Scan Path Architecture for Low Power Testing

A dissertation submitted in partial fulfillment
of the requirements of M.Tech.(Computer Science)
degree of Indian Statistical Institute, Kolkata

by

Praveen Srivastava

under the supervision of

Prof. Bhargab B. Bhattacharya

Indian Statistical Institute,
Kolkata-700 108.

July 13, 2004

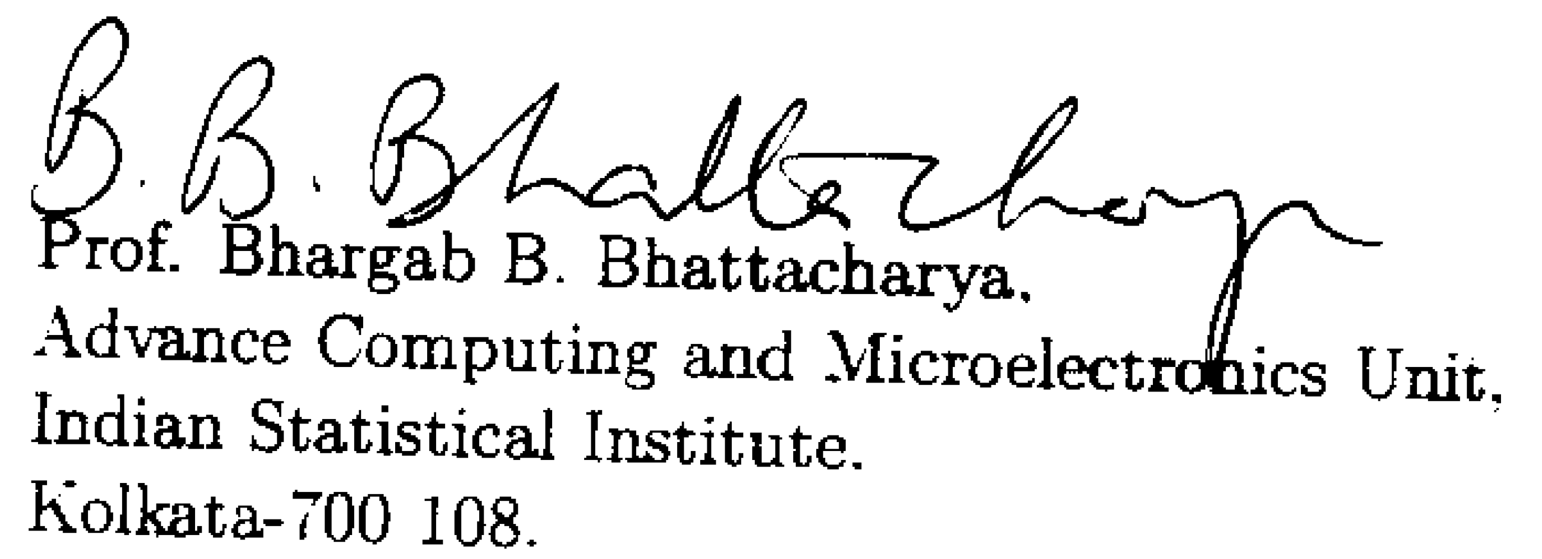
Indian Statistical Institute,

203, Barrackpore Trunk Road,

Kolkata-700 108.

Certificate of Approval

This is to certify that this thesis titled **"Scan Path Architecture for Low Power Testing"** submitted by **Praveen Srivastava** towards partial fulfillment of requirements for the degree of M.Tech in Computer Science at Indian Statistical Institute. Kolkata embodies the work done under my supervision.



Prof. Bhargab B. Bhattacharya.
Advance Computing and Microelectronics Unit.
Indian Statistical Institute.
Kolkata-700 108.

Acknowledgements

I take pleasure in thanking Prof. Bhargab B. Bhattacharya for his valuable guidance throughout the dissertation period. His encouraging words have always kept my spirits up.

I express my deep gratitude to my teachers in the course. I would like to thank Faculty and Research Scholars of Advance Computing and Microelectronics Unit. ISI. Kolkata. Finally I take this opportunity to thank my classmates and friends for their help, support and lighter moments throughout the course.

Praveen Srivastava

Abstract

Scan design has been applied to the problem of testing sequential circuits as a means of increasing the controllability and observability of the circuit. Yet the Scan based testing suffer from prolonged test application time and excessive test power due to numerous shift operations. During the test mode . filling the test data requires shifting the bits one by one into the scan chain .thus creating the increased switching activities in the flip-flops. Average test power can be minimized by minimizing the switching activities in the flip-flops. Modifying the scan chain judiciously may lower both the test application and average test power by minimizing the switching activities. In recent years number of scan architectures are proposed in order to minimize the switching activities and test application time. We have investigated the power consumption , area overheads and hardware overheads of full scanned version of ISCAS'89 benchmark circuits for different architecture namely Single Serial Scan Architecture and Double Tree Scan architecture.

Contents

0.1 Introduction 7
0.2 Background 8
0.2.1 VLSI Design Flow 8
0.2.2 Design for Testability 9
0.2.3 Scan Design 9
0.2.4 Scan Design Rules 10
0.2.5 Scan Design Testing 10
0.3 Low Power Scan Design 13
0.3.1 Power Dissipation Model 13
0.4 Double Tree Scan Architecture 14
0.4.1 Structure of DTS 14
0.4.2 DTS Architecture 16
0.4.3 Clock Controller 17
0.4.4 Naive Clock Controller 17
0.4.5 Hierarchical Clock Controller 18
0.4.6 Shift Controller 19
0.4.7 Power dissipation in DTS 21
0.4.8 Implementation of Hierarchical Clock Controller 22
0.5 Experiments and Results 23
0.5.1 VHDLCodeGen Tool 27
0.5.2 Floor Planning Report for s298.bench(DTS) 28
0.5.3 Hope Fault Simulation Report for s344.scan 29
0.5.4 RTL Schematic View of s298 for DTS Architecture 30
0.5.5 Final Layout of the Circuit s27 for DTS Architecture 31
0.6 Conclusion 32

List of Figures

1 VLSI Design Flow 8
2 D-type Flip-flop and scan D-type Flip-flop 10
3 Scan Design Testability 11
4 Double Tree Scan DTS(2) 14
5 Pruning of DTS 15
6 Pruning of DTS(2) 15
7 Serial Concatenation for f=16 and f=116 16
8 DTS Architecture 16
9 Scan Path in DTS(2) 17
10 Naive Clock Controller for DTS(3) 18
11 hierarchical Clock Controller 18
12 Hierarchical Clock Controller for DTS(3) 19
13 Node used in the Algorithm for hierarchical clock controller 22
14 ISCAS'89 Circuit Experimental Flow 23
15 Area Overhead and Switching Activities 25
16 VHDL CodeGen GUI 27
17 RTL Schematic View of s298 for DTS Architecture 30
18 Final Layout of the circuit s27 for DTS Architecture 31

List of Tables

1 Complete DTS(k) Scan chain parameters 15
2 Shift Controller for DFL Mode 20
3 **Total Internal Area(μ^2)** 24
4 **Worst Case Switching Activity per test Vector** 24
5 DTS(k) scan chain parameters 25
6 **Serial Concatenation of ISCAS'89 circuits** 25

0.1 Introduction

Testing VLSI circuits bridges the gap between the imperfection of integrated circuit (IC) manufacturing and consumer expectations of flawless products. From customer point of view quality and cost are two major requirements. Hence manufacturer needs to test the completed product in order to ensure correct functioning of the product but exhaustive testing increases the testing cost.

The basis of all testing techniques is to apply predefined sets of inputs. called test vectors, to a circuit and compare the output observed with the patterns that a correctly functioning circuit is supposed to produce. The challenge is to derive a relatively small number of test vectors that provide an adequate indication that the circuit is correct. The fault coverage of a test is defined as the percentage of faults covered by the test related to the total number of assumed faults.

Scan based design is a structural design for test technique that has been widely accepted in industry . The basic idea of this method is to convert all or part of the internal registers of the circuit under test (CUT) into scan registers such that the controllability and observability of the CUT can be enhanced and the test generation complexity can be greatly reduced. However, it is well-known that the test application time of a scan system is proportional to the length of the scan chain. In a modern VLSI circuit, the number of internal registers can be in the range of thousands or even higher. Hence high product test cost may become a major concern in determining whether a scan based design should be used or not.

Special structures, called design for testability (DFT) structures is introduced by designers , where during the design phase of a digital system for improving its testability. Several such approaches have been standardized and widely accepted. However, all those approaches entail an overhead in terms of additional silicon area and performance degradation. Therefore it will be highly beneficial to develop DFT solutions that not only are efficient in terms of testability but also require minimal amount of overhead.

0.2 Background

0.2.1 VLSI Design Flow

The design flow of VLSI circuits is divided into three main steps: specification, implementation and manufacturing

1. ***Specification***: Specification is the step of describing the functionality of the VLSI circuit. The specification is done in hardware description languages (HDLs), such as VHDL or Verilog in two different design domains, the behavioral domain or the structural domain, at various levels of abstraction.

2. ***Implementation***: Implementation is the step of generating a structural netlist of components that perform the functions required by the specification.

3. ***Manufacturing***: Manufacturing is the final step of the VLSI design flow and it results in a physical circuit realized in a fabrication technology with transistors connected as specified by implementation.

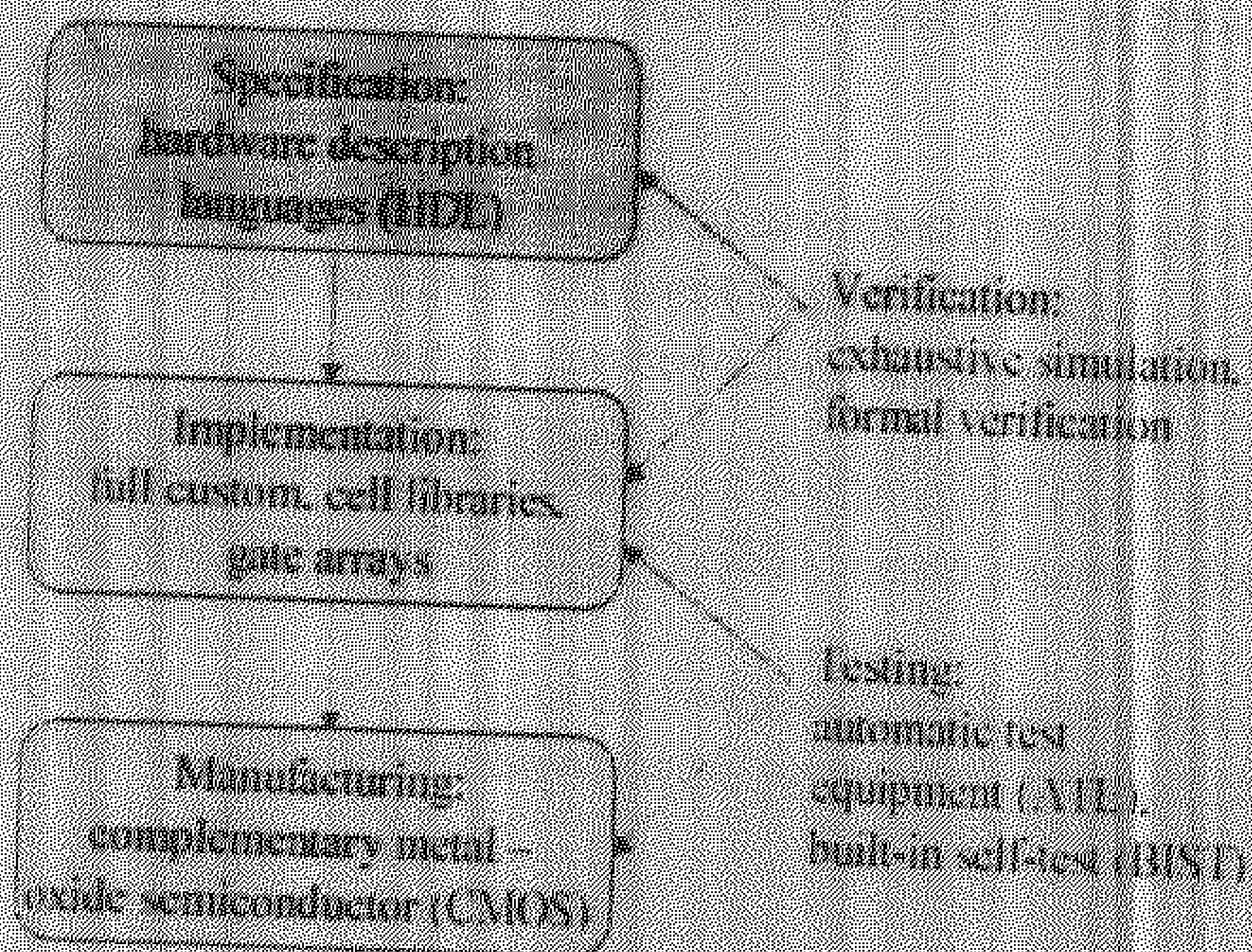


Figure 1: VLSI Design Flow

Verification involves comparing the implementation to the initial specification. If there are mismatches during verification, then the implementation may need to be modified to more closely match the specification.

