

Large Signal Transient Analysis of Parasitic SCR, Latchup Inverters and Bipolar Devices

Thomas Archibald Tetzlaff

B.S., Applied Physics, California Institute of Technology, 1989

M.S., Electrical Engineering, Oregon Graduate Institute, 1993

A dissertation submitted to the faculty of the
Oregon Graduate Institute of Science & Technology
in partial fulfillment of the
requirements for the degree
Doctor of Philosophy
in
Electrical Engineering

October, 1996

The dissertation "Large Signal Transient Analysis of Parasitic SCR, Latchup Inverters and Bipolar Devices" by Thomas Tetzlaff has been examined and approved by the following Examination Committee:

Dr. V. S. Rao Gudimetla, Thesis Advisor
Assistant Professor

Dr. J. Fred Holmes
Professor and Department Head

Dr. Anthony E. Bell
Associate Professor

Dr. Shahriar Ahmed
Adjunct Professor, Intel Corporation

Table of Contents

Abstract	vii
Chapter 1: Introduction	1
Section A: Section Overview	1
Section B: Concepts and Definitions	1
Section C: Static and Transient Behavior	3
Section D: Primary Modeling Approaches	5
Chapter 2: Overall Approach	11
Chapter 3: General System Behavior	19
Chapter 4: Analytic Expressions for Latchup (Ker and Wu Method) ..	25
Chapter 5: Latchup Criteria Based on Voltage Instability	37
Chapter 6: Latchup Predictions for Intel Devices	48
Chapter 7: Analytical Solution with Improved Accuracy	85
Chapter 8: Summary	92
References	94
Appendix - Program Listings	97
Vita	114

List of Figures

Figure 1-1 I-V curve of pnpn structure	9
Figure 1-2 Side view of typical CMOS latch structure	9
Figure 1-3 Detailed equivalent circuit for pnpn structure	10
Figure 1-4 Current flows in device approaching latch state	10
Figure 2-1 Side view of P-well CMOS inverter with latching path	15
Figure 2-2 Circuit representation for parasitic SCR structure	16
Figure 2-3 Voltages vs. Time for non-latching pulse	17
Figure 2-4 Voltages vs. Time for latching pulse	17
Figure 2-5 Voltages vs. Time with large step size	17
Figure 3-1 Curves of zero dV/dt , showing static solutions	24
Figure 4-1 Transient beta product for a 4 Volt Vdd pulse	33
Figure 4-2 Response to a 79 nsec, 1 volt Vdd pulse	34
Figure 4-3 Response to a 0.85 nsec, 100 volt Vdd pulse	34
Figure 4-4 Log plot of current vs. Emitter-base voltage	35
Figure 4-5 Linear fit for collector current vs. V_{be}	35
Figure 4-6 Time to latch variation for different linear fits	36
Figure 5-1 Comparison of Runge-Kutta and SPICE solutions	42
Figure 5-2 Comparison of RK with 0.01 nsec step size and SPICE ...	43
Figure 5-3 Parasitic PNP latchup behavior	44
Figure 5-4 V_{be} vector flow near static point	45
Figure 5-5 Critical curve (separating off and latch regions)	46

Figure 5-6 Response to a 9.7 nsec, 4 volt Vdd pulse	47
Figure 6-1 PNP I-V curve for "A" structure	52
Figure 6-2 NPN I-V curve for "A" structure	53
Figure 6-3 Derived SPICE parameters for "A" structure	54
Figure 6-4 Minimum pulse time vs. Trigger current	55
Figure 6-5 Minimum pulse time vs. Well resistance	56
Figure 6-6 Minimum pulse time vs. Substrate resistance	57
Figure 6-7 Minimum pulse time vs. Well-substrate capacitance	58
Figure 6-8 Minimum pulse time vs. Forward transit time of NPN	59
Figure 6-9 Minimum pulse time vs. Forward transit time of PNP	60
Figure 6-10 Minimum pulse time vs. Beta of PNP	61
Figure 6-11 Latch curves of Rs-30%, Cjc0-30%	62
Figure 6-12 Latch curves of Rs nominal, Cjc0-30%	63
Figure 6-13 Latch curves of Rs+30%, Cjc0-30%	64
Figure 6-14 Latch curves of Rs-30%, Cjc0 nominal	65
Figure 6-15 Latch curves of Rs nominal, Cjc0 nominal	66
Figure 6-16 Latch curves of Rs+30%, Cjc0 nominal	67
Figure 6-17 Latch curves of Rs-30%, Cjc0+30%	68
Figure 6-18 Latch curves of Rs nominal, Cjc0+30%	69
Figure 6-19 Latch curves of Rs+30%, Cjc0+30%	70
Figure 6-20 Effects of Rw and Cjc0 on minimum pulse time	71
Figure 6-21 Physical layout of the "B" test structure	72
Figure 6-22 NPN I-V curve for "B" structure	73
Figure 6-23 NPN beta curve for "B" structure	74
Figure 6-24 NPN Ic curve and parameter fit for "B" structure	75
Figure 6-25 NPN I-V curve and SPICE parameters (includes resistance)	76
Figure 6-26 PNP I-V curve and SPICE parameters	77
Figure 6-27 NPN beta curve and SPICE parameter fit	78
Figure 6-28 Static points for "B" structure	79

Abstract

Existing methods for predicting latchup in CMOS devices are critically examined and a new criterion for latchup is presented. The previous latchup criterion, requiring the transient beta product to be greater than one, is discussed and an improved criterion is determined. The new criterion is based on finding the unstable, static solution of equations for the voltages within the device. If an unstable solution exists, then the separatrix through that solution will divide all possible operating states into those that return to the off state and those that latch. If an unstable solution does not exist, then a stable latch point does not exist and all operating states eventually return to the off state. This new criterion has the benefit of matching exactly with numerical solutions for minimum pulse times to cause latchup. Furthermore, structures which are latchup immune can be quickly identified without exhaustive simulation. Analytical methods of solving the system equations are presented which allow quick estimations of minimum pulse times which will lead to latchup. The benefit of this is to provide latchup behavior for a given CMOS structure in significantly less time than SPICE simulations of similar accuracy would take. The new results are then applied to Intel test structures for process margin determination.

I. Introduction

A. Section Overview

The purpose of this section is to review the published papers about modeling latchup in bulk CMOS. The intention is to cover the basic mechanisms involved, circuit-equivalent models and two-dimensional simulations. In addition, derivations involving both static and transient latchup triggering characteristics will be treated.

Initially, a concise overview is presented. This summary includes a general description of latchup--starting with the concepts and definitions and continuing with the static and transient latchup characteristics. The material for this section is largely taken from the papers by Estreich¹⁻², unless otherwise noted. Next, the basic modeling approaches are covered, consisting of SPICE-type circuit equivalent models, time-dependent two-dimensional numerical simulations (including PISCES), and analytical formulations for transient latchup. The overview will be followed by descriptions of significant findings of the reviewed papers not covered in the overview.

B. Concepts and Definitions

Latchup can be defined as a low resistance path between voltage levels. Usually, the circuit function is disrupted, and currents are frequently large enough to cause permanent damage. In bulk CMOS circuits, latchup is caused by the triggering of a parasitic p-n-p-n

structure to its "on" state. A typical p-n-p-n current-voltage characteristic is shown in figure 1.1⁽³⁾, taken from Troutman's book on latchup because it provides a consistent terminology. The critical points are the switching point (V_s, I_s), which marks the transition from the high resistance "off" state to the negative differential resistance region; and the holding point (V_h, I_h), which marks the transition from the negative differential resistance region to the low resistance "on" state. Also indicated is the turn-off point (V_{to}, I_{to}), where the center junction has no voltage across it; this point is unstable. In all cases, the voltages are measured from p-region on one end to the n-region on the other. No internal measurements are included. The switching point, also referred to as threshold, is the point where regeneration occurs. The holding point is sometimes incorrectly labeled as the lowest current where the p-n-p-n is on; however stable points of lower current exist down to the turn-off point, as pointed out by Rung and Momose⁴, albeit using different terminology. The holding point is equivalently defined as the point where the slope on the I-V curve is vertical ($dV/dI = 0$).

Although many p-n-p-n structures exist in a typical CMOS IC, the basic building block in modeling latchup is a CMOS inverter. Figure 1.2 shows an inverter structure with the p-n-p-n path highlighted schematically by two parasitic bipolar transistors. The arrows indicate the current flows required to forward bias the emitter-base junctions of the transistors. (Although the source diffusions of the MOS transistors also form p-n-p-n structures, latchup would not generally be sustained since V_{out} is not a power supply.) For latchup to be sustained, three conditions must be satisfied:

1. Both parasitic transistors must be biased into the forward active region.

The emitter-base junctions are forward biased which implies minority carrier injection occurs.

2. The beta gain product must be sufficient to allow regeneration. Alternatively, minimum beta product requirements can be recast as minimum alpha sum requirements.

3. The power supply must be capable of supplying current and voltage greater than the current and voltage of the holding point. It should be noted, however, that momentary latchup could still cause temporary disruption of

circuit function.

In order to forward bias the emitter-base junctions, lateral currents causing voltage drops must flow as indicated above. Estreich listed four mechanisms for lateral current generation:

1. Voltage transients -- When $V_{dd}-V_{ss}$ is increased in magnitude, the well-substrate depletion layer widens. Carriers flow away to expose ionized donors / acceptors. The resulting displacement current is given by the junction capacitance multiplied by the time rate of change of voltage across the junction.
2. Carrier injection -- Nearby junctions that are forward biased could inject minority carriers into the region.
3. Overvoltage stress -- If the difference between V_{dd} and V_{ss} is large enough, avalanche breakdown of the well-substrate junction occurs.
4. Radiation induced photocurrents.

The first two of these mechanisms are the ones of concern here. The last two provide an interesting means of investigating latchup, but generally will not be unintentionally triggered to cause latchup.

C. Static and Transient Behavior

Initially, the focus was on developing expressions for sustained latchup, as seen in Estreich's three conditions for latchup. In developing expressions for holding voltage or holding current, the idea was that latchup could be avoided if the supply voltage was lower than the holding voltage. Another justification for focusing on the holding point was that the switching point depends on the switching mechanism, but the holding point is independent of switching mechanism⁵. However, as pointed out by Troutman and others, the holding point criterion only avoids sustained latchup--momentary latch could still occur. So, what is really desired is to guarantee that the switching point never be reached by any mechanism. Another difficulty is that at the holding point, both sides of the well-substrate junction are flooded with carriers. Seitchik⁶ claims that in this case, it is not correct to model the situation in terms of

interacting pnp and npn transistors; the existence of a p-n junction is irrelevant, reducing the p-n-p-n model to a p-i-n structure. He uses this in developing an analytic model of holding voltage.

Several authors have investigated transient latchup through simulation and measurement for various structures⁷⁻¹⁶. In examining a transient latchup criterion, Troutman noted that the previous criteria of the sum of large signal forward gain alphas greater than one was, in fact, incorrect. The improved criteria involved small signal gain alphas, defined as the derivative of collector current with respect to emitter current. Also, effective injection factors accounting for the loss through the shunting resistors modified the gain alphas to arrive at effective small signal alphas. The condition for stability becomes the sum of the two effective small signal alphas must be less than one. By plotting the npn effective alpha on one axis and the pnp effective alpha on the other, then the line with a sum of one is unstable. The points of this line are switching points and the interior of the triangle formed by this line with the two axes are all stable, blocking points. The interior is called SAFE space by Troutman. (SAFE is not an acronym, he just capitalized it to emphasize that these points are safe from latchup). Different methods of stimulating latchup will take different paths through SAFE space and arrive at different switching points; but if the transistor gains are known, the path can be calculated. So, in principle, one can guarantee a structure to be free from latchup if all switching mechanisms can be constrained to be of short enough duration such that operation remains in SAFE space.

An alternative latchup criterion is proposed by Yang and Wu^{17,18}. They suggest that the change in charge stored in the junction depletion capacitance of the p-n-p-n structure is a constant for any triggering current applied for the minimum regeneration time. Unfortunately, no physical basis was given for this phenomena, only that it was consistent with SPICE simulations. However, further investigation of the physics would be required to adopt this as a legitimate criterion for latchup.

D. Primary Modeling Approaches

1. Circuit Equivalent Models

In this section, the lumped-element circuit equivalent is discussed, starting with the features of the parasitic transistors. The transistor whose base is formed by the well (the npn in p-well CMOS) is a vertical device that usually has high gain. In contrast, the lateral device has low gain due to the wide base dimension. As a side note, lateral flows can exist even for the vertical device for special surface conditions¹⁹. In modeling the lateral device, a field aided drift component of the current across the base should be considered, instead of just looking at diffusion. This can be simulated in SPICE using a current-controlled current source from emitter to collector over small portions of the I-V curve. A better choice is to use a simulator like ASTAP (Advanced Statistical Analysis Program), which allows new model equations to be directly input²⁰.

At this point, only the parasitic transistors have been discussed, leaving a circuit equivalent much like to simple SCR. This would imply regeneration would occur for a beta product of the npn and pnp transistors greater than one. However, the existence of well and substrate resistances acting to shunt the emitter-base contacts alters the beta product requirement to the following:

$$\beta_n \cdot \beta_p \geq \frac{I_{DD} + I_{Rw} \beta_n}{I_{DD} - I_{Rw} - I_{Rs} \left(\frac{\beta_n + 1}{\beta_n} \right)}$$

R_w and R_s are the well and substrate emitter-base shunting resistances, respectively. I_{Rw} and I_{Rs} are the currents through the shunt resistors. If these resistances become large, then the resistor currents approach zero and the minimum beta product requirement for latchup becomes one. If either R_s or R_w approach zero, the required supply current to sustain latchup and the minimum beta product both become large. Also, the beta product equation can be written in an equivalent form with the beta of the pnp transistor on the

right-hand side instead of the npn beta. The current flows through the resistor R_s tend to be pinched near the contact and spread out from there, indicating the inherent three-dimensional nature of latchup. Methods of measuring these resistances physically have been suggested by Chen and Wu²¹.

So far, the circuit model had just the two transistors with two emitter-base shunting resistors. Additional circuit elements can be added to more accurately simulate the structure. Of importance in transient analysis is including the well-substrate capacitance, either as a single element or as a distributed RC network by partitioning into segments. A typical equivalent circuit including these elements is shown in figure 1.3. In the figure, the extra capacitances and resistances are from the partitioning of the well-substrate depletion region. The current sources model radiation induced latchup and are not part of the actual structure. Since the behavior of the parasitic transistors connected to the outputs has been observed to play a role²²⁻²³, Troutman included these paths. Base-emitter capacitances, transistor branch resistances⁴, the CMOS transistors²³, and transmission line elements²⁴ have all been used to further refine the circuit equivalent model.

2. Time-Dependent Numerical Simulations

In order to determine current flow patterns and more accurately include capacitive effects, time-dependent numerical simulations were used. The two approaches used were to determine the solution to a modified set of device modeling equations^{7,19,25} or to use PISCES^{8,11,26}. An example of the current flows just prior to latchup are shown in figure 1.4, from Pinto and Dutton²⁷. In this case, the vertical npn turns on due to the triggering current at the $n+$ contact. In order to more efficiently trigger latchup using PISCES, Chatterjee²⁸ used a "charge-triggering" method. If latchup is triggered simply by raising the substrate potential, then switching occurs after a delay corresponding to the transit time that is difficult to predict accurately. Small time-steps during switching are needed for convergence, but large time-steps are desired before and after switching for efficiently-

-making time-step selection difficult. The "charge-triggering" method involves introducing a large number of electron-hole pairs along a track spanning the well-substrate junction so that the response is almost immediate. The time-steps start small and increase exponentially; no need to accurately guess the transit time.

In measuring actual devices, Sangiorgi²⁹ finds that 2D simulations are of questionable validity for real structures due to three-dimensional effects. The emitters on devices 50 to 200 microns wide were split into a number of sections so measurements could be made on each section, while the distance between the p+ and n+ emitters was fixed at 8 microns. Only the middle section of the 200 micron device gave good agreement with numerical simulations. Otherwise, current densities were non-uniformly distributed, with densities higher at the edges than in the middle. Using two-dimensional simulations to model real structures, whose widths would be much less than 50 microns, would appear to be invalid. However, the number of segments on the emitters was quite small; the 100 and 200 micron devices had five emitter sections while the 50 micron device had just three emitter sections.

3. Analytical formulations

The first analytical model of transient latchup was presented by Troutman and Zappe¹⁰. In order to arrive at equations that could be solved explicitly, many assumptions were made. The response was broken into four regions: both transistors off, one transistor on, both on, and both saturated (latchup). The well-substrate capacitance was assumed constant. The transistors were assumed to have no current for emitter-base voltages below a given on-voltage and a constant forward gain α with emitter-base voltage locked at the on-voltage. This circuit was analyzed for a powerup voltage with a constant ramp rate. In other words, the voltage on the latchup structure was initially zero, and increased linearly to some final value, where it remained. The conclusions were that the structure was stable if the sum of the forward gain alphas of the two transistors was less than one, or if the product of the ramp rate and the capacitance was less than a given

constant. If both of these conditions were not met, then the structure was unstable if the ramp-up time was long enough. The main limitation to this analytical model is that the triggering is only for power-on transients, although extending it to transients after the structure is already powered is feasible.

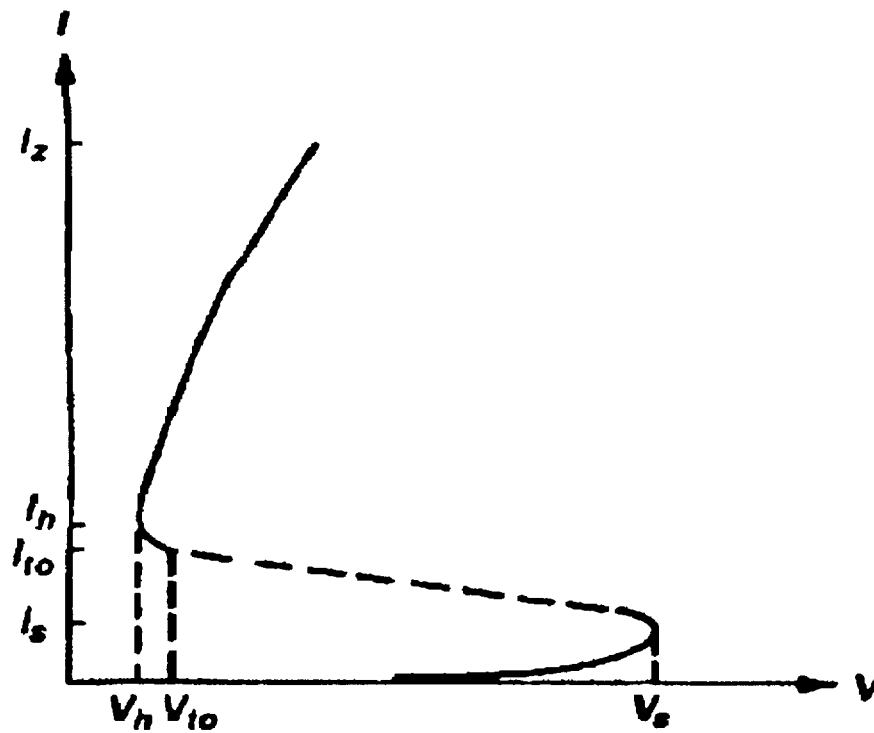


Figure 1.1 I-V curve of pnpn structure, with points of interest labeled.

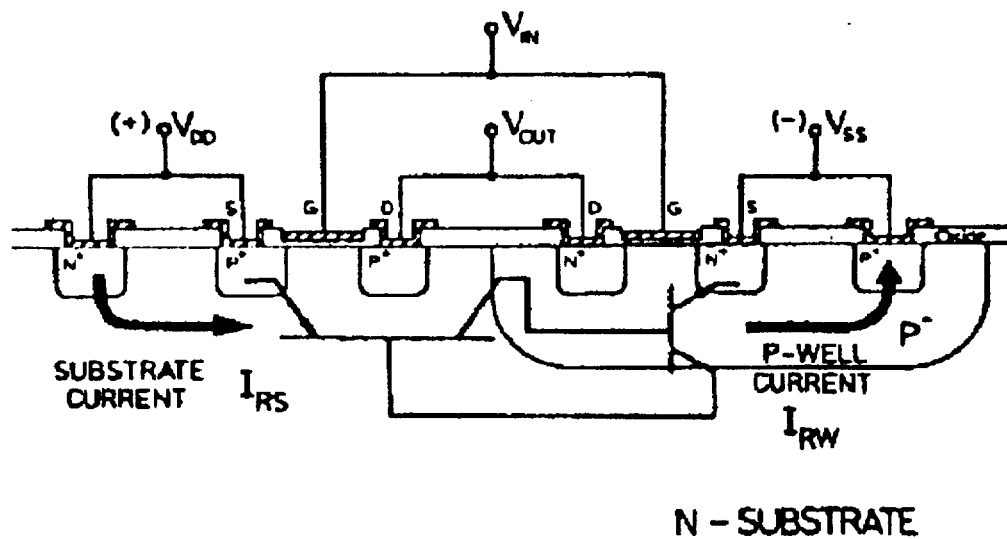


Figure 1.2 Sideview of typical CMOS latch structure with back-to-back transistors overlaid to show their location.

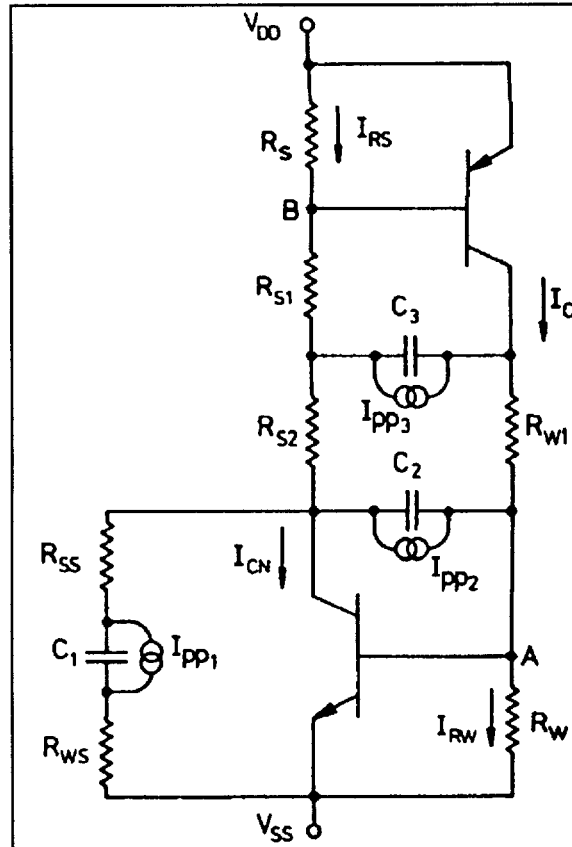


Figure 1.3 Lumped-element equivalent circuit model for pnpn structure.

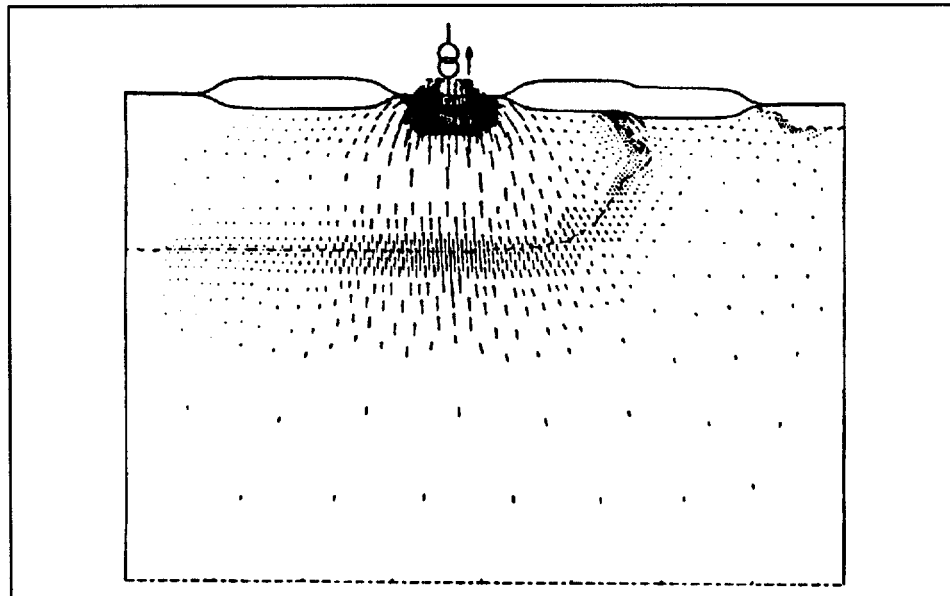


Figure 1.4 Illustration of current flows in device approaching the latch state.

II. Overall Approach

The starting point in considering latchup is to consider a CMOS inverter. Figure 2.1 shows the p-n-p-n latching path in a cross-sectional view of a CMOS inverter. (P-well CMOS is shown, but the choice is arbitrary.) Although the Vout diffusions also form p-n-p-n structures, it is assumed they cannot cause sustained latchup since these diffusions are not connected to a power supply. This latchup path p-n-p-n structure is modeled by two transistors with emitter-base shunting resistors, as is typically done. In order to investigate transient responses, the capacitances at each junction are also included. So, for this structure, the parasitic structures of interest are: a p+ region at Vdd, n-substrate, and p-well forming a lateral pnp transistor; an n+ region at ground, p-well, and n-substrate forming a vertical npn; a substrate resistance from the active n-substrate to the n+ contact at Vdd; a well resistance from the active p-well to the p+ contact at ground and the junction capacitances. The two resistances connect the base region of each transistor with an external contact. Since the external base contact is connected to the same potential as the emitter for each transistor, these resistances act to shunt the emitter-base junctions.

Although latchup has been extensively studied in the past and the pnpn latching path will look familiar to anyone even casually familiar with latchup, an exact definition of what is meant by “latchup” in this work follows. Latchup is a stable, low-resistance path between power and ground. Other authors have attempted to define “momentary latchup,” which has some vagueness as a concept. A momentary latchup is a pnpn structure exhibiting large current flow under the influence of a noise pulse. But, if a noise pulse is introduced which forward biases a pn junction, large currents will flow for the

duration of the pulse and possibly for some time after it, regardless of whether the junction is part of a pnpn structure or not. So, “momentary latchup” would be possible in a structure that has no stable latchup state. Using the strict definition of latchup, the information of interest is whether a given structure has the possibility of a stable, low-resistance path between power and ground, and if so, for what pulses would a structure initially in a stable off state eventually go to the latch state.

Figure 2.2 shows the lumped-element equivalent circuit used as a SPICE model for the parasitic SCR (semiconductor controlled rectifier = pnpn) structure. More complex equivalent circuits can be used, as originally shown in figure 1.3, by modelling the resistances and capacitances in several pieces. However, while conceptually this is simple, in practice determining the element values is quite difficult. And, the resulting model simulations do not provide additional accuracy over the simpler model. Transient signals, which can stimulate latchup, are introduced as noise voltages at the external base contacts. For convenience, the polarity is chosen so that positive voltage noises will tend to forward-bias the emitter-base junctions. The choice of stimulating latchup with a voltage pulse on the external base is equivalent to using a current pulse at the intrinsic base node. In figure 2.2, the current sources can be replaced by noise voltages on the shunting resistors, where $I_1 = V_{\text{noise,vdd}}/R_1$ and $I_2 = V_{\text{noise,gnd}}/R_2$ based on rules of source transformations. The advantage of voltage pulses is that, since these are located at external nodes, latchup predictions can be physically tested on real devices. Current pulses located below the surface of the device, however, can not be accurately introduced in a real device. Hence, predictions cannot be satisfactorily verified using internal currents to trigger latchup. In general, the noise voltages triggering latchup could be acting simultaneously with arbitrarily complex profiles in time. In order to derive analytical expressions and simplify analysis, only square pulses on one noise voltage at a time are examined. By taking a pulse of a given magnitude, the goal is to be able to calculate the minimum time required for that pulse to cause latchup.

At this point, some comments about the approximations in this circuit model are in order. First, distributed resistances and capacitances are lumped into single elements. In

particular, pinching effects are ignored so that the resistors are constant instead of depending on the current flowing through them. By expressing the resistance as a function of applied voltage, pinching effects can be incorporated into numerical simulations of the circuit equations. However, the analytical solution to be discussed later requires constant resistor values. Secondly, it has been suggested that the lateral transistor will be affected by lateral electric field aiding effects, which are not explicitly addressed by the Gummel-Poon model implemented in SPICE. It should be noted, however, that the effects are still accounted for in fitting transistor transfer curves to the given model. Finally, pulse times are frequently of the same order as the forward transit times. Since transit time is a single value based on averaging the velocity distribution, small times depend on the detail of the distribution. These approximations tend to make the circuit equations as simple as possible while still yielding acceptably accurate results.

To determine whether or not a given noise pulse will latch using SPICE, the emitter-base voltage of both parasitic transistors are tracked as a function of time. Figures 2.3 and 2.4 show examples of latching and non-latching pulses. (The circuit parameters used are given in table 1.) Unless otherwise indicated, V_{dd} is taken as five volts above ground. Shortly after the 9.72 nanosecond, four volt pulse ends, the emitter-base voltages return to the neighborhood of zero, which is the off state. By making the pulse slightly longer at 9.74 nanoseconds, the voltages after the pulse ends increase to a stable latched state where both transistors are forward active so that large currents are flowing. The SPICE input file used to generate these plots is listed at the end of the chapter. These simulations illustrate how the voltages change both during and after a transient voltage pulse, and the fact that the "arrival" at the latch or off state might not occur until long after the pulse ends. This makes using SPICE to determine a critical pulse time somewhat inefficient. The simulation must be continued well after the pulse ends, and trying a different pulse time requires a separate simulation run.

Also, obtaining accurate results in SPICE requires small time steps in the transient analysis. In both pSPICE and TEKSPICE, the limit on the allowed number of data points must be increased using an options statement in order to run the simulation. The effects of

step sizes on latchup results are shown in figure 2.5. A step size of 0.10 nanoseconds leads to latchup in the simulation. This step size is simply too large, latchup results from inaccuracies in the numerical approach taken. When the step size is reduced to provide more accuracy, the simulation indicates a return to the off state. This example dramatically highlights the importance of choosing small step sizes. Of course, the penalty for small step sizes is increased solving time and more data points. For example, a 0.01 nsec step size with an end time of 200 nsec will generate 20,000 data points.

SPICE input file for the graphs which follow:

```
* Ker and Wu Paper's parameters 9.74ns pulse - 0.02 ns step
.model PNP1 PNP(Is=2.833e-16 Bf=1.104 Ise=4.250e-14 Ikf=6.909e-5 Br=0.2
+      Tf=20n Tr=10n Cje=0.6p Cjc=2p Mje=0.5)
.model NPN1 NPN(Is=8.112e-16 Bf=277.2 Ise=1.217e-13 Ikf=4.867e-4 Br=2.0
+      Tf=0.25n Tr=2n Cje=1.3p Cjc=0.6p Mje=0.5)
VDD  1 0 DC 5
Q1   6 4 1 PNP1
Q2   7 5 9 NPN1
RS   11 2 800
RW   3 12 5.6K
R3   9 0 1
VB1  2 4 0
VC1  3 6 0
VB2  5 3 0
VC2  7 2 0
VI1  11 0 PULSE 5 1 0N 0N 0N 9.74N
.PRINT TRAN V(4) V(5) V(9) I(I1)
.NODESET V(2)=5
.NODESET V(3)=0
.TRAN .02N 50N
.probe
.END
```

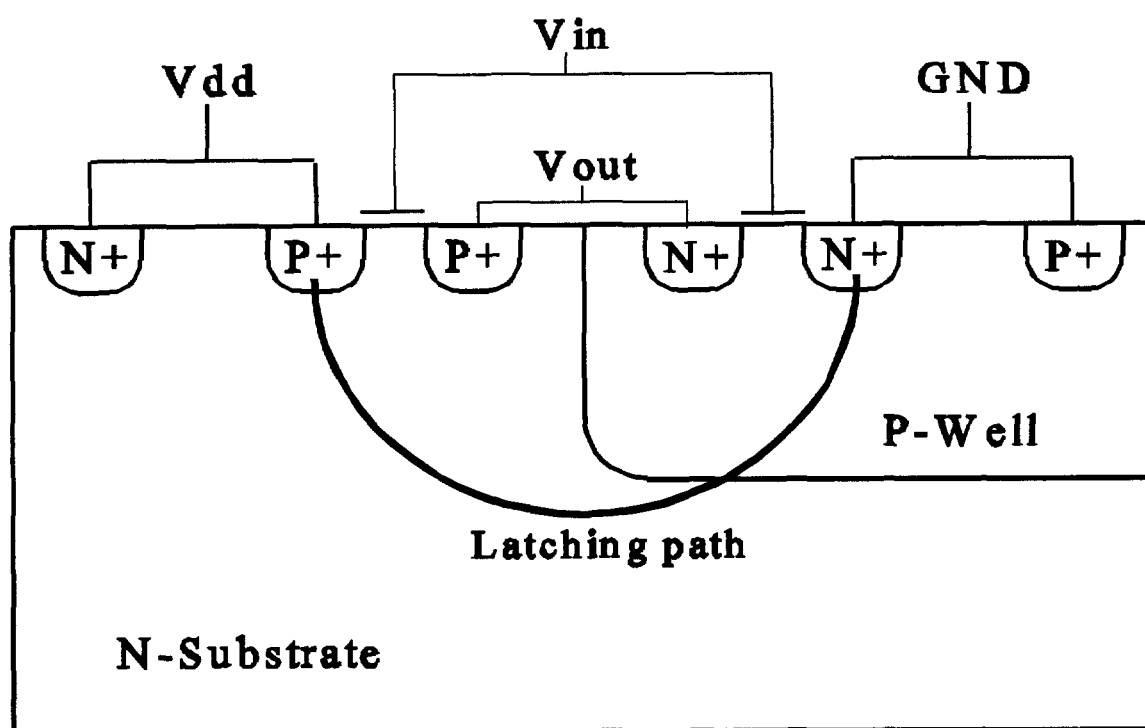


Figure 2.1 Side view of a P-well CMOS inverter with the parasitic p-n-p-n latching path indicated.

