DEVELOPMENT OF HIGH-DENSITY RF BOARD DIAGNOSTIC PROGRAM

by

SUKESHWAR KANNAN

A THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the department of Electrical
and Computer Engineering in the Graduate School
of The University of Alabama

TUSCALOOSA, ALABAMA

2010

ABSTRACT

This thesis describes the development and application of a software tool. RF Analyzer and Diagnostic Program Generation Tool (RADPro) has been developed to automate the process of testing the RF Device Interface Boards (DIB). DIBs are essential components for testing ICs and they contain mixed-signal and RF circuits with several active and passive components that are needed to simulate test conditions for ICs. A new integrated software environment has been developed to automatically generate functional tests for board verification. This software environment utilizes the schematic information, DIB-specific constraints, accessibility provided by the test hardware and instrument automation tools to generate a functional test program. The output of the tool is a generic test specification that is independent of test hardware platform.

RADPro has been developed for testing RF DIBs at Texas Instruments Inc. Test execution has been performed on several RF DIBs and the test results have been recorded in the pseudocode format as well as look up table which provides a detailed analysis of all possible faults that can occur in the passive components. Complete automation of DIB test methodology implementation has been presented in this thesis.

Automatic test generation by RADPro reduces design expenses and time to market significantly in comparison with the existing techniques. The testing methodology presented in

this thesis produces the pseudocode for automating the test procedure on Very Low Cost Tester (VLCT) systems.

LIST OF ABBREVIATIONS AND SYMBOLS

RADPro	RF Analyzer and Diagnostic Program Generation Tool
DIB	Device Interface Board
DUT	Device Under Test
ATE	Automatic Test Equipment
FCMV	Force Current Measure Voltage
FVMC	Force Voltage Measure Current
PCB	Printed Circuit Board
ICT	In-circuit Test
ADS	Advanced Design System
RF	Radio Frequency
ETS	Eagle Test System
CUT	Circuit under Test
BOM	Bill of Materials

ACKNOWLEDGEMENTS

I would like to express my whole hearted gratitude to all the faculty members, mentors, sponsors, family, and friends who have helped me during this project.

I would like to thank Dr. Bruce Kim, my advisor and chairman of the thesis committee, for his technical guidance and administrative help throughout this project. He has been a great mentor and constant driving force to help me reach project goals effectively and efficiently on time. I would also like to thank my committee members, Dr. Tim Mewes and Dr. Qi (Bob) Hao, for their valuable inputs and feedback.

I am indebted to Dr. Ganesh Srinivasan, Texas Instruments Inc., for mentoring and guiding me through the entire duration of this project with technical assistance and support. I would also like to thank Dr. Friedrich Taenzlar, Richard Antley, Craig Force, Chris Vogel, Nick Flores and the entire research team at Texas Instruments for their active participation in all our review meetings during the course of this project, feedback and their valuable time in the midst of their busy work schedule.

I would also like to thank Drs. Falah Mohammed and Byoungchul Ahn for their technical inputs and support.

Finally, I would like to thank my family and friends who have been a constant source of support and inspiration during the project.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Integrated circuits (ICs) constitute an area of microelectronics in which many conventional electronic components are combined into high-density modules. Integrated circuits are made up of active and passive components, such as transistors, diodes, resistors, and capacitors. Because of their reduced size, use of integrated circuits can simplify otherwise complex systems by reducing the number of separate components and interconnections. Their use can also reduce power consumption, reduce the overall size of the equipment, and significantly lower the overall cost of the equipment concerned. ICs technology has come a long way since its inception with their presence in computing, consumer electronics and cellular phones to name a few.

A typical IC production cycle starts from identifying the specifications for the particular application. This is followed by design and layout where a design engineer would validate the design in software environment. The designs are then sent to the fabrication facility where the chip is realized. These chips are then packaged to be used as an end product [1]. Electronic packages provide a means for interconnecting, powering, cooling and protecting (IC) chips. Since semiconductor chips are expensive, a testing scheme is necessary to ensure the integrity

and performance of all the package interconnection paths. Finally the finished product is ready to be tested and shipped to the customers.

Testing involves verification of functional requirements of the IC and elimination of bad products as early as possible, thus saving on cost. This involves verification during the design and layout cycle and also post silicon validation.

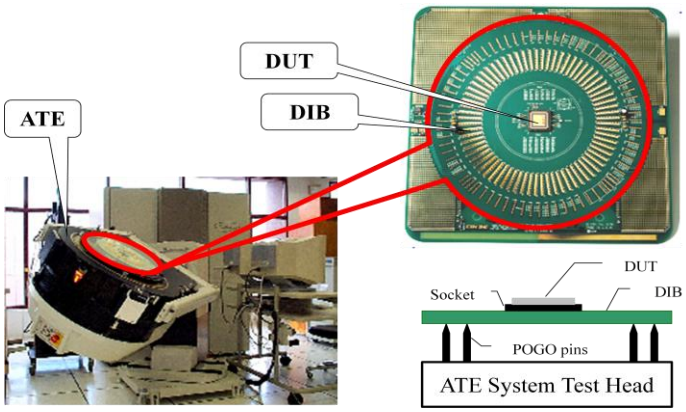
Post silicon validation includes:

1. Wafer level testing.
2. Bench characterization of ICs.
3. Production floor validation and testing.

Wafer level testing involves testing of unpackaged bare silicon die. This process eliminates the defective bare dies which involves detection of process related defects. Post silicon bench characterization is the first step of evaluation of a packaged IC. The next step is the production floor validation and testing to identify the manufacturing defects. They are tested for different specifications based on their applications.

This production testing is done using an Automatic Test Equipment (ATE) that have inbuilt instrumentation and provide programming capability to test the ICs as shown in Figure 1.

Figure 1. Automatic Test Equipment used to test ICs.



There is a constant increase in the complexity of testing procedure with the decrease in footprint of the ICs. The complexity of the DIBs used to test ICs also increase with the DUT complexity. Some of the challenges include development of test techniques, reduction of test time and cost, increasing the access and controllability for testing the IC. Since the test time and cost of testing has direct impact on the time to market the IC, it has become necessary to offer solutions to provide reliable and cost effective methods to test the ICs.

The testing infrastructure plays a vital role in the measurements to be verified, time and cost of testing. The testing infrastructure primarily includes,

1. Hardware platform and instruments to provide input stimulus and measure output response.
2. Software tools for programming the instruments to automate the test procedure.
3. Device Interface Board (DIB) to provide interface between the tester instrument and the ICs to be tested.

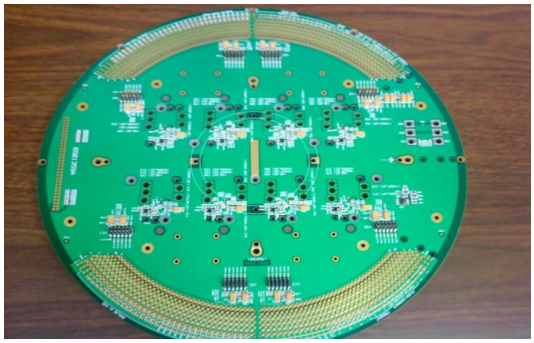
These instruments must be reliable and calibrated to perform measurements. They should also meet the challenges for making measurements for all the specifications of the ICs.

The most essential part of the testing procedure is the DIBs. They contain circuits which facilitate the user to make measurements for different specifications and they also vary for each IC. Their complexity increases with that of the Device Under Test (DUT). A new software tool has been developed to perform tests to verify the DIBs. This thesis discusses in detail the different algorithms and testing procedures used to test the RF DIBs.

Device Interface Board

A load board, interface board, or DUT board is a circuit board designed to serve as an 'interface' circuit between the Automatic Test Equipment (ATE) and the Device Under Test (DUT). They are designed to provide the test input stimulus to the device under test (DUT). A typical DIB (device interface board) for testing RF devices is predominately comprised of analog and RF circuits on a printed circuit board with multiple component types (capacitors, resistors, diodes, filters, baluns and ICs). Presently there are several techniques used for testing assembled DIB boards, such as in-circuit tests, functional tests and flying probes. To achieve high test coverage, each of these test methodologies is labor-intensive and expensive. Other limitations include lack of accessibility to test nodes on the board with small footprint components and connectivity from the DUT socket to the board [2]. Although the importance of testing these device interface boards in a timely and accurate manner is well understood, test engineers currently do not have the proper set of tools to achieve the goal of testing the boards in hours rather than days [3]. In the absence of these tools, the engineer inserts new silicon into a new device interface board that is not certain to be fault-free. Hence there is an urgent need for automatic hardware diagnostic capability to ensure the functionality of the test hardware.

Figure 2. Typical RF Device Interface Board



The ICs are tested for their functionality and characterized extensively for their performance under different specifications. The next stage of production testing mainly involves identifying manufacturing defects in ICs on the production floor [4]. Device interface boards are printed circuit boards used to provide an interface between tester hardware and the device that needs to be tested. They are designed as a means to provide test signals to the device under test (DUT). They also provide the means to stimulate the operating conditions for the device under test to evaluate its performance.

A typical DIB as shown in Figure 2 is comprised of mixed-signal and RF circuits with several component types including capacitors, resistors, RF filters, RF traces, ICs and connectors. Apart from these discrete components, they also have sockets that hold the DUT and enable contact between DUT pins and the traces on the DIB. The complexity of these boards is also determined by the trace interconnecting these components on board. The overall complexity of the DIB primarily depends on the complexity of the DUT, tester hardware platform and test functions required. A DIB used with an ATE in the IC production environment is comparatively simpler than the DIBs designed for bench characterization of ICs. Also, on the IC production floor, where massive IC testing is carried out on ATE, DIBs become faulty because of the failure of components on boards in use. This creates a requirement for techniques to test DIBs before they are employed for IC testing and also at regular intervals on the IC production floor to ensure that the DIBs remain good. The currently existing techniques for testing DIBs are very expensive, time-consuming and limited in their capability due to the tester instrumentation. Thus a new test generation methodology for testing complicated DIBs in a time-efficient manner is presented.

Motivation for Device Interface Board Testing

As the first batch of silicon arrives from the fabrication facility, the test engineer is responsible for testing the IC and characterizing data. This task is accomplished by performing testing on the Automatic Test Equipment which has to be fault free. Once a new IC is made a Device Interface Board (DIB) is created to test the IC [5]. The test engineer does not have the proper set of tools to verify the DIB with its schematic. In the absence of these tools the test engineer inserts the silicon into the DUT socket in the DIB, during the testing process any deviation from expected results cannot be attributed to the malfunction of the DUT or DIB [6]. Also on production floors DIBs already in use and tested to work efficiently can become faulty as some passive component may fall off or a relay becomes faulty. This creates a requirement for the DIBs to be tested before the silicon is inserted and DUT testing is performed. The present techniques for testing the DIBs are very expensive and time consuming; also some tester platforms use the floating ground concept for differential testing. Thus for testing RF DIBs and also mixed signal DIBs on tester platforms with floating ground a new test methodology was developed and test generation was automated with RF analyzer and Diagnostic Program Generation Tool.

Current Device Interface Board Testing Techniques

There are several test techniques used to test DIBs. The most commonly used are In-circuit testing, functional testing and flying probe test. In-circuit test (ICT) is an example of white box testing where an electrical probe tests a populated printed circuit board (PCB), checking for shorts, opens, resistance, capacitance, and other basic quantities which will show whether the assembly was correctly fabricated. It may be performed with a bed of nails type test

fixture and specialist test equipment, or with a fixtureless in-circuit test setup. This technique of testing PCB's is being slowly superseded by Boundary Scan techniques (Silicon Test Nails), Automated Optical Inspection, and built-in self-test, due to shrinking product sizes and lack of space on PCB's for test pads.

In the flying probe tester the bed of nails are replaced by flying probes which make a mechanical contact with the DIB. The capabilities of a Flying Prober should be separated into two categories, Electrical Test and Mechanical Interface. First it is necessary to physically contact the board, then to perform a test [7]. Deficiencies in either of these areas will rapidly diminish the effective use of the system. But first and foremost, a Flying Prober is still an in-circuit tester.

The main challenge faced by the flying probe tester is the accessibility for small footprint components [8]. Also the time taken to test one DIB is in weeks compared to the automatic program generation testing which can test a DIB in hours.

The thesis describes the new test methodology used to test embedded passive RF components which are major bottleneck in testing RF DIBs. The overall approach, software architecture and algorithms are described in chapter 2. The modeling of embedded passive RF components for detecting process related defects are presented in chapter 3. The test methodology, results and board coverage are discussed in chapter 4. The thesis concludes with an overall summary of the proposed test technique and software tool implementation of the technique.

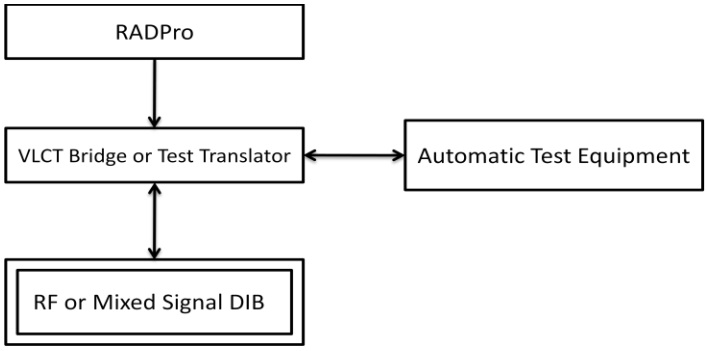
CHAPTER 2

RESEARCH METHODOLOGY AND APPROACH

The test setup for RF Device Interface Boards (DIB) involves the hardware tester system that can interface with the DIB to provide test channels and software that can automate the generation of tests for the RF DIB. In production floors the automatic test equipment (ATE) is used to provide interface to the DIB and test built-in programmable functions [9].

The test hardware consists of tester platform and DIB. The software uses the test methodology proposed in this thesis to automate the test generation process. We have developed a software tool RADPro (RF analyzer and Diagnostic Program Generator) to analyze the DIB and completely automate the DIB testing.

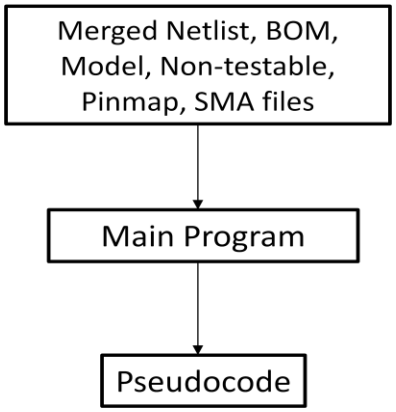
Figure 3. Block Diagram of Automatic RF DIB Testing.



This diagnostic software tool has been developed for debugging ATE mixed-signal load boards. However, the original design is intended for simple load boards, which have few relays and active components with limited number of resources on the ATE. In comparison, the RF

device interface boards used in first silicon characterization are highly complicated. An RF DIB comprises of mixed-signal and RF circuits with several component types including capacitors, resistors, embedded passive RF components, ICs and connectors. The complexity of boards is also determined by trace interconnecting these components on boards. Hence there is a need to analyze these mixed signal and RF components, run characterization test on the device interface board and locate the type of fault. Figure 4 illustrates the RADPro testing procedure.

Figure 4. RADPro Testing Procedure

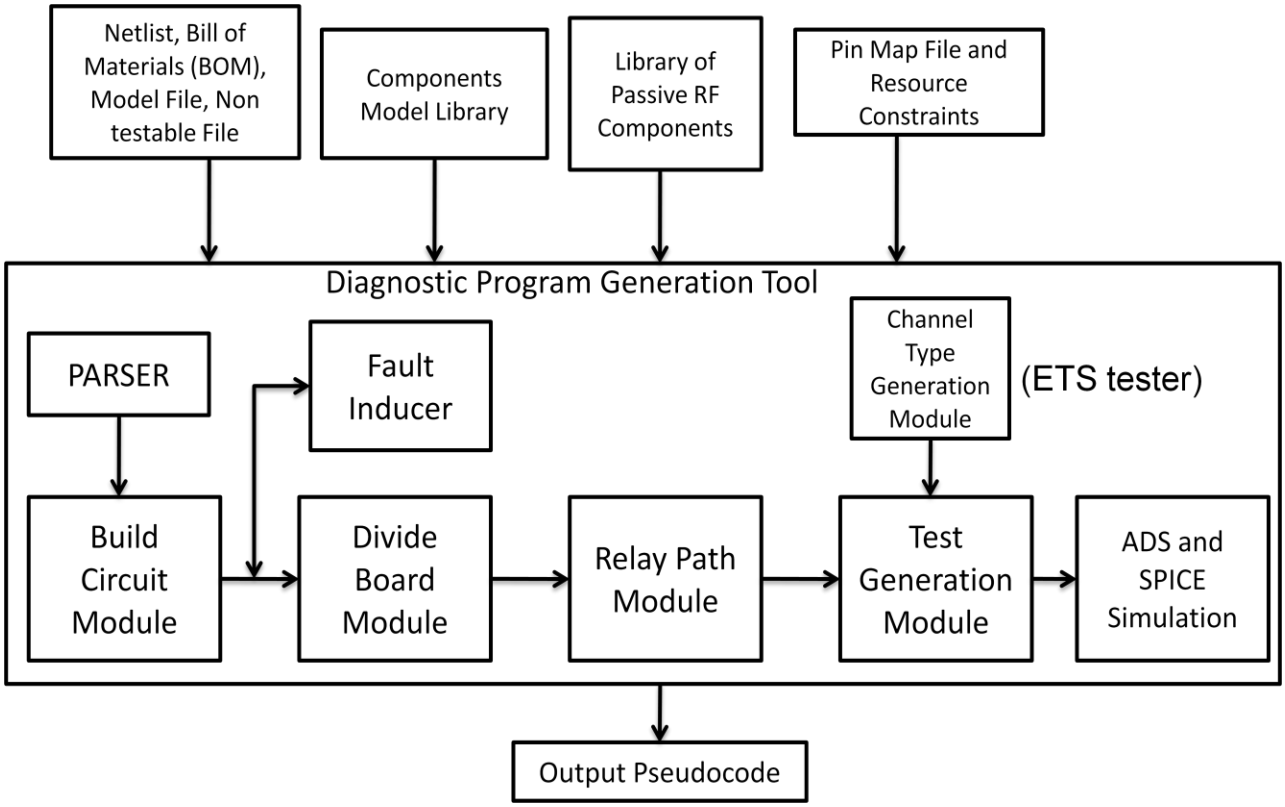


The main RADPro software architecture uses the schematic information of the DIB such as the netlist, bill of materials, pinmap files to replicate the schematic of the board along with all the components and necessary connections. A model library is built for all the active and passive components of the board to increase the test coverage of the respective board. The tool runs a simulation for the testable circuits in the DIB and stores the results in a look up table. The output of the program is a pseudocode which consists of test instructions to be carried out on the ATE. After converting the pseudocode into the native language of the tester using the VLCT bridge or test translation software as shown in Figure 1 the tests are run on the ATE and the output is sent back to the tool which matches the values with the look up table and outputs a pass or fail result

for each individual component in the testable circuit. We have also modeled a set of faults for each component which helps us to identify the likely type of fault associated with the component which fails the test.

