Mathematical Model based Optimized Custom Network on Chip Design

By
Sajid Gul Khawaja
[2009-NUST-MS PhD-ComE-09]

Submitted to the Department of Computer Engineering
In partial fulfillment of the requirements for the degree of

Master of Science
In
Computer Engineering

Advisor
Col. Dr. Shoab Ahmed Khan

College of Electrical & Mechanical Engineering
National University of Sciences and Technology
2011

DECLARATION

We hereby declare that no portion of the work referred to in this Project Thesis has been submitted in support of an application for another degree or qualification of this of any other university or other institute of learning. If any act of plagiarism found, we are fully responsible for every disciplinary action taken against us depending upon the seriousness of the proven offence, even the cancellation of our degree.

COPYRIGHT STATEMENT


بسم الله الرحمن الرحيم

This thesis is dedicated to my parents

Acknowledgements

All praise to the Almighty Allah, the most Merciful and the most gracious one. Without whose help and blessings, I would not have been able to complete this research. Many thanks to my project supervisor, **Dr. Shoab Ahmed Khan**, whose constant motivation, unflagging efforts and uninvolved words of wisdom ever proved a lighthouse for me; it was earnestly felt whenever we swayed. Despite his never ending assignments of university management, student counseling, project supervision and teaching, he did never mind whenever I went for an advice, within or without the time slot allocated.

Special thanks to **Mr. Mian Hamza Mushtaq** who was ever present to provide his guidance and help to me, from any complex to trivial problem.

Acknowledgement is also due to my teachers for dedicatedly instilling and imparting enlightenment to me during the course of studies and afterwards for our project. I am also very thankful to my parents for their tacit and avowed support, patience and understanding.

I would like to thank my friends who gave me confidence to face the difficulties of life. They all gave me good company and everlasting memories.

Abstract

System on chip (SoC) architecture has become more popular as the number of components on a given chip increase. The most important aspect of this architecture is the ability to integrate various heterogeneous components on a single architecture, this requires abstraction and modularity. Another important aspect of this architecture is the methods by which the various components communicate with one another. Network on Chip (NoC) architectures attempt to address these aspects by providing various component level architectures with specific interconnection network topologies and routing techniques.

The main aim of the thesis is to design an optimal Network on Chip (NoC) architecture for custom based applications e.g. high data rate communication systems, complex pattern recognition and video processing systems etc. The various aspects of NoC like number of processing elements, their placement, interconnects width, packet size etc are to be optimized. For designing, the architecture of NoC will be mathematically modeled using integer, linear, or non linear programming techniques. While the application, to be mapped on the architecture, will be modeled as a graph partitioning problem. The main aim of modeling the application would be to find the placement of process entities on the NoC architecture for optimal performance. The mathematical model will be solved using any off the shelf programming solver like GUSEK, LINDO/LINGO etc. The results of the solver will be parsed to extract the solution and specifying the design of NoC.

Table of Contents

Contents

List of Figures

List of tables

List of Equations

List of Abbreviations

NoC:	Network on Chip
SoC:	System on Ship
LP:	Linear Programming
MILP:	Mixed integer linear programming
VC:	Virtual Channel
QoS:	Quality of Service
OPF:	Output flows

Chapter 1 : Introduction

According to the reports by International Technological Roadmap for Semiconductors (ITRS) [12], by 2022 integrated circuits will be implemented in less than 11nm technology, allowing the designers to place several storage cores and computation units as processing entities of System-on-Chip (SoC). This advancement in technology will bring communication problems among several PE's of SoC since the signal propagation will span multiple clock cycles. SoC has added to the design intricacy thus requiring the need for more sophisticated design methodologies. The designers are required to provide efficiency through highly optimized designs. Nowadays high capacity, simple design and the low cost features are the key requirements. Many embedded systems are designed for data intensive application where frequent data transfer between the PE's is unavoidable, therefore interconnect method between these PE's, for a SoC, has become a more challenging problem and a key concern, for SoC architectures, as the performance requirement is increasingly demanding.

In the past, on-chip communication between modules was mostly based on the conventional bus based architecture where a single bus is shared between various slaves and master nodes. Bus based communication structure is only feasible for a system with small number of modules. However, a recent system requirement of high transfer of data between multiple modules across the chip cannot be serviced by bus based interconnects. Several works have proposed enhancement of the traditional on-chip buses to keep up with the performance requirements. To increase the performance of the traditional bus efficient arbitration techniques are proposed in [13, 14]. The performance of system is highly limited because of the very nature of shared buses in spite of their effectiveness. Scalability of the shared bus is the key limiting factor in the performance of systems.

Performance of the shared bus is not scalable with respect to the number of components being attached to the bus.

Point-to-point is another classical communication scheme used in the past. With the increase in the number of system modules on a single chip point to point cannot provide a feasible approach for communication. PTP lacks scalability and flexibility which acts as a bottleneck in high density system. Network-on-Chips (NoC) have been developed to serve the needs of such large system. Figure 1 shows an example of bus based, point-to-point and Network in Chip communication structures.

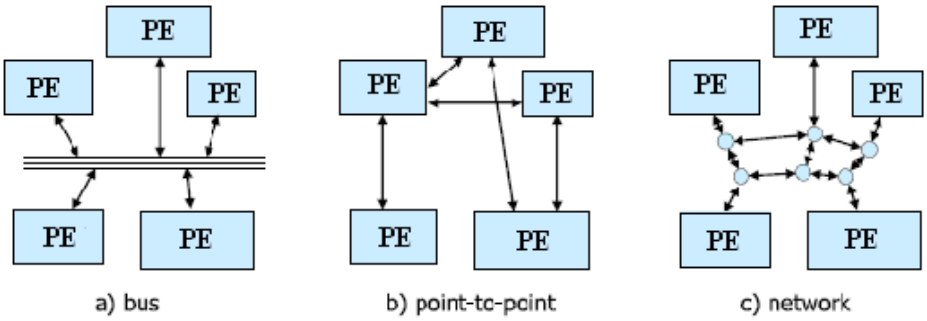

Figure 1.1 Example of communication structures in Systems on Chip a) Traditional bus-based communication b) Dedicated point to point links c) a chip area network

Network on chip proves to be a viable solution that can cope with the performance and the scalability issues. Since the beginning of this century NoC has been proposed as a new communication infrastructure to overcome these issues. Moreover NoC architecture is being actively studied in academia [14, 15, 16, and 17]. NoC architectures mimic the traditional interconnection network concepts on a single chip and several design techniques are adopted from it.

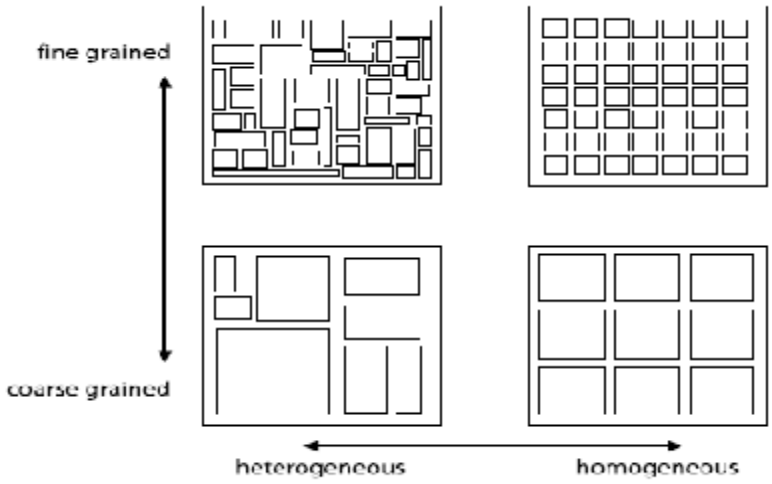

Figure 1.2 Example NoC structure showing fine grain, coarse grain, homogenous and heterogeneous designs.

Performance of network on Chip architectures is greatly dependent upon the topology being used, routing algorithm being adopted placement of processing entities in the NoC etc.

Routing Schemes:

The routing algorithm is one of key research area of a NoC design. NoC's performance, complexity, power consumption etc are greatly affected by the selection of different routing algorithm. A lot of work [4, 2] has been done to provide a routing scheme that offers improved QoS while keeping the design of the switch as simple as possible. Wang Zhang et al, compares common XY and Odd-even (OE) routing algorithms in [4]. Wang Zhang et al proposed OE routing algorithm in [4] which is complex to implement but provides better deadlock-free to the network. A priority based switch design proposed in [2] provides deadlock-free network at the cost of reduced synthesis frequency and an increase in area consumption with respect to the basic XY switch. Thus different routing schemes can provide different latency, throughput and success rate at the cost of area and synthesis frequency of the architecture [2].

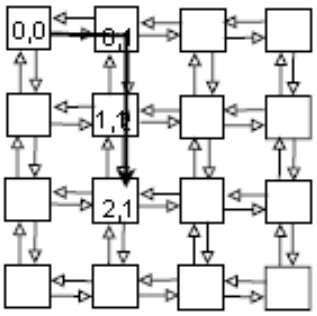

Figure 1.3 Example of XY routing scheme in use

Network Topology:

NoC architectures mimic the traditional interconnection network concepts on a single chip and several design techniques are adopted from it. NoC architectures can be constructed by using either regular topologies or irregular (custom) topologies. The decision to use a certain topology lies with the designer; this decision is generally governed by the application to be mapped on the NoC architecture or more specifically the purpose to which the NoC architecture is being designed for. For a generic NoC architecture regular network architectures are sufficient here the decision to use a certain topology plays a vital part in the performance of the NoC architecture. Regular topologies or irregular (custom) topologies are used for constructing NoC architecture. Both topologies have advantages and disadvantages over one another: Irregular topologies are will suited for the optimization of different requirements such as link size, number of router to be used etc. for the applications where cores are heterogeneous in size and functionality and also demand different communication bandwidth. However regular topologies are easy to design and provide reusability. Regular topologies are employed by most of the multi-core architectures, especially mesh topology. Intel's Teraflop Chip [35] that has 80 cores is connected via 2D mesh network. Dietmar Tutsch et al, shows, in [5], that different topologies outperform other on the bases of scalability, performance of the architecture the buffer cost etc.

