A NEW APPROACH TO STABLE MATCHING AND NETWORKS–ON–CHIP PROBLEM

A THESIS

Submitted in partial fulfillment of the
Requirements for the award of the degree of

DOCTOR OF PHILOSOPHY
IN
COMPUTER SCIENCE & ENGINEERING
BY
NITIN

Department of Computer Science & Engineering and Information Technology
Jaypee University of Information Technology, Waknaghat, Solan–173215,
Himachal Pradesh, INDIA

JULY 2008

CANDIDATE'S DECLARATION

I hereby certify that the work, which is being presented here in the thesis entitled, A NEW APPROACH TO STABLE MATCHING AND NETWORKS–ON–CHIP PROBLEM, submitted in partial fulfillment of the requirements for the award of the degree of Doctor of Philosophy in Computer Science & Engineering and submitted in Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology (JUIT), Waknaghat is an authentic record of my own work carried out (July 2005–July 2008) under the supervision and guidance of Professor Durg Singh Chauhan, Vice Chancellor, JUIT, Waknaghat.

Hereby I declare that this doctoral thesis, my original investigation and achievement has not been submitted by me for the award of any other degree on this work in any other Institution/University.

______

DATE: NITIN

I certify that I have read this thesis and that in my opinion; it is fully adequate in scope and quality as a thesis for the award of degree of the Doctor of Philosophy in Computer Science & Engineering.

____

DATE: PROFESSOR DURG SINGH CHAUHAN
(ADVISOR)

The Doctor of Philosophy (Thesis) Viva–Voice Examination of Mr. NITIN (Enrollment No. 051758), Doctor of Philosophy in Computer Science & Engineering has been held on ________________.

__________ ____________________________ ______________________

ADVISOR HEAD OF THE DEPARTMENT EXTERNAL EXAMINER

____

____

ACKNOWLEDGMENTS

I sincerely express my gratitude to my advisor, Professor Durg Singh Chauhan, for consistent guidance in this research work. I am fortunate to have blessings of almighty, brought closer to my advisor, is always interested in exploring new research directions and ideas with deep vision, encouragement and dimension. He gave freedom in performing research activities and ensured that my various contents are neatly tied together. He has personal touch and support to all his students. One can easily feel friend and philosopher in his disposition.

This research has benefited from the ideas and suggestions of many well–wishers. I am indebted to Professor Pawan Kumar Bansal, who initiated into vast aspects of Multi–stage Interconnection Networks, and Professor Ashok Subramanian, with whom I interacted actively for long duration on the subject matter.

Very special thanks to Management, Jaypee Group specifically to Dr. Yajulu Medury, founder Vice Chancellor and Brigadier (Retired) Balbir Singh, Registrar, Jaypee University of Information Technology, Waknaghat, for providing excellent research environment, learning ambiance at the campus and facilities for research activities on Sun Solaris and High–end Computing Laboratories.

I am eternally grateful to my respected Mother and Father and affectionate younger Brother beyond words and for the most devoted Wife and Sweet Daughter for their love and support they have given me all over these years. Without their sustained moral support, it would have been impossible to succeed in writing this wonderful thesis.

I strongly commit and dedicate this achievement to his holiness, Shri 1008 Shri Sadgurudev Swami Swarupanandji Maharaj, who is always transcendental entity to me and my departed Maternal Grandfather and Grandmother, though they are no more in worldly sense but their living guidance never left my spirit to work more and more and reach ultimate heights of the knowledge.

CONTENTS

PAGE NO.

PARTIAL

LIST OF FIGURES

PARTIAL

LIST OF TABLES

PARTIAL

CHAPTER 1

INTRODUCTION

1.1 *Problem Statement*

This thesis explores and solves the unique problem of the relationship between Stable Matching [1] and Multi–stage Interconnection Networks (MIN) [1–19]. Specifically, the Stable Marriage Problem [20–22] is used as an example of stable matching to solve the stability problem of the MINs. The situations in which the MINs become unstable have reported and proved stable. Two different classes of dynamic fault–tolerant MINs such as irregular Hybrid ZETA Network (HZTN) [1–8], Quad–tree Network (QTN) [1,7] and regular Augmented Shuffle–Exchange Network (ASEN) [1,2,4–6], Augmented Baseline Network (ABN) [1,2,3,7] are used as running examples in the text and prove to be stable using stable matching approach.

The application of the MINs in Systems–on–Chip (SoC) [9,10,23–26] and Networks–on–Chip (NoC) [9,10,24–26] are consistently drawing attention since year 2002. The parallel communication among Application–Specific NoC [9,10] is a major problem to handle by the researchers. Therefore, to overcome this problem the thesis proposes a new method that helps to set up intra (global) communication between Application–Specific (heterogeneous or homogenous) NoCs in Networks–in–Package (NiP) [9,10,25]. The global communication is made possible through the single fault–tolerant irregular and regular MIN, fundamentally behaving as an Interconnects–on–Chip (IoC) [9,10,23].

The literature survey reveals that Star, Common Bus and Ring topologies have been used as a medium to set up intra NoCs communication. However, these communication systems have many tradeoffs, and the disadvantages are as mentioned below:

1. *It has high latency.*
2. *The system scalability is low.*
3. *Its performance is poor.*
4. *It does not provide fault–tolerance.*

5. *As system is not reliable, therefore, On–chip repairability is not possible.*
6. *It has high contention over sharing of channel.*
7. *Presence of livelock.*
8. *Presence of deadlock.*
9. *The probability of acceptance of data packets is low.*

The system proposed in the thesis has the following advantages over existing one:

1. *It has Low latency.*
2. *The system is scalable.*
3. *It possesses high performance.*
4. *It provides Fault–tolerance.*
5. *The system is reliable and hence, online repairability is possible.*
6. *It provides low contention over sharing of channel.*
7. *The system is livelock free.*
8. *The system can tolerate deadlock up to certain limits.*
9. *The probability of acceptance of data packets is high.*

It is common conclusion in the published research papers that with the increasing complexity of chip design due to enormous transistor densities and Deep Sub-Micron (DSM) problems, the pressure on the On–chip communication backbone has become too big to handle by the conventional structures. Conventional structures are typically a heap of all shapes and sharing a single channel. This severely limits the scalability of the bus because with increasing number of masters the bus will inevitably slow down the traffic with increase DSM technology. The density of transistors and operating frequency of Processing Element (PE) is also increasing but the global wire delay remains constant. Therefore, in order to synchronize the pipeline, one should know the global wire delay for efficient and effective communication. Furthermore, the modern Computer Aided Design (CAD) tools are inefficient to predict the possible Quality of Service (QoS). The solution for this kind of problem is Stochastic Communication [28–32] that addresses and solves the following problems:

1. *Latency among PEs.*
2. *Data failure in the network.*
3. *Fault tolerance of the system.*
4. *Energy or power dissipation.*

This thesis solves the latency problem of the Application–Specific NoCs. However, during the study of parallel communication between the IPs and the NoCs, the loss of data packets was inevitable. Therefore, to track this loss of data packets the technique for stochastic communication is recommended.

Since year 2003 and onwards the researchers have done enough work in developing, stochastic communication for inter NoC communication. However, the major drawbacks of these systems are as below:

1. *The communication is only for the broadcast scheme.*
2. *The prediction of the latency is calculated within the groups.*
3. *Probabilistic data tracking is available within the group.*
4. *Designing prospects are worst.*
5. *Stochastic communication technique is available only for inter NoC communication.*
6. *Presence of Livelock.*
7. *Presence of Deadlock.*
8. *Network Unstability.*

The stochastic communication for intra (specifically communication among different NoCs in NiP) NoCs communication was ignored and remains out of limelight. Therefore, this thesis; propose an analytical technique for stochastic communication, which is suitable for homogeneous as well as heterogeneous NoCs. This technique not only separates the computation and communication in NiP module but also predicts the communication performance. The communication structure as a whole in NiP is divided into two zones. The one is On–chip inter NoC communication and the other is On–package intra NoC communication. The stochastic communication technique adopted in the work has following major advantages over existing systems:

1. *The communication is not only for the broadcast scheme but also for parallel and dedicated communication for real time application.*
2. *The prediction of the latency is calculated within the groups as well as in individual.*
3. *Probabilistic data tracking is available within the group as well in individual.*
4. *The prediction based QoS is same for both (theirs and ours), but the designing prospects of our Stochastic Communication method is better e.g. Traffic Controller and Clock Synchronization for Globally Asynchronous, Locally Synchronous (GALS).*

5. *Stochastic communication technique is available for Inter NoC as well as Intra NoCs communication. The architectural design of the NoC may be heterogeneous or homogenous.*
6. *The system is livelock free.*
7. *The system is deadlock free.*

1.2 *Contributions*

This thesis includes several key contributions to the study of stable matching problem of the MINs and application of MINs to NoCs and NiP to solve parallel communication and latency problems.

Firstly, this thesis presents the stable matching problem (similar to pair matching) and its application to MINs stability solution.

- MULTI–STAGE INTERCONNECTION NETWORKS STABILITY PROBLEM [1]. This study provides efficient and fast algorithms to solve stability problem of the irregular (HZTN and QTN) and regular (ASEN and ABN) class of fault–tolerant MINs and are compared on the similar ground. For proving the MINs stable two algorithms are proposed:–the first algorithm generates the MINs Preference Lists in $O(n^2)$ time and second algorithm produces a set of optimal (stable) pairs of the SEs (derived from the MINs preference lists) in $O(n)$ time. Moreover, this thesis also presents and discusses the issues of ties between optimal pairs.

From the stable matching problem of the MINs, the focus extends to its application as IoC (specifically a small prototype of MIN) to set up parallel communication among dynamic NoCs.

- PARALLEL COMMUNICATION AMONG APPLICATION–SPECIFIC NETWORKS–ON–CHIP [9]. The thesis proposes parallel algorithms that allow NoCs (here the architecture design of the NoC may be either heterogeneous or homogenous) in NiP to communicate together using fault–tolerant IoCs. The said architecture uses twin $O(n^2)$ time fault–tolerant parallel algorithms. These algorithms allow different NoCs to communicate efficiently in parallel with minimum number of packet losses. Out of the two IoCs, one is 2 x 2 fault–tolerant irregular Penta multi–stage interconnection networks (PNN), with 3 stages and 5 SEs (all stages include chaining or express links, except the middle one) and other is 2 x 2 fault–tolerant regular Hexa multi–stage interconnection networks (HXN), with 3 stages and 6 SEs (all stages include chaining or express links, except the middle one). The MTTR and Cost of the PNN and HXN have been simulated and compared.