Testing assures that the function of each manufactured circuit corresponds to the function of the implementation.

0.2.2 Design for Testability

Test generation and application can be more efficient when testability is already considered and enhanced during the design phase. The aim of such an enhancement is to improve controllability and observability with minimal area and performance overhead. Controllability and observability together with predictability are the most important factors that determine the complexity of deriving a test set for a circuit.

Controllability is the ability to establish a specific signal value at each node in a circuit by setting values on the circuits inputs.Thus the controllability refers to how easy it is to control a certain wire of the CUT (Circuit under test) , i.e., to put a 0 or a 1 on the wire by PI assignment(s).

Observability, on the other hand, is the ability to determine the signal value at any node in a circuit by controlling the circuits inputs and observing its outputs.The observability refers to how easy it is to observe a certain wire of the CUT, i.e., to deduce the signal value it carries from those of the POs. For sequential circuits, some have added predictability, which represents the ability to obtain known output values in response to given input stimuli. The factors affecting predictability include initialization, races, hazards, oscillations, etc. In this sense, we may formulate testability as follows:
testability = controllability + observability + predictability.

DFT techniques, used to improve a circuits controllability and observability, can be divided into two major categories:

1. Ad Hoc Techniques :DFT techniques which are specific to one particular design and cannot be generalized to cover different types of designs. Typical examples are test point insertion and design partitioning techniques.

2. Structured Techniques DFT techniques are techniques that are reusable. well defined and can be even standardized. (LSSD, Scan Based , Boundary Scan)

To cope with the problems caused by global feedback and complex sequential circuits, several different DFT techniques have been proposed. One of them is internal scan.

0.2.3 Scan Design

The basic idea of Scan Design method is to convert all or part of the internal registers of the circuit under test (CUT) into scan registers such that the controllability and observability of the CUT can be enhanced and the test generation complexity can be greatly reduced. However, it is well-known that the test application time of a scan system is proportional to the length of the scan chain. In a modern VLSI circuit, the number of internal registers can be in the range of thousands or even higher.

Hence high product test cost may become a major concern in determining whether a scan based design should be used or not.

For designing the scan capability in the circuits . only D type flip-flop is used .a typical flip-flop is show in figure. The DFFs are replaced by scan flip flops as show in figure A multiplexer and two new signals scan data SD and test control TC, are added to the D flip-flop. The original data input D is stored in the flip-flop when TC is 1 and SD is stored when TC is 0.

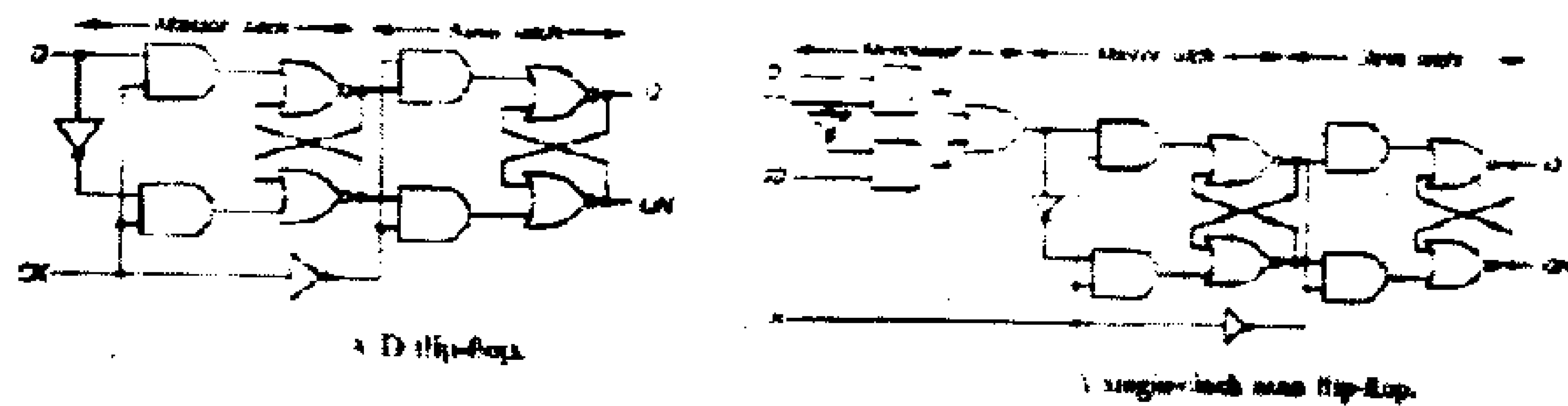


Figure 2: D-type Flip-flop and scan D-type Flip-flop

0.2.4 Scan Design Rules

In order to make the circuits as a scan-testable. it is important that the designer must adhere with certain rules. These rules are termed as Scan Design rules. The following four rules are useful for the design.

1. Only D-type flip-flops should be used.
2. At least one primary input pin must be available for test. if no extra pins are available then any normal primary input can be used as SCANIN.
3. All flip-flops clock must be controllable from primary inputs. This rule is necessary for flip-flops to function as scan register.
4. Clocks must not be feed data inputs of flip-flops. A violation of this rule can potentially lead to race condition.

0.2.5 Scan Design Testing

Testing of scan design done in two phase.

1. First Phase : Testing of scan registers by a shift test. The circuit is set in scan mode by setting the TC(Test Control) value as 0. Now flip-flops forms as a shift registers between SCANIN and SCANOUT. A toggle sequence of 0011001100... of length $n_{sff}+4$, where n_{sff} is the total number of flip-flops is

applied at SCANIN. This sequence produces all four transition 0→ 0,0→1,1→0 and 1→1 in each flip flop and shifts the outputs to the observable SCANOUT point.

2. Second Phase : Testing of stuck-at-faults in combinational circuits. The test vector is generated by ATPG program.The state of the flip-flops is set in the test mode (TC=0) by loading the flip-flops through SCANIN. Primary inputs are loaded with test vectors in the normal mode i.e. when TC=1 and primary output is observed. whereas next states of flip-flops is observed while loading of next test vector as shown in figure.

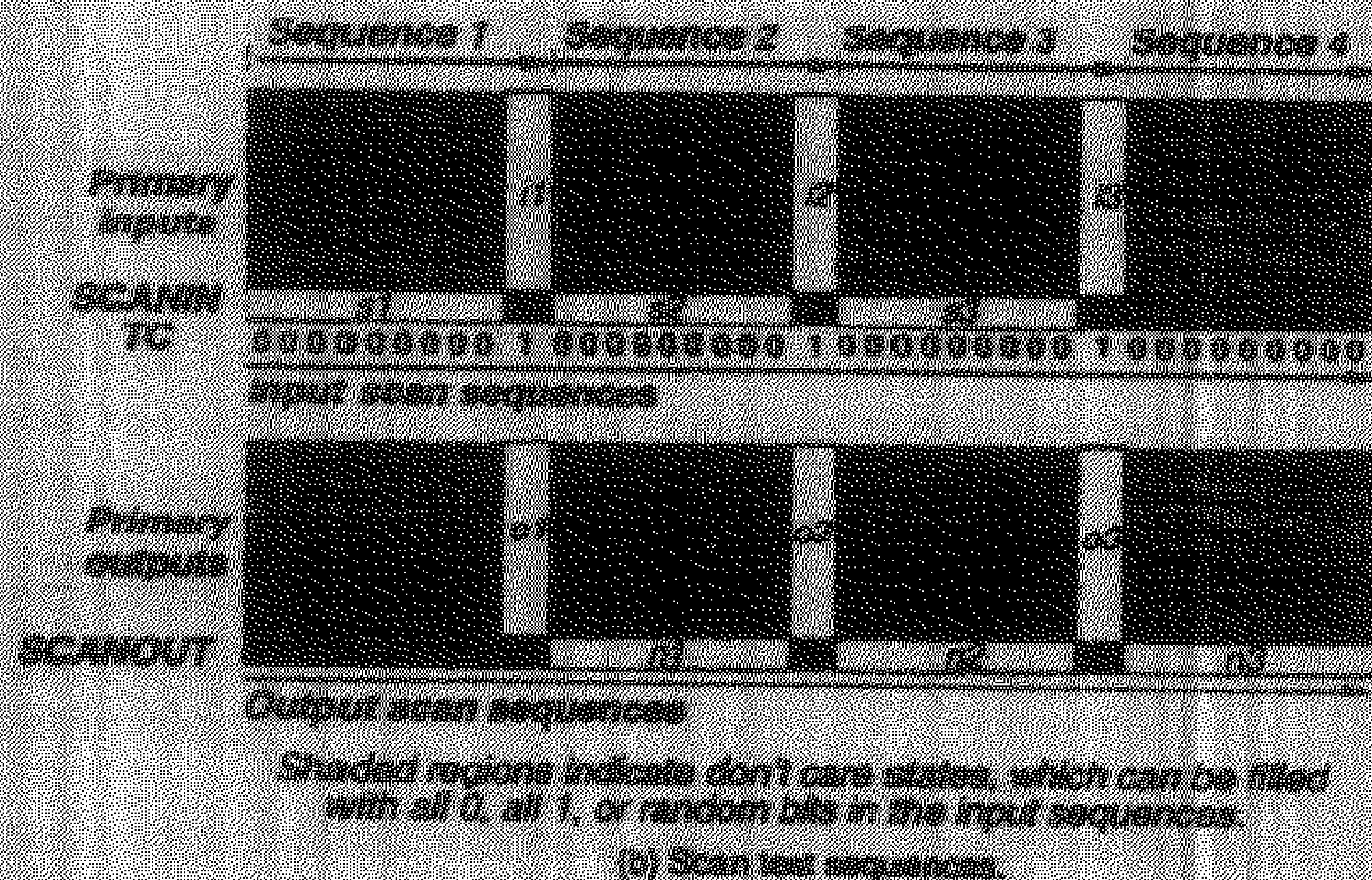


Figure 3: Scan Design Testability

A scan-based test cycle has two distinct cycles: shift and capture. Shifting a test pattern into the scan chain occurs simultaneously with shifting out circuits response to the previous test pattern. In the capture cycle, the test pattern, loaded in the the scan chain during the shift cycle, is applied to the circuit under test and the response of the circuit is captured into the scan chain.

The following parameters are important factors for any type of scan design.

1. ***Test Application Time*** : Test Application time is the clock period required to complete the scan for all test vectors. This usually depends upon the number of scan flip-flops between SCANIN and SCANOUT terminal. If n_{sff} is the number of scan flip-flops between SCANIN and SCANOUT and V is the total number of test vectors . Then

$$\text{Scan Test Length} = (V+1)\times n_{sff}+V$$

If testing of scan registers (first phase) is also included in the scan test length formula can be written as

$$\text{Scan Test Length} = n_{sff}+4+(V+1 \times n_{sff}-V$$

2. ***Hardware Overhead***: One of the penalty of scan design is hardware overhead which increases the area overhead as well. But actual area overhead depends on layout details of the circuits. It is useful to estimate the hardware overheads in terms of gate overhead. Each flip-flop add an overhead of four gates (used in the multiplexors). The total gate overhead thus computed as follows.

$$\text{Gate overhead of scan} = \frac{4 \times n_{sff}}{n_g} \times 100\%$$

3. ***Performance Overhead*** : Scan design has performance degradation even in normal mode due to delay caused by the multiplexors attached with flip-flops . The delay due to multiplexors is equivalent to two gate-delays in all clocked path. In general, scan design can reduce the clock speed by 5 to 10%. However the skewness of the clock signals should be carefully taken care of for the correct operation.

