GOVERNMENT OF TAMILNADU
DIRECTORATE OF TECHNICAL EDUCATION
CHENNAI – 600 025

STATE PROJECT COORDINATION UNIT

Diploma in Electronics and Communication Engineering

Course Code: 1040

M – Scheme

e-TEXTBOOK on TEST ENGINEERING for VI Semester DECE

Convener for ECE Discipline:

Dr.M.JeganMohan M.E., MBA., Ph.D.,(Management), Ph.D.,(Engg)., M.I.S.T.E.,
Principal,
138, Government Polytechnic College,
Uthappanaickanoor,
Usilampatti, Madurai – 625 537.

Team Members for Industrial Electronics:

Mr.G.ATHIRAJAN, B.E., M.E.,
HOD / ECE,
K.L.Nagaswamy Memorial Polytechnic College,
Madurai – 625 009.

Mrs.G.Vijayakumari,
Lecturer /ECE ,
Government Polytechnic College,
Melur – 625001.

Validated By

Dr.Mrs.D.Selvathi,
Professor/ECE,
4960,Mepco Schlenk Engineering College,
Virudhunagar – 626 005.

GOVERNMENT OF TAMILNADU
DIRECTORATE OF TECHNICAL EDUCATION
CHENNAI – 600 025

STATE PROJECT COORDINATION UNIT

Diploma in Electronics and Communication Engineering

Course Code: 1040

M – Scheme

e-TEXTBOOK on TEST ENGINEERING for VI Semester DECE

Convener for ECE Discipline:

Dr.M.JeganMohan M.E., MBA., Ph.D.,(Management), Ph.D.,(Engg)., M.I.S.T.E.,
Principal,
138, Government Polytechnic College,
Uthappanaickanoor,
Usilampatti, Madurai – 625 537.

Team Members for Industrial Electronics:

Mr.G.ATHIRAJAN, B.E., M.E.,
HOD / ECE,
K.L.Nagaswamy Memorial Polytechnic College,
Madurai – 625 009.

Mrs.G.Vijayakumari,
Lecturer /ECE ,
Government Polytechnic College,
Melur – 625001.

Validated By

Dr.Mrs.D.Selvathi,
Professor/ECE,
4960,Mepco Schlenk Engineering College,
Virudhunagar – 626 005.

Contents

Contents

Unit-1Introduction To Test Engineering

UNIT – IIAUTOMATED TESTING METHODS AND TECHNOLOGY

UNIT – III

V-I (Signature) TESTING METHODS AND TECHNOLOGY

UNIT – IV

BOUNDARY SCAN TESTING METHODS AND TECHNOLOGY

UNIT – VATE Test Program generation And Semiconductor testing

Unit-I

INTRODUCTION TO TEST ENGINEERING

Test Engineering is a quality assurance testing, problem solving , where an test engineer will perform testing on devices or products. The role of test engineer is to test and deploy effective test solutions. Test Solutions for Aelectronics equipment, electronics components, software, automated devices, and other products, to ensure that they handle potential duress caused by typical usage, as well as to ensure that there are no flaws, bugs or errors hampering them.

A test engineer is required to full fill the product or system testing to ensure its functionality to meet the needs.

A test engineer is responsible for not only testing the quality of a product, but also must devise and develop tests that will assess the reliability of the product.

How to teach the concepts of such an important subject like Test engineering most effectively?

If students wants to gain a good understanding of "**Test Engineering**", Should understand the concepts of Test Engineering and test them independently.

For this one should acquire a practical knowledge of the Characteristics and Functions of different devices and components, and the various circuits.

Let us try to learn such skills by the proven scheme of "**LEARNING BY DOING**" Learning by doing an old Chinese proverb

says

I READ – I FORGET
I SEE – I REMEMBER
I DO – I UNDERSTAND

That is the way the "**TEST ENGINEERING**" theory and practical will guide to make learn and to understand the concepts of Test Engineering.

1.1 Need and Importance of Test Engineering

Test Engineering is not a new concept in the testing world. In earlier 90's there was no competition for development and to release a new product, so they have enough time line to test and validate a product before release, the product was reliable and precise.

In modern technology world, time to market and high quality has become huge headaches for companies, due to increased competition and complexity.

Several times we have seen in day to day life the product that we purchase those gets repair during the product's warranty period itself even immediately after purchase.

Best Example is Cell/Mobile Phone – Touch Not Working, Display Not Working, Heating Abnormal, No Sound, Network Issues etc. Figure 1.1 Shows the Test Engineering involved in various fields.

To over come those issues and to maintain the standard and quality of product, from product development till production cycle, companies had understand the importance and need of test engineering in modern world, to keep their market intact.

Overall, this drives manufacturing reliability and quality at the end of the line making sure that all units shipped out to customers are well tested, stressed, filtered out of any errors, and configured properly.

Work description of a test engineer varies based on the type of product they are testing, and what they purpose of the testing isFor example, the work description of a test engineer who works in software will vary from a test engineer that works in electronics.
Test engineers can have different expertise which depends on what test process they are more familiar with, figure shows test engineering involved in the various fields. In below figure 1.1 Shows the Test Engineering involved in various fields.

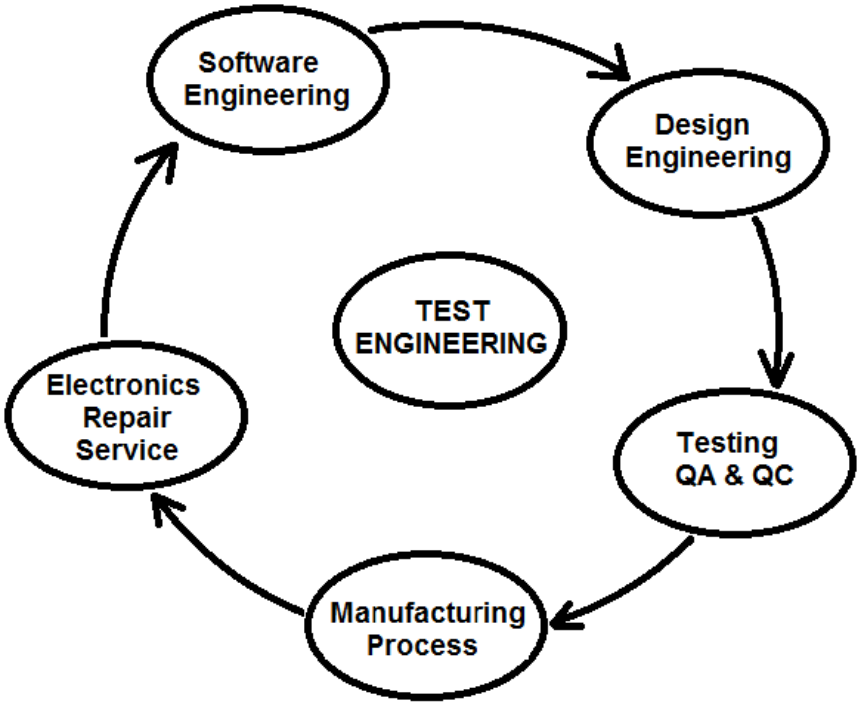


Figure 1.1 Shows the Test Engineering involved in various fields.

- Software Engineering
- Design Engineering
- Testing, Quality Control and Quality Assurances
- Manufacturing Process
- Electronics Repair Services

Software Engineering:-

In Software Engineering (SWE) Software testing is a process of executing a program or application with the intent of finding the software bugs. The universe of testing automation can be neatly split into two predominant testing techniques known as black-box and white-box testing.

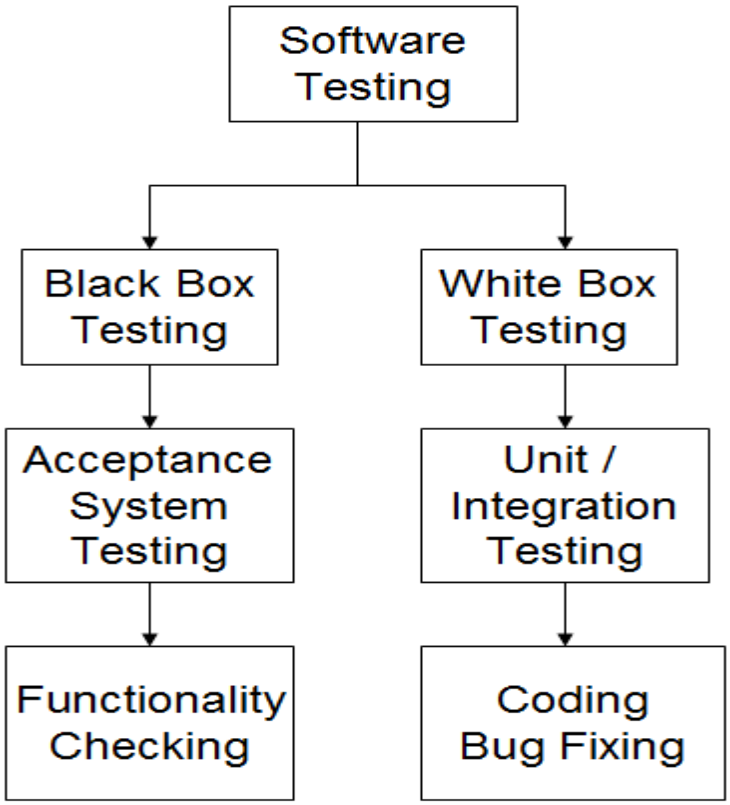


Figure 1.2 Shows the Software testing flow.

Other test design techniques exist, including gray-box testing, which is a combination of the previous two, however black-box and white-box testing approaches are the most widespread.Figure1.2 Shows the Software testing flow. Show in the figure 1.2 Shows the Software testing flow.

Black box testing is the method which is used to test the software without knowing the internal structure of code or program.

White box testing is a testing technique, which examines the program structure and derives test data from the program code.

Criteria	Black Box Testing	White Box Testing
Definition	Black Box Testing is a software testing method in which the internal structure/ design/ implementation of the item being tested is not	White Box Testing is a software testing method in which the internal structure/ design/ implementation of the item being tested is known to

	known to the tester	the tester.
Levels Applicable to	Mainly applicable to higher levels of testing: Acceptance Testing & System Testing	Mainly applicable to lower levels of testing: Unit Testing & Integration Testing
Responsibility	Generally, independent Software Testers	Generally, Software Developers
Programming Knowledge	Not Required	Required
Implementation Knowledge	Not Required	Required
Basis for Test Cases	Requirement Specifications	Detail Design

Table 1.1 differentiates between Black Box & White Box Testing

For absolute newcomers, let us understand these three types of testing with an over simplified example in layman's terms:

For a functional mobile phone, the main parts required are "battery" and "SIM card - Subscriber identity module".

Unit testing: – the battery is checked for its life, capacity and other parameters. SIM card is checked for its activation.

Integration Testing: – battery and SIM card are integrated i.e. assembled in order to start the mobile phone.

Functional Testing: – the functionality of the mobile phone is checked in terms of its features and also battery usage as well as SIM card facilities.

The unit, Integration, and Functional testing: All three of them are correlated. To attain full coverage it is required to have unit tests for the paths/lines of code, functional and Integration tests for assurance that the 'units' work together cohesively.

Design Engineering:-

Test engineering place a major role in design engineering also know as hardware/embedded development design. Depending on the culture of the firm, in some of the firm/organization test engineer are involved in very earlier stages of the product design phase.

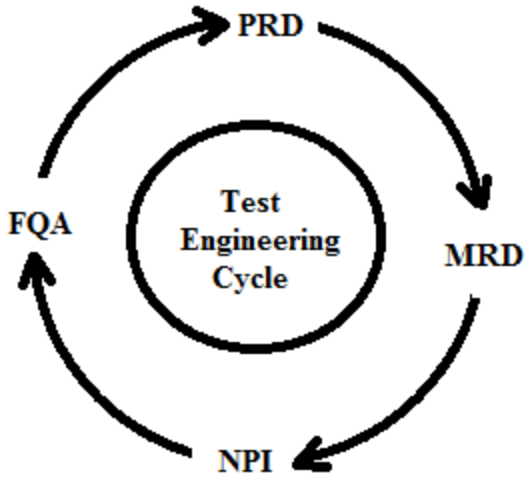


Figure 1.3 The Test Engineering Cycles / Process

Show in the figure 1.3 The Test Engineering Cycles / Process. The earlier stages could refer to Product Requirements Document (PRD) and Marketing Requirements Document (MRD), some of the earliest work done during a new product introduction (NPI). A test engineer ensures that a product is designed for both testability and manufacturability and to make sure standards are followed as per guideline.

- Product Requirements Document - PRD
- Marketing Requirements Document - MRD
- New Product Introduction - NPI
- Final Quality Assurance - FQA

A Product Requirements Document (PRD) is a document containing all the requirements to a certain product. It is written to allow people to understand what a product should do. A PRD should, however, generally avoid anticipating or defining how the product will do it in order to later allow interface designers and engineers to use their expertise to provide the optimal solution to the requirements.

PRDs are most frequently written for software products, but can be used for any type of product and also for services. Typically, a PRD is created from a user's point-of-view by a user/client or a company's marketing department in the latter case it may also be called Marketing Requirements Document (MRD).

In business and engineering, new product development (NPD) is the complete process of bringing a new product to market.

A test engineer involve till the final quality audit (FQA) process, in the electronic hardware manufacturing world final quality audit process, is the last process flow before shipping a product. This process is established to ensure the unit has gone through and passed all the manufacturing or test process and is in good quality.

Test Engineer Responsibility

- Develop and document testing processes and procedures.
- Liaise with design engineers, product development teams and marketing personnel.
- Test technical and/or computer products and devices.

- Report problems or failures identified during testing.
- Make recommendations to product development team concerning improvements and solutions to problems.

Some of the documents that the test engineers maintain or define:

- Test method.
- Diagnostic design specification.
- Manufacturing test requirement design specification.
- Design for testability (DFT).
- Design for manufacturability (DFM).
- Test plan.

How Organizations are differ from with and without Test Engineer

Often organizations employ take shortcuts to be able to deliver final products. Because of these shortcuts, the product's manufacturability and testability becomes complicated (inability to read and write information, creating deviation from the process, etc.) which impacts the manufacturing complexity of a product.

Because of this complexity, bottlenecks in the manufacturing and delivery schedule delays are introduced. With this in mind, test engineers always get involved. Maintaining a Test Engineer in the firm is vital success of the product.

Firm With Test Engineer	**Firm Without Test Engineer**
TE Develop proper test plans	No Proper Plans Executed
TE Develops test cases	Not a valid test case
Ensure version management	No Proper version maintained
Maintain Quality of Product	There is no product Stabilization
Timely Executions	Time Delay
Cost Effective / Maintain Inventory	High Cost / Increased Inventory
Maintain Standards	Improper standard maintained
All are documented	No such valid document maintained
No employ dependency	Firm always depends on employ

Meets the business needs	Lose Customer / Dissatisfactory

Table 1.2shows what cause the firm with and without test engineer.

Testing Quality Control and Quality Assurances

Quality Control and Quality assurances are the part of Software and Hardware test engineering.

The testing world has a lot of terms for the activity that we undertake every day.You'll often hear the words QA, QC, and Test Engineering used interchangeably. While it is usually enough to get your point across with a developer, it is helpful to think about these terms and how they apply to the world of software testing.

In the classic definition QC is short for Quality Control, a process of verifying predefined requirements for quality. In the terms of an assembly-line this might involve pulling manufactured units off at the end of the process and verifying different parts of the assembly process.

For software the QC function may involve checking the software against a set of requirements and verifying that the software meets the predefined requirements.

Quality Assurance, on the other hand, is much more about providing the continuous and consistent improvement and maintenance of process that enables the QC job. We use the QC process to verify a product does what we think it does, and we use the QA process to give us confidence that the product will meet the needs of customers.

To that end the QA process can be considered a higher order level process that includes aspects of the QC process. It also goes beyond that to influence usability and design, to verify that functionality is not only correct, but useful.

Quality Assurance vs Quality Control

Quality Assurance	Quality Control
It is a process which deliberate on providing assurance that quality request will be achieved.	QC is a process which deliberates on fulfilling the quality request.
A QA aim is to prevent the defect.	A QC aim is to identify and improve the defects.
QA is the technique of managing the quality.	QC is method to verify the quality.
QA does not involve executing the program.	QC always involves executing the program.
All team members are responsible for QA.	Testing team is responsible for QC.
QA e.g. Verification.	QC e.g. Validation.
QA means Planning for doing a process.	QC Means Action for executing the planned process.
Statistical Technique used on QA is known as Statistical Process Control (SPC.)	Statistical Technique used on QC is known as Statistical Quality Control (SPC.)
QA makes sure you are doing the right things.	QC makes sure the results of what you've done are what you expected.
QA Defines standards and methodologies to followed in order to meet the customer requirements.	QC ensures that the standards are followed while working on the product.
QA is the process to create the deliverables.	QC is the process to verify that deliverables.

QA is responsible for full software development life cycle.	QC is responsible for software testing life cycle.

Table 1.3 Quality Assurance vs Quality Control

Manufacturing Process

Test automation plays a major role in Semiconductor manufacturing process. Test automation refers to the automation of the process to test a product through the use of machines. Depending on the product, the machines that we are referring to could mean a combination of Automatic Test Equipment (ATE), handler, interface board, and test program that drives the ATE, as with the case of the IC chip testing. Test automation is a big part of test engineering. Further this topic will be discussed in unit – V elaborately.

ERS – Electronics Repair Service

Electronics repair service is also a one of the part of test engineering where engineers create, develop, install, maintain and trouble shoot electronics system and equipments using manual testing or by using automated testing as per the needs.

They are the ones usually operating and maintaining complex systems and equipments. Their expertise is invaluable in the fields of defense, aerospace, commercial electronics, medical technology and telecommunications among others.

1.2 Principles of Fundamental Testing Methods

An electronic circuit consists of discrete and integrated semiconductor (semi-con) devices.

Discrete devices or discrete components denote passive components like Resistor, Capacitor, Inductor and active components like Diode, Transistor, FET, and LED etc.

Passive components do not require power supply and they often attenuate signals.

Active components amplify or process signal(s) and require power to operate.

Device testing can be done at two levels: functional and parametric.

A functional test determines whether or not the device works functionally for the intended use.

A parametric test measures all device parameters such as DC (Static) and AC (Dynamic) parameters to see if they meet the specified values. In the production of semiconductor devices, it is often the case that functional testing is done on all units while parametric testing is done on a small percentage of the units as test samples.

Manual Test verses Test Automation:-

Manual testing is the process where diagnostics are done with minimum measuring instruments with skilled trained test engineers. Manual testing is a good fit for smaller projects as well as companies without significant financial resources. Figure 1.3 Shows the Manual Testing using various
Measuring Instruments

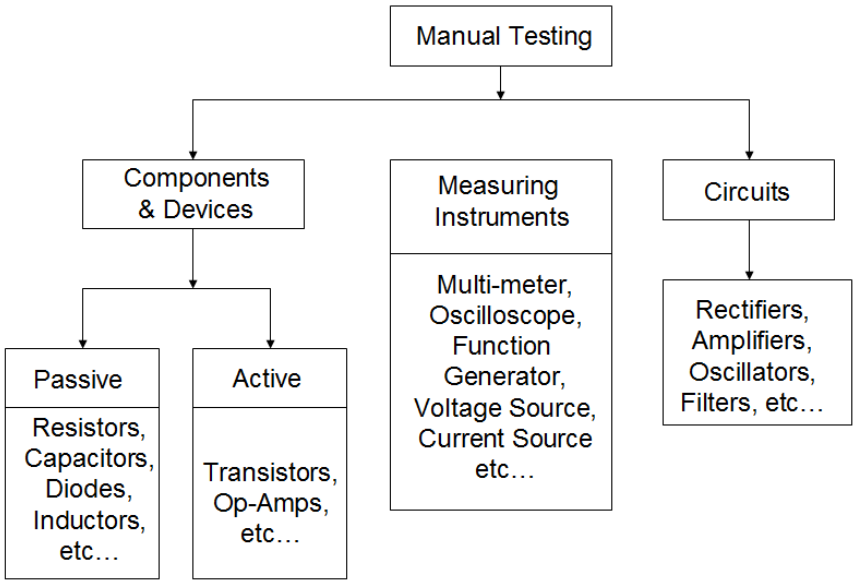


Figure 1.3 Shows the Manual Testing using various Measuring Instruments

The Above figure 1.3 is Show the Manual Testing using various Measuring Instruments Automated testing is the process through which automated tools run tests that repeat predefined actions, comparing a developing program's expected and actual outcomes.

Testing can be done through both automated and manual testing, but which one you choose comes down to the associated costs and benefits of each on your particular project.

Automated vs. Manual Testing: The Pros and Cons of Each

Automated Testing Pros:

- Runs tests quickly and effectively.
- Can be cost effective in the long-term.
- Everyone can see results.
- Useful in Industrial, Telecommunications, Defense and in Manufacturing Sectors where large number of volumes are tested.

Automated Testing Cons:

- Tools can be expensive.
- Tools still take time, time goes into developing the automated tests and letting them run.
- Tools have limitations.

Manual Testing Pros:

- Short-term cost is lower.
- More likely to find real user issues.
- Manual testing is flexible.

- Useful in Commercial Testing with minimal instruments.

Manual Testing Cons:

- Certain tasks are difficult to do manually.
- Not stimulating.
- Can't reuse manual tests.

1.3 Basic Principles of Memory Testing

ROM and EPROM testing:

ROM and EPROM testing is relatively easy. Since the contents are known in advance, all that the user has to do is to verify that all locations can be read from and that none of the original ROM content has changed. In principle, this can be done by manually stepping through each device location and comparing its content against the original documentation. In practice, this is impractical, and a more efficient strategy based on data compression is generally used. Using a data compression algorithm, the complete ROM content is reduced to single numerical value. This number is referred to as its "signature". Now, instead of having to cycle manually through the ROM to verify its content, we need only verify that its signature is correct.

To test using the signature verification method, a known good signature for each ROM is needed.

This is obtained by compressing the data from a known good ROM. These good signatures are then documented. During testing of a malfunctioning system, the signature for each suspect ROM is regenerated using the same procedure as was used to generate the original good signature. If the two signatures match, the ROM test passes.

In addition to checking ROM contents, this test inherently exercises all circuitry associated with ROM, including address and control lines, decoding circuitry, and the data bus interface. Thus a successful test not only verifies the ROM, it also verifies all its associated circuitry as well.

Generating a simple signature

The general method of generating a "good" signature is to compress the ROM contents into a checksum by summing its contents using an exclusive – OR summation. To generate such a checksum, start at the first location in ROM and successively add each word in turn until the complete ROM contents have been summed. Exclusive – OR addition is used instead of simple arithmetic addition, as it provides a higher level of uniqueness. This logic is used to generate the original "good" or "Gold" signature. The same algorithm is used in suspect system and the resulting signature is compared with "good" signature. The Simple checksum approach is not totally reliable. Although it detects all single – bit errors and many multiple errors, there are combinations of errors that it could miss. This method is useful for those in need of a quick and simple test.

An Improved version of the ROM test makes use of a Cyclic Redundancy Check (CRC) check character. This character is generated by a division process, rather than by the addition process described above. The CRC technique is inherently more reliable than the checksum technique and provides a much higher level of security.

CRC techniques are also used for signature analysis testing. In this case, however, data compression is usually performed by hardware rather than by software

RAM testing

RAM testing is basically to write a known test pattern in to the memory and read them back, report errors if a mismatch occurs. Comprehensive testing using all possible patterns would ensure that all patterns could be written to all locations and read back without affecting the contents of any other location. Such comprehensive testing is impractical, however, as it takes an enormous amount of time to test even moderate size memories. Instead, modern testing strategies use groups of tests, each concentrating on finding a subset of possible faults, but cannot guarantee 100 percent certainty.

Some common types of RAM tests are discussed below

Go-No Go Test

A fast way to test for gross failures is to write 0s to all locations, then read them back to ensure that no bits are stuck high, then write 1s to all locations, then read them back to verify that no bits are stuck low. It is used simply to verify the memory hardware interface.

Simple checkerboard Test:

Alternating 1s and 0s are written to memory in a checkerboard fashion, then read back and verified. The pattern is then inversed and the test is repeated. During the verification phase, exclusive-OR comparison of the patterns is used. This results in a bitmap in the accumulator showing the failing positions.

These tests check for hard errors plus failure of decode logic and other support circuits. They are sometimes used with a delay between memory fill and memory read to test for data retention. They do not catch many pattern-sensitive memory failures.

Galloping Pattern Testing;

To search for pattern sensitive and other soft errors, a group of “galloping patterns” test have been devised like “Walking 1s and 0s”, “marching 1s and 0s” etc.

Simple march Test:

This test checks for address sensitivity as well as some cell interaction. Basically, it tests whether each location is capable of storing 1s and 0s and whether writing to any given location disturbs data residing at other locations. The program makes two passes through the RAM address range for each bit-thus, for an 8-bit processor, 2X8=16 passes are the required. Test patterns are produced by shifting a single logic 1 bit through the accumulator.

The first pass for each pattern stores the selected pattern in all locations; on the second pass for that pattern, each location is read and checked to see if anything has changed. If the data check is okay, the location is

marked to show that it has been tested. This is done by complementing its contents. If there is interaction between cells, it will be detected when the test reached the address affected.

Random patterns:

Random pattern testing is sometimes used to test for pattern sensitivity. In such a test, a series of pseudo random numbers is generated by software, written to memory, retrieved, and then checked for errors.

The process is repeated over and over in the hope that pattern sensitive errors will be uncovered.

These tests also verify not only the memory, but also all associated address bus, data bus, and decode and control logic associated with the sub system.

Length of Time to test:

Time limitations are an important factor in memory testing. For some schemes, testing time goes up proportional to the number of memory locations N to be tested (N test), while others go up proportional to the square of the number of locations to be tested (N^2) test. For very large memories, test duration using N^2 testing can take a long time.

Functional Fault Models

Classical fault models are not sufficient to represent all important failure modes in RAM. Sequential ATPG is not possible for RAM. Functional fault models are commonly used for memories: They define functional behavior of faulty memories. New fault models are being proposed to cover new defects and failures in modern memories:

1.4 PCB Track Short Testing Methods

Short circuit faults on PCBs are some of the very common but difficult problems faced in PCB production lines and PCB Repair centers. As the density of PC boards has increased, so have occurrences of shorts between traces. These shorts can result from manufacturing errors where the copper has not been properly etched away and from solder bridges when the boards are assembled. And most of the times, we know that there is a short circuit between two nodes or adjacent tracks in the PCB but do not know where this is exactly located in the PCB.

The problem now is to locate the bridge if it cannot be found simply with a magnifying glass. The short circuit locator allows us to do this, even if the short is hidden under an IC or other installed components.

Normally when using the multi-meters on the whole shorted track, it will show the same impedance. Because, conventional meters cannot distinguish the milli-ohm ranges. Using dedicated Short circuit locator instruments, we can measure the resistance in milli-ohms range.

Concepts behind the Short Locator

As we know diodes are made from semiconductor materials such as silicon or germanium. In order to "turn on" and conduct current in the forward direction, a diode requires a certain amount of positive voltage to be applied across it. The typical voltage required to turn the diode on is called the forward voltage (VF). Show in the figure 1.4 Silicon Diode approximately 0.7V Germanium 0.3V

Forward Voltages varies depend upon the semiconductor materials used for Silicon Diode approximately 0.7V and for Germanium based diode may be lower, around 0.3V. The type of diode also has some importance in defining the forward voltage drop; light-emitting diodes can have a much larger VF, while Schottky diodes are designed specifically to have a much lower-than-usual forward voltage.

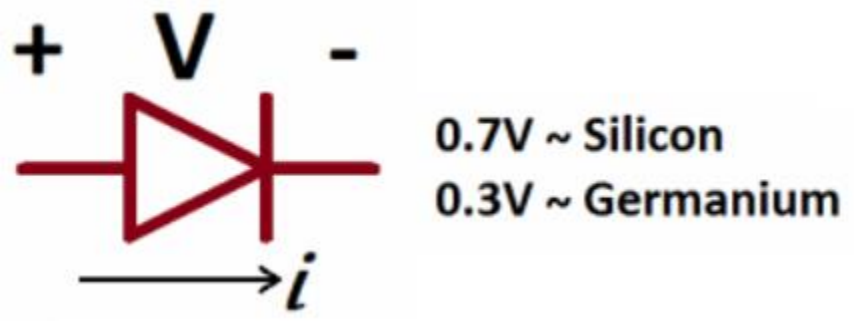


Figure 1.4 Silicon Diode approximately 0.7V Germanium 0.3V

To detect shorts in the populated PCBs, the drive voltage from the short circuit locator equipment should be less than 200 mV. This is to avoid turning on the semiconductor devices inside the circuit, which could lead to misleading results.

The concept is, the resistance will increase away from the short and it will decrease if it is towards the short. So the exact location of the short is where we get the lowest resistance value.

Short Locator is the exact tool that would be very useful to the technicians to pinpoint these faults. It overcomes the need to cut tracks and remove components trying to locate the fault. Before proceeding to understand the actual modes of operation, let us analyze the various reasons due to which the shorts occur in the tracks, components, etc.

How Short Circuit Occurs

Hair line and micro shorts: In highly dense PCBs it is possible that an illegal hair line has formed due to a manufacturing defect. Normally PCB
Manufacturers test them electrically with the use of expensive equipment to
Detect and reject such PCBs. But in small volume production of such PCBs it may not be cost effective in going for a electrical test and hence this defect may go unnoticed and trouble emerges when these PCBs are assembled and tested for its functionality

During Component assembly: While the components are assembled and soldered it is possible that a solder bridge between two adjacent component terminals is formed and causes short circuit.

Defective Components: Capacitors, especially tantalum types can get shorted internally and can cause Vcc-Gnd shorts. Semiconductors like diodes or transistors can get shorted fully or partially

Ideally PCB tracks (electrical conductors) are supposed to have zero resistance, but in actual conditions they have small resistance. For a 15 mils track this could be in the range of 30 milliohms / inch, whereas for 8mils track this could be double. The exact location of the short is where the instrument reads the lowest value of resistance and this resistance could be detected with a sensitive instrument like dedicated short locator, even in power planes.

Short Locator can also be used in measuring the contact resistance of switches and relays in its milli-ohm mode. Shorts Locator is mainly used for detecting and locating short circuits in printed circuit boards (PCBs).

Using Multi-meter and other conventional troubleshooting tools, we can identify the shorts and opens in a circuit. But it may be difficult to identify the exact component responsible for the shorts, especially when the components are connected in parallel in the circuit. And, if a partial short occurs in a component /track, it is highly impossible to detect shorts using conventional tools.

So we have to go for advanced PCB troubleshooting tools such as Short circuit locators. With these tools and an understanding of the circuitry involved, the test engineer can pinpoint the cause of the fault with a high degree of certainty. And that is where dedicated short locator are used for accurately locating shorts between Vcc and Gnd, hairline shorts and also helps in isolating fan-out problems, etc. Needless to say, it pin points the shorted components or tracks.

1.5 Concepts of Trouble Shooting PCBs

The PCBs are expected to give trouble free performance over their expected life duration. A factor like MTBF (mean time between failures) is calculated for a product based on the reliability factors of its constituents. This helps in planning a replacement or expected maintenance involving purchase of spares for the product.

The Mean time between failures (MTBF) is calculated as $1/\lambda$. where λ is the failure rate. Similarly, Mean time to repairs MTTR is calculated as $1/\beta$ where β is the repair rate.

The fraction of time that a system is operating normally (i.e. failure free) is the system availability, which is given by

System availability=MTBF/(MTBF+MTTR)

Any Electronic system may fail in its useful lifetime and it is unpredictable. Unlike Mechanical systems, where one can examine tear and wear and estimate the life expectancy and periodic service requirements, in electronic systems no one can predict when a semiconductor will fail. It may work for its entire life cycle without failure OR it may fail within few hours of its operation.

Though no one can predict when the semi-con device will fail, certain things can be predicted over the age of the circuit boards, they are;

1) Electrolytic Capacitors drying out and changing values over aging.

2) Low quality connectors (not gold coated contacts) getting rusted and not making proper contacts.
3) PCBs without conformal coating and exposed to humidity / air with salt (near sea shores) can corrode and PCB tracks may get open circuited.

Though advances in Semiconductor technology and reliable manufacturing process have considerably reduced the failure rate, still there are failures in the field and it becomes necessary to understand their failure pattern and build expertise to isolate failing components and subsequent repair so that these can be maintained at minimal cost and in shortest possible time.

Repairing PCBs helps in many ways;

1) Low cost of maintenance, replacing entire PCB will cost multifold
2) No need to stock and maintain spare PCB inventory except for few vital ones
3) Reduces e-waste and saves environment
4) Generates employment and develop expertise
5) Generate feed back info on failure patterns and often failing components, which can help improve design and manufacturing technology for more reliable PCB production as well eliminate un-reliable components / suppliers.

Electronic Circuit Boards fail due to many factors both external and internal of the system. It may fail due to external factors such as erratic input power, faulty power supplies, Lighting, static discharge, Excess load or short circuits at output stage etc., The circuit may fail due to internal factor like a faulty resistor or capacitor or failed semi-con junction or an erratic function of an IC in the control circuit.

This first level of maintenance will be to isolate the faulty circuit card with a spare card so that the system can be up and running, the next step will be to get the circuit card repaired by using various trouble-shooting techniques so that exact failing component can be pin pointed and replaced to make the circuit card back to life. Electronic systems may range from simple gadgets to large systems running complex applications**.** Troubleshooting a large system involves considerable time and resources.

A logical approach to troubleshooting saves valuable time and waste of resources. With such a procedure we can trouble shoot any electronic equipment regardless of its complexity.

Let us discuss some steps involved in troubleshooting. A good understanding of these techniques is essential for a Test Engineer.

1. Symptom Recognition.
2. Symptom Elaboration.
3. Listing of probable faulty functions.
4. Localizing the Faulty function.
5. Localizing trouble to a circuit.
6. Replacing the faulty and verification.
7. Failure analysis

Symptom Recognition

A trouble symptom is an indication of some disorder or malfunction in an electronic equipment. Symptom recognition is the act of identifying such a sign when it appears. The knowledge of the normal equipment condition enables us to recognize an abnormal condition.

Symptom Elaboration

Symptom elaboration is the process of obtaining a more detailed description of the trouble symptom.

For example, consider a LCD Display. If its screen is not light, then recognizing that its screen is not lighted is not sufficient information to decide exactly what is causing the trouble but leads to look into the associated circuitry. The symptom may mean that the LCD display is burnt out or there is some disorder in the internal circuitry associated with display or intensity control is turned down too low or the power supply is not correct or even that the equipment is not turned on. Thus so many faults may cause a symptom like this.

In addition, manipulation of operating controls may have caused the damage to produce a symptom. Knowledge of the circuit changes that take place when we adjust a control will enable us to think ahead of each step and anticipate any damage that the action might produce.

Symptom elaboration cannot be accomplished unless the observed displays can be completely evaluated. The easier method for performing this evaluation is to have all the data handy for reference by recording the information as it is obtained. This will enable us to compare with information contained in the operating manual of that equipment. The data recording is an important tool in troubleshooting.

Listing of Probable faulty functions

This step applies to equipment that contains more than one functional area. It allows us to mentally select the function or functions that probably contain the malfunction as indicated by the information contained in the previous steps. This requires knowledge of the various functions of the equipment.

The functional block diagram helps us to arrive at the decision. The equipment functional block diagram is a symbolic representation of the functional units within the equipment as well as the signal flow paths between them. We can select a functional unit and ask the question whether fault in this functional unit could cause the original trouble symptom? If the answer is yes, then we ask the question, whether a fault in this functional unit produces the associated information obtained during symptom elaboration? If the answer is yes, we list this unit as probable faulty unit and go on to consider the details of this unit. If the answer is no, then we move to another unit.

Localizing the Faulty function

We may decide one or more functional units as the probable cause. We have to localize the faulty functional unit by systematically checking each probable functional unit finalized in the previous step.

The faulty functional unit may consist of many circuits. We have to conduct extensive testing to isolate the trouble to a specific circuit within the faulty functional unit.

Bracketing Technique

A technique called Bracketing technique is used to narrow down the trouble to a faulty circuit. This checking requires using the circuit diagram, which gives information about input and output of each functional block and the test point's. This diagram also gives the interconnecting cable details which will help us to conduct the checks.

Localizing the trouble to a circuit

When we place brackets either mentally or in the circuit, at the good input and at the bad output of the faulty function in the individual servicing block diagram, we know that the trouble exists somewhere between the brackets. The idea is to move the brackets one at a time and then make tests to determine whether the trouble is within the new bracketed area. We continue this process until the brackets isolate the defective circuit.

The most important factor in bracketing is determining where the brackets should be moved in this narrowing down process. The determination is made on the basis of

- Our deductions form the analysis of systems and previous tests.
- The type of circuit paths through which the signal flows.
- Accessibility of test point.

Now we have understood the signal paths that will help us to use the bracketing effectively.

There are 3 types of signal paths

- Linear path.
- Branching path.
- Switching path.

Linear Path: The linear path is a series of circuits arranged so that the output of one circuit feeds the input of the following circuit. Thus the signal proceeds in a straight line through the circuit group without any return or branch paths.

Branching Path: The branching path may be either of two kinds: divergent or convergent. A divergent path is one in which two or more signal paths leave a circuit.

When two or more paths enter a circuit, the path is known as a convergent path.

Switching Path: The switching path contains a relay circuit that provides a different signal path for one or more given situations.

Tests are conducted in two modes:

- Signal tracing is accomplished by examining the signal at a test point with an oscilloscope, voltmeter or some other electronic test device.
- Signal injection is a method of applying an artificial signal to circuit to check its performance.

An example of the bracketing procedure can be shown by referring to following figure which represents a faulty functional unit. The Localizing the faulty function step, indicates a good input at test point A and a faulty output at test point E, indicating a trouble between these two test points.

Localizing the fault

Signal tracing would normally be used, since the signal at point **A** is known to be good. The signal at test point **C** is the first logical place to check, since this check will eliminate half of the circuit. The signal at test point C can be checked by placing a meter or scope at this point.

If the signal is satisfactory, the input bracket is moved from point **A** to point **C**. The next check at point **D** will then isolate the defective circuit. If the test is satisfactory, the trouble is in circuit **4**& if the check is unsatisfactory, the trouble in circuit **3**. Only the input bracket was moved.

If the check at point **C** is unsatisfactory, the output bracket is moved to this point. The next check at test point **B** will then isolate the defective circuit. If the signal is satisfactory, the trouble is in circuit **2**; if unsatisfactory, the trouble is in circuit **1**. In this case, only the output bracket was moved.

If signal injection is used, a test signal is injected at test point **C** and the output checked at test point **E**. If the results are satisfactory, the output bracket is moved to point **C**, and a signal is injected at test point **B** to isolate the defective circuit. If the signal is not correct at point **E**, the trouble is isolated between points **C** and **E** and the input bracket is moved to point **C**. Injecting a signal at point **D** will then isolate the defective stage.

The half split method

The first method of bracketing to be considered will be the method for linear circuits. The best method of trouble shooting this type of circuit path is the half-split method.

Assume we have brackets at the input and output of a number of circuits in which the signal path through all circuits is linear. Unless the symptoms point to one circuit in particular that may be the trouble source, the most logical place to move a bracket is to a convenient test point near the center of the bracketed area. If a test indicates that the signal is good at this point, an input bracket should be left there. The brackets will then surround the second half of the linear circuit path, and the other half will be eliminated from the trouble area. If an incorrect signal is found at test point, an output bracket placed at this point will show that the trouble exists in the first half of the linear circuit path. This process should be repeated within the area now enclosed with brackets until the brackets surround only one circuit- the faulty circuit.

Branching Path method

The first steps in the bracketing procedure for branching circuits should be isolating the trouble to one signal path.

The branching paths should be checked until the trouble is isolated to one of the signal paths. Then the half-split method may be applied to the faulty linear signal path.

Switching path method

The last type of signal path is the switching path. We have seen how electronic equipment are composed of various circuit chains interconnected to perform a desired task.

Control of these circuits along a switching branch, we initially test the final signal output for the branch following the switch or relay. When the switch or relay is a multiple contact type, each contact may be connected to a different circuit branch. In this case, it may be necessary to place the switch or cycle the relay to each position and check the final output of the branch associated with that position.

Once we have performed this test and isolated the trouble to one or more branches, we should check the suspected branches to locate the faulty branch. The next step is to apply the half–split or branching method, as required to isolate the faulty circuit.

In general, two or three or even all of the bracketing techniques may have to be used in combination, since most electronic equipment are complex enough to contain several circuit arrangements.

Each test should attempt to eliminate as many circuits from the bracketed area as possible, unless the symptoms indicate that one or more circuits are more likely to be at fault than the rest of the circuits. This process is continued until the brackets are at the input and output of single circuit.

Component Failure Analysis

This final step in the six-step troubleshooting procedure will require testing certain branches of the faulty circuit in order to determine where the faulty component lies. These branches are the interconnected networks associated with each element of the device in the faulty circuit.

A faulty component in one branch of a circuit can be caused by a faulty component in another branch or even in another circuit.

This step is designed to locate and verify all faulty components in the equipments being tested.

The schematic diagrams are helpful in this step. Schematic diagrams illustrate the detailed circuit arrangement of electronic component (represented symbolically) that make up the complete circuits within the equipment or functional unit. These diagrams show what is inside the blocks on an individual servicing block diagram and provide the final picture of electronic equipment.

Types of circuit trouble

The equipment failure can be classified into three types:

Complete failure of equipment will be obvious from the failure of function of the equipment

Degraded performance in which the normal symptoms will be available but the function will be below the expected levels. For example, the voltage output of a circuit may be 3 volts instead of expected 5 volts.

Intermittent failure refers to something that alternatively ceases and begins again. A component operates normally for a period of time, then fails completely or operates on a degraded level for a while, and then returns to normal operation. The cyclic nature of this malfunction is an aid in determining that it exists; however, it is often difficult to locate the actual faulted part.

Review of the previous data

A review of all the symptoms and test information obtained thus far will help to isolate other faulty components, whether the malfunction of these components is due to the isolated malfunction or some entirely unrelated cause. Suppose we have isolated a trouble to component and then discover that a failure of this

component would be not result in all the symptoms and irregularities in our test data. Then there is another defective component. When one component fails, it often results in abnormal voltages or currents that will cause damage to other components.

It is very important that isolating and replacing a malfunctioning component alone is not sufficient. We should analyze what is the cause for this malfunction. For example, we have isolated a particular transistor and found it to be burnt. In addition to replacing it, we should study what could have caused this transistor burnt. Then only, the trouble shooting will be complete. This also forms a feed back to the designer or manufacturing process in correction and in reliable equipment design.

Failure Types and Causes in the circuits

We have to be aware of the most important failure types and their causes in digital as well as analog circuits that can be troubleshoot using VI trace Technique. Three basic types of failure can be distinguished during the operation of digital circuits.

Static Failures (Stuck-At Failures)

These failures appear in the same manner, either during operation or testing, and are independent of the stimuli and their variations.

Breaks in Circuit Tracks (Stuck –At Open)

The absence of a necessary electrical contact causes a permanent fault.

Stuck At Short

An unintentional short exists between one or more tracks and causes a permanent fault.

Stuck At High/ Stuck At Low faults

These permanent faults are also constant from when they first appear up to their removal and are independent of any simulation signals applied to the test object.

Two distinct cases exist: The potential at the faulty point in the circuit or connector pin remains either at the level of a logic high or logic low.

These are called Stuck at low or stuck at zero (S-A-0) for stuck to logic low, or stuck at high or stuck at One (S-A-1) for stuck to logic high**.**

The following possibilities can be taken into consideration for S-A-0 or S-A-1 Faults:

- Short circuit to ground for S-A-0 by either internal to the device or external by other components in the net.
- Short circuit to the supply rails for S-A-1 by either internal to the device or external by other components in the net.

Dynamic Faults

This category of faults is caused by components or circuit configurations which are used at the limits of their specifications, but which do not affect the function of the product at the time of testing. Again the following categories can be distinguished.

Supply failures;

The faults in the product are caused by excursions of the supply voltages outside their specified limits during operation. While testing with automatic testers, the voltage can temporarily assume the correct values. This leads to the conclusion the test object functions properly.

Timing Faults

A failure of the product is caused by faulty timing which falls outside the specified limits. This is especially important for synchronous circuits which are clocked (slow rise or fall times). Varying tolerances of devices lead to timing errors in synchronous circuits.

Parametric Faults;

In general, parametric errors appear when one or more parameters of a component or a circuit in normal operation fall outside their specified limits. These deviations are noticeable as faults if two components cannot work with each other. For example if the Input bias current of a device exceeds its specification, it will draw more current from the previous device, which may be in another circuit card and hence this type of fault is only discovered in normal operation and not during individual testing of circuit cards unless the equipment used to test the individual cards can check the parametric values of the edge devices. These type DC and AC parametric faults can cause trouble on system level test though individual circuit cards may be functional. DC Parametric Tests include Input Current, Output Fan out capability and Tri-State leakage currents. AC parametric Tests include, rise / fall time, propagation delays, set-up and hold times of devices. Necessary real-time tests complicate the fault tracing considerably. Parametric faults can also be caused by environmental influences, such as temperature changes or erroneous adjustments of viable components. These faults are not very frequent.

Intermittent Faults;

Intermittent Faults appear only occasionally, either at regular intervals or sporadically and vanish again. This kind of fault is very difficult to recognize and locate.This type fault requires extravagant means of measurement (constant supervision and under certain circumstances, even special environmental conditions)

Breaks in Circuit Connections;

Breaks in circuit connections can occur because of thermal expansion, contraction orSpecial variable mechanical loading such as torsion, bending, or vibration

Short circuits

In a similar way to open connections, short circuits between neighboring tracks can or connectors can lead to intermittent faults because of changing conditions in the environment.

Exceeding Limits;

Intermittent faults can also be caused by components which are used at the limits of their specifications.

Bad Solder Joints;

Dry or Cold solder joints can lead to intermittent faults

Thermal and Electromagnetic Influences:

A further cause for intermittent faults can be found in a short exposure to heat and electromagnetic radiation.

Fault types and fault Causes in Circuits during Manufacture

In addition to functional faults which can appear during normal operation, faults are known which occur during the manufacturing process. It is possible to distinguish between classes of faults as processing faults and population faults.

Processing faults

Processing faults manifest as breaks in copper tracks or as short circuits (caused by inadequate etching), or they lead to inadequate solder joints during the population process.

Population faults

Population faults occur because of;

- Using wrong components
- Wrong orientation of correct components (diodes or integrated circuits)
- Omission of components during the population procedures
- Using the correct components but with wrong values

Depending on a particular cause, manufacturing faults can, equally lead to static, dynamic or intermittent faults. This is especially true for bad components which are pushed to the limits of their specification or which are already defective.

Failure Frequencies

The following table shows a characteristic example for the distribution of faults in the manufacture of building blocks. This is only an illustration and actual % will vary with assembly and manufacturing process and incoming QC of the components.

Fault Type	Fault Cause	Distribution
Population fault	• Missing components • Components wrongly inserted	20%

	• Swapped components • Wrong component value	
Manufacturing Fault	• Circuit track breaks and short circuits due to etching • Open IC sockets • Soldering fault • Contaminated contacts	60%
Functional fault	• Defective components (10%) • Faulty interaction of components (10%)	20%

Table 1.4 distribution of faults in the manufacture of building blocks

Functional faults, which can appear during tests in manufacturing are mainly made up from the entries in the following table.

Fault Type	Fault Cause	Distribution
Static faults	Defective components (predominantly –SA-1 and S-A-0)	50%
Dynamic faults	Timing faults (28%) -One shots too long or too short -Oscillatory output -Slow edges Driver faults (21%)	49%
Intermittent faults	Component at tolerance limits	1%

Table 1.4 Functional faults

The following table shows some examples for typical failure rates of various components.

Component	Failure rate (%)
IC , linear	3-10
IC, digital	1-5
Transistor	0.75-2
Diode	0.2-1
Capacitor	0.1-1
Resistor	0.05-1

Table 1.5 fault types and troubleshooting methods

Thus we have seen in detail the fault types and troubleshooting methods in electronic circuits.

Manual and Automated PCB Trouble Shooting Techniques

Troubleshooting modes

The manual troubleshooting can be classified as Online troubleshooting and Offline trouble shooting .

Online trouble shooting refers to trouble shooting with the board online – connected to the system. Normally using board extenders if the PCB is in a rack mounted. This means you are working with actual signals and working voltages. All safety considerations have to be taken care in this mode. You are supposed to have knowledge about the working of the board and its components. This mode can be useful when the cause of the trouble is known.

