Reliability and Modeling of 32nm SOI Transistors at Cryogenic Temperatures

by

William Ryan King

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
August 6, 2016

Keywords: electronics, cryogenic, reliability, hot carrier effects

Approved by

Michael C. Hamilton, Chair, Associate Professor of Electrical and Computer Engineering
Robert N. Dean, McWane Professor of Electrical and Computer Engineering
Mark L. Adams, Assistant Professor of Electrical and Computer Engineering

Abstract

With conventional transistor scaling rules and performance trends coming to an end, new methods are being explored to continue the advancements of electronic devices. One such method is to operate the devices in cryogenic environments. A few concerns for cryogenic electronics are the device reliability and lack of cryogenic models. Integrated circuits fabricated using a 32 nm SOI technology containing MOSFET transistors are stressed and measured at room temperature and in liquid helium to determine how the characteristics of the transistors change as a function of temperature. Various high voltage bias are applied to the gate and drain of an nFET transistor to induce hot carrier effects in the gate oxide of the transistor. Different gate widths are being tested to understand the geometrical impacts on reliability. Using the transistor characteristics gathered from the different widths, scalable cryogenic SPICE models are developed.

Acknowledgments

I sincerely thank Dr. Michael Hamilton for serving as my advisor and for his guidance and expertise throughout this work and graduate studies.

I express my thanks Dr. Robert Dean and Dr.Mark Adams for their review and commentary on this work.

I would also like to thank Dr. Charles Ellis, Kyle Owen, Drew Sellers, and Hannah Masten for their support in areas of this work.

I wish to express my most sincere gratitude to Ryan and Cindy King for their encouragement and support throughout my academic career.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

DRC	Design Rule Check
GDS	Graphic Data System
HCE	Hot Carrier Effect
HVT	High Voltage Threshold
IC	Integrated Circuit
IC-CAP	Integrated Circuit Characterization and Analysis Program
KITE	Keithley interactive test environment
LHe	Liquid Helium
LM	Levenberg-Marquardt
LVS	Layout Versus Schematic
MTTF	Mean Time to Failure
PDK	Process Design Kit
RT	Room Temperature
RVT	Regular Voltage Threshold
SMU	Source Measurement Unit
SOI	Silicon on Insulator
SVT	Super-high Voltage Threshold

Chapter 1

Introduction

“If the auto industry advanced as rapidly as the semiconductor industry, a Rolls Royce would get a half a million miles to the gallon and it would be cheaper to throw it away than to park it.” Gordon Moore [1]

Gordon Moore, co-founder of Intel, developed a theory in 1965 stating that the number of transistors that would fit on a square inch of an integrated circuit would double every year and the price would halve [1]. This theory would prove to hold true over several decades that followed the statement. However, after approximately 50 years of being the industry standard for transistor scaling trends, Moores Law has approached the physical limit for improvement. Moores Law coming to an end is requiring scientists and engineers to explore new ways to continue the advancement of transistors. One such way that this is being accomplished is by using the transistors in cryogenic environments.

Cryogenic electronics are designed to operate at temperatures as low as a few millikelvin. Electric systems that operate at these temperatures are commonly called *cryoelectronics*. The use of cryoelectronics has many benefits in terms of circuit performance. A general improvement of cryoelectronics are that conductors, such as niobium and lead, transition into superconductors at sub-10K temperatures [2]. Superconductors have essentially no resistance when at cryogenic temperatures allowing current to flow with approximately zero loss in the conductor. Transistors, when used at cryogenic temperatures, exhibit reduced noise, superior speed, and increased efficiency when compared to RT results. These transistor improvements, when combined with superconducting connections, are enabling cryoelectronic systems to be developed with superior performance and efficiency compared to the same system at RT.

One system that requires the use of cryoelectronics is the quantum computer. Quantum computers use qubits instead of traditional bits that are found in todays computer, allowing the quantum computer to achieve processing speeds hundreds or thousands of times faster than a traditional computer. One downside of the quantum computer is that currently they do not work at RT. Most qubits must be kept at temperatures that are lower than 1K to be able to operate. Because the qubits must be at cryogenic temperatures to operate, cryoelectronic circuits must be used to interface between the qubits and the external inputs and outputs. Two issues with developing these cryoelectronic circuits are the reliability of the transistors at cryogenic temperatures and the lack of having cryogenic models of the transistors.

The following chapters will be comprised of: an overview of the IC design process, a reliability comparison of HCE on a MOSFET at RT and in LHe, and the process of developing a cryogenic model for a MOSFET transistor.

Chapter 2
IC Fabrication Design

This chapter will give an overview of the IC design process using Cadence Virtuoso. Cadence Virtuoso is a custom IC design tool that encompasses the schematic diagram, simulation, and layout all into one software package. The main steps of designing a transistor test chip will be highlighted, and the completed designs used for all of the modeling and reliability test performed in this study will be discussed. Refer to Appendix A for a more detailed tutorial on using Cadence Virtuoso for transistor simulations.

2.1 Design Steps

The entire IC design process can be grouped into four sequential steps:

1. Schematic Design
2. Simulation
3. Layout
4. Checking and Submission

2.1.1 Schematic Design

First, a schematic of the design must be made using component symbols (Figure 2.1). All of the models that are available can be found in the Model Reference Guide of the PDK.

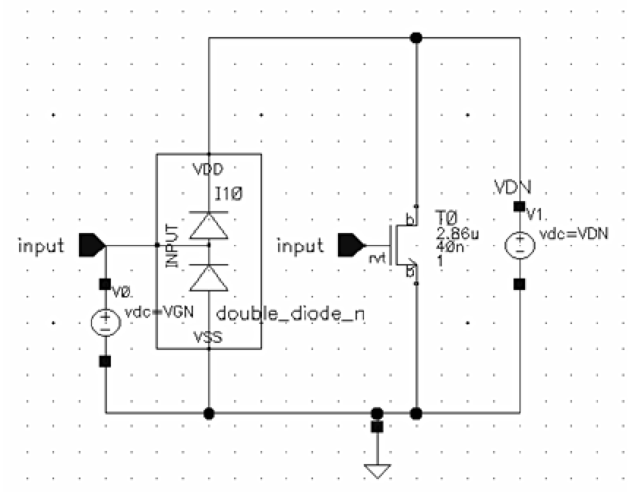


Figure 2.1: Schematic diagram of a single transistor with ESD diodes used to run simulations and layout the design.

2.1.2 Simulation

When the schematic design is complete, simulations are performed on the design to ensure that the circuit is functional and gives a desired result (Figure 2.2). During the simulations, corner simulations are useful to help account for imperfections in the materials and variation of the manufacturing process. FFF and SSF are the best and worst case design manual value limits of the model. FF and SS account for increased and decreased mobility due to chip fabrication imperfections and chip-to-chip connections. By looking at the log scale plot, it is easily seen that results are not compatible for the cryogenic measurements. The models associated with the component symbols in the PDK were developed from RT devices. This issue will be covered in more detail in Chapter 5.

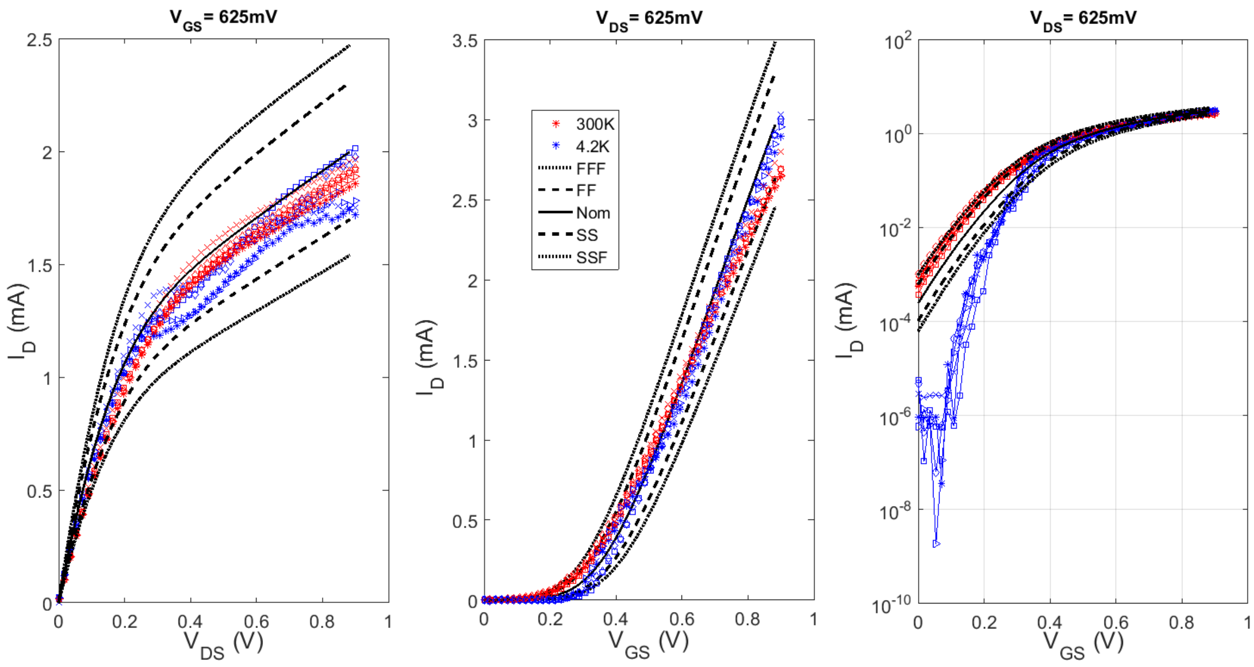


Figure 2.2: Corner simulation results from the 2860nm transistor compared to the measured results at 300K and 4.2K.

2.1.3 Layout

Once the simulations are producing desired results, the chip layout development can begin. Each PDK comes with a design manual for the process. This manual has detailed information about most aspects of the layout design including: metal stack, standard cells, rules, and various other topics. All of the transistors used in all of these designs were included in these standard cells of the PDK. For the test chips, all of routing was done on wide, thick, metal layers to help reduce the resistance of the chip.

2.1.4 Checking and Submission

After the layout has been completed, two checks must be performed before the design can be submitted for fabrication. The first check the design must pass is DRC. DRC is an automated checking feature that compares the physical layout to a series of parameters established by the design manual of the fabrication process. The DRCs main purpose is to make certain that the design will be able to be fabricated. LVS is the final check that must be executed on the design. LVS takes the schematic that was created in the initial step of the design process and compares the electrical connections to the layout. This ensures that there are not any wires crossing, causing shorts, and all of the interconnections formed match the previously simulated schematic. Finally, after both of the design checks are clean, the layout GDS File can be streamed out and submitted to the foundry.

2.2 Test Chips

There were two different chips layouts used to perform all of the measurements included in this study, a 2x2mm and two 1x1mm. Both of these chips were fabricated using a 32nm SOI process. Figure 2.3 and Figure 2.4 show an overview of the 2x2 and 1x1 chips, respectively. By taking a closer look, Figure 2.5 gives a more detailed view of how an individual transistor looks in the layout of the schematic shown in Figure 2.1. While the ESD diodes were primarily added because they are required by the design rule manual, they do provide ESD protection. As seen in Figure 2.1, the diodes allow a resistive path for charge to bleed-off of the gate oxide when the transistor is not in use. Without these diodes, once enough charge builds up, the gate oxide would break down, causing the transistor to fail prematurely. All of the sources of the transistors are tied together and tied to the substrate through the BITIE. Connecting the transistors in this manner ensures that all of the transistors are at the same electrical potential. Table 2.1 gives a list of all the transistors that can be found on the three chips.

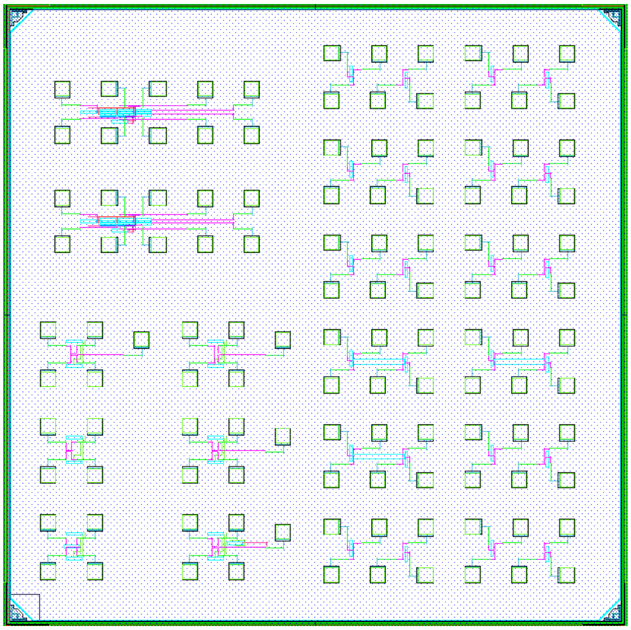


Figure 2.3: Top level view of the 2x2mm pad frame. All of the transistors measured are on the right side of the chip in between the 6x12 pad array.

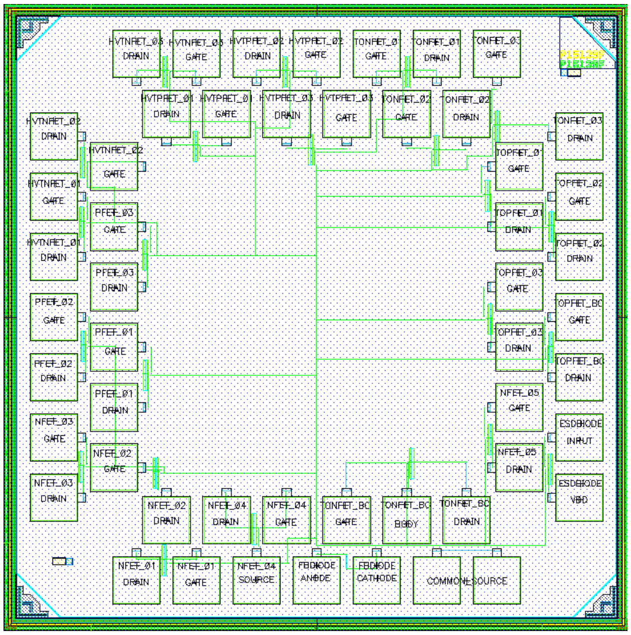


Figure 2.4: Top level view of the 1x1mm pad frame. All of the pads are located around the edge of the chip to make wire bonding easier.

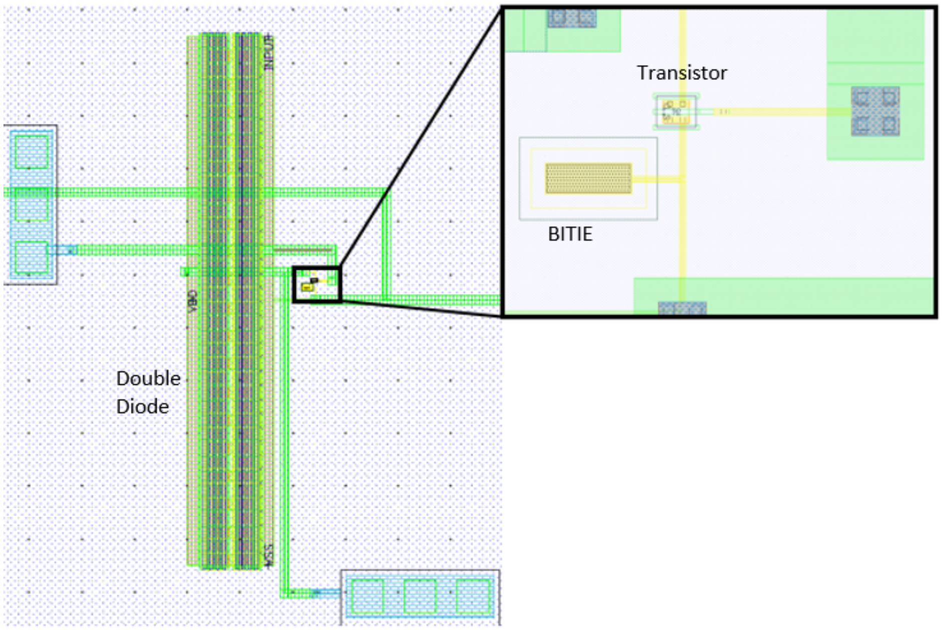


Figure 2.5: Closer look at a transistor on the 1x1mm chip.

2.3 Test Devices

Transistors fabricated in the SOI process are superior to those fabricated in the conventional bulk process. SOI transistors have up to 20 percent greater performance, 30 percent less power consumption, and a 15 percent reduction in size [3]. The SOI process is implemented by adding a silicon dioxide insulating layer in between the silicon substrate and the transistor wells as diagrammed in Figure 2.6 [4].

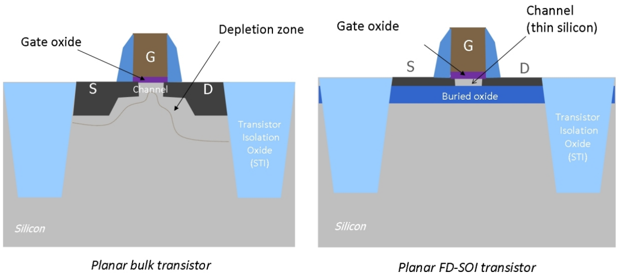


Figure 2.6: Cross section of a transistor fabricated using an SOI process vs a bulk process.

Table 2.1: Complete list of the transistors that were fabricated on the three chips used in this study

Device	w (nm)	L (nm)	T_{ox}[a] (nm)	Nom. Oper. Voltage (V)	Device	w (nm)	L (nm)	T_{ox} (nm)	Nom. Oper. Voltage (V)
RVT nFET	104	40	1.5	0.9	UVT pFET	2860	40	1.5	0.9
RVT nFET	208	40	1.5	0.9	SVT nFET	104	40	1.5	0.9
RVT nFET	520	40	1.5	0.9	SVT nFET	2860	40	1.5	0.9
RVT nFET	1040	40	1.5	0.9	SVT pFET	104	40	1.5	0.9
RVT nFET	2860	40	1.5	0.9	SVT pFET	2860	40	1.5	0.9
RVT pFET	104	40	1.5	0.9	AVT nFET	624	56	1.5	0.9
RVT pFET	208	40	1.5	0.9	AVT PFET	624	56	1.5	0.9
RVT pFET	520	40	1.5	0.9	TOnFET	312	100	2.5	1.5
RVT pFET	1040	40	1.5	0.9	TOnFET	624	100	2.5	1.5
RVT pFET	2860	40	1.5	0.9	TOnFET	1560	100	2.5	1.5
HVT nFET	104	40	1.5	0.9	TOpFET	312	100	2.5	1.5
HVT nFET	208	40	1.5	0.9	TOpFET	624	100	2.5	1.5
HVT nFET	520	40	1.5	0.9	TOpFET	1560	100	2.5	1.5
HVT nFET	2860	40	1.5	0.9	AVT nFET BC	624	56	1.5	0.9
HVT pFET	104	40	1.5	0.9	AVT pFET BC	624	56	1.5	0.9
HVT pFET	208	40	1.5	0.9	AVTL nFET BC	624	490	1.5	0.9
HVT pFET	520	40	1.5	0.9	AVTL pFET BC	624	490	1.5	0.9
HVT pFET	2860	40	1.5	0.9	AVTD nFET BC	624	100	1.5	0.9
UVT nFET	104	40	1.5	0.9	AVTD pFET BC	624	100	1.5	0.9
UVT nFET	2860	40	1.5	0.9	TOnFET BC	624	100	2.5	1.5
UVT pFET	104	40	1.5	0.9	TOpFET BC	624	100	2.5	1.5

[a] Approximated values. See design manual for actual values

From the capacitance charge equation,

$$Q = Cv \tag{2.1}$$

where Q is the charge of the capacitor in coulombs, C is the capacitance in farads, and v is the voltage in volts [5, 93]. As the capacitance increases, the amount of voltage needed to produce a given amount of charge will decrease. For a transistor the capacitive structure is the gate oxide. The gate oxide capacitance resembles a parallel plate capacitor and is calculated by,

$$C_{ox} = \frac{K_o \epsilon_o A_G}{t_{ox}} \tag{2.2}$$

where C_{ox} is the gate oxide capacitance, K_o is an oxide dielectric constant, ϵ_o is the permittivity of free space constant, A_G is the gate area, and t_{ox} is the gate oxide thickness[6, 49]. Shown in Table 2.1, for the same size HVT and RVT transistor t_{ox} and A_G are identical. The only term from Eq. 2.2 that affects the capacitance of the two devices is K_o. K_o is modified during the IC fabrication process through ion implantation [6, 120]. By implanting boron or phosphorus ions near the oxide-semiconductor boundary, K_o of the transistor is modified. For example, if boron ions are implanted, K_o will decrease causing C_{ox} to decrease, which will cause the voltage required to reach an adequate charge to turn on the transistor to increase. Implanting boron ions are how the threshold voltage is adjusted to make HVT devices.

Out of all of the devices listed in Table 2.1, the 104nm and 2860nm RVT MOSFETs were chosen as the devices on which to perform the reliability study. The RVT devices were chosen instead of the HVT devices because in low power electronics, transistors with lower threshold voltages are more commonly used in industry. In addition, the lower the threshold voltage for the device, the more susceptible the devices is to high voltage stress effects. The 104 nm device is the minimum width that can be fabricated using the 32SOI process and, at 27.5 times bigger, the 2860nm device is the largest size that is allowed. Choosing these two devices allows a best and worst case scenario to be developed for the RVT MOSFET.

Furthermore, during the cryogenic model development, having a model that covers the entire range of the process will be beneficial when making future circuit designs.

Chapter 3
Test Procedures

This chapter is an overview of the process used to test transistor reliability at RT and in LHe. The stress method used to determine the transistor reliability, testing equipment, and data processing will be discussed. A complete overview of testing equipment and procedure can be found in Appendix B.

3.1 Stress Introduction

HCE, also called hot carrier injection, is becoming an increasing issue with MOSFETs as the size of the transistors decrease. HCE occurs in MOSFETs when channel carriers gain abundant kinetic energy while traveling through the depletion region and collide with the gate oxide with sufficient force to cause the carrier to become embedded into the oxide [6, 136]. When high voltage biases are applied to the gate (V_{GS}) and drain (V_{DS}) of transistors, these hot carriers are introduced into the devices. Multiple parameters of the MOSFETs are effected by HCE, but arguably the most important parameter is threshold voltage (V_T). With the increase in number of hot carriers embedded into the oxide, V_T of the devices also increases. This causes a higher bias on V_{GS} of the device to achieve the desired drain current (I_D) defined by:

$$I_D = \frac{w\mu C_{ox}}{L}(V_{GS} - V_T - \frac{V_{DS}^2}{2})V_{DS} \tag{3.1}$$

in the linear regime, and by:

$$I_D = \frac{w\mu C_{ox}}{2L}(V_{GS} - V_T)^2 \tag{3.2}$$

in the saturation region, where w is the gate width, μ is the mobility, and L is the gate length [7]. From these equations it can be easily seen that, with all other parameters constant, as

V_T increases, I_D will decrease. Due to HCE continuing to accumulate over time, the life of the MOSFETs is impacted and becomes a critical factor in the reliability of the device. Holes have been proven to cause approximately 100 times more breakdown in the oxide than electrons when traveling with enough energy [8]. The gate current (IG) in pMOS devices is made up of strictly electrons. In nMOS devices IG is a combination of electrons and holes. Due to this fact, pMOS devices have been known tolerate about 1000 times more HCE stress than nMOS devices [8]. For this reason, nMOS devices will be the focus of the reliability so that the worst case scenario can be observed.

3.2 Testing Equipment

In order to simulate this stress, a test set-up has been designed that allows each transistor to be connected to a Keithley 4200 Semiconductor Characterization System. The entire set-up consists of 4 parts: the Keithley 4200, an Agilent B2201A Switch Mainframe with B2211A cards, a custom cryogenic probe, and a Cryofab dewar.

The Keithley 4200 is used for all of the measurements of the devices. The transistor is connected using 3 SMUs. The drain and source contacts both have force and sense connections with the gate contact only using the force connection SMU. 4 point measurements and stresses are performed on the transistor using KITE.

The Aglient B2201A is used to automate the test set-up, allowing multiple transistors to be tested autonomously. All of the inputs and outputs of the B2201A and B2211A cards are triax ports that provide sufficient shielding for the signals to and from the transistors.

The cryogenic probe is used to interface the individual transistors to the switch matrix. The probe is constructed of a hollow stainless steel pipe to allow signal lines to be fed through connecting the PCB board mounted on the bottom to the breakout box on the top. A 40 pin gold plated DIP is used to wire bond the transistor out to a manageable package. In order to eliminate ESD issues, a wedge bonder was used to make the bonds. A wedge wire bonder uses ultrasonic energy to attach gold wires to the exposed chip pads without having the electric arc used to create the ball on a ball bonder. A complete tutorial of how to use the wedge bonder to bond these chips can be found in Appendix C. After the chips are wire bonded to a DIP, they are placed in the DIP socket on the PCB board on the end of the probe. This probe allows testing to be performed at room temperature or cryogenic temperatures when submerged into a dewar.

The Cryofab dewar is a stainless steel vacuum sealed tank that can hold up to 100L of LHe allowing transistors to be measured in temperatures as low as 4.2K when the PCB on the end of the probe is submerged.

3.3 Measurements

A set of measurements is performed on each transistor to conduct the reliability study. The set of measurements includes 3 different measurements:

1. Standard
2. Stress
3. Post

3.3.1 Standard

Standard measurements are where the transistors characteristic curves, I_D-V_{DS} and I_D-V_{GS}, are established. During the standard measurement, one of the voltage biases V_{DS} or V_{GS} is swept from 0 to the nominal operating voltage over 51 data points, while the other bias is stepped from 0 to nominal operating voltage over 5 data points. For the 32nm SOI process used to fabricate these transistors, the nominal operating voltage specified in the design manual is 0.9 V. The standard measurements ensure device functionality and allows the standard parameters of the transistors to be determined before stressing.

3.3.2 Stress

During the stress measurements the reliability of the transistor is determined. Stress measurements induce HCE and by applying a high voltage bias to V_{DS} and V_{GS} of the transistors and periodically taking measurements to determine how much stress has occurred. After various test to determine what stress biases would be used, the voltage range of 1.6 V and 2.1 V was selected for V_{DS}. The lower bound, 1.6 V, was set because by decreasing the voltage anymore required too much time to achieve significant levels of HCE stress. Contrarily, the upper bound of 2.1 V was set because any higher bias caused too much stress over just 1 second of stress and kill the device. If V_{GS} is too large the gate oxide is will break down causing the device to fail before the HCE stress can affect the device. Because of this, the bias on V_{GS} is always set equal to half of V_{DS}. The periodic measurement performed during the stress is I_D-V_{GS} with V_{GS} sweeping from 0 to 0.8 V while V_{DS} is set equal to 50 mV. Setting V_{DS} equal to 50 mV ensures that the transistors will be in the linear regime allowing the extrapolation in the linear regime method to be used when extrapolating the device threshold voltage. By only sweeping to 0.8V instead of the nominal operating range, prevents the measurement from adding any additional stress to the devices.

3.3.3 Post

The post measurement is identical to the standard measurement, except the characteristic curves show the effects that the HCE have on the device.

Table 3.1 contains a list of the measurement parameters. After performing this set of measurements on the transistor, the data can then be imported into MATLAB to analyze the data.

Table 3.1: Measurement Performed on the 32nm RVT nFETs

Measurement	Type	V_{GS}	Steps	V_{DS}	Steps
Standard/Post	I_D-V_{GS}	0-0.9 V	51	0-0.9 V	5
	I_D-V_{DS}	0-0.9 V	5	0-0.9 V	51
Periodic Stress	I_D-V_{GS}	0-0.8 V	51	50 mV	0

3.4 Parameter Extraction

In order to determine how the HCE affects the transistor's parameters, the parameters must be extracted from measured data. V_T is primarily the most important parameter of the transistor in terms of characterization. There are over a dozen ways to extract V_T from MOSFETs. To determine V_T for all of the discussion to follow, the extrapolation in the linear regime method will be used [9]. This method requires that the maximum derivative be calculated from the I_D-V_{GS} curve of the transistor, which corresponds to the maximum transconductance given by:

$$g_m = \frac{dI_D}{dV_{GS}} \tag{3.3}$$

where g_m is the transconductance [10, 500]. When extracting the maximum derivative, a 3-point derivative is performed on the curve. Using

$$f'(x_{i-1}) = \frac{-y_{i+1} + 4y_i - 3y_{i-1}}{x_{i+1} - x_{i-1}} \tag{3.4}$$

a plot can be made of g_m-V_{GS} to determine at what voltage of V_{GS} the maximum gm occurs (Figure 3.1).

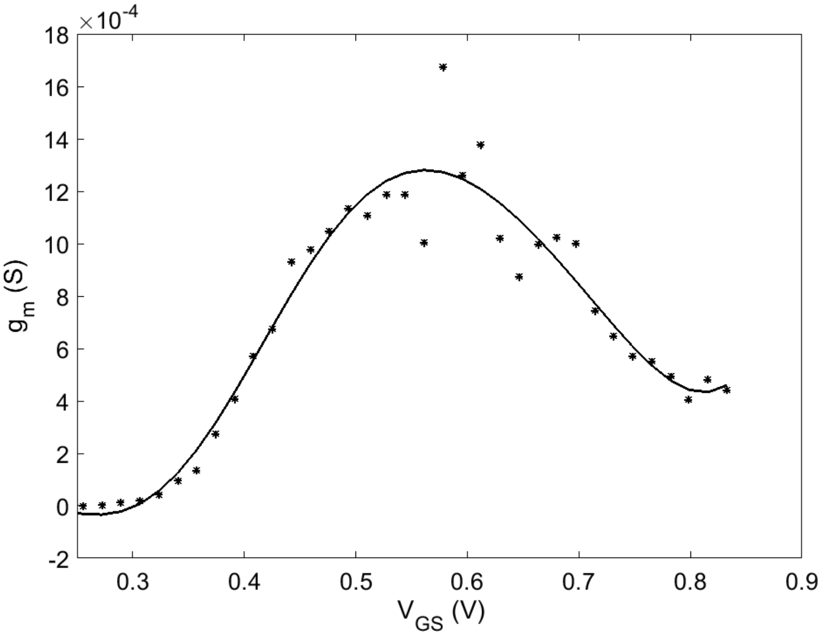


Figure 3.1: First derivative of the I_D-V_{GS} curve using the 3-point derivative and a fit line used to extract where max gm occurs.

After the voltage level of V_{GS} is determined, the slope at that point is extrapolated from the I_D-V_{GS} curve. Once this slope is found, a linear line is plotted on the I_D-V_{GS} curve that has the extrapolated slope and travels through the V_{GS} that was determined from the maximum g_m. Using this linear line, the voltage of V_{GS} can be found to achieve any I_D specified (Figure 3.2). The voltage level of V_{GS} at I_D equal to 0A is the V_T of the transistor.

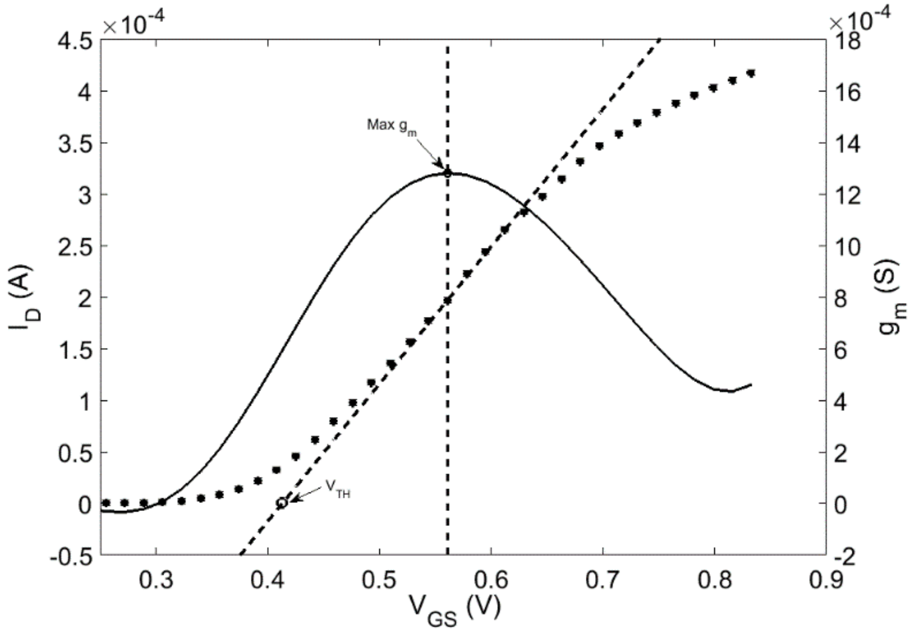


Figure 3.2: Extraction processes used to extrapolate the threshold voltage from an I_D-V_{GS} curve.

As the stress time increases, the number of carriers embedded in the transistors increases causing V_T to increase. Increasing V_T causes the I_D-V_{GS} curve to shift to the left (Figure 3.3). When extrapolating the new V_T after the shift, the slope of the linear line is held constant. Only the V_{GS} level at which the maximum g_m occurs is allowed to change. Keeping the slope constant assumes that the g_m is constant as the amount of HCE exposure in the transistor increases. This assumption is made because the mobility of the channel remains constant because the embedded electrons only affect the gate oxide. When plotted together on the same I_D-V_{GS} curve as a function of HCE stress, all of the linear lines are parallel to one another.

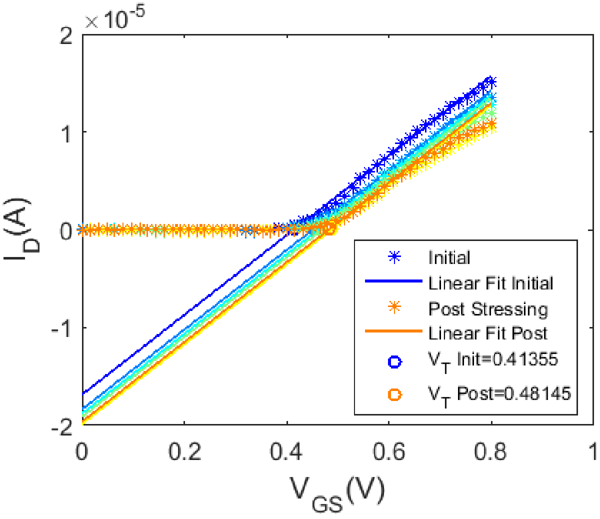


Figure 3.3: Threshold voltage shift during HCE stressing.

Taking multiple V_T measurements as the amount of HCE stress applied to the transistor is increased, plots of change in V_T versus time can be developed. This plot is the basis for the MTTF extractions. The MTTF of a transistor is the time that it takes for the transistor to fail when operating at a given voltage. By determining the time to failure at various high voltage biases when performing the stress measurements, the time to failure at any other voltage bias can be extracted. This allows for the nominal voltage reliability of the device to be estimated without having to perform exceptionally long duration stresses on the transistor. Refer to Appendix D for all of the MATLAB code used for the parameter and MTTF extractions.

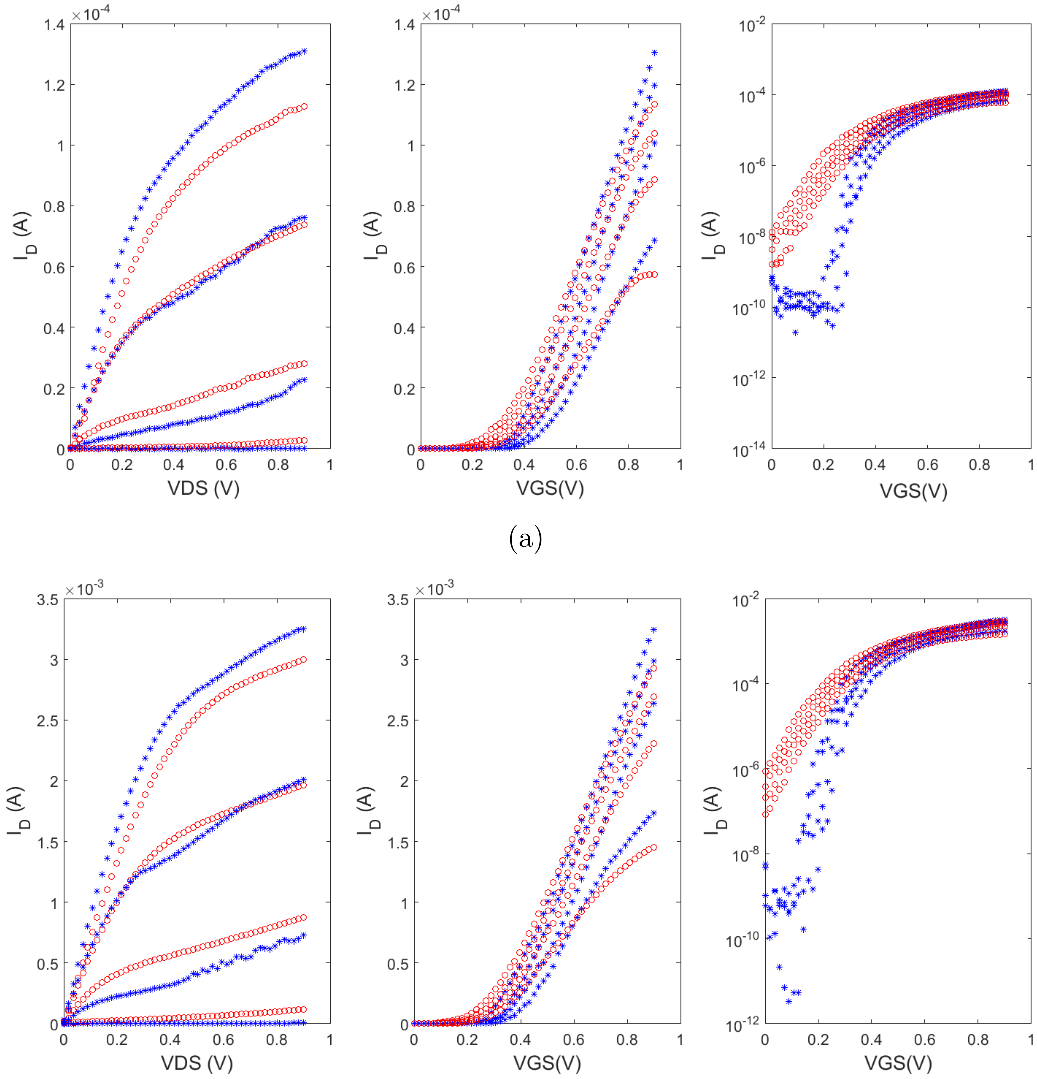
Chapter 4

Measurement Results

The following sections will be used to display the measurement results of the study. Differences in the standard measurements at RT and in LHe will be shown explaining the effect cryogenic temperatures have on transistor parameters. The MTTF will be determined for the 104nm and 2860nm RVT nFET and the results from the determination will be examined.

4.1 RT vs. LHe Standard

The standard measurements can be used to show many of the advantages of using MOSFETs at cryogenic temperatures. By observing standard measurement results in Figure 4.1 there is a visible difference in the RT and LHe results.



(a)

(b)

Figure 4.1: Comparison between RT and LHe standard measurements from (a) 104nm and (b) 2860nm single transistor.