4. ***Power Consumption*** : Power consumption depends mainly on switching activity. Switching Activity is the transition either from $1 \rightarrow 0$ or from $0 \rightarrow 1$ in scan flip-flops. .

The general idea behind internal scan is to break the feedback paths and to improve the controllability and observability of the memory elements by introducing an over-laid shift register called scan path. Despite the increase in fault coverage. there are some disadvantages with using scan techniques:

1. Increase in silicon area,
2. Larger number of pins needed,
3. Increased power consumption,
4. Increase in test application time,
5. Decreased clock frequency.

0.3 Low Power Scan Design

During scan testing, all flip-flops in the design are clocked in every clock cycle. During normal operation only the flip-flops which have to be updated are clocked, while all remaining flip-flops are disabled by the clock gating logic. Hence, internal switching activity during testing can exceed the level corresponding to the normal operation of the circuit.Sustained intense switching activity causes overheating and electro-migration which can permanently damage the chip under test or seriously affect its reliability.Several methods aiming to solve power-related problems associated with scan-based test have been proposed recently. They fall into the following broad categories:

1. ***Low transitions test patterns*** : These methods reduces the number of transitions in the scan-in vectors, and consequently the shift power component caused by scan-in in transitions. These methods does not have any control on transitions caused by the scan-out data.Thus overall power consumption reduction is not guaranteed. Moreover these methods do not address the issue of high peak power consumption.

2. ***Power conscious ATPG algorithms*** These are special ATPG algorithms which aim to decrease the number of transitions in scan-in and scan-out vectors for shift power reductions

3. ***Scan Chain Partitioning*** These methods are used to partition the scan chain into number of scan chain length which ultimately reduces the complexity of the switching activities in the scan chain. Thus the reduction of average power can be achieved.Methods based on scan chain partitioning appear to be efficient solutions in terms of shift power reduction.

0.3.1 Power Dissipation Model

Power dissipation in digital CMOS circuits is divided into static and dynamic power. The static power dissipation is considered negligible as compare to dynamic power dissipation. If the gate is the part of circuits and controlled by global clock, then dynamic power dissipation P_d in the gate is calculated as

$$P_d = 0.5 \times C_{load} \times (V_{DD}^2/T_{cyc}) \times N_G$$

where C_{load} is the load capacitance, V_{DD} is the supply voltage, T_{cyc} is the global clock cycle, and N_G is the total number of gate output transitions. $(0 \rightarrow 1, 1 \rightarrow 0)$ Since supply voltage V_{DD} and global clock cycle T_{cyc} are design constraints, they are not under designer control. Hence C_{load} and N_G are the quantitative measure for power dissipation . It has been assumed that load capacitance for each combinational gate is equal to number of fan-outs.

0.4 Double Tree Scan Architecture

In order to reduce the power minimization due to switching activities during loading of test vec..ors and unloading of response , a double tree scan path archit..ture is proposed. The double tree scan model drastically reduces the scan-shift and clock activity during testing.

0.4.1 Structure of DTS

Complete DTS

In complete double tree scan (DTS) model the scan flip-flops is organized in a structure of DTS by gluing the two complete binary trees at the leaf nodes. The figure is showing the DTS(2) where 2 is the level of the structure.

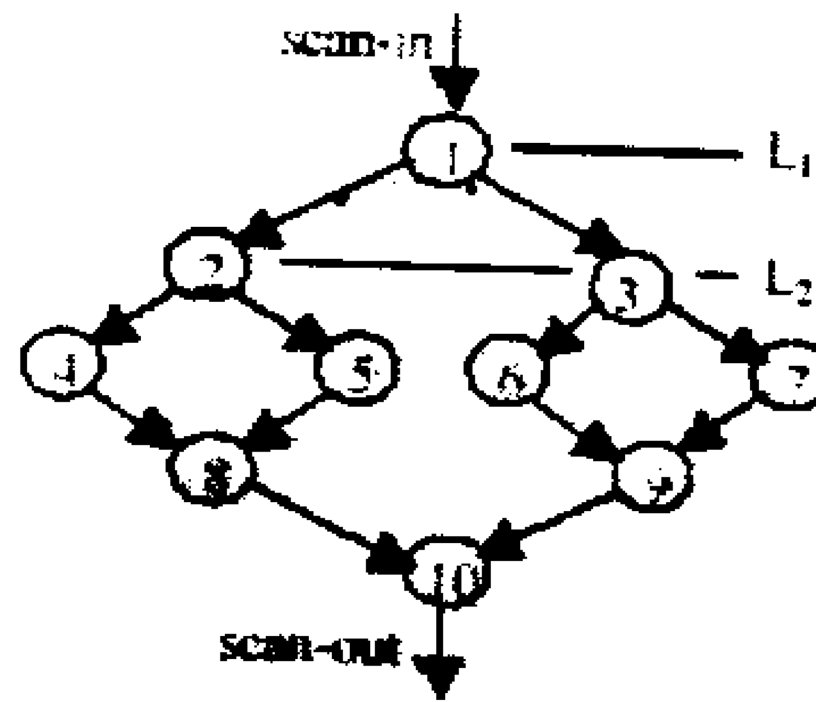


Figure 4: Double Tree Scan DTS(2)

A complete binary tree of level k consists of 2^k leaf node and $2^k - 1$ internal nodes. Thus s full double tree DTS(k) consists of $3 \times 2^k - 2$ number of nodes. Each node of the tree represents the scan flip-flops. For each node of in-degree 2 i.e. the bottom half of the double tree a $2 \rightarrow 1$ multiplexer is needed to select the predecessor flip-flop during scan operation. Full DTS(k) structure is hierarchical in nature as it can be constructed by taking two copies of DTS(k-1) and adding two nodes as source and sink. Hence

$$\text{DTS(k)} := 2 \times DTS(k-1) + \text{source node} + \text{sink node}$$

The full DTS(k) which has $3 \times 2^k - 2$ scan flip-flops and total number of scan path ia DTS(k) is equal to the number of leaf nodes of full binary tree and length of each scan path is equal to $2 \times k + 1$.

Table 1: Complete DTS(k) Scan chain parameters

Total number of flip-flops	$3 \times 2^k - 2$
Total number of Scan Paths	2^k
Scan Path Length	$2 \times k + 1$

DTS for arbitrary number of Flip-Flops

In an actual scan design the number of flip-flops need not to satisfy the number as required by DTS(k). Thus we have incomplete DTS which can be constructed as follows

1. Pruning of full DTS
2. Serial Concatenation of several full DTS.

1. Pruning of Complete DTS : We first choose the DTS which requires the number of flip-flops just greater than the given flip-flops. If f is the given flip-flops then choose k such that $(3 \times 2^k - 2) < f < (3 \times 2^{k-1} - 2)$ Then deletion of the node is performed from the innermost hierarchy of the DTS(k)

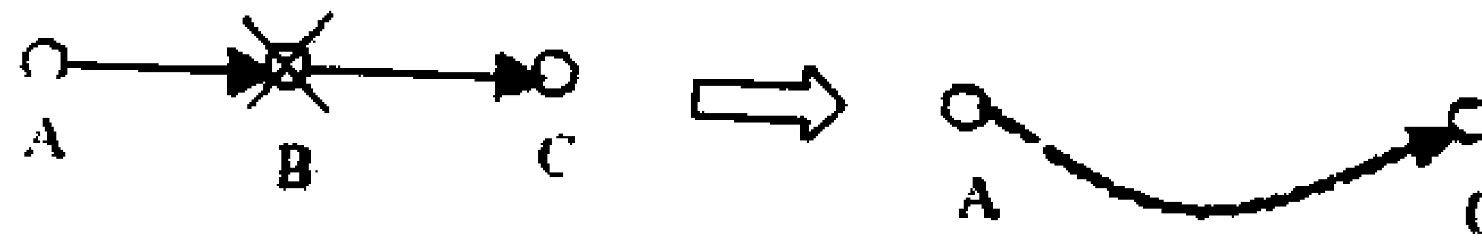


Figure 5: Pruning of DTS

Generally the deletion of node from the innermost hierarchy is performed in such a way that we could retain as many small complete DTS blocks as possible. The resulting DTS may be asymmetric in nature.The following example illustrates the process of pruning of starting DTS which has 10 nodes.

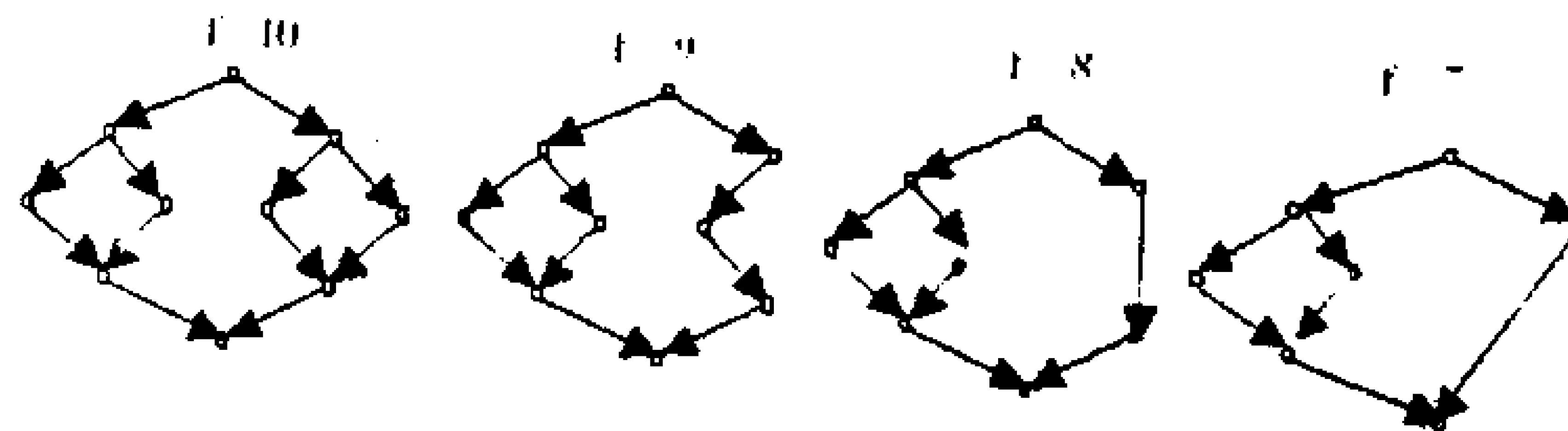


Figure 6: Pruning of DTS(2)

2. Serial Concatenation of several full DTS : In this method we choose DTS(k-1) such that number of flip-flop in DTS(k) is just smaller than f the given number of scan flip-flops. i.e. $(3 \times 2^k - 2) < f < (3 \times 2^{k-1} - 2)$. For remaining nodes the iteration of same process is done till all the nodes fulfills the number as required by f. The figure illustrates the serial concatenation of complete DTS for f=16 and f=116.

