TESTING METHODS FOR CURRENT-MODE INTEGRATED CIRCUITS UNDER THE VERY LOW COST TESTER (VLCT) AND LABVIEW® TEST PLATFORM

By

SERGIO COUTTOLENC VALDÉS

A Project Report submitted in partial fulfillment of the requirements for the degree of

MASTER IN ENGINEERING in ELECTRIC ENGINEERING

UNIVERSITY OF PUERTO RICO
MAYAGUEZ CAMPUS

July 2004

Approved by:

[redacted]
Manuel Jimenez Cedeño, Ph. D.
Graduate Committee Member

Julio 14/2004
Date

[redacted]
Gerson Beauchamp, Ph. D.
Member, Graduate Committee

July 14, 2004
Date

[redacted]
Rogelio Palomera Garcia, Ph. D.
Graduate Committee Advisor

July 14th, 2004
Date

[redacted]
Marco A. Arocha Ordoñez, Ph. D.
Representative of Graduate Studies

july/14th/2004
Date

[redacted]
Jose Fernando Vega Riveros, Ph. D.
Department Associate Director

July 14, 2004
Date

ABSTRACT

Testing Methods for Current-Mode Integrated Circuits under the Very Low Cost Tester (VLCT) and LabView® Test Platform

By

Sergio Couttolenc Valdés

Test software is widely used to perform electrical testing regardless of the hardware platform. From the entire hardware platform, the Very Low Cost Testers (VLCT) are a practical low cost options for manufacturing and education purposes, because they are custom-made. From the Software platform, LabView® (National Instruments Company) is presently one of most used software for testing and measurements in the world.

In spite of the existence of test algorithms and methods which are inescapable steps of the electrical testing art, there is a lack of algorithms focused in testing current-mode Circuits using VLCT and LabView. Current-mode circuits are becoming the more important players in the design of low power – low voltage circuits, thereby making necessary to have such algorithms available.

The Purpose of this Project is to generate algorithms and methods using the VLCT donated by Texas Instruments (TI) and LabView® in order to perform electrical testing of Current-Mode Circuits.

The most useful formula to solve complex problems is by applying the proverb: "divide and conquer". Following this philosophy, a basic and small circuit that holds all the principles of current-mode circuits is the best option to develop algorithms and programs. In consequence, the current project framework comprises only the basic current-mirrors, because with short algorithms and programs is possible to create larger programs for complex circuits. Only DC stimuli will be used. The AC stimuli are out of the scope of this project, it should be proposed for future research.

The contributions of this project are:

- Procedures and algorithms for testing basic current-mirrors using VLCT.
- Integration between VLCT and LabView using a virtual instrument.
- Test programs in Pascal programming language for VLCT that can be used to create more complex programs.
- Example of Pascal programming language for D.O.S. that can be used to compile and create an application or executable file and that can be exported to VLCT Pascal programming platform with minor changes. These changes will be exposed later in this text.

COMPENDIO

Testing Methods for Current-Mode Integrated Circuits under the Very Low Cost Tester and LabView® Test Platform

Por

Sergio Couttolenc Valdés

Los programas de prueba son ampliamente utilizados, sin importar la plataforma de Hardware sobre la cual se trabaje. Dentro de la variedad disponible de Hardware, los "Very Low Cost Testers" (VLCT) constituyen, la solución práctica de bajo costo para manufactura y educación; esto se debe a que estas plataformas de prueba se construyen a la medida de las necesidades. En el campo del Software, LabView de la compañía National Instruments (NI) es actualmente uno de los programas mas usados para mediciones y pruebas en el mundo.

A pesar de que los métodos y algoritmos son pasos indispensables dentro del arte de la prueba eléctrica, hay una sensible ausencia de éstos, en lo que respecta a circuitos que funcionen en modo corriente utilizando el VLCT y LabView. Esto es importante debido al uso extendido de circuitos en modo corriente, por ejemplo en equipos de muy bajo voltaje y muy baja potencia.

Por lo tanto, el propósito de este proyecto es generar algoritmos y métodos de prueba usando el VLCT de la compañía Texas Instruments (TI) y LabView con el objetivo de realizar pruebas eléctricas de circuitos que funcionen en modo corriente.

La fórmula mas útil para resolver problemas complejos es el de dividir y vencer. Siguiendo esta filosofía, la mejor estrategia es escoger un circuito básico y pequeño que tenga todas las características de los circuitos en modo corriente. Por lo tanto el marco de este proyecto comprende solamente a circuitos espejos de corriente básicos, debido a que con programas pequenos es posible crear programas más grandes para circuitos complejos.

El objetivo del proyecto es la generación de métodos de prueba. Solamente se aplican estímulos de corriente directa. Los estímulos de corriente alterna están fuera del alcance de este proyecto y se sugieren para investigaciones futuras.

Las contribuciones de este proyecto son:

- Procedimientos y algoritmos para pruebas eléctricas a espejos de corriente usando el VLCT.
- Integración entre VLCT y LabView.
- Programas de prueba en lenguaje Pascal que pueden ser usados para crear programas más complejos.
- Programas de prueba en lenguaje Pascal para el sistema operativo de Microsoft que pueden ser usados para crear un archivo ejecutable.

Agradecimientos

A mi familia en México que aunque estuvimos lejos, estuvimos cerca.

A Yaritza Griselle Gandulla Álvarez, que con inmenso amor y cariño, me impulso a terminar este proyecto.

Agradezco a las familias en Puerto Rico que me apoyaron durante mis estudios. Sobre todo a las familias Hernández Torres, Arroyo Cordero, Palomera Arias, Laureano Correa y a mis compañeros y amigos José Isaac López Rivera, Sigfredo González Díaz, José García, Elías Beauchamp, Miguel A. Laureano, Joan D. Laureano, Miguel A. Laureano, Maria M. Correa, Arturo J. Fradera, Víctor Vargas, Samuel Rosario, Rafael Arce, Germán Ríos, Carlos Vega, Eva del Carmen Palomera, Thalia Arocha, Yuanil Hernández, Tamara Arroyo, Miriam Lugo y Zorima de Jesús Rivera. Muchas otras amistades quedan en la memoria y en el corazón por siempre.

Mención especial merecen los Doctores Rogelio Palomera García y Marco A. Arocha Ordóñez. Por haber sido ejemplo y guía firme en todo momento. Así mismo, agradezco al Ing. Pablo Rebollo, su ayuda en todo momento.

Agradezco a la Universidad de Puerto Rico, la oportunidad de haber cursado estudios en sus instalaciones.

A Dios por la oportunidad de vivir.

Sergio Couttolenc Valdés

July, 2004

TABLE OF CONTENTS

LIST OF FIGURES x

LIST OF TABLES xii

1 **Introduction** 1

1.1 Definition of Testing …………………………………………………….. 1

1.2 Objectives ……………………………………………………………….. 1

1.3 Project Report description ………………………………………………. 3

1.4 Very Low Cost Testers ………………………………………………….. 4

1.5 VLCT Capabilities ………………………………………………………. 5

1.6 Current-Mode Circuits …………………………………………………... 7

1.7 The Objective of Testing ………………………………………………... 8

1.8 Efficient Use of Equipment ……………………………………………... 9

1.9 Economics and Testing ………………………………………………….. 9

1.10 The Manufacturing Process ……………………………………………... 10

2 **Basic Concepts** 11

2.1 Error and Fault ………………………………………………………….. 11

2.2 The Three Critical Concepts in Testing …….…..………………………… 12

2.3 Purpose of Hardware Test ………………………………………………. 12

2.4 Glossary of Fault Models ……………………..………………………... 13

2.5 Current-mode and Voltage-mode ……………………………………… 18

3 **Testing** 22

3.1 Testing of Current-Mode Circuits using the VLCT ………………… 22

3.2 Current Sources and Current Loads…………………………………… 23

3.3 Bipolar Current Mirror ………………………………………………….. 23

3.4 Basic Measurement Programming ……………………. …………….. 24

3.5	Transfer Procedure from Pascal D.O.S. Language to Pascal VLCT	35
3.6	Input Data File using LabView ……………………………………….	36
3.7	Testing of basic bipolar current-mirror using VLCT …………………	38
3.7.1	The circuits under test …………………………………………………	39
3.7.2	VLCT Test programming for BJT current-mirrors …………………...	46
3.7.3	Flow diagrams for the main programs ………………………………..	48
3.8	Test results ……………………………………………………………….	53
3.8.1	DC sweep test applied to basic bipolar current-mirror ……………….	55
3.8.2	DC sweep test applied to bipolar current-mirror with base compensation …………………………………………………………	56
3.8.3	DC sweep test applied to bipolar Wilson current-mirror ……………..	57
3.8.9	Manufacturing and functional test applied to basic BJT current-mirror	58
4	**Conclusions, Drawbacks and future work** ………….……………...	62
4.1	Conclusions ……………………………………………………………..	62
4.2	Future work ……………………………………………………………..	63
	BIBLIOGRAPHY …………………………………………………….	64
	Appendix “A”: Test Programs ………………………………………..	72
	Appendix “B”: Raw data from output data files ……………………..	114
	Appendix “C”: VLCT basic programming manual ………………….	157
	Appendix “D”: Measurement limitations …………………………….	191
	Appendix “E”: Circuit connections and relay notes ………………….	193

LIST OF FIGURES

1.1	Manufacturing Process ………………………………………………..	10
2.1	Series Circuit …………………………………………………………..	18
2.2	BJT output characteristics ……………………………………………..	21
3.1	BJT connected as diode ……………………………………………….	23
3.2	BJT based Current-mirror ……………………………………………..	24
3.3	Basic Measurement Procedure using VLCT …………………………	24
3.4	Integration between systems using DAT files ………………………..	27
3.5	BJT based Current-mirror with Resistor in collector of Q1 …………...	27
3.6	Diagram Flow for the Current-Mirror Simulation Program …………...	30
3.7	Input File format ……………………………………………………….	32
3.8	Output Data File ……………………………………………………….	32
3.9	D.O.S. Pascal programming window ………………………………….	33
3.10	UNIX Pascal VLCT edition window …………………………………	34
3.11	Input data file creator from LabView …………………………………	36
3.12	Input data file creator block diagram …………………………………	37
3.13	Basic bipolar current-mirror with voltage and current display ……….	40
3.14	DC Sweep for bipolar basic current-mirror ……………………………	41
3.15	Bipolar current-mirror with base compensation ………………………	42
3.16	DC Sweep for bipolar current-mirror with base compensation ……….	43
3.17	Bipolar Wilson current-mirror ………………………………………...	44
3.18	DC Sweep for bipolar Wilson current mirror …………………………	45
3.19	Flow diagram for powersupply6 program …………………………….	48
3.20	Flow diagram for dc ramp program …………………………………...	49
3.21	Flow diagram for Functional_test_1 program …………………………	50

3.22 Flow diagram for functional_test_2 program ………………………… 51
3.23 Flow diagram for functional_test_3 program …………………………. 52
3.24 Graph raw data from DC sweep of basic bipolar current-mirror ……... 55
3.25 Graph raw data from DC sweep of basic bipolar current-mirror with base compensation ……………………………………………………. 56
3.26 Graph raw data from DC sweep of Wilson current-mirror …………… 57

LIST OF TABLES

2.1	Current-mode and Voltage-Mode …………………………………….	19
3.1	Variables used in the program …………………………………………	29
3.2	VLCT test programs developed for this project report ……………….	46
3.3	Data results from manufacturing and functional testing for basic bipolar current mirror ………………………………………………….	58
3.4	Central values ………………………………………………………….	59
3.5	Upper and lower limits ………………………………………………...	59
3.6	Test analysis by CPK and sigma ………………………………………	60
3.7	Upper and lower limit modification …………………………………..	61

Chapter 1

1. Introduction

1.1 Definition of Testing

Testing can be defined as the action of applying a set of techniques that ensures the functionality and quality of a product [58]. Research in this area is very important due to the constant need to reduce production cost, increase quality and improve the design. The role of testing is to detect whether something went wrong and the role of diagnosis is to determine exactly what went wrong, and where the process needs to be altered. Therefore, correctness and effectiveness of testing is most important for quality products.

1.2 Objectives

The Purpose of this Project Report is to generate algorithms and methods using the Very Low Cost Tester (VLCT) from Texas Instruments (TI) and LabView® from National Instruments to perform electrical testing of basic Current-Mode circuits. VLCT and LabView® have proved their reliability in the manufacturing field like a low-cost testing solution. This is the main reason for having them as the preferred testing equipment.

There are several Current-mode circuits currently in use [1, 6, 42, 57]. Some examples that can be mentioned are the current conveyors, current-mode amplifiers, high frequency CMOS transconductors, bipolar current mirrors, dynamic current mirrors, switched-current filters, current-mode analog-to-digital, and digital-to-analog converters, current-copiers, neural networks, and many others.

Inherent to the wide field of current-mode circuits we have a field for electrical testing. Nevertheless, there are not enough methods and algorithms developed for electrical testing of current-mode circuits using the VLCT and LabView® software/hardware platforms.

In the following chapters, the most simple but practical current-mode circuits will be treated from the electrical-testing point of view to develop test methods for simple current-mode circuits rather than complex circuits.

This work uses a simple circuit because the objective is to show the viability of the VLCT to use it for analog current-mode circuits as opposed to digital voltage-mode circuits, thus the example used allows verifying our concept without unnecessary concepts.

Bipolar current-mirrors were used as examples of devices under test (DUT). Discrete transistors NPN 2N2222A were used to create the basic configurations due their relative immunity to Electrostatic discharge (ESD) with respect to metal oxide semiconductor (MOS) transistors, thus handling is simplified. The method, of course is not limited by this.

Advantages of the use of discrete transistors are the capability to replace each transistor in order to obtain a different set of readings due the different parameters of each one, like temperature sensibility and Beta. This does not mean a limitation to discrete components or bipolar current-mirrors, the testing method of this project can be used to test array-based current-mirrors as well. Furthermore the programs can be edited with minimum changes to perform testing to CMOS current-mirrors. These minimum changes could be for example the power supply range, voltmeter range or delay times to take into account new settling times.

As a result the testing method used to perform test to simple circuits can serve as a departing point to test more complex circuits using a similar test program with minor modifications.

The VLCT literature does not give attention to analog testing of basic current-mode circuits, thereby the purpose of showing that VLCT capabilities can be extended to cover analog testing of basic current-mode circuits, and moreover the VLCT literature does not mention that the Pascal test program can use simple text files for input data and output data in the analog case. This project has exploited this versatility to facilitate the test data communications to and from the VLCT.

1.3 Report description

In the rest of this chapter, we will review the concept of very low cost testers, the current capabilities for the VLCT-256 installed at ICDL donated for Texas Instruments and as a background, some basic concepts of Current-mode circuits and Testing are exposed.

Basic programming concepts of Electrical Testing and Pascal programming language are given in Chapter 2, more information for Pascal programming is in appendix C, in order to provide a background, for the present research. Details about programming in LabView® are omitted, due the high concentration of information on-line at the web site of National Instruments and a popular set of books available to the public. A useful list of references can be found on reference [43].

On Chapter 3, the integration between VLCT and LabView® is performed using a virtual instrument as a basic example. Basic test methods of simple Current-mode Circuits are explained. Finally, Chapter 4 presents a summary and conclusions from this research and introduces lines of future work. More detailed information in programming, complete Pascal test programs, basic Pascal programming, circuit notes, relay use and measurement limitations are in the appendices.

1.4 Very Low Cost Testers

Low-Cost Testers are a wide practical low price option. They are custom-made design and their measurement capabilities are not-so expensive, making this their most attractive characteristic [17]. On the other hand, from the testing point of view, it could be cheaper to insert test circuits into the chip rather than to provide testing using an expensive external automated test equipment (ATE), but the investment in design is quite high thus any low cost testing equipment is highly appreciated [1, 10, 22, 23, 33 36].

Texas Instruments is a corporation with a deepest concern in testing. They designed a class of custom-made tester, generally named Very Low Cost Tester (VLCT), which falls under the concept of automated test equipment (ATE) and it is oriented to test digital circuits and voltage measurement.

1.5 VLCT Capabilities

The current equipment present in the ICDL facility in the University of Puerto Rico at Mayaguez donated by Texas Instruments Corporation is a VLCT-256 capable of carrying out the following test coverage [3]:

- Continuity or opens and shorts. - Test two things: parameters of electrostatic discharge diodes and path connections from the tester set-up to de Device under Test (DUT) pins.

- I/O Leakage. – Also called input and output (I/O) current test, is a device specification test, and they verify the device under test (DUT) input and outputs pin to be driven to high and a low state, while consuming no more than a specified amount of current.

- Power Supply Current Testing. - The power supplies of the VLCT have current measurement capability, so any ICC test forces a voltage on the power supply and measures the resulting current.

- Quiescent Current Test. - CMOS devices in this state should draw low currents. A transistor that leaks current when it should be off is defective but may operate functionally until some time later when it degrades further and fails.

- Functional Testing. - Functional test patterns are run to verify that the device outputs respond properly to the inputs as defined in the device truth table.

- Stress Testing. - Usually means applying voltages outside the normal operating values for a specified period of time.

- High-Speed Clock and Frequency Testing. - The VLCT functional is event driven and does not program cycle times. Auxiliary hardware is needed.

- Memory Testing. - The VLCT can test embedded directly or through Memory or CPU built-in self-test (BIST), depending on how the device has been designed. Bit fail data can be saved for use in software redundancy analysis and bitmap display.

- CPU Built-in Self-test. - This is a memory test application that may be necessary when memories are tightly integrated with CPU circuitry.

- DC Parametric Testing - DC parametric testing consisting of forcing voltage or current test conditions and measure voltage or current.

- AC Parametric Testing - AC analog testing can be done using Digital Signal Processing (DSP) techniques. Analog input signals are created by driving a DAC (digital-to-analog) and filtering the output. Analog outputs are converted to digital information by an ADC (analog-to-digital) and saved in memory.

These capabilities do not guarantee that the VLCT is the perfect measurement unit. Many more things need to be done by software [17, 26, 34, 38, 43, 45] or with the help of other instruments. Measurement limitations are exposed in appendix “D”; however, the VLCT is extremely valuable testing equipment.

1.6 Current-Mode Circuits

Analog design has historically been viewed as a voltage dominated form of signal processing [1, 42, 57]. This has been apparent in analog Integrated Circuit (IC) design where current signals are readily transferred into the voltage domain before any analog signal processing takes place. For example all of us are used to assemble transistors into voltage oriented circuits and systems and assume that this form of processing is the most appropriate for the task [1, 57], although bipolar junction transistor (BJT) and field effect transistor (FET) are both current output devices [57].

Recent advances in integrated circuit technologies have meant that state-of-the-art analog IC design is now able to explore the potential of current-mode analog signal processing, providing attractive and elegant solutions for many circuit and system problems.

In CMOS technology, with mixed and digital circuitry in VLSI, developments have been centered on a new generation of analog sampled data processing that may be referred to as switched or dynamic current circuits [1, 2, 6, 7, 8, 12, 13, 57]. One of the primary motivations behind these developments has been the shrinking feature size of digital CMOS devices make necessary the reduction of supply voltages [1, 4, 6, 18, 57].

For example, the one of the major improvements of switched or dynamic current circuits, is the capability of checking and correcting themselves [1, 10, 21, 22, 26, 33, 57, 58], however, self-correcting technique may not work properly in the presence of faulty switching elements and is a very expensive technique [1, 22-26, 29, 36, 41].

To bypass the problem, the test engineer should have the ability to develop proper test software. Therefore, the combination between Test Software and efficient use of measurements instruments is the best path to achieve the low cost commitment and quality [2, 31, 37, 58].

1.7 The Objective of Testing

The goal of the test engineer is the manufacturing test cost reduction [58]. Test time reduction decreases the manufacturing cost, because testing is a very expensive activity. Some strategies for test-related reduction cost are:

- Faster data transfer
- Better use of instruments and software
- Parallel execution of test
- Build/Modify test applications faster.
- Integrating test procedures into operations (on-line production test)

In computer and communications industries, testing of electronic components is an important part of the business. Customers want reliable product at a reasonable cost. A manufacturer with a poor design may try to improve reliability by increasing testing with the added cost passed on to the costumer. However, in a competitive market economy, the consumer benefits by selecting the best product. To stay in business, the manufacturer must find ways to provide the best product at the lowest cost. This requires production at the maximum economic efficiency.

Many reasons arise for having a good testing concept. For example, recently Intel corp. reported that the combination of verification testing and manufacturing testing is its major capital cost, and not the 2 billion silicon fabrication lines. Clearly the cost of testing is quite high, thus any cost-related reduction is highly appreciated [20].

1.8 Efficient Use of Equipment

Many companies consider testing to be more than a half of their equipment manufacturing cost. The new and constant advances in technology have given the capability to quickly design and manufacture very complex circuit at reasonable cost.

However the cost of these products has reduced, the percentage of the total cost attributed to testing has increased. High-cost measurement instruments are always a problem for any company. Consequently, efficient use of these tools is essential.

New techniques must be developed or, as previously showed, adequate use of instruments and software are parts of the success in testing. Improvements in software are always cheaper rather than in hardware.

1.9 Economics and Testing

The rewards of testing are quality and economy [22, 17, 58]. These two characteristics are not independent and neither can be defined without the other. Quality means satisfying the user's needs. Economy takes place, when the product is provided at a minimum cost possible. Tests are used to verify/validate product compliance with design, engineering and industry specifications and standards. As a conclusion, the purpose of testing is to make certain a constant quality under a low cost commitment. The Test-related Manufacturing costs are:

- Number or test stations required
- Number (and cost) of instrument required for testing
- Number of specialized programmers required
- Space used/required by test system
- Reusability of test code
- Test equipment required for each product/family of products
- Test time

1.10 The Manufacturing Process

Testing is part of the manufacturing processes. The list shown below consist of a series of basic activities, such as (but not equal in all industries):

- Inventory (includes work-in-progress)
- Transport
- Operations
- Inspection/Test

Operations are the transformation of raw materials/components into a final product. Manufacturing test is a sequence of testing to be executed in one or more test-stations placed in the production line (figure 1.1). From all these activities, only operations add value to the product. All other activities different than operations, should be minimized or eliminated, when possible (i.e. add test to an operation).

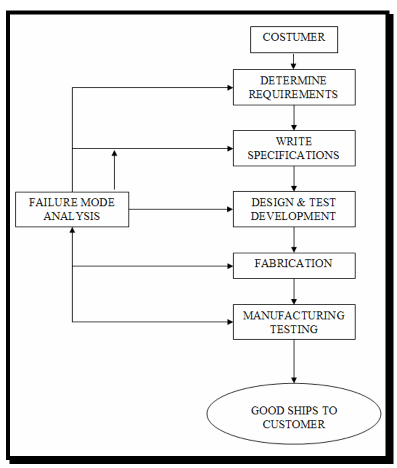


Figure 2.1 Manufacturing Process

Chapter 2

2. Basic Concepts

2.1 Error and Fault

A test Engineer must always be thinking that some kind of error and fault is always present in slight or greater grade. There is not such thing like a perfect measurement. Perfection is a grade of satisfaction. Three concepts arise here: defects, errors and faults [58]. These will be explained through this chapter.

In addition, this chapter will review important test concepts as well as current IC test methods. The basic concepts in this chapter are taken from [58], which is a very selective compilation of several testing IC documents from all over the world. Concepts will be displayed as are in [58]. The reason for this is to transmit the idea without any distortion.

Measure devices are not free of errors; even a well calibrated instrument has always a slightly grade of imperfection. More information about tolerances, instrument validation, percent of error of the instruments and resistors used to measure are available on the appendix “D”.

2.2 The Three Critical Concepts in Testing

The terms defect, error and fault are critical to testing [58]. Formal definitions are as follow:

- **Defect**. - A defect in an electronic system is the unintended difference between the implemented hardware and its intended design [58].
- **Error**. - A wrong output signal produced by a defective system is called an error. An error is an effect whose cause is a defect. This is an imperfection of the hardware [58].
- **Fault**. - A representation of a defect at the abstracted function level is called a fault. This is an imperfection of the functionality of the device [58].

2.3 Purpose of Hardware Test

Hardware test is used to discover any faults caused due to manufacturing defects or errors [58]. Structural testing is the one that depends of the specific circuit structure. It allows us to develop algorithms. Central to these algorithms are fault models. Most generation and test evaluation (fault simulation) algorithms are based on selected fault models. There are levels of fault models:

- *Behavioral level*. - Simulation-based verification rather than in testing.

- *Register-transfer level (RTL) or logic level*. - Consists of a net list of gates and the stuck-at-fault at this level are the most popular fault models in digital testing.

- *Lower levels or Transistor levels* (referred as component levels). - Include stuck-at-open types of faults, and they are also known as technology-dependent faults. Component-level faults are mainly modeled in analog circuit testing.

2.4 Glossary of Fault Models

The following is a glossary of fault models, not all applies to this report, but is useful know the vocabulary [58]:

- Assertion fault. - Means that the corresponding property is not true for some input of the system

- Behavioral. - They refer to incorrect execution of the language constructs used in the description.

- Branch. - Belongs to Behavioral level, the circuit function is described in a programming language. A branch fault affects a branch statement and causes it to branch to an incorrect destination.

- Bridging. - Modeled at the gate or transistor level. Represents a short between groups of signals. Often used as examples of defect-oriented faults.

- Bus. - Specifies the status for each line in a bus as stuck-at-0 or stuck-at-1, or fault-free. Thus, for a n-bit bus, there are 3n-1 bus faults. A total bus fault assumes all lines of the bus to be stuck at the same 0 or 1 state.

- Cross-point. - Modeled in PLA (programmable logic array). The possible fault refers to the cross-point connection or no connection.

- Defect-Oriented Faults. - Faults at the physical level that usually occur during manufacture are called defects. The electrical or logic-level faults that can be produced by physical defects are classified as defect-oriented faults.

- Delay Fault. - These faults cause the combinational delay of a circuit to exceed the clock period.

- Functional Faults. - See behavioral faults.

- Gate-Delay Fault. - The fault increases the inputs to output delay of a single logic gate, while all other gates retain some nominal values of delay.

- Hyperactive Fault. - This one causes a large number of signals in the circuit to differ from their connect values.

- Initialization Fault. - Faults that interfere with such an initialization procedure are called initialization faults.

- Instruction Fault. - Usually modeled in programmable systems like microprocessors or digital processors, an instruction fault causes an intended instruction to be incorrectly executed.

- Intermittent Fault. - A fault that appears and disappears as a function of time is called intermittent fault.

- Line-Delay Fault. - This fault models, the rising and falling delays of a given signal line.

- Logical Faults. - These faults affect the state of logic signals. The state can be modeled as {0, 1, x (unknown), Z (high impedance)}.

- Memory Faults. - Fault modeled in memory blocks are single cell stuck-at-[0, 1] faults, pattern sensitive faults, cell coupling faults, and single stuck-at faults in the address decoder logic.

- Multiple Faults. - Represents a condition caused by the simultaneous presence of a group of single faults. Multiple stuck-at faults are usually not considered in practice.

- Non-Classical Faults. - Refers to a fault other than stuck-at-fault. Term has been used for the stuck-open and stuck-short faults of MOS technologies.

- Oscillation Faults. - Cause oscillating signals in the faulty circuit when the fault-free circuit remains stable.

- Parametric Faults. - Such a fault changes the values of electrical parameters of passive or active devices from their nominal values or expected values.

- Path-Delay Faults. - This fault causes the cumulative propagation delay or a combinational path to increase beyond some specified time duration.

- Pattern Sensitive Fault. - Causes an incorrect behavior in a certain part of the circuit only when a specific state occurs in some other part.

- Permanent Fault. - Any faulty behavior that does not change with time is called a permanent fault.

- Physical Faults. - These faults cause physical changes in the circuit. Also sometimes referred as defect-oriented faults.

- Pin Fault. - The stuck-at faults on the signals pins (not power and ground pins) of all modules in the circuit.

- Potentially Detectable Fault. - When a test is applied to a sequential circuit, certain faults produce an unknown state at the output when a deterministic output is expected in the fault-free circuit.

- Quiescent Current (IDDQ) Fault. - These faults are relevant to CMOS.

- Race Fault. - Stuck-at faults that cause a race condition in the circuit are called race faults.

- Redundant Fault. - Consider a combinational circuit. Any fault that does not modify the input-output function of the circuit is called a redundant fault. It can not be detected by any test.

- Segment-Delay Fault. - A segment-delay fault increases the delay of a segment such all paths containing the segment will have a path-delay fault.

- Structural Faults. - The term structural faults are commonly used not for faults modeled in the layout, but rather in gate-level interconnect.

- Stuck-at Fault. - This fault is modeled by assign a fixed (0, 1) value to a signal line in the circuit.

- Stuck-Open and Stuck-Short Faults. - This fault assumes just one transistor to be stuck-open or stuck-short.

- Transistor Faults. - Stuck-open and stuck-short faults are generally referred to as transistor faults.

- Transition Faults. - It is assumed that in the fault-free circuit all gates have some nominal delays and that the delay of a single gate has changed.

- Un-testable Fault. - Cannot be detected by any mean.

2.5 Current-mode and voltage-mode concepts

The publication diversity of the Current-mode subject supports or rejects this concept [42, 56, 57]. The difference between both concepts seems to be a point of view or an operation region. This subject takes great importance due the high concentration of low-power and low-voltage devices in the electronic market. Therefore is necessary to define Current-mode and Voltage-mode. For a short explanation take for example the DC series-resistor circuit of Figure 2.1:

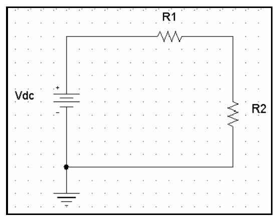


Figure 2.1 Series Circuit

For the current I_t of the circuit in Figure 2.1, we have:

$$V_2 = \frac{V_{DC}\,R_2}{R_1 + R_2} \quad \text{and} \quad I_t = \frac{V_{DC}}{R_1 + R_2}$$

If we introduce the relation $R_2 = XR_1$, we find:

$$I_t = \frac{V_{DC}}{R_1\,(1+X)} \quad \text{and} \quad V_2 = \frac{X}{1 + X}\,V_{DC} = XR_1\,I_t$$

These are equations that involve the circuit current. Now consider the case where $X<<1$, hence $R_1>>R_2$.

If $_{VDC}$ is equal to 5 volts and R_1 is equal to 1 KO, the current value is equal to 0.0005 Amperes, this is the current that R_1 sees. This current will be called $I_{virtual}$. But now remember that X<<1, for example X= 0.0007 then according with:

$$I_t = \frac{V_{DC}}{R_1 + X\,R_1}$$

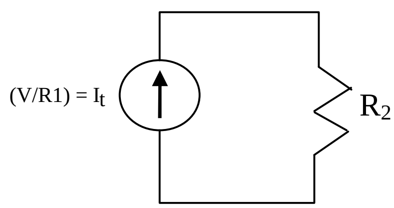
The value for I_t is equal to 0.004996502 amperes. This current will be called I_{real}. This is the first case shown in table 2.1 and our percent of change between both I_{real} and $I_{virtual}$ calculated in table 2.1 is 0.07.

CURRENT-MODE ZONE					**PERCENT OF CURRENT VARIATION**	
V=5	**R_1=1 KO**	**$I_{virtual}$**	**X << 1**	**I_{Real}=(V/R_1)*(1/(1+X)) in amperes**	**($I_{virtual}$ - I_{real})/I_{real} *100**	**V_2 = XR_1*I_{real} in volts**
5	1000	0.005	0.0007	0.004996502	0.07	0.003497552
5	1000	0.005	0.0008	0.004996003	0.08	0.003996803
5	1000	0.005	0.0009	0.004995504	0.09	0.004495954
5	1000	0.005	0.0011	0.004994506	0.11	0.005493957
5	1000	0.005	0.002	0.00499002	0.2	0.00998004
5	1000	0.005	0.05	0.004761905	5	0.238095238
5	1000	0.005	0.1	0.004545455	10	0.454545455
			CURRENT VARIATION	**0.000451048**	**VOLTAGE VARIATION**	**0.451047903**
			Current variation in percent	**9.027 %**	**Voltage variation in percent**	**128.96 %**

VOLTAGE-MODE ZONE					**PERCENT OF CURRENT VARIATION**	
V=5	**R_1=1 KO**	**$I_{virtual}$**	**X << 1**	**I_{Real}=(V/R_1)*(1/(1+X)) In amperes**	**($I_{virtual}$ - I_{real})/I_{real} *100**	**V_2 = XR_1*I_{real} in volts**
5	1000	0.005	100	4.9505E-05	10000	4.95049505
5	1000	0.005	200	2.48756E-05	20000	4.975124378
5	1000	0.005	500	9.98004E-06	50000	4.99001996
5	1000	0.005	1000	4.995E-06	100000	4.995004995
			CURRENT VARIATION	**4.45099E-05**	**VOLTAGE VARIATION**	**0.044509946**
			Current variation in percent	**8.90 %**	**Voltage variation in percent**	**0.89 %**

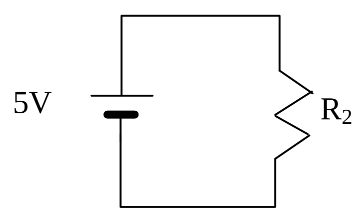
Table 2.1 Current-mode and Voltage-mode

If we repeat this calculation according with Table 2.1, varying the value of X from 0.0007 to 0.1, we find that the current variation of I_{real} is 0.0004 amperes with a maximum change of 9.027%, but the voltage variation is 128.96%. Hence the current variation is negligible in spite the voltage variation of V_2. Therefore we could say that R_2 is operating in current-mode and the following model is acceptable. Within 10% of error from R_2 point of view.



- $R_1 >> R_2$
- $X << 1$
- $R_2 = X R_1$

For the contrary, if X >> 1 as shown in table 2.1, the percent of change in current is very high but the value of V_2 is minimal, therefore, R_2 sees a voltage-source and the following model is acceptable within a 0.89% error:



- $X >> 1$

In conclusion, for this simple circuit, there is a region of operation where the voltage variation in R2 has a minimal impact in the current variation, therefore, we can state that this region of operation could be called Current-mode [42, 56, 57].

This behavior is seen in the relationship between the collector current and the collector emitter voltage for a bipolar junction transistor in figure 2.2:

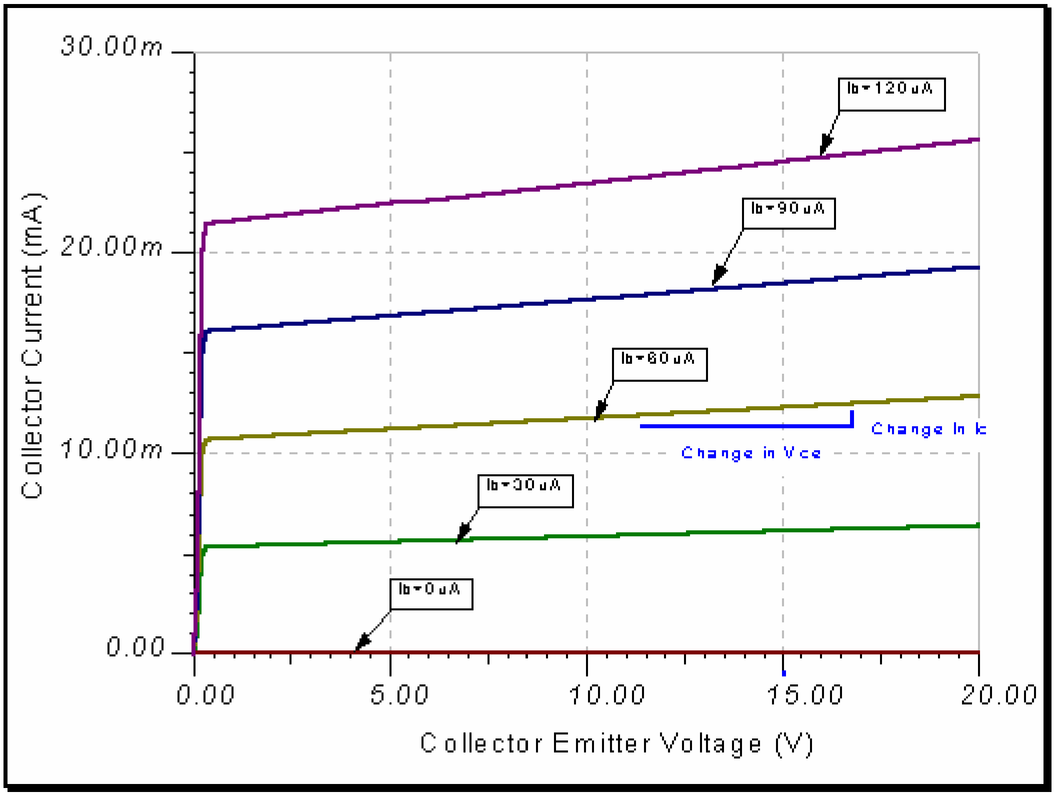


Figure 2.2 BJT output characteristics

Chapter 3

3. Testing

3.1 Testing of Current-Mode Circuits using the VLCT

The chapter's objective is the generation of methods and algorithms to be used in the Very Low Cost Tester (VLCT) testing platform in order to perform electrical testing to Current-Mode circuits. The most useful formula to solve complex problems is to divide and conquer [58]. Following this philosophy, a basic and small circuit that holds all the principles of current-mode circuits is the best option to develop algorithms and programs.

In consequence the project framework comprises only the basic current-mirrors, because with small algorithms and programs is possible to create bigger programs for complex circuits. These simple but practical current-mode circuits will be treated from the electrical-testing point of view to develop test methods for simple current-mode circuits rather than complex circuits. Bipolar current-mirrors were used as an example of device under test (DUT). Discrete transistors NPN 2N2222A were used to create the basic current-mirrors due their relative immunity to Electrostatic discharge (ESD).

Advantages of the use of discrete transistors are the capability to replace each transistor in order to obtain a different set of readings due the different parameters of each one, like temperature sensibility and Beta. This does not mean a limitation to discrete components or bipolar current-mirrors, the testing method of this project can be used to test array-based current-mirrors. Furthermore the programs can be edited with minimum changes to perform testing to CMOS current-mirrors. These minimum changes could be for example the power supply range or voltmeter range.

Only DC stimuli will be used. The AC stimuli are out of the scope of this project, should be proposed for future research.

3.2 Current Sources and Current Loads

Current sources are widely used in integrated electronics circuits as elements of polarization and active loads in amplifying stages. In terms of space they are economical, mainly when the currents are very low.

The current sources as active loads provide incremental resistance of high value being useful in amplifying stages with high gain and operating even with low levels of polarization voltage, specially common emitter or differential stages, where the use of resistances are avoided because of the high values required.

3.3 Bipolar Current Mirror

The Bipolar current-mirror is constituted by two identical Bipolar Junction Transistors (BJT) and both have the same V_{BE} voltage. One of them operates in diode mode (Figure 3.1) with the base and the collector in short-circuit. All the circuit behaves like a current source of value I_o [55].

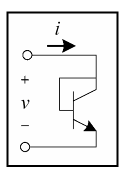


Figure 3.1 BJT connected as diode [55] configuration used in current-mirrors

The Figure 3.2 shows a basic BJT based current-mirror. This is the simplest circuit that we are going to test that holds the behavior of current-mode. All currents are displayed. In the test programs we are going to verify the status of the main currents to know, the reference current (I_{ref}) and the output current (I_o):

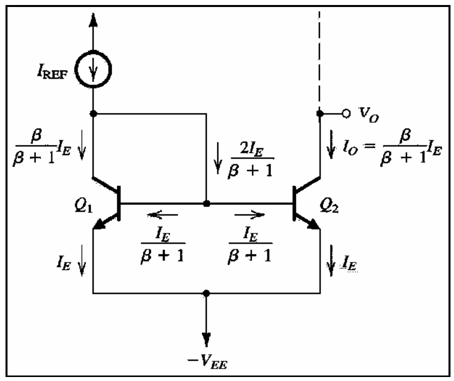


Figure 3.2 BJT based Current-mirror [55]

3.4 Basic Measurement Programming

A measurement procedure for any analog circuit is illustrated by Figure 3.3:

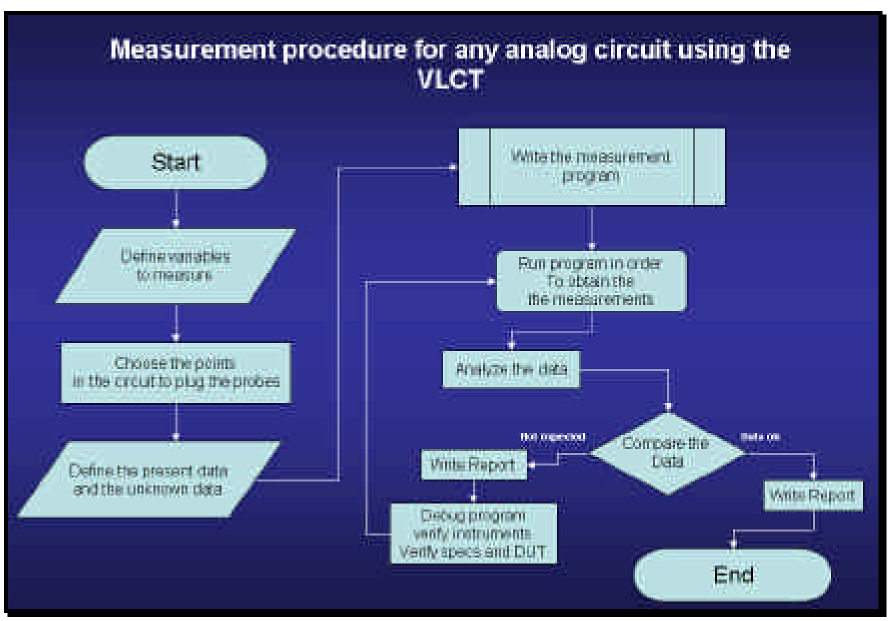


Figure 3.3 Basic measurement procedure using VLCT

Each step procedure is explained next:

1. Define variables to measure
2. Choose the points in the circuit to plug the probes
3. Define the present data and the unknown data
4. Write the measurement program
5. Run program in order to obtain the measurements
6. Analyze data
7. Compare data
8. If the data is expected the procedure ends with the report
9. If the data is unexpected a report is needed to, then debug program, verify instruments or review the DUT
10. Run program again

Next, each step is reviewed:

1. Define what we need to know from the circuit. This is the most important point because is the step where we define the entire process.

