing errors. Error codes for CORS and the electromigration simulator are in the format 'Txx:' and 'Exx:', respectively. See [Ros90] and [Lie90a] for a list of error messages for the two simulators.

BERT-CAS Pre-processor Errors:

prebert.c:

ArgU:

C01: No input file specified!
C02: Cannot open input file!
C03: Specified option not valid!
C04: Incorrect option or file specification!

PreFilter:

C05: Missing .process command in the input deck!
C06: Missing .ageproc command in the input deck!
C07: Missing .age command in the input deck!

FindIsub:

C08: Invalid .printisub or .plotisub command!

FindIgate:

C09: Invalid .printigate or .plotigate command!

GetDelta:

C10: No lifetime criteria given for the .deltavt command!
C11: No lifetime criteria given for the .deltaid command!
C12: No lifetime criteria given for the .deltagm command!

GetAge:

C13: No future time given for the .age command!
C14: Incorrect format for the future time given in the .age command!

AgeDeg:

C15: No future time given for the <command> command!

C16: Incorrect format for the future time given in the <command> command!

FindProc:

C17: Cannot open rawinp1 file!

C18: Insufficient memory space. Reduce the number of model parameter files!

C19: Incorrect .process command format!

C20: Too many model parameter files!

C21: No model parameter file(s) specified!

C22: Missing or incorrect model parameter filename specified!

CreateInpFile:

C23: Insufficient memory space. Too many transistors!

C24: Cannot open one of the rawmodel files!

FindIsubIgateOut:

C25: Invalid .printisub, .plotisub, .printigate, or .plotigate command!

SubstituteLine:

C26: No transistor model name specified!

procsub.c:

Proc2ModSub:

C27: Insufficient memory space!

C28: Cannot open model parameter file <model parameter filename>!

C29: Error in reading model parameter file!

C30: Illegal header line in model parameter file!

CreateRawprocess:

C31: Insufficient memory space!

C32: Illegal header line in model parameter file!

C33: Parameters for BSIM1 model missing in model parameter file!

ChkModel:

C34: Cannot open model parameter file <model parameter filename>!

C35: Cannot write into temporary model parameter file <model parameter filename>!

C36: No MOS model parameters in the specified model parameter file!

getdata:

C37: Premature end of file reading BSIM1 model parameter file!

premisc.c:

OpenInpFile:

C38: Cannot open rawinp1 file!

OpenRaw:

C39: Cannot open rawsub file!

getvalue:

C40: Insufficient memory space in reading in BSIM1 parameters!

C41: Premature end of file reading BSIM1 model parameter file!

age.c:

MemAlloc:

C42: Insufficient memory space!

GetAgecards:

C43: Insufficient memory space!

C44: Incorrect .ageproc command format!

C45: Insufficient memory space. Too many model parameter files!

GetDevAge:

C46: Cannot open agetable!

C47: Insufficient memory space!

C48: Insufficient memory space. Too many aged transistors!

ReadAgePar:

C49: Insufficient memory space!

C50: Incorrect .process command format!

C51: Cannot open model parameter file <model parameter filename>!

C52: Cannot open .ageproc model parameter file <model parameter filename>!

C53: Illegal header line in model parameter file <model parameter filename>!

C54: Mixture of SPICE and BSIM models in same .ageproc command not allowed!

BSIMGetParm:

C55: Insufficient memory space!

C56: Illegal header line in model parameter file <model parameter filename>!

SPICEGetParm:

C57: Insufficient memory space!

C58: Invalid model name declared in model parameter file!

C59: Invalid model type declared in model parameter file!

ParmExt:

C60: Not enough pre-stressed model parameter files for model <model name>!

C61: Not enough pre-stressed model parameter files for model <model name>!

GenInpDeck:

C62: Insufficient memory space!

bsimext.c:

GetWLparm:

C63: Insufficient memory space. Too many model parameter files!

BSRegress:

C64: Not enough pre-stressed model parameter files!

PreRegress:

C65: Unable to do log-log regression to find aged parameters!

BSInterp:

C66: Insufficient memory space!

