

AUTOMATION OF THERMAL MODEL EXTRACTION
AND OPTIMIZATION

by

OBIORAH OJI

Presented to the Faculty of the College of Engineering of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

DECEMBER 2008

Copyright © by Obiorah Oji 2008

All Rights Reserved

ACKNOWLEDGEMENTS

I would first like to thank my family for their continued support of me in my academic endeavors. Also, Dr. Ronald L. Carter, my advisor for giving me an opportunity in the Analog Research group and through whom, I learnt a lot, not only objective technical information but the way to carry out engineering investigation. Drs. Alan Davis and Howard Russell were also instrumental in my time at UTA. I have much indebtedness to them. Finally, I would like to thank the Lord Jesus, the One "who calls things not being as being." I surely experienced Him in such a way in my academic endeavors. Praise Him!

August 18, 2008

ABSTRACT
AUTOMATION OF THERMAL MODEL EXTRACTION
AND OPTIMIZATION METHODOLOGY

Obiorah Oji, M.S.

The University of Texas at Arlington, 2008

Supervising Professor: Ronald L. Carter

Semiconductor device model optimization is highly beneficial; it improves the accuracy with which the performance of real devices and whole circuits can be predicted before they are fabricated. This helps save time and money in being able to predict undesirable circuit responses.

Although, it proves to be very useful, it can become very time consuming. The goal of this thesis is to use readily available tools to enhance and automate the somewhat complex process of optimization and streamline it so that results can be analyzed quicker with the goal of generating VBIC models that to the a high degree of accuracy match the actual behavior of the real device.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS.....	iii
ABSTRACT.....	iv
LIST OF ILLUSTRATIONS.....	vii
LIST OF TABLES.....	viii
Chapter	Page
1. INTRODUCTION.....	1
1.1 Background.....	1
1.2 VBIC Model.....	3
1.2.1 Brief History and Description.....	3
1.2.2 Thermal Resistance.....	4
1.2.3 Thermal Capacitance	4
1.2.4 Self Heating (VBIC Parameter).....	5
1.2.5 The Thermal Model.....	5
1.2.6 Use of the VBIC Model.....	6
1.3 Thermal Modeling and Optimization.....	7
2. MEASUREMENT TOOLS INTERFACE AND ICCAP.....	8
2.1 Introduction.....	8
2.2 Bench Set-up.....	8
2.3 Hardware Interfacing.....	9
2.4 Data Retrieval.....	10
2.4.1 Importing Data into Simulation Environment.....	12
2.5 ICCAP.....	14
2.5.1 Overview.....	14

2.5.2 Components of ICCAP.....	15
2.5.3 ICCAP User Environment.....	16
3. ONE- AND TWO-POLE MODELING IN ICCAP.....	18
3.1 One Pole Model.....	18
3.2 Two Pole Model.....	22
3.3 Results from ICCAP Implementation	25
APPENDIX	
A. READING DATA FROM AGILENT DSO6014A.....	26
B. SOURCE CODE FOR DATA ACQUISITION AND MODELING	29
REFERENCES.....	36
BIOGRAPHICAL INFORMATION.....	37

LIST OF ILLUSTRATIONS

Figure		Page
1.1	Cross Section of VIP 10 Process BJT	2
1.2	Complete thermal network of the VBIC model	6
1.3	Symbol of Transistor with Thermal Nodes.....	7
2.1	Circuit for Model Measurements.....	9
2.2	Test Pattern Sine Wave in ICCAP plot	13
2.3	Test Pattern Sine Wave on the Agilent DSO6014A.....	14
2.4	Components of ICCAP.....	16
2.5	ICCAP User Environment Snapshot.....	19
3.1	Normalized Exponential Decay	20
3.2	Line fit to find 1-Pole Time Constant	20
3.3	Plot of measured data and the $(A1 \cdot \exp(-t/T1))$ plot for the extracted parameters.....	21
3.4	Illustration Time at Minimum Slope	24
3.5	Time Derivative of the Log Function of Measured Data in ICCAP	25

LIST OF TABLES

Table		Page
2.1	Summary of Commands Used for Data Retrieval.....	11
3.1	Result from One-Pole Model Extraction.....	21

CHAPTER 1
INTRODUCTION
1.1 Background

Work has been done over the years to study the thermal behavior of Bipolar Junction Transistors (BJTs), to make accurate models of the thermal effects in circuits as designed. BJTs are now manufactured at increasingly smaller dimensions, thereby making the thermal behavior of greater concern to the analog circuit designer. Thermal effects can be classified into three categories [1]. First is thermal coupling in which two devices on the same chip affect one another so that thermal changes in one device affect the thermal properties of another. It is known that the temperature of the device affects its electrical characteristics. Another thermal effect which is more commonly discussed in manufacturing is the package thermal effects and this is based on the properties of the materials used to house the die on which the circuit is built. This effect raises the temperature of the whole chip. Finally, the thermal effect which is related to the work in this thesis is Self Heating. Self Heating, as the name implies, is the thermal effect whereby the operating device raises its temperature and may alter its operating behavior due to its own power dissipation [1]. Current trends in circuit design using BJTs, which include smaller device geometries, make self heating of significant concern to the circuit designer. In some cases, smaller transistor sizes increase the emitter current density and leads to higher operating temperatures. Another trend is device isolation which together with device scaling increase the thermal resistance and also leads to increased temperatures during operation.

Dielectrically Isolated Bipolar Junction Transistors (DiBJTs) are an example of devices using device isolations. Devices are dielectrically isolated by using trench isolation and silicon-on-insulator (SOI) technologies, in which trenches are dug around the device and filled with an insulating oxide. The advantages of using DiBJTs in circuit design are reduced parasitic capacitances, low leakage currents and latch-up free behavior to name a few. However, with

these promising advantages comes a major drawback: the dielectric used for isolation has poor thermal conductivity. The oxide of the silicon used in trench isolation has a fraction of the thermal conductivity of the pure silicon [2].

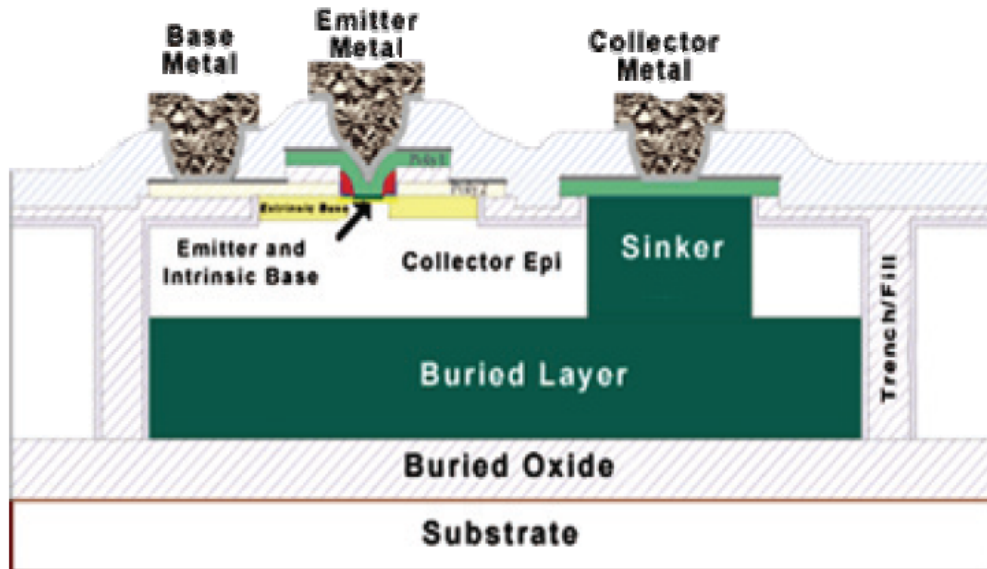


Figure 1.1 Cross Section of VIP 10 Process BJT [3]

The VIP10 process, by National Semiconductor is one of the processes in the VIP (Vertical Integrated PNP) family and is the process used in the thermal studies to which the work done in this thesis is geared. Figure 1.1 [3] shows the cross-section of the Di-electrically Isolated BJT. Trenches around the device and filled with an insulating oxide, which in this case is SiO_2 . The device is surrounded on the side and the bottom by high thermal resistivity material.

