MODELING AND SIMULATION OF HOT-CARRIER EFFECTS IN MOS DEVICES AND CIRCUITS

by

Peter Maurice Lee

Memorandum No. UCB/ERL M90/30

30 April 1990

MODELING AND SIMULATION OF HOT-CARRIER EFFECTS IN MOS DEVICES AND CIRCUITS

by

Peter Maurice Lee

Memorandum No. UCB/ERL M90/30

30 April 1990

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

MODELING AND SIMULATION OF HOT-CARRIER EFFECTS IN MOS DEVICES AND CIRCUITS

by

Peter Maurice Lee

Memorandum No. UCB/ERL M90/30

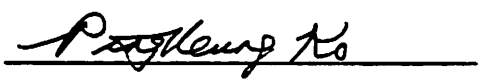
30 April 1990

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

MODELING AND SIMULATION OF HOT-CARRIER EFFECTS IN MOS DEVICES AND CIRCUITS

Ph.D. Peter Maurice Lee Department of EECS



Ping K. Ko
Committee Chairman

ABSTRACT

This dissertation presents a hot-carrier reliability simulator called BERT-CAS which can predict circuit performance degradation using device-level quasi-static models, starting from a parametric substrate current model and extending to the calculation of "aged" model parameters for transistors undergoing dynamic operation within a circuit. By using CAS, circuit designers can not only predict the degraded behavior of their circuits, but can also study which devices in the circuit experience the greatest degradation and which have the most effect to circuit output. Alternative circuit designs can be evaluated, and thus circuits more robust to hot-carrier effects can be designed.

From CAS simulations and experimental results (reported elsewhere and in this dissertation), it is found that device degradation correlates better with the degradation driving force $I_{ds}(I_{sub}/I_{ds})^m$ rather than with I_{sub} alone. Because CAS is based on the full degradation model rather than just I_{sub}, accurate prediction is achieved.

In general, a simulator such as CAS is necessary to predict circuit hot-carrier degradation from device-level concepts. However, for the special case of CMOS inverter-based circuits, a rough rule of thumb has been developed for quick estimation of circuit degradation from device-level stress tests.

A bipolar charge-storage phenomenon causing an extended substrate current flow is also presented. When a NMOSFET used as the driver device in an inverter enters the avalanche breakdown regime of

operation during an input low-to-high transient, a substantial amount of charge is seen to be generated as far as 20μm away from the transistor, with a subsequent long substrate current flow to drain the excess charge. This phenomenon can thus have adverse effects to neighboring structures. Device simulation results using a three-dimensional two-carrier simulator (CADDETH) are presented to study the phenomenon and to show its effects on CMOS latchup. This phenomenon also explains the fact that dynamic periodic inverter-based circuits can tolerate power supply voltages greater than the avalanche breakdown voltage of the individual devices, as long as the signal frequency is low enough.

Acknowledgments

First of all, I would like to extend my sincere appreciation and utmost gratitude to my research advisor, Professor Ping K. Ko. His guidance, patience, and understanding has made this research mature from its humble beginnings. I have emphatically enjoyed his company, both professionally and personally.

I would also like to extend my thanks to Professor Chenming Hu who has provided many additional insights in this research. Professor Hu has the ability of transforming research results by giving them useful, practical, and clear interpretations that can be understood by others. His stimulating discussions have taught me new ways of addressing research problems.

I would like to thank Professor Charles Stone of the Statistics Department for being one of my Qualifying Examination Committee members and for reviewing this dissertation.

Gratitude is due to my officemates, Dr. Min-Chie Jeng, Dr. Hong June Park, Mr. Jon Duster, Mr Clement Szeto, and Mr. Brad Johnson who have provided me with a pleasant atmosphere to work. Many discussions with Miss Elyse Rosenbaum, which resulted in numerous improvements to the CAS program, are greatly appreciated.

I would also like to thank my colleagues at the Central Research Laboratory, Hitachi Ltd., Tokyo, Japan, for their cooperation and help during my 14-month stay during which the research on substrate current transient delay was accomplished. Namely, I would like to thank Dr. Shojiro Asai for inviting me to stay at Hitachi, Dr. Hiroo Masuda, the unit leader of the CADDETH group, who guided me during the more technically difficult parts of the research, and Dr. Toru Toyabe, for advice concerning the CADDETH device simulator. Most of all, I am greatly indebted to Mr. Hitoshi Matsuo, who, being my mentor, helped me greatly concerning research matters as well as living conditions. I will always cherish his friendship. I would like to thank Dr. Masaaki Aoki for providing the devices used to measure the substrate current delay, and Mrs. Hiroko Shukuri who helped me immensely in obtaining the informative plots from CADDETH.

For the CAS verification using ring oscillator measurement results, I am indebted to my colleagues at the Advanced Products Research and Development Laboratory, Motorola, Austin, Texas. It was a stimulating and pleasant experience to work with the simulation and modeling unit of Mr. Bob Garbs. I am indebted to Mr. Frank K. Baker and Mr. Jim Hayden for providing the circuits and devices used for the measurements. I extend my greatest appreciation to Mr. Paul Tsui, who was invaluable both professionally and personally during my two-month stay at Motorola.

Major funding for this research was provided by the Semiconductor Research Corporation (Contract #52055 and #59214), Hitachi Ltd., Motorola, Inc., and the ISTO/SDIO administered by the ONR under Contract N00014-85-K-0603.

Finally, and definitely not least, I want to thank my parents, Kuniko and Maurice. Without their loving support throughout the years this research would not have been possible.

Table of Contents

CHAPTER 1: INTRODUCTION

Hot-carrier degradation has become an increasingly worrisome issue as transistor dimensions continue to shrink while the power supply voltage remains essentially constant at 5 V. Keeping the power supply constant insures compatibility with existing integrated circuits while achieving higher performance because of higher current driveability. Thus, these past several years have seen research concentrating on the device level to study and model hot-carrier degradation [Fai81, Mat81, Tak83a, Tak83b, Tak83c, Hsu84, Hof85, Hu85, Tsu85, Hor86, Bha86, Web86, Cho87abc, Tsu87, Cha88, Web88, Bel89, Chu89], and to design hot-carrier-resistant structures such as the Lightly-Doped-Drain (LDD) and Doubly-Diffused-Drain (DDD) devices. By understanding the mechanisms of hot-carrier degradation and using these degradation-resistant structures, devices with less than 1.0 μm channel lengths can be used at present power supply specifications.

However, as device dimensions continue to diminish to sub-half-micron designs, even hot-carrier-resistant structures exhibit degradation. Thus, the industry has established the lower standard power supply level of 3.3 V so that device degradation can be minimized once more. Now the question has become at what point justification can be made to make the jump to the lower power supply level. Converting to the lower supply voltage decreases circuit performance while making the circuit more susceptible to noise and process variations. Thus, it becomes desirable to maintain existing power supply levels as long as possible until circuit hot-carrier reliability necessitates a change. A simulator to asses circuit reliability can be a valuable tool for making an accurate judgement.

So far, because hot-carrier degradation research has concentrated on the device level, researchers have used device-related parameters such as drain current degradation, transconductance degradation, or threshold voltage shift to quantify and judge hot-carrier reliability. It remains unclear, however, how these device-level degradation changes actual circuit behavior. [Aur87] has demonstrated the varying sensitivity each transistor in the circuit may have to circuit output behavior. For example, a certain transistor M1 may experience 20% drain current degradation but affect the circuit output minimally,

while another transistor M2 may suffer only 5% degradation and yet cause substantial degradation in circuit performance. Furthermore, each transistor may have varying sensitivity depending upon what aspect of the circuit performance is considered critical. Trying to define an arbitrary level of device degradation (such as 10% current degradation) can be misleading and overly conservative or dangerously optimistic. In addition, to predict degraded circuit behavior, the degradation of the individual devices of the circuit subject to the dynamic circuit-determined waveforms must be predicted. This is in sharp contrast to the DC or uniform AC waveforms used to study device-level degradation behavior.

This dissertation discusses the modeling and simulation of hot-carrier effects as applied to circuits. Using device-level concepts, a quasi-static model is established that predicts circuit-level degradation. This model is incorporated into the Circuit Aging Simulator (CAS) , as part of the BErkeley Reliability Tools (BERT) [Lee88, Ros89,Lee90, Lie90, Ros90]. Chapters 2 and 3 describe the methodology and models used in CAS, beginning with the accurate modeling of the substrate current I_{sub} , to predicting degraded device parameters for the transistors in a circuit which has undergone a certain operating time. Chapter 4 discusses a rule-of-thumb that can be used to relate device-level degradation to performance of circuits based on a specific but well-used configuration, the CMOS inverter. Chapter 5 slightly digresses and presents measurement and simulation results of a bipolar charge storage affect that occurs when the drain voltage V_{ds} of an NMOSFET driver transistor in an inverter is high enough to be in the avalanche breakdown regime, and finally the dissertation ends with a conclusion in Chapter 6.

References

[Fai81] Richard B. Fair and Robert C. Sun, "Threshold-voltage instability in MOSFET's due to channel hot-hole emission," *IEEE Transactions on Electron Devices*, Vol. ED-28, No. 1, pp. 83-93, January 1981.

[Mat81] Heihachi Matsumoto, Kokichi Sawada, Sotoju Asai, Makoto Hirayama, and Koichi Nagasawa, "Effect of long-term stress on IGFET degradations due to hot electron trapping," *IEEE Transactions on Electron Devices*, Vol. ED-28, No. 8, pp. 923-928, August 1981.

[Tak83a] E. Takeda and N. Suzuki, "An empirical model for device degradation due to hot-carrier injection," *IEEE Electron Device Letters*, Vol. EDL-4, No. 4, pp. 111-113, April 1983.

[Tak83b] E. Takeda, Y. Nakagome, H. Kume, and S. Asai, "New hot-carrier injection and device degradation in submicron MOSFETs," *IEEE Proceedings*, Vol. 130, Part I, No. 3, pp. 144-149, June 1983.

[Tak83c] E. Takeda, A. Shimizu, and T. Hagiwara, "Role of hot-hole injection in hot-carrier effects and the small degraded channel region in MOSFET's," *IEEE Electron Device Letters*, Vol. EDL-4, No. 9, pp. 329-331, September 1983.

[Hsu84] F.-C. Hsu and S. Tam, "Relationship between MOSFET degradation and hot-electron-induced interface-state generation," *IEEE Electron Device Letters*, Vol. EDL-5, No. 2, pp. 50-52, February 1984.

[Hof85] Karl R. Hofmann, Christoph Werner, Werner Weber, and Gerhard Dorda, "Hot-electron and hole-emission effects in short n-channel MOSFET's," *IEEE Transactions on Electron Devices*, Vol. ED-32, No. 3, pp. 691-699, March 1985.

[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.

[Tsu85] T. Tsuchiya and J. Frey, "Relationship between hot-electron/holes and degradation of p- and n-channel MOSFET's," *IEEE Electron Device Letters*, Vol. EDL-6, No. 1, pp. 8-11, January 1985.

[Hor86] T. Horiuchi, H. Mikoshiba, K. Nakamura, and K. Hamano, "A simple method to evaluate device lifetime due to hot-carrier effect under dynamic stress," *IEEE Electron Device Letters*, Vol. EDL-7, No. 6, pp. 337-339, June 1986.

[Bha86] Anjan Bhattacharyya and Sunhil N. Shabde, "Degradation of short-channel MOSFET's under constant current stress across gate and drain," *IEEE Transactions on Electron Devices*, Vol. ED-33, No. 9, pp. 1329-1333.

[Web86] W. Weber, C. Werner, and A.V. Schwerin, "Lifetimes and substrate currents in static and dynamic hot-carrier degradation," *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.

[Cho87a] J.Y. Choi, P.K. Ko, and C. Hu, "Effect of oxide field on hot-carrier-induced degradation of metal-oxide-semiconductor field-effect transistors," *Applied Physics Letters*, Vol. 50, pp. 1188-1190, April 1987.

[Cho87b] J.Y. Choi, P.K. Ko, and C. Hu, "Hot-carrier-induced MOSFET degradation: AC versus DC stressing," *Proceedings of theSymposium on VLSI Technology*, pp. 45-46, May 1987.

[Cho87c] J.Y. Choi, P.K. Ko, and C. Hu, "Hot-carrier-induced MOSFET degradation under AC stress," *IEEE Electron Device Letters*, Vol. EDL-8, No. 8, pp. 333-335, August 1987.

[Tsu87] Toshiaki Tsuchiya, Toshio Kobayashi, and Shigeru Nakajima, "Hot-carrier-injected oxide region and hot-electron trapping as the main cause in Si nMOSFET degradation," *IEEE Transactions on Electron Devices*, Vol. ED-34, No. 2, pp. 386-391, February 1987.

[Cha88] T.Y. Chan, C.L. Chiang, and H. Gaw, "New insight into hot-electron-induced degradation of n-MOSFET's," *IEDM Technical Digest*, pp. 196-199, December 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.

[Web88] Werner Weber, "Dynamic stress experiments for understanding hot-carrier degradation phenomena," *IEEE Transactions on Electron Devices*, Vol. 35, No. 9, pp. 1476-1486., September 1988.

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

[Bel89] Rudi Bellens, Paul Heremans, Guido Groeseneken, and Herman E. Maes, "On the channel-length dependence of the hot-carrier degradation of n-channel MOSFET's," *IEEE Electron Device Letters*, Vol. 10, No. 12, pp. 553-555, December 1989.

[Chu89] J. Chung, M.-C. Jeng, J.E. Moon, P.K. Ko, and C. Hu, "Low-voltage hot-electron currents and degradation in deep-submicrometer MOSFETs," *Proc. IEEE Rel. Phys. Symp.*, pp. 92-97, April 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.

[Lie90] 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.

[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.

CHAPTER 2: SUBSTRATE CURRENT MODELING

2.1 Introduction

A substantial amount of research has been done correlating substrate current I_{sub} to device-level hot-carrier degradation, at least for the NMOSFET case (see refs. [Fai81] - [Chu89] on p. 1). Because both hot-carrier damage and I_{sub} are induced by the same force, the lateral electric field in the channel (or more notably the peak lateral field E_m), developing an accurate model for I_{sub} is a crucial first step in modeling degradation. For PMOSFETs, there seems to be no unanimous consensus as to whether I_{sub} or the gate current I_{gate} is the better degradation monitor, although a majority believe I_{gate} is the correct current to choose [Tsu85, Hir86, Tzo86, Bra88, Ong88, Ong89]. A preliminary model concerning PMOSFET degradation is discussed in a later chapter (See Section 3.4).

In this chapter, a parametric substrate current model is presented as part of the Berkeley Short-Channel IGFET Model Version 1.0 (BSIM1). 11 bias-independent parameters are added to the existing 20 to accurately simulate substrate current behavior for a variety of processes. In parallel with the other models of the BSIM1 family, each bias-independent parameter is then decomposed into three size-independent parameters which are used to create a single process file for each die. Presently, the model has been carefully evaluated for conventional and Lightly-Doped-Drain (LDD) NMOS enhancement-mode devices. A more detailed description of the general extraction process and a user's guide to the BSIM1 program are given in [Jen85, Jen87].

Section 2.2 contains the theoretical motivation in the development of the substrate current model used in BSIM1. Next, Section 2.3 gives a detailed description of the actual measurement and extraction process. Experimental results from LDD and non-LDD technologies are then summarized in Section 2.4. In Section 2.5, a brief user's guide to the operation of the substrate current measurement portion of the BSIM1 program is given, and finally a summary of the procedures added or modified to the source code is listed in Section 2.6.

2.2 Substrate Current Model

The substrate current model used in BSIM1 is based on work done by El-Mansy [Man75, Man77] and Ko [Ko81]. El-Mansy derived an exponential relationship of the channel electric field in the saturated region of the channel using quasi-two-dimensional concepts, and using this electric field model he derived a simplified model for the substrate current. This model was subsequently improved by Ko to include the effect of junction depth and channel doping. The following is an outline of the derivations done to obtain the substrate current expression. For a more detailed derivation, one is referred to the above references.

Because the main contribution to the substrate current is from electron impact ionization, the derivation is begun by integrating the electron impact ionization coefficient $\alpha_n = A_i \exp(- B_i/E)$ in the velocity-saturated region of the channel,

$$I_{sub} = I_{ds} A_i \int_{y=0}^{\Delta L} \exp(- B_i/E_s(y)) \, dy \qquad (2.2.1)$$

$y = 0$ is at the edge of the saturated region in the channel, $y = \Delta L$ is at the drain, $E_s(y)$ is the electric field in the channel direction, and A_i and B_i are constants. To find $E_s(y)$, a pseudo-two-dimensional analysis is performed of a Gaussian box enclosing the saturated region. This results in an exponential relationship of $E_s(y)$ versus distance,

$$E_s(y) = E_{crit} \cosh\left[\frac{y}{l_c}\right] \qquad (2.2.2)$$

E_{crit} is the critical field for velocity saturation, and l_c can be termed as the effective width of the "pinch-off" region of the channel. $E_s(y)$ can also be expressed in terms of the voltage within the saturated region,

$$E_s(y) = \left[\frac{(V_s(y) - V_{dsat})^2}{l_c^2} + E_{crit}^2\right]^{1/2} \qquad (2.2.3)$$

After an appropriate change in variables, Eq. 2.2.1 can then be rewritten as

$$I_{sub} = I_{ds}\, l_c\, A_i \int_{E_s = E_{crit}}^{E_d} \left[\frac{\exp(-B_i/E_s)}{E_s^2 - E_{crit}^2} \right]^{1/2} dE_s \qquad (2.2.4a)$$

$$\cong \frac{I_{ds}\, A_i\, l_c\, E_d}{B_i} \exp\left[-\frac{B_i}{E_d} \right] \qquad (2.2.4b)$$

where E_d is the electric field at the drain end,

$$E_d = \left[\frac{(V_{ds} - V_{dsat})^2}{l_c^2} + E_{crit}^2 \right]^{1/2} \qquad (2.2.5)$$

In saturation, $E_d >> E_{crit}$, so that Eq. 2.2.5 can be approximated by

$$E_d \cong \frac{V_{ds} - V_{dsat}}{l_c} \qquad (2.2.6)$$

Inserting Eq. 2.2.6 into Eq. 2.2.4b, we obtain the final expression for I_{bs},

$$I_{sub} = \frac{A_i}{B_i} I_{ds} (V_{ds} - V_{dsat}) \exp\left[-\frac{B_i\, l_c}{V_{ds} - V_{dsat}} \right] \qquad (2.2.7)$$

Since A_i and B_i are constants, two parameters remain to be determined, V_{dsat} and l_c.

For short-channel devices, V_{dsat} departs from the well-known relationship $V_{dsat} = V_{gs} - V_{th}$ for long-channel devices because electrons in the channel region become velocity-saturated before V_{ds} reaches $V_{gs} - V_{th}$. The model used to account for this behavior was derived by Sodini and Ko [Sod84],

$$V_{dsat} = \frac{E_{crit}\, L\, (V_{gs} - V_{th})}{E_{crit}\, L + (V_{gs} - V_{th})} \qquad (2.2.8)$$

where L is the channel length. E_{crit} is then extracted as a parameter from measured V_{dsat} values.

Several approximate analytical forms for l_c have been published, including those of El-Mansy et al. [Man75, Man77] and Ko [Ko81] with a $t_{ox}^{1/2}$ dependency, but none can comprehensively account for dependencies on bias and device size. In this work, a semi-empirical approach is established in which these dependencies are extracted directly from measured data.

It has been shown that by making liberal approximations, a relatively compact expression for the substrate current can be derived. In the next section, we will see how to implement this expression into a parameter extraction process that can accurately predict device behavior.

2.3 Parameter Extraction

2.3.1 Introduction

Although the parameter extraction of the substrate current used in BSIM1 is firmly based on theoretical analysis, the desire to achieve more accurate modeling created the need to find empirically-based expressions, especially to predict bias voltage and size dependencies. There is no exact expression for the substrate current, and because of the many approximations that had to be made in deriving the exponential model (Section 2.2), it was inevitable that accuracy would be sacrificed. Thus, the BSIM1 model uses the simple exponential model as a base to extract parameters that are generally bias- and device-size-dependent, and from this data BSIM1 produces bias- and size-independent parameters using empirically-determined analytical expressions.

The parameter extraction process is divided into two stages - one to extract l_c and the other to extract E_{crit}, from Eqs. 2.2.7 and 2.2.8. Bias voltages to extract the substrate current parameters have been carefully chosen so that leakage currents do not affect extraction. Extraction is performed for 5 V_{gs} values (starting at V_{th} + 0.3 to a maximum value of $0.8V_{dd}$ in equal increments, with V_{th} taken at maximum substrate bias) and four user-selectable V_{bs} values. A total of 48 measurements of both I_{bs} and I_{ds} are done in a time span of approximately 5 minutes. After the measurements are completed, all bias-independent parameters are extracted using a least-squares fit routine. These bias-independent parameters are then separated into length and width dependencies in a manner similar to those used in existing BSIM1 models after all desired devices of a single die are measured. The resulting 33 additional size-independent parameters are then simply appended to the standard process file.

2.3.2 Extraction of E_{crit}

An empirical method to determine V_{dsat} was proposed by [Cha84]. In that work, it was found that parallel contours would result by plotting constant I_{sub}/I_{ds} curves in I_{ds} - V_{ds} space, with V_{gs} as a parameter (Fig. 2.1). By parallel shifting the I_{sub}/I_{ds} contour so that it intersects the origin at I_{ds} = 0 and

$V_{ds} = 0$, the contour maps out a $V_{ds} = V_{dsat}$ loci on the I_{ds} - V_{ds} plane, with V_{gs} corresponding to the I_{ds} - V_{ds} loci that happens to intersect the V_{dsat} loci. For example, in Fig. 2.1, point A corresponds to $V_{ds} = V_{dsat} \cong 0.7V$ for $V_{gs} = 1.2V$.

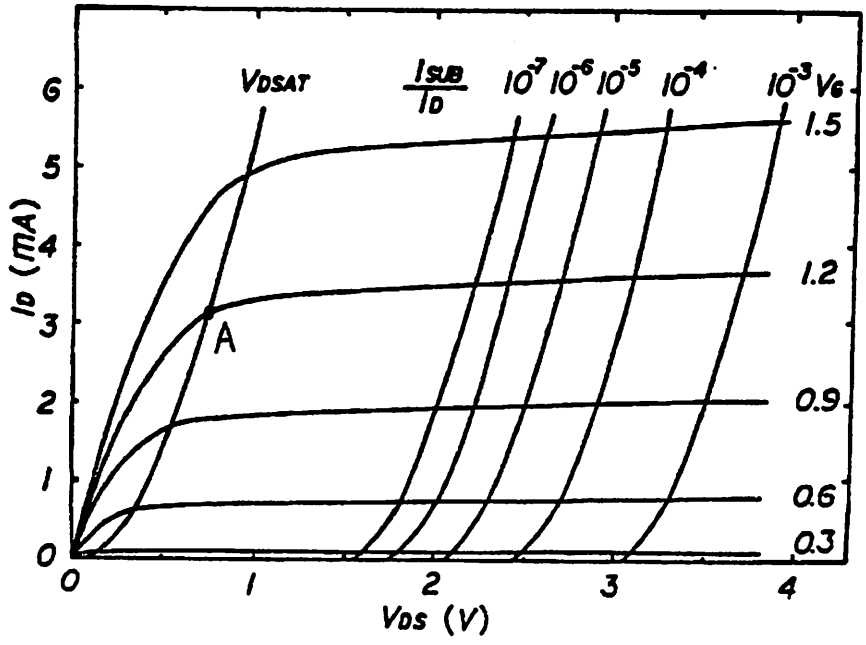


Fig. 2.1 Parallel contours of constant I_{sub}/I_{ds} (from T.Y. Chan et al. [Cha84]).

The actual approach taken in the extraction process is as follows. A specific I_{sub}/I_{ds} current contour is chosen internally by the program. Presently, this pre-set current ratio is set at 0.2 decades below the I_{sub}/I_{ds} value at $V_{ds} = V_{dd}$ and maximum gate bias, or at 10^{-5}, whichever is smaller. This procedure ensures that a current ratio is chosen such that leakage current is not substantial, yet is less than the maximum current ratio measured at $V_{ds} = V_{dd}$ for all gate biases. Next, I_{sub}/I_{ds} is measured at $V_{gs} - V_{th} = 0.3$ with varying V_{ds} (Fig. 2.2). At this low gate bias, V_{dsat} is approximately $V_{gs} - V_{thr}$. The program then notes the V_{ds} value at which the measured I_{sub}/I_{ds} value is equal to the previously set current ratio. An offset voltage $V_{doffset}$ is then found by taking the difference between this drain voltage and the $V_{dsat} \cong V_{gs} - V_{th}$ found at low V_{gs} (Fig. 2.2). Now, because the I_{sub}/I_{ds} current ratio contours are parallel for all

gate biases, V_{dsat} for other gate voltages can be found simply by noting the V_{ds} value at which I_{sub}/I_{ds} for a particular gate voltage is equal to the preset current ratio, and then subtracting $V_{doffset}$ from it. This difference will equal V_{dsat} for that specific gate voltage. Generally, $V_{doffset}$ changes for different substrate biases so that the low gate bias measurement must be done for each substrate bias value.

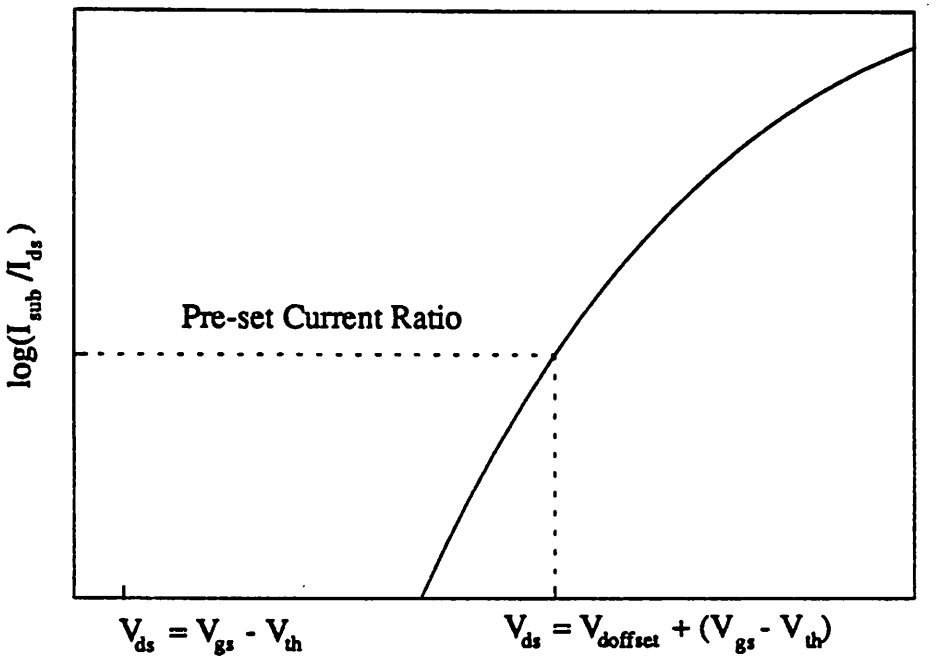


Fig. 2.2 Calculation of $V_{doffset}$. The I_{sub}/I_{ds} ratio is measured at $V_{gs} = V_{th}$ +0.3V. At this low gate bias, $V_{dsat} \cong V_{gs} - V_{th}$.

Once the V_{dsat} values are extracted, they are fitted by linear regression to the analytical model presented in Section 2.2 and repeated here for reference:

$$V_{dsat} = \frac{E_{crit} L (V_{gs} - V_{th})}{E_{crit} L + (V_{gs} - V_{th})} \qquad (2.3.1)$$

After measurements on various wafers, it was found that the V_{dsat} values obtained could accurately be predicted by making the critical electric field parameter E_{crit} dependent on both gate and substrate bias while using the first-order BSIM1 threshold voltage model,

$$V_{th} = VFB + PHIF2 + K1\sqrt{PHIF2 - V_{bs}} - K2\,(PHIF2 - V_{bs}) \qquad (2.3.2)$$

VFB, PHIF2, K1, and K2 are parameters extracted in the normal I_{ds} extraction process [Jen85, Jen87]. The best fit was obtained by using a linear fit in terms of the bias voltages,

$$E_{crit}(V_{bs},V_{gs}) = E_{crit0} + E_{critg}\,V_{gs} + E_{critb}\,V_{bs} \qquad (2.3.3)$$

Figs. 2.3 and 2.4 clearly show the linear E_{crit} dependency to V_{gs} and V_{bs}. The increase in E_{crit} with increasing gate or substrate bias correlates with previously established results [Sod84]. In that work, this increase is attributed to electron mobility degradation caused by vertical fields, which in turn causes an increase of the lateral field necessary to velocity-saturate the channel electrons.

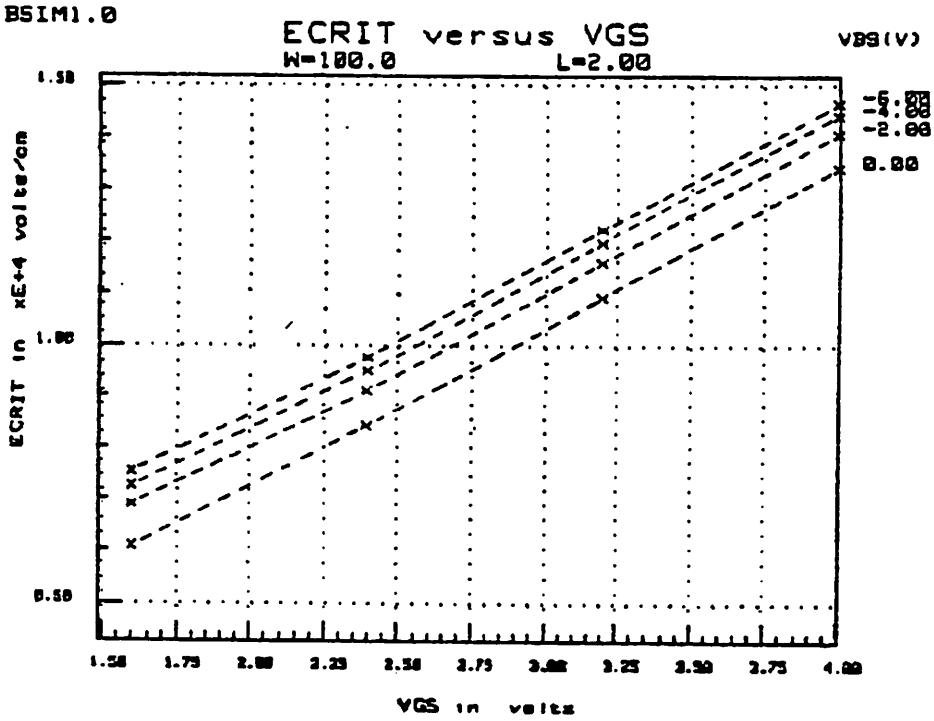


Fig. 2.3 E_{crit} versus V_{gs} (W = 100μm, L = 2μm, non-LDD NMOSFET). Extracted values are represented by x's.

An additional modification was found to be needed to accurately extract the E_{crit} parameters. Because Eq. 2.3.1 becomes inaccurate for low gate bias, it was found that E_{crit} actually decreased before increasing when plotted with increasing gate bias. Thus the program ignores all data that occur before

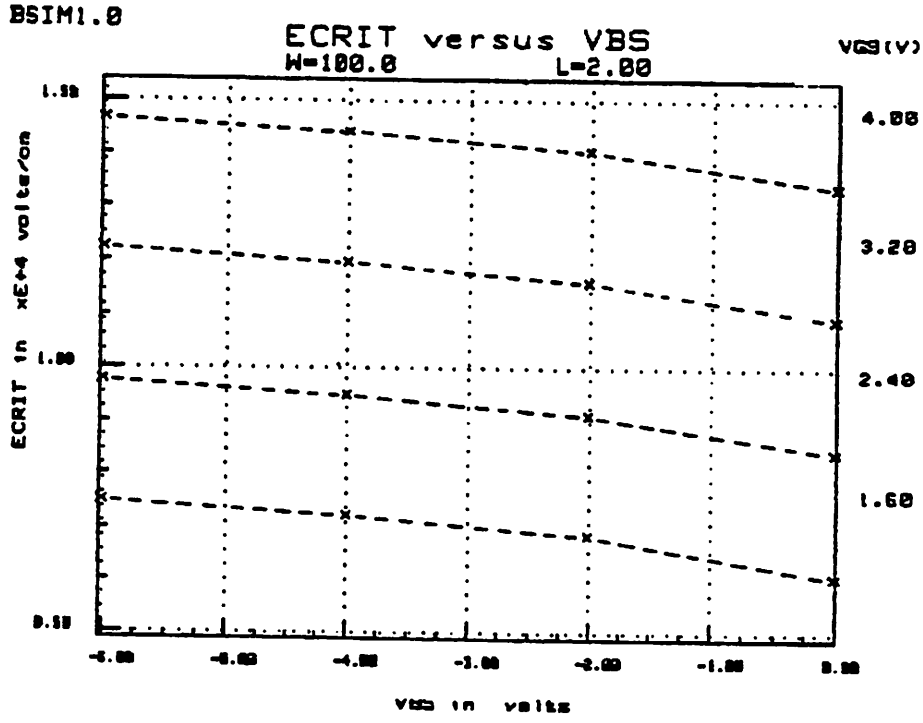


Fig. 2.4 E_{crit} versus V_{bs} (W = 100μm, L = 2μm, non-LDD NMOSFET).

this minimum and uses data points only after it senses a positive slope in E_{crit}.

Fig. 2.5 compares the simulated and measured value of V_{dsat} using this method. As can be seen, the predicted values correlate well with measured behavior.

2.3.3 Extraction of l_c

Once all measured V_{dsat} values are known, l_c can be extracted from measured I_{sub}/I_{ds} values from Eq. 2.2.7,

$$\frac{I_{sub}}{I_{ds}} = \frac{A_i}{B_i}(V_{ds} - V_{dsat})\exp\left[-\frac{B_i l_c}{V_{ds} - V_{dsat}}\right] \qquad (2.3.4)$$

A_i and B_i, constants from the electron impact ionization coefficient α_n, are set at 2×10^6 cm^{-1} and 1.7×10^6 V/cm.

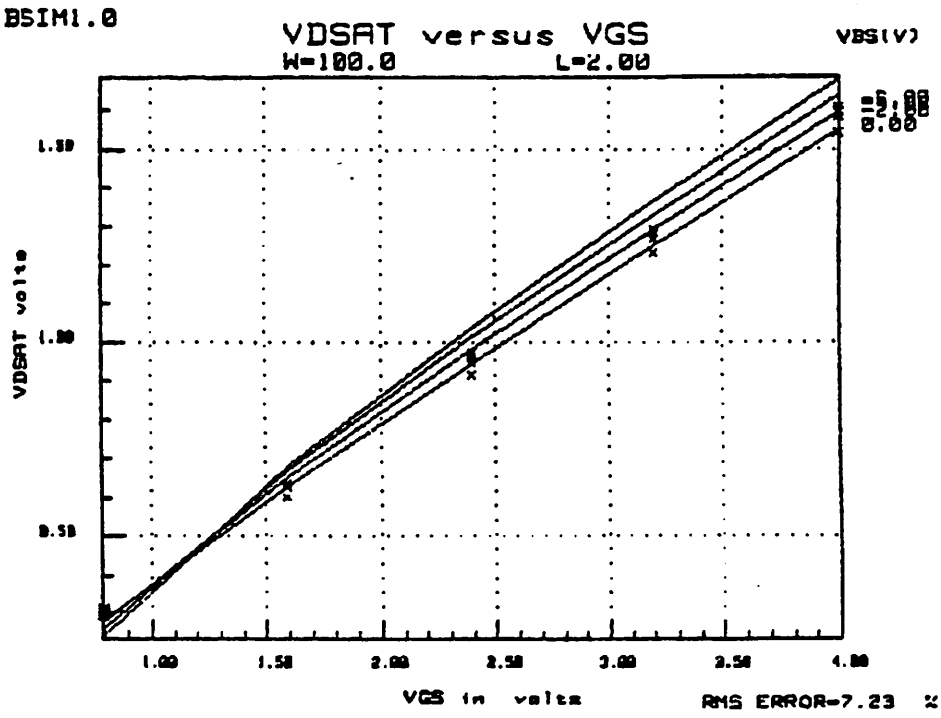


Fig. 2.5 V_{dsat} versus V_{gs}, with V_{bs} as the third parameter (W = 100 µm, L = 200µm, non-LDD device). Data is represented by x's, while simulation is represented by the solid lines.

Extensive measurements were done on NMOS enhancement devices for various processes, with device sizes ranging from 100µm to 4µm in width, and 20µm to 1.5µm in length. All data could not be accurately predicted without making l_c bias dependent. After separating the various bias effects, the best form for l_c was found to be

$$l_c(V_{bs}, V_{gs}, V_{ds}) = \sqrt{t_{ox}}\left[l_1 + l_2\left(\frac{1}{V_{gs}+2}\right)\right] \tag{2.3.5}$$

where

$$l_1 = l_{c0} + l_{c1}\left(\frac{1}{V_{bs}-4}\right) + \left[l_{c2} + l_{c3}\left(\frac{1}{V_{bs}-4}\right)\right]V_{ds} \tag{2.3.6a}$$

$$l_2 = l_{c4} + l_{c5}\left(\frac{1}{V_{bs}-4}\right) + \left[l_{c6} + l_{c7}\left(\frac{1}{V_{bs}-4}\right)\right]V_{ds} \tag{2.3.6b}$$

The "2" in the expression $1/(V_{gs} + 2)$ and the "4" in $1/(V_{bs} - 4)$ were determined so that the expressions would be valid for V_{gs} and $V_{bs} = 0$ while giving a good fit to data measured from a wide variety of wafers. All data taken below a preset drain bias, $V_{dthresh}$, is ignored so that leakage current

effects will not alter the extraction. $V_{dthresh}$ is set at 0.2 volts greater than the drain voltage where I_{sub}/I_{ds} falls to 10^{-8} on the maximum gate bias contour.

The motivation behind using the forms in Eqs. 2.3.5 and 2.3.6 can be seen from Figs. 2.6 - 2.8. These plots show the effect of gate, drain, and substrate bias on extracted values of l_c, l_1, and l_2. The importance of the cross-term parameters l_{c3} and l_{c7} can be realized by looking at Figs. 2.7 and 2.8. Note that the slope of the measured curve is dependent on the substrate bias, V_{bs}. Thus, a simple linear relationship is not possible.

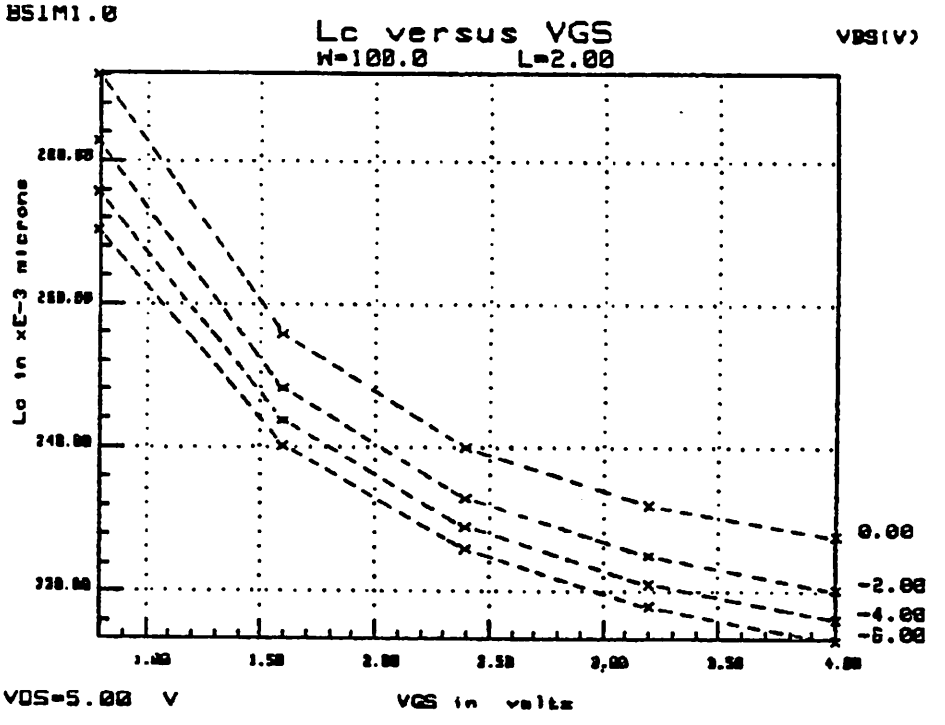


Fig. 2.6 l_c versus V_{gs}, with V_{bs} as the third parameter (W = 100μm, L = 2μm, non-LDD device).

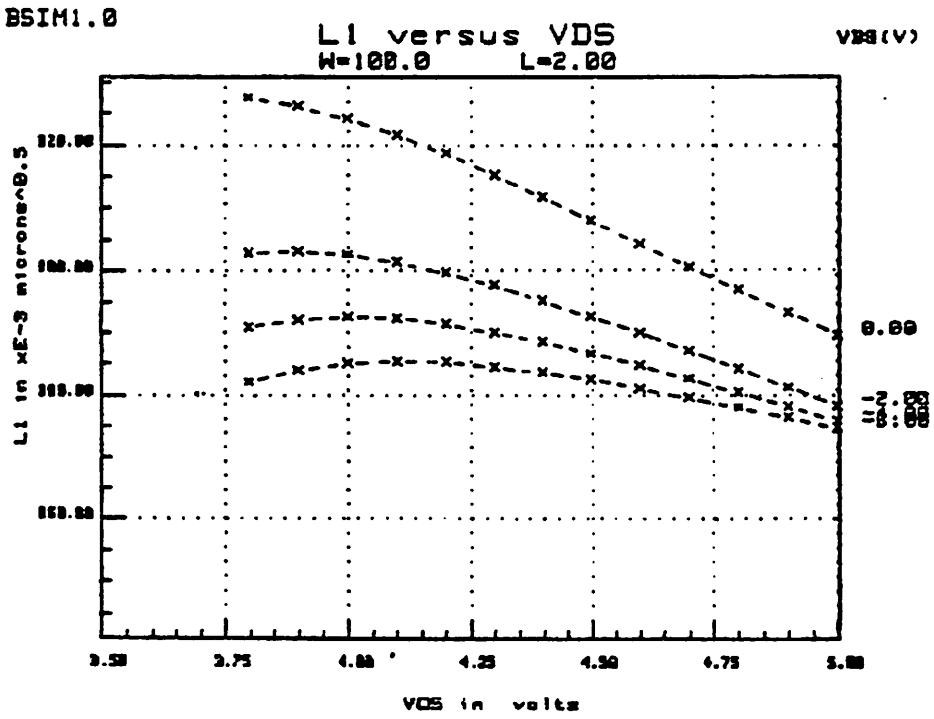


Fig. 2.7 l_1 versus V_{ds}, with V_{bs} as the third parameter (W = 100μm, L = 2μm, non-LDD device).

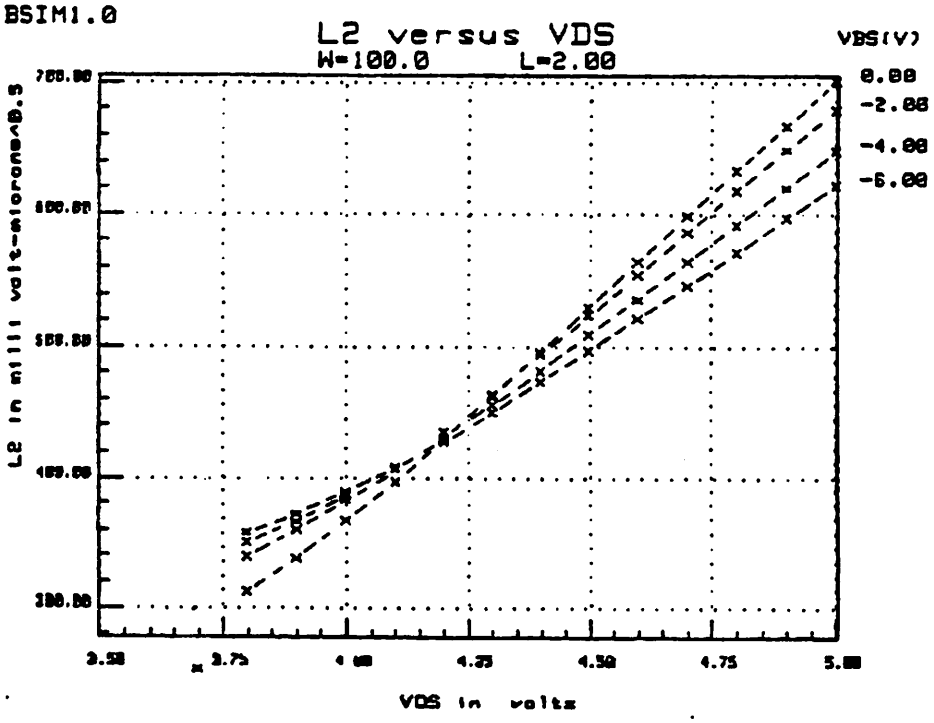


Fig. 2.8 l_2 versus V_{ds}, with V_{bs} as the third parameter (W = 100μm, L = 2μm, non-LDD device).

The inverse V_{gs} dependence of l_c is used to take into account an observable substantial decrease of I_{sub} (and therefore a non-linear increase of l_c) for low V_{gs} - V_{th} (Fig. 2.6). This effect as well as the inverse V_{bs} dependence may be qualitatively explained as being caused by the modulation of the inversion layer thickness of the channel electrons as a secondary effect in addition to the normal decrease in channel charge. As either gate or substrate bias decreases, more electrons flow in a lower field region further away from the silicon-silicon dioxide interface. This results in less electrons experiencing the critical field necessary for impact ionization, and therefore a greater decrease in substrate current than that predicted by theory. This effect is more pronounced for shorter channel lengths and larger drain voltages. At present, no theoretical derivation for this behavior has been published.

2.3.4 Creating Size-Independent Parameters

In parallel with the existing BSIM1 models, each of the 11 parameters are decomposed into three size-independent parameters using the following relation:

$$P_i = P_0 + \frac{P_L}{L_{MK} - \Delta L} + \frac{P_W}{W_{MK} - \Delta W} \qquad (2.3.7)$$

where P_i represents each parameter, and P_0, P_L, and P_W represent its decomposition into size-independent parameters. L_{MK} and W_{MK} are mask length and width, and ΔL and ΔW account for any process bias that may be present. The 33 size-independent parameters, with other statistical information, are then stored in lines 29 to 39 of the process file. These parameters are then used to simulate the I-V characteristics in the graphics playback mode for verification of fit.

2.3.5 Summary

After much analysis, it can be seen that theoretical derivations alone are not sufficient to explain completely and accurately the device behavior observed. It is therefore necessary to empirically

determine an expression that is strongly based upon theory but can also simulate accurately actual device characteristics. In this way, device behavior can be predicted before the theory of that behavior has been firmly established. The accuracy of this parameter extraction process is presented in the following section.

2.4 Experimental Results

Experimental measurements to test the accuracy of the substrate current extraction process were performed on NMOS enhancement devices fabricated by various processes. The results presented here are taken from both LDD and non-LDD devices with an oxide thickness of 285 A, an average substrate doping of 5 x 10^{14} cm^{-3}, and source and drain diffusion junction depths of 0.3 μm, and for the LDD devices, n- region junction depths half as deep as that of the source and drain diffusions. The devices measured have W/L dimensions (in microns) of 8/5, 6/5, 4/5, 4/4, 4/3, 4/2, and 4/1.5 for the non-LDD devices, and 20/2.25, 6/2.25, 4/2.25, 4/3, 4/2, and 4/1.5 for the LDD devices. One size-independent model parameter file per process is used to simulate current characteristics of all devices on the same die.

All measurements were accomplished using a Hewlett Packard 9836 Computer acting as a controller through the HPIB communications bus to a HP4145A Parametric Analyzer, which in turn was connected to a probe station with metal shielding. A detailed description of the measurement setup is given in [Jen85, Jen87]. Two types of plots are available for the substrate current in graphics playback, I_{sub}/I_{ds} versus V_{ds} and I_{sub} versus V_{gs}. Both types are presented in the following figures.

The first several figures present selected data from the non-LDD devices. Figs. 2.9 - 2.19 display each of the 11 substrate parameters for all seven devices tested. Figs. 2.20 and 2.21 show $\log(I_{sub}/I_{ds})$ plotted against V_{ds}, while Figs. 2.22 and 2.23 show I_{sub} versus V_{gs} plots for the 4/1.5 device. Figs. 2.24 through 2.27 display substrate current data and simulated playback of the LDD device in similar fashion.

From the figures, it can be seen that the simulated curves predict quite closely the measured current characteristics. Accuracy can be further improved if a parameter set extracted only from the device of interest is used in the simulated playback. The fact that the simulations are accurate for various device sizes using a single size-independent model parameter set adds confidence to the model.

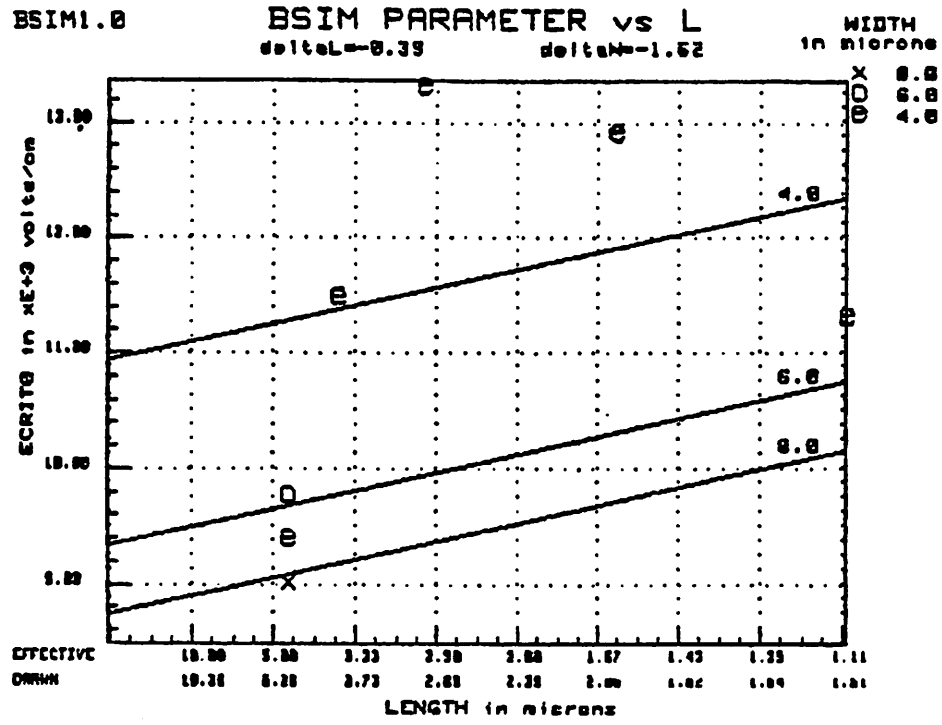


Fig. 2.9 E_{crit0} parameter for the non-LDD devices of various geometries on the same die. The letters represent parameter values extracted from each device, while the solid line represents the least-squares fit used to create the size-independent parameters applicable to all the devices.

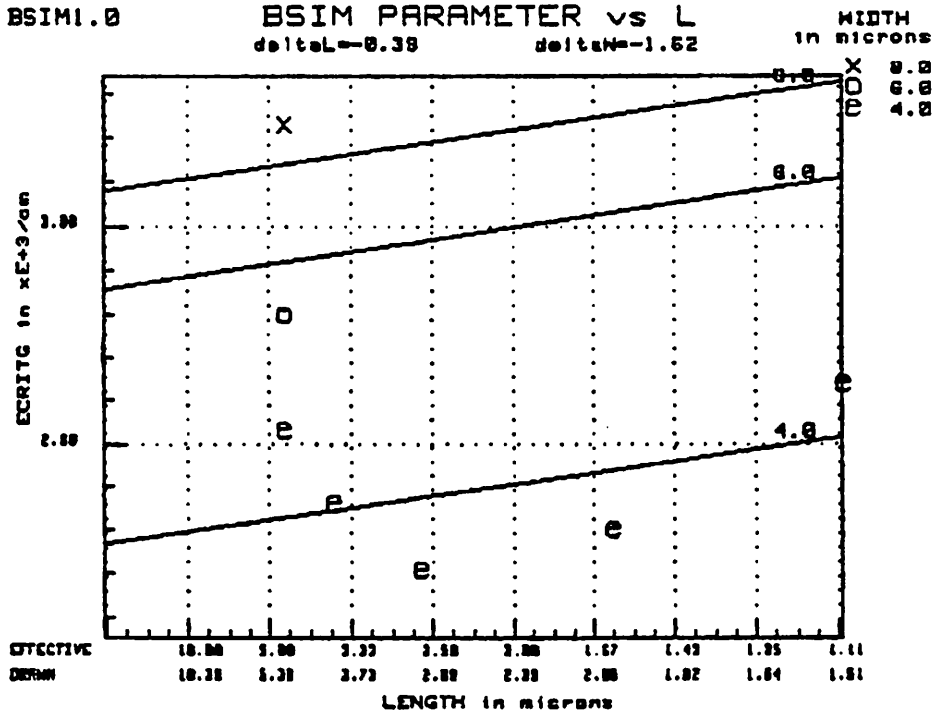


Fig. 2.10 E_{critg} parameter for the non-LDD devices of various geometries on the same die.