Offline trouble shooting refers to trouble shooting with the board taken out of the system. Now you have choice of feeding required signal at the required place and the function of the various stages of the board can be monitored using complex procedures.

The options include isolating/changing a suspected component. The basic factor of troubleshooting in minimum time and optimum use of tools/instruments to bring back a PCB online should be borne in mind.

The manual trouble shooting is limited to repairing board in small volumes as it is very difficult to isolate a device from the other devices and test all possible combinations for all the devices in the board. Also this demands skilled persons experienced in the particular board.

The time to find fault varies, depending on the complexity of the Circuit Card / Available documents / data sheets / experience and knowledge level of the test engineer and so on. This can be from few minutes to few hours to few days or even weeks. This is the greatest dis-advantage of manual trouble-shooting as even after spending a week, one may have to give up the trouble shooting if cannot find the faulty component.

When faced with a malfunctioning or inoperative system, the initial troubleshooting stage is very important. Since troubleshooting is both an art and a science, there is no single, unique way to go about any given problem. There is, however a general order of progression that makes sense to follow, as certain steps and procedures naturally follow after others. We should remember to always look at and correct obvious faults first.

Guidelines for Troubleshooting

An approach that has been found useful by the skilled test engineers for troubleshooting products in the general service environment is outlined below.

It recognizes that a large percentage of problems can be identified by visual inspection, probing and self test.

1. Perform a visual inspection
2. Reseat the boards and cables under the power off condition
3. Check the power supplies
4. Run the system self test if available. If the self test indicates the general area of fault, try “board swapping” if spare boards are available. If the fault cannot be localized using the self test, try minimizing the system, and if it works, build it up a little at a time until the fault re-appears.
5. When the problem is isolated to a particular board or section of a board, we should begin our investigation as follows;

Step1 Visual Inspection;

Check for blown fuses and replace if necessary Look improperly seated connectors, loose or broken wires, wrongly set switches, missing jumpers and so on.

Check for obviously damaged components (i.e. components burned or discolored due to overheating). Check both at the component and at the printed circuit board, since overheating is sometimes not visible on the components but shows up as brownish on the PC card instead.

Step 2 Reseating Boards and components

Edge connectors are a common source of problems. Power down the system and examine the edge connectors for contamination and clean them if necessary. Reseat any loose components (such as socketed ICs). Rest the board and power up.

Step 3 Power supply Checks

Check the power supply for missing voltages out of specifications. As voltage specifications are quite tight, logic supplies must be within +/- 5 percent (maximum tolerance limit) of supply voltage. We should remember that many systems use multiple supplies and all must be checked. Care should be taken to make such checks on the circuit boards at their points of usage, not just at the terminals of the power supplies. Because breaks can occur in interconnection paths due to loose or broken wires, poor solder joints, and so on. Checking the power supply quality for noise and ripple is also very essential.

Step 4 Self-Testing

The equipment may contain self-test routines that let the system test itself. Note however, that for these to be used; at least the main processor of the system must be functional and able to execute code.

ROM based self- test diagnostics. Routine may be supplied in ROM. These generally provide a quick check of ROM and RAM and a simple accessibility test of programmable switches and indicators are often used to involve test sequences and to display pass/fail results.

Step 5-Board Swapping

Board swapping involves the substitution of a known good board for a suspect one. If all the power supply voltages have been verified in a previous step, board swapping should be possible without a great deal of hazard to the good bard. Power should be removed before removing and inserting boards. When a faulty board has been identified, we still face the problem of how to find the actual part or component failure. The problem with board swapping as a service strategy leads to some other difficulties due to availability of good boards. Because removing a good board from the plant may affect the working environment.

VLSI Component swapping:

VLSI devices are very difficult to troubleshoot without advanced automated test equipment because of their complexity. In many systems, these devices are socketed for easy replacement. If so and if spare components are available, swapping VLSI devices in the suspect area can be done. It takes very little time to do this and can save a great deal of investigative time and effort. However, caution is required, as VLSI devices are very susceptible to damage from static electricity. Even if the device is not destroyed outright, static may cause problems that result in progressive degradation and ultimate failure, perhaps weeks or months in the future.

Swapping should therefore be done with a great deal of care and in a static -free environment. This should include a grounded, antistatic table mat and wrist strap, at the very least.

That's why we need Advanced Automated test equipment for troubleshooting where testing LSI and VLSI devices are possible in-circuit condition without removing or replacing it from the board.

Gross Signal Checks

Gross signal checks can be performed using a common oscilloscope. Checking for clocks, verifying logic levels, rise and fall times, and frequency according to data sheets for appropriate specifications. We must ensure that clock signals appear correctly on the buses, not just at the output of the clock circuits.

Oscilloscopes may also be used to investigate general system behavior. However, since they are inherently incapable of synchronizing to the complex signals present in systems, they cannot provide an accurate picture of such activity and are thus basically useful only for things that are periodic or obviously wrong.

Checking for the presence of signals on the various buses (address, data and control) and verify that they look reasonably correct that is, have signal activity on them if they should. (Some lines may not have activity on them because none is called for. For example, if the system ROM is located at the low end of memory and activity on high –order address lines even if there are no faults).

If general activity is present, check for the presence of decoded outputs, such as chip selectors, output enables and so on.

1.6 Automated Troubleshooting Technologies

Automated Test Equipment (ATE) helps the test engineer to find faulty module of an electronic system through an automated testing process.

The main advantage of this automated test is that the predictable time determining if the module is good or bad (Go – No Go Test) and if the module is faulty, to detect a faulty component. This is unlike manual trouble-shooting which takes indefinite time and depends heavily on the skill set of the test engineer. It involves a systematic procedure and logical steps to identify a faulty component with speed and accuracy thereby saving time at a faction of replacement cost of the module.

In the production field, it is mostly preferred especially with medium to high volume production, since it provides an effective method of testing circuit boards at minimum cost and in shortest time.

Basically an ATE system, which can be simply a tester used to test bare boards, components, or fully populated PCBs, comprises a computer-controlled test hardware with test instructions stored as software in the computer. The test pattern is applied to the board under test using test fixtures. This hardware interface, which can be either through Edge connectors, clips & probes or "bed of nails", provides an accurate low resistance contact to the circuit nodes of the board under test. The ATE will also have diagnostic facilities, embedded in the software, and usually a standard bus structure so that other COTS (commercially available off the shelf test system) can be interfaced through an analog highway switch matrix to carry out functional tests if required.

A variety of ATE systems at various levels of cost, complexity and diagnostic ability are available.

The different types are

- Component testers.
- Bare board testers.
- Manufacturing defect analyzers.
- In-Circuit Testers.
- Functional testers.
- Combinational testers.

Component Testers;

These are instruments such as DMMs, LCR Meters and IC testers which can test individual components out of circuit and these have been discussed in the previous chapters.

Bare Board Testers:

These are automated test equipment to eclectically test a bare, unpopulated circuit board after it is manufactured but before it is populated. The type of test ranges from simple continuity test to insulation test or even trace impedance and cross talk checks

Manufacturing Defect Analyzers:

These are automated test equipment with a test fixture, often involving a bed of nail test or Robotic Flying Probers. The primary function of these tester is to detect all manufacturing process faults and are done before the populated PCB is powered up for further tests including functional test.

Process related faults such as solder bridges / open circuits, detecting missing or wrongly oriented passive, checking component values are the primary functions of this kind of testers.

In-Circuit Tester:

These are automated test equipment similar to Manufacturing Defect Analyzers with additional capacity to test digital and analog circuits in actual in-circuit conditions by use of back driving or force driving test data and evaluating the output response of the digital and analog devices by the use of library data and simulators. These testers test individual devices in isolation for its functionality.

Here the principle is that if each and every component are tested for its functionality or value, then the chances of the board functioning are high. The only draw back is that the integrity of the entire board and its function as a whole is not tested.

Functional Testers:

These are designed to test entire circuit from edge connectors for their functionality by use of Digital and Analog I/Os, net-list information. Device Library Data and simulators. Use of external instrumentation and ability to learn and compare output response are additional capabilities.

Combinational Testers:

These Automatic Test Equipment combine all the capability of MDA, In-Circuit and Functional Testers. Often they use multiple test strategies such as digital simulation, analog simulation, ability to learn and compare both digital and analog responses and sometimes combination of simulation and learn and test capability. Additional facilities for RF and Audio / Video Signal Test, Digital Signature and analog signature test etc. Show in the Figure 1.5 Most common failures on PCB assembly in Production Floor

Before we go into the various types of ATE systems, we will briefly look into the types most common faults that can occur on PCBs Most common faults on PCBs after they are populated. Show in figure 1.5 Most common failures on PCB assembly in Production Floor

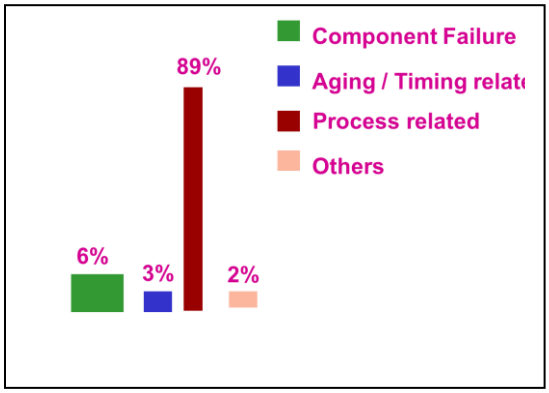


Figure 1.5 Most common failures on PCB assembly in Production Floor

As you can see from the above chart, here the most common failures are process related – such as missing, misplaced, wrong value, mis-oriented components and solder problems like solder bridge, open circuits due to insufficient solder etc.There could not be any faults due to component ageing but may be due to under specification of DC and AC parameters. Component failures are quite low, they can be further improved by in-coming Q/C.

Next we will see the most probable failures on a field return PCB:

Show in the figure 1.6 Most common failures on PCBs from field return

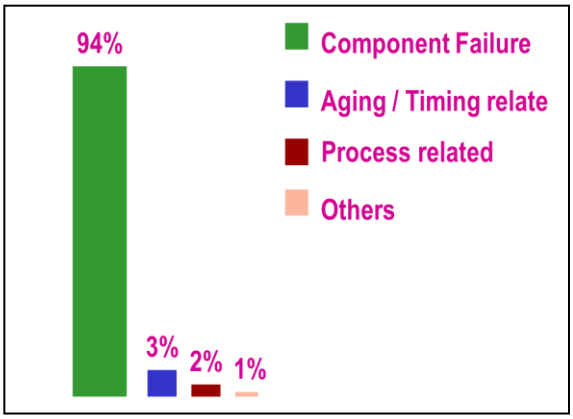


Figure 1.6 Most common failures on PCBs from field return

Here, the most probably failure is a component failure as the PCB had been working all along and have suddenly failed. Other possibilities are component degradation due to ageing and hence timing related problems. Process failure should be very minimal and chance of open and shorts are slim unless there is physical damage. But if the same PCB is re-worked multiple times in an unprofessional way, it could lead to more faults as illustrated below; Show in the figure 1.7 Most common failures on PCBs from field return after multiple unprofessional re-work.

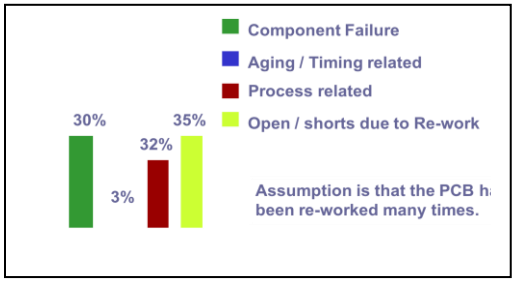


Figure 1.7 Most common failures on PCBs from field return after multiple unprofessional re-work

From the above data, one can easily choose what kind of tester is ideal depending on where it is used.

If on production floor, then of-course an MDA will be the first choice, whereas if the tester is used in maintenance then an In-Circuit Functional tester or board functional tester will be ideal.

Now we will see the various techniques used by ATEs in fault detection and isolation on Digital / CPU – Microprocessor / Analog circuit cards or PCBs.

These include;

a) Emulation techniques
b) IC comparator techniques
c) In-Circuit Functional Test of Digital and Analog Devices
d) Learn and compare methods – Analog Signatures,
e) Learn and Compare techniques – Digital Signatures.
f) Board Functional Test,
g) IEEE standard Boundary Scan Technique

REVIEW QUESTIONS

Part –A (2 MARK)

1.How to teach the concepts of such an important subject like Test engineering most effectively?
2.DefinitionBlack Box Testing.
3.DefinitionWhite Box Testing.
4.What is mean by PRD?
5.What is mean by MRD?
6.Expand DFT & DFM.
7.Expand MTBF.
8.What is System availability?
9.What are the types of signal paths?
10.What are the methods in linear circuits?

PART B (3 Marks)

11.What are the types in equipment failure?
12.Types of circuit trouble
13.What are the S-A-0 or S-A-1 faults?
14.What are the Population faults?
15.What are 3 types of signal paths
16.Explain Online trouble shooting Off line trouble shooting.
17.Explain about Automated Troubleshooting Technologies

PART C (10 Marks)

1.Explain the working principle Pin Electronic with a neat diagram.
2. Write a short note on history of boundary scan.
3.What is the principle of boundary scan architecture?
4.Explain basic boundary scan cell with neat diagram.
5.Briefly explain the IEEE 1149.1 device architecture with neat diagram.
6.What is the difference between 'Boundary Scan Test' and 'Scan Test'?
7.Describe the principle of auto compensation with a neat sketch.
8.Briefly describe how to deal with signature difference with neat diagram.
9.Describe with a neat sketch the effect of open and Short in VI techniques.
10.What is Boundary-Scan explain with neat sketch?
11.List the JTAG TAP Interface signals and explain each.
12.Explain JTAG and its uses in details.

UNIT 2 AUTOMATED TESTING TECHNIQUES

Functional Testing Notes – Draft

This notes organized in 3 sections.

Section1: Component Level Testing(In circuit Functional Test ICFT &Out Circuit Functional Test OCFT)

- In-circuit Functional Testing (ICFT)
- Out Circuit Functional Testing(OCFT)
- Inputs required for Functional Testing
- Simulators
- Special considerations
- Difference between ICFT Vs OCFT

Section 2: Board Level Testing

- Input requirements for board level testing
- Simulation
- Learn & Compare technique
- Go/NOGO Testing
- Diagnostic (Guided Probe Back tracking)

Section 3: Conclusion & Questions

Section 1: Component Level Testing(In circuit Functional Test ICFT &Out Circuit Functional Test OCFT)

2.1.1 In-circuit Functional Test ICFT

- Printed Circuit Boards (PCBs) populated with devices can be directly tested in-circuit condition.
- No need of Test Program Set (TPS).
- No need of removing the components from the board.
- Localized functional testing using a library.
- Power on Test.
- Uses IEEE Std VHDL Based Test Language for Model Creation.
- Ability to test as wired condition with auto compensation applied on the fly, eliminating the need for creation or customization of the Test Program according to In-Circuit conditions.
- Ability test LSI / VLSI devices in-circuit, Digital/Analog & Mixed signal devices.
- Generally the devices are accessed through clip/probe/fixture interfaces.
- Supports test and repair of undocumented boards.

Out Circuit Functional Test OCFT

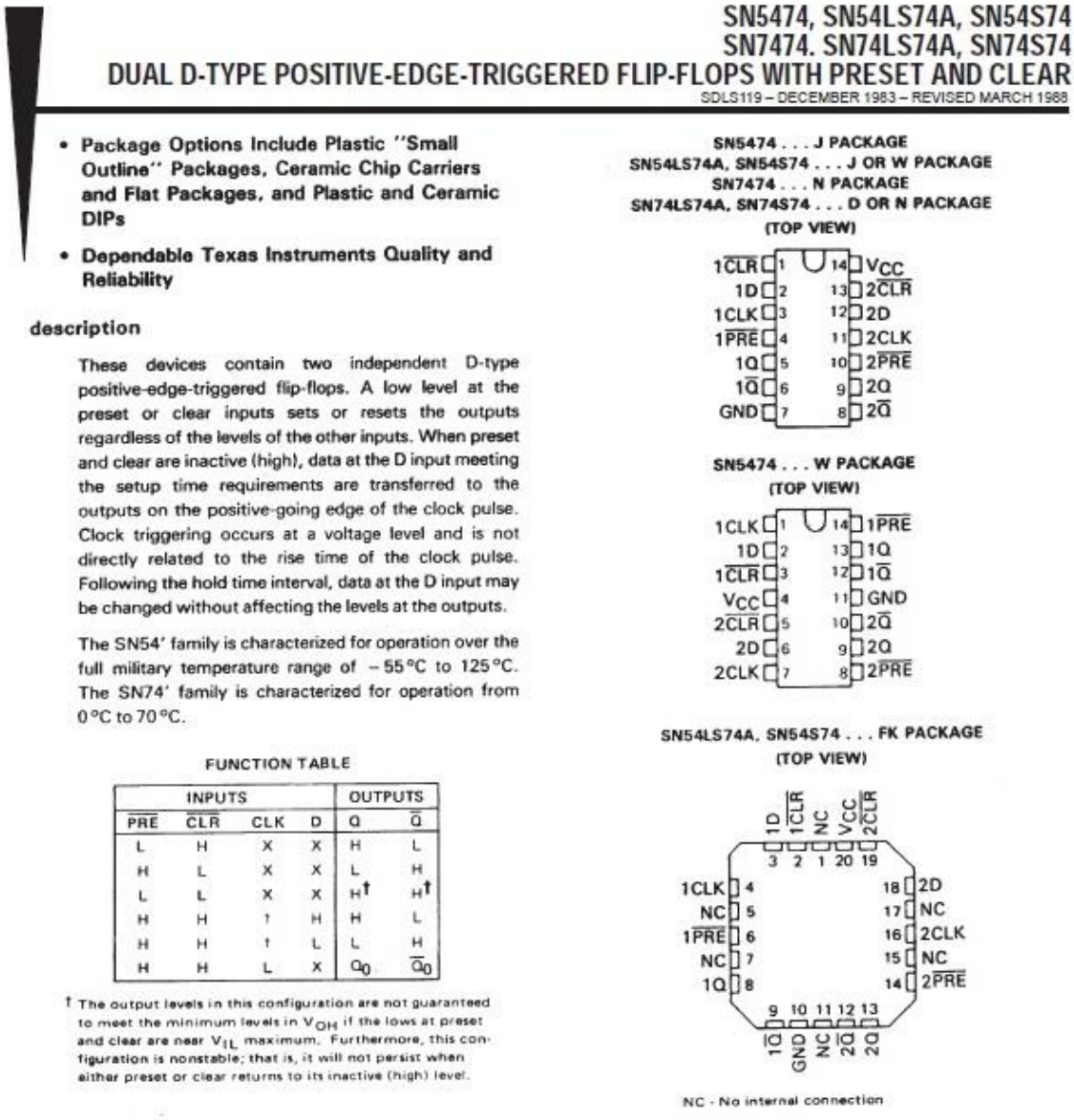
- For testing loose devices.
- Here functional testing of components are carried out in out-circuit condition using adopters.

More differences between OCFT/ICFT described later.

Device Data

The data sheet of any device provides the following information used for testing

a) Device Name
b) Device description
c) Aliases
d) Packages
e) Input/output/bi dir/tri state information along with labels
f) Operating temperature
g) Function/truth table
h) Initialization sequence
i) Logic diagram/symbols
j) Power supply details
k) Threshold levels
l) Time base information.



SN5474, SN54LS74A, SN54S74
SN7474, SN74LS74A, SN74S74
DUAL D-TYPE POSITIVE-EDGE-TRIGGERED FLIP-FLOPS WITH PRESET AND CLEAR
SDLS119 – DECEMBER 1983 – REVISED MARCH 1988

- **Package Options Include Plastic "Small Outline" Packages, Ceramic Chip Carriers and Flat Packages, and Plastic and Ceramic DIPs**
- **Dependable Texas Instruments Quality and Reliability**

description

These devices contain two independent D-type positive-edge-triggered flip-flops. A low level at the preset or clear inputs sets or resets the outputs regardless of the levels of the other inputs. When preset and clear are inactive (high), data at the D input meeting the setup time requirements are transferred to the outputs on the positive-going edge of the clock pulse. Clock triggering occurs at a voltage level and is not directly related to the rise time of the clock pulse. Following the hold time interval, data at the D input may be changed without affecting the levels at the outputs.

The SN54' family is characterized for operation over the full military temperature range of −55°C to 125°C. The SN74' family is characterized for operation from 0°C to 70°C.

FUNCTION TABLE

INPUTS				OUTPUTS	
$\overline{PRE}$	$\overline{CLR}$	CLK	D	Q	$\overline{Q}$
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H†	H†
H	H	↑	H	H	L
H	H	↑	L	L	H
H	H	L	X	Q_0	$\overline{Q}_0$

† The output levels in this configuration are not guaranteed to meet the minimum levels in V_{OH} if the lows at preset and clear are near V_{IL} maximum. Furthermore, this configuration is nonstable; that is, it will not persist when either preset or clear returns to its inactive (high) level.

SN5474 . . . J PACKAGE
SN54LS74A, SN54S74 . . . J OR W PACKAGE
SN7474 . . . N PACKAGE
SN74LS74A, SN74S74 . . . D OR N PACKAGE
(TOP VIEW)

SN5474 . . . W PACKAGE
(TOP VIEW)

SN54LS74A, SN54S74 . . . FK PACKAGE
(TOP VIEW)

NC - No internal connection

Figure 2.1 Excerpt from 7474 Device Data Sheet

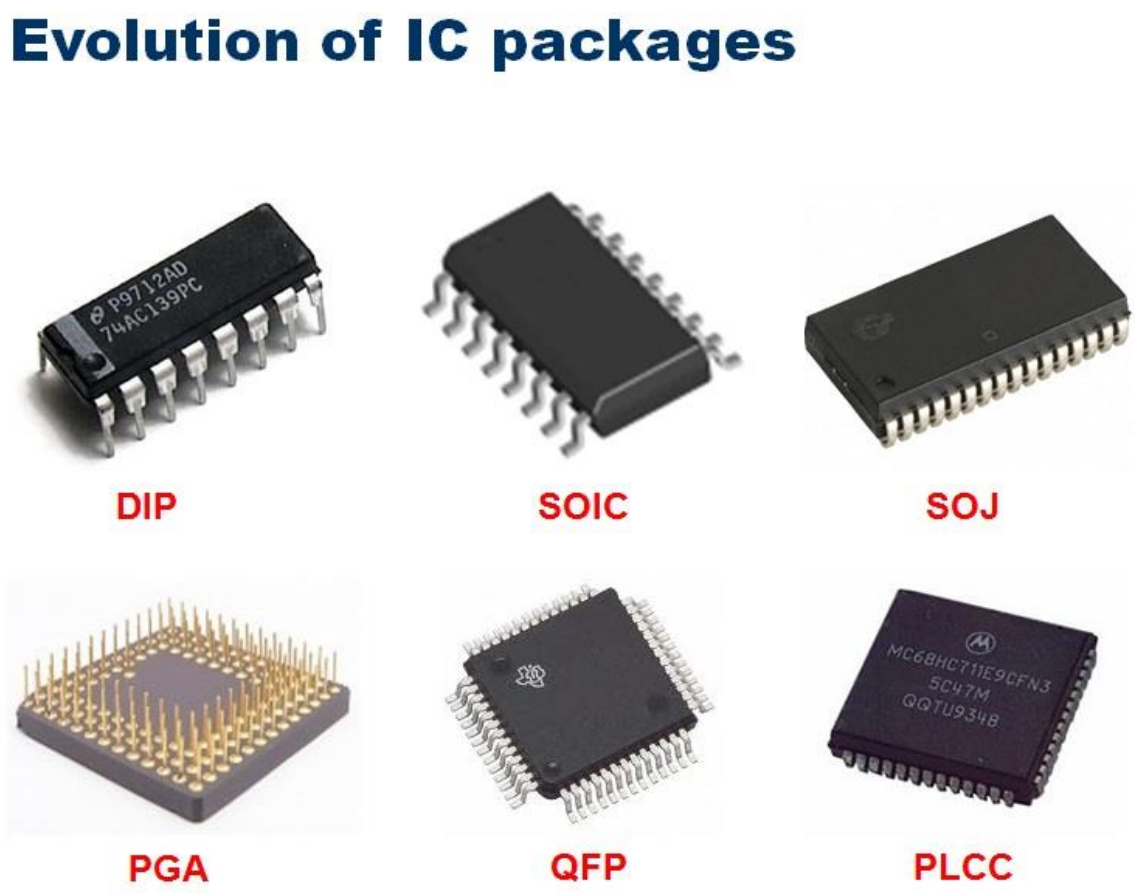


Figure 2.2Evolution of Packages

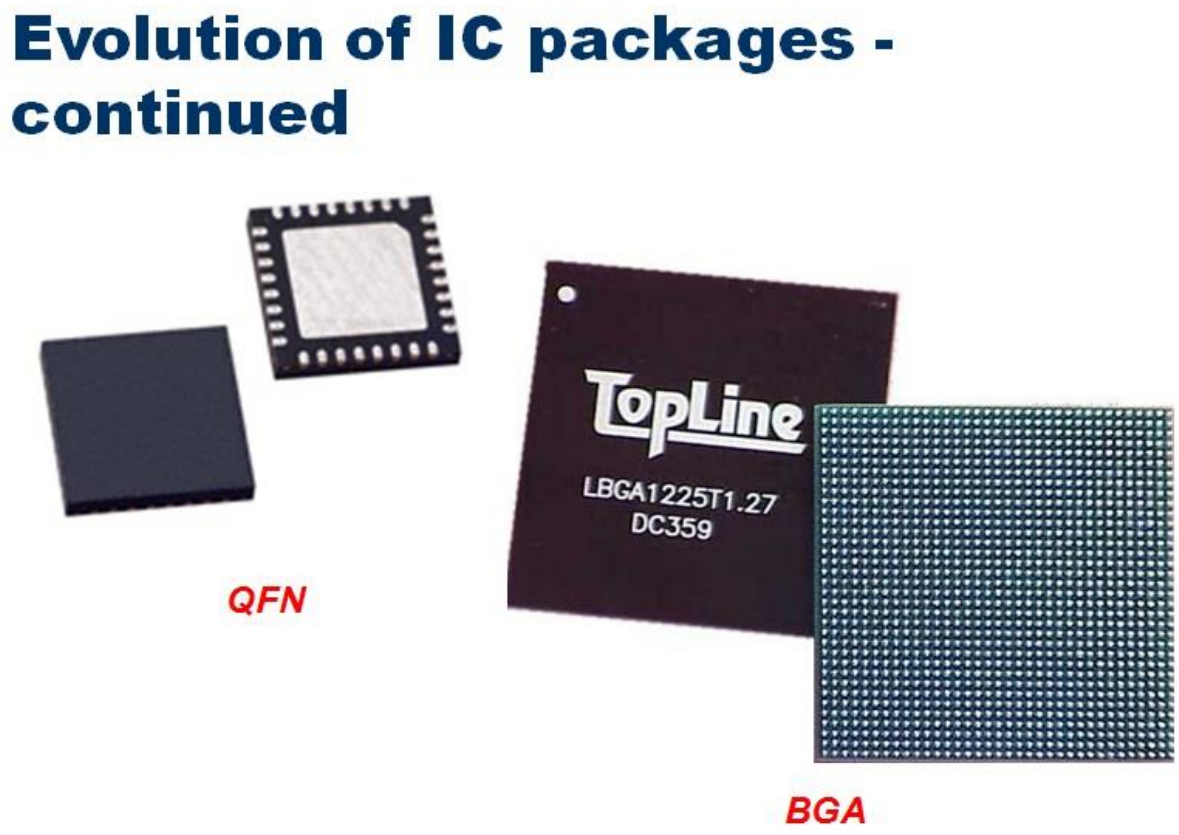


Figure 2.3 Evolution of Packages

The package details are as follows:

Acronym	Expansion
DIP	Dual in-line Package
SOIC	Small Outline Integrated Circuit
SOJ	Small-outline J-leaded package
PGA	Pin grid array
QFP	Quad Flat Package
PLCC	Plastic Leaded Chip Carrier
QFN	Quad Flat No-leads package
BGA	Ball Grid Array

Table 1.1 The package details

Interfaces

Figure 2.4 Clips Interface for ICFT.

Figure 2.5 Clips Interface for ICFT.

2.1.2 Models

Few popular Simulator Examples:

VHDL is an acronym, which stands for VHSIC Hardware Description Language. VHSIC is yet another acronym which stands for Very High Speed Integrated Circuits, VHDL language, which is fast becoming the worldwide standard HDL for digital design & testing.

VHDL describes the behavior, function, inputs, and outputs of a digital circuit design with the syntax to define modern programming languages.

The acronym does have a purpose, though; it is supposed to capture the entire theme of the language, which is to describe hardware much the same way we use schematics.

It is being used for documentation, verification, and synthesis of large digital designs. In addition to being used for each of these purposes, VHDL can be used to take three different approaches to describe hardware. These three different approaches are the structural, data flow, and behavioral methods of hardware description. Most of the time a mixture of the three methods are employed.

VHDL has to date proven to be the most successful and widely used language, used to define models for digital/analog/mixed-signal systems.

Verilog is another hardware description language (HDL) used in the design of verification of digital/analog/mixed circuits.

SPICE - SPICE (Simulation Program with Integrated Circuit Emphasis)is a general-purpose, analog electronic circuit simulator. It is a program used in integrated circuit and board-level design to check the integrity of circuit designs and to predict circuit behavior.

Let us discuss VHDL models here in detail:

Show in the figure 2.6 (a) A sample VHDL model for 7474 D- Flip Flop &Figure 2.6 (b) A sample VHDL model for 7474 D- Flip Flop

- VHDL are text based models to describe a logic circuit.
- Based on IEEE standards logic 1164
- Requires minimum 2 files. Design file contains the code for the device model. Test file contains the test bench i.e. the Test Vectors.

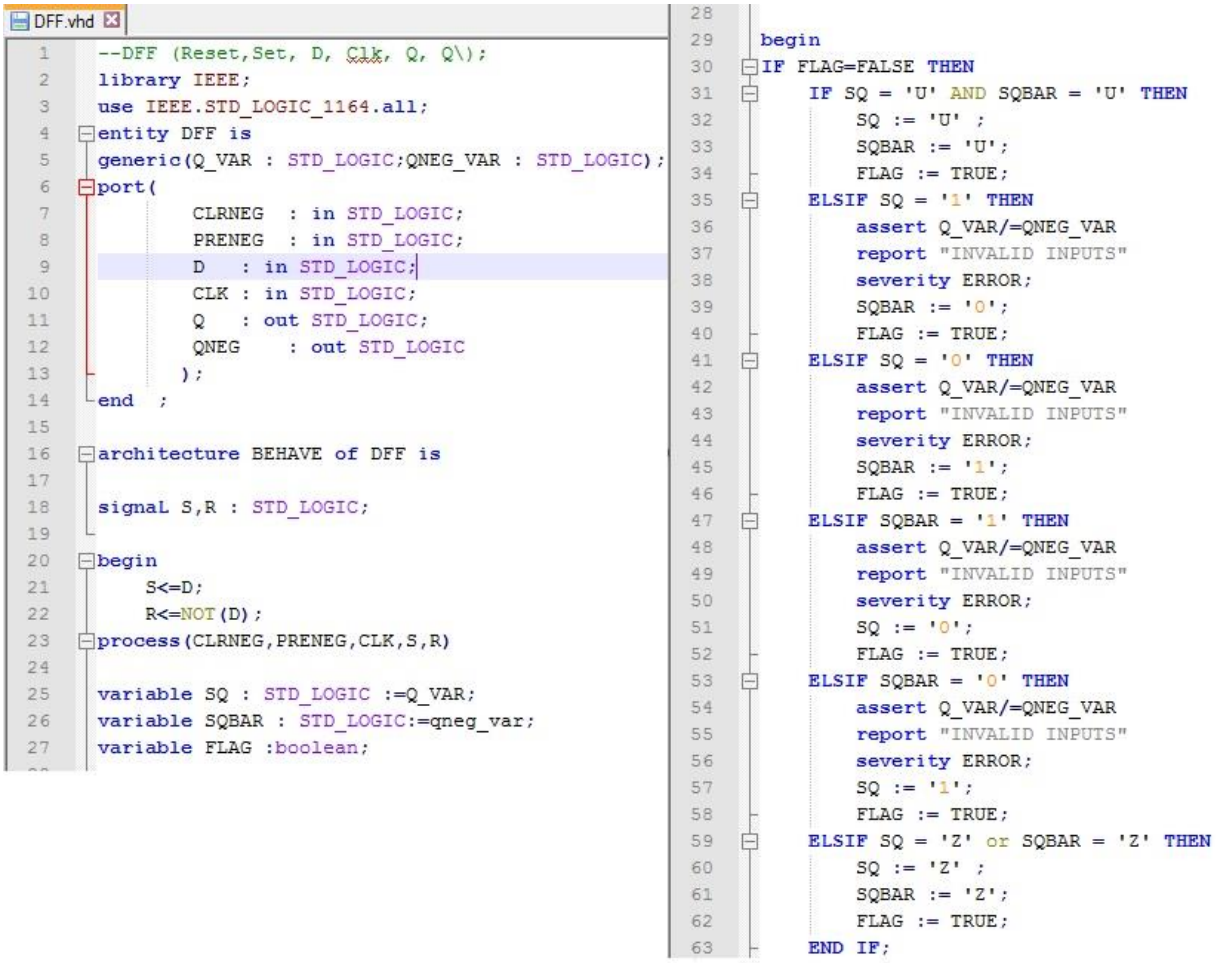


Figure 2.6 (a) A sample VHDL model for 7474 D- Flip Flop

```
ELSIF FLAG=TRUE THEN
    if TO_X01(PRENEG) = '0' and TO_X01(CLRNEG) = '1' then
        SQ := '1';
        SQBAR := '0';
    elsif TO_X01(PRENEG) = '1' and TO_X01(CLRNEG) = '0' then
        SQ := '0';
        SQBAR := '1';
    elsif TO_X01(PRENEG) = '1' and TO_X01(CLRNEG) = '1' then
        if CLK'event and TO_X01(CLK) = '1' then
            SQ := S OR (NOT(R) AND SQ);
            SQBAR := NOT(S OR (NOT(R) AND SQ));
        END IF;
    ELSIF TO_X01(PRENEG) ='0' AND TO_X01(CLRNEG) = '0' THEN
        SQ:='X';
        SQBAR:='X';
    end if;
END IF;
    Q <= SQ;
    QNEG <= SQBAR;
end process;
end ;
```

Figure 2.6 (b) A sample VHDL model for 7474 D- Flip Flop

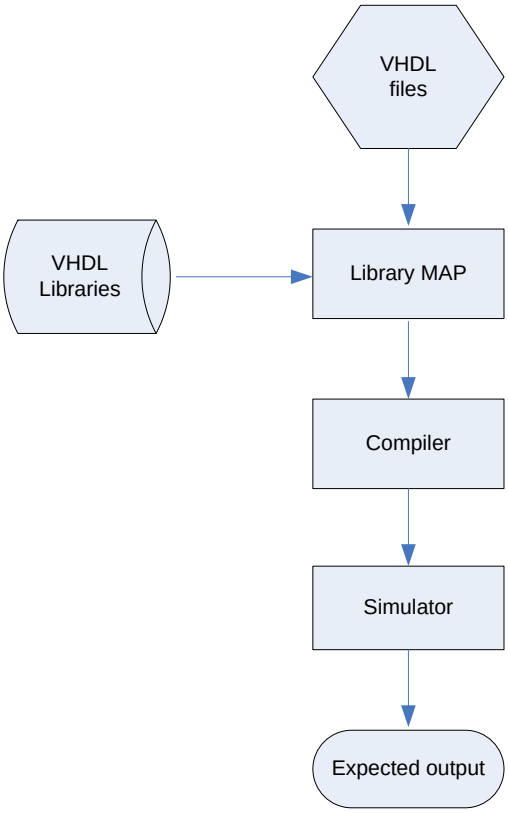


Figure 2.7 VHDL Simulation Flow Diagram

TRUTH TABLE

FUNCTION TABLE

INPUTS				OUTPUTS	
$\overline{PRE}$	$\overline{CLR}$	CLK	D	Q	$\overline{Q}$
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H↑	H↑
H	H	↑	H	H	L
H	H	↑	L	L	H
H	H	L	X	Q_0	$\overline{Q}_0$

Table 2.2 7474 Truth Table

For few more Truth Table examples please refer the end of this section.

2.1.3 TESTVECTOR

The drive pattern/test vector is the input/bi dir signal driven to the component. Generally this is arrived from the Truth table. Please note always the device should be initialized and brought to a known state before the device is actually tested.

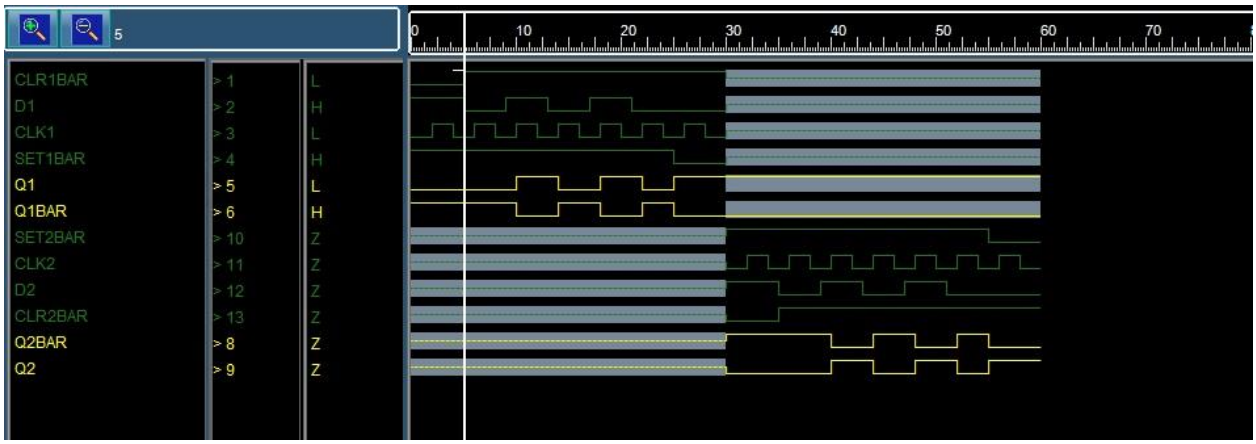


Figure 2.8 7474 Waveform window

The waveform window shows device pin signals in the form of logic level trains for the complete duration of the test cycle. On the left of the screen are shown three columns. The first column reads the pin labels; the second column reads the pin number. The third column reads the logic level on the trains, at the instantaneous position of the cursor. The cursor position is also dynamically shown in top of the cursor with the instant timing and pattern position on the scale**.** Show in the figure 2.8 7474 Waveform window.

The legends for the logic levels are as follows

Legend	Logic Level / Signal
L	Low
H	High
Z	High Impedance
X	Undefined

Table 2.3 The legends for the logic levels

Green Colour – Shows the data to be driven to input signals
Yellow Colour- Expected output from the VHDL simulator

Time Base

This defines the time base for ICFT drive speed. Each device has its drive speed defined in the datasheet. This will be defined in the device library. The device is tested using default drive speed.

1.10 Threshold

Each device has its own drive voltage and receive threshold level defined in the datasheet. This will be defined in the device library. The device is tested using default threshold defined in the library.

For e.g. consider Transistor Transistor Logic (TTL) Family.

Per definition, for the input/bi dir pins, 5.0 V can be driven for logic high, 0.0 V can be driven for logic low. The receive voltage for output/bi dir pins are anywhere between 2.0V to 5.0V is considered as logic high and receive voltage anywhere between 0.0V to 0.8 V is considered as logic low. Show the Figure 2.9 TTL Threshold.

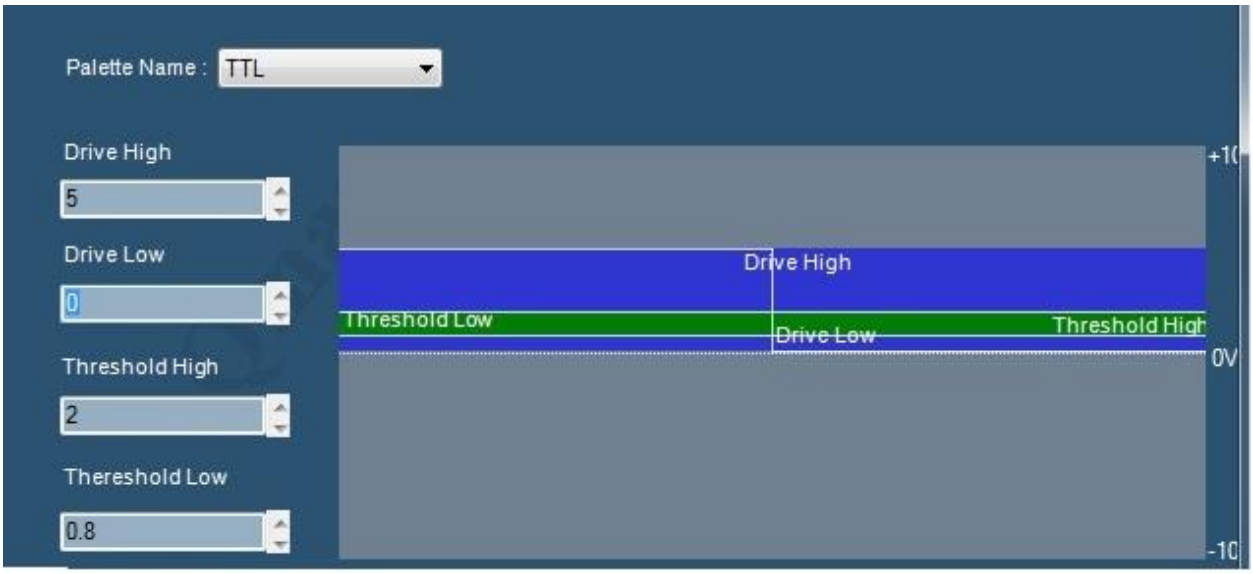


Figure 2.9 TTL Threshold

Palette	Drive High	Drive Low	Threshold High	Threshold Low
TTL - Transistor–Transistor Logic	5.0 V	0V	2.0V	0.8V
CMOS(5V Logic) complementary metal–oxide–semiconductor logic	5V	0V	3.5V	1.5V
CMOS(12V Logic) complementary metal–oxide–semiconductor logic	10V	-0.46V	7V	3V
EIA Electronic Industries Association	10V	-10V	6V	-6V
ECL Emitter-Coupled logic	0V	-2.4V	-1.1V	-1.45V
3.3VLogic	3.3V	0V	2.25V	0.6V
PECL Positive Emitter-Coupled logic	4.98V	0V	3.88V	3.57V
DTL Diode–Transistor Logic	10V	0V	6V	2V
TINY	5V	0V	4.5V	0.2V

Table 2.4 Palette Settings for various families

ICFT

Show in the figure 2.10 sample device 7474 Device Test Window.

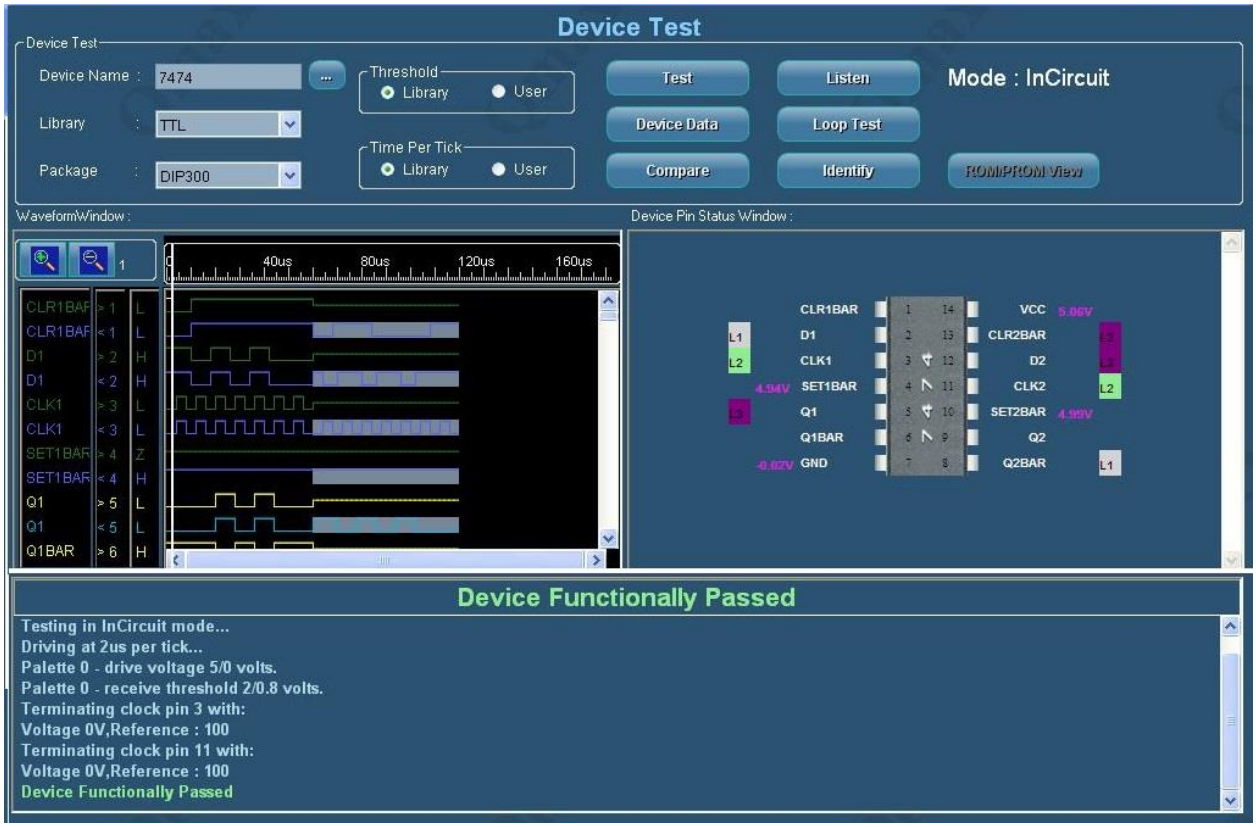


Figure 2.10 sample device 7474 Device Test Window

2.1.4 Special considerations:

1.12.1Back Drive Defense Time Limit:

As per the International Defense Standard (00-53/1), the maximum drive time is limited to 65 milliseconds. If the testing time exceeds this limit, this may cause damages to the other device pins connected to the Device under Test.

1.12.2 Pull up/pull down:

This facilitates in testing Open Collector OC / Open Emitter OE devices which uses in built analog module to pull up / pull down devices. It uses voltage and source impedance defined in library.

Auto compensation:

- Ability to test as wired condition with auto compensation applied on the fly
- Eliminates the need for creation or customization of the Test Program according to In-Circuit conditions.

  Let us discuss in details with an example: Figure 2.11 Auto compensation Techniques.

