10-24-2008

Development of a Software-Defined Integrated Circuit Test System Using a System Engineering Approach on a PXI Platform

Alfonso S. Flores
University of South Florida

Scholar Commons Citation
Flores, Alfonso S., "Development of a Software-Defined Integrated Circuit Test System Using a System Engineering Approach on a PXI Platform" (2008). *Graduate Theses and Dissertations.*
https://scholarcommons.usf.edu/etd/244

Development of a Software-Defined Integrated Circuit Test System Using a System Engineering Approach on a PXI Platform

by

Alfonso S. Flores

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Electrical Engineering
Department of Electrical Engineering
College of Engineering
University of South Florida

Major Professor: Wilfrido Moreno, Ph.D.
Paris Wiley, Ph.D.
James Leffew, Ph.D.

Date of Approval:
October 24, 2008

Keywords: Software-Defined Instrumentation, National Instruments, Semiconductor,
Failure Analysis, Diode, Cost-of-the-Shelf

Dedication

This thesis is dedicated to my parents Alfonso and Maria, my sisters Mariela and Mariana, my brother in law Miguel, my nephews Miguel Andres and Daniel, my fiancé Areany.

Acknowledgements

I would like to give special thanks to my sister and brother in law, Mariela and Miguel, for their continuous and unconditional support. Their support has been positive, unwavering and continuous since I was young. Their patience and confidence in me has been a stabilizing force in my life.

I extend my eternal gratitude to my major professor Dr. Wilfrido Moreno for his encouragement and valuable guidance during my tenure as a Masters Student at the University of South Florida.

I extend my thanks to Dr. Paris Wiley and Dr. James Leffew for agreeing to serve on my supervisory committee.

I thank my friends and colleagues for their support during this research endeavor.

Table of Contents

List of Tables

List of Figures

Development of a Software-Defined Integrated Circuit Test System Using a System Engineering Approach on a PXI Platform

Alfonso S. Flores

Abstract

There are various types of test performed on Integrated Circuits, (IC), for detecting and locating defects and faults during failure analysis. Functional, logic, parametric and I_{DDQ} tests are among the most common. Functional IC tests are designed to verify whether the IC performs its intended function. Logic tests verify the logic operation of gates and registers. AC and DC parametric tests are used to measure time, voltage and current-varying parameters associated with the operational limits of the IC. Test parameters in parametric testing include, among others, propagation delay, operating current and signals rise and fall time.

Currently, almost all ICs are manufactured or refurbished in Asia. A greater portion of the ICs are processed in China and Malaysia. Presently issues with component reliability are compromised since the ICs are not tested before they leave the factory, are sometimes only remarked with different part numbers and date codes or resold even though they do not work properly. These activities lead to a high level of uncertainty among consumers all over the world.

The purpose of this research was the design of a software-defined semiconductor validation test system using the PCI eXtension for Instrumentation, (PXI), platform. The test system was to be capable of performing Open and Short Circuit Tests for CMOS components. Open and Short Circuit Tests verify for faults at the protection diode circuitry of CMOS chips level. The test system reduces the overall test timing compared to the tests performed by a functional instrument such as a curve tracer.

PXI is a modular instrumentation platform originally introduced in 1997 by National Instruments, (NI). PXI is an open, PC-based platform for test, measurement and control. PXI possesses the highest bandwidth and lowest latency with modular inputs and outputs for high-resolution from DC to RF frequencies. PXI was designed for measurement and automation applications that require high-performance.

Concepts associated with the Systems of Systems Engineering, (SoSE), approach were applied to this research in order to facilitate the design process for the test system. The objective was to apply Systems Engineering methodologies to the design of this particular test system.

Chapter 1

Introduction and Motivation

1.1 Introduction

A software-defined Integrated Circuit Test System allows engineers to quickly adapt to challenging test requirements. The functionality of modular instrumentation is characterized through user-defined software residing in the test workstation. Through software, engineers can program a modular instrumentation system to function as a user-defined instrument using programmable I/O and built-in shared clocks and triggers. PXI is an example of a platform for building modular automated test systems based on software-defined instrumentation concepts.

A Software-Defined Test System, (SDTS), can be as simple as a digital multimeter whose operating mode and measurement are controlled and analyzed by a computer. A SDTS can also be as complex as a system containing dozens of specialized test instruments, which are capable of automatic test and diagnoses of faults in complex electronic systems such as tests performed on sophisticated printed circuit boards. A SDTS for semiconductor validation possesses the capability to test a wide range of electronic devices and systems. Such testing extends to the component level and includes all components such as resistors, capacitors, inductors, integrated circuits for printed circuit boards and assembled electronic systems. In addition, SDTS is presently being used in many other industries, both commercial and military, to test various applications that range from testing aerodynamics performance in airplanes to medical devices, [1].

There are various types of test performed on Integrated Circuits, (IC), for detecting and locating defects and faults during failure analysis. Functional, logic, parametric and I_{DDQ} tests are among the most common. These tests are performed in combinations at the wafer, at the bare-die, at the package, at the assembly and at the system levels. Functional IC tests are designed to verify whether the IC performs its intended function. Logic tests verify the logic operation of gates and registers. AC and DC parametric tests are used to measure time, voltage and current-varying parameters associated with the operational limits of the IC, [2]. To insure proper implementation of software-define test systems, all the modules and interfaces must work in synergy. Therefore, applying system design principles to the design of the test system can simplify design work and can avoid problems involved with integrating the modules efficiently.

PXI is a modular instrumentation platform. It was designed for measurement and automation applications, which require high-performance and a rugged industrial form-factor. The flexible, modular and scalable architecture of the PXI platform provides the possibility of creating customized software-defined test systems, which can be easily reconfigured to meet different test needs and requirements.

1.2 System of Systems Engineering Approach

System of Systems, (SoS), engineering is a set of developing processes, tools and methods for designing, re-designing and deploying solutions for complex systems. SoS is widely used in military applications and is increasingly being adopted by non-military companies such as auto makers, aircraft manufacturers, healthcare agencies and global communication networks.

System of systems is a term being used for a collection of dedicated systems that contribute with their resources and capabilities to support the design and development of more complex systems. Complex systems created using this approach possess the capability to offer more functionality and performance than simply the sum of each sub-system, [3].

Approaching a complex system as a system of systems helps to divide the functionality and verify the working of the subsystems independently. This design methodology also helps to enhance the interoperability and portability of the design.

As a result of the diverse methodologies and applications, there exists no single unified consensus for processes involved in System of Systems Engineering. Best practices suggest a three-phase method where the SoS problem is defined, abstracted, modeled and analyzed for behavioral patterns. Based on the overall system requirements, designers identify the subsystem requirements and then design and validate the subsystems. This approach makes it easy to test, debug and integrate the subsystems to achieve the whole system, [4].

1.3 Design Methodology

A good design methodology can help the system design process in different ways. It can help to verify the system functionality and discover early design flaws. It can help the design team coordinate and reduce the overall design effort. The design process should provide a time line for the designers and the deliverables, which will be due at different times. The design methodology involves the global approach of the chief designers to the system design. Since the projects mostly involve many teams working on the project at the same time, this methodology can facilitate coordination, which can be very productive for achieving the design goals. The system engineering methodology helps define a group of solutions. These solutions meet requirements, which can be used in the future to select a final design implementation based on additional constrains such as manufacturing cost, performance and power consumption.

1.4 Top-Down Design

The top-down design approach involves five main phases, which range from requirement identification to system integration. This approach helps the designer achieve the most favorable solution after considering all the possible alternatives. Occasionally, the design effort may focus on trying to fit the solution within the available resources. This approach, which may provide short term achievements, may lead to complications while fulfilling future needs and not necessarily yield the optimum implementation. Any application may have a large number of possible implementations. Selecting the optimal solution based on the target application can provide numerous advantages. The Top Down Design Flow is illustrated in Figure 1.

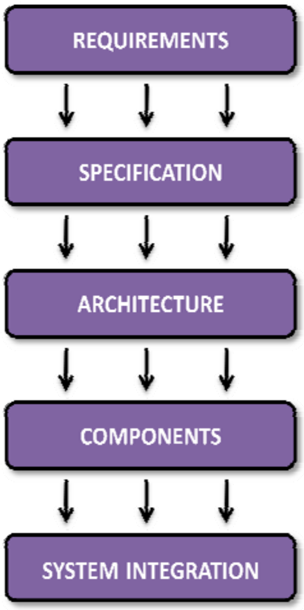


Figure 1: Top-Down Design Flow

The idea of fitting any problem into an already available solution may sound lucrative to the designer due the ease of implementation. Due to the fact that there is always a better implementation for an existing project, the top down approach helps in viewing the requirements in an objective way. In this approach the designer is faced with a large number of different implementation possibilities from which the most advantageous solution can be selected based on system requirements. The process of selecting the most favorable solution is a very important judgment. The selection process involves the analysis of all the available resources such as non-recurring engineering cost, size and power. In addition, it can lead to advantages in terms of lowering the required resources and, consequently, increasing the performance of the system.

Requirements are an abstract description of the system which involves functional as well as non functional requirements. Requirements can also include the customers' expectations about what the system must achieve. Requirements may put monetary and timing constraints on the design, which will have to be considered along with the technical specifications. Designers need to utilize these requirements and build a system, which can perform the expected tasks. It is a good practice to validate each requirement during the design process and make a final verification at the product level.

The system specifications are more focused on system implementation. They offer the designer a road map on how to design the system. The specifications have to be written carefully to ensure they meet requirements. The specifications must be comprehensible and unambiguous so that the designer knows what has to be built. Ambiguous specifications can lead to a wrong implementation, which can defeat the whole purpose of using a system approach.

The architecture describes how the functions are implemented in the system. Architecture is the structural definition of the system and is the framework within which the technical requirements of the system must be met. The architecture will involve specifications with respect to how the subsystems are to be interfaced with each other.

The components specifications are defined as functions of the architecture. The components can be hardware components as well as software components. The components perform system tasks and collectively perform the architecture tasks.

System integration is where the benefits of using the system approach can be witnessed. This phase can be straightforward if the specifications and architecture are designed correctly and validated. The whole system will be put together and the working system realized. The system engineering concepts help to keep track of specifications of separate modules, which make it possible to verify modules independently. Therefore, the system integration process is simplified, [5].

1.5 Software-Defined Semiconductor Validation Test Systems

A typical semiconductor validation test system consists of various instruments or modules used for testing analog, digital or mixed-signal components such as memory and system-on-chip, (SoC). Testing can occur either at the wafer or at the package level. Driven by the demand of electronics, computing and communication markets these test systems continue to evolve. To keep pace with innovation in the semiconductor industry, current test system products must provide more functionality and higher speeds than previous designs.

Software-defined instrumentation plays an important role in the development of test systems by providing flexibility and scalability. Functions such as timing accuracy, memory control, digital signal processing analysis, high speed inputs and outputs capability and jitter compliance are all served by software-defined instrumentation, [6].

1.6 PCI eXtension for Instrumentation

PXI is a modular instrument system designed to take advantage of the very fast data interfaces associated with PCI and Compact PCI bus systems. PXI offers a cost effective solution that satisfies the requirements of test and measurement in industry today.

The PXI standard defines mechanical, electrical and software interfaces, which are provided by PXI compliant products. These products ensure that integration costs and software costs are minimized and allow trouble free multi-vendor solutions to be implemented.

The PXI industry standard has quickly gained adoption and grown in prevalence in test, measurement and control systems since its release in 1998. It is being selected as the platform of choice for thousands of applications such as aerospace, consumer electronics, communications, process control and industrial automation.

One of the key elements driving the rapid adoption of PXI is its use of the PCI bus as the communication backplane. As the commercial computer industry drastically improves the available bus bandwidth by upgrading from PCI to PCI Express, PXI has the ability to meet even more application needs by integrating PCI Express into the PXI standard. With PXI Express, users will benefit from significantly increased bandwidth, guaranteed backward compatibility and additional timing and synchronization features.

Most PXI instruments modules are simple register based products, which use software drivers to configure them as useful instruments. Such reconfiguration takes advantage of the increasing power of computers to improve hardware, which is reconfigured to provide new facilities and features. These features are difficult to emulate in comparable bench instruments.

The PXI instruments function modules are connected to a chassis, which may include its own controller, running an industry standard operating system or a PCI to PXI Bridge, which provides a high speed link to a desktop, [7].

1.6.1 PXI Hardware Architecture

PXI systems consist of three basic components:

- The chassis,
- The system controller,

- The peripheral modules.

These components are pictured in Figure 2.

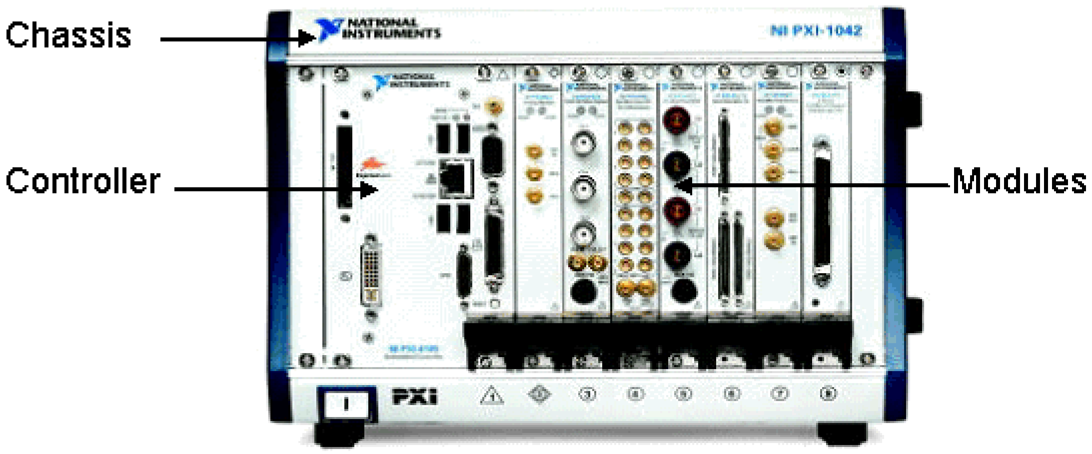


Figure 2: A PXI Chassis with a System Controller and Peripheral Modules

1.6.1.1 Chassis

PXIs' chassis provides the rugged modular packaging for the system. Chassis are available in both 3U and 6U sizes. Chassis generally range in size from 4-slots to 18-slots. Chassis are available with special features such as DC power supplies and integrated signal conditioning. The chassis contains the high-performance PXI backplane, which includes the PCI bus, timing bus and triggering bus. Using the timing and triggering buses, systems for applications requiring precise synchronization can be developed, [8].

1.6.1.2 PXI Controllers

In accordance with the PXI Hardware Specification, all PXI chassis contain a system controller slot, which is located in the leftmost slot of the chassis. Controller options include remote controllers from a desktop, workstation, server or a laptop computer. In addition, options are available for high-performance embedded controllers with Windows 2000/XP or real-time operating systems such as LabVIEW Real-Time, [8].

1.6.1.3 PXI Peripheral Modules

There are approximately 1200 products available from more than 70 members of the PXI Systems Alliance. Some of the existing modules are:

- Analog Input and Output,
- Boundary Scan,
- Bus Interface and Communication,
- Digital Input and Output,
- Digital Signal Processing,
- Functional Test and Diagnostics,
- Image Acquisition,
- Prototyping Boards,
- Instruments,
- Motion Control,
- Power Supplies,
- Switching,
- RF and Communications.

The software-defined test system developed in this research uses three main peripherals modules:

- Digital input/output module,
- Power source measurement unit,
- Switching module, [8].

1.6.1.4 Digital Input/Output (I/O) Module

In principle, these modules have a relatively simple architecture. Digital outputs can be connected to the Device Under Test, (DUT), and digital inputs of the module can be used to measure the logical outputs of the DUT.

I/O modules use a clock to sequence through a pattern of digital input and output conditions. The output signals from the module simulate an external data source while the inputs to the module record the DUT response. A clock can be provided from the module or from an external source, which includes the DUT where synchronous operation is a requirement.

Modules are usually available with different memory depths behind each of the digital outputs. These modules allow the user to select a memory depth, which meets the application requirements. The memory can also be used to store more than one waveform in segments, which can be loaded and then selected for replay, [8].

1.6.1.5 Digital Oscilloscope Module

A Digital Oscilloscope, (DSO), captures an analogue input waveform and converts the time varying signal into a digital representation. The digital representation can be displayed in much the same way as a conventional oscilloscope. However, instead of using a cathode ray tube, (CRT), to display the waveform, a soft front panel displays the signal on the monitor attached to the PXI system controller.

Many of the available functions on an analog oscilloscope can be emulated on a DSO. However, there are also many additional functions, which can be added through the use of an embedded digital signal processor or the driver software. A record of how the signal varies with time is captured as digital data, rather than as fleeting visual information on a screen. Such a record provides opportunities for data analysis and display enhancement capabilities, [8].

1.6.1.6 Switching Module

Switching systems are a key part of most test systems. Signals need to be routed to various parts of the DUT and signals from the device need to be routed to measuring equipment so that test results can be obtained.

The selection of the proper switching systems varies greatly depending upon the type of the test being performed and the number of component I/Os. Signals can vary from high current or high voltage applications to low voltage access points for 4 wire measurement on low power PCBs. There are different switch types such as single pole single throw, single pole double throw, double pole single throw and different configurations such as cascade, multiplexer, tree and crosspoint matrix, [8].

This research chose a software programmable crosspoint matrix configuration. This configuration utilized switches arranged in rows and columns. A switch was located wherever a crosspoint occurred, which allowed the row and column to be connected. There was no limit to the number of connections associated with a particular row or column. However, if more than one connection was made the load on the signal source increased. The crosspoint matrix configuration is illustrated in Figure 3.

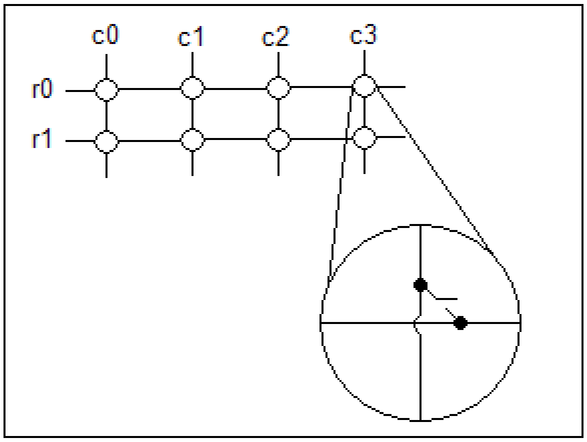


Figure 3: Crosspoint Matrix Configuration

1.7 Motivation

Facilitation of the design of a software-defined test system for semiconductor testing, which followed a System of Systems Engineering approach was the motivation behind the research reported in the thesis.