2. Circuit adaptation or fixture modification that depends on tester configuration. Most of the times, the tester fixture need to be adapted physically to the DUT.

3. According to the equation that models the circuit operation with the available equations, the available data is reviewed and the unknown data is obtained.

4. Write a program to obtain the unknown data without applying a stimulus to the DUT. This is for simulation purposes. The program uses the equations defined in the above step.

5. Program the tester to realize the measurements needed.

6. The program running without any error it should be given the required data.

7. Comparison between the data of the simulation, and the data coming from the measurement operations.

8. Analysis of the available data, it is possible to re-measure the circuit with other stimuli to review any variation in the results or simple to confirm the actual results.

Using the above procedure, let us to construct an introductory Pascal program [48] that will be used to:

1. Obtain an equivalent resistor (R_{ref}) in the collector of Q_1.
2. Obtain the Output Current (I_o).
3. Obtain the Voltage reference (V_{ref}).
4. Obtain the Output Voltage in order to confirm the state of Q_2.

The input data will be extracted from a data file, the results will be saved in a data file. The reasons of this are to establish an interchange data path between the different operative systems and to provide a friendly platform for the reader using the D.O.S. and for future research. The operative systems involved here are:

- For VLCT Pascal Language Programming, the UNIX© operative system [3].
- For the standard Pascal programming language, the DOS© (disk operating system of Microsoft©) [48].

The Figure 3.4 shows the integration between systems: DAT files are the union point <u>because they can be open and closed in any system for any program</u>. The other platforms can be Excel, Matlab, Mathematica [51, 54] or any other analysis or data base program. The DAQ and GIPB are complements for LabView® y are displayed in Figure 3.4 for illustrative purposes only, but remains for future work develop software platform that could use those complements and the VLCT.

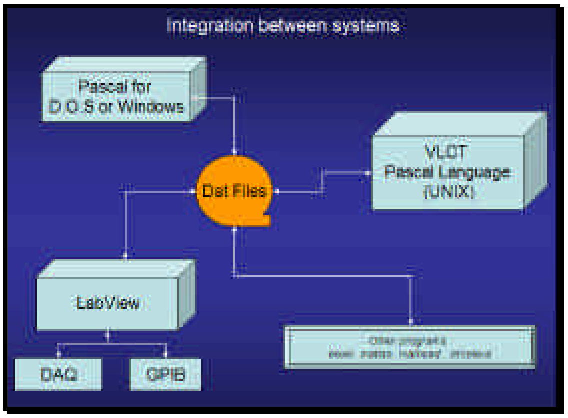


Figure 3.4 Integration between systems using DAT files

Following the procedure, let us work in the analysis of the circuit for this example:

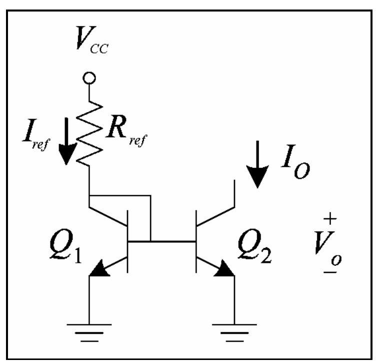


Figure 3.5 BJT based Current-mirror with Resistor in collector of Q_1

The following data is given:

- Power supply voltage (V_{cc})
- Reference Current (I_{ref})
- Base-Emitter Voltage (V_{BE})
- Beta (must be the same for both transistors)

- The early voltage (V_A)

The following data is unknown:

- Output Current (I_o)
- Output Voltage (V_o)
- Reference Resistor (R_{ref})
- Reference Voltage (V_{ref})

It is assumed from the simulation in Pspice [46], that the mismatch between both transistors is very low or can be neglected [49, 55] in consequence we assume that Q_1 must be equal to Q_2. The following equations [49, 50, 55] will be used to write the program:

- For the R_{ref}:

$$R_{ref} = \frac{V_{cc} - V_{BE}}{I_{ref}}$$

- For the I_o:

$$Io = \frac{I_{ref}}{1 + 2/ß}$$

- For the V_{ref}:

$$V_{ref} = V_{cc} - V_{BE}$$

- For the V_O:

$$Vo \sim V_{BE} + \left[V_A \left(\frac{Io\ (1 + 2/ß}{I_{ref}} \right) - 1 \right]$$

Now that the equations were defined, let us replace the data by variables to be used in the program. In many cases variable names like “process_?” are used to divide the calculations in several steps inside the program for easy debugging:

Input Variables	Process Variables
VCC	RES
IREF	IOUTPUT
VBE	VREF
BETA	VOUTPUT
	PROCESS_1
	PROCESS_2
	PROCESS_3
	PROCESS_4

Table 3.1 Variables used in the program

The program in its D.O.S. version will be used the following files:

- For the data input: Dat_inX.dat
- For the data output: Dat_outX.dat

Both files are text files; you can create them using any text editor or the special virtual instrument of LabView®, created specially for this purpose. The number X is used in when the program has more than one version, for example a given code could be used for many current-mirrors variations. Next the Diagram flow of the program:

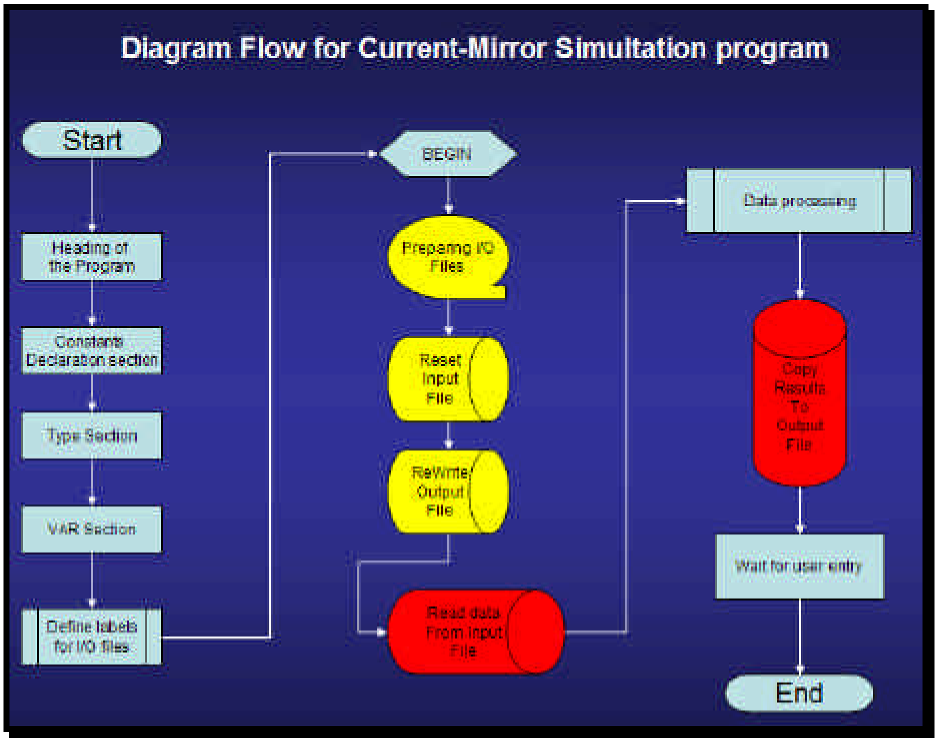


Figure 3.6 Diagram Flow for the Current-Mirror Simulation Program

The key of all the programs in this project, are the code represented by the yellow and red symbols. It is where the files are prepared to be read and written. That code has specific instructions and these instructions allow us to share the information between systems. Now, let us explain each operation showed in the above Figure, the following algorithm helps to clarify the idea:

1. The program must have a heading.
2. In order to be an open code, comments must be included.
3. The section of constants declaration, allow us to define global constants.
4. The “Type” section, is where we define our specific variable types, for example a variable type named “string [40]” will store a chain of 40 characters.
5. The variable declaration section is where we indicate to the program the following information:

- Variables for input/output data transfer.
- Variables used to data processing.
- Variables for program flow control.
- Variables used to manage chains of characters.

6. Special mention is given to the labels for input/output files. The program requires a link between the language and the operative system, in order to open and close files. Here a label is given to each file,. Then during the program, any instruction related with a file will be indicated using the label.
7. After the labels for I/O files are given, the program open for read the input file and the output file for written. These are the sections Reset and Re-Write.
8. Next step is where the program reads the input data, from the input file.
9. Later the program process the data, using equations, decisions, take measurements or any other operation used to obtain any useful data.
10. Finally the program copies the results in the output file and after this, the output file is closed.
11. Last step is a delay for user input.

The format of the input file is defined in the section where the data is read. The data in the input file must be formatted accord with the instructions “readln” from the code. This will allow us to design any type of format for the input file. For this example, the input file has the following format:

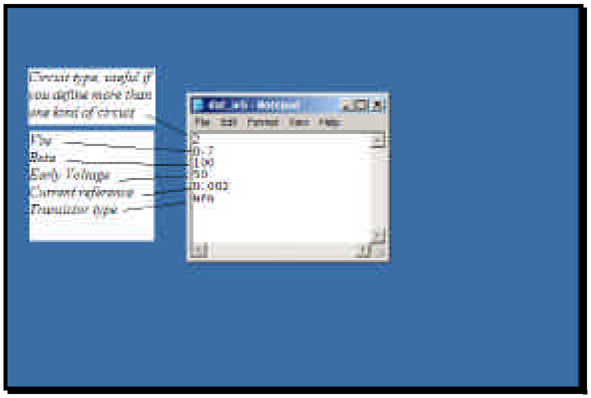


Figure 3.7 Input File format

Next the output data file, using the above input data:

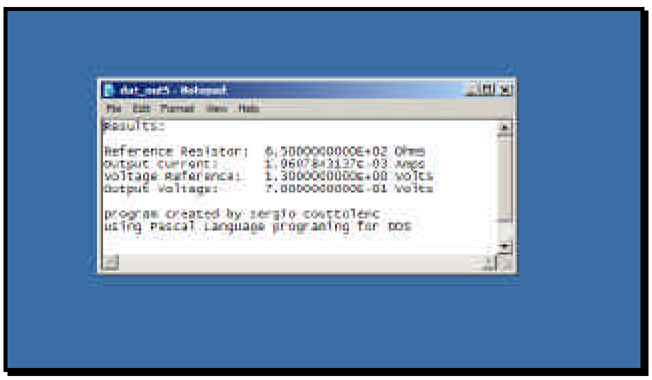


Figure 3.8 Output data file

The complete code is exposed in Appendix "A". Here we will only show the input and output data. Next is the program window for the D.O.S. Pascal programming language:

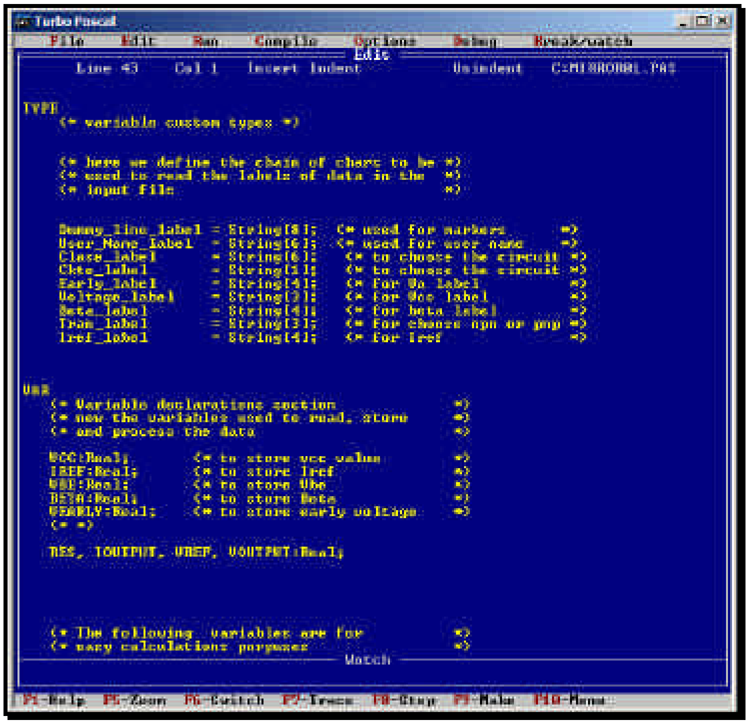


Table 3.9 D.O.S. Pascal Programming Window

The same code can be used with minor changes in the VLCT Pascal programming window. Next the procedure to use the same code in both systems:

1) Write the code in one system first for example in D.O.S.
2) Debug the code
3) When the code is error-free, save the file with another name and without system extension
4) In the new system, for example UNIX, open the file and save it with the proper extension
5) Review the instructions to open and close files
6) Review the calling to procedures
7) Review the variable types and units
8) Debug the code
9) When the code is error-free save it with the proper name.

Figure 3.10 shows the same code in the UNIX environment:

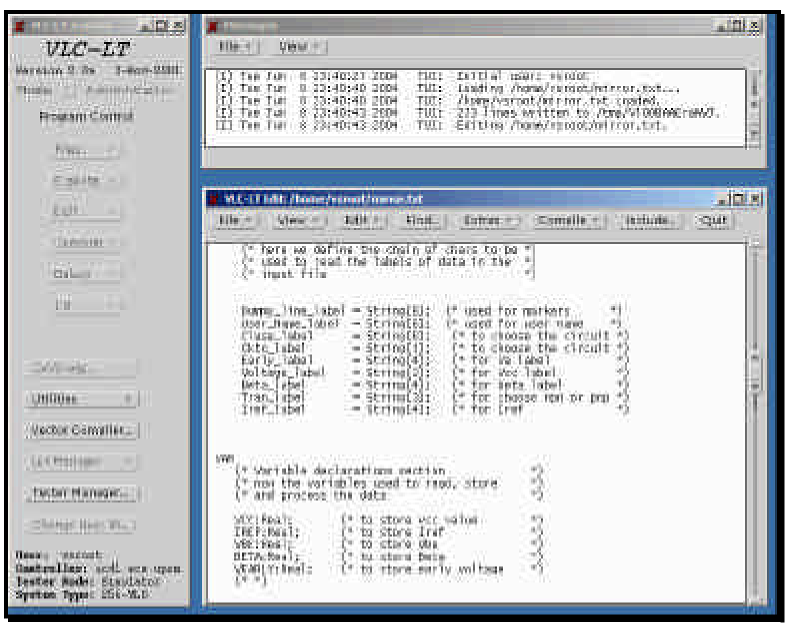


Figure 3.10 UNIX Pascal VLCT edition windows

3.5 Transfer Procedure from Pascal D.O.S Language to Pascal VLTC

In order to use the D.O.S. version of the program, the user must transfer it, manually using secure shell. Then, only two changes must be done in the code:

- Delete the “assign” lines.
- Use the command Reset and Rewrite like the following example:

  Reset(entrada, ‘dat_in5.dat’);

  Rewrite(salida, ‘dat_out5.dat’);
- Replace the instruction “close” by “FileClose”.

The rest of the code remains without change.

The difference between the D.O.S. version and the UNIX version of Pascal programming language are:

- The UNIX VLCT version compiles the file but does not make an executable file
- The D.O.S. versions compile the file and make it executable
- VLCT Pascal test language has a huge support of engineering units including unit conversion and symbol.

The advantage of the D.O.S. version is that the user can create an executable file, that is own custom-made “mirror-spice” to simulate the current-mirrors according to specific needs. This is a useful tool if you want to compare data from simulation programs versus the data from Pascal programming.

Another possible option is that the input and output files can be created by an upper program. This upper program could be written in Visual Basic or LabView® supporting a graphic interface.

The advantage of the UNIX VLCT version is that this version is the native for the hardware of the VLCT. There are many instructions available for digital testing for more information consult VLCT Manuals.

3.6 Input Data File using LabView®

Previously, it was mentioned that the input data file used for both Pascal versions, could be created by another software platform. One of the most suitable software for this purpose is LabView®, because is widely accepted in the testing and measurement field. Next are both parts of the virtual instrument from LabView® which allow us to write down the input data needed by our test programs. This is the virtual instrument on Figure 3.11, that later will be used to send data for a VLCT test program.

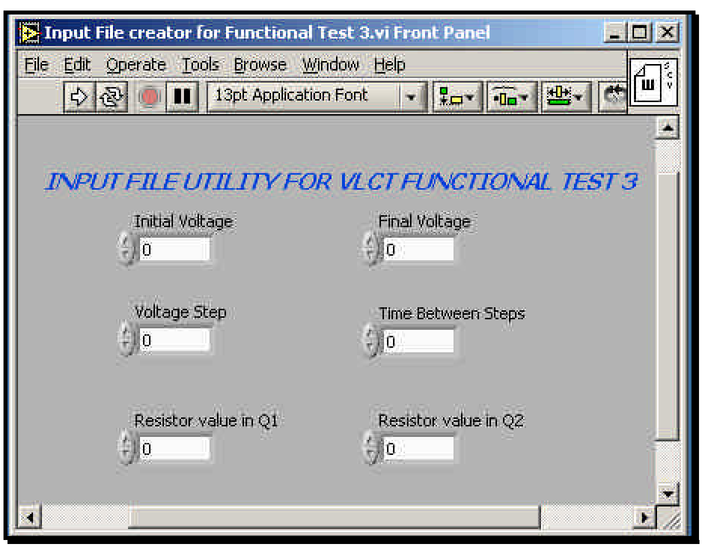


Figure 3.11 Input Data File Creator from LabView®

The internal code in LabView® is very different from any other programming language, for more information for LabView® programming procedures, check references. Next the internal code for this virtual instrument is showed on Figure 3.12:

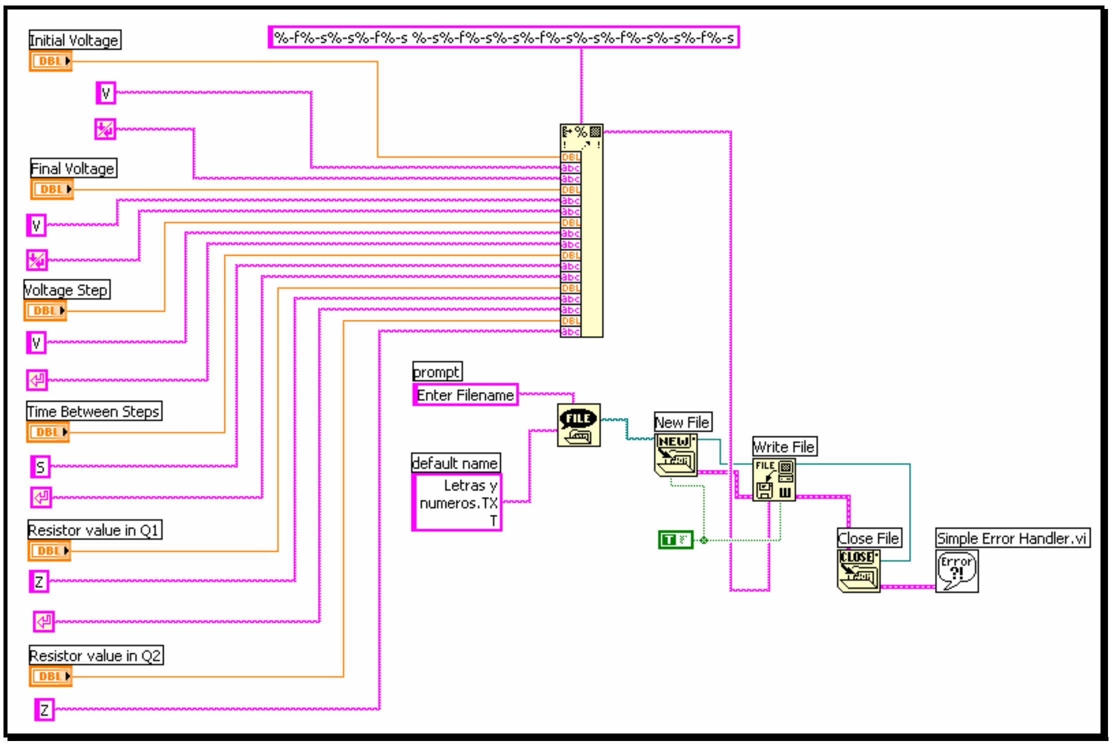


Figure 3.12 Input data file Creator block diagram

The modus operandi of this virtual instrument is very simple:

a) The user must write in the text boxes, the according magnitudes.

b) The user must press the icon: 

c) Then the user must write the file path and name where the output file will be saved.

d) Finally the file is saved and ready to transfer to the VLCT Station.

Later this virtual instrument is used to send data to the Functional_test_3 program, who works under the VLCT test platform. Therefore we are sending data from a

Windows platform to UNIX platform. This virtual instrument could be used to create a single line of VLCT Pascal program; therefore, it is possible the creation of a complete VLCT Pascal program using LabView®. This last issue remains for future investigation.

3.7 Testing of Basic Bipolar Current-mirrors using VLCT

The following sections explain the procedures, methods and programs for testing of basic bipolar current-mirrors. Nevertheless, the algorithms and programs can be edited to perform testing to any kind of analog circuits using a DC power supply. This information is not present in the VLCT manuals, because VLCT is originally oriented to testing of digital circuits. Therefore we are covering the lack of information for this equipment.

The Pspice simulations are used to establish a reference point assuming ideal devices that we are going to compare versus real devices. Simulations does not mean an exact or real value, they are just a reference point. The difference between the simulations and the ideal devices will be more realistic because we are using discrete components that have data dispersion. Bipolar current-mirrors were used as an example of device under test (DUT). Discrete transistors NPN 2N2222A were used to create the basic current-mirrors due their relative immunity to Electrostatic discharge (ESD).

Advantages of the use of discrete transistors are the capability to replace each transistor in order to obtain a different set of readings due the different parameters of each one, like temperature sensibility and Beta. This does not mean a limitation to discrete components or bipolar current-mirrors, the testing method of this project can be used to test array-based current-mirrors. Furthermore the programs can be edited with minimum changes to perform testing to CMOS current-mirrors. These minimum changes could be for example the power supply range or voltmeter range.

Readings are not 100% exactly. Calibration and validation tools for the measurement devices used in this project were not available. Please consult appendix “D” for more information.

3.7.1 The Circuits under Test

The following bipolar current-mirrors are the circuit under test:

- Basic bipolar current-mirror.
- Bipolar current-mirror with base compensation
- Wilson current-mirror

All circuits are using exclusively BJT 2N2222A NPN amplifier and switch at configuration TO-92(72). The reasons why this transistor was chosen are his current collector at 600mA and low power at 625mW.

All circuits present 1 fixed resistor of 1 KO between VCC and Q1 and 1 variable resistor, adjustable from 0 Ohms to 1 KO. The reason for this is to generate an error between the current reference and the output current. Next are the actual schematics of the current-mirrors used like DUT.

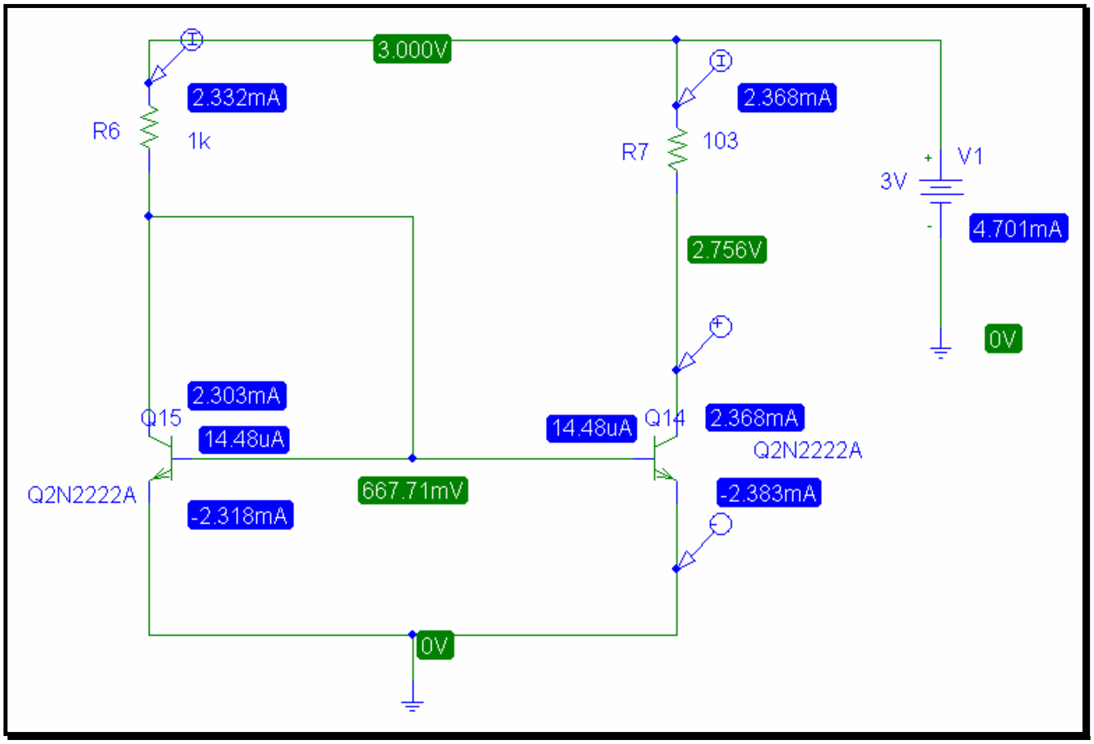


Figure 3.13 Basic Bipolar current-mirror with voltage and current display

The voltage and current values showed for the circuit in Figure 3.13 are the actual result of the simulation. These values are the same that can be obtained using measurement devices. In the real circuit the 103 Ohms are setting by a variable resistor, but for simulation purposes, the same resistance value of the variable resistor is the value for R7.

The above condition is the same for the rest of the circuits. The following Figure 3.14 is a DC Sweep performed for the power supply "V1" with a voltage sweep from 0 to 5 volts. Reason for this is to analyze the behavior for both currents: reference and output when the power supply varies the voltage but keeping the resistor fixed.

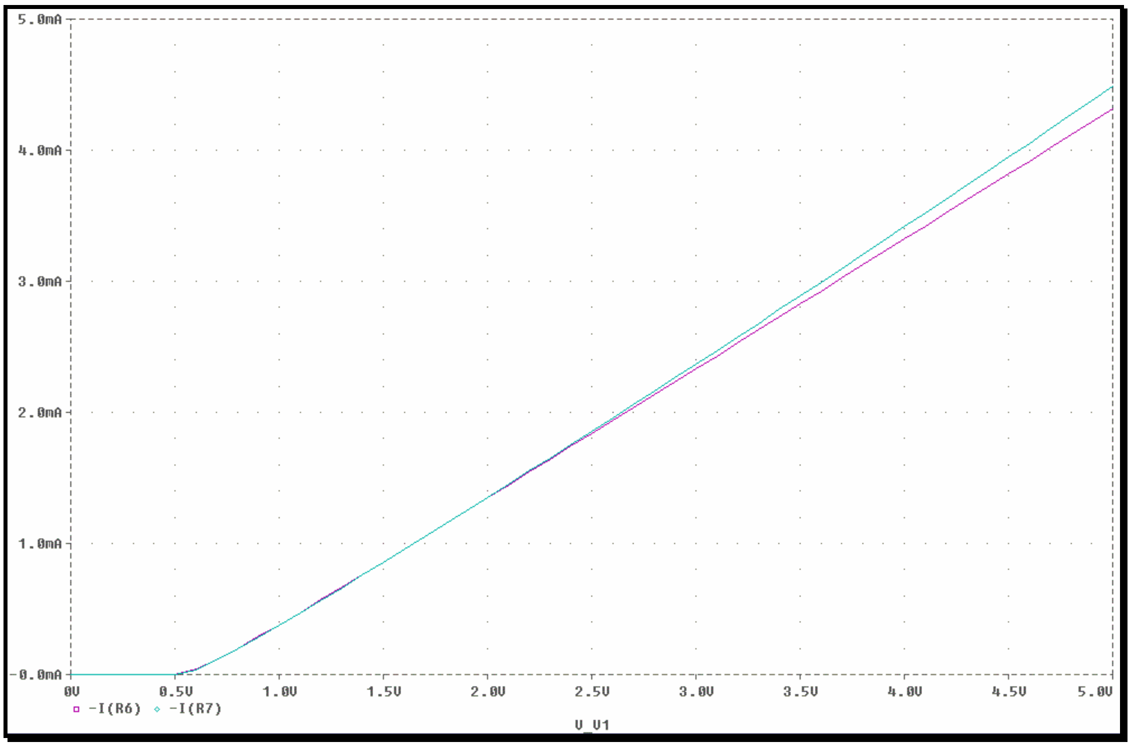


Figure 3.14 DC Sweep for bipolar basic current-mirror

From the Figure 3.14, is straightforward to note that there is a region from 0.7 volts to 2.5 volts, where the reference current and the output current in R7 are almost equal, but when the dc voltage goes up, the currents trends to separate.

Reason for this is the voltage drop caused by the resistors; if both resistors where equals, the currents will be almost equals for practical purposes. Therefore we are generating an error in the circuit with the resistors. Later this will be detected by the VLCT with our test programs.

The second circuit used for this project is next, a bipolar current-mirror with base compensation:

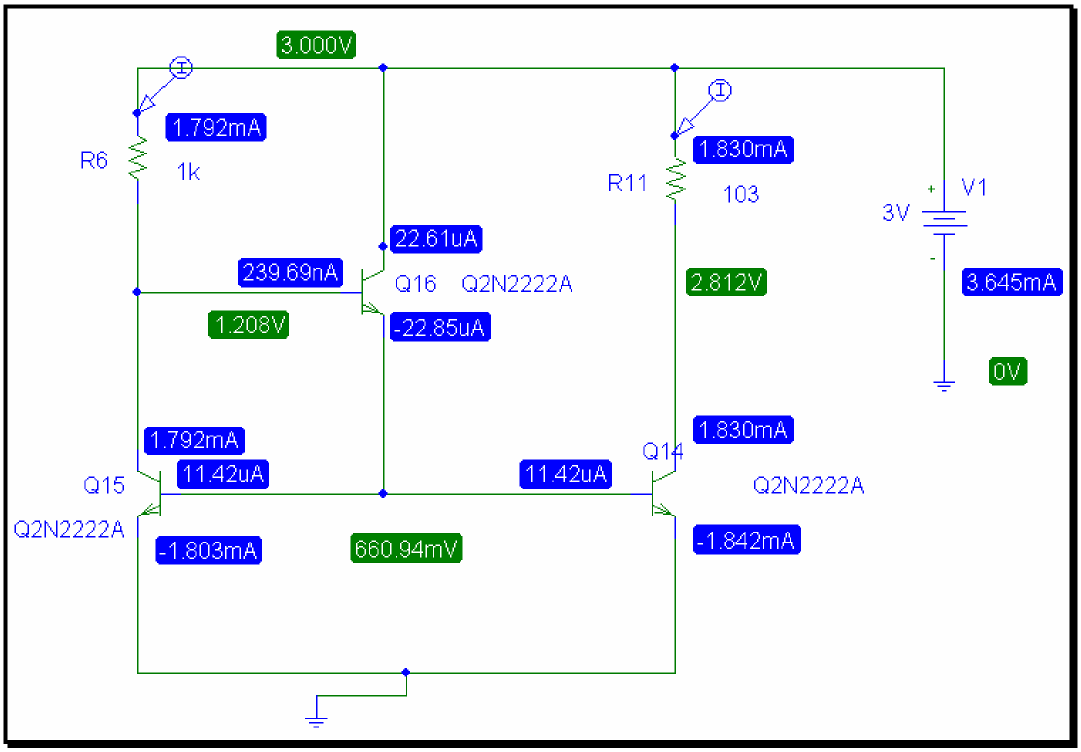


Figure 3.15 Bipolar current-mirror with base compensation

The voltage and current values showed for the circuit in Figure 3.15 are the actual result of the simulation. These values are the same to those that can be obtained using measurement devices or the VLCT with a test program.

The Figure 3.16 is the DC Sweep for the above circuit, using the same parameters for the DC power supply "V1" that those used in the previous example.

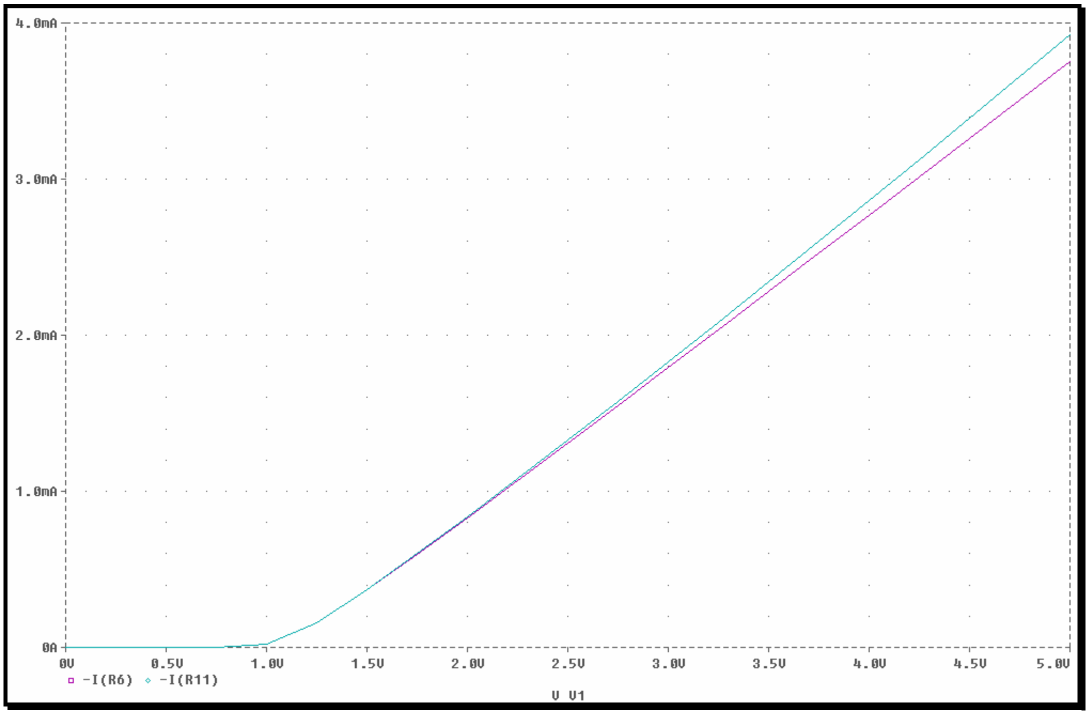


Figure 3.16 DC Sweep for bipolar current-mirror with base compensation

The same characteristic previously appears in the Figure 3.16, there is a region between 1.5 volts and 2.5 volts where the currents are almost equal, but when the DC power supply goes up, the currents trend to be different. This behavior will be verified with the test program in the VLCT.

The following circuit is a bipolar Wilson Current-mirror:

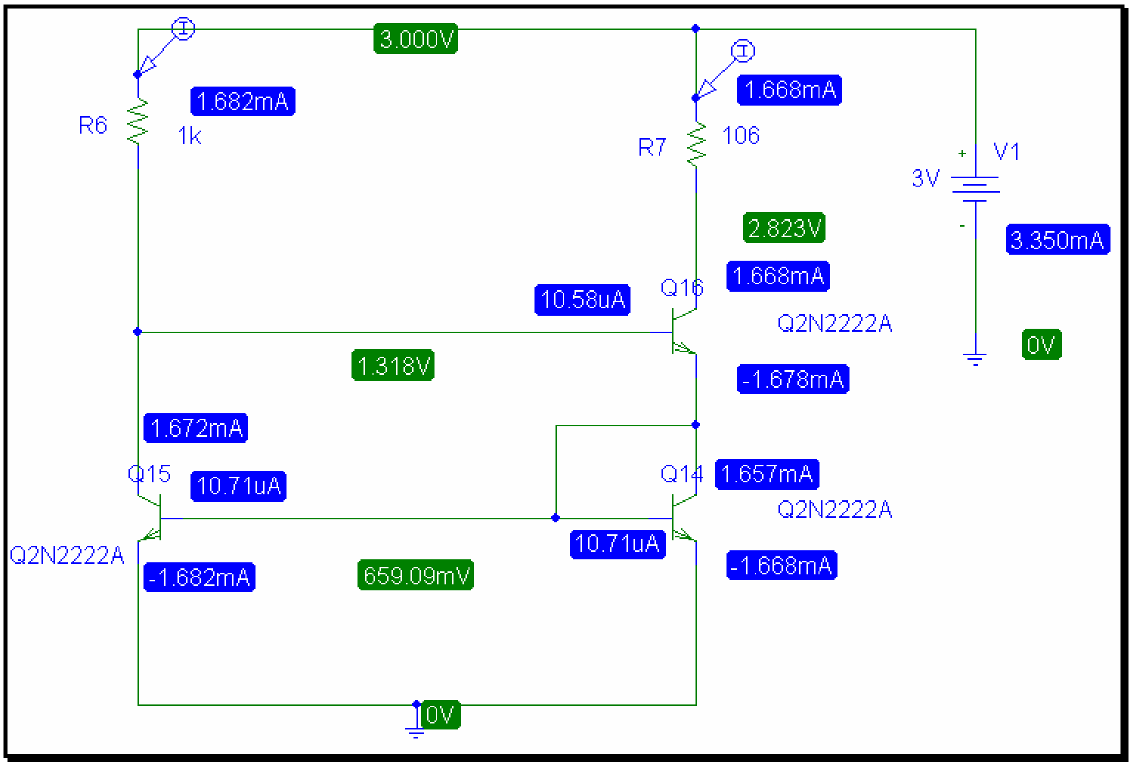


Figure 3.17 Bipolar Wilson current-mirror

The voltage and current values showed for the circuit in Figure 3.17 are the actual result of the simulation. These values are the same to those that can be obtained using measurement devices or the VLCT with a test program.

The Figure 3.18 is the DC Sweep for the above circuit, using the same parameters for the DC power supply “V1” that those used in the previous example.

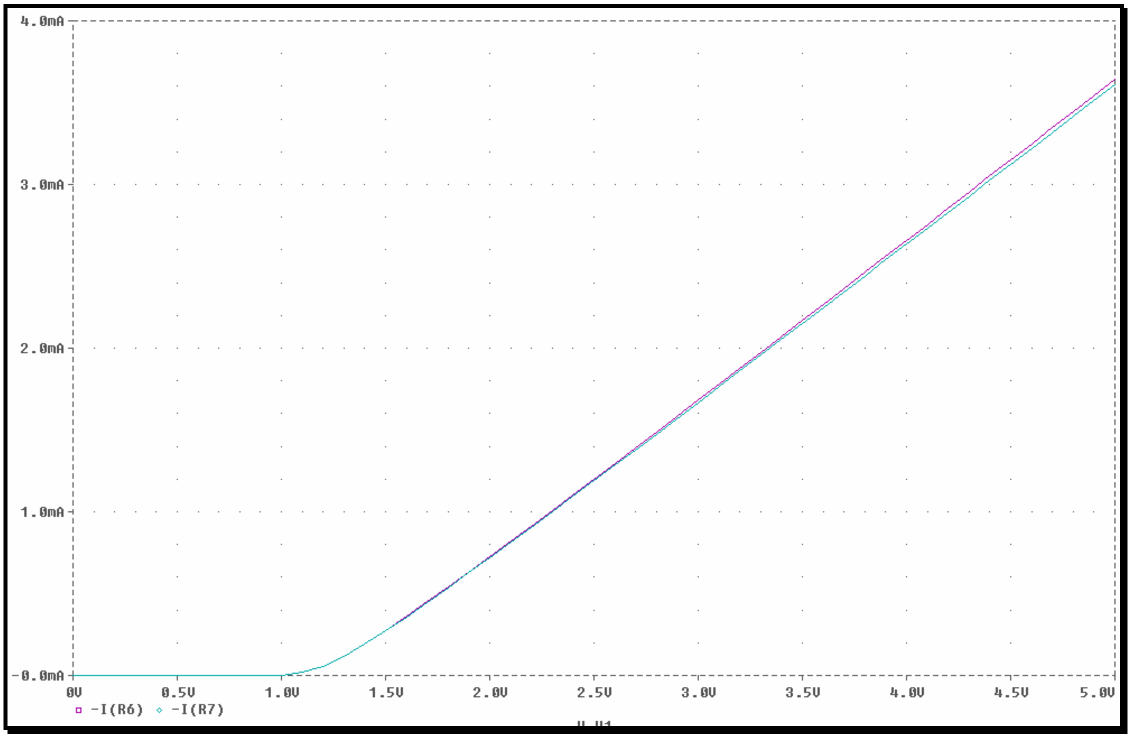


Figure 3.18 Bipolar Wilson current-mirror

In Figure 3.18, the stability of both currents is remarkable, the operation region extends from 1 volt to 5 volts keeping almost the same value for both currents, and this behavior will be verified with the test program in the VLCT.