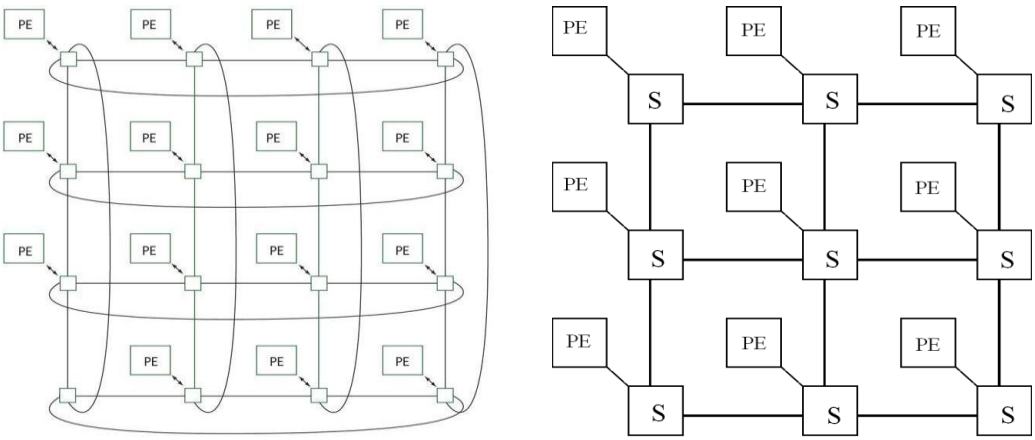

Figure 1.4 Different NoC topologies a) Torus topology b) Mesh topology

A generic NoC architecture can't provide the same level of performance for every application. Thus one needs an irregular network architecture that is in consistence with the application to be mapped on the NoC architecture. Irregular NoC architecture means that the network architecture is not symmetric. The formation of the topology, connection of PE etc is generally governed by the application to be mapped on that architecture.

The following shows an irregular network where the positioning of switches, PE etc is made on the bases of some feature or another. This hybrid interconnects is called an irregular network.

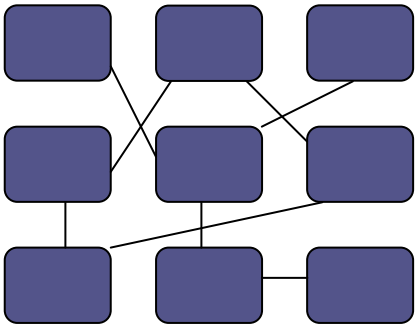

Figure 1.5 Example of an irregular NoC architecture

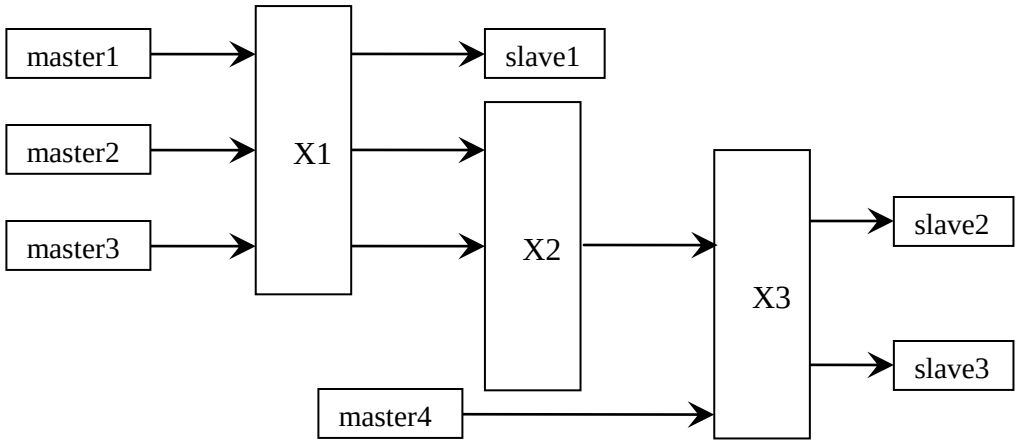

Figure 1.6 Example of an irregular NoC architecture

Among the available flow control strategies wormhole routing stands out as the most preferred one for the application specific topologies. Wormhole routing provides low latency and buffer requirements for the NoC design. Among the many benefits of wormhole routing scheme contention is the major bottle neck of wormhole routing scheme as packets trend to scatter throughout the network during transmission. Main cause of contention within a network occurs when a particular resource is required by more than one packet in the network at a particular moment in time, this can lead to congestion of the contention point propagates throughout the system. Congestion cause s degradation in the systems performance thus the system can suffer from long delays and may not meet the throughput requirements of the system (application).

Motivation and Outline/Goal

This work is motivated by our interest in developing a mathematical model for application specific NoC. Mathematical modeling (MM) provides optimal or near to optimal solution for any model. MM has provided optimized solution for worldwide problems e.g. USA Shell Distribution System [1].

Linear programming (LP) is a component of the more general field of mathematical programming. Mathematical programming is concerned with the development of modeling and solution procedures for the purpose of maximizing the extent to which the goals and objectives of the decision maker are realized. Very special condition must hold true before a general mathematical programming problem is actually an LP problem.

Despite the implications of its name, LP has little to do with computer programming. In LP, the word programming is related to planning. Specially, it refers to modeling a problem and subsequently solving it by mathematical techniques. LP is very similar to setting up and solving a system of linear equations.

Linear programming has been in use by many of the real world applications. United States Shell Oil Company has utilized linear programming codes together with the model builders and report generators to better manage their distribution system. The linear programming model that they developed consists of 575 constraints and 1700 variables. The linear programming model took 8 months to develop; one-fourth of this time was for getting acquainted with the people involved, the system, and sources of data. The cost of running the complete system using IBM's MPSX LP package on an IBM 370/168 computer varies from only $15 to $30, depending on the time of day. [1]

Effective results for the same problem set can be obtained by writing algorithms, but it can fail to provide an optimal solution for the problem. Moreover algorithms are complex to design in comparison to mathematical models. Mathematical model for a NoC covers a number of aspects.

This thesis works its way around to present a model that provides optimized, i.e. low area utilization, NoC architecture parameters by keeping in view the wormhole routing scheme that uses virtual channels. This thesis also keeps in view the traffic flow per switch and provides a feasible solution that does not exceed the frequency requirement of each switch. As a starting point static power consumption of the network is also considered in the model.

Chapter 2, literature review, provides the necessary overview of Mathematical modeling; NoC and IP based NoC architectures being already designed. Chapter 3, Mathematical model of our problem, presents the mathematical model of the problem. Chapter 4, implementation and results, discussed how this model was implemented and discusses the results of the model. Chapter 5, conclusion and future work, concludes this thesis and outlines feature which can be incorporated in the model for better results.

Chapter 2 : Literature Review

In terms of contention, Hu et al. attempt to improve performance by reducing the queueing blocking probability at each buffer [18]. No virtual channels (VC) are considered to relieve contention within the network. Aethereal implement routers to provide both guaranteed and best effort services and include VC insertion [19]. However, due to the complexity of the routers, area and power consumption stand as potential problems which require extra arbitration and buffering considerations. Rezazad and Sarbaziazad and Mullins et al. implement uniform VC distribution within on-chip networks [20, 21]. This technique does not optimize for power and performance, and can be a potential waste of resources for the network components that are not heavily utilized. Van den Brand et al. suggest hardware probes to monitor link congestion in real time [22]. Thismethod 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. It is much more efficient to relieve contention during the topology synthesis stage since the connections between the cores are well defined.

A lot of research is under way to find an optimal solution for the synthesized topology considering a number of NoC aspects in view e.g. synthesis frequency, power consumption, performance, area consumption etc.

A number of approaches have been adopted in the recent past to find the optimized solution. Generally they involve

- Algorithmic approaches
- Mathematical Modeling

Power and power related factors during topology synthesis have been incorporated by a number of researchers. Others have focused on eliminating contention through various means. Static floorplans [23,249] have been used by several researchers in for power based NoC synthesis design where they assume that routers area placed at the corner of

all the cores. Topology synthesis may not consider network components such as NI's and routers. In [23] performance issues have been discussed by Leary et al. while in [24] power based issues have been addressed by Chatha et al... A number of other research factions have formed mesh topologies and have used the energy bit approach to evaluate power consumption within the on- chip network [25, 18, 27]. Energy bit approach is successful only in the first order estimation, as power estimate model can surpass 50% error margin due to traffic congestion [28]. Ogras and Marculescu propose a technique to customize mesh based topologies by long-range link insertion to relieve critical loads [26]. This work assumes a constant set link length, while multiples of these lengths are used to insert the long-range links. Some of these researchers assume predetermined wirelengths and do not incorporate a floorplanner to estimate the actual lengths which can have a significant affect on power dissipation [26,25,18].

