FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Design of ESD Protections for Analog Pins of Integrated Circuits

Cesar Augusto da Fonseca Vidal Junior

Mestrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Prof. Manuel Cândido Duarte dos Santos

April 8, 2026

Design of ESD Protections for Analog Pins of Integrated Circuits

Cesar Augusto da Fonseca Vidal Junior

Mestrado em Engenharia Eletrotécnica e de Computadores

Approved in oral examination by the committee:

President: Prof. João Paulo de Castro Canas Ferreira
Referee: Prof. Pedro Mendonça dos Santos
Referee: Prof. Manuel Cândido Duarte dos Santos

April 8, 2026

Resumo

O trabalho tem como objetivo desenvolver uma metodologia que permita projetar e validar proteções contra descarga eletrostática (ESD) para pinos analógicos de Circuitos Integrados (IC) ainda nas fases iniciais de desenvolvimento, reduzindo dependência do método de "tentativa e erro" e aumentando a probabilidade de sucesso nos primeiros ciclos fabrico. A motivação surge da limitação dos modelos fornecidos pelas fabricantes, que não representam fenómenos de `snapback` ou a ação do transístor bipolar parasita presentes em dispositivos de proteção contra ESD.

Os métodos combinam simulação física em TCAD com modelação do circuito em SPICE. Primeiro, estruturas 2D e 3D de porta aterrada NMOS (GGNMOS) são construídas e simuladas para extrair parâmetros físicos relevantes, como $V_{\mathbf{t1}}$, $V_{\mathbf{h}}$ e $V_{\mathbf{t2}}$. Em seguida, é criado um GGNMOS em SPICE baseado em `BSIM4` e `VBIC`. Por fim, um programa foi desenvolvido para uma calibração automatizada, que compara curvas de referência com combinações de variações paramétricas e elege os parâmetros que melhor replicam o comportamento.

Os resultados evidenciam como parâmetros, e.g. L, W, DCGS, SCGS e TOXE, influenciam o comportamento de `snapback`. O modelo calibrado reproduz com certa precisão a curva de referência e reduz os erros de $V_{\mathbf{h}}$ e $V_{\mathbf{t2}}$ para poucos milivolts. Testes de qualificação para os modelos de Corpo Humano (HBM) e Dispositivo Carregado (CDM) demonstram que o modelo conduz a corrente maioritariamente pelo GGNMOS e reduz significativamente a sobrecarga no IC quando a proteção é incluída.

Por fim, concluiu-se que a metodologia melhora a compreensão dos mecanismos ESD, permite prever alguns detalhes do comportamento do dispositivo de proteção antes do fabrico e fornece um modelo reutilizável para calibração após fabricação do IC. Destaca-se ainda a relevância da metologia de trabalho para estudo de outras estruturas de `snapback`, como o SCR.

Abstract

The work aims to develop a methodology that enables the design and validation of Electrostatic Discharge (ESD) protections for analog Integrated Circuit (IC) pins in the early stages of development, reducing reliance on the "trial-and-error" approach and increasing the likelihood of first tape-out success. The motivation arises from the limitations of the models provided by semiconductor manufacturers, which do not represent `snapback` phenomena or the action of the parasitic bipolar transistor present in ESD protection devices.

The methods combine physical simulation using Technology Computer-Aided Design (TCAD) with circuit-level modeling in SPICE. First, 2D and 3D Grounded Gate NMOS (GGNMOS) structures are built and simulated to extract relevant physical parameters such as $V_{\mathbf{t1}}$, $V_{\mathbf{h}}$, and $V_{\mathbf{t2}}$. Then, a GGNMOS model in SPICE is created based on `BSIM4` and `VBIC`. Finally, a script performs automated calibration by comparing reference curves with combinations of parametric variations and selecting the parameter sets that best reproduce the behavior.

The results show how parameters such as channel length (L), channel width (W), Drain Contact to Gate Spacing (DCGS), Source Contact to Gate Spacing (SCGS), and Electrical gate equivalent oxide thickness (TOXE) influence the `snapback` behavior. The calibrated model reproduces the reference curve with some accuracy and reduces the errors in $V_{\mathbf{h}}$ and $V_{\mathbf{t2}}$ to just a few millivolts. Qualification tests under Human Body Model (HBM) and Charged Device Model (CDM) demonstrate that the model conducts most of the discharge current through the GGNMOS and significantly reduces the stress on the IC when protection is included.

Finally, it is concluded that the methodology improves the understanding of ESD mechanisms, allows predicting certain aspects of the protection device behavior before tape-out, and provides a reusable model for post-fabrication calibration of the IC. The relevance of the methodology for studying other `snapback` structures, such as the Silicon Controlled Rectifier (SCR), is also highlighted.

UN Sustainable Development Goals

The United Nations Sustainable Development Goals (SDGs) provide a global framework for achieving a better and more sustainable future for all. It includes 17 goals[1] to address the world's most pressing challenges, including poverty, inequality, climate change, environmental degradation, peace, and justice.

In the context of this work, the relevance to the SDGs is linked to Integrated Circuit (IC) reliability. As semiconductor manufacturing evolves, ICs integrate more functions while device margins become tighter, which increases sensitivity to Electrostatic Discharge (ESD) at the pin level. For high bandwidth and RF interfaces, on chip ESD protection must be designed together with performance constraints, so robust verification becomes part of the development strategy. This aligns with SDG 9 by supporting innovation in advanced ICs, and with SDG 12 by reducing avoidable material waste and losses associated with ESD related failures.

Industrial studies show that ESD-related failures directly increase costs through material loss, and that improving ESD protection practices is treated as cost avoidance with clear economic benefits [1, p. 4]. In advanced System on a Chip (SoC) development, the literature also highlights that ESD weaknesses found after tape-out translate into long debugging cycles and additional design iterations, with strong schedule and cost exposure [2, p. 178]. The design process proposed in this work reduces the number of design cycles based on trial and error, using an Electronic Design Automation (EDA) based methodology to assess on-chip protection behavior at initial stages of the development flow.

Finally, technology reports published by Electrical Overstress (EOS)/ESD organizations document that qualification target levels continue to move down for some high performance IOs, driven by the tradeoff between robustness and performance. This trend reinforces the need for systematic on-chip protection verification supported by simulation tools, instead of trial and error methodology, which is consistent with SDG 9 through resilient industrial innovation, and with SDG 12 through reduction of avoidable losses across development and production. [3, pp. 2–3].

The specific Sustainable Development Goals mentioned have the following names:

[1] https://sdgs.un.org/goals

SDG	Target	Contribution	Performance Indicators and Metrics
9	9.4	By improving EDA verification of on-chip ESD protection in ICs, this work strengthens product reliability by preventing avoidable catastrophic failures and early-stage redesigns, enabling more robust and resource-efficient industrial development.	• Yield improvement reported up to 10.73% after strengthening ESD practices. [1]
	9.5	The methodology proposed in this work supports the conversion of on-chip protection to new ESD targets, under tighter margins and increasing IC performance requirements.	• Typical Human Body Model (HBM) production targets reported to have moved from 2 kV to 1 kV. [3] • Projected HBM targets below 250 V by 2030 for some high performance ICs. [3] • Projected Charged Device Model (CDM) targets down to 125 V for some devices. [3]
12	12.2	The methodology proposed in this work enhances IC development sustainability by minimizing redesign motivated by ESD-related issues and by focusing exploration on early analysis and verification of on-chip protection, reducing material and resource waste.	• Split lot failure ratios reported from 1.9:1 up to 5.5:1 between unprotected and protected lots. [1] • Outgoing defect reduction reported as 3:1 after ESD controls were implemented. [1]

Acknowledgements

Quero agradecer a todas as pessoas que fizeram parte desta caminhada e que me ajudaram a chegar até aqui.

Ao meu orientador, professor Cândido Duarte, deixo um agradecimento sincero por toda a experiência, paciência e incentivo ao longo deste trabalho. As suas sugestões e instigações foram cruciais para esse trabalho.

À empresa SiliconGate, agradeço pelo ambiente acolhedor e pela forma como fui recebido desde o primeiro dia. Ao professor Marcelino Santos, aos engenheiros Artur Ribeiro, João Brogueira e, em especial, quero agradecer ao engenheiro Miguel Pina por todo o suporte durante o trabalho.

Aos meus amigos, Mário Ledo, Sofia Silva, Nuno Jesus e Francisco Riesenberger obrigado por todas as conversas, risadas e momentos que passamos.

Acima de tudo, agradeço aos meus pais, Cesar Vidal e Célia Vidal, e ao meu irmão João Vidal. O apoio, o carinho e a força que sempre me deram foram essenciais durante todo o percurso, muito obrigado por acreditarem em mim. Nada disto teria sido possível sem vocês.

Cesar Augusto da Fonseca Vidal Junior

"Inventions are, above all, the result of stubborn work."

Santos Dumont

Contents

List of Figures

List of Tables

List of Acronyms

ANSI American National Standards Institute

BJT Bipolar Junction Transistor

CBM Charged Board Model

CC-TLP Capacitively Coupled Transmission Line Pulse

CDE Cable Discharge Event

CDM Charged Device Model

CMOS Complementary Metal-Oxide-Semiconductor

CSM Charged Strip Model

DC Direct Current

DCGS Drain Contact to Gate Spacing

DUT Device Under Test

EDA Electronic Design Automation

EOS Electrical Overstress

EPA Electrostatic Protection Area

ESD Electrostatic Discharge

ESDA ESD Association

FICBM Field Induced Charged Board Model

FICDM Field Induced Charged Device Model

GGNMOS Grounded Gate NMOS

HBM Human Body Model

HMM Human Metal Model

IC Integrated Circuit

IEC International Electrotechnical Commission

IO Input/Output

JEDEC Joint Electron Device Engineering Council

L channel length

LDMOS Lateral Diffused Metal-Oxide-Semiconductor

LNPN lateral NPN

LVTSCR Low Voltage Triggered SCR

MM Machine Model

MOS Metal-Oxide-Semiconductor

MOSFET Metal-Oxide-Semiconductor Field-Effect Transistor

NMOS N-type MOS

OVAT one-variable-at-a-time

PCB Printed Circuit Board

PDK Process Design Kit

SCBE Substrate Current Induced Body Effect

SCGS Source Contact to Gate Spacing

SCR Silicon Controlled Rectifier

SDE Sentaurus Structure Editor

SDEVICE Sentaurus Device

SDM Socket Discharge Model

SEED System-Efficient ESD Design

SoC System on a Chip

SOA Safe Operating Area

SPICE Simulation Program with Integrated Circuit Emphasis

TCAD Technology Computer-Aided Design

TCL Tool Command Language

TLP Transmission Line Pulse

TOXE Electrical gate equivalent oxide thickness

TVS Transient Voltage Suppressor

VF-TLP Very Fast Transmission Line Pulse

W channel width

Chapter 1

Introduction

Electrostatic charging happens everywhere in daily life. A small spark may appear when a charged person touches a conductive surface or even another person with a different charge, but many discharges are invisible and only become evident through unexpected behavior in electronic devices, such as temporary malfunction or even permanent damage. In electronic products, these discharges can reach sensitive nodes and create a Electrostatic Discharge (ESD) that is much more severe than any normal operation (under specific conditions, ESD levels can exceed 10 kV), even if the discharge itself looks harmless. This idea is summarized by reminding that static sparking from common activities can generate ESD damage and that the resulting damage can easily go undetected [4, p. 1]. An example of a visible discharge during human interaction is shown in Figure 1.1.

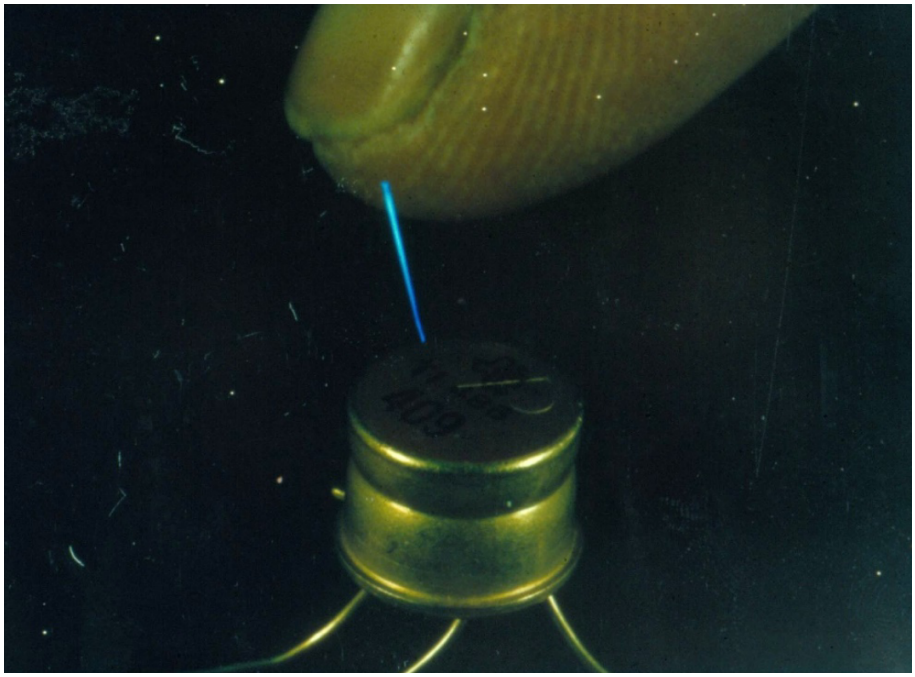


Figure 1.1: Example of a visible spark produced by an ESD event during human interaction with a conductor. Extracted from: [4, p. 1]

In the IC industry, ESD is a major contributor to failures and economic loss. Some companies have reported losses of 30–40% and damages amounting to billions of dollars [5]. Additionally, with the introduction of new technologies for Integrated Circuit (IC) production, which implies smaller transistor and via dimensions, there is an increase in current density and greater sensitivity of IC. This combination explains why ESD remains an issue throughout the full product life cycle.

For this reason, industry applies structured control measures to reduce the probability of electrostatic events. Foundries and assembly lines invest in Electrostatic Protection Areas (EPAs), using grounding practices, material selection, and equipment that helps neutralize charge accumulation, such as ionizers. In practice, this procedure is also supported by clear visual identification of susceptible items and protected work areas. Figure 1.2 and Figure 1.3 show examples of commonly used symbols to mark ESD susceptible devices and the required protected handling conditions.

Figure 1.2: Symbol indicating ESD susceptibility, used to mark items that require controlled handling. Extracted from: [6]

Figure 1.3: Symbol indicating an ESD protected handling area or protective equipment for controlled operation. Extracted from: [6]

However, although EPAs reduce the probability of damage caused by ESD, such measures are not sufficient to guarantee protection during manufacturing, handling, packaging, transportation, and assembly. In addition, practical weaknesses in infrastructure can introduce unexpected charging and discharge paths. Figure 1.4 illustrates a concrete example, where plastic insulating rings at rack junctions prevent proper grounding of stored items, enabling electrostatic charge build-up during storage and handling [4, p. 9].

A further difficulty is that ESD damage is often hard to detect. Some ESD does not leave visible damage that can be confirmed by standard optical inspection. Even when inspection is done, the damage signature may remain invisible at moderate zoom levels [4, pp. 1–2]. As a consequence, the discharge event may not be recognized when it happens, and the device may continue to be handled as if no stress occurred. When a failure is later suspected, the investigation may no longer be able to identify which event or moment caused the damage, since the discharge

Figure 1.4: Example of unintended loss of grounding in storage infrastructure: plastic insulating rings at rack connector junctions can prevent proper grounding and allow electrostatic charge build-up during storage and handling. Extracted from: [4, p. 9]

may not have been recognized at the time it occurred. Figure 1.5 illustrates a typical damage pattern, while Figure 1.6 shows a detailed view of localized damage sites on the die.

This limited visibility also affects how organizations diagnose the problem. Estimating the cost of ESD is difficult because many events are invisible and less understood than other types of failures [1, p. 2]. The most important impacts cannot be reduced to a single number, since ESD-related damage can affect several aspects of a product's lifecycle. In practice, this means that

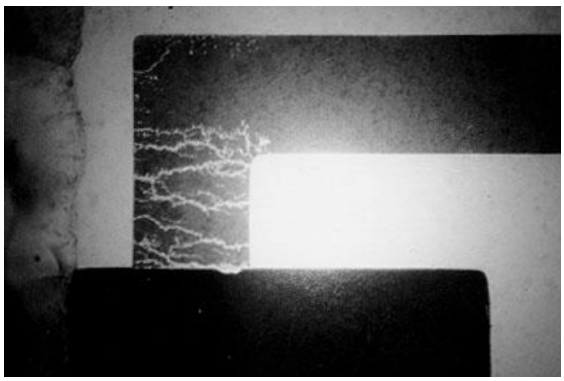


Figure 1.5: Example of transverse cracking associated with relatively high-voltage ESD damage (vicinal illumination view). Extracted from: [4, p. 14]

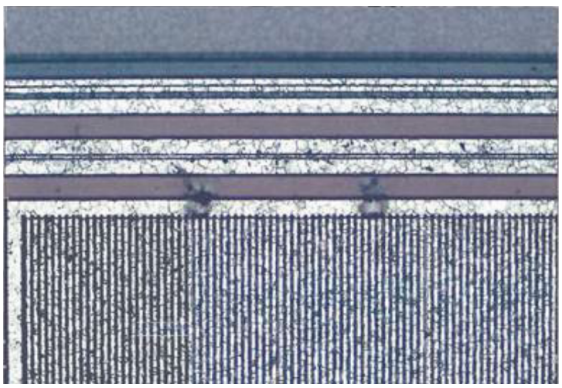


Figure 1.6: Detailed view of localized damage sites on the die attributed to an ESD event. Extracted from: [4, p. 12]

prevention and protection have to be treated as permanent needs, supported by evidence rather than assumptions about the source of the problem.

From the design perspective, this leads to a conclusion: since it is practically impossible to guarantee external protection for an IC, the protection strategy must include on-chip. The challenge is to meet operational requirements while remaining reliable against ESD scenarios. Despite that, the Process Design Kits (PDKs) provided by foundries only includes information about device behavior under normal operating conditions. As a result, the ESD behavior of the protection device is often determined through repetitive "trial and error", because the available design models do not simulate how the device will respond under ESD stress.

Based on this, the focus of this work is:

- reduce the dependence on purely experience-based iterations during the initial development of a new protection circuit, particularly in the first design and tape-out attempts;
- use EDA to build a device protection, simulate an ESD event, and extract the device behavior, including the I–V characteristic and the associated temperature;
- perform a physical analysis to support parameter extraction from the reference device;
- develop a SPICE macromodel for the protection device, using standard compact models to represent the snapback behavior;
- apply the calibrated macromodel within a methodology that automates parameter extraction and calibration, to improve the efficiency of the design process and support System-Efficient ESD Design (SEED).

Chapter 2

Background and bibliography review

2.1 Electrostatic Phenomena

It is crucial to understand the basic ideas of electrostatic phenomena in order to comprehend the problems and solutions related to ESD protection. In Table 2.1, it is possible to observe some cases of electrostatic charging and discharging under various conditions, such as the air's relative humidity, which contributes to an important role in the triboelectric effect.

Table 2.1: Examples of generated electrostatic charge. Source: [7, p. 2]

Means of generation	**10-25% Relative Humidity (kV)**	**65-95% Relative Humidity (kV)**
Walking across carpet	35	1.5
Walking across vinyl tile	12	0.25
Worker at bench	6	0.1
Poly bag picked up from bench	20	1.2
Chair with urethane foam	18	1.5

Hence, it can be observed that increasing air humidity reduces the generation of electrostatic charges, as it enhances air conductivity. However, it is important to emphasize that high levels of humidity in the air are not acceptable for certain environments.

2.1.1 Triboelectric

When two materials come into contact and are then separated, the triboelectric effect occurs, resulting in the transfer of electrons from one material to another. Position in the Table 2.2 determines whether a substance tends to charge positively or negatively. Glass and the human body tend to charge negatively, while polyurethane tends to charge negatively.

Table 2.2: Typical triboelectric series. Extracted from: [7, p. 2]

Material	**Charge Tendency**
Air	+++++++++++++++
Human skin	++++++++++++
Acetate	++++++++++
Glass	+++++++++
Human hair	++++++++
Nylon	+++++++
Wool	++++++
Silk	+++++
Aluminum	++++
Paper	+++
Cotton	++
Steel	+
Hard rubber	-
Nickel, copper	- -
Brass, Silver	- - -
Gold, Platinum	- - - -
Acetate fiber (rayon)	- - - - -
Polyester (mylar)	- - - - - - -
Polystyrene (styrofoam)	- - - - - - - - -
Polyurethane	- - - - - - - - - -
Polyvinyl chloride (PVC)	- - - - - - - - - - -
Silicon	- - - - - - - - - - - -
Teflon	- - - - - - - - - - - - -
Silicon rubber	- - - - - - - - - - - - - -

Since catastrophic or latent damage occurs during assembly, when carrying out tests and during operation cycles, caused by ESD events, this intrinsic characteristic of the materials helps to understand the impact on the entire life cycle of IC.

2.1.2 Electrostatic Induction

The existence of a charged body close to another object results in a redistribution of electric charges within that object, without direct physical contact. This phenomenon is known as electrostatic induction. When the second object is electrically neutral and connected to a reference potential (ground), this redistribution may result in a net charge being retained after the grounding path is removed.

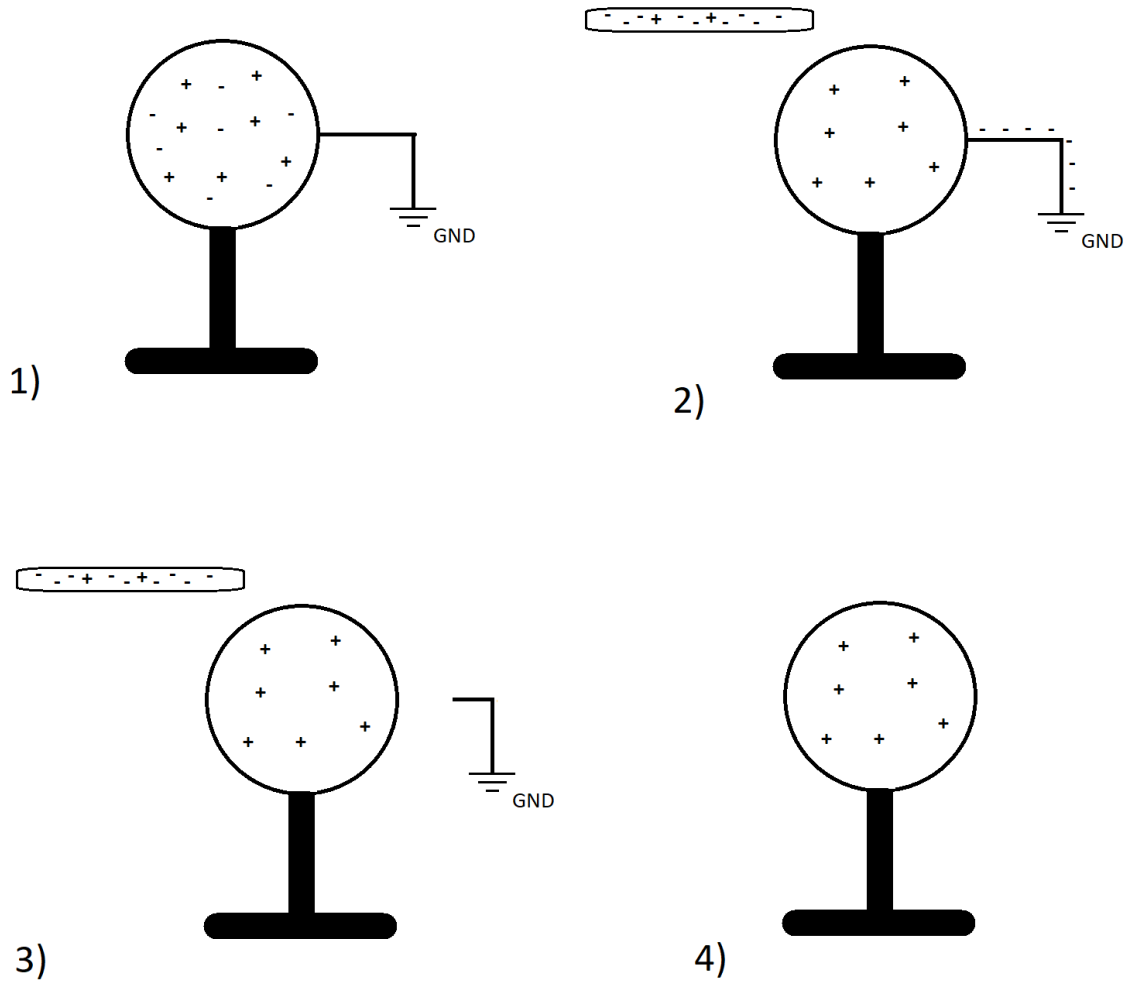


Figure 2.1: Example of charge accumulation through electrostatic induction

As illustrated in Figure 2.1, the object is initially electrically neutral and grounded. When a negatively charged body approaches, the electric field associated with this body repels free electrons within the object. Due to the grounding connection, these electrons can flow toward the reference potential, while positive charges remain bound within the object. This charge movement continues until electrostatic equilibrium is established. After the grounding connection is removed, the object retains a net deficit of electrons and therefore becomes positively charged regarding the reference potential.

Electrostatic induction is particularly relevant in the context of integrated circuits, where conductive structures inside the IC and package leads can experience charge redistribution due to nearby charged objects or tools. Even without direct physical contact, this mechanism can induce substantial electrostatic charge accumulation on electrically isolated nodes, thereby increasing the risk of an ESD event. Such induced charging effects are especially important in scenarios in which rapid discharge of stored electrostatic energy may occur once a discharge path is established.

2.2 Methods for Electrostatic Control

The main strategies to mitigate electrostatic charges in industrial and laboratory environments are the following:

Electrostatic control in industrial and laboratory environments relies on a set of methods, tools, and equipment developed to reduce or eliminate the chance of an ESD event occurring. These methods represent the solutions for controlling electrostatic charge in environments where sensitive electronic devices are handled:

- **EPA**: A controlled environment in which all materials, surfaces, flooring, and workbenches are electrostatically dissipative, and where strict handling procedures are applied. The EPA ensures that any electrostatic charges are continuously neutralized, preventing uncontrolled discharges. International standards such as IEC 61340-5-1 and ANSI/ESD S20.20 define the requirements for implementing and maintaining EPA.

- **Ionizers**: Ionization is required when handling insulators or isolated conductors that cannot be directly grounded. Ionizers generate balanced positive and negative air ions, which neutralize charge accumulation on surfaces. This process reduces the potential difference that could lead to electrostatic discharge. Their performance is validated according to IEC 61340-4-7.

- **Clothing and Personal Equipment**: Operators wear electrostatically dissipative wearables, gloves, and wrist straps connected to ground. Footwear combined with conductive flooring provides a continuous discharge path, ensuring that the body potential remains below safe thresholds (typically <100 V, as required by IEC 61340-4-5 and ANSI/ESD S20.20). These measures prevent charge accumulation on the human body, which is one of the major sources of ESD events.

- **Packaging and Transportation**: IC and other sensitive devices are stored and transported in ESD-protective packaging, including conductive or static-dissipative bags, trays, and shielding boxes. Packaging materials are classified and tested according to IEC 61340-5-3 to ensure adequate protection against both direct discharge and electrostatic induction.

However, since IC are more sensitive to electrostatic discharge than other devices, it becomes necessary to include internal protection within IC. Despite a widespread misunderstanding in

the industry regarding how to properly characterize the most frequent types of ESD and those that cause the greatest damage throughout the IC lifecycle, some of these phenomena have been characterized, quantified, and standardized by several organizations.

2.3 Snapback Phenomenon in Clamp-Based ESD Protection Devices

Industrial IC rely on a set of on-chip ESD protection structures, commonly referred to as ESD clamp devices, which are designed to conduct the ESD current and limit the pad voltage to a safe level. Despite several clamping options, the Grounded Gate NMOS (GGNMOS) remains widely used in Complementary Metal-Oxide-Semiconductor (CMOS) processes because of its robustness and reliability. Hence, this work adopts the GGNMOS as the reference device for snapback analysis and modeling.

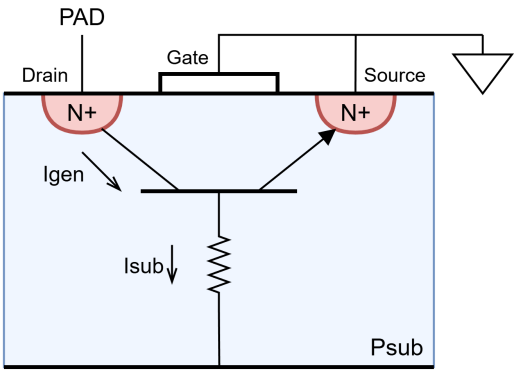


Figure 2.2: GGNMOS cross-section representation

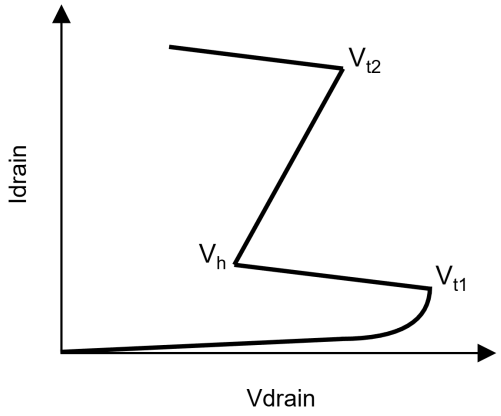


Figure 2.3: I–V curve showing snapback of GGNMOS

To explain the GGNMOS operation during an ESD event, it is useful to view the NMOS as a structure that includes an inherent parasitic bipolar transistor NPN (with drain as collector, substrate as base, and source as emitter), Figure 2.2 illustrates a GGNMOS with its respective parasitic NPN. As the pad voltage rises, the drain–substrate junction becomes reverse biased until avalanche breakdown occurs. Avalanche generates electron–hole pairs: electrons are collected at the drain, while holes drift through the substrate toward the substrate contact, forming a substrate current I_{sub}. This substrate current raises the substrate (base) potential, which forward-biases the base–emitter junction of the parasitic bipolar transistor. When this junction reaches approximately 0.7 V, the parasitic bipolar turns on, and the corresponding drain voltage is defined as the trigger voltage V_{t1}. After the parasitic bipolar turns on, the required drain voltage to sustain the current decreases, causing the device voltage to drop toward the holding level V_h, which is observed as snapback in Figure 2.3. If the stress continues and the current keeps increasing, the device eventually reaches a second breakdown condition associated with thermal failure, characterized by the second breakdown voltage and current, V_{t2} and I_{t2}, meaning that the GGNMOS is permanently damaged [7, pp. 53–54].