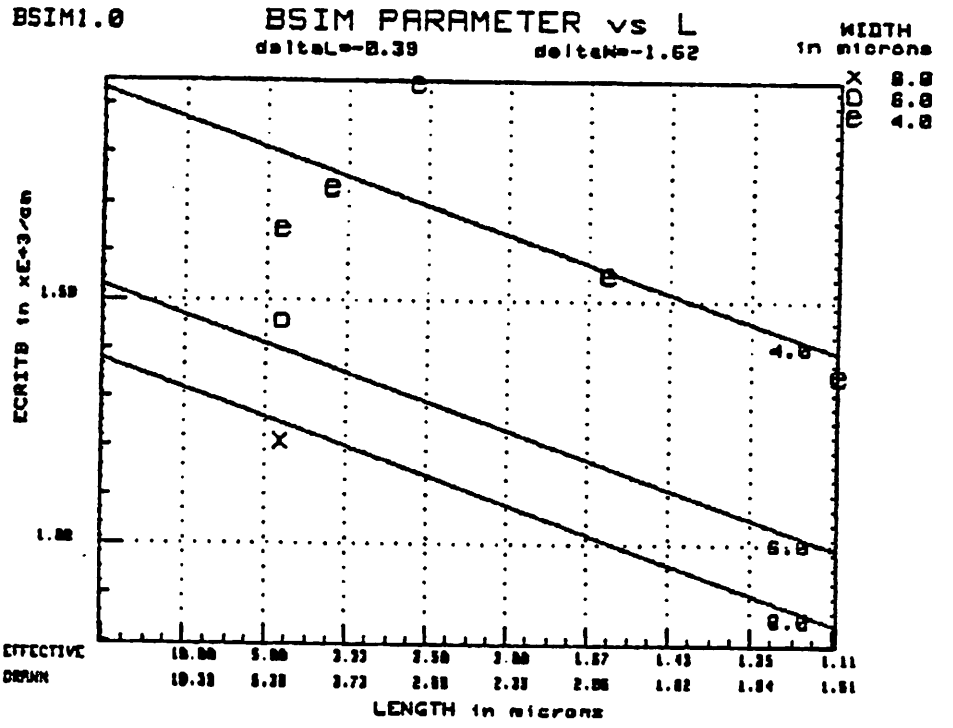


Fig. 2.11 E_{crib} parameter for the non-LDD devices of various geometries on the same die.

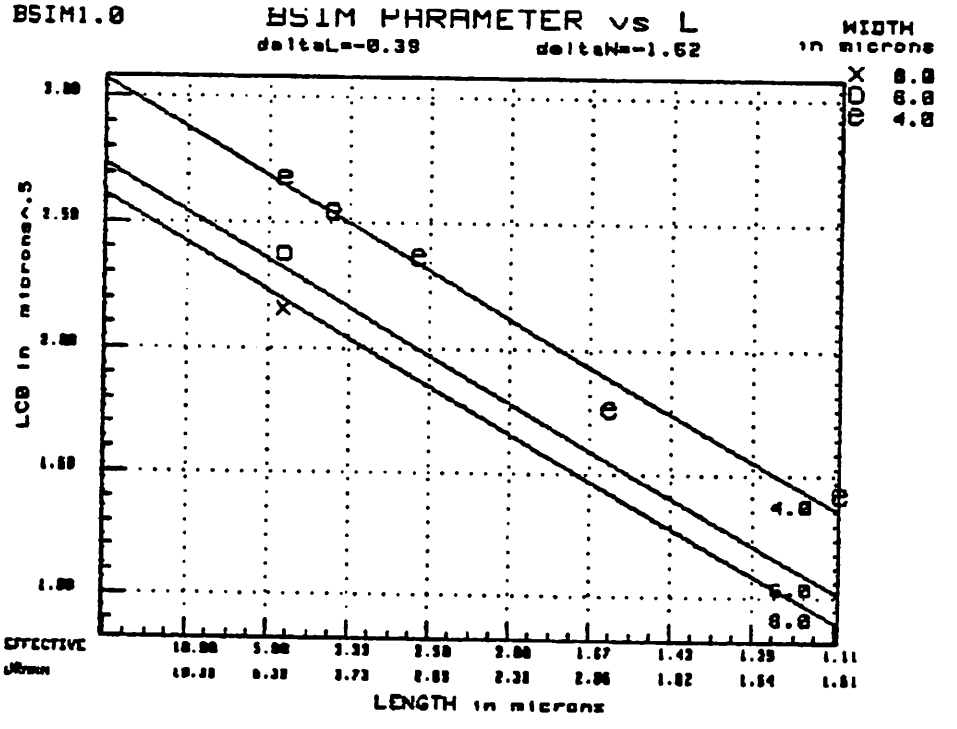


Fig. 2.12 l_{c0} parameter for the non-LDD devices of various geometries on the same die.

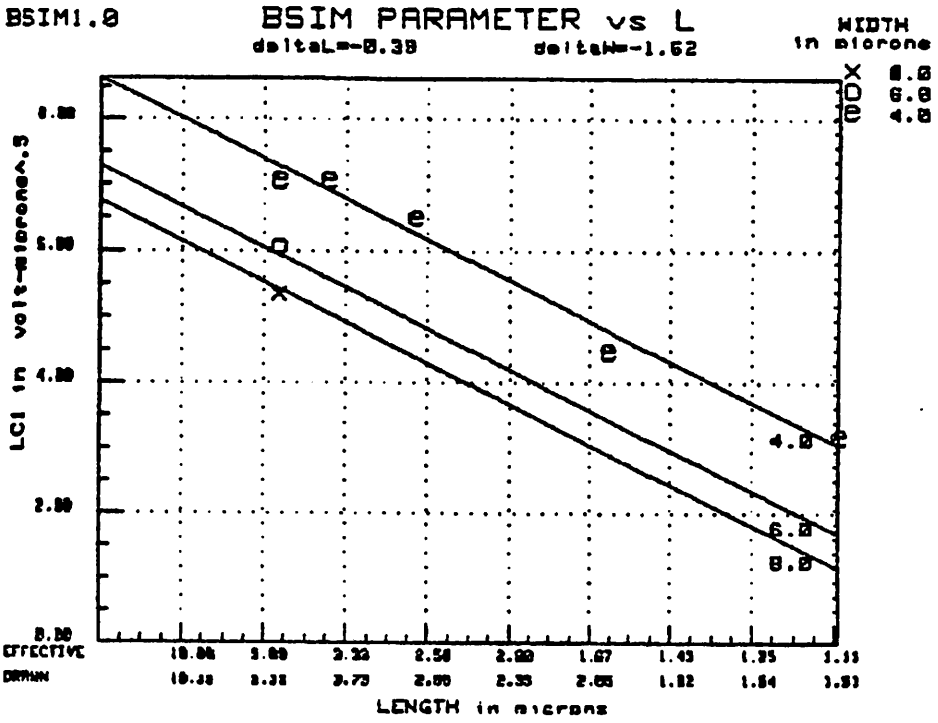


Fig. 2.13 l_{c1} parameter for the non-LDD devices of various geometries on the same die.

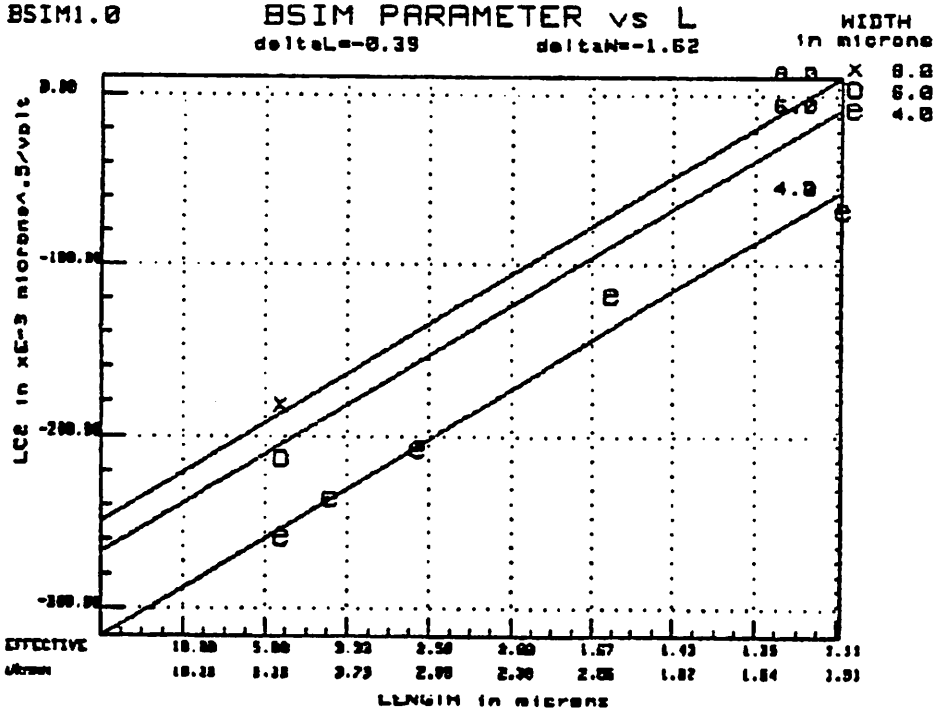


Fig. 2.14 l_{c2} parameter for the non-LDD devices of various geometries on the same die.

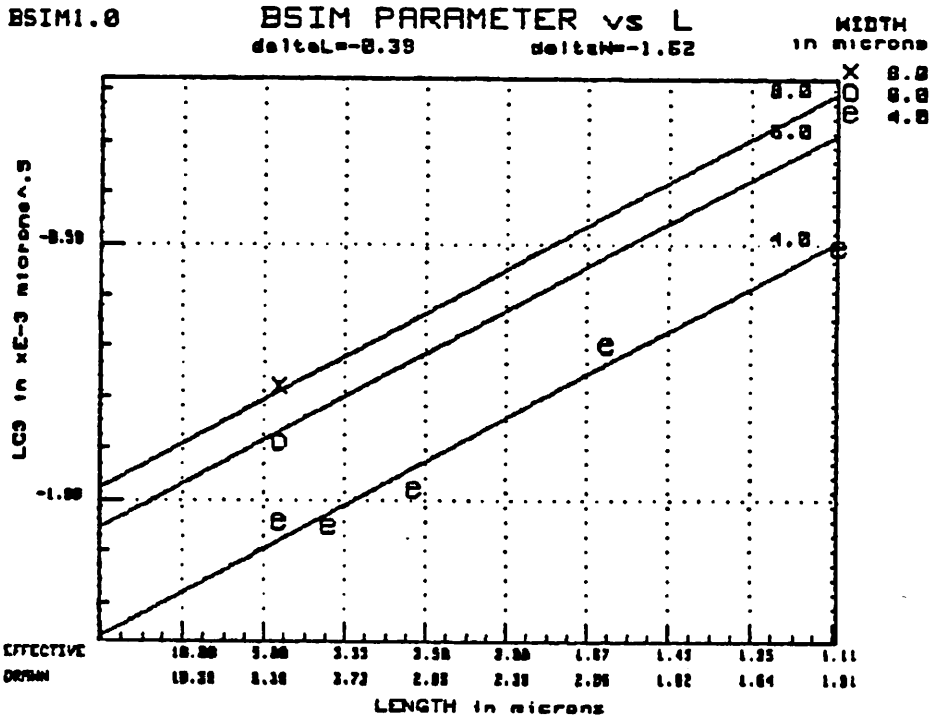


Fig. 2.15 l_{c3} parameter for the non-LDD devices of various geometries on the same die.

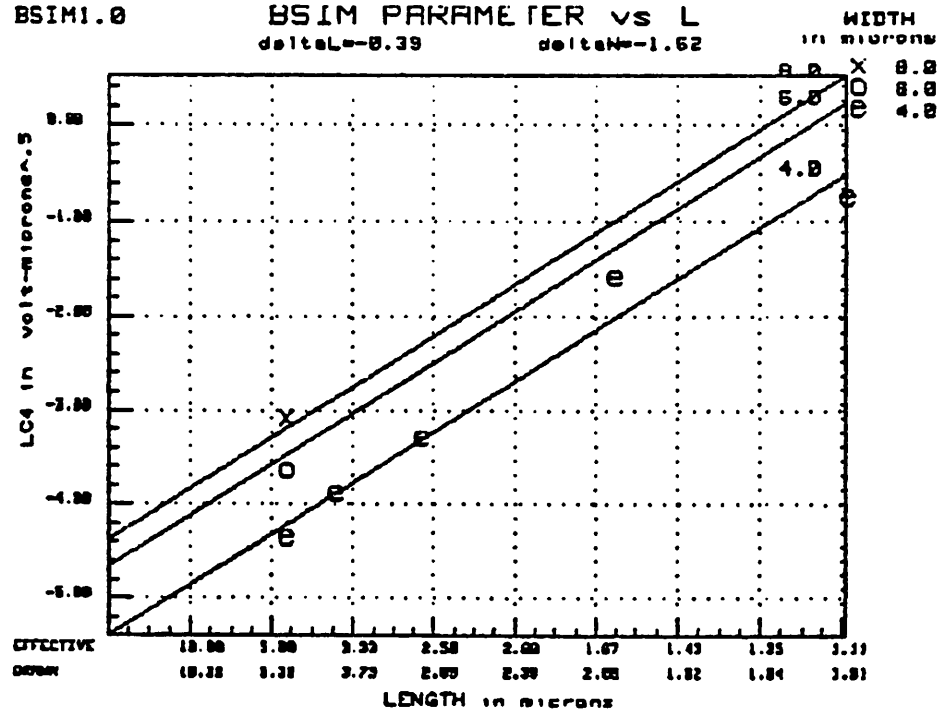


Fig. 2.16 l_{c4} parameter for the non-LDD devices of various geometries on the same die.

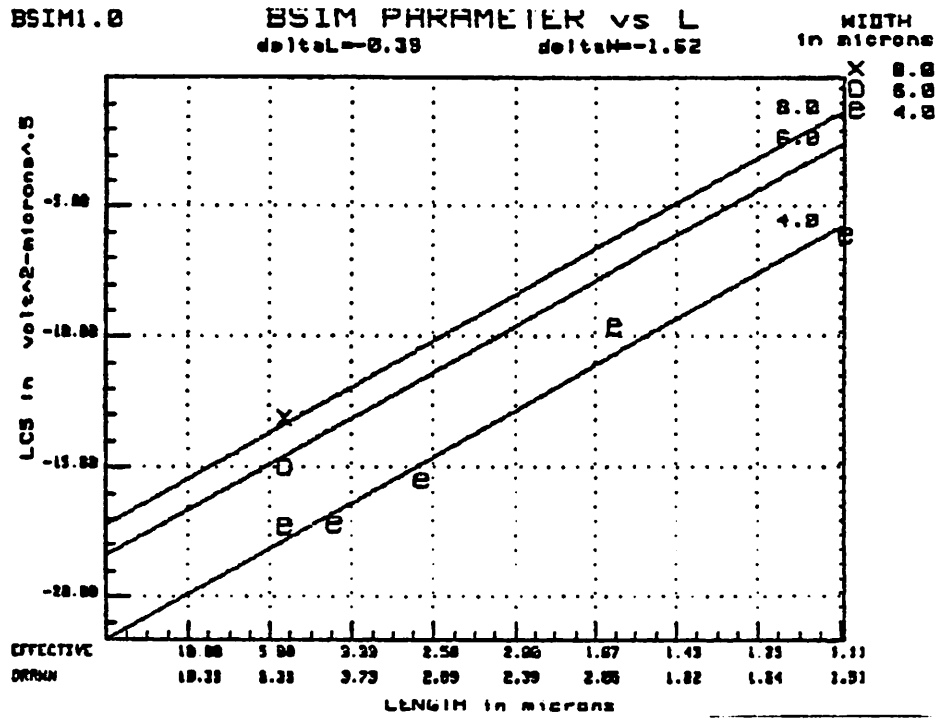


Fig. 2.17 l_{c5} parameter for the non-LDD devices of various geometries on the same die.

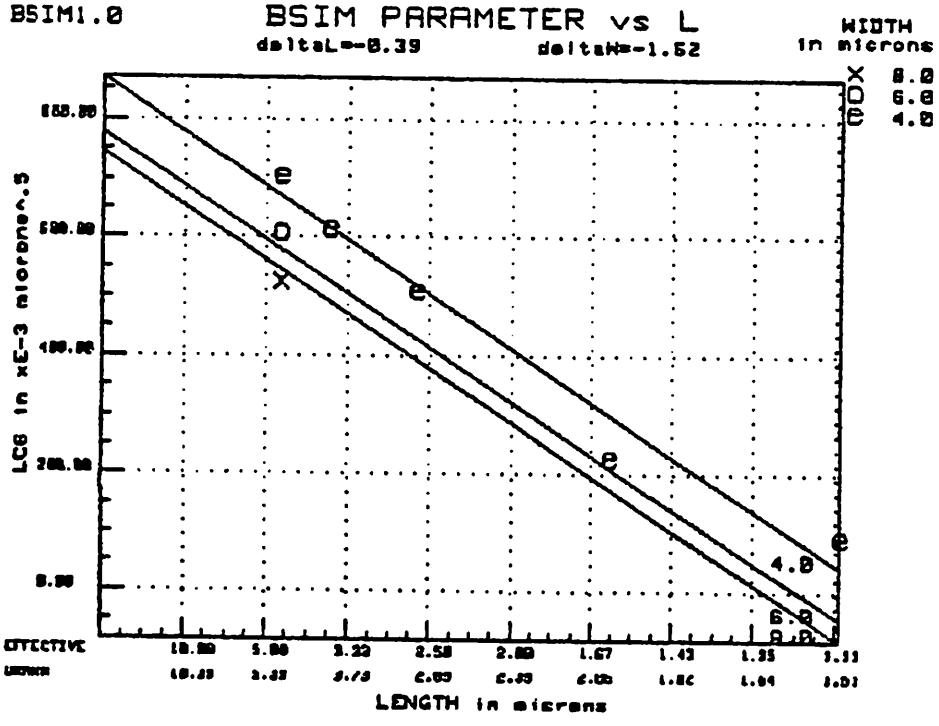


Fig. 2.18 l_{c6} parameter for the non-LDD devices of various geometries on the same die.

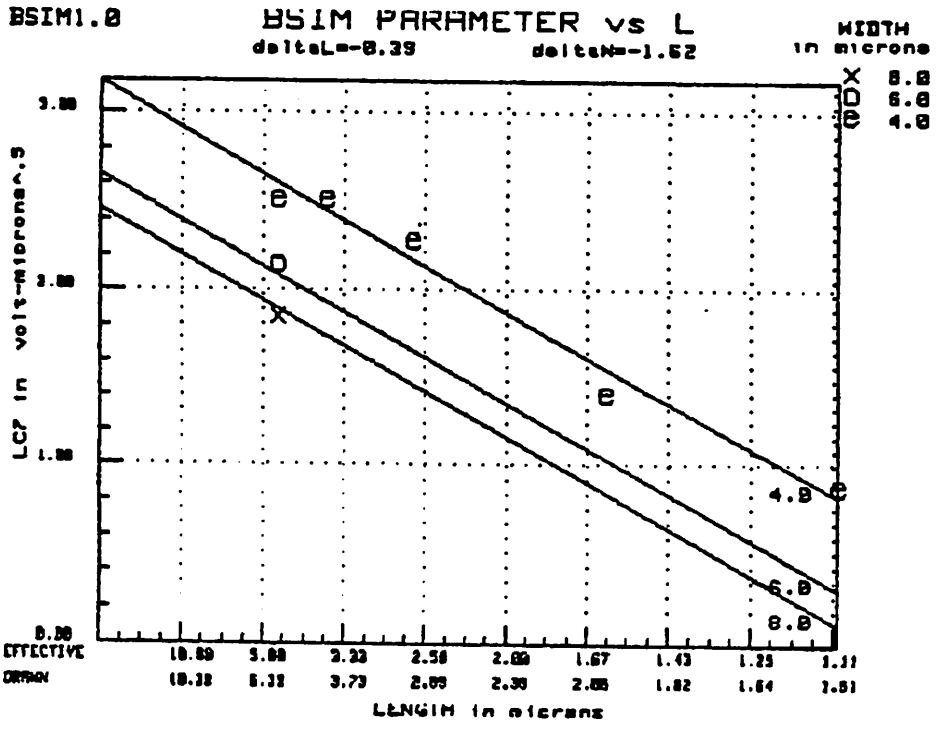


Fig. 2.19 l_{c7} parameter for the non-LDD devices of various geometries on the same die.

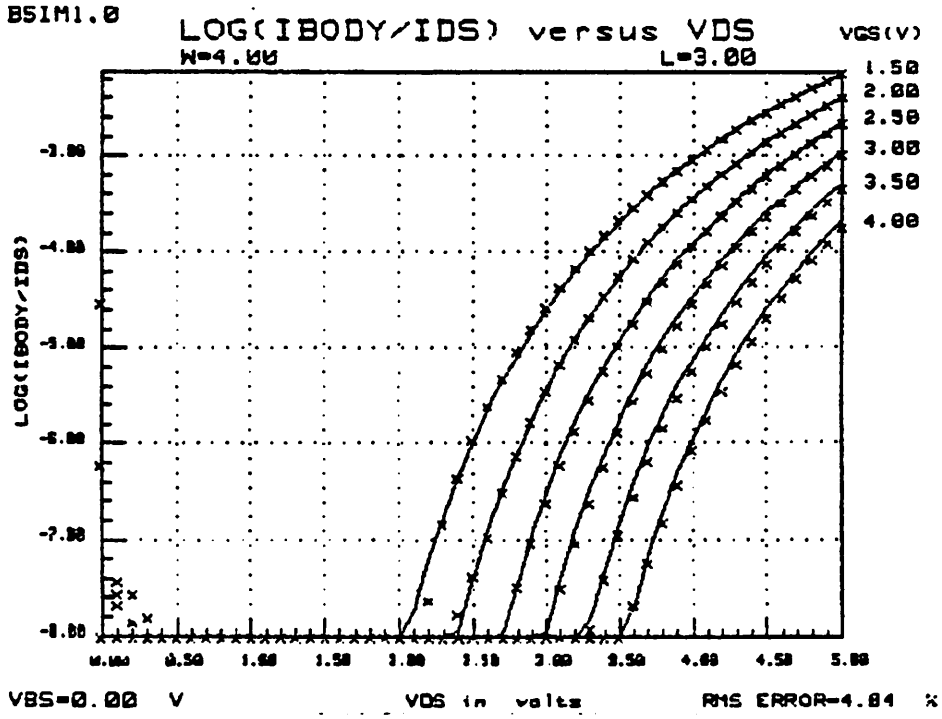


Fig. 2.20 $Log(I_{sub}/I_{ds})$ versus V_{ds} for the non-LDD device. V_{bs} = 0, W = 4μm, L = 3μm. x's represent measured data, while the solid lines represent simulation results from the size-independent parameters.

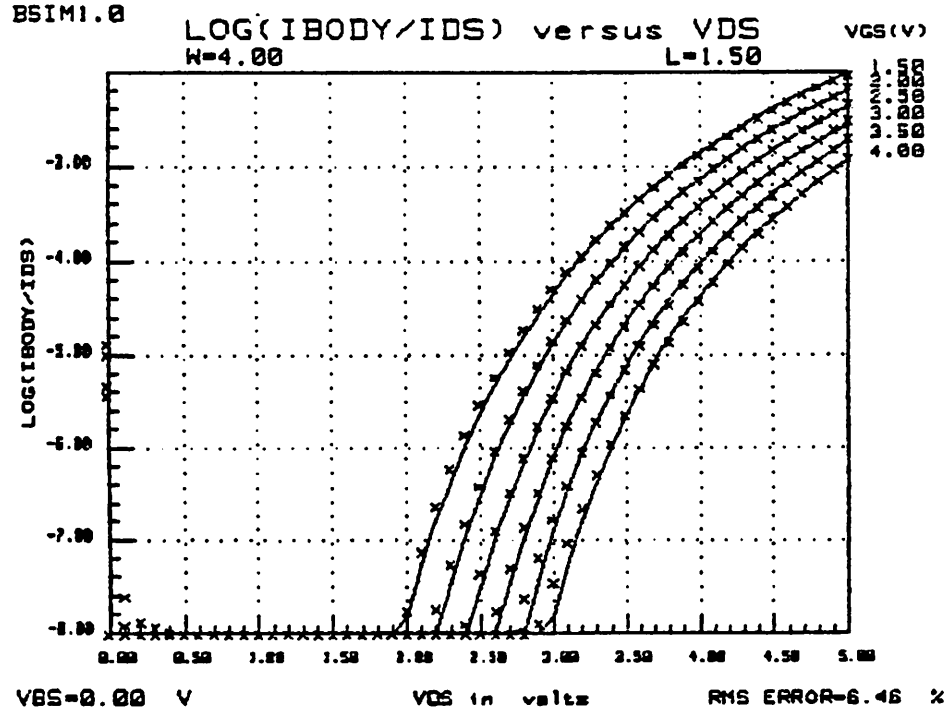


Fig. 2.21 $Log(I_{sub}/I_{ds})$ versus V_{ds} for the non-LDD device. V_{bs} = 0, W = 4μm, L = 1.5μm.

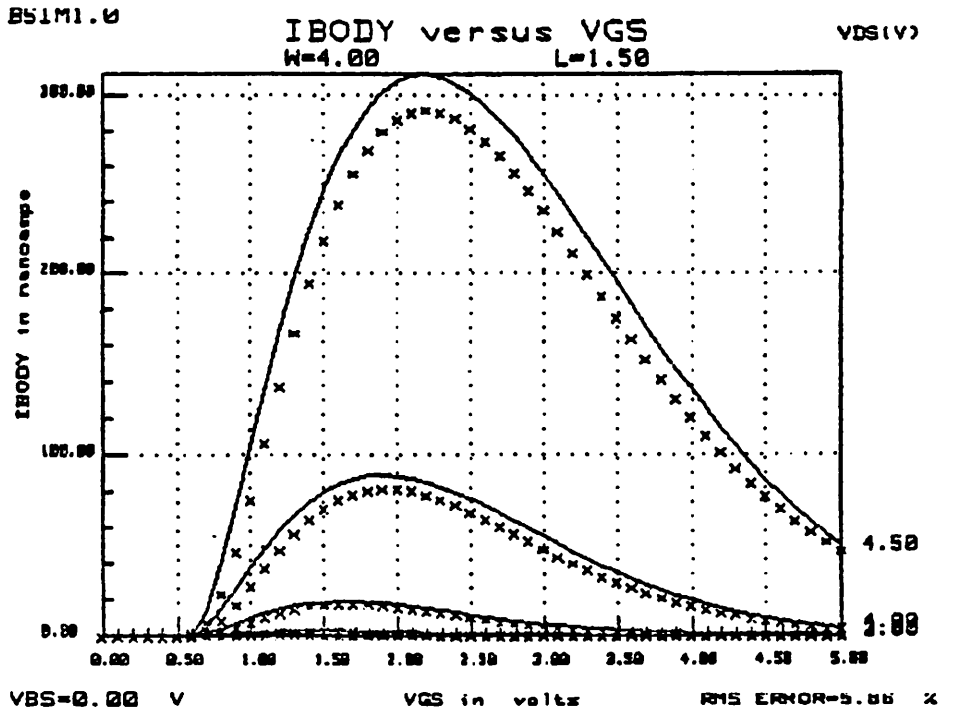


Fig. 2.22 I_{sub} versus V_{gs} for the non-LDD device. V_{bs} = 0, W = 4μm, L = 1.5μm.

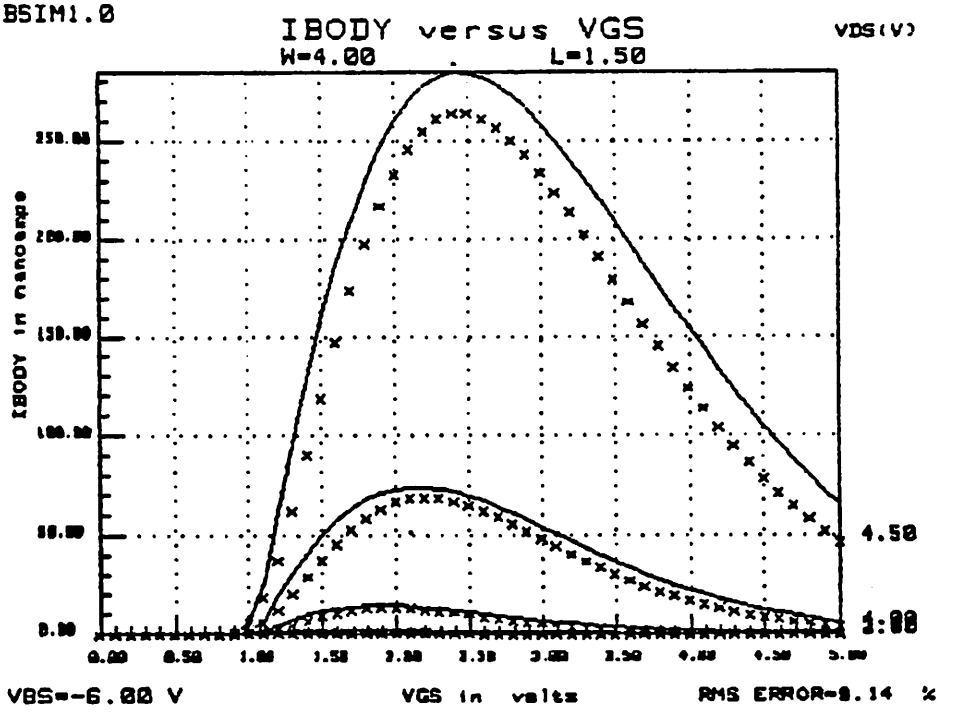


Fig. 2.23 I_{sub} versus V_{gs} for the non-LDD device. V_{bs} = -6, W = 4μm, L = 1.5μm.

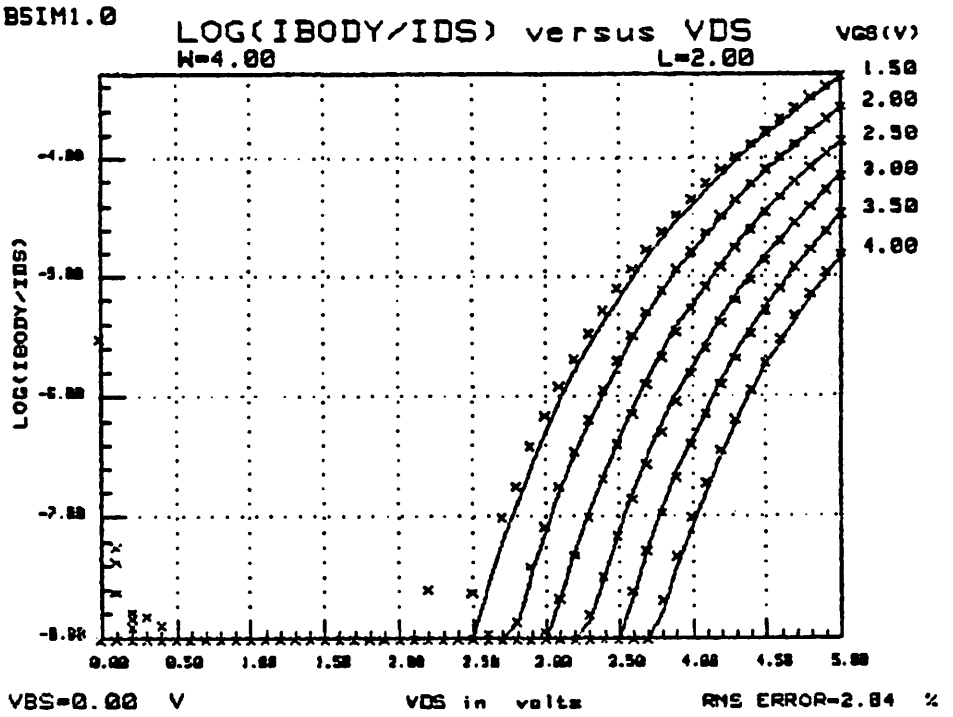


Fig. 2.24 $Log(I_{sub}/I_{ds})$ versus V_{ds} for the LDD device. $V_{bs} = 0$, W = 4μm, L = 2μm.

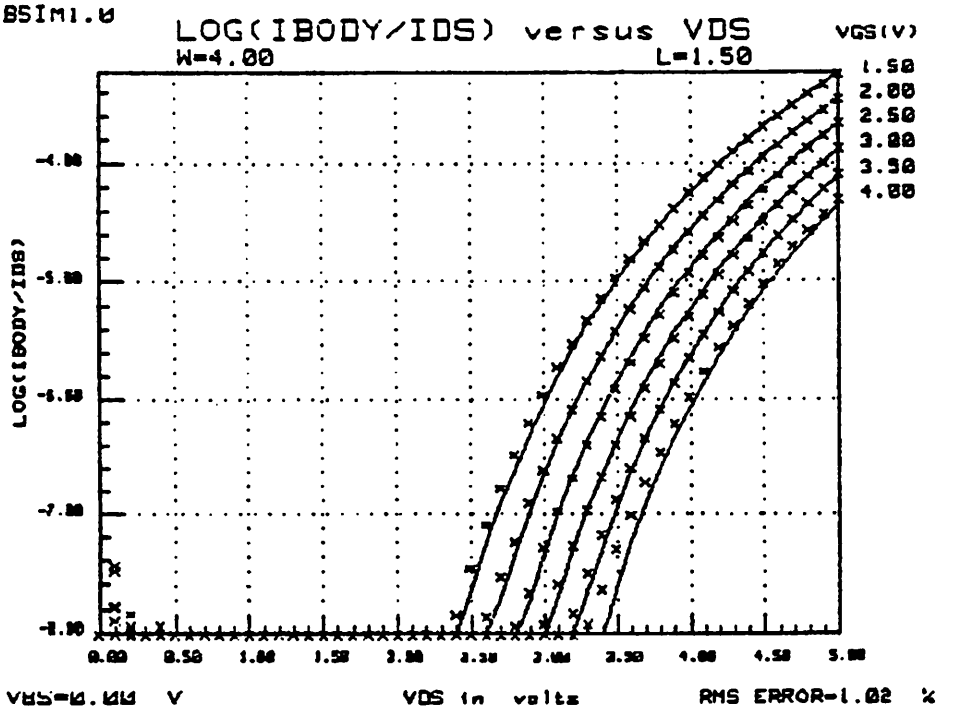


Fig. 2.25 $Log(I_{sub}/I_{ds})$ versus V_{ds} for the LDD device. $V_{bs} = 0$, W = 4μm, L = 1.5μm.

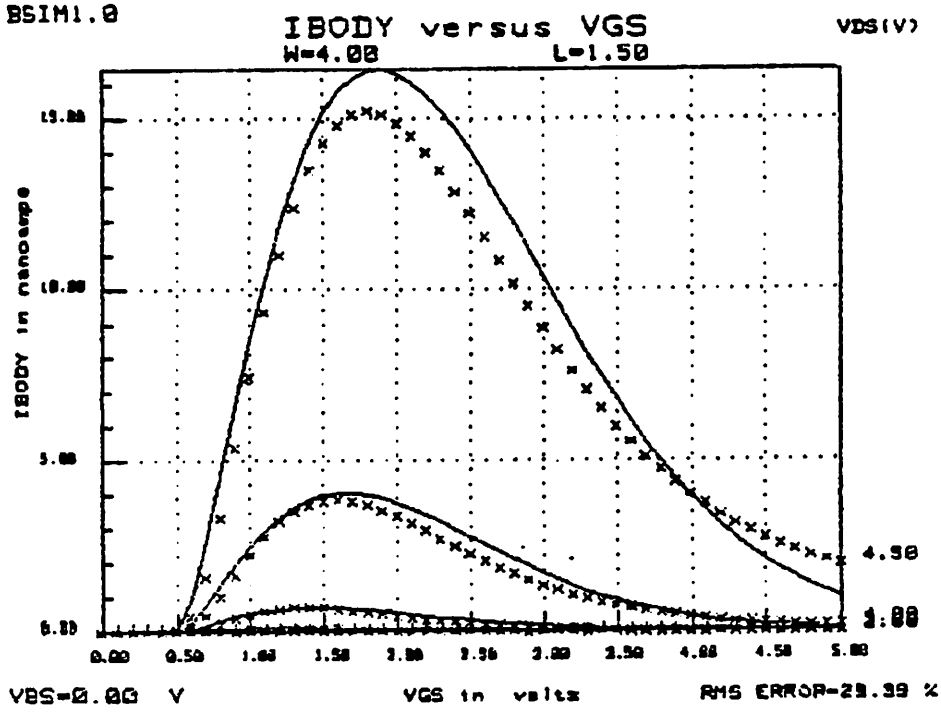


Fig. 2.26 I_{sub} versus V_{gs} for the LDD device. $V_{bs} = 0$, W = 4μm, L = 1.5μm.

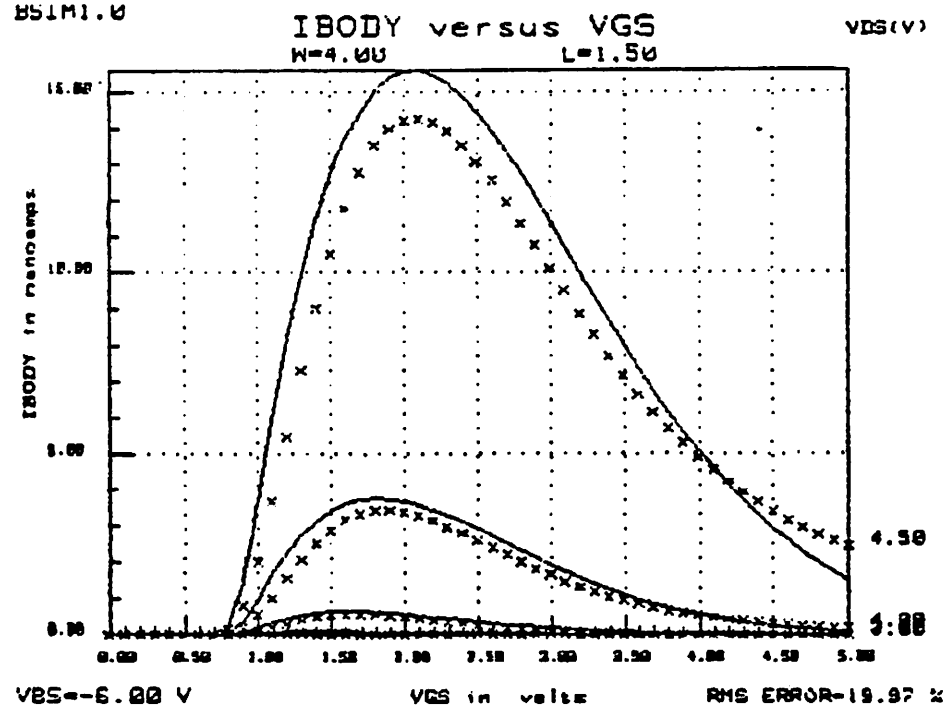


Fig. 2.27 I_{sub} versus V_{gs} for the LDD device. $V_{bs} = -6$, W = 4μm, L = 1.5μm.

2.5 Conclusion

A parametric model of the substrate current has been incorporated into the BSIM1 model. 11 additional parameters are introduced to model the various bias dependencies that exist, three to model V_{dsat}, and eight to model I_c. Each of the 11 parameters can then be decomposed into three size-independent parameters in an identical manner with the other BSIM1 parameters. The now 33 additional parameters are then simply appended to the process file using the same format as before [Jen85, Jen87].

Two additional types of plots have been added to display the substrate current. An I_{sub}/I_{ds} versus V_{ds} plot is available as well as an I_{sub} versus V_{gs} graph showing distinctly the current peaking effect.

It has been shown experimentally that this substrate current model using one model parameter file agrees well with actual data taken from devices of various dimensions and different technologies. Chapter 3 will use this model to predict MOSFET device degradation in a circuit environment.

2.6 References

[Man75] Y.A. El-Mansy and D.M. Caughey, "Modeling Weak Avalanche Multiplication Currents in IGFETS and SOS Transistors for CAD," *IEDM Technical Digest*, pp.31-34, December 1975.

[Man77] Y.A. El-Mansy and A.R. Boothroyed, "A Simple Two-Dimensional Model for IGFET Operation in the Saturation Region," *IEEE Trans. Electron. Devices*, Vol. ED-24, pp.254-262, March 1977.

[Ko81] P.K. Ko, "Hot-Electron Effects in MOSFETS," University of California, Berkeley, Ph.D. Thesis,1981.

[Cha84] T.Y. Chan, P.K. Ko, and C. Hu, "A Simple Method to Characterize Substrate Current in MOSFETs," *IEEE Trans. Electron. Device Letters*, Vol. EDL-5, p.505, December 1984.

[Sod84] C. Sodini and P.K. Ko, "The Effect of High Fields on MOS Device and Circuit Performance," *IEEE Trans. Electron. Devices*, Vol. ED-31, pp.1386-1393, October 1984.

[Hu85] C. Hu, S. Tam, F-C Hsu, P.K. Ko, T.Y. Chan, K.W. Terrill, "Hot-Electron-Induced MOSFET Degradation - Model, Monitor, Improvement," *IEEE Trans. Electron. Devices*, Vol. ED-32, pp. 375-385, February 1985.

[Jen85] M.-C. Jeng, B.J. Sheu, P.K. Ko, "BSIM Parameter Extraction - Algorithms and User's Guide," University of California, Berkeley, ERL Memo, UCB/ERL M85/79, October 7, 1985.

[Tsu85] T. Tsuchiya and J. Frey, "Relationship between hot-electrons/holes and degradation of p- and n-channel MOSFET's," *IEEE Electron Device Letters*, Vol. EDL-6, No. 1, pp. 8-11, January 1985.

[Hir86] Y. Hiruta, K. Maeguchi, and K. Kanzaki, "Impact of hot electron trapping on half micron PMOSFETs with p+ poly Si gate," *IEDM Tech. Digest*, pp. 718-721, December 1986.

[Tzo86] Joseph J. Tzou, C.C. Yao, R. Cheung, and Hugo W.K. Chan, "Hot-carrier-induced degradation in p-channel LDD MOSFET's," *IEEE Electron Device Letters*, Vol. EDL-7, No. 1, pp. 5-7, January 1986.

[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.

[Bra88] Michael P. Brassington, Mark W. Poulter, and Monir El-Diwany, "Suppression of hot-carrier effects in submicrometer surface-channel PMOSFET's," *IEEE Trans. Electron Devices*, Vol. 35, No. 7, pp. 1149-1151, July 1988.

[Ong88] T.-C. Ong, Kouichi Seki, P.K. Ko, and Chenming Hu, "Hot-carrier-induced degradation in p-MOSFET's under AC stress," *IEEE Electron Device Letters*, Vol. 9, No. 5, pp. 211-213, May 1988.

[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.

Appendix 2A: BSIM Substrate Current Users Guide

This brief outline describing the operation of the substrate current extraction portion of BSIM assumes that the reader is already familiar with the general operation of the program. All inputs required from the user are directly analogous to the normal extraction process. For more information, refer to [Jen85, Jen87].

After the normal and (if chosen) subthreshold extraction is completed, the program will prompt whether substrate current extraction is desired (Fig. 2A1). Entering "n" or "N" will terminate the extraction process and the program will continue normally as before through the graphics routines. If "y" or "Y" is entered, then the substrate current extraction routine is entered.

```
                    ***BSIM EXTRACTION STATUS***

PROCESS=                                VDD=5.00 VOLTS
LOT=                                    TEMP=27.00 DEG C
WAFER=                                  TOX=285.00 ANGSTROMS
DATE=                                   XPOS= 1 YPOS= 1
OPERATOR=                               DEVICE=NCHANNEL
OUTPUT FILE=bsimout.TEXT                WIDTH=8.00 MICRONS
PROBER FILE=amdprfile.TEXT              LENGTH=5.00 MICRONS

MINUTES TO DIE COMPLETION=15.1          MINUTES TO WAFER COMPLETION=15.1
DEVICE EXTRACTION LOCATION XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX FINISHED
PRESENT DEVICE BSIM PARAMETERS
VFB=-0.421                              X2U0=0.002207
PHIF2=0.714                             X2U1=-0.000961
K1=0.567                                X3U1=0.002651
K2=0.092                                X2BETA0=0.000001
ETA=-0.001                              X2ETA=0.002958
BETA0=0.000089                          X3ETA=-0.000630
U0=0.053                                BETA0SAT=0.000116
U1=0.074                                X2BETA0SAT=0.000002
N0=                                     X3BETA0SAT=0.000002
ND=                                     NB=
Are you interested in substrate current measurements and extractions?(Y/N) >
```

Fig. 2A1 Prompt (at bottom of screen) to enter substrate current parameter extraction portion of BSIM1.

The display that is now written onto the screen is similar to the previous status display except that the substrate current parameters appear on the screen (Fig. 2A2). Extraction will proceed after the user

has entered minimum and maximum substrate bias values. The program, as before, will indicate any errors that occur and the time remaining until completion. After all measurements are completed and if the user had not chosen the single device mode, the program will prompt whether the measurements are satisfactory to continue the extraction. Once "y" or "Y" is entered, the substrate current parameters are calculated and displayed on the screen. The program then continues by measuring the next device, or by developing the model parameter file and entering either the parameter graphics routines or the current playback routines according to the user's choice.

Substrate current parameter graphics and current playback are viewed in analogous fashion with the other parameters and currents. Thus the operation of the rest of the program is identical to that described in [Jen85].

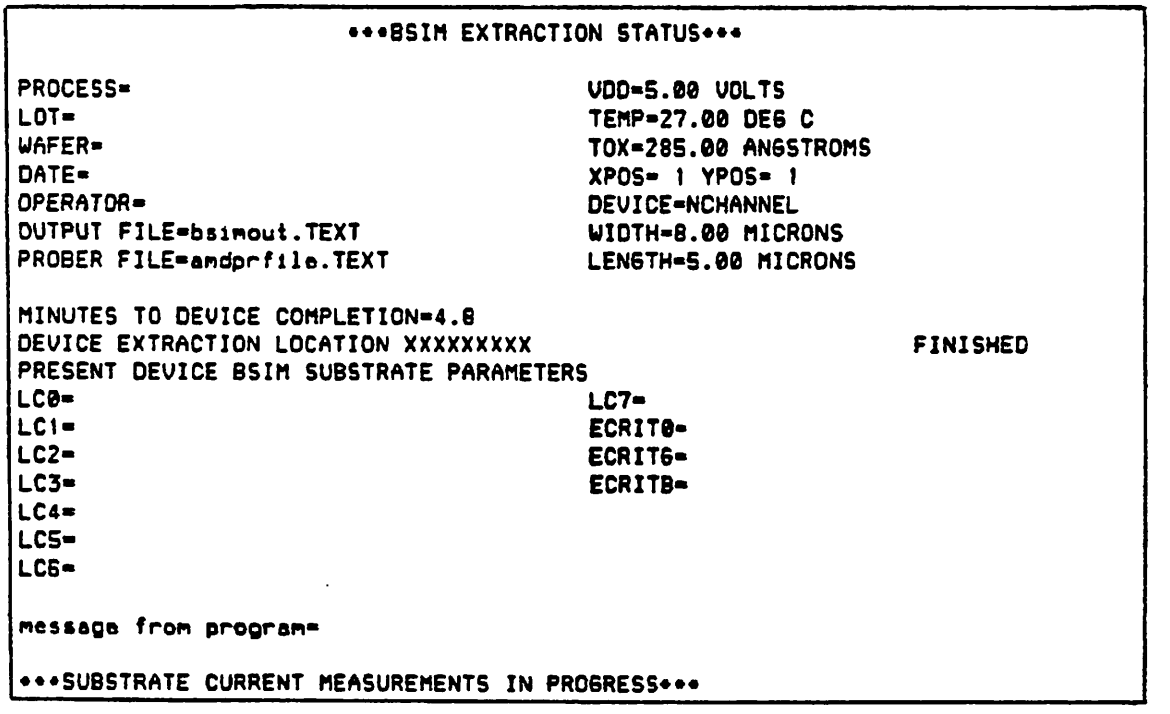


Fig. 2A2 Substrate current measurement status display.

Appendix 2B: Modifications To The BSIM Source Code

2B1: Procedures Added

1. Procedure substrate_status_display

This procedure displays the status of the substrate current measurements on the screen.

2. Procedure string_setup_measure_IBODYvsVDS

Similar to procedure string_setup_measure_IDSvsVGS, this procedure sets up the strings necessary to measure substrate current.

3. Procedure source_setup_measure_IBODYvsVDS

Similar to procedure source_setup_measure_IDSvsVGS, this procedure sends the strings necessary to properly set up the 4145A.

4. Procedure measure_device_functionality_IBODYvsVDS

Similar to procedures source_setup_nchannel_device_functionality and measure_device_functionality combined, this procedure determines whether the substrate is open- or short- circuited to the source and drain.

5. Procedure chan_definition_device_functionality_IBODYvsVDS

Similar to procedure chan_definition_device_functionality, this procedure sets up the channel definition page of the 4145A for the functionality test.

6. Procedure measure_and_reduce_IBODYvsVDS

This procedure contains two procedures within itself to measure substrate and drain current for the extraction of l_c and E_{crit}.

7. Procedure measure_IBODYvsVDS

The actual measurement of the substrate and drain curves are done.

8. Procedure reduce_IBODYvsVDS

This procedure contains two others to reduce data to determine E_{crit} and l_c. The parameter extraction for l_c is also done.

9. Procedure measure_and_reduce_vdsat

Data to determine V_{dsat} parameters is reduced to values of E_c.

10. Procedure reduce_and_extract_parameter_l

Data to determine l_c is reduced, and the bias-dependent parameters for l_c are extracted.

11. Procedure substrate_parameter_extraction

Extraction of the remaining 3 E_{crit} parameters is done, and the 11 parameters are printed on the screen.

12. Function bsimsim_body

Similar to function bsimsim, this function plots the I_{sub}/I_{ds} graphs.

13. Procedure IBODYvsVDS_data

Measurement for I-V graphics playback for I_{sub}/I_{ds} versus V_{ds} is performed.

14. Procedure channel_definition_for_IBODYvsVDS_data

Preliminary setup of the 4145A is done for data playback.

15. Procedure string_setup_measure_IBODYvsVDS

Bias voltage setups are written into strings.

16. Procedure source_setup_measure_IBODYvsVDS

Bias voltage setup strings are sent to the 4145A.

17. Procedure measure_IBODYvsVDS_data

The actual data measurement is done.

18. Procedure IBODYvsVGS_data

This procedure and its related subprocedures are analogous to procedure IBODYvsVDS_data and its related subprocedures (numbered 13 - 17 above) except that data measurement for I_{sub} versus V_{gs} graphics playback is done.

19. Procedure IBODYvsVD

This procedure generates the measured and simulated values for plotting I_{sub}/I_{ds} versus V_{ds} in logarithmic scale.

20. Procedure IBODYvsVG

This procedure generates the measured and simulated values for plotting I_{sub} versus V_{gs} in linear scale.

21. Procedure bsim_timer_body

Similar to procedure bsim_timer, this procedure keeps track of extraction time during substrate current measurement.

22. Procedure clear_yb

This procedure clears the second y-axis plot from the graphics page so that data buffer overflow does not occur on the HP4145A for measurements that are done after the I_{sub}/I_{ds} measurements.

23. Procedure select_substrate_bias

Bias voltages for substrate current IV playback are selected by the user in this procedure.

24. Procedure error_calculations_body

This procedure is identical to procedure error_calculations except valid current levels for error calculations are set according to minimum and maximum currents graphed in substrate current analysis.

2B2: Procedures Modified

1. Procedure measure_device_data

Substrate current measurement routine is added.

2. Procedure measure_device

Substrate current error checking is added.

3. Procedure channel_definition_for_IDSvsVGS_data

This procedure now also sets up the 4145A for substrate current measurements.

4. Procedure draw_menu

The options to plot I_{sub} versus V_{gs} and log(I_{sub}/I_{ds}) versus V_{ds} are added.

5. Procedure graphics

The substrate current plots are added.

6. Procedure make_xyz_axis_labels

Labels for the 11 substrate parameters added.

7. Procedure page3

Choices for the 11 substrate parameters are added.

8. Procedure store_parameters_in_die_files

The substrate parameters are added to the process file.

9. Procedure load_up_process_parameters

Array sizes increased to accommodate 11 more substrate parameters.

10. Procedure bsim1.0 (MAIN BSIM PROGRAM)

Substrate current measurement enhancements added.

CHAPTER 3: BERKELEY RELIABILITY TOOLS - CIRCUIT AGING SIMULATOR

3.1 Introduction

This chapter introduces the Circuit Aging Simulator (CAS) [Lee88], part of the BErkeley Reliability Tools (BERT). CAS predicts hot-carrier degradation of MOS devices undergoing dynamic operation in circuits and can therefore predict the degraded behavior circuits would have after operating a user-specified operating time. Used in conjunction with the SPICE circuit simulator [Nag75, Val80, Qua89], CAS uses the I_{sub} model developed in Chapter 2 with a device degradation model to calculate aged model parameters for each device in a circuit. With CAS, transistor sensitivity to circuit output no longer becomes an issue, and since raw circuit behavior is simulated, the effect of hot-carrier degradation on any aspect of the circuit can be studied.

CAS is a successor to SCALE[1] (Substrate Current And Lifetime Evaluator) [Jen87, Kuo87a, Kuo87b, Kuo88], which computes device-level degradation information such as the I_{sub} waveform and device lifetime. 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 no modifications to SPICE is necessary; 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. 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 devices at varying degrees of stress with model parameter sets associated with each different stress level. 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

[1]Early versions were called SCALP (Substrate Current And Lifetime Processor).

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)[2]. Any mixture of the models can be used in the SPICE input deck.

Presently, verification has been obtained mainly for the dominant NMOSFET degradation. 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 3.2 and 3.3 outlines the NMOSFET device lifetime model and aging models, respectively, while Section 3.4 describes a preliminary degradation and aging model for PMOSFET devices. Section 3.5 includes information on system configuration, program installation, and program usage (including a command summary and simulation example), and Section 3.6 describes the UNIX shell script which enhances the capabilities and convenience of running BERT. Section 3.7 presents a simulation case study on CMOS clocked digital registers, and Section 3.8 provides experimental verification of CAS. Section 3.8 contains a programmers guide to aid in adding new models to BERT as a whole and BERT-CAS. Finally, the chapter ends 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

[2]To use BSIM1 in SPICE2 requires a special version of SPICE2G.6 [She85, Jen87].

BERT manuals [Ros90] and [Lie90a] respectively, as well as in the conference papers [Ros89] and [Lie90b].

This chapter 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 manual [Qua89] for more information. [Jen87] is also recommended for more information concerning the BSIM1 model.

3.2 NMOSFET Device Degradation Model

3.2.1 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.