The software architecture of RADPro consists of six different modules to automate the process of testing. The following Figure 5 shows the architecture of RADPro and how the testing process is done for each DIB board.

Figure 5. RADPro software architecture



Parser

The parser is the front end of the software tool (RADPro). The parser builds the circuit of the device interface board using its schematic information. the schematic information of the device

interface board is passed onto the software tool by the input files such as the netlist, partslist, bill of materials and the RF component input file for RF device interface boards. The parser is built such that it would verify the format of these input files [10]. Once the input file formats are verified then the parser would record the total number of components present in the bill of materials and RF components input list. The RF components input list is created as a text file by the user as shown in Figure 6.

Figure 6. Embedded Passive RF component schematic information

RFR1	D410	50Ohms	L=433.972mils W=20.69mils H=10mils Er=3.9
RFR2	C214	100Ohms	L=456.85mils W=4.85mils H=10mils Er=3.9
RFR3	U303	37.5Ohms	L=400.576mils W=25.961mils H=10mils Er=3.9
RFB1	R439	70.7Ohms	L1=700mils L2=2100mils W1=10.2mils W2=8mils H1=10mils H2=8mils SPL1L2=10mils SP12L1=10mils SPL1=30mils

Building the board circuit also involves prior knowledge of available terminations for the board under test. This information is carried forward to the generation of testable sub-circuits. Once the input files are processed then the tool proceeds to build the circuits as present in the DIB board. The termination information of the board is obtained from the pinmap files for the board which are tester specific and also vary from board to board. The tool also checks for all model files stored in the model library for active components.

Build Circuit Module

The build circuit module receives the output of the parser which consists of verified input files. These input files are used to build the actual circuit of the device interface board in the software tool. The netlist gives the details of how the components are actually connected on the device interface board. However for the RF device interface boards the netlist does not consider the embedded passive RF component as the netlist contains only the components which are

present in the parts list. Embedded passive RF components are present in the device interface board but are not considered to be discrete components hence they are absent from the netlist and the parts list.

Automatically Inducing the RF Component

Originally the embedded passive components are not present in the parts list input file to the tool as they are not classified as a discrete component. Since it is not present in the parts list it is also absent in the netlist as the netlist shows only the connections of all the components in the parts list. Hence the user has to observe the schematic and record the presence of these embedded passive RF components. Once the properties are recorded as shown in Figure 6 they are automatically induced into both the netlist and the parts list in the Parser as shown in Figure 7.

The procedure for automatically inducing the embedded passive RF component into the netlist and parts list is given as follows:

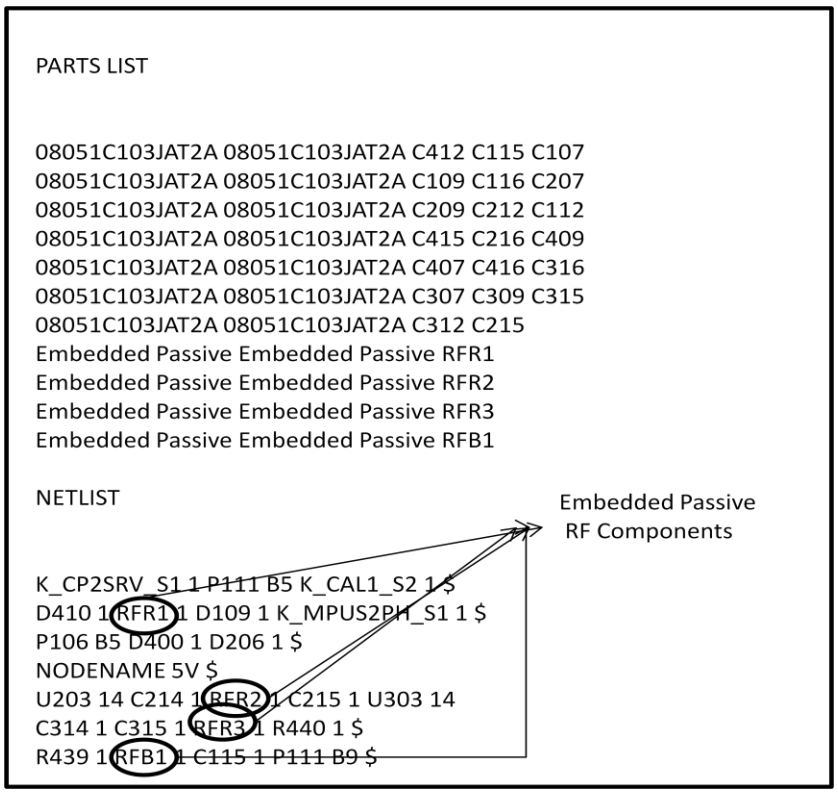
Into Parts List:

1. Read RF input text file.
2. State the component model number as embedded passive.
3. Store the component type as the part number.

Into Netlist:

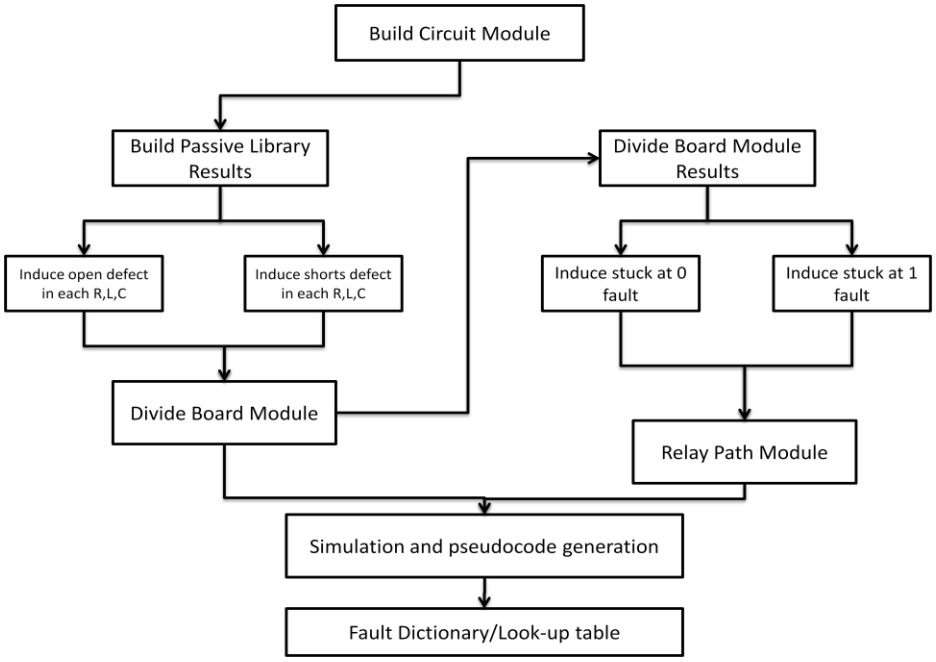
1. Read the RF input text file.
2. Record the component number and part in the netlist after which it has to be placed.
3. Read the netlist.
4. Place the component into the netlist.

Figure 7. Embedded Passive RF component induced into netlist and parts list



In the build circuit module the DIB components are segregated and stored as active and passive libraries. The active components are tested based on the availability of a model file which has to be user created to increase the test coverage of the board. The passive component library is used to create a serially changing fault inducer module. In the module faults are automatically induced in the passive components of the board to obtain possible fault simulation values as shown in Figure 8. These results are stored in the look up table and matched with the actual ATE hardware measured results.

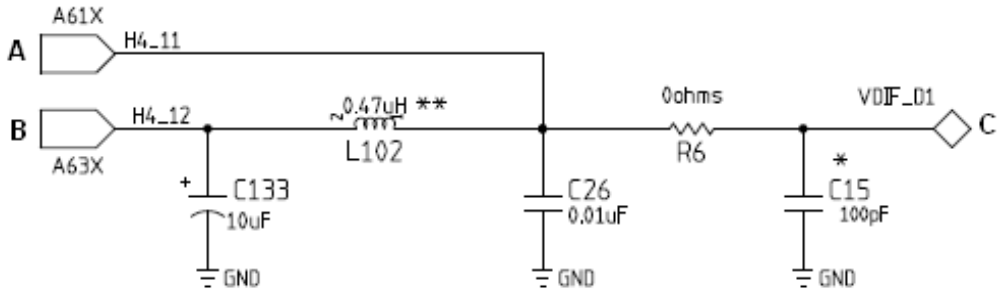
Figure 8. Serially Changing Fault Inducer Procedure



Divide Board Module

Using the partition algorithm we divide the board into small testable circuits. The output of the build circuit module is taken to divide the board into testable sub-circuits based on the availability of terminals for stimulating and measuring test signals [11]. Figure 9 shows a typical RF DIB circuit.

Figure 9. Typical DIB Circuit



We use the following partition algorithm to divide the board into small testable circuits.

1. Choose a node that is connected to a pogo pin of the tester using the information from the pinmap files as shown in Figure 9. Mark this node as visited.
2. Find all the components linked to the start node and group them as a single block. As an example, nodes A and B are connected to start node C.
3. Repeat step 2 for all the nodes in the previously grouped block until another termination node is encountered. Mark the visited nodes.
4. Mark the path between the start node and the node with termination as a testable sub-circuit.
5. Repeat steps 3 and 4 until all the nodes are marked visited.

Once the circuits are divided into testable sub-circuits the circuits with relays in them are tested separately for the different operational modes of the relay. For the ETS boards the circuits have to be divided such that differential testing can be performed. This means that the circuits have to be divided as one single testable circuit with high and low sides. Hence the channel type generation was used to accommodate this change in ETS boards. The pin map file contains the hardware details of the pogo pins that are used for communication to the device interface board. The pin map file for ETS boards carry the hardware equipment type that is connected to the pogo pins. This is used to automatically generate the channel type which facilitates in dividing the board into testable circuits with a differential path.

Floating Ground in ETS DIBs

Floating ground concept is used in ETS platform where the tests are run as a differential pattern with high and low sides in force and sense channels. The pinmap file does not contain the information of the channel type associated with each pogo pin at the input side of the DIB.

During this scenario the divide board module of the tool does not understand that the high and low side of a differential circuit has to be partitioned as a single sub-circuit. Hence the channel type generation module was added to the architecture to overcome this challenge with defines the channel type for each pogo pin of the netlist for ETS DIBs. This is as shown in Figure 10.

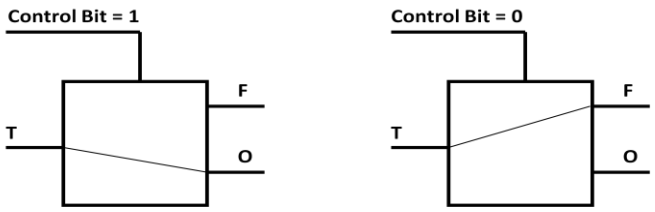
Figure 10. Channel type association to pogo pins for ETS DIBs.

REF	PIN	VALUE	Channel Type
P105	A11	BBUS40-47S	ETSNC
P105	A12	MPUSL89	ETSNC
P105	A13	MPU89SLB	Sense Low B
P105	A14	MPU89SHB	ETSNC
P105	A2	APU35S	Sense
P105	A3	NC	No Connection
P105	A4	BBUS32-39S	ETSNC
P105	A8	APU47S	Sense
P105	A9	APU43S	Sense
P105	B1	APU39F	Force
P105	B10	NC	No Connection
P105	B11	BBUS40-47F	ETSNC
P105	B12	MPUFL89	ETSNC
P105	B13	MPU89FLB	Force Low B
P105	B14	MPU89FHB	ETSNC
P105	B2	APU35F	Force
P105	B3	NC	No Connection
P105	B4	BBUS32-39F	ETSNC
P105	B8	APU47F	Force
P105	B9	APU43F	Force
P105	C1	APU38S	Sense
P105	C10	APU40-47SL	Sense Low
P105	C11	ABUS40-47S	ETSNC
P105	C12	MPUSH89	ETSNC
P105	C13	MPU89FLB	Force Low B
P105	C14	MPU89FHB	ETSNC
P105	C2	APU34S	Sense
P105	C3	APU32-39SL	Sense Low
P105	C4	ABUS32-39S	ETSNC

Relay Path Module

The circuits with relays in them are tested by switching the relay on or off to check for faults in the relay circuits. For fault modeling purpose the relay we have used struck at 0 and struck at 1 fault. Figure 11 shows the two relay configurations used in testing procedure in relay path generation.

Figure 11. Relay configurations for testing and fault modeling



Channel Type Generation Module

In some tester platforms such as the Eagle Test System (ETS) there is a floating ground concept. Here the source meters and measuring instruments do not share a common ground and this is used as a pedestal to apply and measure higher voltages. In such cases the tool must specifically use the high or low channels available at the respective pogo pins. This information is taken from the pinmap files and the channel type is created for each pogo pin as to say whether it is a high or low and force or sense activity is taking place at the pin. Figure 12 shows the tester resource having floating ground instruments, and Figure 13 shows a typical DIB circuit with floating ground.

Figure 12. Tester resource with floating ground instruments

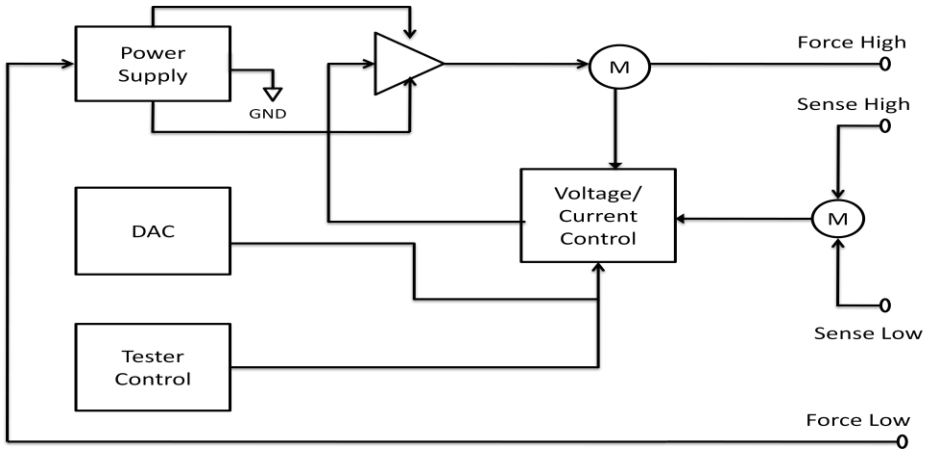
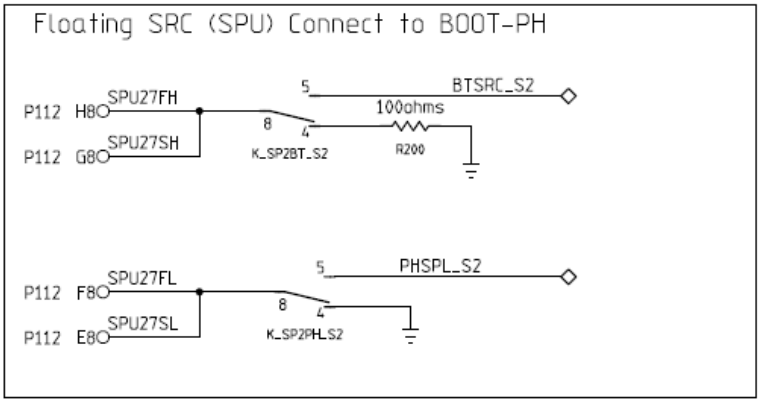


Figure 13. DIB circuit with Floating ground



Test Generation

This module generates pseudo-code that contains information for testing the board. Each line of pseudo-code has specific parameters that include type of test, expected output values, and termination names. The expected output values are obtained by ADS or SPICE simulation of the circuits. Figure 14 shows a sample pseudo-code for capacitor and relay testing.

Figure 14. Pseudocode generated by RADPro

*****CAPACITOR ONLY TEST*****

CONDITION (circuit id = 1013, test type = CAP TEST, test pin = CH-U1-M2, must be grounded pogo pin = GND)

FCMVDT (forced current = 1e-005, range of voltage = ".range_vi(0.5).", driving time = 0.05, driving pin = CH-U1-M14, pin type = JTAGDOT4)

COMPARE_LIMITS (lower limit = 0.45 V, upper limit = 0.55 V, circuit id = 1013, tested components = "C5", effective cap value = 1e-006, tolerance = 10%, effective low = 9e-007, effective high = 1.1e-006)

*****TEST ON CIRCUIT WITH RELAY*****

RELAY ON (K21 , K20)

FCMV (forced current = 1.000000e-003 A, driving pin = CH-U1-D2, pin type = JTAGDOT4, grounded pin = CH-J1-B25)

COMPARE_LIMITS (lower limit = 0.9009 V, upper limit = 1.1011 V, nominal value = 1.001000e+000, circuit id = 300900, tested components = K21,K20)

RELAY OFF (K21 , K20)

Simulation involves fault modeling and test technique. We have developed two different test techniques to perform fault analysis on the RF device interface boards. The results of the simulation are stored in the look up table and matched with the output of the ATE to find out whether a component has passed or failed the test.

ADS/SPICE Simulation Module

The output of the divide board module gives a set of testable and non-testable circuits using the partition algorithm. For each testable circuit simulations will be run and the results of the simulation will be stored in the look-up table. To analyze the RF components in the testable circuits we have performed fault modeling of the commonly present RF components and surface mount passive components. The simulations are run for fault free circuit and also by automatically inducing the faults using the serially changing fault inducer module.

After automatically inducing the faults for all testable circuits we will segregate the RF circuits from mixed signal circuits. SPICE simulation is run for mixed signal components. The tests run on mixed signal circuits are Force Voltage Measure Current (FVMC), Force Current Measure Voltage (FCMV) and Force Current Measure Voltage with Delay Time (FCMVDT). ADS simulation is run for testable circuits with embedded passive RF components. We run S-parameter simulations for RF components. There are two different types of test technique we have developed for detecting faults in RF components – RF power sensor testing and Dither Testing. Using the fault models and the test setup developed we are able to detect faults in mixed signal and embedded passive RF components. The testing procedure and algorithm is shown in Figures 15 and 16.