Finally, the thesis deals with stochastic communication technique develop for inter NoC and intra NoCs (in NiP) communication.

- STOCHASTIC COMMUNICATION NoC [10]. New technique has been proposed for stochastic communication between different IPs. These IPs are connected with different routers or switches and are treated as different compartments on a single chip. The spread of information among IPs is represented using a Closed Donor Controlled Based Compartmental Model, which is convertible into a Stochastic Model that is more realistic and enables to compute the transition probabilities between the IPs as well as latency. A case study on Compartmental–based Probabilistic Data Broadcasting among NoCs using HXN IoC is presented.

CHAPTER 2

PRELIMINARIES AND BACKGROUND

Parallel Processing refers to the concept of speeding–up the execution of a program by dividing the program into multiple fragments that can execute simultaneously, each on its own processor. A program being executed across *n* processors might execute *n* times faster than it would on a single processor.

It is known that one way for processors to communicate data is to use a shared memory and shared variables. However, this is unrealistic for large number of processors. A more realistic assumption is that each processor has its own private memory and data communication takes place using message passing via an Interconnection Networks (IN).

INs originated from the design of high performance parallel computers. When need felt for more bandwidth that has put them to use in network switches and to connect peripherals to a computer. INs is a major factor that differentiates modern multiprocessor architectures and is categorized according to a number of criteria such as topology, routing strategy, and switching technique. IN is building up of switching elements; topology is the pattern in which the individual switches are connected to other elements, like processors, memories and other switches.

2.1 *Interconnection Networks*

"Interconnection Networks should be designed to transfer the maximum amount of information within the least amount of time (and cost, power constraints) so as not to bottleneck the system".

INs has a long development history [11–18]. The Circuit switched networks has been used in telephony. In 1950s, the interconnection of computers and cellular automata as few prototypes was developed until 1960 it awaited full use. Solomon in 1962 developed multicomputer network. Staran with its flip network, C.mmp with a crossbar and Illiac–IV with a wider 2–D network received attention in early 1970s. This period also saw several indirect network used in vector and array processors to connect multiple processors to multiple memory banks. This problem was developed in several variants of multi–stage interconnection networks (MIN). The BBN Butterfly in 1982 was one of the first multiprocessors to use as an indirect network. The binary e–cube or hypercube network was proposed in 1978 and implemented in the Caltech Cosmic Cube in 1981. In the early 1980s, the academic focus was on mathematical properties of these networks and became increasingly separated from the practical problems of interconnecting real systems.

The last decade was the golden period for INs research driven by the demanding communication problems of multicomputer enabled by the ability to construct single–chip Very Large Scale Integration (VLSI) routers. The researchers have made a series of breakthroughs that have revolutionized the digital communication systems. The Torus Routing Chip, in 1985, was one unique achievement. The first of a series of single–chip routing components introduced wormhole routing and virtual channels used for deadlock avoidance. The whole family of chips laid the framework for analysis of routing, flow–control, deadlock and livelock issues in modern direct networks. A flurry of research followed with new theories of deadlock and livelock, new adaptive routing algorithms, and new methods for performance analysis. The research progressed in collective communication and network architectures on a regular basis. By the early 1990s, low–dimensional direct networks had largely replaced the indirect networks of the 1970s and the hypercubes of the 1980s could be found in Cray, Intel, Mercury, and some other machines. The applicability of INs in digital communication systems with the appearance of Myrinet was adopted in 1995. The point–to–point multiple networks technology replaced the use of buses, which were running into a limited performance due to electrical limits and were used in the barrier network in the Cray T3E, as an economical alternative to dedicated wiring. However, the interconnection network technology had certain barriers on design and various researchers and engineers have performed analysis of these networks [15,17,18].

2.1.1 *Multi–stage Interconnection Networks*

As the acceptance and subsequent use of multiprocessor systems increased, the reliability, availability, performability, and performance characteristics of the networks that interconnect processors to processors, processors to memories, and memories to memories are receiving increased attention. A brief survey of INs and a survey of the fault–tolerant attributes of MINs are reported in [11,13]. A MIN in particular is an IN consisting of a cascade of switching stages, each containing switching elements (SE). MINs are widely used for broadband switching technology and for multiprocessor systems. Besides this, MINs offer an enthusiastic way of implementing switches used in data communication networks. With the performance requirement of the switches exceeding several terabits/sec and teraflops/sec, it becomes imperative to make them dynamic and fault–tolerant [3,4].

Based on the paths availability the MINs traditionally have been divided into three classes:

1. *Blocking. In this, there is a unique path between every input/output pair, thus minimizing the number of switches and stages.*
2. *Nonblocking. Any input port can be connected to any free output port without affecting the existing connections.*
3. *Rearrangeable. Any input port can be connected to any free output port. However, the existing connections may require rearrangement of paths.*

The typical modern day application of the MINs includes fault–tolerant packet switches, designing multicast, broadcast router fabrics while Systems–on–Chip (SoC) and Networks–on–Chip (NoC) are hottest research topics in current trends [3]. Normally the following aspects are always considered while deigning the fault–tolerant MINs: the topology chosen, the routing algorithm used, and the flow control mechanism adhered. The topology helps in selecting the characteristics of the present chip technology in order to get the higher bandwidth, throughput, processing power, processor utilization, and probability of acceptance from the MIN based applications, at an optimum hardware cost. Various researchers have already done sufficient work on regular MINs and irregular topologies were out of limelight. Therefore, it has been decided to work on irregular fault–tolerant MINs dominantly and comparing the same with regular networks based on stable matching.

2.1.2 *Network Topology*

2.1.2.1 *Centralized Switched (Indirect) Networks*

In Crossbar network, shown in Figure (2.1), the crosspoint switch complexity increases quadratically with the number of crossbar input/output ports $= N$, i.e., grows as $O(N^2)$ and has the property of being non–blocking.

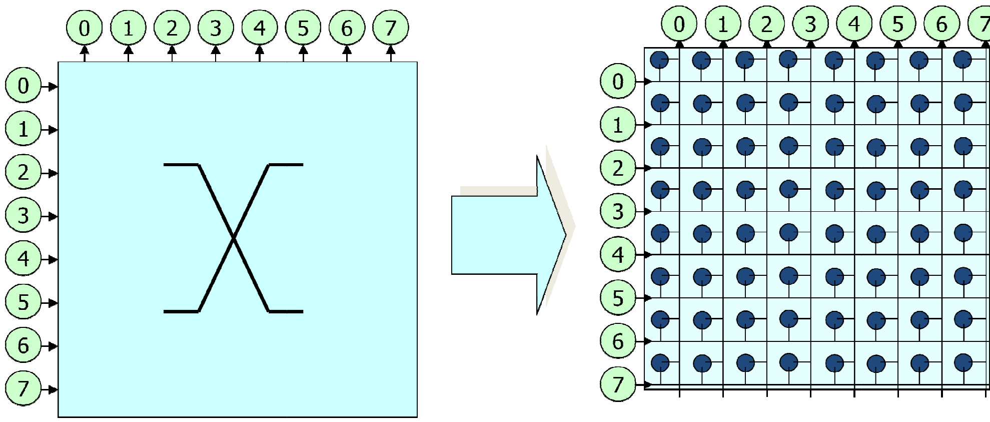

Figure 2.1. Crossbar Network.

In MINs, the crossbar is split into several stages consisting of smaller crossbars and the complexity grows as $O(N \log N)$, where N is # of end nodes. The Inter–stage connections represented by a set of permutation functions. Figure (2.2) is an omega network, with 8 sources and 8 destinations communicating with each other using a MIN. The routing in the said network is done on the basis of perfect shuffle–exchange. Further, using Figure (2.3) a 16 port, 4 stage Omega network have been shown.

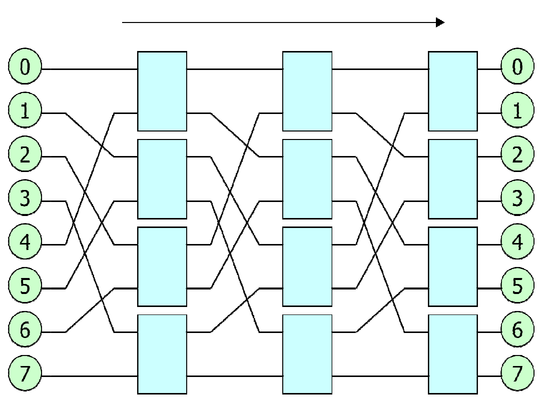

Figure 2.2. An Omega Network.

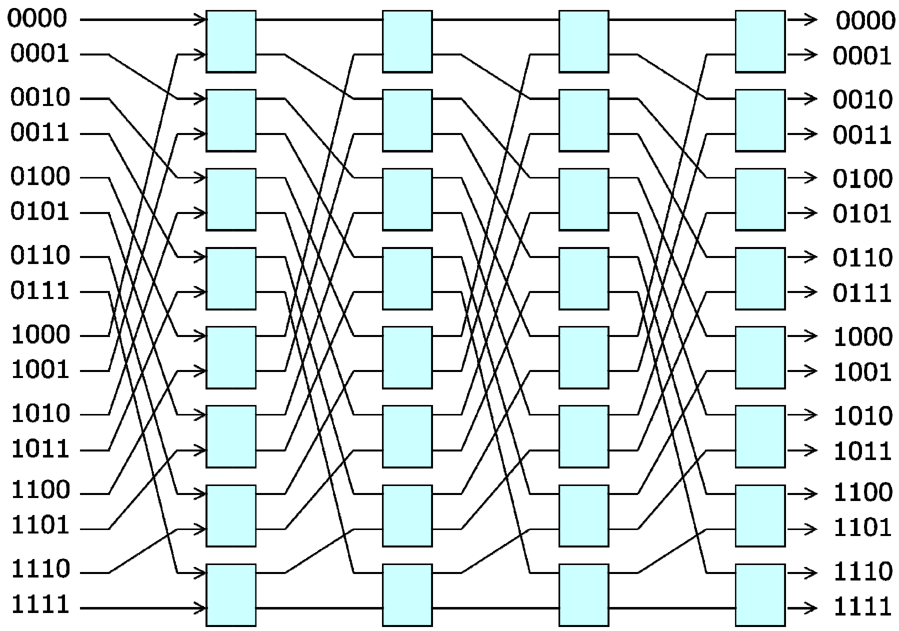

Figure 2.3. 16 port, 4 stage Omega Network.