The two linear scale plots show an increase in current for the high voltage bias on the LHe measurements. This is due to both a decrease in resistance and an increase in transconductance [7]. In metals, as the temperature decreases the thermal vibrations of the lattice atoms decrease allowing free electrons to flow more freely. Increasing the flow of electrons, increases the conductivity of the metal [11, 954]. On the I_D-V_{GS} RT linear curves the roll off at the end of the curves is due to the resistance of all the metal interconnects in the chip and on the probe. From Ohms Law, with a constant voltage applied as the resistance decreases the current, I_D in this case, must increase. This effect is displayed by the LHe I_D-V_{GS} curves continuing on a linear trend of increasing I_D.

The transconductance is directly related to the mobility of the transistor defined by,

$$g_m = \mu C_{ox}\frac{w}{L}(V_{GS} - V_T) \tag{4.1}$$

where, μ is the mobility[7]. Analogous to metals, the mobility of semiconductors increases as the temperature decreases due to a decrease in lattice vibrations (Figure 4.2). From Eq. 4.1 the increase in transconductance is directly related to the rate at which I_D increases. This is why the transconductance, when combined with the temperature dependent resistive effects cause the increase in I_D for the LHe standard measurements.

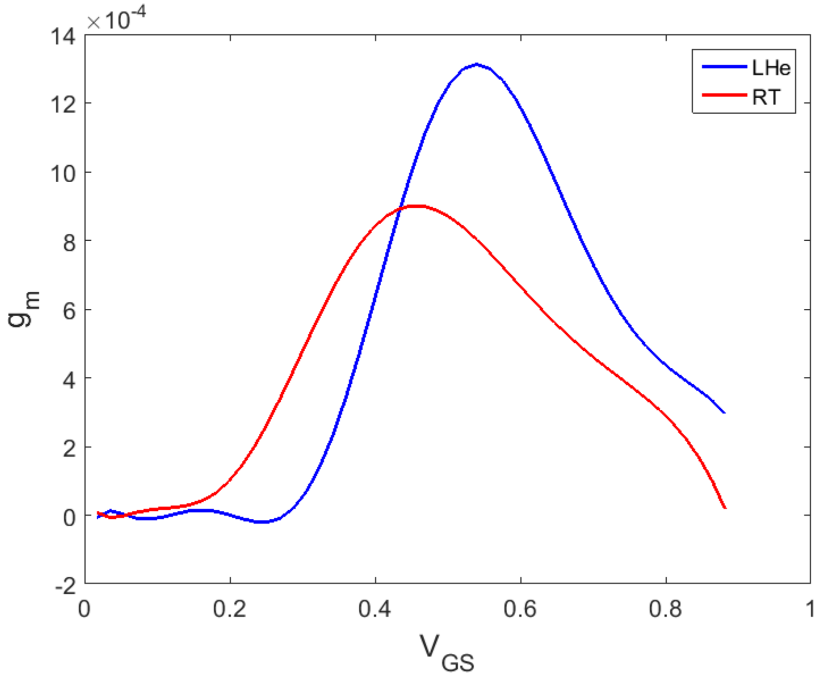


Figure 4.2: Transconductance difference between RT and LHe for a 2860nm transistor.

Similarly, the log scale plot exhibits an increase in the transistor performance for the LHe measurements in two key characteristics, subthreshold leakage current and subthreshold slope. The subthreshold leakage current is defined as the magnitude of I_D when the transistor is in the off state, V_{GS}<<V_T [12]. From looking at the RT and LHe log plots of I_D-V_{GS}, the magnitude of I_D while V_{GS}<<V_T is multiple orders of magnitude lower for the LHe measurements. The rate at which the transistor transitions from the off state to the on state or vise-versa is called the subthreshold slope [13]. The subthreshold slope is inversely related to temperature for transistors. As the temperature decreases, the subthreshold slope becomes steeper. Comparing the slope of the RT to the LHe measurements, the LHe has a subthreshold slope that is over 50% steeper than the device measure at RT (Figure 4.3). Increasing the subthreshold slope allows the transistor to be turned on and off at a faster rate, which increases the speed at which the transistor can operate. The subthreshold leakage and subthreshold slope advantages at cryogenic temperatures are both related to the decrease in lattice vibration causing increased isolation and increased mobility.

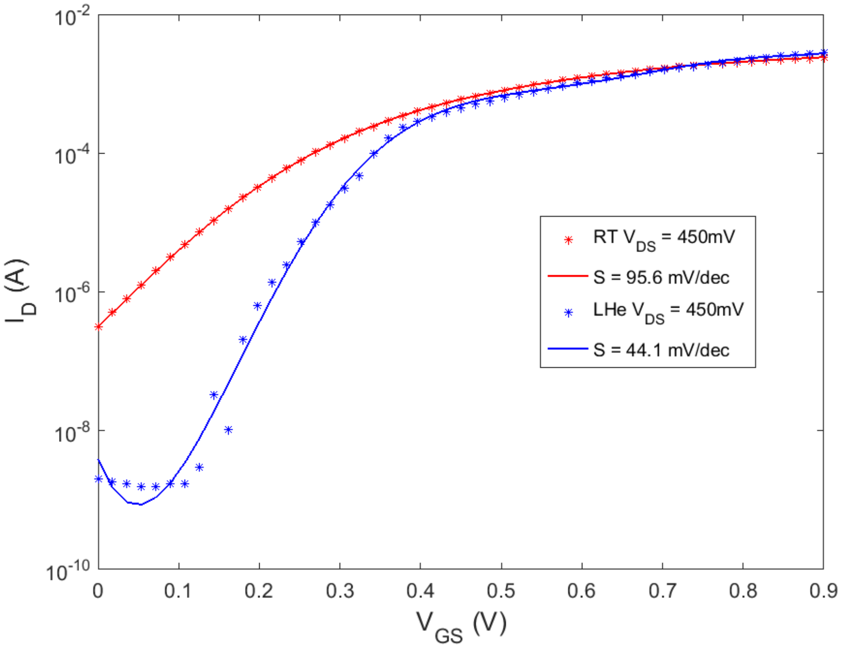


Figure 4.3: Subthreshold slope extraction from a 2860nm transistor.

One disadvantage related to the characteristics of the transistors at cryogenic temperatures is the increase in V_T. From the linear I_D-V_{GS} plot it can be observed on average that the RT measurements turn on around 0.3V for the RT and 0.4V for the LHe. This effect on the V_T is due to the Fermi potential and thermal voltage levels of the semiconductor. This relationship can be defined by:

$$V_T = \phi_{MS} + 2\phi_F - \frac{Q_{ox}}{C_{ox}} - \frac{Q_{depl}}{C_{ox}} \tag{4.2}$$

where, Q_{ox} is the charge density in the gate oxide, Q_{depl} is the depletion charge controlled by the gate, and ϕ_{MS} is the metal semiconductor work function difference defined by:

$$\phi_{MS} = -\frac{E_g}{2} - \phi_F \tag{4.3}$$

where E_g is the silicon band gap and ϕ_F is the Fermi potential defined by,

$$\phi_F = \frac{kT}{q} ln \frac{N_a}{n_i} \tag{4.4}$$

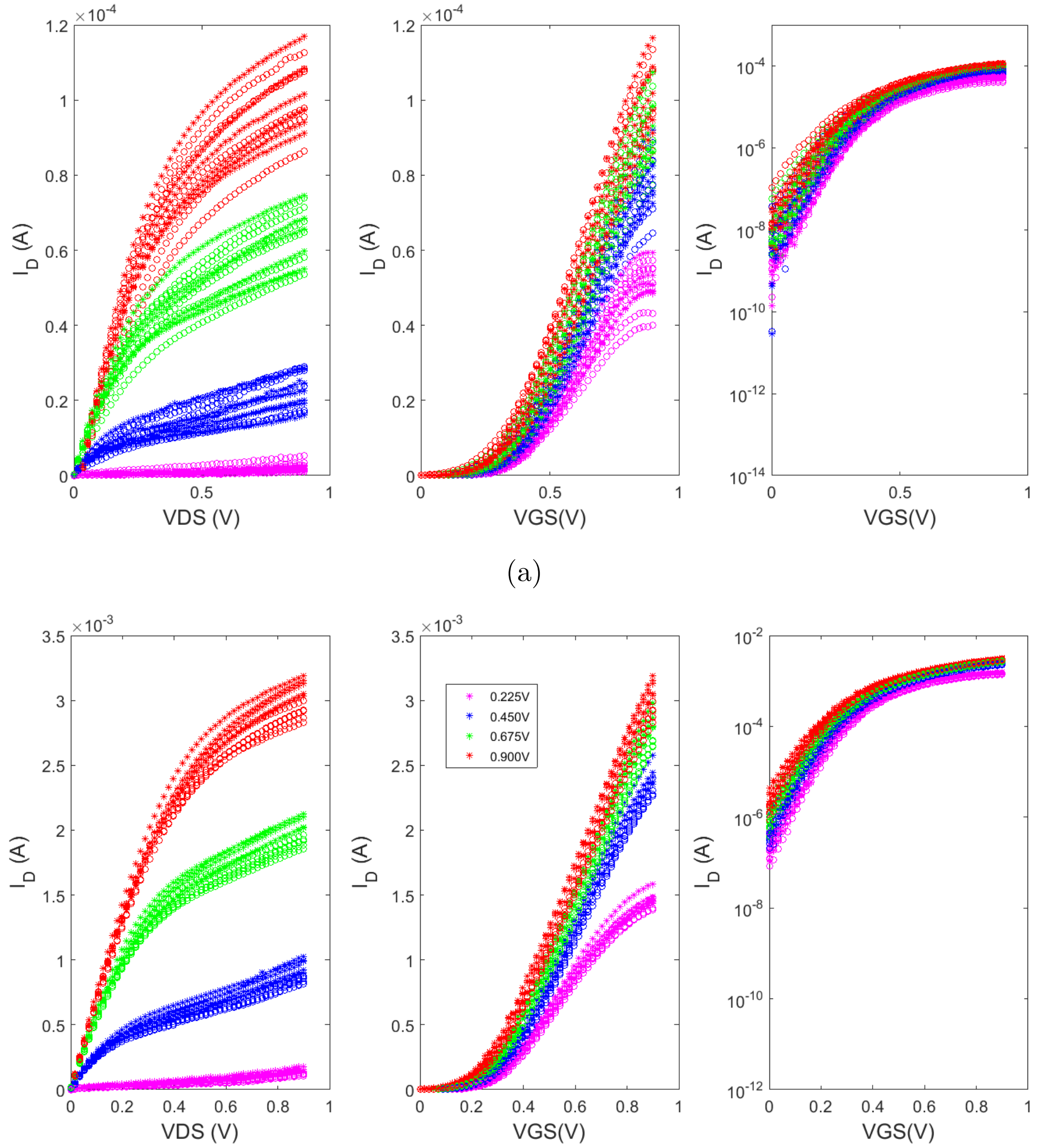
with k being Boltzmanns constant, T is the temperature in Kelvin, q is electron charge, and N_a and n_i are the substrate and intrinsic carrier concentrations, respectively. Assuming all of the charge and concentration quantities are held constant with respect to temperature, Eq. 4.2 can be condensed to

$$\frac{dV_T}{dT} = \frac{d\phi_F}{dT} \tag{4.5}$$

From Eq. 4.4 it is shown that increasing the temperature increases the Fermi potential. Increasing the Fermi potential decreases the amount of external potential that is needed to get an electron in the conduction band in the channel. For MOSFETs, this external potential is V_T[6, 31].

4.2 Reliability Results

The transistor characteristics associated cryogenic temperatures show advantageous results, but because these devices will be used in costly, complex systems the reliability at cryogenic temperatures is currently a large area of interest. Using the stress measurements detailed in Chapter 3, the MTTF for the devices can be determined. Figure 4.4 is a compilation of all of the RT standard measurements used in the reliability study.

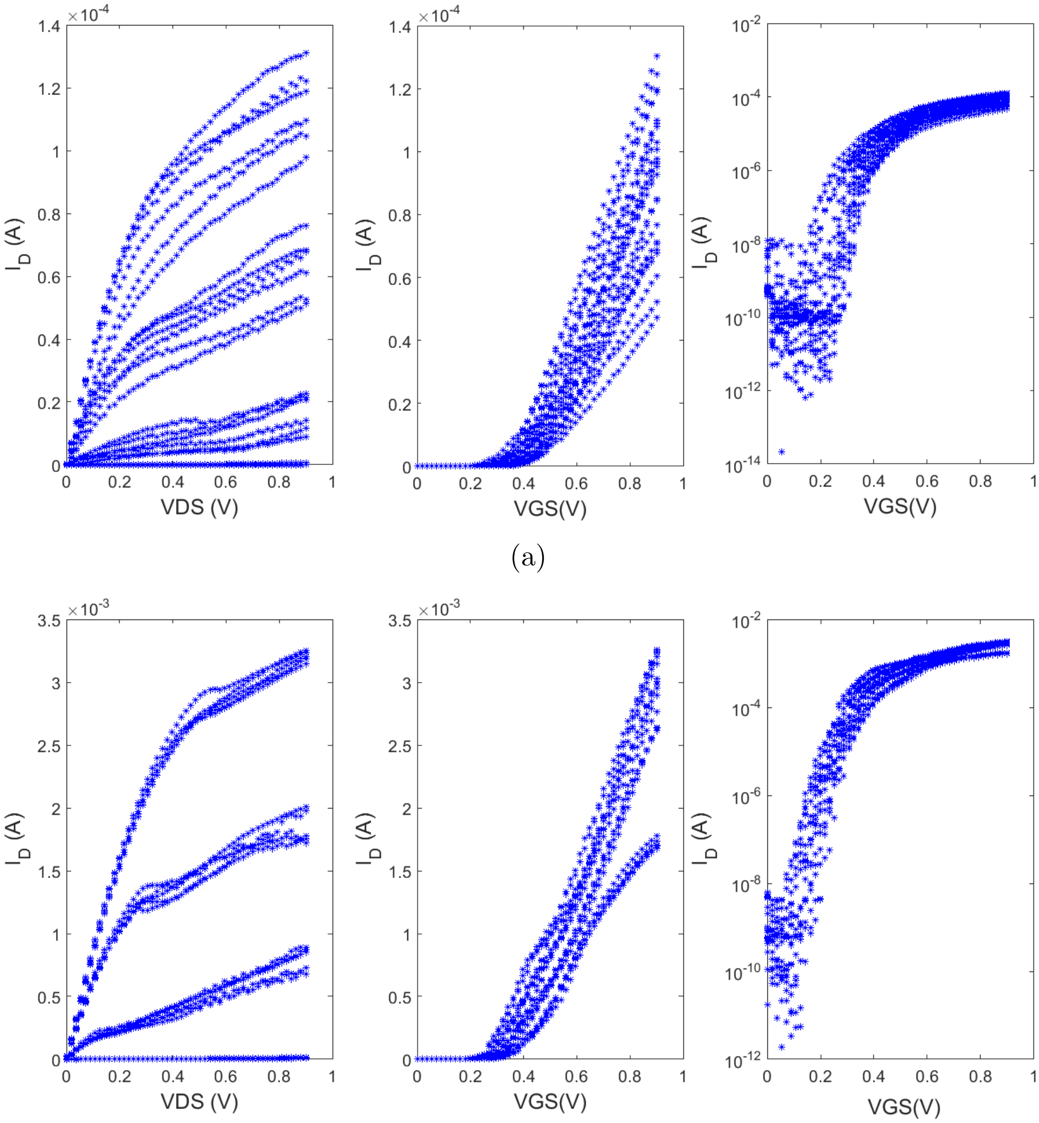


(a)

(b)

Figure 4.4: Complete set of the RT standard measurements used to complete the reliability study from (a) 104nm and (b) 2860nm RVT nFETs.

Curves depicted as circle data points were the devices used to complete the LHe stresses. Comparing all of the devices at RT before they are stressed ensures that initially the devices were similar to allow a logical comparison to be made from the stress results. Figure 4.5 is a compilation of all the LHe standard measurements that were used in the LHe stresses.

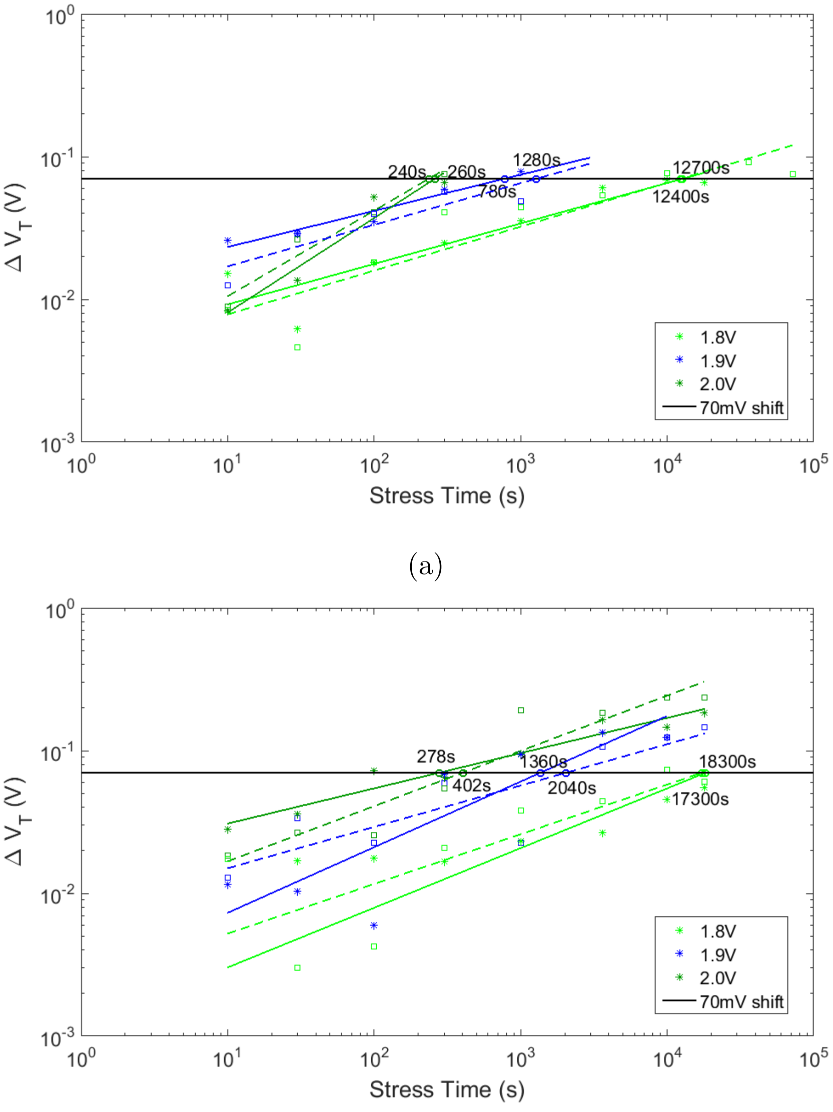


(a)

(b)

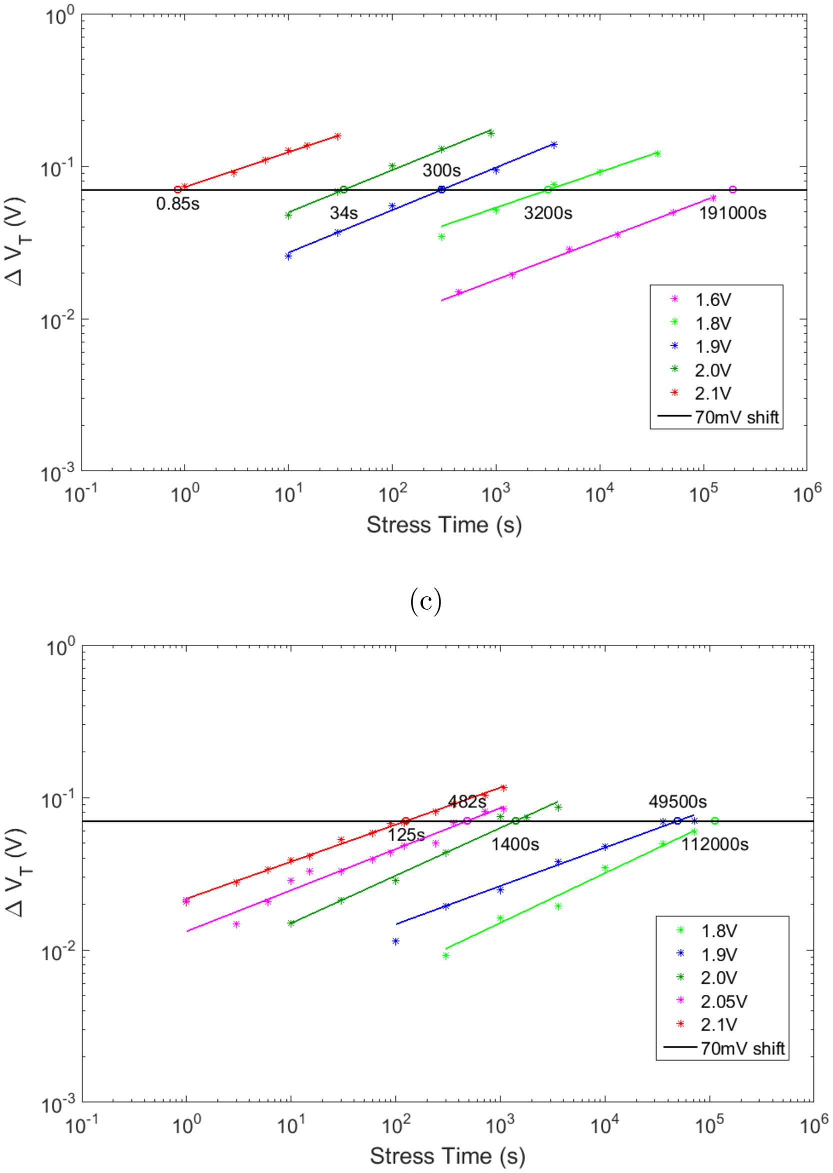
Figure 4.5: Complete set of LHe standard measurements used to perform the LHe temperature stresses from (a) 104nm and (b) 2860nm RVT nFETs.

Due to the variation seen in the 104nm devices, the stress magnitudes were limited to 1.8 V, 1.9 V, and 2.0 V with a pair devices stressed at each level. Averaging the failure times at each level before the MTTF is calculated increases the statistical significance of the results. The threshold voltage shifts from all of the stresses can be found in Figure 4.6. The square data points in the 104nm plots correspond to the dashed fit lines. Taking the stress times for each device to reach a 70mV shift the MTTF for the devices can be plotted (Figure 4.7).



(a)

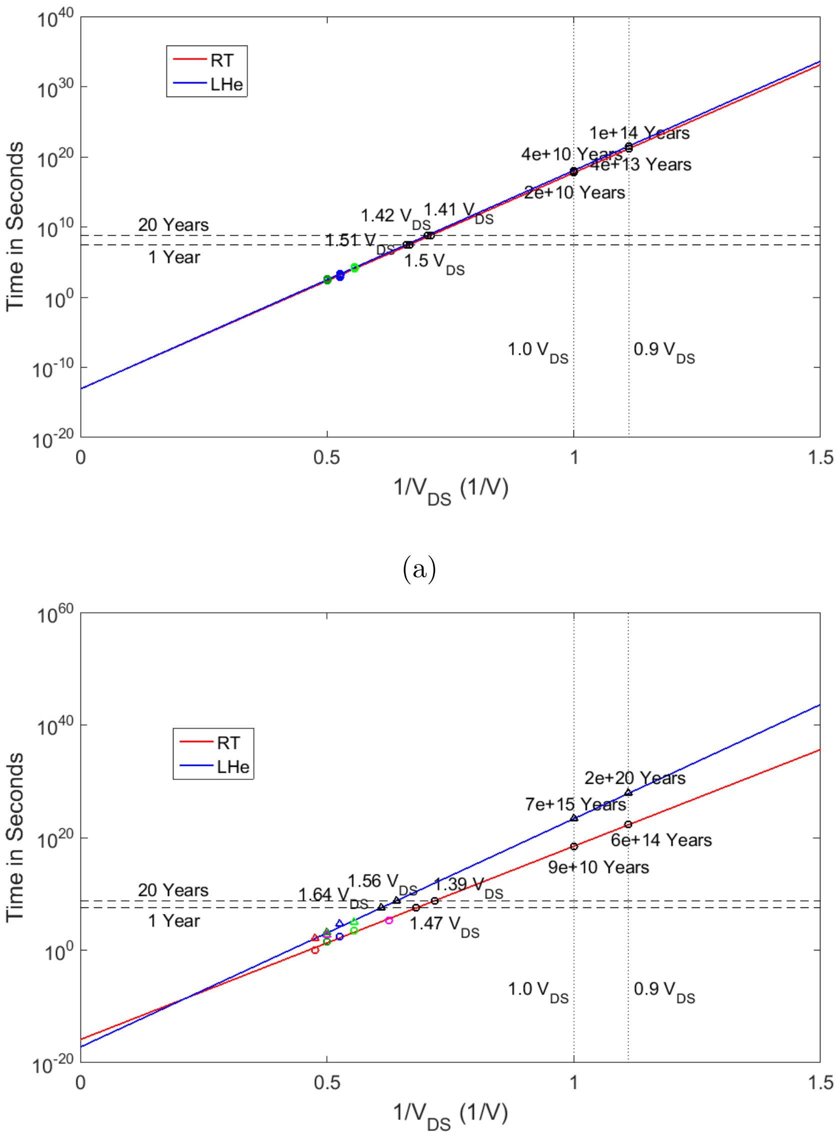
(b)



(c)

(d)

Figure 4.6: Threshold voltage shift from the 104nm devices at (a) RT and (b) in LHe and from the 2860nm devices at (c) RT and (d) in LHe.



(a)

(b)

Figure 4.7: MTTF extraction plots for the (a) 104nm and (b) 2860 nm devices.

By looking at the two MTTF extractions in Figure 4.7, the LHe extraction curve on the top indicates that the reliability of the devices measured in LHe is greater. For the 104nm results in Figure 4.7a the LHe only show a slight advantage over the RT. However, the 2860nm results in Figure 4.7b shows a 170mV increase in the voltage limit for failure in 1 and 20 years for the LHe devices. This data shows that devices operating in LHe can withstand about a 10% increase in drain voltage. Being that it is clearly seen in Figure 4.7

that the LHe devices show superior reliability, the LHe measures of the 104nm and 2860nm were compared to determine how the width of the transistor affects the reliability (Figure 4.8). The results from Figure 4.8 reveal that the 2860nm devices have a higher reliability. This size-dependence increase in reliability is due to the time-dependence HCE. If the same amount of electrons become embedded in the 104nm and 2860nm over a given amount of time, the effects seen by that amount of electrons on the 104nm devices will be greater.

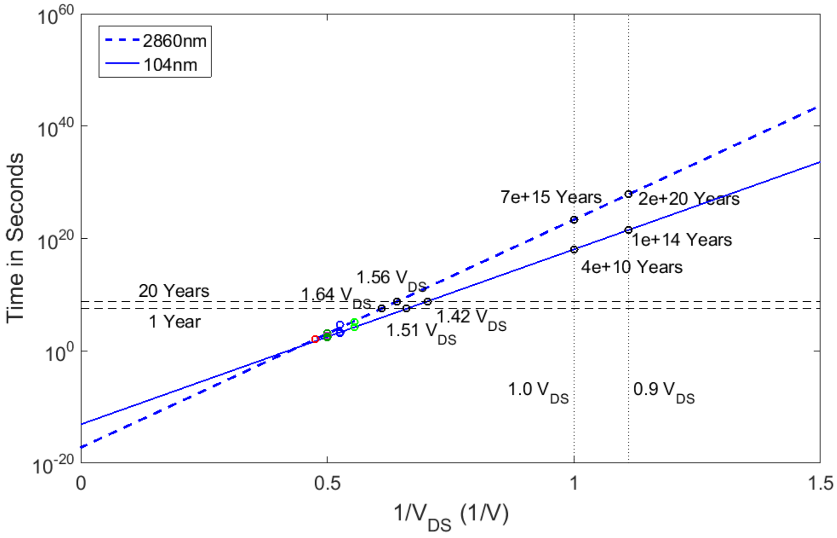
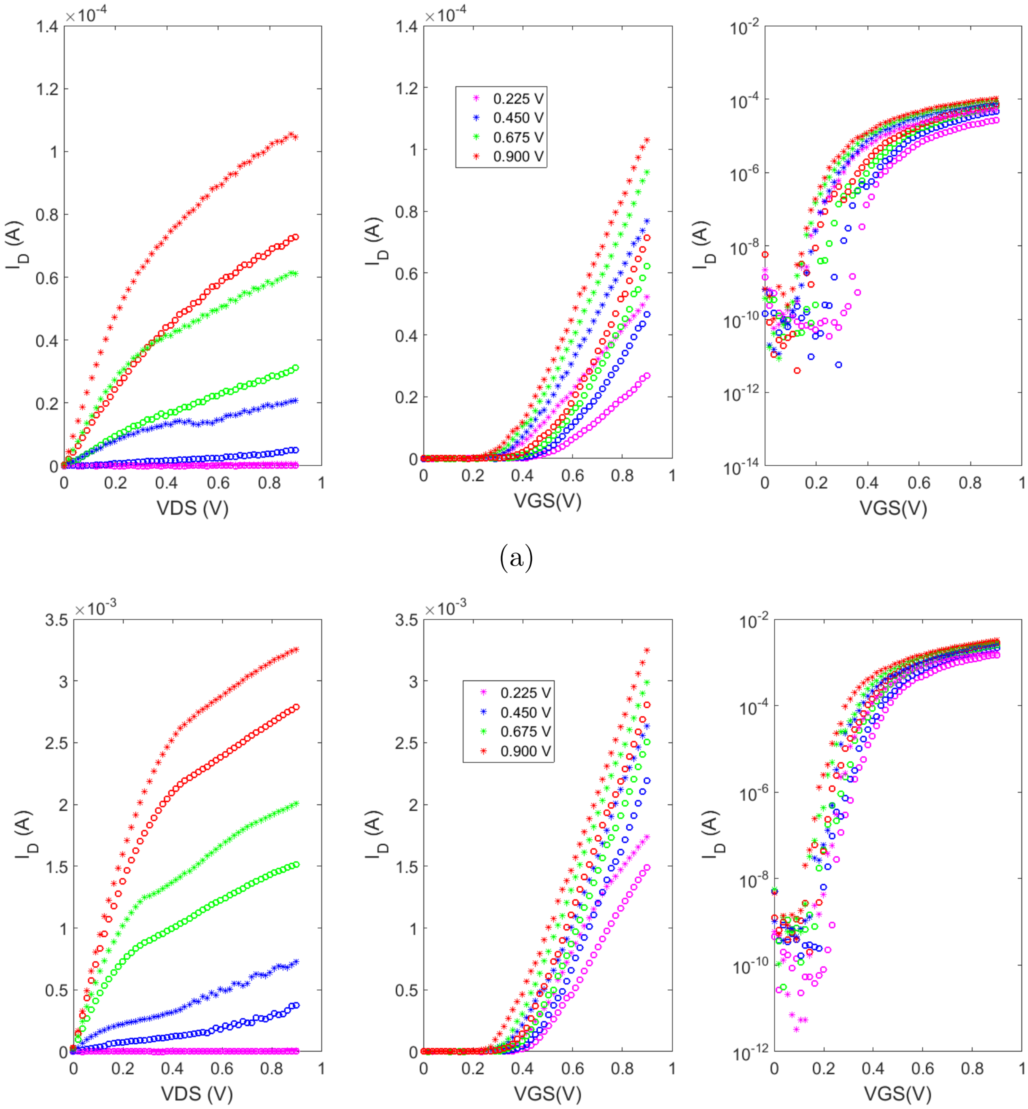


Figure 4.8: MTTF comparison between the 104nm and 2860nm devices in LHe.

Looking at an example of standard vs. post measurement shown in Figure 4.9. The HCE on the characteristic curves of the transistor show sizable degradation. Similarly to the results in Figure 4.8, the 104nm post measurement show greater degradation than the 2860nm devices. The maximum current from the I_D-V_{DS} decreased by 30% and 14% for the 104nm and 2860nm device, respectively.



(a)

(b)

Figure 4.9: Comparison between standard and post (circle) measurements for (a) 104nm and (b) 2860nm stressedin LHe at 2V.

Chapter 5

Modeling and Simulations

With an increasing complexity of electric circuits, the ability to be able to simulate the circuits during the design is becoming a necessity. The same goes for cryoelectronics. This section will cover the process of developing a mathematical model, such as a SPICE model, for a 32nm RVT nFET. All of the results will be shown in comparison with devices measured in LHe.

5.1 Modeling

Keysight's IC-CAP is used to produce the SPICE models of these devices. IC-CAP's device modeling process can be broken down into 3 steps:

1. Model set-up

2. Parameter Optimization

3. Model Extraction

A complete step by step tutorial of transistor modeling using IC-CAP can be found in Appendix E.

5.1.1 Initial Setup

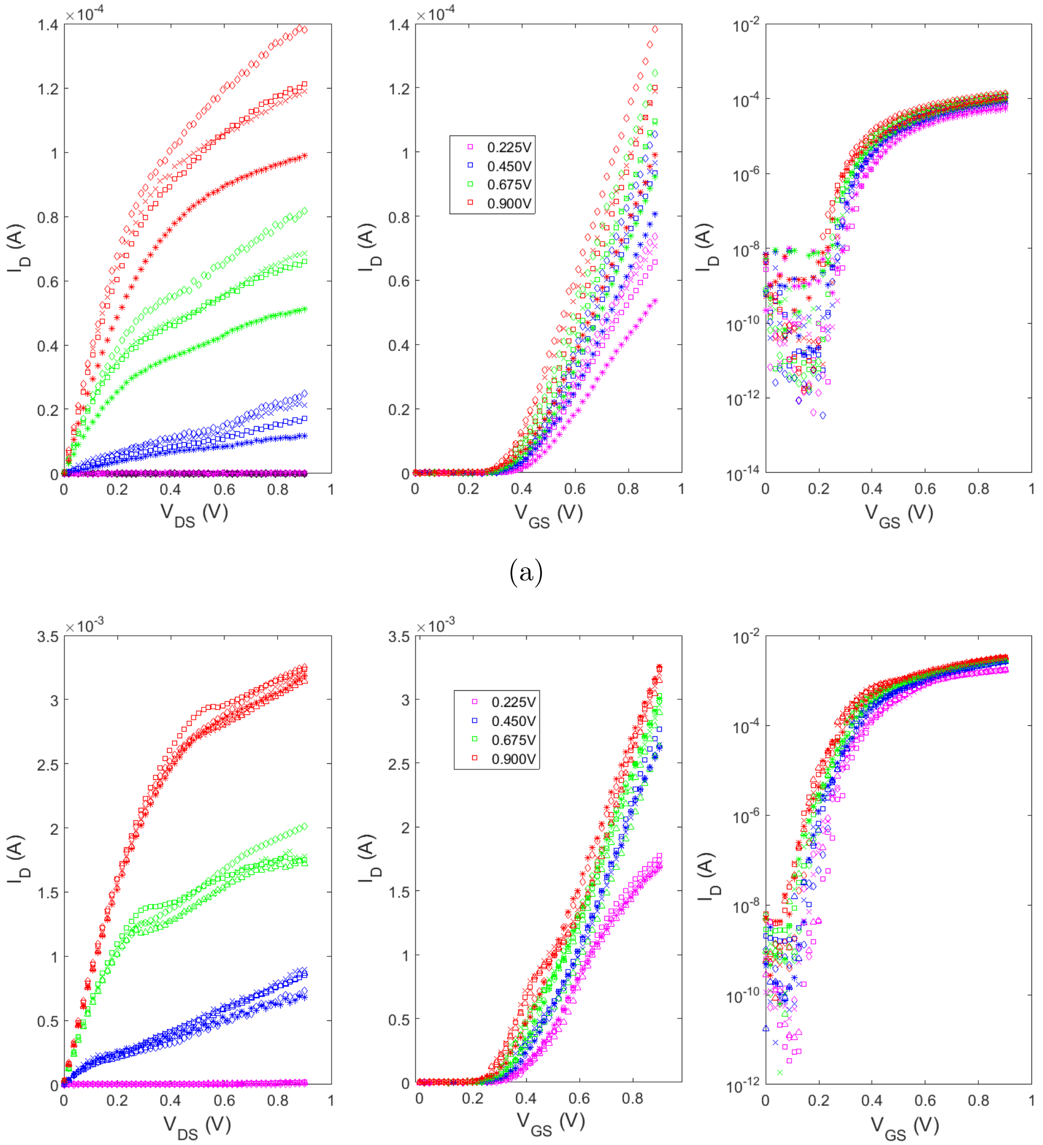
The first step when setting up the model is selecting which model will be used. There are various CMOS transistor models, such as BSIMv3.3, PSP, HISIMv2, and BSIMSOIv4.4. For modeling these transistors, designed with a 32nm SOI process, the BSIMSOIv4.4 model was selected. The BSIMSOIv4.4 is a MOSFET model that can be used to model SOI devices of both n-type and p-type. This model was developed by UC Berkeley in 2010 and is an industry standard for SOI MOSFET modeling [14]. Once the model is selected, the circuit must be added for the model. The circuit contains all of the model parameters of the BSIMSOI model in a format that is compatible with the simulator being used. Because Cadence Virtuoso will be used for the simulations when designing IC chips, Spectre was chosen as the simulator so that the cryogenic SPICE model will match the other RT models already in the PDK.

5.1.2 Optimization

Once, the model has been set-up, the model parameters must be adjusted until the simulated data is fit to the measured data. This is done by using IC-CAPs optimization function. The LM algorithm was chosen to optimize the parameters of the BSIMSOI MOSFET model. The LM algorithm uses a combination of gradient and Taylor series methods to fit a set of nonlinear data to another set of predetermined nonlinear data [15]. LM is an industry standard for nonlinear data fitting optimization and an essential tool that is built into IC-CAP. The parameters that were optimized using LM can be found in Table 5.1. By optimizing both the 104nm and 2860nm transistors at the same time, a single, scalable model can be created that only requires the width parameter to be adjusted when performing simulations. Due to all of the variation seen during the measurements, the measured data that was used to fit the model was an average of multiple devices that are shown in Figure 5.1. Figure 5.2 shows the final result after all of the parameters in Table 5.1 were optimized to the cryogenic data. A 10% margin was used when fitting the data due to the variations shown in Figure 5.1.