Several key parameters are commonly used to describe and evaluate the snapback performance of GGNMOS clamp devices. The trigger voltage (V_{t1}) defines the start of snapback and must be higher than the normal operating voltage of the protected circuit, while remaining sufficiently low to ensure fast activation during an ESD event. The holding voltage (V_h) determines the stability of the conductive state and must be carefully controlled to prevent latch-up during normal operation. In addition, the dynamic on-resistance (R_{on}) governs the voltage rise under high-current stress and directly impacts the clamping effectiveness of the protection device.

Although the present work focuses on the GGNMOS as a representative snapback clamp device, other on-chip ESD protection structures are widely employed in industrial practice. Transient Voltage Suppressor (TVS) diode clamps, for instance, provide voltage clamping without snapback behavior and are commonly used. Additionally, structures such as the Silicon Controlled Rectifier (SCR) also exhibit snapback, typically characterized by lower trigger and holding voltages compared to GGNMOS devices. While these alternatives offer distinct advantages and trade-offs, their detailed analysis is beyond the scope of this work. The focus on GGNMOS allows a clear and consistent investigation of snapback physics and modeling methodology, which can be extended to other clamp devices.

2.4 Summary

This chapter started by reviewing the main mechanisms behind electrostatic charging and discharging. Using practical examples, it was shown that environmental conditions, especially relative humidity, influence the electrostatic charge levels that can be generated.

The triboelectric effect was introduced as a common source of electrostatic charging during handling. The triboelectric series was used to illustrate that different materials have different tendencies to charge, which helps explain why electrostatic events can occur throughout the IC life cycle, from assembly and testing to field operation, leading to either catastrophic or latent failures.

Electrostatic induction was subsequently presented in the chapter as a second charging mechanism that operates without direct contact. It was shown that a nearby charged object can redistribute charges in a conductor; if a discharge path is temporarily available, the object may retain a net charge after the path is removed. This is particularly relevant for IC, where conductive structures and package leads can accumulate induced charge and later discharge when a conduction path is established.

After introducing the charging mechanisms, the chapter summarized the main methods used for electrostatic control in industrial and laboratory environments, including EPA, ionizers, personal grounding equipment, and ESD-safe packaging. These measures reduce the probability of ESD events during handling, assembly, and logistics, but they cannot fully eliminate risk, which motivates the need for on-chip protection.

Finally, the snapback phenomenon was introduced in the context of clamp-based ESD protection, with emphasis on the GGNMOS – adopted as the reference device in this work. The

GGNMOS operation is described as a sequence that starts with avalanche at the drain–substrate junction, followed by substrate current generation and parasitic bipolar activation, leading to the snapback transition from the trigger voltage V_{t1} to the holding voltage V_h. The key parameters used later in this thesis—V_{t1}, V_h, the dynamic on-resistance R_{on}, and the second breakdown point (V_{t2}, I_{t2})—were also established to define the terminology used in the following chapters.

Chapter 3

Test Models for Characterization and Qualification

ESD phenomena, described in chapter 2, highlight how charge build-up and sudden discharge can cause severe damage to integrated circuits. To address these risks, the semiconductor industry has established controlled test procedures that reproduce typical ESD events and define robustness requirements. Toward the design of circuits capable of preventing such failures, international organizations such as ESD Association (ESDA), the Joint Electron Device Engineering Council (JEDEC), the American National Standards Institute (ANSI), and the International Electrotechnical Commission (IEC) developed standardized methods.

These standards are now widely consolidated in industry practices, particularly for the qualification of ESD protection devices. Nevertheless, no single test covers all possible stress scenarios. For this reason, additional models have been proposed and analyzed in order to complement standardized approaches, especially for characterization purposes.

3.1 Standardized Tests

3.1.1 Human Body Model (HBM) JS-001 [8]

HBM simulates the discharge that occurs when a charged person touches one of the terminals of an integrated circuit. In this scenario, the accumulated electrostatic energy is suddenly released into the device under test. According to the JS-001 standard [8], the equivalent circuit is represented by a 100 pF capacitor charged to the desired test voltage and discharged through a 1.5 kΩ resistor. This configuration emulates the resistance and capacitance of the human body.

The resulting current waveform is characterized by a rise time on the order of 2–10 ns, a peak current of approximately 1.5 A at 2 kV, and a decay time of about 150 ns. Actual values may vary depending on the test voltage and according to the person's physical characteristics, but these parameters are consistent with the definition provided by the standard JS-001 [8]. Figure 3.2 shows a typical waveform obtained in an HBM event.

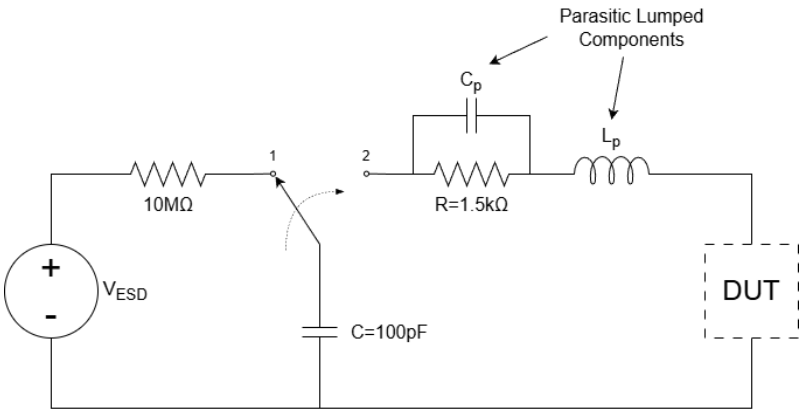


Figure 3.1: HBM ESD equivalent circuit.

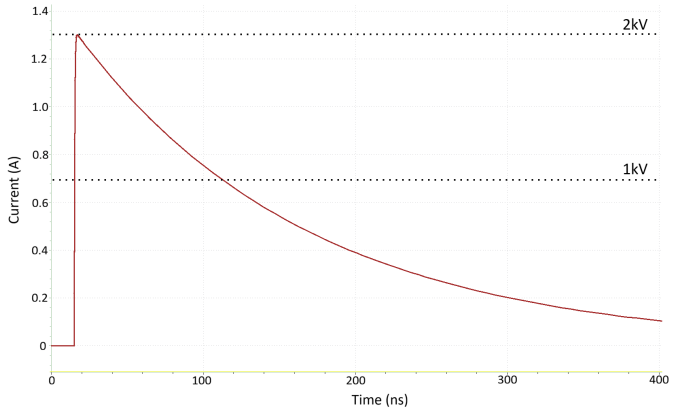


Figure 3.2: Results of HBM test.

The JS-001 standard defines not only the test procedure but also the qualification levels [8], expressed in kilo volts, and their corresponding peak current levels. For this reason, HBM remains the most widely adopted model for device-level ESD qualification and serves as the primary reference for robustness against human discharge events.

3.1.2 Charged Device Model (CDM) JS-002 [9]

The second testing standard developed is the CDM. Because of its high sensitivity to parasitic impedance, shown in Figure 3.3, it is the most difficult test to recreate. This model aims to cover an ESD event that occurs during charging from packaging, transportation, and assembly.

During those stages, it was proven that many IC, which can be susceptible to damage to ESD event, accumulate charges, and when coming into contact with a grounded surface with high conductivity, generate an ESD. Compared to the HBM, this ESD has a short time interval but a high current peak.

Since the CDM tests are sensitive to parasitic inductances and other interference, different test models and simulations were discussed. This leads to different model proposals that attempt to

simulate CDM events with the greatest accuracy. Some CDM examples were presented by organizations and others provided models for Simulation Program with Integrated Circuit Emphasis (SPICE) [10]. Figure 3.4 illustrates a real test and explains the inherent parasitic capacitance.

The test is carried out as follows: the chip is charged via Field Induced Charged Device Model (FICDM). Once charged, the chip is disconnected from ground and a Pogo pin is touched to the pin to be tested. This grounds the Pogo pin, which causes the chip to discharge abruptly. The standard Pogo pin resistance – pin to ground – for the test is equal or lower than 1Ω. These parameters were obtained from the JS-002 CDM test standard [9]; other test parameters, such as relative humidity, are also available.

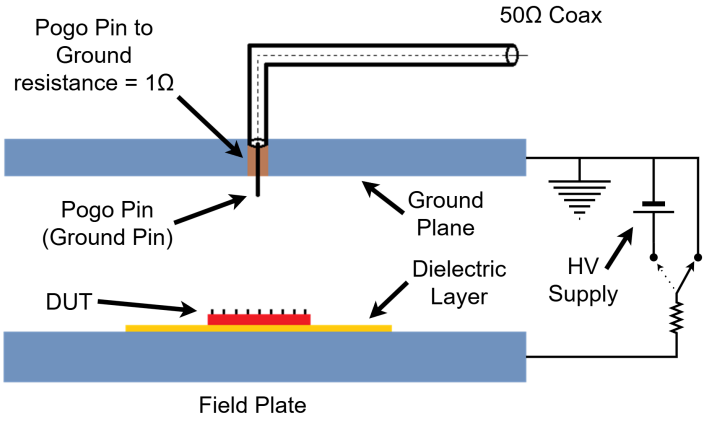


Figure 3.3: FICDM test setup according to JS-002 [9].

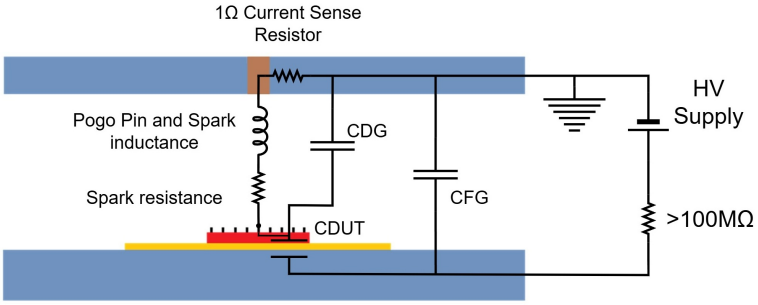


Figure 3.4: Capacitance representations in FICDM as defined in JS-002 [9].

Figure Figure 3.5 presents a SPICE simulation comparing the energy absorbed by the same Device Under Test (DUT) under CDM and HBM stress. Although the CDM waveform has a decay time roughly 50× shorter, it also exhibits a current peak that is about 10× higher than the peak observed in the HBM test.

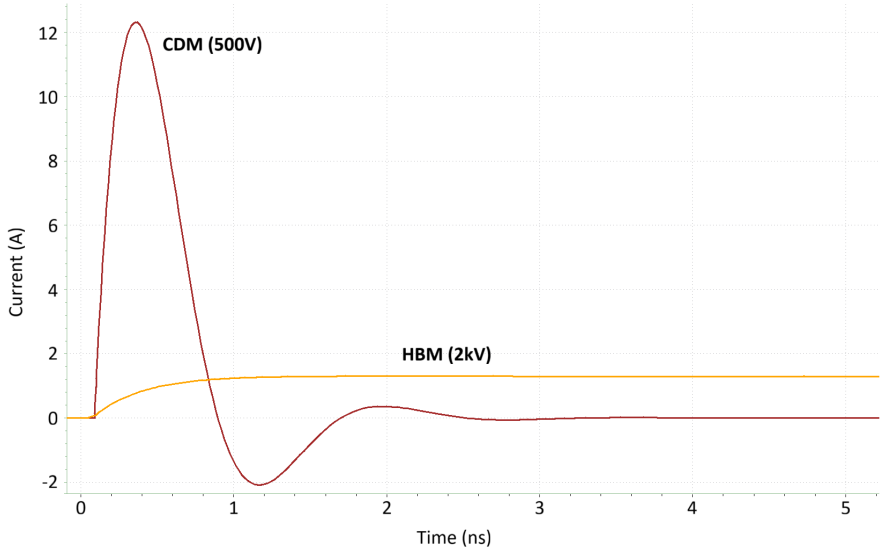


Figure 3.5: Results of CDM test compared with HBM.

For this reason, CDM becomes particularly critical for modern technologies, as the extremely fast rise time combined with a very high peak current can easily trigger localized damage before the device's intrinsic protection structures can respond.

3.1.3 IEC 61000-4-2

Most devices available in the semiconductor industry include protection against HBM and CDM; however, this protection is designed for the manufacturing and assembly process, which already follows several procedures to avoid, control, and minimize exposure to ESD events.

When considering the user handling the IC, it is assumed that they will not always be in a position to operate the device inside an EPA, following all protection protocols defined by the standards to ensure compliance with ESD safety requirements.

Therefore, the IEC systematized a model to simulate ESD in real user scenarios. The discharged into the device resembles a superposition of CDM and HBM, as presented in Figure 3.7, but with a much higher energy, precisely to cover real-world situations without proper protection and thereby ensure the device's reliability.

Two branches depict the test circuit standardized by IEC 61000-4-2, which is also known as the Human Metal Model (HMM). This circuit is designed to simulate a user manipulating the device with metallic tools (e.g., precision tweezers) without appropriate ESD protection. The charge that accumulates on the human body is represented by the branch containing C_1, while the charge that is stored on the metallic tool is represented by C_2. As illustrated in Figure 3.6.

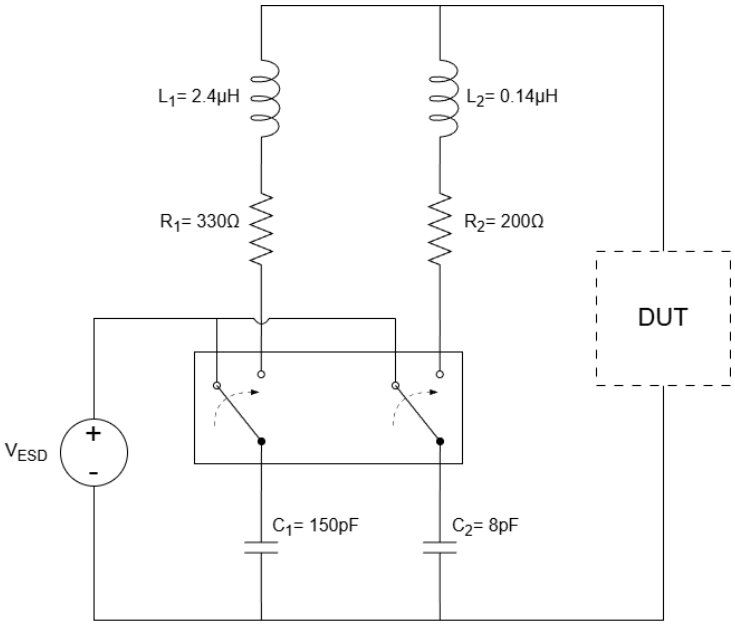


Figure 3.6: Simulation circuit of IEC61000-4-2

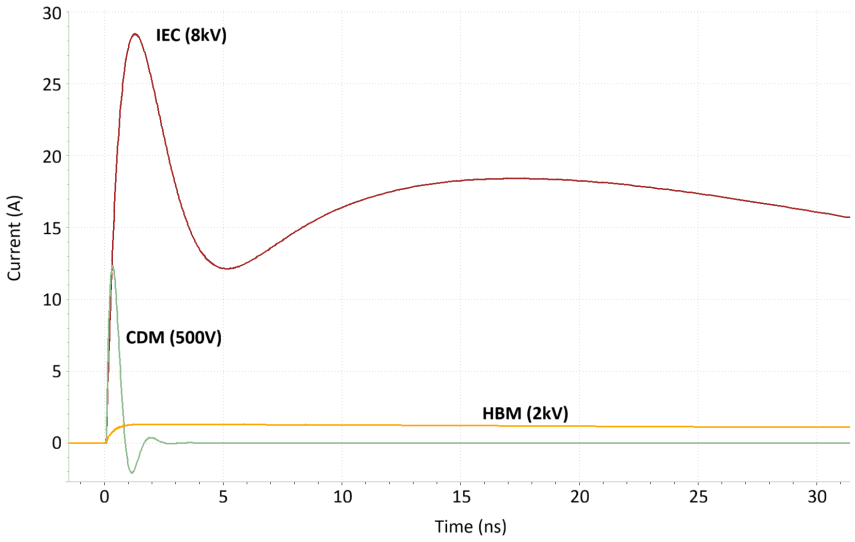


Figure 3.7: HMM compared with CDM and HBM

3.2 Not Standardized Tests

Several other ESD models have been proposed over time, such as Machine Model (MM), Cable Discharge Event (CDE), Charged Board Model (CBM), Socket Discharge Model (SDM), and Charged Strip Model (CSM). Among these, MM has been officially discontinued as a qualification requirement, on the other hand CBM remains relevant for understanding failure mechanisms at the Printed Circuit Board (PCB) level. The following subsections discuss both models in detail.

3.2.1 Machine Model MM

MM was historically introduced as a low-voltage variant of HBM and became a standard in the 1980s. It was presumed to represent discharges from automated equipment or test sockets. The circuit is typically modeled with a 200 pF capacitor discharged through a nearly zero-ohm resistance, leading to current waveforms with rise times of 2–10 ns and peak currents comparable to HBM.

However, extensive industry investigations have indicated that the MM does not introduce failure mechanisms distinct from those already addressed by the HBM and the CDM. Moreover, MM testing is highly sensitive to tester parasitics, leading to increased variability and poor repeatability, while exhibiting no demonstrated correlation with field-return failure data. Based on these findings, major standards organizations, including JEDEC and ESDA, have formally discontinued MM as a qualification requirement. As summarized in JEP172A, ensuring sufficient reliability against HBM and CDM is considered adequate to cover MM conditions, making MM qualification technically redundant rather than additive.

3.2.2 Charged Board Model (CBM)

CBM is a board-level ESD event in which a charged PCB discharges through one of the mounted components; hence, the PCB, rather than the IC package, acts as the charged body. While the discharge mechanism is similar to CDM, the peak currents are significantly higher due to the larger capacitance of the PCB [7, pp. 31-32]. This characteristic distinguishes CBM from standardized ESD models and can provide a complementary test feature that contributes to board-level reliability analysis at the assembly level.

Although CBM has never been formalized as an industry qualification test, it has been consistently reported in real production environments as a cause of field failures [11]. The destructive nature of CBM stems from several factors:

- By the time HBM failures were well mitigated through EPA, charging of entire boards remained largely uncontrolled.
- Failures due to CBM are frequently catastrophic, leaving permanent shorts or opens, and are often interpreted as EOS [11].
- The continuous reduction in IC dimensions makes components less likely to be touched by human operators and more often handled by automated machinery.
- Qualification against HBM or CDM does not guarantee robustness against CBM events.

To reproduce CBM events experimentally, Field Induced Charged Board Model (FICBM) methods are commonly used, where the PCB is charged on a field plate and then discharged in a sequence, similar to CDM test procedures. Although CBM is not formally standardized, CBM testing remains useful for qualifying board-level vulnerabilities and for distinguishing EOS-like failures that may not be captured by standardized tests.

3.3 Characterization tests

The main tests and models currently applied to qualify semiconductor devices are JS-001 [8], JS-002 [9], and IEC-61000-4-2, as described in subsection 3.1.1, subsection 3.1.2, and subsection 3.1.3. These tests are primarily intended to **qualify** a device by verifying whether it can withstand predefined ESD stress levels. In this context, qualification is typically based on pass/fail criteria and often involves destructive stress conditions, which restricts the capacity to observe the protection device behavior during the ESD transient. As a result, while qualification tests indicate whether a given robustness level is met, they provide limited information about the underlying conduction mechanisms, current distribution, or triggering phenomena. Therefore, complementary test methodologies are required to **characterize** the device and to analyze its dynamic behavior during an ESD event.

3.3.1 Transmission Line Pulse (TLP)

TLP testing was originally introduced by Maloney and Khurana in the mid-1980s — and formally standardized in ANSI/ESD STM5.5.1 — as a characterization technique focused on analyzing device behavior under ESD stress. The primary objective of TLP testing is to provide a simple, repeatable, and controlled method to emulate ESD stress conditions while enabling the direct extraction of the electrical response of IC protection structures.

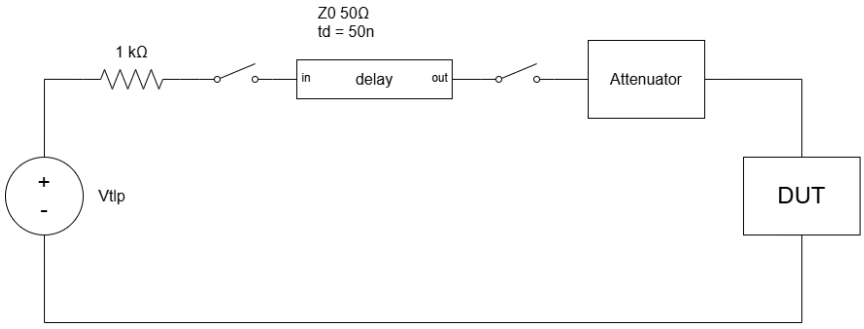


Figure 3.8: Representation of a TLP circuit, adapted from [12, p. 269]

The operating principle of a TLP system is illustrated in Figure 3.8. In this setup, a transmission line is charged using a high-voltage DC source. Once fully charged, a switch is triggered, allowing the stored energy to be discharged into the DUT in the form of a rectangular pulse. Voltage and current at the DUT terminals are monitored using probes, while an attenuator absorbs reflected signals, reducing the impact of multiple reflections and improving measurement stability.

The resulting voltage and current waveforms at the DUT are shown in Figure 3.9. After the initial rise time, the pulse reaches a quasi-static region in which the electrical conditions remain approximately constant. This time window — located between 70% and 90% of the total pulse duration — is used to extract representative voltage and current values. By progressively increasing the charging voltage of the transmission line and repeating the measurement, a set of discrete

voltage–current points is obtained, allowing the construction of a pulsed I-V characteristic of the ESD protection device.

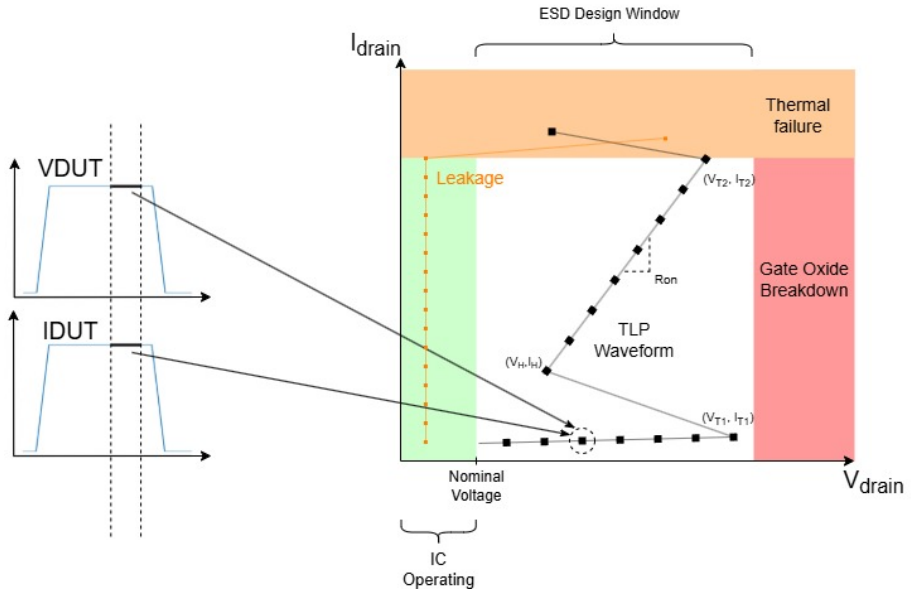


Figure 3.9: Example curve of an ESD protection circuit characterized by TLP [13, p. 48]

The extracted TLP I-V curves provide a more detailed understanding of the behavior of the protection device during an ESD event. By analyzing these curves, it becomes possible to extract key parameters that are essential for achieving a SEED that operates within the device Safe Operating Area (SOA), i.e., considering thermal failure, IC breakdown, and latch-up. Depending on the device architecture and applied stress level, TLP testing may reveal catastrophic failures such as metal melting, junction rupture, or oxide breakdown. Post-pulse Direct Current (DC) leakage current measurements are commonly used to confirm catastrophic failure of the DUT, while latent defects typically require additional functional or parametric testing to determine whether the operating characteristics of the IC have been altered.

The duration of the applied TLP pulse is determined by the physical length of the transmission line. The pulse width T_{pulse} is given by

$$T_{\text{pulse}} = \frac{2L}{v}, \tag{3.1}$$

where L is the transmission line length and v is the signal propagation velocity, which can be approximated by

$$v = \frac{2c}{3}. \tag{3.2}$$

- T_{pulse} is the pulse width (s);
- L is the transmission line length (m);
- v is the signal propagation velocity (m/s);

- c is the speed of light.

TLP testing is widely used in SEED because it relies on a simple test setup and provides clear details relative to how protection devices respond to ESD stress. Its importance is mainly linked to its strong correlation with the HBM test, since both tests drive transient current through the device over similar time scales, even though the applied voltage levels and test purposes are different. Under these conditions, TLP allows key device parameters to be extracted in a non-destructive process, which makes it a practical and effective method to support SEED as an equivalent or substitutive method of HBM [14, pp. 5–7].

3.3.2 Very Fast Transmission Line Pulse (VF-TLP) and Capacitively Coupled Transmission Line Pulse (CC-TLP)

VF-TLP testing was developed intending to provide a characterization method that is closer, in time scale, to CDM events. While conventional TLP is equivalent for HBM test, its rise time and pulse width are longer than what is traditionally observed in CDM, as shown in Figure 3.5. In practice, VF-TLP applies a very short pulse (typically 1–5 ns wide) with an ultra-fast rise time (around 0.1 ns), allowing the evaluation of device behavior under CDM stress conditions. [14, p. 8][2].

Although VF-TLP targets the CDM time domain, a possible miscorrelation can appear because the usual VF-TLP setup is a two-pin measurement, whereas CDM is a one-pin discharge. In CDM, the device is charged as a whole and then discharges through a single pin to the reference plane, so the dominant current path may be strongly affected by capacitive coupling and package-related parasitics, as shown previously in Figure 3.4. Otherwise, a two-terminal VF-TLP pulse tends to enforce a more explicit terminal-to-terminal current path, which can diverge from the true current path from the FICDM test [13, p. 57].

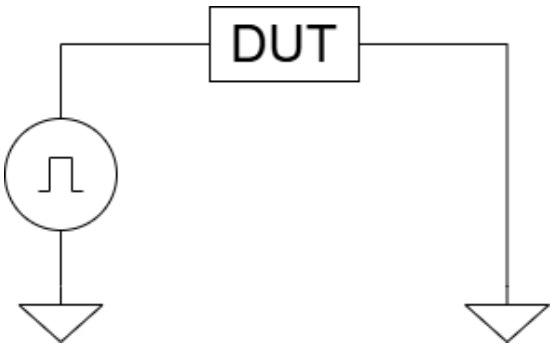


Figure 3.10: VF-TLP test setup.

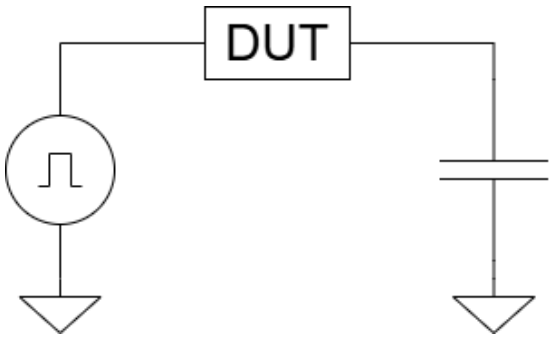


Figure 3.11: CC-TLP test setup.

Figure 3.12: Comparison of fast-pulse characterization methods: VF-TLP and CC-TLP.

That difference inspired the development of CC-TLP. In CC-TLP, the pulse is applied through a coupling capacitor, avoiding a strictly two-terminal current path. This shifts the excitation toward displacement current and parasitic coupling, which can be closer to the single-pin behavior of

FICDM [15]. As a result, this enables a more reproducible CDM stress, where the DUT is charged and discharged mainly through its coupling capacitances rather than a forced pin-to-pin path.

From the perspective of standardization, VF-TLP is documented in STM5.5.1-2022 as a standard test method, which defines waveform and measurement conditions. CC-TLP is documented as a standard practice in SP5.3.4-2022, reflecting a common and useful methodology. Despite these differences, VF-TLP and CC-TLP are often used as complementary tools: VF-TLP provides controlled fast-pulse device curves, while CC-TLP can improve alignment with the single-pin nature of CDM-oriented evaluation.

3.4 Summary

In this chapter, it was explained why ESD qualification is based on standardized tests that, in theory, replicate the most typical discharge situations that an IC might encounter through its lifecycle. In this sense, ESDA, JEDEC, and IEC establish guided procedures that emulate ESD conditions during handling, assembly, and end-use operation, enabling the evaluation of device robustness.