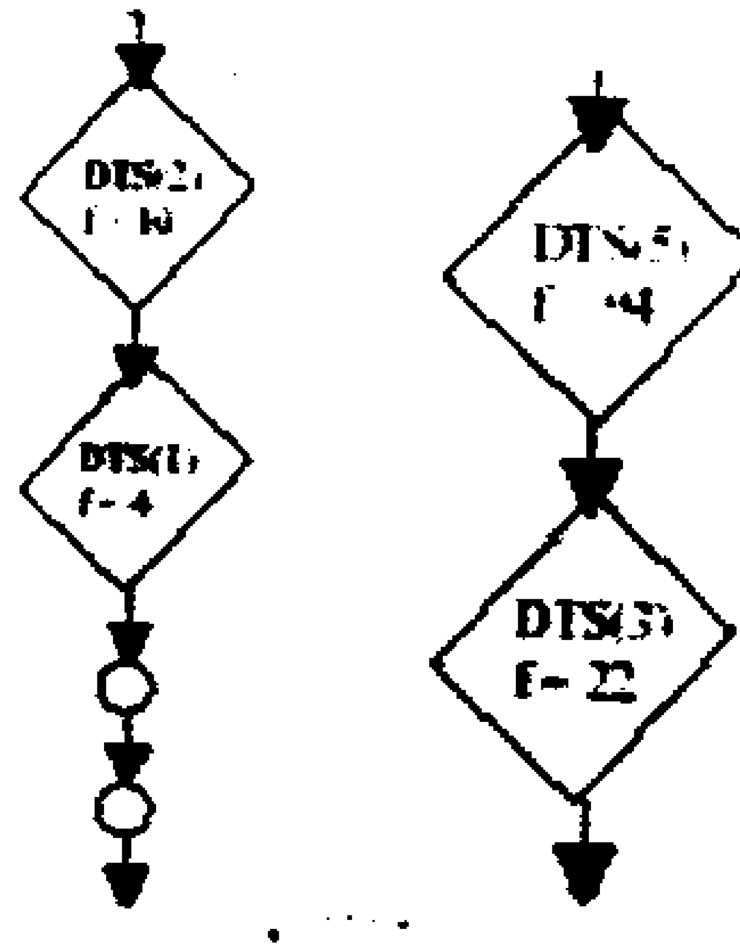


Figure 7: Serial Concatenation for f=16 and f=116

0.4.2 DTS Architecture

The DTS architecture consists of 3 subsections

1. DTS Structure
2. Clock Controller
3. Shift Controller

The DTS architecture is shown in following figure.

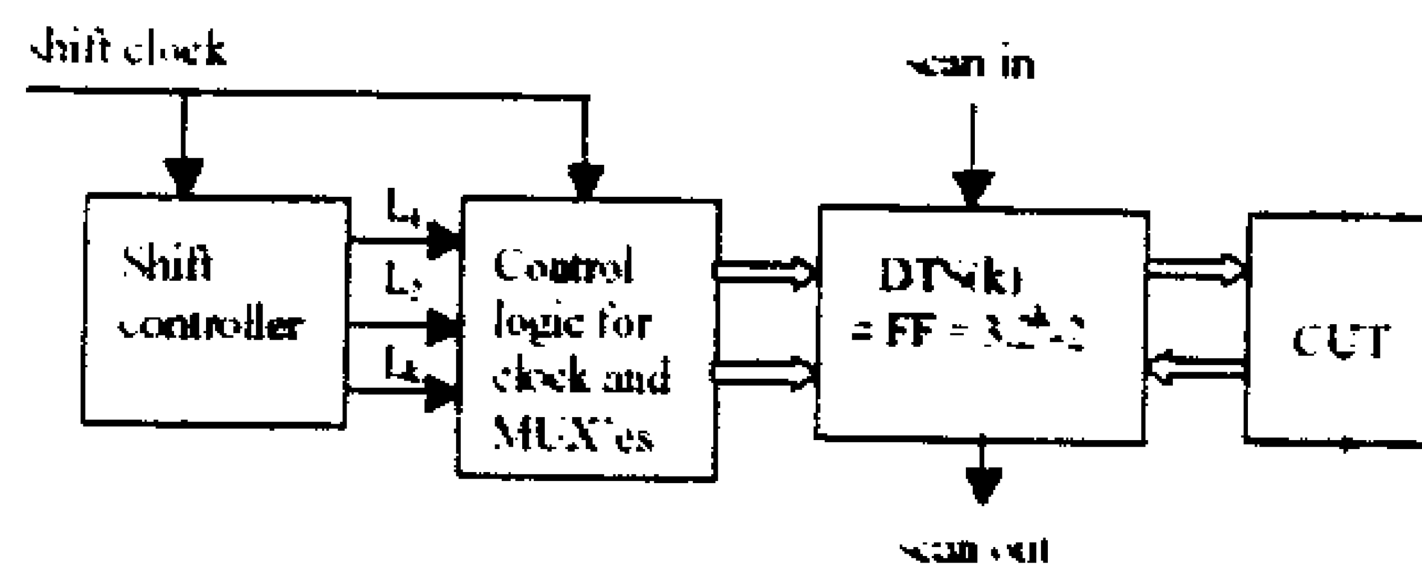


Figure 8: DTS Architecture

0.4.3 Clock Controller

During scan-shift. one should be able to select a single path from the source to sink in the DTS for loading and unloading the flip-flops. Paths in a DTS(k) may be designated by a k-bit vector $L_1, L_2, ..L_k$, where the path number is simply the binary number corresponding to this vector. thus in DTS(2) there exist 4 paths. which is show from different color in figure. each path selection is based on L_1 and L_2 value as follows

$$L_1L_2 = 00 \longrightarrow \text{Path-0} : 1 \rightarrow 2 \rightarrow 4 \rightarrow 8 \rightarrow 10$$
$$L_1L_2 = 01 \longrightarrow \text{Path-1} : 1 \rightarrow 2 \rightarrow 5 \rightarrow 8 \rightarrow 10$$
$$L_1L_2 = 10 \longrightarrow \text{Path-2} : 1 \rightarrow 3 \rightarrow 6 \rightarrow 9 \rightarrow 10$$
$$L_1L_2 = 11 \longrightarrow \text{Path-3} : 1 \rightarrow 3 \rightarrow 7 \rightarrow 9 \rightarrow 10$$

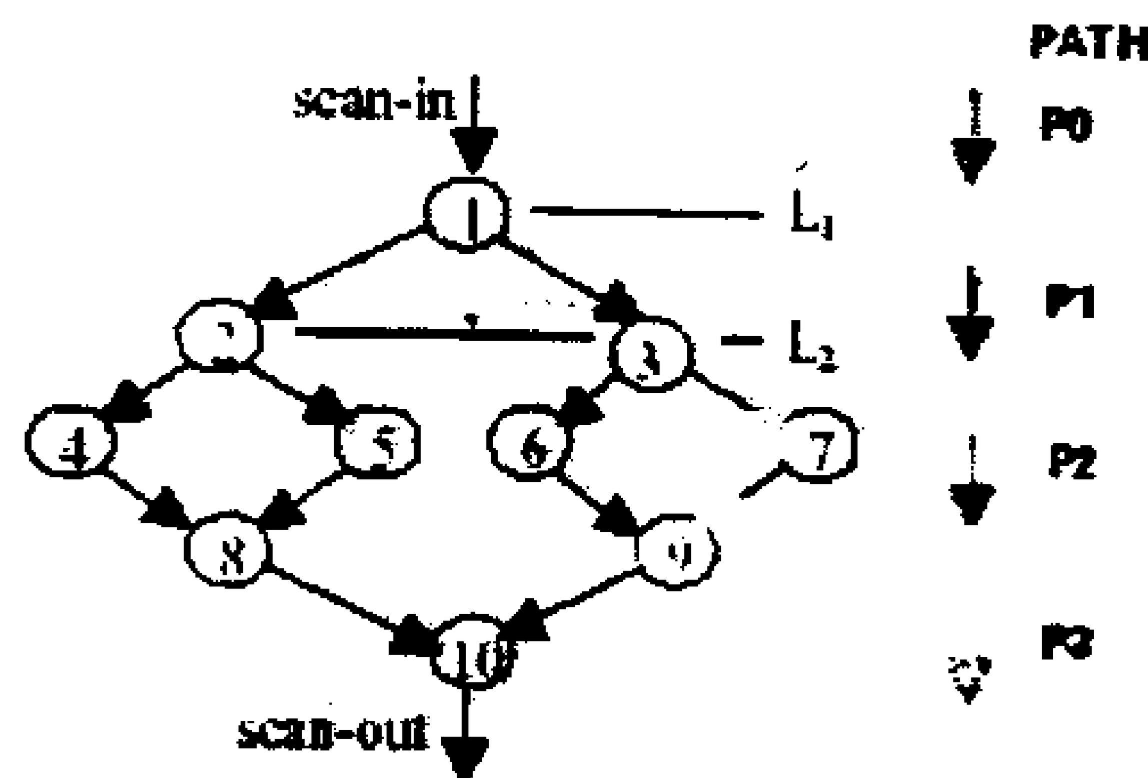


Figure 9: Scan Path in DTS(2)

There are two ways in which the clock controller can be designed Naive Clock controller and Hierarchical Clock Controller.

0.4.4 Naive Clock Controller

A straightforward way to design a clock gating logic for DTS(k) would be use k control lines $L_1, L_2, ..L_k$, and a binary tree circuit consisting of $2^k - 1$ $(1 \rightarrow 2)$ DEMUX units for clock routing mechanism. For each node of DTS with out-degree 2 (i.e., a fork node) in the top half of the DTS, a $1 \rightarrow 2$ demultiplexer is needed to route the clock to an appropriate successor flip-flop. The naive clock controller for DTS(3) is shown in figure.

The number of MUX units needed for join nodes of the bottom half of the tree is $2^k - 1$. Hence. for a DTS(k) with $(3 \times 2^k 2)$ flip-flops. the additional hardware overhead would be a total of $(2 \times 2^k 2)$ DEMUX/MUX units. However, the maximum fan-out of a control line would be 2^{k-1}, which may not be acceptable for a system with a very large number of scan flip-flops.

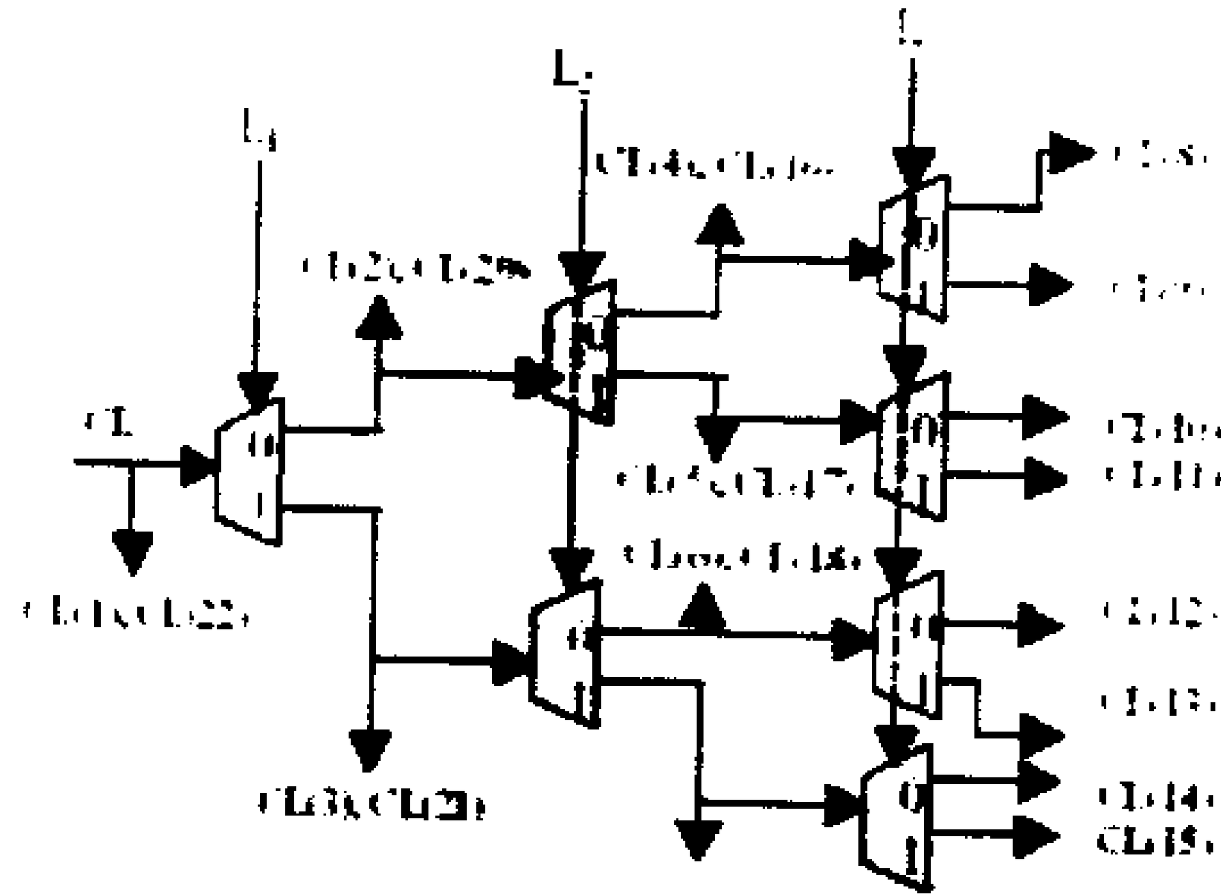


Figure 10: Naive Clock Controller for DTS(3)

0.4.5 Hierarchical Clock Controller

The Hierarchical clock controller design sort out the problem of high fan-out in naive clock controller.In this method DTS(**k**) can be constructed by taking two copies of DTS(k-1) and adding k ($1 \rightarrow 2$) DMUX units. Thus If D(k) denotes the number od DMUX units then

$$D(k)=2 \times D(k-1) + k$$

On solving above iteration equation the number of DMUX units required to construct the Hierarchical clock controller is equal to:

No. of DMUXes in DTS(k) clock controller D(k)= $2^{k+1} - (k+2)$

The generic hierarchical clock controller is shown in below figure.

