EFFICIENT TEST SCHEDULING AND IP MAPPING ALGORITHM FOR NOC BASED SOC

Thesis report submission in the partial fulfillment of the

Requirement for the award of the degree of

MASTERS OF TECHNOLOGYIN

VLSI DESIGN

Submitted by

Prashit Aggarwal

Roll No: 601361018

Under the Guidance of

Ms. Harpreet Vohra

Assistant Professor, ECED

DEPARTMENT OF ELECTRONICAS AND COMMUNICATION

ENGINEERING

THAPAR UNIVERSITY, PATIALA (PUNJAB) – 147004

JUNE-2015

CERTIFICATE

I hearby declare that the work which is being presented in the thesis entitled "**Efficient test scheduling and IP mapping algorithm for NoC base SoC**" in partial fulfillment of the requirement for the award of degree of M.Tech. (VLSI Design) at Electronics and Communication Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of Ms. Harpreet Vohra, Assistant Professor, ECED.

The matter presented in this thesis has not been submitted in any other University/Institute for the award of my degree.

Date: 01/07/15

Prashit Aggarwal

Roll No: 601361018

It is certified that the above statement made by the student is correct to the best of my knowledge and belief.

Ms. Harpreet Vohra

Assistant Professor

ECED, Thapar University

Counter signed by:

Dr. Sanjay Sharma

Professor & Head

ECED, Thapar University

Patiala-147004

Dr. S. S. Bhatia

Dean of Academic Affairs

Thapar University

Patiala-147004

ACKNOWLEDGEMENT

I take this opportunity to express my profound sense of gratitude and respect to all those who helped me through the duration of this thesis. I would have never succeeded in completing my task without the cooperation, encouragement and help provided to me by various people. Words are often too less to reveal one's deep regards. I acknowledge with gratitude and humility my indebtedness to **Ms. Harpreet Vohra, Assistant Professor**, Electronics and Communication Engineering Department, Thapar University, Patiala, under whose guidance I had the privilege to complete this thesis. I wish to express my deep gratitude towards her for providing individual guidance and support throughout the thesis work.

I convey my sincere thanks to **HEAD OF THE DEPARTMENT, Dr. Sanjay Sharma** Electronics and Communication Engineering Department, entire faculty and staff of Electronics and Communication Engineering Department for their encouragement and cooperation.

My greatest thanks to all who wished me success especially my parents and other family members and friends without whom I would not have been able to complete my thesis work.

I thank and own my deepest regards to all of them and all others who have helped me directly or indirectly.

Prashit Aggarwal

ABSTRACT

Advances in semiconductor process and design technology enable the design of complex system chips. Traditional IC design in which every circuit is designed from scratch and reuse is limited to standard-cell libraries, is more and more replaced by a design style based on embedding large reusable modules, the so-called cores. This core-based design poses a series of new challenges, especially in the domains of manufacturing test and design validation and debug.

The manufacturing process may result in defected chips, for instance due to the base material, and therefore testing chips after production is important in order to ensure fault-free chips. The testing time for a chip will affect its final cost. Thus it is important to minimize the testing time for each chip. To reduce test cost, SoCs are being increasingly tested in modular fashion, i.e. their various design modules are tested separately. For core-based SOCs this can be done by testing several cores at the same time, instead of testing the cores sequentially.

Network on Chip (NoC) architecture with regular topology provide scalable platform for designing System on Chip (SoC) with large number of cores. Designers come across a number of challenges and opportunities while developing products and applications using the NoC architecture.

In this thesis report we focus on the various issues faced during the testing of a SoC system. Various schemes have been discussed and implemented to tackle the issues like test time and energy. A genetic algorithm based test solution has been implemented which reduces the test time in a 2D SoC.

The application mapping problem has been optimized by using the Particle Swarm Optimization algorithm to reduce the testing time of the application and the energy dissipated during the communication between the cores.

First a wrapper design algorithm is developed where the testing time of each core is calculated for a given specific bandwidth. Then a partitioning algorithm is developed to reduce the testing time of the circuit by testing different cores in parallel. The partitioning of cores is done considering mesh based NoC architecture. After partitioning the components of the application circuit, the heuristic algorithms are applied for determining the best floor planning of the cores. The proposed schemes are applied on four benchmark circuits from Duke University, Stuttgart University and Philips and they are d695, g1023, p22810, p93791.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

SoC	System on Chip
IP	Intellectual Property
NoC	Network on Chip
GA	Genetic Algorithm
PSO	Particle Swarm Optimization
NP	Non-Polynomial
FF	First Free
NN	Nearest Neighbor
MMC	Minimum Maximum Channel Load
MAC	Minimum Average Channel Load
PL	Path Load
MPSoCs	Multi-Purpose System on Chip
LEC-DN	Lower Energy Consumption based on Dependencies Neighborhood
SMP	Symmetric Multiprocessor
ILP	Integer Linear Programming
MILP	Mixed Integer Linear Programming
EPAM or GMAP	Energy and Performance Aware Mapping
PBB	Performance Aware Branch and Bound
SA	Simulated Annealing

TBMAP	Traffic Balanced IP Mapping Algorithm
NI	Network Interface
SRSI	Single Router to Single IP
SRMI	Single Router to Multiple IP
DRSI	Double Router to Single IP
ACO	Ant Colony Optimization
PMAP	Phase Mapping Algorithm
UMARS	Unified Mapping, Routing and Slot Allocation Algorithm
SMAP	Simulation Based Mapping
BMAP	Binomial IP Mapping
MOCA	Mesh based on-Chip Interconnection Architecture
CMW	Communication Weighted Model
CDCM	Communication Dependence and Computation Model

CHAPTER 1

INTRODUCTION

1.1MOTIVATION

Modern semiconductor process technologies enable the manufacturing of a complete system on one single die, the so-called system chip. Such system chips typically are very large ICs, consisting of millions of transistors, and contain a variety of hardware modules. In order to design these large and complex system chips in a timely manner and leverage from external design expertise, increasingly reusable cores are utilized. ***Cores*** are pre-designed and pre-verified design modules. Examples of cores are CPUs, DSPs, media coprocessors, communication modules, memories, and mixed-signal modules. Core-based IC design divides the IC design community into two groups: ***core providers*** and ***core users.***

Nowadays, large System on Chip (SOC) designs commonly use IP cores that are deeply embedded in the system chip and direct access is often impossible. Individual cores have to be tested on a system level after manufacturing and therefore special Test Access Mechanisms (TAMs) are required. Choosing and scheduling test solutions for SOC embedded IP cores is a very complex problem. In order to facilitate reuse of test vectors provided by the core vendor, an embedded core must be isolated from the surrounding logic, and test access must be provided from the I/O pins of the SOC. Test wrappers form the interface between TAM and core, while TAMs transport test data between SOC pins and wrappers as shown in Figure 1.1.

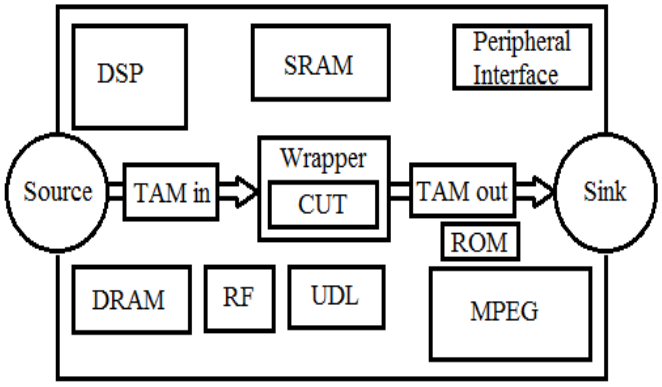

Figure 1.1: SOC Test Model

Test Access mechanism shown in Figure 1.1, is responsible for test data transport between the circuit under test and ATE. The three major components of test access architecture are:

a) Test source and sink

b) TAMs

c) Test Wrappers.

The general problem of SOC test integration includes the design of test architecture, optimization of core wrapper, core assignment and scheduling for test [80]. For SOC testing one can use generic high-performance mixed-signal ATEs, however their high production cost brings limited benefits to complex designs, since cores using heterogeneous technologies still need to be tested sequentially, thus lengthening the testing time and ultimately raising the manufacturing test cost. Due to functional and performance requirements there is a need to optimize the Test Wrapper architecture and the Test Scheduling algorithms.

With the development of the System-on-Chip (SoC) based VLSI products, the integration of various Intellectual Property (IP) cores which perform different functions and working at different clock frequencies, is now a well established one. A shared arbitrated bus forms the basis of the communication system in these SoCs. The performance of the bus based SoC does not improve with the increase in number of attached cores. Thus to meet the demands of the new generation of SoCs with large number of cores, Network-on-Chip (NoC) has emerged as a suitable alternative [1]. A router based network interface is proposed for making the communication architecture modular and scalable [1-3].

Figure1.2 shows the NoC synthesis flow. An application is specified as a collection of tasks in a task graph with different nodes communicating between them. A subset of these tasks can be performed by the cores present inside of the design library, thus the first step is to allocate different tasks to different cores of the library. This forms the core graph, with cores as nodes and communication bandwidth as edge labels. A mapping technique maps this core graph onto a topology graph, and the objective of the mapping algorithm being reduction in the overall communication delay. This mapped graph is scheduled and routed to produce the final NoC.

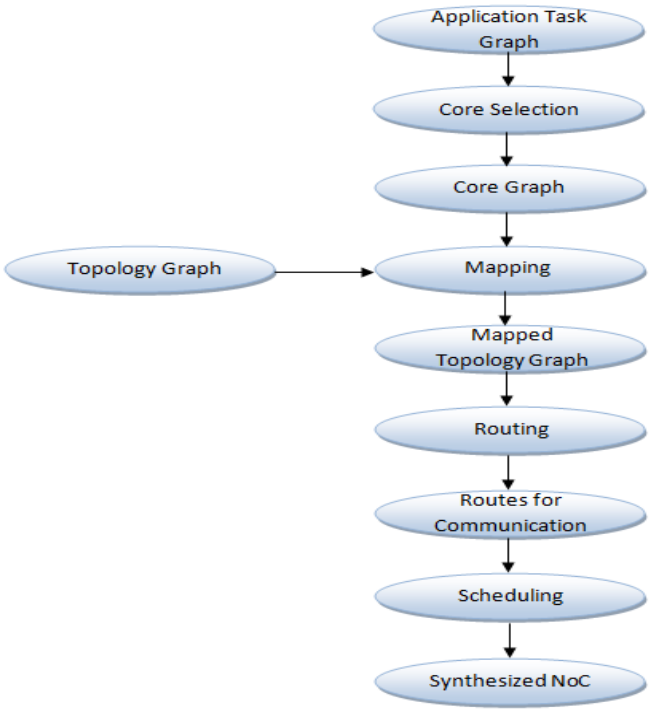

Figure1.2 Application specific NoC Design Flow

The research problem in the NoC design paradigm can be broadly classified into 4 different areas [4-6]. The first area addresses the choice of communication infrastructure, such as, network topology, router architecture, buffer optimization, link design, clocking, floor planning and layout. The second area focuses on the communication paradigm including routing policies, switching techniques, congestion control, power and thermal management, fault tolerance, reliability etc. The third dimension involves designing an evaluation framework for NoC to have a good understanding of achievable throughput, latency, and bandwidth of the network. After the communication infrastructure has been designed the task of allocating the IP cores performing the designated tasks to the routers of the NoC system is performed. This association of cores to routers plays a significant role in determining the performance of the overall system as it influences communication time, required link bandwidth and admissible delay of the router. And thus application mapping forms the forth important area in the NoC research. Figure 1.3 shows the mapping of an application task graph onto the multi/many-core system platform resource. The communication tasks are mapped on the same core or close to each other in order to optimize the communication delay and energy.

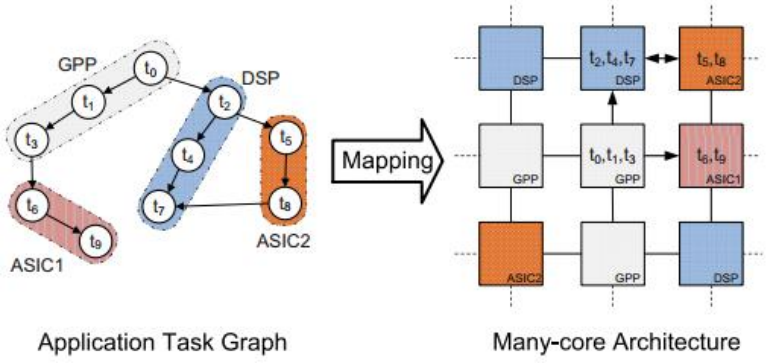

Figure1.3 Application Mapping on many core systems

1.2HEURISTIC ALGORITHMS

The problem of test scheduling and NoC application mapping being NP hard can be optimized by using the heuristic algorithms.

- GENETIC ALGORITHM

The Genetic Algorithm is used to schedule the IP cores for testing such that the testing time of the SoC can be reduced.

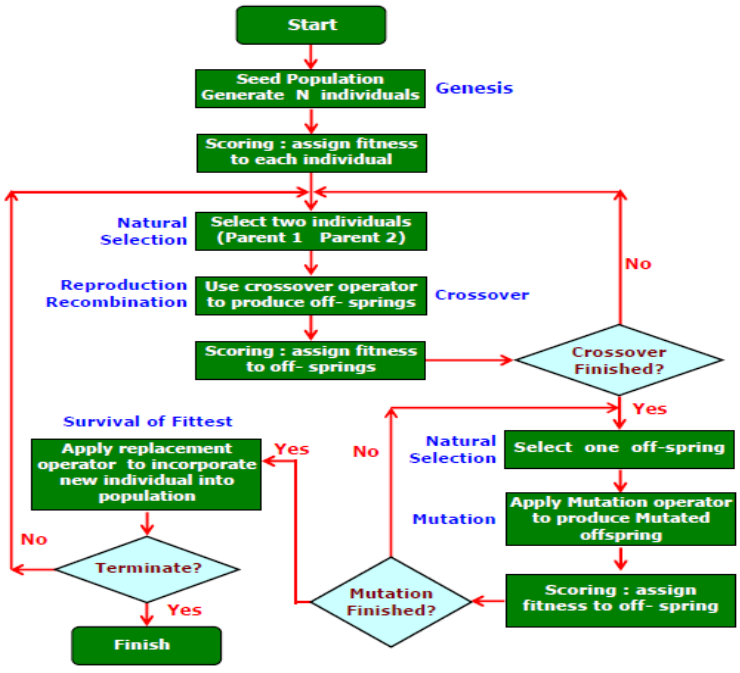

Figure 1.4 Basic Flow diagram of Genetic Algorithm

GA uses a direct analogy of such natural evolution. It presumes that the potential solution of a problem is an individual and can be represented by a set of parameters. These parameters are regarded as the genesof a chromosome and can be structured by a string of values in binary form. A positive value, generally known as fitness value, is used to reflect the degree of "goodness" of the chromosome for solving the problem, and this value is closelyrelated to its objective value.

- PARTICLE SWARM OPTIMIZATION

The other meta-heuristic algorithm used for arriving at an optimal floor planning of the IP cores onto NoC architecture is the Particle swarm optimization algorithm. In PSO system, multiple candidate solutions coexist and collaborate simultaneously. Each particle represents a solution and it evolves on its own experience and also with the contribution from other particles.

1.3THESIS ORGANIZATION

This report is organized into five chapters briefed as under:

Chapter1: Introduces the motivation for our thesis outlines the problem of test scheduling and the application mapping, various mapping algorithms and states the contribution of our work done.

Chapter2: This is the literature review which constitutes of Description of the test scheduling problem and various application mapping strategies

Chapter 3: This chapter gives the description of the proposed work.

Chapter 4: This chapter gives the results and evaluations of the proposed schemes.