Figure 15. ADS Simulation Module testing procedure

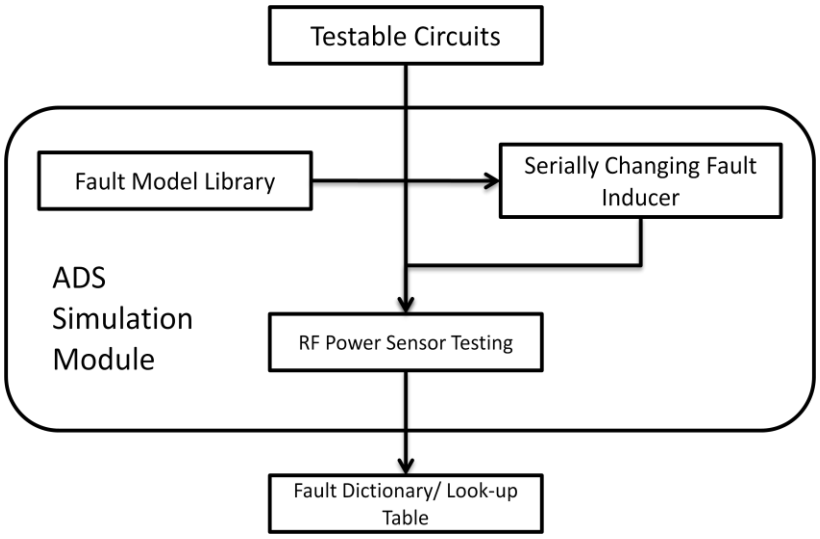
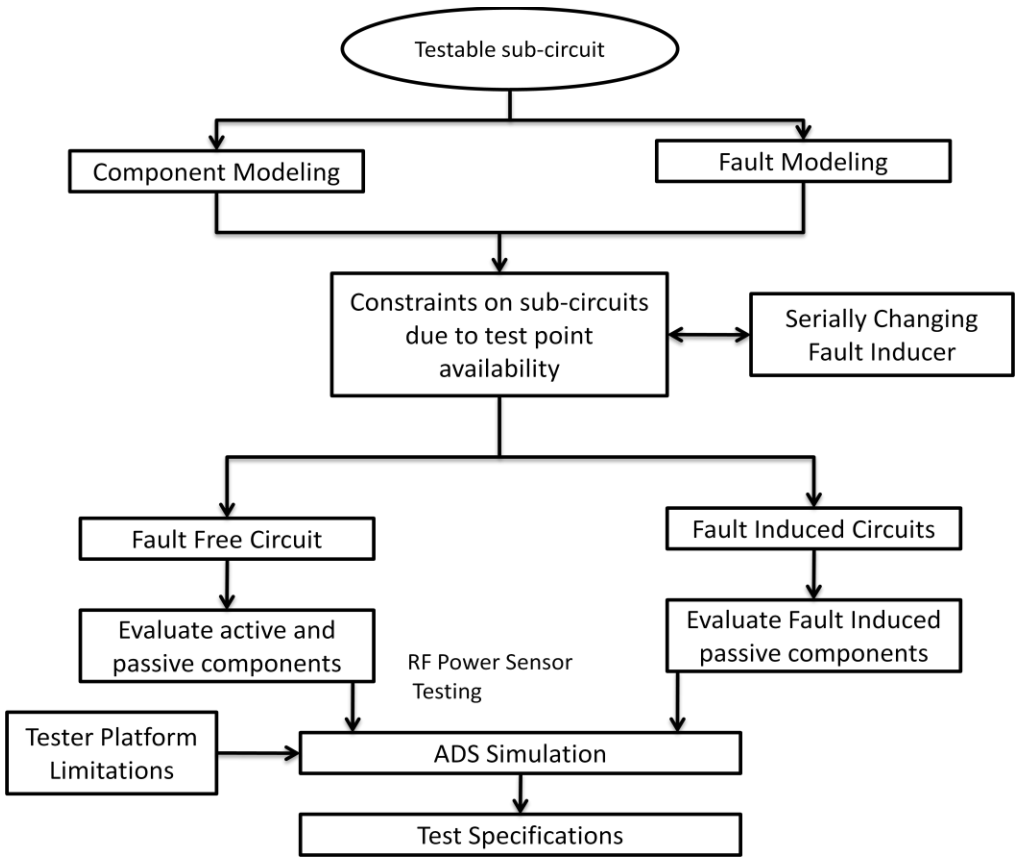


Figure 16. Test Algorithm Structure

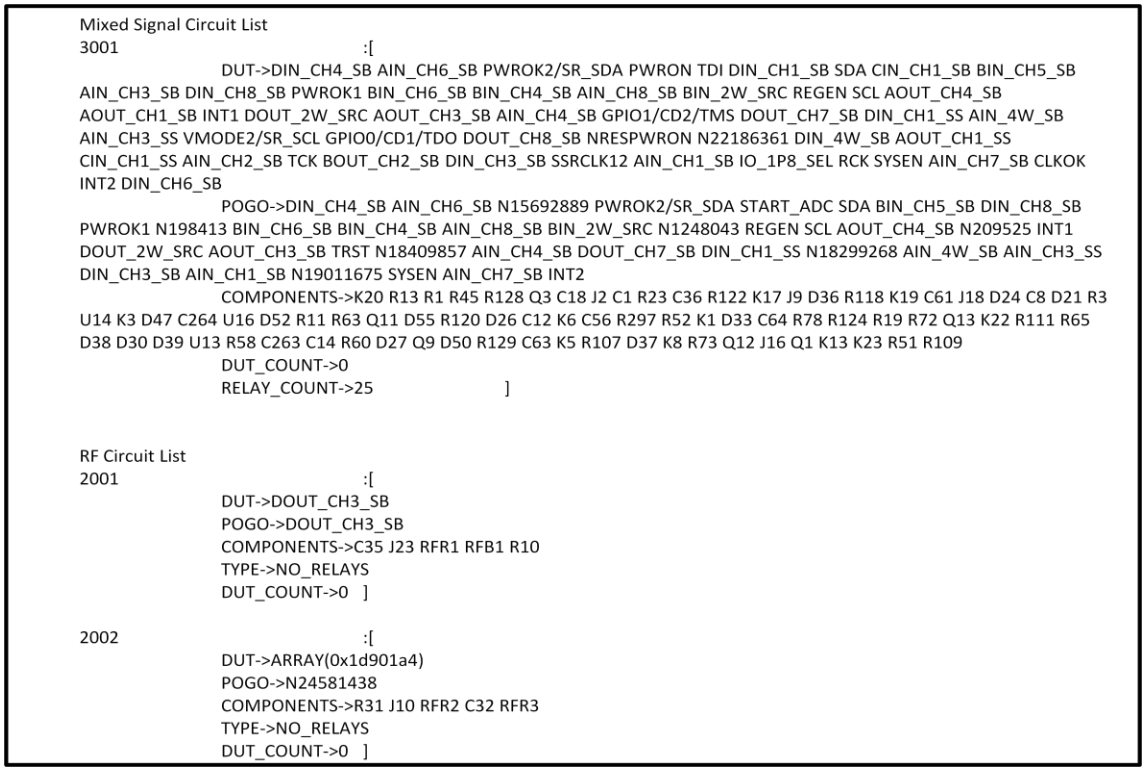


Component Modeling

The testable circuits are simulates using SPICE for mixed signal circuits and ADS for circuits with embedded passive RF components. This makes it necessary to have models prepared for all passive and commonly available active components. With new ICs to be tested the composition of the DIB is also constantly changing. New components from different vendors with varying specifications and parameters need to be modeled before boards are to be tested. In RF DIBs every new shape of embedded passive RF components performing different functions has to be modeled in ADS. Hence the user has to constantly keep updating the model files and model libraries to increase the test coverage of the board.

We have prepared a circuit separator for separating the circuits that have to be simulated in ADS and SPICE separately for RF DIBs. The results of the separator are as shown in Figure 17.

Figure 17. Separator for testable circuits of RF DIBs.



CHAPTER 3

FAULT MODELING

In analog-mixed signal load boards, we modeled faults as complete open or short interconnects. Other types of fault models include out of tolerance component values and assembly errors. In RF load boards, the different types of fault models include process related defects in the embedded passive components. The RF embedded devices include inductors, baluns, and band pass filters. These devices are fabricated between the layers of PCB and they serve important functions in RF circuits [12]. Identification of embedded devices for RF load boards has been difficult. We considered open, short, near open and near short defects for embedded devices as shown in Figure 18.

Figure 18. Examples of interconnect faults.

Type of Defect	Defect Configuration
Defect Free	
Open Defect	
Short Defect	
Near-Open	
Near-Short	

The circuits are divided based upon the presence of a path starting from a pogo pin at the tester to a DUT pin. For all the testable circuits we have to perform testing using the ATE to characterize the DIB. We have built a model library which has the different SPICE and ADS (Advanced Design System) models to perform simulations for the testable circuits. The different fault models are stored inside this model library which helps us to build the simulation setup for each testable circuit.

Let us consider the RF boards in different cases for performing fault modeling as we take up the different faults experienced in the RF load boards. The open connection in traces for analog-mixed signal boards are modeled with a high resistance and shorts are modeled with a capacitance to ground. The stuck at relay faults are also modeled with a capacitance to ground if the relay is struck at off and a resistance if struck at on position.

RF trace is modeled for the process related defects mentioned above. RF Trace is modeled as a transmission line and the power loss across the transmission line is calculated and checked with values obtained from the RMS detector in the physical setup. RF traces are modeled using the line calculator in ADS and the fault conditions are induced into the model synthesized by the line calculator. RF trace acting like a band pass filter is modeled as a passive band pass filter [12]. A look up table with measurements of bandwidth, amplitude, centre frequency etc. for each possible fault occurrence is created by simulating the response for different missing components and this is compared with actual response values to match the type of fault occurring.

Case Study 1 - Open RF trace used in bandpass filter

Consider the circuit in RF load board with embedded devices, the resistors and capacitors as shown in Figure 19. This circuit behaves as a band pass filter. We have performed simulation

and created a look up table for various faults. Figure 20 shows the simulation results for fault free, near open and missing components with respect to frequency. We measured at 400 MHz for the faults and table 1 presents fault dictionary of these faults.

Figure 19. A Part of a DIB Circuit.

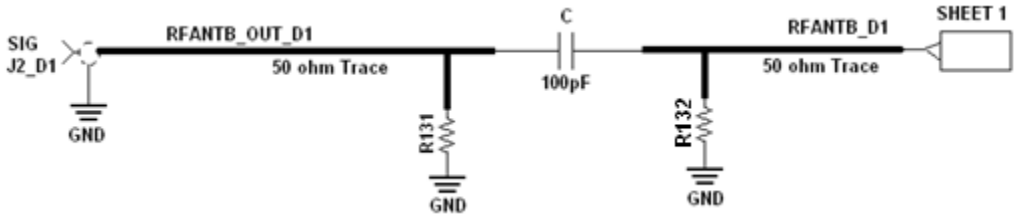


Figure 20. Simulations of Bandpass filter with faults.

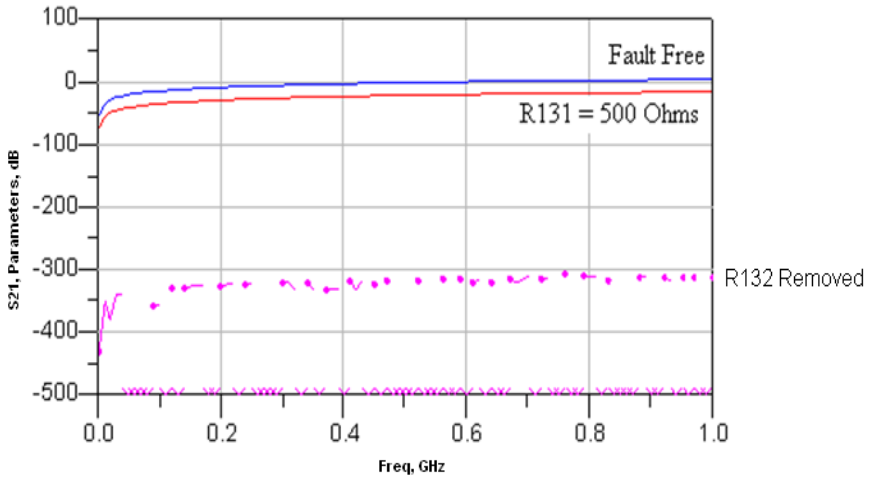


TABLE 1. Fault Dictionary Measurements at 400 MHz.

Name	**Power Gain (dBm)**
Fault Free	-5
Notch (near open fault)	-20
Capacitor removed	-1
R1 removed	-330
R2 removed	-320

Shorted RF trace in bandpass filter

A capacitance is used for modeling short. The value of the capacitance depends on the physical defect on the trace; we simulated shorts and obtained similar results as open defects.

Case Study 2 - Open and short faults in RF balun transformer

Consider a balun formed by an RF trace with open and short faults [11]. Figure 21 shows an RF circuit with balun, and we have inserted open and short defects as shown in Figure 22.

Figure 21. Testable RF Circuit.

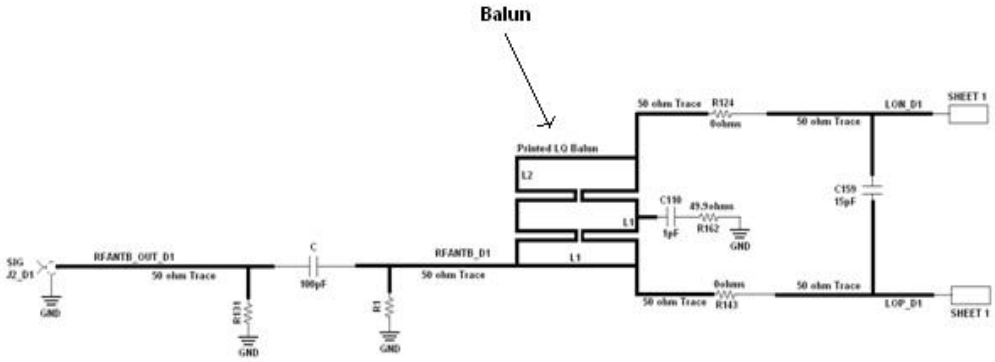


Figure 22. Testable sub-circuit with open and short process related defects at position 1 and 2 in the Balun Transformer.

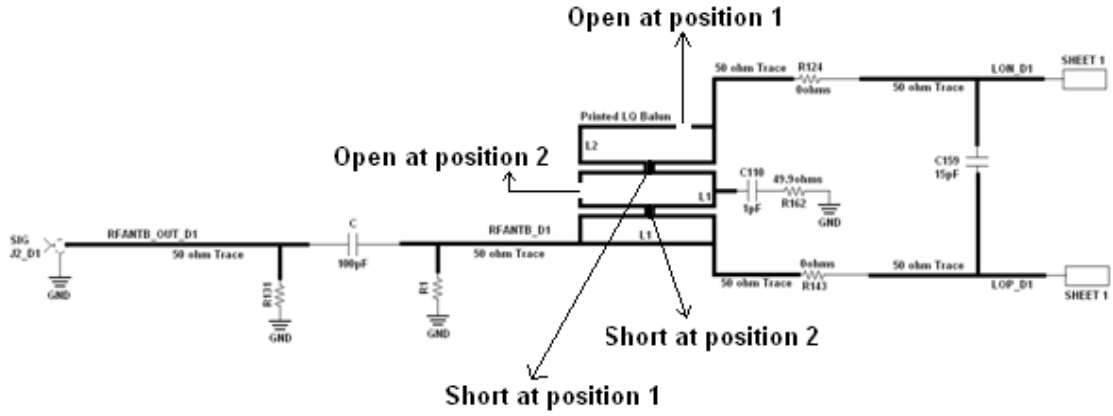


Figure 23 and Figure 24 illustrate measurements of incident waves for open and short defects, respectively. As can be seen from these figures, defects could be distinguished from S-parameter

measurements. We have used the same modeling scheme as described in bandpass filter. Table 2 shows fault dictionary of faults for RF balun measurements at 2 GHz.

Figure 23. Simulated Data for opens in Embedded RF Balun Circuit.

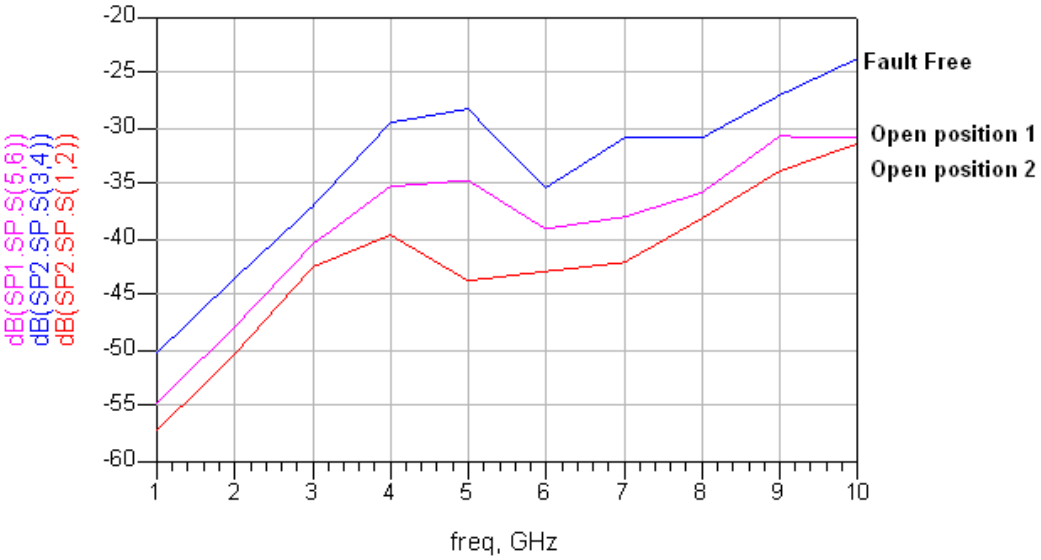


Figure 24. Simulated Data for shorts in Embedded RF Balun Circuit.

