



Dublin City University
Ollscoil Chathair Bhaile Átha Cliath

Power Constrained Test Scheduling in System-On-Chip Design

Liang Gao, B. Eng.

Submitted in fulfillment of the requirement for the
degree of the M.Eng in Electronic Engineering

Supervisor: Dr. Xiaojun Wang

**School of Electronic Engineering
Dublin City University**

September 2004

Approval

Name: Liang Gao
Degree: M. Eng.
Title of thesis: Power Constrained Test Scheduling in System-On-Chip Design
Examine Committee: to be appointed, Dublin City University Chair

Prof. Steve Grainger, Staffordshire
University, External Examiner

Dr. Xiaojun Wang, Lecturer, Dublin City
University, Internal Supervisor

Dr. Sean Marlow, Senior Lecturer, Dublin City
University, Internal Examiner

Date Approved: _____

Declaration

I hereby certify that this material, which I now submit for assessment on the program of study leading to the award of M.Eng. is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: Liang Gao (Candidate)

ID No.: 50161539

Date: September 22, 2004

To my wife Yan

Acknowledgements

My best acknowledgments go to

Dr. Xiaojun Wang for supervising my work and for valuable guidelines, and hints with much freedom, for many interesting discussions, patient correction, sharp criticism but also encouragement.

Valentin Muresan, Gabriel-Miro Muntean, Duro Todinca, Rosemary Kingston, Yongle Liu, and Gerald Considine for their friendly assistance on numerous occasions.

All the colleagues in RINCE, DCU for providing a very friendly atmosphere to work with.

Finally I would appreciate God, who let me come to know that patience will finally bear fruit.

Abstract

With the development of VLSI technologies, especially with the coming of deep sub-micron semiconductor process technologies, power dissipation becomes a critical factor that cannot be ignored either in normal operation or in test mode of digital systems. Test scheduling has to take into consideration of both test concurrency and power dissipation constraints. For satisfying high fault coverage goals with minimum test application time under certain power dissipation constraints, the testing of all components on the system should be performed in parallel as much as possible.

The main objective of this thesis is to address the test-scheduling problem faced by SOC designers at system level. Through the analysis of several existing scheduling approaches, we enlarge the basis that current approaches based on to minimize test application time and propose an efficient and integrated technique for the test scheduling of SOC's under power-constraint. The proposed merging approach is based on a tree growing technique and can be used to overlay the block-test sessions in order to reduce further test application time. A number of experiments, based on academic benchmarks and industrial designs, have been carried out to demonstrate the usefulness and efficiency of the proposed approaches.

List of Acronyms

ALU.....	Arithmetical Logic Unit
APD.....	Average Power Dissipation
ASIC.....	Application Specific Integrated Circuit
AT.....	Automatic Test
ATE.....	Automatic Test Equipment
ATPG.....	Automatic Test Pattern Generation
BILBO.....	Built-In Logic-Block Observer
BIST.....	Built-In Self-Test
BS.....	Boundary Scan
BSR.....	Boundary Scan Ring
BT.....	Block Test
BTS.....	Block Test Scheduling
BUT.....	Block Under Test
CBILBO.....	Concurrent Built-In Logic-Block Observer
CBSOC.....	Core-Based System-on-Chip
CMOS.....	Complementary Metal-Oxide Semiconductor
CPU.....	Central Processing Unit
CPUT.....	Central Processing Unit Time
CTD.....	Compressed Test Data
CTL.....	Core Test Language
CTS.....	Concurrent Test Set

CTT.....Core Testing Techniques
 CUT.....Circuit Under Test
 DFT.....Design for Testability
 DRAM.....Dynamic Random Access Memory
 DS.....Digital System
 DV.....Digital Verification
 DSP.....Digital Signal Processor
 ECT.....Extended Compatibility Tree
 EMC.....Extended Main Clique
 ETB.....Expandable Tree Branch
 FPGA.....Field Programmable Gate Array
 GA.....Genetic Algorithm
 HLLS.....High Level Low-Power Synthesis
 HLS.....High-Level Synthesis
 HLTS.....High-Level Test Synthesis
 IC.....Integrated Circuits
 IP.....Intellectual Properties
 ILP.....Integer Linear Programming
 I/O.....Input/Output
 JTAG.....Joint Test Action Group
 LFSR.....Linear Feedback Shift Register
 LSI.....Large Scale Integration
 MA.....Merging Approach
 MCM.....Multi-chip Module
 MILP.....Mixed Integer Linear Programming
 MISR.....Multiple Input Signature Register
 MPCS.....Maximum Power Compatible Set
 MT.....Macro Test

NP..... Non-deterministic Polynomial
 PCL..... Power Compatible List
 PCS..... Power Compatible Set
 PD..... Power Dissipation
 PTS.....Power-Constrained Block-Test Scheduling
 RISC.....Reduced Instruction Set Computer
 RMS.....Root Mean Square
 RT..... Register Transfer
 RTL..... Register Transfer Level
 SA..... Simulated Annealing
 SOC.....System on Chip
 TAT.....Test Application Time
 TCG.....Test Compatibility Graph
 TDM.....Testable Design Methodology
 TIG.....Test Incompatibility Graph
 TL.....Test Length
 TP.....Test Parallelism
 TPG..... Test Pattern Generation
 TS.....Tabu Search
 TSP.....Traveling Sales Person
 TTR..... Test Time Reduction
 TTT..... Total Test Time
 VLSI.....Very Large Scale Integration

List of Tables

Table 1.1 Level of abstraction in information processing by a digital system.....	3
Table 5.1 Experimental results on Muresan's Design One.....	69
Table 5.2 Experimental results on Design by Kime.....	71
Table 5.3 A comparison of different test schedule approaches on ASIC Z.....	73
Table 5.4 Experimental results on System L.....	75
Table 5.5 Experimental results on ASIC Z Design Two.....	78
Table 5.6 Experimental results on Two Muresan's Design Two	80

List of Figures

Figure 1.1 An Example of SOC.....	2
Figure 1.2 A typical High-Level Synthesis (HLS) System.....	6
Figure 3.1 First Example of Node Under Test.....	20
Figure 3.2 A General Hierarchical Test Structure.....	23
Figure 3.3 Power Dissipation as a Function of Time.....	35
Figure 3.4 Merging Step Example.....	38
Figure 3.5 Test Scheduling Chart and ECT Example.....	40
Figure 3.6 An Example of PTS Chart of 20 Tests with Power Dissipation Constraint (PTC).....	41
Figure 4.1 An Example of Acceptable Overlap of Tests by Merging Approach.....	44
Figure 4.2 The Comparison of the Merging Approach with Tree Growing Technique (Extended Compatibility Tree).....	45
Figure 4.3 EMC Test Length and the First Level Gap Length.....	46
Figure 4.4 Equivalent Test Schedule in Merging Approach.....	46
Figure 4.5 Power-Test Scheduling Result by ECT Approach.....	55
Figure 4.6 Test Compatibility List of 10 Tests.....	56
Figure 4.7 Tree Growing Steps Example.....	58

Figure 4.8 Merging Result Followed by EMC	
Approach to 10 Tests.....	60
Figure 4.9 Merging Steps of the Example.....	62
Figure 5.1 Test Schedule Produced by	
Muresan's Approach on Design One.....	67
Figure 5.2 Test Schedule Produced by Erik's Simulated	
Annealing Implementation on Muresan's Design One.....	67
Figure 5.3 Test Schedule using Erik's Approach with Initial	
Sorting based on (a) Power, (b) Time and	
(c) Power x Time on Muresan's Design One	68
Figure 5.4 Test Schedule using Merging Approach on	
Muresan's Design One	69
Figure 5.5 Test Schedule using Merging Approach on Design	
Kime (no power dissipation constraint).....	70
Figure 5.6 Test Schedules Generated using Zorian's Approach.....	71
Figure 5.7 Test Schedules Generated using Chou's Approach.....	72
Figure 5.8 Test Schedules Generated using Erik's Approach.....	72
Figure 5.9 Test Schedules Achieved using Merging Approach.....	73
Figure 5.10 Designer's Test Schedule on System L.....	74
Figure 5.11 Test Schedules Achieved using Erik's Heuristic	
with Sorting Based on Power on System L.....	74
Figure 5.12 Test Schedules Achieved using Our	
Heuristic on System L.....	75
Figure 5.13 Test Schedules Achieved using Erik's Heuristic	
on ASIC Z Design Two using Initial Sorting Based on	
Based on (a) Power, (b) Time and (c) Power x time.....	77

Figure 5.14 Test Schedules Achieved using Merging Approach	
on ASIC Z Design Two.....	78
Figure 5.15 Test Schedule Produced by Muresan <i>et al.</i>	
on Muresan's Design Two	79
Figure 5.16 Test Schedule using Merging Approach	
on Muresan's Design Two	79

Contents

Approval.....	II
Declaration.....	III
Acknowledgments.....	V
Abstract.....	VI
List of Acronyms.....	VII
List of Tables.....	X
List of Figures.....	XI
Contents.....	XIV
1 Introduction.....	1
1.1 Thesis Scope.....	1
1.1.1 What is SOC.....	1
1.1.2 Digital System Testing.....	3
1.1.3 Testable Design.....	4
1.1.4 Low-Power Design for Test.....	5
1.1.5 High-Level Synthesis.....	5
1.1.6 Test Parallelism and Power Constraints.	7
1.1.7 Power-Constrained Test Scheduling	7
1.2 Thesis Overview.....	8
2 Background.....	9

2.1	Introduction.....	9
2.2	Design for Testability (DFT).....	9
2.3	High-Level Test Synthesis.....	11
2.4	High-Level Low-Power Synthesis.....	13
2.5	Test Parallelism and Test Time Reduction.....	15
2.6	Test Scheduling Heuristics.....	16
2.6.1	Simulated Annealing.	16
2.6.2	Tabu Search.....	17
3	Power-Constrained Test Scheduling Problem.....	19
3.1	System Testing.....	21
3.1.1	Core Testing.....	22
3.1.2	Core Testing Scheduling.....	24
3.2	Power-Conscious Test Parallelism.....	26
3.2.1	Power Minimization During Test Application.....	27
3.2.2	Power-Constraint Test Scheduling.....	31
3.3	PTS Problem Modeling.....	34
3.3.1	System Modeling.....	34
3.3.2	High-Level Power Dissipation Estimation.....	34
3.3.3	PTS Problem Formulation.....	37
3.3.4	The Tree Growing Technique.....	37
3.3.5	Power-Test Scheduling Chart.....	40
4	Merging Approach Based On Tree Growing Technique.....	42
4.1	Introduction.....	42
4.2	Existing Test Scheduling Techniques.....	42
4.3	Merging Approach Based On Tree Growing Technique.....	44
4.3.1	Operating Procedures.....	45

4.3.2	Algorithm Pseudocode.....	48
4.3.3	Algorithm Complexity	54
4.3.4	Test Scheduling Example.....	54
4.4	Conclusions.	65
5	Experimental Results.....	66
5.1	Experiments on Muresan's Design One... ..	67
5.2	Experiments on Design by Kime.....	70
5.3	Experiments on ASIC Z Design One.....	71
5.4	Experiments on System L.....	74
5.5	Experiments on ASIC Z Design Two.....	76
5.6	Experiments on Muresan's Design Two	79
6	Conclusions and Future Work.....	81
6.1	Thesis Summary.....	81
6.2	Contributions.....	83
6.3	Future Work.....	83
	Bibliography.....	84
	Appendix A: Benchmark Examples.....	A
A.1	Muresan's Design One	A
A.2	Design by Kime.....	B
A.3	ASIC Z Design One.....	C
A.4	System L.....	D
A5	ASIC Z Design Two.....	E
A6	Muresan's Design Two.	F
	Appendix B: Publication.....	H

Chapter 1

Introduction

1.1 Thesis Scope

1.1.1 What is SOC

The rapid development of semi-conductor technology, especially the deep sub-micron process technology, has lead to the implementation of system-on chip (SOC). Manufacturers are integrating increasing numbers of components on one chip. Usually, as a complete system, a SOC includes multiple types of circuitry, such as several Application Specific Integrated Circuits (ASIC), memories, microprocessor, and intellectual property (IP) blocks. Typically, SOC's are designed using embedded reusable cores. An example of SOC is shown in Figure 1.1.

Embedded reusable cores make it easier to import technology to a new digital system and differentiate the corresponding product by leveraging intellectual property advantages. Furthermore, the use of embedded cores shortens the time-to-market for new digital systems through design reuse.

1.1.2 Digital System Testing

Testing of a system is a process in which the Digital System (DS) is run and its resulting response is analyzed to ascertain whether it behaved correctly. The complexity of a circuit relates to the level of abstraction. The level of abstraction relating to information processing on a digital system, can be briefly characterized as that shown in Table 1.1

Control	Data	Level of abstraction
Logic value (or sequence of logic values)		Logic level
Logic value	Words	Register level
Ins	Words	Instruction set level
Programs	Data structures	Processor level
Messages		System level

Table 1.1: Level of abstraction in information processing on a digital system.

A complex circuit is usually regarded as a System because it becomes unmanageable or meaningless for us to consider circuit operations only in terms of processing logic values. A system usually comprises two sections-Data and Control.

Testing involves the activity, which aims at ascertaining design errors and physical faults. Sometimes, the testing of design errors is called Design Verification. Several examples of design errors are, incomplete or inconsistent specifications, incorrect mapping between different levels of design, and conflicts of design rules. Physical faults comprise fabrication errors,

fabrication defects and physical failures [Abr94]. A number of examples of physical faults are:

- Wrong components
- Incorrect wiring
- Shorts caused by improper soldering

1.1.3 Testable Design

With the increasing complexity of digital systems, testing becomes more and more important. Test Processing affects the viability of the current semiconductor industry significantly. The cost of testing a digital system has become a major component in the cost of designing, manufacturing and maintaining. The cost of testing reflects many factors such as testing time, Automatic Test Equipment (ATE), etc.

Digital System design must take testability into consideration in addition to its functionality. In the other words, Digital System design must be testable. This is a fundamental concern to the successful implementation of a digital system design.

Examples that embody testable design criterion are the consideration of inserting test points, Built In Self Test circuitry (BIST), and scan chains. Another one is partitioning of large combinational circuits.

1.1.4 Low-Power Design for Test

The principle of lower power seems to be the permanent objective of digital system design. Currently, it is the miniaturization of some communication and information processing products such as mobile phone

and portable computing products that prioritize the relevant research of low power design methodologies. The target of this research is to enhance the lifetime of mobile phone batteries. However, from a test point of view, low power techniques are more significant.

Digital systems designed with Design For Testability (DFT) considerations can operate in Normal Mode and Test Mode. As is commonly known, digital system devices running in Test Mode can consume more power than when running in Normal Mode [Zor93], it can cause excessive heat, and could destroy the device.

1.1.5 High-Level Synthesis

Synthesis is usually defined as the translation of a behavioral representation of a design into a structural one [Ele98]. The whole synthesis process comprises several consecutive steps performed at different abstraction levels. Various basic implementation primitives are used and different synthesis methods are employed in the different steps. Usually Synthesis relates to system level, high level, logic level and physical design. In this thesis, the Synthesis refers to high-level synthesis only.

High-Level Synthesis (HLS) accepts a behavioral specification of a digital system and a set of objectives as inputs, an RT-level implementation is produced by HLS. A general high-level synthesis system comprises of five main steps: Compilation, High-level transformations, Scheduling, Allocation, and Binding. A typical HLS system is shown in Figure 1.2.

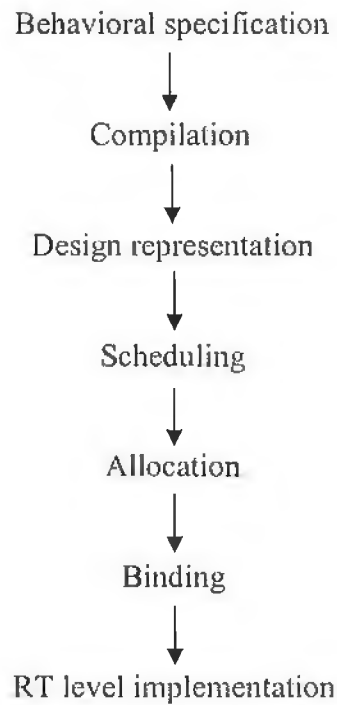


Figure 1.2: A typical High-Level Synthesis (HLS) System.

While the first objective of High-Level Test synthesis is to improve the testability of a design, other constraints, such as performance and area, are also satisfied.

1.1.6 Test Parallelism and Power Constraints

Partitioning has been used widely. Partitioning is significantly important to reduce the number of actual combinational circuits for testing. The notion of Block-Tests (BT) has been brought into Digital System testing. The next problem is to determine the compatibility among the blocks. To reduce test application time, tests of blocks must be performed concurrently as far as possible. However, power constraints have to be considered to avoid chip

overheating and possible damage. Power constraints must be observed during test scheduling. Test Scheduling will be focused on in greater depth in this thesis.

1.1.7 Power-Constrained Test Scheduling

In general, Operation Scheduling deals with the assignment of each operation to a time slot corresponding to a clock cycle or time interval.

Obviously, the main task for Test Scheduling is to minimize the Test Application Time (TAT), by ordering tests in an efficient schedule. Test Scheduling affects the concurrency of testing, therefore determines the parallelism of testing. The maximum number of simultaneous tests should be scheduled under power-dissipation constraint so as to benefit the minimization of TAT.

It is well known that TAT and power dissipation are two increasingly important issues. TAT affects the cost directly.

Since the problem of test scheduling is viewed as NP-Complete problem, the application of Heuristics is essential in Test Scheduling, which is the focus of this thesis.

1.2 Thesis Overview

The overview of the thesis is briefly introduced here. The background of the topics dealt with in this thesis is included in the Second Chapter.

A Power-Constrained Block-Test Scheduling (PTS) problem will be formulated and modeled in the Third Chapter. Here the Test Scheduling issue will be discussed in detail. Relevant definitions and explanations of the terms used in this thesis will be described in this chapter.

A proposed novel Test Scheduling approach, named Merging Approach (MA) based on a tree growing technique, will be introduced in detail in Fourth Chapter, including the algorithm, the complexity analysis and an example of Test Scheduling using the proposed algorithm.

The experimental results will be presented in Chapter Five, including the comparison of experimental results of different Test Scheduling approaches.

The analysis of the advantages and disadvantages of the proposed approach in this thesis will be given in Chapter Six.

Chapter 2

Background

2.1 Introduction

In this chapter, a brief introduction to Block Test Scheduling (BTS) will be given. The following sections will cover the use of Design For Testability (DFT), High-Level Test Synthesis (HLTS), High-Level Low-Power Synthesis (HLLP), Test Parallelism (TP) and Test Time Reduction (TTR). The last section will cover several Heuristics approaches.

2.2 Design for Testability (DFT)

With the increasing complexity of digital systems, traditional methods of electronic device testing are becoming insufficient. The problem of testing digital systems has become more of a challenge.

Undoubtedly, economic considerations are at the heart of all testing problems [Wil94]. Testing application time cannot be too long; otherwise the relevant cost will increase rapidly, which is unacceptable. Because of this, Automatic Testing (AT) is an essential procedure for digital system

verification. This verification idea involves the use of Automatic Test Equipment (ATE).

In our context "Testing" means Digital System Testing (DST). Digital Systems are tested by applying a sequence of signals to their inputs and observing their outputs. If the input and output ports are directly accessible by the Test Equipment, they are called **primary inputs/outputs**. Otherwise, if they are directly inaccessible, they are called **component inputs/outputs**. According to output signals, identification is made to judge that the circuit is correct or not. The practical testing procedure depends on the method used. For example, it is unnecessary to get the observation of output ports for I_{ddq} testing; instead the power supply current is monitored. Built-In Self-Test (BIST) method can be used to compress the outputs and provide only the test outcome in the form of a signature. However, an input pattern, generated based on a given fault model, is required by all methods. (Here, only the stuck-at fault model [Abr90] is considered which captures also many other faults.)

Obviously, the key problem of testing is the derivation of an adequate test set for a particular circuit. This process is known as Test Pattern Generation (TPG). TPG is automated using a computer and is called Automatic TPG (ATPG).

An exhaustive test is not practical for any non-trivial combination circuit, because of the exponential complexity.

Functional testing for blocks containing recognizable functions [Wil94] is used for testing a sequential circuit. A functional test strategy is also employed when testing Medium-Scale Integration (MSI) and Large-Scale Integration (LSI) devices especially where Microprocessors (MP) are concerned.

The testing of digital system is quite complex. To reduce the number of test vectors, partitioning of a digital system is essential, and the treatment of test points is rather flexible.

2.3 High-Level Test Synthesis

As mentioned above the basic aim of High-level Test Synthesis (HLTS) is to improve the testability of a design, while other constraints such as performance and area are satisfied.

HLTS is usually carried out by DFT specific transformations together with traditional High Level Synthesis methods. Testability measures are associated with Test Synthesis as one factor of the cost function to guide the synthesis process. HLTS tries to find a good trade-off between Design Testability, performance, area and Power dissipation. Heuristic algorithms are usually employed to estimate the testability measures, when the final implementation at the gate level is not yet known.

So far, a number of different approaches have been proposed in both the behavioural and structural domains at algorithm or RT levels.

A survey on HLTS is given in [Wag96]. Strategies and challenges of the System-on-Chip test are given in [Agr94, Cho94, Var97, Zor98a, Mar99b]. Information about Structural test-point insertion can be found in [Bat85, Dey94, Gu96, Var98].