3.2.2 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]

$$\Delta D = At^n \tag{3.2.1}$$

where

$$A = C_1 \left[\frac{I_{ds}}{W}\exp\left(-\phi_{it}/q\lambda E_m\right)\right]^n \tag{3.2.2}$$

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],

$$\frac{I_{sub}}{I_{ds}} = C_2 \exp\left(-\phi_i/q\lambda E_m\right) \tag{3.2.3}$$

where ϕ_i is the critical energy required for impact ionization and C_2 is a process technology constant. Eq. 3.2.3 can be re-arranged in the following manner:

$$\exp\left(-\phi_{it}/q\lambda E_m\right) = \exp\left(-\phi_i/q\lambda E_m\right)^{\phi_{it}/\phi_i} = \left[\frac{I_{sub}}{C_2 I_{ds}}\right]^m \tag{3.2.4}$$

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

By substituting the exponential term in Eq. 3.2.2 with Eq. 3.2.4 and merging all constants into the parameter H, we can obtain

$$A = \left[\frac{I_{ds}}{WH}\left[\frac{I_{sub}}{I_{ds}}\right]^m\right]^n \tag{3.2.5}$$

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. 3.2.1 becomes

$$\Delta D = \left[\frac{I_{ds}}{WH}\left[\frac{I_{sub}}{I_{ds}}\right]^m\right]^n t^n \tag{3.2.6}$$

From Eq. 3.2.6, 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):

$$\tau = BWI_{sub}^{-m} I_{ds}^{m-1} \tag{3.2.7}$$

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

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

$$\Delta D(t_1) = A_1 \Delta t^{n_1} = A_2 t'^{n_2} \tag{3.2.8}$$

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

$$\Delta D(t_2) = A_2(t' + \Delta t)^{n_2} \tag{3.2.9}$$

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. 3.2.1 only applies to DC stressing where A and n are constant, and a perfectly fresh device is assumed at t = 0.

Using (3.2.8) to solve for t', we can substitute t' in (3.2.9) and obtain

$$\Delta D(t_2) = A_2 \left[\frac{\Delta D(t_1)^{1/n_2}}{A_2} + \Delta t \right]^{n_2} = [\Delta D(t_1)^{1/n_2} + A_2^{1/n_2} \Delta t]^{n_2} \tag{3.2.10}$$

Eq. 3.2.10 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. 3.2.10 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

$$N = \frac{\tau}{T} \tag{3.2.11}$$

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

$$\Delta D(T) = A_{eff} T^{n_{eff}} \tag{3.2.12}$$

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. 3.2.11 with Eq. 3.2.12, we get

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

or

$$\tau = TN = T\left[\frac{\Delta D_f}{\Delta D(T)}\right]^{1/n_{eff}} \tag{3.2.13}$$

Conversely, to find the amount of degradation at a certain specified future time t_{sp}, Eq. 3.2.13 can be re-arranged so that

$$\Delta D(t_{sp}) = \Delta D(T)\left[\frac{t_{sp}}{T}\right]^{n_{eff}} \tag{3.2.14}$$

To find A_{eff} and n_{eff}, we need to use Eq. 3.2.12 with two different ΔD values because of the two unknown parameters involved. The ΔD calculation of Eq. 3.2.10 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}.

3.2.3 Substrate Current Model

The substrate current model is an empirical model developed for the BSIM1 parameter extraction program discussed in Chapter 2. This model is used for all SPICE models. The following is a summary of the equations and parameters used. Refer to [Jen87] and Chapter 2 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

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

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

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

$A_i = 2 \times 10^2$ 1/μm for NMOS devices

$= 1 \times 10^3$ 1/μm for PMOS devices

$B_i = 1.7 \times 10^2$ V/μm for NMOS devices

$= 3.7 \times 10^2$ V/μm for PMOS devices

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 3.5.4 for default values.

3.2.4 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 (BSIM1 included). More information on creating the modified model parameter file is given in Section 3.5.

3.2.5 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. 3.2.6 of Section 3.2.2 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 [Cho87]. 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. 3.2.7 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. 3.2.7, we can obtain the following expression:

$$\tau\left(\frac{I_{ds}}{W}\right) = H\Delta D_f^{1/n}\left(\frac{I_{sub}}{I_{ds}}\right)^{-m} \qquad (3.2.15)$$

If we plot Eq. 3.2.15 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. 3.2.15, 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.

3.2.6 Enhanced AC Degradation

There have been several publications describing enhanced hot-carrier degradation from AC effects [Wer86, Cho87, Aur89]. These effects can be categorized 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 (Section 3.8). 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) dV_{gs} / dt > 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.

3.2.7 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. 3.2.10, 3.2.13 and 3.2.14 are the actual equations implemented in CAS.

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

3.3 NMOSFET Circuit Aging Model

3.3.1 Introduction

This section describes the models and formulation 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.

3.3.2 Model Formulation

To determine the amount of degradation that occurs in a device, we must look back at the degradation equations of Section 3.2. 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. 3.2.6 of Section 3.2, we can describe this degradation by the form

$$\Delta D = f(At^n) = f\left(\left[\frac{I_{ds}}{WH}\left(\frac{I_{bs}}{I_{ds}}\right)^m\right]^n\right) t^n \qquad (3.3.1)$$

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:

$\Delta D = f(Age^n)$

$$Age = \frac{I_{ds}}{WH}\left(\frac{I_{bs}}{I_{ds}}\right)^m t \qquad (3.3.2)$$

Eq. 3.3.2 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,

$$Age(T) = \int_{t=0}^{T} \frac{I_{ds}}{WH}\left(\frac{I_{bs}}{I_{ds}}\right)^m dt \qquad (3.3.3)$$

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. 3.3.2. 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.

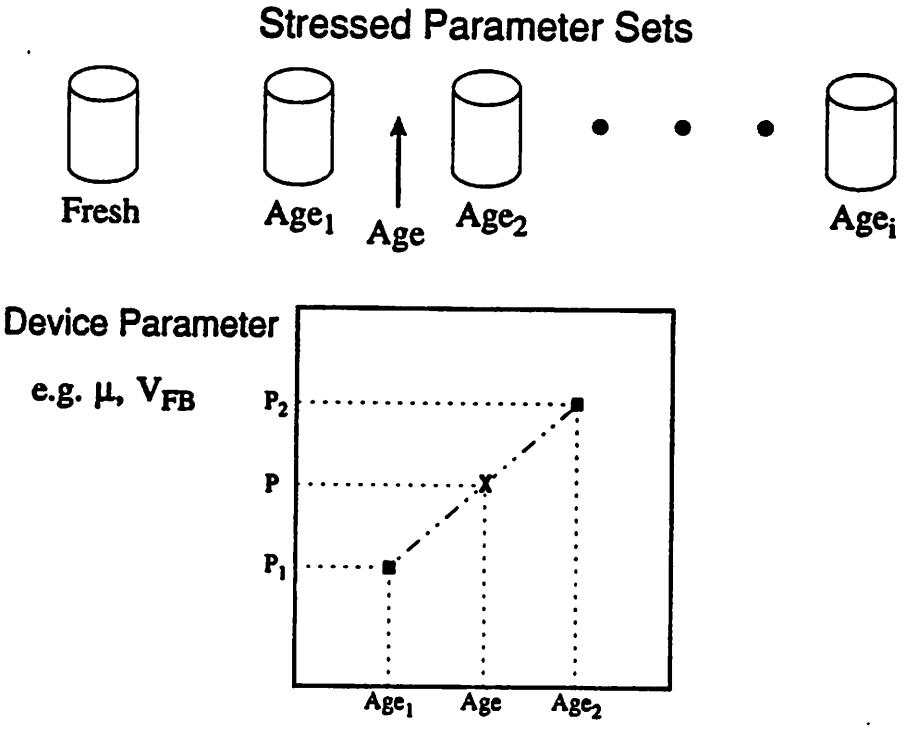


Fig. 3.1 Calculation of the aged parameter from pre-stressed model parameter sets. The barrels represent the various model parameter sets with different ages Age_1, Age_2, etc., while the in-circuit device suffers degradation represented by Age.

The reason we have generalized the relationship between degradation and At^n in Eq. 3.3.1 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. 3.3.1 is assumed.

For BSIM1 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

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 3.2.6 equally applies to circuit aging.

3.3.3 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.

3.4 PMOSFET Degradation And Aging Models

3.4.1 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 [Tsu85, Hir86, Tzo86, Bra88, Ong88, Ong89], we have incorporated both currents through a weighting coefficient that can be specified by the user.

3.4.2 Gate Current Model

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

$$I_{gate} = G_1 \frac{I_{sub} t_{ox}}{\lambda_r} \left[\frac{\lambda E_m}{\phi_b}\right]^2 P(E_{ox}) \exp\left[-\frac{\phi_b}{E_m \lambda}\right] \tag{3.4.1}$$

where

$$P(E_{ox}) = \left[\frac{5.66 \times 10^{-6} E_{ox}}{1 + \frac{E_{ox}}{1.45 \times 10^5}} \times \frac{1}{1 + \frac{2 \times 10^{-3}}{L_{eff}} \exp(-E_{ox} t_{ox}/1.5)} + 2.5 \times 10^{-2}\right] \exp(-300/\sqrt{E_{ox}}) \tag{3.4.2}$$

for $E_{ox} >= 0$, and

$$P(E_{ox}) = 2.5 \times 10^{-2} \exp(-X_{ox}/\lambda_{ox}) \tag{3.4.3}$$

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

$$\phi_b = 3.2 - 2.6 \times 10^{-4} \sqrt{E_{ox}} - \upsilon E_{ox}^{2/3} \tag{3.4.4}$$

where $\upsilon = 4 \times 10^{-5}$ 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.

3.4.3 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. 3.2.7,

$$\tau = B\left[\frac{I_{gate}}{W}\right]^{-m} \tag{3.4.5}$$

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,

$$\Delta D = A\, t^n \tag{3.4.6}$$

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

$$\Delta D_f = A\tau^n \tag{3.4.7}$$

Solving for the coefficient A using Eqs. 3.4.5 and 3.4.7, we get

$$A = \frac{\Delta D_f}{B^n}\left[\frac{I_{gate}}{W}\right]^{mn} t^n \tag{3.4.8}$$

Substituting

$$H = \frac{B}{\Delta D_f^{1/n}} \tag{3.4.9}$$

we finally get

$$\Delta D = \left[\frac{1}{H}\left[\frac{I_{gate}}{W}\right]^m\right]^n t^n \tag{3.4.10}$$

The conversion of the parameter B to H in Eq. 3.4.9 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. 3.4.10, we can parallel the NMOS analysis and propose the following expression for Age,

$$\Delta D = (Age)^n$$

Then,

$$Age = \frac{1}{H}\left[\frac{I_{gate}}{W}\right]^m t \tag{3.4.11}$$

3.4.4 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,

$$Age = W_b \times (Age \text{ from } I_{sub}) + W_g \times (Age \text{ from } I_{gate}) \tag{3.4.12}$$

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:

$$Age = \left\{W_b\left[\frac{I_{ds}}{WH_b}\left(\frac{I_{sub}}{I_{ds}}\right)^{m_b}\right] + W_g\left[\frac{1}{H_g}\left(\frac{I_{gate}}{W}\right)^{m_g}\right]\right\} t \tag{3.4.13}$$

To calculate the degradation expression, to conform with

$$\Delta D = (Age)^n$$

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

$$\Delta D = \left\{W_b\left[\frac{I_{ds}}{WH_b}\left(\frac{I_{sub}}{I_{ds}}\right)^{m_b}\right] + W_g\left[\frac{1}{H_g}\left(\frac{I_{gate}}{W}\right)^{m_g}\right]\right\}^n t^n \tag{3.4.14}$$

Eq. 3.4.13 and 3.4.14 are then used in the degradation and aging calculations discussed in Sections 3.2 and 3.3.

3.4.5 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. 3.4.1 and υ from Eq. 3.4.4 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. 3.4.6). $-m_g$ and B_g are the slope and intercept respectively when device lifetime τ is plotted against I_{gate} in log-log format (Eq. 3.4.5). 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. 3.4.9). 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 ϕb 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 3.5.4). The default value of W_g is 1 if the '.PMOSDEG' command is specified; otherwise W_g defaults to 0.

3.4.6 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.

3.5 BERT Configuration and Operation

3.5.1 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].

3.5.2 System Configuration

Figs. 3.2 and 3.3 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. 3.2). 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. 3.2). To create the aged model parameters, the pre-processor is run once again with the original input file as its argument (Fig. 3.3). 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. 3.3). 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.

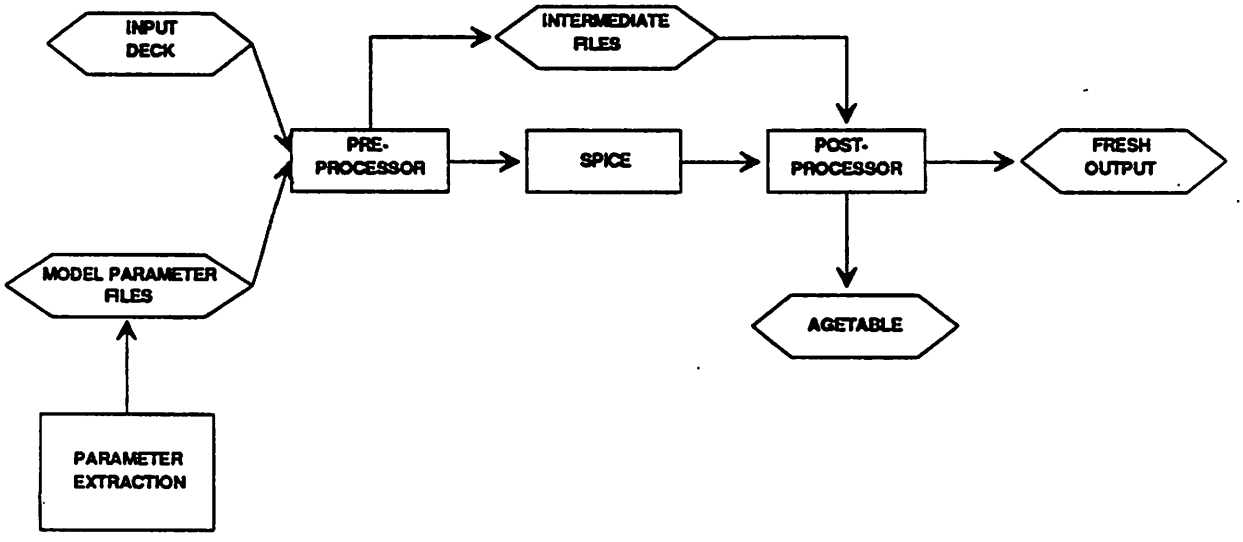


Fig. 3.2 BERT system configuration: First pass is to calculate degradation information (such as device lifetime) and the agetable.

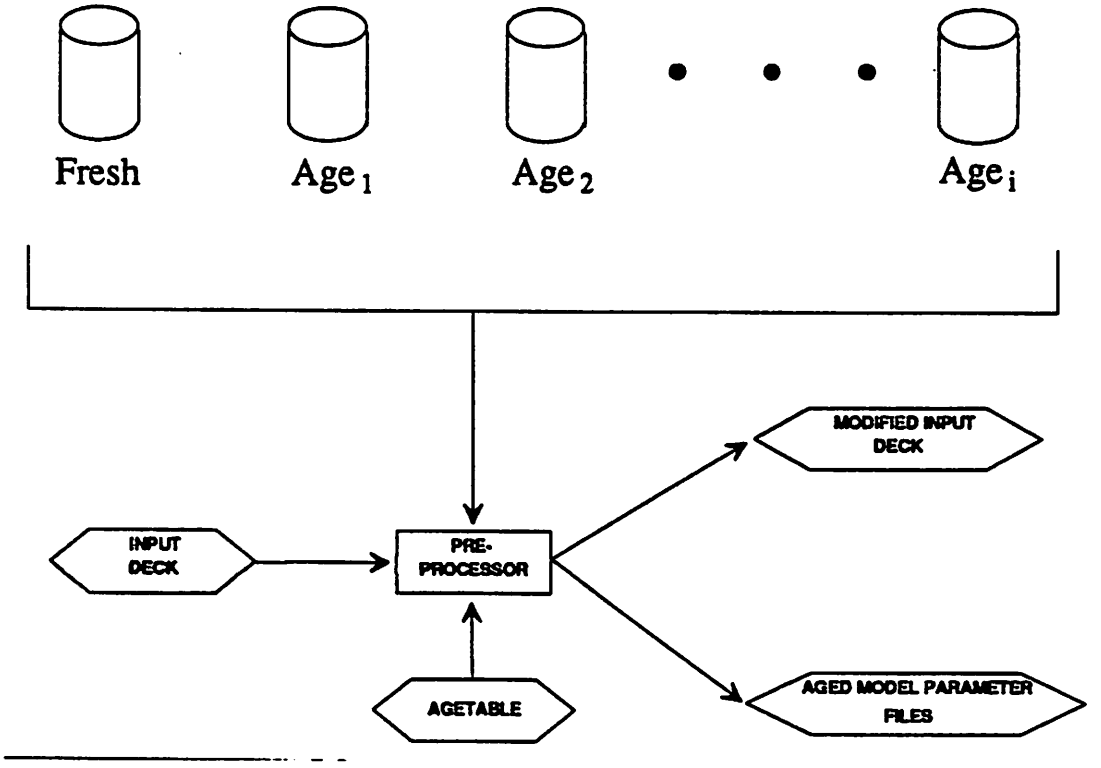


Fig. 3.3 BERT system configuration: Second pass is to generate the aged model parameters at the future time point specified by the '.age' command.

3.5.3 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, and *spice* is the name of the SPICE simulator used. 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. The file **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.

3.5.4 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 $\Delta g_m/g_{m0}$ **(AGEDGM)**, 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. 3.6) 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 Section 3.4.4 for the model description.

(10) **.PRINTIGATE** or **.PLOTIGATE**

.PRINTIGATE *MXXX <MYYY ... MZZZ> <ALL>*
.PRINTIGATE *SXXX <SYYY ... SZZZ> <ALL>*
.PLOTIGATE *MXXX <MYYY ... MZZZ> <ALL> <(MIN,MAX)>*
.PLOTIGATE *SXXX <SYYY ... SZZZ> <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. *SXXX* is the transistor denotation for the BSIM1 model in SPICE2, while *MXXX* 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 *MXXX* <*MYYY* ... *MZZZ*> <ALL>
PRINTISUB *SXXX* <*SYYY* ... *SZZZ*> <ALL>
.PLOTISUB *MXXX* <*MYYY* ... *MZZZ*> <ALL> <(*MIN*,*MAX*)>
.PLOTISUB *SXXX* <*SYYY*... *SZZZ*> <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. *SXXX* is the transistor denotation for the BSIM1 model in SPICE2, while *MXXX* 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. 3.4).

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 $l_c / \sqrt{t_{ox}}$	$\mu m^{1/2} \cdot V$	0.0
48	LC2	Bias-sensitivity term of $l_c / \sqrt{t_{ox}}$	$\mu m^{1/2} \cdot V^{-1}$	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 $l_c / \sqrt{t_{ox}}$	$\mu m^{1/2} \cdot V$	0.0
51	LC5	Bias-sensitivity term of $l_c / \sqrt{t_{ox}}$	$\mu m^{1/2} \cdot V^2$	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 $l_c / \sqrt{t_{ox}}$	$\mu m^{1/2} \cdot V$	0.0
54	HG0	Degradation plot intercept (H_0)	A sec / (m $V^{1/n}$)	1.0E4
55	HGD	Degradation plot intercept (H_{gd})	A sec / (m $V^{(n+1)/n}$)	0.0

56	NN0	Slope of degradation parameter (n_0)		0.5
57	NNGD	Slope of degradation parameter (n_{gd})	V^{-1}	0.0
58	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. 3.4 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.

	Name	L sens. factor	W sens. factor	Units of basic parameter
1	V_{FB} (VFB)	V_{FB1} (LVFB)	V_{FBw} (WVFB)	V
2	ϕ_S (PHI)	ϕ_{S1} (LPHI)	ϕ_{Sw} (WPHI)	V
3	K_1 (K1)	K_{11} (LK1)	K_{1w} (WK1)	$V^{1/2}$
4	K_2 (K2)	K_{21} (LK2)	K_{2w} (WK2)	-
5	η_0 (ETA)	η_{01} (LETA)	η_{0w} (WETA)	-
6	μ_Z (MUZ)	δ_1 (DL)	δ_w (DW)	$cm^2/V\text{-}s$, µm, µm
7	U_{0Z} (U0)	U_{0Z1} (LU0)	U_{0Zw} (WU0)	V^{-1}
8	U_{1Z} (U1)	U_{1Z1} (LU1)	U_{1Zw} (WU1)	µm V^{-1}
9	μ_{ZB} (X2MZ)	μ_{ZB1} (LX2MZ)	μ_{ZBw} (WX2MZ)	$cm^2/V^2\text{-}s$
10	η_B (X2E)	η_{B1} (LX2E)	η_{Bw} (WX2E)	V^{-1}
11	η_D (X3E)	η_{D1} (LX3E)	η_{Dw} (WX3E)	V^{-1}
12	U_{0B} (X2U0)	U_{0B1} (LX2U0)	U_{0Bw} (WX2U0)	V^{-2}
13	U_{1B} (X2U1)	U_{1B1} (LX2U1)	U_{1Bw} (WX2U1)	µm V^{-2}
14	μ_S (MUS)	μ_{S1} (LMS)	μ_{Sw} (WMS)	$cm^2/V^2\text{-}s$
15	μ_{SB} (X2MS)	μ_{SB1} (LX2MS)	μ_{SBw} (WX2MS)	$cm^2/V^2\text{-}s$
16	μ_{SD} (X3MS)	μ_{SD1} (LX3MS)	μ_{SDw} (WX3MS)	$cm^2/V^2\text{-}s$
17	U_{1D} (X3U1)	U_{1D1} (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_{crit01} (LECRIT0)	E_{crit0w} (WECRIT0)	V/cm
25	E_{critg} (ECRITG)	E_{critg1} (LECRITG)	E_{critgw} (WECRITG)	1/cm
26	E_{critb} (ECRITB)	E_{critb1} (LECRITB)	E_{critbw} (WECRITB)	1/cm
27	l_{c0} (LC0)	l_{c01} (LLC0)	l_{c0w} (WLC0)	$\mu m^{1/2}$
28	l_{c1} (LC1)	l_{c11} (LLC1)	l_{c1w} (WLC1)	$\mu m^{1/2}$– V
29	l_{c2} (LC2)	l_{c21} (LLC2)	l_{c2w} (WLC2)	$\mu m^{1/2}$– V^{-1}
30	l_{c3} (LC3)	l_{c31} (LLC3)	l_{c3w} (WLC3)	$\mu m^{1/2}$
31	l_{c4} (LC4)	l_{c41} (LLC4)	l_{c4w} (WLC4)	$\mu m^{1/2}$– V
32	l_{c5} (LC5)	l_{c51} (LLC5)	l_{c5w} (WLC5)	$\mu m^{1/2}$– V^2
33	l_{c6} (LC6)	l_{c61} (LLC6)	l_{c6w} (WLC6)	$\mu m^{1/2}$
34	l_{c7} (LC7)	l_{c71} (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. 3.4 The modified BSIM1 model parameter file format to be used with BERT.

(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 MOSFET's:

SXXX nd ng ns nb mname < W=*value* > <L=*value* > ...etc.
MXXX 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 *SXXX* for the BSIM1 model in SPICE2, or *MXXX* 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).

3.5.5 Circuit Example: 21-Stage CMOS Inverter Chain

Fig. 3.5 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. 3.6 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. 3.7 shows the generated agetable with the Age that all the transistors would have after 10 years of operation. Fig. 3.8 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.

3.5.6 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.

3.5.7 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.

```
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
```

Fig. 3.5 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 two pages).

```
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
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
```

Fig. 3.5 (cont.) SPICE3 input deck for a 21-stage CMOS inverter chain with substrate current, gate current, device lifetime, and circuit aging calculations requested.

```
.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. 3.5 (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. 3.6 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. 3.7 The agetable generated by CAS of all the transistors in the circuit.

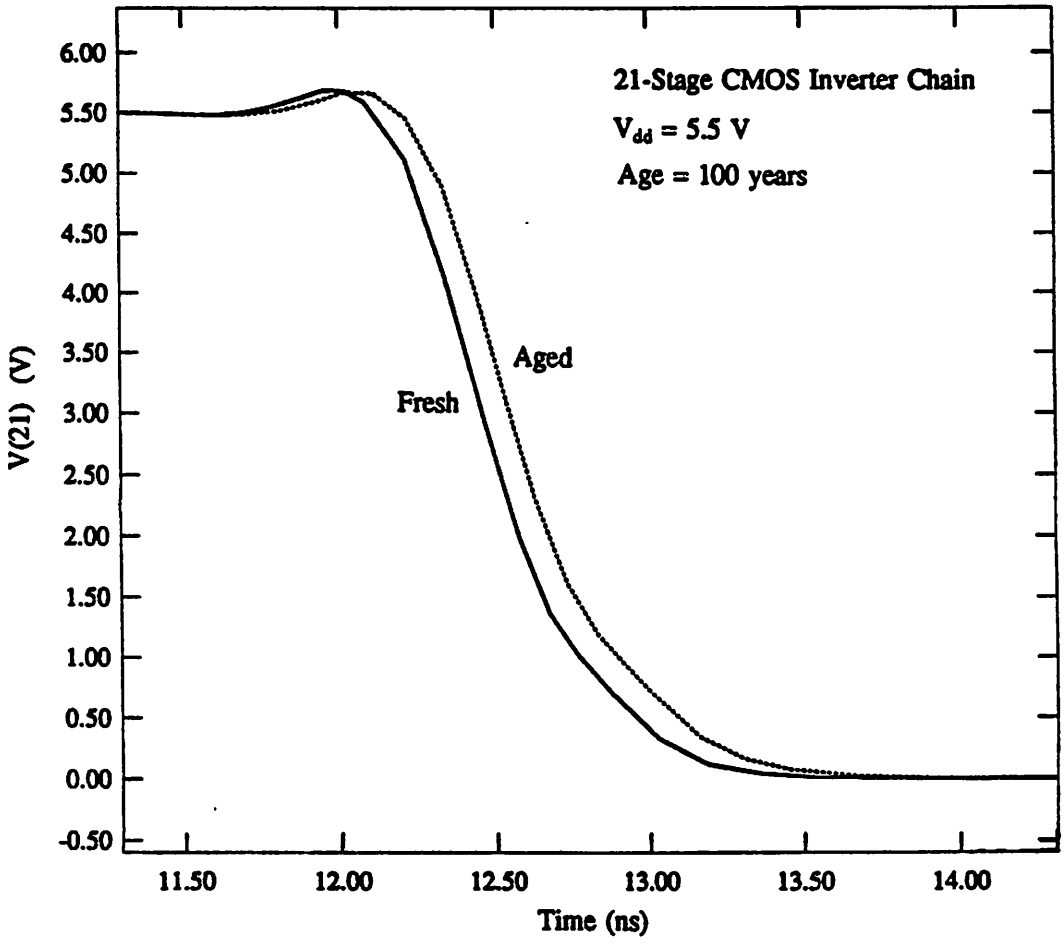


Fig. 3.8 The voltage waveform at the output of the 20th stage showing the propagation delay difference between the fresh and aged inverter chain.

3.6 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. 3.9 shows the main menu. Seven different options are available depending on whether a one-pass simulation (such as calculating the degradation information of Fig. 3.6, 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. 3.10 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. 3.9), and option (6) exits the shell.

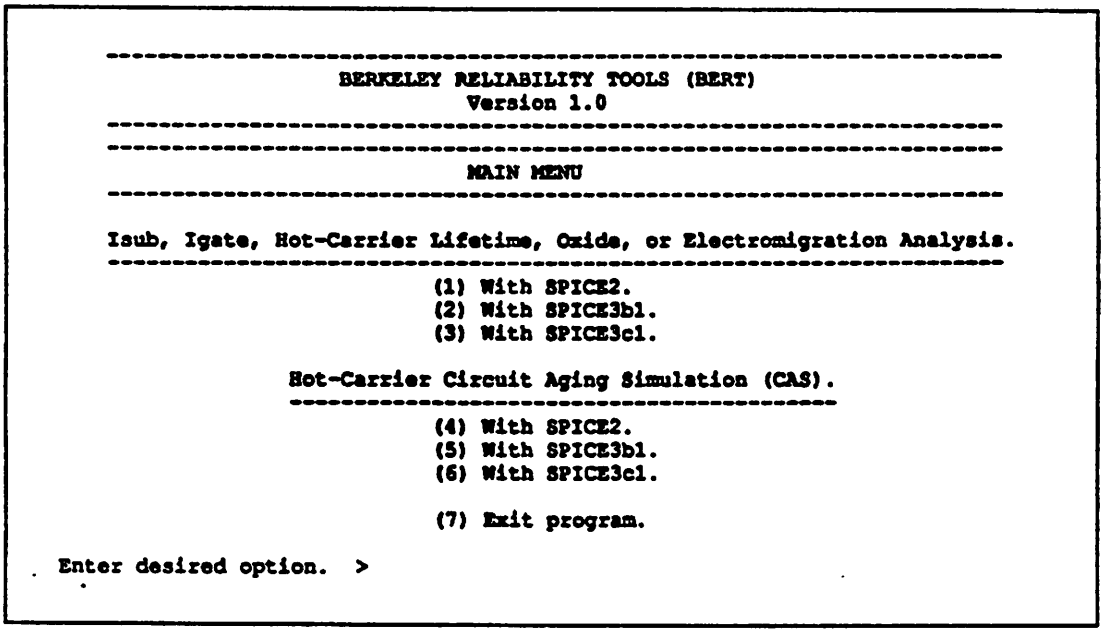


Fig. 3.9 The initial main menu of the shell script program.

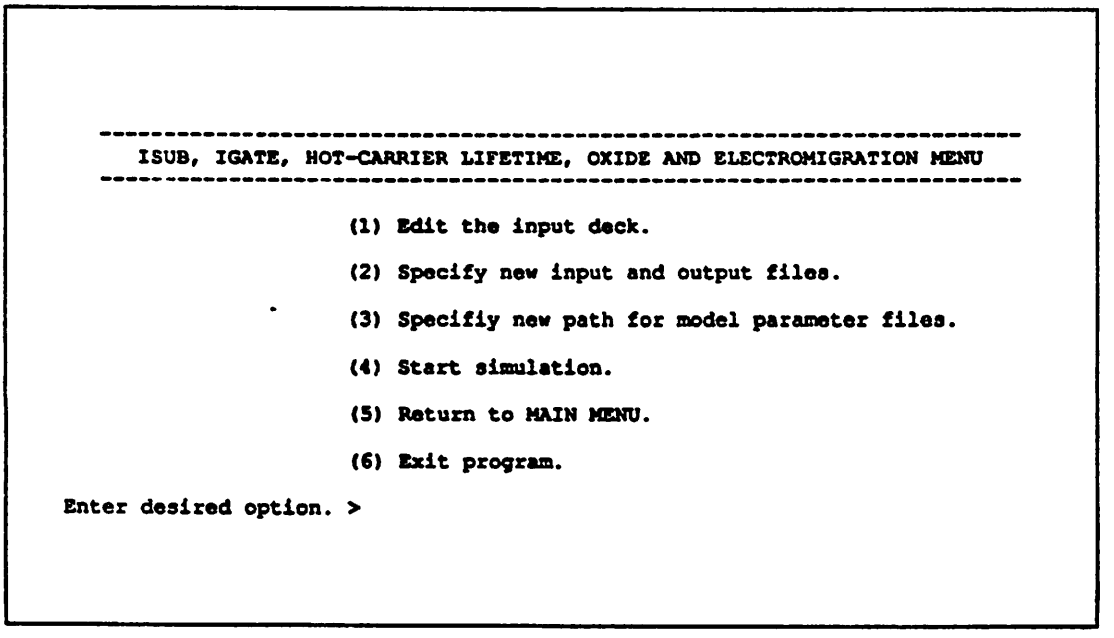


Fig. 3.10 The one-pass menu (option (1), (2), or (3) of the main menu).

Fig. 3.11 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.

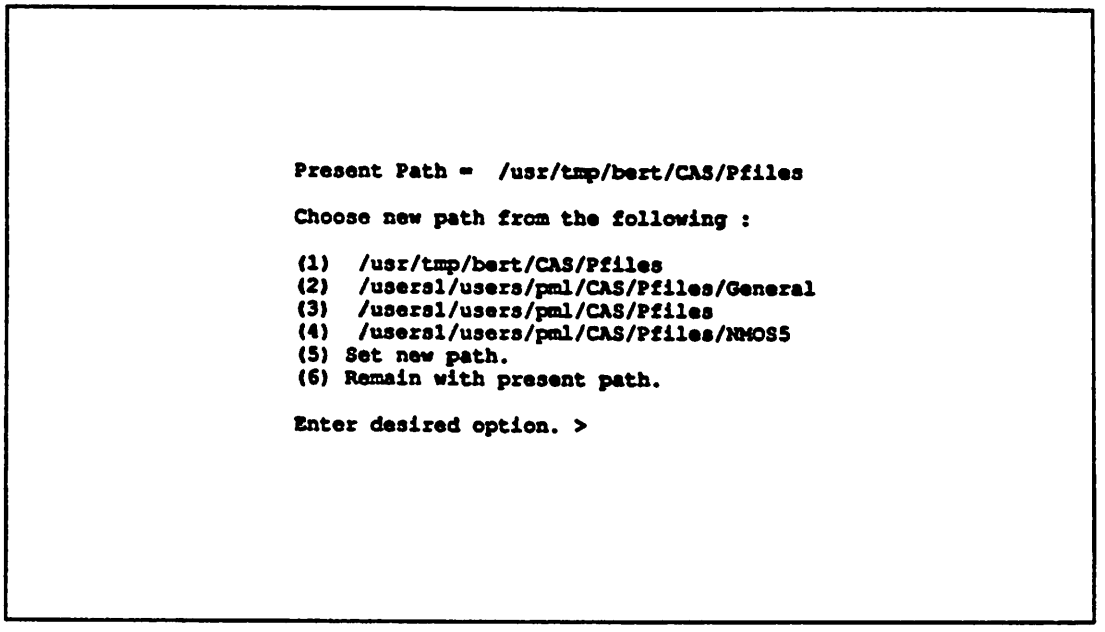


Fig. 3.11 Changing the path of the location of the model parameter files (option (3) of the one-pass menu).

After doing all the necessary adjustments, the user can select option (4) in the one-pass menu (Fig. 3.10) 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. 3.12). Here, the user has the choice of viewing the newly created output file, going back to the main menu (Fig. 3.9), 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. 3.13). 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

```
                    <<<<<< 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. 3.12 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. 3.13 The CAS menu (option (4), (5) or (6) of the main menu).

account. Greater accuracy can undoubtedly 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. 3.13), 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. 3.14). 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.

```
                <<<<<< 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. 3.14 CAS simulation diagnostics and END OF SIMULATION menu.

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. 3.15). 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 3.5.3). The second modification is to set the paths for the location of the process files that will appear in the path selection menu (Fig. 3.11). The text after the equal sign in the "set PfDir*x* = " 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.

```
#!/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. 3.15 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.

3.7 Simulation Case Study: CMOS Clocked Digital Registers

3.7.1 Introduction

In this section, simulation examples are presented that study individual transistor degradation of two different CMOS clocked shift registers[1]. CAS is used to calculate device lifetime and degradation waveforms of individual transistors in the circuit to analyze what factors are responsible for some transistors experiencing more degradation than others.

Block diagrams of the two different CMOS clocked shift registers are shown in Fig. 3.16. The two configurations are similar except that clocking control is accomplished using transmission gates in the circuit labeled Cir1 while Cir2 uses clocked inverters for the same function.

The following sections presents the actual simulation setup and the results and conclusions obtained.

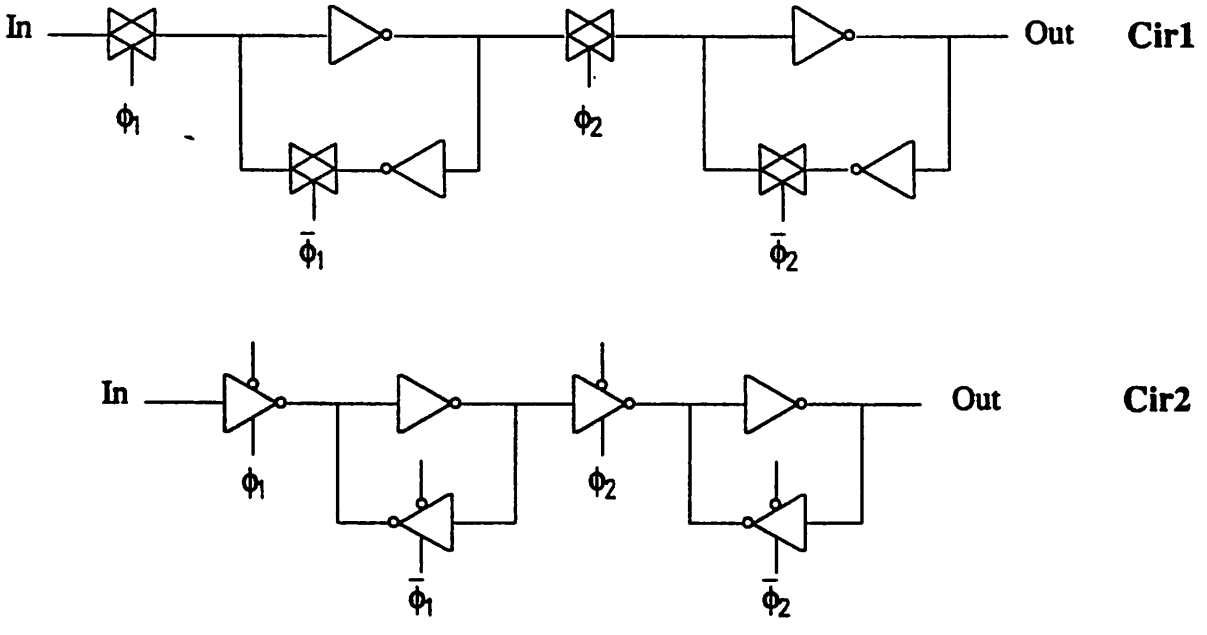


Fig. 3.16 Two different CMOS clocked shift register configurations used in device lifetime studies in this section. Cir1 uses transmission gates, while Cir2 uses clocked inverters, for clocking control.

[1]Circuit configuration supplied by Boeing Electronics, Seattle, WA 98124

3.7.2 Simulation Setup

For device lifetime analysis, worst case operating conditions were assumed. This corresponded to alternately loading 1's and 0's in the input. All devices would then be switching during each clock transient. Device lifetime was based on 10% drain current degradation ($\Delta I_{ds}/I_{ds0}$).

The applied waveforms were common to both circuits. Fig. 3.17 shows the input voltage waveform and the clocked waveforms for ϕ_1 and ϕ_2 that were used for all simulations. Although the clock signals ϕ_1 and ϕ_2 were directly applied to the circuit, the input voltage was conditioned by placing a two-stage inverter chain between the input voltage and the actual input of the shift register. A load capacitance of 2 pF was placed at the output of all circuits. To approximately take into account junction capacitance, a 0.025 pF per drain/source area capacitor was placed at each node, i.e., for each drain or source of a MOSFET connected to a node, 0.025 pF was added to the node.

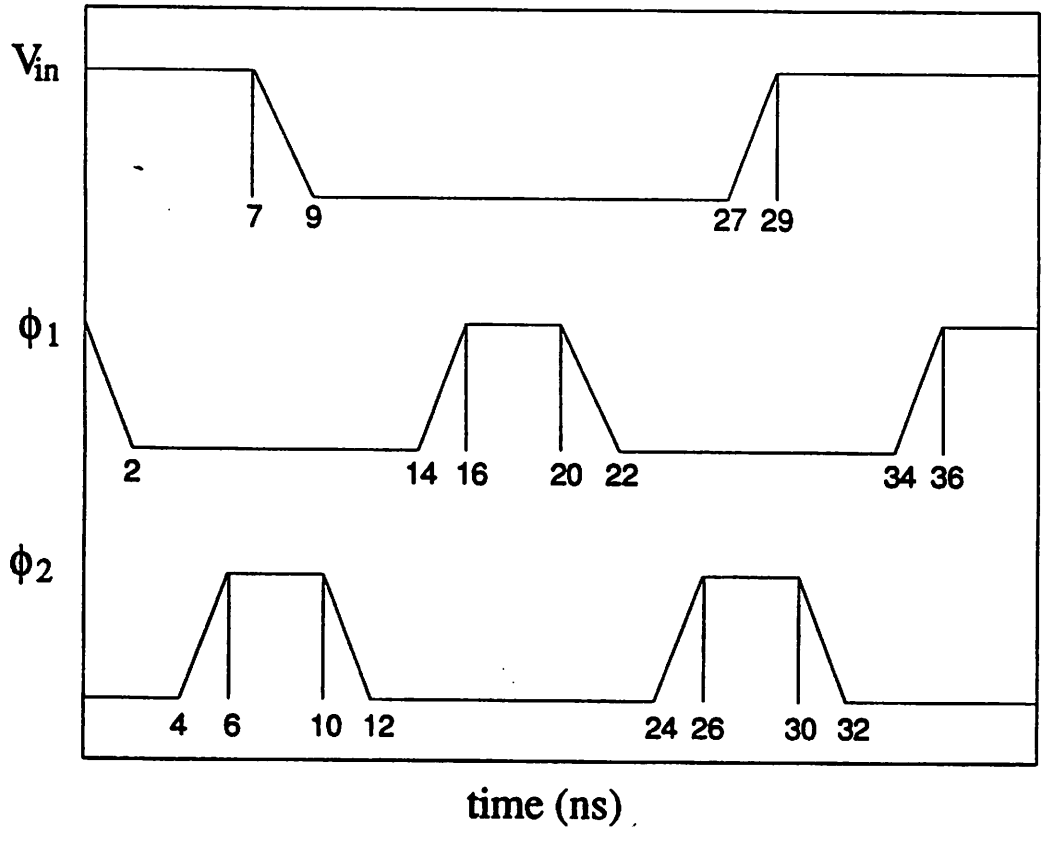


Fig. 3.17 Voltage waveforms for the input voltage and the two clocks.

The BSIM1 model was used for simulating the drain currents of all MOSFET's. NMOS device parameters were extracted from a 1.5 μm technology CMOS LDD process with t_{ox} = 200 A. The PMOS device parameters were extracted from a 1.5 μm technology CMOS conventional process with t_{ox} = 250 A. In all simulations, channel lengths for both devices were set to $L_N = L_P$ = 1.4 μm, and channel widths of the NMOS devices were set to W_N = 20 μm. The channel widths of the PMOS devices were set to 20 μm for the transmission gates, and 40 μm for the inverters. In all cases, PMOS devices were assumed not to degrade.

3.7.3 Simulation Results and Discussion

Circuit Configuration 1 (Cir1)

Fig. 3.18 shows the circuit schematic for this transmission-gate-based shift register, and Table 3.1 shows device lifetimes for some of the NMOSFET's (Transistor names start with 'S' rather than 'M' since a special version of SPICE2G.6 with the BSIM1 model is used [She85]). Note that the devices located in the main signal path of the registers (S1, S3, S7, and S9) all degrade similarly. However, one can see that the transmission gate NMOS devices degrade slightly less than the inverter NMOS devices. Looking at Fig. 3.19 we can see that $I_{ds}(I_{sub}/I_{ds})^m$, which is directly related to device degradation (see Section

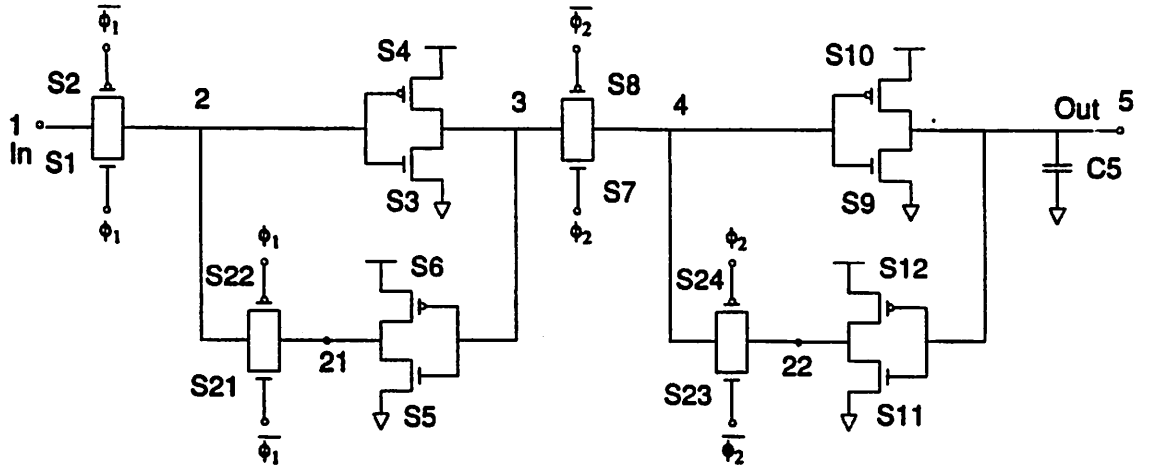


Fig. 3.18 Circuit schematic for register configuration 1 (Cir1). Devices with bubbles at the gate represent PMOS devices.

Cir1: τ in years at $\Delta I_{ds}/I_{ds0}$ = 10%		
Transistor	V_{dd} = 5V	V_{dd} = 5.5V
S1	117.6	5.7
S3	83.9	4.3
S5	128.7	9.9
S7	102.7	5.1
S9	81.5	4.2
S11	47.6	2.4

Table 3.1 Device lifetimes τ for selected NMOSFET's of Cir1. Other NMOS devices had τ = infinity.

3.2), is indeed smaller for S1 when compared to S3. This can be explained somewhat by the fact that when the structures are conducting, total current splits and flows both in the NMOS as well as the PMOS device for the transmission gate, while all of the total current flows through the NMOS device in the inverter. The lesser current results in less degradation in this particular case. However, the lifetime of the transmission gate will vary as the rise/fall time of the clock changes, thus changing the relative amount of degradation as compared to the inverter device.

Another thing to note is the fact that the lightly-loaded inverter device (S3) and the heavily-loaded inverter device (S9) degrade similarly. This behavior can best be explained by comparing the degradation factor $I_{ds}(I_{sub}/I_{ds})^m$ plots (Fig. 3.19 and 3.21) and the voltage waveforms of the gate and drain (Figs. 3.20 and 3.22) of the respective transistors. By looking at the $I_{ds}(I_{sub}/I_{ds})^m$ plots of S3 and S9 (Figs. 3.19 and 3.21), note that the greatest device degradation occurs at low gate voltage, or early in the transient of Figs. 3.20 and 3.22. By comparing the location of the $I_{ds}(I_{sub}/I_{ds})^m$ peak with the voltage transient waveforms of S3 and S9, it can be seen that maximum degradation occurs before the drain voltage has had a chance to drop. Because capacitive loading affects only the fall time of the drain voltage after most of the degradation has taken place, changing the load capacitance has hardly any effect on the device lifetime. Notice again, however, that changing the input voltage rise/fall time will affect the

amount of degradation that occurs. For example, if the input voltage rise is slower, more time is spent during the low V_{gs} high-degradation regime, causing more degradation.

Thirdly, note that from Table 3.1 that the NMOS device of the feedback inverter connected to the output (S11) has a lifetime about half the rest. This result can again be seen by looking at the current and degradation plots. Fig. 3.23 shows the $I_{ds}(I_{sub}/I_{ds})^m$ plots for S9 and S11. Note that S9 suffers degradation only once during the period, while S11 is subject to degradation twice. This again can be explained by looking at the voltage waveforms of the two transistors (Figs. 3.24 and 3.25). Degradation in NMOS devices in inverters mainly occurs when the input voltage changes from low to high [Hsu85]. In this situation, because of the inherent delay of the drain voltage dropping from high to low from loading effects, there is an overlap time when the transistor is on and the drain voltage is high. In the opposite case, when the input voltage changes from high to low, however, the delay of the drain voltage rising from low to high minimizes the time when the transistor is on and saturated. These two cases can be seen for S9, the device in the inverter, from Fig. 3.24. Note that hardly any overlap exists in

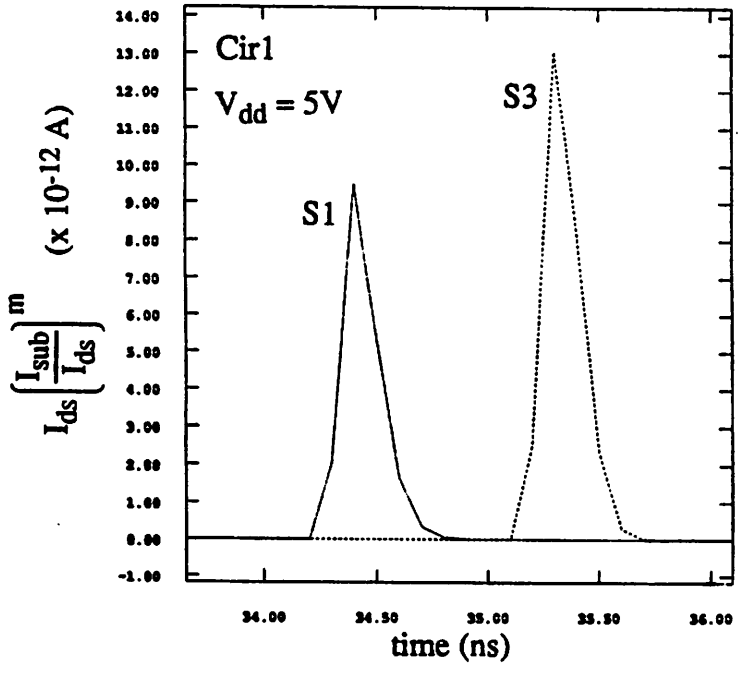


Fig. 3.19 Plots of the degradation factor $I_{ds}(I_{sub}/I_{ds})^m$ for the transmission gate device S1 and the inverter device S3. Note the slightly higher degradation for the inverter transistor.

the initial transient at t ≅ 6 ns, while overlap exists in the second transient at t ≅ 25 ns.

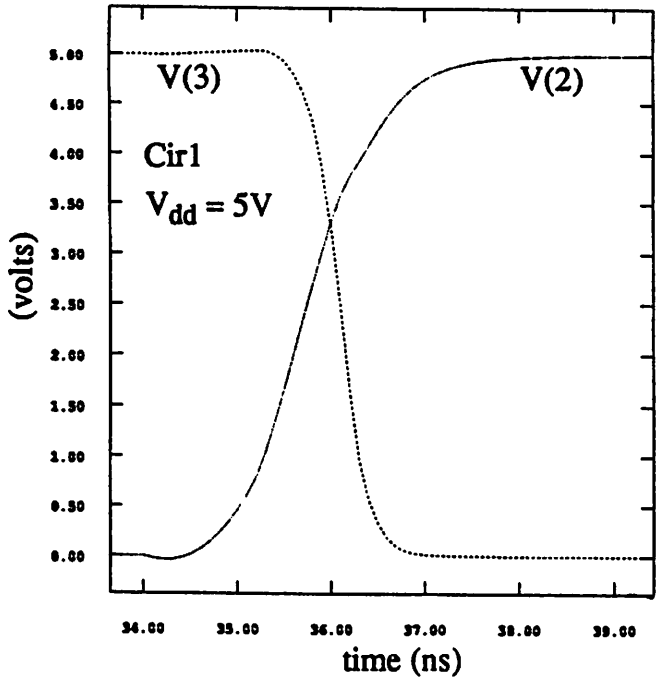


Fig. 3.20 V_{gs} (=V(2)) and V_{ds} (=V(3)) of the inverter device S3.

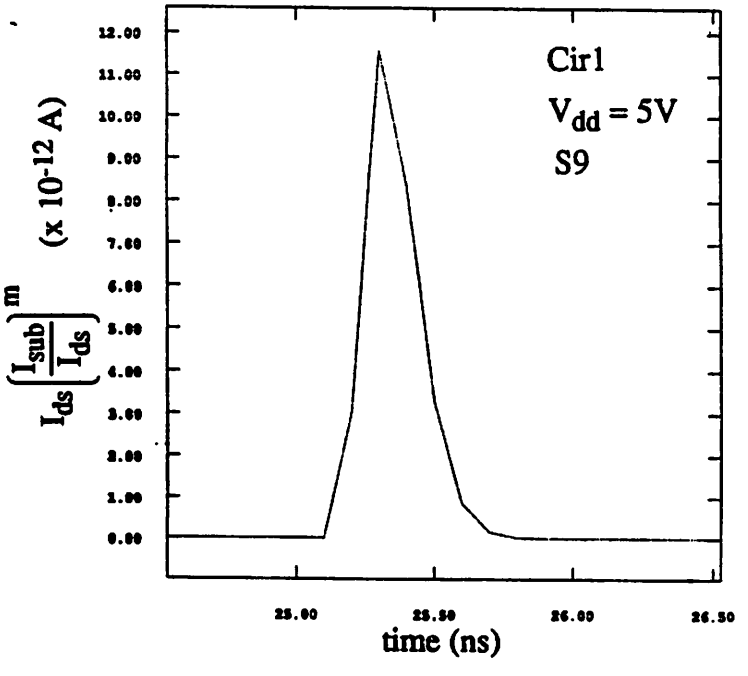


Fig. 3.21 Plot of the degradation factor $I_{ds}(I_{sub}/I_{ds})^m$ for the loaded inverter device S9.

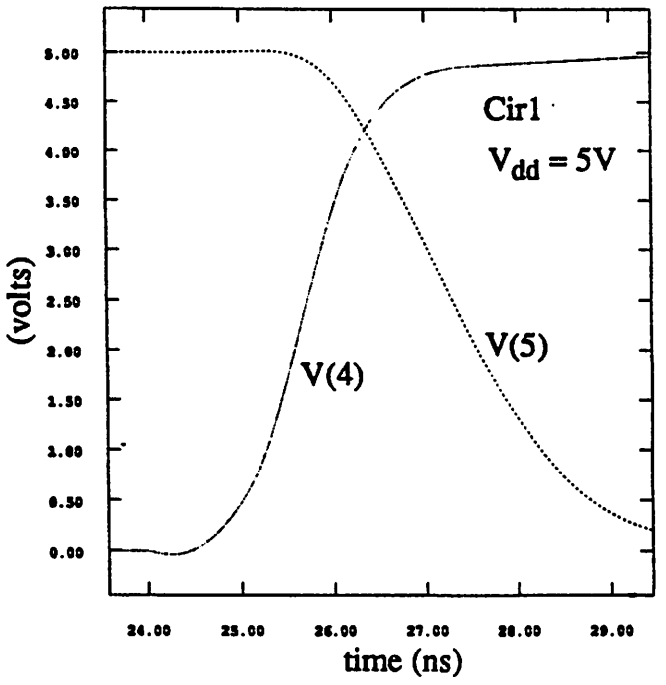


Fig. 3.22 V_{gs} (=V(4)) and V_{ds} (=V(5)) of the loaded inverter device S9.