The standardized qualification methods JS-001 (HBM) [8], JS-002 (CDM) [9], and IEC 61000-4-2 (HMM) were then presented. HBM represents an ESD from a charged human body to a device during typical handling operations and remains the most common device-level qualification reference. CDM is described as a fast, high-peak-current discharge that is highly sensitive to parasitic impedance and therefore more difficult to simulate, while IEC 61000-4-2 was introduced as a user-oriented model designed to cover operation conditions, with an energy magnitude equivalent to a stronger superposition of HBM and CDM.

Finally, the chapter addressed additional, non-standardized models (MM and CBM) to complement the standardized scenario. MM was described as discontinued because it showed a strong correlation with HBM and therefore provided limited additional qualification value, while also being sensitive to tester parasitics and repeatability.

Based on these limitations, a clear distinction was made between qualification and characterization tests. TLP techniques (TLP, VF-TLP, and CC-TLP) were presented as practical tools to extract pulsed I–V data and to support SEED within SOA constraints. It was also noted that, in many design and development flows, these characterization methods are used as substitute evaluations for HBM and CDM, since they provide valuable information regarding the protection device behavior throughout the different phases of an ESD transient and are commonly correlated with standardized qualification tests.

Chapter 4

Technology Computer-Aided Design (TCAD) Tool for SEED

The design and analysis of ESD protection devices require an accurate description of physical mechanisms that occur under extreme operating conditions, such as high electric fields, large current densities, and strong impact ionization. However, the PDK provided by the foundry are primarily optimized for core circuit simulation and do not include snapback behavior or parasitic bipolar conduction embedded in some models. As a result, compact device models available in standard SPICE environments are insufficient to describe the physical origin of triggering, current distribution, and failure mechanisms in snapback-based ESD devices.

To overcome these limitations, TCAD tools are employed as a complementary methodology to circuit-level simulation. TCAD provides a physics-based framework capable of resolving the spatial distribution of electrical and material properties inside semiconductor devices. This capability enables direct investigation of avalanche breakdown, substrate current generation, and parasitic bipolar activation, which are essential for understanding snapback operation and for extracting physically significant parameters later used in SPICE macromodels.

TCAD is a powerful tool during the early stages of ESD protection device development, especially in the first design cycle before tape-out, when limited experimental data and design experience are available. At this stage, device dimensions and layout strategies are defined without full knowledge of their behavior under ESD stress, and TCAD provides the insight needed to make more informed design decisions, reducing uncertainty in the initial choices. After tape-out and tests under ESD conditions, practical experience and measured results allow further refinement of compact models, and the reliance on TCAD naturally decreases. In this sense, TCAD does not replace test validation, but significantly increases the probability of first-cycle success by avoiding purely randomized design iterations.

4.1 Extracting parameters

Since manufacturing processes are unknown and certainly vary between foundries, even if the minimum dimension is the same, it is necessary to obtain as much information as possible to replicate the devices in TCAD. To this end, some parameters may be useful for replicating the device in TCAD. In this case, the device was modeled in BSIM4 and the following parameters were used to obtain information that may be related to the device's cross-section information. They are:

- Xj, Source/Drain junction depth. (m)
- Nsd channel doping concentration at depletion edge for zero body bias (cm^{-3})
- Nsub Substrate doping concentration (cm^{-3})
- TOXE Electrical gate equivalent oxide thickness (m)

Other relevant parameters can be extracted from the device. E.g. the threshold voltage ($V_{\mathbf{th}}$), leakage in normal operation, and geometric parameters of the device, such as DCGS and contact area [7, p. 100]. After obtaining the maximum available information, the design of the devices in 2D or 3D begins in TCAD.

4.2 Design and mesh

To simulate and obtain the device's behavior, it is necessary to create the device structure following the previously extracted parameters. A structure was then created for GGNMOS experiments in 2D and 3D, as illustrated in Figure 4.3. The tool used to create the device was the Sentaurus Structure Editor (SDE), and the dimensions of each block were parameterized using a script based on Tool Command Language (TCL). This simplifies the testing process, improving Electronic Design Automation (EDA), since certain parameters of the structure can be calibrated and changed iteratively, modifying only the variable in Workbench without having to refactor the structure each time a new dimension needs to be tested. After completing the design and adding the doping levels and depths, the most important part begins: mesh preparation.

The mesh strategy followed the same parameterized approach adopted for the device structure. Since the structure was defined through explicit geometrical and doping parameters, the meshing strategy was organized to preserve comparability across iterations rather than to generate a uniformly dense grid, Figure 4.6 illustrates these strategies. In practice, the mesh was defined as non-uniform. A coarse discretization was used in regions far from the active device, while local refinement was applied only where the physical quantities are expected to change quickly. In particular, a finer mesh was used at the source and drain junctions, where abrupt doping transitions occur; in the channel region beneath the gate, where strong electric fields are expected; and at the silicon–oxide interfaces, where material properties change abruptly due to the different electrical

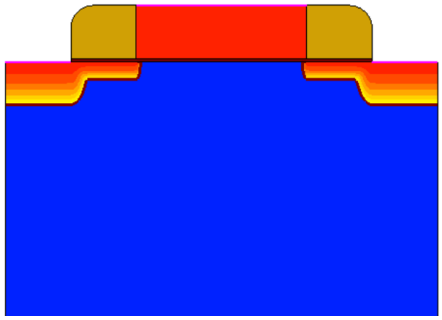


Figure 4.1: 2D design

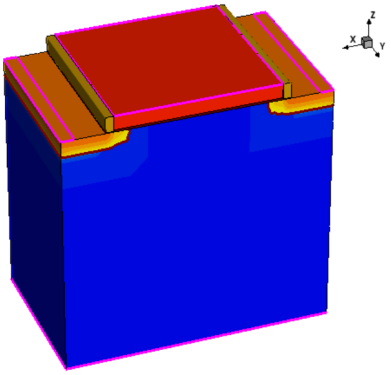


Figure 4.2: 3D design

Figure 4.3: Representation of two GGNMOS devices with different sizes.

response of silicon and oxide to an applied electric field.The resulting mesh is generated using a Delaunay triangulation, producing triangular elements that follow the local refinement constraints.

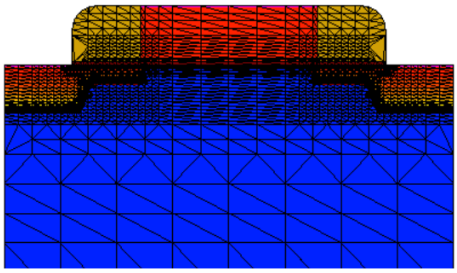


Figure 4.4: GGNMOS mesh refinement regions

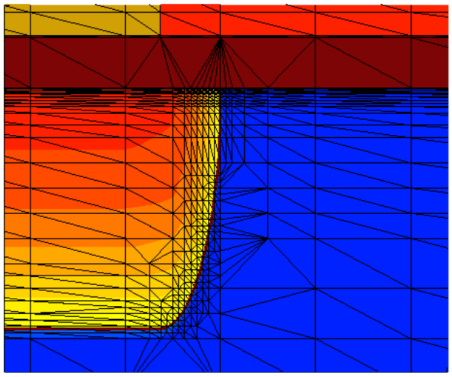


Figure 4.5: Refinement near the drain region

Figure 4.6: TCAD mesh of the GGNMOS: full device view and zoomed detail of the refined region.

Quantities such as electrostatic potential and electric field are computed at the mesh vertices and propagated through the elements. If critical regions are insufficiently detailed, the simulator might fail to correctly represent variations of electric field and potential, causing regions where current concentrates over small areas of the device to be incorrectly identified, which increases the probability of convergence problems. Otherwise, an unnecessarily dense mesh increases computational cost. Based on preliminary tests, a reference binary tree size — constructed from the mesh vertices — of approximately 10 000 nodes in 2D and 30 000 nodes in 3D was adopted. Once the device is discretized into a mesh, Poisson's equation can be solved.

4.3 Poisson's equation

In TCAD, the device must first be defined in terms of its structure—specifying materials and regions—before the electrostatic and transport equations can be solved. Under these circumstances, the mesh is the numerical object that allows the simulator to solve the differential equations that define the device behavior.

4.3.1 Poisson equation

The fundamental equation in semiconductor device simulation is Poisson's equation, which relates the electrostatic potential to the local charge distribution. In a general form, it can be written as

$$\nabla \cdot (\varepsilon \nabla \phi) = -\rho, \tag{4.1}$$

where:

- ϕ is the electrostatic potential (V or J/C);
- ε is the permittivity (F/m);
- ρ is the local charge density (C/m^3).

To help explain: if one assumes a one-dimensional case with constant permittivity, the equation can be simplified to a second-order derivative form,

$$\frac{\mathrm{d}^2\phi(x)}{\mathrm{d}x^2} = -\frac{\rho(x)}{\varepsilon}. \tag{4.2}$$

This one-dimensional formulation allows a direct interpretation of how charge and boundary conditions shape the electrostatic potential. Consider a domain $x \in [0, L]$ with constant permittivity and fixed boundary conditions.

If no charge is present inside the domain, $\rho(x) = 0$, Poisson's equation reduces to

$$\frac{\mathrm{d}^2\phi(x)}{\mathrm{d}x^2} = 0, \tag{4.3}$$

which implies a constant electric field and, consequently, a linear potential variation between the two boundaries.

When charge is present inside the domain, $\rho(x) \neq 0$, the second derivative of the potential becomes nonzero in the charged region. In this case, the electric field is no longer constant, and the potential profile deviates from a straight line. The spatial distribution and sign of $\rho(x)$ directly determine where and how the potential curvature appears.

Hence, Poisson's equation can be understood as a mapping between a given charge distribution and boundary conditions, and the resulting spatial profile of the electrostatic potential.

For this reason, the quality of the mesh is a key factor in the solution and convergence of Poisson's equation. Since spatial derivatives are approximated using differences between neighboring

nodes, the local mesh spacing Δx determines how accurately variations in charge density, electric field, and electrostatic potential are resolved. A coarse mesh may fail to capture localized regions of high charge density or strong field gradients, while an excessively fine mesh increases the size of the algebraic system and the computational cost, even in regions where the potential is constant. Hence, an effective mesh strategy must balance accuracy and efficiency by refining the mesh only where strong spatial variations are expected.

It is important to note that Poisson's equation alone does not uniquely determine the electrostatic potential. As a linear, second-order, and non-homogeneous partial differential equation, its solution depends on a consistent set of boundary conditions. In TCAD simulations, such conditions are applied directly at the mesh nodes located on device contacts and material interfaces, thereby impacting the solution of Poisson's equation. For a well-defined physical domain and a consistent set of boundary conditions, the resulting solution is unique, ensuring that a given structure and charge distribution correspond to a single solution.

4.4 Mixed-Mode Simulation

Once the device structure is discretized and the relevant physical equations are defined, the simulation is carried out using Sentaurus Device (SDEVICE). At this stage, SDEVICE serves as the simulation environment in which the internal device physics and the external circuit response are solved together as a coupled problem. In comparison with SPICE tools, which rely on compact models with predefined functional forms for the device behavior, SDEVICE determines the electrical response directly from the spatial solution of the fundamental physical equations.

In this work, the simulations were performed using a mixed-mode configuration, in which the semiconductor device is excited by an ideal pulsed voltage source applied through an external impedance. The device response is not imposed directly at the terminals; instead, terminal voltages and currents result from the interaction between the applied pulse, the external impedance, and the internal device physics. This circuit reproduces the essential elements of a TLP test setup, enabling the evaluation of device behavior under ESD operating conditions within a physics-based environment. This configuration was selected to maintain a low computational cost while remaining portable to simulations based on SPICE.

The main advantage of the `Mixed-Mode` approach is the ability to simulate a complete system with multiple devices, rather than an isolated component. This allows the development of ESD protection devices whose triggering behavior is influenced by dedicated driving networks during transient operation. Rather than imposing fixed terminal conditions, the devices are embedded within a netlist-based circuit, similar to SPICE tools, in which terminal voltages and currents evolve dynamically, enabling the device to operate in nonlinear regimes such as avalanche breakdown and snapback. As a result, the simulated behavior follows the transient response and current sharing observed in system-level ESD experiments, including TLP measurements.

The physical models enabled in the `Physics` section and their relevance to the snapback I–V characteristic are summarized below:

- **EffectiveIntrinsicDensity (OldSlotboom):** Adjusts carrier concentration to account for high doping levels. This is critical for **BJT** to optimally predict current gain (β), and for **MOSFET** to accurately model behavior in heavily doped regions like the source and drain.
- **Mobility models (DopingDependence, HighFieldSaturation, Enormal):** These models account for carrier mobility degradation caused by impurity scattering, high-velocity saturation, and surface scattering at the oxide interface. Understanding and modeling these reductions is important for determining the electric-field distribution near the drain, which directly affects the impact ionization rates, the beginning of avalanche breakdown, and the snapback trigger voltage (V_{t1}).
- **Avalanche (Okuto):** This model simulates impact ionization, the main source of snapback triggering. Under high temperatures and strong electric fields, the `Okuto` was chosen because it provides a physically consistent description of impact ionization while requiring less computational time than more advanced avalanche models [13, p. 74]. It correctly calculates the substrate current needed to "turn on" the parasitic bipolar transistor within the NMOS structure with efficient convergence.

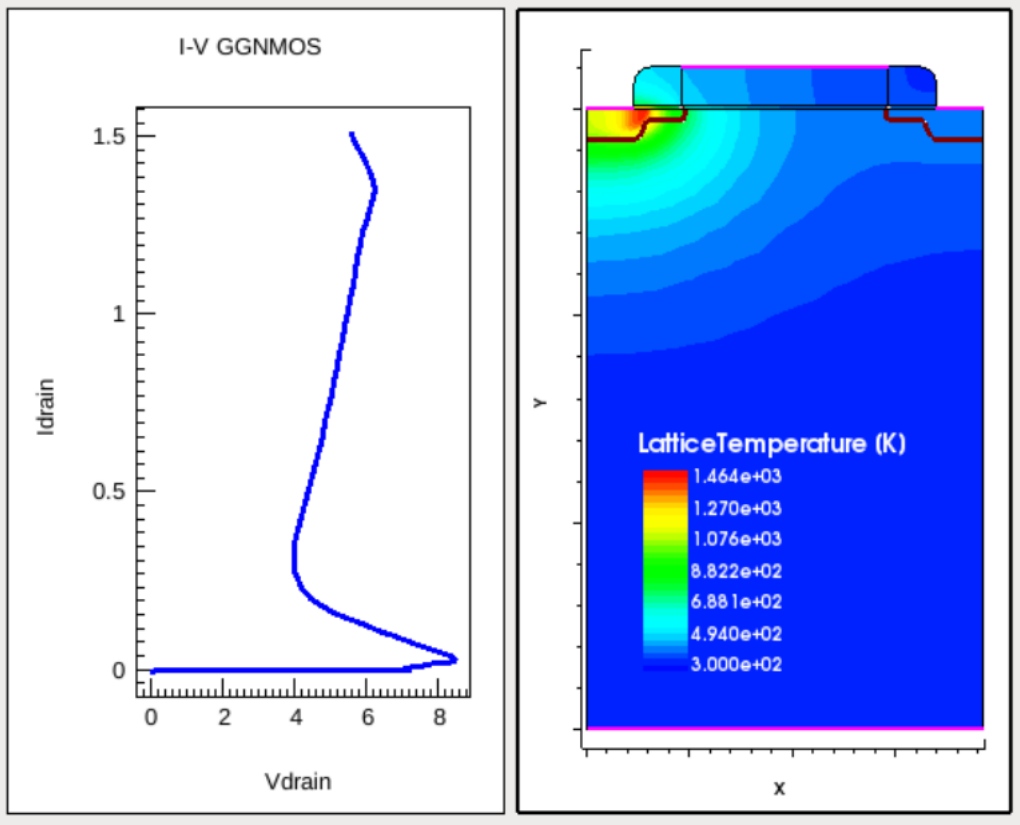


Figure 4.7: Example of a TLP simulation in a GGNMOS with current, voltage, and temperature analysis

- **Recombination mechanisms (SRH and Auger):** `SRH` recombination is the main mechanism at low and moderate injection levels, while Auger recombination becomes important once carrier density increases after snapback. Although `Auger` recombination does not trigger snapback, it helps limit carrier buildup in the post-trigger regime and leads to a more realistic I–V slope, improving numerical stability.

- **Thermodynamic model:** That model enables self-heating by allowing the lattice temperature to evolve during transient operation. Thermal effects influence mobility, recombination, and avalanche rates, making them important for evaluating current paths and device robustness under ESD conditions.

- **AnalyticTEP:** Provides a consistent coupling between electrical transport and thermal effects, improving numerical stability and ensuring that energy flow behaves properly during high-current transients.

- **AreaFactor:** Defines the effective width in 2D simulations by scaling the default 1 μm reference to represent the intended electrical dimension of each component. Its main advantage is allowing different effective widths to be assigned to individual devices within the same circuit, ensuring correct current scaling. This parameter is particularly useful and enables rapid design optimization by adjusting numerical values directly in the command file.

An example of this approach, including the evolution of current, voltage, and lattice temperature during a TLP test, is shown in Figure 4.7. Together, the device construction, mesh strategy, physical models, and mixed-mode configuration establish a simulation tool that allows the analysis of snapback and high-current operation based on a detailed physical device description. This capability is essential for snapback-based protection devices, whose triggering and conduction depend on correlated electrostatic, transport, and thermal phenomena that cannot be explicitly solved in SPICE or obtained directly from standard PDK. For this reason, TCAD is employed in this work as a predictive and experimental tool, allowing for a clearer understanding of ESD behavior and providing a solid physical basis for the parameters extraction and macromodel development presented in the following chapters.

4.5 Summary

This chapter introduced the use of TCAD as a supporting tool for SEED, with particular attention to snapback-based protection devices. During the early stages of development, TCAD makes it possible to explore and validate new protection models at the device level, reducing the need for trial-and-error iterations. Since standard foundry PDK are mainly intended for core design, TCAD was introduced as a way to access a physics-based description of these mechanisms and to support the later development of SPICE models.

The simulation flow adopted in this work was then described. Device parameters extracted from the PDK were used as guidance to build representative 2D and 3D structures. The GGNMOS

geometry was created in SDE using a parameterized TCL script, allowing iterative adjustments of device dimensions with minimal effort. A mesh strategy was applied, with refinement focused on junctions, the channel region, and material interfaces, while a coarser mesh was used in other areas to minimize computational cost. Based on preliminary tests, reference binary tree sizes of approximately 10 000 nodes in 2D and 30 000 nodes in 3D were adopted.

The functioning of the tool was presented, together with a discussion of the importance of mesh and boundary conditions in achieving accurate results. The chapter then described the simulation setup in SDEVICE, emphasizing the mixed-mode approach used to reproduce the TLP test. Additionally, the device was added to the circuit, similar to a SPICE netlist. Furthermore, physical models enabled in the simulations were discussed in terms of their impact on the snapback I–V behavior. The device's behavior under ESD conditions was extracted directly from these simulations, which provide the basis for the parameter extraction and macromodel development discussed in the following chapters.

Chapter 5

SPICE Macromodel of ESD Protection Devices with Snapback

As indicated in section 2.3, GGNMOS is frequently utilized in ESD protection designs due to its ability to conduct high currents after snapback begins. However, this behavior occurs in a regime that is not implemented by conventional MOS compact models included in the PDK for SPICE simulation. As shown in [13, p. 352], the device I–V can be separated into the linear and saturation regions, where standard MOS equations are used, and the avalanche and snapback regions, where the parasitic Bipolar Junction Transistor (BJT) is not included in the model. In this regime, V_{t1} and V_H are strongly influenced by the parasitic lateral NPN action and by the effective substrate resistance, which depends on the device layout [13, Sec. 11.2, pp. 351–354]. Hence, circuit-level ESD analysis requires a model that extends the MOS description to represent these high-current mechanisms while remaining compatible with simulations based on SPICE.

Several GGNMOS macromodels have been proposed to represent the parasitic NPN that are missing from conventional MOS compact models. In [16, Sec. 2–6], the MOS description is extended with additional current components and a substrate resistor R_{sub} to represent avalanche generation and the parasitic bipolar action needed to reproduce snapback in SPICE. In addition, [17, Sec. 1–2] points out that macromodels that try to represent the parasitic BJT through explicit current sources often suffer from convergence issues and make model calibration difficult, since the source parameters cannot be directly related to parameters obtained from TCAD. To avoid this limitation, Zhou proposed a subcircuit composed of a `BSIM` MOS model, a `VBIC` BJT model with built-in collector–base avalanche, and a substrate resistor R_{sub}, avoiding explicit external sources or behavioral implementations [17, Sec. 3].

Based on this methodology, in this chapter, a GGNMOS macromodel is implemented using `BSIM4` plus `VBIC` and calibrated using a reference I–V curve obtained from TCAD, with a script used to automate the fitting process.

5.1 GGNMOS Macromodel for SPICE

To enable the snapback behavior on NMOS, a dedicated macromodel is required to extend the conventional NMOS available in the foundry PDK. In this work, the GGNMOS is represented using a combination of a `BSIM4` NMOS transistor and a parasitic bipolar transistor described by the `VBIC` compact model. This formulation enables the bipolar triggering via avalanche to be reproduced within a standard SPICE simulation flow.

The adopted modeling strategy follows the methodology proposed by Zhou [17], which is based exclusively on standard SPICE elements and built-in compact model formulations, avoiding the use of explicit behavioral current sources. This method improves portability across simulators and provides better convergence compared to approaches that require special implementations or dependent sources. The resulting GGNMOS macromodel schematic is shown in Figure 5.1.

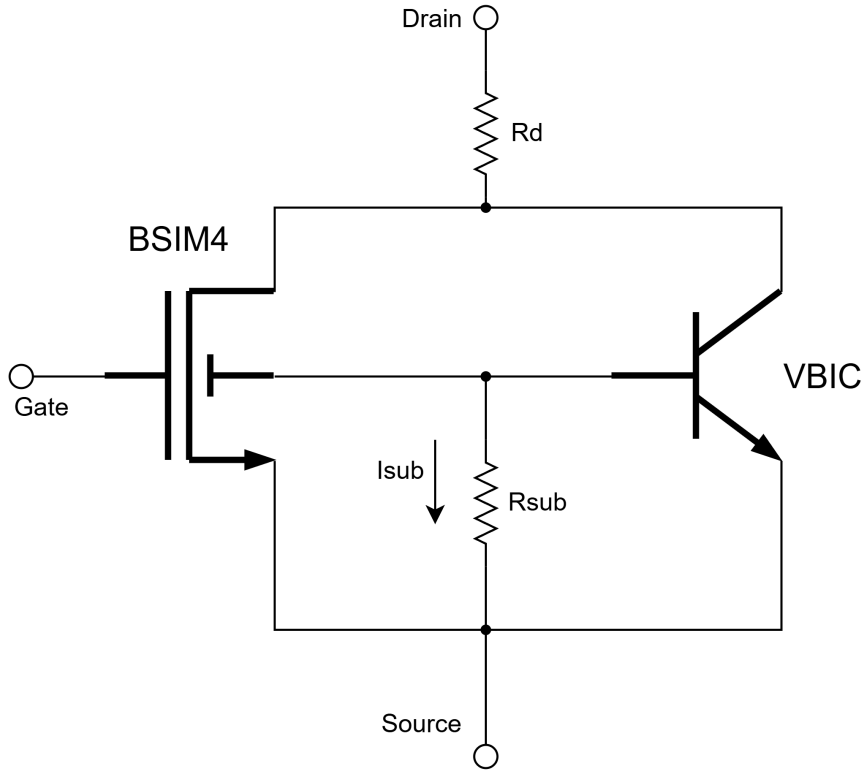


Figure 5.1: GGNMOS macromodel combining a `BSIM4` NMOS with a parasitic lateral NPN modeled using `VBIC`, adapted from [17].

In a GGNMOS device, snapback is initiated by impact ionization in the high-field region near the drain junction. As the drain voltage increases under ESD stress, electron–hole pairs are generated through impact ionization, a process represented in the `BSIM4` compact model by the impact ionization current I_{ii}. The electrons generated by this mechanism are collected by the drain, while the generated holes flow into the substrate, giving rise to the substrate current I_{sub}.

As I_{sub} increases, a voltage drop develops across the substrate resistance R_{sub}, raising the local substrate potential regarding the source channel (V_{BS}). When this voltage becomes sufficient to

forward-bias the source–substrate junction, the parasitic lateral NPN transistor is activated ($V_{BS} > 0.6V$), starting the snapback trigger point V_{t1}. At this point, the conduction path shifts to a lower-resistance channel dominated by the parasitic bipolar transistor, and the voltage required to sustain the current becomes lower than the avalanche voltage that initiated the process (V_H), resulting in snapback in the I–V characteristic.

5.1.1 Impact ionization current in BSIM4 (I_{ii})

In the proposed macromodel, the initiation of snapback is driven by the impact ionization current generated in the high-field drain region of the NMOS transistor. In `BSIM4`, this mechanism is described by the impact ionization current I_{ii}, given by Equation 5.1. This current represents the generation of electron–hole pairs once the device operates higher than saturation and the drain electric field becomes too strong.

$$I_{ii} = \frac{(\alpha_0 + \alpha_1 L_{eff})}{L_{eff}} (V_{ds} - V_{dseff}) \exp\left(\frac{\beta_0}{V_{ds} - V_{deff}}\right) I_{dsNoSCBE} \tag{5.1}$$

The physical meaning of each term in Equation 5.1 is summarized as follows:

- I_{ii} is the impact ionization current generated near the drain junction. In a GGNMOS, this current is the physical origin of the substrate current (I_{sub}) that leads to snapback triggering.
- L_{eff} is the effective channel length, accounting for lateral diffusion and short-channel effects. The scaling factor $(\alpha_0 + \alpha_1 L_{eff})/L_{eff}$ captures the increase in impact ionization efficiency as the effective channel length is reduced.
- α_0, α_1 and β_0 are fitting parameters associated with the impact ionization process. α_0 and β_0 are impact ionization coefficients, while α_1 introduces its dependence on the effective channel length, improving the scalability of I_{ii} with device geometry. These parameters were extracted from PDK and were therefore fixed for the technology used in this work.
- $(V_{ds} - V_{dseff})$ represents the effective voltage drop across the high-field region near the drain. Under ESD conditions, the NMOS operates predominantly in saturation, and V_{dseff} corresponds to the drain–source saturation voltage V_{dsat}.
- $I_{dsNoSCBE}$ is the drain current without the Substrate Current Induced Body Effect (SCBE), representing the drain current that provides carriers to the drain junction for avalanche generation. In the `BSIM4` model, this current already includes other short-channel effects but specifically excludes the positive feedback caused by the substrate current (I_{sub}) on the threshold voltage.

In the `BSIM4` model, the impact ionization current I_{ii} provides the substrate current I_{sub} that raises the substrate potential (V_{sub}) and initiates the turn-on of the parasitic lateral NPN, thus defining the snapback trigger voltage V_{t1}. Once snapback is triggered, additional avalanche and bipolar effects must be represented to correctly describe the sustained conduction regime, which is addressed by the `VBIC` model discussed in the following subsection.

5.1.2 Collector–base avalanche current in VBIC (I_{gc})

In the macromodel, the VBIC transistor is used to represent the parasitic lateral NPN conduction and, most importantly, to provide a collector–base avalanche that is not available in conventional MOS compact models. In [17, Sec. 3], this built-in avalanche formulation is implemented to avoid explicit external current sources, while still enabling an additional avalanche current component in the subcircuit. The avalanche current is represented by I_{gc}, given by Equation 5.2 and was obtained from VBIC's manual.

$$I_{gc} = (I_{tzf} - I_{tzr} - I_{bc})\, A_{VC1}\, V_{gci} \exp\left(-A_{VC2}\, V_{gci}^{(MC-1)}\right) \tag{5.2}$$

The physical meaning of each term in Equation 5.2 is defined as follows:

- I_{gc} is the collector–base weak avalanche current generated in the VBIC parasitic NPN. In the macromodel, this element introduces the avalanche multiplication required to sustain the bipolar conduction after triggering, enabling the post-trigger region of the I–V characteristic to be reproduced.

- $(I_{tzf} - I_{tzr} - I_{bc})$ is an internal VBIC current term that sets the current level available before the avalanche multiplication is applied. This plays the same role as the "collector current without avalanche" term used in the equivalent VBIC avalanche formulation discussed in [17, Sec. 3].

- $V_{\mathbf{gci}}$ is an effective voltage variable that represents how strongly the collector–base junction is reverse biased in the avalanche term. It is written from the intrinsic junction voltage $V_{\mathbf{bci}}$ and the built-in potential P_C (e.g., $V_{\mathbf{gci}} = P_{\mathbf{C}} - V_{\mathbf{bci}}$), so that increasing the collector–base reverse bias increases $V_{\mathbf{gci}}$ and therefore increases the avalanche contribution $I_{\mathbf{gc}}$.

- $AVC1$ and $AVC2$ are model parameters. Together, they scale the magnitude of I_{gc} and set its exponential voltage dependence, providing custom tuning to match the avalanche contribution required for snapback reproduction.

- MC is the junction grading coefficient, shaping how strongly the avalanche contribution depends on the collector–base junction voltage. In practice, it controls the curvature of the avalanche current growth as the junction voltage increases.

- R_{sub} is the effective substrate resistance used in the macromodel. Although it does not appear explicitly in Equation 5.2, it converts the substrate current ($I_{\mathbf{sub}}$) related to avalanche into a substrate voltage rise that biases the parasitic NPN base ($V_{\mathbf{sub}}$), affecting both $V_{\mathbf{t1}}$ and the holding point $V_{\mathbf{h}}$. In [17, Sec. 4], R_{sub} is extracted from measured snapback curves, and minor adjustments of BJT parameters may be required to fit the snapback behavior.