MINs interconnect N input/output ports using k x k switches, $\log_k N$ switch stages, each with N / k switches and $N/k\left(\log_k N\right)$ total number of switches. As the MINs size increases the cost also increases and the reduction in MINs, switch cost comes at the price of performance. The Network has the property of being blocking and the contention is more likely to occur on network links moreover the paths from different sources to different destinations share one or more links. Figure 2.4(a–b) shows the circuits with non–blocking and blocking topology.

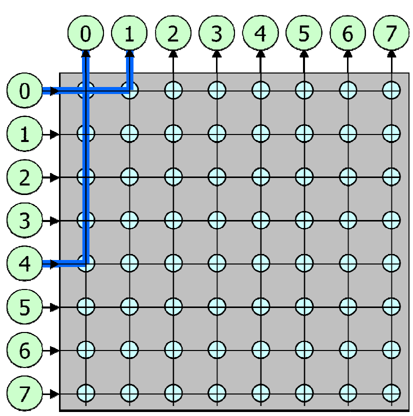

Figure 2.4 (a). Non–blocking topology.

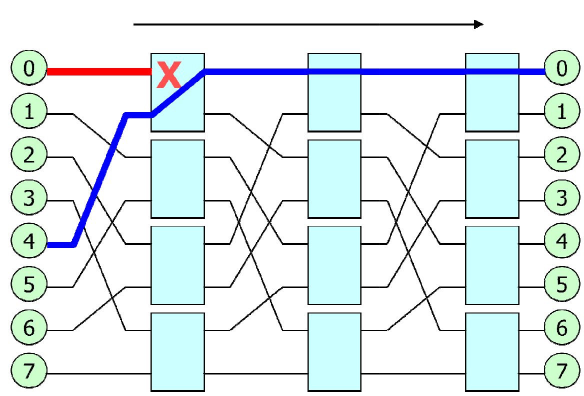

Figure 2.4(b). Blocking topology.

PARTIAL

CHAPTER 3

MULTI–STAGE INTERCONNECTION NETWORKS STABILITY PROBLEM*

3.1 *Introduction and Motivation*

In a Stable Matching problem, the task is to match a number of persons in pairs, subject to certain preference information. Briefly, each person regards some of the others as acceptable mates and ranks them in order of preference. A matching is unstable if two persons did not match and considered together. The task is to find a stable matching, i.e., one that is acceptable to pairs.

The subject of this chapter is the Stable Marriage problem in particular, and stable matching problems in general. Gale and Shapley [20] first studied this problem. Gale and Shapley have shown that a stable matching always exists if the problem is a marriage problem, i.e., if the participants can be divided into two sexes, the men and the women, in such a way that the acceptable mates of each person are all of the opposite sex; in fact, both proposed a linear–time algorithm to find such a matching. Irving, in [33], gave a linear–time algorithm for general problem. An introductory treatment of stable matching appears in reference [34]; a comprehensive treatment is reported in reference [21].

This chapter explores the relationship between stable matching and Multi–stage Interconnection Networks (MIN) Stability Problem. Mayr and Ashok proved that the Network Stability problem is NP–Complete in general [35–37], but when the network is a MIN, then the stability problem becomes equivalent to stable matching. Specifically, stable marriage problem has been used as an example of stable matching to solve the MINs stability problem. It is concluded that the situation in which the MINs become unstable and proved

*Author wishes to express that the stable matching technique used as application to the MINs is a unique contribution, as this has never been reported or used for the MINs before. Therefore, the author cannot compare his work with others. Since the work has already been published and appreciated at International Platform i.e. the Journal [1], author made this achievement as the primary contribution and presented in the thesis in the beginning. Later some properties are compared on the complete performance with different aspects. The referees have appreciated the originality of the research work.

stable using the stable matching approach. The algorithms have been proposed to solve this problem and provide better solutions when the instances of stable matching have ties or when issues of deceit are involved. It explores the structure of all instances of stable matching similar to the research work presented by Gusfield [46], Irving [47] and Feder [48].

MINs are widely used for broadband switching technology and for multiprocessor systems. Besides this, MINs offers an enthusiastic way of implementing switches used in data communication networks. With the performance requirement of the switches exceeding several terabits/sec and teraflops/sec, it becomes imperative to make them dynamic and fault–tolerant [11,13,17,18]. This chapter used Irregular MIN known as Hybrid ZETA Network (HZTN) [1–8], Quad–tree Network (QTN) [41] and Regular MIN known as Augmented Shuffle–exchange Network (ASEN) [42,43], Augmented Baseline Network (ABN) [44] as running examples.

The rest of the work is organized as follows: Section 3.2 contains introduction of INs, MINs, and stable matching problems. Section 3.3 provides the algorithms, preference lists, and optimal pairs to solve the stability problems of HZTN, QTN, ASEN, and ABN MIN. Section 3.4 presents the results.

3.2 *Preliminaries and Background*

3.2.1 *Stable Matching*

An instance of stable matching (also called Stable Roommates) consists of a set of persons, each of whom regards some of the others as acceptable mates, and ranks them in decreasing order of preference. For our purposes it may assume that acceptability is, mutual–x is acceptable to y if and only if y is acceptable to x. A matching is a pairing of some or all of the persons. A matching may be unstable in three ways–two unmatched persons may find each other acceptable; a matched person may prefer an unmatched person to his current mate; or two matched persons may prefer each other to their current mates. A matching that is not unstable is said to be stable.

An instance of stable matching is an instance of stable marriage if the persons divided into two sets, the men and the women, so that the acceptable mates of each person are all of the opposite sex. An instance of stable matching is an instance of Complete Stable Matching if there is an even number of persons and each person is acceptable to everyone else. Similarly, an instance of stable marriage is an instance of complete stable marriage if there are an equal number of men and women and each person is acceptable to every person of the opposite sex.

The size of an instance of stable matching is the sum, over all persons x, of the number of persons acceptable to x. The most common tasks associated with an instance of stable matching are to determine whether a stable matching exists and to construct one if possible. Other tasks might include counting and enumerating all stable matchings of a given instance.

The following scenario for showing the fundamental of stable marriage problem is defined as follows: here are given N men and N women and each man ranks each woman from 1 to N and each woman ranks each man from 1 to N. A set of marriages is always being a one–to–one correspondence between men to women. However, If there does not exist an unmarried pair both preferring each other to their current partners, then it can be said that this set of marriages is declared stable otherwise it is unstable. For example, Mj (Wj) to be Man j (Woman j) [45].

Example 3.1: Consider the following situations:

M1	1	2	3	W1	1	2	3
M2	2	1	3	W2	2	1	3
M3	3	2	1	W3	3	2	1
Man's ranking				Woman's ranking			

1. *Condition for Stability: The set of marriages ((M1, W1), (M2, W2), (M3, W3)) is stable because each man has got his first choice, and each woman her first choice as well.*
2. *Condition for Unstability: The set of marriages ((M1, W2), (M2, W2), (M3, W3)) is not a stable one because there exists a man, namely M1, and a woman, namely W1, who prefer each other to their present partners; M1 prefers W1 to W2 and W1 prefers M1 to M2.*

Example 3.2: Consider the following large situations:

M1	1	2	3	4	5	6	W1	1	2	3	4	5	6
M2	2	1	3	4	5	6	W2	2	1	6	3	4	5
M3	3	1	2	3	4	5	W3	3	2	1	4	5	6
M4	4	5	6	3	1	2	W4	4	3	6	5	2	1
M5	5	6	4	3	2	1	W5	5	4	3	2	1	6
M6	6	5	1	2	3	4	W6	6	5	4	3	2	1
Man's ranking							Woman's ranking						

1. *Condition for Stability: The set of marriages ((M1, W1), (M2, W2), (M3, W3), (M4, W4), (M5, W5), (M6, W6)) is stable because each man has got his first choice, and each woman her first choice as well.*
2. *Condition for Unstability: The set of marriages ((M1, W1), (M2, W1), (M3, W3), (M4, W4), (M5, W5), (M6, W6)) is not a stable one because there exists a man, namely M1, and a woman, namely W1, who prefer each other to their present partners; M1 prefers W1 to W2 and W1 prefers M1 to M2.*

3.2.2 *Interconnection Networks*

INs are a major factor that differentiates modern multiprocessor architectures and can be categorized according to a number of criteria such as topology, routing strategy, and switching technique [46]. INs are made up of switching elements (SE); topology is the pattern in which the individual switches are connected to other elements, like processors, memories and other switches. There are two types of topologies and these are Direct topologies that connect each switch directly to a node, while in Indirect topologies at least some of the switches connect to other switches.

Using switching technique as a criterion one can mention some classes:

Definition 3.1. Circuit Switching. In which the entire path through the network is reserved before a message is transferred.

Definition 3.2. Packet switching with virtual cut `through. In which a packet is forwarded immediately after it determines an appropriate switch output.

Definition 3.3. Wormhole Routing. This relaxes requirements of completely buffering of blocked packets in a single switch, typical for packet switching. Wormhole routing is currently the most popular technique in commercial parallel machines.