Test-Register Minimization (TRM) is another technique derived from either HLS techniques or RTL Transformations [Avr91, Pap91, Avr93, Har93]

The main challenges in relation to System-on- Chip Test are:

- (1) Core Internal Test Challenges
- (2) Core Test Access Challenges

(3) System-chip Test and Diagnosis Challenges.

In relation to Core Level Test, it can be said that a core is typically the hardware description of current standard ICs, e.g., DRAM core, RISC processor, or DSP. A given core is tested as part of the overall System-chip by the System Integrator and not tested individually as in standard ICs. Usually the cores, especially hard cores, are dealt with as a black box, because the system integrator in most cases (except for soft cores) has very limited knowledge of the structural content of the adopted core. So this necessitates that the core provider develops the core test, i.e., the corresponding test pattern and the DFT structures, and delivers the test with the core [Zor98a].

In the traditional approaches, hard cores tests are predetermined for the overall chip test method and the desired fault coverage. So it's the designer that incorporates hard cores test requirements during test development. With a System chip, on the other hand, a core provider is not necessarily familiar with the target application information of components and their quality requirements. Thus, the provided quality level might or might not be adequate. If the fault coverage is too low, the quality level of the system chip is put at risk, and if it is too high, the test cost might become unacceptable.

Core-based System-on-Chip (CBSOC) designs bring us a number of test challenges. Several reusable intellectual property (IP) cores are integrated to form a wide range of functionality on a single die. As IP cores become more complex, the volume of test data for a SOC is growing rapidly. To test these systems effectively, each core must be adequately exercised with a set of precompiled test patterns provided by the core vendor.

Unfortunately, the input-output (I/O) channel's capacity, speed and accuracy, as well as data memory of Automatic Test Equipment (ATE), are

limited. So to apply the enormous volume of test data to the SOC is becoming increasingly difficult.

Reducing test-data volume will not only reduce ATE memory requirements, but also reduce testing time. The testing time depends on the test-data volume, the time required to transfer the data to the cores, the rate at which the test data is transferred (measured by the cores test-data bandwidth and ATE channel capacity), and the maximum scan chain length. Shortening and reorganizing the scan chain can also reduce the total test time. So, test-data compression and decompression techniques offer a promising solution to reduce the enormous test-data volume for SOC's.

A novel approach that uses an embedded processor to aid in deterministic testing of the other components of an SOC is presented in [Jas02]. By this approach, a program containing Compressed Test Data (CTD) can be loaded into the processor on-chip memory by the tester. The proposed approach supports external testing of embedded cores using deterministic test vectors.

A kind of new test-data compression method and decompression architecture is presented in [Cha01b]. Other test-data compression techniques are presented in [Ish98, Jas02].

2.4 High-level Low-Power Synthesis

There are three major sources of power consumption in CMOS circuits- Switching, Short-circuit and Leakage. Usually, if proper design techniques are used, short-circuits and leakage can be made negligible. Thus, switching is the main factor responsible for power consumption. Switching determines the effect of charging and discharging of node capacitance in the circuit and it's given by the following formula [Ped96]:

$$P_{\text{switch}} = \frac{1}{2} C_l V_{\text{dd}}^2 E_{\text{sw}} f_{\text{clk}}$$

Where C_l is the total physical capacitance at the output of the node, V_{dd} is the supply voltage, E_{sw} is the average number of output transistors per clock cycle (also called switching activities), and f_{clk} is the clock-frequency.

According to the above formula, if a lower power consumption design is required, a reduction in supply voltage V_{dd} might be desirable due to its quadratic relation to power. However, the reduction of V_{dd} has a negative impact on the design speed. It is well known that the reduction of V_{dd} increases the delay of the components and thus reduces the throughput of the design, which is a very undesirable effect. Many high-level low-power synthesis techniques have been presented in the literature, a survey on these techniques is given in [Ped96].

High-level power consumption estimation [Meh94, Cha95a] plays an important role in high-level lower-power synthesis. A general approach that uses a parameterized module library together with other heuristics or analytical methods to get the power consumption estimation for a complete design can be found in [Kum95, Mar95].

Power Optimization technique is also used in high-level low-power synthesis. The main goal of a High-Level Power Optimization system is to produce a RT level design, which has minimum power consumption, while achieves the required throughput.

2.5 Test Parallelism and Test Time Reduction

The approaches that reduce the test application time by restructuring the test sequence can be classified into two classes: the Static approaches and the

Dynamic approaches. The main feature of Static Test Sequence restructuring approaches is that they do not increase the complexity of test generation. The test generators are assembled in order so that the overall application time is reduced [Dim 91, Fen91].

Dynamic test sequence restructuring tries to reduce the number of test vectors by carefully assigning the unspecified input signal values to binary constants [Lee92, Pra92].

For a Built-In Self-Test (BIST) circuits, the first test sub session lasts until the sub-circuit with the smallest test length has been tested. The minimized execution techniques, which order the registers included in single scan chain, are proposed in [Gup91, Nar92]. In [Nar93, Nar95], a configuration approach for single scan chain in order to minimize the shift time in applying test patterns on a device is proposed.

For reducing the overall test application time, multiplexers are employed to bypass registers that are not frequently accessed in the process. In [Lai93] a technique that reduces test application time for general scan design circuits, is proposed. The test application time can be reduced to a given scan path by exploiting and eliminating unnecessary scan operations.

In [Lar01c], the authors deal with Scan-chain Sub-division, which is used as a technique to reduce test application time for SOC.

Many attempts have been made to share hardware elements when dealing with optimization of test scheduling. Unfortunately, there is a conflict in the reduction of area overhead and testing time.

A minimal set cover technique [Kim88] is proposed for Built-In Logic-Block Observation (BILBO) minimization. A technique of area optimization, while considering test scheduling using a graph coloring approach, is presented in [Kim82]. In [Cha01], the authors formulated the same problem as an Integer Linear Programming (ILP) as well as a graph search problem with a

heuristic cost function. A High-level Power Optimization survey is contained in [Mac97].

2.6 Test Scheduling Heuristics

Several Heuristics for Test Scheduling will be introduced next. They are Simulated Annealing and Tabu Search respectively.

2.6.1 Simulated Annealing

Simulated annealing is a combinatorial optimization procedure corresponds to the annealing process in physics, where a material is heated up to its melting point and then the temperature slowly lowered to find its minimal energy state.

Its algorithm is similar to the Random Descend Method in that the neighborhood is sampled at random. By allowing uphill moves in a controllable mode, SA provides hill-climbing mechanism to avoid getting stuck in a local optimum.

The Simulated Annealing algorithm is proposed in [Kir83]. After an initial solution is first created, a minor modification of it creates a neighboring solution and the cost of the new solution is evaluated. If the new solution is better than the previous, the new solution is kept. A worse case solution can be accepted at a certain probability, which is controlled by the temperature parameter.

The temperature is decreased gradually during the optimization process and this reduction in temperature leads to the probability of accepting worsening solution decreases. When the temperature value is approximately zeroed, the optimization terminates.

The advantage of the Simulated Annealing algorithm is that it is relatively easy to implement. Among its disadvantages are long computational time and it requires complicated tuning of the relevant annealing parameters [Gaj92]. Regulations and guidelines for parameter selection do not exist, with the rules often depending on the experimental results.

2.6.2 Tabu Search

Tabu Search (TS) [Hal96] is a search approach, which employs an artificial intelligence inspired technique. It can avoid trapping the actual solution at local optima, as a result of this intelligence being kept in memory [Glo86].

As in the case of Simulated Annealing, TS is a high level heuristic procedure used to guide other methods towards an optimal solution.

TS is based on the assumption that an intelligent search should be based on more systematic forms of guidance rather than random selection. It also exploits flexible memory to control the search process. The main mechanism for exploiting flexible memory is to classify a subset of the neighborhood moves as forbidden moves (called Tabu). A short-term memory with a predefined length is used to remember the number of recent moves, which comprises both downhill and uphill moves. These moves are allowed to repeat and are selected intelligently (the best admissible moves are selected).

TS is very useful when the feasibility condition is very strong and the randomly generated neighborhood solutions are usually unfeasible ones. This is partly due to the fact that TS emphasizes complete neighborhood evaluation to identify moves of high quality, while SA samples the neighborhood solutions randomly.

The main drawback of TS is that no theory has yet been formulated to support TS and its convergence behavior. Another obvious difference between TS and SA is that TS uses both short-term and long-term memory intelligently, while SA uses no memory.

Chapter 3

Power-Constrained Test Scheduling

The Digital System / SOC test problems cannot be dealt with at low levels. Test Application Time (TAT), being one of these problems, has to be considered more carefully. Cost considerations affect many designers in their product inventions.

Besides the cost consideration of TAT, the heat dissipated during test application of digital systems also affects the design of test methodologies. It is reported in [Zor93] that one of the major considerations in test scheduling is the fact that heat dissipated during test application is significantly higher than that during a digital system's normal operation. While trying to increase test parallelism in order to reduce test application time, the confining condition of power dissipation constraint (which relates to the overlap of block-tests.) should, of course, also be satisfied.

The so-called Unequal-Length Block-Test Scheduling [Cra88] refers to the scheduling tests for blocks of logic, which has unequal test lengths. It is viewed as part of a system-level block-test approach to be applied on a modular or view of a test hierarchy. The modular elements of this hierarchy include blocks at the following levels: subsystem, back-planes, boards, MCMs, IC dies, macro blocks and RTL blocks. The test hierarchy accepts

RTL blocks as the lowest level blocks, and it is assumed that a test-step level scheduling has been carried out on RTL blocks.

Usually, any node in the hierarchy has different sub-nodes. After the test scheduling optimization has been performed on the node, a few parameters of each test node t_i are determined. These features are given in Figure 3.1, test application time T_i , or Test Length (TL), power dissipation P_i and test resource set $RES.SET_i$.

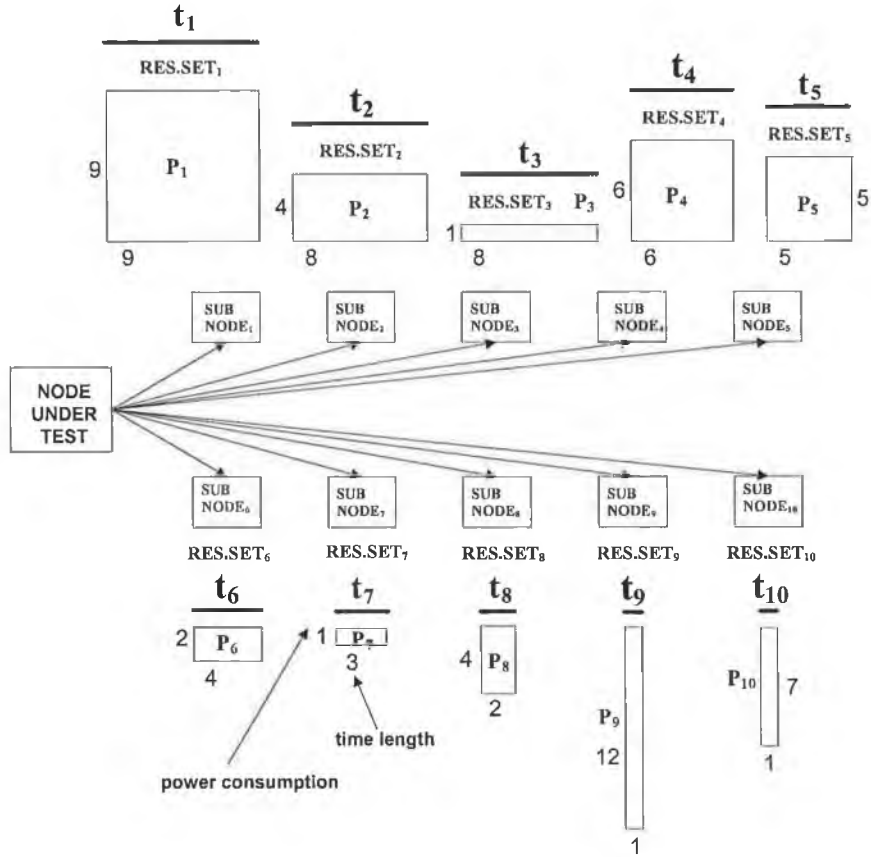


Figure 3.1: First Example of Node under Test.

To a certain node, its sub-nodes are considered for the optimization of parameters, such as test length, power dissipation. This optimization is performed so as to get an optimal or near optimal sequencing or overlaying

(scheduling) of them quickly, while the power dissipation constraints are satisfied. A technique named Merging Approach based on the tree growing technique is proposed here to generate the block-test schedule profile at the node level. It is used to minimize the overall TAT, and to analyze and optimize the characteristics of power dissipation during test.

The first section of this chapter describes the current systems testing issues and the approaches currently employed. The emphasis here is on Core Testing Techniques and Core Related Scheduling approaches.

The second section outlines a brief survey of Power Test Parallelism techniques.

In the last section, system-testing under power constraints is described. Efficient algorithms that can be applied to this power-test model are proposed in the next chapter.

3.1 System Testing

With the steep increase of Digital System design dimensions the tendency is to shift to SOC technology (it has been changing for the classical synthesis methodology at high-level). Because of the cost consideration, design reuse is emphasized widely [And97]. Nowadays, more and more reusable cores are provided to the customers by IP vendors. A core is typically the hardware description of current standard ICs, e.g., Digital Signal Processor (DSP), Reduced Instruction Set Computer (RISC) processor, Dynamic Random Access Memory (DRAM). Such cores may be available in synthesizable RTL (soft) form, gate-level net-list form, or layout-level "hard macro" form [Zor97]. So, a lot of research effort today is concentrated on core synthesis and its testing.

From a digital system-testing viewpoint a core should initially be well characterized. The core internal test (prepared by a core vendor or creator) needs to fit for adequate description, portable and ready for plug and play. This allows for interoperability in the relevant SOC test. An internal test should be described in a standard format, so that compatibility is ensured. The IEEE P1500 [Zor97] proposes such a standard. The SOC test requires adequate test scheduling. The scheduling is required to satisfy a number of system level requirements, such as overall test time, power dissipation, and area overhead. It is necessary to run intra-core, inter-core tests and test scheduling in a certain order so as to avoid impacting the test contents of individual cores or modules.

In the last few years' interest in MCMs has grown rapidly, due to advances in miniaturization techniques. This has contributed to higher performance and reliability in the field of commercial to military electronic products. The production test and field test of large MCM designs will be seriously affected by TAT and Test Mode power dissipation problems unless they have been optimised in testing. The complexity and dimensions of such digital systems (like MCMs) balance the optimisation of TAT with power dissipation constraints.

3.1.1 Core Testing

A general hierarchical test structure is described in Figure 3.2 [Zor97]. Since more and more VLSI chips adopt multiple cores from different vendors, the testing issue and the power dissipation problem become increasingly serious. All core users must face the following two key issues: one is the interconnection of cores within a chip, another is the ability to perform an effective test on the final device.

Test concurrency of core-based system testing is affected by the core supplier's application interface.

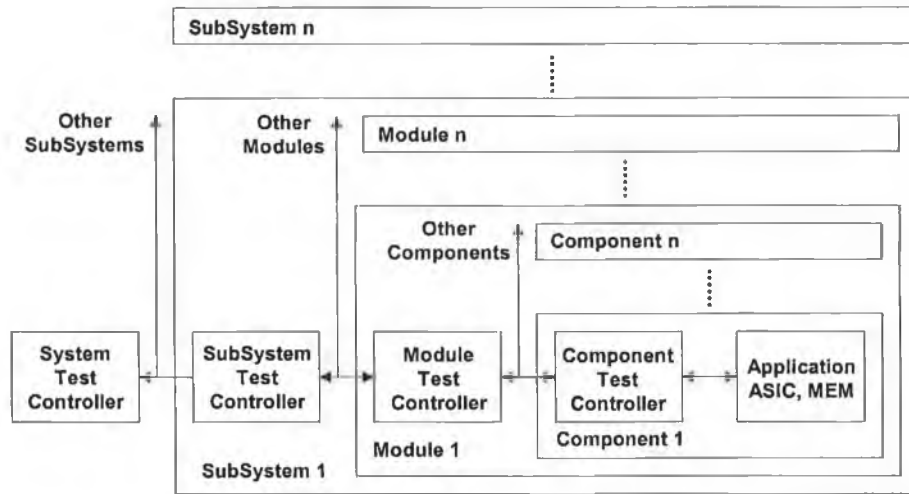


Figure 3.2: A General Hierarchical Test Structure.

A core should, therefore be characterized well from a test perspective, fault coverage and power consumption in test mode or normal mode. Adequate test scheduling is required for the SOC composite test. Test scheduling is performed to satisfy a number of system level requirements, such as Total Test Time (TTT), Power Dissipation (PD), area overhead, etc. Furthermore, test scheduling is also necessary to run intra-core and inter-core tests in a certain order so as to avoid impacting the testing contents of other individual cores or modules.

Many general approaches concerning core testing are introduced in [And97]. Firstly, to finalize core functional tests that run in the complete chip, parallel multiplexed access mechanism from chip pins is provided. The degree of test concurrency drops when there are more I/Os than chip pins or when routing is complex. Secondly, by encapsulating cores in a JTAG (Joint Test

Action Group) scheme, such as Boundary Scan Ring (BSR), to run core tests in parallel is isolated with little need for external support. Thirdly, by using BIST or scan techniques to test each core, internal control and observation are provided.

If there is no method to ensure that multiple cores are tested in parallel, system test time may be unacceptably long. Fortunately, being a test method, one of the features of BIST is autonomy and self-sufficiency. So, BIST is considered ideal for a modular-based system [Zor97].

There are BIST strategies, such as the one referred to in [Zor90], which tries to solve the core test scheme problem, by using a divide and conquer approach to enhance the overall control and observation. There are still pending problems, however, when using this strategy, in isolating and accessing the boundaries of the modules. There are also problems in automating the process of assembling the set of inter-module and intra module set of tests in the overall chip [Ben97].

A Macro Test is an approach used for testing embedded modules as stand-alone units. This approach is very suited for core-based testing and from this point of view it is very suitable for hierarchical and divide and conquer approaches.

3.1.2 Core Test Scheduling

Scheduling can be used to reduce the overall Test Vector Set (TVS) substantially in the various core tests, but the test quality of IC design is not improved. An example of this is a core-based design, with a given set of cores, and given corresponding test protocols and sets of test patterns.

Through test protocol scheduling, the various expanded test protocols can be scheduled, and the total TVS of the system can be minimized [Mar99a]. At

test protocol level, the test scheduling offers a good trade-off between test vectors set reduction and the computational effort to achieve this.

A method is introduced in [Sug98] that selects a test-set for each core from a set of tests provided by the core vendor. Meanwhile, the problem of scheduling their tests in order to minimize the testing time is addressed also. Each test set comprises a subset of patterns for BIST and a subset of patterns for external testing. It is the core vendor that provides multiple test sets for each core, including varying pattern proportions for BIST and external testing for the test sets. The core test-scheduling problem can be formulated as a combinatorial optimization problem and solved using heuristics. Two restrictive assumptions are made in the method. The first one is that every core has its own BIST logic. In other words, the BIST components of the test set for any two cores can be assigned identical starting times. The second assumption is that external testing can be carried out for only one core at a time (i.e. there is only one test access bus at the system level).

An optimal solution approach for the test-scheduling problem for core-based systems is proposed in [Cha01a]. This approach is based on a mixed-integer linear programming model. The drawback is that, when the number of cores in large test-scheduled systems grows, this approach features non-polynomial time. A heuristic-scheduling algorithm, named Shortest-Task-First (STF), is proposed instead to handle such systems. Given a set of test-tasks, a set of test resources and the test access architecture, the test scheduling solution refers to the problem of determining start time for the tasks, so that the total test application time is minimized. Other approaches [Zor98b] deal with the core test problem at system level by focusing the design of efficient test access architectures.

In [Lar00a, b], a greedy heuristic is proposed for core test scheduling under power constraints. The relevant work is developed in [Lar01a, b]. It

considers test scheduling and design of test bus infrastructure at the same time. With this approach, test time, test bus length and width are minimized while power consumption constraints and test resources are considered. There are two steps for this approach. In the first step, a heuristic is used repetitively to select a feasible solution at a low computational cost. The second step optimizes the feasible solution by a simulated annealing approach.

3.2 Power-Conscious Test Parallelism

Power consumption limitation is a critical issue in computing devices, particularly in portable and mobile platforms such as laptop computers and cell phones. Power dissipation during test has not yet been thoroughly researched with much more research to be done. Power consumption during test is important since excessive heat dissipation can damage the circuit under test. Since power consumption in Application Test Mode (ATM) is significantly higher than that during normal operation, special attention must be taken to ensure that the power rating of the SOC is not exceeded during test [Zor93].

A number of techniques to control power consumption in test mode have been presented in the literature. These include the following:

Test-Scheduling Algorithm under power constraint [Abr90], low-power Built-In Self-Test (BIST) [Agr93a, Agr95], and techniques for minimizing power during Scan Testing [Agr93b, Agr94, Ait99]. Power consumption is especially important for SOC's, because test-scheduling techniques for system integration attempt to reduce the test time by applying scan BIST vectors to several cores simultaneously [Ali94, AMS]. Therefore, it is extremely important to control power consumption while testing the IP cores in a SOC.

There are a few Structural Domain approaches that tackle the power dissipation problem during test application at logic level. These include Test Vector Reordering (TVR), Test Vector Inhibiting (TVI), switching activity conscious ATPG and Scan Latch Reordering (SLR). Unfortunately, the above approaches are inefficient at high levels. An efficient solution that partitions the system under test at system-level is proposed in [Zor93], which includes appropriate test planning and scheduling to solve the test-scheduling problem under high-level power constraints. A feasible solution is proposed in the next chapter in this thesis.