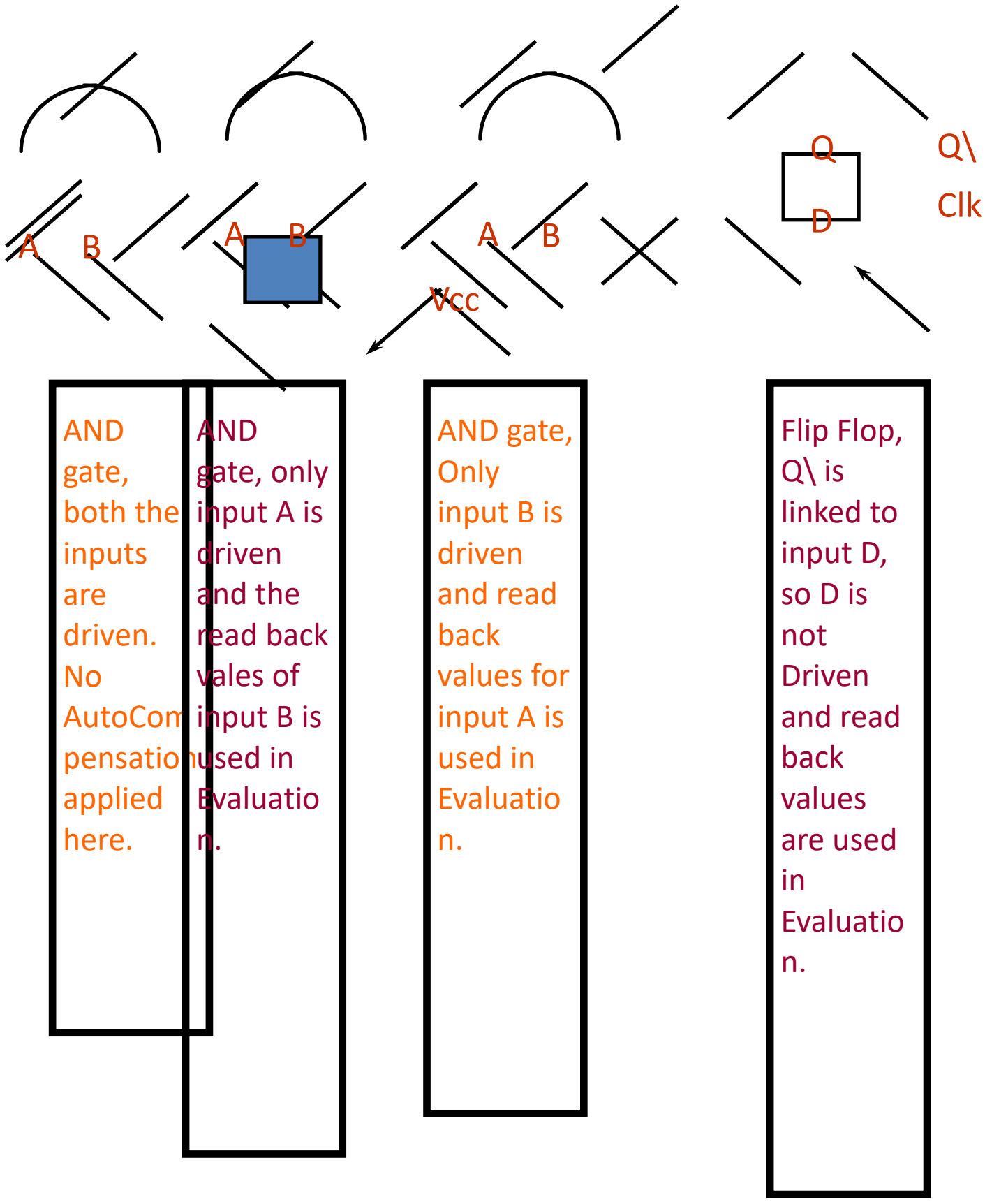


Figure 2.11 Auto compensation Techniques

Clock pin termination:

State machine devices such as flip flops / counters might often fail due to clock signal having under / over shoots and acting as another rising / falling edge.

If any state machine device fails, this condition may be retested with “Clock pin termination” by applying impedance with pull up/pull down voltages.

The values used are generally 50-100 ohms with pull up / down voltages of -1 to -2v or zero volts.

Guarding:

Guarding is a technique used to isolate the UUT from rest of the devices in a board.

Proprietary Devices

Proprietary devices such as are ROMS / EPROMS and EEPROMS are learnt/verified for it’s data rather. Data can be learnt from the known good board (KGB) and verified against faulty board.

Show in the Figure 2.12 Proprietary Data.

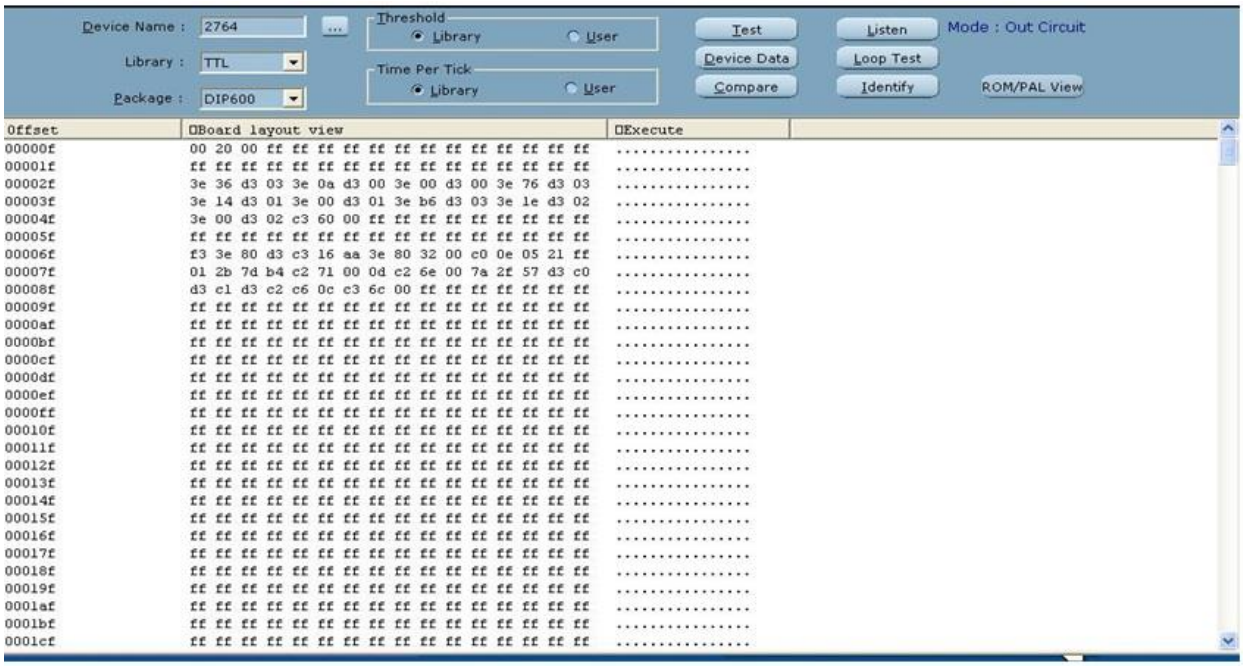


Figure 2.12 Proprietary Data

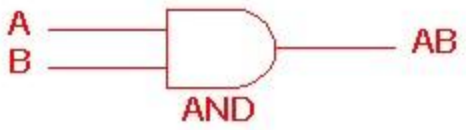
ICFT Vs OCFT

ICFT	OCFT
Can have Links between the IC pins.	Links not permitted
Can have pins struck to Gnd/VCC based on design	Except power pins, other pins cannot be strucked to VCC/GND

Pull up/pull down optional based on board design	Pull up/pull down mandatory
Force driving required as interconnect between the Ics will affect the testing	Force driving not required
Guarding may be required to avoid BUS contention created by other devices.	Guarding not required.
Auto compensation required	auto compensation not required.

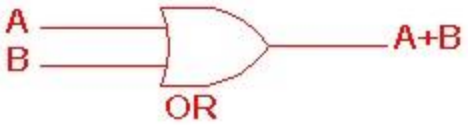
2.5 Truth Tables – Examples

AND gate



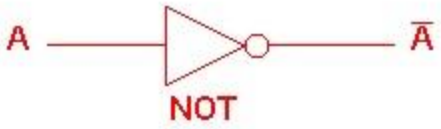
2 Input AND gate		
A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

OR gate



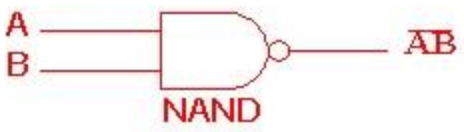
2 Input OR gate		
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

NOT gate



NOT gate	
A	$\overline{A}$
0	1
1	0

NAND gate



2 Input NAND gate		
A	B	$\overline{A.B}$
0	0	1
0	1	1
1	0	1
1	1	0

Figure 2.13 Truth Table for Gates

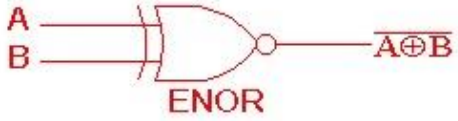
NOR gate

2 Input NOR gate		
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

EXOR gate

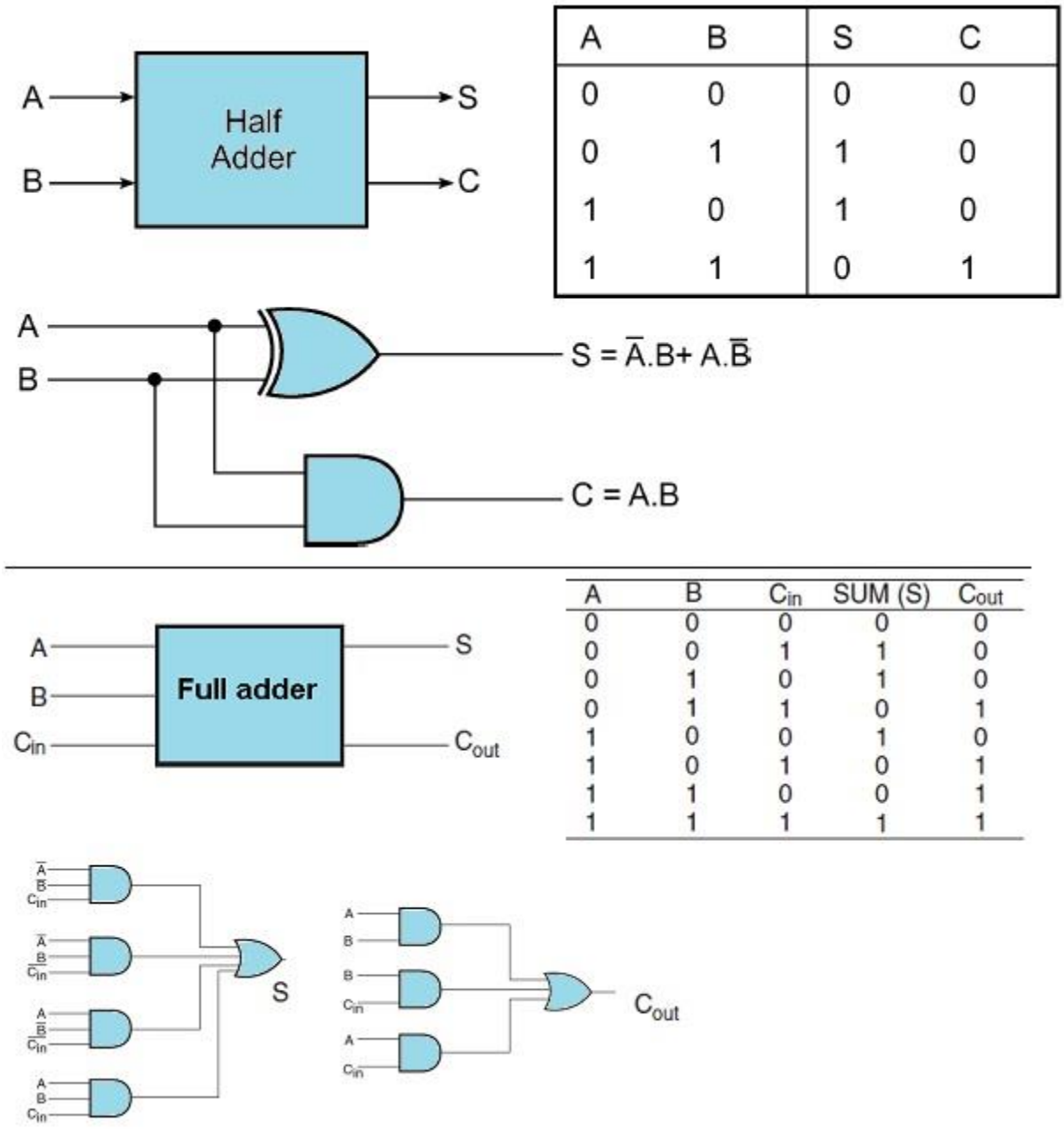
2 Input EXOR gate		
A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

EXNOR gate



2 Input EXNOR gate		
A	B	$\overline{A⊕B}$
0	0	1
0	1	0
1	0	0
1	1	1

Figure 2.14 Truth Table for Gates



A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

A	B	C_{in}	SUM (S)	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Figure 2.15 Truth Table for Adders

Section 2: Board Level Testing

2.1 Board Testing

Generally a board is tested using Edge Connectors and/or Bed of Nails for it's functionality. These Edge connectors are used to drive the test pattern and to receive the response patterns. The actual response pattern is compared against the expected response and the board is declared as pass or fail. The expected response can be derived either from simulation or by learning the signatures from the good board. If the board functionally passes there is no need of further testing. If it fails, further diagnosing testing should be carried out to identify the fault at the node/component level.

The functionality testing is conducted by,

- Powering up the board
- Application of appropriate input signals based on the components presence and it's interconnectivity.
- Measurement of the output/bi-dir signals from the board.
- The interconnects between the board components are also tested.
- The board integrity will be checked fully and PCB track opens / shorts can be covered in the test.
- This process requires an Automated Test Equipment (ATE), as the number of signal points to be checked at any instant is beyond the manual modes of testing.

2.2 GO/NOGO Test Process Flow

The following is the input requirements to develop a Test Program Set (TPS)

1) Bill of Materials (BOM).
2) Schematic diagram
3) Net List
4) Primary I/O
5) Fixture definition
6) Cluster Definition
7) Models(e.g. VHDL/very log)
8) Test Vectors.
9) Simulated output/Learnt Data.

We will have a detailed look into each input requirements.

2.2.1 Bill of Materials(BOM)

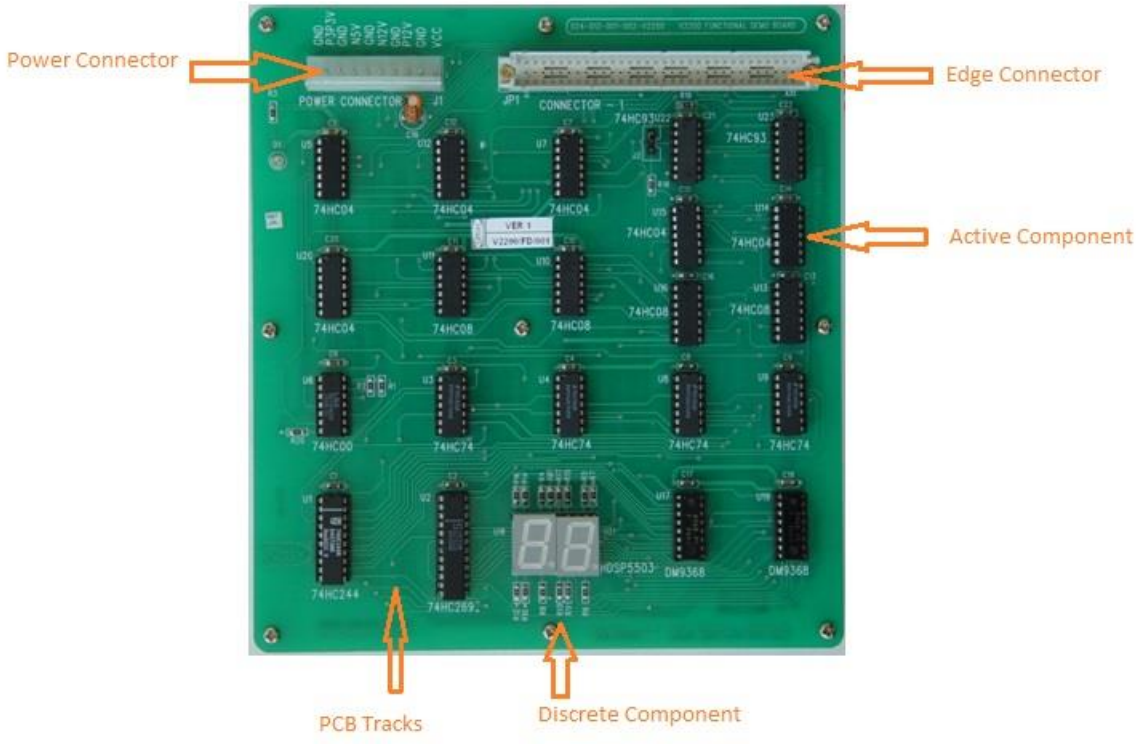


Figure 2.16 Sample Printed Circuit Board

Show in the figure 2.16 Sample Printed Circuit Board .The BOM represents the list of components used on printed circuit board (PCB). BOM is used as a database to identify the parts in the PCB. This includes the active, passive, connectors, custom and discrete components List.

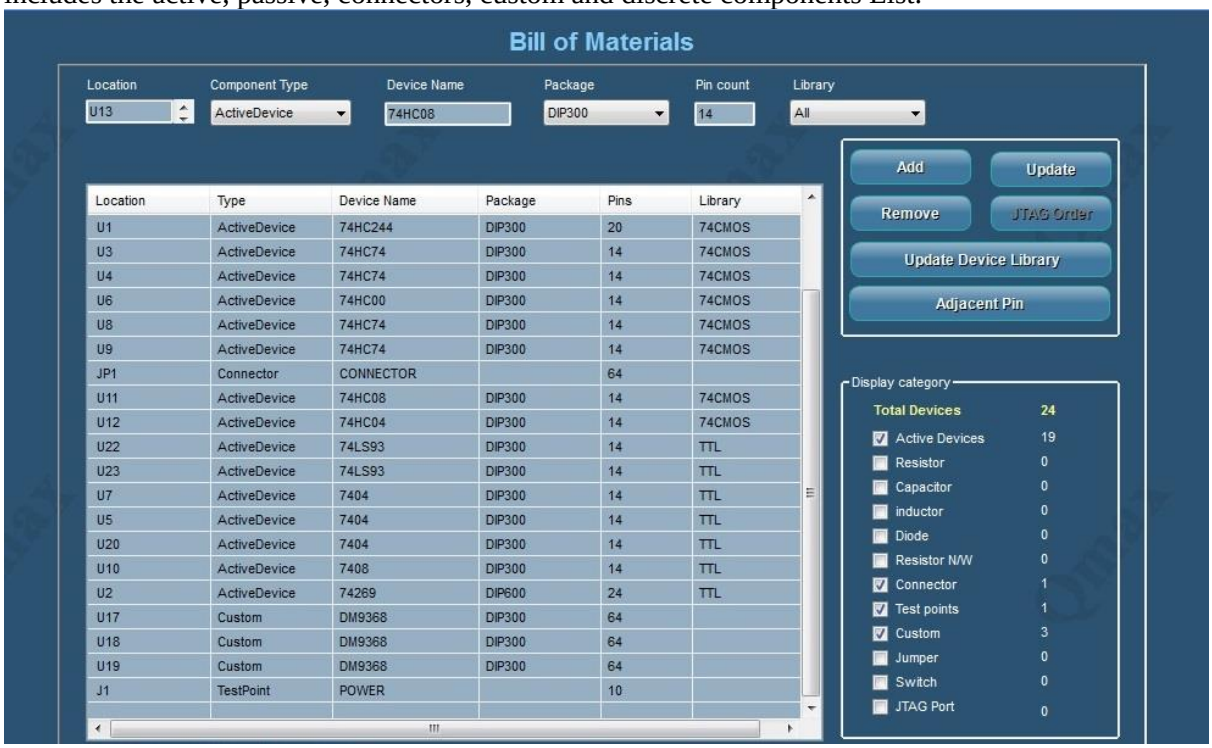


Figure 2.17 An example of a BOM for a PCB.

Show in the figure 2.17 is An example of a BOM for a PCB**.**In location U1, the device 74HC244 is placed in the PCB. The device location is very important since multiple devices of same type, package can be present at multiple locations. It has 20 Pins and the package type is Dual in Line Package DIP300. Please note for a selected device, the number of pins may differ based on the Package selected. The device 74HC244 belongs to CMOS library. Likewise, all the devices on the board should be entered as part of BOM.

2.2.2 Schematic diagram

A schematic diagram also called as circuit diagram shows the components and interconnections of the circuit using standardized symbolic representations. Schematic helps to generate the net list which in turn helps to generate the Test Program Set(TPS).

Why Schematic diagram is important?

a) Provides inputs to create the BOM database.
b) Describes the interconnect between the components present in the board

c) Helps the TPS developer to create the NETLIST

d) Provides information about the power supply details.

e) Provides information the TPS developer to understand the functionality of the board.

f) Assists the TPS developer in creating the Test Vector.

g) Helps the TPS developer to create the fixture and cluster.

h) Helps the TPS developer to create the Go/NOGO Test & Diagnostic (Back tracking Test).

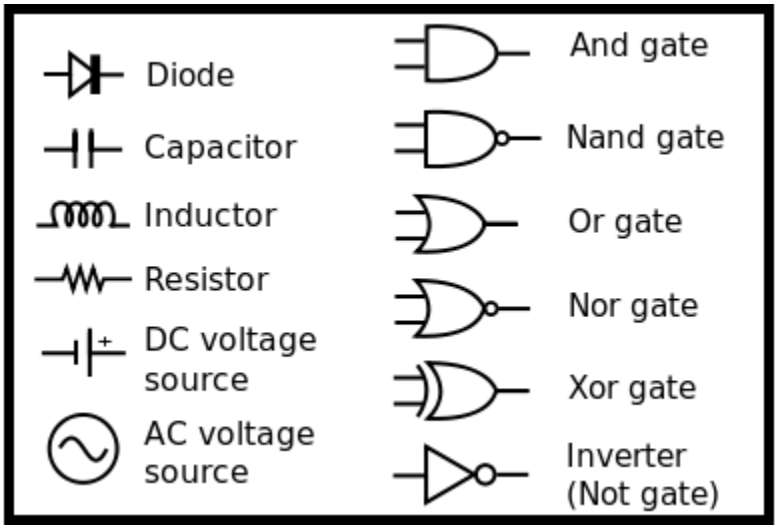


Figure 2.18 Common Schematic Diagram symbols

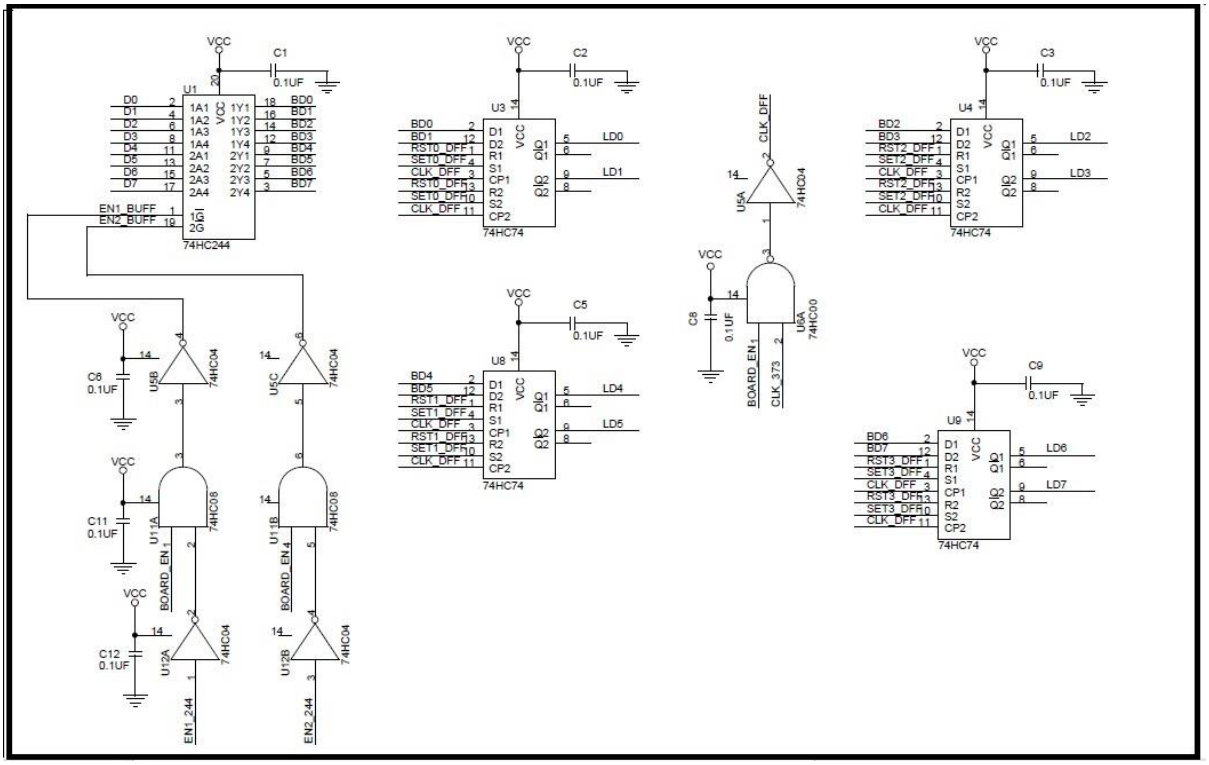


Figure 2.19 A Schematic part 1

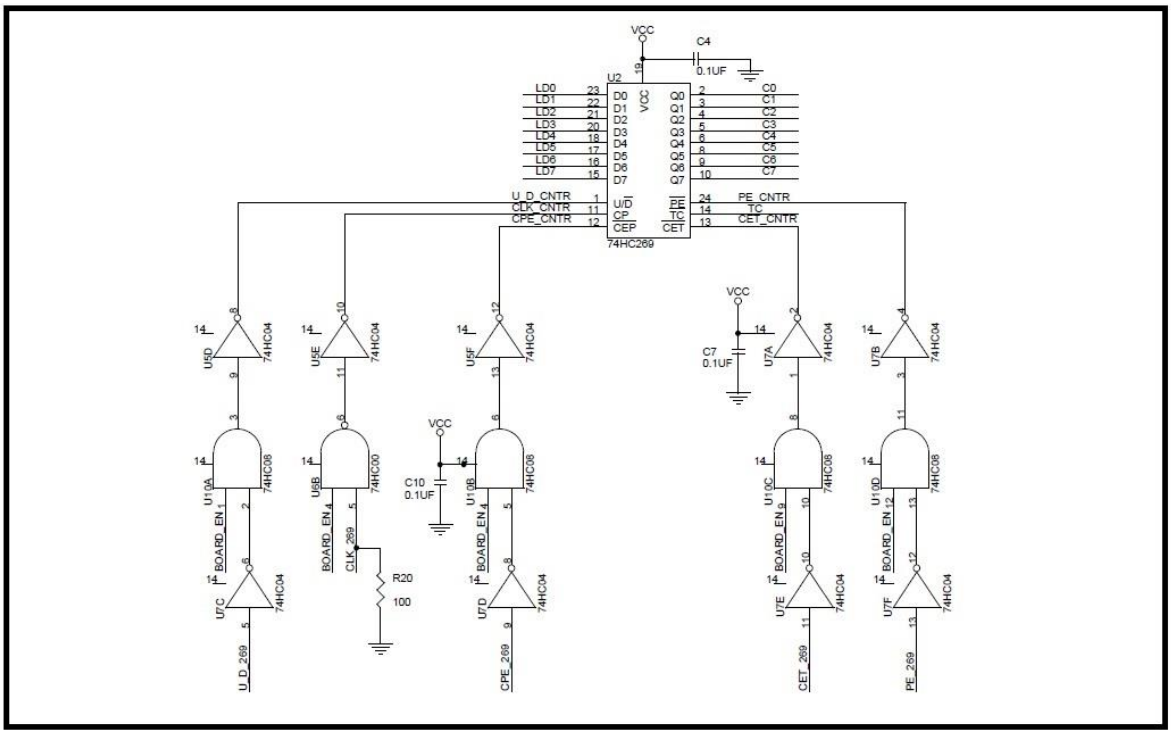


Figure 2.20 A Schematic part 2

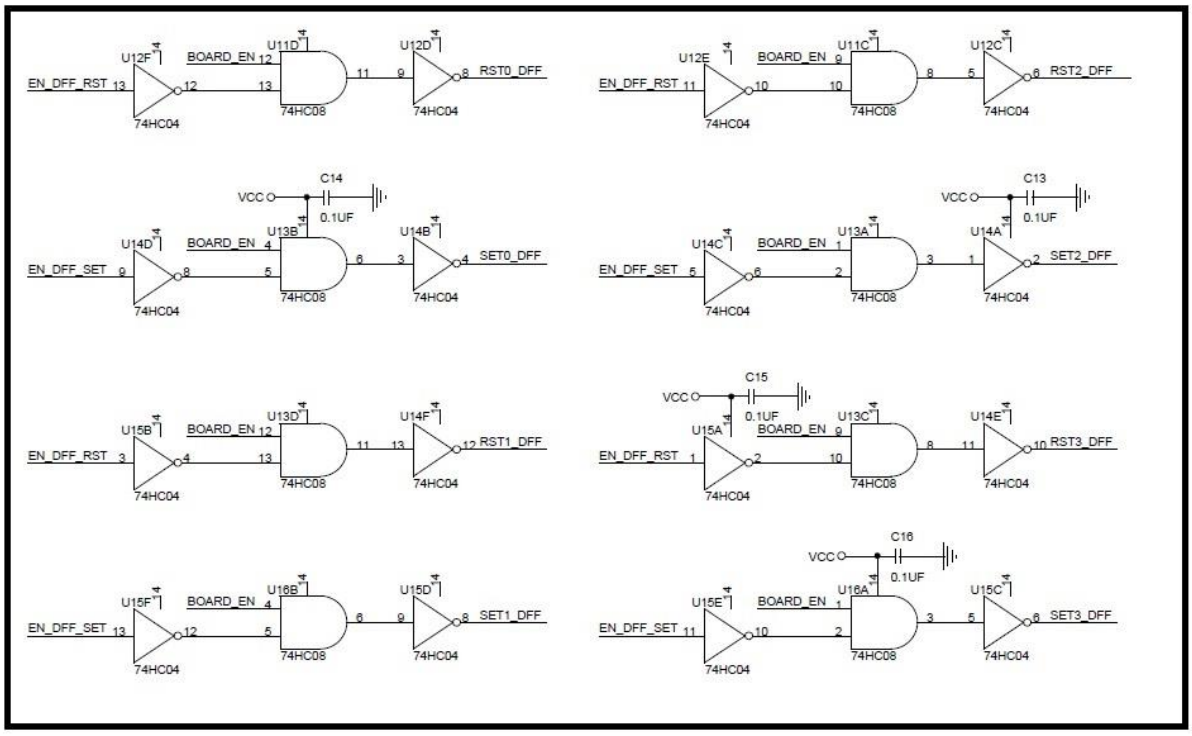


Figure 2.21 A Schematic part 3

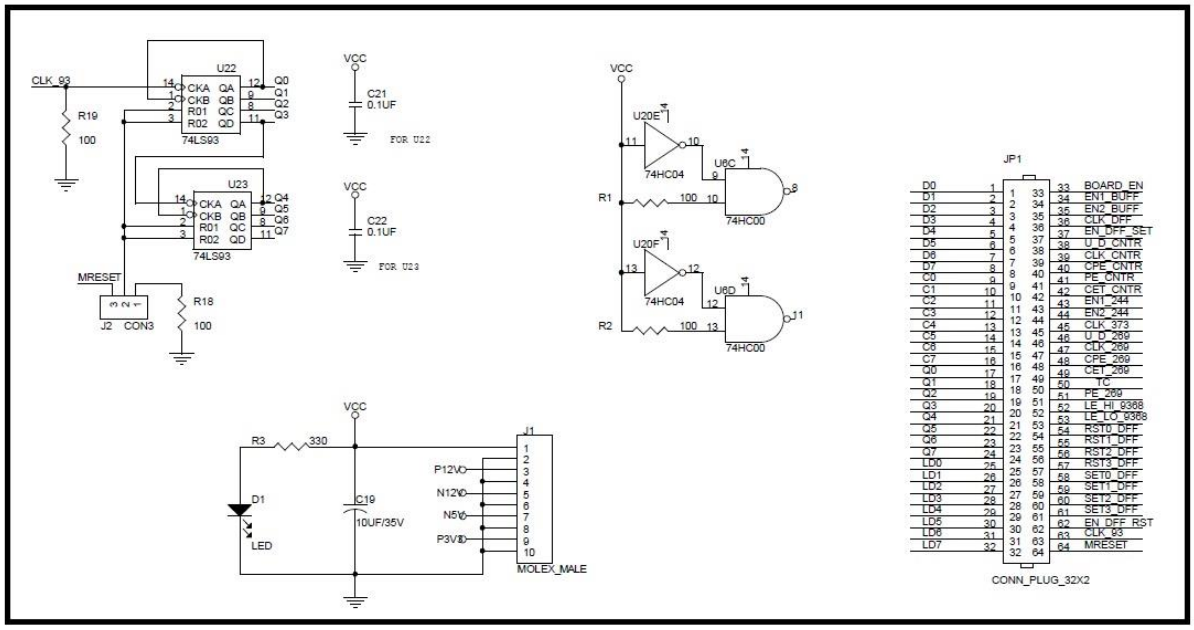


Figure 2.22 A Schematic part 4

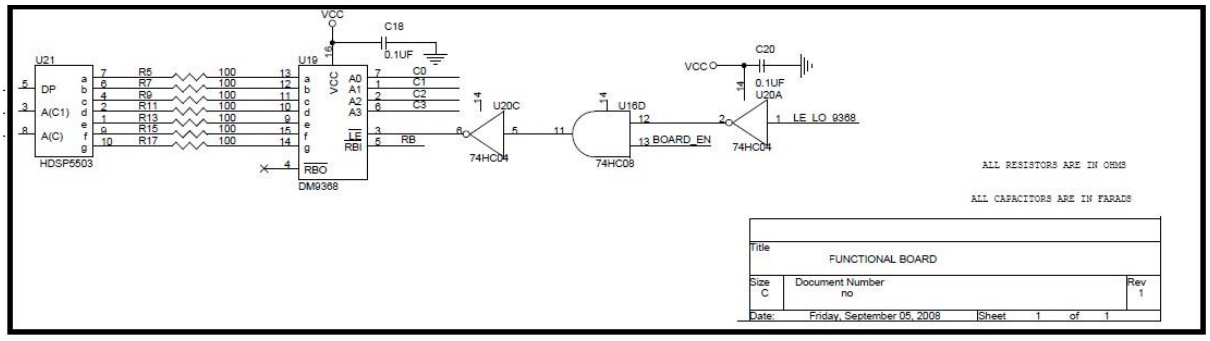


Figure 2.23 A Schematic part 5

2.2.3 NET LIST

A Net list is a description of the connectivity of an electronic circuit. Net list gives the interconnection details and grouping of the devices. This information is critical in Board diagnostic tests. A Net can have one or more nodes connected at the same terminal. A net type may be any one of the following

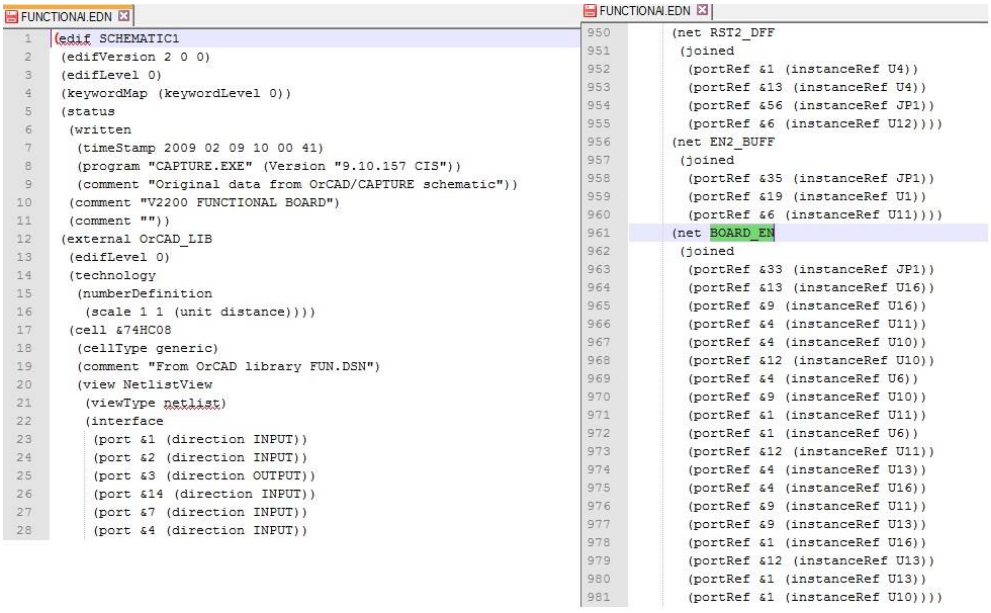
a) Normal Net (Signal NET)
b) Power Net High – Nodes that are connected to VCC including device power pins
c) Power Net Low - Nodes that are connected to GND including device power pins
d) CLOCK

In PCB repair, the Automatic Test Systems generally provides two methods to create the NETLIST. Show in the Figure 2.24 – Sample EDN Format (.edn) Figure 2.24 – Sample wire list Format (.wrl) Figure 2.25 – Sample NETLIST (.NET)

a) Import of NETLIST

The user can import the NETLIST which are created by Computer Aided Design CAD/ Computer Aided Engineering CAE tools.

The Net list files are used in Electronic Data Interchange Format (EDIF). The .EDN Electronic Design Interchange format,.NET, WIRELIST .wrl formats are few commonly available schematic files. This can be imported directly to get the NET List.



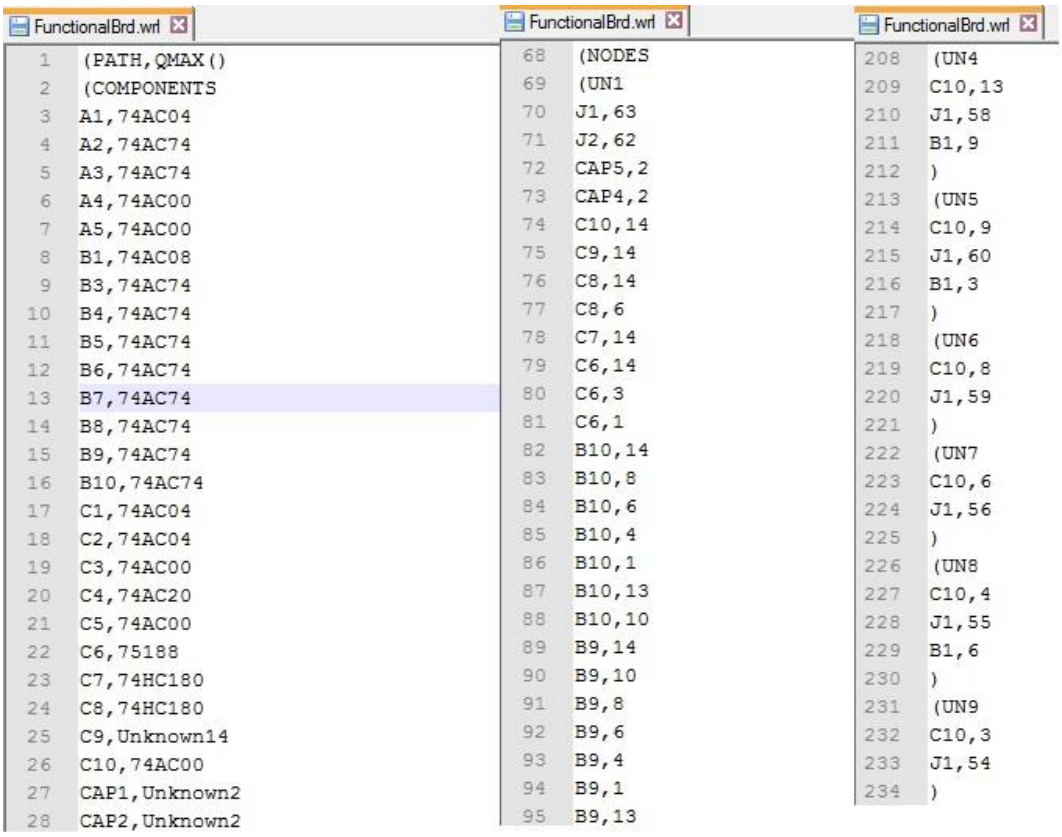
FUNCTIONAL.EDN

```
(edif SCHEMATIC1
 (edifVersion 2 0 0)
 (edifLevel 0)
 (keywordMap (keywordLevel 0))
 (status
  (written
   (timeStamp 2009 02 09 10 00 41)
   (program "CAPTURE.EXE" (Version "9.10.157 CIS"))
   (comment "Original data from OrCAD/CAPTURE schematic"))
  (comment "V2200 FUNCTIONAL BOARD")
  (comment ""))
 (external OrCAD_LIB
  (edifLevel 0)
  (technology
   (numberDefinition
    (scale 1 1 (unit distance))))
  (cell &74HC08
   (cellType generic)
   (comment "From OrCAD library FUN.DSN")
   (view NetlistView
    (viewType netlist)
    (interface
     (port &1 (direction INPUT))
     (port &2 (direction INPUT))
     (port &3 (direction OUTPUT))
     (port &14 (direction INPUT))
     (port &7 (direction INPUT))
     (port &4 (direction INPUT))
```

FUNCTIONAL.EDN

```
  (net RST2_DFF
   (joined
    (portRef &1 (instanceRef U4))
    (portRef &13 (instanceRef U4))
    (portRef &56 (instanceRef JP1))
    (portRef &6 (instanceRef U12))))
  (net EN2_BUFF
   (joined
    (portRef &35 (instanceRef JP1))
    (portRef &19 (instanceRef U1))
    (portRef &6 (instanceRef U11))))
  (net BOARD_EN
   (joined
    (portRef &33 (instanceRef JP1))
    (portRef &13 (instanceRef U16))
    (portRef &9 (instanceRef U16))
    (portRef &4 (instanceRef U11))
    (portRef &4 (instanceRef U10))
    (portRef &12 (instanceRef U10))
    (portRef &4 (instanceRef U6))
    (portRef &9 (instanceRef U10))
    (portRef &1 (instanceRef U11))
    (portRef &1 (instanceRef U6))
    (portRef &12 (instanceRef U11))
    (portRef &4 (instanceRef U13))
    (portRef &4 (instanceRef U16))
    (portRef &9 (instanceRef U11))
    (portRef &9 (instanceRef U13))
    (portRef &1 (instanceRef U16))
    (portRef &12 (instanceRef U13))
    (portRef &1 (instanceRef U13))
    (portRef &1 (instanceRef U10))))
```

Figure 2.24 – Sample EDN Format (.edn)



FunctionalBrd.wrl

```
(PATH,QMAX()
(COMPONENTS
A1,74AC04
A2,74AC74
A3,74AC74
A4,74AC00
A5,74AC00
B1,74AC08
B3,74AC74
B4,74AC74
B5,74AC74
B6,74AC74
B7,74AC74
B8,74AC74
B9,74AC74
B10,74AC74
C1,74AC04
C2,74AC04
C3,74AC00
C4,74AC20
C5,74AC00
C6,75188
C7,74HC180
C8,74HC180
C9,Unknown14
C10,74AC00
CAP1,Unknown2
CAP2,Unknown2
```

FunctionalBrd.wrl

```
(NODES
(UN1
J1,63
J2,62
CAP5,2
CAP4,2
C10,14
C9,14
C8,14
C8,6
C7,14
C6,14
C6,3
C6,1
B10,14
B10,8
B10,6
B10,4
B10,1
B10,13
B10,10
B9,14
B9,10
B9,8
B9,6
B9,4
B9,1
B9,13
```

FunctionalBrd.wrl

```
(UN4
C10,13
J1,58
B1,9
)
(UN5
C10,9
J1,60
B1,3
)
(UN6
C10,8
J1,59
)
(UN7
C10,6
J1,56
)
(UN8
C10,4
J1,55
B1,6
)
(UN9
C10,3
J1,54
)
```

Figure 2.24 (b) – Sample wire list Format (.wrl)

CADNETICS.NET

```
PARTS LIST

100nF/50V-0603    100nF/50V-0603  C1
10nF/35V 0805     10nF/35V 0805   C10
10uF/35V 0805     10uF/35V 0805   C11
100nF/50V 0805    100nF/50V 0805  C2
100nF/50V 0805    100nF/50V 0805  C3
10uF/16V/Low ESR  10uF/16V/Low ES C4
100nF/50V 0805    100nF/50V 0805  C5
10uF/35V 0805     10uF/35V 0805   C6
100uF/50V 0805    100uF/50V 0805  C7
100nF/50V 0805    100nF/50V 0805  C8
100nF/50V 0805    100nF/50V 0805  C9
LED-0805,GREEN    LED-0805,GREEN  D1
LED-0805,GREEN    LED-0805,GREEN  D2
DIODE SCHOTTKY-08 DIODE SCHOTTKY- D3
MSS1P6-M3--0805   MSS1P6-M3--0805 D4
MSSIP6-M30805     MSSIP6-M30805   D5
JST_B4B-PH-K-S    RMC4PINMALESTRA JP1
JST_B2B-PH-K      RMC2PINMALESTRA JP10
JST_B2B-PH-K      RMC2PINMALESTRA JP11
JST_B4B-PH-K-S    RMC4PINMALESTRA JP2
JST_B6B-PH-K      RMC6PINMALESTRA JP3
JST_B2B-PH-K      RMC2PINMALESTRA JP4
JST_B2B-PH-K      RMC2PINMALESTRA JP5
JST_B2B-PH-K      RMC2PINMALESTRA JP6
JST_B2B-PH-K      RMC2PINMALESTRA JP7
JST_B2B-PH-K      RMC2PINMALESTRA JP8
```

CADNETICS.NET

```
NET LIST

NODENAME +12V              $
    D4          1  C2          1  JP10        2  JP9         2  $
    JP8         2  JP11        2  U4          3  R17         2  $
    L2          2  C9          1  C10         1  C11         1
NODENAME +24V              $
    JP4         2  JP1         4  JP1         3  JP2         4  $
    JP2         3  D3          1  JP5         2  JP6         2  $
    JP7         2
NODENAME +5V               $
    C1          2  SW1         8  SW1         7  SW1         6  $
    SW1         5  U3         17  U3         20  R5          2  $
    U4          4  U4          2  C4          1  R15         2  $
    C3          1  R14         2
NODENAME GND               $
    C1          1  JP1         1  JP1         2  JP2         1  $
    JP2         2  U1          1  U1          3  U5          2  $
    C2          2  D1          K  JP3         1  JP3         4  $
    U2          1  U2          3  JP10        1  JP9         1  $
    JP8         1  JP11        1  U3         16  U3         10  $
    U3         15  R9          1  R10         1  R11         1  $
    R12         1  R13         1  C4          2  R16         1  $
    C3          2  D2          K  C5          2  C6          2  $
    C7          2  D5          1  R18         1  C9          2  $
    C10         2  C11         2
```

Figure 2.25 – Sample NETLIST (.NET)

b) Manual Entry

Net list of the Board and the interconnectivity of components can be entered manually, if there is no Net list available.Net list entry for the all the components covered in the BOM list helps to track the full coverage of components under test.

General Guidelines:

Filter Nodes by components or connectors (this is for easy identification of nets) and the nets can be entered by selecting the components pins and the other pins connected one by one.

Default status of the nodes can be marked as Logic High or Low, unread etc as the case may be by pressing the respective node type button. This marking helps in diagnostic testing, where backtracking guides you to the faulty node.
The user, for reference, can give net names.

Power Nets can be declared separately so as to identify them separately. The Net list including Power Pins is also entered to identify the Power Nets.

Mark as Logic H and mark as Logic L denote marking nodes used in Pull up or Pull down circuits.

When testing, these marked pins are to be kept in H or L states. Hence this marking guides the programmer.

Mark as clock, non-tri stable, unused are useful during Boundary scan device testing when these marked device pins are not to be included in Boundary scan test.

Always take a note of hanging nets since this may have only single nodes& Unused NETS as this may affect the test coverage.

The NETLIST will be used in the GO/NOGO Test as well as in the guided probe back tracking technique to display the faulty area and components. Hence care must be given to give the correct name and numbers so as to avoid confusion in the later stages. Figure 2.16 Sample Net List details shown below.

Figure 2.16 Sample Net List details

Sample Nets & Nodes shown below.