The networks can be classified as static or dynamic. Static INs are mainly used in message–passing architectures; the following types are reported in the literature:

1. *Completely–connected Network.*
2. *Star–connected Network.*
3. *Ring of Processors.*
4. *Mesh Network. Each processor has a direct link to four/six (in 2D/3D) neighbor processors. An extension of this kind of networks is a wraparound mesh or torus.*
5. *Tree Network of Processors. Communication bottleneck which is likely to occur in large configurations can be alleviated by increasing the number of communication links for processors closer to the root, which results in the fat–tree topology, efficiently used in the TMC CM5 computer. CM5 could be also an example of indirect network topology.*
6. *Hypercube Network. This is a multidimensional mesh of processors with exactly two processors in each dimension. An example of such a system is the Intel iPSC/860 computer.*

Dynamic INs are available as below:

1. *Bus–based Networks. The simplest and efficient solution when the cost and moderate number of processors are involved. Its main drawback is a bottleneck to the memory when number of processors becomes large and a single point of failure. To overcome the problems, sometimes several parallel buses are incorporated* [47]. *The classical example of such machine is the SGI Power Challenge computer with packet data bus.*
2. *Crossbar Switching Networks. The network employs a grid of SEs and is nonblocking, since the connection of a processor to a memory bank does not block the connection of any other processor to any other memory bank. In spite of high speed, their use is limited, due to nonlinear complexity (* $O(p^2), p-$ *number of processors). Its applications are mostly in multiprocessors like Cray YMP and in multiprocessors with multilevel interconnections.*
3. *Multi–stage Interconnection Networks. These networks formulate the most advanced pure solution, which lies between the two extremes. A typical example is the omega network, which consists of* $\log p$ *stages, where p is number of inputs and outputs (usually number of processor and of memory banks). Its complexity is* $O(p \log p)$*, less than for the crossbar switch. However, in the omega network some memory accesses*

can be blocked. Although machines of this kind of interconnections offer virtual global memory model of programming and still not much popular in view of ease of use. Example BBN Butterfly, at present IBM RS6K SP incorporates multistage interconnections with Vulcan switch.

3.2.3 *Multi–stage Interconnection Networks*

MINs consist of multiple stages of SEs. Popular among them is a class of regular networks which in their basic form, consist of $\log_m N$ stages of m x m SEs connecting N input terminals to N output terminals. Sometimes MINs can also be built using large SEs and correspondingly have less number of stages, with similar properties. There exist many different topologies for MINs, which are characterized by the pattern of the between links between stages. The omega network maintains a uniform connection pattern between stages, known as perfect shuffle; many other MIN have non–uniform connection pattern between stages. The minimum requirement of any of these networks is to provide full access capability, which means that any input terminal of the network should be able to access any output terminal in one pass through the network. MINs differ in the interconnection pattern between stages, the type, and operation of individual SEs and the control scheme for setting up the SEs. However, data travel through different switches. Hence, these networks have a longer internal delay as compared to crossbar.

Based upon the topology a Dynamic MIN can be regular, irregular, and hybrid.

Definition 3.4. Any network is regular if the number of SEs in different stages of the network is same.

Definition 3.5. Any network is irregular if the number of SEs in different stages of the network is different.

Definition 3.6. Any network is Hybrid if the network possesses the regular as well as irregular topology.

3.2.3.1 *Network Architecture of Hybrid ZETA Network*

A HZTN (Figure (3.1)) of size $2^n x\ 2^n$ (where 2^n are Sources, 2^n are Destinations, $n = \log_2 N$ and $m = \log_2(N/4)$) is constructed with two identical groups of SEs G^N [where (N = 0, 1) i.e. G^0 and G^1], the two groups are formed based on the most significant bit (MSB) of the source–destination terminals. Thus, half of the source–destination terminals with MSB 0 fall into the G^0 group and the others having MSB 1 fall into G^1 group.

Each source and destination is connected to both groups with the help of multiplexer (MUX) and demultiplexers (DEMUX). The Source (S) and Destination (D) of the MIN in binary code are represented as:

$$S = s_{n-1} \ldots\ldots\ldots s_1.s_0$$

$$D = d_{n-1} \ldots\ldots\ldots d_1.d_0$$

The source and destinations are connected to the MUX and DEMUX are as follows: If $(s_{n-2} \ldots\ldots\ldots s_1.s_0)$ bits are the same for the two sources, then these two sources are linked through the same pair of MUX and if $(d_{n-2} \ldots\ldots\ldots d_1.d_0)$ bits are the same for the two destinations, then these two destinations are linked through same pair of DEMUX. From now onwards the different stages of the network i.e. stage 0, stage 1, stage 2, stage 3 and stage 4 will be denoted as $(2m-4)$, $(2m-3)$, $(2m-2)$, $(2m-1)$ and $(2m)$. The HZTN is regular in 2 stages i.e. stage $(2m-4)$ and $(2m)$ but irregular in 3 intermediate stages i.e. stage $(2m-3)$, $(2m-2)$ and $(2m-1)$. The overall topology of HZTN is irregular with 5 stages. The SEs pertaining to the two groups with the same number in all the stages, except in the final stages, is connected to each other through links known as express links, which are used only if the SE in next stage is busy or faulty these interconnected switches are called Associative Switches. Its circuitry is having in total of $(2^{m+3}-4)$ number of SEs, out of which $(2^{n-1}+4)$ are of size 2 x 2 and rest $(2^{n-1}+8)$ are of size 3 x 3 all are connected to source and destination by $(2\,x\,1)\ 2^n$ MUX and $(1\,x\,2)\ 2^n$ DEMUX. Each of the stage $(2m-4)$ and $(2m)$ stage has exactly (2^{n-1}) switches. Total number of associative switches in intermediate stages i.e. stage $(2m-3)$, $(2m-2)$ and $(2m-1)$ is $(3*2^{n-2})$. The comprehensive (i.e. topology, construction, routing procedure, reliability, and fault–tolerance) study of HZTN can be found in [2–4,19].

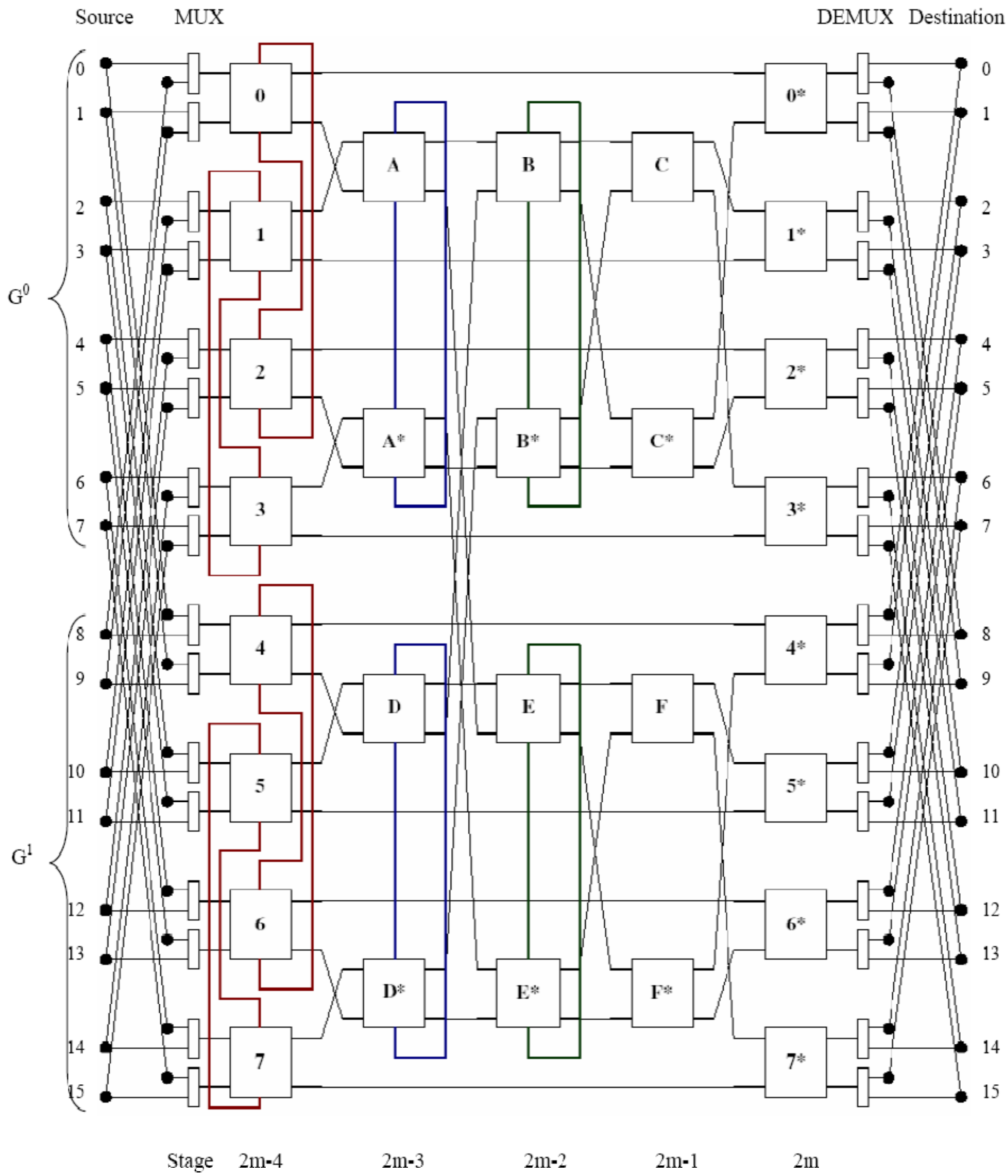

Figure 3.1. A 16 x 16 Hybrid Zeta Network (HZTN).

3.2.3.2 *Network Architecture of Quad–tree Network*

A $2^n x\ 2^n$ QTN (Figure (3.2)), is constructed by using two identical groups each consisting of modified double–tree networks (MDOT) [41] of size N/2 x N/2, which are arranged one above the other. A MDOT is a modified version of a double–tree networks (DOT) [48], consists of a right and a left half. Each half of the network resembles a Binary tree, with the left half and right half being mirror image of each other. In particular a DOT of size N x N has N source and N destination terminals, comprising $(2n-1)$ number of stages with $(2^{n+1}-3)$ SEs. A i th and $(2n-1)$ th stage has (2^{n-i}) SEs of size 2 x 2 and for $i = 1,2,3.....n$, irregular DOT has shorter path length for a connection between a processor and its favorite memory module.

In QTN, loops are formed by the switches having the same number in the same stage, which is formed in all the stages except the last one. The QTN is regular in 2 stages i.e. stage 0 and 4 but irregular in 3 intermediate stages i.e. stage 1, 2, and 3. The overall topology of QTN is irregular with 5 stages. The SEs pertaining to the two groups with the same number in all the stages, except in the final stage, is connected to each other through links known as express links, which are used only if the SE in next stage is busy or faulty. These interconnected switches are called Associative Switches.