Ultimately, these devices are used in high performance circuits such as High-frequency Operational Amplifiers and Current Feedback Operational Amplifiers (CFOA). Studies have been done to show that thermal effects directly affect the performance of the amplifiers [2]. In precision applications, performance is drastically degraded because self-heating causes the device parameters to vary. This is especially when the thermal attributes of the transistor are not modeled adequately and provided to the circuit designer to incorporate in the design

process. A circuit parameter of a CFOA that is known to suffer the effects of self-heating is input offset voltage.

It is necessary, in high performance analog circuit design which uses these types of transistors to be able to predict the effect of thermal behavior of the transistor on overall circuit designs. There are a number of apparent options to this. Two of the options are 1) gain access to the model equations in the simulator and alter them based on the physical geometry of each device, [2] or 2) use electro-thermal coupled simulation. In the first option the electrical model will be changed to match the geometry of the device based on an empirical model. In the second option, thermal simulations will be run based on the dimensions of the device

Both of these options are not very popular because in the first scenario, the equations will be tied to the layout and operating conditions of the circuit and would have to be changed as either is altered. In the second case, that solution implies an overhaul of source code of the simulator (to include thermal simulations) or a change in the design of the simulator to accommodate the solving of three dimensional heat equations along with the electrical equations and coupling the two. The more popular alternative is to use the VBIC (Vertical Bipolar Inter Company) model which will be described in the following section.

1.2 VBIC Model

Three parameters that are included in the VBIC model are thermal capacitance, thermal resistance and self-heating. A brief description of the VBIC model is given followed by specific details of the advantages of this model.

1.2.1 Brief History and Description

The Gummel-Poon (GP) model had been the industry standard in circuit simulation of BJTs for 20 years. The VBIC model was developed to replace the Gummel Poon model because as time went on and technology of BJTs progressed, there were many deficiencies in the model. Although there were improved BJT models presented during the 20 years, none

became an industry standard. The VBIC model was a collaborative effort by professionals in the integrated circuit and CAD (Computer Aided Design) industries. There were many benefits of the VBIC model over the GP model and they are listed below [4]. 1. Improved Early effect modeling, 2. Quasi-saturation modeling, 3. Parasitic substrate Transistor modeling, 4. Parasitic fixed (oxide) capacitance modeling, 5. Avalanche multiplication modeling, 6. Improved temperature dependence modeling, 7. Decoupling of base and collector currents, **8. Electro-thermal (self heating) modeling**, As shown in bold the addition of the self heating was one of the brand new additions to this model.

1.2.2 Thermal Resistance

Generally, thermal resistance is defined as the temperature difference between two closed isothermal surfaces divided by the heat flow between them. The mathematical expression is below

$$R_{th} = \frac{T_j - T_a}{P} \quad \text{Kelvin /Watt (K / W)} \quad (1.1)$$

where T_a = ambient temperature, T_j = junction temperature and P = heat flow rate

Thermal resistance in transistors depends on geometry, because a change in the geometry changes the heat flux network. The thermal circuit is necessary to study the heat conduction of a device's hot core to its outer surface. Thermal resistivity is defined as the ratio of the thermal gradient to the heat flux density for one dimensional heat conduction. It is a property of the material that describes its thermal conduction and has the units $KW^{-1}cm^{-1}$ [3].

1.2.3 Thermal Capacitance

Thermal capacitance is the measure of how much heat energy can be stored in a device. It is expressed in units of J/K. It is also known as *volumetric heat capacity*.

It is expressed mathematically below;

$$C_{th} = \rho_s \times C_s \times w \times l \times d \text{ (J/K)} \quad (1.2)$$

where, ρ_s = specific weight of the substance ($\text{kg}/\mu\text{m}^3$), C_s = specific heat capacity of the substance ($\text{J kg}^{-1} \text{K}^{-1}$) and w, l and d = width, length and thickness of the substance in μm .

1.2.4 Self Heating (VBIC parameter)

Self heating can be toggled on or off by setting the self heating switch (model parameter) in the VBIC model to 1 or 0. When set to 1 the self-heating effects are taken into account during simulation, and when set to 0 self-heating effects are not taken into account.

1.2.5 The Thermal Model

Figure 1.2 shows the thermal resistance and thermal capacitance network in the complete VBIC model. A fuller description of the model is included in the appendix. The thermal network is the sub-circuit which models the interaction between electrical characteristics and the thermal effects [2]. A rise in temperature in the device is specified by ΔT , which is defined as the product of power dissipated and thermal impedance,

$$\Delta T = Z_{th}' \cdot P \text{ (K)} \quad (1.3)$$

where, P = power dissipated (W) and Z_{th}' = thermal impedance (K / W)

With the above initial equation, the following shows the expression of the elements included in the thermal sub-circuit.

$$I_{th} = P \cdot 1 \text{ A/Watt} \quad (1.4)$$

$$Z_{th} = Z_{th}' \cdot 1 \text{ Ohm W/K}, \quad (1.5)$$

where Z_{th} is the electrical analogy of the thermal impedance. From this, it is easy to see that the voltage at node t (or the 'dt' in the VBIC Model) is numerically equal to the local temperature rise. This is used to calculate the instantaneous electrical characteristics of the device.

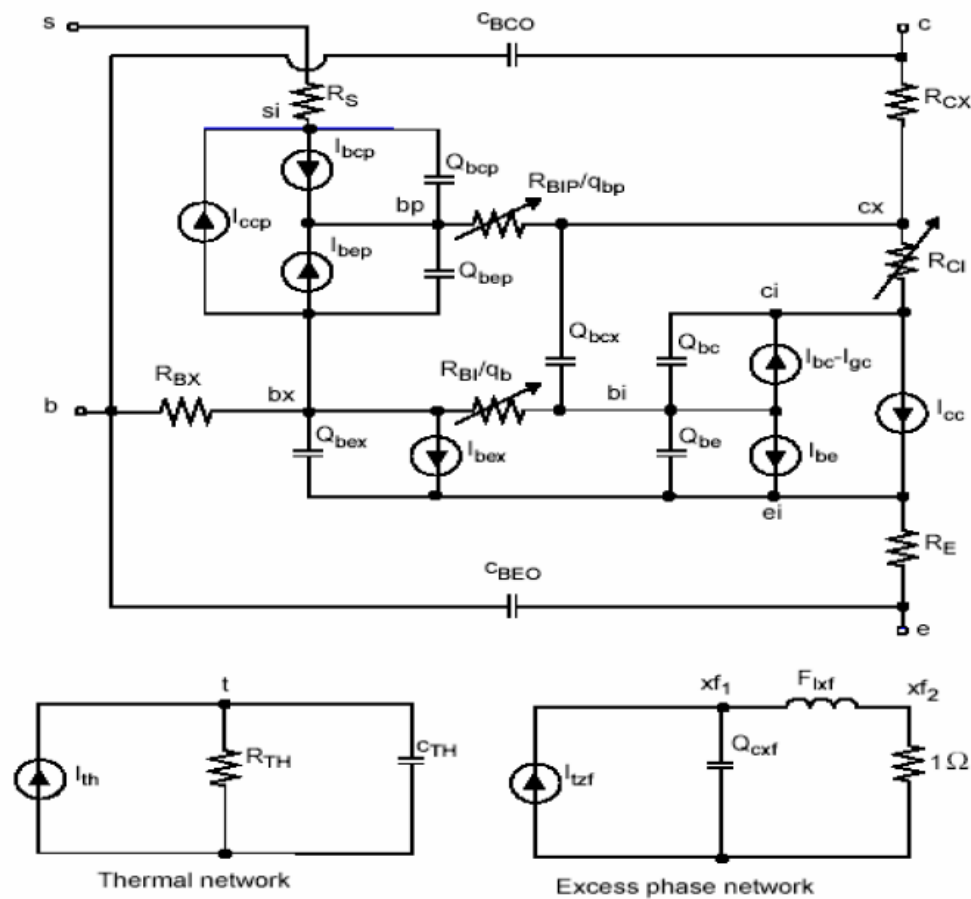


Figure 1.2 Complete thermal network of the VBIC model

1.2.6 Use of the VBIC Model

Figure 1.3 shows the VBIC model symbol as used in Cadence. Thermal nodes related to the thermal network are included in the VBIC model, and when the device is heated, the other device parameter values change. The VBIC model allows the simulator to take into account these temperature dependent parameters and provides the accurate values in calculation as the device temperature changes during operation. For the thermal studies in simulations environment the above symbols are used. The B, C and E represent the base, emitter and collector and are typical in a standard BJT symbol used for design. The dt and tl