Net Name	Nodes
BOARD_EN	U13_1,U13_4,U13_9,U13_12,U16_1,U16_4,U16_9,U16_13,U6_1,U6_4,JP1_33,U11_1,U11_4, U11_9,U11_12
N266975	U13_2,U14_6

Table 2.15 Sample Net List details

2.2.4 Primary I/O

Identification of primary I/O plays an important role in the board testing. Since the primary I/O signals are connected to Edges, they are used to do perform Go/No Go testing. A Go/No Go testing basically tests the board for its functionality and determines whether the board is good or faulty.
If the Go/No Go test passes there is no need to test the board functionally further. If it fails, then diagnostics test i.e. backtracking should be performed to identify the fault at the node level. Shown below the figure 2.17 – Card Edge Interface
The primary I/O pin can be any of the following:

a) Input
b) Output
c) BIDIR
d) ANALOG

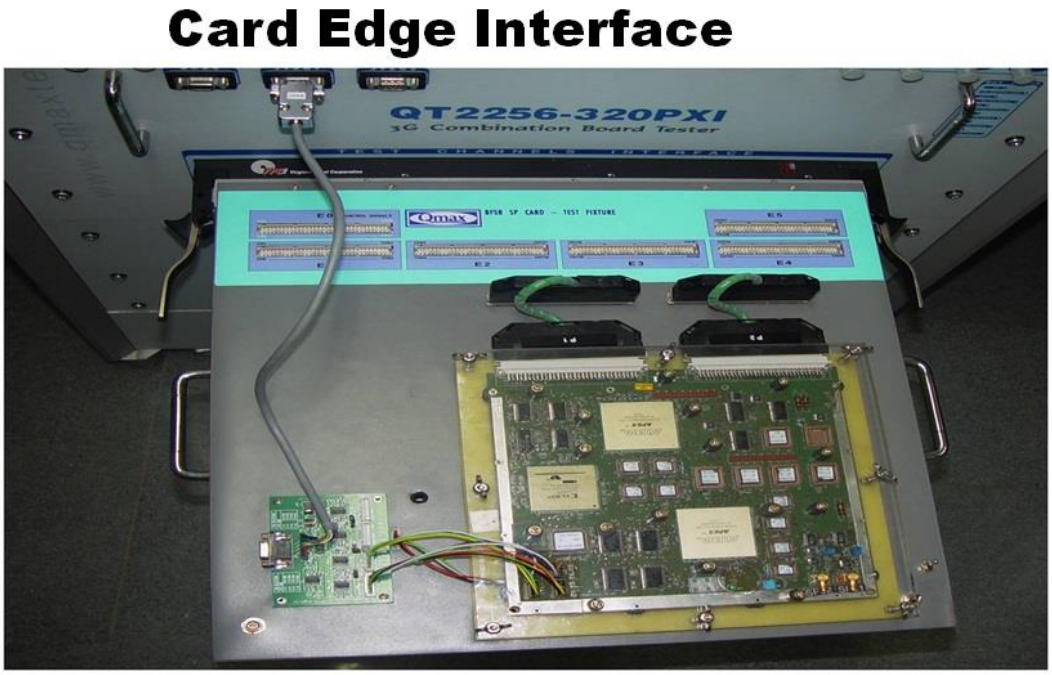


Figure 2.17 – Card Edge Interface

During the programming there are circumstances the user has to use some channels for monitoring some outputs and inputs in the PCB under test.

These nodes can be programmed as additional Test Points.

These test points will not be covered during diagnosis of a fault in the PCB and these are used only for monitoring purposes in Learn and Compare.

2.2.5 Fixture Definition

Fixture is nothing but the channel mapping between the Automated Test Equipment channels and the Edges/Bed of Nails of the board under test. Test vector is driven through this mapped channels to the Board Under Test and response from the board is received through the ATE channels. Figure 2.18 – Fixture shown below.

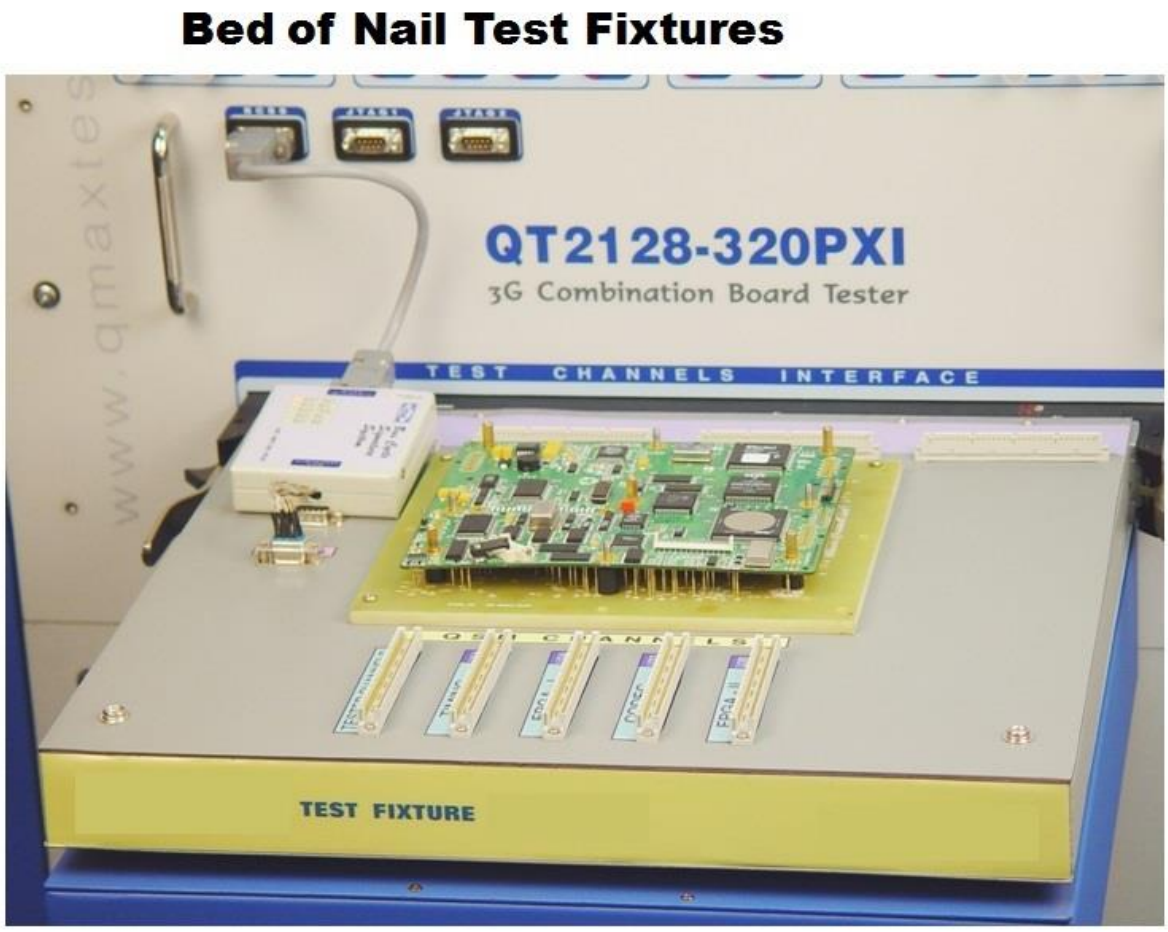


Figure 2.18 – Fixture

2.2.6 Cluster Definition

Cluster is a logical grouping of devices sharing the functionality among them. A board can have multiple clusters.

For e.g. if we take a Television circuit, it will have many clusters like

TUNER circuit

Demodulation circuit

Decoder circuit

Amplifier circuit

Video display

Audio controls

By identifying the clusters, it makes the testing easy.

Figure 2.19 Block-diagram-of-Colour-Television-Sets shown below.

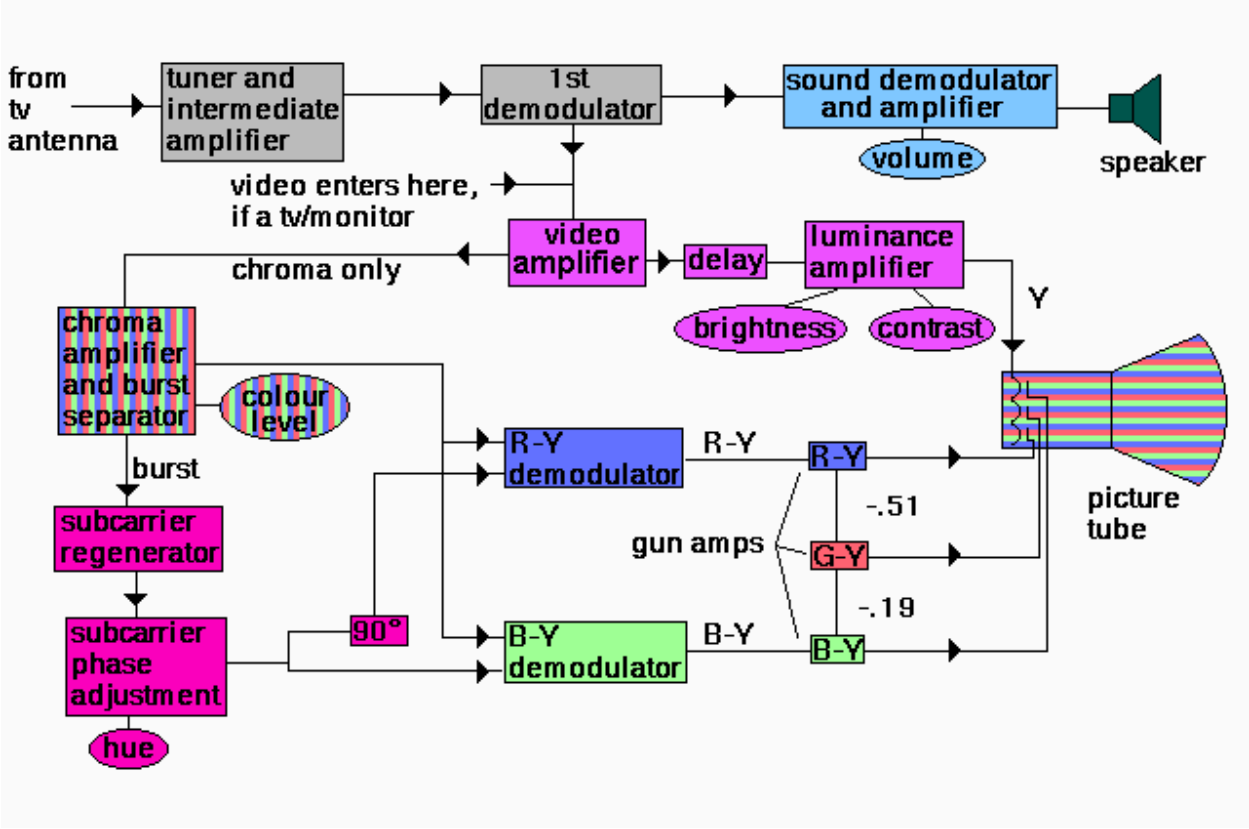


Figure 2.19 Block-diagram-of-Colour-Television-Sets

2.2.7 Simulated output/Learnt Data:

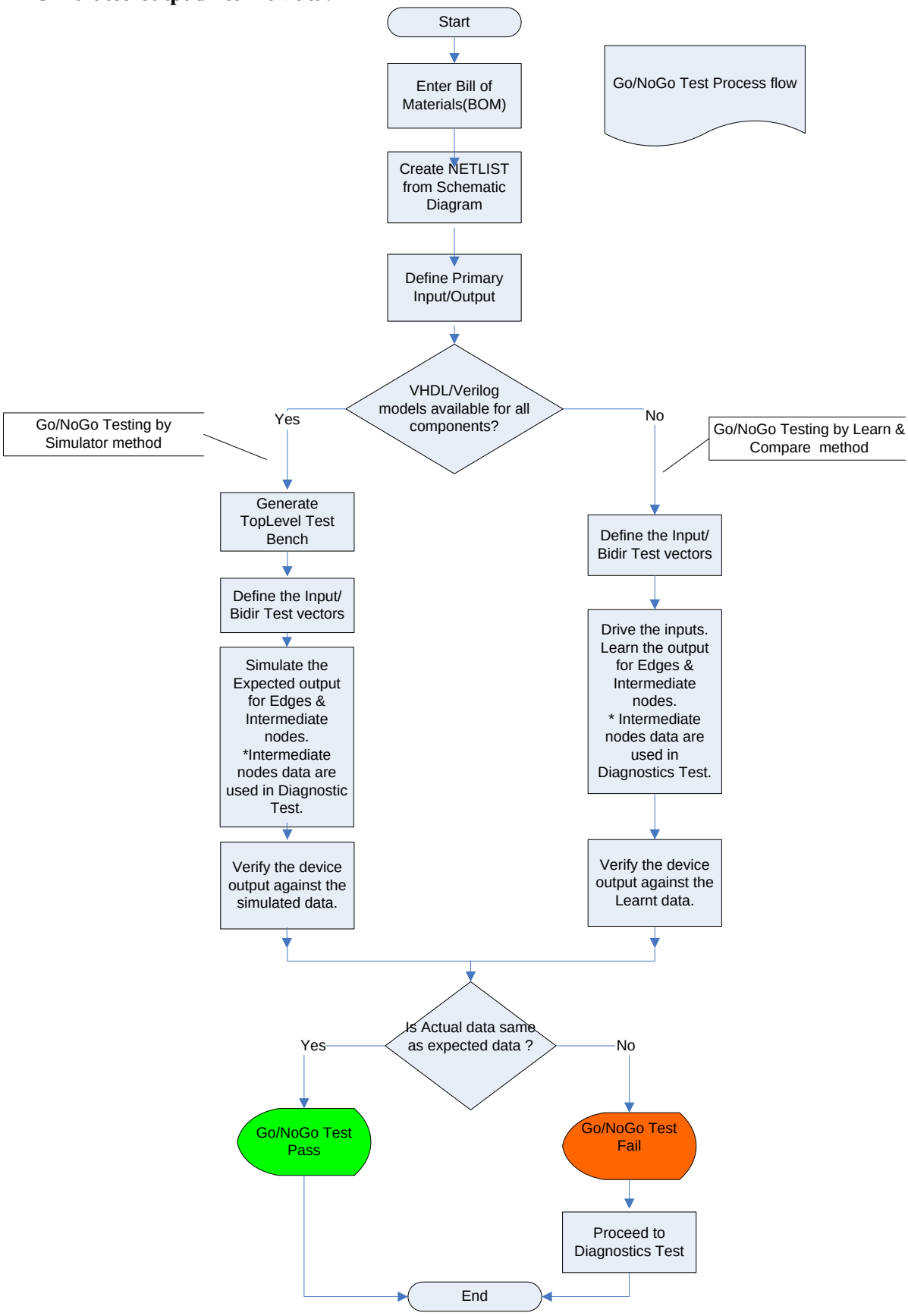


Figure 2.20 Flow Diagram – Go/NOGO Test

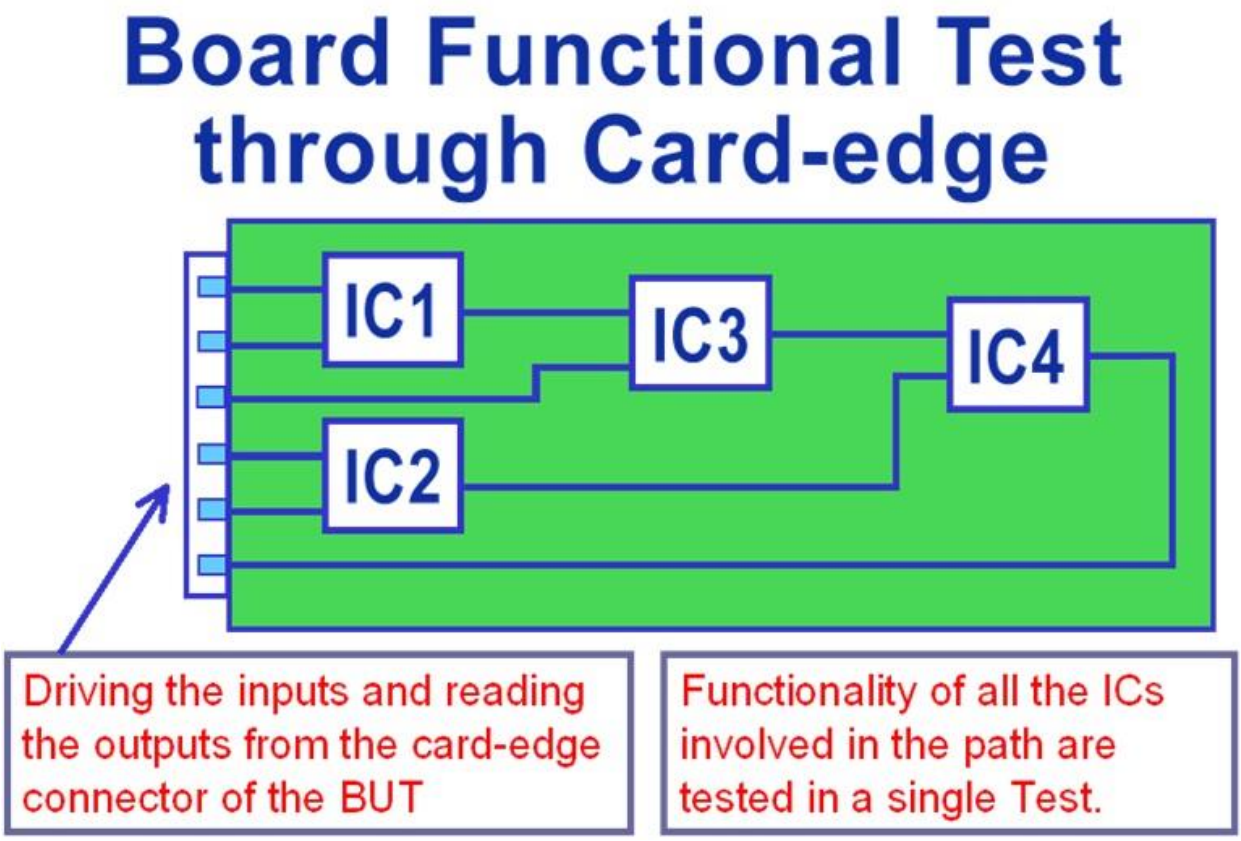


Figure 2.21 Board Functional Test Go-NoGo

Shown above figure 2.21 Board Functional Test Go-No Go

A board can be tested in 2 ways:

a) Simulation – Here the developed TPS depends on the external Simulator tools. The tools are capable of generating the stimulus data based on the models & Test vectors. The actual output from the board is compared against the stimulus and result is declared as pass/fail.

   Stimulus data can be generated for any pcb with digital/analog & mixed devices.

b) Learn & Compare- Here the expected response need to be learnt from the known good board(KGB) by probing the test pins. This will be compared against the board under Test.

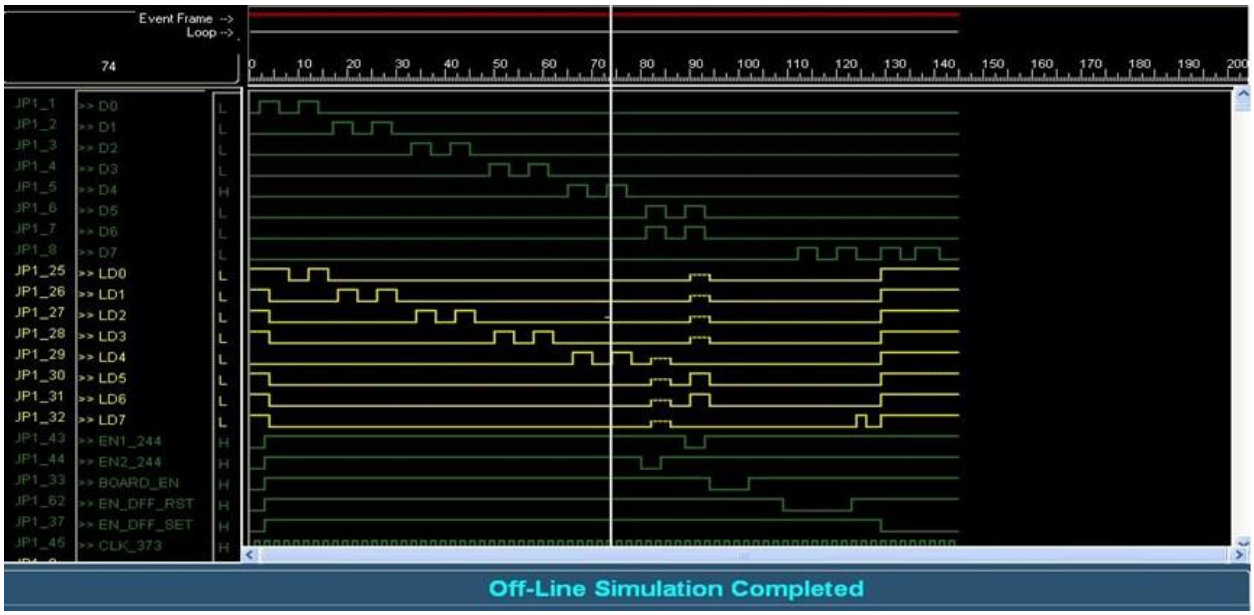


Figure 2.22 offline simulation

The waveform window shows device pin signals in the form of logic level trains for the complete duration of the test cycle. On the left of the screen are shown three columns. The first column reads the pin labels; the second column reads the NET name. The third column reads the logic level on the trains, at the instantaneous position of the cursor. The cursor position is also dynamically shown in top of the cursor with the instant timing and pattern position on the scale. Figure 2.22 offline simulation shown above.

The legends for the logic levels are as follows:

Legend	Logic Level / Signal
L	Low
H	High
Z	High Impedance
X	Undefined

Green Colour – Shows the data to be driven to input signals
Yellow Colour- Expected output from the VHDL simulator

2.2 .8 Simulation vs Learn & Compare

Simulation	Learn & Compare
Requires device models for all participating devices	Device Models not required as signature is learnt from the board
No dependency on Known Good board	Learn signature Depends on the Known Good board
Extra cost involved for simulation tools	no extra cost
Error less likely to occur	prone to human errors as probing is done manually
Test Vector can be improved by analyzing Fault Coverage	cumbersome task and with large test pins almost impossible
Less Time Consuming	More Time consuming & Tedious
off line simulation - without ATE possible with simulator.	For Learn, the ATE is must

2.2.9 Diagnostic Testing/Guided Probe Back Tracking

If the Go/No Go Test fails, next step is to identify the fault at the net/node level.

This is achieved by guided probe back tracking algorithm and it is automatic.

Each intermediate node data is captured by manually probing in Learn & Compare method from the known good board. In case of simulation method, the intermediate node data is automatically generated and stored during offline simulation.

Here, a back track tree is generated based on the Net List.

Usually from the first failure point, the probe data in the faulty board is compared. Whenever a node is probed the test program is run and the probe data gathered is compared against the Good activity for that node.In testing a PCB, even when one output is bad, the whole process of backtrack is repeated until a point on the PCB test is reached where all inputs to the PCB are good.

The algorithm is a minimum search algorithm, i.e. it does not perform an exhaustive check of all input pins to a PCB and then backtrack on the earliest failing pin, but backtracks on the first pin it finds to be failing at or earlier than the current earliest first fail.

Once a test program for a PCB has completed all the stages explained above it is normally necessary to identify the source of the fault so that the corrective action may be carried out on the PCB.

On simple PCBs it may be possible to identify the failure from the fault symptoms or from diagnostic messages included by the programmer. Many PCBs are however too complex to use this conventional approach.

In order to successfully diagnose the faults on complex and often high speed electronics testers have a variety of software and hardware tools available which permit diagnosis usually to a single failing device or manufacturing fault such as open /short in tracks or fan out problems of devices on board etc.,

Generally the entire good activity database about the signal information learned from the Good PCB is stored, as learnt signatures from PCB and this data will be compared by the software using the actual data from the PCB.

The selection of output signal leads to the particular net selected and the corresponding tree of components in the PCB net is generated automatically. Figure 2.23 Guided probe back track & Figure 2.24 Back Track Tree Generation shown below.

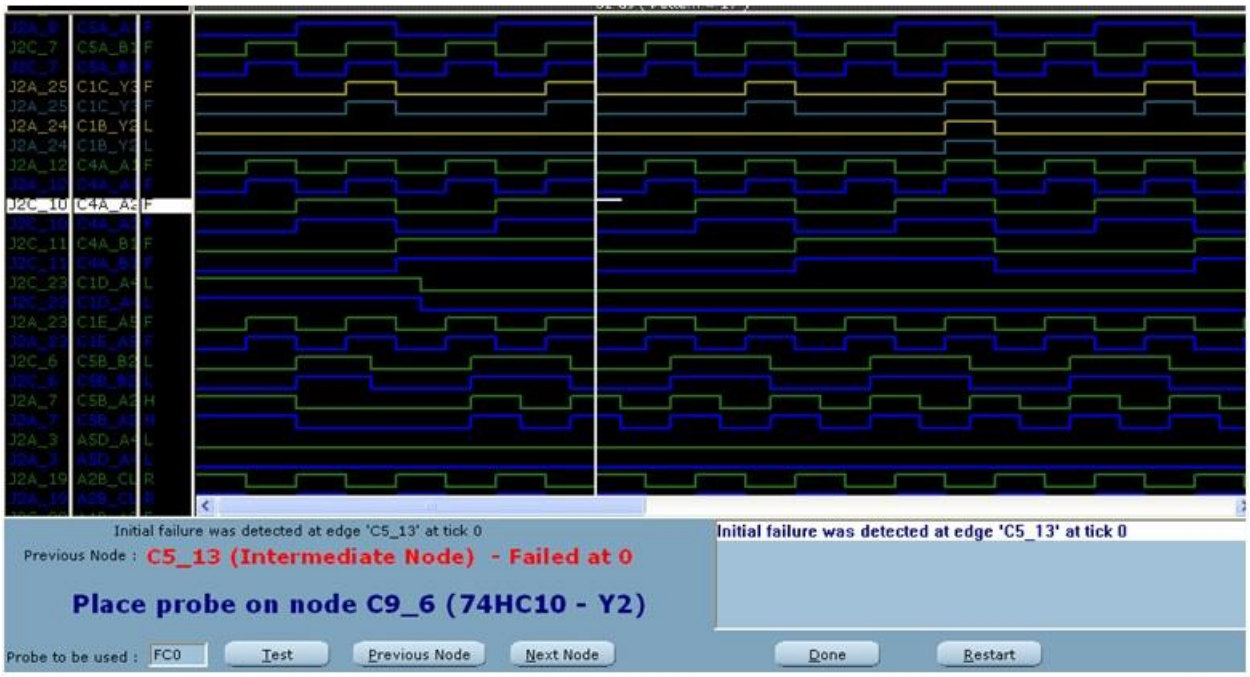


Figure 2.23 Guided probe back track

The steps continued until the fault node is identified.

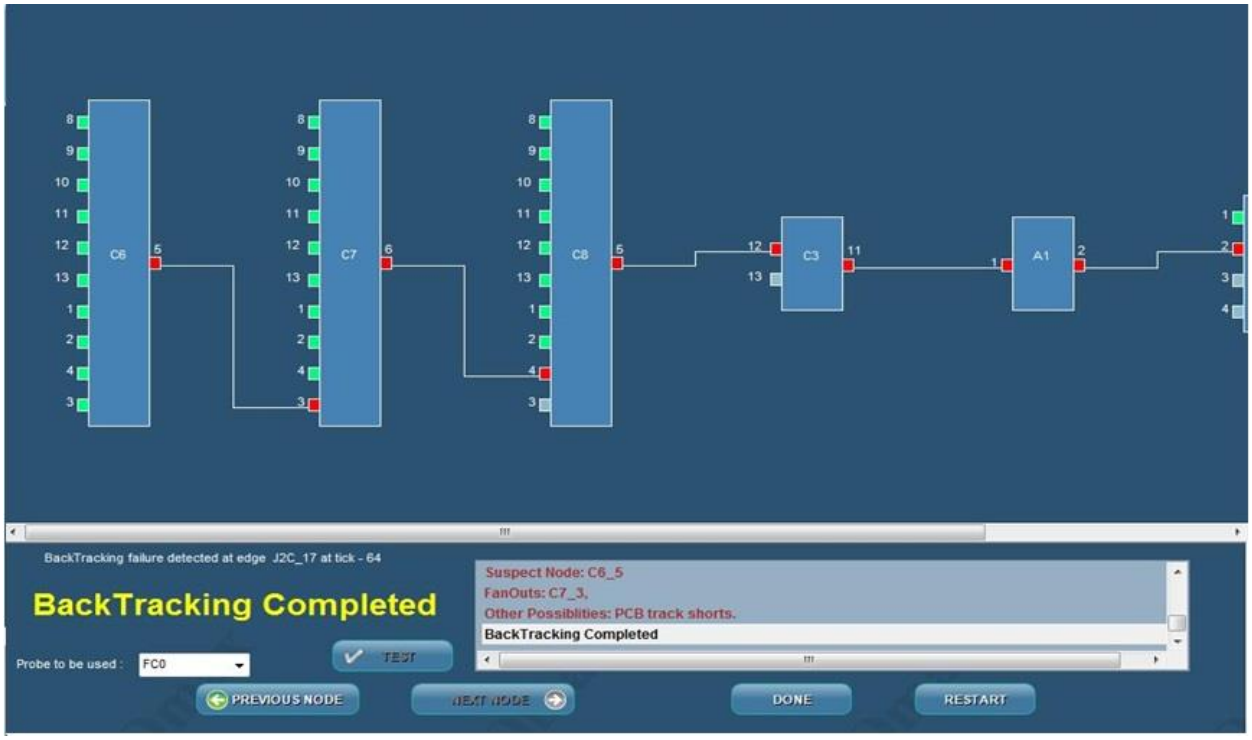


Figure 2.24 Back Track Tree Generation

2.3 ICFT/OCFT Vs Board Test

ICFT/OCFT	**Board Test**
Chip/component level testing	Whole Board tested as a device.
TPS development not required	TPS required
No Expertise Needed.	Needs expertise
Very effective for PCB repair especially when no Schematics available but the Board is with known standard devices	Requires full documentation of the board under Test.
Less Fault Coverage	Good Fault Coverage
In Expensive	Expensive

Section 3: Conclusion & Questions

3.1 Conclusion

In this, we learnt about

a) Component Level testing
   - In circuit Functional Testing (ICFT)
   - Out Circuit Functional Testing(OCFT)
   - Inputs required for Functional Testing
   - Simulators
   - Special considerations
   - Difference between ICFT Vs OCFT

b) Board Level Testing
   - Input requirements for board level testing
   - Simulation
   - Learn & Compare technique
   - Go/NOGO Testing
   - Diagnostic (Guided Probe Back tracking)

Unit 2

REVIEW QUESTIONS

PART –A (2 MARK)

1.What are the types in Component Level Testing?
2.What are the types in Board Level Testing?
3.Definition Cluster?
4.How many ways in Board Functional Test?
5.Comparison between Simulation vs Learn & Compare.
6.Why Schematic diagram is important?
7.Comparison between ICFTvs OCFT.
8.Draw the logic diagram of OR,NAND,EX-OR.
9.What is mean by TTL?
10.Write the legends for the logic levels.

PART B (3 MARKS)

1.What is a truth table? For JK flip-flop, write the truth table.
2.What are the different type of packages available?
3.Difference between out circuit &in circuit functional testing?
4.What is Threshold? Give examples for TTL, CMOS and ECL devices.
5.What is guarding? Is it applicable to both ICFT & OCFT?
6.Difference between component testing & Board Testing?
7.What are the various ways, the components can be accessed for testing?
8.What are devices with proprietary data? How will you test such device?
9.What is back drive defense limit?
10.What is clock pin termination?

PART C (10 MARKS)

1.What are the inputs to be considered for component level testing? Explain in detail
2.What are the special considerations in ICFT?
3.What are the inputs to be considered for board level testing? Explain in detail.
4.What is Go/No go testing? Explain in detail
5.What is diagnostic testing? Explain in detail
6.Difference between simulation & learn/compare method in board level testing?
7.What is auto compensation? Explain with various examples.
8.Why schematic diagram is important?

VI SIGNATURE TESTING METHODS AND TECHNOLOGY

An Introduction to VI Testing, Analog signature analysis is electronic component and circuit board troubleshooting technique which applies a current-limited AC sine wave across two points of an electronic component or circuit which is know as (DUT) Device Under Test. The resulting Current/Voltage waveform is shown on a signature display using vertical deflection for current and horizontal deflection for voltage. This unique analog signature represents the overall health of the part being analyzed. By comparing the signatures of known good circuit boards to those of suspect boards, faulty nets and components can be quickly identified. Figure 3.1 Shows a diagram of Analog Signature Analysis Test Techniques shown below.

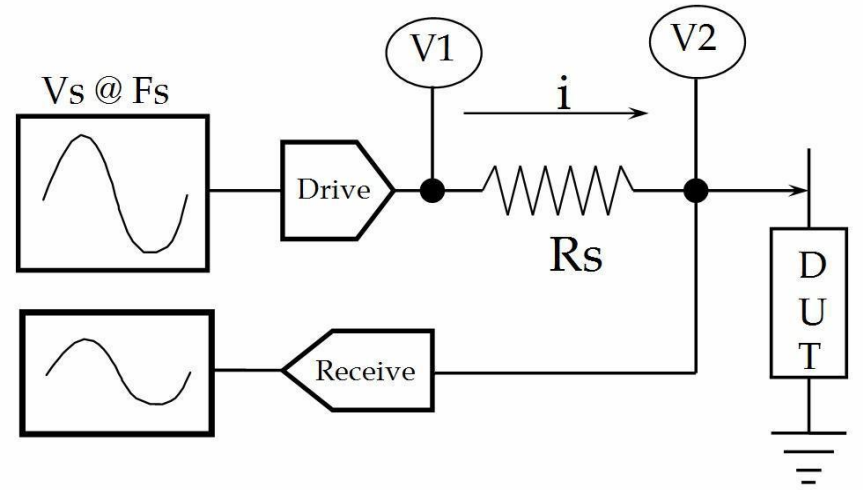


Figure 3.1 Shows a diagram of Analog Signature Analysis Test Techniques

Here: - Vs = Voltage Source, is used to vary the amplitude.
Fs = Frequency Source, is used to vary the frequency range.
Rs = Source Impedance, is to limit the current.

3.1 Theory of Operations

Analog signature analysis test system applies a test signal across two terminal of the device (DUT). This test signal causes current to flow through the device and a voltage drop across its terminals.

The current flow is processed in such a way as to cause a vertical deflection of the scope trace, while voltage drop across the test components causes a horizontal deflection of the scope trace.

Usually, we keep constant values in x axis and variable values in y axis. In VI characteristics, the current value 'I' can be varied depends on resistance (name as source impedance), where applied voltage is constant.

3.2 VI Characteristics Display

A VI Characteristics display has a graticule consisting of a horizontal axis which represents

voltage, and a vertical axis which represents current. The axis divided the display into four quadrants. Each quadrant displays different portions of the VI Characteristics signature.

Quadrant I displays positive (+V) and positive (+I) current
Quadrant II displays positive (-V) and positive (+I) current
Quadrant III displays positive (-V) and positive (-I) current
Quadrant IV displays positive (+V) and positive (-I) current

Figure3.2 Shows VI characteristics display divided into Four Quadrant shown below.

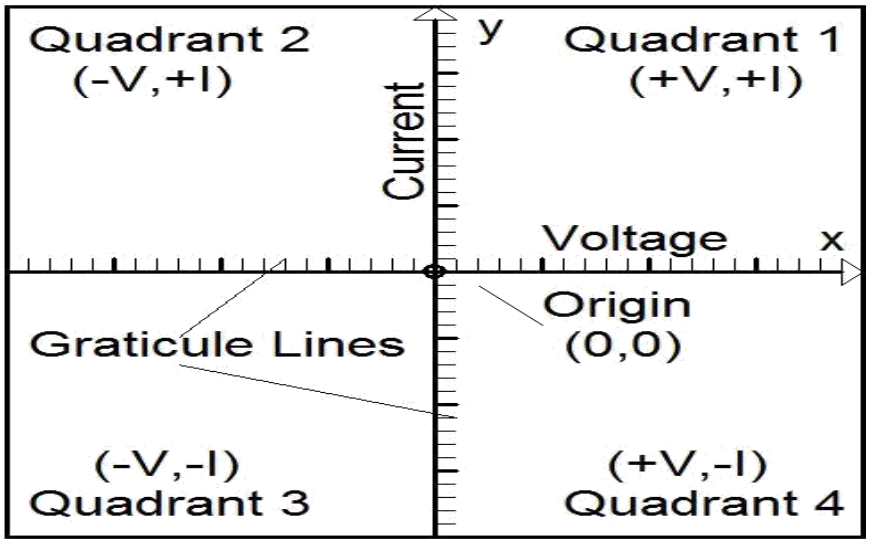


Figure3.2 Shows VI characteristics display divided into Four Quadrant

3.3 VI - Effective Test System

During handling or even manufacturing printed circuit boards and integrated circuits, defects may develop. Those defects such as open circuits or short circuits may appear in or between circuit pathways and electronic components. Effective testing system is necessary for maintenance purposes and also for manufacturing quality insurance. Shown below figure 3.3 shows manual testing methods with various measuring instruments

Analog Signature Analysis relies on a change in electrical characteristics to detect problems on a circuit board. Nodal impedance test is a proven method of troubleshooting electronic circuit cards, and is independent of the technology under test. Nodal impedance test is also called V/I trace test or ASA - analog signature analysis. The technique is equally effective in locating faults on both analog and digital PCBs.

The rapid development of electronic module assembly manufacturing requires a parallel development in test procedures. Printed circuit boards (PCBs) testing is becoming more ex- pensive and difficult due to the complexity of PCBs design. The common methods for diagnosing PCBs still suffering from many difficulties; it needs long time, a lot of manual work, direct contact

with PCB, and it is so expensive. Previously, the unique method to inspect printed circuit boards was manual testing method; it involves using visual inspection, multi meters, oscilloscopes and other testing equipments.

This method is almost inapplicable for the recent printed circuit boards since the huge mounted number of components installed on PCBs. Moreover, using integrated circuits (ICs) limits the ability of manual testing and makes it so difficult. Manual testing takes long time to be performed.

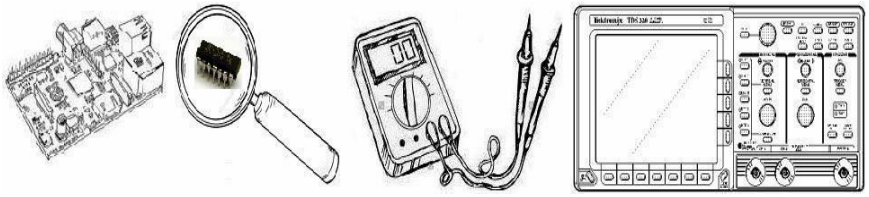


Figure 3.3 Shows Manual testing methods with various Measuring Instruments

The efficiency of such diagnose method depends on the repairer knowledge and experience. In manual testing, al- ways the repairer needs to choose the suitable testing equipment according to the device to be tested.

3.4 Fundamental of Electrical Characteristics

Fundamentals of electrical characteristics are based on four basic VI traces available: Resistance traces, Inductor traces, Capacitance traces and Semi-conductor traces. Traces of any other component will be a combination of these four basic traces. Figure 3.4 shows four basic vi characteristics shown below

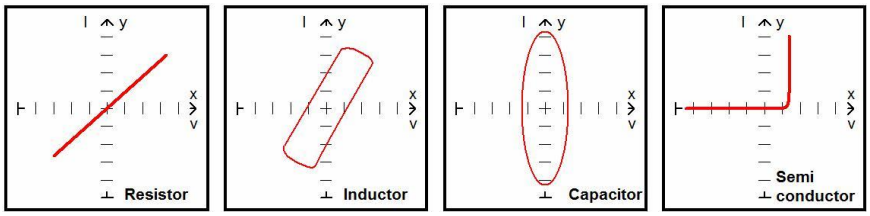


Figure 3.4 shows four basic VI characteristics

3.5 Open Circuit Trace:-

In the figure 3.5 shows open circuit vi characteristics**.** An open circuit is the absence of a connection between two terminals. An “Open Circuit” condition would have zero current flowing

through the test leads and would show maximum voltage across test the leads. This is represented by straight horizontal trace from maximum left from the maximum right scope face as show in figure Open Circuit VI Characteristics.

The voltage across the component under test controls the amount of horizontal trace deflection on the instrument display. When the component under test is removed, creating an open circuit (e.g., RL = ∞), the voltage at the output terminals is at its maximum and thus the trace on the display is a straight horizontal line with its maximum width.

The open circuit maintains zero current and can allow any voltage drop, so its I-V characteristic is a horizontal line at I=0. Notice that an open circuit behaves identically to a zero current source with Is=0 A.

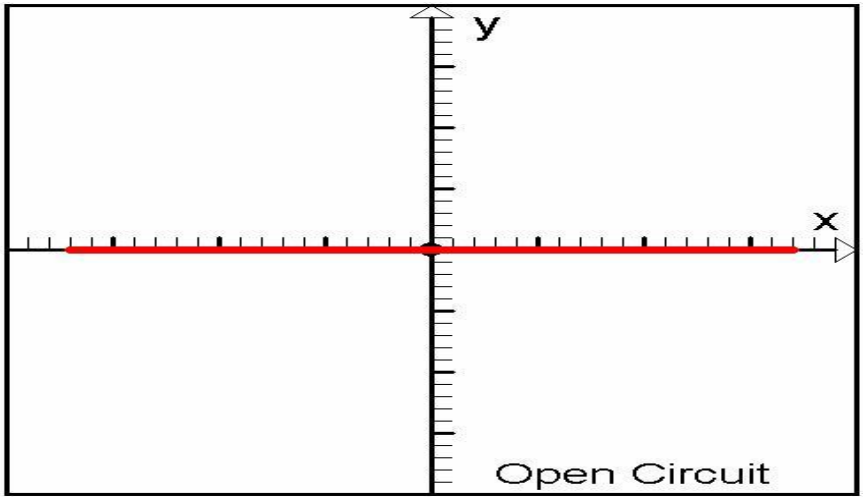


Figure 3.5 shows Open Circuit VI characteristics

3.6 Short Circuit Trace:-

A short circuit is a direct connection between two terminals. The short circuit maintains zero voltage drop and can allow any current, so its I-V characteristic is a vertical line at V = 0. When the RL is zero ohms (0Ω) by shorting the output terminal to the common terminal, there is no voltage dropped across RL causing no horizontal component displayed in the analog signature. This short circuit signature is a vertical line trace on the instrument display as shown in Figure Short Circuit VI Characteristics.

The amount of vertical trace deflection on the instrument display is controlled by the voltage dropped across the internal impedance Rs of the instrument. Because Rs is in series with the load RL, this voltage will be proportional to the current flowing through RL. Figure 3.6shows Short Circuit VI characteristics Shown below.

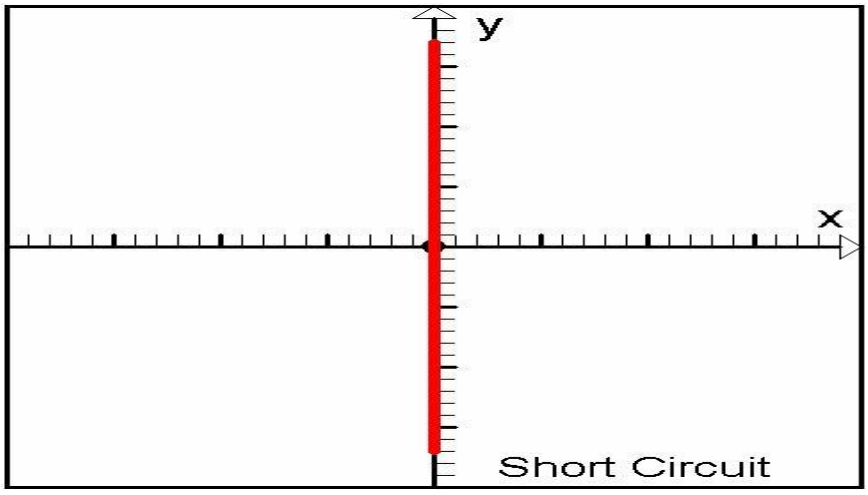


Figure 3.6shows Short Circuit VI characteristics

3.7 Resistor Trace:-

Resistors will be represented by a straight line with the angle of the slope being directly proportional to the value of resistance. A resistor R satisfies Ohm's law I=V/R, so its I-V characteristic goes through the origin and has slope 1/R.

The electric current I, flowing through a resistor R, is linearly proportional to the voltage V, across it. The displayed signature of a resistor's I-V curve will produce a straight line on the screen, inclined at an angle dependent on the resistance value of the resistor. The trace will rotate anti-clockwise about its central origin. The degree of rotation is a function of resistance value.

The angle of incline depends on the value of R (load) in comparison with R (sense). A short-circuits (zero resistance) will produce a vertical line whilst an open circuit (infinite resistance) will produce a horizontal line. When R (load) has the same resistance value as R (sense), the straight-line signature will be inclined at an angle of 45º because the voltage across each resistance is equal.

Show in the Figure 3.7 Resistance is given by the slope of **V** verses **I** graph.

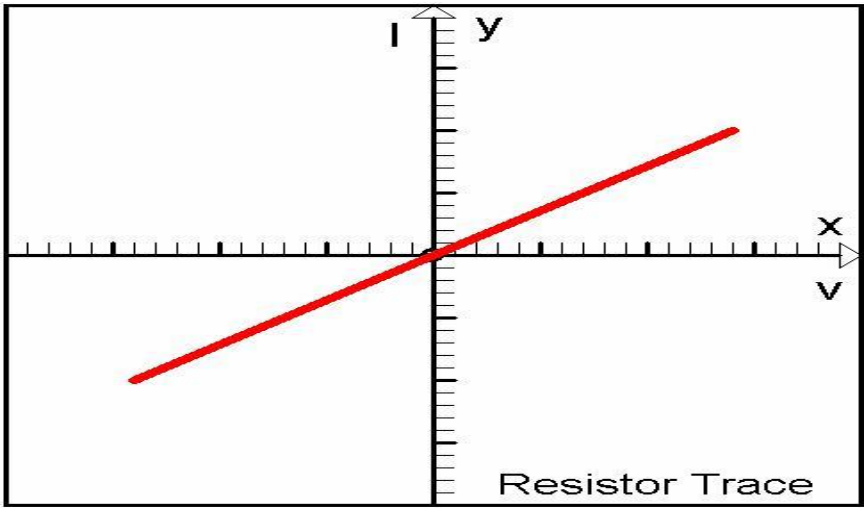


Figure 3.7 shows Resistor Trace VI characteristics

Pure resistance is not frequency sensitive because the voltage across R (load) is always in phase with the current through it. This is not the case where a reactive component is added to R (load) in the form of an inductance or capacitance. The voltage and current are then out of phase with each other producing a different signature or trace.

3.8 Capacitor Trace:-

Show in the figure 3.8 shows capacitor trace vi characteristics.In both digital and analog electronic circuits a capacitor is a fundamental element. It enables the filtering of signals and it provides a fundamental memory element. The capacitor is an element that stores energy in an electric field.

The current going through a capacitor and the voltage across the capacitor are 90 degrees out of phase. It is said that the current leads the voltage by 90 degrees. In DC the capacitor acts as an open circuit, Capacitors do like to pass current at low frequencies. Capacitors like to pass current at high frequencies. Capacitors connected in parallel combine like resistors in series. Capacitors in series combine like resistors in parallel.

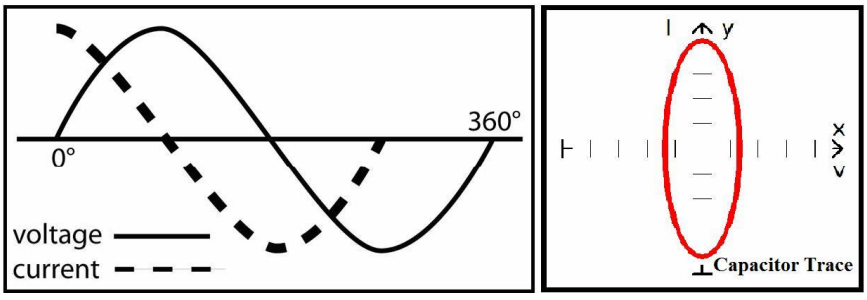


Figure 3.8 shows Capacitor Trace VI characteristics

3.9 Inductor Trace:-

The current going through an inductor and the voltage across the inductor are 90 degrees out of phase. Here the voltage leads the current by 90 degrees. Figure 3.9 shows Inductor Trace VI characteristics

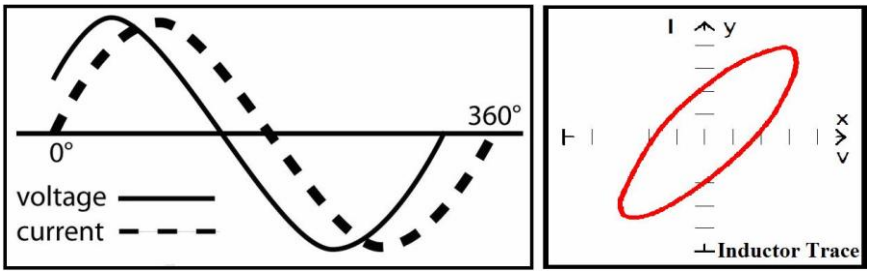


Figure 3.9 shows Inductor Trace VI characteristics

3.10 Semiconductor Trace:-

Diode, the simplest semiconductor, allows current to flow in one direction and not the other. These signatures are displayed by a horizontal line that goes vertical just after the center axis of the display. A non linear component such as a semiconductor junction would allow a large current to flow during the half cycle when it is forward biased and a very little current to flow during the reverse biased half cycle. Also the voltage drop across the junction would be small: i.e., 0.7V volt: this would appear as a near short during the forward biased mode and would cause a vertical trace to appear during that portion of the cycle. The reverse biased condition would cause very little current to flow with a large voltage drop and look like a horizontal trace on the scope. Figure 3.10 shows Inductor Trace VI characteristics shown below.