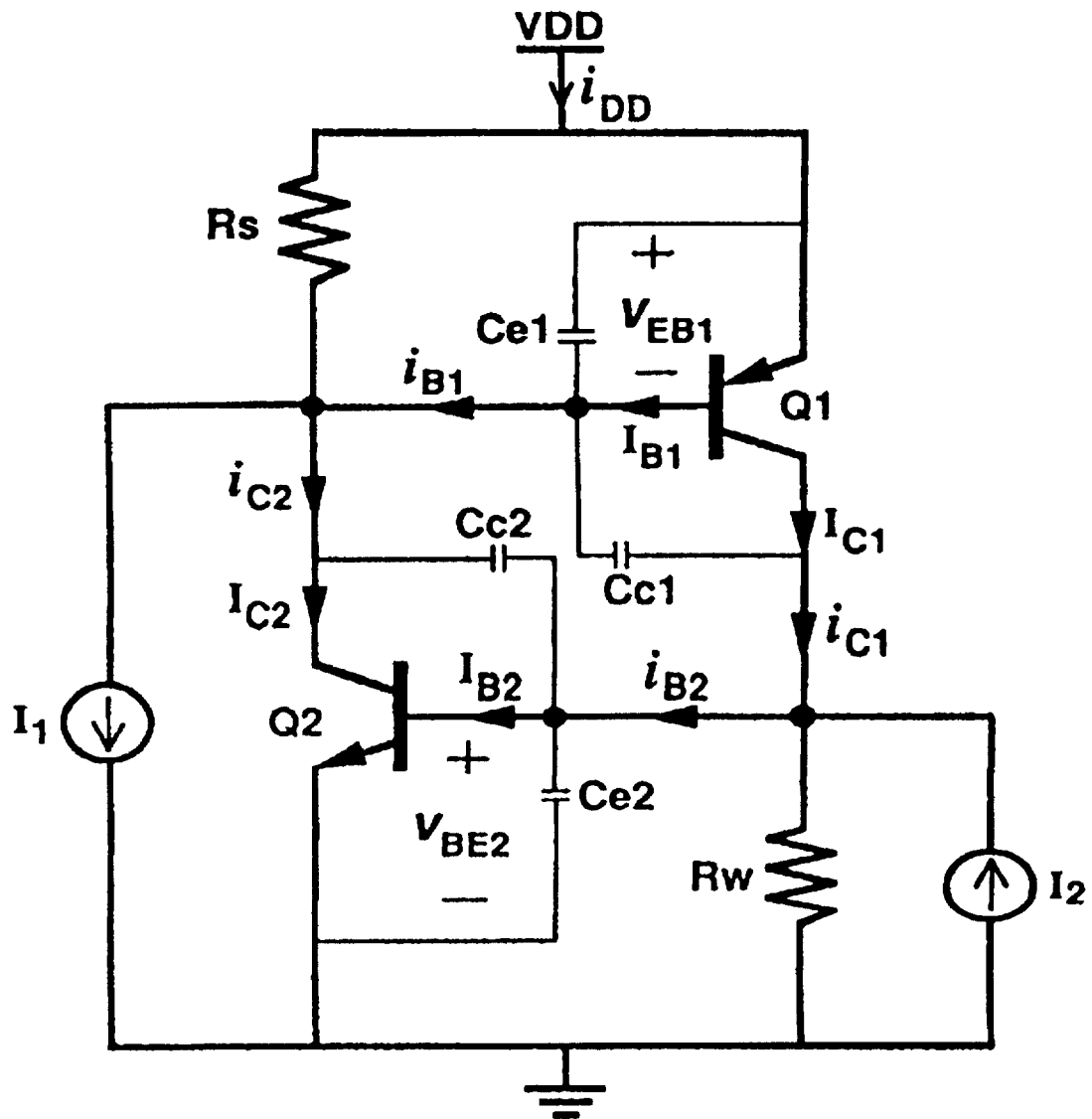


Figure 2.2 Lumped-element circuit representation for the parasitic p-n-p-n SCR structure in a CMOS inverter.

Response to 9.72 nsec, 4 Volt Vdd Noise
(0.01 nsec step size)

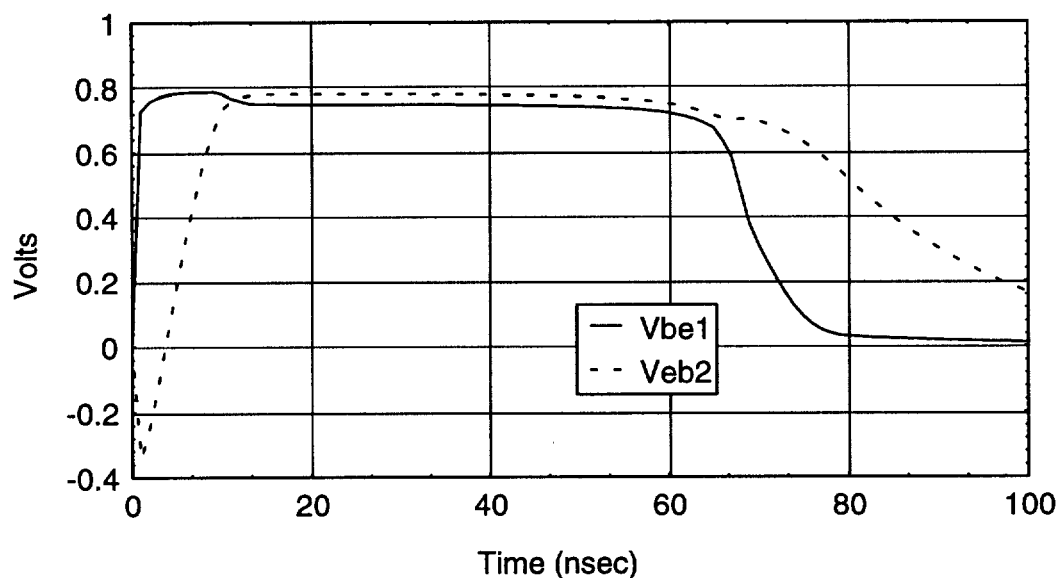


Figure 2.3 Emitter-base voltages as a function of time for a four volt Vdd noise pulse which ends at 9.72 nanoseconds. This is not sufficient to cause latchup.

Response to 9.74 nsec, 4 Volt Vdd Noise
(0.01 nsec step size)

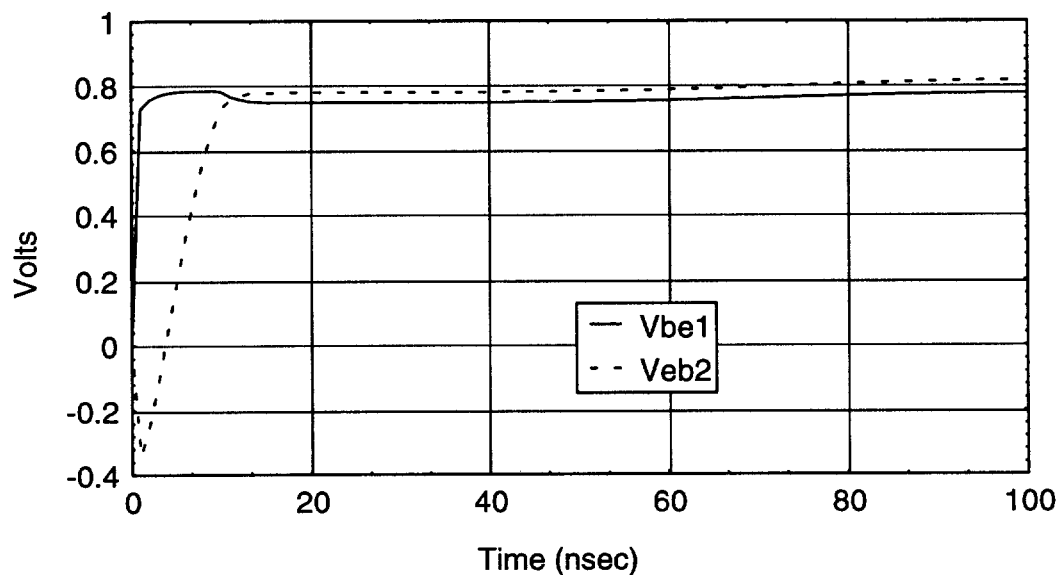


Figure 2.4 Emitter-base voltages for a four volt Vdd noise pulse which ends at 9.74 nanoseconds. This is sufficient to cause latchup.

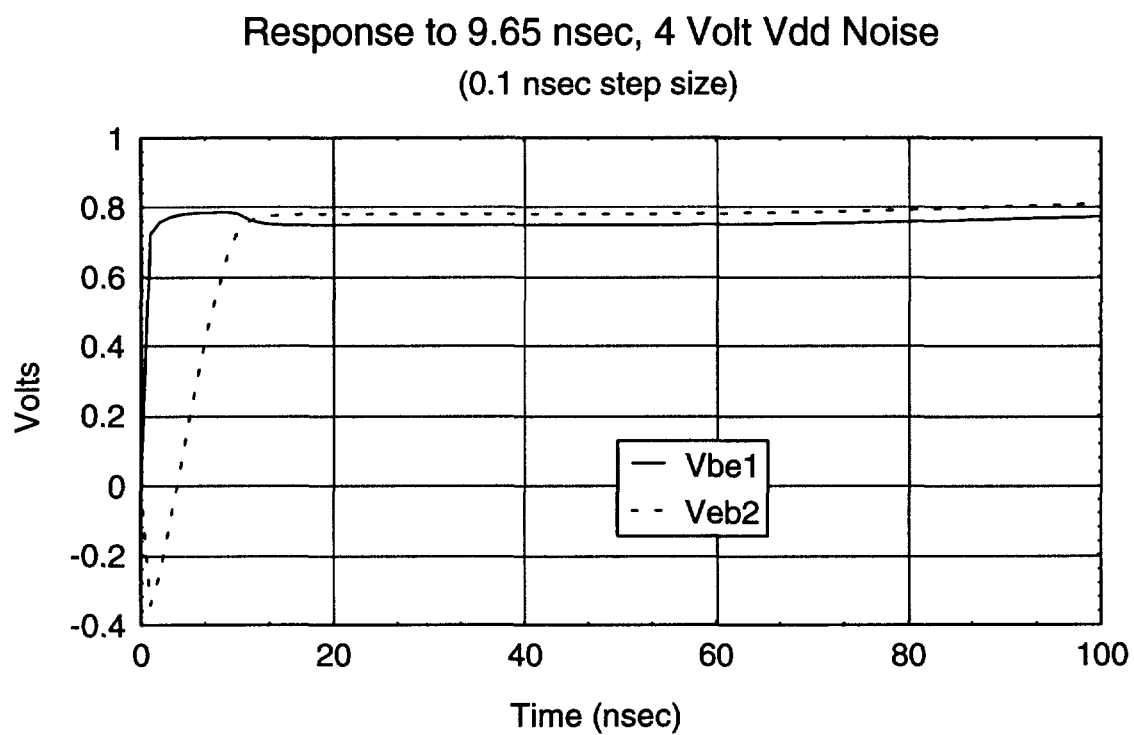


Figure 2.5 Emitter-base voltages as a function of time for a four volt Vdd noise pulse which ends at 9.65 nanoseconds. A large step size of 0.1 nsec was used, which resulted in an incorrect latch response.

III. General System Behavior

The notation to be used is as follows: $v1$ refers to the emitter-base voltage of the pnp transistor, $v2$ refers to the base-emitter voltage of the npn transistor. Both quantities will be considered positive if the junction is forward biased. The system equations for the equivalent circuit are of the form:

$$\frac{\partial v1}{\partial t} = f_1(v1, v2) \quad , \quad \frac{\partial v2}{\partial t} = f_2(v1, v2) \quad (1)$$

The equation solutions can be found through numerical techniques, but analytical descriptions of the latchup behavior are of interest here. The functions depend only on the state variables $v1$ and $v2$, with no explicit time dependence. These functions are therefore autonomous equations and the behaviors of the static points can be qualitatively categorized³⁰⁻³². The terms static and equilibrium solutions will be used interchangeably and refer to points where the functions $f1$ and $f2$ equal zero. At these points, the derivatives with respect to time of $v1$ and $v2$ are zero, hence the system will not change in time--it is static. In the neighborhood of an equilibrium point, the system equations can be linearized, yielding:

$$\frac{\partial v1}{\partial t} \approx a_{11}(v1 - v10) + a_{12}(v2 - v20) \quad , \quad \frac{\partial v2}{\partial t} \approx a_{21}(v1 - v10) + a_{22}(v2 - v20) \quad (2)$$

where $a_{ij} = \partial f_i / \partial v_j$, evaluated at the equilibrium point. Writing a_{ij} in matrix form, the eigenvalues determine the type of equilibrium state for the linearized system. The

behavior in the neighborhood of the equilibrium point is the same for the nonlinear and linearized state equations provided the equilibrium state is not a center. To be a center, the eigenvalues are purely imaginary, which will not occur for the latchup equations. In addition, a requirement is that f_1 and f_2 have continuous first order partial derivatives, which is clearly true for the latchup equations.

Consider the plot of $dv_1/dt=0$ and $dv_2/dt=0$, shown in figure 3.1. The two curves will be referred to as C1 and C2, respectively. The intersections of C1 and C2 are the equilibrium or static points. At the intersection points, the slopes of both C1 and C2 are positive. Also, the direction vectors tend toward the curves, instead of diverging to infinity. So, at any equilibrium point, a_{11} is negative and a_{12} is positive, since the gradient of f_1 must point in the negative v_1 , positive v_2 direction. Likewise, a_{22} is negative and a_{21} is positive, since the gradient of f_2 must point in the positive v_1 , negative v_2 direction. Solving the characteristic polynomial for a to obtain the eigenvalues, the solution is:

$$\lambda = \frac{a_{11} + a_{22} \pm \sqrt{(a_{11} + a_{22})^2 + 4a_{12}a_{21}}}{2} \quad (3)$$

Since the slopes of C1 and C2 at an equilibrium point are positive, it immediately follows that the square root term is positive, which implies that the eigenvalues are always real (no center points). If the slope of C1 is greater than C2, then $a_{11}a_{22}$ is greater than $a_{12}a_{21}$, and both roots are negative. In this case, the equilibrium point is a stable node. If the slope of C1 is smaller than C2, then there is one positive and one negative root. In this case, the equilibrium point is a saddle point.

Although the slopes of C1 and C2 can be seen from the plot, it remains to be shown that this behavior is in fact generally true, not just for specific cases. Linearizing at an equilibrium point, the following relations are determined:

$$\frac{\partial f_1}{\partial V_i} = I_{f1} \frac{\partial}{\partial V_i} (C_c + C_{e2}) + (C_c + C_{e2}) \frac{\partial}{\partial V_i} I_{f1} - I_{f2} \frac{\partial}{\partial V_i} C_c - C_c \frac{\partial}{\partial V_i} I_{f2} \quad (4)$$

$$\frac{\partial f_2}{\partial V_i} = I_{f2} \frac{\partial}{\partial V_i} (C_c + C_{e1}) + (C_c + C_{e2}) \frac{\partial}{\partial V_i} I_{f2} - I_{f1} \frac{\partial}{\partial V_i} C_c - C_c \frac{\partial}{\partial V_i} I_{f1} \quad (5)$$

At equilibrium, the first and third terms are zero since I_{f1} and I_{f2} are zero. Derivatives of transistor currents with respect to the opposite emitter-base voltage are zero. So, expanding the definitions of I_{f1} and I_{f2} and keeping only the non-zero terms results in:

$$\frac{\partial f_1}{\partial V_1} = (C_c + C_{e2}) \left(-\frac{\partial I_{B1}}{\partial V_1} - \frac{1}{R_1} \right) - C_c \frac{\partial I_{C1}}{\partial V_1} \quad (6)$$

$$\frac{\partial f_1}{\partial V_2} = (C_c + C_{e2}) \frac{\partial I_{C2}}{\partial V_2} - C_c \left(-\frac{\partial I_{B1}}{\partial V_2} - \frac{1}{R_2} \right) \quad (7)$$

Capacitances are always positive, so the first equation is always negative and the second always positive if base current, collector current, and current through the resistor increases with increasing emitter-base voltage. This will always be the case for physical devices. By similar reasoning, a_{21} is always positive, and a_{22} is always negative under the same conditions. The curves $C1$ and $C2$ will always be perpendicular to their gradients, so the slopes must be positive at an equilibrium point. This formally shows what was asserted from looking at the $C1$ and $C2$ plots for one specific example.

Away from equilibrium points, I_{f1} and I_{f2} are no longer zero. However, the first and third terms in the df_1/dV_1 equation are still approximately zero, because C_c changes slowly and C_{e2} has effectively no dependence on V_1 . This means that $C1$ will be a single-valued function of V_2 . If the gradient always has a positive $x=V_1$ component, then the curve, which is perpendicular to the gradient, will always have a positive $y=V_2$ component. The curve can, and in many cases does, bend "backwards" with respect to the V_1 axis, but will always increase along the V_2 axis as it goes out from the origin. By exactly the same reasoning, the $C2$ curve will be a single-valued function of V_1 . This means that if a given equilibrium point has slope of $C1$ greater than $C2$, then the next

equilibrium point will necessarily have slope of C1 less than C2. From the interpretation of slopes classifying stability, this means that stable points and saddle (unstable) points will alternate.

At the origin where both voltages are zero, it is assumed that the transistor currents are zero. The actual very small leakage from the reverse-biased base-collector junction can be ignored for now. Now, by inspection, I_{f1} and I_{f2} are both zero, so the origin is always one equilibrium point. By setting the appropriate currents to zero in the aij expressions, the slopes of C1 and C2 at the origin are determined as:

$$C1 \text{ slope} = \frac{C_c + C_{e2}}{C_c} \cdot \frac{R_2}{R_1}, \quad C2 \text{ slope} = \frac{C_c}{C_c + C_{e1}} \cdot \frac{R_2}{R_1} \quad (8)$$

Since capacitances are always positive, this implies that the origin is always a stable solution. This unsurprising result formally shows what is trivially observed in physical structures. As V_1 and V_2 are made very large, it is assumed that the current gains become approximately zero, so that collector currents are much, much smaller than base currents. Also, since the base current has an exponential dependence on voltage, it will dominate the current in the shunting resistor. For large V_1 and V_2 , the slopes of C1 and C2 are given by:

$$C1 \text{ slope} = \frac{C_c + C_{e2}}{C_c} \cdot \frac{\partial I_{B1}}{\partial V_1} \cdot \frac{\partial V_2}{\partial I_{B2}}, \quad C2 \text{ slope} = \frac{C_c}{C_c + C_{e1}} \cdot \frac{\partial I_{B1}}{\partial V_1} \cdot \frac{\partial V_2}{\partial I_{B2}} \quad (9)$$

So, for large enough voltages, the slope of C1 will be larger than C2, which implies that the last solution must be stable. Since it was already determined that stable and unstable solutions will alternate, this means that there will be an odd number of solutions. It is easy to show that as long as the slope of the beta curve for the individual transistors decreases with increasing emitter-base forward biasing voltage, there will be no more than three solutions. This will be true for all real devices.

From the mathematical model given for modelling a general CMOS latchup

structure, the following behaviors were derived. First, there will always be a stable solution at the origin where the emitter-base voltages of both transistors is zero. Second, there will always be either one or three static solutions to the system equations. If only one solution exists, then the structure is latchup immune. Otherwise, there will be a stable, static latchup solution, the zero solution, and an unstable solution between the two stable solutions.

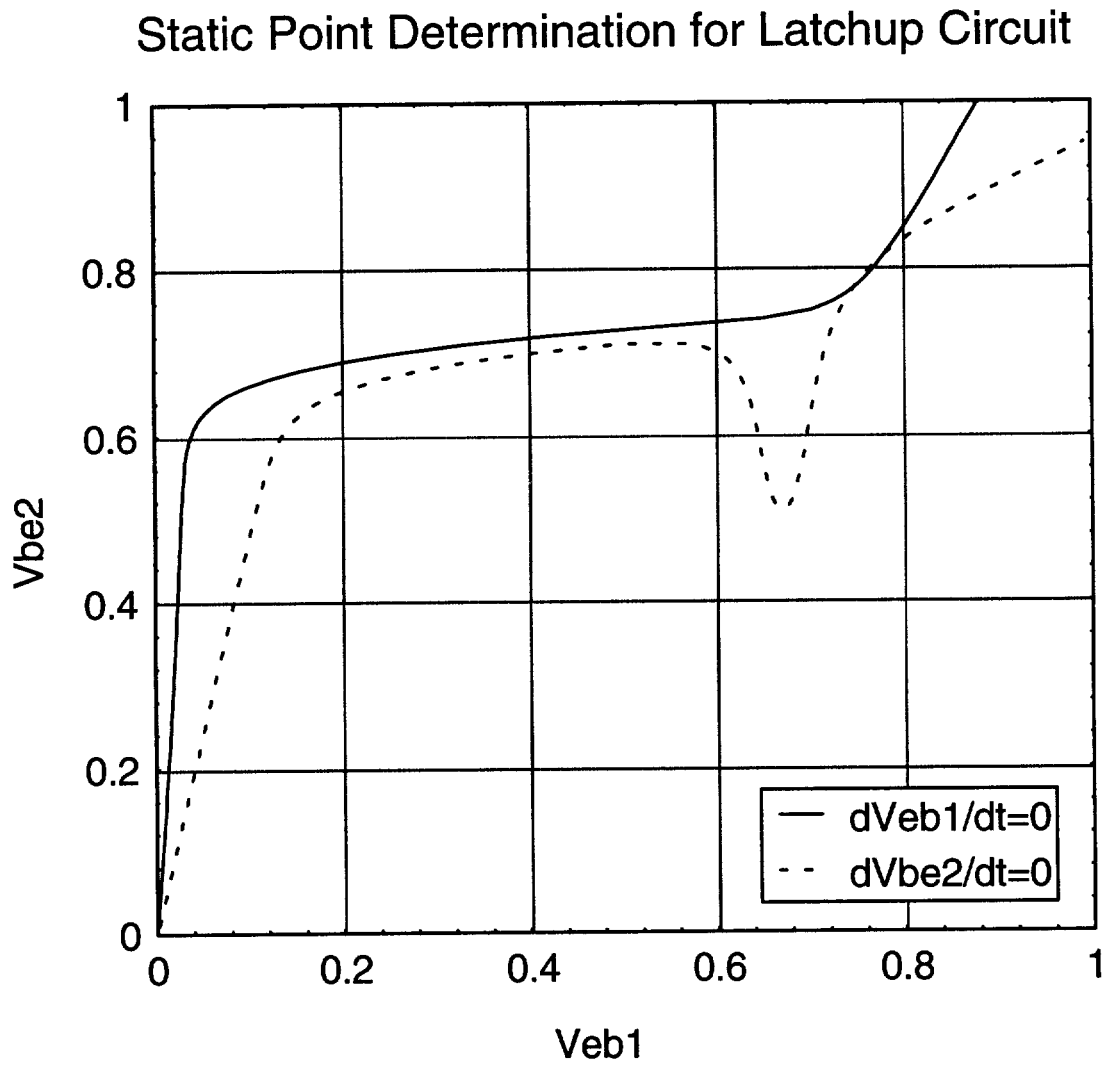


Figure 3.1 Curves of zero dV/dt , intersections are static points.

IV. Analytic Expressions for Latchup (Ker and Wu method)

The equivalent circuit for the parasitic SCR structure is the starting point for the investigation of transient signals causing latchup. Ideally, a method that efficiently yields latchup judgements and matches SPICE simulations is sought, and this section will critically examine the approach of Ker and Wu. Since SPICE simulations must be continued for some time after the pulse ends before a judgment can be made, a latchup criterion which can be applied at the instant the pulse ends is beneficial. Ker and Wu suggest such a criterion and also make some simplifying assumptions so that the circuit equations can be solved analytically. Unfortunately, their method does not agree well with SPICE, despite the authors' claims to the contrary. The latchup criterion developed makes use of time-dependent transistor branch currents, which include the transient currents of junction capacitances. These currents are defined in Equations 1 through 4, as follows:

$$i_{B1}(t) = I_{B1} + C_{e1} \cdot \frac{\partial v_{EB1}}{\partial t} + C_{c1} \cdot \frac{\partial v_{CB1}}{\partial t} \quad (1)$$

$$i_{C1}(t) = I_{C1} - C_{c1} \cdot \frac{\partial v_{CB1}}{\partial t} \quad (2)$$

$$i_{C2}(t) = I_{C2} - C_{c2} \cdot \frac{\partial v_{BC2}}{\partial t} \quad (3)$$

$$i_{B2}(t) = I_{B2} + C_{e2} \cdot \frac{\partial v_{BE2}}{\partial t} + C_{c2} \cdot \frac{\partial v_{BC2}}{\partial t} \quad (4)$$

The element names are taken from the figure showing the circuit equivalent (figure 2.2). The capacitor I's represent the static large-signal base and collector currents. The original form for these equations placed the capacitance terms within the derivative with respect to time. It was claimed that the capacitance will vary much slower than the voltage, and can therefore be moved outside the derivative. In fact, this is an instance where two wrongs do make a right; the capacitance can vary much faster than the voltage, but capacitance should not be within the time derivative from the outset. Time-dependent large signal transient current gains are defined as $b_{1tr}(t) = i_{C1}(t) / i_{B1}(t)$ and $b_{2tr}(t) = i_{C2}(t) / i_{B2}(t)$. In deriving an expression for total supply current -- the current flowing from Vdd, the term $b_{1tr}(t)$ times $b_{2tr}(t)$ minus one appears in the denominator. The authors suggest that the supply current must be positive after the pulse ends for regeneration to occur and sustain latchup, but a rigorous justification is not given. Thus, the latchup criterion is that the transient beta product must be greater than one for latchup to occur. As supporting evidence, the criterion was claimed to be verified by many SPICE simulations.

Now that the "transient beta product greater than one" criterion has been defined, it will be shown that this is not a precise criterion for latchup. As in initial indication that something is amiss, the transient beta product depends on the partitioned values of the well-substrate capacitance, C_{c1} and C_{c2} . From a circuit perspective, however, these capacitances appear in parallel and can be replaced with a single capacitance C_c equal to the sum of the two. So, while the circuit behavior can only depend on the sum, the beta's depend on how that sum is split into

Cc1 and Cc2. Also, only positive values for the transient beta product are considered in the supply current flow arguments. So, the product must go through a value of one in order to reach values greater than one. This suggests that the criterion is satisfied when the supply current becomes infinite (the denominator is zero). However, this type of argument for static latchup has been discredited by Troutman; since the supply current cannot be made arbitrarily large before latchup occurs. Due to the influence of the capacitors, negative beta products are entirely possible. In this case, where one beta is positive and the other is negative, the supply current could be flowing in either direction. This situation will occur when a base current goes through zero. In this case, the transient beta product will flip from negative infinity to positive infinity. This is exactly what happens in figure 4.1. The transient beta product immediately at the end of the pulse is plotted against pulse time. To state it explicitly, for a 10 ns pulse, the product immediately after the pulse ends is -2. For a 12 ns pulse, the product immediately after the pulse ends is +2.5. So, in this example, the criterion is met for a pulse width of 10.2 nsec. The assertion is that this is the minimum pulse time, for this pulse voltage, that will result in latchup. (In actual SPICE simulations, even shorter pulses will latch.)

While these arguments strongly question the validity of the transient beta product criterion, the simplest manner to disprove it is to provide counterexamples. Figure 4.2 shows the emitter-base voltages for a one volt, 79 nsec Vdd noise voltage. The transient beta product following the pulse is 374, so the criterion is met. However, the emitter-base voltages clearly show that the circuit returns to the off state. In order to latch, the pulse would have to be slightly more than 82 nanoseconds long. For these parameters, the window for which the beta product predicts latchup before the actual minimum time to latch is small, so the difference is not very

large. However, for certain circuit parameters, it is possible to find a pulse which satisfies the beta product after a given time but would in fact never latch no matter how long the pulse is. Figure 4.3 shows the emitter-base voltages for a 100 volt, 0.85 nsec Vdd noise voltage. In this case, the transient beta product is only 0.22, so the criterion is not satisfied. However, the circuit clearly goes to the latch state. (Also of interest is that the second transistor is off when the pulse ends and turns on some time later). These two examples clearly show that the transient beta product criterion is unable to accurately predict whether or not a given transient pulse will latch.

The next area to critically analyze is the solution to the circuit equations. After all, if the solutions are accurate, then one would only need to use a better latchup criterion to predict whether the solution will latch or not. The equivalent circuit of figure 2.2 yields two node equations using Kirchhoff's current law, expressed as:

$$I_{C2} - I_{B1} - Cc \cdot \frac{\partial v_{BC}}{\partial t} - Ce1 \cdot \frac{\partial v_{EB1}}{\partial t} - \frac{v_{EB1}}{R_1} + \frac{V_{dd,noise}}{R_1} = 0 \quad (5)$$

$$I_{C1} - I_{B2} - Cc \cdot \frac{\partial v_{BC}}{\partial t} - Ce2 \cdot \frac{\partial v_{BE2}}{\partial t} - \frac{v_{BE2}}{R_2} + \frac{V_{gnd,noise}}{R_2} = 0 \quad (6)$$

In these equations, Vbc is the voltage across the well-substrate junction and is negative if the junction is reverse-biased. The substitution $V_{bc} = -(V_{dd} - V_{eb1} - V_{be2})$ can be made so that the only unknown voltages that appear are Veb1 and Vbe2. Since there are only two nodes, these two voltages completely specify the operation of the circuit. Making the substitution for Vbc and rearranging the equations to isolate the time derivatives results in:

$$\frac{\partial v_{EB1}(t)}{\partial t} = \frac{I_{F1} \cdot (Cc + Ce2) - I_{F2} \cdot Cc}{Cc \cdot Ce1 + Cc \cdot Ce2 + Ce1 \cdot Ce2} \quad (7)$$

$$\frac{\partial v_{BE2}(t)}{\partial t} = \frac{I_{F2} \cdot (Cc + Ce1) - I_{F1} \cdot Cc}{Cc \cdot Ce1 + Cc \cdot Ce2 + Ce1 \cdot Ce2} \quad (8)$$

where

$$I_{F1} \equiv I_{C2} - I_{B1} - \frac{v_{EB1}}{R_1} + \frac{V_{dd,noise}}{R_1} \quad (9)$$

$$I_{F2} \equiv I_{C1} - I_{B2} - \frac{v_{BE2}}{R_2} + \frac{V_{gnd,noise}}{R_2} \quad (10)$$

While the currents and capacitances depend only on V_{eb1} and V_{be2} , the dependencies are sufficiently complex to make analytical solutions (of V_{eb1} and V_{be2} as a function of time) impossible. So, some approximations are made. First, the transistor behavior for a minimum latching pulse time is assumed to progress as follows: initially, both transistors are in an off mode; then, one transistor is on and one off; finally, both transistors are on until the latchup criterion is satisfied. This behavior is not strictly correct; as already shown, it is possible for latchup to be imminent yet only one transistor is on when the pulse ends. The second transistor does not turn on until later. However, this will only occur for relatively large voltage noise pulses and very short minimum pulse times. So, dividing up the circuit behavior into these three modes is valid as long as the noise pulses are restricted to small voltage values. For each mode,

capacitances are taken as constant and set equal to an average value over the voltage range applicable to each mode. In other words, the capacitances are constant for each mode, but change from one mode to the next. Also, the static collector and base currents are approximated as zero below a turn-on voltage and linear beyond it. With these simplifications, the emitter-base voltages will appear in the equations in no higher order than linear terms. Now, from given initial conditions, the voltages as a function of time can be solved using a Laplace transform. For all the modes, the solutions have the form $V_{eb} = A_0 + A_1 e^{-pt} + A_2 e^{-qt}$. To exit the first two modes, the appropriate voltage is set equal to its turn-on voltage. To exit the final mode, the transient beta product must be equal to one. One additional point: while analytical expressions for the emitter-base voltages as a function of time are derived, inverting this to get a time at which a voltage equals some desired value requires numerical techniques. There are no expressions generated for the minimum time to latch, just quantities.

To better illustrate the method of solution, the flow of the computer program written to solve for minimum pulse times implementing Ker and Wu's methods is described. The first step is to assign values to the pairs of voltage points used to calculate slopes and turn-on voltages for both transistors. Also, voltages for the stimulating pulses are chosen. Then, for each operating mode, the steps are as follows:

1. Guess the final voltage values at the end of the mode.
2. Calculate the average capacitances over the mode's voltage range.
3. Calculate the parameters so that the emitter-base voltages as a function of time are known.
4. Solve for the time at which the mode ends.

- 4a. For modes other than the final mode, this implies setting the appropriate voltage as a function of time equal to a turn-on voltage and solving for the end-of-mode time. Then, calculate the other emitter-base voltage at the end-of-mode time.
- 4b. For the final mode, the mode ends when the transient beta product is one. So, the time for which this happens is calculated.
5. Compare the calculated voltages at the end of the mode with the values guessed in step 1. If these are not the same, then make a new guess and go to step 2.
6. The calculated end-of-mode voltages become the initial conditions for the next mode.
7. The minimum pulse width to initiate latchup is the sum of the times in each mode.

The simplifying assumptions unfortunately introduce a great deal of inaccuracy. Initially, the use of constant capacitances will be discussed. The capacitances, which are functions of the voltages, are replaced with average capacitances over a voltage range. If the capacitances were fairly constant over the range, this would be a good simplification. However, the diffusion capacitance of a forward-biased junction is proportional to the exponential of emitter-base voltage and can vary by an order of magnitude over a 50 millivolt range. In addition, the voltages at the end of a mode are unknown until the calculations are done. So, the capacitance is based on an initial guess as to what the emitter-base voltages will be at the end of the mode. If the initial guess does not match what is calculated, then a refined guess is made and new average capacitances calculated. This makes the process very iterative, but avoiding iteration is the prime motivation for moving away from SPICE simulations. Thus, the average capacitances leads to inaccurate results and introduces repetition to the solution method.

The other simplification was claiming that static collector and base currents were linear above some emitter-base turn-on voltage and zero below it. The zero approximation is quite

reasonable with a "normal" value chosen for the turn-on voltage. Above the turn-on voltage, the currents are exponential functions of the emitter-base voltage, and a linear approximation is quite poor. The values for the slope and intercept are going to be critically dependent on where the two fitting points are chosen along the exponential curve. Figures 4.4 and 4.5 illustrate exactly what is meant here. The net effect is that the minimum time to latch varies widely for different linear "fits" for the exponentials. Figure 4.6 shows how the minimum time to latch calculation can change for a four volt Vdd noise pulse. With this much variation, the actual SPICE simulation result will probably fall within the two extremes, and agreement with SPICE can be claimed if one is fortunate in picking the linear fit that matches.