Simulated annealing (SA) method has been used by Ahonen et al. [29] to optimized NoC synthesis with the help of a design automation tool OIDIPUS. This technique exhibits that SA and system partitioning degrade the attainable level of optimization via block placement due to the lack of freedom that the blocks deserve when determining a position in the topology. Traffic flow within resulting partitions is increased and the boundaries of possible solutions in the search space are limited by aiming to lower the cost of partitioning. This method can only use one objective function at a time, and only considers the wirelengths within the system while disregarding factors such as router consumption while optimizing for power.

Ascia et al. [30] use power and performance as objectives in order to optimize mesh based NoC architecture using a state-machine approach. A NoC simulator evaluates mapping alternatives and the objectives to be optimized. The exploration engine evaluates the next produced core mapping scheme using a Genetic Algorithm (GA) to decide whether to allow the next move or not, until a stopping criteria is met. The method fails to consider the wiring costs on power, and overall performance issues during the topology design. Leary et al. [31] implement a 3-level GA to represent each topology mapping as a set of binary and integer arrays. The algorithm employs the reproduction, crossover and mutation techniques until a stopping criterion is met. The algorithm also

invokes a fitness function and a floorplanner to map the elements as an application-specific architecture. Both GA based techniques generate either performance or power solutions based on a Pareto curve, where the designer is expected to choose the best option between the two objectives. In general, GAs possess long execution times and do not know how to sacrifice short-term fitness to gain longer-term fitness. For this reason, GAs often fall into local optimal as opposed to finding the global optimum. In general, if the core graph fed into the topology generator requires directed edges from all vertices, all of the core sources and destinations are well known. Therefore, if the NoC generator is aware of the communication requirements within a system, there is no logic in randomly generating an initial solution or to keep generating random solutions. SoCs are expected to deal with hundreds of cores, and hence using a GA can increase execution time and may not necessarily lead to an optimal global solution.

Murali et al. [33] has designed application-specific topologies by integrating floorplanner to assess the power consumption on wires, and establish whether the NoC can function at a certain frequency given a set of design parameters by the user. It also tries to integrate dynamic effects of congestion via a user specific mismatch parameter in order to reduce execution time. From a design viewpoint however, it is much more suitable and less time consuming if all dynamic effects, as well as power and performance factors are taken care of by the generator itself.

Algorithmic Approach:

TABU SEARCH

The Tabu search (TS) [11] is a meta-heuristic algorithm designed to escape the trap of local optimality [32]. Contrary to other optimization methods such as Genetic Algorithms and Simulated Annealing, the feature that distinguishes the TS from other algorithms is its adaptive ability to retrieve prior optimal and non-optimal solutions by using a Tabu list. If the method falls into a cycle of moves that do not improve, a backtracking process starts by reverting back to some local optimum found in the Tabu list, and once again attempts to search for a new optimal solution.By keeping track of non-improving factors

within a topology, it is possible to design an optimal or near optimal system. The general TS algorithm is presented in Fig. 1.

```
Generate initial solution N(s)
Evaluate current solution conditions
WHILE stopping criteria NOT met
DO
Identify a new neighbourhood solution s'
Move to the temporary solution
Evaluate current solution
Refer to Tabu List T(s) && Aspiration List A(s) to check
for optimality
IF optimal solution
Place solution as an optimal T(s) entry
Update current solution, N(s) = s'
IF Constraints satisfied
EXIT
END
ELSE
Place as a non-optimal T(s) entry
Refer to Aspiration List A(s) to revert back to last
optimal solution
END
END WHILE
```

Figure 2.1 Example of an irregular NoC architecture

Current Research using Mathematical Modeling

Suleyman Tosun et al. has presented a new Integer Linear Programming based application mapping tool for meshbased Network-on-chip architectures in [9]. The tool shows the results for multimedia benchmarks to find close to optimal results under given

amount of time. The work aslo demonstrated the effects of the mesh sizes on the final mapping.

The GRID based topology requires more routers than the MILP based topology showing that the design methodology of NoC architectures is crucial for achieving high performance. Ben A. Abderazek et al. presented a new mathematical formulation for synthesize of application specific NoC architectures, such that the performance constraints are satisfied and the communication power consumption is minimized [8].

Dara Rahmati et al. has proposed two different methods to characterize bandwidth and latency for NoC-based real-time SoCs, aiming at guaranteed QoS provisions [10]. The choice of the most suitable method depends on the performance demands of the system and on whether dedicated hardware facilities can be supplied in the NoC. One method is aimed at applications demanding minimum latencies and requires injection regulation, while the other is suitable for applications where packet injection must be flexible to accommodate for higher average injected bandwidths and no hardware regulation is available.

Drawbacks

Although the TS technique is successful in finding optimal solutions, its downfall occurs due to memory allocation. As more neighborhood set solutions are incurred, the amount of memory needed to memorize solutions can become quite large and demanding. Therefore, the user is to specify the largest amount of entries allowed within the Tabu list. Thus when the list becomes full, it updates itself with the newest optimal/non-optimal solution and acts as a FIFO, releasing its oldest andmost non useful data. Given a problem space with a predetermined amount of cores and edges within a topology, the user can estimate the required memory allocation needed for a successful optimal solution to be reached.

Currentlty much of the work in the field of MILP/IP based optimized NoC has taken place either without the consideration of virtual channels or the focus has been on providing optimized solutions keeping in view the performance or power issues [34]. Most if the designs are not focused towards Cost Area minimization.

This thesis presents a mathematical model for NoC architecture that gives an optimized solution keeping in veiw the area considerations along with the requirements of virtual channels required in the NoC design.

Chapter 3 : Proposed Model

In this thesis a mixed integer linear programming (MILP) model for network-on-chip architecture is formulated. Primary concerns of the model are to provide an optimized solution complying with the specifications provided by the application (user). MILP models come under the umbrella of mathematical programming which in essence is the art of presenting a model in terms of equations. Mathematical programming provides an efficient way of finding an optimized solution to any given problem. Linear, non-linear and integer programming are an essential part of mathematical programming [1]. MILP model basically represents a problem in linear expressions whose model contains integer and binary variables. Integer variables provide numerical solution on the basis of our problem and constraint set whereas binary variables act as decision variables for the model.

Main motivation for using MILP, as a programming technique, for our NoC problem arises from our need of different decision variables. These decision variables form the crux of our model and are integral part of our constraints.

The proposed system consists of four phases namely

- Requirements
- Model Code Generation
- Mathematical Modeling
- Verilog Code Generation

The application requirements include master, slave, number of switches and traffic pattern for the system as well as frequency and bandwidth requirement per switch. Model code generator gives expressions for link, virtual channel and traffic flow once the

number of channels is given as input. Mathematical modeling takes the application requirements as input and returns the topology connections. Verilog code generator is provided with parameters such as number of switches, packet size, number of input output ports and traffic flow; and the required Verilog code is generated subsequently. The next section explains in detail the fourth phase of the system.

This section explains how the model for custom NoC architecture was achieved while adhering to the above mentioned guidelines.

Before moving on it should be noted that some decision variables proposed by Minje et al, in [3] are adopted in our model as the starting point of this research. Addition constraints and variables are integrated with the ones already presented in [3] to from an optimized solution.

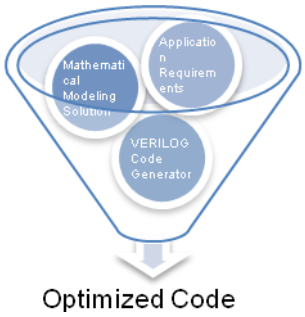

Figure 3.1 Achieving optimized design

The overall implementation of the thesis is divided into the four different stages. These stages are depicted in the figure 3.2

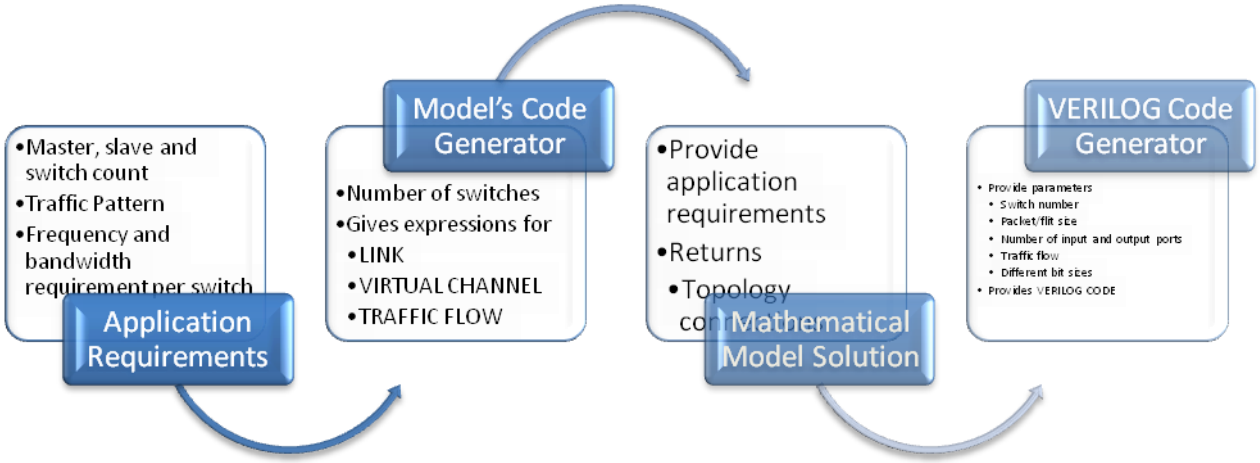

Figure 3.2 Implementation phases of the thesis

The main aim of the thesis is to find an optimized solution for the NoC design given a specific application thus the first part of our thesis is to get the application requirements such as

- Master, Slave and Switch number
- Traffic flow
- Frequency
- Bandwidth requirements etc

The next part of the thesis aims at finding some of the expressions that are the part of MILP for the given number of switches. The most important part of the thesis is the MILP model that focuses on finding an optimized solution for the provided application requirements. Final part of the thesis is aimed to find the verilog code for NoC switch using the information provided by the MILP solution such as:

Switch number

- Traffic Flow pattern
- Virtual channel count
- Packet size
- Header bit length etc
- Number of input and output links

Model formulation:

MILP model formulation is the main focus of this thesis. MILP model formulation works to find an optimized solution provided some specific condition are met using the application requirements.