Table 5.1: List of Parameters used in Device Optimization

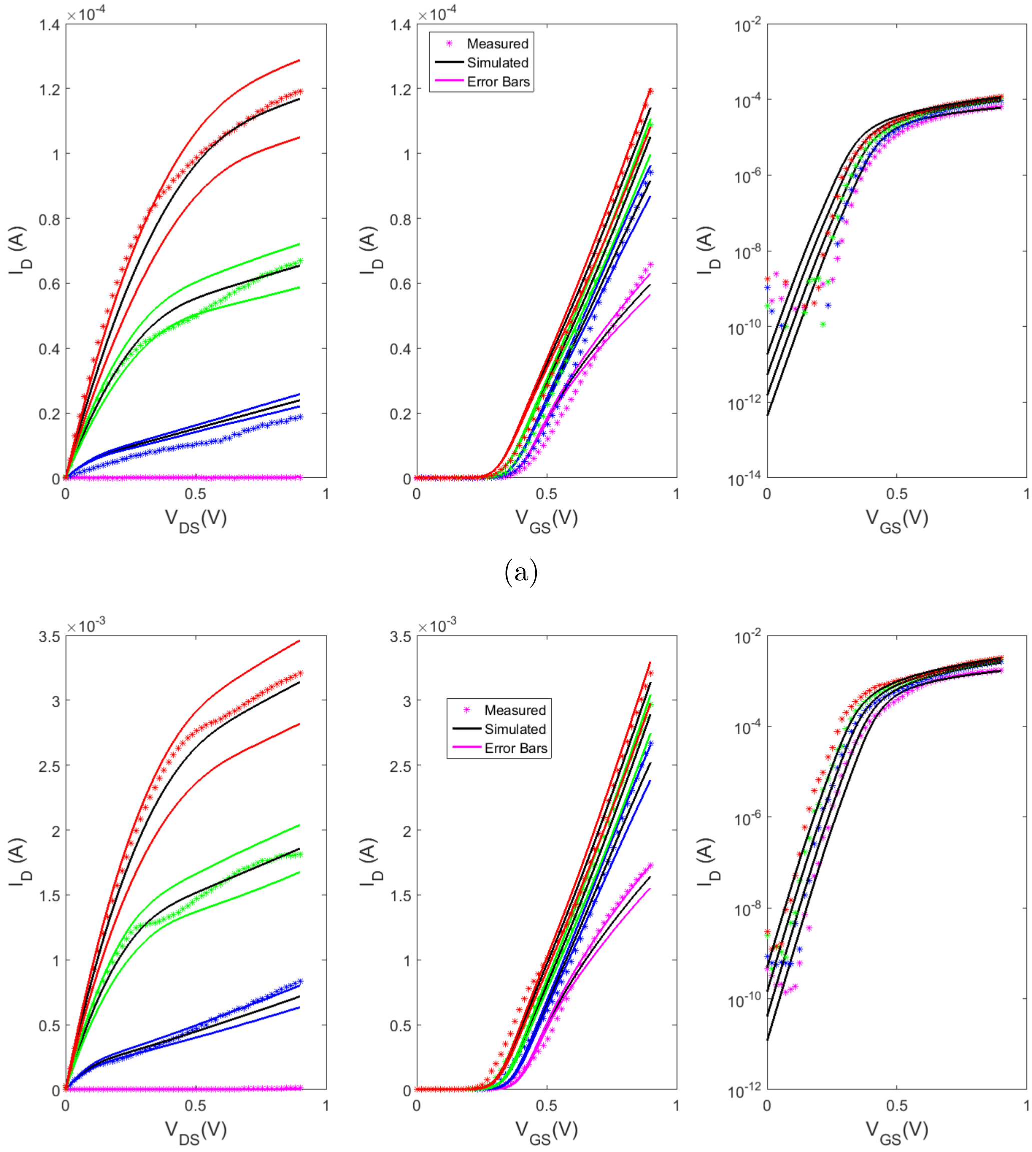
Parameter	Description
Vth0	Threshold Voltage
Dvt0	First coefficient of short-channel effect on Vth
Dvt1	Second coefficient of short-channel effect on Vth
Vsat	Saturation Velocity
Voff	Offset voltage in the subthreshold region
U0	Mobility
Nfactor	Subthreshold swing factor
Kt1	Temperature coefficient for threshold voltage
Eta0	Drain-induced barrier lowering coefficient in subthreshold region
Ute	Mobility temperature exponent
At	Temperature coefficient for saturation velocity
A2	Second non-saturation effect parameter



(a)

(b)

Figure 5.1: Cryogenic measurements from (a) 104nm RVT nFET and (b) 2860nm RVT nFET that were averaged together and used to fit the BSIMSOI model.



(a)

(b)

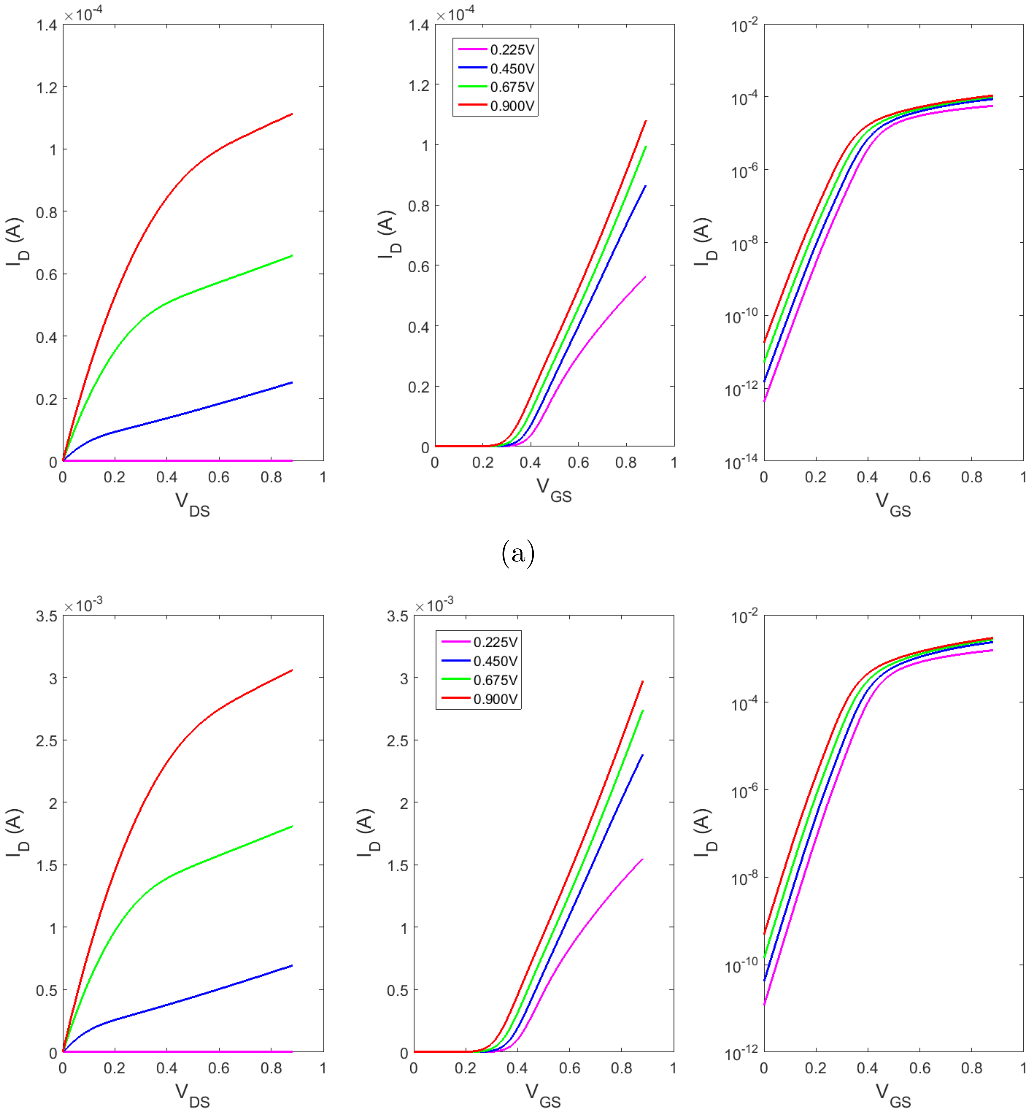
Figure 5.2: BSIMSOI model vs. the averaged measured data from Figure 5.1 (a) 104nm RVT nFET and (b) 2860nm RVT with 10 percent error bars plotted along the simulated data.

5.1.3 Extraction

Finally, after the model is optimized, the SPICE model must be extracted so it can be used in Cadence to simulate circuit diagrams. The model data is exported from IC-CAP using the Extracted Deck function. This function extracts all of the SPICE parameters from the model into a text file. In order to use these parameters with Cadence, they must be converted to a Model Card. The Model Card contains all of the SPICE parameters of the model. Then, after changing the reference location of the transistor symbol to the new cryogenic Model Card, the simulation can be performed using Cadence as detailed in Appendix A.

5.2 Simulations

Adding in the ESD diodes as shown in Figure 2.1, the complete circuit that was used on the test chips can be simulated (Figure 5.3). Furthermore, due to the fact that the model is scalable, the width can be swept over a range of 104nm to 2860nm to generate simulated results for any RVT nFET transistor within the PDK width range (Figure 5.4).



(a)

(b)

Figure 5.3: Cadence simulation the cryogenic model at (a) 104nm and (b) 2860nm with the ESD diodes connected as shown in Figure 2.1.

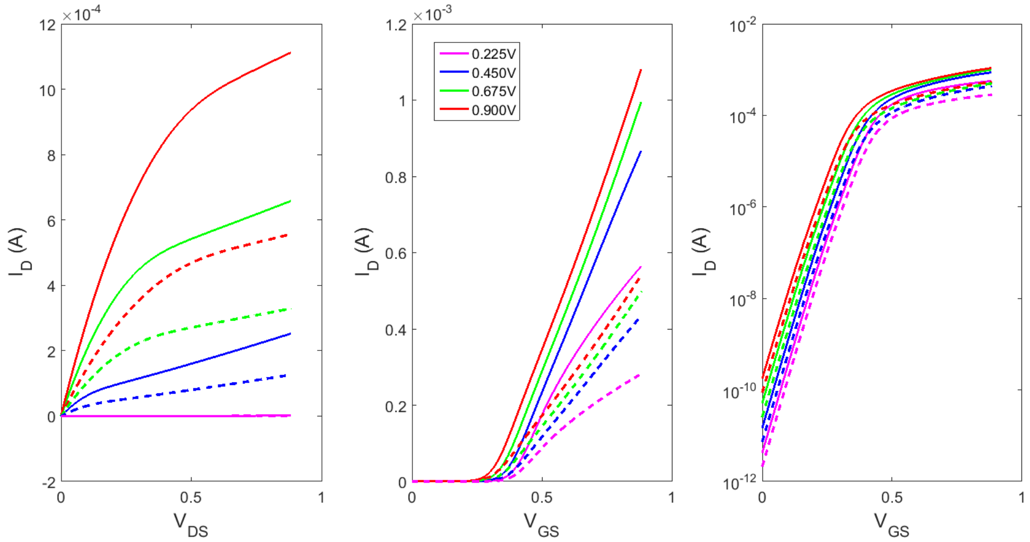


Figure 5.4: Cadence simulation results from changing the width to 520nm (dashed) and 1040nm (solid) in the cryogenic model.

By sweeping the step voltage, the functionality of the model can be tested over the entirety of the transistors rated voltage range (Figure 5.5). Taking the lowest voltage step, 50mV, on the I_D-V_{GS} curve, and comparing it to the initial measurement performed on the stress measurements shows that the model is valid for all voltage ranges when tested against measured data (Figure 5.6).

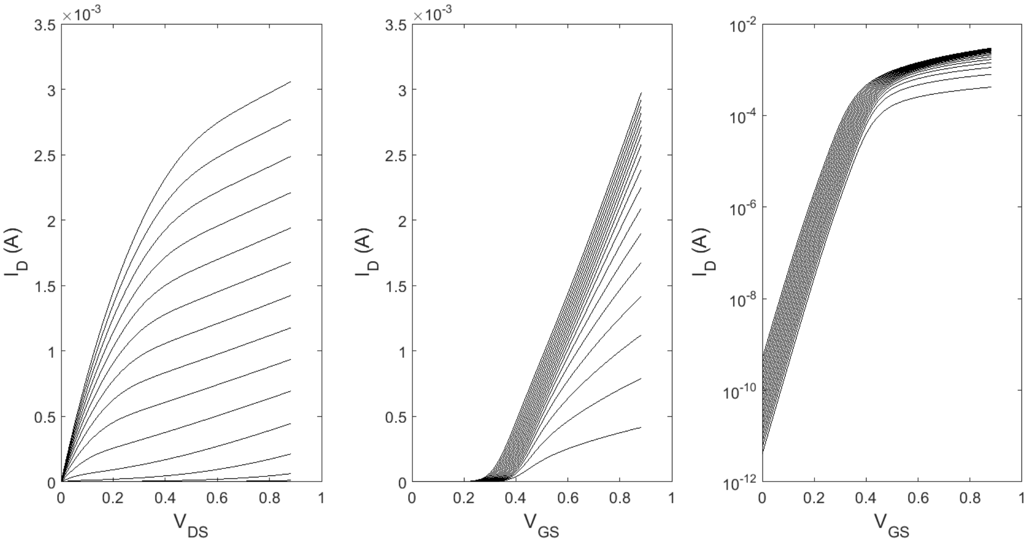


Figure 5.5: Cryogenic model with voltage step swept from 50mV to 900mV in 50mV increments.

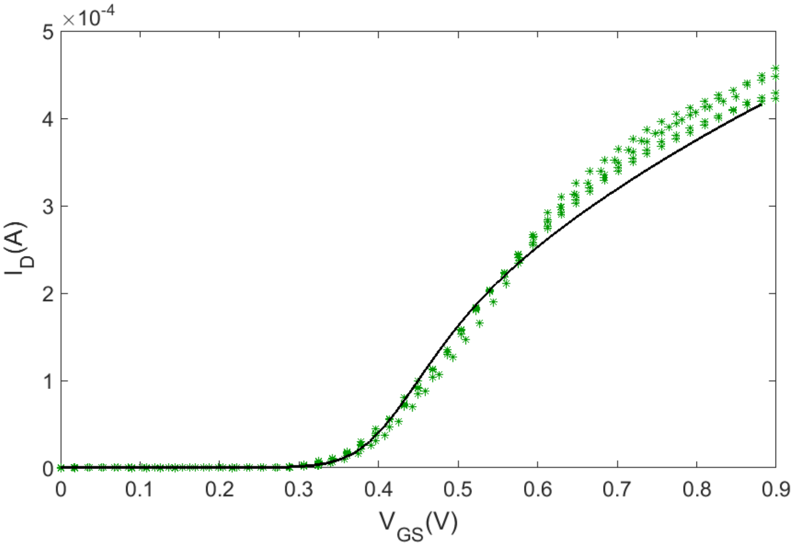


Figure 5.6: Comparison between the cryogenic model and the initial measurements taken during the stress measurements with V_{DS} equal to 50mV.

Chapter 6

Conclusion and Future Work

This chapter will cover the conclusions of the study along with succeeding reliability studies objectives to enhance the understanding of advanced node cryogenic transistors.

6.1 Conclusion

The use of cryogenic electronics open new doors for future electronic systems. Using cryogenic transistors in conjunction with superconducting material and q-bits, the supercomputers that can be developed will bring about tremendous performance improvements. However, with the continual push to decrease the fabrication node and decrease the operating temperature, reliability and modeling is becoming a greater concern. The study that was presented can be used as a step-by-step guide for testing and modeling advanced node RT and cryogenic transistors.

Inducing HCE by applying high voltage biases on the transistor accelerated the degradation from electrons being embedded into the gate oxide. By extrapolating the MTTF curves, the nominal failure times were calculated for transistors at RT and in LHe. The results presented in this study display an increase in reliability for the same size transistors stressed at cryogenic temperatures. Combining the increased performance with higher reliability shows promising results for the future of cryoelectronics. The development of cryogenic models enables future, more complex, circuits to be designed. The results illustrate the ability to produce circuit simulations that better emulate the circuit operation at cryogenic temperatures.

6.2 Future Work

All of the results, and models presented in this study are the forefront of a detailed cryogenic transistor database. Continuation of the 32nm RVT nFET testing will allow probability density functions to be developed, increasing the statistical significance of the results presented. Various stresses including: thermal cycling, bias temperature instability, and AC pulse stressing can be performed allowing the degradation effects to be compared to HCE degradation. Ion implantation effects can be observed by testing other V_T levels, such as HVT and SVT. Gate length and t_{ox} affects on gate leakage can be determined by testing thick-oxide transistors that have lengths of 100nm and t_{ox} of 2.5nm. Completing all of these test for the 32nm SOI nFETs will allow greater understanding of the devices while developing cryogenic models for each device. By testing pFETs fabricated from 32nm SOI, all of the transistors for the process can be evaluated based on reliability. Using the pFET data, cryogenic pFET models can be generated to allow complete cryogenic digital circuits to be simulated. Using this same measurement and modeling process, various fabrication processes such as: 12SO, 10RF, and 14LPP can be evaluated. This will allow for reliability comparisons to be made across a wide range of advanced node technologies. Determining the MTTF and developing models across all of the technologies and transistor types enables a complete cryogenic transistor database to be developed that can be referenced for reliability considerations and modeling of future cryogenic circuits.

Bibliography

[1] S. R. Das. *The Chip that Changed the World.* New York, NY: The New York Times, 2008.

[2] J. Eisenstein. "Superconducting Elements," *Reviews of Modern Physics.*, vol. 26, pp. 277-299, Jul. 1954

[3] A. M. Kodeti. (June 2009). *White Paper on Silicon on Insulator (SOI) Implementation* [Online]. Avaliable: www.infotech-enterprises.com [Jan. 13, 2016].

[4] X. Caunchy. (Feb. 20,2013). *Fully Depleted Silicon Technology to Underlie Energy-Efficient Designs at 28nm and Beyond* [Online]. Available: http://www.edn.com/design/consumer [Jan. 20, 2016].

[5] R. C. Jaeger, "Solid-State Diodes and Diode Circuits," in *Microelectronic Circuit Designs*, 4th ed, New York: Mcgraw-Hill, 2011, ch. 3, sec. 7, pp. 93.

[6] R. F. Pierret, *Field Effect Devices*, 2nd ed. Reading: Addison-Wesley, 1990

[7] J. Hu, "Low Temperature Effects on CMOS Circuits," Dept. Elec. Eng., Northwestern Univ, Evanston, IL

[8] E. Rosenbaum *et al.*, "Effects of Hot-Carrier Injection on n- and pMOSFET Gate Oxide Integrity," *IEEE Electron Device Lett.*, vol. 12, no. 11 pp. 599-601, Nov. 1991.

[9] A. Ortiz-Conde *et al.*, "A Review of Recent MOSFET Threshold Voltage Extraction Methods," *Microelectronics Reliability*, vol. 42, pp. 583-596, Apr. 2002.

[10] D. K. Schroder, "MOSFET Mobility," in *Semiconductor Material and Device Characterization*, 3rd ed. Hoboken: Wiley, 2006, ch. 8, sec. 6, pp. 500.

[11] R. D. Knight, "Current and Resistance," in *Physics for Scientists and Engineers a Strategic Approach*, 2nd ed. San Francisco, Pearson, 2008, ch. 31 sec. 4, pp. 954-955.

[12] A. O. Adan and K.Hisgashi, "OFF-State Leakage Current Mechanisms in BulkSi and SOI MOSFETs and Their Impact On CMOS ULSIs Standby Current," *IEEE Trans. Electron Devices*, vol.48, no. 9 pp.2050-2057, Sep. 2001.

[13] R.Singh *et al.*, "Analysis of the Effect of Temperature Variations on Sub-threshold Leakage Current in P3 and P4 SRAM Cells at Deep Sub-micron CMOS Technology," *International Journal of Computer Applications*, vol.35, no.5, pp. 8-13, Dec. 2011.

[14] *BSIMSOIv4.4 MOSFET MODEL Users' Manual*, Department of Elect. Eng. and Comput. Sci., University of California Berkeley, Berkeley, CA, 2010.

[15] D. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," in *J. of the Soc. for Ind, and Appl. Math.* vol.11 no.2, pp. 431-441, June, 1962.

Appendices

The following appendices include tutorials, equipment descriptions, and code used to complete this study. These appendices are intended to be supplemental documents for all of the procedures summarized.

Appendix A
Cadence Simulation Tutorial

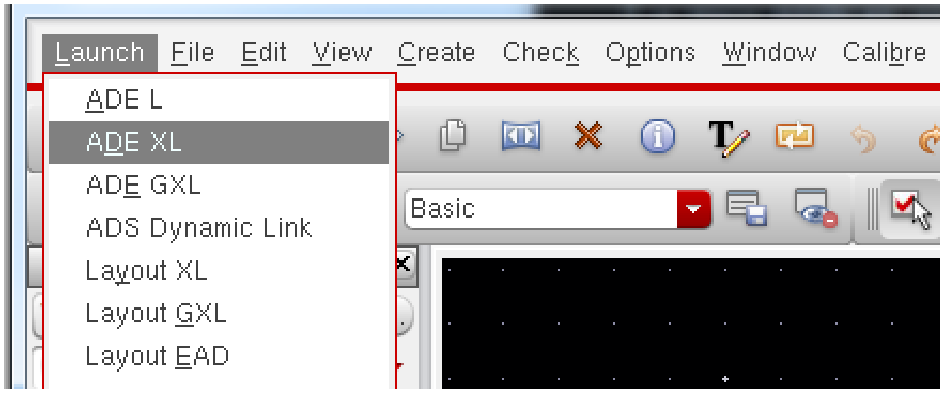
The following appendix will cover the procedure for performing a DC analysis on a single transistor using Cadence Virtuoso.

A.1 Simulation Steps

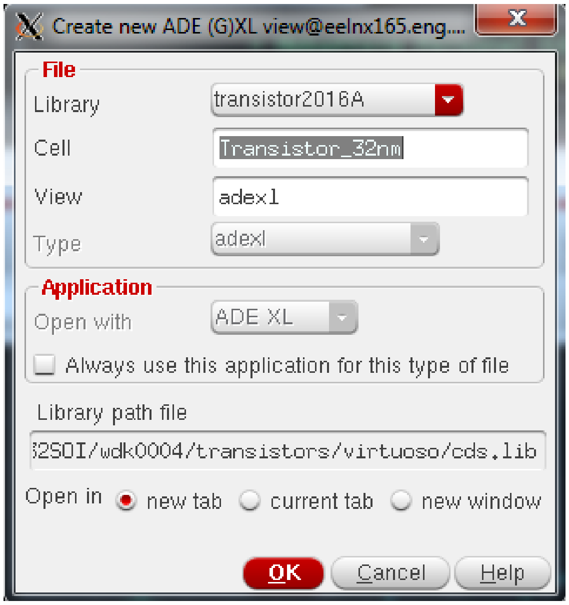
1. Launch ADE XL
2. Create new tests
3. Set Global Variables and Corners
4. Run simulation and view data

A.1.1 Launching ADE XL

1. ADE XL is launched from the schematic window of the design.



2. Create a new view for the current schematic



File

Library: transistor2016A

Cell: Transistor_32nm

View: adexl

Type: adexl

Application

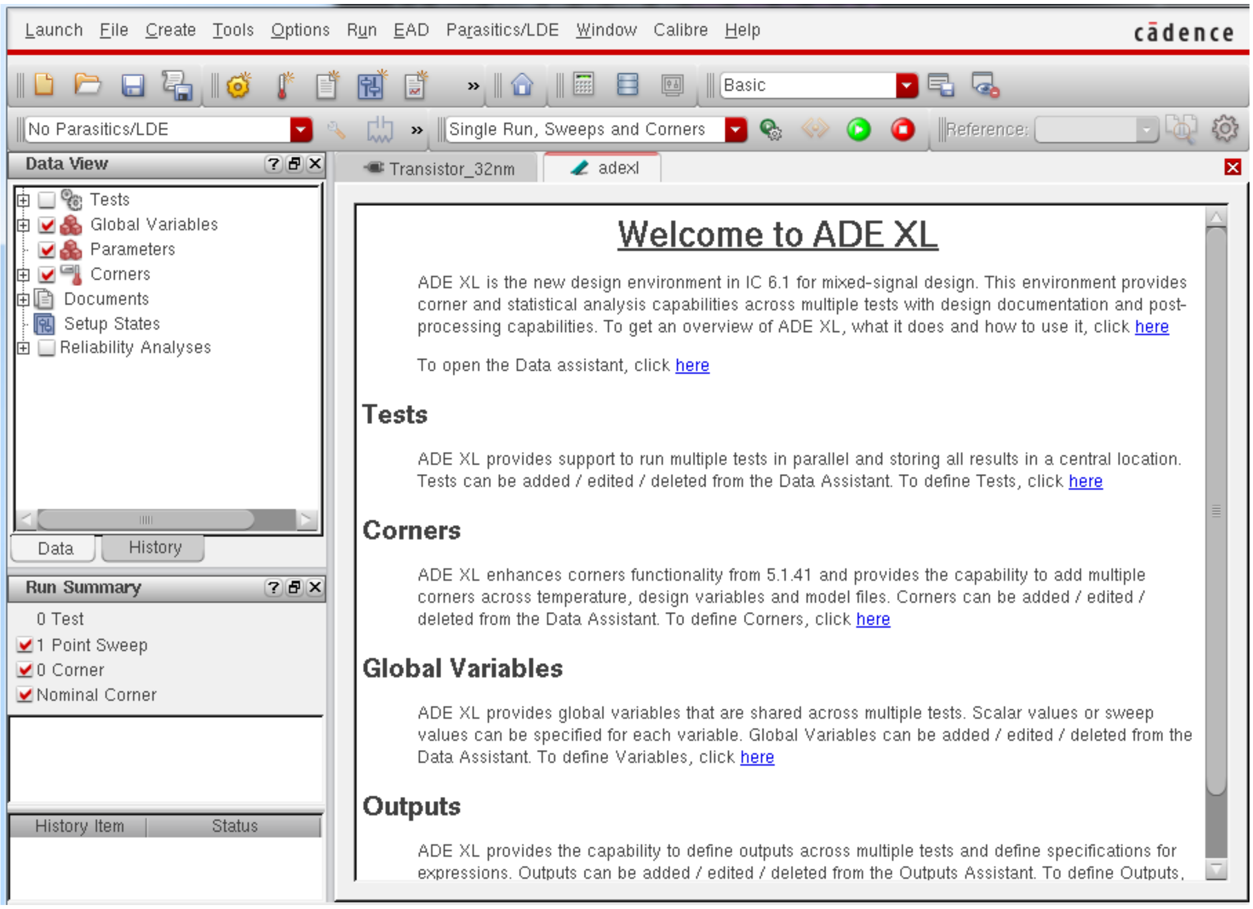
Open with: ADE XL

☐ Always use this application for this type of file

Library path file: 32SOI/wdk0004/transistors/virtuoso/cds.lib

Open in ● new tab ○ current tab ○ new window

3. ADE XL start screen



Data View:

- ☐ Tests
- ☑ Global Variables
- ☑ Parameters
- ☑ Corners
- Documents
- Setup States
- ☐ Reliability Analyses

Run Summary:

0 Test
☑ 1 Point Sweep
☑ 0 Corner
☑ Nominal Corner

Welcome to ADE XL

ADE XL is the new design environment in IC 6.1 for mixed-signal design. This environment provides corner and statistical analysis capabilities across multiple tests with design documentation and post-processing capabilities. To get an overview of ADE XL, what it does and how to use it, click here

To open the Data assistant, click here

Tests

ADE XL provides support to run multiple tests in parallel and storing all results in a central location. Tests can be added / edited / deleted from the Data Assistant. To define Tests, click here

Corners

ADE XL enhances corners functionality from 5.1.41 and provides the capability to add multiple corners across temperature, design variables and model files. Corners can be added / edited / deleted from the Data Assistant. To define Corners, click here

Global Variables

ADE XL provides global variables that are shared across multiple tests. Scalar values or sweep values can be specified for each variable. Global Variables can be added / edited / deleted from the Data Assistant. To define Variables, click here

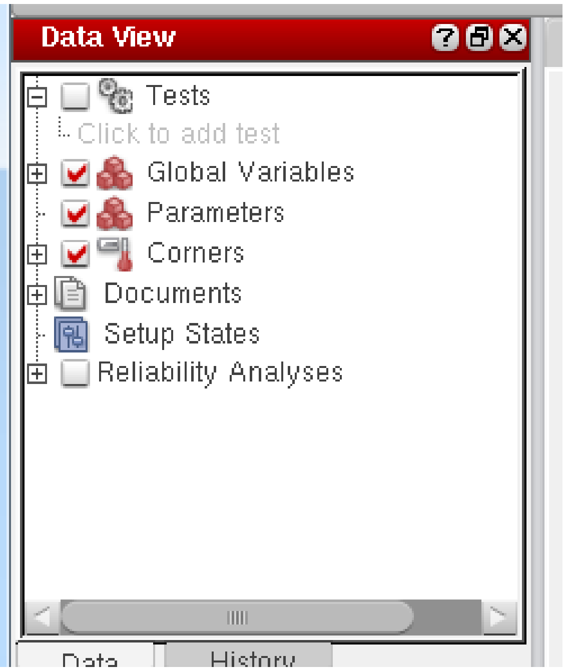
Outputs

ADE XL provides the capability to define outputs across multiple tests and define specifications for expressions. Outputs can be added / edited / deleted from the Outputs Assistant. To define Outputs,

A.1.2 Creating a new test

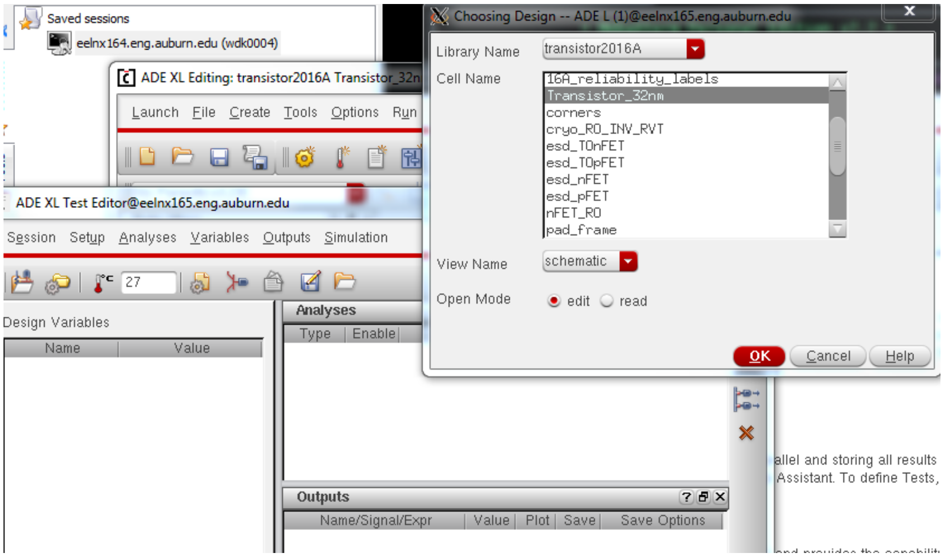
1. Adding a new test

   Add a new test by clicking under ”Tests” on the left side.



2. Selecting the reference design

   Select the schematic that is used as a reference.


Saved sessions
eelnx164.eng.auburn.edu (wdk0004)
ADE XL Editing: transistor2016A Transistor_32n
Launch File Create Tools Options Run
ADE XL Test Editor@eelnx165.eng.auburn.edu
Session Setup Analyses Variables Outputs Simulation
27
Design Variables
Name Value
Analyses
Type Enable
Outputs
Name/Signal/Expr Value Plot Save Save Options
allel and storing all results
Assistant. To define Tests,
Choosing Design -- ADE L (1)@eelnx165.eng.auburn.edu
OK Cancel Help

Library Name: transistor2016A

Cell Name:

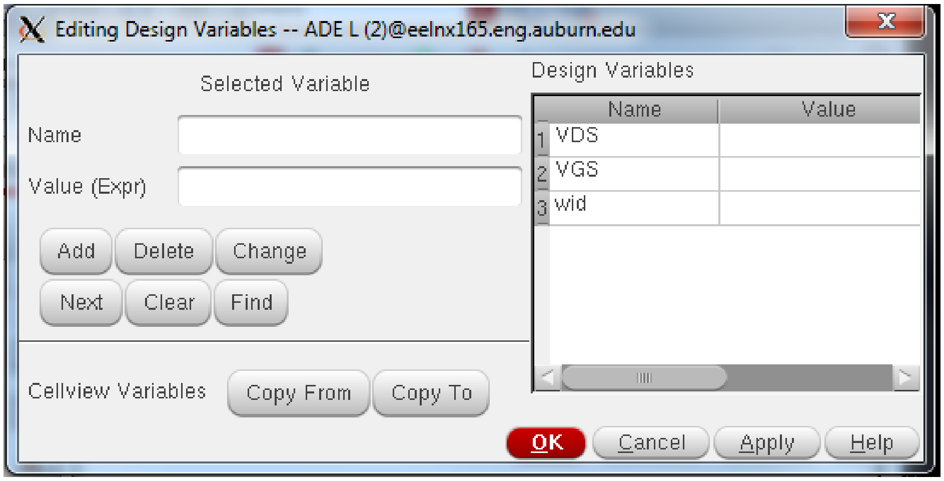
```
16A_reliability_labels
Transistor_32nm
corners
cryo_RO_INV_RVT
esd_TOnFET
esd_TOpFET
esd_nFET
esd_pFET
nFET_RO
pad_frame
```

View Name: schematic

Open Mode: ● edit ○ read

3. Adding variables

   Add variables by clicking "Copy From" under the "Variables" tab of the test window.



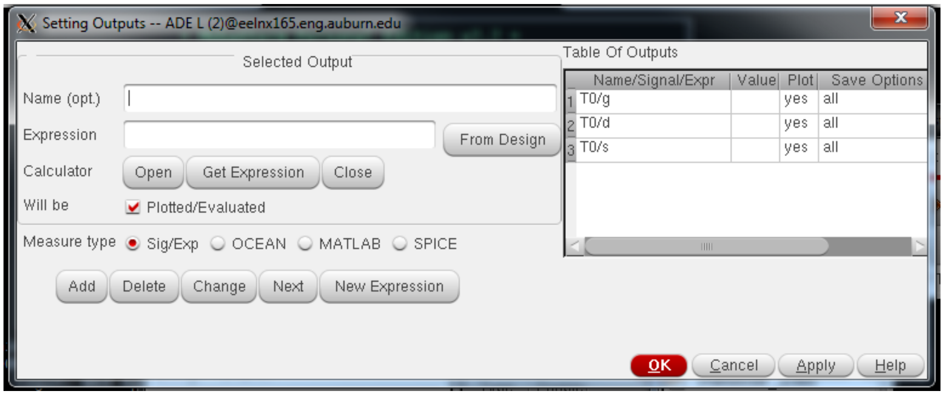
Editing Design Variables -- ADE L (2)@eelnx165.eng.auburn.edu

Design Variables

	Name	Value
1	VDS	
2	VGS	
3	wid	

4. Adding outputs

   Add output by clicking "From Design" under the "Output" tab of the test window and selecting the components of interest



Setting Outputs -- ADE L (2)@eelnx165.eng.auburn.edu

Will be ☑ Plotted/Evaluated

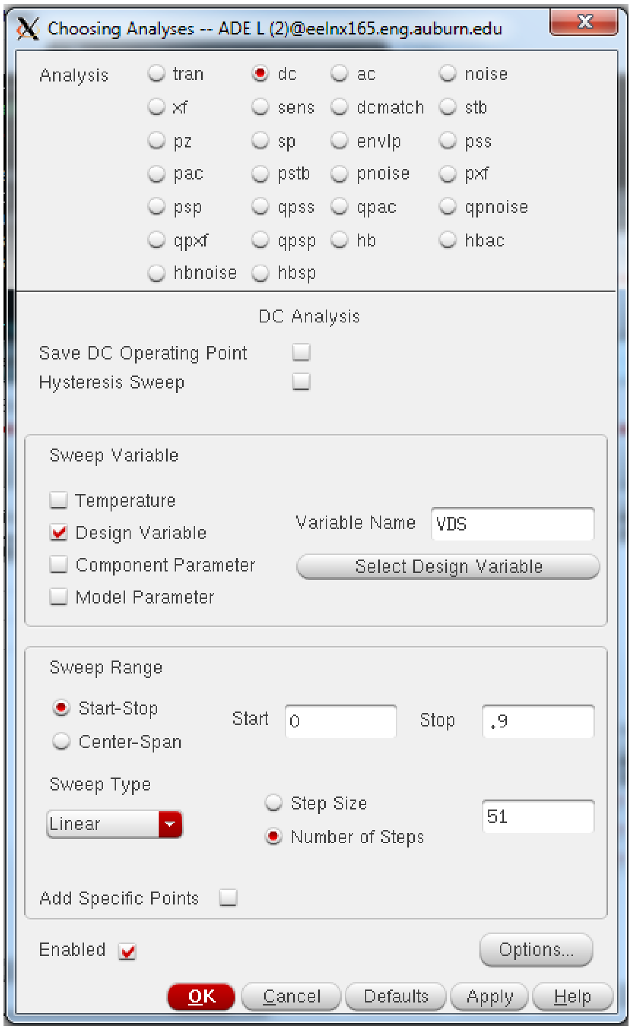
Measure type ● Sig/Exp ○ OCEAN ○ MATLAB ○ SPICE

Table Of Outputs

	Name/Signal/Expr	Value	Plot	Save Options
1	T0/g		yes	all
2	T0/d		yes	all
3	T0/s		yes	all

5. Selecting analysis

   Select the analysis under the ”Analysis” tab of the test window. For the transistor simulations the analysis used is dc.



Choosing Analyses -- ADE L (2)@eelnx165.eng.auburn.edu

Analysis ○ tran ● dc ○ ac ○ noise ○ xf ○ sens ○ dcmatch ○ stb ○ pz ○ sp ○ envlp ○ pss ○ pac ○ pstb ○ pnoise ○ pxf ○ psp ○ qpss ○ qpac ○ qpnoise ○ qpxf ○ qpsp ○ hb ○ hbac ○ hbnoise ○ hbsp

DC Analysis

Save DC Operating Point □

Hysteresis Sweep □

Sweep Variable

□ Temperature

☑ Design Variable

□ Component Parameter

□ Model Parameter

Variable Name VDS

Sweep Range

● Start-Stop

○ Center-Span

Start 0 Stop .9

Sweep Type Linear

○ Step Size

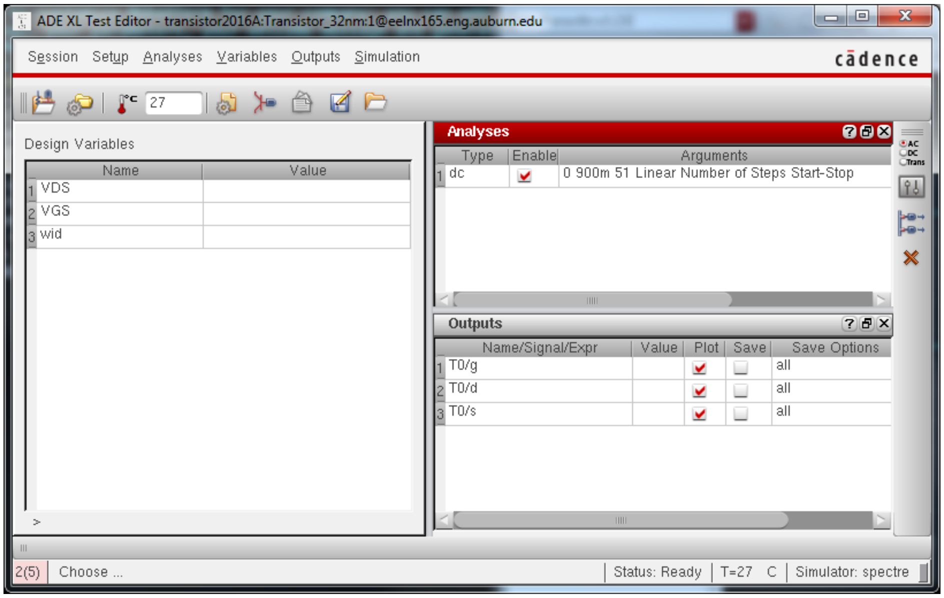
● Number of Steps 51

Add Specific Points □

Enabled ☑

6. Selecting output plots

   Select output plots checking the box in the ”Output” section of the test window.