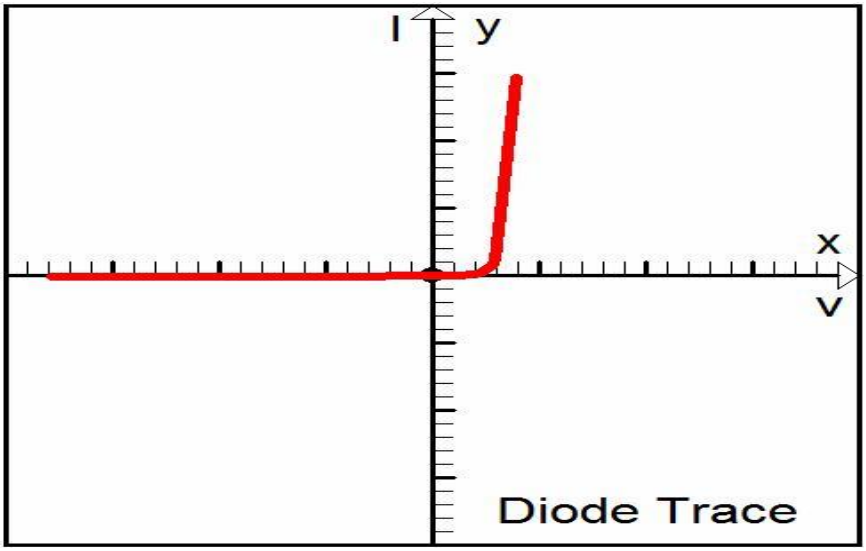


Figure 3.10 shows Inductor Trace VI characteristics

Integrated circuits are made up of transistors (whose trace will be like back-to-back connected diodes). The signatures displayed by ICs are composite signatures. These devices have built-in protection circuits

which allow current to flow in both directions displaying a signature that is vertical in the bottom half, then horizontal, then vertical in the top half, looking very much like a zener trace shown in figure above.

The signatures for ICs will vary from manufacturer to manufacturer. CMOS integrated circuits are built with capacitors causing their signatures to display a loop. Integrated circuits of the same type (e.g.7400) but differing in manufacturer can display slight differences because of the various ways in which the components are constructed. This is a normal condition and should not be confused with a device failure.

3.11 Effect of Curve Trace

Reasons for Signature Difference in VI Characteristics:

While testing using VI tester, not all differences in trace can be taken as faults.We need to remember that VI tester is only a tool that helps to point out the differences and it is for the user to do the fault finding.

In the case of VI trace Board Learn & Test modes, the comparison signature is the signature saved during the Board Learn mode. When the "Test" signature deviates from the "Learnt" signature by a user set amount (tolerance), the program logs that particular pin as being different.

First of all, let us understand the reasons for difference in VI trace between boards. The trace difference could be due to any of the following reasons:

1. Difference due to variations in the manufacturing process by different IC makers.
2. Some switches and jumpers are at different settings.
3. Trim pots are adjusted differently in the learnt & the tested boards.
4. The learnt & tested boards are of different versions.
5. Difference due to noisy and oscillating signatures, charging of capacitors.

Interpreting the reason for difference in VI trace comes from practice. Let us look into each of the above causes to have a better understanding of the VI trace differences.

Difference due to manufacturing process by various IC makers:

The signatures for ICs will vary from manufacturer to manufacturer. CMOS integrated circuits are built with capacitors causing their signatures to display a loop. Integrated circuits of the same type (e.g.7400) but differing in manufacturer can display significant differences in VI trace because of the various ways in which the components are constructed even though the two makes are functionally identical. This is a normal condition and should not be confused with a device failure.

To deal with such Make related problems

Look for similar traces among same type of pins. For example, all of the output pins on a TI 74LS00 will have the same VI trace w.r.t Gnd in out circuit. The output pins of a Motorola 74LS00 should also have similar signatures w.r.t Gnd although they may be different from that for a TI device. Similarly all pins in the Address or Data bus of a device can be checked to have similar signature w.r.t the GND pin. Select the ranges that minimize the effect of trace variation due to manufacturer difference. Lower range of 2.5 V, although best for checking shorts in a board, will tend to exaggerate differences due to MAKE.

Noisy or Oscillating Signatures

Some components will exhibit signatures that appear noisy or tend to oscillate. Generally, this is not considered a problem but a common occurrence among CMOS devices. However, if you are used to testing

certain components without experiencing any oscillations and suddenly come across one that does then that particular component could have a problem. Here are some suggestions for dealing with oscillating signatures:

Using the Interactive VI mode on these types of components, is not normally a problem since you are viewing the signatures at "real-time" on the Trace View window.

When using the Board Learn & Test mode, traces are taken at only one particular point in time. This can cause signatures to compare as "different" since the learnt & tested traces may have not been captured at the same instant. To help eliminate these small differences, try using the Step method. Increasing the number of samples to 2, 4 or 8. Will not help with oscillating signatures but simply increase the test time.

3.12 Difference due to Capacitor Charge/Discharge:

Another cause of signatures changing is due to capacitors charging or discharging on the board as the VI trace is taken. These differences typically show up as a change in the horizontal (voltage) portion of the signature. The charging effect can sometimes be seen when viewing the traces in the interactive mode. To deal with differences caused by the charging effect of capacitors, try the following:

Increase the number of samples to 2, 4 or 8. This will allow the capacitive circuit to stabilize before the signatures are captured.

Shorting out the capacitive circuit will eliminate the charging effect but could cause you to miss important signature differences. This method can be used as a last resort.

Difference due to version change/switch setting, jumpers, pots:

Comparison of traces between boards with version change, difference in switch or jumper setting, pots trimmed differently can cause significant differences in the VI trace. It is important to make a note of these differences using the Board Setup option for future reference.

All the switch and jumper setting etc can be entered in the Board Setup option during Board Learn. It is also important to test identically matching boards in the Board Learn / Test mode. Otherwise you could waste a lot of time in isolating a signature difference that is caused by version change and not a fault in the board.

The PCBs are expected to give trouble free performance over their expected life duration. A factor like MTBF (mean time between failures) is calculated for a product based on the reliability factors of its constituents. This helps in planning a replacement or expected maintenance involving purchase of spares for the product.

Effects of current on capacitor

Characteristics of Passive and Active Components

While testing using VI tester, not all differences in trace can be taken as faults. We need to remember that VI tester is only a tool that helps to point out the differences and it is for the user to do the fault finding.

This section will explore how VI characteristics are differing with passive and active components while change in test value voltage, source impedance and frequency range.

3.13 Resistor VI Characteristics:

Figure 3.11 shows 100Ω Source Impedance with different resistor load trace & figure 3.12shows 1KΩ Source Impedance with different resistor load trace

Testing the resistor with different load and source impedance. When R (load) has the same resistance value as R (sense), the straight-line signature will be inclined at an angle of 45º because the voltage across each resistance is equal.

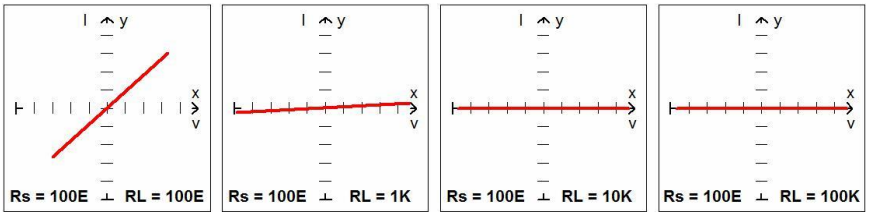


Figure 3.11 shows 100Ω Source Impedance with different resistor load trace

In the signatures above note that as the value of the resistance being tested increases, the signature displayed becomes more horizontal. Note that the signature of the 100Ω resistor is showing a typical angled of 45° because the voltage across each resistance is equal, straight line resistive signature but the other resistor signatures are still relatively flat.

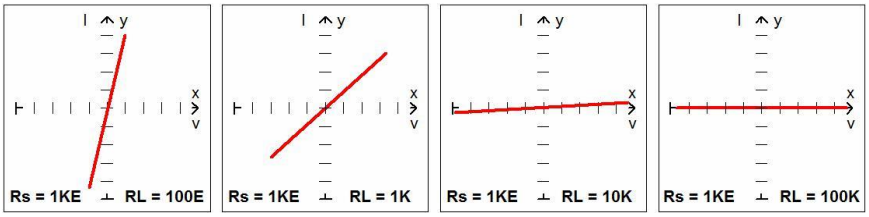


Figure 3.12shows 1KΩ Source Impedance with different resistor load trace

Show in the figure 3.13shows 10KΩ Source Impedance with different resistor load trace.Note that the signature of the 100Ω is now more vertical. The 1KΩ resistor is showing typical angled, straight line resistive signature but the other resistor signatures are still relatively flat.

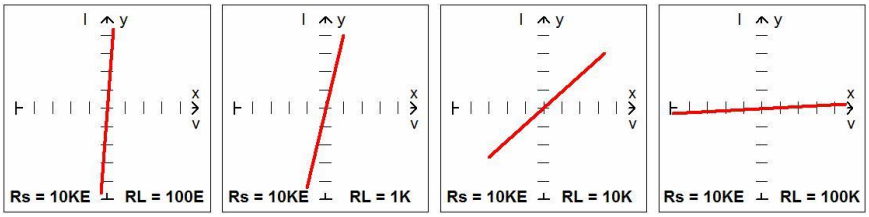


Figure 3.13shows 10KΩ Source Impedance with different resistor load trace

Note that the signatures of the 100Ω and 1KΩ resistors are showing almost vertical, straight line resistive signatures. The 10KΩ resistor is showing a typical straight line angled signature and the 100KΩ resistor is relatively flat. Figure 3.14shows 100KΩ Source Impedance with different resistor load trace Shown below.

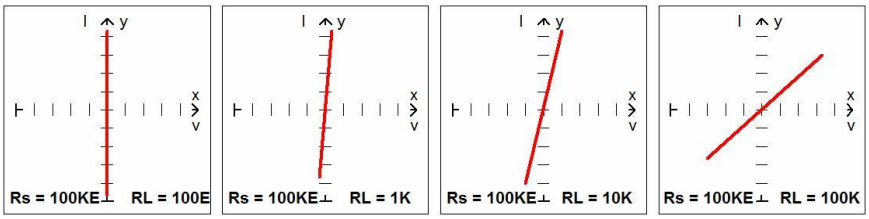


Figure 3.14shows 100KΩ Source Impedance with different resistor load trace

Note that the signatures of the 10KΩ and 100KΩ resistors are showing typical angled, straight line resistive signatures but the 100Ω and 1KΩ resistor signatures are almost vertical. You may also see slight capacitance (loop signature) as the resistance value is increases. This happens as the internal capacitance of the instrument begins to show in the signature.

Resistive Signatures - Changing Voltage and Frequency

The analog signature of a resistor does not change when voltage is varied.

The resistive load across the test leads remains unchanged compared to the internal source resistance of the instrument.

The analog signature of a resistor does not change when frequency is varied.

Resistors are not a reactive device and the resistive load across the test leads remains unchanged compared to the internal source resistance of the instrument.

Review for resistive signatures

The signature of a purely resistive circuit will display a straight line signature because the relationship between voltage and current in a purely resistive circuit is linear.

This straight line signature can vary from a completely horizontal (open circuit) to completely vertical (short circuit).

As resistance across the test leads increases, the current decreases and the signature will become more horizontal.

As the ASA test instrument range resistance increases, a resistive signature becomes more vertical.

3.14 Capacitor VI Characteristics:

Unlike resistive circuits, the relationship between induced voltage, current and capacitance is not linear. In a capacitive circuit, current is at its maximum when voltage across the component is at zero. When voltage across the component is at its maximum, current in the circuit is at zero. This “time delay” is a function of the capacitive reactance where the current leads the voltage.

The time delay in a capacitive circuit causes the analog signature to be displayed as an elliptical shape. The width of the ellipse is directly related to the value of the capacitor being tested and the range parameters set on the instrument.

Changing Component Capacitance Value

The following examples illustrate how changing the capacitor value under test affects the signature displayed on the instrument display. Figure 3.15 shows 100Ω Source Impedance with different capacitor load trace shown below.

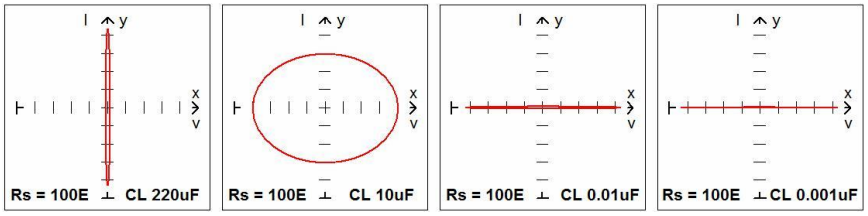


Figure 3.15 shows 100Ω Source Impedance with different capacitor load trace

Note from the signatures shown above that as the value of the capacitance being tested decreases, the signature displayed becomes more horizontal.

Source Impedance less than 100Ω ranges are best suited for large capacitor values. Note that elliptical signatures are displayed for the 220uF and 10uF.Figure 3.16 shows 1KΩ Source Impedance with different capacitor load trace shown below.

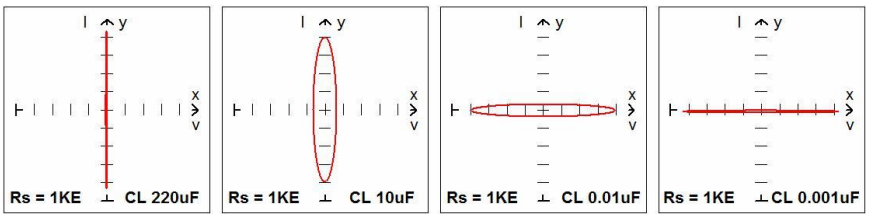


Figure 3.16 shows 1KΩ Source Impedance with different capacitor load trace

For 1KΩ Source impedance, note that the 10uF and 0.01uF capacitors display a typical elliptical signature. The signatures displayed for the other components appear either as a vertical or horizontal line.

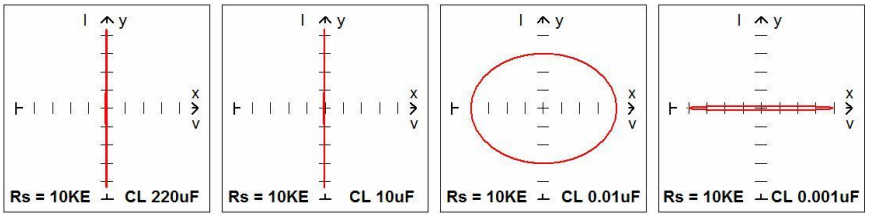


Figure 3.17 10KΩ Source Impedance with different capacitor load trace

Figure 3.17 10KΩ Source Impedance with different capacitor load trace & Figure 3.18shows 10KΩ Source Impedance with different capacitor load trace shown below For 10KΩ source impedance, note that the signature of the 0.01uF capacitor displays a typical elliptical signature and the .001uF capacitor is now showing a very slight elliptical shape.

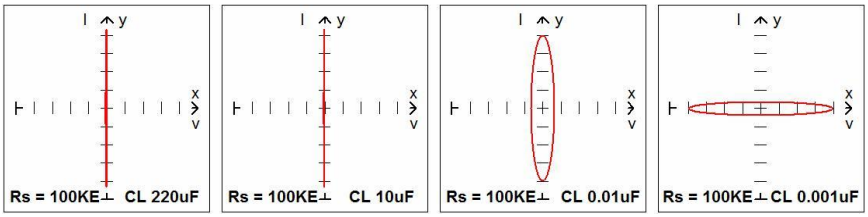


Figure 3.18 shows 10KΩ Source Impedance with different capacitor load trace

Note that the signatures of the 0.01uF and .001uF capacitors display an elliptical signature. The signatures displayed for the other components will appear as a vertical line.

The analog signature of a capacitor does not change when voltage is varied. The capacitive reactance across the test leads remains unchanged compared to the internal source resistance of the instrument.

Effects of changing range frequency on capacitive signatures

The analog signature of a capacitor changes when the test frequency is varied since the capacitive reactance is a function of frequency. Lower frequencies work better for testing larger capacitances.

The analog signature of a capacitor changes when the instrument frequency is varied since the capacitive reactance is a function of frequency.

Higher frequencies work better for testing smaller capacitances. Figure 3.18 (a) shows 10uF capacitor Good and Bad Traces shown below**.**

Leakage Capacitors

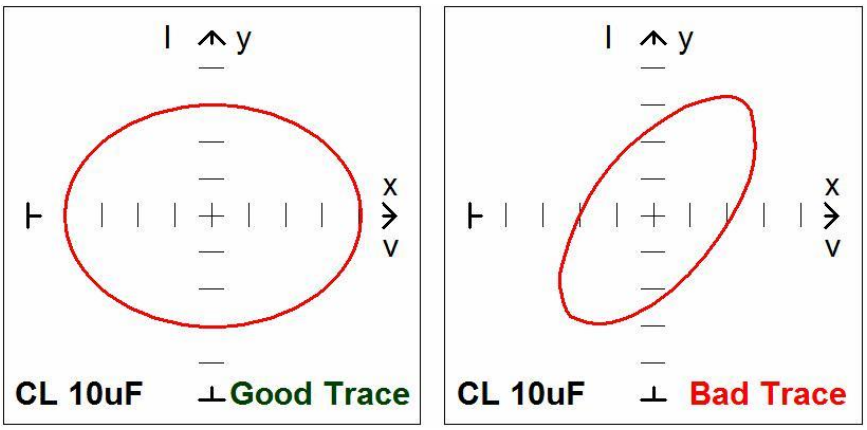


Figure 3.18 (a) shows 10uF capacitor Good and Bad Traces

Note that the signatures of the capacitors appear different from each other in that the bad capacitor is showing an angled orientation when compared to good one. Capacitive leakage or dielectric failure is a common type of failure especially in electrolytic capacitors.

Review for capacitive signatures

The signature of a capacitor will display an elliptical or circular signature because the relationship between voltage and current in a capacitive circuit is out of phase.

As the internal instrument test resistance changes, the reactive load across the test leads changes in relation to the internal source resistance and will cause the displayed signature to change.

As test frequency increases, the signature will become more vertical due to decreasing capacitive reactance within the capacitor being tested. High frequencies work best for small capacitor while low frequencies work well for large capacitors. Radial lead electrolytic capacitors can be test by touching the exposed metal top of the component.

One common capacitive failure type is capacitive leakage where the signature will appear to be tilted at an angle. This is indicating internal resistance and is primarily an issue with electrolytic capacitors.

3.15 Inductors VI Characteristics:

Similar to capacitors, the relationship between induced voltage, current and inductance is not linear. In an inductive circuit, voltage and current are out of phase with current lagging voltage. This “time delay” is a function of the inductive reactance where the voltage leads the current.

The time delay in an inductive circuit causes the analog signature to be displayed as an elliptical shape. The width of the ellipse is directly related to the value of the inductor being tested and the range parameters set on the instrument. Since pure inductors are theoretical, most inductive signatures exhibit a resistive tilt caused by the resistance of the wire used to construct the component and some degree of distortion such as the “egg” shape caused by inductive hysteresis.

Changing Component Inductance Value

The following examples illustrate how changing the inductor value under test affects the signature displayed on the instrument display. Figure 3.18shows 100Ω & 1KΩ Source Impedance with different Inductor load trace shown below.

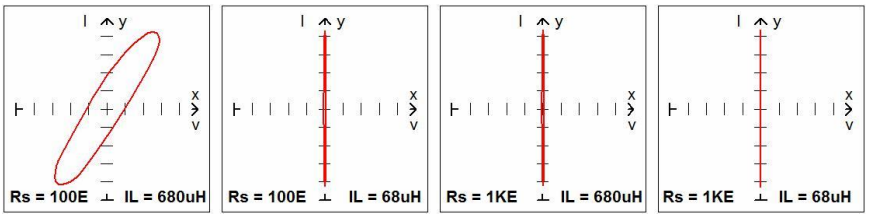


Figure 3.18shows 100Ω & 1KΩ Source Impedance with different Inductor load trace

Note from the signatures shown above that as the value of the inductance being tested decreases, the signature displayed becomes more vertical. The width of the signature will also change.

Note that the elliptical signatures have changed in both angle and shape when compared to the signatures using the 10Ω Tracker resistance setting.

Note that the signatures are now very close to vertical. This particular resistance setting is not suitable for testing these two inductors. The 10KΩ and 100KΩ resistance settings will also display vertical signatures.

3.16 Effects of changing range voltage on inductive signatures

Inductive signatures will change very little when voltage is varied. The inductive reactance across the test leads remains relatively unchanged compared to the internal source resistance of the instrument.

Effects of changing range frequency on resistive signatures

Inductive signatures will change when the test frequency is varied since the inductive reactance is a function of frequency. Higher frequencies work better for testing larger inductances.

Review for inductive signatures

The signature of an inductor will display an elliptical or circular signature because the relationship between voltage and current in an inductive circuit is out of phase. The signature will typically display a resistive tilt caused by the resistance of the wire used to construct the component.

As the internal test resistance changes, the reactive load across the test leads changes in relation to the internal source resistance and will cause the displayed signature to change.

As test frequency increases, the signature will become more horizontal due to increasing inductive reactance within the inductor being tested. High frequencies work best for large inductor values while low frequencies work well for small inductor values. Use the information presented in table 4-1 as guide for the minimum and maximum limits at the extreme range combinations listed. These values will vary depending on the ASA test instrument used.

Because inductors come in various types and values and can exhibit wide varieties of signature distortion, troubleshooting inductive components are best accomplished using comparisons.

Testing Diodes Introduction

Diodes semiconductors are the basic building block of all other semiconductors (i.e. transistors, integrated circuits). Diodes are formed by creating a junction between P-type and N-type semi conductive materials. The resulting component exhibits polarity and will conduct electrical current in one direction but not the other. Current flows in a diode when the voltage on the positive terminal (anode) is more positive than the negative terminal (cathode). This characteristic is reflected in the instrument signature displayed. Figure 3.19 shows a typical diode signature and it symbol shown below.

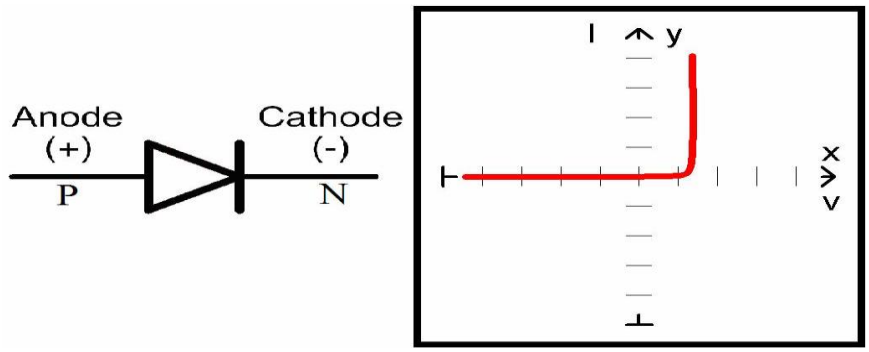


Figure 3.19 shows a typical diode signature and it symbol

Diode signatures reflect the basic nature of a semiconductor junction. There is a threshold voltage at which the diode begins to conduct, typically 0.6V for a silicon based component. As long as the anode to cathode voltage differential remains below the threshold, the diode will act as an open circuit. As the anode to cathode voltage increases positively the diode will begin to conduct. Once the current flow begins, very small increases in anode voltage will cause very large increases in current flow.

Effects of Changing Source Frequency on Diode Signatures

The following exercises illustrate how a diode signature changes in response to variation of instrument frequency, resistance and voltage settings.

Here the signature changes very little when changing from a high frequency to its lowest. The voltage characteristics inherent to a diode are not sensitive to changes in frequency. Figure 3.20shows the change in frequency diode signature shown below.

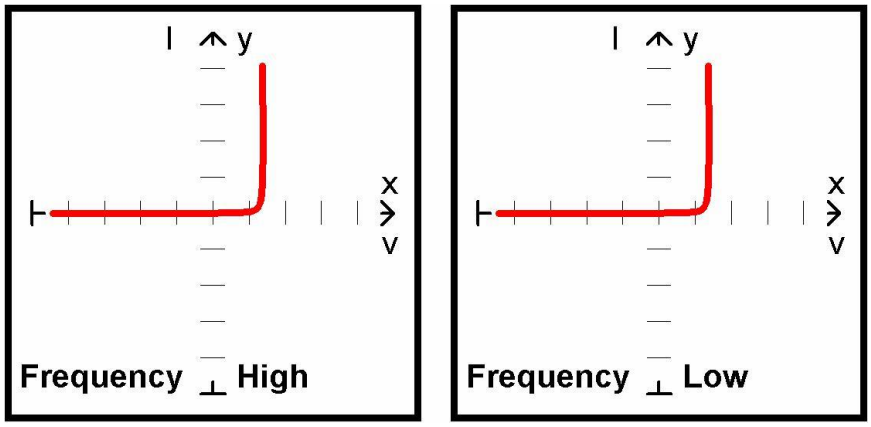


Figure 3.20shows the change in frequency diode signature

Effects of Changing Source Impedance on diode signature

Note that the signature changes very little when the resistance is varied from a low setting to a high setting. The slight variation observed to due to the change in available current from the instrument. Figure3.21 shows the change in source impedance diode signature shown below.

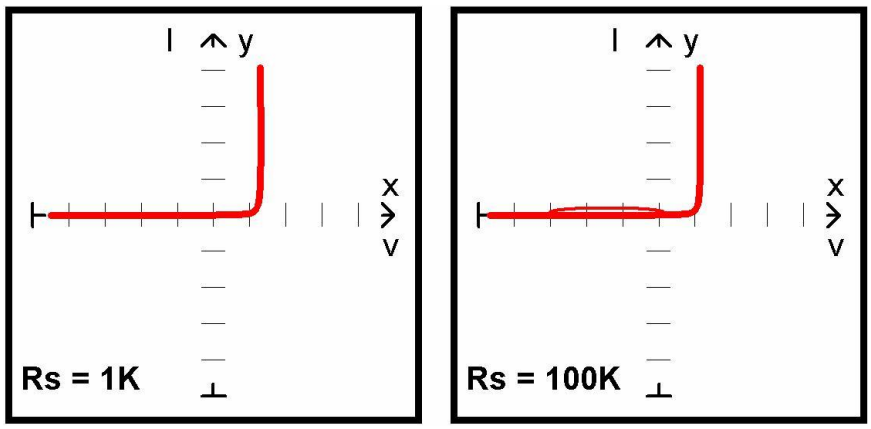


Figure3.21 shows the change in source impedance diode signature

Since a diode has either very high or very low resistive characteristics depending on the voltage potential across the component, changing the instrument resistance setting will have little effect.

Effects of Changing Source Voltage on diode signature

Note that the signature appears to change its breakdown point but what is actually happening is that the horizontal voltage scale is changing. As the voltage increases, the volt per division increases making the diode signature change. Figure 3.22 shows the change in source voltage diode signature shown below.

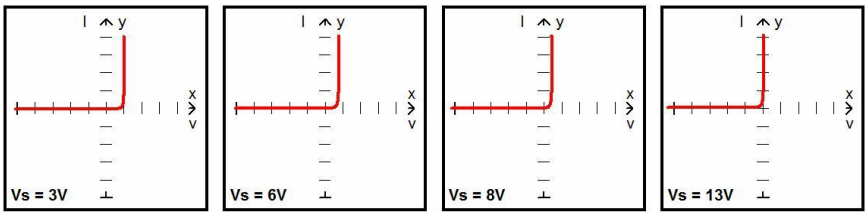


Figure 3.22 shows the change in source voltage diode signature

Diode Failures

Other than open or short circuits, semiconductor failures are generally resistive in nature.
In The Figure 3.22 shows diode signature with and without internal resistance

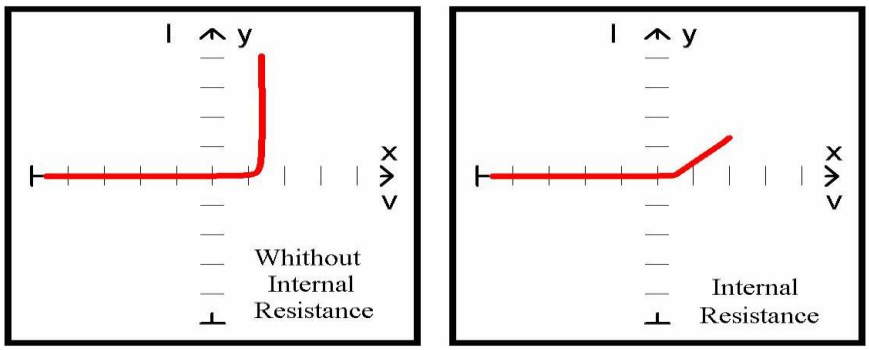


Figure 3.22 shows diode signature with and without internal resistance

Note that the vertical portion of the first signature is angled indicating the presence of internal resistance. This is similar to the signature displayed when a resistance is added in series.

The vertical portion of the signature returns to normal in the second signature when the series resistor is removed from the circuit.

Leakage in a Diode

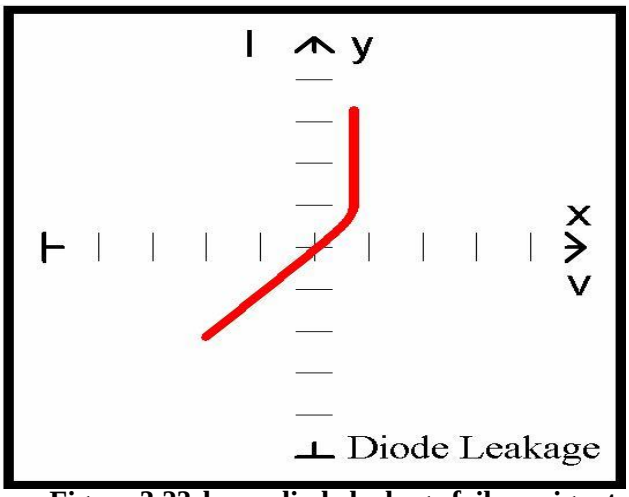


Figure 3.23shows diode leakage failure signature

Note that the horizontal portion of the signature is angled indicating the presence of diode leakage. This signature indicates that there is current flow through the device when it should be in a non-conducting state.

Understanding Composite VI-Curve and its deviations

Signatures that exhibit characteristics of several different types of components that are interconnected are called composite signatures.

Composite Diode Signatures

For example, a diode in parallel with a resistor will display a composite signature because characteristics of both a semiconductor and a resistor are shown in the signature. Composite signatures are more indicative of the signatures experienced in the "real world" of in-circuit troubleshooting.

Parallel Diode combinations

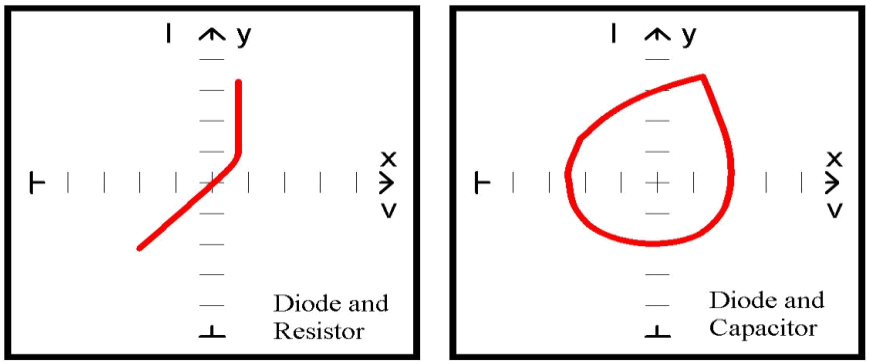


Figure 3.24 shows diode and resistor trace, diode and capacitor trace

Show in the figure 3.24 shows diode and resistor trace, diode and capacitor trace.By viewing the signature progression as components are added in parallel with the diode, you can see the effect of added resistance or capacitance on the horizontal portion of the signature. These types of signatures are common when troubleshooting components while in-circuit.

To illustrate the strength of variable ranges in the instrument to enhance or disregard various portions of a composite signature examine the following examples.

Example 2: Diode with parallel 10Kresistor at 3V
Example 3: Diode with parallel 10uF capacitor at low voltage (200mV)
Example 4: Diode with parallel 10uF capacitor at 3V

These steps illustrate how by manipulating the instrument range settings, signatures of the parallel components can be examined individually or in combination. Examples 1 and 3 are examples of "passive" testing where the test voltage is set below the 0.6V breakdown threshold of most silicon semiconductors. This essentially takes the diode out of the signature equation. Also observed is the method where by changing the resistance setting, the individual capacitor and resistor signatures can be isolated.

An example of this is shown in Example 1 signature where the resistor is clearly displayed with little effect from the parallel diode.

Zener Diodes

Standard diodes conduct when forward biased only and act as an open when reversed biased. A zener diode is designed to conduct current when both forward and reversed biased.

When forward biased it acts much like a standard diode and begins to conduct current with the forward voltage reaches approximately 0.6V. When reversed biased, they act as an open until the reverse voltage reaches the rated zener voltage at which time they begin to conduct current.

For example, a 5V zener diode will begin to conduct reverse current when the reverse bias voltage reaches 5V. Even if the voltage increases higher than 5V, the measured voltage drop across the component will remain at 5V. This is a feature of zener diodes that allows them to be used for voltage regulation. Because they conduct in both directions, expect their analog signature to display two breakdown points or "knees".Figure 3.25 shows zener diode and two zener diode trace in series shown below.

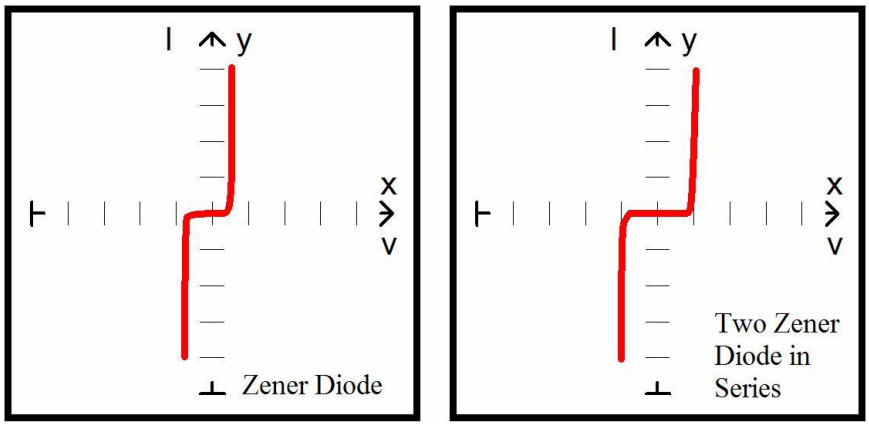


Figure 3.25 shows zener diode and two zener diode trace in series

Note that the signature displayed has two voltage breakdown points. This type of signature is commonly referred to as a "zener pattern" or "zener signature". Zener signatures are the most common type of signature encountered when testing integrated circuits (ICs). Combining two zener diodes in series essentially combines their voltage ratings.

Review for Diode Signatures

The signature of a diode will display a forward breakdown point referred to as a "knee". Zener diodes will have both a forward breakdown point and a reverse breakdown relating to its rated voltage.

Diodes have polarity with the positive connection being the anode and the negative connection being the cathode. The signature will reverse if the component or test leads are reversed.

Failures in diodes and other semiconductive components are usually resistive in nature. These faults usually require setting the ASA test instrument resistance range to a higher value such as 10K . In some cases, the failure may appear as a rounding of the "knee".

The horizontal graticule can be used to approximate the breakdown voltage of a diode. Divide the instrument voltage setting by 4 to determine the horizontal volts per division.

Testing Transistors Introduction

Figure 3.26 shows transistor symbol representations shown below. A bipolar junction transistor is a three layer device of which there are two types. A PNP transistor has a layer of N type material inserted between two layers of P type material. A NPN transistor has a layer of P type material inserted between two layers of N type material. Figure below illustrates their construction and also shows the schematic symbol used to represent both PNP and NPN transistors.

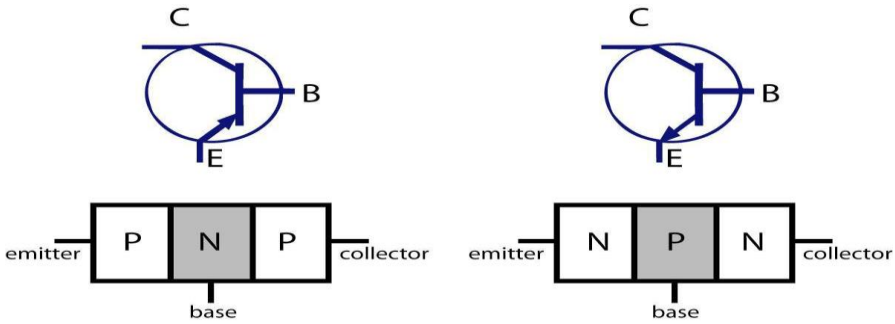


Figure 3.26 shows transistor symbol representations

In order to better understand the nature of transistor signatures we can model these devices in the terms of equivalent diode circuits shown in figure below. This diagram shows the collector to base junction appears as a simple diode signature and the base to emitter junction appears as a zener diode signature. These signatures should be familiar to you from the previous section on diodes. Figure 3.27 shows transistor equivalent diode model shown below.

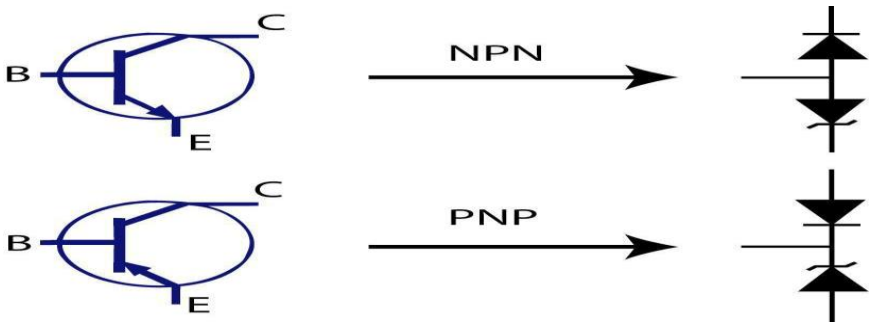


Figure 3.27 shows transistor equivalent diode model

PNP and NPN Transistor Signatures

The following examples illustrate how NPN and PNP transistor signatures are displayed. Figure 3.26 shows PNP transistor signature & figure 3.27 shows NPN transistor signature shown below.

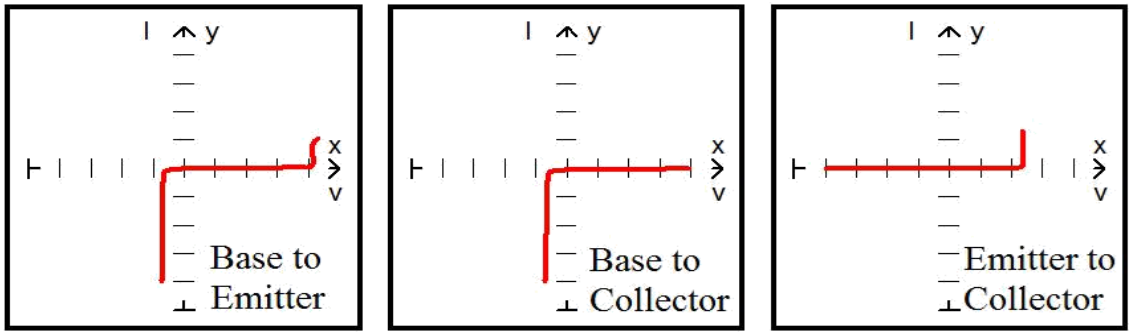


Figure 3.26shows PNP transistor signature

Note that the signatures displayed reflect the equivalent PNP circuit shown in figure. Also note that the reverse breakdown point shown on the emitter to collector signature is the close to the reverse breakdown for the base to emitter signature.

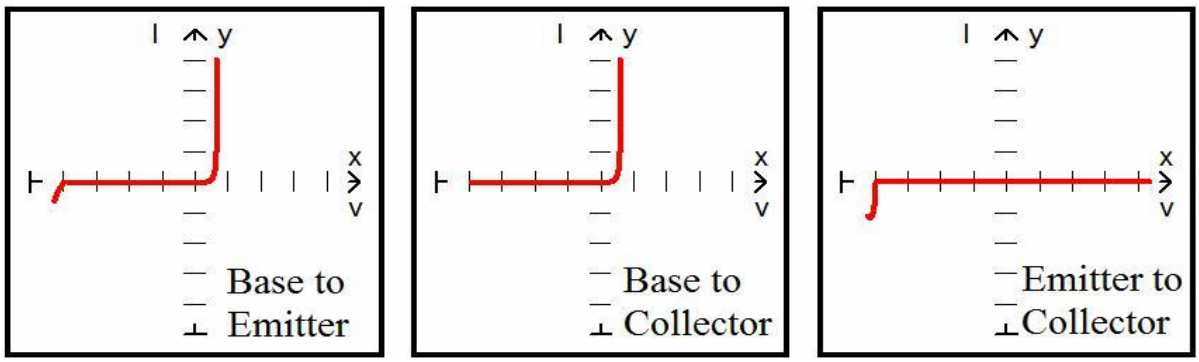


Figure 3.27 shows NPN transistor signature

Note that the signatures displayed reflect the equivalent NPN circuit shown in figure and are opposite in polarity when compared to the PNP signatures. Also note that the reverse breakdown point shown on the emitter to collector signature is close to the reverse breakdown for the base to emitter signature.

Review for Transistor Signatures

The signatures of a transistor will display a forward breakdown point and a reverse breakdown point.

Transistors have polarity and the signatures will reverse if the component or test leads are reversed. NPN and PNP transistors will display signatures that exhibit reversed polarity when compared to each other.

Failures in transistors and other semiconductor components are usually resistive in nature. These faults usually require setting the ASA test instrument resistance range to a higher value such as 10K. In some cases, the failure may appear as a rounding of the “knee”.

ASA test instruments can be used to determine transistor type (bipolar, Darlington, etc.), polarity (PNP or NPN), or pin configuration (base, emitter, collector) and also be used as a basic curve tracer for matching transistor pairs.

The flexibility of the ASA test instrument ranges allows for enhancing and disregarding parts of a composite signature for components being tested in-circuit.

Testing the Operation of Switching Devices Introduction

Switches are electrical devices that either stop or allow current flow within a circuit. In the world of electronics, “switching” is often referred to as a basic function that can be performed by a variety of devices. All of these devices are similar in that they are either on or off. They are different because of the way they are turned on or off.

Switching devices come in many different configurations from simple mechanical switches such as relays to semiconductive devices such as optocouplers or SCRs. Because of this variety of switching devices, each type is tested in a unique way. Using ASA to test switching devices allows us to ignore the differences and concentrate more on the switching function itself. This section will explore how different switching devices are tested and also how the DC voltage source or Pulse Generator can be used to gate these devices.

Testing Switch Devices with the DC Voltage Source or Pulse Generator

Many switching devices are voltage controlled. Devices such as relays respond to a changing voltage to turn the switch on and off. Semiconductive switches such as SCRs and TRIACs respond to voltages above a certain level to activate the switching function. All of these types of devices can be tested dynamically using the DC Voltage Source, Pulse Generator or an external power supply to apply a controlled voltage that will activate the switching function.

Testing a Silicon Controlled Rectifier (SCR)

Figure 3.28 shows SCR Signature without and with applied voltage shown below.

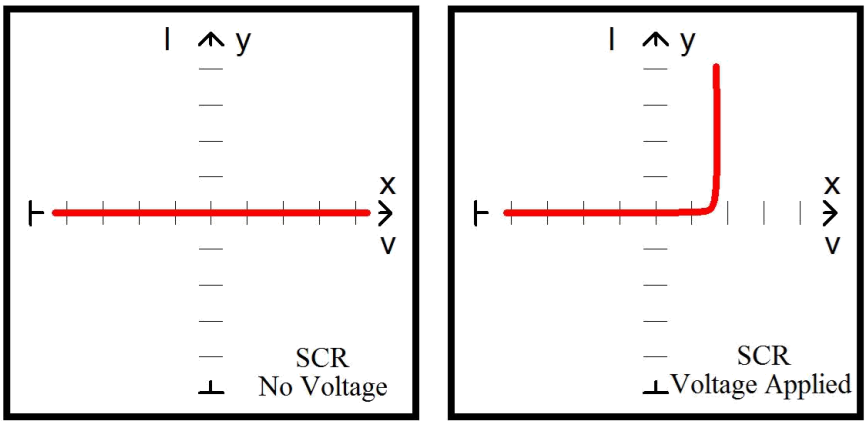


Figure 3.28 shows SCR Signature without and with applied voltage

Silicon controlled rectifiers are essentially a voltage controlled diode. If the gate is at the same voltage level as the cathode then the SCR acts as an open. When the voltage level applied to gate is more positive than the cathode (typically at 0.6V), current flows between the anode and cathode.

Testing a TRIAC is very similar to the SCR test with the exception that you see the cathode to anode signature bias in two directions (similar to a zener diode). This because a TRIAC is a bidirectional switched device.

Component Identification of Ageing Effect with VI Curve Trace

The PCBs are expected to give trouble free performance over their expected life duration. A factor like MTBF (mean time between failures) is calculated for a product based on the reliability factors of its constituents. This helps in planning a replacement or expected maintenance involving purchase of spares for the product.

Input and Output Characteristics of Digital Integrated Circuits

Even when you test a digital IC using ASA you are actually displaying the analog behavior of the input protection circuitry. Many failures in digital devices in service are due to damage in the input/output region of the device (e.g. caused by lightning strikes, etc. in telecommunication equipment). Damage to the I/O region of a digital device will be easily revealed using ASA. Consider the typical digital IC input protection circuit shown below The positive and negative excursions of the Fault Locator drive voltage will cause both diodes to conduct so the signature will appear as below. Suppose a large transient severely damages the protection diodes so that the circuit appears as a simple resistor; the signature is dramatically altered and very easy to see - the device is an obvious candidate for replacement ASA is a widely used fault finding technique, independent of the PCB technology employed. It's safe, simple to use and highly effective. If you have a trouble-shooting technique you would like to share with other Polar fault locator users please fax or email Polar Instruments on the number below.

Good versus Suspect Interpretation Comparison

In most cases, analog signature analysis is used for comparison troubleshooting. This means that the signatures of a good printed circuit assembly (PCA) are compared to those of a suspect PCA. Signature differences can indicate a potential problem.

Unit 3

REVIEW QUESTIONS

PART A (2 MARK)

1.What is Boundary Scan?
2.What is the use of function testing?
3.what are the registers associated with boundary scan?
4.Define Test Access Port?.
5.write the application of Boundary Scan Test applications
6.Explain Serial Vector Format Shorty.
7.Write the any 4 commands in SVF.
8.what are the Limitations of Boundary Scan Technology
9. Define JTAG.
10.what is memory testing.

PART B (3 MARKS)

1.Write the BSDL file is a data that provides specifications
2.Define Serial Vector Format & any one example.
3.Write any 6 DR scan operations.
4.Draw the non BS device in Integration of Boundary Scan Test methodology in Automated Testing.
5.Explain JTAG interface signals to support the operation of boundary scan.
6.Write the Boundary Scan Instructions.
7.Define Testing non-JTAG OR Non-BS Devices & any one example.
8.Classification of BSDL files elements.

PART –C (10 MARKS)

1. Explain the difference between NAND and NOR flash and list out its applications.
2. What are the limitations of clip-on testing as compared to whole board testing?
3. Explain the behavior of diode under forward conditions with neat sketches.
4. Explain the resistor construction details and V.I. characteristics, state the field of application.
5. Draw the circuit symbol of Transistors, JFET and MOSFETS and mark their terminals.

