1-1-2011

Multi-objective Tabu search based topology synthesis for designing power and performance efficient NoC architectures

Anita Tino
Ryerson University

Follow this and additional works at: <http://digitalcommons.ryerson.ca/dissertations>

Part of the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Tino, Anita, "Multi-objective Tabu search based topology synthesis for designing power and performance efficient NoC architectures" (2011). *Theses and dissertations*. Paper 729.

This Thesis is brought to you for free and open access by Digital Commons @ Ryerson. It has been accepted for inclusion in Theses and dissertations by an authorized administrator of Digital Commons @ Ryerson. For more information, please contact bcameron@ryerson.ca.

MULTI-OBJECTIVE TABU SEARCH BASED TOPOLOGY SYNTHESIS FOR DESIGNING POWER AND PERFORMANCE EFFICIENT NOC ARCHITECTURES

By

Anita Tino
Bachelor of Engineering
Ryerson University, 2009

A thesis
presented to Ryerson University
in partial fulfillment of the
requirements for the degree of
Masters of Applied Science
in the Program of
Electrical and Computer Engineering

Toronto, Ontario, Canada

© Anita Tino 2011

I hereby declare that I am the sole author of this thesis of dissertation.

I authorize Ryerson University to lend this thesis or dissertation to other institutions or individuals for the purpose of scholarly research.

Anita Tino

I further authorize Ryerson University to reproduce this thesis or dissertation by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Anita Tino

Multi-Objective Tabu Search Based Topology Synthesis for Designing Power and Performance Efficient NoC Architectures

Anita Tino
Masters of Applied Science, 2011
Program of Electrical and Computer Engineering
Ryerson University

ABSTRACT

Network-on-Chip (NoC) communication interconnects have emerged as a solution to complex heterogeneous core systems such as those found in Multiprocessor System-on-Chip architectures. Many previous works have used the objectives of power or performance during topology synthesis for regular or application specific based NoC design. However, given the crucial requirements and demands of future on-chip applications, it is imperative that designs consider both power and performance aspects, in addition to other important system constraints. Therefore, in order to address such issues, this thesis work presents a multi-objective Tabu search based topology synthesis technique for designing power and performance efficient Network-on-Chip architectures. The methodology incorporates an analytical and simulation approach in order to compromise between computational time and effort within the algorithm. Furthermore, this work also presents a novel approach for a power and performance tradeoff during contention and deadlock removal within synthesis. The proposed method was tested using seven different multimedia and network benchmark application, where results displayed an increase in performance and decrease in power dissipation in comparison to other previous application specific and regular mesh designs. The analysis method was successful during topology generation, yielding an overall accuracy rate deviation of 19.8% within the worst case scenario.

ACKNOWLEDGMENT

I would like to extend my sincerest appreciation and thanks to my supervisor, Dr. Gul N. Khan, for his guidance and support during my graduate studies while pursuing the M.A.Sc degree. I also wish to acknowledge the financial support of Ryerson University (FEAS), the National Science and Engineering Research Council of Canada (NSERC), and the Canadian Microsystems Corporation (CMC) during this thesis work.

TABLE OF CONTENTS

1. INTRODUCTION

1.1.	Background	1
1.2.	Research Motivation and Contribution	3
1.3.	Thesis Outline.....	4

2. NETWORK-ON-CHIP: CONCEPTS AND COMMUNICATION ARCHITECTURE

2.1.	Network-on-Chip: Concept Introduction.....	6
2.2.	NoC Architecture.....	9
2.2.1.	Topology Selection: Network Layer.....	9
2.2.2.	Routing Schemes: Transport Layer	10
2.2.3.	Packet Switching Techniques.....	11
2.2.4.	Flow Control: Data-Link Layer	12
2.2.5.	Power and Performance Constraints.....	13

3. NOC DESIGN OVERVIEW

3.1.	Power and Performance Based Topology Generation With an Optimization Approach.....	19
3.1.1.	Simulated Annealing	20
3.1.2.	Genetic Algorithms.....	21
3.1.3.	Related Heuristic Methods.....	23
3.2.	Performance Degradation in NoC Design	24
3.3.	Power-Aware Topology Generation	26
3.4.	Problem Statement.....	28

4. TABU SEARCH OPTIMIZATION

4.1.	Tabu Search Background	30
4.2.	Tabu Search Algorithm.....	31
4.2.1.	Memory Structures in Tabu Search	32
4.3.	Advantages and Disadvantages of TS.....	34
4.4.	Tabu Search and Topology Generation.....	35

5. POWER AND PERFORMANCE EFFICIENT NOC DESIGN

5.1.	Input Model	37
------	-------------------	----

5.2.	Analytical Performance Modeling.....	38
5.2.1.	Performance Problem Formulation.....	38
5.2.2.	Deadlock-Free Model.....	41
5.2.3.	Contention Model.....	42
5.3.	Performance Modeling.....	44
5.3.1.	LQN Models and Performance.....	44
5.3.2.	LQN Modeling.....	45
5.3.2.1.	Router-to-Core Modeling.....	46
5.3.2.2.	Router-to-Router Model.....	46
5.4.	Power Modeling	47
5.5.	VC Insertion: The Predicament	48
5.5.1.	Case Study: A Simplistic Example.....	48
5.6.	NoC Topology Generation & Optimization	51
5.6.1.	Synthesis Flow	51
5.6.2.	Initial TS Based Topological Solution.....	52
5.6.3.	Problem Formulation	53
5.6.4.	Neighbourhood Selection.....	54
5.6.5.	Virtual Channel Insertion Technique.....	57
5.6.6.	Complexity Analysis	59
5.7.	Method Limitations.....	60
5.8.	Conclusion	61

6. EXPERIMENTAL RESULTS

6.1.	Benchmark Applications.....	63
6.2.	Test Setup.....	70
6.3.	Topological Power and Performance Comparison.....	71
6.4.	Contention Analysis.....	86
6.5.	Deadlock Analysis & Removal Technique.....	90
6.6.	Conclusion of Results	90

7. CONCLUSION

7.	Conclusion	92
	Publications.....	94
	References.....	94
	Appendix.....	96

LIST OF TABLES

5.1 LQN Performance Model Conversion	46
6.1 Benchmark Summary.....	70
6.2 Comparison of Methods Using Normalized Metrics	71
6.3 Area Comparison.....	71
6.4 d26_media Benchmark Core Descriptions.....	73
6.5 Set Top Box Benchmark Core Descriptions	75
6.6 MWD Benchmark Core Descriptions	77
6.7 A/V Benchmark Core Descriptions	79
6.8 Layer-3 Switch Benchmark Core Descriptions	81
6.9 NCS1 Benchmark Core Descriptions	82
6.10 MPEG-4 Decoder Benchmark Core Descriptions	85
6.11 Power/Performance Tradeoff for Contention Analysis	89
8.1 Definition of Terms Used in Tabu Topology Generation Formulation.....	95
8.2 Execution Run Times for Benchmark Topology Generation	96
8.3 Power Dissipation Breakdown	96
8.4 Frequency of Operation for Final Topological Solution	97

LIST OF FIGURES

2.1 NoC Architecture Organization	7
2.2 Standard Router/ Switch Configuration	8
2.3 Power Breakdown in NoCs	16
4.1 The Four Principle Dimensions of Tabu Search Memory	32
5.1 Core Graph Example	38
5.2 TSG Example	38
5.3 4-Port Router Configuration Model	40
5.4 Cycle in LRG	41
5.5 Eliminating Cycle in LRG	41
5.6 LQN Model Example	44
5.7 Inter-Router LQN Model	47
5.8 (a) Initial Router and Topology Contention Example	49
5.8 (b) VC Insertion at Router-to-Router Connection	49
5.8 (c) VC Insertion at Destination Port Connection	49
5.9 Power vs Performance for Figures (a), (b) and (c)	50
5.10 Overall NoC Topology Synthesis Flow	51
5.11 Successive Filter Strategy Example	56
5.12 Successive Filter Strategy Temporary Solutions $N(s)$	56
6.1 MPEG-4 Decoder Core Graph	63
6.2 Layer-3 Switch Core Graph	64
6.3 A/V Benchmark Application Core Graph	65
6.4 NCS1 Benchmark Core Graph	66
6.5 MWD Application Core Graph	67
6.6 Set-Top Box Application Core Graph	68
6.7 d26_Media Application Core Graph	69
6.8 Tabu Based Topology Solution for d26_media (B1)	72
6.9 Application-Specific [53] Topology Solution for d26_media (B1)	72
6.10 Mesh Based Topology for d26_media (B1)	72

6.11 Tabu Based Topology Solution for Set Top Box (B2).....	74
6.12 Application-Specific [21] Topology Solution for Set Top Box (B2).....	74
6.13 Mesh Based Topology for Set Top Box (B2).....	74
6.14 Tabu Based Topology Solution for MWD (B3).....	76
6.15 Application-Specific [6] Topology Solution for MWD (B3).....	76
6.16 Mesh Based Topology for MWD (B3).....	76
6.17 Tabu Based Topology Solution for A/V (B4).....	78
6.18 Application-Specific [6] Topology Solution for A/V (B4).....	78
6.19 Mesh Based Topology for A/V (B4).....	78
6.20 Tabu Based Topology Solution for Layer-3 Switch (B5).....	80
6.21 Application-Specific [48] Topology Solution for Layer-2 Switch (B5).....	80
6.22 Mesh Based Topology for Layer-3 Switch (B5).....	80
6.23 Tabu Based Topology Solution for NCS1 (B6).....	82
6.24 Mesh Based Topology for NCS1 (B6).....	82
6.25 Tabu Based Topology Solution for MPEG-4 Decoder (B7).....	84
6.26 Application-Specific [6] Topology Solution for MPEG-4 Decoder (B7).....	84
6.27 Second Application-Specific [48] Topology Solution for MPEG-4 Decoder (B7).....	84
6.28 Mesh Based Topology for MPEG-4 Decoder (B7).....	85
6.29 ASIC2 Slave Sub-Network Power/ Performance Tradeoff	87
6.30 CMEM3 Slave Sub-Network Power/ Performance Tradeoff	87
6.31 VC Resource Insertion Comparison for Deadlock Removal	90

1. INTRODUCTION

1.1. BACKGROUND

As Multiprocessor platforms persist to extend the applicability of Moore's Law, silicon technologies continue to grow in both size and complexity. The System-on-Chip (SoC) design process has been employed to cope with the increasing demands of low power, high-performance, and integration density in deep sub-micron embedded technologies. More advanced SoCs integrate multiple processors when a single processors will not suffice, commonly referred to as Multi-Processor System-on-Chip (MPSoC) architectures. MPSoCs integrate cores such as microprocessors, Digital Signal Processors (DSPs), memory blocks including RAM and FLASH, dedicated hardware accelerators, and external interfaces (i.e. USB, SPI, and UART standards) onto a single chip. MPSoCs are predominantly used in multimedia and networking communication fields which require large information transmission between cores. It has become evident however that bus-based SoCs which are expected to function at a gigascale level are limited in their ability to efficiently interconnect cores and accommodate the respective communication and energy requirements [1]. Consequently, bus-based systems can no longer withstand the computational demands of future SoCs, where the Network-on-Chip (NoC) concept has been proposed to overcome such design challenges and meet the various constraints of future SoCs.

NoCs adopt and apply the basic concepts of multi-processor systems and wide-area networks to the on-chip domain. NoCs replace the busses in SoCs with routers and links, where packets communicate simultaneously between cores through a multi-hop path, improving overall system performance. Power consumption in the MPSoC is also reduced by using shorter links as opposed to long shared busses. One of the most important challenges in NoC design is realizing the most efficient topological mapping of cores onto the chip. NoC architectures emphasize modularity and are often oriented towards supporting heterogeneous implementations [2]. Therefore many designs solely consider regular, tile-based topologies such as mesh or torus, assuming that NoC systems contain homogeneous cores, where in fact many high performance SoCs make use of heterogeneous cores. As a result, designs containing non-uniform core sizes do not match the regular, tile-based floorplan of standard topologies [3]. For the majority of SoCs, it is known that sizes ranging from small to state-of-the art systems can be designed with static (or semi-static) mapping of tasks to cores, and hence the communication traffic characteristics of the SoC can also be obtained statically [4]. Scalability of regular topologies is therefore not as significant as other factors that SoC designers must maintain. As a result, research has provided evidence that the application-specific NoC architecture is superior to regular topologies in terms of power consumption, performance, and on-chip resources [5][6].

NoC synthesis is initiated subsequent to mapping application tasks to their respective processing cores in the target architecture. The communication topology is generated based on the analytical information provided during the refinement of design constraints. NoC designers use analytical models or simulations tools to design on-chip networks which meet the constraints specified by the target application. The success of this methodology however depends critically on the availability of adequate power and/or performance analysis tools that provide meaningful feedback to guide the design process [7]. Simulation techniques are often time consuming as the design space of the NoC can be demanding in both size and complexity. Therefore, analytical models are often used to generate solutions in a fraction of the time, while also taking effort to derive complicated models for various network components and scenarios. Simulations are very accurate and compensate for the disadvantages of analytical models

including dynamic system effects. These simulation advantages however do not entirely compensate for the disadvantages of the analytical approach.

Previous works in NoC design have dealt with generating topologies with minimal power dissipation and/or maximum performance. However, when taking actual SoC applications into consideration, focusing strictly on power can imply failure to meet the performance constraints of a system. Similarly, concentrating solely on maximizing system performance can lead to an unwarranted amount of power dissipation. Many simulations and analytical models do not account for latencies such as arbitration, NI packing/de-packing delays for read/write operations, contention delays, and utilization within the network. When evaluating for factors of power, automated techniques often focus on simulating power consumption and area overhead while disregarding performance related factors in the system [8][9]. For these reasons, it is clear that a tradeoff between simulations and analytical modeling is needed to compromise between complexity and accuracy, while simultaneously employing a power and performance tradeoff to meet the various constraints imposed during the design phase.

1.2. RESEARCH MOTIVATION AND CONTRIBUTION

The main focus in this thesis work is to develop a technique which combines multiple system constraints and objectives to design efficient, low power, high-performance application-specific NoCs. Objectives such as the power consumption of various on-chip components, in addition to latencies and dynamic effects must be taken into consideration in order to meet the system constraints of the target application. The gap between the analytical and simulation approach must also be filled in order to compromise between the advantages of both techniques and allow for both a reasonable amount of time and effort during the design phase. Therefore, this thesis works addresses these issues and presents a methodology to design a power and performance efficient NoC synthesis tool while employing both an analytical and simulation tactic.

The thesis work accomplishes its goals by employing the following contributions:

- Application of a multi-objective Tabu search optimization method for topology generation which is able to incorporate multiple objectives of power, performance, latencies, and dynamic effects
- A simple, yet novel deadlock and contention avoidance method which compromises between power and performance during virtual channel insertion.
- Tabu search and integrated deadlock/contention method which employs both the analytical and simulation based approaches to execute a faster, in-depth analysis of the system during topology generation.

The overall methodology presented in this thesis will demonstrate that designing multi-objective NoC with power and performance efficiency for future SoC architectures is well within the reach of on-chip designers.

1.3. THESIS OUTLINE

Prior to providing a detailed analysis of the proposed work, the design and communication architecture parameters of NoCs must be established. Hence, background of these design issues is provided in Chapter 2. The chapter concludes by discussing issues of power and performance, where the analytical modeling approach in NoC architectures which will be addressed. Chapter 3 provides an in-depth analysis of the Tabu search optimization method. An explanation describing the Tabu method and its ideal application in topology generation is discussed with a comparison to other heuristics. Subsequent to achieving an in-depth analysis of the NoC concept, Chapter 4 outlines a survey of related work and existing approaches within the area of NoC research and design. Specifically, previous works in optimization, power, and performance based topology generation are discussed, where the various techniques advantages and disadvantages are observed and noted.

Once sufficient background information on NoCs and previous work has been provided, the main contribution of this thesis is presented. Chapter 5 presents a detailed analysis of the proposed power and performance efficient NoC design method, outlining the design objectives and problem formulation for topology generation. In addition, the power and performance modeling and simulation tools are described in detail, where the optimized multi-objective Tabu synthesis technique is given. The test setup and experimental results are given in Chapter 6, comparing the generated topologies to mesh based and other previous application-specific works. The result section is able to analyze and clearly indicate the benefits of our Tabu-based topology generation methods used to meet the target application requirements. Finally, Chapter 7 concludes the thesis report, re-instating our contribution to NoC design, followed by the Appendix and referenced work sections.

2. NETWORK-ON-CHIP: CONCEPTS AND COMMUNICATION ARCHITECTURE

Having developed a background on the general limitations posed by Network-on-Chip (NoC) design, this chapter provides a detailed introduction on the subject, outlining various design aspects and considerations for on-chip design. This chapter also describes the various network architectures, switching techniques, and routing algorithms used in the on-chip communication infrastructures, establishing a basis for the direction of NoC design. Finally, an overall tradeoff summary of the network performance and power evaluation scheme is given for all NoC design selections discussed.

2.1. NETWORK-ON-CHIP: CONCEPT INTRODUCTION

With an increase in the amount of cores posed by future SoC based applications, the NoC concept has replaced the traditional bus-based interconnects to provide on-chip systems with high bandwidth, low power, area, and the ability for scalability. Dedicated components are used to provide data transfers and synchronization between the processing elements accordingly. The general design point for discussion is NoC implementations which are distinguished by employing different methods of information transmission, namely packet or circuit switching.

Figure 2.1: NoC Architecture Organization

In circuit switching, a dedicated link between communicating cores is established, where the connection is reserved solely for specified source and respective destination nodes [10]. Therefore, an increase in point-to-point connections for the cores is also needed within the system which requires more connection wiring, essentially effecting the power consumption and scalability of the network. Due to this reason, packet switching is often employed in NoCs for more complex systems. Packet switching routes packets from source to destination using a multi-hop routing path connected by links and routers. This approach offers more scalability in addition to a power conscience system implementation. Issues for packet switching techniques also arise however when addressing routing and flow control strategies needed to transmit data. This factor will be further discussed throughout the next subsections. In general, a dedicated Network Interface (NI) is used to convert messages to an appropriate format recognized by both the network and the core [10]. The terms for converting information are referred to “packing” for data which needs to be converted at the core to traverse on the network, and “de-packing” for data retrieval from the network to the core. The general Network-on-Chip architecture organization is presented in Figure 2.1.

Figure 2.2: Standard Router/ Switch Configuration

In packet switching, the router is a dedicated component which manages the on-chip communication between various cores and adjacent routers within the system. The routers are independent and thus able to work in parallel within the network, increasing overall performance. There are several different architectures which can be employed for routers given the topology, switch fabric, routing strategy, buffer size, and dedicated virtual channel placement in the network. In regular topologies such as mesh and torus, each router node is positioned in a grid form, and is assigned a specific network ID. In irregular topologies, routers are placed arbitrarily within the layout, where general look-up tables (LUTs) are often employed to forward data packets to the next hop towards its respective destination core. Certain irregular topologies rely on a static routing technique where transmission information is embedded within the packet header, and the router ports contain header decoding capabilities to arbitrate accordingly. Certain on-chip networks store the LUT memory locally within each router, while other networks implement an adaptive routing strategy to monitor the status of the network and route packets accordingly. The general architectural configuration of an on-chip router is presented in Figure 2.2, where each respective input and output port contains a buffer for temporary storage during data transmissions. Control signals for flow control implementation have been omitted for the sake of simplicity.