Transient Beta Product for 4V Vdd Pulse

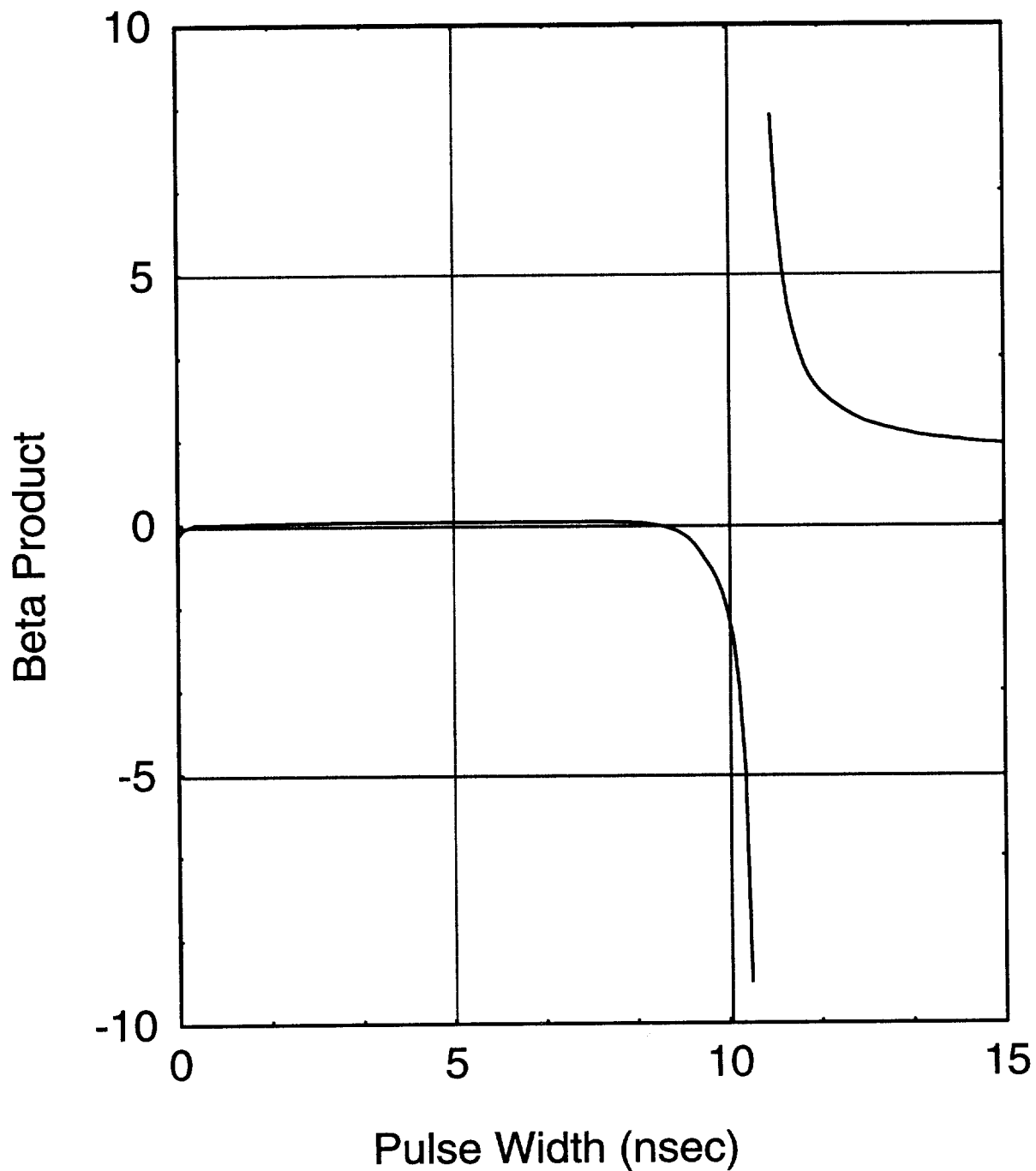


Figure 4.1 Transient beta product immediately after the pulse ends vs. pulse length. A Vdd noise of 4 volts was the stimulus. The product changes sign when the base current of the PNP goes through zero.

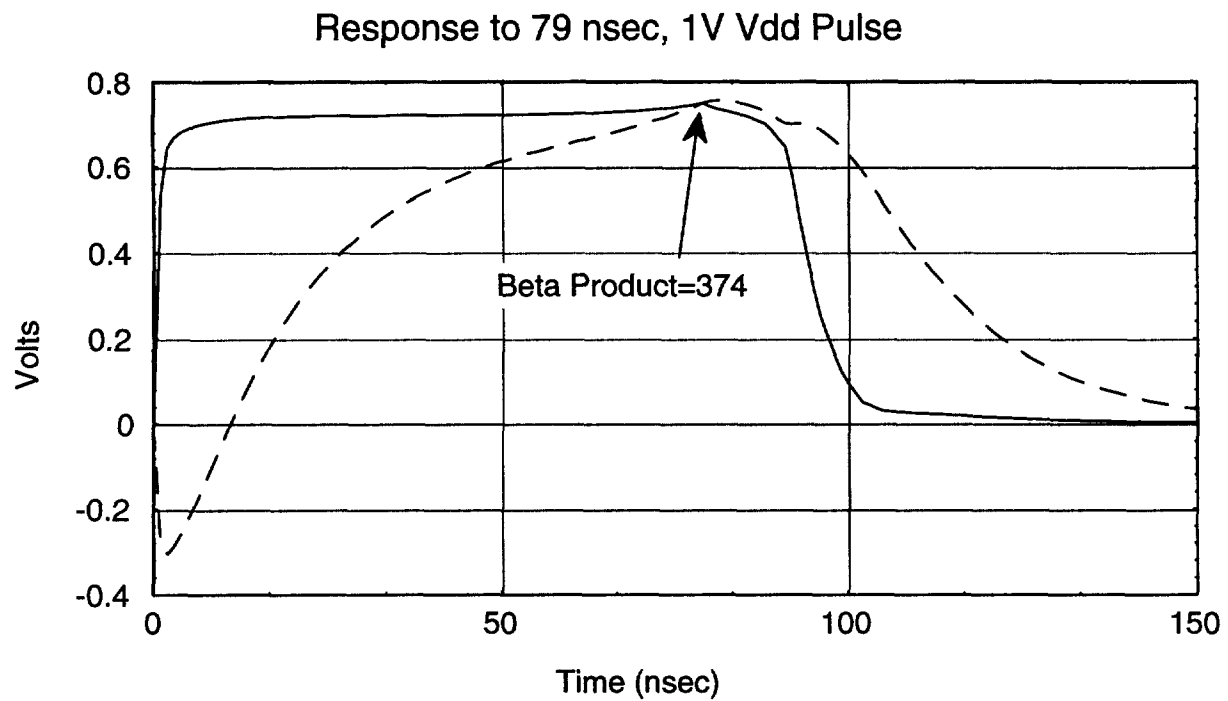


Figure 4.2 Emitter-base voltages as a function of time for a one volt Vdd noise pulse which ends at 79 nanoseconds. The transient beta product at the end of the pulse was 374.

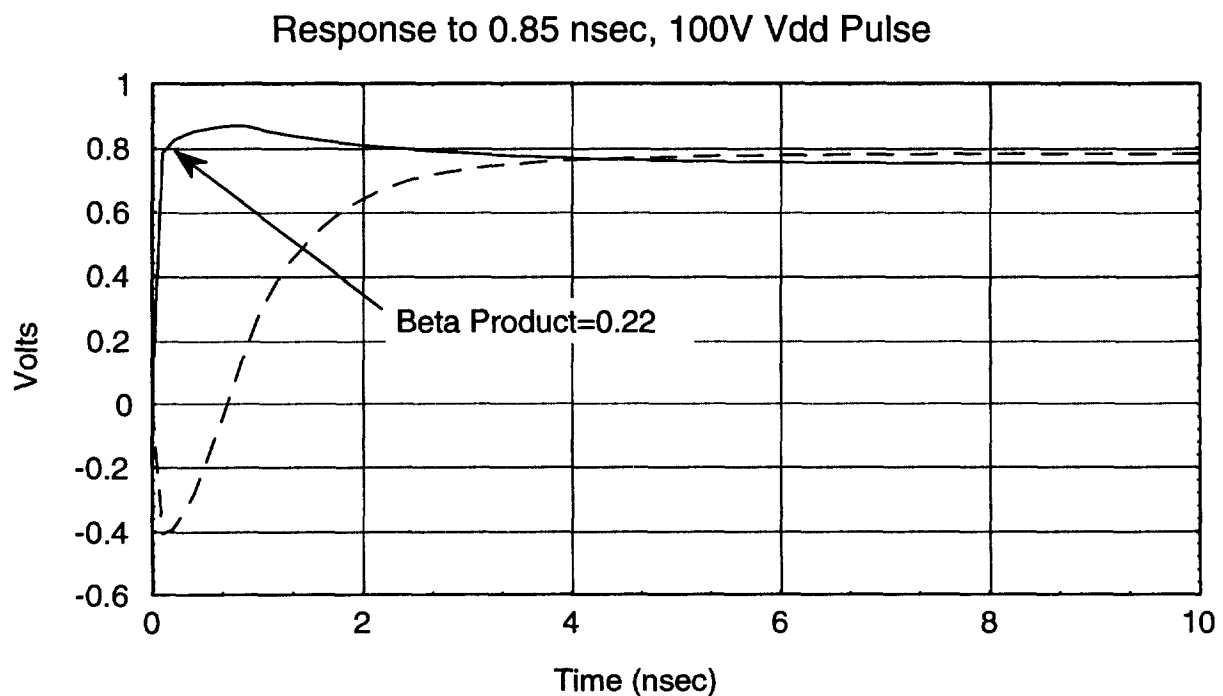


Figure 4.3 Emitter-base voltages as a function of time for a 100 volt Vdd pulse which ends at 0.85 nanoseconds. The transient beta product at the end of the pulse was 0.22.

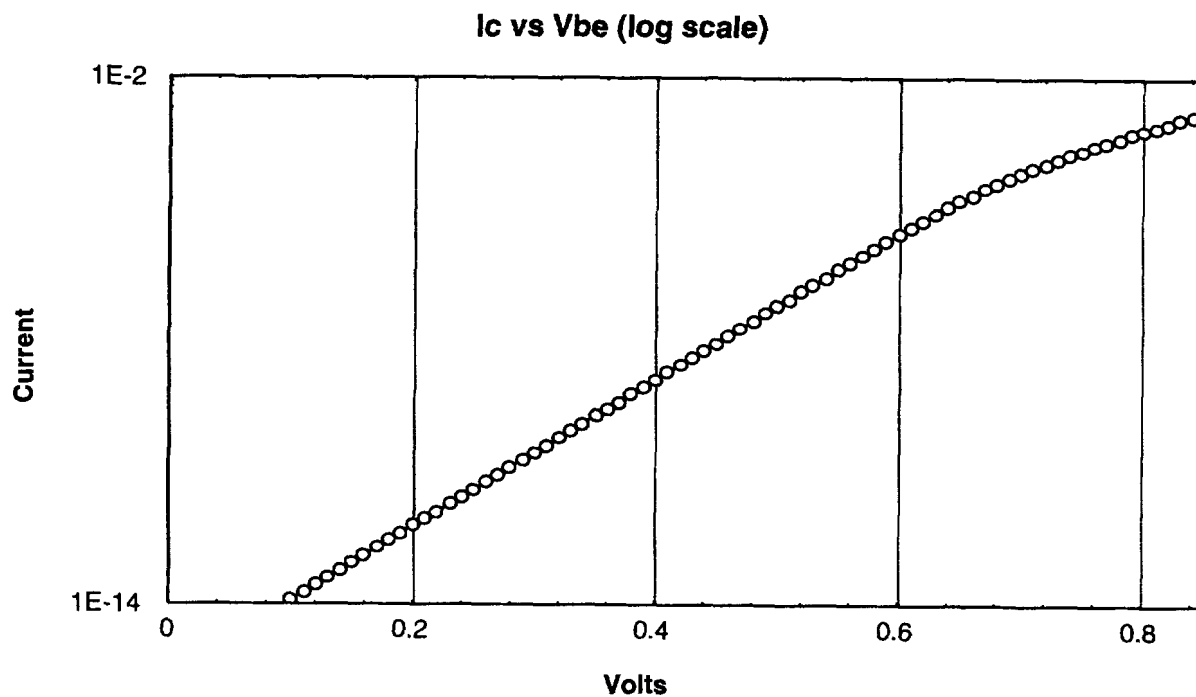


Figure 4.4 Log plot of current vs. Emitter-base voltage, clearly showing a linear dependence.

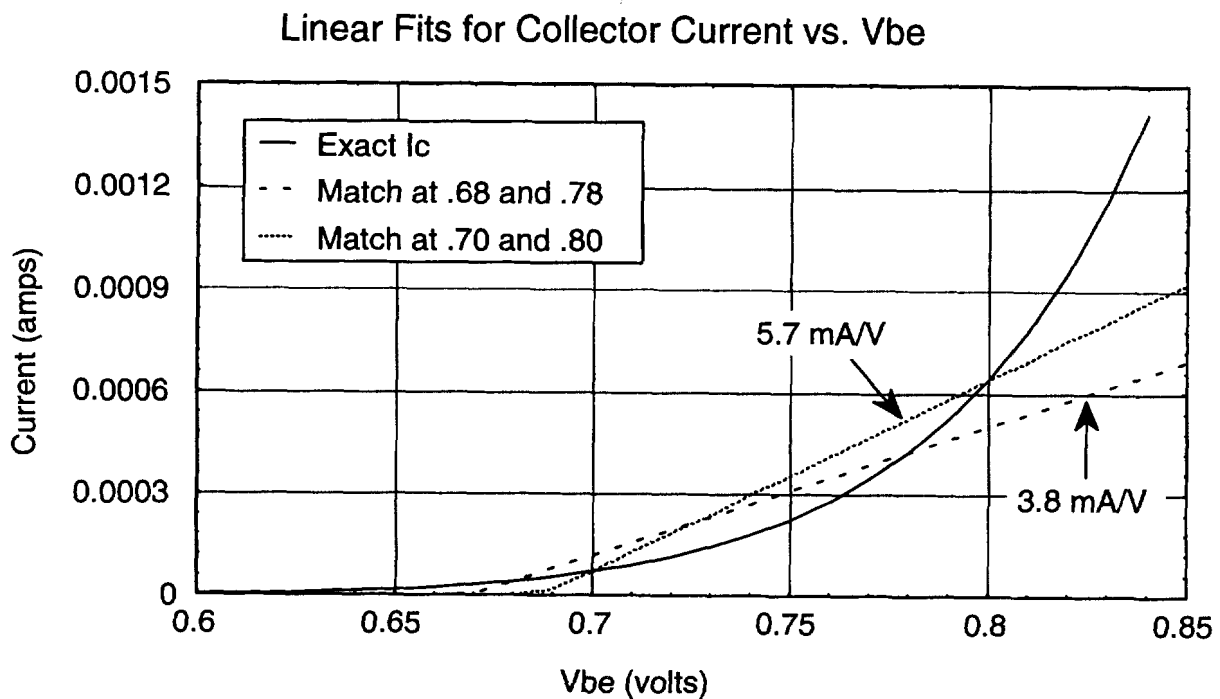


Figure 4.5 Linear fits to the I_c curve. Any linear approximation is a poor fit to the inherently exponential I_c curve.

Time to Latch Variation for Different Linear Fits
(4 volt Vdd pulse)

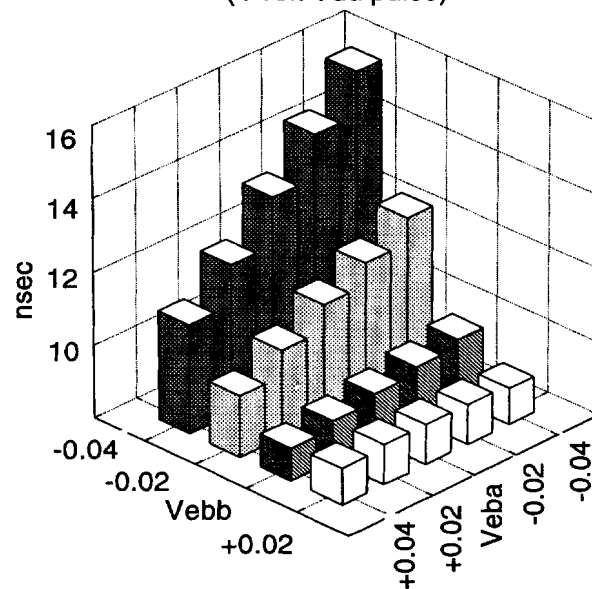


Figure 4.6 The variation in calculated time to latch based on fitting points chosen. Vebb is equal to 0.78 volts, Veba = 0.68 volts.

V. Latchup Criteria Based on Voltage Instability

Since the transient beta product criterion is not an exact condition for latchup, the problem remains of finding a useful and more accurate criterion. The new rule will be based on static solutions of the system equations discussed earlier, in particular the unstable static solution that exists for structures that are not latchup immune. Also, the new rule will provide exact agreement with numerical simulations such as SPICE. As motivation that an unstable, static solution exists; consider the plots of the emitter-base voltages for a four-volt noise voltage of 9.72 ns and 9.74 ns respectively, originally shown in figures 2.4 and 2.5. The shorter pulse returns to the off state while the longer one results in latchup. Note in particular the region in both figures where the slopes in the emitter-base voltages are nearly zero. If dV_{be}/dt is zero, this implies a static solution. It appears that these two near critical pulses approach a static solution of the node equations, and that the relation between the voltages and the exact position of the unstable point will determine whether the circuit latches or not. What is desired is a criteria that can be tested at the moment a pulse ends, so that the critical pulse time can be determined by simply making the pulse longer until the criteria is met. Such a latchup rule, derived from the circuit equations, will be explained.

Switching the subject temporarily, some discussion of the equations for the dV/dt 's is in order. The paper by Ker and Wu resorted to forcing the transistor currents to a simple, but inaccurate, form in order to solve these equations. However, it should be noted that these equations can be solved numerically by using the Runge-Kutta numerical technique to integrate them from some initial known conditions³³⁻³⁵. This method is

effectively the same as using SPICE to calculate voltages as a function of time using transient analysis. Actually, the numerical solution of the pnpn system equations will provide more accuracy than SPICE for the same step size. Figure 5.1 shows the transient response for the same structure and stimulus simulated using TekSpice, pSpice, and Runge-Kutta integration. For the same step size, the Runge-Kutta solution appears to be the best, having less pronounced “kinks” in the curve. When the step size in pSpice is reduced to 0.01 ns from 0.05 ns, as shown in figure 5.2, the solutions are almost identical. In addition, starting at an arbitrary bias point, the equations can be solved backwards in time, giving the path that led to that particular bias point.

The new criteria is based on the system behavior, summarized in figure 5.3. Here, the separatrix through the unstable point has been added, which separates the stable off region from the stable latch region. Once any stimulus pulse has ended, all points under the separatrix have trajectories that will eventually lead to the neighborhood of the off static solution. Likewise, all points on the other side of the separatrix will lead to the latch static point. This is precisely the new latchup criteria. Since it is based on the mathematical behavior of the system equations, agreement with SPICE is guaranteed. This is a significant advantage over a criteria that cannot be derived and merely is motivated by the steady-state criteria. Furthermore, this rule can be tested at the moment a pulse ends, so that the critical pulse time can be determined by simply making the pulse longer until the criteria is met. Using this “voltage instability” criterion, the numerical solution will now be discussed, saving the analytical solution for a later chapter.

The initial step is to determine the emitter-base voltages of the unstable, static solution where $dV_{eb1}/dt=0$ and $dV_{be2}/dt=0$; if one exists. The equations can be simplified to solving for $I_{f1}=0$ and $I_{f2}=0$, since these must both be zero if both dV/dt 's are zero, which decreases the computational effort. This simplification is equivalent to noting that capacitors behave like open circuits in a static situation. Beforehand, we do not know the neighborhood of the unstable point, so a Newton-Raphson technique will likely fail to converge on the unstable point starting from an arbitrarily chosen seed value. So, the procedure being used is to solve for the V_{be2} values which make I_{f1} and I_{f2} zero as a

function of V_{b1} , using a bisection method. V_{b1} is started from a small value (but larger than the off state value), and increased in small steps. The difference in the V_{be2} values is checked and the procedure stopped when this difference changes sign. Now, this is in the neighborhood of the unstable point, and this value can be used as a seed value in a Newton-Raphson routine. It is possible that there is only one solution, namely the off state point. Here, the circuit is immune to latchup, since no stable latch point exists. In this case, the difference in V_{be2} values never changes sign, and the absence of an unstable point is noted. Another possibility is that the unstable point and latch point are so close to each other that the V_{b1} step “skips” past the unstable and the unstable point will appear to be absent. This will only happen for a circuit that is very close to being latchup immune, and the only remedy is to decrease the step size in the V_{b1} steps during the successive bisection solutions. In terms of the curves of zero dv/dt , shown originally in figure 3.1, the bisection is measuring the distance along the V_{be2} axis between the two curves at a particular V_{b1} value. Then, the V_{b1} is stepped and the process continues until the distance along the V_{be2} axis between the two curves changes sign, or the V_{b1} reaches a large cutoff value.

By zooming in on the region of the unstable solution and plotting the dv/dt vectors, figure 5.4 illustrates visually how the voltages will change in time in this vicinity. Near the upper intersection (the stable latch point), there are two areas where the arrows point “inward”, indicating stability. The arrows in the other two areas will lead to one of these two areas. Near the unstable point, the upper right arrows lead to the latched state, while the lower left arrows point away towards the off state. Again, the upper-left and lower-right arrows will lead to one of these two areas. With the bisection and Newton-Raphson techniques for root finding, the unstable point has been calculated; unless it does not exist.

The next step is to find the separatrix curve going through the unstable, static point which separates the off region from the latch region. By starting out at some bias points very close to the unstable point, the path which leads there can be calculated using Runge-Kutta techniques going backwards in time. An offset of 10 microvolts was

somewhat arbitrarily chosen since it provided more than sufficient accuracy for the separatrix solution without causing loss of precision in the matching floating-point representation. One pair of points chosen were V_{be2} equal to its unstable solution value plus the offset, which is guaranteed to lead to the latched state; and V_{be1} equal to its unstable solution value minus the offset, which is guaranteed to lead to the off state. Switching the signs of the offset produce the other pair of points chosen. These pairs trace out paths very close to each other which sandwich a “critical” curve--the separatrix, as shown in figure 5.5. The two resulting curves which sandwich the critical curve are averaged, and this is output as the best solution to the critical curve. This critical curve is used as the new latchup criterion. Specifically, if at the instant a pulse ends, the emitter-base biases lie beyond the critical curve, then the circuit will latch. For a given pulse amplitude, the voltages can be calculated until the critical curve is crossed, which gives the minimum time for the pulse to just cause latchup. This rule no longer requires an iterative approach, since the rule is checked at the instant the pulse ends. Simulation with SPICE alone, of course, does require many iterations, since the simulation must be continued long after the pulse ends before the voltages will settle to either the latch or off region. In addition, since the equations are the same as for SPICE, essentially exact agreement with SPICE is achieved.

To further illustrate the equivalence to SPICE simulations, note the path traced to the off state in figure 5.5. Near the unstable point, it is in a region where both voltages decrease. Then, it crosses the $dV_{be2}/dt=0$ line and V_{be1} continues to decrease while V_{be2} increases slightly. Finally, it crosses the line again and both voltages decrease again approaching zero. Now, compare the SPICE simulation shown in figure 5.6, using the same latching structure parameters used in generating the previous figure. Here, at the instant the pulse ends at 9.7 ns, the voltages are very close to the critical curve, but still in the off region. Initially, V_{be1} decreases and V_{be2} increases until the unstable point is almost reached. Now, both voltages remain nearly unchanged for some time and eventually both decrease. Then, at approximately 80 ns, V_{be2} has a slight increase before both voltages decrease towards zero. This slight bump in V_{be2} is precisely what was

expected from the critical curve tracing.

To find the minimum pulse times for given pulse amplitudes, it is possible to run SPICE simulations and output node voltages for a sufficient length of time. This time would be longer than the suspected minimum pulse time, but would not have to be as long as using SPICE alone where the voltages must be traced until they settle. Then, the voltages can be checked for when it crosses the critical curve and the minimum pulse time is determined. However, it is much more efficient to avoid SPICE altogether. Simply solve the equations using the same Runge-Kutta routine used previously, only now with a stimulating pulse and starting with zero volts as both emitter-base voltages. The solution can be stopped as soon as the critical curve is crossed. If the curve is not crossed after some maximum allowed solve time, then the chosen pulse amplitude is not sufficient to trigger latchup. Results of using this method compared to using SPICE yield a difference of approximately one-tenth of one percent in the minimum pulse times.

Vbe Comparison (0.05 ns step)

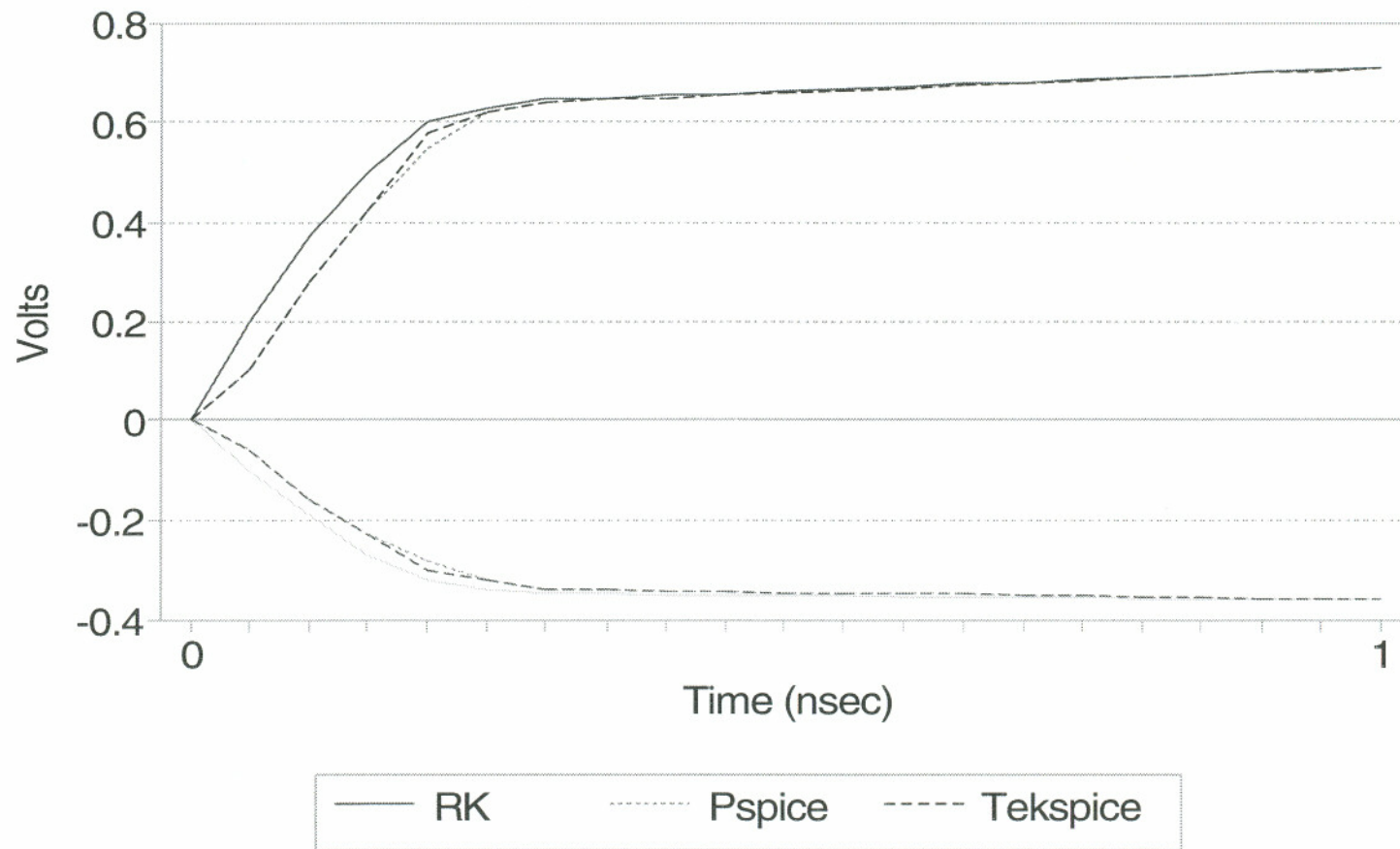


Figure 5.1 Comparison of Runge-Kutta direct solution, pSPICE, and TEKSPICE with the same step size used for each.

Vbe's: Pspice vs RK (.01ns step Pspice,.05ns step RK)

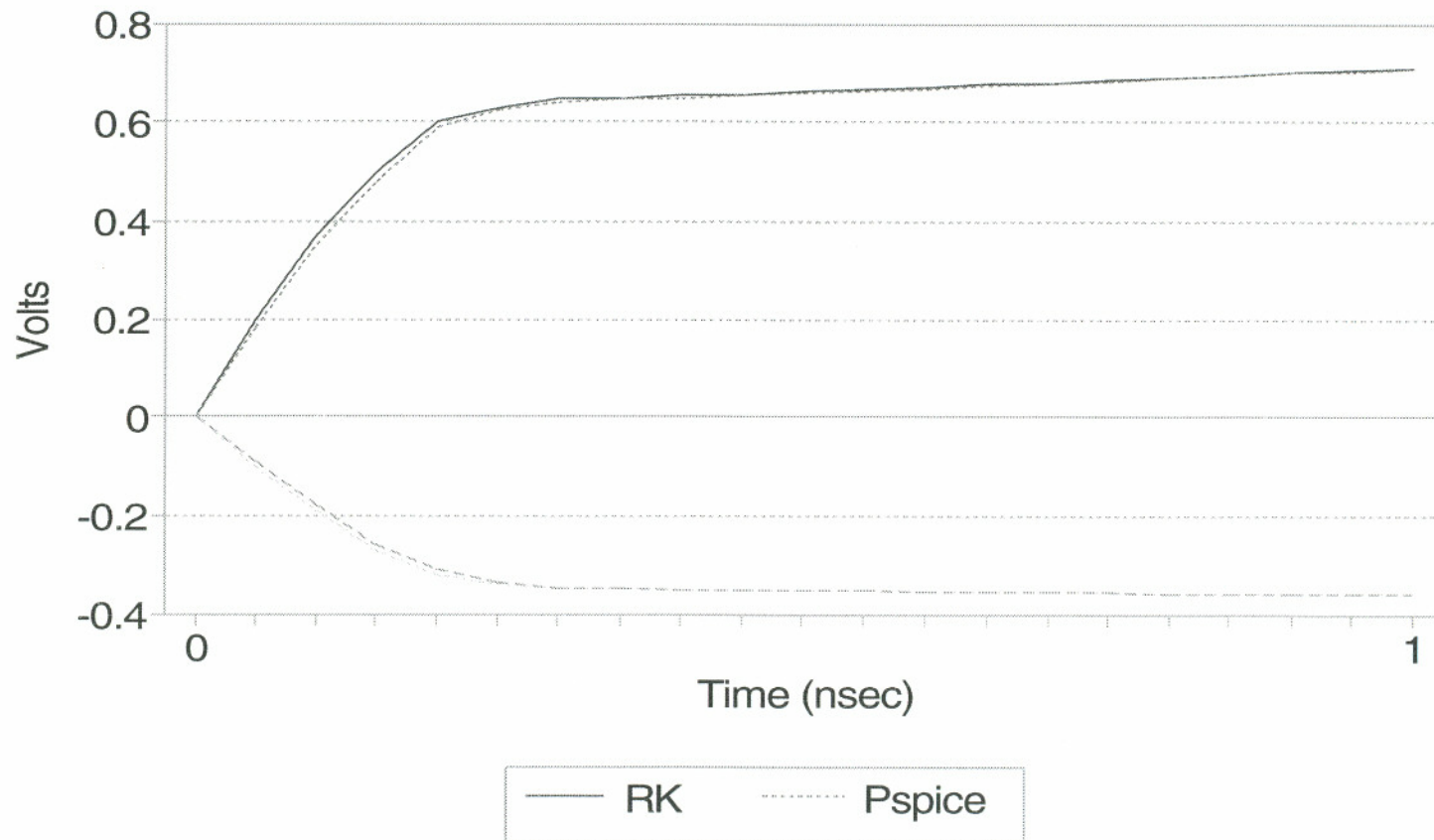


Figure 5.2 Comparison showing that Runge-Kutta with a 0.05 nanosecond step-size is roughly the same accuracy as pSPICE with a 0.01 nanosecond step-size.

Parasitic PNPN Latchup Behaviour

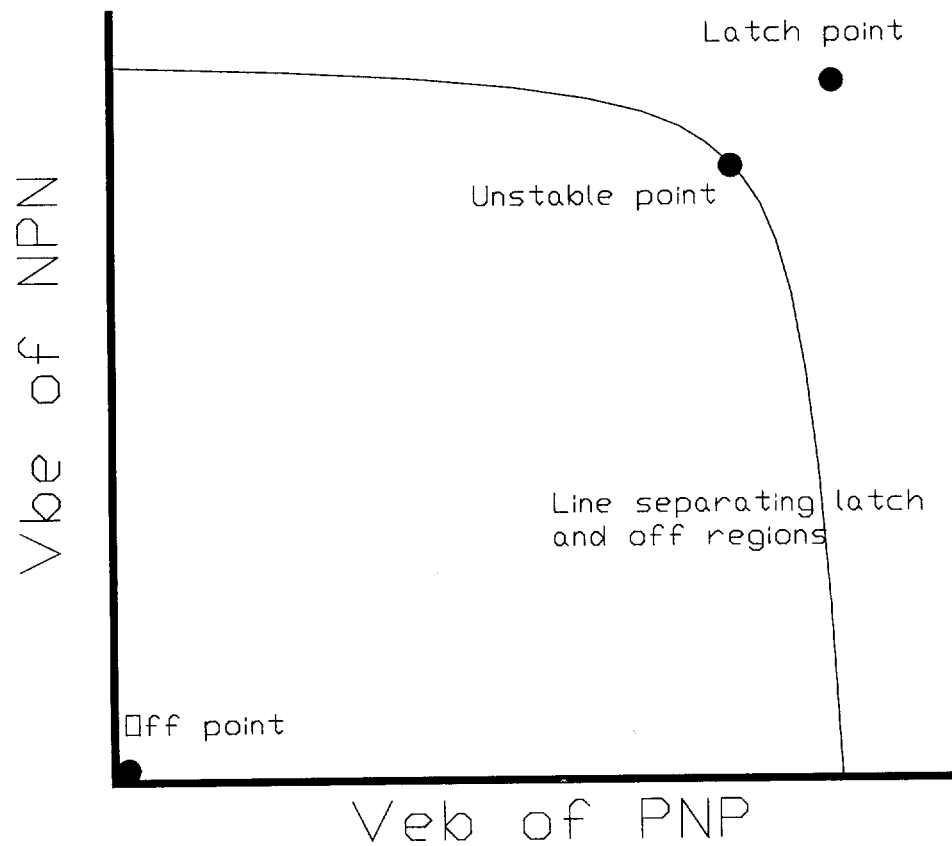


Figure 5.3 Illustration of three static points and line separating the two stable solutions.

