TEST METHODOLOGIES OF VLSI CIRCUITS USING SCANNING ELECTRON MICROSCOPE

A Thesis
Presented to the Department of Electronic Engineering
and the Graduate School
of the Chinese University of Hong Kong
in Partial Fulfillment of the Requirements
for the Degree of
Master of Philosophy

by
Chan Lap-kong
June 1994

ABSTRACT

E-beam testing for VLSI circuits has already been a well-established technique. The observability of circuits is increased because internal circuit nodes can be used as test points. The flexibility of making use of these internal circuit nodes requires the development of new ATPG algorithm. Previous reported algorithm does not guarantee full fault-coverage and assumes all internal circuit nodes are test points. The new algorithm described in this thesis will generate a full fault-coverage test set for a fan-out free combination circuit. The main characteristic of the algorithm is that it generates test vectors as well as probe points. As a result, the probe points are different for each test vector, and the number of probe points is the minimum for the test set generated. Results obtained show that an average of 30% test vector reduction is achieved compared with the conventional testing method which uses only output pins as test points.

ACKNOWLEDGEMENTS

The project is supported by the Croucher Foundation grant reference number CF 92/01/01. The author of this thesis would like to express his deepest gratitude to his supervisor, Dr. Oliver C.S. Choy for his supervision over the whole research. His guidance, advice and support really help the author much to overcome the difficulties encountered.

TABLE OF CONTENTS

Page

LIST OF FIGURES

1. INTRODUCTION

1.1 Background

This work is concerned with test methods for combination logic networks. Test methods can be classified into the external test method and the internal test method. An external test is defined as a test where the device-under-test(DUT) can only be accessed through its primary input and output pins. Test vectors are fed into the DUT through its primary inputs(PIs) and the result is observed through its primary outputs(POs). This is the normal situation with conventional automatic testing equipments(ATEs). An internal test is defined as a test where the DUT can be accessed through its PIs, POs and internal test points. The internal test points are contacted through a bed of nails or a movable probe. This is the normal situation with conventional mechanical probing, CrossCheck design technology[1, 2] and e-beam testing[3, 4, 5].

The external test method relies on the terminal characteristics of the device. However, the internal test method has considered both terminal characteristics and internal characteristics of the DUT. With internal test methods, the circuit becomes less complex and the device is more easy to test.

CrossCheck design technology and e-beam testing are common internal test methods. Both test methods improve the observability of the DUT by providing the ability to access internal nodes. The CrossCheck design technology has an embedded test structure which can access internal test points of VLSI circuits by memory-array like addressing scheme. An e-beam tester can measure the logical value of internal signal lines running in the top-level metal of the circuit. Internal test methods can simplify the testing of VLSI circuits, so many current research are concentrated on this subject.

The conventional test generation and fault simulation algorithms are based on the external test method. Commercial software tools for automatic test pattern generation and fault simulation do not fully support internal test method. For testing VLSI circuits, the following problems arise: the number of test vectors is large, and the computation time is long.

As the complexity of VLSI technology increases, an e-beam test system becomes an indispensable instrument for probing internal behavior of VLSI circuits[6, 7]. However, it is also necessary for a test generation algorithm to account for such a new test situation. It is assumed that all internal signal lines of the circuit can be observed in dealing with the test generation problem. Of course, not all the internal nodes are required as test points. Hence, an e-beam test generation algorithm and a test points selection algorithm are needed in e-beam test systems.

1.2 Problems in Testing VLSI Circuits

The major problem in testing VLSI circuits is the high test cost. The test cost is high because the test-cost-per-gate and tester cost are increasing. This problem is more serious in testing Application Specific Integrated Circuits(ASICs) since the design cycle is short.

1.2.1 Test-cost-per-gate:

Test-cost-per-gate increases because both the test time and the test generation cost increase faster than the gate count.

The gate count per chip in VLSI circuits is increasing, but the increase in the number of input and output pins is limited by the constraint of bonding pad and the perimeter of the chip. The inadeque number of input and output pins leads to more test vectors and longer testing time. As a result, test-cost-per-gate for VLSI circuits increases.

With limited number of test input and ouput pins, the complexity of test pattern generation goes up rapidly for large circuits. The test pattern generation time is longer. Computers are getting faster and cheaper and this contributes an improvement in the test pattern generation cost. However, the high test-cost-per-gate is still a serious problem in testing VLSI circuits.

1.2.2 Tester Complexity:

The number of input and output pins is large in VLSI circuits. Large number of test pins and high flexiblity of the programmable pin electronics are basic requirements of the tester. Moreover, since the number of test vectors is large, the tester should have the capability to handle a large database efficiently. The tester technology becomes more and more difficult to catch up with the VLSI technology. As a result, the VLSI tester becomes expensive.

1.3 Tester Based on Terminals Characteristics - Automatic Testing Equipment(ATE)

The conventional automatic testing equipment(Figure 1) is an external test method. Test vectors are fed to the DUT through its primary inputs. A fault unit can be determined by comparing the output responses of the DUT and a golden unit. The traditional testability design has been based on improving the controllability and observability of a circuit. By incorporating the testability design(e.g. Level Sensitive Scan Design), test vectors are fed serially to the IC through an input pin and the results are received at an output pin.

The performance of ATE had historically lagged behind that of newly developed devices. With the VLSI circuits, the tester should provide a sufficient number of I/O pins and testing speed. The tester development also depends on the pin-electronics and test fixture since these two technologies will affect the accuracy, flexibility and speed of the tester[8]. For high gate density integrated circuits, the traditional ATE system will not be a good tester in the future.

ATE belongs to testers based on terminals characteristics. The testers based on terminals and internal characteristics are discussed in the following sections.

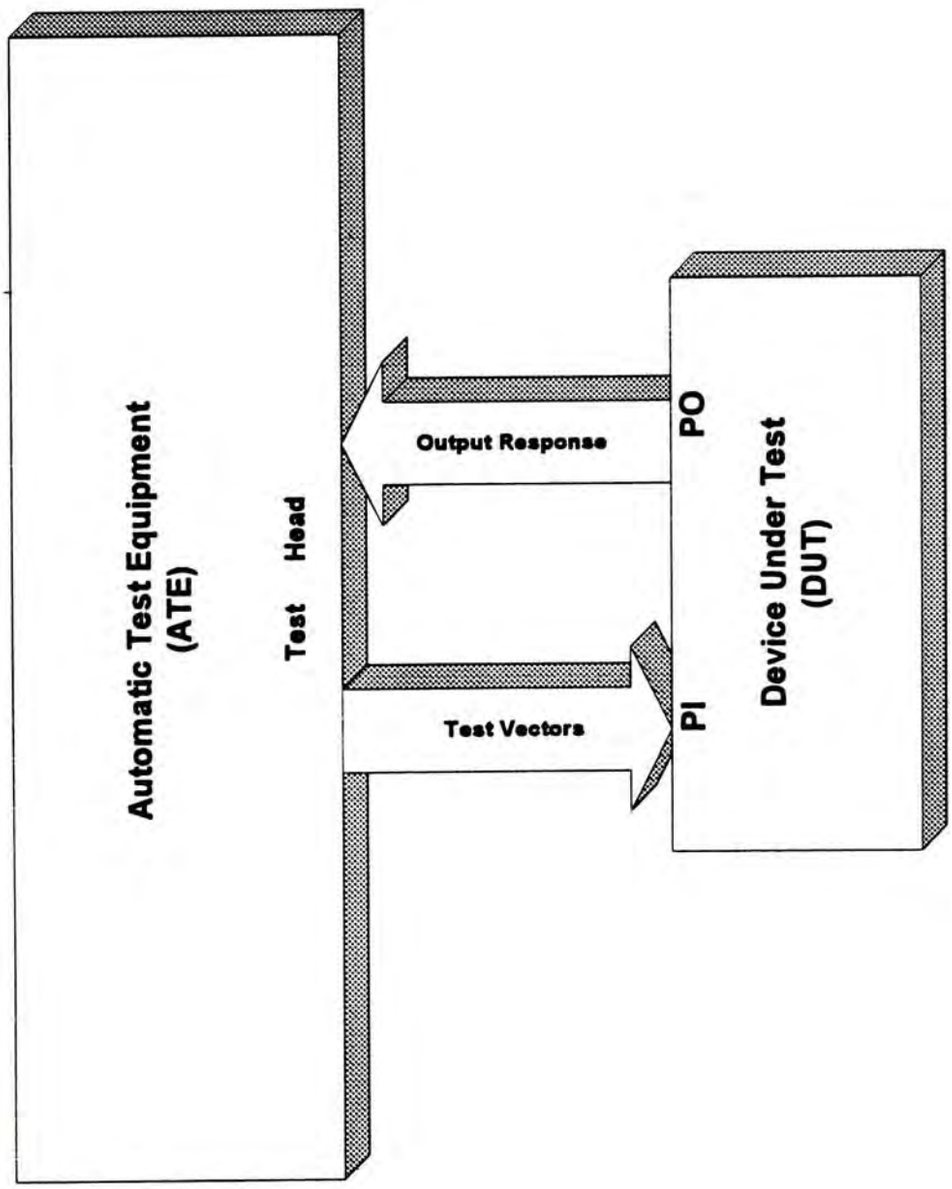


Figure 1 A Conventional ATE

1.4 Tester Based on Terminal and Internal Characteristics

VLSI devices have high circuit complexity. It is difficult to locate faults simply by measuring terminal characteristics. Some forms of internal testing are necessary.

Traditionally, the mechanical probing method has been employed to locate faults of ICs. With the development of the scanning electron microscope, e-beam probing method becomes increasingly important. These two methods allow individual components to be exercised. Voltages and waveforms can be measured on individual metal tracks. Let us have a brief discussion on these two methods.

1.4.1 Mechanical Probing Method

Mechanical probing is a traditional diagnostic technique to locate faults of IC during the development phase of IC. However, it has some drawbacks.

The small feature size of circuits must be compatible with the size of a micro-probe. With the advances of VLSI technology, the feature size is about 1 μm for bulk production. The diameter of a mechanical microprobe is about 3-5 μm. It becomes rather difficult to make reliable contact as VLSI circuit size shrinks. Micro-probing of internal tracks will become impractical.

The circuits become more susceptible to capacitive loading as the line geometries shrink. The speed of circuits is continuously increasing. It will require a very large measurement bandwidth to measure electrical performance of a gate. The introduction of stray capacitance on signal lines will introduce error to the measured signal.

Because of the small size of circuit elements and the capacitive loading effect, mechanical probing method is in practise not suitable to measure internal signals of high density integrated circuits.

1.4.2 E-beam Probing Method

In the failure analysis of integrated circuits, the conventional mechanical probing can be replaced by the electron-beam probing. In a scanning electron microscope(SEM), the position of an electron beam can be easily controlled by varying the voltage of the deflection coils. E-beam testing has higher spatial and spectral resolutions than mechanical probing. It is a non-destructive contactless technique which does not introduce capacitive loading to the device. Hence, it is a good candidate for the testing VLSI circuits. SEM is a mature quantitative measuring equipment and will be discussed in chapter 2.

1.5 Movitation of this Research

The advances in processing techniques have pushed device dimensions into the sub-micron region. Correspondingly, the design complexity of integrated circuits also reaches the ultra-large scale. With such a complexity, the testing cost of an IC is escalating. It is now not surprised to have a design which is impossible to be satisfactorily tested because of the excessive testing cost.

With e-beam testing, one has almost total freedom to observe any internal circuit nodes. This has already been used to its best advantage in failure diagnosis[9]. The same technique can also be applied in production testing.

Recent research and development in e-beam testing is either hardware oriented or software oriented. The main hardware problem is how to construct an efficient VLSI e-beam testing system. The software problem is how to make use of the internal test points in the e-beam test pattern generation.

Conventional automatic test pattern generation techniques are designed for ATE testing. It is based on the assumption that responses are observed through a fixed number of output pins. For e-beam testing, many internal test points are available, and their number and locations per test vector are arbitrary. Current research in e-beam test pattern generation algorithm is based on the modification of conventional Larger Scale Integration(LSI) test pattern

generation algorithm. E-beam testing will require a new ATPG algorithm to take into account the fact that test outputs are no longer restricted to the output pins. The correlation between each test vector and the corresponding probe points is a new research area in the automatic test pattern generation for e-beam test systems.

Most test generation algorithms need a fault simulator to verify and evaluate the test vectors generated. However, there isn't any commerical fault simulator which can be employed to accomodate arbitrary observation points for each test vector. In order to tackle this problem and improve the efficiency of the test generation, a new and better method to generate test is required. Unlike conventional test generation algorithm, critical path tracing[10] generates test vectors together with faults detected. This algorithm generates critical paths from primary outputs toward primary inputs. Lines on the critical paths have critical values which determine the detected faults. Hence, the algorithm can generate test vectors without a fault simulator. This advandage makes it a suitable basis for a new test generation algorithm specific for e-beam testing.

In order to take the advantage of the possibility of aribitrary probe points in e-beam testing, a new strategy to select probe points is required. With the critical path tracing algorithm and the probe points selection algorithm, multiple critical paths can be created and the fault coverage for each test vector is increased. As a result, the number of test vectors is decreased.

In this research, a e-beam test generation algorithm was developed to generate test vectors with corresponding test points. The algorithm makes sensible selection of test points for each test vector instead of all internal nodes.

1.6 Outline of the Remaining Chapters

In the next chapter, we begin with a brief introduction to the configuration of an e-beam tester. With voltage contrast mode operation and beam vector rastering method, an e-beam tester can be used to observe internal test points of integrated circuits.

Chapter 3 describes the basic principles for the e-beam test generation algorithm. It begins with an introduction to the single-stuck-at fault model and the standard netlist format ISCAS85. Algorithms to find levels and reconvergent fanouts of digital circuits for preprocessing steps of test generation are discussed.

We reviews conventional test generation methods in chapter 4. The conventional e-beam test generation algorithm, the Kinch's algorithm, is discussed.

In chapter 5, we proposed that e-beam testing should be applied in the wafer stage of IC production. Assumptions and rules for the e-beam test pattern generation algorithm are introduced. Critical paths generation algorithm and probe points selection algorithm are presented with examples. An evaluation routine to reduce redundant potential probe points is introduced. The generation of critical paths at a fanout site is also considered.

Examples of the application of the algorithm on simple circuits are reported in chapter 7.

Conclusions and suggestions for further research are discussed in Chapter 8.

2. E-BEAM TESTER

2.1 State-of-art of E-Beam Tester

Electron-beam testing offers a viable alternative to ATE. Electron-beam testing relies on the voltage contrast phenomenon which occurs when an electron beam is incident on metal tracks with different potentials(Figure 2). A track with higher potential will deflect more secondary electrons from the detector thus the image obtained will appear less bright. In other words, tracks with different potentials will have different brightness. Therefore, one can resolve the voltage of a circuit node inside an IC.

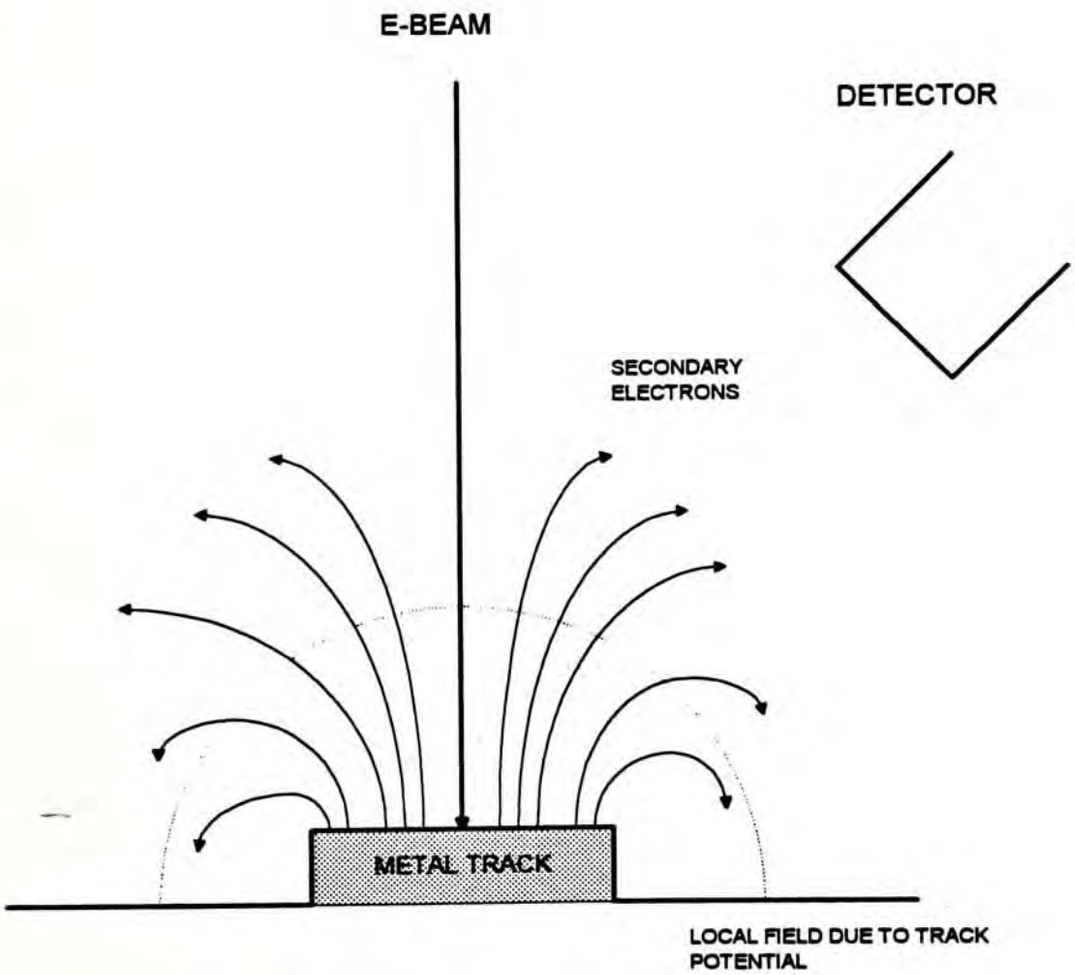


Figure 2 Voltage Contrast Phenomenon

E-beam tester(EBT) is similar in construction to a conventional scanning electron microscope(SEM) as indicated in figure 3. A SEM is employed to observe the structure of semiconductor devices in the electronic industry. An EBT is used in the very-large-scale IC testing. The differences lie in the design of the electron-optical column and of the automatic computer control. In an EBT, LaB_6 cathode allows the electron optics to be optimized for 0.5-1.5 keV and 2 nA beam current working conditions. Automatic positioning of the probe point is a common feature in an EBT. For high speed ICs, fast beam blanking systems are available with 5 ps pulse width to attain GHz frequency measurement[11].

In this chapter, we will focus on hardware components of an EBT.

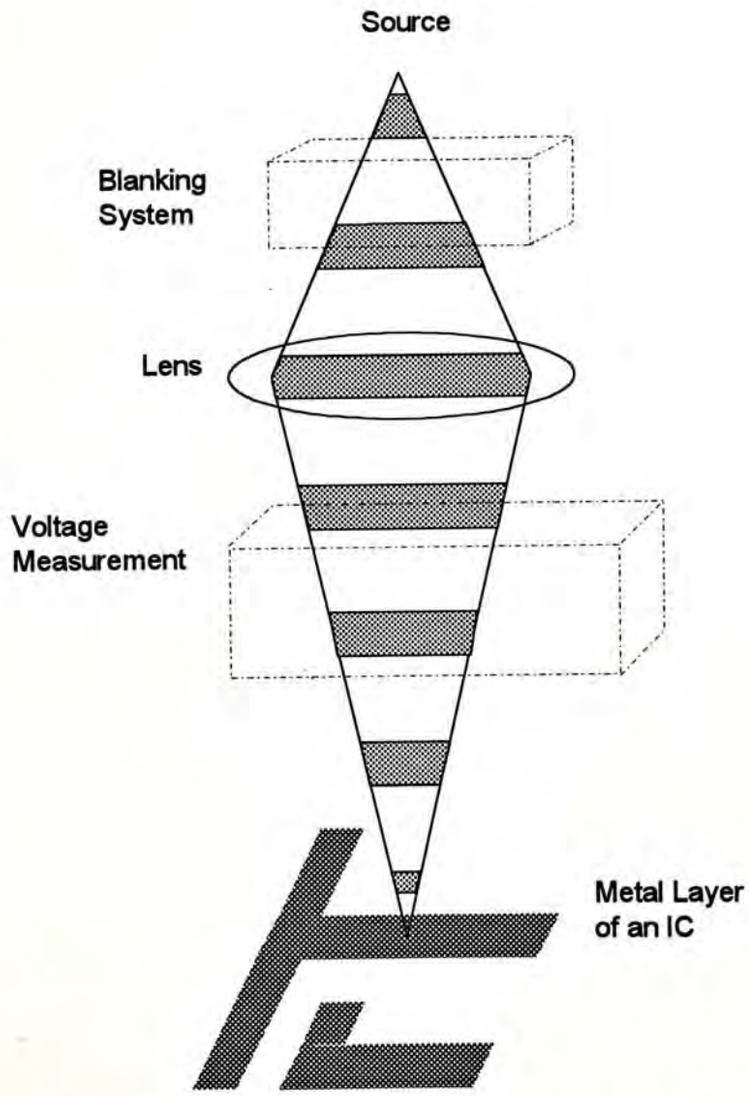


Figure 3 The Principle of an EBT System

2.2 An Electron-optical Column of a SEM

A SEM consists of two major parts: an electronics interface and the electron-optical column. An electronics interface allows the digital control of the e-beam probe, e.g. its focusing, intensities, location and beam current. The interface will be discussed in section 2.5. Inside the electron-optical column, an electron beam is generated and focused to a small spot. The beam is scanned across the specimen to generate an image on the viewing screen.

The electron-optical column consists of an electron source, an accelerating anode, a beam blanker, electron lenses, a scanning system, an objective aperture, a specimen chamber, a detector and a vacuum system[12].

Since electrons travel only very short distances in air, the column should be kept at vacuum. The vacuum is produced by an oil diffusion pump backed by a turbomolecular pump and an ion pump. The oil diffusion pump uses a stream of hot oil vapor to strike gas molecules in the vacuum and expels the gas molecules from the system.

At the electron source, electrons are emitted from a LaB_6 cathode. The anode accelerates the electrons to a low energy about 1 keV. A beam blanker is used to stop the electron beam when moving between two inspection locations. Electron lenses are used to focus the electron beam to a small spot. An image is formed on the screen when the beam is scanning across the specimen. The scanning mechanism and the use of beam blanker will be discussed in the next section. The function of objective aperture is to limit the angular width of the electron beam in order to reduce the lens aberration effects and to improve the depth-of-field in the image. The specimen chamber contains an adjustable specimen stage and an electron detector. The detector is sensitive to the number of secondary electrons(SEs) collected.

2.3 Beam Rastering Methods

To display an image on the screen, the electron beam on the screen of CRT should move in synchronism with the electron beam scanned across the specimen surface inside the electron column. When the detector in the vicinity of the specimen receives more secondary electrons, the beam in the CRT appears to be more bright. When the beam reaches the end of a line, it is blanked, and it is moved to the beginning of the next line. This is repeated line by line. Consequently, an image is displayed on the screen of CRT. This is the conventional beam rastering method for SEM imaging.

If the scanning and blanking mechanism is controlled by a computer, a vector scanning method can be employed for e-beam testing. In this technique, the beam location is determined by voltages applied on the deflection coils. The detector gives the voltage of this location. During the movement of the beam from one location to another, it is blanked. In e-beam testing, only a small number of internal signal lines needs to inspect. The vector scanning method is employed because there is no need to scan across the whole image.

Hence, the electron beam acts as a movable probe which is fully controlled by a computer through a PC-SEM interface. For the PC-SEM interface, please refer to section 2.5. For ICs testing, the voltage contrast mode operation of SEMs is used and is introduced in the next section.