3.7.2 VLCT Test Programming for Bipolar Current-mirrors

Previously, some current-mirror circuits were introduced and their DC sweep behavior explained in order to understand the trends in the current reference and the output current. The following test programs for the VLCT are explained next:

Program Name	***Used to***	***input data file***	***Data in screen***	***Output data file***
Example_1	Create a basic while-do cycle		¦	
Powersupply6	Setup, shutdown and control the DPS1A power supply.		¦	
Mirror_vlct	Calculate basic parameters in bipolar current mirror, originally the program was written in Pascal for D.O.S. and later exported to VLCT Pascal programming language.	¦		¦
Analog_1	Measures a voltage drop between analog channels 35 and 40. No use of the VLCT power supply.		¦	
Mirror_1	Measures a voltage drop, using analog channels 34, 40 (for the first resistor) and 43, 48 for the second resistor. No use of VLCT power supply		¦	
Shortckt_test_1	Detects if are any short circuit between vcc pin and ground pin. Use the VLCT power supply.			¦
DC_Ramp_Edited_by_Omar	Generate a DC ramp using an incremental sequence. This program was originally created by Sergio Couttolenc and later edited by Omar Torres. Use the VLCT power supply.		¦	
Functional_Test_1	Test 1 circuit current-mirror at a fixed DC voltage. Use the VLCT power supply.		¦	¦
Functional_Test_1b	Test any quantity of circuit current-mirror at a DC fixed voltage. Use the VLCT power supply.		¦	¦
Functional_Test_2	Performs a DC Sweep in one current-mirror, reads the voltage drop in two resistors and calculates current. Provides the amount of current drawn from the power supply to the circuit		¦	¦
Functional_Test_3	Same like the above program, but this one uses an input file that can be created in a Microsoft text editor, LabView® virtual instrument or/and Unix text editor. Use the VLCT power supply.	¦	¦	¦

Table 3.2 VLCT Test programs developed for this work

In table 3.2 are all the programs developed for this work and the black boxes means that the program has the capability, a blank space means that the program can not perform the capability.

Due the lack of examples and high inconsistencies in the VLCT manuals (like wrong index, confuse text, diagrams without explanation, confuse tester channels numbers, and others), the philosophy behind the creation of the test programs was:

1. Start with the basic programming tools like the while-do cycle.
2. Create code in Pascal for D.O.S. using input and output files.
3. Export this code to the Pascal VLCT.
4. Check for any differences between versions.
5. Create basic programs in VLCT Pascal like the programs in the appendix.
6. Write a first code to setup and power up the DPS1A power supply in the VLCT.
7. Write code to verify the presence of analog channels and the differential voltmeter. Do not use the DPS1A power supply.
8. Add the DPS1A power supply statements to replace the external power supply in the previous step.
9. In a new code, add another pair of analog channels.
10. Add a while-do cycle to perform continuous testing.
11. Verify input/output data file capabilities.
12. Create a basic LabView® virtual instrument for the input data file.
13. Execute test.
14. Arrange test data.
15. Present results.

The above step is the algorithm that reflects the work performed for this project report, the one will be a base for future research.

3.7.3 Flow Diagrams for the main programs.

The main programs are:

- Powersupply6.
- DC_Ramp.
- Functional_test_1.
- Functional_test_2.
- Functional_test_3.
- Functional_test_1b (manufacturing version of functional_test_1).

The following Figures are the flow diagrams for the above programs. Each step is explained in the algorithm, and for more detailed explanation consult the appendix where the program is printed. In the appendix you will find the program with comments.

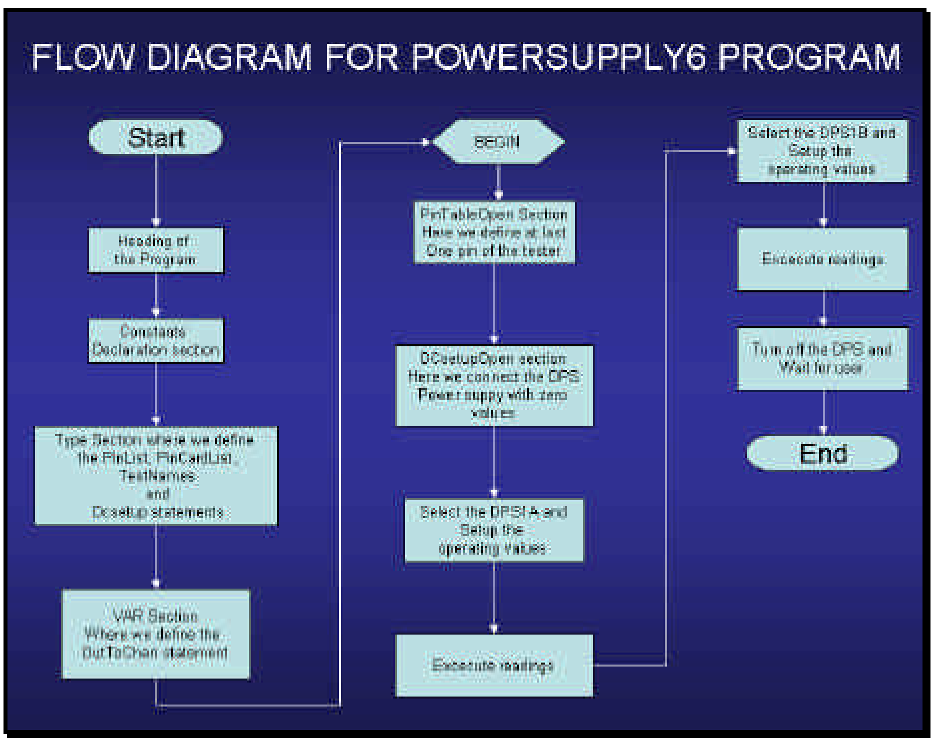


Figure 3.19 Flow Diagram for powersupply6 program

The Figure 3.19 belongs to the powersuply6 program, in this program, the user can setup the DPS1A (Device Power Supply 1A) and the DPS1B. The program only provides the code necessary for the setup.

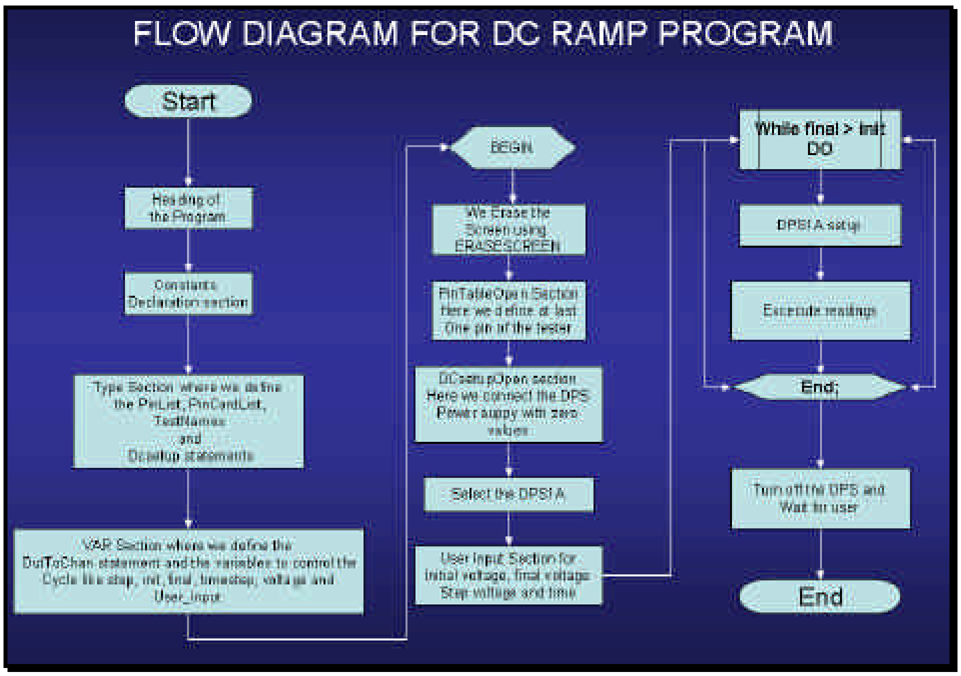


Figure 3.20 Flow Diagram for DC Ramp Program

The Figure 3.20 shows the flow diagram for the DC Ramp program, this program is used to generate an incremental value in the voltage of DPS1A, the setup of DPS1A is given by the user, and the data are: initial voltage, final voltage, step voltage and time between voltage increment steps. For more detailed information, consult the appendix where the program is printed with comments.

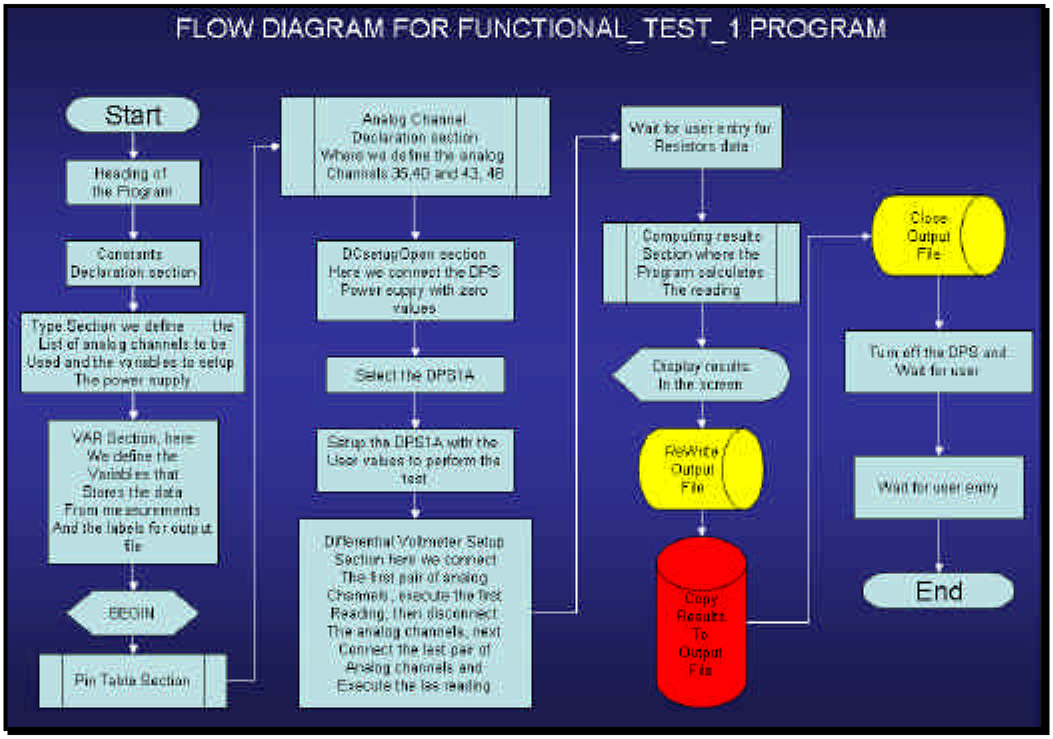


Figure 3.21 Flow Diagram for functional_test_1 program

The Figure 3.21 shows the flow diagram for the Functional_test_1 program, this program is used to generate a fixed value in the voltage of DPS1A, connect the differential voltmeter to the analog channels and take the reading of the voltage drop across the resistors. Finally, using the "Rewrite" command, the data is sent to the output file.

The Functional_test_1b is almost the same program that the exposed in Figure 3.21, but the difference is that the user can repeat the test for any number of current-mirrors using the same DPS1 voltage and the same resistor for all. The only difference in the flow diagram is the cycle Do-while. For more detailed information, consult the appendix where the programs are printed with comments.

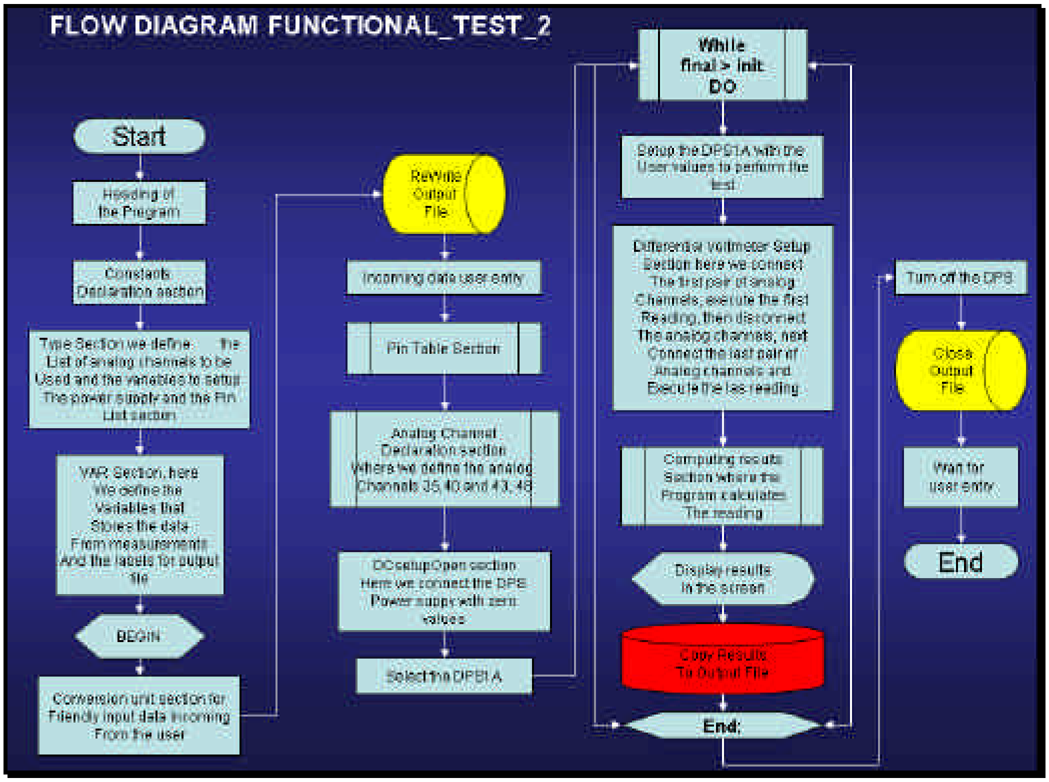


Figure 3.22 Flow Diagram for Functional_test_2 program

This program in Figure 3.22 perform almost the same operations that functional_test_1, but the difference is straightforward, the program has a cycle While-Do, that allow it to develop a DC Ramp and in each incremental step of voltage, the readings are measured. Later the readings are sent to the output data file, like in previous programs.

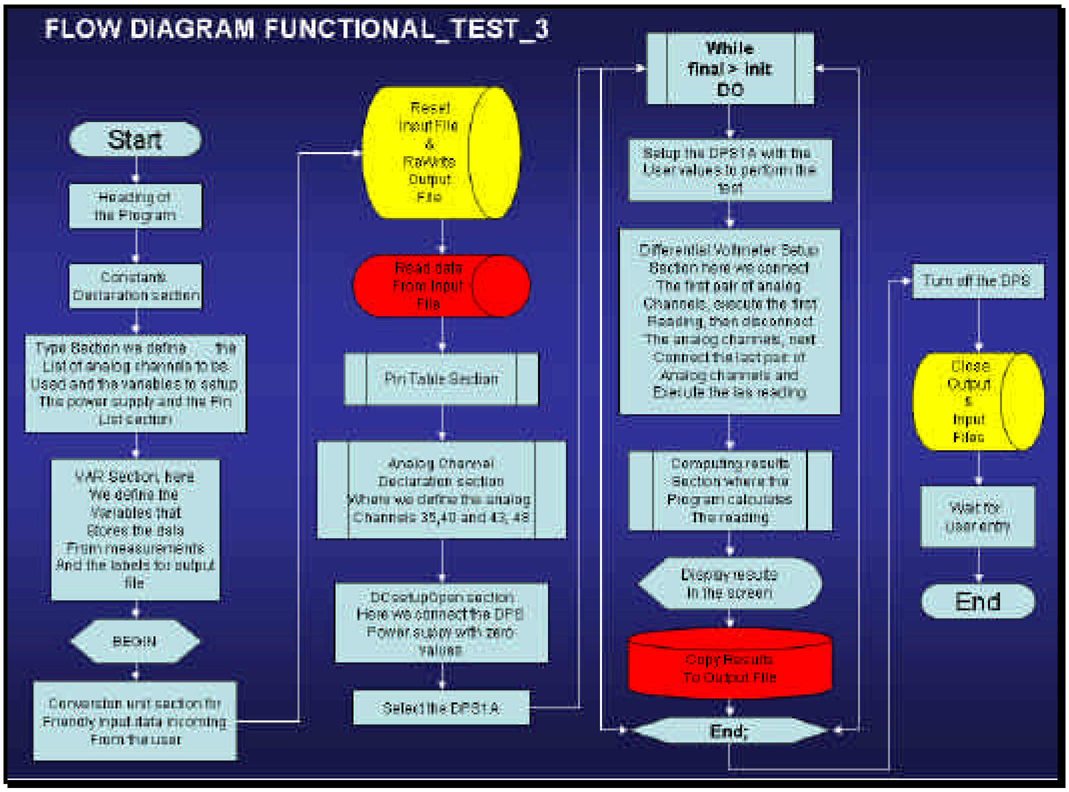


Figure 3.23 Flow Diagram for Functional_Test_3 program

This program, perform the same operations that functional_test_2, but here the difference is that the user is not needed to input data. The input data is coming from an external input data file, which can be created with a Microsoft text editor, LabView® virtual instrument or UNIX text editor. The output file can be viewed in any software platform.

3.8 Test Results

In section 3.7.1 were defined the current-mirror circuits to be the DUT. In this section we show the data from the test of these circuits using the test programs from the previous section. The raw data, without any arrange are exposed in the appendix. The following Figures and tables are the result of rearrange the data in Excel.

The following concepts are necessary to understand the test and raw data:

1. **Central value** - This is the value obtained during simulations and manual calculations.

2. **USL**. - An upper specification limit, also known as an upper spec limit, or USL, is a value below which performance of a product or process is acceptable. Upper Specific Limit: representing the maximum acceptable value of a variable (see also LSL). According with specific requirements of costumer, the USL is setup. Could be for example a 25% magnitude above the central.

3. **LSL**. - Lower specification limit is a value above which performance of a product or process is acceptable. This is also known as a lower spec limit or LSL. Lower Specific Limit: representing the minimum acceptable value of a variable. According with specific requirements of costumer, the USL is setup. Could be for example a 25% magnitude below the central.

4. **Sigma**. - The Greek letter s (sigma) refers to the standard deviation of a population. Sigma, or standard deviation, is used as a scaling factor to convert upper and lower specification limits to Z. Therefore, a process with three standard deviations between its mean and a spec limit would have a Z value of 3 and commonly would be referred to as a 3 sigma process.

5. **CPK**. - Process Capability index (equivalent), the ratio between permissible deviation, measured from the mean value to the nearest specific limit of acceptability, and the actual one-sided 3 x sigma spread of the process. As a formula, Cpk = either (USL-Mean)/ (3 x sigma) or (Mean-LSL)/ (3 x sigma) whichever is the smaller (i.e. depending on whether the shift is up or down). Note this ignores the vanishingly small probability of defects at the opposite end of the tolerance range. Cpk of at least 1.33 is desired.

6. **Yield**. - Percent of acceptable parts among all parts that are fabricated [58].

The application of those concepts to the raw data is known like the Test by itself, no test concepts is valid without the comparison between the data expected coming from simulation versus the data coming from measurements.

The above concepts will be exposed in the following Figures and tables, and they represent the data for each circuit tested.

3.8.1 DC Sweep test applied to basic bipolar current-mirror

The following Figure 3.24 is the Excel graph for the DC Sweep of the basic bipolar current-mirror. Due the format of the output data file, only voltage and miliamperes values are displayed. This graph probes that the VLCT is taking readings, and these readings are very near than the simulations previously showed.

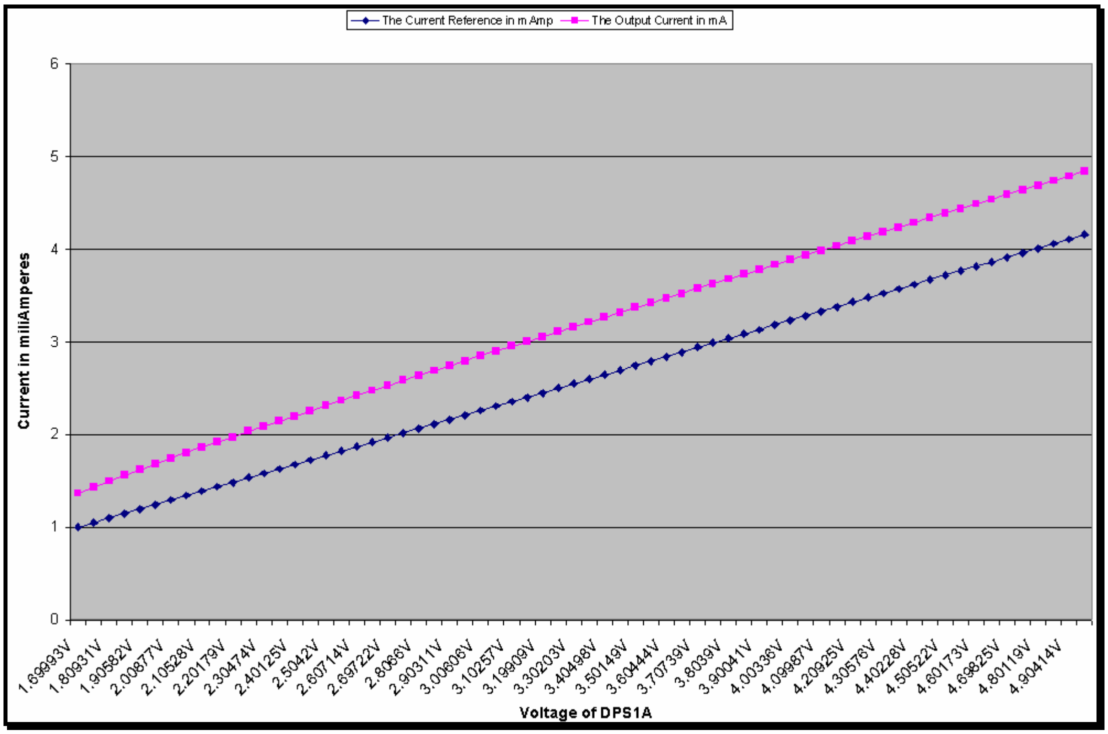


Figure 3.24 Graph for the raw data coming from the DC Sweep of basic bipolar current-mirror

This graph shows us that, in any moment, both currents are exactly the same. This is due to the voltage drop in both transistors, and the voltage drop was caused by the difference in resistance between R6 and R7.

3.8.2 DC Sweep test applied to bipolar current-mirror with base compensation

The following Figure 3.25 is the DC Sweep graph using Excel, to display the data coming from the output data file for the bipolar current-mirror with base compensation. Due the format of the output data file, only voltage and miliamperes values are displayed. Like in previous subsection, this graph probes that the VLCT is taking readings, and these readings are very near than the simulations previously showed.

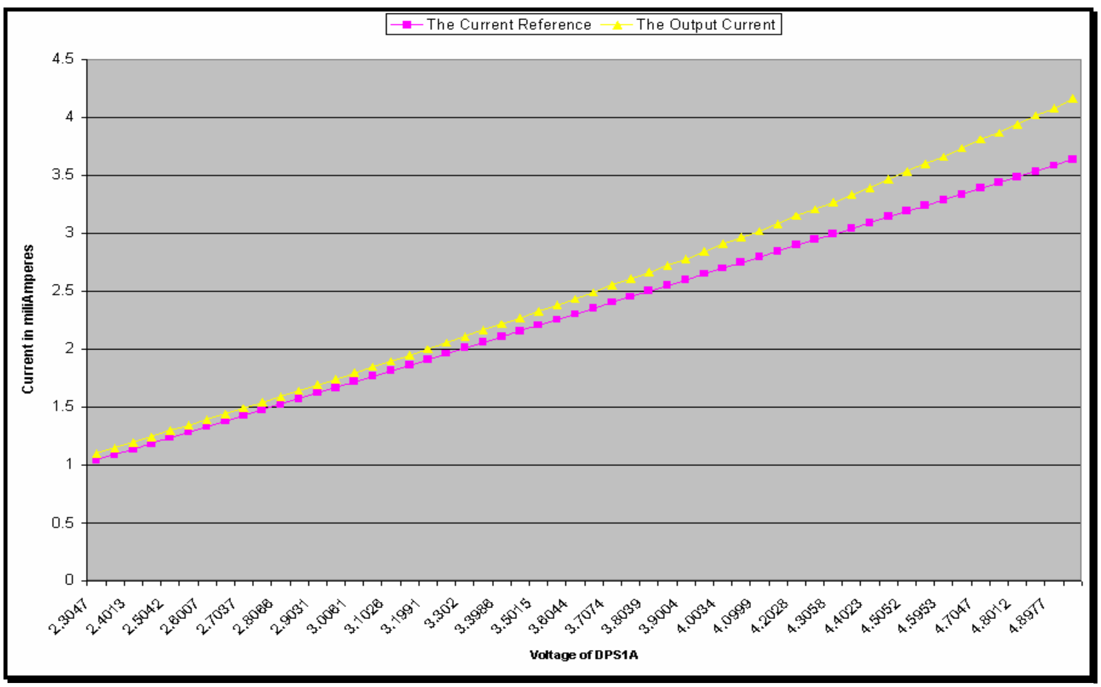


Figure 3.25 Excel Graph from the raw data coming from DC Sweep for bipolar current-mirror with base compensation.

In Figure 3.25, the behavior for both currents is quite different versus the basic current-mirror. There is a region from 2.3 volts to 3.7 volts where both current are almost equal. This conclusion matches the same conclusion given due the previous simulations.

3.8.3 DC Sweep test applied to bipolar Wilson current-mirror

According with the simulations, the bipolar Wilson current-mirror is quite superior to the other circuits. The following Figure 3.26 is the Excel graph coming from the output file for the DC Sweep performed to this circuit.

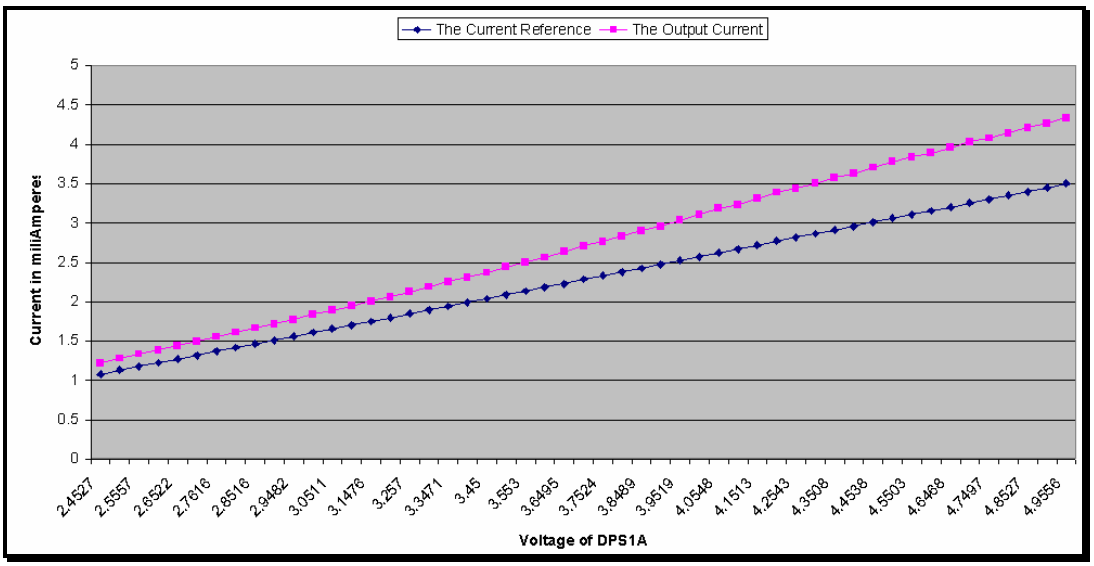


Figure 3.26 Excel Graph from raw data coming for the DC sweep for bipolar Wilson current-mirror

In Figure 3.26, both currents are almost equal below the 3 volts of DPS1A, and later they start to separate. We are neglecting the values below 2.4 volts of DPS1A because the output data coming from the VLCT is incompatible with Excel. This is against the simulation, the only explainable reason, is that the physical device BJT used in the test, is out the specifications marked by the simulation or there is an error due quality manufacturing process.

The last conclusion is valid only if we neglect problems with the measuring device or variations in room temperature and in the internal temperature of both BJT.

3.8.9 Manufacturing and Functional Test applied to basic bipolar current-mirror

The program Functional_1b, performs a basic test for drop voltage in both resistors R6 and R7, and calculates both currents. The advantage of this program is:

- The program asks for a "serial number of the DUT".
- The program has the capability to test any quantity of units.
- All the data from all units is saved in one output data file.

The test was applied to a 6 different bipolar basic current-mirrors, the transistors were different, the resistor's values weren't changed and all test performed at 3 volts. The following table is the excel worksheet coming from the raw data from the output file for this program.

Serial number of the unit is	1	2	3	4	5	6
Voltage of DPS1A is	3.00606V	3.00606V	3.00606V	3.00606V	2.99963V	3.00606V
Current from DPS1A is	21.98828MA	21.98828MA	21.98828MA	21.98828MA	21.98828MA	21.98828MA
The voltage in RREF	2.34983V	2.31198V	2.31198V	2.31511V	2.31323V	2.31323V
The voltage in ROUTPUT	336.65389MV	336.23159MV	338.62459MV	331.68021MV	339.87583MV	335.95006MV
The Current Reference	2.34983MA	2.31198MA	2.31198MA	2.31511MA	2.31323MA	2.31323MA
The Output Current	3.17598MA	3.172MA	3.19457MA	3.12906MA	3.20638MA	3.16934MA

Table 3.3 Data results from manufacturing and functional testing for basic bipolar current-mirror

The concepts of central value, LSL, USL, CPK and sigma will are applied in the following table to determine the variation of the readings from the expected values.

Now is necessary to define which will be the central readings. According with Figures 3.13, 3.15 and 3.17 the central values obtained in the simulation are:

Circuit	Reference Current	Output Current
Basic	2.332mA	2.368mA

Table 3.4 Central values

With these data, we apply our 25% criteria in the following table:

Circuit	LSL	Central value for Reference Current	USL	LSL	Central value for Output Current	ULS
Basic	1.749mA	2.332mA	2.915mA	1.776mA	2.368mA	2.96mA

Table 3.5 Upper and lower limits

Very important to note, that the 25% criteria value for upper and low limits is not a monolithic law, this criterion depends on the final application for the DUT and the client requirements. But for the scope of this project report, a 25% is a good value.

With the data given we proceed to calculate the CPK and the rest of the concepts. The units are omitted, but the values of voltages are in volts and the values for currents are in miliamperes.

Unit	The Current Reference	The Output Current
1	2.34983	3.17598
2	2.31198	3.172
3	2.31198	3.19457
4	2.31511	3.12906
5	2.31323	3.20638
6	2.31323	3.16934
LSL	1.749	1.776
USL	2.915	2.96
AVERAGE	2.319226667	3.174555
SIGMA	0.015036352	0.026529157
CPU	13.20740015	-2.695838937
CPL	12.64106891	17.57255261
CPK	12.64106891	-2.695838937

Table 3.6 Test Analysis by CPK and Sigma

According with table 3.6, we are detecting a failure. The CPK for Output Current is a negative value, and the minimum accepted value must be 1.33 and the CPU is a negative number too, because the average value coming from simulation is below the reading.

The possible solutions for this test problem are:

- Verify instruments.
- Verify test program.
- Verify the circuit.
- Open test limits.

The instrument can be neglected because there is not other instrument that can replace the VLCT. The test program has the proper values and the relay sequence is correct. The circuit can be modified or the test limits could be open.

Considering the case that the currents values given by the test are adequate for the circuit application, therefore the upper limit should be modified.

The following tables present the data with the upper limit modification.

Unit	The Current Reference	The Output Current
1	2.34983	3.17598
2	2.31198	3.172
3	2.31198	3.19457
4	2.31511	3.12906
5	2.31323	3.20638
6	2.31323	3.16934
LSL	1.749	**2.9**
USL	2.915	**3.5**
AVERAGE	2.319226667	3.174555
SIGMA	0.015036352	0.026529157
CPU	13.20740015	4.089148716
CPL	12.64106891	3.449726454
CPK	12.64106891	**3.449726454**

Table 3.7 Upper and lower limit modification

Now, the CPK are improved. Is advisable modify test limits in pairs. This is only possible when the electrical characteristics of our DUT meets the circuit application, for the contrary, a circuit modification must be done by the Design Engineer.

Chapter 4

Conclusions and future work

4.1 Conclusions

Several algorithms, diagram flows and program were developed to realize measurement and testing to current-mode circuits, and can be used to test other analog circuits. The concept of Current-mode was clarified. Providing a simple concept of operation mode and giving a new front to the actual discussion in the subject.

Current-mirrors based in Bipolar Junction Transistor were used to confirm that the VLCT is a suitable equipment for analog testing. A new technique for integration between VLCT Pascal language and LabView® was explained and expose. This technique is not explained in any VLCT user manual.

VLCT Pascal programs were developed to realize the testing of current-mirrors and can be adapted to test any other analog DC circuit. Flow diagrams for each program were explained and they are essential tools for understand the concept of testing using VLCT.

The test programs were applied to the basic bipolar current-mirror, performing hundreds of readings; those readings were used for the test example that used the CPK concept.

The objective of this research was the developing of test methods to testing of current-mode circuits using the VLCT-256, according with the above, this objective was reached.

4.2 Future work

Some lines for future work are:

- New algorithms and test methods for the VLCT and LabView® are needed in order to extend possible integrated circuits to be tested.
- Develop more LabView® virtual instruments for the VLCT, those virtual instruments could be used to better analysis for output data files or even develop VLCT Pascal code.
- The creation and continue develop of a custom-made Spice for current-mode circuits or any other, according the necessities of the research personnel.
- Creation and development of a data base that can handle the amount of data from the test performed by the VLCT.
- Develop virtual instruments that can manipulate the GPIB and DAQ and send data to the VLCT.
- Develop virtual instruments that can extract data from the VLCT and sent it to the DAQ or the GPIB.
- Develop a Calibration and validation method in hardware for the measurement devices used in this project.
- Develop a software calibration program that can be able to correct hardware errors without adjusting any hardware.
- Research in data memory storage.

BIBLIOGRAPHY

1. Bapiraju Vinnakota, *Analog and Mixed-Signal Test*, New Jersey: Prentice Hall, 1998.

2. Burns and Roberts, *Introduction to Mixed-Signal Test and Measurement*, Dallas: Texas Instruments Incorporated, 1999.

3. Test Technology Center, *VLCT ULSI/VLSI Test System Programming Guide*, Dallas: Texas Instruments Incorporated, 2001.

4. A.Mastsuzawa, "*Low-voltage and low-power circuit design for mixed analog/digital systems in portable equipment*," IEEE journal of Solid-State Circuits, vol. 29, no. 4, pp. 470-480, April 1994.

5. C.L. Wey, "Mixed-signal testing: a review," *IEEE International Conference on Electronics, Circuits and Systems*, Rodos, Greece, pp. 1064-1067, October 1996.

6. C.Toumazou, J. B. Hughes, and N. C. Battersby, *Switched-Currents: An Analogue Technique for Digital Technology*, London: Peter Peregrinus Ltd, 1993.

7. R. H. Zele and D. J. Allstot, "Low-voltage fully-diferential CMOS switched-current filters," *Proc. of IEEE Custom Integrated Circuits Conference*, pp. 6.2.1-6.2.4, 1993.

8. J.B. Hughs and K. W. Moulding, "S^2I: A switched-current technique for high performance," *Electronic Letters*, vol. 29, no. 16, pp. 1400-1401, August 1993.

9. C. –P. Wang and C. L. Wey, "Test generation of analog switched-current circuits," *Proc. Asian Test Symposiums*, Taiwan, pp. 376-381, November 1996.

10. T. Olbrich and A. Richardson, "Design and self-test for switched-current building blocks," IEEE Design & Test of Computers, pp. 10-17, summer 1996.

11. G. Taylor, P. Wrighton, C. Toumazou, and N. Battersby, "Mixed-signal testing considerations for switched current signal processing," Proc. Midwest Symposiums on Circuits and Systems, Washington D. C., August 1992.

12. Waveform Measurement and Analysis Technical Committee of the IEEE Instrumentation and Measurement Society, "IEEE Std 1241-2000 For Terminology and Test Methods for Analog-to-Digital Converters", *IEEE-SA Standards Board*, December 2000.

13. Microsystems Electronics Subcommittee of the IEEE Solid-State Devices Committee, "IEEE Std 274-1966 Reaffirmed 1980 Definitions of Terms for Integrated Electronics", *IEEE-SA Standards Board*, March 1980.

14. Design Automation Standards Committee of the IEEE Computer Society, "IEEE Std 1481-1999 for Integrated Circuit (IC) Delay and Power Calculation System", *IEEE-SA Standards Board*, June 1999.

15. Design Automation Standards Committee, "IEEE Std 1076.4-2000 (Revision of IEEE Std 1076.4-1995)", *IEEE-SA Standards Board*, September 2000.

16. Test Technology Technical Committee of the IEEE Computer Society, “IEEE Std 1149.4-1999 for a Mixed-Signal Test Bus”, *IEEE-SA Standards Board*, June 1999.

17. David Harriz and David Diaz, “TestosterICs: A Low-Cost Functional Chip Tester”, *Proceedings of the 2003 IEEE International Conference on Microelectronics Systems Education*, 2003.

18. R. Huang, “High-performance CMOS Switched-Current Circuits for Low-Voltage Analog Signal Processing Applications”, *Ph.D. Dissertation*, Department of Electrical Engineering, Michigan State University, August 1997.

19. Sooryong Lee, Brad Cobb, Jennifer Dworak, Michael R. Grimaila and M. Ray Mercer: “A New ATPG Algorithm to Limit Test Set Size and Achieve Multiple Detections of all Faults”, *Proceedings of the 2002 Design, Automation and Test in Europe Conference and Exhibition*, Department of Electrical Engineering Computer Engineering Group, Texas A&M University, 2002 .

20. L.M. Patnaik, V.K. Agrawal and B.K.S.V.L.Varaprasad, The State of VLSI Testing, “IEEE Potentials”, IEEE September 2002.

21. C.L. Wey, “Concurrent error detection in current-mode A/D converter,” *Electronic Letters*, vol. 27, no. 25, pp. 2370-2372, December 1991.

22. Lous Y. Ungar, Tony Ambler, “Economics of Buit-in Self-Test”, International Test Conference, *Design & Test of Computers*, IEEE, October 2001.

23. Gert Jervan, Zebo Peng, Raimund Ubar, Helena Kruss, "A Hybrid BIST Architecture and its Optimization for SoC Testing", *Proceedings of the International Symposium on Quality Electronic Design*, IEEE 2002.

24. S. Sahli, Test Generation and Concurrent Error Detection in Current-Mode A/D Converteres, *M. S. Thesis*, Department of Electrical Engineering, Michigan State University, May 1992.

25. S. Krishnan, S. Sahli, and C. L. Wey, "Test generation and concurrent error detection in current-mode A/D converters," *Proc. IEEE International Test Conference*, Baltimore, MD., pp. 312-320, September 1992.

26. J. Velasco-Medinay, I. Rayane and M. Nicolaidis, "On-Line BIST for Testing Analog Circuits", *Reliable Integrated Systems Group, TIMA/INPG*, France, IEEE, 2000.

27. P. Pawlowski, A. Guzinski, "Comparative Investigations of Substrate Noise caused by voltage-mode and current-mode gates", *Technical University of Koszalin Poland, ISCAS 2000 - IEEE International Symposium on Circuits and Systems*, 2000.

28. Saman Adham, Dhamin al-kahalili, Come Rozon, Douglas Racz, "Comprehensive Defect Analisys and Testability of Current-mode logic circuits", *ISCAS 2000 - IEEE International Symposium on Circuits and Systems*, May 28-31, 2000.

29. Takahiro HANYU, Tsukasa IKE and MIchitaka KAMEYAMA, "Integration of Asynchronous and Self-Checking Multiple Valued Current-Mode Circuits", *Tohoku University*, IEEE, 2000.

30. Come Rozon, Dhamin Al-khalili, Saman Adham and Douglas Racz, "Comparing Defect Coverage for Current-Mode Logic and CMOS VLSI Cells", *Royal military College, Logic Vision & Nortel Networks*, IEEE, 2000.

31. Fei Yuan, "Statistical Analysis of Nonlinear Current-mode circuits", *VLSI Research Group Ryerson University*, IEEE, 2000

32. J.A. Bell, J.W. Bruce & B.J. Blalock, "CMOS Current-Mode Flash Analog to Digital Converter", *Mississipi State University*, IEEE, 2001.

33. Rochit rajsuman, "Extending EDA Environment From Design to Test", *Proceedings of the 11Th Asian Test Symposium*, IEEE, 2002.

34. Andreas Veneris, Alexander Smith and Magdy S. Abadir, "Logic Verification Based on Diagnosis Techniques", *University of Toronto and Motorola*, IEEE, 2003.

35. Hak-soo Yu, Jacob A. Abraham, Sungbae Hwang, jeongjin Roh, "Efficient loop-back testing of on-chip ADC's and DAC's", *University of Texas and University of Korea*, IEEE, 2003.

36. Youhua Shi and Zhe Zhang, "Multiple Test Set Generation Method for LFSR-Based BIST", *Waseda University and Southeast University*, IEEE, 2003.

37. C.V. Guru Rao, D.Roy Chowdhury., “A New Design-for-Test Technique for Reducing SOC Test Time”, *Indian Institute of Technology*, IEEE, 2003.