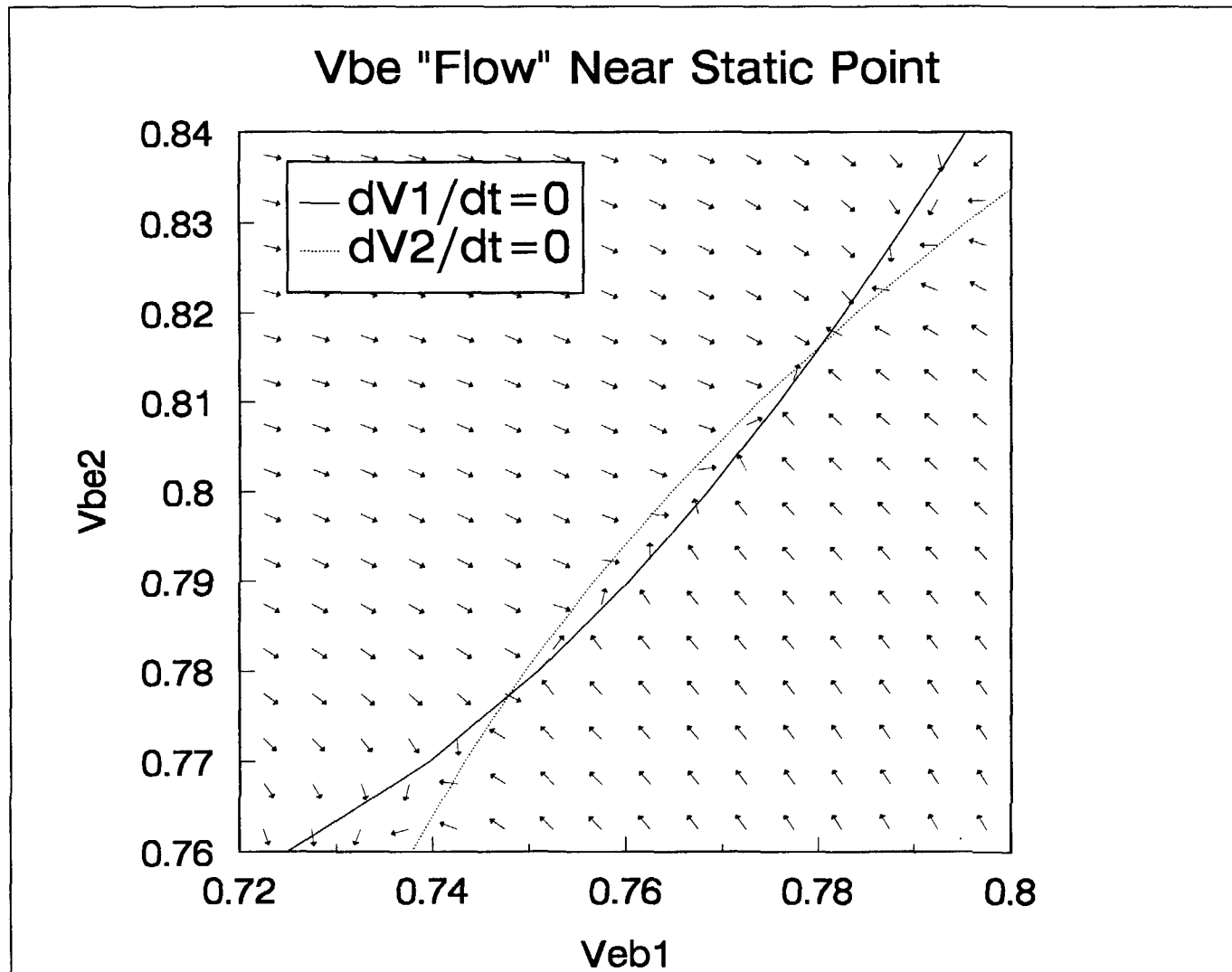


Figure 5.4 Vector flow showing how emitter-base voltages will change in time in the vicinity of the unstable static point.

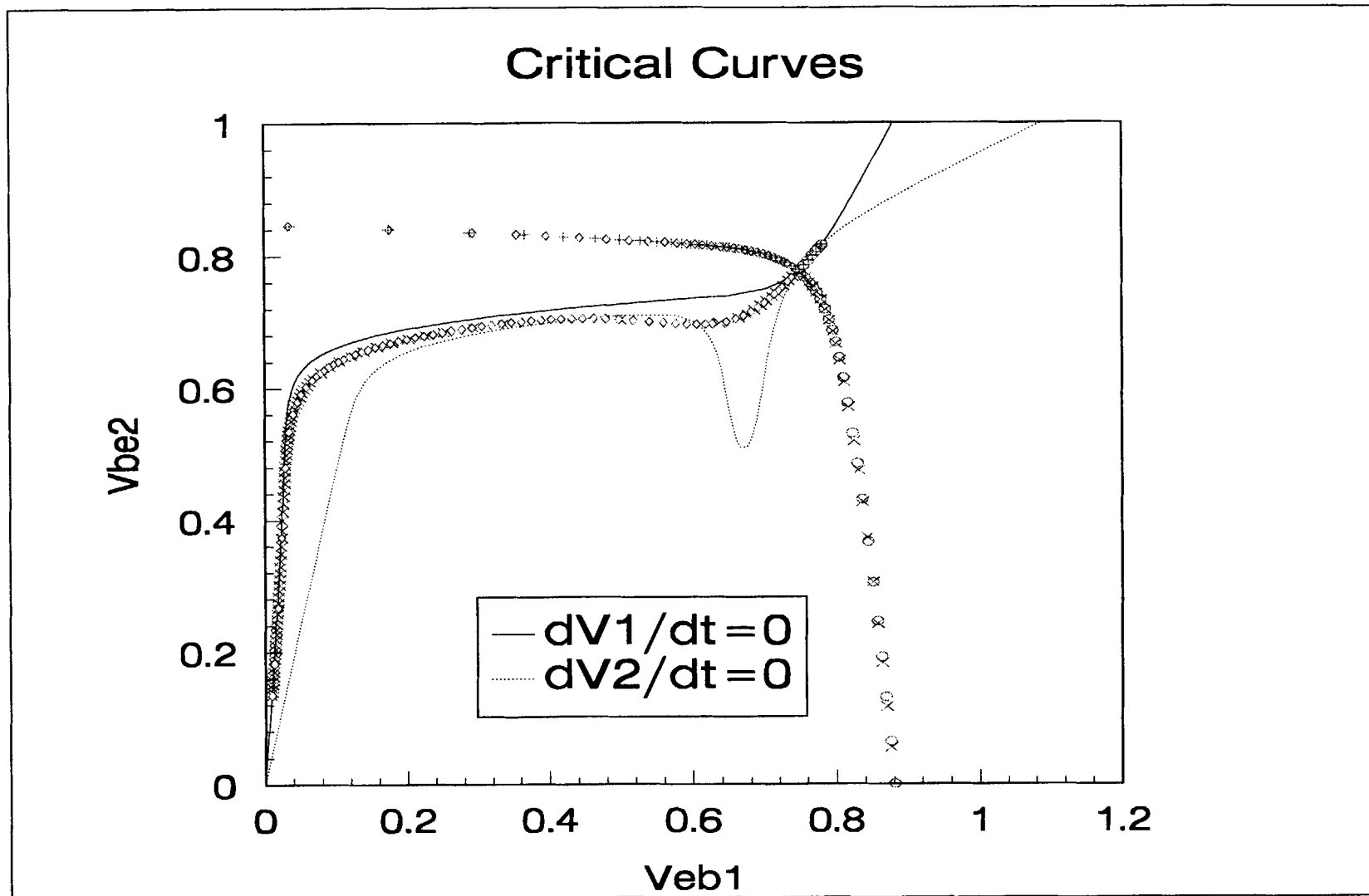


Figure 5.5 Calculation of critical curve. The critical curve will lie between the diamonds and plus signs at the top, and between the circles and multiplication signs at the right.

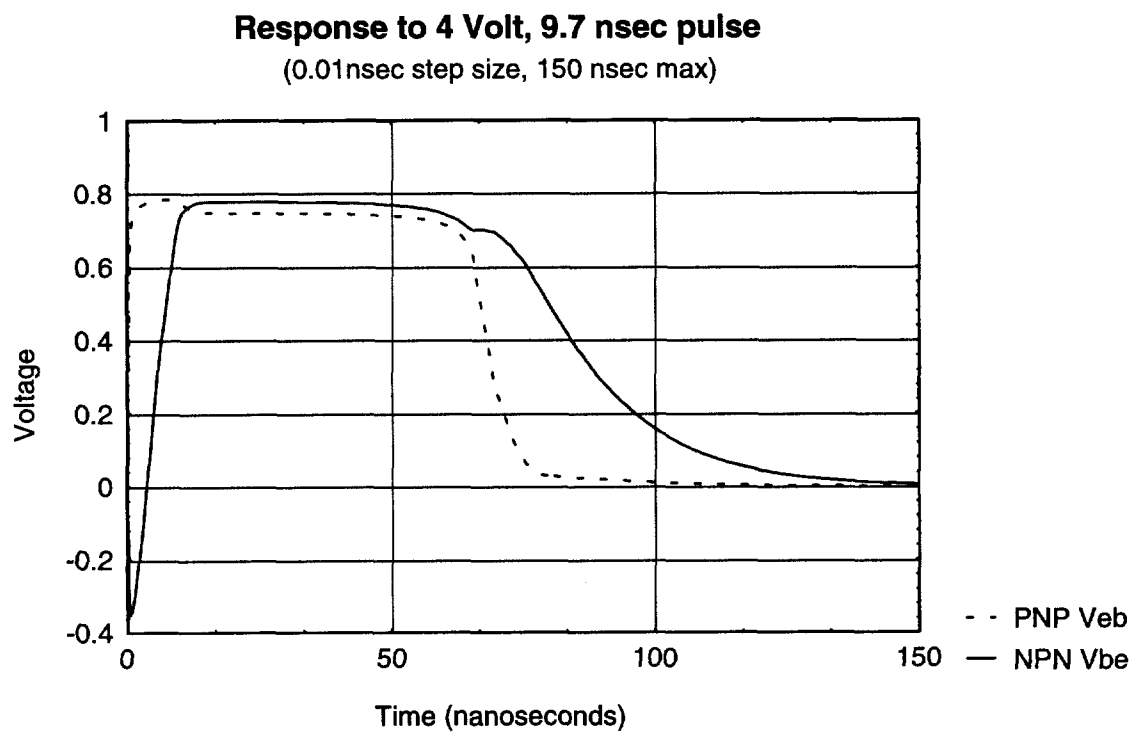


Figure 5.6 Transient response showing slight increase in V_{be} of the NPN at 80ns, after it has started decreasing.

VI. Latchup Predictions for Intel Devices

This section will provide numerical results for the method outlined in the previous chapters. The IV curves (base and collector currents versus emitter-base voltage) were provided for two different CMOS inverter test structures. For the first structure, there was little information as to the physical layout, but the IV curves of the two parasitic transistors were known (not reproduced here). The structure will be referred to as the “A” structure (the original designation referred to the epitaxial layer depth and has been omitted for confidentiality). Points were read off of the chart since the numbers making up the plots were unavailable. These points were fit to a Gummel curve with a small subset of the parameters. Due to the inaccuracies in reading the data off of the chart, doing a statistically precise fit would have been wasted effort. Figure 6.1 and 6.2 plots the transistor base and collector actual currents and the model currents for the two transistors. Figure 6.3 summarizes the errors between the actual and model currents and lists the model parameter values. These parameters were used to compare the simulation results for DC latchup versus the previously measured values. For NPN initiated latchup, the base trigger current was simulated as 939uA, measured as 650uA. The V_{be} at latch was simulated as 0.85V, measured as 0.847V. For PNP initiated latchup, the base trigger current was simulated as 5.88mA, measured as 5.75mA. The V_{be} at latch was simulated as -1.24V, measured as -1.15V. The N-well resistor value was increased by 25% from the layout calculated value to obtain better agreement--the DC results were used to calibrate the model for the transient pulse latchup predictions. Figure 6.4 is the minimum pulse width versus trigger current curve. Remember that the internal trigger current is equivalent to an external trigger voltage. Figure 6.5 through 6.10 show how the minimum pulse width is affected for varying model parameter values. For all these

curves, the trigger current was fixed at 20 mA (equivalent to a 4 volt trigger voltage). From these, it is clear that the values of the shunting resistors have the largest effect on the time of the latching pulse. The next set of nine figures (6.11-6.19) produces an entire set of minimum pulse time curves for all combination of parameter variation, instead of varying just one parameter at a time as in the previous figures. Here, the stimulus is a voltage pulse on either the external base connection of the NPN or the PNP, representing a noise at that point on Vdd or Vss respectively. The values on R_w , R_s , and C_{jc0} were simulated at -30%, nominal, and +30%. Figure 6.20 takes just the set of points for a Vdd noise of 1.5 volts and R_s nominal and plots the variation in minimum pulse times for varying R_w and C_{jc0} . Clearly, once again the resistance values have the largest impact on the pulse times.

The other structure, referred to as the “B” structure, had a layout as shown in figure 6.21. This structure had the silicon doping regions as is found in a CMOS inverter, but there were no MOS transistors which would complicate the analysis. A complete table for the collector and base currents of the two transistors was provided, and figure 6.22 and 6.23 plots this data and the resulting beta curve for the NPN transistor. These curves were then modeled by the usual Gummel-Poon model and a least-squares fit for the various regions was performed to determine the model parameters, as shown in figure 6.24. Although these figures only show the NPN transistor, the same procedure was used for the PNP. In the high current region, a non-linear model for the shunting resistance was used to give a better fit than a single resistor value could. The excellent agreement between the experimental data and the fit parameters is clear from figures 6.25-6.27. The base and collector currents for both the NPN and PNP show the near overlap in model and experimental data. The differences are easier to detect on the NPN beta curve, but the agreement is still quite good. The statistical parameter fit resulted in a complete specification of the equivalent circuit transistors and emitter-base shunting resistances. Junction capacitances were estimated from the geometry and doping. Capacitances are not required for the (static) determination of whether or not the structure is latchup immune. The capacitance, as expected, only enters the equations when solving for voltages as a function of time.

The first step was to determine if the structure was latchup immune, and to vary the circuit parameters to find the range under which it was latchup immune. All of the programs written to

implement the method are listed in the Appendix. The circuit transistor currents and capacitances were calculated using `equation.c`, which all of the programs would call. The curves for which the derivative of each base-emitter voltage with respect to time was zero were generated using `zero.c`; the intersection of these curves are the static points. The results, summarized in figures 6.28-6.33, clearly indicate that the structure was latchup immune. The structure did not exhibit a stable latched point unless the betas of the transistors were increased by a factor of 50 or the shunting resistors were increased by a factor of 100. The important feature of this structure which makes it so immune to latchup is the extremely low values for the shunting resistors. With such small resistances, a large quantity of current must flow through them to sustain a 0.7 volt emitter-base voltage drop, required to turn-on the transistors.

For a structure which does have a stable latch point, the minimum pulse width to cause latchup is found in a two-step process. First, the program `critcrv.c` determines the coordinates of the unstable point and solves the differential equations backwards in time to trace out the critical curves separating the stable off and stable latch regions. The Runge-Kutta numerical routines to accomplish this are modified versions of the routine appearing in Numerical Recipes in C³⁵. Second, `pulse.c` finds the time for a given pulse to cross the critical curve, thus entering the latch region, and will generate the results for a range of pulse voltages. As indicated in a previous chapter, this is much more efficient than using iterative SPICE simulations.

The conclusions that can be drawn from these figures is that the latchup immunity of a structure can be determined very quickly using the voltage-instability criteria. In a circuit simulation, the immunity can be determined by setting the base node to initial values such that the transistor on driven extremely on. Then, let the circuit relax to a steady state and see if that state is at zero volts or not. While this is not particularly time consuming, it does not give an indication of the magnitude of parameter change required to have a latch state. Unlike the previous charge-storage criteria, the voltage-instability criteria gives accurate results.

In order to quantify the difference in time between the new method and the iterative SPICE method, benchmarks were performed. For the 54 pulse width versus stimulus voltage curves produced in the preceding figures, the computer solve time was recorded for both methods. In each case, the same 486 PC was used, running DOS. For the new method, each

curve took three minutes, but only for unique parameter variations. The curves for both Vdd and Vss pulsing were done at the same time. So the total time was 81 minutes to generate the 54 curves. By doing SPICE simulations, the 54 curves require about 1000 data points, since about 20 points per curve are calculated. Each point requires 14 SPICE simulations to narrow down the minimum pulse width to acceptable accuracy. Each iteration takes 30 seconds to run. So, the total time is about 7000 minutes, or 120 hours to generate the 54 curves. Just to be explicit, these comparisons are for generating the identical curves and approximately the same accuracy. The factor of about 85 times slower clearly shows the advantage of the new method.

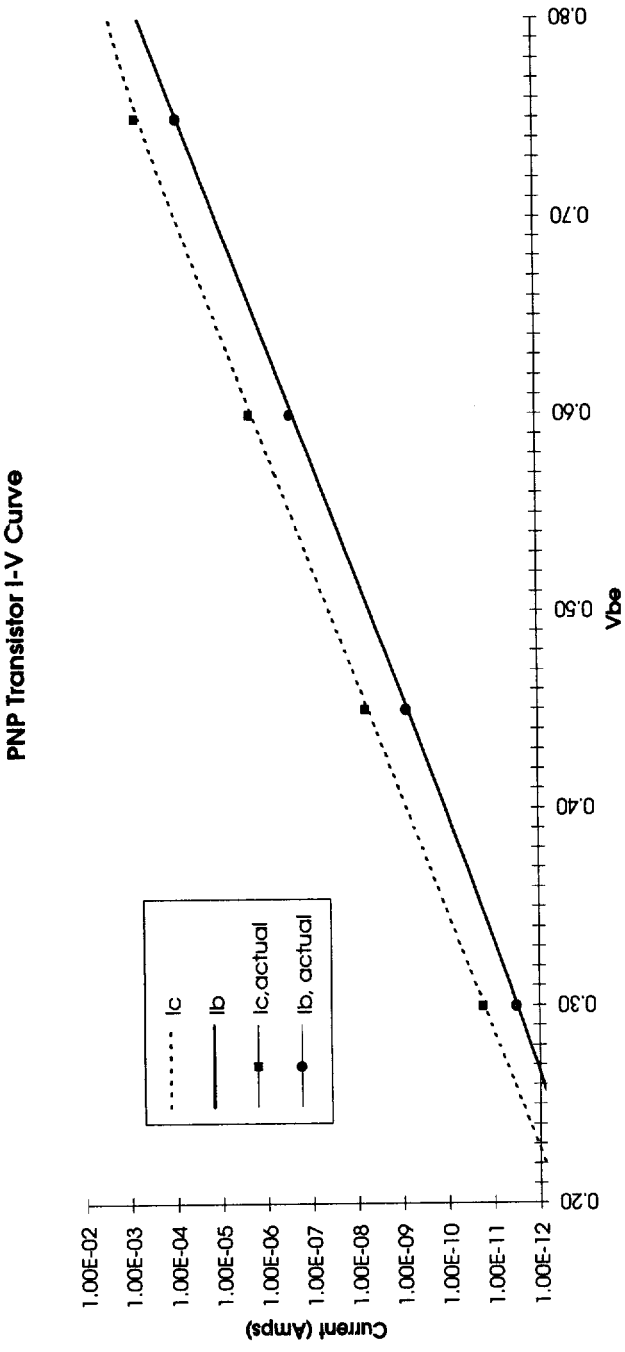


Figure 6.1 PNP transistor curve showing actual measurements and fit curve

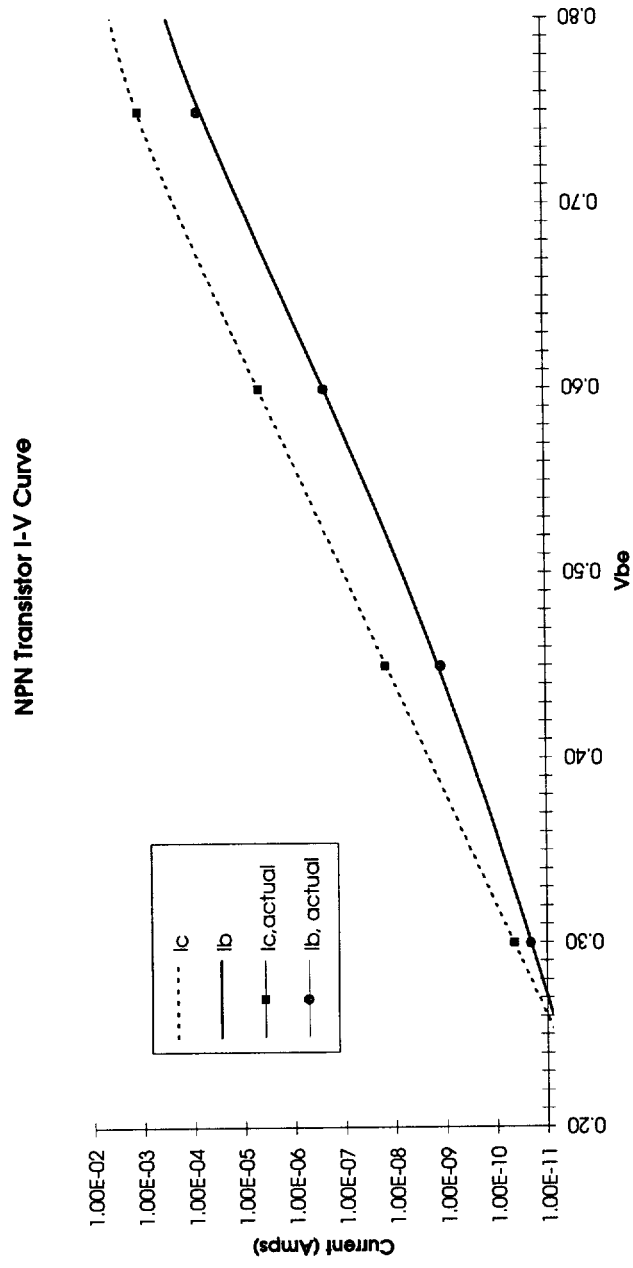


Figure 6.2 NPN transistor curve showing actual measurements and fit curve.

NPN “A” Datapoints

Vbe	Ic	SPICE delta	Ib	SPICE delta	Beta
0.3	4.37e-11	+1.4%	2.03e-11	-0.5%	2.15
0.45	1.45e-8	-3.4%	1.16e-9	+13.8%	12.5
0.6	4.40e-6	+0.5%	2.23e-7	+1.8%	19.0
0.75	1.00e-3	+4%	6.70e-5	-13.6%	14.9

PNP “A” Datapoints

Vbe	Ic	SPICE delta	Ib	SPICE delta	Beta
0.3	1.74e-11	+0.6%	3.16e-12	+0.0%	5.5
0.45	6.12e-9	-3.1%	7.59e-10	+4.1%	8.06
0.6	2.03e-6	-1.0%	2.51e-7	+1.2%	8.09
0.75	5.93e-4	-7.6%	7.13e-5	+2.8%	8.3
0.90	6.51e-3	-2.9%	1.16e-3	+2.15%	5.6
1.05	1.10e-2	+8.2%	3.98e-3	-7.33%	2.8

Derived SPICE Parameters

SPICE symbol	PNP	NPN
Is	1.53e-16	4.3e-16
Nf	0.995	1.0073
Beta (forward)	8	22
Rb	55	50
Ise	4.5e-16	1.3e-14
Ikf	8e-3	5e-3

Figure 6.3 Derived SPICE parameters for “A” test structure

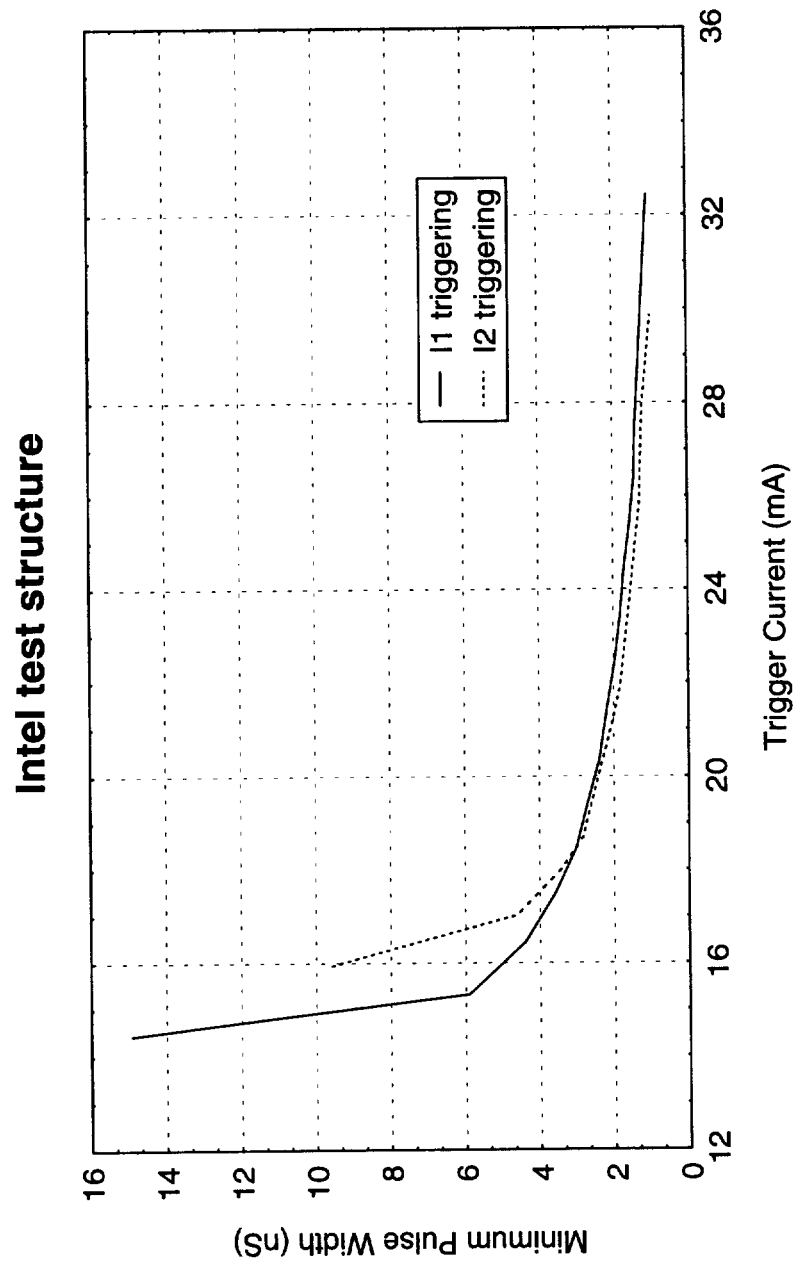


Figure 6.4 Curve of trigger current versus minimum pulse time to cause latchup

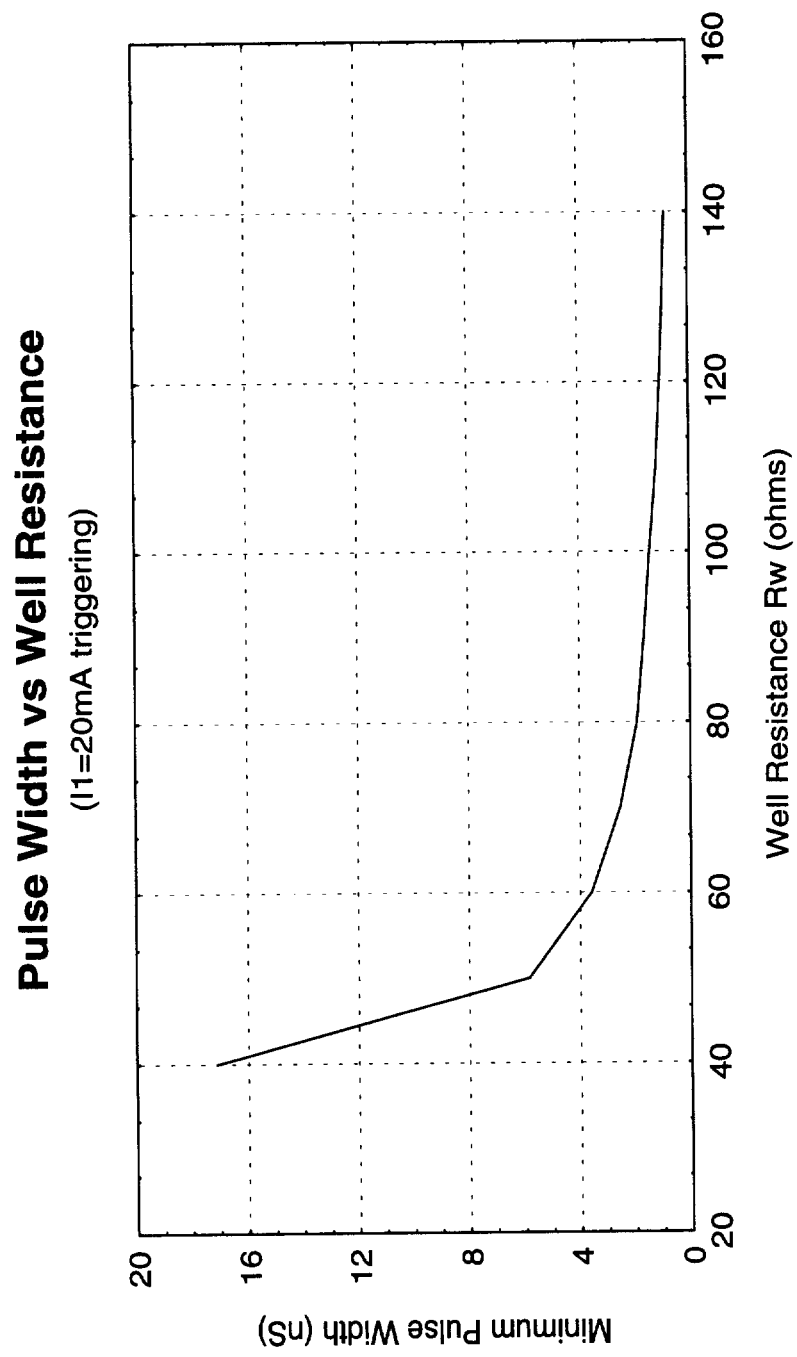


Figure 6.5 Effects of varying well resistance on time to latchup

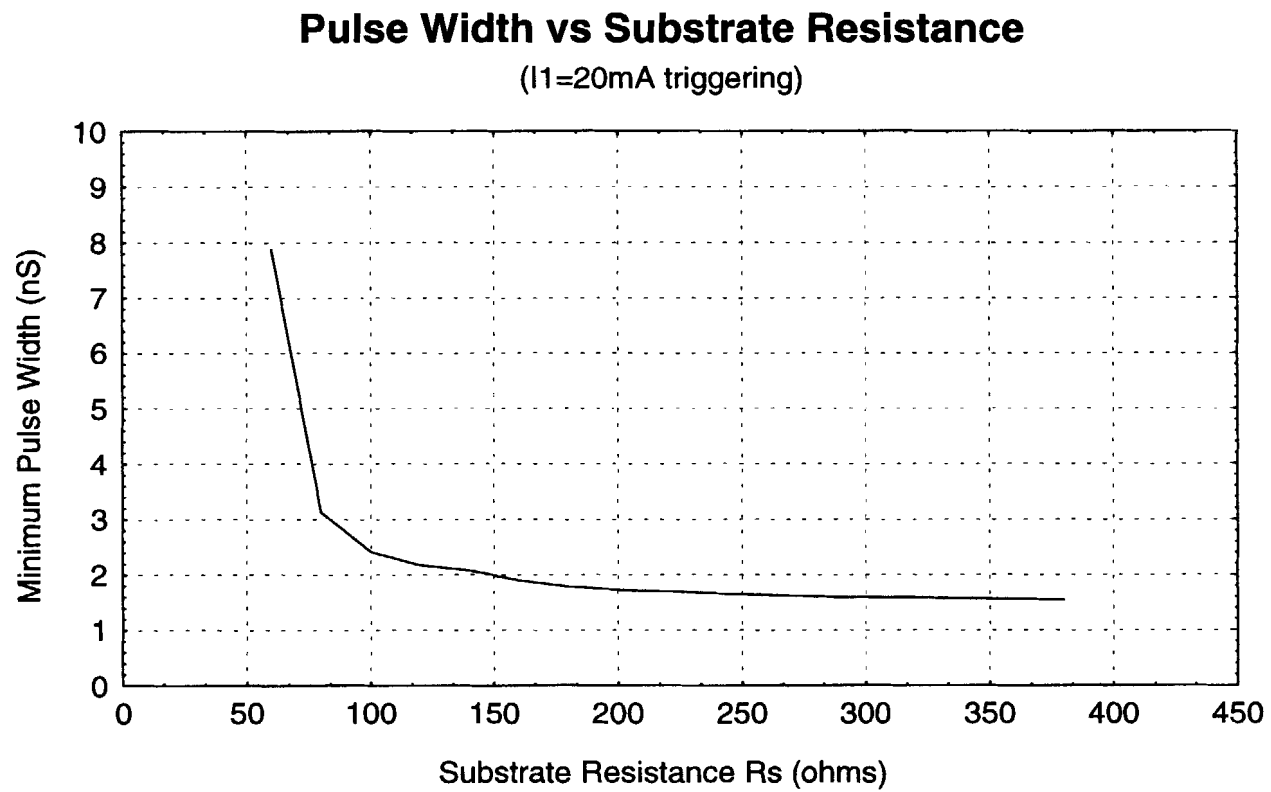


Figure 6.6 Effect of varying substrate resistance on time to latchup

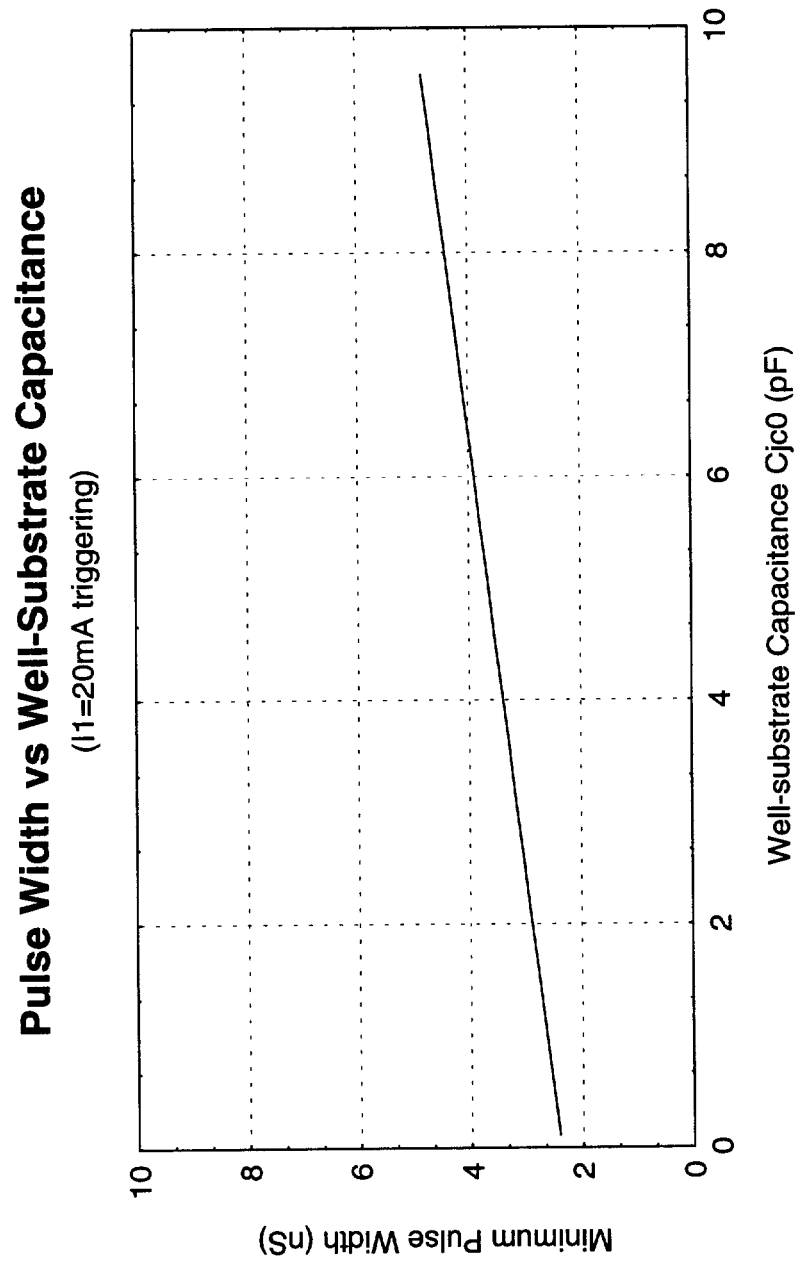


Figure 6.7 Effect of varying well-substrate capacitance on time to latchup

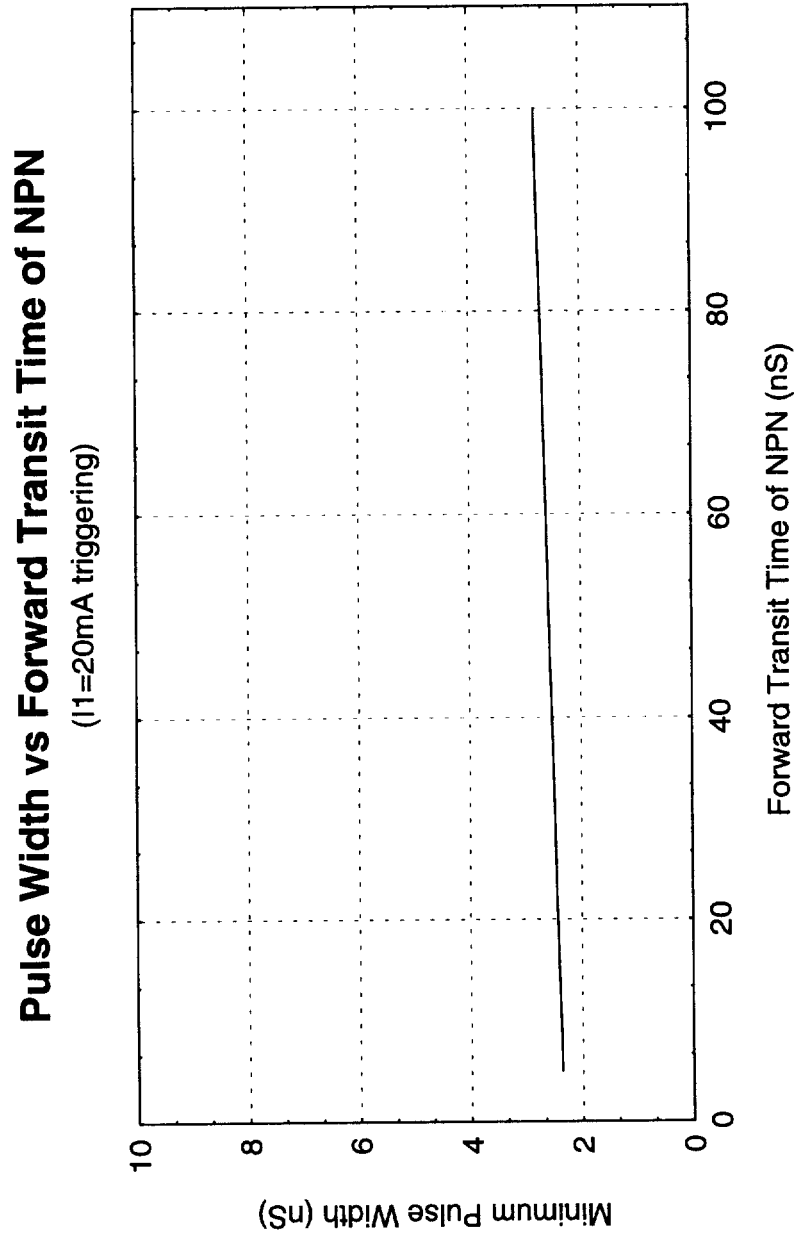


Figure 6.8 Effect of varying forward transit time of the NPN transistor on time to latchup