2.4 Voltage Contrast Phenomenon

In voltage contrast mode, a SEM can produce a voltage contrast image. The voltage contrast image shows us the voltage distribution inside the internal lines of an IC. It is traditionally used for visual fault diagnosis of ICs. In figure 4, a voltage contrast image shows the central track is connected to 5V, a track on its adjacent right is grounded and other tracks appeared in the image are not connected.

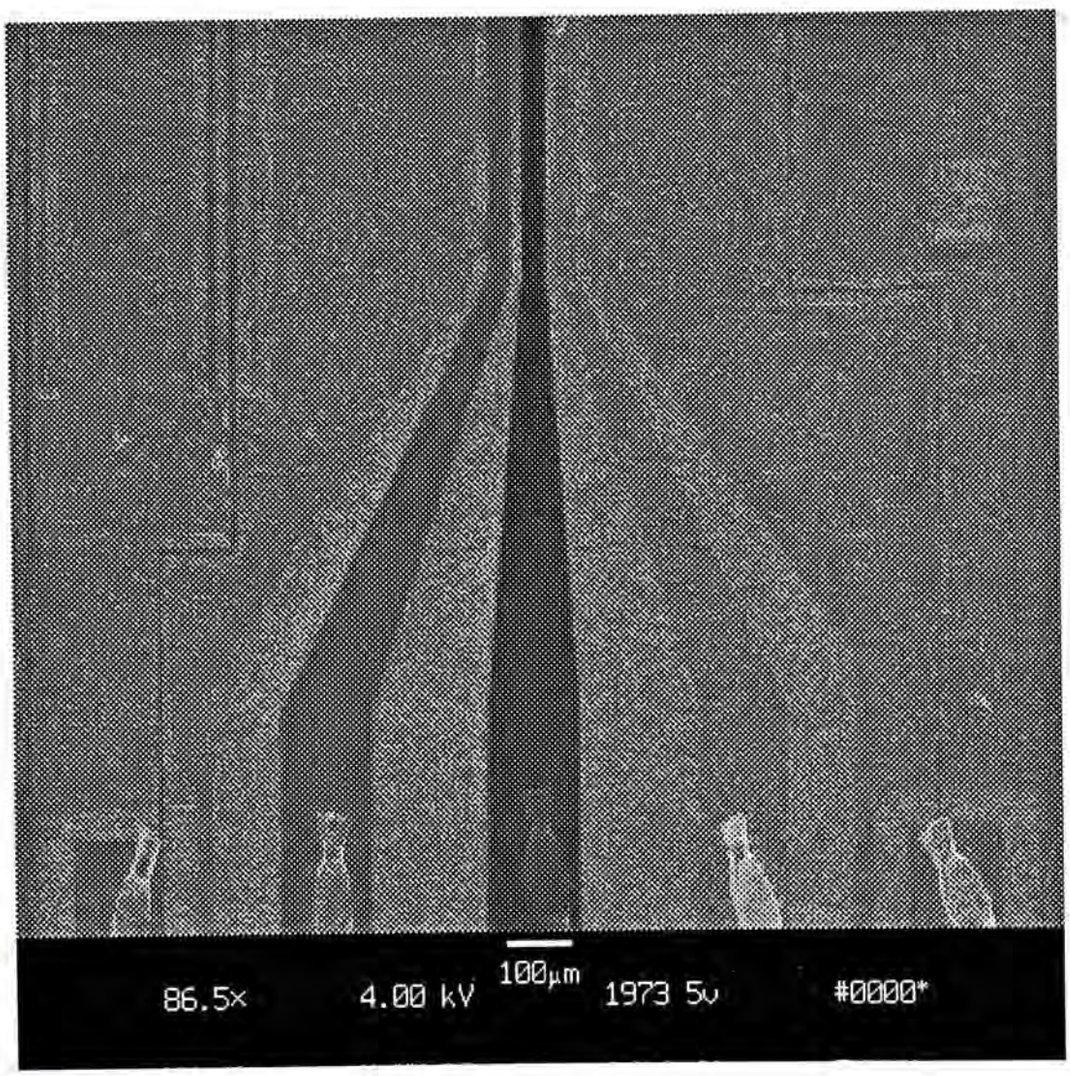


Figure 4 A Voltage Contrast Image

In voltage-contrast testing, a ray of low keV primary electrons beam is generated from the electron gun and focused on the specimen surface. The beam is rastering scanned across the specimen. Primary electrons lose their energy during bombardment. These electrons, that gain sufficient energy to against the work function, will escape from the specimen surface and enter the secondary electron detector. These secondary electrons are relatively low in energy, so their trajectories are easily influenced by electric fields. More secondary electrons are collected in region of low potential than that of higher potential. It is because some of the emitted secondary electrons are attracted back to the surface(Figure 5). The result is a voltage contrast image: the region of positive potential appears black and the region of zero potential appears bright.

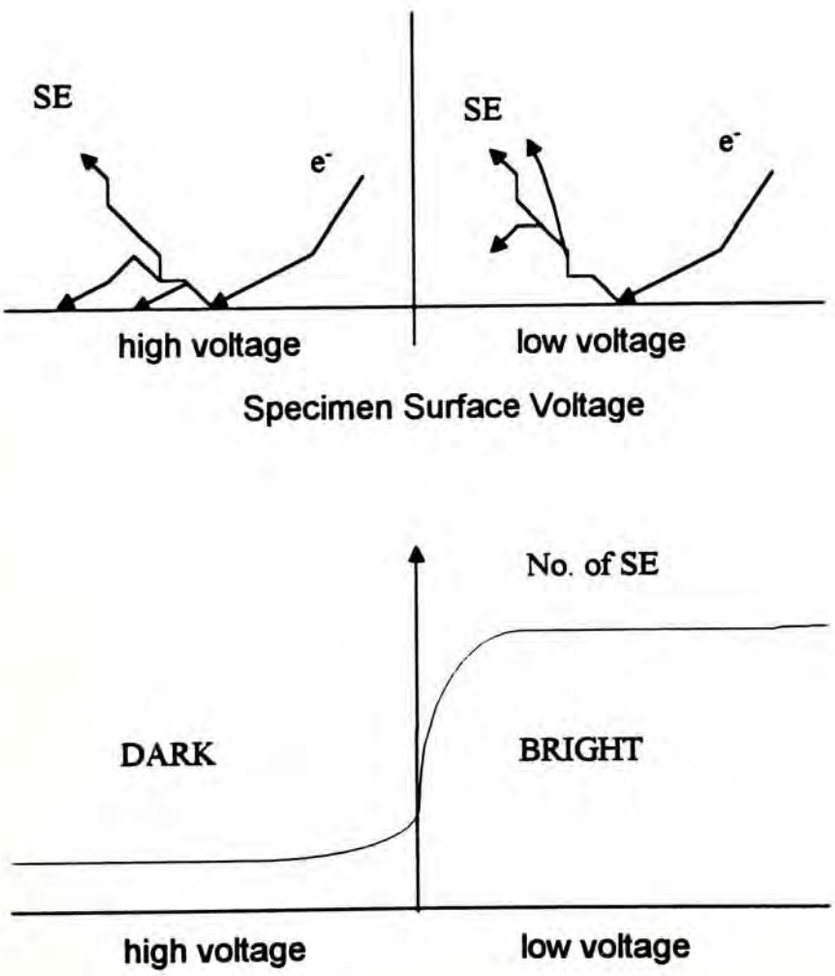


Figure 5 Voltage Contrast Phenomenon

This secondary electron energy forms the basis of quantitative voltage contrast. Figure 6 shows the variation of energy distribution with specimen voltage. The secondary electrons always have the same energy distribution with respect to the sample. With negative specimen voltage, the distribution shifts to higher energy and the secondary electrons possess higher kinetic energy. As a result, more secondary electrons are collected by the detector. The shift of secondary electron energy distribution by different specimen voltages acts as a basis for voltage contrast analysis.

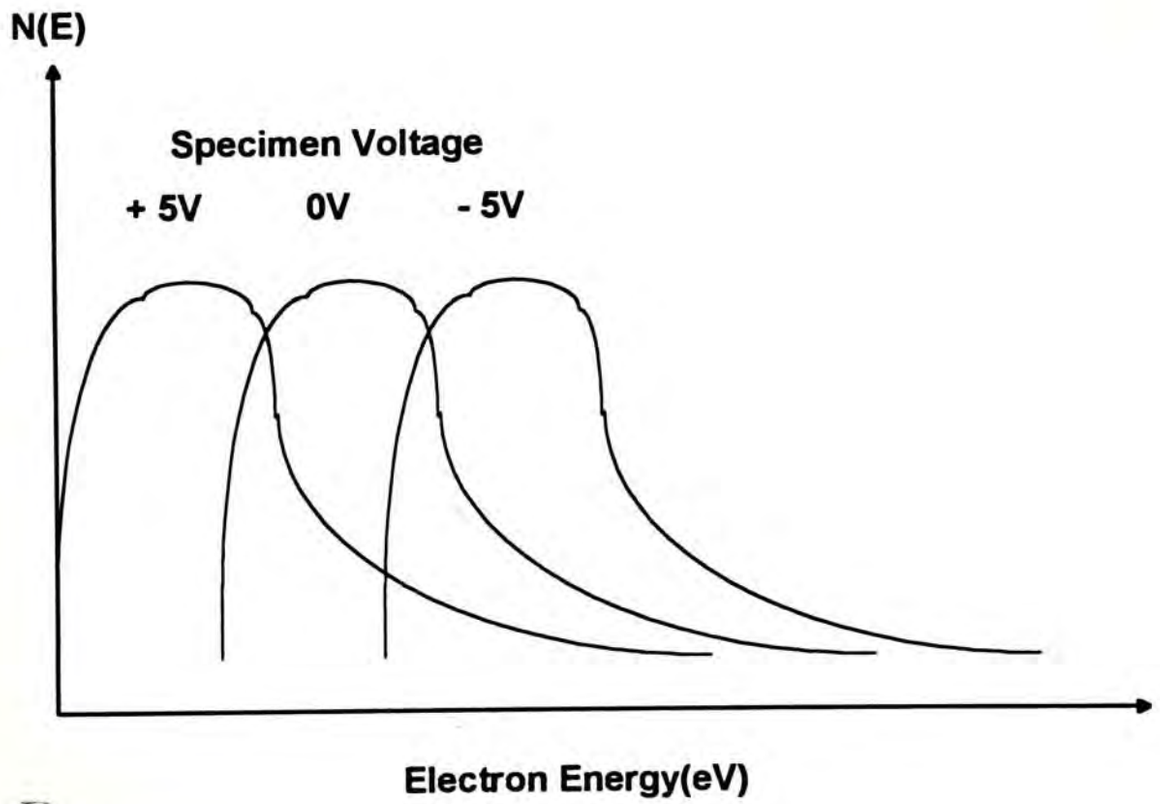


Figure 6 SE Energy Spectrum Versus Specimen Voltage

Voltage contrast signal is very sensitive to electron beam energy, specimen tilt angle, local field effect, detector configuration, ... etc. Hence accurate measurement of voltages is difficult. In our application, precise measurement of the potential at a particular point of the specimen is not required. Before taking any measurement, a calibration is performed to distinguish high potentials from low potentials.

Hence, the internal signal lines of a VLSI circuit is observed by collecting secondary electrons which are generated by the bombardment of primary electrons at nodes of interest. This non-contact, non-loading probing makes it possible to observe the internal condition of a working IC.[13, 14]

If the probing is control led by a computer, an e-beam tester is formed. In the next section, a computerized control SEM is introduced.

2.5 Configuration of an E-Beam Test System

In the past, SEM images are viewed in the CRT. Analog image processing of intensities is now often replaced by digital operations of a computer. An e-beam test system is a computer controlled scanning electron microscope.

The test vectors are generated from a computer. These vectors are applied to the specimen through a cannon plug. An electron beam simulates a probe with very high impedance and low capacitance. The computer controls the electron beam to probe internal signal lines instead of scanning across the whole device. The internal signal lines are measured when the IC is in working condition[15, 16].

Figure 7 shows the overall block diagram of an e-beam tester. It consists of a computer, a PC-SEM interface[17, 18], and a SEM. The PC-SEM interface(Figure 8) provides a communication link between SEM and PC to enable SEM control and data capture.

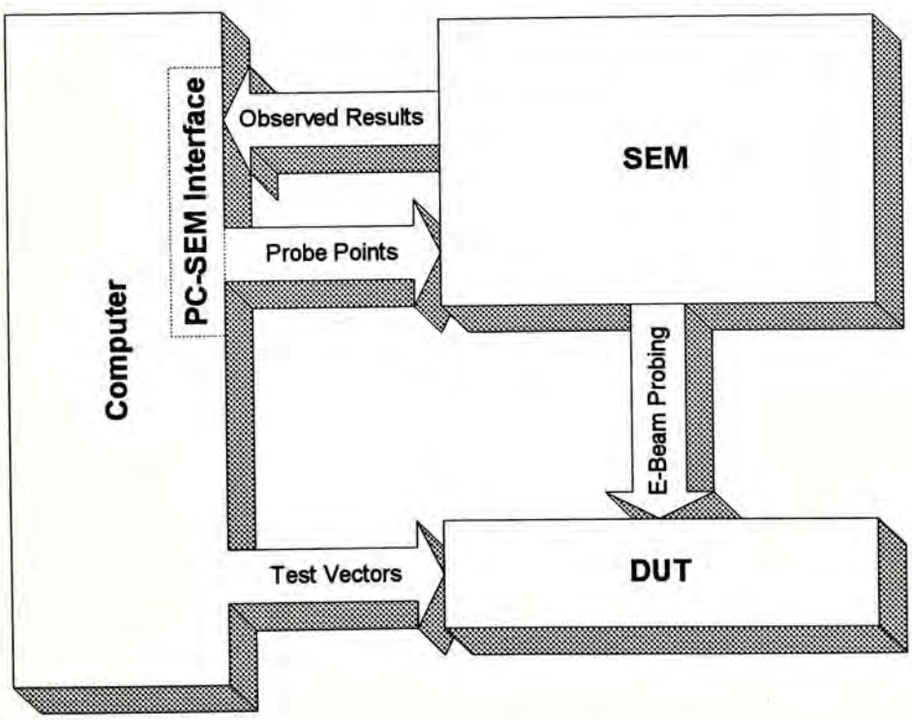


Figure 7 Configuration of an E-Beam Tester

In the PC-SEM interface, a beam blanker signal blanks the beam during the beam positioning. The beam position is determined through a digital-to-analog(D/A) converter. The electrical signal from the SE detector undergoes gain control and analog-to-digital(A/D) conversion. A digitized gray-level signal of the probe location is then further processed by the computer.

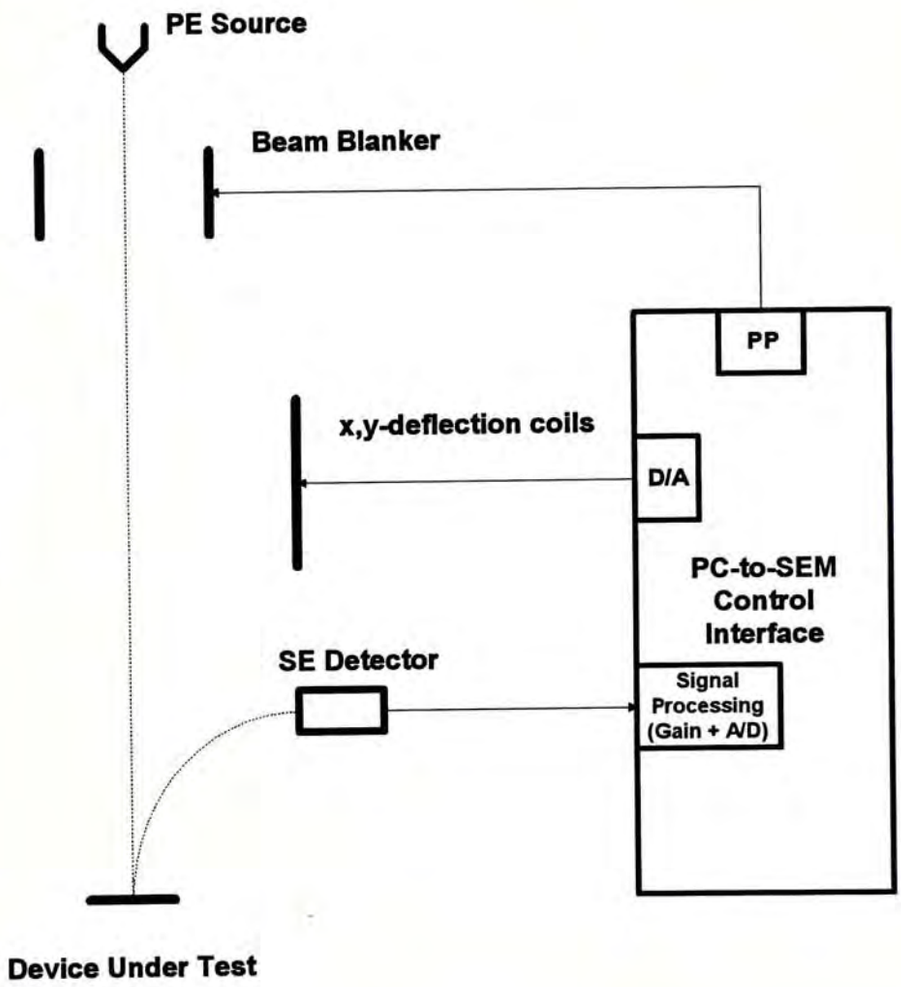


Figure 8 A PC-SEM Control Interface

2.6 Advantages of an E-beam Tester

In testing integrated circuits, an e-beam tester has the following advantages:

1. It is a non-destructive and non-capacitive loading testing method.

2. An integrated circuit can be tested under normal working conditions. Short primary electron pulses are generated by a blanking system synchronized to the test signal within the integrated circuit. The e-beam is directed to the point of interest and secondary electrons emitted are collected. The voltage measurement module determines the logic value of the line.

3. The testing process is computerized. With a PC-SEM interface, the position of the electron beam is determined by test points of the current test vector. The voltages of the points of inspection are further processed by the computer.

4. The testing cost is reduced. The observability of the circuit and hence the fault coverage of each test vector is also increased. Consequently, there is a decrease in the number of test vectors. As a result, the cost of test pattern generation and the testing time are decreased.

The e-beam tester is a mature qualitative voltage measurement equipment. It has been widely used in the fault diagnosis of ICs. It's hardware is improving in order to become a production tester. At the same time, a test pattern generation software, which fully support e-beam testing, is urgently required.

3. BASIC PRINCIPLES

3.1 Single-Stuck-At Fault Model

For computational simplicity, a fault model is often used to represent physical defects in digital circuits. Intermittent and transient faults are very difficult to model. Only permanent faults are considered here. A convenient approach is to assume that only one type of fault is present at a time. Single-stuck-at model is the most popular model for fault simulation and test generation.

In many technologies, a short between ground or power and a signal line can make the line remaining at a fixed logical value. This line will stuck at either a logic 0 or 1 value no matter what input vector is applied in testing. The line is said to have a stuck-at fault. The stuck-at model is the most popular fault model used in gate level simulation, test pattern generation and fault simulation. This basic type of stuck-at fault can be tested simply by directly observing the logical value of the signal line.

Single-stuck-at fault model assumes only one line to be faulty at a time. Single-stuck-at faults are permanent. A line l stuck-at a value v(0 or 1) can be perceived as cutting the line l and applying a constant signal v to the output of the line l. Line l is said to be stuck-at-v. In other words, the line l has a suck-at-v fault which is denoted by l_v.

The single-stuck-at fault of a line, previously discussed, is caused by a short between a fixed voltage and the signal line. It is a basic stuck-at fault model which is widely used in conventional ATE softwares. For e-beam testing, direct access of internal signal lines is possible. The stuck-at model is revised so as to include the stuck-at fault effect due to the defects of circuit components.

Physical defects of circuit component can make an input line appear to permanently remain at a particular value. These defects can only be detected at the output of the logic gate. Basic stuck-at fault model is not sufficient to tackle this problem. The stuck-at faults should be extended to model the effect of defects inside transistors of a gate.

For example, the CMOS NAND gate in Figure 9 has a discontinuity in the diffusion layer of one of the switching transistors(marked in the schematic). This fault is equivalent to a stuck-at-1 fault at the corresponding input. However, the input concerned may not be physically stuck at a logic level. Therefore the stuck-at-1 fault cannot be detected by probing this input directly but has to be detected at the gate output.

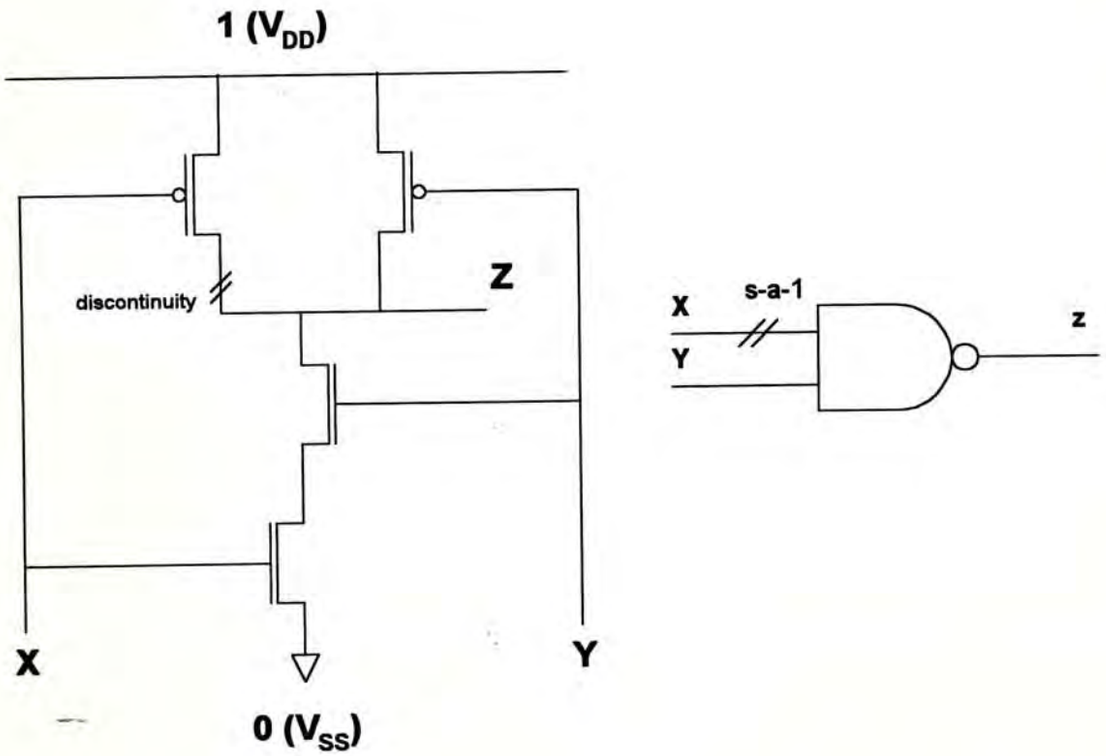


Figure 9 An Extended Stuck-at Model of a CMOS NAND Gate

The following advantages of the single-stuck-at fault model are highlighted.

1. In single-stuck-at fault model, faults are often restricted to the input or output lines of gates. In a circuit containing n lines, it has 2n distinct possible stuck-at faults.

2. Single stuck-at faults are particularly suited for use with common fault simulation and test-generation procedures such as critical path tracing[19] and D-algorithm[20]. These methods attempt to compute the circuit conditions that enable the fault signal to be sensed at an observable primary output line. Several stuck-at faults can be detected along a sensititized path which in turn decreases the cost of test generation and the testing time.