Chapter 5: This chapter concludes the work.

CHAPTER 2

LITERATURE REVIEW

2.1 2D SoC Test Model

Integrated circuits (ICs) are embedded now-a-days in a wide range of products and systems, from consumer electronics and medical equipment to automotive and aviation systems, which usually require high availability and where the cost of failures can be immense. The first IC available commercially was produced by Fairchild Semiconductor Corp. in 1961; it contained one transistor, three resistors and one capacitor. The everlasting improvements in semiconductor fabrication technology have led to ICs with billions of transistors. Such large ICs can contain all necessary electronic circuitry for a complete system and are referred to as SOCs.

ICs can be extremely complex and time-consuming to design. In order to meet short time to-market requirements, it is therefore common to make use of a modular core-based design approach where a system is composed of pre- designed and pre-verified blocks of logic, so-called cores. The cores can be designed in-house or bought from core vendors, and it is the task of the system integrator to integrate them into a system.

The increasing cost for IC testing is in part due to the huge test-data volume (number of bits), test stimuli and expected responses, which can be in the order of tens of gigabits.

The huge test-data volume leads to long test application time and requires large tester memory.

Test cost for large SOCs can be viewed as consisting of [77]:

1) **Explicit test cost(Cost of investing in a new ATE, also known as Capital Expenditure):** Complex cores often require expensive ATE resources such as high-frequency channels, high pin counts, large memory depths as well as special features for analog and RF cores. As a result, older-generation ATE are often inadequate and large investments in new ATE must be made.

2) **Implicit test cost:** Large SOCs require long test sequences to guarantee high levels of fault coverage for embedded cores. This has led to an increase in testing time during which the SOC sits on an expensive ATE, thereby preventing other SOCs from being tested. This in turn leads to increased time-to-market and decreased profitability

The cost of test is a significant part of the overall manufacturing cost (including the cost of fabrication and the cost of test). The high test cost for core-based SOCs can be reduced by adequate test planning, which includes:

• Test-architecture design

• Test scheduling

• Test-data compression

The cores in a system can have different origin and they can be classified into three categories:

- Hard Cores
- Soft cores
- Firm Cores

Hard cores are given as layout files that cannot be modified. These types of cores are highlyoptimized for area and performance, and synthesized to a specific technology. And alsothe test sets (test stimuli and test responses) are given. Soft cores, on the other hand, aregiven in a hardware description (HDL) language, hence, technology independent, and thesoft cores can easily be modified compared to hard cores. The soft core specification hasto be synthesized and optimized, and also it is required to perform test generation. Firmcores are given as technology-dependent netlists using a library with cells that can bechanged according to the core integrator's need.

SOC test development is especially challenging for several reasons. Embedded coresrepresent intellectual property, and core vendors are reluctant to divulge structuralinformation about their cores to users. Thus, users cannot access core net lists and insertdesign-for-testability (DFT) hardware that can ease test application from the surroundinglogic. Instead, the core vendor provides a set of test patterns that guarantees specific faultcoverage. These test patterns must be applied to the cores in a given order, using aspecific clocking strategy. Care must often be taken to ensure that undesirable patternsand clock skews are not introduced into these test streams. Furthermore, cores are oftenembedded in several layers of user-designed or other core-based logic and are not alwaysdirectly accessible from chip I/Os. Propagating test stimuli to core inputs may thereforerequire dedicated test transport mechanisms. Moreover, it is necessary to

translate testdata at the inputs and outputs of the embedded-core into a format or sequence suitable for
application to the core.

A core test wrapper named TestShell consists of following components:

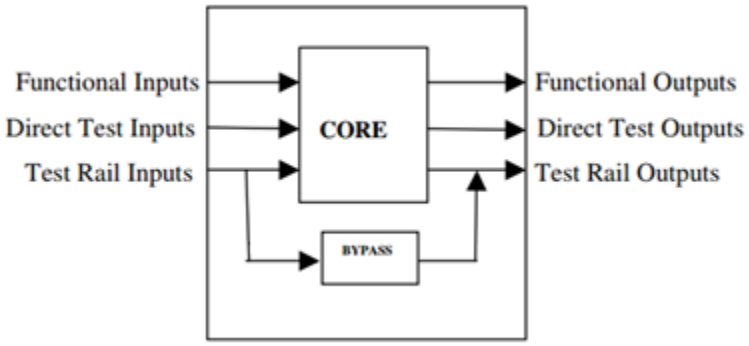

Figure 2.1 Testshell/Wrapper

a) **A Test Cell** for every core terminal. The test cell provides controllability as well as observability.
b) **A Bypass Register** allows a TAM to bypass core and wrapper, in order to test another core that is connected to the same TAM.
c) **A Test Control Block (TCB)** The TCB has a bit-slice nature and consists of a shift and an update register. The TCB is primarily meant to control the operation of the Testshell, through several mandatory bit slices

For SOC designswhere modules are not tested prior to mounting leads to an increasing amount of test data volume to be transported in a system. For that reason, core wrappers such as TestShell, TestCollar and P1500 have to be developed [76]. The TestShell consists of three layers of hierarchy:

• The core or the IP module,

• The TestShell, and

• The host.

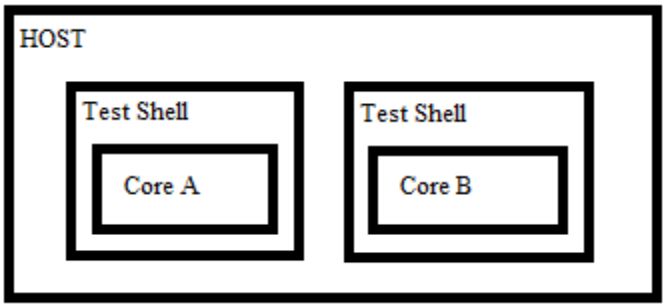

Figure 2.2 SoC Model with Wrapped IP Cores

The core is the object to be tested. The host is the environment where the core is embedded. It can be a complete IC, or a design module which will be an IP module itself. Finally, the TestShell is the interface between the core and the host and it contains three types of input/output terminals:

• Function input/output corresponds one-to-one to the normal inputs and outputs of the core.

• Testrail inputs and outputs are the test access mechanism for the TestShell with an optional bypass.

• Direct test input/output are used for signals which cannot be produced through the TestRail

The general problem of test scheduling for SOCs is related to the NP-Hard. The test scheduling is described for the busbased TAM-architecture and for Flexible-width architecture. Using bus-based TAM architectures, for transporting test-data usually entails a sequential schedule, and hence, only one core is tested at a time, as shown in Figure 2.3. The transportation of tests on the functional bus bf1 is shown in Figure 2.3 (a). The example shows that the bus is fully occupied all the time. Still, the cores are only activated one after the other (Figure 2.3 (b)). This makes the scheduling very simple but the drawback is the long test application time obtained because the cores are not tested in parallel

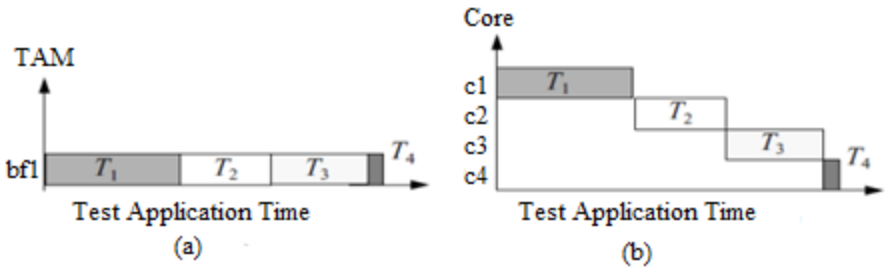

Figure 2.3 (a) sequential test schedulingusing one functional bus bf1 and (b) test application

When a bus-based TAM is used, concurrent test scheduling where multiple tests are scheduled in parallel is only possible by adding multiple TAMs. Such a test-architecture is shown in Figure 2.4 where two Test buses, bt1 and bt2, are used. In the example, c1 and c2 have been assigned to bt1 and c3 and c4 have been assigned to bt2. An example of a concurrent test schedule using bt1 and bt2 is shown in Figure 2.4. The transportation of tests on the Test buses bt1 and bt2 is shown in Figure 2.4 (a). The corresponding test application, where c1 is tested at the same time as c3 and c4, is shown in Figure 2.4 (b).

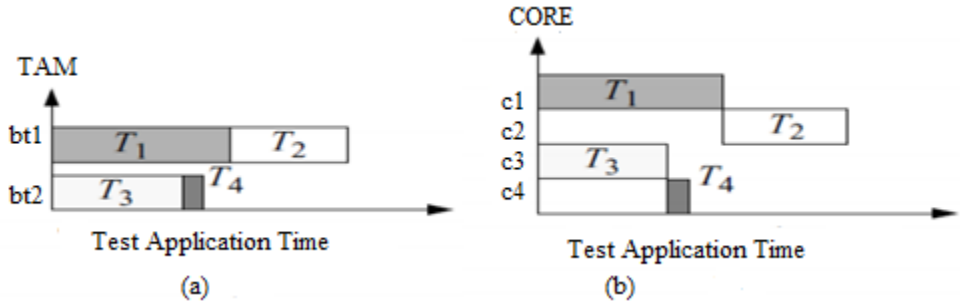

Figure 2.4 (a) concurrent test scheduling and (b) test application using two Test buses bt1 and bt2

As opposed to the bus-based TAM architectures, the Flexible-width architecture allows concurrent test transportation and application even if only one TAM is used. An example of concurrent test scheduling and application using the Flexible-width architecture is shown in Figure 2.5. The transportation of tests on the TAM wires is shown in Figure 2.5(a). The corresponding test application, where c1 is tested at the same time as c2 and c3, is shown in Figure 2.5(b).

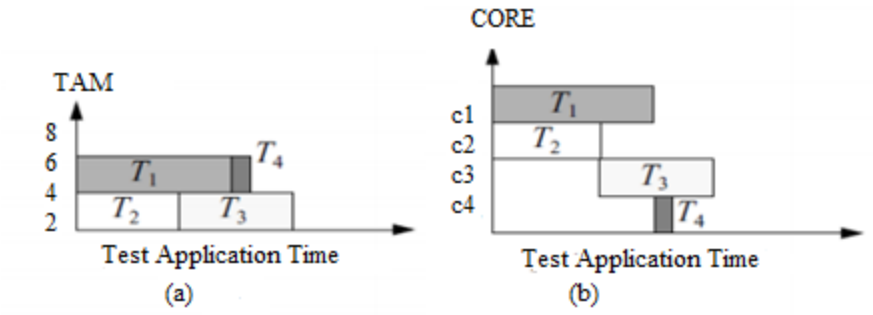

Figure 2.5 (a) concurrent test scheduling and (b) test application using Flexible-width architecture

Test scheduling techniques can be divided into the following three categories:

a) Non-partitioned testing

b) Partitioned testing with run to complete

c) Partitioned testing.

The three test scheduling techniques using the four tests. In the example, it is assumed that the cost function is the test application time, which will be minimized without violating the hardware constraint given by the maximum number of TAM wires. Figure 2.5 shows an example of a test schedule using a non-partitioned technique. In non-partitioned test scheduling no new test is allowed to start until all tests in a session are completed. This method produces long test application times due to long periods of time when no core in the system is tested, so called idle times. The test schedule can be improved by using partitioned (session less) techniques [76]. In the partitioned technique, tests are allowed to be scheduled as soon as possible, which can decrease the test application time. However, a more advanced test controller is required for the invocation of tests since more possible start times of tests can be used. In order to further optimize the schedule, a pre-emptive test scheduling technique can be used.

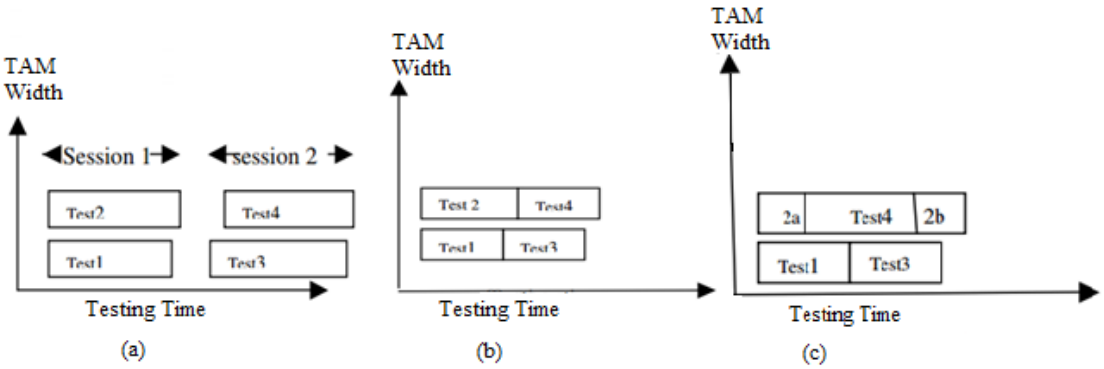

Figure 2.6 (a) Non-Partitioning Testing (b) Partitioned testing with run to completion (c) Partitioned testing [76]

2.2 NoC Application Mapping Techniques

The application mapping problem is a non-polynomial (NP) hard problem [7]. Depending upon the time of assignment of tasks to the IPs, the mapping problem can be classified into dynamic mapping and static mapping as show in Figure 2.7.

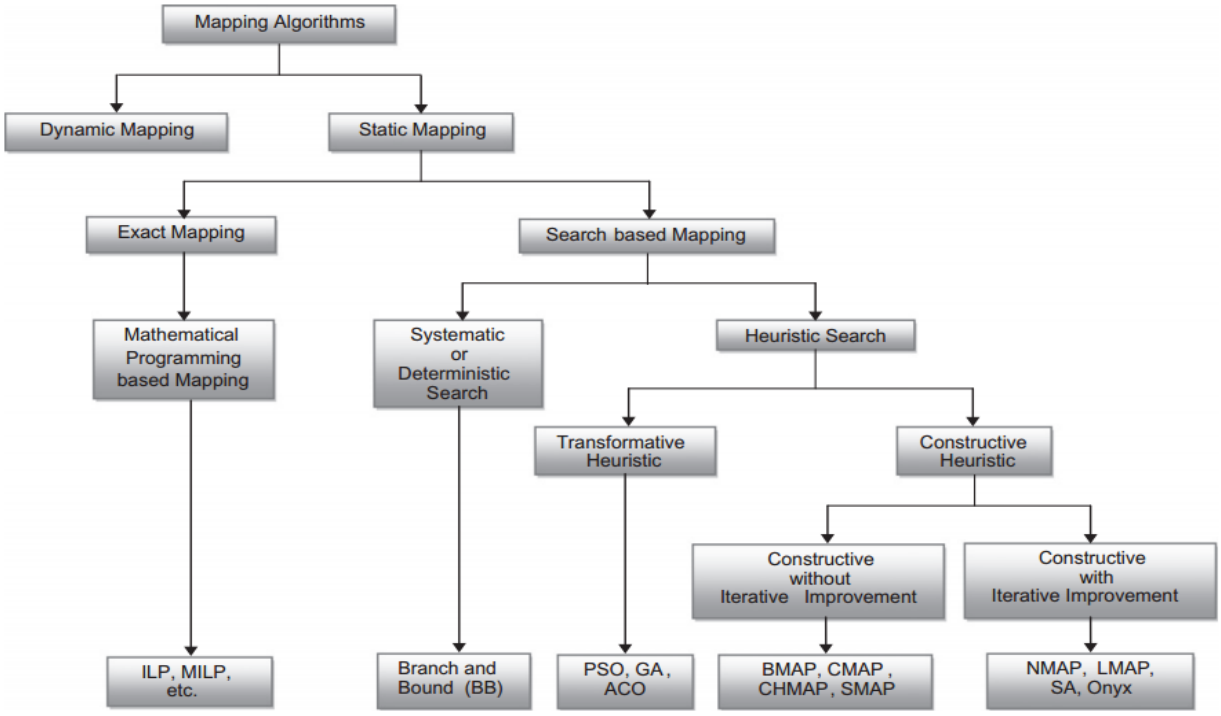

Figure 2.7Classifications of Mapping Algorithms [78]

When the assignment and ordering of tasks is performed at the time of execution of the application then it is called on-line or dynamic mapping. Dynamic mapping is effective as it distributes the load among the processors by detecting the performance bottlenecks. The computational overheads of mapping algorithms increase the delay and energy consumption and thus shadow the positive effects of dependence of mapping on the current load of processors. While in case of static mapping the mapping of tasks is performed off-line, before the application is run. Static mapping algorithm is run only once as the mapping is performed before the execution. Thus for NoC, static mapping is recommended as the excess system overheads significantly affect system performance, increase the overall delay of the system [7].