ADE XL Test Editor - transistor2016A:Transistor_32nm:1@eelnx165.eng.auburn.edu

Design Variables

	Name	Value
1	VDS	
2	VGS	
3	wid	

Analyses

	Type	Enable	Arguments
1	dc	☑	0 900m 51 Linear Number of Steps Start-Stop

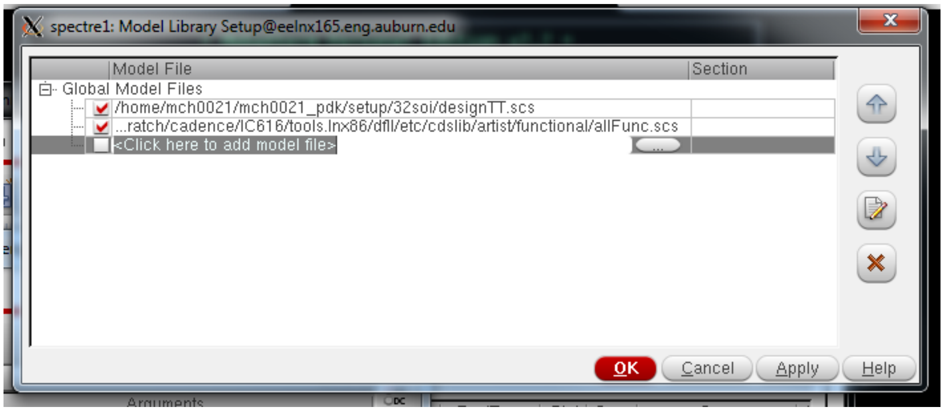
Outputs

	Name/Signal/Expr	Value	Plot	Save	Save Options
1	T0/g		☑	□	all
2	T0/d		☑	□	all
3	T0/s		☑	□	all

2(5) Choose ... Status: Ready T=27 C Simulator: spectre

7. Changing Model Libraries (Optional)

   Change the model library under the ”Setup” tab of the test window. (The default model library only needs to be changed when using cryogenic models.)



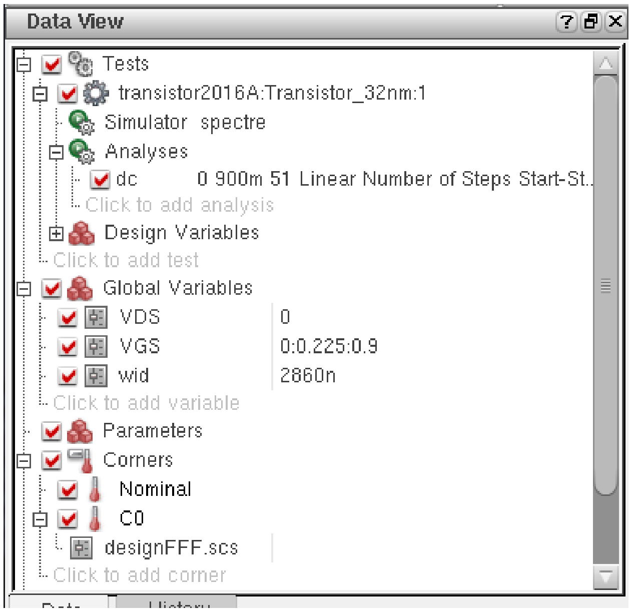
spectre1: Model Library Setup@eelnx165.eng.auburn.edu

Model File	Section
Global Model Files	
☑ /home/mch0021/mch0021_pdk/setup/32soi/designTT.scs	
☑ ...ratch/cadence/IC616/tools.lnx86/dfII/etc/cdslib/artist/functional/allFunc.scs	
□ <Click here to add model file>	

A.1.3 Setting Global Variable and Corners

1. Setting global variables

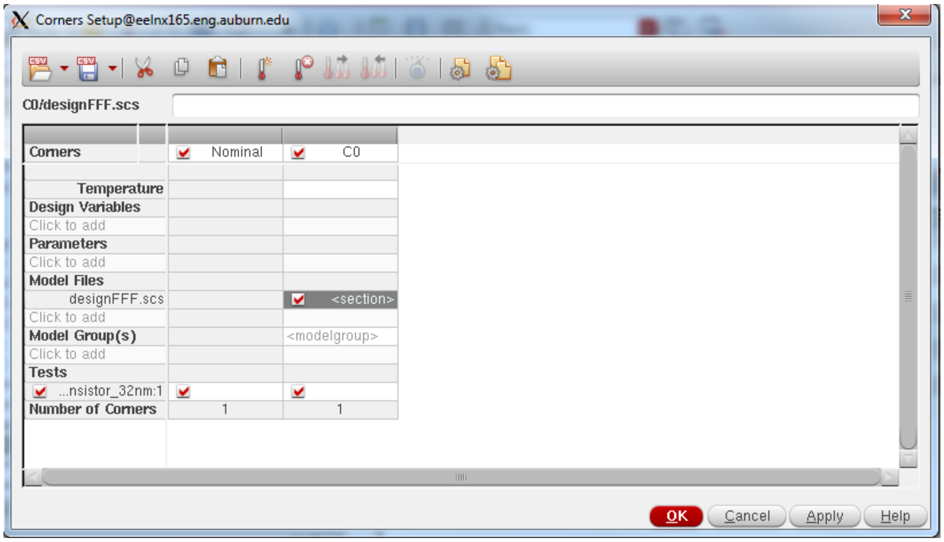
   Set global variables on the left side of the page. 0:0.225:0.9 steps the variable from 0 to 0.9 over 0.225 invervals.



```
☑ Tests
  ☑ transistor2016A:Transistor_32nm:1
    Simulator  spectre
    Analyses
      ☑ dc        0 900m 51 Linear Number of Steps Start-St..
      Click to add analysis
    Design Variables
  Click to add test
☑ Global Variables
  ☑ VDS        0
  ☑ VGS        0:0.225:0.9
  ☑ wid        2860n
  Click to add variable
☑ Parameters
☑ Corners
  ☑ Nominal
  ☑ C0
    designFFF.scs
  Click to add corner
```

2. Adding multiple corners

   Add a new corner by clicking under ”Corners” on the left side of the page. By clicking the thermometer with the asterisk at the top of the page a new column is added to the corner window. Then add the corner model file need. Corner model files for 32nm can be found under mch0021_group>lowtemp_group>32SOI>Corners.

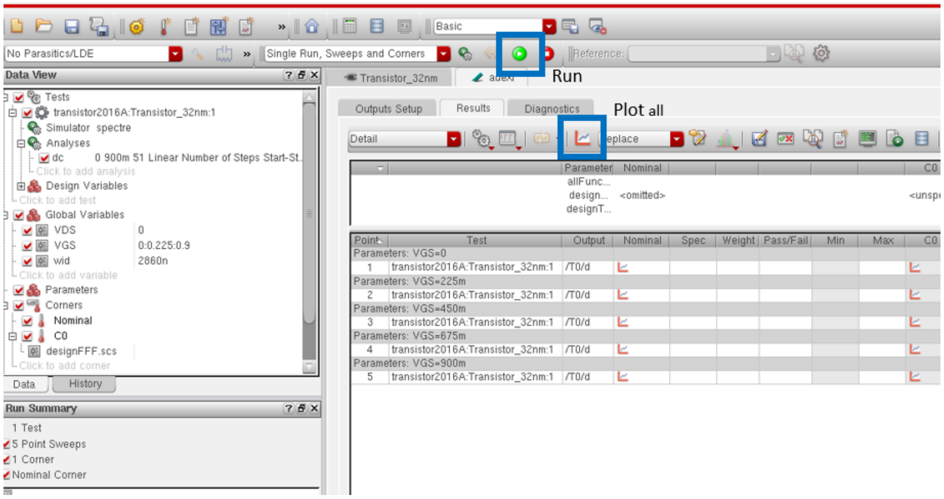


C0/designFFF.scs

Corners	✓ Nominal	✓ C0
Temperature		
Design Variables		
Click to add		
Parameters		
Click to add		
Model Files		
designFFF.scs		☑ <section>
Click to add		
Model Group(s)		<modelgroup>
Click to add		
Tests		
✓ ...nsistor_32nm:1	✓	✓
Number of Corners	1	1

A.1.4 Running Simulation and Viewing Data

1. Run simulation by clicking the green arrow at the top of the screen.



Data View

- Tests
  - transistor2016A:Transistor_32nm:1
    - Simulator spectre
    - Analyses
      - dc 0 900m 51 Linear Number of Steps Start-St...
      - Click to add analysis
    - Design Variables
  - Click to add test
- Global Variables
  - VDS 0
  - VGS 0:0.225:0.9
  - wid 2860n
  - Click to add variable
- Parameters
- Corners
  - Nominal
  - C0
    - designFFF.scs
  - Click to add corner

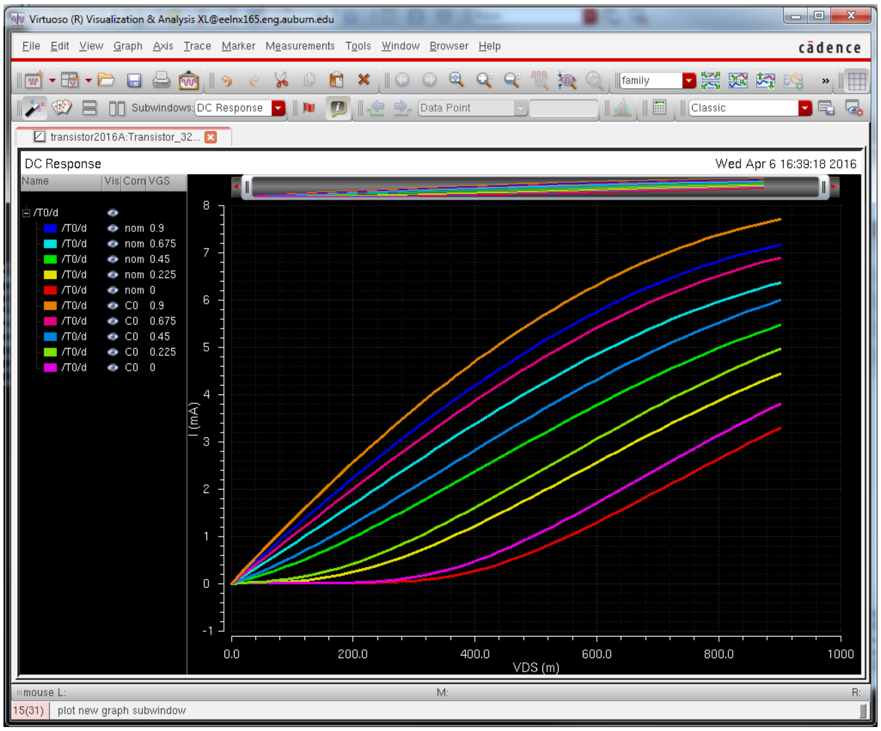
Run Summary

1 Test
5 Point Sweeps
1 Corner
Nominal Corner

	Parameter	Nominal	C0
	allFunc...		
	design...	<omitted>	<unsp
	designT...		

Point	Test	Output	Nominal	Spec	Weight	Pass/Fail	Min	Max	C0
Parameters: VGS=0									
1	transistor2016A:Transistor_32nm:1	/T0/d	📈						📈
Parameters: VGS=225m									
2	transistor2016A:Transistor_32nm:1	/T0/d	📈						📈
Parameters: VGS=450m									
3	transistor2016A:Transistor_32nm:1	/T0/d	📈						📈
Parameters: VGS=675m									
4	transistor2016A:Transistor_32nm:1	/T0/d	📈						📈
Parameters: VGS=900m									
5	transistor2016A:Transistor_32nm:1	/T0/d	📈						📈

2. View all outputs plot by clicking ”Plot all” as shown above


Virtuoso (R) Visualization & Analysis XL@eelnx165.eng.auburn.edu
File Edit View Graph Axis Trace Marker Measurements Tools Window Browser Help
cādence
family
Subwindows: DC Response
Data Point
Classic
transistor2016A:Transistor_32...
8
7
6
5
4
3
2
1
0
-1
I (mA)
0.0
200.0
400.0
600.0
800.0
1000
VDS (m)
mouse L:
M:
R:

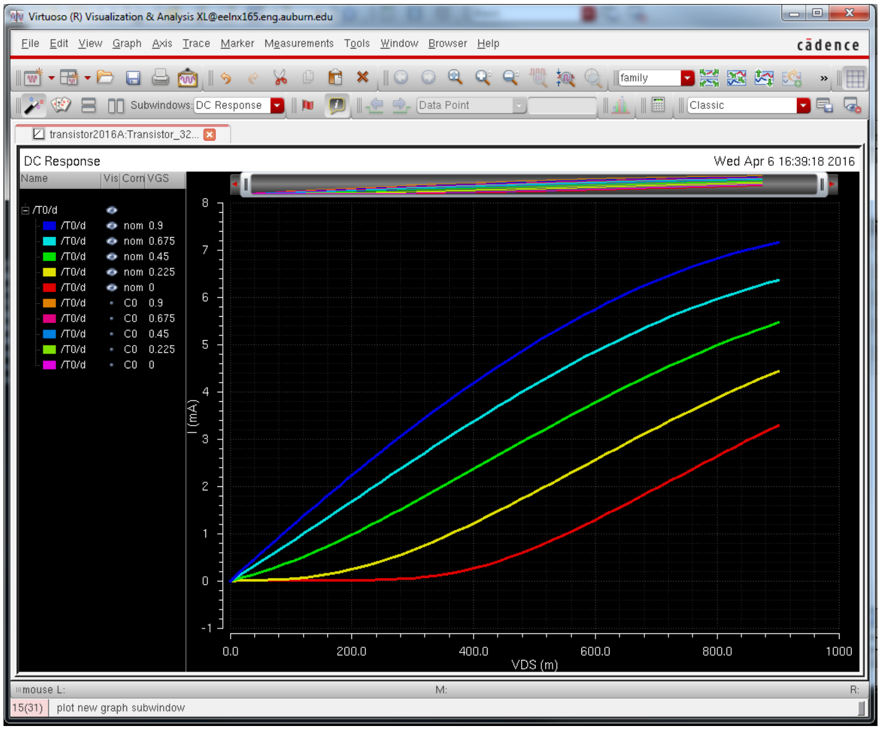
DC Response

Wed Apr 6 16:39:18 2016

Name	Vis	Corn	VGS
/T0/d	👁		
🟦 /T0/d	👁	nom	0.9
/T0/d	👁	nom	0.675
🟩 /T0/d	👁	nom	0.45
🟨 /T0/d	👁	nom	0.225
🟥 /T0/d	👁	nom	0
🟧 /T0/d	👁	C0	0.9
/T0/d	👁	C0	0.675
🟦 /T0/d	👁	C0	0.45
🟩 /T0/d	👁	C0	0.225
🟪 /T0/d	👁	C0	0

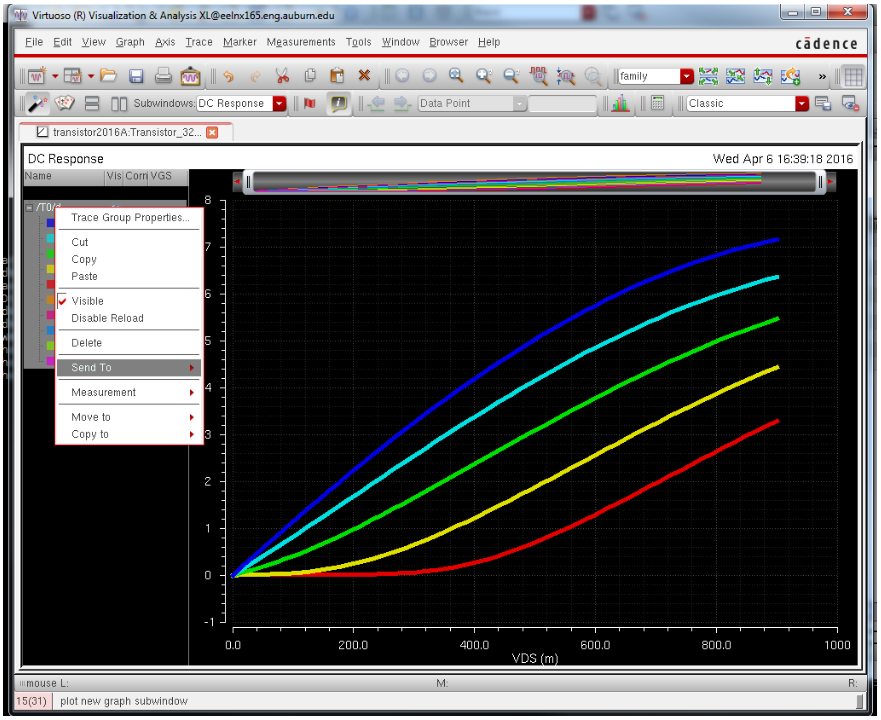
15(31) plot new graph subwindow

3. Toggle output views



plot new graph subwindow

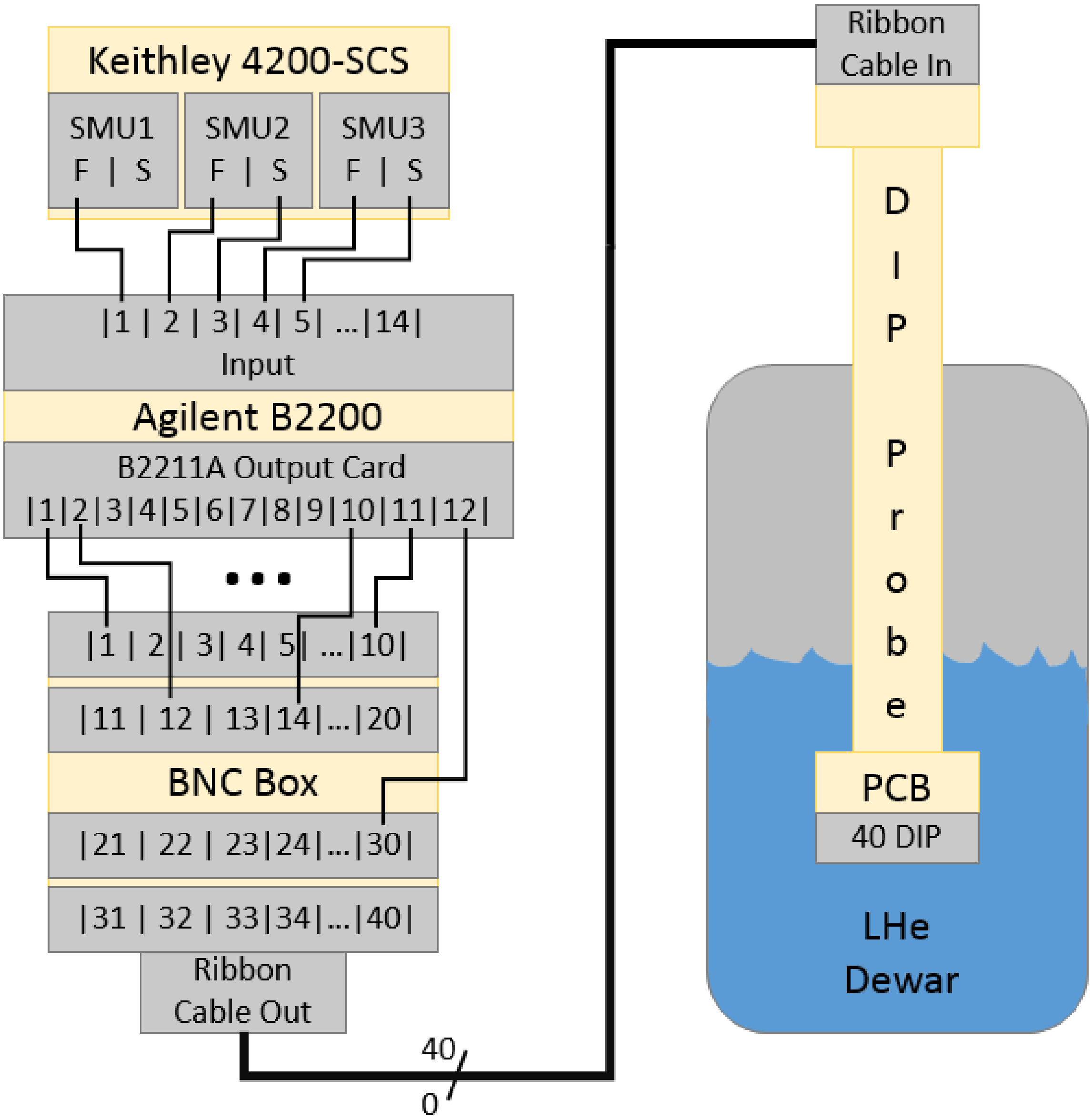
4. By right clicking on the data, the data points plotted can be sent to a table that is saved as CSV file.



plot new graph subwindow

Appendix B
Testing Equipment and Procedures

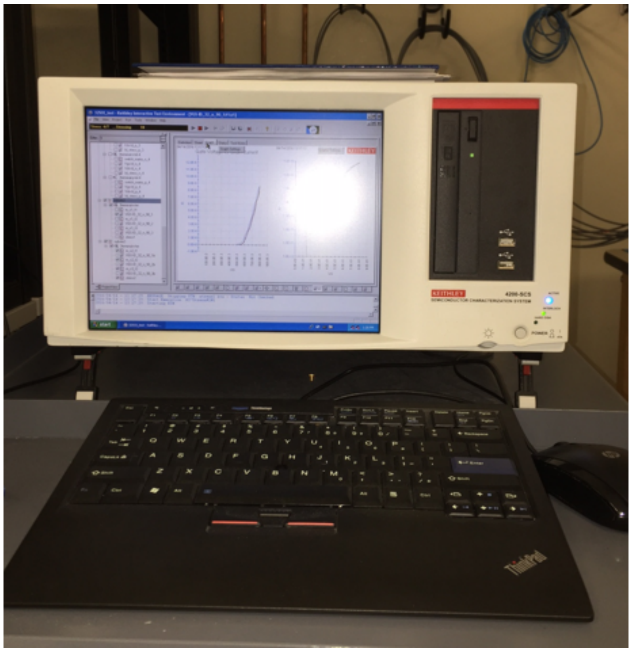
All of the equipment, settings, and code used to perform all measurements will be displayed. A diagram detailing the interconnections between the equipment is shown below. When performing RT measurements, the connection are the same, but the probe was left in the holding rack on the lab table.



B.1 Testing Equipment

1. Keithley 4200-SCS Parameter Analyzer

   3 SMUs are used to perform the 4 point measurements on the transistors. SMU 1 and 2 both have preamps connected that extend the measurement resolution down to 0.1fA. These two SMUs are connected to the gate and drain so that the current measured has higher precision.

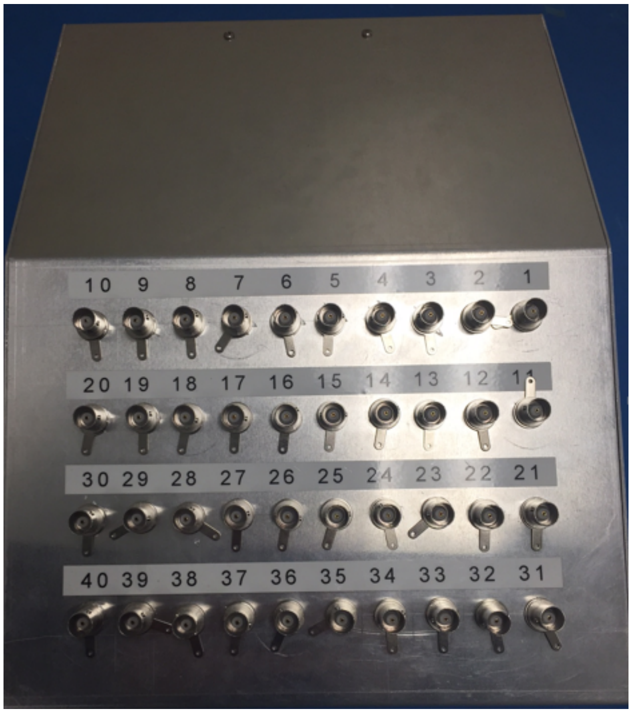


2. Aglient B2200 Low Leakage Switch Matrix

   The SMUs from the 4200 connect to the first 5 inputs on the front of the B2200. The cables coming from the BNC box connect to the B2211A cards connected to the output of the B2200.

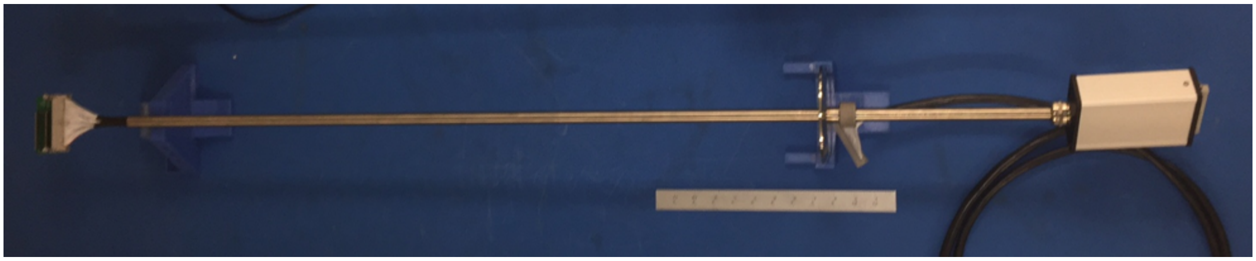
3. BNC Box

   The BNC box is used to connect the ribbon cable from the probe to the triax cables that connect to the output of the B2200.



4. Cryogenic probe

   Cryogenic probe with 40 pin DIP socket attached the a PCB board on the bottom.



5. 100L Cryofab dewar

   Dewars are filled with LHe allowing the probe to be submerged enabling cryogenic testing to be performed.

B.2 Measurement Process Steps

There are 5 steps taken to perform a stress measurement in LHe:

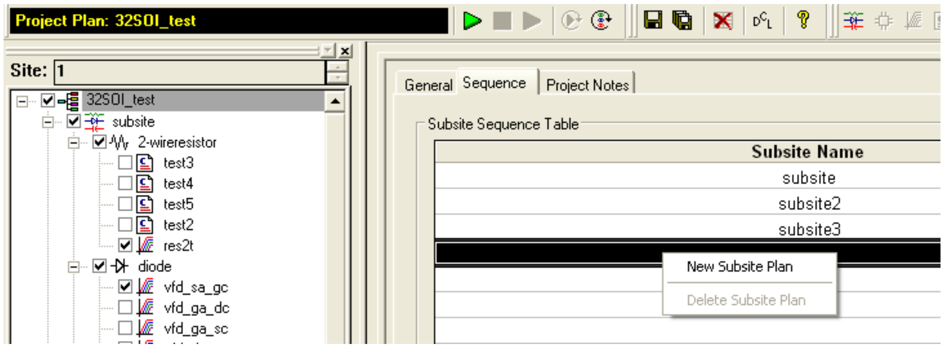
1. Standard at RT
2. Pre-cool/Standard in liquid nitrogen
3. Standard in LHe
4. Stress in LHe
5. Post in LHe

To perform a RT stress measurements, omit steps 2 and 3, and perform steps 4 and 5 at RT. Using reliability chips, multiple devices can be stressed in parallel. In order to stress devices in parallel, steps 1, 2, and 3 on devices independently, then using the B2200 and the code in Section B.4 step 4 is performed automatically on all the devices. After, the stress is finished step 5 is performed on each device independently.

B.3 Kite Settings

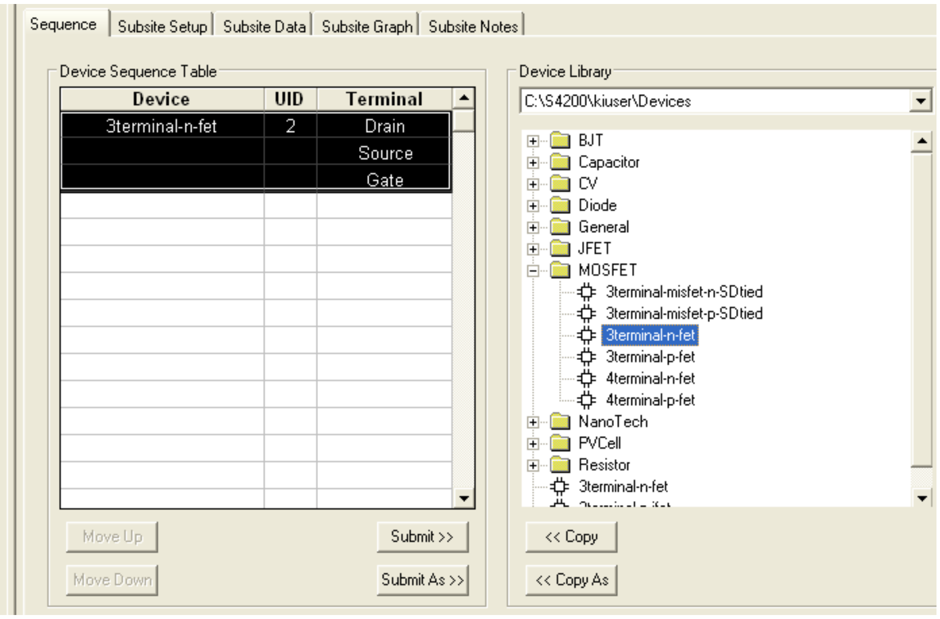
1. Creating Subsites

   On the 4200 test plans are called Subsites. Subsites are added to the project sequence by right clicking on a new cell under the "Sequence" tab.



Subsite Name
subsite
subsite2
subsite3

For performing the reliability measure two subsites are needed. The first subsite includes the standard/post I_D-V_{DS} and I_D-V_{GS} measurements. The second subsite is used to perform the stress and make the periodic I_D-V_{GS} measurement. New test for transistor testing are added by copying the 3 terminal-FET test over from the MOSFET device sections built into KITE to the selected subsite.



Sequence | Subsite Setup | Subsite Data | Subsite Graph | Subsite Notes

Device Sequence Table

Device	UID	Terminal
3terminal-n-fet	2	Drain
		Source
		Gate

Device Library

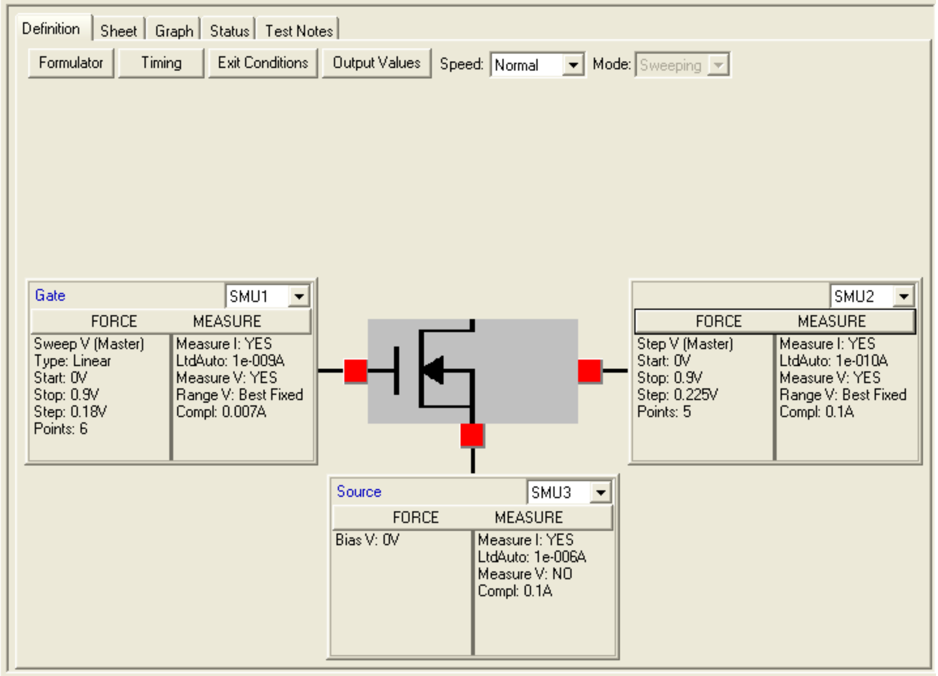
C:\S4200\kiuser\Devices

- BJT
- Capacitor
- CV
- Diode
- General
- JFET
- MOSFET
  - 3terminal-misfet-n-SDtied
  - 3terminal-misfet-p-SDtied
  - 3terminal-n-fet
  - 3terminal-p-fet
  - 4terminal-n-fet
  - 4terminal-p-fet
- NanoTech
- PVCell
- Resistor
- 3terminal-n-fet

When performing stress measurements in parallel, functions are added in between the I_D-V_{GS} measurements to switch the B2200 to the correct connections. All of the settings are shown of the Subsites are shown below

(a) Subsite 1

I_D-V_{GS} settings



Definition | Sheet | Graph | Status | Test Notes

Formulator | Timing | Exit Conditions | Output Values | Speed: Normal | Mode: Sweeping

Gate — SMU1

FORCE	MEASURE
Sweep V (Master) Type: Linear Start: 0V Stop: 0.9V Step: 0.18V Points: 6	Measure I: YES LtdAuto: 1e-009A Measure V: YES Range V: Best Fixed Compl: 0.007A

SMU2

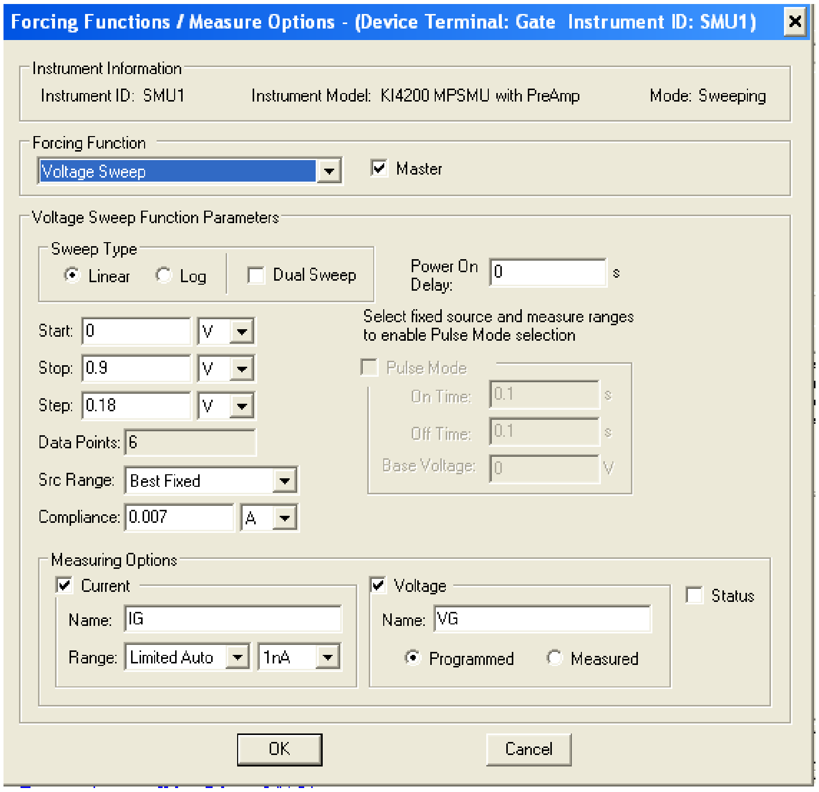
FORCE	MEASURE
Step V (Master) Start: 0V Stop: 0.9V Step: 0.225V Points: 5	Measure I: YES LtdAuto: 1e-010A Measure V: YES Range V: Best Fixed Compl: 0.1A

Source — SMU3

FORCE	MEASURE
Bias V: 0V	Measure I: YES LtdAuto: 1e-006A Measure V: NO Compl: 0.1A

By clicking the "Force/Measure" button on the SMU description the settings can be modified.