3. Single-stuck-at fault model has a high effectiveness in the testing of digital circuits. Tests for stuck-at faults tend to thoroughly exercise all logic gates of a circuit. For example, an n-input AND gate has 2(n+1) distinct stuck-at faults associated with its input and output lines. A unique set of at least n + 1 test patterns are sufficient to detect all the stuck-at faults. Stuck-at faults based tests tend to apply almost all possible input patterns to the gates. With these test patterns exercising each gate, most physical faults are likely detected if an incorrect logic signal appears at the gate output.

In our e-beam test pattern generation algorithm, the extended single-stuck-at fault model is used.

3.2 Observability and Controllability

The two measures, controllability and observability, are frequently employed to estimate the testability of an IC. The controllability is a measure of how easy the logic of an internal circuit node can be controlled from the input pins. The observability is a measure of how easy the internal circuit node can be observed at the output pins.

An e-beam tester improves the observability of an IC by its ability to observe any internal circuit nodes without physical contact[21]. The observability problem is then solved because the number of probe points can be increased at will. As a result, the IC becomes more 'testable' and it implies that the number of test vectors is reduced. Under this highly observable environment, a new test generation algorithm is required.

3.3 Netlist Format

Many benchmark circuits are written in the ISCAS '85 netlist format. It is a standard netlist format which is used by many researchers as a basis for fault simulation and test generation. The netlist of our test circuits are also written in this netlist format. The circuit is described by lines of node. A node is defined as follows:

address name type fanout fanin
(addresses of fanin nodes if any)

where

- address is a unique number that differentiates this node line from all others in the circuit.
- name is a string of characters used to provide more meaningful information about the node usage.
- type is the function performed by the gate driving this node.

For example, a brief description of a simple circuit sc1(Figure 10) is stored in the file "sc1.isc". The first line of this netlist tells the following information about a single node in the circuit:

- the node address is 1
- the node name is 1gat
- the node type is inpt
- the number of fanout is 1
- the number of fanin is 0

For example, looking at the node with address 13 in the sc1 circuit, we see that it has a fanin of 2 and the two fanin nodes addresses are 9 and 10.

Circuit Description File: "sc1.isc":

```
1       1gat  inpt  1     0
2       2gat  inpt  1     0
3       3gat  inpt  1     0
4       4gat  inpt  1     0
5       5gat  inpt  1     0
6       6gat  inpt  1     0
7       7gat  inpt  1     0
8       8gat  inpt  1     0
9       9gat  and   1     2
  1      2
10      10gat and   1     2
  3      4
11      11gat or    1     2
  5      6
12      12gat or    1     2
  7      8
13      13gat or    1     2
  9      10
14      14gat and   1     2
  11     12
15      15gat and   0     2
  13     14
```

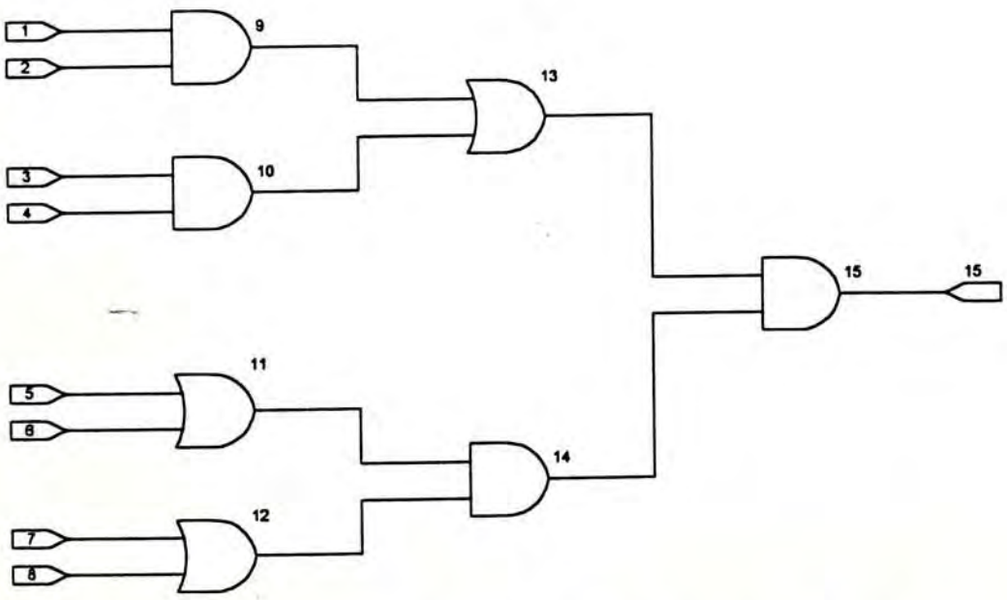


Figure 10 A Simple Circuit sc1

3.4 Level

The level of a line is defined as follows: the level of a gate output is L_{max}+1 if the highest level among its input lines is L_{max}. Line near primary input is of smaller level whereas line near primary output is of larger level. The level of primary input is zero.

One of preprocessing steps before test generation is to define levels for all the lines in a circuit. Usually, primary inputs are assigned a level of zero. Each time a gate is traversed, towards to a primary output, the level is incremented by one. An algorithm to obtain levels for all lines of a circuit is listed below.

Step 1 Set levels of all Primary Inputs = 0;

Step 2 Set current_level = 0;

Step 3 Consider each gate with level = current_level,
If levels of all inputs to the gate is defined,
then level of its output = maximum{levels of all its inputs};

Step 4 If levels of all lines are defined,
then go to Step 5,
else current_level = current_level + 1
and go to Step 3;

Step 5 Levels of all lines are obtained.

For example, the level of the simple circuit sc1(Figure 10) is listed below.

level	node line address
0	1, 2, 3, 4, 5, 6, 7, 8
1	9, 10, 11, 12
2	13, 14
3	15

Table 1 Level of lines of Circuit sc1

3.5 Reconvergent Fanout

In a circuit, a node may propagate the electrical signal to more than one node. In such a case, the node is said to have fanout. A circuit without fanout node is said to be fanout-free. The simple circuit sc1(Figure 10) is a fanout-free circuit. A reconvergent fanout refers to different paths from the same fanout node reconverging to the same node. For example, in figure 11, nodes Y and Z are fanout nodes, and the node Y reconverges at gate G4. At node Y, line 5 is a fanout stem and its fanout branches are lines 5a and 5b. The fanout paths {5a, 6} and {5b, 7} reconverge at gate G4. M.W. Roberts and P.K. Lala's algorithm[22] can be employed to find reconvergent fanouts. The outline of the algorithm is described in appendix A.

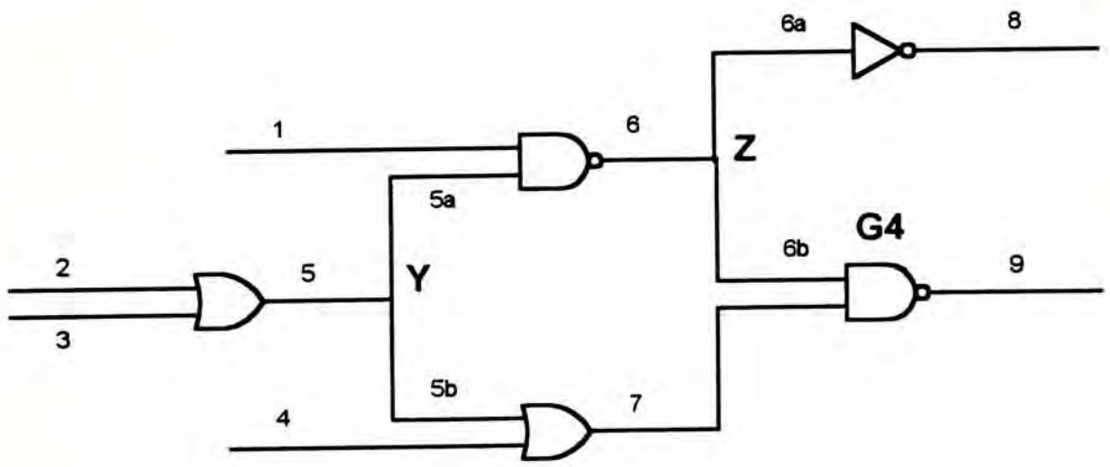


Figure 11 A Circuit with 2 Fanout Nodes and 1 Reconvergent Site

Fanout in circuits is the most troublesome aspect that test generation and fault simulation algorithms have to deal with. Test generation and fault simulation become more complex for circuits with fanouts.[19, 23, 24] A fanout-free circuits can be described by a tree structure and decisions about fault propagation in such circuits can be made more easily. To deal with the problem, the circuit is partitioned into regions each of which has no internal fanout. The input line of a fanout node is the fanout stem and its output lines are fanout branches.

4. CONVENTIONAL TEST GENERATION

4.1 Conventional Automatic Test Generation for ATEs

The conventional testing strategy for ATEs is based on the comparison of the output responses of the DUT and the expected responses. Test vectors are fed to the PIs and output responses are observed through the POs. The observable points are restricted to POs only. The goal of test generation is to generate a test which propagates a fault to the POs.

The majority of the existing test generation algorithms for combinational circuits uses gate-level single-stuck-at fault model. D-algorithm[25], PODEM[26], and FAN[27] are among the best known. These algorithms are based on computing an input test pattern that enables an error signal generated due to a single-stuck-at fault to propagate from the fault site to the primary outputs through some sensitized paths in a circuit.

D-algorithm is one of the most popular algorithms used in automatic test generation systems. A five-value $\{0, 1, X, D, \overline{D}\}$ calculus is employed. D-algorithm divides into forward implication, D-drive and backward justification steps. These steps are repeated for each fault. At each pass, the effect of a fault is sensitized to POs and the values of PIs will be a test vector.

The PODEM algorithm is generally more effective than D-algorithm for circuits containing many exclusive-or gates. Contrary to D-algorithm that assigns values to internal signals, PODEM only assigns values to PIs. Since the backtracking in PODEM can only occur at PIs, the total number of backtrackings is much reduced. In the worse case, PODEM will exhaustively examine all possible input patterns. The FAN algorithm is a refinement of PODEM and performs special processing of fanout points. It has been shown to be more efficient than PODEM.

These test generation algorithms usually start with a specified stuck-at fault at a site. The test vector generated can detect this line stuck-at fault. Other stuck-at faults may be detected and are determined by a fault simulator. Hence,

during test generation, a fault simulator is also needed to evaluate the test vector and to determine the faults detected by this test vector.

Figure 12 is a flow diagram for conventional automatic test generations. At the start of test generation, a fault list which consists of all the stuck-at faults of a circuit is built. A fault is selected from the fault list according to the strategies adopted by the algorithm. The test generation process generates a test vector for this fault. A fault simulator is used to determine the faults detected by this test vector. The detected faults are then removed from the fault list. The process is repeated until the fault list is empty. Each test vector has a corresponding fault dictionary which specify the faults detected.

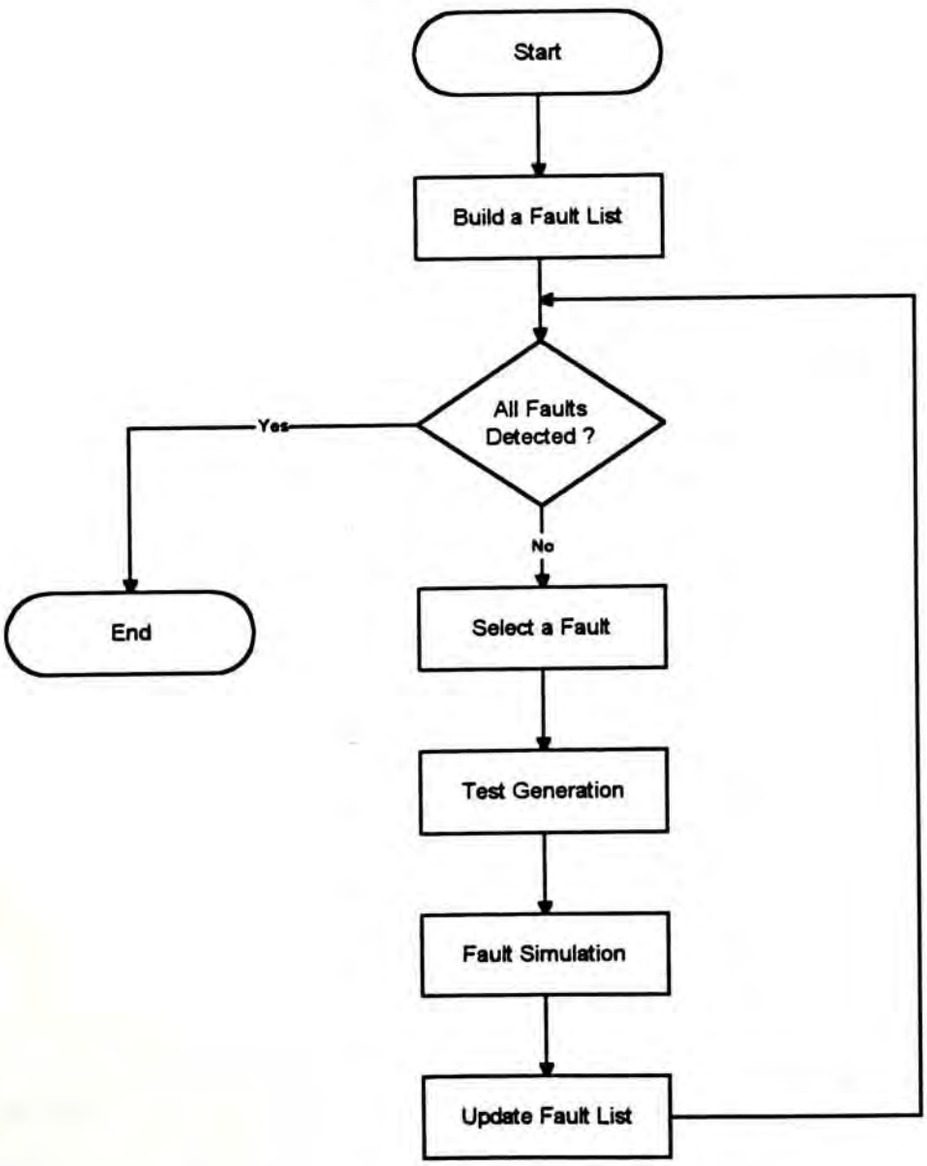


Figure 12 A Conventional ATPG System for ATEs

4.2 Conventional E-Beam Test Generation

Kinch et. al.[28, 29] are probably the first to publish on the subject of test generation algorithm for e-beam testing. Their algorithm assumes that the digital circuit under test may be modeled as a class of synchronous sequential circuit with interconnected states and Boolean elements. In other words, there is a clear separation between the combinational logic block and the sequential block (flip-flops). Since e-beam can probe any internal circuit nodes, states can be observed without the need of a scanning path. As a result, test generation concerns only with combinational logic. This assumption is also valid in our algorithm.

Kinch's algorithm will generate a test set which excites each circuit node to both logic '1' and '0'. According to the stuck-fault model, any circuit node which permanently stays at a logic level implies a faulty circuit. The problem of e-beam test generation is thus seen as the problem of computing a set of test vectors such that every node in the fault-free circuit has a '1' value for at least one of the test vectors, and a '0' value for at least one other test vector. The algorithm hence derived will be simpler than the conventional test generation algorithm because even a randomly chosen test vector is a test for half of the stuck-at faults of a circuit.

Kinch's algorithm suffers from two pitfalls. First, the test set generated may not cover all stuck-at faults, because some stuck-at faults at a gate input can only be detected at the corresponding gate output.

Second, the e-beam has to probe all internal circuit nodes at each test cycle. This is unacceptable if one considers the scanning speed of the e-beam system. It will take excessive time for an e-beam to probe over thousands of nodes that one will normally find in a VLSI circuit.

The e-beam test generation algorithm in the next chapter will guarantee 100% fault coverage for a fanout-free circuit, and at the same time, is trying to limit the number of probe points. The algorithm can also be extended to process non-reconvergent fanout circuits.

5. E-BEAM TEST GENERATION ALGORITHM

5.1 Wafer Stage E-beam Testing

In e-beam testing, internal circuit nodes can also be used as test points. The layout data of the circuit is linked to the electron beam tester, and an appropriate probe point is automatically selected for each signal line[30]. However, there is a snag. E-beam testing does not work with the majority of packaged devices. Even if it does, a relatively long time is required in loading and unloading the DUT to the DUT holder because e-beam testing has to work in a vacuum. In addition, time is needed to align the DUT for probing.

It is proposed that e-beam testing should be applied when the ICs are still in the wafer stage. In a normal fabrication process, individual IC die on a wafer is tested. Faulty die will be discarded and will not be packaged. Making use of e-beam testing at the wafer stage has two advantages. First, testing can be faster than conventional testing because of the reduced number of test vectors. Second, time for probe point alignment is shorter because the alignment is performed on the whole wafer, only slight adjustment is required in stepping from one die to another. Once a die is thoroughly function tested, timing test can then be performed on the packaged die. Timing test is short because only delay critical paths are considered which involve only a small portion of the circuit.

In conventional testing, only the primary outputs are used as test points. In e-beam testing, the test points can be assigned dynamically. Correspondingly, the observability and the fault coverage by each test vector are increased in e-beam testing. For the simple circuit sc1 shown in figure 10, the number of faults detected by the test vector pin(1,2,3,4,5,6,7,8) = vector(1,0,1,1,0,1,0,0) is 5 in conventional testing. In e-beam testing, the number of faults detected for the same test vector is 10. As a result, the number of test vectors can be significantly reduced with e-beam testing. The circuit sc1 requires 6 test vectors in conventional testing. In e-beam testing, the number of test vectors required is reduced to 4. There is about 30% reduction in the number of test vectors.

5.2 Critical Paths Generation

Critical path tracing method was originally developed as an alternative to fault simulation[19, 24]. It was then further applied to handle sequential circuits[31, 32]. The concept of critical path was introduced by Wang[10] in his work on test generation algorithm. For test generation, the critical path tracing method is able to generate a test set in a single pass to cover all faults. This is distinctly different from conventional test generation algorithms which are designed to generate a test for a single fault. For fault simulation, the method has the advantages of directly identifying the faults detected by a test, without simulating the set of all possible faults, and of avoiding the computation of the effect of faults. As a result, critical path tracing is a more efficient alternative to fault simulation.

A critical path is similar to a sensitized path defined by the concept of critical values. The critical value can be defined as follows.

Definition: A line l has a critical value v in the test t if t detects the fault l s-a-$\bar{v}$. A line with a critical value in t is said to be critical in t.[10] The value of this line is denoted by Cv.

Definition: A gate input i is sensitive if the complement value of i changes the value of the gate output.

If a gate output is critical, then its sensitive inputs, if any, are also critical. Primary outputs are critical in any tests. Critical path generation algorithm method will determine paths of critical lines, called critical paths. By finding the critical paths in a test t, the faults detected by t are obtained. This test generation method generates test vectors with faults detected. Hence it does not need a fault simulator. If critical path tracing is employed, the fault simulation step of the test pattern generation system is not required.

For example, the circuit shown in figure 13 has two critical paths, namely, path(1,7,10, 11) and path(6,9,11). These lines are all critical with values as indicated. If a critical line changes its value, all the succeeding lines along the corresponding critical paths will change. Six lines have critical values and hence six stuck-at faults can be detected. The test vector is primary input pin(1,2,3,4,5,6) = vector(1,0,1,1,0,1). It can detect a stuck-at-1 fault of the line 7 and stuck-at-0 faults of the lines 1, 6, 9, 10 and 11.

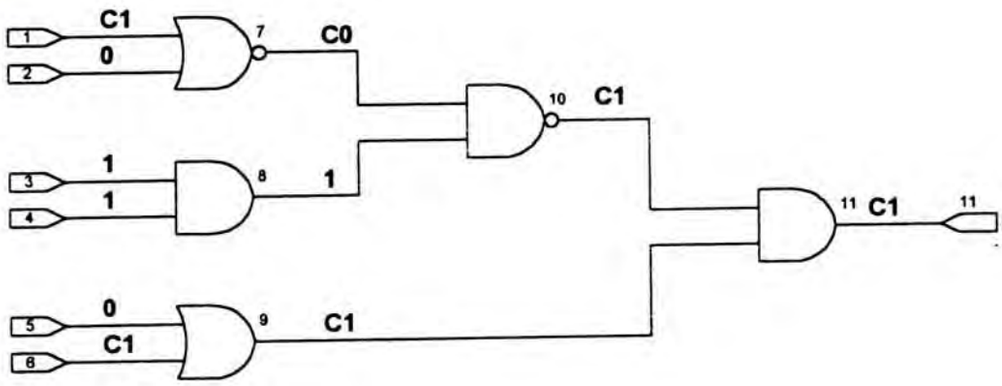


Figure 13 Example of Critical Paths

Traditional critical path method consists of justification processes from primary outputs[10]. The algorithm is modified so that critical path can be created at a seed point. A critical path is obtained with successive justification and propagation processes starting from a seed point.

An automatic e-beam test generation was developed. The test generation algorithm generates test vectors with corresponding probe points. With an evaluation routine, the number of probe points for each test vector is further reduced to reduce testing time.

5.3 Assumptions of Test and Probe Point Generation Algorithm

The test and probe point generation problem is considered with the following assumptons.

1. All internal signal lines of the circuit can be observed. Actually, only signal lines in the top metal layer can be observed. The signal lines not in the top layer can be observed by signal propagation to the top metal layer through the critical paths.

2. The number of probe points is varied from one test vector to another.

3. The fault considered is the permanent stuck-at type. The stuck-at-1 and stuck-at-0 faults of a line **l** are denoted by $\mathbf{l}_1$ and $\mathbf{l}_0$.

4. The stuck-at faults of an input line of a gate should be detected through the output line of the gate.

5.4 Rules of the Test and Probe Point Generation Algorithm

The algorithms have the following rules.

RULE 1: Seed line selection rule for each test vector generation

The seed line must be the highest level line in the fault list. Hence, usually primary output, PO, is the seed line at the beginning of test generation.

RULE 2: Critical value C*v* propagation rule towards primary outputs, POs

To propagate a critical value C*v* of an input line of a gate to its output line, the output critical value of the gate should be C*v*⊕i. There are two possible cases in determining the value of other inputs:

case 1: If this input critical value equals to controlling value c, then all other inputs will be assigned value $\bar{c}$.[RULE 2a]

case 2: If this input critical value equals to negated controlling value $\bar{c}$, all other inputs will be assigned value C*v*.[RULE 2b]

note: the controlling value c is the logic value which will dominate other inputs. For example, if one of the inputs of an AND gate is 0, the output will be 0 no matter what the other inputs are. The AND gate is said to have 0 controlling value. Similarly, the controlling value of an OR gate is 1.

RULE 3: Critical value C*v* justification rule towards primary inputs, PIs

To justify a critical value C*v* of an output line of a gate to its input, the input critical value will be C*v*⊕i. There are two cases in determining which input to be assigned this critical value:

case 1: If the input critical value C*v*⊕i equals to the controlling value c, only one input is assigned C*v*⊕i and all other inputs will be assigned value $\bar{c}$. In this case, a critical value is preferred to be assign to an input line of the gate so that a fault in the fault list can be detected.[RULE3a]

case 2: If the input critical value C*v*⊕i equals to negated controlling value $\bar{c}$, all the inputs will be assigned C*v*⊕i.[RULE 3b]

RULE 4: Potential probe points selection rule

When a critical value C*v* of an output line of a gate is justified, values are assigned to its inputs. If the value assigned to an input is non-critical(RULE 3 case 1), this input line is a potential probe point.

If a critical value is propagated to primary outputs from a seed line, these primary outputs are potential probe points.

RULE 5: Probe points reduction rule

After a test vector and corresponding potential probe points have been generated, some potential probe points are removed if they cannot detect a new stuck-at fault.