2.2. NoC ARCHITECTURE

As discussed, there are many design parameters which characterize the network architecture and design. NoC architectures can be classified based on topology, structure, and physical organization of the interconnection network. Many of these aspects will affect the overall topological solution and respective performance, power, and area overhead outcomes for the on-chip network. This thesis work focuses on the architectural and control implementation of NoC design, i.e the transport, network, and data-link layers of NoC design. In order to cover background information on these areas, the topological selection, including regular and irregular topologies will first be briefly discussed, followed by routing methods, packet switching techniques, flow control, and finally properties of network protocols and performance degradation. These topics will then identify the main design issues which will be addressed during this thesis work.

2.2.1. TOPOLOGY SELECTION: NETWORK LAYER

Certain NoC implementations employ regular topologies such as mesh and torus for modularity purposes. Mesh based topologies (also similar to torus) consist of numerous tiles which can be divided into three essential parts: a core, NI, and router. These parts are connected together to form a 2D mesh, where each tile is connected to four adjacent tiles. As applications continue to grow in size and complexity, the size of the mesh grid will also grow linearly with respect to the number of tiles needed to implement the network communication [11]. As discussed previously, cores are often of a heterogeneous nature and do not coincide with regular topologies which assume a homogeneous tile based solution. Therefore, it is clear that the mesh and torus based networks suffer from the disadvantages of power consumption due to links, excessive router resources, area adaptation issues, and performance degradation due to bisection bandwidth imposition [11]. As a result, the irregular topological approach has gained significant attention within NoC research.

Irregular NoCs are designed to the specific communication characteristics of a target application. These types of NoCs often provide a better solution in terms of power, performance, area, and resource costs due to the specific implementation and design of the on-chip network. A challenging problem arises however in attempting to design a tool which can synthesize an optimal NoC for an application with specific design constraints. As will be demonstrated, there is now flexibility in choices for implementation due to the application-specific nature of the NoC with respect to routing, flow control strategies, switching techniques, and guaranteeing the quality of service to avoid performance degradation. Hence, the routing algorithm schemes in NoC design are presented in the next subsection.

2.2.2. ROUTING SCHEMES: TRANSPORT LAYER

The most fundamental routing design aspect is determining the implementation of a static (deterministic) or dynamic (adaptive) routing technique. In order to apply a static route, the paths from a source to its respective destination must be well-defined and permanent in order to guarantee service between cores. In the dynamic routing scheme, transmission and routing decisions are made dynamically during packet traversal given the state of the network and the availability of components. Thus, the anticipated time for a packet to reach its destination can vary. As seen between the two methods, static routing is typically the easiest to implement, however contention and other dynamic effects must be taken into consideration during topology generation or runtime which entails real-time hardware adjustments. Static routing is often employed for irregular, application-specific topologies where constraints and data requirements are known *a priori*. Dynamic routing can be applied to regular topologies, but is not advised in application-specific arrangements due to the one hop nature of the architecture and the inability to allow for an alternate route. Dynamic routing schemes are therefore used in regular architectures as the topological arrangement allows for routing adjustments given the regularity of the grid format. Dynamic schemes are also able deal with performance degradation during routing.

There are two principle ways in which static routing is accomplished, namely routing tables (LUTs) or encoded packet header information as discussed previously. The routing tables often require extra hardware in each router and can account for an increase in both area overhead and power. By employing the header encoding method, area and power are reduced, where the packet sizes are increased to compensate for the extra routing information embedded within the header. In dynamic routing, LUTs can also be implemented in every router to account for the dynamic, adaptive nature of routing the packets for the given status of the network. Various adaptive routing techniques are applied such as XY/Odd-Even (OE), hot-potato (deflective routing), and the turn-model to avoid deadlock, contention, and starvation [12][13][14].

2.2.3. PACKET SWITCHING TECHNIQUES

As discussed previously, the switching techniques of NoC can be narrowed to circuit and packet switching. The advantages of packet switching over circuit have been clearly articulated. The packet technique is predominately used in both the regular and irregular topological architectures. The three well-known switching methods in NoCs are the store and forward (SAF), virtual cut through (VCT), and wormhole (WH) techniques.

The SAF technique is the most straightforward switching method of the three. In this method, a packet will only be forwarded to the next hop if the port's buffer has enough allocated space to store the entire packet. Hence, the buffers within the routers must be the sized to the length of the packet, if not greater, in order to accommodate entire packet traversal. The VCT method is similar to that of the SAF, but divides the packets into smaller units know as 'flits'. Opposed to forwarding the entire packet, the flits are now forwarded to the next hop when buffer space is available, where the other flits follow without delay. Therefore the packet is essentially still being transmitted, however now in smaller units. Thus the buffering requirements remain the same as that of the SAF technique.

The packet switching technique primarily used in irregular topologies is wormhole routing. WH allows individual flits from different packets to traverse the network without the need for an entire packet to be able to fit within the buffer of the next hop. As a result, the WH technique leads to lower buffering requirements and reduced latency in data traversal. In relation to these advantages, there also exist the respective disadvantages. As the flits of a packet can be spread throughout a network, they can spread throughout the routers and in turn causing blockage to other packet flits. In addition, WH routing is also susceptible to deadlock and starvation. Hence, techniques to overcome such performance degradation factors within WH routing must be considered.

2.2.4. FLOW CONTROL: DATA-LINK LAYER

Flow control manages the flow between two devices, namely master and slave components. In an on-chip network atmosphere, the master signifies the router in arbitration needing to transmit to the next hop, where the next hop represents the slave device. The master hop ensures that data packets are only sent to the neighbouring hop when there is enough buffer space allocation available. Flow control can be divided into two methods, namely those methods that require resource reservations for buffers and those that do not. Deflection routing is a non-resource reservation technique used for flow control [13]. In this routing scheme, packets that do not fit into the next hop will be sent back to the initiating router with no intent of conditional packet acceptance. Depending on the routing technique and topological implementation, deflection routing may attempt to transmit data to another adjacent hop. Consequently, this flow control strategy can cause large delays within the system given that traffic rates exist within the network.

In terms of resource reservation, the credit based end-to-end flow control scheme is used to prevent buffer overflow in adjacent hops. In this type of method, the master sends a request signal to the slave for data transmission, where the slave ensures that its resources have space to store a packet/flit. There are essentially three resource reservation flow control protocols used in NoCs: STALL/GO, ACK/NACK, and T-Error. In the STALL/GO protocol, the transmitting hop sends a request to the next

hop. If there is space in the hop's buffer, the slave sends a GO signal in response to the request, confirming that there is space within the buffer. If not, the STALL signal is asserted indicating that the slave hop is not yet ready for transmission. This method provides a compromise between the other two techniques, using a very simple approach which allows for good performance, low power, and low area resources for implementation purposes. Its downfall however occurs with respect to error detection as this implementation assumes that data will be retrieved without error. Despite this fact, data loss is not necessarily problematic in an on-chip domain as data will remain on the link until the GO signal is asserted.

The ACK/ NACK protocol ensures that the data is received without transmission errors, where the ACK signal is asserted if the information is received correctly, and the NACK signifying that retransmission of previous data is needed due to data corruption. Although this type of flow control strategy ensures correct data retrieval, it also requires high powering requirements as the flit retransmission between devices can cause an increase in power consumption. The protocol can also not give a guarantee on performance due to its focus on data integrity. Lastly, the T-Error protocol allows a fast, yet noisy method to transmit packet communication between routers [15]. Therefore, the method uses one of two ways to increase speed: using repeaters on the physical links or by increasing the frequency of operation which often creates timing errors [11]. In addition, performance is also not guaranteed due to the increased frequency of data and its respective need for synchronization, in turn creating self-induced errors.

2.2.5. POWER AND PERFORMANCE CONSTRAINTS

Now that various design parameters within NoCs have been discussed along with the respective advantages and disadvantages, the power and performance related factors are now discussed. These factors are essential during the design phase as the on-chip network must adhere to the various system constraints. In addition, the application of analytical modeling and simulation tools must be addressed to

account for detailed and accurate system assessments during topology generation. Given a target application, there are a certain amount of transactions that must occur between cores within a specified time frame. If these transactions do not take place, performance factors are not met, thus creating degradation which can cause system failure and possible malfunction when applied to real-time system applications. Therefore, it is imperative that these performance parameters are met. The main causes for performance degradation include factors of contention, deadlock, livelock, and starvation.

Contention occurs when packets of different flows contend for the same links causing system blockage. As flits from different packets traverse throughout the network in both deterministic and adaptive routing technique, flits often require the same resources to reach their respective destinations causing contention. Hence, contention frequently leads to an increase in latency and decrease in throughput. Similar to contention, deadlock occurs due to packet blockage, but on the contrary may stay blocked for an indefinite amount of time which packets await for an event that will never occur [11]. Although the advantages of wormhole routing have been clearly stated, deadlock is often a key result in this type of routing protocol due to the spreading nature of flits throughout the network. Deadlock therefore arises when no packet can advance, as the next resource depends on the next, etc.

Livelock is due to cyclical dependencies. Packets are redirected from one hop to the next through the network, where the packets that experience livelock do not reach their final destinations. Generally in adaptive routing cases, deflection routing techniques such as hot potato allow packets to continually be deflected if the neighbouring routers persist to be blocked. The final type of performance degradation, starvation, occurs when a packet does not reach its destination due to a lack of access from its master. Starvation can occur in systems which exhibit a perpetual denial of access to resources, most common in router arbitration where one port continues to take priority over other ports and the packets continue to endlessly wait. This can occur when the higher priority flits in on-chip networks continuously take precedence over those of lower priority. As a result of these degradation factors, constraints and dynamic effects must be addressed in advanced to ensure data transmission for proper system functionality.

There are various ways to deal with issues of performance degradation. Contention can be addressed in multiple ways depending on the routing strategy employed. For instance, dynamic routing schemes avoid contention by implementing deflection routing in regular topologies. However, this dynamic nature can also lead to deadlock or livelock if the packets are deflected to another router in an endless cyclic nature. Other methods also include injecting packets at a slower rate to reduce the blocking probability. However, given that traffic is generated with specified temporal patterns provided by the target application (i.e. the amount of bandwidth in Mb/s that must be transmitted to the source and destination), this method does not stand as a probable solution. Scheduling packets globally is also another implementation for contention avoidance, often referred to as contention-free routing or Time Division Multiplexing (TDM). Although this technique is able to construct networks which avoid contention, the average latency per packet is increased and as a result relatively high due to the added delays. Virtual channel (VC) insertion is also implemented to avoid contention, deadlock, and livelock, by providing more than one output path per channel where the probability of blockage is decreased and the overall throughput is increased [15].

Deadlock is most prone in WH networks. Dally proposed a channel dependency graph based on a topological configuration to analyze and determine the cyclical channel dependencies within a network [16]. Cycles are broken by inserting an additional channel within the cycle. Deadlock can also be avoided in NoCs by employing a restrictive routing method such as the *turn prohibition* (also referred to as the turn model) technique [11]. The technique imposes a prohibition on certain turns to break dependency cycles while maintaining connectivity between every pair of routers. This technique is widely used in regular network such as mesh and torus due to its grid like structure. In an application-specific topology, turns can not be restricted given the reduced hop count and irregular structure of the chip. Therefore, the VC insertion technique is the preferred choice and often employed in irregular topologies, or topologies which use a WH routing strategy to overcome such issues of deadlock.

Livelock solutions are resolved by using priority assignments in packets which is mostly applied within deflection routing schemes in regular topologies. Livelock is prevented by incorporating minimum length paths from source to destination cores in irregular topologies, effectively allowing for minimal hop count. Starvation in routers is averted by incorporating a round-robin arbitration scheme amongst the ports to guarantee equal transmission probability. The round-robin approach may not apply however to methods which employ priority, such as deflection routing and real-time systems. Thus, careful design must be taken into consideration and implemented accordingly.

Figure 2.3: Power Breakdown in NoCs [17]

The most critical constraint considered in NoC design is power consumption. As MPSoC designs continue to increase in complexity, the associative bandwidth constraints also increase in turn demanding more of the interconnection network [17]. The most significant power dissipation in NoCs is in relation to the router and wirelength in the layout of the system. A detailed analysis of the router power consumption breakdown is provided in Figure 2.3. As seen in the figure, the most significant power consumption experienced in the router is clocking, amounting to 33% of the overall dissipation. Therefore, the frequency of operation in the on-chip network is important, greatly effecting power consumption during the design phase analysis. Second to clocking is the FIFO buffer, accounting for 22% of the total router consumption. As anticipated, the VCT and SAF switching techniques can add a significant amount of power due to the size of the buffers needed for packet/ flit traversal. Hence, WH is often chosen due to low buffering requirements, and therefore lower power consumption and an overall decrease in latency. FIFO buffers can be designed in an SRAM or register implementation. Register design is simplistic and

allows for decreased arbitration delays. On the contrary, SRAM is a slightly more complex implementation, where flits are accessed based on an address as opposed to a first-in, first-out based concept. SRAM also consists of an average two cycle delay for write operations. An advantage to the SRAM type however is its low area overhead and slightly less power consumption in comparison to the register implementation. Referring back to Figure 2.3, the third most significant factor is the links dissipation, consisting of 17% of the overall power. Link length is also dependent on the frequency of operation, where power consumption may vary within the system.

It is deduced from both the discussion above and Figure 2.3 that routers can greatly effect power consumption in an on-chip network. Router architectures are also crucial to power and can have a direct influence on the performance aspects of a system. As discussed by Mishra et al [59], the less demanding the performance requirements of a system, the lower the power consumption experienced within the routers. Therefore, as the injection rate of packets in the system increases, the experienced latency during transmission also increases causing congestion and arbitration delays. The frequency is therefore often adjusted to meet the performance demands of the application and account for delays and congestion in the system, often requiring more power to do so. Conversely, increasing the number of ports within a router can also improve the performance of a network as the hop count is reduced during transmission [60]. Additional router ports can however add to link and router contention and has proven to increase on-chip power quadratically due to the tree crossbar connection demand between the I/O ports [61]. As a result, the operational frequency of the router is again tuned (increased) to account for the bandwidth requirements. Thus, it is now apparent that in order to improve performance, designers must closely examine and account for the power dissipation values and other various system constraints that are caused by adjusting certain parameters within the architecture.

It is evident that there are multiple power and performance aspects when designing NoC architectures. The most significant choice to be made is the type of topology, which in turn will determine other parameters in the design layers. It is clear that the irregular, application-specific topology is optimal

in terms of buffer power, area, decreased latency, and the demands of future SoC based architectures. In terms of routing, both static and dynamic have disadvantages and advantages. While static routing is easier to implement at the cost of packet size, one must also consider the dynamic effects within the system. On the contrary, the dynamic approach can combine both dynamic effects and routing factors in one instance. The dynamic routing technique however also entails a cost of area and power for LUT implementation in order to avoid contention and deadlock. Thus, static routing becomes the more attractive method in terms of power, while performance factors can be addressed during the design phase. The preferred switching technique choice for various reasons is wormhole routing as the protocol provides a better outcome in terms of both power and performance in comparison to VCT and SAF. Given the WH selection, the designer must then cope with performance degradation factors of contention and deadlock while deciding on the best approach in handling such effects. In choosing the irregular, application-specific topology, the turn prohibition technique for deadlock avoidance becomes inapplicable. Therefore, VC insertion is often chosen during the design and/or runtime phase to deal with issues of contention alleviation and deadlock elimination.

In conclusion, there are many choices to be made during NoC synthesis. Although one designer may chose to use a certain method over another at the cost of power or performance, there are in fact numerous ways to implement an on chip application. Although many designers employ the regular based architecture for NoCs, there are also many ways to implement an application-specific design. Therefore, now that various design parameters have been discussed, the question is now narrowed down to how such an optimal architecture can be created given a set of NoC design selections.

3. NoC DESIGN OVERVIEW

The basic selections for design parameters in NoC synthesis have now been discussed. Therefore, it is imperative that a review of existing NoC designs are described and analyzed. Attention is primarily focused on topology generation with power and performance design constraints implemented by previous works. This will then identify and evaluate the advantages and disadvantages of various design approaches, in addition to various optimization method applications. Previous performance-aware NoC designs with degradation alleviation approaches are also briefly discussed. Through the discussion of previous works, the underlying areas of refinement and advancements needed to be applied to the area of NoC synthesis will be revealed and applied to this thesis work.

3.1. OPTIMIZATION BASED APPROACHES FOR TOPOLOGY GENERATION

A number of researchers have incorporated power and performance related factors using optimization methods and heuristics for NoC topology generation. The following subsections outline the specific techniques employed in NoC design worth noting.

3.1.1. SIMULATED ANNEALING

The work by Ahonen et al implements a design automation tool entitled OIDIPUS (On-Chip Interconnection Design Interface for Point-to-Point Unidirectional Signaling) to optimize power and latency in communication networks [18]. The tool employs a Simulated Annealing (SA) technique which partitions the communication network into subsystems. During topology generation, the OIDIPUS tool aims at optimizing constraints such as the maximum number of links to and from a node, the maximum word width on the links, and the maximum area to be incurred by the NoC. The tool however can only use a single variable for cost optimization, with a selection of either optimizing for power consumption or latency reduction within the network. In order to find an optimal solution, the SA algorithm partitions the PE blocks into two major parts. The cost for partitioning the network is then minimized through SA, where a ring based topology is created afterwards to support the generated partitions. Although this methodology aims at implementing a SA technique in order to solve the topology generation problem, it also displays that SA and system partitioning degrade the achievable level of optimization through the block placement. The block placement experiences a lack of freedom when determining a position within the topology. By aiming to lower the cost of the partition in the SA, the traffic is increased within the resulting partition. Therefore, the SA technique limits the boundaries in space exploration and within the neighbourhood search space of possible solutions. The OIDIPUS method can only invoke one objective of either power or performance in order to optimize NoC topologies. Furthermore, OIDIPUS does not invoke a floorplanner to obtain actual power values for the links and routers in the network, but instead relies on values estimated by the user.

The work employed by Beraha et al seeks a way to optimize both power and performance by also using a SA optimization technique [19]. The general method aims at attaining the best trade-off between the two factors by generating a topology with minimal power consumption which aims to meet the required latency demands of the application. As opposed to finding the average latency of the entire on-chip network, the method utilizes the functional timing requirements of the application by targeting the

end-to-end traversal delays of each module. Overall, the method incorporates the area of the routers and bandwidth of the links together with latency factors to generate a topological design which meets the two constraints. The SA technique maps the PEs onto a mesh grid in an initial topology, where swaps are then made in a random fashion between two cores to generate another mesh topology with a lower cost. The cost function is the summation of the total area consumed by all the routers and the summation of all the bandwidths incurred by the mapped links. Solutions which do not meet the minimum latency requirements are automatically rejected by the SA algorithm.

Contention analysis for each method is taken care of subsequent to designing the on-chip network. Therefore, when analyzing the throughput and latency requirements during topology generation, the methods do not account for the dynamic effects such as contention in order to accurately determine whether the performance demands are being met. As a result, the minimum performance requirements may not end up meeting the constraints needed by the application. Although the topology generator in [19] was able to find a solution, a method to leverage all the requirements at once was not given. Furthermore, the generator does not maintain accurate values for the power dissipation of the system using floorplanning or power models, but simply relies on the router area and BW of the links to account for the power factor imposed by the cost function.