nodes however are not so common, and the voltage at the dt node represents a rise in temperature of the device above local temperature [3]. The voltage-temperature correspondence is 1V representing 1 degree rise in temperature above ambient temperature.

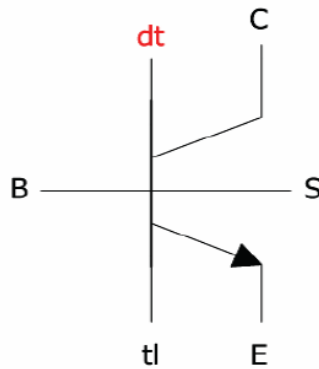


Figure 1.3 Symbol of Transistor with Thermal Nodes

1.3 Thermal Modeling and Optimization

The VBIC model has been defined and following, it is necessary for an actual model to be created to be used for real devices. Once devices are fabricated, it is necessary to develop accurate models by optimizing the predicted values for the model parameters. For the thermal model parameters, responses to input voltage changes are interpreted and fitted to predict simplified models and optimized until the parameter in simulation behaves similar to the real device. The measurements done on workbenches are simulated in virtual workbenches and the models are altered until there is a close match.

The work done in this thesis is to look at one of the processes to carry this out which uses a number of tool including ICCAP and optimized the process using the available tools in the ICCAP modeling program.

CHAPTER 2

MEASUREMENT TOOLS INTERFACE AND ICCAP

2.1 Introduction

Device parameter extraction and optimization procedures normally require many steps and can be time consuming when done repetitively. At times, measurements must be repeated because of data corruption during one of the many steps. Automation of steps such as reading the measured data directly into ICCAP saves time. The automation details are addressed in this chapter.

2.2 Bench Set-up

Figure 2.1 below is the circuit for measuring the thermal properties of the actual device. What is required is to assess the response of the transistor at the collector due to a step input change at the base. The resulting data from the response contains effects from the thermal resistance and capacitance, which will be extracted using mathematical tools in ICCAP. The details of this will be discussed in the subsequent chapter. Described in this chapter is the process of placing measured data directly into ICCAP where the available program tools are used to work on the data. The physical circuit shown in Fig 2.1 consists of a printed circuit board with the transistor and resistor directly soldered onto the board. The power supply and signal generator were connected to the circuit. In the following subsection, the procedure by which the bench setup is interfaced directly with ICCAP is shown including the necessary program commands. This interface is helpful in by-passing unnecessary steps of manual data manipulation.

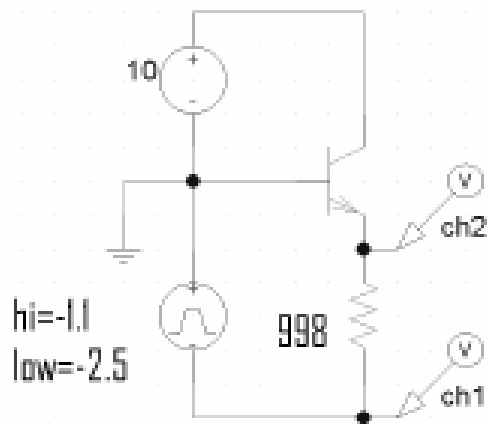


Figure 2.1 Circuit for Model Measurements

2.3 Hardware Interfacing

The key piece of hardware for the interface is the oscilloscope on the physical bench. This is because it is the equipment that takes the time-domain measurements which are necessary for observing the time dependent thermal changes. The necessity of the work described below comes from the lack in ICCAP of a comprehensive set of hardware drivers (code that controls of measurement equipment.) The drivers available are mainly higher-end models of measurement equipment. Also, the drivers which are available do not work for other models of the same brand. One approach to this situation would be to develop a complete driver with a full set of command features. This would require extensive ICCAP expert programming training. It would also go beyond the scope of the thesis and be time consuming. Another approach, which is used, is to take advantage of the available GPIB interface commands available to ICCAP users and create application specific macros. This will yield a macro set in ICCAP with lines of code that are application specific. The macros handle the interfacing with the bench set-up, retrieve the data, manipulate the data with any necessary mathematical functions, and modify the simulation setup (where circuits for simulation are

defined) in ICCAP. Then the measured data is imported into the setup where it can be easily compared with the simulated data. The comparison is done to optimize the models generated in the simulated environment.

2.4 Data Retrieval

The available and common interface for applications such as this is the GPIB protocol which was mentioned previously. This is available on the Unix-based client (computer). The details of initializing the oscilloscope and the ICCAP environment are provided in the Appendix. After the ICCAP client (connected to the DSO6014A) has been initialized to receive instructions, the measurement bench is setup and desired the waveform is displayed on the oscilloscope.

Commands for ICCAP are used to initialize and control the oscilloscope. Table 2.1 [5] indicates the necessary tasks, their equivalent commands, inputs and comments on limitations.

A table showing other commands in the waveform retrieval category is included in the APPENDIX and a full description of the functionality of the commands is in the Agilent manual reference.

In writing the macro, most of the command statements can be written as instructions or queries. The commands set the required parameters of the retrieval process that initializes the DSO6014A oscilloscope. The queries are used to retrieve information from the oscilloscope. They can also be used to check if the oscilloscope was properly initialized, to retrieve the actual data and get information about the data.

Table 2.1 Summary of Commands Used for Data Retrieval

Task	Agilent Command	Input	Comment
Specify Channel	WAVE:SOUR	CHANX*	CHANX means CHAN1-CHAN4 *DSC6014A is 4 channel scope.
Resolution of Read	WAVE:POIN	100 250 500 1000	This sets the number of data points to be retrieved from the scope
Format of Data	WAVE:FORM	ASCII	This converts the data from bytes to ASCII readable real y-axis values, it avoids the need to convert data to ASCII readable values manually
Retreive Data	WAVE:DATA	(Query)	This sends query to the scope based on the settings to retrieve the data that is being displayed on the screen.