2.2.1Dynamic Mapping Techniques

Dynamic mapping is an on-line mapping strategy, the tasks ready at the time of execution are mapped to the processors by observing the load of the processors at the run time. In [8], G. Chen et. al proposes a mapping scheme which performs task scheduling, data mapping and packet routing. The drawback of the proposed algorithm is the requirement of high computation time which degrades the system performance. Authors in [9, 10] have presented a heuristic with an initial task mapping phase for dynamically mapping the tasks. The dynamic mapping phase may use any one of the techniques, such as, First Free (FF), Nearest Neighbor (NN), Minimum MaximumChannel Load (MMC), Minimum Average Channel Load (MAC), and Path Load (PL). In FF, NoC does a column search to find out the first free node that can perform the requested task. NN mapping being similar to FF, places the requested task at the free neighboring node of the node making the request. MMC is used to reduce the maximum loads in the links. MAC technique being similar to MMC, distributed the load in NoC to reduce the average load in the link. MAC and MMC are time consuming mapping schemes as they consider all the NoC links for mapping a new task. This problem is overcome by the PL technique which considers only the links that are used by the task being mapped. [9, 10] show that the PL technique produces the best solution as compared to others. A run-time application mapping scheme has been proposed by the authors in [11] for multiple voltage levels. A heuristic region selection algorithm is used for different regions which are being operated at different voltage levels. This saves about 50% more communication energy as compared to the random mapping schemes. In [12], author incorporates the user behavior information in the resource allocation process. This provides a better adaptation to the user needs. In [13], Dynamic mapping of real time applications is done onto embedded MPSoCs where communication is performed via NoC. An energy-aware heuristic for dynamic task mapping,named lower energy consumption based on dependencies-neighborhood(LEC-DN) has been presented in [14, 15]. The cost functions in these schemes include distance in hops between communicating tasks, proximity in number of hops and the communication volume among the tasks. LEC-DN uses the Nearest Neighbor (NN) search in a spiral fashion when the target task has only one communicating task. In case when more than one communicating tasks are available that

have already been mapped, the algorithm searches for a processing elementinside the bounding box defined by the position of such tasksdepending upon the communication volume. Author in [16] proposes a dynamic decentralized, application- driven and resource-aware mapping, where tasks can be embedded incrementally by an already mapped predecessor task.

2.2.2 Static Mapping Techniques

Static mapping is defined as the technique where the resources allocated for the execution of the task are decided before the execution and are not changes thereafter. It is an off-line mapping where all the cores are mapped to routers at design time. Based upon the technique used for the search for a mapping solution, static mapping can be classified as either Exact mapping or Search based mapping.

2.2.2.1 Exact Mapping

Mapping done with the help of mathematical programming produces optimal solutions. In [17], a mixed integer linear programming (MILP) based task mapping has been proposed for heterogeneous multiprocessor systems where some processors are application specific and some are programmable. An MILP formulation for mapping cores onto NoC whileconsidering the choice of core placements, switches for each core,and network interfaces for communication has been proposed in[18]. This technique provides less energy consumption in mapping for real and random benchmarks. For synthesis of custom NoC architectures, author suggests a MILP formulation in [19]. The main objective here is to reduce the power consumption under the performance constraint. Author uses an algorithm to partition the task graph into number of clusters to reduce the runtime which is the main bottleneck in a Linear Programming system. Partial solutions are generated using the MILP formulation for topology design. Final mapped custom topology is generated by adding physical links between the ports of neighboring routers of the clusters.

Features like symmetric multiprocessing(SMP), block multi-threading, and multiple memoryelements to support high performance networking applications are incorporated into network processors. Mapping an application to a complex multi-processor is a

difficult task. A two stage Integer Linear Programming (ILP) formulationfor process allocation and data mapping on SMP and blockmulti-threading based network processor has been proposed by the author in [20]. Reduction of energy by shutting down certain communication links has been attempted in [21], where a formulation is used for selecting the links in use and voltage, frequency for those links. The problem of minimization of energyconsumption during application execution while satisfying theperformance constraint may be combination of some sub-problems,such as, mapping of application tasks to IPs, mapping of IPsto the routers of NoC architecture, assigning operating voltagesto IPs, and routing. Authors in [22] discuss an ILP formulation for mapping an application onto a mesh based NoC, but they do not include the bandwidth constraint. The other drawback of the technique used i.e, the high CPU time has been addressed in [23] by using a cluster based relaxation for ILP formulation.

2.2.2.2 Search Based Mapping

The search based mapping techniques can be divided on the basis of search type and results into – (i) systematic or deterministic search and (ii) heuristic search.

(i) Deterministic Search

Branch and bound search algorithms are used in this category. These are systematic search algorithms that topologically finds the mapping by searching the solution in tree branches and bounding unallowable solutions.

An energy and performance aware mapping for tile based regular NoC architectures has been discussed in [24-26] to satisfy the specific design constraints through bandwidth reservation. [25, 26] suggest that the most effective routing technique for NoC should be deterministic, deadlock-free, minimal and wormhole based. The network wires being structured and modular provide better controllability and optimization of electrical parameters. First an Energy- and Performance-Aware Mappings (EPAM or GMAP) in topological sense is formulated and then an efficient Performance-awareBranch-and-Bound (PBB) algorithm is utilized to improve the solution quality. Better energy saving has been reported forEPAM combined with PBB, compared to Simulated Annealing (SA) based solutions. In the above mapping, single IP is connected to a router. An IP with

large communication volume will result in a heavy traffic load to certain routers, which may become hotspotdue to high power density that affects the reliability of the chip. [27], proposes a traffic balanced IP mapping algorithm (TBMAP) for 2-D mesh network. TBMAP balances the traffic on all the routes without sacrificing the network performance. The unbalanced trafficloads are reduced by using various network interfaces (NIs),Single-Router to Single-IP (SRSI), Single-Router to Multiple-IP(SRMI), and Double-Router to Single-IP (DRSI). [24-26] use a modified branch and bound search to map all the IPs.

(ii) Heuristic Search

A large number of heuristic approaches can be used to solve the problem of application mapping. They can be broadly classified into Transformative heuristics and Constructive heuristics.

- Transformative Heuristics

Transformative heuristics like Genetic Algorithm (GA), particle Swarm Optimization (PSO), Ant Colony optimization (ACO) and so on arrive at a better solution by transforming an existing mapping solution.

GA based transformative heuristics

Genetic Algorithm (GA) is a stochastic search algorithm based on operations of natural genetics. Here, fixed-sized population of chromosomes evolves over a number of generations following the principle of natural selection. Each chromosome identifies a potential solution. A chromosome has got an associated fitness measure. Using the operatorssimilar to crossover and mutation in nature, the population evolves through generations. To evolve a new generation, generally top few percentages of chromosomes are directly copied to the next generation. Rest of the population is created by two operators – crossover and mutation. The crossover operator selects two parent chromosomes to participate in the operation. Their parts are exchanged to create new offspring. The crossover may be singlepointor multi-point. The mutation operator may be implemented by selecting a parent chromosome and randomly changing some of its

portions. The mutation rate can be controlled to control the rate of convergence to local or global minima. The termination criterion is often set to be ‘‘no improvement in last few generations’’ or ‘‘a specified number of generations for which the GA has run’’.

Author in [27] proposes a two step Genetic Algorithm (GA) for mapping an application onto NoC. The proposed algorithm reduces the overall execution time by first assigning the tasks onto different IPs assuming the edge delays to be constant and equal to an average value. In the second step based on the actual traffic model the assigned IPs assume the actual edge delays and the total system delay is minimized.

In [28] the author proposes a delay model considering the factors such as the message sending probability of cores, packet length and the network contention for communication. These factors were not considered by the previous work [27]. In [28], the population corresponds to the core position to the NoC topology. The initial population is randomly created with average waiting time. To generate the new population a randomly chosen multi-point crossover operator is applied. The probability of a core participating in crossover is inversely related to its waiting time. To avoid a local maxima or minima, mutation operator is used. The above operators are applied repeatedly until a chromosome with lowest average waiting time is produced. The best solution at end generates the optimal core positions on the NoC.For example, let the cores a to f to be mapped onto a (2 _ 3) 2-D mesh, and the final chromosome structure is (1 2 1 3 3 6). In this structure, the first integer is used to map a, the second integer is used to map b, and so on. All the solutions will have a 1 in the first position. The second core b can be mapped onto two places, that is, before a (with a value 1) or after a (with a value 2). So, now b is placed after a as the integer value for b is 2 in the chromosome structure. Then there are three placeholders for c, that is before a (with a value 1), in between a and b (with a value 2), or after b (with a value 3). For c, the integer value is 1, so it is placed before a. Similarly the other cores are placed and the final positions of cores are obtained as (c a e d b f) according to the chromosome structure. Utilizing this representation for the chromosomes, cores are placed in a (2 _ 3) 2-D mesh connected NoC according to the positions obtained in the final solution as shown in Figure 2.8.

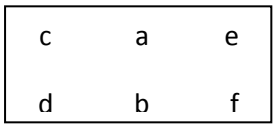

Figure 2.8. Final core placement [28]

In [29] author proposes a pareto based multi-objective evolutionary computing technique to optimize performance and power-consumption of mapping NoC. Same author in [30] used the above technique for application task mapping. An event driven trace based simulator has been used to compare their results with pareto based Branch- and-Bound approach [30] and pareto based NMAP approach [30]. A multi objective genetic algorithm using network assignment for heterogeneous distributed embedded systems is proposed in [31]. The network assignment is done after task mapping to improve the performance and reduce the power consumption and area.

The Genetic Algorithm based optimization technique MGAP proposed in [32] minimizes the power consumption by reducing the number of switches in the communication path between cores and also maximizes the throughput. Though similar technique has been used in [27], but here authors have considered the dynamic effect of traffic. They have also given a set of solutions using pareto mapping as used in [29, 30]. A multi-objective Genetic Algorithm (MOGA) based application mapping technique has been proposed in [33], where one–one as well as many–many mapping between switches and tiles has been taken into consideration to minimize energy consumption and required link bandwidth. It is used to find optimal solution from the pareto optimal solutions as in [32]. The chromosome is representation of the mapping solution [32, 33] which is formed by m×n genes, where, m is the number of rows and n is the number of columns of the mesh connected NoC. The ith gene corresponds to the core in the tile having row $\lceil i/n \rceil$ and column (i%n). Here the crossover is single-point, and during the crossover the maximum communicating cores are remapped to random tiles result in a new chromosome. The mutation operation is performed upon a chromosome by choosing highly communicated cores and placing them nearby to each other.

The main drawback of the genetic approach a the slow rate of convergence as sometimes it is required to generate a large number of generations to arrive at an optimal solution.

To improve the rate of convergence the mutation rate can be improved, but it mostly converges to local best solution rather than the global solution.

PSO and ACO based transformative heuristics

Particle swarm optimization (PSO) [34] is a population based stochastic technique developed by Eberhart and Kennedy in 1995, inspired by social behavior of fish schooling or bird flocking. In a PSO system, multiple candidate solutions co-exist which evolve according to their own experience as well as the experience of others.

In [35] author proposes a PSO based two phase application mapping algorithm which minimizes the NoC communication energy and allocates the routing path for balancing the link loads. The particle structure and the initial particle generation is similar to the GA based technique proposed in [28]. In [35] author proposes use of PSO algorithm for NoC mapping and reports improvements over genetic algorithm based mapping. In [36], Dijkstra's shortest path algorithm has been used in a hybrid multi-objective algorithm to find the shortest path among communicating cores to satisfy bandwidth constraint and then pareto based PSO is used to improve the results.

In [37] a particle structure has been shown as in Figure 2.9. The numbers shown within circles in the boxes are the core numbers present in the core graph. The numbers outside the box are the router numbers of the topology graph. It is assumed that the routers are numbered in an increasing order from top left to bottom right position. The figure shows that core 1 is attached to router 0, core 4 is attached to router 1, and so on. If the number of nodes (routers) present in the topology graph is greater than the number of cores present in the core graph, dummy nodes are added to the core graph to make the two numbers same. Dummy nodes are connected to all core nodes and between themselves. Edges connecting a core node to dummy nodes and the edges between dummy nodes are assigned a cost zero. Let N be the number of cores present in the core graph, after connecting dummy nodes, if required. For these N cores, there are N node positions in the topology graph. A particle is a permutation of numbers from 1 to N, which shows the placement of cores to the node positions of the topology graph. The overall communication cost is influenced by the position of cores in a particle. In our formulation, the overall communication cost forms the fitness function. Fitness of a

particle pi is equal to the overall communication cost after placement of cores of the core graph to different routers, as specified by the particle.

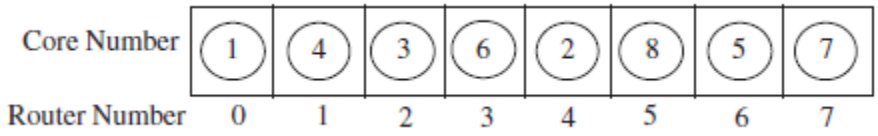

Figure 2.9. Particle Structure [37].

In the evolution process, every particle i has its correspondinglocal best $pbest^i$, which is the permutation of core positions thatgives the minimum communication cost, among all permutationsthat the particle has seen so far. The local best permutation valueguides partially the evolution of the particle. For a particular generation,the particle resulting in the minimum communication costis the global best (g_{bestk}) for that generation. This parameter alsoguides the evolution of particles. The particles evolve through generationsto create new particles which are expected to give resultscloser to the optimum. In the first generation, the initial populationis created randomly and the fitness of individual particles is evaluated.The local best (p_{best}^i) of each particle is set to be same as theinitial particle. The global best (g_{bestk}) of a generation is the particlegiving the least communication cost (smallest fitness function)in that generation. Further generations are evolved through a seriesof operations called swap operations [38]. The local best of eachparticle and the global best of a generation are modified if the correspondingvalues in the current generation are lesser than the valuesin the previous generation.

For a particle p, the router associated with a core is identified by the position index of the core in p. The indexing of the position takes value between 0 and N – 1 (N being the number of routers). The index corresponds to the router number. Let the swap operator be $SO_{j,k}$ (where, j and k = 0,1,. . .N _ 1) that swaps j^{th} and k^{th} positions of the particle p to create a new particle p_{new}. For example, let us consider the particle p = {1, 4, 3, 6, 2, 8, 5, 7}, where the numbers represent the core numbers of the core graph and the position represents the router numbers in the topology graph. The swap operator $SO_{4,6}$ swaps the cores at positions 4 and 6, which creates a new particle p_{new} = {1, 4, 3, 6, 5, 8, 2, 7}.

A swap sequence SS is made up of one or more swap operators. The swap operators of the swap sequence are applied, in order, upon the particle p to create a new particle p_{new}.

For example, let the swap sequence SS = {$SO_{4,6}$, $SO_{2,5}$} be applied upon the particle p = {1, 4, 3, 6, 2, 8, 5, 7}. It creates a new particle p_{new} = {1, 4, 8, 6, 5, 3, 2, 7}.