The QTN circuitry is having in total of $(2^{m+3}-6)$ SEs, out of which (2^{n-1}) are of size 2 x 2 and rest $(2^{n-1}+10)$ are of size 3 x 3 all are connected to source and destination by $(2\,x\,1)\ 2^n$ MUXs and $(1\,x\,2)\ 2^n$ DEMUXs. Each of the stage 0 and stage 4 has exactly (2^{n-1}) SEs. Total number of associative switches in intermediate stages is $(3*2^{n-2}-2)$ (where 2^n are Sources, 2^n are Destinations, $n = \log_2 N$ and $m = \log_2(N/4)$). Alternate paths are available for the data from an input port to reach its final output port selected based on an algorithm that bypasses a faulty or a busy SE. The delay of the network is directly proportional to the path length encountered on the way to the destination. The comprehensive (i.e. topology, construction, routing procedure, reliability, and fault–tolerance) study of QTN can be found in [41].

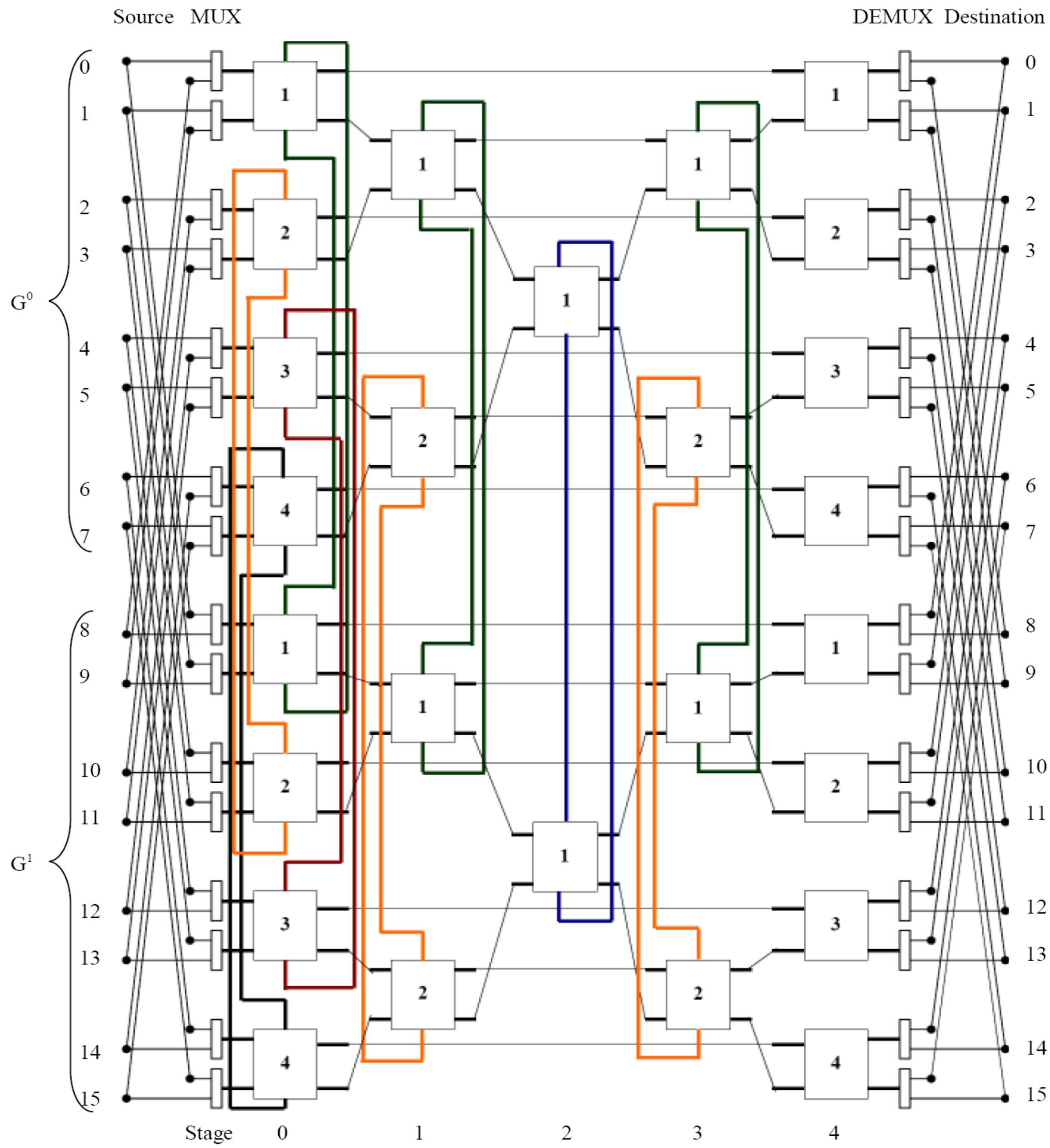

Figure 3.2. A 16 x 16 Quad–tree network (QTN).

3.2.3.3 *Network Architecture of Augmented Shuffle–Exchange Network*

A $2^n x\ 2^n$ ASEN (Figure (3.3)), is a regular hybrid MIN by its class and is constructed using the well–known shuffle–exchange MIN. The study of shuffle–exchange MIN is attributed to [49,50]. Briefly an ASEN is constructed by adding a stage of 2 x 1 MUX and 1 x 2 DEMUX at the input side and output side of the switches and by adding chaining links (also known as express link–used for providing the fault–tolerance and to improve the speed of the network) to connect certain groups of switches within each stage in loops (to provide an alternate way of routing in each stage). There are several versions of the ASEN, i.e. ASEN–4, ASEN–2 etc. depending upon the number of switches included in each conjugate loop (e.g. loop containing SE 0 & 2 and SE 1 & 3 in first stage). In this chapter, the discussions are limited to ASEN–2, in which the loops contain exactly two switches. The comprehensive (i.e. topology, construction, routing procedure, reliability, and fault–tolerance) study of ASEN can be found in [42,43].

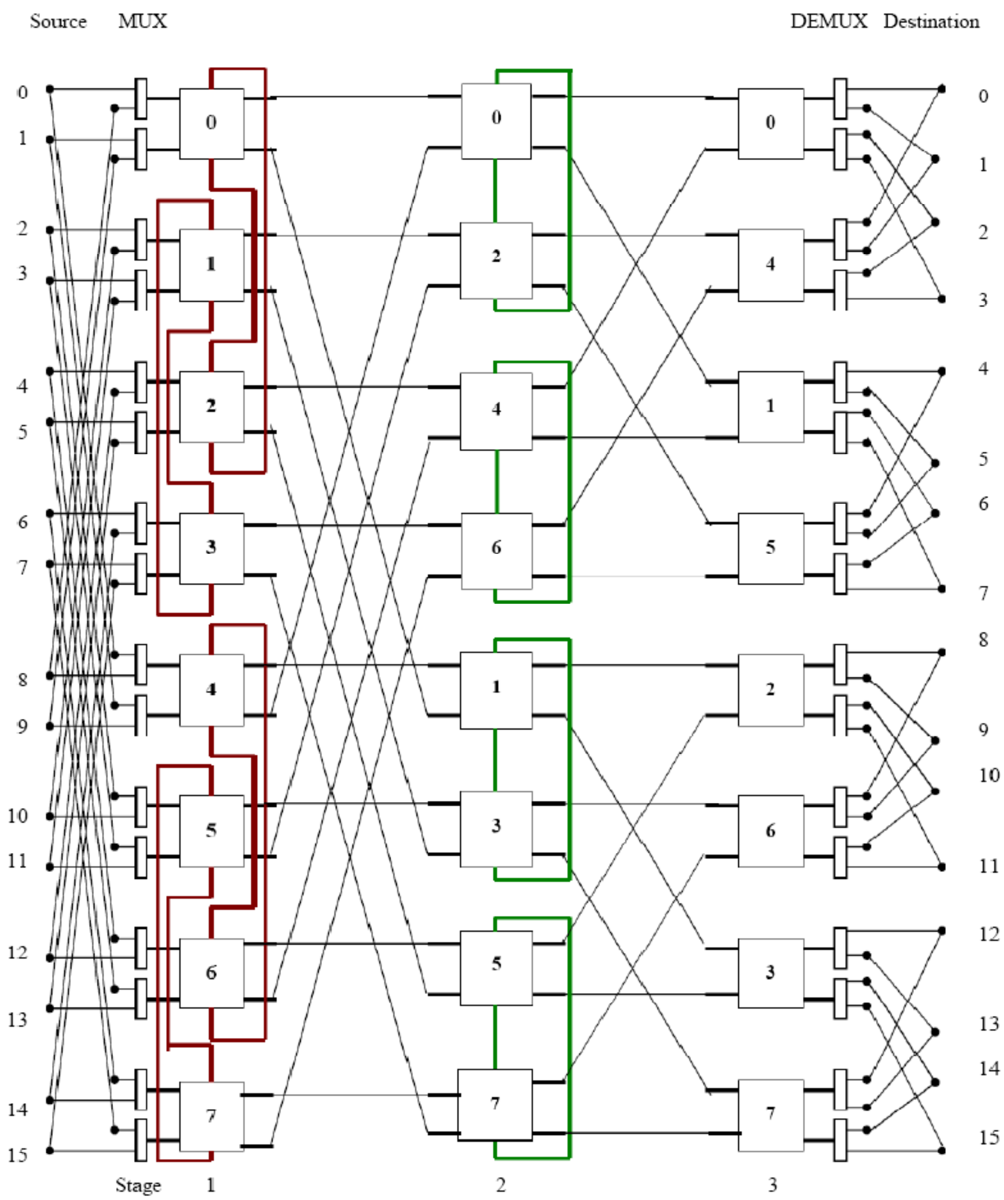

Figure 3.3. A 16 x 16 Augmented Shuffle–exchange Network (ASEN).

3.2.3.4 *Network Architecture of Augmented Baseline Network*

A $2^n\, x\ 2^n$ ABN (Figure (3.4)) consists of two identical subnetworks each consisting of N/2 sources and equal number of destinations. Links are provided among switches belonging to the same stage, forming several loops of switches of size 3 x 3 in all stages except the last one where the SEs are of size 2 x 2, and are connected to MUXs and DEMUXs at the input stage and at the output stage respectively. According to the design and architecture of the ABN network the conjugate loop, (e.g. loop containing SE 0 & 2 and SE 1 & 3 in first stage). The comprehensive (i.e. topology, construction, routing procedure, reliability, and fault–tolerance) study of ABN can be found in [44].