5.4 Probe Points Selection and Reduction

There are three cases to insert probe points.

1. If a primary output is the seed line, a probe point is added to it.

2. If the seed line is an input line of a gate, a probe point is added to the output line of the gate.

3. A probe point is assigned to a highest level non-critical value line beside critical paths during justification[RULE 3b].

Critical paths have been created with potential probe points and these probe points can be further reduced.

The most time consuming part in electron beam testing is moving between probe locations. Hence, the number of probe points should be reduced. A test vector with its potential probe points has been generated. These potential probe points will be removed if it cannot detect any fault in the current fault list. After the reduction step, a test vector and a set of reduced probe points are obtained. The corresponding faults detected by this test vector will be revised.

E.g. 1: At the beginning of test generation for the circuit shown in figure 14, the primary output line 7 is the seed line. To detect 7_1 fault, line 7 is assigned C0 and a critical path is generated towards PIs.

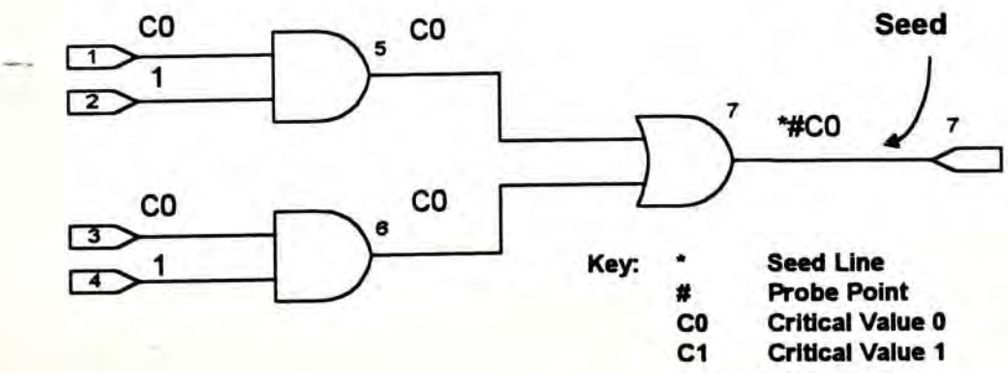


Figure 14 Example of Seed Line Selection Rule

E.g. 2: In figure 15, the current fault list is { 2_0, 5_0 }, so line 5 is the seed line and is assigned C1. Critical paths will be created starting from the seed.

A critical path is created from the line 5 to the primary output. Line 7 is assigned C1 and line 6 is assigned 0. A potential probe point is added to the line 6 and another critical path will be generated towards PIs. Finally, two critical paths are generated with potential probe points line{6, 7}

The faults {2_0, 5_0} in the fault list are detected from the potential probe point line 7. None of fault in fault list is detected from line 6. Hence, potential probe point line 6 is removed and only line 7 is a probe point.

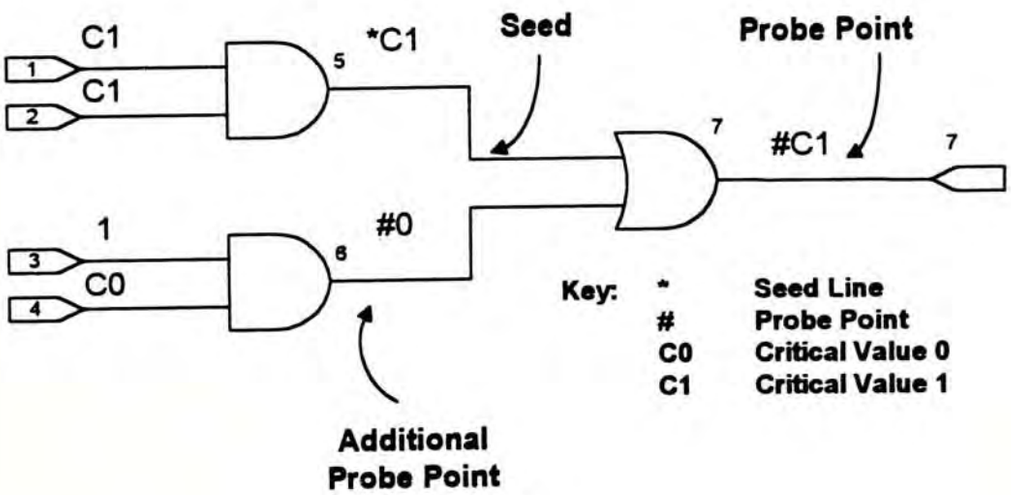


Figure 15 Example of Multiple Critical Paths Generation

5.6 Automatic E-Beam Test Generation Algorithm[33, 34]

Conventional test generation process is associated with two sub-processes. Firstly, a test vector is generated for a selected target fault. Secondly, a fault simulator is employed to find all faults detected by the test vector. These detected faults are then discarded from the fault list. These processes are repeated until the fault list is empty. In this chapter, the concept of critical value is used so that faults detected are obtained with the test vector. Hence, a fault simulator isn't needed. A block diagram of an e-beam test and probe point generation system is shown in figure 16.

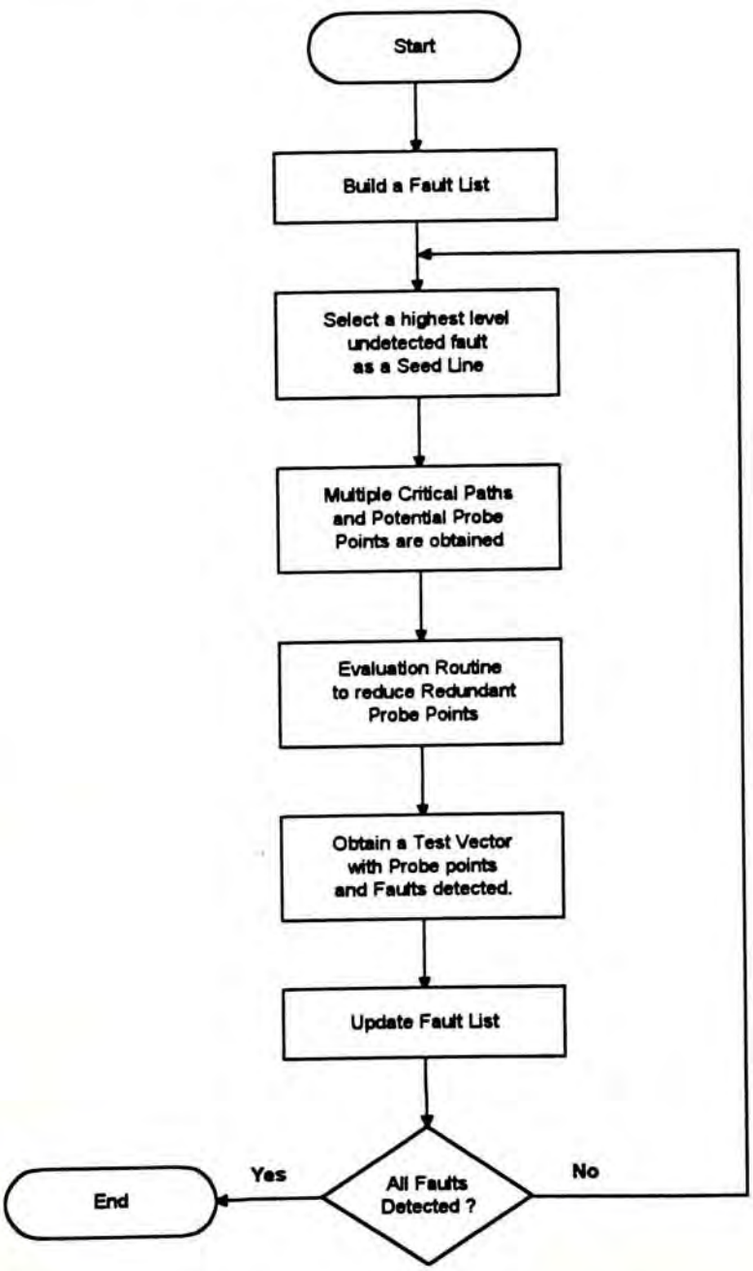


Figure 16 An E-Beam Test and Probe Point Generation System

The algorithm starts with a circuit description file and a fault list consisting of all stuck-at faults. A seed line is then chosen and critical paths are in turn determined by successive steps of propagation, justification and potential probe point addition. Redundant probe points are removed, then a test vector with probe points are generated. This procedure is repeated until all stuck-at faults are detected. The followings are a more detailed description of the procedure.

Step 1: Read the circuit description file.

Step 2: Build a fault list.

Step 3: Select a seed line(RULE 1).

Step 4: Apply propagation and justification rules:
A critical path is generated forward towards the POs(RULE 2).
A critical path is generated backward towards the PIs(RULE 3).
Potential probe points are added and justification is carried out from these points(RULE 4).
Then, a test vector with corresponding probe points are generated.
The stuck-at faults detected by this test vector are also obtained.

Step 5: Remove redundant probe points.(RULE 5)
Then, probe points and detected faults are revised.

Step 6: Remove detected faults from the fault list.

Step 7: Repeat from step 3 if the fault list is not empty.

Step 8: A set of test vectors with corresponding probe points and faults detected are generated.

5.7 Propagation and Justification at Fanout Node

In step 3 of the algorithm, a seed line is selected and is assigned to have a critical value. A critical path is generated forward towards primary outputs and backward towards primary inputs according to the propagation and justification rules. A fanout node(Figure 17) may be reached during the critical path generation. In this chapter, we consider fanout nodes which are not reconvergent at a site. The effect of reconvergent fanouts will be discussed in chapter 7.

When a critical value C*v* reaches the fanout stem of a fanout node, its fanout branches will be assigned the same critical value(Figure 18). Then, these critical values will be propagated towards primary outputs.

When a critical value C*v* or a value *v* reaches a fanout branch of a fanout node, both its fanout stem and the other fanout branch are assigned a critical value C*v*(Figure 19). A critical path is generated from the fanout stem towards primary inputs according to the justification rule. Another critical path is generated from the other fanout branch towards primary outputs according to the propagation rule.

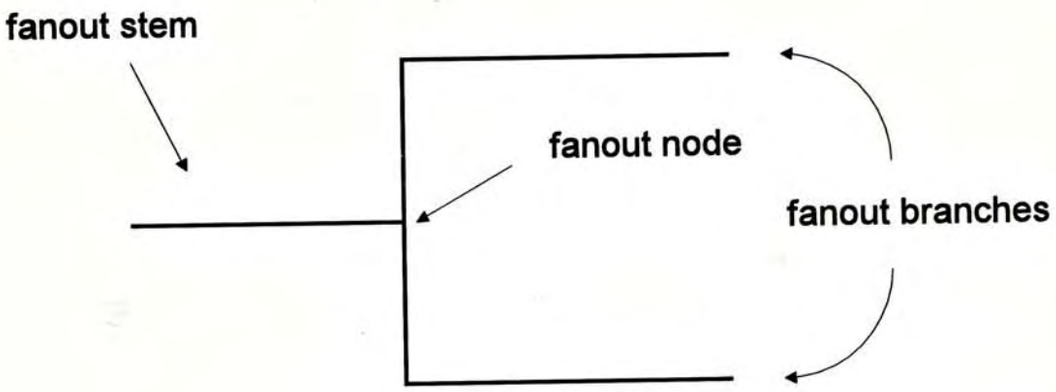


Figure 17 A Fanout Node

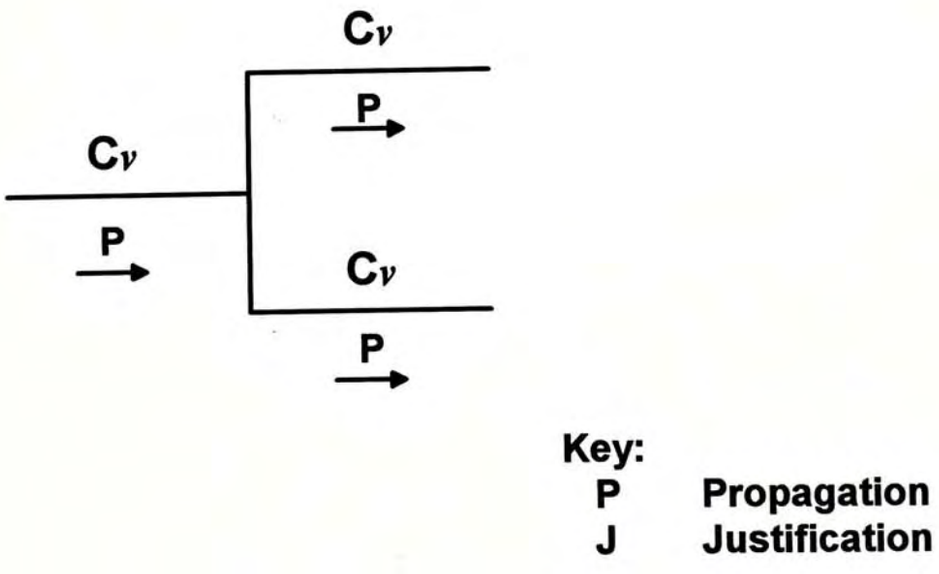
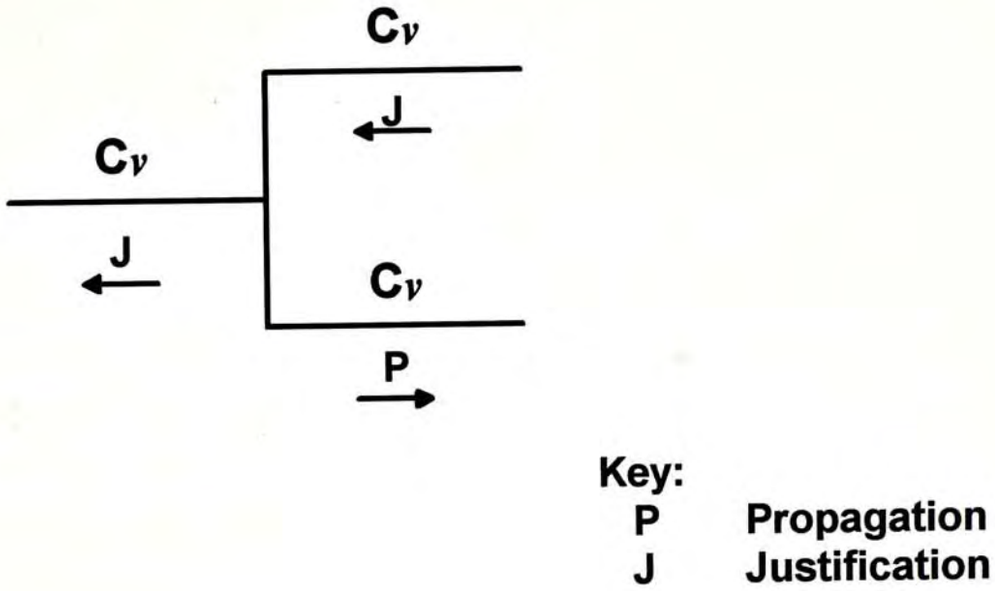
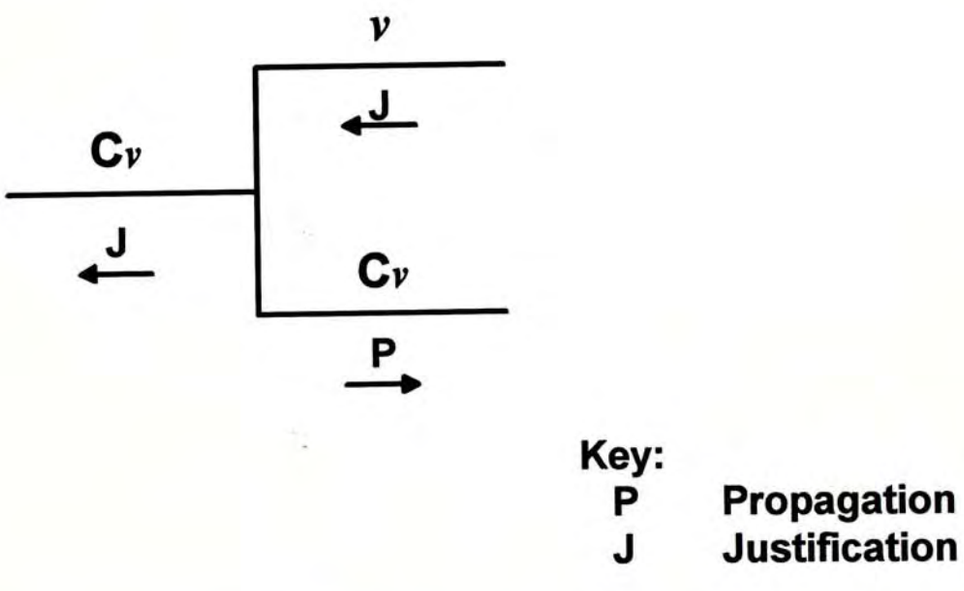


Figure 18 Propagation Reaches a Fanout Stem



(a) A critical value C*v* reaches a fanout branch



(b) A value *v* reaches a fanout branch

Figure 19 Justification Reaches a Fanout Branch

6. EXAMPLES

6.1 Example of Test and Probe Point Generation for Circuit sc2

Simple circuit sc2, in figure 20, requires 7 test vectors in the conventional testing which uses only primary outputs as test points. In e-beam testing, the number of test vectors is reduced to 4. The followings show what happens at each algorithm step of execution.

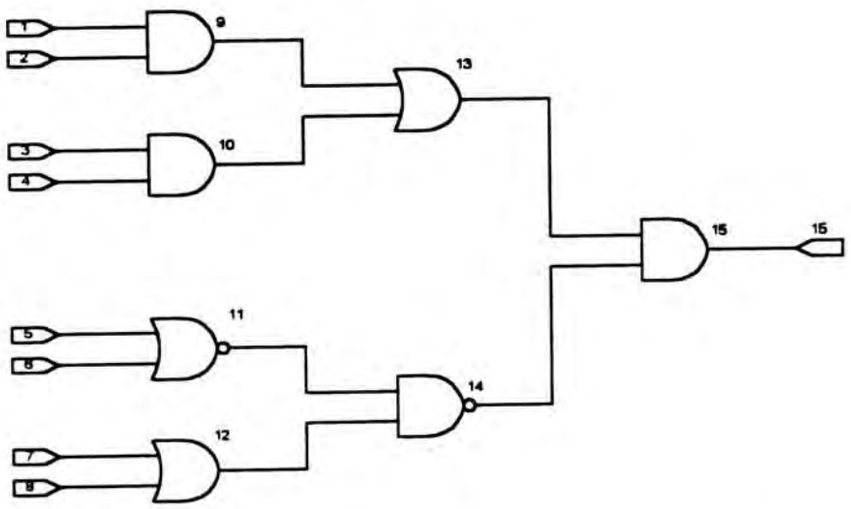


Figure 20 A Simple Circuit sc2

Step Executed: Detailed Descriptions

Step 1: Read in the circuit description file of the circuit sc2.

Step 2: Fault List = $\{1_0, 1_1, 2_0, 2_1, 3_0, 3_1, 4_0, 4_1, 5_0, 5_1, 6_0, 6_1, 7_0, 7_1, 8_0, 8_1, 9_0, 9_1, 10_0, 10_1, 11_0, 11_1, 12_0, 12_1, 13_0, 13_1, 14_0, 14_1, 15_0, 15_1\}$.

Step 3: Primary output line 15 with fault 15_1 is chosen as a seed line and C0 is assigned.

Step 4: The critical path established by probe point at line 15 is 15(C0), 14(C0),12(C1),11(C1),8(C1),6(C0),5(C0).
Probe point at line 13 is added and the corresponding critical path is 10(C1), 4(C1), 3(C1).
Probe point at line 9 is added and the corresponding critical path is 2(C0).
The test vector is pin(1,2,3,4,5,6,7,8) = v(1,0,1,1,0,0,0,1) and the probe points are {15, 13, 9}.

Step 5: No potential probe point is removed.

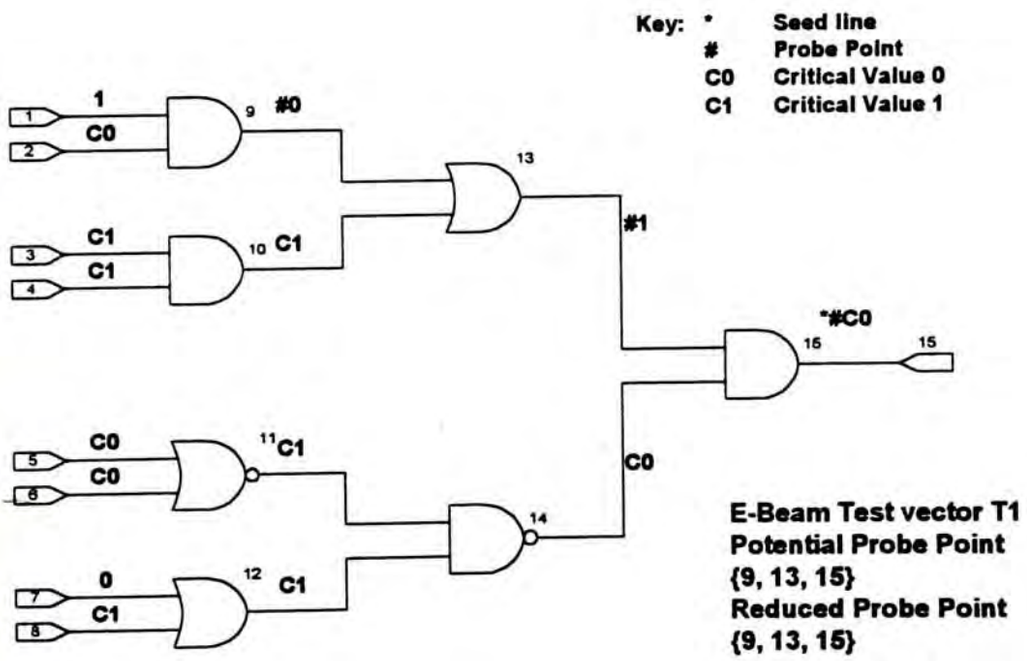


Figure 21 Test Vector 1 for Simple Circuit sc2

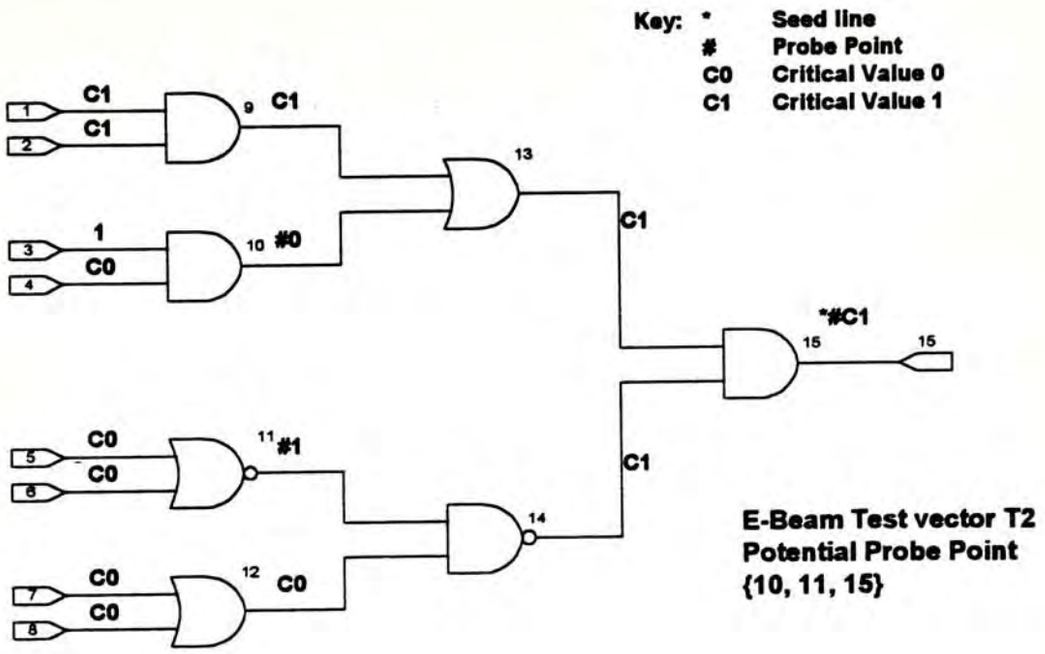
Step 6: New faults detected are
{ $\mathbf{15_1}$, $\mathbf{14_1}$, $\mathbf{12_0}$, $\mathbf{11_0}$, $\mathbf{10_0}$, $\mathbf{8_0}$, $\mathbf{6_1}$, $\mathbf{5_1}$, $\mathbf{4_0}$, $\mathbf{3_0}$, $\mathbf{2_1}$}
Current fault list is
{ $\mathbf{1_0}$, $\mathbf{1_1}$, $\mathbf{2_0}$, **3**, $\mathbf{4_1}$, $\mathbf{5_0}$,
$\mathbf{6_0}$, $\mathbf{7_0}$, $\mathbf{7_1}$, $\mathbf{8_1}$, $\mathbf{9_0}$, $\mathbf{9_1}$, $\mathbf{10_1}$,
$\mathbf{11_1}$, $\mathbf{12_1}$, $\mathbf{13_0}$, $\mathbf{13_1}$, $\mathbf{14_0}$, $\mathbf{15_0}$ }.