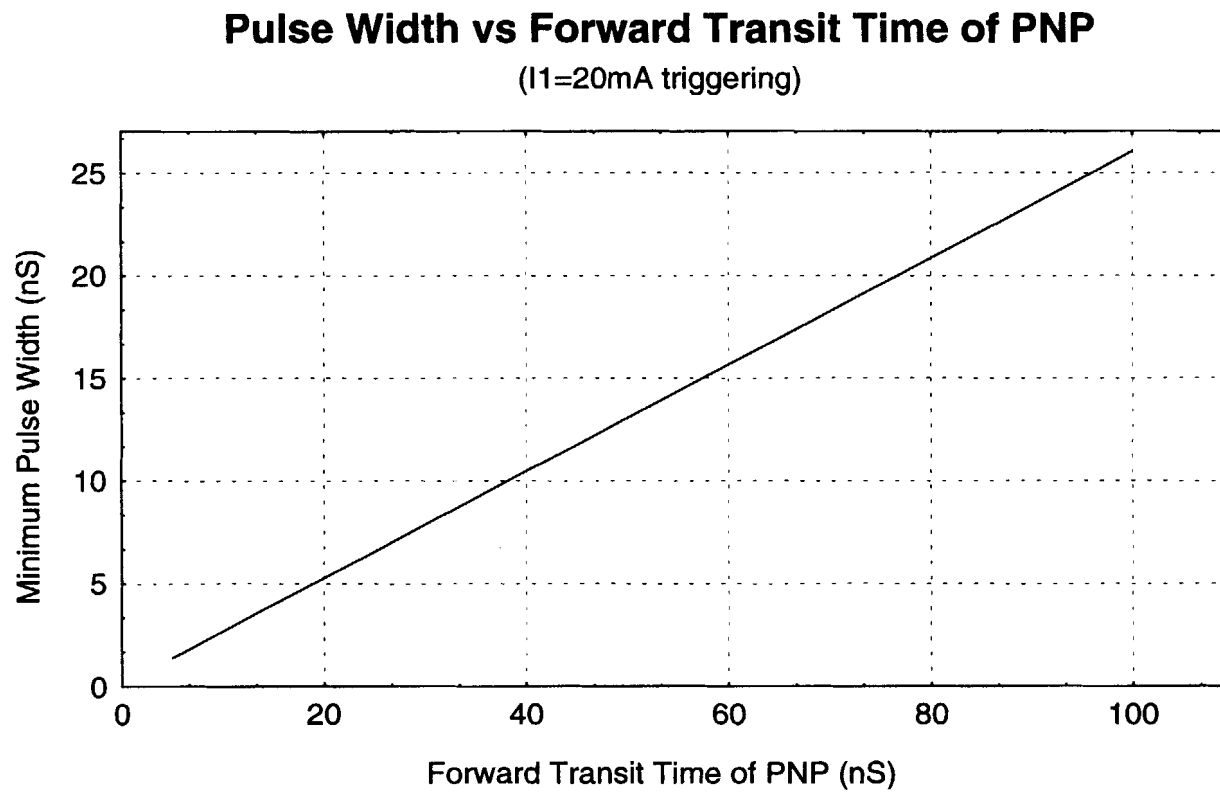


Figure 6.9 Effect of varying forward transit time of the PNP transistor on time to latchup

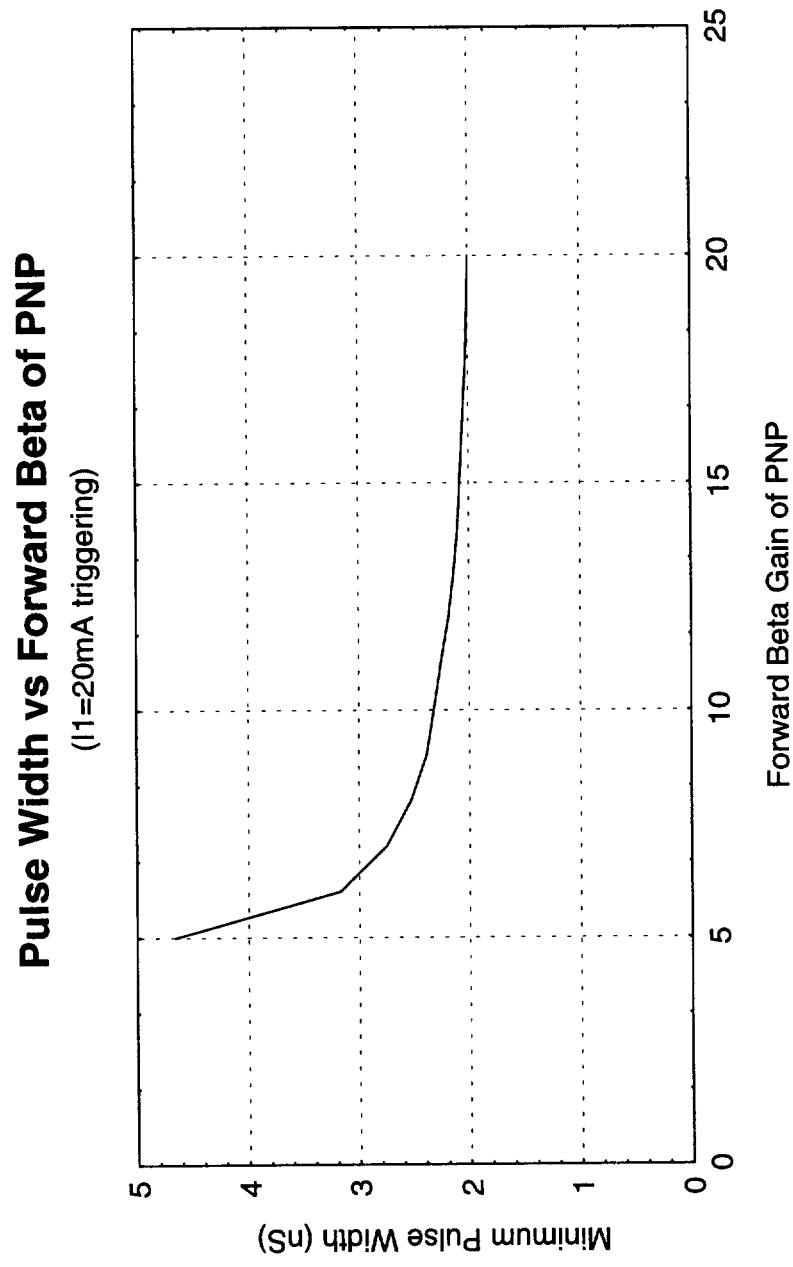


Figure 6.10 Effect of varying forward beta gain of the PNP transistor on time to latchup

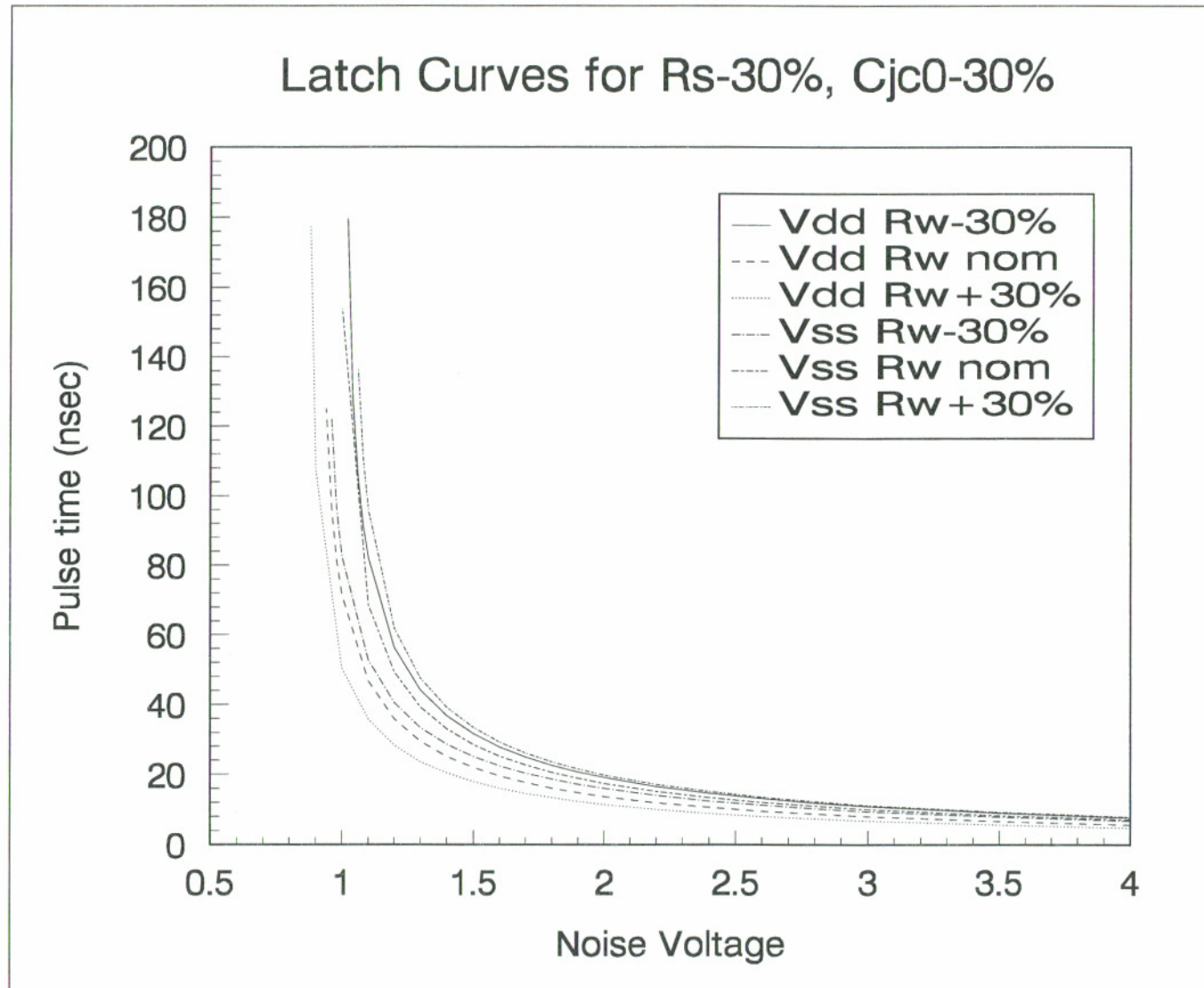


Figure 6.11 Minimum time to latch for a given pulse magnitude using Rs-30% and Cjc0-30%

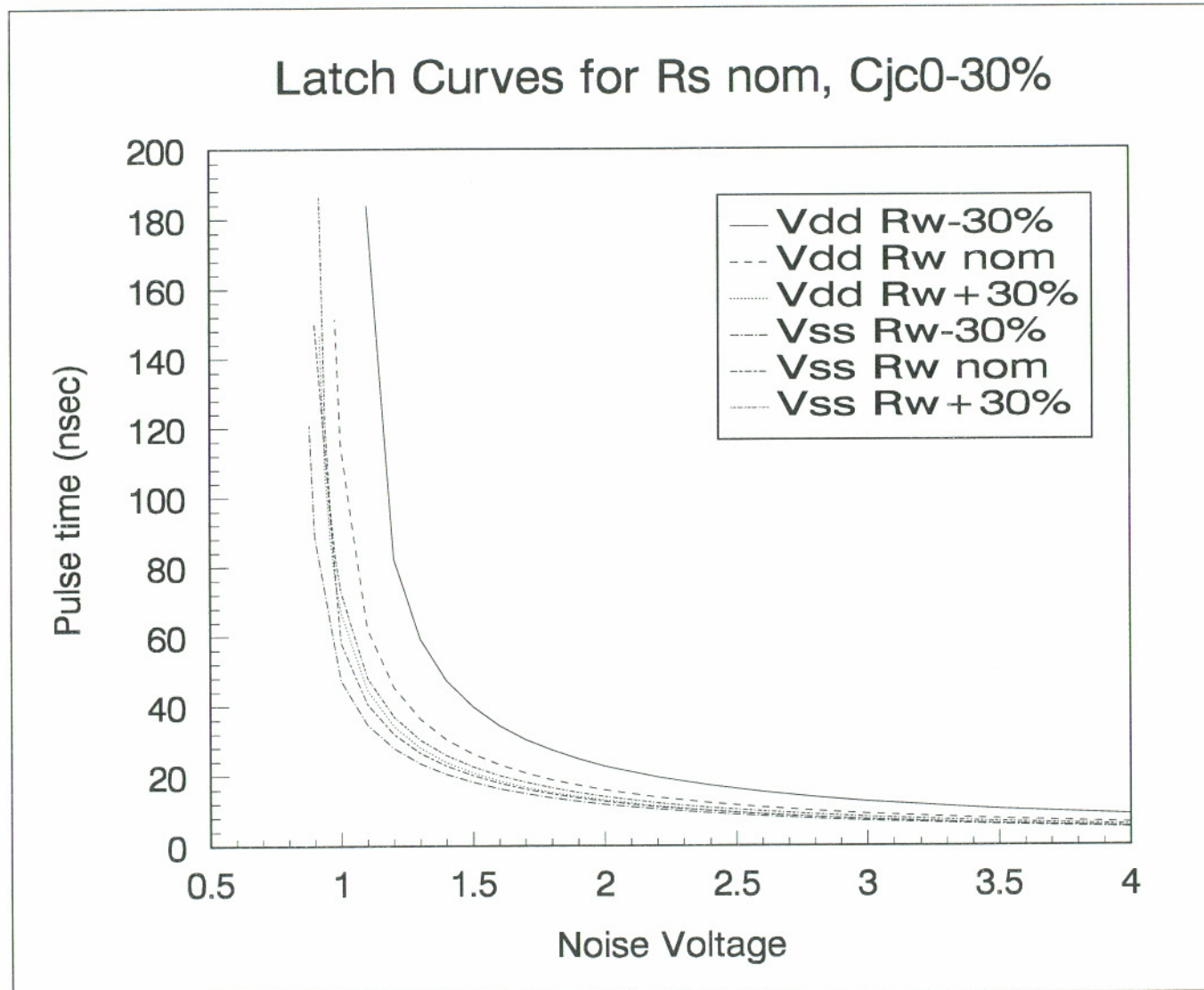


Figure 6.12 Minimum time to latch for a given pulse magnitude using R_s nominal and $C_{jc0-30\%$

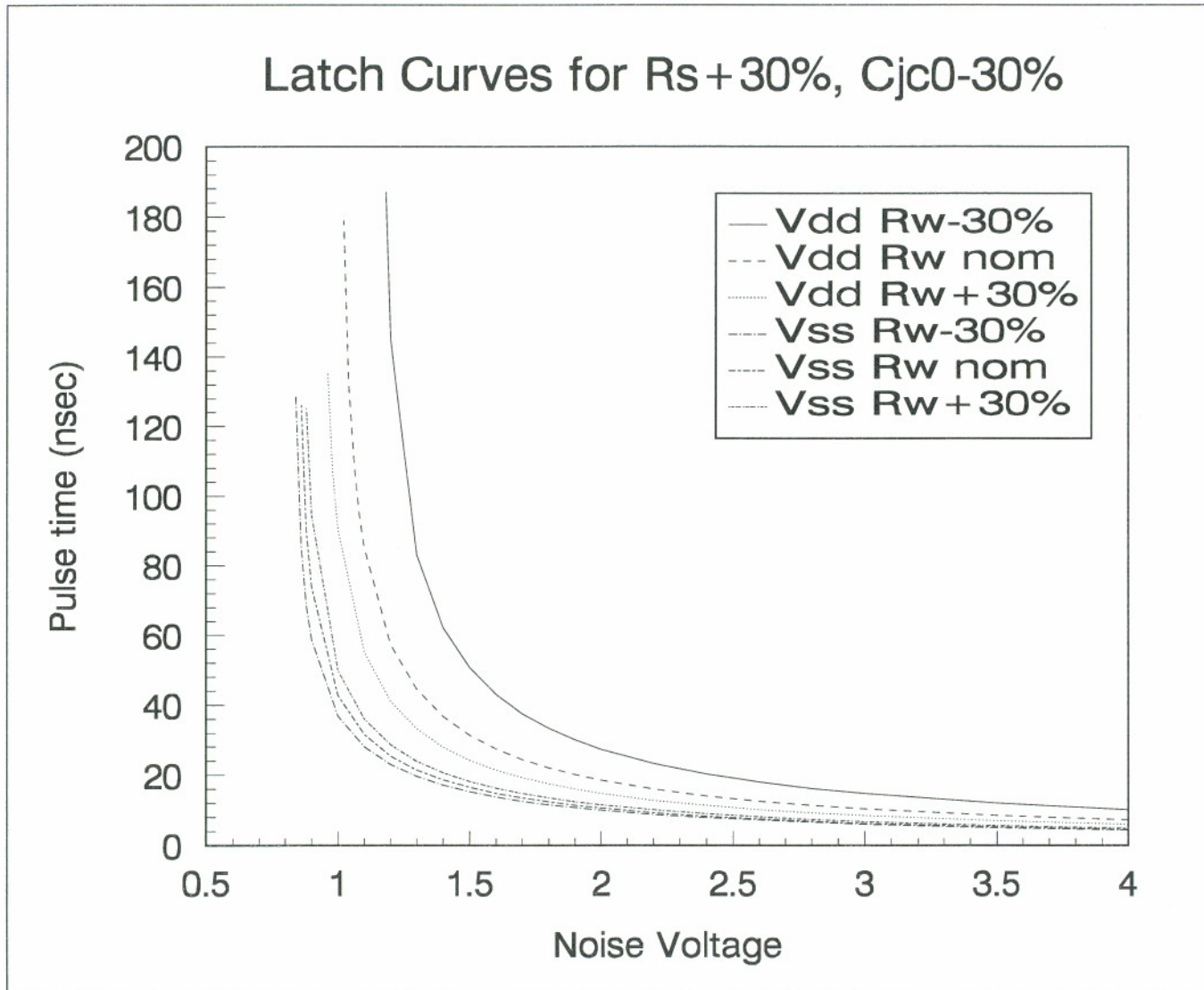


Figure 6.13 Minimum time to latch for a given pulse magnitude using $R_s + 30\%$ and $C_{jc0} - 30\%$

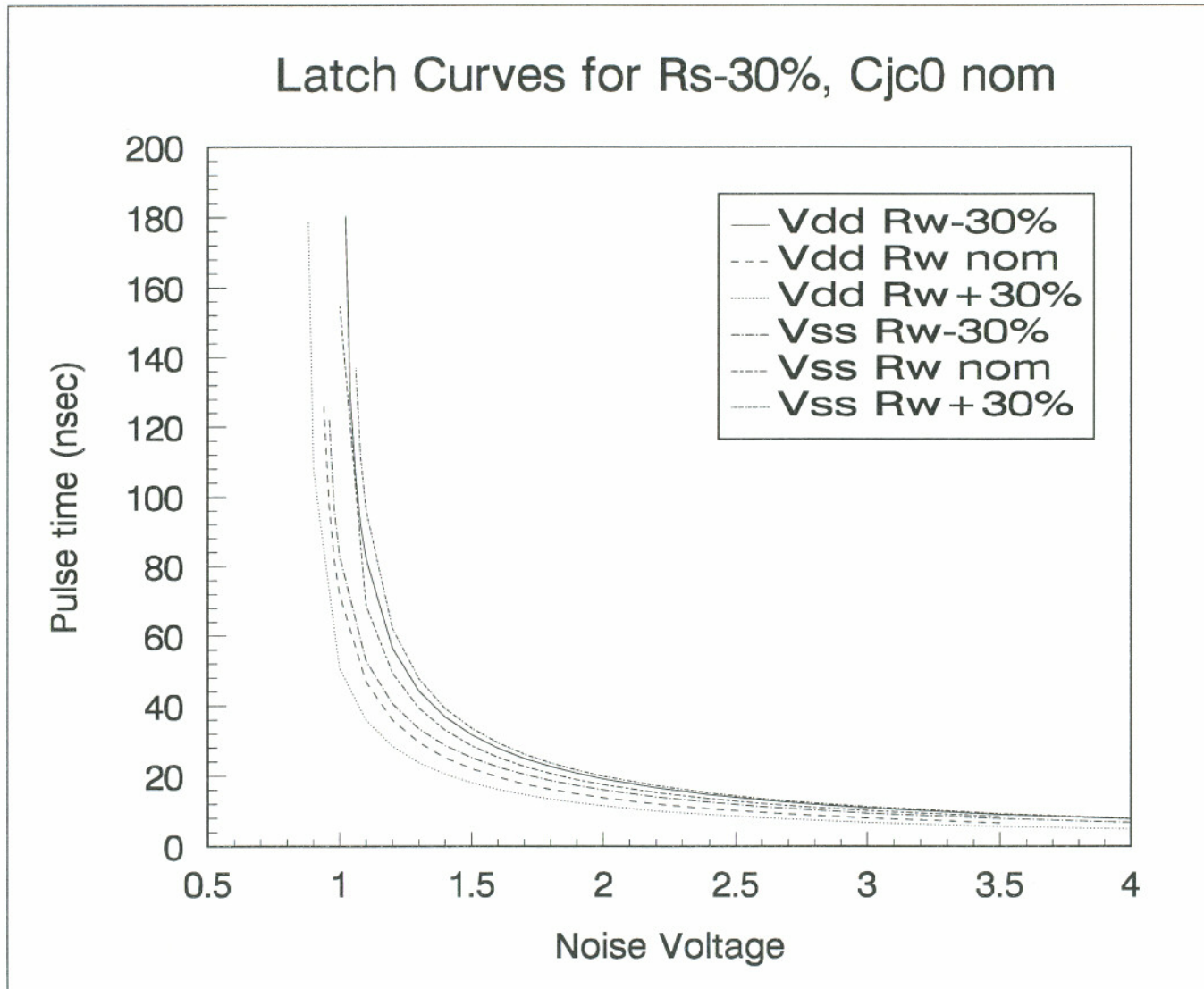


Figure 6.14 Minimum time to latch for a given pulse magnitude using Rs-30% and Cjc0 nominal

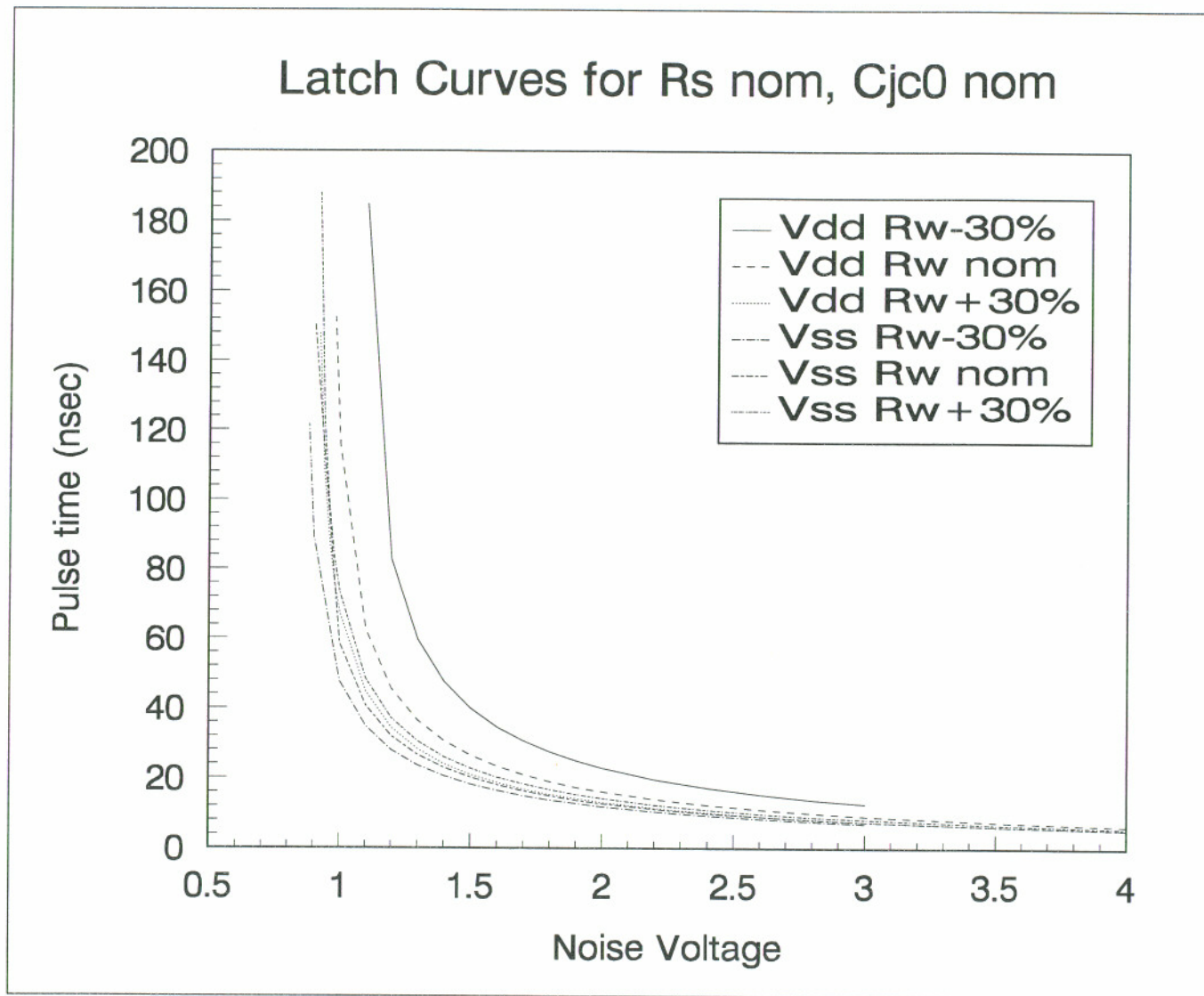


Figure 6.15 Minimum time to latch for a given pulse magnitude using R_s nominal and C_{jc0} nominal

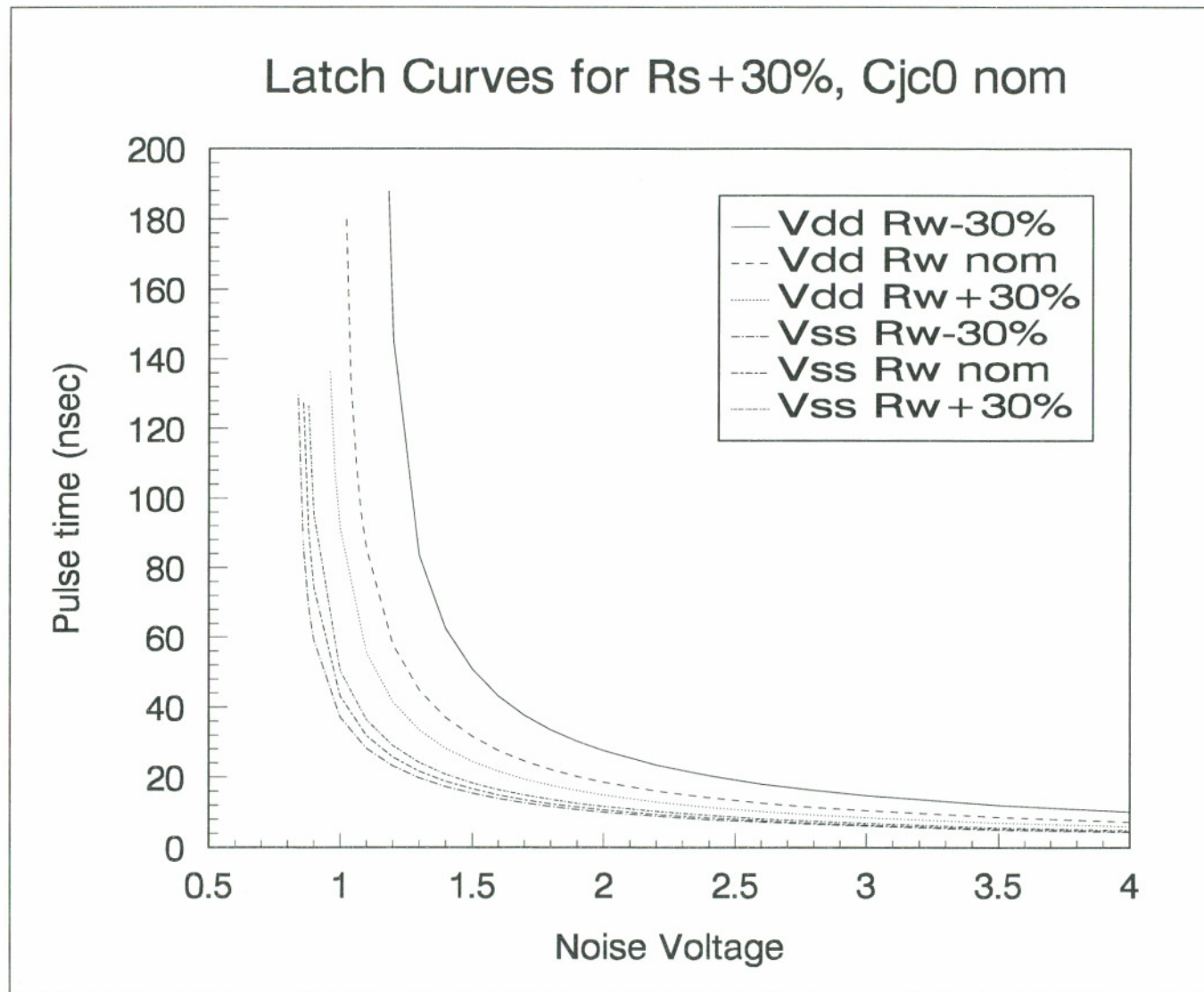


Figure 6.16 Minimum time to latch for a given pulse magnitude using $R_s + 30\%$ and C_{jc0} nominal

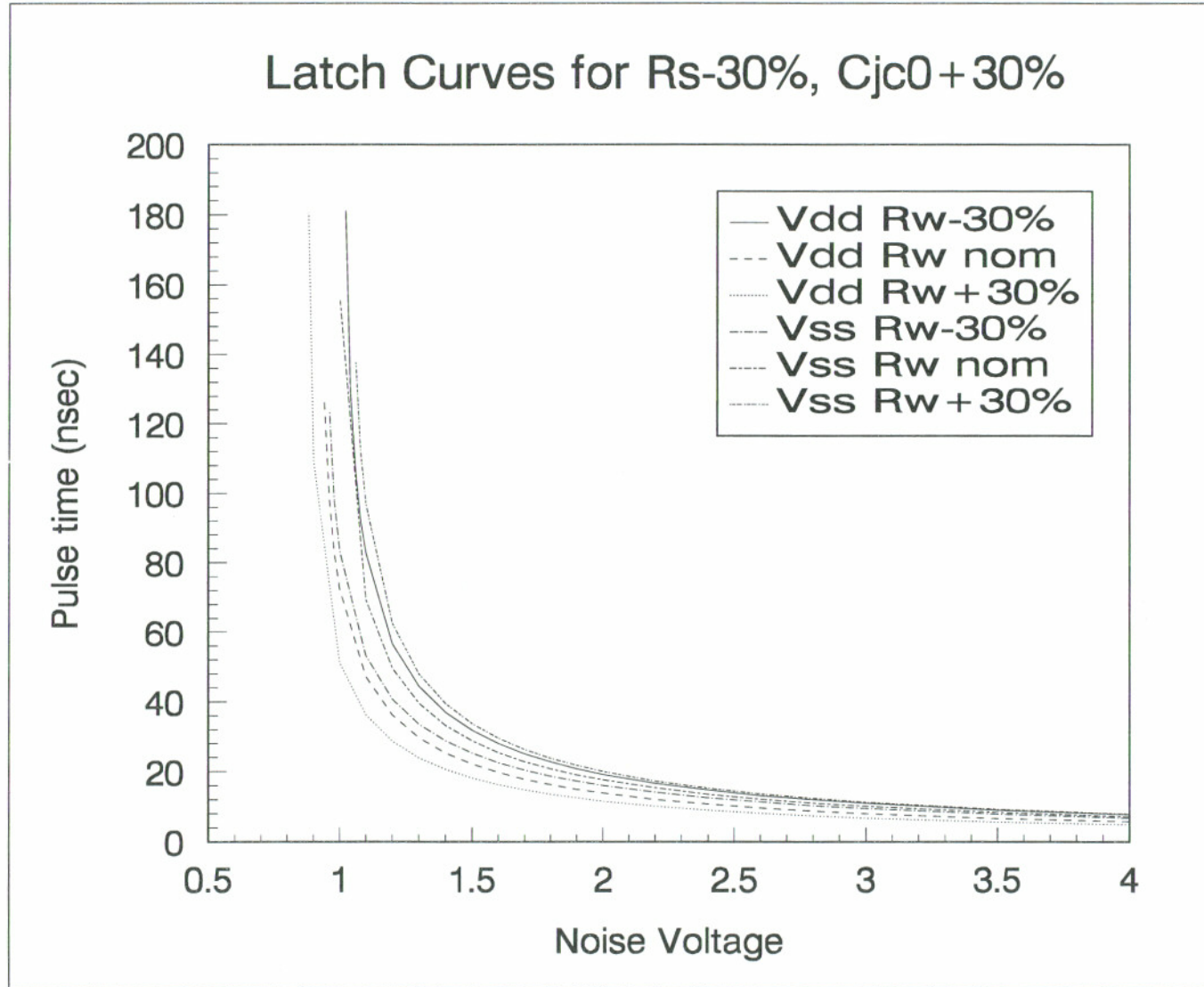


Figure 6.17 Minimum time to latch for a given pulse magnitude using Rs-30% and Cjc0+30%