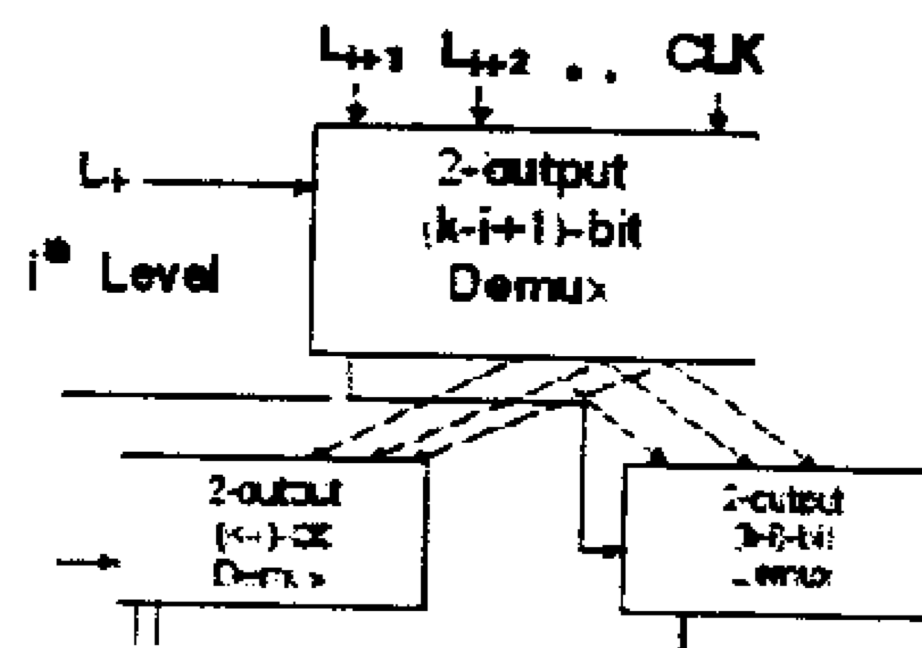


Figure 11: hierarchical Clock Controller

A hierarchical controller of DTS(k) has several advantages. First, the maximum fanout of a control line is k. Second, it takes care of driving clock signals to all

the scan flip-flops, and further, each output of a DEMUX unit drives at most two signals. Hence, it obviates the need of a separate clock tree for buffering. Third, for each shift clock, only a single path of length k is activated through the interior of the clock tree, and hence, the additional power loss in the tree is low. The following diagram shows the hierarchical clock controller for DTS(3).

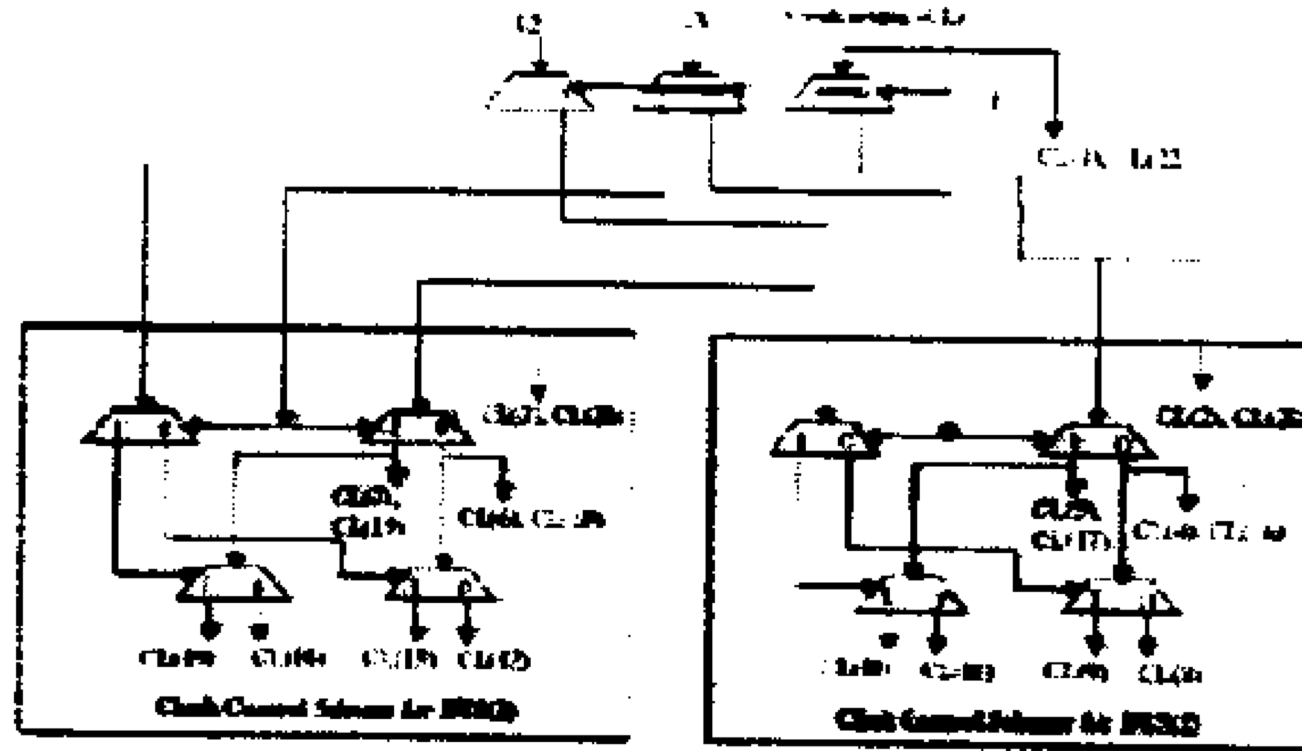


Figure 12: Hierarchical Clock Controller for DTS(3)

0.4.6 Shift Controller

As we know the input of clock controller is $L_1, L_2, .., L_k$ which decides the path in DTS where the clock remains active. Hence the selection of the path during the loading and unloading of the data in DTS depends on $L_1, L_2, .., L_k$. There are two ways the scan paths can be filled:

1. Depth-first load (DFL)
2. Breadth-first load (BFL).

Depth-first load (DFL) In this scheme, a scan-path is loaded serially up to a certain depth in the tree once for all, and then the other paths are processed. The shift controller can be implemented as a simple finite-state machine that drives the control lines internally during scan shift. Hence, no external I/O pins are needed for these control lines.

Let (p10 p9 p8 p7 p6 p5 p4 p3 p2 p1) denote a 10-bit vector to be shifted in the DTS. Assume that the current contents of the Flip-flops are Q1, Q2, ., Q10, where Qi denotes the content of the ith FF. Then scanin and scanout sequence for the DTS(2) is as follows.

Table 2: Shift Controller for DFL Mode

shift clock	L_1	L_2	active path	scanned-in bit	scanned-out bit
1	0	0	Path-0	p1	Q10
2	0	0	Path-0	p2	Q8
3	0	0	Path-0	p3	Q4
4	0	1	Path-1	p4	Q2
5	0	1	Path-1	p5	Q1
6	1	0	Path-2	p6	Q5
7	1	0	Path-2	p7	Q9
8	1	1	Path-3	p8	Q6
9	1	1	Path-3	p9	Q3
10	1	1	Path-3	p10	Q7

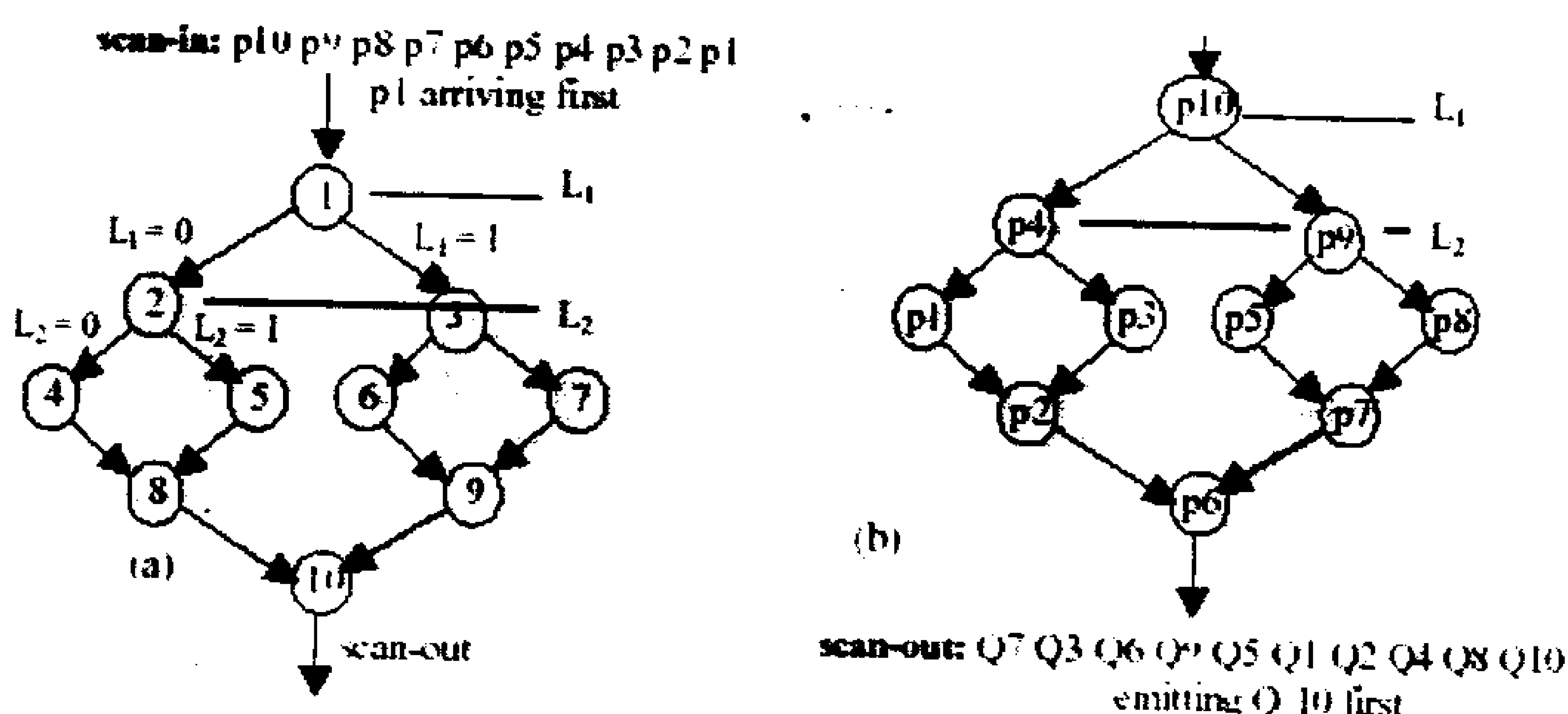


Figure : Scan-in and scan-out sequence for DTS(2) architecture in DFL mode

Breadth-first load (BFL) : In DFL the shifted outputs do not preserve the order in which the inputs are loaded. Although it is not necessary criterion for scan design. But one can achieve this preservation by applying breadth first loading. In this scheme . a modulo-2_k is counter continuously to load and unload the flip-flops, where the counter value is used to choose the corresponding path in the tree.For instance in DTS(2) with 10 flip-flops the counter is run for 10 shifting clock as 0, 1, 2. 3. 0. 1, 2. 3. 0. 1 .To unload from the tree (and to load the next 10- bit vector concurrently). we continue the count from the last value, i.e.. run the counter for 10 cycles again as follows: 2, 3, 0, 1, 2, 3, 0, 1, 2, 3. It can be verified that the input and output order remain the same in this scheme.