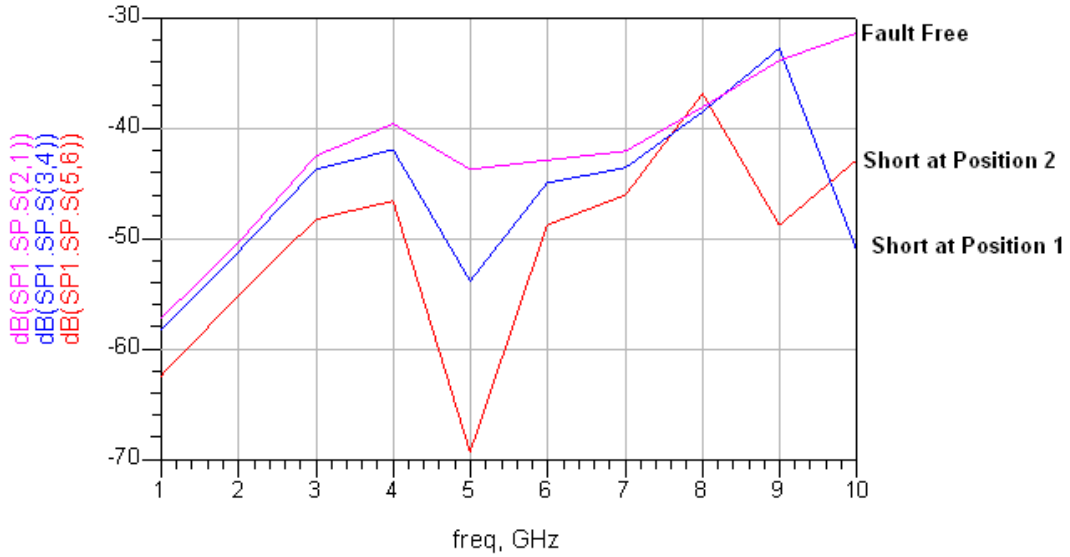


TABLE 2. Fault Dictionary for Balun.

Name	Power Gain(dBm)
Fault Free	-50
Open at Position 1	-45
Open at Position 2	-52
Short at Position 1	-52
Short at Position 2	-56

Table 2 indicates that there are aliasing due to faults. This is unavoidable as we move faults in different locations.

Case Study 3 - Fault models for ETS load boards

For analog mixed signal load boards, the fault modeling is introduced by inducing the desired fault conditions in the bill of materials and executing RADPro. The result obtained from RADPro is the pseudocode which is a set of test instructions to be performed by the Eagle Tester System. The tool is capable of providing a library of Force Current Measure Voltage (FCMV), Force Voltage Measure Current (FVMC) and Force Current Measure Voltage with Delay Time (FCMVDT). We will check the change in values specified in the line with COMPARE_LIMITS. Figure 25 shows fault free pseudocode from RADPro and Figure 26 shows pseudocode with open fault. As can be seen; RADPro provided different values for measurements.

Figure 25. RADPro Pseudocode for Fault free ETS Board.

```
*****POWER TO GROUND ONLY TEST*****
CONDITION(VERIFY COMPONENT = JP19-1_2)
FCMV(forced current = 1e-4 A, force high pin = CH-J1-I16, sense high pin = CH-J1-H9, type = force high)
FCMV(forced current = 1e-4 A, force low pin = LowPin_Num3, sense low pin = LowPin_Ref3, type = force low)
COMPARE_LIMITS(lower limit = 0.06 V, upper limit = 1.99 V, nominal value = 0.075 V, tested pwr node = +3P3V_MB)
```

Figure 26. RADPro Pseudocode for Fault Induced ETS Board.

```
*****POWER TO GROUND ONLY TEST*****
CONDITION(VERIFY COMPONENT = JP19-1_2)
FCMV(forced current = 1e-4 A, force high pin = CH-J1-I16, sense high pin = CH-J1-H9, type = force high)
FCMV(forced current = 1e-4 A, force low pin = LowPin_Num3, sense low pin = LowPin_Ref3, type = force low)
COMPARE_LIMITS(lower limit = 0.003 V, upper limit = 0.167 V, nominal value = 0.059 V, tested pwr node = +3P3V_MB)
```

CHAPTER 4

EXPERIMENTS AND RESULTS

RF Power Sensor Testing

ADS is used for RF simulation. Power loss is checked across the passive RF trace using a power detector/RMS detector as shown in Figure 27. The hardware setup of the RF power sensor board testing is shown in Figure 28. Design Kit library is made for all testable components. Look up table with different faults and measurements is created and simulation results are matched to find type of fault in trace and whether any component connected to RF trace is missing as shown in Figure 29. The RF traces are modeled according to the model library and simulation is performed in ADS. The results of the simulation is compared with the look-up table where the results are stored for all possible faults and then the result is obtained in the form of a pseudocode and output report.

Figure 27. RF power chip used for power loss measurements in the device interface board.

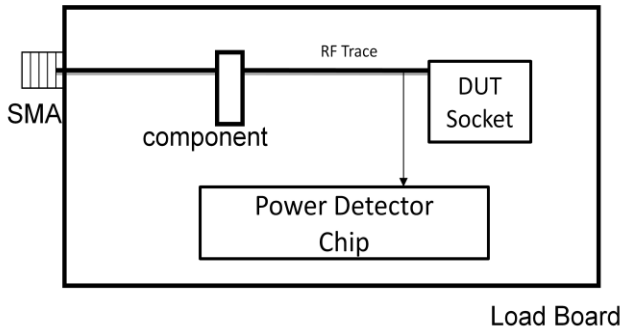


Figure 28. Hardware Setup of RF power sensor implemented on DIB.

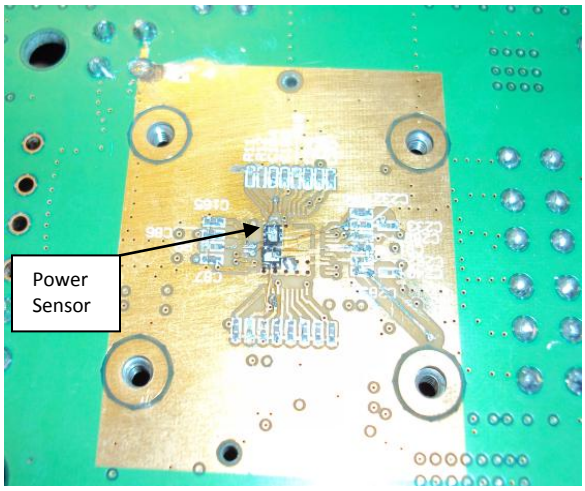


Figure 29. Simulation results of a sub-circuit connected to a power detector.

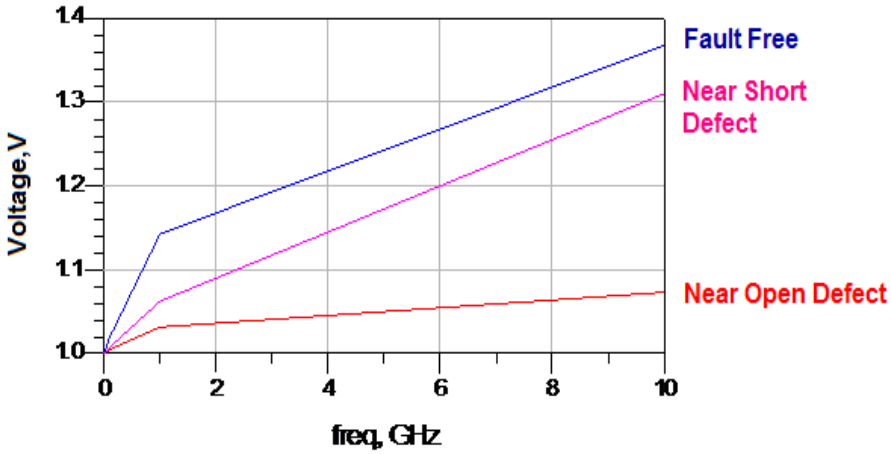


Figure 29 shows that, the differences in voltage levels are clearly distinguished between the fault-free and defective RF traces.

Integration of ADS into RADPro

Advanced Design System (ADS) is the software tool used for RF circuit simulation. We used the AEL commands in ADS to pass the netlist onto the command line mode window and run it as a nutshell. These AEL files were automatically generated for each testable sub-circuit

with RF components as shown in Figure 30. These AEL files are run as a batch simulation from the main RADPro software tool.

Figure 30. AEL command netlist for testable RF sub-circuit.

```
de_set_window_by_sequence(0, 0);
de_new_project("RADPro");
de_set_window_by_sequence(1, 0);
de_save_design_as("untitled1", "RADPro1");
de_set_window_by_sequence(1, 0);
decl itemInfo0SP = de_init_item("C");
de_rotate_inc();
de_rotate_inc();
de_place_item(itemInfo0SP, 0, 0);
itemInfo0SP = de_free_item(itemInfo0SP);
decl itemInfo0SP = de_init_item("R");
de_rotate_inc();
de_place_item(itemInfo0SP, 0, -1);
de_place_item(itemInfo0SP, -1, -1);
de_select_range(-1.934, -2.649, -1.934, -2.649, 1, 1);
de_select_range(-1.934, -2.649, -1.934, -2.649, 1, 2);
decl itemInfo0SP = de_edit_item("DAC1");
de_set_item_id(itemInfo0SP, "DAC1");
de_set_item_parameters(itemInfo0SP,
list(prm("dac","\"RADPro1\""),prm("T1"),prm("StringAndReference",""),prm("M2"),prm("IDAC1"),prm("Extrap1"),prm("StringAndReference",""),prm("Std
Form",""),prm("StringAndReference",""),prm("StdForm",""),prm("StringAndReference",""),prm("StdForm",""),prm("StringAndReference",""),prm("StdFor
m",""),prm("StringAndReference",""),prm("StdForm",""),prm("StringAndReference",""),prm("StdForm",""),prm("StringAndReference",""),prm("StdForm",
""),prm("StringAndReference",""),prm("StdForm",""),prm("StringAndReference",""),prm("StdForm",""),prm("StringAndReference",""),prm("StdForm",""))
);
de_end_edit_item(itemInfo0SP);
de_set_item_id(itemInfo0SP, "DAC1");
de_set_item_parameters(itemInfo0SP,
list(prm("dac","\"RADPro1\""),prm("T1"),prm("StringAndReference",""),prm("M2"),prm("IDAC1"),prm("Extrap1"),prm("StringAndReference",""),prm("Std
Form",""),prm("StringAndReference",""),prm("StdForm",""),prm("StringAndReference",""),prm("StdForm",""),prm("StringAndReference",""),prm("StdFor
m",""),prm("StringAndReference",""),prm("StdForm",""),prm("StringAndReference",""),prm("StdForm",""),prm("StringAndReference",""),prm("StdForm",
""),prm("StringAndReference",""),prm("StdForm",""),prm("StringAndReference",""),prm("StdForm",""),prm("StringAndReference",""),prm("StdForm",""))
);
de_end_edit_item(itemInfo0SP);
de_end_edit_item(itemInfo0SP);
decl itemInfo0SP = de_edit_item("spOutput1");
de_set_item_id(itemInfo0SP, "spOutput1");
de_set_item_parameters(itemInfo0SP, list(prm("StringAndReference","\"RADPro1\""),prm("touchstone"),prm("MA")));
de_end_edit_item(itemInfo0SP);
de_end_edit_item(itemInfo0SP);
itemInfo0SP = de_free_item(itemInfo0SP);
de_analyze();
```

Using the AEL commands above the testable sub-circuit is built in ADS and simulation is performed for RF testing. Co-simulation setup in ADS is used to import the MATLAB model of the power sensor. The template is stored as a model file and the testable circuit is inserted into the template to perform simulation. The AEL files are run as a batch simulation file shown as in Figure 31.

Figure 31. AEL batch simulation file

```
set HOME=C:\ads2009\bin
set LM_LICENSE_FILE = @watto.coe.eng.ua.edu
set HPEESOF_DIR=C:\ads2009
set COMPL_DIR=%HPEESOF_DIR%
cd %HOME%
#C:\Documents and Settings\fmahmoud2003\Desktop -m %1
start ads.exe -m RADPro.ael
```

The simulation run is an S-parameter simulation and we force power at the input and measure the output response of the power sensor in terms of voltage. The test performed is a force power measure voltage. The following pseudocode is obtained from RADPro on testing for a RF sub-circuit using the power sensor technique as shown in Figure 32.

*Figure 32.*Pseudocode for RF subcircuit testing

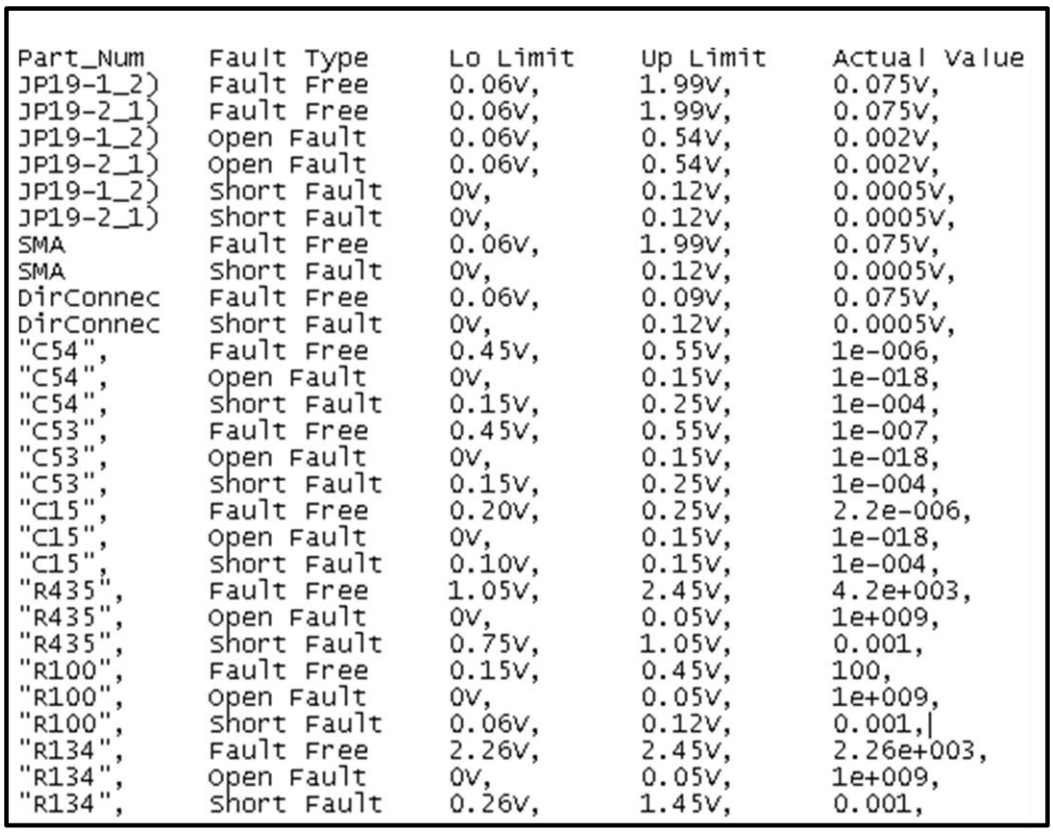
```
*****S-PARAMETER TEST FOR RF SUB-CIRCUIT*****
CONDITION(VERIFY COMPONENT = RFR1)
FPMV(forced power = -15dBm, force pin = J2_D1, sense pin = RFANTB_D1, type = embedded passive)
COMPARE_LIMITS(lower limit = 1.1 V, upper limit = 1.45 V, nominal value = 1.32 V, tested pwr node = RFANTB_OUT_D1)

*****S-PARAMETER TEST FOR RF SUB-CIRCUIT*****
CONDITION(VERIFY COMPONENT = RFB1)
FPMV(forced power = -30dBm, force pin = J3_D1, sense pin = LOP_D1, type = embedded passive)
COMPARE_LIMITS(lower limit = 2.1 V, upper limit = 2.54 V, nominal value = 2.32 V, tested pwr node = LO_INPUT_D1)

*****S-PARAMETER TEST FOR RF SUB-CIRCUIT*****
CONDITION(VERIFY COMPONENT = RFR2)
FPMV(forced power = -25dBm, force pin = J1_D1, sense pin = IFP_D1, type = embedded passive)
COMPARE_LIMITS(lower limit =1 .195 V, upper limit = 1.54 V, nominal value = 1.32 V, tested pwr node = IF_INPUT_D1)
```

The results from the pseudocode are stored in a look-up table. The Figure 33 shows the sample look-up table for DIB.

Figure 33. Look-up table.



Part_Num	Fault Type	Lo Limit	Up Limit	Actual Value
JP19-1_2)	Fault Free	0.06V,	1.99V,	0.075V,
JP19-2_1)	Fault Free	0.06V,	1.99V,	0.075V,
JP19-1_2)	Open Fault	0.06V,	0.54V,	0.002V,
JP19-2_1)	Open Fault	0.06V,	0.54V,	0.002V,
JP19-1_2)	Short Fault	0V,	0.12V,	0.0005V,
JP19-2_1)	Short Fault	0V,	0.12V,	0.0005V,
SMA	Fault Free	0.06V,	1.99V,	0.075V,
SMA	Short Fault	0V,	0.12V,	0.0005V,
DirConnec	Fault Free	0.06V,	0.09V,	0.075V,
DirConnec	Short Fault	0V,	0.12V,	0.0005V,
"C54",	Fault Free	0.45V,	0.55V,	1e-006,
"C54",	Open Fault	0V,	0.15V,	1e-018,
"C54",	Short Fault	0.15V,	0.25V,	1e-004,
"C53",	Fault Free	0.45V,	0.55V,	1e-007,
"C53",	Open Fault	0V,	0.15V,	1e-018,
"C53",	Short Fault	0.15V,	0.25V,	1e-004,
"C15",	Fault Free	0.20V,	0.25V,	2.2e-006,
"C15",	Open Fault	0V,	0.15V,	1e-018,
"C15",	Short Fault	0.10V,	0.15V,	1e-004,
"R435",	Fault Free	1.05V,	2.45V,	4.2e+003,
"R435",	Open Fault	0V,	0.05V,	1e+009,
"R435",	Short Fault	0.75V,	1.05V,	0.001,
"R100",	Fault Free	0.15V,	0.45V,	100,
"R100",	Open Fault	0V,	0.05V,	1e+009,
"R100",	Short Fault	0.06V,	0.12V,	0.001,
"R134",	Fault Free	2.26V,	2.45V,	2.26e+003,
"R134",	Open Fault	0V,	0.05V,	1e+009,
"R134",	Short Fault	0.26V,	1.45V,	0.001,

The results of the two different DIBs run on RADPro are available as a test report which indicates all the tested components and also the percentage of each component type tested along with the total board coverage.