38. Test Technology Standards Committee, “IEEE STD 1149.6 for Boundary-Scan Testing of Advanced Digital Networks”, *Sponsored by the IEEE Computer Society*, IEEE, April 2003.

39. Amit K. Varshney, Bapiraju Vinnakota, Erick Skuldt and Brion Keller, “High Performance Parallel Fault Simulation”, *Intel Corporation, University of Minnesota & IBM Corporation*, IEEE, 2001.

40. O. Omeni, C. Toumazou, “A CMOS Micro-power Wideband Data/Power Transfer System for Biomedical Implants”, *Imperial College London*, IEEE, 2003.

41. Apisak Worapishet, John B. Hughes and Chris Toumazou, “Speed and Accuracy Enhancement Techniques for High-Performance Switched-Current Comparators”, *Journal of Solid-State Circuits, vol. 36, no. 4*, IEEE, April, 2001.

42. Jirayuth Mahattanakul and Chris Toumazou, “Current-Mode Versus Voltage-Mode”, *IEEE transactions on circuits and systems, vol. 45, no. 2*, February, 1998.

43. Ashley, Kenneth L., “Analog electronics with LabView”, National Instruments & Pearson Education Inc., *National Instruments virtual instrumentations series*, Prentice Hall, 2003, ISBN: 013047065-1.

44. Test Technology Standards Committee, "IEEE STD 1149.1-2001 for Test Access Port and Boundary-Scan Architecture", *Sponsored by the IEEE Computer Society*, IEEE, June 2001.

45. Doug Mirizzi, "Very Low Cost Testers (VLCT) Programming and Applications", *Test Technology Center*, Texas Instruments Incorporated.

46. Marc E. Herniter, "Schematic Capture Using Microsim Pspice for Windows 95/98/NT", *Prentice Hall*, 2000, ISBN: 0-13-081404-0.

47. A. Bruce Buckman, "Computer-Based Electronic Measurement", *National Instruments & Prentice Hall*, 2000, ISBN: 0-201-36182-5.

48. Nell Dale & Chip Weems, "Turbo Pascal", *D.C. Heath and Company*, 1988, ISBN: 0-669-14591-2.

49. Robert Boylestad & Louis Nashelsky, "*Electronica Teoria de Circuitos (Spanish edition)*", Prentice Hall, 1989, ISBN: 968-880-154-2.

50. Paul R. Gray & Robert G. Meyer, "*Analysis and Design of Analog Integrated Circuits*", John Wiley & Sons, 1993, ISBN: 0-471-57495-3.

51. Ronald W. Larsen, "*Introduction to MathCAD*", Prentice Hall, 1999, ISBN: 0-13-937493-0.

52. Don Lancaster, "*CMOS Cookbook*", Newnes, 1997, ISBN: 0-7506-9943-4.

53. Walter G. Jung, "*IC Op-Amp Cookbook*", Prentice Hall, 1997, ISBN: 0-13-889601-1.

54. Stephen Wolfram, “*The Mathematica Book*”, Wolfram Media & Cambridge University Press, 1999, ISBN: 1-57995-004-5 & 0-521-64314-7.

55. Adel S. Sedra & Kenneth C. Smith, “*Microelectronics Circuits*”, Oxford University Press, 1998, ISBN: 0-19-511-663-1.

56. Hanspeter Schmid, “Why the terms Current-mode and Voltage-mode neither divide nor qualify circuits”, Switzerland. IEEE 2002.

57. C. Toumazou & F. J. Lidgey & D. G. Haigh, “Analogue IC Design, the Current-mode approach”. *Peter Peregrinus LTD*, 1998. ISBN: 0-86341-297-1

58. Michael L. Bushnell & Vishwani D. Agrawal, “Essentials of Electronic Testing”. Kluwer Academic Publishers. Lucent Technologies 2000. ISBN: 0792379918.

Appendix "A"

Basic Bipolar Current-mirror, introductory program

```
Program mirror01 (Output, entrada, salida);
(* heading of the program            *)
(* declaration of input output       *)



(* *************************************************** *)
(*                                   *)
(* This program calculates several parameters of      *)
(* current mirrors. Not Dynamic mode                  *)
(* using readln for read data on input file          *)
(* this version uses arrays to read strings!!!       *)
(* by Sergio Couttolenc                        *)
(* Date May 31 2004                            *)
(*                                   *)
(* *************************************************** *)
(*                                   *)
(* The input data file is DAT_IN5.DAT                *)
(* The output data file is DAT_OUT5.DAT              *)
(*                                   *)
(* *************************************************** *)
(*                                   *)
(* This program calculates:                    *)
(*                                   *)
(* Resistance reference                       *)
(* Output Current                             *)
(* Voltage reference                          *)
(* Output voltage                             *)
(*                                   *)
(* Given the following data:                  *)
(*                                   *)
(* VCC, IREF, VBE, BETA, VA                   *)
(* *************************************************** *)


CONST
   (* Constants declarations section *)
```

```
  A = 2;

  (* this constant is not used at this moment *)


TYPE
  (* variable custom types *)


  (* here we define the chain of chars to be *)
  (* used to read the labels of data in the  *)
  (* input file                              *)


  Dummy_line_label = String[8];  (* used for markers      *)
  User_Name_label  = String[6];  (* used for user name    *)
  Clase_label      = String[6];   (* to choose the circuit *)
  Ckto_label       = String[1];   (* to choose the circuit *)
  Early_label      = String[4];   (* for Va label          *)
  Voltage_label    = String[3];   (* for Vcc label         *)
  Beta_label       = String[4];   (* for beta label        *)
  Tran_label       = String[3];   (* for choose npn or pnp *)
  Iref_label       = String[4];   (* for Iref              *)


VAR
  (* Variable declarations section            *)
  (* now the variables used to read, store    *)
  (* and process the data                     *)

  VCC:Real;       (* to store vcc value        *)
  IREF:Real;      (* to store Iref             *)
  VBE:Real;       (* to store Vbe              *)
  BETA:Real;      (* to store Beta             *)
  VEARLY:Real;    (* to store early voltage    *)
  (* *)

  RES, IOUTPUT, VREF, VOUTPUT:Real;
```

```
(* The following  variables are for          *)
(* easy calculations porpuses                *)

Process_1, Process_2, Process_3, Process_4:Real;
Process_1b, Process_1c, Process_1d, Process_1e:Real;
(* *)



(* now the variables for program control     *)

tecla_1, tecla_2:Char;
(* *)




(* now the variables for i/o file link       *)

entrada, salida:Text;
(* *)





(* now custom variable for label             *)

Label_dummy:Dummy_line_label;
Label_user:User_Name_label;
Label_clase:Clase_label;
Label_ckto:Ckto_label;
Label_early:Early_label;
Label_vcc:Voltage_label;
Label_beta:Beta_label;
Label_tran:Tran_label;
Label_iref:Iref_label;
```

```
(*  !!!!!THE EXECUTABLE CODE STARTS HERE!!!!!   *)

BEGIN (* *****STARTS THE EXECUTABLE CODE*****   *)


Writeln('Programa listo para ejecutar presione cualquier tecla');
Read(tecla_1);




(* preparing files *)


Assign(entrada, 'dat_in5.dat');
(* entrada is linked to dat_in5.dat file *)

Assign(salida, 'dat_out5.dat');
(* salida is linked to dat_out5.dat file *)

Writeln('.....Input/Output files linked ');

(* now we open the files *)
Reset(entrada);
(* entrada is set to be readed *)

Rewrite(salida);
(* salida is set to be written *)

Writeln('.....I/O files ready to RW ');




(* Now we read the data from input file *)

Readln(entrada, VCC);      (* catch the vcc label and value *)
Readln(entrada, VBE);
Readln(entrada, BETA);    (* for beta *)
Readln(entrada, VEARLY); (* for early voltage Va *)
Readln(entrada, IREF);    (* for input curr reference *)
```

```
Readln(entrada, Label_tran);          (* for transistor type *)



(* Then we process the data *)

RES:= ((VCC-VBE)/IREF);

Process_1b:= (2/BETA);

Process_1c:= (Process_1b + 1);

Process_1d:= (1/Process_1c);

Process_1e:= (Process_1d * IREF);

IOUTPUT:= Process_1e;

VREF:= (VCC-VBE);

Process_1:= (IOUTPUT * Process_1c);

Process_2:= (Process_1/IREF);

Process_3:= (Process_2-1);

Process_4:= (Process_3*VEARLY);

VOUTPUT:= (VBE+Process_4);



(* We have the results then we are going to write them in the file  *)
(* Note that the 10 is the power on scientific notation             *)
(* Then 10:2 means number by 10 to power 1 with 2 decimal positions *)


Writeln(salida, 'Results: ');
Writeln(salida);
Writeln(salida, 'Reference Resistor: ', RES, ' Ohms');
Writeln(salida, 'Output Current:     ', IOUTPUT, ' Amps');
```

```
Writeln(salida, 'Voltage Reference:  ', VREF, ' Volts');
Writeln(salida, 'Output Voltage:     ', VOUTPUT, ' Volts');


Writeln(salida);
Writeln(salida, 'program created by sergio couttolenc');
Writeln(salida, 'using Pascal Language programing for DOS ');
Writeln(salida);
Writeln(salida);
Writeln;
Writeln('Job Done');
Writeln('Please Review output file "dat_out.dat" ');
Writeln;
Writeln('Press any key to finish');
Readln(tecla_2);

(* now we close the files *)

Close(entrada);
Close(salida)
(* check there is not puntuation here *)

END.
```

DO-WHILE EXAMPLE PROGRAM

```
Program example1;
var
innum: integer;

begin
        read(innum);
        while(innum<>99)do
                begin
                case innum of
                1: writeln('goodbye');
                2: writeln('hello');
                3: writeln('no good');
                4: writeln('no say');
                otherwise writeln('do over');
                end;
        read(innum);
end;
pause;
end.
```

POWERSUPPLY6 PROGRAM

```
program powersupply_6;

TYPE
PinList = (PIN1);
PinCardList = (PC1);
TestNames = (Test1);
DCSetup = (DCSetup1);

VAR


DutToChan
1:1;

begin

PinTableOpen;
PinSet(PIN1, 1);
PinCardSet(PC1, S_PINCARD1);
PinTableClose;

DCSetupOpen(DCSetup1);
DCSet(0v, 0v, 0v, 0V, 0ma, 0ma, PC1);
DCConnect(PIN1, S_LOW, S_OPEN);
```

```
DCSetupClose;

writeln('Example of How to Setup the DPS');
{next we select the DC}
SetupSelect(DCSetup1);
{next we setup the voltage and current limit for DPS1A}
SupplySet(S_DPS1A, 3V, 50mA, S_IGAIN_1X, S_HIGH_CURR_OVERRIDE_BIAS);
wait(50mS);
{waiting for stabilice the readings}

{now we take the reading and output to screen}

Writeln('voltage on 1A is ', SupplyRead(S_DPS1A, S_VOLTAGE));
writeln('current on 1A is ', SupplyRead(S_DPS1A, S_CURRENT));
writeln('Press any key to shutdown DPS1A');

pause;

{now we turn off the DPS1A}

SupplySet(S_DPS1A, 0V, 1ma, S_IGAIN_1X, S_HIGH_CURR_OVERRIDE_BIAS);
SupplyDisconnect(S_DPS1A);

wait(50ms);

writeln('DPS1A is off');
writeln('Voltage in 1A is ', SupplyRead(S_DPS1A, S_VOLTAGE));

{next we setup the DPS1B}

writeln('press any key to setup DPS1B');

pause;

SupplySet(S_DPS1B, 6.5V, 50mA, S_IGAIN_1X, S_HIGH_CURR_OVERRIDE_BIAS);

wait(50mS);

{readings}

Writeln('voltage on 1B is ', SupplyRead(S_DPS1B, S_VOLTAGE));
writeln('current on 1B is ', SupplyRead(S_DPS1B, S_CURRENT));
writeln('Press any key to shutdown DPS1B');

pause;


SupplySet(S_DPS1B, 0V, 1ma, S_IGAIN_1X, S_HIGH_CURR_OVERRIDE_BIAS);
SupplyDisconnect(S_DPS1B);


end.
```

PROGRAM ANALOG1

```
program analog_1;
{Program that measures the voltage between analog channels 35 and 40}
{where channel 35 is the positive side and ch 40 is the negative    }
{created by sergio couttolenc                                        }


TYPE
{here we define the list of channels      }
{because they are a user defined variable }
AnalogChanList=(POS_35, NEG_40, Analog_1);

VAR
{This is the variable that stores the measurement}
V_Meas:Treal;


begin
{the proccess start here}

AnalogChanTableOpen;
{begin the channel declarations}
        AnalogChanSet(POS_35, S_Analog, 35); {initialize ch 35 as POS_35}
        AnalogChanSet(NEG_40, S_Analog, 40); {initialize ch 40 as NEG_40}
        AnalogChanListSet(Analog_1, POS_35, NEG_40);
        {in the upper line we define both pines in Analog_1}
AnalogChanTableClose; {End the Channel declarations}

{now we connect the pines}
AnalogSetConnect(POS_35, S_Diff_Pos); {connect 35 as Diff_Pos}
AnalogSetConnect(NEG_40, S_Diff_Neg); {connect 40 as Diff_Neg}

{new we choose the range}
pause;
AnalogSet(POS_35, S_Diff, 10V); {10 volts range}

{now we initialize the differential voltage read}
AnalogReadInit(POS_35, S_Diff, false); {boolean is false for DC}

{we execute the reading}
V_Meas:= AnalogRead;

writeln(' The voltage is: ', V_Meas);  {display meas in the screen}

pause;

{here program ends}
end.
```

PROGRAM MIRROR_1

```
program Mirror_1;


{ * This Program measures the voltages in two resistors      * }
{ * in one basic current-mirror, then calculates the current * }
{ * passing throug the resistors.                            * }
{ * One current is Iref and the other is Ioutput             * }
{                                                            * }
{ * Program Created by Sergio Couttolenc Valdes              * }
{ * June 2004                                                * }



TYPE
{here we define the list of channels                          * }
{because they are a user defined variable                     * }
{POS_35 and NEG_40 are the first Diff-Volt used to measure   * }
{voltage drop in Resistor reference in collector Q1          * }
{POS_43 and NEG_48 are the second Diff-Volt used to measure  * }
{voltage drop in Load Resistor in Collector Q2               * }


AnalogChanList=(POS_35, NEG_40, POS_43, NEG_48, Analog_1);



VAR
{This is the variable that stores the measurement}

Ohm_Ref:Treal;  { used to store the value in ohms of RREF }
Ohm_Output:Treal;       { To store ohms of ROUTPUT       }
V_REF:Treal;            { To store the meas voltage      }
V_Output:Treal;         { To store the meas voltage      }
I_REF:Treal;            { To store the Current reference  }
I_Out:Treal;            { To Store the Output Current     }



begin
{the proccess start here}

AnalogChanTableOpen;
{begin the channel declarations}

        AnalogChanSet(POS_35, S_Analog, 35); {initialize ch 35 as POS_35}
        AnalogChanSet(NEG_40, S_Analog, 40); {initialize ch 40 as NEG_40}
        AnalogChanSet(POS_43, S_Analog, 43); {initialize ch 35 as POS_43}
        AnalogChanSet(NEG_48, S_Analog, 48); {initialize ch 40 as NEG_48}
```

```
    AnalogChanListSet(Analog_1, POS_35, NEG_40, POS_43, NEG_48);
    {in the upper line we define both pines in Analog_1}

AnalogChanTableClose; {End the Channel declarations}


{now we connect the pines for measure voltage drop in RREF}
AnalogSetConnect(POS_35, S_Diff_Pos); {connect 35 as Diff_Pos}
AnalogSetConnect(NEG_40, S_Diff_Neg); {connect 40 as Diff_Neg}


{new we choose the range}
pause;
AnalogSet(POS_35, S_Diff, 10V); {10 volts range}


{now we initialize the differential voltage read}
AnalogReadInit(POS_35, S_Diff, false); {boolean is false for DC}


{we execute the reading}
V_REF:= AnalogRead;  { Stores the voltage Drop of RREF }


{now we disconnect the pines for measure voltage drop in RREF}
AnalogSetConnect(POS_35, S_OPEN); {disconnect 35 as Diff_Pos}
AnalogSetConnect(NEG_40, S_OPEN); {disconnect 40 as Diff_Neg}




{now we connect the pines for measure voltage drop in ROUTPUT}
AnalogSetConnect(POS_43, S_Diff_Pos); {connect 43 as Diff_Pos}
AnalogSetConnect(NEG_48, S_Diff_Neg); {connect 48 as Diff_Neg}


{new we choose the range}
pause;
AnalogSet(POS_43, S_Diff, 500mV); {500mV volts range}


{ now we initialize the differential voltage read}
AnalogReadInit(POS_43, S_Diff, false); {boolean is false for DC}


{we execute the reading}
V_Output:= AnalogRead;  { Stores the voltage Drop of ROUTPUT }


{ Now we ask the resistance values for each resistor load }
```

```
Writeln(' Resistance in ohms of RREF ? ');
Readln(Ohm_Ref);

Writeln(' Resistance in ohms of ROUTPUT ? ');
Readln(Ohm_Output);


{ Now we calculate the currents }

I_REF:= (V_REF/Ohm_Ref);

I_Out:= (V_Output/Ohm_Output);


{ Display the results }


writeln(' The voltage in RREF: ', V_REF);  {display meas in the screen}

writeln(' The voltage in ROUTPUT: ', V_Output);

writeln(' The Current Reference: ', I_REF);

writeln(' The Output Current: ', I_Out);


pause;

{here program ends}
end.
```

PROGRAM SHORTCKT_TEST_1

```
program shortckt_test_1;


{Short Circuit Test Program, from VCC pin to Ground Pin            }
{This program setup the DC power supply at 3 volts and measures }
{the output current.                                               }
{In order to pass, the measured current must be between the test}
{limits.                                                   }
{program created by sergio couttolenc                              }



TYPE
{here we define the list of channels     }
{because they are a user defined variable }

PinList = (PIN1);
PinCardList = (PC1);
TestNames = (Test1);
DCSetup = (DCSetup1);


VAR
UnitNumber:Treal;          {used to store the measure}
Output_File:Text; {output data file}
Meas:Treal;
FAILURE:STRING[6];
SUCCESS:STRING[6];
Test_Result:STRING[6];


DutToChan
1:1;



begin
{the proccess start here}

PinTableOpen;
PinSet(PIN1, 1);
PinCardSet(PC1, S_PINCARD1);
PinTableClose;

DCSetupOpen(DCSetup1);
DCSet(0v, 0v, 0v, 0V, 0ma, 0ma, PC1);
DCConnect(PIN1, S_LOW, S_OPEN);
DCSetupClose;
```

```
{here we manage the status labels}
FAILURE:='FAIL!!';
SUCCESS:='PASS!!';


SetupSelect(DCSetup1);    {select and setup of the power supply}
SupplySet(S_DPS1A, 3V, 50mA, S_IGAIN_1X, S_HIGH_CURR_OVERRIDE_BIAS);
wait(50mS); {waiting for stabilice the readings}



{now we take the reading and output to screen}

Writeln('voltage on Power Supply 1A is ', SupplyRead(S_DPS1A, S_VOLTAGE));
writeln('current on Power Supply 1A is ', SupplyRead(S_DPS1A, S_CURRENT));
writeln;
{display meas in the screen}


{Now We execute the test}

Meas:=SupplyRead(S_DPS1A, S_CURRENT); {value of supplyread stored in meas}

{here 25MA is the upper limit}
IF(Meas > 25MA) THEN
Test_Result:=FAILURE
ELSE
Test_Result:=SUCCESS;


pause;

{now the data is stored in the file}
Rewrite(Output_File, 'Sct_1.dat');   {output file is ready}
Writeln(Output_File, 'Short Circuit Test 1');
Writeln(Output_File);
Writeln(Output_File, SupplyRead(S_DPS1A, S_VOLTAGE)); {write voltage}
Writeln(Output_File, SupplyRead(S_DPS1A, S_CURRENT)); {write current}
Writeln(Output_File, 'STATUS:', Test_Result);   {writes the result of the test}

Writeln('Data stored in Scdt.dat file');

{now we turn off the DPS1A}

SupplySet(S_DPS1A, 0V, 1ma, S_IGAIN_1X, S_HIGH_CURR_OVERRIDE_BIAS);
SupplyDisconnect(S_DPS1A);

wait(50ms);

writeln('DPS1A is off');
writeln('Test ended');
```

```
pause;

{now we close the files}
FileClose(Output_File);

{here program ends}
end.
```

PROGRAM DC RAMP

```
program DC_Ramp;

{THIS PROGRAM IS INTENDED TO BE USED AS A DC RAMP GENERATOR}
{FOR MIRROR-CURRENT ANALYSIS}
{CODE BY SERGIO COUTTOLENC 27 OCT 2003}

{*************************************************
MAJOR REVISIONS:

=> MAJOR REVISIONS TABLE
   BY: OMAR A. TORRES
   DATE: NOVIEMBRE 7, 2003
   DESCRIPTION:
           The purpose of this table is to keep track of
           program updates.

=> REAL -> TREAL CONVERSION TABLE
   BY: OMAR A. TORRES
   DATE: NOVIEMBRE 7, 2003
   DESCRIPTION:
           The purpose is to eliminate the nessecity
           of UNITS input by the user. Now the program
           is less sensitive to errors by the user typos.
**************************************************}

TYPE
  PinList = (PIN1);
  PinCardList = (PC1);
  TestNames=(Test1);
  DCSetup=(DCSetup1);

VAR
{variable definition for the DC ramp, using TREAL types !!}
 step, init, final, timestep, V:treal;
{timestep is the variable to be used for time between steps}
{step, is the variable for the space between voltage changes}
{init, is the variable for initial voltage}
{final, is the variable for final voltage}
{timestep, is the variable for time between voltage changes}
{V, is the variable that goes into the DC PS}

TIME_UNIT, VOLTAGE_UNIT, CURRENT_UNIT :TREAL;

USER_INPUT:real;

DutToChan
  1:1;

begin
```

```
ERASESCREEN;
{***************************************************}
{           REAL-> TREAL CONVERSION TABLE                    }
TIME_UNIT:=1S;                          {Time Base Unit-> Seconds}
VOLTAGE_UNIT:=1V;   {Voltage Base Unit-> Voltage}
CURRENT_UNIT:=1A;   {Current Base Unit-> Ampere}
{***************************************************}
  PinTableOpen;
    PinSet(PIN1, 1);
    PinCardSet(PC1, S_PINCARD1);
  PinTableClose;

  DCSetupOpen(DCSetup1);
    DCSet(0v,0v,0v,0v,0ma,0ma, PC1);
    DCConnect(PIN1, S_LOW, S_OPEN);
  DCSetupClose;

  writeln('');

    {Next we select the DC}

    SetupSelect(DCSetup1);

{here we start the program}
writeln('**** PROGRAM FOR SETUP A DC RAMP ****');
{
            writeln('');
            writeln('THIS PROGRAM SUPPORTS UNITS');
            writeln('');
            writeln('TO INPUT SOME VOLTAGE USE THIS FORM: 5V INSTEAD OF 5');
            writeln('');
            writeln('TO INPUT SOME TIME USE THIS FORM: 0.5S INSTEAD OF 500mS');
}
pause;




writeln('please input the initial voltage value for DC ramp');
readln(USER_INPUT);
init:= USER_INPUT*VOLTAGE_UNIT;

WRITELN('USER_INPUT: ', USER_INPUT, ' INIT: ', init);

writeln('please input the final voltage value for DC ramp');
readln(USER_INPUT);
final:=USER_INPUT*VOLTAGE_UNIT;

WRITELN('USER_INPUT: ', USER_INPUT, ' FINAL: ', FINAL);

writeln('please input the step voltage value for DC ramp');
```

```
readln(USER_INPUT);
step:=USER_INPUT*VOLTAGE_UNIT;

WRITELN('USER_INPUT: ', USER_INPUT, ' STEP: ', STEP);

writeln('please input the time between steps in Seconds');
readln(USER_INPUT);
timestep:=USER_INPUT*TIME_UNIT;

WRITELN('USER_INPUT: ', USER_INPUT, ' TIMESTEP: ', TIMESTEP);

writeln('the program is ready to run !!');
pause;


V:=init;



While(final > init) DO
begin   {CYCLE STARTS HERE!!!}
            writeln(V);
            writeln('ramp in progress!!!');
            writeln('');

            {next we setup the voltage and current limit for DSP1A};

            SupplySet(S_DPS1A, V, 500ma, S_IGAIN_1X, S_HIGH_CURR_OVERRIDE_BIAS);
            wait(timestep); {time to change to a new step}
            writeln('Voltage on 1A is ',SupplyRead(S_DPS1A, S_VOLTAGE));
            V:= V + step; {here we do the increment of voltage for dc ramp};
            init:=V;


end;        {CYCLE ENDS HERE !!!!}



    {Now we turn off the DSP1A}
     SupplySet(S_DPS1A,  0v, 1ma, S_IGAIN_1X, S_HIGH_CURR_OVERRIDE_BIAS);
     SupplyDisconnect(S_DPS1A);
     wait(50ms);
     writeln('Now DPS1A is Off');
     writeln('Voltage on 1A is ',SupplyRead(S_DPS1A, S_VOLTAGE));


   pause;

end.
```

PROGRAM FUNCTIONAL_TEST_1

```
program Functional_Test_1;

{*********************************************************** }
{ * Functional Bipolar Current-Mirror Test program          * }
{ * Measures voltage in both resistors                      * }
{ * Calculate the current passing through                   * }
{ * Execute the Test and writes result in output file       * }
{ * On this version input file is not used                  * }
{ * This version uses the vlct power supply                 * }
{ * Program Created by Sergio Couttolenc Valdes             * }
{ * June 2004                                               * }
{*********************************************************** }


{********************** TYPE SECTION ************************}

TYPE
{here we define the list of channels                        * }
{because they are a user defined variable                   * }
{POS_35 and NEG_40 are the first Diff-Volt used to measure   * }
{voltage drop in Resistor reference in collector Q1         * }
{POS_43 and NEG_48 are the second Diff-Volt used to measure  * }
{voltage drop in Load Resistor in Collector Q2              * }

AnalogChanList=(POS_35, NEG_40, POS_43, NEG_48, Analog_1);

{now the variables to setup the power supply}
PinList = (PIN1);
PinCardList = (PC1);
TestNames = (Test1);
DCSetup = (DCSetup1);
{************************************************************}



{*************** VARIABLES DECLARATION SECTION  ***************}

VAR
{This is the variable that stores the measurement}
Ohm_Ref:Treal;  { used to store the value in ohms of RREF }
Ohm_Output:Treal;       { To store ohms of ROUTPUT       }
V_REF:Treal;            { To store the meas voltage             }
V_Output:Treal;         { To store the meas voltage      }
I_REF:Treal;            { To store the Current reference  }
I_Out:Treal;            { To Store the Output Current     }
Output_File:Text;       { Link to output text file      }

DutToChan
1:1;
```

```
{*************************************************************}

begin
{!!!!!!!!!!!!!!!!!!!!!!!THE MAIN CODE STARTS HERE!!!!!!!!!!!!!!!!!!!!!!!}

{******************** DC POWER SUPPLY PIN DECLARATION SECTION ***********}
PinTableOpen;
        PinSet(PIN1, 1);
        PinCardSet(PC1, S_PINCARD1);
PinTableClose;
{*********************************************************************}

{*********************** ANALOG CHANNEL DECLARATION SECTION *************}
AnalogChanTableOpen;
{begin the channel declarations}
        AnalogChanSet(POS_35, S_Analog, 35); {initialize ch 35 as POS_35}
        AnalogChanSet(NEG_40, S_Analog, 40); {initialize ch 40 as NEG_40}
        AnalogChanSet(POS_43, S_Analog, 43); {initialize ch 35 as POS_43}
        AnalogChanSet(NEG_48, S_Analog, 48); {initialize ch 40 as NEG_48}
        AnalogChanListSet(Analog_1, POS_35, NEG_40, POS_43, NEG_48);
        {in the upper line we define both pines in Analog_1}
AnalogChanTableClose;
{End the Channel declarations}
{*********************************************************************}

{********************** POWER SUPPLY SETUP SECTION ********************}
{first we setup and connect the power supply with 0 volts}
DCSetupOpen(DCSetup1);
DCSet(0v, 0v, 0v, 0v, 0ma, 0ma, PC1);
DCConnect(PIN1, S_LOW, S_OPEN);
DCSetupClose;

{now we select the power supply}
SetupSelect(DCSetup1);

{now we startup the power supply}
SupplySet(S_DPS1A, 3V, 150ma, S_IGAIN_1X, S_HIGH_CURR_OVERRIDE_BIAS);
wait(50ms); {wait to stabilice}
{*********************************************************************}

{********************** VOLTMETER SETUP SECTION ************************}

{now we connect the pines for measure voltage drop in RREF}
AnalogSetConnect(POS_35, S_Diff_Pos); {connect 35 as Diff_Pos}
```

```
AnalogSetConnect(NEG_40, S_Diff_Neg); {connect 40 as Diff_Neg}
{new we choose the range}
AnalogSet(POS_35, S_Diff, 10V); {10 volts range}
{now we initialize the differential voltage read}
AnalogReadInit(POS_35, S_Diff, false); {boolean is false for DC}
wait(50ms); {wait for reading}
{*************************************************************************}



{******************** ANALOG MEASUREMENT SECTION **********************}
{we execute the reading}
V_REF:= AnalogRead;  { Stores the voltage Drop of RREF }
{now we disconnect the pines for measure voltage drop in RREF}
AnalogSetConnect(POS_35, S_OPEN); {disconnect 35 as Diff_Pos}
AnalogSetConnect(NEG_40, S_OPEN); {disconnect 40 as Diff_Neg}

{now we connect the pines for measure voltage drop in ROUTPUT}
AnalogSetConnect(POS_43, S_Diff_Pos); {connect 43 as Diff_Pos}
AnalogSetConnect(NEG_48, S_Diff_Neg); {connect 48 as Diff_Neg}

{new we choose the range}
pause;
AnalogSet(POS_43, S_Diff, 500mV); {500mV volts range}

{ now we initialize the differential voltage read}
AnalogReadInit(POS_43, S_Diff, false); {boolean is false for DC}

{we execute the reading}
V_Output:= AnalogRead;  { Stores the voltage Drop of ROUTPUT }
{**************************************************************************}



{ **************** DATA USER QUESTION SECTION **********************}
{Now we ask the resistance values for each resistor load }
writeln('I am goint to ask you the value of each resistor');
writeln('After the maginitude, add a "Z", example: 1000Z');
writeln('because "Z" for the VLCT means "resistance in ohms"');
writeln;
Writeln(' Resistance in ohms of RREF ? '); {value for reference R}
Readln(Ohm_Ref);
Writeln(' Resistance in ohms of ROUTPUT ? '); {value for Output R}
Readln(Ohm_Output);
{*********************************************************************}
```

```
{**************** COMPUTING RESULTS SECTION ************************ }

I_REF:= (V_REF/Ohm_Ref); {calculates the current reference}
I_Out:= (V_Output/Ohm_Output); {calculates the Colector #2 current}

{*****************************************************************}


{ *********** SECTION FOR RESULTS DISPLAY IN SCREEN *************** }
writeln('Test Results are...');
writeln;
writeln('Voltage of DPS1A is: ', SupplyRead(S_DPS1A, S_VOLTAGE));
writeln('Output current DPS1A is: ', SupplyRead(S_DPS1A, S_CURRENT));
writeln('The voltage in Rref: ', V_REF);  {display meas in the screen}
writeln('The voltage Routput: ', V_Output); {display v_output}
writeln('The Current Reference: ', I_REF);  {display i_ref}
writeln('The Output Current: ', I_Out);    {display i_output}
Writeln;
Writeln('Program ready to sent data to file "Functional_out_1.dat" ');
pause;
{****************************************************************}


{ *********** OUTPUT FILE SECTION ********************************}
{now we prepare the output file}
Rewrite(Output_File, 'Functional_out_1.dat');
{now we sent the data}
writeln(Output_File, 'FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS');
writeln(Output_File, 'Voltage of DPS1A is: ', SupplyRead(S_DPS1A, S_VOLTAGE));
writeln(Output_File, 'Current from DPS1A is: ', SupplyRead(S_DPS1A, S_CURRENT));
writeln(Output_File, 'The voltage in RREF: ', V_REF);  {save v_ref}
writeln(Output_File, 'The voltage in ROUTPUT: ', V_Output); {save v_output}
writeln(Output_File, 'The Current Reference: ', I_REF); {save i_ref}
writeln(Output_File, 'The Output Current: ', I_Out);
{now we close the output file}
FileClose(Output_File);
{*****************************************************************}


{**************** POWER SUPPLY SHUTDOWN SECTION ********************}
{now the shutdown power supply section}
SupplySet(S_DPS1A, 0v, 1ma, S_IGAIN_1X, S_HIGH_CURR_OVERRIDE_BIAS);
SuppLyDisconnect(S_DPS1A);
wait(50ms);
writeln('DPS1A is off');
```

```
{************************************************************************}



{************* END PROGRAM SECTION ***********************************}
writeln;
writeln('data send to file');
pause;
{ ******************** here program ends ****************************}


end.
```

PROGRAM FUNCTIONAL_1B

```
program Functional_1B;

{************************************************************ }
{ * Functional Bipolar Current-Mirror Test program          * }
{ * Measures voltage in both resistors                      * }
{ * Calculate the current passing through                   * }
{ * Execute the Test and writes result in output file       * }
{ * On this version input file is not used                  * }
{ * This version uses the vlct power supply  for DC SWEEP   * }
{ * Program Created by Sergio Couttolenc Valdes             * }
{ * June 2004                                               * }
{************************************************************ }


{********************** TYPE SECTION ************************}

TYPE
{here we define the list of channels                    * }
{because they are a user defined variable               * }
{POS_35 and NEG_40 are the first Diff-Volt used to measure   * }
{voltage drop in Resistor reference in collector Q1     * }
{POS_43 and NEG_48 are the second Diff-Volt used to measure  * }
{voltage drop in Load Resistor in Collector Q2          * }

AnalogChanList=(POS_35, NEG_40, POS_43, NEG_48, Analog_1);
```

```
{now the variables to setup the power supply}
{we need to use them because we are using the DPS1A}
PinList = (PIN1);  {just pin 1}
PinCardList = (PC1);  {pincard 1}
TestNames = (Test1);
DCSetup = (DCSetup1);  {name of the dps1a function}
{************************************************************}


{*************** VARIABLES DECLARATION SECTION  ***************}

VAR
{This is the variable that stores the measurement}
Ohm_Ref:Treal;  { used to store the value in ohms of RREF }
Ohm_Output:Treal;        { To store ohms of ROUTPUT       }
V_REF:Treal;             { To store the meas voltage            }
V_Output:Treal;          { To store the meas voltage      }
I_REF:Treal;             { To store the Current reference  }
I_Out:Treal;             { To Store the Output Current     }
Output_File:Text;        { Link to output text file      }
Number_units:real;       { number of units to be tested          }
Serial_number:real;      { serial number of the unit      }

{next variables used in conversion of data input}
VOLTAGE_UNIT, CURRENT_UNIT, RES_UNIT :TREAL;
USER_INPUT, final:Treal;

{next variables used in the counter}
init:real;
init2:real;


DutToChan {pin 1 of DUT to Pin 1 from DPS1A}
1:1;

{*************************************************************}


begin
{!!!!!!!!!!!!!!!!!!!!!!!THE MAIN CODE STARTS HERE!!!!!!!!!!!!!!!!!!!!!!}


{**************************************************}
{          REAL-> TREAL CONVERSION TABLE                   }
VOLTAGE_UNIT:=1V;   {Voltage Base Unit-> Voltage}
CURRENT_UNIT:=1A;   {Current Base Unit-> Ampere}
RES_UNIT:=1Z;              {Resistor Base Unit-> Ohm}
{**************************************************}
```

```
{******************* OUTPUT FILE LINK ****************}
{now we prepare the output file}
Rewrite(Output_File, 'Functional_out_1B.dat');
{*************************************************}




{ **************** DATA USER QUESTION SECTION **********************}

writeln('FUNCTIONAL PROGRAM NUMBER 1B');
writeln;
writeln('Functional Bipolar Current-Mirror Test program ');
writeln('Measures voltage in both resistors');
writeln('Calculate the current passing through');
writeln('Execute the Test and writes result in output file');
writeln('This program can test any number of units!!!!');
writeln('On this version input file is not used');
writeln('This version uses the vlct power supply');
writeln('Program Created by Sergio Couttolenc Valdes');
writeln;
pause;
ERASESCREEN;


writeln('please input the number of units to be tested');
readln(Number_units);
WRITELN('Number of units to be tested: ', Number_units);
pause;

writeln('please input the voltage value for DC Power Supply');
readln(USER_INPUT);
final:=USER_INPUT*VOLTAGE_UNIT;
WRITELN('USER_INPUT: ', USER_INPUT, ' FINAL: ', FINAL);

pause;

{Now we ask the resistance values for each resistor load }
writeln('I am goint to ask you the value of each resistor');
writeln('The symbol for Ohm in the VLCT is "Z"');
writeln('is not necesary to type it');
writeln;
Writeln(' Resistance in ohms of RREF ? '); {value for reference R}
Readln(USER_INPUT);
Ohm_Ref:=USER_INPUT*RES_UNIT;
writeln('USER_INPUT: ', USER_INPUT, ' Resistor Ref Value: ', Ohm_Ref);

{now the value for Output Resistor}
Writeln(' Resistance in ohms of ROUTPUT ? '); {value for Output R}
Readln(USER_INPUT);
```

```
Ohm_Output:=USER_INPUT*RES_UNIT;
writeln('USER_INPUT: ', USER_INPUT, ' Output Resistor Value: ', Ohm_Output);

{******************************************************************}




{******************** DC POWER SUPPLY PIN DECLARATION SECTION ***********}
{this section is a DPS1A necessary step}
{for more information consult VLCT programming manual}
{and good luck !!...jajajajaja }
PinTableOpen;
        PinSet(PIN1, 1);
        PinCardSet(PC1, S_PINCARD1);
PinTableClose;
{*********************************************************************}



{************************ ANALOG CHANNEL DECLARATION SECTION *************}
AnalogChanTableOpen;
{begin the channel declarations}
        AnalogChanSet(POS_35, S_Analog, 35); {initialize ch 35 as POS_35}
        AnalogChanSet(NEG_40, S_Analog, 40); {initialize ch 40 as NEG_40}
        AnalogChanSet(POS_43, S_Analog, 43); {initialize ch 35 as POS_43}
        AnalogChanSet(NEG_48, S_Analog, 48); {initialize ch 40 as NEG_48}
        AnalogChanListSet(Analog_1, POS_35, NEG_40, POS_43, NEG_48);
        {in the upper line we define both pines in Analog_1}
AnalogChanTableClose;
{End the Channel declarations}
{*********************************************************************}



{********************** POWER SUPPLY SETUP SECTION ********************}
{first we setup and connect the power supply with 0 volts}
DCSetupOpen(DCSetup1);
DCSet(0v, 0v, 0v, 0v, 0ma, 0ma, PC1); {all parameters are in cero}
DCConnect(PIN1, S_LOW, S_OPEN);
DCSetupClose;

{now we select the power supply}
SetupSelect(DCSetup1);

{*********************************************************************}

init:=0;
init2:=1;
{**************************** ATTENTION CYCLE STARTS HERE !!! ***********}
```

```
While(Number_units > init) DO
begin   {CYCLE STARTS HERE!!!}
        writeln('TEST IN PROGRESS');
        writeln('INSERT THE UNIT!!!');
        pause;
        writeln('please input the serial number of the unit');
        readln(Serial_number);
        WRITELN('The serial number of the unit is: ', Serial_number);
        writeln('');
        pause;

        {next we setup the voltage and current limit for DSP1A};

        SupplySet(S_DPS1A, final, 500ma, S_IGAIN_1X, S_HIGH_CURR_OVERRIDE_BIAS);
        wait(100ms); {delay}
        writeln('Voltage on 1A is ',SupplyRead(S_DPS1A, S_VOLTAGE));
        {**********************************************************************}



{********************** VOLTMETER AND ANALOG MEAS SECTION **************}

{now we connect the pines for measure voltage drop in RREF}
AnalogSetConnect(POS_35, S_Diff_Pos); {connect 35 as Diff_Pos}
AnalogSetConnect(NEG_40, S_Diff_Neg); {connect 40 as Diff_Neg}
{new we choose the range}
AnalogSet(POS_35, S_Diff, 10V); {10 volts range}
{now we initialize the differential voltage read}
AnalogReadInit(POS_35, S_Diff, false); {boolean is false for DC}
wait(50ms); {wait for reading}
{we execute the reading}
V_REF:= AnalogRead;  { Stores the voltage Drop of RREF }
{now we disconnect the pines for measure voltage drop in RREF}
AnalogSetConnect(POS_35, S_OPEN); {disconnect 35 as Diff_Pos}
AnalogSetConnect(NEG_40, S_OPEN); {disconnect 40 as Diff_Neg}



{now we connect the pines for measure voltage drop in ROUTPUT}
AnalogSetConnect(POS_43, S_Diff_Pos); {connect 43 as Diff_Pos}
AnalogSetConnect(NEG_48, S_Diff_Neg); {connect 48 as Diff_Neg}

{new we choose the range}

AnalogSet(POS_43, S_Diff, 500mV); {500mV volts range}
wait(50ms);
{ now we initialize the differential voltage read}
AnalogReadInit(POS_43, S_Diff, false); {boolean is false for DC}

{we execute the reading}
```

```
V_Output:= AnalogRead;  { Stores the voltage Drop of ROUTPUT }
{***********************************************************************}


{**************** COMPUTING RESULTS SECTION ************************ }

I_REF:= (V_REF/Ohm_Ref); {calculates the current reference}
I_Out:= (V_Output/Ohm_Output); {calculates the Colector #2 current}

{*******************************************************************}




{ ************ SECTION FOR RESULTS DISPLAY IN SCREEN **************** }
writeln('Test Results are...');
writeln;
writeln('Voltage of DPS1A is: ', SupplyRead(S_DPS1A, S_VOLTAGE));
writeln('Output current DPS1A is: ', SupplyRead(S_DPS1A, S_CURRENT));
WRITELN('The serial number of the unit is: ', Serial_number);
writeln('The voltage in Rref: ', V_REF);  {display meas in the screen}
writeln('The voltage Routput: ', V_Output); {display v_output}
writeln('The Current Reference: ', I_REF);  {display i_ref}
writeln('The Output Current: ', I_Out);    {display i_output}
Writeln;
Writeln('PROGRAM IS SENDING DATA!!! to file "Functional_1B.dat" ');
wait(100ms);
{******************************************************************}



{ ************ OUTPUT FILE SECTION **********************************}

{now we sent the data}
writeln(Output_File, 'FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS');
writeln(Output_File, 'Serial number of the unit is: ', Serial_number);
writeln(Output_File, 'Voltage of DPS1A is: ', SupplyRead(S_DPS1A, S_VOLTAGE));
writeln(Output_File, 'Current from DPS1A is: ', SupplyRead(S_DPS1A, S_CURRENT));
writeln(Output_File, 'The voltage in RREF: ', V_REF);  {save v_ref}
writeln(Output_File, 'The voltage in ROUTPUT: ', V_Output); {save v_output}
writeln(Output_File, 'The Current Reference: ', I_REF); {save i_ref}
writeln(Output_File, 'The Output Current: ', I_Out); {save_i_out}

{*******************************************************************}


{**************** POWER SUPPLY SHUTDOWN SECTION ********************}
{now the shutdown power supply section}
```

```
SupplySet(S_DPS1A, 0v, 1ma, S_IGAIN_1X, S_HIGH_CURR_OVERRIDE_BIAS);
SuppLyDisconnect(S_DPS1A);
wait(700ms);
writeln('DPS1A is off');
{*********************************************************************}


{next we disconnect the channels}
AnalogSetConnect(POS_35, S_OPEN); {disconnect 35 as Diff_Pos}
AnalogSetConnect(NEG_40, S_OPEN); {disconnect 40 as Diff_Neg}

{here we do the increment of the counter};
init:=init+init2;

end;
{************* ATTENTION CYCLE ENDS HERE !!!! ************************}



{************* END PROGRAM SECTION ***********************************}
writeln;
FileClose(Output_File);
writeln('Test Ended...');
pause;
{ ******************** here program ends ***************************}


end.
```