To align a particle pi with its local best, the swap sequence is identified. Let this be $SS_i^{l\_best}$. Then another swap sequence is identified to align the particle with the global best. Let this be $SS_i^{g\_best}$, now the swap sequence $SS_i^{l\_best}$ is applied on particle p_i with a probability of α[39]. Let the modified particle be $P_i^{l\_best}$. Then the swap sequence $SS_i^{g\_best}$ is applied on $P_i^{l\_best}$ with a probability of β[39]. This creates a new particle P_i^{new} . Its fitness is evaluated and the local best is updated for particle i, if it is better than the previous local best for the particle. If the best fitness in a generation is better than the global best of the previous generation, the global best is also updated.

Ant colony Optimization (ACO) technique was developed by A. Colorni and M. Dorigo in 1991. It is a population based probabilistic technique inspired by the behavior of ants in finding a path from their colony to the food source. Thus when a ant finds a good path from colony to food source a positive feedback eventually leads others to follow the similar path. Author in [40] presents an ACO based mapping technique to minimize the bandwidth requirement. The results were compared to a random mapping technique.

- Constructive Heuristics

In this technique, partial solutions are generated in sequence and at the end final solution is generated at the end. These techniques are much faster than the transformative heuristics. They can either be constructive heuristics with iterative improvement or constructive heuristics without iterative improvement.

Constructive heuristics without iterative improvement.

In constructive heuristic without improvement the cores are selected for mapping onto the NoC topology one at a time according to some predefined criteria. Once placed, the placement of the core is not changed. No optimization technique is applied upon the initial solution to arrive at a better one.

Placing of clusters onto processors using a two phase mapping algorithm has been proposed in [41], A two phase mapping algorithm (PMAP) where highly communicating nodes are placed at the adjacent nodes of the processor network. All tasks needed to be performed by the processor are inside one cluster so as to achieve zero interconnection

overhead to increase parallelism. A unified mapping algorithm (UMARS) is proposed in [42] couples mapping, path allocation and time slot allocation to reduce communication energy. This technique maps cores onto NoC topology, route the communication and allocate TDMA slots on network channels so that application constraints are met. Simulation based mapping algorithm (SMAP) proposed in [43] is a simulation based environment which performs application mapping and task routing for 2D mesh based NoC to minimize execution time and communication energy. The highest priority task is mapped at the centre and other tasks are mapped from the mapped task spirally [44] as shown in Figure 2.10.

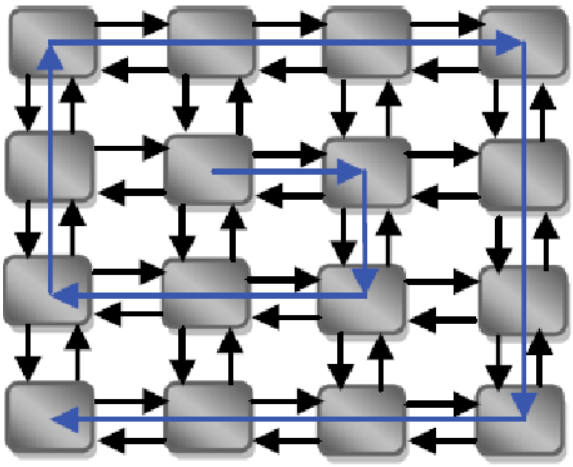

Figure 2.10. Application mapping onto NOC [43, 44]

Author in [45] presents an efficient binomial IP mapping (BMAP) and optimization algorithm to reduce hardware cost of on-chip network. It is a very fast and effective algorithm having less computation complexity than NMAP [46]. The binomial mapping comprises of three steps – IP ranking, merging IP set, and refreshing IP set. IP ranking depends upon the communication bandwidth between them. The communication bandwidth of an IP is the sum of the bandwidth from it to other IPs and from other IPs to it. Depending on the IP ranking, the most communicated IP sets are merged two-by-two every iteration as shown in Figure 2.11. The new requirements of merged IP sets are recalculated by taking each IP set as an individual IP, which refresh the IP set.

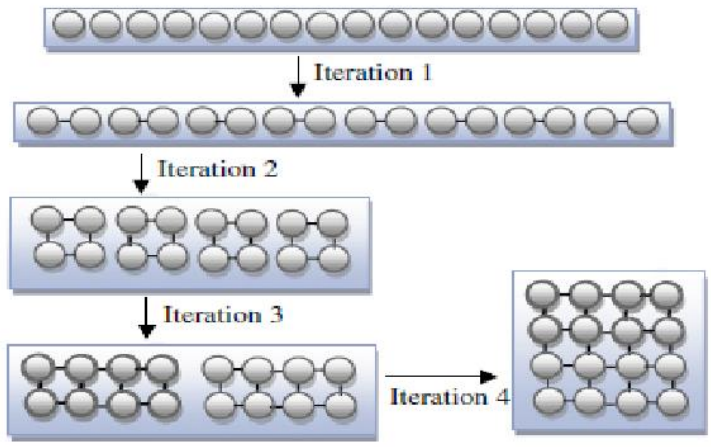

Figure 2.11. An example of binomial merging (N=16) iterations [45]

Energy aware and application mapping and routing technique CastNet, for 2D NoC has been proposed in [47]. Based upon the total communication bandwidth and average communication bandwidth a priority list for the tasks is formed. Depending upon this priority list the tasks are selected, if there is a conflict then the tasks are selected at random. The next task which is most communicated with mapped task is selected for mapping, in case of a tie, higher priority task is selected. For mapping the first task, a set of initial node positions is selected as shown in Figure2.12. A set of solutions are generated by this technique for each initial node position for the initial task. The remaining tasks are placed on the nodes of NoC according to the priority list. After each mapping the priority list is updated. Finally, from the set of solutions, the best one is taken as the solution for mapping of application onto NoC.

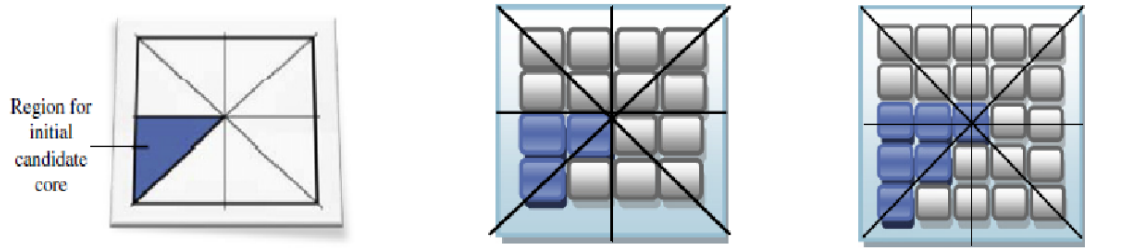

Figure 2.12. Candidate for initial core selection [47], (a) selected region for initial candidate core. (b) Initial candidate core (4X4 mesh). (c) Initial candidate core (5X5 mesh)

Constructive heuristics with iterative improvements.

In constructive heuristic with improvement the cores are selected for mapping onto the NoC topology one at a time according to some predefined criteria. Once placed, the placement of the core is not changed. Optimization technique is applied upon the initial solution to arrive at a better one.

In [46], NMAP, a minimum path routing mapping technique has been proposed which satisfies the bandwidth constraint and minimizes the average communication delay. This algorithm has 3 stages. In first stage, the core having the maximum communication demand is placed to a node having maximum neighbors. Then the core having maximum communication demand with the already placed core is selected and placed such that the communication cost is minimized (hop-count × Bandwidth). This is done by examining all the nodes remaining in the mesh. In the second stage, Dijkstra's shortest path algorithm is applied to the quadrant graph for minimum path computation with satisfaction of bandwidth constraints. In the last phase, the initial solution is improved iteratively by invoking the second phase for each pair-wise swapping of mapped cores. It also proposes traffic splitting that considers the mapping problem together with the possibility of splitting traffic among various paths. For various benchmark applications, NMAP produces better results than the reported mapping algorithms before it.

A tool, SUNMAP, has been presented in [70] to automatically select the best standard topology for a given application and producing a mapping of cores onto that topology. It minimizes the average communication delay, area, power dissipation subject to bandwidth and area constraints. MOCA, a two phase heuristic for low energy mesh based on-chip interconnection architecture has been proposed in [71], to reduce the communication energy considering the bandwidthand latency constraints. In the first phase, the cores are mapped to different routers of the mesh by invoking a bi-partitioning based slicing tree generation technique. In the second phase, it attempts to find a minimal path from source to destination for each traffic trace. It does not give good solution when latency constraints are considered.

All the mapping techniques proposed previously, use communication weighted model (CWM) to account for the overall communication volume of each channel. It does not consider communication timing. To capture both timing of application communication

and communication volume, communication dependence and computation model (CDCM) has been proposed in [72, 73], which maps applications on regular NoC under bandwidth constraint and minimizes average communication delay.

A Simulated Annealing (SA) based application mapping technique has been proposed in [74] for 2-D mesh based NoC which minimizes the area requirement and the maximum bandwidth. It also proposes an efficient routing algorithm which selects a route among alternative paths based on the network state and occupancy of queues. Cluster based techniquecombined with simulated annealing has been proposed in [75] for application mapping onto 2-D mesh-based NoC. In thistechnique, mapping is done cluster-wise, instead of node-wise, to reduce the mapping complexity. Clustering is a technique to partition nodes into groups according to the physical distance among them in the network topology. Clustering exploits the knowledge about the network architecture and communication demand of applications. So in this mapping technique, first cluster-based core to node initial mapping is done and then a simulated annealing technique is applied upon it to find good mapping solution.

Topology design is one of the significant factors that affects the net delay and energy consumption of an application specific NoC. The topology must satisfy the design constraints. For very high I/O rate streaming type of application mapping, a guaranteed and high throughput pipelined mechanism for NoC. In this paper, authorshave proposed a pipeline-based high throughput low energy mapping algorithm which performs task allocation, pipelined task scheduling, and communication scheduling simultaneously on the heterogeneous NoC and minimizes the energy consumption.

Onyx, a new bandwidth constrained application mapping has been presented in [64] to minimize the overall communication cost of NoC. In this technique, a core with the highest communication bandwidth has been mapped at the centre. Then the ranking of other unmapped cores are settled according to the communication volume with mapped cores. The unmapped cores are placed at the nearest possible distance with its related core by looking the lozenge-shape path with one hop or two hop distances and so on till the empty tile is identified (Figure 2.13). In [75], Crinkle, a mapping algorithm has been presented to reduce the overall communication cost. In this technique, priority lists are prepared depending on the interconnection degree of nodes and communication

bandwidth before mapping onto mesh based NoC. Depending on the priority lists, the heuristic maps the tasks from the corner of 2-D mesh platform and ends on another corner in a zigzag manner (Figure 2.14).

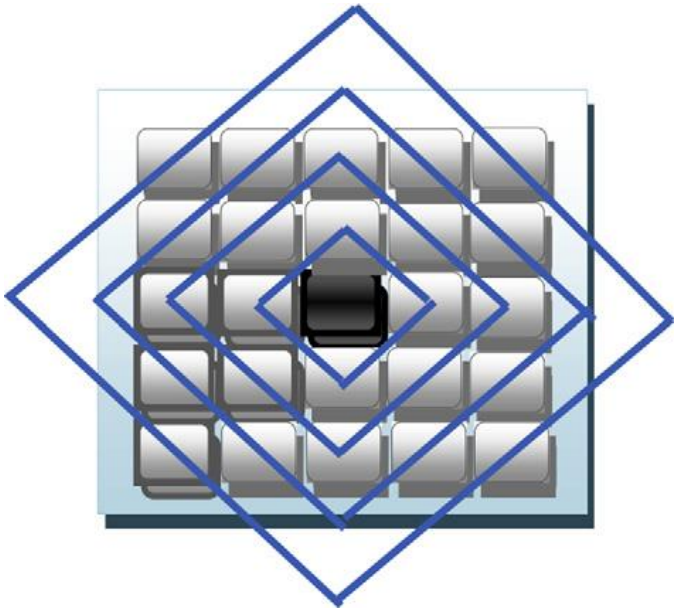

Figure 2.13. Concept of Lozenge-shape path selection [64]

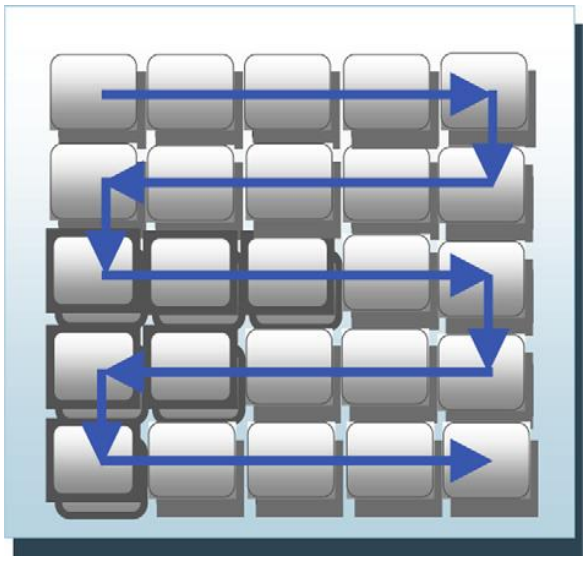

Figure 2.14. Zigzag path for core mapping [77].

TABLE I

TAXONOMY OF THE OPTIMIZATION ALGORITHMS

Algorithm	**Nature**	**Kind**
First Free (FF)	Dynamic	Heuristic
Nearest Neighbor (NN)		
Packing-based Nearest Neighbor (PNN)		
Minimum Maximum Channel Load (MMC), Minimum Average Channel Load (MAC)		
Path Load (PL)		
Best Neighbor (BN)		
Dynamic Spiral Mapping (DSM)		
Low Energy Consumption based on Dependencies-Neighbor (LEC-DN)		
ILP,NILP,MILP	Static	Exact
Genetic Algorithm, Particle Swarm Optimization, Ant Colony Organization, Population-Based Incremental Learning	Static	Heuristic Transformative
Binomial Mapping Algorithm, Constructive Mapping Algorithm, Chain Mapping Algorithm, Mapping on NoC, Simulation environment Mapping, LMAP Algorithm, Simulated Annealing, Onyx, Search Tabu	Static	Heuristic Constructive
Branch and Bound	Static	Deterministic

TABLE II
SUMMARY OF REPORTED MAPPING SOLUTIONS

Ref.	Factor		
	Optimization Criterion (Figures of Merit)	**Common Domain Semantic / Optimization Algorithm**	**Target Architecture and Abstraction Level**
S. Tosum et.al [22]	Execution time	CTG / Exact optimization	Homogeneous architectures and Algorithm abstraction
L. Zhong et. al[57]	Energy Consumption	CTG / Heuristic Constructive	
E. Khajekarimi et. al [58]	Energy Consumption	TG / Exact optimization	Heterogeneous architecture and Algorithm abstraction
Y. Z. Tei et. al [60]	Communication cost	APCG / Heuristic Transformative	
A.Racu et. al[61]	Communication volume	CTG / Heuristic Transformative	
F. Bolanos et. al[62]	Multi objective	TG / Heuristic Transformative	
H. M. Harmanani et. al [63]	Bandwidth, Area	CTG / Heuristic Constructive	
M. Janidarmian et. al [64]	Energy Consumption, Latency	ARG / Heuristic Constructive	
S. Murali et. al [46]	Communication Cost, Bandwidth	CG / Heuristic Constructive	Heterogeneous architecture and TLM abstraction
E. Carvalho et. al [9]	Execution Time	AG / Dynamic	
A. K. Singh et. al [65]	Execution Time Energy Consumption , Average channel load, Latency	AG / Dynamic Optimization	
L. Ost et.al [67]	Energy Consumption	CTG / Dynamic optimization	Heterogeneous architecture and RTL abstraction
G. Wang et. al [68]	Execution Time	SDFG / Heuristic Transformative	
C.A. M. Marcon et.al [69]	Energy Consumption, Execution Time	CWG, CRG / Heuristic Constructive Dynamic	Homogeneous architecture and RTL abstraction

CHAPTER 3

PROPOSED WORK

3.1 Wrapper Design and Test Scheduling

A general problem for SOC test integration consists of the design of test access mechanism (TAM) architecture, that transports test data between SOC pin and core wrapper. Wrapper provides an interface between TAM and the core and can be operated in several modes. So test scheduling is a process that determines the starting and ending time of testing each core in the SOC such that the overall test application time is minimized given TAM architecture and power constraints.