To appreciate the usefulness of the foregoing, the following is an outline of the legacy procedure. Once the physical bench is setup and the oscilloscope is displaying the waveform of the response of the circuit, the waveform has to be adjusted as desired. Then, the data is transferred to an external storage media. The media is connected to a PC and the data is loaded into data processing software to be formatted to ICCAP specifications. It is then transferred into a text file for more formatting to be placed in the circuit simulation setup. While on the PC, a check is done of the data. Some preliminary measurements are taken with a spreadsheet application (in this case Microsoft EXCEL) by plotting the data to check the fidelity of the data. Finally the data can be transferred to the ICCAP client computer via an FTP (file transfer protocol) protocol. This has to be done for each measurement set taken and for changes in the setup. Every time a measurement needs to be repeated, the whole process

needs to be repeated. This could be daunting and time consuming. Generally, an effort is made in any procedure made to avoid repetitive steps carried out by engineers. This gives time for actual analysis and innovation while letting the repetitive work to be done by the computer as much as possible. In the procedure described above, it is immediately apparent that a script could be written in Microsoft Excel to do the formatting and quick measurements however avoiding these steps as much as possible and using the mathematical tools in ICCAP is clearly more practical and economical. Transferring data directly into ICCAP removes the need of formatting because ICCAP is able to place the data automatically in the write format with the help of the macro,

One scenario where this method of transferring the data is when there is a need for data to be analyzed from different perspectives. For example, with a measurement taken over the time span of 10 ms, it may be necessary to view the whole time period with the maximum amount of resolution. Thereafter, there could be a need to parse the data set to a few micro seconds with high resolution (an actual need in calculating thermal resistance and capacitance from a response waveform). In this situation, there will be tremendous time savings by having direct ICCAP communication with the oscilloscope, in contrast to the legacy procedure described above.

2.4.1 Importing Data into Simulation Environment

Figure 2.2 shows an example of the replication of a waveform produced in the waveform generator previewed on the oscilloscope. Figure 2.3 shows the same data extracted from the scope and plotted on the graph plot option in ICCAP. It shows that the imported data graphing is satisfactory for checking the validity of data and making visual observations.

Once the data has been transferred from the oscilloscope to ICCAP, there is a need for it to be placed in the simulation environment within ICCAP. Although simple in concept this only works if the time base in the environment matches the measured data. Otherwise the data has to be truncated before it is imported into the simulation environment. For different

measurements, different time bases may be needed and it would not be desirable to alter the test environment continually when making multiple types of measurements. The best option is to create multiple environments and altering the time base in each one based on the time period in which the data was acquired.

When the data is acquired, all that is acquired is the y-axis real values of the waveform. To make the values meaningful they have to be matched with an x-axis which is the time base. The creation of this time base requires the knowledge of the $\mu\text{s}/\text{div}$ setting on the oscilloscope when the data is acquired. A routine was implemented to acquire this information and automatically create this time base. This is an example of a way in which this procedure avoids unnecessary steps and harnesses the capability of the computing power available.

Figures 2.2 and 2.3 below show the plot of a sine wave on the oscilloscope and the representation of the same data in an ICCAP plot after extraction from the oscilloscope. The sine wave is used as a test pattern. The voltage of the 20 KHz signal was 100 mV and both figures show this accurately. The ability to replicate the waveforms shown on the oscilloscope is of great utility in ensuring the data that is transferred is the same as that which is measured before any further analysis is done.

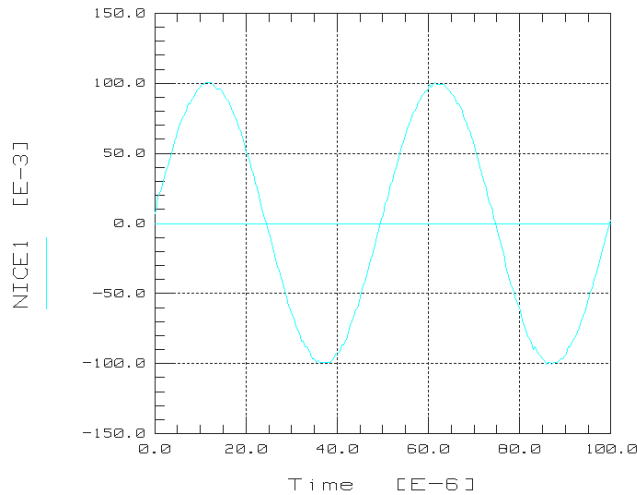


Figure 2.2 Test Pattern Sine Wave in ICCAP plot.

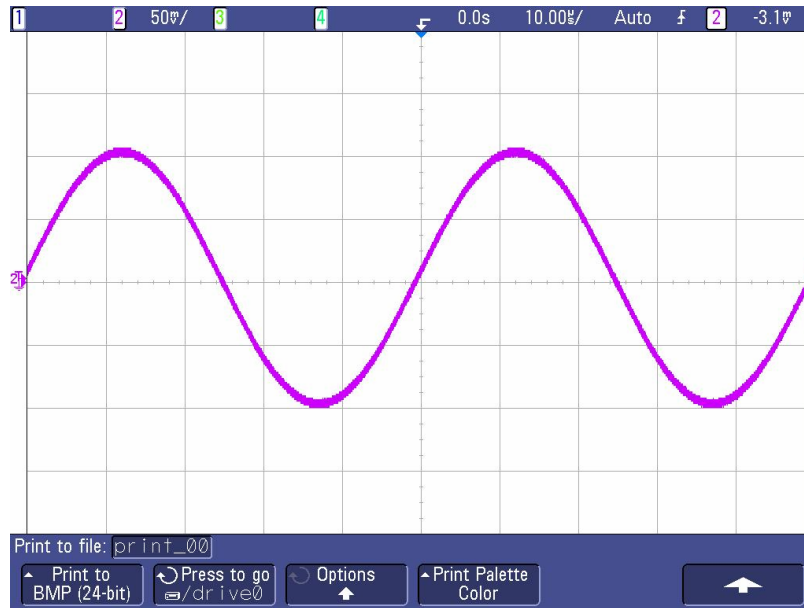


Figure 2.3 Test Pattern Sine Wave on the Agilent DSO6014A

2.5 ICCAP

In this section a brief overview will be given of the ICCAP tool and the specific parts that relate to the work done in this thesis.

2.5.1 Overview

The Integrated Circuit Characterization and Analysis Program (IC-CAP) is a parameter extraction and device modeling program that provides characterization and analysis for semiconductor modeling. It is owned and supported by Agilent Technologies Inc. With ICCAP, engineers are given software that can perform instrument control, data acquisition, graphical analysis, simulation, optimization, and statistical analysis all in one package. ICCAP has an Open Software Architecture (OSA) and thus it allows the engineers to develop their own extraction or modeling methods. An open measurement interface is also employed in ICCAP,

and this allows the user to write measurement drivers to control instruments with the Parameter Extraction Language (PEL) unique to ICCAP. These two properties of the program are used in the work done in this thesis. The measurement drivers were discussed in this chapter and the extraction and modeling methods will be discussed in the following chapter.

2.5.2 Components of ICCAP

Figure 2.4 shows the components of ICCAP. The user environment is central and it is there that most of the work is done by the user. There are other modules that could be attached to the environment to meet specific engineering needs. These are the instrument drivers, Statistic Packages and Extraction and Analysis Modules (SPEAM). Due to the nature of the work done in modeling, there are a lot of different methods that do not have modules. This is where the open system architecture of ICCAP is very useful because it allows the user to write his own module and perhaps publish it for the use of others who may encounter a similar engineering problem. The main components of ICCAP used in the work presented in this thesis are the user environment with the PEL and the analysis module.