Test engineers in industries ranging from aerospace to consumer electronics are facing the challenge of increasingly complex designs with shrinking timelines and budgets. To address these issues engineers and scientists are incorporating new test and measurement technologies, which are capable of meeting complex design requirements while keeping testing costs on budget.

One issue facing test engineers is that test instrumentation is not updated as rapidly as the testing needs for new devices. The functionality of these complex devices such as most smart phones is being defined by the embedded software, which gives design engineers the ability to add features faster than ever before. This is increasingly challenging for many test engineers since most stand-alone instruments often lack the measurement capabilities required by the latest standards due to the fixed user interface and embedded firmware. Therefore, test engineers are turning to a software-defined instrumentation approach. This approach provides the ability to quickly customize equipment and user interfaces to meet specific application needs and integrates testing directly into the design process, which reduces development time. PXI is an example of a widely used software-defined instrumentation standard for building modular, reconfigurable high-performance automated test systems, [9].

Chapter 2

Background

2.1 Related Work

The scopes of test challenges, which are encountered in the military and aerospace fields, are good examples of how software defined instrumentation plays an important role in solving these challenges. Not so many years ago, testing for Mil-Aero products was mostly accomplished via specific-purpose test systems. These systems were often designed as part of a program by a prime or sub-contractor test systems group. Presently, it is much more likely that test requirements are met with mostly commercial off-the-shelf, (COTS), components. Still, the requirements imposed on a given test system are evolving at the same fast rate as test instrument technology.

Economic pressures on Mil-Aero programs and test management are tending toward the same conditions as found in the commercial field. The primary pressures relate to reduction of capital expenditures, adhere to bring-up schedules and decrease test time, which would result in reduced time to market. Additionally, there is pressure to reduce the total cost of testing even as the functionality and speed of the device or DUT are becoming more complex.

There is an enormous engineering effort behind U.S. government programs that take on challenges such as the problems of battlefield communication or the problem of inter-agency communication in the event of a catastrophe. With new trends such as battlefield networking with self-repair, heightened security measures and agile radio technology the landscape for testing is quite complex.

The reality of multi-vendor sourcing to address costs for the radio components threatens the need for successful seamless operation over the several protocols that the radio modules must support. This makes interoperability testing a very high priority. Use of Golden radios, which are known to work perfectly as designed, for bit error rate, (BER), testing in manufacturing is very common. This approach is much less practical for modular radios integrated from bins of parts, components or modules from diverse vendors.

Traditionally, high volume test applications for Military-Aerospace products are rare. However, the DUT test volume for electronically steered phased array systems can escalate significantly. The test requirements are very demanding. For example, test equipment must have the capability to measure power levels that exceeds 50 dBm, (pulsed), and third harmonic bandwidths that can reach 36 GHz or higher. In addition, measurements to less than one degree of phase accuracy in some applications are required.

In addition to the technical challenges associated with the applications presented, there are cost issues, which need to be managed. It can take hours to calibrate a system, which operates through 36 GHz. Failure rates and repair times must be carefully factored into capacity models. The implementation time and expertise level required for the application can be critical. The ability of the system to prove that it is functioning properly can have surprising leverage in total test cost. System longevity and system upgradeability are equally critical. These parameters determine the mission fulfillment efficiency of a test system. Costs related to these parameters are sometimes overlooked when focusing on the capital costs at the beginning of a program.

Solutions to the applications' challenges outlined are increasingly offered in the form of software defined test instrument, (SDI), systems rather than as conventional "rack and stack" systems. Based on the configuration of a group of COTS subsystems, SDIs perform major functions of the measurement science. However, they are not delineated along the lines of classic instrument technology. For example, a synthetic instrument applies the combination of a down-converter and digitizer to collect raw measurement data. The synthetic instrument negates the need for a network analyzer, spectrum analyzer and modulation analysis instruments. The measurement results, which would normally come from a classical instrument, are processed using DSP technology. This is the central concept of the synthetic system capability. Using this approach, the test engineer also has finer-grained access to the "knobs" or setting variability of the measurement procedure. Stimulus and response measurement channels are all multi-purpose in capability.

Important Mil-Aero customers are clearly delineating their expectations for the use of modular microwave test systems with software-defined instruments for the future. Current and future Mil-Aero test systems will be implemented using scalable slices of more integrated technology. These slices will feature industry standard board instrument platforms and industry standard software. There are two examples in the greater test market where modular technology evolution has already been demonstrated.

The IC test industry determined that meeting the diverse requirements of their customer's DUTs required board-based, modular instrumentation. Consequently, digital I/O boards, analog to digital boards, digital to analog boards, power supply boards and, more recently, RF stimulus-measurement subsystems are all offered in configurable slices in virtually any IC testers. There is not yet any particular standardization of the card cage or buses utilized across multiple vendors. However, there are certainly major customers such as Intel and the Open Architecture Consortium demanding that the industry institute appropriate standards.

PXI-served markets provide another case in point. Engineering and production test applications can be implemented more cost-effectively using a configurable PXI. A 100 M sample/s, 14-bit digitizer capability in PXI will cost approximately one tenth the price of a proprietary-architecture-based IC test system, [10]. This illustrates the leverage of higher production volumes and industry standards-based technology.

State-of-the-art modular test technologies are addressing the two application examples described previously. The first of these two applications is the software defined radio test solutions. The data bandwidths of the majority of Mil-Aero radios are still fairly modest. The majority fall well below 10 Mbits/s and many are still in the Kbit/s range. The data rate will probably increase dramatically over time. However, currently test bandwidths are being driven not by data rates. They are being driven by the rates and ranges over which frequency-agile radios can hop. Frequency hopping Hop performance has reached the 100 KHz level and hopping range requirements exceed 250 MHz.

The waveform and spectral deviations, which can occur as a function of relatively small differences in physical circuit design, put great pressure on interoperability quality assurance. To thoroughly explore possible interoperability issues, it is necessary to probe deeper into the operation of the radio. Investigating the dynamic behavior of the radio designs is required as a result, it is necessary to employ systems, which can generate and capture events, across the operating bandwidth of the radio and record all the events for a significant period of operating time. In this case, industry-leading broadband signal generators, broadband signal recorders, broadband signal analyzers and appropriate up-converter and down-converter interfaces are employed. These instruments enable capture of waveforms transmitted by a radio. They also enable overlay of practical impairments and playback of the resulting waveforms to test radio receivers with appropriate application stressors applied. Impairment signal overlay is controlled by software, which can deal directly with Doppler shift, fading, noise and other practical channel parameters. This software frees test engineers from dealing with arbitrary waveform generator loads, Fast Fourier Transforms, (FFT,) and inverse FFTs. Vector Signal Source and Channel Simulator software enables the test engineer to move from theoretical or measurement parameters directly to hardware generation. Additionally, the software provides considerable simulation and graphical tools along the way.

On the transmitter side of the radio, recorders can be implemented to capture events at very high sample rates for up to 20 seconds. This capability allows observation of the important dynamic events of a modulated radio hopping throughout a 250 MHz instantaneous bandwidth. Record and analysis software highlights the differences, which could create radio interoperability issues.

The phased array transceiver module testing application is particularly demanding. The higher counts of modules required to implement an electronically steered beam radar array demand both higher throughput as well as state-of-the-art accuracy and resolution. The modules in an array must perform and conform to strict amplitude and phase specifications over a range of frequencies, power levels and control signals. Operating bandwidths for the radar may be fairly limited. However, the test system is required to operate at three times the maximum in-band frequency of operation in order to verify harmonic and spurious behavior.

A sophisticated synthetic instrument-based system can provide multiple capabilities. For example, a system can offer the combination of a broadband up-converter and a broadband down-converter with variable bandwidth baseband capability to provide test systems to meet these requirements up to 26 GHz. The system meets rigorous requirements for calibration and measurement. These requirements can include capabilities such as very fast measurement of error-corrected s-parameters, spectral characteristics, modulation, demodulation, noise figure and/or noise power using a single non-switched channel. The system can be verified in-situ without removing the DUT. These capabilities are performed to equivalent network analyzer, error-corrected, s-parameter levels of confidence.

Military and Aerospace customers represent the vanguard of demand for RF and microwave test solutions that can be instantiated in modular, software defined instrumentation systems. Even the most successful of the rack and stack domain instrument vendors are persuaded to talk in terms of future solutions based upon modular synthetic instrument technologies.

COTS synthetic instrumentation subsystems are already available and are becoming more integrated and more modular. The systematic movement from COTS instrumentation components to COTS systems is dependent on the software provided. The software required to integrate the modular components into seamless systems is becoming less instrument and driver-specific and more measurements oriented. The advent of moving toward a measurement defined environment will lead to increasing improvements in efficiency, flexibility and cost.

As semiconductor devices become more complex, the process of testing each part completely with a traditional vector-based methodology becomes increasingly difficult. Complex systems-on-a-chip and systems-in-a-package, (SiP), require a system-level functional test more closely related to testing components placed on a printed circuit board than a typical chip test. However, they still require the high speeds demanded in production tests for the semiconductor industry. The strategy of testing a device by emulating actual real-world signals provides a better method of functional test for these types of high-speed systems, [10].

2.2 Requirements for Designing a Semiconductor Validation Test

Understanding the requirements is the most important task for designers and engineers in any development effort. In the Systems of System Engineering approach understanding the requirements is the essential process. The requirement identification process helps divide the systems into the number of subsystems necessary for arriving at the proper solution of the whole system.

The general requirements for the validation of a semiconductor system using the PXI platform must:

- Reduce time and investment for test system development and maintenance,
- Optimize test throughput,
- Reduce false failures or false passes,
- Use commercial off the shelf, (COTS), software,
- Provide reliability,
- Allow scalability in size and functionality.

In more detail, the test system and the type of test must comply with the following requirements:

- The test needs to be fast enough to satisfy approximately three or four devices per minute.
- The test system needs to be capable of testing all CMOS devices with diode protection and no restriction with respect to packaging. For this purpose, a standard interface must be developed.
- The test must validate the results according to nominal parameter values. For this requirement, a graphical user interface is necessary, which allows the operator to introduce manufacturer datasheet parameters. For example, values for maximum and minimum voltage or maximum and minimum drain current.
- At the end of the test the system must be able to generate a final report with the following information:
  - Customer Information, which includes name, business mailing address, purchase order number and total quantity of devices,

    - Part number and part description,
    - Total number of devices tested,
    - Number of devices that passed and failed.
- Save nominal parameter values for the DUT. Provide reusable test configurations for the same device in the future.

2.2.1 Hardware Limitations

Some of the hardware limitations encountered during the requirement analysis process were:

- Power Supply Voltage, (30 Vdc Max.),
- Maximum frequency for DIO approximately 20MHz,
- System capable of handling current values up to 1 A.

2.2.2 Test System Cost Analysis

Cost analysis is an important part of the requirement process since most companies have budget limitations. The cost percentages presented in this thesis are statistics taken from a cost survey associated with the development of software-defined instrumentation in companies with different development fields, [11]. The cost percentages, presented graphically in Figure 4, were determined to be:

- Hardware and Software – 36%,
- Development costs – 64%,
    - Software development – 30%,
    - System integration – 23%,

- System configuration – 7%,
- System validation – 4%.

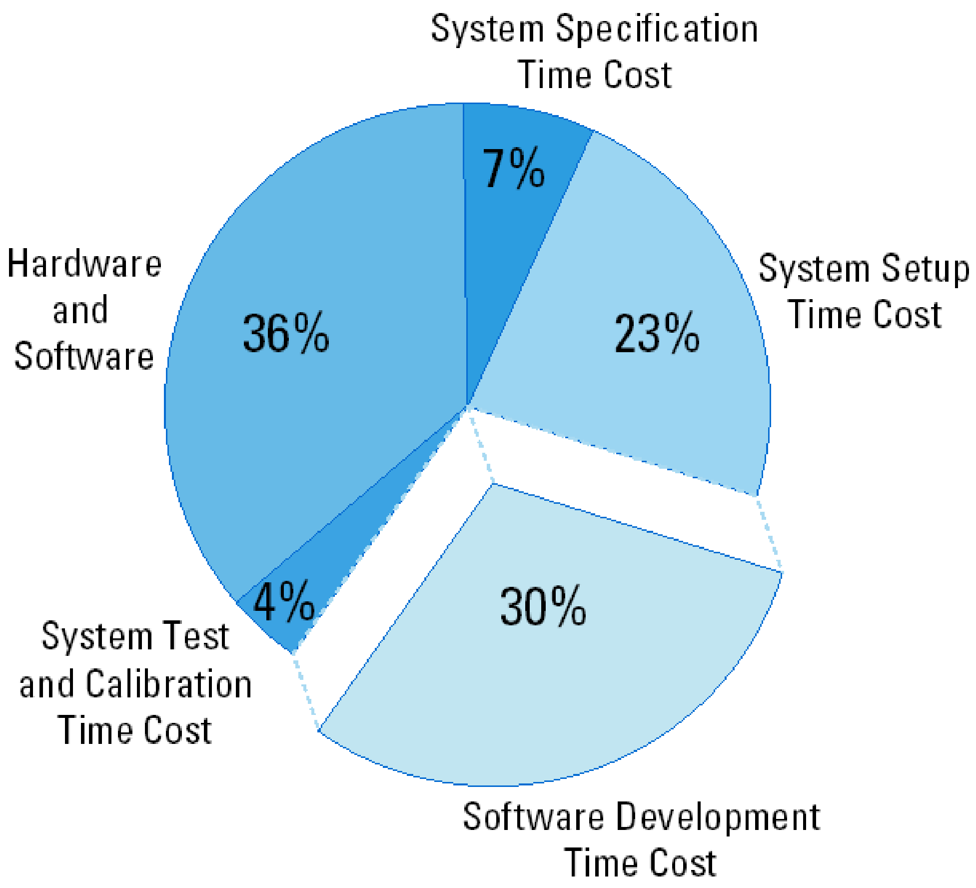


Figure 4: Test System Cost Analysis

2.2.3 Required System Elements

The typical system elements to consider, from a high-level to low-level, when building an automated test equipment system consist of the test executive, test development software, the automation controller and instrumentation as well as the fixture. These elements are illustrated in Figure 5.

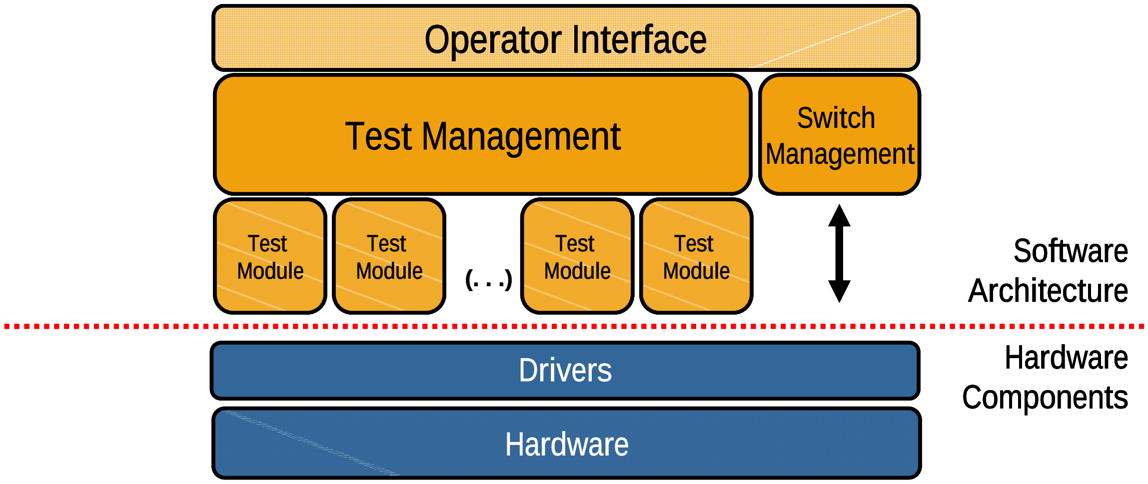


Figure 5: Tests System Software Architecture

The most important part of designing a software-defined test system is the development of an appropriate test software architecture. If the architecture is not given appropriate and careful consideration designers run the risk of limiting reuse, hindering maintenance and greatly shortening the lifetime or the test system. Although the architecture presented may seem relatively simple, the implications with respect to the test system are great.

From bottom up, the hardware and necessary specific drivers are common to every test system. The hardware abstraction layer that sits above the specific drivers plays a vital role in the system. Its main function is to protect the investment in the software above it by providing a common interface to instrument and specific drivers. This arrangement allows designers to reuse code modules with different equipment or exchange or replace equipment without affecting the test module layer.

An important fact is that eighty percent, (80%), of the development time is spent developing the actual test modules. The test module layer is where specific tests and code are defined for each product being tested. By protecting and separating this layer in modules, designers greatly improve the system reuse capability and maintenance as well as increasing system longevity.

The test management layer sits above the test modules and is designed to provide a modular and reusable framework for the tests. Its primary function is to handle the common tasks needed by each test module. These common tasks include result evaluation, reporting, database logging, user management, configuration management, switching and unit under test tracking. By handling these functions and providing a simple interface between the modules, development on the modules, which accounts for most of the test system development time, is greatly reduced. All of these common tasks are also maintained in one location making updates to the system much simpler.

Switching is often used to reduce complexity, increase automation and reduce instrument costs. However, if switching is included within a test module's code that test module is no longer reusable since it is tied specifically to the switching system, which greatly increases development costs. Use of a switch manager, which is tied to the test management system, allows switching externally to be applied to the module test step directly. This direct access to the hardware abstraction layer keeps the reuse capability high and maintenance low on switching.

The final piece of the architecture is the user interface. Typically, designers will want a standard interface applied throughout the organization, which is decoupled from the testing process or the tests themselves. This provides a common look and feel, which reduces operator training costs and allows any operator to work on any station. By decoupling the interface code from the rest of the test system, designers can test any product without requiring a change to the interface, [12].

Figure 6 presents those COTS-based software and hardware systems for each level of the system architecture, which were presented in Figure 5. The Software components and the hardware components are explained in detail in the discussion of NI LabVIEW in section 2.3.