- $R_{\mathbf{d}}$ is the drain series resistance included in the macromodel to represent resistance related to the device's layout. Although it does not appear explicitly in Equation 5.2, $R_{\mathbf{d}}$ increases

the voltage drop for a given current after triggering, affecting the extracted R_{on} and the post-snapback I–V slope.

Within the subcircuit, $I_{\mathbf{gc}}$ provides an avalanche contribution in parallel with the MOS substrate-current path, increasing the number of fitting parameters to reproduce the voltage drop across R_{sub} with good accuracy. For this reason, the calibration of $AVC1$, $AVC2$, MC, R_{sub}, and $R_{\mathbf{d}}$ is central to matching the post-trigger conduction level and the holding condition observed in the reference I–V curve [17, Sec. 4].

Furthermore, the analysis indicated the relevance of the C_{JC} parameter for the macromodel, since the collector–base junction capacitance produces displacement currents during short rise time transitions. In this case, the current can increase the voltage drop across R_{sub}, changing the effective base bias of the parasitic NPN (V_{sub}) and shifting the V_{t1} compared to a quasi-static measurement. The rise time study shows that V_{t1} is sensitive to C_{JC}, these results motivate a calibration approach in which $AVC1$, $AVC2$, MC, R_{sub}, and $R_{\mathbf{d}}$ are first adjusted to match the quasi-static snapback curve, while C_{JC} is used later to refine the transient response.

5.2 Calibration Strategy and Parameter Extraction

The macromodel calibration is made by comparing a reference snapback I–V characteristic obtained from TCAD or TLP tests and a set of candidate I–V curves generated in SPICE by sweeping the macromodel fitting parameters, where each parameter combination is identified by an `ALTER` index. The strategy implemented is based on a script that automates this procedure, since manually calibrating the macromodel parameters through trial-and-error would be time-consuming within a SEED flow, where multiple design iterations and parameters must be tested. The final calibration workflow is organized as follows:

(i) obtaining a reference snapback I–V curve from TCAD (or from TLP measurements);

(ii) generating a set of candidate I–V curves in SPICE by sweeping the macromodel parameters through combinations of predefined values (starting from coarse ranges and step sizes), where each parameter set is identified by an index;

(iii) running the script to compare the reference curve with the candidate curves by extracting snapback points (V_{t1}, I_{t1}, V_H, I_H, V_{t2}, I_{t2}) and ranking all parameter sets using a weighted error metric.

(iv) refining the parameter search by returning to step (ii) only around the best-ranked parameter sets, restricting each parameter to a smaller range centered on its current best value and using finer step sizes to increase resolution. This local refinement reduces the total number of set combinations that must be simulated, lowering computational cost while improving calibration accuracy.

5.2.1 Extraction of snapback points (V_{t1}, V_h, V_{t2})

To compare curves using a compact set of metrics, each I–V dataset is reduced to a set of characteristic snapback points: the trigger point (V_{t1}, I_{t1}), the holding point (V_h, I_h), and an optional second trigger point (V_{t2}, I_{t2}) when a second snapback is present. The extraction follows a temporal scan of the sampled voltage sequence $V[k]$ and current sequence $I[k]$, using local limits:

- A candidate trigger point is detected as the first local maximum in voltage,

$$V[i-1] < V[i] \ \wedge \ V[i] > V[i+1], \tag{5.3}$$

  followed by the first local minimum after that maximum,

$$V[j-1] > V[j] \ \wedge \ V[j] < V[j+1], \qquad j > i. \tag{5.4}$$

- To avoid selecting small oscillations or weak snapback-like features, the pair (i, j) is accepted only if the voltage drop satisfies

$$V[i] - V[j] > \Delta V_{\mathrm{SB,min}}, \tag{5.5}$$

  where $\Delta V_{\mathrm{SB,min}}$ is the minimum voltage drop required to accept a snapback event. If Equation 5.5 is not satisfied, the search continues from the next local maximum.

- Once Equation 5.5 is satisfied, the snapback trigger and holding points are defined as

$$(V_{t1}, I_{t1}) = (V[i], I[i]), \qquad (V_H, I_H) = (V[j], I[j]). \tag{5.6}$$

- A second trigger point, when present, is obtained by searching for the next local maximum after the holding point,

$$V[t-1] < V[t] \ \wedge \ V[t] > V[t+1], \qquad t > j, \tag{5.7}$$

  and defining $(V_{t2}, I_{t2}) = (V[t], I[t])$. If no second local maximum exists, the curve endpoint is used to represent the high-current limit of the available dataset.

This procedure is applied identically to the TCAD reference and to every SPICE candidate curve, ensuring that the comparison is based on the same set of snapback landmarks. The points that were obtained and used for comparison using the algorithm described in this subsection are clearly visible in Figure 5.2.

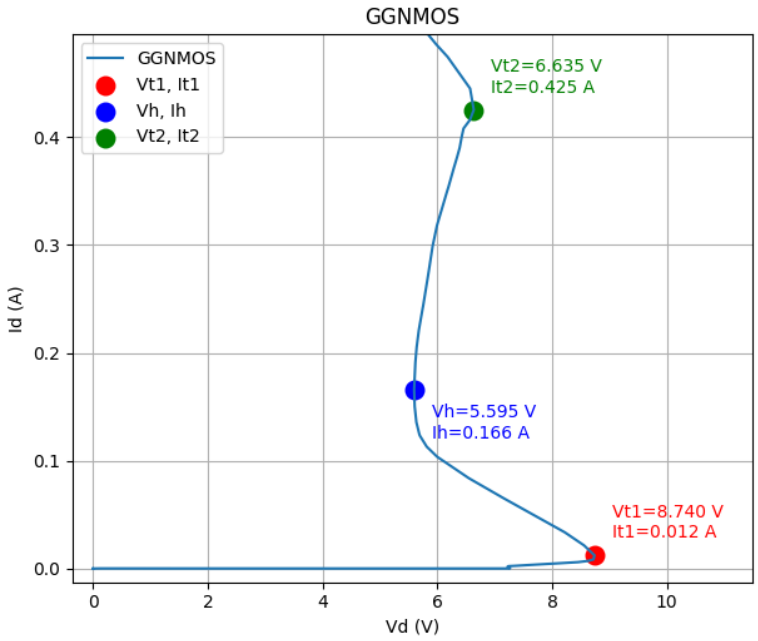


Figure 5.2: Example of reference snapback I–V curve obtained from TCAD, highlighting the extracted characteristic points (V_{t1}, I_{t1}), (V_H, I_H), and (V_{t2}, I_{t2}) used for macromodel calibration.

5.2.2 Weighted error metric and ranking

After extracting the snapback points, each candidate is ranked according to a weighted distance to the reference. Let the reference point set be

$$\varepsilon_{\text{ref}} = \{V_{t1}^{\text{ref}}, V_H^{\text{ref}}, V_{t2}^{\text{ref}}, I_{t1}^{\text{ref}}, I_H^{\text{ref}}, I_{t2}^{\text{ref}}\}, \tag{5.8}$$

and similarly define ε_k for the k-th `ALTER` candidate. The script computes a weighted root-mean-square error of the form

$$\varepsilon_k = \sqrt{w_{Vt1}(\Delta V_{t1})^2 + w_{VH}(\Delta V_H)^2 + w_{Vt2}(\Delta V_{t2})^2 + w_I\left[(\Delta I_{t1})^2 + (\Delta I_H)^2 + (\Delta I_{t2})^2\right]}, \tag{5.9}$$

where $\Delta V_{t1} = V_{t1}^{\text{ref}} - V_{t1}^{(k)}$, $\Delta V_H = V_H^{\text{ref}} - V_H^{(k)}$, and analogous definitions apply to the remaining terms. The weights w_{Vt1}, w_{VH}, w_{Vt2}, and w_I are chosen to reflect the relative importance of each landmark for the intended macromodel behavior. The candidates are then sorted by ε_k, and the best-ranked parameter sets are selected for detailed inspection.

Finally, the script produces:

(i) a ranked table summarizing the extracted points for the best candidates

(ii) an overlaid I–V plot comparing the reference curve to the top-ranked SPICE curves. An example of a test is shown in Figure 5.3.

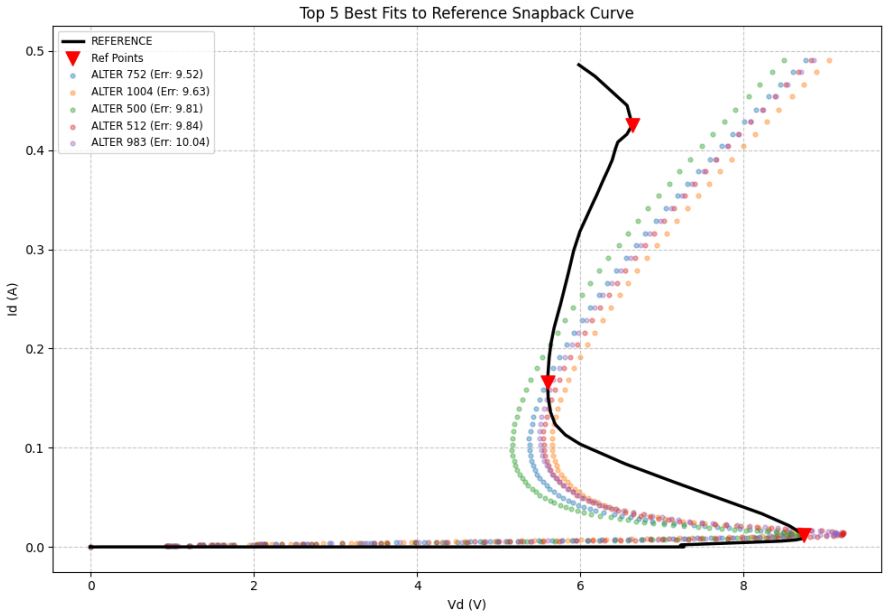


Figure 5.3: Comparison between the TCAD reference snapback I–V curve and the five best-ranked SPICE candidate curves, with the reference points (V_{t1}, I_{t1}), (V_H, I_H), and (V_{t2}, I_{t2}) highlighted in red.

This method simplifies the choice of parameter sets that best reproduce the TCAD snapback behavior in SPICE. By extracting the same snapback points from every curve and scoring each `ALTER` case with a weighted error metric, the fitting becomes a structured selection process instead of manual trial-and-error. In practice, this approach identifies the optimal parameter set, preserves a clear link to the specific SPICE simulation run from which it was derived, and increases the SEED by reducing the time spent manually testing one-by-one combinations of parameter values.

5.3 Summary

This chapter presented the development of a GGNMOS macromodel to reproduce the snapback behavior of GGNMOS devices under ESD conditions in SPICE. The proposed subcircuit, based on Zhou work [17], combines a `BSIM4` NMOS with a parasitic lateral NPN (LNPN) described by `VBIC`, extending the conventional PDK description to cover the avalanche and snapback regions. In this way, the impact ionization-driven substrate current and the parasitic BJT activation can be represented using standard compact-model formulations, while avoiding behavioral current sources that may cause convergence errors.

A calibration strategy was also described to fit the macromodel to a reference snapback I–V curve obtained from TCAD (or TLP test). Candidate curves were generated in SPICE by sweeping the fitting parameters, and a script-based flow was used to extract snapback landmarks (V_{t1}, V_H, and V_{t2}) and rank all parameter sets through a weighted error. This procedure provides a structured alternative to manual trial-and-error, and it supports iterative refinement by narrowing the

parameter ranges around the best candidates.

In addition, the relevance of junction capacitances such as C_{JC} was discussed for transient pulse conditions, since rise-time effects can shift the extracted trigger point (V_{t1}) and modify the apparent snapback response. Overall, the calibrated macromodel provides a practical basis for circuit-level evaluation of triggering and holding conditions and for subsequent integration of protection devices in a SEED design flow. This makes it possible to compare the core behavior with and without protection, and to assess how additional input components – such as series impedance or capacitance – can modify both the protection response and the core electrical behavior.

Chapter 6

Simulations and Results

The experimental stage of this work is defined by the fundamental objective of any ESD protection device: to avoid permanent damage caused both by the large discharge current and caused by an overvoltage during an ESD event. In advanced CMOS nodes, the margin between normal operation and irreversible damage can be small, meaning that the protection solution must be evaluated under requirements that combine robustness, total area, and impact on circuit performance [7, p. 97]. In this context, pass/fail qualification alone is not sufficient to support design decisions, because it does not explicitly reveal how the protection device behaves across the relevant current and voltage regimes.

Following the SOA interpretation, the protection device must operate inside a limited electrical range defined by:

(i) the minimum voltage expected to turn on the protection device, i.e., $V_{\mathbf{t1}}$ must occur within the `ESD Design Window`, avoiding clamping within the `Operating` area plus a margin for signal noises; these areas are illustrated in Figure 3.9;

(ii) the maximum pulsed voltage that the devices connected to the pin can tolerate without entering destructive regimes, illustrated in Figure 3.9 as the `Gate Oxide Breakdown` area.

(iii) Every simulation will stop if the device reaches the silicon melting point, which the temperature is approximately 1687K [7, p. 6] [13, p. 327] [2, p. 45].

For snapback devices, this translates into explicit requirements on the I–V characteristic: the first triggering voltage $V_{\mathbf{t1}}$ should remain above the operational voltage (e.g., $V_{\mathbf{DD}}$ plus a noise margin), while the holding voltage $V_{\mathbf{h}}$ should also stay above $V_{\mathbf{DD}}$ to mitigate latch-up during the ESD stress [7, p. 99]. These constraints motivate the set of characterization tests reported in this chapter, which are structured to verify whether the GGNMOS design covers the protection window while maintaining off outside the stress regime.

Therefore, the goal of the tests presented next is not only to confirm that GGNMOS can conduct high transient currents without failure but also to quantify the key operating points, namely $V_{\mathbf{t1}}$, $V_{\mathbf{h}}$, and $R_{\mathbf{on}}$ under pulsed conditions [7, p. 13]. The evidence base for the rest of this chapter

is provided by these measurements, which are used to discuss the observed trade-offs between protection robustness and circuit-level constraints as well as to relate the extracted parameters to the design requirements.

6.1 TCAD

After designing the GGNMOS in TCAD, as discussed in section 4.2, some fast transient simulations are performed in order to emulate the behavior of the DUT according to typical ESD events, more specifically a TLP pulse, equivalent to a HBM. The goal is to extract as much information as possible about the device and to understand how each parameter influences its performance. As a practical thermal-failure stopping criterion, each simulation is terminated if the maximum device temperature reaches the silicon melting point.

6.1.1 Gate channel length (L)

The channel length has a significant impact on GGNMOS behavior, since a shorter channel effectively reduces the lateral distance between the source and drain diffusions, which corresponds to a smaller effective base width of the parasitic lateral NPN that conducts after snapback.

In this sense, the gate L becomes a key layout parameter for ESD conduction efficiency, since it can be interpreted as the equivalent base width of the parasitic NPN [2, p. 266]. This interpretation is also consistent with the device level description that links the spacing between the relevant

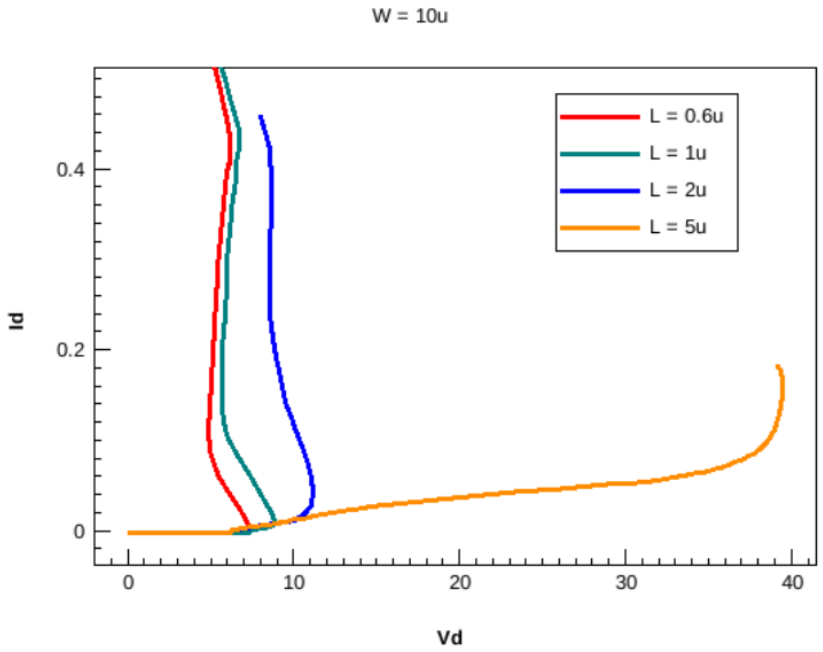


Figure 6.1: TCAD I–V plot showing snapback characteristics of the GGNMOS for different channel lengths L, with the device width fixed to W $= 10\,\mu$m.

diffusions to the base width of the dominant parasitic NPN involved in the snapback process [13, p. 202].

As L is reduced, the effective base width decreases, which tends to increase the current gain of the parasitic bipolar path, and this is reflected in the pulsed I–V as changes in the characteristic snapback points. In particular, results reported for GGNMOS structures show that increasing the gate length produces a clear shift in the triggering and holding related points (e.g., $V_{\mathbf{t1}}$ and $V_{\mathbf{h}}$), consistent with the change of the parasitic BJT gain associated with base width variations [2, p. 218].

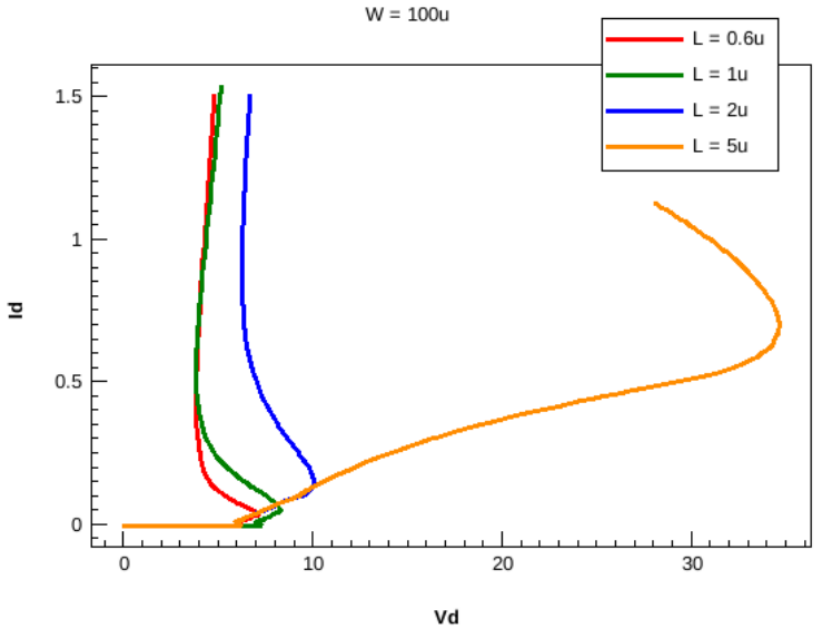


Figure 6.2: TCAD snapback I–V characteristics of the GGNMOS for different channel lengths L, with the device width fixed to W $= 100\,\mu$m.

In Figure 6.1, it is possible to observe a systematic shift of the snapback-related points as L is varied. As the channel length is reduced, both the triggering voltage V_{t1} and the holding voltage V_h move towards lower values. This pattern is consistent with the interpretation that the NMOS channel length is closely related to the effective base width of the dominant parasitic lateral NPN that sustains conduction after snapback; therefore, reducing L tends to strengthen the bipolar conduction path and modifies the observed trigger and holding conditions [2, p. 266]. In practice, this means that shorter L can facilitate earlier triggering (lower $V_{\mathbf{t1}}$) and reduce the post-trigger operating voltage (lower $V_{\mathbf{h}}$), which can be used to limit pad over-voltage during a discharge, better fitting the voltage and currents inside the `ESD Design Window`.

The same qualitative dependence on L is observed in Figure 6.2, indicating that the channel-length scaling effect is not limited to the narrow device case. With a larger width, the curves mainly shift due to the increased current capability, yet the relative displacement of $V_{\mathbf{t1}}$ and $V_{\mathbf{h}}$ with L remains visible, supporting the idea that L acts as a primary geometric parameter of the parasitic lateral BJT. From a design perspective, these results reinforce that L must be selected by

balancing two competing requirements: ensuring that V_{t1} remains above the IC functional voltage range plus margin (to avoid unintended triggering), while keeping the post-trigger conduction sufficiently strong (low effective resistance and acceptable V_h) to limit the stress voltage during ESD transients.

6.1.2 Gate channel width (W)

With the L fixed at 1μm, the GGNMOS was tested with progressively increased values of W to analyze the device's behavior. In practice, increasing W mainly increases the available discharge current under snapback, since the conduction path after triggering is distributed along the layout width (typically implemented as multiple fingers [7, p. 57]). This scaling aspect is one of the reasons why width variations are commonly used in test structures to evaluate how the protection I–V evolves with device size [13, p. 123].

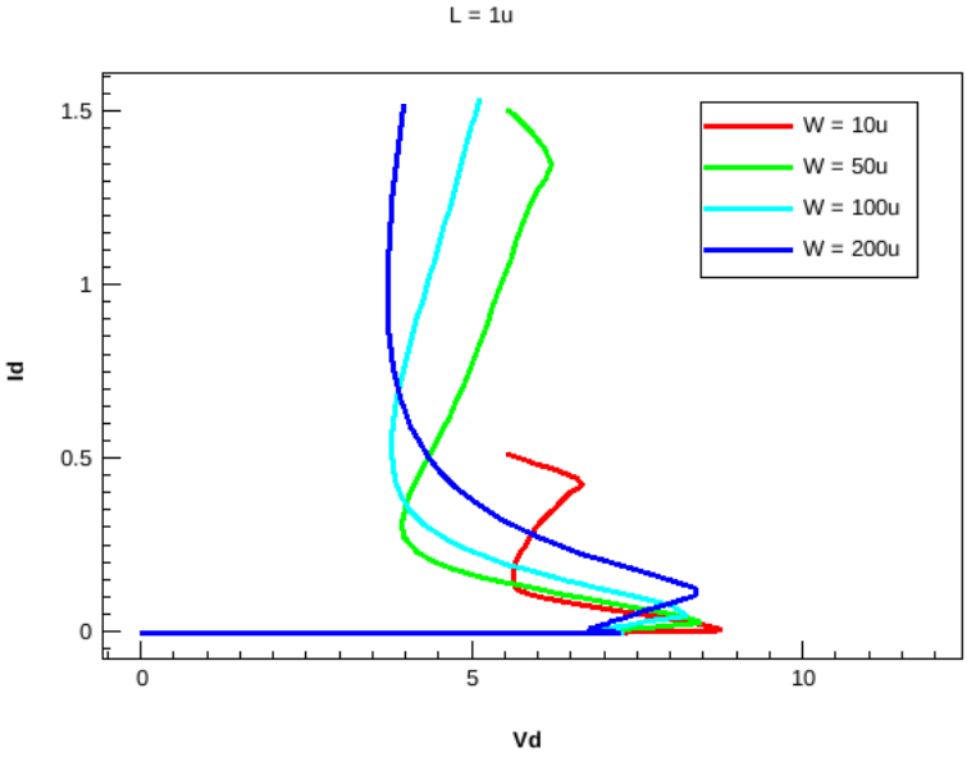


Figure 6.3: TCAD snapback I–V characteristics of the GGNMOS for different device widths W, with the channel length fixed to L = 1 μm.

In Figure 6.3, the most visible effect of increasing W is the increase in the post-trigger current level, while the voltage landmarks associated with snapback (such as V_{t1} and V_h) tend to vary less than the current capability. This observation follows the conventional sizing strategy: the target robustness is converted into a required total conduction width by considering how much current the device can conduct for each unit of width. For snapback devices based on CMOS, Semenov discusses the use of a current capability normalized per unit width as a practical basis to

estimate the total width required for a given ESD stress level [7, p. 59]. In this sense, the curves in Figure 6.3 provide evidence that widening the structure is an effective way to increase the safe discharge current before reaching destructive regimes, which is directly related to pushing the thermal breakdown limit (often associated with (V_{t2}, I_{t2})) to higher currents [2, p. 190].

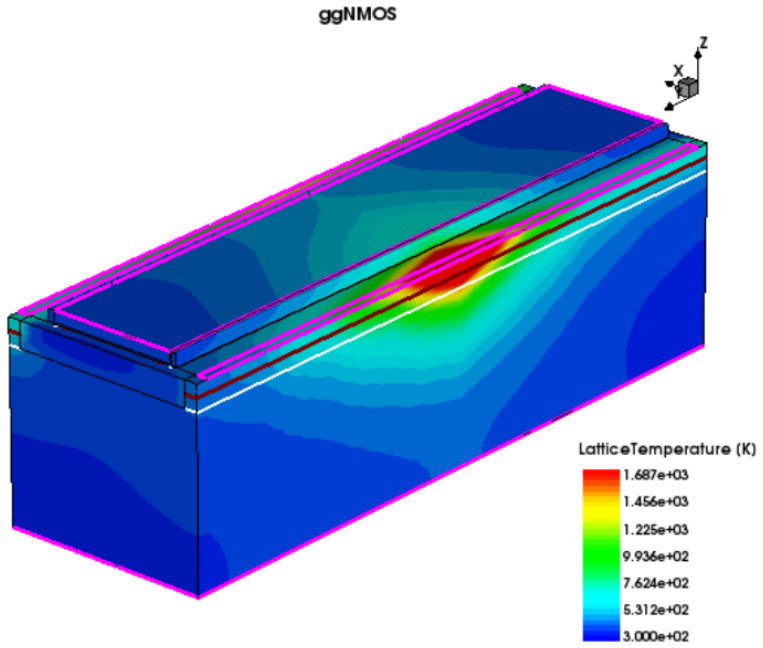


Figure 6.4: Visualization of thermal crowding obtained from TCAD during a TLP simulation for a GGNMOS with L = 1 μm and W = 10 μm.

However, the benefit of increasing W is not proportional, because uniform triggering and uniform current conduction across the full width are not guaranteed in large devices. Practical designs therefore avoid very long single fingers, since high ESD current can lead to local current crowding and heat crowding effects, as observed in the simulation in Figure 6.4, creating hot spots that reduce the thermal failure threshold [2, p. 113]. Amerasekera and Duvvury also report that the second breakdown current I_{t2} of a single finger can degrade with finger width due to nonuniform conduction, meaning that only a fraction of the finger effectively carries the ESD current, which reduces the effective robustness [13, p. 196]. Consistently, Semenov correlates width scaling to the concept of an effective conduction under snapback, noting that current crowding limits how much of the device's geometrical width actually conducts during stress [7, p. 107–108]. For this reason, multi-finger layouts and ballasting concepts are commonly used to promote more uniform conduction, and Voldman explicitly discusses how multi-finger MOSFET structures require careful source and substrate referencing and, when needed, added resistive ballast to reduce current localization [14, p. 229–230].

From a design perspective, the width selection must also consider the overhead introduced by a larger protection device. Increasing W generally implies higher parasitic loading at the pad (including capacitance), which is a known trade-off between robustness and circuit performance,

particularly relevant in analog and high frequency interfaces [7, p. 309]. Voldman also frames the same compromise by relating ESD robustness to capacitance limited design targets in performance constrained applications [14, p. 868]. Therefore, the role of W in Figure 6.3 can be summarized as follows: it is the primary parameter to increase the discharge current capability and improve margin against thermal breakdown, but it must be implemented with layouts that preserve its uniformity, and it must respect the constraints imposed by the `ESD Design Window` and by the pin performance requirements [2, p. 306].

6.1.3 Electrical gate equivalent oxide thickness (TOXE)

The equivalent gate oxide thickness, TOXE, is a key parameter when defining the maximum transient voltage that the devices connected to the protected pin can tolerate without gate oxide breakdown. In Figure 3.9, this limit is highlighted by the `Gate Oxide Breakdown` region (marked in red), which represents the maximum voltage that the gate dielectric of the connected devices can tolerate before irreversible damage occurs.

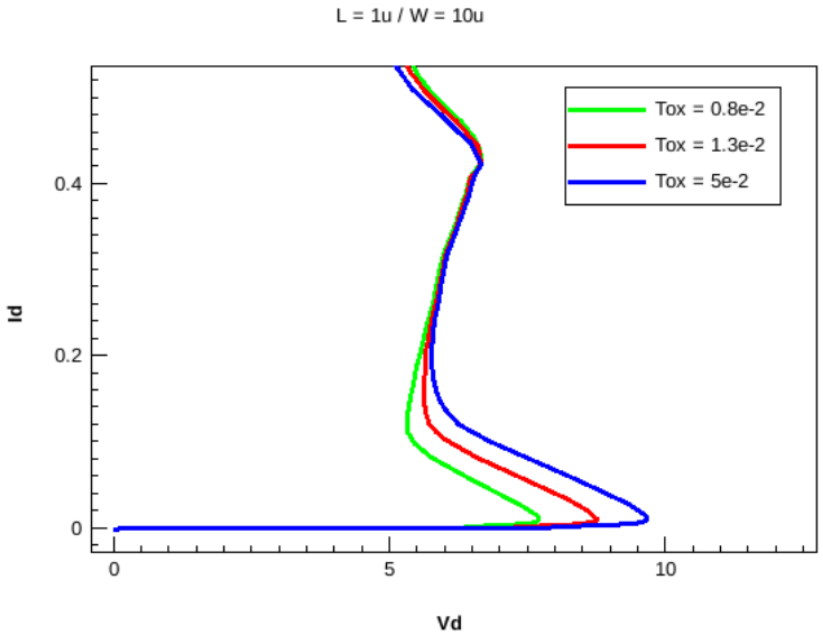


Figure 6.5: TCAD snapback I–V characteristics of the GGNMOS for different TOXE values, with the device geometry kept constant.