The power dissipation problem during test application is described in this section. Then the main techniques, which have been applied to solving the problem, are surveyed. Finally, previous work on Power-Constrained Test Scheduling techniques, the main topic of this thesis, will be focused on.

3.2.1 Power Minimization During Test Application

Performance, cost, and testability are the main parameters targeted during the Synthesis and Optimization phase of integrated circuits. The following research outlined solutions for minimizing power dissipation during normal (functional) operation mode. High-level power minimization techniques in [Abr90, Agr93a, Agr95] yield trade-off throughput, area and power dissipation during scheduling, allocation and binding. At logic level, two successful power management techniques, based on pre-computation [Agr93b, Agr94] and graded evaluation [Ait99], have been presented. However, to consider only power dissipation during the normal operation mode is not enough. It is essential to scrutinize it during test operation mode as well.

In [Zor93, Ali94], it is proposed that power dissipated during test application is significantly higher than power dissipated during normal

functional operation, which can lead to loss of yield and decrease the reliability of circuits under test. The reasons for high power dissipation during test application are as follows:

- 1). The correlation between consecutive test vectors generated by an automatic test pattern generator (ATPG) is very low, since a test is generated for a given target fault without any consideration of the previous test vector in the test sequence.

- 2). The use of design for testability (DFT) scan techniques destroy the correlation, which typically exists between successive states of the sequential circuit by allowing the application of any desired value to the state latches.

During the VLSI design flow, minimizing power dissipation increases the reliability and the lifetime of circuits [Cha95b, Rey00, Wed96]. It is reported [Ped96] that the Deterministic Dominant Factor of power dissipation is dynamic power dissipation caused by switching activities [Cha95b, Rey00]. The additional power dissipation in test mode is caused by significantly higher switching activity during testing than in functional operation. Techniques developed in the above references have successfully reduced the circuit power dissipation during functional operation.

Testing of low power circuits has recently become an area of concern for the following reasons:

Firstly, it is reported [Zor93] that there is significantly higher switching activity during Test Mode than during normal operation and, hence higher power dissipation in test mode. This can decrease the reliability of the circuit in Test Mode due to excessive temperature and current density. Circuit designed using power minimization techniques may not tolerate this.

Secondly, high switching activity during Test Application can lead to manufacturing yield loss, which can be explained as follows: High switching activity during test application causes a high rate of current flowing in power

and ground lines leading to excessive power and ground noise. And this noise can change the logic state of circuit lines leading to incorrect operation of circuit gates causing some good dies to fail the test [Wag98]. Hence, it has become an important issue to address the problems associated with testing low power VLSI circuits.

Spurious transitions (i.e. glitches) during functional operation do not have any useful function and cause useless power dissipation. So power can be saved during test application and during normal operation, by eliminating spurious transitions [Cha95b, Ped96, Rey00]. Many (new) techniques have been presented in the literature.

1. Memory optimization techniques
2. Hardware-Software partitioning
3. Instruction-level power optimization
4. Control-Data-Flow transformations
5. Variable-Voltage techniques
6. Dynamic power management
7. Interface power minimization
8. Approximate signal processing

Many techniques have been proposed to overcome the problem of high power dissipation during test application. Usually, the ordering of both scan flip-flops and the test patterns influences power and energy. Most of the techniques relate to BIST methodologies at logic level. They can be classified into those that apply to Test-per-Scan BIST schemes and those that apply to Test-per-Clock BIST schemes. In Test-per-Scan BIST systems, a test pattern is applied to the Circuit-Under-Test (CUT) via a scan chain every $m+1$ clock cycles, where m is the number of flip-flops in the scan chain. The response is

captured into the scan chain and scanned out during the next m clock cycles; meanwhile the next test vector can be scanned in simultaneously. A modification of the scan cell design is proposed in [Her98]. By this method, the Circuit-Under-Test inputs remain unchanged during operation, and a significant energy saving can be realized by means of this novel design for scan path elements. A low transition random pattern generation technique is proposed in [Wan99]. Using this technique, signal activities in the scan chain can be reduced. Using a K -input Gate and T Latch a high correlation between neighboring bits in the scan chain can be generated. Consequently, the number of transitions, and thus the average power dissipation is significantly reduced.

A post ATPG phase technique is proposed in [Cha94a, b, c], to reduce power dissipation for full-scan and for pure combination [Dab94] circuits. In [Dab98], the authors summarize the above techniques and use a transition graph for low power consumption in scan circuits and combination circuits. Firstly, in the full-scan case, a fixed scan-latch ordering is assumed and then, using a greedy heuristic, a test-vector ordering is computed so as to minimize the power dissipation during test application. Secondly, two heuristics are used to minimize power dissipation. Scan-Latch Ordering uses the Random Ordering Heuristic and Test-Vector Ordering uses Simulated Annealing. Finally, by the methods of circuit disabling, switching activity is inhibited in the embedded combination circuits, meanwhile the test values are scanned-in and scanned-out.

In test-per-clock BIST systems, the outputs of a test pattern generator are connected directly to the inputs of the CUT. A new test pattern is applied at each clock cycle and the response is loaded into the response analyzer. By generating test vectors from TPGs that cause fewer transitions at circuit inputs, switching activity in the CUT can be reduced. In the same vane, a BIST strategy based on two different speed LFSRs is proposed in [Wan97b]. Its

objective is to decrease the overall internal activity of the circuit by means of connecting inputs with elevated transition density to the slow LFSR. This approach can reduce the average power consumption without any loss of fault coverage.

A technique named as Reseeding Scheme in conjunction with a Vector Inhibiting Technique, is proposed in [Gir99a] for the purpose of minimizing the energy dissipation during test. This is an effort at tackling the increased activity during test operation of hard-to-test circuits that contain pseudo-random resistant faults. An improvement of this technique is proposed in [Man99], where the filtering action is extended to all the non-detecting vectors of the pseudo-random test sequence. However, a circuit cannot be prevented from excessive peak power consumption by these techniques.

In [Moh02], an approach for reducing power consumption in the checkers used for Concurrent Error Detection (CED) is proposed. Spatial correlation between the outputs of the circuit that drives the primary inputs of the checkers is analyzed to order them such that switching activity (and hence power consumption) in the checker is minimized. The reduction in power consumption comes at no additional impact to area or performance and does not require any alteration to the design flow. The only cost involved is the computing time in the input ordering for the checker that minimizes the power consumption.

3.2.2 Power-Constraint Test Scheduling

Most of the Block-Test Scheduling techniques proposed so far, are only addressed at logic-level in order to schedule for test time minimization by using parallelism, or to schedule for area overhead optimization by sharing test resource in data path blocks [Cra88]. These techniques are certainly valid for

logic-level, or at most, RT level blocks. Unfortunately, their function is limited. They cannot schedule BIST of parallel blocks for a complex VLSI device.

In [Zor93], the BIST scheduling approach has taken power dissipation into account during block-test scheduling for the first time. This approach not only performs global optimization, but also considers other factors such as block type, adjacency of blocks (device floor plan). The latter are unknown at high-level, however. In complex VLSI circuit design, the block test set is large and varies in test length. So it is impractical to expect that this approach can provide any polynomial complexity algorithm. This approach is useful only for defining and analyzing the problem.

Even if there is no resource conflict on a pair of tests, it still does not necessarily mean the two tests can be performed concurrently. This is because the combined power consumption must be ensured carefully not to exceed the maximum power limit. One example is memories. Usually they are organized into blocks of a fixed size. In normal operation mode, just one block is activated per memory access at the same time; other blocks are in the power-down mode so as to minimize the power consumption. For memory system in test mode, it is desirable to concurrently activate as many blocks as possible so as to minimize test time provided that the power consumption limit of the system is not exceeded.

Testing of MCMs is another example. For testing MCMs, an attractive approach is to use BIST block executing in parallel. In normal operation mode, the blocks are not activated simultaneously. So the inactivated blocks do not contribute significantly to total power dissipation. However, in test operation mode, a concurrent execution of BIST in many blocks will bring significantly higher power dissipation, and it might exceed the maximum power dissipation limit. In consideration of the reliability of a digital system

under test, execution of self-test blocks must be scheduled carefully following a certain principle so that the maximum power dissipation limit is not exceeded at any time during testing.

The problem of minimizing power dissipation during test applications is addressed at a higher level in [Cho94, Cho97]. In [Cho97], where the problem of scheduling equal length tests with power constraints is formulated. The objective is to find a test schedule with power-constraint, which covers every test in at least one test session, so that the total test application time is minimized. The solution is divided into two steps: identifying the solution space, and then, searching the solution space for an optimum solution. For solution space identification, the following definitions are given. The first one is Power Compatible Set (PCS), which is a set of tests that can be performed concurrently. The second one is Maximum PCS (MPCS), which is a subset of PCS, in which no compatible tests can be added without exceeding the maximum power consumption limit.

Macro Test (MT), is an approach being used to test embedded modules as stand-alone units, and is very suited for core-based testing. From this point of view it is very suitable for hierarchical and divide and conquer approaches. Macro Test is based on the following concept [Mar97]. A test can be divided into a test protocol and test patterns, where the test protocol gives the regulation on how to apply the test patterns to the inputs and how to observe the outputs of the macro under consideration. Through test protocol expansion, a translation of macro-level test to IC-level test is performed. Macro Test and test protocol expansion are designed to support multiple levels of hierarchy in a design. Once the various core tests are expanded to chip level, they can be applied in a simple sequential order, or be scheduled by the test protocol scheduler [Mar99a]. Test Protocol Scheduler attempts to perform the various cores tests in parallel as much as possible so that the Test Application Time

(TAT) can be reduced. However, the possibility of power dissipation increases while reducing the TAT is not taken into account yet. Macro Test (MT) supports every kind of test accesses, including parallel direct access, serial scan and BIST.

3.3 PTS Problem Modeling

The modeling of PTS problem will be discussed in detail in this section. It comprises of five subsections: System Modeling, High-Level Power Dissipation Estimation, PTS Problem Formulation, The Tree Growing Technique and Power-Test Scheduling Chart.

3.3.1 System Modeling

The problem of Power-Constrained Block-Test Scheduling (PTS) was first theoretically analyzed in [Cho97] at IC level. Generally, it can be viewed as a compatible test-clustering problem, which is a known NP-Complete problem. A merging approach based on the tree growing technique is proposed in this thesis to tackle the PTS problem. The approach has a polynomial complexity, and this is very important for the system level test scheduling efficiency. The proposed approach deals with the so-called Unequal-Length block test schedule problem, i.e., the tests for blocks of logic are of unequal length. In this approach, the order of the tests within a block test set is not considered.

3.3.2 High-Level Power Dissipation Estimation

When we talk about power in relation to the current topic we usually mean the instantaneous power.

Instantaneous power can be represented as $p(t)$, which is the power dissipation value at any time instant t : $p(t) = i(t) \times v(t)$, where $i(t)$ and $v(t)$ are the instantaneous current and voltage in the circuit respectively. In general, the voltage is a constant and equals to the power supply, i.e., $v(t) = V_{dd}$.

Provided $p_i(t)$ is the instantaneous power dissipation of test t_i and $p_j(t)$ is the instantaneous power dissipation of test t_j , then the total power dissipation of a test session (i.e., two overlapped tests) is approximately the sum of the instantaneous power of test t_i and test t_j . This relation as depicted in [Cho97] is shown in Figure 3.3. Normally it is unacceptable for the instantaneous power to exceed a maximum power dissipation limit P_{max} , because the IC might be destroyed if this occurs. Unfortunately, the instantaneous power of test vectors are difficult to obtain, as different test schedules will result in diverse instantaneous power dissipation profiles for the same test.

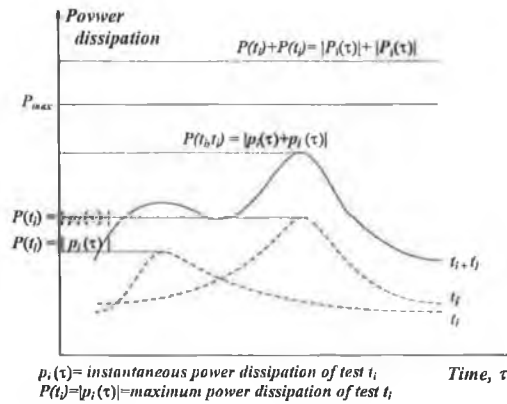


Figure 3.3: Power Dissipation as a Function of Time.

In order to simplify the analysis, a fixed power value P_i is assigned to all test vectors in test t_i so that at any time instant the power dissipation is no higher than P_i .

To evaluate the power properties of BIST architecture several parameters are important; most of which are detailed in [Ger99]. The consumed power is directly determined by the switching activity and affects the battery lifetime or junction temperature during test. The maximum power corresponds to the maximum power consumption rate during test time. If the maximum power limit is exceeded, the IC may be destroyed. The Time-average power (Average power) is the total consumed power divided by the test time. This parameter affects reliability caused by constant high-power consumption.

The approach for power analysis as described above, is suitable for work with the proposed algorithms in this thesis. Accurate high-level power evaluation is impossible, so power estimation is the only viable solution. A constant additive model is employed for power estimation. For the purpose of simplification, only a constant Power Dissipation (P_i) value is associated with each block-test. As to the total power dissipation at a given test schedule time, it is only the P_i summing relation of the running block-tests.

Usually there are three ways to estimate the Power Dissipation, P_i of a block-test at a high level:

- **Maximum P_i ,**
- **Average P_i and**
- **RMS P_i .**

Firstly, P_i can be defined as the Maximum power dissipation (Peak Power) over all test vectors in test t_i . It is the upper bound power dissipation in test t_i , and its definition is pessimistic. In this case, two tests t_i and test t_j , where peak power occurs at different time intervals, are not allowed be scheduled in the same test session.

Secondly, P_i can be defined as the Average power dissipation over all test vectors in a block-test t_i . In the analysis of power dissipation, its definition might be optimistic when many test vectors are applied simultaneously, as the

average value cannot describe the instantaneous power dissipation of each test vector. Therefore, at some time intervals, it is possible that the power dissipation limit of the IC might be exceeded.

Thirdly, RMS power dissipation is needed to tackle the problem when instantaneous power dissipation includes short power spikes and a more accurate estimation is sought.

3.3.3 PTS Problem Formulation

The circuit activity should be maximized so that the circuit can be tested thoroughly in the shortest possible time. However, in a test environment, the difference between the various power estimation values for each test is very small [Cho97]. In this thesis the lowest level block considered is at the RTL (Register Transfer Level) in the test hierarchy, and it is assumed that a test-step level scheduling has already been applied at this level. Additionally, by using the approach proposed here for optimizing the blocks in the test hierarchy from the lowest level (RTL) to the top level (System Level), the difference between the power values could be further reduced. The reason is that, after applying the PTS algorithm at each level, the circuit activity or power consumption is maximized and balanced.

So P_i can be viewed as the maximum Power Dissipation over all tests vectors in test t_i [Cho97]. In further analysis, P_i is assumed to be the maximum Power Dissipation of test t_i .

3.3.4 The Tree Growing Technique

A tree growing technique was first proposed in [Jon89] and further developed in [Mur00a, b].

It is used to exploit the potential of test parallelism by merging and constructing the Concurrent Test Set (CTS). This is achieved by means of a Binary Tree Structure (not necessarily complete), called Compatibility Tree, which is based on the compatibility relation between tests.

A drawback in the original technique [Jon89] is that the compatibility trees are binary trees. This limits the number of children test-nodes that could be

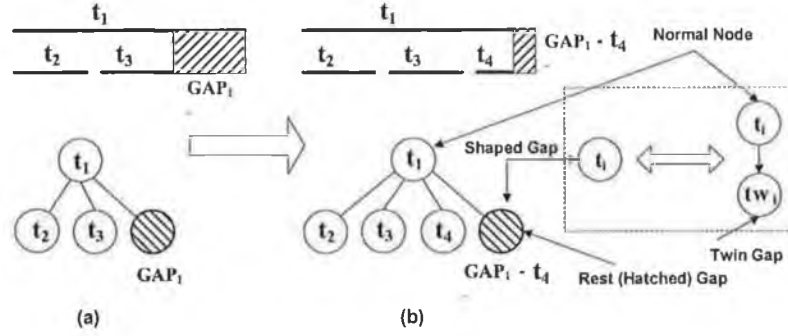


Figure 3.4: Merging Step Example.

overlapped to the parent test node to only two. The number of children test nodes in practice can be larger than two, as in the example depicted in Figure 3.4. [Mur00c]. An Extended Compatibility Tree (ECT), given by means of a generalized tree, is proposed in [Mur01] to break this limitation.

The compatibility relationship comprises three components.

- (1) The power dissipation accumulated on each tree branch should not exceed the power dissipation constraint P_{max} .
- (2) The test lengths of the nodes in a tree branch should be non-increasing from root to leaf. In other words, the boundary of test sessions cannot be broken when growing the tree.
- (3) Tests have to be compatible from the resource usage point of view.

In the merging step example in Figure 3.4, the partial test schedule chart is given at the top, while the partially grown compatibility tree is at the bottom. Let's assume as follows:

(1) Tests t_2 , t_3 and t_4 are compatible with test t_1 , while they are not compatible with each other;

(2) T_1 , T_2 , T_3 and T_4 are the test lengths of test t_1 , t_2 , t_3 and t_4 respectively;

(3) $T_2 + T_3 < T_1$;

(4) $T_4 < T_1 - (T_2 + T_3)$.

As can be seen in Figure 3.4(a), there is a gap GAP_1 given by the following test length difference: $GAP_1 = T_1 - (T_2 + T_3)$ after tests t_1 , t_2 and t_3 have been scheduled. So a merging step can be achieved, because $T_4 < GAP_1$, by inserting test t_4 in the partial test schedule and its associated ECT as in Figure 3.4(b).

The process of constructing CTS's is implemented by growing the ECT from the roots to their leaf nodes. The root nodes are regarded as test sessions, whereas the expanded tree branches are regarded as their test sub-sessions. When a new test has to be merged to the CTS, the algorithm should avail of all possible branches in the ECT. In order to keep track of the available tree branches and to avoid the complexity of the generalized tree travel problem, a list of potentially Expandable Tree Branches (ETB) is maintained. This list is kept by means of special nodes that are inserted as leaf nodes in each ETB of the expanded compatibility tree. These leaf nodes are called *gaps* and are depicted as hatched or shaded nodes in Figure 3.5.

There are two types of *gaps*. The first set of *gaps* (hatched), called "remnants *gaps*" are those left behind each merging step, as in the cases of GAP_1 and $GAP_1 - t_4$ in the above example. They are similar to the incomplete branches of the binary tree from [Jon89].

The second set of *gaps* (shaded), are auxiliary *gaps* created as the superposition of the leaf nodes and their twins as in the equivalence given at the right in Figure 3.5. They are generated in order to keep track of "non-saturated" tree branches, which are also potential ETBs. "Non-saturated" tree branch means any ETB with accumulated power dissipation still under the given power dissipation limit. The root nodes (test sessions) are considered by default "shaded" *gaps* before being expanded.

3.3.5 Power-Test Scheduling Chart

A Test Schedule generated by the so-called List Scheduling-Based PTS Algorithm (PTS-LS) is given in Figure 3.5 [Mur00b]. It can be easily translated into a PTS chart as in Figure 3.6, which gives a clear view of the power dissipation distribution over the test application time.

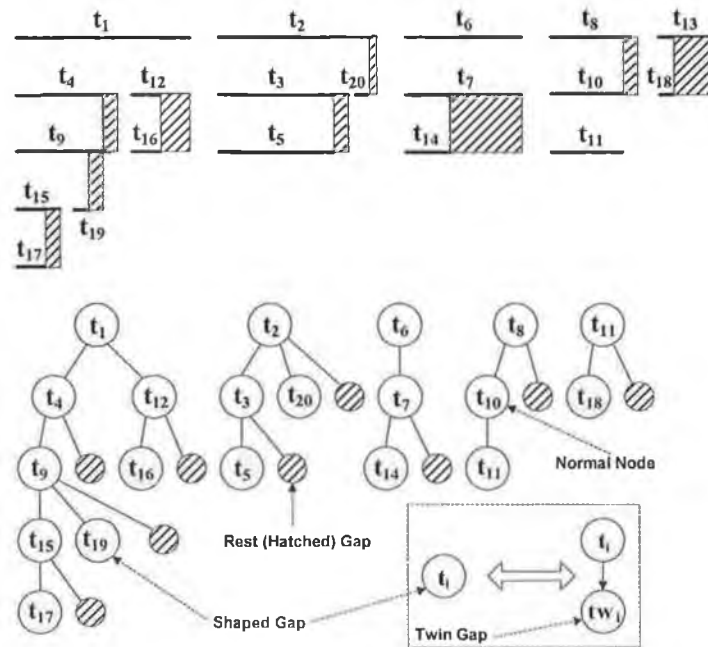


Figure 3.5: Test Scheduling Chart and ECT Example.