Step 7: Since the current fault list is not empty, go to Step 3.

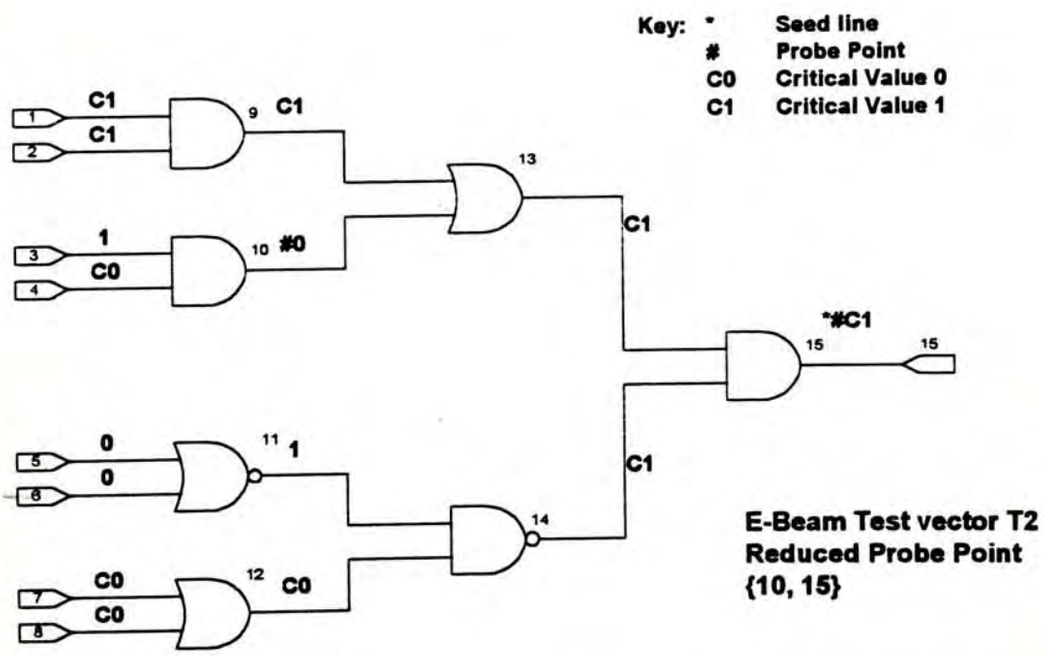
Step 3: Primary output line 15 with fault $\mathbf{15_0}$ is chosen as a seed line and C1 is assigned.

Step 4: The critical path established by probe point at line 15 is 15(C1),14(C1),13(C1),12(C0),9(C1),8(C0),7(C0),2(C1),1(C1).
Probe point at line 11 is added and the corresponding critical path is 6(C0),5(C0).
Probe point line at line10 is added and the corresponding critical path is 4(C0).
The test vector is v(1,1,1,0,0,0,0,0) and the probe points are {15,11,10}.

Step 5: The potential probe point at line 11 is removed because faults{$\mathbf{6_1}$,$\mathbf{5_1}$} are already detected.
Probe points become { 15, 10 }.



(a) Potential Probe Points



(b) Reduced Probe Point

Figure 22 Test Vector 2 for Simple Circuit sc2

Step 6: New faults detected are
{ 15_0, 14_0, 13_0, 12_1, 9_0, 8_1, 7_1, 4_1, 2_0, 1_0 }
Current fault list is { 1_1, 3_1, 5_0, 6_0, 7_0, 9_1, 10_1, 11_1, 13_1 }.

Step 7: Since the current fault List is not empty, go to Step 3.

Step 3: Primary output line 13 with fault 13_1 is chosen as a seed line and C0 is assigned.

Step 4: The critical path established by probe point at line 15 is 15(C0),13(C0),10(C0),9(C0),3(C0),1(C0).
Probe point at line 14 is added and the corresponding critical path is 11(C0),6(C1).
Probe point at line 12 is added and the corresponding critical path is 7(C1).
The test vector is v(0,1,0,1,0,1,1,0) and the probe points are {15,14,12}.

Step 5: No potential probe point is removed.

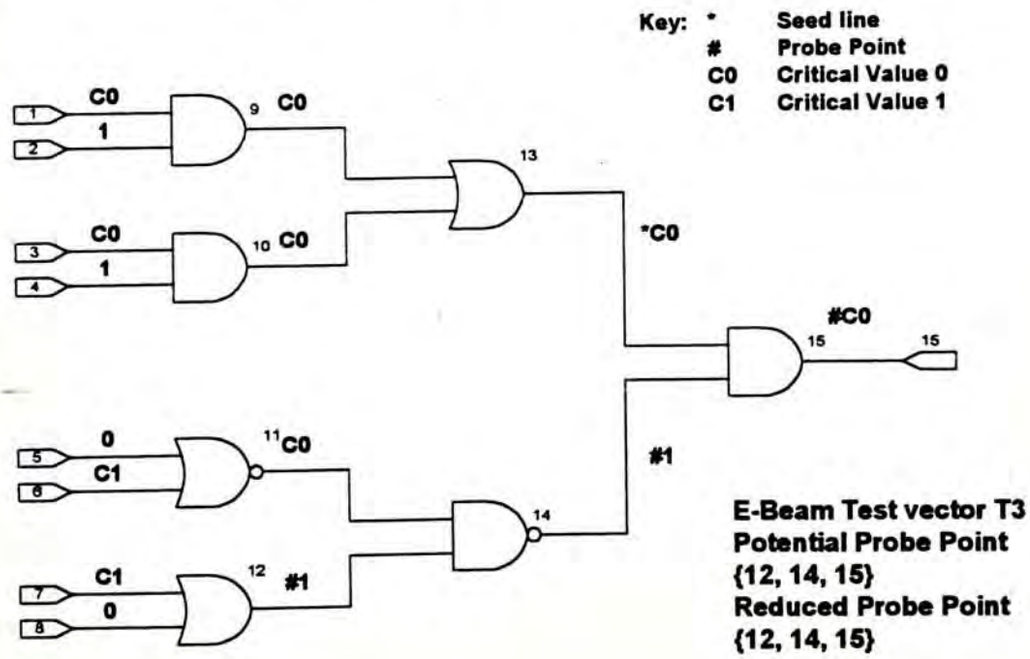


Figure 23 Test Vector 3 for Simple Circuit sc2

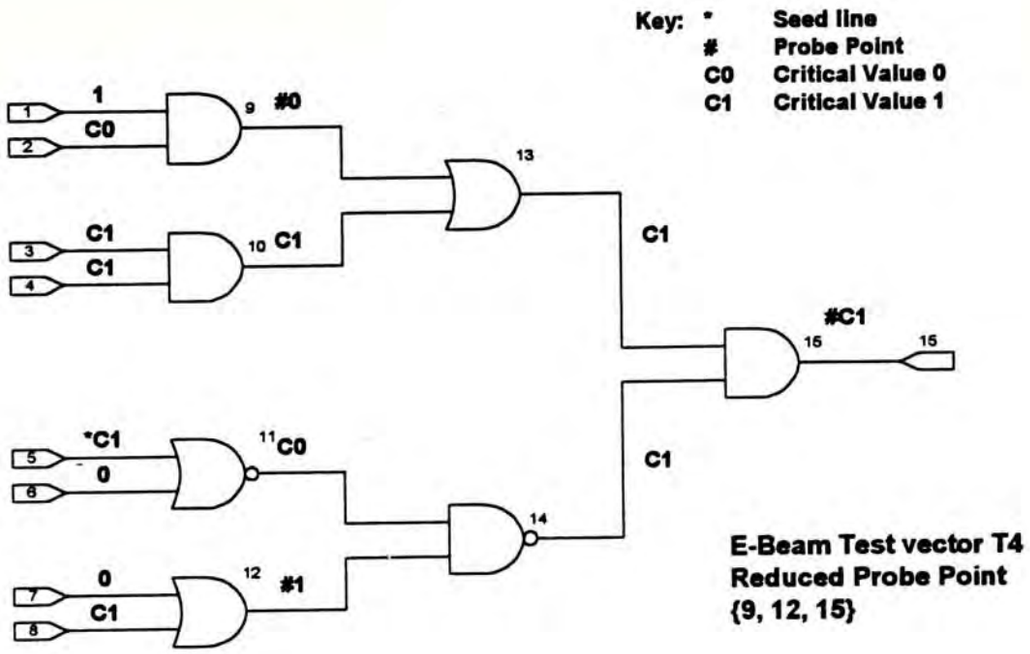
Step 6: New faults detected are { $\mathbf{13}_1$, $\mathbf{11}_1$, $\mathbf{10}_1$, $\mathbf{9}_1$, $\mathbf{7}_0$, $\mathbf{6}_0$, $\mathbf{3}_1$, $\mathbf{1}_1$}
Current fault list is { $\mathbf{5}_0$ }.

Step 7: Since the current fault list is not empty, go to Step 3.

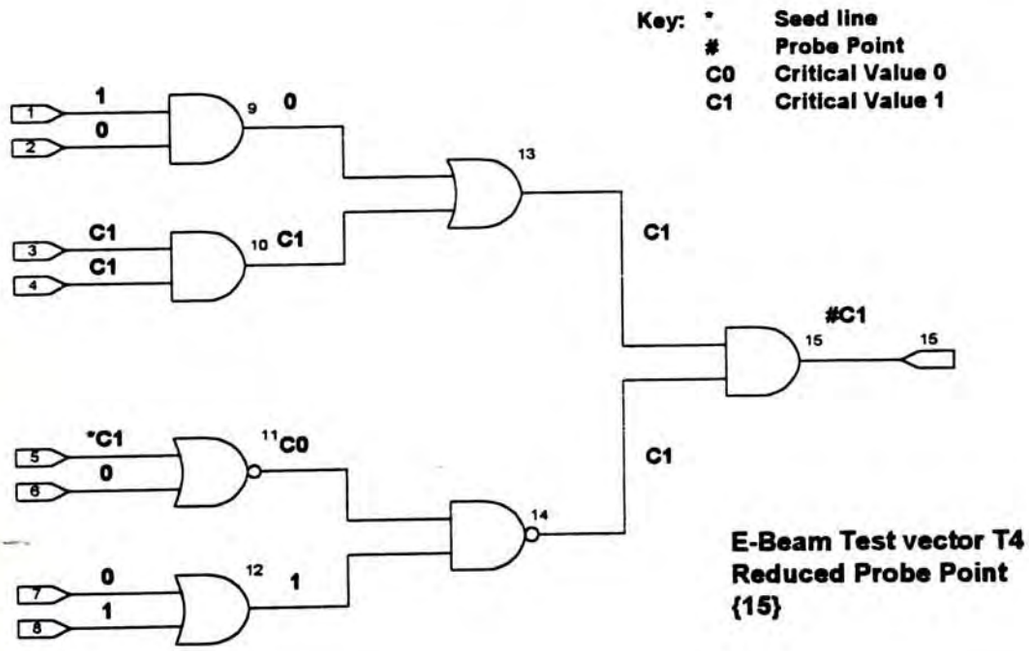
Step 3: Primary output line 5 with fault $\mathbf{5}_0$ is chosen as a seed line and C1 is assigned.

Step 4: After propagating C1 to POs, the critical path established by probe point line at 15 is
15(C1),14(C1),13(C1),11(C0),10(C0),5(C1),4(C1),3(C1).
Probe point at line 12 is added and the corresponding critical path is 8(C1).
Probe point at line 9 is added and the corresponding critical path is 2(C0).
The test vector is v(1,0,1,1,1,0,0,1) and the potential probe points are { 15, 12, 9 }.

Step 5: The potential probe points at line 12 and line 9 are removed.
Probe point becomes { 15 }.



(a) Potential Probe Points



(b) Reduced Probe Point

Figure 24 Test Vector 4 for Simple Circuit sc2

Step 6: New fault detected is { 5_0 }.
Current fault list is empty.

Step 7: Since the current fault list is empty, go to Step 8.

Step 8: The test set of four vectors and probe points are summaried in the table 2.

	Test Vector			
	T1	T2	T3	T4
line 1	1	C1	C0	1
line 2	C0	C1	1	0
line 3	C1	1	C0	C1
line 4	C1	C0	1	C1
line 5	C0	0	0	C1*
line 6	C0	0	C1	0
line 7	0	C0	C1	0
line 8	C1	C0	0	1
line 9	0 #	C1	C0	0
line 10	C1	0 #	C0	C1
line 11	C1	1	C0	C0
line 12	C1	C0	1 #	1
line 13	1 #	C1	C0*	C1
line 14	C0	C1	1 #	C1
line 15	C0#*	C1#*	C0#	C1#
no. of probe points	3	2	3	1
no. of faults detected	11	10	9	8
no. of new faults detected	11	10	8	1

Remarks: * seed line C0 critical value 0
probe point C1 critical value 1

Table 2 Test and Probe Point Generated for Circuit "cs2.isc"

6.2 Example of Test and Probe Point Generation for Circuit sfc4

Simple circuit scf4 in figure 25 has a fanout node with a fanout stem at line 8 and two fanout branches at lines 8a and 8b. It requires 6 test vectors in conventional testing. In e-beam testing, the number of test vectors required is reduced to 4. The followings show what happens at each algorithm step of execution.

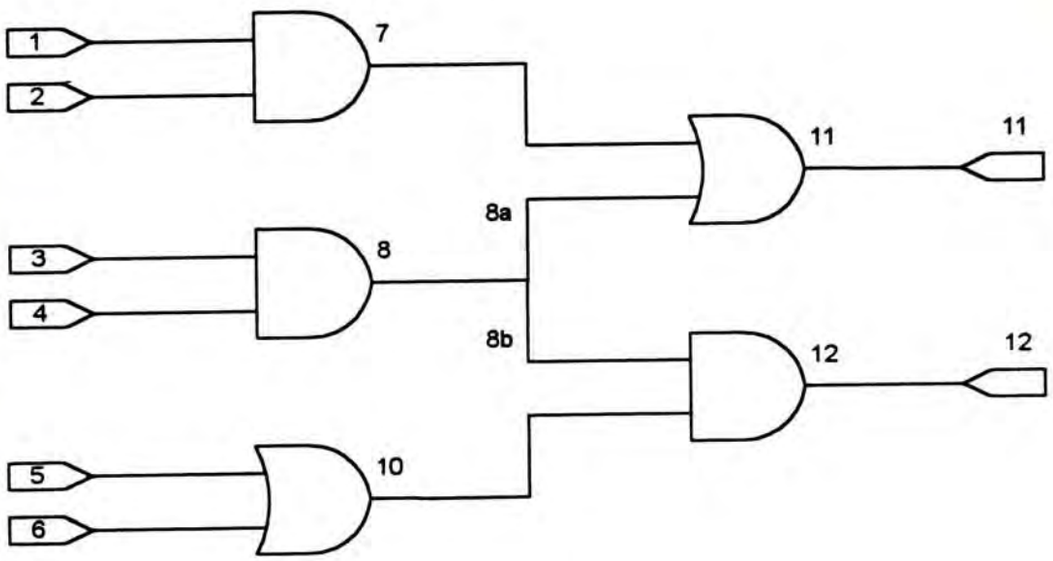


Figure 25 Simple Circuit sfc4

Step Executed: Detailed Descriptions

Step 1: Read in the circuit description file of the circuit sfc4.

Step 2: Fault list = { $\mathbf{1}_0$, $\mathbf{1}_1$, $\mathbf{2}_0$, $\mathbf{2}_1$, $\mathbf{3}_0$, $\mathbf{3}_1$, $\mathbf{4}_0$, $\mathbf{4}_1$, $\mathbf{5}_0$, $\mathbf{5}_1$, $\mathbf{6}_0$, $\mathbf{6}_1$, $\mathbf{7}_0$, $\mathbf{7}_1$, $\mathbf{8}_0$, $\mathbf{8}_1$, $\mathbf{8a}_0$, $\mathbf{8a}_1$, $\mathbf{8b}_0$, $\mathbf{8b}_1$, $\mathbf{10}_0$, $\mathbf{10}_1$, $\mathbf{11}_0$, $\mathbf{11}_1$, $\mathbf{12}_0$, $\mathbf{12}_1$}.

Step 3: Primary output line 12 with fault $\mathbf{12}_1$ is chosen as a seed line and C0 is assigned.

Step 4: The critical path established by probe point at line 12 is 12(C0), 10(C0), 5(C0), 6(C0).
The fanout branch line 8b is assigned a non-critical value 1. A critical value C1 is assigned to fanout stem line 8 and fanout branch line 8b.
Probe point at line 11 is added and the corresponding critical path is 11(C1), 8a(C1), 4(C1), 3(C1).
Probe point at line 7 is added and the corresponding critical path is 2(C0).
The test vector 1 is pin(1,2,3,4,5,6) = v(1,0,1,1,0,0) and the probe points are {7,11,12}.

Step 5: No potential probe point is removed.

Step 6: New faults detected by test vector 1 are
{ $\mathbf{2}_1$, $\mathbf{3}_0$, $\mathbf{4}_0$, $\mathbf{5}_1$, $\mathbf{6}_1$, $\mathbf{8}_0$, $\mathbf{8a}_0$, $\mathbf{10}_1$, $\mathbf{11}_0$, $\mathbf{12}_1$ }.
Number of new faults detected is 10.
Number of probe points is 3.
Current fault list is
{$\mathbf{1}_0$, $\mathbf{1}_1$, $\mathbf{2}_0$, $\mathbf{3}_1$, $\mathbf{4}_1$, $\mathbf{5}_0$, $\mathbf{6}_0$, $\mathbf{7}_0$, $\mathbf{7}_1$,
$\mathbf{8}_1$, $\mathbf{8a}_1$, $\mathbf{8b}_0$, $\mathbf{8b}_1$, $\mathbf{10}_0$, $\mathbf{11}_1$, $\mathbf{12}_0$}.

Step 7: Since the current fault list is not empty, go to Step 3.

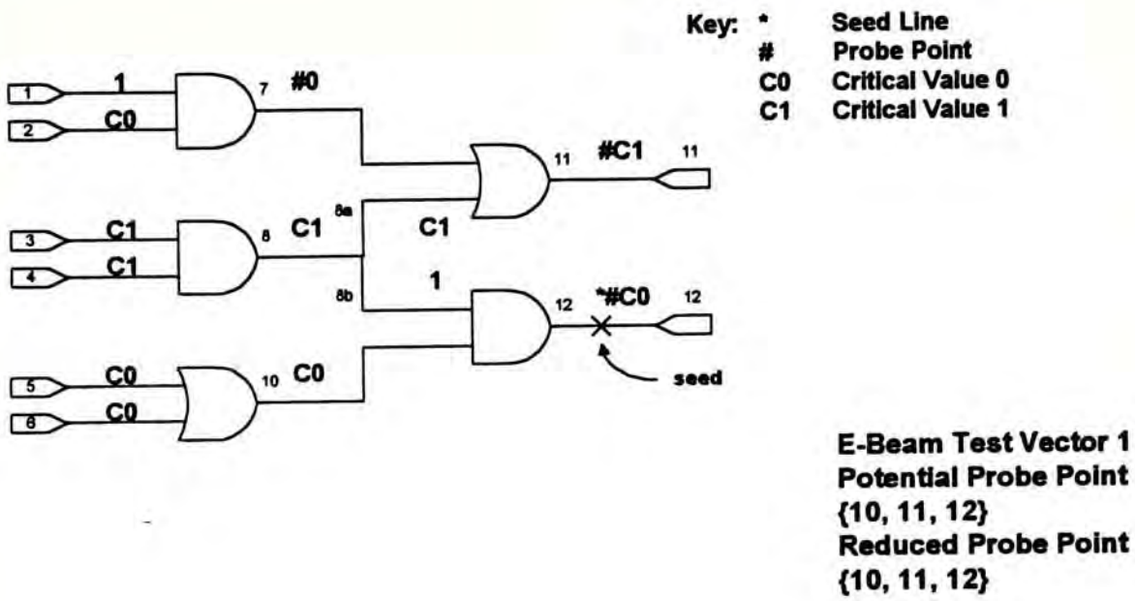
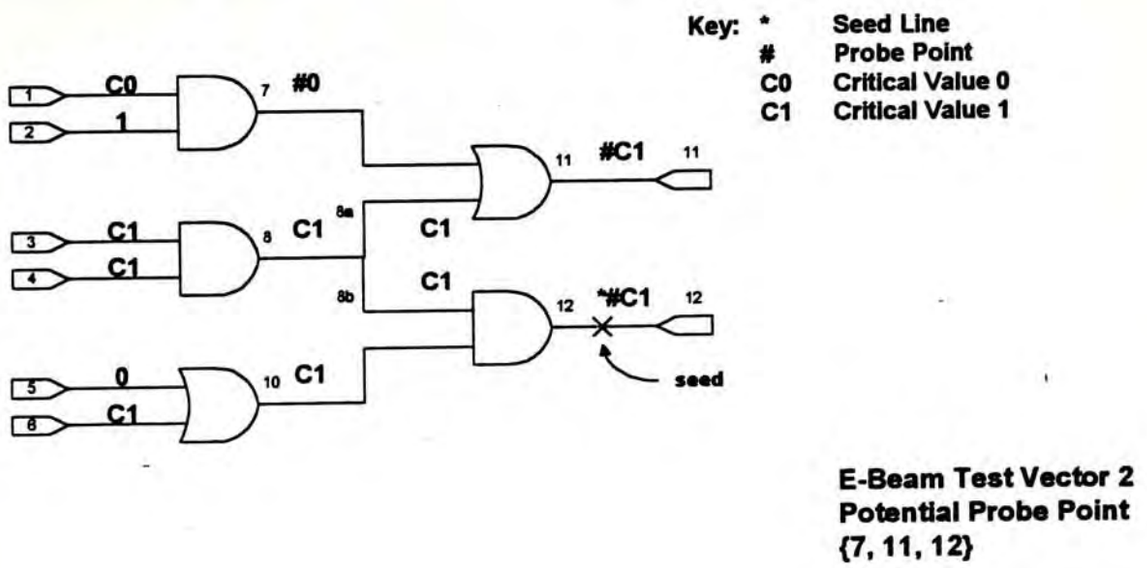


Figure 26 Test Vector 1 for Simple Circuit sfc4

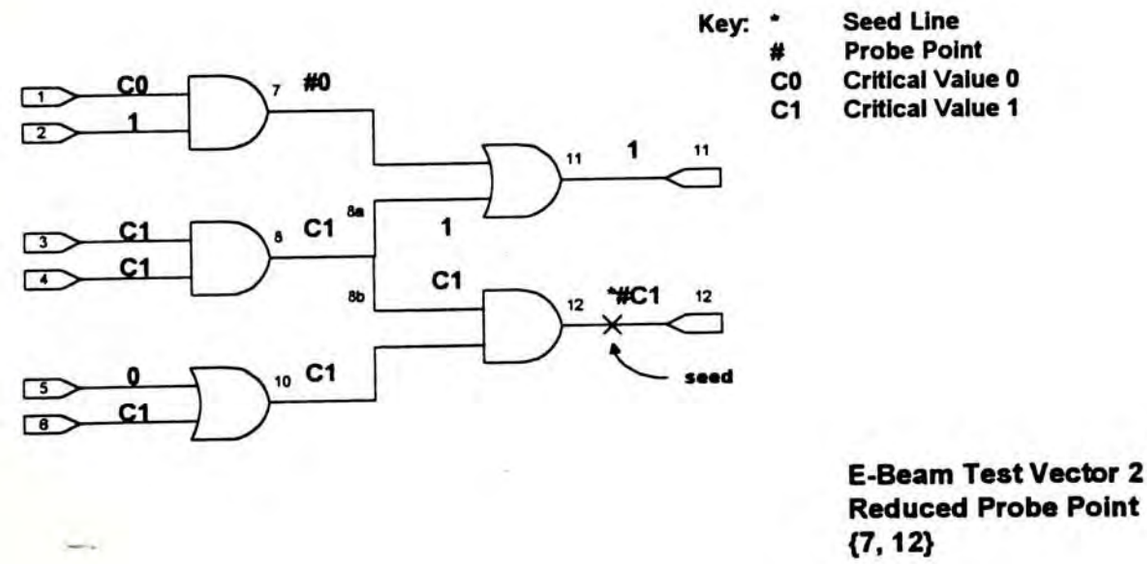
Step 3: Primary output line 12 with fault $\mathbf{12}_0$ is chosen as a seed line and C1 is assigned.