3.1.2. GENETIC ALGORITHMS

The work implemented by Ascia et al aims at optimizing both power and performance in a multi-objective Genetic Algorithm (GA) optimization method [20]. The GA method maps the cores onto a mesh topology in a random fashion. The generator assumes a static XY, wormhole, mesh based topology. It also assumes uniform traffic between all the cores within the on-chip network. A state machine is employed to assess and simulate the power and performance attributes within each core and router used in the topology. In order to assess the power and performance factors, the cores and routers are divided into states, where a predetermined value for latency and energy are assigned to the router or core, depending

on the type of state that they are currently in. After determining the state, trace files are generated to determine the events and allow for state transitions. The NoC simulator evaluates the mapping alternatives produced by the GA topology generator in terms of power and performance. An exploration engine invoked in the tool evaluates the next produced mapping of the GA to determine whether the move is of a better fitness than the previous solution. The GA technique and exploration engine continue to produce mapping and evaluations until a stopping criteria is met. The multi-objectives mappings are obtained from a Pareto Curve technique. Although the work by Ascia et al aims at designing NoCs with the objectives of power and performance using Pareto a curve, the experimental outcomes represent either a high performance mapping or low energy mapping. Therefore, the work does not combine both objectives when aiming to create an on-chip network, but on the contrary provides the user with a mono-objective topology where the user is expected to choose the best outcome. In terms of power optimization, the work does not assess the effect of link power consumption within the system.

The work of Leary et al design NoC architectures by also using a GA based approach [21]. The work uses static routing to avoid deadlock. In the cases where deadlock cannot be avoided, virtual channels are inserted in the on-chip network for alleviation. The initial populations for the router and nodes are assigned randomly, where the shortest path routine is used to connect the components. The stopping criterion is based on N successive generations, equal to the number of vertices and edges in the communication trace core graph. The cost function is established based on these two properties of the shortest path and inter-router distance. The cost is assigned an infinite value (illegal solution) if the Manhattan distance between two routers violates the maximum allowable distance as specified by the user. In the case where latency or path constraints are violated, the solution is left unmapped and restarts the GA to search for another possible solution. The Pareto curve technique is also implemented based on the factors of power consumption and the amount of routers used in the on-chip network topology, where the designer is again given the choice of selecting the best solution. When the authors compared their methodology to others, they observed that their GA method consumes a slightly higher amount of power. However, the method was successful in minimizing the amount of routers being used at the cost of a

higher dissipation for the power mapping. In general, the GA method randomly generates solutions. However given that information, communication, and system requirements are a priori, there is no logic in continuing to generate random solutions. In addition to these facts, due to the random nature of the GA, the topology generation scheme invoked in this work can lead to the use of longer wires which in turn produces more invalid solutions, leading to high execution times and not necessarily guaranteeing a global or near-optimal solution.

Arjomand et al introduce a queuing performance and energy model to estimate values by also invoking a GA method [22]. Similar to the work of [21], the technique implements a Pareto curve method based on the factors of power consumption or performance in the topology. The work employs a variation of the energy bit model which is successful in a first order estimation, but does not consider effects of traffic congestion which can cause a power estimation model to surpass a 50% error margin [23]. Although the method targets at designing NoCs with the objectives of power and performance, the experimental outcomes represent a performance or power mapping. Hence, the work does not combine both objectives when aiming to create an on-chip network, but instead provides the user with a mono-objective topology.

3.1.3. RELATED HEURISTIC METHODS

Murali et al describe a method for generating NoC topologies using the two objectives of minimum power and/or hop count [24]. Hop count however does not directly address performance factors within the system. Performance degradation factors are considered subsequent to topology generation by employing a mismatch parameter which must be manually fined tuned to satisfy the application's performance level. The heuristic is of a fairly exhaustive nature, accounting for various sets of possible frequency and link widths by invoking a min-cut partitioning method to generate topologies. The frequency values obtained ignore delays present within the NoC such as arbitration, and are therefore only valid for situations which exhibit a lack of contention.

Srinivasan et al [25] define an Integer Linear Programming (ILP) heuristic for topology generation, also minimizing power and hop count. The ILP method invoked uses an objective function to create an aspect ratio between the two factors. The method arranges cores in a static floorplan such that area is minimized and routers are placed at all corners of the layout and connected accordingly to the ILP. Redundant routers which are not connected are eliminated and the final topology is connected to meet the power constraints. The ILP heuristic exhibits an exponential runtime and is essentially a single objective algorithm for minimizing power. As tested in [21], the method takes several hours to converge and reach an optimal solution.

Finally, the Application-specific NoC (ANOC) technique implemented by Srinivasan et al [26] create a mono-objective heuristic for synthesizing application-specific topologies by minimizing power given a system-level floorplan. Topologies are generated based on a recursive slicing tree partitioning method which divides the layout into horizontal and vertical components. While placing routers at the corners of the sliced partitions, the heuristic concurrently disregards factors such as router ports and other modeling issues which can lead to further increases in power. The method is also limited to small design spaces due to its low complexity, unsuitable for future MPSoCs based design.

3.2. PERFORMANCE DEGRADATION IN NoC DESIGN

Specific work for performance degradation factors such as deadlock and contention have been addressed by many previous works. Works discussed in this subsection give a very brief outline of previous methods which omit the topology design phase and solely deal with factors of performance degradation in NoCs.

The necessary and sufficient condition for deadlock-free deterministic routing was presented in the work of Dally and Seitz [27]. Dally also presented a resource ordering method to prevent deadlocks in custom topologies by inserting VCs to various ordered classes [28]. The methods are able to achieve a

deadlock-free configuration, but do not account for the many ways in which a VC can be inserted into a network in terms of power and performance factors. The work of Duato provides a necessary and sufficient condition for deadlock-free adaptive routing but can only be applied to regular network topologies [16]. The work employed by Linder and Harden prohibits turns to avoid deadlock in irregular custom topologies [29]. The method is able to eliminate deadlocks but can only be used during topology generation due to connectivity issues subsequent to synthesis.

Several researchers have aimed to create methods which relieve contention within on-chip networks. Rezazad et al and Mullins et al implement uniform VC distribution within on-chip networks [34][35]. The techniques do not optimize for power and performance, and can be wasteful of resources for network components that are not heavily utilized. Van den Brand et al suggest hardware probes to monitor link congestion in real time [30]. The method requires real time adjustments, extra hardware, links, and delays in order to control the congestion. It also does not directly address power dissipation due to the probes. Deniziak et al suggest rescheduling tasks and performing link insertion within the network to avoid contention [31]. The method only considers mesh topologies and does not directly deal with the power consumption of extra routers and links. Furthermore, delaying and rescheduling messages does not necessarily avoid the system from experiencing further delays and performance degradation. Techniques such as those proposed by Bjerregaard et al for adaptive routing require more resources and tend to concentrate traffic within the center of the network, implementing adaptive routers which require more on-chip resources [32]. Huang et al implement VC planning for NoCs in a customized manner in order to reduce contention [33]. Power however is not directly addressed when considering the performance improvement that the VCs provide to the on-chip network. Furthermore, mesh based topologies are only considered.

A previous work worth mentioning in both performance-degradation and power is the long-range link insertion technique proposed by Ogras and Marculescu [37]. The method attempts to exploit the benefits of regular NoC architectures by inserting long-range links to free congestion on the links

exhibiting heavy traffic. The links are inserted at critical traffic loads if there are enough resources and the congestion is improved. Insertion relies on an equation which determines the respective Manhattan distances, delays and frequency between components. Long range links are used in multiples of the regular sized links in the mesh network, which are connected by repeaters. The technique is able to reduce switching energy at the cost of larger crossbar implementations within the router due to the extra links, thus consuming more energy. The assumption that links can be applied in multiples presumes that the topology contains homogeneous core sizes. In addition, the technique does not employ a floorplanner to attain accurate wirelengths and power consumption estimations. The method is also not necessarily optimizing for the best performance and power outcome since a regular topology is inferior to the application-specific implementation.

3.3. POWER-AWARE TOPOLOGY GENERATION

Now that various performance-degradation related works has been relayed, it is evident that many of these works do not consider the power effects due to the alleviation and elimination of contention and deadlock. Conversely, previous work on power-aware topology generation techniques are now discussed and evaluated, outlining the advantages and disadvantages of general power-aware schemes.

Chatha et al propose an automated technique for the synthesis of application-specific NoC architectures [36]. The method initiates with an input floorplan containing cores and place routers at each of the channel intersection, in turn making the connections from cores to the routers accordingly. The techniques objectives rely primarily on reducing power consumption and the number of routers in the floorplan. Iterations within the topology generator merge routers and evaluate the resulting power values. Since the topology generation is based on a predetermined, set floorplan, the solution can not guarantee all shortest paths and therefore needs to make use of additional routers to achieve a near optimal solution.

The work assumes that routers can be placed at all corners of cores, and can therefore contribute to needless on-chip area overhead. The power consumption values do not consider wiring, and do not incorporate frequency of operation which can have significant effects on the power values in the floorplan. Moreover, the method can not guarantee that the communication requirements are being met while simultaneously minimizing power.

Hu et al implement a point-to-point communication synthesis technique while invoking floorplan information to design NoC interconnects for SoCs [38]. The method incorporates a communication-driven floorplan given the information volume between IPs, and the respective size of the cores. Hence during synthesis, wirelength is weighted not only by the length cost, but also by the communication volume between adjacent cores. This method is successful in energy savings of approximately 22% when focusing strictly on wirelengths, but disregards other power factors such as router sizes and clocking frequency. In addition, the work does not directly mention information on NI considerations.

Orgras and Marculescu propose an energy and performance driven NoC architecture using the energy bit approach [39]. The work assumes a floorplan has already been generated, where the input core coordinates must be supplied to the synthesizer assuming a mesh based topology. The method divides the core graph into subgraphs to analyze communication primitives which are classified as gossip or broadcasting transmissions. The subgraphs with the least energy cost are chosen, where the cost is determined as *energy per bit x communication volume*. The overall synthesizer is implemented in Matlab and tested on an FPGA. As discussed previously, the energy bit approach does not accurately attain power values given situations of heavy traffic and contention. The floorplan is assumed to be static, where routers and NIs are not taken into consideration. There exists a predetermined value for wirelength which can be subject to change given the heterogeneity of a given system. Therefore, the method mainly addresses minimizing energy costs while indirectly dealing with performance. The algorithm also exhibits fairly large runtimes during synthesis.

3.4. PROBLEM STATEMENT

As seen in the preceding works discussed, many designs which incorporate performance degradation solely focus on performance aspects while ignoring respective power consumption factors in the system. Similarly, certain works focus strictly on power while disregarding the performance effects of applying such a technique. The related works which used optimization methods to account for both issues often provided a final solution using a Pareto curve based technique which provided either a lowest energy or highest performance mapping. Through observation, it is evident that incorporating both constraints in NoC topology synthesis is problematic. Thus, now that these designs have been summarized and analyzed, it is possible to derive a problem statement of the underlying areas of refinement, and respective advancements needed to be applied to NoCs and within this thesis work.

It is clear that a multi-objective based technique is needed to generate topologies which account for various power and performance factors and related constraints within a system. It is also evident that a fusion between the modeling and simulation based approach is needed to compromise between low design runtimes and accuracy of power and performance values. During power analysis, it is imperative that all components be taken into consideration, i.e. routers, wirelength, and the layout in order to attain accurate values. It is also important that performance degradation and corresponding methods of alleviation are handled while assessing various performance constraints such as latencies, throughput, and utilizations. Frequency of operation is an important aspect for both power (link and router models) and performance (expected latencies during transmissions). Thus, frequency must also be analyzed during topology generation to achieve approximate values for these parameters.

In response to these demands, this work presents a topology generation technique for performance and power efficient application-specific NoCs using a Tabu search based optimization method. The technique compromises between analytical modeling and simulation tools to analyze and refine topologies for system constraints during synthesis. Performance models are run through an

integrated Layered Queuing Network Solver analysis (LQNS) tool to evaluate the rendezvous interactions within the network, assuming a best-effort, static wormhole routing method. By employing LQNS, the synthesis tool is capable of accurately predetermining performance factors and dynamic effects during generation of temporary topological solutions. A novel method for deadlock analysis is also introduced in this work to account for a power and performance constraint tradeoff specified by the target application. Power models have been previously simulated and fined-tuned for analytical purposes prior to the design phase, taking into account the NoC clocking frequency, along with static, dynamic and leakage power dissipation. In addition, the synthesis tool also uses an automated technique to attain accurate wirelengths, areas, and power consumption values at significant stages of the Tabu method. This thesis work is therefore able to combine the advantages of both modeling and simulation techniques, invoking a multi-objective optimization method to design low-power, high-performance NoCs. The next sections will describe implementation details, and how achieving both power and performance efficient topologies is in fact possible for future SoC designers.

4. TABU SEARCH OPTIMIZATION

The Tabu search (TS) algorithm is employed in this thesis work to incorporate an optimized method which takes multiple objectives into account when finding a solution. Hence, a background and in depth analysis of the TS based optimization method will be discussed in detail during this chapter.

4.1. TABU SEARCH BACKGROUND

Tabu search (TS) is a meta-heuristic algorithm that was originally established by Fred Glover in 1989. The initial aim of the optimization was to develop an efficient procedure that could cope with the complexity of current optimization problems [40]. The intended concept was to exploit various principles of intelligent problem solving, such that an optimal solution could be discovered in a reasonable amount of time. Glover had the idea of uniting the fields of artificial intelligence and optimization into one method. Hence, Glover noted the disadvantages of the SA and GA methods, and thought of applying a memory application concept in an optimization method. Given that prior optimal and non-optimal solutions are encountered when searching for a solution, employing memory would allow a method to recall several solution instances and thus move in a direction which would find a global optimal solution.

The following sections outline the details of the Tabu search based implementation, followed by the advantages of applying the optimization method to NoC generation in comparison to other methods and heuristics.

4.2. TABU SEARCH ALGORITHM

Tabu search (TS) is a meta-heuristic algorithm that employs an aggressive search procedure. The procedure progresses iteratively from one solution to another by moving in a neighbourhood space with the assistance of adaptive memory [40]. TS obtains a better solution by considering the influence of a move within the neighbourhood and incorporating factors of search history and the problem context [41]. The TS method escapes the trap of local optimality by using its ability to retrieve prior optimal and non-optimal solutions from memory. By keeping track of solutions within the search, it is possible to locate the global optimal solution with less computational effort and time as compared to other optimization methods.

Algorithm 1	General Tabu Search Algorithm
1:	Generate initial solution $s = N(s)$
2:	Evaluate current solution conditions
3:	$TL(s) = \{\}$ //initial empty Tabu List
4:	WHILE stopping criteria NOT met
	DO
5:	Identify $s' = neighbourhoodSet()$
6:	Move to the temporary solution s'
7:	Evaluate s' solution
8:	ConstraintCheck($s', AL(s), TL(s)$)
9:	IF solution meets constraints
10:	Place solution as last optimal $TL(s)$ entry
11:	Update current solution, $N(s) = s'$
12:	IF Constraints satisfied optimally
13:	EXIT
	END
	ELSE
14:	Place as a non-optimal $TL(s)$ entry
15:	Refer to $AL(s)$ to revert back to last optimal solution
	END
	END

As presented in Algorithm 1, the method commences with an initial feasible solution $N(s)$ and explores other possible neighbourhood space solutions in $neighbourhoodSet()$. TS inquires with its short term memory lists in order to prevent the reversal of recent moves as performed in $ConstraintCheck(s',$

$AL(s)$, $TL(s)$). The function ensures that the new temporary solution s' coincides with the Aspiration List $AL(s)$ and Tabu list $TL(s)$ to find a new local or global optimal solution which has not been previously encountered. In order to escape a local minimum, $AL(s)$ refers to $TL(s)$ to understand whether the current solution is inferior to a previous solution. If the latter is true, the algorithm reverts back to the state it was previously in and places the current solution into the $TL(s)$ to escape a local minimum. The algorithm continues until all stopping criteria are met.

Figure 4.1: The Four Principle Dimensions of Tabu Search Memory

4.2.1. MEMORY STRUCTURES IN TABU SEARCH

The memory structures employed in the Tabu search are of utmost importance to the algorithm in order to provide a best solution. The memory serves as a guide which directs the TS in searching for a new possible solution s' without reverting back to non-optimal or previously encountered solutions. Memory in the TS method relies on the four principles of recency, frequency, quality and influence. Recency and frequency are often considered as one intertwined principle which allow the algorithm to explore all areas of the neighbourhood and keep track of the directions which have been previously taken. The memory helps provide a quality solution by supporting multiple objectives when searching for a solution. Figure 4.1 outlines the principles of the memory structures experienced by the TS, where the darkest area signifies the main objective of incorporating all memory principles to find an optimal global solution. The TS method makes use of these four principles by invoking explicit and attributive memory types to guide the search in finding an optimal or near-optimal solution.

Explicit memory directs the search towards an influential and quality-based solution, keeping record of past moves within the TS to avoid cyclic behavior. Explicit memory services are provided by the Aspiration List $AL(s)$, and Tabu List $TL(s)$. The $TL(s)$ is responsible for keeping track of the non-optimal solutions to prevent the algorithm from revisiting previously encountered solutions which consumes time and decreases efficiency. During each TS iteration, the algorithm checks the $TL(s)$ entries to verify that solution s' is an optimal solution and that its current evaluated criteria does not match other non-optimal entries. The Aspiration criterion is satisfied if a move yields a solution better than the best obtained so far. The $AL(s)$ is therefore responsible for overriding restricted $TL(s)$ solution entries if the outcome of the move under consideration is sufficiently desirable [40]. Thus, the current restriction in the $TL(s)$ would be overwritten, where it is possible that the new current feasible solution can lead to a best solution.

The final type of short term explicit memory employed by the TS is known as the Candidate List $CL(s)$. $CL(s)$ is responsible for generating a list of possible moves within the neighbourhood, where each possible generated solution is evaluated by the $AL(s)$ and $TL(s)$ for optimality. More advanced TS methods employ candidate list strategies to narrow the examination of possible solutions when evaluating for s' , achieving a high quality solution within shorter period of time and reasonable amount of effort.

The second type of memory, known as attributive long term memory, is made use of by the *NeighborhoodSet()* function. The long term memory often referred to as the Frequency-Recency based memory (FR-Memory) structure, encourages the search to explore different regions within the neighbourhood and allow for the diversification amongst the different feasible solutions of s' . The FR-Memory keeps track of the frequency of moves within each area of the neighbourhood, and the recency of the vertices which have been previously moved.

In general, TS relies on the initial solution in order to generate better feasible solutions within the algorithm. Memory allocation can become large and demanding depending on the solution space and amount of moves needed to find an optimal solution from the initial generated solution. However by using this memory, TS avoids being entrapped within local minimum or local optima. Therefore, TS is typically given a specified maximum amount of entries to be used by the search to prevent excessive

memory usage. TS is conceptually much more simple than Simulated Annealing (SA) and Genetic Algorithms (GA) as well as easier to implement. Unlike SA and GAs, the TS is not limited to the evaluation of an objective function to find a solution. In contrast, TS obtains a best solution by considering the influence of a move within the neighbourhood and incorporating factors of search history and the problem context [40]. By making use of memory structures, a candidate list, and associative strategies, the Tabu search method unites the fields of artificial intelligence and optimization together to create an optimal or near-optimal solution [40].