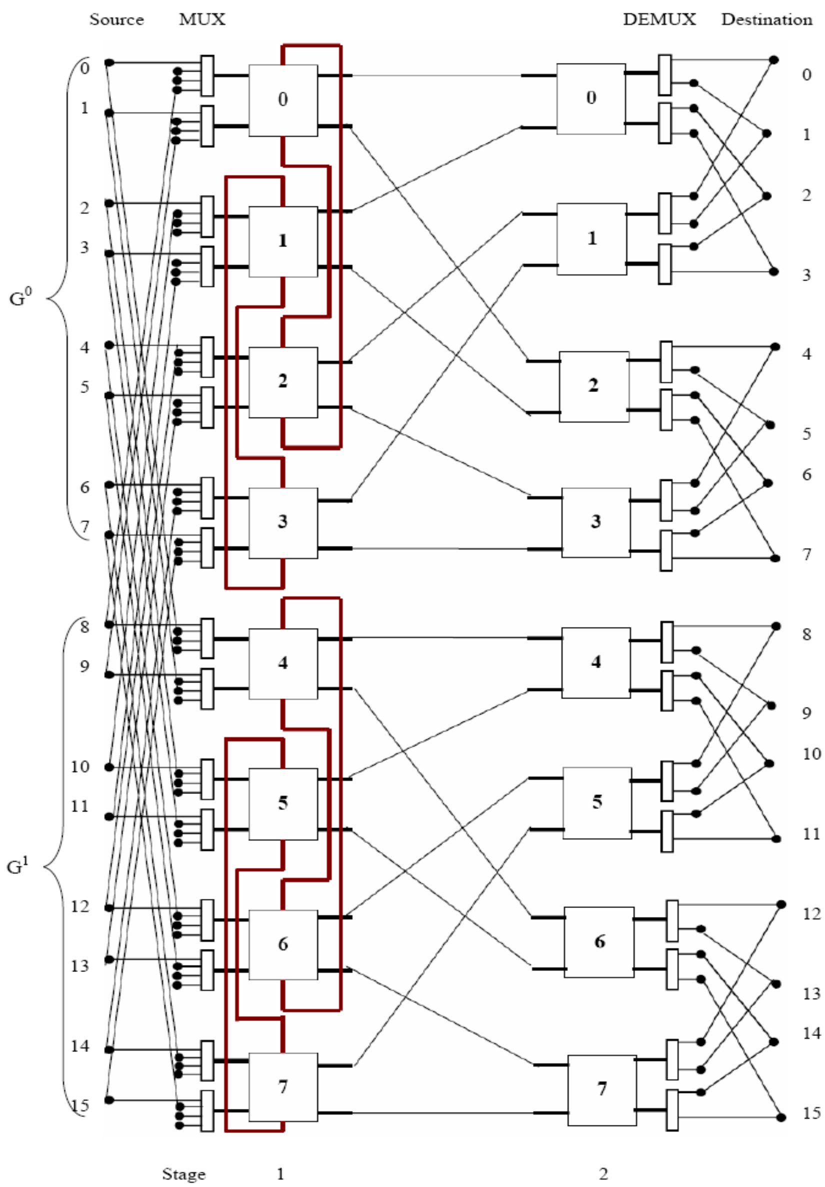

Figure 3.4. A 16 x 16 Augmented Baseline Network (ABN).

3.3 *Solving Multi–stage Interconnection Networks Stability Problem using Stable Matching*

3.3.1 *MINs Stability Problem*

MINs provide an easy way through which the information is routed via the specified switches, however it varies greatly with the type of topology that is used and it may be unstable for many instances. The routing mechanisms via the switches occur through the path–length algorithm upon the basis of which the shortest path to the destination is selected. Unstability in any MIN (regular or irregular) may occur if at any instance, a node fails and no alias path is available for routing through any of the nodes.

The switches are highly independent of each other as such no conjugation occurs amongst them thereby yielding no possible track and leaves the entire network as unstable. As the switches have no dependency, no backtracking mechanism is available thus if the initial nodes as in Figures (3.8, 3.12, 3.16, 3.20) fail the path is deadlocked and the entire network becomes unstable. The topology of the network has little significance associated with the unstability, as the network is not fault tolerant in case of failure thus unstability is bound to occur. The switches are unaware of the next immediate/most optimal path to follow to achieve successful delivery thereby deadlock remains causing unstability.

3.3.1.1 *Conjugation*

Here HZTN is taken as the example and the definition given here is remaining same for all other MIN discussed further. In Figure (3.1), each of the SE in the first stage is connected with the source through 1 x 2 MUX. In both the subnetworks (i.e. G^0 and G^1) of HZTN, have the conjugate pairs (in stage 1 of Figure (3.1), SE 0 – 7 forms a conjugate subset; within that subset, SE 0 & 1 and SE 4 & 5 are a conjugate pair; and SE 0 & 2, SE 1 & 3, SE 4 & 6, and SE 5 & 7 forms a conjugate loop). The HZTN is operational as long as one of the MUX attached with the 3 x 3 SE in first stage is operational and the conjugate pair in their respective subnetworks is operational [3,4].

3.3.2 *Stable Matching and MINs Stability Problem*

As mentioned in the above context due to the unstability of the network it becomes less fault–tolerant, which leads to deadlock situation. To tackle this problem a mechanism of stable matching is improvised to prevent failure from occurring. Since at every level n = 0, 1, 2, 3 the switches are aware of their immediate neighbors, chooses the best fit on the basis of the preference list created using the path–length approach from the preference matrix of the specific MIN.

With the application of the stable matching approach even in case of the path failure the conjugate pairs are active based on preference path length and the desired path is accepted. The same is carried forward for the backtracking approach thus no path is failed at every segment hence even in case of failures of first SE of each segment there is an alias path available so that the destination is reached. For example:

Case 1: In Figure (3.5) if SE 1 fails then it backtracks with an alternate path to SE 5 and reaches the destination via it with a path–length of 4 hence successfully transferring data that can be seen from the preference matrix of the switches in Figure (3.6) resulting in keeping the entire network stable as no congestion occurs.

Case 2: Similarly in Figure (3.9) if SE 1 fails then it backtracks with an alternate path to SE 5 and reaches the destination via it with a path–length of 4 hence successfully transferring data which again can be seen from the preference matrix Figure (3.10) resulting in keeping the entire network stable.

Case 3: In Figure (3.13) if SE 1 fails then it backtracks to SE 5 and reaches the destination via it with a path–length of 4 hence successfully transferring data which can be verified from the preference matrix Figure (3.14).

Case 4: In Figure (3.17) if SE 1 fails then it backtracks to SE 5 and reaches the destination via path–length of 4 hence successfully transferring data which can be verified from the preference matrix Figure (3.18) resulting in keeping the entire network stable.

3.3.3 *Assumptions*

Before writing the required algorithm, here are some assumptions that have to be taken care of. The assumptions while implementing the stable matching algorithm as following:

1. *No Consideration of the paths from the MUX and DEMUX: When the circuit is being considered on a whole the calculation of the path length algorithm will be made from the node marked as 0, 1, 2, 3, 4, 5, 6, 7 and the distances or paths from the MUXs and DEMUXs shall be neglected.*
2. *Conjugate Pairs of Nodes in the Network: The circuit thereby consists of segments* G^0 *and* G^1 *the corresponding alternate pairs of the levels 0, 1, 2 have conjugate roots between them thus the path can be traversed from these possible routes however it increases the net effective cost involved as it increases the specified path length.*
3. *Priority of the Traversal of the Paths: The algorithm that is employed in the calculation of paths is based on the concept of path length algorithm. Priority is given to the node by means of which the destination can be reached in minimum time and cost in comparison to any other node in the entire circuit.*
4. *Neglecting pairs of the level with minimum number of nodes: Since at the level with number of nodes the amount of inflowing paths is very high thereby the probability of it selecting the most optimized pair is very low, as it has multiple out flowing paths to the destination of relatively similar path lengths within the circuit and hence, get neglected.*

The sole purpose of choosing the assumptions is the fact that without them the stability of the network cannot be proved. There are a large number of observable features present in the network that have to be neglected to prove the above cause. Assumptions have been included to enhance our effort to provide an efficient approach in proving the network to be stable. In addition, by assuming them, it helps us to decide the broad criteria of defining the constraints under which the network is going to act effectively and the concept of stable matching augmented well. The fundamental approach of assuming these conditions is to provide us with an initial approach that laid emphasis on the key aspect of stability using stable matching algorithm. Furthermore, if these assumptions are not considered then it will leads to NP–Completeness problem.

3.3.3.1 *NP–Completeness*

A problem is called NP (nondeterministic polynomial) if its solution (if one exists) can be guessed and verified in polynomial time, nondeterministic means that no particular rule is followed to make the guess. Thus, finding an efficient algorithm for any NP–Complete problem implies that an efficient algorithm can be found for all such problems, since any problem belonging to this class can be recast into any other member of the class. As far as the above solution is concerned the problem of NP–Completeness arises from the fact that the stability of the network can be rendered from the stable matching approach in a polynomial time solution hence the problem is solved. The solution of the optimal pairs of all the networks as per the algorithm is given above and is produced assuming into consideration of all the assumptions otherwise the solution fails.

Applying by the concept of stable matching, it will render us with an exact solution to the above problem in a defined polynomial time expression.

3.3.4 *Algorithm for Deriving Preference Lists from the MINs*

The algorithm to generate the preference lists of the MIN is explained here. This algorithm is on the similar lines of the Gale–Shapley Algorithm.