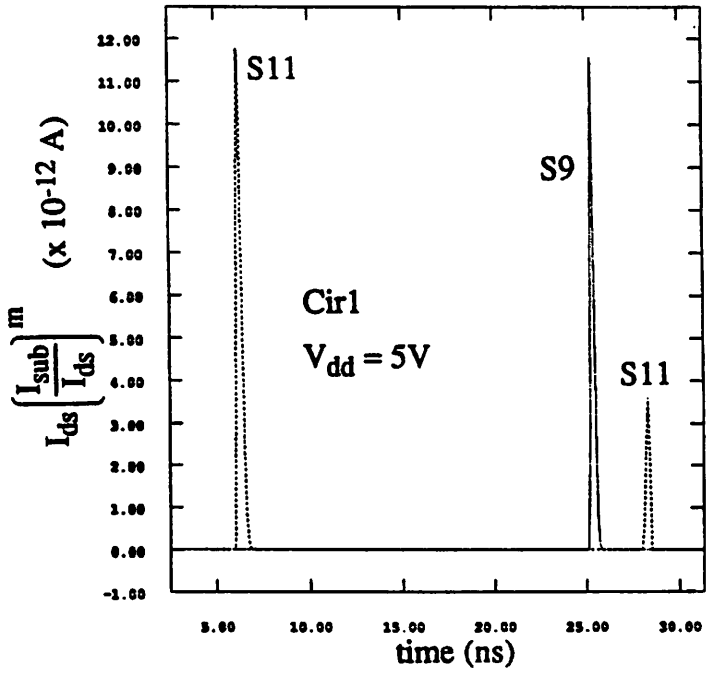


Fig. 3.23 Plot of the degradation factor $I_{ds}(I_{sub}/I_{ds})^m$ for the devices S9 and S11. Notice that S11 experiences degradation twice during the time period.

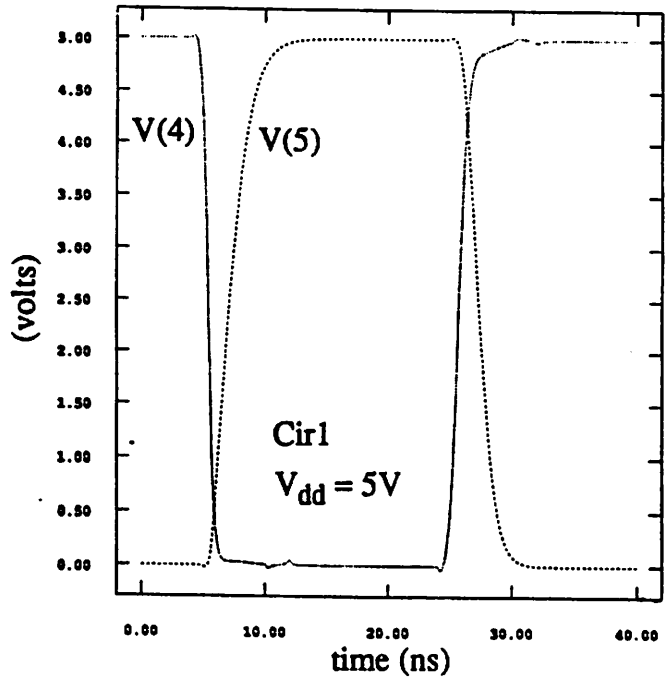


Fig. 3.24 V_{gs} (=V(4)) and V_{ds} (=V(5)) of the inverter device S9. Notice the minimal overlap of the first transient.

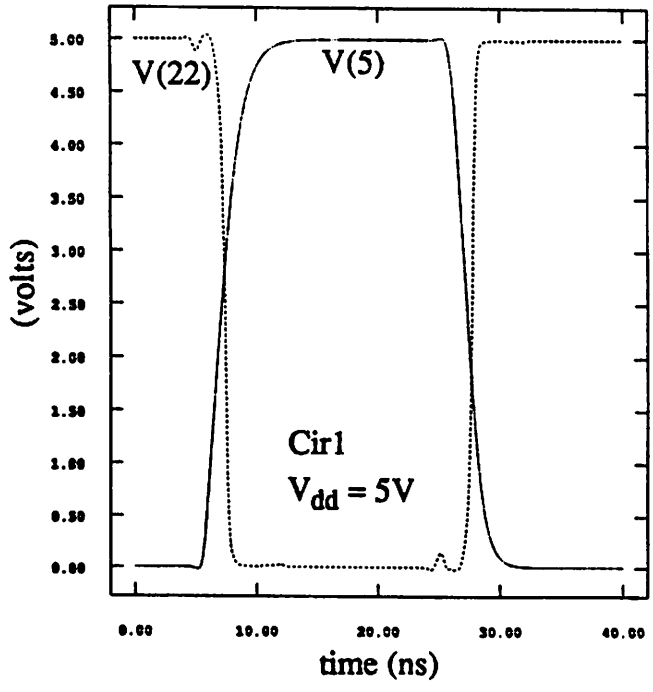


Fig. 3.25 V_{gs} (=V(5)) and V_{ds} (=V(22)) of the inverter device S11. Notice that overlap exists in both transients.

However, for the case of the S11 device, note that the output of the inverter (node 22) does not change solely as a consequence of the input voltage of the inverter changing, but that the voltage of this node changes when the transmission gate turns on with the inverse of ϕ_2 clock. In the situation when the input voltage of the inverter goes from high-to-low ($t \cong 25$ ns), the inverse of ϕ_2 turns on the transmission gate at the same time the voltage transient is taking place. Thus, node 4 (which is already high) will pull the output of the inverter up earlier than if the inverter itself was solely responsible for pulling the output node up. This speed-up increases the overlap time between the gate and drain voltage of S11 (Fig. 3.25, $t \cong 25$ ns), causing degradation to occur even in this "non-overlap" case. In addition, note that the input of this inverter is connected to the heavily-loaded output. Thus, the slow gate voltage fall in combination with the fast drain voltage rise causes more degradation to occur (hence the difference in lifetime between the two feedback inverter devices S5 and S11).

Finally, circuit aging analysis was done. Degraded device model parameters were calculated assuming circuit operation at $V_{dd} = 5$ volts for a period of 100 years.

Fig. 3.26 shows the simulated results of the output voltage V(5) for both the fresh and 100-year aged case. Note that practically no degradation can be seen. Fig. 3.27 is an expanded view of the falling transient, showing that the falling transient is slightly slower in the aged case. The fact that no noticeable degradation can be seen is a result of two factors:

(1) From previous simulation and measurement results [Lee88], it has been seen that digital inverter-based circuits seem to be fairly robust to hot-carrier degradation.

(2) In clocked circuits, the propagation delay time of signals is equal to the total delay time of the stages that are between the clocked stages, rather than the total delay time of all stages in non-clocked circuits.

In Cir1, the maximum number of stages between clocks is equal to one. Thus, the degradation that can be seen at the output of the circuit is effectively of only one inverter, namely the last one. In contrast, for a non-clocked inverter chain of, for example, 100 stages, the degradation of each inverter

stage would be added to the next, totaling 100 times the delay degradation of a single inverter at the output.

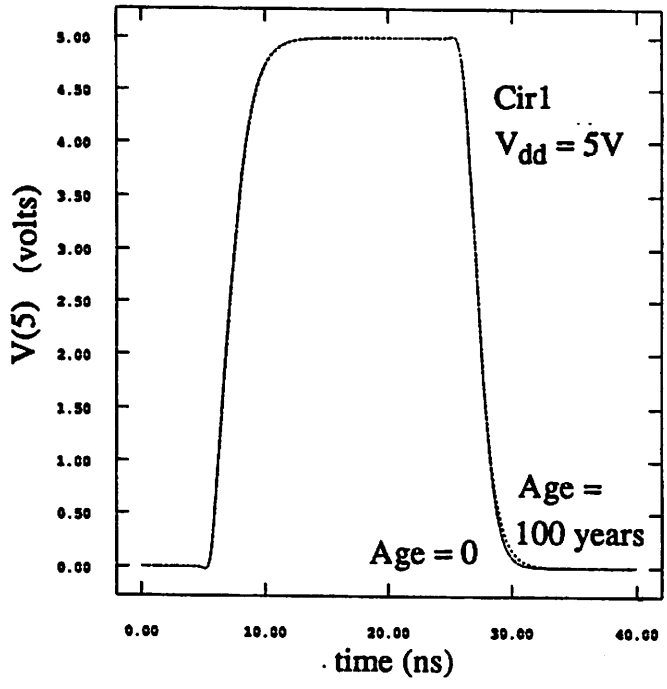


Fig. 3.26 Output voltage V(5) for the fresh circuit and for a circuit dynamically aged for 100 years. Notice the minimal amount of degradation.

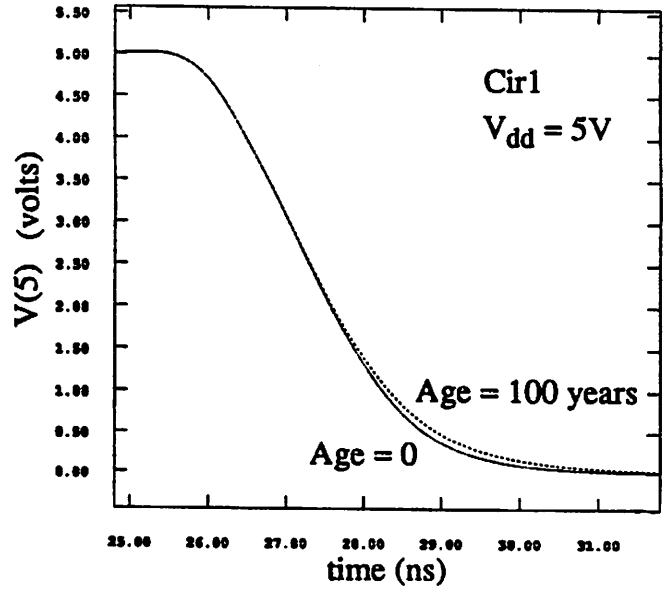


Fig. 3.27 An expanded view of the falling transient of V(5) for the fresh and aged case of Fig. 3.26.

Circuit Configuration 2 (Cir2)

Fig. 3.28 shows the circuit schematic for Cir2, and Fig. 3.29 shows the input and output voltage waveform of the register. Because of the fact that inputs to the clocked inverters change only when the clocks are low, it can be seen that the lower NMOS devices attached to the input (S1, S7, S11, S17) do not degrade, since the input switching transient occurs when the column of devices are non-conducting. It is thus the NMOS devices attached to the clock (S3, S9, S13, S19) which experience degradation. The lifetimes given in Table 3.2 are therefore for these devices only.

From Table 3.2, we can see that the device lifetimes for the devices located in the main signal path are observed as follows:

(1) The lifetimes of the clocked inverters are similar.

(2) The lifetimes of the normal inverters are similar.

(3) The NMOS devices in the clocked inverters have slightly longer lifetimes than the ones in

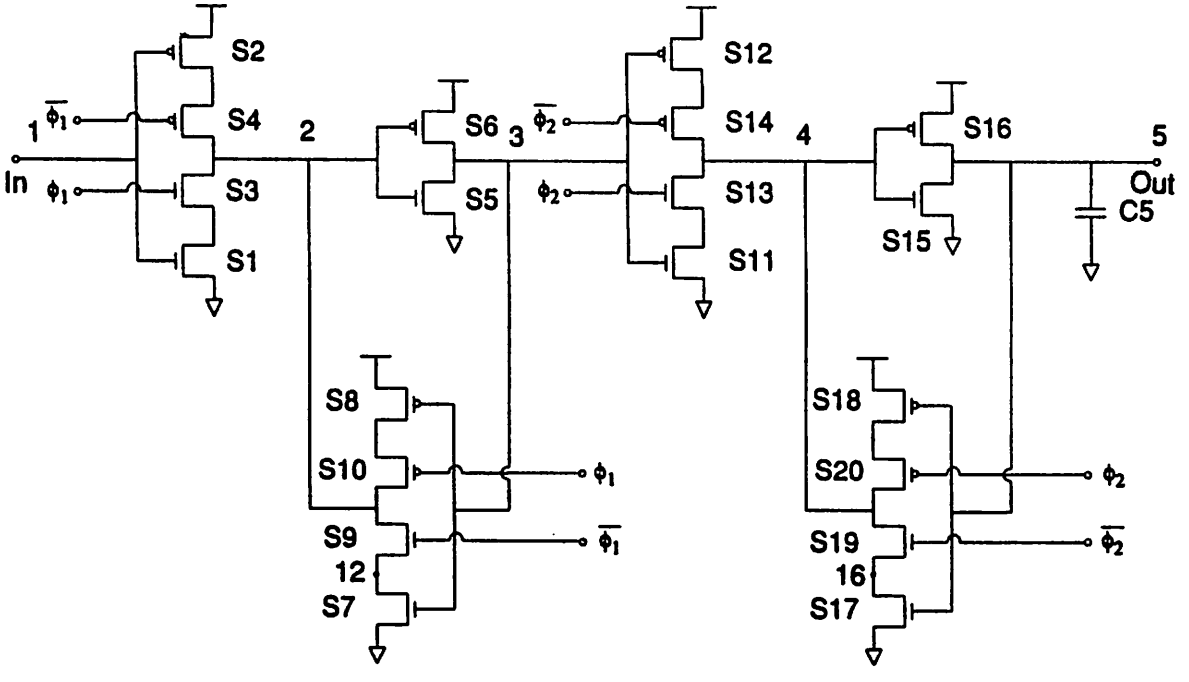


Fig. 3.28 Circuit schematic for register configuration 2 (Cir2). Devices with bubbles at the gate represent PMOS devices.

the normal inverters.

(1) and (2) are from the fact that signals applied to the main signal path are similar from stage to stage except for the last stage connected to the output load. However, we have already explained in the Cir1

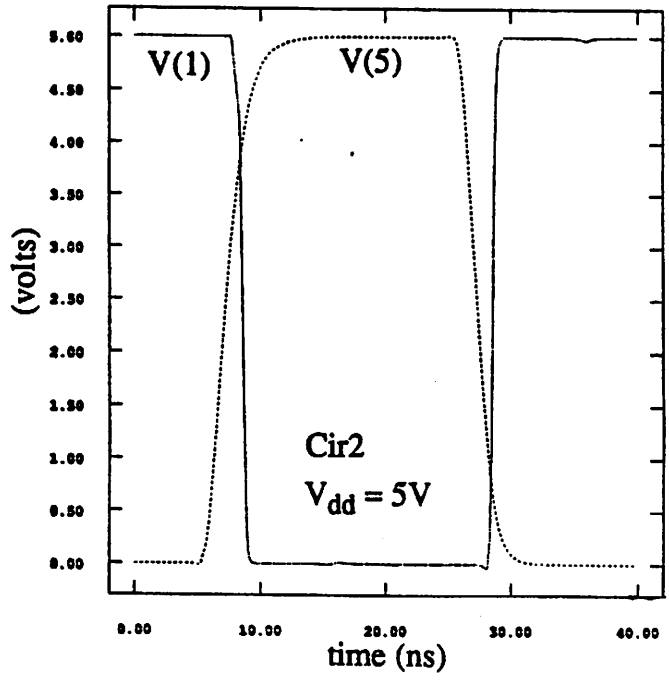


Fig. 3.29 Input voltage V(1) and output voltage V(5) for Cir2.

Cir2: τ in years at $\Delta I_{ds}/I_{ds0}$ = 10%		
Transistor	V_{dd} = 5V	V_{dd} = 5.5V
S3	134.4	5.6
S5	96.9	6.8
S9	32.2	2.1
S13	164.7	8.1
S15	83.9	4.0
S19	188.1	7.2

Table 3.2 Device lifetimes τ for selected NMOSFET's of Cir2. Other NMOS devices had τ = infinity.

case how the slow transient of the output has very little affect on the degradation of the last stage. (3) is particular only to this case; as mentioned in Cir1, because it is the rise/fall time of the input voltage waveform that has the greatest effect on degradation, (3) will depend directly on the relative rise/fall time of the clock as compared to the rise/fall time of the nodes in the main signal path.

The interesting point to note in this particular circuit is the difference in lifetimes of the two feedback clocked inverters containing S9 and S19. Here S9 is seen to have a much shorter lifetime than S19. The reason for this contrast can be shown in Figs. 3.30 - 3.34. Note that the source of S9 (Fig. 3.30, V(12), $t \cong 16$ ns) is induced to -0.5 volts from capacitive coupling of the input (V(3)) as it makes a fast high-to-low transition when ϕ_1 enables the input voltage to propagate through. Since S7 and S9 are off (V(3) = 0 and inverse of $\phi_1 = 0$), V(12) stays negative until S9 is turned on by the inverse of ϕ_1. Because V(2) is already high, S9 experiences a drain voltage maximum of 5.5 volts (Fig. 3.31), enhancing degradation. In contrast, by looking at the source voltage of S19 (Fig. 3.32, V(16), $t \cong 30$ ns), the slower high-to-low transition of the heavily-loaded output couples less of the negatively-going transient to the

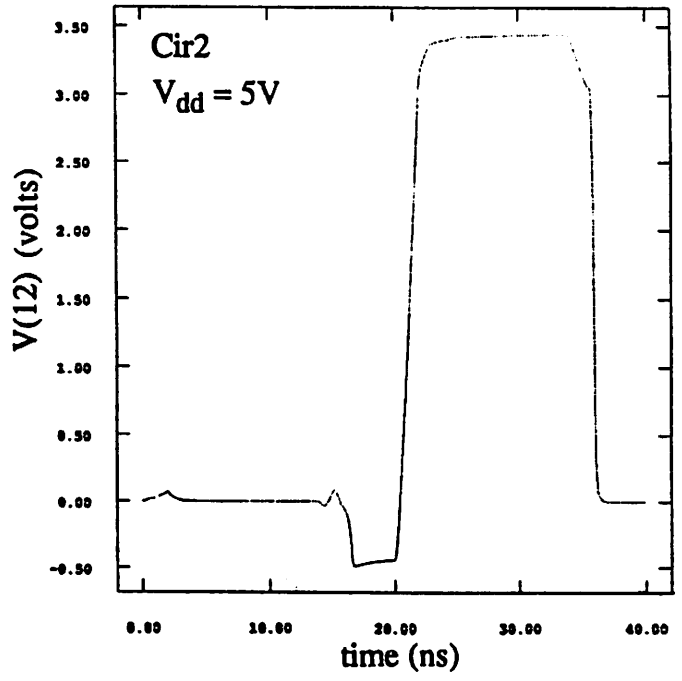


Fig. 3.30 Source voltage of device S9 in the clocked inverter. Notice that the negative transient of the input V(3) at around t = 16ns causes this voltage to go negative by approximately 0.5 volts from capacitive coupling.

source of S19. Thus, when S19 turns on, it sees a maximum drain voltage of roughly 5.25 V (Fig. 3.33), less than in the S9 case and therefore less degradation occurs. This again is verified by the plot of $I_{ds}(I_{sub}/I_{ds})^m$ of both S9 and S19 in Fig. 3.34.

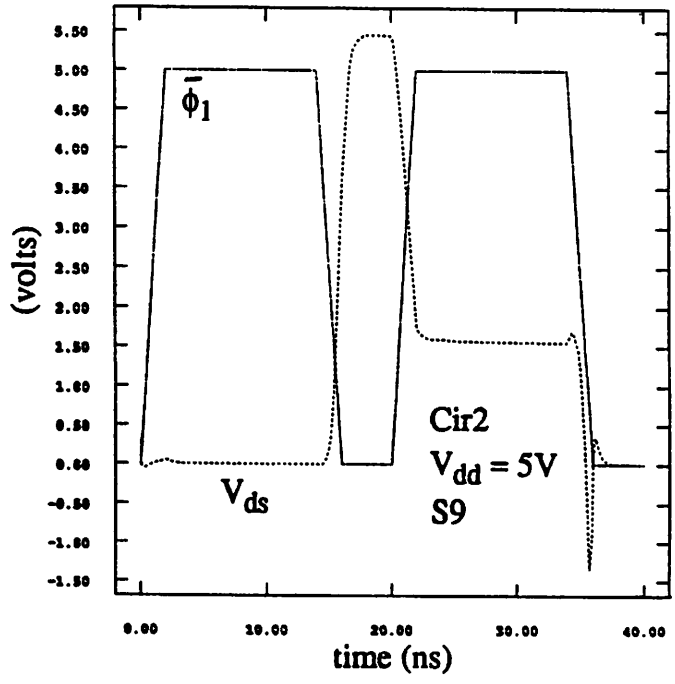


Fig. 3.31 V_{gs} (= inverse of ϕ_1) and V_{ds} of device S9 showing that $V_{dsmax} \cong 5.5V$, and thus enhancing degradation.

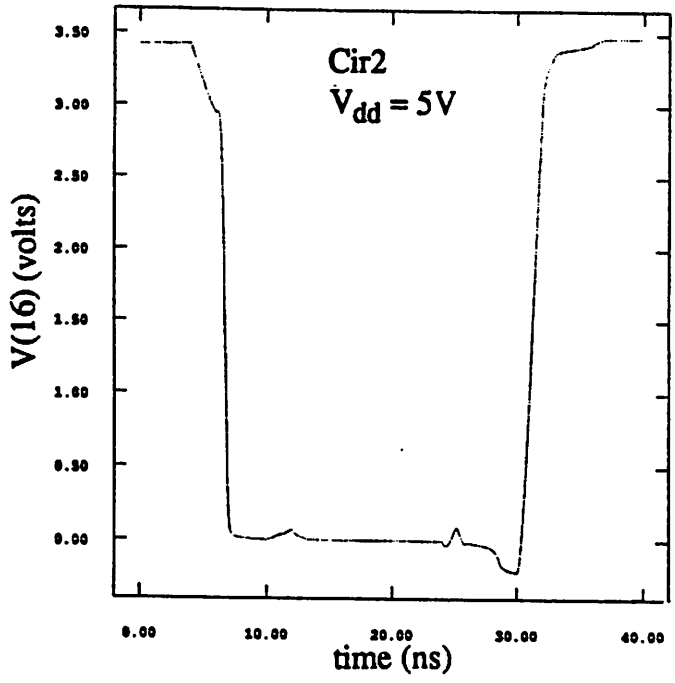


Fig. 3.32 Source voltage of device S19 in the clocked inverter attached to the output. Notice that the slower input voltage transient V(5) at t ≅ 28ns reduces the amount of coupling that occurs to this source node.

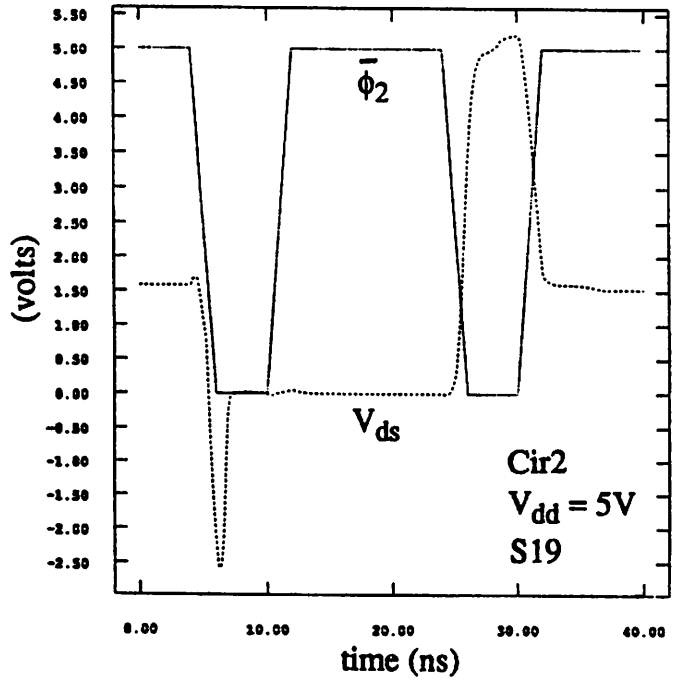


Fig. 3.33 V_{gs} (= inverse of ϕ_2) and V_{ds} of device S19 showing that $V_{dsmax} \cong$ 5.25V, and thus showing less than S9.

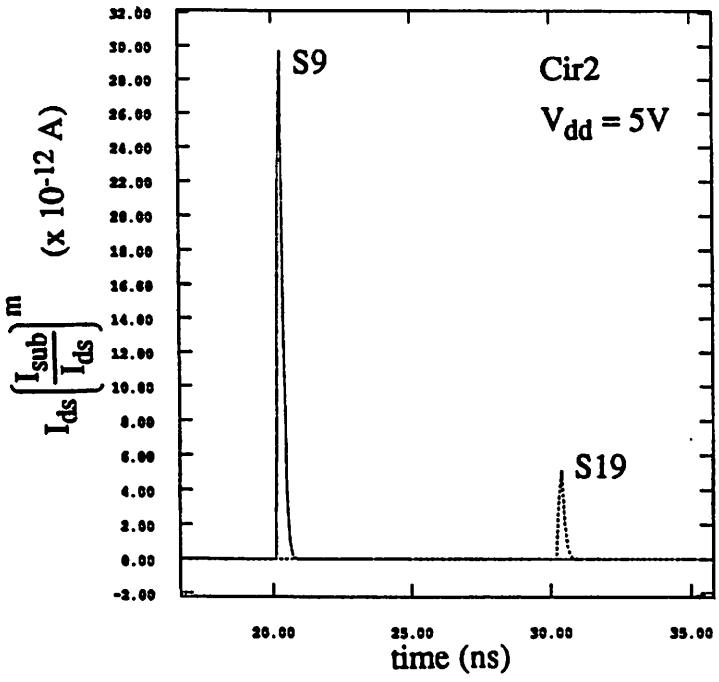


Fig. 3.34 Plots of $I_{ds}(I_{sub}/I_{ds})^m$ for devices S9 and S19 verifying the lower degradation level of S19.

3.7.4 Summary

Additional insight has been found from doing device lifetime calculations for two different shift register configurations, one based on transmission gate clocking, while the other based on clocked inverters. Some surprising results were seen but logically explained using voltage waveforms from circuit simulation. The overall conclusion is that these registers are very robust to hot-carrier degradation, especially because the speed of operation is limited by the clock cycle rather than the inherent switching speed of the circuit. The performance degradation is expected to have an effect, however, in the maximum clock speed that can be used in these circuits.

3.7.5 References

[Lee88] Peter M. Lee, Mary M. Kuo, Koichi Seki, Ping Ko, and Chenming Hu, "Circuit Aging Simulation (CAS)," *International Electron Devices Meeting Technical Digest*, pp. 134-137, December 1988.

3.7.6 Appendix

The following pages contain the SPICE input decks used to simulate circuits Cir1 and Cir2.

Circuit Configuration 1 (Cir1):

```
CMOS REGISTER - CONFIGURATION 1
* Circuit ID: Cir1
* Input transmission gate of register 1
s1 1 60 2 0 PC1_nm1_du1 w=20u l=1.4u
s2 2 65 1 50 PC2_pm1_du1 w=20u l=1.4u
* Register 1 inverter
s3 3 2 0 0 PC1_nm1_du1 w=20u l=1.4u
s4 3 2 50 50 PC2_pm1_du1 w=40u l=1.4u
* Feedback inverter of register 1
s5 21 3 0 0 PC1_nm1_du1 w=20u l=1.4u
s6 21 3 50 50 PC2_pm1_du1 w=40u l=1.4u
* Feedback transmission gate of register 1
s21 2 65 21 0 PC1_nm1_du1 w=20u l=1.4u
s22 21 60 2 50 PC2_pm1_du1 w=20u l=1.4u
* Input transmission gate of register 2
s7 3 70 4 0 PC1_nm1_du1 w=20u l=1.4u
s8 4 75 3 50 PC2_pm1_du1 w=20u l=1.4u
* Register 2 inverter
s9 5 4 0 0 PC1_nm1_du1 w=20u l=1.4u
s10 5 4 50 50 PC2_pm1_du1 w=40u l=1.4u
* Feedback inverter of register 2
s11 22 5 0 0 PC1_nm1_du1 w=20u l=1.4u
s12 22 5 50 50 PC2_pm1_du1 w=40u l=1.4u
* Feedback transmission gate of register 2
s23 4 75 22 0 PC1_nm1_du1 w=20u l=1.4u
s24 22 70 4 50 PC2_pm1_du1 w=20u l=1.4u
*
* Internal node capacitances
c2 2 0 0.1pF
c3 3 0 0.1pF
c4 4 0 0.1pF
c21 21 0 0.1pF
c22 22 0 0.1pF
*
* Output capacitive loading
c5 5 0 2pF
*
* Power Supply
vdd 50 0 dc 5
*
* Input voltage: The input voltage waveform is conditioned by
* a two-stage inverter chain.
vin 100 0 pwl(0,5,7ns,5,9ns,0,27ns,0,29ns,5)
s101 101 100 0 0 PC1_nm1_du1 w=20u l=1.4u
```

```
s102 101 100 50 50 PC2_pm1_du1 w=40u l=1.4u
s103 1 101 0 0 PC1_nm1_du1 w=20u l=1.4u
s104 1 101 50 50 PC2_pm1_du1 w=40u l=1.4u
c101 101 0 0.05pF
c1 1 0 0.1pF
*
* Clocks (Two phase):
vphi1 60 0 pwl(0,5,2ns,0,14ns,0,16ns,5,20ns,5,22ns,0,34ns,0,36ns,5)
vphi1bar 65 0 pwl(0,0,2ns,5,14ns,5,16ns,0,20ns,0,22ns,5,34ns,5,36ns,0)
vphi2 70 0  pwl(0,0,4ns,0,6ns,5,10ns,5,12ns,0,24ns,0,26ns,5,30ns,5,32ns,0)
vphi2bar 75 0  pwl(0,5,4ns,5,6ns,0,10ns,0,12ns,5,24ns,5,26ns,0,30ns,0,32ns,5)
*
* Calculation control
*
.options method=gear
.nodeset v(1)=5 v(2)=5 v(3)=0 v(4)=5 v(5)=0
+ v(21)=5 v(22)=5
+ v(50)=5
+ v(60)=5 v(65)=0
+ v(70)=0 v(75)=5
*
* Process Files
*
.process PC1 filename=MLG0OUT
.process PC2 filename=GEPM1OUT
*
* Output control
*
.width out=80
.options limpts=1000
.tran 0.1ns 40ns
.print tran v(1)
.print tran v(5)
*
* Special CAS commands
*
.isubwidth=80
.deltaid 0.1
.printisub s1 s3 s9 s11
.age 100yrs
.ageproc PC1 filenames=MLG0OUT, MLG1OUT, MLG3OUT, MLG5OUT, MLG7OUT
.end
```

Circuit Configuration 2 (Cir2):

```
CMOS REGISTER - CONFIGURATION 2
* Circuit ID: Cir2
* Input clocked inverter of register 1
s1 10 1 0 0 PC1_nm1_du1 w=20u l=1.4u
s3 2 60 10 0 PC1_nm1_du1 w=20u l=1.4u
s2 11 1 50 50 PC2_pm1_du1 w=40u l=1.4u
s4 2 65 11 50 PC2_pm1_du1 w=40u l=1.4u
* Register 1 inverter
s5 3 2 0 0 PC1_nm1_du1 w=20u l=1.4u
```

```
s6 3 2 50 50 PC2_pm1_du1 w=40u l=1.4u
* Feedback clocked inverter of register 1
s7 12 3 0 0 PC1_nm1_du1 w=20u l=1.4u
s9 2 65 12 0 PC1_nm1_du1 w=20u l=1.4u
s8 13 3 50 50 PC2_pm1_du1 w=40u l=1.4u
s10 2 60 13 50 PC2_pm1_du1 w=40u l=1.4u
* Input clocked inverter of register 2
s11 14 3 0 0 PC1_nm1_du1 w=20u l=1.4u
s13 4 70 14 0 PC1_nm1_du1 w=20u l=1.4u
s12 15 3 50 50 PC2_pm1_du1 w=40u l=1.4u
s14 4 75 15 50 PC2_pm1_du1 w=40u l=1.4u
* Register 2 inverter
s15 5 4 0 0 PC1_nm1_du1 w=20u l=1.4u
s16 5 4 50 50 PC2_pm1_du1 w=40u l=1.4u
* Feedback clocked inverter of register 2
s17 16 5 0 0 PC1_nm1_du1 w=20u l=1.4u
s19 4 75 16 0 PC1_nm1_du1 w=20u l=1.4u
s18 17 5 50 50 PC2_pm1_du1 w=40u l=1.4u
s20 4 70 17 50 PC2_pm1_du1 w=40u l=1.4u
*
* Internal node capacitances
c2 2 0 0.1pF
c3 3 0 0.05pF
c4 4 0 0.1pF
c10 10 0 0.05pF
c11 11 50 0.05pF
c12 12 0 0.05pF
c13 13 50 0.05pF
c14 14 0 0.05pF
c15 15 50 0.05pF
c16 16 0 0.05pF
c17 17 50 0.05pF
*
* Output capacitive loading
c5 5 0 2pF
*
* Power Supply
vdd 50 0 dc 5
*
* Input voltage: The input voltage waveform is conditioned by
* a two-stage inverter chain.
vin 100 0 pwl(0,5,7ns,5,9ns,0,27ns,0,29ns,5)
s101 101 100 0 0 PC1_nm1_du1 w=20u l=1.4u
s102 101 100 50 50 PC2_pm1_du1 w=40u l=1.4u
s103 1 101 0 0 PC1_nm1_du1 w=20u l=1.4u
s104 1 101 50 50 PC2_pm1_du1 w=40u l=1.4u
c101 101 0 0.05pF
c1 1 0 0.05pF
*
* Clocks (Two phase):
vphi1 60 0 pwl(0,5,2ns,0,14ns,0,16ns,5,20ns,5,22ns,0,34ns,0,36ns,5)
vphi1bar 65 0 pwl(0,0,2ns,5,14ns,5,16ns,0,20ns,0,22ns,5,34ns,5,36ns,0)
vphi2 70 0 pwl(0,0,4ns,0,6ns,5,10ns,5,12ns,0,24ns,0,26ns,5,30ns,5,32ns,0)
vphi2bar 75 0 pwl(0,5,4ns,5,6ns,0,10ns,0,12ns,5,24ns,5,26ns,0,30ns,0,32ns,5)
```

```
*
* Calculation control
*
.options node method=gear vntol=1e-3 abstol=1e-6 reltol=0.01
.nodeset v(1)=5 v(2)=0 v(3)=5 v(4)=5 v(5)=0
+ v(10)=0 v(11)=5 v(12)=0 v(13)=5 v(14)=0 v(15)=0
+ v(16)=5 v(17)=5
+ v(50)=5
+ v(60)=5 v(65)=0
+ v(70)=0 v(75)=5
*
* Output control
*
.width out=80
.options limpts=1000
.tran 0.1ns 40ns
.print tran v(1)
.print tran v(5)
*
* Process Files
*
.process PC1 filename=MLG0OUT
.process PC2 filename=GEPM1OUT
*
* Special CAS commands
*
.plotisub s9 s19
.deltaid 0.1
.isubwidth=80
.end
```

3.8 Experimental Results

3.8.1 Introduction

This section presents experimental results obtained by accelerated hot-carrier stressing for two circuit examples, a CMOS ring oscillator and a NMOS operational amplifier. In each case, the BSIM1 model was used to simulate individual devices.

3.8.2 Case 1: CMOS Ring Oscillator[1]

In this first example, hot-carrier performance degradation of special 125-stage CMOS ring oscillators were studied under accelerated stressing conditions. The ring oscillators were fabricated in a 0.5μm technology that was modified specifically to enhance hot-carrier degradation of the NMOSFET devices. Two modifications were made: 1) the PMOSFET's were fabricated as LDD devices while the NMOSFET's were processed as non-LDD devices to decrease the effect of PMOSFET degradation; and 2) nitride passivation with higher than usual hydrogen content[2] was directly deposited over the first metal layer. Circuit stressing was performed at V_{dd} = 5.9V, while frequency degradation was monitored at the lower supply voltages of V_{dd} = 3.3V, 4V, and 4.5V. The performance criteria used for comparison was percentage frequency degradation $\Delta f/f_0$. Two capacitive loading configurations were also measured to see how loading effects hot-carrier degradation.

BSIM1 model parameter extractions were done from individual devices on the same wafer as the ring oscillator circuits. Because of the limitation of the BSIM1 model in extracting parameters from devices shorter than 1μm in channel length using the normal extractor [Jen87], the SUXES (Stanford University eXtractor of modEl parameterS) optimizer [Dog83] was used to extract all model parameters.

---

[1]Circuit fabrication and stressing measurements done at the Advanced Products Research and Development Laboratory, Motorola, Inc., Austin, Texas.

[2]High hydrogen content is know to cause bonds with the SiO_2 at the Si - SiO_2 interace . These bonds can break during hot-carrier activity, causing the creation of interface traps and enhancing the degradation of the MOSFET [Hu85].

Device stressing and measurements were performed using an automated measurement system integrated with other extraction tools such as SUXES [Tsu90].

In the I_{ds} model parameter extractions, a three-pass approach was used. First, I_{ds} versus V_{gs} data in the linear region (V_{ds} = 0.05V) for various V_{bs} values was used to extract the linear-region parameters VFB, K1, K2, U0, X2U0, MUZ, and X2MZ. Limits on the parameters K1, U0, and MUZ were imposed to prevent them from becoming less than zero. Next, I_{ds} versus V_{ds} data (with V_{gs} and V_{bs} as a parameter) was used to extract the V_{ds}-dependent parameters and saturation region parameters ETA, X2E, X3E, U1, X2U1, X3U1, MUS, X2MS, and X3MS. A limit was placed on U1 and MUS so that is also remained greater than zero. Lastly, in the third pass, the parameters VFB, U0, U1, MUS, MUZ, and ETA were re-extracted from the I_{ds} versus V_{ds} data (with V_{gs} as a parameter) for V_{bs} = 0. This final pass is necessary to accurately model the transition region between the linear and saturation region. This region exhibits the greatest drain current degradation (in magnitude, not percentage), and is thus crucial in correctly simulating degradation in inverter-based circuits (see Chapter 4). Also note that in this final pass, only the V_{bs} = 0V case was used since the NMOSFET of a CMOS inverter does not experience any body bias. Using this three-pass method with SUXES, a fairly accurate parameter set can be extracted, even down to the half-micron regime. Figs. 3.35 compares the measured and simulated I_{ds} - V_{ds} curves for the NMOSFET.

SUXES was also used to extract I_{ds} model parameters for the aged (stressed) devices. To obtain consistent shifts in the parameters, only the major parameters were allowed to vary for each stressed device, while all second-order parameters were held constant at values extracted from the fresh device. After careful analysis, the best fit to measurement data that provided consistent parameter shifts was achieved by performing only the third pass of the SUXES extraction for each stressed device so that only the BSIM1 parameters VFB, U0, U1, MUS, MUZ, and ETA were allowed to vary for the aged model parameter files. For this example, 11 different model parameter sets were extracted after 10 different stress times from the same device. Fig. 3.36 shows the good I_{ds} - V_{ds} fit when using this methodology for a device stressed at V_{gs} = 2.25V and V_{ds} = 5.5V for 100,000 seconds. Figs. 3.37 and 3.38 show the

parameter shifts versus stress time for the flat-band voltage VFB and the linear region mobility MUZ. Because of the irregular form of the plot, interpolation in the linear-log domain was chosen to be the method to calculate the aged device parameters.

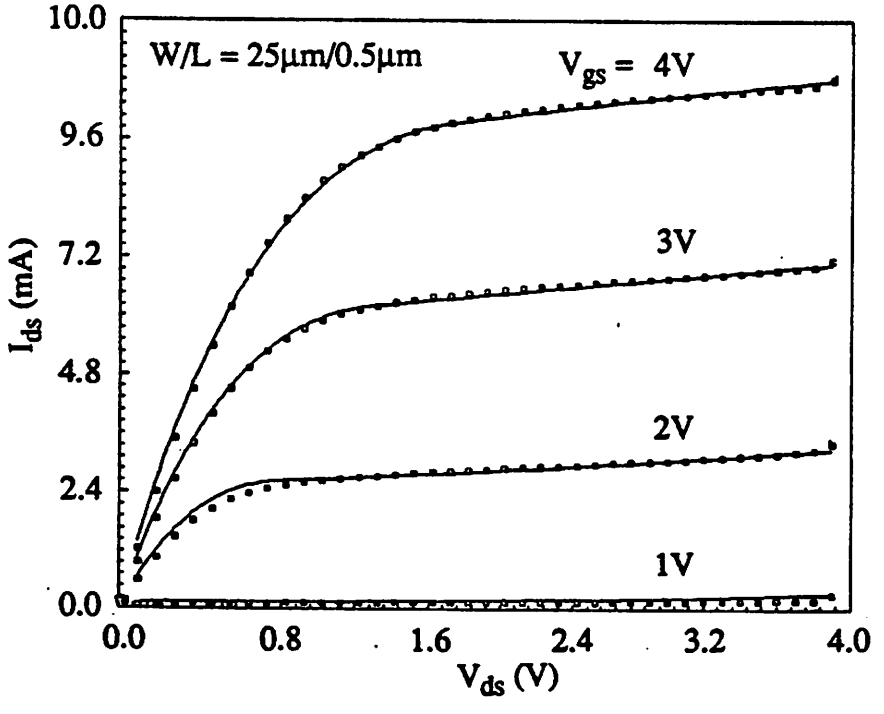


Fig. 3.35 Measured and SUXES-simulated NMOSFET I_{ds} versus V_{ds} curves for V_{gs} = 1, 2, 3, 4, and 5V for the fresh device.

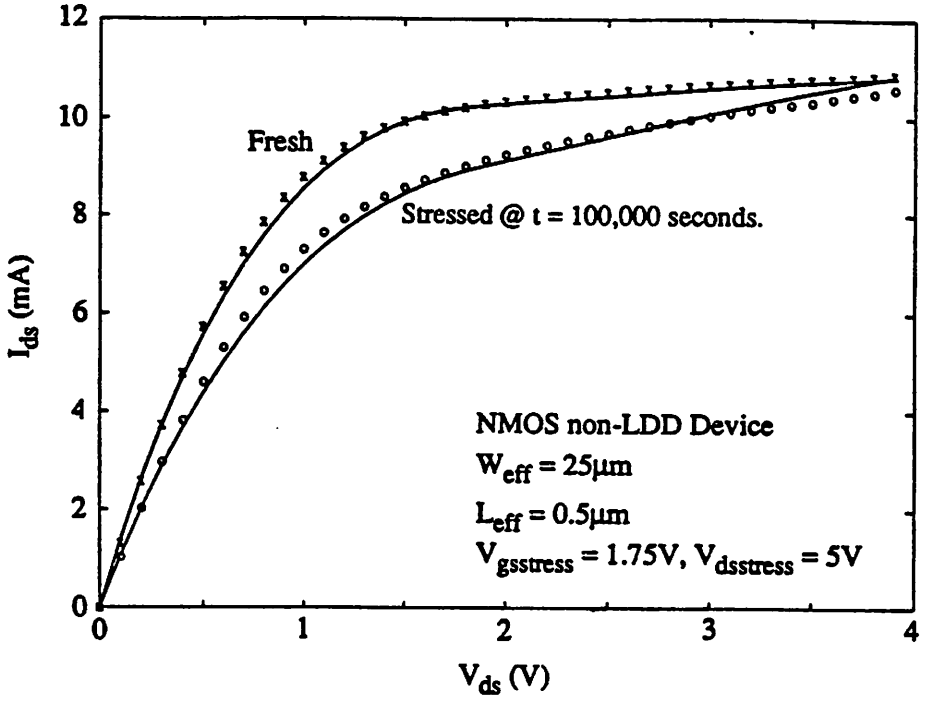


Fig. 3.36 Measured and simulated I_{ds} versus V_{ds} curves for a fresh and 100,000-second-stressed device after BSIM1 parameter extraction using SUXES and the methodology described in the text. V_{gs} = 4V.

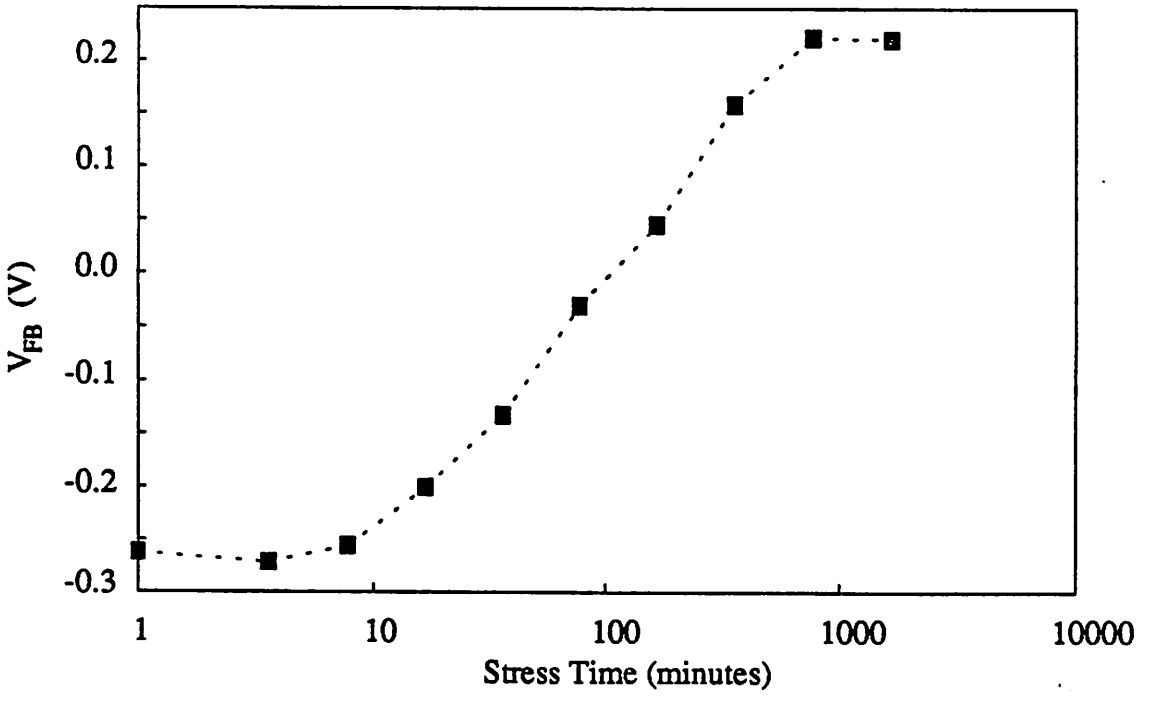


Fig. 3.37 The BSIM1 parameter VFB versus stress time plotted in log-log format.

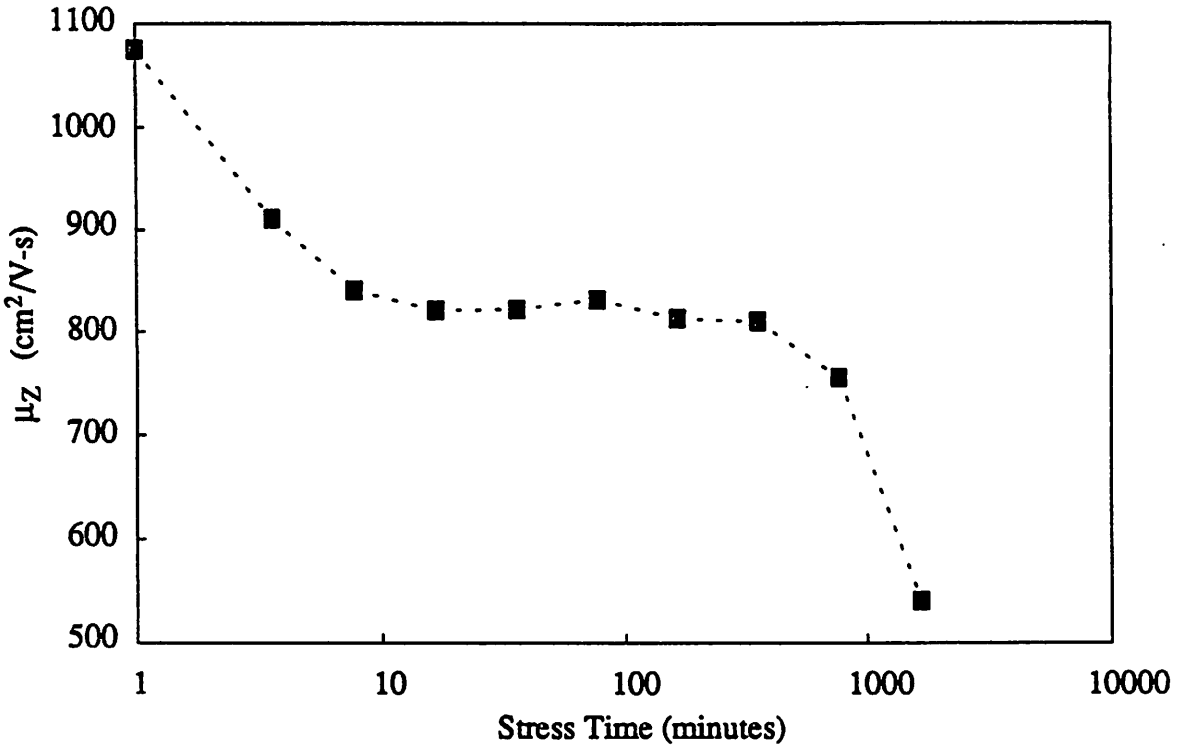


Fig. 3.38 The BSIM1 parameter MUZ versus stress time plotted in log-log format.

Because substrate current parameter extraction had not yet been implemented in SUXES, I_{sub} parameters were obtained manually. Fig. 3.39 shows the fairly good fit of I_{sub} versus V_{gs} obtainable for V_{ds} = 3.05V and 4.05V.

Finally, the degradation parameters H, m, and n were extracted from a group of devices. In this specific case, all stressing was done at V_{gd} = 2.25V so that no V_{gd}-sensitivity terms were extracted. The stressing voltages ranged from V_{gs} = 1.5V to 2.25V and V_{ds} = 4.75V to 5.5V. The parameter n was calculated from an average of the n values obtained from each device, and the parameters H and m were extracted from the $\tau I_{ds}/W$ versus I_{sub}/I_{ds} plot (see Section 3.2).

Fig. 3.40 shows the percentage frequency degradation $\Delta f/f_0$ versus stress time for a nitride-passivated case using the I_{sub} parameters simulated in Fig. 3.39. Stressing power supply voltage was V_{dd} = 5.9V, and the frequency degradation was measured at V_{dd} = 3.3V, 4V, and 4.5V. Note that both CAS and the data show larger percentage degradation for lower V_{dd}. This is because, in general, measuring at

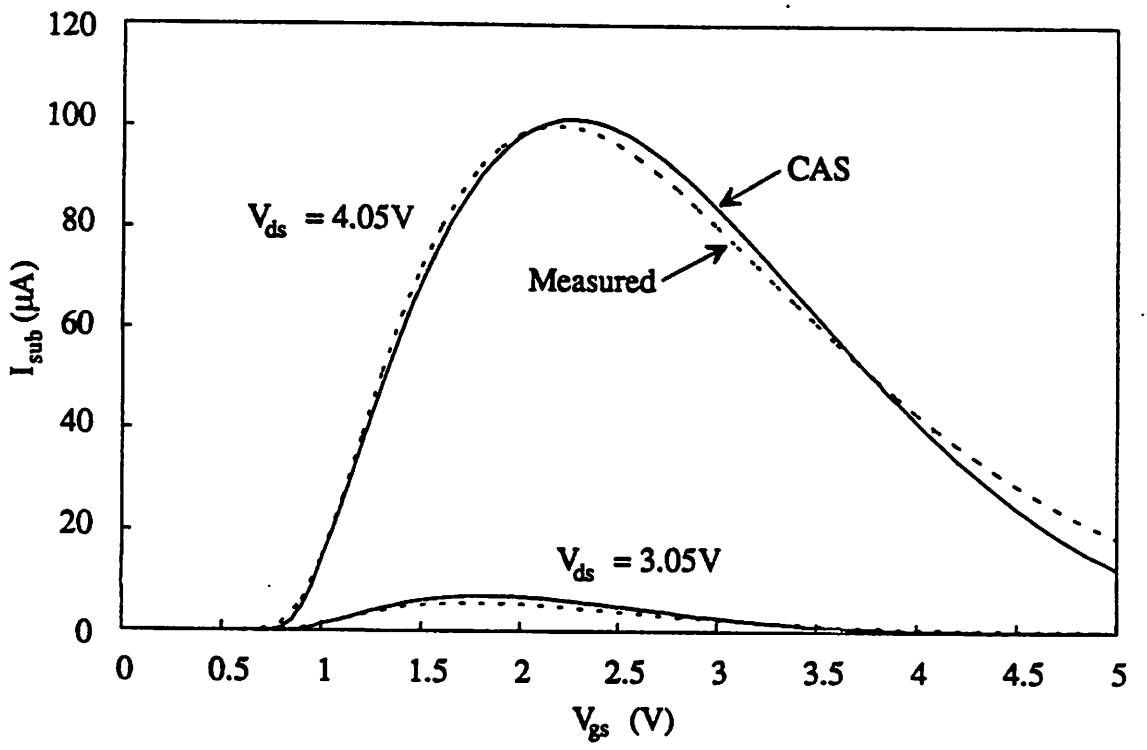


Fig. 3.39 Measured and CAS-simulated I_{sub} versus V_{gs} fit for V_{ds} = 3.05V and 4.05V after manual adjustment of the parameters.

lower V_{dd} gives larger degradation because on the device level a higher drain current degradation (percentage-wise) exists at the lower voltages.