As fabrication technologies advance, ICs dimensions continue to scale down, consequently reducing the gate oxide thickness and, as a result, the gate oxide breakdown voltage also decreases, shrinking the available ESD design window between normal operation (around V_{DD}) and destructive dielectric regimes [7, p. 98]. This constraint is consistently emphasized in the literature [14, p. 882] [2, p. 141] [13, p. 9], therefore the protection device must be sized and tuned so that its triggering (V_{t1}) occurs before the protected node reaches destructive voltage levels,

while still remaining above the normal operating range. This requirement becomes more critical in scaled technologies because thinner gate oxides tolerate less overvoltage during Electrical Overstress (EOS)/ESD events.

In Figure 6.5, increasing TOXE shifts the first triggering voltage V_{t1} towards higher values. This trend is consistent with the snapback triggering mechanism described for a GGNMOS, where avalanche generation produces substrate current that raises the base potential of the parasitic bipolar transistor, and V_{t1} is reached when this bipolar path turns on [7, p. 54]. If the TOXE is increased, the effective gate control is weakened through its impact on the oxide capacitance term used in the MOSFET relations, which modifies both the threshold voltage and the drain current [2, pp. 94-99]. Under these conditions, a higher drain voltage may be required to generate sufficient current and impact ionization to reach the bipolar turn-on condition, which appears in the simulated pulsed I–V as a higher V_{t1} for larger TOXE.

6.1.4 Drain Contact to Gate Spacing (DCGS) and Source Contact to Gate Spacing (SCGS)

The layout distances DCGS and SCGS are commonly used to tune CMOS-based snapback clamps, including GGNMOS, because they change the effective series resistance and the current distribution during an ESD discharge [7, p. 122] [13, p. 134] [2, p. 188]. Although these spacings are also discussed in smart-power devices such as LDMOS [7, p. 177] [14, p. 698], the same concepts apply to a GGNMOS protection device because they refer to how the contacts are placed relative to the gate in the layout [2, p. 188].

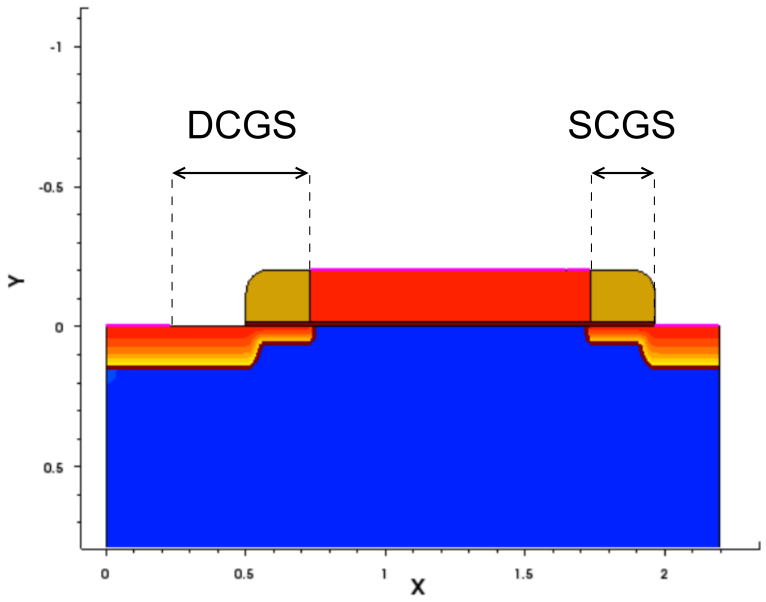


Figure 6.6: Cross-section of the GGNMOS illustrating the definitions of Drain Contact to Gate Spacing (DCGS) and Source Contact to Gate Spacing (SCGS).

From a physical and layout perspective, DCGS is directly related to the drain-side ballasting effect, since the distance between the gate edge and the drain contact influences the effective series resistance and, therefore, the tendency for current crowding during snapback [7, p. 4]. Semenov explicitly notes that reducing the drain to gate spacing can lead to poor ballasting in GGNMOS clamps [7, p. 122], while Amerasekera and Duvvury include the drain contact to gate spacing as one of the main CMOS design parameters for ESD performance [13, p. 134]. In practice, Wang summarizes this as a design guideline where SCGS is kept small and DCGS is kept sufficiently large to support the intended discharge behavior [2, p. 188].

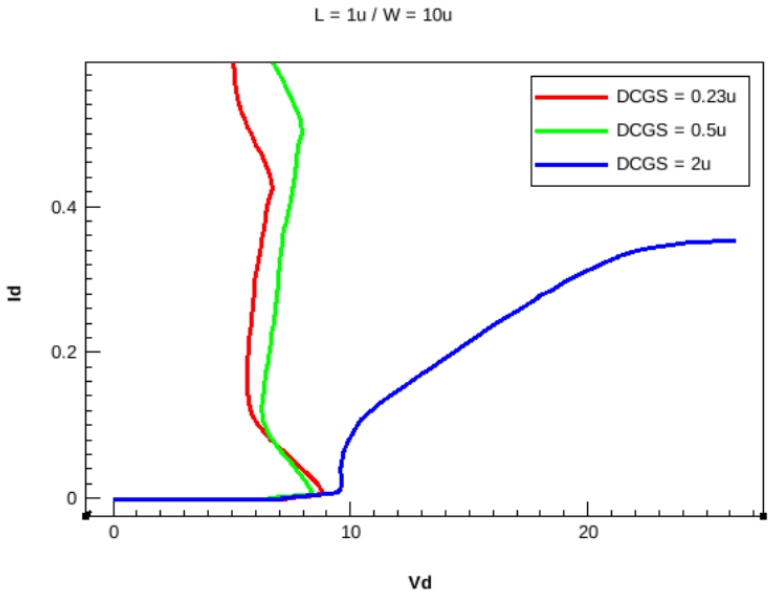


Figure 6.7: TCAD snapback I–V characteristics for different DCGS values (0.23 μm, 0.5 μm, and 2 μm), with L fixed to 1 μm and W fixed to 10 μm.

The DCGS sweep in Figure 6.7 shows that changing DCGS modifies the pulsed I–V response mainly through the drain-side series contribution, which affects how the device supports current after snapback. This is consistent with the interpretation that DCGS provides an effective ballasting mechanism: smaller DCGS reduces the available ballast, which can enhance current localization, whereas larger DCGS increases the effective series contribution and tends to promote a more distributed conduction path [7, p. 122] [13, p. 137]. From a design point of view, this means that DCGS is not only a geometric parameter, but also a trade-off factor between conduction efficiency and robustness against non-uniform triggering [2, p. 188].

For SCGS, the literature often treats it as a secondary electrical parameter, and it is commonly kept close to the minimum design value in CMOS-based clamps [13, p. 134]. However, Wang adds an important practical restriction for advanced nodes: one should not simply choose the minimum allowed SCGS, because heat generated at the drain junction can spread across the short channel and lead to metal melting at the source contact [2, p. 188]. The sweep in Figure 6.8 therefore helps to assess how sensitive the pulsed I–V is to SCGS, and to identify a spacing range that remains

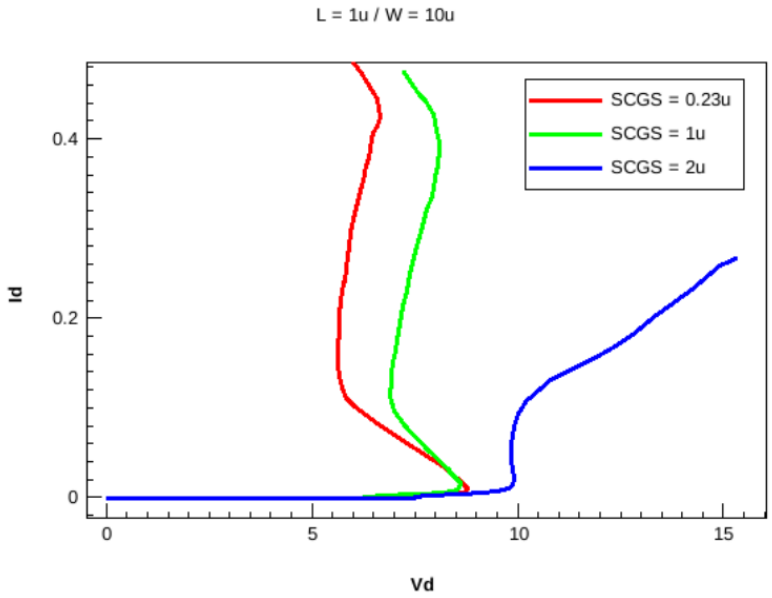


Figure 6.8: TCAD snapback I–V characteristics for different SCGS values (0.23 μm, 1 μm, and 2 μm), with L fixed to 1 μm.

compatible with the intended protection behavior while avoiding overly aggressive source contact placement [2, p. 188].

6.2 GGNMOS SPICE macromodel

After obtaining the behavior of the protection device, either in TCAD or even after tape-out and test, the parameters associated with the GGNMOS SPICE macromodel are then calibrated. Details of the model structure, the physical meaning of its elements, and the extraction and ranking workflow are given in chapter 5 (section 5.2 and Figure 5.3). The discussion therefore focuses on what the calibrated curves demonstrate and on how well the best parameter sets reproduce the reference snapback behavior for circuit-level analysis.

The calibration is performed by comparing a reference snapback I–V curve with candidate SPICE curves generated through `ALTER` sweeps of the fitting parameters (Appendix C), following the workflow defined in section 5.2. Each candidate curve is reduced to the same set of turning points used throughout this work, namely (V_{t1}, I_{t1}), (V_H, I_H), and, when present, (V_{t2}, I_{t2}), together with a post-trigger conduction indicator summarized by an effective R_{on} extracted from the post-trigger region, as shown in Figure 3.9. The purpose of these metrics is to provide a compact, comparable description for all simulated parameter values.

6.2.1 Validation of calibration parameters

To improve sensitivity to the calibration parameters and to build a robust starting point for the sweep ranges, the following procedure was adopted:

(i) a few initial simulations were run with small parameter changes until snapback became clearly visible in the I–V curve;

(ii) an initial parameter set that roughly mirrored the reference curve was chosen as the baseline;

(iii) with all other parameters fixed, each parameter was swept individually above and below the baseline to observe its isolated effect on snapback landmarks;

(iv) the same separate sweep was repeated for the remaining parameters.

The following one-variable-at-a-time (OVAT) method sweeps provide technical evidence of how each parameter perturbs the macromodel response in the regions that is relevant for calibration. The goal is not to re-derive the underlying equations (already covered in chapter 5), but to document how each knob displaces the extracted landmarks and which regions of the I–V are primarily affected.

6.2.1.1 $R_{\mathbf{sub}}$

The OVAT sweep of R_{sub} in Figure 6.9 produces a clear displacement of the snapback curve and of the extracted landmarks. In particular, $V_{\mathbf{t1}}$ shifts noticeably as R_{sub} is varied, which is consistent with the fact that the snapback trigger is associated with the turn-on of the parasitic bipolar path. On the other hand, V_H and $R_{\mathbf{on}}$ show a less pronounced change, although a small difference can still be observed across the sweep.

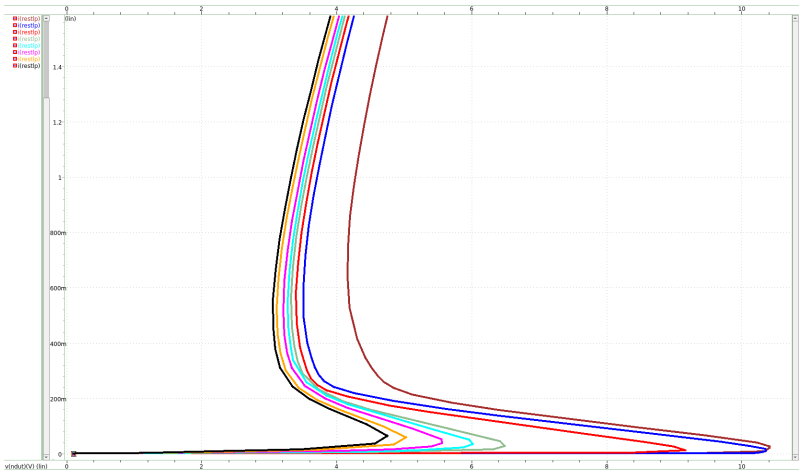


Figure 6.9: I–V plot of OVAT sweep of R_{sub} with all other parameters fixed. Curves are ordered from left to right as $R_{\text{sub}} \in \{3000, 1500, 800, 600, 500, 400, 300, 100\}\Omega$.

From a calibration perspective, R_{sub} is treated as a primary tuning parameter during the initial coarse search, since it helps place V_{t1} and V_H within the correct range relative to the reference curve. Once snapback is established and the landmark positions are close to the target, R_{sub} is adjusted more gently in the refinement stage, as large changes tend to shift the entire curve and make it harder to isolate mismatches at higher currents.

6.2.1.2 Avalanche shaping parameters ($AVC1$, $AVC2$, and MC)

In the macromodel, $AVC1$, $AVC2$, and MC are used to tune the collector–base avalanche contribution in the parasitic bipolar path, as defined by the `VBIC` avalanche term introduced in chapter 5. Physically, this tuning is relevant because snapback triggering is initiated by avalanche generation near the drain, which produces substrate current that forward biases the parasitic BJT and defines $V_{\mathbf{t1}}$, followed by the voltage drop to the holding level $V_{\mathbf{h}}$ [7, p. 54]. This same chain is also described in terms of avalanche multiplication, where the generated holes provide base current for the lateral bipolar element and the resulting substrate potential rise depends on R_{sub} [14, p. 249]. Amerasekera and Duvvury further emphasize that snapback is sustained by the interaction between multiplication and bipolar gain, and that the sustaining voltage depends on the conditions required to keep the bipolar path on [13, p. 355]. From a circuit simulation perspective, this explains why the GGNMOS snapback I–V cannot be represented by conventional compact models without extending the avalanche and bipolar description [2, p. 97].

The OVAT sweeps in Figure 6.10, Figure 6.11, and Figure 6.12 show that these avalanche shaping parameters can shift the extracted snapback landmarks and also modify the curvature of the curve after triggering.

In Figure 6.10, varying $AVC1$ produces an evident translation of the snapback curve along the voltage axis, and both $V_{\mathbf{t1}}$ and $V_{\mathbf{h}}$ move by a comparable amount, which indicates that the trigger and holding levels are displaced together. By contrast, changes in $AVC2$ (Figure 6.11) and MC (Figure 6.12) tend to be more visible in how the curve evolves as current increases above the holding region, affecting the curvature and, when present in the dataset, the location and prominence of the high current landmark associated with $(V_{\mathbf{t2}}, I_{t2})$. In practical terms, $AVC1$ mainly controls a voltage level offset of the snapback transition, while $AVC2$ and MC are more associated with shaping the transition towards higher current.

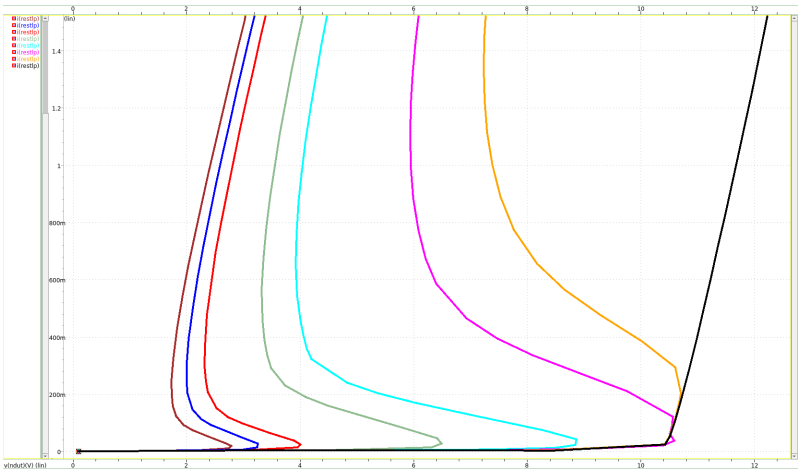


Figure 6.10: I–V plot of OVAT sweep of $AVC1$ with all other parameters fixed. Curves are ordered from left to right as $AVC1 \in \{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1.0, 2.0\}$.

For calibration, this observation motivates a staged use of these parameters after a baseline

snapback response is already obtained. Once R_{sub} has placed the response in the correct qualitative regime, $AVC1$ can be used to reduce the remaining offset in $V_{\mathbf{t1}}$ and $V_{\mathbf{h}}$ with respect to the reference, and then $AVC2$ and MC can be refined to improve agreement in the region above $V_{\mathbf{h}}$ and near the high current end of the available curve. This ordering reduces the risk of correcting high current curvature at the cost of misplacing $V_{\mathbf{t1}}$ inside the operating range, or of keeping $V_{\mathbf{t1}}$ too high and approaching destructive limits before the clamp becomes effective.

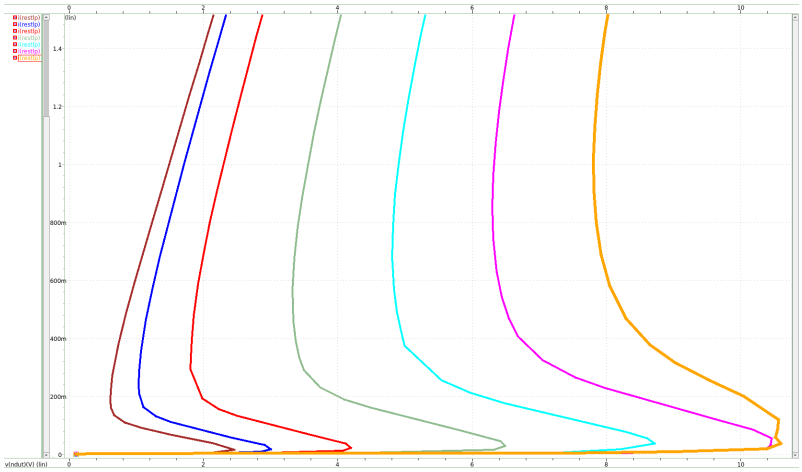


Figure 6.11: I–V plot of OVAT sweep of $AVC2$ with all other parameters fixed. From left to right, the curves correspond to $AVC2 \in \{1,\ 2,\ 4,\ 8,\ 12,\ 16,\ 20\}$.

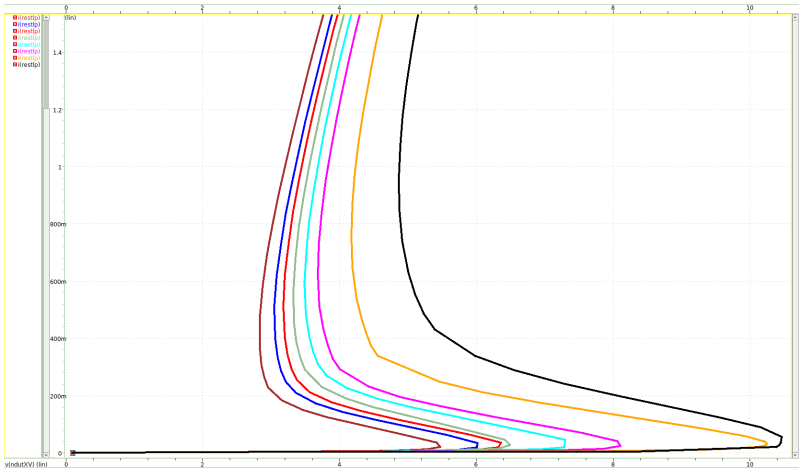


Figure 6.12: I–V plot of OVAT sweep of MC with all other parameters fixed. From left to right, the curves correspond to $MC \in \{0.10,\ 0.20,\ 0.25,\ 0.30,\ 0.35,\ 0.40,\ 0.50,\ 0.60\}$.

6.2.1.3 *IS*

The OVAT sweep of IS in Figure 6.13 changes the current scale associated with the bipolar conduction path, and this is reflected in the post-trigger region of the I–V curve. Even when the triggering location remains comparable, modifying IS can change how the curve progresses after

snapback, which can influence the extracted holding-related quantities and the relative shape of the conduction region.

For calibration, IS is treated as a secondary adjustment knob that is useful once the major voltage landmarks are already close to the reference. In practice, IS can help reduce mismatch in the current levels around the holding region, improving consistency of the curve without requiring large changes in parameters that primarily determine trigger placement.

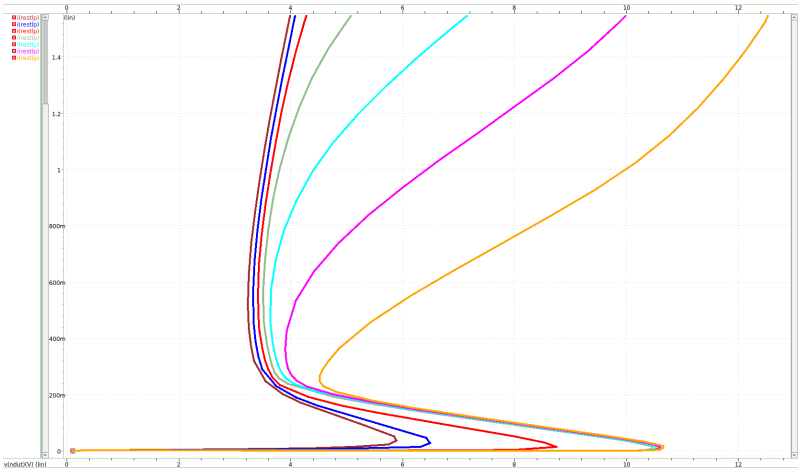


Figure 6.13: I–V plot of OVAT sweep of IS with all other parameters fixed. Values: $IS \in \{10^{-6}, 10^{-7}, 10^{-8}, 10^{-9}, 10^{-10}, 10^{-11}, 10^{-12}\}$ A.

6.2.1.4 R_d

The OVAT sweep of R_d in Figure 6.14 changes the voltage drop associated with the series path, which is primarily observed in the post-trigger region where current is high. As R_d is increased, the curve exhibits a change in voltage level for comparable current, which decreases the overall slope in the conduction region.

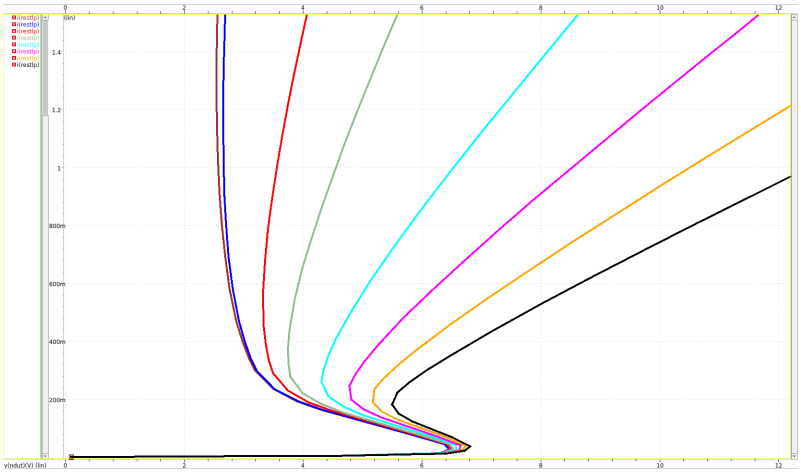


Figure 6.14: I–V plot of OVAT sweep of R_d with all other parameters fixed. Values: $R_d \in \{0.01, 0.1, 1, 2, 4, 6, 8, 10\}\,\Omega$.

In calibration terms, R_d is mainly used to tune the effective conduction resistance after snapback, since it has limited influence on $V_{\mathbf{t1}}$ but it can shift $V_{\mathbf{h}}$ and, more strongly, the high-current region near $V_{\mathbf{t2}}$. Therefore, R_d is applied in the refinement stage to tune the voltage levels after snapback triggering, mainly $V_{\mathbf{h}}$ and the region around $V_{\mathbf{t2}}$.

6.2.1.5 $C_{\mathbf{JC}}$

The OVAT sweep of $C_{\mathbf{JC}}$ in Figure 6.15 targets the transient component of the macromodel response, since junction capacitance introduces displacement current during fast voltage transitions. Even when the underlying quasi-static snapback mechanism is unchanged, this capacitive current can momentarily alter the effective bias conditions that lead to snapback triggering, which may shift the extracted $V_{\mathbf{t1}}$ under short rise-time stress.

After obtaining a baseline snapback fit using the main parameters related to avalanche and substrate effects, $C_{\mathbf{JC}}$ acts as a fine-tuning parameter. At this stage, $C_{\mathbf{JC}}$ mainly shifts $V_{\mathbf{t1}}$ under fast transitions, while the holding level $V_{\mathbf{h}}$ and the post-trigger conduction remain approximately unchanged across the sweep.

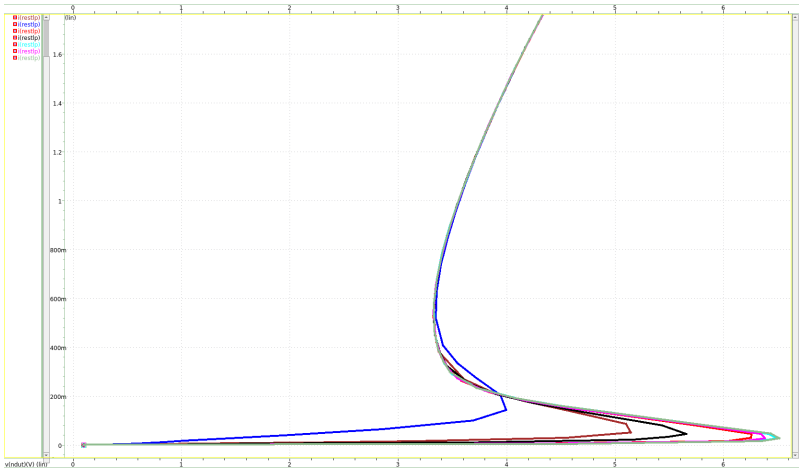


Figure 6.15: I–V plot of OVAT sweep of $C_{\mathbf{JC}}$ with all other parameters fixed. From left to right, the curves correspond to $C_{\mathbf{JC}} \in \{1\times10^{-16}, 5\times10^{-15}, 1\times10^{-15}, 5\times10^{-14}, 1\times10^{-14}, 5\times10^{-13}, 1\times10^{-13}\}$ F.

6.2.2 Reference data and simulation conditions

To ensure consistency, the turning-point definitions introduced in section 5.2 are applied to the reference curve and to every SPICE candidate. From each I–V curve, the landmarks $(V_{\mathbf{t1}}, I_{\mathbf{t1}})$, $(V_{\mathbf{h}}, I_{\mathbf{H}})$, and, when present, $(V_{\mathbf{t2}}, I_{\mathbf{t2}})$ are extracted. This reduces all curves to a common set of metrics, enabling candidates to be ranked using the weighted error.

Candidate curves are generated in SPICE using `ALTER` sweeps (Appendix C), where each index corresponds to a unique parameter combination. This indexing ensures traceability between the ranked candidates and the specific simulation runs that produced them, which is essential

when the selected macromodel is later integrated into circuit-level simulations associated with the IC core [7, p. 97].

6.2.3 Ranking results and iterative calibration methodology

After the OVAT sweeps reported previously, the macromodel calibration proceeds as an iterative process. The goal is to move from a parameter set close to the default values toward a best-fit solution through a structured search, applying only a small amount of manual adjustment once the solution is already near a local minimum of the ranking error. This approach reduces reliance on "trial and error" while still allowing precise fine-tuning guided by the observed parameter sensitivities.

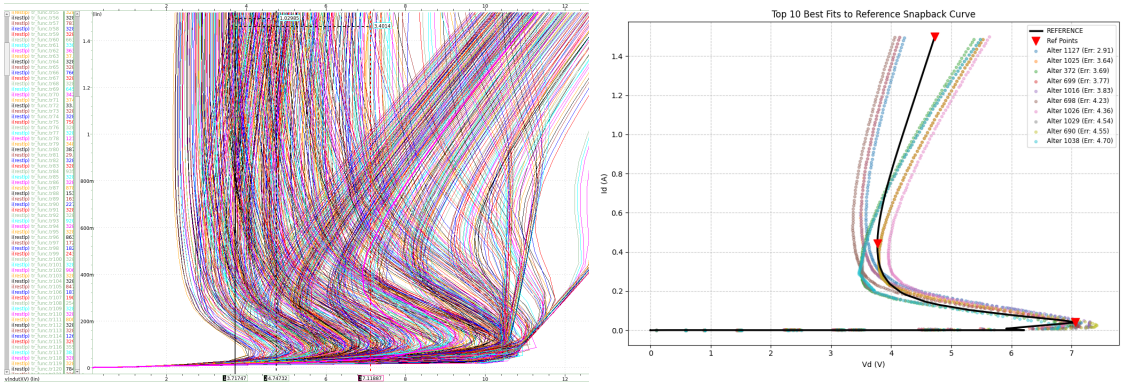
The calibration process was organized into these stages:

(i) *Baseline initialization:* start from values close to `VBIC` defaults and verify that the macromodel produces a physically meaningful response (`BSIM4` was not modified, since the NMOS device used was provided by the PDK);

(ii) *Snapback activation:* increase R_{sub} and IS to obtain a clear snapback transition, ensuring that avalanche and the parasitic bipolar path are visible in the I–V response;

(iii) *Coarse sweep and ranking:* generate a coarse `ALTER` sweep around the baseline and rank candidates using the extracted landmarks and the weighted metric defined in section 5.2;

(iv) *Refinement and manual adjustment:* repeat the sweep around the best-ranked candidates with finer steps, then apply a small manual adjustment based on the parameter influence observed in the OVAT analysis.