3.3.1 Problem Formulation

Let the SOC design consist of N cores and each core C_i ($1 \leq i \leq N$) has n_i input terminals, m_i output terminals, sc_i internal scan chains.

Given a set of N cores, their specific test patterns and the number of I/O pins for an SOC, design the test schedule with wrapper designs for all wrapper based cores such that overall testing time is minimized.

There are two kinds of wrapper designs, balanced and unbalanced wrapper design. For cores having no internal scan chains (that is, containing input and output pins only), unbalanced wrapper design is preferred since it can obtain a lower test time than the balanced one. Test time T for a wrapper for different TAM widths is given by

$$T_i = \{1+\max(S_i,S_o)\}*P + \min(S_i,S_o)$$

$$T = \sum_{j=1}^{B}\sum_{i=1}^{n} x_{ij}T_{ij}$$

Where P is the number of test patterns and Si (S_o) denotes the length of longest wrapper scan chain used during scan-in (out) for a core.

3.3.2Design_Wrapper and Genetic Algorithm

The wrapper design algorithm used in test scheduling is shown below,

Design_Wrapper procedure:
Wmax= number of TAM connections *NbLines= int(Wmax/2)* *#SC= number of Scan Chains* *Process* *Sort the internal scan chain in decreasing length order* *Select the (NbLines) longest scan chains as the (NbLines) lines* *While (#SC>NbLines)* *Chain the longest line with the shortest scan chain* *Update #SC* *Update length of the longest of the longest scan chain* *Sort scan chains in decreasing length order* *End Process* *Add functional I/Os to balance the scan chains* *End* *Calculate the test time for the core.*

The Design wrapper algorithm is used on each core of the d695 and p93791 benchmark circuits of ITC'02 benchmark.

The test scheduling algorithm is developed based ongenetic algorithm. The general integrated wrapper/TAMco-optimization and test scheduling problem that weaddress is as follows:

Given the test set parameters for the cores of the SOC,as well as the total TAM width, an optimalassignment of cores to each TAM, is determined, such that the overall testingtime is minimized.

Genetic algorithms (GA) are stochastic optimizationsearch algorithms based on the mechanics of naturalselection and natural genetics. The genetic formulation ofour problem involves the careful and efficient choice ofthe following.

1. A proper encoding of the solutions to formchromosomes.
2. To decide upon a crossover operator.
3. To identify a proper mutation operator.
4. A cost function measuring the fitness of thechromosome in a population.

Chromosome Representation

The chromosome consists of 3 parts, The first part known as the partition part is the bandwidth assigned to the TAM wire, The second part known as distribution part gives the distribution of each core among the TAMs. This is an array of integers consisting of 1and 2.

The j^{th} entry of this array defines to which TAM is the core-j is assigned. The length of the array is equal to the number f cores in the Benchmark circuit. The third part is the test time for this core distribution.

TAM Partition		Core Assignment										Time
5	11	1	1	1	1	2	2	2	1	2	1	42523

Genetic Operators

a) Crossover

GA formulation has been biased towards selecting the chromosomes with better fitness to participate in crossover. For this purpose, the whole population is sorted according to their fitness values. A certain percentage of population with better fitness value is defined to be the "Best Class". To select a chromosome participating in cross-over, first a uniform random number between 0 and 6 is generated. If the number is greater than 3, a chromosome form the "Best Class" is selected randomly. Otherwise, a chromosome from the entire population is selected. After selecting two chromosomes to participate in crossover, a single point crossover is applied to the distribution part.

b) Mutation

Mutation brings effectiveness into the chromosomes introducing newer search options. A chromosome is randomly selected from the (Total Population – Best Class).

c) Fitness Measure

Fitness of the chromosome is measured in terms of the cost of a solution, which is the total time required to test all cores in the system.

3.3.3 Memory constraint

Minimum TAM width directly leads to the minimization of the ATE channels used , Test scheduling is performed such that "idle" bits appearing between core tests on ATE channels are move d to the end of each channel [80].

These bits can be classified as:

Type-1 idle bits: These idle bits are used to fill in space between separate core test sets on ATE channels. These are due to certain TAM wires being idle while others are transporting test data during the schedule. The idle bits are transported to the idle TAM wires to equalize testing time on all TAM wires. In terms of rectangle packing, these correspond to the empty spaces between rectangles in the bin. They can be minimized by more efficient rectangle packing.

Type-2 idle bits: These idle bits appear within the boundaries of a core test set if Pareto-optimal TAM widths are not chosen, resulting in wasteful TAM widths supplied to cores. Type-2 idle bits correspond to the extra area at the top of rectangles of non-Pareto-optimal widths. These bits can be completely eliminated by assigning only Pareto-optimal TAM widths to cores, as we have done in this paper.

Type-3 idle bits: These idle bits appear as a result of the idle time introduced by unbalanced wrapper scan chains. These bits cannot be removed unless all wrapper scan chains are of equal length. Thus, for a TAM width of 1, there are no Type-3 idle bits in the schedule. These bits correspond to the space within rectangles that can be removed only if equal-length wrapper scan chains are created.

After the Genetic Algorithm has produced the arrangement of cores, it if found that these bits are moved to the end of each channel, thus minimizing the ATE channels used.

3.2 NoC Application Mapping

With the development in the chip integration technology and the amount of data communication between the Intellectual Property (IP) cores, Network on Chip (NoC) is becoming the widely accepted solution for the complex SoC (System on Chip) architecture [48]. NoC has become a new paradigm for the design of future SoCs [49, 50, 51]. Mapping of an application on to the NoC architecture determines the placement of IP cores of a SoC to the routers of NoC architecture. The placement aims at optimizing the parameters of concern like energy consumption and delay. Mapping is an NP-complete problem, meaning exhaustive search algorithms cannot be used to generate the optimal solution.

Energy minimization can be achieved by reducing the distance between the communicating cores. As shown in Figure 3.1. Tile based NoC architecture contains a grid of regular tiles where each tile can be a general purpose processor, a memory, a DSP etc. A mesh network is preferred for a tile based architecture as the network wires are structured and wired beforehand, their electrical parameters can be easily controlled and optimized. This makes it easy to use aggressive signaling circuits to reduce the power dissipation and propagation delay significantly. Modularity and standard network interfaces in a mesh network makes a module reusable and interoperable.

We propose a novel application mapping scheme to reduce the overall energy consumption and delay. A Partition algorithm is used to distribute the cores into different partitions according to their test times. Then Genetic Algorithm is used to determine the optimized mapping of IP cores to the NoC tiles.

Hereafter, Section 3.4 gives a brief introduction of the NUT (Network on chip under test) and the associated problem. Section 3.5 presents the NoC mapping scheme with the optimization of embedded IP cores testing.

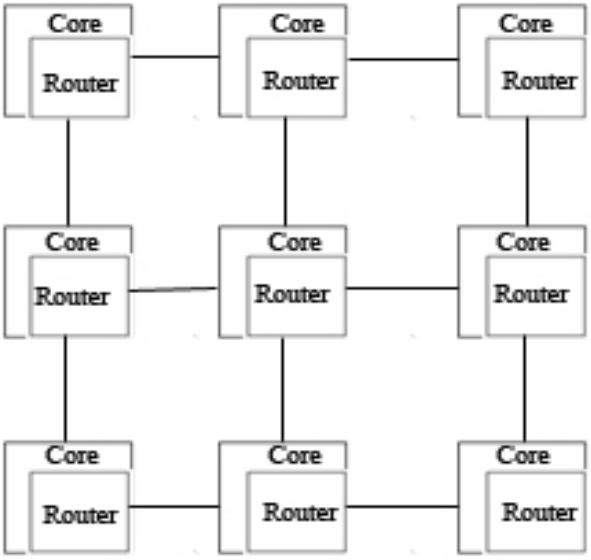

Figure 3.1. 3×3 Mesh Network.

3.2.1 Problem Description and Mesh Network

The NoC architecture can be represented as an undirected connected graph *G(N,L)*, where $N=\{n_1,n_2,n_3 \ldots n_i\}$ and $L=\{l_1,l_2,l_3, \ldots l_i\}$ are set of nodes and links respectively. A mesh network is used to represent the NoC architecture as it has good scalability and is easy to construct. The NoC problem can be formulated using the following definitions:

Definition 1: Given directed acyclic weighted graph *G*(*V*, *E*)as application characteristics chart, each vertex $v_i \in V$ shows the associated IP core, directed arc $e_{i,j} \in E$ shows the communication relationship between $v_i$and v_j, coefficient $\omega_{i,j}$shows the traffic between $v_i$and v_j.

Definition 2: Given directed graph *P*(*R*, *P*)as NoC structure feature graph, each vertex $r_i \in R$ shows resource node, directed arc $p_{i,j} \in P$ shows the router between $v_i$and v_j, $E(p_{i,j})$ shows the power consumption of one bit transfer between $v_i$and v_j.

Definition 3: Power consumption functions are defined as (1) and (2):

$$\text{Energy} = \sum_{\forall e_{i,j}} \omega_{i,j} \times E(p_{i,j}) \quad (1)$$

$$E(p_{i,j}) = n_{router} \times E_{Rbit} + (n_{router} - 1) \times E_{lbit} \quad (2)$$

$n_{router}$is the number of routers, E_{Rbit} and E_{Ibit} are separately power consumption of routers and interconnections. The E_{Rbit} and E_{Ibit} are respectively set to 0.43pJ and 5.445pJ respectively according to [52]. Considering the 2D Mesh architecture and XY routing algorithm, $E(p_{i,j})$is actually determined by the hamming distance between $v_i$and v_j, so that it can be induced as in equation 3.

$$\text{Cost} = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \omega_{i,j} \times \text{distance}_{i,j} \quad (3)$$

Among them, *Cost* is the mapping overhead, $\omega i, j$ is the traffic between $v_i$and v_j, $distance_{i,j}$ is the hamming distance between v_i and v_j, and the NoC is N×M 2D Mesh architecture.

NoC mapping problem description: Given two graphs G and P, a search for mapping function map() is done, for minimizing the Cost and satisfying the following constraints.

$$\forall v_i \in V \Rightarrow map(v_i) \in R \quad (4)$$

$$\forall v_i \neq v_j \Leftrightarrow map(v_i) \neq map(v_j) \quad (5)$$

$$size(G) \leq size(P) \quad (6)$$

Y. Zhang et. al [59] provides various ways of accessing the NoC network through ATE, it also gives a performance evaluation of several ATE access modes. In the best approach proposed by Y. Zhang in [59], The Network under test (NUT) is approached from all four directions as shown in Figure 3.2. Thus providing opportunity for parallel testing of four blocks and further reducing the test time and the test cost.

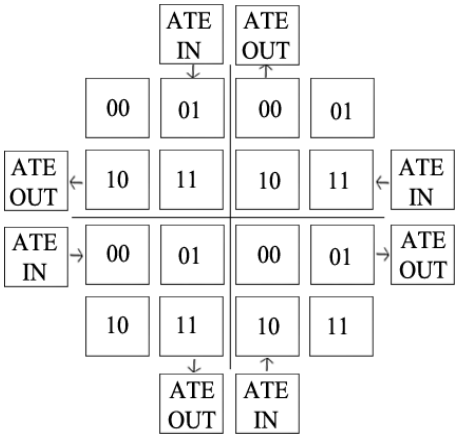

Figure 3.2. ATE access scheme

The core test time optimization problem is effectively reduced to a simple core partitioning problem. Given a set of IP cores C= $\{c_1,c_2,c_3,..c_N\}$,(where, N = Total number of IP cores) with each core having an individual testing time of T_i needs to be partitioned into four partition groups P=$\{p_1,p_2,p_3,p_4\}$ each having a test time of TP_i. The number of cores inside each partition is given by N_p given by equation 8:

$$N_p \leq \lceil N/4 \rceil \tag{8}$$

The objective of the partitioning problem is thus reduced to

$$\text{Min } (TP_i) \text{ where } i = 1, 2, 3, 4 \tag{9}$$

3.2.2 NoC Mapping and Algorithm

For each module, number of test patterns is p_m, number of input signals is i_m, number of output signals is o_m, number of bidirectional signals is b_m, number of scan chains is s_m, length of scan chain k is $l_{m,k}$, total number of scan Flip Flops is $f_m$which can be calculated as:

$$f_m = \sum_{k-1}^{s_m} (l_{m,k}) \tag{10}$$

The number of test stimulus and response for each test vector is given as:

$$s_{in} = i_m + b_m + f_m \tag{11}$$

$$s_{om} = o_m + b_m + f_m \tag{12}$$

The input data and output data of each test vector can be transmitted simultaneously in the scan test except the last one.

Moreover, there is an extra cycle needed for function. Therefore, the independent testing time for an IP core T_c without considering the TAM width is as given in equation 13.

$$T_c = \{1 + \max(s_{in}, s_{om})\} \times p_m + \min(s_{in}, s_{om}) \tag{13}$$

Assuming a TAM width of W, Test time of each core T_c will be

$$T_c = \left\lceil \frac{\max(s_{im},s_{om}) \times p_m + \min(s_{im},s_{om})}{W} \right\rceil + p_m \tag{14}$$

TAM (Test access mechanism) is defined as a set of wires which carry the test patterns and the test stimuli to and from the CUT. In our analysis we have considered the TAM width W=16, 32. The data is transmitted in form of packets consisting of flits. Each flit consumes one clock cycle to transfer. The testing time depends on the number of scan chain being greater than or equal to or less than the transmission bandwidth. Thus the testing times are calculated as follows:

- *IP core having no scan chain*

Number of flits of each packet is

$$N_f = \frac{\max(s_i, s_o)}{w} \tag{15}$$

Where, $s_i = i_m + b_m$, $s_o = o_m + b_m$,

Testing time is

$$T_c = \max\left\{\left\lceil\frac{s_i}{w}\right\rceil, \left\lceil\frac{s_o}{w}\right\rceil\right\} \times p_m + \min\left\{\left\lceil\frac{s_i}{w}\right\rceil, \left\lceil\frac{s_o}{w}\right\rceil\right\} + p_m \tag{16}$$

- *IP core having scan chains*

A. The number of scan chain is much smaller than current bandwidth.

Since the input and output can be transmitted together in the scan chain, so that $N_f = \max(l_{m,k})$ $k=1,2,\ldots,s_m$, testing time is

$$T_c = N_f \times p_m + p_m + N_f$$

B. The number of scan chain is equal to the bandwidth.

The input and output will not be transmitted together in the scan chain, so

$$N_f = N_{f1} + N_{f2}$$

$$N_{f1} = \text{avg}(l_{m,k})\ k=1,2\ldots,s_m,$$

$$N_{f2} = \max\left\{\left\lceil\frac{s_i}{w}\right\rceil, \left\lceil\frac{s_o}{w}\right\rceil\right\}, \text{ and } T_c = T_{c1} + T_{c2}$$

$$T_{c1} = N_{f1} \times p_m + N_{f1} + p_m \tag{17}$$

$$T_{c2} = p_m \times N_{f2} + \min\left\{\left\lceil\frac{s_i}{w}\right\rceil, \left\lceil\frac{s_o}{w}\right\rceil\right\} \tag{18}$$

C. The number of scan chain is greater than the bandwidth, So,

$$N_f = \max\left\{\left\lceil\frac{f_m+s_i}{w}\right\rceil, \left\lceil\frac{f_m+s_o}{w}\right\rceil\right\},$$

Testing time is

$$T_c = p_m \times N_f + p_m + \min\left\{\left\lceil\frac{f_m+s_i}{w}\right\rceil, \left\lceil\frac{f_m+s_o}{w}\right\rceil\right\} \quad (19)$$

According to the above test time calculation scheme, the test times for 4 ITC'02 circuits have been obtained. For example the test times of d695 circuit are displayed in Table III.