TABLE 3. Test coverage statistics obtained for DIB using RADPro.

Components	Total Number of Components in BOM	Total Number of Testable Components	% of tested components	% of Board Coverage
Capacitors	193	174	90.15	84
Resistors	67	54	80.5	78
Relays for functionality	23	19	82.6	79
SMA Connectors	31	31	100	100
Diodes	56	45	80.35	76.5
Jumper Wires	27	23	85.18	83

The test coverage percentage depends on the accessibility provided by the board. Tests are run only for circuits that originate at a pogo pin and terminate at a DUT pin as these two pins are used to provide input stimulus and measure the output response. Also the total number of active components depends on the availability of their model in the model library which the users have to keep updating to increase the board coverage. The Table 4 provides the test coverage on an ETS board with differential testing capability.

TABLE 4. Summary of test coverage for ETS DIBs.

% Coverage	DIB 1(6486024C)	DIB 2(WG2K_ETS_364)
Total number of components	1174	1058
Capacitors	67	81
Resistors	52	74
Inductors	29	23
Relays	47	58
Overall Coverage	71.32	84.48

The above results indicate that the test coverage varies over a range of about 70% to 85%. This is due to accessibility that can be obtained on the DIB.

CHAPTER 5

CONCLUSION AND FUTURE WORK

A novel test methodology for verification of device interface boards used for IC testing has been presented in this thesis. This includes RF DIBs and also tester platforms with floating ground concept. The challenge of test generation by manual inspection of schematics has been addressed by RADPro which automatically generates test instructions from schematic information of the DIB. The test generation is independent of hardware tester platform as it has overcome the problem of floating ground used in differential testing in eagle tester platforms by automatically assigning the channel types to the pogo pins and classifying the high and low sides of the circuit as one single testable circuit.

There is a novel test methodology discussed in the thesis for testing embedded passive RF components. This involves the use of a power sensor at the DUT terminals to identify power loss in the sub-circuit to identify process related defects. The tool has overcome the lack of component information for embedded passive RF components by utilizing a user created text file holding the schematic information and parameter properties from the schematic files to automatically place the embedded passive RF components into the netlist and parts list. It also

serially induces faults in all testable passive components to create a fault dictionary containing output values which are matched to the ATE results to identify the occurrence of possible faults. The tool has been tested on mixed signal DIBs and ETS DIBs to generate pseudocodes which carry the test instructions to the ATE. RADPro reduces test development time from several weeks to days and reduces the time taken to market the IC.

Development efforts are in progress to obtain the pseudocode for a RF DIB and involve generation of test instructions compatible with the RF tester systems in Texas Instruments. Also work is in progress for developing bridge software which will translate the test instructions to the ATE and obtain the hardware results to match with the fault dictionary output values to identify faults in DIB components.

REFERENCES

1. K. Hird, K.P. Parker, and B. Follis; "Test coverage: what does it mean when a board test passes," *IEEE International Test Conference*, Page(s):1066 – 1074, 2002.

2. K.P. Parker, "The impact of boundary scan on board test," *ITC Design and Test of Computers* Volume 6, Issue 4, Page(s):18 – 30, Aug. 1989.

3. K. Pinjala, B. Kim, P. Variyam, "Automatic Diagnostic Program Generation for Mixed Signal Load Board," *IEEE International Test Conference*, pp. 403-409, 2005.

4. V. Kalyanaraman, B. Kim, P. Variyam, S. Cherubal, "DIBPro: Automatic Diagnostic Program Generation Tool," *IEEE International Test Conference*, 2006.

5. S. Sundar, B. Kim, "Circuit Analyzer and Test Generator (CATGEN) for Mixed-Signal Device Interface Boards," *IEEE VLSI Test Symposium*, San Diego, April 2008.

6. S. Sundar, B. Kim, S. Wokhlu, E. Beskar, R. Rousselin, T. Byrd, F. Toledo, D. Cotton, G. Kendall, "Low Cost Automatic Mixed-Signal Board Test Using IEEE 1149.4," *IEEE International Test Conference*, Santa Clara, CA, October 2007.

7. R. Schuttert, D.C.L. van Geest; "On-chip mixed-signal test structures re-used for board test," *IEEE International Test Conference*, pp. 375-383, 2004.

8. R.E. Tulloss, C.W. Yau, "BIST and boundary-scan for board level test: test program pseudo-code," *First IEEE European Test Conference*, 1989.

9. Sukeshwar Kannan, Bruce Kim, "Low Cost Test Technique for RF Interconnects in MCM Substrate", *IMAPS ATW on RF and Microwave Packaging*, San Diego, 2009.

10. Sukeshwar Kannan, Bruce Kim, Ganesh Srinivasan, Friedrich Taenzlar, Richard Antley, Chris Vogel and Craig Force, "Automatic Diagnostic Tool for Analog Mixed Signal and RF Load Boards", *IEEE International Test Conference*, Austin, TX, November 2009.

11. Sukeshwar Kannan, Bruce Kim, Ganesh Srinivasan, Friedrich Taenzlar, Richard Antley, Chris Vogel and Craig Force, "Development of Automatic Program Generation Tool for Analog-Mixed Signal and RF Load Boards", *IEEE International Workshop on Reliability Aware System Design and Test,* India January, 2010.

12. Sukeshwar Kannan, Bruce Kim, Ganesh Srinivasan, Friedrich Taenzlar, Richard Antley, Chris Vogel and Craig Force, "Automatic Diagnostic Tool for RF Device Interface Boards", *IEEE Wireless Test Workshop*, Santa Cruz, California, April 2010.
13. "Quantization and dither: A theoretical survey," Journal of Audio Engineering Society, vol. 40, no.5, pp. 355-375, May 1992.

14. P. Carbone, C. Narduzzi and D. Pedri, "Dither signal effects on the resolution of nonlinear quantizers," *IEEE Transactions on Instrumentation and Measurement*, vol. 43, no. 2, pp. 139-145, April 1994.

15. P.Carbone and D. Petri, "Effect of additive dither on the resolution of ideal quantizers," IEEE Transactions on Instrumentation and Measurement, vol. 43, no. 3, pp. 389-396, June 1994.

16. G. Srinivasan, S. Cherubal, P. Variyam, M. Teklu, C.P.Wang, D. Guidry, A. Chatterjee, " Accurate Measurement of Multi-Tone Power Ratio (MTPR) of ADSL Devices using Low Cost Testers", *IEEE European Test Symposium*, Estonia, May 2005.

17. R.A. Belcher, "Multi-tone testing of quantisers using time and frequency analysis", 4th Workshop on ADC Modeling and Testing, France, September 2009.

18. Stephen Boyd, "Multitone Signals with Low Crest Factor", *IEEE Transactions on Circuits and Systems, Vol. Cas-33, No. 10,* October 1986.

19. Mathias Friese, "Multitone Signals with Low Crest Factor", *IEEE Transactions on Communications, Vol. 45, No. 10,* October 1997.

20. S. Narahashi and T. Nojima, "New phasing scheme of N-multiple carriers for reducing peak-to-average power ratio," *Electronic Letters* vol. 30, pp. 1382–1383, Aug. 1994.

21. D. R. Gimlin and C. R. Patisaul, "On minimizing the peak-to-average power ratio for the sum of N sinusoids," *IEEE Transactions on Communications*, Vol. 41, pp. 631–635, Apr. 1993.

22. J. Pumplin, "Low-noise noise," *Journal of Acoustical Society of America,* vol. 78, pp. 100–104, July 1985.

23. S. Narahashi, K. Kumagai, and T. Nojima, "Minimizing peak-to average power ratio of multitone signals using steepest descent method," *Electronics Letters*, vol. 31, pp. 1552–1554, Aug. 1995.

24. P. Guillaume, J. Schoukens, R. Pintelon, and I. Koll´ar, "Crest-factor minimization using nonlinear Chebyshev approximation methods," *IEEE Transactions on Instrumentation and Measurement,* vol. 40, pp. 982–989, Dec. 1991.

APPENDIX A

Model File Generation for ETS DIBs

1. Model name:

   Model name of the component as it appears in the Bill of Materials

2. Power Pins: PWR <Pin numbers separated by commas>

   If there is no power nodes use “NA”.

3. Terminals of the component: PINS<list of inputs and outputs separated by commas>

   This is a list of one set of input and output separated by “|”.

4. Disregarded nodes: DISREGARD <Pin numbers separated by commas>

   Pins that is not included in any of the above categories.

5. Component Reference Designator: COMP<Ref Designator>

   The following is the list of reference designators used for different components.

   a. Relay: K
   b. Resistor: R
   c. Capacitor: C
   d. Inductor: L
   e. IC: U
   f. Diode/ LED: D
   g. Jumper/ SMA: J
   h. Transistor: Q
   i. RF Resistive Trace: RFR
   j. RF Balun Transformer: RFB
   k. RF Passive Filter: RFP

Model file example for an ETS DIB

```
08051C103JAT2A PWR<NA> PINS<1,2> DISREGARD<NA> COMP<C>
08055A150JAT2A PWR<NA> PINS<1,2> DISREGARD<NA> COMP<C>
08055C104JAT2A PWR<NA> PINS<1,2> DISREGARD<NA> COMP<C>
0805YA103JAT2A PWR<NA> PINS<1,2> DISREGARD<NA> COMP<C>
0805ZC105KAT2A PWR<NA> PINS<1,2> DISREGARD<NA> COMP<C>
12101C104KAT2A PWR<NA> PINS<1,2> DISREGARD<NA> COMP<C>
ECE-V2AA4R7UP PWR<NA> PINS<1,2> DISREGARD<NA> COMP<C>
T491D225K050AS PWR<NA> PINS<1,2> DISREGARD<NA> COMP<C>
199D104X9050AA1 PWR<NA> PINS<1,2> DISREGARD<NA> COMP<C>
1N914 PWR<NA> PINS<1,2> DISREGARD<NA> COMP<D>
1N5819 PWR<NA> PINS<1,2> DISREGARD<NA> COMP<D>
STPS1H100A PWR<NA> PINS<1,2> DISREGARD<NA> COMP<D>
SN74AS1034AD PWR<14,7> PINS<1,2|3,4|5,6|9,8|11,10|13,12> DISREGARD<NA> COMP<U>
IRF1607 PWR<NA> PINS<1,2,3> DISREGARD<NA> COMP<Q>
AD8561AN PWR<1,6|4,6> PINS<2,3,7|2,3,8> DISREGARD<5> COMP<U>
CAT24C04LI-G PWR<8,4> PINS<1,2,3,6,7,5> DISREGARD<NA> COMP<U>
OP27EZ PWR<7,4> PINS<1,2,6> DISREGARD<1,8,5> COMP<U>
JUMPER PWR<NA> PINS<1,2> DISREGARD<NA> COMP<J>
SML-LX1206GC-TR PWR<NA> PINS<1,2> DISREGARD<NA> COMP<D>
FBR51ND12-W1 PWR<NA> PINS<1,3> DISREGARD<4,2,5> COMP<K>
2911-12-321  PWR<NA> PINS<1,4|8,5> DISREGARD<2,3,6,7> COMP<K>
CRCW12062490F PWR<NA> PINS<1,2> DISREGARD<NA> COMP<R>
TNPW12061503BT PWR<NA> PINS<1,2> DISREGARD<NA> COMP<R>
CRCW12101002F PWR<NA> PINS<1,2> DISREGARD<NA> COMP<R>
CRCW12104991F PWR<NA> PINS<1,2> DISREGARD<NA> COMP<R>
ERA-6YEB102V PWR<NA> PINS<1,2> DISREGARD<NA> COMP<R>
ERA-6YEB104V PWR<NA> PINS<1,2> DISREGARD<NA> COMP<R>
ERA-6YEB392V PWR<NA> PINS<1,2> DISREGARD<NA> COMP<R>
MP9100-50 PWR<NA> PINS<1,2> DISREGARD<NA> COMP<R>
RES0805 PWR<NA> PINS<1,2> DISREGARD<NA> COMP<R>
AD8052ARM PWR<4,15|8,-15> PINS<3,2,1|5,6,7> DISREGARD<NA> COMP<U_T>
AD8605ART PWR<1,2> PINS<4,3,1> DISREGARD<5,7> COMP<U_T>
THS4012ID PWR<8,15|4,12> PINS<2,3,1|6,5,7> DISREGARD<NA> COMP<U_T>
2332-12-020 PWR<NA> PINS<1,4|8,5> DISREGARD<1,2,4,7,8> COMP<K>
AGN200A4H PWR<1,2> PINS<3,2,4|6,7,5> DISREGARD<2,3> COMP<U>
AD8051ART PWR<1> PINS<4,3,1> DISREGARD<3> COMP<U>
AD9631AR PWR<1> PINS<2,3,6> DISREGARD<3> COMP<U>
OP4177ARU PWR<1> PINS<2,3,1|6,5,7|13,12,14|9,10,8> DISREGARD<3> COMP<U>
RF303YZ-12 PWR<1> PINS<2,3,4|8,6,7> DISREGARD<3> COMP<K>
9852-05-00 PWR<NA> PINS<6,3,5> DISREGARD<NA> COMP<K>
SN74LVC1G125DBV PWR<NA> PINS<2,4> DISREGARD<NA> COMP<U>
SN74LVC1G32DBV PWR<NA> PINS<1,2,4> DISREGARD<NA> COMP<U>
SN75468D PWR<NA> PINS<1,2,3,4,5,6,7,10,11,12,13,14,15,16> DISREGARD<NA> COMP<U_NT>
AGN200A12 PWR<1> PINS<3,2,4|6,7,5> DISREGARD<3> COMP<U>
RC1206USERDEF PWR<1> PINS<1000000000> DISREGARD<NA> COMP<R>
TC32-1002BXT1 PWR<10> PINS<10000> DISREGARD<NA> COMP<R>
S102K1K200.01% PWR<1> PINS<10000> DISREGARD<NA> COMP<R>
RES-SMT0402_UNINSTALLED PWR<1> PINS<19.99> COMP<R>
RES0402(UN) PWR<21> PINS<1G> COMP<R>
CAP-SMT1206-UNINSTALLED PWR<12> PINS<100000> COMP<C>
CAP-SMT0402-UNINSTALLED PWR<12> PINS<1p> DISREGARD<2> COMP<C>
```

APPENDIX B

Input Files for ETS DIB

Parts List and Netlist

PARTS LIST

08051C103JAT2A 08051C103JAT2A C412 C115 C107
08051C103JAT2A 08051C103JAT2A C109 C116 C207
08051C103JAT2A 08051C103JAT2A C209 C212 C112
08051C103JAT2A 08051C103JAT2A C415 C216 C409
08051C103JAT2A 08051C103JAT2A C407 C416 C316
08051C103JAT2A 08051C103JAT2A C307 C309 C315
08051C103JAT2A 08051C103JAT2A C312 C215

08055A150JAT2A 08055A150JAT2A C417 C317 C217
08055A150JAT2A 08055A150JAT2A C117

08055C104JAT2A 08055C104JAT2A C401 C301 C201
08055C104JAT2A 08055C104JAT2A C101

0805YA103JAT2A 0805YA103JAT2A C306 C106 C406
0805YA103JAT2A 0805YA103JAT2A C206

0805ZC105KAT2A 0805ZC105KAT2A C211 C108 C111
0805ZC105KAT2A 0805ZC105KAT2A C213 C110 C114
0805ZC105KAT2A 0805ZC105KAT2A C208 C410 C313
0805ZC105KAT2A 0805ZC105KAT2A C210 C411 C308
0805ZC105KAT2A 0805ZC105KAT2A C310 C113 C311
0805ZC105KAT2A 0805ZC105KAT2A C214 C408 C314
0805ZC105KAT2A 0805ZC105KAT2A C414 C413

12101C104KAT2A 12101C104KAT2A C405 C205 C105
12101C104KAT2A 12101C104KAT2A C305

199D104X9050AA1 199D104X9050AA1 C99

1N5819 1N5819 SD103 SD203 SD303
1N5819 1N5819 SD403

1N914 1N914 D209 D211 D207
1N914 1N914 D206 D204 D216
1N914 1N914 D202 D100 D201
1N914 1N914 D200 D101 D210
1N914 1N914 D408 D203 D110
1N914 1N914 D102 D306 D304
1N914 1N914 D416 D112 D118
1N914 1N914 D104 D106 D413
1N914 1N914 D412 D117 D113
1N914 1N914 D116 D111

2911-12-321 2911-12-321 K_CAL2_S3K_CAL1_S3K_RT_S3
2911-12-321 2911-12-321 K_MPUS2PH_S3K_CAL4_S3K_CP2SRV_S3
2911-12-321 2911-12-321 K_CAL3_S3K_CAL2_S3K_CAL3_S1
2911-12-321 2911-12-321 K_CP2SRV_S3K_CP2VS_S3K_SRV1X_S4
2911-12-321 2911-12-321 K_CP2VS_S3K_CAL3_S3K_SRVCAL_S3
2911-12-321 2911-12-321 K_CAL2_S1K_DIV_S3 K_SRVCAL_S4
AD8561AN AD8561AN U302 U402 U202
AD8561AN AD8561AN U102

B66.20.06.00.19 B66.20.06.00.19 SITE2 SITE3 SITE1
B66.20.06.00.19 B66.20.06.00.19 SITE4

CAT24C04LI-G CAT24C04LI-G U99

CRCW06034021F CRCW06034021F R235 R435 R135
CRCW06034021F CRCW06034021F R335

CRCW0805000Z CRCW0805000Z R331 R328 R128
CRCW0805000Z CRCW0805000Z R219 R217 R216
CRCW0805000Z CRCW0805000Z R215 R214 R231
CRCW0805000Z CRCW0805000Z R101 R301 R114
CRCW0805000Z CRCW0805000Z R228 R119 R131
CRCW0805000Z CRCW0805000Z R117 R116 R115
CRCW0805000Z CRCW0805000Z R201 R417 R319
CRCW0805000Z CRCW0805000Z R401 R431 R428
CRCW0805000Z CRCW0805000Z R317 R316 R315
CRCW0805000Z CRCW0805000Z R414 R415 R416
CRCW0805000Z CRCW0805000Z R419 R314