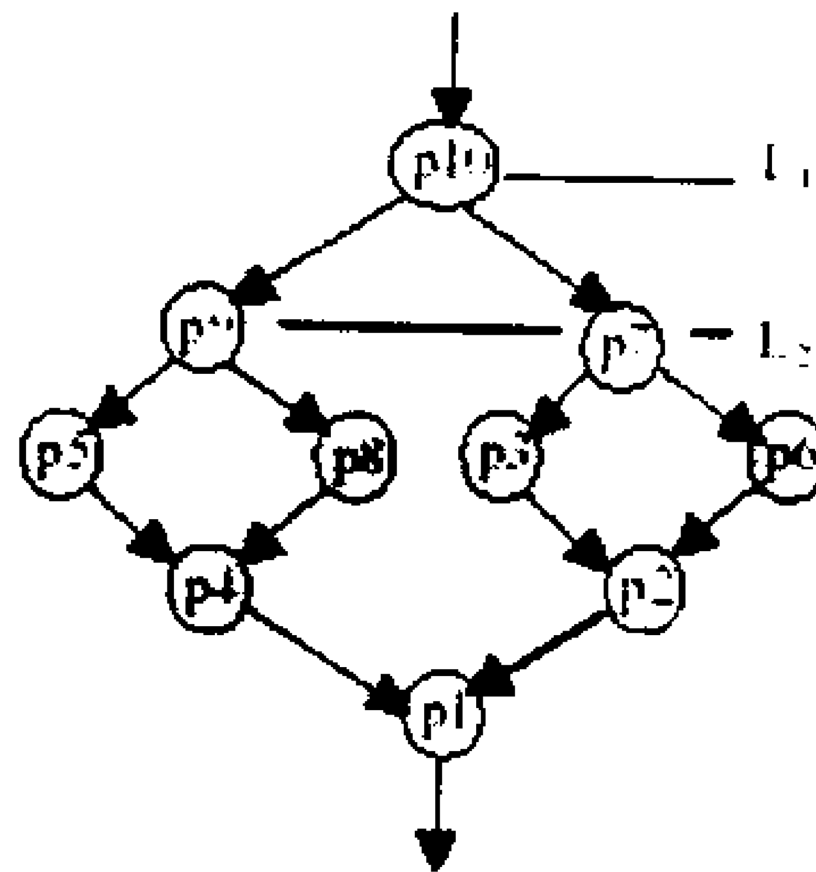


Fig. : Loading unloading in BFL mode:
scan-out: Q1 Q2 Q5 Q3 Q7 Q4 Q8 Q6 Q9 Q10:
emitting Q10 first

0.4.7 Power dissipation in DTS

As we have seen from the power dissipation model the shift power in the circuit mainly depends on In a classical linear scan chain of length f, the total worst case switching activity during scan shifting is of the order of $O(T.f^2)$,where T is the number of test vectors. In the multiple scan chain If s is the number of linear chains, the worst case complexity of switching activity is equal to $O(T.f^2/s)$, since in the multiple scan paths there is restriction of choosing number of scan paths as it increases the number of pins, hence the improvement in the worst case complexity of switching activity in multiple scan as compare to single chain serial scan is by a constant factor. In case of double tree scan architecture the switching activity worst case complexity is of the order of $O(T.f.(\ln f))$. This is a drastic improvement in the worst case complexity of switching activity.

0.4.8 Implementation of Hierarchical Clock Controller

In the implementation of hierarchical clock controller the data Structure that is used is 3-way Linked list as shown in figure. Each node consist of three child left, right and next node. Each node consist of all the information that is needed to specify the $(2 \rightarrow 1)$DMUX unit. The algorithm which is used for the implementation of the hierarchical clock controller for DTS(k) is as follows

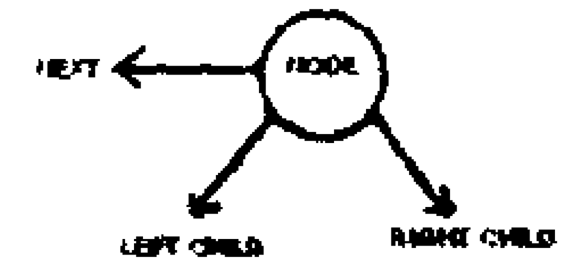


Figure 13: Node used in the Algorithm for hierarchical clock controller

ALGORITHM : Hierarchical Clock Controller

INPUT : Level k of DTS(k)
OUTPUT : Graph G(V,E) pointed by header1

```
procedure HCC()
begin
 1: node *header1, *header2 ;
 2: integer j=2;
 3: header1 = Create_Node() ;
 4: while j ≤ k do
 5:    header2=Copy_Graph(header1) ;
 6:    header1=Set_Connection(header1, header2);
 7:    j++;
 8: end while
 9: Set_Name(header1);
end
```

There are four functions used in the above algorithm create_node() allocates the space for the node and initialize the node. Copy_Graph() basically performs the deep copy the graph pointed by header1 . It is implemented by using the breadth first search Algorithms on the header1 when the node makes its first visit (i.e. when the node turns its color from white to grey) in bfs algorithm the create_node() is called with connection as in the header1 graph. The third function Set_connection() function allocates the j number of nodes at each level in the algorithm. the children of nodes are suitably connected by header1 and header2.How the connections are made is claer from the hierarchical clock controller digram shown in figure ....After making the connections between two graph pointed by header1 and header2 it returns the header1 which points the graph created after connections. finally. Set_Name() function is performed for the naming of each node according to DMUX specification. This function also implements the breadth first search algorithm for traversing each node.

0.5 Experiments and Results

Due to the rapid advances in technology and the progress in the development of design methodologies and tools, the fabrication of more and more complex electronic systems has been made possible in recent years. In order to manage complexity, design activities are moving toward higher levels of abstraction and the design process is decomposed into a series of subtasks , which deal with different issues. The Hardware description language that has been used for design flow is IEEE standard VHDL which allows multilevel description and provides support for both a behavioral and a structural view of hardware models with their mixture in description being possible.

Experiments were carried out on the ISCAS89 benchmark circuits for comparing the single chain serial scan and double tree scan architecture.Experiments were carried out using the flow diagram and tools shown in figure.First the VHDL code is generated by using the VHDL Codegen Tool. We designed and developed this tool for different scan architectures. Once the VHDL code is generated the wave form is verified by using the test vectors generated randomly and keeping the fault coverage as approx 90% for most of the circuits. The circuit is simulated and output is verified from ModelSim. The VHDL code for the circuit is given input to the Leonordo Spectrum tool which generates the RTL schematic diagram and convert the VHDL code into verilog code. Finally the verilog code is used in Mentor IC Station which generates the final layout of the circuit.

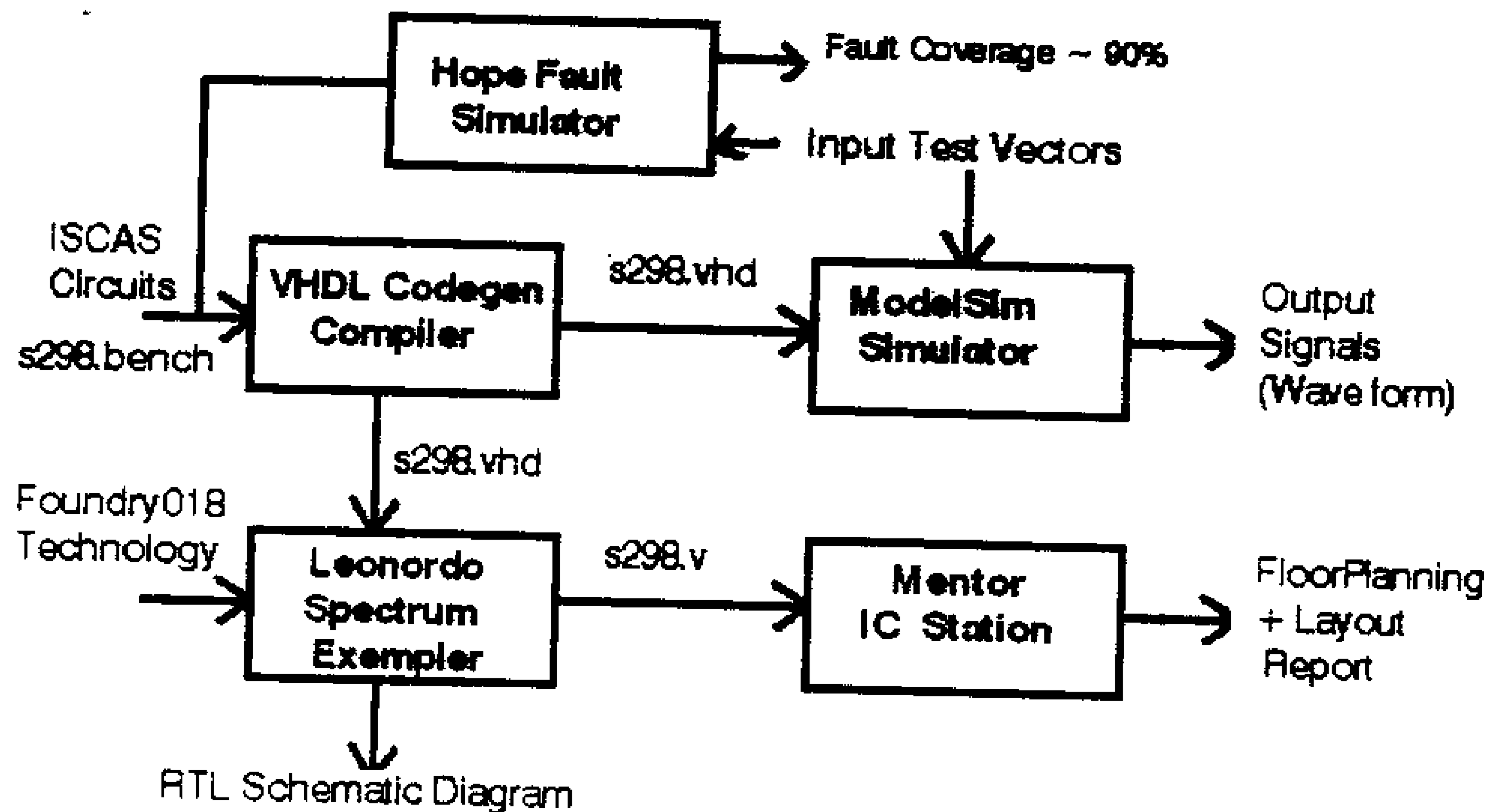


Figure 14: ISCAS'89 Circuit Experimental Flow

The following table illustrates the comparison of internal area (cell −macro − channel) of various benchmark circuits for single chain serial architecture and double tree scan architecture . this result is obtained by using the Mentor IC station layout.

Table 3: **Total Internal Area(μ^2)**

Circuit	# FFs	Serial Scan	DTS Arch.	Area Overhead(%)
s27	3	1914	1914	0
s298	14	21995	22920	4.2
s382	21	29665	30181	1.7
s510	6	34304	34621	0.9
s713	19	64540	65194	1.01
s820	5	50493	50831	0.67
s953	29	76504	78471	2.57
s1196	18	97278	98273	1.02
s1494	6	125643	126021	0.3
s5378	179	609177	621292	0.2
s9234.1	211	1340682	1363947	0.19

The following table compares the switching activity of the two architecture.