4.3. ADVANTAGES AND DISADVANTAGES OF TS

As with all optimization methods, there exist both advantages and disadvantages to every approach. Thus we now examine the pros and cons of the general Tabu search optimization technique. The main reason for employing the TS based approach in this thesis work is mainly due to the methods ability to incorporate multiple objectives during solution exploration, which inherently overcomes the shortcomings of other optimization methods such as GA and SA. Although this is one of the methods primary strengths, TS also requires a large number of evaluations per optimization step. Hence, the time and effort need to evaluate every step can increase at the expense of a multi-objective approach. As the objectives within a feasible solution are achieved however, the computational effort is then gradually reduced per solution. Regardless of this issue, the TS method is able to reach a near-optimal solution in a reasonable amount of searching time in comparison to other methods by giving each objective equal weight without the use of a function.

Through the use of memory, the TS method avoids entrapment of non-optimal and local minima solutions until a near-optimal solution is found. In terms of implementation, Tabu is conceptually much simpler and computationally less demanding than other techniques. One of its downfalls however occurs during local and global solution searches. As the method continues to search in the neighbourhood

selection space, searches are often conducted locally as opposed to globally. Therefore, TS exhibits a low global search capability in comparison to other optimization methods – a problem which will be addressed during the topology generation technique invoked in this work. An additional fact worth mentioning is the memory space allocation needed to provide the design space with enough memory to find an optimal solution. Therefore, memory management must also be a design variable during topology synthesis to guarantee a near optimal NoC solution.

4.4. TABU SEARCH AND TOPOLOGY GENERATION

Optimization methods such as SA rely on min-cut partitioning to generate high-quality solutions while being limited to a cost function. Given that a cost function should invoke multiple objectives, as in the case of analyzing various power and performance factors, the function then becomes very expensive and slow to compute. All factors are amalgamated as one cost variable, where each variable may not necessarily exhibit the same weight as the other factors. In addition, due to the limitation of the cost function, the method can not determine whether an optimal solution has been encountered in comparison to previous generated solutions.

GA's are typically utilized in a random nature for large solution spaces. Similarly to SA, the solution quality of the method is also limited by the fitness function and may not necessarily lead to an optimal solution. GAs are not capable of obtaining initial knowledge of the base system and thus commence with a initial random configuration. However, given that the NoC system constraints and target application characteristics are a well defined priori, there is no logic in generating random solutions. Therefore, GAs often act as excessive and time consuming methods in topology generation. Furthermore, in order to invoke multiple objectives during topology generation, both GA and SA employ a Pareto curve technique to generate multiple single-objective mappings of power or performance.

Heuristics such as the ANOC and ILP techniques have also been applied to NoC topology generation. However, as noted in Chapter 3's related work section, the techniques are also limited to single objectives, often considering the minimization of power consumption over performance constraints. The ANOC heuristic is of low complexity, having difficulties exploring large solution spaces and therefore not suitable for the growing demands of NoC applications. The ILP method is limited to a single objective and takes several hours to converge to an efficient solution which is undesirable.

Contrary to these heuristics and optimization methods, the Tabu method allows for multiple objectives when searching for an optimal solution, where TS is not limited by a cost function. The method is able to keep track of previous optimal and non-optimal solutions through memory, reducing run-times and using less computational effort in comparison to other techniques. By using memory, the method is also able to generate solutions based on previously obtained information, and is not limited to randomly generated solutions. More advanced Tabu methods, such as the one invoked in this work, use candidate list strategies to help narrow the examination of solutions to help achieve a high quality solution for complex systems within a shorter time frame. Given that a disadvantage to the TS based method is its susceptibility to local searches, this method incorporates a detailed analysis of the source and destination cores of the system, allowing a smaller, local search to be conducted, where irrelevant global data is ignored. Therefore, by using the information provided by the target application, the Tabu optimization is ideal for generating multi-objective topologies needed to meet the constraints and complexities of future MPSoC based applications.

5. POWER AND PERFORMANCE EFFICIENT NoC DESIGN

This thesis work employs a multi-objective Tabu search based technique, incorporating both analytical models and simulation tools to compromise between the advantages of both approaches. The Tabu topology generation method is able to account for multiple power and performance constraints during synthesis, contrary to the related works discussed in Chapter 3. The following subsections outline the proposed analytical models for evaluating the power and performance of the target application, where the topology generator is able to accurately assess the quality of each temporary solution and allow for an overall efficient final solution.

5.1. INPUT MODEL

It is assumed that the target application selected by the user has been mapped to processing cores, and that the corresponding communication volume requirements between the cores have been determined statically. The application input model supplied to the generator can then be specified as a directed graph $G(V,E)$, where:

Figure 5.1: Core Graph Example

Figure 5.2: TSG Example

- Each vertex $v_i \in V$ represents a core within the graph.
- The communication between vertex i and j represent a directed edge (v_i, v_j) , expressed as $e_{ij} \in E$.
- The weight found on an edge e_{ij} denoted by $b(e_{ij})$ characterizes the bandwidth.
- A destination vertex (core) d_x , where $d_x \in V$ may have 1 to many sources cores s_x .
- The source vertex $s_x \in V$, and $\forall x \in 1 \dots N$.
- N represents the number of cores in the core graph.

The input graph is a model based on the cores and their communication requirements, with specific transaction details about the source and destinations used within the system and respective core sizes. The user can stipulate topology generation constraints such as maximum frequency of operation, buffer and flit sizes, an initial arbitration delay T_{arb_init} , router port constraints, and the anticipated packing/ de-packing (T_{pk} , T_{dpk}) delays of the NIs given the anticipated standard interface (i.e. OCP, AMBA, AXI).

5.2. ANALYTICAL PERFORMANCE MODELING

5.2.1. PERFORMANCE PROBLEM FORMULATION

A Topology Solution Graph (*TSG*) is created based on the current topological solution under evaluation as generated by the Tabu method, where $TSG = G(V, L)$. The vertices of *TSG* represent the processing elements (i.e. cores and routers) within the topology, whereas the edges denote the communication links between the vertices. Associated with each edge l_i is the expected transactions to be incurred on the link. An example of a *TSG* can be seen in Figure 5.2, with its respective core graph

shown in Figure 5.1. Given the *TSG* elements, a connection function (*CNX*) connects the link l_i to the next communicating link l_j in order to route the source core s_x to its destination d_x . A link is not capable of routing to itself, and therefore $l_i \neq l_j$.

Performance models in this work assume a static wormhole routing network when evaluating for factors such as delays, throughput, and performance degradation. Traffic generation is based on fully specified temporal patterns as determined by the required bandwidth of the target application. Given a router $r \in R$, with R total routers within the topology, each router may exhibit a different number of ports $p \in P$. Hence, it is worth mentioning that our model can be applied to any number of router ports as encountered in application-specific topologies, and is not limited to the standard four or five ported-routers as found in regular topologies.

The mean latency for a message to traverse from source s_x to its respective destination d_x is expressed as T_{lat} . This value incorporates various transmission delays including packing T_{pk} and de-packing T_{dpk} delays within network interfaces (NI), arbitration delays T_{arb} within the router for each encountered hop y , and possible contention and blocking factors T_{blk} which can hinder transmission of the packets. These factors can then be expressed as:

$$T_{lat} = T_{pk} + \sum_{i=0}^y T_{arb} + \sum_{i=0}^y T_{blk} + T_{dpk} \quad (5.1)$$

By specifying an initial arbitration delay and packing/de-packing delays for the models, the blocking delay probability per router is obtained through the LQNS analysis tool by evaluating the models for system contention. The fully specified temporal patterns allow each router to experience different arbitration delays, and thus each router within the topology is modeled in a unique way accordingly. The number of expected transactions to be incurred by each router port is calculated from the specified bandwidth values, and converted into flits/second. Given a traffic flow Λ from a source s_x to a destination d_x , the number of expected transactions within a router r , traversing through port p can then be expressed as:

Figure 5.3: 4-Port Router Configuration Model

$$\Gamma_{r,p} = \sum_{\forall s_x} \sum_{\forall d_x} \Lambda(s_x, d_x, r, p) \quad (5.2)$$

Given this transactional information for each router and port, the expected arbitration between the ports is now determined. The basic router model with four ports is presented in Figure 5.3, exhibiting one input FIFO per port with static header decoding capabilities and a respective output port. As illustrated, an input port p can not output data to its associated output port. Therefore, there are $P(P-1)$ possible connections within a router r . The expected overall traffic flow from a port i to port j (where $i \neq j$) with $N_{i \rightarrow j}$ transactions is then expressed as:

$$\theta(\Lambda(s_x, d_x, r, p), i, j) = \begin{cases} N_{i \rightarrow j} & \text{if } \exists \Lambda \text{ from port } i \text{ to port } j \\ 0 & \text{otherwise} \end{cases} \quad (5.3)$$

Given an initial expected arbitration value T_{arb_init} , the ideal arbitration delay experienced in a router r is then be expressed as:

$$T_{arb_r} = \sum_{r=0}^{\forall R} \left[\sum_{p=0}^{\forall P, r} \Gamma_{r,p} \left(\frac{\theta(\Lambda(s_x, d_x, r, p), i, j)}{\sum_{p=0}^{\forall P, r} \Gamma_{r,p}} \right) T_{arb_init} \right] \quad (5.4)$$

The arbitration delays per router and port are now determined given the traffic to be incurred, and are applied to the router models for analysis using the LQNS tool. Reflecting on equation 5.1, where T_{pk} and T_{dpk} are determined at design time, T_{blk} is analyzed and fine-tuned by the LQNS tool during the contention/ deadlock analysis and VC insertion. The delay can then be incorporated to determine the final frequency of operation for the current solution under analysis.

5.2.2. DEADLOCK-FREE MODEL

Figure 5.4: Cycle in LRG

Figure 5.5: Eliminating Cycle in LRG

A Link Reliance Graph (*LRG*) is a deadlock solution graph, where $LRG = G(L, E)$. The vertices of the graph represent the links between the vertices of the *TSG*, where the edges signify link reliance and associated transactions connected using *CNX*, such that:

$$E = \{ (l_i, l_j) \mid CNX(l_i, V) = l_j, v \in V \} \quad (5.5)$$

An example of the *LRG* derived from the *TSG* of Figure 5.2 is provided in Figure 5.3. A cycle which occurs within the *LRG* is defined as a set of links ϕ_j such that $\phi_j = \{l_1, l_2, \dots, l_k\}$, where $l_i \in L$ and the *LRG* contains an edge $e(l_i, l_{i+1})$, where $i = \{1..k\}$ and an edge $e(l_k, l_1)$ directed back to the initial vertex l_1 . As previously stated, a link is not able to route to itself in order to satisfy the deadlock-free criteria. The links which are then found to be of a cyclic nature in the *LRG* are labeled as ϕ for deadlock evaluation.

Once the *LRG* has been determined, the algorithm conducts a breadth first search (BFS) from every vertex $e \in E$ in order to determine any cycles within the graph. As employed in graph theory, the BFS is known for being able to compute large solution spaces which are generally needed for the growing demands of SoC applications. A cycle is encountered when a vertex in the *LRG* is detected along a path more than once. In order to remove a cycle, a vertex must be inserted into the *LRG* such that the cycle is eliminated. In practical application, this vertex insertion signifies a VC added to the system, as a vertex in the *LRG* is representative of an edge in the *TSG*. It is possible however that inserting a vertex may not

clear the cyclical dependency if more than one flow edge e is present between a set of vertices in the LRG , creating an embedded multi-loop scenario. Therefore, the algorithm checks all vertices once again after eliminating a cycle to guarantee that all deadlocks have been avoided. Given that the deadlock point is eliminated, the label ϕ is then removed from the link. An example of cycle elimination in the LRG and the corresponding modified topological solution is shown in Figures 5.4 and 5.5 respectively.

A cost is associated with breaking deadlocks in terms of both power and performance. By inserting the extra vertex within the LRG , or VC in the actual topology, an increase in power and performance is also experienced by the system. Therefore, our technique also evaluates the possibility of increasing performance due to the extra vertex insertion which is eliminating the deadlock, while taking into account the increase in area and power dissipation. In practice, the highest performance achievement is attained by inserting VCs at the links which exhibit a high amount of transactions, specified by the application. As a result, our method first assesses high transactional links in order to guarantee an increase in throughput while simultaneously eliminating contention and deadlock as described in the next subsection.

5.2.3. CONTENTION MODEL

The contention analysis in this work is assessed using criteria of link utilization and router port usage. As a result, the contention model makes use of both the TSG and LRG . The LRG is representative of the link connectivity and reliance in the NoC system. Therefore, if a link is heavily requested and utilized within the LRG , the requests can result in a potential bottleneck within the on-chip network. The degree of a given link vertex l_i must then be determined as:

$$\sigma_{l_i} = \sum_{k=0}^{E_{l_i}} e_k \quad (5.6)$$

Where,

$$\sum_{i=0}^L l_i = 2E \quad (5.7)$$

Here, E_{li} represents the total amount of edges entering and leaving the link vertex l_i in the LRG , where E and L signify the total edges and link vertices in the LRG respectively. In order to determine the heavy utilized links of σ_{l_i} , the following expression is employed:

$$Max^\circ = \max(\sigma_{l_i}) \quad \forall i \in L \quad (5.8)$$

The vertices in the LRG that exhibit a high degree of edges when compared to other vertices will satisfy the first criteria and be considered for contention alleviation. The second criteria that must be satisfied for contention relief is router port usage. As previously stated, the TSG incorporates core and router vertex connections. In order to determine the T_{arb} delays per router, the number of transactions for each router and port was determined and expressed as $\theta(\Lambda(sx, dx, r, p), i, j)$. Thus in order to find the heavy transaction utilized p for the corresponding router r , the following expression is applied:

$$\mu_{r_{p_z}} = \max(\theta(\Lambda(sx, dx, r, p), i, j)), \forall p \in P_r, \forall r \in R \quad (5.9)$$

where $\mu_{r_{p_z}}$ represents the highly transaction utilized ports for router r and respective port p_z of the total P_r ports. For both equations (5.8) and (5.9), vertices which exhibit a degree of three or greater will be considered for contention alleviation. The vertices however which have the maximum $\mu_{r_{p_z}}$ and Max° values take precedence over other vertices, given the availability of VCs in the resource pool.

In addition to assessing router port usage, the TSG is used to verify the results obtained in the LRG by evaluating the associated transactional link usage on the TSG edges. Thereafter, the links and router ports which met the specified criteria according to equations (5.8) and (5.9) are verified as an expected contention point δ such that:

$$\delta = \{Max^\circ \cup \mu_{r_{p_z}} \cup l_i\} \quad (5.10)$$

The contention points δ , in addition to the deadlock links ϕ within the target topology are taken note of and placed in an array $\Omega[]$ by order of importance with deadlock taking precedence. The contention and deadlock analysis relief method then employ the data in the $\Omega[]$ array to model contention and deadlock points as an LQN.

5.3. PERFORMANCE MODELING

Performance modeling is established based on the modeling conducted in the previous subsections. This section will develop the concept of a Layered Queuing Network (LQN), which will be used to model the on-chip network and predetermine performance degradation and possible factor improvements.

Figure 5.6: LQN Model Example

5.3.1. LQN MODELS AND PERFORMANCE

LQN was developed as an extension to the concept of queuing networks by employing a layered structure approach [42]. LQNs are capable of estimating the performance of a system given a set of predetermined values. A LQN models a system based on its functionality which is carried out by various tasks. The overall network is divided into layered tasks which demand services amongst each other. An LQN task can also consist of multiple entries which characterize its different operational capabilities. These tasks are then interconnected to simulate the functionality of the entire system. The role of an LQN is to model the wait-and-reply interactions that multiple layers in a system experience, referred to as rendezvous interactions.

An example of an LQN sub-network is shown in Figure 5.6 where each element is clearly labeled. The example presents two source cores (Sx_1 and Sx_2), arbitrating through the switch (T_x and R_x), to the sources transactional destination (D_x). Each layer in this example consists of one task, where Task 3

contains two entries, referred to as E1 and E2 (T_x and R_x). Task 1 and 2 are referred to as reference tasks, represented by a square, which await other tasks to complete. The thinking time Z represents the reference task's time to process a demand, in this case referring to the packetization delay within the source core's NI (T_{pk}). Tasks 3 and 4 are non-reference tasks (represented by parallelograms), where the number in brackets within the task characterizes the execution time to perform its functionality. Task three's execution time represents the arbitration time T_{arb} as specified in the performance modeling formulation, where task four signifies the message de-packing delay T_{dpk} . The values on the arcs represent the number of transactions that a task i makes to a task j , as obtained through the analytical parameter $\theta(\Lambda(sx, dx, r, p), i, j)$.

Heavy utilization of a task can create bottlenecks and limit the throughput of a system. The LQN is able to model various rendezvous interactions within the on-chip network and assess the throughput, utilizations, and other system performance factors for the tasks and their transactional requirements. The proposed methodology divides the NoC into sub-networks given the performance models and current topological solution being assessed. The inter-router, router-to-core, and router-to-router connections are modeled as different layers where the rendezvous interactions between the deadlock and/or contention prone layers are evaluated. The LQNS tool is used to evaluate points of performance degradation and their respective improvements due to the VC insertions in various contention and deadlock points.

5.3.2. LQN MODELING

The LQN model conversion is presented in Table 5.1, where each LQN element models an element in the contention and deadlock analysis method. There are three overall types of LQN models for modeling the $\Omega[]$ points within the NoC topology. The Inter-Router sub-network models are first created by our tool to represent the individual routers and ports within the topology. The generated models are then run through the LQNS tool to analyze and verify the anticipated transactions and execution times per port, and possible contention that would add to the arbitration delay T_{arb_init} . In the next step, the Router-to-Core sub-networks are modeled given the previous information obtained from the Inter-Router analyses, where

contention and deadlock points in the system are modeled based on the *TSG* and *LRG* data. Router-to-Router types (the third type) are also modeled given the anticipated amount of hops from a source to its respective destination. When all the sub-networks have been evaluated by LQNS, all performance factors are then obtained and the topology generator can assess performance benefits of VC insertion to the current topological solution.

Table 5.1: LQN-Performance Model Conversion

Model Type	LQN Component	Performance Model
Router-to-Core	Thinking Time (Z)	T_{pk}
	Reference Task	Source Core
	Non-Reference Task	Router/ Destination Core
	Execution Time	Reference Task T_{pk} , Non-Reference Task T_{dpk} , T_{arb}
	No. of Transactions	$N_{i \rightarrow j}$
Inter-Router / Router-to-Router	Thinking Time (Z)	T_{arb}
	Reference Task	Transmitting router port
	Non-Reference Task	Receiving router port
	Execution Time	T_{arb}
	No. of Transactions	$N_{i \rightarrow j}$

5.3.2.1. ROUTER-TO-CORE MODELING

The router-to-core LQN type has been provided as an example in Figure 5.6. As demonstrated, each source core in the NoC is considered as a reference task which awaits a response from the lower layers. The lower layers are considered to be the switch/router ports and/or destination cores. The source core and its respective NI have a packetization delay, and are modeled as the reference task's thinking time Z . Each router/switch task is modeled as a non-reference task with two entries representing the router operations of receiving (Rx) and transmitting (Tx) data. The execution time within the router entries represent the arbitration delay.