(a) Coarse sweep: I–V overlay of all `ALTER` candidates in WaveView.

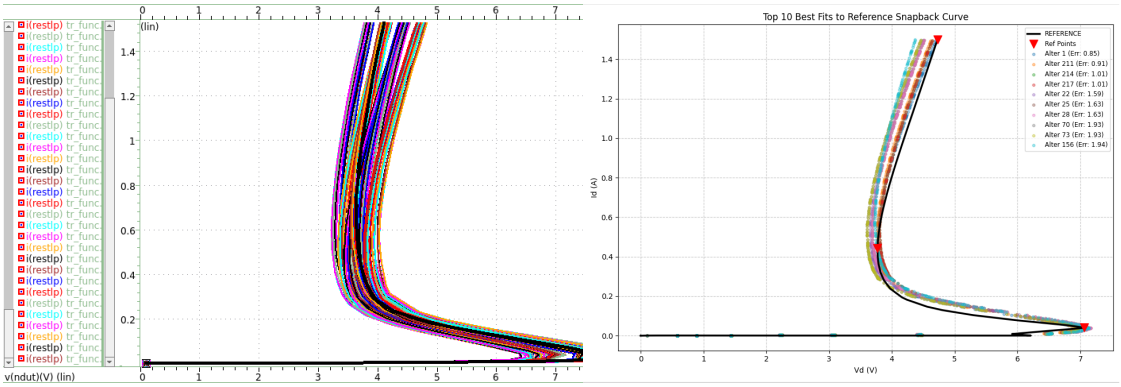
(b) Coarse sweep: top-10 I–V candidates ranked by the landmark error metric.

Figure 6.16: First iteration of the calibration campaign: coarse exploration followed by candidate ranking.

In the first iteration, the coarse overlay in Figure 6.16a shows a wide spread of candidate curves, which is expected because the initial goal is a "global" investigation rather than precision.

At this stage, the ranking in Figure 6.16b serves as a filter: instead of manually selecting curves that "look close", candidates are ranked using the landmark error metric, which directly reflects how well (V_{t1}, V_h, V_{t2}) match the reference.

The refinement stage in Figure 6.17 supports a more critical observation: once the search space is restricted around promising candidates, the overlay in Figure 6.17a becomes visibly more clustered, indicating that the method is not simply producing isolated outliers, but rather converging toward a stable set of parameter values. The second ranking in Figure 6.17b is then interpreted as a local comparison among candidates that are already similar, which is exactly where parameter sensitivity (previously studied with OVAT) becomes valuable in guiding the choice of parameters.



(a) Refined sweep: I–V overlay of candidates generated around the best coarse results.

(b) Refined sweep: top-10 I–V candidates ranked by the landmark error metric.

Figure 6.17: Refinement stage of the calibration campaign: local exploration around the best coarse candidates and ranking of the refined set.

The parameter trajectory across iterations is summarized in Table 6.1. Two methodological points are highlighted by this table. First, it documents how the parameter choices evolve from the baseline to the refinement stages, making the progression of the calibration strategy explicit. In particular, the increase of R_{sub} and IS from the baseline to the first refinement reflects the step where the macromodel is moved into a regime in which snapback is clearly present, so that the subsequent sweeps focus on parameter regions that replicate the intended mechanism. Second, the updates of $AVC1$, $AVC2$, and MC across refinements indicate that the method does not rely on a single parameter to correct all mismatches; instead, it progressively adjusts the remaining offset by combining parameters that influence the avalanche and bipolar turn-on, consistent with the sensitivities observed in the OVAT sweeps.

The final solution is evaluated using two complementary views: the curve overlay in Figure 6.18 and the landmark table in Table 6.2. The table makes the improvement explicit and visible: compared to the baseline, the best fit reduces the mismatch in V_h from 0.196 V to 0.006 V and in V_{t2} from 0.510 V to 0.024 V, while V_{t1} shifts from a $+0.096$ V deviation (baseline) to a -0.118 V deviation (best fit). This result is relevant because it shows that the ranking approach can strongly reduce error in the holding and high-current landmarks, but it also reveals a limitation: improving V_h and V_{t2} can come with a residual offset in V_{t1}, suggesting coupling between the

Table 6.1: Calibration parameter sets used in the SPICE campaign: baseline and refinement iterations.

Baseline and refinement parameter sets for SPICE calibration					
Parameter	**Baseline**	**Refinement 1**	**Refinement 2**	**Final**	**Units / notes**
R_{sub}	300.0	500.0	500.0	500.0	Ω
$AVC1$	1.0	0.1	0.05	0.065	–
$AVC2$	5.0	8.0	8.0	9.0	–
IS	1.0×10^{-10}	1.0×10^{-7}	1.0×10^{-7}	1.0×10^{-7}	A
$C_{\mathbf{JC}}$	1.0×10^{-15}	1.0×10^{-15}	5.0×10^{-15}	2.0×10^{-15}	F
MC	0.33	0.3	0.2	0.18	–
$R_{\mathbf{d}}$	1.0	1.0	1.0	1.27	Ω

mechanisms that place the trigger and those that shape the conduction regime.

This evidence motivates the final tuning stage described in the subsection 6.2.3. Once the solution is near a local minimum, small manual changes can be justified because they are close to the optimum result, and these corrections are guided by the observed parameter influence – the OVAT test. At the same time, the figures and tables also highlight the main risk of manual fine-tuning: while it can adjust one landmark, it can unintentionally degrade another landmark if parameter interactions are not respected. For this reason, the methodology keeps manual changes limited and preserves traceability by capturing the final parameter set in Table 6.1 and validating the resulting landmark alignment in Figure 6.18 and Table 6.2.

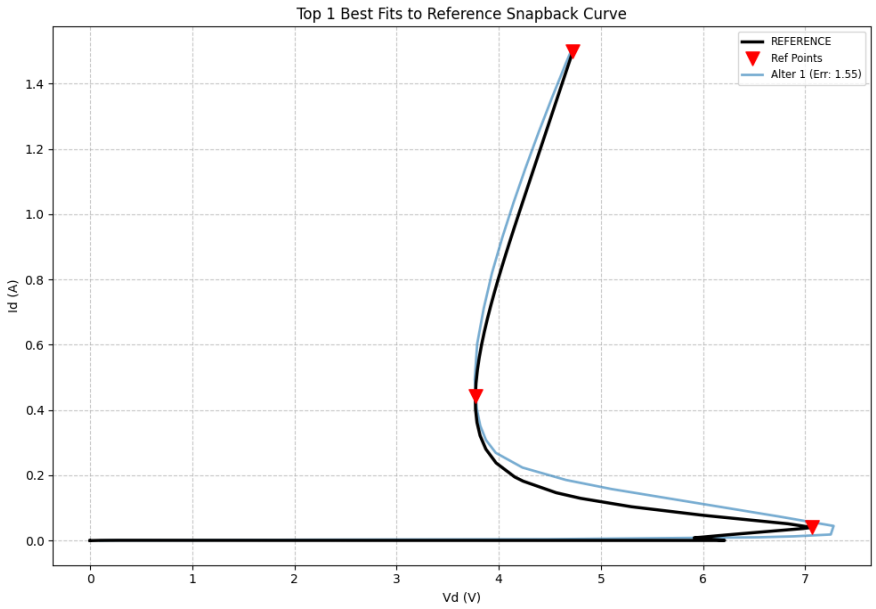


Figure 6.18: Best-fit result after refinement: reference curve versus the best-ranked SPICE candidate, with $V_{\mathbf{t1}}$, $V_{\mathbf{h}}$, and $V_{\mathbf{t2}}$ indicated.

Table 6.2: Extracted snapback voltage landmarks for the reference geometry: reference (TCAD), baseline SPICE, and best-fit calibrated SPICE.

Extracted snapback voltage landmarks			
Case	V_{t1} (V)	V_H (V)	V_{t2} (V)
Reference (TCAD)	7.065	3.771	4.730
Baseline (SPICE)	7.161	3.575	4.220
Best fit (SPICE)	6.947	3.765	4.706

6.3 Qualification tests

After calibrating the GGNMOS macromodel, the next step is to evaluate its behavior under representative discharge conditions. This stage serves as a preliminary qualification step, rather than a full validation of real ESD performance, since a circuit-level setup in SPICE cannot reproduce all discharge paths and interactions of a complete IC, where the attached blocks are optimized for functional operation rather than for ESD current conduction.

The goal is therefore not only to confirm that the circuit exhibits snapback in simulation, but also to verify which conduction path effectively supports the discharge current and whether adding the protection changes the stress observed at the DUT when compared to an unprotected case. The three tests presented in this section focus on the contributions of the N-type MOS (NMOS) with LNPN and on transient comparisons under HBM and CDM conditions, performed with and without the GGNMOS clamp.

6.3.1 Verification of the dominant discharge path

Figure 6.19 decomposes the discharge current into two contributions: the current flowing through the NMOS path (top graph) and the current flowing through the parasitic LNPN (bottom graph). The key observation is that the NPN current closely follows the total discharge profile, while the NMOS current remains comparatively smaller. This provides direct support for the intended physical interpretation of the macromodel: after triggering, the discharge is mainly sustained by the parasitic bipolar conduction, which is the mechanism that must be captured to reproduce snapback behavior in circuit-level simulation.

This current decomposition is also useful as a model validation check. If the calibrated macromodel were producing a snapback I–V while still conducting most of the current through the NMOS channel, the result would be harder to interpret and could indicate an artificial fit. Instead, the separation in Figure 6.19 indicates that the calibrated parameters lead to a discharge behavior that is consistent with the intended snapback conduction path, which increases confidence in using the model for system-level comparisons.

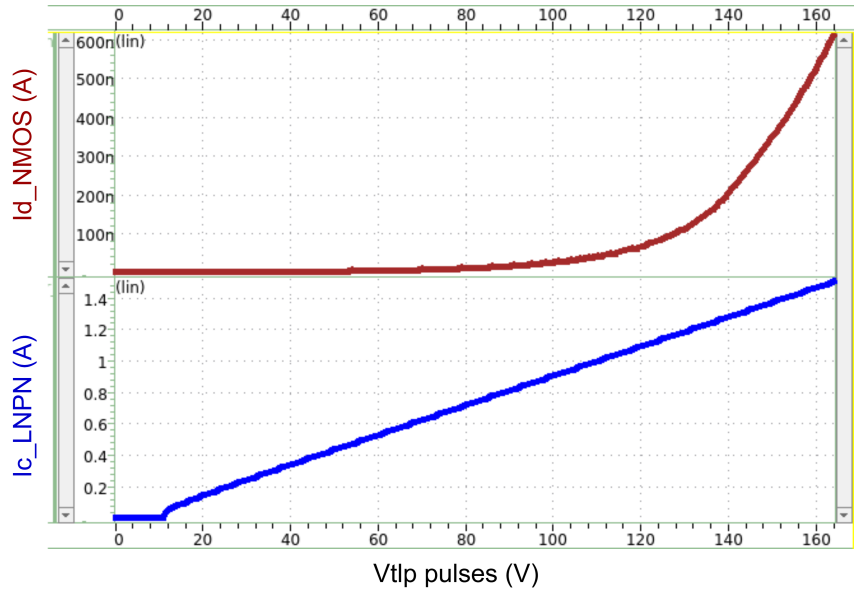


Figure 6.19: TLP test with current decomposition: current through the NMOS path (top) and through the parasitic LNPN path (bottom). The plots indicate that the parasitic bipolar path supports most of the discharge current after triggering.

6.3.2 HBM with GGNMOS

The second qualification step is a direct comparison of the DUT response with and without the GGNMOS protection device. In Figure 6.20, the unprotected case (red) and the protected case (blue) are overlaid in the same plot. This comparison is important because an I–V calibration alone does not guarantee that the transient stress at the DUT node is reduced when the protection device is embedded in the circuit; the discharge waveform depends on the full network, including the DUT impedance and the discharge source.

Figure 6.20 shows two distinct behaviors under a simulated HBM stress of 2 kV. In the unprotected DUT (red), the response evolves into a higher-stress regime, meaning that the DUT is forced to absorb a large current without a dedicated clamp path. When the GGNMOS is added, the response shows a snapback transition and a lower operating voltage during the discharge, indicating that the clamp provides an alternative low-impedance path that changes the stress distribution. In practical terms, this is the expected role of the protection element: to intercept the discharge current and to keep the protected node away from destructive regimes.

6.3.3 CDM with GGNMOS

In addition to the HBM comparison, the calibrated macromodel is also qualified under a CDM test, as explained in subsection 3.1.2. The goal is to verify if the protection element turns on fast enough to protect the node. In this context, the comparison with and without the GGNMOS focuses on the initial voltage peak and on the snapback triggering, which is a necessary condition for the device to be protected during ESD.

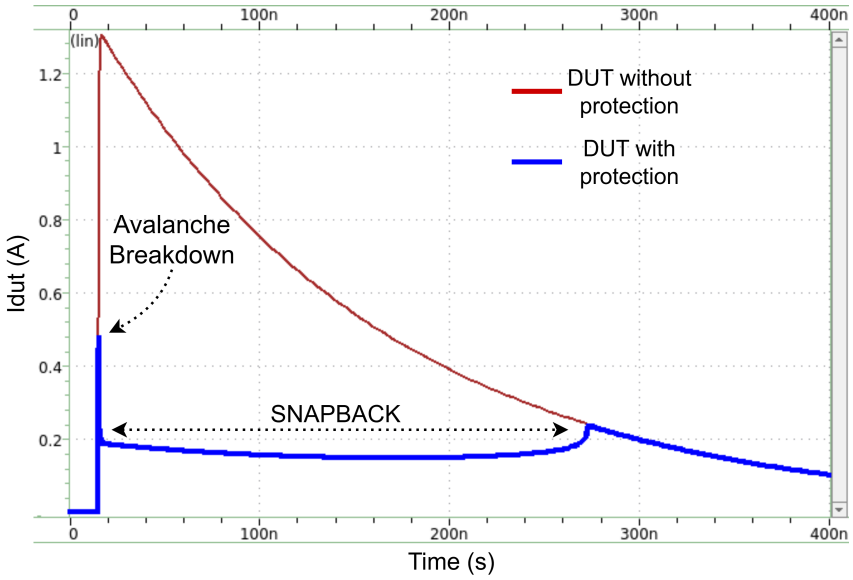


Figure 6.20: HBM test comparison for the DUT without protection and with GGNMOS protection. The protected case exhibits snapback behavior and a reduced stress level compared to the unprotected response, which evolves toward a breakdown-dominated regime.

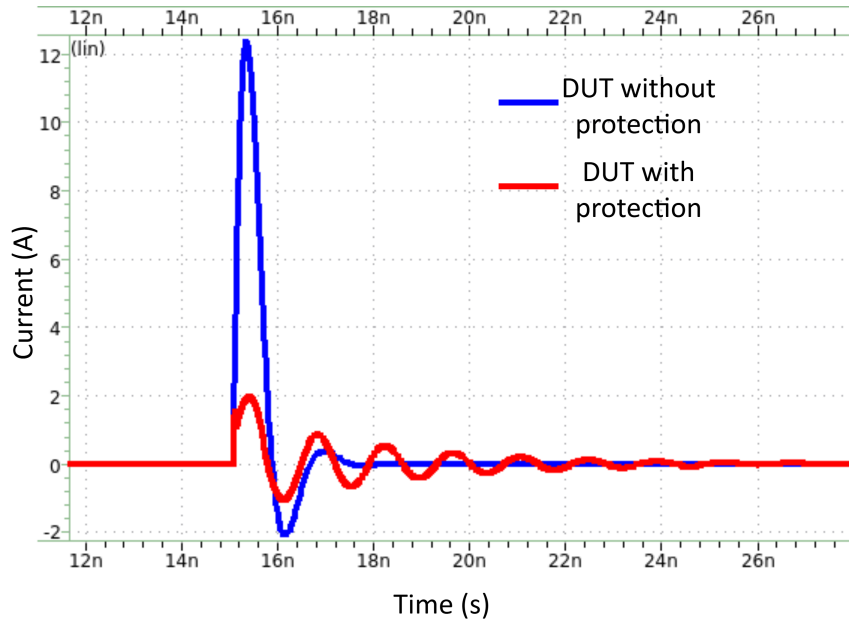


Figure 6.21: CDM test comparison at 500 V for the DUT without protection and with GGNMOS protection.

The CDM comparison in Figure 6.21 leads to a similar conclusion to the HBM case: adding the GGNMOS changes the current path at the protected node. In the unprotected case (blue), the DUT was exposed to the discharge where the current reached over 12A. In contrast, the protected case (red) shows a clear reduction of the initial current peak, indicating that the GGNMOS provides an alternative conduction path early in the event and limits the stress seen at the DUT node. This is a

useful qualification result because it shows that the calibrated macromodel does not only reproduce snapback in an I–V sense, but also reacts under a fast transient discharge.

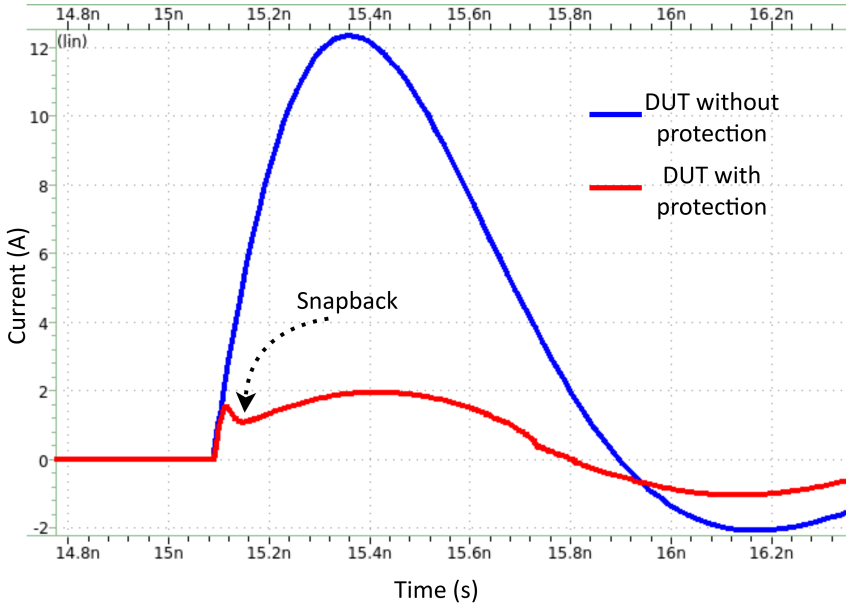


Figure 6.22: Zoom of Figure 6.21 highlighting the snapback transition in the simulation with protection.

The benefit of the CDM plot becomes more evident in the zoom of Figure 6.22. The protected waveform presents the GGNMOS triggering instant, followed by the transition into the snapback conduction phase. The visualization of this transition is important because it confirms that the macromodel can enter the low-voltage discharge state even in the faster discharge (compared with HBM). In practical terms, the zoom supports the interpretation that the clamp activation happens fast enough to prevent the DUT from remaining at the higher voltage levels observed in the unprotected response.

Overall, the three qualification tests indicate a confirmation of the calibrated GGNMOS macromodel: the current decomposition in Figure 6.19 indicates that, after triggering, the discharge is mainly sustained by the parasitic LNPN path, while the HBM and CDM comparisons (Figure 6.20 and Figure 6.21) show that, once the clamp device is added, the node becomes protected. In CDM, the zoom in Figure 6.22 makes the triggering instant and the snapback transition visible, supporting that the model also reacts under a fast discharge.

However, the result should be interpreted critically. In this work, the circuit setup is used to test the GGNMOS macromodel and its interaction with a representative load at the protected node, rather than to reproduce the complete discharge paths of a full IC and its functional core blocks. Therefore, these simulations confirm how inserting the ESD clamp affects the response at the protected node. In contrast, the ESD robustness remains outside the scope of this simplified testbench and must be verified after tapeout.

6.4 Summary

This chapter presents simulation evidence of the GGNMOS snapback behavior under TLP, HBM and CDM tests, focusing on the key design quantities $V_{\mathbf{t1}}$, $V_{\mathbf{h}}$ and $R_{\mathbf{on}}$. At device level, the TCAD results show how these landmarks shift with geometry: reducing L lowers both $V_{\mathbf{t1}}$ and $V_{\mathbf{h}}$, while increasing W raises the post-snapback current, however does not guarantee uniform distribution of current and temperature. Sweeps of TOXE, DCGS and SCGS confirm that oxide thickness and contact spacing affect triggering and the post-trigger I–V snapback shape.

The calibration results show that the SPICE macromodel reproduces the main snapback landmarks with good accuracy. The ranking-based refinement significantly reduces mismatch, improving $V_{\mathbf{h}}$ from a 0.196 V deviation to 0.006 V and $V_{\mathbf{t2}}$ from 0.510 V to 0.024 V, while keeping a small residual offset in $V_{\mathbf{t1}}$. This reflects the trade-off between matching the trigger point ($V_{\mathbf{t1}}$) and the conduction region, since the same parameters influence the overall snapback behavior.

The qualification tests verify the macromodel in a protection context. Current decomposition confirms that, after triggering, the parasitic LNPN path carries most of the discharge current, consistent with the physics of the GGNMOS. Under a simulated 2 kV HBM stress, the protected case shows a clear reduction of the current at the DUT node compared to the unprotected case. A similar effect is observed under the 500 V CDM test, where the GGNMOS reduces the initial peak and reacts correctly to the fast transient, which indicates the validation of the device's protection functionality.

Chapter 7

Conclusion

In this work, the design and validation of on chip ESD protection was investigated based on a design limitation: during the first design cycle phase, the engineers do not have a model that describes how the devices will behave under an ESD event, since the foundry PDK is primarily optimized for core circuit simulation and does not include snapback behavior or parasitic bipolar conduction. To overcome this, a methodology was developed that combines EDA tools, such as TCAD device simulation with circuit simulation in SPICE: TCAD provides a physics data reference to support parameter extraction, while the compact macromodel makes this behavior functional in SPICE. The workflow was automated with a dedicated script for snapback parameter extraction and calibration, so that protection behavior can be analyzed before tapeout, when experimental data is limited, and the same macromodel can be recalibrated after tapeout using measured I–V data.

However, two conditions make ESD protection more challenging today:

(i) as semiconductor manufacturing processes continue to evolve, parameters such as Electrical gate equivalent oxide thickness (TOXE) become increasingly smaller to optimize data processing and energy efficiency. On the other hand, this scaling makes devices more sensitive to ESD, requiring more robust protection compared with what was needed in larger-geometry technologies, and, as a result, the protection circuits end up requiring more area. (as illustrated in Figure 7.1).

(ii) as IC performance increases, ESD protection becomes progressively constrained, since adding protection devices to circuit nodes can degrade performance – especially in high-frequency pads – due to the additional impedance, capacitance, and leakage introduced by the ESD protection circuitry. HBM/CDM target levels will tend to be reduced, and the remaining risk is mitigated mainly through a better ESD controls (more details of the ESD roadmaps can be seen in [3]).

The work was intentionally focused on one protection device class to serve as a methodological base: snapback protection represented by GGNMOS. Because of time constraints and the need to keep the development coherent, other clamp families were not explored, in particular SCR based clamps. This is an important disadvantage, since the literature describes SCR devices as

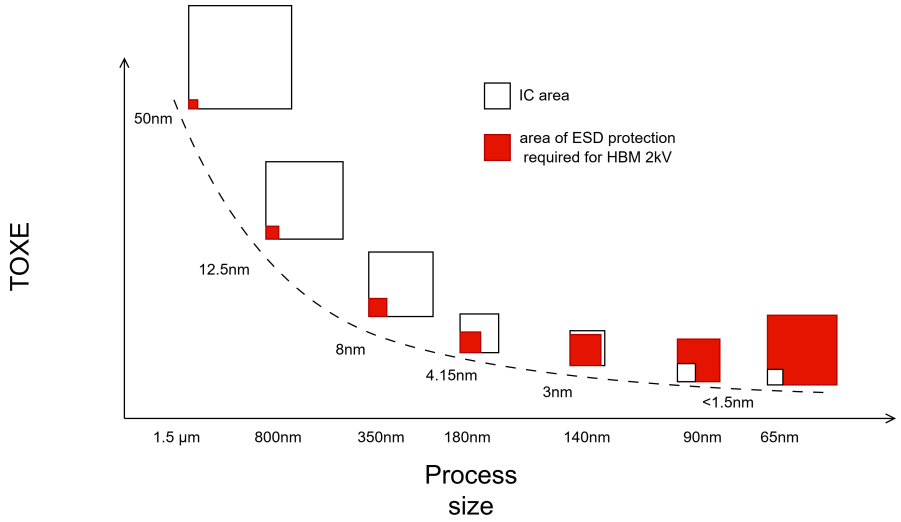


Figure 7.1: Evolution of the IC area and the ESD protection area as a function of process node and TOXE. Extracted from [18]

among the most area-efficient ESD protection structures: once triggered, they can provide a low-impedance discharge path ($R_{\mathbf{on}}$) with a relatively low clamping voltage ($V_{\mathbf{t1}}$), which reduces power dissipation during stress events [13, pp. 150–151][7, p. 220][2, p. 106]. Extending the present methodology to SCR families, e.g. Low Voltage Triggered SCR (LVTSCR), would therefore be a natural continuation, since the same core problem remains: design a compact model that captures snapback under ESD stress, and calibrate this behavior using reference data.

Another constraint present in ESD design protection is the lack of information to extract the parameters. Even when advanced simulation is available (commonly via PDK), industrial ESD validation still depends strongly on real measurement and tapeout. This dissertation states that TCAD does not replace test validation, but increases the chance of first-cycle success by avoiding purely randomized design guesses. After tapeout and testing, the measured results enable further refinement of compact models and reduce the dependence on TCAD. In addition, the SPICE verification performed in the qualification part of the work was interpreted as hypothetical, since it cannot reproduce every possible discharge path in a full protection network. This is consistent with the observation that typical industrial ESD design practice often applies limited simulation and relies on simplified EDA checks and experience procedures, rather than physical verification comparable to normal IC core design workflows [2, p. 346]. In this sense, the present contribution should be interpreted as a methodology that improves early design and also supports post-tapeout refinement, not as a claim that simulation alone can validate the protection performance.

Despite these limitations, the results indicate that the proposed methodology is effective in two perspectives: it improves understanding of ESD phenomena, reducing misunderstanding related to the test models, and it provides a practical calibration and simulation path that can be reused after tapeout.

First, the dissertation presents that ESD is too complex to be represented by just one test or one compact model, unless the results are interpreted carefully. The background discussion

highlights that real discharge events depend on many parameters and present reproducibility challenges, and that even standardized stress models are simplifications of complex physical conditions [13, pp. 12–13]. In practice, this complexity shows up as mismatches and interpretation issues between characterization methods and real ESD conditions. Hence, TLP techniques were presented as practical tools to extract I–V data and to support SEED within SOA constraints, and also as methods correlated with HBM and CDM. This perspective is important for avoiding misinterpretation: characterization data can be useful, but it must be mapped to model parameters with a clear understanding of what it does and does not capture.

Second, the work showed that implementing ESD devices in TCAD is both beneficial and complex. The benefit is that TCAD provides a framework to resolve spatial distributions and to directly investigate avalanche, substrate current generation, and parasitic bipolar activation, which are essential mechanisms for snapback operation. The complexity is that manufacturing processes are unknown and vary between foundries, making it necessary to extract as much information as possible to replicate device cross sections, and to build parameterized structures where geometry, doping, and mesh refinements can be iterated. In other words, TCAD brings physical insight, but it also demands careful calibration strategy to avoid false confidence from an unrealistic structure.

Third, the dissertation contributed to develop a portable SPICE macromodel approach and an automation script for calibration. A GGNMOS macromodel was implemented combining `BSIM4` with `VBIC` models, following a methodology that avoids explicit current sources and remains compatible with standard SPICE. The calibration framework then provided a structured method for parameter search. The adopted procedure combined an initial OVAT step (to understand the impact of any parameter), progressive refinement, and a final best-fit selection, leading to an optimum set of parameters. This is relevant because it turns calibration from a manual task into a repeatable process with traceable results, making it easier to reuse once measurements are available.

Fourth, simulation experiments indicated the efficiency of the GGNMOS. In the current decomposition analysis, the results reveal that the lateral NPN (LNPN) handled the discharge behavior during snapback, matching the expected physical behavior for GGNMOS under ESD stress. In the HBM comparison with and without GGNMOS, the simulations demonstrate how the presence of the clamp device changes the discharge path through the circuit, which reinforces that ESD behavior cannot be evaluated by analyzing the protection device alone. These experiments do not attempt to provide full-chip verification, but they indicate that the macromodel is sufficiently consistent to support circuit-level analyses within the proposed workflow.

In summary, the dissertation results show that a combined TCAD and SPICE methodology, supported by automated calibration, can improve both comprehension and System-Efficient ESD Design (SEED) for snapback protection devices. The work does not remove the need for silicon validation, but it reduces the dependence on purely experience based during the most uncertain design phase and provides a structured path for post-tapeout model refinement. Under the current constraints because of device scaling, available area, and more limited Safe Operating Area (SOA), this type of methodology is consistent with SEED and can serve as a basis for extending the approach to other clamp families, including SCR based solutions, in future work.

References

[1] T. Welsher, "The "Real" Cost of ESD Damage," en, [Online]. Available: https://www.esda.org/assets/News/RealCostofESDDamageFinal2-18-2010.pdf.

[2] A. Wang, *Practical ESD Protection Design* (Practical ESD Protection Design), English. wiley, 2021, Journal Abbreviation: Practical ESD Protection Design Pages: 417 Publication Title: Practical ESD Protection Design, ISBN: 978-111985043-4 (ISBN); 978-111985040-3 (ISBN). DOI: 10.1002/9781119850434. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85131670961&doi=10.1002%2f9781119850434&partnerID=40&md5=c2467f92032e32b799acff9699e9e1e6.