CRCW08051000F CRCW08051000F R312 R412 R200
CRCW08051000F CRCW08051000F R100 R300 R112
CRCW08051000F CRCW08051000F R400 R212

CRCW080510R0F CRCW080510R0F R438 R440 R439
CRCW080510R0F CRCW080510R0F R437

CRCW08051502F CRCW08051502F R403 R203 R303
CRCW08051502F CRCW08051502F R103

CRCW080540R2F CRCW080540R2F R118 R318 R418
CRCW080540R2F CRCW080540R2F R218

CRCW12061002F CRCW12061002F R320 R420 R120
CRCW12061002F CRCW12061002F R220

CRCW120610R0F CRCW120610R0F R324 R224 R124
CRCW120610R0F CRCW120610R0F R424

CRCW12062211F CRCW12062211F R98 R99

CRCW12065111F CRCW12065111F R311 R411 R111
CRCW12065111F CRCW12065111F R211

CRCW12066490F CRCW12066490F R91

CRCW12067500F CRCW12067500F R93 R90

CRCW12101002F CRCW12101002F R108 R308 R408
CRCW12101002F CRCW12101002F R208

CRCW12104991F CRCW12104991F R107 R407 R307
CRCW12104991F CRCW12104991F R207

ECE-V2AA4R7UP ECE-V2AA4R7UP C204 C304 C104
ECE-V2AA4R7UP ECE-V2AA4R7UP C404

ERA-6YEB102V ERA-6YEB102V R329 R229 R129
ERA-6YEB102V ERA-6YEB102V R429

ERA-6YEB104V ERA-6YEB104V R204 R404 R304
ERA-6YEB104V ERA-6YEB104V R104

ERA-6YEB392V ERA-6YEB392V R432 R132 R332
ERA-6YEB392V ERA-6YEB392V R232

ETS600-POGO_143P ETS600-POGO_143P P111 P106 P111
ETS600-POGO_143P ETS600-POGO_143P P106 P114 P106
ETS600-POGO_143P ETS600-POGO_143P P114 P106 P114
ETS600-POGO_143P ETS600-POGO_143P P106 P111 P114
ETS600-POGO_143P ETS600-POGO_143P P111 P106 P114
ETS600-POGO_143P ETS600-POGO_143P P111 P114 P111
ETS600-POGO_143P ETS600-POGO_143P P214 P103 P214
ETS600-POGO_143P ETS600-POGO_143P P103 P214 P106
ETS600-POGO_143P ETS600-POGO_143P P214 P103 P214
ETS600-POGO_143P ETS600-POGO_143P P103 P214 P103
ETS600-POGO_143P ETS600-POGO_143P P214 P103 P214
ETS600-POGO_143P ETS600-POGO_143P P103 P214 P103

ETS600-POGO_143P ETS600-POGO_143P P214 P106 P214
ETS600-POGO_143P ETS600-POGO_143P P106 P114 P106
ETS600-POGO_143P ETS600-POGO_143P P114 P106 P103
ETS600-POGO_143P ETS600-POGO_143P P106 P214 P106
ETS600-POGO_143P ETS600-POGO_143P P103 P106 P103
ETS600-POGO_143P ETS600-POGO_143P P214 P103 P214
ETS600-POGO_143P ETS600-POGO_143P P206 P214 P206

NET LIST

NODENAME 12V $
K_SRV1X_S1 1 D111 1 D105 1 K_CAL4_S1 1 $
D112 1 D405 1 K_CAL2_S1 1 $
D402 1 K_SRV1X_S4 1 K_CAL1_S4 1 $
K_RT_S4 1 K_RT_S2 1 K_DIV_S1 1 $
D113 1 D202 1 K_CAL3_S1 1 $
D116 1 P114 B5 D403 1 $
C91 1 D401 1 K_CBULK_S4 1 $
K_CBULK_S2 1 K_BOOT_S2 1 K_CAL3_S4 1 $
D416 1 D108 1 K_CAL2_S4 1 $
K_SRVCAL_S1 1 D411 1 D210 1 $
D200 1 K_DIV_S4 1 D409 1 $
D207 1 D203 1 D404 1 $
K_CP2SRV_S4 1 D407 1 K_MPUS2PH_S4 1 $
D408 1 D412 1 D201 1 $
K_SRVCAL_S4 1 K_CP2VS_S4 1 D406 1 $
K_CAL4_S4 1 K_EN2G_S4 1 D413 1 $
D217 1 D117 1 D318 1 $
K_CP2SRV_S3 1 D309 1 K_MPUS2PH_S3 1 $
K_SP2BT_S3 1 D317 1 D311 1 $
K_CAL2_S3 1 K_CAL4_S2 1 K_SP2PH_S2 1 $
D307 1 K_SP2BT_S2 1 R91 1 $
D316 1 K_CAL3_S3 1 D212 1 $
D313 1 K_DIV_S3 1 K_SP2PH_S1 1 $
K_MPUF2PH_S3 5 K_DIV_S2 1 K_BOOT_S4 1 $
K_SP2BT_S1 1 K_CAL4_S3 1 D218 1 $
K_SRV1X_S2 1 D208 1 D107 1 $
K_MPUF2PH_S4 5 K_MPUF2PH_S2 5 K_MPUF2PH_S1 5 $
D310 1 D300 1 K_BOOT_S3 1 $
K_SRVCAL_S2 1 D301 1 K_CBULK_S3 1 $
K_SP2PH_S3 1 K_RT_S3 1 D312 1 $
D205 1 D303 1 K_SP2PH_S4 1 $
D418 1 K_CAL1_S3 1 K_SP2BT_S4 1 $
D417 1 D304 1 D306 1 $
K_CP2VS_S3 1 K_EN2G_S2 1 D302 1 $
D204 1 K_RT_S1 1 D103 1 $
D213 1 K_CP2VS_S2 1 D118 1 $
D104 1 D106 1 K_CP2VS_S1 1 $
D102 1 P103 B5 K_CAL1_S1 1 $
K_CP2SRV_S1 1 P111 B5 K_CAL1_S2 1 $
D410 1 D109 1 K_MPUS2PH_S1 1 $
P106 B5 D400 1 D206 1 $
K_MPUS2PH_S2 1 D110 1 K_EN2G_S3 1 $
K_CAL3_S2 1 D305 1 K_SRV1X_S3 1 $
D216 1 K_SRVCAL_S3 1 K_CAL2_S2 1 $
D308 1 D211 1 D209 1 $
K_CP2SRV_S2 1 K_CBULK_S1 1 D101 1 $
D100 1 K_BOOT_S1 1 K_EN2G_S1 1
NODENAME 15V $
C90 1 R222 1 P103 G3 R122 1 $
R422 1 P103 G4 P103 F2 $
P103 F3 R322 1 R90 1
NODENAME 24V $
P103 H3 P103 I4 P103 H2 P103 I3 $
C94 1
NODENAME 5V $
U203 14 C214 1 C215 1 U303 14 $
C314 1 C315 1 R440 1 $
R439 1 C115 1 P111 B9 $
R437 1 P106 B9 U99 8 $

P114 B7 R92 1 P111 B7 $
C114 1 P106 B7 U103 14 $
U403 14 C414 1 C415 1 $
R438 1 C92_4 1 P103 B9 $
P114 B9 R99 1 C92_3 1 $
C92_1 1 P103 B7 C99 1 $
C92_2 1 R98 1
NODENAME 12V1 $
P103 C3 P103 C4 C97 2
NODENAME 15V1 $
R223 1 P103 E3 P103 D2 R123 1 $
R323 1 C93 2 P103 E4 $
R423 1 R93 1 P103 D3
NODENAME 24V1 $
P103 J2 C95 2 P103 J3
NODENAME 52V $
P106 B11 P114 B11 P103 B11 P111 B11 $
C96 2
NODENAME AGND1 $
C105 2 SITE1 9F SITE1 9S P113 D10 $
P113 C3 P113 D3 R103 1 $
P113 C10 P113 E7 P113 F7 $
J101 1 SD102 2 C104 2
NODENAME AGND2 $
C205 2 P112 F9 C204 2 R203 1 $
P112 E9 J201 1 P112 C12 $
SD202 2 P112 D12 P112 D5 $
P112 C5 SITE2 9S SITE2 9F
NODENAME AGND3 $
P105 F7 P105 C3 P105 D3 P105 C10 $
P105 D10 R303 1 C304 2 $
C305 2 P105 E7 SD302 2 $
SITE3 9S J301 1 SITE3 9F
NODENAME AGND4 $
P104 F9 P104 D12 SD402 2 P104 E9 $
C405 2 P104 C5 P104 D5 $
P104 C12 SITE4 9S R403 1 $
J401 1 C404 2 SITE4 9F
NODENAME BOOTS1 $
SITE1 1S SITE1 1F SD101 1 R101 1 $
K_BOOT_S1 5
NODENAME BOOTS2 $
SITE2 1S SITE2 1F K_BOOT_S2 5 SD201 1 $
R201 1
NODENAME BOOTS3 $
K_BOOT_S3 5 SD301 1 SITE3 1S SITE3 1F $
R301 1
NODENAME BOOTS4 $
SITE4 1F SD401 1 SITE4 1S R401 1 $
K_BOOT_S4 5
NODENAME BTSRCS1 $
K_SP2BT_S1 5 K_BOOT_S1 8 P113 H2 P113 G2 $
SD101 2
NODENAME BTSRCS2 $
SD201 2 K_SP2BT_S2 5 P112 G7 P112 H7 $
K_BOOT_S2 8
NODENAME BTSRCS3 $
P105 G2 K_SP2BT_S3 5 P105 H2 SD301 2 $
K_BOOT_S3 8
NODENAME BTSRCS4 $
K_BOOT_S4 8 P104 G7 P104 H7 K_SP2BT_S4 5 $
SD401 2
NODENAME CALS1 $
K_CAL1_S1 5 K_CAL2_S1 8
NODENAME CALS2 $
K_CAL2_S2 8 K_CAL1_S2 5
NODENAME CALS3 $
K_CAL2_S3 8 K_CAL1_S3 5
NODENAME CALS4 $
K_CAL2_S4 8 K_CAL1_S4 5

Bill of Materials

ITEM	QTY	MFG	MFG PART#	REF DES	DESCRIPTION	VALUE or FUNCTION
	MILLENNIUM (TKY)					
REF	NONE REQUESTED	6486024B	-	ASSEMBLY	-	
REF	-	6486024C	-	SCHEMATIC	-	
1	NONE REQUESTED	6486024B	-	FABRICATION	-	
REF	-	6486024B	-	ARTWORK	-	
1	1	NEWARK	6437988	"HARDWARE KIT, EAGLE ETS-364/ETS-600 HIB"		
2	20	AVX	08051C103JAT2A	"C107, C109, C112, C115, C116, C207, C209, C212, C215, C216, C307, C309, C312, C315, C316, C407, C409, C412, C415, C416"	"CAP,SMT,0805"	"CAPACITOR,SMT,0805,CERAMIC,0.01uF,100V,5%,X7R"
3	4	AVX	08055A150JAT2A	"C117, C217, C317, C417"	"CAP,SMT,0805"	"CAPACITOR,SMT,0805,CERAMIC,50V,5%, 15pF,COG(NPO)"
4	4	AVX	08055C104JAT2A	"C101, C201, C301, C401"	"CAP,SMT,0805"	"CAPACITOR,SMT,0805,CERAMIC,0.1uF,50v,5%,X7R"
5	4	AVX	0805YA103JAT2A	"C106, C206, C306, C406"	"CAP,SMT,0805"	"CAPACITOR,SMT,0805,CERAMIC,16V,5%, 0.01uF,COG(NPO)"
6	20	AVX	0805ZC105KAT2A	"C108, C110, C111, C113, C114, C208, C210, C211, C213, C214, C308, C310, C311, C313, C314, C408, C410, C411, C413, C414"	"CAP,SMT,0805"	"CAPACITOR,SMT,0805,CERAMIC,1.0uF,10V,10%,X7R"
7	4	AVX	12101C104KAT2A	"C105, C205, C305, C405"	"CAP,SMT,1210"	"CER,CAP,SMT,0.1uF,100nF,100V,10%,X7R"
8	4	PANASONIC	ECE-V2AA4R7UP	"C104, C204, C304, C404"	"CAP,SMT,ELEC"	"CAP,SMT,ELE,RAD,VS SERIES,20%,100V, 4.7uF"
9	11	KEMET	T491D225K050AS	"C90, C91, C93, C94, C95, C96, C97, C92_1, C92_2, C92_3, C92_4"	"CAP,SMT,TAN,CASE-D"	"CAP,SMT,TAN,2.2uF,50V,10%,ESR-2.5ohm@100K"
10	1	VISHAY / SPRAGUE	199D104X9050AA1	C99	"CAPACITOR,THU,2P"	"CAPACITOR,THU,2P,TANTALUM,0.1uF,50V,10%"
11	4	MULTITEST	B66.20.06.00.19	"SITE1, SITE2, SITE3, SITE4"	CONTACTOR	CUSTOMER INSTALL
12	68		1N914	"D100, D101, D102, D103, D104, D105, D106, D107, D108, D109, D110, D111, D112, D113, D116, D117, D118, D200, D201, D202, D203, D204, D205, D206, D207, D208, D209, D210, D211, D212, D213, D216, D217, D218, D300, D301, D302, D303, D304, D305, D306, D307,"	"DIODE,1N914"	"DIODE,1N914"	
				"D308, D309, D310, D311, D312, D313, D316, D317, D318, D400, D401, D402, D403, D404, D405, D406, D407, D408, D409, D410, D411, D412, D413, D416, D417, D418"		
13	8	ST MICRO	STPS1H100A	"SD101, SD102, SD201, SD202, SD301, SD302, SD401, SD402"	"DIODE,SMT,SMA"	"SCHOTTKY RECTIFIERS,SMT,100V,1A"
14	4	DIODES INC	1N5819	"SD103, SD203, SD303, SD403"	"DIODE,THU,2P"	SCHOTTKEY DIODE
15	4	TI	SN74AS1034AD	"U103, U203, U303, U403"	"IC,SMT,SOIC-14N"	"HEX DRIVERS, RoHS"
16	8	INTERNATIONAL RECTIFIER	IRF1607	"M101, M102, M201, M202, M301, M302, M401, M402"	"IC,THU,3P,TO220AB"	"POWER MOSFET,N-CHAN,75V,142A,0.0075 OHM,380W"

17 4 ANALOG DEVICES AD8561AN "U102, U202, U302, U402" "IC,THU,DIP-8" ULTRAFAST 7ns SINGLE SUPPLY COMPARATOR

18 1 CATALYST SEMICONDUCTOR CAT24C04LI-G U99 "IC,THU,DIP-8" "4-Kb CMOS SERIAL EEPROM,1.8~5.5V,DIP-8,NiPdAu"

19 4 ANALOG OP27EZ "U101, U201, U301, U401" "IC,THU,DIP-8W,300MIL" "LOW NOISE, HIGH SPEED PREC. OPAMP"

20 4 ANY wire jumper "J101, J201, J301, J401" JUMPER INSTALLED ANY BRAND 2PIN THU JUMPER

21 4 LUMEX SML-LX1206GC-TR "LED1, LED2, LED3, LED4" "LED,SMT" "GREEN,SMT,20mA"

22 4 FUJITSU FBR51ND12-W1 "K_MPUF2PH_S1, K_MPUF2PH_S2, K_MPUF2PH_S3, K_MPUF2PH_S4" "RELAY,THU,5P" "RELAY,THU,5P,EMR,SPDT,1FC,12V,25A,240R COIL,W1 OPTION"

23 64 *COTO TECHNOLOGY 2911-12-321 "K_CP2SRV_S1, K_CP2SRV_S2, K_CP2SRV_S3, K_CP2SRV_S4, K_CP2VS_S1, K_CP2VS_S2, K_CP2VS_S3, K_CP2VS_S4, K_CBULK_S1, K_CBULK_S2, K_CBULK_S3, K_CBULK_S4, K_RT_S1, K_RT_S2, K_RT_S3, K_RT_S4, K_CAL1_S1, K_CAL1_S2, K_CAL1_S3, K_CAL1_S4, K_CAL3_S1, K_CAL3_S2," "RELAY,THU,7P" "RELAY,THU,7P,RR,SPDT,1FC,12V,0.25A,1.5K COIL,COAXIAL SHIELD"

"K_CAL3_S3, K_CAL3_S4, K_CAL2_S1, K_CAL2_S2, K_CAL2_S3, K_CAL2_S4, K_BOOT_S1, K_BOOT_S2, K_BOOT_S3, K_BOOT_S4, K_MPUS2PH_S1, K_MPUS2PH_S2, K_MPUS2PH_S3, K_MPUS2PH_S4, K_CAL4_S1, K_CAL4_S2, K_CAL4_S3, K_CAL4_S4, K_DIV_S1, K_DIV_S2, K_DIV_S3, K_DIV_S4,"

"K_EN2G_S1, K_EN2G_S2, K_EN2G_S3, K_EN2G_S4, K_SRV1X_S1, K_SRV1X_S2, K_SRV1X_S3, K_SRV1X_S4, K_SRVCAL_S1, K_SRVCAL_S2, K_SRVCAL_S3, K_SRVCAL_S4, K_SP2BT_S1, K_SP2BT_S2, K_SP2BT_S3, K_SP2BT_S4, K_SP2PH_S1, K_SP2PH_S2, K_SP2PH_S3, K_SP2PH_S4"

24 4 VISHAY CRCW06034021F "R135, R235, R335, R435" "RES,SMT,0603" "RESISTOR,SMT,0603,1%,1/10W,4.02K"

25 32 VISHAY CRCW0805000Z "R101, R114, R115, R116, R117, R119, R128, R131, R201, R214, R215, R216, R217, R219, R228, R231, R301, R314, R315, R316, R317, R319, R328, R331, R401, R414, R415, R416, R417, R419, R428, R431" "RES,SMT,0805" "RESISTOR,SMT,0805,THICK FILM,0 OHM,1/8W"

26 8 VISHAY CRCW08051000F "R100, R112, R200, R212, R300, R312, R400, R412" "RES,SMT,0805" "RESISTER,SMT,0805,THICK FILM,1%,1/8W,100 OHM"

27 4 VISHAY CRCW080510R0F "R437, R438, R439, R440" "RES,SMT,0805" "RESISTER,SMT,0805,THICK FILM,1%,1/8W,10.0 OHM"

28 4 VISHAY CRCW08051502F "R103, R203, R303, R403" "RES,SMT,0805" "RESISTER,SMT,0805,THICK FILM,1%,1/8W,15.0K"