UNIT –IV
Boundary Scan Testing Methods and Technology

What is Boundary Scan – a New Test Technology

From the day the integrated chips developed, there has been the necessity to check their functions. In the case of digital circuitries, a test is quite simple: all possible test vectors are applied in succession, and then the circuitries' reactions at the outputs (actual value) are compared to the expected patterns (nominal value). If there are no differences the circuitry is correct.

The number of test vectors is manageable for a simple AND gate with two inputs. According to Moore and Mc Cluskey the following formula calculates this number: Q=2 (x+y)

Q = minimum number of test vectors

x = number of inputs

y = number of storage elements (for sequential circuits)

Because an AND gate normally doesn't have storage elements, there are only four necessary test vectors – which is a manageable number. If this calculation is done for a circuitry with assumed 25 inputs and 50 storage elements, the problems in chip developments the engineers faced in the 1970s become obvious.

In the early 1970s, IBM gave birth to a path breaking idea: the invention of the first "Level Sensitive Scan Design (LSSD)" method. For this purpose, existing storage elements in a chip are extended in their functions. They get four additional connectors: an input (IN), an output (OUT) and two clocks (A and B); With these additional resources it is also possible to access the storage elements' inputs and outputs.

In the beginning of the 1980s, the problem of "increasing complexity of the PCBs with higher packaging density" at board level was tackled. The "Joint European Test Action Group" (JETAG), founded in 1985, was one of the first institutions that dealt with the topic. In those days, this group consisted of test engineers from the big European integrated chip manufacturers.

In 1986, additional North American companies joined, and the group was renamed "Joint Test Action Group (JTAG). JTAG engineered a methodology, which came close to the LSSD method developed by Ed Eichel berger. It also defines storage elements within a chip which are connected in a shift chain. The only difference was that the storage elements were additionally placed at the component's peripheral, "at boundaries".

1

For this reason, the developed method was named Boundary Scan. It was standardized as 1149.1 “Standard Test Access Port and Boundary Scan Architecture” by the “Institute of Electrical and Electronics Engineers (IEEE)” in 1990.

Let’s begin with a bit of history on why Boundary Scan Test came about.

Way back in the late 80s and early 90s, several developments in printed circuit board (PCB) technology incorporated in design and these new techniques came to gether to make board test more difficult, namely:

- The surface mounted devices (SMD)
- The use of multilayer circuit boards
- The miniaturization of electronic components
- The lack of test points on printed circuit boards, due to these issues

As a result of these changes, traditional in-circuit test (ICT) tools, so-called bed of nails testers were no longer able to fully test Printed circuit boards (PCBs).

To address this vexing problem, a consortium of companies–known as the Joint Test Access Group or JTAG.

The JTAG group was originally know as JETAG (Joint European Test Access Group) since it members were European companies when it was formed. The name was then changed to JTAG shortly after inception when US and Asian companies joined the effort.

After the ratification of the core boundary scan specification by the Std known as IEEE1149.1, industry continued to advance the technology by proposing enhancements.

The suggestions led to new versions and new standards that helped to keep pace with advancing IC and board technology for testing . The major standards are outlined below .

IEEE Standards

IEEE 1149.1 JTAG

This Std describes the core functionality for JTAG Boundary Scan as it applies to digital circuit testing this tutorial will describe the standard in detail. The number , .1 in specs supports static digital testing via Boundary Scan.

IEEE 1149.6 JTAG

This Std adds support AC coupled nodes like LVDS.

It allows you to test by sending pulses back and forth between devices using cells that support LVDS signals.

In order to do this sort of testing you are required to have a cell with 1149.6 support at each end of the bus design.

IEEE 1149.7 JTAG

IEEE 1149.7 is a super-set of IEEE 1149.1 It provides increased capability and decreases the pin count on the TAP controller from five pins down to two. (1149.7 only requires the TMS and TCK signals)

This scheme reduces the number of pins needed on the IC for control and test. Also this Std, direct access to devices 1149.7 supports a star configuration.

The IEEE 1149.1 Standard JTAG specification is available directly from IEEE: http://standards.ieee.org/

Who uses Boundary Scan?

Design Engineers

Design engineers use Boundary Scan for hardware bring up, validation and prototype debug. They may also use the tools to generate tests for use in design and manufacturing of electronic products.

Test Engineers

Test Engineers use Boundary Scan to create tests for manufacturing and board repair.

Production Staff

Manufacturing staff use Boundary Scan tools to test and debug circuit boards in a production environment.

JTAG Boundary Scan Tools

The term JTAG is used loosely to describe board test and embedded debug interfaces based on the above mentioned Std. specification. Terms typically used include:

Production test tools

Manufacturing staff use Boundary Scan tools to test and debug printed circuit boards in a production environment and for testing during production completion.

JTAG ICE Debuggers

The same JTAG interface that is used for Boundary Scan test can also be used by tools that enable debugging of embedded firmware on microprocessors and microcontrollers or any other programmable devices on board .

JTAG Device Programmers

Many popular programmable devices use the JTAG interface as well.

Applications of Boundary SCAN

Boundary SCAN Test supports chip, board and system level tests.

This section introduces the types of tests possible with Boundary SCAN technology.

Prototype Debug and Bring Up

When bringing up and debugging a new hardware designed, Boundary Scan comes to the rescue in several important ways:

Testing Partially Populated hardware

Initially , not all devices may be fitted. With Boundary Scan, it is desired to have only good power, ground and at least one part on the JTAG Chain to begin testing. You should be able to ID the part on the chain and then test for opens and shorts for any board area that is included in the circuitry by this device.

Initializing and Programming Devices

It will be easy to do initial device programming. For example, if the device on the chain is a microprocessor or DSP, most likely you will have access to RAM and FLASH memory via the address, data and control bus. This can allow you to ID and Program and test these so-called (cluster, or non-JTAG) devices. FPGAs, CPLDs (XILINX and ALTERA, for example) may also fall into this category.

Finding Assembly Defects

Prototypes are often rushed through assembly in order to make engineering deadlines. As a result, assembly and manufacturing problems will exist. Boundary Scan is perfect for testing for common problems like unfitted or wrongly mounted devices, solder issues (cold or hot Joints), as well as open, shorts, stuck at and device functional failures.

Hardware Validation

As initial firmware or diagnostics are written for a new hardware, Boundary Scan can be used to rule out “bad hardware” by providing a “golden” test that validates that a board is good. With this type of testing, the debug will be easy and release of firmware versions can be checked and controlled at our end.

Production Test

Manufacturing/Production test is the clearly sweet spot for Boundary Scan technology. It was designed to complement existing test methods and to overcome the problems of evolving board technology.

Device Programming

As previously mentioned devices can be programmed via JTAG boundary Scan. CPLD and FPGA images can be loaded as well as boot loader code and diagnostics software. Boundary Scan also makes a perfect tool for production test repair stations that allow test engineers to troubleshoot complex and expensive boards in the “bone yard”.

Functional Testing

JTAG can be used to do basic functional testing of your hardware, including DDR RAM, FLASH based devices and others. Since JTAG is a clocked serial interface, some devices will be too fast to be tested at full speed but can be tested for general operation.

Processor-Controlled JTAG Test

Processor controlled test uses the same JTAG hardware interface and protocol as Boundary Scan testing, the difference is that the microprocessor on the board under test is used to implement the

tests. Processor controlled test supports access to devices not accessible to JTAG and also supports "at-speed" testing.

4.1 Boundary SCAN Theory of Operation

The process of boundary scan can be most easily understood with reference to the schematic diagram shown in figure 4.1.

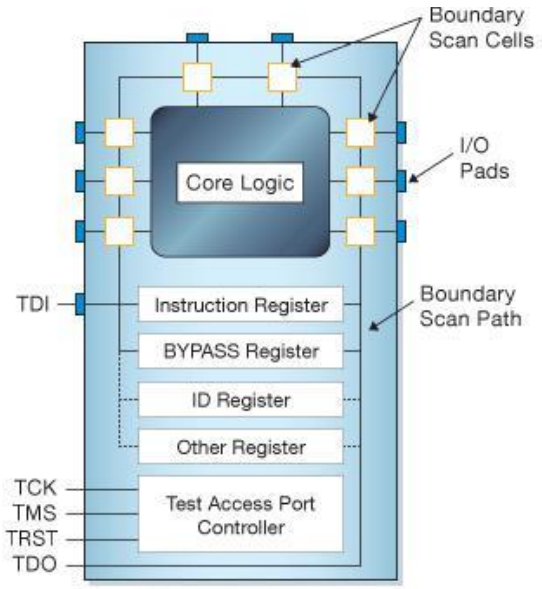


Figure 4.1 Schematic Diagram of a boundary scan architecture

4.2 Boundary Scan Cell

The Boundary Scan cell is the essential element of the Boundary Scan Test methodology. All described constructs' functions are only for the correct control of the respective Boundary Scan cells.

The Boundary Scan cell is the ingenious opportunity to control a component pin disengaged from its normal functionality, i.e. to drive or measure a particular level. For this purpose, the Boundary Scan cell is situated between the component's core logic and peripheral (output driver, input driver).

Because of the functionality similar to the physical contact nails of the In Circuit test technology, which implement access to the test points on a board, the Boundary Scan cells are also called "electronic nails". Figure 4.2 Comparison of the test methods ICT and Boundary Scan shown below.

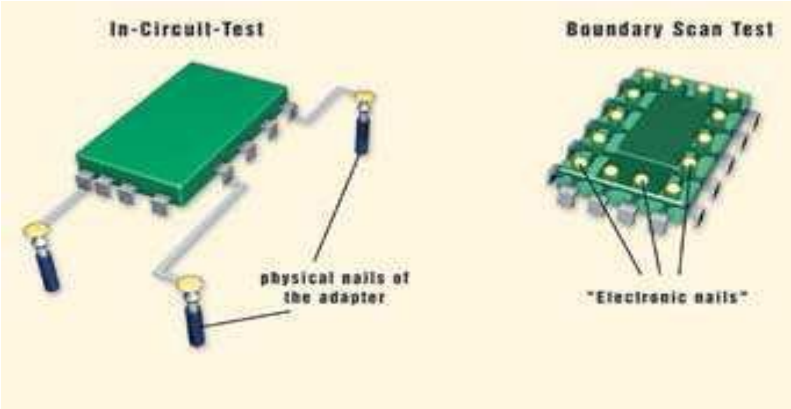


Figure 4.2 Comparison of the test methods ICT and Boundary Scan

A Boundary Scan cell's internal architecture can be highly different. In its version from 2001, the IEEE Std. 1149.1 describes ten different cell types (BC_1 to BC_10). The cell may have individual structures, whereby the arrangements are very often very similar.

All the signals between the device's core logic and the 'pins' are intercepted by a serial scan path known as the Boundary Scan Register (BSR). In normal operation these boundary scan cells are invisible. However, in test mode the cells can be used to set and/or read values: in external mode these will be the values of the 'pins'; in 'internal' mode these will the values of the core logic.

Interface Signals

The JTAG interface, collectively known as a Test Access Port, or TAP, uses the following signals to support the operation of boundary scan.

TCK – the TCK or 'test clock' synchronizes the internal state machine operations

TMS – the TMS or 'test mode state' is sampled at the rising edge of TCK to determine the next state. TDI – the TDI or 'test data in' represents the data shifted into the device's test or programming logic. It is sampled at the rising edge of TCK when the internal state machine is in the correct state.

TDO – the TDO or 'test data out' represents the data shifted out of the device's test or programming logic and is valid on the falling edge of TCK when the internal state machine is in the correct state. TRST – the TRST or 'test reset' is an optional pin which, when available, can reset the TAP controller's state machine.

Registers

There are two types of registers associated with boundary scan. Each compliant device has one instruction register and two or more data registers.

- Instruction Register – the instruction register holds the current instruction. Its content is used by the TAP controller to decide what to do with signals that are received. Most commonly, the content of the instruction register will define to which of the data registers signals should be passed.

- Data Registers – there are three primary data registers, the Boundary Scan Register (BSR), the BYPASS register and the IDCODES register. Other data registers may be present, but they are not required as part of the JTAG standard.

- BSR this is the main testing data register. It is used to move data to and from the I/O pins of a device.

- BYPASS – this is a single-bit register that passes information from TDI to TDO. It allows other devices in a circuit to be tested with minimal overhead.

- IDCODES – this register contains the ID code and revision number for the device. This information allows the device to be linked to its Boundary Scan Description Language (BSDL) file. The file contains details of the Boundary Scan configuration for the device.

Test Access Port (TAP) Controller

The TAP controller, a state machine whose transitions are controlled by the TMS signal, controls the behavior of the JTAG system incorporated in the device. Figure 4.3 below, shows the state-transition diagram.

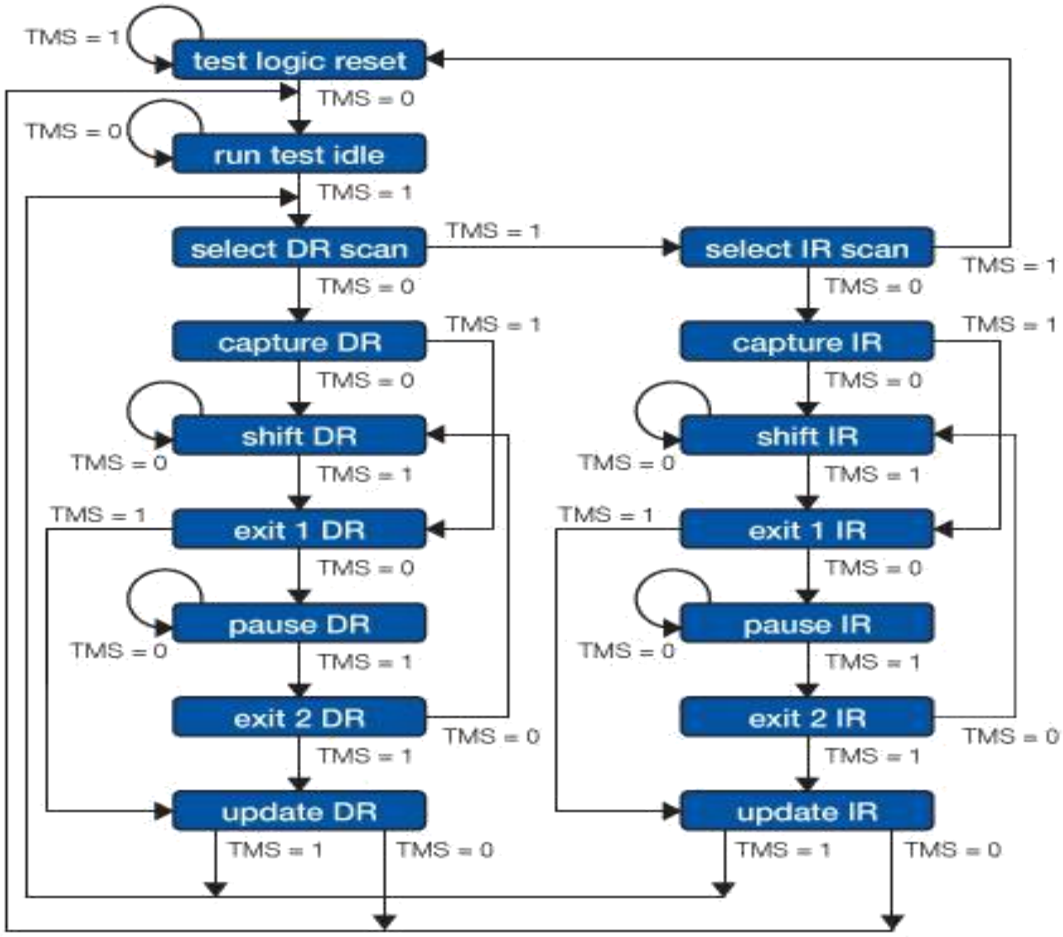


Figure 4.3. TAP controller State Machine Process Diagram

All states have two exits, so all transitions can be controlled by the single TMS signal sampled on TCK. The two main paths allow for setting or retrieving information from either a data register or the instruction register of the device. The data register operated on (e.g. BSR, IDCODES, BYPASS) depends on the value loaded into the instruction register.

For more detail on each state, refer to the IEEE 1149.1 Standard JTAG document for further reading.

4.3 Boundary Scan Instructions

The IEEE 1149.1 standard defines a set of instructions that must be available for a device to be considered compliant. These instructions are:

BYPASS – the BYPASS instruction causes the TDI and TDO lines to be connected via a single-bit pass-through register (the BYPASS register). This instruction allows the testing of other devices in the JTAG chain without any unnecessary overhead.

EXTEST – the EXTEST instruction causes the TDI and TDO to be connected to the Boundary Scan Register (BSR). The device's pin states are sampled with the 'capture dr' JTAG state and new values are shifted into the BSR with the 'shift dr' state; these values are then applied to the pins of the device using the 'update dr' state.

SAMPLE/PRELOAD – the SAMPLE/PRELOAD instruction causes the TDI and TDO to be connected to the BSR. However, the device is left in its normal functional mode. During this instruction, the BSR can be accessed by a data scan operation to take a sample of the functional data entering and leaving the device. The instruction is also used to preload test data into the BSR prior to loading an EXTEST instruction.

Other commonly available instructions include:

IDCODE – the IDCODE instruction causes the TDI and TDO to be connected to the IDCODE register.

INTEST – the INTEST instruction causes the TDI and TDO lines to be connected to the Boundary Scan Register (BSR). While the EXTEST instruction allows the user to set and read pin states, the INTEST instruction relates to the core-logic signals of a device.

BSDL Files

Boundary Scan Description Language (BSDL) is a subset of VHDL that is used to describe how JTAG (IEEE 1149.1) is implemented in a particular device.

For a device to be JTAG compliant, it must have an associated BSDL file.

These files are often available for download from original equipment manufacturers'(OEM) of devices websites.

JTAG systems such as XJTAG use the information contained in a BSDL file to work out how to access a device in the JTAG chain.

The BSDL file is a data that provides specifications about:

- available test bus signals (particularly information about the existence of an optional /TRST signal and maximum TCK frequency, up to which the component can be operated)
- possible “compliance” pins
- instruction register (available instructions incl. bit code; instruction register length)
- data register (available data register incl. Possible predefined values, e.g. IDCODE of the chip)
- Boundary Scan cell structure (number, type, function, assignment to IC pin)

BSDL files contain the following elements:

- Entity Description: Statements naming the device or a section of its functionality.
- Generic Parameter: A value such as a package type. The value may come from outside the current entity.
- Port Description: Describes the nature of the pins on the device (input, output, bidirectional, linkage).

- Use Statements: References external definitions (such as IEEE 1149.1).
- Pin Mapping(s): Maps logical signals in the device to physical pins.
- Scan Port Identification: Defines the pins used on the device to access the JTAG capabilities (TDI, TDO, etc - the Test Access Port).
- Instruction Register Description: The signals used for accessing JTAG device modes.
- Register Access Description: Which register is placed between TDI and TDO for each JTAG instruction.
- Boundary Register Description: List of the boundary scan cells and their functionality.

Boundary Scan Test – Interconnect Test

The boundary-scan test architecture provides a means to test interconnects between integrated circuits on a board without using physical test probes. It uses the boundary-scan cell that includes a multiplexer and latches, provided to each pin on the device. Boundary-scan cells in a device can capture data from pin or core logic signals, or force data onto pins called the scan path or scan chain.. Captured data is serially shifted out and externally compared to the expected results. Forced test data is serially shifted into the boundary-scan cells. All of this is controlled from a serial data path

Using Scan Path

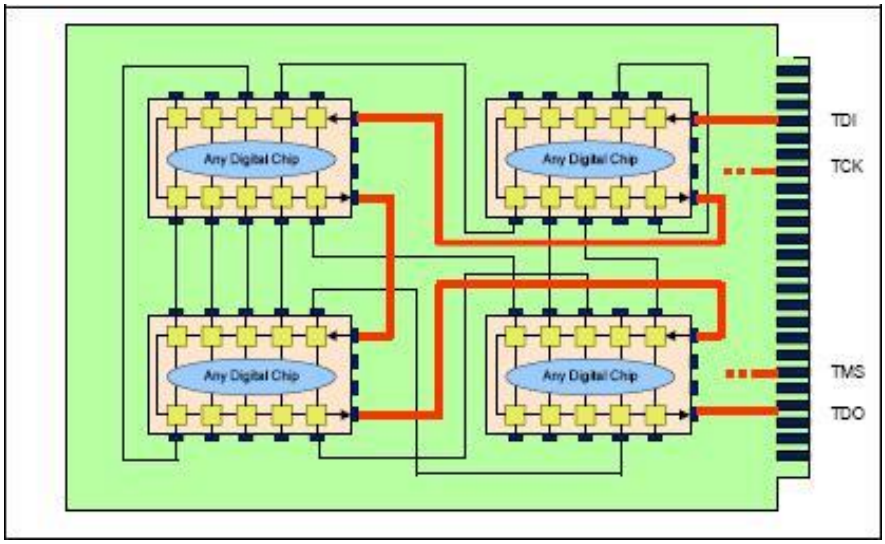


Fig 4.4 Boundary Scan Device interconnection in a board

At the device level, the boundary-scan elements contribute nothing to the functionality of the internal logic. In fact, the boundary-scan path is independent of the function of the device. The value of the scan path is at the board level.

The figure 4.4 shows a board containing four boundary-scan devices. Notice that there is an edge-connector input called TDI connected to the TDI of the first device. TDO from the first device is permanently connected to TDI of the second device, and so on, creating a global scan path terminating at the edge connector output called TDO. TCK is connected in parallel to each device TCK input. TMS is connected in parallel to each device TMS input.

What the tester sees from the edge connector is simply the concatenation of the various boundary-scan registers – that is, a single register that provides access to all device outputs now considered to be **drivers**

(sometimes called a **transmitter**) onto an interconnect and all device inputs now considered to be **sensors** (sometimes called a **receiver**) from the interconnect.

Testing non-JTAG OR Non-BS Devices

Including tests for devices that do not have JTAG is very important in achieving the highest possible test coverage while testing a board . Connection testing is very good at checking the physical, manufacturing integrity of a lot of a circuit; however there are some faults that it cannot detect.

For example it is only possible for a connection test to check that a pin is not open circuit if communication can be seen between that pin and another pin in the design. This means that a connection test can only check for open circuit faults on pins that have JTAG capability and are on nets with at least one other JTAG enabled pin.

However, by using the connections between the accessible pins/busses on devices in the JTAG chain to drive and monitor signals on non-JTAG devices, it is possible to test other pins for open circuit faults. Exercising the functionality of non-JTAG devices in this way means that open circuit faults can be found on both the peripheral device and the JTAG enabled device.

Short circuit faults and stuck-at faults can also be detected in this way; however connection testing is a more effective tool for finding these types of fault.

This form of testing can be used on individual devices but is often applied to a group, or cluster, of non-JTAG devices in the circuit. Some JTAG tool vendors refer to this type of testing as cluster testing.

Memory Testing

One variant of this method is memory testing. A sequence of JTAG test signals is created to manipulate the address and data busses of a memory device so as to write information into memory, then a second set of test signals is created to read this information back. This can apply to SRAM, SDRAM, Flash memory or any variant.

Other Possibilities

If any non-JTAG device is connected to a JTAG device, its functionality and/or connections can be tested to some extent.

For example :

- External Connectors
- Video chips
- IIC devices
- Ethernet Controllers
- LEDs
- Switches

Boundary Scan Test methodology is in its third decade as an Industry Standard. It is yet to go far from the technology point of view in its future applications. Basically the following are the applications of Boundary Scan Test in device level in a board .

- Inner Connection testing and In System Programming (ISP) are the two applications most commonly associated with JTAG. However the technology has far more to offer.

- Non-JTAG-compliant sections of a circuit can be tested by using the interconnecting nets between the devices in the JTAG chain and other devices in the circuit.
- XJTAG has been designed to unlock the potential of JTAG, with its development system of software products and JTAG controller hardware (available with a range of connectivity options).

Serial Vector Format (SVF)

The SVF, or Serial Vector Format, was developed as a vendor-independent way of representing JTAG test patterns in ASCII (text) files. The BSDL text file is been converted to test vector file of this type before driving in to IC Test Pins .

SVF files consist of a list of statements and/or comments.

The following is an example statement:

SDR 64 TDI(0) TDO(0123456789ABCDEF) MASK(0FFFFFFFFFFFFFFF);

This will scan 64 bits out from the data registers of devices in the JTAG chain, scanning in 64 zeros and expecting to read 0x0123456789ABCDEF out, with the mask of 0FFFFFFFFFFFFFFF indicating that the first 4 bits are not significant, but all the rest are.

Headers and trailers can also be set up, enabling you to target a specific device, or set of devices, in the JTAG chain without having to consider the other devices in the chain at each step.

The complete list of SVF commands is as follows.

ENDDR	Specifies default end state for DR scan operations.
ENDIR	Specifies default end state for IR scan operations.
FREQUENCY	Specifies maximum test clock frequency for IEEE 1149.1 bus operations.
HDR	(Header Data Register) Specifies a header pattern that is prepended to the beginning of subsequent DR scan operations.
HIR	(Header Instruction Register) Specifies a header pattern that is prepended to the beginning of subsequent IR scan operations.
PIO	(Parallel Input/Output) Specifies a parallel test pattern.
PIOMAP	(Parallel Input/Output Map) Maps PIO column positions to a logical pin.
RUNTEST	Forces the IEEE 1149.1 bus to a run state for a specified number of clocks or a specified time period.
SDR	(Scan Data Register) Performs an IEEE 1149.1 Data Register scan.
SIR	(Scan Instruction Register) Performs an IEEE 1149.1 Instruction Register scan.
STATE	Forces the IEEE 1149.1 bus to a specified stable state.
TDR	(Trailer Data Register) Specifies a trailer pattern that is appended to the end of subsequent DR scan operations.
TIR	(Trailer Instruction Register) Specifies a trailer pattern that is appended to the end of subsequent IR scan operations.
TRST	(Test Reset) Controls the optional Test Reset line.

Integration of Boundary Scan Test methodology in Automated Testing

If all the devices used in a PCB design are BS compliant then it is quite easy to use JTAG with a simple BS controller and software to test interconnection between all device pins, functionality of devices (if provided in BSDL file), but in practice there are many devices that are not boundary scan

compliant. For example, to check the interconnection between device pins and edge connector you need physical test pins to connect with the edge connector and an integrated BS Interconnect software. This Physical test pin can be an I/O pin of another BS Device inside your Tester systems or it can be an ATE test pin.

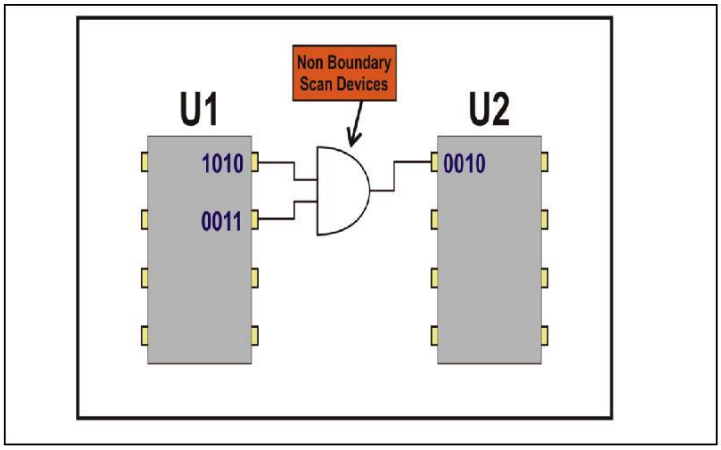
If it is another I/O pin of a BS device inside your system and in the same chain, it can detect interconnections to the edge connector but the test will be slow as there will be too many devices on the chain.

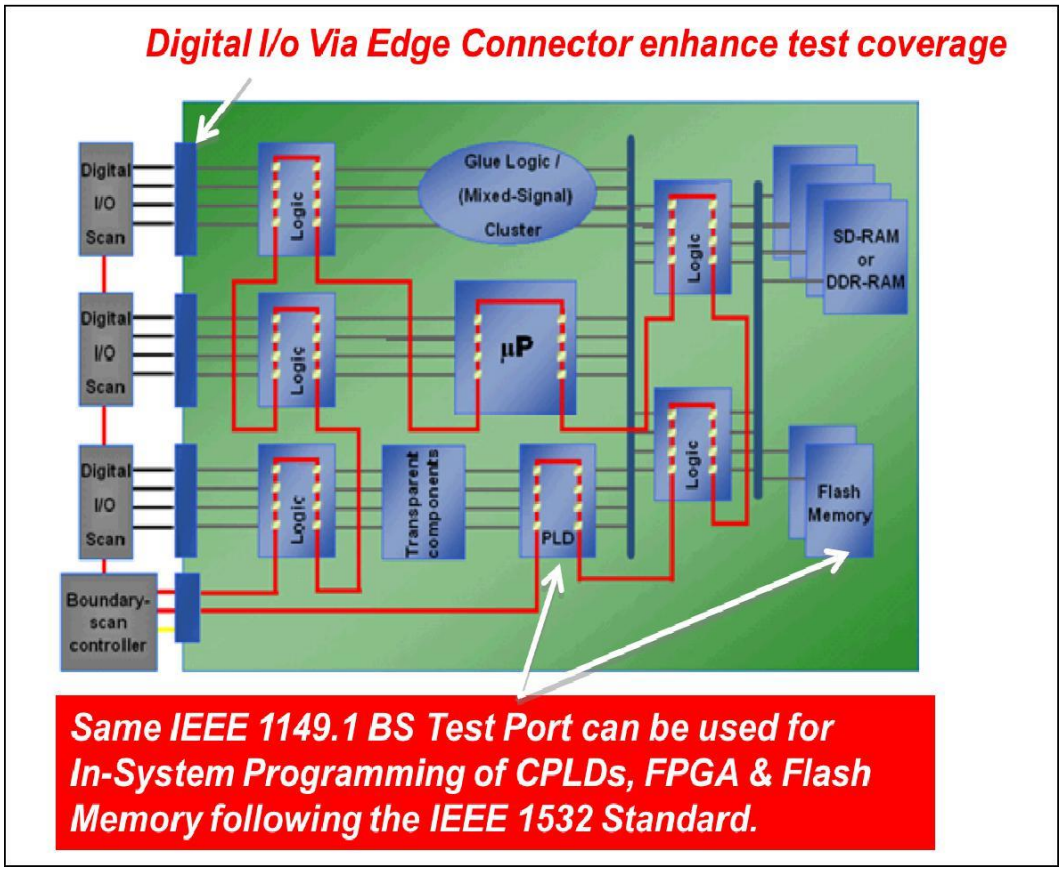
On the other hand if the ATE test channels are used to connect the edge connector and with an integrated BS test system, the interconnections between BS devices and edge connector can be tested at a faster rate.

Also testing a non BS device that is connected between two BS devices can be done through BS software by driving the U1 test pins, which are inputs to the Non BS device as shown below and reading the output of the non BS device through U2, which is a BS device.

4.5 Digital Input Output Edge Connector Enhance Test Coverage

Figure 4.5 Integration of Boundary Scan Test methodology in Automated Testing





Using Boundary Scan Utility Software, It is possible to toggle a particular pin of a device in loop mode and it can be physically to also verified by probing the signal of desired.

These utilities provide debug tool for the developers or repair men to diagnose faults in complex PCBs.

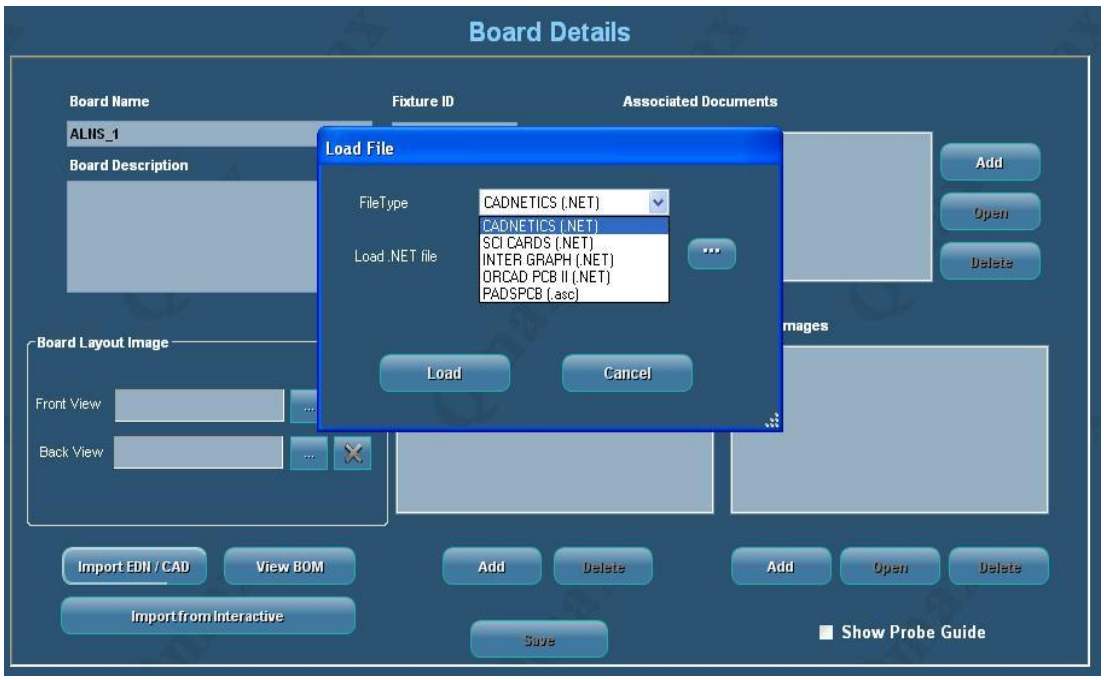


Figure 4.6 debug tool for the developers or repair men to diagnose faults in complex PCBs.

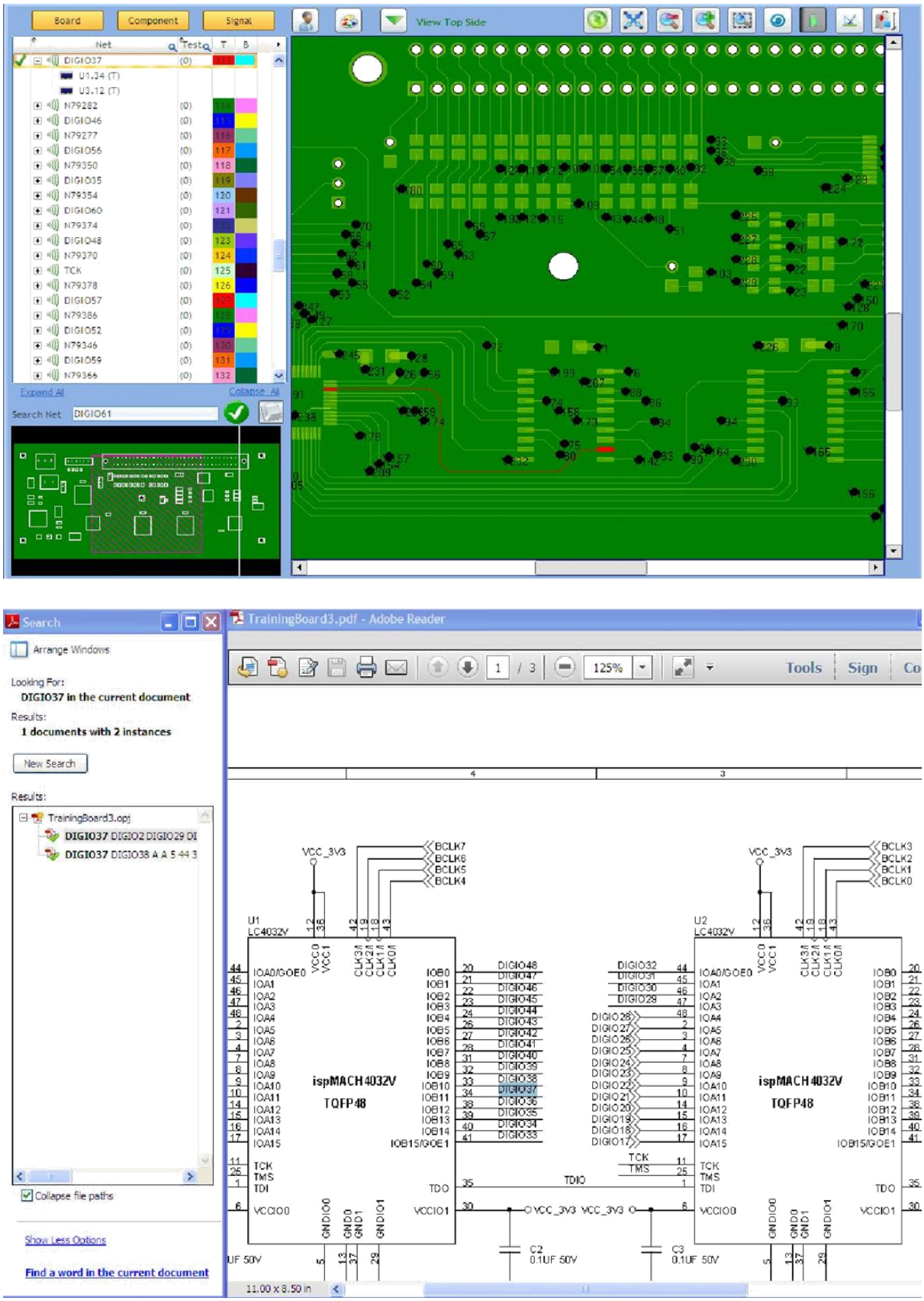


Figure 4.7 simulate output.

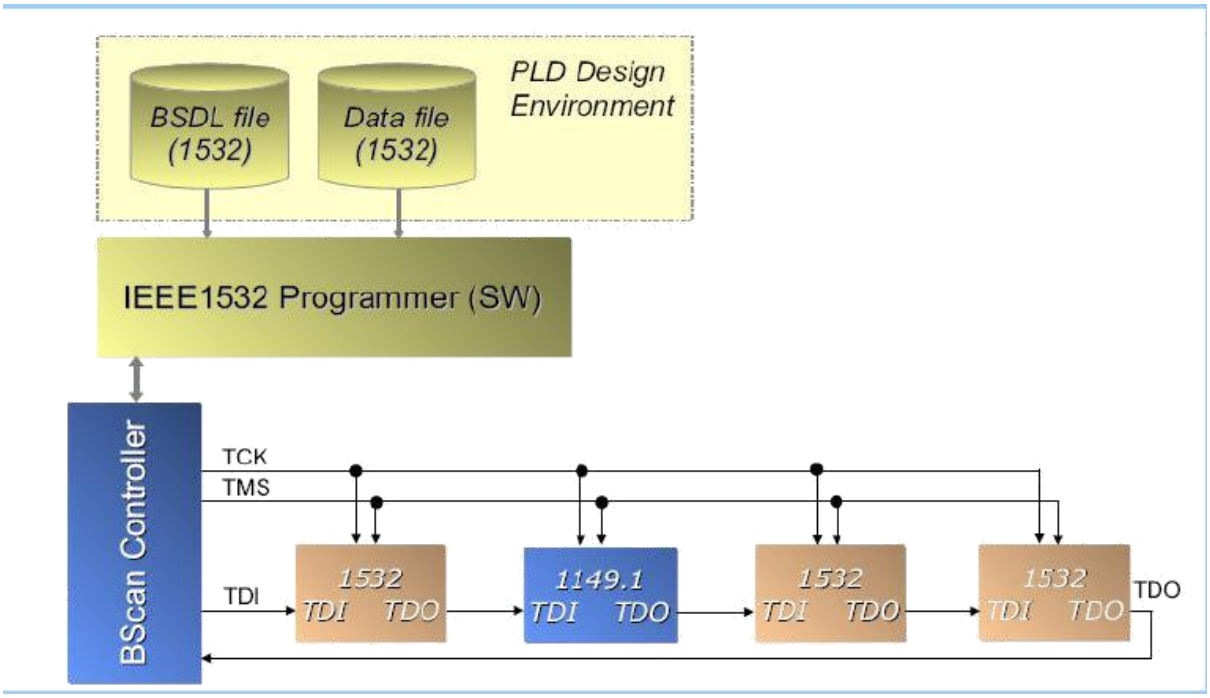


Figure 4.7 PLD design environment

Allows configuring, programming, read back, verify, erase of programmable devices after it has been assembled on the PCB. Show in the figure 4.7 PLD design environment

—

— Allows concurrent programming which improve significantly the time it cost to program a board with several programming devices. Eliminate the need to have multiple vendor programming tools to program their respective devices

Limitations of Boundary Scan Technology

The static, digital interconnection test compliant with IEEE Std. 1149.1 enables everything that is situated in the digital area and is not time critical. Thus, it is possible to test resistors (for availability), crystals, driver ICs, logic gates, reset ICs and even RAM ICs or Flash ICs (parallel as well as serial). For example, for the later the necessary write and read protocols are simply imitated via the Boundary Scan component pins. This is the same functionality as a functional test, but slower because of the serial Boundary Scan chain. And that's the test methodology's limitation: the maximum possible switch/measuring frequency at the IC pins. It is the result of the number of Boundary Scan cells and the "Test Clock" frequency. It doesn't matter whether the signal level of one or several pins should be changed - in each case it must be shifted through ALL the cells.

The serial bits sending in a medium sized Boundary Scan component with 500 Boundary Scan cells and a typical frequency of 10 MHz takes 50 s. However, one shift process can initiate a single signal change at the IC pin. For the opposite edge another shift process is required which results in a maximum achievable frequency of 10 kHz.

Unit-4

REVIEW QUESTIONS

PART A (2 MARK)

1. Define RVS.
2. Draw boundary scan Instruction Register.
3. Draw basic boundary scan cell.
4. With a neat sketch explain Clock Pin Termination?
5. What are Signal Pins?
6. What is mean by Test Channel in ATE?
7. Describe BSDL Entity Descriptions with example.
8. State BSDL Generic Parameters with example.
9. Explain BSDL Logic Port Description with example.
10. What is BSDL User Statements with example?
11. What are BSDL Pin mapping give examples?
12. Explain BSDL Scan Port Identification with example.
13. Explain the term BSDL Instruction Register Descriptions with example.
14. Explain BSDL Register Access Description with example.
15. Explain BSDL Boundary Register Description with example.
16. List out some common device package types.
17. What are the limitations of clip-on testing as compared to whole board testing?
18. What is time base error correction?

PART B (3 MARK)

1.List the JTAG TAP Interface signals and explain each.
2.Explain JTAG and its uses in details.
3.How is BSDL Used and why?
4.What is Net list and explain?
5.What is Node in a circuit? Explain with neat circuit diagram.

6. Draw VI Characteristics of Zenor Diode.
7. Explain Auto-Compensation technique with neat sketch.
8. Write a short note on SRAM and EEPROM.
9. Write a note on types of memory devices.
10. Briefly explain VI characteristic curve of an Ideal Resistor with neat sketch.
11. What are Sequential logic explain any two with neat diagram?
12. What are combinational logic circuit explain with neat diagram?
13. What are universal gates draw a symbol of various gates and its truth table using universal gate.
14. Explain what volatile and non-volatile memory with examples?

PART C (3 MARK)

1.Explain the difference between NAND and NOR flash and list out its applications.
2.What are the limitations of clip-on testing as compared to whole board testing?
3.Explain the behavior of diode under forward conditions with neat sketches.
4.Explain the resistor construction details and V.I. characteristics, state the field of application.
5.Draw the circuit symbol of Transistors, JFET and MOSFETS and mark their terminals.
6.Draw the basic boundary scan cell and explain it working principle.
7.What are boundary scan standard instructions? Explain any four in details.
8.Explain boundary scan TAP controller with state diagram.

Unit 5: ATE System and Test program Generation

5.1 ATE systems for Board Test

With the manual troubleshooting becoming increasingly difficult, the need of the hour is an equipment, which helps to find faulty components in the PCB being serviced, through an automated process with minimal manual inference**.**

ATE Systems are used to test a wide range of electronic devices and systems, from simple passive components to integrated circuits (ICs), Populated Printed Circuit Boards (PCBs) and Complex Electronic Systems (System Level Testers)

In Testing Populated PCBs, the entire circuit board is tested for its functionality by the ATE through the card edge connectors. Ideally, this equipment should be capable of testing any type of PCBs with Analog, Digital, CPUs or Mixed Signal devices and RF circuits. They should also be capable of isolating the faults up to the component level by using clips or probes.

Such automated test equipment offers advantages like,

- Capable of testing the PCB for Go – No Go in few minutes.
- Capable of locating faults down to component level using a software Guided Probe Back Tracking in a pre determined amount of time and thus eliminating indefinite time factor that is inherent in manual trouble-shooting.
- High accuracy in locating exact faulty components and thus avoiding replacement of good components by suspicion or trial and error methods, which is normal in manual trouble-shooting.
- Ability to diagnose any faults with a pre determined fault coverage up to near

100% The PCB test equipment could be broadly classified into two major categories;

1) Those used in the production floors capable of detecting / analyzing manufacturing defects.
2) Those used for detecting faulty components in Board recovery process in production floors or in the after sales repair environment.

Automated Test Systems are widely used in PCB testing in both production testing and PCB maintenance environments.
All proprietary ATE has some type of test management software. The functions of this software include:

- Organizing a set of test routines
- Sequencing, branching, and looping through tests
- Collecting, organizing test results and inferring test results
- Providing a consistent and easy to use user interface.

•

The hardware configuration generally provides for

- In–Circuit testing of passive components (ICT)
- In–Circuit Functional test of Active and semi-conductor devices (ICFT)
- Board functional test through edge connectors
- Integrated Boundary Scan Test
- Measurement tests like Voltage, Frequency RF parameters etc.,
- DC parametric tests for input bias current, fan-out capacity and tri-state leakage currents
- Learn and Compare Tests using analog and digital signatures.
- Use of new technologies such as IEEE's Boundary Scan Test
- Use of optical and thermal imaging technologies.

An ATE employs most of the techniques discussed above for an effective trouble shooting solution. Since ATE is the mainstay of Test Engineering, it will be discussed later in more detailed manner.

Auxiliary Instrumentation

ATEs generally covers all PCB test requirements. Still, there may be some requirement for precision driving / measurement of voltage, current, audio or RF frequencies above the specifications of standard specs of the ATE system. The need for using such special instruments in combination with an ATE calls for special data transfer protocols. We will see some of them like GPIB, LXI, PXI, USB etc., A Test Engineer will definitely need to interface external instrumentation and use them in conjunction with ATE routines to test certain parameters of the board under test.

GPIB

The GPIB (**G**eneral **P**urpose **I**nterface **B**us) bus was developed to connect and control programmable instruments, and to provide a standard interface for communication between instruments from different sources.

It is also known as IEEE-488 or HPIB (Hewlet Packard Interface). To use the GPIB you need a GPIB adaptor card in your computer and a GPIB lead. Fourteen devices can be connected to one GPIB and data can be transferred at up to 200,000 bytes per second. However, devices shouldn't be more than a couple of meters from the controlling computer.

Almost any instrument can be used with the IEEE-488 specification, because it says nothing about the function of the instrument itself, or about the form of the instrument's data. Instead the specification defines a separate component, the interface, that can be added to the instrument. The signals passing into the interface from the IEEE-488 bus and from the instrument are defined in the standard. The instrument does not have complete control over the interface. Often the bus controller tells the interface what to do. The Active Controller performs the bus control functions for all the bus instruments.

Technically, the GPIB uses a 16-line parallel connection which is strictly defined for its mechanical and electrical properties. The 16 lines are divided into 8 data lines, 3 handshake lines to synchronize the transfer and 5 management lines to control use of the bus.

At any time, there must be one device on the bus which is the controller. This device issues commands to other devices, and in most of the systems, it is the computer. Other devices may be Talkers - putting data onto the bus, Listeners - reading from the bus, or inactive - neither talking nor listening. Only 1 device may talk at once, but more than 1 may listen to the Talker.

You can link devices in either a linear, star or combination configuration using a shielded 24-conductor cable. The bus uses standard TTL level negative logic.

Usually, you only have one GPIB interface board per computer. However, you can have several boards if you want to control more instruments (the limit is 14 instruments per System Controller). If you do not have an empty slot in your computer for your GPIB board, the controllers provide GPIB functionality through another outlet, such as serial or parallel port.

In addition to these interfaces, GPIB controllers can escape the confines of the computer case and find their way into the realm of the Ethernet, parallel, serial, USB, and FireWire ports. These alternative GPIB interfaces have all the functionality of plug-in GPIB interfaces.

Several types of GPIB controllers are available: Ethernet, parallel port, IEEE 1394, USB, 232 (serial), and 485 (serial). A controller connects the GPIB instrument to your computer instead of using a GPIB board for the connection. For example, the serial controller makes a connection from your serial port to your GPIB instrument.