[3] A. Bong et al., "EOS/ESD Association, Inc. Technology Roadmap 2026," en, 2025. Accessed: Feb. 4, 2026. [Online]. Available: https://www.esda.org/assets/Documents/EOS-ESD-Association-Inc.-Technology-Roadmap-2026.pdf.

[4] P. Majewicz, "Compendium Special Edition on Electrostatic Discharge (ESD)," en-US, NASA Office of Safety and Mission Assurance, Washington, DC, Tech. Rep. Volume 10, Issue 2, Apr. 2019, p. 42. [Online]. Available: https://sma.nasa.gov/docs/default-source/sma-disciplines-and-programs/quality/qlf/nasa-eee-parts_esd_compendium_june-2018-sept-18.pdf?sfvrsn=551c0f8_0.

[5] A. Z. Wang, H. G. Feng, K. Gong, R. Y. Zhan, and J. Stine, "On-chip ESD protection design for integrated circuits: An overview for IC designers," *Microelectronics Journal*, vol. 32, no. 9, pp. 733–747, Sep. 2001, ISSN: 1879-2391. DOI: 10.1016/S0026-2692(01)00060-X. Accessed: Feb. 11, 2026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S002626920100060X.

[6] A. Novack, *ESD Symbols: Meaning, Types, and Uses | ESD Defender*, en-US, Feb. 2025. Accessed: Feb. 27, 2026. [Online]. Available: https://www.esdgate.com/esd-symbol/.

[7] O. Semenov, H. Sarbishaei, and M. Sachdev, *ESD Protection Device and Circuit Design for Advanced CMOS Technologies*, en. Dordrecht: Springer Netherlands, 2008, ISBN: 978-1-4020-8300-6 978-1-4020-8301-3. DOI: 10.1007/978-1-4020-8301-3. Accessed: Oct. 23, 2024. [Online]. Available: http://link.springer.com/10.1007/978-1-4020-8301-3.

[8] I. EOS/ESD Association and J. S. S. T. Association, *ANSI/ESDA/JEDEC JS-001-2024: For Electrostatic Discharge Sensitivity Testing — Human Body Model (HBM) — Device Level*, English, Limited revision of ANSI/ESDA/JEDEC JS-001-2023, Rome, NY; Arlington, VA, Oct. 2024.

[9] I. Eos/Esd Association and J. S. S. T. Association, *ANSI/ESDA/JEDEC JS-002-2022: For Electrostatic Discharge Sensitivity Testing — Charged Device Model (CDM) — Device Level*, English, Revision of ANSI/ESDA/JEDEC JS-002-2018, Rome, NY; Arlington, VA, Jul. 2022.

[10] R. Ashton, *Simulating Small Device CDM Using Spice*, en-US, Aug. 2010. Accessed: Nov. 2, 2024. [Online]. Available: https://incompliancemag.com/simulating-small-device-cdm-using-spice/.

[11] A. Olney, B. Gifford, J. Guravage, and A. Righter, "Real-world charged board model (CBM) failures," in *2003 Electrical Overstress/Electrostatic Discharge Symposium*, Sep. 2003, pp. 1–10. Accessed: Nov. 3, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/5272041/?arnumber=5272041.

[12] V. V. Kuznetsov and V. V. Andreev, "Transmission Line Pulse Setup for Electrostatic Discharge Robustness Testing of the Semiconductor Devices," en, *Instruments and Experimental Techniques*, vol. 67, no. 2, pp. 268–273, Apr. 2024, ISSN: 0020-4412, 1608-3180. DOI: 10.1134/S0020441224700453. Accessed: Jan. 13, 2025. [Online]. Available: https://link.springer.com/10.1134/S0020441224700453.

[13] A. Amerasekera and C. Duvvury, *ESD in Silicon Integrated Circuits*, en. John Wiley & Sons, Ltd, ISBN: 978-0-470-84605-6. DOI: 10.1002/0470846054. [Online]. Available: https://onlinelibrary.wiley.com/doi/book/10.1002/0470846054.

[14] S. Voldman, *The ESD Handbook* (The ESD Handbook), English. wiley, 2021, Journal Abbreviation: The ESD Handb. Pages: 1122 Publication Title: The ESD Handb., ISBN: 978-111923309-1 (ISBN); 978-111996517-6 (ISBN). DOI: 10.1002/9781119233091.

[15] H. Wolf, H. Gieser, K. Bock, A. Jahanzeb, C. Duvvury, and Y.-Y. Lin, "Capacitive coupled TLP (CC-TLP) and the correlation with the CDM," in *2009 31st EOS/ESD Symposium*, Aug. 2009, pp. 1–8. Accessed: Jan. 13, 2026. [Online]. Available: https://ieeexplore.ieee.org/document/5340132/.

[16] A. Amerasekera, S. Ramaswamy, M.-C. Chang, and C. Duvvury, "Modeling MOS snapback and parasitic bipolar action for circuit-level ESD and high current simulations," in *Proceedings of International Reliability Physics Symposium*, Apr. 1996, pp. 318–326. DOI: 10.1109/RELPHY.1996.492137. Accessed: Jul. 29, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/492137/.

[17] Y. Zhou, D. Connerney, R. Carroll, and T. Luk, "Modeling MOS snapback for circuit-level ESD simulation using BSIM3 and VBIC models," in *Sixth international symposium on quality electronic design (isqed'05)*, Mar. 2005, pp. 476–481. DOI: 10.1109/ISQED.2005.81. Accessed: Jul. 29, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/1410631/.

[18] "AN10853 ESD and EMC sensitivity of IC," en, 2010. Accessed: Jan. 13, 2026. [Online]. Available: https://www.nxp.com/docs/en/application-note/AN10853.pdf.

Appendix A

Python script for extracting V_{t1}, V_h, and V_{t2} from TCAD results

After the conclusions and bibliographical references, the text used to complete the dissertation is presented in this numbered annex.

```
import sys
import re
import argparse
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# ======================================================================
# WEIGHT CONFIGURATION
# ======================================================================
# Weights for error calculation (Priority: Vt1, Vh > Vt2)
WEIGHT_VT1 = 50.0
WEIGHT_VH  = 100.0
WEIGHT_VT2 = 1.0
WEIGHT_I   = 10.0   # Base weight for currents (It1, Ih, It2)

DV_TH_DEFAULT = 0.5 # Voltage jump threshold to ignore
 ↪  mini-snapbacks
# ======================================================================

def load_dfise_plt(path: str) -> pd.DataFrame:
    """Parse Sentaurus DF-ISE xyplot (.plt) into a pandas
     ↪  DataFrame."""
    with open(path, "r", errors="ignore") as f:
```

```
        txt = f.read()
    ds_match = re.search(r'datasets\s*=\s*\[(.*?)\]\s*functions',
    ↪  txt, re.S)
    datasets = re.findall(r'"([^"]+)"', ds_match.group(1))
    data_match = re.search(r'Data\s*\{(.*)\}\s*$', txt, re.S)
    nums = np.array([float(x) for x in
    ↪  re.findall(r'[-+]?(?:\d+\.\d*|\.\d+|\d+)(?:[Ee][+-]?\d+)?',
    ↪  data_match.group(1))], dtype=float)
    return pd.DataFrame(nums.reshape((-1, len(datasets))),
    ↪  columns=datasets)

def find_snapback_points(V: np.ndarray, I: np.ndarray, dv_th: float
↪  = 0.5):
    """Iterative temporal algorithm to identify Vt1, Vh, and
    ↪  Vt2."""
    idx_t1, idx_h, idx_t2 = None, None, None
    start_search_idx = 1

    while start_search_idx < len(V) - 1:
        local_max_idx = None
        for i in range(start_search_idx, len(V) - 1):
            if V[i-1] < V[i] and V[i] > V[i+1]:
                local_max_idx = i
                break
        if local_max_idx is None: break

        local_min_idx = None
        for k in range(local_max_idx + 1, len(V) - 1):
            if V[k-1] > V[k] and V[k] < V[k+1]:
                local_min_idx = k
                break

        if local_min_idx is not None:
            if (V[local_max_idx] - V[local_min_idx]) > dv_th:
                idx_t1, idx_h = local_max_idx, local_min_idx
                break
            else:
                start_search_idx = local_min_idx + 1
        else:
            start_search_idx = local_max_idx + 1
```

```
    if idx_t1 is None or idx_h is None: return None

    for j in range(idx_h + 1, len(V) - 1):
        if V[j-1] < V[j] and V[j] > V[j+1]:
            idx_t2 = j
            break

    return {
        'Vt1': V[idx_t1], 'It1': I[idx_t1],
        'Vh':  V[idx_h],  'Ih':  I[idx_h],
        'Vt2': V[idx_t2] if idx_t2 else V[-1],
        'It2': I[idx_t2] if idx_t2 else I[-1]
    }

def calculate_weighted_error(ref, test):
    """Calculates weighted Root Mean Square Error."""
    voltage_error = (WEIGHT_VT1 * (ref['Vt1'] - test['Vt1'])**2 +
                     WEIGHT_VH  * (ref['Vh']  - test['Vh'])**2 +
                     WEIGHT_VT2 * (ref['Vt2'] - test['Vt2'])**2)

    current_error = WEIGHT_I * ((ref['It1'] - test['It1'])**2 +
                                (ref['Ih']  - test['Ih'])**2 +
                                (ref['It2'] - test['It2'])**2)

    return np.sqrt(voltage_error + current_error)

def main():
    parser = argparse.ArgumentParser(description="GGNMOS Snapback
     ↪  Ranking Script")
    parser.add_argument("plt_ref", help="Reference .plt file from
     ↪  Sentaurus")
    parser.add_argument("csv_i", help="Current data CSV (ALTER
     ↪  columns)")
    parser.add_argument("csv_v", help="Voltage data CSV (ALTER
     ↪  columns)")
    parser.add_argument("--k", type=int, default=5, help="Number of
     ↪  top results to show (default: 5)")
    parser.add_argument("--step", type=int, default=300, help="Plot
     ↪  point density (default: 300)")
```

```
    parser.add_argument("--imax_plot", type=float, default=None,
    ↪  help="Maximum current (A) for the plot Y-axis")
    args = parser.parse_args()

    # 1. Process Reference
    df_ref = load_dfise_plt(args.plt_ref)
    V_ref, I_ref = df_ref["drain InnerVoltage"].to_numpy(),
    ↪  df_ref["drain TotalCurrent"].to_numpy()
    ref_points = find_snapback_points(V_ref, I_ref, DV_TH_DEFAULT)

    if not ref_points:
        sys.exit("Error: Snapback not detected in reference.")

    # 2. Process Test Curves
    df_i, df_v = pd.read_csv(args.csv_i, comment="#"),
    ↪  pd.read_csv(args.csv_v, comment="#")
    ranking_list = []
    alter_regex = re.compile(r"ALTER\s*=\s*(\d+)", re.I)

    for col_i in df_i.columns:
        match = alter_regex.search(col_i)
        if match:
            alter_id = match.group(1)
            col_v = next((c for c in df_v.columns if
            ↪  f"ALTER={alter_id}" in c), None)
            if col_v:
                V_raw, I_raw = pd.to_numeric(df_v[col_v],
                ↪  errors='coerce').to_numpy(),
                ↪  pd.to_numeric(df_i[col_i],
                ↪  errors='coerce').to_numpy()
                mask = np.isfinite(V_raw) & np.isfinite(I_raw)
                test_points = find_snapback_points(V_raw[mask],
                ↪  I_raw[mask], DV_TH_DEFAULT)
                if test_points:
                    ranking_list.append({
                        'id': alter_id, 'error':
                        ↪  calculate_weighted_error(ref_points,
                        ↪  test_points),
                        'V': V_raw[mask], 'I': I_raw[mask],
                        ↪  'points': test_points
```

```
                })

    ranking_list.sort(key=lambda x: x['error'])
    top_results = ranking_list[:args.k]

    # 3. Output Table
    print(f"\n{'='*75}")
    print(f"REFERENCE: Vt1={ref_points['Vt1']:.3f}V |
     ↪  Vh={ref_points['Vh']:.3f}V | Vt2={ref_points['Vt2']:.3f}V")
    print(f"{'='*75}")
    print(f"{'RANK':<5} | {'ALTER':<8} | {'ERROR':<10} | {'Vt1
     ↪  (V)':<8} | {'Vh (V)':<8} | {'Vt2 (V)':<8}")
    for i, res in enumerate(top_results):
        p = res['points']
        print(f"{i+1:<5} | {res['id']:<8} | {res['error']:<10.4f} |
         ↪  {p['Vt1']:<8.3f} | {p['Vh']:<8.3f} | {p['Vt2']:<8.3f}")

    # 4. Plotting
    plt.figure(figsize=(10, 7))

    # Apply Imax truncation for clean visualization
    def filter_data(V, I):
        if args.imax_plot is not None:
            idx = I <= args.imax_plot
            return V[idx], I[idx]
        return V, I

    V_ref_p, I_ref_p = filter_data(V_ref, I_ref)
    plt.plot(V_ref_p, I_ref_p, 'k-', linewidth=2.5,
     ↪  label='REFERENCE', zorder=5)
    plt.scatter([ref_points['Vt1'], ref_points['Vh'],
     ↪  ref_points['Vt2']],
                [ref_points['It1'], ref_points['Ih'],
                 ↪  ref_points['It2']],
                color='red', marker='v', s=120, label='Ref Points',
                 ↪  zorder=6)

    for i, res in enumerate(top_results):
        V_p, I_p = filter_data(res['V'], res['I'])
```

```
        label = f"ALTER {res['id']} (Err: {res['error']:.2f})" if i
         ↪  < 5 else None
        plt.scatter(V_p[::args.step], I_p[::args.step], s=12,
         ↪  alpha=0.4, label=label)

    if args.imax_plot:
        plt.ylim(-args.imax_plot*0.05, args.imax_plot * 1.05)

    plt.xlabel("Vd (V)")
    plt.ylabel("Id (A)")
    plt.title(f"Top {args.k} Best Fits to Reference Snapback
     ↪  Curve")
    plt.grid(True, linestyle='--', alpha=0.7)
    plt.legend(loc='best', fontsize='small', ncol=2 if args.k > 10
     ↪  else 1)
    plt.tight_layout()
    plt.show()

if __name__ == "__main__":
    main()
```

Appendix B

Candidate curve ranking for macromodel calibration using snapback reference data

```
#!/usr/bin/env python3
"""
GGNMOS Snapback Ranking Script ( reference + sweeps)

Info:
- Does NOT require "ALTER" labels in CSV headers.
- Assumes curve indexing by column position:
    Column 1 = time
    Column 2 = Alter1
    Column 3 = Alter2
    ...

- Snapback extraction:
    Vt1 by dvth filtering
    Vt2 behavior:
      * if --imax_plot is provided:
            Vt2 is defined as V where I = imax_plot (linear
             ↪ interpolation),
            searched AFTER Vh (first time it reaches the target
             ↪ current).
            If target not reached: fallback to last point.
      * else:
            Vt2 = first local maximum after Vh; if not found =>
             ↪ fallback to last point.
```

```
"""

import sys
import re
import argparse
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt


# ============================================================
# DEFAULT WEIGHT CONFIGURATION
# ============================================================
WEIGHT_VT1 = 50.0
WEIGHT_VH  = 50.0
WEIGHT_VT2 = 20.0
WEIGHT_I   = 50.0

DV_TH_DEFAULT = 0.5  # Voltage filter to ignore noises that can be
 ↪  identified as mini-snapbacks
# ============================================================


def load_dfise_plt(path: str) -> pd.DataFrame:
    """Parse Sentaurus DF-ISE xyplot (.plt) into a pandas
     ↪  DataFrame."""
    with open(path, "r", errors="ignore") as f:
        txt = f.read()

    ds_match = re.search(r'datasets\s*=\s*\[(.*?)\]\s*functions',
     ↪  txt, re.S)
    if not ds_match:
        raise ValueError("Could not find 'datasets=[...]' block in
         ↪  DF-ISE .plt file.")
    datasets = re.findall(r'"([^"]+)"', ds_match.group(1))

    data_match = re.search(r'Data\s*\{(.*)\}\s*$', txt, re.S)
    if not data_match:
        raise ValueError("Could not find 'Data { ... }' block in
         ↪  DF-ISE .plt file.")
```

```
    num_pat = r'[-+]?(?:\d+\.\d*|\.\d+|\d+)(?:[Ee][+-]?\d+)?'
    nums = np.array([float(x) for x in re.findall(num_pat,
     ↪  data_match.group(1))], dtype=float)

    ncols = len(datasets)
    if nums.size % ncols != 0:
        raise ValueError(f"Data count ({nums.size}) not multiple of
         ↪  datasets ({ncols}).")

    return pd.DataFrame(nums.reshape((-1, ncols)),
     ↪  columns=datasets)


def read_waveview_csv(path: str) -> pd.DataFrame:
    """
    Read WaveView-like CSV where:
      - line 1 is junk
      - line 2 is header
      - data starts at line 3
    -> skiprows=1 makes line 2 become the header row.

    Also supports separators with sep=None (sniffing).
    """
    return pd.read_csv(
        path,
        comment="#",
        skiprows=1,      # ignore first useless line
        header=0,        # second line becomes the header
        sep=None,        # sniff delimiter (comma/semicolon/etc.)
        engine="python"
    )


def interp_v_at_i(V: np.ndarray, I: np.ndarray, I_target: float,
 ↪  start_idx: int = 0):
    """
    Return (V_at_target, I_target, ok_interpolated).
    Finds the FIRST index k >= start_idx such that I[k] >=
     ↪  I_target.
```

```
    If I[k] == I_target -> direct.
    If I[k] > I_target and k>start_idx -> linear interpolation
     ↪  between k-1 and k.
    If target never reached -> return (V[-1], I[-1], False).
    """
    n = len(I)
    if n == 0:
        return None

    # Find first k where I[k] >= target
    k = None
    for idx in range(max(0, start_idx), n):
        if I[idx] >= I_target:
            k = idx
            break

    if k is None:
        return float(V[-1]), float(I[-1]), False  # not reached

    if I[k] == I_target:
        return float(V[k]), float(I_target), True

    if k == start_idx:
        # Can't interpolate backwards; just return that point (flag
         ↪  False)
        return float(V[k]), float(I[k]), False

    I1, I2 = float(I[k-1]), float(I[k])
    V1, V2 = float(V[k-1]), float(V[k])

    if I2 == I1:
        # Flat current segment; cannot interpolate reliably
        return float(V[k]), float(I[k]), False

    t = (I_target - I1) / (I2 - I1)
    Vt = V1 + t * (V2 - V1)
    return float(Vt), float(I_target), True


def find_snapback_points(V: np.ndarray, I: np.ndarray,
```

```
                        dv_th: float = DV_TH_DEFAULT,
                        imax_for_vt2: float | None = None):
    """
    Temporal algorithm robust to plateaus caused by
     ↪  rounding/quantization.
      1) remove NaNs/Infs
      2) compress consecutive equal-V runs (keep LAST point of each
       ↪  run)
      3) local max/min with >=/<= plus at least one strict
       ↪  inequality

    Vt2 definition:
      - if imax_for_vt2 is not None:
          Vt2 is defined as V where I = imax_for_vt2 (linear
           ↪  interpolation),
          searched AFTER Vh (first time it reaches target).
      - else:
          Vt2 is the first local maximum after Vh; if not found =>
           ↪  fallback to last point.
    """
    if V is None or I is None:
        return None

    V = np.asarray(V, dtype=float)
    I = np.asarray(I, dtype=float)

    m = np.isfinite(V) & np.isfinite(I)
    V = V[m]; I = I[m]
    if len(V) < 3:
        return None

    # --- compress consecutive duplicates in V (plateaus) ---
    eps = 0.0
    keep = np.concatenate((np.abs(np.diff(V)) > eps, [True]))
    V = V[keep]; I = I[keep]
    if len(V) < 3:
        return None

    def is_local_max(i: int) -> bool:
```

```
    return (V[i] >= V[i-1] and V[i] >= V[i+1]) and (V[i] >
     ↪  V[i-1] or V[i] > V[i+1])

def is_local_min(i: int) -> bool:
    return (V[i] <= V[i-1] and V[i] <= V[i+1]) and (V[i] <
     ↪  V[i-1] or V[i] < V[i+1])

idx_t1, idx_h = None, None
start_search_idx = 1

# --- find Vt1 and Vh ---
while start_search_idx < len(V) - 1:
    local_max_idx = None
    for i in range(start_search_idx, len(V) - 1):
        if is_local_max(i):
            local_max_idx = i
            break
    if local_max_idx is None:
        break

    local_min_idx = None
    for k in range(local_max_idx + 1, len(V) - 1):
        if is_local_min(k):
            local_min_idx = k
            break

    if local_min_idx is not None:
        if (V[local_max_idx] - V[local_min_idx]) > dv_th:
            idx_t1, idx_h = local_max_idx, local_min_idx
            break
        else:
            start_search_idx = local_min_idx + 1
    else:
        start_search_idx = local_max_idx + 1

if idx_t1 is None or idx_h is None:
    return None

# --- Vt2 logic ---
vt2_fallback = False
```

```
    vt2_mode = "localmax"

    if imax_for_vt2 is not None:
        # Define Vt2 at I = imax_for_vt2 (searched after Vh)
        Vt2, It2, ok = interp_v_at_i(V, I, imax_for_vt2,
         ↪  start_idx=idx_h)
        vt2_mode = "imax"
        if not ok:
            vt2_fallback = True
        return {
            "Vt1": float(V[idx_t1]), "It1": float(I[idx_t1]),
            "Vh":  float(V[idx_h]),  "Ih":  float(I[idx_h]),
            "Vt2": float(Vt2),       "It2": float(It2),
            "vt2_fallback": vt2_fallback,
            "vt2_mode": vt2_mode
        }

    # else: original behavior (local max after Vh)
    idx_t2 = None
    for j in range(idx_h + 1, len(V) - 1):
        if is_local_max(j):
            idx_t2 = j
            break

    if idx_t2 is None:
        idx_t2 = len(V) - 1
        vt2_fallback = True
        vt2_mode = "fallback_last"

    return {
        "Vt1": float(V[idx_t1]), "It1": float(I[idx_t1]),
        "Vh":  float(V[idx_h]),  "Ih":  float(I[idx_h]),
        "Vt2": float(V[idx_t2]), "It2": float(I[idx_t2]),
        "vt2_fallback": vt2_fallback,
        "vt2_mode": vt2_mode
    }


def calculate_weighted_error(ref, test,
```

```
                                w_vt1=WEIGHT_VT1, w_vh=WEIGHT_VH,
                                ↪  w_vt2=WEIGHT_VT2, w_i=WEIGHT_I):
    """Weighted RMSE on snapback points."""
    voltage_error = (
        w_vt1 * (ref["Vt1"] - test["Vt1"]) ** 2 +
        w_vh  * (ref["Vh"]  - test["Vh"])  ** 2 +
        w_vt2 * (ref["Vt2"] - test["Vt2"]) ** 2
    )

    current_error = w_i * (
        (ref["It1"] - test["It1"]) ** 2 +
        (ref["Ih"]  - test["Ih"])  ** 2 +
        (ref["It2"] - test["It2"]) ** 2
    )

    return float(np.sqrt(voltage_error + current_error))


def truncate_with_imax(V: np.ndarray, I: np.ndarray, imax: float):
    """
    Sequential truncation for visualization:
    return arrays ending EXACTLY at I = imax (linear
    ↪  interpolation), if reached.
    If imax never reached, returns full arrays.
    """
    V = np.asarray(V, dtype=float)
    I = np.asarray(I, dtype=float)
    if len(V) == 0:
        return V, I

    # find first k where I[k] >= imax
    k = None
    for idx in range(len(I)):
        if I[idx] >= imax:
            k = idx
            break

    if k is None:
        return V, I  # never reaches
```

```
    if I[k] == imax:
        return V[:k+1], I[:k+1]

    if k == 0:
        return V[:1], I[:1]

    # interpolate between k-1 and k
    I1, I2 = float(I[k-1]), float(I[k])
    V1, V2 = float(V[k-1]), float(V[k])

    if I2 == I1:
        return V[:k+1], I[:k+1]

    t = (imax - I1) / (I2 - I1)
    Vt = V1 + t * (V2 - V1)

    V_out = np.concatenate([V[:k], [Vt]])
    I_out = np.concatenate([I[:k], [imax]])
    return V_out, I_out


def main():
    parser = argparse.ArgumentParser(description="GGNMOS Snapback
     ↪  Ranking Script (Reference DF-ISE + CSV sweeps)")
    parser.add_argument("plt_ref", help="Reference DF-ISE .plt file
     ↪  from Sentaurus")
    parser.add_argument("csv_v", help="Voltage data CSV (WaveView
     ↪  export)")
    parser.add_argument("csv_i", help="Current data CSV (WaveView
     ↪  export)")
    parser.add_argument("--k", type=int, default=5, help="Number of
     ↪  top results to show (default: 5)")
    parser.add_argument("--step", type=int, default=300, help="Plot
     ↪  point density (default: 300)")
    parser.add_argument("--imax_plot", type=float, default=None,
                        help="Maximum current (A) for plot
                         ↪  trimming; also defines Vt2 as V where
                         ↪  I=imax_plot")
    parser.add_argument("--dvth", type=float,
     ↪  default=DV_TH_DEFAULT,
```

```
                    help=f"Snapback threshold Vt1-Vh in volts
                     ↪  (default: {DV_TH_DEFAULT})")
args = parser.parse_args()

# 1) Process Reference
df_ref = load_dfise_plt(args.plt_ref)

try:
    V_ref = df_ref["drain InnerVoltage"].to_numpy()
    I_ref = df_ref["drain TotalCurrent"].to_numpy()
except KeyError as e:
    raise SystemExit(f"Missing required DF-ISE dataset column:
     ↪  {e}")

ref_points = find_snapback_points(V_ref, I_ref, args.dvth,
 ↪  imax_for_vt2=args.imax_plot)
if not ref_points:
    sys.exit("Error: Snapback not detected in reference (no
     ↪  Vt1/Vh found).")

# 2) Read CSVs (position-based curves)
df_v = read_waveview_csv(args.csv_v)
df_i = read_waveview_csv(args.csv_i)

if df_v.shape[1] < 2 or df_i.shape[1] < 2:
    sys.exit("Error: CSV must contain at least 2 columns (time
     ↪  + at least one curve).")

# Assume first column is TIME; curves start at column index 1
v_cols = list(df_v.columns[1:])
i_cols = list(df_i.columns[1:])

n_curves = min(len(v_cols), len(i_cols))
if n_curves == 0:
    sys.exit("Error: No curve columns detected (columns after
     ↪  TIME are missing).")

ranking_list = []

for curve_idx in range(n_curves):
```

```
        alter_id = str(curve_idx + 1)  # Alter1 = col2, Alter2 =
         ↪ col3, ...

        col_v = v_cols[curve_idx]
        col_i = i_cols[curve_idx]

        V_raw = pd.to_numeric(df_v[col_v],
         ↪ errors="coerce").to_numpy()
        I_raw = pd.to_numeric(df_i[col_i],
         ↪ errors="coerce").to_numpy()

        mask = np.isfinite(V_raw) & np.isfinite(I_raw)
        if mask.sum() < 3:
            continue

        Vt = V_raw[mask]
        It = I_raw[mask]

        test_points = find_snapback_points(Vt, It, args.dvth,
         ↪ imax_for_vt2=args.imax_plot)
        if not test_points:
            continue

        ranking_list.append({
            "id": alter_id,
            "error": calculate_weighted_error(ref_points,
             ↪ test_points),
            "V": Vt,
            "I": It,
            "points": test_points
        })

    if not ranking_list:
        sys.exit("No valid curves found (or no snapback detected in
         ↪ any curve).")

    ranking_list.sort(key=lambda x: x["error"])
    top_results = ranking_list[:args.k]

    # 3) Output Table
```

```
print("\n" + "=" * 110)
ref_mode = ref_points.get("vt2_mode", "localmax")
ref_flag = " (fallback)" if ref_points.get("vt2_fallback",
 ↪  False) else ""
if args.imax_plot is not None:
    print(f"REFERENCE: Vt1={ref_points['Vt1']:.3f}V |
     ↪  Vh={ref_points['Vh']:.3f}V | "
          f"Vt2@I={args.imax_plot:g}A ->
           ↪  {ref_points['Vt2']:.3f}V{ref_flag} |
           ↪  mode={ref_mode}")
else:
    print(f"REFERENCE: Vt1={ref_points['Vt1']:.3f}V |
     ↪  Vh={ref_points['Vh']:.3f}V |
     ↪  Vt2={ref_points['Vt2']:.3f}V{ref_flag} |
     ↪  mode={ref_mode}")
print("=" * 110)
print(f"{'RANK':<5} | {'ALTER':<6} | {'ERROR':<12} | {'Vt1
 ↪  (V)':<10} | {'Vh (V)':<10} | {'Vt2 (V)':<12} |
 ↪  {'VT2_MODE':<12}")
print("-" * 110)

for i, res in enumerate(top_results, start=1):
    p = res["points"]
    mode = p.get("vt2_mode", "localmax")
    if p.get("vt2_fallback", False):
        mode += "_fallback"
    print(f"{i:<5} | {res['id']:<6} | {res['error']:<12.4f} | "
          f"{p['Vt1']:<10.3f} | {p['Vh']:<10.3f} |
           ↪  {p['Vt2']:<12.3f} | {mode:<12}")

# 4) Plotting
def filter_data_seq(V, I):
    if args.imax_plot is None:
        return V, I
    return truncate_with_imax(V, I, args.imax_plot)

plt.figure(figsize=(10, 7))

V_ref_p, I_ref_p = filter_data_seq(V_ref, I_ref)
```

```
    plt.plot(V_ref_p, I_ref_p, "k-", linewidth=2.5,
     ↪  label="REFERENCE", zorder=5)

    plt.scatter([ref_points["Vt1"], ref_points["Vh"],
     ↪  ref_points["Vt2"]],
                [ref_points["It1"], ref_points["Ih"],
                 ↪  ref_points["It2"]],
                color="red", marker="v", s=120, label="Ref Points",
                 ↪  zorder=6)

    step = max(1, args.step)
    for res in top_results:
        Vp, Ip = filter_data_seq(res["V"], res["I"])
        label = f"Alter {res['id']} (Err: {res['error']:.2f})"
        plt.plot(Vp[::step], Ip[::step], '-', linewidth=2,
         ↪  alpha=0.6, label=label)

    if args.imax_plot is not None:
        plt.ylim(-args.imax_plot * 0.05, args.imax_plot * 1.05)

    plt.xlabel("Vd (V)")
    plt.ylabel("Id (A)")
    plt.title(f"Top {args.k} Best Fits to Reference Snapback
     ↪  Curve")
    plt.grid(True, linestyle="--", alpha=0.7)
    plt.legend(loc="best", fontsize="small", ncol=2 if args.k > 10
     ↪  else 1)
    plt.tight_layout()
    plt.show()


if __name__ == "__main__":
    main()
```