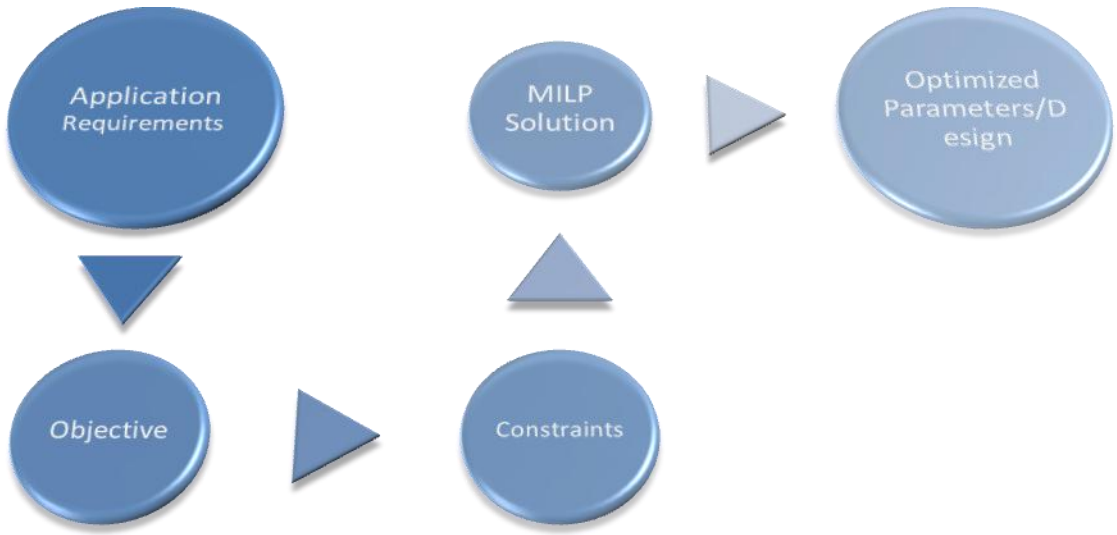

Figure 3.3 How mathematical model helps in achieving optimized design

In order to formulate an LP model successfully, the decision maker must:

1. Understand the problem
2. Identify the decision variable
3. Numerical measure of effectiveness for the objective function to be chosen

4. Measure of effectiveness involving the decision variables to be represented as a linear expression
5. All constraint to be identified and represented as linear expressions involving the decision variables
6. Data collection or appropriate estimation for all the parameters of the model is required [1]

Understand the Problem:

Our method aims at providing an optimized, synthesizable, switch network for a custom application. Objective function of out model is to minimize the area consumption of our network. It is assumed that custom traffic, frequency and bandwidth information is provided.

Assumption2 single communication path exists between a pair of master and slave.

Provided; number of masters, number of slaves, number of switches, latency requirements, traffic flow, frequency requirement per switch and latency.

A switch based NoC consists of three main components, namely, master, slave and switch. The network contains three types of connections; master to switch, slave to switch and switch to switch for cascaded network.

In the following subsection MILP formulation of the problem is presented. The formulation consists of 5 types of constraints including Topology, single communication path, Latency, Frequency/Traffic flow per switch and static power utilization and one design cost factor total network area.

Identify the Decision Variables:

The following decision variables are used in the model

- Master-Switch connection (MX)
  - To check the which master port is connect to the which switch
- Slave-Switch connection (SX)
  - To check the which slave port is connect to the which switch
- Switch-Switch connection (XX)
  - To check the switch-switch (cascade) connections

- Link (link)
  - To check a pair of master and slave ports form a path
- Connection between master and Slave Ports (cconnect)
  - To check whether a certain pair of master and slave ports are connected or not

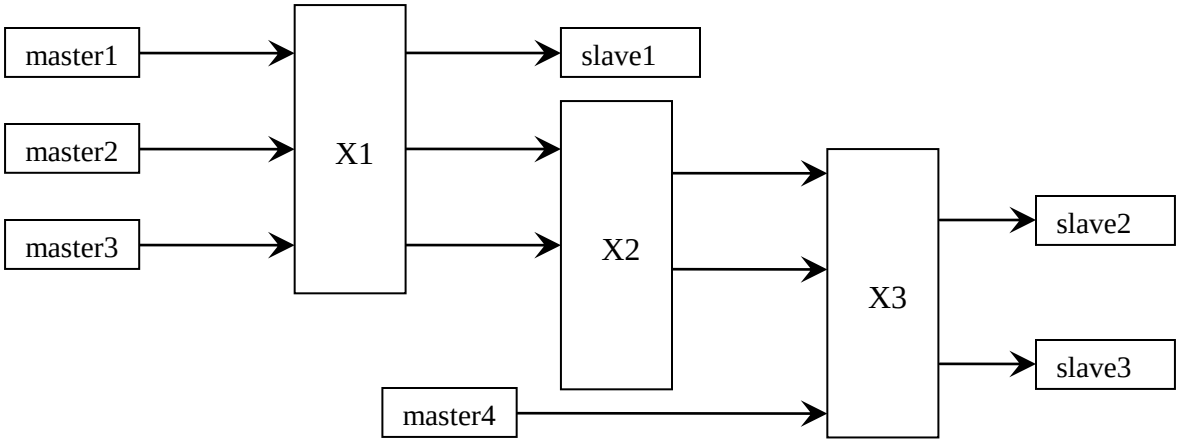

Figure 3.4 Example of an irregular NoC architecture

Numerical Measure for the Objective function:

In application specific NoC there exist a number of critical aspects which are to be considered. Area and power form a few of these concerns. Aim of the proposed model is to minimize the ***Area*** utilized by the NoC architecture. Area of NoC architecture depends on a number of features, e.g. number of switches used in the network. Number of switches alone cannot correctly identify how much area will be consumed by the NoC architecture. As switch's area consumption is directly proportional to the number of virtual channels and output flows required per switch. Thus each switch will have different area requirement.

In other words we can say that area consumption of NoC architecture is more or less directly proportional to the number of virtual channels required at each switch and number of output flows required per switch.

Thus,

Area directly proportional to VCs and OPFs

Expressing area utilization in terms of number of switches, virtual channels and outputs flows formed a non-linear expression. Thus linear regression was used to form a linear expression for area consumption in terms of virtual channels and the output flows used.
Online Linear regression tool [5] was utilized to provide us with areas linear expression.
Correlation Coefficient: refers to Goodness of fit, which refers to how well a statistical model fits observations by summarizing the discrepancy between expected values and the observed values under the model in question**.** [6]
Residual sum of squares: It is a measure of the divergence between the data and an estimation model. [6]

Linear expressions for the decision variables:

Model contains the following decision variables:

- Master-Switch Connection Decision variable
- Slave-Switch Connection Decision variable
- LINK
- CConnect

These decision variables are explained below:
Master-Switch Connection Decision variable
Expression forces the MX variable to be one when a master is connected with a given switch. Equation 3-1 represents the linear expression for MX decision variable.

Slave-Switch Connection Decision variable
Expression forces the MX variable to be one when a master is connected with a given switch. Equation 3-2 represents the linear expression for SX decision variable.

LINK
Expression forces the LINK variable to be one when a master-slave pair is forming a single path. LINK expression is dependent on the depth of cascade

No. of expressions (summation of XX) is directly proportional to n (cascade depth) as show in the equations. Equation 3-8 represents the linear expression for LINK decision variable.

CConnect

Expression forces the decision variable to be one when traffic flow exists between a pair of master and slave port else it may be 0. Equation 3-1 represents the linear expression for MX decision variable.

All constraints as linear expressions:

Complete model is dependent upon the following constraints five constraints:

- Topology Constraint
- Single Communication Path Constraint
- Latency Constraint
- Traffic Flow/Bandwidth Constraint
- Power Constraint

Topology Constraints:

Constraint 1(a):

Constraint 1 assumes that a master or a slave must be connected with only one switch because providing arbitrary connection does not provide most feasible solution. The following decision variables (MTX, STX and XTX) were proposed by Minje et al in [3]. The same decision variables are used in this model.

Constraint one says that each master and slave must only be connected with a single switch. To force this constraint MTX and STX decision variable are used. MTX contains one at the index (master number, switch number) where it is connected with a switch. Similarly STX contains one at the index (master number, switch number) where it is connected with a switch.

$$\sum_{j=1}^{x} MX_{ij} = 1$$

Equation 3-1 Constraint 1(a) for master-switch connection

$$\sum_{j=1}^{x} SX_{ij} = 1$$

Equation 3-2 Constraint 1(a) for slave-switch connection

Constraint 1 (b):

Degree of Switch must be greater than 2.