PROGRAM FUNCTIONAL_TEST_2

```
program Functional_Test_2;

{************************************************************ }
{ * Functional Bipolar Current-Mirror Test program          * }
{ * Measures voltage in both resistors                      * }
{ * Calculate the current passing through                  * }
{ * Execute the Test and writes result in output file       * }
{ * On this version input file is not used                 * }
{ * This version uses the vlct power supply  for DC SWEEP    * }
{ * Program Created by Sergio Couttolenc Valdes             * }
{ * June 2004                                              * }
{************************************************************ }


{********************** TYPE SECTION ************************}

TYPE
{here we define the list of channels                    * }
{because they are a user defined variable               * }
{POS_35 and NEG_40 are the first Diff-Volt used to measure   * }
{voltage drop in Resistor reference in collector Q1         * }
{POS_43 and NEG_48 are the second Diff-Volt used to measure  * }
{voltage drop in Load Resistor in Collector Q2              * }

AnalogChanList=(POS_35, NEG_40, POS_43, NEG_48, Analog_1);

{now the variables to setup the power supply}
{we need to use them because we are using the DPS1A}
PinList = (PIN1);  {just pin 1}
PinCardList = (PC1);  {pincard 1}
TestNames = (Test1);
DCSetup = (DCSetup1);  {name of the dps1a function}
{************************************************************}



{*************** VARIABLES DECLARATION SECTION  ***************}

VAR
{This is the variable that stores the measurement}
Ohm_Ref:Treal;  { used to store the value in ohms of RREF }
Ohm_Output:Treal;         { To store ohms of ROUTPUT       }
V_REF:Treal;              { To store the meas voltage              }
V_Output:Treal;           { To store the meas voltage      }
I_REF:Treal;              { To store the Current reference  }
I_Out:Treal;              { To Store the Output Current     }
Output_File:Text;         { Link to output text file      }
```

```
{variable definition for the DC ramp, using TREAL types !!}

 step, init, final, timestep, V:treal;

{timestep is the variable to be used for time between steps}
{step, is the variable for the space between voltage changes}
{init, is the variable for initial voltage}
{final, is the variable for final voltage}
{timestep, is the variable for time between voltage changes}
{V, is the variable that goes into the DC PS}
{RES_UNIT is the unit variable for resistance}

TIME_UNIT, VOLTAGE_UNIT, CURRENT_UNIT, RES_UNIT :TREAL;
USER_INPUT:real;


DutToChan {pin 1 of DUT to Pin 1 from DPS1A}
1:1;

{************************************************************}


begin
{!!!!!!!!!!!!!!!!!!!!!!!THE MAIN CODE STARTS HERE!!!!!!!!!!!!!!!!!!!!!!}


{**************************************************}
{          REAL-> TREAL CONVERSION TABLE                  }
TIME_UNIT:=1S;                       {Time Base Unit-> Seconds}
VOLTAGE_UNIT:=1V;   {Voltage Base Unit-> Voltage}
CURRENT_UNIT:=1A;   {Current Base Unit-> Ampere}
RES_UNIT:=1Z;            {Resistor Base Unit-> Ohm}
{*************************************************}



{****************** OUTPUT FILE LINK ***************}
{now we prepare the output file}
Rewrite(Output_File, 'Functional_out_2.dat');
{***************************************************}




{ **************** DATA USER QUESTION SECTION *********************}

writeln('FUNCTIONAL PROGRAM NUMBER 2');
writeln;
writeln('Functional Bipolar Current-Mirror Test program ');
writeln('Measures voltage in both resistors');
writeln('Calculate the current passing through');
```

```
writeln('Execute the Test and writes result in output file');
writeln('On this version input file is not used');
writeln('This version uses the vlct power supply');
writeln('Program Created by Sergio Couttolenc Valdes');
writeln;
pause;
ERASESCREEN;

writeln('please input the initial voltage value for DC ramp');
readln(USER_INPUT);
init:= USER_INPUT*VOLTAGE_UNIT; {here is where we convert treal to unit}
WRITELN('USER_INPUT: ', USER_INPUT, ' INIT: ', init);

writeln('please input the final voltage value for DC ramp');
readln(USER_INPUT);
final:=USER_INPUT*VOLTAGE_UNIT;
WRITELN('USER_INPUT: ', USER_INPUT, ' FINAL: ', FINAL);

writeln('please input the step voltage value for DC ramp');
readln(USER_INPUT);
step:=USER_INPUT*VOLTAGE_UNIT;
WRITELN('USER_INPUT: ', USER_INPUT, ' STEP: ', STEP);

writeln('please input the time between steps in Seconds');
readln(USER_INPUT);
timestep:=USER_INPUT*TIME_UNIT;
WRITELN('USER_INPUT: ', USER_INPUT, ' TIMESTEP: ', TIMESTEP);

pause;

{Now we ask the resistance values for each resistor load }
writeln('I am goint to ask you the value of each resistor');
writeln('The symbol for Ohm in the VLCT is "Z"');
writeln('is not necesary to type it');
writeln;
Writeln(' Resistance in ohms of RREF ? '); {value for reference R}
Readln(USER_INPUT);
Ohm_Ref:=USER_INPUT*RES_UNIT;
writeln('USER_INPUT: ', USER_INPUT, ' Resistor Ref Value: ', Ohm_Ref);

{now the value for Output Resistor}
Writeln(' Resistance in ohms of ROUTPUT ? '); {value for Output R}
Readln(USER_INPUT);
Ohm_Output:=USER_INPUT*RES_UNIT;
writeln('USER_INPUT: ', USER_INPUT, ' Output Resistor Value: ', Ohm_Output);

{*****************************************************************}



{******************** DC POWER SUPPLY PIN DECLARATION SECTION ***********}
```

```
{this section is a DPS1A necessary step}
{for more information consult VLCT programming manual}
{and good luck !!...jajajajaja }
PinTableOpen;
        PinSet(PIN1, 1);
        PinCardSet(PC1, S_PINCARD1);
PinTableClose;
{**********************************************************************}


{*********************** ANALOG CHANNEL DECLARATION SECTION *************}
AnalogChanTableOpen;
{begin the channel declarations}
        AnalogChanSet(POS_35, S_Analog, 35); {initialize ch 35 as POS_35}
        AnalogChanSet(NEG_40, S_Analog, 40); {initialize ch 40 as NEG_40}
        AnalogChanSet(POS_43, S_Analog, 43); {initialize ch 35 as POS_43}
        AnalogChanSet(NEG_48, S_Analog, 48); {initialize ch 40 as NEG_48}
        AnalogChanListSet(Analog_1, POS_35, NEG_40, POS_43, NEG_48);
        {in the upper line we define both pines in Analog_1}
AnalogChanTableClose;
{End the Channel declarations}
{**********************************************************************}


{********************** POWER SUPPLY SETUP SECTION ********************}
{first we setup and connect the power supply with 0 volts}
DCSetupOpen(DCSetup1);
DCSet(0v, 0v, 0v, 0v, 0ma, 0ma, PC1); {all parameters are in cero}
DCConnect(PIN1, S_LOW, S_OPEN);
DCSetupClose;

{now we select the power supply}
SetupSelect(DCSetup1);

{**********************************************************************}


{**************************** ATTENTION CYCLE STARTS HERE !!! ***********}
{next, the cycle that performs a DC Sweep}
V:=init; {variable that is used like incremental index for the cycle}

While(final > init) DO
begin   {CYCLE STARTS HERE!!!}
        writeln(V);
        writeln('DC RAMP IN PROGRESS!!!');
        writeln('');

        {next we setup the voltage and current limit for DSP1A};

        SupplySet(S_DPS1A, V, 500ma, S_IGAIN_1X, S_HIGH_CURR_OVERRIDE_BIAS);
```

```
        wait(timestep); {time to change to a new step}
        writeln('Voltage on 1A is ',SupplyRead(S_DPS1A, S_VOLTAGE));
        V:= V + step; {here we do the increment of voltage for dc ramp};
        init:=V;
{*************************************************************************}


{********************** VOLTMETER AND ANALOG MEAS SECTION **************}

{now we connect the pines for measure voltage drop in RREF}
AnalogSetConnect(POS_35, S_Diff_Pos); {connect 35 as Diff_Pos}
AnalogSetConnect(NEG_40, S_Diff_Neg); {connect 40 as Diff_Neg}
{new we choose the range}
AnalogSet(POS_35, S_Diff, 10V); {10 volts range}
{now we initialize the differential voltage read}
AnalogReadInit(POS_35, S_Diff, false); {boolean is false for DC}
wait(50ms); {wait for reading}
{we execute the reading}
V_REF:= AnalogRead;  { Stores the voltage Drop of RREF }
{now we disconnect the pines for measure voltage drop in RREF}
AnalogSetConnect(POS_35, S_OPEN); {disconnect 35 as Diff_Pos}
AnalogSetConnect(NEG_40, S_OPEN); {disconnect 40 as Diff_Neg}



{now we connect the pines for measure voltage drop in ROUTPUT}
AnalogSetConnect(POS_43, S_Diff_Pos); {connect 43 as Diff_Pos}
AnalogSetConnect(NEG_48, S_Diff_Neg); {connect 48 as Diff_Neg}

{new we choose the range}

AnalogSet(POS_43, S_Diff, 500mV); {500mV volts range}

{ now we initialize the differential voltage read}
AnalogReadInit(POS_43, S_Diff, false); {boolean is false for DC}

{we execute the reading}
V_Output:= AnalogRead;  { Stores the voltage Drop of ROUTPUT }
{**************************************************************************}


{**************** COMPUTING RESULTS SECTION ************************ }

I_REF:= (V_REF/Ohm_Ref); {calculates the current reference}
I_Out:= (V_Output/Ohm_Output); {calculates the Colector #2 current}

{**********************************************************************}
```

```
{ ************ SECTION FOR RESULTS DISPLAY IN SCREEN **************** }
writeln('Test Results are...');
writeln;
writeln('Voltage of DPS1A is: ', SupplyRead(S_DPS1A, S_VOLTAGE));
writeln('Output current DPS1A is: ', SupplyRead(S_DPS1A, S_CURRENT));
writeln('The voltage in Rref: ', V_REF);  {display meas in the screen}
writeln('The voltage Routput: ', V_Output); {display v_output}
writeln('The Current Reference: ', I_REF);  {display i_ref}
writeln('The Output Current: ', I_Out);    {display i_output}
Writeln;
Writeln('PROGRAM IS SENDING DATA!!! to file "Functional_out_2.dat" ');
wait(500ms);
{******************************************************************}



{ ************ OUTPUT FILE SECTION ***********************************}

{now we sent the data}
writeln(Output_File, 'FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS');
writeln(Output_File, 'Voltage of DPS1A is: ', SupplyRead(S_DPS1A, S_VOLTAGE));
writeln(Output_File, 'Current from DPS1A is: ', SupplyRead(S_DPS1A, S_CURRENT));
writeln(Output_File, 'The voltage in RREF: ', V_REF);  {save v_ref}
writeln(Output_File, 'The voltage in ROUTPUT: ', V_Output); {save v_output}
writeln(Output_File, 'The Current Reference: ', I_REF); {save i_ref}
writeln(Output_File, 'The Output Current: ', I_Out);


end;
{************* ATTENTION CYCLE ENDS HERE !!!! ************************}

{*******************************************************************}



{***************** POWER SUPPLY SHUTDOWN SECTION *********************}
{now the shutdown power supply section}
SupplySet(S_DPS1A, 0v, 1ma, S_IGAIN_1X, S_HIGH_CURR_OVERRIDE_BIAS);
SuppLyDisconnect(S_DPS1A);
wait(50ms);
writeln('DPS1A is off');
{*********************************************************************}



{************* END PROGRAM SECTION **********************************}
```

```
writeln;
FileClose(Output_File);
writeln('data send to file');
pause;
{ ******************** here program ends ****************************}

end.
```

PROGRAM FUNCTIONAL_TEST_3

```
program Functional_Test_3;

{*********************************************************** }
{ * Functional Bipolar Current-Mirror Test program          * }
{ * Measures voltage in both resistors                      * }
{ * Calculate the current passing through                   * }
{ * Execute the Test and writes result in output file       * }
{ * On this version input file is used                      * }
{ * This version uses the vlct power supply for DC SWEEP    * }
{ * Program Created by Sergio Couttolenc Valdes             * }
{ * June 2004                                               * }
{*********************************************************** }


{********************** TYPE SECTION ************************}

TYPE
{here we define the list of channels                        * }
{because they are a user defined variable                   * }
{POS_35 and NEG_40 are the first Diff-Volt used to measure  * }
{voltage drop in Resistor reference in collector Q1         * }
{POS_43 and NEG_48 are the second Diff-Volt used to measure  * }
{voltage drop in Load Resistor in Collector Q2              * }

AnalogChanList=(POS_35, NEG_40, POS_43, NEG_48, Analog_1);

{now the variables to setup the power supply}
{we need to use them because we are using the DPS1A}
PinList = (PIN1);  {just pin 1}
PinCardList = (PC1);  {pincard 1}
```

```
TestNames = (Test1);
DCSetup = (DCSetup1);  {name of the dps1a function}
{**************************************************************}



{*************** VARIABLES DECLARATION SECTION  ***************}

VAR
{This is the variable that stores the measurement}
Ohm_Ref:Treal;  { used to store the value in ohms of RREF }
Ohm_Output:Treal;         { To store ohms of ROUTPUT       }
V_REF:Treal;              { To store the meas voltage            }
V_Output:Treal;           { To store the meas voltage     }
I_REF:Treal;              { To store the Current reference  }
I_Out:Treal;              { To Store the Output Current     }
Output_File:Text;         { Link to output text file      }
Input_File:Text;  { Link to input text file       }


{variable definition for the DC ramp, using TREAL types !!}

 step, init, final, timestep, V:treal;

{timestep is the variable to be used for time between steps}
{step, is the variable for the space between voltage changes}
{init, is the variable for initial voltage}
{final, is the variable for final voltage}
{timestep, is the variable for time between voltage changes}
{V, is the variable that goes into the DC PS}
{RES_UNIT is the unit variable for resistance}

TIME_UNIT, VOLTAGE_UNIT, CURRENT_UNIT, RES_UNIT :TREAL;
USER_INPUT:real;


DutToChan {pin 1 of DUT to Pin 1 from DPS1A}
1:1;

{***************************************************************}


begin
{!!!!!!!!!!!!!!!!!!!!!!!THE MAIN CODE STARTS HERE!!!!!!!!!!!!!!!!!!!!!!}


{***************************************************}
{         REAL-> TREAL CONVERSION TABLE                    }
TIME_UNIT:=1S;                {Time Base Unit-> Seconds}
VOLTAGE_UNIT:=1V;   {Voltage Base Unit-> Voltage}
CURRENT_UNIT:=1A;   {Current Base Unit-> Ampere}
RES_UNIT:=1Z;             {Resistor Base Unit-> Ohm}
```

```
{****************************************************}


{****************** OUTPUT FILE LINK ****************}
{now we prepare the input and output files}
Reset(Input_File, 'Functional_in_3.dat');
Rewrite(Output_File, 'Functional_out_3.dat');
{****************************************************}



{ **************** MESSAGE SECTION ***********************}

writeln('FUNCTIONAL PROGRAM NUMBER 2');
writeln;
writeln('Functional Bipolar Current-Mirror Test program ');
writeln('Measures voltage in both resistors');
writeln('Calculate the current passing through');
writeln('Execute the Test and writes result in output file');
writeln('On this version input file is not used');
writeln('This version uses the vlct power supply');
writeln('Program Created by Sergio Couttolenc Valdes');
writeln;
pause;
ERASESCREEN;
{**********************************************************}



{******************** INPUT DATA READ SECTION *******************}
{HERE WE READ DATA FROM INPUT FILE!}

Readln(Input_File, init);
writeln('Initial voltage: ', init);
Readln(Input_File, final);
writeln('Final voltage:', final);
Readln(Input_File, step);
writeln('Step voltage value:', step);
Readln(Input_File, timestep);
writeln('Time between steps:', timestep);
Readln(Input_File, Ohm_Ref);
writeln('Resistor in Q1:', Ohm_Ref);
Readln(Input_File, Ohm_Output);
writeln('Resistor in Q2:', Ohm_Output);

pause;
{*******************************************************************}
```

```
{******************** DC POWER SUPPLY PIN DECLARATION SECTION ***********}
{this section is a DPS1A necessary step}
{for more information consult VLCT programming manual}
{and good luck !!...jajajajaja }
PinTableOpen;
        PinSet(PIN1, 1);
        PinCardSet(PC1, S_PINCARD1);
PinTableClose;
{*********************************************************************}


{************************ ANALOG CHANNEL DECLARATION SECTION *************}
AnalogChanTableOpen;
{begin the channel declarations}
        AnalogChanSet(POS_35, S_Analog, 35); {initialize ch 35 as POS_35}
        AnalogChanSet(NEG_40, S_Analog, 40); {initialize ch 40 as NEG_40}
        AnalogChanSet(POS_43, S_Analog, 43); {initialize ch 35 as POS_43}
        AnalogChanSet(NEG_48, S_Analog, 48); {initialize ch 40 as NEG_48}
        AnalogChanListSet(Analog_1, POS_35, NEG_40, POS_43, NEG_48);
        {in the upper line we define both pines in Analog_1}
AnalogChanTableClose;
{End the Channel declarations}
{*********************************************************************}


{********************** POWER SUPPLY SETUP SECTION ********************}
{first we setup and connect the power supply with 0 volts}
DCSetupOpen(DCSetup1);
DCSet(0v, 0v, 0v, 0v, 0ma, 0ma, PC1); {all parameters are in cero}
DCConnect(PIN1, S_LOW, S_OPEN);
DCSetupClose;

{now we select the power supply}
SetupSelect(DCSetup1);

{*********************************************************************}


{**************************** ATTENTION CYCLE STARTS HERE !!! ***********}
{next, the cycle that performs a DC Sweep}
V:=init; {variable that is used like incremental index for the cycle}

While(final > init) DO
begin   {CYCLE STARTS HERE!!!}
        writeln(V);
```

```
        writeln('DC RAMP IN PROGRESS!!!');
        writeln('');

        {next we setup the voltage and current limit for DSP1A};

        SupplySet(S_DPS1A, V, 500ma, S_IGAIN_1X, S_HIGH_CURR_OVERRIDE_BIAS);
        wait(timestep); {time to change to a new step}
        writeln('Voltage on 1A is ',SupplyRead(S_DPS1A, S_VOLTAGE));
        V:= V + step; {here we do the increment of voltage for dc ramp};
        init:=V;
{**********************************************************************}



{********************** VOLTMETER AND ANALOG MEAS SECTION **************}

{now we connect the pines for measure voltage drop in RREF}
AnalogSetConnect(POS_35, S_Diff_Pos); {connect 35 as Diff_Pos}
AnalogSetConnect(NEG_40, S_Diff_Neg); {connect 40 as Diff_Neg}
{new we choose the range}
AnalogSet(POS_35, S_Diff, 10V); {10 volts range}
{now we initialize the differential voltage read}
AnalogReadInit(POS_35, S_Diff, false); {boolean is false for DC}
wait(50ms); {wait for reading}
{we execute the reading}
V_REF:= AnalogRead;  { Stores the voltage Drop of RREF }
{now we disconnect the pines for measure voltage drop in RREF}
AnalogSetConnect(POS_35, S_OPEN); {disconnect 35 as Diff_Pos}
AnalogSetConnect(NEG_40, S_OPEN); {disconnect 40 as Diff_Neg}


{now we connect the pines for measure voltage drop in ROUTPUT}
AnalogSetConnect(POS_43, S_Diff_Pos); {connect 43 as Diff_Pos}
AnalogSetConnect(NEG_48, S_Diff_Neg); {connect 48 as Diff_Neg}
{new we choose the range}
AnalogSet(POS_43, S_Diff, 500mV); {500mV volts range}
{ now we initialize the differential voltage read}
AnalogReadInit(POS_43, S_Diff, false); {boolean is false for DC}
{we execute the reading}
V_Output:= AnalogRead;  { Stores the voltage Drop of ROUTPUT }
{**********************************************************************}



{**************** COMPUTING RESULTS SECTION ************************ }

I_REF:= (V_REF/Ohm_Ref); {calculates the current reference}
I_Out:= (V_Output/Ohm_Output); {calculates the Colector #2 current}

{********************************************************************}
```

```
{ ************ SECTION FOR RESULTS DISPLAY IN SCREEN **************** }
writeln('Test Results are...');
writeln;
writeln('Voltage of DPS1A is: ', SupplyRead(S_DPS1A, S_VOLTAGE));
writeln('Output current DPS1A is: ', SupplyRead(S_DPS1A, S_CURRENT));
writeln('The voltage in Rref: ', V_REF);  {display meas in the screen}
writeln('The voltage Routput: ', V_Output); {display v_output}
writeln('The Current Reference: ', I_REF);  {display i_ref}
writeln('The Output Current: ', I_Out);    {display i_output}
Writeln;
Writeln('PROGRAM IS SENDING DATA!!! to file "Functional_out_2.dat" ');
wait(500ms);
{*******************************************************************}



{ ************ OUTPUT FILE SECTION ***********************************}

{now we sent the data}
writeln(Output_File, 'FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS');
writeln(Output_File, 'Voltage of DPS1A is: ', SupplyRead(S_DPS1A, S_VOLTAGE));
writeln(Output_File, 'Current from DPS1A is: ', SupplyRead(S_DPS1A, S_CURRENT));
writeln(Output_File, 'The voltage in RREF: ', V_REF);  {save v_ref}
writeln(Output_File, 'The voltage in ROUTPUT: ', V_Output); {save v_output}
writeln(Output_File, 'The Current Reference: ', I_REF); {save i_ref}
writeln(Output_File, 'The Output Current: ', I_Out);


end;
{************* ATTENTION CYCLE ENDS HERE !!!! ***********************}

{*******************************************************************}



{**************** POWER SUPPLY SHUTDOWN SECTION ********************}
{now the shutdown power supply section}
SupplySet(S_DPS1A, 0v, 1ma, S_IGAIN_1X, S_HIGH_CURR_OVERRIDE_BIAS);
SuppLyDisconnect(S_DPS1A);
wait(50ms);
writeln('DPS1A is off');
{*******************************************************************}



{************* END PROGRAM SECTION **********************************}
```

```
writeln;
FileClose(Input_File);
FileClose(Output_File);
writeln('data send to file');
pause;
{ ********************* here program ends ****************************}


end.
```

Appendix "B"

RAW DATA FROM OUTPUT DATA FILES

FUNCTIONAL_1 OUTPUT DATA FILE

FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.00606V
Current from DPS1A is: 21.98828MA
The voltage in RREF: 2.33857V
The voltage in ROUTPUT: 308.31331MV
The Current Reference: 2.33857MA
The Output Current: 2.90862MA

FUNCTIONAL_TEST_2 OUTPUT DATA FILE FOR DC SWEEP BIPOLAR BASIC CURRENT-MIRROR

FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 46.34513MV
Current from DPS1A is: 12.78388MA
The voltage in RREF: 312.80995UV
The voltage in ROUTPUT: -39.10124UV
The Current Reference: 312.80995NA
The Output Current: -368.87966NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 104.25262MV
Current from DPS1A is: 13.80659MA
The voltage in RREF: -312.80995UV
The voltage in ROUTPUT: -23.46075UV
The Current Reference: -312.80995NA
The Output Current: -221.32779NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 155.72595MV
Current from DPS1A is: 13.80659MA
The voltage in RREF: -312.80995UV
The voltage in ROUTPUT: -23.46075UV
The Current Reference: -312.80995NA
The Output Current: -221.32779NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 207.19928MV

Current from DPS1A is: 13.80659MA
The voltage in RREF: -312.80995UV
The voltage in ROUTPUT: -23.46075UV
The Current Reference: -312.80995NA
The Output Current: -221.32779NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 252.23844MV
Current from DPS1A is: 13.80659MA
The voltage in RREF: 0V
The voltage in ROUTPUT: -39.10124UV
The Current Reference: 0A
The Output Current: -368.87966NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 303.71177MV
Current from DPS1A is: 13.80659MA
The voltage in RREF: 0V
The voltage in ROUTPUT: -23.46075UV
The Current Reference: 0A
The Output Current: -221.32779NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 361.61927MV
Current from DPS1A is: 13.80659MA
The voltage in RREF: 312.80995UV
The voltage in ROUTPUT: -23.46075UV
The Current Reference: 312.80995NA
The Output Current: -221.32779NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 406.65843MV
Current from DPS1A is: 13.80659MA
The voltage in RREF: 0V
The voltage in ROUTPUT: -7.82025UV
The Current Reference: 0A
The Output Current: -73.77593NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 451.69759MV
Current from DPS1A is: 13.80659MA
The voltage in RREF: 938.42985UV
The voltage in ROUTPUT: 101.66323UV
The Current Reference: 938.42985NA
The Output Current: 959.08711NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 503.17092MV
Current from DPS1A is: 13.80659MA
The voltage in RREF: 938.42985UV
The voltage in ROUTPUT: 523.95666UV
The Current Reference: 938.42985NA
The Output Current: 4.94299UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 554.64425MV
Current from DPS1A is: 13.80659MA
The voltage in RREF: 7.19463MV
The voltage in ROUTPUT: 2.21313MV

The Current Reference: 7.19463UA
The Output Current: 20.87859UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 606.11757MV
Current from DPS1A is: 13.80659MA
The voltage in RREF: 25.0248MV
The voltage in ROUTPUT: 4.95022MV
The Current Reference: 25.0248UA
The Output Current: 46.70016UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 657.5909MV
Current from DPS1A is: 13.80659MA
The voltage in RREF: 54.11612MV
The voltage in ROUTPUT: 8.40677MV
The Current Reference: 54.11612UA
The Output Current: 79.30913UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 702.63007MV
Current from DPS1A is: 13.80659MA
The voltage in RREF: 89.46365MV
The voltage in ROUTPUT: 12.26997MV
The Current Reference: 89.46365UA
The Output Current: 115.75444UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 754.10339MV
Current from DPS1A is: 13.80659MA
The voltage in RREF: 129.81613MV
The voltage in ROUTPUT: 16.30522MV
The Current Reference: 129.81613UA
The Output Current: 153.82282UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 805.57672MV
Current from DPS1A is: 13.80659MA
The voltage in RREF: 175.48638MV
The voltage in ROUTPUT: 21.04429MV
The Current Reference: 175.48638UA
The Output Current: 198.53103UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 857.05005MV
Current from DPS1A is: 14.8293MA
The voltage in RREF: 219.27977MV
The voltage in ROUTPUT: 25.43927MV
The Current Reference: 219.27977UA
The Output Current: 239.9931UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 902.08921MV
Current from DPS1A is: 14.8293MA
The voltage in RREF: 263.69879MV
The voltage in ROUTPUT: 29.92809MV
The Current Reference: 263.69879UA
The Output Current: 282.34049UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS

Voltage of DPS1A is: 959.99671MV
Current from DPS1A is: 14.8293MA
The voltage in RREF: 309.05623MV
The voltage in ROUTPUT: 34.4482MV
The Current Reference: 309.05623UA
The Output Current: 324.98298UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.00504V
Current from DPS1A is: 14.8293MA
The voltage in RREF: 354.10086MV
The voltage in ROUTPUT: 38.93702MV
The Current Reference: 354.10086UA
The Output Current: 367.33036UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.05651V
Current from DPS1A is: 14.8293MA
The voltage in RREF: 405.7145MV
The voltage in ROUTPUT: 44.08274MV
The Current Reference: 405.7145UA
The Output Current: 415.87492UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.10798V
Current from DPS1A is: 14.8293MA
The voltage in RREF: 451.69757MV
The voltage in ROUTPUT: 48.60285MV
The Current Reference: 451.69757UA
The Output Current: 458.51741UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.15302V
Current from DPS1A is: 14.8293MA
The voltage in RREF: 497.36782MV
The voltage in ROUTPUT: 53.26371MV
The Current Reference: 497.36782UA
The Output Current: 502.48787UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.2045V
Current from DPS1A is: 15.85201MA
The voltage in RREF: 542.72526MV
The voltage in ROUTPUT: 59.11326MV
The Current Reference: 542.72526UA
The Output Current: 557.67227UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.24953V
Current from DPS1A is: 15.85201MA
The voltage in RREF: 589.95956MV
The voltage in ROUTPUT: 67.59041MV
The Current Reference: 589.95956UA
The Output Current: 637.64537UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.30744V
Current from DPS1A is: 15.85201MA
The voltage in RREF: 643.13725MV

The voltage in ROUTPUT: 76.83394MV
The Current Reference: 643.13725UA
The Output Current: 724.84853UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.35248V
Current from DPS1A is: 15.85201MA
The voltage in RREF: 690.68437MV
The voltage in ROUTPUT: 85.29545MV
The Current Reference: 690.68437UA
The Output Current: 804.67408UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.40395V
Current from DPS1A is: 16.87473MA
The voltage in RREF: 736.66743MV
The voltage in ROUTPUT: 93.38159MV
The Current Reference: 736.66743UA
The Output Current: 880.9584UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.45543V
Current from DPS1A is: 16.87473MA
The voltage in RREF: 784.21454MV
The voltage in ROUTPUT: 101.60849MV
The Current Reference: 784.21454UA
The Output Current: 958.57068UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.50047V
Current from DPS1A is: 16.87473MA
The voltage in RREF: 831.13603MV
The voltage in ROUTPUT: 109.42874MV
The Current Reference: 831.13603UA
The Output Current: 1.03235MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.55837V
Current from DPS1A is: 16.87473MA
The voltage in RREF: 883.06249MV
The voltage in ROUTPUT: 117.76513MV
The Current Reference: 883.06249UA
The Output Current: 1.11099MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.59698V
Current from DPS1A is: 17.89744MA
The voltage in RREF: 927.16869MV
The voltage in ROUTPUT: 124.66258MV
The Current Reference: 927.16869UA
The Output Current: 1.17606MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.64845V
Current from DPS1A is: 17.89744MA
The voltage in RREF: 977.21828MV
The voltage in ROUTPUT: 132.34207MV
The Current Reference: 977.21828UA
The Output Current: 1.24851MA

FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.69993V
Current from DPS1A is: 17.89744MA
The voltage in RREF: 1.02445V
The voltage in ROUTPUT: 139.31773MV
The Current Reference: 1.02445MA
The Output Current: 1.31432MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.7514V
Current from DPS1A is: 16.87473MA
The voltage in RREF: 1.07356V
The voltage in ROUTPUT: 146.32467MV
The Current Reference: 1.07356MA
The Output Current: 1.38042MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.80287V
Current from DPS1A is: 17.89744MA
The voltage in RREF: 1.12705V
The voltage in ROUTPUT: 153.86339MV
The Current Reference: 1.12705MA
The Output Current: 1.45154MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.85435V
Current from DPS1A is: 18.92015MA
The voltage in RREF: 1.17648V
The voltage in ROUTPUT: 160.62009MV
The Current Reference: 1.17648MA
The Output Current: 1.51528MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.90582V
Current from DPS1A is: 17.89744MA
The voltage in RREF: 1.22371V
The voltage in ROUTPUT: 166.98577MV
The Current Reference: 1.22371MA
The Output Current: 1.57534MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.95086V
Current from DPS1A is: 18.92015MA
The voltage in RREF: 1.27157V
The voltage in ROUTPUT: 173.30453MV
The Current Reference: 1.27157MA
The Output Current: 1.63495MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.00877V
Current from DPS1A is: 18.92015MA
The voltage in RREF: 1.32412V
The voltage in ROUTPUT: 180.18635MV
The Current Reference: 1.32412MA
The Output Current: 1.69987MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.06024V
Current from DPS1A is: 18.92015MA

The voltage in RREF: 1.37324V
The voltage in ROUTPUT: 186.45819MV
The Current Reference: 1.37324MA
The Output Current: 1.75904MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.10528V
Current from DPS1A is: 18.92015MA
The voltage in RREF: 1.42141V
The voltage in ROUTPUT: 192.46414MV
The Current Reference: 1.42141MA
The Output Current: 1.8157MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.15032V
Current from DPS1A is: 18.92015MA
The voltage in RREF: 1.46958V
The voltage in ROUTPUT: 198.53265MV
The Current Reference: 1.46958MA
The Output Current: 1.87295MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.20179V
Current from DPS1A is: 18.92015MA
The voltage in RREF: 1.51619V
The voltage in ROUTPUT: 204.31964MV
The Current Reference: 1.51619MA
The Output Current: 1.92754MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.25326V
Current from DPS1A is: 18.92015MA
The voltage in RREF: 1.57187V
The voltage in ROUTPUT: 211.02941MV
The Current Reference: 1.57187MA
The Output Current: 1.99084MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.30474V
Current from DPS1A is: 18.92015MA
The voltage in RREF: 1.61879V
The voltage in ROUTPUT: 216.73819MV
The Current Reference: 1.61879MA
The Output Current: 2.0447MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.35621V
Current from DPS1A is: 18.92015MA
The voltage in RREF: 1.66603V
The voltage in ROUTPUT: 222.4939MV
The Current Reference: 1.66603MA
The Output Current: 2.099MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.40125V
Current from DPS1A is: 19.94286MA
The voltage in RREF: 1.71357V
The voltage in ROUTPUT: 228.09319MV
The Current Reference: 1.71357MA

The Output Current: 2.15182MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.45272V
Current from DPS1A is: 19.94286MA
The voltage in RREF: 1.76331V
The voltage in ROUTPUT: 233.95838MV
The Current Reference: 1.76331MA
The Output Current: 2.20715MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.5042V
Current from DPS1A is: 19.94286MA
The voltage in RREF: 1.81711V
The voltage in ROUTPUT: 240.21458MV
The Current Reference: 1.81711MA
The Output Current: 2.26618MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.55567V
Current from DPS1A is: 20.96557MA
The voltage in RREF: 1.86466V
The voltage in ROUTPUT: 245.79824MV
The Current Reference: 1.86466MA
The Output Current: 2.31885MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.60071V
Current from DPS1A is: 20.96557MA
The voltage in RREF: 1.91346V
The voltage in ROUTPUT: 251.35061MV
The Current Reference: 1.91346MA
The Output Current: 2.37123MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.65218V
Current from DPS1A is: 20.96557MA
The voltage in RREF: 1.96194V
The voltage in ROUTPUT: 256.93427MV
The Current Reference: 1.96194MA
The Output Current: 2.42391MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.70366V
Current from DPS1A is: 20.96557MA
The voltage in RREF: 2.00949V
The voltage in ROUTPUT: 262.47101MV
The Current Reference: 2.00949MA
The Output Current: 2.47614MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.75513V
Current from DPS1A is: 20.96557MA
The voltage in RREF: 2.06329V
The voltage in ROUTPUT: 268.53952MV
The Current Reference: 2.06329MA
The Output Current: 2.53339MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.8066V

Current from DPS1A is: 20.96557MA
The voltage in RREF: 2.11178V
The voltage in ROUTPUT: 274.01369MV
The Current Reference: 2.11178MA
The Output Current: 2.58503MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.85164V
Current from DPS1A is: 21.98828MA
The voltage in RREF: 2.16058V
The voltage in ROUTPUT: 279.55043MV
The Current Reference: 2.16058MA
The Output Current: 2.63727MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.90311V
Current from DPS1A is: 21.98828MA
The voltage in RREF: 2.20875V
The voltage in ROUTPUT: 284.91512MV
The Current Reference: 2.20875MA
The Output Current: 2.68788MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.94815V
Current from DPS1A is: 21.98828MA
The voltage in RREF: 2.2563V
The voltage in ROUTPUT: 290.18597MV
The Current Reference: 2.2563MA
The Output Current: 2.7376MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.00606V
Current from DPS1A is: 21.98828MA
The voltage in RREF: 2.31073V
The voltage in ROUTPUT: 296.20756MV
The Current Reference: 2.31073MA
The Output Current: 2.79441MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.0511V
Current from DPS1A is: 21.98828MA
The voltage in RREF: 2.35765V
The voltage in ROUTPUT: 301.46277MV
The Current Reference: 2.35765MA
The Output Current: 2.84399MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.10257V
Current from DPS1A is: 21.98828MA
The voltage in RREF: 2.40551V
The voltage in ROUTPUT: 306.68669MV
The Current Reference: 2.40551MA
The Output Current: 2.89327MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.14761V
Current from DPS1A is: 21.98828MA
The voltage in RREF: 2.45243V
The voltage in ROUTPUT: 311.91062MV

The Current Reference: 2.45243MA
The Output Current: 2.94255MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.19909V
Current from DPS1A is: 21.98828MA
The voltage in RREF: 2.50092V
The voltage in ROUTPUT: 317.36915MV
The Current Reference: 2.50092MA
The Output Current: 2.99405MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.25699V
Current from DPS1A is: 21.98828MA
The voltage in RREF: 2.55628V
The voltage in ROUTPUT: 323.24998MV
The Current Reference: 2.55628MA
The Output Current: 3.04953MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.30203V
Current from DPS1A is: 23.01099MA
The voltage in RREF: 2.60571V
The voltage in ROUTPUT: 328.67723MV
The Current Reference: 2.60571MA
The Output Current: 3.10073MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.34707V
Current from DPS1A is: 23.01099MA
The voltage in RREF: 2.65294V
The voltage in ROUTPUT: 333.86988MV
The Current Reference: 2.65294MA
The Output Current: 3.14972MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.39855V
Current from DPS1A is: 23.01099MA
The voltage in RREF: 2.7033V
The voltage in ROUTPUT: 339.26585MV
The Current Reference: 2.7033MA
The Output Current: 3.20062MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.45002V
Current from DPS1A is: 23.01099MA
The voltage in RREF: 2.75023V
The voltage in ROUTPUT: 344.39593MV
The Current Reference: 2.75023MA
The Output Current: 3.24902MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.50149V
Current from DPS1A is: 23.01099MA
The voltage in RREF: 2.80434V
The voltage in ROUTPUT: 350.19856MV
The Current Reference: 2.80434MA
The Output Current: 3.30376MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS

Voltage of DPS1A is: 3.55297V
Current from DPS1A is: 24.0337MA
The voltage in RREF: 2.85001V
The voltage in ROUTPUT: 355.2348MV
The Current Reference: 2.85001MA
The Output Current: 3.35127MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.60444V
Current from DPS1A is: 23.01099MA
The voltage in RREF: 2.901V
The voltage in ROUTPUT: 360.55257MV
The Current Reference: 2.901MA
The Output Current: 3.40144MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.64948V
Current from DPS1A is: 24.0337MA
The voltage in RREF: 2.94855V
The voltage in ROUTPUT: 365.76085MV
The Current Reference: 2.94855MA
The Output Current: 3.45057MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.70739V
Current from DPS1A is: 24.0337MA
The voltage in RREF: 3.00485V
The voltage in ROUTPUT: 371.79808MV
The Current Reference: 3.00485MA
The Output Current: 3.50753MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.75242V
Current from DPS1A is: 24.0337MA
The voltage in RREF: 3.0524V
The voltage in ROUTPUT: 376.75612MV
The Current Reference: 3.0524MA
The Output Current: 3.5543MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.8039V
Current from DPS1A is: 24.0337MA
The voltage in RREF: 3.10214V
The voltage in ROUTPUT: 381.98005MV
The Current Reference: 3.10214MA
The Output Current: 3.60359MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.84894V
Current from DPS1A is: 25.05641MA
The voltage in RREF: 3.14968V
The voltage in ROUTPUT: 387.21962MV
The Current Reference: 3.14968MA
The Output Current: 3.65302MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.90041V
Current from DPS1A is: 25.05641MA
The voltage in RREF: 3.19754V