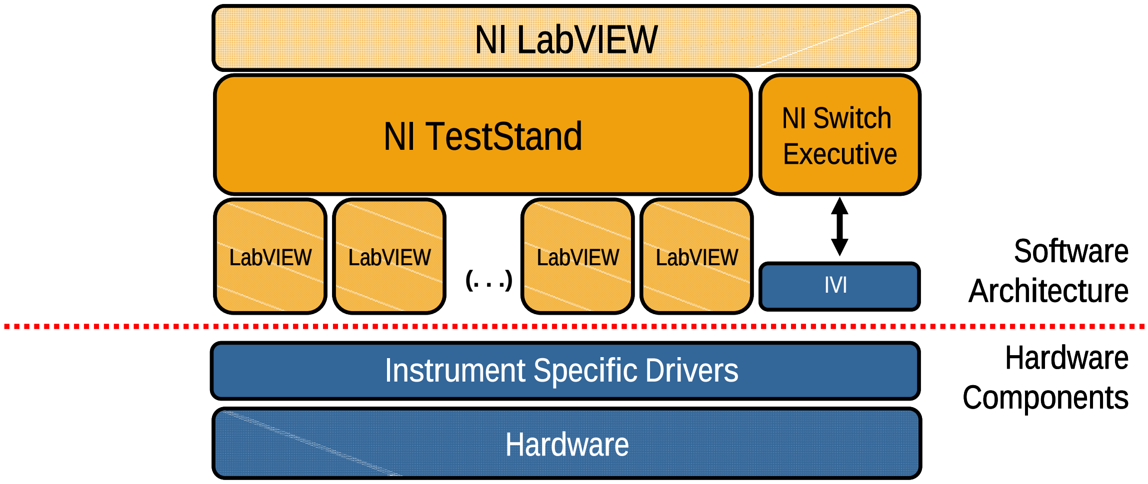


Figure 6: COTS Test System Software Architecture

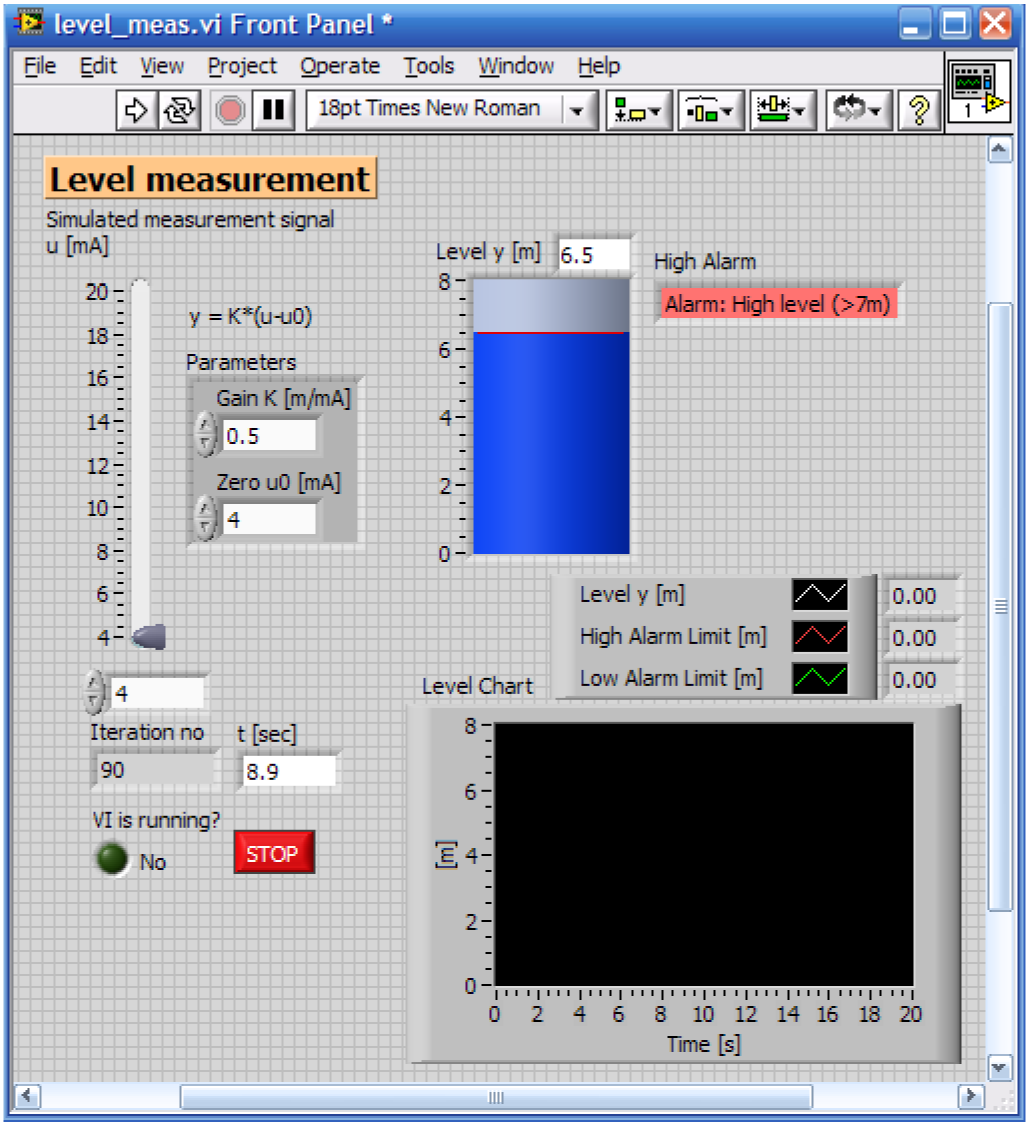
2.3 NI LabVIEW

Laboratory Virtual Instrumentation Engineering Workbench, (LabVIEW), is a platform and development environment for a visual programming language from National Instruments. Originally released for the Apple Macintosh in 1986, LabVIEW is commonly used for data acquisition, instrument control and industrial automation on a variety of platforms, [11].

The programming language used in LabVIEW, also referred to as "G", is a dataflow programming language. Execution is determined by the structure of a graphical block diagram, (the LV-source code), on which the programmer connects different function-nodes by drawing wires. These wires propagate variables and any node can execute as soon as all its input data becomes available. Since this might be the case for multiple nodes simultaneously, G is inherently capable of parallel execution. Multi-processing and multi-threading hardware is automatically exploited by the built-in scheduler, which multiplexes multiple OS threads over the nodes ready for execution.

The data-flow completely defines the execution sequence, which can be fully controlled by the programmer. Thus, the execution sequence of the LabVIEW graphical syntax is as well-defined as with any textually coded language such as C. Furthermore, LabVIEW does not require type definition of the variables. The wire type is defined by the data-supplying node. LabVIEW supports polymorphism in that wires automatically adjust to various types of data.

LabVIEW ties the creation of user interfaces, called front panels, into the development cycle. LabVIEW programs/subroutines are called virtual instruments, (VIs). Each VI consists of a block diagram, a front panel, and a connector pane. The connector pane is used to represent the VI in the block diagrams of other calling VIs. Controls and indicators on the front panel allow an operator to input data into or extract data from a running virtual instrument. However, the front panel can also serve as a programmatic interface. Thus a virtual instrument can either be run as a program, with the front panel serving as a user interface. Alternatively, when a VI is dropped as a node onto the block diagram, the front panel defines the inputs and outputs for the given node through the connector pane. This implies each VI can be easily tested before being embedded as a subroutine into a larger program. A typical front panel is presented in Figure 7.



Gain K [m/mA] 0.5

Zero u0 [mA] 4

u [mA] 4

Iteration no 90

t [sec] 8.9

VI is running? No

Level y [m] 6.5

Alarm: High level (>7m)

Level y [m] 0.00

High Alarm Limit [m] 0.00

Low Alarm Limit [m] 0.00

Figure 7: Typical LabVIEW Front Panel

The corresponding block diagram for the front panel is presented in Figure 8.

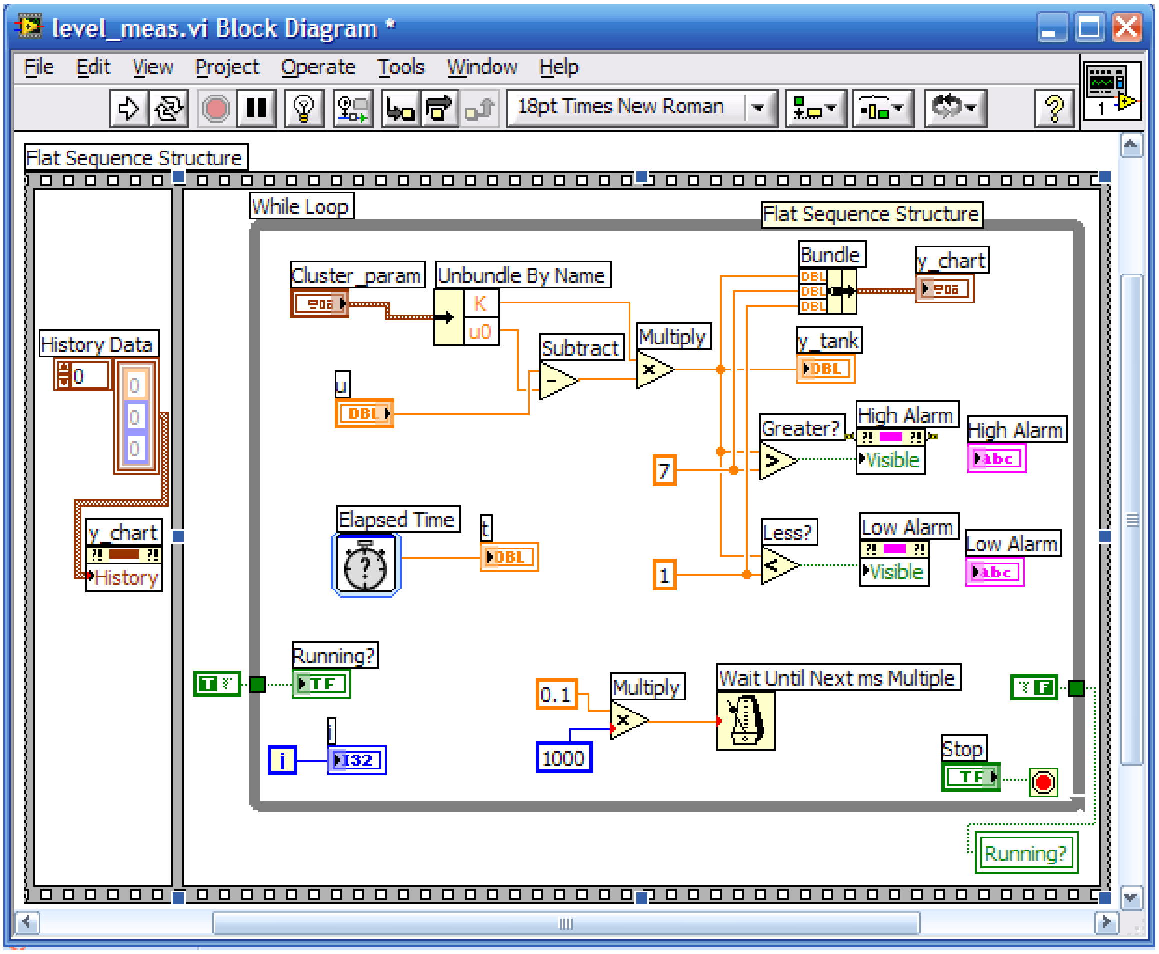


Figure 8: Typical LabVIEW Block Diagram

The graphical approach also allows non-programmers to build programs simply by dragging and dropping virtual representations of laboratory equipment with which they are already familiar. The LabVIEW programming environment, with the included examples and the documentation, makes it simple to create small applications. This is a benefit. However, there is also a certain danger of underestimating the expertise required for good quality "G" programming. For complex algorithms or large-scale code, it is important that the programmer possesses an extensive knowledge of the special LabVIEW syntax and the topology of its memory management. The most advanced LabVIEW development systems offer the possibility of building stand-alone applications.

Furthermore, it is possible to create distributed applications, which communicate by a client/server scheme. Therefore, distributed applications are easier to implement due to the inherently parallel nature of G-code.

2.4 NI TestStand

NI TestStand is ready to run test management software designed to help engineers accelerate the development of automated test and validation systems. NI TestStand can be used for developing, executing and deploying test system software. Additionally, engineers can create test sequences, which integrate code modules written in any test programming language. Sequences also specify execution flow, report database logging and connectivity to other enterprise systems. In addition, test systems can be deployed to aid production with easy-to-use operator interfaces, [11].

2.5 NI Switch Executive

One key component of National Instruments test management software is the NI Switch Executive, (NISE), which simplifies the management and control of complex switching systems. NISE is integrated with LabVIEW and LabWindows/CVI to decrease development time for validation of test and research and development test. NISE is also integrated with the NI Teststand to provide ease of use of test step in sequence execution and for production test, [11]. The following tasks can also be performed with NISE:

- Develop multiple device switch systems represented as a single virtual device,
- Create end-to-end signal routing,
- Make route selections based on signal characteristics and switch capabilities,

- Create route groups used to make intelligent resource selections when creating routes,
- Create switch routing for end-to-end calibration and maximum execution speed,
- Pre-configure routes and route groups that can be called by name at run-time,
- Generate a report of the switching system,
- Create routes graphically with an interactive schematic view of switch devices,
- Configure large and complex switching systems quickly with Excel integration.

2.6 Virtual Instrumentation

Virtual Instrumentation is the use of customizable software and modular measurement hardware to create user-defined measurement systems, called virtual instruments. Traditional hardware instrumentation systems are made up of pre-defined hardware components such as digital multimeters and oscilloscopes, which are completely specific to their stimulus, analysis or measurement function. Due to their hard-coded function these systems are more limited in their versatility than virtual instrumentation systems. The primary difference between hardware instrumentation and virtual instrumentation is that software is used to replace a large amount of hardware. The software enables complex and expensive hardware to be replaced by computer hardware. For example, an analog to digital converter can act as a hardware complement of a virtual oscilloscope or a potentiostat, which can enable frequency response acquisition and analysis in electrochemical impedance spectroscopy with virtual instrumentation.

The concept of a synthetic instrument is a subset of the virtual instrument concept. A synthetic instrument is a type of virtual instrument, which is purely software defined. A synthetic instrument performs a specific synthesis, analysis or measurement function on completely generic measurement agnostic hardware. Virtual instruments can still have measurement specific hardware and tend to emphasize modular hardware approaches, which facilitate specificity. Hardware supporting synthetic instruments is by definition not specific to the measurement nor is it necessarily or usually modular.

2.7 LabVIEW Signal Express

National Instruments LabVIEW Signal Express is interactive measurement software, with no programming required, for quickly acquiring, analyzing and presenting data from data acquisition devices and instrumentation. Signal Express can define measurement procedures by adding and configuring different functions in an interactive measurement environment. Most of these functions process input signals and produce output signals, [11].

2.8 NI PXI Chassis and Modules

The National Instruments PXI-1045 is a high-performance 18-slot chassis designed for a wide range of test and measurement applications. By programmatically configuring the trigger routing modules on the chassis backplane, triggers can be routed between devices with ease. The wide operating temperature range of 0 to 55 °C is ideal for extended-temperature environments. The Compact PCI-compatible chassis features a low-jitter 10 MHz reference clock for device synchronization, [11]. The NI PXI 1045 18-slot chassis is pictured in Figure 9.

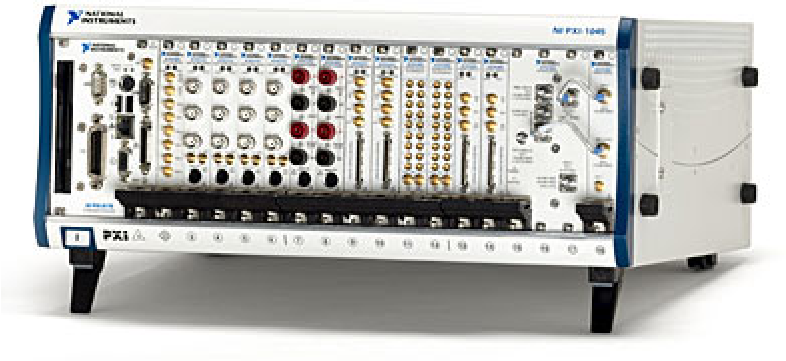


Figure 9: NI PXI 1045 18-Slot Chassis

The National Instruments PXI-8105 is high-performance Intel Core Duo T2500-based embedded controller for use in PXI and Compact PCI systems. This embedded controller, with its 2.0 GHz dual-core processor and dual-channel 667 MHz DDR2 memory, is ideal for applications requiring intensive analysis or system development. The dual-core processor feature is advantageous in multitasking environments such as Windows XP where multiple applications can be running simultaneously, [11]. The PXI-8105 embedded controller is pictured in Figure 10.

Figure 10: NI PXI 8105 Embedded Controller for PXI

The National Instruments PXI-6534 is a high-speed, 32-bit, parallel digital 5V TTL/CMOS I/O interface. It performs pattern I/O and high-speed data transfer using a wide range of handshaking protocols at speeds of 20, 40, 60 and 100 MB/s. The NI PXI-6534 delivers digital I/O coupled with large onboard memory. It features user-defined power-up states, start and stop triggering, pattern matching, and change detection. The 32 lines can be operated as individually configurable single-line I/O or as 8, 16, or 32-bit ports for pattern I/O and handshaking, [11]. The PXI-6534 digital I/O module is pictured in Figure 11.

Figure 11: PXI 6534 Digital Input/Output Module

The National Instruments PXI-4130 PXI source measure unit, (SMU), is a programmable high power source measure unit in a single slot. The NI PXI-4130 has a single isolated SMU channel, which offers a four-quadrant ±20 V output incorporating remote four-wire sense. This channel is capable of sourcing up to 40 watts in quadrants I and III and sinking up to 10 watts in quadrants II and IV. With five current ranges, providing measurement resolution down to 1 nA, this precision source was ideal for the validation of semiconductor test applications developed during this research, [11].

The PXI-4130 module also includes a utility channel, which can source either current or voltage with 16-bit set-point and measurement resolution. It can be used as an output, which provides up to 6 V and 1 A, and as a complementary power source to the SMU channel. Both channels of this SMU module can act as either a constant voltage source or a constant current source with a settable compliance limit for either mode. The PXI 4130 Power Source Measurement Unit is pictured in Figure 12.

Figure 12: PXI 4130 Power Source Measurement Unit

The National Instruments PXI-2535 high-density field effect transistor, (FET), switch matrix module features 544 crosspoints in a compact, single-slot 3U PXI form factor. The NI PXI-2535 is configured as a 4x136 one-wire matrix. The module uses field effect transistor, (FET), switch technology. Therefore, it offers unique benefits such as unlimited switch lifetime, unlimited simultaneous connections and switching speeds as high as 50,000 crosspoints/s. These features make this switch matrix module ideal for routing low-power DC signals in validation test systems of mass produced devices such as semiconductor chips and printed circuit boards.