As can be seen, the general trend is predicted, but CAS overestimates the measured data by a fair amount for the longer stressing times. After careful analysis, it was found that in this case the CAS-simulated results were very sensitive to the I_{sub} current because of a high value of m extracted (m = 8.18). Recall that the Age is proportional to $I_{ds}(I_{sub}/I_{ds})^m$. Thus, any change in I_{sub} will be magnified by a high value of m. Fig. 3.41 shows the sensitivity of the frequency degradation predicted by CAS to percentage I_{sub} difference where frequency degradation is taken at T_{stress} = 400 minutes for V_{dd} = 4V (last stress point in Fig. 3.40) and I_{sub} is taken at V_{gs} = 1V and V_{ds} = 4V (bias point where $I_{ds}(I_{sub}/I_{ds})^m$ is a maximum). From the graph, it can be seen that roughly 10% change in I_{sub} causes 5% change in the frequency degradation. In view of these results, because I_{sub} of the circuit devices can rarely be measured, and there is no guarantee that the I_{sub} of an individual device (even on the same die) will

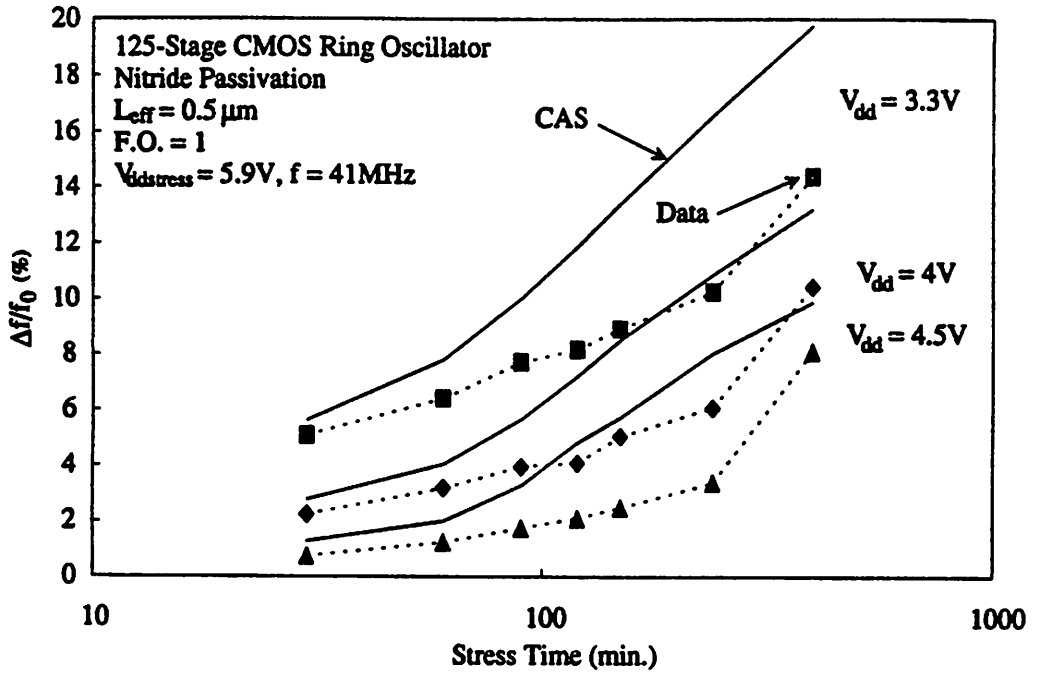


Fig. 3.40 Measured and uncalibrated CAS-simulated percentage change in frequency ($\Delta f/f_0$) versus stress time at different V_{dd} values for the nitride-passivated oscillator. Fan-out is equal to one.

match that of the circuit devices, a better way to calibrate CAS would be to match the measured and simulated frequency degradation at one data point on Fig. 3.40 by modifying the I_{sub} parameters.

For this case, the data point for T_{stress} = 400 minutes and V_{dd} = 4V for the fanout = 1 ring oscillator of Fig. 3.40 was chosen as the calibration point. For simplification, only the parameter l_{c0} was changed so that none of the bias-dependencies of l_c would be affected. Fig. 3.42 shows the measured data of Fig. 3.40 with the calibrated CAS simulation results. Much better agreement is obtained. Note how CAS can predict the results quite well for other stress times and V_{dd} although only one point was used for calibration.

Fig. 3.43 shows the nitride-passivated loaded ring oscillator (fanout of 3) at the same measuring and stressing V_{dd} as Fig. 3.42. The I_{sub} parameters are unchanged from that of the fanout = 1 case of Fig. 3.42. Note how CAS can again closely predict the measured results. This suggests that calibration need be done only once for a particular process. Also note that the frequency degradation is almost the same as that of the unloaded case in Fig. 3.42. Although higher capacitive loadings generally increases

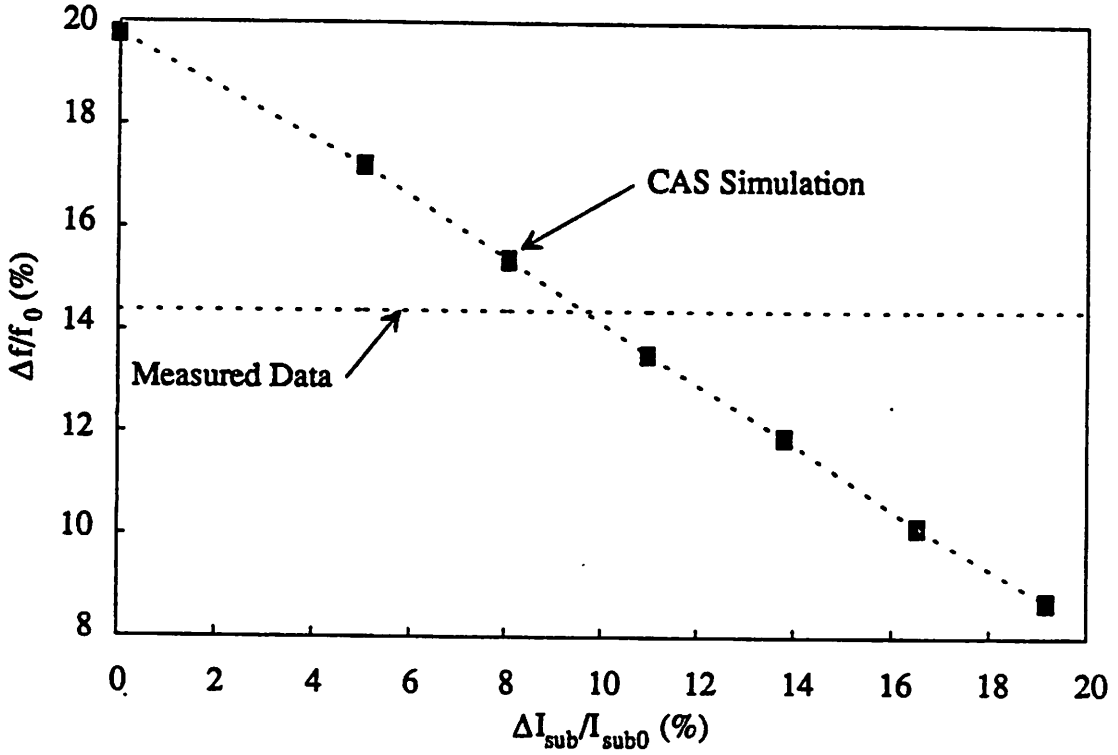


Fig. 3.41 Frequency degradation $\Delta f/f_0$ of the ring oscillator of Fig. 3.40 at T_{stress} = 400 minutes and V_{dd} = 4V versus percentage I_{sub} difference $\Delta I_{sub}/I_{sub}$ taken at V_{gs} = 1V and V_{ds} = 4V (maximum of $I_{ds}(I_{sub}/I_{ds})^m$).

degradation, the loaded ring oscillator oscillates at a lower frequency, thus reducing the effective stress time of each stage. The two effects seem to counterbalance each other for this particular circuit.

Finally, Fig. 3.44 shows the frequency degradation of the nitride-passivated ring oscillator of Fig. 3.42 compared with that of the PSG-passivated ring oscillator. Both are measured at V_{dd} = 3.3V, and have a fanout = 1. A separate calibration was done for the PSG case, since a different process was used. As expected, the nitride-passivated oscillator degraded much more severely than the PSG-passivated one, by about an order of magnitude in this case. This result correlates with previous studies at the device level where hydrogen involved in the deposition of the passivation layer was found to aggravate degradation [Mit86]. CAS simulation results again provide good agreement with the experimental trends. From all the simulation results, it can be seen that properly implemented quasi-static calculations do not underestimate the AC degradation for the "good" AC waveform case seen in inverter-based circuits (see Section 3.2 for the definitions of the "good" and "bad" waveforms).

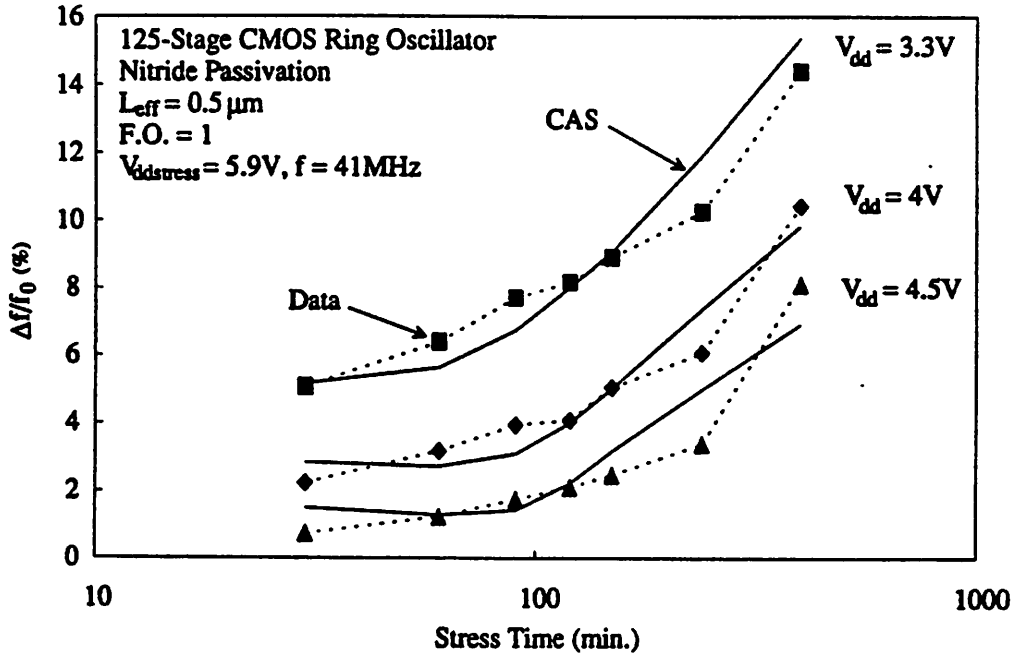


Fig. 3.42 Measured and calibrated CAS-simulated percentage change in frequency ($\Delta f/f_0$) versus stress time at different V_{dd} values for the nitride-passivated oscillator. Fan-out is equal to one.

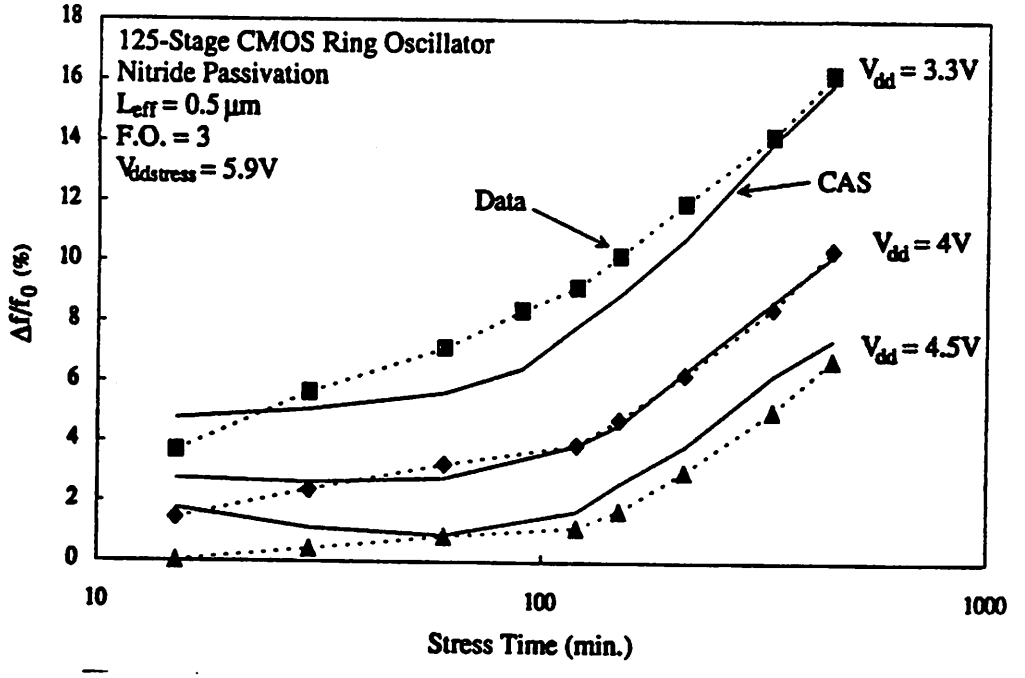


Fig. 3.43 Measured and calibrated CAS-simulated percentage change in frequency ($\Delta f/f_0$) versus stress time at different V_{dd} values for the nitride-passivated oscillator. Fan-out is equal to three.

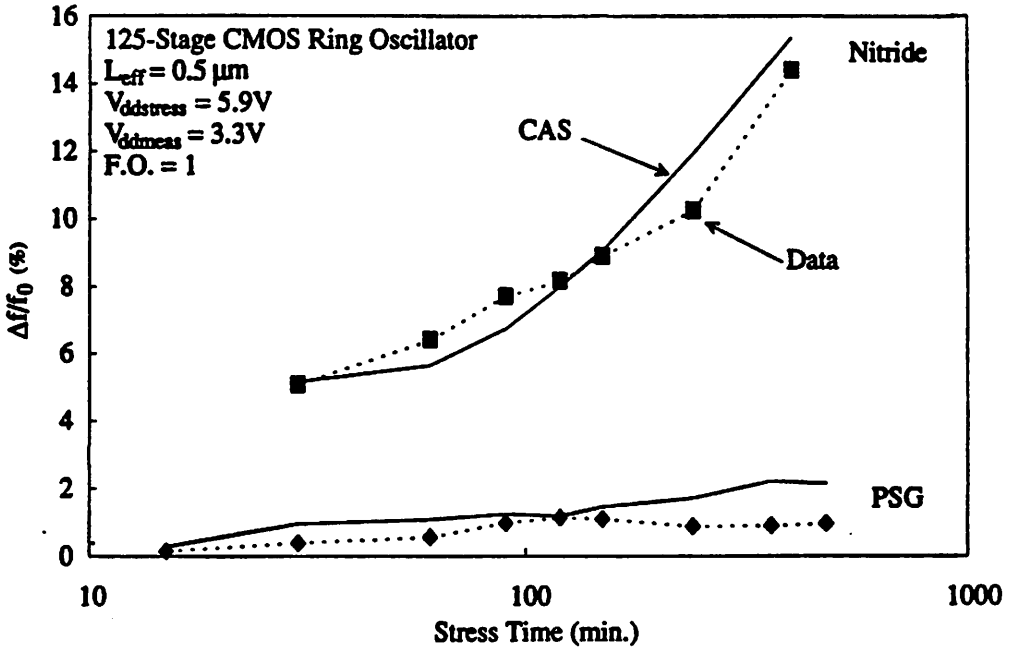


Fig. 3.44 Measured and calibrated CAS-simulated percentage change in frequency ($\Delta f/f_0$) versus stress time for the hot-carrier-enhanced nitride- and PSG-passivated 125-stage ring oscillators measured at V_{dd} = 3.3V.

3.8.3 Case 2: NMOS Operational Amplifier

In this second case, a NMOS operational amplifier fabricated in the Microfabrication Laboratory was used as an analog circuit example. Circuit configuration shown in Fig. 3.45 was taken from [Tsi76] and fabricated in a 2µm enhancement-mode non-LDD NMOS technology with t_{ox} = 200A. The circuit was stressed under the following conditions: V_{dd} = 10V, V_{inv} = 5V, $V_{non\text{-}inv}$ = 0V, and V_{ss} = 0V.

Fig. 3.46 shows the agetable generated by CAS simulation after 10 minutes of stress, with the transistors ordered by greatest to least age. From inspecting this table, it can be seen that the transistors M21 and M13 have by far the largest age values compared to the rest. These two transistors are circled in Fig. 3.45. From the circuit schematic, it can be seen that M21 and M13 belong to the low-gain output stage portion of the amplifier. Thus, both CAS and measurement data show no change in the overall gain of the amplifier; however, the input offset voltage is seen to change with stress time. Fig. 3.47 shows

both experimental and CAS-simulated results showing this effect. Although the actual magnitude of the voltage shift is somewhat off (believed to be due to an inaccurate m value), the general trend is predicted quite accurately.

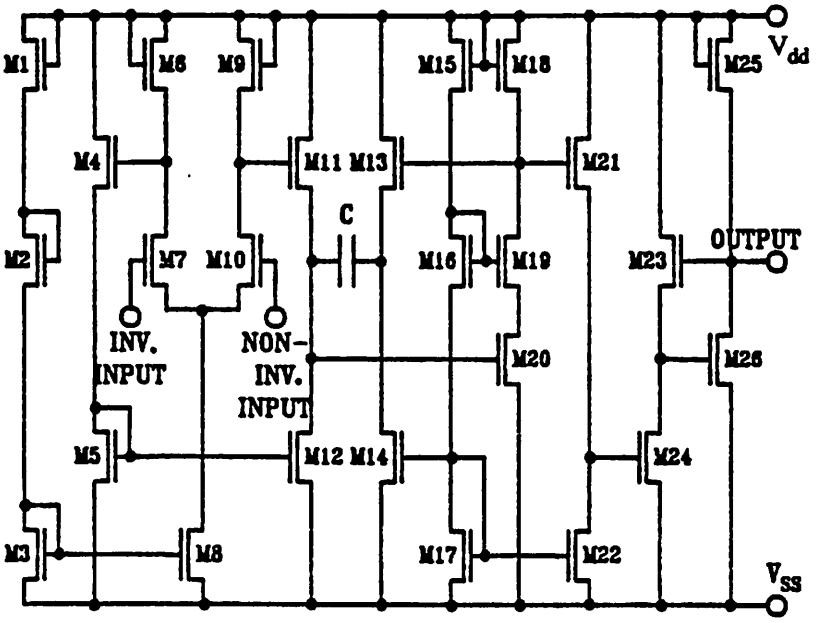


Fig. 3.45 Circuit configuration of the NMOS operational amplifier stressed in this analog example. Circuit was taken from [Tsi76]. CAS simulations show that the transistors M21 and M13 circled in the schematic suffer the most degradation.

Device Name	Process Name	Age	Vds(V)	Vgs(V)
M21	PC2 nm1 du1	1.003760e+00	9.989	0.335
M13	PC2 nm1 du1	8.384320e-01	9.978	0.324
M23	PC1 nm4 du1	5.207166e-04	8.457	0.955
M4	PC1 nm5 du1	1.177822e-09	9.062	6.688
M11	PC1 nm5 du1	1.061775e-09	8.517	6.793
M8	PC1 nm4 du1	2.912096e-11	3.849	1.090
M7	PC2 nm1 du1	1.394939e-11	3.777	1.151
M25	PC1 nm5 du1	4.040922e-14	7.502	7.502
M26	PC1 nm4 du1	9.684229e-25	2.498	1.543

Fig. 3.46 Agetable generated by CAS after 10 minutes of stress. Transistors are ordered by greatest to least age values. Notice that transistors M21 and M13 have by far the largest age values.

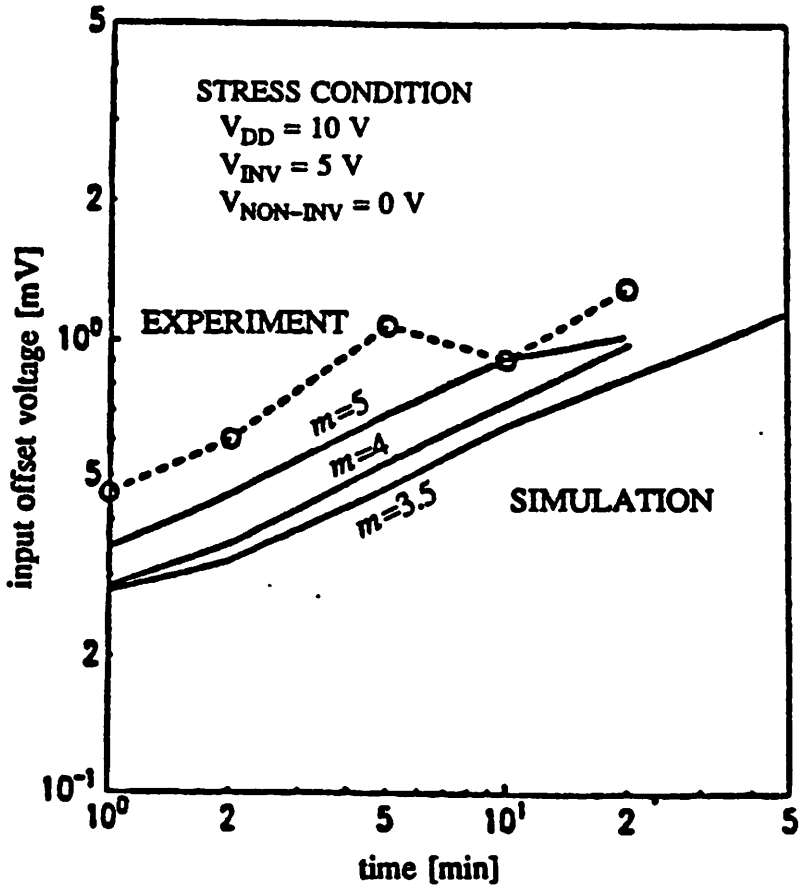


Fig. 3.47 The input offset voltage plotted as a function of stress time. Dotted line denotes experimental data, while solid lines denote CAS simulations at different m values.

3.8.4 Summary

In this section, two experimental examples were presented to provide CAS verification and to suggest parameter extraction and calibration techniques that provide the best accuracy to measured data. Although in the case presented here the special hot-carrier-enhanced ring oscillator showed a fair amount of degradation, in general, for circuits using normal process techniques and operating conditions, it has been found that CMOS inverter-based circuits are fairly robust towards hot-carrier degradation. Part of the reason for this robustness is due to the fact that only approximately one-eighth of the device-level degradation shows up at the circuit level (this topic is discussed in Chapter 4). However, it is expected that analog circuits will be more susceptible to hot-carrier degradation because of their heavy reliance on device matching.

3.9 BERT Programmer's Guide

3.9.1 Introduction

This chapter describes the basic structure of the BERT program. Three modules, the hot-carrier degradation module CAS, the oxide reliability module CORS, and an electromigration module are all included in BERT in modular fashion. Section 3.9.2 explains how new modules can be incorporated into BERT, Section 3.9.3 probes in further detail into the CAS portion of the program and suggests methods of adding new models, and the remaining two sections deal with debugging options and modifying BERT for use with a circuit simulator other than SPICE.

3.9.2 Overall BERT Structure

Fig. 3.2 already showed the basic configuration of the pre- and post-processor in relation to SPICE. Basically, the pre-processor prebert reads in the SPICE input deck and device model parameter files, filters from the input deck all non-SPICE commands, and adds SPICE commands that are needed for BERT calculations (e.g. voltage node printouts for CAS or CORS). All model parameter files declared in the **.PROCESS** command are converted to the SPICE .model format (if the file is a BSIM1 process file (see [Jen87] or Section 3.5)) and copied to the temporary files *rwmdx*, where *x* is an integer to differentiate between the different model parameter sets. The *rwmdx* files are created for use by the post-processor postbert for both SPICE2 and SPICE3. For the BSIM1 model in SPICE2, the temporary files *RWPRCX* are also created which are in the same format as the BSIM1 process files except all lines dealing with the I_{sub} and degradation parameters are commented out. These files are used in the **.PROCESS** declaration in the modified SPICE2 input deck. For SPICE3 and non-BSIM1 models in SPICE2, the temporary files *rwmdx* are appended to the input deck, with I_{sub} and degradation parameters being commented out during the copy. Prebert also creates a file called *rawsub* for communication with postbert. The file *rawsub* contains information such as what BERT-specific commands have been specified (since all BERT-specific commands are filtered out of the input deck before they reach postbert), number of transistors in which analysis is requested, etc.

When prebert is run with the file *agetable* present, prebert will read in the transistor ages from *agetable*, generate the aged model parameters, and output them into files named *AGEx*, where *x* differentiates between model parameters sets created either from different models, or from the same model but from different ages. Prebert then will create the file *inpdeck* which modifies the input file so that the new model parameter files *AGEx* can be used. The aging commands **.AGE** and **.AGEPROC** are also deleted in *inpdeck*.

Postbert reads the SPICE output file through standard input and opens *rawsub* for reading. All model parameters are loaded by opening the *rwmdx* files. After all calculations are finished, postbert writes to standard output the SPICE output file with the results of BERT appended to it. All temporary files such as *rawsub* and *rwmdx* files are erased.

Both prebert and postbert are written in standard UNIX C and use dynamic memory allocation (using malloc() and calloc()) so that there is no inherent limitations to the number of transistors or number of SPICE timesteps that can be accommodated (available memory and hard disk space become the determining factor). Program testing has been done on DEC VAX mainframe machines, the DEC VAXstations and DECstation[1] 3100 workstations, the Sun 3/60, and the Sun SPARCstation[2].

The program has also been successfully compiled on the IBM PS/2[3] using Microsoft C Version 5.1. Although no known bugs exist using this compiled version, extensive testing has not yet been done in the PC environment.

The following subsections describe prebert and postbert in more detail.

Prebert:

Fig. 3.48 shows how the main routine in the source code file *prebert.c* is organized. The routine OpenRaw() opens *rawsub* for writing, and the routine ArgU() reads in all arguments and makes sure

---

[1]VAX, VAXstations, and DECstations are trademarks of Digital Equipment Corporation.
[2]SPARCstation is a trademark of SUN Microsystems, Inc.
[3]PS/2 is a trademark of Intenational Business Machines, Incorporated

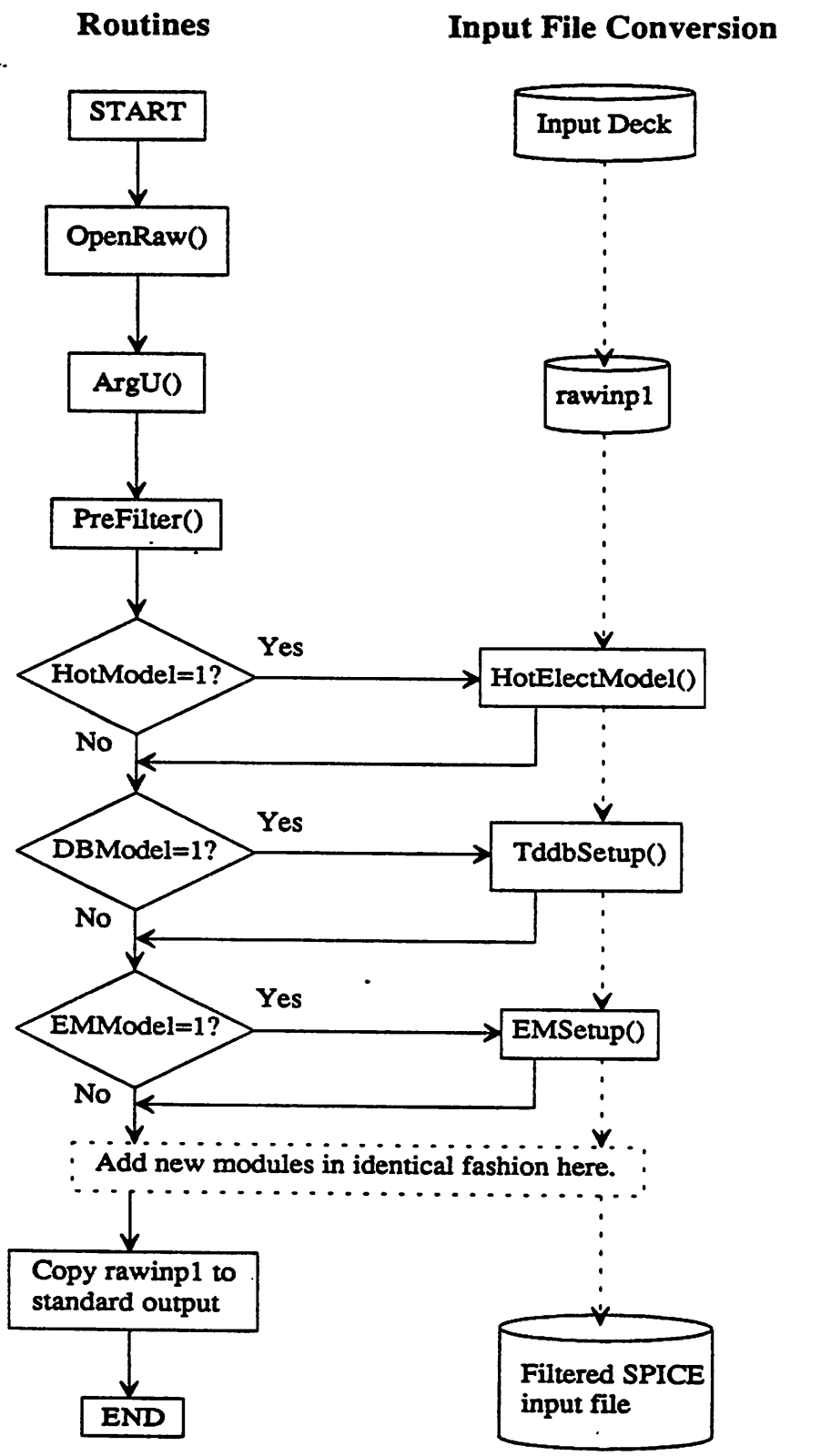


Fig. 3.48 The structure of the main program in *prebert.c*. The solid arrows show the path of program execution, while the dotted arrows show the path the input deck takes through the different modules. Square symbols represent routines or a particular program function, diamonds are decision points, and barrels represent files.

relevant files can be opened. At this stage, the input deck is now temporarily written to a file called *rawinp1*.

At this point, control is transferred to the routine PreFilter()[4] which scans *rawinp1* for BERT-specific commands and sets up the necessary flags. The flags HotModel, DBModel, and/or EMModel are set to 1 if CAS-specific, CORS-specific, or electromigration-specific commands are present, respectively. Once this is done, control is passed to the three modules depending upon the status of the flags.

Any module executed then reads in *rawinp1*, adds lines to *rawsub* for communication to postbert, and modifies *rawinp1* as necessary (for example, for CAS simulations, **.PRINT** commands for voltage printouts are added). In practice, since reading and writing simultaneously to the same file is not permitted, the unmodified portions of *rawinp1* are copied to a file *rawinp2*. All additions or modifications necessary are also done in the file *rawinp2*. Finally, before the particular module is exited, *rawinp2* is moved to *rawinp1*. Once all necessary modules are executed, the main routine checks for any errors that may have occurred, and if not, *rawinp1* is dumped to standard output after appending the model parameter files. Prebert then erases *rawinp1* since the file is no longer necessary.

Error flagging is handled by the global integer variable 'Error' and the character string 'ErrMsg' which contains a description of the error (including the one-letter module identifier and two-digit error code for prebert errors). If an error occurs, ErrMsg is printed to standard error so that the user can immediately see that something is wrong by looking at his screen. ErrMsg is also placed in the rawsub file so that postbert can print it out onto the standard output file as well.

Following is a list of the source code files and a brief description of each:

1) **prebert.c:** This file contains the main routines of BERT and CAS.

2) **procsub.c:** This file contains routines that create the *rwmdx* and *RWPRCX* files. The *rwmdx* files are in SPICE '.model' format, while the *RWPRCX* files are in BSIM1 process file format with the I_{sub} and degradation parameters commented out.

---

[4]For version 1.1 and later. Version 1.0 has the routine PreFilter() imbedded in the CAS routine HotElectModel() in *prebert.c*.

3) **age.c:** This file contains the main routines that calculate the aged model parameter sets for CAS.

4) **bsimext.c:** This file contains routines that extract the aged model parameter sets for the BSIM1 model (SPICE3 Level 4).

5) **spext.c:** This file contains routines that extract the aged model parameter sets for SPICE Level 1, 2, and 3 models.

6) **tddb.c:** This file contains all routines related to CORS.

7) **preem.c:** This file contains all routines related to electromigration simulation.

8) **premisc.c:** This file contains miscellaneous routines used by the rest of the program.

9) **bertpre.h, bertpr2.h:** These files contain the global variable declarations of prebert. *bertpre.h* contains all variable declarations and is included only in *prebert.c* and *procsub.c*. All variables in *bertpr2.h* are declared as extern variables and are used for the other source files .

10) **tdbprdef.h:** This file contains the variable declarations for CORS.

11) **empredef.h:** This file contains the variable declarations for the electromigration module.

To summarize, the following items must be done when adding a new module to prebert:

1) Searches for new keywords and commands should be added to the routine PreFilter().

2) Appropriate global flags must be set according to the keywords found in PreFilter() so that control is transferred to the new module when required. The global flags should be declared in the variable declarations files *bertpre.h* and as an extern variable in *bertpr2.h*.

3) Each module must adhere to the rule of reading *rawinp1*, writing the modified input file to *rawinp2*, then moving *rawinp2* to *rawinp1* (*rawinp2* should not remain after the module is exited). All non-SPICE commands related to the module should be commented out. The module should be in its own source file, and it may have its own variable declaration file. Any other

intermediate files that need to be created must begin with the word 'raw', and any files not needed by the remaining portion of prebert or postbert must be deleted before exiting the module.

4) If any errors are detected, the variable 'Error' must be set to 1 and a description of the error including its error code should be copied to the string variable 'ErrMsg'. The error code consists of one or two letters identifying the module involved, followed by a two-digit number, in turn followed by a colon before the actual error message (see Appendix 3A for CAS error codes for examples).

Postbert:

Fig. 3.49 shows the routines within the main program of *postbert.c*. The format for program execution path, path of the output file, and symbols are the same as in Fig. 3.48. First, after some error checking, standard input (the SPICE output file) is copied to the file *rawout1*. Similar in function to *rawinp1* in prebert, *rawout1* becomes the file that each module reads to do its calculations. Any deletions to the output file (for example, deleting all CAS-requested voltage node printouts) are made and copied to the file *rawout2*, which is subsequently moved to *rawout1* before the module is exited. Any information that needs to be added to the output file by each module is saved in separate files (for instance, the file *rawhot* in CAS), and these files are appended to the final *rawout1* file that exists after all necessary modules have been executed. Finally, *rawout1* is placed into standard output.

Error flagging is handled in similar fashion to prebert, except that the integer variable Error is equal to 1 if a postbert error has occurred, and Error is equal to 2 if a prebert error previously occurred (the prebert error assignment is automatically handled by the main program as long as each module in prebert correctly assigns the error variable and message). Again, all error messages are written into the character string ErrMsg, including the one- to two-letter module identifier and, this time, a three-digit error code for postbert errors, and the error messages are printed out both to standard error (which shows up immediately on the screen) and to the output file.

Following is a list of the source code files and a brief description of each:

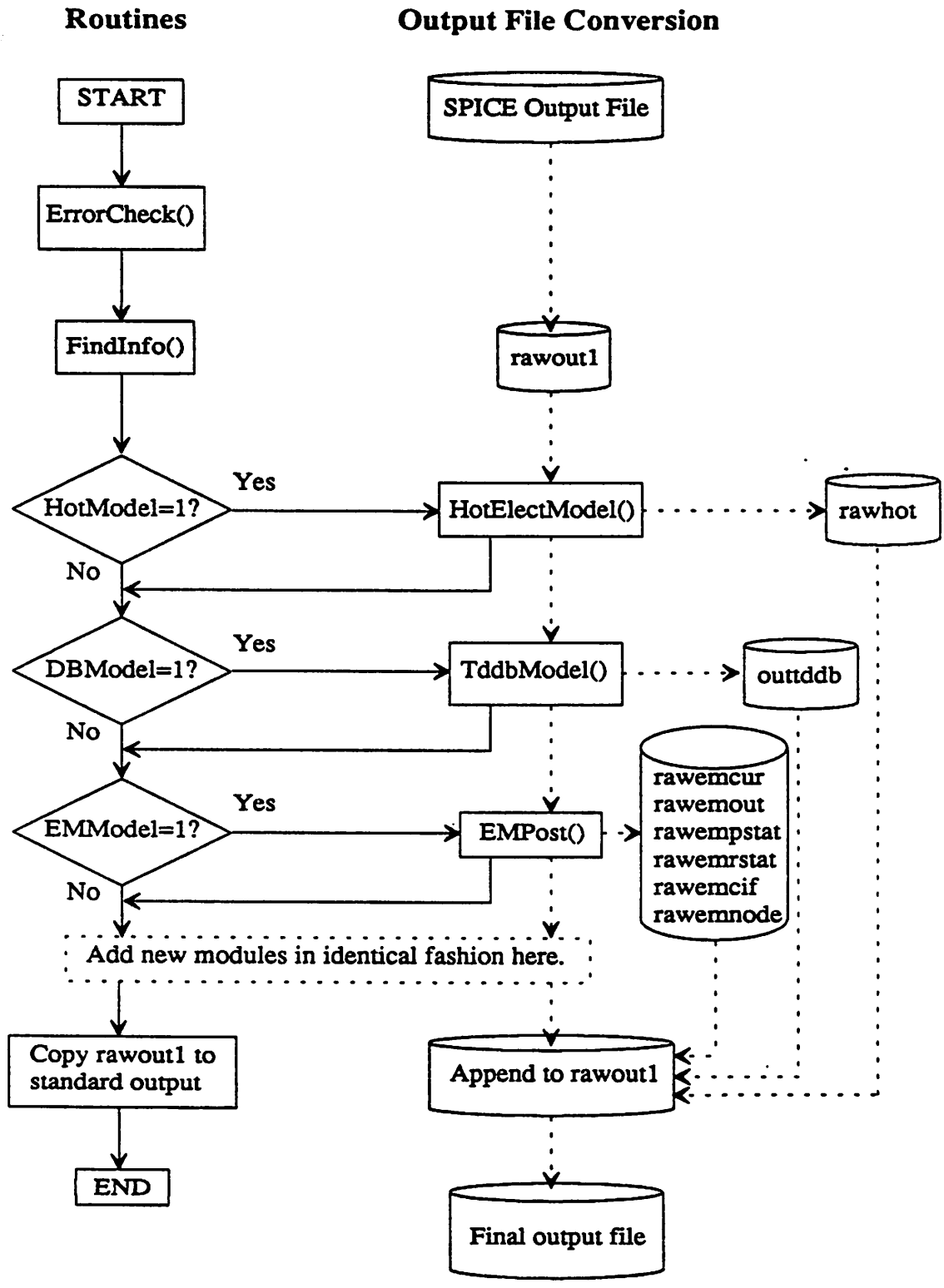


Fig. 3.49 The structure of the main program in *postbert.c*. The solid arrows show the path of program execution, while the dotted arrows show the path the SPICE output file takes through the different modules. Square symbols represent routines or a particular program function, diamonds are decision points, and barrels represent files.

1) **postbert.c:** This file contains all the main routines of BERT.

2) **readpar.c:** This file contains routines that read various transistor information and obtain model parameters from the *rwmdx* files.

3) **degcalc.c:** This file contains the I_{sub}, I_{gate}, and degradation models for CAS.

4) **mos.c:** This file contains all the drain current models (SPICE Level 1, 2, 3, and 4 (BSIM1)).

5) **output.c:** This file contains all the routines used for CAS output file printout.

6) **postddb.c, nrcalcs.c:** These files contain all CORS-related routines.

7) **postem1.c, postem2.c:** These files contain all routines associated with the electromigration module.

8) **postmisc.c:** This file contains miscellaneous routines used by the rest of the program.

9) **bertpo.h, bertpo2.h:** These files contain the global variable declarations of postbert. *bertpo.h* contains all variable declarations and is included only in *postbert.c*. All variables in *bertpo2.h* are declared as extern variables and are used for the other source files .

10) **tdbpodef.h:** This file contains the variable declarations for CORS.

11) **empodef.h:** This file contains the variable declarations for the electromigration module.

When adding new modules to postbert, these guidelines should be followed:

1) Declare global flagging variables in the variable declaration files *bertpo.h* and *bertpo2.h* (declare as extern in *bertpo2.h*).

2) Add module to the main routine.

3) Add search for new keywords in rawsub file in the routine FindInfo() in the main routine[5].

[5]FindInfo() is imbedded in ReadParameters() in *readpar.c* in Version 1.0.

4) Each module should be written in its own source code file, with any variables specific to the module declared in its own variable declarations file.

5) Within each module, the file *rawout1* should be read for calculations purposes. *rawout1* should be copied unaltered to *rawout2* except for items requested by the module itself in prebert that are no longer necessary. However, care must be taken not to delete information that will be used by any subsequent modules (for example, voltage node printouts are used by both CAS and CORS; although CAS usually deletes the printout, it will save them in *rawout2* if CORS is requested; CORS then does the deletion). Any information that needs to be *added* to the output file should be stored in a separate file. Finally, *rawout2* should be copied to *rawout1* before the module is exited.

6) The last item to take care of is to append the module's output file to *rawout1* in the main routine after all modules have been executed.

3.9.3 Adding New Models In CAS While Using SPICE

Adding new models into the CAS portion of BERT involves more effort than incorporating a new module because new models require new parameters, which mean new keywords must be searched for and new data structures must be created. However, the necessary routines can be created in parallel with the existing routines for the present models.

First of all, in prebert, no modification is necessary if the device model parameters are to be declared in the usual '.model' format. If the model parameter format is unlike that of SPICE or BSIM1, a translator routine such as ProcModSub() in *procsub.c* must be written to translate the format to a form recognizable by SPICE. Note that the new model that is implemented must also be implemented in SPICE.

In postbert, the formulation of the new model should be added as a routine to the file *mos.c* for drain current models, and *degcalc.c* for models involved with device degradation (including substrate and gate currents). Once these have been added, the routines involving calculation of currents should be used in

the routine CalculateCurrents() in *postbert.c*, while routines involved with calculating device degradation should be included in SubAnalysis() in *postbert.c* in similar fashion to the existing degradation routines.

The new model parameters should be appended to the existing TrnArray data structure in the variable declaration files *bertpo.h* and *bertpo2.h*. A new routine in parallel with ReadParameters() should be created in *readpar.c* to recognize new parameter keywords and load the model parameters into the appropriate variables. A routine similar to BSIMsetup() and SPICEsetup() should be created to assign default values to parameters not assigned explicitly.

Fig. 3.50 shows the algorithm in *age.c* used in prebert to calculate aged model parameters for each transistor listed in the agetable. After general memory allocation (MemAlloc()), the SPICE input file is read to find the files listed in the .AGEPROC command (GetAgeCards()). The agetable is read by GetDevAge(), and then the loop to calculate the aged model parameters for each device listed in the agetable begins. The appropriate pre-stressed model parameter files are loaded in ReadAgePar(), and a check is made to see what model is used. The routines BSGenAgeParm() and SPGenAgeParm() do the actual calculation to find the aged model parameter set from the pre-stressed model parameters and device age, and the newly-created parameters are placed in the *AGEX* files. Memory space is freed for use by the next loop if more devices remain in the agetable. Once all aged model parameters have been found for all devices, then GenInpDeck() takes the original SPICE input file and creates a modified input deck *inpdeck* for use with the new aged parameter sets.

The aging portion of CAS uses separate data structures for the BSIM1 model and the other SPICE models. In addition, separate data structures are used for the pre-stressed model parameters and calculated aged model parameters. For new models, new structures should be created, and the routine ReadAgePar() must be modified to load the new parameters. New aged-parameter extraction routines specific to the model should be written (as indicated in the dotted box in Fig. 3.50). The extraction routines written for the SPICE Level 1, 2, and 3 models are the best routines to follow when creating the new routines.

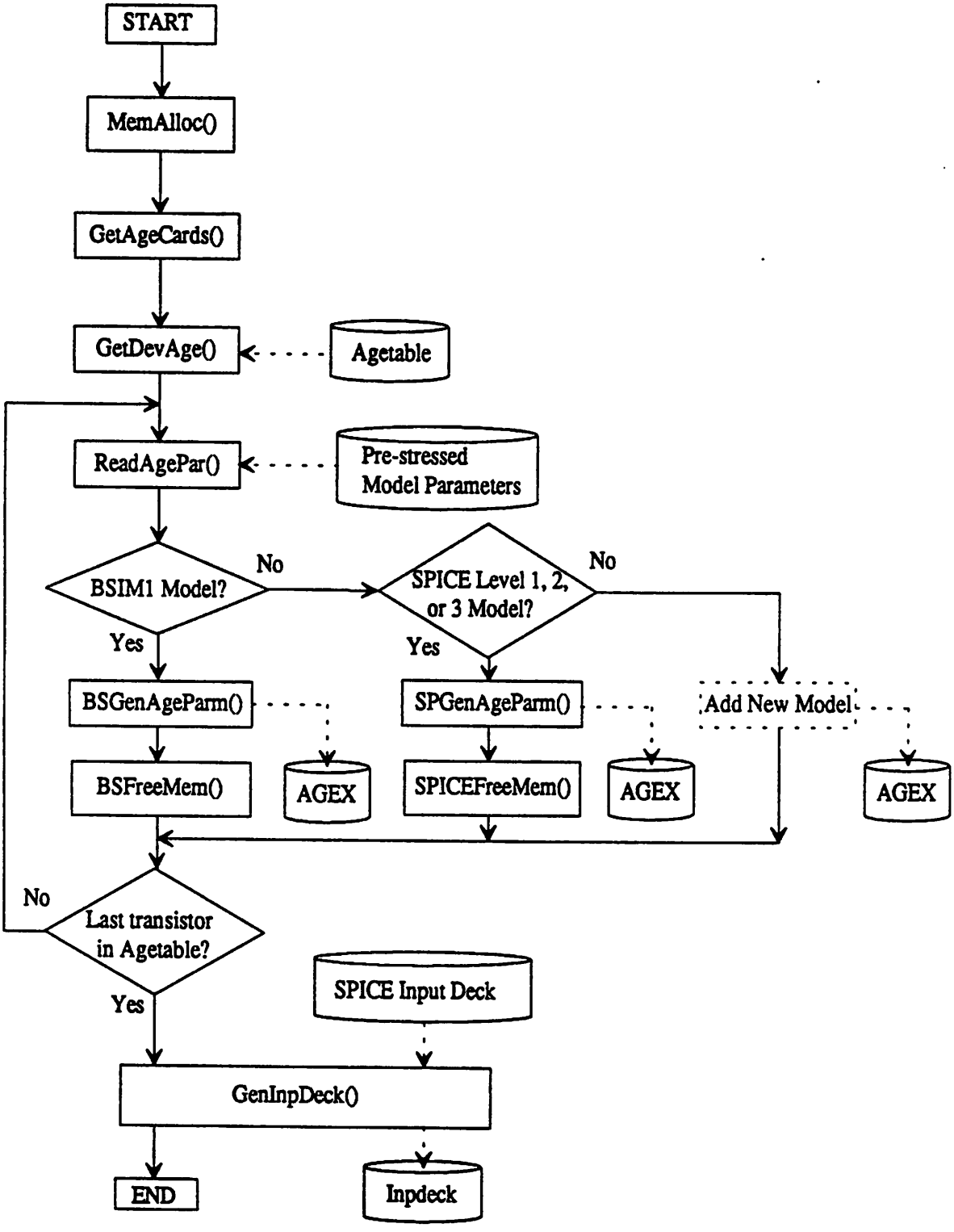


Fig. 3.50 Algorithm implemented in *age.c* of prebert to calculate aged model parameters from pre-stressed model parameters and device age. Symbols have the same meaning as in Figs. 3.48 and 3.49.

3.9.4 Modifying BERT To Work With Other Circuit Simulators

BERT, in its present form, cannot be used with any other circuit simulator except for SPICE2G.6, SPICE3B1, and SPICE3C1 because of its use of specific keywords and patterns to recognize input deck commands and output file information. All parsing routines for both prebert and postbert must be checked to make sure the correct keywords are searched, while all output routines in prebert must be examined to verify the formats are correct. These routines are generally interspersed throughout prebert but are isolated to the routines (and subroutines of) ReadVoltage(), TddbModel(), and EMPost() in *postbert.c* in postbert.

3.9.5 Debugging Options in BERT

To use the UNIX C debuggers (such as dbx), alter the top-level *Makefile* so that CFLAGS is defined as '-g' (to include the symbolic table used for debugging) rather than '-O' (optimized for execution). For current debugging purposes in CAS, delete the comment symbols surrounding the definition of the constant 'DEBUGCUR' near the beginning of *postbert.c*.

3.9.6 Summary

This section has provided some hints and guidelines to follow when adding new modules to BERT, new models to CAS, and when use of a different circuit simulator besides SPICE is desired. Generally, adding new modules to BERT is relatively simple because of the modular structure. Adding new models to CAS involves more effort because of the new model parameters that need to be recognized. Modifying BERT to be used for non-SPICE circuit simulators involves modifying input parsing and output routines and requires a varying degree of effort depending upon how similar the circuit simulator is to SPICE.

3.10 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.

3.11 References

[Nag75] L.W. Nagel, "SPICE2: A computer program to simulate semiconductor circuits," University of California, Berkeley, Electronics Research Laboratory Memorandum UCB/ERL M75/520, May 1975.

[Tsi76] Y.P. Tsividis and P.R. Gray, "An integrated NMOS operational amplifier with internal compensation," *IEEE J. Solid-State Circuits*, Vol. SC-11, pp. 748-754, December 1976.

[Vla80] Andrei Vladimirescu and Sally Liu, "The simulation of MOS integrated circuits using SPICE2," University of California, Berkeley, Electronics Research Laboratory Memorandum UCB/ERL M80/7, February 1980.

[Dog83] K. Doganis and D.L. Sharfetter, "General optimization and extraction of IC device model parameters," *IEEE Trans. Electron Devices*, Vol. ED-30, No. 9, pp. 1219-1228, September 1983.

[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.

[Hsu85] Fu-Chieh Hsu and Kuang Yi Chiu, "Hot-electron substrate-current generation during switching transients," *IEEE Trans. Electron. Devices*, Vol. ED-32, No. 2, pp. 394-399, February 1985.

[She85] Bing J. Sheu, D.L. Sharfetter, and P.K. Ko, "SPICE2 implementation of BSIM," University of California, Berkeley, Electronics Research Laboratory Memorandum UCB/ERL M85/42, May 1985.

[Tsu85] T. Tsuchiya and J. Frey, "Relationship between hot-electrons/holes and degradation of p- and n-channel MOSFET's," *IEEE Electron Device Letters*, Vol. EDL-6, No. 1, pp. 8-11, January 1985.

[Hir86] Y. Hiruta, K. Maeguchi, and K. Kanzaki, "Impact of hot electron trapping on half micron PMOSFETs with p+ poly Si gate," *IEDM Tech. Digest*, pp. 718-721, December 1986.

[Mit86] J. Mitsuhashi et al., "Mechanical stress and hydrogen effects on hot carrier injection," *IEDM Tech. Digest*, pp. 386-389, December 1986.

[Tzo86] Joseph J. Tzou, C.C. Yao, R. Cheung, and Hugo W.K. Chan, "Hot-carrier-induced degradation in p-channel LDD MOSFET's," *IEEE Electron Device Letters*, Vol. EDL-7, No. 1, pp. 5-7, January 1986.

[Wer86] W. Weber, C. Werner, and A.V. Schwerin, "Lifetimes and substrate currents in static and dynamic hot-carrier degradation," *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.

[Cho87] J.Y. Choi, P.K. Ko, and C. Hu, "Effect of oxide field on hot-carrier-induced degradation of metal-oxide-semiconductor field-effect transistors," *Applied Physics Letters*, Vol. 50, pp. 1188-1190, April 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.

[Bra88] Michael P. Brassington, Mark W. Poulter, and Monir El-Diwany, "Suppression of hot-carrier effects in submicrometer surface-channel PMOSFET's," *IEEE Trans. Electron Devices*, Vol. 35, No. 7, pp. 1149-1151, July 1988.

[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.

[Ong88] T.-C. Ong, Kouichi Seki, P.K. Ko, and Chenming Hu, "Hot-carrier-induced degradation in p-MOSFET's under AC stress," *IEEE Electron Device Letters*, Vol. 9, No. 5, pp. 211-213, May 1988.

[Aur89] S. Aur, "Kinetics of hot carrier effects for circuit simulation," *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

[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.

[Qua89] Thomas Quarles, "SPICE3 version 3C1 user's guide," University of California, Berkeley, Electronics Research Laboratory Memorandum UCB/ERL M89/46, April 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 *IEEE Electron Device Letters*.

[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 3A: 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'!

CHAPTER 4: RELATING CMOS INVERTER LIFETIME TO DC HOT-CARRIER LIFETIME OF NMOSFETS

4.1 Introduction

Previous chapters have presented methods to predict circuit-level hot-carrier degradation using the Circuit Aging Simulator (CAS) in conjunction with SPICE. Using simulators such as CAS is imperative when accurate prediction of circuit degradation for a wide variety of circuits and waveforms is necessary. However, for purposes of quick estimation, this chapter provides a pragmatic, albeit incomplete and tentative, answer to an urgent question: how to relate DC device lifetime to circuit lifetime for a specific class of circuits: CMOS-inverter-based circuits. There are three issues: how to determine the worst case DC lifetime of a MOSFET, how to relate the MOSFET DC lifetime to the lifetime of a MOSFET in a circuit, and finally how to relate the MOSFET current degradation $\Delta I_{ds}/I_{ds0}$ to the change in propagation delay $\Delta\tau_p/\tau_{p0}$. With the help of CAS, these three issues will be addressed in the following sections.