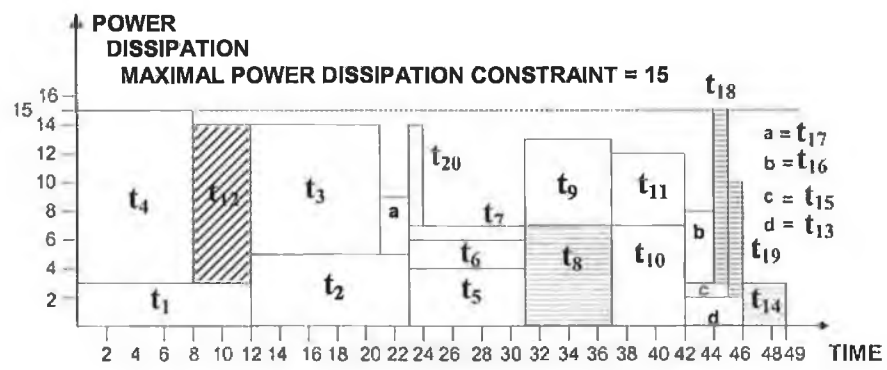


Figure 3.6: An Example of PTS Chart of 20 Tests with Power Dissipation Constraint.

Chapter 4

Merging Approach Based on the Tree Growing Technique

4.1 Introduction

The goal of this chapter is to seek an approach with better efficiency and lower computational cost, e.g., less computational time. The comparison of existing test scheduling approaches is given in Section 4.2. In Section 4.3, our approach, the merging approach based on tree growing technique, is described in detail, including the description of the operating procedures, algorithm pseudocode, the analysis of the algorithm complexity and a test schedule example. In Section 4.4, the conclusion about this chapter will be given.

4.2 Existing Test Scheduling Techniques

Many approaches have been proposed to solve the test-scheduling problem. Zorian takes into account for the first time the problem of the power

dissipation during test scheduling [Zor93], however, his work focuses mainly on the definition of the problem itself rather than proposing a solution. Chou *et al.* give for the first time a thorough analysis of the power constrained test scheduling (PCTS) at IC level [Cho97]. They use a compatible test clustering technique that is based on the compatibility of tests, to produce the power compatible set (PCS), and apply the minimization technique of the weighted cover table to obtain an optimum schedule. However, this work is confined to be a limited theoretical analysis, because the computation is too excessive due to the enormous covering table generated. Muresan *et al.* propose an Extended Compatibility Tree technique [Mur00a, b] to exploit the potential of test parallelism by merging the block-test intervals of compatible sub-circuits to expand compatible tree. Although the effect of filling in the idle time with shorter tests based on the compatibility relations among the tests is improved, this approach has a drawback, e. g., test stretch is restricted by the test session boundaries. This approach can generate a good enough result, but there still exists room for improvement. To get an even better result, it is possible to take this result as a starting point, and apply some optimization techniques, such as, Simulated Annealing (SA) [Kir83], Tabu Search (TS) [Ree93] and Genetic Algorithm (GA) [Hol75]. Larsson *et al.* propose an algorithm [Lar01a] to increase the parallelism by greedy approach, however, the efficiency that they define the function of rectangle to model block test is not high. Chakrabarty *et al.* think the problem of test schedule as open shop problem. They tackle the test schedule problem [Cha01] [Lar01] using a mixed integer linear programming (MILP) approach combined with power constraints. However the computational time of MILP grows exponentially with the number of cores and test resources.

Muresan *et al.* apply the tree growing technique [Jon89] to the field of Power-Constrained Block-Test Scheduling, and improve the tree growing

technique to Extended Compatibility Tree (ECT), which is a practical solution to the problem of PTS.

4.3 Merging Approach Based On Tree Growing Technique

The main limitation of the extended compatibility tree approach is that test stretch is restricted by the test session boundaries.

Because of the boundary limitation, the result produced by extended compatibility tree technique can be improved further by our approach, the merging approach based on tree growing technique, with only a small increasing of computational cost. This is the focus of the thesis.

The proposed merging approach breaks the boundary limitation of the extended compatibility tree technique, it allows test schedule overlap as shown in Figure 4.1.

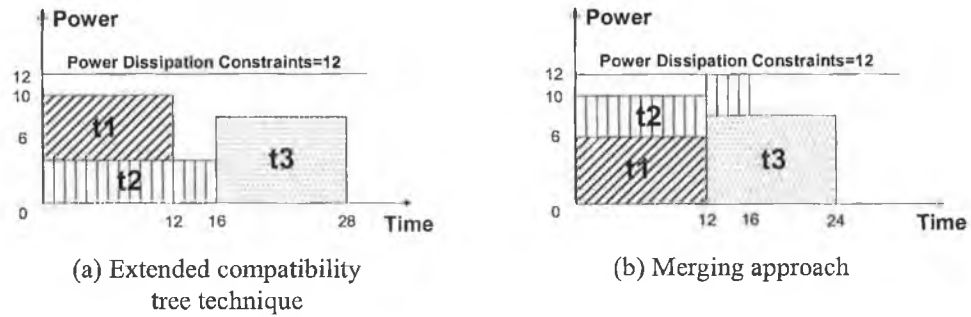


Figure 4.1: An example of acceptable overlap of tests by merging approach.

Once break the boundary limitation, the actual computational cost will grow exponentially. If test scheduling is made initially with extended compatibility tree approach, and then break some boundaries of some test sessions in the schedule generated by ECT for further improvement by merging approach, a better test schedule might be achieved, with limited increase of computational cost.

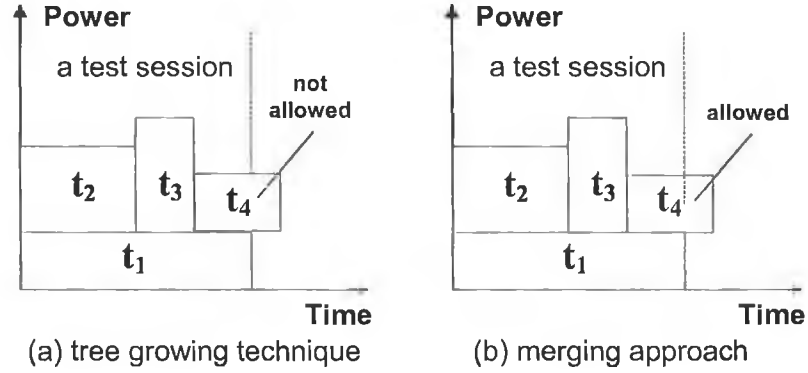


Figure 4.2: The comparison of the merging approach with tree growing technique (Extended Compatibility Tree).

4.3.1 Operating Procedures

From PCS to EMC

The power compatible set (PCS) notion is introduced in [Cho97], on this basis, Muresan *et al.* propose extended compatibility tree (ECT) approach in [Mur00a, b], and efficient algorithm that can produce PCS. We give the notion of PCS a new meaning here. We use it as extended main clique (EMC) consisting of several block-tests.

The Test Length of EMC

The test length of an EMC is the test length of the root block test, in the corresponding PCS generated by the ECT approach, as is illustrated in Figure 4.3.

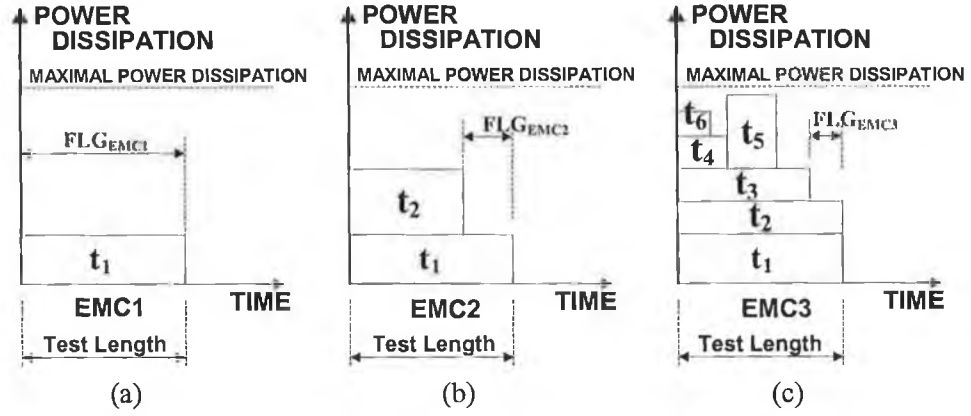


Figure 4.3: EMC Test Length and the First Level Gap Length.

The Feature of EMC

The main feature of EMC is that the positions of block tests in an EMC is only fixed relatively, there is a certain degree of freedom within a time window determined by the corresponding test session boundaries. For example, the scheduling results in Figure 4.4 are considered to be equivalent in the proposed merging approach. In this example, block-test t_2 has the freedom of being scheduled in any position within the time window of block-test t_1 .

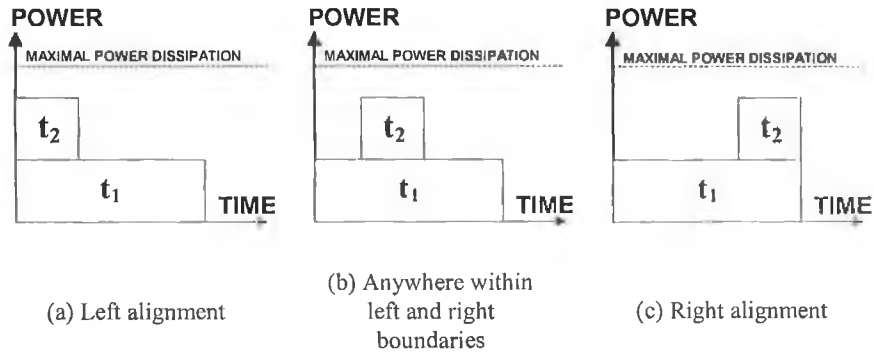


Figure 4.4: Equivalent Test Schedule in Merging Approach.

The First Level Gap Length in EMC

First Level Gap (FLG) is defined as the test length difference between the root test and the second longest test in a test session produced by the ECT approach. The length of the root test is the length of the test session and is also the length of the corresponding *EMC*. We use notation FLG_{EMCi} to represent the first level gap of $EMCi$. For example, in Figure 4.3 (b) FLG_{EMC2} represents the first level gap of EMC_2 , with the length of FLG_{EMC2} equals to $T_1 - T_2$; in Figure 4.3(b) FLG_{EMC3} represents the first level gap of EMC_3 with the length of $T_1 - T_3$. T_1 , T_2 and T_3 represent the length (i.e., test application time) of tests t_1 , t_2 and t_3 respectively. Similarly, we can also define *Second Level Gap* as the difference of the second and the third longest tests in a test session, i.e., in an *EMC*. Third level and higher level gaps in an *EMC* can also be defined. However, to avoid the complicated compatibility relations between block tests in different *EMC*'s, the proposed merging approach in this thesis deals only with the *first level gap*. That is to say, when we try to merge one $EMCi$ with another $EMCj$ in a view to reduce test application time, we only consider the merging possibility of FLG_{EMCi} and FLG_{EMCj} . In other words, only consider whether it is possible to overlap FLG_{EMCi} and FLG_{EMCj} , so that $T_{EMCi,j} < T_{EMCi} + T_{EMCj}$, where T_{EMCi} , T_{EMCj} represent the test lengths of $EMCi$, $EMCj$ respectively, and $EMCi,j$ represents the merger of $EMCi$ and $EMCj$.

The Method of Merging

It is too complicated to try to merge all *EMCs* together at the same time. To get a trade-off between efficiency and cost, we simplify the problem as follows:

From the initial block-test schedule obtained by extended compatibility tree approach [Mur00a], and according to the definition of *EMC* stated above, we get all *EMCs*, each corresponding to a test session in the ECT approach: $EMC_1, EMC_2, EMC_3 \dots EMC_k$. Try to merge EMC_1 with $EMC_2, EMC_3, \dots EMC_k$ respectively, select the pair of block-tests that gives the maximal test time saving if merged. After this pair of *EMC*'s is identified, put them in the new PTS chart. If there are several pairs *EMCs*, which give the same maximal savings, then the first pair will be selected. If there isn't any *EMC* can be merged with EMC_1 , then EMC_1 itself is put into the new PTS chart, and then the next remaining *EMC* is selected and to be merged with the other remaining *EMCs*, until all *EMCs* have been merged.

4.3.2 Algorithm Pseudocode

The biggest achievement of the tree growing technique is that proven efficient HLS algorithm can be easily applied to the PTS problem modeled as an extended tree growing process. Use the algorithm proposed by Muresan [Mur00a] to produce an initial schedule, i.e., to find all the *EMCs*, then the merging is executed to improve the initial schedule.

The algorithm for the initial block-test scheduling is described by the following pseudocode of the extended compatibility tree approach [Mur00a].

PSEUDOCODE 1:

- ◊ Sort all the tests by their mobility in two steps (test length, power dissipation);
- ◊ Initialize the *GrowingTree* and the *GapsList*;
- ◊ While there are unscheduled tests { /* *BlockTestList* is not empty */
 - If (*GapsList* is empty) then {
 - *CurTest* = head of *BlockTestList*;
 - Insert *CurTest* as the tail of *GrowingTree* roots; /* new test session */
 - Mark *CurTest* "used";
 - Remove *CurTest* from *BlockTestList*;
 - Generate a *TwinGap* gap as the twin of *CurTest*;
 - Insert *TwinGap* into *GapsList*; } /* if */
 - Else {
 - *CurGap* = head of *GapsList*;
 - *CurTest* = head of *Comp.List_{CurGap}*;
 - While *CurGap* is the head of *GapsList* AND *CurTest* did not reach the end of *Comp.List_{CurGap}* {
 - * If ($T_{CurTest} \leq T_{CurGap}$ AND $PD_{CurGap} + PD_{CurTest} \leq PD_{MAX}$ AND *CurTest* NOT "used") then {
 - Schedule (*CurTest*, *CurGap*, *GrowingTree*, *GapsList*, *BlockTestList*);
 - /* Schedule *CurTest* into the power-test scheduling Chart and inserts it into the *GrowingTree*, marks *CurTest* "used" */
 - Break;}
 - * Else *CurTest* = *CurTest* → *next*;
 - * /* next in the *Comp.List_{CurGap}* */
 - } /* While */
 - If (*CurGap* is still the head of *GapsList*) then
 - /* It means that there is no compatible test left for *CurGap* */
 - * Remove *CurGap* from the *GapsList*;
- /* Else */ /* while */

The data structures used to implement the algorithm are the following: the *GrowingTree* to model the ECB, the *GapsList* to model the list of potentially expandable gaps (shaded and hatched gaps), and the *BlockTestList* to keep the ordered but not yet merged tests. *CurTest* is the test to be merged at each iteration. *CurGap* is the gap under focus at each iteration in order to see whether it is expandable (compatible) with the *CurTest*. In the pseudocode the term “used” means that the test has already been merged in the ECB. *TwinGap* is the newly generated shaded gap at every iteration. It will not be inserted in the *GapsList* after its generation if its resulting compatibility list is null, i.e. it will not be an ETB. *RestGap* is meant to keep the non-null hatched gap generated at every iteration, i.e., *CurTest* does not cover *CurGap* completely, that is $T_{CurGap} > T_{CurTest}$. Additionally, T_{node} , PD_{node} and $Comp.List_{node}$ are, respectively, the test length, the power dissipation and the compatibility list of the node, which can be either a test or a gap. If a new gap (test subsessions) is generated inside the current one, the new one replaces the current gap in the *GapsList* and in the *GrowingTree*, and the procedure is repeated having a new *GapsList*.

As can be seen from the pseudocode itself, the algorithm is repeated until all the tests in the initial *BlockTestList* are scheduled in the ECT. If the list of currently available gaps (*GapsList*) is empty then a new test session (and indirectly a new gap) is generated with the current test, which is removed from the *BlockTestList*. If the *GapsList* is not empty then the first gap in the list is taken for further expansion. Its compatibility list is scanned starting with the test exhibiting the lowest mobility (long test length and high power dissipation). The first yet unscheduled test in the *BlockTestList*, which is compatible with the current gap, is scheduled in the Growing Tree generating two new gaps (twin and remaining). *BlockTestList* and *GapsList* structures are

then updated as well. If the current gap turns out to be unexpandable, it is removed from the *GapsList* and the process is repeated for the next gap in the list.

Having obtained a PTS chart, using the above algorithm, next we will improve the PTS chart using the following merging algorithm, so as to get a better block-test schedule. The pseudocode of the merging algorithm is as follows:

Pseudocode

```

/* Initialising the EMC(i), i = 1 to k; corresponding to the k test sessions
generated by the Tree Growing Technique, all EMCs are initially "unmerged"*/
EMClist = {EMC(i), i = 1 to k};

/* The merged EMC list is initially empty */
MergedEMClist = {null};

/* Try to merge an "unmerged" EMC with other "unmerged" EMCs */
FOR i = 1 to k -1 LOOP
    IF (EMC(i) is "unmerged") THEN
        /* Initialising maximum time saving by merging EMC(i) with other EMCs */
        max_saving = 0;

        /* m keeps the index of another EMC that can be merged with EMC(i), if
        EMC(i) cannot be merged with other EMC then it merges with itself, i.e., no
        merge */
        m = i;

        FOR j = i+1 to k LOOP
            IF (EMC(j) is "unmerged") THEN
                IF (EMC(i) can be merged with EMC(j)) THEN
                    IF ((saving of merging EMC(i) and EMC(j)) > max_saving
THEN

```

```

        /* updating the maximum time saving that can be achieved */
        max_saving = (saving of merging EMC(i) and EMC(j));

        /* remember the index of the EMC when merged with EMC(i)
           gives the maximum time saving */
        m = j;
    END IF;
END IF;
END FOR; /* FOR j = i+1 to k */

IF (max_saving > 0) THEN
    /* EMC(i) can be merged with EMC(m) */
    • Mark EMC(i) and EMC(m) “merged” in the EMCList
    • Copy merged EMC(i) and EMC(m) to the MergedEMCList

ELSE
    /* EMC(i) can not be merged with any other EMC, m still equals i */
    • Mark EMC(i) “merged” in the EMCList
    • Copy EMC(i) to the MergedEMCList
END IF;
END IF;
END FOR; /* FOR i = 1 to k – 1 */
IF (EMC(k) is still “unmerged”) THEN
    • Mark EMC(k) “merged” in the EMCList
    • Copy EMC(k) to the MergedEMCList
END IF;

```

The *EMCList* contains a list of Extended Main Cliques each corresponds to a test session generated by the Extended Compatibility Tree technique. Initially all *EMCs* are marked “unmerged”. The *MergedEMCList* contains the

list of merged *EMCs*, which is initially empty. The two nested FOR loops implement the process of merging. With the outer FOR loop, one “unmerged” *EMC*, the *EMC(i)*, is taken from the *EMClist* at a time and the maximum time saving by merging *EMC(i)* with another *EMC* is reset to 0 ($\text{max_saving} = 0$). Attempt is then made to merge *EMC(i)* with each “unmerged” *EMC* using the inner FOR loop.

If the max_saving is greater than zero at the end of the inner FOR loop, this means *EMC(i)* can be merged with at least one other *EMC* and, the index of the *EMC* that generates the largest time saving when merged with *EMC(i)* is kept in variable *m*.

The new *EMC* created by the merging of *EMC(i)* and *EMC(m)* is put into the *MergedEMClist*. *EMC(i)* and *EMC(m)* are not removed from the *EMClist* after merging because removing them from the *EMClist* will cause the FOR loops to collapse, instead they are both marked “merged” in the *EMClist* so that they will not be taken for further merging.

If the max_saving remains zero at the end of the inner FOR loop, this means *EMC(i)* can not be merged with any other *EMC* and variable *m* must still equal to *i* which is the initial value before get into the inner FOR loop. The *EMC(i)* is marked “merged” in the *EMClist*, and a copy of it is put into the *MergedEMClist*.

At the end of the outer FOR loop each *EMC* in the *EMClist* upto the second last one should have been marked “merged”. The final IF statement makes sure that the last *EMC* in the *EMClist* is not left behind. If it is not marked “merged” in the previous merging steps, now it is marked “merged” in the *EMClist* and a copy of it is put into the *MergedEMClist*, and the merging process is complete.

EMC(i) and *EMC(j)* can be merged if and only if

1. test(s) in the first level gap of $EMC(i)$ and the test(s) in the first level gap of $EMC(j)$ are compatible
2. the sum of the power dissipation in the first level gap of $EMC(i)$ and the power dissipation in the first level gap of $EMC(j)$ does not exceed the power constraint

4.3.3 Algorithm Complexity

The complexity of the algorithm in our approach is given next.

The algorithm complexity of all pseudocode of merging approach based on tree growing technique is $O(N^2)$, where N is the number of block tests.

In tree growing technique, this is given by the two nested while loops, one to run through the *GapList* and another one to run through the *BlockTestList*. In the algorithm, the number of tests in the *BlockTestList* is initially N , but it decreases each step by one. In the merging algorithm, the complexity of the two nested for loops is $O(k^2/2)$, where k is the number of test sessions generated by the extended compatibility tree approach. K is general much smaller than N . $O(k^2/2)$ is much smaller than $O(N^2)$, so the overall computational complexity of the merging approach based on tree growing technique can still be considered as $O(N^2)$.

4.3.4 Test Scheduling Example

The following example should provide a deeper insight into the workings and the results of the proposed algorithms. The first part, before the merging procedure, is introduced from [Mur00a].

Figure 4.5 depicts the power-test scheduling results using extended compatibility tree approach generated with power dissipation constraint for the tests given next.

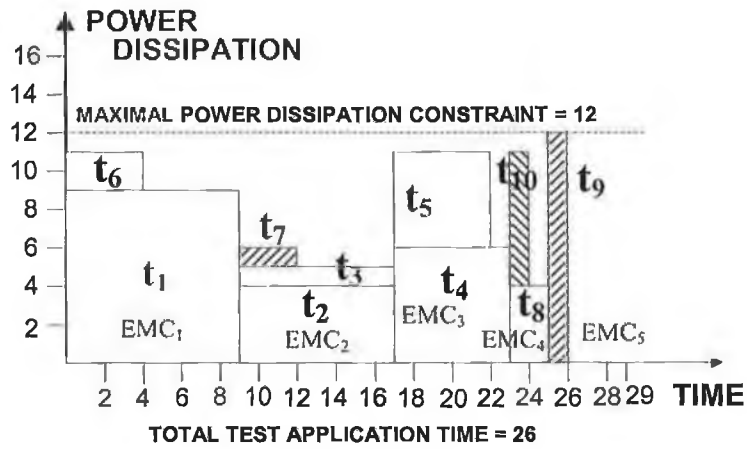


Figure 4.5: Power-Test Scheduling Result by ECT Approach.