The PXI-2535 breaks out switch connections into three VHDCI connectors. The upper left and right connectors mate with the 136 columns on the switch while the lower left connector mates with the four rows. The lower right connector, which also connects to the rows, is meant for matrix expansion, [11]. The National Instruments PXI-2535 crosspoint switch matrix is pictured in Figure 13.

Figure 13: PXI 2535 128-Crosspoint Matrix Switch

Chapter 3

Design and Implementation

3.1 Semiconductor Test

The software-defined semiconductor validation test system developed during this research is intended to perform Open and Short Circuit Tests. The test system was designed to check for faults in the protection diode circuitry of semiconductor chips. The design of the system was concerned with both hardware and software components.

The system hardware consists of a source measure unit and a high-density switching matrix. Signals from individual pins, on the device under test, are routed sequentially to the SMU through the switching matrix.

The system software is composed of switching, source and measure and test management modules. The switching software configures the high-density switch matrix for stimulus and measurements by making appropriate connections. The source and measure software programs the SMU and synchronizes it with the switching software to sequentially test the pins on the DUT. All code is incorporated into the test management software, which provides a scalable framework for easily adding supplementary tests.

3.2 Hardware Components

Semiconductor validation is generally segmented into structural and functional arenas. Structural tests ensure that the chip was built correctly. Functional tests determine whether the chip meets design specifications and performs as intended in its final environment. Open and Short Circuit Test checks for faults in the diode protection circuitry of Integrated Circuits. Therefore, these tests are associated with the structural arena. There are two protection diodes for each pin of a CMOS semiconductor device. The diode protection circuitry for the input and output pins of a CMOS semiconductor device are presented schematically in Figure 14.

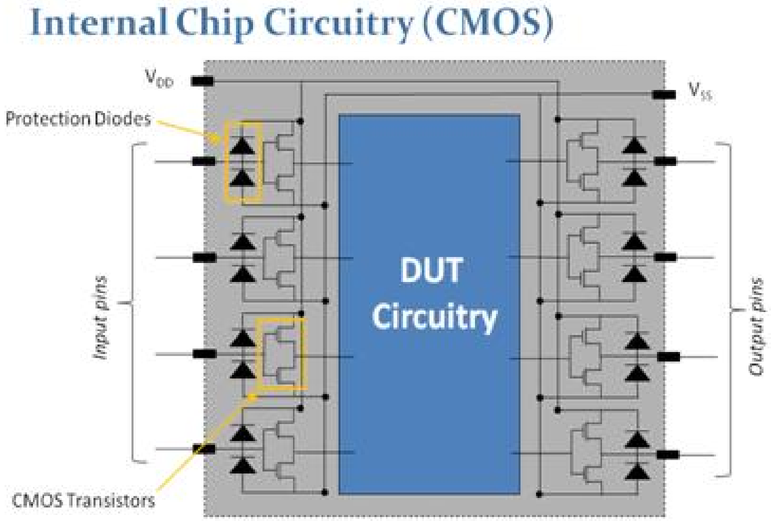


Figure 14: Internal Circuitry of a CMOS Chip

The first diode is placed between the signal pin and the supply voltage pin, (VDD), and the second diode is placed between the signal pin and ground pin, (V_{SS}), for the chip. Testing each pin on the diodes is performed using the source measure unit and scaled to multiple pins using switching hardware. The protection circuitry for every pin is tested by forcing a current through each diode and measuring the resulting voltage using the SMU. The voltage measurement is compared with a predefined value to determine the presence of open and short circuits in the circuit. Switching hardware is used in order to automate the making and breaking of connections, which are required for testing each individual pin. In order to connect the switching hardware to the SMU and individual pins of the chip, a cable and screw-terminal block is also necessary.

The implementation of Open and Short Circuit Integrated Circuit Tests requires the hardware listed in Table 1, which was explained in detail in chapter 2.

Table 1: Required Hardware Components for
Open and Short Circuit Tests in PXI

Component	Model Name	Description
PXI Chassis	PXI-1045	18-Slot 3U PXI Chassis with Universal AC Power Supply
PXI Controller	PXI-8105	2.0 GHz Dual-Core PXI Embedded Controller
SMU	PXI-4130	Source Measure Unit
Switch	PXI-2535	544-crosspoint FET Matrix Switch
Switch Cable	SCH68-68	68-pin VHDCI to SCSI cable
Switch Terminal Block	TBX-68	68-pin external screw-terminal block

The PXI platform is inherently suited for Open and Short Circuit Tests. Its modular architecture facilitates scalability and flexibility. Incorporating additional test points is as easy as adding a switch module in an available slot. Reducing test time is also possible by simply adding an SMU and conducting parallel measurements. The Open and Short Circuit Test System, designed during this research was architected in PXI using the PXI-1045 18-slot chassis and the PXI-8105 2.0 GHz Dual-core embedded controller. This Open and Short Circuit Test system was designed to test the protection diodes on the 128 pins of a single chip. The two main components of test system consisted of the PXI-4130 Source Measure Unit and the PXI-2535 544 crosspoint switch matrix. Other important components of this test system included cables and connector blocks, which facilitated signal connections to the switch. Connections to the PXI-2535 were made using an external connector block and VHDCI cables. PXI-2535 signal connection possibilities are pictured in Figure 15.

Figure 15: Connecting Signals to the PXI-2535

The top two connectors on the PXI-2535 4x136 switch matrix were used for the 136 column connections. Two VHDCI cables and two TBX-68 terminal blocks were required in order to connect to all 136 columns. The bottom left connector was used to connect signals to rows. One VHDCI cable and one TBX-68 terminal block was required in order to connect signals to rows.

The bottom right connector also provides access to the rows of the matrix module and facilitates matrix expansion. Building large matrices with the PXI-2535 is very simple and can be accomplished by connecting the bottom right connectors of two modules using a VHDCI cable.

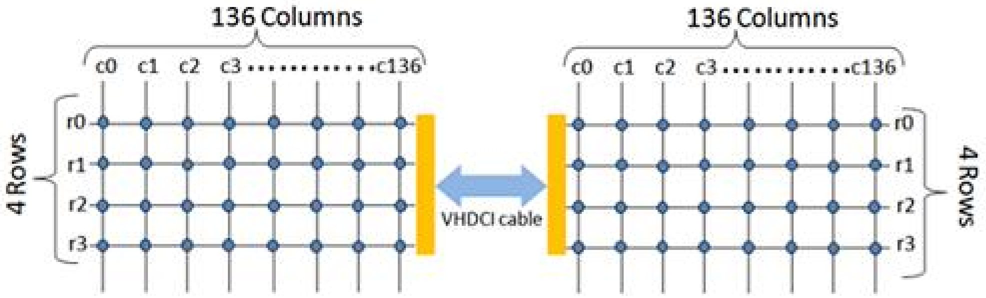


Figure 16: Building a 4x272 Matrix Using Two PXI-2535 FET Matrix Modules

3.3 Software Components

The software for the Open and Short Circuit Test System was developed using NI LabVIEW and NI Switch Executive. LabVIEW was used as the primary application development environment while switch executive was used to configure routes on the high-density matrix module.

Table 2: Required Software Components for Open and Short Circuit Tests in PXI

Component	Description
NI LabVIEW 8.5	Graphical Application Development Environment
Switch Executive 2.1	Switch Management Software

The NI DC-Power, which is the SMU driver software, and the switch executive possess intuitive APIs for LabVIEW. These APIs offer high-level programming tools for quick deployment of hardware as well as low-level functions for greater control. The NI DC-Power and NI Switch Executive APIs are presented in Figure 17.

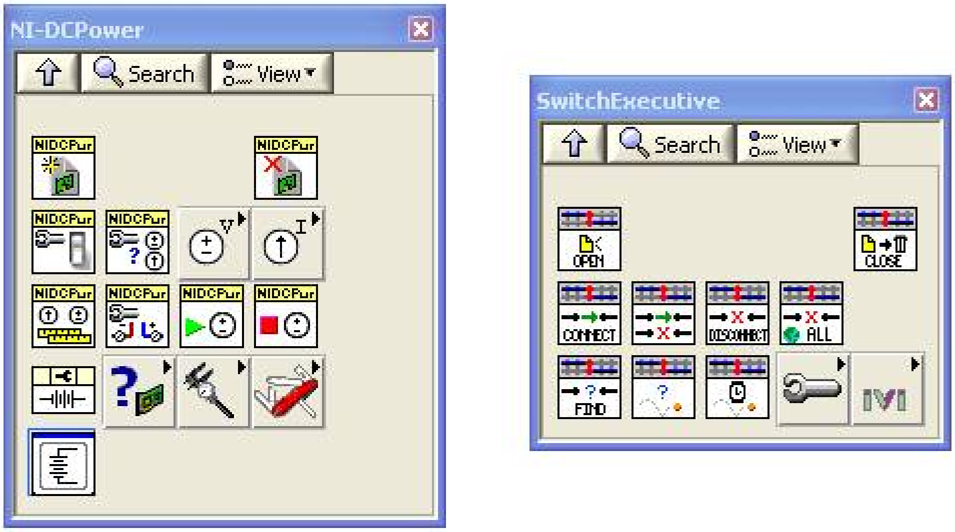
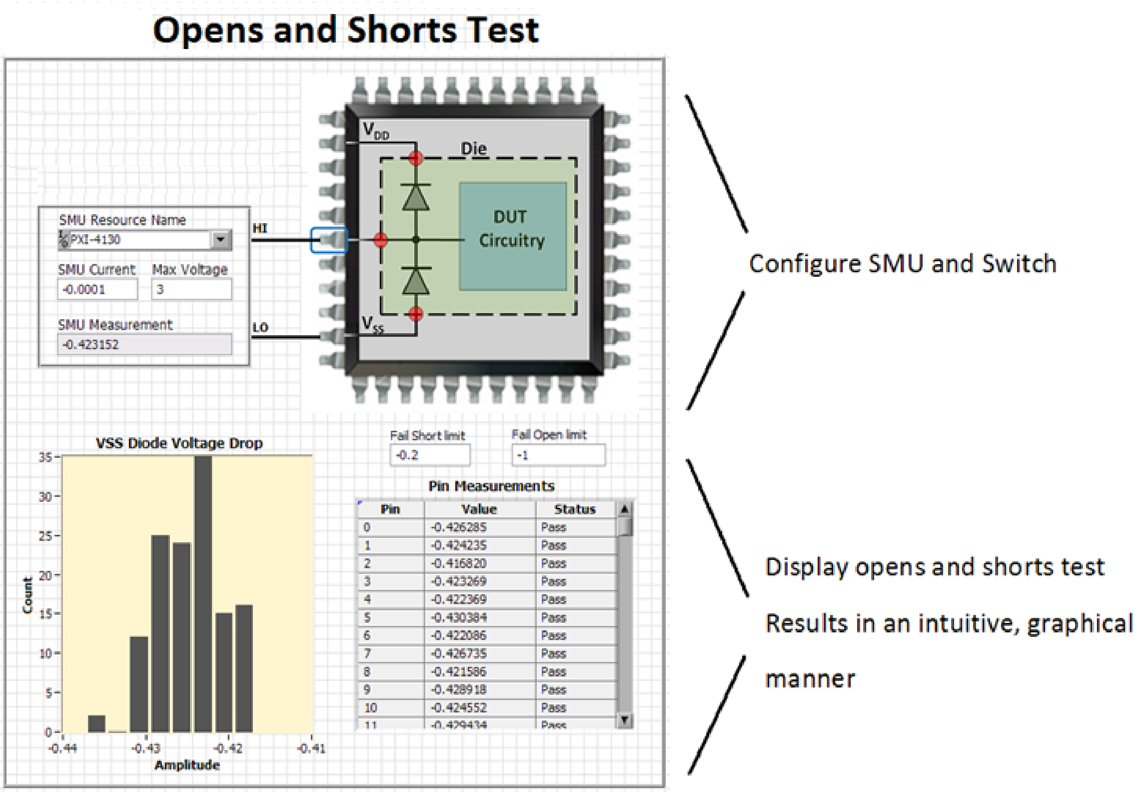


Figure 17: NI-DCPower and NI Switch Executive APIs for LabVIEW

The presentation and reporting features of LabVIEW are a major component of why the application development environment, (ADE), is so well suited for test software development. LabVIEW contains multiple graphs, charts, meters, knobs and switches, in both 2D and 3D, in order to facilitate the representation of measurement data graphically. The ADE also includes the LabVIEW Report Generation Toolkit, which facilitates the creation of reports in Microsoft Word and Excel format. In this Open and Short Circuit Test System, LabVIEWs powerful graphical user interface was used to develop an intuitive front panel. The LabVIEW front panel is presented in Figure 18. The upper block of the front panel was used to configure the SMU and switch while the lower block presents test results in a graphical manner.



SMU Resource Name: PXI-4130
SMU Current: -0.0001
Max Voltage: 3
SMU Measurement: -0.423152

Fail Short limit: -0.2
Fail Open limit: -1

Pin Measurements

Pin	Value	Status
0	-0.426285	Pass
1	-0.424235	Pass
2	-0.416820	Pass
3	-0.423269	Pass
4	-0.422369	Pass
5	-0.430384	Pass
6	-0.422086	Pass
7	-0.426735	Pass
8	-0.421586	Pass
9	-0.428918	Pass
10	-0.424552	Pass
11	-0.429434	Pass

Configure SMU and Switch

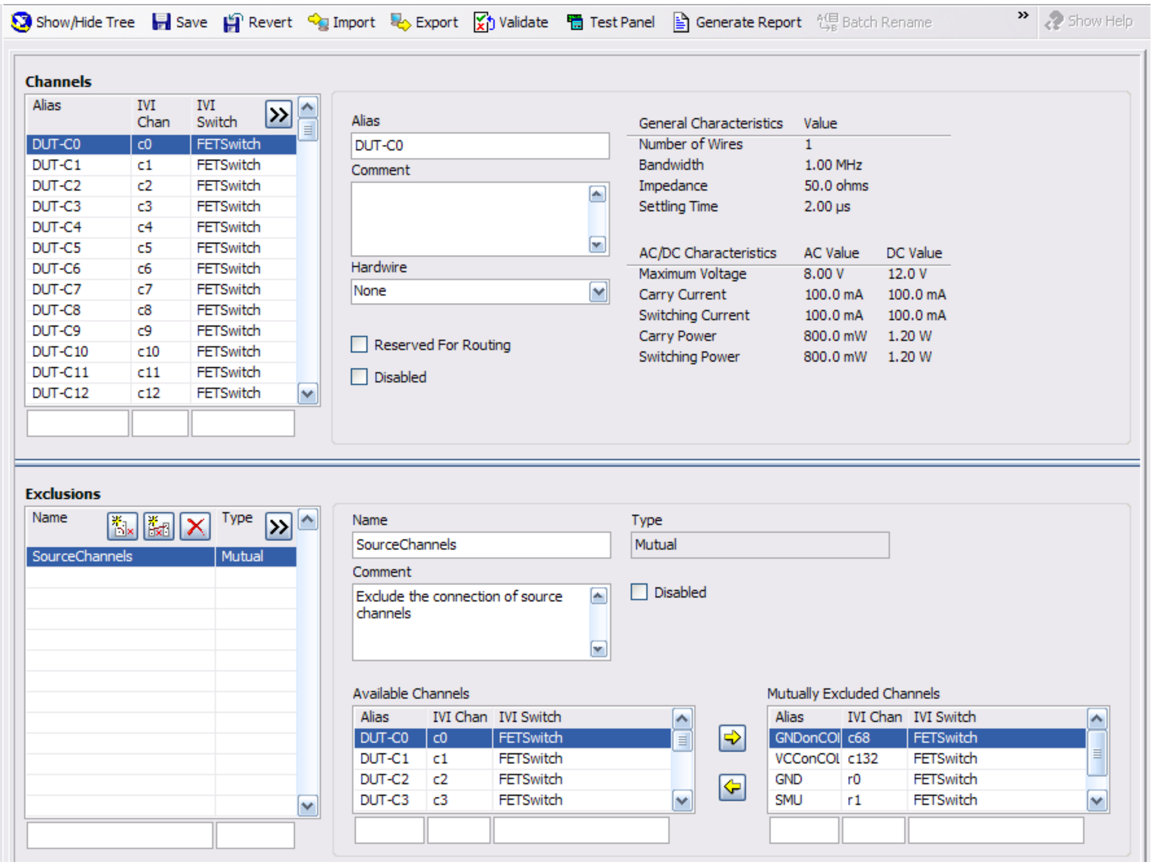
Display opens and shorts test
Results in an intuitive, graphical
manner

Figure 18: Front Panel for Open and Short Circuit Test System Created in LabVIEW

Open and Short Circuit Test Systems often require the ability to connect to hundreds of test points. This is accomplished by configuring the switch matrix in to different states. NI Switch Executive is the intelligent switch management and routing application, which renders simple this complex task. With Switch Executive, the development productivity is increased by interactively configuring and naming switch modules, external connections and signal routes. Switch Executive can also increase test code reuse and system performance with switch programming in conjunction with National Instruments TestStand, LabVIEW and Measurement Studio.

The switch hardware in the Open and Short Circuit Test System was deployed in two steps using Switch Executive. First, a Switch Executive “virtual device” was created. Next, the VI was deployed using the Switch Executive API in LabVIEW.

With NI Switch Executive alias names can be created and unique comments can be added for each channel. This capability greatly simplifies the management of hundreds or thousands of switch channels in large switch systems since the designer can refer to a channel as "SMU" or "Pin_0" instead of "c0" or "c2". Alias construction with NI Switch Executive is illustrated in Figure 19.