4.2 DC MOSFET Lifetime

Today, device lifetime prediction is commonly done with the assumption that maximum substrate current is the worst case for stressing. The theory of hot-carrier degradation suggests [Hu85]

$$\text{Degradation} = \left[C_1 \frac{I_{sub}^{\ m}}{W I_{ds}^{\ m-1}} t \right]^n \tag{4.1}$$

where Degradation denotes $\Delta I_{ds}/I_{ds0}$, $\Delta g_m/g_{m0}$, ΔV_{th}, etc., W is the device width, n is approximately 0.5, and m is around 3 and varies, as does C_1, with oxide field (V_{gd}) [Kuo88, Cho87b]. The degradation is actually related to the degradation driving force $D = I_{sub}^{\ m}/(I_{ds}^{\ m-1} W)$. For fixed V_{gs} and channel length L, $I_{ds} \alpha W$ and $D \alpha (I_{sub}/W)^m$; therefore I_{sub}/W correlates well with the degradation rate. One can then plot the logarithm of the device lifetime τ versus the logarithm of I_{sub}/W, and extrapolate to find the DC lifetime for the intended operating voltage, e.g. 5 V, at maximum I_{sub} (point A in Fig. 4.1). Whenever I_{ds}

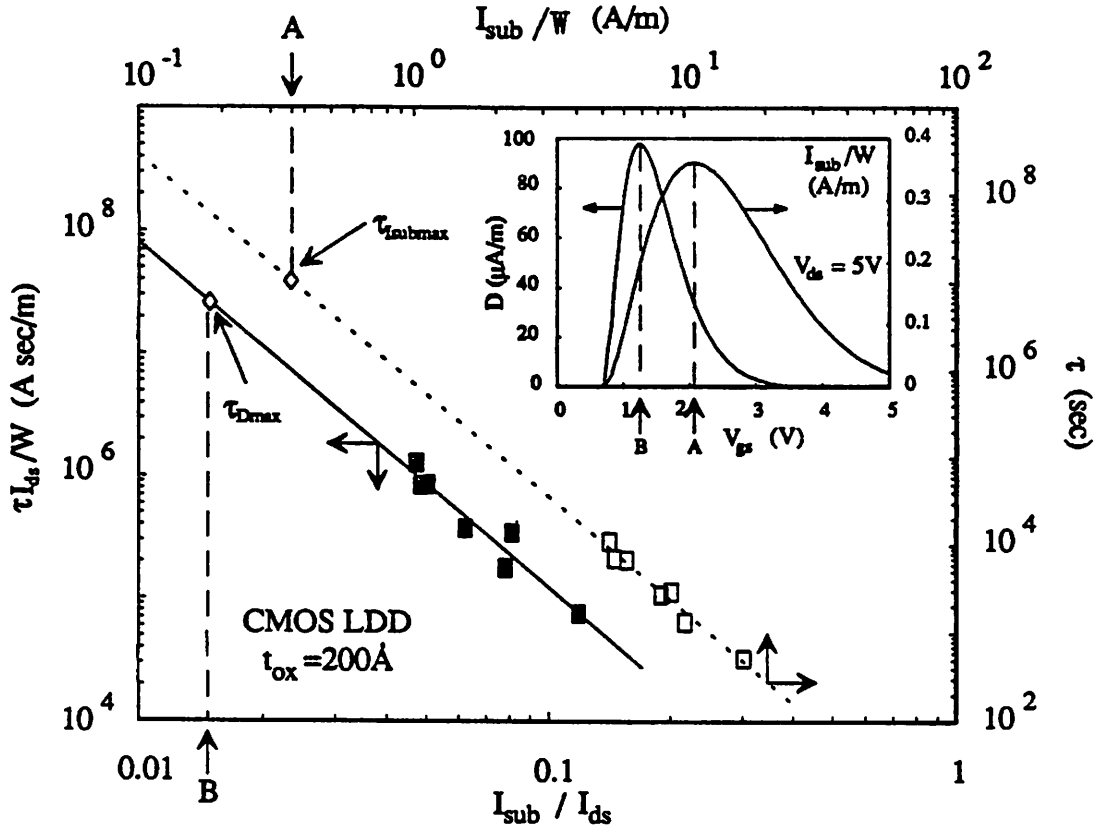


Fig. 4.1 Two commonly-used methods to extrapolate device lifetime using I_{sub}/W and I_{sub}/I_{ds}, while the inset shows degradation driving force D and substrate current per unit width I_{sub}/W of an NMOSFET plotted against gate bias. Point A corresponds to the lifetime at maximum I_{sub} for V_{dd} = 5V. Point B corresponds to the lifetime at maximum D for V_{dd} = 5V. This lifetime can also be extrapolated by plotting the lifetime measured at maximum D against $1/V_{ds}$. Process is CMOS LDD, with t_{ox} = 200 A.

varies over a large range because L varies [Cha88] or V_{gs} varies such as in AC stressing [Kuo88, Web86], lifetime has been found to correlate well with D but not with I_{sub}/W. The inset of Fig. 4.1 shows that in this example with V_{ds} = 5 V, maximum I_{sub} occurs around V_{gs} = 2.1 V, while maximum D occurs at a lower gate bias, around V_{gs} = 1.3 V. One can determine the worst-case DC stress lifetime at V_{ds} = 5 V by extrapolating the plot of the logarithm of τ (I_{ds}/W) measured at peak D versus I_{sub}/I_{ds} (point B, Fig. 4.1). Lifetime calculated at peak D (point B) is about 5 times lower than that calculated at the I_{sub} peak (point A). Lifetime in Fig. 4.1 is based on 10% reduction in I_{ds} measured at V_{gs} = 5 V and V_{ds} = 0.05 V (hereafter denoted by $\Delta I_{ds}/I_{ds0}$ unless otherwise noted).

4.3 AC Lifetime And Circuit Aging Simulation

Eq. 4.1 with its parameters determined from DC tests can easily be modified to calculate degradation for AC stress when I_{sub} and I_{ds} are time varying in the so-called quasi-static model as seen in Section 3.2 [Kuo88, Lee88]:

$$\text{Degradation} = \left[\int \frac{C_1 I_{sub}{}^m}{W I_{ds}{}^{m-1}} dt\right]^n \tag{4.2}$$

with C_1 and m being functions of V_{gd} and therefore also time varying [Kuo88, Cho87b]. This quasi-static model has been shown to correctly predict AC degradation in the 1 MHz range for the inverter-like waveforms [Kuo88, Cho87b] and is implemented in CAS [Lee88]. There are reports that quasi-static models underestimate AC degradation rates even for inverter-like waveforms at tens of MHz [Web86, Aur89]. These reports either used I_{sub} rather than D as the degradation driving force [Aur89] or did not consider C_1 and m to be functions of V_{gd} [Web86]. These omissions are known to cause underestimation of the AC degradation rate [Kuo88].

Section 3.8 presented experimental verification of CAS with measured data. We conclude that CAS simulated the in-circuit NMOSFET degradation quite accurately. In any event, there is no evidence that properly implemented quasi-static calculation underestimates the degradation of CMOS inverter propagation delay even at 40 MHz.

4.4 Relating DC Lifetime To In-Circuit MOSFET Lifetime - Duty Factor

Using CAS, we can relate device-level DC degradation lifetime to device lifetime in a circuit. We have simulated a 16-stage CMOS inverter chain undergoing 5V operation with a 100 MHz 50% duty cycle signal applied at the input. Fig. 4.2 shows the time required for $\Delta I_{ds}/I_{ds0}$ of the NMOS transistors in the circuit to equal 10%. This is labeled τ_{CAS} and is plotted for different capacitive loadings. Also plotted on the same graph are the DC lifetimes extrapolated from maximum I_{sub} stressing ($\tau_{Isubmax}$) and from maximum D stressing (τ_{Dmax}) tests (points A and B of Fig. 4.1). The in-circuit MOSFET lifetime, τ_{CAS},

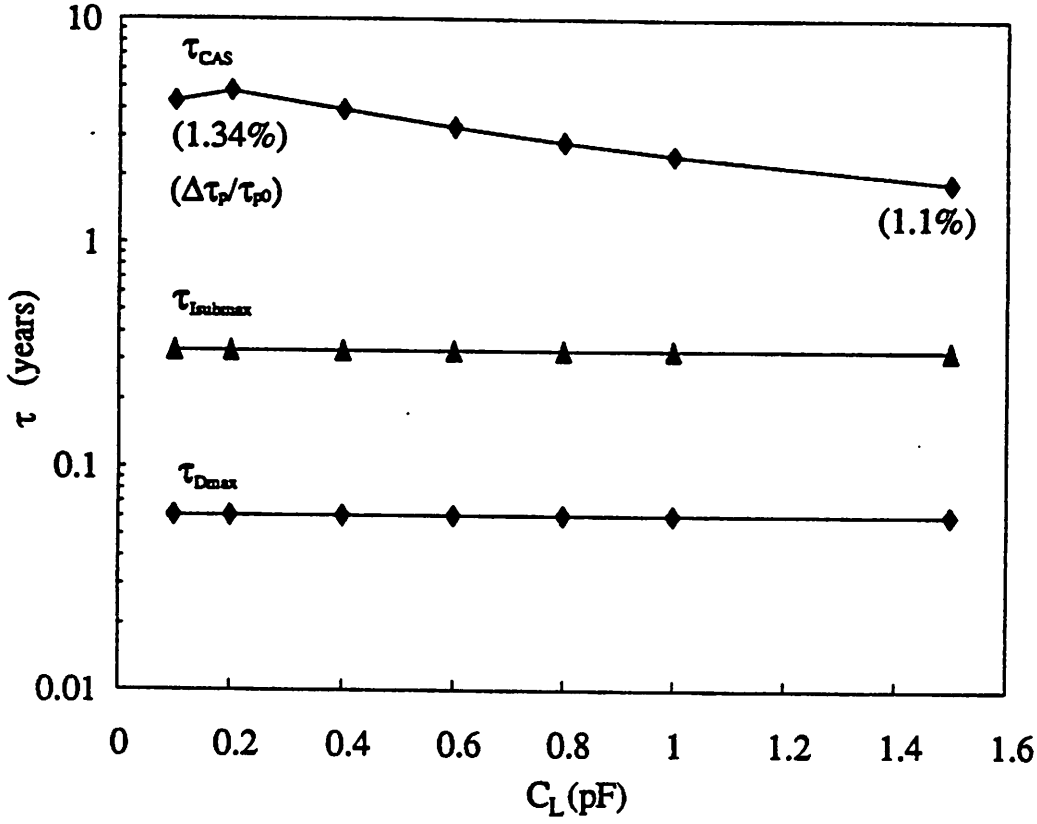


Fig. 4.2 NMOSFET lifetimes corresponding to a linear drain current degradation of 10% plotted against capacitive loading of a CMOS inverter chain. V_{dd} = 5 V, and $(W/L)_n = (W/L)_p$ = 20/1.0 in μm. τ_{CAS} is the in-circuit device lifetime when the circuit is subject to a 5 V 50% duty cycle input at 100MHz (higher than the maximum clock rate for 1 μm technology). $\tau_{Isubmax}$ denotes the DC device lifetime at maximum I_{sub} (Fig. 4.1, point A), while τ_{Dmax} denotes the DC device lifetime at maximum D (Fig. 4.1, point B). Note that $\tau_{Isubmax}$ and τ_{Dmax} are independent of capacitive loading since only one lifetime at points A and B of Fig. 4.1 is chosen.

is about 6 times the commonly-measured $\tau_{Isubmax}$ and 30 times τ_{Dmax}. This agrees with the simulated typical duty factor of D(t) around 3.5%.

4.5 Relating $\Delta I_{ds}/I_{ds0}$ To Degradation Of Propagation Delay

The inset of Fig. 4.3 shows the typical I_{ds} - V_{ds} curves of a fresh and stressed NMOS transistor. Note that the degradation in current is large in the linear region and decreases toward zero at large V_{ds}. We expect the change in propagation delay to be

$$\frac{\Delta\tau_p}{\tau_{p0}} \sim \frac{1}{2}\frac{\Delta\tau_{fall}}{\tau_{fall}} \sim \frac{1}{4}\left[\frac{\Delta I_{ds}}{I_{ds0}}\right]_{V_{dd}/2} \sim \frac{1}{8}\left[\frac{\Delta I_{ds}}{I_{ds0}}\right] \tag{4.3}$$

where $(\Delta I_{ds}/I_{ds0})_{V_{dd}/2}$ is the percentage drain current degradation at V_{gs} = 5V and $V_{ds} = V_{dd}/2$ = 2.5V. The first factor of 1/2 is because the output rise time (pull-up time) is not affected by the NMOSFET and remains unchanged while τ_{fall} accounts for about 1/2 of the τ_p. The second factor of 1/2 comes from the fact that $\Delta\tau_{fall}/\tau_{fall}$ is about the average of $\Delta I_{ds}/I_{ds0}$ at $V_{ds} = V_{dd}$ (beginning of pull-down, $\Delta I_{ds} \sim 0$) and $V_{ds} = V_{dd}/2$ (effective completion of pull-down as far as the next inverter stage is concerned). Thus $\Delta\tau_{fall}/\tau_{fall}$ is half of the percentage current degradation at $V_{ds} = V_{dd}/2$ (see inset of Fig. 4.3). Finally, the third factor of 1/2 is because linear region $\Delta I_{ds}/I_{ds0}$ is about twice the $\Delta I_{ds}/I_{ds0}$ measured at $V_{ds} = V_{dd}/2$.

Fig. 4.3 shows a plot of simulated $\Delta\tau_p/\Delta\tau_{p0}$ versus linear region $\Delta I_{ds}/I_{ds0}$ for different capacitive

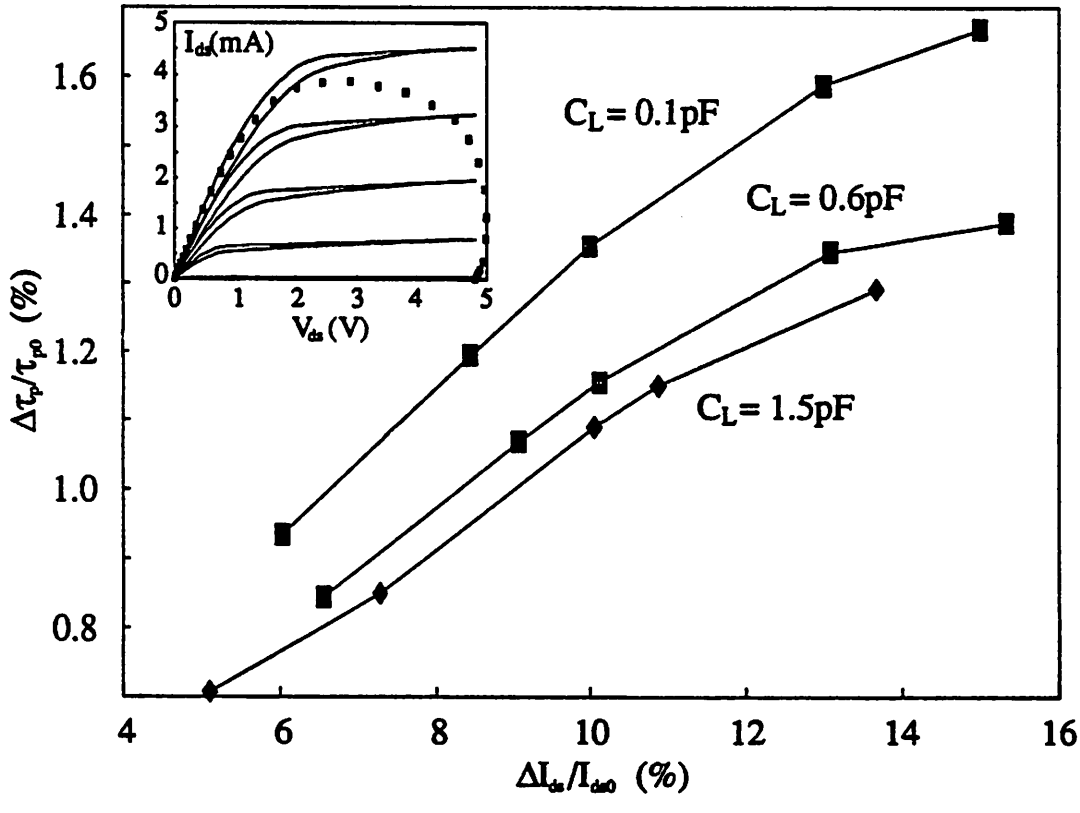


Fig. 4.3 Simulated inverter chain propagation-delay degradation, $\Delta\tau_p/\tau_{p0}$, plotted against NMOSFET linear current degradation $\Delta I_{ds}/I_{ds0}$ taken at V_{ds} = 0.05 V and V_{gs} = 5 V. The inset shows the fresh and degraded NMOS IV curves plotted with the loci (represented by square symbols) traversed by the NMOSFET during a pull-down transition. The loci of square symbols represents equal time intervals in the transition.

loadings. As predicted from the above analysis, every 8% of $\Delta I_{ds}/I_{ds0}$ results in about 1% of $\Delta\tau_p/\tau_{p0}$ in agreement with Eq. 4.3. Finally, the $\Delta\tau_p/\tau_{p0}$ values shown in Fig. 4.3, 1.1% to 1.3% for 10% $\Delta I_{ds}/I_{ds0}$ are also in agreement with Eq. 4.3.

4.6 Summary

As a pragmatic guideline, the following steps may be taken for estimating the reliability of inverter-based circuits:

(1) Determine the maximum tolerable propagation delay degradation, $\Delta t_p/\tau_{p0}$ (of the critical path), e.g. 1.25%.

(2) Calculate the corresponding maximum tolerable linear drain current degradation $\Delta I_{ds}/I_{ds0} \equiv 8$ $(\Delta t_p/\tau_{p0})$, e.g. 10%.

(3) Conduct device stressing at or near maximum D to find the DC device lifetime τ_{Dmax} corresponding to the level of degradation calculated in (2), e.g. 1 year.

(4) Estimate the duty factor of D from I_{ds} and I_{sub} simulations or by assuming that maximum D is present during the time for V_{in} to rise from V_{th} to 2V. A default estimate may be 4%.

(5) $t_{Circuit} = \tau_{Dmax}/$(Duty cycle of D), e.g. 25 years.

(6) If necessary and at a loss of accuracy, steps 3, 4, and 5 may be replaced by finding $t_{Isubmax}$ from stressing the device at or near maximum I_{sub} and using $\tau_{Circuit} = \tau_{Isubmax}/(3$ x (Duty cycle of $I_{sub}))$. $I_{sub}(t)$ width is roughly the time for V_{in} to rise from V_{th} to 3.5 V. A default duty cycle may be 6 %.

(7) Reverse the order of (1) through (6) to estimate $\Delta\tau_p/\tau_{p0}$ for a given circuit operating time.

For a quicker estimation, the following rule of thumb may be taken to estimate the lifetime of an inverter-based circuit from the DC NMOSFET lifetime: propagation delay percentage-wise increases by 1/8 of $\Delta I_{ds}/I_{ds0}$ (eg. 10%) in six times the NMOSFET DC lifetime (measured at maximum I_{sub}) at a 100 MHz clock frequency. Decreasing the clock frequency will increase the factor of six multiplying the

NMOSFET DC lifetime by the same proportion (e.g. at 50 MHz, the factor of six becomes a factor of 12).

The methods discussed in this chapter are only rough estimations. The estimate is believed to be conservative because PMOSFET current increase is not considered and a high clock rate is assumed. More accurate and general circuit aging analysis would require a simulator such as CAS and perhaps better understanding of the degradation mechanism than available today. Circuits involving V_{gs} turn-off in the presence of high V_{ds} (~ V_{dd}) are subject to enhanced degradation and may not be estimated in the above manner [Cho87a].

4.7 References

[Hu85] C. Hu, S.C. 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, No. 2, pp. 375-385, February 1985.

[Web86] W. Weber, C. Werner, and A.v. Schwerin, "Lifetimes and substrate currents in static and dynamic hot-carrier degradation," *IEDM Technical Digest*, pp. 390-393, December 1986.

[Cho87a] J.Y. Choi, P.K. Ko, and C. Hu, "Hot-carrier-induced MOSFET degradation under AC stress," *IEEE Electron Device Letters*, pp. 333-335, August 1987.

[Cho87b] J.Y. Choi, P.K. Ko, and C. Hu, "Effects of oxide field on hot-carrier-induced degradation of metal-oxide-semiconductor field-effect transistors," *Applied Physics Letters*, Vol. 50, pp. 1188-1190, 1987.

[Cha88] T.Y. Chan, C.L. Chiang, and H. Gaw, "New Insight into Hot-Electron-Induced Degradation of n-MOSFET's," *IEDM Technical Digest*, pp. 196-199, December 1988.

[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. 35, No. 7, pp. 1004-1011, July 1988.

[Lee88] P.M. Lee, M.M. Kuo, K. Seki, P.K. Ko, and C. Hu, "Circuit Aging Simulator (CAS)," *IEDM Technical Digest*, pp. 134-137, December 1988.

[Aur89] S. Aur, "Kinetics of hot carrier effects for circuit simulation",*Proceedings of the IEEE Reliability Physics Symposium*, pp. 88-91, April 1989.

CHAPTER 5: SUBSTRATE CURRENT TRANSIENT DELAY OF NMOSFETS[1]

5.1 Introduction

As discussed in Chapters 1 through 4, I_{sub} can be used as a useful monitor of hot-carrier device degradation to ultimately predict circuit performance deterioration. In this chapter, an effect *caused* by I_{sub} flow, bipolar charge storage, is presented. More specifically, this phenomenon, which causes I_{sub} flow long after the device turns off, is studied in a MOSFET inverter configuration.

In normal MOSFET inverter operation, a brief substrate current spike is generated from the inverter NMOS device with the pulse width dependent on the rise and fall times of the input and output which determines when the device is in saturation (Fig. 5.1) [Hsu85]. In general, these time factors are small so that the pulse width is usually very short.

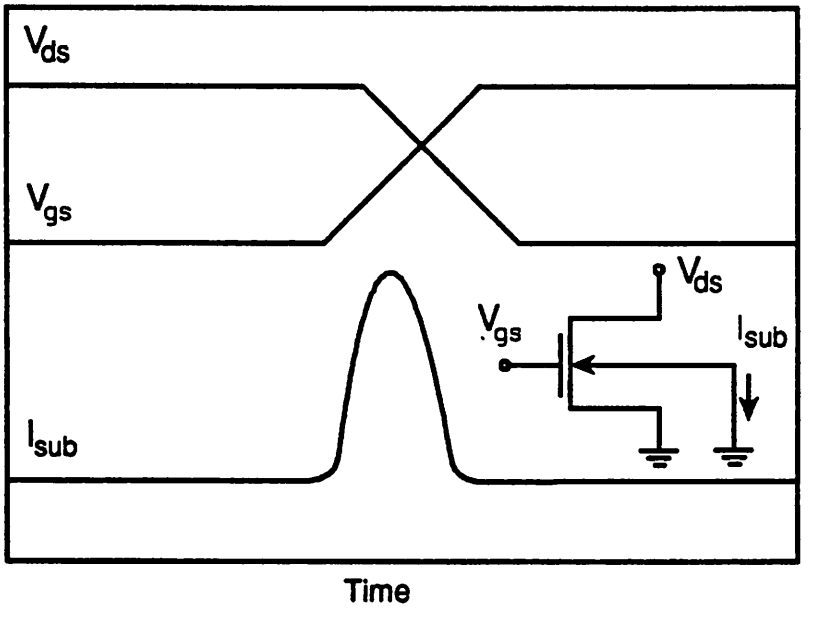


Fig. 5.1 I_{sub} pulse for a NMOSFET experiencing inverter-like waveform under normal conditions.

[1]This research was done at the Central Research Laboratories, Hitachi, Ltd., Kokubunji, Tokyo, Japan.

However, when a MOSFET inverter is operated in the regime of avalanche breakdown, a substrate current flow can be seen in transient analysis long after the MOSFET has turned off. The main source of this parasitic substrate current flow is from drainage of excess charge which is present from impact ionization and parasitic bipolar action near the MOSFET. The situation is similar to charge storage in the base of a bipolar transistor biased in the high-injection regime.

In this chapter, both measurement and simulation results of this transient substrate current delay caused by bipolar charge storage is presented. Section 5.2 presents measurement results verifying that this effect exists. Section 5.3 introduces simulation results using CADDETH (Computer Aided Device DEsign in THree dimensions), a two-carrier three-dimensional device simulator running on a Hitachi S-810 Supercomputer [Toy85]. Section 5.4 presents an extension of the normal avalanche breakdown model developed by F.-C. Hsu [Hsu83] to incorporate bipolar charge storage. Sections 5.5 considers the effects of clocked signals to the transient delay, and finally Section 5.6 demonstrates the effect this charge storage phenomenon has on CMOS latchup.

5.2 Measurement Results

Measurements were performed on several devices to verify that transient substrate current delay exists. Device measurements were conducted on non-LDD CMOS devices in a standard 18-pin dual-inline (DIP) package and connected externally in inverter fashion. An aluminum-shielded jig was constructed to house the package as well as the peripheral circuitry and wiring needed to measure the substrate current (a schematic is shown in Fig. 5.2). All active pins were separated as far as possible, and coaxial cables were used to shield individual lines. The voltage generated by substrate current flow through an external resistor R_{ext} was measured by a high-impedance probe (1 MΩ, 10 pF) connected to an oscilloscope. Other equipment used were a DC power supply and a pulse generator to produce the inverter input pulse.

Table 5.1 shows various processing parameters and physical dimensions of the NMOS device of the CMOS inverter measured, while Fig. 5.3 shows the normal I_{ds} versus V_{ds} characteristic extending into

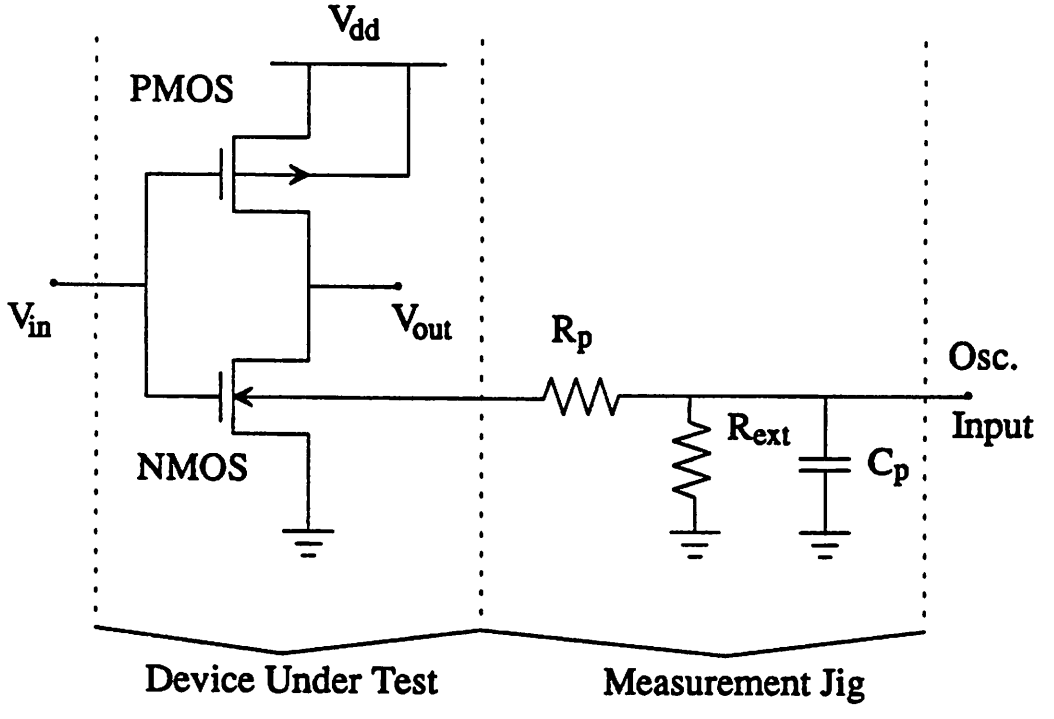


Fig. 5.2 Measurement jig setup to detect I_{sub}. R_{ext} = 100Ω, $R_p \cong 0.3\Omega$, $C_p \cong$ 400pF. R_p and C_p are parasitic elements of the jig. I_{sub} is measured by monitoring the voltage across R_{ext}.

the avalanche breakdown region. Since the device enters the avalanching region at around 6.5 volts, we can expect to see the substrate current delay phenomenon for $V_{dd} > 6.5$ V.

PHYSICAL PARAMETERS OF THE MEASURED NMOS DEVICE	
Process	Twin-well n-substrate CMOS
Gate Length (L_G)	1.25μm
Effective Channel Length (L_{eff})	0.85μm
Gate Width (W_G)	15μm
Gate Oxide Thickness (t_{ox})	250A
Source/Drain Junction Depth (x_j)	0.2μm
Threshold Voltage (V_{th})	0.1V
Avalanche Breakdown Voltage	6.5V

Table 5.1 Physical and process parameters of the NMOS driver device of the measured CMOS inverter.

Fig. 5.4 shows the transient I_{sub} characteristic for $V_{dd} = 8$ V, with the input and output voltage pulses shown below I_{sub} for comparison. Note that the substrate current peaks long after the transistor leaves saturation, and in this case, even after V_{out} drops to zero. Because of unavoidable parasitic capacitances and resistances of the measurement setup, the I_{sub} pulse has a gradual rise and fall characteristic. It is thus hard to see the flat plateau of current, but in Fig. 5.5, which shows the delay for $V_{dd} = 9$ V, the plateau can be seen more easily. Fig. 5.6 shows superimposed I_{sub} pulses for $V_{dd} = 6, 7, 8,$ and 9 V. Fig. 5.7 points out the major features of the measured characteristic.

Thus, to summarize, substrate current transient delay has been experimentally verified to exist for drain voltages within the avalanche breakdown regime of the NMOS device. The next section will present device simulation results to predict and analyze this effect.

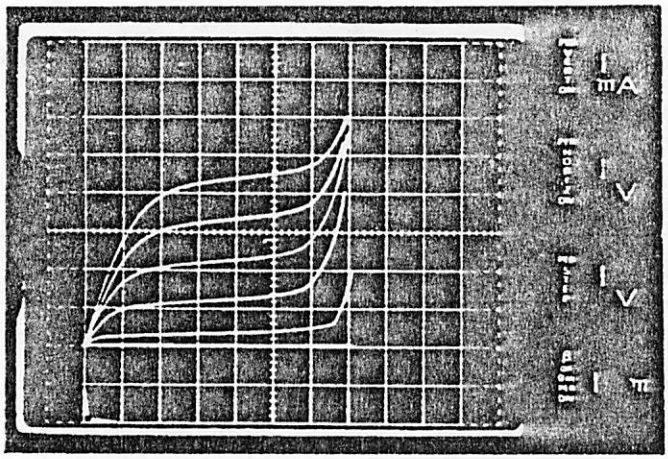


Fig. 5.3 I_{sub} versus V_{ds} characteristic for the NMOS device. V_{gs} = 0 to 5V in 1V increments. V_{ds}: 1 V/div.; I_{ds}: 1 mA/div. Avalanche breakdown can be seen to occur at $V_{ds} \cong 6.5V$.

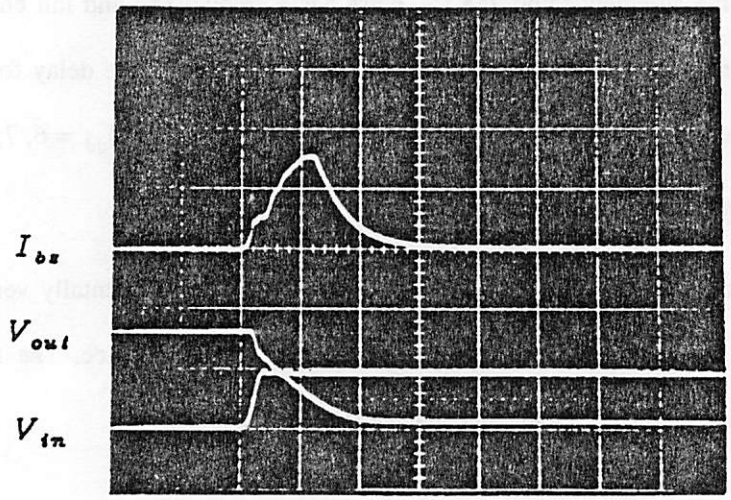


Fig. 5.4 I_{sub} versus time characteristic showing the transient delay for V_{dd} = 8V. time: 50 ns/div.; I_{sub}: 50 μA/div.; V_{in} and V_{out}: 5 V/div. $\tau_d \cong 60$ ns.

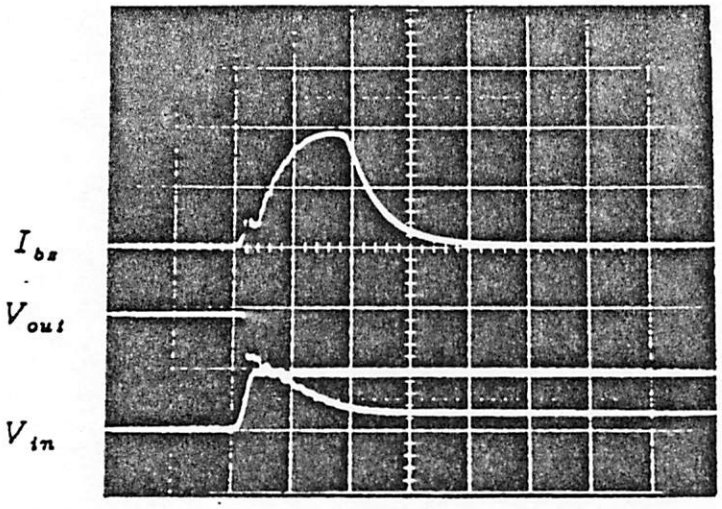


Fig. 5.5 I_{sub} versus time characteristic showing the transient delay for V_{dd} = 9V. time: 50 ns/div.; I_{sub}: 50 μA/div.; V_{in} and V_{out}: 5 V/div. $\tau_d \cong 90$ ns.

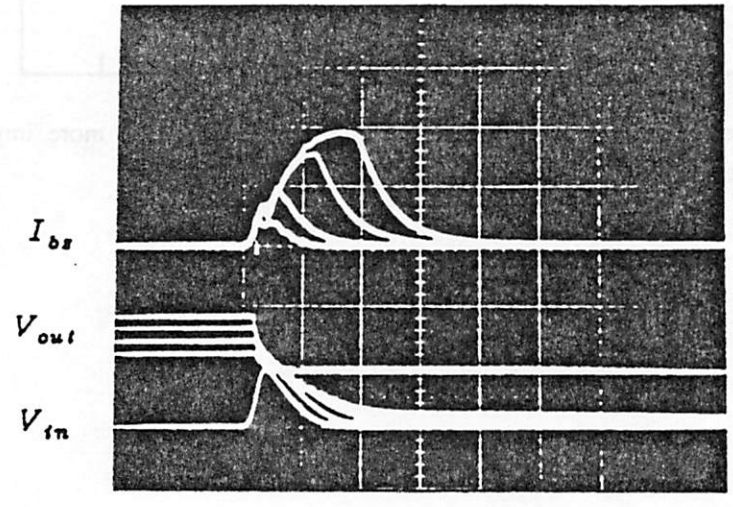


Fig. 5.6 I_{sub} versus time characteristic showing the transient delay for V_{dd} = 6, 7, 8, and 9V. time: 50 ns/div.; I_{sub}: 50 μA/div.; V_{in}: 5 V/div.; and V_{out}: 2 V/div.

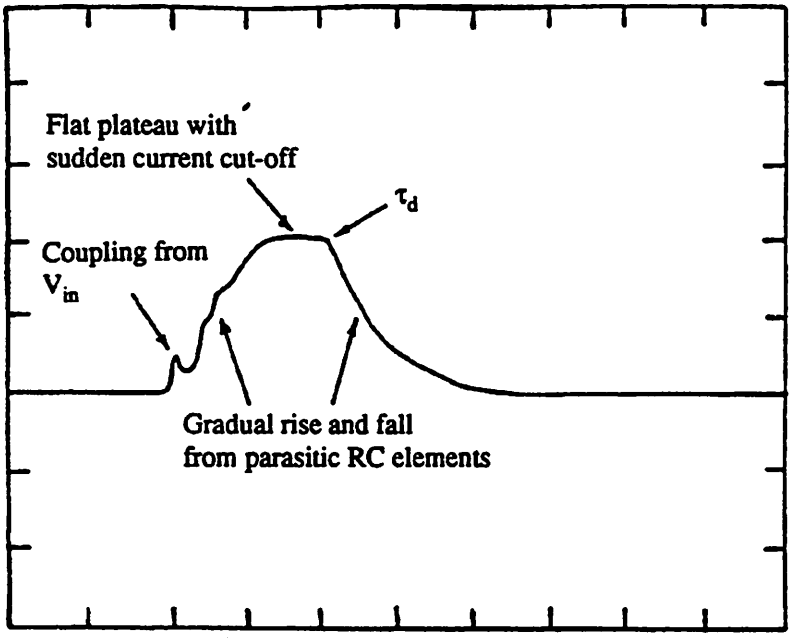


Fig. 5.7 I_{sub} versus time characteristic for V_{dd} = 9V showing the more important features of the transient delay.

5.3 Simulation Using CADDETH

5.3.1 Introduction

All simulations of substrate current transient delay were done using CADDETH (Computer Aided Device DEsign in three dimensions) on the Hitachi S-810 Supercomputer [Toy85]. CADDETH is a two-carrier three-dimensional device simulator which uses the electron and hole current continuity equations and Poisson's equation to solve for the potentials, electric fields, and electron and hole currents and concentrations within a device structure using either the Gummel-Poon method (iterating each of the three equations separately to find the solution) or Newton's method (iterating all three equations simultaneously). Both DC and transient solutions can be calculated. To reduce computational time, CADDETH code is vectorized so that the vector processor of the S-810 can be utilized. This results in approximately a 30-fold increase in computational speed compared to a normal mainframe computer, such as the Hitachi M680. Output can either be text, or by using a graphics post-processor, one-, two-, or three-dimensional line, mountain, or color contour plots can be created.

To conserve computer memory and CPU time, only the NMOS device of the inverter was simulated. Voltages were applied to the gate and drain of the NMOS device to mimic that of a CMOS inverter. Variable-size rectangular meshing is used, with denser meshing near junctions and interfaces where the solutions are expected to vary rapidly with distance. Results of both a two-dimensional and three-dimensional NMOS structure of the same device are presented.

5.3.2 Two-Dimensional Simulation

The basic simulated NMOSFET structure for the two-dimensional simulations is shown in Fig. 5.8, while the applied voltage waveforms on the gate and drain are shown in Fig. 5.9. In each case the input voltage V_{gs} was changed from 0 to 7.3V during the time interval from t = 1ns to t = 9ns, while V_{ds} was

changed from V_{dd} to 0V during the time interval from t = 5ns to t = 13ns. All terminal voltages remained fixed after 13ns, with the drain, source, and substrate grounded and the gate at 7.3V.

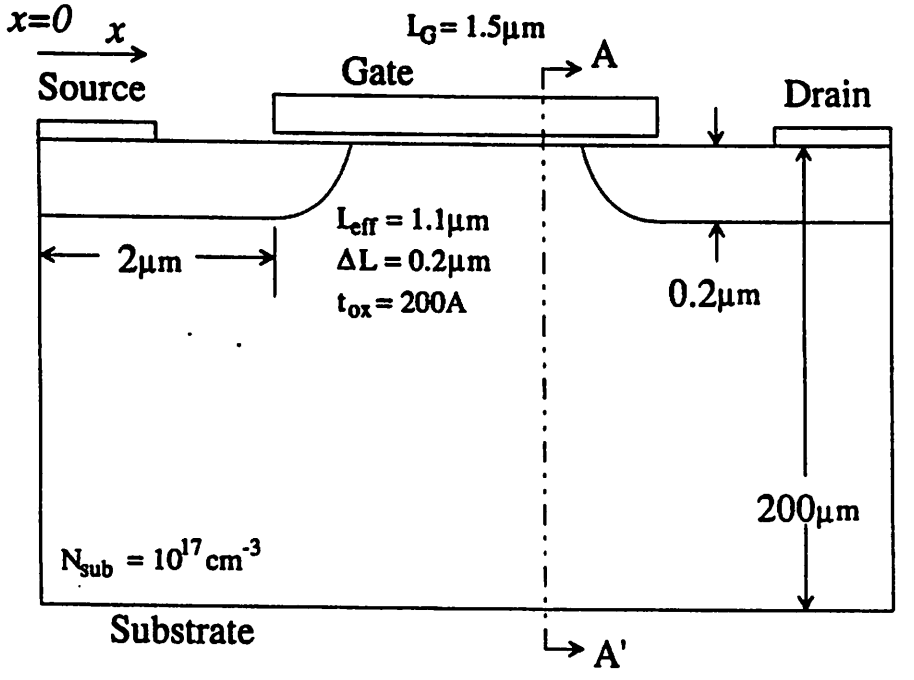


Fig. 5.8 The NMOSFET structure used in the two-dimensional CADDETH simulations.

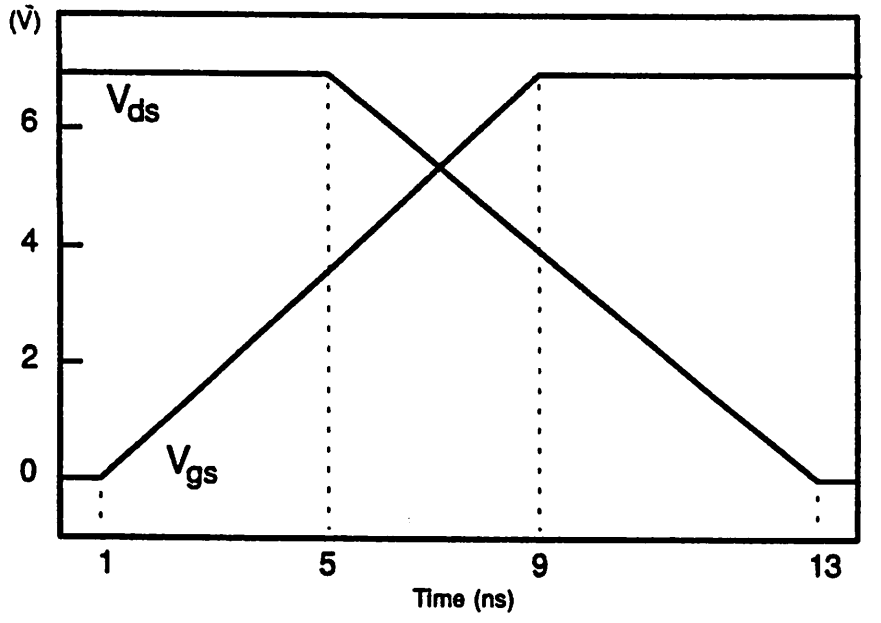


Fig. 5.9 The voltage waveform applied to the NMOSFET to simulate inverter-like behavior. The starting drain voltage was varied to see the effect on the transient delay.

To investigate variations in delay, V_{dd} (the starting drain voltage), N_{sub} , the substrate thickness d, and bulk recombination lifetime were changed from the initial setup.

The simulated transient characteristics of the substrate current for various values of V_{dd} are shown in Fig. 5.10. Note that the substrate current is nonzero long after V_{ds} is zero at t = 13ns (extreme left edge), and that during this delay, I_{sub} is relatively flat with the level of current approximately equal for all values of V_{dd}.

A qualitative explanation of the phenomenon is as follows. Initially, as drain current increases as V_{gs} rises, holes from impact ionization begin flowing into the substrate as I_{sub}. Because of the nonzero resistivity of the substrate, an increase in the local potential occurs. The rate of increase of the local potential is determined by the balance of two forces - the push of holes created by impact ionization, and the resistance of this push by the nonzero resistivity of the substrate. The magnitude of the local potential determined by these two forces and the substrate resistance determines the substrate current density flowing deep into the substrate. At this point, the space charge concentration that is produced

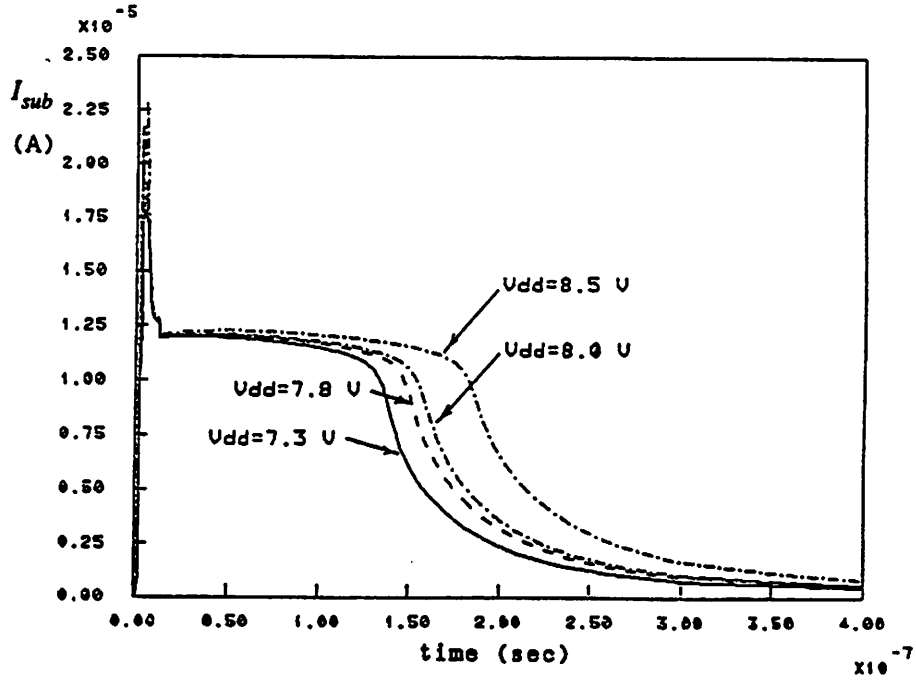


Fig. 5.10 I_{sub} versus time showing the long delay in I_{sub} cutoff. The NMOSFET is on only during the I_{sub} spike seen in the extreme left edge of the plot.

near the active device area where impact ionization occurs is equal to the excess hole concentration.

As impact ionization, and therefore I_{sub}, increases, the rise in local potential (due to the increased resistive drop in the substrate) begins attracting electrons from the source as the source/substrate pn junction begins to turn on. Because of this influx of electrons, the rate of increase of the space charge decreases and the rate of potential increase decreases as the region becomes pinned to the source. At this point, high injection conditions exist since both the concentration of holes and electrons exceed the equilibrium hole concentration of the substrate. Normally the substrate current would then approach a constant value approximately equal to the drift current that would result from an electric field with a magnitude of the pn junction forward-bias voltage divided by the thickness of the substrate. Indeed, this is the case for the time period after the drain voltage becomes zero, but as can be seen from Fig. 5.10, an I_{sub} current spike can be seen to occur (extreme left edge of the plot). This current spike can be explained by the fact that the source, because of its resistance and a large drain current flowing across it, experiences a voltage increase from the source contact to near the gate. Thus, although the voltage across the source-substrate junction is equal to the normal junction turn-on voltage, the absolute local potential of the substrate near the gate side of the source (and further to the drain) is higher (t = 5.2 ns of Fig. 5.11). This results in a higher I_{sub} flowing than would be expected if the substrate was assumed to be pinned to a grounded source. As V_{ds} decreases, I_{ds} decreases. This lowers the potential of the source close to the gate, which in turn lowers the local substrate potential and causes the slant in the I_{sub} spike. I_{sub} finally reaches the predicted pinned current level when I_{ds} equals zero at $V_{ds} = 0$.

When the source of holes is removed when V_{ds} falls to zero (13ns < t < 125ns), excess hole concentration as well as electron concentration decrease from diffusion and recombination (Figs. 5.12 and 5.13). It is important to note, however, that although excess carrier concentration decreases, the amount of space charge remains relatively constant (i.e. the difference between excess hole and electron concentrations remain the same). This can be seen from Fig. 5.14. The reason for this approximately constant space charge behavior is again found from a balance of two forces: 1) the excess charge wanting to diffuse more rapidly by increasing the plateau current level (and therefore increasing the substrate

potential at the source end and thus increasing space charge concentration), and 2) the tendency for the compensating electrons from the nearby source and drain regions to decrease the amount of space charge present (the pinned junction effect). Because of this constant space charge versus time behavior, the potential remains steady (Fig. 5.11), and the value of the electric field in the bulk remains unchanged (Fig. 5.15). The shape of the electric field and the current density (Fig. 5.16) near the surface change in shape because of the pulse of excess carriers, but their values deep in the substrate remain constant, causing the plateau in I_{sub} . As carriers diffuse and recombine further (t > 125ns), excess electron concentration is the first to disappear. After this point in time, any reduction in excess hole charge results in a reduction in space charge (Fig. 5.14). Thus, the potential starts to drop, as well as the electric field, current density, and therefore I_{sub} (Figs. 5.11, 5.15, 5.16, and 5.10, respectively). The time where electron concentration first disappears (at which the knee point in the I_{sub} characteristic occurs) can be defined as the substrate current delay time τ_d.

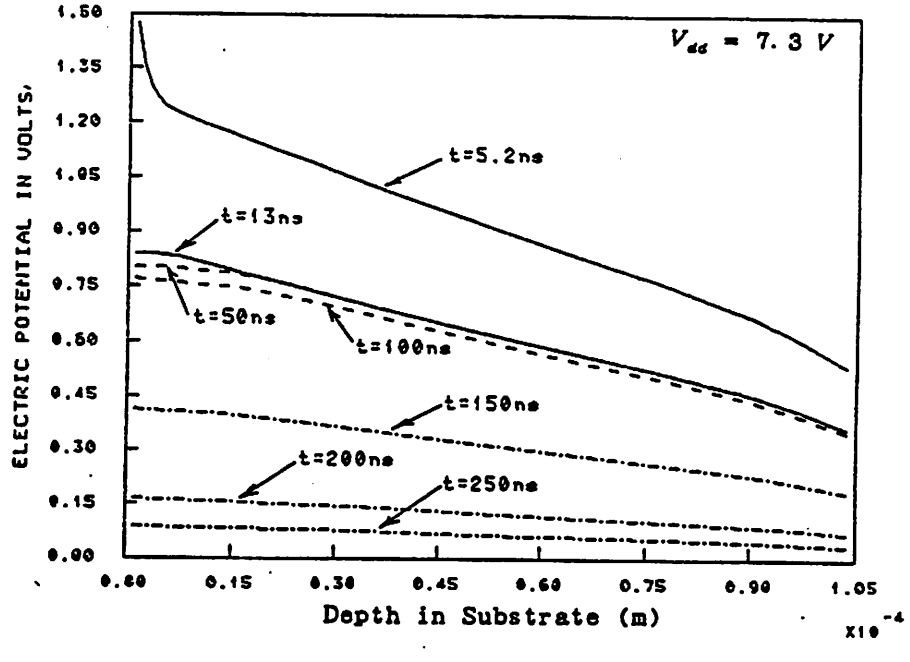


Fig. 5.11 Electric potential plotted as a function of the depth into the substrate along section A-A' of Fig. 5.8. Note the high potential value at the surface when I_{ds} is flowing for t = 5.2ns, and the relatively constant potential during the I_{sub} plateau for t = 13ns - 100ns.

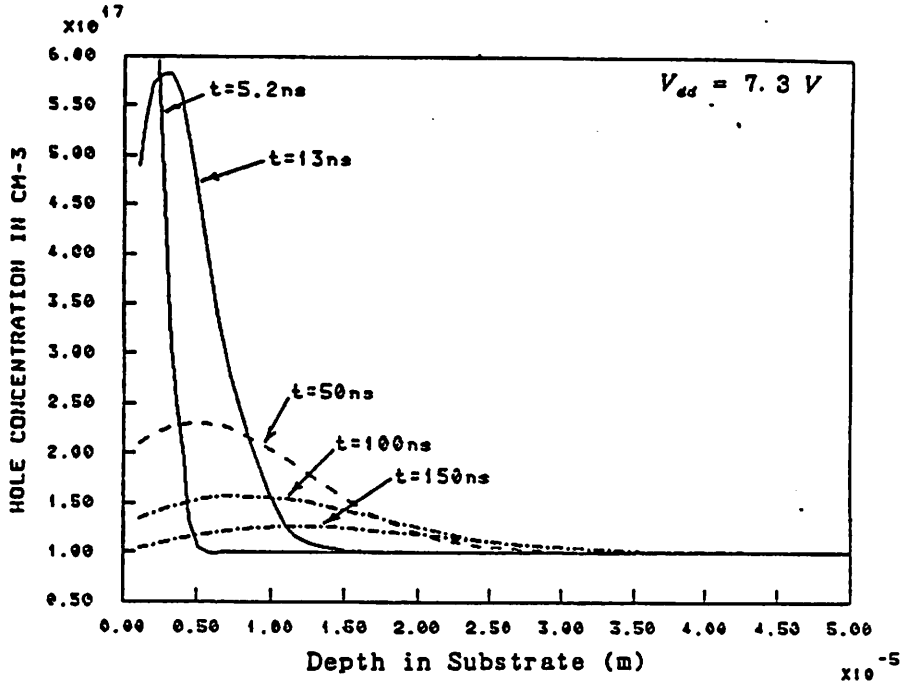


Fig. 5.12 Hole concentration plotted as a function of the depth into the substrate along section A-A' of Fig. 5.8. Note that the holes diffuse fairly deeply, in the order of 20 to 25 μm.

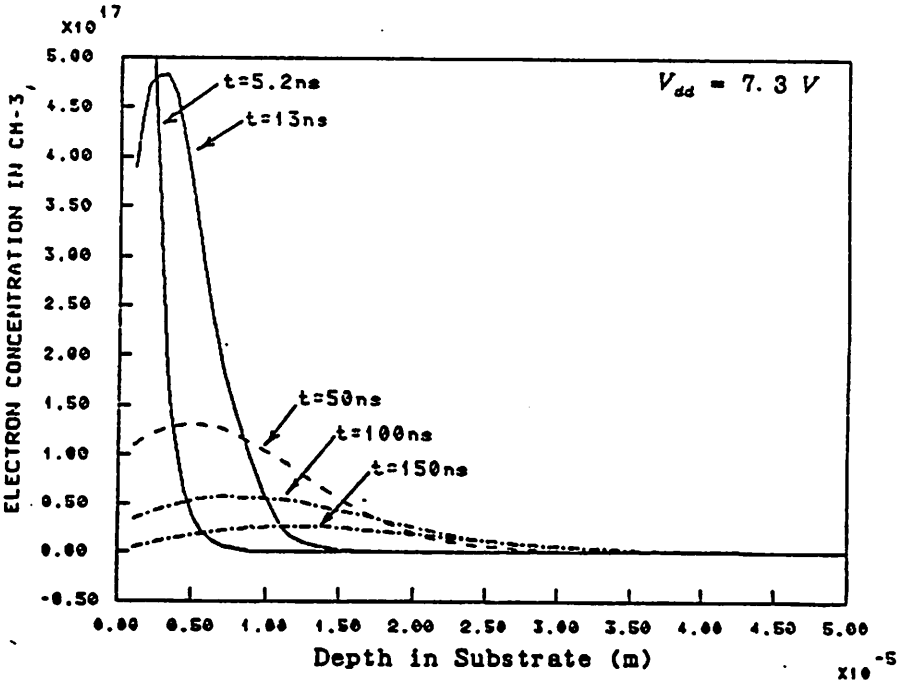


Fig. 5.13 Electron concentration plotted as a function of the depth into the substrate along section A-A' of Fig. 5.8. Note that the electrons also diffuse fairly deeply, in the order of 20 to 25 μm.