Suppose the following ten tests (10 BTS) are to be scheduled under a maximal power dissipation constraint (PDC = 12) and that their parameters are specified in the order: *power dissipation, test length* and *their compatibility list*.

$test_i$ (power dissipation, test length, { *test compatibility list* })

For simplicity reasons, the tests listed below are already ordered by test length and power dissipation, as depicted in Figure 4.6.

t_1	(9, 9, { $t_2, t_3, t_5, t_6, t_8, t_9$ })
t_2	(4, 8, { t_1, t_3, t_7, t_8 })
t_3	(1, 8, { $t_1, t_2, t_4, t_7, t_9, t_{10}$ })
t_4	(6, 6, { t_3, t_5, t_7, t_8 })
t_5	(5, 5, { t_1, t_4, t_9, t_{10} })
t_6	(2, 4, { t_1, t_7, t_8, t_9 })
t_7	(1, 3, { $t_2, t_3, t_4, t_6, t_8, t_9$ })
t_8	(4, 2, { $t_1, t_2, t_4, t_6, t_7, t_9, t_{10}$ })
t_9	(12, 1, { $t_1, t_3, t_5, t_6, t_7, t_8, t_{10}$ })
t_{10}	(7, 1, { t_3, t_5, t_8, t_9 })

Figure 4.6: Test Compatibility List of 10 Tests.

The initial values for the data structures used inside the algorithm are:
GrowingTree (GT) = 0, *GapsList* (GL) = 0, *BlockTestList* (BTL) =
 $\{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}\}$, *CurrentTest* (ct) = 0, *CurrentGap* (cg) = 0,
TwinGap (tw)=0, *RestGap* (rg) = 0, while $PD_{max} = 12$ is the power dissipation
constraint. Since the number of tests to be scheduled is ten, there are ten main
steps all together, which are depicted in Figure 4.7.

Step 1. The first test is selected from *BTL* ($ct = t_1$) in order to merge it to
the *GT* but, since *GL* is initially empty, the first test session is generated (see
the first step from Figure 4.7). A twin gap tw_{t_1} is generated and inserted in *GL*
so that $GL = \{tw_{t_1}\}$, while the t_1 node inserted into *GT* is shaded.

Step 2. At the beginning of the second step $BTL = \{t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}\}$
and $GL = \{tw_{t_1}\}$. Thus, $ct = t_2$ and $cg = tw_{t_1}$. Even though ct and cg are
compatible from the test length and the resource point of view, the accumulate
power dissipation would be $PD_{t_2} + PD_{tw_{t_1}} = 13$, which is higher than the PD_{max}
constraint. Therefore, t_1 and t_2 cannot run in parallel and the solution is

sequential as in the second step of Figure 4.7. After this step $BTL = \{t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}\}$ and $GL = \{tw_{i2}, tw_{i1}\}$.

Step 3. The next test to be scheduled is $ct = t_3$, while the head of GL is $cg = tw_{i2}$. Because ct and cg are compatible from all points of view, they can be scheduled in parallel. A rest gap rg is not generated here because $T_{i3} - T_{tw_{i2}} = 0$, thus t_2 (tw_{i2}) and t_3 overlap completely. A twin gap $tw = tw_{i2,3}$ is generated though with the following parameters: $T_{tw_{i2,3}} = T_{i3}$, $PD_{tw_{i2,3}} = PD_{tw_{i2}} + PD_{i3} = 5$ and $Comp.List_{tw_{i2,3}} = Comp.List_{tw_{i2}} \cap Comp.List_{i3} = \{t_1, t_7\}$. The new *GapsList* is $GL = \{tw_{i2,3}, tw_{i1}\}$, while the test list is $BTL = \{t_4, t_5, t_6, t_7, t_8, t_9, t_{10}\}$.

Step 4. During the 4th step, the test $ct = t_4$ has to be scheduled. Initially, $cg = tw_{i2,3}$ is checked for compatibility with $ct = t_4$, but they are not compatible because $Comp.List_{tw_{i2,3}} = \{t_1, t_7\}$, and $t_4 \notin Comp.List_{tw_{i2,3}}$. Thus, the algorithm proceeds to the next gap in GL , which is $cg = tw_{i1}$, but t_4 is not compatible with t_1 either. Therefore, a new test session is generated for t_4 and, consequently, a twin gap tw_{i4} is also generated, updating $GL = \{tw_{i4}, tw_{i2,3}, tw_{i1}\}$ and $BTL = \{t_5, t_6, t_7, t_8, t_9, t_{10}\}$.

Step 5. For this step $ct = t_5$ and $cg = tw_{i4}$, and they are compatible from all points of view. Thus, a *RestGap* and a *TwinGap* have to be subsequently generated and then inserted into the *GapsList* and *GrowingTree* structures. The *RestGap* rgt_4 has the following parameters:

$$T_{rgt_4} = T_{i4} - T_{i5} = 1, PD_{rgt_4} = PD_{i4} = 6$$

and $Comp.List_{rgt_4} = Comp.List_{i4} = \{t_3, t_5, t_7, t_8\}$. The *TwinGap* $tw_{i4,5}$ has the following parameters: $T_{tw_{i4,5}} = T_{i5} = 5$, $PD_{tw_{i4,5}} = PD_{tw_{i4}} + PD_{i5} = 11$ and $Comp.List_{tw_{i4,5}} = Comp.List_{i4} \cap Comp.List_{i5} = \{\emptyset\}$, and, therefore, it will not be inserted into the *GapsList* anymore. Thus, after this step $GL = \{rgt_4, tw_{i2,3}, tw_{i1}\}$ and $BTL = \{t_6, t_7, t_8, t_9, t_{10}\}$.

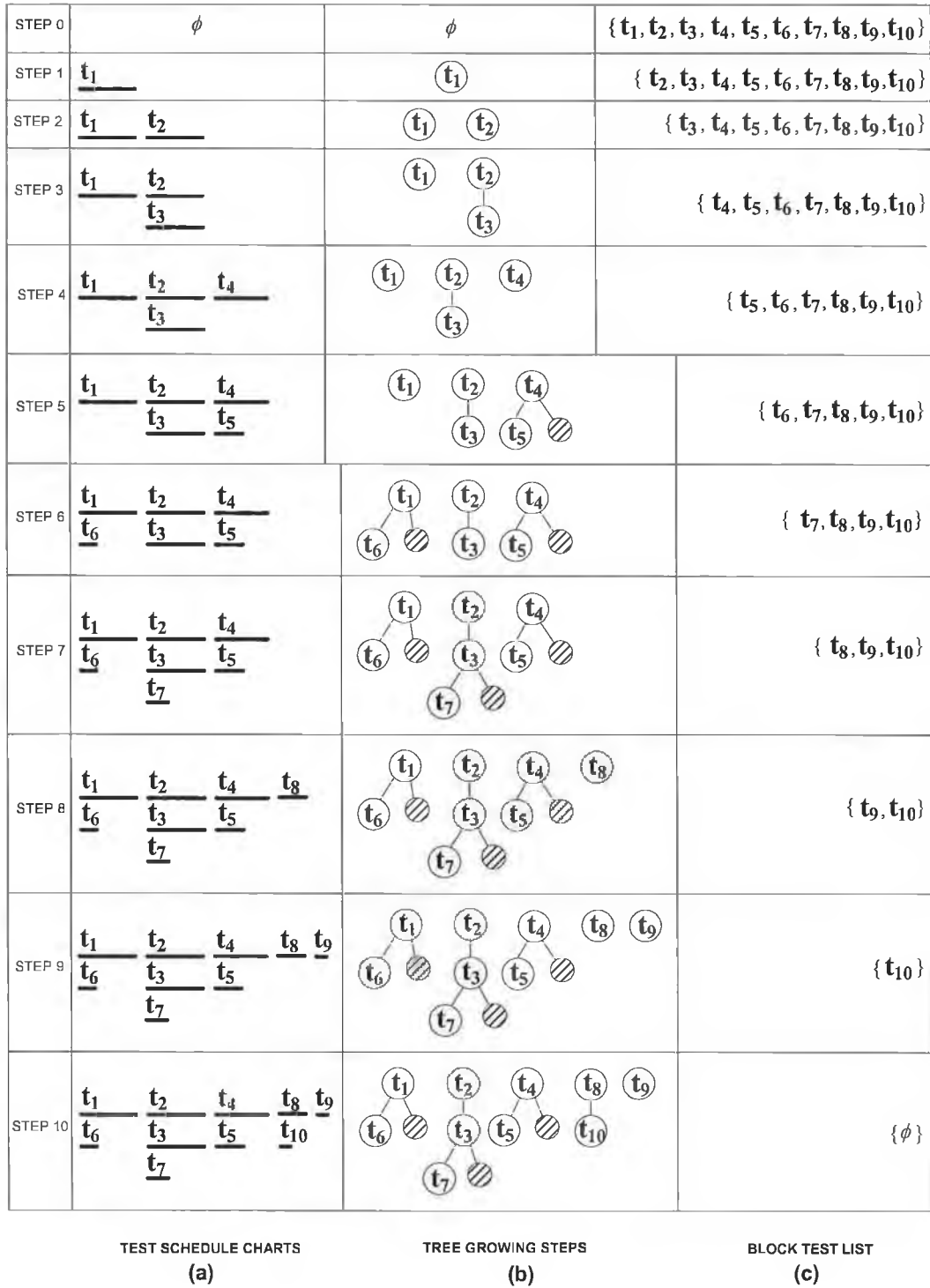


Figure 4.7: Tree Growing Steps Example.

Step 6. During this step the test $ct = t_6$ has to be scheduled. The algorithm goes through the *GapsList* starting with $cg = rg_{i4}$ (not compatible from the resource point of view), then $cg = tw_{i2,3}$ (not compatible from the resource point of view), and ending with $cg = tw_{i1}$. The last gap, $cg = tw_{i1}$, is compatible with $ct = t_6$. A *RestGap* $rg = rg_{i1}$ is generated having the following parameters: $T_{rg_{i1}} = T_{i1} - T_{i6} = 5$, $PD_{rg_{i1}} = PD_{i1} = 9$ and $Comp.List_{rg_{i1}} = Comp.List_{i1} = \{t_2, t_3, t_5, t_6, t_8, t_9\}$. The *TwinGap* $tw_{i1,6}$ is generated with the following parameters: $T_{tw_{i1,6}} = T_{i6} = 4$, $PD_{tw_{i1,6}} = PD_{tw_{i1}} + PD_{i6} = 11$ and $Comp.List_{tw_{i1,6}} = Comp.List_{tw_{i1}} \cap Comp.List_{i6} = \{t_8, t_9\}$. Then both gaps will be inserted into the *GapsList* $GL = \{tw_{i1,6}, rg_{i1}, rg_{i4}, tw_{i2,3}\}$, while $BTL = \{t_7, t_8, t_9, t_{i0}\}$.

Step 7. In order to schedule $ct = t_7$, the algorithm has to find firstly a gap compatible with it. t_7 is incompatible with $cg = tw_{i1,6}$ and $cg = rg_{i1}$ from test resources point of view. t_7 is also incompatible with $cg = rg_{i4}$ because the gap's test length is shorter than the test length of t_7 . However, t_7 is compatible with $cg = tw_{i2,3}$. Therefore, a *RestGap* $rg = rg_{i2,3}$ is generated having the following parameters: $T_{rg_{i2,3}} = T_{tw_{i2,3}} - T_{i7} = 5$, $PD_{rg_{i2,3}} = PD_{tw_{i2,3}} = 9$ and $Comp.List_{rg_{i2,3}} = Comp.List_{tw_{i2,3}} = \{t_1, t_7\}$. Because both t_1 and t_7 have already been scheduled at this stage, and $rg_{i2,3}$ is not compatible with any other tests, it would be pointless to insert this gap into the *GapsList*. The *TwinGap* $tw_{i2,3,7}$ has the following parameters: $T_{tw_{i2,3,7}} = T_{tw_{i7}} = 3$, $PD_{tw_{i2,3,7}} = PD_{tw_{i2,3}} + PD_{i7} = 6$ and $Comp.List_{tw_{i2,3,7}} = Comp.List_{tw_{i2,3}} \cap Comp.List_{i7} = \{\emptyset\}$. Because its compatibility list is empty, it will not be inserted into the *GapsList* either. After this step $GL = \{tw_{i1,6}, rg_{i1}, rg_{i4}\}$, while $BTL = \{t_8, t_9, t_{i0}\}$.

Step 8. The $ct = t_8$ test cannot be scheduled in $cg = tw_{i1,6}$ because the accumulated power dissipation would overflow, it cannot be merged with $cg = rg_{i1}$ for the same reason, and cannot be scheduled in $cg = rg_{i4}$ because the test length left $T_{rg_{i4}} = 1$ is not enough for $T_{i8} = 2$. Thus, a new test session t_8 is

generated together with its twin gap tw_{t8} ($PD_{tw_{t8}} = PD_{t8} = 4$). Consequently, $GL = \{tw_{t8}, tw_{t16}, rg_{t1}, rg_{t4}\}$ and $BTL = \{t_9, t_{10}\}$.

Step 9. Virtually the same happens during this step because the power dissipation of $ct = t_9$ is $PD_{t9} = 12$, which is already equal to PD_{max} so that $ct = t_9$ could not be power dissipation compatible with any of the existing gaps: $tw_{t8}, tw_{t16}, rg_{t1}, rg_{t4}$. Therefore, a new test session t_9 is generated together with its twin gap tw_{t9} . Consequently, $GL = \{tw_{t9}, tw_{t8}, tw_{t16}, rg_{t1}, rg_{t4}\}$ and $BTL = \{t_{10}\}$.

Step 10. During the last step $ct = t_{10}$ is scheduled in gap $cg = tw_{t8}$, because it is not compatible with $cg = tw_{t9}$ for the same power dissipation reasons. A *RestGap* $rg = rg_{t8}$ is generated having the following parameters: $T_{rg_{t8}} = T_{t8} - T_{t10} = 1$, $PD_{rg_{t8}} = PD_{t8} = 4$ and $Comp.List_{rg_{t8}} = Comp.List_{t8} = \{t_1, t_2, t_4, t_6, t_7, t_9, t_{10}\}$. Since all tests in the compatibility list have already been scheduled it would be pointless to insert this *RestGap* into the *GapsList*. The *TwinGap* $tw_{t8,10}$ has the following parameters: $T_{tw_{t8,10}} = T_{tw_{t10}} = 1$, $PD_{tw_{t8,10}} = PD_{tw_{t8}} + PD_{t10} = 11$ and $Comp.List_{tw_{t8,10}} = Comp.List_{tw_{t8}} \cap Comp.List_{t10} = \{t_9\}$.

The test schedule obtained with the ECT approach as depicted in Figure 4.5 can be improved with our merging approach to arrive at a more compact schedule as shown in Figure 4.8.

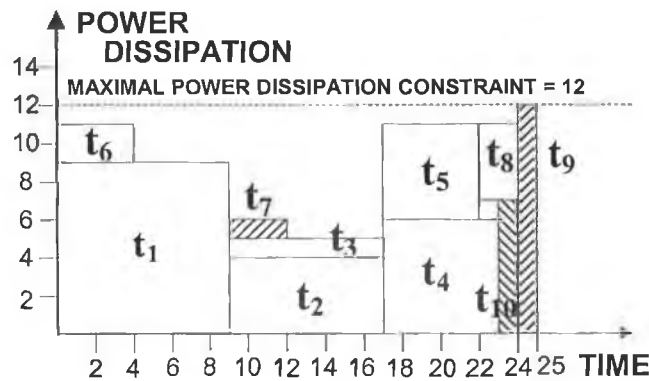


Figure 4.8: Merging result followed by EMC approach to 10 tests.

Our proposed approach is described below:

According to the definition of Extended Main Cliques (*EMC*), for this example, there are five *EMCs*, which corresponds to the five root test sessions in the schedule obtained with the ECT approach.

The five *EMCs* and the block tests in each *EMC* are listed below:

$$EMC_1 = \{t_1, t_6\}$$

$$EMC_2 = \{t_2, t_3, t_7\}$$

$$EMC_3 = \{t_4, t_5\}$$

$$EMC_4 = \{t_8, t_{10}\}$$

$$EMC_5 = \{t_9\}$$

Next, the steps of merging will be given:

The constraint of merging is determined by three factors. The first one is the compatibility between block-tests, especially the block-tests that represent the length of each *EMC* respectively. The second one is the power dissipation constraint criterion, which ensures the power dissipation of merged pair of *EMCs* not to exceed the power dissipation limit. The third one, the determination of merged pairs of *EMCs* with trade-off between the efficiency and computational cost.

The merging procedure (and then scheduling procedure) can be realized by executing the following steps, see Figure 4.9.

Step 11. Check the possibility of merging *EMC*₁ with other *EMCs*, in this case, they are *EMC*₂, *EMC*₃, *EMC*₄ and *EMC*₅. For *EMC*₁, the block-test, which determines the length of *EMC*₁, is test *t*₁, the power dissipation of test *t*₁ is 9 power units, because the power dissipation limit is 12, so there are 12-9=3 power units remaining in the first level gap *FLG*_{*EMC*₁}. This is a maximal power dissipation value that other block-test in other *EMCs* can be merged. Since

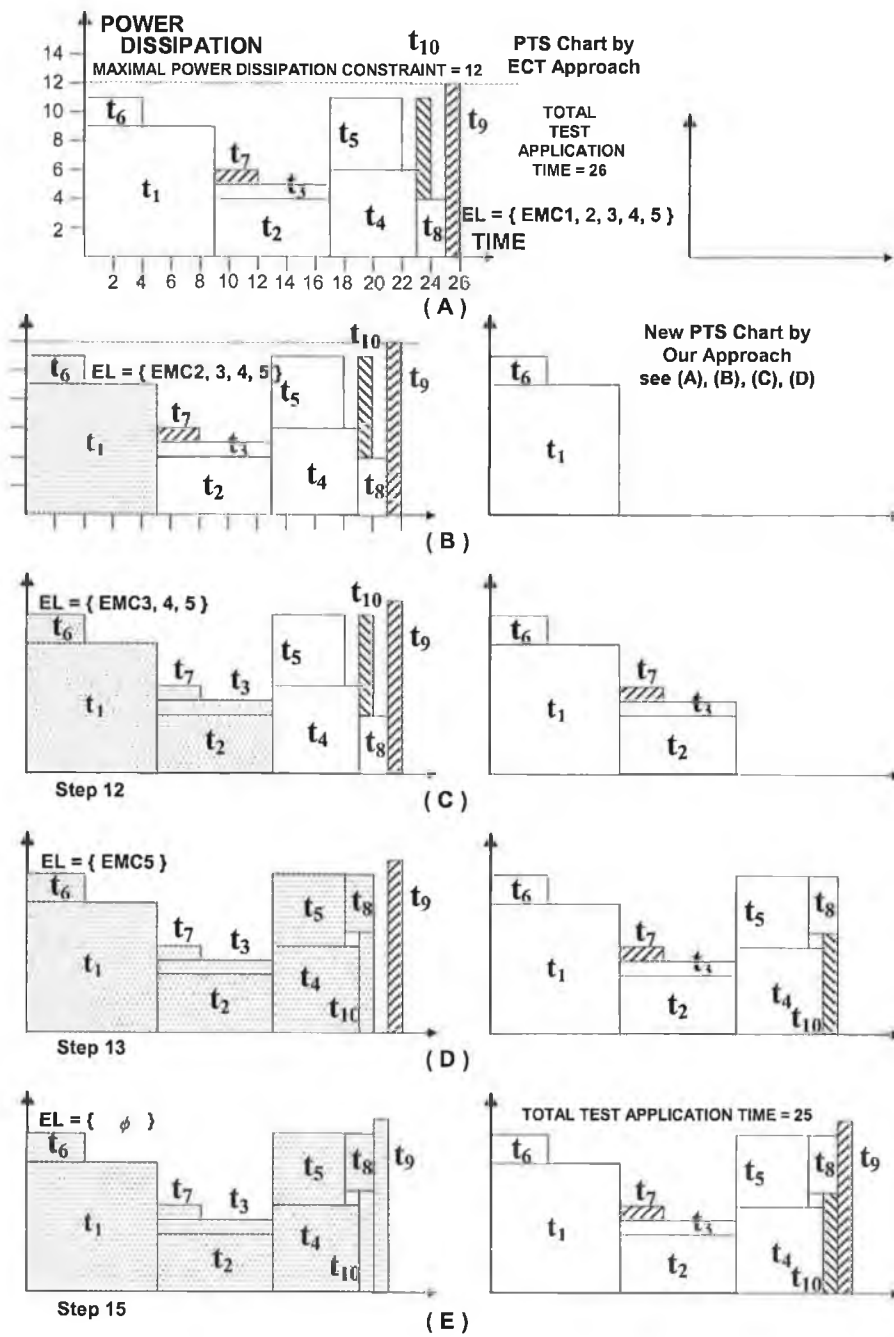


Figure 4.9: Merging Steps of the Example.

the power dissipation values of tests t_2, t_4, t_8, t_9 , which are the tests that determine the length of $EMC_2, 3, 4, 5$ respectively, are 4, 6, 4, 12 respectively, they are all bigger than the power dissipation value 3. So, a conclusion can be drawn that there is no possibility to merge EMC_1 with any of EMC_2, EMC_3, EMC_4 and EMC_5 respectively. The EMC_1 is drawn up from PTS chart produced by tree growing technique and is put into the new PTS chart produced by the merging approach. After the scheduling step 11, the $EMClist = \{EMC_1, EMC_2, EMC_3, EMC_4, \text{ and } EMC_5\}$. $EMC(1)$ is made strikethrough to represent that it is "merged". See Figure 4.9(B).