Table 4: **Worst Case Switching Activity per test Vector**

Circuit	Serial Scan	DTS Arch.	SA Savings(%)
s298.bench	196	112	43
s382.bench	441	231	48
s713.bench	361	228	37
s953.bench	841	377	56
s5378.bench	32041	6802	79
s9234.bench	51984	6840	87
s38584.bench	2108304	95832	95

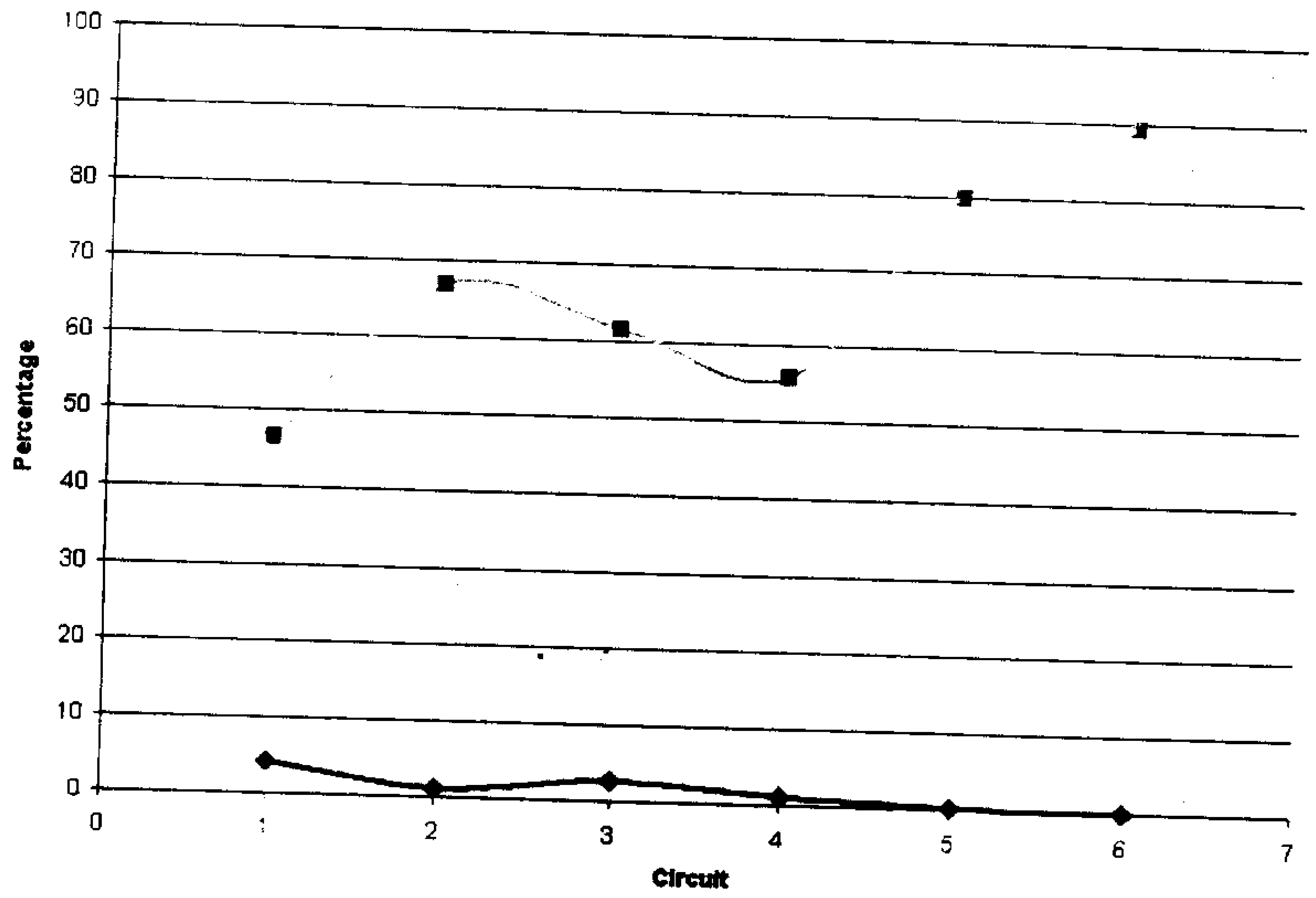


Figure 15: Area Overhead and Switching Activities

Table 5: DTS(k) scan chain parameters

k	# FFs	# No. of scan paths	Scan Path Length
0	1	1	1
1	4	2	3
2	10	4	5
3	22	8	7
4	46	16	9
5	94	32	11
6	190	64	13
7	362	128	15
8	766	256	17
9	1534	512	19
10	3070	1024	21

Table 6: **Serial Concatenation of ISCAS'89 circuits**

Circuit	# FFs	DTS(k) Concatenation	Scan Path length
s298.bench	14	DTS(2)+DTS(1)	8
s382.bench	21	DTS(2)+DTS(2)+DTS(0)	11
s713.bench	19	DTS(2)+DTS(1)+DTS(1)+DTS(0)	12
s838.1.bench	32	DTS(3)+DTS(2)	12
s953.bench	29	DTS(3)+DTS(1)+DTS(0)+DTS(0)+DTS(0)	13
s5378.bench	179	DTS(5)+DTS(4)+DTS(3)+DTS(2) +DTS(1)+DTS(0)+DTS(0)+DTS(0)	38
s9234.bench	228	DTS(6)+DTS(3)+DTS(2)+DTS(1) +DTS(0)+DTS(0)	30
s13207.bench	669	DTS(7)+DTS(6)+DTS(5)+DTS(0) +DTS(0)+DTS(0)	42
s15850.bench	597	DTS(7)+DTS(6)+DTS(3)+DTS(0) DTS(0)+DTS(0)	38
s35932.bench	1728	DTS(9)+DTS(6)+DTS(1)	35
s38417.bench	1636	DTS(9)+DTS(5)+DTS(1)+DTS(1)	36
s38584.bench	1452	DTS(8)+DTS(7)+DTS(6)+DTS(5) DTS(2)+DTS(2)	66
s38584.1.bench	1426	DTS(8)+DTS(7)+DTS(6)+DTS(4) +DTS(3)+DTS(2)+DTS(2)	71

0.5.1 VHDLCodeGen Tool

The VHDLCodeGen tool is designed and developed to generate the VHDL code by incorporating the different scan architectures that we have discussed so far.

The tool is designed to take the input as a standard benchmark circuits ISCAS'89 benchmarks and generate the VHDL code according to the selected architecture. The compiler designed in the tool takes the standard benchmark circuits as an input. the grammar is in the format of out = structure (inp1, inp2, ...). . The structural VHDL code is generated and modifies the scan path according to the selected scan architecture. For instance in the multiple scan architecture the D flip-flop is modified into scan D Flip-flop by adding the extra multiplexors and serially connected in parallel. The VHDLCodegen also generates vhdl code for the clock controller and shift controller required by the double tree scan. architecture.
It implements the following architecture

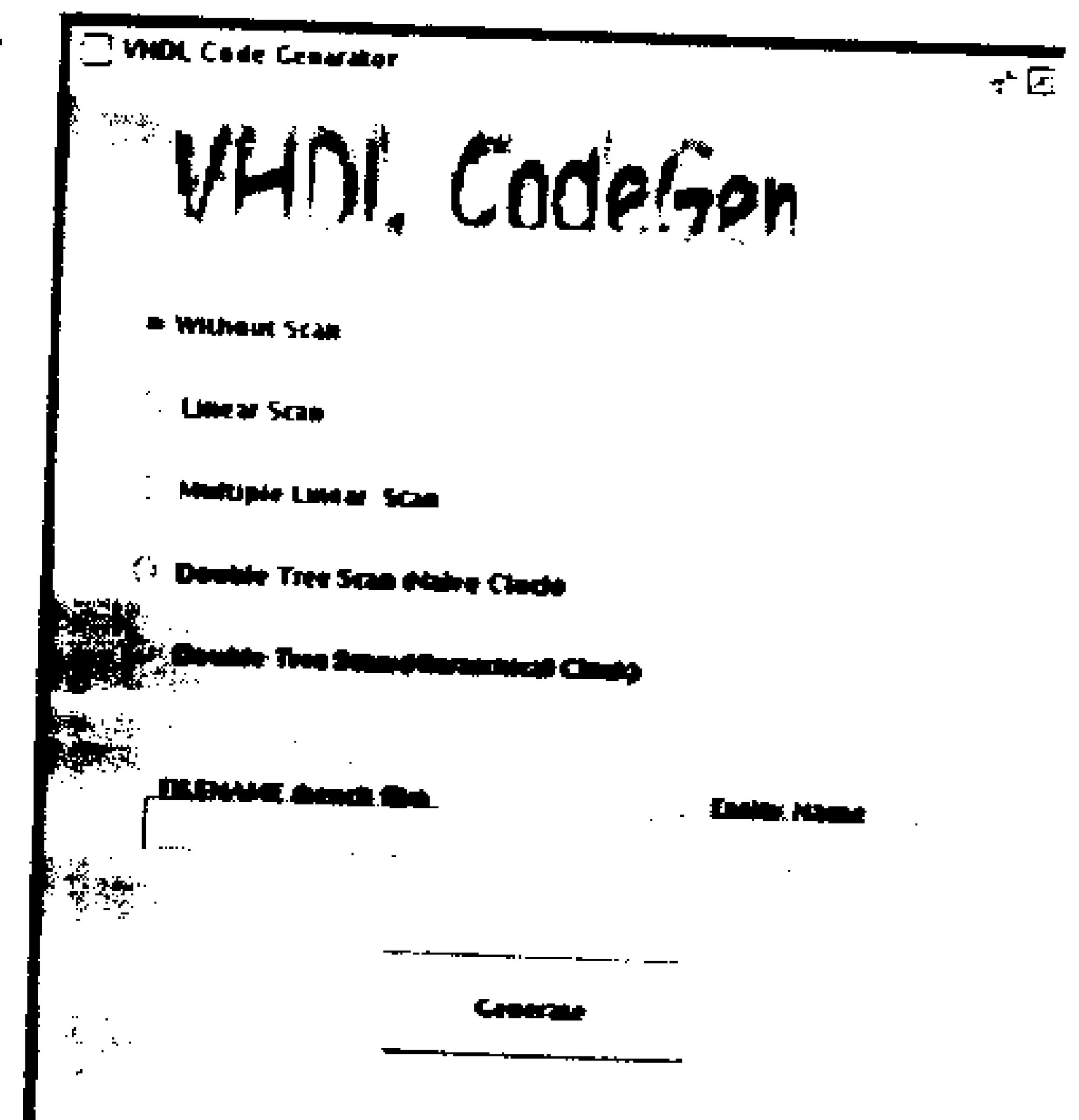


Figure 16: VHDL CodeGen GUI

1. Single Chain Serial Scan Architecture
2. Multiple Chain Serial Scan Architecture
3. Double Tree Scan Architecture with naive clock controller
4. Double Tree Scan Architecture with hierarchical clock controller.

The reports and diagrams enclosed are as follows.

1. RTL Schematic diagram for 298.bench (DTS)
2. Input Output Signal waveform for s27.bench(DTS)
3. Hope Fault Simulation report for s344.bench
4. Floor Planning Report for s298.bench(DTS)
5. Final Layout diagram of s298.bench(DTS)

0.5.2 Floor Planning Report for s298.bench(DTS)

Track spacing:
1 X-track = 660 grids = 1 microns.
1 Y-track = 660 grids = 1 microns.