Step 4: Fanout branch line 8b is assigned to a critical value C1. To justify this critical value, both fanout stem line 8 and fanout branch line 8a are assigned to C1.
The critical path established by probe points lines 11 and 12 is 12(C1), 11(C1), 8b(C1), 8a(C1), 8(C1), 10(C1), 6(C1), 4(C1), 3(C1).
Probe point at line 7 is added and the corresponding critical path is 1(C0).
The test vector 2 is v(0,1,1,1,0,1) and the probe points are { 7, 11, 12 }.

Step 5: The potential probe point line 11 is removed because faults $\{ \mathbf{11}_0, \mathbf{8a}_0 \}$ are already detected.
Probe points become { 7, 12 }.



(a) Potential Probe Points



(b) Reduced Probe Points

Figure 27 Test Vector 2 for Simple Circuit sfc4

Step 6: New faults detected are { $\mathbf{1}_1$, $\mathbf{6}_0$, $\mathbf{8b}_0$, $\mathbf{10}_0$, $\mathbf{12}_0$}
Number of new faults detected is 5.
Number of probe points is 2.
Current fault list is
{$\mathbf{1}_0$, $\mathbf{2}_0$, $\mathbf{3}_1$, $\mathbf{4}_1$, $\mathbf{5}_0$, $\mathbf{7}_0$, $\mathbf{7}_1$, $\mathbf{8}_1$, $\mathbf{8a}_1$, $\mathbf{8b}_1$, $\mathbf{11}_1$}.

Step 7: Since the current fault list is not empty, go to Step 3.

Step 3:. Primary output line 11 with fault $\mathbf{11}_1$ is chosen as a seed line and C0 is assigned.

Step 4: Fanout stem line 8 and fanout branch line 8b are assigned C0 because fanout branch line 8a is assigned C0.
The critical path established by probe points at lines 11 and 12 is 12(C0), 11(C0), 8b(C0), 8a(C0), 8(C0), 7(C0), 4(C0), 2(C0).
Probe point at line 10 is added and the corresponding critical path is 5(C1).
The test vector 3 is v(1,0,1,0,1,0) and the probe points are {10, 11, 12 }

Step 5: No potential probe point is removed.

Step 6: New faults detected by test vector 3 are
{ $\mathbf{11}_1$, $\mathbf{8b}_1$, $\mathbf{8a}_1$, $\mathbf{8}_1$, $\mathbf{7}_1$, $\mathbf{5}_0$, $\mathbf{4}_1$}.
Number of new faults detected is 7.
Number of probe points is 3.
The current fault list is { $\mathbf{1}_0$, $\mathbf{2}_0$, $\mathbf{3}_1$, $\mathbf{7}_0$ }.

Step 7: Since the current fault list is not empty, go to Step 3.

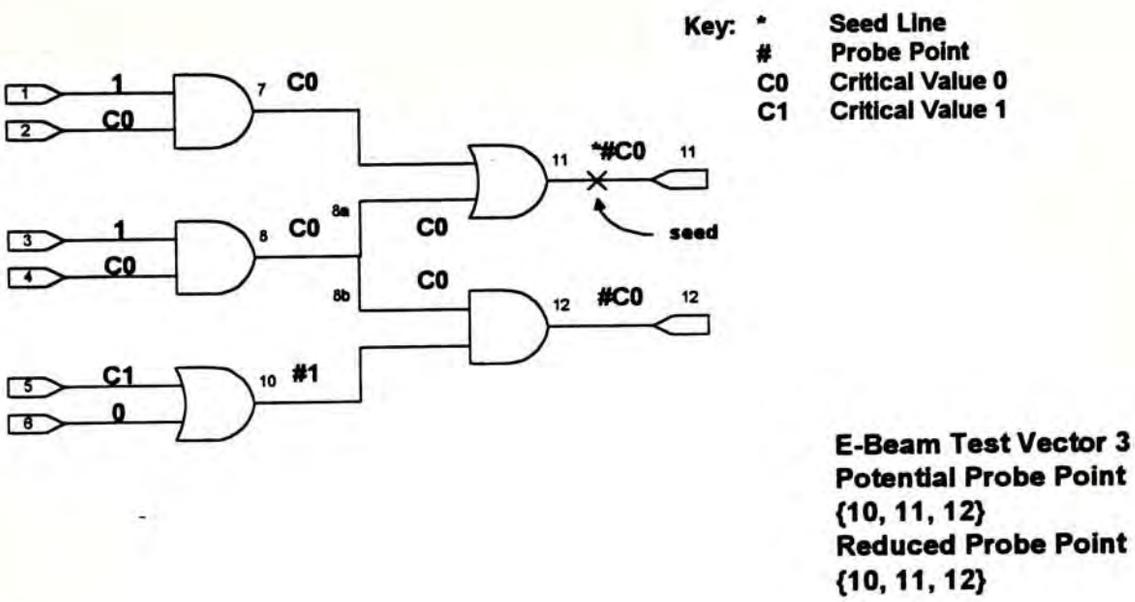
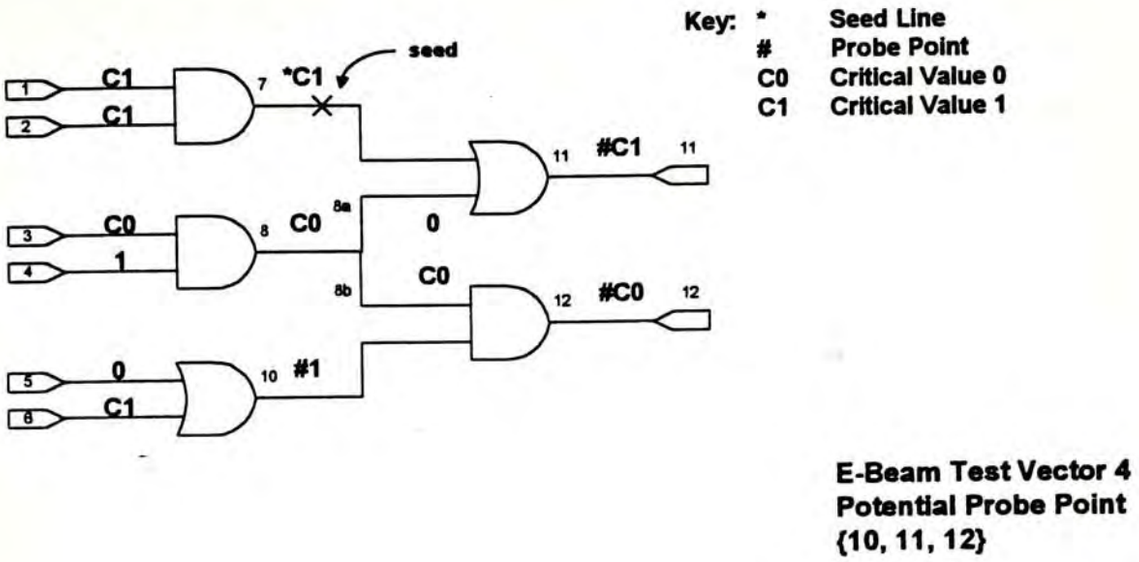


Figure 28 Test Vector 3 for Simple Circuit sfc4

Step 3: Line 7 with fault 7_0 is chosen to a seed line and C1 is assigned.

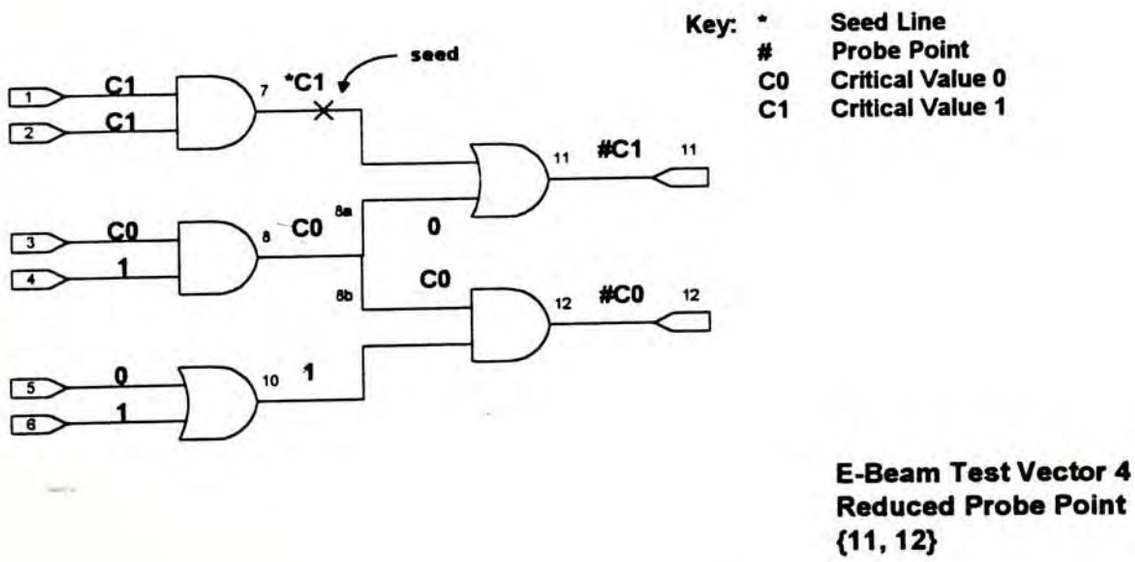
Step 4: The critical path established by probe point at line 11 is 11(C1), 7(C1), 2(C1), 1(C1).
Fanout branch 8a has a non-critical value 0.
Fanout stem line 8 and fanout branch line 8b are assigned C0.
Probe point at line 12 is added and the corresponding critical path is 12(C0), 8b(C0), 8(C0), 3(C0).
Probe point at line 10 is added and the corresponding critical path 6(C1).
The test vector is v(1,1,0,1,0,1) and the probe points are { 10, 11, 12 }

Step 5: The potential probe point line 10 is removed because fault { 6_0 } is already detected.
Probe points become {11, 12}



E-Beam Test Vector 4
Potential Probe Point
{10, 11, 12}

(a) Potential Probe Points



E-Beam Test Vector 4
Reduced Probe Point
{11, 12}

(b) Reduced Probe Points

Figure 29 Test Vector 4 for Simple Circuit sfc4

Step 6: New faults detected are { 1_0, 2_0, 3_1, 7_0 }.
Number of new faults detected is 4.
Number of probe points is 2.
The current fault list is empty.

Step 7: Since the current fault list is empty, go to Step 8.

Step 8: The test set of four vectors and probe points are summarized in table 3.

	Test Vector			
	v1	v2	v3	v4
pin 1	1	0	1	1
pin 2	0	1	0	1
pin 3	1	1	1	0
pin 4	1	1	0	1
pin 5	0	0	1	0
pin 6	0	1	0	1
No. of Probe Pts.	3	2	3	2
No. of New Faults Detected	10	5	7	4

Table 3 Test and Probe Point Generation for Circuit sfc4

7. CONCLUSIONS

7.1 Summary of Results

In e-beam testing, internal circuit nodes can be used as test points. It is proposed that ICs should be tested in the wafer stage. E-beam testing at wafer stage has the following advantages:

1. E-beam testing is a non-destructive and non-capacitive loading method.

2. ICs can be tested when they are in working condition.

3. Alignment time is small since only slight adjustment is required in stepping from one die to another.

4. Testing is faster than the conventional testing because of the number of test vectors is reduced.

This thesis offers a test and probe point generation algorithm for e-beam testing. It generates a set of test vectors with corresponding probe points and single-stuck-at faults detected. With decisions at fanout nodes, it can generate a test set for circuits with non-reconvergent fanouts.

The major results concerning the algorithm are discussed below:

1. Stuck-at faults at an input line of a gate should be detected at the output line of the gate.

2. The algorithm generates a test vector with the corresponding probe points. Dynamic probe points improve fault coverage of the test vector.

3. The algorithm generates a test vector with faults detected. No fault simulator is required. Actually, no commercial fault simulator software for e-beam testing is available.

The circuit in figure 30 is one of the ISCAS85 benchmark circuits, c17[35]. In conventional testing, the number of test vectors required is 6. In E-beam testing, the number of test vectors is 3.

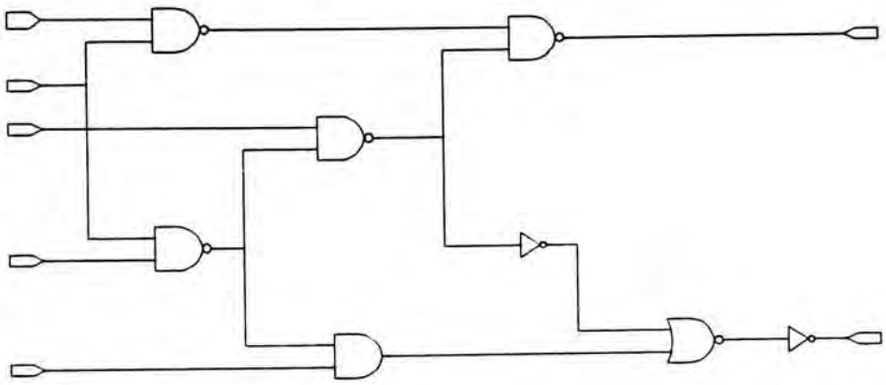


Figure 30 Circuit c17

Table 4 summaries the results obtained by the algorithm on several simple circuits. The results of test generation of circuits sc1 and sc3 are listed in Appendixes B and C. In conclusion, the reduction in the number of test vectors is about 30%. For complex circuits, the result is expected to be even better because more probe points will be available, that in turn implies more critical paths can be generated with each test vector. However, it is not common to find fan-out free circuits more complex than the simple circuits used in the analysis. Our next task will be to modify the algorithm to handle circuits with fanouts. This can only be accomplished if re-convergence in circuit paths can be resolved.

example circuits	sc1	sc2	sc3	sfc4	c17
no. of s-a-faults	30	30	46	26	38
no. of test vectors in conventional testing	6	7	9	6	6
no. of test vectors in e-beam testing	4	4	7	4	3
average no. of probe points per test vector	2.5	2.3	1.6	2.5	2.0
% reduction in no. of test vectors	33%	42%	22%	33%	50%

Table 4 Results of Test Generation

7.2 Further Research

For circuits with reconverging fanouts, test generation problem becomes difficult. First, a lot of retrials are performed in a region containing fanout stems and reconverging fanout sites. Second, analysis is required to prevent the self-masking problem. As a result, the time and complexity of the test generation are increased.

Self-masking occurs when two fanout paths from a fanout node reconverge at a site with different inversion parities. Fault effect at the fanout stem propagates along the two fanout paths cancel out each other at the reconverge site. In figure 31, primary output line 9 with fault $\mathbf{9}_0$ is a seed line. The critical path established by probe points { 8, 9 } is 1(C1), 3(C1), 5(C1), 5a(C1), 6(C0), 6a(C0), 6b(C0), 8(C1), 9(C1). However, in figure 32, line 5 is not critical because effects of fault line 5 stuck-at-0 propagate on two paths with different inversion parities and they cancel each other at the reconverge fanout site G4. This phenomenon is called self-masking.

To avoid self-masking problem, the fault effect should arrive at the reconvergent site through the critical fanout path and values of lines along the non-critical fanout path should remain unchange. This can be done if an input line of a gate along the non-critical fanout path is assigned a controlling value. For example, in figure 31, stem line 5 is critical if line 7 does not change its value under the effect of fault $\mathbf{5}_0$. A controlling value 1 is added to line 4 so that line 7 is kept at value 1 even fault $\mathbf{5}_0$ exits.

Further task will be required to modify the algorithm to handle circuits with reconvergent fanouts.

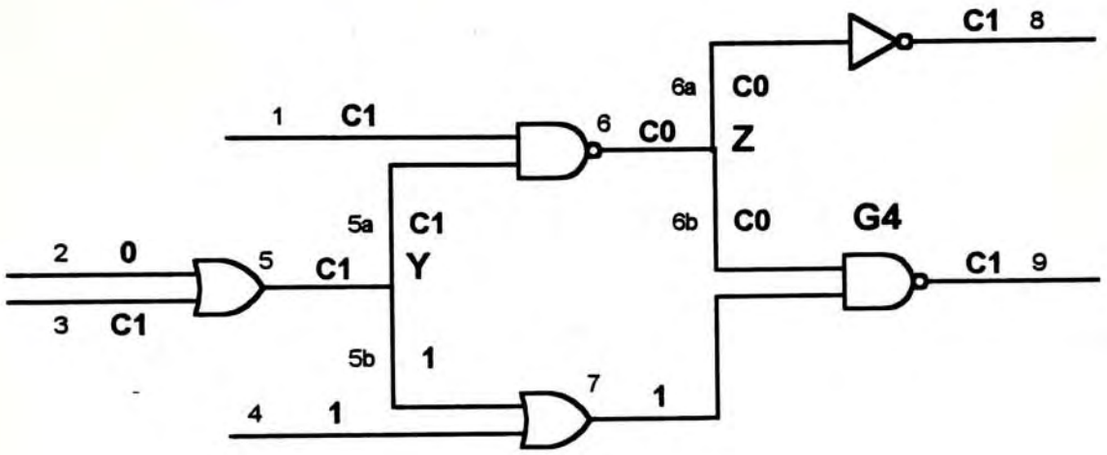


Figure 31 Stem Line 5 is Critical

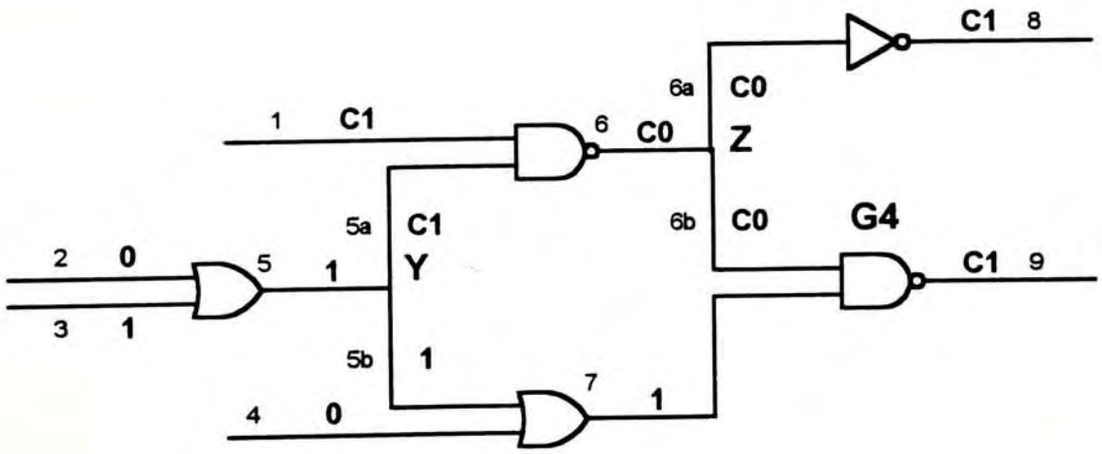


Figure 32 Stem Line 5 is Non-Critical(Self-masking Problem)

Appendix A: Algorithm to Find Reconvergent Fanouts

An algorithm[22] to find the reconvergent fanouts in a circuit is listed as follows:

Step 1: Read in the circuit description file.

```
read_circuit_file( )
```

Step 2: Build the reach_no. count of each node,

```
build_rcount( )
{
for all nodes do
  {
  reach_no. count of each node = no. of inputs to this node
  }
}
```

Step 3: Set the next node list NL to list of PIs,

```
initial_NL( )
```

Step 4: Set the contents of current node list CL to NL,

```
build_CL( )
```

Step 5: Clear the NL,

```
null_NL( )
```

Step 6: For every node in the CL, determine the next node list NL,

```
build_NL( )
{
for each node x in the CL
  {
    for each node y fed by x
      {
      reach_no. count of y = reach_no. count of y -1;
      if reach_no. count of y == 0
        put node y into the next node list NL;
      }
  }
}
```

Step 7: For every node in the CL, build the fol and rfol for each node,
build_fols( ) and
reduce_rfol( )

Step 8: For each node in the CL, add to the fol if it is a fanout node,
add_fol( )

Step 9: If the next list NL is not empty, go back to step 4.

Step 10: Save rfol to file,
save_result( )

where
the reach_no_count of a node is the number of inputs of that node.