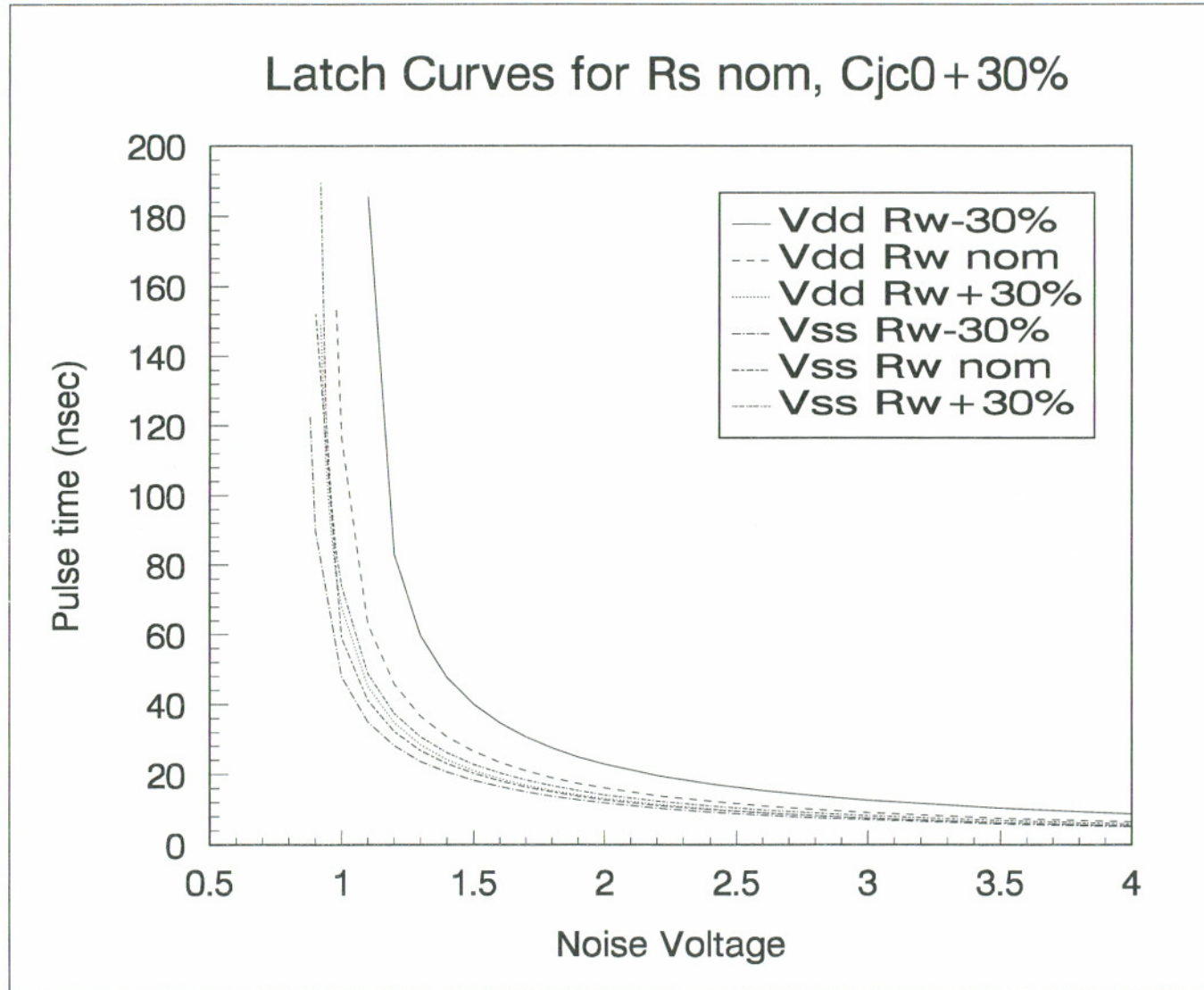


Figure 6.18 Minimum time to latch for a given pulse magnitude using R_s nominal and $C_{jc0} + 30\%$

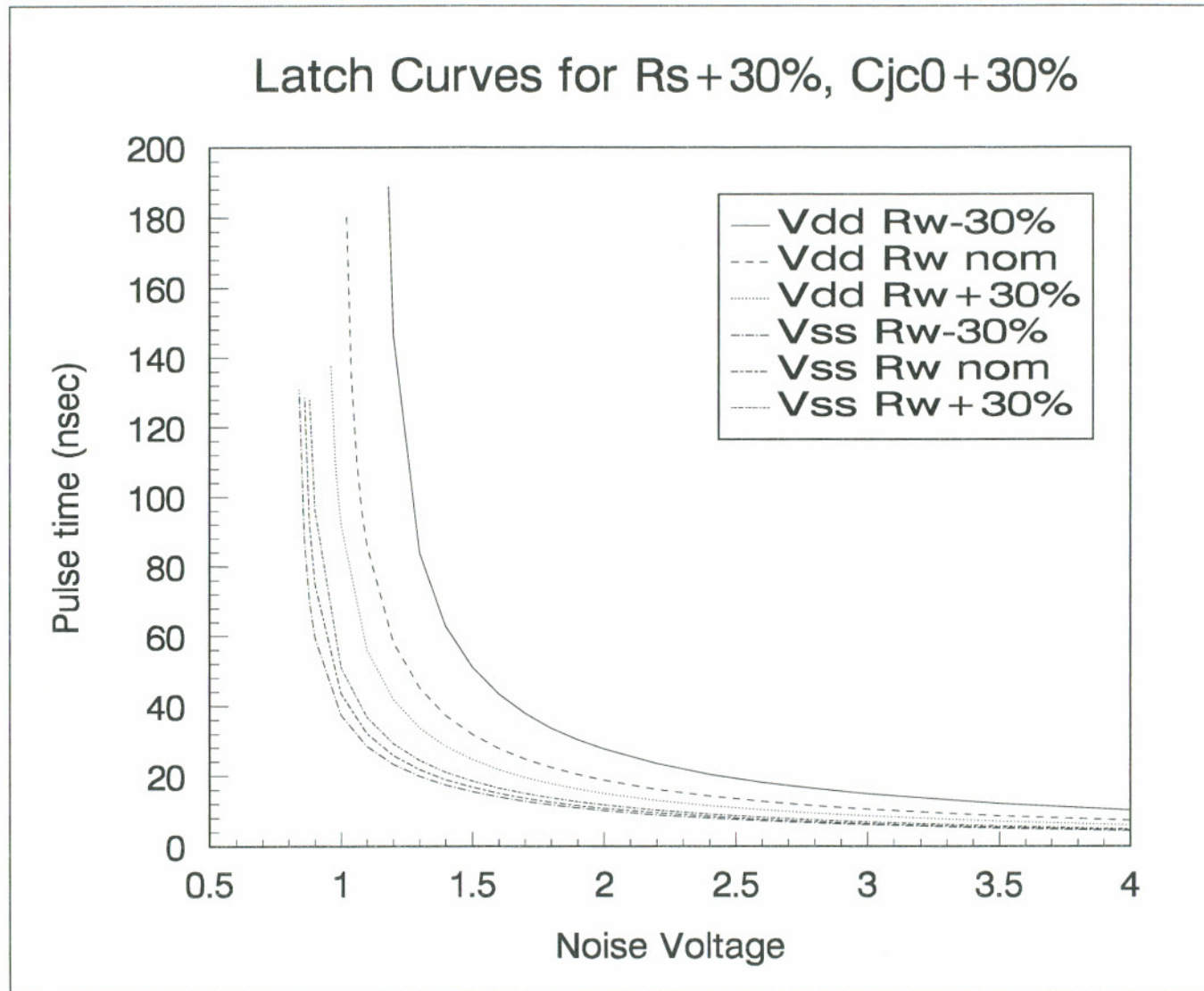


Figure 6.19 Minimum time to latch for a given pulse magnitude using $R_s + 30\%$ and $C_{jc0} + 30\%$

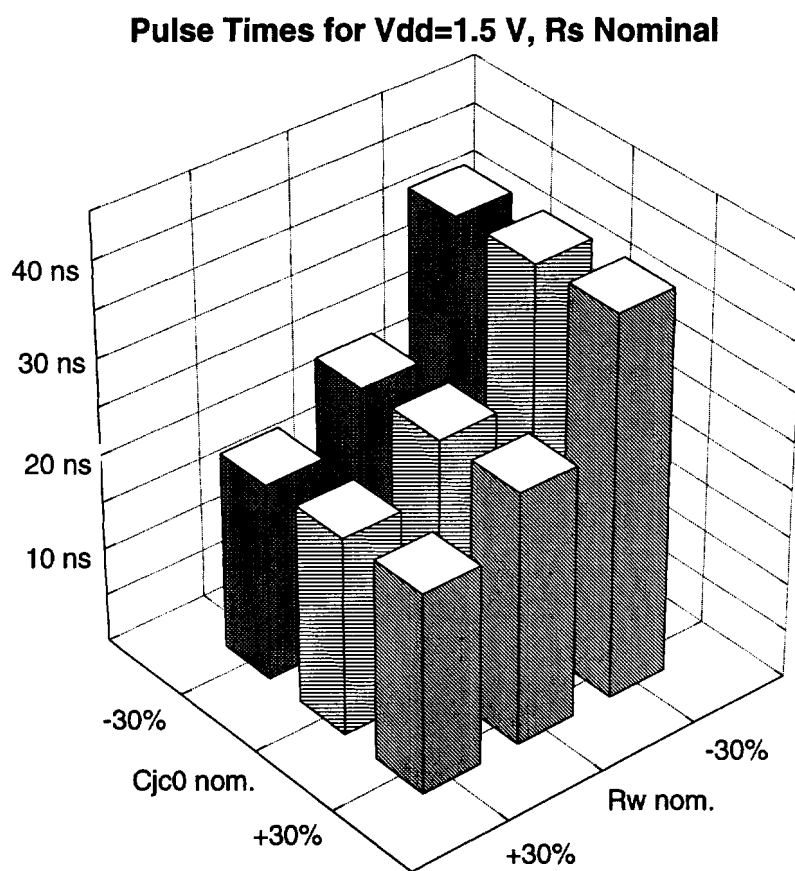


Figure 6.20 Comparison of effects R_w and C_{jc0} have on minimum time to latch for a 1.5V pulse

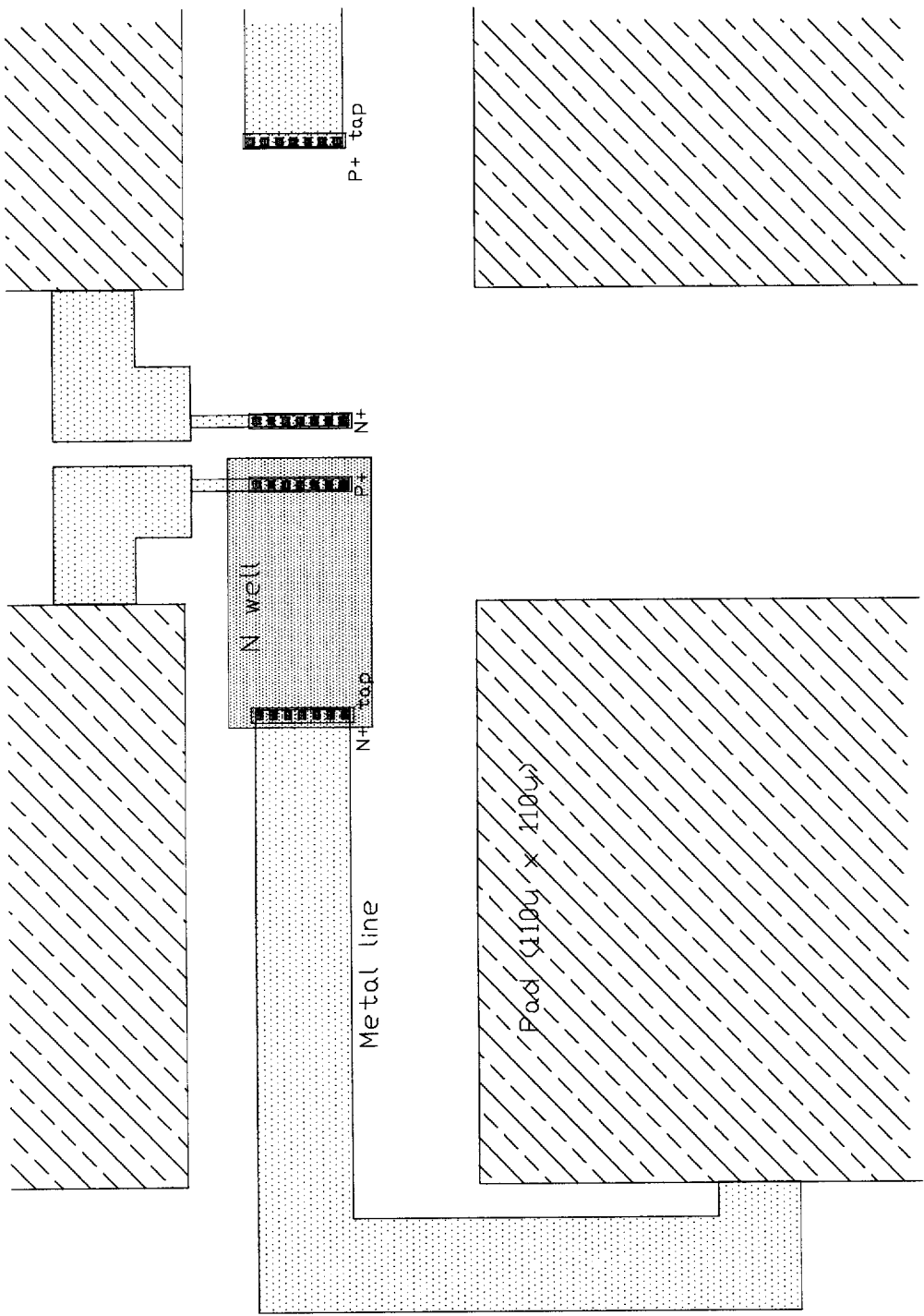


Figure 6.21 Physical layout of the "B" test structure

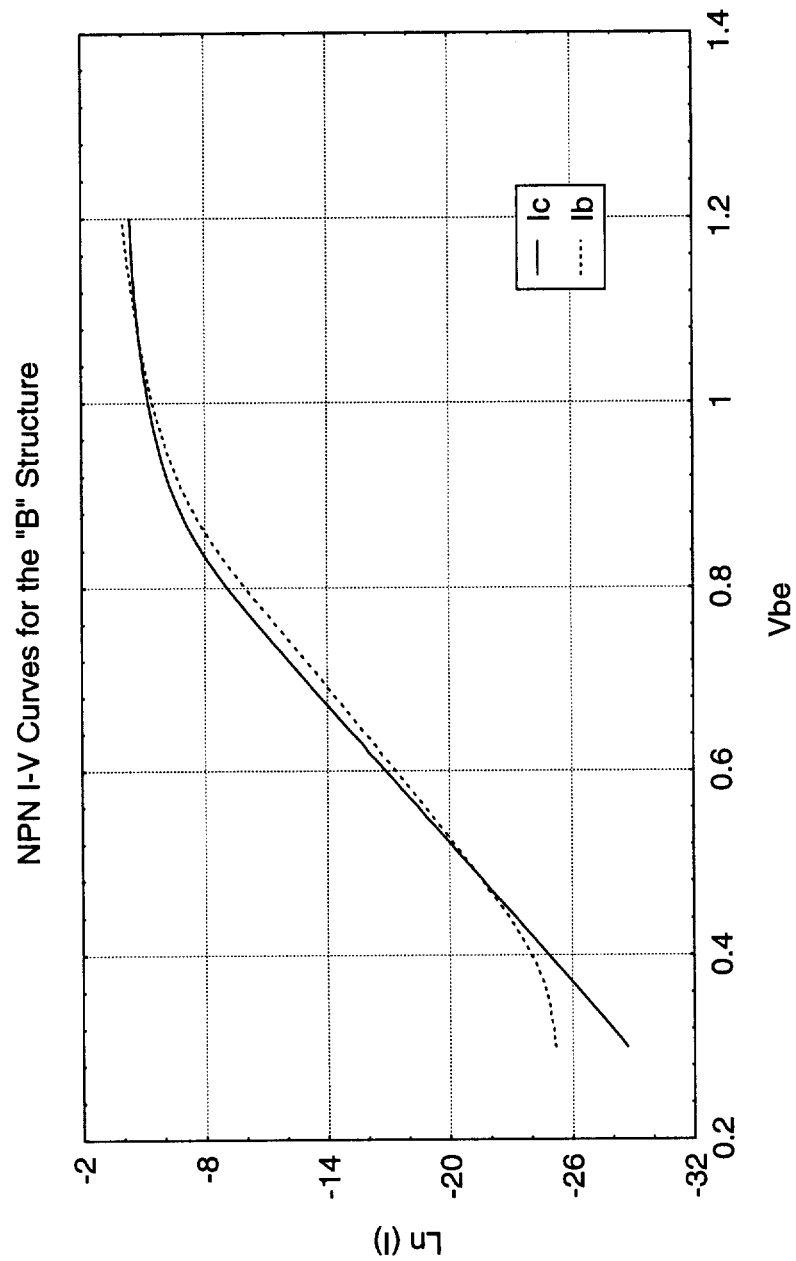


Figure 6.22 NPN I-V Curves for the "B" structure.

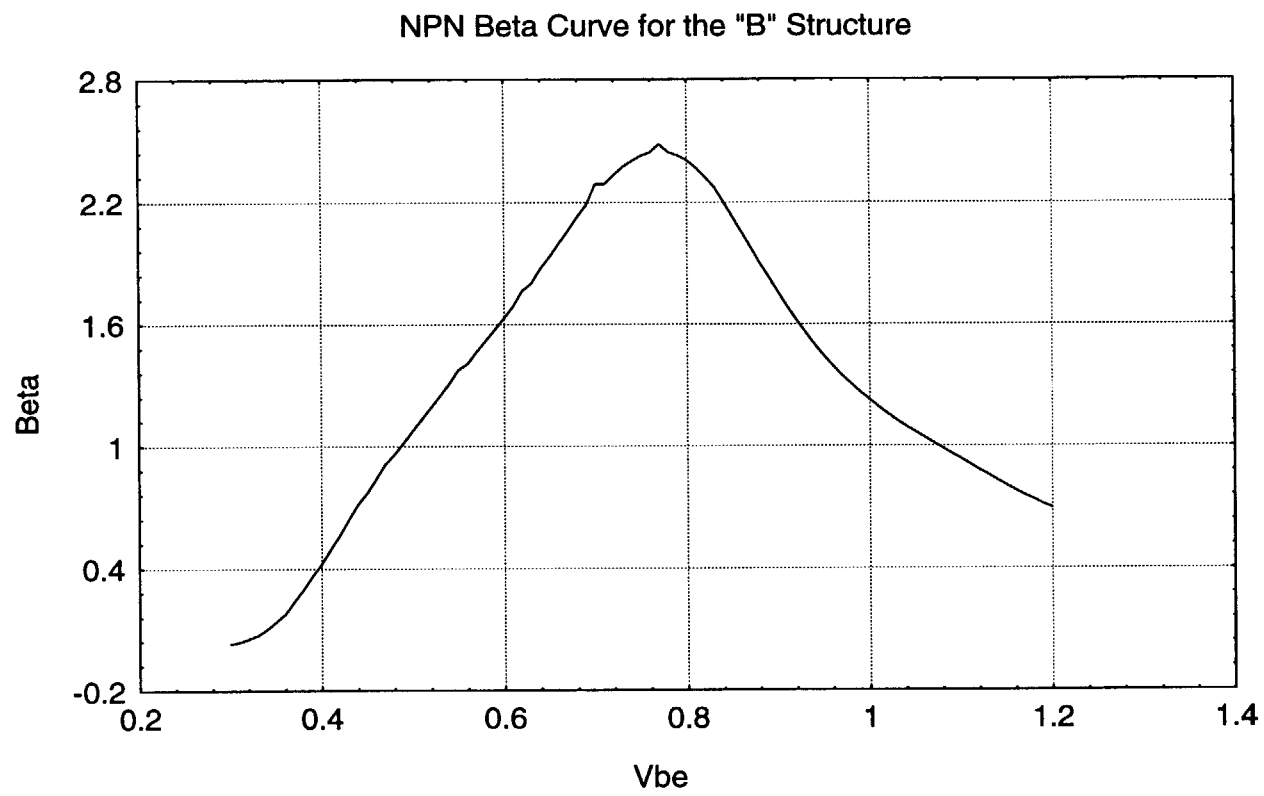


Figure 6.23 NPN Forward beta gain curve for the “B” structure.

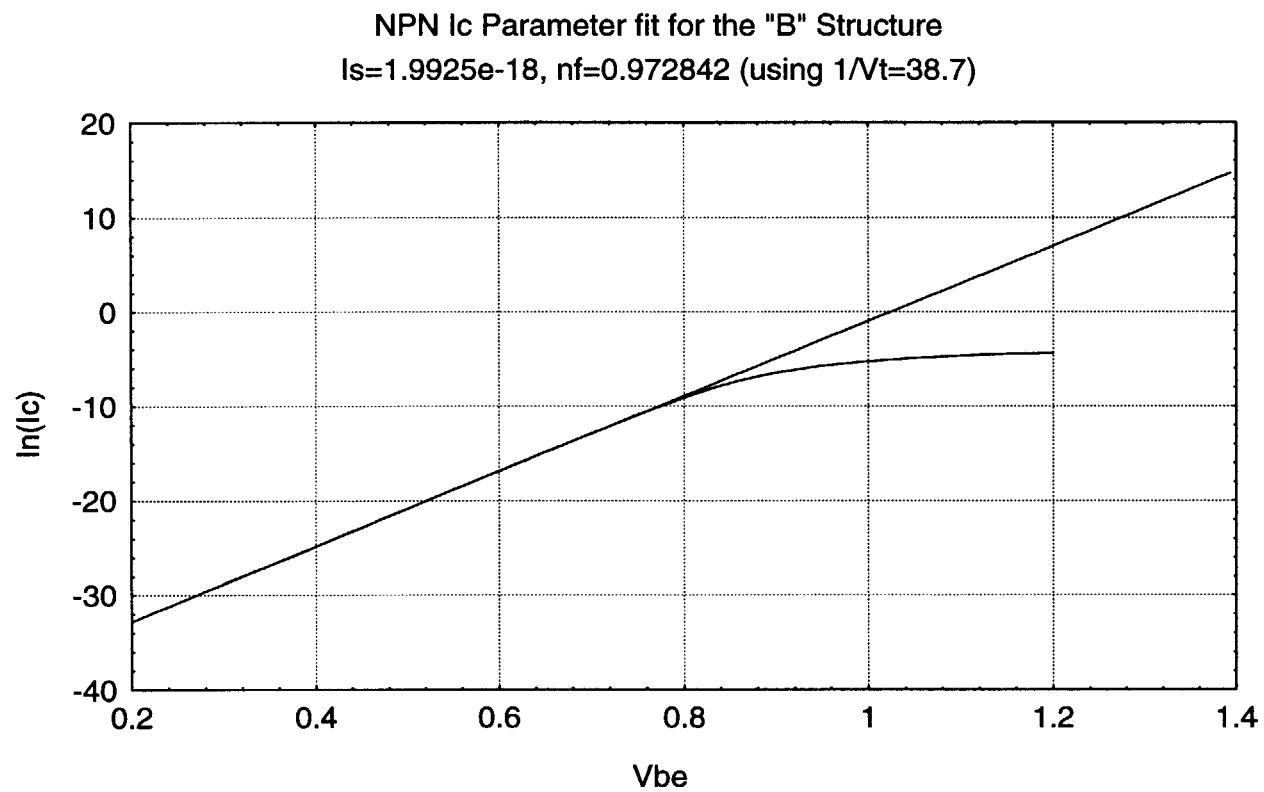


Figure 6.24 NPN I_c curve only and parameter fit for the “B” structure on a log scale.

Ic and Ib Curves NPN "B" Structure

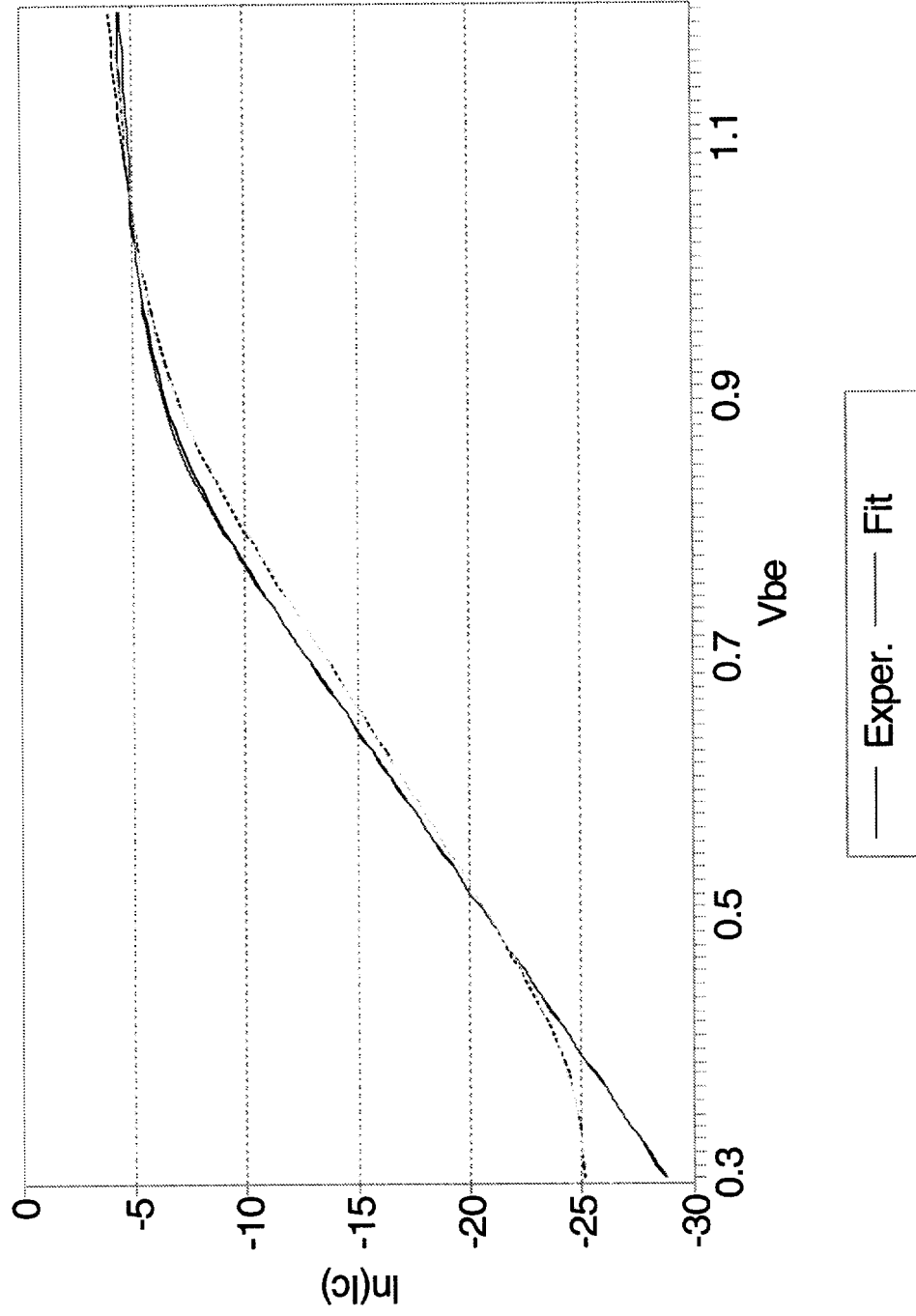


Figure 6.25 NPN I_c and I_b curves with parameter fit curves for the "B" structure.

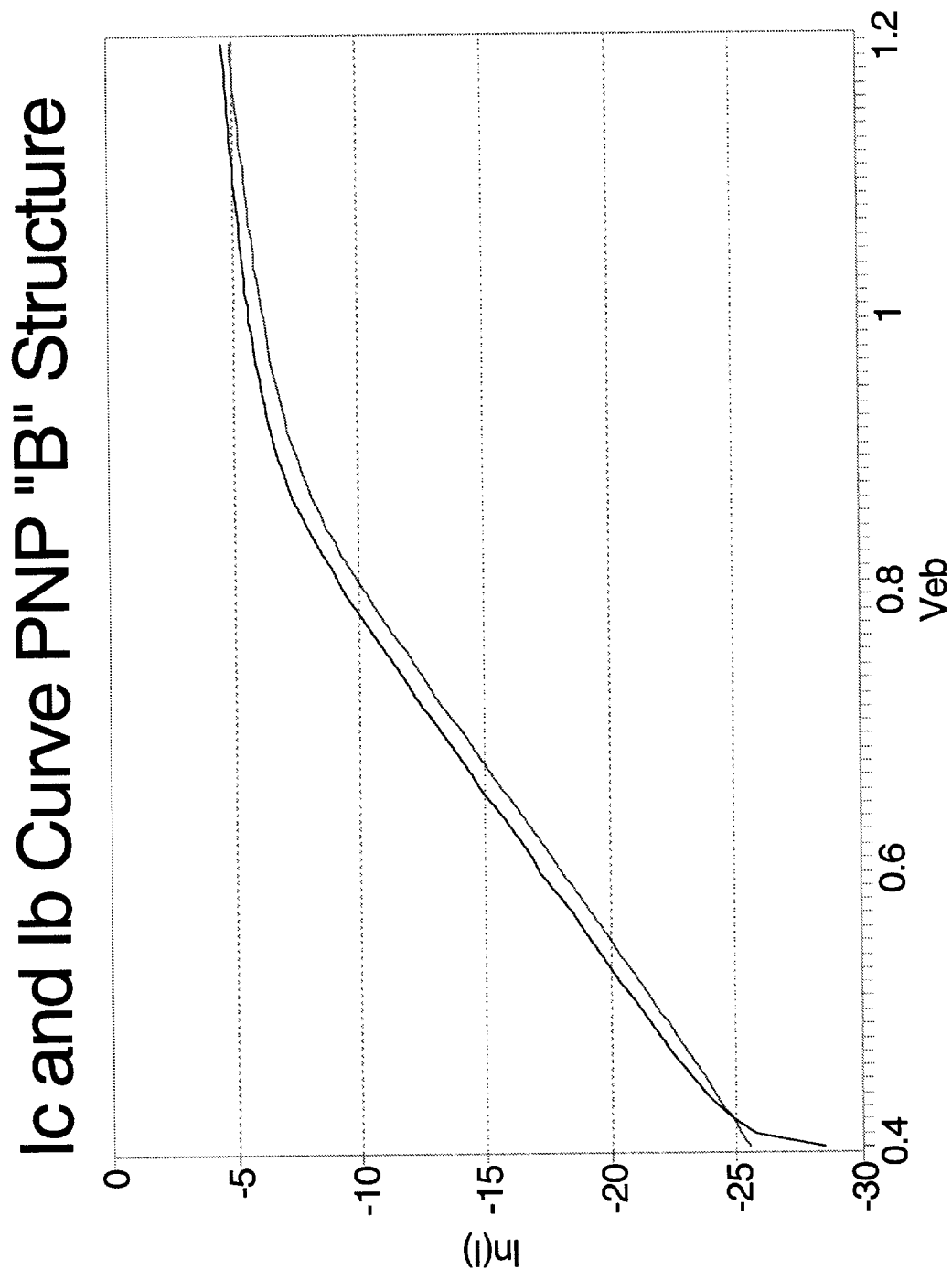


Figure 6.26 PNP I-V Curves for the "B" structure.

Beta Curves NPN "B" Structure

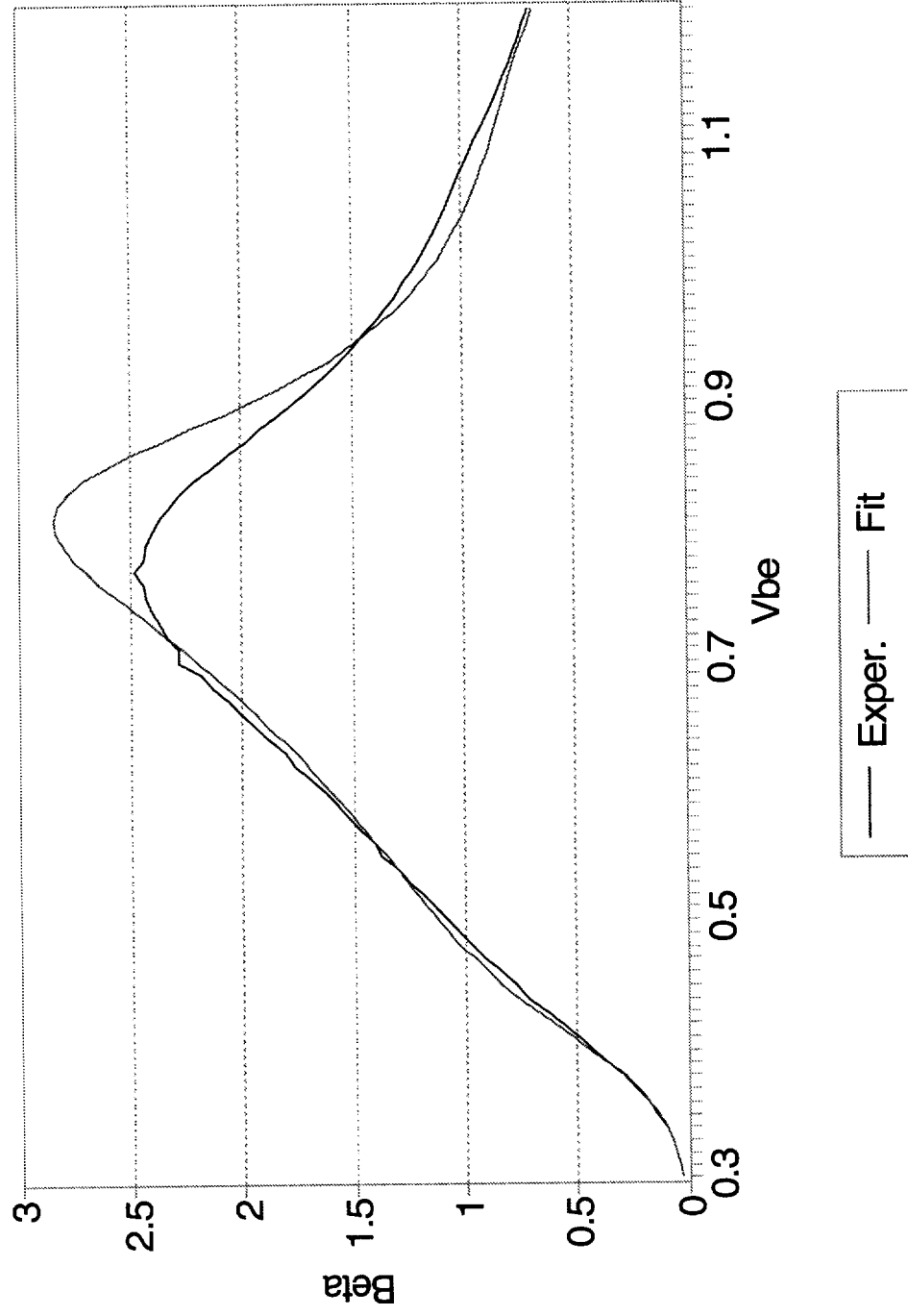


Figure 6.27 NPN forward beta gain curve and parameter fit curve for the "B" structure.