SMU 1



Forcing Functions / Measure Options - (Device Terminal: Gate Instrument ID: SMU1) ☒

Instrument Information

Instrument ID: SMU1 Instrument Model: KI4200 MPSMU with PreAmp Mode: Sweeping

Forcing Function

Voltage Sweep ☑ Master

Voltage Sweep Function Parameters

Sweep Type: ● Linear ○ Log □ Dual Sweep

Power On Delay: 0 s

Select fixed source and measure ranges to enable Pulse Mode selection

Start:	0	V
Stop:	0.9	V
Step:	0.18	V
Data Points:	6	
Src Range:	Best Fixed	
Compliance:	0.007	A

□ Pulse Mode

On Time: 0.1 s

Off Time: 0.1 s

Base Voltage: 0 V

Measuring Options

☑ Current

Name: IG

Range: Limited Auto 1nA

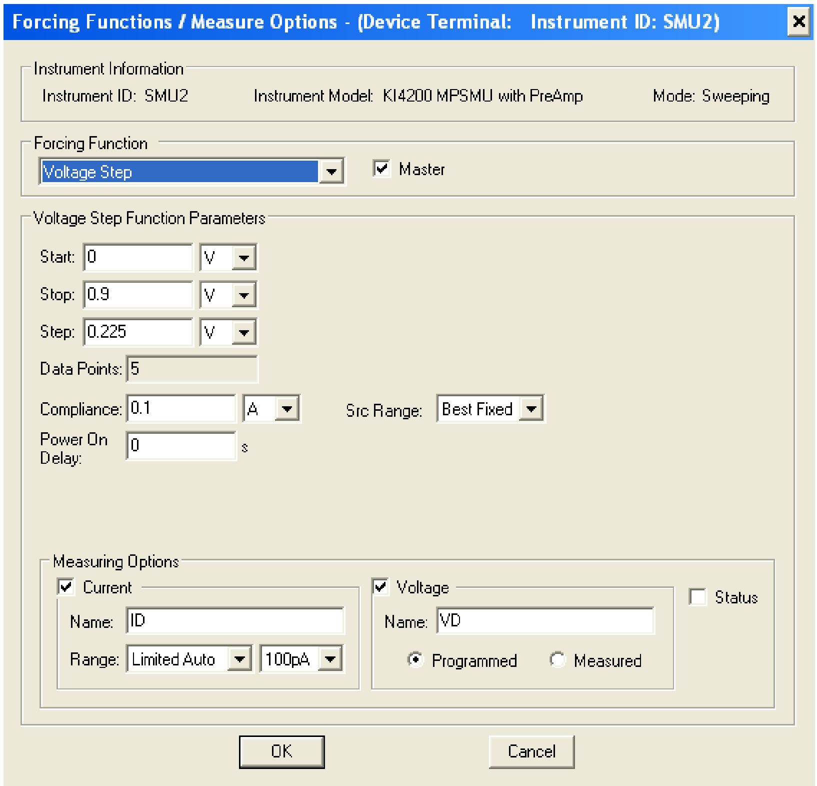
☑ Voltage

Name: VG

● Programmed ○ Measured

□ Status

SMU 2



Forcing Functions / Measure Options - (Device Terminal: Instrument ID: SMU2) ☒

Instrument Information

Instrument ID: SMU2 Instrument Model: KI4200 MPSMU with PreAmp Mode: Sweeping

Forcing Function

Voltage Step ☑ Master

Voltage Step Function Parameters

Start:	0	V
Stop:	0.9	V
Step:	0.225	V
Data Points:	5	
Compliance:	0.1	A

Src Range: Best Fixed

Power On Delay: 0 s

Measuring Options

☑ Current

Name: ID

Range: Limited Auto 100pA

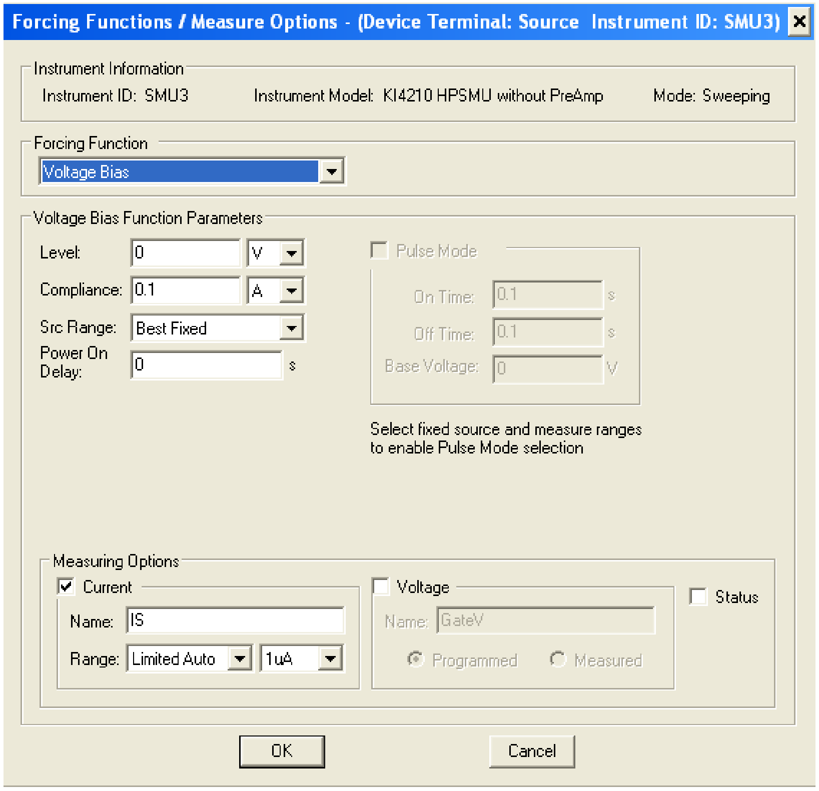
☑ Voltage

Name: VD

● Programmed ○ Measured

□ Status

SMU 3


Forcing Functions / Measure Options - (Device Terminal: Source Instrument ID: SMU3)

Instrument Information
Instrument ID: SMU3
Instrument Model: KI4210 HPSMU without PreAmp
Mode: Sweeping

Forcing Function
Voltage Bias

Voltage Bias Function Parameters
Level: 0 V
Compliance: 0.1 A
Src Range: Best Fixed
Power On Delay: 0 s

☐ Pulse Mode
On Time: 0.1 s
Off Time: 0.1 s
Base Voltage: 0 V

Select fixed source and measure ranges to enable Pulse Mode selection

Measuring Options
☑ Current
Name: IS
Range: Limited Auto 1uA

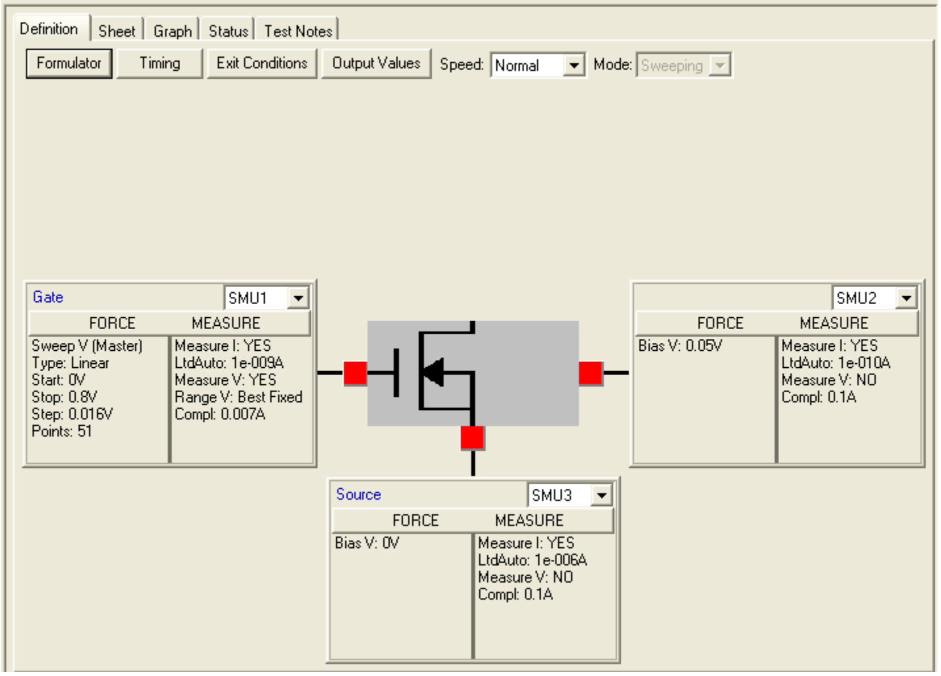
☐ Voltage
Name: GateV
● Programmed ○ Measured

☐ Status

The settings of SMU 1 and 2 are swapped when performing I_D-V_{GS} measurements.

(b) Subsite 2

I_D-V_{GS} linear regime settings



Speed: Normal
Mode: Sweeping

Gate SMU1

FORCE	MEASURE
Sweep V (Master) Type: Linear Start: 0V Stop: 0.8V Step: 0.016V Points: 51	Measure I: YES LtdAuto: 1e-009A Measure V: YES Range V: Best Fixed Compl: 0.007A

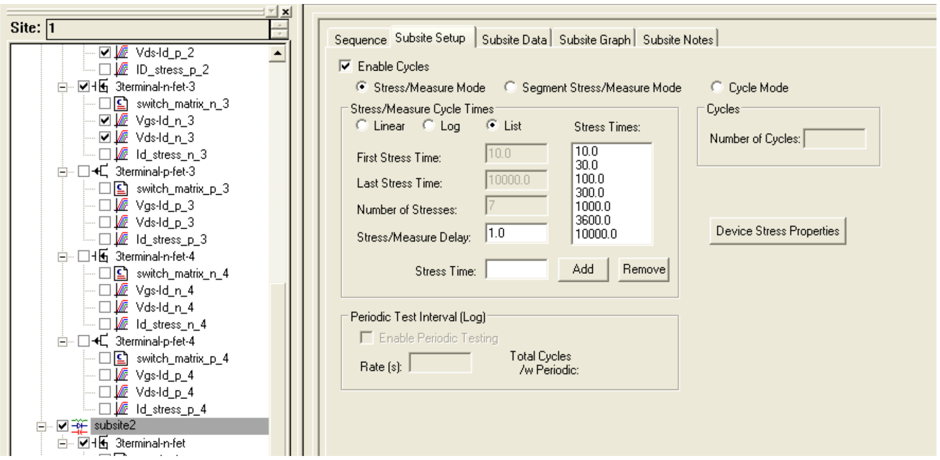
SMU2

FORCE	MEASURE
Bias V: 0.05V	Measure I: YES LtdAuto: 1e-010A Measure V: NO Compl: 0.1A

Source SMU3

FORCE	MEASURE
Bias V: 0V	Measure I: YES LtdAuto: 1e-006A Measure V: NO Compl: 0.1A

Under the "Subsite Setup" tab, cycling can be enabled the periodic measurement times are added.



Site: 1

- ☑ Vds-Id_p_2
- □ ID_stress_p_2
- ☑ 3terminal-n-fet-3
  - □ switch_matrix_n_3
  - ☑ Vgs-Id_n_3
  - ☑ Vds-Id_n_3
  - □ Id_stress_n_3
- □ 3terminal-p-fet-3
  - □ switch_matrix_p_3
  - □ Vgs-Id_p_3
  - □ Vds-Id_p_3
  - □ Id_stress_p_3
- □ 3terminal-n-fet-4
  - □ switch_matrix_n_4
  - □ Vgs-Id_n_4
  - □ Vds-Id_n_4
  - □ Id_stress_n_4
- □ 3terminal-p-fet-4
  - □ switch_matrix_p_4
  - □ Vgs-Id_p_4
  - □ Vds-Id_p_4
  - □ Id_stress_p_4
- ☑ subsite2
  - ☑ 3terminal-n-fet

Sequence | Subsite Setup | Subsite Data | Subsite Graph | Subsite Notes

☑ Enable Cycles

● Stress/Measure Mode ○ Segment Stress/Measure Mode ○ Cycle Mode

Stress/Measure Cycle Times

○ Linear ○ Log ● List

First Stress Time: 10.0

Last Stress Time: 10000.0

Number of Stresses: 7

Stress/Measure Delay: 1.0

Stress Times: 10.0, 30.0, 100.0, 300.0, 1000.0, 3600.0, 10000.0

Stress Time: Add Remove

Cycles

Number of Cycles:

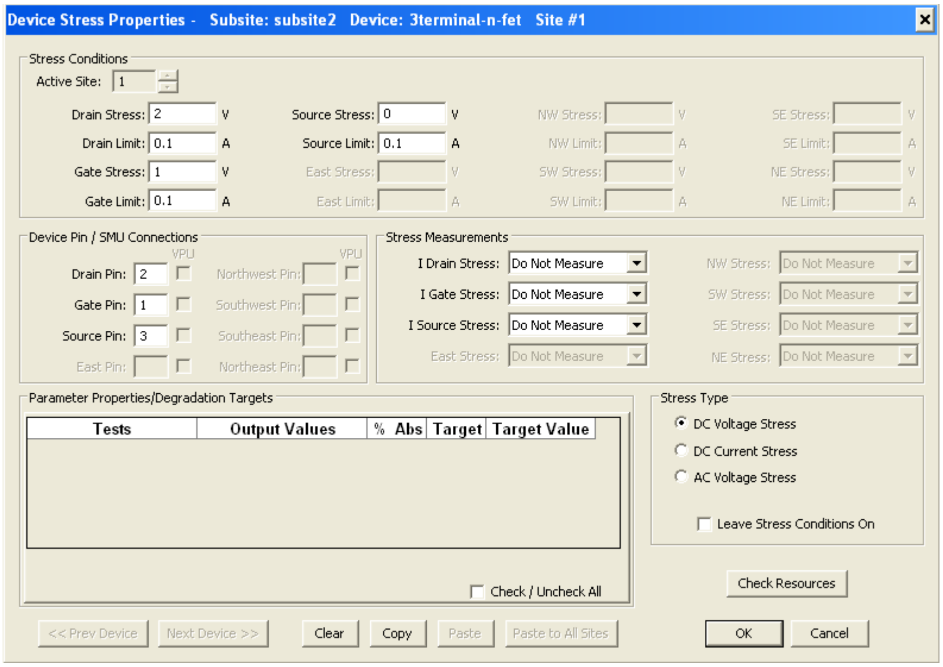
Device Stress Properties

Periodic Test Interval (Log)

□ Enable Periodic Testing

Rate (s): Total Cycles /w Periodic:

By clicking "Device Stress Properties", the voltages of the drain and gate stresses can be set and the Pin/SMU connections are listed.



Device Stress Properties - Subsite: subsite2 Device: 3terminal-n-fet Site #1

Stress Conditions

Active Site: 1

Drain Stress: 2 V	Source Stress: 0 V	NW Stress: V	SE Stress: V
Drain Limit: 0.1 A	Source Limit: 0.1 A	NW Limit: A	SE Limit: A
Gate Stress: 1 V	East Stress: V	SW Stress: V	NE Stress: V
Gate Limit: 0.1 A	East Limit: A	SW Limit: A	NE Limit: A

Device Pin / SMU Connections

	VPU		VPU
Drain Pin: 2	□	Northwest Pin:	□
Gate Pin: 1	□	Southwest Pin:	□
Source Pin: 3	□	Southeast Pin:	□
East Pin:	□	Northeast Pin:	□

Stress Measurements

I Drain Stress: Do Not Measure	NW Stress: Do Not Measure
I Gate Stress: Do Not Measure	SW Stress: Do Not Measure
I Source Stress: Do Not Measure	SE Stress: Do Not Measure
East Stress: Do Not Measure	NE Stress: Do Not Measure

Parameter Properties/Degradation Targets

Tests	Output Values	%	Abs	Target	Target Value

□ Check / Uncheck All

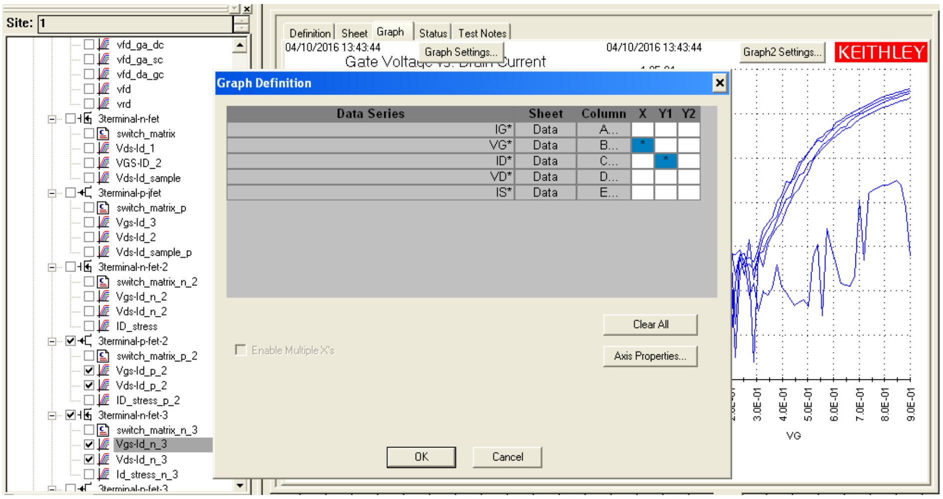
Stress Type

● DC Voltage Stress
○ DC Current Stress
○ AC Voltage Stress

□ Leave Stress Conditions On

2. Setting up plots

   By right clicking on the plot under the graph tab of the test module dual plots can added. After setting the definitions for each axis by clicking the "Graph Settings" button at the stop of the plot, the y-axis on one of the plots can be modified so that the log scale output is shown. This is used for all of the I_D-V_{GS} curves.


Site: 1

- □ vfd_ga_dc
- □ vfd_ga_sc
- □ vfd_da_gc
- □ vfd
- □ vrd
- □ 3terminal-n-fet
  - □ switch_matrix
  - □ Vds-Id_1
  - □ VGS-ID_2
  - □ Vds-Id_sample
- □ 3terminal-p-jfet
  - □ switch_matrix_p
  - □ Vgs-Id_3
  - □ Vds-Id_2
  - □ Vds-Id_sample_p
- □ 3terminal-n-fet-2
  - □ switch_matrix_n_2
  - □ Vgs-Id_n_2
  - □ Vds-Id_n_2
  - □ ID_stress
- ☑ 3terminal-p-fet-2
  - □ switch_matrix_p_2
  - ☑ Vgs-Id_p_2
  - ☑ Vds-Id_p_2
  - □ ID_stress_p_2
- ☑ 3terminal-n-fet-3
  - □ switch_matrix_n_3
  - ☑ Vgs-Id_n_3
  - ☑ Vds-Id_n_3
  - □ Id_stress_n_3

04/10/2016 13:43:44

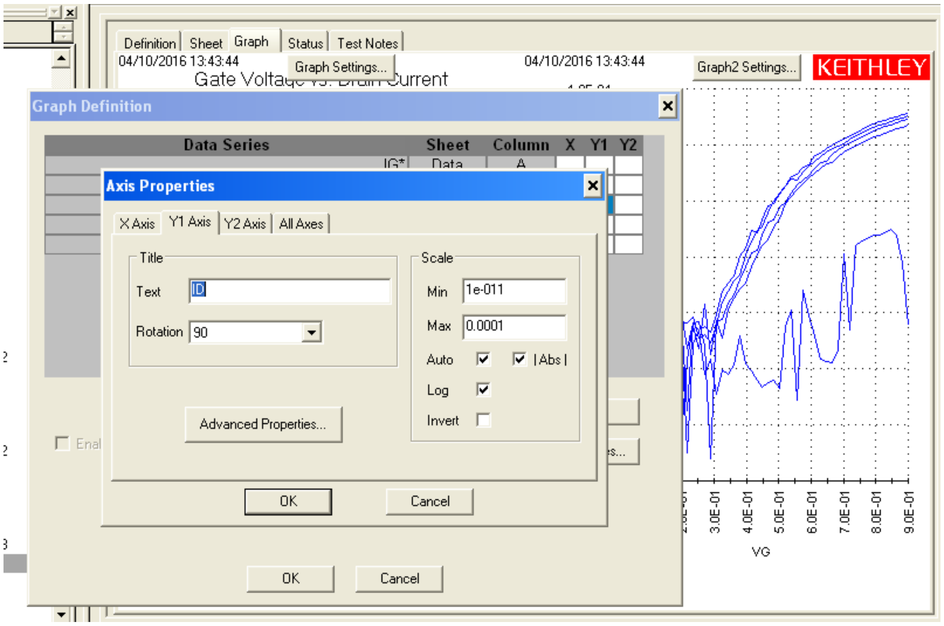
Gate Voltage vs. Drain Current

04/10/2016 13:43:44

Graph Definition

Data Series	Sheet	Column	X	Y1	Y2
IG*	Data	A...			
VG*	Data	B...	*		
ID*	Data	C...		*	
VD*	Data	D...			
IS*	Data	E...			

□ Enable Multiple X's


04/10/2016 13:43:44

Gate Voltage vs. Drain Current

04/10/2016 13:43:44

Graph Definition

Axis Properties

Y1 Axis

Title

Text: ID

Rotation: 90

Scale

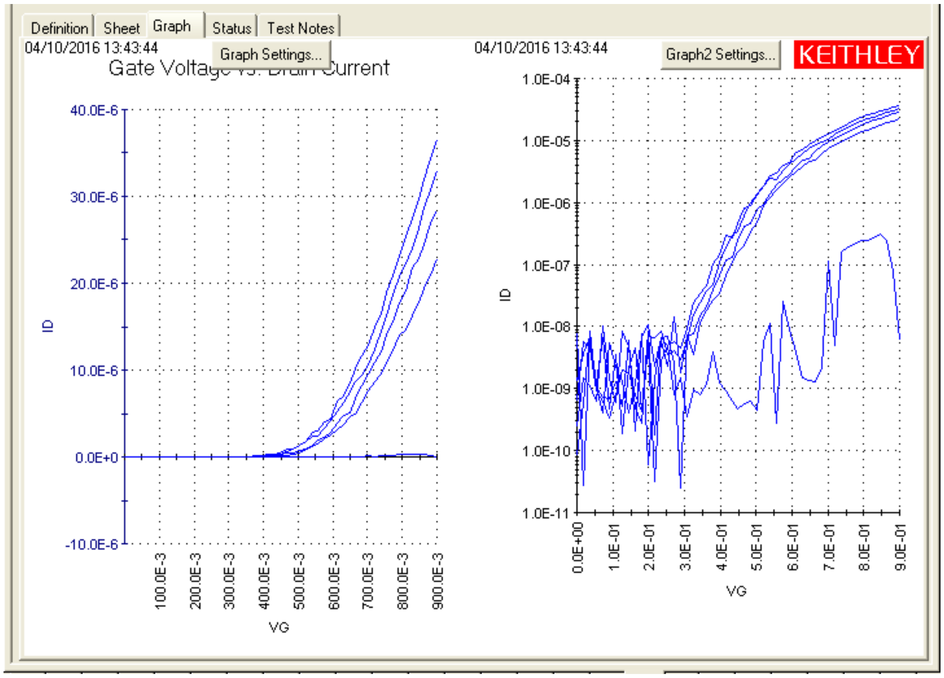
Min: 1e-011

Max: 0.0001

Auto ☑ ☑ |Abs|

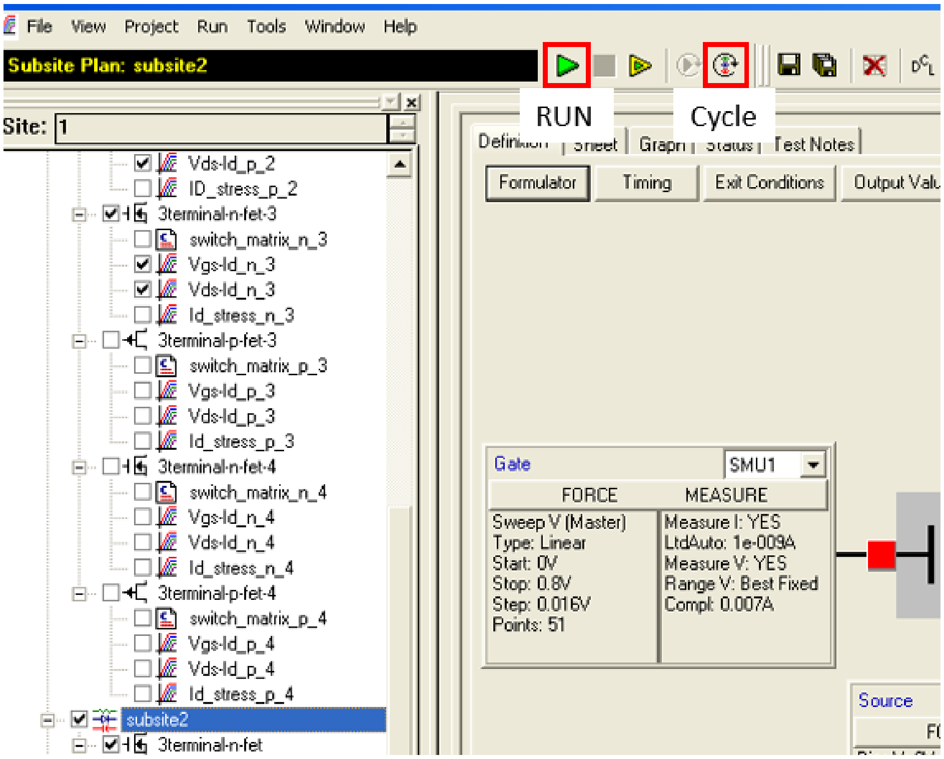
Log ☑

Invert □



3. Running Subsite

   By clicking the green arrow at the top of the selected test module will run one time. In order to perform a cycle measurement, click the with the subsite selected icon labeled cycle below and the subsite will cycle for the defined times.

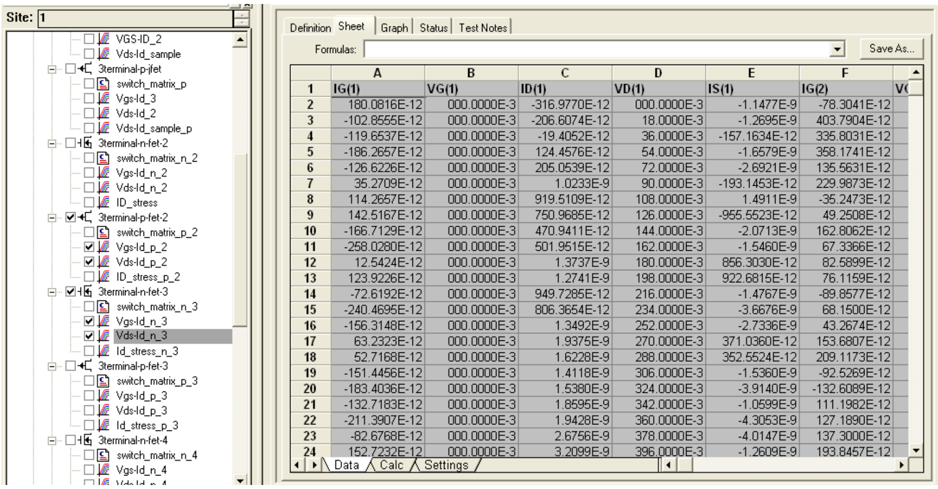


Gate SMU1

FORCE	MEASURE
Sweep V (Master) Type: Linear Start: 0V Stop: 0.8V Step: 0.016V Points: 51	Measure I: YES LtdAuto: 1e-009A Measure V: YES Range V: Best Fixed Compl: 0.007A

4. Saving Results

   Under the "Sheet" tab of the test module is where all of the data points are saved. By clicking "Save As" on the right side the data is saved in an .CSV format. This data can be transferred to another computer using a thumb drive so the data can be processed.



	A	B	C	D	E	F
1	IG(1)	VG(1)	ID(1)	VD(1)	IS(1)	IG(2)
2	180.0816E-12	000.0000E-3	-316.9770E-12	000.0000E-3	-1.1477E-9	-78.3041E-12
3	-102.8555E-12	000.0000E-3	-206.6074E-12	18.0000E-3	-1.2695E-9	403.7904E-12
4	-119.6537E-12	000.0000E-3	-19.4052E-12	36.0000E-3	-157.1634E-12	335.8031E-12
5	-186.2657E-12	000.0000E-3	124.4576E-12	54.0000E-3	-1.6579E-9	358.1741E-12
6	-126.6226E-12	000.0000E-3	205.0539E-12	72.0000E-3	-2.6921E-9	135.5631E-12
7	35.2709E-12	000.0000E-3	1.0233E-9	90.0000E-3	-193.1453E-12	229.9873E-12
8	114.2657E-12	000.0000E-3	919.5109E-12	108.0000E-3	1.4911E-9	-35.2473E-12
9	142.5167E-12	000.0000E-3	750.9685E-12	126.0000E-3	-955.5523E-12	49.2508E-12
10	-166.7129E-12	000.0000E-3	470.9411E-12	144.0000E-3	-2.0713E-9	162.8062E-12
11	-258.0280E-12	000.0000E-3	501.9515E-12	162.0000E-3	-1.5460E-9	67.3366E-12
12	12.5424E-12	000.0000E-3	1.3737E-9	180.0000E-3	856.3030E-12	82.5899E-12
13	123.9226E-12	000.0000E-3	1.2741E-9	198.0000E-3	922.6815E-12	76.1159E-12
14	-72.6192E-12	000.0000E-3	949.7285E-12	216.0000E-3	-1.4767E-9	-89.8577E-12
15	-240.4695E-12	000.0000E-3	806.3654E-12	234.0000E-3	-3.6676E-9	68.1500E-12
16	-156.3148E-12	000.0000E-3	1.3492E-9	252.0000E-3	-2.7336E-9	43.2674E-12
17	63.2323E-12	000.0000E-3	1.9375E-9	270.0000E-3	371.0360E-12	153.6807E-12
18	52.7168E-12	000.0000E-3	1.6228E-9	288.0000E-3	352.5524E-12	209.1173E-12
19	-151.4456E-12	000.0000E-3	1.4118E-9	306.0000E-3	-1.5360E-9	-92.5269E-12
20	-183.4036E-12	000.0000E-3	1.5380E-9	324.0000E-3	-3.9140E-9	-132.6089E-12
21	-132.7183E-12	000.0000E-3	1.8595E-9	342.0000E-3	-1.0599E-9	111.1982E-12
22	-211.3907E-12	000.0000E-3	1.9428E-9	360.0000E-3	-4.3053E-9	127.1890E-12
23	-82.6768E-12	000.0000E-3	2.6756E-9	378.0000E-3	-4.0147E-9	137.3000E-12
24	152.7232E-12	000.0000E-3	3.2099E-9	396.0000E-3	-1.2609E-9	193.8457E-12

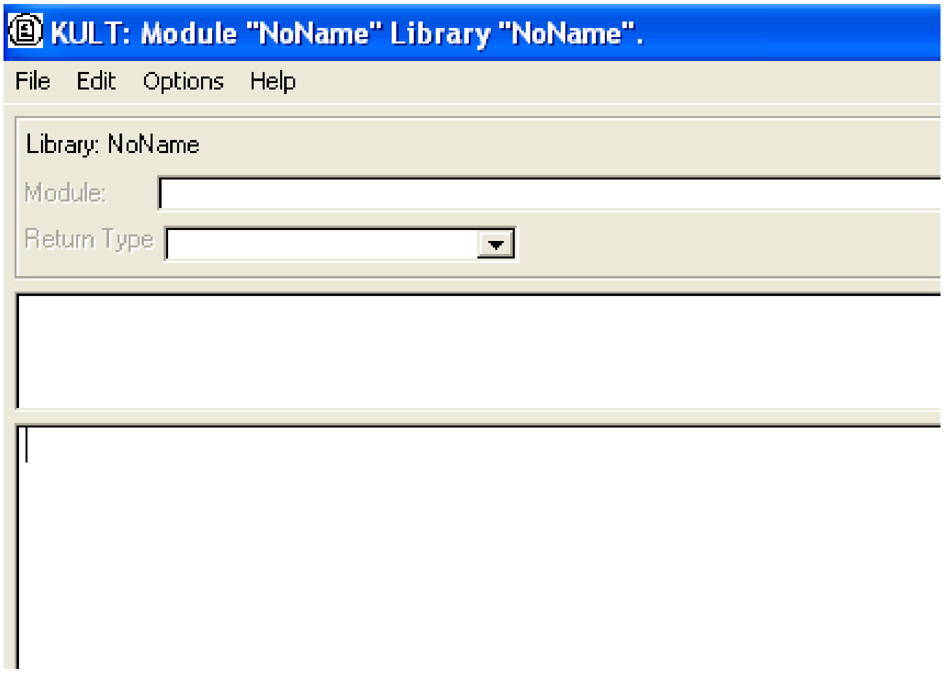
B.4 External Instrument Code

The script below is used to control the B2200 for connecting and disconnecting the devices to the 4200. The syntax for specifying the connection is *XYYZZ*. Where *X* is the card number, *YY* is the input port, and *ZZ* is the output port. By simply adding sequential connections allows multiple connections to be made at once.

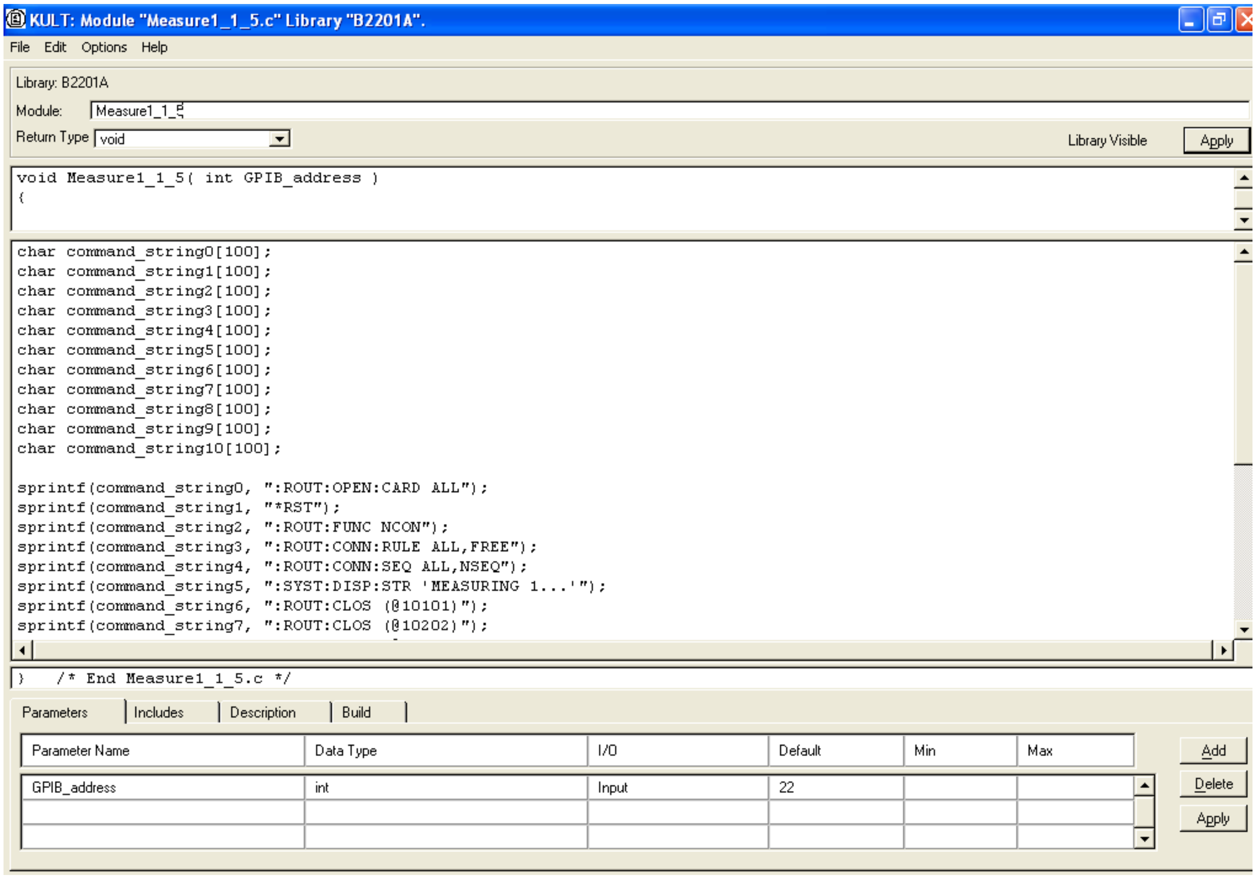
```
char command_string0[100];
char command_string1[100];
char command_string2[100];
char command_string3[100];
char command_string4[100];
char command_string5[100];
char command_string6[100];
char command_string7[100];
char command_string8[100];
char command_string9[100];
char command_string10[100];
/*Open all*/
sprintf(command_string0, ":ROUT:OPEN:CARD_ALL");
/*B2200 Setup8?
sprintf(command_string1, "*RST");
sprintf(command_string2, ":ROUT:FUNC NCON");
sprintf(command_string3, ":ROUT:CONN:RULE ALL,FREE");
sprintf(command_string4, ":ROUT:CONN:SEQ ALL,NSEQ");
sprintf(command_string5, ":SYST:DISP:STR 'MEASURING 1...'");
/*Close the following Connections*/
sprintf(command_string6, ":ROUT:CLOS␣(@10101)");
sprintf(command_string7, ":ROUT:CLOS␣(@10202)");
sprintf(command_string8, ":ROUT:CLOS␣(@10303)");
sprintf(command_string9, ":ROUT:CLOS␣(@10404)");
sprintf(command_string10, ":ROUT:CLOS␣(@10505)");

kibsnd(22, -1, GPIBTIMO, strlen(command_string0), command_string0);
kibsnd(22, -1, GPIBTIMO, strlen(command_string1), command_string1);
kibsnd(22, -1, GPIBTIMO, strlen(command_string2), command_string2);
kibsnd(22, -1, GPIBTIMO, strlen(command_string3), command_string3);
kibsnd(22, -1, GPIBTIMO, strlen(command_string4), command_string4);
kibsnd(22, -1, GPIBTIMO, strlen(command_string5), command_string5);
kibsnd(22, -1, GPIBTIMO, strlen(command_string6), command_string6);
kibsnd(22, -1, GPIBTIMO, strlen(command_string7), command_string7);
kibsnd(22, -1, GPIBTIMO, strlen(command_string8), command_string8);
kibsnd(22, -1, GPIBTIMO, strlen(command_string9), command_string9);
kibsnd(22, -1, GPIBTIMO, strlen(command_string10), command_string10);
delay(3000);
```

This scripted is added to the tests using the following steps:

1. Create new library and module under the ”File” tab



2. Add Script



Library: B2201A

Module: Measure1_1_5

Return Type void

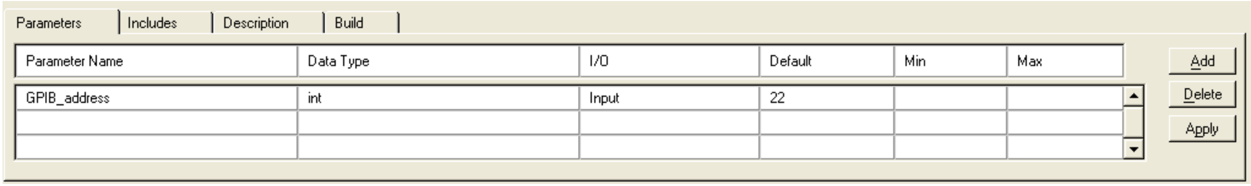
```
void Measure1_1_5( int GPIB_address )
{
```

```
char command_string0[100];
char command_string1[100];
char command_string2[100];
char command_string3[100];
char command_string4[100];
char command_string5[100];
char command_string6[100];
char command_string7[100];
char command_string8[100];
char command_string9[100];
char command_string10[100];

sprintf(command_string0, ":ROUT:OPEN:CARD ALL");
sprintf(command_string1, "*RST");
sprintf(command_string2, ":ROUT:FUNC NCON");
sprintf(command_string3, ":ROUT:CONN:RULE ALL,FREE");
sprintf(command_string4, ":ROUT:CONN:SEQ ALL,NSEQ");
sprintf(command_string5, ":SYST:DISP:STR 'MEASURING 1...'");
sprintf(command_string6, ":ROUT:CLOS (@10101)");
sprintf(command_string7, ":ROUT:CLOS (@10202)");
```

```
}   /* End Measure1_1_5.c */
```

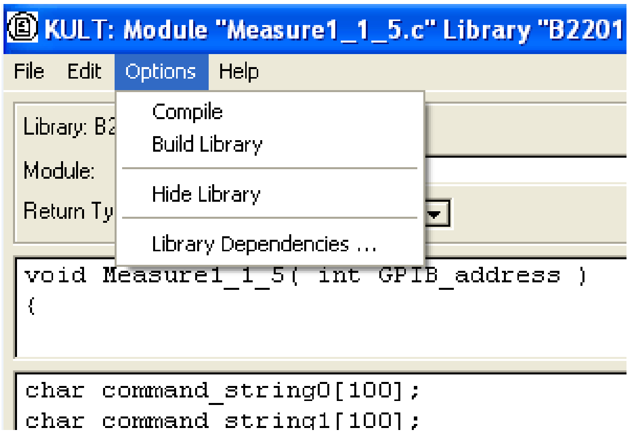
Parameter Name	Data Type	I/O	Default	Min	Max
GPIB_address	int	Input	22		

3. Adding I/O



Parameter Name	Data Type	I/O	Default	Min	Max
GPIB_address	int	Input	22		

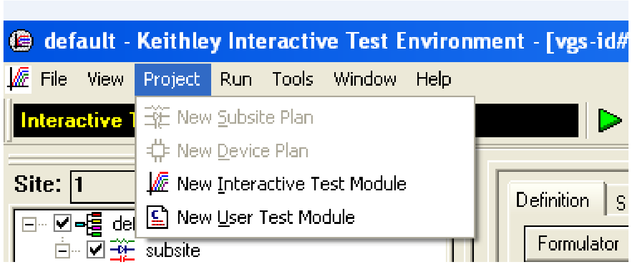
4. Compile and build library



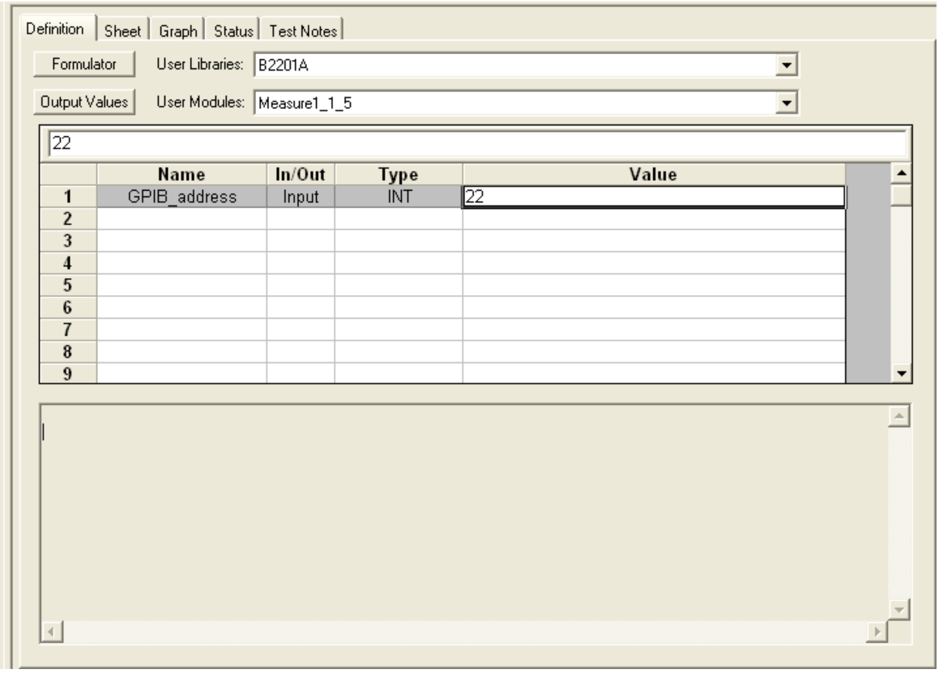
```
void Measure1_1_5( int GPIB_address )
{
```

```
char command_string0[100];
char command_string1[100];
```

5. Add a New User Test Module to Subsite



6. Select Library, Modules, and Inputs



User Libraries: B2201A

User Modules: Measure1_1_5

22

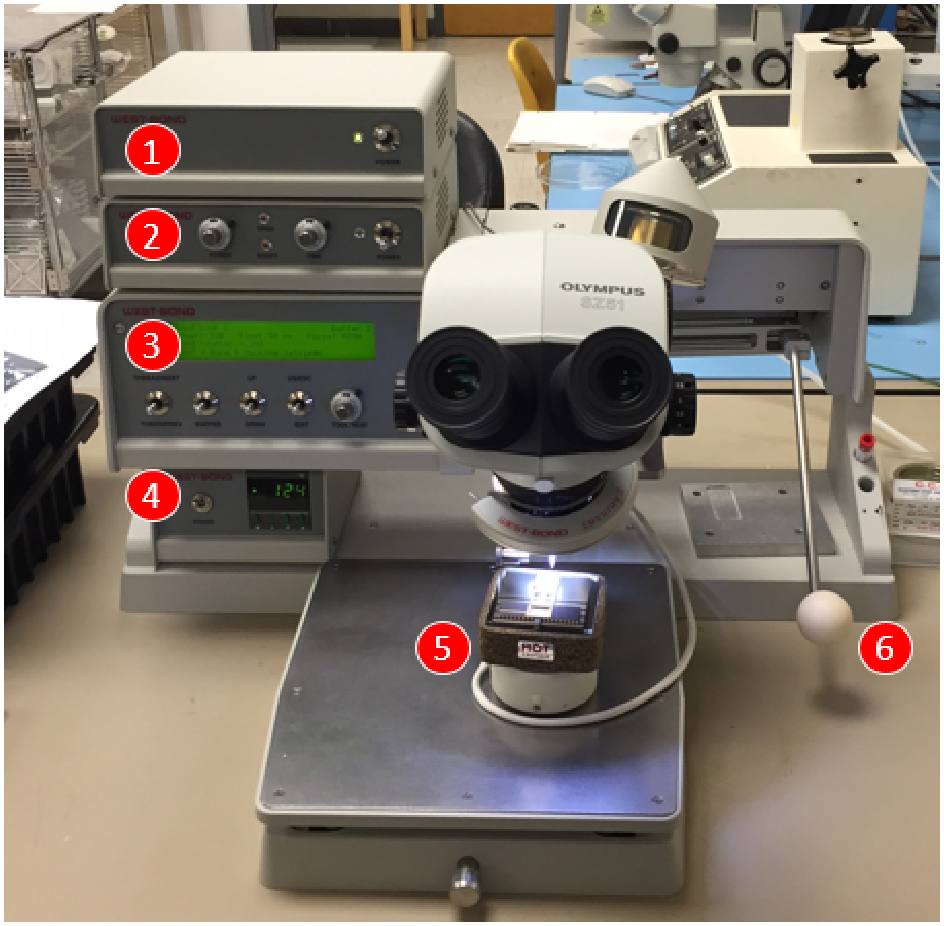
	Name	In/Out	Type	Value
1	GPIB_address	Input	INT	22
2				
3				
4				
5				
6				
7				
8				
9				

Appendix C
Wedge Wire Bonding

The wire bonder used is a West Bond Model 7KE. This model is capable of doing wedge and ball bonds. All of the bonds made for this study were 45 degree wedge bonds.