Channels

Alias	IVI Chan	IVI Switch
DUT-C0	c0	FETSwitch
DUT-C1	c1	FETSwitch
DUT-C2	c2	FETSwitch
DUT-C3	c3	FETSwitch
DUT-C4	c4	FETSwitch
DUT-C5	c5	FETSwitch
DUT-C6	c6	FETSwitch
DUT-C7	c7	FETSwitch
DUT-C8	c8	FETSwitch
DUT-C9	c9	FETSwitch
DUT-C10	c10	FETSwitch
DUT-C11	c11	FETSwitch
DUT-C12	c12	FETSwitch

Alias: DUT-C0
Comment:
Hardwire: None
☐ Reserved For Routing
☐ Disabled

General Characteristics	Value
Number of Wires	1
Bandwidth	1.00 MHz
Impedance	50.0 ohms
Settling Time	2.00 µs

AC/DC Characteristics	AC Value	DC Value
Maximum Voltage	8.00 V	12.0 V
Carry Current	100.0 mA	100.0 mA
Switching Current	100.0 mA	100.0 mA
Carry Power	800.0 mW	1.20 W
Switching Power	800.0 mW	1.20 W

Exclusions

Name	Type
SourceChannels	Mutual

Name: SourceChannels
Type: Mutual
Comment: Exclude the connection of source channels
☐ Disabled

Available Channels

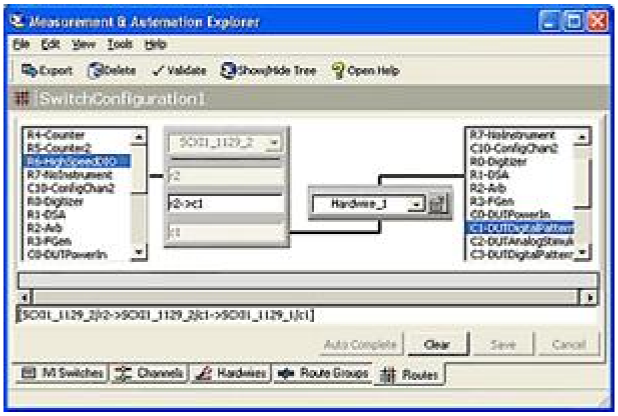
Alias	IVI Chan	IVI Switch
DUT-C0	c0	FETSwitch
DUT-C1	c1	FETSwitch
DUT-C2	c2	FETSwitch
DUT-C3	c3	FETSwitch

Mutually Excluded Channels

Alias	IVI Chan	IVI Switch
GNDonCOl	c68	FETSwitch
VCConCOl	c132	FETSwitch
GND	r0	FETSwitch
SMU	r1	FETSwitch

Figure 19: Creating Channel Aliases in Switch Executive

Once all the required channels have been configured, Switch Executive provides an interactive utility to assist the designer in connecting pairs of channels to form routes. The interface to this capability is presented in Figure 20.



[SCXI_1129_2/r2->SCXI_1129_2/c1->SCXI_1129_1/c1]

Figure 20: Switch Executive Interactive Switch Configuration

Using the Interactive Switch utility, two channels are selected to be connected from the list of alias channel names or full channel names. Then Switch Executive recommends an available route based on the previously specified channel and hardwire information. After the route has been selected, designers can name it using an alias name for quick reference in test software programs. For example, the route to connect the SMU to each pin on the DUT was given the alias 'SMU_DUT_n', where 'n' is the pin number. Similarly, the route group, which connects all pins to ground, was named 'GND to DUT'.

The ability to connect and disconnect individual routes and route groups facilitates the deployment of complex tests such as those for detecting open and short circuits. For example, to test for open and/or short circuits on pin 'n', the route group 'GND to DUT' is connected. This route group connects the ground terminal to all pins on the chip. Next, the route ‘GND_ DUT_n’ is disconnected keeping ‘SMU_DUT_n’ connected. The process of programmatically deploying the Switch Executive ‘Virtual Device’ was performed using the Switch Executive API in LabVIEW. The process is depicted in Figure 21.

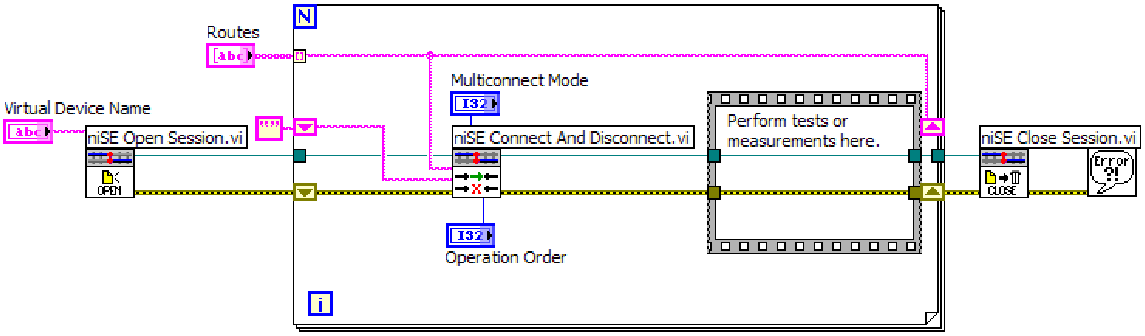


Figure 21: Deploying a Switch Executive ‘Virtual Device’ in LabVIEW

3.4 Open and Short Circuit Testing Design

Structural tests on Integrated circuits can quickly identify those, which were not built correctly. Each pin has a network of protection diodes and CMOS transistors. CMOS transistors on each input pin act like switches by allowing current to flow from V_{DD} into the DUT circuitry and from the DUT circuitry to V_{SS}. CMOS transistors can be damaged if an over-voltage condition is induced at an input or output pin. To protect these devices, two diodes are placed at each signal pin. The first sits between the signal pin and V_{DD} and the second between the signal pin and V_{SS}.

If a positive overvoltage greater than V_{DD} is applied on any pin, the V_{DD} diode becomes forward-biased and allows current to flow between the signal pin and V_{DD}. Similarly, if a negative overvoltage greater than V_{SS} is applied on any pin, the V_{SS} diode becomes forward-biased and allows current to flow between V_{SS} and the signal pin. This arrangement of protection diodes prevents damage to the CMOS transistors and DUT circuitry in overvoltage conditions. The V_{DD} and V_{SS} protection diodes must be tested for open and short conditions to ensure proper operation. An open circuit condition can occur if a protection diode is missing or is functioning incorrectly. A short circuit condition can occur if a direct connection exists in areas such as:

- Between the signal pin and $V_{DD,}$
- Between the signal pin and $V_{SS,}$
- Between one signal pin and another signal pin.

Each of these short-circuit failure modes prevent correct operation of the device. Open and Short Circuit Tests check for all the failure modes considered.

It is important to recall that CMOS integrated circuits are based on FET technology. Therefore, they use the V_{DD} and V_{SS} terminology for positive supply voltage and negative supply voltage, which is often referred to as ground. These terminals can also be documented as V_{CC} and Gnd.

3.4.1 Test Setup

The test set up for open and short circuits is separated into three phases. Phase one is concerned with testing the V_{DD} protection diode. Phase two is concerned with testing the V_{SS} protection diode. Phase three generates the test report.

3.4.2 Testing the V_{DD} Protection Diode

The test setup consists of connecting V_{SS}, V_{DD} and all other signal pins to SMU ground. In order to detect an open or short across the V_{DD} protection diode of a signal pin a minimal current, (i.e. 100 µA), is sent into the signal pin. If the V_{DD} protection diode operates correctly it will become forward-biased and the current will flow between the signal pin and V_{DD}. The V_{DD} protection diode test is depicted in Figure 22.

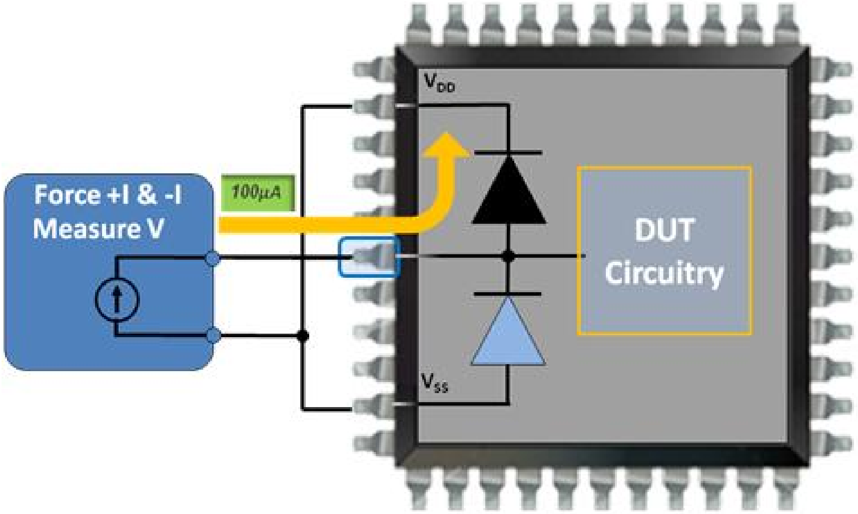


Figure 22: Testing the V_{DD} Diode

Measuring the voltage drop across the forward-biased V_{DD} diode can determine if it is working correctly. If the voltage measured between the signal pin and ground is close to 0 V, which is ground, then one or more short-circuits exist between the signal pin and ground through V_{SS}, V_{DD}, and/or another signal pin. If the voltage measured between the signal pin climbs to a potential, which is higher than an acceptable forward-biased voltage drop, then there is an open circuit between the signal pin and ground. If the measured voltage is an acceptable forward-biased voltage drop the V_{DD} protection diode is operating correctly. Table 3 presents an example of V_{DD} protection diode test results and the resulting pass/fail specifications.

Table 3: V_{DD} Protection Diode Test Specifications

Voltage Reading at Signal Pin	**Test Result**
Less than +0.2 V	Fail: Shorted
In between +0.2 V and +1.5 V	Pass
Greater than +1.5 V	Fail: Open

The voltage between the signals pin and ground should be close to 0 V if the diode is shorted and the test result should indicate Fail: Shorted. However, if the other signal pins are not all grounded and the diode is shorted current would still flow through the forward-biased V_{DD} protection diode, as depicted in Figure 23, and the test result would be pass. No current, other than a small amount of leakage current, should flow through the V_{SS} protection diode during this test since it will be reverse-biased. Also, the acceptable forward-bias voltage drop is typically dependent upon the material from which the semiconductor diode is fabricated. However, manufacturing techniques may also be used to lower the forward-biased voltage drop. The forward-biased voltage drop of a silicon diode is generally accepted to be 0.65 V. The exact voltage drop is dependent on the magnitude of the current flowing through the diode's p-n junction, the temperature of the junction and several physical constants. The relationship between the forward-biased voltage drop, the applied current and the associated variables is commonly known as the diode equation, which is given by:

$$I_D = I_S \times (e^{V_D/(N \times V_t)} - 1)$$

The variables in the diode equation are:

- I_D = Diode current, (mA),
- I_S = Saturation current, (mA),
- V_D = Voltage drop across the diode, (V),
- N = Ideality coefficient, between 1 and 2,
- V_t = Thermal voltage (V), approximately 25.85 mV at room temperature.

3.4.3 Testing the V_{SS} Protection Diode

The process for testing the V_{SS} diode is the same as that for testing the V_{DD} diode. All pins including V_{SS} and V_{DD} are connected to SMU ground. However, for the V_{SS} diode a negative current of the same value, (i.e. -100 µA), is sent into the signal pin. If the V_{SS} protection diode operates correctly, it will become forward-biased and current will flow between V_{SS} and the signal pin. The V_{SS} protection diode test is depicted in Figure 23.

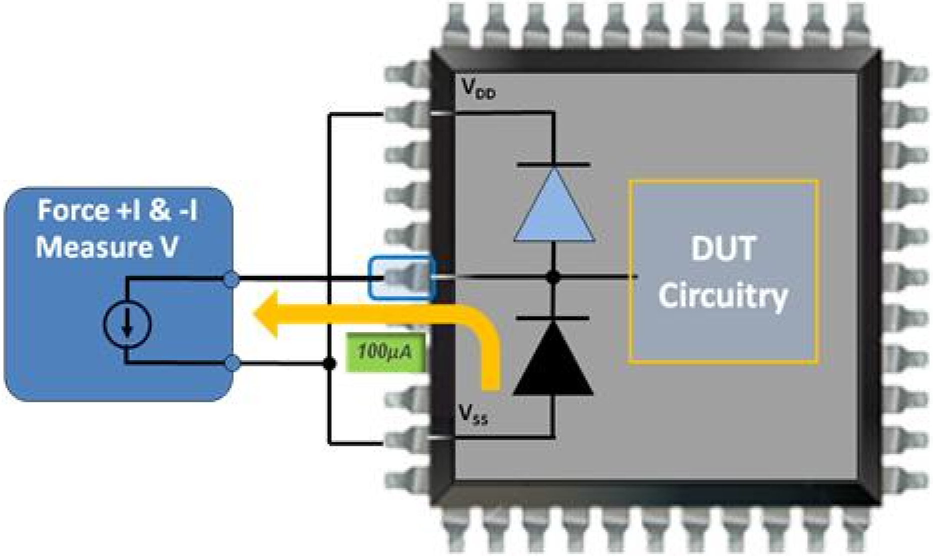


Figure 23: Testing the V_{SS} Diode

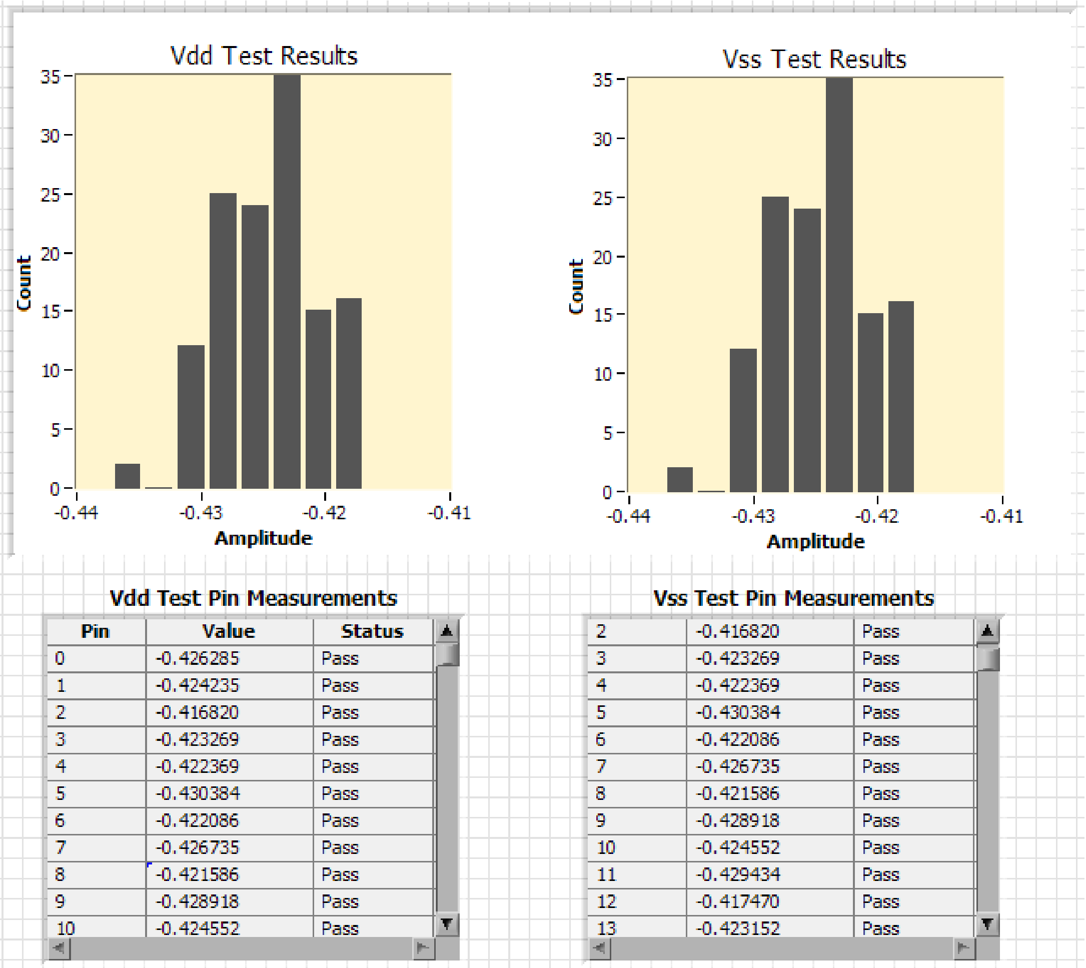
No current other than a small amount of leakage current should flow through the V_{DD} protection diode since it will be reverse-biased. Like the V_{DD} diode case, measuring the voltage drop across the forward-biased V_{SS} diode can determine if it is functioning properly. Table 4 presents the test parameters for the V_{SS} protection diode.

Table 4: V_{SS} Diode Test Specifications

Voltage Reading at Signal Pin	**Test Result**
Greater than -0.2 V	Fail: Shorted
In between -0.2 V and -1.5 V	Pass
Less than -1.5 V	Fail: Open

3.4.4 Generation of the Test Report

The report section collects all the values measured during the V_{DD} and V_{SS} tests and places them together in a predefined format. Report layout includes customers' name, purchase order number, part number, manufacturer and job number. The last part of the report includes a list of all the equipment used during the test. Figure 24 presents the VI front panel for the test report. The graphical displays present the measured data as bar graphs and the tables tabulate the corresponding voltages measured at each pin. A sample test report, generated during this research, is presented in appendix A.



Vdd Test Pin Measurements

Pin	Value	Status
0	-0.426285	Pass
1	-0.424235	Pass
2	-0.416820	Pass
3	-0.423269	Pass
4	-0.422369	Pass
5	-0.430384	Pass
6	-0.422086	Pass
7	-0.426735	Pass
8	-0.421586	Pass
9	-0.428918	Pass
10	-0.424552	Pass

Vss Test Pin Measurements

2	-0.416820	Pass
3	-0.423269	Pass
4	-0.422369	Pass
5	-0.430384	Pass
6	-0.422086	Pass
7	-0.426735	Pass
8	-0.421586	Pass
9	-0.428918	Pass
10	-0.424552	Pass
11	-0.429434	Pass
12	-0.417470	Pass
13	-0.423152	Pass

Figure 24: VI Front Panel for the Test Report

3.5 Automated Test Setup

An external switching system front end and programmable source measure unit can be utilized to automate the V_{DD} and V_{SS} protection diode testing. The switching system can scan through pre-configured states and create the required current and ground paths to the V_{DD}, V_{SS} and signal pins of the semiconductor device. The source measure unit can send the required currents and measure the resulting voltages from each signal pin to ground. The setup for automated testing is presented in Figure 25.