After calculating the testing time for each core, the Partitioning algorithm Partition is applied to the cores to partition them into 4 groups so as to reduce the overall testing time of the NoC. Details of the Partition algorithm are given below in Figure 3.3. as:

Partition Algorithm

Step. 1. Sort the IP cores in descending order of their test times.

Step. 2. Define an upper limit to the number of cores in a partition as N_p.

Step. 3. Add the first core to P_0 and next three to P_1, P_2, P_3 respectively.

Step. 4. For the next core of the descending list

- Find the partition with the minimum total test time (TP_i), and number of cores in that partition being less than N_p. Add the core in this partition.
- If the core with minimum total test time (TP_i) has number of cores should be greater than N_p, then find the partition with the second least test time and add the core to this partition if it meets the N_p constraint.

Figure 3.3. The Partition Algorithm

Now that the partitioning of the cores is complete, the mapping of the IP cores to the NoC tiles can be started. The mapping is optimized by the use of Particle Swarm Optimization Algorithm.

3.2.2.1 Particle Swarm Optimization Algorithm

It is an evolutionary algorithm which can be used to optimize results of a NP hard problem. An initial population of particles is created with each particle representing a solution to the problem. The translation of the mapping problem to the PSOalgorithm is done in particles as represented by the Figure 3.4.

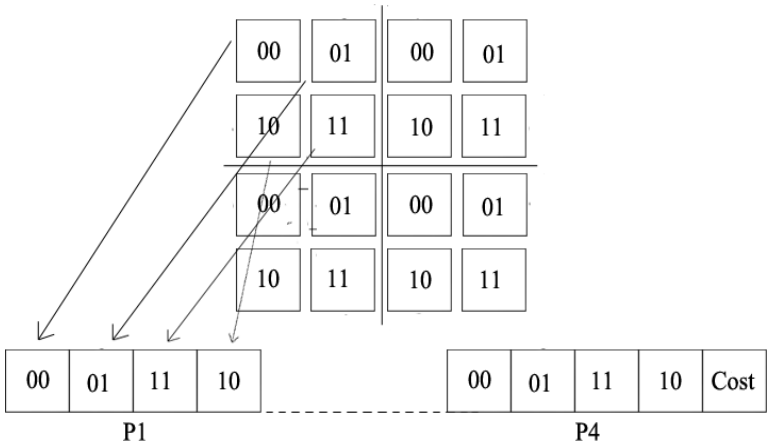

Figure 3.4. Particle representation.

During the generation of particles special constrains are laid such that the IP cores are not mapped to the extreme corners of the partition which would increase the cost of communication with other cores. This previous knowledge of placement generates an initial population which itself gives a near optimal results. This is clear from the Figure 4.5 and Figure 4.6 where the empty tiles are marked with 0 as the number of core placed on them.

With the help of the basic mapping represented by the particles and the traffic information between the cores, the cost of the function is calculated using equation 3. This cost represents the fitness of the particle and is the total energy consumed in the communication process between the cores.

Fitness of a particle *Pi* is equal to the overall communication cost due to association of cores to different routers, as specified by the particle. The communication cost is defined as:

$$\sum_{(u,v)\in set\ of\ cores} Bandwidth(u,v) \times HopCount(u,v)$$

Wherehop count(u, v) is the minimum number of hops between the cores u and v after mapping. Every particle has a local best (p_{best}), which is the permutation of core positions that gives minimum communication cost, among all permutations that the particle has seen so far. For a particular generation, the particle resulting in the minimum communication cost is the global best (g_{best}) for that generation.

- Evolution of Generations

In the first generation, the initial population is created randomly and the fitness values for individual particles are evaluated. The local best (*pbest*) of each particle is taken to be same as the initial particle. The global best (*gbest*) of the first generation is the particle giving the least communication cost (smallest fitness function) in the generation. By exchanging the core positions within the particles randomly, second generation is evolved. If they give better fitness values, the local and the global best values are updated. Further generations are evolved through a series of operations called*swap* operations. The local best of each particle and the global best of a generation are modified if the corresponding values in the current generation are lesser than the values in the previous generation.

- Swap Operator

The position of a core in a particle is represented by its position index. The indexing runs from 0 to N-1(N being the number of routers). Let the swap operator be $SO_{j,k}$ (where, j and k = 0,1, … N-1) that swaps the j^{th} and k^{th} positions of the particle P to create a new particle P_{new}. For example, let us consider the particle P = {1, 3, 5, 7, 4, 6, 8, 2}, where the numbers represent the core numbers of the core graph and the position represents the router number in the topology graph. The swap operator $SO_{4,6}$ swaps the cores at the position 4 and 6, which creates a new particle P_{new}= {1, 3, 5, 7, 8, 6, 4, 2}.

- Swap Sequence

A swap sequence SS is made up of one or more swap operators. The swap operators of the swap sequence are applied in order on the particle P to create a new a particle P_{new}. For example, let the swap sequence SS = {$SO_{4,6}$, $SO_{2,5}$ } be applied on the particle P={1, 4, 3, 6, 2, 8, 5, 7}. It creates a new particle P_{new} = {1, 4, 8, 6, 5, 3, 2, 7}.

To align a particle P_i with its local best, the swap sequence is identified. Let this be $SS_i^{l\_best}$. Another swap sequence is identified to align the particle with the global best. Let this be $SS_i^{g\_best}$. Now swap sequence $SS_i^{l\_best}$ is applied on the particle P_i with a probability α. Let the modified particle be $P_i^{l\_best}$. Then the swap sequence $SS_i^{g\_best}$ is applied on $P_i^{l\_best}$ with a probability of β. This creates a new particle P_i^{new}.

CHAPTER 4

RESULTS AND EVALUATION

The input to the Design_wrapper algorithm used for wrapper design for Test scheduling and NoC mapping is the same and is shown in Figure 4.1.

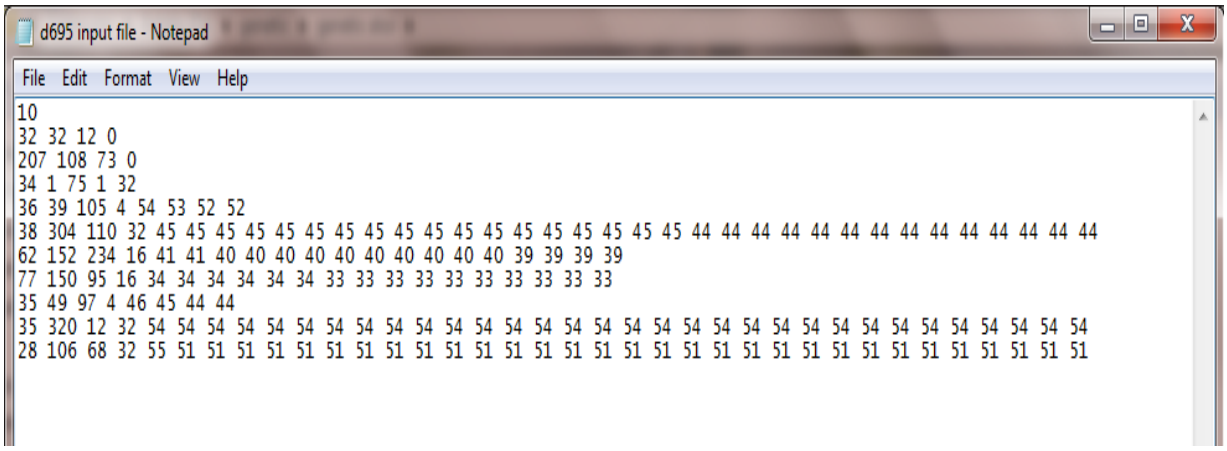

Figure 4.1. Input file of the Design_Wraper algorithm

Where, the first line defines the number of cores in the benchmark

From second line the data specific to cores is defined in the format

- Number of inputs
- Number of outputs
- Test patterns
- Number of scan chains (N)
- Length of N number of scan chains

And the number of TAM wire bandwidth is passed to this function.

The output of the Design_wrapper function is shown in Figure 4.2.

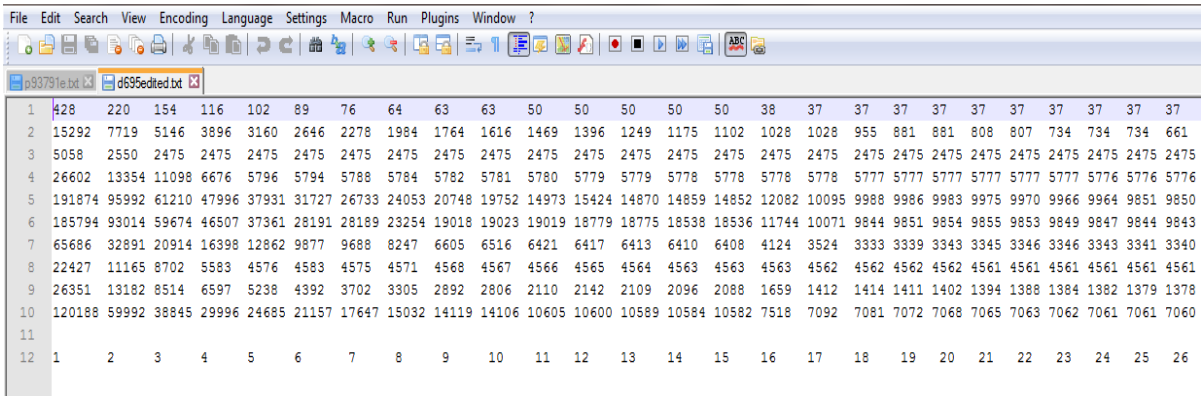

Figure 4.2. The output file of Design_Wrapper algorithm

Here, each row defines the test time for the core(i) and each column defines the TAM bandwidth.

The output of the genetic algorithm when run for Width = 16 to find the optimal test schedule for d695 is as shown in Figure 4.3,

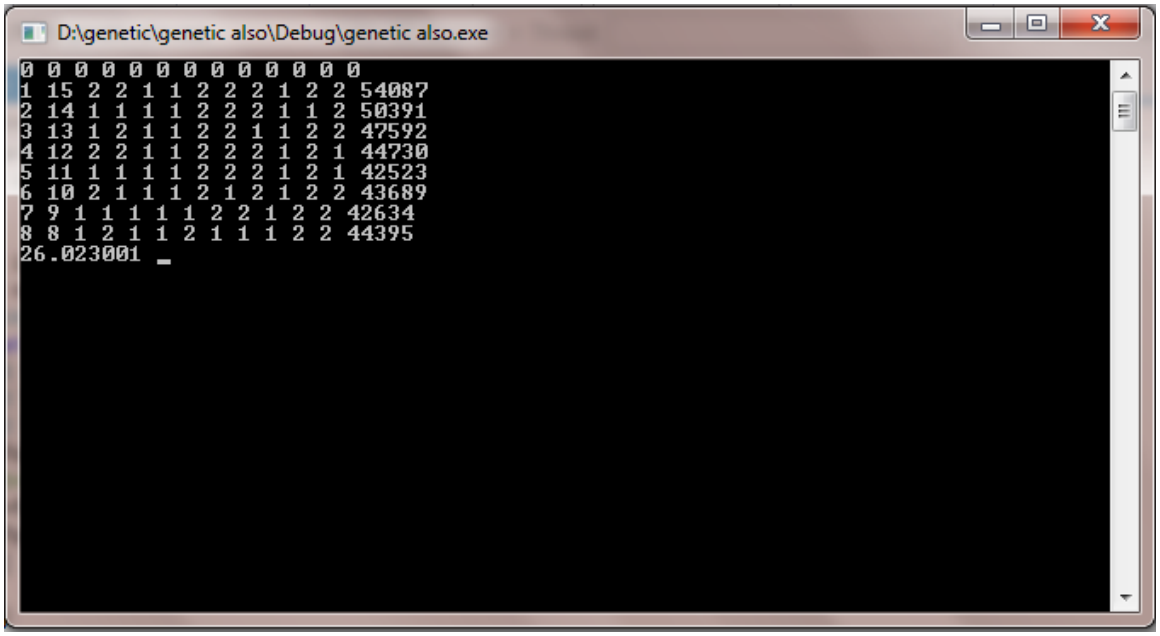

Figure 4.3 Result of the Genetic Algorithm for test scheduling

The best fit chromosome is selected.

TABLE III
TESTING TIME AFTER TEST SCHEDULING OF d695 CIRCUIT

W	B	2D TAM Partition	TAM Partition [79]	Test time	Test time [79]	Execution time in sec
16	2	(5,11)	(7,9)	42523	44225	28.105
24	2	(6,18)	(5,19)	27868	31070	43.417
32	2	(14,18)	(11,21)	24616	28524	57.782
40	2	(8,32)	(19,21)	21133	25761	68.408
48	2	(16,32)	(21,27)	19685	25761	89.301
56	2	(23,33)	(23,33)	17847	23256	94.004
64	2	(31,33)	(23,41)	16986	23232	116.459

TABLE IV
TESTING TIME AFTER TEST SCHEDULING OF p93791 CIRCUIT

W	B	2D TAM Partition	TAM Partition [79]	Test time	Test time [79]	Execution time in sec
16	2	(4,12)	(7,9)	1767909	1869200	8.362
24	2	(1,23)	(7,17)	1186335	1294512	9.761
32	2	(9,23)	(9,23)	877692	966752	13.042
40	2	(17,23)	(15,25)	660186	778480	16.037
48	2	(23,25)	(15,33)	560659	766270	19.063
56	2	(26,30)	(11,45)	418288	686577	21.948
64	2	(26,38)	(15,49)	384967	662198	25.051

In TABLE III and TABLE IV, Genetic algorithm is applied to the d695 and p92791 benchmark circuit [82].

Column 1 gives the test bandwidth available for testing of the SOC. Column 2 gives the number of TAM wires or the number of partitions.Column 3 gives the wire partition between the two TAM wires. Column 4 gives the test time obtained for that distribution.Column 5 and column 6 give the test times when power constraint is introduced and column 7 and 8 give the number of TSV used for that power constraint.

The output of the particle swarm optimization algorithm when run for the benchmark circuit g1024 to find the optimal floor plan of the cores in an NoC architecture, is shown below in Figure 4.4.