In addition, depending on the controller used, you can extend GPIB communication. For example, the RS-485 controller can extend GPIB up to 1.2 km.

A majority of instruments are designed to be used on the bench or in a system –they therefore have the displays and controls needed by users in a stand-alone mode, and the GPIB interface to enable them to be used in a system for effective usage. This provides the users the flexibility to choose the appropriate instrument from a wide range of manufacturers. GPIB based systems have size and speed significant considerations.

VME

VME bus (Versa Module Europa) is a flexible open-ended bus system which makes use of the Euro card standard. It was introduced by Motorola, Phillips, Thompson, and Mostek in 1981. VME bus was intended to be a flexible environment supporting a variety of computing intensive tasks. It is defined by the IEEE 1014-1987 standard.

VME provides a modular means to implement computer-independent systems for real-time data capture, industrial processing, instrumentation, automation and communications for use in Science, Offices, and Industry.

The bus allows multiple masters, and contains a powerful interrupt scheme. A resource manager is required to handle the interrupts.

VME bus has a 32-bit address bus (up to 4 gigabytes of addressable memory), a 32-bit data bus, and can handle data transfers at speeds in excess of 40 Mbytes/sec.

A typical transfer consists of an arbitration cycle (to gain bus control), an address cycle (to select the register) and the actual data cycle. Read, write, modify and block transfers are supported.

The VME bus system consists of 4 sub-buses: The Data Transfer Bus, the Arbitration Bus, the Priority Interrupt Bus and the Utility Bus. Data transfer is asynchronous supporting modules with a broad variety of response times.

PXI

PXI (**P**CI Extensions for **I**nstrumentation) is a rugged PC-based platform for measurement and automation systems.
PXI is designed for measurement and automation applications that require high-performance and a rugged industrial form-factor. With PXI, you can select the modules from a large number of vendors and easily integrate them into a single PXI system. A typical 3U PXI module measures approximately 4x6 inches (100x160mm) in size, and a typical 8-slot PXI chassis is 4U high and half rack width, full width chassis contain up to 18 PXI slots.
(U (Unit) A unit of measurement (1.75") of the height of a rack-mounted device. Thus, a 1U product has a vertical measurement of 1.75"; 2U is 3.5"; 3U is 5.25" and so on.)
PXI uses PCI-based technology and an industry standard governed by the PXI Systems Alliance (PXISA) to ensure standards compliance and system interoperability. There are PXI modules available for almost every conceivable test, measurement, and automation application. PXI is based on Compact PCI, and it offers all of the benefits of the PCI architecture including performance, industry adoption, COTS technology. PXI adds a rugged Compact PCI mechanical form-factor, an industry consortium that defines hardware, electrical, software, power and cooling requirements, leaving nothing to chance. Then PXI

adds integrated timing and synchronization that is used to route synchronization clocks, and triggers internally.
(The Peripheral Component Interconnect(PCI), or PCI Standard, specifies a computer bus for attaching peripheral devices to a computer motherboard. These devices can take any one of the following forms:

- An integrated circuit fitted onto the motherboard itself, called a planar device in the

PCI specification.
- An expansion card that fits into a socket.

PCI Express, officially abbreviated as PCI-E or PCI- E, is a computer expansion card interface format. It was designed to replace PCI, PCI-X and AGP (graphics card interface). PCI-E is based around serial links called lanes. The PCI-E 1.1 specification supports x1 (pronounced "by one"), x2, x4, x8, x16, and x32 lanes. In each lane, the most common version PCI-E 1.1 carries 250 MB/s in each direction. Every lane of the PCI- E is a full duplex link; capable of simultaneous transmit and receive.

A Compact PCI (C PCI) system is a 3U or 6U Euro are-based industrial computer, where all boards are connected via a passive PCI backplane. The pin assignments of the connectors are documented in standards, published by the organization PICMG US and PICMG Europe. PICMG stands for PCI Industrial Computers Manufacturers Group. The connectors and the electrical rules allow for 8 boards in a PCI segment. Multiple segments are allowed with bridges)

Most PXI instrument modules are register-based products, which use software drivers hosted on a PC to configure them as useful instruments, taking advantage of the increasing power of PCs to improve hardware access and simplify embedded software in the modules. The open architecture allows hardware to be reconfigured to provide new facilities and features that are difficult to emulate in comparable bench instruments. PXI system data bandwidth performance easily exceeds the performance of the older VXI test standard.

PXI modules providing the instrument functions are plugged into a PXI chassis which may include its own controller running an industry standard Operating System such as Windows XP, Windows 2000, or even Linux (which is not yet PXISA approved), or a PCI to PXI bridge that provides a high speed link to a desktop PC controller. Likewise, multiple PXI racks can be linked together with PCI bridge cards, to build very large systems such as multiple source microwave signal generator test stands for complex ATE applications.

Compact PCI and PXI products are interchangeable, i.e. they can be used in either Compact PCI or PXI chassis, but installation in the alternate chassis type may limit the functionality of certain bus-specific features. So for example you could mount a Compact PCI Network interface controller in a PXI rack to provide additional network interface functions to a test stand.

PXI systems are comprised of three basic components – chassis, system controller, and peripheral modules.

Figure 5.1A standard 8-Slot PXI chassis contains an embedded system controller and seven peripheral modules. *(Courtesy: National Instruments)*

PXI Controllers

As defined by the PXI Hardware Specification, all PXI chassis contain a system controller slot located in the leftmost slot of the chassis (slot 1). Controller options include remote controllers from a desktop, workstation, server, or a laptop computer and high-performance embedded controllers with either a Microsoft OS (Windows 2000/XP) or a real-time OS (Lab VIEW Real-Time)

Laptop Control of PXI

With Express Card MXI (Measurement Extensions for Instrumentation) and PCMCIA Card Bus interface kits, users can control PXI systems directly from laptop computers. During boot-up, the laptop computer will recognize all peripheral modules in the PXI system as PCI devices.

PC Control of PXI

With MXI-Express and MXI-4 interface kits, users can control PXI systems directly from desktop, workstation, or server computers. During boot-up, the computer will recognize all peripheral modules in the PXI system as PCI devices.

The MXI-4 interface kit provides a 78 MB/s PCI -to- PCI bridge from PC to the PXI system. Using a MXI-4 link, you can implement either a daisy-chain or a star topology to build multi chassis systems .

Because all remote control products are software transparent, no additional programming is required.

Embedded controllers are ideal for portable systems and contained "single-box" applications where the chassis is moved from one location to the next.

PXI is an open industry standard and nearly 1000 modules are available from the 65+ members of the PXI Systems Alliance.

VXI

VXI (VME Extensions for Instruments)is a modular instrument standard for the U.S. military. These modular instruments (instruments-on-a-card) became very popular in Aerospace/Defense industry and manufacturing test applications where size and throughput were important.

LXI

Agilent Technologies (formerly Hewlett-Packard) and VXI Technology, Inc. introduced LXI (LAN-based extensions for Instrumentation); combining the best of GPIB instruments and VXI modules. With LXI you get the reduced size of VXI, high throughput of LAN, and the high performance measurements of GPIB. No card cage, no slot 0 and no expensive PC to instrument communications link are required in LXI.
LXI is the next generation of test systems combining state-of-the-art measurements in a small package at a cost-effective price. LXI modules are full-rack width, 1U tall or half-rack width, 1U or 2U tall. Signals enter and exit the module from its front panel while LAN (IEEE 802.3), power, and trigger cables are found on the back of the module. All modules are designed to be easily mounted in a standard 19” rack or stacked on the bench.
The IEEE 1588 standard is another key enabling technology of LXI. In a nutshell, the purpose of IEEE 1588 is to synchronize the various clocks within a system. The inclusion of this standard in LXI Class B and Class A instruments provides precise timing that enables simultaneous measurements across local and remote devices.
Unlike card cage-based systems, LXI modules can be easily distributed in a test rack, across a lab or throughout a building. This allows you to place instruments where they can best meet the needs of each measurement or application.

Physical layer: To help ensure successful instrument interaction, the LXI standard specifies automatic negotiation of LAN transmission speed and duplex communication. The standard also recommends Auto MDIX, a feature that enables the use of either straight-through or crossover LAN cables in direct controller-to instrument or peer-to-peer connections. The instrument automatically adjusts to the existing cable and its communication counterpart.

Network (IP) layer: LXI instruments support automatic IP configuration through a DHCP server (often available in managed corporate networks and in cable/DSL routers) or through

dynamic configuration of local addresses (typically used in small or ad-hoc networks). LXI also recommends support for DNS, which instruments can use to publish their host name through a DNS server (another feature usually available in corporate networks).

Application layer: LXI-compliant instruments support the VXI-11 protocol (based on remote procedure calls) for automatic discovery of new instruments and identification through the *IDN? query.

LXI devices include a built-in Web server to enable configuration and troubleshooting.
The LXI standard spans classic instruments, faceless modular instruments, and functional building block modules (synthetic instruments or SIs).

With their embedded processors, many of today's test instruments have enough computing power to carry out measurement tasks on their own, freeing the system controller for other tasks. LXI uses this power to provide greater flexibility in communication, too: instruments can communicate without arbitration through the system controller. Instead, they can use TCP for peer-to-peer communication and UDP for multicast (one to-many) messages.

One key advantage of LXI is its ability to leverage ongoing innovations in Local Area Networks that satisfy the need for speed.

With a Fast Ethernet connection (IEEE 803.2u, 100 Mb/s), the maximum payload data rate is approximately 12.5 MB/s. systems. There is no need to create custom gateways—remote access comes without extra effort. Using your corporate intranet or the public Internet, large distances can be bridged

easily and the connection is transparent to the end user. Ethernet is also widely available in urban areas through cable or DSL modems or wireless LAN hotspots, and in rural areas through wireless IP services based on GPRS or UMTS.

If a distributed application needs to access the public Internet, you can use a virtual private network (VPN) to send IP packets securely, encrypted through IP sec or other encryption protocols.

In a typical LXI-enabled distributed application the system will include intelligent instruments capable of performing measurement tasks on their own, independent of the system controller. To make this approach practical, the instruments will typically include a local clock that enables them to time-stamp measurements and events.

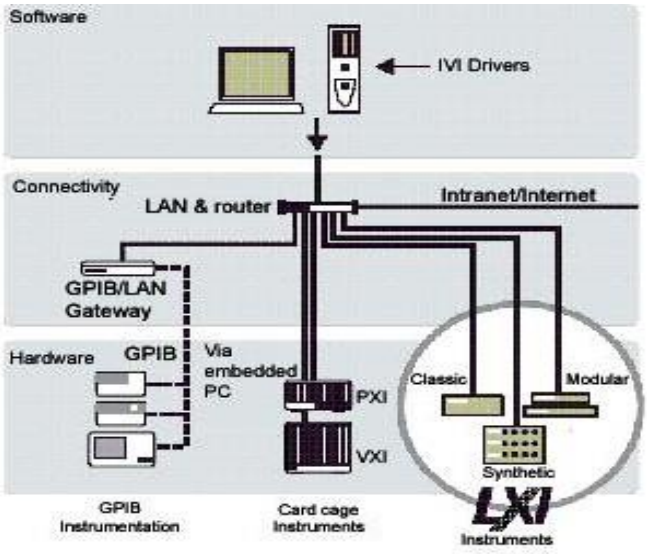


Figure 5.2 Agilent LXI System

The capability of instrumentation to respond to events or trigger commands to perform measurements is an important part of a test system. All instrumentation platforms have this feature.

5.2 Test Fixtures Basics of Automatic Test Program Generation

Test Stand

In the past, the interface between hardware and software was a challenging development task, typically requiring extensive knowledge of the hardware, low-level programming, and even real-time system development

Each proprietary ATE system using their own operational software and thus no standardization of user interface and interoperability of software between different ATE hardware.
In 1998, National Instruments introduced Test Stand, a test management environment that provided the key features of software typically found on proprietary ATE. Test Stand brought an important ATE technology – test management – to a very broad set of users. Anyone doing automated testing who wanted to organize, execute, and record information from a set of test routines could use Test Stand.
Consequently, the user base became very broad, and today Test Stand is used by a large set of users, integrators, and third parties.

The tight integration of Test Stand with development environments such as Lab VIEW make it a very flexible tool for creating automated tests. Together, Test Stand and Lab VIEW provide an optimal test environment not just for the manufacturing floor, but also for desktop applications.

ATE system

Automated Test Equipment (ATE) Systems are used to test a wide range of electronic devices and systems, from simple components to integrated circuits (ICs), printed circuit boards (PCBs), and complex, completely-assembled electronic systems. Hence they are equipped with necessary hardware and software for multiple test techniques and advanced test software.

The basic function of an ATE is to test a component or a board by driving a set of drive patterns, reading back the patterns and comparing them against expected drive patterns. These expected patterns can be from an off-line or on-line software simulator or defined by the test programmer or even learnt from a known good sample. The ATE software then declares the result of comparison as Pass or Fail.

This function is extended to cover a sequence of various tests and branching of tests based on pass/fail for each type of device/ board.

These steps constitute a test program. A test program may consist of hundreds of various tests which are executed in a particular sequence automatically before declaring a result. An effective test program should cover 100 percent of the board's components including the inter connection between the components.

The next step is diagnosing a fault when a test program result is *Fail.*
An ATE should guide the user to locate the fault. On repairing the fault, again the test program is run and the result should be a *Pass.*
The above conditions require a matching hardware and software components for reliable and repeatable performance.

Test vectors define the parameter space or set of test cases you want to run. They are sequences of signals applied to the pins of an integrated circuit to determine whether the integrated circuit is performing as it was designed.
Most Automatic Test Equipment (ATE) use functional test vectors that are based on tester timing cycles.

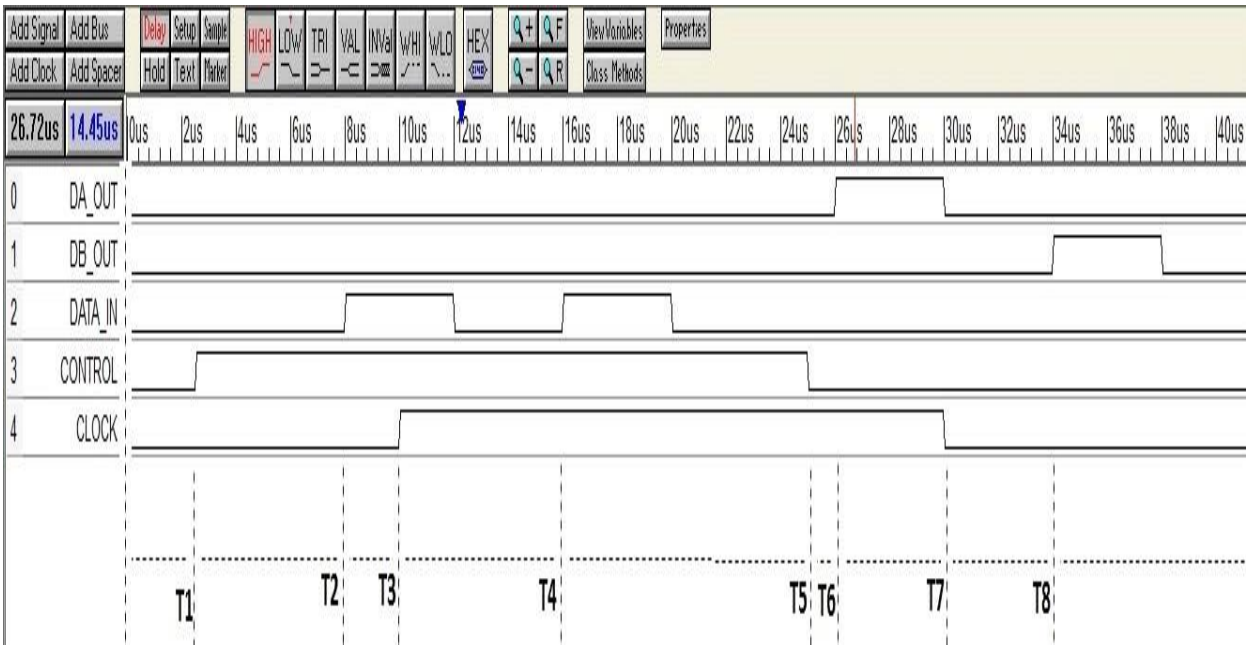
An ATE timing "cycle" is a particular fixed timing pattern that usually repeats. The pattern consists of a time interval, lets say, from 0 to Tm. During this interval, various fixed times can be defined to do the following things:

- set a clock high or low (there can be multiple clocks)
- apply data (there can be different application times that apply to different signals)

3. "strobe" results (that is, capture the value of one or more signals). There can be different strobe times that apply to different signals. The data captured at the strobe time is then compared to expected results and any differences are output as errors.

strobe - A signal that clocks a logic state into a latch or D flip flop. In a semiconductor test, an edge strobe records a comparison result at a specified instant in the cycle; a window strobe records any error occurring during a timed interval.

An example of a timing cycle is depicted as in the Fig. below:



Courtesy: Qmax Test Equipments PLtd.

Figure 5.3 Timing cycle Diagram

As you can see, the clock rises at T3 and falls at T7; control inputs are applied at T1 and held until T5, data inputs applied at T2 and held until T4; and the da_ out signals are strobed at T6 and the db_out signals are strobed at T8.

The similarity between a timing cycle and data sheet timing diagrams is more than coincidental. You will be selecting T1 and T2 to correspond to setup times, T4 and T5 to correspond to hold times, and T6 and T8 to correspond to max propagation times relative to the rising clock edge.

Test or simulation vectors then consist of sets of data to be applied within a given cycle, and sets of results to be compared. The clocking and timing information is not contained within the test vectors, but is contained within the definition of the test cycle instead.

A test may contain more than one type of test timing cycle. For instance, when testing a memory, you will probably need to define a write cycle, a read cycle and possible a read_modify_write cycle. Processor chips may have several complex cycles.

The ATE architectures can be divided into two major sub components, the data generator and the pin electronics. The data generator supplies the input test vectors for the DUT (Device Under Test), while the pin electronics are responsible for formatting these vectors to produce waveforms of the desired shape and timing. The pin electronics are also responsible for sampling the DUT output responses at the desired time.

A test program, usually written in high level language, is an important ingredient for controlling the data generation and formatting. Algorithmically generated test patterns may consist of subroutines, pattern and test routine calls or sequenced events.

The test programs also specifies the timing format in terms of the tester edge set. An edge set is a data format with timing information for applying new data to a chip input pin and includes the input setup time, hold time, and the waveform type. (source: VLSI TEST Principles and Architectures: GOOGLE books)

ATE system components

Main Processor

The heart of an ATE system is dedicated test vector processor. This test vector processor controls the timing and construction of drive waveforms that will be driven into the UUT and to acquire response data. This sequencer processes the software commands registered in the in-built registers and generates the sequence of addressing the write and read RAMs, which are behind each pin driver channel. Addressing of multiplexer relays are also processed by the sequencer.

Digital sub system

The digital sub system contains drivers for each digital channel and measures the feedback from each device digital pin (digital channels).

Analog sub system

This system contains drive/ read facility testing analog devices (analog channels).

Interface

The ATE interface consists of hundreds of channels for simultaneous drive / measurement of digital / analog signals from a board. Multiplexers are also used when more number of channels are required but not all of them used simultaneously. The connectors vary from simple probes to specialized clips for various device types. A board can be tested through a test fixture or even a Robotic Probers.

Power Supply system

The power supply system contains various supplies like 3.3, 5, 12 V for testing various families of devices. For special voltage requirements ATE provides for linking external Programmable power supplies (e-x 32V, 64V).It is to be noted that the ATE automatically selects the required supply voltage(s) and supplies to the board/ device under test only for the duration of test.

JTAG Terminals

An ATE may contain JTAG terminals for connecting to Boundary scan devices.

External Instrumentation

The ATE configuration also includes facility for linking with external instruments through an analog / digital highway matrix.

External controller

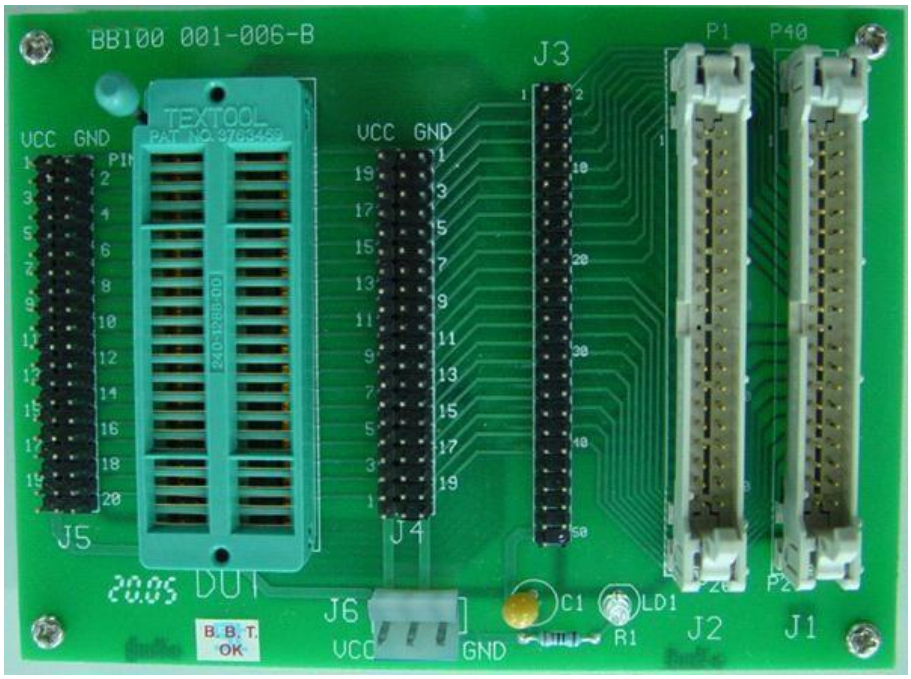
The Human interface to an ATE can be a simple stand alone PC called as External controller. This controller will be interfaced via PCI card/ USB connection/ serial connection to the ATE and its components.

Out-Circuit Functional Tester

Out-Circuit Functional Test (OCFT) is done when the device is taken out of circuit for testing. An Out-Circuit Board Adaptor is to be used for this testing. Out-circuit mode is useful for checking discrete device's functionality with or without loading.

This Out-Circuit Board Adaptor board is provided with a tiny lever operated ZIF (Zero Insertion Force) socket for easy plug-in and plug-out of device. It is provided with Loading Hybrids (one each for TTL and CMOS) to load all the pins to facilitate testing of ICs while loaded. These act as loads as well as pull-up resistors to the Device output pins. The appropriate loading hybrid is to be plugged in respective SIP jacks provided.

The Out-circuit board adaptor is to be connected to the tester with suitable Interface Jig.



• **Figure 5.4 Out-circuit Board Adapter**

The ATE System consisting of the components described above along with the dedicated software modules takes the shape of large standalone system capable of meeting all the test requirements of board troubleshooting.

The Pin Electronics

The pin electronics is the interface between the test system resources and the DUT. It supplies input signals to the DUT and receives output signals from the DUT.

Each test system has its own unique design but generally the Pine electronics circuitry will contain:

- Driver circuitry to supply input signals with programmable drive levels (VIH and VIL)
- Programmable slew rate of the drive signals (optional)
- I/O switching circuitry for turning drivers and current loads on and off.
- Voltage comparator circuitry for detecting sense logic levels.
- A connection point to the PMU
- Programmable current loads
- Possibly additional circuitry for making high speed current measurements
- RAM for storing drive, Receive and Compare patterns.

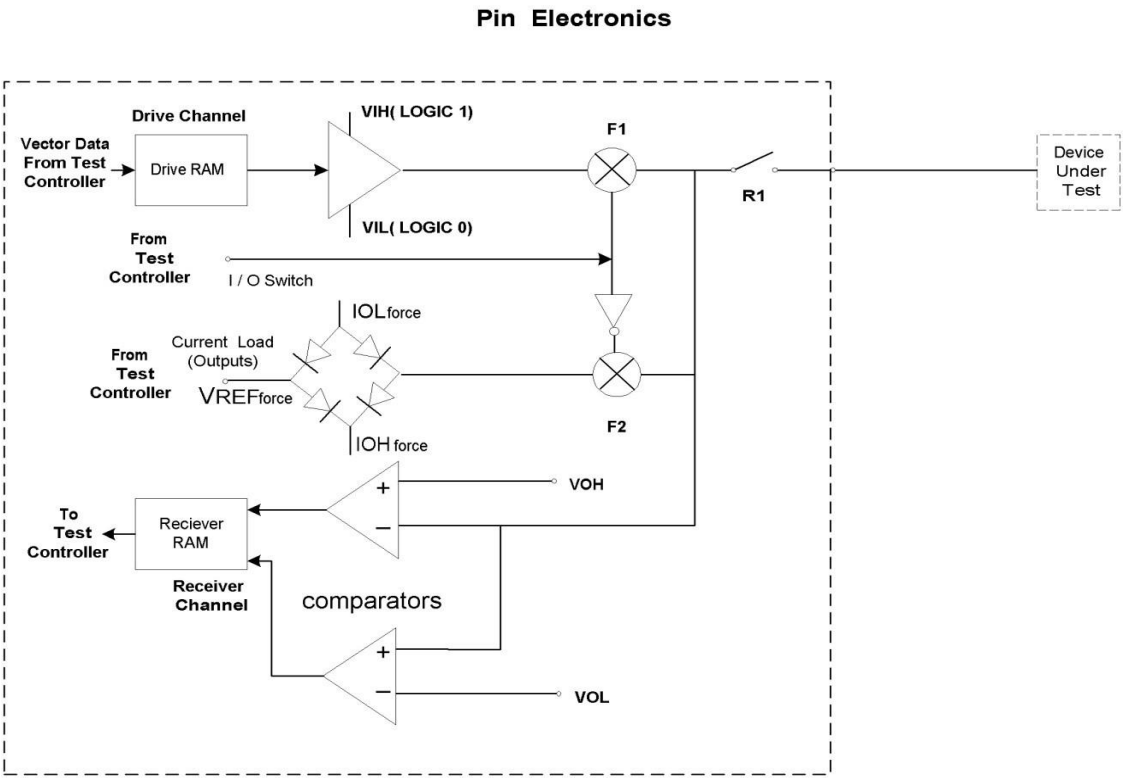


Figure 5.5 Digital Pin Electronics

The Driver

The driver circuitry receives vector data from the test controller section of the test system. As the signal passes through the driver, the VIL/ VIH references from the Reference voltage supplies (RVS) are applied to the formatted data. If the test controller instructs the driver to drive to a logic 0, the driver will drive to the VIL reference. VIL (Voltage In Low) represents the maximum guaranteed voltage value that can be applied to an input and still be recognized as a logic 0 by the DUT circuitry.
If the test controller instructs the driver to drive to a logic 1, the driver will drive to the VIH reference. VIH (Voltage in High) represents the guaranteed minimum voltage value that can be applied to an input and still be recognized as a logic 1 by the DUT circuitry.

When the tester channel is programmed as an input, F1 FET turns on and the R1 relay is closed allowing the signal from the driver to pass through to the DUT. When the tester channel is programmed as an output or is in a “Don’t care” mode the FI FET is turned off and the signal from the driver will not pass through to the DUT.

The F1 FET is a field effect transistor used as a very high speed switch. It isolates the driver circuitry from the device under test. F1 FET is used during IO switching, which is when the DUT alternates between receiving data from the test system (reading data) and supplying data to the test system (writing data). The same pins function as both input and outputs.

Current Loads

The current loads also known as Dynamic Loads or Programmable Current Loads, act as a load to the DUT outputs during functional tests and can be programmed to supply positive and negative currents. Positive current flows from the test system into the device and the negative current flows from the DUT into the test system.

The dynamic loads provide both IOH (Current Out High), which is the amount of current that a DUT output must source when driving a logic 1 and IOL current (current out low), which is the amount of current that a DUT output must sink when driving a logic 0.
After the IOL and IOH current values are set by the test program, the VREF voltage is used to set the switching point of the IOL and IOH currents.

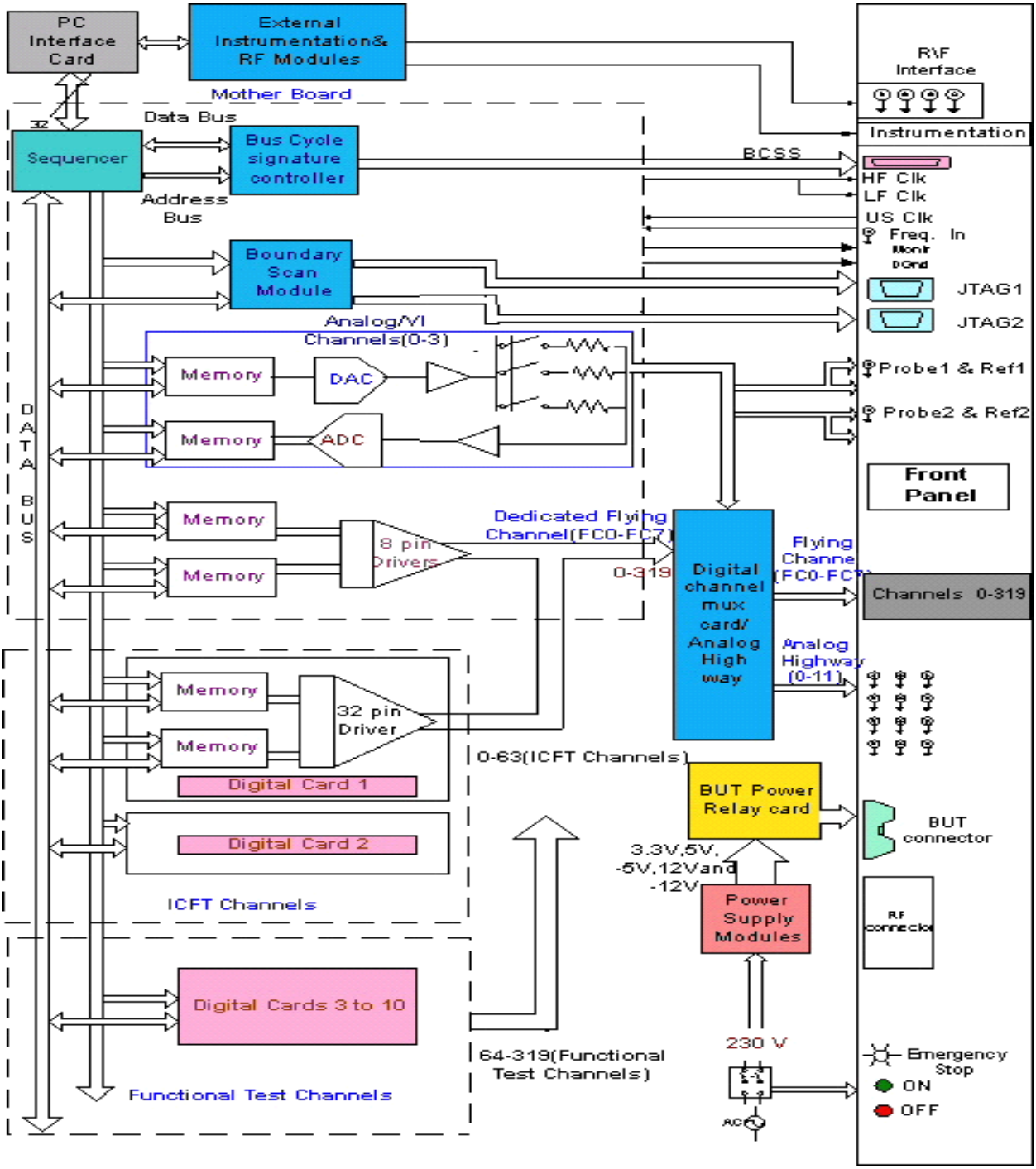
The switching point is the output voltage above which IOH flows and below which IOL flows. When the output voltage from the device under test is more negative than VREF, IOL current flows. When the output voltage from the device under test is more positive than VREF, IOH current flows. The current loads are also used during the functional open/short test.

The F2 FET is used as high speed switch. The F2 FET isolates the current load circuitry from the device under test and is used during IO switching.

If the tester channel is programmed as an output, F2 is on, allowing current to flow to and from the device under test. If the tester channel is programmed as an input, F2 is off.

The voltage comparators

The voltage comparators are used during functional testing to compare the output voltage of the device under test to reference voltages supplied by the RVS. The RVS supplies a reference for a valid logic1(VOH) and a valid logic 0 (VOL). If the DUT output voltage is equal to or less than VOL, it is recognized as a logic one. If the output voltage is greater than VOH, it is considered as a to be a tri -state or a bad output.



• **Figure 5.6 ATE Block Diagram**

A sample block diagram of an ATE is shown above. The features shown above are present in a commercially available ATE. They vary for different categories.

Data drive formats

The start of each new cycle is time zero or T0.

The input data is created by combining:

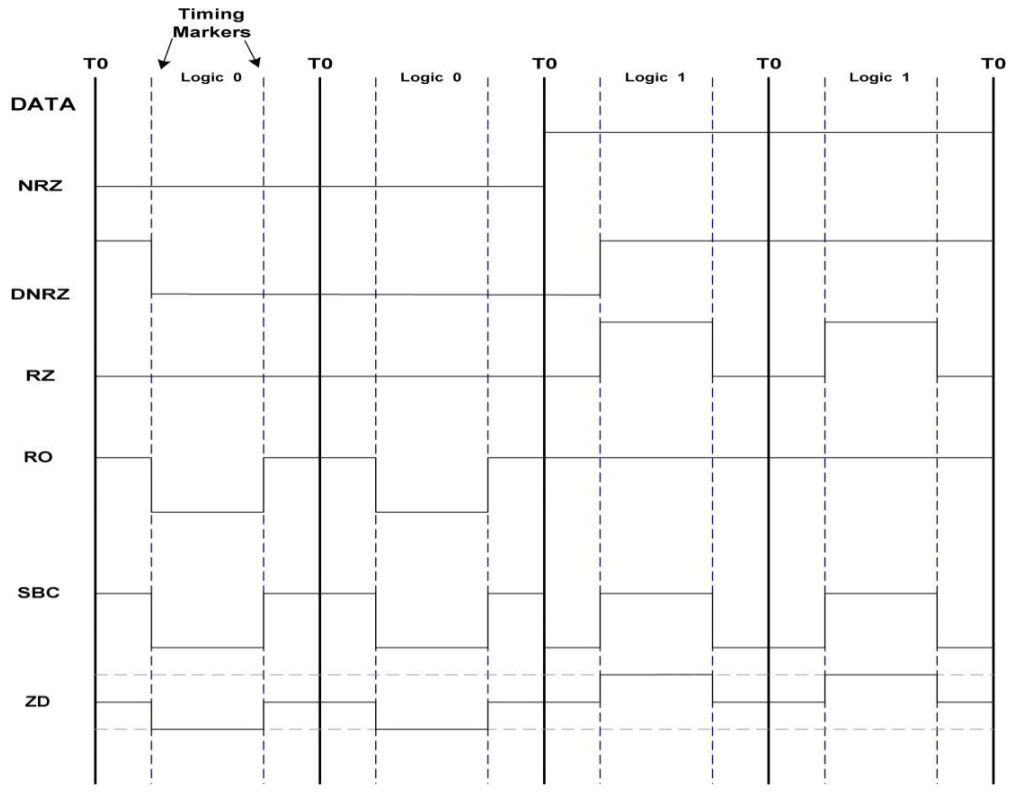
- Test vector data (instructions or stimuli to the DUT)
- Input Signal timing (signal transition points)
- Input signal formats (wave shape)
- Input levels (VIL/ VIH)
- Time set selections (if more than one-time set is used).

Input data in its simplest form consists of a logic 0 or logic1 level stored as test vector data. The voltage levels which represent logic 0 or 1 are produced at the test head by the VIL/VIH reference voltages.

Many input signals require more complex data containing unique formats (wave shape) and timings (edge placements). This information is contained in the main test program and is controlled through the format and timing statements of the test language.

Input signal formats

Signal formatting is very important. Signal formats, when combined with vector data, edge placements and input levels, define the wave shape of input signals to the DUT.



- **Figure 5.7 Input Signal Formats**

NRZ Non return to Zero represents the actual data stored in vector memory and contains no edge timing. The amplitude level is high during bit-time, if a "1 "is transmitted, and is low, if a "0" is transmitted. If two or several "1"s are transmitted in sequence, then, the level remains high.
NRZ data changes only at the beginning of each cycle (T0).

DNRZ
RZ Return to zero provides a positive pulse when vector data is logic 1 and no pulse when vector data is logic 0(the signal remains at logic 0), RZ signals have a leading (rising)edge and a trailing (falling)edge. This signal format can provide a positive clock when all vector data for the pin is logic 1. Active high signals such as CS2 (chip select signal) need RZ format.

RZ format requires that a "1" always return to the low state during the bit time even when two "1" s are transmitted in sequence, whereas the "0" bit remains at a low level. This eliminates the possibility that a long string of "1" s will produce a constant high level but does not prevent a long string of '0" s from producing a constant low level.

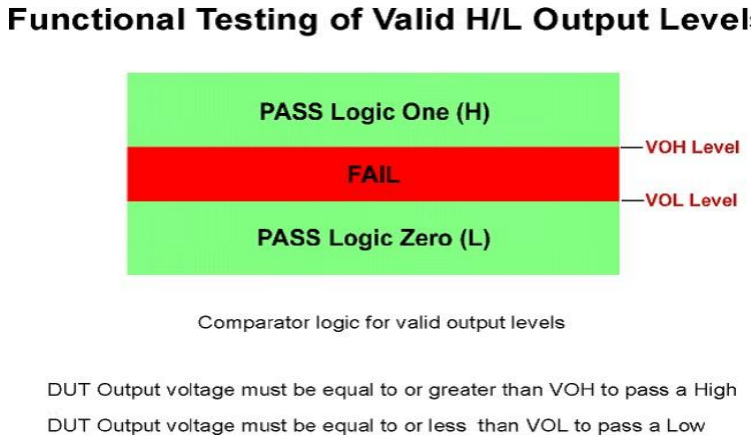
RO Return to one provides a negative pulse when vector data is logic 0 and no pulse when vector data is logic 1(the signal remains at logic 1). RO signals have a leading (falling) edge and a trailing (rising) edge. This signal format can provide a negative clock when all vector data for the pin is logic 0. Active low signals such as output enable (OE/) use RO format.

SBC Return to (surround by) compliment can provide three edge transitions per cycle. The signal format creates a complex signal based on the vector data. It inverts the data at the start of the cycle(T0), waits a predefined delay, presents the actual vector data for the specified pulse width, then inverts the data again for the remainder of the cycle. This signal format is the only format that will guarantee both setup and hold time in a single execution of the test vectors. SBC format is also known as exclusive-OR (XOR format).

ZD Return to Z State: Z (impedance) Drive/ no drive allows the input drivers to turn on and off within a cycle. When the driver is off the tester channel is the high impedance state. When the driver is on, the DUT input will be driven to a logic 0 or1 depending on vector data.

Clock signals are usually RZ (positive pulse) or RO (negative pulse) formats. Active high control signals such as CS (chip select) or READ are often RZ format. Active low control signals such as CS/ (chip select bar) or OE/ (output enable bar) are often RO format. Data signals that have a setup and hold time parameters require SBC formats. Other inputs may require NRZ or DNRZ formats.

Testing Valid (L/H) Output Levels



- **Figure 5.8 VOH/VOL Limits**

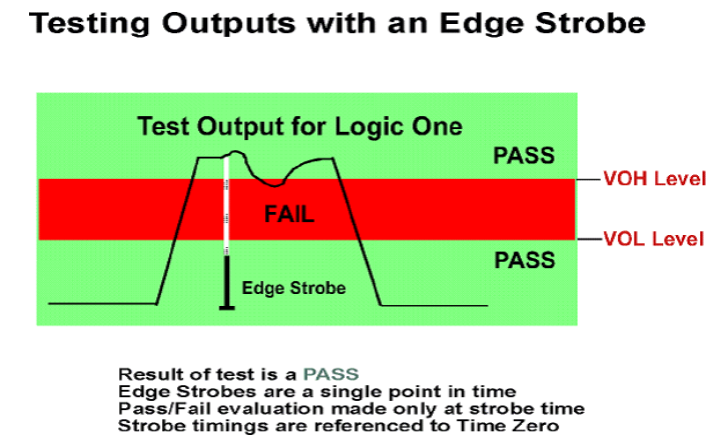
Figure shows the pass/fail/pass relationship between the DUT output and the VOL/VOH reference values for testing valid (normal) output levels.
The output voltage produced by the DUT must be equal to or greater than the VOH reference level of the comparator in order to qualify as a valid output high.

The output voltage produced by the DUT must be equal to or less than the VOL reference level of the comparator in order to qualify as a valid output Low.

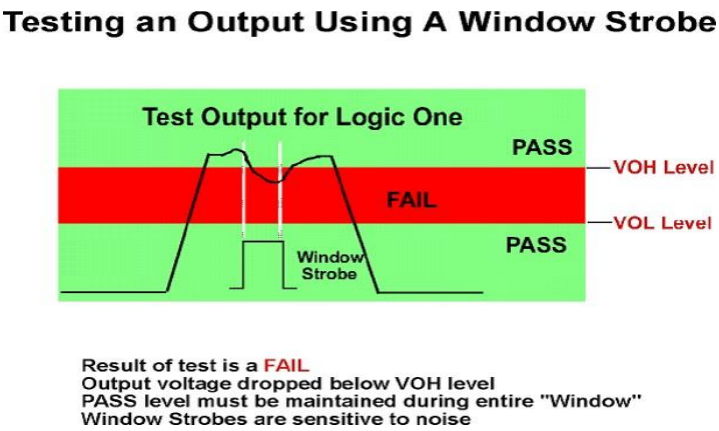
Output Testing using an Edge Strobe

The voltage value of a digital output signal is evaluated at a particular point in time to determine the logic state of the output. An output *Strobe* is the timing marker within the test system that is used as the timing reference for output signal evaluation. Many test systems offer an individual strobe marker on each tester channel, this allows each output signal to be evaluated independently. There are two types of strobe markers commonly available: edge and window.

Figure 4.7 shows the placement of an edge strobe and the point of output evaluation. In this example the output signal is being tested for a logic one. As shown above the output voltage is greater than the VOH reference level at the time that the edge strobe occurs, in this example the result of the test is a PASS. An edge strobe makes an evaluation at a single point in time. Edge strobes are often used in high frequency testing because they are less affected by noise or ringing of the output signal. The exact placement of the output strobe is determined from information contained in the device AC specification, but strobe timing is programmed relative to T0 (time zero of the tester cycle).



- **Edge Strobe**



- **FIGURE 5.9WINDOW PROBE**

Output Testing using a Window Strobe

Figure 9.7 shows the placement of a window strobe and the duration of output evaluation. In this example the output signal is being tested for a logic one. Notice that the output voltage falls below the VOH reference level during the time that the window strobe is active, the result of this test is a FAIL. A window strobe makes the evaluation during the entire strobe width. The test will fail if the output voltage fails to correctly qualify at any point during the strobe window timing.

Ringing or noise on the output signal can cause the test to fail when using a window strobe.

Analog drivers/ sensors

The analog sub system analog channels with 12/14-bit accuracy and sampling rate of the order 20MHZ or more. These channels are programmable and also can be used as stimulus/ response channels. These analog channels are fully synchronized with digital channels. These channels are also used for advanced VI trace methods and DC parametric measurements.

DC parametric measurements

These units are normally found in high end ATEs.

ATE pin drivers drive up to 50 mA or more and thus will be able to drive a faulty input pin that overdraws current and may pass functional tests. This undetected fault may cause a problem when the PCB is fitted back into the equipment.

Force Voltage Measure Current Test (FVMC) is used to detect this problem by measuring the input bias current. If this parameter is not within the specifications, the device is faulty.

Force Current Measure voltage (FCMV) test is used to test a device current load. This test helps to find out whether a device like an IC draws current as per specifications.

Digital Highway

A common path or a set of parallel paths over which signals from more than one channel pass with separation achieved by time division.

In ATE, Digital highways provide for synchronization of the test cycle to UUT events and digital timing measurements through tester pins. An AC PMU (Parametric Measurement Unit) in ATE is used to measure the frequency, propagation delay, RISE/FALL time and pulse width through digital highway.

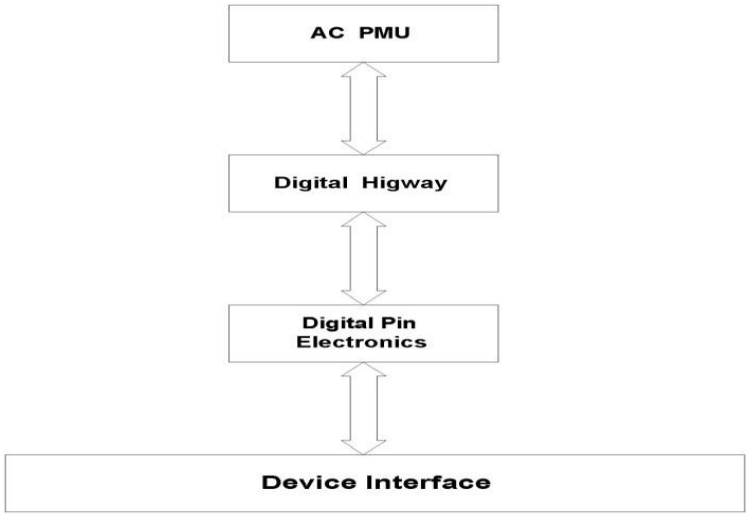


Figure 5.10 Digital Highway

Analog Highways

Analog Highways are used to route the test channels to take measurements using external instruments like PXI instruments and controlled through the application software.

The Highways are routed inside the system through analog highway routing module card.

The Highway channels are bi-directional paths of force or sense characteristics. Driving and receiving data through this highway is possible.

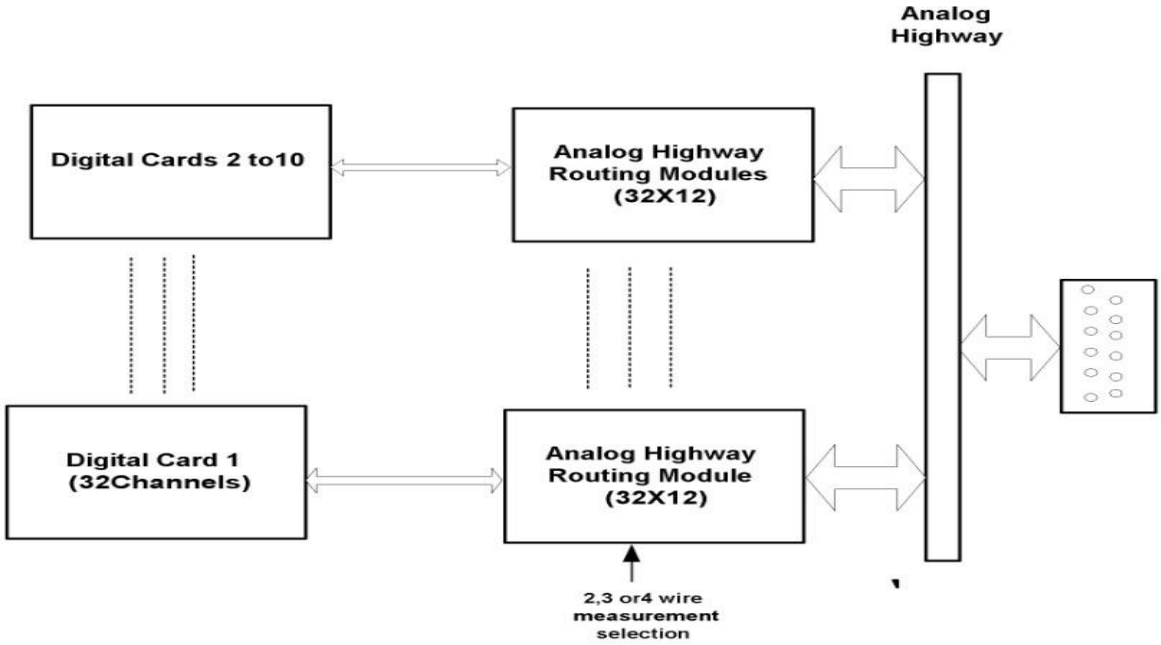


Figure 5.11 Analog Highway

User programmable relay control

Programmable Relay Control are crucial in multiplexing operations in ATE. The routing of digital, analog signals to interface terminals are established through test programs.

Bus Cycle Signature System Integration (BCSS)

Qmax ATE is configured with the channels provision for using Bus Cycle system. These channels are facilitated with D-type connector in the front panel.