5.3.2.2. ROUTER-TO-ROUTER/ INTER-ROUTER MODEL

The Inter-Router LQN model is displayed in Figure 5.7. The figure is an example of the utilization of router port three (destination-based port), derived from a sample *TDG*. In this model type, the ports of the

router which transmit data to other ports are modeled as a reference task, where the destination port which receives the data is modeled as the non-reference task. The thinking time for the router ports in the inter-router model signifies the arbitration timing for the data. All router ports are modeled individually given the data obtained through the contention and deadlock models.

Figure 5.7: Inter-Router LQN Model

The final type of LQN model is the router-to-router model which is a slightly more complex model as it must model two or more hops. Our topologies generated using the Tabu search method however allows for a maximum of two hops from s_x to its respective d_x . As a result, the model has a maximum of four layers of rendezvous interactions in the LQN, similar to the example provided in Figure 5.7 which contains two layers (or one hop).

5.4. POWER MODELING

Router components found in the model library assume a wormhole, static routing technique. As illustrated in Figure 5.3, each router consists of one input buffer per port with header decoding capabilities, requesting access from the arbiter to forward the flit to the output port assuming a round-robin arbitration scheme. The routers and NIs employ a stop/go flow control strategy to avoid the overflow of buffers. The power aspects of the models are based on the 65nm technology library, established on NVT with a V_{dd} of 1.0 V. The power model library invokes the Orion library [45] to derive the power values of the routers and wires based on different router configurations. Specifically, the link lengths obtained by the Parquet floorplanner[44], the specified NoC frequency of operation, and the amount of VCs used per router have

been alternated to cover various scenarios within the power models. Routers assume a register-FIFO, cross-point fabric implementation to further reduce delay and static power consumption [45]. The VC routers are modeled similar to the standard router, where additional buffers are applied to the input channels.

5.5. VC INSERTION: THE PREDICAMENT

As previously stated, this thesis work employs a virtual channel (VC) insertion technique to alleviate blockage at contention points and eliminate deadlock cycles. Although VCs can also improve performance, multiple VC insertions require extra buffers and arbiters within the routers which can greatly increase power dissipation [46]. As previously demonstrated in subsection 2.2.5 Power and Performance Constraints, the two factors of buffers and arbiters can amount to 29% of the overall power dissipation in the NoC [17]. Furthermore, as technology continues to reduce in size and follow Moore's law, leakage power can no longer be neglected, especially when accounting for virtual channel usage. Thus, VCs should then be placed at critical places within the topology in order to maximize on resources and the extra power consumption, while simultaneously increasing performance in an optimal way. A simplistic case study on the effects of VC insertion to a topology will now be demonstrated, outlining the potential benefits and disadvantages of adding such resources.

5.5.1. CASE STUDY: A SIMPLISTIC EXAMPLE

Figure 5.8(a) illustrates an example of a given contention scenario, where possible bottlenecks are visible at port 4 of router 1 (μ_4) and port 2 of router 2 (μ_2). Bottleneck point μ_2 can be seen as a possible contention point which can also cause congestion within router 1. Figure 5.8(b) and 5.8(c) outline two different possibilities of VC insertion for the given topology. Figure 5.8(b) aims to relieve contention and congestion point μ_4 by inserting an additional virtual channel between router 1 and router 2. This

establishes an increase in both the area distributions of routers 1 and 2, and the power consumption due to the additional port buffers and arbitration needed in each router. Figure 5.8(b) alleviates some of the contention in router 1, but does not directly address the contention which continues to occur in router 2. Similarly, the example of Figure 5.8(c) seeks a way to relieve the bottleneck point μ_2 by inserting one VC at port 2 which is estimated to consume less power and area in comparison to the example of 5.8(b).

Figure 5.8(a): Initial Router and Topology Contention Example

Figure 5.8(b): VC Insertion at Router-to-Router Connection

Figure 5.8(c): VC Insertion at Destination Port Connection

Figure 5.9 provides the normalized power and performance outcomes for the example of Figure 5.8. As seen, the topology configuration of Figure 5.8 (a) provides the least performance and power

consumption due to the lack of contention alleviation. The router configuration of Figure 5.8 (b) consumes the greatest amount of power compared to the other configurations, but does however provide the system with a slight throughput improvement in comparison to (a). Finally, by inserting the 1 VC in router 2 as seen in (c), the system experiences a slight power increase in comparison to (a), but exemplifies a greater amount of throughput improvement in performance when compared to (a) and (b). In terms of area distribution, (a) provided the least area overhead (0.29mm^2), where configuration (b) provided the greatest area of 0.388mm^2 . The example shown in Figure 5.8(c) not only provided a fair-tradeoff of power and performance, but also gave a reasonable area distribution of 0.3374mm^2 .

Figure 5.9: Power vs Performance for Figure 5.8 (a), (b) and (c)

As demonstrated in this example, there are VC insertions which allow for little or no performance improvements with an increase in power dissipation, where other insertions provide the system with a considerable amount of throughput and latency improvements with an equal or reduced amount of power consumption. Similarly, as also seen in the examples of Figures 5.4 and 5.5, there are several ways in which VCs can be inserted in a system to remove cyclical dependencies. It is important however that several factors are taken into consideration to optimize for performance and power within a NoC system. Therefore, this thesis work proposes a power and performance efficient VC insertion technique to overcome such challenges and avoid contention within application-specific topologies. Given a resource pool of VCs specified by the user, LQN models of the topology, and accurate power models of the NoC

components, the VC insertion algorithm places the VCs in an efficient way to eliminate contention, while providing the system with a maximum increase in performance, minimal power dissipation, and low resource costs.

5.6. NOC TOPOLOGY GENERATION & OPTIMIZATION

This section will discuss the topology generation technique employed in this thesis work. The overall design flow is first discussed, with details of interactions between the simulation tools and our synthesizer. The Tabu based topology generation technique will then be discussed, with specifications of the initial solution and subsequent neighbourhood selected solutions. Performance degradation and the respective VC insertion technique will then follow, where a complexity analysis is then provided for the overall NoC Tabu topology method.

Figure 5.10: Overall NoC Topology Synthesis Flow

5.6.1. SYNTHESIS FLOW

The overall flow for the NoC synthesis tool is depicted in Figure 5.10. The TS topology generator assesses all NoC characteristics given in the system core graph and specified constraints. Each step in the NoC generation method analyzes system performance and observes changes in the frequency of operation

and power, while optimizing for other performance constraints within the NoC topology arrangement. Orion models are called upon by the tool to compute a quick analysis of power consumption given the current NoC frequency, router port constraints, and associated wirelengths. The topology is run through the system-level floorplanner to assess the wirelengths and area at significant stages of the solution to make use of time in an optimal way. In cases where the objectives and constraints do not meet the criteria, the TS method aims at discovering a new best solution that will satisfy the objectives as further elaborated in the following subsection. Once the system requirements have been met for a given temporary topological solution, contention and deadlock analysis is then performed. The on-chip network is modeled as a LQN to analyze possible contention and deadlock points within the system as specified by the $\Omega[]$ criteria. The potential bottlenecks are identified through contention analysis by calling upon the integrated Layered Queuing Network Solver (LQNS) tool, and relieved using a temporary virtual channel (VC) insertion method. Once a solution has been derived which meets all criteria in an optimized manner, the floorplanner performs wirelength minimization and compaction on the solution, in addition to detecting timing violations within the system to generate a final topological solution.

5.6.2. INITIAL TS BASED TOPOLOGICAL SOLUTION

Given the constraints specified by the user, the topology generator creates an initial NoC topology, referred to as $N(s, f, P)$. Each vertex V_i in the core graph is assigned to a NI. The initial NoC topology generation is referred to as the fully-connected crossbar approach, where all the NIs/vertex cores are connected to one central router [6]. An initial NoC frequency is determined based on all the connections within the on-chip network, where T_{arb_init} , T_{blk} , and T_{lat} are determined to be a very high values given the application and size of the initial router. The iterations within the TS method divide the initial large crossbar router, where preference is given to grouping frequently communicating cores within the same routers. Router connections are based on the communication requirements between each s_x , and their respective d_x .

5.6.3. PROBLEM FORMULATION

Let $N(s, f, P)$ represent the current feasible NoC topology solution s consuming power P at a frequency f , and $N(s)$ express a new possible solution s' within the neighbourhood set. We define a TS based $TL(s)$ that contains non-optimal solutions, and $AL(s)$ that is responsible for consulting the Tabu list to ensure that s' is optimal and more desirable than the previous encountered solutions. Thus in order for the new solution $N(s) = s'$ to be an optimal solution $\omega(s)$, and a possible current feasible solution $N(s, f, P)$, the following must be satisfied:

$$\omega(s) = \{N(s) \cap TL(s)\} = \emptyset \quad (5.11)$$

If the $\omega(s)$ condition in (5.11) holds true, $N(s)$ is disjoint with the Tabu list set and is an optimal entry with respect to $TL(s)$. $N(s)$ is then consulted $AL(s)$ to verify that it is an element which is optimal with respect to the previous encountered solutions. This can be expressed as:

$$N(s) \in \{\omega(s) \cup AL(s)\} \quad (5.12)$$

If expression (5.12) holds true, the old solution s can be updated to the new solution s' , and frequency f can be changed to reflect the new frequency f' . The TS based NoC design displayed in Algorithm 2 iterates through feasible solutions, each time identifying a new possible topology configuration $N(s)$. The method rearranges the topology to the new solution in order to assess the factors within the system. Latency, NoC frequency of operation, and router port constraints for the new temporary topological solution are evaluated during each iteration of '*ConstraintCheck*' to verify whether the stopping criteria have been met. The power is also determined for each new solution, where the NoC design method ensures that the power consumption has not significantly increased from the last $N(s, f, P)$ move. If the operating frequency exceeds the maximum possible frequency, or the power is significantly increased within the current possible solution being evaluated, $AL(s)$ is referred to and the last optimal solution is restored. The undesired solution is placed into $TL(s)$ and the algorithm attempts to obtain another neighbourhood solution.

Algorithm 2 Tabu Search Topology Generation Algorithm

```
1: Generate initial topology solution  $N(s, f, P)$ 
2: Evaluate  $N(s, f, P)$  for initial frequency  $f$  and power  $P$ 
3:  $TL(s) = \{\}$  //ensure empty Tabu List
4: WHILE performance and power constraints NOT met
  DO
5:   Identify  $s' = N(s)$ 
6:   Move to the temporary solution  $s'$ 
7:   Evaluate  $s'$  solution for  $f'$  and  $P'$  using Orion models
8:    $ConstraintCheck(s', AL(s), TL(s))$ 
9:   IF solution meets  $AL(s)$ ,  $TL(s)$ ,  $f'$ , and  $P'$  constraints
10:    Create LQN models for sub-networks
11:    Invoke LQNS tool for performance analysis
12:    Place solution as last optimal  $TL(s)$  entry
13:    Update current solution,  $N(s, f, P) = s'$ 
14:    Check for deadlock, contention, utilization
15:     $VCInsertion(); (T_{arb})$ 
16:    Determine  $f'$  based on updated  $T_{arb}$ 
17:    IF  $f' \leq f_{max}$  THEN
18:      Run through system-level floorplanner
19:    ELSE
20:      Go to line 22
21:    END
22:    IF power/ perf constraints & router ports satisfied
23:       $N(s, f, P)$  and EXIT
24:    END
25:  ELSE
26:    Place as a non-optimal  $TL(s)$  entry
27:    Refer to  $AL(s)$  to restore last optimal  $N(s)$ 
28:  END
29: END
```

5.6.4. NEIGHBOURHOOD SELECTION

The initial topological solution in this work implements a large switch and can be considered to be a poor solution as this configuration consumes a considerable amount of power. Furthermore, contention within this central router is also at the highest since all connected cores contend for the same crossbar. Finding a new solution within the neighbourhood of possible solutions is therefore greatly desired. The $CL(s)$ and the candidate list strategy employed in this work, known as the Successive Filter Strategy (SFS), assist the TS in finding a new solution $N(s)$ within the neighbourhood. A high quality solution in this work results in a low power, high performance topological arrangement which does not exceed the maximum frequency of operation. Hence, the SFS initially aims at first filtering the cores with low transaction rates since their moves will have less effect on the frequency and performance of the topology as compared to

the highly utilized cores. Power consumption is also decreased in comparison to the initial crossbar approach, as the large central router is divided into smaller components.

The head of the $CL(s)$ is the core chosen by the SFS based on either the initial low transactional cores, or the neighbourhood cores being kept track of by the FR-Memory. Given the initial head candidate list condition, low transactional cores are chosen by employing the following expression:

$$MinTrans = \min_n \left[V_n \left(\sum_{\forall s_x}^N \sum_{\forall d_x}^N N_{tr} \right) \right] \quad (5.13)$$

Given a vertex/core n , N is the total number of vertices/cores in the core graph, where $n = \{1, 2, \dots, N\}$. N_{tr} is the number of source (s_x) and/or destination (d_x) based transactions that the vertex V_n is expected to incur. X is the total amount of source/destination sets for the respective source core n , where $x = \{1, 2, \dots, X\}$. $V_n(w)$ represents vertex n and its expected total number of transactions w .

The candidate list is formed initially based on the minimum ceiling function of the vertex which exemplifies the lowest amount of transactions with other cores. In addition to this, the candidate list and SFS also verify with the FR memory whether the core has been previously selected, which allow for other low transactional cores to also take the head candidate position. The SFS then filters through the cores which communicate with the head of the candidate list. Subset arrangements of these cores are formed in different combinations, where the topology generator selects the subsets and configures the topology such that the cores are positioned according to the SFS combinations. The subset core combinations are then evaluated for solution quality. Let:

- $\pi \in \Pi$, where Π is a set of positions in the search space.
- $\pi(j)$ represent core j attaining the head candidate position.
- $\varphi(s)$ denote the set of possible moves that core j can have, when core j has occupied the position $\pi(j)$.
- m signify all possible combination of moves formed by the SFS.
- $\varphi(s)$ be divided into subsets $\varphi(1,s), \varphi(2,s), \dots, \varphi(m,s)$, where $\varphi(1,s)$ denotes the 1st subset move in the possible set of total moves generated by the topology generator etc.

Figure 5.11: Successive Filter Strategy Example

Figure 5.12: Successive Filter Strategy Temporary Solutions $N(s)$

An example of the SFS applied to a 16 core application is given in Figure 5.11. The head of the candidate list during this particular iteration is selected as core 15. Core 15 performs transactions with cores 9, 13 and 16. When finding a new $N(s)$, core 15 has occupied the position $\pi(15)$ in the possible set of moves $\varphi(s)$. The SFS then generates a set of m possible combination of moves given core 15, analyzing cores 9, 13 and 16 and their respective transactions with other cores as well. The resulting topological arrangements are then evaluated for quality given the factors of operational frequency, power, and port

constraints. Two possible subset combinations $\varphi(1,s)$ and $\varphi(2,s)$ for the SFS example is shown in Figure 5.12.

As discussed throughout this section, this work focuses on a multi-objective optimization with the two prime objectives of frequency and power. Sub-objective constraints such as maximum ports per router, latency values, deadlock avoidance etc. are also included, yet not incorporated into the algorithm as main factors as seen in Algorithm 2. Although the work pertains to the two objectives of frequency and power, the Tabu search technique could in fact be extended to three or more objectives due to the Tabu search's ability to account for such variability. The power and performance efficient topology generation scheme as presented in Algorithm 2 can be adjusted to account for additional objectives by adding to the criteria of lines 8 and 9 within the algorithm.

5.6.5. VIRTUAL CHANNEL INSERTION TECHNIQUE

The contention and deadlock analyzer is given a finite amount of VC resources imposed on by the designer and aims at equally distributing the VCs according to the hot spots within the system. Deadlock is first dealt with in the VC insertion technique as it is critical that these points are removed, followed by contention analysis as specified in the deadlock and contention spot tracking array $\Omega[]$. Algorithm 3 outlines the proposed VC insertion technique for deadlock and contention elimination.

Subsequent to determining the δ factors (as per equation (5.10)) within the *TSG* and *LRG*, the technique commences by analyzing the performance and power outcomes for temporary VC insertions of the generated deadlock LQN models. If there are in fact k multiple ways to insert the VC to avoid deadlock, where the k ways signifies the k vertices in the cycle, the technique analyzes all possible insertions for the best performance and power outcomes where preference is given to the highly utilized vertices. The best solution in terms of a performance and power tradeoff after the k iterations is used to

determine where the VC is permanently inserted. The analyzer continues to check for other possible deadlock points ϕ until all points within the $\Omega[]$ array are satisfied.

Algorithm 3 *VCInsertion()* - VC Insertion Algorithm

```

1:  Generate TSG, LRG based on  $N(s)$  solution
2:  Evaluate graphs for  $\text{Max}^\circ$ ,  $\mu$ ,  $\delta$ 
3:  IF  $\exists \phi \in \Omega[]$  THEN // Deadlock
4:    IF  $> 1$  way to insert VC
5:      Find  $k$  ways to insert, initialize  $i = 0$ 
6:      WHILE  $i \neq k$ 
7:        Insert VC at point  $i$ 
8:        Evaluate performance and power
9:        IF solution superior to previous insertion  $i-1$ 
10:       Save solution, determine  $T_{\text{arb}}$ 
11:       END
12:        $i = i + 1$ 
13:     END WHILE
14:   ELSE == 1 way to insert VC
15:     Insert VC at  $\phi$ 
16:   END
17: remove  $\phi$  from  $\Omega[]$  list
18: ELSE
19:   WHILE  $\Omega[] \neq \emptyset \ \&\& \ \exists \delta \in \Omega[]$  //Contention
20:     IF  $\delta$  previously encountered || previous  $\phi$  VC point
21:       Restore  $\psi$  &&  $P$  variables, VCs resources
22:     END
23:     Determine  $\psi_{\text{init}}$  and  $P_{\text{init}}$  factors
24:     Add 1 VC to  $\delta$  point
25:     Determine  $\psi_j$  performance cost
26:     IF  $\psi_{\text{init}} < \psi_j \parallel \psi_{j-1} < \psi_j$ 
27:       Determine  $P_j$ 
28:       IF  $P_j$  && enough resources
29:         Insert VC && update VC resources
30:       END
31:     END
32:   END WHILE
33:   Save solution, determine  $T_{\text{arb}}, f^*$ , and search for next  $\delta$ 
34: END WHILE
35: END

```

Contention analysis is conducted such that the VCs in the VC resource pool in excess from the deadlock avoidance are made use of. Once a bottleneck point is pinpointed, the LQN and power models are used to determine the initial throughput ψ_{init} , and power P_{init} factors respectively. Thereafter, temporary VCs are inserted at the bottleneck points to model the task once again and evaluate the improvement in performance ψ_j on the j th iteration per contention point, where the j th iteration signifies the j number of VCs inserted at the contention point under evaluation. The extra power dissipation P_j incurred by the system due to the VCs is also analyzed. The following criteria must be satisfied in order for the temporary VC insertion to take permanent effect:

- The performance improvement ψ_j is greater than the extra power dissipation P_j that the on-chip network will experience.
- There are enough VC resources for the insertion to take place.