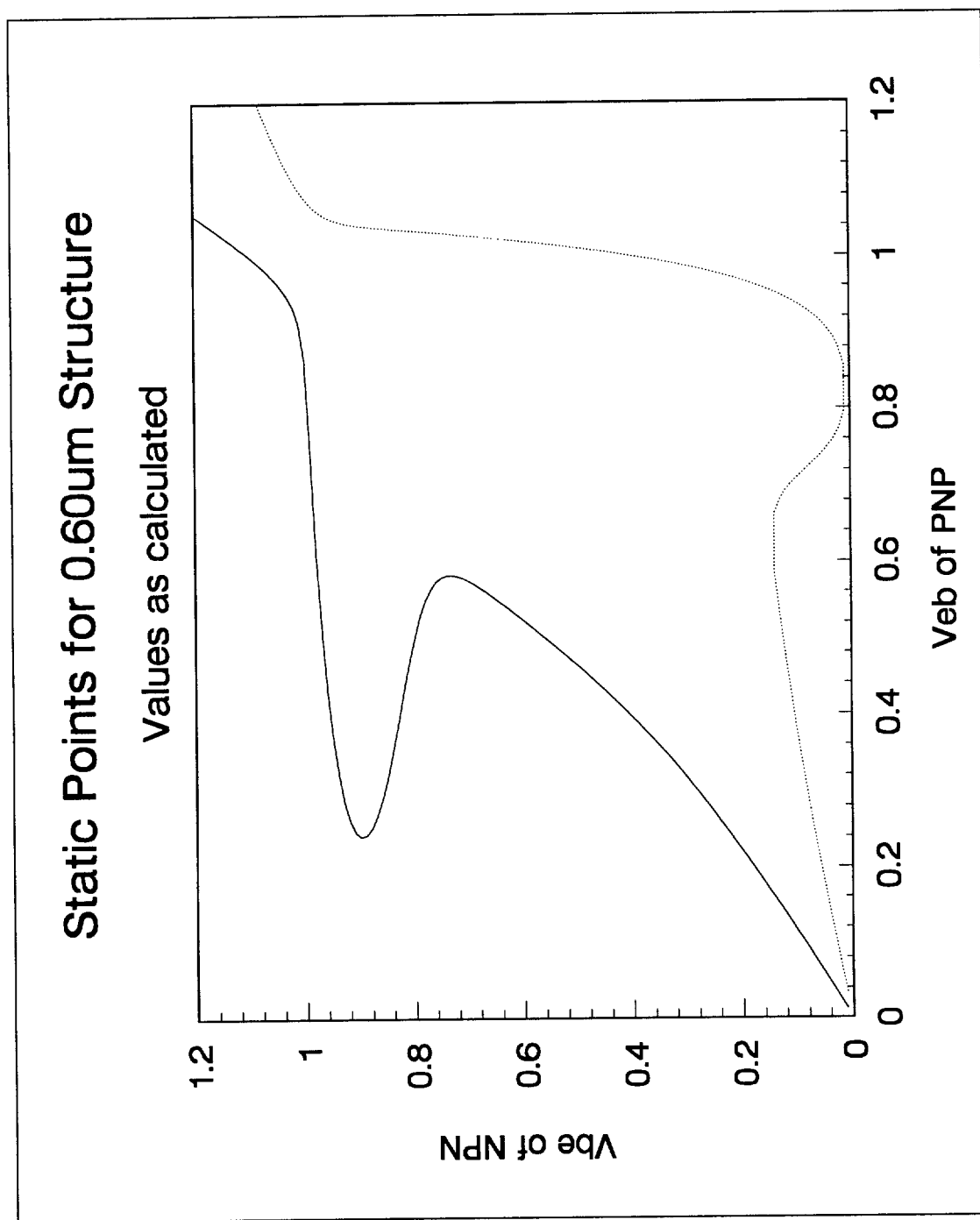


Figure 6.28 Curves of zero dv/dt for the 0.60um structure; which is clearly latchup immune.

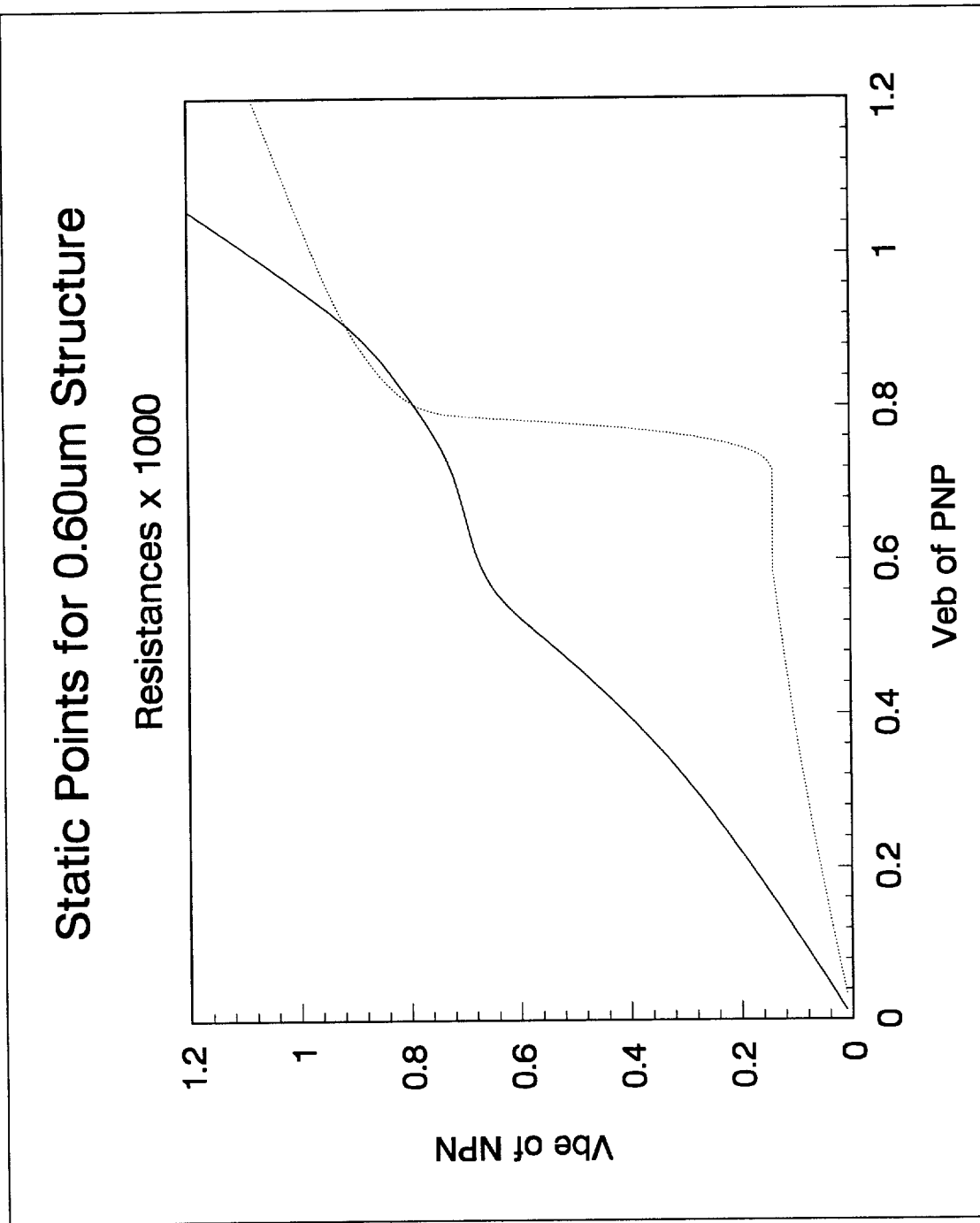


Figure 6.29 Curves of zero dv/dt for the 0.60um structure with base currents divided by 10.

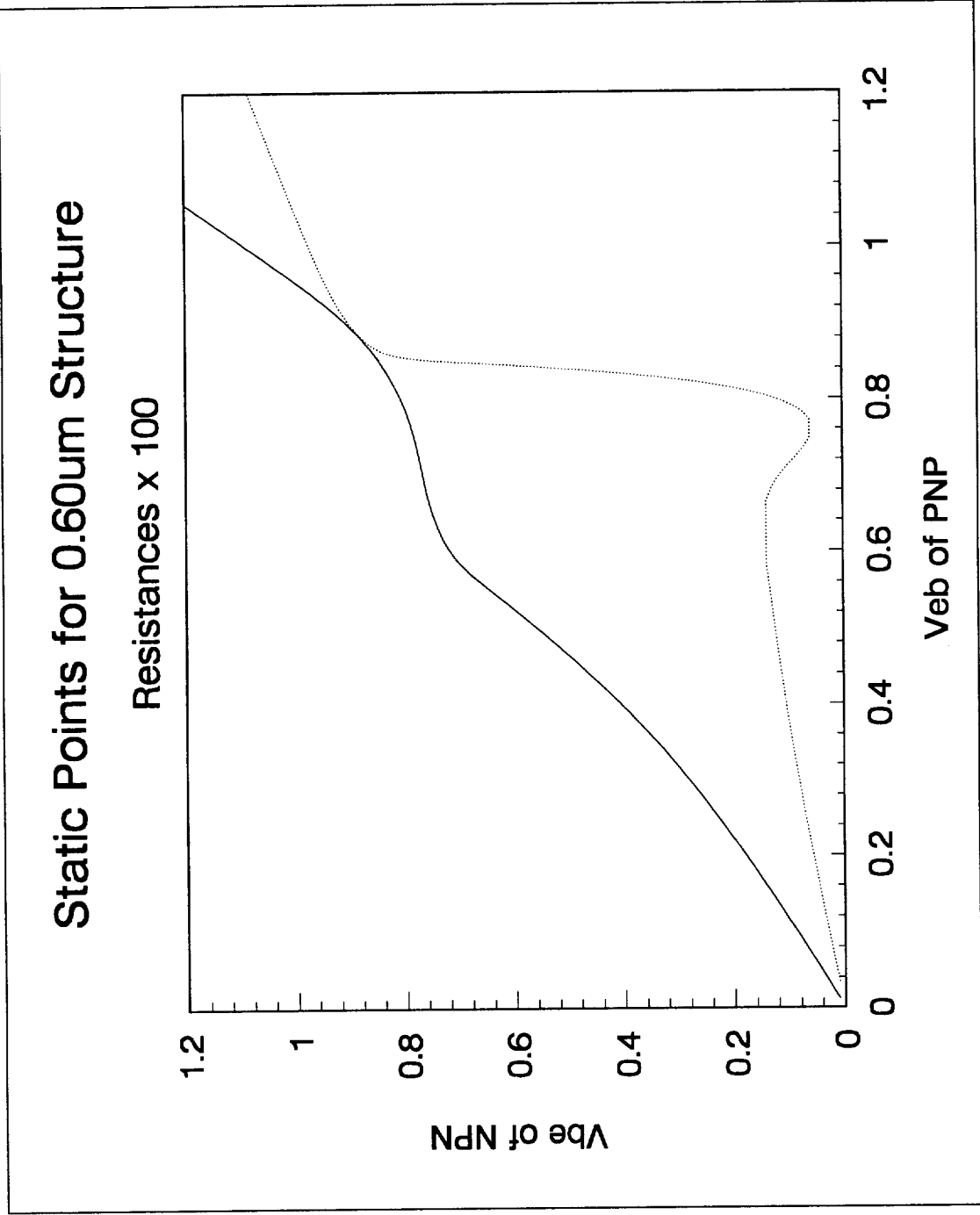


Figure 6.30 Curves of zero dv/dt for the structure with collector currents multiplied by 10.

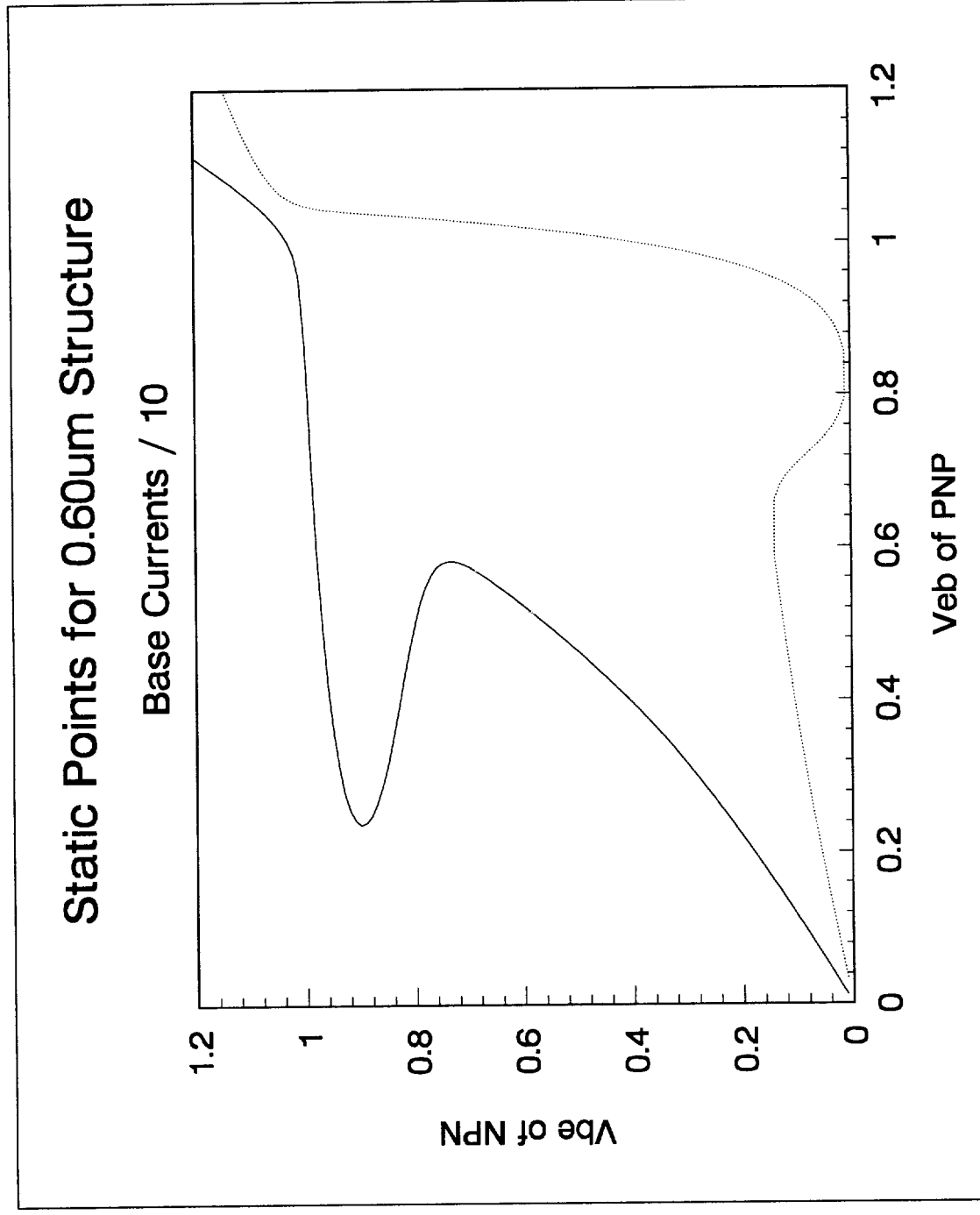


Figure 6.31 Curves of zero dv/dt with collector currents multiplied by 50. It has a stable latchup point.

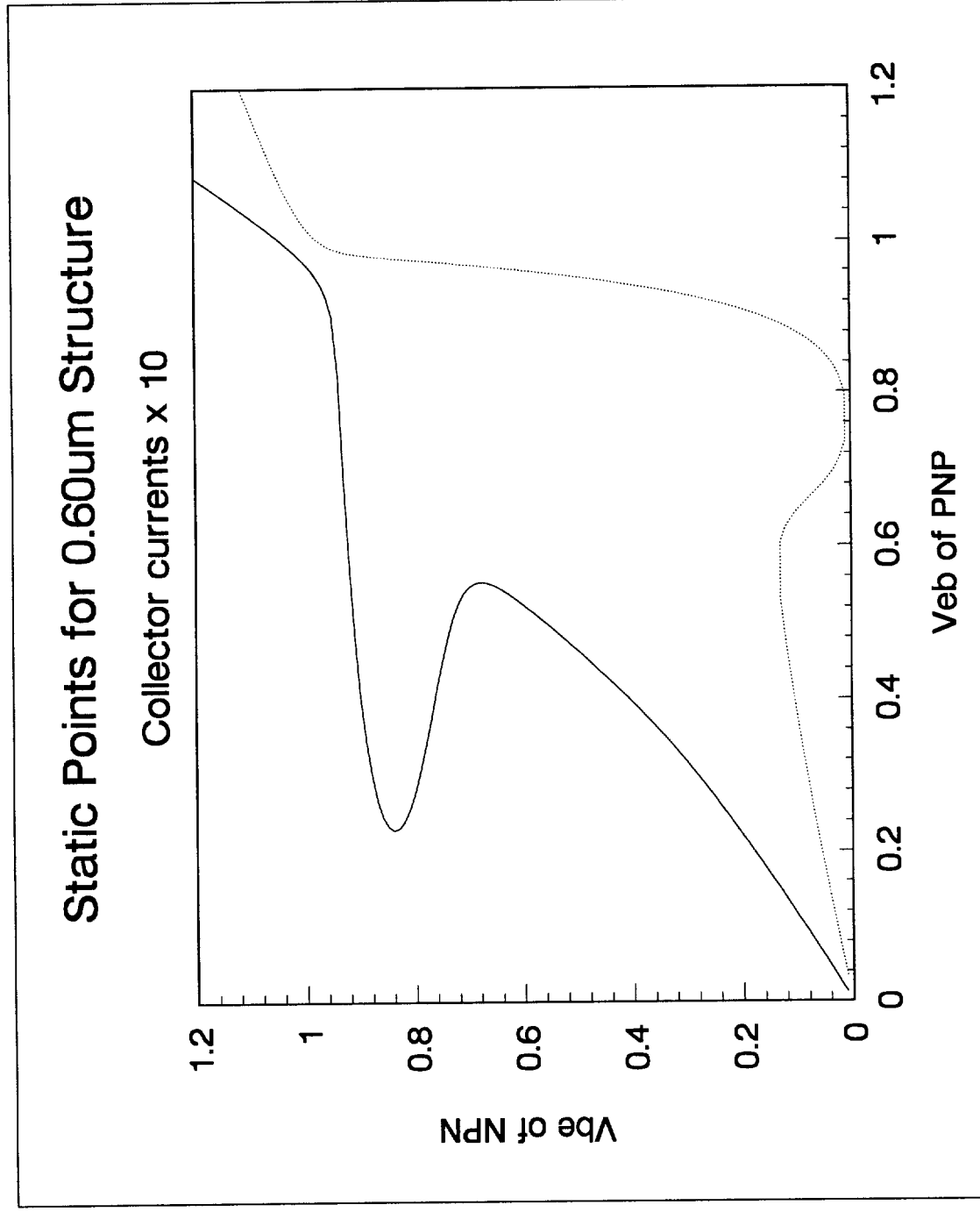


Figure 6.32 Curves of zero dv/dt for the structure with shunting resistors multiplied by 100.

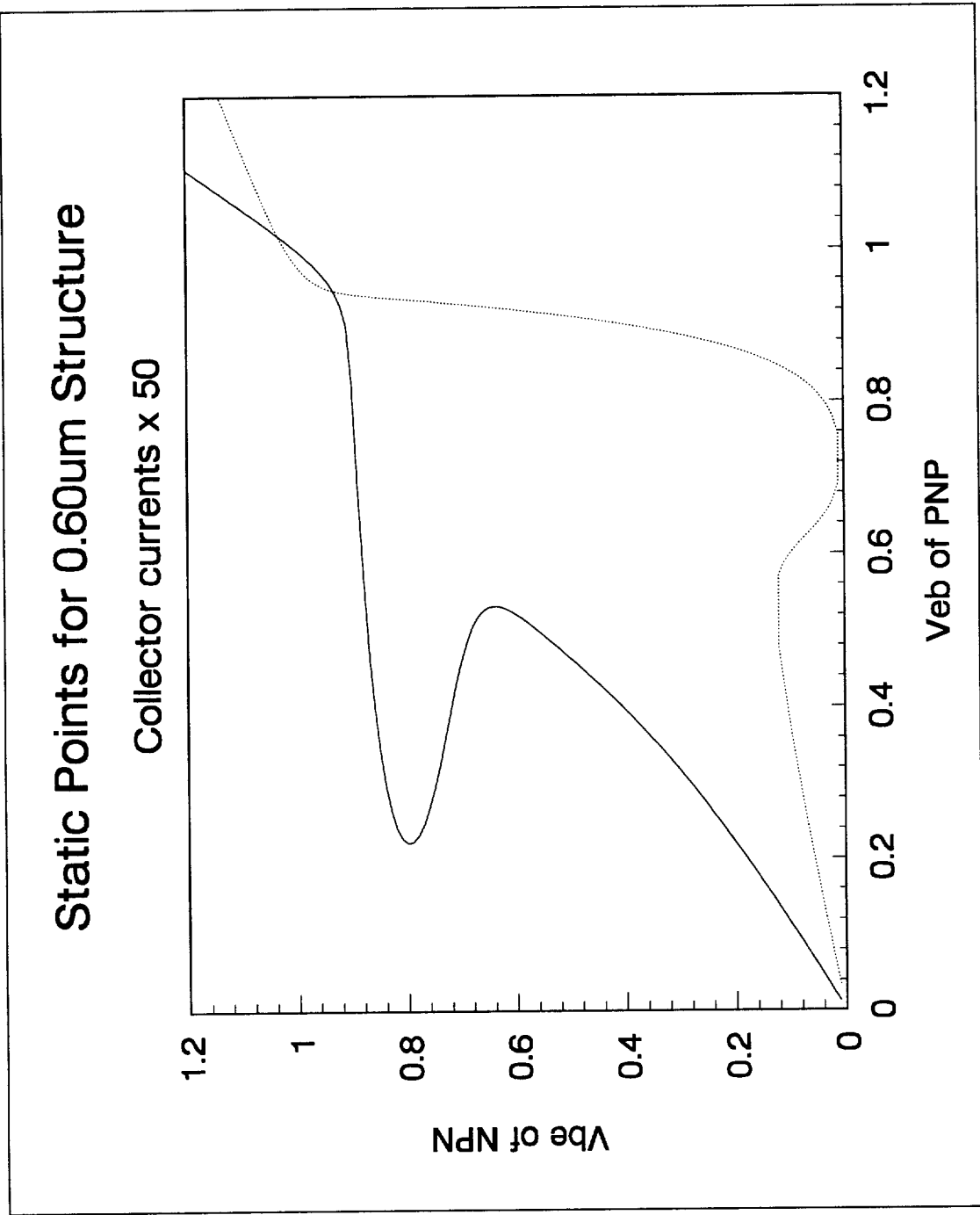


Figure 6.33 Curves of zero dv/dt for the structure with shunting resistors multiplied by 1000.

VII. Analytical Solution with Improved Accuracy

The shortcomings of linearizing the base and collector currents at somewhat arbitrary points has already been highlighted in the previous attempt at an analytical solution. The purpose of this chapter is to explain a new approach to an analytical solution for transient latchup pulses utilizing the voltage instability criterion. The benefit of having an accurate analytical solution is just that the speed to solution is faster than numerical techniques. Frequently, analytical solutions also provide additional insight into the phenomenon being studied. In this case, however, the solutions are sufficiently complex that interpretation of the results on physical grounds was not possible.

The analytical characterization is divided into three parts. First, the emitter-base voltage of the two transistors at the unstable point are determined. Second, the minimum pulse voltage which will lead to latchup is found. Finally, the curve of minimum pulse time versus pulse voltage is approximated. The agreement between the analytical results and simulated results will be shown to be sufficiently close to make the analytic approach useful.

The first step of the latchup characterization is to find the unstable point. The general static equations are linearized about a point in the vicinity of where the unstable point is expected. This point is chosen as the point where the emitter-base voltage is such that the base current is equal to the current through the shunting resistor. Simpler metrics can be used; the only requirement for better accuracy is that smaller resistances push the unstable point guess to higher voltages, and larger resistances move the unstable point to smaller voltages. Once the initial guess is determined, the transistor base current and

current gain (beta) are linearized at that point. Attempts to linearize the base and collector currents yielded unacceptable results. By linearizing beta instead of collector current, the relation between base and collector current is more accurately modeled. However, the penalty for this extra accuracy is that the collector current, which is the product of beta and base current, is now a quadratic expression. The new expressions are as follows:

$$I_{b1} = a_{i1} \cdot (V_{eb1} - V_{s1}) + I_{bs1} \quad (1)$$

$$\beta_{2} = a_{\beta 2} \cdot (V_{be2} - V_{s2}) + \beta_{s2} \quad (2)$$

$$I_{b2} = a_{i2} \cdot (V_{be2} - V_{s2}) + I_{bs2} \quad (3)$$

$$\beta_{1} = a_{\beta 1} \cdot (V_{eb1} - V_{s1}) + \beta_{s1} \quad (4)$$

where the “Vs” terms are the voltages at which the functions are linearized, the “a” terms are slopes at those points, and the “s” terms are the values at Vs. Substituting the linearized expressions into the static equations now leads to a quartic expression. This is the maximum order for which roots can be derived analytically and the expressions are rather long.

Original static system equations:

$$I_{C2} - I_{B1} - \frac{V_{EB1}}{R_1} = 0 \quad (5)$$

$$I_{C1} - I_{B2} - \frac{V_{BE2}}{R_2} = 0 \quad (6)$$

The resulting quartic equation when substituting the linear approximations:

$$\begin{aligned}
 & ai2(V2 - Vs2) + Ibs2 + \frac{V2}{R2} - \\
 & \left(ab1 \left(ai1 Vs1 - Ibs1 + ab2 V2^2 ai2 - 2 ab2 V2 ai2 Vs2 + ab2 V2 Ibs2 + ab2 Vs2^2 ai2 \right) - ab2 Vs2 Ibs2 + Bs2 ai2 V2 - Bs2 ai2 Vs2 + Bs2 Ibs2 \left(ai1 + \frac{1}{R1} \right) - Vs1 \right) + Bs1 \\
 & \cdot \left(ai1 \left(ai1 Vs1 - Ibs1 + ab2 V2^2 ai2 - 2 ab2 V2 ai2 Vs2 + ab2 V2 Ibs2 + ab2 Vs2^2 ai2 \right) - ab2 Vs2 Ibs2 + Bs2 ai2 V2 - Bs2 ai2 Vs2 + Bs2 Ibs2 \left(ai1 + \frac{1}{R1} \right) - Vs1 \right) + Ibs1
 \end{aligned} \tag{7}$$

This results in an equation in $V2$ of the form:

$$x^4 + a3 x^3 + a2 x^2 + a1 x + a0 = 0 \tag{8}$$

The coefficients are each rather complicated expressions, and the solution to the general fourth-order equation is a complex expression of the coefficients. Unfortunately, lower orders give poor results, as mentioned. There are, of course, four roots to the equation. Of these, two are always complex and not of interest. The other two roots give the unstable static point and the stable latch point. If these roots are complex, the conclusion is that the structure is latchup immune. The degree to which it is immune can be judged from the root values. If the imaginary portions are relatively small, the structure is “almost latchable”. If the roots are real and nearly equal, the structure is “almost immune”.

Although the first step is rather complicated, it corresponds to simply defining the latchup criterion. The problem of propagating the system equations to find the voltages as a function of time remains. It has already been determined that the previous method was subject to wide variations based on the linearizing points chosen in linearizing the transistor currents. Since no higher order allow solutions to the laplace transform, a different approach must be taken.

The second step of the characterization is to find the minimum pulse voltage which will lead to latchup. Equivalently, it could be stated that the second step is to find the maximum pulse voltage which will not lead to latchup; the dividing point is the value determined. To find this voltage, it is assumed that the emitter-base voltage of the non-pulsed transistor is just below its unstable point value, which will be below turn-on. The

transistor currents of the non-pulsed transistor can be ignored, while a maximum current is flowing through its shunting resistor. With these simplifications, the static equations can be solved for the minimum pulse voltage leading to latchup. The reason the minimum voltage is found is that the static equations with stimulus apply. Since the stimulus is on mathematically forever, the static equations determine the voltages at the end of the pulse. The general equations with time derivatives are unnecessary. So, the equations to be solved are now:

$$-I_{B1} - \frac{V_1}{R_1} + \frac{V_{stim,min}}{R_1} = 0 \quad (9)$$

$$I_{C1} - \frac{V_2}{R_2} = 0 \quad (10)$$

where I_{B2} and I_{C2} have been neglected because V_2 will be below turn-on for the minimum stimulating voltage across the V_1 junction. Alternatively, one could put the stimulus across the V_2 junction; the solution will be equivalent to the case presented. The minimum voltage pulse which can lead to latchup will certainly have a V_2 value less than the V_2 voltage at the critical point at the moment the stimulating pulse will turn off. The amount below the critical value is about 0.15 volts, which will be the assumed value of the fitting parameter K . So, equation 10 becomes:

$$I_{C1} = \frac{V_{2,crit} - K}{R_2} \quad (11)$$

I_{C1} is a function of V_1 , and the terms on the right are known values. So, this can be solved for V_1 . This step can not be done analytically, due to the exponential expressions in I_{C1} , but it is a relatively straightforward numerical procedure. With the value for V_1 known, equation 9 can now be solved for $V_{stim,min}$, the minimum stimulus voltage that will lead to latchup.

The final step is to approximate the minimum pulse duration which leads to latchup for an arbitrary pulse voltage. This is done by modifying the time-domain solution where both transistors are off, which can be solved analytically, so that it approximates the behavior for the entire range. The time-domain solution with both transistors off gives the voltage as a function of time. To simplify the analysis, the critical curve is assumed to be equal to the minimum stimulus voltage which will lead to latchup, for either V_{eb} or V_{be} biasing. The system equations to be solved in the region where both transistors are off simplify to the following:

$$\frac{\partial V_1(t)}{\partial t} \cdot C_{fact} = \left(\frac{-V_1(t)}{R1} + \frac{V_{stim}}{R1} \right) (Cc + Ce2) + \frac{V_2(t) Cc}{R2} \quad (12)$$

$$\frac{\partial V_2(t)}{\partial t} \cdot C_{fact} = \frac{-V_2(t) (Cc + Ce1)}{R2} - \left(\frac{-V_1(t)}{R1} + \frac{V_{stim}}{R1} \right) Cc \quad (13)$$

where

$$C_{fact} \equiv Cc Ce1 + Cc Ce2 + Ce1 Ce2 \quad (14)$$

Initially, both V_1 and V_2 are equal to zero. Using a laplace transform, the solutions to these equations are found to be:

$$V_1(t) = V_{stim} \left[1 - e^{-a_1 t} \cosh(a_2 t) + \frac{e^{-a_1 t} \sinh(a_2 t)}{\sqrt{B}} (R2(Cc + Ce2) - R1(Cc + Ce1)) \right] \quad (15)$$

$$V_2(t) = V_{stim} \frac{-2 R2 e^{-a_1 t} \sinh(a_2 t) Cc}{\sqrt{B}} \quad (16)$$

where

$$B = (C_c R_1 + C_c R_2 + C_e R_1 + C_e R_2)^2 - 4 R_1 R_2 C_{fact} \quad (17)$$

$$a_1 = 1/2 \frac{(C_e R_1 + C_e R_2 + C_c R_1 + C_c R_2)}{R_1 R_2 C_{fact}} \quad (18)$$

$$a_2 = 1/2 \frac{\sqrt{B}}{R_1 R_2 C_{fact}} \quad (19)$$

By inspection, a_1 will always be larger than a_2 ; so the exponential decay will always dominate the exploding cosh term. At time equal to infinity, the value of the emitter-base voltage is equal to the stimulus voltage. By subtracting off the value of the minimum pulse voltage to latch, the function will go to zero at time equal to infinity for a stimulus equal to the minimum pulse voltage. For other stimulus voltages, the time which makes the V1 function zero is calculated. Effectively, the critical curve is approximated as a constant equal to the minimum stimulus voltage to cause latchup, which means the V2 solution can be ignored. This is important since the correct V2 behavior in the vicinity of the critical curve would require solving in the region when one transistor is "on". Plotting the calculated time against the stimulus voltages approximates the exact minimum pulse time-stimulus voltage curve. By subtracting the minimum pulse voltage to latch, it will match at the minimum pulse voltage end of the curve. For large pulse voltages, most of the time is spent with both transistors off, and extrapolating this solution until it crosses the critical curve is an excellent approximation to the minimum pulse time required to latch. So, this approach will also provide agreement for the large stimulus voltage end of the curve as well.

To summarize the method of solution, there are basically three steps. The initial step is to calculate the unstable point. This will determine whether the structure is latchup immune or not. The next step is to approximate the critical curve, using the unstable point solution to determine the minimum pulse voltage required for latchup. Finally, for a given

stimulus, the voltages as a function of time are solved to determine the time at which the critical curve is crossed. By approximating the critical curve as a constant equal to the minimum pulse voltage required for latchup, the two voltages as a function of time do not need to be solved simultaneously in order to determine when the critical curve is crossed. The penalty is that high voltage pulses will cross the approximated critical curve before it crosses the actual critical curve, but the differences are acceptably small to justify the approximation. The only viable alternative is to find a more accurate approximation for the critical curve and a solution for the voltages when one transistor is conducting. But, numerical integration is easier than an analytical approach of this detail.