Switch network must utilize the full potential provided by the switch. Switches are best utilized when it has multiple input and output ports. A switch having a single input and a single output port are simply useless as they are no better than a single dedicated bus. Such a design lays waste to a valuable resource i.e. a switch. A switch must have either more than two input and output ports or it must not be connected to any input and output port i.e. it may not be used at all. Thus we can say that switches degree must be either greater than 2 or it must be 0. (Degree of a switch is known as the sum of input and output ports connected to that switch.) Minje Jun et al has addressed this issue in his MILP model [2]. The expressions proposed by Minje Jin et al are generally easy to understand but difficult to implement as MILP. Minje Jin et al proposed a substitute of these expressions but those expressions are totally model dependent and are difficult to understand and implement. The same decision variables, PM and PS, proposed in [3] are used in our model but the linear expression used to model the constraint have been developed at our own.

$$PM_j = \sum_{i=0}^{m} MX_{ij} + \sum_{i=1}^{x} XX_{ij}$$

Equation 3-3 Constraint 1(b) calculating input connections per switch

$$PS_j = \sum_{i=0}^{m} SX_{ij} + \sum_{i=1}^{x} XX_{ji}$$

Equation 3-4 Constraint 1(b) calculating output connections per switch

Actual constraint to force the degree of switch to be two is as follows:

$$PS_i + PM_i > 2$$

Equation 3-5 Constraint 1(b) Constraint for degree of switch > 2

Degree of a switch can be greater than if there are the input ports and no slave port or vice versa. Such a switch and its connections are useless. Thus we need to define another constraint that forces the switch to have at least one input port and one output port.

$$PM_i > 0$$

Equation 3-6 Constraint 1(b) Forcing at least one input port per switch

$$PS_i > 0$$

Equation 3-7 Constraint 1(b) Forcing at least one output link per switch

Constraint 2:

Single communication path:

Some of the available commercial solutions (PL301, SMX) for crossbar networks have a single communication path from a master to slave [3]. In the model it has also been assumed that single master to slave communication paths exits between any pair of master and slave. This approach seems to limit a pair of IP's to communicate with each other through more than one path. A solution to this problem exists, IP's can communicate with each other through multiple paths using multiple master and slave ports i.e. master 1 and master 2 can represent same IP though from our models point of view they act as independent master ports.

Consider the following switch network. In this network there exist two different communication paths through which master 1 is connected to slave 2. Same is the case with master 2 and 3.

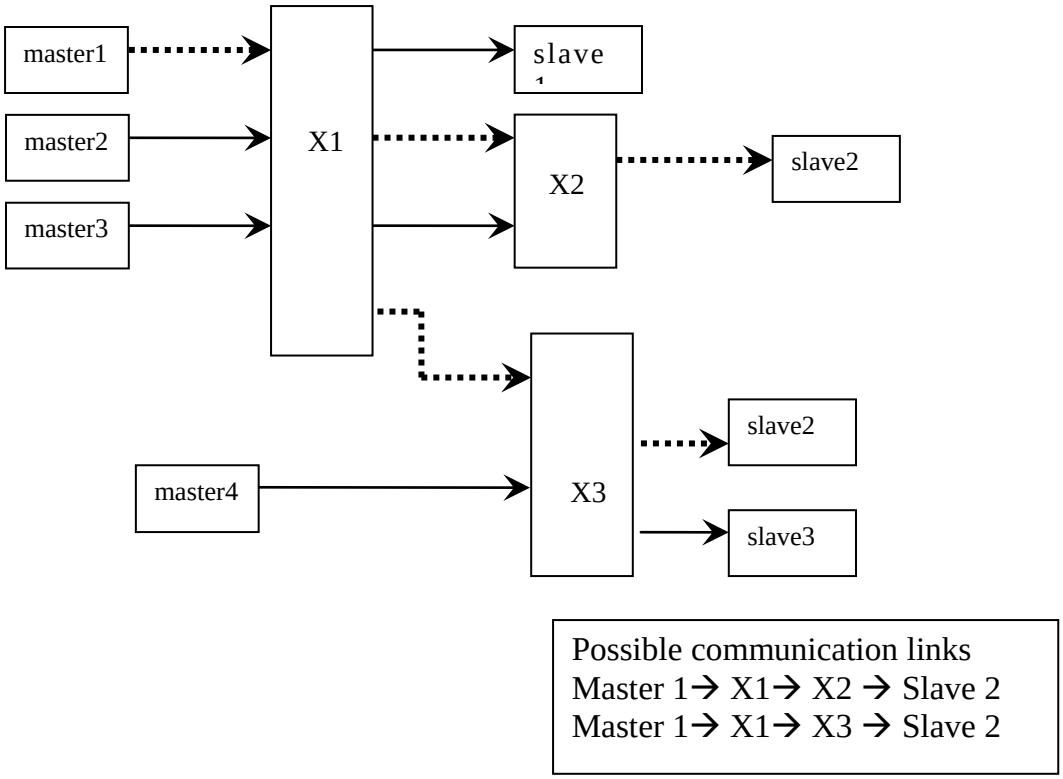

Figure 3.5 Switch network with multiple communication paths between all the master/slave pairs

Such network is contrary to our assumptions of single communication path. Constraint 3 works its way around to force that only a single communication path must exist in the network thus restructuring the network as shown in the following figure (3.6). Only a single communication path now exists between all the master and slave pairs

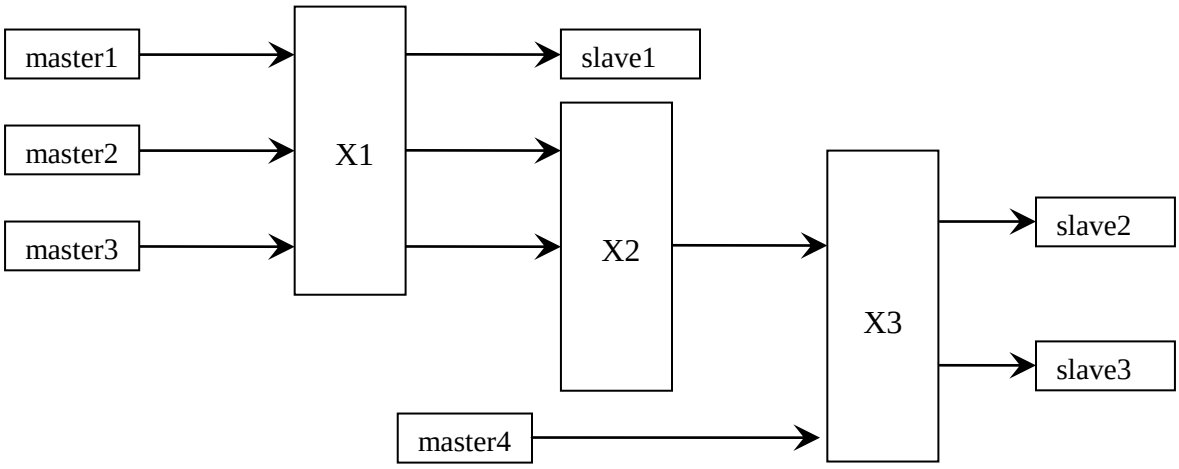

Figure 3.6 Switch network with single communication path between all the master/slave pairs

To ensure a single communication path between each master and slave pair a decision variable linked is used.

$$link_{m,x1,x2,x3,\ldots,xn,s} \geq MX_{m,x1} + SX_{s,xn} + \sum_{k=1}^{n-1} XX_{xk,xk+1} - n$$

Equation 3-8 Constraint 2 forcing the link decision variable to be one in case a pair of master slave ports are connected

Constraint 3:

Traffic Flow optimization:

Traffic flow becomes a major factor when one is looking for a viable solution for a set of given input traffic and frequency per switch.