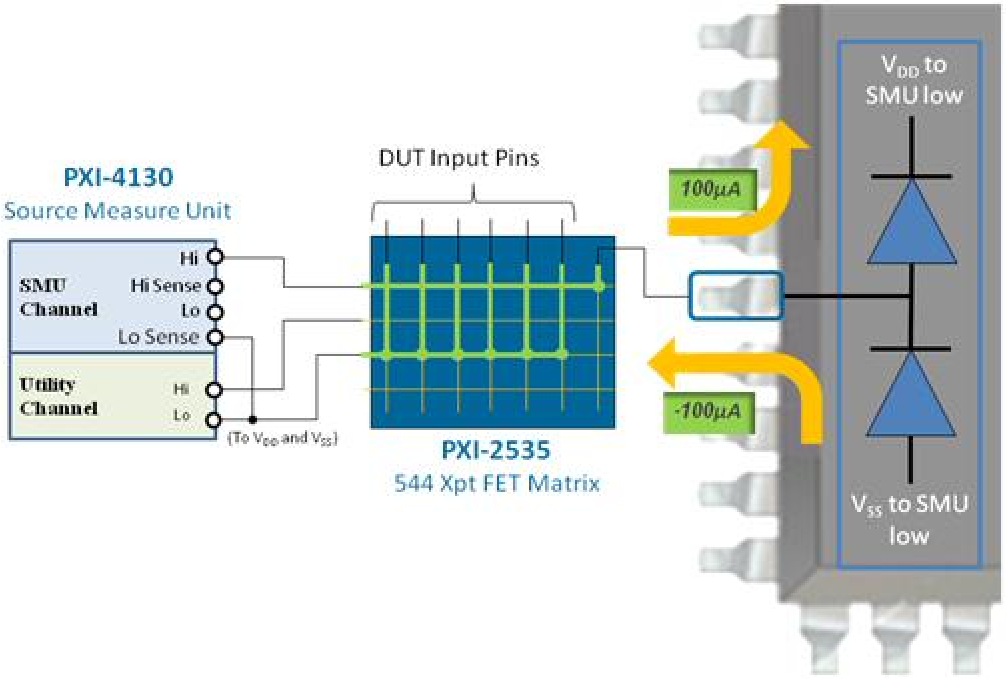


Figure 25: Automated Open and Short Circuit Test Setup

In order to connect the SMU to the DUT through the FET switch, a matrix topology was used with pins from the SMU connected to rows in the matrix and pins from the chip connected to columns. These connections are illustrated in Figure 26.

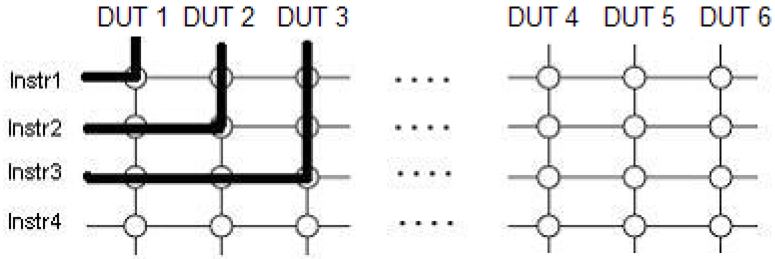


Figure 26: Connecting the SMU and DUT

Grounding all pins on the DUT was accomplished by closing all the connections on the matrix that route the ground of the PXI-4130 SMU to the pins on the DUT. Connections from the PXI-4130 SMU Low pin to V_{DD} and V_{SS} were established directly through a cable instead of through the switch since the V_{DD} and V_{SS} pins are always connected to the SMU Low pin. The matrix switch was used for physically connecting all signal pins initially to ground and then each pin was sequentially tested using the SMU measurement channel.

It was important to connect V_{SS} and V_{DD} to ground. In addition, all other signal pins were connected to ground before testing the protection diodes could proceed. Grounding all the signal pins ensured that short-circuits were properly detected. The need to ground all pins is illustrated in Figure 27. When a short-circuit is detected between two signal pins, the voltage between the pin under test and SMU low should be close to zero volts as presented in Tables 3 and 4. If the other signal pins were not all grounded, current would still flow through the forward-biased V_{DD} protection diode and the test result would be Pass instead of Fail. This situation is depicted in Figure 27

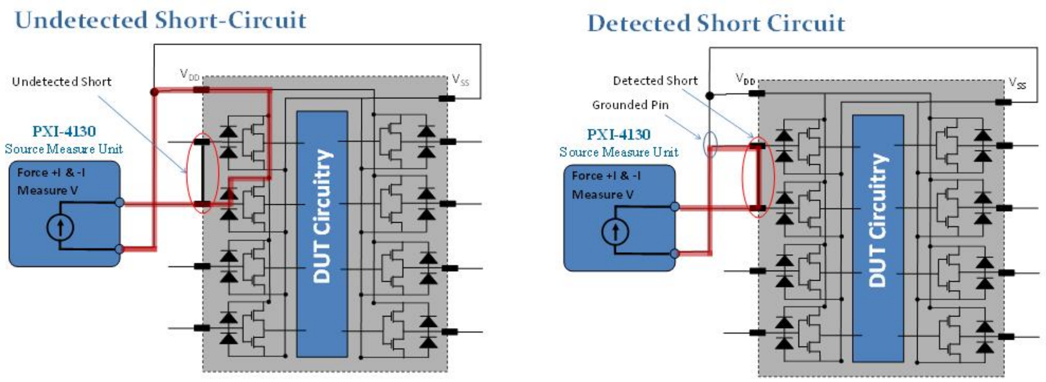


Figure 27: Grounding Pins is Essential for Detecting Short Circuit

In order to limit the voltage produced during open circuit conditions, an upper limit voltage clamp was set on the SMU in order to prevent permanent damage to the

DUT. The voltage clamp level on the PXI-4130 is software programmable to 3 Volts as per most CMOS IC specifications.

The SMU forces ±100 µA of current into the diode and measures the resulting voltage. For this test, a voltage of approximately ±0.65 V is expected, which is the voltage drop across a forward-biased silicon diode. The voltage resulting from the current was measured and then compared to the test specification tables in order to complete the failure analysis test.

The software for the Open and Short Circuit Test System developed during this research was created using NI LabVIEW and NI Switch Executive. LabVIEW was used as the primary Application Development Environment while Switch Executive was used to configure routes on the high-density matrix.

The Open and Short Circuit Tests were separated into three test procedures:

- Testing the V_{DD} protection diode
- Testing the V_{SS} protection diode
- Generating the test report.

The first two procedures were performed using the same hardwire connections. Depending on the procedure, the forcing current can be programmed to change in direction. In addition the forced current can also change its step resolution at the SMU. Due to the similarities, between procedures ‘a’ and ‘b’, only the procedure for testing the V_{SS} protection diode will be described. The procedure for testing the V_{SS} protection diodes is outlined in Figure 28.

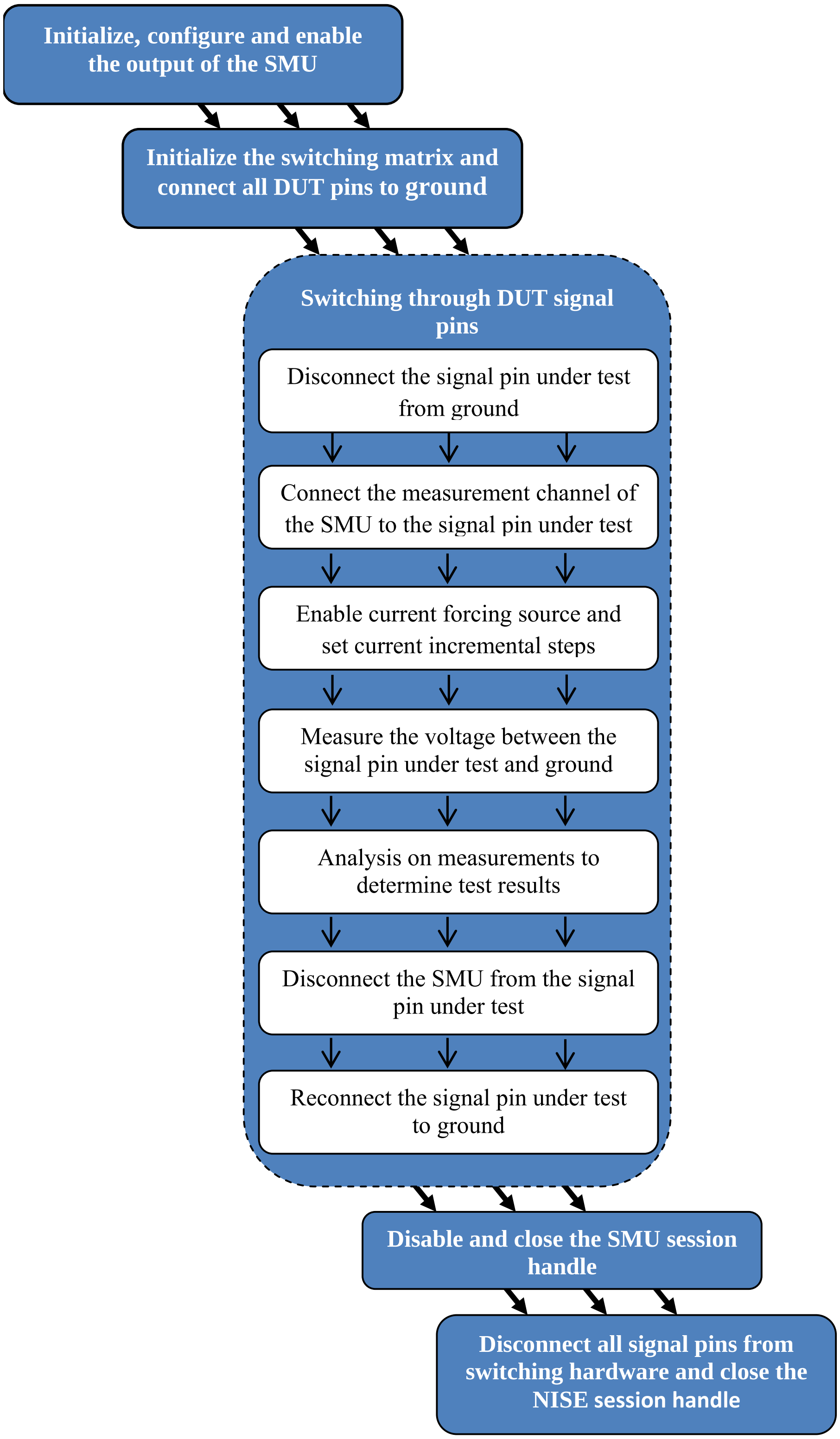


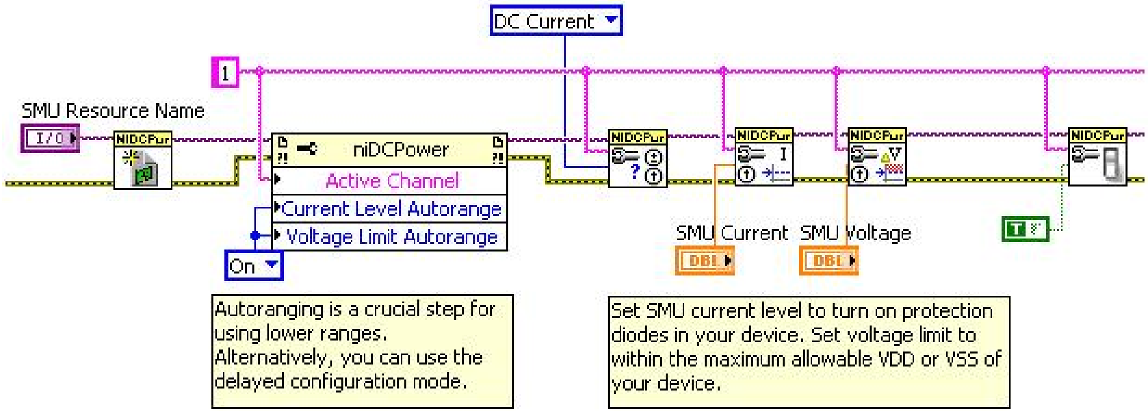
Figure 28: VSS Protection Diode Testing Procedure

A detailed description of the steps of the test procedure outlined in Figure 28 is presented including their respective LabVIEW block diagrams.

3.5.1 Initialize the SMU, Configure and Enable the Output of the SMU

The SMUs' resource name is provided to the Initialize VI to initialize the SMU and to provide a SMU session handle. The SMU session handle is passed to all subsequent NI-DCPower VIs. Next, channel 1 of the SMU is configured for the DC Current, which it will be sourcing during conduct of the test. The NI-DCPower configuration VIs can be wired in any order as long as all the parameters are set before the SMU output is enabled. The required parameters include the current level, the current level range, the voltage limit and the voltage limit range.

The step following the Initialize VI is the NI-DCPower property node. This step instructs the SMU to automatically determine and set current level and voltage limit ranges based on the current level and voltage limit inputs. The VI following the property node confirms that the SMU is to be used in DC Current mode. Next, the current level is set to -100 µA. In order to test the V_{SS} protection diode, channel 1 is configured such that the current flows into the SMU, the V_{SS} protection diode is forward-biased and the voltage limit set to 3 V, (equivalent to ±3 V). A "true" Boolean constant is used to enable the SMU output. A block diagram for this procedure step is presented in Figure 29.



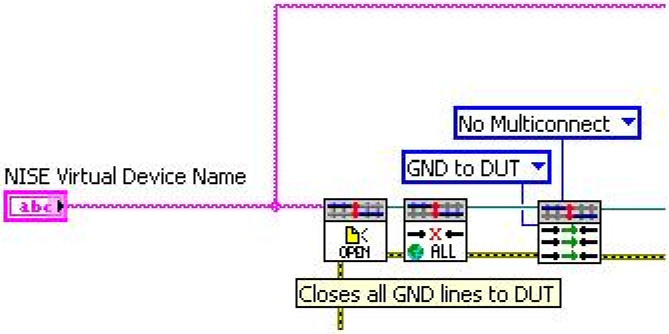
Autoranging is a crucial step for using lower ranges. Alternatively, you can use the delayed configuration mode.

Set SMU current level to turn on protection diodes in your device. Set voltage limit to within the maximum allowable VDD or VSS of your device.

Figure 29: Initializing, Configuring, and Enabling the SMU

3.5.2 Initialize the Switching Hardware and Connect DUT Pins to Ground

This procedure step initializes the switching hardware and sets it to a state where V_{DD}, V_{SS} and all the DUT's signal pins are connected to ground. There are multiple ways to control switches using LabVIEW. However, the best way to program switching hardware, as a system, is with the NI Switch Executive API. Figure 30 presents an image of the block diagram code for this procedure step.



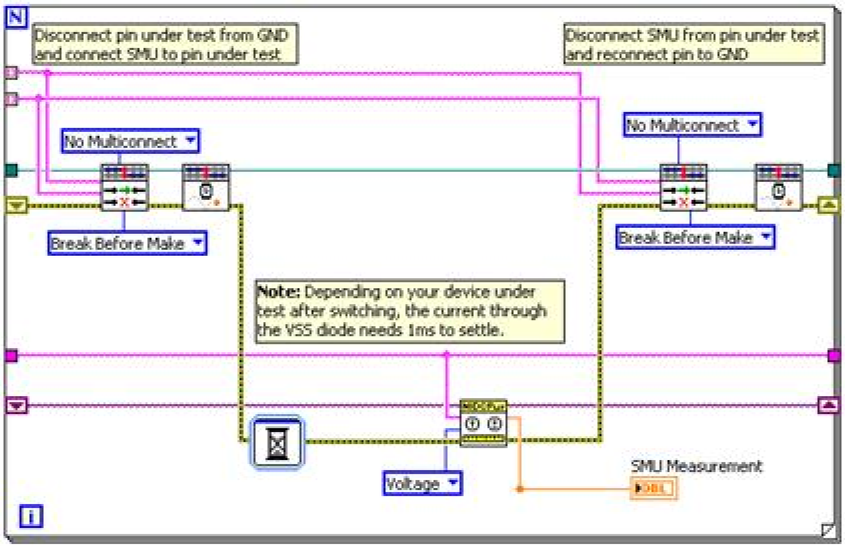
Closes all GND lines to DUT

Figure 30: Initialize the Switching Module and Connect the DUT Pins to Ground

The NISE Virtual Device Name is placed in the input of the Open Session VI for opening the session, which handles all the switches in the system. NI Switch Executive stores this session in one NISE session handle, which is passed to all subsequent NISE VIs. The 'Disconnect All' VI disconnects all connections on every switch device managed by the NI Switch Executive session. This action sets the switch system configuration in a known state where no switch routes are connected. Then all routes in the GND to DUT route group are connected, which results in the switching hardware being set to a state where V_{DD}, V_{SS} and all the DUT's signal pins are connected to ground.

3.5.3 Switching Through the Signal Pins

Switching through the signal pins is implemented using a 'For Loop' structure. The NI Switch Executive VIs were used within this structure to disconnect the signal pin under test from ground before connecting it to channel 1 of the SMU. Then, the 'NI-DCPower VI' was used to measure the voltage from each DUT signal pin to ground. Before the next iteration, the signal pin was disconnected from channel 1 and connected back to ground. These activities are depicted in Figure 31.


Disconnect pin under test from GND and connect SMU to pin under test

Disconnect SMU from pin under test and reconnect pin to GND

Note: Depending on your device under test after switching, the current through the VSS diode needs 1ms to settle.

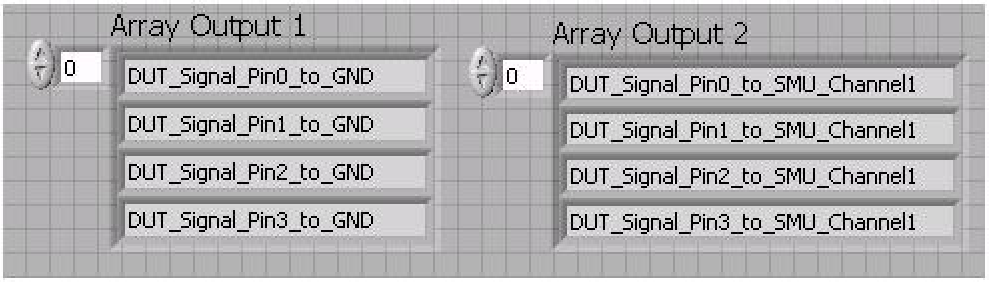
Figure 31: Disconnect GND, Connect SMU, Measure, Disconnect SMU, and Reconnect GND

The process of accessing DUT signal pins was performed using route groups, which were created in the NI Switch Executive. Route Groups were used to put the switch matrix into a desired state. The first route group contained routes connecting the DUT signal pins to ground. The second route group contained routes connecting the DUT signal pins to channel 1 of the SMU. Table 5 presents the makeup of the route groups.

Table 5: Set up Route Groups to Simplify Indexing the Correct DUT Signal Pin

	Route group 1	**Route group 2**
Index 0	DUT_Signal_Pin0_to_GND	DUT_Signal_Pin0_to_SMU_Channel1
Index 1	DUT_Signal_Pin1_to_GND	DUT_Signal_Pin1_to_SMU_Channel1
Index 2	DUT_Signal_Pin2_to_GND	DUT_Signal_Pin2_to_SMU_Channel1
Index 3	DUT_Signal_Pin3_to_GND	DUT_Signal_Pin3_to_SMU_Channel1

The NI Switch Executive configuration API in LabVIEW provides full programmatic access to all NI Switch Executive features. It was used to extract individual route names from all route groups. The result was two string arrays comprised of the routes from Route Group 1 and Route Group 2. The arrays are listed in Figure 32.