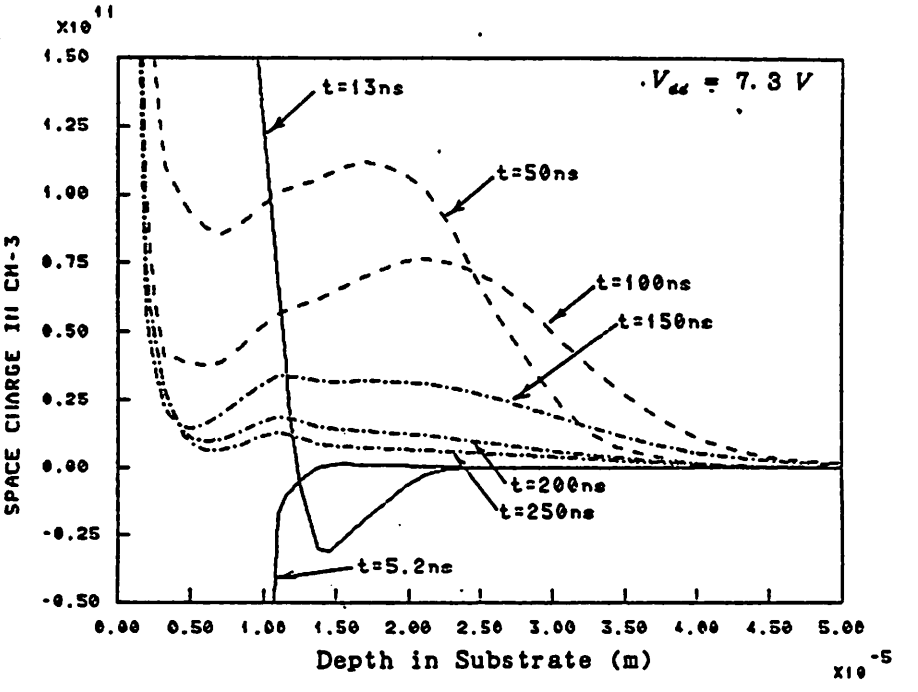


Fig. 5.14 Space charge concentration plotted as a function of the depth into the substrate along section A-A' of Fig. 5.8.

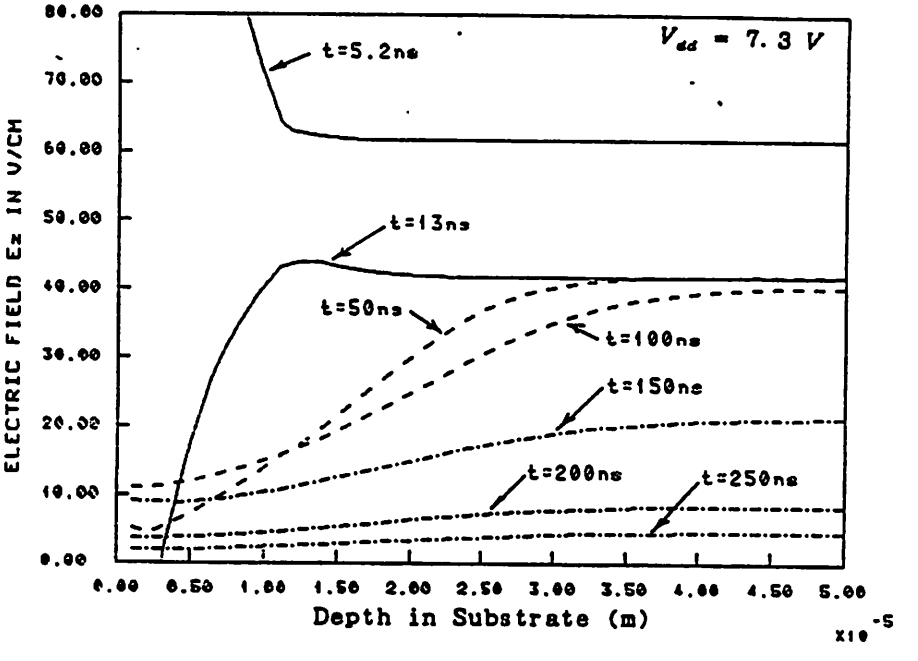


Fig. 5.15 Electric field plotted as a function of the depth into the substrate along section A-A' of Fig. 5.8. Note the fairly constant field deep in the substrate during the I_{sub} plateau for t = 13ns - 100ns.

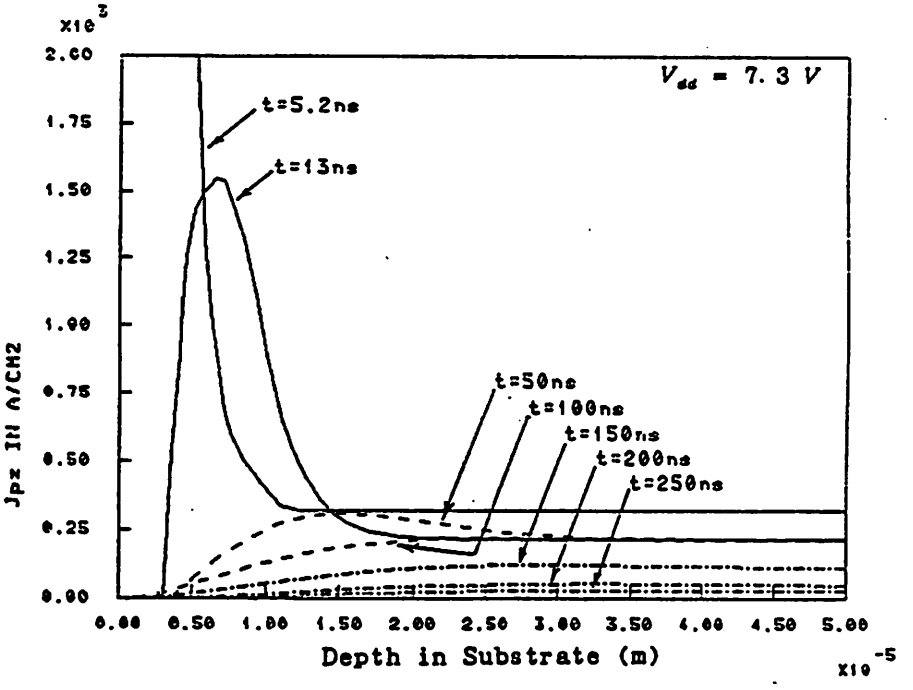


Fig. 5.16 Hole current density plotted as a function of the depth into the substrate along section A-A' of Fig. 5.8. Note the fairly constant current density deep in the substrate during the I_{sub} plateau for t = 13ns - 100ns.

Now that a general qualitative description of the phenomenon has been given, a brief explanation of the effects of varying V_{dd}, substrate doping N_{sub}, substrate thickness d, and bulk recombination lifetime can be given.

V_{dd}:

Increasing V_{dd} increases impact ionization, but since the surface substrate potential is pinned, a dramatic increase in the actual I_{sub} that flows out of the substrate contact is not possible (Fig. 5.10). Thus more excess charge builds up near the surface, as can be seen by the CADDETH simulations in Fig. 5.17. Because the I_{sub} plateau is determined only by the total substrate resistance, this current plateau will also be fixed. As a result, more time is needed to drain the excess charge away.

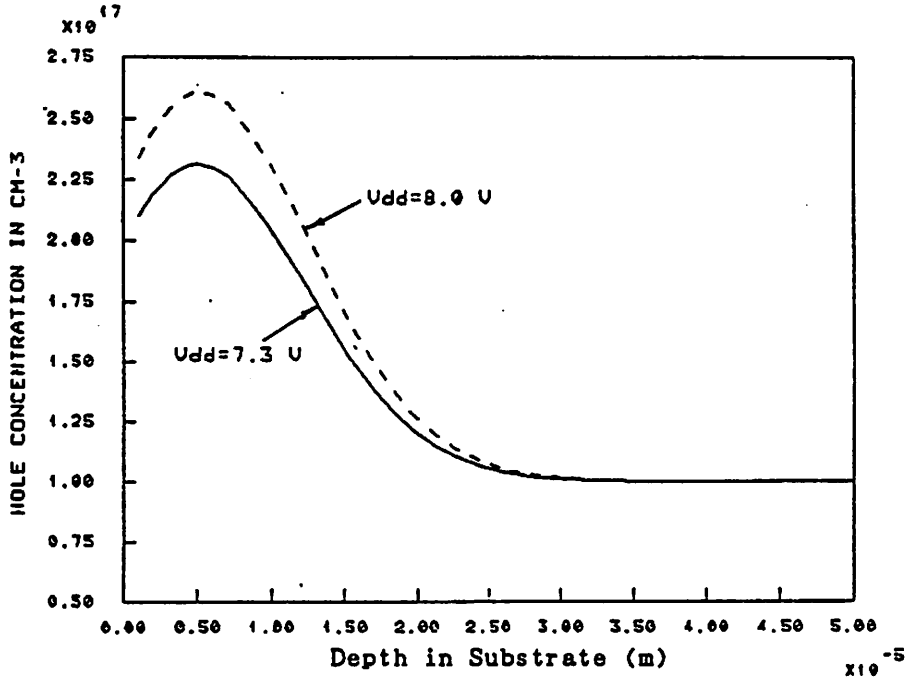


Fig. 5.17 Excess hole charge concentration plotted into the substrate for Vdd = 7.3V and 8.0V along section A-A' of Fig. 5.8.

N_{sub}:

Changing the substrate doping N_{sub} has a direct influence on the level of the current plateau. Increasing N_{sub} will decrease the total resistance of the substrate, which increases the capability of the substrate to carry more current. As a result, less excess charge accumulation and a higher current plateau will both reduce τ_d. Fig. 5.18 shows the CADDETH simulation results of this effect.

d:

Similar to increasing N_{sub}, decreasing the substrate thickness d decreases the total resistance of the substrate, which decreases the delay time. Fig. 5.19 show the CADDETH simulation result showing this effect.

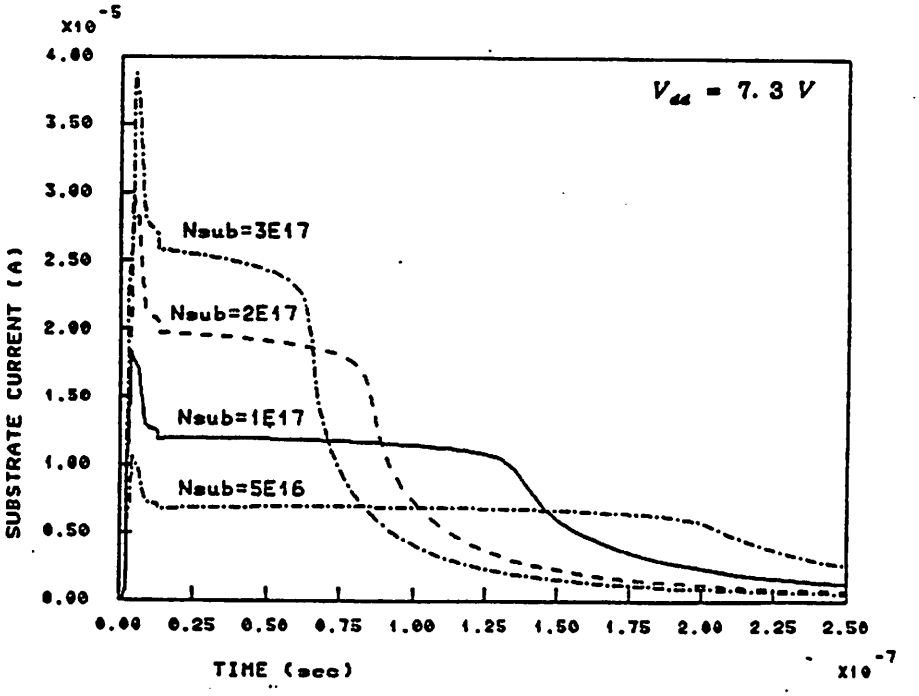


Fig. 5.18 I_{sub} versus time for various bulk substrate dopings. Note that both the I_{sub} plateau level and delay time vary.

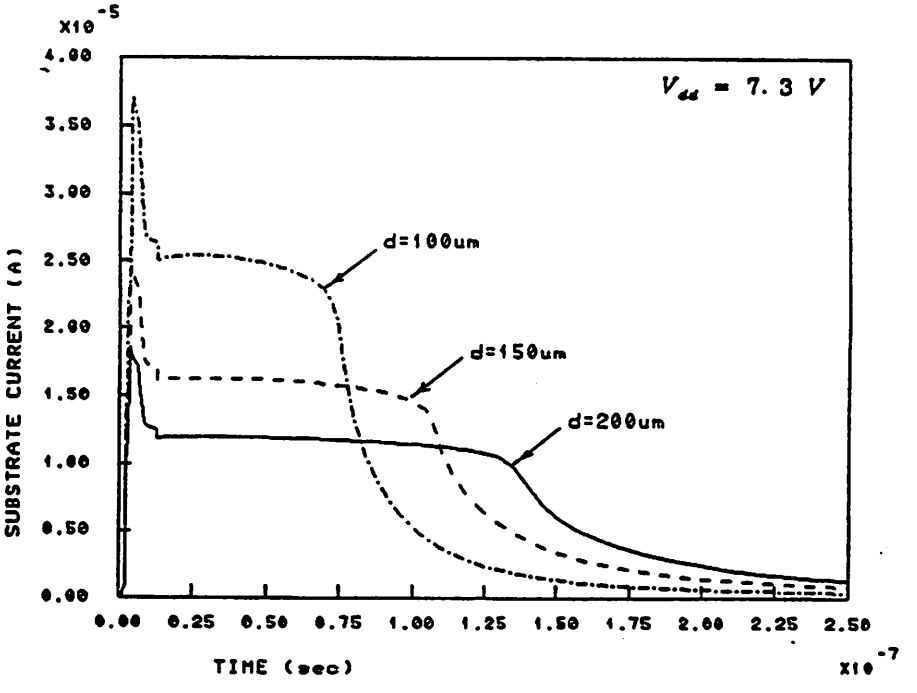


Fig. 5.19 I_{sub} versus time for various substrate thicknesses. Note that both the I_{sub} plateau level and delay time vary.

Recombination Lifetime:

Decreasing recombination lifetime will cause the excess holes and electrons to recombine more rapidly, decreasing the amount of charge that drains as substrate current and therefore decreasing the substrate current delay τ_d. Fig. 5.20 shows the change of bulk lifetime versus depth into the substrate placed in the NMOSFET structure (lifetime of the other previous cases were 1μs throughout the substrate), and Fig. 5.21 shows the shortened I_{sub} delay resulting from the shortened lifetime.

5.3.3 Three-Dimensional Simulation

The NMOSFET structure used for the three-dimensional CADDETH simulations is shown in Fig. 5.22. The substrate of the simulated structure was kept to a short 30μm so that physical behavior near the substrate could be seen in the output graphics. To compensate for the low resistance of the short substrate, a 10kΩ external resistor was added connecting the back of the substrate to ground. Furthermore, because the NMOS structure is symmetric about an axis drawn through the center of the

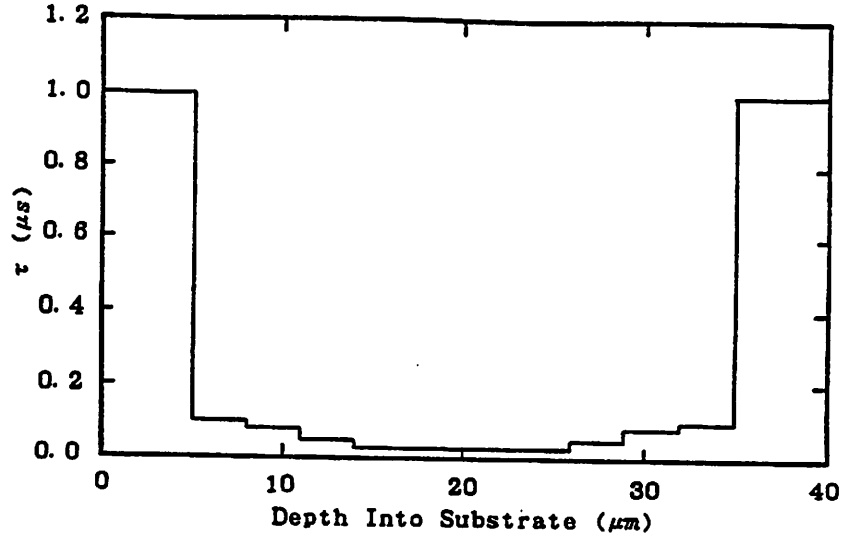


Fig. 5.20 A graphical plot of carrier recombination lifetime versus the depth into the substrate.

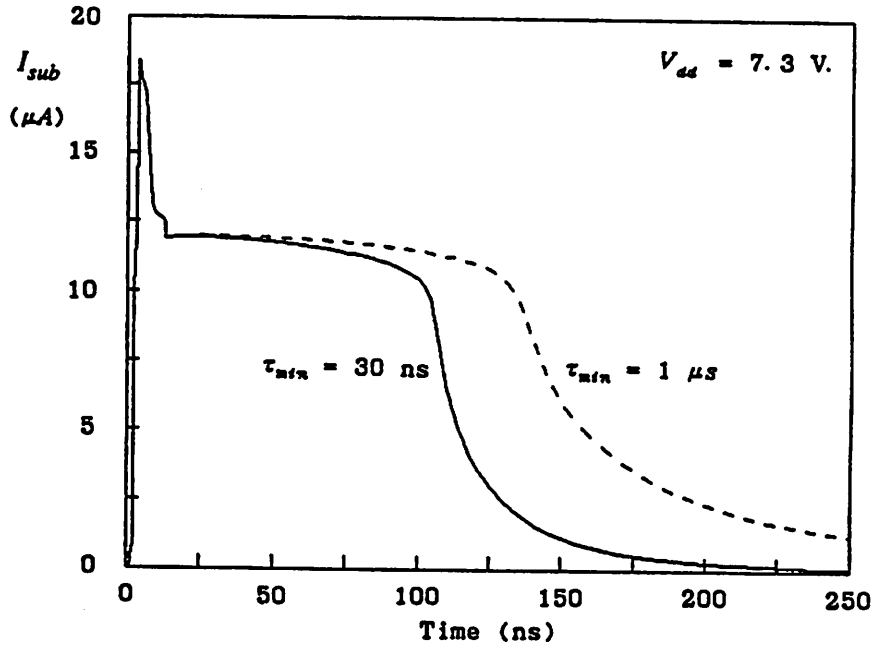


Fig. 5.21 I_{sub} versus time comparing the standard NMOSFET simulation of constant bulk recombination lifetime of 1μs (Fig. 5.10) to that with the lifetime distribution shown in Fig. 5.20.

channel, only half the device with a perfectly reflecting boundary at the axis of symmetry was simulated to decrease the number of mesh points. Field oxide 6000A in thickness was placed on the substrate surface in all areas not occupied by the channel and source/drain diffusions. Table 5.2 shows some of the physical and process parameters of the NMOSFET structure.

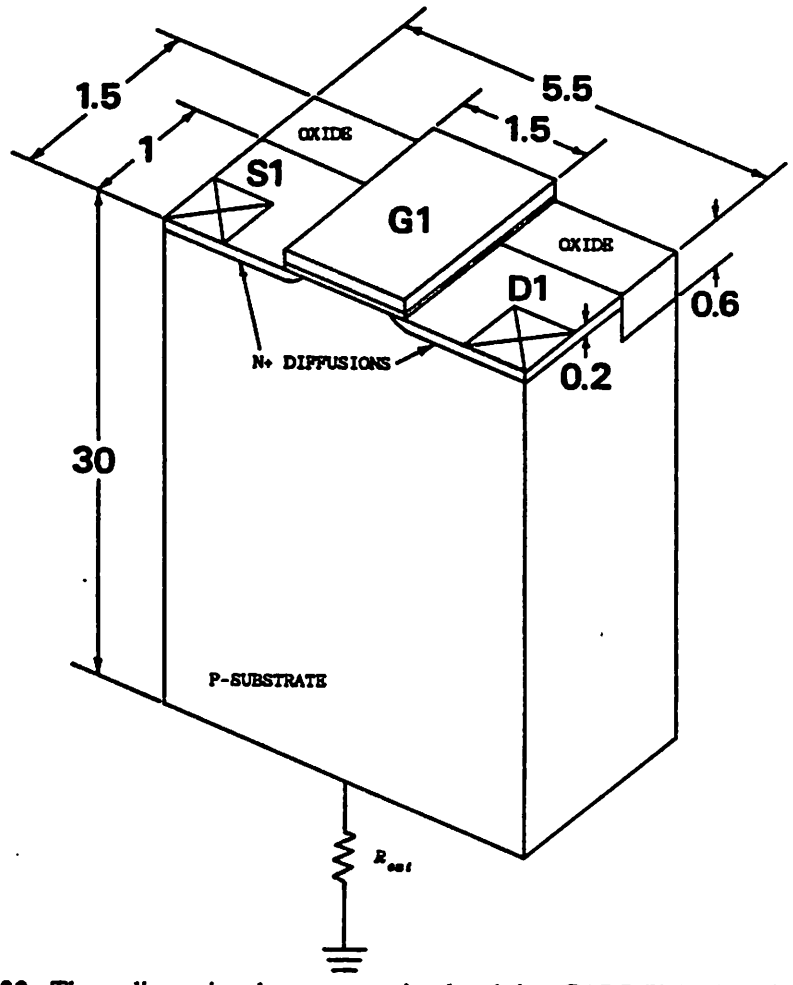


Fig. 5.22 Three-dimensional structure simulated by CADDETH (specifications listed in Table 5.2). All dimensions are in microns.

PHYSICAL PARAMETERS OF THE 3-D SIMULATED NMOSFET	
Gate Length (L_G)	1.5μm
Effective Channel Length (L_{eff})	1.1μm
Mask Channel Width (W_M)	1.5μm
Effective Channel Width (W_{eff})	1.5μm
Substrate Thickness (d)	30μm
Substrate Doping (N_{sub})	10^{17} cm^{-3}
Gate Oxide Thickness (t_{ox})	200A
Source/Drain Junction Depth (x_j)	0.2μm
Source/Drain Doping Distribution	Gaussian
Source/Drain Peak Doping (n_p)	2 x 10^{20} cm^{-3}
Depth of Source/Drain Peak Doping (R_p)	0μm

Table 5.2 Physical and structural parameters for the three-dimensional NMOSFET CADDETH simulations.

The waveform applied to the NMOSFET structure is slightly different from that applied to the two-dimensional structure and is shown in Fig. 5.23. Maximum gate voltage was kept at 5V, while maximum starting drain voltage was varied to see the effect of changing this parameter on the transient delay. The opposite waveform, that of the input falling from high to low, was not simulated since inverter propagation delay minimizes substrate current flow in this case [Hsu85].

Fig. 5.24 shows the simulated I_{sub} versus time behavior for the three-dimensional structure of Fig. 5.22. Again, the substrate current first peaks during the inverter transition, then drops down to a fairly flat plateau from 19ns to 60ns or 125ns depending upon the initial drain voltage value, then suddenly decreases in value. Figs. 5.25 - 5.28 show two-dimensional mountain plots of the hole concentration in the substrate during the various stages of charge accumulation and drainage. Note that initially there is a larger concentration of holes near the source (the left n+ region) since junction turn-on occurs at the source/substrate junction. As time passes, the holes diffuse from the area of generation until no more excess hole charge exists at t = 150ns.

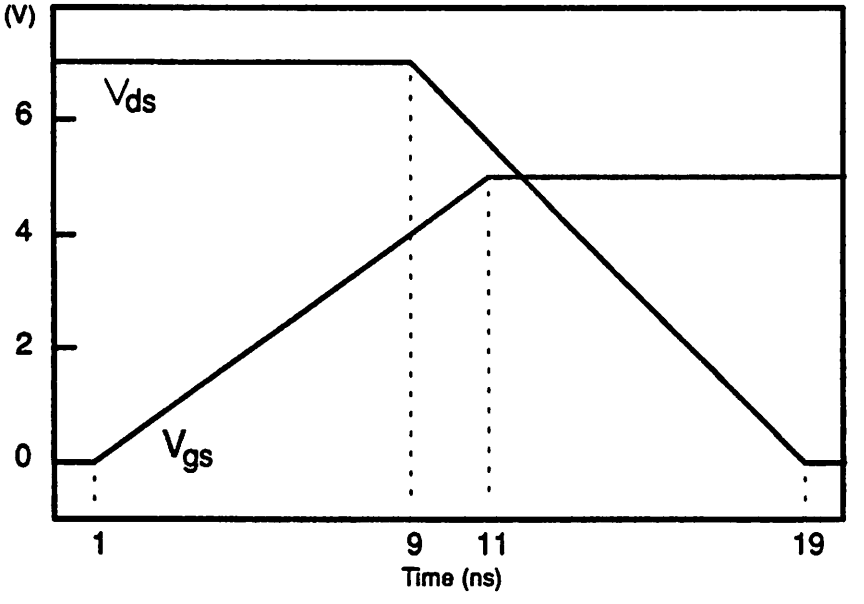


Fig. 5.23 Voltage waveforms applied to the NMOSFET device to simulate inverter behavior for the 3-D CADDETH simulations. Starting V_{ds} value was varied to see the effects on transient delay.

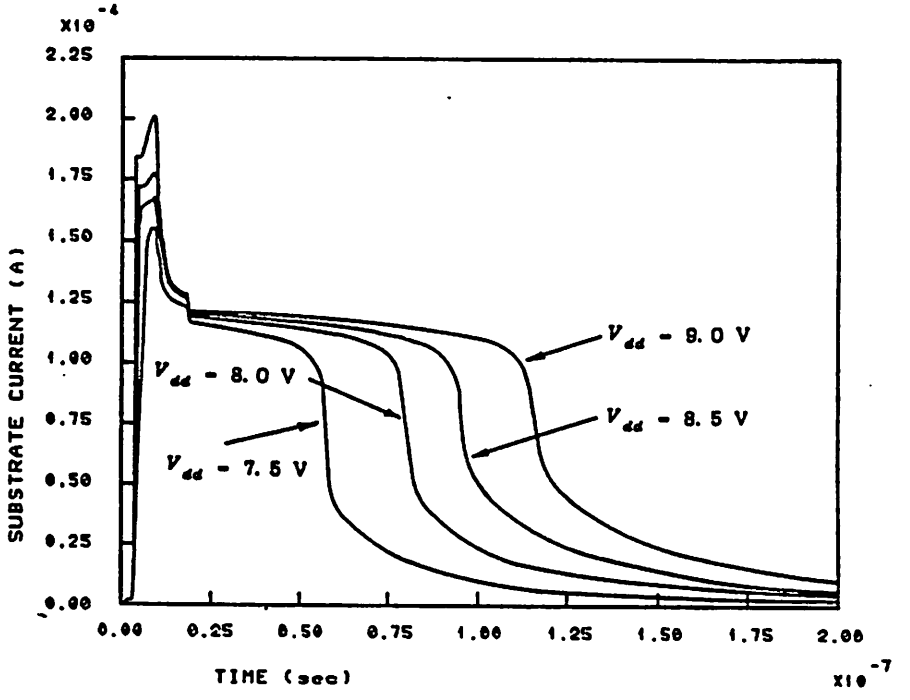


Fig. 5.24 CADDETH-simulated I_{sub} versus time behavior for the three-dimensional structure of Fig. 5.22.

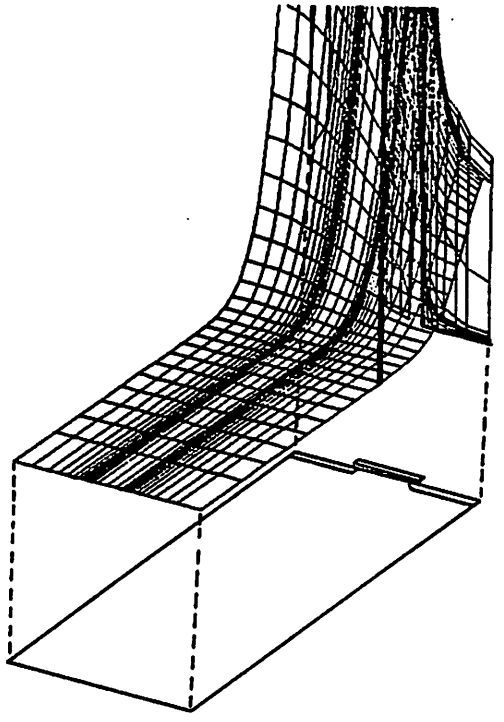


Fig. 5.25 Hole concentration in the substrate at t = 9ns.

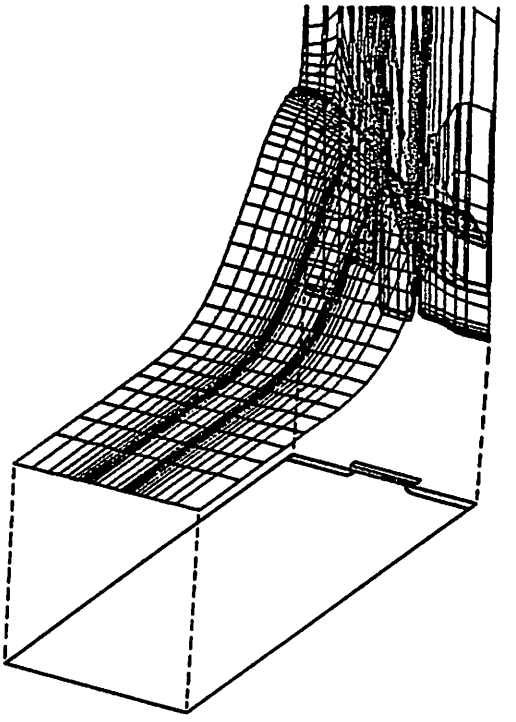


Fig. 5.26 Hole concentration in the substrate at t = 19ns.

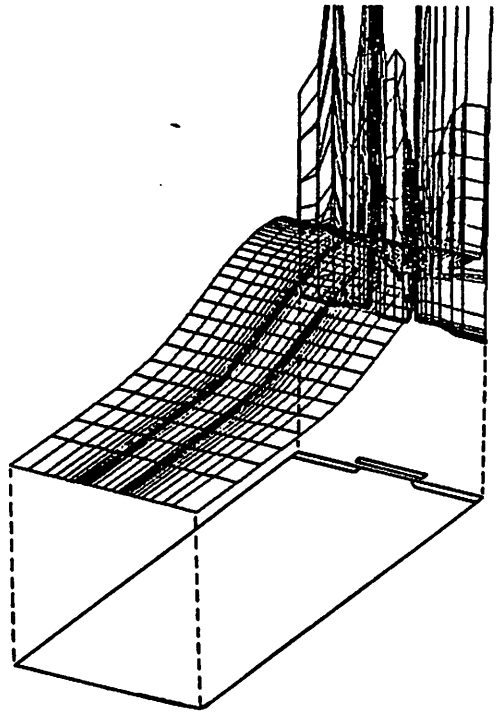


Fig. 5.27 Hole concentration in the substrate at t = 40ns.

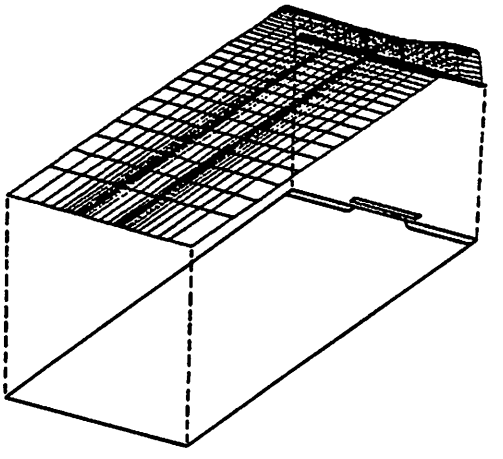


Fig. 5.28 Hole concentration in the substrate at t = 150ns.

Figs. 5.29 - 5.32 show the same hole concentration in the substrate but three-dimensionally using contour plots. Again, the higher hole concentration can initially be seen to be near the source, while as time passes the concentration becomes more evenly distributed.

Fig. 5.33 shows a two-dimensional mountain plot of the electric potential in the substrate, again illustrating the unchanging potential during the I_{sub} plateau for t = 19ns - 40ns.

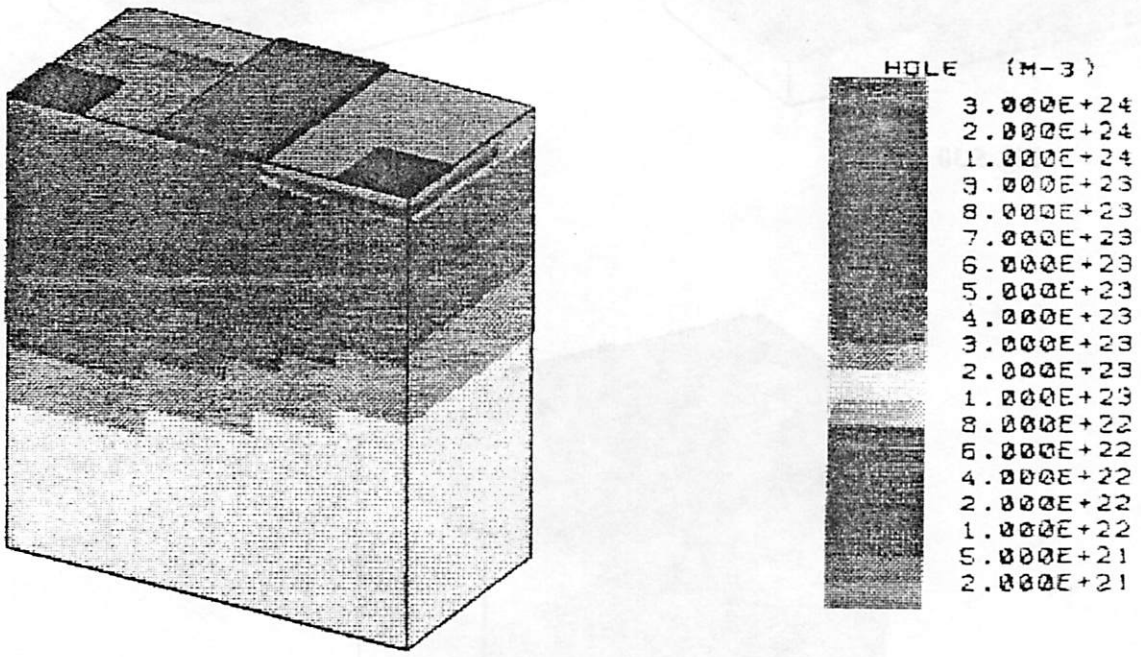


Fig. 5.29 t = 9ns.

Figs. 5.29 - 5.32 Three-dimensional contour plots of the hole concentration in the substrate for V_{dd} = 8V. Concentration values given in the legend are in units of m^{-3}.

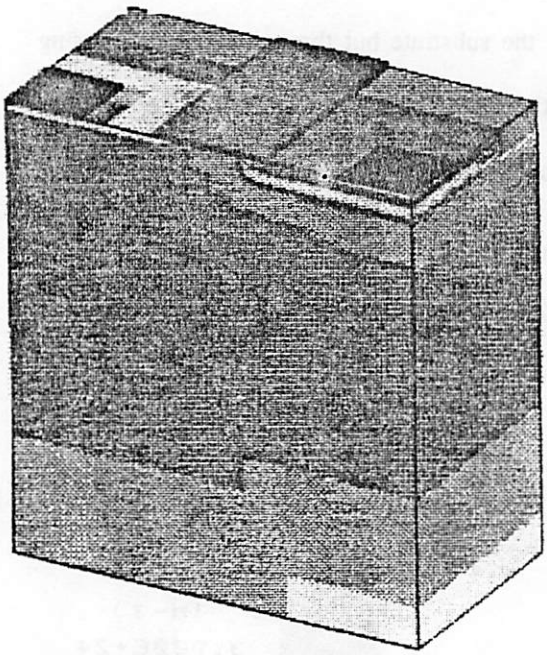


Fig. 5.30 t = 19ns.

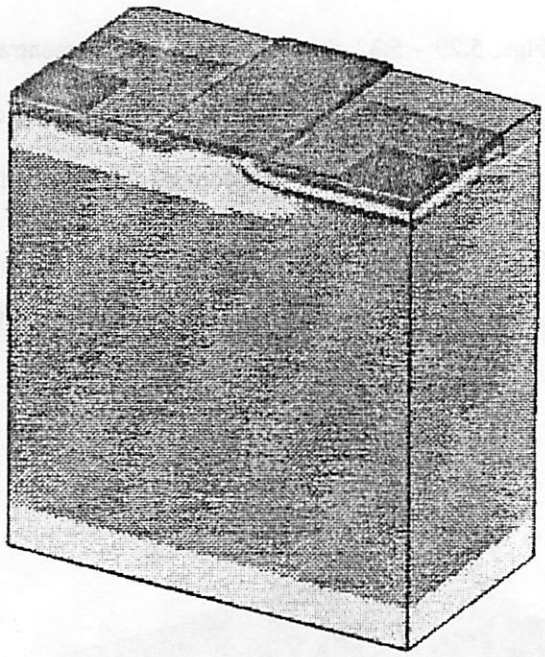


Fig. 5.31 t = 40ns.

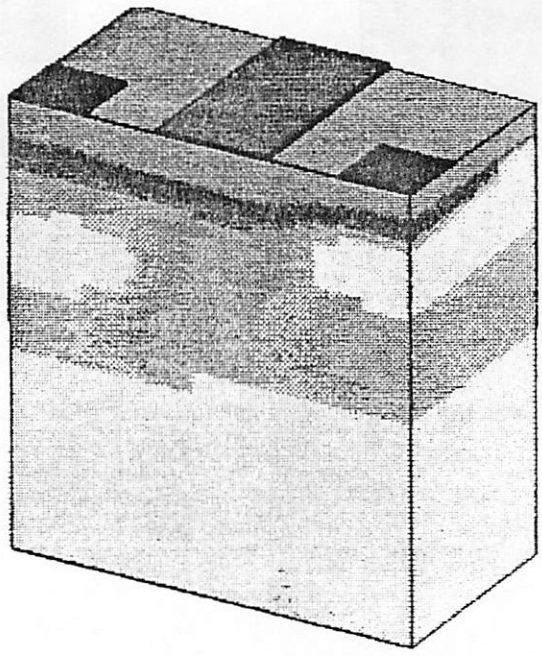


Fig. 5.32 t = 150ns.

Figs. 5.29 - 5.32 Three-dimensional contour plots of the hole concentration in the substrate for V_{dd} = 8V (continued).

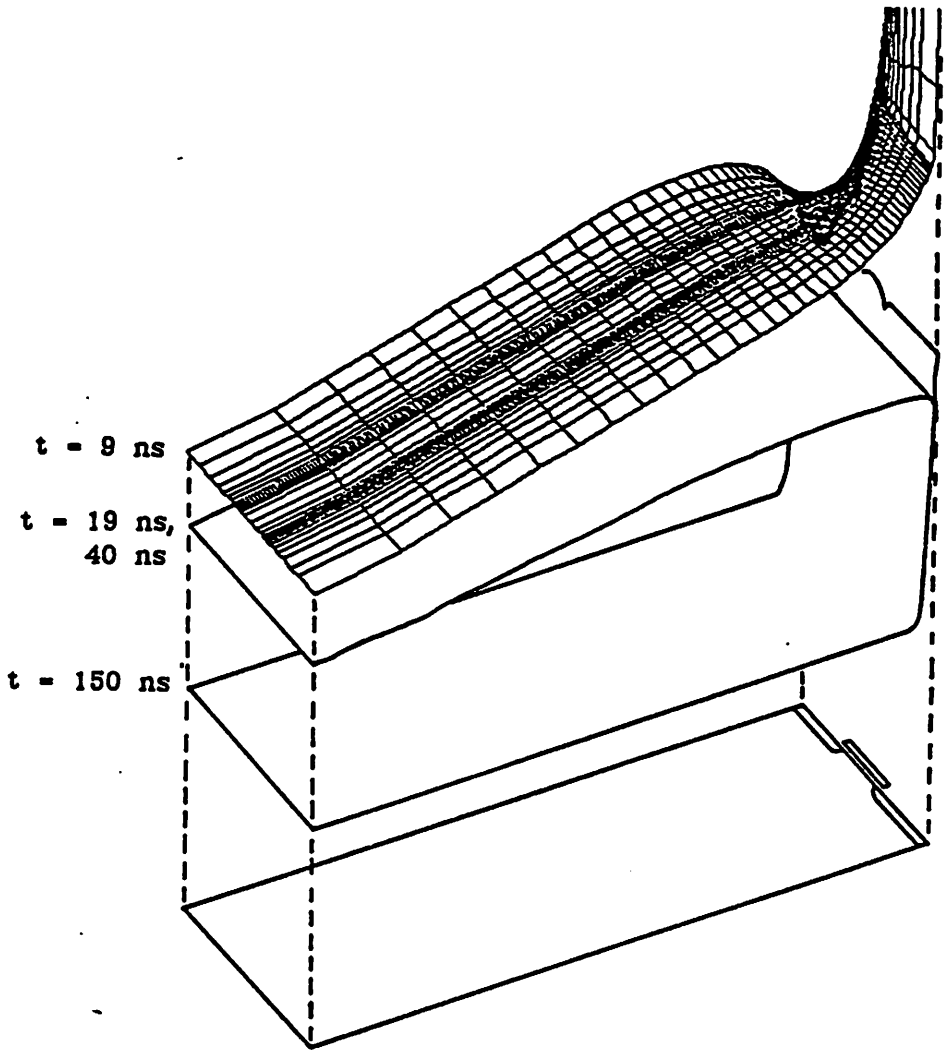


Fig. 5.33 Overlapped bird's eye view mountain plots of the electric potential in the substrate for $V_{dd} = 8V$.

5.3.4 Comparison With Measurement Results

A rough comparison was also made between measurement and CADDETH-simulated results. The two-dimensional structure shown in Fig. 5.34 and similar to the measured NMOS device was used for the CADDETH simulations. All doping profiles were calculated using a process simulator, and the distance between the NMOS transistor and the p-well contact was visually measured using a Nikon HFX-II microscope. All simulations were done in two-dimensions so that a finer mesh could be used. The input and output voltages were changed to approximately match measured characteristics. V_{in} was changed from 0 to 5V in 10ns, while V_{out} was lowered from V_{dd} to 0V in 10ns, with an 8ns delay between the start of the V_{in} change and the V_{out} change.

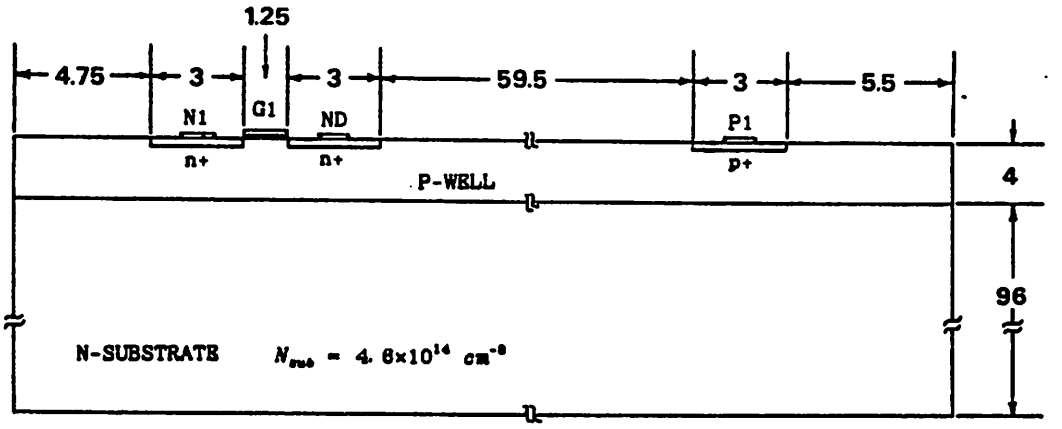


Fig. 5.34 Two-dimensional structure simulated by CADDETH to match the experimentally measured device. All doping profiles calculated by a process simulator. All dimensions are in microns.

Fig. 5.35 shows the simulated result for V_{dd} = 8V. To compare this with measured results, it is necessary to add the parasitic capacitances and resistances of the measurement apparatus (Fig. 5.2). Thus, the CADDETH-simulated substrate current was transposed point by point into a circuit simulator, with the parasitic elements included in the input deck. Fig. 5.36 shows the circuit-simulated result (dashed line), plotted with the measured results. As can be seen, the shape of the CADDETH-simulated result is similar to that of the measured result, except for the current levels. Because of the many non-uniform process-related factors involved, a fairly extensive analysis would be necessary to accurately match the simulation to the actual device.

5.3.5 Summary

The mechanisms of substrate current transient delay have been studied from two- and three-dimensional device simulation using CADDETH. A comparison was made between the simulated and measured results. Qualitative agreement can be achieved once parasitic elements of the measurement apparatus are added to the simulation. Better and more thoughtful parameter matching may eliminate the current level discrepancy between measurement and simulation.

The next section will introduce a model to predict the substrate current delay time τ_d.

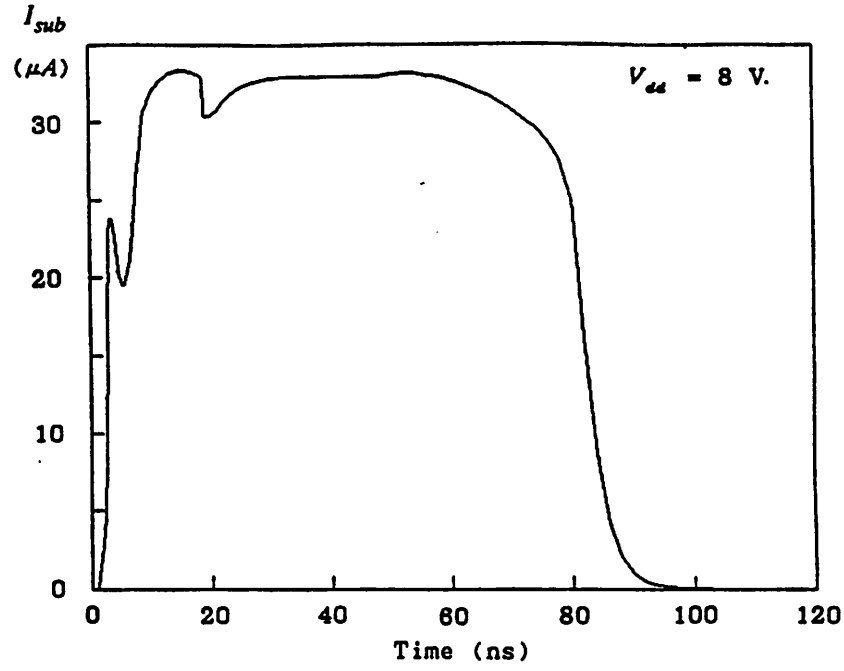


Fig. 5.35 CADDETH-simulated I_{sub} versus time of the structure shown in Fig. 5.34.

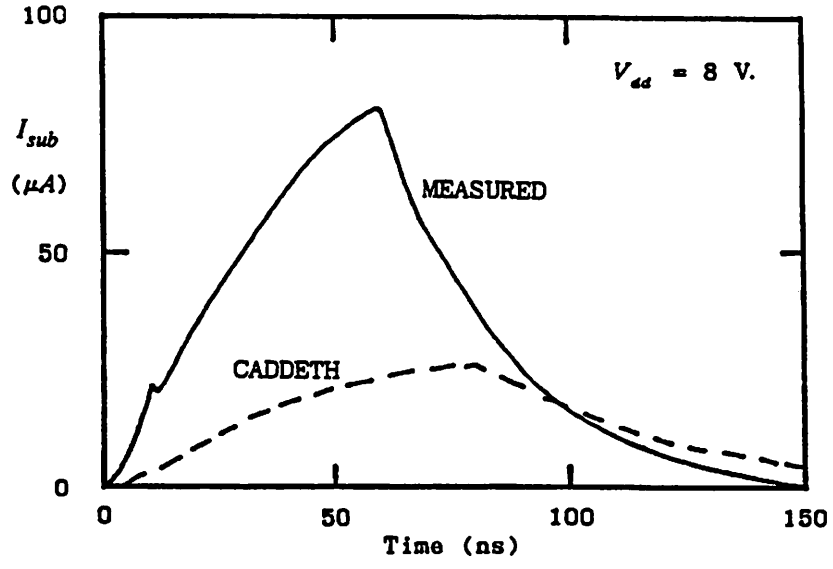


Fig. 5.36 A comparison of the measured and CADDETH-simulated results after parasitic elements of Fig. 5.1 are added to the simulation results.

5.4 Charge Storage Model

Predicting substrate current transient delay time, because it occurs beyond avalanche breakdown, is a difficult task. Several mechanisms, some related in a complex manner, affect the length of the delay. In this section, a general physical model will be presented which is applicable to voltages past DC avalanche breakdown. Fitting parameters are reduced to a minimum to maintain the physicalness of the model.

The transient delay model presented here is divided into two parts. The first deals with the mechanisms responsible for creating the accumulated charge in the substrate, while the second portion treats the mechanisms responsible for eliminating this stored charge. The delay time is then determined from the amount of accumulated charge calculated in the first part and from the rate of charge elimination found in the second portion.

5.4.1 Charge Accumulation

The avalanche breakdown model presented here is an extension of that developed by F.-C. Hsu [Hsu83]. Hsu treats the avalanching regime up to actual DC breakdown, but not beyond, where charge storage occurs. This model attempts to describe this charge accumulation regime beyond the point of normal DC breakdown. For a more detailed derivation of the basic avalanche breakdown model itself, the reader is encouraged to refer to Hsu's paper. In this section, only an overview will be given to facilitate the explanation of the extension.

Avalanching current is generated when the source-substrate junction becomes forward-biased because of current flowing through the resistive substrate. Whether the device suffers actual breakdown depends on two key requirements that must both be satisfied [Hsu83]: 1) the source-substrate junction must be forward-biased, and 2) a positive feedback condition must exist between the forward-bias current and the additional impact ionization it generates. Depending upon the resistivity of the substrate, either 1) or 2) can be the determining factor of whether breakdown occurs. In the following analysis, it is

assumed that the substrate is fairly resistive so that condition 1) is satisfied first, making condition 2) the limiting factor in determining breakdown.

Figs. 5.37 and 5.38 show the situation when the source-substrate junction is forward biased but current levels are not large enough to create a positive feedback condition. I_d represents the normal drain current that flows by MOS transistor action, while I_2 and I_3 are the electron and hole forward-bias currents respectively. kI_2 represents the fraction of the electrons injected from the forward-biased junction that travels through the high-field region near the drain and experiences impact ionization with multiplication factor M. From basic pn junction theory, an expression for I_{sub} can be derived from Fig. 5.38 and the following relationships [Mul77]:

$$I_1 = I_d + kI_2$$

$$I_2 = A_{sd}\, q\, n_i^2 \left(\frac{D_n}{N_{sub}\, L_n}\right) [\exp(V_j/V_t) - 1] = I_{es}[\exp(V_j/V_t) - 1]$$

$$I_3 = A_{sd}\, q\, n_i^2 \left(\frac{D_p}{N_d\, x_j}\right) [\exp(V_j/V_t) - 1] = I_{hs}[\exp(V_j/V_t) - 1]$$

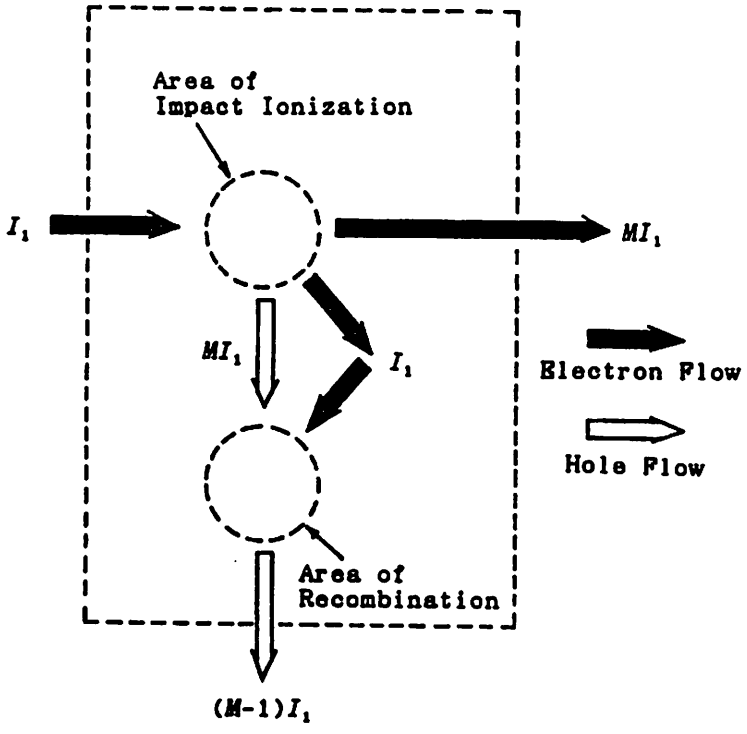


Fig. 5.37 High-field impact ionization region.