C.1 Components

1. Power supply
2. Force settings (For ball bonding option)
3. Control panel
4. Heated workholder temperature controller
5. Workholder
6. Manual operation arm

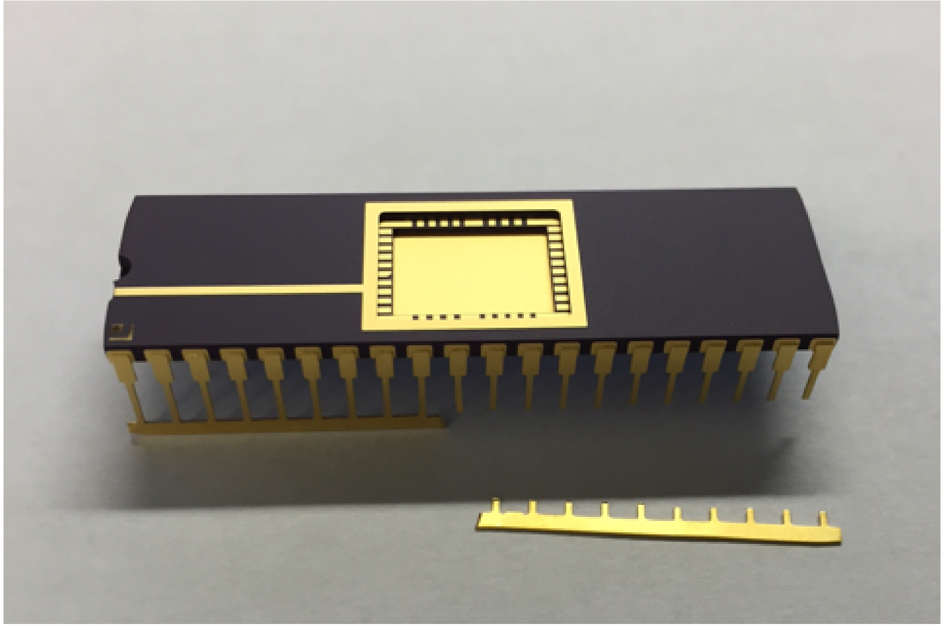


C.2 Steps

1. Sheer DIP package leads
2. Mount a chip to the package
3. Bond

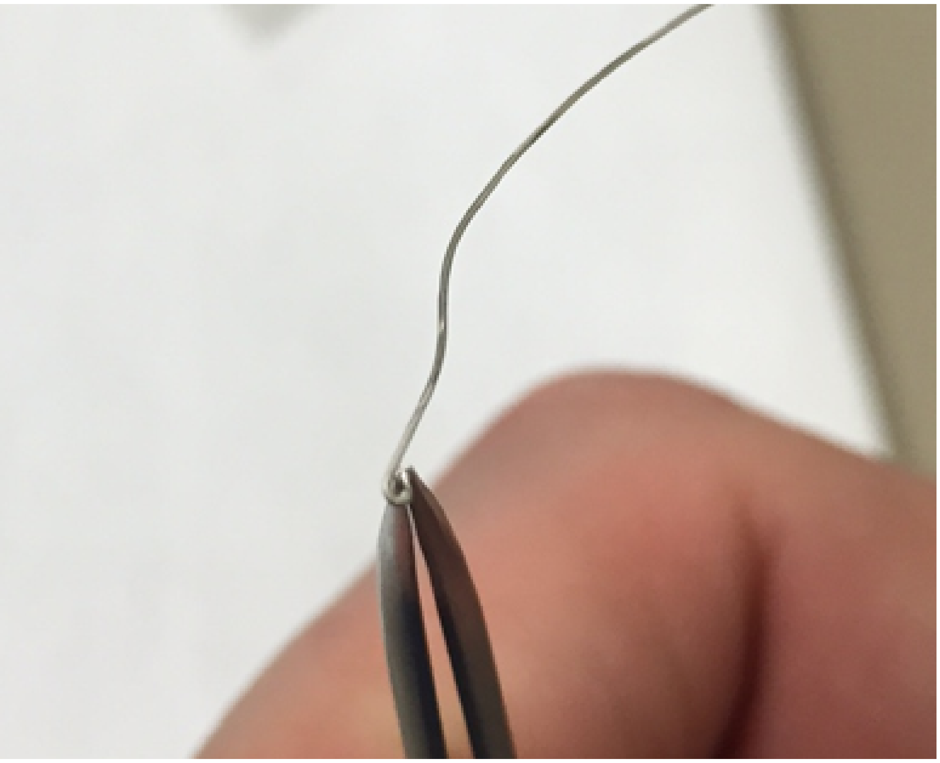
C.2.1 Sheering DIP package leads

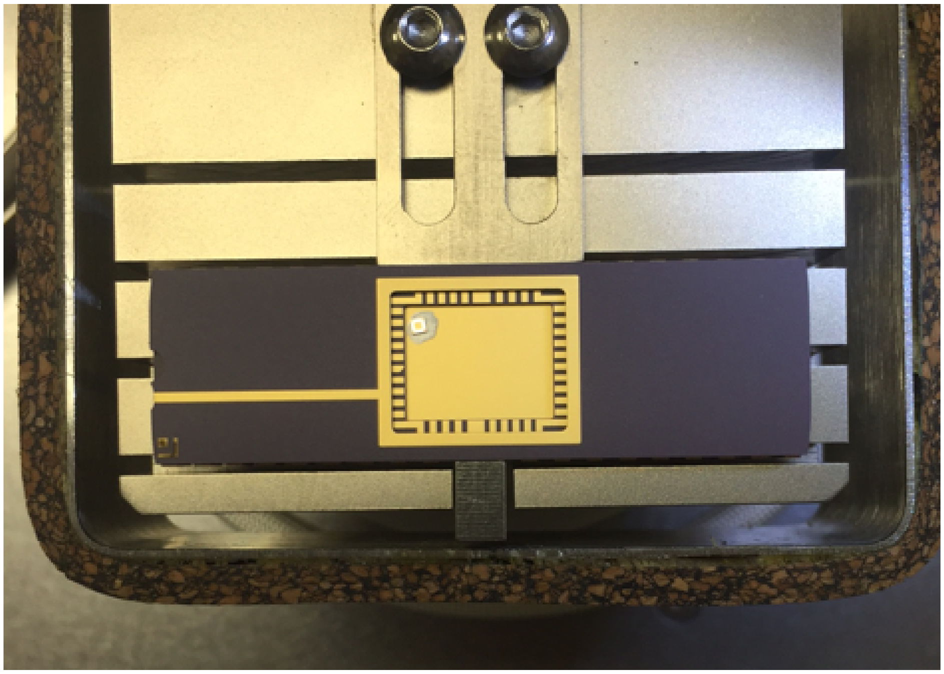
Using an angle cutter, cut off the bottom on the package leads that tie all of the lead together. Make sure all of the leads are approximately the same length.



C.2.2 Mounting a chip to the package

Place the DIP in the workholder with the heat off. Using a small gauge wire, create a small loop using tweezers to dip into the silver paste. Apply small amount of silver paint to desired location on DIP package. Using tweezers, place chip on silver paste and allow to dry for 5-10 minutes.

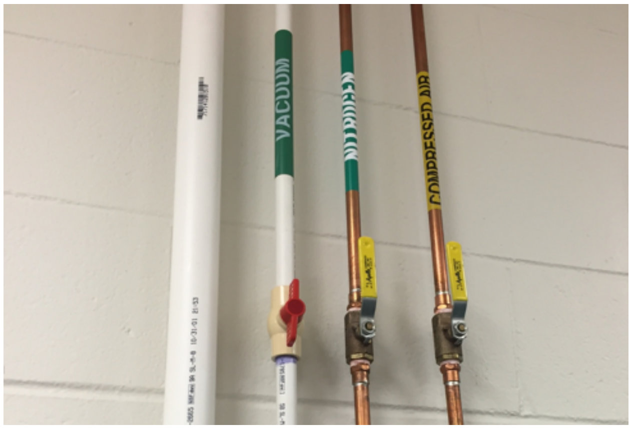




C.2.3 Bonding

1. Start up

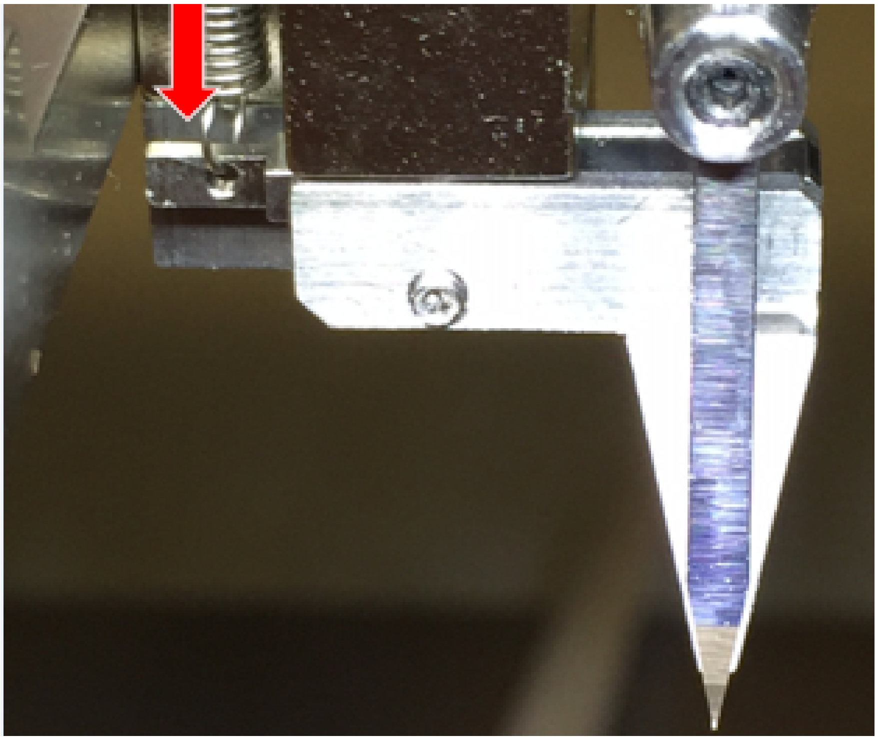
   Turn on the power supply switch, the air supply, and the heated workholder temperature controller. The temperature of the work holder should be set to 125° C. The compressed air value is on the wall located to the right of the bonder as shown below.

2. Completing Bonds

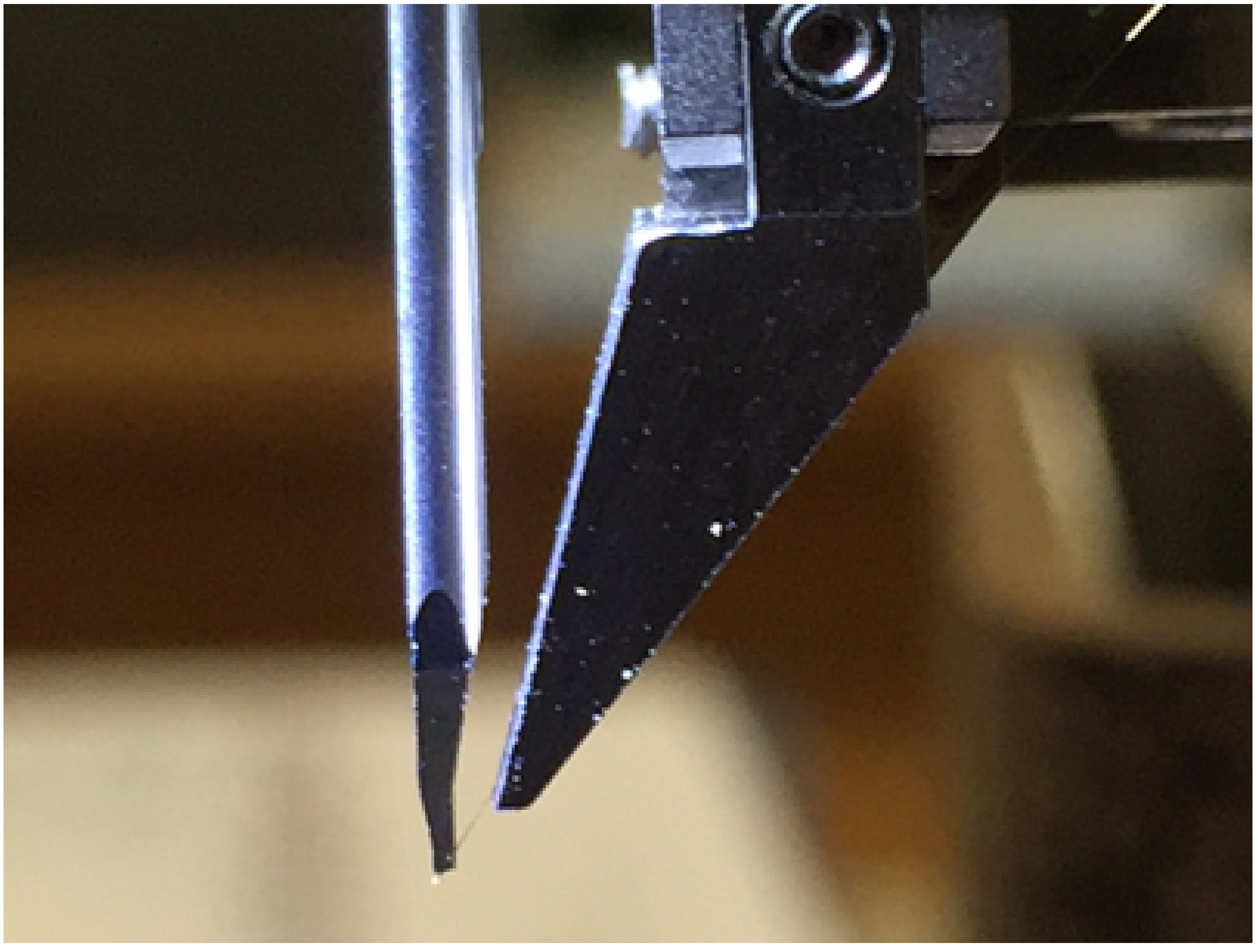
   Make all of the bonds from the bonding diagram. Always start with the first bond on the DIP package to ensure that any excess wire will not be on the smaller chip pad. If the wire come out of the tool head, the head is re-threaded using the following steps:

   (a) Open clamps by using tweezers to extend the spring releasing the wire

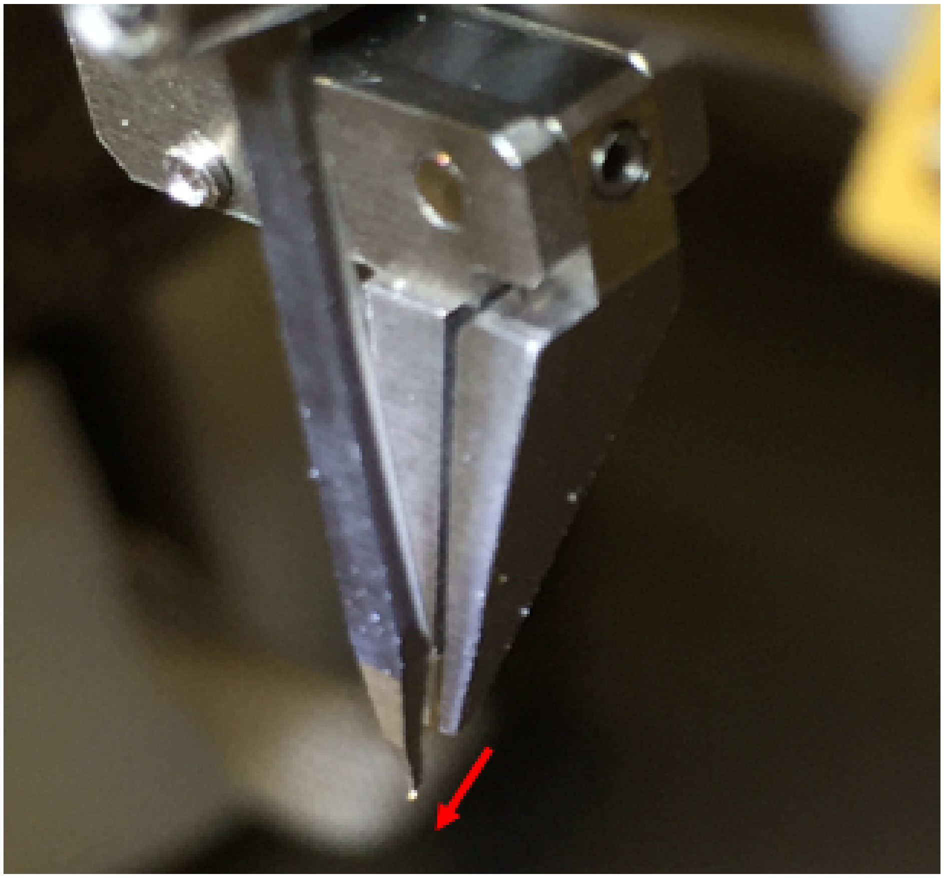
   

(b) Pull off the end to make sure the end is straight

(c) Re-thread wire at approximately a 45° angle through the end of the tool head using tweezers.



(d) Open clamps while pulling wire through tool head so that the wire is in between the clamps.

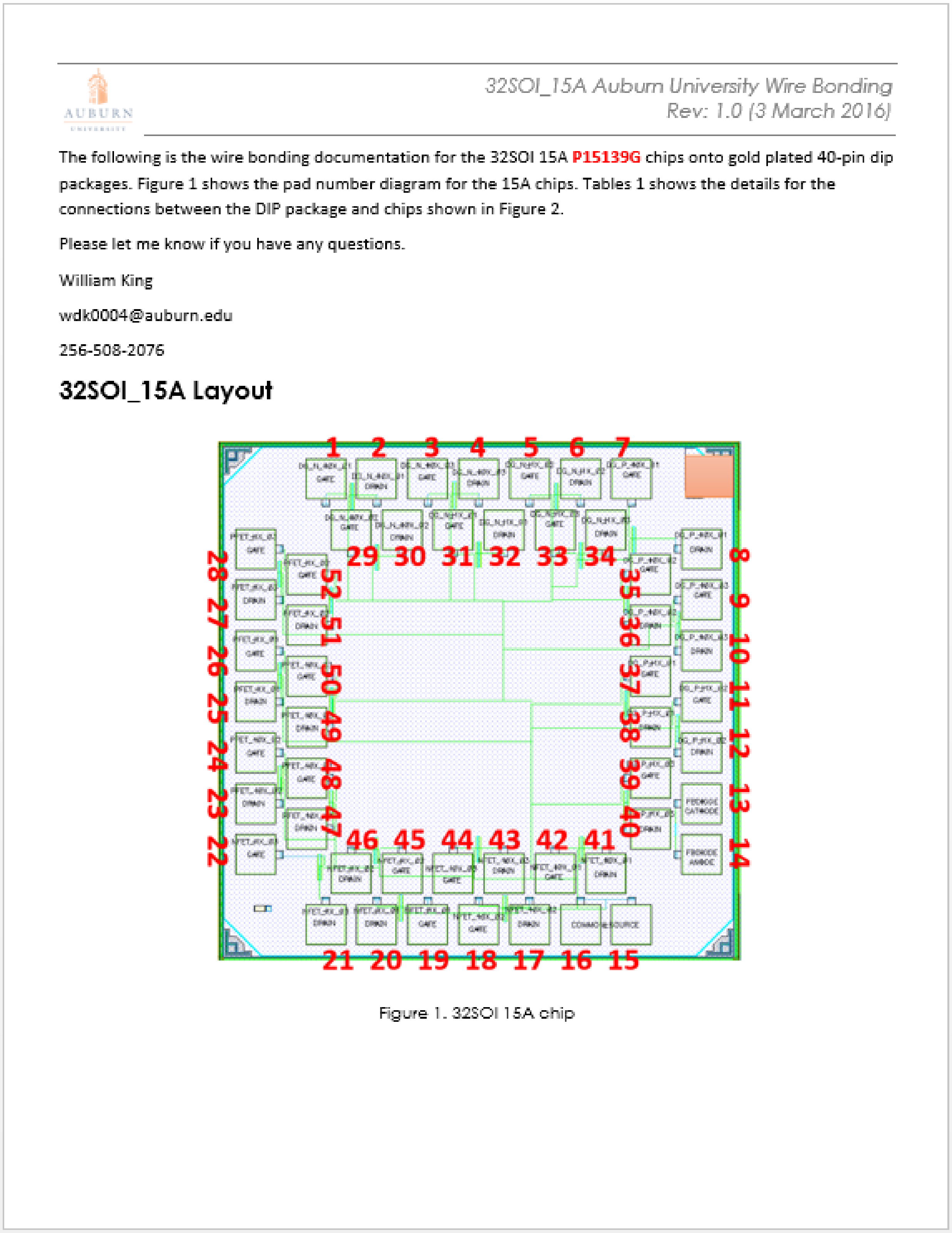


3. Shut down

   Remove the DIP package from the workholder and place it on the base of the bonder to cool. Turn off the power supply switch, air supply, and heated workholder temperature controller

C.3 Example bonding diagrams

This is an example of a wire bonding diagram made for the 32nm 1x1mm reliability chip.



32SOI_15A Auburn University Wire Bonding
Rev: 1.0 (3 March 2016)

The following is the wire bonding documentation for the 32SOI 15A P15139G chips onto gold plated 40-pin dip packages. Figure 1 shows the pad number diagram for the 15A chips. Tables 1 shows the details for the connections between the DIP package and chips shown in Figure 2.

Please let me know if you have any questions.

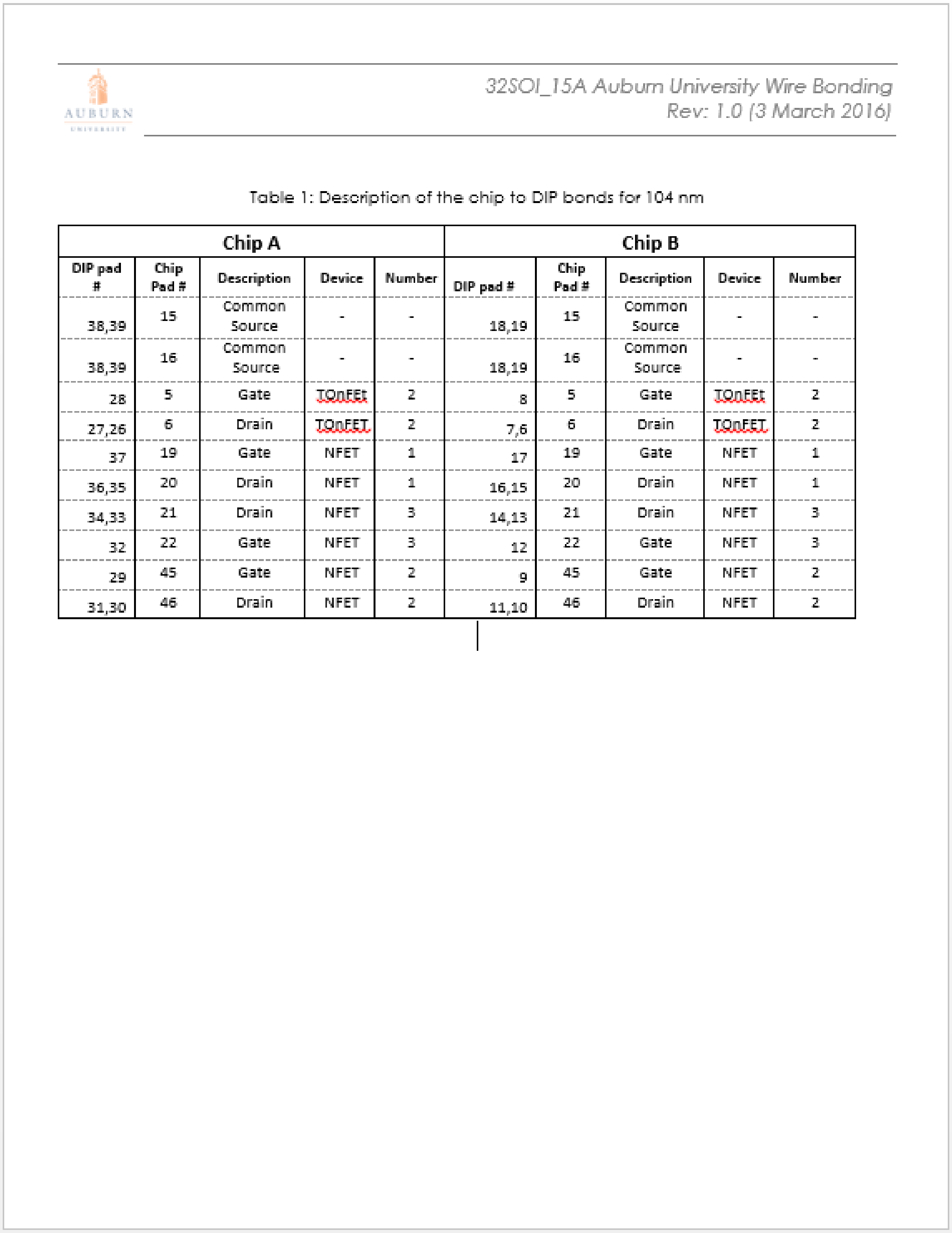
William King

wdk0004@auburn.edu

256-508-2076

32SOI_15A Layout

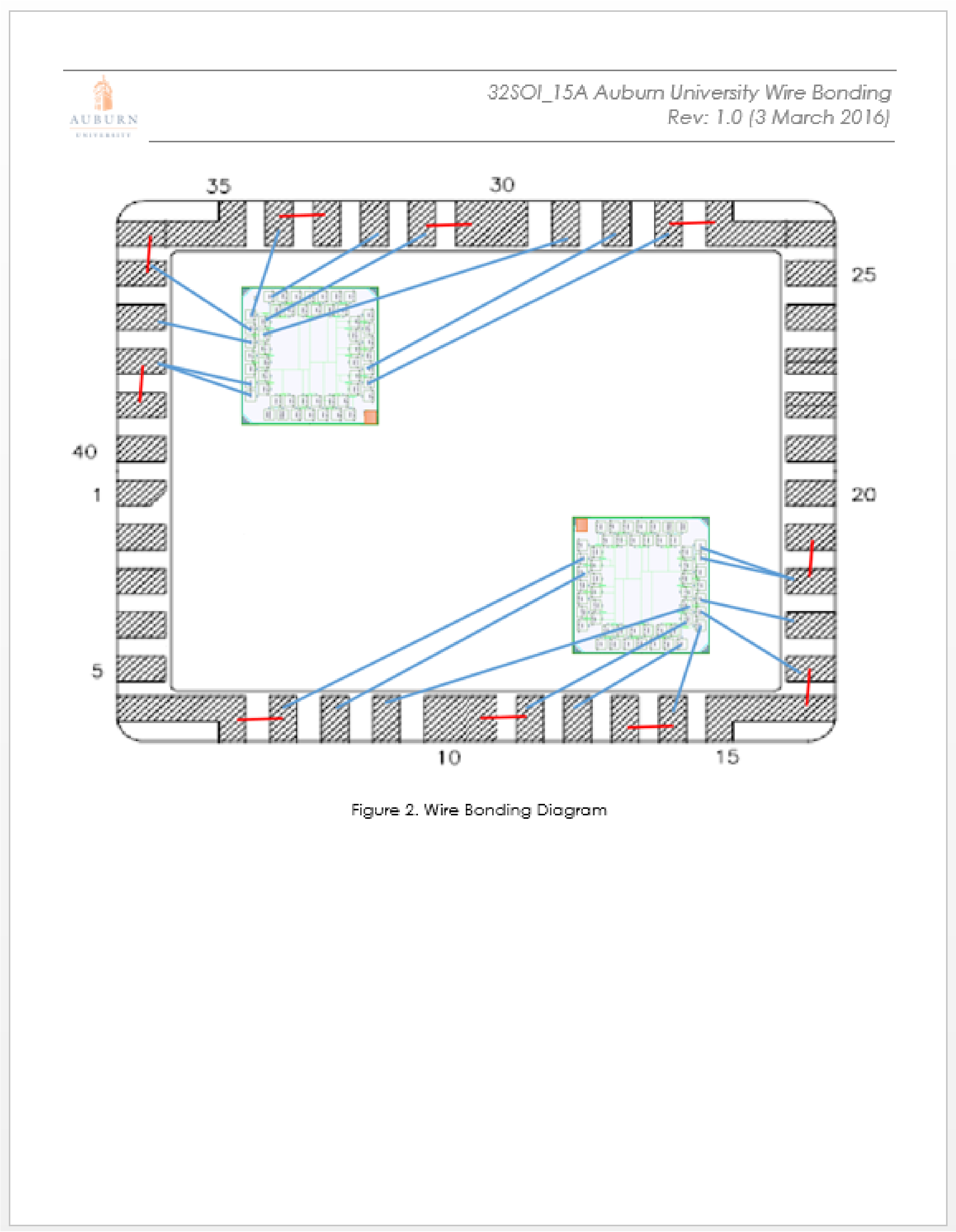
Figure 1. 32SOI 15A chip



32SOI_15A Auburn University Wire Bonding
Rev: 1.0 (3 March 2016)

Table 1: Description of the chip to DIP bonds for 104 nm

Chip A					Chip B				
DIP pad #	Chip Pad #	Description	Device	Number	DIP pad #	Chip Pad #	Description	Device	Number
38,39	15	Common Source	-	-	18,19	15	Common Source	-	-
38,39	16	Common Source	-	-	18,19	16	Common Source	-	-
28	5	Gate	TOnFET	2	8	5	Gate	TOnFET	2
27,26	6	Drain	TOnFET	2	7,6	6	Drain	TOnFET	2
37	19	Gate	NFET	1	17	19	Gate	NFET	1
36,35	20	Drain	NFET	1	16,15	20	Drain	NFET	1
34,33	21	Drain	NFET	3	14,13	21	Drain	NFET	3
32	22	Gate	NFET	3	12	22	Gate	NFET	3
29	45	Gate	NFET	2	9	45	Gate	NFET	2
31,30	46	Drain	NFET	2	11,10	46	Drain	NFET	2



32SOI_15A Auburn University Wire Bonding
Rev: 1.0 (3 March 2016)

Figure 2. Wire Bonding Diagram

Appendix D
MATLAB Code

D.1 Batch Read Scripts

D.1.1 Standard/Post

```
close all
global fig
%File directory where the Excel files are
foldersInit = {'Insert␣File␣Location'};
DATA = cell(numel(foldersInit),1);
k = 1;
%Save file names
for fo = 1:numel(foldersInit)
    files = dir(foldersInit{fo}); % get array of all files the folder
    files([files(:).isdir]) = [];
    DATA{fo} = cell(numel(files),1);
    for fi = 1:numel(files)
        filename = fullfile(foldersInit{fo},files(fi).name);
        file(k) = cellstr(filename);
        g = num2str(k);
        display(['[',g,']', files(fi).name])
        k = k+1;
        [status,sheets] = xlsfinfo(filename);
        if ~isempty(status)

            lastsheet = sheets{end};
            [num,txt,raw] = xlsread(filename,lastsheet);
            DATA{fo}{fi} = num; % # or txt, or raw

        end
    end
end
%Display number of files to check the read-in process
number_files = fi
x = 'A':'Z';
yy = ['2' , '80'];
y = num2str(yy);
for z = 1:number_files
fig(z).title = strrep(files(z).name,'_','-');
n = 2;
V = xlsread(char(file(z)),'Data',[[x(n),y(1)],...
':',[x(n),[y(2),y(3)]]]);
%Determine whether the file is ID-VGS or ID-VDS and set  variable
%columns
if V(2) == V(3)
m = 3; %start with column C for Id
```

```
n = 4; %start with column D for Vds
h= 1; %start with column C for Ig
g= 5; %start with column C for Is
 fig(z).xx = 'VDS␣(V)';
else
     m= 3; %start with column C for Id
     n= 2; %start with column B for Vgs
     h= 1; %start with column C for Ig
     g= 5; %start with column C for Is
 fig(z).xx = 'VGS(V)';
end
%Cycle through all of the data saving in a structured array "fig"
for k = 1:5
I = xlsread(char(file(z)),'Data',[[x(m),y(1)],...
':',[x(m),[y(2),y(3)]]]);
IG = xlsread(char(file(z)),'Data',[[x(h),y(1)],...
':',[x(h),[y(2),y(3)]]]);
IS = xlsread(char(file(z)),'Data',[[x(g),y(1)],...
':',[x(g),[y(2),y(3)]]]);
V = xlsread(char(file(z)),'Data',[[x(n),y(1)],...
':',[x(n),[y(2),y(3)]]]);
   m = m+5;
   n = n+5;
   h = h+5;
   g = g+5;
% z is the device and k is the step in the measurement
fig(z).data(k).v = V;
fig(z).data(k).i = I;
fig(z).data(k).ig = IG;
fig(z).data(k).is = IS;
end
end
```

D.1.2 Stress

```
%close all
global stress
%File directory where the excel files are
folderStress = {'Input␣File␣Location'};
DATA = cell(numel(folderStress),1);
k = 1;
%Save file names
for fo = 1:numel(folderStress)
    files = dir(folderStress{fo}); %get array of all files the folder
    files([files(:).isdir]) = [];
    DATA{fo} = cell(numel(files),1);
    for fi = 1:numel(files)
        filename = fullfile(folderStress{fo},files(fi).name);
        file(k) = cellstr(filename);
        g = num2str(k);
        display(['[',g,']', files(fi).name])
        k = k+1;
        [status,sheets] = xlsfinfo(filename);
        finalsheet(k-1) = length(sheets);
        if ~isempty(status)
            lastsheet = sheets{end};
            [num,txt,raw] = xlsread(filename,lastsheet);
            DATA{fo}{fi} = num; % # or txt, or raw
        end
    end
end
%Display number of files to check read-in process
number_files = fi
x = 'A':'Z';
yy = ['2' , '80'];
y = num2str(yy);
%Cycle through files
for z = 1:number_files
    stress(z).title = strrep(files(z).name,'␣','-');
    V = xlsread(char(file(z)),'Data',[[x(2),y(1)],...
    ':',[x(2),[y(2),y(3)]]]);
    I = xlsread(char(file(z)),'Data',[[x(3),y(1)],...
    ':',[x(3),[y(2),y(3)]]]);
    stress(z).cycle(1).v = V;
    stress(z).cycle(1).i = I;
    name = 'Cycle';
    %Read all of the sheets in the stress Excel file
    for k = 2:(finalsheet(z)-2)
        N = num2str(k);
        input = [name,N];
        I = xlsread(char(file(z)),input,[[x(3),y(1)],...
        ':',[x(3),[y(2),y(3)]]]);
        %z is the device and k is the stress cycle
        stress(z).cycle(k).v = V;
        stress(z).cycle(k).i = I;
    end
end
```

D.1.3 Simulations

```
%File directory where the .csv files are
foldersInit = {'H:\Simulations\cryo_model_results\voltagesweep'};
DATA = cell(numel(foldersInit),1);
k = 1;
%Read all file names
for fo = 1:numel(foldersInit)
    files = dir(foldersInit{fo}); % get array files in the folder
    files([files(:).isdir]) = [];
    DATA{fo} = cell(numel(files),1);
    for fi = 1:numel(files)
        filename = fullfile(foldersInit{fo},files(fi).name);
        file(k) = cellstr(filename);
        g = num2str(k);
        display(['[',g,']', files(fi).name])
        k = k+1;
        [status,sheets] = xlsfinfo(filename);
        if ~isempty(status)
            lastsheet = sheets{end};
            [num,txt,raw] = xlsread(filename,lastsheet);
            DATA{fo}{fi} = num; %# or txt, or raw
        end
    end
end
%Display number of files to check the read-in process
number_files = fi
x = 'A':'Z';
yy = ['2' , '80'];
y = num2str(yy);
%Read in all of the data in the csv file
for z = 1:number_files
    [num,txt,raw] = xlsread(char(file(z)));
%number of steps
    for k = 1:5;
        for g = 1:length(num)-1
            if strcmp((files(z).name(2)),'d')
%z is the device, k is the step, and g is the data point
                sim(z).vds(k).v(g) = num(g,1);
                sim(z).vds(k).i(g) = num(g,1+k);
            elseif strcmp((files(z).name(2)),'g')
                sim(z).vgs(k).v(g) = num(g,1);
                sim(z).vgs(k).i(g) = num(g,1+k);
            end
        end
    end
end
```