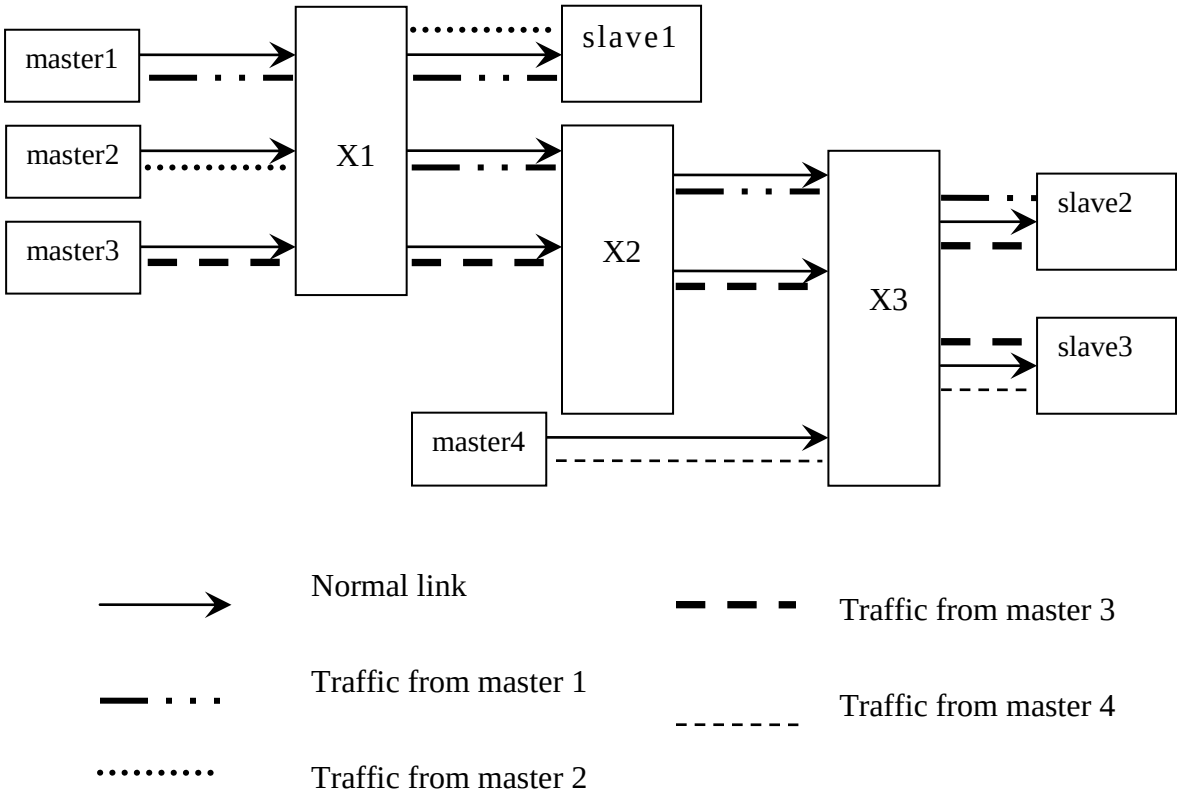

Figure 3.7 Showing traffic flows

Traffic flow per switch must not exceed the peak acceptable frequency provided by the application. To ensure that we need a number of variables who act to save the traffic flow per switch per slave and the traffic flow per cascaded switch. These are used in the actual

constraint that will ensure that the traffic flow per switch does not exceed the maximum possible limit for the switch.

$$trafS_{i,j,k} = (link_{i,j,0,\dots,0,k} + link_{i,0,j,0,\dots,0,k} + \cdots + link_{i,0,0,\dots,j,k}) * tpatt_{i,k}$$

Equation 3-9 Constraint 3 Calculating traffic flow per switch per slave port per master

$$trafX_{i,j,k,l} = (link_{i,j,k,0,\dots,0,l}) * tpatt_{i,l}$$

Equation 3-10 Constraint 3 Calculating traffic flow per switch in case of a cascaded network per master

$$traffic\_S_{i,j} = \sum_{k=0}^{m} trafS_{k,i,}$$

Equation 3-11 Constraint 3 Calculating traffic flow per switch per slave

$$traffic_{X_{i,j}} = \sum_{k=0}^{m} \sum_{l=0}^{s} traf_{X\ k,i,j,l}$$

Equation 3-12 Constraint 3 calculating traffic flow per switch per cascaded switch

$$traffic_i = \frac{\sum_{j=0}^{s} traffic_{S_{i,j}} + \sum_{k=1}^{x} traffic_{X_{i,k}}}{bandwidth * freq_i}$$

Equation 3-13 Constraint 3 calculating traffic flow per switch (max. 1)

Once the above said calculations are completed the model should ensure that total traffic flow per switch must not exceed 1 as 1 represents the complete utilization of the switches data transfer resource.

$$traffic_i \geq 1$$

Equation 3-14 Constraint 3 actual constraint

Constraint 4:

Latency Constraint:

Most of the high communication applications have specific latency requirements that must be considered for accurate application execution. Thus there must exist latency constraint in the model must be satisfied in order to formulate an optimized design. In this model the latency expression calculated is non-linear thus it has not been considered in the model.

In its place another constraint has been used namely Connection Check. Aim of this constraint is to force such paths between any pair of master and slave ports which have some traffic flow requirements.

Connect Constraint:

$$CConnect_{ij} = \sum_{n=1}^{x} \sum_{m=1}^{x} \dots \sum_{k=1}^{x} link_{i,k,l,\dots,j}$$

Equation 3-15 Constraint 4 Check for possible path between a pair of master and slave ports

$$CConnect_{ij} \geq (tpatt_{ij}/(tpatt_{ij} + 1))$$

Equation 3-16 Constraint 4 Forces the cconnect decision variable to be one if traffic flows are required by the application

Constraint 5:

Static Power constraint:

Static power is directly proportional to the number of virtual channels and output flows required per switch.

$$Pow_i = a * VCf_i + b * PS_i + c$$

Equation 3-17 Constraint 5 Power consumption expression

As discussed earlier Cost AREA of the NoC architecture was selected as the objective function for the MILP model. The following expression shows the area calculation for the model. Model aims at decreasing the Cost AREA utilization for the NoC architecture.

$$Area = \sum_{i=1}^{x} (a * VCf_i + b * PS_i + c)$$

Equation 3-18 Area Consumption Expression

For both constraint 5 and cost area OPF's and virtual channels per switch are to be calculated. The following equations are used to calculate the virtual channels per switch

$$VCS_{ij} = \sum_{k=0}^{m} link_{k,i,0,0,\dots,0,j} + \sum_{k=0}^{m} \sum_{l=1}^{x} link_{k,l,i,0,\dots,0,j} + \dots + \sum_{k=0}^{m} \sum_{l=1}^{x} \dots \sum_{z=1}^{x} link_{k,z,y,\dots,l,j}$$

Equation 3-19 Expression for calculating virtual channels per switch per slave

$$VCX_{ij} = \sum_{k=0}^{m} \sum_{l=0}^{s} link_{k,i,j,0,\dots,0,l} + \sum_{k=0}^{m} \sum_{l=0}^{s} \sum_{l=1}^{x} link_{k,i,l,j,\dots,0,l} + \cdots + \sum_{k=0}^{m} \sum_{l=0}^{s} \sum_{n=1}^{x} \dots \sum_{z=1}^{x} link_{k,i,z,y,\dots,j,l}$$

Equation 3-20 Expression for calculating virtual channels per switch per cascaded switch

$$VCf_i = \sum_{j=0}^{s} VCS_{ij} + \sum_{k=0}^{x} VCX_{ik}$$

Equation 3-21 Expression calculating virtual channels per switch

Data Collection:

Unlike the other problems that are dealt by mathematical modeling NoC optimization does not require a hectic data collection process. This is made possible by the fact that the digital designing has only certain number of parameters which can be easily obtained wither by the application developer or the NoC designed oneself. Application developer provides the following requirements:

- Frequency
- Bandwidth
- Traffic flow requirement etc

Data such as area and static power are in control of the NoC designer as the results for the mathematical modeling vary according to the designer's selection of routing scheme, power optimization technique or the performance optimization technique such as pipelining. Here ISE XILINX Webpack was used to gather the required information about area and power consumption.

Chapter 4 : Implementation and Results

The complete system was developed in GUSEK, a freeware and a platform independent tool. A number of expressions of the model are directly proportional to the number of switches allowed for the design. To accommodate such behavior of the model a C# application was formed that helps in the generation of such expression.

The overall application covers two areas, namely:

- MODEL's Code Generator
- VERILOG Code Generator

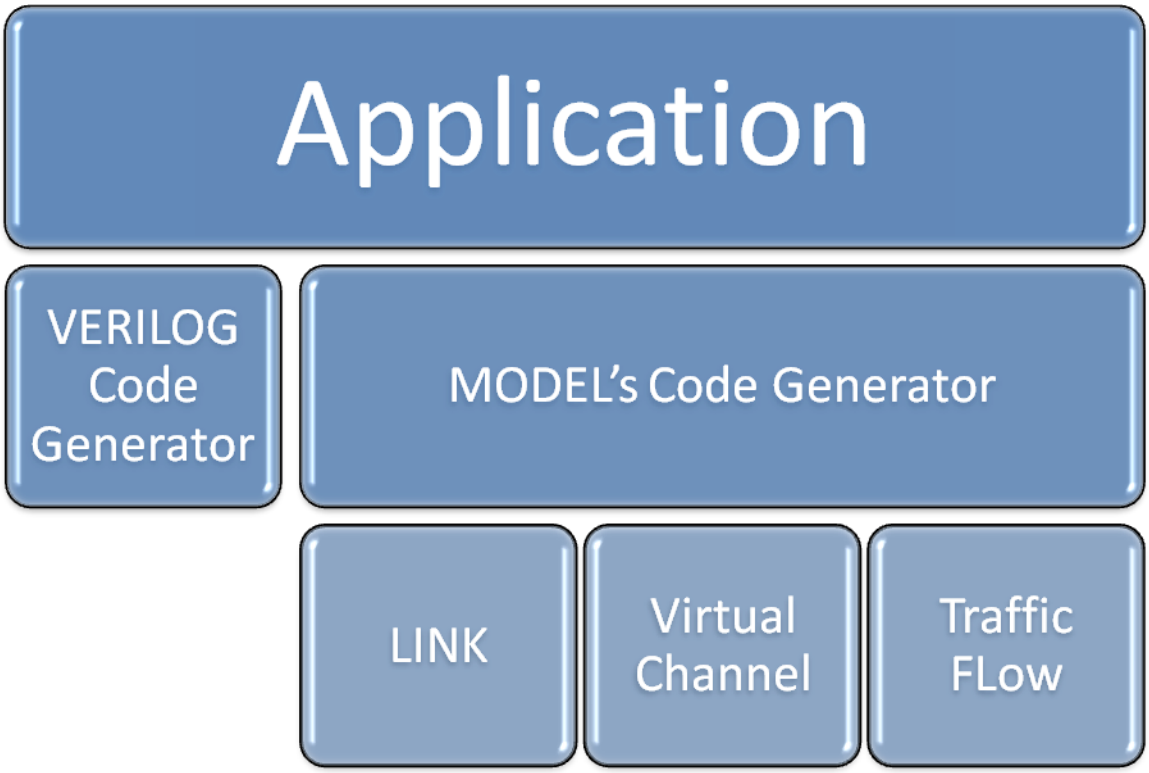

Figure 4.1 Application overview

Code generator for the model caters the following expressions of the model:

- Linear expression for Link
- Linear expression for Traffic
- Linear expression for Virtual Channels

VERILOG code generator part is used to parse the results of the GUSEK model and based upon those results generates VERILOG files of the switch with different input and output port connection.