29 4 VISHAY CRCW080540R2F "R118, R218, R318, R418" "RES,SMT,0805" "RESISTER,SMT,0805,THICK FILM,1%,1/8W,40.2 OHM"

30 4 VISHAY CRCW12061002F "R120, R220, R320, R420" "RES,SMT,1206" "RESISTOR,SMT,1206,10K,1%,1/4W"

31 4 VISHAY CRCW120610R0F "R124, R224, R324, R424" "RES,SMT,1206" "RESISTOR,SMT,1206,10 OHM,1%,1/4W"

32 2 VISHAY CRCW12062211F "R98, R99" "RES,SMT,1206" "RESISTOR,SMT,1206,1%,1/4W,2.21K"

33 8 VISHAY CRCW12062261F "R126, R134, R226, R234, R326, R334, R426, R434" "RES,SMT,1206" "RESISTOR,SMT,1206,1%,1/4W,2.26K"

34 1 VISHAY CRCW12062490F R92 "RES,SMT,1206" "RESISTOR,SMT,1206,1%,1/4W,249 OHM"

35 8 VISHAY CRCW12064990F "R109, R110, R209, R210, R309, R310, R409, R410" "RES,SMT,1206" "RESISTOR,SMT,1206,1%,1/4W,499 OHM"

36 16 VISHAY CRCW120649R9F "R106, R122, R123, R127, R206, R222, R223, R227, R306, R322, R323, R327, R406, R422, R423, R427" "RES,SMT,1206" "RESISTOR,SMT,1206,49.9 OHM,1%,1/4W"

37 8 VISHAY CRCW120649R9FK "R105, R113, R205, R213, R305, R313, R405, R413" "RES,SMT,1206" "THICK FILM RESIST0R,SMT,1206,49.9 OHM,1%,1/4W,100ppm"

38 4 VISHAY CRCW12065111F "R111, R211, R311, R411" "RES,SMT,1206" "RESISTOR,SMT,1206,1%,1/4W,5.11K"

39 1 VISHAY CRCW12066490F R91 "RES,SMT,1206" "RESISTOR,SMT,1206,1%,1/4W,649 OHM"

40 2 VISHAY CRCW12067500F "R90, R93" "RES,SMT,1206" "RESISTOR,SMT,1206,1%,1/4W,750 OHM"

41 4 VISHAY TNPW12061503BT9 "R121, R221, R321, R421" "RES,SMT,1206" "RESISTOR,SMT,1206,150K,0.1%,1/8W,25ppm,T9"

42 4 VISHAY CRCW12101002F "R108, R208, R308, R408" "RES,SMT,1210" "RESISTOR,SMT,1210,1%,1/3W,10.0K OHM"

43 4 VISHAY CRCW12104991F "R107, R207, R307, R407" "RES,SMT,1210" "RESISTOR,SMT,1210,1%,1/3W,4.99K OHM"

44 4 PANASONIC ERA-6YEB102V "R129, R229, R329, R429" "RES,SMT,2P" "RESISTOR,SMT,0805,1.0K,0.1%,1/10W,25ppm"

45 4 PANASONIC ERA-6YEB104V "R104, R204, R304, R404" "RES,SMT,2P" "RESISTOR,SMT,0805,100K,0.1%,1/10W,25ppm"

46 4 PANASONIC ERA-6YEB392V "R132, R232, R332, R432" "RES,SMT,2P" "RESISTOR,SMT,0805,3.9K,0.1%,1/10W,25ppm"

47 8 CADDOCK MP9100-50.0-1% "R125, R133, R225, R233, R325, R333, R425, R433" "RESISTOR,THU,TO247-2" "RESISTOR,THU,TO247-2,POWER FILM,50.0 OHM,1%,100W"

48 4 NOT INSTALLED RES0805(UN) "R130, R230, R330, R430" "UNINSTALLED RES,SMT,0805" "UNINSTALLED,RESISTOR,SMT, 0805"

49 16 PEM KFS2-M2.5 SITE1-SITE4 PEM NUTS INSTALL FIRST (MANUALLY CALCULATE THE QTY)

50 16 HEAVY METAL 6447013 SITE1-SITE4 "QFN THUMSCREWS, MT9918" (MANUALLY CALCULATE THE QTY)

51 520 TYCO 147444-1 "K_BOOT_S1-K_BOOT_S4, K_CAL1_S1-K_CAL1_S4, K_CAL2_S1-K_CAL2_S4, K_CAL3_S1-K_CAL3_S4, K_CAL4_S1-K_CAL4_S4, K_CBULK_S1-K_CBULK_S4, K_CP2VS_S1-K_CP2VS_S4, K_CP2SRV_S1-K_CP2SRV_S4, K_DIV_S1-K_DIV_S4, K_EN2G_S1-K_EN2G_S4, K_MPUS2PH_S1-K_MPUS2PH_S4," SOCKET PIN - INSTALL 1ST "DIA_038, PIN_013-020, EXP_208, B187"

"K_RT_S1-K_RT_S4, K_SP2BT_S1-K_SP2BT_S4, K_SP2PH_S1-K_SP2PH_S4, K_SRV1X_S1-K_SRV1X_S4, K_SRVCAL_S1-K_SRVCAL_S4, U99, U101, U102, U201, U202, U301, U302, U401, U402"

52 20 MILL-MAX 0355-0-15-01-02-27-10-0 K_MPUF2PH_S1-K_MPUF2PH_S4 SOCKET PIN - INSTALL 1ST "DIA_100, PIN_040-050, EXP_274"

SPECIAL NOTES AND INSTRUCTIONS

1. OK to subsitute Fijitsu FBR51-ND12-W1 for FBR51-ND12-W

"2. If FBR51-ND12-W1 is loose fitting in MIL-MAX 355 socket pin, apply solder to FB51-ND12_W1 socket pins to tighten fit"

END OF BOM

Non-Testable List

SN74AS1034AD
AD8561AN
CAT24C04LI-G
OP27EZ
CASTLE_1.1IN_2COL
KA330/254EEFB21TAH
2506-2-00-50-00-00-07-0
OPC-125-G-D-A
CASTLE_179GHH
SMTPOGO3
1929173
VLCTPOGO8
TSW-107-07-G-S
ADT1-1WT
TEST POINT BLACK
CASTLE_208GGW
69190-403
H1102
ECX-5564-35.328M
CASTLE_324GDW
SN74CB3Q3257DGVR
SN75469D
IRLR8203
HZ_0603_A_102_R
CASTLE_2.4IN_2COL
ADS5410IPFB
1PS74SB43
THS5651AIPW

Pinmap File

REF	PIN	VALUE	Channel Type
P104	C4	NC	No Connection
P104	C5	APU48-55SL	Ground
P104	C6	APU54S	Sense
P104	C7	APU50S	Sense
P104	D1	MPU88FLA	Force Low A
P104	D10	NC	No Connection
P104	D11	NC	No Connection
P104	D12	APU56-63FL	Ground
P104	D13	APU62F	Force
P104	D14	APU58F	Force
P104	D2	MPU88FHA	Force High A
P104	D3	MPUFH88	ETSNC
P104	D4	NC	No Connection
P104	D5	APU48-55FL	Ground
P104	D6	APU54F	Force
P104	D7	APU50F	Force
P104	E1	MPU88FLA	Force Low A
P104	E10	NC	No Connection
P104	E11	BBUS56-63S	ETSNC
P104	E12	NC	No Connection
P104	E13	APU61S	Sense
P104	E14	APU57S	Sense
P104	E2	MPU88FHA	Force High A

Embedded Passive RF Input List

RFR1	D410	50Ohms	L=433.972mils W=20.69mils H=10mils Er=3.9
RFR2	C214	100Ohms	L=456.85mils W=4.85mils H=10mils Er=3.9
RFR3	U303	37.5Ohms	L=400.576mils W=25.961mils H=10mils Er=3.9
RFB1	R439	70.7Ohms	L1=700mils L2=2100mils W1=10.2mils W2=8mils H1=10mils H2=8mils SPL1L2=10mils SP12L1=10mils SPL1=30mils

APPENDIX C

Induce RF Component into Netlist:

```
#:... Induce the RF component in Netlist :
  open(DAT,"test4.txt") || die("ERROR : Could not open RF file!");
  open(TAD,"netlist.txt") || die("ERROR : Could not open RF file!");
  foreach $line (<DAT>)
        {
             chomp($line);
             my @field = split (' ',$line);
             my $Component1_Type    = $field[0]; # Component Type
                    my $Component1_Number  = $field[1]; # Place after this component in netlist
                    my $Component2_Type    = $field[2]; # Component Type
                    my $Component2_Number  = $field[3]; # Place after this component in netlist
                    my $Component3_Type    = $field[4]; # Component Type
                    my $Component3_Number  = $field[5]; # Place after this component in netlist
                    my $Component4_Type    = $field[6]; # Component Type
                    my $Component4_Number  = $field[7]; # Place after this component in netlist
  foreach $line (<TAD>)
        {
             chomp($line);
             my @field = split (' ',$line);
             my $Cmp1_Num    = $field[0]; # Component Number
                    my $Node1_Num   = $field[1]; # Node Number
                    my $Cmp2_Num    = $field[2]; # Component Number
                    my $Node2_Num   = $field[3]; # Node Number
                    my $Cmp3_Num    = $field[4]; # Component Number
                    my $Node3_Num   = $field[5]; # Node Number
                    my $Cmp4_Num    = $field[6]; # Component Number
                    my $Node4_Num   = $field[7]; # Node Number
                    my $Cmp5_Num    = $field[8]; # Component Number
                    my $Node5_Num   = $field[9]; # Node Number
my $Node_Num = "1";
if ($Component1_Number eq $Cmp1_Num) {
    $Cmp5_Num = $Cmp4_Num;
    $Node5_Num = $Node4_Num;
    $Cmp4_Num = $Cmp3_Num;
    $Node4_Num = $Node3_Num;
    $Cmp3_Num = $Cmp2_Num;
    $Node3_Num = $Node2_Num;
    $Cmp2_Num = $Component1_Type;
    $Node2_Num = $Node_Num;
} elsif ($Component1_Number eq $Cmp2_Num) {
    $Cmp5_Num = $Cmp4_Num;
    $Node5_Num = $Node4_Num;
    $Cmp4_Num = $Cmp3_Num;
    $Node4_Num = $Node3_Num;
    $Cmp3_Num = $Component1_Type;
    $Node3_Num = $Node_Num;
} elsif ($Component1_Number eq $Cmp3_Num) {
    $Cmp5_Num = $Cmp4_Num;
    $Node5_Num = $Node4_Num;
    $Cmp4_Num = $Component1_Type;
    $Node4_Num = $Node_Num;
} elsif ($Component1_Number eq $Cmp4_Num) {
    $Cmp5_Num = $Component1_Type;
    $Node5_Num = $Node_Num;
} elsif ($Component2_Number eq $Cmp1_Num) {
    $Cmp5_Num = $Cmp4_Num;
    $Node5_Num = $Node4_Num;
    $Cmp4_Num = $Cmp3_Num;
    $Node4_Num = $Node3_Num;
    $Cmp3_Num = $Cmp2_Num;
    $Node3_Num = $Node2_Num;
    $Cmp2_Num = $Component2_Type;
    $Node2_Num = $Node_Num;
} elsif ($Component2_Number eq $Cmp2_Num) {
    $Cmp5_Num = $Cmp4_Num;
    $Node5_Num = $Node4_Num;
    $Cmp4_Num = $Cmp3_Num;
```

```
    $Node4_Num = $Node3_Num;
    $Cmp3_Num = $Component2_Type;
    $Node3_Num = $Node_Num;
  } elsif ($Component2_Number eq $Cmp3_Num) {
    $Cmp5_Num = $Cmp4_Num;
    $Node5_Num = $Node4_Num;
    $Cmp4_Num = $Component2_Type;
    $Node4_Num = $Node_Num;
  } elsif ($Component2_Number eq $Cmp4_Num) {
    $Cmp5_Num = $Component2_Type;
    $Node5_Num = $Node_Num;
  } elsif ($Component3_Number eq $Cmp1_Num) {
    $Cmp5_Num = $Cmp4_Num;
    $Node5_Num = $Node4_Num;
    $Cmp4_Num = $Cmp3_Num;
    $Node4_Num = $Node3_Num;
    $Cmp3_Num = $Cmp2_Num;
    $Node3_Num = $Node2_Num;
    $Cmp2_Num = $Component3_Type;
    $Node2_Num = $Node_Num;
  } elsif ($Component3_Number eq $Cmp2_Num) {
    $Cmp5_Num = $Cmp4_Num;
    $Node5_Num = $Node4_Num;
    $Cmp4_Num = $Cmp3_Num;
    $Node4_Num = $Node3_Num;
    $Cmp3_Num = $Component3_Type;
    $Node3_Num = $Node_Num;
  } elsif ($Component3_Number eq $Cmp3_Num) {
    $Cmp5_Num = $Cmp4_Num;
    $Node5_Num = $Node4_Num;
    $Cmp4_Num = $Component3_Type;
    $Node4_Num = $Node_Num;
  } elsif ($Component3_Number eq $Cmp4_Num) {
    $Cmp5_Num = $Component3_Type;
    $Node5_Num = $Node_Num;
  } elsif ($Component4_Number eq $Cmp1_Num) {
    $Cmp5_Num = $Cmp4_Num;
    $Node5_Num = $Node4_Num;
    $Cmp4_Num = $Cmp3_Num;
    $Node4_Num = $Node3_Num;
    $Cmp3_Num = $Cmp2_Num;
    $Node3_Num = $Node2_Num;
    $Cmp2_Num = $Component4_Type;
    $Node2_Num = $Node_Num;
  } elsif ($Component4_Number eq $Cmp2_Num) {
    $Cmp5_Num = $Cmp4_Num;
    $Node5_Num = $Node4_Num;
    $Cmp4_Num = $Cmp3_Num;
    $Node4_Num = $Node3_Num;
    $Cmp3_Num = $Component4_Type;
    $Node3_Num = $Node_Num;
  } elsif ($Component4_Number eq $Cmp3_Num) {
    $Cmp5_Num = $Cmp4_Num;
    $Node5_Num = $Node4_Num;
    $Cmp4_Num = $Component4_Type;
    $Node4_Num = $Node_Num;
  } elsif ($Component4_Number eq $Cmp4_Num) {
    $Cmp5_Num = $Component4_Type;
    $Node5_Num = $Node_Num;
  }
    print "$Cmp1_Num\n";
    print "$Node1_Num\n";
    print "$Cmp2_Num\n";
    print "$Node2_Num\n";
    print "$Cmp3_Num\n";
    print "$Node3_Num\n";
    print "$Cmp4_Num\n";
    print "$Node4_Num\n";
    print "$Cmp5_Num\n";
    print "$Node5_Num\n";
```

```
}
}
exit 0;
```

Induce RF Component into Parts List:

```
#:... Induce the RF component in Netlist :
  open(DAT,"test3.txt") || die("ERROR : Could not open RF file!");
  open(TAD,"partslist.txt") || die("ERROR : Could not open RF file!");
  foreach $line (<DAT>)
        {
             chomp($line);
             my @field = split (' ',$line);
             my $Component_Type    = $field[0]; # Component Type
  foreach $line (<TAD>)
        {
             chomp($line);
             my @field = split (' ',$line);
             my $Cmp_Typ    = $field[0]; # Model Number
                    my $Cmp_Typ1   = $field[1]; # Model Number
                    my $Cmp1_Num    = $field[2]; # Component Number
                    my $Cmp2_Num   = $field[3]; # Component Number
                    my $Cmp3_Num    = $field[4]; # Component Number
    print "$Cmp_Typ\n";
    print "$Cmp_Typ1\n";
    print "$Cmp1_Num\n";
    print "$Cmp2_Num\n";
    print "$Cmp3_Num\n";
    }
my $Cmp_Typ = "Embedded Passive";
my $Cmp_Typ1 = "Embedded Passive";
$Cmp1_Num = $Component_Type;
print "$Cmp_Typ\n";
print "$Cmp_Typ1\n";
print "$Cmp1_Num\n";
}
exit 0;
```