Gaps between internal zone and external rows (mic):
Left = 0.0, Right = 0.0, Top = 0.0, Bottom = 0.0

Row/dimension constraints:
Minimum No of rows = 0
Minimum x dimension, mic = 15.2
Minimum y dimension, mic = 0.0
Max chip width, mic = 3.0
Max chip height, mic = 3.0

Internal zone parameters:
Total internal (cell – macro + chan) area, mic^2 = 22920.0
Internal (std. cell – chan) area, mic^2 = 22920.0
Total large macro area, mic^2 = 0.0
Number of internal rows = 15
Internal routing/cell area ratio = 81 (auto-computed)
Avg. instance height, mic = 5.8
Avg. channel height, mic = 4.6
Avg. row height rounded to tracks, mic = 10.5
Total connections (pins - nets) = 1709
Internal zone area, mic^2 = 24119.5
Internal zone width, mic = 151.8
Internal zone height, mic = 158.9
Aspect ratio = 0.96
Specified lower aspect ratio = 0.5
Specified upper aspect ratio = 1.5

Overall chip dimensions:
Chip X-dimension, mic: 157.1
Chip Y-dimension, mic: 164.3

0.5.3 Hope Fault Simulation Report for s344.scan

```
*******************************************************
*                                                     *
*          Welcome to HOPE (version 2.0)              *
*                                                     *
*            Dong S. Ha (ha@vt.edu)                   *
*          Web: http://www.ee.vt.edu/ha               *
*  Virginia Polytechnic Institute & State University  *
*                                                     *
*******************************************************

******   SUMMARY OF SIMULATION RESULTS   ******

1. Circuit structure
   Name of circuit                        : 9_inputs
   Number of primary inputs               : 9
   Number of primary outputs              : 11
   Number of flip-flops                   : 15
   Number of gates                        : 160
   Level of the circuit                   : 20

2. Simulator input parameters
   Simulation mode                        : file (inp.tv)

3. Simulation results
   Number of test patterns applied        : 500
   Fault coverage                         : 90.643 %
   Number of collapsed faults             : 342
   Number of detected faults              : 310
   Number of undetected faults            : 32

4. Memory used                            : 134758 Kbytes

5. CPU time
   Initialization                         : 0.017 secs
   Fault simulation                       : 0.200 secs
   Total                                  : 0.217 secs
```

0.5.4 RTL Schematic View of s298 for DTS Architecture

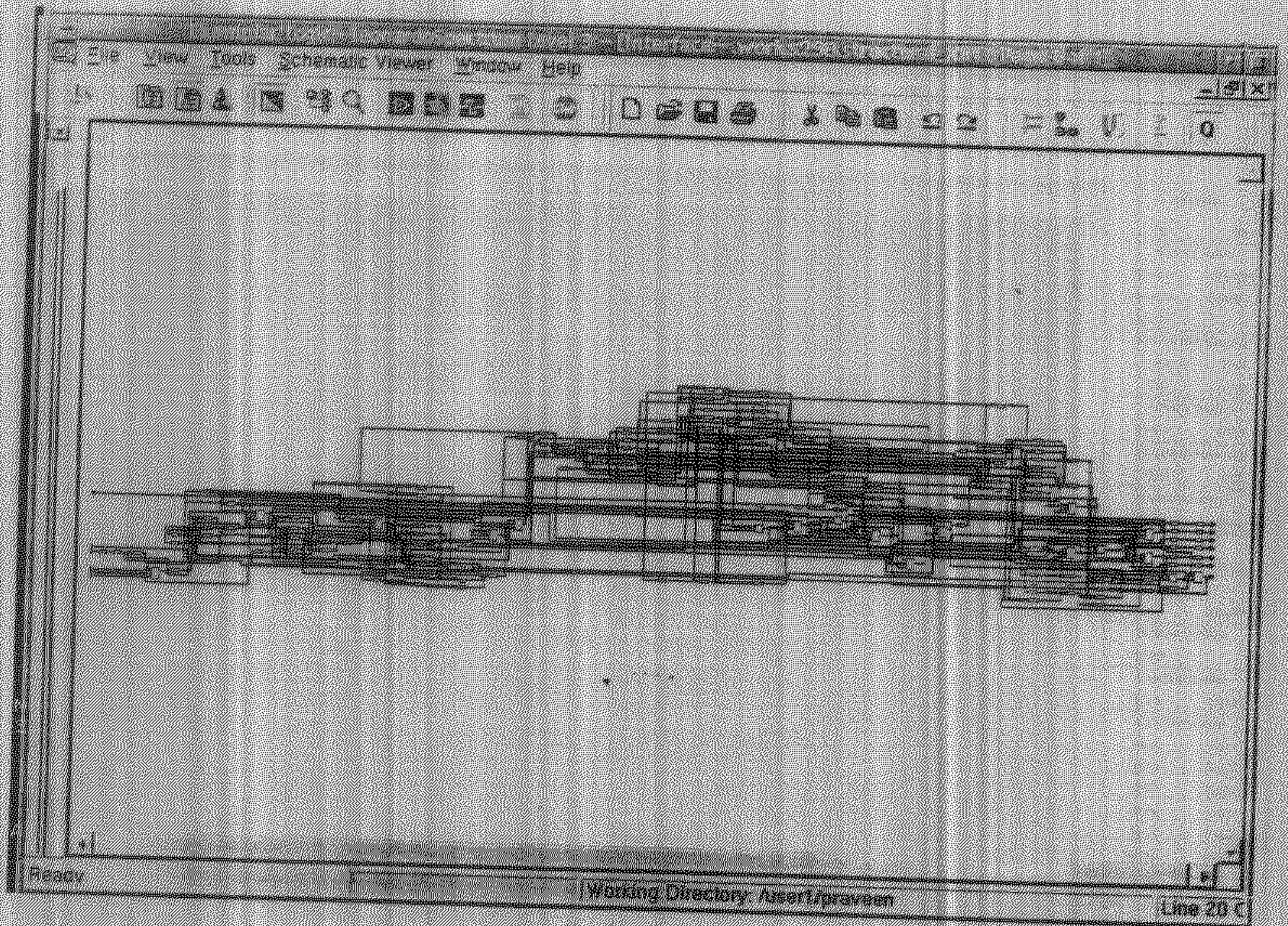


Figure 17: RTL Schematic View of s298 for DTS Architecture

0.5.5 Final Layout of the Circuit s27 for DTS Architecture

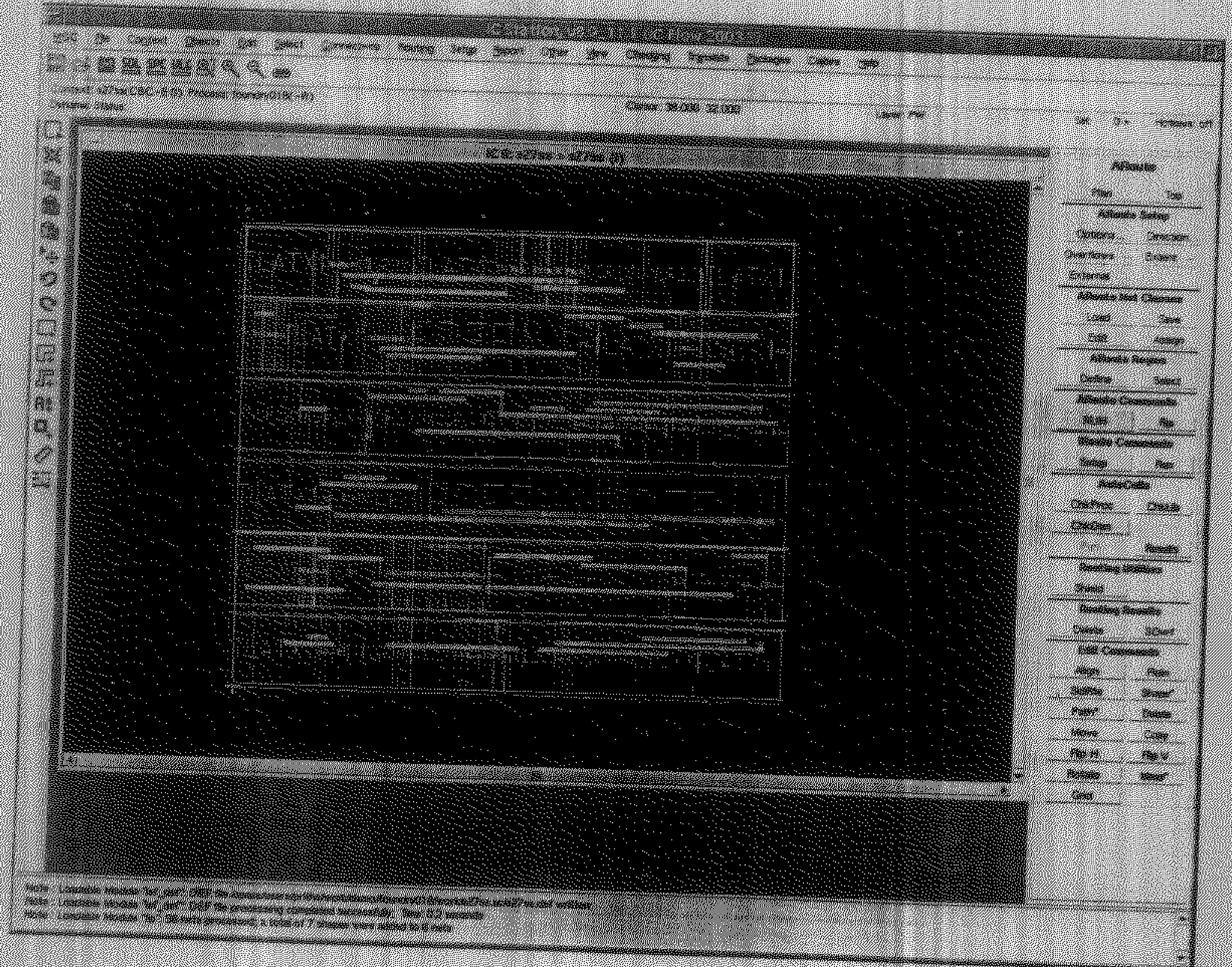


Figure 18: Final Layout of the circuit s27 for DTS Architecture

0.6 Conclusion

In the dissertation we have analysed the power estimation and area for the scan based vlsi testing. There is a need of trade off between power reduction and area overhead in scan design. We have also developed the tool for VHDL code generation for scan based design. There are certain open field in DTS architecture which can be taken as a future work. Mainly area overhead that is caused by clock controller and also clock skewness problem can be analysed in DTS architecture. The Test Application time analysis can also be done as a future work in the DTS architecture.

Bibliography

[Ash] Peter J. Ashenden, *High level vlsi synthesis.*

[BA] Michael L. Bushnell and Vishwani D. Agarwal, *Essential of electronic testing for digital, memory and mixed signal vlsi circuits.*

[BBBZ03] S. C. Seth B. B. Bhattacharya and S. Zhang, *Double-tree scan: A novel low-power scan-path architecture*, Proceedings 2003 International Test Conference (2003), 470–479.

[CA90] Kwang-Ting Chen and Vishwani D. Agrawal, *A partial scan method for sequential circuits with feedback*, Transactions on Computers **Vol 39 No. 3** (1990), 544–548.

[CW91] R. Camposano and Wayne Wolf, *High level vlsi synthesis*, Kluwer Academic Publishers, 1991.

[DGZ00] A. Paschalis M. Psarakis D. Gizopoulos, N. Kranitis and Y. Zorian, *Low power/energy bist scheme for datapaths*, TProc. VTS (2000), 23–28.

[FBK89] D. Bryan F. Brglez and K. Kozminski, *Combinational pro-files of sequential benchmark circuits*, ISCAS vol. 14 n. 2 (May 1989), 19291934.

[F.N95] F.N.Najm, *A survey of power estimation techniques in vlsi circuits*, IEEE Transactions VLSI Systems, vol. 2 no. 4 (January 1995), 446–455.

[IEE87] 1076 Standard IEEE, *Ieee standard vhdl language reference manual*, IEEE Press, 1987.

[MA90] A. D. Friedman M. Abramovici, M. A. Breuer, *Digital systems testing and testable design*, IEEE Press, 1990.

[McC] E. J. McCluskey, *Logic design principles.*

[PC91] Scott R. Powell and Paul M. Chau, *A model for estimating power dissipation in a class of dsp vlsi chips*, IEEE Transactions on Circuits and Systems Vol. 38 No. 6 (June 1991).

[THCR] C. E. Leiserson T. H. Cormen and R. L. Rivest, *Introduction to algorithms.*

[WP83] T. W. Williams and K. P. Parker, *Design for testability–a survey*, Proc. IEEE 31 No. 1 (January 1983), 18–22.