Appendix C

Program to generate `ALTER` combinations

```
"""
generate_alters.py
==================

Purpose
-------
Generate an SPICE-friendly parameter sweep using .ALTER blocks and
 ↪  .PARAM lines.
Each combination of input variables becomes one ".ALTER k" section.

Usage
-----
  python3 generate_alters.py input.txt output.txt

Input file format (input.txt)
-----------------------------
One variable per line:

  <name>  <mode>  <args...>

Supported modes:
  1) LIST  v1 v2 v3 ...
     - Explicit list of values.

  2) LIN   min  max  step
```

```
    - Linear sweep from min to max inclusive (within a small
     ↪  tolerance).
    - step must be non-zero.
    - If min > max, values are generated descending (min,
     ↪  min-step, ...).

  3) GEO   min  max  factor
    - Geometric sweep from min to max inclusive (within a small
     ↪  tolerance).
    - factor must be > 1.
    - If min > max, values are generated descending by dividing
     ↪  (min, min/factor, ...).
    - "factor" can be written as "2" or "2x" (both are accepted).

Numbers:
  - Use plain decimals or scientific notation (e.g., 0.23, 1e-13,
   ↪  2.5e3).
  - Units are not interpreted (m,k,meg,f,p...); write values
   ↪  exactly as you want them in .PARAM.

Example input.txt
-----------------
  param1   GEO      500   8000   2x      # 500, 1000, 2000, 4000,
   ↪  8000
  param2   LIN      0.12  0.28   0.04    // 0.12, 0.16, 0.20, 0.24,
   ↪  0.28
  param3   LIST     9 10 11 12


Output (output.txt)
-------------------
Creates blocks like:

  ****************************************
  .ALTER 1 * rsubpar 500.0 || avc1par 0.12 || ...
  .PROT
  .PARAM rsubpar=500.0
  .PARAM avc1par=0.12
  ...
  .UNPROT
```

```
Note
----
Total number of .ALTER cases equals the product of all lists:
  N_total = N1 * N2 * ... * Nk
Warning: adding one more variable can explode the number of runs.
"""


#!/usr/bin/env python3
import itertools
import sys

def read_var_values(filename):
    """
    Read variables and their value lists from a text file.

    Each non-empty, non-comment line must be:
    var_name  val1  val2  val3 ...
    """
    vars_and_values = []

    with open(filename, "r") as f:
        for line in f:
            line = line.strip()
            if not line:
                continue  # skip empty lines
            if line.startswith("#"):
                continue  # skip comments

            parts = line.split()
            if len(parts) < 2:
                continue  # no values, ignore

            var_name = parts[0]
            values = parts[1:]  # keep as strings
            vars_and_values.append((var_name, values))

    return vars_and_values
```

```
def write_alters(vars_and_values, output_filename):
    """
    Given a list like [(var1, [v11, v12, ...]), (var2, [v21, v22,
     ↪  ...]), ...]
    write all combinations to output_filename in the required
     ↪  format.
    """
    if not vars_and_values:
        print("No variables found. Nothing to do.")
        return

    names = [name for name, _ in vars_and_values]
    value_lists = [vals for _, vals in vars_and_values]

    with open(output_filename, "w") as out:
        combo_index = 1

        for combo in itertools.product(*value_lists):
            out.write("**********************************\n")

            # Header: .ALTER N *var_name1 1 || var_name2 4 ...
            header_parts = [f"{name} {val}" for name, val in
             ↪  zip(names, combo)]
            header_line = ".ALTER {} *{}".format(
                combo_index, " || ".join(header_parts)
            )
            out.write(header_line + "\n")

            out.write(".PROT\n")
            # .PARAM lines
            for name, val in zip(names, combo):
                out.write(f".PARAM {name}={val}\n")
            out.write(".UNPROT\n")

            combo_index += 1

        out.write("**********************************\n")
```

```
def main():
    # Usage: python script.py input.txt output.txt
    if len(sys.argv) != 3:
        print("Usage: python script.py <input_file> <output_file>")
        sys.exit(1)

    input_file = sys.argv[1]
    output_file = sys.argv[2]

    vars_and_values = read_var_values(input_file)
    write_alters(vars_and_values, output_file)


if __name__ == "__main__":
    main()
```

Appendix D

SDevice TCL script for snapback extraction

```
!(
   set EQNS1  "nmos1.Poisson nmos1.Electron nmos1.Hole
    ↪  nmos1.Contact Circuit"
   set EQNS2  "nmos1.Poisson nmos1.Electron nmos1.Hole
    ↪  nmos1.Contact nmos1.Temperature Circuit "
   set INITEMP "297"
)!



Device GGNMOS {
File{
  Grid      = "@tdr@"
  Plot      = "@tdrdat@"
  Parameter = "@parameter@"
  Current   = "@plot@"

}

Electrode{

  { Name="drain"     Voltage=0.0 }
  { Name="source"    Voltage=0.0 }
  { Name="gate"      Voltage=0.0 }
  { Name="substrate" Voltage=0.0 }
```

```
}

Thermode{
          { name= "substrate" temperature= !(puts $INITEMP)!
           ↪  surfaceresistance= 0.001}
          { name= "drain"     temperature= !(puts $INITEMP)!
           ↪  surfaceresistance= 5e-3}
          { name= "source"    temperature= !(puts $INITEMP)!
           ↪  surfaceresistance= 0.001}
          { name= "gate"      temperature= !(puts $INITEMP)!
           ↪  surfaceresistance= 0.001}

}

Physics{

        EffectiveIntrinsicDensity( OldSlotboom )
        Mobility(
                DopingDependence
                HighFieldSaturation
                Enormal
        )
        Recombination(
                SRH(DopingDependence TempDependence)
                Auger
                Avalanche(Okuto)

        )

        Thermodynamic
        AnalyticTEP
        AreaFactor= @Wgate@
}

}

File {
   Output= "@log@"
}
```

```
*CurrentPlot {
*  ImpactIonization(Maximum(Semiconductor Coordinates))
*  ElectricField   (Maximum(Semiconductor Coordinates))
*}




Plot{

*--Density and Currents, etc
   *eDensity hDensity
   *TotalCurrent/Vector eCurrent/Vector hCurrent/Vector
   *eMobility hMobility
   *eVelocity hVelocity
   *eQuasiFermi hQuasiFermi

   LatticeTemperature Temperature # eTemperature hTemperature
   *TotalHeat RecombinationHeat hJouleHeat eJouleHeat

*--Fields and charges
   Potential **ElectricField/Vector SpaceCharge

*--Doping Profiles
   Doping ***DonorConcentration AcceptorConcentration

*--Generation/Recombination
   SRH Auger * Band2Band
   AvalancheGeneration eAvalancheGeneration hAvalancheGeneration

*--Driving forces
   ***eGradQuasiFermi/Vector hGradQuasiFermi/Vector
   ***eEparallel hEparallel eENormal hENormal

*--Band structure/Composition
   ***BandGap
   ***BandGapNarrowing
   * Affinity
```

```
   ***ConductionBand ValenceBand
   * eQuantumPotential hQuantumPotential
   ***eQuantumPotential
*--Gate Tunneling
   * eBarrierTunneling hBarrierTunneling  BarrierTunneling
   * eDirectTunnel hDirectTunnel
}

Math {

   Method= ILS
   Transient= BE
   Digits=6
   DrForceRefDens= 1e12

   Extrapolate
   Notdamped= 50
   Iterations= 15
   ErrRef(electron) = 1e8
   ErrRef(hole)     = 1e8
   RhsFactor    = 1e18
   ExtendedPrecision
   Number_Of_Threads= maximum
   NoCheckTransientError
   ElementAvalancheMinAngle= 0


   BreakCriteria {
      Current (Contact= "drain" maxval= 1.5)
      LatticeTemperature (maxval= 1687)
   }
}

System{

  Vsource_pset vesd (nesd 0) {
    pulse = (0.0        # dc
             @vtlp@        # amplitude
             10e-12     # td
             10e-9     # tr
```

```
              10e-9     # tf
              100e-9     # ton
             200e-9)    # period
  }

  GGNMOS nmos1 ( "source"=0  "drain"=ndut "gate"=0 "substrate"=0 )

  Resistor_pset rtest ( nesd ndut ){ resistance = 50 }

  Plot "n@node@_sys_des.plt" (time() v(ndut)
                              i(rtest,ndut))
}

Plot{
*--Density and Currents, etc
   *****eDensity hDensity
   *****TotalCurrent/Vector eCurrent/Vector hCurrent/Vector
   * eMobility hMobility
   * eVelocity hVelocity
   *****eQuasiFermi hQuasiFermi

*--Temperature
   * eTemperature Temperature hTemperature
   LatticeTemperature *****Temperature

*--Fields and charges
   Potential *****ElectricField/Vector  SpaceCharge

*--Doping Profiles
   Doping *DonorConcentration AcceptorConcentration

*--Generation/Recombination
   *****SRH Band2Band Auger
   *****AvalancheGeneration eAvalancheGeneration
    ↪  hAvalancheGeneration

*--Driving forces
   ***eGradQuasiFermi/Vector hGradQuasiFermi/Vector
   * eEparallel hEparallel eENormal hENormal
```

```
*--Band structure/Composition
   *****BandGap
   * BandGapNarrowing
   * Affinity
   *****ConductionBand ValenceBand
   * eQuantumPotential hQuantumPotential

*--Gate Tunneling
   * eBarrierTunneling hBarrierTunneling  BarrierTunneling
   * eDirectTunnel hDirectTunnel
}


Solve {
*- Creating initial guess:
   Poisson
   Coupled{ Poisson Electron Hole Temperature}

   Transient (
      InitialTime= 0.
      FinalTime= 11e-9
      Increment= 1.35
      Decrement= 2.1
      InitialStep= 1e-10
      MinStep= 1e-15
      MaxStep= 0.1
      Plot { range= (0,       10e-9) intervals= 50 }
   ){  Coupled { !(puts $EQNS2)! }
   }


}
```

Appendix E

Parameterized two-dimensional (2D) GGNMOS geometry in SDE

```
(sdegeo:set-default-boolean "ABA")

; --------------------------------------------------------------------
; Setting parameters
; - lateral
(define Lg    @Lgate@)              ; [um] Gate length
(define Lsd   0.23)                 ; [um] S/D length
(define Lsp   Lsd)                  ; [um] Spacer length
(define Ldcgs @dcgs@)               ; [um] Drain-contact to gate
 ↪  spacing
(define Lscgs @scgs@)               ; [um] Source-contact to gate
 ↪  spacing


(define L0 0)
(define L1 Lsd)
(define L2 (+ Lsd Ldcgs))
(define L3 (- L2 Lsp))
(define L4 (+ L2 Lg))
(define L5 (+ L4 Lsp))
(define L6 (+ L4 Lscgs))
(define L7 (+ L6 Lsd))

; - layers
(define Hsub (* 20 @Xj@))                   ; [um] Substrate
 ↪  thickness
```

```
(define Tox  (* -1 @Tox@))                    ; [um] Gate oxide
 ↪  thickness
(define Hpol (* -1 @Hpoly@))                  ; [um] Poly gate
 ↪  thickness


; - other
; - spacer rounding
(define fillet-radius 0.08)                    ; [um] Rounding radius

; - pn junction resolution
(define Gpn 0.001)                            ; [um]

; - Substrate doping level
(define Nsub @Ndep@)                           ; [1/cm3]
(define Nsd @Nsd@)                             ; [1/cm3] S/D active
 ↪  doping

(define Xj  @Xj@)                              ; [um] Junction depth
(define Xject (* 0.4 Xj))                     ; [um] Extension
 ↪  junction depth

; ------------------------------------------------------------------
;Create regions

(sdegeo:create-rectangle (position L0 Hsub 0)
                         (position L7 0 0) "Silicon" "R.Silicon")

(sdegeo:create-rectangle (position L3 0 0)
                         (position L5 Hpol 0) "Oxide" "R.GateOxi")

(sdegeo:create-rectangle (position L3 Hpol 0)
                         (position (+ L2 1e-2) Tox 0) "Si3N4"
                          ↪  "R.SpacerD")

(sdegeo:create-rectangle (position L5 Hpol 0)
                         (position (- L4 1e-2) Tox 0) "Si3N4"
                          ↪  "R.SpacerS")

(sdegeo:create-rectangle (position L2 Hpol 0)
```

```
                        (position L4 Tox 0) "PolySi" "R.Gate")


; - spacer rounding
(define fillet-radius 0.08)          ; [um] Rounding radius

(sdegeo:fillet-2d (list (car (find-vertex-id
                (position L3 Hpol 0)))) fillet-radius)
(sdegeo:fillet-2d (list (car (find-vertex-id (position
                L5 Hpol 0)))) fillet-radius)
; -------------------------------------------------------------------
; Contact

(sdegeo:define-contact-set "substrate" 1  (color:rgb 1 0 1 ) "##")
(sdegeo:define-contact-set "drain" 1  (color:rgb 1 0 1 ) "##")
(sdegeo:define-contact-set "source" 1  (color:rgb 1 0 1 ) "##")
(sdegeo:define-contact-set "gate" 1  (color:rgb 1 0 1 ) "##")



(sdegeo:insert-vertex (position L1 0 0.0))
(sdegeo:insert-vertex (position L6 0 0.0))

(sdegeo:define-2d-contact (list (car (find-edge-id (position 0.1
↪  Hsub 0)))) "substrate")
(sdegeo:define-2d-contact (list (car (find-edge-id (position 0 0
↪  0)))) "drain")
(sdegeo:define-2d-contact (list (car (find-edge-id (position (- L7
↪  0.1) 0 0)))) "source")
(sdegeo:define-2d-contact (list (car (find-edge-id (position (+ L2
↪  0.01) Hpol 0)))) "gate")



; -------------------------------------------------------------------
; Create doping

; Define analytic doping profiles
(sdedr:define-refeval-window "BaseLine.Drain"
  "Line" (position (* L2 -2) 0.0 0.0) (position L3 0.0 0.0))
```

```
(sdedr:define-refeval-window "BaseLine.Source"
  "Line" (position  L5 0.0 0.0) (position  (+ L7 (- L7 L4)) 0.0
   ↪  0.0))
(sdedr:define-refeval-window "BaseLine.DrainExt"
  "Line" (position (* L2 -2) 0.0 0.0) (position L2 0.0 0.0))
(sdedr:define-refeval-window "BaseLine.SourceExt"
  "Line" (position  L4 0.0 0.0) (position  (+ L7 (- L7 L4)) 0.0
   ↪  0.0))


(sdedr:define-constant-profile "Const.Silicon"
 ↪  "BoronActiveConcentration" Nsub)
(sdedr:define-constant-profile-material "PlaceCD.Substrate"
 ↪  "Const.Silicon" "Silicon")

(sdedr:define-constant-profile "Const.Gate"
 ↪  "ArsenicActiveConcentration" 1e20)
(sdedr:define-constant-profile-material "PlaceCD.Gate" "Const.Gate"
 ↪  "PolySi")

(sdedr:define-gaussian-profile "Gauss.SourceDrain"
  "ArsenicActiveConcentration" "PeakPos" 0.0 "PeakVal" Nsd
  "ValueAtDepth" Nsub "Depth" Xj "Gauss" "Factor" 0.4)

(sdedr:define-analytical-profile-placement "PlaceAP.Source"
  "Gauss.SourceDrain" "BaseLine.Source" "Positive" "NoReplace"
   ↪  "Eval")

(sdedr:define-analytical-profile-placement "PlaceAP.Drain"
  "Gauss.SourceDrain" "BaseLine.Drain" "Positive" "NoReplace"
   ↪  "Eval")

(sdedr:define-gaussian-profile "Gauss.SourceDrainExt"
  "ArsenicActiveConcentration" "PeakPos" 0.0 "PeakVal" Nsd
  "ValueAtDepth" Nsub "Depth" Xject "Gauss" "Factor" 0.25)

(sdedr:define-analytical-profile-placement "PlaceAP.SourceExt"
  "Gauss.SourceDrainExt" "BaseLine.SourceExt" "Positive"
   ↪  "NoReplace" "Eval")
```

```
(sdedr:define-analytical-profile-placement "PlaceAP.DrainExt"
  "Gauss.SourceDrainExt" "BaseLine.DrainExt" "Positive" "NoReplace"
   ↪  "Eval")

; -------------------------------------------------------------------
; Meshing Strategy (adapted from the reference script)

; ---- 0) Helper magnitudes (because Hpol and Tox are negative in
 ↪  this script)
(define Hpol_abs (abs Hpol))
(define Tox_abs  (abs Tox))

; ---- 1) Global silicon refinement (coarse base + doping-gradient
 ↪  control)
(sdedr:define-refinement-size "Ref.SiBase"
  (/ L7 6.0)      (/ Hsub 10.0)      ; max dx, max dy
  Gpn             Gpn                ; min dx, min dy
)
(sdedr:define-refinement-function "Ref.SiBase"
  "DopingConcentration" "MaxTransDiff" 1
)
(sdedr:define-refinement-region "RefPlace.SiBase"
  "Ref.SiBase" "R.Silicon"
)

; ---- 2) Active top-silicon window (captures junctions +
 ↪  high-field region)
(sdedr:define-refeval-window "RWin.Act" "Rectangle"
  (position 0.0  0.0       0.0)
  (position L7   (* 1.2 Xj) 0.0)
)
(sdedr:define-refinement-size "Ref.SiAct"
  (/ L7 10.0)     (/ Xj 8.0)         ; max dx, max dy
  (/ Lg 40.0)     (/ Xj 30.0)        ; min dx, min dy (tighter than
   ↪  base)
)
(sdedr:define-refinement-function "Ref.SiAct"
  "DopingConcentration" "MaxTransDiff" 1
)
(sdedr:define-refinement-placement "RefPlace.SiAct"
```

```
  "Ref.SiAct" "RWin.Act"
)

; ---- 3) Gate oxide region refinement (very fine vertical
 ↪  resolution)

(sdedr:define-refinement-size "Ref.GOX"
  (/ Lg 6.0)          (/ Hpol_abs 2.0)      ; max dx, max dy
  (/ Lg 40.0)         (/ Hpol_abs 10.0)     ; min dx, min dy
)
(sdedr:define-refinement-region "RefPlace.GOX"
  "Ref.GOX" "R.GateOxi"
)

; ---- 4) Multibox around the gate stack (poly + spacer
 ↪  neighborhood)
(sdedr:define-refeval-window "MBWin.Gate" "Rectangle"
  (position (- L2 (* 0.5 Lsp))  Hpol 0.0)
  (position (+ L4 (* 0.5 Lsp))  Tox  0.0)
)
(sdedr:define-multibox-size "MBSize.Gate"
  (/ Lg  6.0)    (/ Hpol_abs 4.0)     ; outer max dx, dy
  (/ Lg 30.0)    (/ Hpol_abs 20.0)    ; inner min dx, dy
  1.0            -1.35                ; grading (keep as in
   ↪  reference)
)
(sdedr:define-multibox-placement "MBPlace.Gate"
  "MBSize.Gate" "MBWin.Gate"
)

; ---- 5) Multibox in the channel (under the gate in silicon)
(sdedr:define-refeval-window "MBWin.Channel" "Rectangle"
  (position L2    0.0          0.0)
  (position L4    (* 0.5 Xj)   0.0)
)
(sdedr:define-multibox-size "MBSize.Channel"
  (/ Lg  6.0)    (/ Xj 6.0)           ; outer
  (/ Lg 30.0)    (/ Xj 30.0)          ; inner (tight)
  1.0            1.35
)
```

```
(sedr:define-multibox-placement "MBPlace.Channel"
  "MBSize.Channel" "MBWin.Channel"
)

; ---- 6) Extra refinement at gate-edge junctions (GD and GS edges)

(sedr:define-refinement-size "Ref.EdgeJ"
  (/ Lg 20.0)   99
  (/ Lg 60.0)   66
)

(sedr:define-refeval-window "RWin.GD" "Rectangle"
  (position (- L2 0.01)       0.0        0.0)
  (position (+ L3 0.01)       (* 0.25 Xj) 0.0)
)
(sedr:define-refinement-placement "RefPlace.GD"
  "Ref.EdgeJ" "RWin.GD"
)

; Source-side gate edge neighborhood (around L4/L5)
(sedr:define-refeval-window "RWin.GS" "Rectangle"
  (position (- L5 0.01)       0.0        0.0)
  (position (+ L4 0.01)       (* 0.25 Xj) 0.0)
)
(sedr:define-refinement-placement "RefPlace.GS"
  "Ref.EdgeJ" "RWin.GS"
)

(sedr:define-refinement-function "Ref.SiBase"
  "MaxLenInt" "Silicon" "Oxide"  0.0002 1.5 "DoubleSide"
)









(sdeio:save-tdr-bnd(get-body-list)"n@node@_bnd.tdr")
(sde:save-model"/home/sentaurus/STDB/ESD/GGNMOSv4")
(sedr:write-cmd-file"n@node@_msh.cmd")
```

```
(system:command"noffset3d n@node@_msh")
(sde:build-mesh "mesh""-s""n@node@_msh")
```

Appendix F

Parameterized three-dimensional (3D) GGNMOS geometry in SDE

```
(sde:clear)
(sdegeo:set-default-boolean "ABA")

;-----------------------------------------------------------------
; Setting parameters
; - lateral
(define Lg    @Lgate@)                ; [um] Gate length
(define Lsd   0.08)                   ; [um] S/D contact length
(define Lsp   0.05)                   ; [um] Spacer length
(define Ldcgs @dcgs@)
(define Lscgs @scgs@)
(define Wg @Wgate@)

(define Wtren 0.0)
;(define Htren (* -1 0.2))
;(define Ltren 0.1)


; - heigh layers
(define Xj @xj@)
(define Hsub (* -10 Xj))              ; [um] Substrate thickness
(define Tox  @tox@)                   ; [um] Gate oxide thickness
(define Hpol @Hpoly@)                 ; [um] Poly gate thickness
```

```
(define L0 0)                                   ; Starting
 ↪  point
;(define L1 Ltren)
(define L2 Lsd)                                  ; End of drain
 ↪  contact
(define L3 (+ L2 Ldcgs))                  ; Starting Gate
(define L4 (- L3 Lsp))                          ; Starting Drain
 ↪  spacer
(define L5 (+ L3 Lg))                           ; End of Gate
(define L6 (+ L5 Lsp))                           ; Finishing Source
 ↪  spacer
(define L7 (+ L5 Lscgs))                  ; Starting source contact
(define L8 (+ L7 Lsd))                          ; End of device
;(define L9 (- L8 Ltren))


(define W0 0)
(define W1 (+ W0 Wtren))
(define W2 (+ W1 Wg))
(define W3 (+ W2 Wtren))




; - other
; - spacer rounding
(define fillet-radius 0.01)          ; [um] Rounding radius

; - pn junction resolution
(define Gpn 0.001)                   ; [um]

; - Substrate doping level
(define Nsub @Ndep@)                 ; [1/cm3]
(define Nsd @Nsd@)                   ; [1/cm3] S/D active doping

(define Xject (* 0.4 Xj))            ; [um] Extension junction depth

; -------------------------------------------------------------
```

```
; Creating cuboid regions
(sdegeo:create-cuboid (position L0 W0 0.0)
                      (position L8 W3 Hsub)
        "Silicon" "SubsSilicon")




(sdegeo:create-cuboid (position L3 W1 0)
                      (position L5 W2 Tox)
        "Oxide" "GateOxide")



;(sdegeo:create-cuboid (position L1 W2 0) (position L9 W3 Htren)
;"Oxide" "TrenchOxide_Right")


;(sdegeo:create-cuboid (position L1  W0 0) (position L9  W1 Htren)
; "Oxide" "TrenchOxide_Left")

(sdegeo:create-cuboid (position L3 W1 Tox)
                      (position L5 W2 Hpol)
        "PolySi" "PolyGate")

; Poly Reoxidation
(sdegeo:set-default-boolean "BAB")
(sdegeo:create-cuboid (position (- L3 0.003) W1 0)
                      (position (+ L5 0.003) W2 Hpol)
        "Oxide" "PolyReOxide1")

(sdegeo:create-cuboid (position L4 W1 0)
                      (position L6 W2 0.005)
        "Oxide" "PolyReOxide2")

#; Ni Spacers
(sdegeo:create-cuboid (position L4 W1 0)
```

```
                        (position L6 W2 Hpol)
        "Nitride" "NiSpacer")

;(sde:define-parameter "fillet-radius" 0.1 0.0 0.0 )

(sdegeo:fillet
  (list (car (find-edge-id (position L4 (+ W1 0.01) Hpol)))
        (car (find-edge-id (position L6 (+ W1 0.01) Hpol)))
  ) fillet-radius)

; Contact definitions
(sdegeo:set-contact (find-face-id (position 0.1 0.1 Hsub))
        "substrate")
(sdegeo:set-contact (find-face-id
                        (position (+ L3 0.1) (+ W1 0.1) Hpol))
                "gate")

(define SOURCE
        (sdegeo:create-cuboid (position L7 W1 0)
                (position L8  W2 0.05)
        "Metal" "Source"))
(sdegeo:set-contact SOURCE "source" "remove")

(define DRAIN
        (sdegeo:create-cuboid (position  L0  W1 0)
                (position  L2 W2 0.05)
        "Metal" "Drain"))
(sdegeo:set-contact DRAIN "drain" "remove")

; Constant doping profile definions
(sdedr:define-constant-profile "Const.Bulk"
        "BoronActiveConcentration" 1e17)
(sdedr:define-constant-profile-region "PlaceCD.Bulk"
  "Const.Bulk" "SubsSilicon")

(sdedr:define-constant-profile "Const.Poly"
        "ArsenicActiveConcentration" 1e20)
(sdedr:define-constant-profile-region "PlaceCD.Poly"
  "Const.Poly" "PolyGate")
```

```
; Analytic doping profile definions
(sedr:define-refeval-window "BaseLine.Drain" "Rectangle"
 (position  (- L0 0.2) (- W0 0.05) 0.0)
 (position L4 (+ W3 0.05) 0.0) )

(sedr:define-refeval-window "BaseLine.Source" "Rectangle"
 (position (+ L8 0.2) (- W0 0.05) 0.0)
 (position L6 (+ W3 0.05)  0.0) )

(sedr:define-gaussian-profile "Gauss.SourceDrain"
  "ArsenicActiveConcentration" "PeakPos" 0.0 "PeakVal" 1e19
  "ValueAtDepth" 1e17 "Depth" 0.1 "Gauss" "Factor" 0.8)

(sedr:define-analytical-profile-placement "PlaceAP.Source"
 "Gauss.SourceDrain" "BaseLine.Source" "Both" "NoReplace" "Eval")

(sedr:define-analytical-profile-placement "PlaceAP.Drain"
 "Gauss.SourceDrain" "BaseLine.Drain" "Both" "NoReplace" "Eval")

; Meshing strategies
(sedr:define-refeval-window "RefWin.Global" "Cuboid"
  (position L0 W0 Hsub) (position L8 W3 Hpol))

(sedr:define-refinement-size "RefDef.Global"
  (* 0.2 Lg)  (* 0.2 Wg)  (* -0.1 Hsub)
  (* 0.1 Lg) (* 0.1 Wg) (* -0.05 Hsub))

(sedr:define-refinement-placement "Place.Global"
 "RefDef.Global" "RefWin.Global" )

(sedr:define-refeval-window "RefWin.Active" "Cuboid"
 (position L0 W0 (- Hsub 0.2)) (position L8 W3 0.0))

(sedr:define-refinement-size "RefDef.Active"
  (* 0.2 Lg)  (* 0.2 Wg)  (* -0.1 Hsub)
  (* 0.01 Lg) (* 0.01 Wg) (* -0.005 Hsub) )

(sedr:define-refinement-placement "Place.Active"
"RefDef.Active" "RefWin.Active")
```

```
; Multiboxes
(sdedr:define-refeval-window "RefWin.Channel"
 "Cuboid" (position (- L3 0.02) W1 -0.2)
          (position (+ L5 0.02) W2 0))

(sdedr:define-multibox-size "RefDefMB.Channel"
 (* 0.05 Lg)   (* 0.05 Wg)   (* -0.01 Hsub)                  ; max
 (* 0.025 Lg) (* 0.025 Wg) (* -0.001 Hsub)                   ; min
 1     1    -1.5 )                                          ; ratio

(sdedr:define-multibox-placement "PlaceMB.Channel"
  "RefDefMB.Channel" "RefWin.Channel" )

; Meshing structure
(sde:build-mesh "n@node@")
```