Switch utilizes wormhole routing scheme of the switch which as discussed in the background section provides the best approach for contention resolution while keeping the use of buffers to a minimum. VERILOG code generator uses this technique as the switches routing scheme.

As discussed earlier application is divided into two parts, as the starting screen the application asks the user to choose the relevant part.

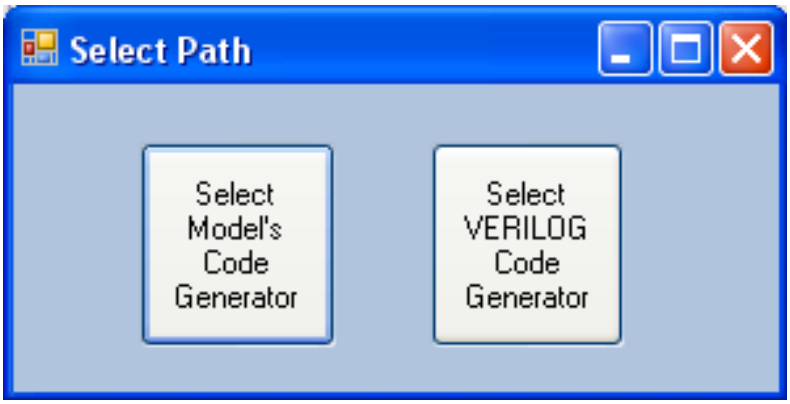

Figure 4.2 Main window of the application

VERILOG's code generator part expects a number of input parameters according to which the code must be generated. These parameters include

- Switch number
- Packet/flit size
- Number of input and output ports
- Traffic flow
- Different bit sizes (Priority, header, message id etc)

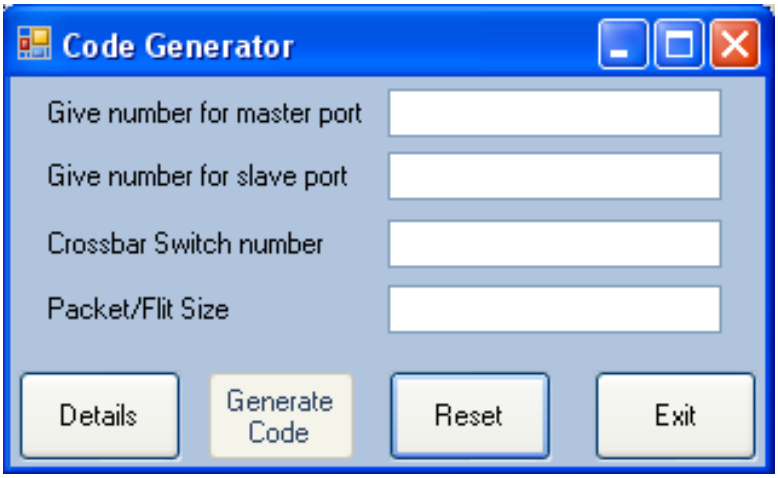

Figure 4.3 Main window for VERILOG Code generator

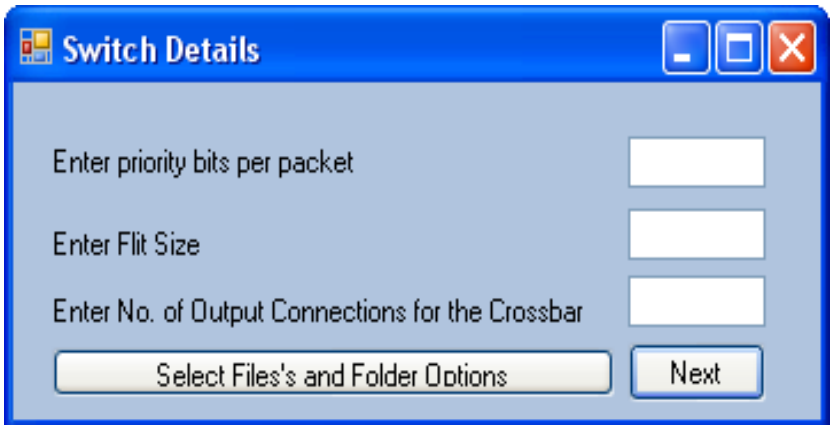

Figure 4.4 Detailed data input for the VERILOG code generator

Apart from the above information there are further details required such as the possible traffic flows per switch. This information is read from different files which are used to calculate number of virtual channels required per output link.

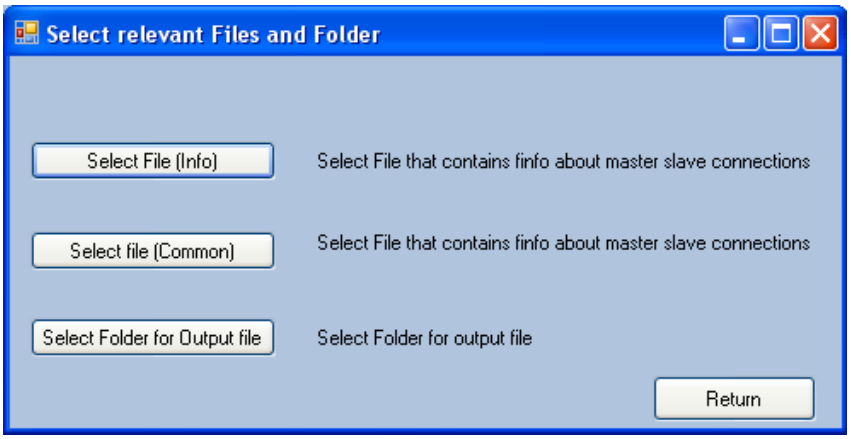

Figure 4.5 Window selecting files and folder

The Model code generator part of the application provides three different options for the user. These include code generation of

- Link expressions
- Virtual Channel expressions
- Traffic flow expression

The above mentioned equations are dependent upon the number of switches provided as the backbone of the NoC design.

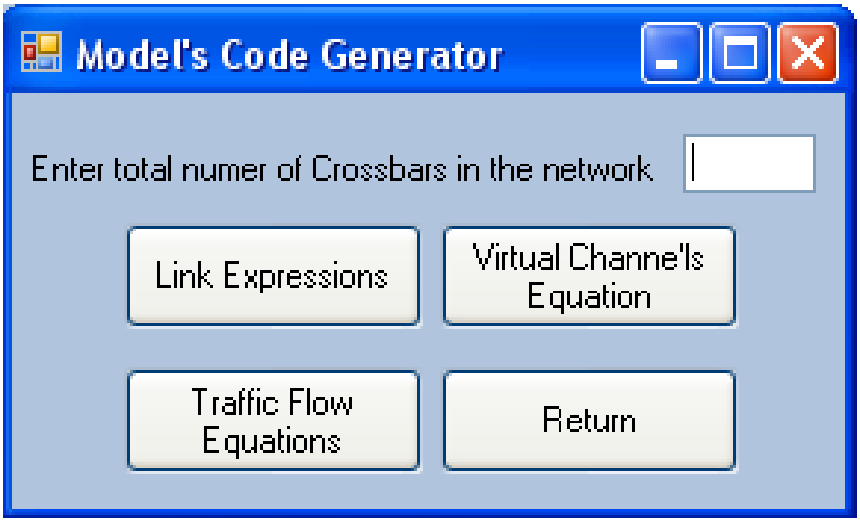

Figure 4.6 Main window for model's code generator

Experimentation and Results:

For experimental case study, we consider a multi-media SoC, *Triple Video Object Plane Decoder*, which has 38 cores (D 38 tvopd) [34]. The application is highly heterogeneous in nature, having three independent decoders working in parallel to improve performance. Each decoder has 12 cores organized in a pipeline fashion. There are two extra memories that are shared between the pipelines that serve as input and output buffers. We consider the design implemented on to 3 layers in 3D. For the topology generation we target the backplane with 12 master and 4 slaves to a total of 21 flows. For our topology generation purposes we set the data bit-width to 8 bits, and select the lowest possible frequency which satisfies the bandwidth requirements of the node with heaviest traffic flows for each SoC and generate solutions. For the MILP model we

use GUSEK and from it generate an MPS file which is then solved using the Gurobi solver. We run the software in a virtual machine inside a core-i5 system and the solver itself ran on two threads.