The voltage in ROUTPUT: 392.16201MV
The Current Reference: 3.19754MA
The Output Current: 3.69964MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.95188V
Current from DPS1A is: 25.05641MA
The voltage in RREF: 3.25103V
The voltage in ROUTPUT: 397.73003MV
The Current Reference: 3.25103MA
The Output Current: 3.75217MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.00336V
Current from DPS1A is: 25.05641MA
The voltage in RREF: 3.30077V
The voltage in ROUTPUT: 403.14164MV
The Current Reference: 3.30077MA
The Output Current: 3.80322MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.0484V
Current from DPS1A is: 25.05641MA
The voltage in RREF: 3.34957V
The voltage in ROUTPUT: 408.33429MV
The Current Reference: 3.34957MA
The Output Current: 3.85221MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.09987V
Current from DPS1A is: 25.05641MA
The voltage in RREF: 3.39868V
The voltage in ROUTPUT: 413.33925MV
The Current Reference: 3.39868MA
The Output Current: 3.89943MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.15134V
Current from DPS1A is: 25.05641MA
The voltage in RREF: 3.44591V
The voltage in ROUTPUT: 418.45369MV
The Current Reference: 3.44591MA
The Output Current: 3.94768MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.20925V
Current from DPS1A is: 26.07912MA
The voltage in RREF: 3.50222V
The voltage in ROUTPUT: 424.31888MV
The Current Reference: 3.50222MA
The Output Current: 4.00301MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.25429V
Current from DPS1A is: 26.07912MA
The voltage in RREF: 3.54977V
The voltage in ROUTPUT: 429.22999MV
The Current Reference: 3.54977MA
The Output Current: 4.04934MA

FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.30576V
Current from DPS1A is: 26.07912MA
The voltage in RREF: 3.59794V
The voltage in ROUTPUT: 434.29751MV
The Current Reference: 3.59794MA
The Output Current: 4.09715MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.3508V
Current from DPS1A is: 26.07912MA
The voltage in RREF: 3.64643V
The voltage in ROUTPUT: 439.38067MV
The Current Reference: 3.64643MA
The Output Current: 4.1451MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.40228V
Current from DPS1A is: 27.10183MA
The voltage in RREF: 3.69522V
The voltage in ROUTPUT: 444.41691MV
The Current Reference: 3.69522MA
The Output Current: 4.19261MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.45375V
Current from DPS1A is: 26.07912MA
The voltage in RREF: 3.74934V
The voltage in ROUTPUT: 450.09442MV
The Current Reference: 3.74934MA
The Output Current: 4.24617MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.50522V
Current from DPS1A is: 27.10183MA
The voltage in RREF: 3.7972V
The voltage in ROUTPUT: 455.16194MV
The Current Reference: 3.7972MA
The Output Current: 4.29398MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.55026V
Current from DPS1A is: 27.10183MA
The voltage in RREF: 3.84569V
The voltage in ROUTPUT: 460.22946MV
The Current Reference: 3.84569MA
The Output Current: 4.34179MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.60173V
Current from DPS1A is: 27.10183MA
The voltage in RREF: 3.89417V
The voltage in ROUTPUT: 465.1875MV
The Current Reference: 3.89417MA
The Output Current: 4.38856MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.64677V
Current from DPS1A is: 27.10183MA

The voltage in RREF: 3.94172V
The voltage in ROUTPUT: 470.27066MV
The Current Reference: 3.94172MA
The Output Current: 4.43652MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.70468V
Current from DPS1A is: 27.10183MA
The voltage in RREF: 3.99677V
The voltage in ROUTPUT: 475.94816MV
The Current Reference: 3.99677MA
The Output Current: 4.49008MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.74972V
Current from DPS1A is: 27.10183MA
The voltage in RREF: 4.04463V
The voltage in ROUTPUT: 480.87491MV
The Current Reference: 4.04463MA
The Output Current: 4.53656MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.80119V
Current from DPS1A is: 28.12454MA
The voltage in RREF: 4.09406V
The voltage in ROUTPUT: 486.0832MV
The Current Reference: 4.09406MA
The Output Current: 4.58569MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.85267V
Current from DPS1A is: 27.10183MA
The voltage in RREF: 4.14285V
The voltage in ROUTPUT: 491.27584MV
The Current Reference: 4.14285MA
The Output Current: 4.63468MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.89771V
Current from DPS1A is: 27.10183MA
The voltage in RREF: 4.19134V
The voltage in ROUTPUT: 496.17132MV
The Current Reference: 4.19134MA
The Output Current: 4.68086MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.95561V
Current from DPS1A is: 28.12454MA
The voltage in RREF: 4.24608V
The voltage in ROUTPUT: 501.89574MV
The Current Reference: 4.24608MA
The Output Current: 4.73487MA

FUNCTIONAL_2 OUTPUT DATA FILE DC SWEEP FOR BIPOLAR CURRENT-MIRROR WITH BASE COMPENSATION

FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 52.77929MV
Current from DPS1A is 13.80659MA
The voltage in RREF 312.80995UV
The voltage in ROUTPUT -54.74174UV
The Current Reference 312.80995NA
The Output Current -536.68374NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 104.25262MV
Current from DPS1A is 13.80659MA
The voltage in RREF 0V
The voltage in ROUTPUT -7.82025UV
The Current Reference 0A
The Output Current -76.66911NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 155.72595MV
Current from DPS1A is 13.80659MA
The voltage in RREF -312.80995UV
The voltage in ROUTPUT -23.46075UV
The Current Reference -312.80995NA
The Output Current -230.00732NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 200.76511MV
Current from DPS1A is 13.80659MA
The voltage in RREF 312.80995UV
The voltage in ROUTPUT -7.82025UV
The Current Reference 312.80995NA
The Output Current -76.66911NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 252.23844MV
Current from DPS1A is 13.80659MA
The voltage in RREF -312.80995UV
The voltage in ROUTPUT 7.82025UV
The Current Reference -312.80995NA
The Output Current 76.66911NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 303.71177MV
Current from DPS1A is 12.78388MA
The voltage in RREF -312.80995UV
The voltage in ROUTPUT -23.46075UV
The Current Reference -312.80995NA
The Output Current -230.00732NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS

Voltage of DPS1A is	355.1851MV
Current from DPS1A is	13.80659MA
The voltage in RREF	0V
The voltage in ROUTPUT	-23.46075UV
The Current Reference	0A
The Output Current	-230.00732NA

FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS

Voltage of DPS1A is	406.65843MV
Current from DPS1A is	13.80659MA
The voltage in RREF	625.6199UV
The voltage in ROUTPUT	-7.82025UV
The Current Reference	625.6199NA
The Output Current	-76.66911NA

FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS

Voltage of DPS1A is	451.69759MV
Current from DPS1A is	13.80659MA
The voltage in RREF	0V
The voltage in ROUTPUT	-7.82025UV
The Current Reference	0A
The Output Current	-76.66911NA

FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS

Voltage of DPS1A is	503.17092MV
Current from DPS1A is	13.80659MA
The voltage in RREF	312.80995UV
The voltage in ROUTPUT	-23.46075UV
The Current Reference	312.80995NA
The Output Current	-230.00732NA

FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS

Voltage of DPS1A is	554.64425MV
Current from DPS1A is	13.80659MA
The voltage in RREF	0V
The voltage in ROUTPUT	7.82025UV
The Current Reference	0A
The Output Current	76.66911NA

FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS

Voltage of DPS1A is	606.11757MV
Current from DPS1A is	13.80659MA
The voltage in RREF	312.80995UV
The voltage in ROUTPUT	-23.46075UV
The Current Reference	312.80995NA
The Output Current	-230.00732NA

FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS

Voltage of DPS1A is	657.5909MV
Current from DPS1A is	13.80659MA
The voltage in RREF	0V
The voltage in ROUTPUT	-7.82025UV
The Current Reference	0A
The Output Current	-76.66911NA

FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS

Voltage of DPS1A is	702.63007MV
Current from DPS1A is	13.80659MA
The voltage in RREF	312.80995UV

The voltage in ROUTPUT -7.82025UV
The Current Reference 312.80995NA
The Output Current -76.66911NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 754.10339MV
Current from DPS1A is 13.80659MA
The voltage in RREF -312.80995UV
The voltage in ROUTPUT 39.10124UV
The Current Reference -312.80995NA
The Output Current 383.34553NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 805.57672MV
Current from DPS1A is 13.80659MA
The voltage in RREF 312.80995UV
The voltage in ROUTPUT 54.74174UV
The Current Reference 312.80995NA
The Output Current 536.68374NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 857.05005MV
Current from DPS1A is 13.80659MA
The voltage in RREF 312.80995UV
The voltage in ROUTPUT 132.94423UV
The Current Reference 312.80995NA
The Output Current 1.30337UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 902.08921MV
Current from DPS1A is 13.80659MA
The voltage in RREF 938.42985UV
The voltage in ROUTPUT 414.47318UV
The Current Reference 938.42985NA
The Output Current 4.06346UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 953.56254MV
Current from DPS1A is 13.80659MA
The voltage in RREF 1.87686MV
The voltage in ROUTPUT 1.02445MV
The Current Reference 1.87686UA
The Output Current 10.04365UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 1.00504V
Current from DPS1A is 13.80659MA
The voltage in RREF 5.31777MV
The voltage in ROUTPUT 2.11929MV
The Current Reference 5.31777UA
The Output Current 20.77733UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 1.05651V
Current from DPS1A is 14.8293MA
The voltage in RREF 13.45083MV
The voltage in ROUTPUT 4.19947MV
The Current Reference 13.45083UA
The Output Current 41.17131UA

FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 1.10798V
Current from DPS1A is 14.8293MA
The voltage in RREF 27.84009MV
The voltage in ROUTPUT 6.67067MV
The Current Reference 27.84009UA
The Output Current 65.39875UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 1.15302V
Current from DPS1A is 14.8293MA
The voltage in RREF 49.11116MV
The voltage in ROUTPUT 9.67365MV
The Current Reference 49.11116UA
The Output Current 94.83968UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 1.19806V
Current from DPS1A is 14.8293MA
The voltage in RREF 74.44877MV
The voltage in ROUTPUT 12.92687MV
The Current Reference 74.44877UA
The Output Current 126.73403UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 1.24953V
Current from DPS1A is 14.8293MA
The voltage in RREF 106.04257MV
The voltage in ROUTPUT 16.57111MV
The Current Reference 106.04257UA
The Output Current 162.46183UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 1.30744V
Current from DPS1A is 14.8293MA
The voltage in RREF 145.45663MV
The voltage in ROUTPUT 20.87224MV
The Current Reference 145.45663UA
The Output Current 204.62984UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 1.35248V
Current from DPS1A is 14.8293MA
The voltage in RREF 183.30663MV
The voltage in ROUTPUT 24.87621MV
The Current Reference 183.30663UA
The Output Current 243.88442UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 1.40395V
Current from DPS1A is 14.8293MA
The voltage in RREF 221.15663MV
The voltage in ROUTPUT 28.86454MV
The Current Reference 221.15663UA
The Output Current 282.98567UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 1.44899V
Current from DPS1A is 15.85201MA

The voltage in RREF 262.44755MV
The voltage in ROUTPUT 33.11875MV
The Current Reference 262.44755UA
The Output Current 324.69366UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 1.50047V
Current from DPS1A is 14.8293MA
The voltage in RREF 303.11284MV
The voltage in ROUTPUT 37.26349MV
The Current Reference 303.11284UA
The Output Current 365.32829UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 1.55837V
Current from DPS1A is 15.85201MA
The voltage in RREF 348.78309MV
The voltage in ROUTPUT 41.98692MV
The Current Reference 348.78309UA
The Output Current 411.63643UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 1.60341V
Current from DPS1A is 15.85201MA
The voltage in RREF 389.7612MV
The voltage in ROUTPUT 46.0378MV
The Current Reference 389.7612UA
The Output Current 451.35102UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 1.64845V
Current from DPS1A is 15.85201MA
The voltage in RREF 435.11864MV
The voltage in ROUTPUT 50.65175MV
The Current Reference 435.11864UA
The Output Current 496.58579UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 1.69993V
Current from DPS1A is 15.85201MA
The voltage in RREF 479.53765MV
The voltage in ROUTPUT 55.10929MV
The Current Reference 479.53765UA
The Output Current 540.28718UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 1.74496V
Current from DPS1A is 16.87473MA
The voltage in RREF 523.95666MV
The voltage in ROUTPUT 59.69196MV
The Current Reference 523.95666UA
The Output Current 585.21528UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 1.80931V
Current from DPS1A is 16.87473MA
The voltage in RREF 574.63188MV
The voltage in ROUTPUT 64.77512MV
The Current Reference 574.63188UA

The Output Current 635.0502UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 1.85435V
Current from DPS1A is 16.87473MA
The voltage in RREF 621.55337MV
The voltage in ROUTPUT 69.45163MV
The Current Reference 621.55337UA
The Output Current 680.89832UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 1.90582V
Current from DPS1A is 16.87473MA
The voltage in RREF 665.65957MV
The voltage in ROUTPUT 73.94045MV
The Current Reference 665.65957UA
The Output Current 724.90639UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 1.95729V
Current from DPS1A is 16.87473MA
The voltage in RREF 711.32982MV
The voltage in ROUTPUT 78.52312MV
The Current Reference 711.32982UA
The Output Current 769.83448UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 2.00877V
Current from DPS1A is 16.87473MA
The voltage in RREF 762.00504MV
The voltage in ROUTPUT 83.55936MV
The Current Reference 762.00504UA
The Output Current 819.20939UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 2.0538V
Current from DPS1A is 16.87473MA
The voltage in RREF 808.92653MV
The voltage in ROUTPUT 88.37663MV
The Current Reference 808.92653UA
The Output Current 866.43756UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 2.10528V
Current from DPS1A is 17.89744MA
The voltage in RREF 854.90959MV
The voltage in ROUTPUT 93.05314MV
The Current Reference 854.90959UA
The Output Current 912.28568UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 2.15032V
Current from DPS1A is 16.87473MA
The voltage in RREF 901.20546MV
The voltage in ROUTPUT 97.76093MV
The Current Reference 901.20546UA
The Output Current 958.44048UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 2.20179V

Current from DPS1A is	16.87473MA
The voltage in RREF	946.56291MV
The voltage in ROUTPUT	102.39052MV
The Current Reference	946.56291UA
The Output Current	1.00383MA

FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS

Voltage of DPS1A is	2.25326V
Current from DPS1A is	17.89744MA
The voltage in RREF	1.00099V
The voltage in ROUTPUT	107.78649MV
The Current Reference	1.00099MA
The Output Current	1.05673MA

FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS

Voltage of DPS1A is	2.30474V
Current from DPS1A is	17.89744MA
The voltage in RREF	1.04635V
The voltage in ROUTPUT	112.49428MV
The Current Reference	1.04635MA
The Output Current	1.10289MA

FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS

Voltage of DPS1A is	2.35621V
Current from DPS1A is	17.89744MA
The voltage in RREF	1.09202V
The voltage in ROUTPUT	117.26463MV
The Current Reference	1.09202MA
The Output Current	1.14965MA

FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS

Voltage of DPS1A is	2.40125V
Current from DPS1A is	18.92015MA
The voltage in RREF	1.13769V
The voltage in ROUTPUT	121.98806MV
The Current Reference	1.13769MA
The Output Current	1.19596MA

FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS

Voltage of DPS1A is	2.45272V
Current from DPS1A is	18.92015MA
The voltage in RREF	1.1868V
The voltage in ROUTPUT	126.96174MV
The Current Reference	1.1868MA
The Output Current	1.24472MA

FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS

Voltage of DPS1A is	2.5042V
Current from DPS1A is	17.89744MA
The voltage in RREF	1.23841V
The voltage in ROUTPUT	132.45155MV
The Current Reference	1.23841MA
The Output Current	1.29854MA

FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS

Voltage of DPS1A is	2.55567V
Current from DPS1A is	18.92015MA
The voltage in RREF	1.28534V
The voltage in ROUTPUT	137.25319MV

The Current Reference 1.28534MA
The Output Current 1.34562MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 2.60071V
Current from DPS1A is 18.92015MA
The voltage in RREF 1.33226V
The voltage in ROUTPUT 142.14866MV
The Current Reference 1.33226MA
The Output Current 1.39361MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 2.65218V
Current from DPS1A is 18.92015MA
The voltage in RREF 1.3798V
The voltage in ROUTPUT 147.1849MV
The Current Reference 1.3798MA
The Output Current 1.44299MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 2.70366V
Current from DPS1A is 18.92015MA
The voltage in RREF 1.42673V
The voltage in ROUTPUT 152.14294MV
The Current Reference 1.42673MA
The Output Current 1.4916MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 2.75513V
Current from DPS1A is 18.92015MA
The voltage in RREF 1.47959V
The voltage in ROUTPUT 157.47635MV
The Current Reference 1.47959MA
The Output Current 1.54389MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 2.8066V
Current from DPS1A is 18.92015MA
The voltage in RREF 1.52714V
The voltage in ROUTPUT 162.49695MV
The Current Reference 1.52714MA
The Output Current 1.59311MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 2.85808V
Current from DPS1A is 19.94286MA
The voltage in RREF 1.57531V
The voltage in ROUTPUT 167.58011MV
The Current Reference 1.57531MA
The Output Current 1.64294MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 2.90311V
Current from DPS1A is 19.94286MA
The voltage in RREF 1.62255V
The voltage in ROUTPUT 172.63199MV
The Current Reference 1.62255MA
The Output Current 1.69247MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS

Voltage of DPS1A is	2.94815V
Current from DPS1A is	18.92015MA
The voltage in RREF	1.66884V
The voltage in ROUTPUT	177.49619MV
The Current Reference	1.66884MA
The Output Current	1.74016MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS	
Voltage of DPS1A is	3.00606V
Current from DPS1A is	19.94286MA
The voltage in RREF	1.72139V
The voltage in ROUTPUT	183.37701MV
The Current Reference	1.72139MA
The Output Current	1.79781MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS	
Voltage of DPS1A is	3.0511V
Current from DPS1A is	19.94286MA
The voltage in RREF	1.76863V
The voltage in ROUTPUT	188.5853MV
The Current Reference	1.76863MA
The Output Current	1.84888MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS	
Voltage of DPS1A is	3.10257V
Current from DPS1A is	19.94286MA
The voltage in RREF	1.81555V
The voltage in ROUTPUT	193.57462MV
The Current Reference	1.81555MA
The Output Current	1.89779MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS	
Voltage of DPS1A is	3.14761V
Current from DPS1A is	20.96557MA
The voltage in RREF	1.86184V
The voltage in ROUTPUT	198.75162MV
The Current Reference	1.86184MA
The Output Current	1.94855MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS	
Voltage of DPS1A is	3.19909V
Current from DPS1A is	19.94286MA
The voltage in RREF	1.91033V
The voltage in ROUTPUT	204.10067MV
The Current Reference	1.91033MA
The Output Current	2.00099MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS	
Voltage of DPS1A is	3.25056V
Current from DPS1A is	20.96557MA
The voltage in RREF	1.96413V
The voltage in ROUTPUT	209.91894MV
The Current Reference	1.96413MA
The Output Current	2.05803MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS	
Voltage of DPS1A is	3.30203V
Current from DPS1A is	20.96557MA
The voltage in RREF	2.01293V

The voltage in ROUTPUT 215.45567MV
The Current Reference 2.01293MA
The Output Current 2.11231MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 3.35351V
Current from DPS1A is 20.96557MA
The voltage in RREF 2.06017V
The voltage in ROUTPUT 220.89857MV
The Current Reference 2.06017MA
The Output Current 2.16567MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 3.39855V
Current from DPS1A is 20.96557MA
The voltage in RREF 2.10865V
The voltage in ROUTPUT 226.15377MV
The Current Reference 2.10865MA
The Output Current 2.21719MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 3.45002V
Current from DPS1A is 21.98828MA
The voltage in RREF 2.15651V
The voltage in ROUTPUT 231.48718MV
The Current Reference 2.15651MA
The Output Current 2.26948MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 3.50149V
Current from DPS1A is 21.98828MA
The voltage in RREF 2.20875V
The voltage in ROUTPUT 237.64954MV
The Current Reference 2.20875MA
The Output Current 2.3299MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 3.55297V
Current from DPS1A is 21.98828MA
The voltage in RREF 2.25473V
The voltage in ROUTPUT 242.93603MV
The Current Reference 2.25473MA
The Output Current 2.38173MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 3.60444V
Current from DPS1A is 21.98828MA
The voltage in RREF 2.30416V
The voltage in ROUTPUT 248.36328MV
The Current Reference 2.30416MA
The Output Current 2.43493MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 3.64948V
Current from DPS1A is 21.98828MA
The voltage in RREF 2.35171V
The voltage in ROUTPUT 254.18154MV
The Current Reference 2.35171MA
The Output Current 2.49198MA

FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 3.70739V
Current from DPS1A is 21.98828MA
The voltage in RREF 2.4077V
The voltage in ROUTPUT 260.60979MV
The Current Reference 2.4077MA
The Output Current 2.555MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 3.75242V
Current from DPS1A is 21.98828MA
The voltage in RREF 2.45431V
The voltage in ROUTPUT 265.89628MV
The Current Reference 2.45431MA
The Output Current 2.60683MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 3.8039V
Current from DPS1A is 21.98828MA
The voltage in RREF 2.50311V
The voltage in ROUTPUT 271.96479MV
The Current Reference 2.50311MA
The Output Current 2.66632MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 3.84894V
Current from DPS1A is 23.01099MA
The voltage in RREF 2.55128V
The voltage in ROUTPUT 278.01766MV
The Current Reference 2.55128MA
The Output Current 2.72566MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 3.90041V
Current from DPS1A is 23.01099MA
The voltage in RREF 2.5982V
The voltage in ROUTPUT 283.50748MV
The Current Reference 2.5982MA
The Output Current 2.77949MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 3.95188V
Current from DPS1A is 23.01099MA
The voltage in RREF 2.65138V
The voltage in ROUTPUT 290.02956MV
The Current Reference 2.65138MA
The Output Current 2.84343MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 4.00336V
Current from DPS1A is 23.01099MA
The voltage in RREF 2.69986V
The voltage in ROUTPUT 296.59857MV
The Current Reference 2.69986MA
The Output Current 2.90783MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 4.05483V
Current from DPS1A is 23.01099MA

The voltage in RREF 2.74897V
The voltage in ROUTPUT 302.60452MV
The Current Reference 2.74897MA
The Output Current 2.96671MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 4.09987V
Current from DPS1A is 23.01099MA
The voltage in RREF 2.79683V
The voltage in ROUTPUT 308.0787MV
The Current Reference 2.79683MA
The Output Current 3.02038MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 4.15134V
Current from DPS1A is 23.01099MA
The voltage in RREF 2.84469V
The voltage in ROUTPUT 314.53822MV
The Current Reference 2.84469MA
The Output Current 3.08371MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 4.20282V
Current from DPS1A is 24.0337MA
The voltage in RREF 2.90006V
The voltage in ROUTPUT 321.49825MV
The Current Reference 2.90006MA
The Output Current 3.15194MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 4.25429V
Current from DPS1A is 24.0337MA
The voltage in RREF 2.9473V
The voltage in ROUTPUT 327.12882MV
The Current Reference 2.9473MA
The Output Current 3.20715MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 4.30576V
Current from DPS1A is 24.0337MA
The voltage in RREF 2.99516V
The voltage in ROUTPUT 333.40066MV
The Current Reference 2.99516MA
The Output Current 3.26863MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 4.3508V
Current from DPS1A is 24.0337MA
The voltage in RREF 3.0427V
The voltage in ROUTPUT 340.12608MV
The Current Reference 3.0427MA
The Output Current 3.33457MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is 4.40228V
Current from DPS1A is 25.05641MA
The voltage in RREF 3.09119V
The voltage in ROUTPUT 346.0069MV
The Current Reference 3.09119MA

The Output Current	3.39222MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS	
Voltage of DPS1A is	4.45375V
Current from DPS1A is	25.05641MA
The voltage in RREF	3.14499V
The voltage in ROUTPUT	353.37358MV
The Current Reference	3.14499MA
The Output Current	3.46445MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS	
Voltage of DPS1A is	4.50522V
Current from DPS1A is	25.05641MA
The voltage in RREF	3.19254V
The voltage in ROUTPUT	360.69333MV
The Current Reference	3.19254MA
The Output Current	3.53621MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS	
Voltage of DPS1A is	4.55026V
Current from DPS1A is	25.05641MA
The voltage in RREF	3.2404V
The voltage in ROUTPUT	367.34054MV
The Current Reference	3.2404MA
The Output Current	3.60138MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS	
Voltage of DPS1A is	4.5953V
Current from DPS1A is	25.05641MA
The voltage in RREF	3.28888V
The voltage in ROUTPUT	373.29957MV
The Current Reference	3.28888MA
The Output Current	3.6598MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS	
Voltage of DPS1A is	4.64677V
Current from DPS1A is	26.07912MA
The voltage in RREF	3.33612V
The voltage in ROUTPUT	381.01034MV
The Current Reference	3.33612MA
The Output Current	3.7354MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS	
Voltage of DPS1A is	4.70468V
Current from DPS1A is	26.07912MA
The voltage in RREF	3.39023V
The voltage in ROUTPUT	388.68982MV
The Current Reference	3.39023MA
The Output Current	3.81068MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS	
Voltage of DPS1A is	4.74972V
Current from DPS1A is	26.07912MA
The voltage in RREF	3.43841V
The voltage in ROUTPUT	394.57065MV
The Current Reference	3.43841MA
The Output Current	3.86834MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS	
Voltage of DPS1A is	4.80119V

Current from DPS1A is	26.07912MA
The voltage in RREF	3.48752V
The voltage in ROUTPUT	401.82784MV
The Current Reference	3.48752MA
The Output Current	3.93949MA

FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS

Voltage of DPS1A is	4.85267V
Current from DPS1A is	26.07912MA
The voltage in RREF	3.53632V
The voltage in ROUTPUT	409.9609MV
The Current Reference	3.53632MA
The Output Current	4.01922MA

FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS

Voltage of DPS1A is	4.89771V
Current from DPS1A is	26.07912MA
The voltage in RREF	3.58386V
The voltage in ROUTPUT	416.12325MV
The Current Reference	3.58386MA
The Output Current	4.07964MA

FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS

Voltage of DPS1A is	4.95561V
Current from DPS1A is	27.10183MA
The voltage in RREF	3.63861V
The voltage in ROUTPUT	424.5222MV
The Current Reference	3.63861MA
The Output Current	4.16198MA

FUNTIONAL_TEST_2 OUPUT DATA FILE DC SWEEP FOR WILSON CURRENT-MIRROR.

FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 46.34513MV
Current from DPS1A is: 13.80659MA
The voltage in RREF: 312.80995UV
The voltage in ROUTPUT: -39.10124UV
The Current Reference: 312.80995NA
The Output Current: -368.87966NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 104.25262MV
Current from DPS1A is: 13.80659MA
The voltage in RREF: 0V
The voltage in ROUTPUT: -23.46075UV
The Current Reference: 0A
The Output Current: -221.32779NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 155.72595MV
Current from DPS1A is: 13.80659MA
The voltage in RREF: 0V
The voltage in ROUTPUT: -23.46075UV
The Current Reference: 0A
The Output Current: -221.32779NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 207.19928MV
Current from DPS1A is: 13.80659MA
The voltage in RREF: 312.80995UV
The voltage in ROUTPUT: 7.82025UV
The Current Reference: 312.80995NA
The Output Current: 73.77593NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 252.23844MV
Current from DPS1A is: 12.78388MA
The voltage in RREF: 312.80995UV
The voltage in ROUTPUT: 7.82025UV
The Current Reference: 312.80995NA
The Output Current: 73.77593NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 303.71177MV
Current from DPS1A is: 13.80659MA
The voltage in RREF: 0V
The voltage in ROUTPUT: -23.46075UV
The Current Reference: 0A
The Output Current: -221.32779NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 355.1851MV
Current from DPS1A is: 13.80659MA

The voltage in RREF: 312.80995UV
The voltage in ROUTPUT: -23.46075UV
The Current Reference: 312.80995NA
The Output Current: -221.32779NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 406.65843MV
Current from DPS1A is: 13.80659MA
The voltage in RREF: 312.80995UV
The voltage in ROUTPUT: -7.82025UV
The Current Reference: 312.80995NA
The Output Current: -73.77593NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 451.69759MV
Current from DPS1A is: 13.80659MA
The voltage in RREF: -625.6199UV
The voltage in ROUTPUT: -23.46075UV
The Current Reference: -625.6199NA
The Output Current: -221.32779NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 503.17092MV
Current from DPS1A is: 13.80659MA
The voltage in RREF: -312.80995UV
The voltage in ROUTPUT: -7.82025UV
The Current Reference: -312.80995NA
The Output Current: -73.77593NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 554.64425MV
Current from DPS1A is: 13.80659MA
The voltage in RREF: 0V
The voltage in ROUTPUT: -7.82025UV
The Current Reference: 0A
The Output Current: -73.77593NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 606.11757MV
Current from DPS1A is: 13.80659MA
The voltage in RREF: 0V
The voltage in ROUTPUT: 7.82025UV
The Current Reference: 0A
The Output Current: 73.77593NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 651.15674MV
Current from DPS1A is: 13.80659MA
The voltage in RREF: 0V
The voltage in ROUTPUT: -7.82025UV
The Current Reference: 0A
The Output Current: -73.77593NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 702.63007MV
Current from DPS1A is: 13.80659MA
The voltage in RREF: -312.80995UV
The voltage in ROUTPUT: 7.82025UV
The Current Reference: -312.80995NA

The Output Current: 73.77593NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 747.66923MV
Current from DPS1A is: 13.80659MA
The voltage in RREF: 0V
The voltage in ROUTPUT: -7.82025UV
The Current Reference: 0A
The Output Current: -73.77593NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 805.57672MV
Current from DPS1A is: 13.80659MA
The voltage in RREF: 312.80995UV
The voltage in ROUTPUT: -23.46075UV
The Current Reference: 312.80995NA
The Output Current: -221.32779NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 857.05005MV
Current from DPS1A is: 13.80659MA
The voltage in RREF: -312.80995UV
The voltage in ROUTPUT: 23.46075UV
The Current Reference: -312.80995NA
The Output Current: 221.32779NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 908.52338MV
Current from DPS1A is: 13.80659MA
The voltage in RREF: 0V
The voltage in ROUTPUT: 70.38224UV
The Current Reference: 0A
The Output Current: 663.98338NA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 953.56254MV
Current from DPS1A is: 13.80659MA
The voltage in RREF: 312.80995UV
The voltage in ROUTPUT: 195.50622UV
The Current Reference: 312.80995NA
The Output Current: 1.8444UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.00504V
Current from DPS1A is: 13.80659MA
The voltage in RREF: 938.42985UV
The voltage in ROUTPUT: 492.67567UV
The Current Reference: 938.42985NA
The Output Current: 4.64788UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.05651V
Current from DPS1A is: 13.80659MA
The voltage in RREF: 2.50248MV
The voltage in ROUTPUT: 1.29034MV
The Current Reference: 2.50248UA
The Output Current: 12.17303UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.10155V

Current from DPS1A is: 14.8293MA
The voltage in RREF: 5.94339MV
The voltage in ROUTPUT: 2.71363MV
The Current Reference: 5.94339UA
The Output Current: 25.60025UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.15302V
Current from DPS1A is: 13.80659MA
The voltage in RREF: 13.45083MV
The voltage in ROUTPUT: 5.02842MV
The Current Reference: 13.45083UA
The Output Current: 47.43792UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.2045V
Current from DPS1A is: 14.8293MA
The voltage in RREF: 26.27604MV
The voltage in ROUTPUT: 7.98447MV
The Current Reference: 26.27604UA
The Output Current: 75.32523UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.24953V
Current from DPS1A is: 14.8293MA
The voltage in RREF: 46.92149MV
The voltage in ROUTPUT: 11.78511MV
The Current Reference: 46.92149UA
The Output Current: 111.18033UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.30744V
Current from DPS1A is: 14.8293MA
The voltage in RREF: 76.32563MV
The voltage in ROUTPUT: 16.61803MV
The Current Reference: 76.32563UA
The Output Current: 156.77385UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.35892V
Current from DPS1A is: 14.8293MA
The voltage in RREF: 106.981MV
The voltage in ROUTPUT: 21.3571MV
The Current Reference: 106.981UA
The Output Current: 201.48207UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.40395V
Current from DPS1A is: 14.8293MA
The voltage in RREF: 140.45167MV
The voltage in ROUTPUT: 26.17437MV
The Current Reference: 140.45167UA
The Output Current: 246.92804UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.44899V
Current from DPS1A is: 14.8293MA
The voltage in RREF: 177.67605MV
The voltage in ROUTPUT: 31.33574MV

The Current Reference: 177.67605UA
The Output Current: 295.62016UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.5069V
Current from DPS1A is: 14.8293MA
The voltage in RREF: 214.90043MV
The voltage in ROUTPUT: 36.51274MV
The Current Reference: 214.90043UA
The Output Current: 344.45982UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.55837V
Current from DPS1A is: 15.85201MA
The voltage in RREF: 259.00664MV
The voltage in ROUTPUT: 42.89406MV
The Current Reference: 259.00664UA
The Output Current: 404.66098UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.60341V
Current from DPS1A is: 14.8293MA
The voltage in RREF: 297.16945MV
The voltage in ROUTPUT: 49.0877MV
The Current Reference: 297.16945UA
The Output Current: 463.09152UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.64845V
Current from DPS1A is: 15.85201MA
The voltage in RREF: 341.58846MV
The voltage in ROUTPUT: 56.17285MV
The Current Reference: 341.58846UA
The Output Current: 529.93251UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.69993V
Current from DPS1A is: 15.85201MA
The voltage in RREF: 384.13062MV
The voltage in ROUTPUT: 63.03902MV
The Current Reference: 384.13062UA
The Output Current: 594.70778UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.7514V
Current from DPS1A is: 16.87473MA
The voltage in RREF: 428.54963MV
The voltage in ROUTPUT: 70.04597MV
The Current Reference: 428.54963UA
The Output Current: 660.81102UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.80931V
Current from DPS1A is: 16.87473MA
The voltage in RREF: 477.03517MV
The voltage in ROUTPUT: 77.94442MV
The Current Reference: 477.03517UA
The Output Current: 735.32471UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS

Voltage of DPS1A is: 1.85435V
Current from DPS1A is: 16.87473MA
The voltage in RREF: 522.39261MV
The voltage in ROUTPUT: 85.15469MV
The Current Reference: 522.39261UA
The Output Current: 803.34612UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.90582V
Current from DPS1A is: 16.87473MA
The voltage in RREF: 566.49882MV
The voltage in ROUTPUT: 92.02087MV
The Current Reference: 566.49882UA
The Output Current: 868.12138UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 1.95086V
Current from DPS1A is: 16.87473MA
The voltage in RREF: 610.29221MV
The voltage in ROUTPUT: 98.93397MV
The Current Reference: 610.29221UA
The Output Current: 933.33931UA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.00877V
Current from DPS1A is: 17.89744MA
The voltage in RREF: 660.02899MV
The voltage in ROUTPUT: 106.59781MV
The Current Reference: 660.02899UA
The Output Current: 1.00564MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.0538V
Current from DPS1A is: 17.89744MA
The voltage in RREF: 706.63767MV
The voltage in ROUTPUT: 113.76116MV
The Current Reference: 706.63767UA
The Output Current: 1.07322MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.10528V
Current from DPS1A is: 17.89744MA
The voltage in RREF: 751.3695MV
The voltage in ROUTPUT: 120.65862MV
The Current Reference: 751.3695UA
The Output Current: 1.13829MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.15675V
Current from DPS1A is: 17.89744MA
The voltage in RREF: 797.66537MV
The voltage in ROUTPUT: 127.72812MV
The Current Reference: 797.66537UA
The Output Current: 1.20498MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.20179V
Current from DPS1A is: 17.89744MA
The voltage in RREF: 843.96124MV

The voltage in ROUTPUT: 134.6725MV
The Current Reference: 843.96124UA
The Output Current: 1.2705MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.2597V
Current from DPS1A is: 17.89744MA
The voltage in RREF: 897.13893MV
The voltage in ROUTPUT: 142.63352MV
The Current Reference: 897.13893UA
The Output Current: 1.3456MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.30474V
Current from DPS1A is: 17.89744MA
The voltage in RREF: 944.37324MV
The voltage in ROUTPUT: 149.71866MV
The Current Reference: 944.37324UA
The Output Current: 1.41244MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.34978V
Current from DPS1A is: 18.92015MA
The voltage in RREF: 990.3563MV
The voltage in ROUTPUT: 156.75689MV
The Current Reference: 990.3563UA
The Output Current: 1.47884MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.40125V
Current from DPS1A is: 17.89744MA
The voltage in RREF: 1.03665V
The voltage in ROUTPUT: 163.6387MV
The Current Reference: 1.03665MA
The Output Current: 1.54376MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.45272V
Current from DPS1A is: 18.92015MA
The voltage in RREF: 1.08514V
The voltage in ROUTPUT: 170.95846MV
The Current Reference: 1.08514MA
The Output Current: 1.61282MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.5042V
Current from DPS1A is: 18.92015MA
The voltage in RREF: 1.13738V
The voltage in ROUTPUT: 179.07588MV
The Current Reference: 1.13738MA
The Output Current: 1.6894MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.55567V
Current from DPS1A is: 18.92015MA
The voltage in RREF: 1.18367V
The voltage in ROUTPUT: 186.31743MV
The Current Reference: 1.18367MA
The Output Current: 1.75771MA

FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.60071V
Current from DPS1A is: 18.92015MA
The voltage in RREF: 1.22997V
The voltage in ROUTPUT: 193.30873MV
The Current Reference: 1.22997MA
The Output Current: 1.82367MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.65218V
Current from DPS1A is: 18.92015MA
The voltage in RREF: 1.27658V
The voltage in ROUTPUT: 200.62848MV
The Current Reference: 1.27658MA
The Output Current: 1.89272MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.70366V
Current from DPS1A is: 19.94286MA
The voltage in RREF: 1.3235V
The voltage in ROUTPUT: 207.77619MV
The Current Reference: 1.3235MA
The Output Current: 1.96015MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.75513V
Current from DPS1A is: 19.94286MA
The voltage in RREF: 1.37543V
The voltage in ROUTPUT: 215.67464MV
The Current Reference: 1.37543MA
The Output Current: 2.03467MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.8066V
Current from DPS1A is: 19.94286MA
The voltage in RREF: 1.42203V
The voltage in ROUTPUT: 222.96311MV
The Current Reference: 1.42203MA
The Output Current: 2.10343MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.85808V
Current from DPS1A is: 19.94286MA
The voltage in RREF: 1.46927V
The voltage in ROUTPUT: 230.48619MV
The Current Reference: 1.46927MA
The Output Current: 2.1744MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.90311V
Current from DPS1A is: 20.96557MA
The voltage in RREF: 1.51588V
The voltage in ROUTPUT: 237.74338MV
The Current Reference: 1.51588MA
The Output Current: 2.24286MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 2.94815V
Current from DPS1A is: 19.94286MA

The voltage in RREF: 1.56186V
The voltage in ROUTPUT: 244.59392MV
The Current Reference: 1.56186MA
The Output Current: 2.30749MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.00606V
Current from DPS1A is: 20.96557MA
The voltage in RREF: 1.6141V
The voltage in ROUTPUT: 252.8521MV
The Current Reference: 1.6141MA
The Output Current: 2.3854MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.0511V
Current from DPS1A is: 20.96557MA
The voltage in RREF: 1.66008V
The voltage in ROUTPUT: 259.92161MV
The Current Reference: 1.66008MA
The Output Current: 2.45209MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.10257V
Current from DPS1A is: 20.96557MA
The voltage in RREF: 1.70638V
The voltage in ROUTPUT: 266.86599MV
The Current Reference: 1.70638MA
The Output Current: 2.5176MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.14761V
Current from DPS1A is: 21.98828MA
The voltage in RREF: 1.75205V
The voltage in ROUTPUT: 273.71653MV
The Current Reference: 1.75205MA
The Output Current: 2.58223MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.19909V
Current from DPS1A is: 21.98828MA
The voltage in RREF: 1.79866V
The voltage in ROUTPUT: 280.87987MV
The Current Reference: 1.79866MA
The Output Current: 2.64981MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.25699V
Current from DPS1A is: 21.98828MA
The voltage in RREF: 1.85246V
The voltage in ROUTPUT: 288.74704MV
The Current Reference: 1.85246MA
The Output Current: 2.72403MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.30203V
Current from DPS1A is: 21.98828MA
The voltage in RREF: 1.90032V
The voltage in ROUTPUT: 295.66014MV
The Current Reference: 1.90032MA