Step 12. For EMC_2 , according to the power dissipation analyzing method, there are $12 - (4 + 1) = 7$ ($P_{max} - P_{t2} - P_{t3}$) power units remaining in FLG_{EMC2} . Only EMC_3 and EMC_4 satisfy the power dissipation constraint condition. Then, check the compatible relation of EMC_2 with EMC_3 and EMC_4 respectively. For EMC_2 , test length of test t_2 and t_3 are the same, 8 time units. Because the block-test t_4 in EMC_3 is not compatible with block-tests t_2 and test t_3 in EMC_2 , so, there is no possibility of merging EMC_2 with EMC_3 . Due to the same reason, there is no possibility of merging EMC_2 with EMC_4 . So, the EMC_2 in the old PTS chart is put into the new PTS chart. After the scheduling step 12, the $EMClist = \{EMC_1, EMC_2, EMC_3, EMC_4, \text{ and } EMC_5\}$. See Figure 4.9(C).

Step 13. Attempt to merge EMC_3 with EMC_4 and EMC_5 . According to the power dissipation analyzing method, it is known that the power dissipation value of test t_4 in EMC_3 is 6 power units (t_5 is 5). There is $P_{max} - P_{t4} = 12 - 6 = 6$ power units remaining for merging with other block tests. The power dissipation value of test t_8 in EMC_4 is 4 power units, so it is test t_8 in EMC_4 that satisfy its power dissipation constraint. Check the compatibility list in Figure 4.6, it is found that test t_4 is compatible with test t_8 . So, EMC_3 can be merged with EMC_4 . The actual operation of merging is as follows:

Since the length of first level gap in EMC_3 is $FLG_{EMC_3} = T_4 - T_5 = 6-5=1$ time unit. So the 1 time unit is actually the overlap time of test t_4 in EMC_3 and test t_8 in EMC_4 . The actual scheduling of EMC_3 and EMC_4 after merging is shown in Figure 4.9 (D).

It can be seen clearly that the reduction of total test application time can be realized by the above step. Before merging, the total test application of EMC_3 and EMC_4 is $T_4 + T_8 = 6 + 2 = 8$ time units. After merging, it is length ($EMC_3 + EMC_4$ - overlap time) = (length $_{test\ 14}$ + length $_{test\ 19}$ - overlap time of 1) = $6+2-1=7$ time units. Now, EMC_3 and EMC_4 in the old PTS chart can be put into the new PTS chart. After this step, only EMC_5 is left in $EMClist = \{\cancel{EMC_1}, \cancel{EMC_2}, \cancel{EMC_3}, \cancel{EMC_4}, EMC_5\}$.

Step 14. When the outer FOR Loop comes to the fourth iteration, it finds that $EMC(4)$ is already marked "merged", so it goes directly to the next iteration. Because $EMC(4)$ is the second last EMC in the $EMClist$, the outer FOR Loop exits.

Step 15. The final IF statement checks to see if the last EMC (EMC_5 in this case) has been merged or not. If has been merged already in previous steps the program finishes; Otherwise, before the program finishes, EMC_5 is marked "merged" in $EMClist$ and a copy of it is put into the $MergedEMClist$.

When the merging procedures finish, all $EMCs$ in the $EMClist$ should have been marked "merged" and a $MergedEMClist$ is generated. See Figure 4.9(E).

Thus, after the above scheduling steps, a final schedule with 25 time units is obtained as shown in Figure 4.8.

4.4 Conclusions

In this chapter, a comparison of existing test scheduling techniques is given. On the basis of this, through the analysis of the advantages and disadvantages of extended compatibility tree technique, our merging approach is proposed, including the description of operating procedures, algorithm pseudocode, the analysis of algorithm complexity, and a test schedule example.

Chapter 5

Experimental Results

Several experiments using academic benchmarks and industrial designs were carried out. The results obtained using our approach is compared with those obtained with other approaches and with known optimal solutions. If no optimal solution is known, results obtained with Erik's approach [Lar00c] that uses a Simulated Annealing (SA) implementation is compared with our results.

All experiments, where the computational costs are stated, are performed on a PC with a 450 MHz processor and 32 Mbytes RAM.

C programming language is chosen to implement the algorithm proposed in this thesis.

In Section 5.1, 5.3, 5.5 and 5.6 we report the results from experiments where test application time is minimized while considering test conflicts and test power constraints. In Section 5.2 and Section 5.4 we perform experiments where the test application time is minimized considering test conflicts only.

5.1 Experiments On Muresan's Design

The Design by Muresan *et al.* contains test conflicts and power constraint, full description of this experiment is given in Appendix A.1 [Mur00a]. The total test application time using the Muresan approach is 26 time units, and the test schedule obtained is shown in Figure 5.1.

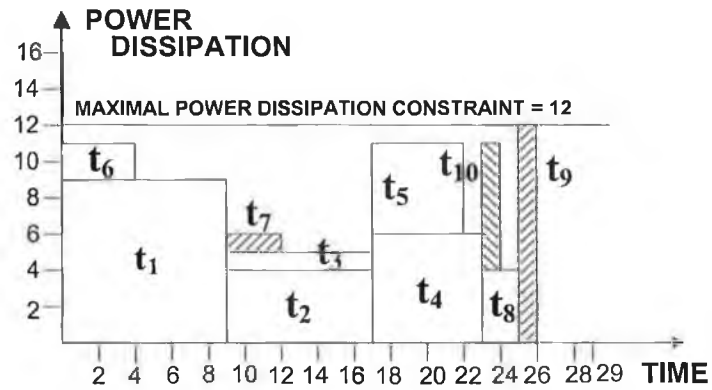


Figure 5.1: Test schedule produced with Muresan's approach.

The schedule obtained with Erik's SA optimization approach on Muresan's design one is shown in Figure 5.2.

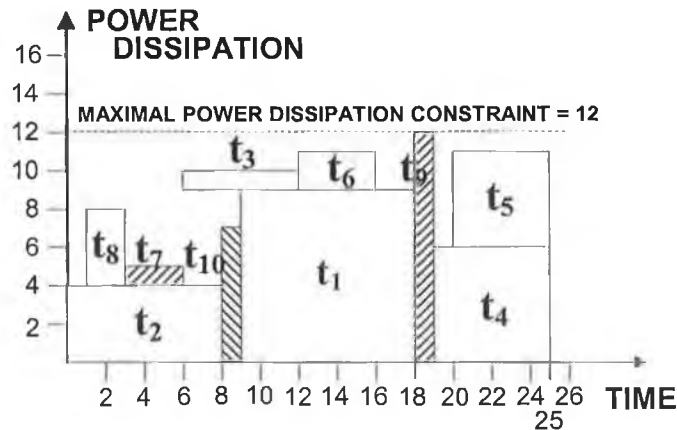
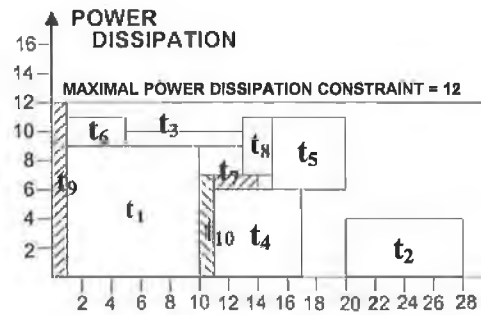


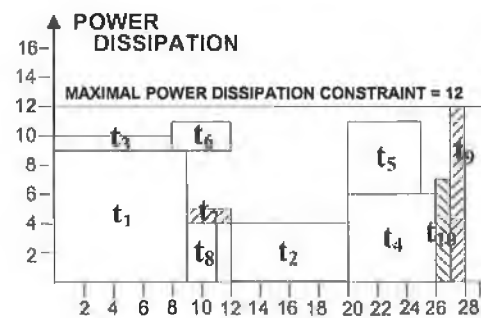
Figure 5.2: Test schedule produced by Erik's Simulated Annealing implementation on Design One by Muresan.

As can be seen from the schedule, the total test application time is 25 time units.

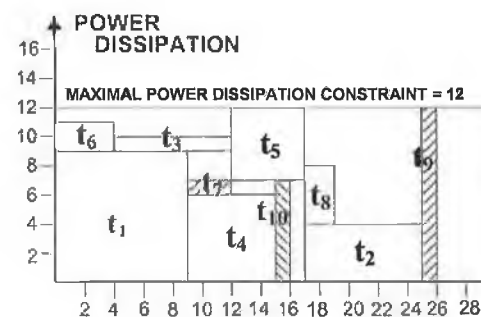
The schedule obtained using Erik's approach with initial sorting of tests based on *power*, *time* and *power x time* are shown in Figure 5.3(a), (b), (c) respectively.



(a) power sort



(b) time sort



(c) power x time sort

Figure 5.3: Test schedule using Erik's approach with initial sorting based on (a) power, (b) time and (c) power x time in Muresan's Design.

The total test application time required for the test schedules obtained with initial sort on either power or time is 28 time units. A better schedule that needs 26 time units can be obtained with initial power x time sorting.

Using our merging approach, the total test application time is 25 time units. See Figure 5.4.

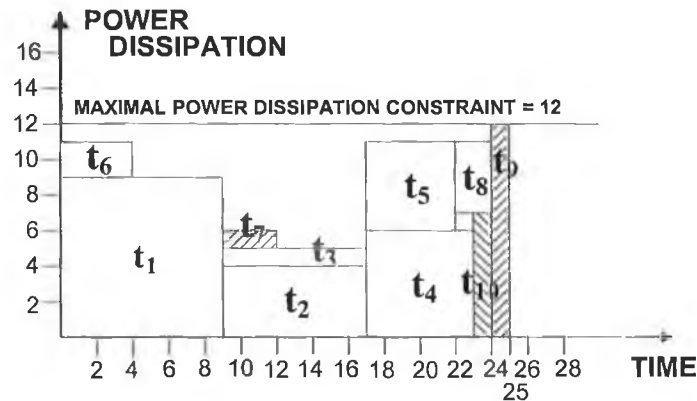


Figure 5.4: Test schedule using merging approach on Muresan's Design.

Our approach achieved the same results as Erik's SA optimization approach, with much less computational requirements. It is also interesting to compare Figures 5.2 and Figure 5.4; the schedules are different but both achieved the known shortest test application time.

Results of all experiments on Muresan's design are summarized in Table 5.1. Our merging approach achieved better results compared to Muresan's solution.

Approach	Test time	Difference to SA
Muresan <i>et al</i>	26	4%
Erik's heuristic, (power sort)	28	12%
Erik's heuristic, (time sort)	28	12%

Erik's heuristic, (power x time sort)	26	4%
Erik's Simulated annealing	25	-
Our approach	25	0%

Table 5.1: Experimental results on Muresan's Design.

5.2 Experiments on Design by Kime

The Design by Kime, described in Appendix A.2, has been used by Kime and Saluja [Kim82], Craig *et al.* [Cra88], Jone *et al.* [Jon89] and Garg *et al.* [Gar91]. The design contains test conflicts only, the test application time for the optimal solution is 318 time units [Lar00c]. Since no power consumption is given for the tests, we only performed the experiment using our approach with an initial sorting of the tests based on time. The solution from our approach is shown in Figure 5.5 and it was produced within one second.

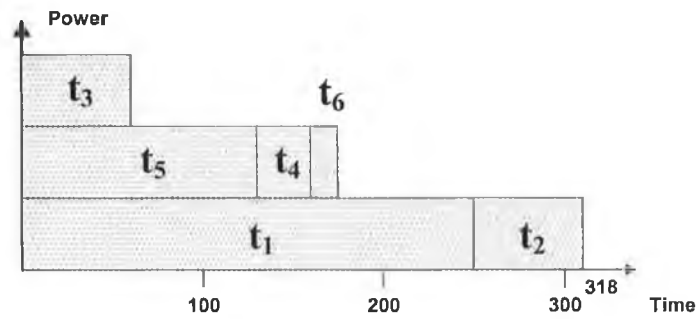


Figure 5.5: Test schedule using Merging Approach on Design by Kime (no power dissipation constraints).

All approaches but the one proposed by Kime and Saluja, can find the optimal solution. Test application times required for schedules obtained with different approaches are listed in Table 5.2.

Approach	Test time
Optimal	318
Kime and Saluja	349
Craig <i>et al.</i>	318
Jone <i>et al.</i>	318
Grag <i>et al.</i>	318
Erik's heuristic (time sort)	318
Our approach	318

Table 5.2: Experimental results on Design by Kime.

5.3 Experiments on ASIC Z Design One

With the ASIC Z Design One, we compare our test scheduling technique with the approaches proposed by Zorian [Zor93] and Chou *et al.* [Cho97]. See Appendix A.3. The assumptions for the experiments are the same as Chou *et al.* [Cho97], namely:

- Maximal power dissipation is limited to 900Mw,
- all tests can be applied concurrently,
- the power consumption for idle blocks are excluded.

The test schedules generated by the approaches proposed by Zorian and Chou *et al.* are presented in Figure 5.6 and Figure 5.7 respectively.

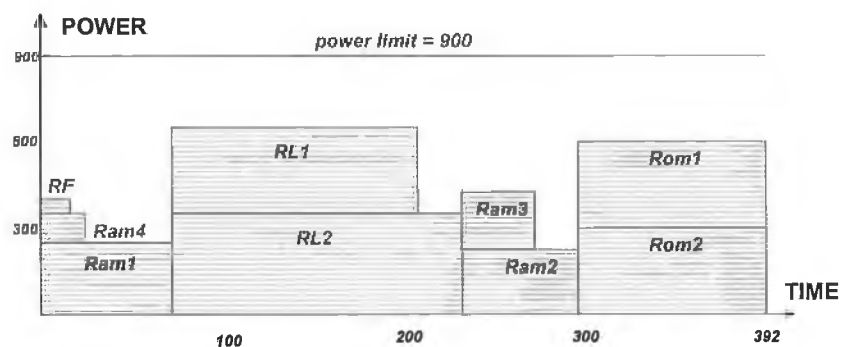


Figure 5.6: Test schedules generated using Zorian's approach.

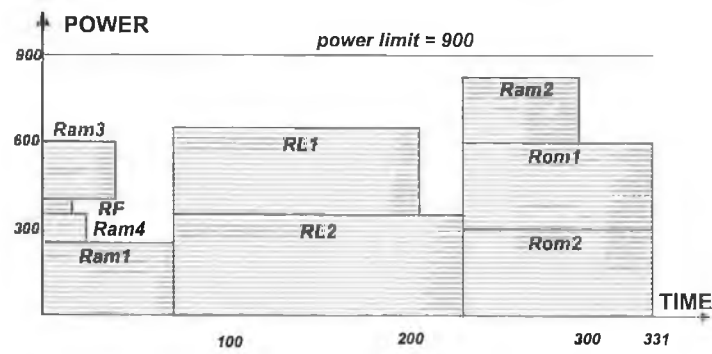


Figure 5.7: Test schedules generated using Chou's approach.

The test schedule achieved by Erik's heuristic is shown in Figure 5.8.

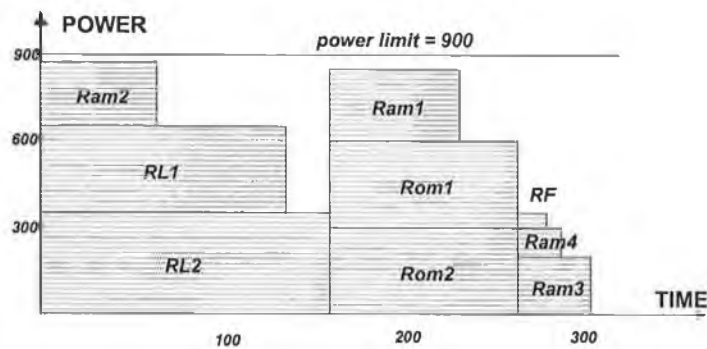


Figure 5.8: Test schedules generated using Erik's approach.

Using Erik's approach, the total test application time is 300 in all cases of initial sorting. The approach proposed by Zorian results in a solution with four test sessions and total test application time of 392. The approach proposed by Chou *et al.* results in a solution with three test sessions and total test time of 331. The approach proposed by us results in a solution with three-test sessions and total test application time of 300, see Figure 5.9.

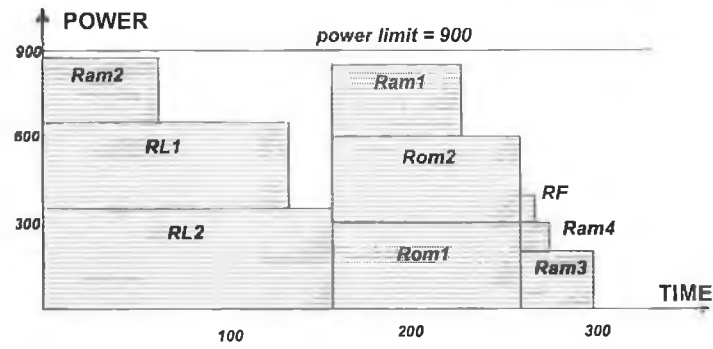


Figure 5.9: Test schedules achieved using merging approach.

All experimental results are summarized in Table 5.3. The optimal solution has a test application time of 300. The schedule created by the approach proposed by Zorian needs 30.7% more test time compared with the optimal schedule. The schedule created by the approach proposed by Chou *et al.* needs 10.3% more test time than the optimum schedule. Our approach finds the optimal test schedule within a second.

Approach	Test time
Optimum	300
Zorian	392
Chou <i>et al.</i>	331
Erik's heuristic (time sort)	300
Erik's heuristic (power sort)	300
Erik's heuristic (power x time sort)	300
Our approach	300

Table 5.3: A comparison of different approaches on ASIC Z Design One.

5.4 Experiments on System L

The System L is an industrial design, see Appendix A.4, where no data is available for test D, G and F, they are therefore excluded from the experiments. The designer schedules the 15 tests with a test application time of 1592 time units as shown in Figure 5.10. Erik's approach with an initial sorting based on power is shown in Figure 5.11 and the test application time is 1077 time units.

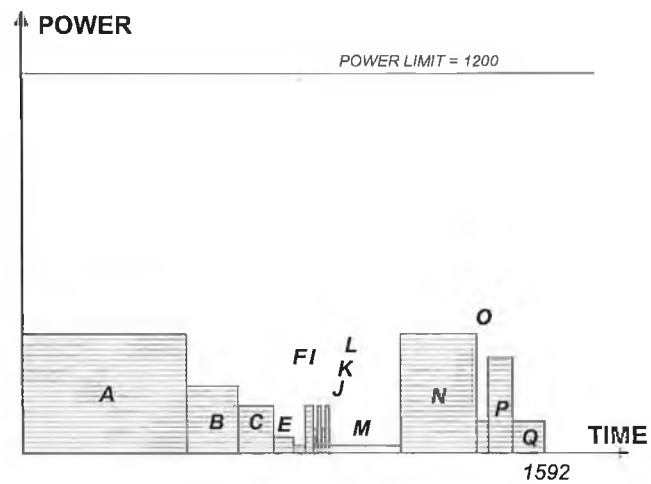


Figure 5.10: Designer's test schedule on System L.

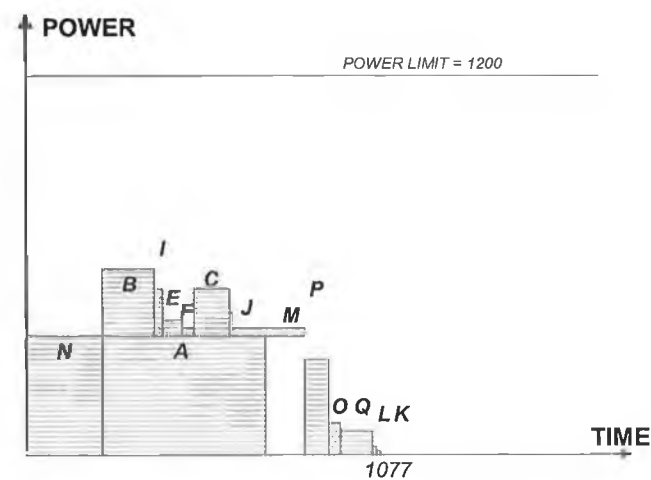


Figure 5.11: Test schedules achieved using Erik's heuristic with sorting based on power on System L.

The schedule obtained with our approach is shown in Figure 5.12 and the test application time is 1077 time units.

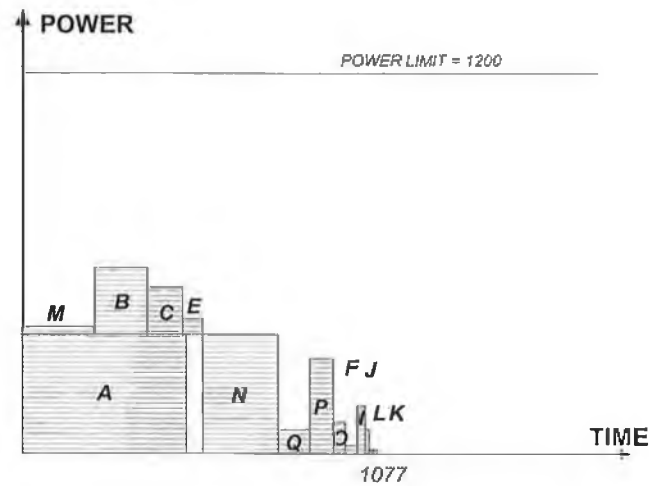


Figure 5.12: Test schedules achieved using Merging Approach on System L.