Solution time (Sec)	10.16
Total Solution cost (LUT+Reg)	2184
Number of slice LUTs	1360
Number of slice registers	824
Bit width	8
Freq (Mhz)	202.5

Table 4-1 Result parameters for application

Router Details	No of flows	Assigned Traffic to Capacity%
1	4	88.0247
2	4	84.9383
3	4	97.9012
4	6	99.321
5	6	98.642
6	5	99.6296
7	11	99.8765
8	8	99.0741

Table 4-2 Assigned router details

Flow No	Traffic Requirement (mb/s)	Assigned router	Flow No	Traffic Requirement (mb/s)	Assigned router
1	70	8	25	16	7
2	362	8	26	540	4

3	362	8	27	126	6
4	362	8	28	300	4
5	49	8	29	313	4
6	27	8	30	313	4
7	357	8	31	94	5
8	353	7	32	500	3
9	16	8	33	70	5
10	540	7	34	362	3
11	126	7	35	362	3
12	300	7	36	362	3
13	313	6	37	49	4
14	313	6	38	27	7
15	94	7	39	357	2
16	500	6	40	353	2
17	70	7	41	16	7
18	362	6	42	540	2
19	362	5	43	126	2
20	362	5	44	300	1
21	49	7	45	313	1
22	27	7	46	313	1
23	357	5	47	94	4
24	353	5	48	500	1

Table 4-3 Per flow traffic and router assignment for application

The results are displayed in tables 1-3. As can be seen in table, the overall router cost is very low. Although we do not generate the number of slices to keep the result platform independent, the resulting LUTS and Slice registers would be synthesized to a very area efficient solution for FPGA platforms in terms of area resources available.

Another interesting result is that even for the application with 48 flows the results are computed in 10.16 seconds. This shows that for practical SoC applications we can use a MILP formulation to generate application specific network on chip topologies without

resorting to heuristics. This is in counter to previous examples in literature which state the large times to generate results.

Chapter 5 : Conclusion and Future Work

The design methodology of NoC architectures is crucial for achieving high performance and energy efficient systems. In this thesis Mathematical formulation for synthesize of application specific NoC architectures has been presented such that the performance constraints are satisfied and the cost area consumption is minimized. The most interesting result that is inferred from the experimentation is that even with the application for nearly 50 flows the results are computed in merely 10.6 seconds. This shows that for practical SoC applications we can use a MILP formulation to generate application specific network on chip topologies without resorting to heuristics. This is in counter to previous examples in literature which state the large times to generate results.

The mathematical model provides the following parameters

- Topology
- Master/Slave PE connections
- Traffic flow
- Virtual channels

While adhering to the constraints set by the user and the application requirements provided by the user.

In this model the objective was

- Cost Area minimization

The use of Networks on Chips (NoCs) for communication in 3D chips has posed new opportunities and challenges for designers. One of the most important problems is to design the most power performance efficient NoC topology that satisfies the application characteristics and 3D technology requirements.

In future new power, performance, traffic models are required to be developed that can provide better optimization. To achieve this one must work to develop better application and traffic models along with the introduction of new optimization techniques. Improvement in this thesis can be made on the bases of introduction of power consumption constraints. This can be done by using clock gating or other such techniques. Clock gating runs contrary to our main assumption of area minimization as it requires extra logic for bypassing the units that are powered off thus resulting in increased area utilization.

Another aspect that hasn't been considered in the thesis is performance of the NoC design in case of priority based masters and slave ports. QoS remains an interesting area where exists scope for future research.

References

1. Thomas M. Cook, Robert A. Russell, “Introduction to Management Sciences”

2. Sajid Gul Khawaja, Mian Hamza Mushtaq and Dr. Shoab Khan, “Prioritized Direction based Switch for Bufferless Network on Chip Architecture” IJENS Vol. 11 Issue04 pg. 19-26.

3. Minje Jun, Sungjoo Yoo and Eui-Young Chung “Mixed Linear Integer Programming-based Optimal Topology Synthesis of Cascaded Switch Switches”

4. Wang Zhang, Ligang Hou, Jinhui Wang, Shuqin Geng, Wuchen Wu “Comparison Research between XY and Odd-Even Routing Algorithm of a 2-Dimension 3X3 Mesh Topology Network-on-Chip” VLSI & Syst. Lab., Beijing Univ. of Technol., Beijing, China.

5. Dietmar Tutsch, Bergische Universität Wuppertal, Wuppertal, “Comparison of network-on-chip topologies for multicore systems considering multicast and local traffic” Miroslaw Malek Humboldt-Universität zu Berlin, Berlin, Germany.

6. http://www.xuru.org/rt/MLR.asp

7. http://en.wikipedia.org

8. Ben A. Abderazek, Mushfiquzzaman Akanda, Tsutomu Yoshinaga, and Masahiro Sowa, “Mathematical Model for Multiobjective Synthesis of NoC Architectures”, Graduate School of Information Systems The University of Electro-communications

9. Suleyman Tosun, Ozcan Ozturk, Meltem Ozen “An ILP Formulation for Application Mapping onto Network-on-Chips”

10. Dara Rahmati*, Srinivasan Murali¥§, Luca Benini‡, Federico Angiolini¥§,“A Method for Calculating Hard QoS Guarantees for Networks-on-Chip”

11. Anita Tino *, Gul N. Khan “Designing power and performance optimal application-specific Network-on-Chip architectures”

12. Visit http://www.itrs.net for ITRS 2007 edition.

13. M. Jun, K. Bang, H. J. Lee, N. Chang, and E. Y. Chung, "Slack based Bus Arbitration Scheme for Soft Real-time Constrained Embedded Systems", ASPDAC 2007, pp. 159-164K. Lahiri, A. Raghunathan, and G. Lakshminarayana, "LOTTERYBUS: A New High Performance Communication Architecture for System-on-Chip Designs", DAC 2001, pp. 15-20

14. M. Dall'Osso, G. Biccari, L. Giovannini, D. Bertozzi, and L. Benini, “xpipes: a Latency Insensitive Parameterized Network- on-chip Architecture For Multi-Processor SoCs”, International Conference on Computer Design 2003, pp.536-539

15. K. Srinivasan, K. S. Chatha, “A low complexity heuristic for design of custom network-on-chip architectures”, DATE 2006, pp.130-135

16. K. Srinivasan, K. S. Chatha, and G. Konjevod, "Linear Programming Based Technique for Synthesis of Network-on- Chip Architectures", IEEE TVLSI, April 2006, vol. 14, pp. 407-420

17. M. Millberg, E. Nilsson, R. Thid, S. Kumar, and A. Jantsch, “The Nostrum backbone - a communication protocol stack for networks on chip”, Proceedings of the VLSI Design Conference, Jan. 2004, pp. 693-696

18. J. Hu, U. Ogras, R. Marculescu, System-level buffer allocation for application-specific networks-on-chip router design, IEEE Trans. CADICS 25 (2006) 2919–2933.

19. K. Goosens, J. Dielissen, A. Radulescu, Aethereal network on chip: concepts, architectures, and implementations, IEEE Des. Test 22 (5) (2005) 414–421.

20. M. Rezazad, H. Sarbaziazad, The effect of virtual channel organization on the performance of interconnection networks, IEEE IPDPS 15 (2005) 264.

21. R. Mullins, A. West, S. Moore, Low-latency virtual channel routers for on-chip networks, in: IEEE Symposium on Computer Architecture, 2004, p. 188.

22. J.W. Van den Brandt, C. Ciordas, K. Goosens, T. Basten, Congestion-controlled best-effort communication for networks-on-chip, IEEE DATE (2007) 948–953

23. G. Leary, K. Chatha, Automated technique for design of NoC with minimal communication latency, in: CODES + ISSS 2009, October 2009, pp. 471–480.

24. K.S. Chatha, K. Srinivasan, G. Konjevod, Automated techniques for synthesis of application-specific network-on-chip architectures, IEEE Trans. CAD Integr. Circ. Syst. 27 (8) (2008) 1425–1438

25. J. Hu, R. Marculescu, Energy-aware communication and task scheduling for NoCs under real-time constraints, in: Proc. DATE, vol. 1, 2004, pp. 234–239

26. U. Ogras, R. Marculescu, Application-specific network-on-chip architecture customization via long-range link insertion, in: Proc. ICCAD, 2005, pp. 246– 253.

27. U. Ogras, R. Marculescu, Energy-and performance-driven NoC communication architecture synthesis using a decomposition approach, in: Proc. DATE, vol. 1, 2005, pp. 352–357.

28. G. Guindani, C. Reinbrecht, T. Da Rosa, F. Moraes, Increasing NoC power estimation accuracy through a rate-based model, in: Proc. 3rd ACM/IEEE NoC Symposium, 2009, p. 89.

29. T. Ahonen, D. Siquenza-Tortosa, H. Bin, J. Nurmi, Topology optimization for application-specific networks-on-chip, in: Proc. SLIP, 2004, pp. 53–60.

30. G. Ascia, V. Catania, M. Palesi, Multi-objective mapping for mesh-based networks-on-chip architectures, in: Proc. IEEE /ACM/IFIP, 2004, pp. 182–187.

31. G. Leary, K. Chatha, Srinivasan, K. Mehta, Design of network-on-chip architectures with a genetic algorithm-based technique, IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 17 (5) (2009) 674–687

32. F. Glover, M. Laguna, Tabu Search, Kluwer Publishers, 1997.

33. S. Murali, P. Meloni, F. Angiolini, D. Atienza, S. Carta, L. Benini, G. De Micheli, L. Raffo, Designing application specific network on chips with floorplan information, in: Proc. ICCAD, 2006, pp. 355–362.

34. Srinivasan Murali, Ciprian Seiculescu, Luca Benini, Giovanni De Micheli, "Synthesis of Networks on Chips for 3D Systems on Chips"

35. Visit http://techresearch.intel.com/articles/Tera-Scale/1421.html for Tera-Scale Computing Research Program