PreInterp:

C67: Pre-stressed model parameter files not ordered from least to most aged!

C68: Method of interpolation not specified!

leastsq2:

C69: Least square approximation failed due to bad parameter data!

leastsq2_2vars:

C70: Least square approximation failed due to bad parameter data!

leastsq3b:

C71: Least squares approximation reduction failed due to small pivot!

spext.c:

SPICERegress:

C72: Insufficient memory space!

C73: Not enough pre-stressed model parameter files!

SPICEInterp:

C74: Insufficient memory space!

SPGenAgeParm:

C75: Cannot write aged model parameter files in present directory!

BERT-CAS Post-processor errors:

postbert.c:

main:

C100:Could not open SPICE output file!

C101:Could not create 'rawout' file!

C102:Cannot open rawout1 file!

ErrorCheck:

C103:Cannot open rawsub file!

SubAnalysis:

C104:Cannot open agetable!

C105:Insufficient memory space!

C106:Could not open 'Itest' file!

AddSubParam:

C107:Insufficient memory space. Too many model parameter files!

C108:BSIM1 interconnect model parameters could not be found!

C109: .END command is missing from the input file!

MemAlloc:

C110:Timestep too small in reading voltage values!

C111:Insufficient memory space. Too many timesteps!

ReadVoltage:

C112:Voltage printout for substrate current analysis not found!

C113:Division by zero in reading voltages!

C114:Timesteps too small in reading voltages!

readpar.c:

FindInfo:

C115:Invalid spice type specification!

C116: .tran card missing!

ObtainTrans:

C117:Insufficient memory. Too many Isub- or Igate-requested transistors!

C118:Insufficient memory space. Too many transistors!

ObtainModelCards:

C119:Cannot open rwmd<x> file!

C120:Insufficient memory!

C121:Insufficient memory. Too many model parameter files!

BSIMsetup:

C122:Division by zero in BSIM1 parameter calculation!

SPICEsetup:

C123:Nsub < Ni!

C124:Effective channel length less than zero!

ObtainPMOSDegPar:

C142:Insufficient memory!

C125:Invalid parameter in .pmosdeg command!

bsim1.c:

BSIMevaluate:

C126:Phi is negative in BSIM1 (Level 4) model!

C127:Phi = 0 in BSIM1 (Level 4) model!

C128:Vdd = 0 in BSIM1 (Level 4) model!

C128:Vdd = 0 in BSIM1 (Level 4) model!

C129:Non-positive mobility given in BSIM1 (Level 4) model!

degcalc.c:

BSIMDeltaVth:

C130:Degradation of transistor m<xx> too large!

output.c:

PlotSubCurrent:

C131:Insufficient memory space. Too many timesteps!

C132:Timestep too small to plot substrate current!

C133:Substrate current too large to plot!

PrintSubCurrent:

C134:Insufficient memory space. Too many timesteps!

C135:Timestep too small to print substrate current!

PlotGateCurrent:

C136:Insufficient memory space. Too many timesteps!

C137:Timestep too small to plot gate current!

C138:Gate current too large to plot!

PrintGateCurrent:

C139:Insufficient memory space. Too many timesteps!

C140:Timestep too small to print gate current!

postmisc.c:

OpenRaw:

C141:Cannot open the rawsub file!

BERT Shell Script Errors:

ageconv.c:

main:

B01: Cannot open agetable!
B02: Cannot open rawagetable!
B03: Cannot create rawtempage file!

agefilt.c:

main:

B04: Cannot open SPICE input file!
B05: Cannot create SPICE input file!
B06: Temporary file is missing!
B07: Cannot open temporary file!

convinp.c:

main:

B08: Cannot open input file!
B09: Cannot open inpdeck file!
B10: Cannot create intermediate file!
B11: Not enough memory!
B12: Improper age given in .age command!

ChangeModelName:

B13: Model parameter file not found!
B14: Cannot write into directory!

copyproc.c:

main:

B15: Cannot open input file!
B16: Cannot create intermediate file!
B17: Not enough memory!
B18: Premature end-of-file in input deck!

delproc.c:

main:

B19: Cannot open 'rawpfile'!