The Output Current: 2.78925MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.35351V
Current from DPS1A is: 21.98828MA
The voltage in RREF: 1.9463V
The voltage in ROUTPUT: 302.54196MV
The Current Reference: 1.9463MA
The Output Current: 2.85417MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.39855V
Current from DPS1A is: 21.98828MA
The voltage in RREF: 1.99479V
The voltage in ROUTPUT: 309.17353MV
The Current Reference: 1.99479MA
The Output Current: 2.91673MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.45002V
Current from DPS1A is: 21.98828MA
The voltage in RREF: 2.04171V
The voltage in ROUTPUT: 316.00843MV
The Current Reference: 2.04171MA
The Output Current: 2.98121MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.50793V
Current from DPS1A is: 23.01099MA
The voltage in RREF: 2.09332V
The voltage in ROUTPUT: 323.76612MV
The Current Reference: 2.09332MA
The Output Current: 3.0544MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.55297V
Current from DPS1A is: 23.01099MA
The voltage in RREF: 2.13899V
The voltage in ROUTPUT: 330.11616MV
The Current Reference: 2.13899MA
The Output Current: 3.1143MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.60444V
Current from DPS1A is: 23.01099MA
The voltage in RREF: 2.18748V
The voltage in ROUTPUT: 336.66953MV
The Current Reference: 2.18748MA
The Output Current: 3.17613MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.64948V
Current from DPS1A is: 23.01099MA
The voltage in RREF: 2.2344V
The voltage in ROUTPUT: 343.55135MV
The Current Reference: 2.2344MA
The Output Current: 3.24105MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.70739V

Current from DPS1A is: 23.01099MA
The voltage in RREF: 2.29008V
The voltage in ROUTPUT: 351.26211MV
The Current Reference: 2.29008MA
The Output Current: 3.31379MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.75242V
Current from DPS1A is: 23.01099MA
The voltage in RREF: 2.33575V
The voltage in ROUTPUT: 357.08038MV
The Current Reference: 2.33575MA
The Output Current: 3.36868MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.8039V
Current from DPS1A is: 24.0337MA
The voltage in RREF: 2.38392V
The voltage in ROUTPUT: 363.79015MV
The Current Reference: 2.38392MA
The Output Current: 3.43198MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.84894V
Current from DPS1A is: 24.0337MA
The voltage in RREF: 2.43085V
The voltage in ROUTPUT: 370.40608MV
The Current Reference: 2.43085MA
The Output Current: 3.4944MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.90041V
Current from DPS1A is: 24.0337MA
The voltage in RREF: 2.47745V
The voltage in ROUTPUT: 376.56844MV
The Current Reference: 2.47745MA
The Output Current: 3.55253MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 3.95188V
Current from DPS1A is: 24.0337MA
The voltage in RREF: 2.53032V
The voltage in ROUTPUT: 383.4659MV
The Current Reference: 2.53032MA
The Output Current: 3.6176MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.00336V
Current from DPS1A is: 24.0337MA
The voltage in RREF: 2.57912V
The voltage in ROUTPUT: 390.23823MV
The Current Reference: 2.57912MA
The Output Current: 3.68149MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.0484V
Current from DPS1A is: 25.05641MA
The voltage in RREF: 2.62635V
The voltage in ROUTPUT: 396.55699MV

The Current Reference: 2.62635MA
The Output Current: 3.7411MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.09987V
Current from DPS1A is: 25.05641MA
The voltage in RREF: 2.67453V
The voltage in ROUTPUT: 402.4691MV
The Current Reference: 2.67453MA
The Output Current: 3.79688MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.15134V
Current from DPS1A is: 25.05641MA
The voltage in RREF: 2.72082V
The voltage in ROUTPUT: 408.95991MV
The Current Reference: 2.72082MA
The Output Current: 3.85811MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.20282V
Current from DPS1A is: 25.05641MA
The voltage in RREF: 2.77556V
The voltage in ROUTPUT: 416.15454MV
The Current Reference: 2.77556MA
The Output Current: 3.92599MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.25429V
Current from DPS1A is: 26.07912MA
The voltage in RREF: 2.82248V
The voltage in ROUTPUT: 421.9728MV
The Current Reference: 2.82248MA
The Output Current: 3.98088MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.30576V
Current from DPS1A is: 26.07912MA
The voltage in RREF: 2.86941V
The voltage in ROUTPUT: 428.24464MV
The Current Reference: 2.86941MA
The Output Current: 4.04004MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.3508V
Current from DPS1A is: 25.05641MA
The voltage in RREF: 2.91601V
The voltage in ROUTPUT: 434.4852MV
The Current Reference: 2.91601MA
The Output Current: 4.09892MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.40228V
Current from DPS1A is: 26.07912MA
The voltage in RREF: 2.96544V
The voltage in ROUTPUT: 440.60063MV
The Current Reference: 2.96544MA
The Output Current: 4.15661MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS

Voltage of DPS1A is: 4.45375V
Current from DPS1A is: 26.07912MA
The voltage in RREF: 3.0183V
The voltage in ROUTPUT: 447.09144MV
The Current Reference: 3.0183MA
The Output Current: 4.21784MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.50522V
Current from DPS1A is: 26.07912MA
The voltage in RREF: 3.06491V
The voltage in ROUTPUT: 453.26944MV
The Current Reference: 3.06491MA
The Output Current: 4.27613MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.55026V
Current from DPS1A is: 27.10183MA
The voltage in RREF: 3.11215V
The voltage in ROUTPUT: 459.57256MV
The Current Reference: 3.11215MA
The Output Current: 4.33559MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.60173V
Current from DPS1A is: 26.07912MA
The voltage in RREF: 3.16032V
The voltage in ROUTPUT: 465.25006MV
The Current Reference: 3.16032MA
The Output Current: 4.38915MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.64677V
Current from DPS1A is: 27.10183MA
The voltage in RREF: 3.20693V
The voltage in ROUTPUT: 471.50626MV
The Current Reference: 3.20693MA
The Output Current: 4.44817MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.70468V
Current from DPS1A is: 27.10183MA
The voltage in RREF: 3.26042V
The voltage in ROUTPUT: 478.26295MV
The Current Reference: 3.26042MA
The Output Current: 4.51191MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.74972V
Current from DPS1A is: 27.10183MA
The voltage in RREF: 3.30828V
The voltage in ROUTPUT: 483.76841MV
The Current Reference: 3.30828MA
The Output Current: 4.56385MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.80119V
Current from DPS1A is: 27.10183MA
The voltage in RREF: 3.35708V

The voltage in ROUTPUT: 489.80564MV
The Current Reference: 3.35708MA
The Output Current: 4.62081MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.85267V
Current from DPS1A is: 27.10183MA
The voltage in RREF: 3.40525V
The voltage in ROUTPUT: 496.09312MV
The Current Reference: 3.40525MA
The Output Current: 4.68012MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.89771V
Current from DPS1A is: 28.12454MA
The voltage in RREF: 3.45217V
The voltage in ROUTPUT: 501.75498MV
The Current Reference: 3.45217MA
The Output Current: 4.73354MA
FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS
Voltage of DPS1A is: 4.95561V
Current from DPS1A is: 28.12454MA
The voltage in RREF: 3.5066V
The voltage in ROUTPUT: 508.43347MV
The Current Reference: 3.5066MA
The Output Current: 4.79654MA

FUNCTIONAL_1B OUTPUT DATA FILE FOR MANUFACTURING TEST OF BASIC BIPOLAR CURRENT-MIRROR

FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS

Serial number of the unit is	1	2	3	4	5	6
Voltage of DPS1A is	3.00606V	3.00606V	3.00606V	3.00606V	2.99963V	3.00606V
Current from DPS1A is	21.98828MA	21.98828MA	21.98828MA	21.98828MA	21.98828MA	21.98828MA
The voltage in RREF	2.34983V	2.31198V	2.31198V	2.31511V	2.31323V	2.31323V
The voltage in ROUTPUT	336.65389MV	336.23159MV	338.62459MV	331.68021MV	339.87583MV	335.95006MV
The Current Reference	2.34983MA	2.31198MA	2.31198MA	2.31511MA	2.31323MA	2.31323MA
The Output Current	3.17598MA	3.172MA	3.19457MA	3.12906MA	3.20638MA	3.16934MA

Appendix C

VLCT Basic Programming Manual

VLCT means "very low cost tester", because it is custom-made equipment [2, 3, 45]. The VLCT consists of three major components: The workstation, the system power supplies, and the test head. The concept could be applied to any custom-made testing equipment.

Figure B.1 VLCT-256 Equipment from TI

The workstation is a SUN Ultra SPARC installed in a mainframe (except for the terminal) with the system power supplies. The test head consists of one to four quadrants with up to 256 channels per quadrant. The programming language for the available VLCT is PASCAL; there is not clear if there is support for other programming languages.

VLCT Programming Environment

The VLCT controller runs the Solaris UNIX Operating System. The graphical user interface (GUI), “OpenWindows” creates windows to perform various tasks like file management, sending/receiving mail, shells for running other applications and many others.

Basically, OpenWindows takes individual UNIX commands and uses X-Windows graphics to present them in a user-friendlier environment. The tester User Interface (TUI) is an application that is run from an Open-Windows shell. The TUI software was written and is supported by TTC (test training center) to allow communication with the VLCT Tester.

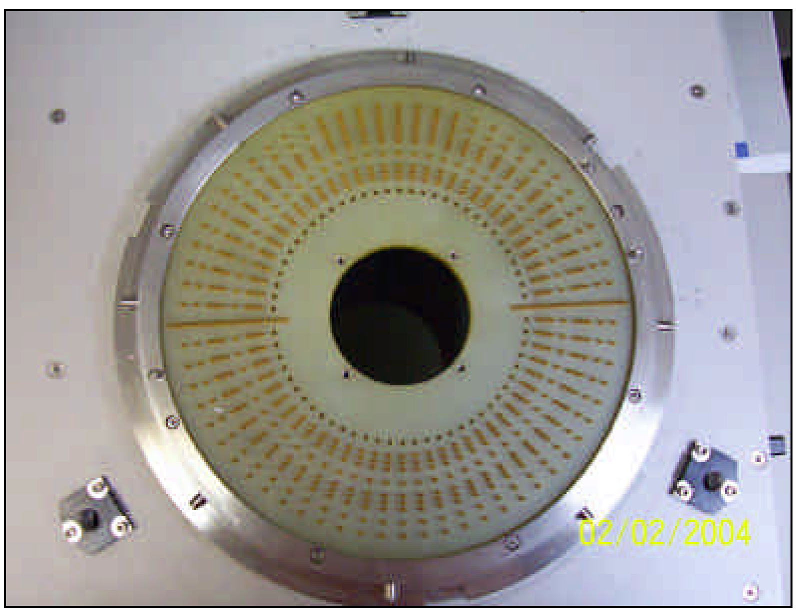


Figure B.2: Test Head 256 Pin

The Performance Board

The VLCT performance board is laid out as the V-series 256-pin performance board. A translator block is available for Polaris boards. This translator is the circular gold one board. Newer boards have been developed to allow more room for performance board circuitry – one uses the same pogo pin block and another one allows even more extra room space with a different pogo block

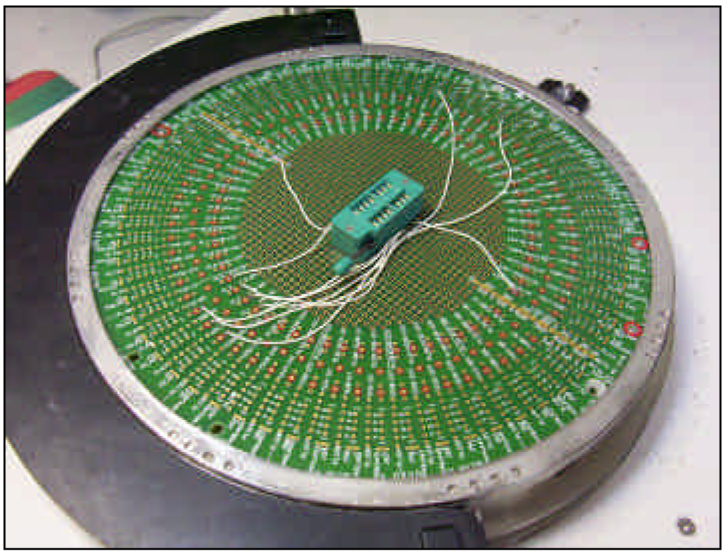


Figure B.3: Performance Board

VLCT general test coverage

The goal of testing with very low cost equipment requires re-defining device test flows to use measurements and methods that can be implemented. Traditional tests may need to be modified, narrowly defined or eliminated. For the VLCT, the general test coverage is:

a) Continuity. Per-pin with a loose PMU measurement.
b) I/O Leakage. - Per-pin with a low current PMU.
c) ICC. Measured with a power supply.
d) ICCQ. Measured with a power supply.
e) Slow Functional. Functional patterns run by the system controller.
f) Fast Functional. Only using device BIST and shared high-speed clocks.
g) DC Parametric. Using shared-resource PMU.
h) AC Parametric. Using dedicated, shared hardware.

Device Power Supplies

The VLCT-256 has two power supplies, DPS1A and DPS1B are used to provide the DC voltage. Many devices requires specific power up and power down sequences (or shutdown sequence) to prevent damage to the device and ensure a proper operation. A general power up sequence would be:

- DUT VCC pin(s) to power supply & wait for stabilization.
- DUT Input pin(s) to driver pins.
- DUT Output pin(s) to receiver load.

The power down sequence will occur in reverse order. The VLCT software does not directly define a power up/down sequence. Procedures must be defined in the test program to set any desired sequences and wait times.

In the VLCT, there is no such thing as a disable power supply or device pin. Device pins are either connected or disconnected, and power supplies must be programmed for 0 Volts and a small current to power down.

Continuity Testing

To prevent wasting time on devices that have no chance of passing the full test, continuity testing verify that the DUT has working ESD diodes and electrical connection from the tester to the DUT pins. Continuity test for two things:

- Parameters of the Electrostatic Discharge (ESD) Diodes
- Path connections from the Test set-up to the DUT pins

The ESD diodes are internal to the device and are used to protect DUT from circuitry from electrostatic discharge voltage spikes and interface transients. There are two types of ESD diodes, VCC and Ground.

The VCC diodes are reverse bias connected from the DUT pin to a DUT power supply, and Ground diodes are reverse bias connected from DUT pin to ground. A typical ESD diode will have a forward voltage of approximately +0.6 volts at 1mA

General Continuity Test Description:

1. Gnd Diode: Apply Gnd to the DUT Gnd-pin.
2. Vcc Diode: Apply 0V to the DUT Vcc-Pin.
3. Vcc Diode: Force 1mA and measure voltage min= 0.4V, max = 0.8 V.
4. Gnd Diode: Force – 1mA and measure voltage min = - 0.4 V, max = -0.8V

Pin Shorts

These VLCT continuity test methods only use single magnitude limit for testing and can check for opens but not for shorts at the same time. Pin shorts testing are usually done with the pin PMU and just contains of forcing a small voltage and measuring for excessive current.

Leakage Test or Input Current

Input current test are device specification tests and verify the DUT inputs can be driven to a high and a low, while consuming no more than a specified amount of current.

The VLCT Software

You will need a proper account in ICDL, contact the administrator for it. At the moment that you own an account, use it to get access to the test software by Secure Shell (SSH). In SSH you must type your password to connect with the server ICDL.UPRM.EDU after that you must type "VLCT" in the SSH Window, then the following screen will appear in the screen:

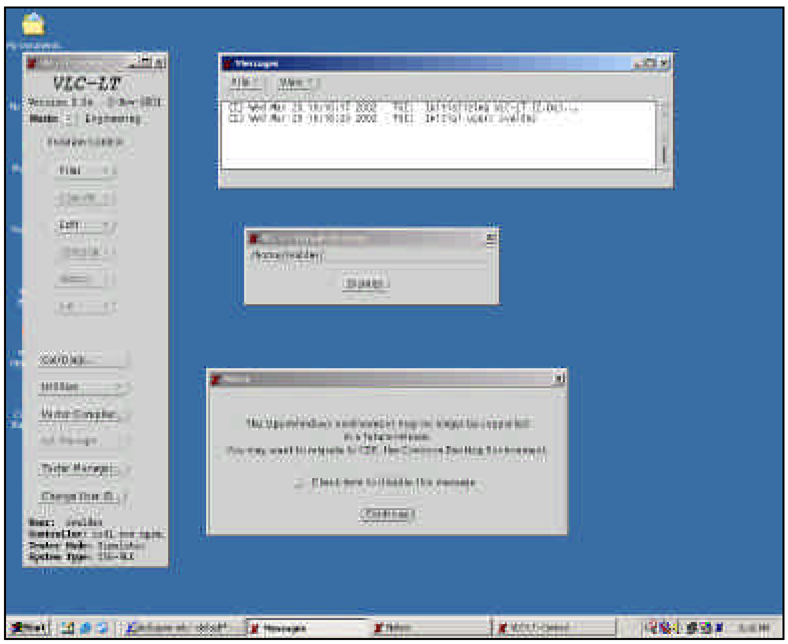


Figure B.3: VLCT Interface

You will see 4 windows: the VLC-LT window, the message window, the default directory window and sometimes the Notice window.

To load an existing file, on the VLC-LT window, select the command "*file*", the VLC-LT load files window will appear:

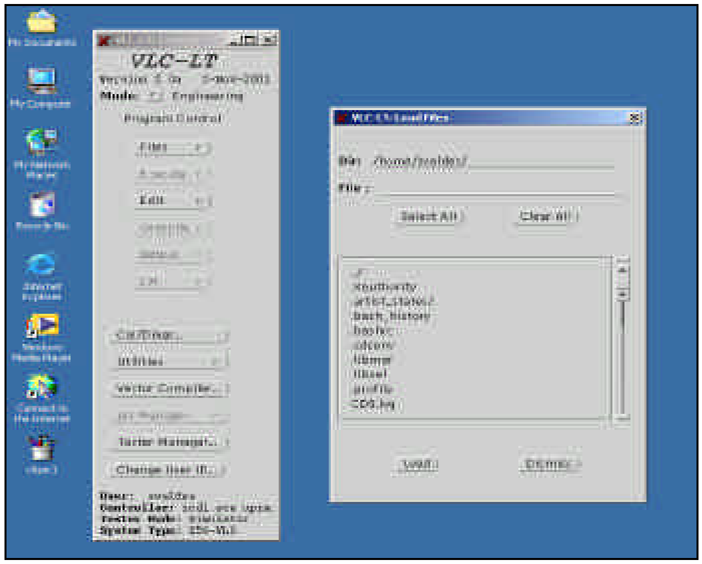


Figure B.4: VLC-LT load file Window

To edit a new file, select "edit" on the VLC-LT window, the edition window will appear:

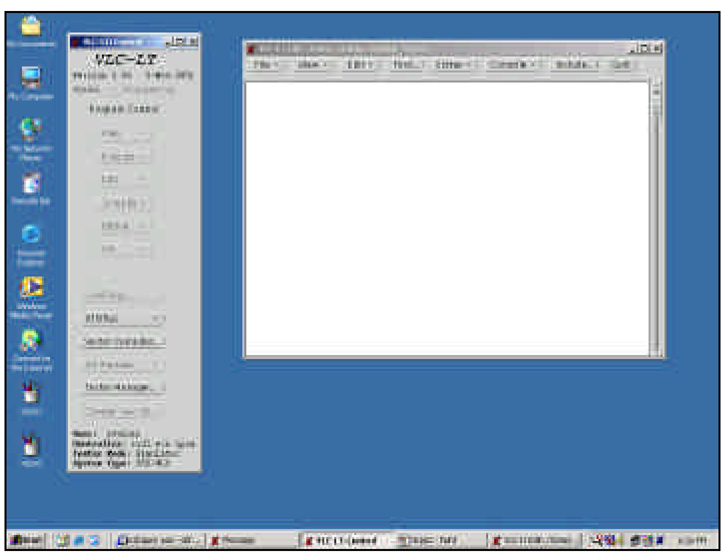


Figure B.5: Edition Window

In this window we create and edit all the files that we use to build our programs. The procedure to use, edit, create and compile them will be exposed later.

With a right mouse button click in the “file” command of the edit window you will find a small menu with the file options, the “save to disk” option saves the file in the hard drive and “save to memory” saves the file in RAM.

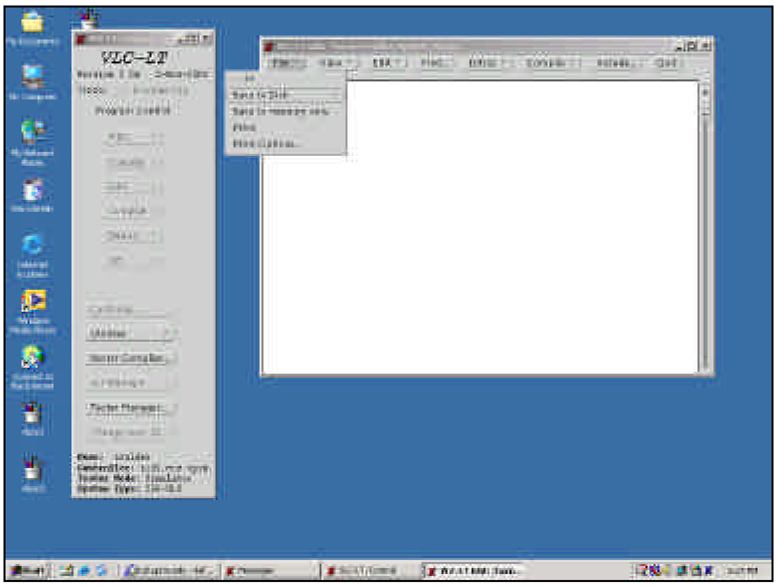


Figure B.6: Saving File options

Use the “save to disk” option in order to save your files in the hard-drive. Always use the same directory. Check with your network administrator the properties of a network printer.

At the VLC-LT main menu window, the "mode" option is used to change the test mode of the tester, after press it, the following screen will appear:

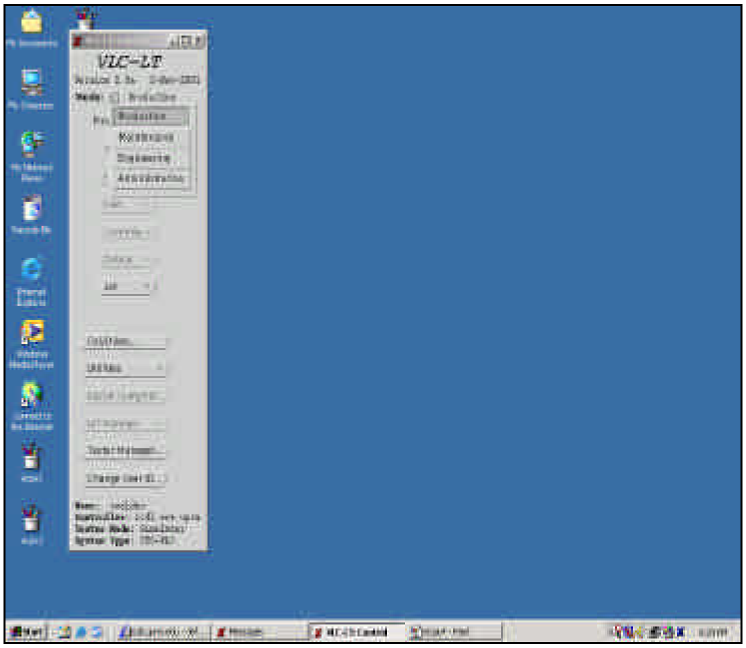


Figure B.7: Modes

The "operation mode" has 4 options: production, maintenance, engineering and administration. In production and administration modes the user is allowed to perform any operation. The maintenance mode is use to perform a self check and engineering mode is used for testing simulation purposes.

PASCAL Test Programming

Next is an example of Pascal program:

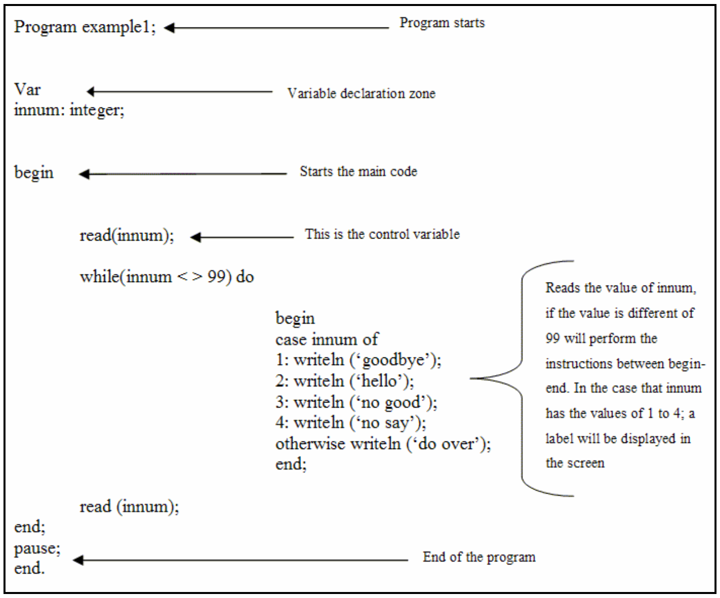


Figure B.8: Example of PASCAL Code

This example, you will find that:

1) The program starts with a declaration.
2) The variable declaration zone is defined with "var"
3) The main code starts with a "begin" and ends with an "end."
4) The program waits for an input from the user (this is the pause).
5) The program ends.

The following example is a Pascal Test program, where the difference with the previus example, are the instruction related to the test hardware:

```
Module clockfunc_p;
#include '$TYPES';
#include 'var.h';
#include 'utilities.h'

Function CLOCK_FUNC : Boolean; Export;
Var
Result:Boolean;
Freq:Treal;

Begin
        TestOpen( ClockTest );
        SupplySet( VDDC, v[VDDnom], 5ma, S_IGAIN_1X, S_High_Curr_OverRide);
        SetupSelect( ClockDC ); {forces input high, connects clock and counter }
        Discard( VoltageCompareOpt( S_DRVON ) ); {turns the drivers on}
        Wait( 5ms );
        ClockSet( S_CLOCK1A, False, 10MgHz, 2.8v, 0.3v );
        ClockStartFreeRun( S_CLOCK1A );
        ClockSetFreqCounter( S_FREQCNT1A );
        ClockStopFreeRun( S_CLOCK1A );
        WriteLn('Test read a frequency of', freq);
        If (Freq>4.5MgHz) and (Freq<5.5MgHz) Then
                                                        Result := TRUE
                                                Else
                                                        Result := FALSE;
        SetupSelect(PowerDownDC);
        SupplySet( VDDC, 0v, 5ma, S_IGAIN_1X, S_High_Curr_OverRide );
        ResultRecord( Result );
        CLOCK_FUNC := Result;
TestClose;
End;
```

Figure B.9: Pascal Test Code

Programming in VLCT Pascal

The most common instructions are:

Begin/End	Repeat-Until	Import
Pause	For-To	Export
Writeln/Write	For-Downto	Use
String	Case	Module
Read	Procedure	#Include
If-Then	Function	#Define
If-Then-Else	Type	
While-Do	Var	

All values to be stored in a same variable must be of the same type. A type defines what kind of a group the variable is (as we go on you'll see there are character types (text), real types (decimal numbers), and Boolean types (true and false). A variable is declared by:

Variable_name:type;

in the original programming language we have the following variables types:

a) Real.- number with decimals that could be positive or negative
b) Integers.- number without decimals
c) Char.- character or chain of characters stored in a variable
d) Boolean.- true and false logic values

But in the VLCT PASCAL Programming Language we have an extended set of variable types. We will review them later.

Logic Operations

The logic operations are:

a) Not
b) And.
c) Or.
d) Nand.
e) Not.
f) Nor.

Mathematical Operators

The mathematical operators are: (+, -, =, /, DIV, MOD). Like in other languages you are only allowed to perform any mathematical operations with the same variable type, you cannot perform a division with mixed variables types.

2.15.4 Comparison Operators

The Comparison operators are: (<, >, =, <>, >=, <=) and are used in if-then statement and others.

<	Less than
>	Greater than
=	equal to
<>	not equal to
>=	equal to or greater than
<=	equal to or less than.

Array

Array is a space memory defined with X by Y dimensions. To work with arrays follow the next steps:

a) Define the array.
b) Declare the variable.
c) Fill the array.

For example the array 1 x 6.

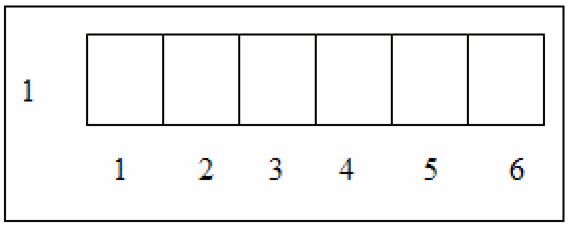


Figure 2.10 Array 1 by 6

You must define this array in the next way:

```
TYPE
        Array_name = ARRAY[1..6] of type;
Var
        Array_variable : array_name
```

For example the following program, defines an integer array in a size of 1 by 6, then, the program defines two variables: integer and array_example. Later, starts the code with the instruction "begin" and starts the first of two cycles, the first cycle will ask for a value to be stored in each matrix cell, the second cycle will display in the screen the value of each matrix cell. Comments are always between (* *).

```
Program Example_array;                       (* program declaration* )

TYPE                    (*type section where the user describes the set of possible values and gives the set a
name or a type identifier*)
array_example = ARRAY[4..9] of integer;      (* here the user declares an array from 4 to 9
integer *)

VAR                                          (*VAR keyword indicates the start of variable
declaration section*)
i:integer;                                   (* declares variable "i" like integer*)
store:array_example;                         (* declare "store" like an array*)

begin                                        (*keyword used to start the main code *)
        FOR i:= 4 TO 9 DO                    (* starts a loop it will count from 4 to 9 at "i" and
                                             will execute all below*)

        Begin                                (*starts the loop's code*)

        writeln('input an integer to fill position', i:2, 'in the array');
                                             (*writes text in the screen and the value of "i"*)

        read(store[i]);                      (*waits for the input from the user to be stored in
                                             "store"*)
        end;                                 (*ends the loop*)

        FOR i:= 4 TO 9 DO                    (* starts a loop it will count from 4 to 9 at "i" and
                                             will execute all below*)
        writeln('position # ',i:1, '=', store[i]:1);   (*writes text and values of "i" and "store"*)
        writeln('');                         (* write a blank line*)
Pause;                                       (*pauses program execution *)
End.
```

Single quote twice (')... NOT double quote (")

Figure B.11 Example Array in Pascal

To run the program, select the option "edit" and then "new file".

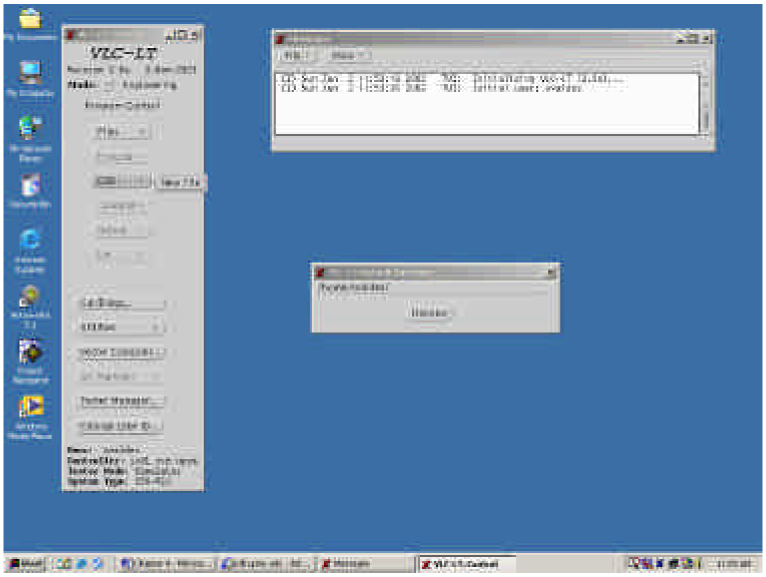


Figure B.12 New File Command

Then start to write the code in the blank or white window, note the windows that show info of the current directory.

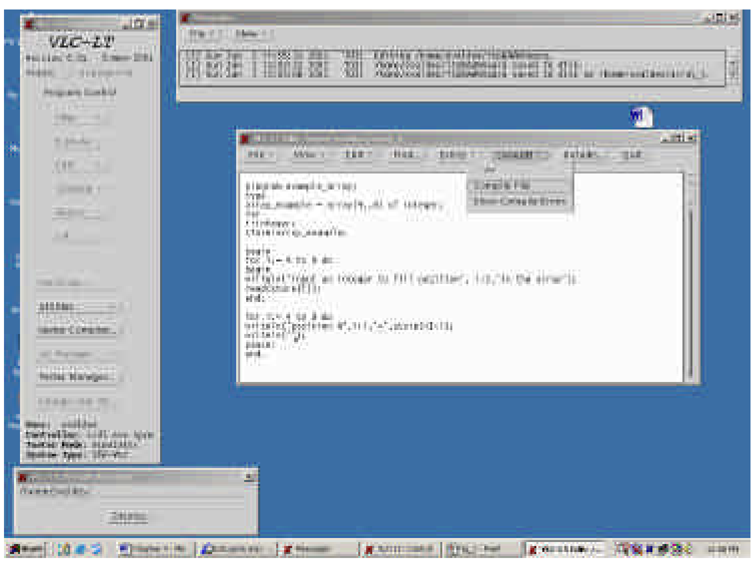


Figure B.13 Programming Array in Pascal

When all code is correctly written, you must compile the code, like the picture shows.

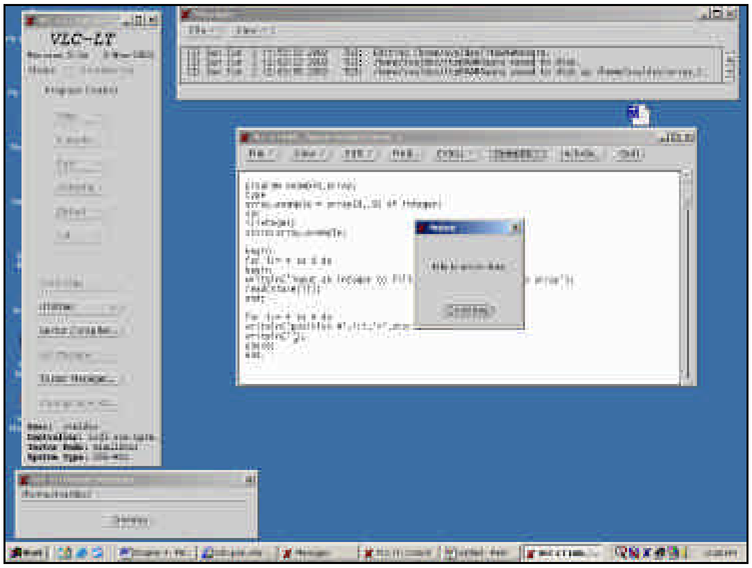


Figure B.14 Compile Code

If the compilation is successful, the VLCT will show the “file is error free” window. Then you must confirm the window and then close the window to start the run command.

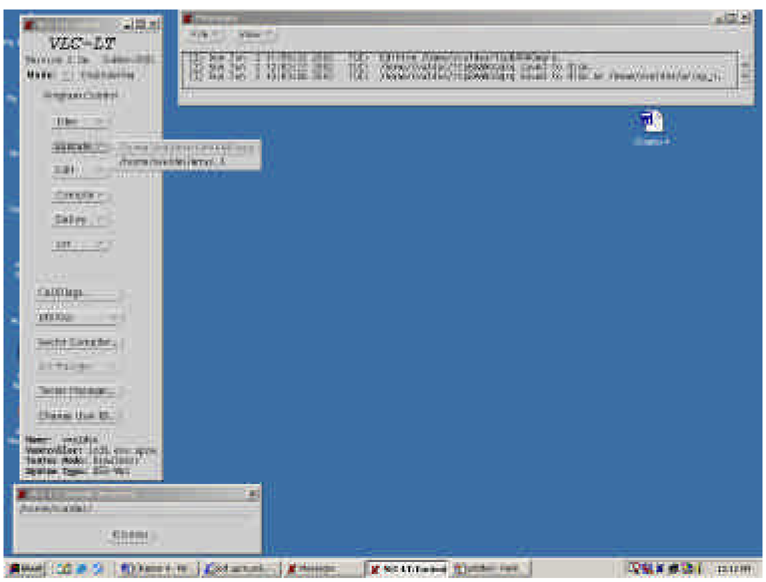


Figure B.15 Execute command

The run command here is located in the execute option, you will select it and the VLCT will show you a list of the program recently edited and ready to execute, you only need to select it. Finally, this is the real output of the program:

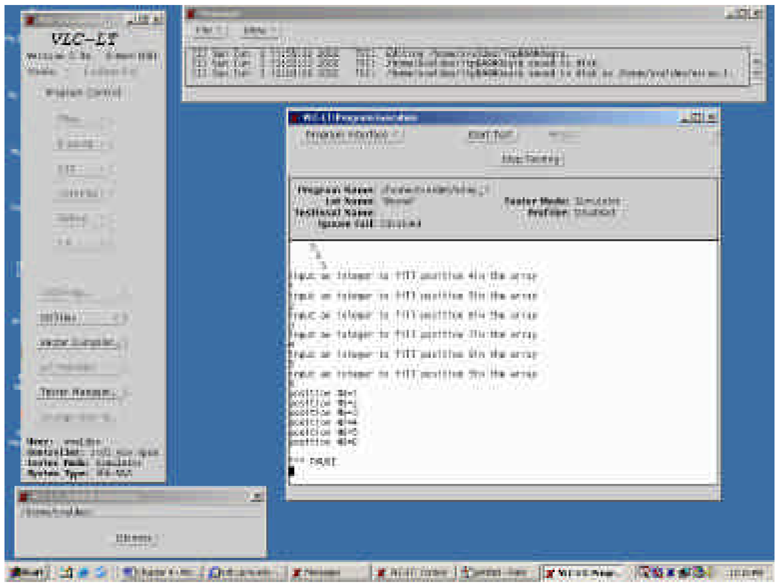


Figure B.16 Output of the Array Code

String Array

This is used to store text characters. You must follow two rules: Must be a packed array and lower limit must start with 1.

```
Program example_string_array;                       (*program declaration*)
        TYPE                                        (*type section where the user describes the set of
                                                    possible values and gives the set a name or a type
                                                    identifier*)
array_example = PACKED ARRAY[1..5] of char;         (*here the user declares an array from 1 to 5 char*)

VAR                                                 (*VAR keyword indicates the start of variable
declaration section*)
i:integer;                                          (* declares variable "i" like integer*)
store: array_example;                               (* declare "store" like an array*)

begin                                               (*keyword used to start the main code *)
    For i:= 1 TO 5 DO                               (* starts a loop it will count from 4 to 9 at "i" and
will execute all below*)
    Begin                                           (*starts the loop's code*)
        writeln('Input an alpha character');        (*write the text in the screen*)
        readln(store[i]);                           (*waits for the input from the user to be stored in
                                                    "store" with index = i *)
    end;                                            (*ends the loop*)

    FOR i:= 1 TO 5 DO                               (* starts a loop it will count from 4 to 9 at "i" and
                                                    will execute all below*)
    writeln('position # ', i:1, '=', store[i]);     (*writes text and values of "i" and "store"*)

pause;
                                                    (*pauses program execution *)
end
```

Figure B.17 String Array in Pascal

The Program output is:

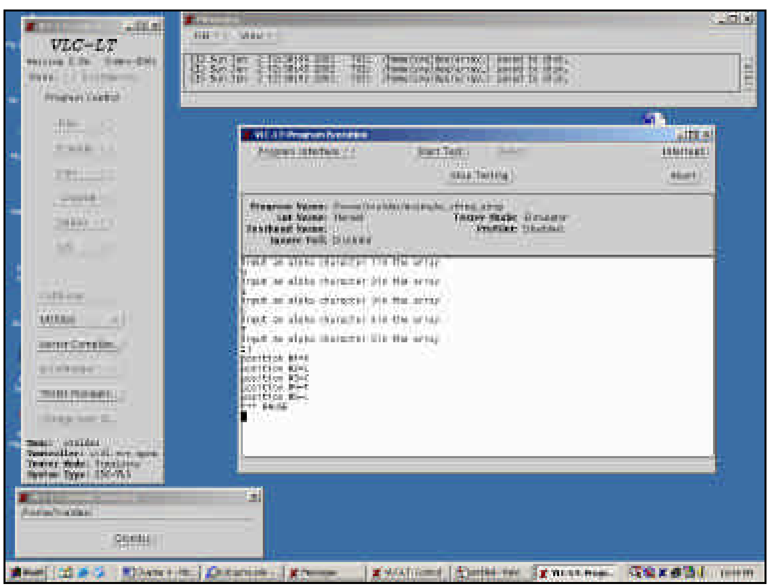


Figure B.18 String Array

The next program defines an array of 1 by 40 to store enough characters to be displayed later in the program.

```
Program name_pascal;

VAR
name:string[40]

begin
        writeln('please type your complete name');
        readln(name);
        writeln('');
        writeln('your name is', name);
pause;
end.
```

Program Output:

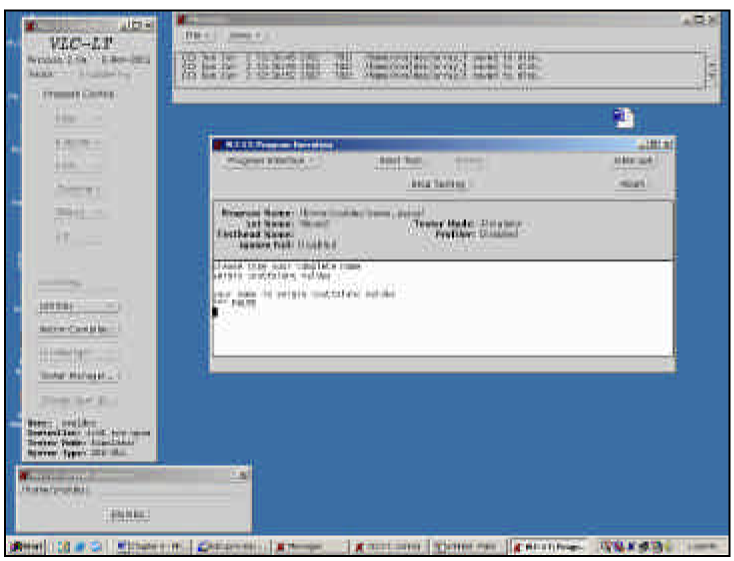


Figure B.19 Output Program

Two-Dimensional Arrays

An “n by m” matrix is defined in this example. The example shows a 2 by 6 matrix using two cycles ‘FOR”:

1	1,1	1,2	1,3	1,4	1,5	1,6
2	2,1	2,2	2,3	2,4	2,5	2,6
	1	2	3	4	5	6

Figure B.20 Array 2 by 6

Example Code:

```
begin
        FOR I:= 1 TO 6 DO                          (* this loop will fill the positions 1,1 to 1,6 *)
                begin
                writeln(‘input an integer to fill position 1’,’,’,i:1,’ in the array’);
                read(store[1,i]);                  (*read the imput form the user and store it in the
                                                   array *)
                end;

        FOR I:=1 TO 6 DO                           (* this loop will fill the positions 2,1 to 2,6 *)
                begin
                writeln(‘input an integer to fill position 2’,’,’,i:1,’ in the array’);
                read(store[2,i]);                  (*read the imput form the user and store it in the
                                                   array *)
                end;

        FOR i:= TO 6 DO                            (*this loop will display all values of all the array
                                                   elements*)
                begin
                writeln(‘position # ‘,’,’1,’,i:=1,’=’, store[1,i]:1);
                writeln(‘position # ‘,’,’2,’,i:=1,’=’, store[1,i]:1);
                writeln(“);
                end;

pause;
end.
```

Figure B.21 Program for Array 6 by 2

Then Program Output is:

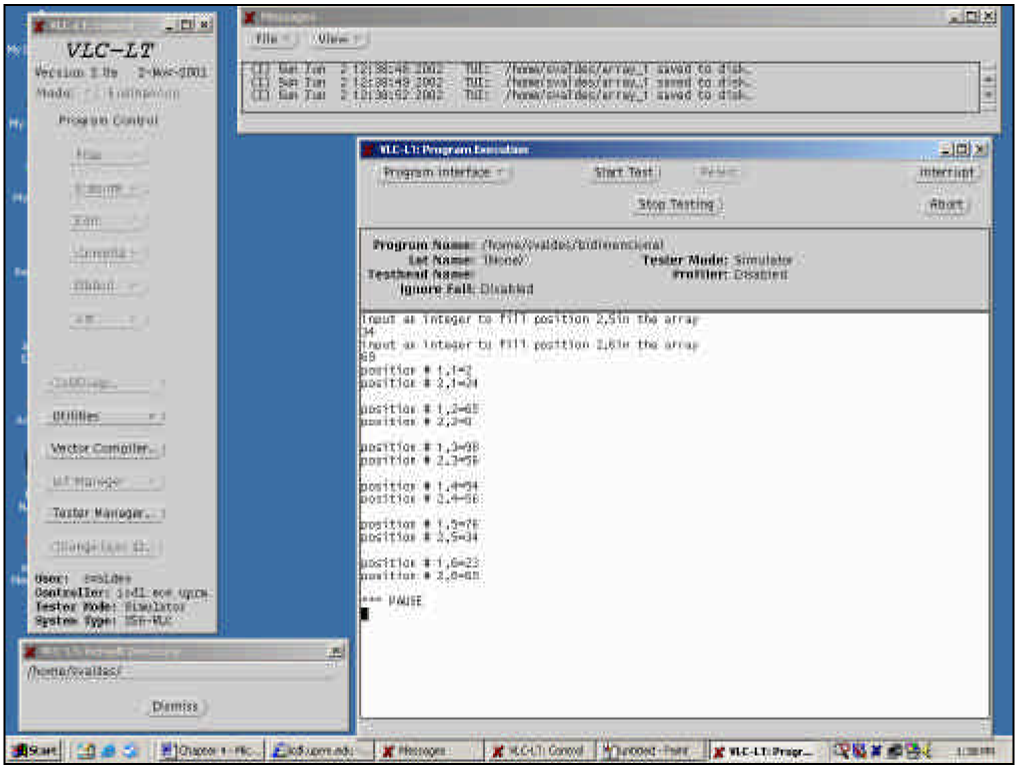


Figure B.22 Output for Array 6 by 2 Code

Procedures

A procedure is a subroutine, that it is defined before the main code (before the "begin" of the main program). See the following illustration:

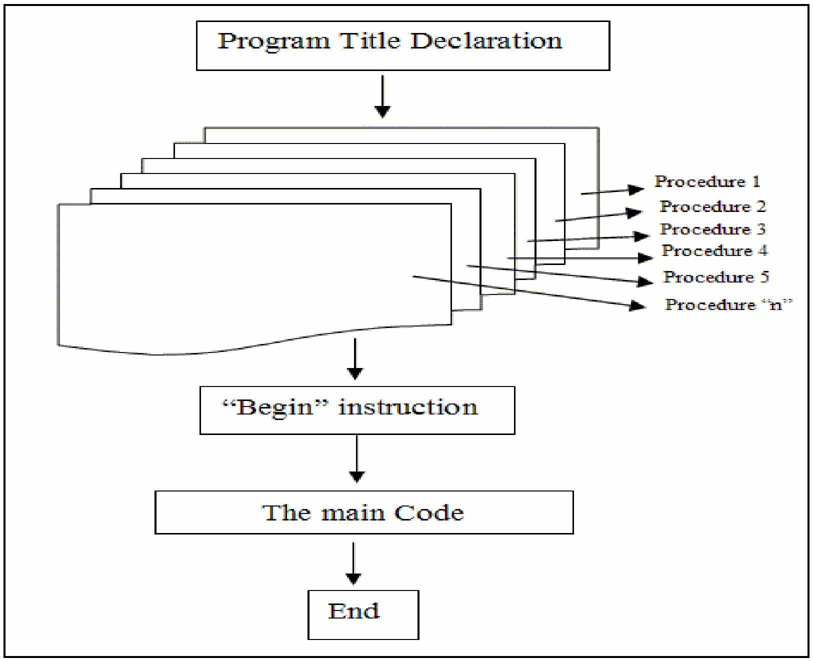


Figure B.23 Procedures

Each procedure has is particular code and variables, these variables If they are defined inside the procedures, they are called "local" variables because only the code inside these procedures will reach these variables. The global variables are defined before the procedures, all code will reach them.

Any number of procedures could be defined, according with the suggested rules:

- Do not use the same names to define local variables and global variables.

- Do not use mathematical operations in the main code mixing local and global variables.

- Be specific with the procedures, each procedure belongs to one operation, this make clear your code to understand.

This is the way to use the procedures:

- Define which will be local and global variables.
- Write each procedure with a specific task.
- Define if the procedure will be receiving or sending data to the main code.
- Study if the mathematical operation inside the procedure will reach all the proper variables.
- End the procedure.
- Starts the main code
- Study If the main code is sending or receiving data from/to the procedures.
- Make a “call” to the procedure.
- End the main code.

Next there is an example code that shows how to use the procedures:

```
Program Example procedures;
Var
i: integer;

        procedure demo1;
        VAR
        i: integer;
        begin
        i :=3;
        writeln('procedure demo1 = ', i:2);
        end;

        procedure demo2;
        VAR
        i: integer;
        begin
        i := 100;
        writeln('procedure demo2 = ', i:2);
        end;

begin
        i:= 10;
        demo1;
        demo2;
        writeln('main line = ', i:2);
        pause;
end.
```

Program title declaration and Global variables definition

Procedure named "demo1", note that It has his local variable definitions and his own code.

Procedure named "demo2", note that It has his local variable definitions and his own code.

The main code

Figure B.23 Procedure Example Code

The main code makes a "call" to the procedure, this is only writing the name of the procedure in the code, and it will be executed. Control of variables is the path to control the flow of data between the main code and the procedure.

For example, you can define the same global and local variables, then in the main code, store values in those variables and call a procedure to perform a mathematical operation with these values or vice versa. Many programmers use only local variables, like in the following example. You can easily add another procedure to the program without conflicts of global variables.

The next is a code using only local variables, due the length of the code, it will provided in text format:

```
Program Localonly;
Procedure chn_menu;
VAR
i, var1 : integer;
Begin
REPEAT                    (*loop condition tested at the end of the loop in the until statement*)
EraseScreen;                      (*will erase the screen*)
For i:= 1 to 10 DO
Writeln;
Writeln;(‘ ******************************’);
Writeln;(‘ CHANNEL SELECT MENU ‘);
Writeln;(‘ 0. SELECT ALL CHANNEL’);
Writeln;(‘ 1. SELECT CHANNEL A’);
Writeln;(‘ 2. SELECT CHANNEL B’);
Writeln;(‘ 3. SELECT CHANNEL C’);
Writeln;(‘ 4. SELECT CHANNEL D’);
Writeln;(‘ 5. SELECT CHANNEL E’);
Writeln;(‘ 6. SELECT CHANNEL F’);
Writeln;(‘ 7. SELECT CHANNEL G’);
Writeln;(‘ 8. SELECT CHANNEL H’);
Writeln;(‘ 9. SELECT CHANNEL A-’D);
Writeln;(‘10. SELECT CHANNEL E-H’);
Writeln;
Writeln(‘ SELECT YOUR OPTION……..’);
Readln (var1);                          (*wait for the user input to store it in “var1” *)
Until ((var1>=0) and (var1 < 11 ));     (* condition test if true go to case if false go to repeat *)

Case var1 OF              (*start of the case sentences. Case will compare the var1 with the values*)
0: begin                                (*and it will execute the code *)
        writeln(‘you selected option 0’);
        writeln(‘in the diagnostic program, a procedure could be ‘);
        writeln(‘called here to run test on all pins ‘);
   end;
9: begin
        writeln(‘you selected option 9’);
        writeln(‘in the diagnostic program, a procedure could be ‘);
        writeln(‘called here to run test on CHANNEL A-D ‘);
   end;
10: begin
        writeln(‘you selected option 10’);
        writeln(‘in the diagnostic program, a procedure could be ‘);
        writeln(‘called here to run test on CHANNEL E-H ‘);
    end;
1..8: begin
        writeln(‘you selected option ’, var1:2);
        writeln(‘in the diagnostic program, a procedure could be ‘);
        writeln(‘called here to run test on the selected channel ‘);
     end;
end;

begin                                   (*main program begin*)
CHN_MENU;                               (*calling procedure*)
Pause;
End.                                    (*main program ends*)
```

The output of the program will be:

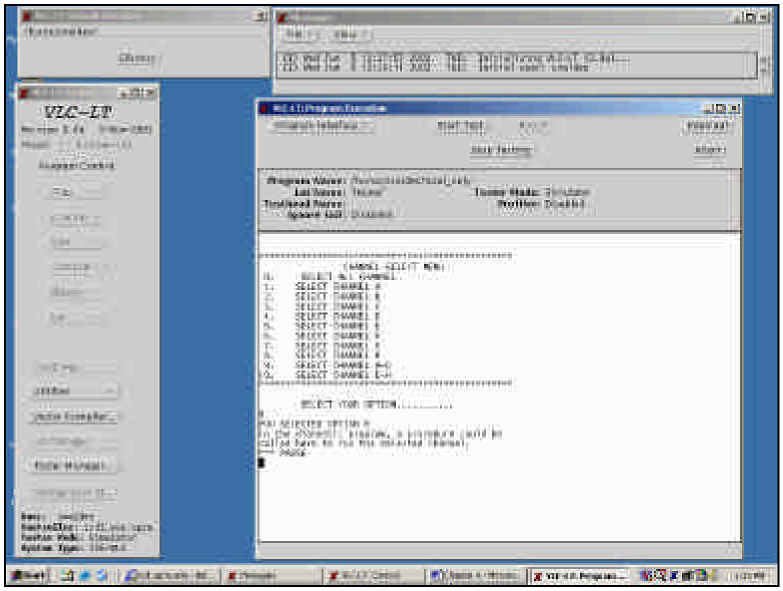


Figure B.24 Outputs for Procedure Program

Functions

A function is a subroutine, but the function always needs input and will always output. They must be declared before the main program (before the "begin" instruction). The main program will execute statements, invoke a function, run the function code and then return and run the next statement in the main program after the function is invoked. The syntax of a function is:

function function_name(VAR var1:type; VAR var2:type): function_type

- "var1 and var2" are the variables that are inputs for the function, note the **";"** between different types of variables.
- "Function_type", is the type of value that will be returned to main program via the "function_name".

Next an example of a function:

```
Program example_function;
VAR
Num: real;

Function kiss_my_foot(VAR var1:real):real
Begin
Kiss_my_foot := (var1*var1);
End;

Begin
Writeln('this program will calculate the square of a number');
Writeln('please input a real number = ');
Read(num);
Writeln('The square of', num:2, '=', kiss_my_foot(num));
Pause;
End
```

Figure B.25 Function Example Code

Program Output:

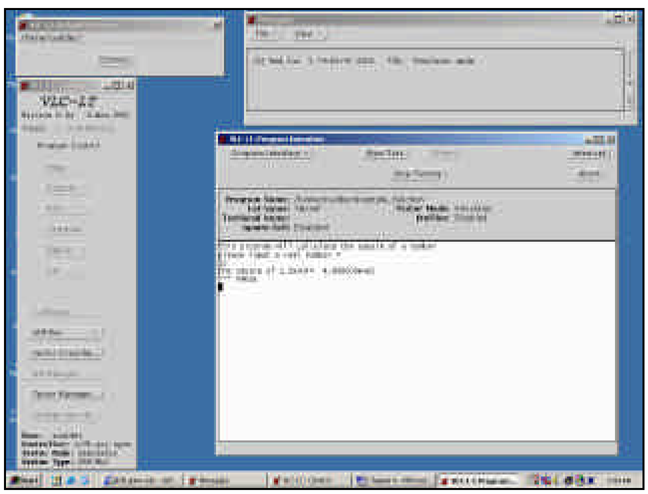


Figure B.26 Outputs for Function Example Code

First the program declare the global variables, then the function with is own variables and code and finally, the main code, the which one invoke the function passing "num" as a parameter from the main program via "var1", but the function will return the parameter "kiss_my_foot", instead of returning "var1" and this is the difference between functions and procedures.

Use of IMPORT, EXPORT and #INCLUDE

The procedures and functions are codes that actually are inside the main program, but with the use of IMPORT, EXPORT and #INCLUDE we will use code stored in an external file instead of use in the same file where the main program resides. Each of these, will only “load and compile”, because they will be executed after be called by main program.

Next there is an example of the use of IMPORT/EXPORT/USE instructions. First we build the auxiliary code that will be called by the main program. The code must be saved in a directory with a easy to understand file name. Note the path directory where the auxiliary code will be saved. This path is under your discretion, but you must write it down.

```
MODULE AUXILIARY;
TYPE
pro_vector = array [1..10] of integer;
PROCEDURE sum(data1: pro_vector; VAR results1: pro_vector); EXPORT;
VAR
sublength, current, tempsum: integer;

begin
        for sublength := 1 to 10 DO
                begin
                        tempsum := 0;
                        for current := 1 to sublength DO
                        tempsum := tempsum + data1[current];
                        results1[sublength] := tempsum;
                end;
end;
```

Figure B.27 Auxiliary Code

The code above should be saved with “USP” extension, and then saved it with this format, without the quotes: “filename.usp”.

Note that after the procedure declaration is the "EXPORT" keyword, this mean that this auxiliary code will export or return data to a main code. The path where the filename.usp is stored is an important data, because the main program will look for the file using the path like data to locate and load it.

So the path will be included in the main program and if this path is incorrect this will cause a "fatal error". Next is a example of a main program using the IMPORT/USE instructions and calling the above code. Note the "IMPORT" keyword, it means that the main code will import or retrieve data from an auxiliary code:

```
PROGRAM store_arrays1;
USE '/drive??/path_file??/filename.usp';
TYPE
vector = array[1..10] of integer;
VAR
first, second: vector;
i: integer;

PROCEDURE sum(data: vector; VAR results: vector); IMPORT;

begin
        for i := 1 to 10 do
        first[i] := i;
        sum(first, second);

        for i:= 1 to 10 DO
        write('first[i]:4);
        writeln;

        for i := 1 TO 10 do
        writeln;
        for i:= 1 TO 10 do
        write(second[I]:4);
        writeln;
        pause;
end.
```

Figure B.28 Import/Export examples

The program output is:

```
1 2 3 4 5 6 7 8 9 10

1 3 6 10 15 21 28 36 45 55
***PAUSE
```

The “#INCLUDE” instruction

We can use the #INCLUDE to separate the variable declaration from the main code. Then, we will write a special file with the TYPE data inside it. Modify the above code in this way:

1) Open and save another file with a file name “my_types.include”

2) In this new file, write the following code:

```
TYPE
vector = array[1..10] of integer;
```

3) In the main program, below the USE instruction, write:

```
#DEFINE $TYPES ‘/drive/path/my_types.include file’;
#INCLUDE ‘/drive/path/my_types.include file’;
```

4) Delete the Type declaration section in your main program.

Next is a complete example of the use of USP and INCLUDE files. First we should write the code for a USP file.

```
MODULE AUXILIARY;
TYPE
pro_vector = array [1..10] of integer;
PROCEDURE sum(data1: pro_vector; VAR results1: pro_vector); EXPORT;
VAR
sublength, current, tempsum: integer;

begin
        for sublength := 1 to 10 DO
                begin
                        tempsum := 0;
                        for current := 1 to sublength DO
                        tempsum := tempsum + data1[current];
                        results1[sublength] := tempsum;
                end;
end;
```

Figure B.29 USP File using Export

Then we write the code for the include file:

```
TYPE
vector = array[1..10] of integer;
```

Both stored with the same names. Then the Main program will be:

```
PROGRAM store_arrays1;

USE '/drive??/path_file??/filename.usp';
#DEFINE $TYPES '/drive/path/my_types.include file';
#INCLUDE '/drive/path/my_types.include file';

VAR
first, second: vector;
i: integer;

PROCEDURE sum(data: vector; VAR results: vector); IMPORT;

begin
        for i := 1 to 10 do
        first[i] := i;
        sum(first, second);

        for i:= 1 to 10 DO
        write('first[i]:4);
        writeln;

        for i := 1 TO 10 do
        writeln;
        for i:= 1 TO 10 do
        write(second[i]:4);
        writeln;
        pause;
end.
```

Figure B.29 Example using Define, Include and USP

Next is the program output, please check with your instructor or your computer room manager the right path to store and call programs.

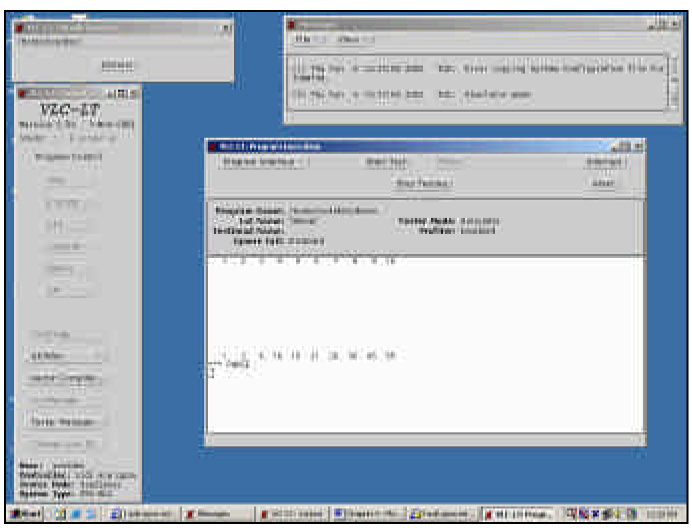


Figure B.30 Output example Define

Appendix "D"

Measurement limitations

Every tester and bench instrument must ultimately correlate to standards maintained by a central authority, such as the National Institute of Standards and Technology (NIST). This U.S. government agency is responsible for maintaining the standards for pounds, gallons, inches, and electrical units such as volts, amperes, and ohms. The chain of correlation between the NIST and the tester's measurements involves a long series of calibration stages [2].

The VLCT-256 used in this project, has a calibration due date by 03/09/2001, hence a calibration is considered necessary. The digital voltmeter Agilent 34401A used to verify and adjust the values of fixed and variables resistors, does not show any evidence of hardware calibration. In consequence there is no warranty of accuracy in the readings of this project and furthermore, the objective of this project was the development of testing methods, in consequence the instrument's calibration and validation should be performed by professional services.

Table D1 shows the tolerances for each measurement, component and device used for this project. According with this table, there is a possible error in readings of 5% maximum due the power supply accuracy plus the fixed resistor tolerance by 10%. No available information about tolerance of variable resistor.

	Voltage range	Voltage resolution	Voltage accuracy	Current accuracy no bypass	Current accuracy 2uf at 500ua	Current accuracy 2uf bypass 5mA-2A	Tolerance
VLCT Power Supplies	-10V to +12V	6 mV	24mV	0.5% of range	5% of the range	2% of the range	
Differential voltmeter	2.5mV to 10V	16 bit	4 lsb + 20up				
Agilent Voltmeter			0.06%				
Fixed resistor 1 KO							10%

Table D1 Tolerances

Therefore the only available method to improve readings in the VLCT case is by software, using a mathematical routine written in software [2]. This is proposed as future work.

Appendix “E”

Circuit, differential voltmeters and relays notes

On a VLCT there are two different voltmeter inputs fed into a single ADC, which limits the usage of the two inputs to one measurement at any given time [45]. The standard differential voltmeter has twelve ranges available from 2.5mV up to 10V. The high-voltage differential voltmeter has only three ranges from 1V to 5V. The inputs on the standard differential voltmeter can only be 10V relative to ground and this voltmeter were chosen to perform all voltage measurements because the 10V range is more suitable than 30V scale. Figure E1 shows the standard differential voltmeter, the analog channels, and the relays paths.

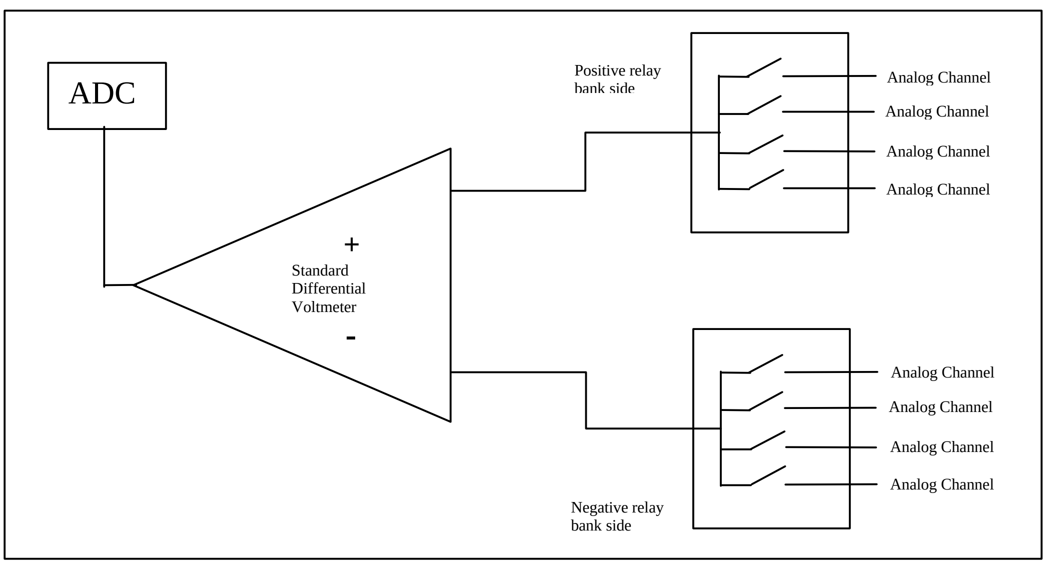


Figure E1 Differential voltmeter and channels connections

The differential voltmeters always work with differential inputs and require both inputs to be connected in pairs.

For example:

- Channel A35 (+) and channel A40 (-)
- Channel A43 (+) and channel A48 (-)
- Channel A51 (+) and channel A56 (-)
- Channel A59 (+) and channel A64 (-)

A single-ended measurement with respect to ground must connect the negative side of the voltmeter to ground through an analog channel.

To perform a differential voltmeter measurement, follow these instructions:

1. Only one measurement at any time
2. Positive channel connected to positive DUT terminal
3. Negative channel connected to negative DUT terminal
4. Declare the analog channels
5. Connect the differential voltmeter through analog channels (both positive and negative)
6. Program the voltage range of the differential amplifier
7. Perform the reading
8. Disconnect the channels
9. Repeat procedure if necessary

The following is an example, note that the instructions are directly related to the channels, differential voltmeter, and relays are in bold. The VLCT Pascal at the instruction level does not use the "relay" word but the program closes and opens the relay that is directly connected to the channel using the instruction "*AnalogChanSet*". More examples of programming are located in appendix "A".

PROGRAM FUNCTIONAL_TEST_3

```
program Functional_Test_3;

{*********************************************************** }
{ * Functional Bipolar Current-Mirror Test program          * }
{ * Measures voltage in both resistors                     * }
{ * Calculate the current passing through                  * }
{ * Execute the Test and writes result in output file       * }
{ * On this version input file is used                      * }
{ * This version uses the vlct power supply for DC SWEEP    * }
{ * Program Created by Sergio Couttolenc Valdes             * }
{ * June 2004                                               * }
{*********************************************************** }


{********************** TYPE SECTION ***********************}

TYPE
{here we define the list of channels                     * }
{because they are a user defined variable                * }
{POS_35 and NEG_40 are the first Diff-Volt used to measure   * }
{voltage drop in Resistor reference in collector Q1       * }
{POS_43 and NEG_48 are the second Diff-Volt used to measure  * }
{voltage drop in Load Resistor in Collector Q2            * }

AnalogChanList=(POS_35, NEG_40, POS_43, NEG_48, Analog_1);

{now the variables to setup the power supply}
{we need to use them because we are using the DPS1A}
PinList = (PIN1);  {just pin 1}
PinCardList = (PC1);  {pincard 1}
TestNames = (Test1);
DCSetup = (DCSetup1);  {name of the dps1a function}
{************************************************************}



{*************** VARIABLES DECLARATION SECTION  ***************}

VAR
{This is the variable that stores the measurement}
Ohm_Ref:Treal;  { used to store the value in ohms of RREF }
Ohm_Output:Treal;         { To store ohms of ROUTPUT       }
V_REF:Treal;              { To store the meas voltage            }
V_Output:Treal;           { To store the meas voltage      }
I_REF:Treal;              { To store the Current reference  }
I_Out:Treal;              { To Store the Output Current     }
Output_File:Text;         { Link to output text file      }
```

```
Input_File:Text;  { Link to input text file        }


{variable definition for the DC ramp, using TREAL types !!}

 step, init, final, timestep, V:treal;

{timestep is the variable to be used for time between steps}
{step, is the variable for the space between voltage changes}
{init, is the variable for initial voltage}
{final, is the variable for final voltage}
{timestep, is the variable for time between voltage changes}
{V, is the variable that goes into the DC PS}
{RES_UNIT is the unit variable for resistance}

TIME_UNIT, VOLTAGE_UNIT, CURRENT_UNIT, RES_UNIT :TREAL;
USER_INPUT:real;


DutToChan {pin 1 of DUT to Pin 1 from DPS1A}
1:1;

{************************************************************}


begin
{!!!!!!!!!!!!!!!!!!!!!!!!THE MAIN CODE STARTS HERE!!!!!!!!!!!!!!!!!!!!!!!}


{**************************************************}
{          REAL-> TREAL CONVERSION TABLE                  }
TIME_UNIT:=1S;                   {Time Base Unit-> Seconds}
VOLTAGE_UNIT:=1V;   {Voltage Base Unit-> Voltage}
CURRENT_UNIT:=1A;   {Current Base Unit-> Ampere}
RES_UNIT:=1Z;              {Resistor Base Unit-> Ohm}
{*************************************************}



{******************* OUTPUT FILE LINK ****************}
{now we prepare the input and output files}
Reset(Input_File, 'Functional_in_3.dat');
Rewrite(Output_File, 'Functional_out_3.dat');
{****************************************************}




{ **************** MESSAGE SECTION **********************}

writeln('FUNCTIONAL PROGRAM NUMBER 2');
```

```
writeln;
writeln('Functional Bipolar Current-Mirror Test program ');
writeln('Measures voltage in both resistors');
writeln('Calculate the current passing through');
writeln('Execute the Test and writes result in output file');
writeln('On this version input file is not used');
writeln('This version uses the vlct power supply');
writeln('Program Created by Sergio Couttolenc Valdes');
writeln;
pause;
ERASESCREEN;
{**************************************************************}



{********************* INPUT DATA READ SECTION ********************}
{HERE WE READ DATA FROM INPUT FILE!}

Readln(Input_File, init);
writeln('Initial voltage: ', init);
Readln(Input_File, final);
writeln('Final voltage:', final);
Readln(Input_File, step);
writeln('Step voltage value:', step);
Readln(Input_File, timestep);
writeln('Time between steps:', timestep);
Readln(Input_File, Ohm_Ref);
writeln('Resistor in Q1:', Ohm_Ref);
Readln(Input_File, Ohm_Output);
writeln('Resistor in Q2:', Ohm_Output);

pause;
{***********************************************************************}



{********************* DC POWER SUPPLY PIN DECLARATION SECTION ************}
{this section is a DPS1A necessary step}
PinTableOpen;
        PinSet(PIN1, 1);
        PinCardSet(PC1, S_PINCARD1);
PinTableClose;
{**************************************************************************}
```

```
{*********************** ANALOG CHANNEL DECLARATION SECTION *************}
AnalogChanTableOpen;
{begin the channel declarations}
        AnalogChanSet(POS_35, S_Analog, 35); {initialize ch 35 as POS_35}
        AnalogChanSet(NEG_40, S_Analog, 40); {initialize ch 40 as NEG_40}
        AnalogChanSet(POS_43, S_Analog, 43); {initialize ch 35 as POS_43}
        AnalogChanSet(NEG_48, S_Analog, 48); {initialize ch 40 as NEG_48}
        AnalogChanListSet(Analog_1, POS_35, NEG_40, POS_43, NEG_48);
        {in the upper line we define both pines in Analog_1}
AnalogChanTableClose;
{End the Channel declarations}
{************************************************************************}


{********************** POWER SUPPLY SETUP SECTION *********************}
{first we setup and connect the power supply with 0 volts}
DCSetupOpen(DCSetup1);
DCSet(0v, 0v, 0v, 0v, 0ma, 0ma, PC1); {all parameters are in cero}
DCConnect(PIN1, S_LOW, S_OPEN);
DCSetupClose;

{now we select the power supply}
SetupSelect(DCSetup1);

{************************************************************************}


{**************************** ATTENTION CYCLE STARTS HERE !!! ***********}
{next, the cycle that performs a DC Sweep}
V:=init; {variable that is used like incremental index for the cycle}

While(final > init) DO
begin   {CYCLE STARTS HERE!!!}
        writeln(V);
        writeln('DC RAMP IN PROGRESS!!!');
        writeln('');

        {next we setup the voltage and current limit for DSP1A};

        SupplySet(S_DPS1A, V, 500ma, S_IGAIN_1X, S_HIGH_CURR_OVERRIDE_BIAS);
        wait(timestep); {time to change to a new step}
        writeln('Voltage on 1A is ',SupplyRead(S_DPS1A, S_VOLTAGE));
        V:= V + step; {here we do the increment of voltage for dc ramp};
        init:=V;
```

```
{*********************************************************************}


{********************** VOLTMETER AND ANALOG MEAS SECTION **************}

{now we connect the pines for measure voltage drop in RREF}
AnalogSetConnect(POS_35, S_Diff_Pos); {connect 35 as Diff_Pos}
AnalogSetConnect(NEG_40, S_Diff_Neg); {connect 40 as Diff_Neg}
{new we choose the range}
AnalogSet(POS_35, S_Diff, 10V); {10 volts range}
{now we initialize the differential voltage read}
AnalogReadInit(POS_35, S_Diff, false); {boolean is false for DC}
wait(50ms); {wait for reading}
{we execute the reading}
V_REF:= AnalogRead;  { Stores the voltage Drop of RREF }
{now we disconnect the pines for measure voltage drop in RREF}
AnalogSetConnect(POS_35, S_OPEN); {disconnect 35 as Diff_Pos}
AnalogSetConnect(NEG_40, S_OPEN); {disconnect 40 as Diff_Neg}


{now we connect the pines for measure voltage drop in ROUTPUT}
AnalogSetConnect(POS_43, S_Diff_Pos); {connect 43 as Diff_Pos}
AnalogSetConnect(NEG_48, S_Diff_Neg); {connect 48 as Diff_Neg}
{new we choose the range}
AnalogSet(POS_43, S_Diff, 500mV); {500mV volts range}
{ now we initialize the differential voltage read}
AnalogReadInit(POS_43, S_Diff, false); {boolean is false for DC}
{we execute the reading}
V_Output:= AnalogRead;  { Stores the voltage Drop of ROUTPUT }
{**********************************************************************}


{**************** COMPUTING RESULTS SECTION ************************ }

I_REF:= (V_REF/Ohm_Ref); {calculates the current reference}
I_Out:= (V_Output/Ohm_Output); {calculates the Colector #2 current}

{*******************************************************************}


{ ************ SECTION FOR RESULTS DISPLAY IN SCREEN **************** }
writeln('Test Results are...');
writeln;
writeln('Voltage of DPS1A is: ', SupplyRead(S_DPS1A, S_VOLTAGE));
writeln('Output current DPS1A is: ', SupplyRead(S_DPS1A, S_CURRENT));
writeln('The voltage in Rref: ', V_REF);  {display meas in the screen}
```

```
writeln('The voltage Routput: ', V_Output); {display v_output}
writeln('The Current Reference: ', I_REF);  {display i_ref}
writeln('The Output Current: ', I_Out);    {display i_output}
Writeln;
Writeln('PROGRAM IS SENDING DATA!!! to file "Functional_out_2.dat" ');
wait(500ms);
{******************************************************************}



{ ************ OUTPUT FILE SECTION **********************************}

{now we sent the data}
writeln(Output_File, 'FUNCTIONAL BIPOLAR CURRENT-MIRROR TEST RESULTS');
writeln(Output_File, 'Voltage of DPS1A is: ', SupplyRead(S_DPS1A, S_VOLTAGE));
writeln(Output_File, 'Current from DPS1A is: ', SupplyRead(S_DPS1A, S_CURRENT));
writeln(Output_File, 'The voltage in RREF: ', V_REF);  {save v_ref}
writeln(Output_File, 'The voltage in ROUTPUT: ', V_Output); {save v_output}
writeln(Output_File, 'The Current Reference: ', I_REF); {save i_ref}
writeln(Output_File, 'The Output Current: ', I_Out);


end;
{************* ATTENTION CYCLE ENDS HERE !!!! ************************}

{******************************************************************}



{**************** POWER SUPPLY SHUTDOWN SECTION ********************}
{now the shutdown power supply section}
SupplySet(S_DPS1A, 0v, 1ma, S_IGAIN_1X, S_HIGH_CURR_OVERRIDE_BIAS);
SuppLyDisconnect(S_DPS1A);
wait(50ms);
writeln('DPS1A is off');
{*******************************************************************}



{************* END PROGRAM SECTION **********************************}
writeln;
FileClose(Input_File);
FileClose(Output_File);
writeln('data send to file');
pause;
{ ******************** here program ends ***************************}


end.
```

Figure E2 is the physical localization of the analog channels and is very important to note the orientation of the board.

- Analog channels A35 to A64 are located to the right
- Pin of DPS1A is located to the left

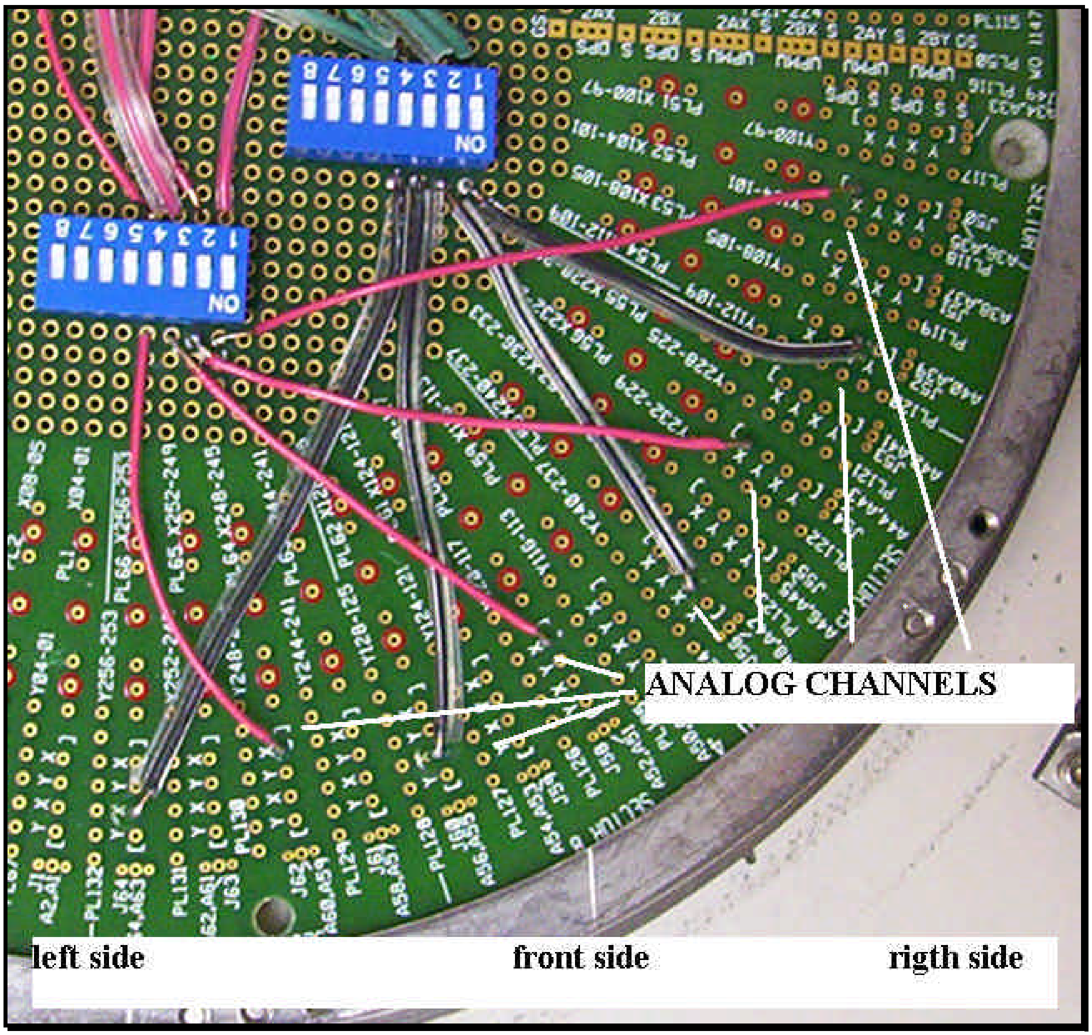


Figure E2 localization of analog channels

Figure E3 shows the physical connections for all the examples of this project:

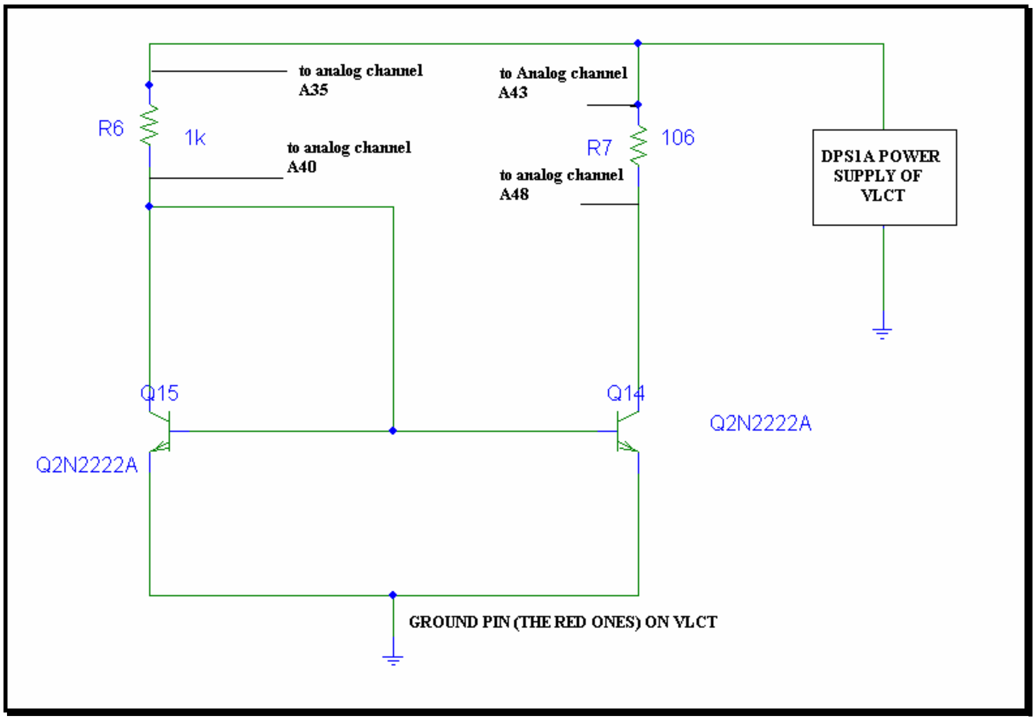


Figure E3 Physical connections