The algorithm continues iterating while all the criteria continue to be satisfied. If all criteria are satisfied, the system permits the temporary VC to be inserted in the system at the designated spot to relieve contention where the updated value of T_{arb} is determined. The algorithm inserts multiple VCs if all criteria are satisfied and there are enough resources. Given the scenario that all the contention points have not been analyzed, or one of the criteria has not been satisfied, the analyzer proceeds to search for and relieve other potential bottleneck points within the topology as specified by the order of $\Omega[]$. Given the scenario of pending contention points within the topology due to VC resource pool limitations, the analyzer will report back with respect to any additional performance improvements that can be achieved, and the extra VCs which would be needed to do so. When the solution has been analyzed for performance, the updated value of T_{arb} is applied to T_{lat} . If all contention points are eliminated, T_{blk} can also be eliminated from (5.1), else the blocking probability is obtained from LQNS and incorporated into T_{lat} . T_{lat} is then converted into time units and multiplied by the initial frequency value to obtain f' .

5.6.6. COMPLEXITY ANALYSIS

Given a solution space of N cores, determining a move within the TS given the constraints imposed by the SFS yields $N(N-1)/2$ moves, expressed as $O(N^2)$. The swaps needed to place the cores in the new topological arrangement results in a complexity of $O(I)$, where $O(N)$ time is needed to evaluate the N cores. Given k total iterations within the search, and an average $TL(s)$ search time of i , the overall complexity of the proposed method can be expressed as $O(k(N^2+N+i))$, further simplified as $O(N^2 + N)$, assuming k and i as negligible.

Other NoC topology generation techniques such as the SA based method in [18] yield a maximum complexity effort of $O(n^3)$, where n indicates the number of cores within the application. Similarly, the synthesis approach for the GA based methods can lead to a computational effort of $O(n^2)$ to solve the fitness function, in turn yielding an overall complexity of $O(n^4 \log(n))$ [62]. Techniques such as the Point-to-Point algorithm presented by Dumitriu in [48] directs the search to a simplified complexity of $O(n^5)$. Therefore, the TS based technique presented in this work is able to provide a fair quadratic complexity as compared to the cubic, quartic logarithmic and quintic degrees of the other methods discussed.

5.7. METHOD LIMITATIONS

Now that the analysis method, topology generation, and VC insertion technique have been fully specified and established, limitations of the actual methodology are now discussed. Previously in Chapter 3, an outline was given which provided the advantages and disadvantages of all network design layers and the various on-chip parameter selections. As observed, the topology generation technique supported in this work implements best-effort, static wormhole routing. The method would therefore have to incorporate additional performance models, parameters, and power models in order to implement other routing methods, flow-control procedures, switching techniques etc. The actual Tabu based topology generation technique however can be applied to different types of topology architectures. Given the scenario that a regular mesh or torus based topology was under evaluation in the Tabu based approach, the power and performance tradeoff technique would not be as beneficial to power and area as factors would not vary significantly from solution to solution given the standard grid topological format. Thus in a grid like architecture, the TS would be ideal for incorporating multiple performance constraints.

A second limitation of the method is the strong dependence of source to destination transactional based connections. The method also focuses attention to the bandwidth which must be met as specified by

the input core graph. During synthesis, the successive filtering strategy strictly relies on this information to generate optimal topologies. Hence, this technique would not be suitable for stream-based architectures for example, which require specific attention to memory hierarchies and their locations in relations to their streaming cores. Therefore, a slightly modified version of the Tabu based approach would need to be employed with attention focused on these additional factors. Finally, as will be seen in the Experimental Results section of Chapter 6, a slight discrepancy in the estimated values obtained by LQNS in comparison to the actual performance values obtained during simulation at flit-level is observed. Although the technique is not able to accurately assess performance values to the decimal place, it is however capable of accurately estimating performance degradation, and specific throughput values within a 19.8% accuracy range.

5.8. CONCLUSION

This chapter has presented the theory and analysis of the Tabu based topology generation technique proposed in this thesis work. The chapter has described details about implementation, strategies, and how the performance degradation factors are addressed and alleviated using a power and performance tradeoff technique for virtual channel insertion. A complexity analysis of the method was outlined, along with a brief discussion on the topology synthesis' limitations. The subsequent chapter will now present experimental results obtained by applying this methodology to various benchmark applications.

6. EXPERIMENTAL RESULTS

In this section, a number of benchmark applications are used to test the proposed Tabu topology generation and analysis technique. Through experimental testing, the generated topologies are compared and contrasted to other previous application-specific and regular NoC works. The chapter commences with a description of the benchmark applications used throughout the experimental testing. The next section will then describe the test setup, as well as present the topologies generated by the Tabu based technique and those generated by the other previous methods. These NoC topologies will then be evaluated, where the benefits of this multi-objective method will be clearly observed. The performance degradation alleviation for each of the benchmarks will also be analyzed in detail, where the deadlock removal technique is compared to another well-known methodology. Contention will also be analyzed in depth, where power and performance trade-off benefits will be demonstrated through the use of various figures and tables. The subsequent section will then outline the various execution times per application, where finally the last section will conclude the results section and summarize the findings during testing.

6.1. BENCHMARK APPLICATIONS

This section presents the seven different benchmark applications used to test the proposed Tabu method for topology generation in terms of performance, power, and area. The benchmarks consist of various multimedia and networking applications used in MPSoC systems. Each application will be presented as a directed core graph, $G(V,E)$, with specifications of communication characteristics (edges) for each of the cores (vertices) in the application. The data-width values used in the core graphs have been omitted, as we assume a constant value of 32 bits unless otherwise specified. The following provides a detailed description of all application benchmarks.

Figure 6.1: MPEG4 Decoder Core Graph

MPEG-4 DECODER

The MPEG-4 Decoder can be found in many embedded applications. The decoder however is most popular in mobile devices which provide video decoding over IP applications. In decoding this audio visual data, it is seen in Figure 6.1 that the multimedia application exhibits high on-chip communication demands between cores, where the requirements have been previously specified in [47]. Specifically, it is the memory cores which experience a high communication demand due to heavy utilization. In a bus-based application, it is clear that these memory cores such as the SRAM (shown in Figure 6.1) can develop into the bottlenecks of such a system. Therefore, the MPEG-4 benchmark has been utilized for

many NoC design examples in order to show the benefits of applying such an application to the NoC domain. The communication volumes shown on the edges of the core graph represent the communication requirements in MB/s. In order to achieve greater throughput, the data-width values for the slave cores (i.e. memory cores SDRAM, SRAM1, and SRAM2) have been specified as 64 bits.

Figure 6.2: Layer-3 Switch Core Graph

LAYER-3 SWITCH

The Layer-3 Switch was adopted from the thesis of Dumitriu in [48], where the application was originally designed with the intention for backing up data in multiple servers. The Layer-3 Switch consists of Ethernet port controllers (labeled EC#), a general purpose CPU for routing purposes, and two Direct Memory Address (DMA) units. The DMA units are placed between the CPU/ memory, where the Ethernet ports experience uninterrupted transfers of information between the ports and memory in order to backup server data. As seen in the Layer-3 Switch of Figure 6.2, the DMA units are in high demand for communication transfer, which can also act as potential bottlenecks between the ports, DMA units and memory core. As a result, the thesis work will utilize this benchmark application to verify the advantage

of the on-chip approach. The results will also be compared to that of Dumitriu [48] to show the improvement to a NoC system by employing a power and performance tradeoff technique.

Figure 6.3: A/V Benchmark Application Core Graph

AUDIO/VIDEO BENCHMARK APPLICATION

The Audio/Video (A/V) benchmark application was originally presented by Hu et al [51]. A core graph was constructed based on the task flows in [6], where three additional memories were added to allow communications between the DSPs and CPU as seen in Figure 6.3. The benchmark is used as a compression standard for digital audio-video which has gradually replaced the MPEG-2 standard. As this is a multimedia based benchmark, the data-widths for the various links have been adjusted to 128 bits respectively to accommodate the large communication volume requirements between cores. Given that there are in fact many communication links between the cores, this poses is an excellent benchmark for demonstrating the need to compromise between power and performance constraints during topology generation. The bandwidth values in the figure are expressed in MB/s.

Figure 6.4: NCS1 Benchmark Core Graph

NETWORK COMMUNICATION SYSTEM BENCHMARK

The Network Communication System (NCS1) is used as an industrial network communication SoC subsystem for fast paced data processing and forwarding [50]. The core graph for the NCS is depicted in Figure 6.4, where communication bandwidth on the edges is expressed in MB/s. This benchmark has been previously used for NoC testing by Leary et al [21]. The SoC is responsible for supporting two TCP constraints: an encryption engine, and a USB subsystem. In the encryption engine, the EXT_IF core fetches data and stores it in RAM3, where the ARM926 processor and ASIC1 core takes the data for processing and stores it back in the RAM. The EXT-IF block then fetches the data for streaming data back. In the USB subsystem, the system receives data from the USB and transfers it to RAM1 where the DMA engine streams it to the SDRAM_IF for further streaming transmissions. The cores which have not been mentioned reside in third subsystem which is of lower priority and not used for TCP streaming purposes. The TCP subsystems described here assume a 128-bit datawidth value, where the lower priority subsystem remains as 32 bits for testing purposes during topology generation.

Figure 6.5: MWD Application Core Graph

MULTI-WINDOW DISPLAY APPLICATION

The Multi-Window Display (MWD) application is used for high-speed television displays. The MWD has been used as a benchmark in numerous previous NoC works for testing [6][47][52]. The intent of the MWD chip is to allow multiple windows to overlap on a display, previously restricted due to the window buffer capabilities of speed to store and retrieve window images. Thus, dedicated processing elements are connected in an SoC format to improve the efficiency of the overall system, where the work of Dumitriu and Khan [6] have added master and slave components for the various cores as demonstrated in Figure 6.5. The chip-set specifically described here allows for multiple Teletext (TXT) and internet pages with up to three live windows of video processing, where each window is capable of encompassing an arbitrary size [58]. The communication volumes in Figure 6.5 are presented as MB/s, where the data-width value is specified as 64-bits for all links given the multimedia demands imposed by the application.

Figure 6.6: Set-Top Box Application Core Graph

SET-TOP BOX APPLICATION

The Set-Top Box application has been previously employed by Hu et al [49] and Leary et al [21] as a benchmark test application for NoCs. Figure 6.6 presents the Set-Top Box, with communication links representing the bandwidth volume in MB/s. The Set-Top Box in general is used as a device which takes a signal as input, and transforms the data into content which can be displayed on a screen, TV, or display device accordingly. In addition, as seen in Figure 6.6, the application is also capable of processing auditory mp3 signals. Therefore, in order to display the signal as a motion picture of frames with audio, the application makes use of many digital signal processor (DSP) cores, ASICs, and memory to convert the content for audio visual displaying purposes. It is evident that this application consisting of 25 cores requires heavy multimedia communication transfers. Thus, during NoC topology generation, the heavy utilized cores in the H263 encoder and decoder core connections use 128 bit data-width links, where the mp3 encoder and decoder employ 64 bit widths, and the remaining non-demanding cores employ the standard 32 bit data-width links.

Figure 6.7: D26_Media Application Core Graph

D26_MEDIA APPLICATION

The D26_Media application is a realistic multimedia and wireless communication SoC consisting of 26 irregular sized cores. The system consists of an ARM core, DSPs, memory units, DMA engines, several peripheral devices, and L2C signal devices. The L2C cores give the SoC wireless communication capabilities as they employ the L2 frequency for improving signal strength during navigation. The core graph is presented in Figure 6.7, where the edges are represented in MB/s. The application has been used previously by Rahmati et al [53], where the specific core graph has been provided in [54]. As seen in the core graph, there is heavy communication demand for the DMA and ARM cores as they must process the various data from other cores. However, due to the nature of the benchmark, the data-width remains 32 bits except for the link between the DMA and DEBUG cores, which will be specified as 128 bits during topology generation in this work. Similar to the MPEG-4 Decoder benchmark, this application will serve as an interesting means for topology generation in terms of heavy utilization and contention within the memory and DMA cores.

6.2. TEST SETUP

The seven different multimedia and networking benchmark applications previously discussed are summarized Table 6.1. The application-specific topology which will be compared to our TS based solution is also referenced in the table under the App-Specific heading. All benchmarks will be compared to a regular mesh topology in terms of power, performance, and area, and have been mapped similar to the method proposed by Murali and De Micheli [47].

Table 6.1: Benchmark Summary

Benchmark	Graph ID	Cores	App-Specific
d26_media	B1	26	[53]
Set-Top Box	B2	25	[21]
MWD	B3	15	[6]
Audio/Video (A/V)	B4	21	[6]
Layer-3 Switch	B5	12	[48]
NCS1	B6	21	[21]
MPEG4 Decoder	B7	12	[6]

The routers are modeled as individual components within the floorplanner which take VC area into account. Power is also calculated with VC consumption being considered, where router port buffers are sized to 4 flits. Topologies in this work were generated using a restriction of 6 ports per router, and a maximum operating frequency of 2 GHz. Network Interfaces are taken into account within the core area, where each NI is estimated as approximately 0.2 mm^2 [55]. The accurate performance results for the generated topologies and previous works were obtained using an in-house cycle-accurate (flit-level) SystemC simulator, where the mesh regular topologies have been simulated using the worm_sim NoC simulator [56]. The routers invoked in this work employ round-robin arbitration amongst all the ports to prevent the system from starvation. Livelock is prevented by incorporating minimum length paths from source to destination cores within the topological solution. The topology generator implementation was tested using a computer with a 1.66 GHz atom processor and 1 GB of RAM, running a Linux operating system. Simulations were run for 20,000 cycles with an initial warm-up period of 100 cycles. Due to inconsistencies between power libraries used in the various works, we have re-designed layouts for each

of the application-specific topologies with the 65nm library used in this work to accurately compare power dissipation and area values.

6.3. TOPOLOGICAL POWER AND PERFORMANCE COMPARISON

Table 6.2: Comparison of Methods Using Normalized Metrics

Method	Normalized Metric	Benchmark						
		B1	B2	B3	B4	B5	B6	B7
Tabu w/o VC	Power	0.7277	0.6999	0.7299	0.71822	0.6936	0.796	0.6895
	Throughput	0.612	1.120	0.871	1.453	0.393	0.967	1.113
Tabu w/ VC	Power	0.7813	0.7391	0.807	0.7827	0.7711	0.851	0.723
	Throughput	1.594	1.323	1.215	1.891	0.532	1.476	1.213
Application-Specific	Power	0.88737	0.84749	0.7299	0.77137	0.8292	--	0.9551
	Throughput	1.557	1.122	0.979	0.95	0.423	--	0.949
Mesh	Power	1.60356	1.71335	1.735	1.7276	1.6846	1.3527	1.708
	Throughput	0.0611	0.0921	0.1855	0.711	0.1457	0.557	0.721

Table 6.3: Area Comparison

Benchmark	Area (mm ²)		
	Tabu	Mesh	App-Specific
B1	8.89	14.93	18.22
B2	6.41	13.4	6.54
B3	3.07	22.03	10.87
B4	11.19	32.35	22.92
B5	2.35	16.3	9.129
B6	4.00	32.35	--
B7	5.05	16.3	22.1

This section compares the Tabu based topological results to various previous works and to a regular mesh topological solution for each benchmark. The normalized power and performance results for the Tabu, with and without VC insertion, mesh, and application specific referenced topological solutions are presented in Table 6.2, where the area overhead comparisons between the various topologies are displayed in Table 6.3. Actual values for the power and throughput values can be found in the Appendices. The symbol ‘--’ was used to denote no solution for the normalized metrics obtained within Table 6.2. In terms of performance, our Tabu based NoC designs were able to achieve a 33.6% throughput improvement (flits/sec) in comparison to other methods. On average, our tool was also able to design NoC interconnects which consume 33.1% less power as compared to the other two techniques.

The run times for our tool ranged from 2.53 seconds for a 12 core application, where the A/V and Set-Top box applications took approximately 45 seconds to generate a final topology. Details of specific execution times are provided in the Appendix.

Figure 6.8: Tabu Based Topology Solution for d26_media (B1)

Figure 6.9: Application-Specific [53] Topology Solution for d26_media (B1)

Figure 6.10: Mesh Based Topology for d26_media (B1)

Table 6.4: d26_media Benchmark Core Descriptions

ID	Core	ID	Core	ID	Core	ID	Core
0	CONT2	7	P1	14	MEM4	21	MEM1
1	CONT1	8	MEM5	15	PE1	22	DSP
2	IMPRO2	9	L2C	16	P3	23	SDRAM2
3	FLASH	10	SDRAM1	17	PE2	24	DSPL2C
4	ARM	11	L2CC	18	MEM2	25	DSPL2CC
5	P2	12	DMA	19	MEM3		
6	IMPRO1	13	DEBUG	20	PE3		

Figures 6.8, 6.9, and 6.10 display the d26_Media topological solutions for the Tabu based approach, referenced application-specific topology, and the regular mesh based topology respectively. The double links presented in Figure 6.8 signify the VC insertion within the topological solution. Table 6.4 presents the core description for the benchmark as presented in the core graph of Figure 6.7 for the benchmark application.

The work employed by Rahmati et al [53] emphasized a QoS based topology generation for the d26_Media (B1) application by making use of fairly large sized routers. As a result, the work sacrificed power in order to meet and exceed the performance demands required by the application. The application-specific work also solely made use of an analytical approach in order to predetermine latency values expected within the NoC. On the contrary, this work employs a power and performance efficient based technique where the generator was restricted to a maximum of 6 ports per router while maintaining a two hop connection between all respective source and destination cores to maintain the tradeoff. This work also invoked both the analytical and simulation based approaches to predetermine and guarantee the performance and power outcomes. Performance outcomes of our VC insertion technique in comparison to [53] yielded fairly similar results, with our method leading to a slight throughput improvement of 1.16x, while dissipating 8.2% less power due to smaller switches and the use of VCs to improve performance. In comparison to the mesh based topology, the Tabu based technique was able to save 51.27% power while increasing performance by 9.02x the throughput rate.

Figure 6.11: Tabu Based Topology Solution for Set Top Box (B2)

Figure 6.12: Application-Specific [21] Topology Solution for Set Top Box (B2)

Figure 6.13: Mesh Based Topology for Set Top Box (B2)

Table 6.5: Set Top Box Benchmark Core Descriptions

ID	Core	ID	Core	ID	Core	ID	Core	ID	Core
0	DSP4	6	ASIC5	12	DSP1	18	DSP6B	24	DSP3
1	DSP5	7	MEM4	13	CPU1	19	DSP5A		
2	DSP6C	8	DSP6A	14	MEM1	20	DSP4A		
3	CPU2	9	ASIC2B	15	ASIC1	21	ASIC2C		
4	ASIC3	10	ASIC2A	16	MEM2	22	DSP4B		
5	MEM2A	11	DSP2	17	CPU2	23	DSP5B		

Figures 6.11, 6.12, and 6.13 represent the Set Top Box topological solutions for the Tabu based approach, referenced application-specific topology, and the regular mesh based topology respectively. The double links presented in Figure 6.11 signify the VC insertion within the topological solution. Table 6.5 presents the core description for the benchmark as presented in the core graph of Figure 6.6 for the benchmark application.