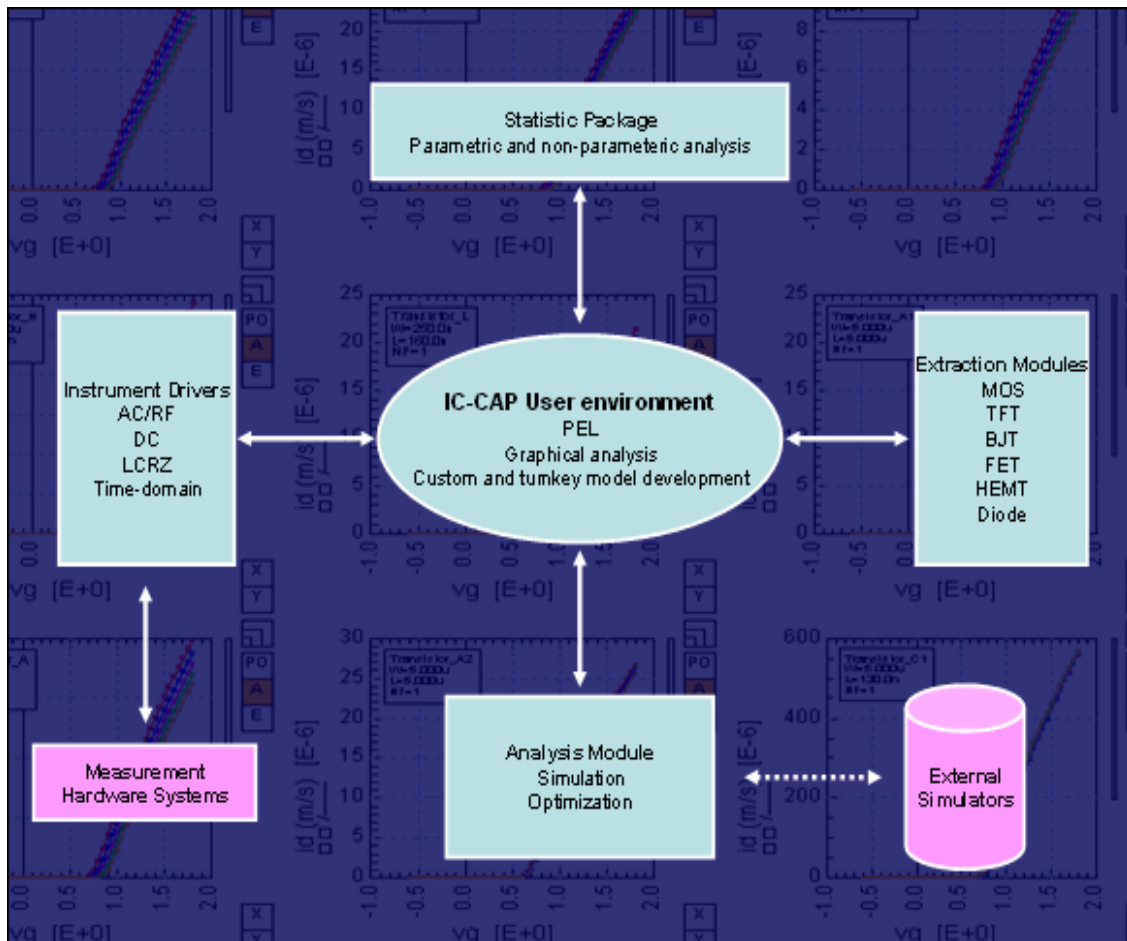


Figure 2.4 Components of ICCAP [7]

2.3.3 ICCAP User Environment

Figure 2.5 shows the ICCAP user environment. For most uses, the simulation circuit is setup in the tab named “Circuit” and the different configurations of tests are run in the “DUT/Setup” tab. This is where most of the work is done in writing macros or optimizing and extracting models. As shown in the snapshot, the function “Program” is used to generate user commands that carry out the instrument interfacing or model optimization.

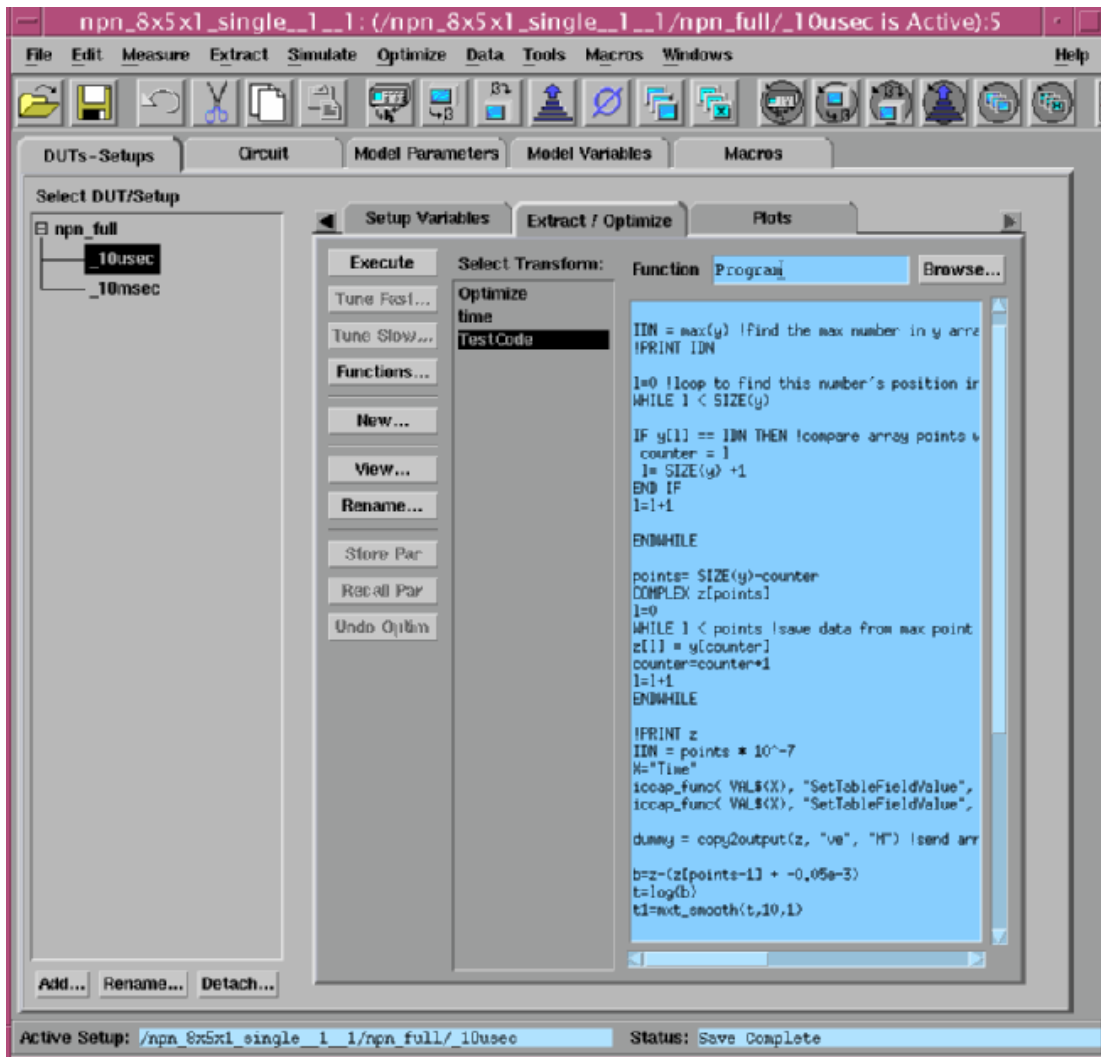


Figure 2.5 ICCAP User Environment Snapshot

CHAPTER 3

ONE- AND TWO-POLE MODELING IN ICCAP

After the connection of the equipment and retrieval of the data, operations are performed to extract the thermal time constant from the time domain response of the device. The ICCAP program must be set to correspond to the data received from the oscilloscope. That is, an x axis has to be applied to the data. The display of the oscilloscope is queried soon after the data is received for the time/div setting. The time/div value is used to generate the time base for ICCAP. This will ensure an accurate representation of the data for analysis.

The first challenge is to extract the portion of the data which contains the time constant of the decay as shown in Figure 3.1. For this, two options were considered. The first possibility was to use the trigger point of the oscilloscope to see where the switching point is. The second possibility is to search to find the area around the peak point and parse the data. The first method was chosen.

3.1 One Pole Extraction

For this section [8] the voltage response of the circuit in Figure 3.1 which contains the thermal information is assumed to have a one pole model thermal response. This one pole model is immediately extracted upon execution of the ICCAP macro and a comparison is made between the simulation and the data. The time domain voltage response is given by

$$V(t) = Ae^{-\frac{t}{T}} \quad (3.1)$$

where t is time and T is the time constant, which is needed to calculate the thermal capacitance. The steps below are used to relate the graph in Figure 3.1 to the thermal time constant so that the value of T can be extracted from the measured graph.