Fault Dictionary:

```
  #:... Open the Build ciruit module results :
  open(DAT,"test1.txt") || die("ERROR : Could not open BOM file!");
  foreach $line (<DAT>)
        {
             chomp($line);
             my @field = split (' ',$line);
             my $Component_Num    = $field[0]; # Item number field
                    my $Component_Type   = $field[2]; # Number of ref designators by line
                    my $Value            = $field[3]; # Model field
             my $Limit            = $field[4]; # Model field
    my $Resistor = "COMPONENT:R";
    my $Capacitor = "COMPONENT:C";
    my $inductor = "COMPONENT:L";
    my $Fal_valr = "VALUE:1e+009";
    if ($Component_Type eq $Resistor) {
    $Value = $Fal_valr;
    print "$Component_Num\n";
    print "$Component_Type\n";
    print "$Value\n";
    print "$Limit\n";
    } else {
    print "$Component_Num\n";
    print "$Component_Type\n";
    print "$Value\n";
    print "$Limit\n";
    }
    my $Fal_valc = "VALUE:1e-018";
    if ($Component_Type eq $Capacitor) {
    $Value = $Fal_valc;
```

```
  print "$Component_Num\n";
  print "$Component_Type\n";
  print "$Value\n";
  print "$Limit\n";
  } else {
  print "$Component_Num\n";
  print "$Component_Type\n";
  print "$Value\n";
  print "$Limit\n";
  }
  my $Fal_vall = "VALUE:1e+009";
  if ($Component_Type eq $Inductor) {
  $Value = $Fal_vall;
  print "$Component_Num\n";
  print "$Component_Type\n";
  print "$Value\n";
  print "$Limit\n";
  } else {
  print "$Component_Num\n";
  print "$Component_Type\n";
  print "$Value\n";
  print "$Limit\n";
  }
  }
exit 0;
 #:... Open the Build ciruit module results :
 open(DAT,"test1.txt") || die("ERROR : Could not open BOM file!");
 foreach $line (<DAT>)
       {
            chomp($line);
            my @field = split (' ',$line);
            my $Component_Num    = $field[0]; # Item number field
                  my $Component_Type   = $field[2]; # Number of ref designators by line
                  my $Value            = $field[3]; # Model field
            my $Limit            = $field[4]; # Model field
  my $Resistor = "COMPONENT:R";
  my $Capacitor = "COMPONENT:C";
  my $inductor = "COMPONENT:L";
  my $Fal_vall = "VALUE:0.001";
  if ($Component_Type eq $Inductor) {
  $Value = $Fal_vall;
  print "$Component_Num\n";
  print "$Component_Type\n";
  print "$Value\n";
  print "$Limit\n";
  } else {
  print "$Component_Num\n";
  print "$Component_Type\n";
  print "$Value\n";
  print "$Limit\n";
  }
  my $Fal_valc = "VALUE:1e-006";
  if ($Component_Type eq $Capacitor) {
  $Value = $Fal_valc;
  print "$Component_Num\n";
  print "$Component_Type\n";
  print "$Value\n";
  print "$Limit\n";
  } else {
  print "$Component_Num\n";
  print "$Component_Type\n";
  print "$Value\n";
  print "$Limit\n";
  }
  my $Fal_vall = "VALUE:0.001";
  if ($Component_Type eq $Inductor) {
  $Value = $Fal_vall;
  print "$Component_Num\n";
  print "$Component_Type\n";
  print "$Value\n";
```

```
    print "$Limit\n";
    } else {
    print "$Component_Num\n";
    print "$Component_Type\n";
    print "$Value\n";
    print "$Limit\n";
    }
    }
exit 0;
#:... Open the Build ciruit module results :
  open(DAT,"test2.txt") || die("ERROR : Could not open BOM file!");
  foreach $line (<DAT>)
        {
            chomp($line);
            my @field = split (' ',$line);
            my $f1   = $field[0]; # Item number field
                    my $f2   = $field[1]; # Number of ref designators by line
                    my $f3   = $field[2]; # Model field
            my $f4   = $field[3]; # Model field
            my $f5   = $field[4]; # Model field
            my $f6   = $field[5]; # Model field
            my $f7   = $field[6]; # Model field
my $f8   = $field[7]; # Model field
my $replf = ":[T->1";
my $newf = ":[T->0";
my $sa1 = "0";
my $news1 = "";
if ($f4 eq $replf) {
    $f4 = $newf;
    print "$f1\n";
    print "$f2\n";
    print "$f3\n";
    print "$f4\n";
    print "$f5\n";
    print "$f6\n";
 print "$f7\n";
} elsif ($f1 eq $sa1) {
    $f1 = $news1;
    $f2 = $news1;
    $f3 = $news1;
    $f4 = $news1;
    $f5 = $news1;
    print "$f1\n";
    print "$f2\n";
    print "$f3\n";
    print "$f4\n";
    print "$f5\n";
    print "$f6\n";
print "$f7\n"
}
else {
    print "$f1\n";
    print "$f2\n";
    print "$f3\n";
    print "$f4\n";
    print "$f5\n";
    print "$f6\n";
print "$f7\n";
}
}
exit 0;
#:... Open the Build ciruit module results :
  open(DAT,"test2.txt") || die("ERROR : Could not open BOM file!");
  foreach $line (<DAT>)
        {
            chomp($line);
            my @field = split (' ',$line);
            my $f1   = $field[0]; # Item number field
                    my $f2   = $field[1]; # Number of ref designators by line
                    my $f3   = $field[2]; # Model field
```

```
            my $f4   = $field[3]; # Model field
            my $f5   = $field[4]; # Model field
            my $f6   = $field[5]; # Model field
            my $f7   = $field[6]; # Model field
my $f8   = $field[7]; # Model field
my $replf = ":[T->0";
my $newf = ":[T->1";
my $sa1 = "1";
my $sa2 = "2";
my $sa3 = "3";
my $news1 = "";
if ($f4 eq $replf) {
   $f4 = $newf;
   print "$f1\n";
   print "$f2\n";
   print "$f3\n";
   print "$f4\n";
   print "$f5\n";
   print "$f6\n";
 print "$f7\n";
}
elsif ($f1 eq $sa1) {
   $f1 = $news1;
   $f2 = $news1;
   $f3 = $news1;
   $f4 = $news1;
   $f5 = $news1;
   print "$f1\n";
   print "$f2\n";
   print "$f3\n";
   print "$f4\n";
   print "$f5\n";
   print "$f6\n";
print "$f7\n";
}
elsif ($f1 eq $sa2) {
   $f1 = $news1;
   $f2 = $news1;
   $f3 = $news1;
   $f4 = $news1;
   $f5 = $news1;
   print "$f1\n";
   print "$f2\n";
   print "$f3\n";
   print "$f4\n";
   print "$f5\n";
   print "$f6\n";
   print "$f7\n";
}
elsif ($f1 eq $sa3) {
   $f1 = $news1;
   $f2 = $news1;
   $f3 = $news1;
   $f4 = $news1;
   $f5 = $news1;
   print "$f1\n";
   print "$f2\n";
   print "$f3\n";
   print "$f4\n";
   print "$f5\n";
   print "$f6\n";
   print "$f7\n";
}
else {
   print "$f1\n";
   print "$f2\n";
   print "$f3\n";
   print "$f4\n";
   print "$f5\n";
   print "$f6\n";
```

```
print "$f7\n";
}
}
exit 0;
```

Look-up Table:

```
#:... Open the Pseudocode :
  open(DAT,"pseu.txt") || die("ERROR : Could not open pseudocode file!");
  open(DAT1,"pseuopen.txt") || die("ERROR : Could not open pseudocode file!");
  open(DAT2,"pseushort.txt") || die("ERROR : Could not open pseudocode file!");
  open OUTPUT, ">output.txt";
  print OUTPUT "Part_Num    Fault Type     Lo Limit    Up Limit    Actual Value\n";
  foreach $line (<DAT>)
        {
             chomp($line);
             my @field = split (' ',$line);
             my $Component_Num    = $field[0]; # Item number field
                    my $Component_Type   = $field[2]; # Number of ref designators by line
                    my $Value            = $field[3]; # Model field
             my $Limit            = $field[4]; # Model field
             my $Value1             = $field[8]; # Upper Limit
             my $Limit1             = $field[9]; # Upper Limit
             my $Value2             = $field[13]; # Upper Limit
             my $Limit2             = $field[14]; # Upper Limit
             my $passcomp         = $field[17]; # Passive Component
             my $pascmpval        = $field[22]; # Passive Component Value
    my $Comp_Typ = "CONDITION(VERIFY";
    my $Comp_Lim = "COMPARE_LIMITS(lower";
    my $Part_Num = "";
    my $Fault_Type = "Fault Free";
    my $SMDC1 = "******SMA";
    my $SMDC2 = "******DIRECT";
    my $test = "tested";
    my $Part_Num = "";
    if ($Component_Num eq $SMDC1) {
    $Part_Num = "SMA      ";
    print OUTPUT "$Part_Num";
    } elsif ($Component_Num eq $SMDC2) {
    $Part_Num = "DirConnec";
    print OUTPUT "$Part_Num";
    } elsif (($Limit2 eq $test) && ($Component_Num eq $Comp_Lim)) {
    $Part_Num = $passcomp;
    print OUTPUT "$Part_Num";
    print OUTPUT "      $Fault_Type     $Value$Limit      $Value1$Limit1      $pascmpval\n";
    } elsif ($Component_Num eq $Comp_Typ) {
    $Part_Num = $Value;
    print OUTPUT "$Part_Num";
    } elsif ($Component_Num eq $Comp_Lim) {
    print OUTPUT "   $Fault_Type     $Value$Limit      $Value1$Limit1      $Value2$Limit2\n";
    }
    }
    foreach $line (<DAT1>)
        {
             chomp($line);
             my @field1 = split (' ',$line);
             my $Component_Num1    = $field1[0]; # Item number field
                    my $Component_Type1   = $field1[2]; # Number of ref designators by line
                    my $Value1            = $field1[3]; # Model field
             my $Limit1            = $field1[4]; # Model field
             my $Value11             = $field1[8]; # Upper Limit
             my $Limit11             = $field1[9]; # Upper Limit
             my $Value21             = $field1[13]; # Upper Limit
             my $Limit21             = $field1[14]; # Upper Limit
             my $passcomp1         = $field1[17]; # Passive Component
             my $pascmpval1        = $field1[22]; # Passive Component Value
    my $Comp_Typ1 = "CONDITION(VERIFY";
    my $Comp_Lim1 = "COMPARE_LIMITS(lower";
    my $Part_Num1 = "";
```

```
my $Fault_Type1 = "Open Fault";
my $test1 = "tested";
my $Part_Num1 = "";
if (($Limit21 eq $test1) && ($Component_Num1 eq $Comp_Lim1)) {
$Part_Num1 = $passcomp1;
print OUTPUT "$Part_Num1";
print OUTPUT "      $Fault_Type1     $Value1$Limit1      $Value11$Limit11      $pascmpval1\n";
} elsif ($Component_Num1 eq $Comp_Typ1) {
$Part_Num1 = $Value1;
print OUTPUT "$Part_Num1";
} elsif ($Component_Num1 eq $Comp_Lim1) {
print OUTPUT "   $Fault_Type1     $Value1$Limit1      $Value11$Limit11      $Value21$Limit21\n";
}
}
foreach $line (<DAT2>)
    {
        chomp($line);
        my @field = split (' ',$line);
        my $Component_Num    = $field[0]; # Item number field
                my $Component_Type   = $field[2]; # Number of ref designators by line
                my $Value            = $field[3]; # Model field
        my $Limit           = $field[4]; # Model field
        my $Value1            = $field[8]; # Upper Limit
        my $Limit1            = $field[9]; # Upper Limit
        my $Value2            = $field[13]; # Upper Limit
        my $Limit2            = $field[14]; # Upper Limit
        my $passcomp        = $field[17]; # Passive Component
        my $pascmpval       = $field[22]; # Passive Component Value
my $Comp_Typ = "CONDITION(VERIFY";
my $Comp_Lim = "COMPARE_LIMITS(lower";
my $Part_Num = "";
my $Fault_Type = "Short Fault";
my $SMDC1 = "*****SMA";
my $SMDC2 = "*****DIRECT";
my $test = "tested";
my $Part_Num = "";
if ($Component_Num eq $SMDC1) {
$Part_Num = "SMA      ";
print OUTPUT "$Part_Num";
} elsif ($Component_Num eq $SMDC2) {
$Part_Num = "DirConnec";
print OUTPUT "$Part_Num";
} elsif (($Limit2 eq $test) && ($Component_Num eq $Comp_Lim)) {
$Part_Num = $passcomp;
print OUTPUT "$Part_Num";
print OUTPUT "      $Fault_Type     $Value$Limit      $Value1$Limit1      $pascmpval\n";
} elsif ($Component_Num eq $Comp_Typ) {
$Part_Num = $Value;
print OUTPUT "$Part_Num";
} elsif ($Component_Num eq $Comp_Lim) {
print OUTPUT "   $Fault_Type     $Value$Limit      $Value1$Limit1      $Value2$Limit2\n";
}
}
exit 0;
```

APPENDIX D

LMV225 Datasheet:

National Semiconductor

March 2005

LMV225/LMV226/LMV228
RF Power Detector for CDMA and WCDMA

General Description

The LMV225/LMV226/LMV228 are 30 dB RF power detectors intended for use in CDMA and WCDMA applications. The device has an RF frequency range from 450 MHz to 2 GHz. It provides an accurate temperature and supply compensated output voltage that relates linearly to the RF input power in dBm. The circuit operates with a single supply from 2.7V to 5.5V. The LMV225/LMV226/LMV228 have an integrated filter for low-ripple average power detection of CDMA signals with 30 dB dynamic range. Additional filtering can be applied using a single external capacitor.

The LMV225 has an RF power detection range from −30 dBm to 0 dBm and is ideally suited for direct use in combination with resistive taps. The LMV226/LMV228 have a detection range from −15 dBm to 15 dBm and are intended for use in combination with a directional coupler. The LMV226 is equipped with a buffered output which makes it suitable for GSM, EDGE, GPRS and TDMA applications.

The device is active for Enable = HI, otherwise it is in a low power consumption shutdown mode. During shutdown the output will be LOW. The output voltage ranges from 0.2V to 2V and can be scaled down to meet ADC input range requirements.

The LMV225/LMV226/LMV228 power detectors are offered in the small 1.0 mm x 1.0 mm X 0.6 mm micro SMD package. The LMV225 and the LMV228 are also offered in the 2.2 mm x 2.5 mm x 0.8 mm LLP package.

Features

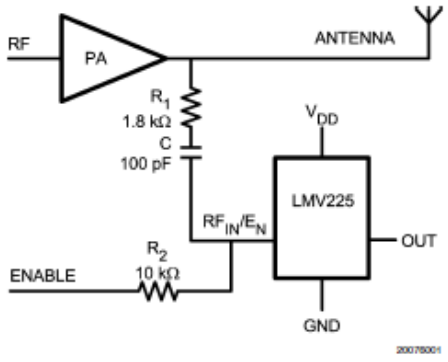
- 30 dB linear in dB power detection range
- Output voltage range 0.2 to 2V
- Logic low shutdown
- Multi-band operation from 450 MHz to 2000 MHz
- Accurate temperature compensation
- Packages:
  - — micro SMD package 1.0 mm x 1.0 mm x 0.6 mm
  - — LLP package 2.2 mm x 2.5 mm x 0.8 mm (LMV225 and LMV228)

Applications

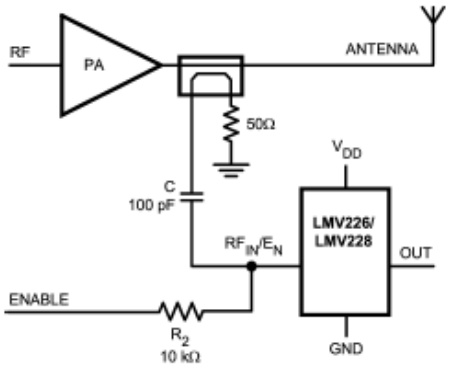
- CDMA RF power control
- WCDMA RF power control
- CDMA2000 RF power control
- PA modules

Typical Application

LMV225



LMV226/LMV228



Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

Supply Voltage	
V_{DD} - GND	6.0V Max
ESD Tolerance (Note 2)	
Human Body Model	2000V
Machine Model	200V
Storage Temperature Range	−65°C to 150°C
Junction Temperature (Note 3)	150°C Max
Mounting Temperature	
Infrared or convection (20 sec)	235°C

Operating Ratings (Note 1)

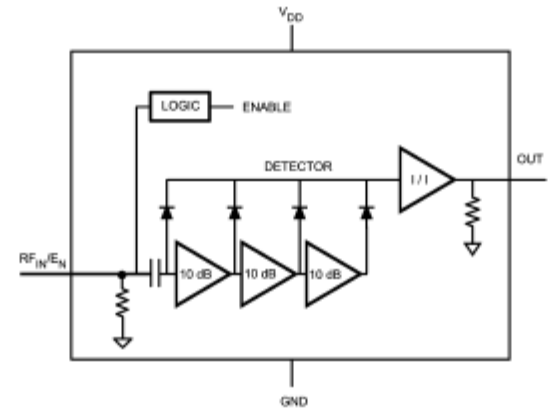
Supply Voltage	2.7V to 5.5V
Temperature Range	−40°C to +85°C
RF Frequency Range	450 MHz to 2 GHz

2.7 DC and AC Electrical Characteristics

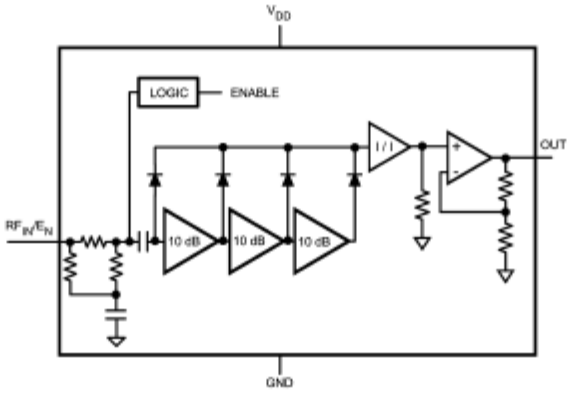
Unless otherwise specified, all limits are guaranteed to V_{DD} = 2.7V; T_J = 25°C. **Boldface** limits apply at temperature extremes. (Note 4)

Symbol	Parameter	Condition		Min	Typ	Max	Units
I_{DD}	Supply Current	Active Mode: RF_{IN}/E_N = V_{DD} (DC), No RF Input Power Present	LMV225		4.8	7 **8**	mA
			LMV226		4.9	6.2 **8**	
			LMV228		4.9	6.2 **8**	
		Shutdown: RF_{IN}/E_N = GND (DC), No RF Input Power Present			0.44	**4.5**	µA
V_{LOW}	E_N Logic Low Input Level (Note 6)					**0.8**	V
V_{HIGH}	E_N Logic High Input Level (Note 6)			**1.8**			V
t_{on}	Turn-on-Time (Note 9)	No RF Input Power Present, Output Loaded with 10 pF	LMV225		2.1		µs
			LMV226		1.2		
			LMV228		1.7		
t_r	Rise Time (Note 7)	Step from no Power to 0 dBm Applied, Output Loaded with 10 pF	LMV225		4.5		µs
		Step from no Power to 15 dBm Applied, Output Loaded with 10 pF	LMV226		1.8		
			LMV228		4.8		
I_{EN}	Current into RF_{IN}/E_N Pin					**1**	µA
P_{IN}	Input Power Range (Note 5)	LMV225			−30 0		dBm
					−43 −13		dBV
		LMV226			−15 15		dBm
					−28 2		dBV
		LMV228			−15 15		dBm
					−28 2		dBV

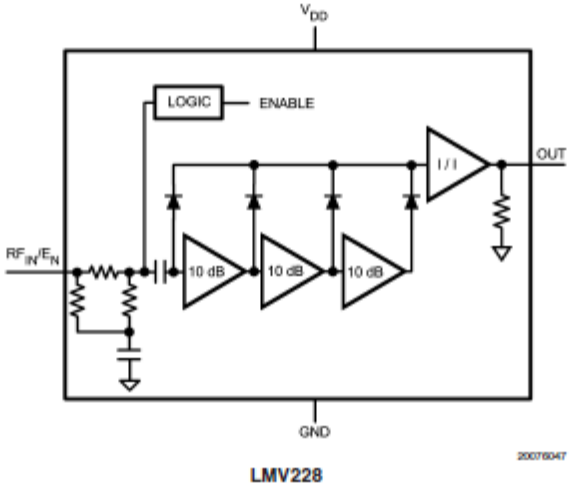
Block Diagrams



LMV225



LMV226



LMV228