Experimental results on System L are summarized in Table 5.4.

Approach	Time
Designer's test schedule	1592
Erik's approach	1077
Merging approach	1077

Table 5.4: Experimental results on System L.

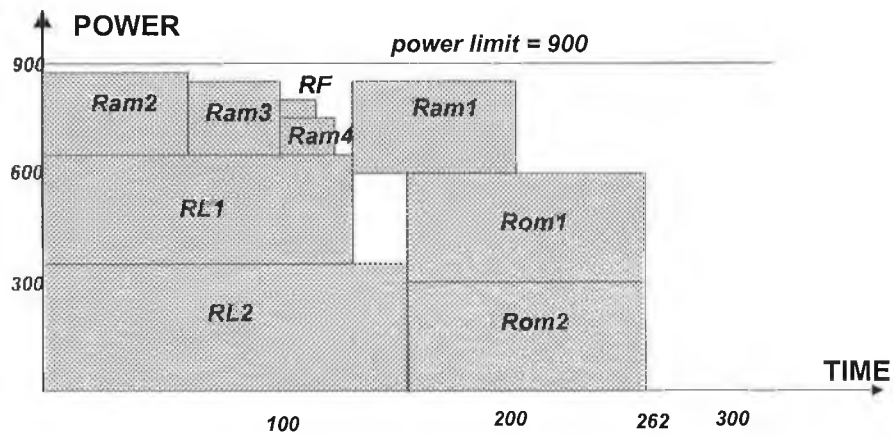
Our approach finds the better schedule, which needs 32% less test time than the schedule produced by the designer. The time required to produce the schedule using our approach was less than one second.

5.5 Experiments on ASIC Z Design Two

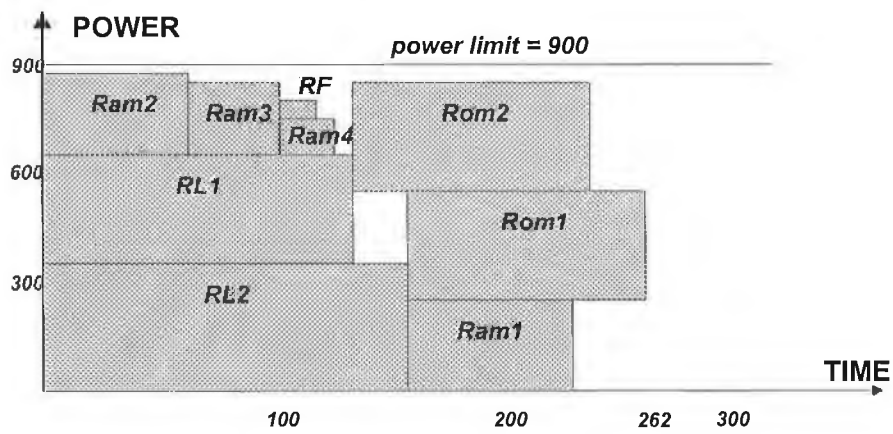
We performed experiments on the ASIC Z Design Two. See Appendix A.5, with the following assumptions:

- Maximal power dissipation is limited to 900mW,
- all tests can be applied concurrently,
- the power consumption for idle blocks are not considered, and
- new tests are allowed to start before all tests in the previous test session are completed. A test is allowed to start even if other tests in the previous test session are not yet completed.

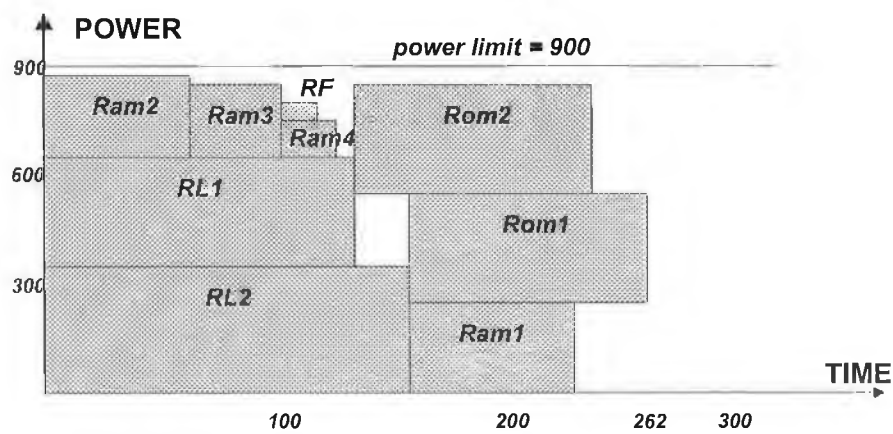
The test schedules using Erik's approach with the initial sorting on *power*, *time* and *power x time* is shown in Figure 5.13 (a), (b) and (c) respectively.



(a) power sort



(b) time sort



(c) power and time sort

Figure 5.13: Test schedules achieved using Erik's heuristic on ASIC Z Design Two using different initial sortings.

The test schedule using our approach is shown in Figure 5.14.

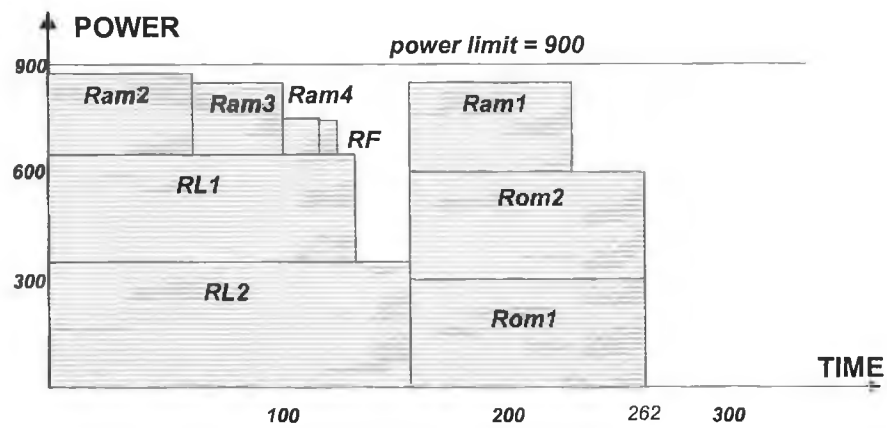


Figure 5.14: Test schedule achieved using merging approach on ASIC Z Design Two.

The experimental results are summarized in Table 5.5, as can be noticed, all approaches result in the same test application time of 262.

Approach	Idle power considered	Test time
Erik's Simulated annealing	No	262
Erik's heuristic (power sort)	No	262
Erik's heuristic (time sort)	No	262
Erik's heuristic (power x time sort)	No	262
Merging approach	No	262

Table 5.5: Experimentl results on ASIC Z Design Two.

5.6 Experiments on Muresan's Design Two

The Design Two by Muresan *et al.* contains test conflicts and power constraints. See Appendix A.6 [Mur00a]. The total test application time using the approach by Muresan *et al.* is 49 time units. See the test schedule in Figure 5.15.

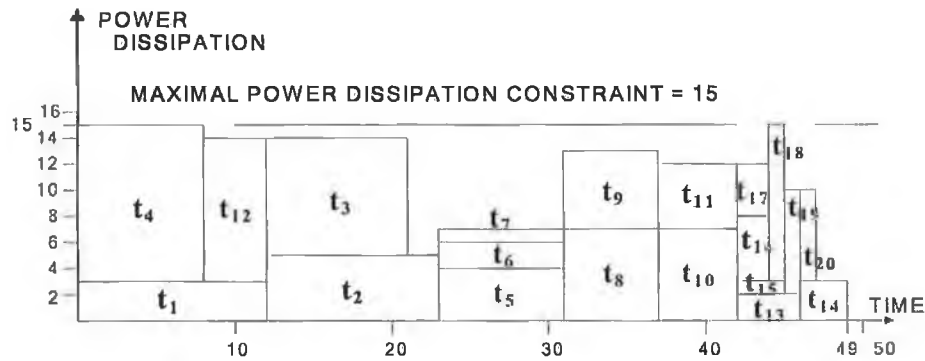


Figure 5.15: Test schedule produced by Muresan *et al.* on Muresan's Design Two.

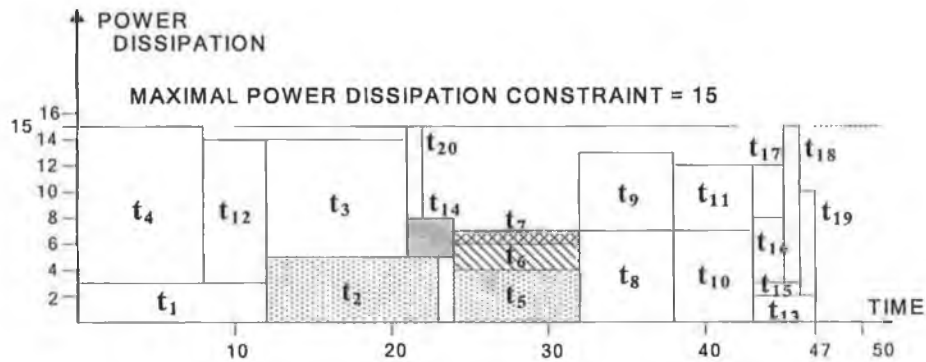


Figure 5.16: Test schedule using merging approach on Muresan's Design Two.

The total test application time using merging approach is 47 time units. See the test schedule in Figure 5.16.

The experimental results are summarized in Table 5.6. Our approach produces a better schedule that needs less time than Muresan's approach.

Approach	Test time
Muresan <i>et al.</i>	49
Merging approach	47

Table 5.6: Experimental results on by Muresan's Design Two.

For all the experiments on the five different designs, our approach has always achieved either better or same good schedules as other approaches.

Chapter 6

Conclusions and Future Work

6.1 Thesis Summary

The aim of the work presented in this thesis is to develop useful methods to give a designer an early feeling for the test problems and guidance in the search for an efficient test solution. The methods are developed mainly at the system level since we believe that it is important for a designer to have an overall perspective of the system and its test problems as early as possible. The proposed technique minimizes the test application time while considering several other issues and constraints.

A SOC consists of several cores where each core may consist of several blocks. A sequential testing of such a system leads to an unacceptably long test time. Several tests must be applied concurrently. However, concurrent testing can lead to high-test power consumption, which may damage the system. Furthermore, several constraints limit concurrent testing. In this thesis, a methodology for the testing of SOC has been developed. The methodology minimizes test application time, while considering test conflicts and test power consumption. The methodology considers both test scheduling and test parallelism, so that the test application time is reduced.

We have performed several experiments on academic benchmarks and on industrial designs and we have compared our approach with several other approaches. We have demonstrated that the proposed technique is useful and efficient for large industrial designs.

This thesis proposes a polynomial-time solution to the NP-Complete Power-Constrained Block-Test Scheduling (PTS) problem stated in [Cho97]. It is a practical approach proposed as a solution to the aforementioned problem. It is based on the classical tree growing technique, especially, the Extended Compatibility Tree technique [Mur00a, b].

This work focused only on the high-level PTS Problem. The proposed algorithm is part of a system-level block-test approach, which is applied on a modular view of a test hierarchy. The modular elements of this hierarchy could be: subsystems, boards, Multi-chip Modules MCMs, ICs (dies), macro-blocks and Register Transfer Level (RTL) blocks.

The algorithm given in the thesis deals with tests for blocks of logic, which do not have equal test length. Thus, they are unequal-length block-test scheduling algorithms. In these algorithms, the test order within the test sets of various modules in a circuit is not considered important.

For simplicity, a constant additive model is employed for power dissipation analysis and estimation throughout the approach.

This algorithm can get better block-test scheduling result, with little extra computational cost, compared with the extended tree growing technique.

The PTS algorithm proposed in this thesis uses greedy heuristics that can produce better test schedules in polynomial time. This is very important to the rapid system prototyping of today's VLSI/SOC designs. Though our algorithm cannot guarantee the optimal solutions, it still can be viewed as an efficient and not time-consuming practical approach.

6.2 Contributions

This thesis brings certain contribution to the solution to the NP-Complete problem of power-constrained block-test scheduling in [Cho97]. Our approach is based on tree growing technique and it has a polynomial complexity. A test schedule block-tests can be quickly generated with this efficient approach.

The achievement of this thesis can be the basis of future research work towards finding more efficient and less computational cost solutions to other scheduling problems in the field of system-level low-power testing design.

6.3 Future Work

Test Order Requirements

In the current algorithm, the order of testing blocks of logic is not considered important. The algorithm could be improved to take test order of the blocks as another constraint. For example, to allow the user to specify that test t_i should be carried out before test t_j ; because test t_i is more likely to find more common faults that test t_j does. In situations where a single fault means the whole system should be discarded, to test the blocks more likely to have faults first can potentially save test time, because once a fault is found in a block (and generally internal faults are not repairable in VLSI/SOC). The other blocks need not be tested at all (which save test time), as the whole system will be discarded.

Higher Level Merging

Only first level gaps of EMCs are considered for merging in this thesis. Future work could consider more level gaps in EMCs for possible further merging.

Test Scheduling and Test Access Mechanism (TAM)

In the future, we could consider test access mechanism together with test scheduling.

References

- [Abr90] Miron Abramovici, Melvin A. Breuer, and Arthur D. Friedman. Digital Systems Testing and Testable Design. *IEEE Press*, ISBN 0-7803-1062-4, 1990.
- [Abr94] Abramovici, M. A. Breuer and A. D. Friedman. Digital System Testing and Testable Design. *IEEE Press*, 1994.
- [Agr93a] V. D. Agrawal, C. R. Kime, and K. K. Saluja. A tutorial on built-in self test - part 1: Principles. *IEEE Design and Test of Computers*, 10(1): 73-82, March 1993.
- [Agr93b] V. D. Agrawal, C. R. Kime, and K. K. Saluja. A tutorial on built-in self test - part 2: Applications. *IEEE Design and Test of Computers*, 10(1): 69-77, June 1993.
- [Agr94] V. D. Agrawal, C. J. Lin, P. W. Rutkowski, S. Wu, and Y. Zorian. Built-In Self-Test for Digital Integrated Circuits. *AT&T Technical Journal*, page 30-39, March 1994.
- [Agr95] V. D. Agrawal. Editorial-Special Issue on Partial Scan Design. *Journal of Electronic Testing: Theory and Application (JETTA)*, 7(5): 5-6, August 1995.
- [Ait99] R. C. Aitken. Nanometer Technology Effects on Fault Models for IC Testing. *Computer*, 32(11): 46-51, November 1999.
- [Ali94] M. Alidina, J. Monteiro, S. Devadas. A. Ghost and M. Papaefthymiou. Pre-computation-Based Sequential Logic Optimization for

Low Power. *IEEE Transactions on Very Scale Integration (VLSI) Systems*, 2(4): 426-436, December 1994.

- [AMS] AMS. 0.35 Micro CMOS Process Parameters. *Austria Micro System International AG*, 1998.
- [And97] T. L. Anderson. Thoughts on Core Integration and Test. In *Proceedings of the International Test Conference*, ITC'97 pages 1039-1045, 1997.
- [Avr91] L. Avra. Allocation and Assignment in High-Level Synthesis for Self-Testable Data Paths. In *Proceedings of the 1991 International Test Conference*, pages 463-472, Oct 1991.
- [Avr93] L. Avra and E. J. McCluskey. Synthesizing for Scan Dependence in Built Self-Testable Designs. In *Proceedings the International Conference of Computer-Aided Design*, pages 30-35, 1993.
- [Bas92] A. Basu, T. C. Wilson, D. K. Banerji and J. C. Majithia. An Approach to Minimize Testability Overhead for BILBO Based Built-In Self-Test. In *Proceedings of the IEEE VLSI Test Symposium*, pages 55-59, 1992.
- [Bat85] John Bateson. In-Circuit Testing. *Van Nostrand Reinhold Company*, New York, 1985.
- [Ben97] B. Bennetts. A Design Strategy for System-on-a Chip Testing. *Electronic Products*, pages 57-59, Jun 1997.
- [Cha94a] S. Chakrabarty and V. P. Dabholkar. Minimizing Power Dissipation in Scan Circuits During Test Application. In *Technical Report No. 94-06*, Dept. of Computer Science, SUNY at Buffalo, 1994.
- [Cha94b] S. Chakrabarty and V. P. Dabholkar. Minimizing Power Dissipation in Scan Circuits During Test Application. In *Proceedings of IEEE International Workshop on Low Power Design*, pages 51-56, 199.
- [Cha94c] S. Chakrabarty and V. P. Dabholkar. Two Techniques for Minimizing Power Dissipation in Scan Circuits During Test

Application. In *Proceedings of the 3rd Asian Test Symposium*, pages 324-329, 1994.

- [Cha95a] A. P. Chandrakasan, M. Potkonjak, R. Mehra, J. Rabaey and R. W. Brodersen. Optimizing Power Using Transformations. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 14, no. 1, January 1995.
- [Cha95b] A. P. Chandrakasan, R. W. Brodersen. Low Power Digital CMOS Design. *Kluwer Academic Publishers*, 1995.
- [Cha01a] Krishnendu Chakrabarty. Test Scheduling for Core-Based Systems Using Mixed-Integer Linear Programming. *Transaction CAD of IC and System*, Vol. 19, No. 10, pp. 1163-1174, October 2001.
- [Cha01b] A. Chandra and Krishnendu Chakrabarty. System-on-a-Chip Test Data Compression and Decompression Architectures Based on Golomb Codes. *IEEE Transactions on Computer-Aided Design*, 20: 113-120, March 2001.
- [Cho94] R. M. Chou, K. K. Saluja and V. D. Agrawal. Power Constraint Scheduling of Tests. In *Proceedings of the 7th International Conference on VLSI Design*, pages 271-274, Jan 1994.
- [Cho97] R. M. Chou, K. K. Saluja and V. D. Agrawal. Scheduling Tests for VLSI Systems under Power Constraints. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 5(2): 175-185, Jun 1997.
- [Cra88] G. L. Craig, C. R. Kime and K. K. Saluja. Test Scheduling and Control for VLSI Built-in Self-Test. *IEEE Transactions on Computers*, Vol. 37, No.9, pp. 1099-1109, September 1988.
- [Dab94] V. P. Dabholkar and S. Chakrabarty. Minimizing Power Dissipation in Combinatorial Circuits During Test Application. In *Technical Report No. 94-100*, Dept. of Computer Science, SUNY at Buffalo, 1994.
- [Dab98] V. P. Dabholkar and S. Chakrabarty, I. Pomeranz and S. Reddy. Techniques for Minimizing Power Dissipation in Scan and Combination Circuits During Test Application. *IEEE Transactions on Computers*, 17(12): 1325-1333, Dec 1998.

- [Dey94] S. Dey and M. Potkonjak. Non-Scan Design-for-Testability of RTL Data Paths. In *Proceedings of the International Conference of Computer-Aided Design*, pages 640-645, 1994.
- [Dim91] A. A. Diman. An Algorithm for Minimizing the Number of Test Cycles. In *Proceedings of the 4th IEEE International Symposium on VLSI Design*, pages 154-156, 1991.
- [Ele98] Petru Eles, Krzysztof Kuchcinski and Zebo Peg. System Synthesis with VHDL. *Kluwer Academic Publishers*, 1998.
- [Fen91] S. Feng and Y. K. Malaiya. Optimization of Test Parallelism with Limited Hardware Overhead. *Microelectronic Reliability*, 31(2/3): 271-276, 1991.
- [Gaj92] Daniel Gajski, Nikil Dutt, Allen Wu and Steven Lin. High-Level Synthesis: Introduction to Chip and System Design, *Kluwer Academic Publisher*, ISBN 0-7923-9194-2, 1992.
- [Gar91] M. Garg, A. Basu, T. C. Wilson, D. K. Banerji and J. C. Majithia. A New Test Scheduling Algorithm for VLSI Systems, In *Proceedings of the Symposium on VLSI Design*, pp, 148-153, January 1991.
- [Gir99a] P. Girard, C. Guiller, C. Landrault, S. Pravossoudovitch. A Test Vector Inhibiting Technique for Low Energy BIST Design. In *Proceedings of the VLSI Test Symposium*, pages 407-412, 1999.
- [Glo86] Fred Glover. Future Paths for Integer Programming and Links to Artificial Intelligence, *Computer and Ops. Res.*, 5, pp. 533-549, 1986.
- [Gu96] X. Gu. TT Level Testability Improvement by Testability Analysis and Transformations. *PhD Thesis, Linköping University*, 1996.
- [Gup91] R. Gupta and M. A. Breuer. Ordering Storage Elements in a Single Scan Chain. In *Proceedings of the International Conference on Computer-Aided Design*, pages 408-411, 1991.
- [Hal96] J. Hallberg. High-Level Synthesis under Local Timing Constraints. *Master's Thesis, Linköping University*, 1996.