D.2 Standard/Post Plots

```
global fig
font = 20;
color = {'k','m','b','g','r'};
figure(1)
for x = 1:length(fig)
    if strcmp(fig(x).xx,'VDS␣(V)')
        for k = 2:5
            subplot(1,3,1)
            ax1 = plot(fig(x).data(k).v,fig(x).data(k).i,...
            '*','color',color{k});
            hold all
            %formatting
            xlabel(fig(x).xx,'FontSize',(font))
            ylabel('I_D␣(A)','FontSize',(font))
            a = get(gca,'XTickLabel');
            set(gca,'XTickLabel',a,'fontsize',(font −3),...
            'XTickLabelMode','auto')
            box on
        end
    else
        %Plot ID–VGS
        for k = 2:5
            %Linear
            subplot(1,3,2)
            plot((fig(x).data(k).v),(fig(x).data(k).i),...
            '*','color',color{k});
            hold all
            %formatting
            ylabel('I_D␣(A)','FontSize',(font))
            a = get(gca,'XTickLabel');
            set(gca,'XTickLabel',a,'fontsize',(font −3),...
            'XTickLabelMode','auto')
            box on
            xlabel(fig(x).xx,'FontSize',(font))
            %Log Scale
            subplot(1,3,3)
            semilogy((fig(x).data(k).v),abs(fig(x).data(k).i),...
            '*','color',color{k});
            hold all
            %formatting
            box on
            ylabel('I_D␣(A)','FontSize',(font))
            a = get(gca,'XTickLabel');
            set(gca,'XTickLabel',a,'fontsize',(font −3),...
            'XTickLabelMode','auto')
            xlabel(fig(x).xx,'FontSize',(font))
        end
    end
end
```

D.3 MTTF Plots

```
global stress
close all
g = 1;
font = 20;
%Calculate VT for all devices
for z = 1:length(stress)
    fig =figure(z);
    fig.Units = 'inches';
    fig.Position = [.5*z, .5, 5.5, 4.75];
    N = length(stress(z).cycle) ;
    for n = 1:N;
        %calculate gm
        N = length(stress(z).cycle) ;
        %3 point derivative calc
        for k = 2:length(stress(1).cycle(1).i)-1
            f(n).run(k-1) =
            (-stress(z).cycle(n).i(k+1)+4*stress(z).cycle(n).i(k)...
            -3*stress(z).cycle(n).i(k-1))/...
            (stress(z).cycle(n).v(k+1)-stress(z).cycle(n).v(k-1));
        end
        %Fit line of gm to calculate gm max
        p = polyfit(stress(z).cycle(n).v(2:end-1),transpose(f(n).run),10);
        fit(n).gm = polyval(p,stress(z).cycle(n).v(2:end-1));
        V = stress(z).cycle(n).v;
        %find where max gm occurs and get vg
        xIndex(n) = find(fit(n).gm == max(fit(n).gm), 1, 'first')+1;
        maxVg(n) = V(xIndex(n));
        %fit line slope at Vgs of gm max
        front =
        (stress(z).cycle(n).i(xIndex(n)+1)-stress(z).cycle(n).i(xIndex(n)))
           /...
            (stress(z).cycle(n).v(xIndex(n)+1)-stress(z).cycle(n).v(xIndex(n))
               );
        back =
        (stress(z).cycle(n).i(xIndex(n))-stress(z).cycle(n).i(xIndex(n)-1))
           /...
            (stress(z).cycle(n).v(xIndex(n)+1)-stress(z).cycle(n).v(xIndex(n))
               );
        slope(n) = (front+back)/2;
        %constant mobility calculation of threshold
        yth(n).vt = slope(1).*(stress(z).cycle(n).v-maxVg(n))...
        +stress(z).cycle(n).i(xIndex(n));
        %find where line hits mag
        mag = 0;
        Vt(z).cycle(n) = interp1(yth(n).vt,V,mag);
    end
    %plot results
    plot(V,stress(z).cycle(1).i,'*','color',[0 .6 0])
    box on
    hold all
    plot(V,yth(1).vt,'Linewidth',1.25,'color',[0 .6 0])
    hold all
    plot(V,stress(z).cycle(N).i,'*','color','b')
    plot(V,yth(N).vt,'Linewidth',1.25,'color','b')
    plot(Vt(z).cycle(1),mag,'o','Linewidth',1.25,'color',[0 .6 0])
    plot(Vt(z).cycle(N),mag,'o','Linewidth',1.25,'color','b')
    %format plots
    a = get(gca,'XTickLabel');
    set(gca,'XTickLabel',a,'fontsize',(font-3),'XTickLabelMode','auto')
    box on
    xlabel('V_{GS}(V)','FontSize',font)
    ylabel('I_D(A)','FontSize',font)
    title([['Voltage␣Threshold␣at␣I_D␣=␣',num2str(mag)],'A'],'FontSize',font
        -4)
    h = legend('Initial','Linear␣Fit␣Initial', 'Post␣Stressing',...
        'Linear␣Fit␣Post',['V_{T}␣Init=',num2str(Vt(z).cycle(1))],...
        ['V_{T}␣Post=',num2str(Vt(z).cycle(N))],'Location','Best');
    set(h,'FontSize',(font-6));
    tightfig();
    set(gcf,'color','w');
    %input times of stresses
    timeL = [0,10,30,100,300,1000,3600,10000,36000,72000];
    timeS = [0,1,3,6,10,15,30,60,90,120,240,360,720,1080];
    timeXL = [0,10,40,140,440,1440,5040,15040,51040,123040];
    timeM = [0,10,30,100,300,1000,1800,3600];
    %normalized vth vs time calc
    for n = 1: length(Vt(z).cycle)
        vt_change(z).shift(n) =abs(Vt(z).cycle(n)-Vt(z).cycle(1));
    end
end
%change in vth plot vs time
fig =figure(z+1);
fig.Units = 'inches';
fig.Position = [.5*6, .5, 6, 5];
vt_goal= 0.07;
g = 1;
q = 1;
%set color
matchcolor = { 'b';'g';'g';'b';[0 .6 0];[0 .6 0]};
for z= 1: length(vt_change)
    %Use to omit outliners at the beginning and end
    start = [1 1 1 1 1 1];
    stop =
    [length(vt_change(1).shift) length(vt_change(2).shift)...
    length(vt_change(3).shift) length(vt_change(4).shift)...
    length(vt_change(5).shift) length(vt_change(6).shift)-2];
    %determine stress voltage
    str  = stress(z).title(24:26);
    numbers(z) = strread(lower(str), '%f', 'whitespace',['a':'z' '␣\t']);
    switch numbers(z)
        case 20
            %set correct time
            time = timeM;
            %plot data points
            p20(1).vt = loglog(time((start(z):stop(z))),...
            vt_change(z).shift(start(z):stop(z)),'*','color',matchcolor{z});
            hold all
            %fit data points
            pvt = polyfit(log(time((start(z)+1:stop(z)))),...
            log(vt_change(z).shift((start(z)+1:stop(z)))),1);
            fit(z).vt = exp(pvt(1)*log(time((start(z):stop(z))))+pvt(2));
            p20(2).vt = loglog(time((start(z):stop(z))),fit(z).vt,'color',...
            matchcolor{z},'Linewidth',1.25);
            %interpolate failure time
            fail(z).vt =
            interp1(abs(fit(z).vt),time((start(z):stop(z))),vt_goal,'spline');
            p20(3).vt =
            plot(fail(z).vt,vt_goal,'o','Linewidth',1.25,'color',matchcolor{z
                });
            %display faiure time
            str1 = ['',num2str(round(fail(z).vt,2)),'s'];
            text(fail(z).vt,vt_goal-.01,str1,'VerticalAlignment','top',...
            'HorizontalAlignment','center','FontSize',(font-4))
        case 21
            time = timeS;
            p21(1).vt =  loglog(time((start(z):stop(z))),...
            vt_change(z).shift(start(z):stop(z)),'*','color',matchcolor{z});
            hold all
            pvt = polyfit(log(time((start(z)+1:stop(z)))),...
            log(vt_change(z).shift((start(z)+1:stop(z)))),1);
            fit(z).vt = exp(pvt(1)*log(time((start(z):stop(z))))+pvt(2));
            p21(2).vt =  loglog(time((start(z):stop(z))),fit(z).vt,'color',...
            matchcolor{z},'Linewidth',1.25);
            fail(z).vt =
            interp1(fit(z).vt,time((start(z):stop(z))),vt_goal,'spline');
            p21(3).vt =
            plot(fail(z).vt,vt_goal,'o','Linewidth',1.25,'color',matchcolor{z
                });
            str1 = ['',num2str(round(fail(z).vt,2)),'s'];
            text(fail(z).vt,vt_goal-.001,str1,'VerticalAlignment','top',...
            'HorizontalAlignment','center','FontSize',(font-4))
            q=q+1;
        case 205
            time = timeS;
            p205(1).vt = loglog(time((start(z):stop(z))),...
            vt_change(z).shift(start(z):stop(z)),'*','color',matchcolor{z});
            hold all
            pvt = polyfit(log(time((start(z)+1:stop(z)))),...
            log(vt_change(z).shift((start(z)+1:stop(z)))),1);
            fit(z).vt = exp(pvt(1)*log(time((start(z):stop(z))))+pvt(2));
            p205(2).vt =  loglog(time((start(z):stop(z))),fit(z).vt,'color'
                ,...
            matchcolor{z},'Linewidth',1.25);
            fail(z).vt =
            interp1(fit(z).vt,time((start(z):stop(z))),vt_goal,'spline');
            p205(3).vt =
            plot(fail(z).vt,vt_goal,'o','Linewidth',1.25,'color',matchcolor{z
                });
            str1 = ['',num2str(round(fail(z).vt,2)),'s'];
            text(fail(z).vt,vt_goal+.01,str1,'VerticalAlignment','bottom',...
            'HorizontalAlignment','center','FontSize',(font-4))
        case 19
            time = timeL;
            %used to extend fit line if needed
            timeper = linspace(10,3000,10^4);
            p19(1).vt =  loglog(time(start(z):stop(z)),...
            abs(vt_change(z).shift(start(z):stop(z))),'*','color',matchcolor{z
                });
            hold all
            pvt = polyfit(log(time((start(z)+1:stop(z)))),...
            log(abs(vt_change(z).shift((start(z)+1:stop(z))))),1);
            fit(z).vt = exp(pvt(1)*log(timeper)+pvt(2));
            p19(2).vt =
            loglog(timeper,fit(z).vt,'color',matchcolor{z},'Linewidth',1.25);
            fail(z).vt = interp1(fit(z).vt,timeper,vt_goal,'spline');
            p19(3).vt =
            plot(fail(z).vt,vt_goal,'o','Linewidth',1.25,'color',matchcolor{z
                });
            str1 = ['',num2str(round(fail(z).vt,2)),'s'];
            text(fail(z).vt,vt_goal+.01,str1,'VerticalAlignment','bottom',...
            'HorizontalAlignment','center','FontSize',(font-4))
        case 18
            time = timeL;
            timeper = time(start(z):stop(z));
            p18(1).vt = loglog(time(start(z):stop(z)),...
            vt_change(z).shift(start(z):stop(z)),'*','color',matchcolor{z});
            hold all
            pvt = polyfit(log(time((start(z)+1:stop(z)))),...
            log(vt_change(z).shift((start(z)+1:stop(z)))),1);
            fit(z).vt = exp(pvt(1)*log(timeper)+pvt(2));
            p18(2).vt =
            loglog(timeper,fit(z).vt,'color',matchcolor{z},'Linewidth',1.25);
            fail(z).vt = interp1(fit(z).vt,timeper,vt_goal,'spline');
            p18(3).vt =
            plot(fail(z).vt,vt_goal,'o','Linewidth',1.25,'color',matchcolor{z
                });
            str1 = ['',num2str(round(fail(z).vt,2)),'s'];
            text(fail(z).vt,vt_goal-.01,str1,'VerticalAlignment','top',...
            'HorizontalAlignment','center','FontSize',(font-4))
        case 16
            time = timeXL;
            timeper = linspace(300,1.3*10^5,10^4);
            p16(1).vt = loglog(time(start(z):stop(z)),...
            vt_change(z).shift(start(z):stop(z)),'*','color',matchcolor{z});
            hold all
            pvt = polyfit(log(time((start(z)+1:stop(z)))),...
            log(vt_change(z).shift((start(z)+1:stop(z)))),1);
            fit(z).vt = exp(pvt(1)*log(timeper)+pvt(2));
            p16(2).vt =
            loglog(timeper,fit(z).vt,'color',matchcolor{z},'Linewidth',1.25);
            fail(z).vt = interp1(fit(z).vt,timeper,vt_goal,'spline');
            p16(3).vt =
            plot(fail(z).vt,vt_goal,'o','Linewidth',1.25,'color',matchcolor{z
                });
            str1 = ['',num2str(round(fail(z).vt,2)),'s'];
            text(fail(z).vt,vt_goal-.01,str1,'VerticalAlignment','top',...
            'HorizontalAlignment','center','FontSize',(font-4))
        otherwise
    end
    clear time
end
%plot goals and format plots
goal_vt = plot(get(gca,'Xlim'),[vt_goal vt_goal],'k','Linewidth',1.25);
h = legend([p18(1).vt p19(1).vt p20(1).vt goal_vt ],...
'1.8V','1.9V', '2.0V','70mV␣shift','Location','Best');
set(h,'FontSize',(font-4));
box on
ylabel('\Delta␣V_{T}␣(V)','FontSize',font)
xlabel('Stress␣Time␣(s)','FontSize',font)
set(gcf,'color','w');
a = get(gca,'XTickLabel');
set(gca,'XTickLabel',a,'fontsize',(font-3),'XTickLabelMode','auto')
goal_vt = plot(get(gca,'Xlim'),[vt_goal vt_goal],'k','Linewidth',1.25);
tightfig();
%mttf plot
fig =figure(z+2);
fig.Units = 'inches';
fig.Position = [.5*8, .5, 6, 5];
%determine 1/(stress voltage)
for z = 1:length(fail)
    vds_stress(z) =
    1/(str2num(stress(z).title(24))+str2num(stress(z).title(25))/10);
    mttfV(z)=
    semilogy(vds_stress(z),fail(z).vt,'o','color',matchcolor{z},'Linewidth'
        ,1.25);
    fail_time(z) = fail(z).vt;
    hold all
end
%fit line through fail times
voltage_range = linspace(0,1.5,200);
p = polyfit(vds_stress,log(fail_time),1);
fit_time = exp(polyval(p,voltage_range));
plot(voltage_range,fit_time,'color','k')
hold all
year = 3.1536*10^7; %time of a year in seconds
%plot year and 20 year lines
plot(get(gca,'Xlim'),[year year],'--k');
plot(get(gca,'Xlim'),[20*year 20*year],'--k');
% find voltage at year and 20 years
yearVolt = interp1(fit_time,voltage_range,year,'pchip');
plot(yearVolt,year,'o','color','k','Linewidth',1.25)
str1 = ['',num2str(round(1/yearVolt,2)),'␣V_{DS}'];
text(yearVolt+.1,year-10000,str1,'VerticalAlignment',...
'top','HorizontalAlignment','center','FontSize',(font-4))
year = 20*year;
yearVolt = interp1(fit_time,voltage_range,year,'pchip');
```

```
plot(yearVolt,year,'o','color','k','Linewidth',1.25)
str1 = ['',num2str(round(1/(yearVolt),2)),'␣V_{DS}'];
text(yearVolt+.1,year+1000000000,str1,'VerticalAlignment',...
'bottom','HorizontalAlignment','center','FontSize',(font-4))
% find time of failure at 0.9V
nomVolt = interp1(voltage_range,fit_time,1.1111,'pchip')
plot(1.1111,nomVolt,'o','color','k','Linewidth',1.25);
str1 = ['',num2str(round(nomVolt/(year/20)),1),'␣Years'];
text(1.1111,nomVolt+100000,str1,'VerticalAlignment',...
'bottom','HorizontalAlignment','center','FontSize',(font-4))

% find time of failure at 1V
%plot([1 1],get(gca,'Ylim'),':k');
nomVolt = interp1(voltage_range,fit_time,1,'pchip')
plot(1,nomVolt,'o','color','k','Linewidth',1.25);
str1 = ['',num2str(round(nomVolt/(year/20)),1),'␣Years'];
text(1,nomVolt+10000,str1,'VerticalAlignment','top',...
'HorizontalAlignment','center','FontSize',(font-4))
%formatting
ylabel('Time␣in␣Seconds','FontSize',font)
xlabel('1/V_{DS}␣(1/V)','FontSize',font)
a = get(gca,'XTickLabel');
set(gca,'XTickLabel',a,'fontsize',(font-3),'XTickLabelMode','auto')
str1 = ['1␣Year'];
text(0.2-.01,year/20-10,str1,'VerticalAlignment','top',...
'HorizontalAlignment','center','FontSize',(font-4))
str1 = ['20␣Years'];
text(0.2-.01,year+1000,str1,'VerticalAlignment','bottom',...
'HorizontalAlignment','center','FontSize',(font-4))
str1 = ['0.9␣V_{DS}'];
text(1.1111+.01,10^-10,str1,'VerticalAlignment','bottom',...
'HorizontalAlignment','left','FontSize',(font-4))
str1 = ['1.0␣V_{DS}'];
text(1-.01,10^-10,str1,'VerticalAlignment','bottom',...
'HorizontalAlignment','right','FontSize',(font-4))
%plot voltage limits
plot([1.1111 1.1111],get(gca,'Ylim'),':k');
plot([1 1],get(gca,'Ylim'),':k');
h = legend([mttfV(2) mttfV(3) mttfV(1)],...
'1.8V', '1.9V', '2.0V','Location','Best');
set(h,'FontSize',(font-4));
tightfig();
set(gcf,'color','w');
```

D.4 Subthreshold

```
color = {'k';'m';'b';'g';'r'};
font = 20;
%file number
x= 1;
%step number
for k = 3
    semilogy(abs(fig(x).data(k).v),abs(fig(x).data(k).i),...
    '*','color',color{k});
    hold all
    box on
    ylabel('I_D (A)','FontSize',(font))
    xlabel('V_{GS} (V)','FontSize',(font))
    hold all
end
%fit log curve
P = polyfit(abs(fig(x).data(k).v), log(abs(fig(x).data(k).i)),6);
fit.log = exp(polyval(P, abs(fig(x).data(k).v)));
plot(abs(fig(x).data(k).v),fit.log,'-')
%get tick values on plot
tick = (get(gca,'ytick'))
%find value of Vgs at each tick
for n = 1:length(tick)
    Vx(n) =  interp1(fit.log,fig(x).data(k).v,tick(n));
end
diffe = zeros(1,length(tick));
%calculate difference between ticks
for n = 2:length(tick)
    if isnan(Vx(n-1))
    else
        diffe(n) = Vx(n)-Vx(n-1);
    end
end
dd = abs(diffe);
%display highest subthreshold slope
maxV = min(dd(dd>0))
```

D.5 IC-CAP

```
vg_start =num2str(0); %Measurement settings
vg_stop =num2str(0.9);
vg_step =num2str(0.0225);
vg_points =num2str(51);
vd_start =num2str(0.225);
vd_stop =num2str(.9);
vd_step =num2str(0.225);
vd_points =num2str(4);
%inData This data format is found on the MATLAB GUI on the lab
% computer. Input numbers of the files that are being averaged
inDataNum = [1 2 3 4 5];
%open file with desired name
fileID = fopen('VD.mdm','w');
%Writing file
fprintf(fileID,'!␣VERSION␣=␣6.00\r\n');
fprintf(fileID,'BEGIN_HEADER\r\n');
fprintf(fileID,'␣ICCAP_INPUTS\r\n');
fprintf(fileID,[[['␣␣vd␣␣␣␣␣␣␣␣␣␣V␣␣D_GROUND_SMU1␣0.1␣LIN ...
1␣␣␣␣',vg_start],['␣␣␣␣␣␣␣␣␣␣␣',vg_stop],['␣␣␣␣␣␣␣␣␣␣␣',vg_points],...
['␣␣␣',vg_step]],'\r\n']);
fprintf(fileID,[[['␣␣vg␣␣␣␣␣␣␣␣␣␣V␣␣G_GROUND_SMU2␣0.1␣LIN ...
2␣␣␣␣',vd_start],['␣␣␣␣␣␣␣␣␣␣␣',vd_stop],['␣␣␣␣␣␣␣␣␣␣␣',vd_points],...
['␣␣␣␣',vd_step]],'\r\n']);
fprintf(fileID,'␣␣vs␣␣␣␣␣␣␣␣␣␣V␣␣S_GROUND_SMU3␣0.1␣LIST ...
3␣1␣0\r\n');
fprintf(fileID,'ICCAP_OUTPUTS\r\n');
fprintf(fileID,'␣␣id␣␣␣␣␣␣␣␣␣␣I␣␣D_GROUND_SMU1_B\r\n');
fprintf(fileID,'␣␣ig␣␣␣␣␣␣␣␣␣␣I␣␣G_GROUND_SMU2_B\r\n');
fprintf(fileID,'␣␣is␣␣␣␣␣␣␣␣␣␣I␣␣S_GROUND_SMU3_B\r\n');
fprintf(fileID,'ICCAP_VALUES\r\n');
fprintf(fileID,'␣␣MASTER_SETUP_TYPE␣"~soi_dc_idvg~"\r\n');
fprintf(fileID,'␣␣LINVT_VALUE␣""\r\n');
fprintf(fileID,'␣␣LISTVT_VALUE␣""\r\n');
fprintf(fileID,'␣␣TYPE␣"1"\r\n');
%Input Width and length
fprintf(fileID,'␣␣MAIN.W␣"2.86u"\r\n');
fprintf(fileID,'␣␣MAIN.L␣"40n"\r\n');
fprintf(fileID,'␣␣MAIN.AD␣"5.000p"\r\n');
fprintf(fileID,'␣␣MAIN.AS␣"5.000p"\r\n');
fprintf(fileID,'␣␣MAIN.PD␣"11.00u"\r\n');
fprintf(fileID,'␣␣MAIN.PS␣"11.00u"\r\n');
fprintf(fileID,'␣␣MAIN.NF␣"1.000"\r\n');
fprintf(fileID,'␣␣MAIN.M␣"1"\r\n');
fprintf(fileID,'␣␣MAIN.NRD␣"0"\r\n');
fprintf(fileID,'␣␣MAIN.NRS␣"0"\r\n');
fprintf(fileID,'␣␣MAIN.LDRIFT1␣"1E-06"\r\n');
fprintf(fileID,'␣␣MAIN.LDRIFT2␣"1E-06"\r\n');
fprintf(fileID,'␣␣MAIN.SA␣"3.000u"\r\n');
fprintf(fileID,'␣␣MAIN.SB␣"3.000u"\r\n');
fprintf(fileID,'␣␣MAIN.SD␣"0.000"\r\n');
fprintf(fileID,'␣␣MAIN.SCA␣"0"\r\n');
fprintf(fileID,'␣␣MAIN.SCB␣"0"\r\n');
fprintf(fileID,'␣␣MAIN.SCC␣"0"\r\n');
fprintf(fileID,'␣␣MAIN.SC␣"0"\r\n');
fprintf(fileID,'␣␣MODULE␣"Mod_T1"\r\n');
fprintf(fileID,'␣␣PIN_D␣"1"\r\n');
fprintf(fileID,'␣␣PIN_G␣"2"\r\n');
fprintf(fileID,'␣␣PIN_S␣"31"\r\n');
fprintf(fileID,'␣␣PIN_B␣"32"\r\n');
fprintf(fileID,'␣␣PIN_E␣"33"\r\n');
fprintf(fileID,'␣␣PIN_6␣""\r\n');
fprintf(fileID,'␣␣XY_ID␣""\r\n');
fprintf(fileID,'␣␣FUNC␣"Large"\r\n');
fprintf(fileID,'␣␣STI␣"SA␣ref"\r\n');
fprintf(fileID,'␣␣WPE␣"WPE␣ref"\r\n');
fprintf(fileID,'␣␣COMMENT␣""\r\n');
fprintf(fileID,'␣␣DIS_MEAS␣""\r\n');
fprintf(fileID,'␣␣TypeTCI␣"11"\r\n');
fprintf(fileID,'END_HEADER\r\n\n');
%input step for header
step = [0.225 0.450 0.675 0.900];
for n = 2:size(inData.Data_Structure{inDataNum(1)}.Cycle.VG,2);
    fprintf(fileID,'BEGIN_DB\r\n');
    fprintf(fileID,['␣ICCAP_VAR␣vd␣␣␣␣␣␣␣␣␣␣',...
    num2str(step(n-1)),'\r\n']);
    fprintf(fileID,'␣ICCAP_VAR␣vs␣␣␣␣␣␣␣␣␣␣0\r\n\n');
    fprintf(fileID,'␣#vd␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣id...
␣␣␣␣ig␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣ib\r\n');
    %read data and average
    for k = 1:length(inData.Data_Structure{inDataNum(1)}.Cycle.VD);
        fprintf(fileID,['␣␣',
        num2str(mean([...
        inData.Data_Structure{inDataNum(1)}.Cycle.VD(k,n);...
        inData.Data_Structure{inDataNum(2)}.Cycle.VD(k,n);...
        inData.Data_Structure{inDataNum(3)}.Cycle.VD(k,n) ...
        ;inData.Data_Structure{inDataNum(4)}.Cycle.VD(k,n)])),'␣␣␣␣␣␣␣␣',
        num2str(mean([...
        inData.Data_Structure{inDataNum(1)}.Cycle.ID(k,n);...
        inData.Data_Structure{inDataNum(2)}.Cycle.ID(k,n);...
        inData.Data_Structure{inDataNum(3)}.Cycle.ID(k,n);...
        inData.Data_Structure{inDataNum(4)}.Cycle.ID(k,n)])),'␣␣␣␣␣␣␣␣',...
        num2str(mean([...
        inData.Data_Structure{inDataNum(1)}.Cycle.IG(k,n);...
        inData.Data_Structure{inDataNum(2)}.Cycle.IG(k,n);...
        inData.Data_Structure{inDataNum(3)}.Cycle.IG(k,n);...
        inData.Data_Structure{inDataNum(4)}.Cycle.IG(k,n)])),'␣␣␣␣␣␣␣␣',...
        num2str(mean([...
        inData.Data_Structure{inDataNum(1)}.Cycle.IS(k,n);...
        inData.Data_Structure{inDataNum(2)}.Cycle.IS(k,n);...
        inData.Data_Structure{inDataNum(3)}.Cycle.IS(k,n);...
        inData.Data_Structure{inDataNum(4)}.Cycle.IS(k,n)])),'\r\n']);
    end
    fprintf(fileID,'END_DB\r\n\n');
end
fclose(fileID);
```

Appendix E
IC-CAP Modeling Fitting Tutorial

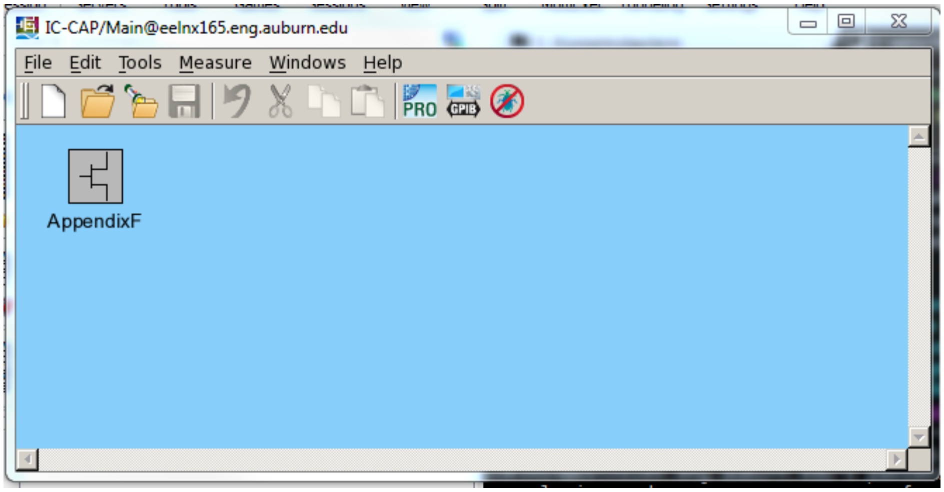
The following tutorial details the steps of creating a cryogenic SPICE model using IC-CAP. The model circuit used is the BSIMSOIv4.4 model.

E.1 Process Overview

1. Create a new model
2. Add SPICE circuit and variables
3. Add DUT
4. Device setup
5. Optimization
6. Extraction

E.1.1 Creating a New Model

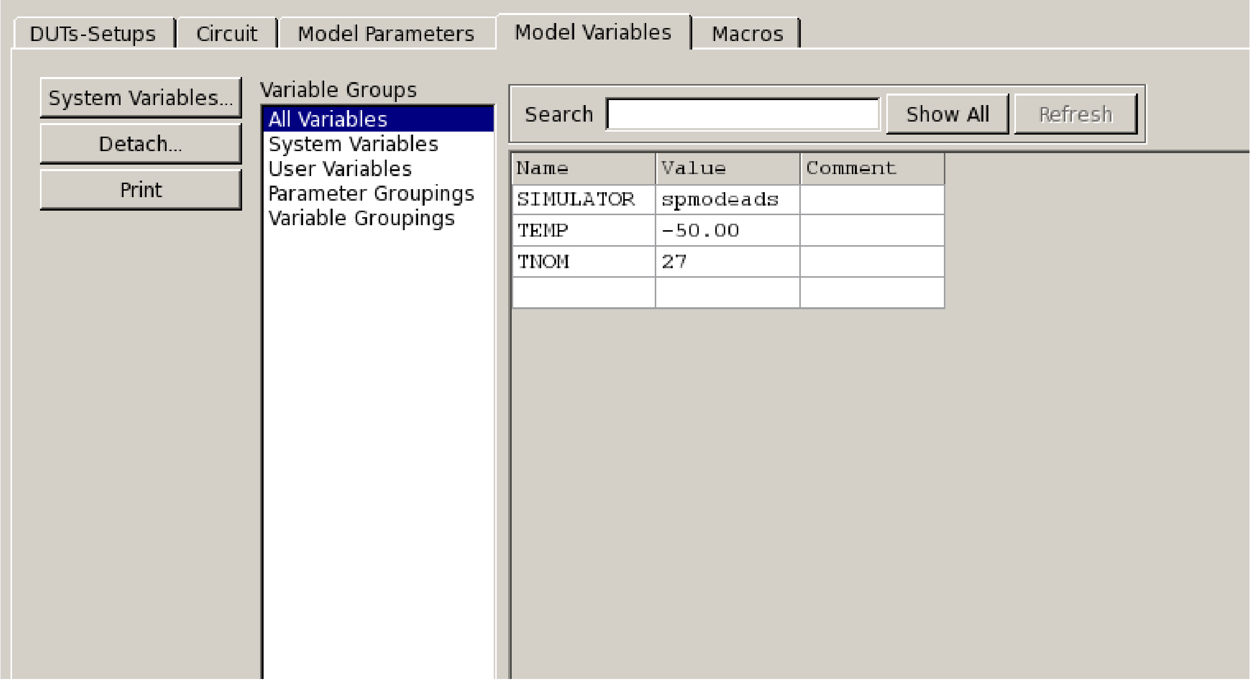
On the IC-CAP main screen click the white blank sheet of paper on the left side of the tool bar to create a new model.



E.1.2 Adding SPICE Circuit and Variables

1. Setting model variables

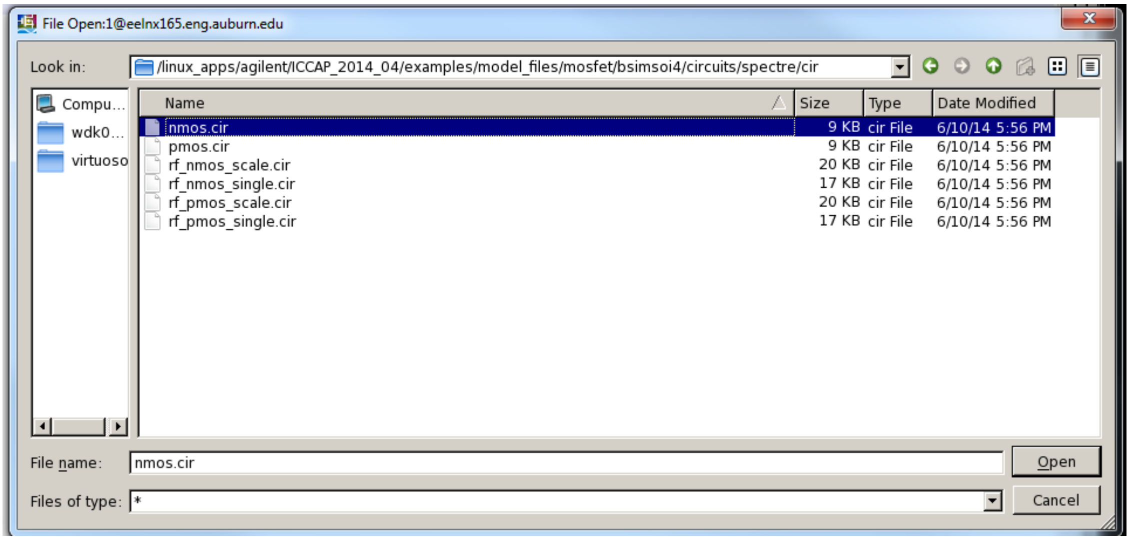
   SPMODEADS is a simulator template that allows IC-CAP to read Spectre files.



Name	Value	Comment
SIMULATOR	spmodeads	
TEMP	-50.00	
TNOM	27	

2. Adding the transistor circuits

   By clicking on ”Import Text” under the circuits tab model circuits can be added to the model. Model circuits for the BSIMSOIv4.4 model can be found under the ICCAP example files.