This BCSS connector channels are to be interfaced to the board under test through an interface Pod for each processor as per its type. This Pod is to be designed for each processor board and be made accessible for connection to the test points in the board.

Enabling the ATE tester channels to receive the signatures and drive the software program to run, helps to see the faults in the Board.

This automated process is used to find the faults in the processor mounted on boards.

Interface POD or Personality Adapter

The BCSS testing is highly depending on the design of the interface POD for the selected CPU on board. This generates the required strobes such as General Strobe (GS)/Data Strobe (DS) and Bus Cycle ON test window.

BCSS hardware also provides a facility to place General Strobe in steps of 1 nS in advance or retard, using the built in clock Vernier for 10 nS. This will be useful in testing timing related problems on board.

The Pod is to be connected between the ATE front panel BCSS connector and the Board interface points or test points as defined in the program.

From a Known Good Board (KGB) the signatures learnt are stored in the database and compared with that of any suspected board and hence declare the suspected board faulty.

The learn and compare methodology used here is easy to use and fix the problematic processors on board so as to minimize the effort of testing the processors using complex testing tools and jigs.

Figure 5.12 BCSS Interface POD

In the picture shown above, The BCSS terminal in ATE front panel is connected to the Interface POD (A small box). This Interface POD connects to the Board BCSS terminals (accessed through a test fixture.). The board is connected to this fixture through bed-of nails from the bottom.

5.3 Basics of Automatic Test Program Generation

Test Program Generation

Device Model library

The device library file is a special library file which contains all of the information required to render and otherwise support the devices available.

The stimulus and response patterns for a device are known as a "test routine." Each test routine is stored in a device model in a device models library. As a result, tests for many common digital IC's can be programmed once, in advance, stored in the library, and then called upon when needed. This greatly simplifies test generation since this pre-programmed test can be used over and over again.

A hardware description language (HDL) is used in creation of a device library.

A **hardware description language** or **HDL** is a language for formal description of electronic circuits. It can describe the circuit's operation, its design and organization, and tests to verify its operation by means of simulation.

A Hardware Description Language (HDL) is a standard text-based expression of the temporal behavior and/or (spatial) circuit structure of an electronic system. In contrast to a software programming language, an HDL's syntax and semantics include explicit notations for expressing time and concurrency which is the primary attributes of hardware. Languages whose only characteristic is to express circuit connectivity between hierarchies of blocks are properly classified as net list languages.

HDLs are used to write executable specifications of some piece of hardware. A simulation program, designed to implement the underlying semantics of the language statements, coupled with simulating the progress of time, provides the hardware designer with the ability to model a piece of hardware before it is created physically. It is this execute-ability that gives the illusion of HDLs being a programming language. Simulators capable of supporting discrete event (digital), and continuous time (analog) modeling exist and HDLs targeted for each are available.

The library of a device consists of a **Source** file and **Edit drive** file. A source file contains the full description of the device like device name, family, package No. of pins, device pin details like power supply and ground pins Input and output pins (including pins not connected). This file also contains the functional description of each of its functional block. Once a source file is created, it is compiled and checked for errors. If the simulation results are okay, the device is added to the ATE library of devices.

The Edit drive file consists of creation of test pattern in graphical mode according to the device data.

An ATE is provided with a comprehensive library of test programs for all types of TTL, CMOS, ECL, MSI and LSI devices. This device library is continuously upgraded to include all the new devices in its fold.

Test Languages

The test programs for various types of devices can be made using any of the languages as listed below:

- Digital SSI/MSI devices using VHDL or QDDL or Verilog
- Digital LSI/VLSI devices using VHDL or WEST or Verilog
- Digital and Analog devices and Mixed Signal devices using Python TD

Verilog

Verilog is a hardware description language (HDL) used to model electronic systems. The language (sometimes called *Verilog HDL*) supports the design, verification, and implementation of digital, and mixed-signal circuits at various levels of abstraction.

A Verilog design consists of a hierarchy of modules. Modules are defined with a set of input, output, and bidirectional ports. Internally, a module contains a list of wires and registers. Concurrent and sequential statements define the behavior of the module by defining the relationships between the ports, wires, and registers. Sequential statements are placed inside a begin/end block and executed in sequential order within the block. But all concurrent statements and all begin/end blocks in the design are executed in parallel. A module can also contain one or more instances of another module to define sub-behavior.

VHDL

VHDL is an acronym, which stands for VHSIC (Very High Speed Integrated Circuit) Hardware Description Language. VHDL language, is fast becoming the worldwide standard HDL for digital design & testing.

VHDL can be used in many ways. It is being used for documentation, verification, and synthesis of large digital designs. In addition to being used for each of these purposes, VHDL can be used to take three different approaches to describe hardware. These three different approaches are the structural, data flow, and behavioral methods of hardware description. Most of the time a mixture of the three methods are employed.

VHDL has to date proven to be the most successful and widely used language.

Using a standard HDL virtually guarantees that you will not have to throw away and recapture design concepts simply because the design entry method you have chosen is not supported in a newer generation of design tools. Using a standard language also means that you are more likely to be able to take advantage of the most up-to-date design tools, and that you will have access to a knowledge base of thousands of other engineers, many of whom are solving problems similar to your own.

The most evident application is probably the development of a formal model of the behavior of a system.

Because of the self-documenting character of VHDL, this model can serve as System documentation to a certain degree.

The big advantage of hardware description languages is the possibility to actually execute the code. In principle, they are nothing else than a specialized programming language. Coding errors of the formal model or conceptual errors of the system can be found by running simulations.

Compiler and Simulation packages for testing VHDL programs are available like GMVHDL, Model Sim etc., There, the response of the model on stimulation with different input values can be observed and analyzed.

Software support is available for the necessary refinement steps.

Additionally, hardware description languages offer so called design reuse capabilities. Similar to simple electronic components, like for example a resistor, the corresponding HDL model can be re-used in several designs/projects. It is common use that frequently needed function blocks (macros) are collected in model libraries.

Qmax Device Description Language (QDDL)

The test programs for individual MSI devices are written using the Qmax Device Description Language (QDDL) also. The QDDL test programs are compiled to the library database. The compiler converts the source code into an object format after checking it for the syntax and does semantic analysis.

The object code is then included into the standard library. QDDL programs either internally generate test patterns like binary or gray codes or uses a custom drive pattern stored in the library. With the use of waveform editor, the user can create the required test patterns and store them into the standard library. QDDL is a very simple programming language and makes use of primitives for describing the function of a device. The primitives include Gates, JK Flip Flop, D Flip-Flop, Tri-state Buffer, De multiplexer, Multiplexer etc. These primitives can be cascaded with the use of variables to form a device. With this principle, the user can easily write test programs for any logic device.

QDDL is a very simple programming language and makes use of primitives for describing the function of a device. Some of the primitives used are Gates, JK Flip Flop, D Flip Flop, Tri-state Buffer, De multiplexer and Multiplexer. These primitives can be cascaded with the use of variables to form a device. With this principle, the user can easily write test programs for any logic device.

Wave Event Specification and Test (WEST) Language

The test programs for LSI devices can be written using the WEST. WEST programs are interpreted during run time and generate the required test patterns as the source is interpreted. WEST allows grouping of data, address and control pins for easier emulation of bus cycles that would access the device under test for various operations and tests. It also provides macro functions for repeated operations.

WEST is the test language in which the waveforms that drive the Device Under Test are specified in terms of events on pins. You also specify what should be the response waveforms. In the beginning, pins are grouped and mapped to pin-variables e.g. 16data pins can be mapped to a pin variable DATA. The values of pin variables denote the waveform on the DATA pins at the current tick. Each pin variable has its own independent current tick which is advanced as events occur on it.

Python Language

Python is an interpreted, interactive, object-oriented programming language.

Python combines remarkable power with very clear syntax. It has modules, classes, exceptions, very high-level dynamic data types, and dynamic typing. Python is also usable as an extension language for applications that need a programmable interface.

The Python implementation is portable: it runs on many brands of UNIX, on Windows, OS/2, Mac, Amiga, and many other platforms.

Graphical User Interface (Waveform Editor)

The waveform window shows device pin signals in the form of logic level trains for the complete duration of the test cycle. Waveform Editor offers a number of ways for viewing the waveforms on its View menu.

The legends for the logic levels are as follows:

Legend Logic Level	Signal
0	Low
1	High
Z	High Impedance
X	Undefined

Table 5.1 Waveform Editor Legends

Test Vector Generation

Test Vectors (or Test Patterns) are the input and output states which represent the logical functions that the DUT is designed to perform. Input and output data are represented by characters-1/0 are often used for input data, L\H\Z for output data and X for no input drive and no output compare.

The test vector sequence is stored in vector memory and each individual vector represents the raw data for one test cycle. The input data is combined with timing, format, and voltage level data and supplied to

the DUT via the pin electronics driver circuitry. The outputs of the DUT are monitored via the comparator circuitry located on the pin electronics cards and compared against the data stored in vector memory at the appropriate strobe time. This type of testing is called stored response because the expected response for the DUT is stored in vector memory.

The test vector sequence may have, in addition to the DUT data, instructions to the test system. For example, timing may be changed on the fly, which means that timing values or signal formats may also be selectively monitored from one cycle to the next. Many test systems also support instructions such as branching, looping, vector repeats, subroutines etc. The way tester instructions appear within a vector file differ from one tester to the next. The key to successful vector generation is thorough understanding of both the DUT and the test system. Notice that the I/O signal data uses all of the vector character: inputs(0/1), outputs(L/H), three state outputs(Z) and don't care or masked (X) conditions. The microcode statements represent additional vector control. In this example, the micro code "Normal" indicates that the vector is executed as a single cycle., the" Repeat 25" statement repeats the vector for 25 cycles. The "Halt" statement terminates the execution of the vector sequence. Micro code statements add great flexibility and power to the test vector patterns, but they are actually part of the test programming language and are dependent upon the target test system.

```
Pattern    (v1)/*file    name*/
 character Definition:
 /* Define what the test system will do for each
 character*/ (
 0 = logic 0     driver on        comparator          off
 /*input*/ 1 = logic 1           driver                on
                 comparator off  /*input*/
 L = logic 0       driver off     comparator            on
 /*output*/  H  =  logic  1                 driver  off
```

comparator on /*output*/ Z = float driver of comparator on /*output*/ X = don't care driver off comparator off /*ignore*/

)

/*Memory Location	Input signals	OUTPUT signals	I/O signals	time set	Micro Code*/ Start_loc:
0	11010	xxxxx	xxxxx	Reset	Normal
1	01010	xxxxx	xxxxx	Reset	Repeat 25
2	11010	HHxxL	01010	Write	Normal
3	01111	LHLLL	LHLHL	Read	Normal
4	11000	LHHHL	10101	Write	Normal
5	01101	HLLLH	HLHLH	Read	Normal
6	11000	LHHxx	11111	Write	Normal stop_loc:
7	11111	HLxxl	HHHHH	Read	Halt End.

The method used to create test vectors are enumerated below:

- Text method
- Graphical method

The options of the above said methods can be selected before entering the test vector creation.

The options selected once cannot be interchanged later. However the changes incorporated in the text or in graphics can be altered and the respective changes will be appended in the other i.e., for any change in text for a particular pattern, the corresponding graphical pattern will change automatically.

Programmer need not work in both modes.

5.4 Standard Test Data Format STDF

Text method creation

With the details entered so far about the PCB under test, the top level function declaration is done automatically for the inputs and outputs with the respective text either in VHDL or in Python.
VHDL text for stimulus only to be created for the inputs of the PCB to be done by the Programmer. This is done using Schematic diagram and the signal path for every input. The predicted responses of the chips in the Board can be seen using Simulator icon and hence the programmer can correct the responses from the simulator and the actual responses from the PCB through learn and compare technique.

Programming is made easy and simple by generating the top level model for the PCB with the available details given as input. The selection of signals and the pattern are only to be entered either in Graphics or in Text using Statements mainly covering conditional statements for the test pattern to be generated.

With the above-mentioned options for PCB setup for programming the programmer is now completely ready to derive the input drive patterns for the PCB connector points.
The Inputs and their default values as entered is now ready for accepting the values driven by the programmer.

Creating Test Patterns

The program for test patterns can be created in Text or in Graphics mode.

In text mode the VHDL and Python Language editor is used for typing the commands.

The Graphical Waveform Editor is used as the direct input to the input drive pins as per the patterns entered by the programmer.

VHDL

VHDL has many features appropriate for describing the behavior of electronic components ranging from simple logic gates to complete microprocessors and custom chips. Features of VHDL allow electrical aspects of circuit behavior (such as rise and fall times of signals, delays through gates, and functional operation) to be precisely described. The resulting VHDL simulation models can then be used as building blocks in larger circuits (using schematics, block diagrams or system-level VHDL descriptions) for the purpose of simulation and testing.

VHDL allows the behavior of complex electronic circuits to be captured into a design system for automatic simulation for design and testing.

VHDL provides features allowing concurrent events to be described. This is important because the hardware described using VHDL is inherently concurrent in its operation.

One of the most important applications of VHDL is to capture the performance specification for a circuit, in the form of what is commonly referred to as a test bench. Test benches are VHDL descriptions of circuit stimuli and corresponding expected outputs that verify the behavior of a circuit over time. Test benches should be an integral part of any VHDL project and should be created in ---------- Comment Line

```
-- Eight-bit comparator
--
library ieee;                          Library statement
use ieee.std_logic_1164.all;           use clause
entity compare is                      entity declaration
port (A, B: in std_logic_vector(0 to
      7); EQ: out std_logic);
end compare;

architecture compare1 of compare is    architecture defines the functioning
begin
    EQ <= '1' when (A = B) else '0';
end compare1;                          end of architecture
```

tandem with other descriptions of the circuit.

The process statement in VHDL is the primary means by which sequential operations can be described.

A process describes the sequential execution of statements that are dependent on one or more events occurring.

A flip-flop is a perfect example of such a situation; it remains idle, not changing state, until there is a significant event (either a rising edge on the clock input or an asynchronous reset event) that causes it to operate and potentially change its state.

Concurrent statements are those statements that appear between the **begin** and **end** statements of a VHDL architecture. This area of your VHDL architecture is what is known as the concurrent area. In VHDL, all statements in the concurrent area are executed at the same time, and there is no significance to the order in which the statements are entered.

Sequential statements are executed one after the other in the order that they appear between the **begin** and **end** statements of a VHDL process, procedure or function.

The interaction of concurrent and sequential statements is illustrated in the example below.

```
architecture rotate2 of rotate is
      signal Qreg: std_logic_vector(0 to 7);

   begin                                          -- Concurrent section starts here
     reg: process(Rst,Clk)
     begin                                        -- Sequential section starts here
    if  Rst  =  '1'  then  --  Async
       reset Qreg <= "00000000";
    elsif  (Clk  =  '1'  and  Clk'event)
       then if (Load = '1') then

            Qreg <= Data;
         else
            Qreg <= Qreg(1 to 7) & Qreg(0);
       end if;
    end if;
     end process;                                 -- Sequential section ends here


    Q <= Qreg;


   end rotate2;

                                        -- Concurrent section ends
                                        here
```

Python TD

Python TD contains mainly performance testing and component level testing. This could be broadly applied to detect failures on PCBs to test any Digital/Analog or hybrid devices mounted on it.

Python TD guides the user for programming of commonly used features of PCB testers at a higher level than a conventional program Editor. The following advantages of using Python Language are listed below:

- For generation of Test Vector for Digital/Analog or mixed Signal components;

- Auto run PCB initialization routine support with conditional loops.
- Defining expected output versus learn and compare with mask at user control
- Alternatively, VHDL simulator for prediction of outputs
- Advanced Online simulation support
- User friendly test sequencing for board repair for each switch over

It also offers much more error checking, and, being a *very-high-level language*, it has high-level data types built in, such as flexible arrays and dictionaries. Because of its more general data types, PythonTD is applicable to a much larger problem domain than *traditional Test Languages/proprietary Test Languages.*

PythonTD is an interpreted language, which can save us considerable time during program development because no compilation and linking is necessary. The interpreter can be used interactively, which makes it easy to experiment with features of the language, to write throw-away programs, or to test functions during bottom-up program development. It is also a handy desk calculator.

Operational Amplifier LM348 Python program for drive pattern:

```
def BLOCK1 ( OUT1,INV1,NI1, dataDir, deviceName, isLearn):
    BeginMacro('BLOCK1')
    amp = 8.0
    amp1 = 8.0
    invamp = -8.0
    noninvamp = 8.0
for tick in range ( 0 ,75): INV1 << amp
        - 0.2
            NI1 << 0.0
            amp = amp - 0.2
for tick in range (75,150): INV1 << 0.0
            NI1 << amp1-0.2
            amp1 = amp1-0.2
for tick in range (150,225): INV1 <<
        invamp + 0.2 invamp = invamp +
        0.2 NI1 << -invamp
    markTestWindow( OUT1, "", 0,224, 1, " )
EndMacro('BLOCK1')
```

Graphical Waveform Editors

In this method the program is edited by graphical means and the language interpretation is not necessary. Programmer has to enter the input drive pattern and check for the output from the PCB dynamically and change the input pattern accordingly if required. The program thus edited is compiled and saved as a file in the software so as to run the Board as and when required.

The graphical editor is interactive to enter the patterns.

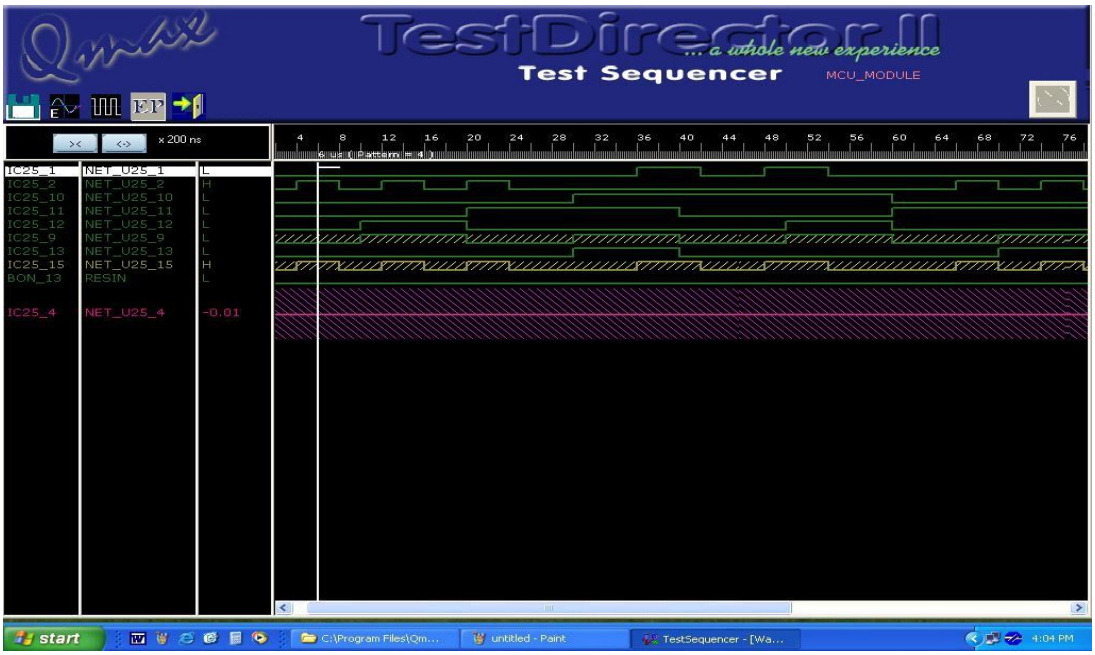


Figure 5.13 Wave form Editor Screen

In the graphical editor both the Digital and Analog waveform generation features are provided so as to generate the required waveform for inputs of the PCB.

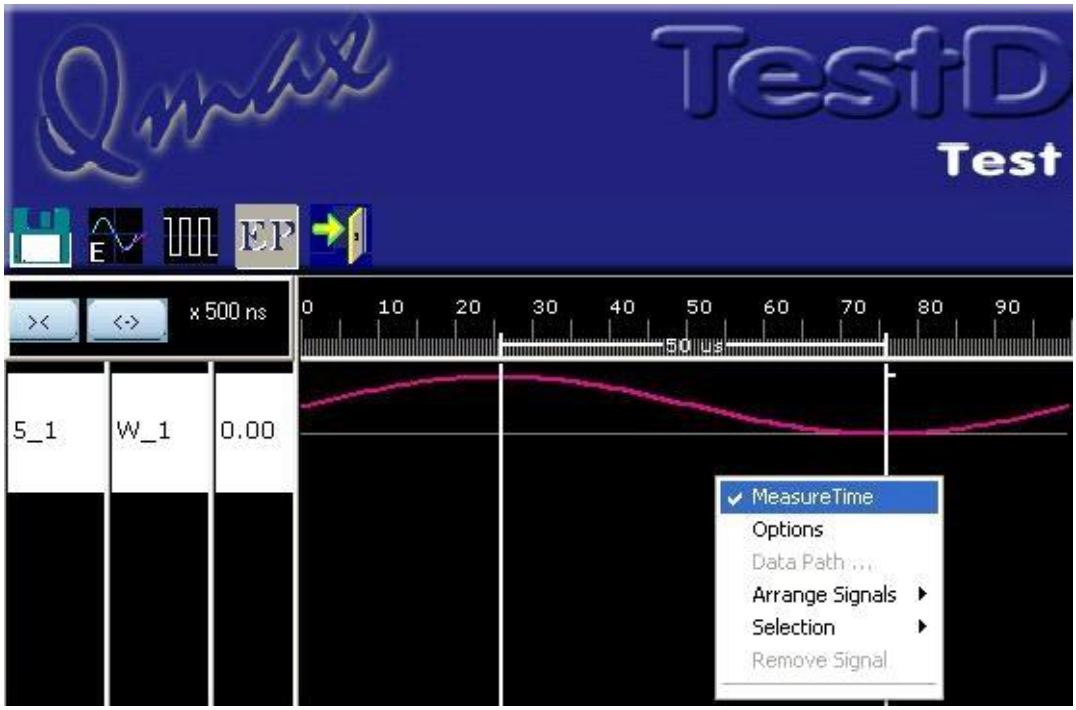


Figure 5.14 Analog signal in Wave form Editor

5.5 Basic of Digital Simulator

Fault Simulation

Regardless of the type of test being performed on the board (or chip or any product, for that matter), it is valuable to know the quality of the test. That is, what percentage of the potential defects that will be exposed by the application of the test? We call this number test coverage. Obviously the higher the test coverage, the better, 100% being the ultimate goal. A higher test coverage means a decreased chance that defective product will be delivered to the customer. Sometimes, particularly for chips and modules, a technique known as fault simulation is used to determine the test coverage number. This technique utilizes a simulation of the logic design, contained in software, wherein potential defects, or faults, are injected into the design and simulated in conjunction with the test stimuli to determine whether or not they are detected. The software program used to perform this analysis is called a fault simulator.

For systems using software simulators, automatic fault insertion by software and fault simulation is obvious, but what happens to systems using expected data defined by the test programmer or it's a learnt data? Obviously automatic fault insertion and determination of expected data with faults can not be determined and in this case the only way of determination of test coverage is to rely on manual insertion of faults onto a known good board.

It has often been impossible to obtain software models of all the components that comprise a PCB. Most PCBs contain several components for which the internal logic design is unavailable,

often due to the fact that it is proprietary information of the manufacturer. Generating a software model for these components can be difficult and often impractical. Without such a model for every PCB component, a fault simulation is not possible by automation.

Fault simulation has often been coupled with some form of automatic test generation. An automatic test generator (ATG) program develops test patterns to be applied to the unit under test, with the goal of achieving a high test coverage. The fault simulator grades the test patterns produced by the ATG, and indicates the test coverage. ATG programs can only function effectively on designs that have built-in

testability structures (e.g., level sensitive scan design). While it is not uncommon for complex chips to have such structures, it is very unusual for an entire PCB to be designed in such a fashion. Thus the use of non-functional testing techniques, and the associated fault simulation, has been very limited at the board level.

Fault simulation also follows the basic testing flow. In addition to the input test vectors, a list of faults that are to be emulated in the Board under test is also applied. The fault simulator emulates each fault (or set of faults) in the list and applies to the test vectors. Each output response is then compared to the associated expected response obtained from a fault-free circuit simulation. If a mismatch is obtained for any test vector, the faulty circuit has produces an erroneous response compared to the fault-free circuit and the fault is considered to be detected. As soon as a fault is detected, most fault simulators will stop simulating that fault (referred as Fault dropping) and restart simulation at the beginning of the test vectors with the next fault in the list being simulated. If there are no mismatches in the output response for the complete set of test vectors, then the fault is undetected. At this point, the next fault is emulated in the BUT and the simulation restarts at the beginning of the set of test vectors.

Once the fault simulation is complete (all faults have been emulated), the fault coverage can be determined for the set of test vectors. The fault coverage, FC is a quantitative measure of the effectiveness of the set of test vectors in detecting faults, and FC can be expressed as FC=D/T. D is the number of detected faults and T is the total number of faults in the Fault list. Obviously 100% fault coverage is desired but in practice, fault coverage in excess of 95% is considered to be acceptable for testing.

In the diagnosis application, a fault simulator is used to generate fault dictionaries. A fault dictionary is a list of faults detected for each test vector. Additionally a fault dictionary may also store the actual output response for each fault, or a compressed version of the response, called a signature of the faults. The diagnosis process(identification and location of the fault) relies on matching the response (or signature) from the circuit under test to the simulated response(or signature) stored in the dictionary.

Creating a Test Program

A test program has to be prepared before starting the device or board test. This process starts with studying the device specification in detail and the device timing. Preparation of a test program flow chart will be helpful.

Review the voltage and current requirements for the device under test. High current devices may require hardware with dedicated power and ground planes. Device pins that require high VDD voltages may need to be isolated from the tester driver and comparator circuitry. Some test systems provide a driven or buffered ground, while others have only a hard ground.

Exceptions

Exceptions may be associated with a particular parameter for a device.. Some parameters may need to be relaxed during testing due to fixture noise. Some device specifications may define the amount by which a given parameter may be relaxed. Additional information may also be given which applies to the specific tests.

Functional Test Timing

When developing a functional test all of the timing defined in the test specifications should be carefully reviewed, and then a test timing diagram should be developed for use within the test program.
It may not be possible to exercise all of the various input timing edges and signal formats or to test all of the AC parameters at one time. Develop a basic timing strategy that will work well for a gross functional test. Once the basic timing is developed you can then begin to add complexity to your timing diagram.

The first step in developing the test timing is to define the frequency (test rate) and placement of the clock and control signals. Next, determine the active edges of the clock or control signals, when input signals are read (latched), when output signals are gated out. Look closely at the delay time parameters for output signals. Make certain that outputs have sufficient time to propagate out before the end of test cycle. It is important to understand the device timing fully.
If the test vectors are being developed from simulation data, review the timing used during simulation. The simulation timing should reflect the device specification timing and must also be compatible with performance of the test system.

Test Program

Begin by entering the device pin to tester mapping. This should include the device pin, signal name, and function(input, output, Bi-directional (I/O) , power) and the tester pin associated with each signal.
Once the definition of each individual pin has been entered, pin groups can be defined. Pin groups are created based on similar characteristics, timings, voltages, currents etc. Once entered, be sure to check your work, this information will be used again and again throughout the development o the test program. Sequence the tests starting from open/shorts to functional tests. This will help to isolate the problem quickly.

Tester Diagnostics

It is important to include a quick tester diagnostic before starting the tests. This diagnostic test ensures the reliability of the tester. A basic diagnostic test can consist of forcing the input reference low supply to 0.8V and the input reference high supply to 2.0 V, then test all pins with the DC measurement system.

A fast functional test of all the pins is also very useful. The driver and comparator circuitry for each pin can be tied together to perform a self-test as an easy way to functionally verify that all pins are working correctly.

Verifying the test setup

Before functionally testing the device, loop on the test pattern and using an oscilloscope, observe each tester channel. Are the timing and voltages correct? Check VDD and ground. Be sure to look at every signal(including the output strobe, if possible).If pull up resistors and current load are required, make sure they are connected and working correctly.

Documentation

The documentation is an important step in test program. Summarize all the test development activity. Make certain to note any deviations from the test specification and reference any supportive documentation(data logs, memos etc).Include the test time and detail any special requirements such as external hardware

Example of Test Vector generation

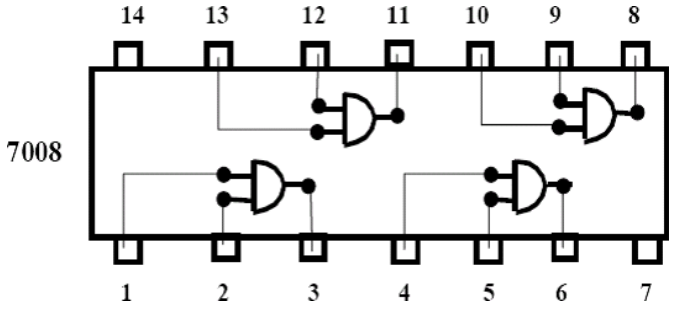


Figure 5.15 (a) Example Of Test Vector Generation

As shown, this IC 7408 has fourteen pins out of which twelve are used for four two-input AND gates and two are used for supplying power to the device (V supply = 5 volts, and Ground = 0 volts).
Let us discuss how the device 7408 is tested. Basically the 4 logic AND gates in 7408 are tested in 4 blocks. Each block tests individual gates in 15 ticks at 2µs time base. The 1st block tests the 1st AND gate (1 & 2 inputs, 3output). For the first four ticks the values of i/p's and o/p's (expected output i.e. theoretical output appears in yellow color) are shown below.

A1 B1 Y1

A1	B1	Y1
0	0	0
1	0	0
0	1	0
1	1	1

Table 5.2 Truth Table

The user can practically realize the above values by either using a text editor or a wave from editor (as shown below) moving the hairline cursor in the waveform window tick by tick.In the waveform editor, the wave form for first (input) signal A1 is constructed as per the table above(0 1 0 1) by creating low ,high, low, high. This completes 4 ticks. Repeat the same sequence for another four times.
For second (input) signal B1,the signal (0 0 1 1) is constructed in the second line as low, low, low, high waveform and repeat it four times.
For (output) signal Y1, the expected signal can be from a software simulator or defined. If defined the signal (0 0 0 1) is constructed as low, low, low, high waveform in the third line and repeated for four times. This completes the construction of waveforms for the test of first block (first AND gate).This is the expected signal for the first block.
Similarly block 2 tests the 2nd AND gate, block 3 tests the 3rd AND gate and so on.

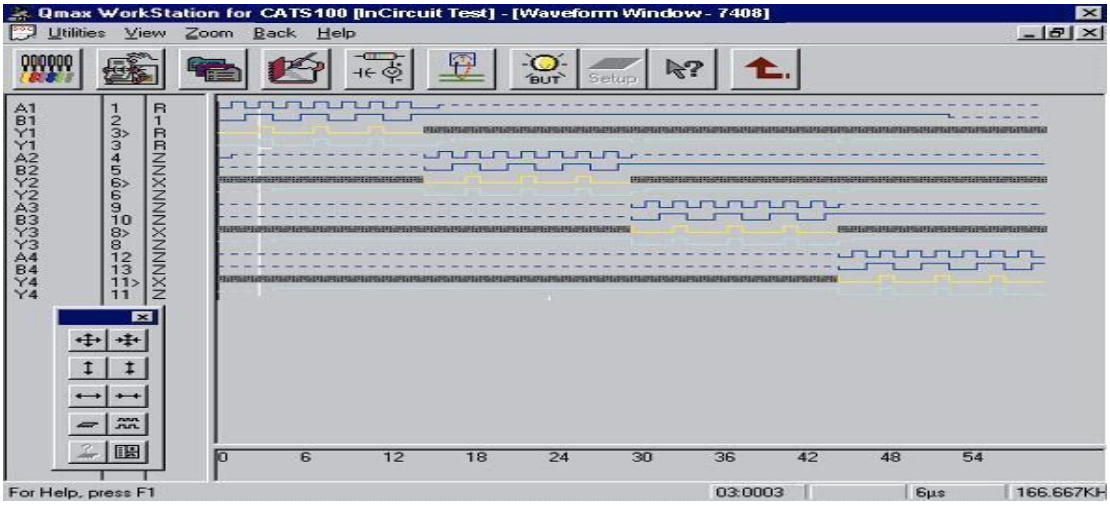


Figure 5.15 (b) expected signal for the first block

This completes the wave form construction for testing one device. If the ATE runs the above test on the device, the test vectors will be driven and the device output will be recorded in the same waveform editor screen on the fourth line. This is the actual output. The ATE will compare the expected (expected value in yellow color) and actual (Actual value in Cyan color) waveforms and if both of them match exactly, then the result is declared as Pass .Otherwise, it is declared as Fail.

Test vector in wave form pattern for a 7408 device

Introduction to Semiconductor Test, Use of Load Boards.

A circuit may contain many active digital devices with possible passive devices (resistor, capacitors) in between them.

Construction of waveforms for a combination of digital devices is a similar process. Only the output of the digital device combination has to be worked out for all the input logic levels and this has to be represented as waveforms in the waveform editor.

Remember a capacitor may introduce a time delay. Then we have to modify the output waveform,(by shifting the output further in time line) .When test program is constructed for a device combination, it has to be evaluated by running it on the devices. If the actual output wave form matches with the expected waveform, then test vector generation for that device combination is correct. If it is not matching, then we have to thoroughly check logic creation steps and correct. Once finalized, this has to be properly named and saved.

Test vectors for the analog devices can be created similarly. Testing of passive components (resistor, capacitor, inductor) is carried out separately using digital multi meter functions of an ATE.

The above process has to be extended for the whole board.

ATEs have provision to save the test programs. They can run the combinations of test programs in any sequence as per our requirement. For example, we can plan a test sequence of testing passive components, analog devices, sections of digital device combinations, and the whole board if required. Once we understand the elaborate process of test vector creation, we need responses from various points in a circuit, to effectively decide the faulty section. This brings in the necessity of Bed of nails test fixture, using which we can introduce testing points on the PCB.

Our aim is to design a test program to troubleshoot a circuit with minimum cost and time and reliability.

Shorts location

We can identify the shorts and opens in a circuit using multi meters and other conventional troubleshooting tools. But it may be difficult to identify the exact component, which creates shorts, especially when the components are connected in parallel in the circuit. And, if a partial short occurs in a component / track, it is highly impossible to detect shorts using conventional tools.

So we have to go for advanced PCB troubleshooting tools such as Short circuit locators. With these tools and an understanding of the circuitry involved, the test

engineer can pinpoint the cause of the fault with a high degree of certainty. A shorts locator comes in handy for accurately locating short-circuits between Vcc-Gnd, hairline shorts and also helps in isolating fan-out problems, etc. Needless to say, it pin points the shorted components or tracks.

Figure 5.16 Pcb Short Locator

Short circuit faults on PCBs are some of the very common but difficult problems faced in PCB production lines and PCB Repair centers. As the density of PC boards has increased, so have occurrences of shorts between traces. These shorts can result from manufacturing errors where the copper has not been

properly etched away and from solder bridges when the boards are assembled. And most of the times, we know that there is a short circuit between two nodes or adjacent tracks in the PCB but do not know where this is exactly located in the PCB.The problem now is to locate the bridge if it cannot be found simply with a magnifying glass. The short circuit locator allows us to do this, even if the short is hidden under an IC or other installed components.
Normally when using the multi meter on the whole shorted track, it will show the same impedance. Because, conventional meters cannot distinguish the milli-ohm ranges. Using Short circuit locator, we can measure the resistance in milli-ohms range. Now keep one probe fixed and move the other probe in either direction.

The concept is, the resistance will increase if we move the probe away from the short and it decreases if we move the probe towards the short. So the exact location of the short is, where we get the lowest resistance value.QT25 is the exact tool that would be very useful to the technicians to pinpoint these faults. It overcomes the need to cut tracks and remove components trying to locate the fault. Before proceeding to understand the actual modes of operation, let us analyze the various reasons due to which the shorts occur in the tracks, components, etc.

Hair line and micro shorts: In highly dense PCBs it is possible that a illegal hair line has formed due to a manufacturing defect. Normally PCB manufacturers test them

electrically with the use of expensive equipment to detect and reject such PCBs. But in small volume production of such PCBs it may not be cost effective in going for a electrical test and hence
this defect may go unnoticed and trouble emerges when these PCBs are assembled and tested for its functionality.

During Component assembly: While the components are assembled and soldered it is possible that a solder bridge between two adjacent component terminals is formed and causes short circuit.

Defective Components: Capacitors, especially tantalum types can get shorted internally and can cause Vcc-Gnd shorts. Semiconductors like diodes or transistors can get shorted fully or partially.

In all above cases it becomes necessary that we need some sort of equipment to locate these defects for fixing them up. QT25 offers this facility in accurately spotting the location of such defects in the PCB, thereby saving time and money.

The two basic modes of operation are:

- Milli-ohm measurement mode (or the Ohm mode)
- The Offset mode.

Milli-Ohm Measurement Mode:

This mode of operation is simple. Suppose you have to locate a short between the suspected tracks, Switch on power and the ohm mode is selected by default. Use the 'Range Selector Switch' on the front panel to select the 200 milli-ohm range. Now using the probe, measure the resistance between the suspected tracks. If the resistance is less than 200 milli-ohms i.e. the selected range, then you may proceed further in the same mode. Suppose the resistance is more than 200 milli-ohm then the next possible range say 2 ohms or 200 ohms should be selected using the 'Range Selector switch' before proceeding further. Also, the tester can then be used to locate the shorts only by activating the 'Offset mode', and the next section clearly describes how the offset mode used for offsetting this higher value. Here, let's assume the resistance measured is 150 milli-ohms. Now try moving one of the probes across the tracks in either direction. The Resistance will increase if you move the probe away from the location of the short circuit and the sound will be of lower pitch consisting of long beeps.

When you move the probe closer to the shorts location, the reading will reduce and the sound will be of increased pitch consisting of shorter beeps. You will get the lowest reading and high pitch sound at the location where the short is found. Some of the examples described below, helps in understanding the tester operation clearly

Consider the following example where there is a short circuit between adjacent pins of the card edge connector, caused by a solder bridge. We can easily locate these kind of shorts with the use of QT25's milliohm measurement. Let's look at figure for illustration.

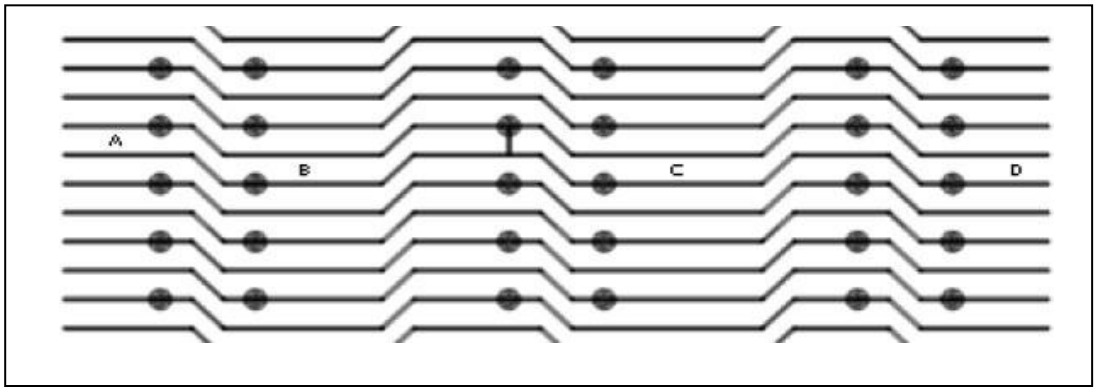


Figure 5.17 Short Circuit Between Adjacent Pins Of The Card Edge Connector, Caused By A Solder Bridge.

Here as you see the short lies between sections B and C. Let's select 200 milli-ohm range. Measure resistance between the tracks at section B. Say, the reading is 100 milli-ohms. Now keep one of the probes stationary. Move the other probe to section A. It may read higher, say, 150 milli-ohms. This means you are going away from the short. Now move the probe closer to Section C. It may read lower now, say, 50 milliohms, indicating you are closer to short. If you move the probe beyond section C it will read higher again indicating you are again going away from the short location. Now move both the probes to the section in between B and C. The spot at which you get the lowest reading indicates the location of the short circuit.

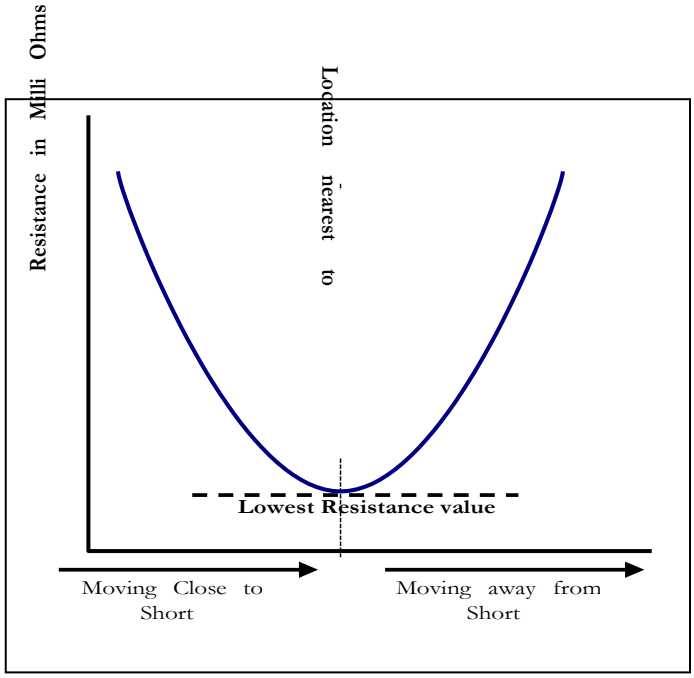


Figure 5.18 Graph For Moving Close Short & Moving Away From Short With Resistance

Needless to say, this is also complimented by the audio tone, which would be of lower pitch and will consist of longer beeps when the probe is away from the location of the short circuit and will consist of shorter beeps with increased pitch when you move the probe closer to the shorts location.

Often times you might have come across problems like the one described below, where short locator is being used to locate shorts between stuck nodes in a circuit. Such a problem could be caused by a faulty driver chip or by an internal or external short clamping the node to a fixed level. The first step in troubleshooting is to determine the reason for the node to be clamped.

The problem could be caused by an internally shorted input, an internally shorted driver output, or an external short to Vcc or ground due to a solder or copper bridge.

In the following example, illustrated in Figure 5.19, the node is being clamped low through an internal short to ground at one of the gate inputs.

In Fig 5.19 the output of U1 is reported bad (shorted to Ground). This output is connected to two other devices U2 and U3.

The possible reasons for this problem could be,

a) U1 output could be bad
b) U2 input could have an internal short, thus pulling U1 low
c) U3 input could be bad
d) None of the devices are bad, but it could be a PCB track short.

Using the equipment, it is easy to find out the actual cause. Measure milli-ohm at pin 3 of U1, pin 4 of U2 and pin 1 of U3 with respect to Ground. The lowest reading indicates the location of the fault. In this case the pin 4 of U2 is internally short. So it will give the lowest reading compared to pin 3 of U1 or pin 1 of U3. The measurement should be made with respect to Ground in this case. If the short is to Vcc then the measurements should be made with respect to Vcc.

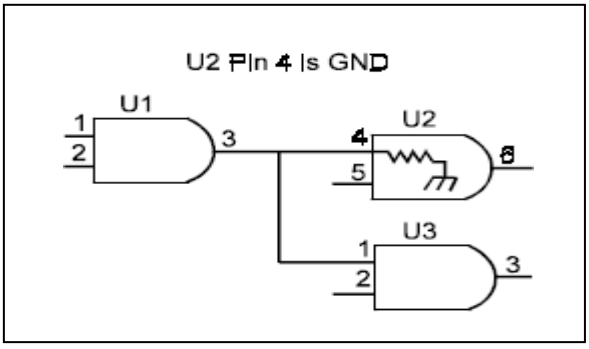


Figure 5.19 Node Is Being Clamped Low Through An Internal Short To Ground At One Of The Gate Inputs.

If the short is partial and has some resistance say up to 200 ohms, this can be offset by the offset mode discussed in the next mode.

Off set Mode:

The offset mode is an extension of the milli-ohm measurement mode because it is used in only in conjunction with the ohm mode.

While the equipment is being used, if the resistance measured is higher than the default range i.e. 200 milli-ohms, the offset button needs to be pressed, to invoke the offset mode. Then the tester automatically changes the range suitably, while offsetting the higher resistance value.

- The display does not show any resistance value during this process, but just displays the message,**' Offsetting'**.
- Once the offsetting is completed, then the message, **'Offset completed'** is displayed.
- However, if the measured resistance range is beyond 200 ohms, then the message, **'Cannot offset'** is displayed.

The actual procedure followed can be better understood with the following example.

In the example described below, a partially shorted de-coupling capacitor causing a Vcc-Gnd short has be located using QT25's milli-ohm measurement option.

Normally, by measuring the milli-ohm across each suspected capacitor we can find out the short. Set the QT25 for ohm mode and at 200 milliohm range. Measure the resistance between Vcc and Gnd tracks. Move the probe pair across each one of the suspected capacitors one by one, and the capacitor where you measure the lowest resistance is the shorted one. This is the normal practice that would be adopted by QT25 in trying to locate the shorted capacitor.

But in this example the capacitor happens to be a partially shorted capacitor, and so the resistance measured across may not fall in the milliohm range, and it could fall in the higher range. Say, the de-coupling capacitor C4 is the culprit, and exhibits a resistance of 100 ohms. Hence we cannot use the 200 milli-ohm range and we have to select the 200-ohm range. Now, measure the resistance between the shorted tracks, and lets assume the reading to be 100 ohms.

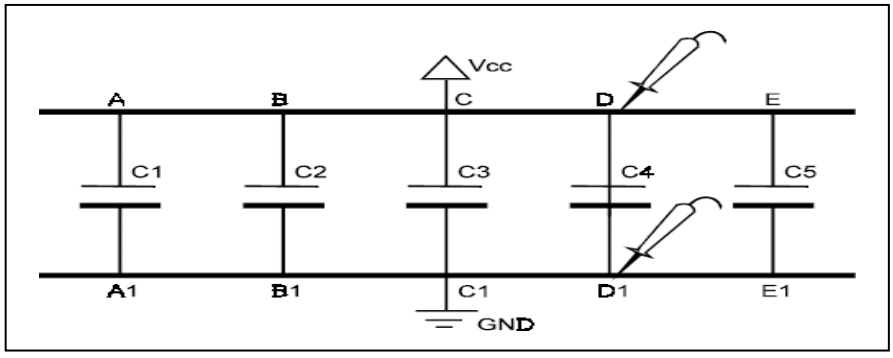


Figure 5.20 press and release the Offset switch

Now holding the probes firmly on the same spot, press and release the Offset switch. Wait for the tester to stabilize and display a value which would be near zero. The set-up is now ready for locating the short, as done in the ohm mode.

Now, moving the probe away from the short location will show an increased resistance value in the LCD, along with low-pitched sound consisting of long beeps. Moving closer to the short will show smaller or negative value in the LCD. The short location, in this case the shorted capacitor is the one at the spot, where you get the lowest value in the LCD display and highest pitched sound consisting of short beeps.

Short Locator equipment will be very handy and useful in field application for finding shorts in PCB and cables etc. effectively.

This equipment will be useful in locating shorts in PCB tracks and Cable Shorts etc., in industrial applications. It can be used in low resistance measurement comparison in electronics applications in board.

UNIT-5
REVIEW QUESTIONS

PART A (2 MARK)

1. Define TCK and TMS.
2. What are TDO and TDI?
3. What is Test Fixture?
4. Define the term Load-Board.
5. What is simulator?
6. Define Package Test.
7. Define VDDmin.
8. What is Wafer Test?
9. What is the purpose of input leakage test?
10. Where (IOZ) high impedance current test is performed.
11. What preconditioning is required to make the IOS (output short circuit) test?

PART B(3MARK)

1.Define Clock Pin Termination.
2.What is automatic test pattern generation?
3.Define Clips and mention some of its type.
4.Write short notes on Manual Testing?
5.Define clip-on test versus whole board functional test.
6.Whether board integrity can test using ICFT?

PART C (3MARK)

7.What is ISP? How its helps in hardware design change implementation.
8.Explain the concepts of fault simulation and fault dictionary in reference to test program set.
9.What are the prerequisites for developing an effective test program to test a PCB module?
10Name two parameters which fall under DC/AC parametric measurements and explain.