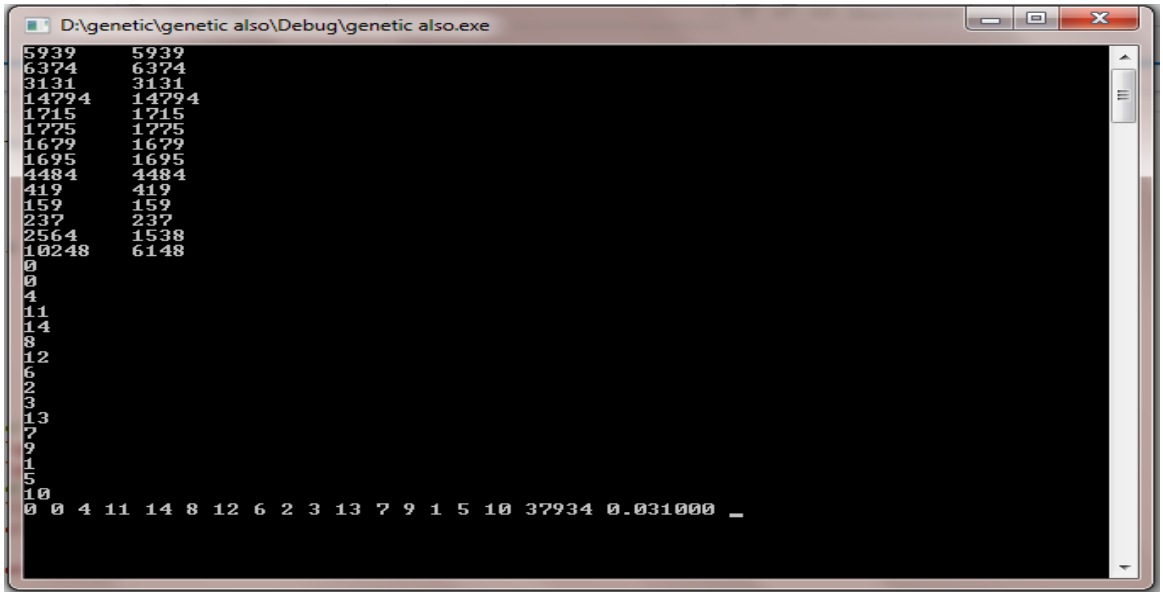

Figure 4.4. The output of the PSO Algorithm when applied on g1024

Here, the first two columns show the test time calculated by the Design_wrapper algorithm after that the partitioning of cores is defined. While in the last line the chromosome with the best fitness function is shown in result.

The results of the proposed partition algorithms are obtained for four ITC'02 benchmark circuits [66] which were introduced by the Duke University, Stuttgart University and Philips and they are d695, g1023, p22810, p93791. Table V, VI, VII and VIII give the results for the test time calculation of the d695, g1024, p22810 and p93791 benchmark circuits.

TABLE V
TESTING TIME OF D695 CIRCUIT

Core	W=16	W=32
	T_c	T_c
1	38	25
2	1029	588
3	2507	2507
4	5829	5829
5	12192	6160
6	11931	9869
7	4254	3359
8	4605	4605
9	1676	838
10	7589	3869

TABLE VI
TESTING TIME OF G1024 CIRCUIT

Core	W=16	W=32
	T_c	T_c
1	5939	5939
2	6374	6374
3	3131	3131
4	14794	14794
5	1715	1715
6	1775	1775
7	1679	1679
8	1695	1695
9	4484	4484
10	419	419
11	159	159
12	237	237
13	2564	1538
14	10248	6148

The results of the Partition algorithm are represented in the Table IX.

The results of Table IX show that the testing of ITC'02 benchmark circuit is reduced as compared to the results in [54]. The testing time is reduced by 16.58% in case of g1023 when W=16. While the test is reduced by 8.2% when W=32 for g1023. An overall reduction of 12.62% and 14.14% is found for W=16 and W=32 respectively when Partition algorithm is used.

TABLE VII
TESTING TIME OF P22810 CIRCUIT

Core	W=16	W=32
	T_c	T_c
1	1075	627
2	99428	99428
3	49299	36974
4	12434	9325
5	1113	668
6	30651	43644
7	4276	2850
8	15796	10530
9	10433	7825
10	26574	21647
11	3899	3899
12	8454	8454
13	7801	7801
14	209	209
15	8065	8065
16	3077	3077
17	989	989
18	2339	2339
19	44504	44504
20	2595	2595
21	9749	9749
22	87141	87141
23	4679	4679
24	4755	4755
25	2855	2855
26	33694	39311
27	132856	72981
28	104	104
29	2726	2726

TABLE VIII
TESTING TIME OF P93791 CIRCUIT

Core	W=16	W=32
	T_c	T_c
1	179574	90196
2	771	578
3	2594	1945
4	178	95
5	73534	42894
6	322654	166436
7	532	355
8	532	355
9	771	578
10	20960	11644
11	2814	1501
12	114053	57222
13	120114	60251
14	120114	60251
15	1155	866
16	3964	2378
17	89833	45133
18	255	170
19	66668	33544
20	200151	100492
21	255	170
22	171	128
23	115114	57806
24	9217	6145
25	8065	5377
26	387	290
27	180647	90782
28	3172	1982
29	72827	36500
30	771	578
31	2249	1227
32	55520	30844

TABLE IX
COMPARISION OF TESTING TIME

	W=16		W=32	
Circuit	Base Case[60]	Partition algorithm	Base Case[60]	Partition algorithm
g1023	17925	14953	16489	14953
d695	16197	13519	10705	9869
p22810	166800	152965	150921	135108
p93791	502876	456356	333091	232599

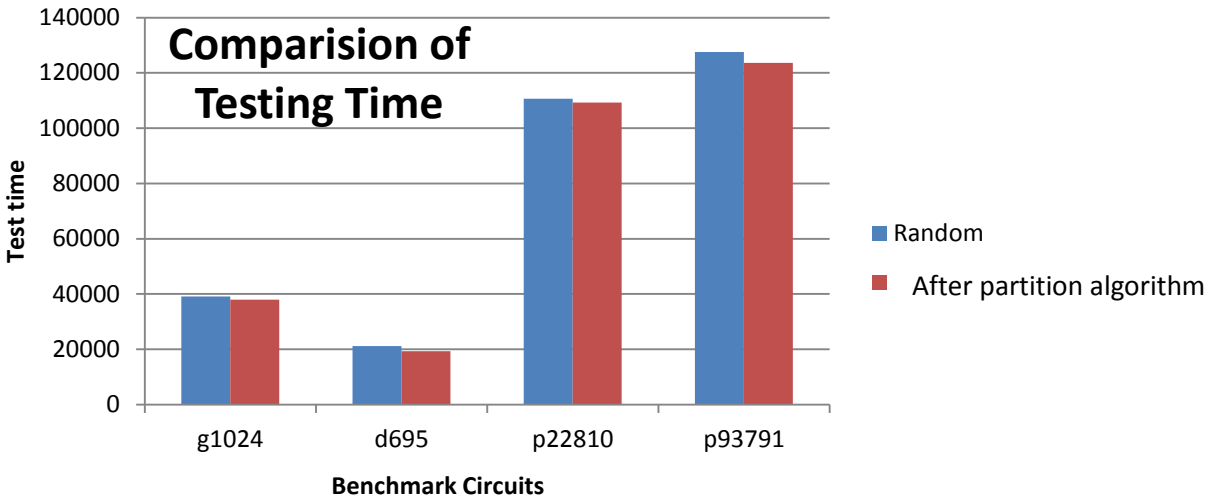

Figure 4.5.Comparision of Testing time

Now for applying the genetic algorithm and calculating the energy consumed, we need the determined traffic information between various IP cores of the ITC'02 benchmark circuits. The four circuits adopt the TGFF randomly generated traffic as shown in [55][56]. Traffic diagram of d695 is shown in Figure 4.6.

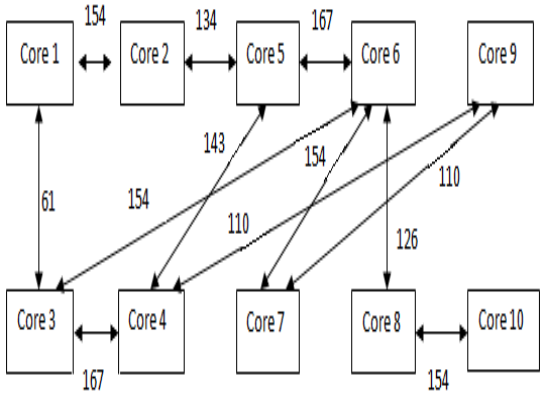

Figure 4.6.Traffic diagram for IP core d695.

Using the traffic information and the partitioning information obtained using the Partition algorithm we can determine a basic mapping of the IP cores to the NoC network as shown in Figure 4.7.

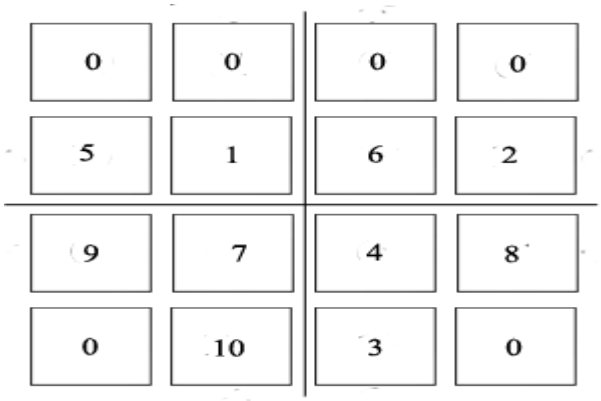

Figure 4.7. Random NoC mapping for d695 for W=16

After applying Genetic algorithm to the given mapping scheme an optimal mapping is obtained as shown in Figure 4.8. having a better cost function and consumes less energy for communication as compared to the random mapping obtained earlier in Figure 4.7.

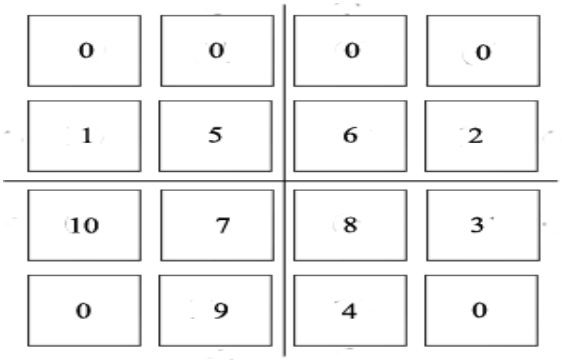

Figure 4.8. Optimized NoC mapping for d695 for W=16.

The results of applying Partition algorithm and the particle swarm optimization algorithm on the ITC'02 benchmark circuits are shown in Table X. The results clearly indicate that our proposed mapping scheme provides better results than the random mapping.

TABLE X
RESULTS OF PROPOSED SCHEME

	W=16		W=32	
Circuit	Random	Optimized Using PSO	Random	Optimized using PSO
g1023	39174	35873	28269	22957
d695	21158	16452	23208	18739
p22810	110672	89637	117730	93892
p93791	127602	104578	141798	109834

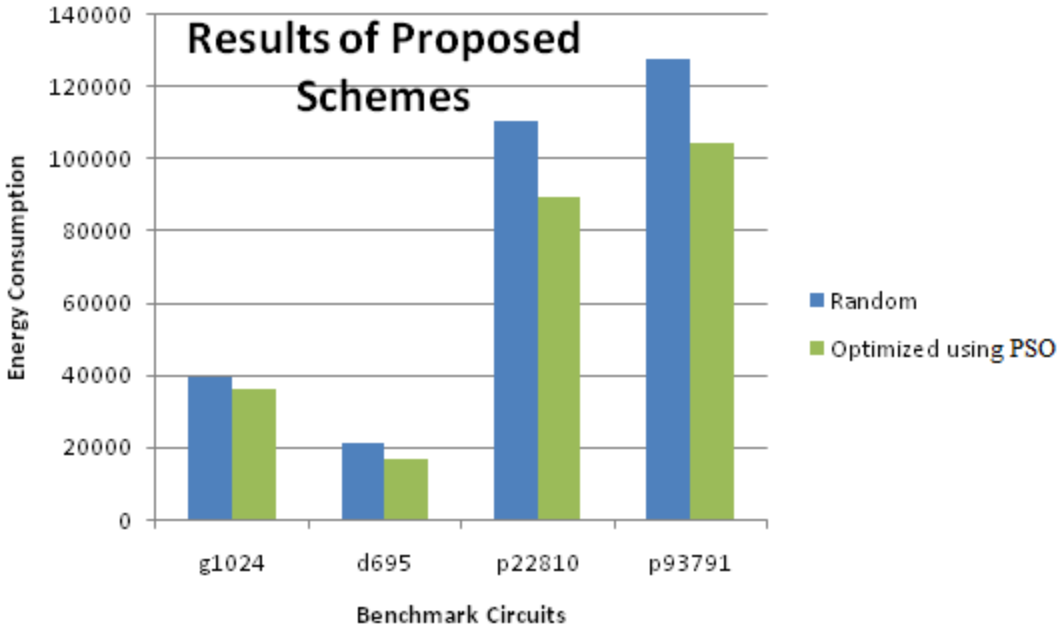

Figure 4.9. Results of proposed Schemes

CHAPTER 5

CONCLUSION AND FUTURE WORK

In this thesis report the problem of test scheduling has been optimized by using the genetic algorithm for a 2D SoC. And the results have been evaluated for two benchmark circuits d695 and p93791. The results show a reduction in test time by an average of 16% as compared to the results in [79].

Particle swarm optimization algorithm for mapping an application onto NoC architecture is also developed. The work done in this thesis aims at reducing the testing time and providing an efficient floor planning of the Intellectual Property cores when mapped onto NoC architecture which can result in less energy consumption during communication between the cores.

The proposed scheme consists of a partitioning algorithm which reduces the testing time of the circuit by dividing it into 4 groups which can be parallel tested by the application of test vectors from an ATE. This scheme results in a test time reduction on an average of 18% in case of bandwidth = 16 and of 20% when bandwidth is 32.

The Particle Swarm Optimization algorithms is applied on 4 benchmark circuit i.e. d695, g1024, p22810 and p93791. A reduction of 15% is achieved in case of Particle Swarm Optimization over a random mapping.

The future work in field of test scheduling can be done to include various hierarchy and 3D SoCs where the cores are present at different levels.

The future work in the application mapping could include developing other strategies of mapping other than the ones used here which may also include a thermal aware mapping scheme which considers the heat generated by the cores while developing a floor plan for the circuit. Also mapping algorithm for the 3D mesh network can be developed in next phase.

REFERENCES

[1] L. Benini, G. De Micheli, "Network on Chips: a new SoC paradigm", In IEEE Computers, 2002 , pp. 70-78.

[2] W.J. Dally, B. Towles, "Route packets, not wires: on-chip interconnection networks," In Proceedings of the 38th Design Automation Conference (DAC), 2001, pp. 684-689.

[3] S. Kumar, A. Jantsch, J.P. Soininem, M. Forsell, M. Millberg, J. Oberg, K. Tiensyrja, A. Hemani, "A Network-on-Chip Architecture and Design Methodology", In Proceedings of ISVLSI, 2002, pp. 117-124.

[4] U.Y. Ogras, J. Hu, R. Marculescu, "Key research problem in NoC Design: a holistic perspective," In Proceedings of IEEE International Conference on Hardware/Software Co-design and System Synthesis, 2005, pp.69-74.

[5] R. Marculescu, U.Y. Ogras, L.S. Peh, N.E. Jerger, Y.Hoskote, "Outstanding research problems in NoC design: system, micro-architecture and circuit perspective," In IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems, Vol 28, 2009, pp. 03-21.

[6] A. Agarwal, C. Iskander, R. Shankar, "Survey of NoC architecture and contributions," In Journal of Engineering, Computing and Architecture, Vol 3, 2009.

[7] R. Pop, S. Kumar, "A survey of techniques of mapping and scheduling applications on NoC systems," ISSN 1404-0018, Research Report 04:4, School of Engineering, Jonkoping University, 2004.

[8] G. Chen, F. Li, M. Kandemir, "Compiler directed application mapping for NoC based chip microprocessors," In Proceedings of LCTES, 2007, pp. 155-157.

[9] E. Carvalho, N. Calazans, F. Moraes, " Heuristics of dynamic task mapping in NoC based heterogeneous MPSoCs," In IEEE International workshop on rapid system prototyping (RSP), 2007, pp.34-40.

[10] E. Carvalho, N. Calazans, F. Moraes, "Dynamic task mapping for MPSoCs," In IEEE design and test for computers, 2010, pp. 242-255.