Since the voltage response is of the exponential form, it is very useful to take the natural log of the data points to get an equation that directly relates to the time constant

$$\ln(V(t)) = \ln(A) + \left(\frac{-t}{T}\right) \quad (3.2)$$

As shown in equation (3.2), what results is the inverse of T as a multiple of the time variable plus a constant. Another operation is the time derivative of the voltage which will give a clean equation yielding a value for T

$$\frac{d(\ln(V))}{dt} = -\frac{1}{T} \quad (3.3)$$

$$T = -\left[\frac{d(\ln(V))}{dt}\right]^{-1} \quad (3.4)$$

Here, it is shown that the following operations could be applied to the imported data and to find T. Once T is calculated, the thermal capacitance can be calculated using the value of the thermal resistance using the equation below.

$$\tau = C_{th} R_{th} \quad (3.5)$$

A value for R_{th} is found calculated from other experiments [3]

However, before the operations are able to be performed on the data set, there is a need to normalize the data to the settling value. The natural logarithmic function in (3.2) has a limit that goes to zero as time goes to infinity. In this data set, as time goes to infinity the limit goes to a non-zero number. For the preceding operations to be applied to the data set, there is a need to predict the non-zero limit as time goes to infinity. This value is used to normalize the data while maintaining the properties of the log function with no values going into the negative y-plane.

It is good to mention that a more accurate model of the thermal capacitance is a 5-pole model. The one pole simplification, although useful in some ways, is not accurate enough to be

directly implemented as a model parameter in design work. The value that results from the one pole model could be used as a starting value for more accurate modeling procedures.

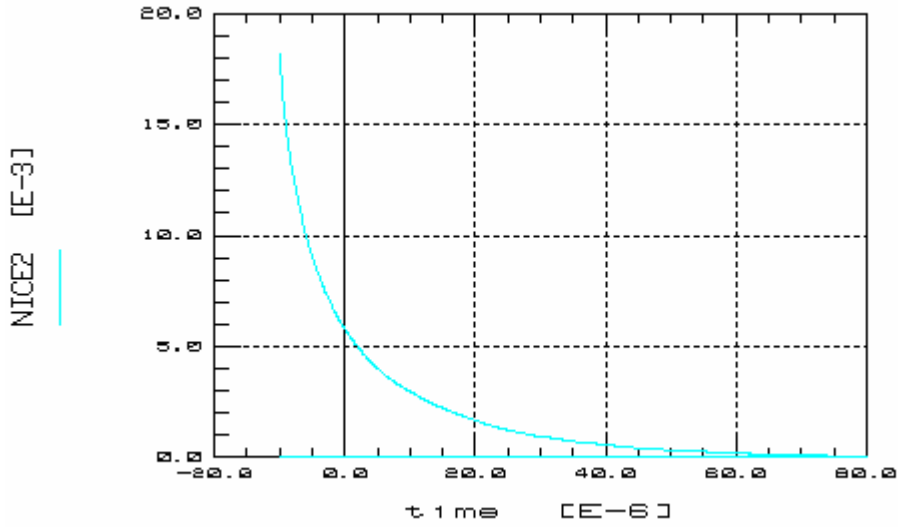


Figure 3.1 Normalized Exponential Decay

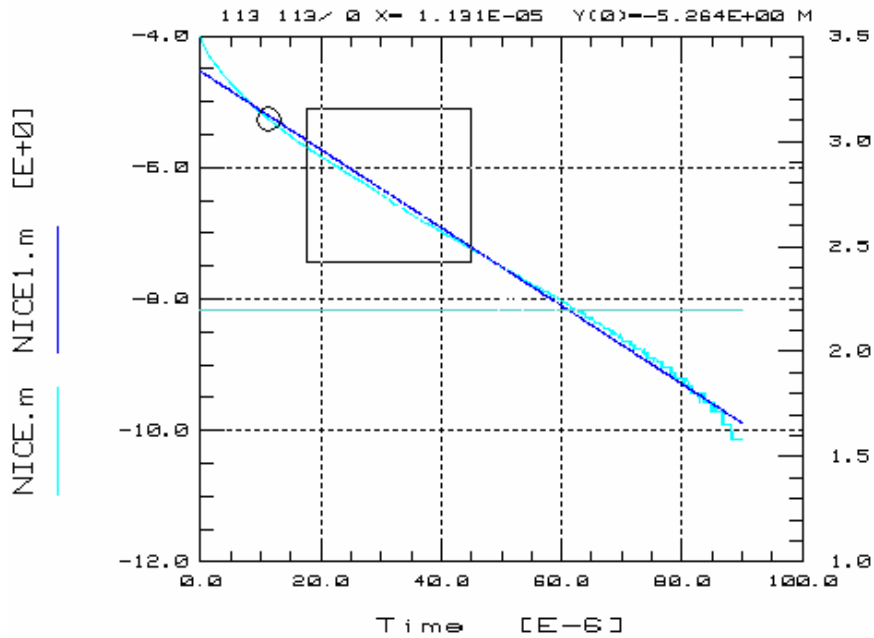


Figure 3.2 Line fit to find 1-Pole Time Constant

Figures 3.1 and 3.2 show the results of the procedure described above as implemented in the ICCAP Parameter Extraction Language (PEL) Macro described in Chapter 2. The first step needing to be done to the data is to crop only the portion of the data with the useful information. This will be the time period in which the device is responding to the input change and the point from which the oscilloscope is set to trigger.

Table 3.1 Result from One-Pole Model Extraction

	1-Pole Extraction Results	Optimized Results
Tau (T_1)	16.91us	6.62us
Rth	853.30	853.30
Cth	19.82n	7.76n
A_1	17.08m	

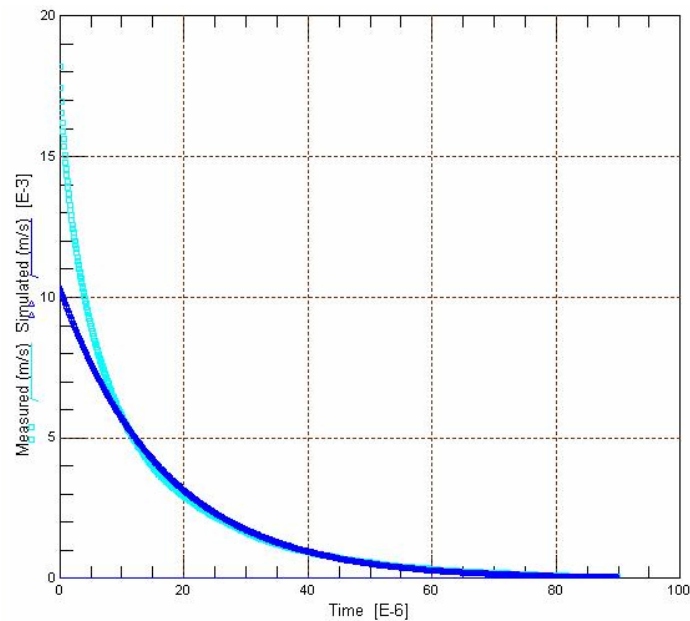


Figure 3.3 Plot of measured data and the ($A_1 \cdot \exp(-t/T_1)$) plot for the extracted parameters

3.2 Two-Pole Model

A more precise model than the one pole assumption for the thermal response would be the two-pole model [7]. This assumes that the time response of the transistor includes two time constants. Equations (3.6) and (3.7) below show the response of this type in mathematical form. This includes four unknowns with A_1 , A_2 , T_1 and T_2 . The important parameters are the time constants T which are essential for determining the thermal property, thermal capacitance of the transistor.

$$y(t) = A_1 e^{-\frac{t}{T_1}} + A_2 e^{-\frac{t}{T_2}} \quad (3.6)$$

$$y(0) = A_1 + A_2 \quad (3.7)$$

Similar to the one pole model, the time derivative of the response is found

$$\frac{dy}{dt} = -\frac{A_1}{T_1} e^{-\frac{t}{T_1}} - \frac{A_2}{T_2} e^{-\frac{t}{T_2}} \quad (3.8)$$

The assumption is made that $T_1 < T_2$. If T_1 is much less than the time value, t , the dominant time constant will be the larger T_2 . This fact is taken advantage of in the following equation for the time derivative of time response.

$$\lim_{t \gg 5T_1} \frac{dy}{dt} = -\frac{A_2}{T_2} e^{-\frac{t}{T_2}} \quad (3.9)$$

The same is true also for the time response

$$\lim_{t \gg 5T_1} y = -A_2 e^{-\frac{t}{T_2}} \quad (3.10)$$

Since equations (3.9) and (3.10) are very similar in appearance and differ only in one term, the time constant, it is possible to perform an operation in which the difference of the two equations will be left.