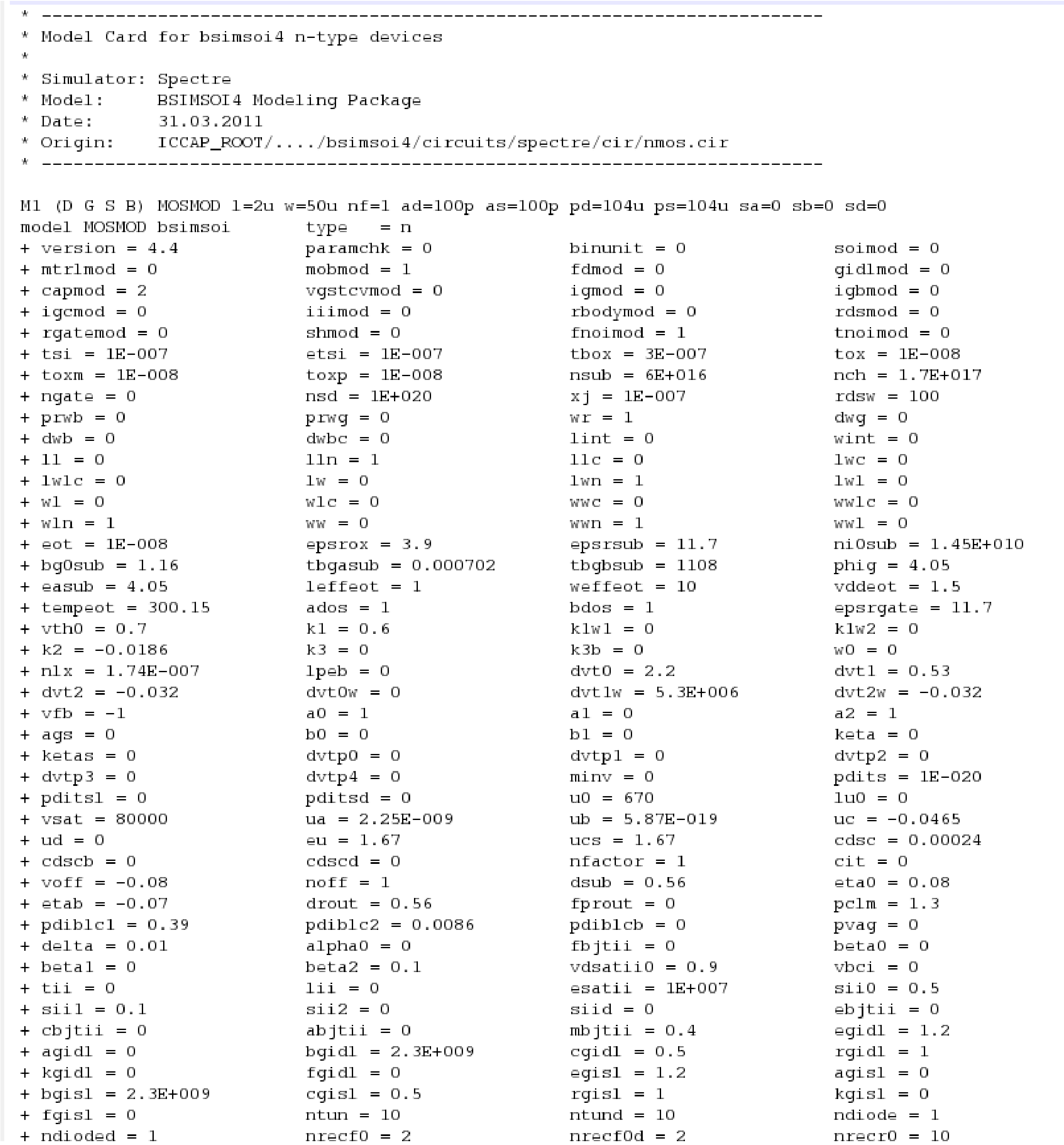
Look in: /linux_apps/agilent/ICCAP_2014_04/examples/model_files/mosfet/bsimsoi4/circuits/spectre/cir

Name	Size	Type	Date Modified
nmos.cir	9 KB	cir File	6/10/14 5:56 PM
pmos.cir	9 KB	cir File	6/10/14 5:56 PM
rf_nmos_scale.cir	20 KB	cir File	6/10/14 5:56 PM
rf_nmos_single.cir	17 KB	cir File	6/10/14 5:56 PM
rf_pmos_scale.cir	20 KB	cir File	6/10/14 5:56 PM
rf_pmos_single.cir	17 KB	cir File	6/10/14 5:56 PM

File name: nmos.cir

Files of type: *

3. Example BSIMSOIv4.4 circuit

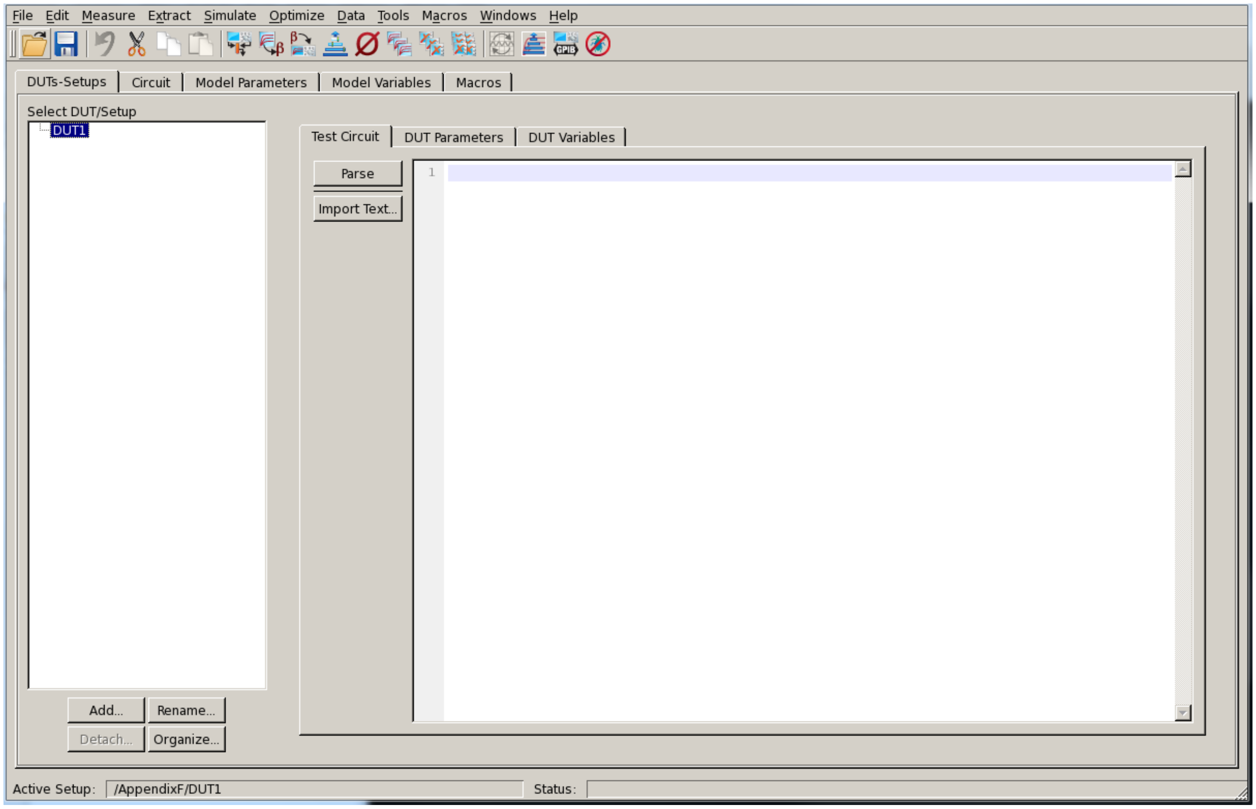


```
* --------------------------------------------------------------------------
* Model Card for bsimsoi4 n-type devices
*
* Simulator: Spectre
* Model:     BSIMSOI4 Modeling Package
* Date:      31.03.2011
* Origin:    ICCAP_ROOT/..../bsimsoi4/circuits/spectre/cir/nmos.cir
* --------------------------------------------------------------------------

M1 (D G S B) MOSMOD l=2u w=50u nf=1 ad=100p as=100p pd=104u ps=104u sa=0 sb=0 sd=0
model MOSMOD bsimsoi       type   = n
+ version = 4.4            paramchk = 0             binunit = 0              soimod = 0
+ mtrlmod = 0              mobmod = 1               fdmod = 0                gidlmod = 0
+ capmod = 2               vgstcvmod = 0            igmod = 0                igbmod = 0
+ igcmod = 0               iiimod = 0               rbodymod = 0             rdsmod = 0
+ rgatemod = 0             shmod = 0                fnoimod = 1              tnoimod = 0
+ tsi = 1E-007             etsi = 1E-007            tbox = 3E-007            tox = 1E-008
+ toxm = 1E-008            toxp = 1E-008            nsub = 6E+016            nch = 1.7E+017
+ ngate = 0                nsd = 1E+020             xj = 1E-007              rdsw = 100
+ prwb = 0                 prwg = 0                 wr = 1                   dwg = 0
+ dwb = 0                  dwbc = 0                 lint = 0                 wint = 0
+ ll = 0                   lln = 1                  llc = 0                  lwc = 0
+ lwlc = 0                 lw = 0                   lwn = 1                  lwl = 0
+ wl = 0                   wlc = 0                  wwc = 0                  wwlc = 0
+ wln = 1                  ww = 0                   wwn = 1                  wwl = 0
+ eot = 1E-008             epsrox = 3.9             epsrsub = 11.7           ni0sub = 1.45E+010
+ bg0sub = 1.16            tbgasub = 0.000702       tbgbsub = 1108           phig = 4.05
+ easub = 4.05             leffeot = 1              weffeot = 10             vddeot = 1.5
+ tempeot = 300.15         ados = 1                 bdos = 1                 epsrgate = 11.7
+ vth0 = 0.7               k1 = 0.6                 k1w1 = 0                 k1w2 = 0
+ k2 = -0.0186             k3 = 0                   k3b = 0                  w0 = 0
+ nlx = 1.74E-007          lpeb = 0                 dvt0 = 2.2               dvt1 = 0.53
+ dvt2 = -0.032            dvt0w = 0                dvt1w = 5.3E+006         dvt2w = -0.032
+ vfb = -1                 a0 = 1                   a1 = 0                   a2 = 1
+ ags = 0                  b0 = 0                   b1 = 0                   keta = 0
+ ketas = 0                dvtp0 = 0                dvtp1 = 0                dvtp2 = 0
+ dvtp3 = 0                dvtp4 = 0                minv = 0                 pdits = 1E-020
+ pditsl = 0               pditsd = 0               u0 = 670                 lu0 = 0
+ vsat = 80000             ua = 2.25E-009           ub = 5.87E-019           uc = -0.0465
+ ud = 0                   eu = 1.67                ucs = 1.67               cdsc = 0.00024
+ cdscb = 0                cdscd = 0                nfactor = 1              cit = 0
+ voff = -0.08             noff = 1                 dsub = 0.56              eta0 = 0.08
+ etab = -0.07             drout = 0.56             fprout = 0               pclm = 1.3
+ pdiblc1 = 0.39           pdiblc2 = 0.0086         pdiblcb = 0              pvag = 0
+ delta = 0.01             alpha0 = 0               fbjtii = 0               beta0 = 0
+ beta1 = 0                beta2 = 0.1              vdsatii0 = 0.9           vbci = 0
+ tii = 0                  lii = 0                  esatii = 1E+007          sii0 = 0.5
+ sii1 = 0.1               sii2 = 0                 siid = 0                 ebjtii = 0
+ cbjtii = 0               abjtii = 0               mbjtii = 0.4             egidl = 1.2
+ agidl = 0                bgidl = 2.3E+009         cgidl = 0.5              rgidl = 1
+ kgidl = 0                fgidl = 0                egisl = 1.2              agisl = 0
+ bgisl = 2.3E+009         cgisl = 0.5              rgisl = 1                kgisl = 0
+ fgisl = 0                ntun = 10                ntund = 10               ndiode = 1
+ ndioded = 1              nrecf0 = 2               nrecf0d = 2              nrecr0 = 10
```

E.1.3 Adding DUT and Setups

1. Adding DUT

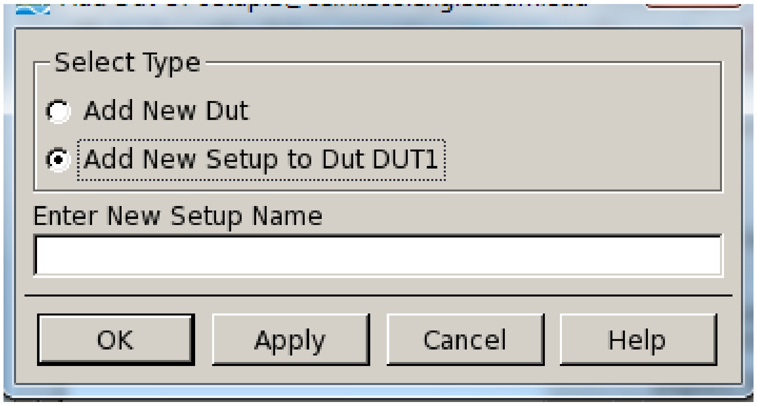
   Clicking on the ”DUTs-Setups” tab, a new DUT can be added by clicking ”Add” in the lower left corner.



Active Setup: /AppendixF/DUT1

2. Adding Setups

   By clicking ”Add” in the bottom left corner and selecting ”Add New Setup to Dut” a new setup is added. A setup is is a measurement that is performed, such as $I_D - V_{GS}$ for transistors.



Select Type

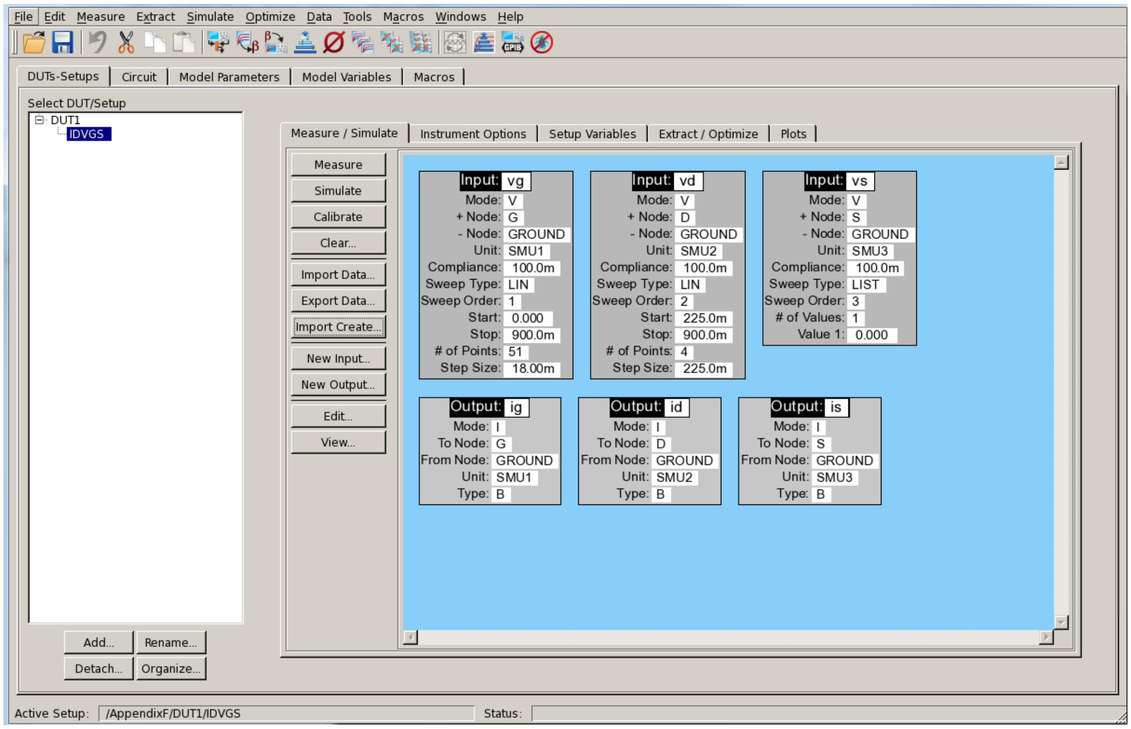
○ Add New Dut

● Add New Setup to Dut DUT1

Enter New Setup Name

3. Import create

   Using the MATLAB script in Appendix D.5 to generate the file, the measured data collected can be added to the setup by clicking ”Import Create” in the ”Measure/Simulate” tab.



Select DUT/Setup
DUT1
IDVGS

Input: vg
Mode: V
+ Node: G
- Node: GROUND
Unit: SMU1
Compliance: 100.0m
Sweep Type: LIN
Sweep Order: 1
Start: 0.000
Stop: 900.0m
# of Points: 51
Step Size: 18.00m

Input: vd
Mode: V
+ Node: D
- Node: GROUND
Unit: SMU2
Compliance: 100.0m
Sweep Type: LIN
Sweep Order: 2
Start: 225.0m
Stop: 900.0m
# of Points: 4
Step Size: 225.0m

Input: vs
Mode: V
+ Node: S
- Node: GROUND
Unit: SMU3
Compliance: 100.0m
Sweep Type: LIST
Sweep Order: 3
# of Values: 1
Value 1: 0.000

Output: ig
Mode: I
To Node: G
From Node: GROUND
Unit: SMU1
Type: B

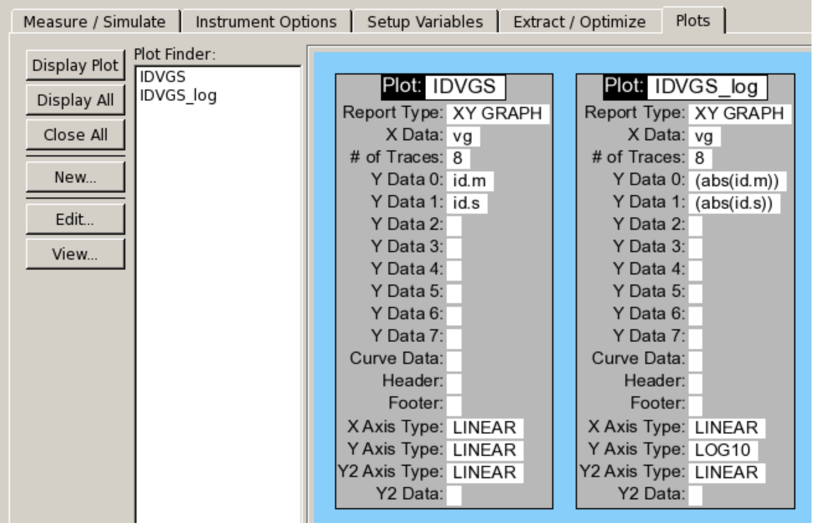
Output: id
Mode: I
To Node: D
From Node: GROUND
Unit: SMU2
Type: B

Output: is
Mode: I
To Node: S
From Node: GROUND
Unit: SMU3
Type: B

Active Setup: /AppendixF/DUT1/IDVGS
Status:

4. Setting up plots

   Under the plot tab of each Setup, plots are added to plot the results by clicking ”New” on the left side. The notations *id.m* and *id.s* stand for I_D of the measured and simulated data.



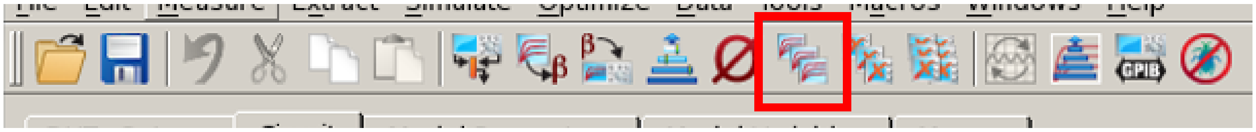
Plot Finder:
IDVGS
IDVGS_log

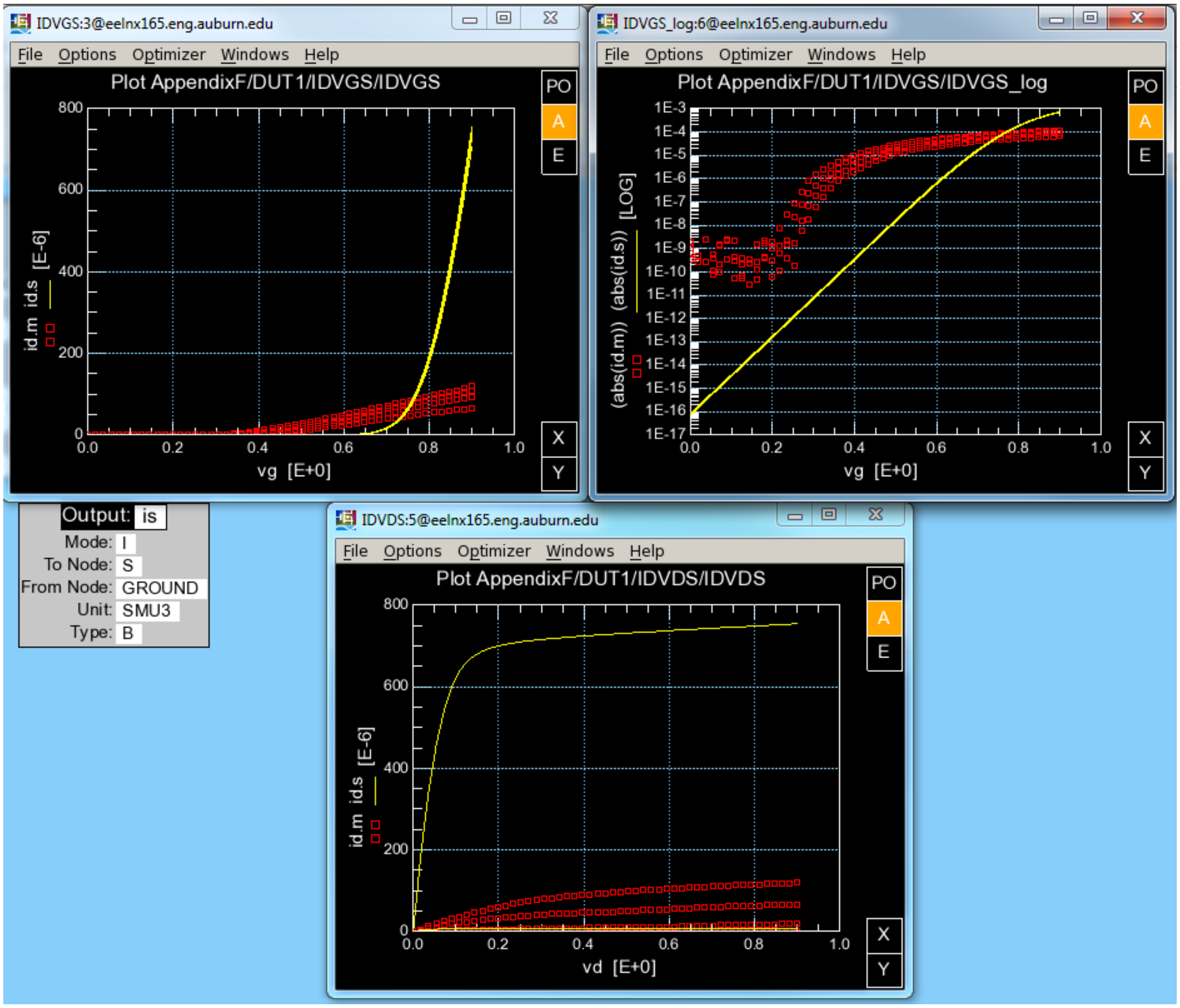
Plot: IDVGS
Report Type: XY GRAPH
X Data: vg
# of Traces: 8
Y Data 0: id.m
Y Data 1: id.s
Y Data 2:
Y Data 3:
Y Data 4:
Y Data 5:
Y Data 6:
Y Data 7:
Curve Data:
Header:
Footer:
X Axis Type: LINEAR
Y Axis Type: LINEAR
Y2 Axis Type: LINEAR
Y2 Data:

Plot: IDVGS_log
Report Type: XY GRAPH
X Data: vg
# of Traces: 8
Y Data 0: (abs(id.m))
Y Data 1: (abs(id.s))
Y Data 2:
Y Data 3:
Y Data 4:
Y Data 5:
Y Data 6:
Y Data 7:
Curve Data:
Header:
Footer:
X Axis Type: LINEAR
Y Axis Type: LOG10
Y2 Axis Type: LINEAR
Y2 Data:

5. Simulate and plot

   Finally after all the Setups are complete, each Setup can be simulated by clicking ”Simulate” under the ”Measure/Simulate” tab. Then, by clicking ”Display Plots” in the tool bar shown below, the initial plots of the measured and simulated data are displayed.





IDVGS:3@eelnx165.eng.auburn.edu
File Options Optimizer Windows Help
Plot AppendixF/DUT1/IDVGS/IDVGS
PO
A
E
X
Y
800
600
400
200
0
0.0
0.2
0.4
0.6
0.8
1.0
id.m id.s [E-6]
vg [E+0]
IDVGS_log:6@eelnx165.eng.auburn.edu
File Options Optimizer Windows Help
Plot AppendixF/DUT1/IDVGS/IDVGS_log
1E-3
1E-4
1E-5
1E-6
1E-7
1E-8
1E-9
1E-10
1E-11
1E-12
1E-13
1E-14
1E-15
1E-16
1E-17
(abs(id.m)) (abs(id.s)) [LOG]
vg [E+0]
IDVDS:5@eelnx165.eng.auburn.edu
File Options Optimizer Windows Help
Plot AppendixF/DUT1/IDVDS/IDVDS
vd [E+0]

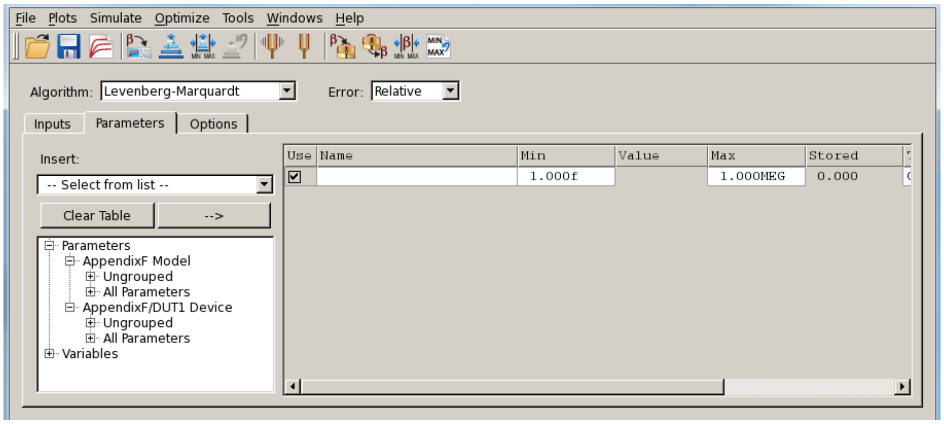
Output: is
Mode: I
To Node: S
From Node: GROUND
Unit: SMU3
Type: B

E.1.4 Optimization

1. Enable plot optimization by clicking ”PO” in the corner of all the plots.

2. Adding optimization function

   To open a new optimization function click ”Open Optimizer” under the optimizer tab on the plot window.



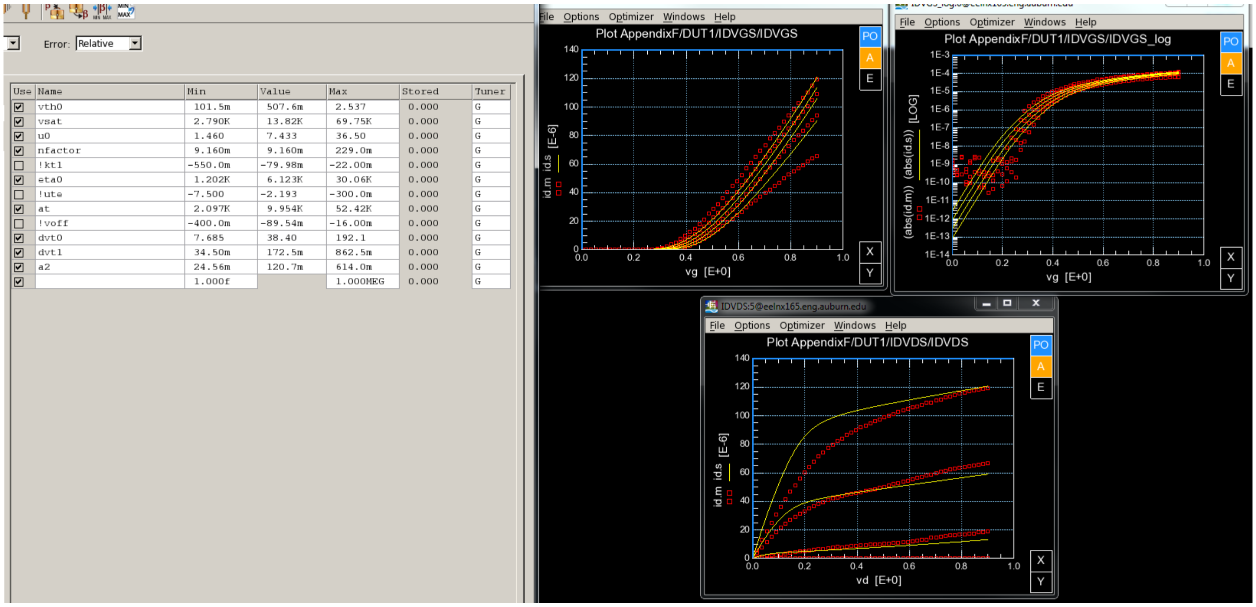
Algorithm: Levenberg-Marquardt

Error: Relative

Use	Name	Min	Value	Max	Stored
☑		1.000f		1.000MEG	0.000

3. Adding parameters

   All of the parameters from the model circuit are listed under the ”Parameters” tab on the right side. Clicking on the parameter will enable the parameter to be optimized. By clicking ”Autoset Min/Max and Optimized” in the tool bar all of the parameters enabled are optimized using the algorithm selected. After a few cycles of clicking ”Autoset Min/Max and Optimized” the simulated data will begin to conform to the measured data.

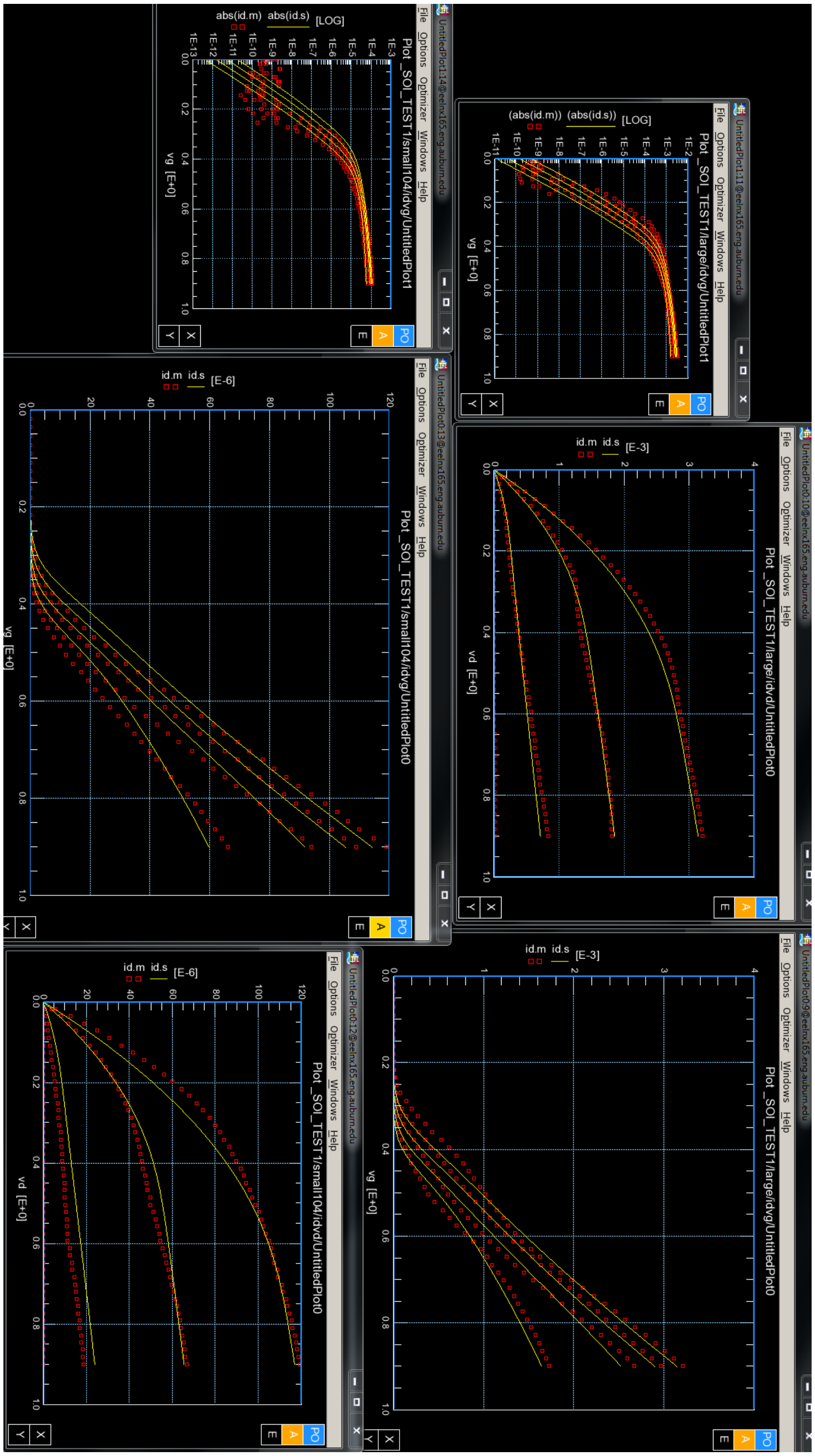

Plot AppendixF/DUT1/IDVGS/IDVGS
id.m ids [E-6]
vg [E+0]
Plot AppendixF/DUT1/IDVGS/IDVGS_log
(abs(id.m)) (abs(ids)) [LOG]
vg [E+0]
IDVDS:5@eelnx165.eng.auburn.edu
File Options Optimizer Windows Help
Plot AppendixF/DUT1/IDVDS/IDVDS
id.m ids [E-6]
vd [E+0]
PO
A
E
X
Y

Error: Relative

Use	Name	Min	Value	Max	Stored	Tuner
☑	vth0	101.5m	507.6m	2.537	0.000	G
☑	vsat	2.790K	13.82K	69.75K	0.000	G
☑	u0	1.460	7.433	36.50	0.000	G
☑	nfactor	9.160m	9.160m	229.0m	0.000	G
□	!kt1	-550.0m	-79.98m	-22.00m	0.000	G
☑	eta0	1.202K	6.123K	30.06K	0.000	G
□	!ute	-7.500	-2.193	-300.0m	0.000	G
☑	at	2.097K	9.954K	52.42K	0.000	G
□	!voff	-400.0m	-89.54m	-16.00m	0.000	G
☑	dvt0	7.685	38.40	192.1	0.000	G
☑	dvt1	34.50m	172.5m	862.5m	0.000	G
☑	a2	24.56m	120.7m	614.0m	0.000	G
☑		1.000f		1.000MEG	0.000	G

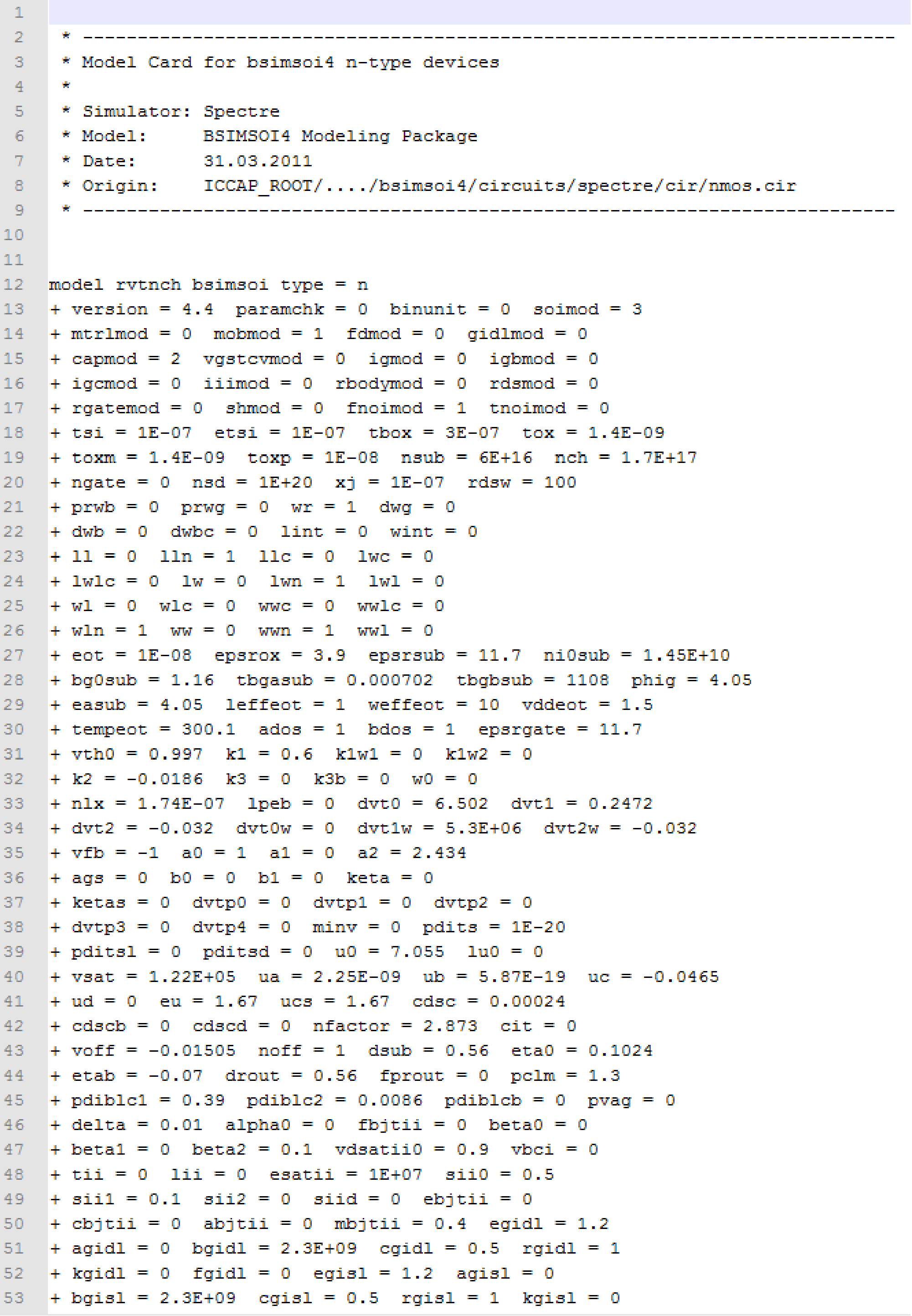
4. Multiple setup optimization

   By completing E.1.3 for multiple DUTs the model can be optimized for multiple devices simultaneously.


UntitledPlot1.14@eelnx165.eng.auburn.edu
File Options Optimizer Windows Help
Plot_SOI_TEST1/small104/dvdg/UntitledPlot1
abs(id.m) abs(id.s) [LOG]
vg [E+0]
UntitledPlot1.11@eelnx165.eng.auburn.edu
Plot_SOI_TEST1/large/dvdg/UntitledPlot1
(abs(id.m)) (abs(id.s)) [LOG]
vg [E+0]
UntitledPlot0.13@eelnx165.eng.auburn.edu
Plot_SOI_TEST1/small104/dvdg/UntitledPlot0
id.m id.s [E-6]
vg [E+0]
UntitledPlot0.10@eelnx165.eng.auburn.edu
Plot_SOI_TEST1/large/idvd/UntitledPlot0
id.m id.s [E-3]
vd [E+0]
UntitledPlot0.12@eelnx165.eng.auburn.edu
Plot_SOI_TEST1/small104/idvd/UntitledPlot0
id.m id.s [E-6]
vd [E+0]
UntitledPlot0.9@eelnx165.eng.auburn.edu
Plot_SOI_TEST1/large/dvdg/UntitledPlot0
id.m id.s [E-3]
vg [E+0]
PO
A
E
X
Y

E.1.5 Extraction

On the model window, clicking ”File”>”Export Data”>”Extracted Deck” the model card is extracted. By taking this model card and adding it to the model library of Cadence, simulation can be performed on the new cryogenic model.



```

 * ---------------------------------------------------------------------------
 * Model Card for bsimsoi4 n-type devices
 *
 * Simulator: Spectre
 * Model:     BSIMSOI4 Modeling Package
 * Date:      31.03.2011
 * Origin:    ICCAP_ROOT/..../bsimsoi4/circuits/spectre/cir/nmos.cir
 * ---------------------------------------------------------------------------


model rvtnch bsimsoi type = n
+ version = 4.4  paramchk = 0  binunit = 0  soimod = 3
+ mtrlmod = 0  mobmod = 1  fdmod = 0  gidlmod = 0
+ capmod = 2  vgstcvmod = 0  igmod = 0  igbmod = 0
+ igcmod = 0  iiimod = 0  rbodymod = 0  rdsmod = 0
+ rgatemod = 0  shmod = 0  fnoimod = 1  tnoimod = 0
+ tsi = 1E-07  etsi = 1E-07  tbox = 3E-07  tox = 1.4E-09
+ toxm = 1.4E-09  toxp = 1E-08  nsub = 6E+16  nch = 1.7E+17
+ ngate = 0  nsd = 1E+20  xj = 1E-07  rdsw = 100
+ prwb = 0  prwg = 0  wr = 1  dwg = 0
+ dwb = 0  dwbc = 0  lint = 0  wint = 0
+ ll = 0  lln = 1  llc = 0  lwc = 0
+ lwlc = 0  lw = 0  lwn = 1  lwl = 0
+ wl = 0  wlc = 0  wwc = 0  wwlc = 0
+ wln = 1  ww = 0  wwn = 1  wwl = 0
+ eot = 1E-08  epsrox = 3.9  epsrsub = 11.7  ni0sub = 1.45E+10
+ bg0sub = 1.16  tbgasub = 0.000702  tbgbsub = 1108  phig = 4.05
+ easub = 4.05  leffeot = 1  weffeot = 10  vddeot = 1.5
+ tempeot = 300.1  ados = 1  bdos = 1  epsrgate = 11.7
+ vth0 = 0.997  k1 = 0.6  k1w1 = 0  k1w2 = 0
+ k2 = -0.0186  k3 = 0  k3b = 0  w0 = 0
+ nlx = 1.74E-07  lpeb = 0  dvt0 = 6.502  dvt1 = 0.2472
+ dvt2 = -0.032  dvt0w = 0  dvt1w = 5.3E+06  dvt2w = -0.032
+ vfb = -1  a0 = 1  a1 = 0  a2 = 2.434
+ ags = 0  b0 = 0  b1 = 0  keta = 0
+ ketas = 0  dvtp0 = 0  dvtp1 = 0  dvtp2 = 0
+ dvtp3 = 0  dvtp4 = 0  minv = 0  pdits = 1E-20
+ pditsl = 0  pditsd = 0  u0 = 7.055  lu0 = 0
+ vsat = 1.22E+05  ua = 2.25E-09  ub = 5.87E-19  uc = -0.0465
+ ud = 0  eu = 1.67  ucs = 1.67  cdsc = 0.00024
+ cdscb = 0  cdscd = 0  nfactor = 2.873  cit = 0
+ voff = -0.01505  noff = 1  dsub = 0.56  eta0 = 0.1024
+ etab = -0.07  drout = 0.56  fprout = 0  pclm = 1.3
+ pdiblc1 = 0.39  pdiblc2 = 0.0086  pdiblcb = 0  pvag = 0
+ delta = 0.01  alpha0 = 0  fbjtii = 0  beta0 = 0
+ beta1 = 0  beta2 = 0.1  vdsatii0 = 0.9  vbci = 0
+ tii = 0  lii = 0  esatii = 1E+07  sii0 = 0.5
+ sii1 = 0.1  sii2 = 0  siid = 0  ebjtii = 0
+ cbjtii = 0  abjtii = 0  mbjtii = 0.4  egidl = 1.2
+ agidl = 0  bgidl = 2.3E+09  cgidl = 0.5  rgidl = 1
+ kgidl = 0  fgidl = 0  egisl = 1.2  agisl = 0
+ bgisl = 2.3E+09  cgisl = 0.5  rgisl = 1  kgisl = 0
```