A comparison of the analytic and numerical solutions was done for two different sets of structure parameters. For the parameters used by Ker and Wu, the results for the unstable point calculation were as follows: (V_{eb1}, V_{be2}) equal to (0.747V, 0.776V) using numerical techniques, (0.748V, 0.779V) using the analytical approximation. The exact minimum pulse voltage to cause latchup was calculated as 0.918 volts and the approximation was 0.903 volts. For the Intel 8.5u structure, the unstable point was at (0.831 volts, 0.831 volts) with the analytical solution yielding (0.841 volts, 0.847 volts). The minimum pulse voltage to latch was 0.959 volts; the approximation was 0.971 volts. As can be seen, the agreement is acceptably close. It should be pointed out that the method for finding the linearizing point and emitter-base voltage of the non-pulsed transistor could be refined to improve the accuracy. But, the variation caused by adjusting the parameters is not nearly as great as the variation introduced when linearizing currents at two arbitrary voltage points.

VIII. Summary

The main goals of this project were to provide a computationally fast and accurate method for predicting when latchup will occur in CMOS structures when responding to a voltage pulse, to develop better analytical approximations than what are available in the current literature for the same case and finally apply these results to specific Intel test structures. These goals were accomplished successfully as detailed earlier in this report. Here we summarize the main highlights of this report:

A new condition for latchup to occur was presented. The new condition for latchup to occur is based on the possible existence of an unstable solution to the system equations of the CMOS structure. This mathematical condition provides latchup predictions which exactly match painstaking, trial-and-error SPICE simulations. The new latchup criterion is an improvement over previous empirical conditions, such as the transient beta product criterion, in terms of both accuracy and ability to justify its use.

For a given pulse voltage, the minimum pulse time which will lead to latchup can be calculated significantly faster using the new voltage-instability condition rather than through SPICE simulations. This helps in a rapid estimation of determination of process corners avoiding extensive SPICE simulations.

An improved method of analytically solving the system equations for the device was derived which allows an even faster characterization of latchup performance with more accuracy than the previous method using Laplace transforms. The analytical solution of determining when the system equations satisfy the new latchup condition

provides an excellent estimation of minimum pulse times without resorting to numerical evaluation of the differential system equations. The main advantage over the previous analytical solution was avoiding the approximation of transistor currents as having a linear dependence on voltage, when the actual dependence is exponential. The only drawback to the current solution is the high degree of complexity in the expressions, caused by the solution of a fourth-order polynomial. However, lower orders provided unacceptable accuracy, and the complexity of the expressions is no worse than the Laplace transform solution of the system equations.

Finally the accomplished results were applied to several Intel test structures. The physical layout and electrical data for the Intel test structures was provided by Dr. Shariar Ahmed and Mr. Jim Chen, for which the author is grateful. This allowed the use of actual device parameters in determining latchup behavior. For those structures that were found to be latchup immune, the device parameters were varied until latchup was possible so that the degree of latchup immunity could be judged. An extension of this would translate the device parameter changes back to process changes so that critical process corners could be determined.

References

1. D. B. Estreich, "The Physics and Modeling of Latchup and CMOS Integrated Circuits," Stanford Tech. Rep. no. G-201-9, Nov. 1980.
2. D. B. Estreich and R. W. Dutton, "Modeling latchup in CMOS integrated circuits," *IEEE Trans. Computer-Aided Design of ICAS*, vol. CAD-1, pp. 157-162, 1982.
3. R. R. Troutman, *Latchup in CMOS Technology*, New York, Kluwer Academic, 1986.
4. R. D. Rung and H. Momose, "DC holding and dynamic triggering characteristics of bulk CMOS," *IEEE Tran. Electron Devices*, vol. ED-31, pp. 1647-1655, 1983.
5. D. J. Sleeter and E. W. Enlow, "The Relationship of Holding Points and a General Solution for CMOS Latchup," *IEEE Tran. Electron Devices*, vol. 39, no. 11, pp. 2592-2599, Nov. 1992.
6. J. Seitchik et. al., "An analytic model of holding voltage for latch-up in epitaxial CMOS," *IEEE Electron Device Lett.*, vol. EDL-8, pp. 157-159, 1987.
7. M. Strzempa-Depre, J. Harter, et al., "Static and transient latchup simulation of VLSI-CMOS with an improved physical design model," *IEEE Tran. Electron Devices*, vol. ED-34, pp. 1290-1296, 1987.
8. T. Ohzone and H. Iwata, "Transient Latchup Characteristics in n-Well CMOS," *IEEE Tran. Electron Devices*, vol. 39, no. 8, pp. 1870-1875, August 1992.
9. G. J. Hu, "A Better Understanding of CMOS Latch-up," *IEEE Tran. Electron Devices*, vol. ED-31, pp. 62-67, 1984.
10. R. R. Troutman and H. P. Zappe, "A transient analysis of latchup in bulk CMOS," *IEEE Tran. Electron Devices*, vol. ED-30, pp. 170-179, 1983.
11. R. Fang and J. L. Moll, "Latchup model for the parasitic p-n-p-n path in bulk CMOS," *IEEE Tran. Electron Devices*, vol. ED-31, pp. 113-120, 1984.

12. G. Goto, et.al., "Latch-up immunity against noise pulses in a CMOS double well structure," in *IEDM Tech. Dig.*, pp. 169-171, 1983.
13. E. Hamdy and A. Mohsen, "Characterization and modeling of transient latchup in CMOS technology," in *IEDM Tech. Dig.*, pp. 172-175, 1983.
14. G.J. Hu and R. H. Bruce, "A CMOS Structure with high latchup holding voltage," *IEEE Electron Device Lett.*, vol. EDL-4, p. 414, 1983.
15. A. W. Wieder, C. Werner, and J. Harter, "Design model for bulk CMOS scaling enabling accurate latchup prediction," *IEEE Trans. Electron Devices*, vol. ED-30, pp. 240-245, 1983.
16. W. Li and M. E. Nokali, "Transient analysis for a new CMOS latchup model," *Solid State Elec.*, vol. 30, pp. 1331-1339, 1987.
17. C.-Y. Wu et al., "A new approach to model CMOS latchup," *IEEE Tran. Electron Devices*, vol. ED-32, pp. 1642-1653, 1985.
18. Yeu-haw Yang and Chung-yu Wu, "A New Criterion for Transient Latchup Analysis in Bulk CMOS," *IEEE Tran. Electron Devices*, vol. 36, no. 7, pp. 1336-47, July 1989.
19. S. Odanaka, M. Wakabayashi, and T. Ohzone, "The dynamics of latchup turn-on behavior in scaled CMOS," *IEEE Tran. Electron Devices*, vol. ED-32, pp. 1334-1340, 1985.
20. G. W. Mahoney et. al., "A statistical analysis program for computer-aided circuit design," *1973 IEEE Southeast Conf. Digest of Tech. Papers*, pp. C-1-1-6, May 1973.
21. M.-J. Chen and C-Y Wu, "An efficient two-dimensional model for CMOS latchup analysis," *Solid State Elec.*, vol. 29, no. 4, pp. 395-407, 1986.
22. R. Menozzi et. al., "Latch-up testing in CMOS IC's," *IEEE J. of Solid State Circuits*, vol. 25, no. 4, pp. 1010-1014, 1990.
23. F. Corsi et. al., "SPICE simulation of latch-up anomalous effects observed by electrical measurements and IR microscopy," *IEE Proc.*, vol. 136, Pt. G, no. 6, pp. 293-297, Dec. 1989.
24. A. Bandyopadhyay and A. B. Bhattacharayya, "Modeling Three Dimensional Effects in CMOS Latch-up," *IEICE Trans. Electronics*, vol. E75-C, no. 8, pp. 943-952, August 1992.

25. E. C. Sangiorgi, et al., "Two-dimensional numerical analysis of latchup in a VLSI CMOS technology," *IEEE Trans Electron Devices*, vol. ED-32, pp. 2117-2130, 1985.
26. M. R. Pinto and R. W. Dutton, "An efficient numerical model of CMOS latch-up," *IEEE Electron Device Lett.*, vol. EDL-4, p. 414, 1983.
27. M. R. Pinto and R. W. Dutton, "Accurate trigger condition analysis for CMOS latchup," *IEEE Electron Device Lett.*, vol. EDL-6, no. 2, pp. 100-102, Feb. 1985.
28. A. Chatterjee et.al., "Latchup modeling using PISCES," *IEEE Circuits and Devices Mag.*, pp. 35-38, Nov. 1987.
29. E. Sangiorgi, et.al., "Three-dimensional distribution of CMOS latch-up current," *IEEE Electron Device Lett.*, vol. EDL-8, pp. 154-156, 1987.
30. L. O. Chua, C. A. Desoer, and E. S. Kuh, *Linear and Nonlinear Circuits*, New York, McGraw-Hill, 1987.
31. F. Brauer and J.A. Nohel, *The Qualitative Theory of Ordinary Differential Equations*, New York, Dover Publication, Inc., 1989.
32. Chi-Tsong Chen, *Linear System Theory and Design*, New York, Holt, Rinehart, and Winston, 1984.
33. P. Antognetti and G. Massobrio, *Semiconductor Device Modeling with SPICE*, McGraw-Hill, 1988.
34. M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions*, New York, Dover Publication, Inc., 1972.
35. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C*, Cambridge University Press, 1988.

Appendix

Program listing for equation.c:

```
#include <math.h>
#define BF p[1][x]
#define BR p[2][x]
#define IS p[3][x]
#define IKF p[4][x]
#define ISE p[5][x]
#define ISC p[6][x]
#define TF p[7][x]
#define TR p[8][x]
#define CJC p[9][x]
#define CJE p[10][x]
#define VAF p[11][x]
#define VAR p[12][x]
#define NF p[13][x]
#define NE p[14][x]
#define NC p[15][x]
#define NR p[16][x]
#define MJE p[17][x]
#define MJC p[18][x]
#define PHI 0.75
#define FC 0.5

extern float p[20][2],Vt;

float Ibe1(float Vbe,int x)
{
    if (Vbe > -5.0*Vt*Nf)
        return( IS/BF*(exp(Vbe/(Vt*Nf))-1) );
    else
        return( IS/BF*(-1.0*((Vbe+5.0*Vt*Nf)/(Nf*Vt)+1.0)*exp(-5.0)) );
}

float Ibe2(float Vbe,int x)
{
    if (Vbe > -5.0*Vt*NE)
        return( ISE*(exp(Vbe/(Vt*NE))-1) );
    else
        return( ISE*(-1.0*((Vbe+5.0*Vt*NE)/(NE*Vt)+1.0)*exp(-5.0)) );
}
```

```

float Ibc1(float Vbc,int x)
{
    if (Vbc > -5.0*Vt*NR)
        return( IS/BR*(exp(Vbc/(Vt*NR))-1) );
    else
        return( IS/BR*(-1.0*((Vbc+5.0*Vt*NR)/(NR*Vt)+1.0)*exp(-5.0)) );
}

```

```

float Ice(float Vbe,float Vbc,int x)
{
    float Kb(float,float,int);
    return( (BF*Ibe1(Vbe,x)-BR*Ibc1(Vbc,x))/Kb(Vbe,Vbc,x) );
}

```

```

float Kb(float Vbe,float Vbc,int x)
{
    float q1,q2;

    q1=1/(1-(Vbe/VAR)-(Vbc/VAF));
    q2=BF*Ibe1(Vbe,x)/IKF;
    return(q1/2*(1+sqrt(1+4*q2)));
}

```

```

float Cbe2(float Vbe,float Vbc,int x)
{
    float Kb(float,float,int);

    return( TF*BF*(Ibe1(Vbe+1e-4,x)/Kb(Vbe+1e-4,Vbc,x)-
        Ibe1(Vbe-1e-4,x)/Kb(Vbe-1e-4,Vbc,x))/2e-4 );
}

```

```

float Cbe1(float Vbe,int x)
{
    if (Vbe < FC*PHI)
        return( CJE*pow(1.0-Vbe/PHI,-MJE) );
    else
        return( CJE*pow(1-FC,-MJE)*(1+MJE*(Vbe-FC*PHI)/(PHI*(1-FC))) );
}

```

```

float Cbc1(float Vbc,int x)
{
    if (Vbc < FC*PHI)
        return( CJC*pow(1.0-Vbc/PHI,-MJC) );
}

```



```

    else
        return( CJC*pow(1-FC,-MJC)*(1+MJC*(Vbc-FC*PHI)/(PHI*(1-FC)))) );
}

float Cbc2(float Vbe,float Vbc,int x)
{
    float Kb(float,float,int);

    return( TR*BR*(Ibc1(Vbc+1e-4,x)/Kb(Vbe,Vbc+1e-4,x)-
        Ibc1(Vbc-1e-4,x)/Kb(Vbe,Vbc-1e-4,x))/2e-4 );
}

```

Program listing for zero.c

```

#include <math.h>
#include <stdio.h>
#include <string.h>
#include "nr.h"
#include "equation.h"
#define DEL 1e-4

float p[20][2],Rs,Rw,Vt,I1,I2;
float v[3],result[3];
float **yp,*xp,dxsav;
int kmax,kount;

void main()
{
    void readparams(void);
    /* float rtbis(float (*func)(float),float,float,float);*/
    float fn1(float);
    float fn2(float);
    float (*fnptr[3])(float);
    float **matrix(int,int,int,int);
    float *vector(int,int);
    void derivs(float *,float *);
    float temp1,temp2;
    int i;

    readparams();
    fnptr[1]=fn1;
    fnptr[2]=fn2;

```

```

for (i=1;i<=2;i++) {
    for (v[2]=0.01;v[2] <= 1.001; v[2] += 0.01) {
        v[1]=-0.1;
        derivs(v,result);
        temp1=result[i];
        for (v[1]=-0.09; v[1] <= 1.1; v[1] += 0.01) {
            derivs(v,result);
            if (result[i]*temp1 < 0.0) {
                temp1=result[i];
                temp2=rtbis(fnptr[i],v[1]-0.01,v[1],1e-6);
                printf("%d %f %f\n",i,temp2,v[2]);
            }
        }
    }
}

```

/* file starts with temp, then transistor params, finally resistances */

void readparams(void)

```

{
    FILE *in;
    char inline[255];
    int i;

    in=fopen("params.dat","r");
    for (i=0;i<20;i++) {
        fgets(inline,255,in);
        sscanf(inline,"%f %f",&(p[i][0]),&(p[i][1]));
    }
    Rs=p[19][0];
    Rw=p[19][1];
    Vt=(p[0][0]+273.15)*8.617384436e-5; /* convert temp to volts */
    fclose(in);
}

```

float fn1(float vv)

```

{
    void derivs(float *,float *);

    v[1]=vv;
    derivs(v,result);
}

```

```

        return(result[1]);
    }

float fn2(float vv)
{
    void derivs(float *,float *);

    v[1]=vv;
    derivs(v,result);
    return(result[2]);
}

void usrfun(float *x, float **alpha, float *beta)
{
    void usrfn2(float *,float *);
    float temp[5][3];
    int i,j;

    x[1] -= DEL;
    usrfn2(x,temp[1]);
    x[1] += 2*DEL;
    usrfn2(x,temp[2]);
    x[1] -= DEL;
    x[2] -= DEL;
    usrfn2(x,temp[3]);
    x[2] += 2*DEL;
    usrfn2(x,temp[4]);
    x[2] -= DEL;
    usrfn2(x,beta);
    beta[1] = -beta[1];
    beta[2] = -beta[2];

    for (j=1;j<=2;j++)
        for (i=1;i<=2;i++)
            alpha[i][j]=(temp[j*2][i]-temp[j*2-1][i])/(2*DEL);
}

void usrfn2(float *v, float *if_)
{
    float vbc,ib1,ib2,ic1,ic2;

    vbc=-5.0+v[1]+v[2];
    ic2=Ice(v[2],vbc,1)-Ibc1(vbc,1);

```

```

    ic1=Ice(v[1],vbc,0)-Ibc1(vbc,0);
    ib1=Ibe1(v[1],0)+Ibe2(v[1],0)+Ibc1(vbc,0);
    ib2=Ibe1(v[2],1)+Ibe2(v[2],1)+Ibc1(vbc,1);
    if_[1]=ic2-ib1-v[1]/Rs;
        if_[2]=ic1-ib2-v[2]/Rw;
}

void derivs(float *v, float *dvdv)
{
    float delc,cc1,cc2,ce1,ce2,vbc,if_[3];

    vbc=-5.0+v[1]+v[2];
    cc1=Cbc1(vbc,0)+Cbc2(v[1],vbc,0);
    cc2=Cbc1(vbc,1)+Cbc2(v[2],vbc,1);
    ce1=Cbe1(v[1],0)+Cbe2(v[1],vbc,0);
    ce2=Cbe1(v[2],1)+Cbe2(v[2],vbc,1);
    delc=(cc1+cc2)*(ce1+ce2)+ce1*ce2;
    usrfn2(v,if_);

    dvdv[1]=(if_[1]*(cc1+cc2+ce2)-if_[2]*(cc1+cc2))/delc;
    dvdv[2]=(if_[2]*(cc1+cc2+ce1)-if_[1]*(cc1+cc2))/delc;
}

```

Program listing for criterv.c

```

#include <math.h>
#include <stdio.h>
#include <string.h>
#include "nr.h"
#include "equation.h"
#define DEL 1e-4

float p[20][2],Rs,Rw,Vt,I1,I2;
float v[3],result[3];
float **yp,*xp,dxsav;
int kmax,kount;

void main()
{
    void readparams(void);
    void odeint2(float *ystart, int nvar, float x1, float x2, float eps,
        float hi, float hmin, int *nok, int *nbad,

```

```

        void (dderivs)(float,float *,float *) ,
        void (*rkqc) (float *,float *,int,float *,float,float,float *,
                        float *,float *,void (*) (float,float *,float *)),int i) ;
/* float rtbis(float (*func)(float),float,float,float);*/
    float fn1(float);
    float fn2(float);
    float **matrix(int,int,int,int);
    float *vector(int,int);
    void derivs(float,float *,float *);
    float v1a,v1b,crit[3],y1sav[100],y2sav[100];
    float lookup(float,float *,float *,int);
    int flag=1,nok=0,nbad=0,i,savcnt=0;
    FILE *out;

    readparams();
    yp=matrix(1,2,1,100);
    xp=vector(1,100);
    dxsav=0.01;
    kmax=100;

    for (v[2]=0.01;v[2] <= 1.001 && flag; v[2] += 0.01) {
        v1a=rtbis(fn1,-0.1,1.1,1e-6);
        v1b=rtbis(fn2,-0.1,1.1,1e-6);
/*    printf("%f %f %f\n",v[2],v1a,v1b); */
        if ( (v1a>v1b) && (flag) ) {
            flag=0;
            crit[1]=v1b;
            crit[2]=v[2];
        }
    }
    if (flag) {
        printf("No critical point solution found");
        exit(0);
    }
    printf("Initial critical point at %f,%f\n",crit[1],crit[2]);

    v[1]=crit[1];
    v[2]=crit[2];
    mnewt(25,v,2,1e-6,0.0);
    printf(" Final critical point at %f,%f\n",v[1],v[2]);

/* now, v[1] & v[2] should have the critical voltage's to be
    solved backward in time using Runge-Kutta */

```

```

out=fopen("critcrv.dat","w");
for (i=0;i<4;i++) {
    crit[1]=v[1]-1e-5*sin(i*M_PI_2);
    crit[2]=v[2]+1e-5*cos(i*M_PI_2);
    odeint2(crit,2,0.0,-1.0,1e-6,1e-10,0.0,&nok,&nbad,derivs,rkqc,1+i/2);
    if (i<2) /* invert order so v1's increasing w/index */
for (nok=1;nok<=2;nok++)
    for (flag=1;flag <= kount/2;flag++) {
        v1a=yp[nok][flag];
        yp[nok][flag]=yp[nok][kount+1-flag];
        yp[nok][kount+1-flag]=v1a;
    }
if (i & 1)
    for (flag=1;flag<=kount;flag++) {
        v1a=lookup(yp[1][flag],y1sav,y2sav,savcnt);
        fprintf(out,"%12.7f
%12.7f\n",yp[1][flag],(yp[2][flag]+v1a)/2.0);
    }
    else {
        savcnt=kount;
        for (flag=1;flag<=savcnt;flag++) {
            y1sav[flag]=yp[1][flag];
            y2sav[flag]=yp[2][flag];
        }
    }
    if (i==1)
        fprintf(out,"%12.7f %12.7f\n",v[1],v[2]);
}
fclose(out);
}

```

/* file starts with temp, then transistor params, finally resistances */

```
void readparams(void)
```

```

{
    FILE *in;
    char inline[255];
    int i;

    in=fopen("params.dat","r");
    for (i=0;i<20;i++) {
        fgets(inline,255,in);

```

```

        sscanf(infile,"%f %f",&(p[i][0]),&(p[i][1]));
    }
    Rs=p[19][0];
    Rw=p[19][1];
    Vt=(p[0][0]+273.15)*8.617384436e-5; /* convert temp to volts */
    fclose(in);
}

float fn1(float vv)
{
    void usrfn2(float *,float *);

    v[1]=vv;
    usrfn2(v,result);
    return(result[1]);
}

float fn2(float vv)
{
    void usrfn2(float *,float *);

    v[1]=vv;
    usrfn2(v,result);
    return(result[2]);
}

void usrfun(float *x, float **alpha, float *beta)
{
    void usrfn2(float *,float *);
    float temp[5][3];
    int i,j;

    x[1] -= DEL;
    usrfn2(x,temp[1]);
    x[1] += 2*DEL;
    usrfn2(x,temp[2]);
    x[1] -= DEL;
    x[2] -= DEL;
    usrfn2(x,temp[3]);
    x[2] += 2*DEL;
    usrfn2(x,temp[4]);
    x[2] -= DEL;
    usrfn2(x,beta);
}

```

```

        beta[1] = -beta[1];
        beta[2] = -beta[2];

        for (j=1;j<=2;j++)
            for (i=1;i<=2;i++)
                alpha[i][j]=(temp[j*2][i]-temp[j*2-1][i])/(2*DEL);
    }

void usrfn2(float *v, float *if_)
{
    float vbc,ib1,ib2,ic1,ic2;

    vbc=-5.0+v[1]+v[2];
    ic2=Ice(v[2],vbc,1)-Ibc1(vbc,1);
    ic1=Ice(v[1],vbc,0)-Ibc1(vbc,0);
    ib1=Ibe1(v[1],0)+Ibe2(v[1],0)+Ibc1(vbc,0);
    ib2=Ibe1(v[2],1)+Ibe2(v[2],1)+Ibc1(vbc,1);
    if_[1]=ic2-ib1-v[1]/Rs;
    if_[2]=ic1-ib2-v[2]/Rw;
}

void derivs(float t, float *v, float *dvdx)
{
    float delc,cc1,cc2,ce1,ce2,vbc,if_[3];

    vbc=-5.0+v[1]+v[2];
    cc1=Cbc1(vbc,0)+Cbc2(v[1],vbc,0);
    cc2=Cbc1(vbc,1)+Cbc2(v[2],vbc,1);
    ce1=Cbe1(v[1],0)+Cbe2(v[1],vbc,0);
    ce2=Cbe1(v[2],1)+Cbe2(v[2],vbc,1);
    delc=(cc1+cc2)*(ce1+ce2)+ce1*ce2;
    usrfn2(v,if_);

    dvdx[1]=(if_[1]*(cc1+cc2+ce2)-if_[2]*(cc1+cc2))/delc;
    dvdx[2]=(if_[2]*(cc1+cc2+ce1)-if_[1]*(cc1+cc2))/delc;
}

#define MAXSTP 10000
#define TINY 1.0e-30

void odeint2(float ystart[],int nvar,float x1,float x2,float eps,float h1,
             float hmin,int *nok,int *nbad,void (*derivs)(float,float *,float *),
             void (*rkqc)(float *,float *,int,float *,float,float,float *,float *,

```



```

float *,void (*)(float,float *,float *) ),int loop )
{
    int nstp,i;
    float xsav,x,hnext,hdid,h;
    float *yscal,*y,*dydx,*vector(int,int);
    void nrerror(char *),free_vector(float *,int,int);
    extern float **yp,*xp,dxsav;
    extern int kmax,kount;

    yscal=vector(1,nvar);
    y=vector(1,nvar);
    dydx=vector(1,nvar);
    x=x1;
    h=(x2>x1) ? fabs(h1) : -fabs(h1);
    *nok>(*nbad)=kount=0;
    for (i=1;i<=nvar;i++)
        y[i]=ystart[i];
    if (kmax > 0) xsav=y[loop]-dxsav*2.0;
    for (nstp=1;nstp<=MAXSTP;nstp++) {
        (*derivs)(x,y,dydx);
        for (i=1;i<=nvar;i++)
            yscal[i]=fabs(y[i])+fabs(dydx[i]*h)+TINY;
        if (kmax > 0) {
            if (fabs(y[loop]-xsav) > fabs(dxsav)) {
                if (kount < kmax-1) {
                    xp[++kount]=x;
                    for (i=1;i<=nvar;i++)
                        yp[i][kount]=y[i];
                    xsav=y[loop];
                }
            }
        }
        if ((x+h-x2)*(x+h-x1) > 0.0) h=x2-x;
        (*rkqc)(y,dydx,nvar,&x,h,eps,yscal,&hdid,&hnext,derivs);
        if (hdid==h) ++(*nok);
        else ++(*nbad);
        if ( (y[1]<0.01) || (y[2]<0.01) || (fabs(x)>2e-7) ) {
            for (i=1;i<=nvar;i++)
                ystart[i]=y[i];
            if (kmax) {
                xp[++kount]=x;
                for (i=1;i<=nvar;i++) yp[i][kount]=y[i];
            }
        }
    }
}

```

```

        free_vector(dydx,1,nvar);
        free_vector(y,1,nvar);
        free_vector(yscal,1,nvar);
        return;
    }
    if (fabs(hnext) <= hmin) nrerror("Step size too small in ODEINT");
    h=hnext;
}
nrerror("Too many steps in routine ODEINT");
}

float lookup(float y,float *yy1,float *yy2,int count)
{
    int i=2;

    while (yy1[i]<y && i<count) ++i;
    return ( (y-yy1[i])*(yy2[i]-yy2[i-1])/(yy1[i]-yy1[i-1])+yy2[i] );
}

```

Program listing for pulse.c

```

#include <math.h>
#include <stdio.h>
#include <string.h>
#include "nr.h"
#include "equation.h"
#define DEL 1e-4

float p[20][2],Rs,Rw,Vt,I1,I2;
float v[3],save[6],cv1[100],cv2[100];
float **yp,*xp,dxsav;
int kmax,kount,ccount;

void main()
{
    void readparams(void);
    void odeint2(float *ystart, int nvar, float x1, float x2, float eps,
        float hi, float hmin, int *nok, int *nbad,
        void (dderivs)(float,float *,float *) ,
        void (*rkqc) (float *,float *,int,float *,float,float,float *,
        float *,float *,void (*) (float,float *,float *)) ) ;
    float **matrix(int,int,int,int);
}

```

```

float *vector(int,int);
void derivs(float,float *,float *);
float v1a,v1b,start[3],y1sav[100],y2sav[100];
float d[3],mid[3],temp[3],vnoise,delta;
float v2lookup(float);
int nok=0,nbad=0,i,j,k;
FILE *out;
char *fname;

readparams();
yp=matrix(1,2,1,100);
xp=vector(1,100);
dxsav=0.01;
kmax=100;
start[1]=start[2]=0.0;
fname="pulse0.dat";

for (i=0;i<2;++i) {
    vnoise=4.5;
    delta=0.5;
    ++fname[5];
    out=fopen(fname,"w");
    do {
        vnoise -= delta;
        I1=(1-i)*vnoise/Rs;
        I2=i*vnoise/Rw;
        odeint2(start,2,0.0,1.0,1e-6,0.2e-10,0.0,&nok,&nbad,derivs,rkqc);
        if (save[3]<2e-7) {
            for (j=0;j<3;j++) {
                d[j]=save[j+3]-save[j];
                mid[j]=save[j];
            }
            for (k=1; k<=30 && d[0]>1e-12 ;k++) {
                for (j=0;j<3;j++)
                    temp[j]=mid[j]+(d[j] *= 0.5);
                if (temp[2] < v2lookup(temp[1]) )
                    for (j=0;j<3;j++)
                        mid[j]=temp[j];
            }
        }
    } while (vnoise>0);
    fprintf(out,"%8.2f %8.3f %f\n",I1*Rs+I2*Rw,mid[0]*1e9,mid[1],mid[2]);
    fprintf(out,"%8.2f %8.3f\n",I1*Rs+I2*Rw,mid[0]*1e9);
}

```

```

    }
    if (fabs(vnoise-3.0) < .01) delta=0.2;
    if (fabs(vnoise-2.0) < .01) delta=0.1;
    if ( (save[3] >= 2e-7) && (delta>=0.1) )    {
        vnoise += delta;
        delta *= 0.2;
        save[3]=0.0;
    }
}
while (save[3] < 2e-7);
fclose(out);
}
}

/* file starts with temp, then transistor params, finally resistances */
void readparams(void)
{
    FILE *in;
    char inline[255];
    int i;

    in=fopen("params.dat","r");
    for (i=0;i<20;i++) {
        fgets(inline,255,in);
        sscanf(inline,"%f %f",&(p[i][0]),&(p[i][1]));
    }
    Rs=p[19][0];
    Rw=p[19][1];
    Vt=(p[0][0]+273.15)*8.617384436e-5; /* convert temp to volts */
    fclose(in);
    in=fopen("critcrv.dat","r");
    i=0;
    while (fgets(inline,255,in) != NULL) {
        if ( sscanf(inline,"%f %f",&(cv1[i]),&(cv2[i]))==2 )
            ++i;
    }
    ccount=i-1;
    fclose(in);
}

void usrfun(float *x, float **alpha, float *beta)
{
    void usrfn2(float *,float *);

```

```

float temp[5][3];
int i,j;

x[1] -= DEL;
usrfn2(x,temp[1]);
x[1] += 2*DEL;
usrfn2(x,temp[2]);
x[1] -= DEL;
x[2] -= DEL;
usrfn2(x,temp[3]);
x[2] += 2*DEL;
    usrfn2(x,temp[4]);
    x[2] -= DEL;
usrfn2(x,beta);
    beta[1] = -beta[1];
beta[2] = -beta[2];

for (j=1;j<=2;j++)
    for (i=1;i<=2;i++)
        alpha[i][j]=(temp[j*2][i]-temp[j*2-1][i])/(2*DEL);
}

void usrfn2(float *v, float *if_)
{
    float vbc,ib1,ib2,ic1,ic2;

    vbc=-5.0+v[1]+v[2];
    ic2=Ice(v[2],vbc,1)-Ibc1(vbc,1);
    ic1=Ice(v[1],vbc,0)-Ibc1(vbc,0);
    ib1=Ibe1(v[1],0)+Ibe2(v[1],0)+Ibc1(vbc,0);
    ib2=Ibe1(v[2],1)+Ibe2(v[2],1)+Ibc1(vbc,1);
    if_[1]=ic2-ib1-v[1]/Rs+I1;
    if_[2]=ic1-ib2-v[2]/Rw+I2;
}

void derivs(float t, float *v, float *dvdx)
{
    float delc,cc1,cc2,ce1,ce2,vbc,if_[3];

    vbc=-5.0+v[1]+v[2];
    cc1=Cbc1(vbc,0)+Cbc2(v[1],vbc,0);
    cc2=Cbc1(vbc,1)+Cbc2(v[2],vbc,1);
    ce1=Cbe1(v[1],0)+Cbe2(v[1],vbc,0);

```

```

ce2=Cbe1(v[2],1)+Cbe2(v[2],vbc,1);
delc=(cc1+cc2)*(ce1+ce2)+ce1*ce2;
usrfn2(v,if_);

dvdx[1]=(if_[1]*(cc1+cc2+ce2)-if_[2]*(cc1+cc2))/delc;
dvdx[2]=(if_[2]*(cc1+cc2+ce1)-if_[1]*(cc1+cc2))/delc;
}

#define MAXSTP 450
#define TINY 1.0e-30

void odeint2(float ystart[],int nvar,float x1,float x2,float eps,float h1,
            float hmin,int *nok,int *nbad,void (*derivs)(float,float *,float *),
            void (*rkqc)(float *,float *,int,float *,float,float,float *,float *,
            float *,void (*)(float,float *,float *)) )
{
    int nstp,i;
    float xsav,x,hnext,hdid,h;
    float *yscal,*y,*dydx,*vector(int,int);
    void nrerror(char *),free_vector(float *,int,int);
    float v2lookup(float);
    extern float **yp,*xp,dxsav;
    extern int kmax,kount;

    yscal=vector(1,nvar);
    y=vector(1,nvar);
    dydx=vector(1,nvar);
    x=x1;
    h=(x2>x1) ? fabs(h1) : -fabs(h1);
    *nok>(*nbad)=kount=0;
    for (i=1;i<=nvar;i++)
        y[i]=ystart[i];
    if (kmax > 0) xsav=y[1]-dxsav*2.0;
    for (nstp=1;nstp<=MAXSTP;nstp++) {
        (*derivs)(x,y,dydx);
        for (i=1;i<=nvar;i++)
            yscal[i]=fabs(y[i])+fabs(dydx[i]*h)+TINY;
        save[0]=x;
        save[1]=y[1];
        save[2]=y[2];
/*    if (kmax > 0) {
            if (fabs(y[1]-xsav) > fabs(dxsav)) {
                if (kount < kmax-1) {

```

```

                                xp[++kount]=x;
                                for (i=1;i<=nvar;i++)
                                    yp[i][kount]=y[i];
                                xsav=y[1];
                            }
                        }
                    }
*/    if ((x+h-x2)*(x+h-x1) > 0.0) h=x2-x;
        (*rkqc)(y,dydx,nvar,&x,h,eps,yscal,&hnext,derivs);
        if (hnext==h) ++(*nok);
        else ++(*nbad);
        if ( ( y[2]>v2lookup(y[1]) ) || (fabs(x)>2e-7) ) {
/*    for (i=1;i<=nvar;i++)
            ystart[i]=y[i];
            if (kmax) {
                xp[++kount]=x;
                for (i=1;i<=nvar;i++) yp[i][kount]=y[i];
            }
*/    save[3]=x;
        save[4]=y[1];
        save[5]=y[2];

        free_vector(dydx,1,nvar);
        free_vector(y,1,nvar);
        free_vector(yscal,1,nvar);
        return;
    }
    if (fabs(hnext) <= hmin) nrerror("Step size too small in ODEINT");
    h=hnext;
}
save[3]=1.0;
free_vector(dydx,1,nvar);
free_vector(y,1,nvar);
free_vector(yscal,1,nvar);
/*    nrerror("Too many steps in routine ODEINT");*/
}

float v2lookup(float v1)
{
    int i=1;
    while (cv1[i]<v1 && i<ccount) ++i;
    return ( (v1-cv1[i])*(cv2[i]-cv2[i-1])/(cv1[i]-cv1[i-1])+cv2[i] );
}

```

Vita

The author was born in Kansas City, Kansas on September 14, 1966. He graduated from Shawnee Mission East High School in 1985. He attended the California Institute of Technology, where he received a Bachelor of Science in Applied Physics in June, 1989. After leaving Caltech, he worked for IBM in East Fishkill as a diagnostics engineer in the packaging division. In 1992, after working for IBM for 3 years, he went to the Oregon Graduate Institute and received a Master of Science in Electrical Engineering in June, 1993, and then continued to pursue his doctoral degree. During the course of study, he has been an intern at Intel in the Intel Development Labs.