DUT_Signal_Pin0_to_GND
DUT_Signal_Pin1_to_GND
DUT_Signal_Pin2_to_GND
DUT_Signal_Pin3_to_GND

DUT_Signal_Pin0_to_SMU_Channel1
DUT_Signal_Pin1_to_SMU_Channel1
DUT_Signal_Pin2_to_SMU_Channel1
DUT_Signal_Pin3_to_SMU_Channel1

Figure 32: Array Output Using NI Switch Executive Configuration API

Passing these arrays into a ‘For Loop’ automatically indexed the arrays. For example, on the first iteration of the loop, the indexed routes to be connected and disconnected inside the loop were DUT_Signal_Pin0_to_GND and DUT_Signal_Pin0_to_SMU_Channel1. On the second iteration of the loop, the indexed routes were DUT_Signal_Pin1_to_GND and DUT_Signal_Pin1_to_SMU_Channel1. This iteration continued until the arrays indexed each route.

Based on the pass and fail criteria of the DUT, the voltage measurement indicates if the V_{SS} protection diode on the signal pin under test passed, failed open, or failed due to a short-circuit. The determination of the test results is illustrated in Figure 33.

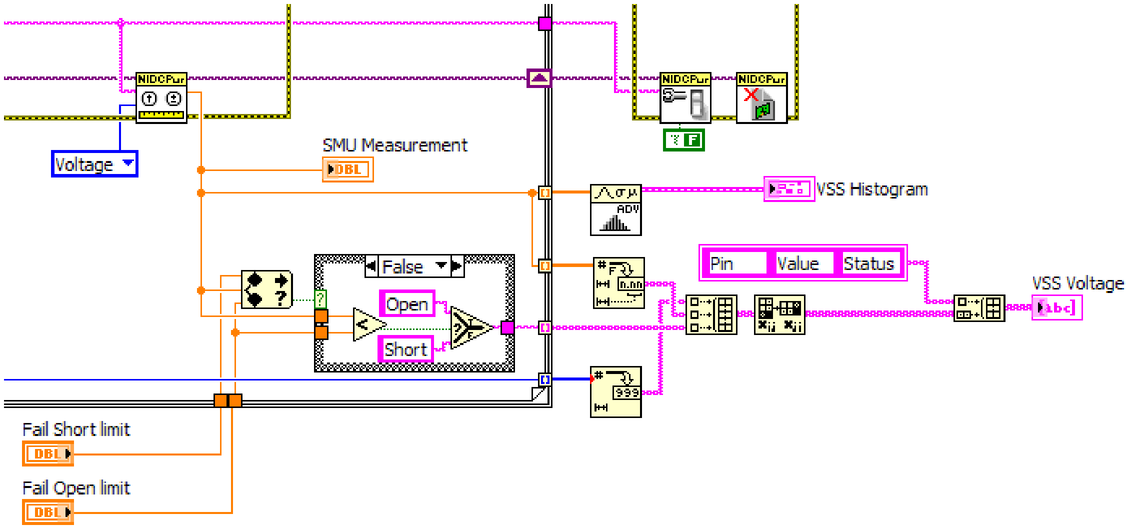


Figure 33: Determine Test Result Based on the Voltage Measurement

3.5.4 Disable and Close the SMU Session Handle

The NI-DCPower VIs were used to disable the output of the SMU and close the SMU session handle. Figure 34 displays an image of the block diagram code for this procedure step.

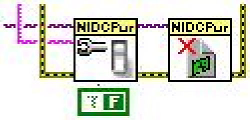


Figure 34: Disable the SMU Output and
Close the SMU Session Handle

A Boolean ‘False’ constant was connected to the ‘Configure Output Enabled VI’, which disabled the output of the SMU. Therefore, the current flow of -100 µA was turned-off. The ‘Close VI’ closed the SMU session handle and reallocated the SMU resources to default values.

3.5.5 Disconnect All Signal Pins from the Switching Hardware and Close the NISE Session Handle

NI Switch Executive VIs were used to disconnect V_{DD}, V_{SS} and the signal pins from the switching hardware and close the NISE session handle. Figure 35 displays an image of the block diagram code.

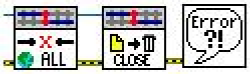


Figure 35: Disconnect all Signal Pins and
Close the NISE Session Handle

The 'Disconnect All VI' disconnects all connections on every switch device managed by the NI Switch Executive session. This action sets the switch system configuration to a known state where no switch routes are connected. The 'Close Session VI' closes session handles to all the switches in the system. Although not required, it is often helpful to include an error handler. In case an error occurs, the VI returns a description of the error and optionally displays a dialog box with the error information.

3.5.6 Testing the VDD Protection Diode

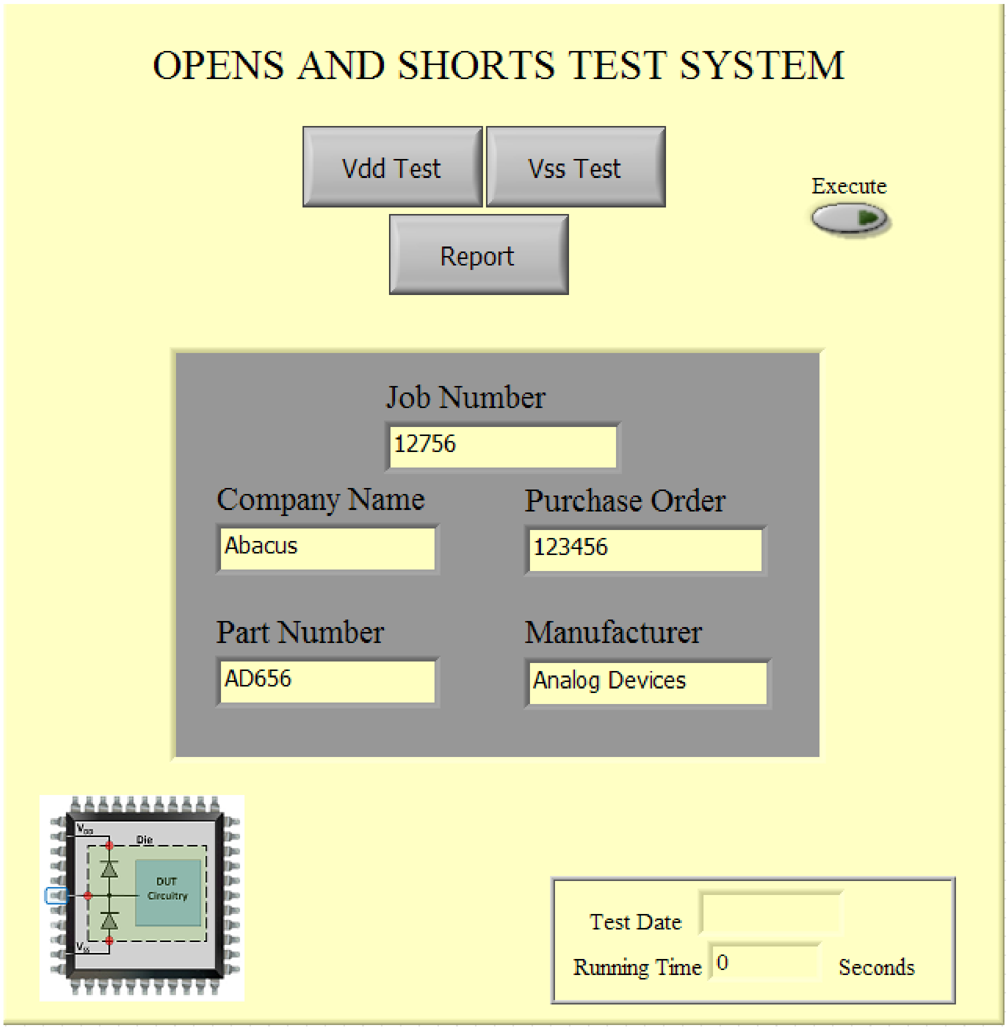
To modify the V_{SS} protection diode test, the SMU must first be disabled. Then the current polarity must be changed from -100 µA to +100 µA on channel before re-enabling the SMU. The 'For Loop' structure is used to perform the same test sequences to generate the test V_{DD} protection diode test report.

3.5.7 Combining the V_{SS} and V_{DD} Protection Diode Tests and Report Generation Procedure

Three independent VIs were created. One VI was used for testing the V_{DD} protection diode. A second VI was used for testing the V_{SS} protection diode. A third VI was used for generating the test report. A fourth VI was created to combine the three VIs in sequence. The fourth VI possessed the capability of performing all the functionality necessary to comply with all requirements. After typing in test identification information, the EXECUTE button was pressed to start the test sequence. V_{DD} Test, V_{SS} Test and Report indicators change their color from gray to red indicating, which procedure is running. When the sequence of the program arrives at the report generation phase, a 'File Save as' window appears where the name of the report file is typed and the file location path is selected. The report is generated in PDF file format.

After test completion, the running time is displayed in the bottom right corner of the front panel. The run time is for reference purposes only and could be used by a test engineer to estimate total test time for a large number of DUTs.

The front panel of the fourth VI is presented in Figure 36.



Job Number: 12756

Company Name: Abacus

Purchase Order: 123456

Part Number: AD656

Manufacturer: Analog Devices

Test Date:

Running Time: 0 Seconds

Figure 36: VI Front Panel for the Open and Short Circuit Test System

The block diagram of the fourth VI is presented in Figure 37.

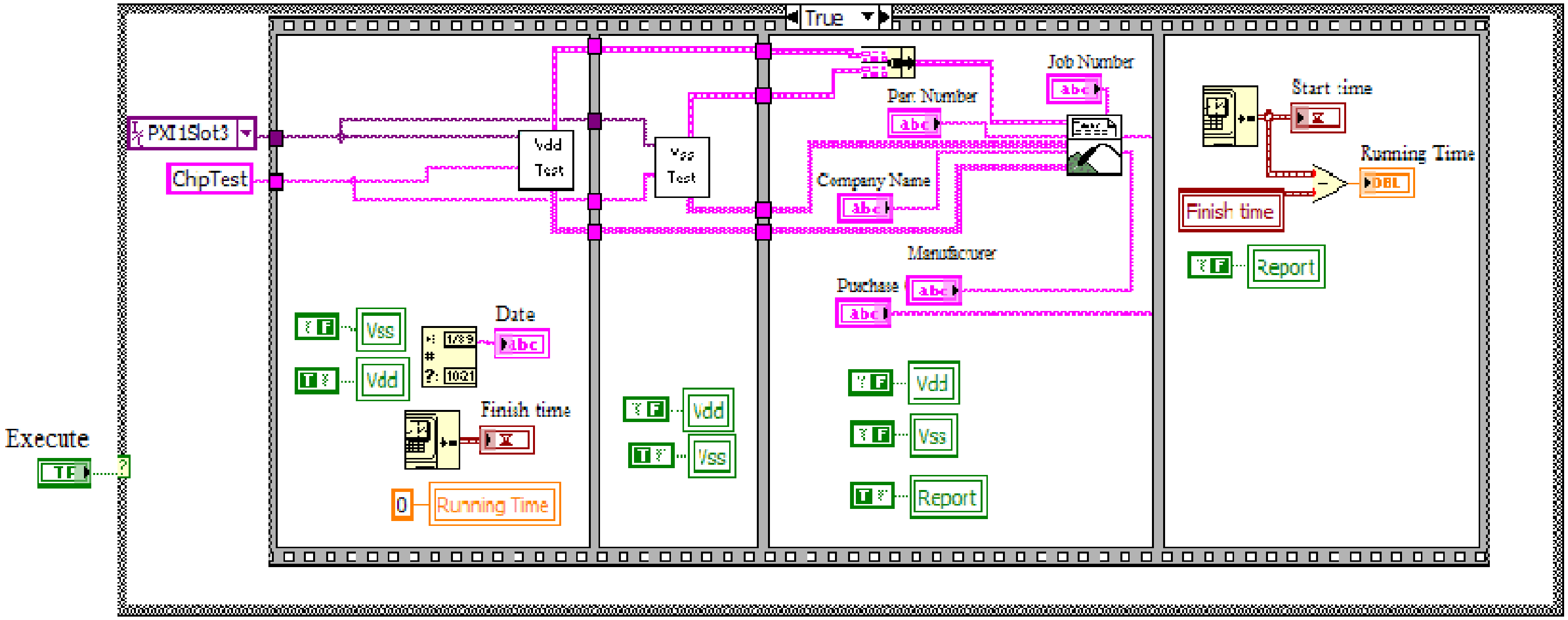


Figure 37: VI Block Diagram for the Open and
Short Circuit Tests System

Chapter 4

Testing and Verification

4.1 Results

Using Interchangeable Virtual Instruments, (IVI), drivers for the PXI 4130 SMU and the PXI 2530 switch matrix, it was possible to emulate the behavior of the system before wiring the DUTs to the modules and to terminal blocks. Simulation of these two instruments allowed the switch group routes and the SMU terminals to be validated. This simulation guaranteed that there were no short circuits between the routed signals. Also, it was important to recall the NI Switch Executive software to validate routes according to the conditionals set by the designer.

In order to run the emulation, a small change of the V_{DD} and V_{SS} test VIs was required. Recall that the 'NI-DCPower Measure' function measures the voltage on each pin. Then the value is compared within the Fail Open and Fail Short limits specified in the front panel. Figure 38 presents a section of the original block diagram of the 'NI-DCPower Measure' function. Given that the instruments are driver simulated, the voltages values need to be generated. This can be accomplished by replacing the 'NI-DCPower Measure' function by the 'Random Number' function. The 'Random Number' function is illustrated in Figure 39. The 'Random Number' function generates numbers randomly between 0.0 and 1.0. Since the open and short circuit test limits are between 0.2 and 1.5, respectively, a scale factor of +/- 1.7 was incorporated.

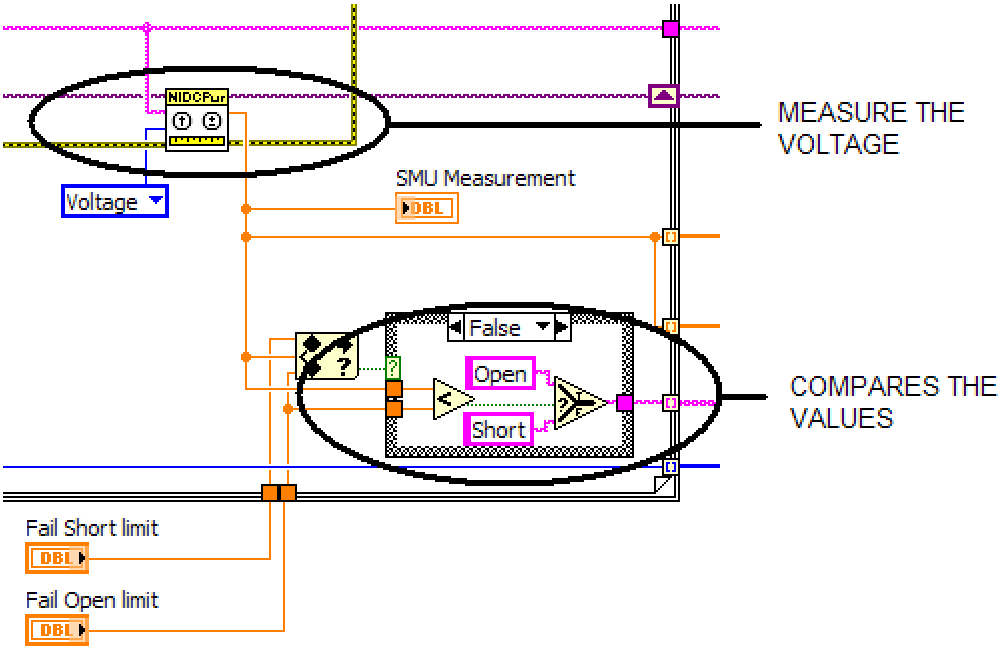


Figure 38: NI-DCPower Measure Function

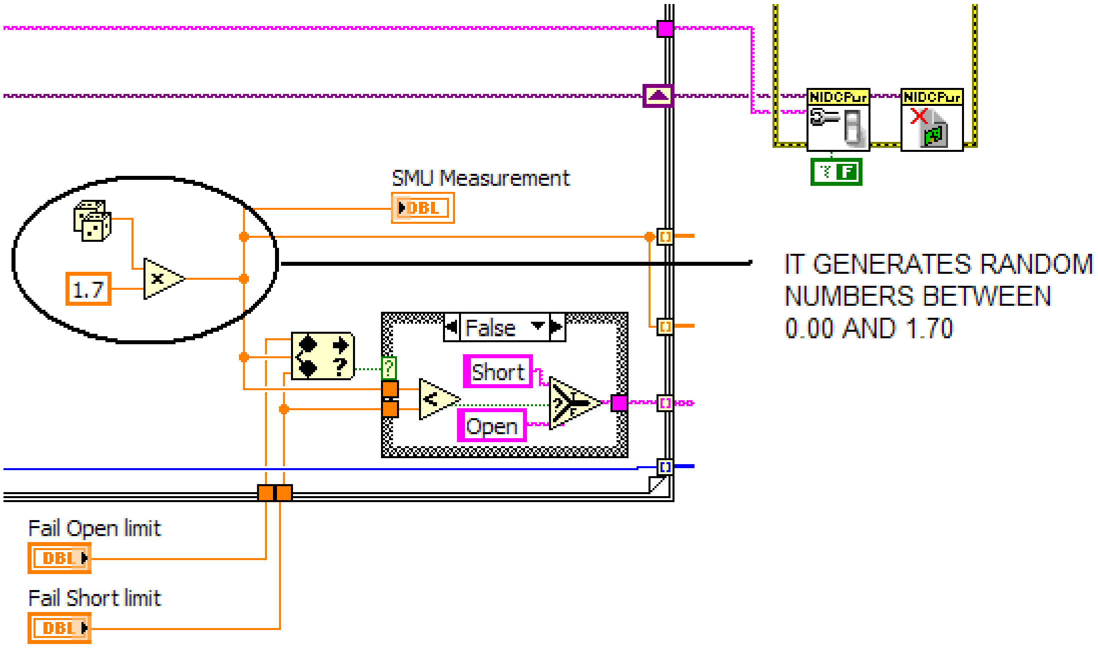


Figure 39: Generation of Random Voltage Values

After completing the changes to the original VI, mentioned previously, a series of Open and Short Circuit Tests were performed and recorded different execution times. The average time obtained was 14.649 seconds for eleven test runs. Figure 40 graphically presents the different values achieved during each execution.