Algorithm: PREFERENCE_LISTS

Inputs: Priority of SE/Nodes based on shortest path concept of reaching the goal.
Output: Provides a Priority Preference Lists from which the Optimal Pairs are selected.
Precondition: Each list has a collection of only those SE that in turn are always connected to.
Postcondition: The Optimized Preference lists are generated.

```
1.  Stable ← TRUE (No condition of Tie occurs with two SE having the same Priority Pairs)
2.  FOR each Switch SE 1
3.    FOR each Switch SE 2
4.        IF ((SE 1 prefers SE 2 to its existing pair as it has a shorter path length to reach
              Destination SE and both are connected)) and
            ((SE 2 prefers SE 1 to its existing pair as it has a shorter path length to reach
              Destination SE and both are connected))
5.          THEN the Switches SE 1 and SE 2 exist mutually in their list.
6.            ELSE IF (If SE 1 and SE 2 have Tie for their list elements order them both in their
                    lists)
7.                ELSE (If SE 1 and SE 2 do not have a path amongst each other)
8.                    WRITE "SE 1 and SE 2 do not have a Stable Pair"
9.                            Stable ← FALSE
10.           END IF
11.     END IF
12.   END FOR
13. END FOR
14. WRITE "The Preference Lists is generated"
```

15. EXIT

Complexity: The run time complexity of the Algorithm: PREFERENCE_LISTS is $O(n^2)$.

Proof of Complexity or Correctness:

Let SE 1 = SE 2 = n

For lines from #2 to 13 the Time = $n\ x\ n$ (time taken in generating the MINs preference lists)

$$= n^2\ x \text{ Constant}$$

Therefore, Complexity in Big (O) notation is $O(n^2)$.

3.3.5 *Reduction of the Ties in Irregular and Regular MINs*

The reduction of the ties in the irregular and regular networks is discussed here. After deriving the preference lists of an irregular and regular network that has been created based on the patterns described in the previous section a basic aspect that has borne in mind is that while creating the preference list there are a large number of cases where ties occurs, which means for a specific SE that has to be resolved as it will result in congestion as two pairs have the same pair of optimal switches defined in the preference list. Thus in such a case priority is set in a such a way that the switch next in the list is tested for priority with all other switches and case of resolution of this clause it is allocated to the specific switch/ node and if this is not acceptable the procedure is carried on with other switches in the list and vice versa.

3.3.6 *Deriving Optimal Pairs from MINs Preference Lists*

The algorithm for a solution to a stable marriage instance in MIN is based on a sequence of "proposals" from one switch to the other based on shortest path length to reach the destination. Each switch proposes, in order, to the nodes (switches) on his preference list, pausing when a node agrees to consider his proposal, but continuing if a proposal is rejected either immediately or subsequently.

When another node receives a proposal, it rejects it if the specified node already holds a better proposal, but otherwise agrees to hold it for consideration, simultaneously rejecting any poorer proposal that the node may currently hold i.e. the preference is given to the node which is higher or first in the priority list than any other nodes also specified later in its specific list.

It is not difficult to show, as in that the sequence of proposals so specified ends with every switch holding a unique proposal, and that the proposals held constitute a stable matching. The Two fundamental implications of this initial proposal sequence are:

1. *If SE 1 proposes to SE 2, then there is no stable matching in which SE 1 has a better partner than SE 2.*
2. *If SE 2 receives a proposal from SE 1, then there is no stable matching in which SE 2 has a worse partner than SE 1.*

These observations suggest us explicitly to remove SE 1 from SE 2's list and SE 2 from SE 1's list. If SE 1 receives a proposal from some node that is better in priority than SE 2 then the resulting lists or pairs as the shortlists for the given problem instance is referred.

Algorithm: SELECTING_STABLE_PAIRS

Inputs: Preference lists of SE.
Output: A matching consisting of list of engaged pairs.
Precondition: Each list includes the connection of one SE with all the other.
Postcondition: A matching is produced which is stable for each SE.

```
FOR each Switch SE
        Engaged (SE) ← FALSE
END FOR
 WHILE there is a SE which is not engaged
      FOR each Switch SE y
            IF Switch SE y is not yet engaged
              THEN SE x ← highest on SE y list, which is not yet engaged
                     ADD (SE y, SE x) to the Stable Pair List
            END IF
      END FOR
 END WHILE
 Write "List of Optimal (Stable) Pairs"
```

<u>Complexity</u>: The run time complexity of the Algorithm: SELECTING_STABLE_PAIRS is $O(n)$.

Proof of Complexity or Correctness:

Let time of adding (SE y, SE x) to stable pair list = t_1

Number of SEs = n

Hence Time = $n \; x \; t_1$

Therefore, Time Complexity in Big (O) notation is $O(n)$

3.3.7 *Application of Stable Matching Approaches to Solve MINs Stability Problems*

3.3.7.1 *HZTN*

The path–length algorithm (which helps in deriving the preferences list) for every source–destination pair for the levels m = 0, 1, 2 is explained here. The application of the algorithm starts from the nodes (see Figure (3.5)) entitled 1, 2,……….,8 and up till nodes entitled 20, 21, ………..,28. However, the nodes of level 2 i.e. node 20, 21,………,28, are not taken into consideration as all of them are the destination nodes.

In the algorithm the source and the destination is represented as:

$$S = s_{n-1} \ldots\ldots\ldots\ldots\ldots\ldots s_1.s_0$$

$$D = d_{n-1} \ldots\ldots\ldots\ldots\ldots\ldots d_1.d_0$$

The path–length algorithm is as follows: To derive the path–length algorithm for the HZTN the following variable are to be assumed:

1. *j = Total number of stages,*
2. *n = Starting source (0) in most cases,*
3. *m = Number of switches,*
4. *d = Destination number,*
5. *A = Number of states,*
6. *B = Number of switches in first level* G^0 *or* G^1*, and*
7. *C = Number of reflective switch in middle order.*

Consider here Operator: – $\otimes$ = Comparator with NOT Gate = for same input, it will give 0 as output and for different input it will give 1 as output.

Algorithm: HZTN_PATH_LENGTH

---

1. IF $\{(s_n \otimes d_n)+(s_{n+1} \otimes d_{n+1})+.....+(s_{j-1} \otimes d_{j-1})\} := 0$
2. THEN Minimum path length for s_n = 2
3. ELSE IF $\{(s_{n+1} \otimes d_{n+1})+(s_{n+2} \otimes d_{n+2})+.....+(s_{j-1} \otimes d_{j-1})\} := 0$
4. THEN Minimum path length s_n = 2
5. ELSE IF $\{(s_{j-1} \otimes d_{j-1})\} = 0$
6. THEN Minimum path length s_j = 2 and all
   Path lengths ≥ 2 are possible
7. ELSE longest path length: $=(2m+1)$ = 5 i.e. path length = $\frac{(A+B)}{C}$ = 5

Example 3.3. Consider the case in which a data has to be routed from source 0 = 0000 to destination 15 = 1111. Therefore, according to the algorithm the following calculations has to be follow and that are:

$0 \otimes 1 + 0 \otimes 1 + 0 \otimes 1 = 0$

Then path–length = 2 and 5 are possible. i.e. in this particular case the path length will be 5 and the specified route from S = 0000 to Destination = 1111 will be SE 5 – SE 11 – SE 15 – SE 19 – SE28.

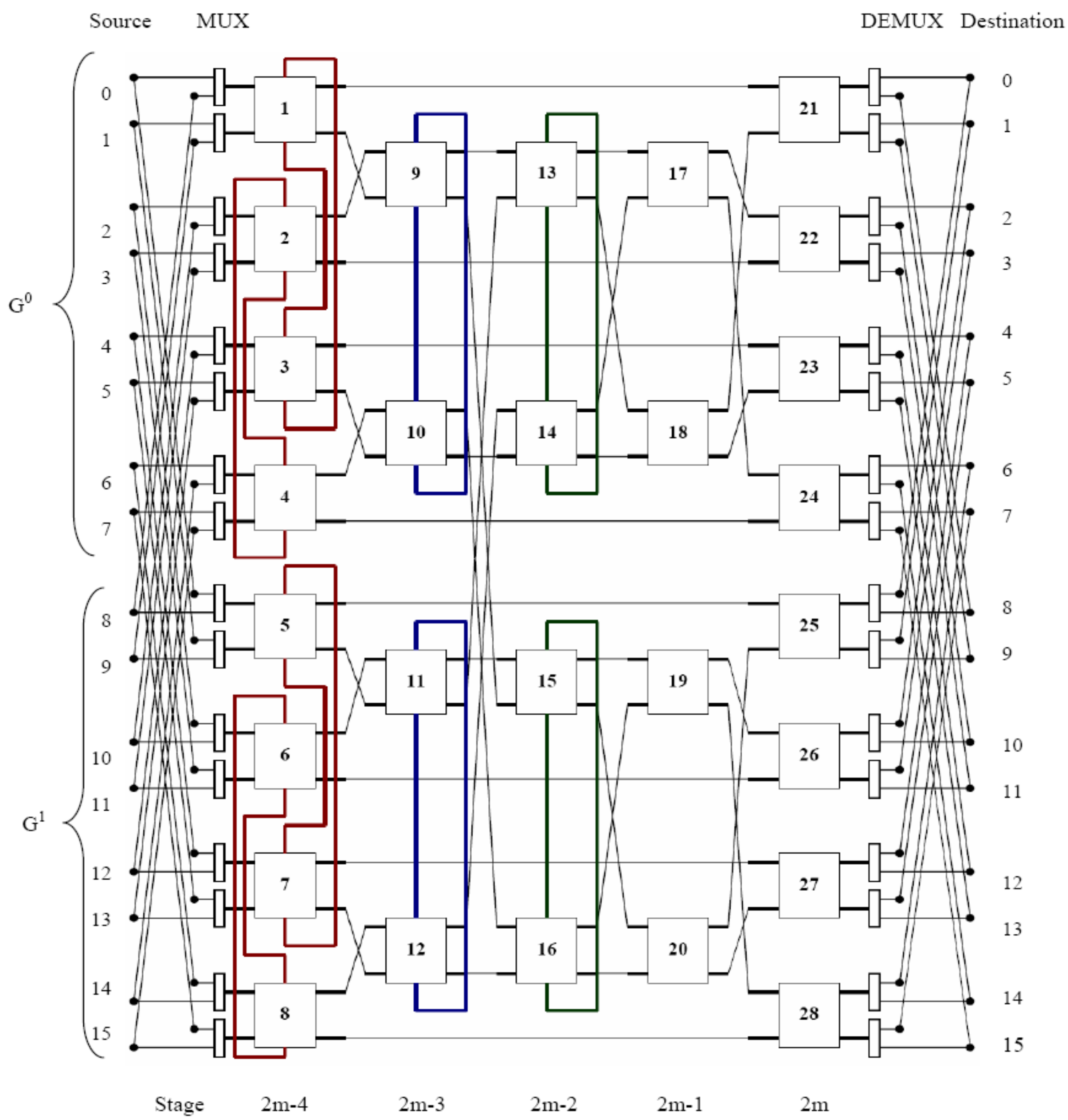

Figure 3.5. A 16 x 16 HZTN. Here SEs are renumbered to solve the stability problem.