Appendix B: Results of Test Generation for Circuit sc1

Results of e-beam test and probe point generation for circuit sc1 are given below:

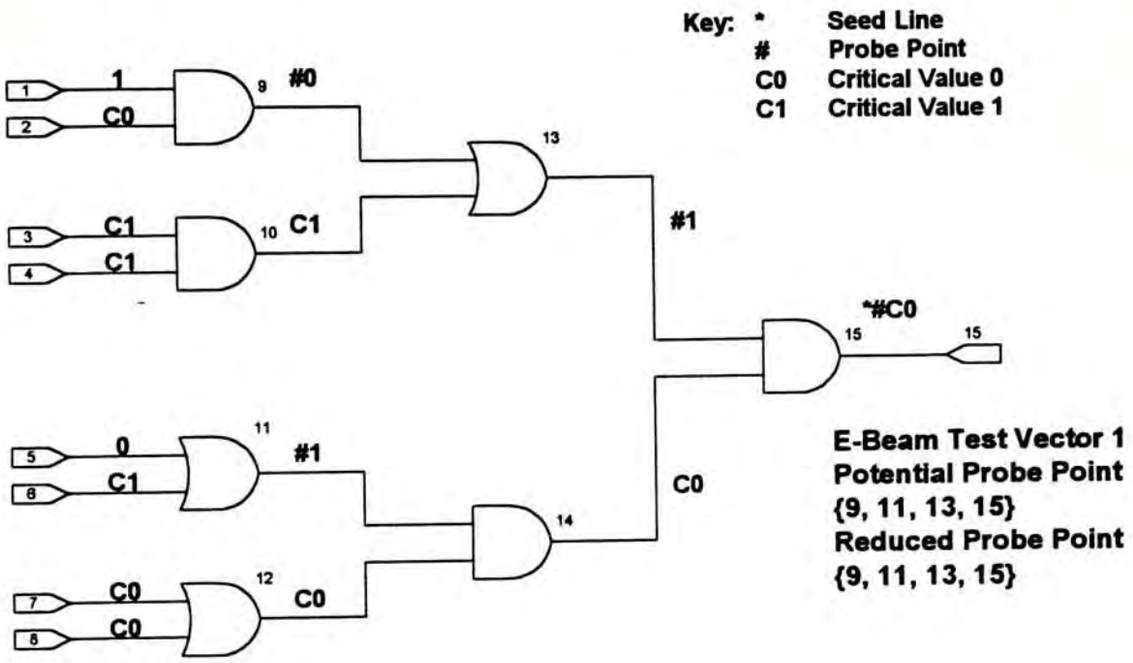


Figure B-1 Test Vector 1 for Circuit sc1

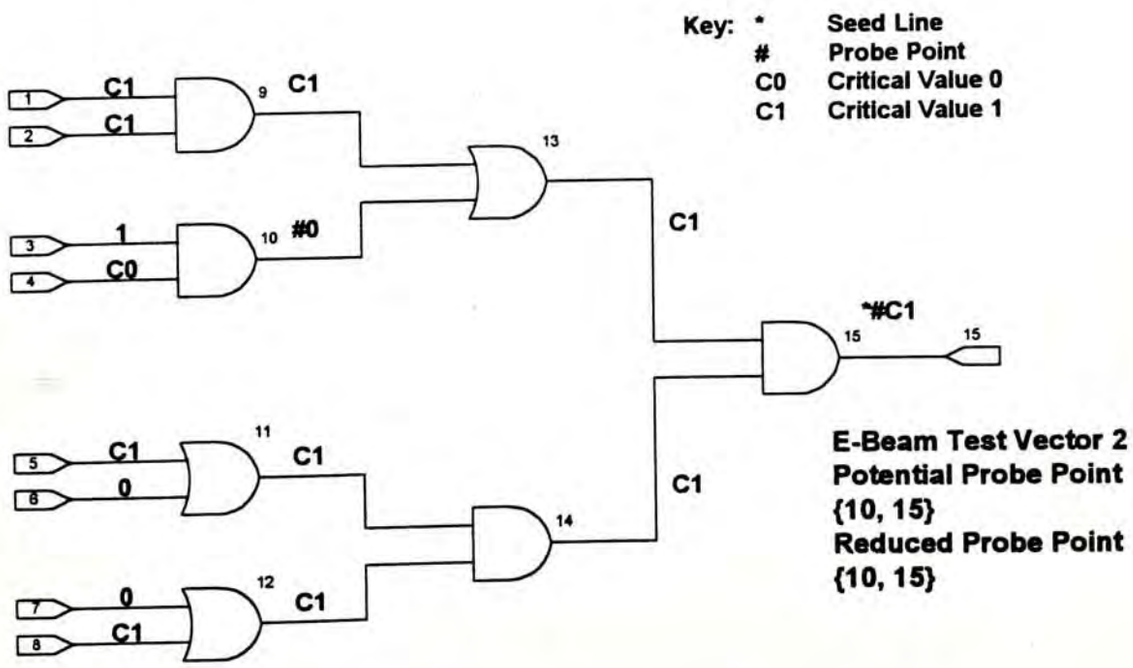


Figure B-2 Test Vector 2 for Circuit sc1

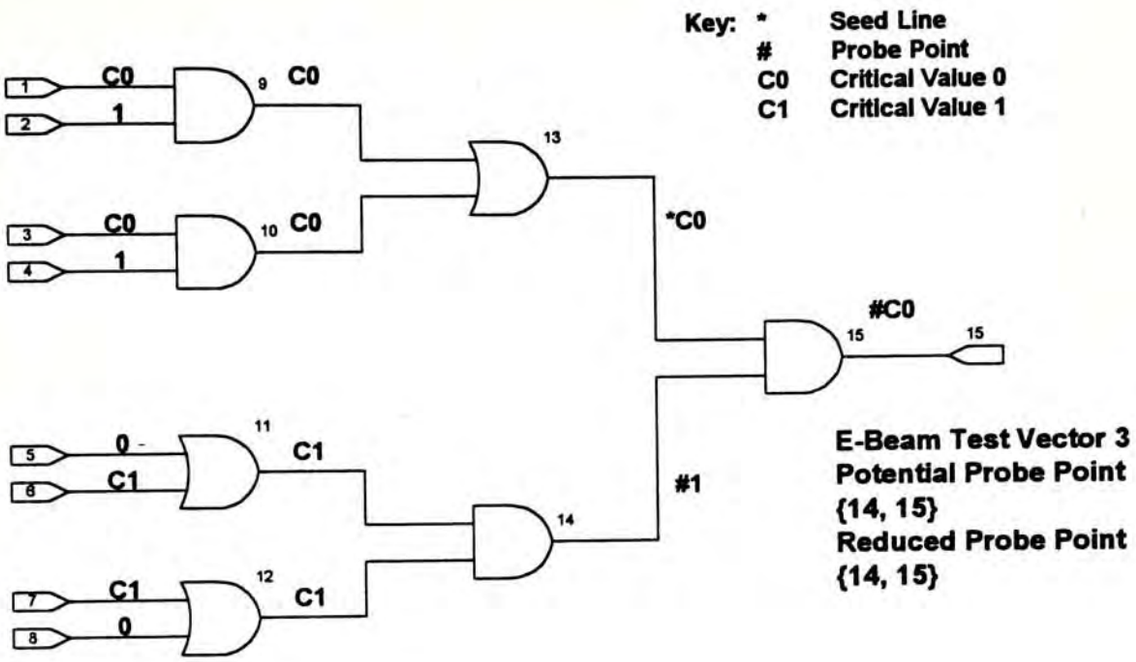


Figure B-3 Test Vector 3 for Circuit sc1

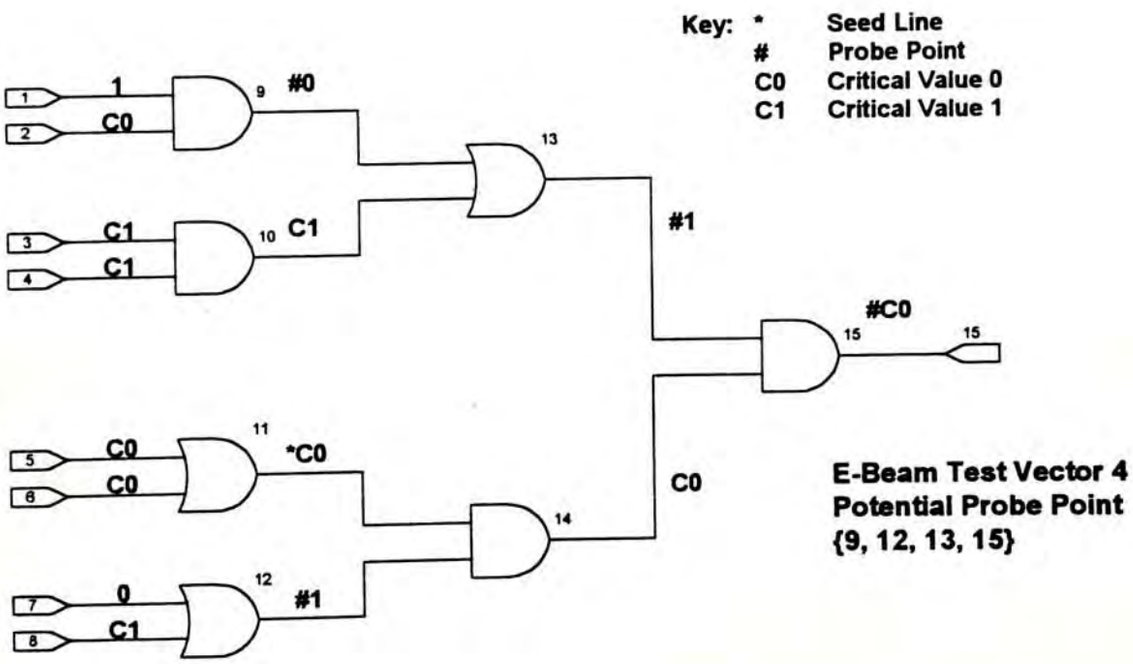


Figure B-4a Test Vector 4 for Circuit sc1 - Potential Probe Points

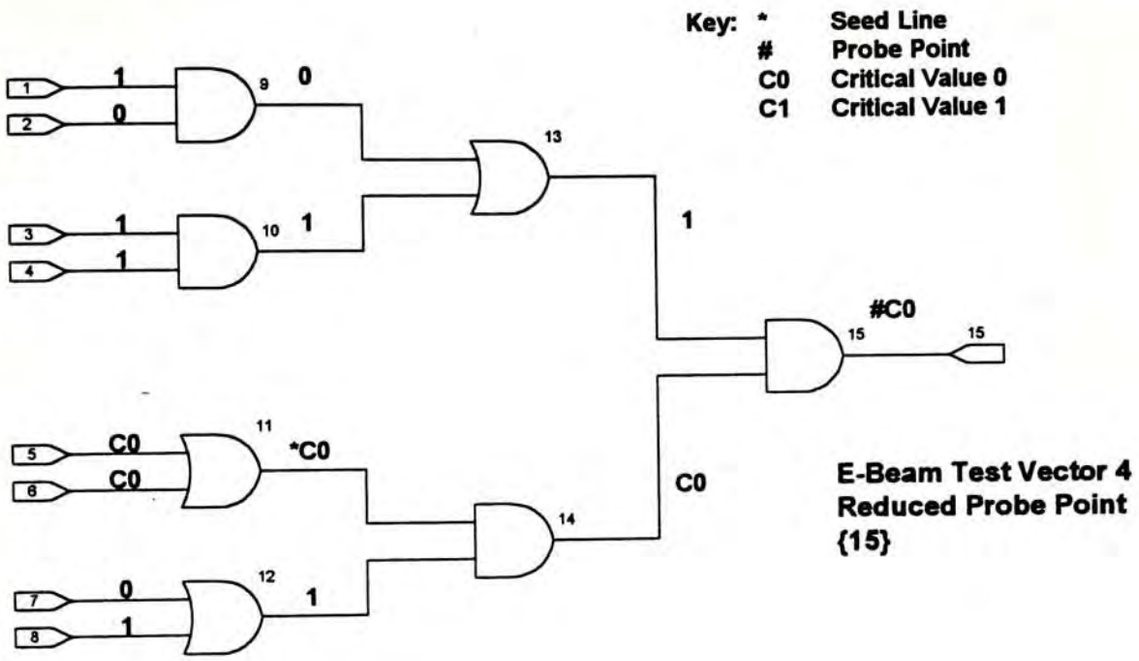


Figure B-4b Test Vector 4 for Circuit sc1 - Reduced Probe Point

	test vector			
	v1	v2	v3	v4
pin 1	1	1	0	1
pin 2	0	1	1	0
pin 3	1	1	0	1
pin 4	1	0	1	1
pin 5	0	1	0	0
pin 6	1	0	1	0
pin 7	0	0	1	0
pin 8	0	1	0	1
probe pts	9,11,13,15	10,15	14,15	15

Table B-1 Results of Test and Probe Point Generation

Appendix C: Results of Test Generation for Circuit sc3

Results of e-beam test and probe point generation for circuit sc3 are given below:

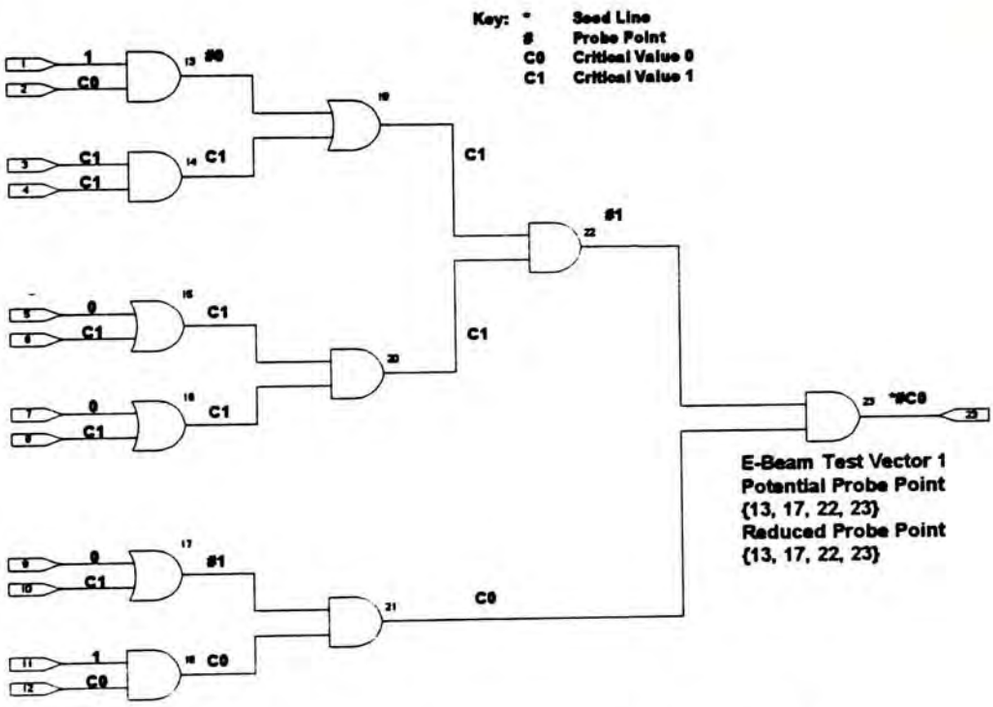


Figure C-1 Test Vector 1 for Circuit sc3

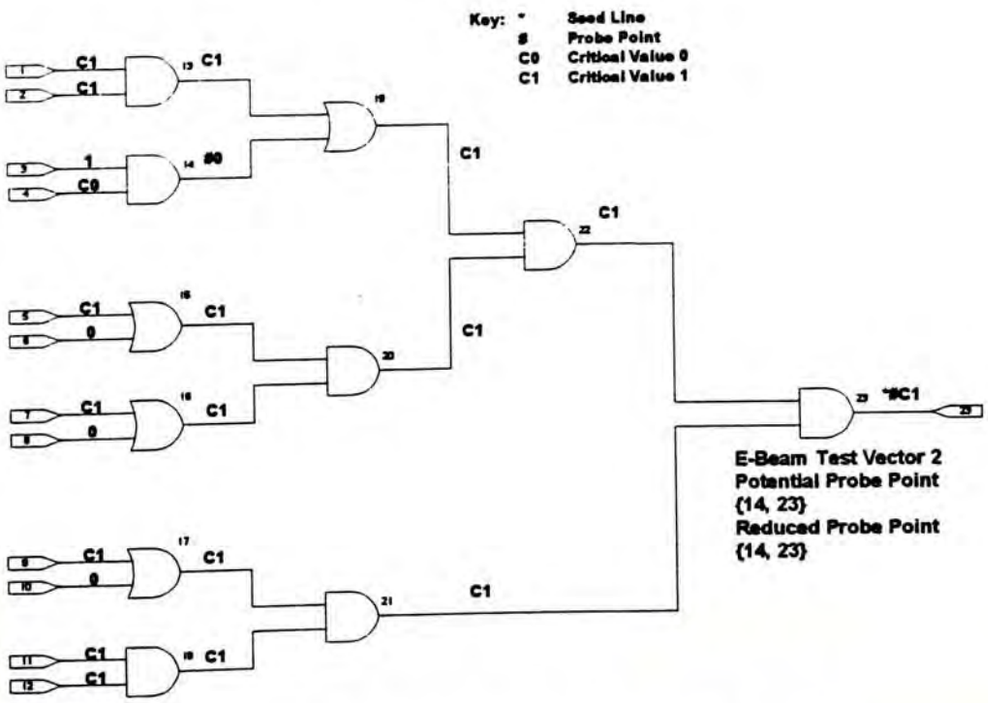


Figure C-2 Test Vector 2 for Circuit sc3

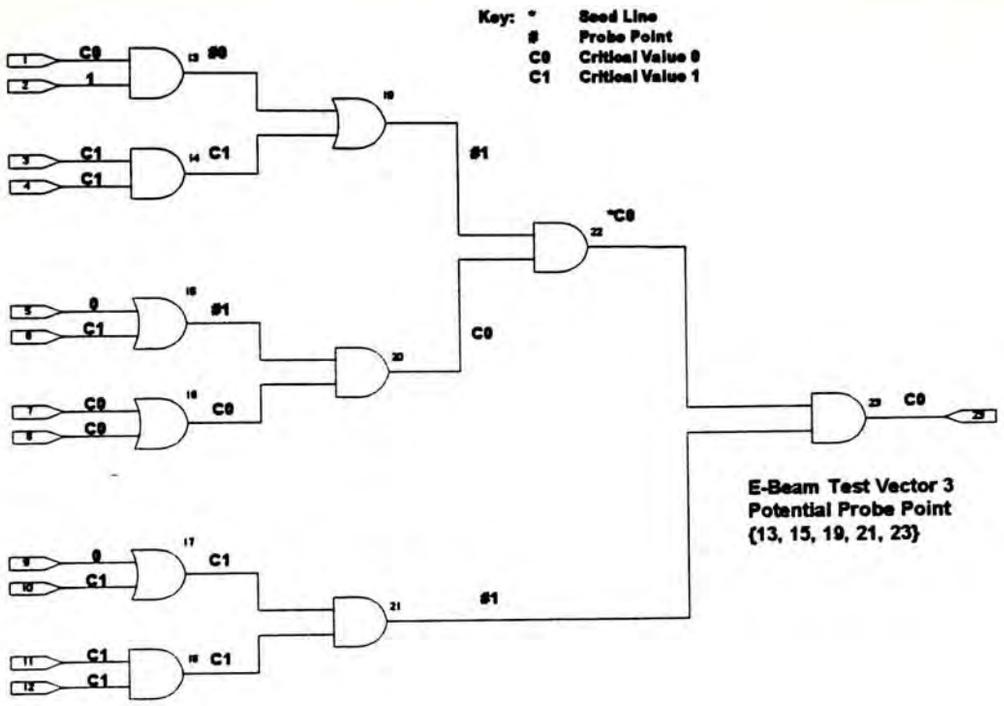


Figure C-3a Test Vector 3 for Circuit sc3 - Potential Probe Points

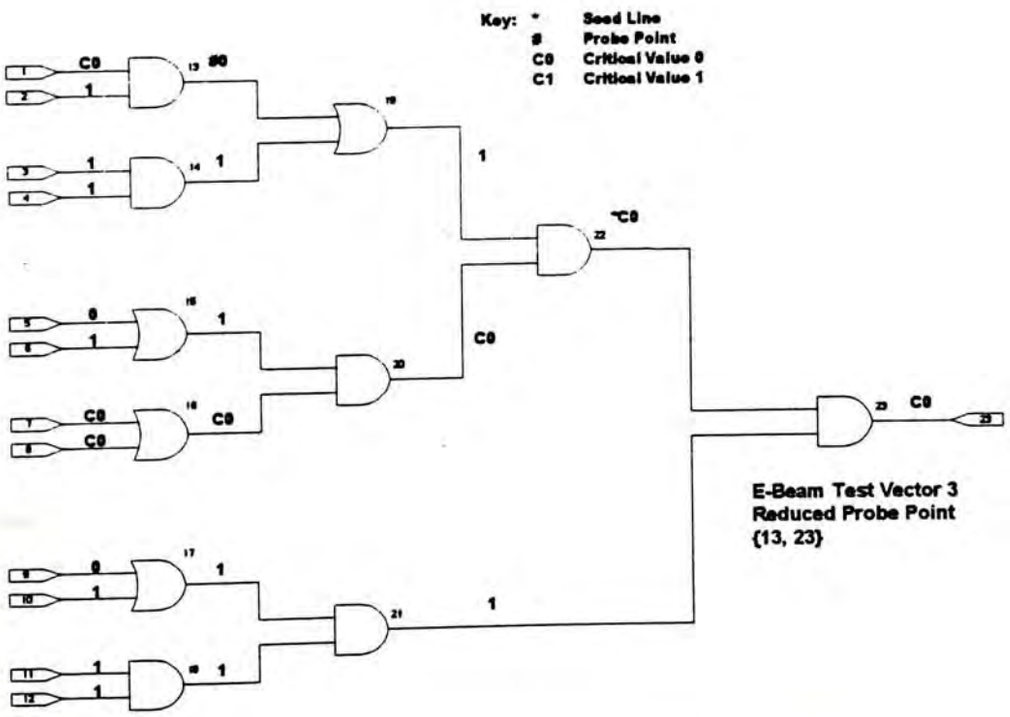


Figure C-3b Test Vector 3 for Circuit sc3 - Reduced Probe Points

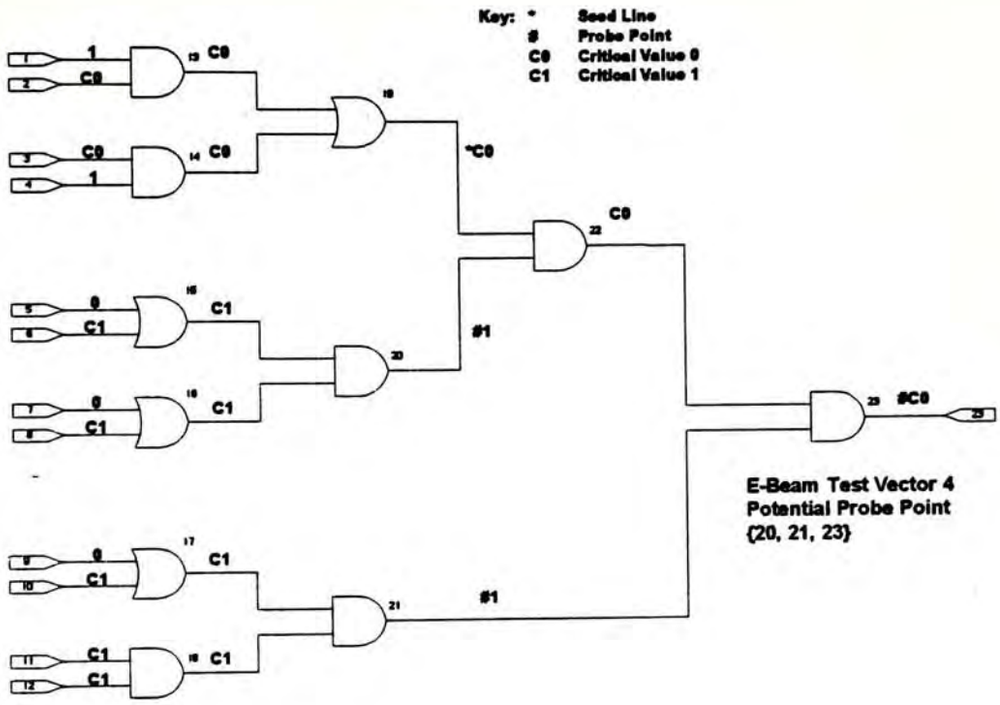


Figure C-4a Test Vector 4 for Circuit sc3 - Potential Probe Points

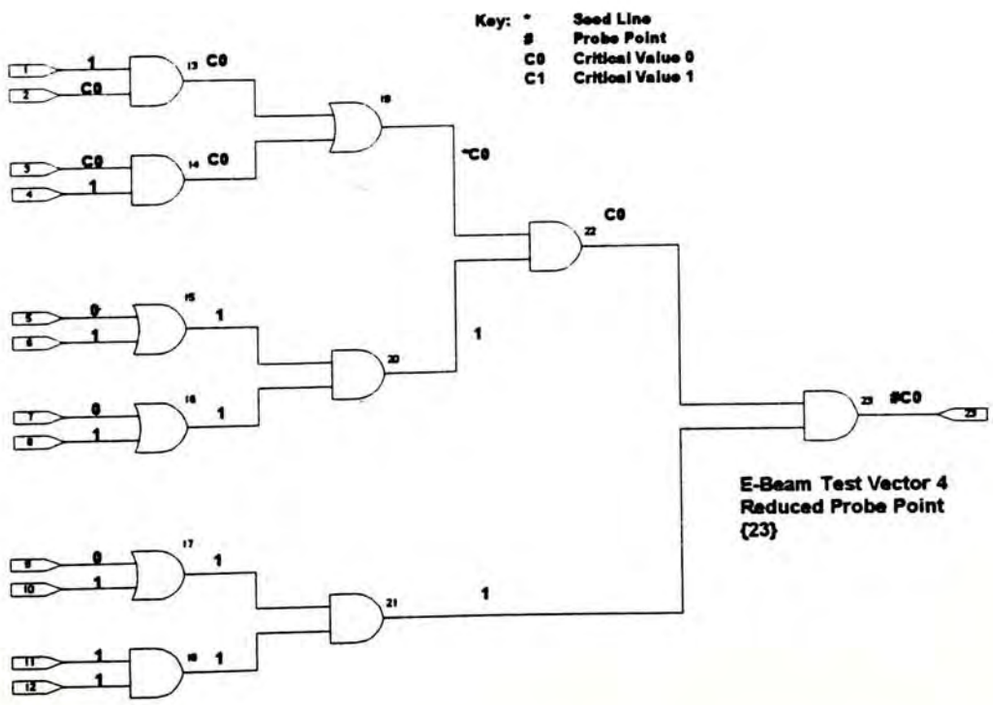


Figure C-4b Test Vector 4 for Circuit sc3 - Reduced Probe Point

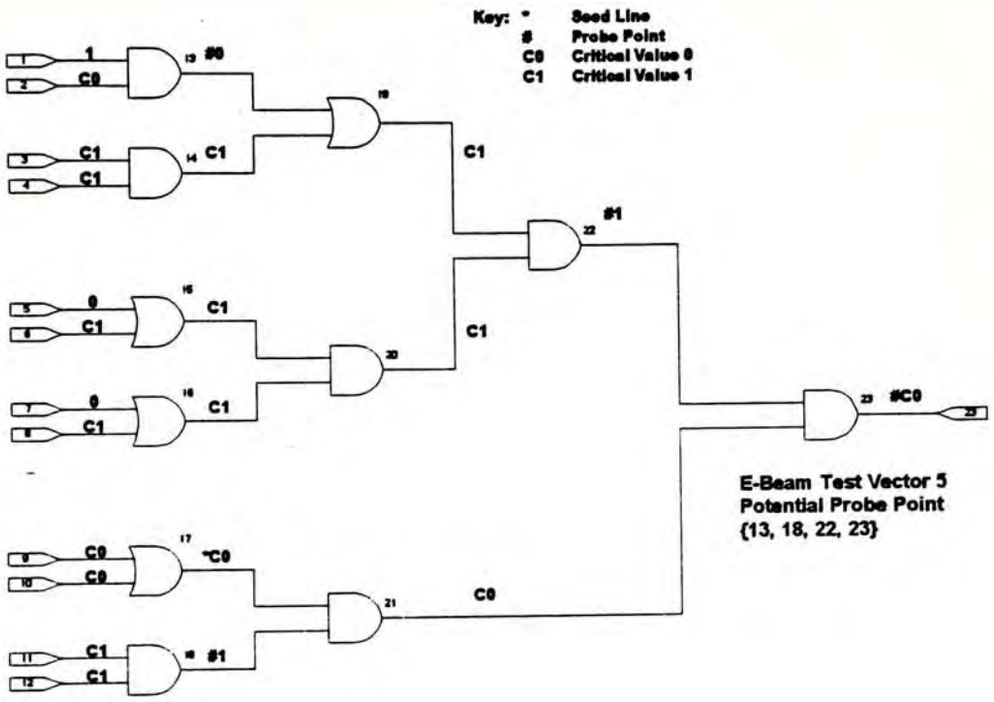


Figure C-5a Test Vector 5 for Circuit sc3 - Potential Probe Points

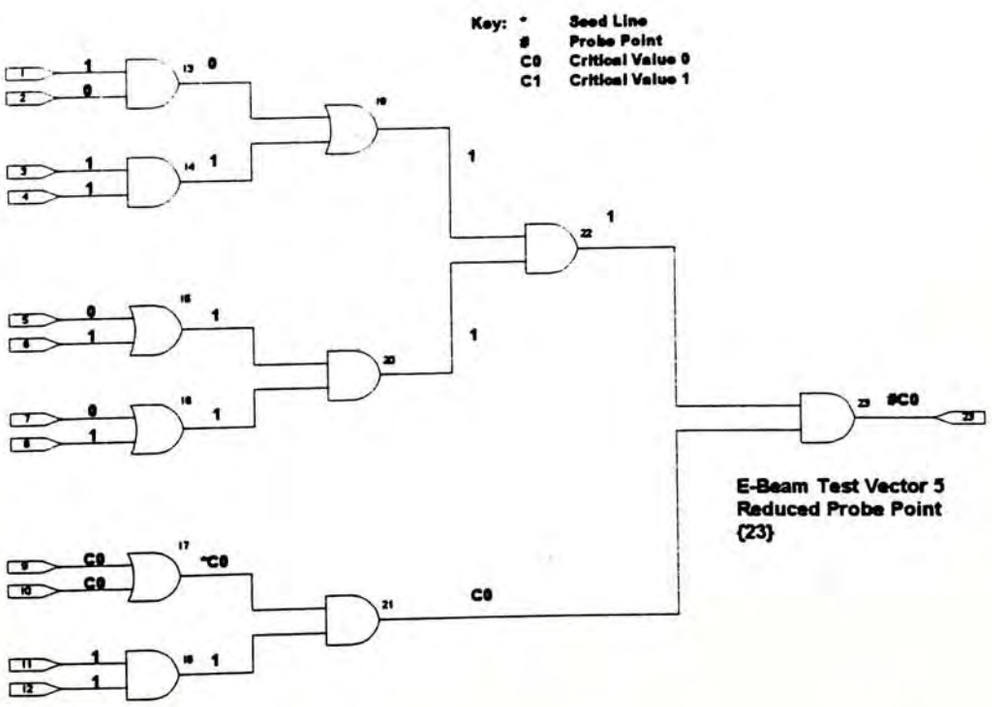


Figure C-5b Test Vector 5 for Circuit sc3 - Reduced Probe Point

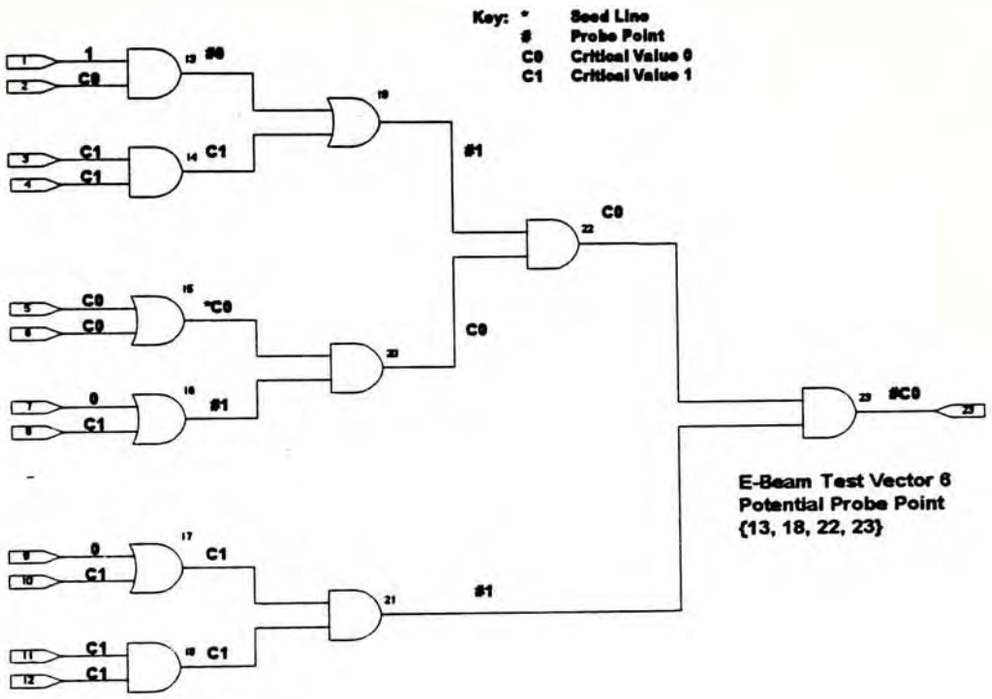


Figure C-6a Test Vector 6 for Circuit sc3 - Potential Probe Points

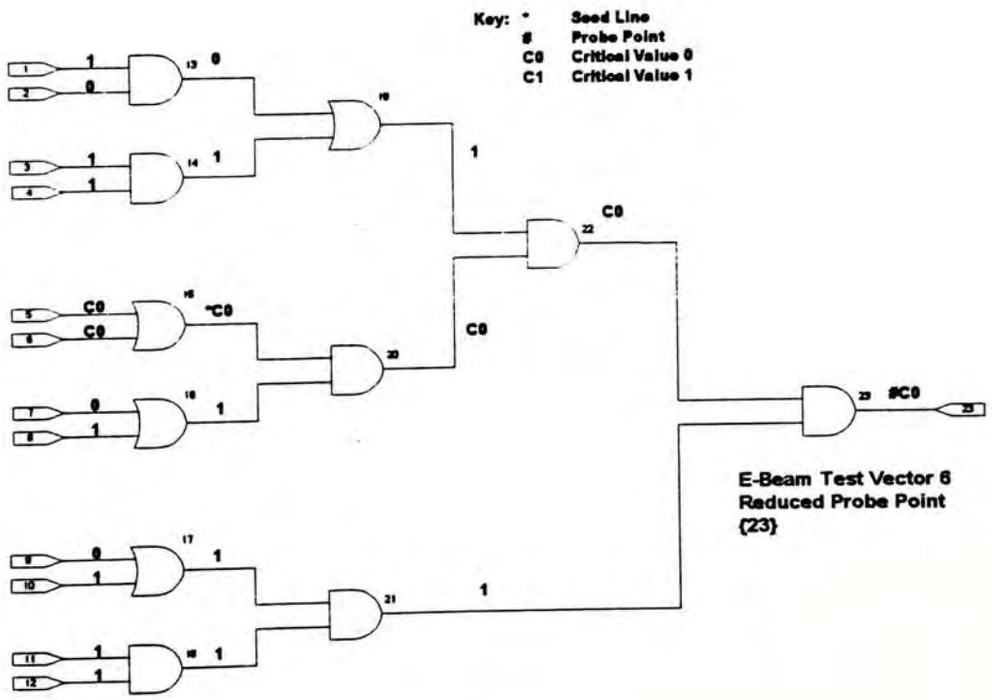


Figure C-6b Test Vector 6 for Circuit sc3 - Reduced Probe Point

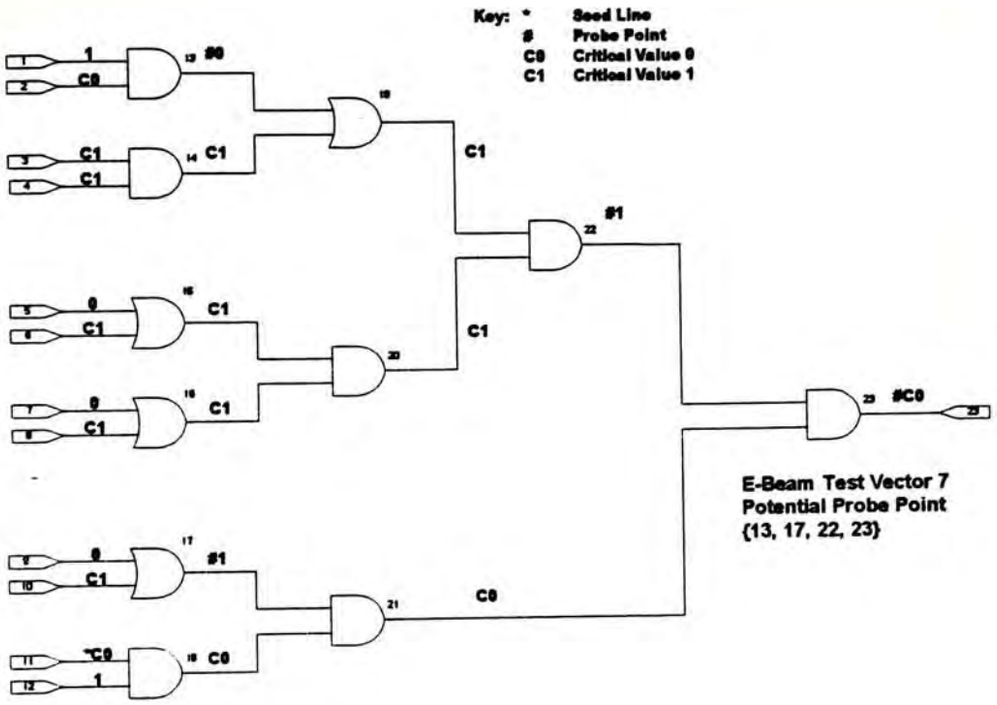


Figure C-7a Test Vector 7 for Circuit sc3 - Potential Probe Points

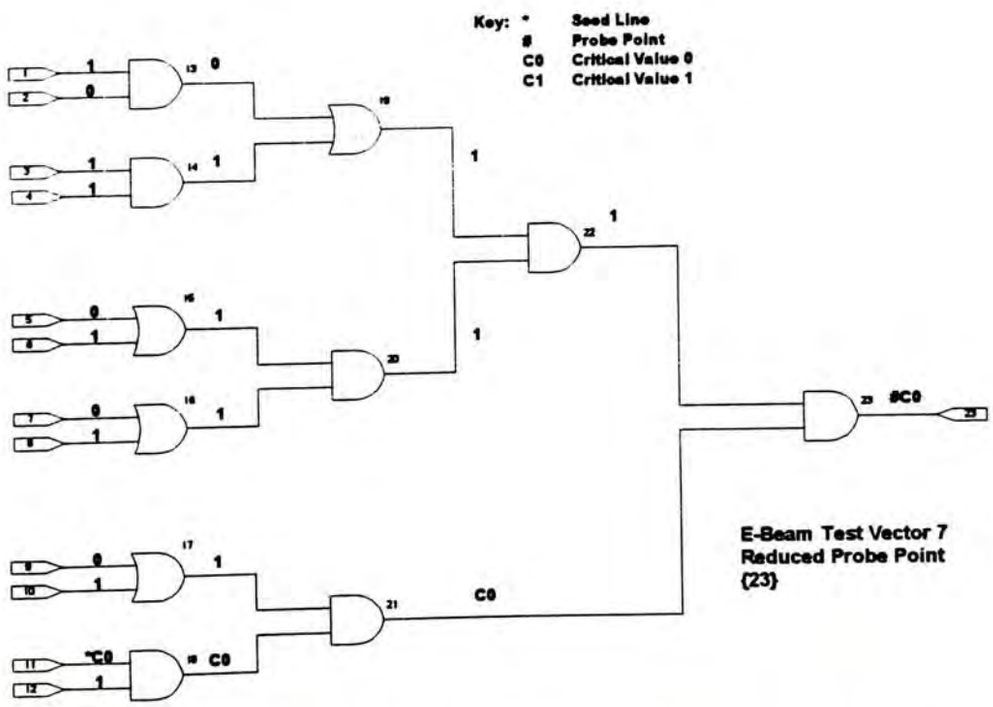


Figure C-7b Test Vector 7 for Circuit sc3 - Reduced Probe Point

	test vector			
	v1	v2	v3	v4
pin 1	1	1	0	1
pin 2	0	1	1	0
pin 3	1	1	1	0
pin 4	1	0	1	1
pin 5	0	1	0	0
pin 6	1	0	1	1
pin 7	0	1	0	0
pin 8	1	0	0	1
pin 9	0	1	0	0
pin 10	1	0	1	1
pin 11	1	1	1	1
pin 12	0	1	1	1
probe pts.	13,17,22,23	14,23	13,23	23

	test vector		
	v5	v6	v7
pin 1	1	1	1
pin 2	0	0	0
pin 3	1	1	1
pin 4	1	1	1
pin 5	0	0	0
pin 6	1	0	1
pin 7	0	0	0
pin 8	1	1	1
pin 9	0	0	0
pin 10	0	1	1
pin 11	1	1	0
pin 12	1	1	1
probe pts.	23	23	23

Table C-1 Results of Test and Probe Point Generation for Circuit sc3

REFERENCES

1. G. Swan, Y. Trivedi, and D. Wharton, "CrossCheck - A Practical Solution for ASIC Testability," Proc. IEEE International Test Conference, August 1989.

2. Kerry Pierce, Robert Lipp, Eric Chan and Tony Wong, "High Performance CMOS Array with an Embedded Test Structure," IEEE 1990 Custom Integrated Circuits Conference, pp.4.1.1-4.1.4, 1990.

3. D.W. Ranasinghe, "Advances in Electron Beam Testing," Testing and Diagnosis of VLSI and ULSI, Kluwer Academic Publishers, pp.509-526, 1988.

4. Teruo Tamama and Norio Kuji, "Integrating an Electron-Beam System into VLSI Fault Diagnosis," IEEE Design and Test, pp.23-29, August 1986.

5. D.W. Ranasinghe, "Device Testing and SEM Testing Tools," Testing and Diagnosis of VLSI and ULSI, Kluwer Academic Publishers, pp.469-508, 1988.

6. T. Tamama and N. Kuji, "Integrating an Electron-Beam System into VLSI Fault Diagnosis," IEEE Design & Test of Computer, pp.23-29, August 1986.

7. J. Mucha, "Bridging the Gap between Conventional and E-Beam Testing," Microelectronic Engineering, Vol. 16, pp.185-192, Special issue on Electron and Optical Beam Testing of Integrated Circuits.

8. Tohru Kazamaki, "Milestones of New-Generation of ATE," IEEE Design and Test, pp. 83-89, Oct. 1985.

9. A. Stivers and D. Ferguson, "Fault Contrast: A New Voltage Contrast VLSI Diagnosis Technique," Proceedings of IEEE International Reliability Physics Symposium, pp. 109-114, 1986.

10. D.T. Wang, "An Algorithm for the Generation of Test Sets for Combinational Logic Networks," IEEE Trans. on Computers, Vol. C-24, No. 7, pp.742-756, July 1975.

11. Schmitt, D. Winkler and B. Lischke, 'E-beam Testing of High-Speed Electronic Devices," Microelectronic Engineering, vol. 5, pp. 523-530, 1986.

12. B.L. Redmond, "Scanning Electron Microscope Operators Manual AMRAY Models 1820, 1830," AMRAY Incc., 1990.

13. Kirk J. Bertsche and Harry K. Charles, "The Practical Implementation of Voltage Contrast as a Diagnostic Tool," IEEE Proc. of IRPS, pp. 167-178, 1982.

14. M. Macari, K. Thangamuthu and S. Cohen, "Automated Contactless SEM Testing for VLSI Development and Failure Analysis," IEEE Proc. of IRPS, pp.163-166, 1982.

15. Norio Kuji, Teruo Tamama, and Takao Yano, "A Fully-Automated Electron Beam Test System for VLSI Circuits," IEEE Design and Test, pp.74-82, Oct. 1982.

16. C. Morandi, M. Vanzi, F. Bianco and R. Neri, " A PC-AT-Based System for the Acquisition of SEM Images," Scanning Vol. 11, pp.81-85, 1989.

17. J.C.H. Phang, K.K.W. Ong, D.S.H. Chan and T.S. Low, "SEM Image Collection and Processing System(SEMICAPS) Part I-Collection System," Proc. Int. Symp. Physical and Failure Analysis of Integrated Circuits, Singapore, pp.50-55, Oct. 1987.

18. Oliver C.S. Choy, R.Chan, L.K. Chan and C.F. Chan, "Conventional SEM for Digital Logic Verification," 5th International Symposium on IC Technology, System and Applications, pp.737-740, Sept. 1993.

19. Abramovici, P.R. Menon, and D.T. Miller, "Critical Path Tracing: An Alternative to Fault Simulation," IEEE Design and Test of Computers, Vol. 1, No. 1, pp.83-93, Feb. 1984.

20. J.P. Roth, W.G. Bouricius and P.R. Schneider, "Programmed Algorithms to Compute Tests to Detect and Distinguish Between Failures in Logic Circuits," IEEE Trans. on Electronic Computers, Vol. EC-16, No. 10, pp.567-579, Oct. 1967..

21. I.S.M. Chin, W.K. Chim, P.C.T. Koh, D.S.H. Chan, T.S. Low, J.C.H. Phang and P.Y.S. Ho, "Observation of IC surface Potentials with Voltage Contrast", Proc. NUS Symp. Applications of Scanning Microscopy, pp.82-87, Sept. 1988.

22. M.W. Roberts and P.K. Lala, "Algorithm to Detect Reconvergent Fanouts in Logic Circuits," IEE Proceedings, Part E, Computers and Digital Techniques, Vol. 134, pp.105-111, March 1987.

23. Fadi Maamari and Janusz Rajski, "A Method of Fault Simulation Based on Stem Regions," IEEE Trans. on Computer-Aided Design, Vol. 9, No. 2, pp.212-220, Feb. 1990.

24. Fadi Maamari and Janusz Rajski, "Reconvergent Fanout Analysis and Fault Simulation Complexity of Combinational Circuits," Electronic Letter, Vol. 23, No. 21, pp.1131-1133, Oct. 1987.

25. J.P. Roth, "Diagnosis of Automata Failures: A Calculus and A Method," IBM Journal of Research and Development, pp.278-281, Oct. 1966.

26. P. Goel, "An Implicit Enumeration Algorithm to Generate Tests for Combinational Logic Circuits," IEEE Trans. on Computers, Vol. C-30, pp.215-222, March 1981.

27. H. Fujiwara, "Logic Testing and Design for Testability," MIT Press, Cambridge, Mass. , 1986.

28. Kinch R. and Pottle C., 'Automatic Test Generation for Electron-beam Testing of VLSI circuits', Proc. IEEE ICCC, pp. 548-551, 1982.

29. Kinch, Richard John, "Automatic Test Generation for Electron-beam Testing of VLSI Circuits," Ph. D. thesis, School of Electrical Engineering, Cornell University, Ithaca, NY 14853, May, 1982.

30. P GGarino and M. Battu, "Automatic Selection of Optimal Probing Points for E-Beam Measurement," Mircroelectronic Engineering, Vol. 16, pp.111-119, 1992.

31. Menon P., Levendel Y. and Abramovici M., 'Critical Path Tracing in Sequential Circuits', Proc. Int. Conf. CAD, pp.162-165, 1988.

32. Menon P., Levendel Y. and Abramovici M., "SCRIPT: A Critical Path Tracing Algorithm for Synchronous Sequential Circuits", IEEE Transactions on Computer-Aided Design, vol. 10, no. 6, 1991.

33. Oliver C.S. Choy, L.K. Chan, R. Chan and C.F. Chan, "Test Generation for E-Beam Testing of VLSI Circuits," Second Asian Test Symposium, pp.101-106, Nov. 1993.

34. Oliver C.S. Choy, L.K. Chan, R. Chan and C.F. Chan, "Test Generation for E-Beam Testing of Digital Circuits," 5th International Symposium on IC Technology, System and Applications, pp.328-332, Sept. 1993.

35. F. Brglez and Fujiwara, "A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translator in Fortran," Proceedings of IEEE Int. Symposium of Circuits & Systems, 1985.