[11] C.L. Chou, R. Marculescu, "Incremental run time application mapping for homogenous NoCs with multiple voltage levels," In ACM International Conference on Hardware/Software co-design and system synthesis, 2007, pp. 161-166.

[12] C.L. Chou, R. Marculescu, "User aware dynamic task allocation in NoC," In Proceedings of Design, Automation and Test in Europe (DATE),2008, pp. 1232-1237.

[13] C.L. Chou, R. Marculescu, U.Y. Ogras "Energy and performance aware incremental mapping for NoCs with multiple voltage levels" In IEEE Transactions on Computer aided Design for Integrated circuits and systems, Vol 10, 2008, pp. 1866-1879.

[14] M. Mandelli, L. Ost. E. Carara. G. Guindani, T. Gouvea, G. Medeiros, F.G. Moraes, "Energy aware dynamic task mapping for NoC based MPSoCs," In Proceedings of ISCAS,2011, pp.1676-1679.

[15] M.Mandelli, A. Amory, L. Ost, F.G. Moraes, "Multi-task dynamic mapping onto NoC based MPSoCs," In Proceedings of 24th symposium on Integrated Circuits and System Design, 2011, pp. 191-196.

[16] A. Weichslgartner, S. Idermann, J. Teich, "Dynamic decentralized mapping of tree structured applications on NoC architectures," In IEEE/ACM International symposium on NoCs, 2011, pp. 201-208.

[17] A. Bender, "MILP based task mapping for heterogeneous multiprocessor systems," In Proceedings of International conference on Design and Automation (EURO-DAC), 1996, pp. 190-197.

[18] C. Rhee, H. Jeong, S. Ha, "Many-to-many core switch mapping in 2-D mesh NoC architecture," In IEEE conference on computer design: VLSI in computers and processors (ICCD), 2004, pp. 438-443.

[19] K. Srinivasan, K.S. Chatha, G. Konjevod, "Linear programming based techniques for synthesis of NoC architectures," In IEEE transactions on VLSI systems 14, 206, pp. 407-420.

[20] C. Ostler, K.S. Chatha, “An ILP formulation for system level application mapping on network processor architecture,” In Proceedings of Design Automation and Test in Europe (DATA), 2007, pp. 1-6.

[21] O. Ozturk, M. Kandemir, S.W.Son, “An ILP based approach to reduce energy consumption in NoC based CMPs,” In IEEE International symposium on low power electronics and design (ISLPED), 2007, pp. 411-414.

[22] S. Tosun, O. Ozturk, M.Ozen “An ILP formulation for application mapping onto NoCs,” In International conference on Application of information and communication technologies (AICT), 2009, pp. 1-5.

[23] S. Tosun, “Clustered based application mapping method for NoC,” In Journal of advances in Engineering software, 2011, pp. 868-874.

[24] J.Hu, R. Marculescu, “Energy aware mapping for tile based architectures under performance constraints,” In Asia and South Pacific Design Automation Conference (ASP-DAC), 2003, pp. 233-239.

[25] J.Hu, R. Marculescu, “Exploiting the routing flexibility for energy/performance aware mapping of regular NoC architectures,” In Proceedings of Design Automation and Test in Europe (DATE), 2003, pp. 688-693.

[26] J.Hu, R. Marculescu, “Energy and performance aware mapping for regular NoC architectures,” In IEEE Transactions on Computer Aided Design of Integrated circuits and Systems, 2005, pp. 551-562.

[27] T. Lei, S. Kumar, “A two-step genetic algorithm for mapping task graphs to a network on chip architecture,” In Proceedings of the Euromicro Symposium on Digital System Design (DSD), 2003, pp. 180–187.

[28] W. Zhou, Y. Zhang, Z. Mao, “An application specific NoC mapping for optimized delay,” In IEEE International Conference on Design and Test of Integrated Systems in Nanoscale (DTIS), 2006, pp. 184–188.

[29] G. Ascia, V. Catania, M. Palesi, “Multi-objective mapping for mesh-based NoC architectures,” In ACM International Conference on Hardware/Software Codesign and System Synthesis, 2004, pp. 182–187.

[30] G. Ascia, V. Catania, M. Palesi, “Multi-objective genetic approach to mapping problem on Network-on-Chip,” Journal of Universal Computer Science 2006, pp. 370–394.

[31] A.H. Benyamina, P. Boulet, “Multi-objective mapping for NoC architecture,” Journal of Digital Information Management, 2007, pp. 378–384.

[32] R.K. Jena, G.K. Sharma, “A multi-objective evolutionary algorithm based optimization model for Network-on-Chip synthesis,” In IEEE International Conference on Information Technology (ITNG), 2007, pp. 977–982.

[33] K. Bhardwaj, R.K. Jena, “Energy and bandwidth aware mapping of IPs onto regular NoC architectures using multi-objective genetic algorithms,” In International Symposium on System-on-Chip (SOC), 2009, pp. 27–31.

[34] I. Kennedy, R.C. Eberhart, “Particle swarm optimization,” In Proceedings of IEEE International Conference on Neural Networks, NJ, 1995, pp. 1942–1948.

[35] W. Lei, L. Xiang, “Energy- and latency-aware NoC mapping based on discrete particle swarm optimization,” In Proceedings of IEEE International Conference on Communications and Mobile Computing, 2010, pp. 263–268

[36] A.H. Benyamina, P. Boulet, A. Aroul, S. Eltar, K. Dellal, “Mapping real time applications on NoC architecture with hybrid multi-objective algorithm,” In International Conference on Metaheuristics and Nature Inspired, Computing, 2010, pp. 1–10.

[37] P.K. Sahu, P. Venkatesh, S. Gollapalli, S. Chattopadhyay, “Application mapping onto mesh structured Network-on-Chip using particle swarm optimization,” In IEEE International symposium on VLSI (ISVLSI), 2011, pp. 335–336.

[38] K. Wang, L. Huang, C. Zhou, W. Pang, “Particle swarm optimization for traveling salesman problem,” In Proceedings of the Second International Conference on Machine Learning and Cybermetics, 2003, pp. 1583–1585

[39] Yuhui Shi, Russell Eberhart, “Parameter Selection in Particle Swarm Optimization,” Springer Berlin/ Heidelberg, vol. 1447/1998, 2006, pp. 591-600

[40] J. Wang, Y. Li, S. Chai, Q. Peng, “Bandwidth-aware application mapping for NoC-based MPSoCs,” Journal of Computational Information Systems 7, 2011, pp. 152–159.

[41] N. Koziris, M. Romesis, P. Tsanakas, G. Papakonstantinou, "An efficient algorithm for the physical mapping of clustered task graphs onto multiprocessor architectures," In Proceedings of 8th Euro PDP, 2000, pp. 406–413.

[42] A. Hansson, K. Goossens, A. Radulescu, "A unified approach to constrained mapping and routing on Network-on-Chip architectures," In IEEE/ACM International Conference on Hardware/Software Codesign and System, Synthesis (CODES+ISSS), 2005, pp. 75–80.

[43] S. Saeidi, A. Khademzadeh, A. Mehran, "SMAP: An intelligent mapping tool for network on chip," In International Symposium on Signals, Circuits andSystems (ISSCS), 2007, pp. 1–4.

[44] R. Mehran, S. Saeidi, A. Khademzadeh, A.A. Kusha, "Spiral: a heuristic mapping algorithm for network on chip," IEICE Electronics Express 4, 2007, pp. 478– 484.

[45] T. Shen, C.H. Chao, Y.K. Lien, A.Y. Wu, "A new binomial mapping and optimization algorithm for reduced-complexity mesh-based on-chip network," In Proceedings of NOCS'07, 2007, pp. 317–322.

[46] S. Murali, G. De Micheli, "Bandwidth constrained mapping of cores onto NoC architectures," In Proceedings of Design, Automation and Test in EuropeConference and Exhibition (DATE), vol. 2, 2004, pp. 896–901.

[47] S. Tosun, "New heuristic algorithm for energy aware application mapping and routing on mesh-based NoCs," Journal of System Architecture 57 (2011) 69–78.

[48] T.Bjerregaard, S.A. Mahadevan. "Survey of Research and Practices of Network-on-Chip," *ACM Computing Surveys*, vol. 38, no. 1, 2006, pp. 1-51.

[49] Axel Jantsch, and Hannu Tenhunen (Editors), *Networks on Chip*, Kliwer Academic Publishers, Boston, USA, January 2003.

[50] D.Winhard, "Micronetwork Based Integration for SoCs", Proceedings of the 38th Design Automation Conference, ACM/IEEE, Las Vegas, Nevada, USA, June 2001, pp.673-677.

[51] Willian J. Dally, and Brian Towles, "Route Packets, Not Wires: On-Chip Inteconnection Networks", Proceedings of the 38th Design Automation Conference, ACM/IEEE, Las Vegas, Nevada, USA, June 2001, pp.684-689.

[52] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[53] T.Ye,L. Benini and G. De Micheli, "Analysis of Power Consumption on Switch Fabrics in Network Routers," in Design Automation Conference, 2002. Proceedings. 39th, 2002,pp. 524-529.

[54] C. Liu, J. Shi, E. Cota et al. "Power-aware test scheduling in Network-on-Chip Using Variable-Rate on-Chip Clocking," in Proc. 23th IEEE VLSI Test Symposium. 2005, pp. 349-354.

[55] Y. Zhang, N. Wu, F. Ge, X. Chen," Novel NoC mapping scheme optimized for testing time," Proceedings of world congress on Engineering and computer science, 2014.

[56] P.R.Dick, L.D. Rhodes, W. Wayne. "TGFF: Task Graphs for Free," in Proc. 6th International Hardware/Software Codesig Workshop, 1998, pp. 97-101.

[57] L. Zhong, J. Sheng, M. Jing, Z. Yu, X. Zeng, D. Zhou. "An Optimized Mapping Algorithm Based on Simulated Annealing for Regular NoC Architecture," IEEE 9th International Conference on ASIC (ASICON), Oct. 2011, pp. 389-392.

[58] E. Khajekarimi, M. R. Hashemi. "Energy-Aware ILP Formulation for Application Mapping on NoC Based MPSoCs," 21st Iranian Conference on Electrical Engineering (ICEE), May 2013, pp. 1-5.

[59] Y. Zhang, N.Wu, F.Ge. "The Co-Design of Test Structure and Test Data Transfer Mode for 2D-Mesh NoC," IAENG Trans. On Engineering Technologies, 2013, pp. 171-184.

[60] Y. Z. Tei, M. N. Marsono, N. Shaikh-Husin, Y. W. Haut "Network Partitioning and GA Heuristic Crossover for NoC Application Mapping," IEEE International Symposium on Circuits and Systems (ISCAS), May 2013, pp.1228- 1231.

[61] A. Racu, L.S. Indrusiak, "Using Genetic Algorithms to Map Hard Real-Time on NoC-based Systems," 7th International Workshop on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC), Jul 2012, pp. 1-8.

[62] F. Bolaños. Mapping Techniques for Embedded Systems Design with Reliability Considerations. Tesis de Doctorado Universidad de Antioquia 2012.

[63] H. M. Harmanani, R. Farah. “A Method for Efficient Mapping and Reliable Routing for NoC Architectures with Minimum Bandwidth and Area,” Joint 6th International IEEE Northeast Workshop on Circuits and Systems and TAISA Conference, NEWCAS-TAISA, June 2008, pp. 29-32.

[64] M. Janidarmian1, A. Khademzadeh, M. Tavanpour. “Onyx: A new heuristic bandwidth-constrained mapping of cores onto tile-based Network on Chip,” IEICE Electronics Express, vol.6, 2009, pp. 1–7.

[65] A. K. Singh, T. Srikanthan, A.Kumar,W.Jigang.“Communication-aware heuristics for run-time task mapping on NoC-based MPSoC platforms,” Journal of Systems Architecture: the EUROMICRO Journal, vol. 56, Jul 2010, pp. 242-255.

[66] E.J. Marinissen, V. Iyengar, K. Chakrabarty, “A Set of Benchmarks for modular testing of SoCs,” Proceedings of IEEE Test Conference, 2002, pp.519-528.

[67] L. Ost, G. Almeida, M. Mandelli, E. Wachter, S. Varyani, G. Sassatelli, L. S. Indrusiak, M. Robert, F. Moraes. “Exploring Heterogeneous NoC-based MPSoCs: from FPGA to High-Level Modeling,” Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC), 6th International Workshop, June 2011, pp. 1-8.

[68] G. Wang, W. Gong, B. DeRenzi, R. Kastner. “Ant Colony Optimizations for Resource and Timing Constrained Operation Scheduling,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 26, , Jun 2007, pp. 1010-1029.

[69] C.A.M. Marcon, E. I. Moreno, N. L. V. Calazans, F.G. Moraes. “Comparison of network-on-chip mapping algorithms targeting low energy consumption,” Computers & Digital Techniques, IET, vol. 2, Nov 2008, pp. 471-482.

[70] S. Murali, G. De Micheli, “SUNMAP: a tool for automatic topolog selection and generation for NoCs,” In Proceedings of 41st Design Automation Conference(DAC), 2004, pp. 914–919.

[71] K. Srinivasan, K.S. Chatha, “A technique for low energy mapping and routing in Network-on-Chip architecture,” In IEEE International Symposiun on LowPower Electronics and Design (ISLPED), 2005, pp. 387–392.

[72] C. Marcon, N. Calazans, F. Moraes, A. Susin, I. Reis, F. Hessel, "Exploring NoC mapping strategies: an energy and timing aware technique," In Proceedings of Design, Automation and Test in Europe Conference and Exhibition (DATE), vol. 1, 2005, pp. 502–507.

[73] C. Marcon, A. Borin, A. Susin, L. Carro, F. Wagner, "Time and energy efficient mapping of embeded applications onto NoCs," In Proceedings of Asia andSouth Pacific Design Automation Conference (ASP-DAC), vol. 1, 2005, pp. 33– 38.

[74] H.M. Harmanani, R. Farah, "A method for efficient mapping and reliable routing for NoC architectures with minimum bandwidth and area," In IEEEInternational Workshop on Circuits and systems and TAISA Conference, 2008, pp. 29–32.

[75] Z. Lu, L. Xia, A. Jantsch, "Cluster-based simulated annealing for mapping cores onto 2D mesh Networks on Chip," In Proceedings of Design and Diagnostics of Electronic Circuits and Systems (DDECS), 2008, pp. 1–6.

[76] E. Larsson. "Introduction to Advanced SoC Test Design and Optimization", In Frontiers of electronic testing, Springer, 2005.

[77] S. Saeidi, A. Khademzadeh, F. Vardi, Crinkle: a heuristic mapping algorithm for network on chip, IEICE Electronics Express 6 (24) (2009) 1737–1744.

[78] A. Sehgal, V. Iyenger, K. Chakrabarty "SOC Test Planning Using Virtual Test Access Architecture," In IEEE Transactions on VLSI Systems. Vol. 12, pp. 1263-1276.

[79] Yibo Chen ; Chakrabarty, K. ; Yuan Xie,"Test-access mechanism optimization for Core-based three-dimensional SOCs", in Microelectronic Journal 41, pp. 601-615, 2010.

[80] Iyengar, V., Goel. S.K., Marinissen. E.J., Chakrabarty. K. "Test Resource optimization for multi-site testing of SoCs under ATE memory depth constraint," In Proceedings of International Test Conference, 2002, pp.1159-1168.

[81] R. M. Chou, K. K. Saluja and V. Agrawal, "Scheduling Tests for VLSI Systems under Power Constraints", in IEEE Transactions On Very Large Scale Integration (Vlsi) Systems, Vol. 5, No. 2, pp. 175-189, 1997.

[82] N. Dewan, P. Aggarwal, H. Vohra “Test time and power optimization of 2D SoCs using Genetic and Greedy Algorithm”, In Journal of power electronics and power systems (Accepted), 2015.