As discussed previously, the work of Leary et al employed a Genetic Algorithm based approach to generate application-specific topologies [21]. The method however was limited to a floorplan based approach, where routers were placed at all corners of the cores within the floorplan. Through iterations without the GA technique, a topological solution was generated, where the redundant routers within the floorplan were eliminated. However, as seen in Figure 6.12, the method makes use of several 2-ported routers due to this floorplan methodology, resulting in an increase in wirelength, extra routers, and power to generate topologies. Two solutions were provided in [21] for low power or low router usage based on a Pareto curve technique. The low power solution is presented in Figure 6.12 and is compared to the TS based technique. On the contrary, the TS based technique treats routers as individual components within the floorplan to eliminate the use of unnecessary resource usage and the restriction of employing routers at corners of the cores within the layout. As a result, the Tabu method was able to surpass the GA technique in performance, and consume approximately 18% less power in comparison. When contrasted to the mesh based topology, the Tabu approach was able to consume 56.8% less power and increase performance by a factor of 8.89x.

Figure 6.14: Tabu Based Topology Solution for MWD (B3)

Figure 6.15: Application-Specific [6] Topology Solution for MWD (B3)

Figure 6.16: Mesh Based Topology for MWD (B3)

Table 6.6: MWD Benchmark Core Descriptions

ID	Core	ID	Core	ID	Core
0	IN	5	SE_M	10	VS_S
1	NR_M	6	GFX_M	11	JUG_S
2	HS_M	7	OUT	12	MEM
3	VS_M	8	NR_S	13	SE_S
4	JUG_M	9	HS_S	14	GFX_S

Figures 6.14, 6.15, and 6.16 represent the MWD topological solutions for the Tabu based approach, referenced application-specific topology, and the regular mesh based topology respectively. The double links presented in Figure 6.14 signify the VC insertion within the topological solution. Table 6.6 presents the core description for the benchmark as presented in the core graph of Figure 6.5 for the benchmark application.

The work of Dumitriu and Khan was referenced for the application-specific topological solution [6]. The method under evaluation is a mono-objective technique as previously explained, dealing with throughput-oriented NoCs based on transactional information. The method however did not take factors of power into consideration during generation, and hence often ended up employing very large routers in order to meet transactional requirements which required high power consumption and area. The method created two topological solutions, one referred to as the Point-to-Point topology, and the second as the Partitioned Crossbar topology. The second method was taken as a comparison for this particular benchmark. The application-specific method was able to surpass the performance outcomes of the initial based Tabu solution with a greater amount of power dissipation. However, when VCs were added to the system during contention and deadlock analysis, the Tabu method was able to surpass the application-specific topology in terms of performance by 0.4x, with a slight increase in power dissipation. In comparison to the mesh topology, the Tabu based approach was able to decrease power by 53.46 % while increasing performance by a factor of 3.862x.

Figure 6.17: Tabu Based Topology Solution for A/V (B4)

Figure 6.18: Application-Specific [6] Topology Solution for A/V (B4)

Figure 6.19: Mesh Based Topology for A/V (B4)

Table 6.7: A/V Benchmark Core Descriptions

ID	Core	ID	Core	ID	Core
0	DSP8	7	DSP4	14	CMEM2
1	CMEM3	8	ASIC2S	15	ASIC4M
2	ASIC1	9	ASIC3S	16	ASIC4S
3	ASIC2M	10	ASIC3M	17	DSP3
4	DSP1	11	MEM1	18	CPU
5	DSP2	12	MEM2	19	DSP5
6	DSP7	13	CMEM1	20	DSP6

Figures 6.17, 6.18, and 6.19 represent the A/V topological solutions for the Tabu based approach, referenced application-specific topology, and the regular mesh based topology respectively. The double links presented in Figure 6.17 signify the VC insertion within the topological solution. Table 6.7 presents the core description for the benchmark as presented in the core graph of Figure 6.3 for the A/V benchmark application.

The work of Dumitriu and Khan was once again referenced for the application-specific topological solution for the A/V benchmark application [6]. The Partitioned Crossbar approach was used to compare the two topologies. The Point-to-Point topology was not compared to the Tabu technique as the topology invoked seven 3-port routers, four 4-port routers, one 5-port, three 6-port, and one 7-port router which consumed a great deal of router and wire power during analysis. Therefore, the topology of Figure 6.18 was used instead to provide a fair comparison between the two application-specific topologies generated, where the method invoked in [6] implemented one 10-port, 9-port and 5-port router alternatively within the NoC. Similarly to the MWD application, the throughput-oriented technique was mono-objective. As visible in Figure 6.18, large routers are used to adhere to throughput constraints. When the two topologies were compared, the Tabu method with VC insertion was able to surpass the throughput-oriented method of [6] in terms of performance by 0.33x while maintaining the same power dissipation and less area overhead. In comparison to the mesh based approach, the Tabu technique was able to decrease power dissipation by 54.7% while increasing throughput by a total factor of 6.0x.

Figure 6.20: Tabu Based Topology Solution for Layer-3 Switch (B5)

Figure 6.21: Application-Specific [48] Topology Solution for Layer-3 Switch (B5)

Figure 6.22: Mesh Based Topology for Layer-3 Switch (B5)

Table 6.8: Layer-3 Switch Benchmark Core Descriptions

ID	Core	ID	Core
0	EC1	6	EDMA2 S
1	EC3	7	EC5
2	DMA1_M	8	CPU1
3	EC2	9	DMA1_S
4	MEM1	10	EC4
5	EC6	11	DMA2_M

Figures 6.20, 6.21, and 6.22 represent the Layer-3 Switch topological solutions for the Tabu based approach, referenced application-specific topology, and the regular mesh based topology respectively. The double links presented in Figure 6.20 signify the VC insertion within the topological solution. Table 6.8 presents the core description for the benchmark as presented in the core graph of Figure 6.2 for the Layer-3 Switch benchmark application.

The application-specific referenced topological solution was provided by the thesis of Dumitriu [48]. Similarly to the journal paper previously discussed in [6], both the first and second methods of topology generation, namely the Point-to-Point and Partitioned Crossbar approach, yielded the same solution for the Layer-3 Switch application as presented in Figure 6.21. The throughput-oriented technique once again employed large routers in both the Point-to-Point and Partitioned Crossbar technique, thus we observe a 10-port and 4-port router within the topology, as compared to the Tabu based approach which invoked three 4-port and one 6-port router to compromise between power and performance efficiency. In terms of results, the Tabu based approach was able to reduce power by 7% and increase performance by a factor of 0.3x when compared to the application-specific method found in [48]. When compared to the mesh regular based topology generation scheme, the Tabu method was able to reduce power dissipation by 54.8% and increase overall throughput performance by a factor of 2.65x.

Figure 6.23: Tabu Based Topology Solution for NCS1 (B6)

Figure 6.24: Mesh Based Topology for NCS1 (B6)

Table 6.9: NCS1 Benchmark Core Descriptions

ID	Core	ID	Core
0	DMA	8	TIMER
1	RAM1	9	ARM
2	SDRAM IF	10	ASIC1
3	UART	11	EXT IF
4	USB 2.0	12	RAM
5	SWITCH	13	SWITCH
6	ITC	14	ROM
7	RTC		

Figures 6.23 and 6.24 represent the Network Communication System (NCS1) topological solutions for the Tabu based approach and regular mesh based topology respectively. The double links presented in Figure 6.23 signify the VC insertion within the topological solution. Table 6.9 presents the core description for the benchmark as presented in the core graph of Figure 6.4 for the NCS1 benchmark application. The application-specific topological solution is omitted due to the lack of NoC generation in the genetic algorithm based technique of Leary et al [21].

As previously discussed, the Genetic Algorithm based approach generated two solutions based on a Pareto curve technique, each of which was not able to come to a final solution due to timing violations within the floorplan. The work in [21] also compared their technique to the ANOC technique [26] to their various test benchmarks. This approach was also not able to reach a final solution for the NCS1 benchmark. Similarly to the GA based technique, the ANOC technique worked with a static floorplan of all the cores, and thus was even more restricted in block and router placement in comparison to the GA method. Therefore, the ANOC technique also did not determine a final solution due to timing violations. The timing violations within the GA based technique were mostly due to the static router floorplan surrounding the heterogeneous cores within the NoC. By not considering the routers to be present as separate components within the layout, both techniques were therefore not capable of achieving a final optimal solution without encountering timing violations. On the contrary, the Tabu based technique considered the cores and routers to be separate components within the layout of the floorplan, and thus by also compromising between the power and performance factors, was able to generate an efficient, near-optimal solution for the NCS1 benchmark application.

When the Tabu based method was compared to the mesh based approach, our method was able to surpass the performance of the regular topology by a factor of 1.65x and concurrently consume 41.1% less power.

Figure 6.25: Tabu Based Topology Solution for MPEG-4 Decoder (B7)

Figure 6.26: Application-Specific [48] Topology Solution for MPEG-4 Decoder (B7)

Figure 6.27: Second Application-Specific [57] Topology Solution for MPEG-4 Decoder (B7)

Figure 6.28: Mesh Based Topology for MPEG-4 Decoder (B7)

Table 6.10: MPEG-4 Decoder Benchmark Core Descriptions

ID	Core	ID	Core	ID	Core
0	RISC	4	RAST	8	IDCT
1	CPU	5	ADSP	9	SDRAM
2	VU	6	UPSAMP	10	SRAM1
3	AU	7	BAB	11	SRAM2

Finally the last benchmark is discussed during this section, where Figures 6.25, 6.26, 6.27, and 6.28 represent the MPEG-4 Decoder topological solutions for the Tabu based approach, the two referenced application-specific topologies, and the regular mesh based topology respectively. The double links presented in Figure 6.25 signify the VC insertion within the topological solution. Table 6.10 presents the core description for the benchmark as presented in the core graph of Figure 6.1 for the MPEG-4 Decoder benchmark application.

The MPEG-4 Decoder was compared to two different topological solutions as presented in the work of Dumitriu [48] and Srinivasan et al [57]. The work of [48] was only compared to the Tabu method in Table 6.2 and 6.3 due to consistency between the benchmarks. The first method of Dumitriu which employs the Point-to-Point topology was chosen to compare to the Tabu method as displayed in Figure 6.26. This technique does not consider the physical layout of components, yet has quite similar structure when compared to the work in [57] and Figure 6.27. The method proposed by Srinivasan et al commences with a physical layout of all the components, and adds routers to all the corners of the cores within the floorplan. A subset of the total switches is used to connect the components together according to the core

graph communication, where the redundant routers are then eliminated within the layout to reduce power consumption. However, as seen in Figure 6.27, there exists an unnecessary usage of routers for cores 4 and 8 due to this static configuration of floorplan information. Therefore, the Tabu method was able to reduce power by 15.0% and improve performance by a factor of 0.71x in comparison to Srinivasan et al [57].

The work of Dumitriu [48] was successful in grouping frequently communicating cores within the same routers as seen in Figure 6.26. This in turn aided in the throughput achievements within the NoC. The Tabu based approach was also capable of grouping frequently communicating cores within the same switch. However, as seen between Figure 6.25 and 6.26, the Tabu based approach was capable of creating a power and performance tradeoff by making use of 5 and 6-port routers, where the work in [48] used a large router, in conjunction with several smaller routers. From a power perspective however, the smaller routers could have been amalgamated together to form a larger router which consumed less power in comparison to the several small routers, wiring, and area overhead. As a result, the limitations of the Point-to-Point throughput-oriented technique are visible in comparison to the multi-objective Tabu based approach. When the two topological solutions were compared, the Tabu based technique yielded an improvement in performance by a factor of 0.28x while dissipating approximately 24.2% less power in comparison to the application specific topology of [48].

6.4. CONTENTION ANALYSIS

This subsection presents analysis of the VC insertion technique for contention analysis applied to the various benchmarks. The MWD application has been excluded from this section as VC insertion would not lead to an increase in performance without an excess of power consumption to the network. However, as will be seen in the deadlock analysis section, VCs did aid in the removal of cyclical dependencies within the MWD network.

Figure 6.29: ASIC2 Slave Sub-Network Power/Performance Tradeoff

Figure 6.30: CMEM3 Sub-Network Power/Performance Tradeoff

An example of the VC insertion power and performance tradeoff for the ASIC2 Slave sub-network (consisting of the ASIC2S, DSP2, DSP5, DSP6 and router) and the CMEM3 sub-network (consisting of the CMEM3, DSP, CMEM2 and router) in the A/V TS topology are presented in Figures 6.29 and 6.30 respectively. The ASIC2-S sub-network displayed an increase in performance of approximately 18.2% due to a 13.99% increase in power dissipation for the insertion of one VC. When two VCs were temporarily added to the system during the second iteration of the $\Omega[]$ criteria loop, the NoC system showed a 22.47% performance improvement due to a 19.77% additional increase in power. The analyzer then ceased in searching for further improvements at the 3rd VC insertion as performance would no longer be of an asset to the network in comparison to power dissipation (as seen in the intersection of Figure 6.29). Similarly, when the rendezvous interaction contention analysis was performed on the CMEM3 sub-network, a performance improvement of 29.48% and 12.95% due to a

power dissipation increase of 12.70% and 20.78% for one and two VC insertions respectively was observed. The performance and power tradeoff therefore intersected at 2 VCs, where performance was no longer at an advantage. As a result, one VC was entered at the designated contention point within the topology to alleviate contention.

Table 6.11 presents the specific experimental power and performance tradeoff values of the rendezvous interaction contention analysis method for the benchmark sub-networks. Actual performance and power increases from contention alleviation are outlined in bold. Asterisks (*) present next to the value in the total VCs column indicate a VC previously inserted during deadlock analysis. Hence, the actual number of VC resources inserted during contention analysis is subtracted by the number of asterisks.

As an additional example of the VC insertion method, consider the Set-Top Box application topology which was also evaluated for contention. The MEM1 sub-network (consisting of MEM1, CPU1, ASIC1, DSP3, and the respective connected router) was first considered for alleviation due to its high transactional utilization. A temporary VC was previously placed at the router port for deadlock avoidance. Hence the algorithm noted that the VC was also applicable to contention alleviation, with no additional resources needed during the first iteration. The next core under assessment was the ASIC 1 core also within the same sub-network as MEM1 which met the contention criteria imposed by the models. Thus, when the temporary VC insertion was placed at the ASIC1 core, it was observed that the insertion would lead to a router which now consisted of 2 VC ports due to the previous temporary VC added at MEM1. During the next iteration back to the MEM1 core, adding an additional VC to the router would add a cost of 59.1% increase in power to the network, unworthy of insertion from a power perspective.

An interesting observation during experimental results was the several instances of deadlock which also occurred at contention points within the topologies. This is an important point to note as it can be seen that the dependence of resources that both deadlock and contention points endure are similar in nature. Actual performance results of the generated topologies were simulated using an in-house SystemC simulator and displayed a worst case deviation from the contention analyzer of approximately 19.8%

during simulations. Therefore, the contention analyzer was able to predict the bottleneck points and performance improvements accurately during topology generation. Overall, the analysis method was able to relieve various contention points and as a result increase the performance of the initial topological solution by an average of 25.3%.

Table 6.11: Power/Performance Tradeoff for Contention Analysis

<i>Topology</i>	<i>Sub-Network</i>	<i>Performance (%)</i>	<i>Power (%)</i>	<i>Total VCs</i>
MPEG4	SDRAM2	0.3215	6.37	0
	SDRAM	11.5646 12.01	6.37 14.57	1
	RISC	0	7.79	0
	Actual Performance/Power Increase: 9.0 / 3.2 (%)			
NCS	ITC	12.24 19.10 20.89	5.87 10.46 20.79	2*
	ARM926	4.18	4.70	0*
	Router 0 to Router 3	5.877 9.04 12.65	4.71 8.47 16.54	2
	Actual Performance/Power Increase: 32.6 / 10.1 (%)			
Layer-3 Switch	DMA_1 Master	5.7 6.38	2.00 7.814	1*
	DMA_2 Master	36.157 36.851 37.023	6.390 18.269 37.54	2*
	MEM1	1.483	8.945	0
	Actual Performance/Power Increase: 24.8 / 7.8 (%)			
A/V	CPU	30.00 31.2	12.70 30.9	1
	CMEM2	0.6	12.7	0
	ASIC2 Slave	18.2 22.47 24.52	13.99 19.77 27.35	2
	CMEM3	19.48 20.95	12.70 20.78	1*
	Actual Performance/Power Increase: 30.14 / 10.01 (%)			
Set-Top Box	MEM1	18.9 32.7	12.1 59.1	1*
	ASIC1	49.0 52.3	26.16 59.1	1
	CPU2	21.99 26.98	12.1 26.16	1
	Actual Performance/Power Increase: 18.3 / 9.3 (%)			
D26_Media	DMA	23.4 25.1	13.22 24.9	1*
	ARM CPU	15.2 18.3	13.22 24.8	1
	Actual Performance/Power Increase: 16.7 / 13.68 (%)			

6.5. DEADLOCK ANALYSIS & REMOVAL TECHNIQUE

Figure 6.31: VC Resource Insertion Comparison for Deadlock Removal

As demonstrated in Table 6.11, the resource ordering technique of Dally [28] is able to remove deadlocks and often provides the system with an increase in throughput. Both our deadlock removal technique and [28] were applied to our initial Tabu based topologies. In comparison to the proposed VC insertion technique presented in this work, our method was able to surpass the throughput values of the resource ordering technique slightly by 4%. Conversely, when the power dissipation values between the methods were evaluated, our Tabu technique was able to save approximately 9.35x less power in comparison to that of the resource ordering method. When contrasting this work to that of [28] with the number of VCs needed in order to relieve both contention and deadlock, our method managed to save approximately 84.8% additional resources. Figure 6.31 outlines the number of extra VCs incurred by each method. As illustrated in the figure, the MPEG-4 Decoder (B7) did not suffer from contention in either deadlock removal technique, displaying no extra VCs needed.

6.6. CONCLUSION OF RESULTS

The experimental results presented in this chapter validate the success of the proposed Tabu search based multi-objective method for topology generation. The results obtained in this section address various performance and power factors, where each of the benchmarks was compared to other proposed

techniques in NoC design. The results show an overall three-fold experimental outcome. Firstly, when compared to regular topologies, the Tabu method is able to provide a superior performance and power dissipation results due to its power and performance efficiency and application specificity. Second, the comparison in this chapter displayed an excellent tradeoff in terms of performance outcomes when compared to previous application specific employed works, while demonstrating equal, if not significantly less power consumption in comparison. Thirdly and most importantly, the contention and deadlock analysis method invoked during topology generation was capable of predicting the performance and power outcomes within a 19.8% deviation (in the worst case) while concurrently requiring less than a minute for execution. Thus, the Tabu method was not only successful in discovering power and performance efficient NoC designs within a short period of time, but was also able to achieve a tradeoff between the analytical and simulation approach in order to achieve an accurate prediction for the factors invoked during topology generation.

7. CONCLUSION

This thesis document presented a multi-objective Tabu search based technique for application specific NoC topology generation. The main motivation and contribution within this work was the development of a methodology to design efficient NoCs by incorporating multiple objectives of power, performance, and various system constraints within MPSoCs. The thesis therefore defined a formal approach to developing power and performance efficient NoC interconnects using a tradeoff between an analytical and simulation based approach during topology synthesis. During analytical evaluation, the method also incorporated latency values within the NoC components in order to compensate for the discrepancies in frequency of operation during packet transmissions. Furthermore, this thesis work contributed a novel contention and deadlock analysis method using a power and performance tradeoff for VC insertion, where the analyzer was able to accurately predict performance and power results during synthesis.