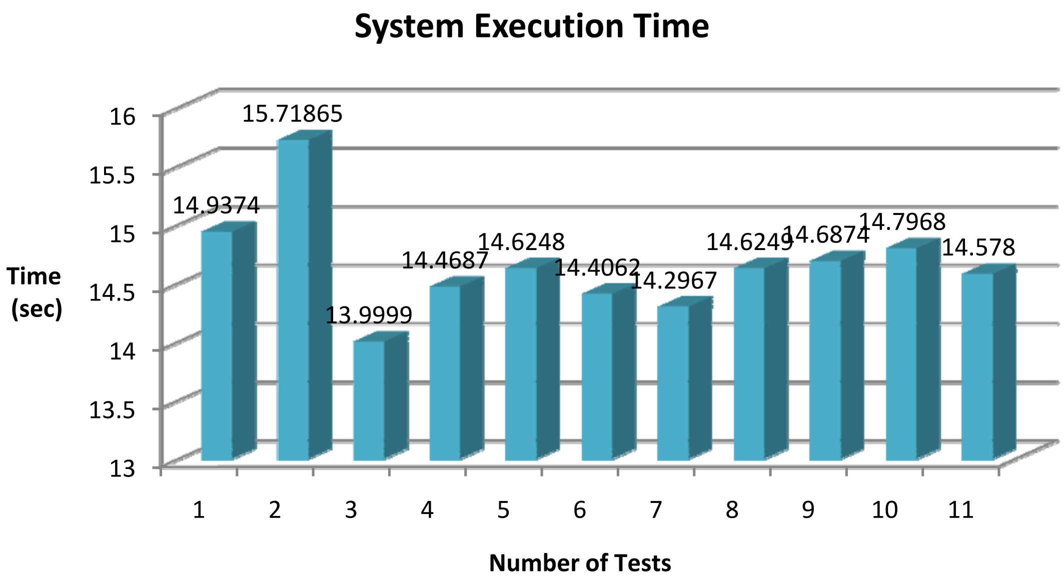


Figure 40: System Execution Time Results

Due to the modularity of the system design, V_{DD} Test and V_{SS} Test could be run separately. Therefore, execution times for these two VIs were obtained independently for analysis. The V_{DD} test execution time results are presented graphically in Figure 41. The V_{SS} test execution time results are presented graphically in Figure 42.

The average times for the V_{DD} and V_{SS} protection diode tests were 3.655 seconds and 3.138 seconds, respectively. The same numbers of test runs were executed, which yielded approximately 0.0265 seconds per pin. The time required for a test engineer to manually test each pin using a non-automated test system is approximately 1.48 seconds per pin. The advantage of a software-defined instrumentation based test system is clear. It is important to mention that the1.48 seconds per pin test time does not include the time for data logging the results and making the test report.

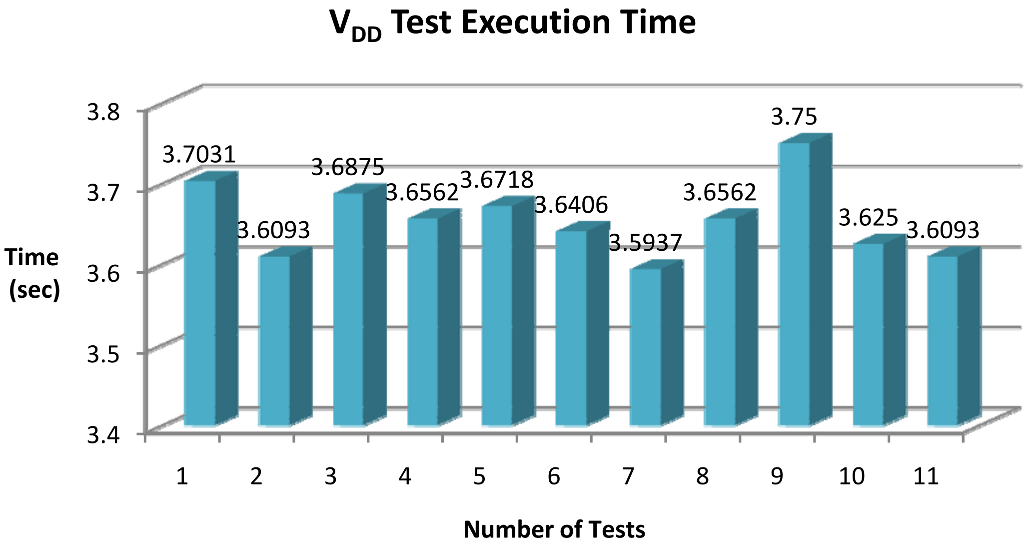


Figure 41: V_{DD} Test Execution Time Results

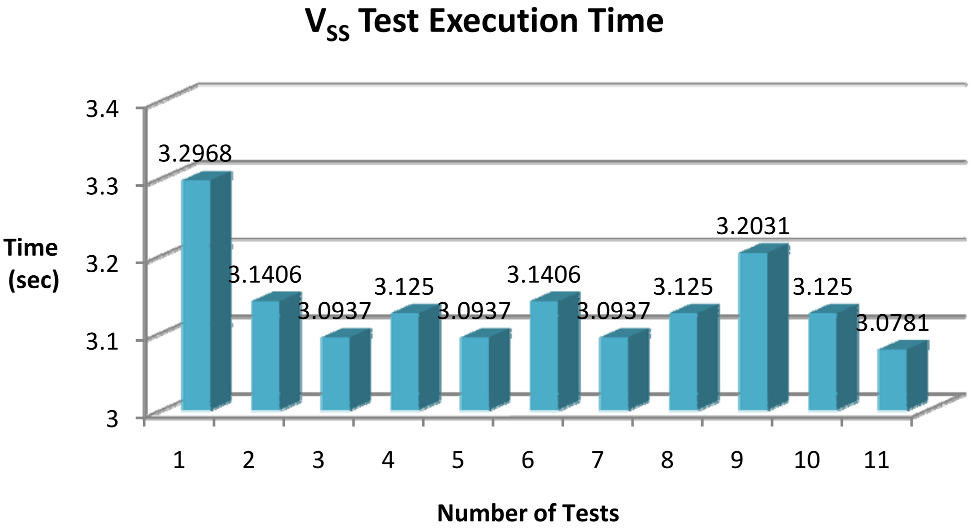


Figure 42: V_{SS} Test Execution Time Results

Table 6 presents the data for three different technicians on three different parts test using a curve tracer and manually testing pin by pin. A comparison of the times listed in the table and the average time, (14.694 seconds), required by the Open and Short Circuit Test System clearly indicates the superior approach. The benefit of using software-defined instrumentation test systems are clearly superior to manual testing.

Table 6: Test Times Using the Non-Automated Test System

Part Number	**Description**	**Pins**	**Package**	**Time (sec)**
QL5130	Programmable Logic Device	144	TQFP	**188**
ICS9148B04	Frequency Generator & Integrated Buffers	48	SSOP	**90**
UDP70325	Microcontroller	94	QFP	**132**

Chapter 5

Conclusions and Future Work

5.1 Conclusions

A Software-Defined Integrated Circuit Test System was successfully developed for testing the CMOS ICs protection diode structure. Protection diode testing is an important part of a series of tests performed for Failure Analysis, which help to detect damages and identify counterfeit ICs.

Software-defined instrumentation is a versatile approach for different applications and is the primary method currently employed in automated test systems. PXI is the leading platform for developing many software-defined instruments for multidisciplinary applications. Due to the accessibility of diverse modules for this platform and the proposed system design approach, the design of test systems with complex systems requirements and functionality is becoming straightforward.

The system engineering approach was used for designing the CMOS ICs protection diode test system structure. All the defined requirements were verified through the emulation process of the PXI modules. The feature test system was divided into three small subsystems. The subsystems consisted of V_{DD} diode test, V_{SS} diode test and report generation. The test system was designed using a modular approach, which was implemented in three subsystems. This approach made the design flow easy to implement and verify.

5.2 Future Work

Several functionalities were identified as possible future development areas for improving the test system. Improvement areas identified were found to be:

- Design a functional test for digital or mixed signal devices using the DIO module available in the PXI platform,
- Different test sequences should be used for testing large number of parts,
- Develop test modules for RF testing for communication applications,
- Expand the capability of performing testing remotely.

References

[1] www.altera.com/end-markets/test-measurements/test-products/ate/tes-ate.html

[2] Athan, Stephan, “IDDQ Testability of Deep Submicron CMOS Integrated Circuits”, University of South Florida. December, 1995

[3] en.wikipedia.org/wiki/main_page

[4] www.incose.org

[5] Mehendale, Vikram, “System Approach to Embedded System Design”, University of South Florida. April, 2007

[6] National Instruments, “Designing Next Generation Test Systems; an In-Depth Developers Guide”

[7] PCI eXtensions for Instrumentation, Revision 2.2, PXI Systems Alliance, August 22, 2005

[8] Pickering Interfaces, “PXImate”, Fourth Edition, 2005

[9] Bisking, Kevin, “Software-Defined Instrumentation”, Design Magazine, August, 2008

[10] Humphrey, Rick, “Addressing Modern Mil-Aero RF Test Challenges with Modular Software Defined Test Systems”, COTS Journal, January, 2005

[11] www.ni.com

[12] Sally J. F. Baron, Ph.D, “Change-out: A System of Systems Approach to COTS Management”, Sixth International IEEE Conference on Commercial-off-the-Shelf, (COTS), IEEE, 2007

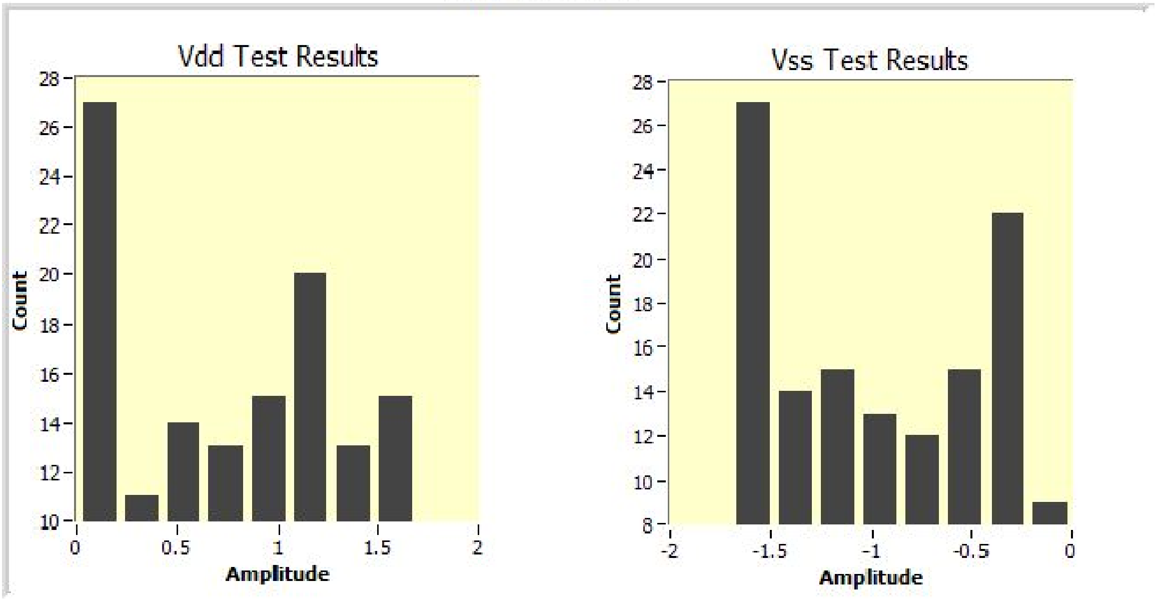
Appendices

Appendix A: Open and Short Circuit Test System Report

Open and Short Test Report for University of South Florida

COMPANY:	Abacus	PURCHASE ORDER:	123456
JOB NUMBER:	12756		
PART NUMBER:	AD656	MANUFACTURER:	Analog Devices

Graph Results



	Vdd Measurement		Vss Mesuarement	
Pin	Value	Status	Value	Status
0	0.187126	Short	-1.119785	Pass
1	1.524439	Open	-0.431643	Pass
2	0.577031	Pass	-0.607497	Pass
3	0.805448	Pass	-0.860666	Pass
4	0.432947	Pass	-0.449302	Pass
5	0.102208	Short	-0.736958	Pass
6	0.945887	Pass	-0.263317	Pass
7	1.435655	Pass	-1.659638	Open
8	1.682729	Open	-1.614601	Open

Appendix A (Continued)

9	1.166671	Pass	-1.651084	Open
10	0.647581	Pass	-0.340525	Pass
11	1.677161	Open	-1.533502	Open
12	0.122543	Short	-1.385024	Pass
13	0.357408	Pass	-0.394095	Pass
14	1.182493	Pass	-0.258710	Pass
15	1.343115	Pass	-0.382751	Pass
16	0.601738	Pass	-1.542001	Open
17	1.192540	Pass	-0.548545	Pass
18	1.112865	Pass	-1.523647	Open
19	1.649690	Open	-1.569187	Open
20	1.142805	Pass	-1.414294	Pass
21	0.432354	Pass	-0.922331	Pass
22	0.791020	Pass	-1.258069	Pass
23	0.382837	Pass	-1.124609	Pass
24	0.861317	Pass	-0.675634	Pass
25	0.180155	Short	-1.078821	Pass
26	1.235588	Pass	-0.439198	Pass
27	1.354221	Pass	-1.427448	Pass
28	1.129196	Pass	-0.888540	Pass
29	1.248231	Pass	-1.606556	Open
30	0.466716	Pass	-0.700830	Pass
31	0.047458	Short	-1.695703	Open
32	0.162436	Short	-1.671912	Open
33	0.980140	Pass	-1.642306	Open
34	0.132179	Short	-1.633050	Open
35	0.146708	Short	-1.129255	Pass
36	.043155	Short	-1.656370	Open
37	1.005788	Pass	-1.466424	Pass
38	0.691093	Pass	-1.693167	Open
39	1.518586	Open	-0.374513	Pass
40	0.872069	Pass	-0.406839	Pass
41	0.124659	Short	-1.381942	Pass
42	0.058358	Short	-0.177010	Short
43	1.286479	Pass	-1.178606	Pass

Appendix A (Continued)

44	0.117081	Short	-0.256172	Pass
45	1.274879	Pass	-1.360276	Pass
46	1.402936	Pass	-0.847809	Pass
47	0.228714	Pass	-0.369905	Pass
48	0.525161	Pass	-1.181097	Pass
49	0.640141	Pass	-1.550513	Open
50	1.323500	Pass	-1.500063	Open
51	1.656221	Open	-1.102876	Pass
52	0.942711	Pass	-1.626993	Open
53	0.001086	Short	-0.806499	Pass
54	0.574649	Pass	-1.405907	Pass
55	1.199725	Pass	-1.075051	Pass
56	0.447933	Pass	-0.769696	Pass
57	0.171234	Short	-0.154223	Short
58	1.245246	Pass	-1.580868	Open
59	0.653812	Pass	-0.203305	Pass
60	0.405995	Pass	-0.499202	Pass
61	0.452557	Pass	-0.253531	Pass
62	0.274412	Pass	-1.089120	Pass
63	0.925508	Pass	-1.387281	Pass
64	1.513428	Open	-1.372787	Pass
65	0.923241	Pass	-1.034993	Pass
66	1.222140	Pass	-0.220205	Pass
67	1.396794	Pass	-1.661005	Open
68	0.161699	Short	-0.963521	Pass
69	1.460785	Pass	-1.434476	Pass
70	1.231847	Pass	-0.736539	Pass
71	0.870046	Pass	-0.316153	Pass
72	1.590089	Open	-0.554404	Pass
73	1.092615	Pass	-0.984914	Pass
74	1.058062	Pass	-1.266852	Pass
75	1.263365	Pass	-0.790903	Pass
76	1.060462	Pass	-1.290061	Pass
77	1.659684	Open	-1.174182	Pass
78	1.049395	Pass	-1.674798	Open
79	0.836322	Pass	-1.200990	Pass
80	1.108909	Pass	-0.352898	Pass

Appendix A (Continued)

81	0.724420	Pass	-1.453072	Pass
82	0.027815	Short	-1.572945	Open
83	0.576465	Pass	-0.187800	Short
84	1.605419	Open	-0.281719	Pass
85	1.475880	Pass	-0.440498	Pass
86	0.676584	Pass	-0.856728	Pass
87	1.158302	Pass	-1.032188	Pass
88	0.348097	Pass	-0.258112	Pass
89	0.408434	Pass	-0.617304	Pass
90	1.686674	Open	-1.367849	Pass
91	1.120596	Pass	-0.962751	Pass
92	1.687446	Open	-0.507889	Pass
93	0.602361	Pass	-0.122315	Short
94	1.020057	Pass	-0.713826	Pass
95	0.608204	Pass	-0.628230	Pass
96	1.488129	Pass	-0.157910	Short
97	1.301591	Pass	-1.699087	Open
98	0.086991	Short	-0.016451	Short
99	0.053475	Short	-1.693068	Open
100	0.101683	Short	-0.500861	Pass
101	1.045759	Pass	-0.330425	Pass
102	1.287201	Pass	-0.247618	Pass
103	0.173470	Short	-0.381635	Pass
104	0.330563	Pass	-0.834305	Pass
105	0.821084	Pass	-1.462698	Pass
106	0.333805	Pass	-1.546624	Open
107	1.688960	Open	-0.510436	Pass
108	0.102617	Short	-1.690308	Open
109	0.299792	Pass	-0.009033	Short
110	0.030264	Short	-0.530117	Pass
111	0.164182	Short	-0.986906	Pass
112	0.776102	Pass	-0.165435	Short
113	0.863756	Pass	-0.866064	Pass
114	0.107027	Short	-0.902936	Pass
115	1.337861	Pass	-1.095554	Pass
116	1.016959	Pass	-1.001589	Pass
117	1.265560	Pass	-0.349866	Pass

Appendix A (Continued)

118	0.083394	Short	-0.424164	Pass
119	0.876180	Pass	-0.006515	Short
120	0.611386	Pass	-0.248144	Pass
121	0.181871	Short	-0.727131	Pass
122	0.659669	Pass	-1.606379	Open
123	1.505025	Open	-1.671724	Open
124	0.064656	Short	-0.427799	Pass
125	0.844302	Pass	-0.767644	Pass
126	0.275767	Pass	-1.169666	Pass
127	0.539814	Pass	-0.525704	Pass

The test was run with the following test equipment:

Control # 8854: PXI-8105 Embedded Computer
Control # 8866: PXI-2530 FET Switch Matrix
Control # 8856: PXI-4130 Suorce Measurement Unit
Control # 1253: PXI-1045 PXI Chassis