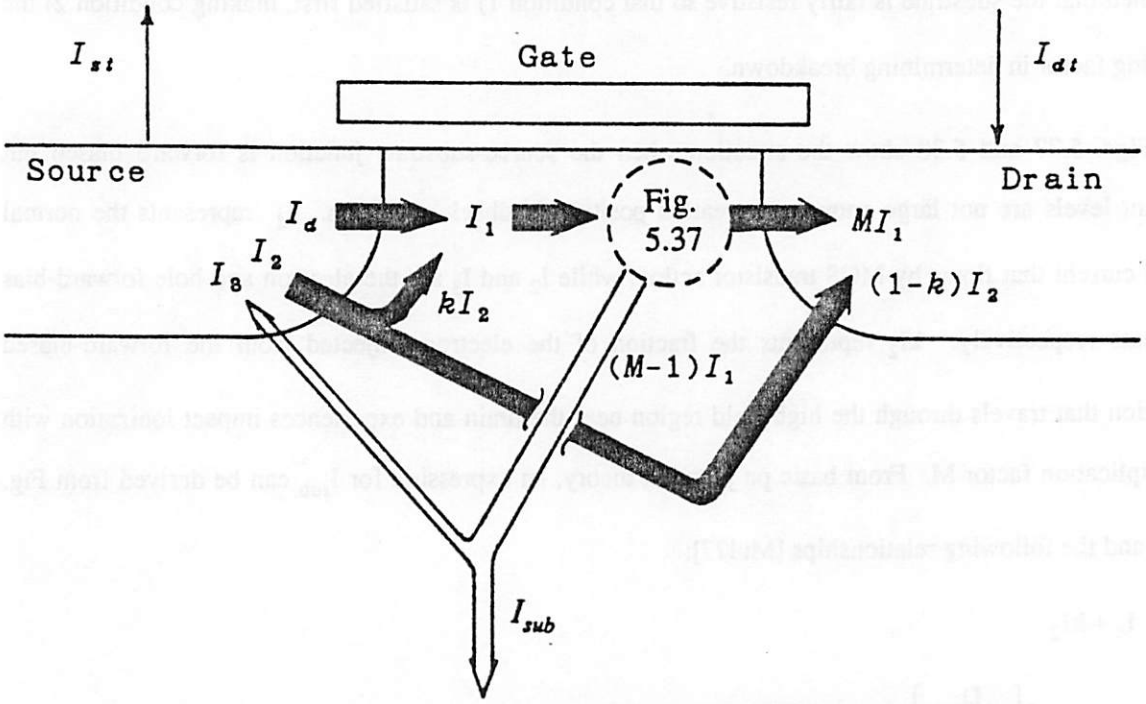


Fig. 5.38 Current flow after source-substrate junction forward biases but before positive feedback occurs. No charge accumulation.

$$M = \frac{1}{1 - \int \alpha_n \, dx}$$

A_{sd} and x_j are the cross-sectional area and junction depth of the source/drain regions, D_n and L_n are the electron diffusion constant and diffusion length in the substrate, D_p is the hole diffusion constant in the source, N_d is the doping concentration in the source, V_j is the actual voltage across the source-substrate junction, $V_t = k_B T/q$ is the thermal voltage, and α_n is the impact ionization coefficient. Although high level injection conditions exist, low level injection expressions for I_2 and I_3 are used as an approximation to simplify the analysis. Because the source/drain diffusions are very shallow, the short-base diode expression (using junction depth rather than diffusion length) is used for I_3. An effective doping value for N_d can be used, such as the average doping, to represent the non-uniform source/drain doping profile. M can be derived using a unified model for hot electron currents [Ko81],

$$M = \frac{1}{\frac{1 - A_i l_c E_d}{B_i} \exp\left(-\frac{B_i}{E_d}\right)}$$

where A_i and B_i are the impact ionization coefficients in

$$\alpha_n = A_i \exp\left(-\frac{B_i}{E_c}\right)$$

with E_c being the critical field necessary for impact ionization, and l_c can be described as an effective width of the "pinch-off" region of the channel and can be approximately formulated by [Sod84]

$$l_c \cong \left(\frac{t_{ox} x_j}{3}\right)^{1/2}$$

or extracted as a device parameter (see Chapter 2). E_d is the electric field at the drain end and can be formulated by [Sod84]

$$E_d = \frac{V_{ds} - V_{dsat}}{l_c}$$

where V_{dsat} can also be determined from extracted device parameters (see Chapter 2). By combining the above equations, M can be expressed by

$$M = \frac{1}{1 - \frac{A_i}{B_i}(V_{ds} - V_{dsat}) \exp\left(-\frac{B_i l_c}{V_{ds} - V_{dsat}}\right)} \tag{5.4.1}$$

By algebraic manipulation, I_{sub} is given by

$$I_{sub} = (M - 1)I_1 - I_3$$

$$= (M - 1)I_d + ((M - 1)kI_{es} - I_{hs})[\exp(V_j/V_t) - 1] \tag{5.4.2}$$

The first term of Eq. 5.4.2, $(M-1)I_d$, is the substrate current generated purely from the normal drain current. Denoting this current as I_{subN}, we can find this current from previously derived models dealing with the substrate current in the non-avalanching normal operating regime of the transistor [Sod84,Cha84],

$$I_{subN} = \frac{A_i}{B_i} I_d (V_{ds} - V_{dsat}) \exp\left[-\frac{B_i l_c}{V_{ds} - V_{dsat}}\right] \tag{5.4.3}$$

The second remaining term of Eq. 5.4.2, hereafter denoted by I_{subA}, is the substrate current generated or deducted by the forward-bias currents of the source-substrate junction either directly or through additional impact ionization.

By rearranging Eq. 5.4.2, k can be found:

$$k = \frac{I_{sub} - (M-1)I_d + I_{hs}[\exp(V_j/V_t) - 1]}{(M-1)\, I_{es}[\exp(V_j/V_t) - 1]} \tag{5.4.4}$$

I_d in this case could be determined by extrapolating the I_d - V_{ds} curve from the normal flat saturation regime into the avalanching regime. Neglecting body bias and external resistances, V_j is given by

$$V_j = I_{sub} R_{sub} \tag{5.4.5}$$

where R_{sub} is the substrate spreading resistance. By measuring the substrate current, k can be found using the value for V_j determined in Eq. 5.4.5.

Now, if the device is biased further into the avalanching regime, there will be a point where the increase in V_j caused by the increase in I_{sub} will generate enough injected electrons from the source to create more impact ionization for a positive feedback effect, leading to device breakdown in DC conditions. In transient conditions, however, this situation results in an accumulation of charge near the surface of the substrate, as shown in Fig. 5.39. This charge accumulation occurs because of the fact that I_{sub} cannot increase to compensate for the additional impact ionization current because it is effectively pinned by the forward bias voltage V_j (current flow into the source has an exponential dependence on V_j, while I_{sub} from Eq. 5.4.5 has only a linear dependence). Thus, we need to introduce another factor, k_s, shown in Fig. 5.39, to account for the current that flows into this stored charge region. I_4 is the current that actually causes the hole charge storage, but to maintain charge neutrality (high injection conditions exist), and equal amount of electrons must flow into the stored charge area. This current can be represented by a fraction of the total injected electron current I_2 by

$$I_4 = k_s (I_2 - I_{2o}) = k_s I_{es}[\exp(V_j/V_t) - \exp(V_{jo}/V_t)]$$

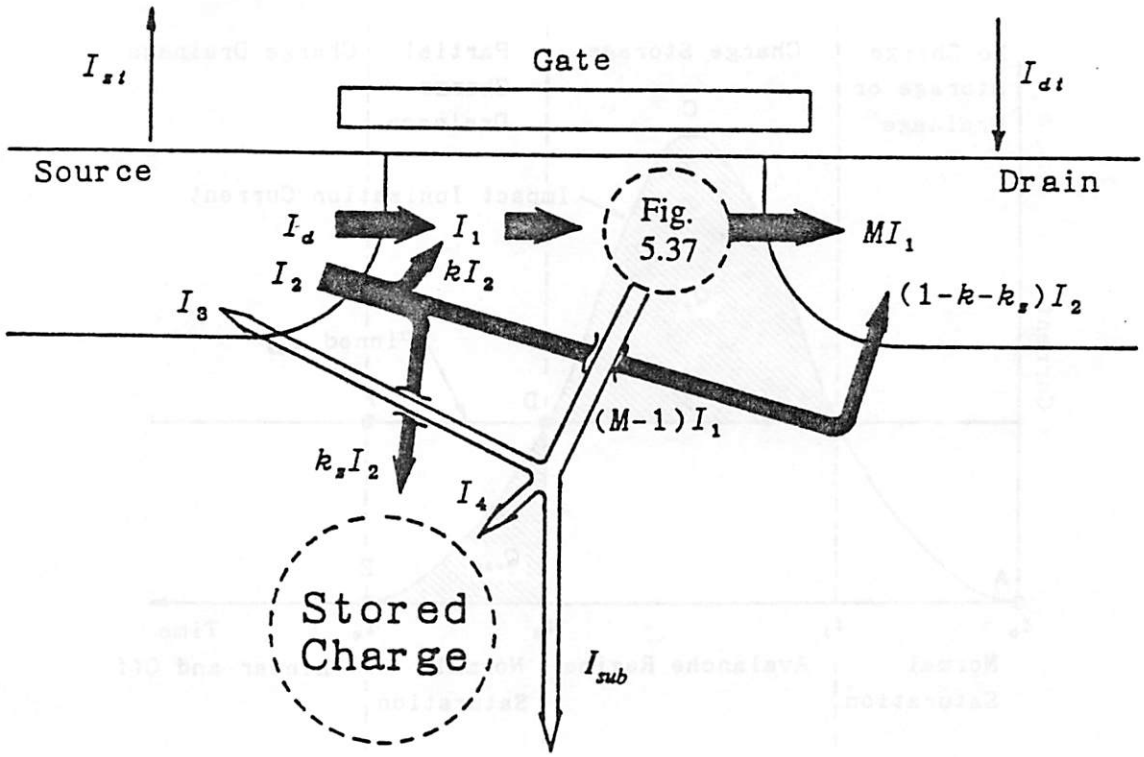


Fig. 5.39 Current flow after entering avalanche breakdown regime where both the source-substrate junction is forward biased and positive feedback is occurring. Charge accumulation occurs.

where I_{2o} and V_{jo} represent I_2 current and local junction potential when the device first enters the charge storage regime. From straightforward algebraic manipulation, I_{sub} can then be found:

$$I_{sub} = (M\text{-}1)I_1 - I_3 - I_4$$

$$= (M\text{-}1)I_d + [((M\text{-}1)k - k_s)I_{es} - I_{hs}][\exp(V_j/V_t) - 1] \qquad (5.4.6)$$

The two-term form is very similar to Eq. 5.4.2, except for the new constant k_s . This new variable takes into account the limiting effect on the substrate current flow caused by the source-substrate junction voltage in the transient case. k_s is a very difficult parameter to extract, but it will be shown that it is actually not needed for time delay calculations.

A graphical plot summarizing the different regions of operation is shown in Fig. 5.40 in the case when a low-to-high gate voltage and high-to-low drain voltage transition is experienced as is the case for a typical inverter. The curve ABCDE is the current generated by impact ionization, the solid line BDF

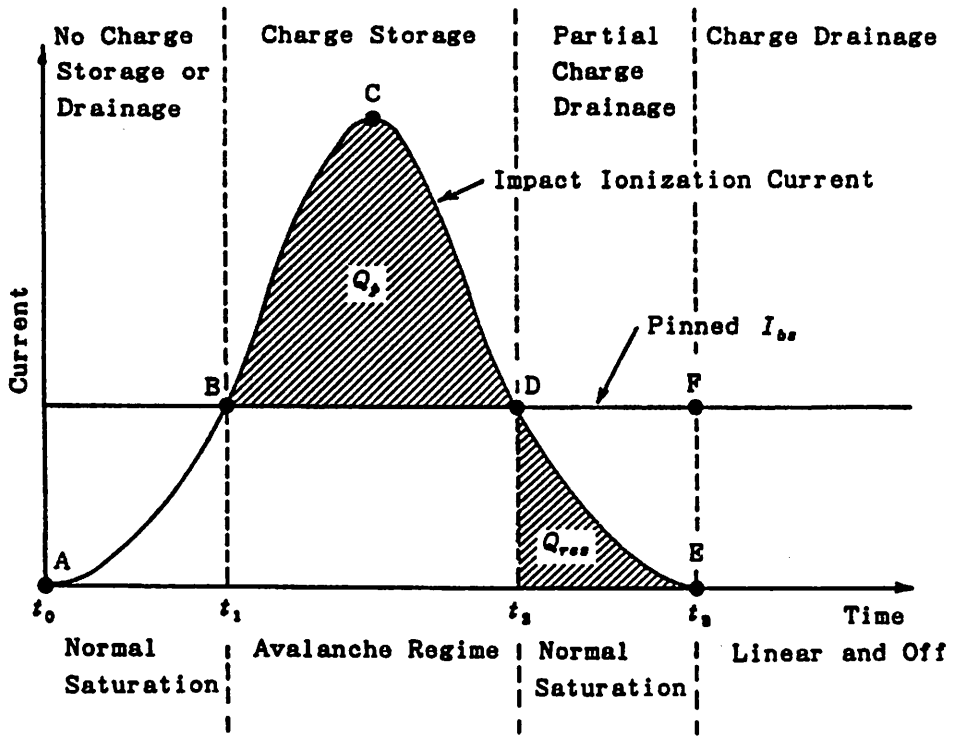


Fig. 5.40 Graphical schematic plot of current flow in the substrate versus time illustrating where charge storage and drainage takes place.

represents the pinned substrate current, and the curve ABDF is the actual I_{sub} that flows out of the contact. Although the varying current I_3 should also be included with the pinned substrate current during $t_1 < t < t_2$ (line BD), because the n+ doping of the source is typically much higher than the substrate doping, I_3 is assumed to be small, especially when compared to I_{sub}. The time period $t_0 < t < t_1$ represents normal saturation, $t_1 < t < t_2$ represents charge accumulation occurring in the avalanching regime, $t_2 < t < t_3$ represents partial charge drainage after leaving the avalanche regime (some of the charge drained away by I_{sub} is replaced by the still present impact ionization Q_{res}), and $t > t_3$ represents the remaining time period where charge is draining away when the transistor first enters the linear region and then turns off. t_0 through t_3 can all be determined by knowing the terminal voltages of the NMOSFET and the normal drain and substrate current parameters, including k.

Note that in the case where (M-1) I_1 (the impact ionization current of Fig. 5.40) varies from changing bias conditions, most of the change is reflected through I_3 and I_4, with a corresponding change in $k_s I_2$. Thus, although I_{sub} is effectively pinned by the source-substrate forward bias voltage V_j , this voltage

actually fluctuates slightly to accommodate the variations in I_2 and I_3. Because V_j has a natural logarithmic dependence on I_2 and I_3, for simplicity we can make an approximation that V_j is pinned to a constant voltage V_{jp} when the junction is forward biased. We can arbitrarily set V_{jp} as the junction potential when the transistor first enters the charge storage region.

To find this transition point, we can perform an analysis of the currents I_1, I_2, I_3, and I_{sub} of Fig. 5.38 and determine when positive feedback occurs. This positive feedback effect can be said to occur when an increase in currents, resulting in an increase in V_j, cannot be handled by the substrate current. From Eq. 5.4.2 and 5.4.5,

$$I_{sub} \equiv \frac{V_j}{R_{sub}} \equiv (M-1)I_d + [(M-1)kI_{es} - I_{hs}][\exp(V_j/V_t) - 1] \tag{5.4.7}$$

this is equivalent to saying that when the terms on the right-hand side (RHS), which represents the impact ionization current, increases more quickly than the left-hand side (LHS) with increasing V_j, then breakdown occurs. By taking derivatives of each side, we obtain

$$\frac{dLHS}{dV_j} = \frac{dI_{sub}}{dV_j} = \frac{1}{R_{sub}}, \qquad \frac{dRHS}{dV_j} \equiv \frac{1}{V_t}[(M-1)kI_{es} - I_{hs}]\exp(V_j/V_t)$$

The V_j dependence of M and I_d have been neglected on the right-hand side of the equation since the exponential term dominates. The condition for breakdown can then be stated as the point where $dRHS/dV_j > dLHS/dV_j$, when the exponential increases faster than the substrate current. Re-writing the above expression, this condition can be mathematically expressed as

$$V_j > V_t \ln\left(\frac{V_t}{R_{sub}[(M-1)kI_{es} - I_{hs}]}\right) \tag{5.4.8}$$

Since the multiplication factor M varies with bias condition, V_{jp} cannot be found from Eq. 5.4.8 alone. M and V_j must both be determined at the same bias point by using Eqs. 5.4.1 and 5.4.7, then compared in Eq. 5.4.8. If the condition is not satisfied, then the bias condition should be changed to increase M until the inequality of Eq. 5.4.8 is satisfied. At this point,

$$V_{jp} = V_t \ln\left(\frac{V_t}{R_{sub}[(M_p-1)kI_{es} - I_{hs}]}\right) \tag{5.4.9}$$

where M_p is the multiplication factor M at the transition point.

Finally, we need to find the amount of charge that accumulates within the region near the surface of the substrate before substantial recombination takes place. From Fig. 5.39, the current that supplies this stored charge has a magnitude of $I_4 = (M-1)I_1 - I_{sub} - I_3$, and the amount of charge that accumulates is then given by this current times the duration of the current t_{ov} (= $t_3 - t_1$ of Fig. 5.40). t_{ov} can be determined by the transient output characteristics by identifying the time interval where the device is biased in the breakdown regime. To find the charge per area, Q_p, we further need to determine an effective area A_{eff} through which this current flows. A good estimation would be the effective gate area, or $A_{eff} = W L_{eff}$. Thus, the total charge accumulation can be given by

$$Q_p \cong \frac{1}{qA_{eff}} \int_{t_1}^{t_3} I_4 \, dt$$

$$= \frac{1}{qA_{eff}} \left\{ \int_{t_1}^{t_3} (M-1) I_d \, dt - \frac{V_{jp}}{R_{sub}} (t_3 - t_1) + \left[I_{es} \int_{t_1}^{t_3} (M-1) \, dt - I_{hs} \right] [\exp(V_j/V_t) - 1] \right\}$$

Note that the integrand of the first integral is the normal substrate current I_{subN} in absence of the forward bias currents associated with avalanche breakdown. Thus, by incorporating that model and the expression previously derived in this section for M, Q_p can be expressed as

$$Q_p \cong \frac{1}{q A_{eff}} \left\{ \frac{A_i}{B_i} \int_{t_1}^{t_2} (V_{ds} - V_{dsat}) \exp\left(- \frac{B_i I_c}{V_{ds} - V_{dsat}} \right) dt - \frac{V_{jp}}{R_{sub}} (t_2 - t_1) \right.$$

$$\left. + \left[I_{es} \int_{t_1}^{t_2} \left[1 - \frac{A_i}{B_i} (V_{ds} - V_{dsat}) \exp\left(- \frac{B_i I_c}{V_{ds} - V_{dsat}} \right) \right]^{-1} dt - I_{hs} \right] [\exp(V_{jp}/V_t) - 1] \right\} \qquad (5.4.10)$$

Note that the partial charge drainage that occurs between t_2 and t_3 is automatically taken into account. By knowing the transient behavior of the terminal voltages, the integrals of Eq. 5.4.10 can be numerically integrated to find Q_p.

In summary, to find the accumulated charge Q_p, the following steps should be followed:

1) Extract normal drain current (I_d) parameters.

2) Extract normal substrate current parameters (which are identical to the parameters for M).

3) Extract the avalanche breakdown parameter k.

4) By iteration and using the parameters extracted above, find V_{jp}.

5) By inputting the actual transient voltage waveforms, calculate Q_p from Eq. 5.4.10.

Device parameters specific to the transistor can be extracted in (1) - (4) by a parameter extraction program, while (5) can be incorporated into circuit simulation with the charge drainage model that will be described next.

5.4.2 Charge Recombination and Drainage

We must now determine the mechanisms that eliminates the excess charge. The charge recombination and drainage effect is very similar to the Haynes-Shockley experiment with an applied field [Sze81], except with a crucial difference that on one side of the charge pulse is a forward-biased pn junction. In the normal pulse drift experiment, majority carriers are supplied by the high potential contact on one end of the silicon bar. Then from current continuity the amount of majority carriers supplied to the high field side of the pulse is equal to that leaving the pulse from the low field end (recall that carrier mobility is greater than the mobility of a charge disturbance because of ambipolar effects if the charge concentration is large). Charge depletion of the pulse would thus only be caused by recombination. But because in the present case there is no supply of majority carriers (holes) on the pn-junction side of the pulse, and because of the fact that maximum potential exists within the silicon sample and not at one of the contacts, the hole current flowing to the substrate contact is wholly supplied by the excess charge pulse. Thus, in addition to the normal recombination term, there is a substrate current term that needs to be considered when calculating charge drainage.

Let us denote the excess charge remaining at some time t as $Q_{pp}(t)$, to distinguish it from the total accumulated charge Q_p of the previous section. From Fig. 5.40, $Q_{pp}(t_2) = Q_p$, with charge

recombination and drainage occurring after t_2 . As mentioned in the previous section, the small amount of charge Q_{res} created by the still-saturated transistor between t_2 and t_3 is ignored. Now at some later time t' , we can approximately calculate the amount of charge remaining if dt = t' - t is small:

$$Q_{pp}(t') = Q_{pp}(t) \exp\left[-\frac{t'-t}{\tau_c}\right] - \frac{I_{sub}(t)}{q A_{eff}}(t'-t)$$

$$= Q_{pp}(t) \exp\left[-\frac{t'-t}{\tau_c}\right] - \frac{V_{jp}}{q A_{eff} R_{sub}}(t'-t) \qquad (5.4.11)$$

The first term is the normal recombination term, while the last term represents charge drained away by the substrate current plateau. Eq. 5.4.11 is a recursive relationship; thus by successively substituting the previous $Q_{pp}(t)$ value in the recombination term with the present one, charge remaining at the next time step can be determined. τ_d, the transient substrate current delay time, can then be found from

$$Q_{pp}(\tau_d) = 0 \qquad (5.4.12)$$

Because Eq. 5.4.11 is a recursive relation, an analytical expression cannot be found, but incorporation into computer simulation is a straightforward process.

5.4.3 Summary

A general physical model has been developed to predict the substrate current transient delay. Most of the needed parameters are easily extractable under DC conditions, with the exception of the avalanche parameter k. Because of the complexity of the charge storage phenomenon, many approximations had to be made, and the resulting expressions have to be iterated to find the delay time, but incorporation into computer simulation is a straightforward process.

5.5 Charge Buildup Effect From Periodic Signals

This section discusses the effect of clocked periodic signals on the substrate current transient delay that is seen. In a digital circuit environment, input signals consist mainly of periodic pulses with frequencies controlled by a clock circuit. Thus, a typical inverter will create packets of charge generated by the periodic impact ionization that occurs. Generally, substrate current generation is minimal during the high-to-low input voltage case since the overlap time during which the input and output voltages are nonzero is minimal as a consequence of the propagation delay of the inverter [Hsu85]. Thus, most of the charge generation will occur at the rising edge of the input, once per cycle. As long as charge accumulation at this input rising edge is small and the period of the signal is long, the small amount of charge that accumulates will be able to drain away, and the resulting disturbance will be minimal.

As clock frequency increases, however, the charge that accumulates during the input rising edge may not have a chance to disappear before the next input rising edge. As a result, charges may successively build with each new cycle and cause undesirable effects even if charge accumulation per cycle is small.

CADDETH was used to see the effect of applying a second low-to-high input pulse before the charge accumulated from the first pulse had a chance to disappear. The example used here is taken from a two-dimensional NMOS structure (Fig. 5.8) with a single pulse substrate current transient delay behavior for V_{dd} = 8 V shown in Fig. 5.10. Note that the delay is approximately 150ns long. Thus, if the next input rising edge occurs before the 150ns, charge buildup will result. Fig. 5.41 shows the two-pulse transient voltage waveforms applied on the device, while Fig. 5.42 shows the resulting substrate current flow. The lengthened delay is very noticeable, as now substrate current flows as long as 800ns past the first pulse compared to 150ns of flow that occurred with the first pulse alone.

An interesting side-effect of the charge accumulation can be seen from this plot. Because the excess charge creates a conducting path between the source and drain, drain current is seen to flow when the drain voltage is nonzero even if the gate voltage is zero. The drain current flow causes a resistive drop to occur between the source/drain contacts and the source/drain regions near the channel, enabling the local

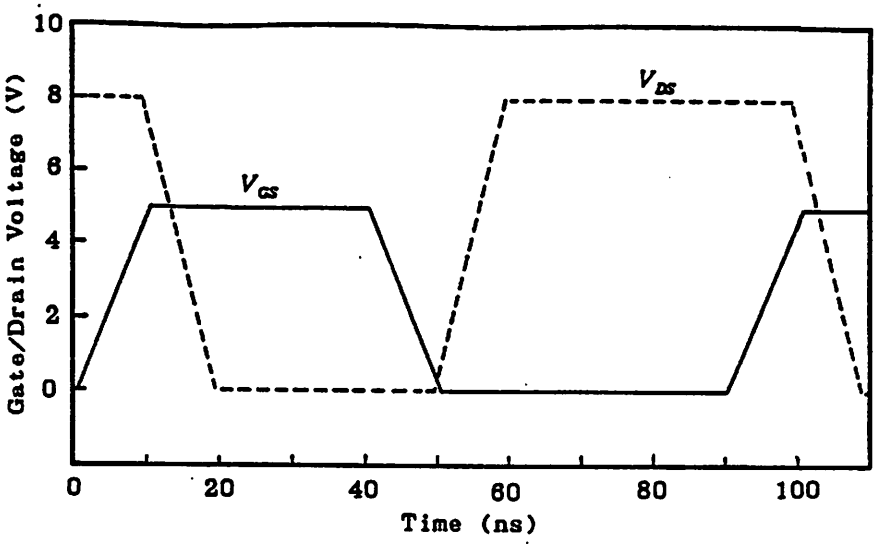


Fig. 5.41 The two-pulse gate and drain voltage waveform applied on the NMOS transistor.

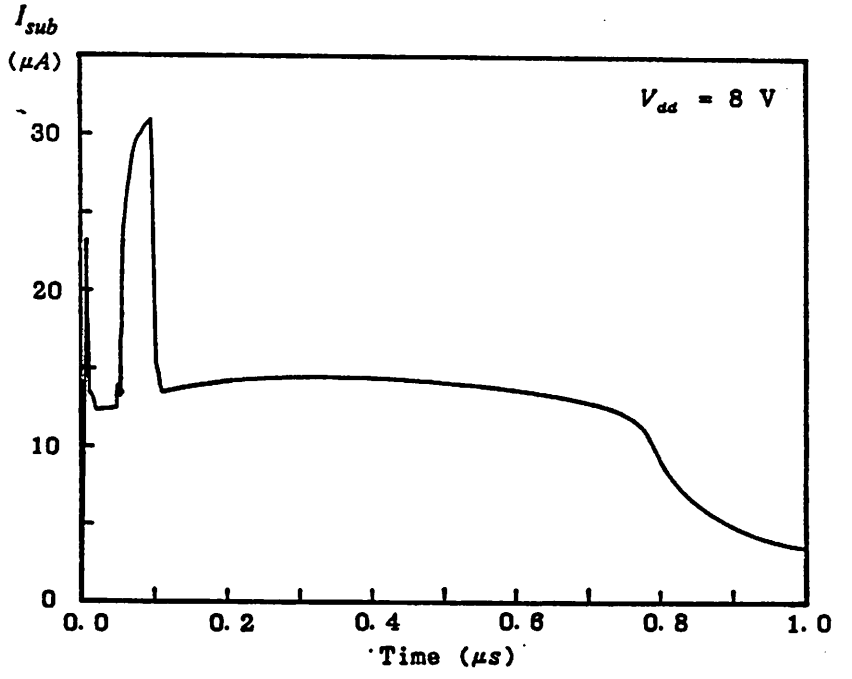


Fig. 5.42 I_{sub} versus time. Note that the delay is much longer than the single pulse case and that the second pulse is wider than the first because of source-drain current flow enabled by the accumulated charge.

potential of the substrate near the channel side of the source to increase past the pn forward-bias voltage. This enables a larger substrate current to flow, as explained in Section 5.3. Once the drain voltage drops to zero, no drain current flows and the local substrate potential near the channel side of the source is once again pinned to the forward-bias voltage, and the substrate current drops to its plateau level. Note that because additional impact ionization occurs during the entire duration when the drain voltage is high and the gate voltage is low, the length of the delay is greater than twice the one-pulse delay.

Fig. 5.43 shows the excess hole concentration underneath the source after the second transistor "on" period. The additional charge generated can be seen to add to the already present charge and form deeper into the substrate, as shown by the dotted curves. As time progresses, the excess charge diffuses out further into the substrate than during the single voltage pulse case. It can easily be perceived how successive application of gate pulses can add to this charge accumulation, resulting in adverse effects to

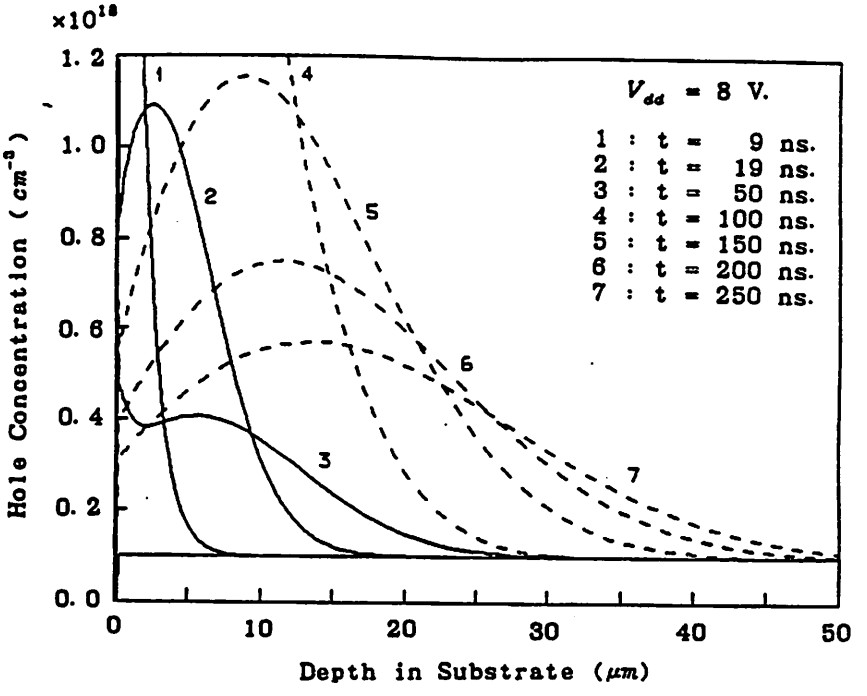


Fig. 5.43 Hole concentration versus depth into the substrate. The solid curves represent the charge diffusion after the first pulse but before the second, while the dotted curves show the behavior after the second pulse.

areas further away from the device, or eventual device breakdown. To prevent such a buildup, the frequency f must be such that the input high-to-low transition does not occur until after all the charge has had a chance to drain away. This translates to the following rule-of-thumb to prevent charge buildup,

$$f < \frac{1}{\tau_d} \tag{5.5.1}$$

where τ_d is the I_{sub} delay from one input low-to-high transition.

So far, we have pictured clocked digital signals as aggravating the breakdown condition (compared to the single pulse case). From another point of view, since bipolar charge storage occurs only past the point of DC avalanche breakdown, circuits operating with clocked waveforms may operate at a power supply voltage exceeding the DC avalanche breakdown voltage of the individual devices of the circuit. As the power supply voltage in increased, there will be a point where Eq. 5.5.1 no longer is satisfied, and breakdown even in the dynamic AC case will occur.

5.6 Implications of Charge Storage to CMOS Latchup Susceptibility

5.6.1 Introduction

Because the charge storage effect of this phenomenon can extend fairly deep into the substrate (on the order of 20μm in Figs. 5.12 and 5.13 of Section 5.3), any neighboring structures or circuitry can be affected adversely from the excess charge and currents generated by the NMOSFET. This section presents two-dimensional CADDETH simulation results demonstrating how CMOS latchup can occur when the charge-generating transistor is placed near a n-well structure.

5.6.2 Simulation Setup

Fig. 5.44 shows the two-dimensional structure used in the simulations. A NMOS structure dimensionally identical to the two-dimensional case simulated in Section 5.3 (Fig. 5.8) is placed on the left 5μm away from the n-well. N1, ND, G1, and P1 represent the source, drain, gate and topside substrate terminals of the NMOS device, respectively, while P2 represents the source terminal of the normally present PMOS device, and N2 represents the n-well contact. Field oxide 3000A in thickness is used to isolate the various diffusions near the surface. The p-substrate has a uniform doping of 5×10^{15} cm^{-3}, the n-well has a peak doping of 2×10^{16} cm^{-3} at the surface with well depth of 3 μm, and a boron channel implant with 10^{17} cm^{-3} peak doping at the surface exists for threshold voltage adjustment of the

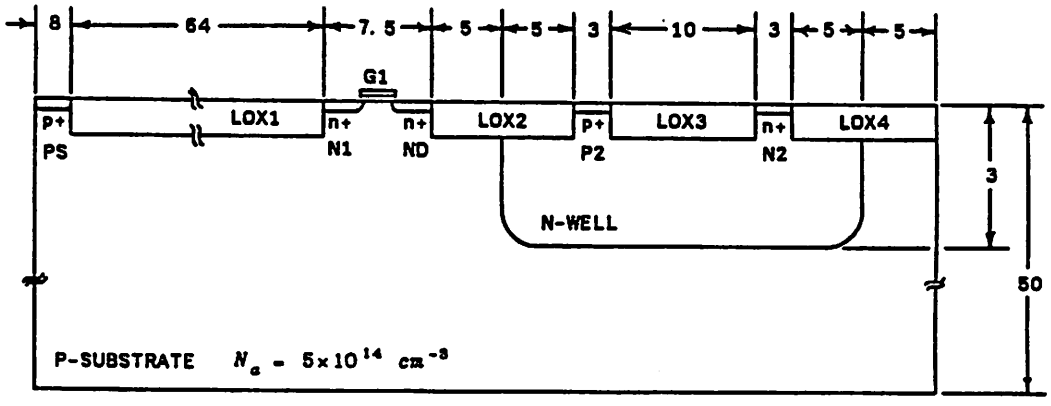


Fig. 5.44 The single-NMOSFET single-well structure simulated in this section to study latchup effects. All dimensions are in microns.

NMOS device. Simulated V_{th} of the device is 1V. Terminals P2 and N2 are tied to V_{dd} = 5V, and P1 and N1 are grounded. The gate bias on G1 and drain bias on ND are then changed in inverter-like fashion identical to that done in Section 5.3 for the three-dimensional device structure (Fig. 5.23).

5.6.3 Simulation Results

Fig. 5.45 shows the current versus time behavior for the PS (substrate), ND (NMOS drain), PS (n-well p+), and N2 (n-well) contacts. Positive current for I_{PS} denotes current flowing out of the PS contact, while positive current for the remaining terminals denote current flowing into the contact. Recall that all voltage transients stop and the NMOS turns off at t = 19ns (Fig. 5.23), but the substrate current I_{PS} still flows from charge drainage. As described in Section 5.2 and 5.3, because charge accumulation is a bipolar effect, excess electrons are also diffusing out and being collected by the positively-biased n-well, as seen by the current flow through I_{N2} . At approximately 5ns later, at t = 24ns, the current through I_{N2} is large enough to cause the P2 p+/n-well junction to forward bias. As a result, I_{P2} and I_{ND} suddenly

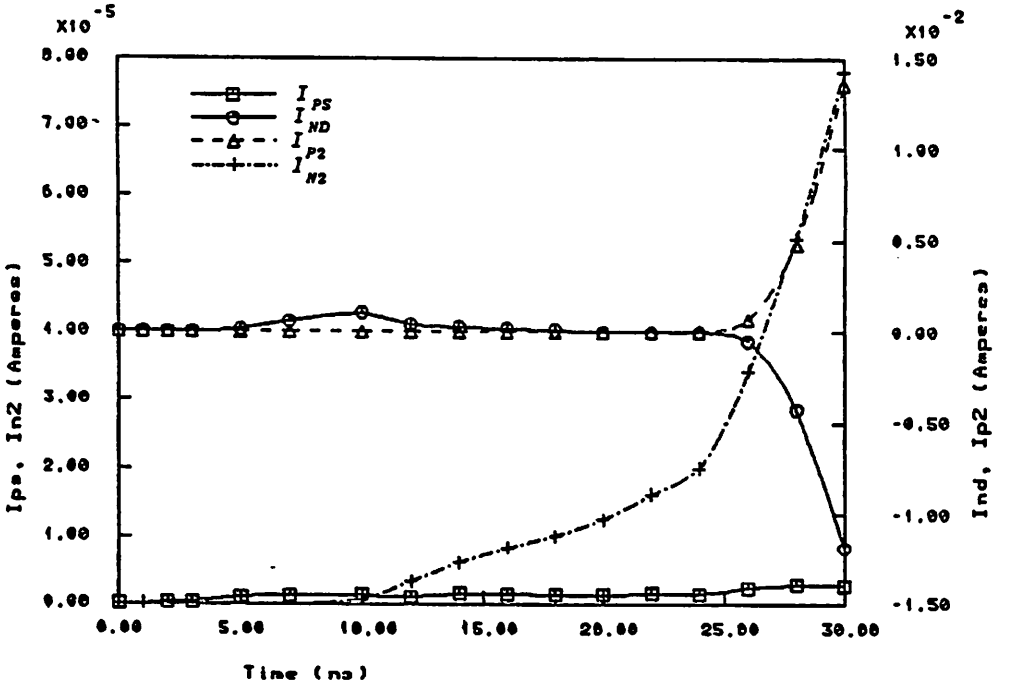


Fig. 5.45 The various terminal currents of the latchup structure versus time showing the latchup behavior. Note that latchup occurs after the NMOS transistor is off but during the I_{sub} plateau.

increase as the structure enters the latched regime.

Figs. 5.46 - 5.49 show bird's eye view mountain plots of the hole concentration in the substrate between the drain of the NMOS device (ND contact) and the p+ n-well contact (P2). At t = 9ns and 19ns, the n-well can still clearly be seen, but at t = 25ns, latchup can be seen to start as excess holes from the P2 contact flood the n-well as the p+/n-well contact forward biases. By t = 30ns, the n-well has all but disappeared and a smooth hole gradient can be seen instead.

Figs. 5.50 - 5.54 show two-dimensional contour plots of the hole concentration at the same time points, showing the successive stages of latchup from a different viewpoint. Fig. 5.50 (t = 0ns) shows the structure before the inverter pulse is applied. At t = 9ns and 19ns, the hole accumulation in the substrate can clearly be seen, spreading deep into the substrate, but there is no effect yet to the n-well (Figs. 5.51 and 5.52). At t = 25ns, however (Fig. 5.53), the P2 p+/n-well junction can be seen to forward bias as holes are injected into the n-well. The size of the n-well has also diminished somewhat by this point. By t = 30ns (Fig. 5.54), the only remnants of the n-well is near the N2 n-well contact; the left side of the well is totally flooded with holes. A direct coupling can now also be seen between the P2 well contact and the NMOS source and drain as the structure is now well into the state of latchup.

Thus, in summary, we have seen through two-dimensional simulation that the substrate charge storage effect can readily cause CMOS structures to latch. More importantly, it was demonstrated that latchup can occur after the NMOS device has turned off, thus not insuring circuit safety even out of active operating times.

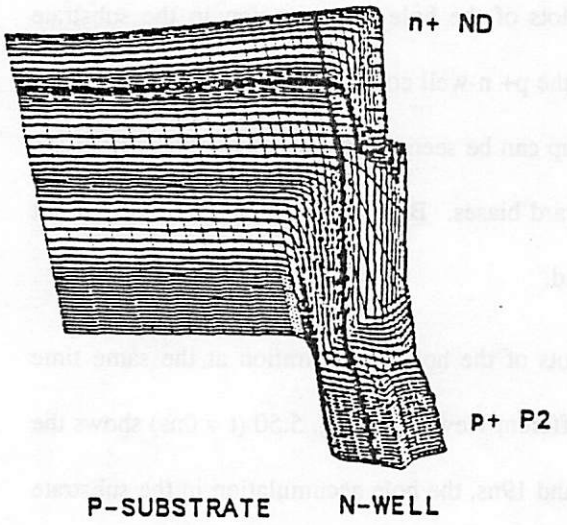


Fig. 5.46 t = 9ns.

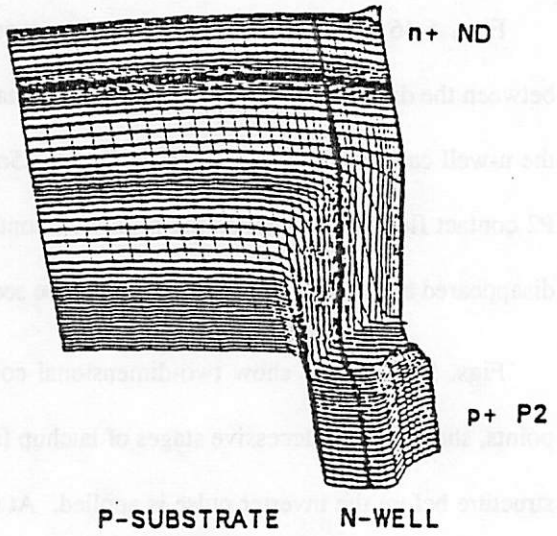


Fig. 5.47 t = 19 ns.

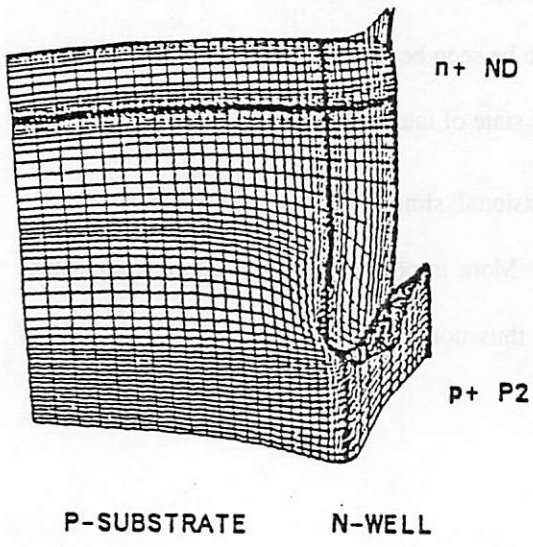


Fig. 5.48 t = 25ns.

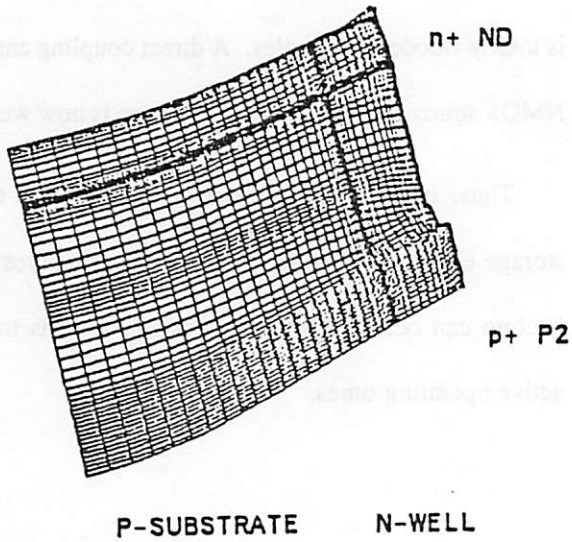


Fig. 5.49 t = 30ns.

Fig. 5.46 - 5.49 Bird's eye view mountain plots of the hole concentration in the n-well and substrate showing the steps leading to latchup. Only the area between the drain of the NMOS transistor and the p+ contact in the n-well, to a depth of 15μm into the substrate, is shown.

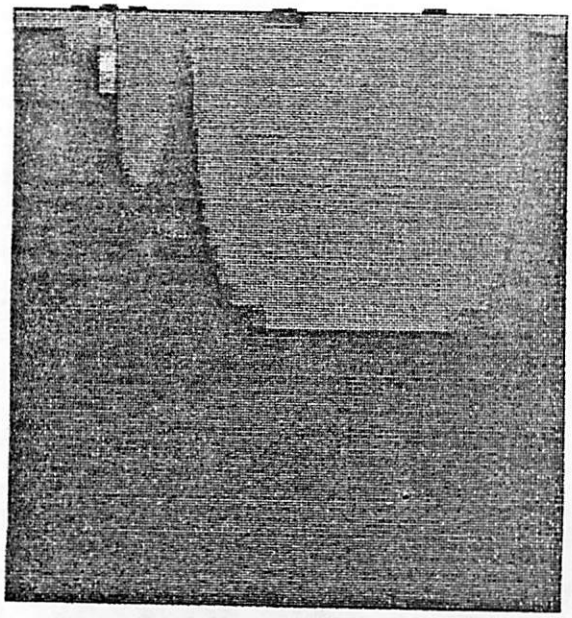


Fig. 5.50 t = 0 ns.

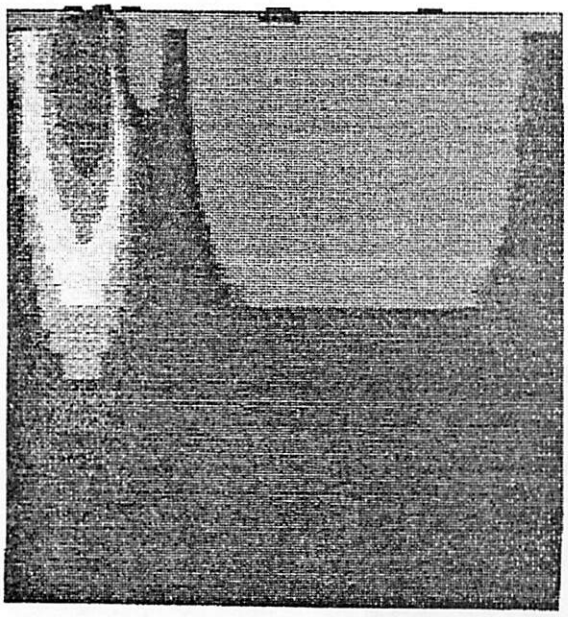


Fig. 5.51 t = 9 ns.

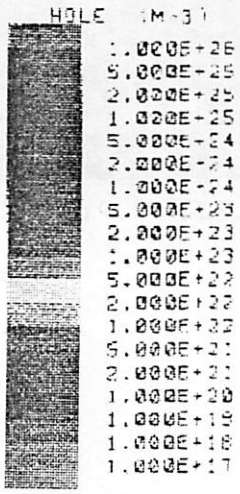


Fig. 5.50 - 5.54 Two-dimensional contour plots of the hole concentration in the substrate between the NMOS transistor and the n-well (p-substrate topside contact is to the left, out of the area of display). The substrate is displayed down to 10 μm. Concentration values given in the legend are in units of m^{-3}.

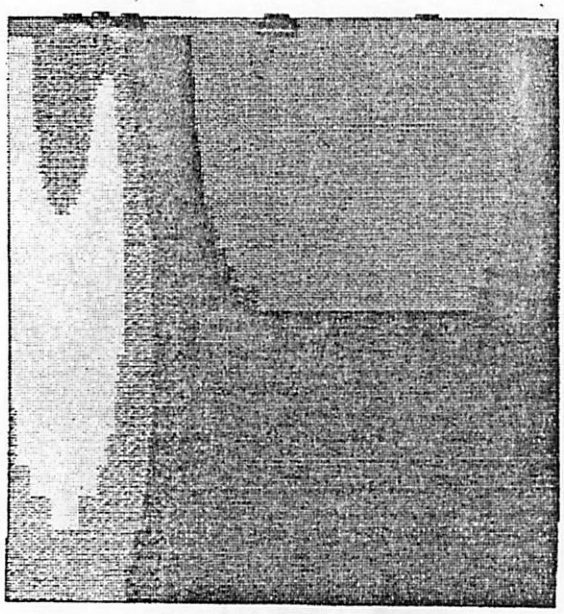


Fig. 5.52 t = 19 ns.

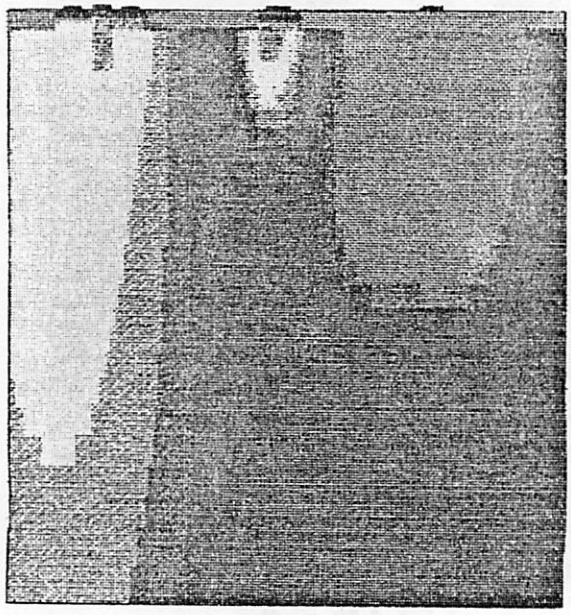


Fig. 5.53 t = 25 ns.

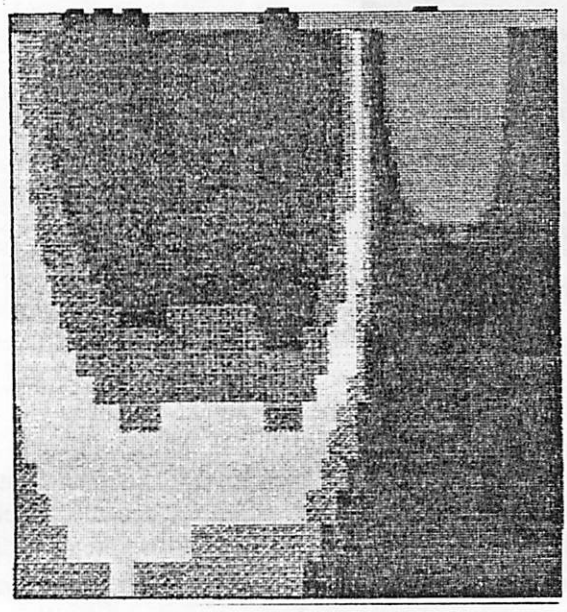


Fig. 5.54 t = 30 ns.

5.7 Conclusion

A bipolar charge storage phenomenon induced by substrate current flow in a resistive substrate has been presented. Measurements done on a CMOS inverter configuration have verified the existence of the phenomenon, and two- and three-dimensional device simulations using CADDETH have made it possible to analyze the mechanisms involved, and to see the effects of various bias and technological factors in changing the length of the delay.

A physical model with device extractable parameters has been developed as an extension to the avalanche breakdown model to predict the transient delay from general concepts.

From clocked digital input analysis it is found that inverter-based circuits can be biased with a power supply voltage higher than the DC avalanche breakdown voltage of the individual transistors. However, it has been shown that circuits subject to a clock frequency greater than the inverse of the I_{sub} delay time will experience charge buildup and will eventually break down. Finally, simulation has shown that excess charge flow can induce CMOS latchup, even after the NMOS device has turned off.

5.8 References

[Mul77] Richard S. Muller and Theodore I. Kamins, *Device electronics for integrated circuits*, John Wiley and Sons, New York, 1977, p. 26.

[Ko81] P.K. Ko, R.S. Muller, and C. Hu, "A unified model for hot-electron currents in MOSFETs," *IEDM Tech. Digest*, December 1981.

[Sze81] S.M. Sze, *Physics of semiconductor devices*, John Wiley and Sons, New York, 1981, p. 55.

[Hsu83] F.-C. Hsu, "Breakdown mechanisms and related effects in MOSFET's," University of California, Berkeley, Ph.D. Thesis, April 1983.

[Cha84] T.Y. Chan, P.K. Ko, and C. Hu, "A Simple Method to Characterize Substrate Current in MOSFETs," *IEEE Trans. Electron. Device Letters*, Vol. EDL-5, p.505, December 1984.

[Sod84] C. Sodini and P.K. Ko, "The effect of high fields on MOS device and circuit performance," *IEEE Trans. Electron. Devices*, Vol. ED-31, pp. 1386-1393, October 1984.

[Hsu85] Fu-Chieh Hsu and Kuang Yi Chiu, "Hot-electron substrate-current generation during switching transients," *IEEE Trans. Electron. Devices*, Vol. ED-32, No. 2, pp. 394-399, February 1985.

[Toy85] T. Toyabe et. al., "Three-dimensional device simulator CADDETH with highly convergent matrix solution algorithms", *IEEE Trans. Electron. Devices*, Vol. ED-32, No. 10, pp. 2038-2044, October 1985.

CHAPTER 6: CONCLUSION

In this dissertation we have presented hot-carrier modeling concepts that have culminated in the development of the BERT-CAS circuit aging simulator. Using the quasi-static models incorporated in CAS, we have seen that DC stressing measurements and an accurate parametric substrate current model are adequate to predict AC circuit degradation, at least for the CMOS inverter case. From looking at CMOS inverter-based circuits, it is found that the traditional 10% drain current degradation used for device lifetime may be overly conservative. In fact, for other types of circuits, it has been explained that using device lifetime as a criteria for circuit hot-carrier reliability can be misleading because of the varying sensitivity each transistor may have to circuit output behavior. Because raw circuit output behavior is simulated, circuit designers can look at any aspect of circuit performance when judging the hot-carrier reliability of his circuit.

CAS is based on a more comprehensive degradation model based on the degradation driving force $I_{ds}(I_{sub}/I_{ds})^m$ rather than I_{sub} alone. This expression, which has a more plausible theoretical basis, has been shown to correlate better with device degradation experiments, and the fact that CAS can predict circuit-level degradation adds confidence that this is the correct model to use.

A rough rule of thumb has been introduced to relate a particular class of circuits, CMOS inverter-based circuits, to circuit speed. Such a guideline should prove useful for quick back-of-the-envelope calculations for initial estimates.

Finally, a bipolar charge-storage phenomenon has been presented both from experiments and from three-dimensional device simulation. Using the device simulator CADDETH, it was possible to probe inside the device to analyze the mechanisms involved. A model has been introduced to roughly predict the length of time needed for the stored charge to disappear. Adverse effects this phenomenon my have to neighboring structures was graphically demonstrated by simulating a CMOS structure which was induced into the latchup state by the bipolar charge generation. It has also been shown that this phenomenon explains the fact that dynamic inverter-based circuits with a clocking waveform slower than

the inverse of the delay time can tolerate power supply voltages greater than their DC avalanche breakdown voltages.

This concludes this dissertation concerning hot-carrier effects in MOS devices and circuits. It is hoped that additional insights have been gained in predicting circuit behavior and degradation from device-level concepts, and that therefore higher performance circuits can be designed without sacrificing reliability.