With the assumption still remaining that, $t \gg 5T_1$

$$\frac{dy}{dt} \frac{1}{y} = \frac{d(\ln y)}{dt} = \frac{-\frac{A_2}{T_2} e^{-\frac{t}{T_2}}}{A_2 e^{-\frac{t}{T_2}}} = -\frac{1}{T_2} \quad (3.11)$$

By dividing the time response by the time derivative of the time response, the time derivative of the natural log of the time response is evaluated. This yields the inverse of the negative of the second time constant.

The time constants are the sought quantities so to evaluate the time constant, T_2 , the following is used.

$$T_2 = -\left(\frac{d(\ln y)}{dt}\right)^{-1} \quad (3.12)$$

To find the values of A_2 , t_{\min} will be defined as the value of time, t , at which the minimum value of the derivative of $\ln(y)$ results. Fig. 3.4 illustrates this. From Eqn. 3.10

$$\lim_{t \gg 5T_1} y(t_{\min}) = -A_2 e^{-\frac{t_{\min}}{T_2}} \quad (3.13)$$

Therefore,

$$A_2 = y(t_{\min}) e^{\frac{t_{\min}}{T_2}} \quad (3.14)$$

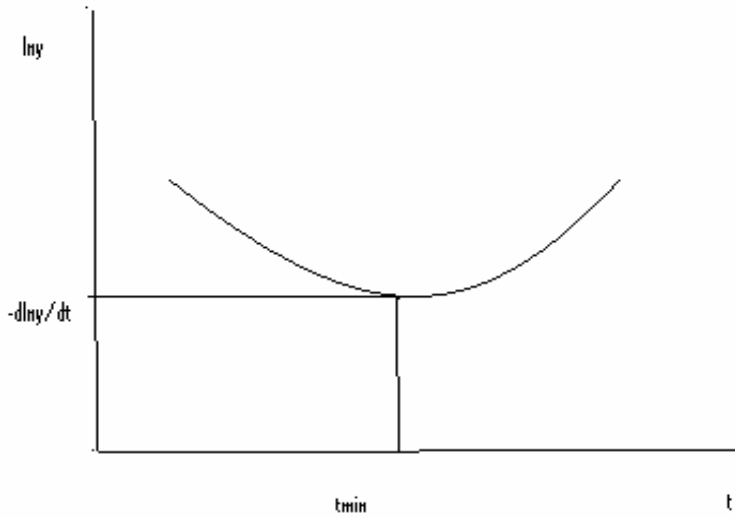


Figure 3.4 Illustration Time at Minimum Slope

Since t_{\min} and T_2 are known values from the preceding definitions, Eqn. 3.14 yields the value of A_2 . From Eqns. 3.12 and 3.14, the values of T_2 and A_2 are obtained. With the relationships defined in (3.7) and (3.8), T_1 and A_1 could be calculated.

$$A_1 = y(0) - A_2 \quad (3.15)$$

and

$$T_1 = \frac{-A_1}{\left. \frac{dy}{dt} \right|_0 + \frac{A_2}{T_2}} \quad (3.16)$$

In the next subsection the preceding equations are applied to the measured data in ICCAP. The averaging of the data was done with the IC-CAP smoothing function. This is due to the nature of the data that was measured.

3.3 Results from ICCAP Implementation

In Chapter 2 and Section 3.1, the voltage response that contains the thermal transient information has been extracted. Similar to the procedure in Section 3.1, the natural log of the truncated data set was found and is show in Figure 3.5.

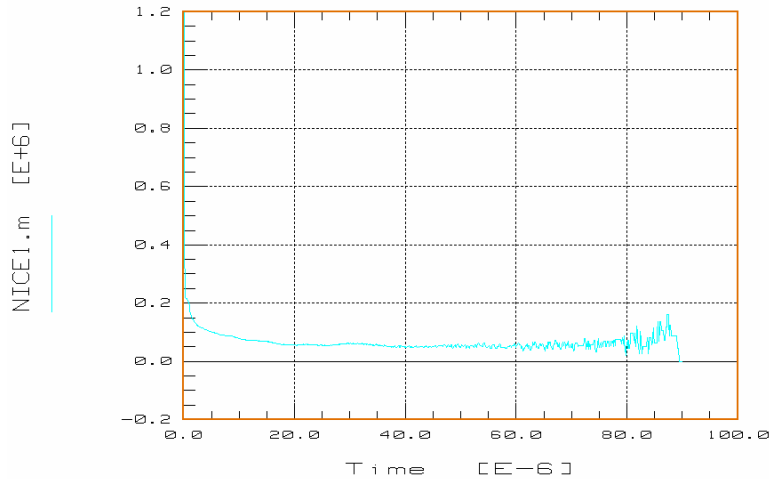


Figure 3.5 Time Derivative of the Log Function of Measured Data in ICCAP

It is noticeable that the result in Figure 3.5 is very noisy due to the nature of the physical bench setup (mainly the oscilloscope) and a smoothing function had to be applied to the data to make it readable. In any case, to extract the time constant from this figure needs an exponential graph extraction tool that could accurately model an exponential graph. This is because the minimum value of the data in Figure 3.5 is difficult to determine. This could be addressed this using the Levenberg–Marquardt routine available in ICCAP.

APPENDIX A

READING DATA FROM AGILENT DSO6014A

THINGS TO KNOW

- There is no existing driver for the Agilent Scope DS6014A in ICCAP
- Scopes with Available Drivers are in pg. 89-107 of ICCAP Reference manual
- Implementation is a Macro that is specific to the Agilent DS6014A Oscilloscope for retrieving displayed waveform on channel and with points specified by the user.
- Communication protocol used is "gpib" (sometimes referred to as "hpib" –could be confusing when writing code)

CONFIGURING THE DSO6014A

- STEPS
 - Connect GPIB cable to the GBIP port on the back of Scope
 - Set the I/O to communicate over the GPIB
 - Press Utility => I/O => Set Controller (to GPIB)
=> Set Configure (to GPIB)
=> Set Address (to XX) e.g. 7
(REMEMBER THE ADDRESS VALUE)

SETTING UP THE PC

- Ensure the GPIB card is installed on the computer
- Ensure the GPIB I/O library is installed on the computer
- Know the Terminal Specific Command line for opening this interface
 - SUN => "/dev/gpib0"
 - HP700=> "hpib"
 - Linux => "gpib0"

ACQUIRING THE DATA

- Open the setup in ICCAP and create a setup for the MACRO by going to the “Macro” tab and creating a new “Program” which will run the code and copy and paste the code in to it (4 sets of Code for each of the Channels on the Scope)
- Take measurements on the scope
- Ensure the number of points in destination output (and input) variables are the number of points being acquired by the program (typically 250| 500|750|1000)
- Execute the program(s)