Experiments were conducted on seven multimedia and networking benchmark applications, where the Tabu based topologies were compared to previous application specific NoCs and that of a regular mesh based design. Results displayed that the Tabu based approach was able to surpass the performance results prior or subsequent to VC insertion, where power results achieved equal if not a significant decrease in power dissipation in comparison to the application specific and mesh based

methods. One major contribution observed during experimental results was the automated techniques ability to allow routers to act as separate components within the layout. When compared to other works, this automated method allowed for a reduction in wirelength and router power, resulting in an overall decrease in power dissipation and timing violations within the system. The predictive analysis method was able to determine performance and power results during synthesis within a 19.8% accuracy rate in the worst case. All NoC topologies invoked in this work for the various benchmark applications were obtained in less than one minute of execution time.

This thesis work has incorporated both high-level and automated synthesis in order to anticipate the physical implementation of the on-chip network. The method has also incorporated analysis of the layout constraints. However, future work within this thesis could incorporate the actual NoC chip implementation by verifying various physical constraints such as parasitic capacitances, logical effort within the circuit, and such impeding factors as the critical path in order to determine if the frequency of operation will in fact allow for a sufficient speed for the transactional requirements between the cores. Therefore, by allowing for an actual implementation and employing such low-level synthesis tools as Synopsys and Cadence, the success of this methodology can truly be tested and verified. In addition, the topology generation technique could also be further enhanced by incorporating the physical layout implementation to adhere to the forthcoming 3D chip concept which has gained significant attention in both academic and industrial applications.

PUBLICATIONS

This work has been published in the following conference proceedings:

Anita Tino, Gul N. Khan, "Power and Performance Tabu Search Based Multicore Network-on-Chip Design," *International Conference on Parallel Processing Wrks (ICPPW)*, pp.74-81, 2010

Anita Tino, Gul N. Khan, "Multi-Objective Tabu Search Based Topology Generation Technique for Application-Specific Network on Chip Architectures", *Design Automation and Test in Europe (DATE 2011)*, Grenoble, France, March 2011.

The work presented has been accepted for publication as a regular journal paper in the following:

Anita Tino, Gul N. Khan, "Designing Power and Performance Optimal Application-Specific Network-on-Chip Architectures," *Microprocessors and Microsystems*, 2011.

8. APPENDIX

This section presents all numerical values for the data presented in Chapter 6 for various normalized and unspecified numbers during experimental results, where Table 8.1 summarizes the formulation parameters used within the Tabu topology generation technique.

Table 8.1: Definition of Terms used in Tabu Topology Generation Formulation

Term	Definition
TDG	Topology Dot Graph is a directed graph G where $TDG = G(V, L)$, V is a set of vertices which represent the resources (cores and routers) and L is a set of edges denoting the links within the topology
CNX	Connection function, which connects the link l_i to the next communicating link l_j in order for s_x to reach d_x
LRG	Link Reliance Graph is a directed graph G where $LRG = G(L, E)$, L is now a set of vertices signifying links, and E is a set of edges denoting the pairs of links connected through CNX
s_x	A source core, or source vertex in the TDG
d_x	A destination core, which source s_x in the TDG
T_{arb}	Arbitration delay
T_{lat}	Overall latency delay
$T_{pk/dpk}$	Packing or de-packing delay of the Nis
$\Gamma_{r,p}$	Number of transactions expected in/out port p of router r
$\theta(\Lambda(sx, dx, r, p), i, j)$	Overall traffic flow from port i to j
ϕ	Deadlock communication link
σ_{l_i}	Degree of a vertex link l_i
Max°	Heavily utilized links of LRG
$Trans_{r_y p_z}$	Amount of transactions incurred by router r_y port p_z
$\mu_{r_{p_z}}$	Highly utilized ports of TDG
δ	An expected contention point within the topology
$\Omega[]$	1D array which holds all contention and deadlock points to be modeled with LQNs

Table 8.2: Execution Run Times for Benchmark Topology Generation

Benchmark	Execution Time (s)
B1: D26_Media	14.703
B2: Set Top Box	13.975
B3: MWD	6.390
B4: Audio/ Video (A/V)	12.821
B5: Layer-3 Switch	6.230
B6: NCS1	7.168
B7: MPEG-4 Decoder	5.403

Table 8.3: Power Dissipation Breakdown

Benchmark	Method	Router Power (mW)*	Wire Power (mW)
B1: D26_Media	Tabu w/o VC	268.12	10.38
	Tabu w/ VC	249.02	10.38
	Application-Specific [53]	306.2	10.09
	Mesh	546.58	25.02
B2: Set Top Box	Tabu w/o VC	217.05	13.84
	Tabu w/ VC	233.2	13.84
	Application-Specific [21]	266.66	16.568
	Mesh	552.64	20.00
B3: MWD	Tabu w/o VC	160.54	4.785
	Tabu w/ VC	172.76	4.785
	Application-Specific [6]	152.1	7.912
	Mesh	300.26	8.1345
B4: Audio/ Video (A/V)	Tabu w/o VC	198.33	9.812
	Tabu w/ VC	208.15	9.812
	Application-Specific [6]	195.06	19.72
	Mesh	460.07	21.02
B5: Layer-3 Switch	Tabu w/o VC	92.77	4.657
	Tabu w/ VC	103.66	4.657
	Application-Specific [48]	111.27	5.210
	Mesh	231.75	7.872
B6: NCS1	Tabu w/o VC	139.38	1.657
	Tabu w/ VC	148.104	1.657
	Application-Specific[21]	--	--
	Mesh	231.75	7.872
B7: MPEG-4 Decoder	Tabu w/o VC	95.66	5.629
	Tabu w/ VC	100.64	5.629
	Application-Specific [57]	116.56	6.91
	Application-Specific [6]	132.76	7.522
	Mesh	231.75	7.872

*Power takes into consideration static, dynamic and leakage currents within components

Table 8.4: Frequency of Operation for Final Topological Solution

Benchmark	Frequency of Operation
B1: D26_Media	1.5GHz
B2: Set Top Box	1.37GHz
B3: MWD	912MHz
B4: Audio/ Video (A/V)	1.20GHz
B5: Layer-3 Switch	330MHz
B6: NCS1	900MHz
B7: MPEG-4 Decoder	1GHz

REFERENCES

- [1] Goosens, K., Dielissen, J., Radulescu, A., "AETHEReal Network-on-Chip: Concepts, Architectures, and Implementations," *IEEE Design and Test of Computers*, pp. 414-421, 2005.
- [2] Dally, W. J., Towles, B., "Route Packets, Not Wires: On-chip Interconnection Networks." *Proc. DAC*, pp. 684-689, 2001
- [3] Ogras, U., Marculescu, R., "Application-Specific Network-on-Chip Architecture Customization via Long-Range Link Insertion," *Proc. ICCAD*, pp 246-253, 2005.
- [4] Millberg, M., Nilsson, E., Thid, R., Jantsch, A., "Guaranteed Bandwidth Using Looped Containers in Temporally Disjoint Networks Within the Nostrum Network-on-Chip," *Proc. DATE*, pp. 890-895, 2004.
- [5] Murali, S., Meloni, P., Angiolini, F., Atienza, D., Carta, S., Benini, L., De Micheli, G., Raffo, L., "Designing Application Specific Network on Chips with Floorplan Information," *Proc. ICCAD*, pp. 355-362, 2006.
- [6] Dimitriu, V., and Khan G. N., "Throughput-Oriented NoC Topology Generation and Analysis for High Performance SoCs," *IEEE Trans. VLSI Sys.*, vol. 17, no. 10, pp. 1433-1446, 2009.
- [7] Ogras, U., Bogdan, P., Marculescu, R., "Analytical Approach for Network-on-Chip Performance Analysis," *IEEE Trans. CAD Integr. Circuits Syst.*, vol. 29, no. 12, pp. 2001-2013, 2010.
- [8] Rahmati, D., Kiasari, A. E., Hessabi, S., Sarbazi-Azad, H., "A Performance and Power Analysis of WK-Recursive and Mesh Networks for Network-on-Chips," *Proc. Int'l. Conf. Computer. Design*, pp. 142-147, 2006.
- [9] Wang, H. S., Peh, L. S., Malik, S., "A Technology-aware and Energy Oriented Topology Exploration for On-chip Networks," *Proc. DATE*, pp. 1238-1243. 2005.
- [10] Duato, J., Yalamanchili, S., Ni, L. M., "Interconnection Networks – An Engineering Approach," *Morgan Kaufmann Publishers*, 2003.
- [11] De Micheli, G., Benini, L., "Networks on Chips: Technology and Tools," *Morgan Kaufmann Publishers*, 2006.
- [12] Zhang, W., Hou, L., Wang, J., Geng S., Wu, W., "Comparison Research between XY and Odd-Even Routing Algorithm of a 2-Dimension 3X3 Mesh Topology Network-on-Chip," *Intelligent Systems GCIS*, pp.329-333, 2009.
- [13] Nilsson, E., Millberg, M., Oberg, J., Jantsch, A., "Load Distribution with the Proximity Congestion Awareness in a Network on Chip," *Proc DATE*, pp. 1126- 1127, 2003.
- [14] Glass, C. J., Ni, L., "The Turn Model for Adaptive Routing", *Proc. ISCA*, pp. 278-287, 1992.

- [15] Agarwal, A., Iskander, C., Shankar, R., "Survey of Network-on-Chip (NoC) Architectures & Contributions," *Journal of Engineering, Computing and Architecture*, vol. 3 no. 1, pp. 15, 2009.
- [16] Duato, J., "A Necessary and Sufficient Condition for Deadlock-Free Adaptive Routing in Wormhole Networks", *IEEE Trans. Parallel and Distributed Systems*, vol. 6, no. 10, pp. 1055-1067, 1995.
- [17] Samadi, K., Kahng, A., Li, B., Peh, L., "ORION 2.0: A Fast and Accurate NoC Power and Area Model for Early-Stage Design Space Exploration," *GSRC Annual Symposium*, 2008.
- [18] Ahonen, T., Siquenza-Tortosa, D., Bin, H., Nurmi, J., "Topology Optimization for Application-Specific Networks-on-Chip," *Proc. SLIP*, pp. 53 – 60, 2004.
- [19] Beraha, R., Walter, I., Cidon, I., Kolodny, A., "Leveraging Application-Level Requirements in the Design of a NoC for a 4G SoC – a Case Study," *Proc. DATE*, pp. 1408 – 1413, 2010.
- [20] Ascia, G., Catania, V., Palesi, M., "Multi-Objective Mapping for Mesh-Based Networks-on-Chip Architectures," *Proc. IEEE IFIP*, pp. 182-187, 2004.
- [21] Leary, G., Chatha, K., Srinivasan, Mehta, K., "Design of Network-on-Chip Architectures with a Genetic Algorithm-Based Technique," *IEEE Trans. VLSI Systems*, vol 17, no. 5, pp. 674-687, 2009.
- [22] Arjomand, M., Sarbazi-Azad, H., Amiri, S. H., "Multi-objective Genetic Optimized Multi-processor SoC Design," *Proc. Int'l Symp. System-on-Chip*, pp. 69–72, 2008.
- [23] Guindani, G., Reinbrecht, C., Da Rosa, T., Moraes, F., "Increasing NoC Power Estimation Accuracy Through Rate-Based Model," *IEEE NoC Symposium*, pp. 89, 2009.
- [24] Murali, S., Meloni, P., Angiolini, F., Atienza, D., Carta, S., Benini, L., De Micheli, G., Raffo, L., "Designing Application Specific Network on Chips with Floorplan Information," *Proc. ICCAD*, pp. 355-362, 2006.
- [25] Srinivasan, K., Chatha, K.S, and Konjevod, G., "Linear-programming Based Techniques for Synthesis of Network-on-Chip Architectures," *IEEE Trans. on VLSI Syst.*, vol 14, no. 4, pp. 405-420, 2006.
- [26] Srinivasan, K., Chatha, K., "A Low Complexity Heuristic for Design of Custom Network-on-Chip Architectures," *Proc. DATE*, pp. 1-6, 2006.
- [27] Dally, W., Seitz, C., "Deadlock-Free Message Routing in Multiprocessor Interconnection Networks", *IEEE Trans on Computers*, vol. 36, no. 5, pp. 547-552, 1987.
- [28] Dally, W., Towles, B., "Principles and Practices of Interconnection Networks", *Morgan Kauffman Publishers*, 2004.
- [29] Linder, D., Harden, J., "An Adaptive and Fault Tolerant Wormhole Routing Strategy for k-ary n-cubes," *IEEE Trans. On Computers*, vol. 40, no.1, pp. 2 -12, 1991.

- [30] Van den Brandt, J. W., Ciordas, C., Goosens, K., Basten, T., "Congestion-Controlled Best-Effort Communication for Networks-on-Chip," *Proc. DATE*, pp. 948-953, 2007.
- [31] Deniziak, S., Tomaszewski, R., "Contention-Avoiding Custom Topology Generation for Network-on-Chip," *International Symposium DDEC*, pp. 234-237, 2009.
- [32] Bjerregaard, T., Mahadevan, S., "A Survey of Research and Practices of Network-on-Chip," *ACM Computing Surveys*, vol. 38, no. 1, 2006.
- [33] Huang, T., Ogras, U., Marculescu, R., "Virtual Channels Planning for Networks-on-Chip," *Symp. ISQED*, pp. 879 – 844, 2007.
- [34] Rezazad, M., Sarbaziazad, H., "The Effect of Virtual Channel Organization on the Performance of Interconnection Networks," *IEEE IPDPS*, vol. 15, pp. 264, 2005.
- [35] Mullins, R., West, A., Moore, S., "Low-Latency Virtual Channel Routers for On-Chip Networks," *IEEE Symposium on Computer Architecture*, pp. 188, 2004.
- [36] Chatha, K., Srinivasan, K., Konjevod, G., "Automated Techniques for Synthesis of Application-Specific Network-on-Chip Architectures," *IEEE Trans. CAD of Integr. Circuits Syst.*, vol.27, no.8, pp.1425-1438, 2008.
- [37] Ogras, U., Marculescu, R., "Application-specific Network-on-Chip Architecture Customization via Long-Range Link Insertion," *Proc. ICCAD*, pp. 246- 253, 2005.
- [38] Hu, J., Deng, Y., Marculescu, R., "System-level Point-to-Point Communication Synthesis using Floorplanning Information," *Proc. ASPDAC*, pp.573-579, 2002.
- [39] Ogras, U., Marculescu, R., "Energy- and Performance-Driven NoC Communication Architecture Synthesis Using a Decomposition Approach," *Proc. DATE* , pp. 352- 357, 2005.
- [40] Glover, F., M. Laguna, "Tabu Search", *Kluwer Publishers*, Norwell, MA. 1997.
- [41] Xu J., Chiu S., Glover F., "Optimizing a Ring-Based Private Line Telecommunication Network Using Tabu Search," *Management Science*, Vol. 45, No. 3, pp. 330-345, 1998.
- [42] Woodside, C., Nelson, J., Petriu, D., Majumdar, S., "The Stochastic Rendezvous Network Model for Performance of Synchronous Client-Server Like Distributed Software," *IEEE Trans On Computers*, vol.44, pp. 20-34, 1995.
- [43] Samadi, K., Kahng, A., Li, B., Peh, L., "ORION 2.0: A Fast and Accurate NoC Power and Area Model for Early-Stage Design Space Exploration," *GSRC Annual Symposium*, 2008.
- [44] Adya, S. N., Markov, I. L., "Fixed Outline Floorplanning: Enabling Hierarchical Design," *IEEE Trans. VLSI Sys.*, vol. 11, no. 6, pp. 1120-1135, 2003.

- [45] Ye, T. T., Benini, L., De Micheli, G., "Analysis of Power Consumption on Switch Fabrics of Network Routers," *Proc. DAC*, pp. 524-529, 2002.
- [46] Banjaree, N., Vellanki, P., Chatha, K., "A Power and Performance Model for Network-on-Chip Architectures", *Proc. DATE*, vol. 2, pp. 1250 – 1255, 2004.
- [47] Murali, S; De Micheli, G., "SUNMAP: a tool for automatic topology selection and generation for NoCs," *Proc. DATE*, pp. 914- 919, 2004
- [48] Dumitriu, V., "Network-on-Chip Topology Generation and Analysis for Transaction-Based Systems-on-Chip", *MASc Thesis, Ryerson University*, 2008.
- [49] Hu, J., Marculescu, R., "Energy-Aware Mapping for Tile-Based NoC Architectures Under Performance Constraints," *Proc. ASPDAC*, pp. 233-239, 2003.
- [50] Pasricha, S., Dutt, N., Bozorgzadeh, E., Ben-Romdhane, M., "FABSYN: Floorplan-aware Bus Architecture Synthesis," *IEEE Trans. VLSI Sys.*, vol. 14, no. 3, pp. 241-253, 2006.
- [51] Hu, J., Ogras, U., Marculescu, R., "System-Level Buffer Allocation for Application-Specific Networks-on-Chip Router Design," *IEEE Trans. CAD of Integr Circuits and Syst*, vol.25, no.12, pp.2919-2933, 2006.
- [52] Srinivasan, K., Chatha, K.S., "Layout aware design of mesh based NoC architectures," *Proc. CODES+ISSS*, pp.136-141, 2006
- [53] Rahmati, D., Murali, S., Benini, L., Angiolini, F., De Micheli, G., Sarbazi-Azad, H., "A Method for Calculating Hard QoS Guarantees for Networks-on-Chip," *Proc. ICCAD*, pp.579-586, 2009.
- [54] Seiculescu, C., Murali, S., Benini, L., De Micheli, G., "SunFloor 3D: A Tool for Networks on Chip Topology Synthesis for 3-D Systems on Chips," *IEEE Trans. CAD of Integr. Circuits Syst*, vol.29, no.12, pp.1987-2000, 2010.
- [55] Alho, M., Nurmi, J., "Implementation of Interface IP for Proteo Network-on-Chip," *Int'l Wrks Design Diagnostic for Electr. Circuit Syst.*, 2003.
- [56] worm_sim *Cycle-Accurate NoC Simulator* [online] Available: http://www.ece.cmu.edu/~sld/software/worm_sim.php
- [57] Srinivasan, K., Chatha, K., Konjevod, G., "An Automated Technique for Topology and Route Generation of Application Specific on-Chip Interconnection Networks," *Proc. ICCAD*, pp. 231-237, 2005.
- [58] Jaspers, E., de With, P., "Chip-Set for Video Display of Multimedia Information," *IEEE Trans. on Consumer Electronics*, vol. 45, pp. 706-715, 1999.
- [59] Mishra, A. K., Yanamandra, A., Das, R., Eachempati, S., Iyer, R., Vijaykrishnan, N., Das, C., "RAFT: A Router Architecture with Frequency Tuning for On-Chip Networks", *Journal of Parallel and Distributed Computing*, (In Press), 2011.

- [60] Sethuraman, B., Vemuri, R., "optiMap: a Tool for Automated Generation of NoC Architectures Using Multi-port Routers for FPGAs," *Proc. DATE*, pp. 6-10, 2006.
- [61] Jeong, K., Kahng, A.B., Lin, B., Samadi, K., "Accurate Machine-Learning-Based On-Chip Router Modeling," *IEEE Embedded Systems Letters*, vol.2, no.3, pp.62-66, 2010.
- [62] Shen, W., Chao, C., Lien, Y., Wu, A., "A New Binomial Mapping and Optimization Algorithm for Reduced-Complexity Mesh-Based On-Chip Network," *Proc. Int'l Symp NOCS*, pp.317-322, 2007.