- [Har93] H. Harmanani and C. Papachristou. An Improved Method for RTL Synthesis with Testability Tradeoffs. In *Proceedings of the International Conference on Computer-Aided Design*, pages 30-35, 1993.
- [Hol75] Holland. J. H. Adoption in National and Artificial System. *University of Michigan Press*, Ann Arbor. 1975.
- [Her98] A. Hertwig and H. J. Wunderlich. Low Power Serial Built-in Self-Test. In *Proceedings of the IEEE European Test Workshop*, pages 49-53, 1998.
- [Ish98] M. Ishida, D. S. Ha and T. Yamaguchi. Compact: A Hybrid Method for Compressing Test Data. In *Proceedings of IEEE VLSI Symposium*, 62-69, April 1998.
- [Jas02] Abhijit Jas, Nura A. Touba. Deterministic Test Vector Compression/Decompression for System-on-a-Chip Using an Embedded Processor. *Journal of Electronic Testing : Theory and Applications* 18, 503-514, 2002.
- [Jon89] W. B. Jone, C. Papachriston and M. Pereira. A Scheme for Overlaying Concurrent Testing of VLSI Circuits. In *Proceedings of Design Automation Conference*, pages 531-536, Jun 1989.
- [Kim82] C. R. Kime and K. K. Saluja. Test Scheduling in Testable VLSI Circuits. In *Proceedings of the International Symposium of Fault-Tolerant Computers*, pages 406-412, Jun 1982.
- [Kim88] K. Kim, J. G. Trout and D. D. Ha. Automatic Insertion of BIST Hardware using VHDL. In *Proceedings of the 25th IEEE Design Automation Conference*, pages 9-15, 1988.
- [Kir83] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi. Optimization by Simulated Annealing, *Science*, Vol. 220, No. 4598, pp. 671-680, 1983.
- [Kum95] N. Kumar, S. Katkooli, L. Rader and R. Vemuri. Profile-Driven Behavioral Synthesis for Low-Power VLSI Systems. *IEEE Design and Test of Computers*, Fall 1995.

- [Lai93] W. J. Lai, C. P. Kung and C. S. Lin. Test Time Reduction in Scan Designed Circuits. In *Proceedings of EuroAsic Conference*, pages 489-493, 1993.
- [Lar00a] Erik Larsson and Zebo Peng. System-on-Chip Test Bus Design and Test Scheduling. *International Test Synthesis Workshop*, Santa Barbara, March, 2000.
- [Lar00b] Erik Larsson and Zebo Peng. Test Infrastructure Design and Test Scheduling Optimization. *Informal Digest of the European Test Workshop*, Cascais, Portugal, May 2000.
- [Lar00c] Erik Larsson. An Integrated System-Level Design for Testability Methodology. *PhD Thesis, Linköping University*, Sweden, 2000.
- [Lar01a] Erik Larsson and Zebo Peng. System-on-Chip Test Parallelization under Power Constraints. In *Proceedings of IEEE European Test Workshop*, May 2001.
- [Lar01b] Erik Larsson G. Carlsson and Zebo Peng. The Design and Optimization of SOC Test Solutions. In *Proceedings of the International Conference on Computer-Aided Design*, pages 523-530, Nov. 2001.
- [Lar01c] Erik Larsson and Zebo Peng. Test Scheduling and Scan-Chain Division under Power Constraint. In *Proceedings of Asian Test Symposium (ATS)*, pp. 259-264, Nov. 2001.
- [Lee92] S. Y. Lee and K. K. Saluja. An Algorithm to Reduce Test Application Time IN Full Scan Design. In *Proceedings of the IEEE International Conference on Computer-Aided Design*, pages 17-20, 1992.
- [Mac97] Enrico Macii, Massoud Pedram and Fabio Somensi. High-Level Power Modeling, Estimation and Optimization. *Design Automation Conference 97*, Anaheim, California, 1997.
- [Man99] S. Manich, A. Gabarro, J. Figueras, P. Girard, C. Guiller, C. Landrault, S. Pravossoudovitch, P. Teixeira and M. Santos. Low Power BIST by Filtering Non-Detecting Vectors . In *Proceedings of the IEEE European Test Workshop*, pages 165-170, 1999.

- [Mar95] R. San Martin and J. P. Knight. Power-Profile: Optimizing ASICs Power Consumption at the Behavioral Level. In *Proceedings of the 32nd Design Automation Conference*, San Francisco, USA, June 1995.
- [Mar97] Erik Jan Marrinissen and Maurice Lousberg. Macro Test: A Liberal Test Approach for Embedded Reusable Cores. In *Digest of Papers of IEEE International Workshop on Testing Embedded Core-Based Systems*, pages 1.2-1-9, 1998.
- [Mar99a] Erik Jan Marrinissen and Maurice Lousbery. The Role of Test Protocols in Testing Embedded Core-Based Systems ICs. In *Proceedings of the IEEE European Test Workshop (ETW)*, May 1999, pages, 70-75.
- [Mar99b] Erik Jan Marrinissen and Yervant Zorian. Challenge in Testing Core-Based Systems ICs. *IEEE Communications Magazine*, pp. 104-109, June 1999.
- [Meh94] R. Mehra and J. Rabey. Behavior Level Power Estimation and Exploration. In *Proceedings of the 1994 International Workshop on Low-Power Design*, Napa Valley, CA, April 1994.
- [Moh02] Kartik Mohanram and Nur A. Touba. Input Ordering in Concurrent Checkers to Reduce Power Consumption. In *Proceedings of 17th IEEE International Symposium on Defect Tolerance in VLSI Systems (DFT'02)*, November 06-08, 2002, Vancouver, BC, Canada.
- [Mur00a] V. Muresan, X. Wang and M. Vladutiu. The Left Edge Algorithm and the Tree Growing Technique in Block-Test Scheduling under Power Constraints. In *Proceedings of 18th IEEE VLSI Test Symposium*, pages 417-422, April 30-May, 2000, Montreal, Canada.
- [Mur00b] V. Muresan, X. Wang. Power-Constrained Block-Test List Scheduling. In *Proceedings IEEE International Workshop on Rapid System Prototyping*, pp. 182-187, 21-23 June 2000, Paris, France.

- [Mur00c] V. Muresan, X. Wang. A Comparison of Classical Scheduling Approaches in Power-Constrained Block-Test Scheduling. In *Proceedings of the 31st International Test Conference*, pp. 882-891, October 3-5, 2000, Atlantic City, NJ, USA.
- [Mur01] V. Muresan, X. Wang. Mixed Classical Scheduling Algorithm and Tree Growing Technique in Block-Test Scheduling under Power Constraints. In *Proceedings of the Twelfth IEEE International Workshop on Rapid System Prototyping*, pp. 162-167, 25-27 June 2001, Monterey, CA.
- [Nar92] S. Narayanan, C. Njinda and M. A. Breuer. Optimal Sequencing of Scan Registers. In *Proceedings of the International Test Conference*, pages 293-302, 1992.
- [Nar93] S. Narayanan and M. A. Breuer. Reconfigurable Scan Chain: A Novel Approach to Reduce Test Application Time. In *Proceedings of the IEEE International Conference on Computer-Aided Design*, pages 710-715, 1993.
- [Nar95] S. Narayanan and M. A. Breuer. Reconfiguration Techniques for a Single Scan Chain. In *Proceedings of the IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 14(6): 750-765, 1995.
- [Pap91] C. Papachristou, S. Chiu and H. Harmanani. SYNTTEST: A Method for High-Level Synthesis with Self-Testability. In *Proceedings of the International Conference on Computer Design, ICD*, pages 45-62, 1991.
- [Ped96] M. Pedram. Power Minimization in IC Design: Principles and Applications. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 1(1): 3-56, January 1996.
- [Pra92] D. K. Pradhan and J. Saxena. A Design for Testability Scheme to Reduce Test Application Time in Full Scan. In *Proceedings of the IEEE VLSI Symposium*, pages 55-59, 1992.
- [Ree93] C. R. Reeves. Modern Heuristic Techniques for Combinatorial Problems. *Blackwell Scientific Publishers*, London, 1993.

- [Rey00] K. Rey and S. Prasad. Low-Power CMOS VLSI Circuit Design. *John Wiley & Sons*, 2000.
- [Sug98] Makoto Sugihara, Hiroshi Date and Hiroto Yasuura. A Test Methodology for Core-Based System LSIs. *IEICE Transactions on Fundamentals*, pp. 2640-2645, Vol. E81-A, No. 12, December 1998.
- [Var97] Prab Varma and Sandeep Bhatia. A Structured Test Re-use Methodology for System on Silicon. In Digital of Paper of *IEEE International Workshop on Testing Embedded Core-Based Systems*, page 3.1-1-8, November 1997.
- [Var98] P. Vorma and S. Bhatie. A Structured Test Re-use Methodology for Core-Based System Chip. In *Proceedings of IEEE Int'l Test Conference*. IEEE CS Press, Los Alamitos, Calif., 1998, pp. 130-143.
- [Wil94] B. R. Wilkins. Test Digital Circuits: An Introduction. *Chapman and Hall*, 1986.
- [Wag96] Kenneth D. Wagner and Sujit Dey. High-Level Synthesis for Testability: A Survey and Perspective. In *Proceedings of the Design Automation Conference*, pp. 131-136, Las Vegas, June 1996.
- [Wan97b] S. Wang and S. K. Gupta. DS-LFSR: A New BIST TPG for Low Heat Dissipation. In *Proceedings of the IEEE International Test Conference*, pages 848-857, 1997.
- [Wan99] S. Wang and S. K. Gupta. LT-RTPG: A New Test-Per-Scan BIST TPG for Low Heat Dissipation. In *Proceedings of the IEEE International Test Conference*, pages 85-94, 1999.
- [Zor90] Yervant Zorian. A Structured Approach to Macrocell Testing using Built-in Self-Test. In *Proceedings of the IEEE Conference on Custom Circuits Conference*, pages 28.3.1-28.3.4, 1990.
- [Zor93] Yervant Zorian. A Distributed BIST Control Scheme for Complex VLSI Devices. In *Proceedings of the 11th IEEE Test Symposium*, pages 4-9, April 1993.

- [Zor97] Yervant Zorian. Test Requirements for Embedded Core-Based Systems and IEEE P1500. In *Proceedings of the IEEE International Test Conference*, ITC'97, pages 191-199, 1997.
- [Zor98a] Yervant Zorian. System-Chip Test Strategies. In *Proceedings of the IEEE Design Automation Conference*, pages 752-757, 1998.
- [Zor98b] Yervant Zorian, E. J. Marinissen and Sujit Dey. Test Embedded Core-Based System Chips. In *Proceedings IEEE Int'l Test Conference*, IEEE CS Press, Los Alamitos, Calif., 1998, pp. 130-143.

Appendix A

Academic benchmarks and industrial designs used to illustrate approaches in the thesis are described in this appendix. The benchmark examples are, a design presented by Kime and Saluja [Kim82], two designs presented by Muresan *et al.* [Mur00a, b]. The industrial designs are ASIC Z presented by Zorian [Zor93] with added data by Chou *et al.* [Cho97], an extended version of ASIC Z, and the System L.

A.1 Muresan's Design One

Muresan *et al.* present a design with the design data presented in Table A.2 [Mur00a]. For instance, test t_2 requires 8 time units and 4 power units and it is test compatible with the following tests: $\{t_1, t_3, t_7, t_9\}$. For instance, it means that test t_2 can be scheduled at the same time as test t_3 .

The power limit for the design is 12 power units.

Test	Test power	Time time	Test Compatibility
t_1	9	9	$t_2, t_3, t_5, t_6, t_8, t_9$
t_2	4	8	t_1, t_3, t_7, t_8
t_3	1	8	$t_1, t_2, t_4, t_7, t_9, t_{10}$
t_4	6	6	t_3, t_5, t_7, t_8
t_5	5	5	t_1, t_4, t_9, t_{10}
t_6	2	4	t_1, t_7, t_8, t_9
t_7	1	3	$t_2, t_3, t_4, t_6, t_8, t_9$
t_8	4	2	$t_1, t_2, t_4, t_6, t_7, t_9, t_{10}$
t_9	12	1	$t_1, t_3, t_5, t_6, t_7, t_8, t_{10}$
t_{10}	7	1	t_3, t_5, t_8, t_9

Table A.1: Design data for Design One by Muresan.

A.2 Design by Kime

The test compatibility graph of a design with six tests is taken from Kime and Saluja [Kim82]. See Figure A.2. Test t_1 and t_6 may be scheduled concurrently since an arc exists between node t_1 and node t_6 . On the other hand, test t_1 and t_2 may not be scheduled concurrently since no arc exists between the node t_1 and node t_2 . Each node has its test time attached to it. For instance, test t_1 requires 255 time units.

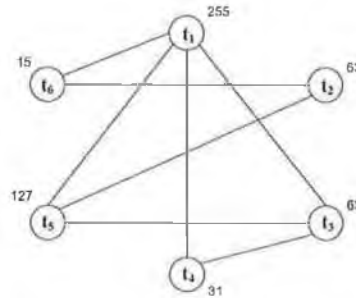


Figure A.2: Test compatibility graph of Design by Kime.

A.3 ASIC Z Design One

The ASIC Z Design One presented by Zorian [Zor93] with the estimations on test length made by Chou *et al.* is in Figure A.3 and Table A.3. The power consumption for each block when it is in idle mode and for each test when it is in test mode is given by Zorian. The test length for each test is computed by Chou *et al.* with an assumption of linear dependency between test length and block size, see Table A.3 [Cho97].

The design originally consists of 10 cores. However, no data is available for one block therefore it is excluded from the design. The maximal allowed power dissipation of the system is 900mW. All blocks have their own dedicated BIST, which means that all tests can be scheduled concurrently.

(The placement has been added also. See Table A.3, where each block is given x-placement and y-placement.)

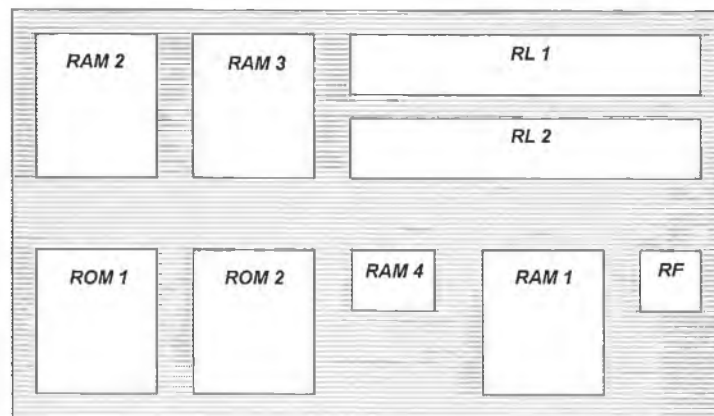


Figure A.3: ASIC Z Design One floor plan.

Block	Size	Test time	Idle Power	Test Power	Placement x y	
RL1	13400 gates	134	0	295	40	30
RL2	16000 gates	160	0	352	40	20
RF	64 × 17 bits	10	19	95	50	10
RAM1	768 × 9 bits	69	20	282	40	10
RAM2	768 × 8 bits	61	17	241	10	20
RAM3	768 × 5 bits	38	11	213	20	20
RAM4	768 × 3 bits	23	7	96	30	10
ROM1	1024 × 10 bits	102	23	279	10	10
ROM2	1024 × 10 bits	102	23	279	20	10

Table A.3: ASIC Z characteristics.

A.4 System L

System L is an industrial design consisting of 14 cores named A through N. See Table A.4. It is tested by 17 tests distributed over the system as block-level tests and top-level tests. The block-level tests and the top-level tests cannot be executed simultaneously. Furthermore, all block-level using the test bus cannot be executed concurrently. The top-level tests use the functional pins, which make concurrent scheduling among them impossible.

All tests are using external test resources and the total power limit for the system is 1200mW.

Test	Block	Test	Test time	Idle power	Test power	Test port
Block-level tests	A	Test A	515	1	379	Scan
	B	Test B	160	1	205	Test-bus
	C	Test C	110	1	23	Test-bus
	D	Test D	Tested as part of other top-level test			
	E	Test E	61	1	57	Test-bus
	F	Test F	38	1	27	Test-bus
	G	Test G	Tested as part of other top-level test			
	H	Test H	Tested as part of other top-level test			
	I	Test I	29	1	120	Test-bus
	J	Test J	6	1	13	Test-bus
	K	Test K	3	1	9	Test-bus
	L	Test L	3	1	9	Test-bus
	M	Test M	218	1	5	Test-bus
	A	Test N	232	1	379	Functional Pins
Top-level tests	N	Test O	41	1	50	Functional Pins
	B	Test P	72	1	205	Functional Pins
	D	Test Q	104	1	39	Functional Pins

Table A.4: System L characteristics

A.5 ASIC Z Design Two

The ASIC Z Design Two presented by Zorian [Zor93] with the estimations on test length made by Chou *et al.* is in Figure A.3 and Table A.3. The power consumption for each block when it is in idle mode and for each test when it is in test mode is given by Zorian. The test length for each test is computed by

Chou *et al.* with an assumption of linear dependency between test length and block size, see Table A.3 [Cho97].

The design originally consists of 10 cores. However, no data is available for one block therefore it is excluded from the design. The maximal allowed power dissipation of the system is 900mW. All blocks have their own dedicated BIST, which means that all tests can be scheduled concurrently.

The experiments on the ASIC Z Design Two are performed. See Appendix A.5, with the following assumptions:

- maximal power dissipation is limited to 900mW,
- all tests can be applied concurrently,
- idle power is not considered, and
- new tests are allowed to start even if all tests are not completed.

A.6 Muresan's Design Two

Muresan *et al.* present a design with the design data presented in Table A.2 [Mur00b]. For instance, test t_2 requires 11 time units and 5 power units and it is test compatible with the following tests: $\{t_3, t_4, t_5, t_9, t_{12}, t_{13}, t_{14}, t_{17}, t_{19}, t_{20}\}$. For instance, it means that test t_2 can be scheduled at the same time as test t_3

The power limit for the design is 15 power units.

Tests	Test power	Test time	Test Compatibility
t_1	(3, 12,	{	$t_4, t_5, t_8, t_9, t_{10}, t_{12}, t_{15}, t_{16}, t_{17}, t_{19}, t_{20}$ }
t_2	(5, 11	{	$t_3, t_4, t_5, t_9, t_{12}, t_{13}, t_{14}, t_{17}, t_{19}, t_{20}$ }
t_3	(9, 9	{	$t_2, t_5, t_7, t_{10}, t_{11}, t_{12}, t_{13}, t_{14}, t_{17}, t_{18}$ }
t_4	(12, 8	{	$t_1, t_2, t_7, t_9, t_{11}, t_{14}, t_{15}, t_{17}, t_{19}$ }

t_5	(4, 8 { $t_1, t_2, t_3, t_6, t_7, t_8, t_{12}, t_{15}, t_{17}, t_{18}, t_{20}$ })
t_6	(2, 8 { $t_5, t_7, t_8, t_{11}, t_{14}, t_{17}, t_{20}$ })
t_7	(1, 8 { $t_3, t_4, t_5, t_6, t_9, t_{12}, t_{14}, t_{15}, t_{16}, t_{18}, t_{19}, t_{20}$ })
t_8	(7, 6 { $t_1, t_5, t_9, t_{10}, t_{11}, t_{14}, t_{16}, t_{17}, t_{19}, t_{20}$ })
t_9	(6, 6 { $t_1, t_2, t_4, t_6, t_7, t_8, t_{11}, t_{12}, t_{15}, t_{17}, t_{19}$ })
t_{10}	(7, 5 { $t_1, t_3, t_8, t_{11}, t_{15}, t_{16}, t_{17}, t_{18}$ })
t_{11}	(5, 5 { $t_3, t_4, t_6, t_8, t_9, t_{10}, t_{14}, t_{16}, t_{18}, t_{20}$ })
t_{12}	(11, 4 { $t_1, t_2, t_3, t_5, t_7, t_9, t_{13}, t_{14}, t_{16}, t_{19}$ })
t_{13}	(2, 4 { $t_2, t_3, t_{12}, t_{15}, t_{16}, t_{17}, t_{18}, t_{19}$ })
t_{14}	(3, 3 { $t_2, t_3, t_4, t_6, t_7, t_8, t_{11}, t_{12}, t_{16}, t_{18}, t_{20}$ })
t_{15}	(1, 3 { $t_1, t_4, t_5, t_7, t_9, t_{10}, t_{13}, t_{16}, t_{17}, t_{18}$ })
t_{16}	(5, 2 { $t_1, t_7, t_8, t_{10}, t_{11}, t_{12}, t_{13}, t_{14}, t_{15}, t_{17}, t_{19}, t_{20}$ })
t_{17}	(4, 2 { $t_1, t_2, t_3, t_4, t_5, t_6, t_8, t_9, t_{10}, t_{13}, t_{15}, t_{16}, t_{18}, t_{19}, t_{20}$ })
t_{18}	(12, 1 { $t_3, t_5, t_7, t_{10}, t_{11}, t_{13}, t_{14}, t_{15}, t_{17}, t_{19}, t_{20}$ })
t_{19}	(8, 1 { $t_1, t_2, t_4, t_7, t_8, t_9, t_{12}, t_{13}, t_{16}, t_{17}, t_{18}, t_{20}$ })
t_{20}	(7, 1 { $t_1, t_2, t_5, t_6, t_7, t_8, t_{11}, t_{14}, t_{16}, t_{17}, t_{18}, t_{19}$ })

Table A.6: Design data for Design Two by Muresan.

Table A.6 is its compatibility list that is already ordered by test length.

Appendix B

Publication

Liang Gao, Xiaojun Wang and Bin Liu. "Merging Technique Based on Tree Growing Technique for Block-Test Scheduling Under Power Constraints." In *Proceedings of 13th IEEE North Atlantic Test Workshop*, pp. 7-12, Vermont, USA. May 13-14, 2004.