APPENDIX B

SOURCE CODE FOR DATA ACQUISITION AND MODELING

SOURCE CODE FOR MODEL EXTRACTION

```
y=ve.m !Location where measured data is saved

IDN = max(y) !find the max number in y array
!PRINT IDN

l=0 !loop to find this number's position in the array using counter l
WHILE l < SIZE(y)

IF y[l] == IDN THEN !compare array points with max value and if match
force end loop
counter = l
l= SIZE(y) +1
END IF
l=l+1

ENDWHILE

points= SIZE(y)-counter
COMPLEX z[points]
l=0
WHILE l < points !save data from max point to end of array "y" in a new
array "z"
z[l] = y[counter]
counter=counter+1
l=l+1
ENDWHILE

!PRINT z
IDN = points * 10^-7
X="Time"
iccap_func( VAL$(X), "SetTableFieldValue", "Stop", IDN) ! Set the Time
Variable to have same time base as scope
iccap_func( VAL$(X), "SetTableFieldValue", "# of Points", points)

dummy = copy2output(z, "ve", "M") !send arrays to output

b=z-(z[points-1] + -0.05e-3)
t=log(b)
t1=mxt_smooth(t,10,1) !smoothing funtion
```

SOURCE CODE FOR MODEL EXTRACTION Cont'd

```
p=derivative(Time,t1,1)
!p1=abs(p)
p1=p
mini=min(p1)
!p1=p1-(mini)
!PRINT mini
!p1=log(p1)
!p1=p

!p=fit_line(Time, t, 1)
dummy = copy2output(t, "NICE", "M")
dummy = copy2output(p1, "NICE1", "M")
dummy = copy2output(b, "NICE2", "M")
COMPLEX s[3]
s=linfit(Time, t, 0)
```

SOURCE CODE FOR DATA ACQUISITION [8]

```
!LINPUT "What is the Name of your Time Variable", X

!PRINT X
X="TIMEX"
Y="TIME1"
hpib_num=HPIB_open("/dev/gpib0") ! Open up the GPIB channel
instr_addr=7 ! Specify the Scopes GPIB address
sum=250 !number of points to get from the scope

COMPLEX y[sum] !array to store the entire waveform
COMPLEX z[sum] !array to make tail begin at zero time

status=HPIB_write(hpib_num,instr_addr, "TIM:SCAL?") !Check the
Time/Div on Scope
status=HPIB_read (hpib_num, instr_addr, 100, "A", "IDN") !Store Results

PRINT IDN !Print out Results
IDN = IDN * 10 ! Calculate the Time Base
iccap_func( VAL$(X), "SetTableFieldValue", "Stop", IDN) ! Set the Time
Variable to have same time base as scope

status=HPIB_clear(hpib_num, instr_addr)

status=HPIB_write(hpib_num,instr_addr, "*IDN?\n") ! A check of the Scopes
Identification number
status=HPIB_read (hpib_num, instr_addr, 100, "A", "IDN") !Read and store
in variable IDN

!PRINT IDN

status=HPIB_write(hpib_num,instr_addr, "WAV:SOUR CHAN2") ! Set the
Source Channel for the Scope
status=HPIB_write(hpib_num,instr_addr, "WAV:SOUR?") !Check the Source
Channel for the Scope
status=HPIB_read (hpib_num, instr_addr, 100, "A", "IDN") ! Read back
!PRINT IDN
```


SOURCE CODE FOR DATA ACQUISITION Cont'd

```
status=HPIB_write(hpib_num,instr_addr, "WAV:POIN " &VAL$(sum)) ! Set
the number of points to read from scope
status=HPIB_write(hpib_num,instr_addr, "WAV:POIN?") !Check number of
points to be readback from scope
status=HPIB_read (hpib_num, instr_addr, 100, "A", "IDN") ! Readback
!PRINT IDN
```

```
status=HPIB_write(hpib_num,instr_addr, "WAV:FORM ASCII") !Format of
Readback set to ASCII
status=HPIB_write(hpib_num,instr_addr, "WAV:FORM?")
status=HPIB_read (hpib_num, instr_addr, 100, "A", "IDN")
!PRINT IDN
```

```
status=HPIB_write(hpib_num,instr_addr, "WAV:DATA?") ! Execute to get
data from SCOP
status=HPIB_read (hpib_num, instr_addr, 3262, "A", "IDN") !Sets how much
data to get with the number (3262)
```

```
!PRINT IDN
```

```
i=11 ! Starting point for extractions of data point from string
k=0 !Counter
m=23 !Ending point for extraction of data point from String
WHILE k < SIZE(y)
IDNN = substr$(VAL$(IDN), i, m) !remove point between character point i
and m
IDNN = trim$(VAL$(IDNN)) !remove any space
IDNN = trim$(VAL$(IDNN), ",") !remove and comma
y[k]=IDNN !save to array
```

```
i=i+13 !increment to next datapoint start
m=m+13 !increment to next datapoint end
k=k+1 !increment counter
!PRINT IDNN
```

SOURCE CODE FOR DATA ACQUISITION Cont'd

```
END WHILE

IDN = max(y) !find the max number in y array
PRINT IDN

l=0 !loop to find this number's position in the array using counter l
WHILE l < SIZE(y)

IF y[l] == IDN THEN !compare array points with max value and if match
force end loop
  counter = l
  l= SIZE(y) +1
END IF
l=l+1

ENDWHILE

PRINT "COUNTER:"
PRINT counter
PRINT " "
PRINT "VALUE AT COUNTER"
PRINT y[counter]
PRINT " "
!PRINT l
points= SIZE(y)-counter !number of point between max number and end of
array

!PRINT points

l=0
WHILE l < points !save data from max point to end of array "y" in a new
array "z"
z[l] = y[counter]
counter=counter+1
l=l+1
ENDWHILE
PRINT y[125]
PRINT z[120]
PRINT z[0]
```

SOURCE CODE FOR DATA ACQUISITION Cont'd

```
dummy = copy2output(y, "NICE1", "M") !send arrays to output
dummy = copy2output(z, "NICE", "M")
p=derivative(Time,t1,1)
!p1=abs(p)
p1=p
mini=min(p1)
!p1=p1-(mini)
!PRINT mini
!p1=log(p1)
!p1=p

!p=fit_line(Time, t, 1)
dummy = copy2output(t, "NICE", "M")
dummy = copy2output(p1, "NICE1", "M")
dummy = copy2output(b, "NICE2", "M")
COMPLEX s[3]
s=linfit(Time, t, 0)
```

REFERENCES

- [1] Sang-Gug Lee, "A review of self-heating effects on bipolar circuit," in *Proc. Southcon/94 Conf.*, March 1994, pp. 138 – 142.
- [2] Anurag Lakhani, "Device Characterization and Experimental Determination of Thermal Resistance for Dielectrically Isolated BJTs," MS Thesis, EE Department, UTA, May 2003.
- [3] MD Mahbub Hossain, "Thermal Effects on Analog Integrated Circuit Design." PhD Dissertation, EE Department, UTA, May 2007.
- [4] Colin C. McAndrew et al, "VBIC95; The Vertical Bipolar Inter-Company Model," IEEE Journal of Solid-State Circuits, October 1996
- [4] ICCAP 2002 Reference, Agilent Technologies, June 2002.
- [5] Agilent 6000 Series Oscilloscopes Programmer's Reference, Agilent Technologies, July 23, 2007
- [6] "IC-CAP Parameter Extraction and Device Modeling Software" Retrieved July 7, 2008 from Agilent Technologies Web site:
http://eesof.tm.agilent.com/products/iccap_main.html
- [7] Private communication with Dr. R. L. Carter, Electrical Engineering Department, UTA, December 2007.
- [8] IC-CAP Training: 31 suppl1 PEL Quick Guide, Agilent Technologies, Dec 12, 2002

BIOGRAPHICAL INFORMATION

Obiorah Oji is a native of Nigeria, West Africa. He completed his BSEE from UTA in 2005. He desires to work in the semiconductor industry as a Product Development Engineer. His hobbies are reading and working out.