3.3.7.1.1 *Preference Lists*

Refer algorithm explained in Section (3.3.4) for deriving preference lists for the HZTN. The SEs in Figure (3.1) are renumbered and put up again in Figure (3.5). Figure (3.6), shows the preference lists having Ties. All Ties have been solved for HZTN.

SE 1	21	8	3	23	13	17	22	24	10	14	18	15	19	26	28	16	20	27	25
SE 2	22	9	4	24	13	17	22	24	10	14	18	15	19	26	28	16	20	27	25
SE 3	23	10	1	21	14	18	9	13	17	22	24	16	20	25	27	15	19	26	28
SE 4	24	10	2	22	14	18	9	13	17	22	24	16	20	25	27	15	19	26	28
SE 5	25	11	7	27	15	19	26	28	12	16	20								
SE 6	26	11	8	28	15	19	12	16	20	25	27								
SE 7	27	12	5	25	16	20	11	15	19	26	28								
SE 8	28	12	6	26	16	20	25	27	11	15	19								
SE 9	13	17	22	24	10	14	18	21	23	15	19	26	28	16	20	25	27		
SE 10	14	18	21	23	9	13	17	22	24	16	20	25	27	19	26	28			
SE 11	15	19	26	28	12	16	20	25	27										
SE 12	16	20	25	27	15	19	26	28											
SE 13	17	22	24	14	18	21	23												
SE 14	18	21	23	13	17	22	24												
SE 15	19	26	28	16	20	25	27												
SE 16	20	25	27	15	19	26	28												
SE 17	22	24																	
SE 18	21	23																	
SE 19	26	28																	
SE 20	25	27																	

Figure 3.6. The complete preference lists of the HZTN.

3.3.7.1.2 *Reduction of the Ties*

Refer procedure explained in Section (3.3.5). The same is used here to derive the optimal pairs for the HZTN. However, in case of the above network fault tolerance scheme is provided at level 3 as a situation occurs of all the switches SE 17 18 19 20 have a priority of the destination switches, which already have optimal pairs. Since the transference from these nodes has been made in such a way that the next possible switch is a destination the above switches are henceforth neglected.

See Figure (3.6) the preference list for the SE 17 and SE 18 stands as:

SE 17   22 24

SE 18   21 23

Both the above cases have been rendered in such a method that SE 22 and SE 23 comes in priority 1 of them as such both can form the optimal pairs. However, SE 2 has SE 22 as an optimal pair thereby it can be neglected.

3.3.7.1.3 *Deriving Optimal Pairs from the Preference Lists*

Refer procedure explained in Section (3.3.6). The same is used here to derive the optimal pairs for the HZTN. See Figure (3.6) the preference list of HZTN MIN for the switch SE 1 stands as:

SE 1   21 8 3 23 13 17 22 24 10 14 18 15 19 26 28 16 20 27 25

SE 1 has highest priority been set to SE 21 as such appears first in the priority list and next priority has been set to SE 8 as such appears second and SE 3 as third followed by SE 23 thereby the rest exists as similar pattern.

As it can be seen in Figure (3.6), that SE 3 has specified as:

SE 3   23 10 1 21 14 18 9 13 17 22 24 16 20 25 27 15 19 26 28

Thus the priority of SE 23 is more on the of SE 3 thus both will exist as a stable matched pair and the set can be stable thus reduce the above list by eliminating the corresponding SE 23 from the list of SE 1 as someone else (SE 3) holds a better proposal for the SE 23 to follow the path and reach to its destination.

SE 1   21 8 3 – – – 13 17 22 24 10 14 18 15 19 26 28 16 20 27 25   and it becomes;

SE 1   21 8 3 13 17 22 24 10 14 18 15 19 26 28 16 20 27 25

Similarly the nodes SE 17 and SE 22 occur higher in the priority list of switches SE 13 and SE 2 thereby the above two Switches are eliminated from the list of SE 1 and similarly the final list of optimal set is:

SE 1 21 8 3 10 14 15 19 16

Thus, the final pair becomes SE 1 and SE 19, which is stable in nature. Based on this assumption and analysis the following Figure (3.7) {which shows all the optimal pairs} has been compiled.

$$(1,21),\ (2,22),\ (3,23),\ (4,24),\ (5,25),$$
$$(6,26),\ (7,27),\ (8,28),\ (9,13),\ (10,14)$$
$$(11,15),\ (12,16),\ (13,17),\ (14,18),\ (15,19), \textit{and}$$
$$(16,20)$$

Figure 3.7. The optimal pairs, which have been short–listed from the HZTN preference lists.

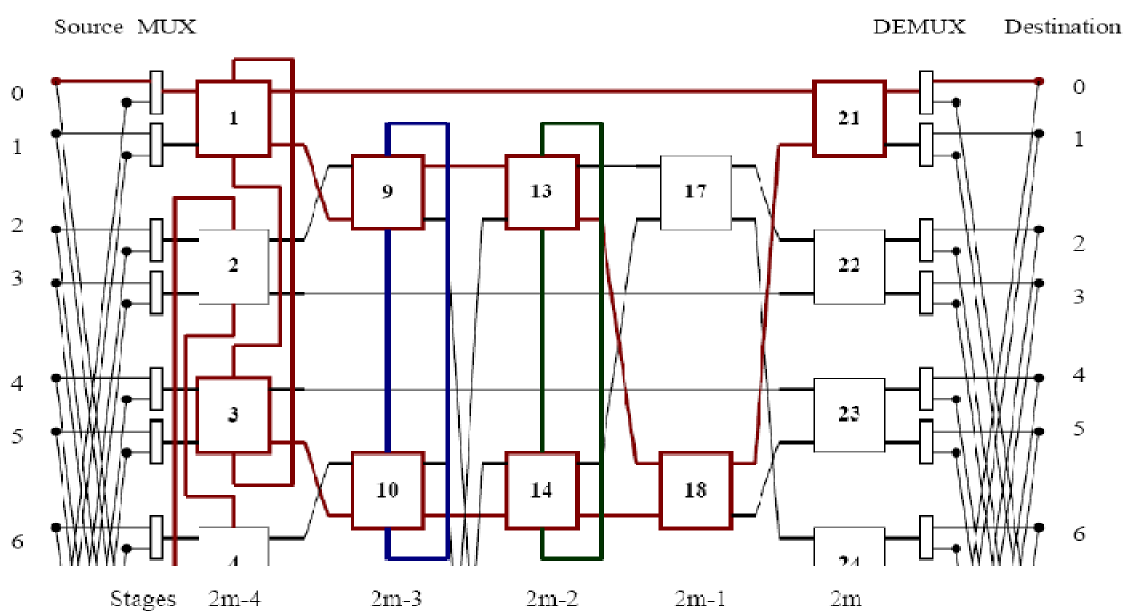

Figure 3.8. The partial cut way part of HZTN.

Example 3.4: See Figure (3.8) and Table (3.1) for all possible routes and path–lengths. In this particular example a request is routed from source 0 to destination 0 i.e. source 0000 to destination 0000.

Table 3.1. The routing table of HZTN.

Routes	Path–length
SE 1 – SE 21	2
SE 1 – SE 9 – SE 13 – SE 18 – SE 21	5
SE 1 – SE 3 – SE 10 – SE 14 – SE 18 – SE 21	5

Explanation: In this example (Table (3.1)), all the possible paths from the source to destination are explored. To route a request from a given source to given destination can have possible route and possible path–lengths. In this particular example, there are three paths from one source to destination. The first path (SE 1 – SE 21) is termed as the primary path, whereas the path (SE 1 – SE 9 – SE 13 – SE 18 – SE 21) is termed as secondary path and finally the path (SE 1 – SE 3 – SE 10 – SE 14 – SE 18 – SE 21) is the express path used, when the primary and secondary path are busy. The respective path–lengths at all the paths mentioned are 2, 5, and 5. Since QTN is an irregular MIN, therefore the 50% of the path–lengths are passed at the path–length of 2 and rest at path–lengths of 5.

3.3.7.2 *QTN*

The path–length algorithm (that helps in deriving the preferences list) for every source–destination pair for the levels m = 0, 1, 2 is explained here. The application of the algorithm starts from the nodes (see Figure (3.9)) entitled 1, 2,……….,8 and up till nodes entitled 19, 20,………..,26. However, the nodes of level 2 i.e. node 19, 20, ………,26, are not taken into consideration as all of them are the destination nodes.

In the algorithm the source and the destination is represented as:

$$S = s_{n-1}.................s_1.s_0$$

$$D = d_{n-1}.................d_1.d_0$$

Algorithm: QTN_Path_Length

1. IF $\{(s_{n-2} \oplus d_{n-2}) + (s_{n-3} \oplus d_{n-3}) + ... + (s_1 \oplus d_1)\} := 0$
2. THEN minimum path length of 2, 4, and 5 are possible to be followed in the network.
3. ELSE IF $\{(s_{n-2} \oplus d_{n-2}) + (s_{n-3} \oplus d_{n-3}) + ... + (s_2 \oplus d_2)\} := 0$

4. THEN path–length equal to 4 and 5 are only possible.
5. ELSE path–length of $2l-1$ is possible (where l = level = 3).

Example 3.5. Consider the case in which a data has to be routed from source 0 = 0000 to destination 15 = 1111. Therefore, according to the algorithm the following calculations has to be follow and that are:

$0 \oplus 1 + 0 \oplus 1 + 0 \oplus 1 = 0$

Then path–length = 2, 4 and 5 are possible. i.e. in this particular case the path length will be 5 and the specified route from S = 0000 to Destination = 1111 will be SE 1 – SE 5 – SE 11 – SE 14 – SE 18 – SE 26.

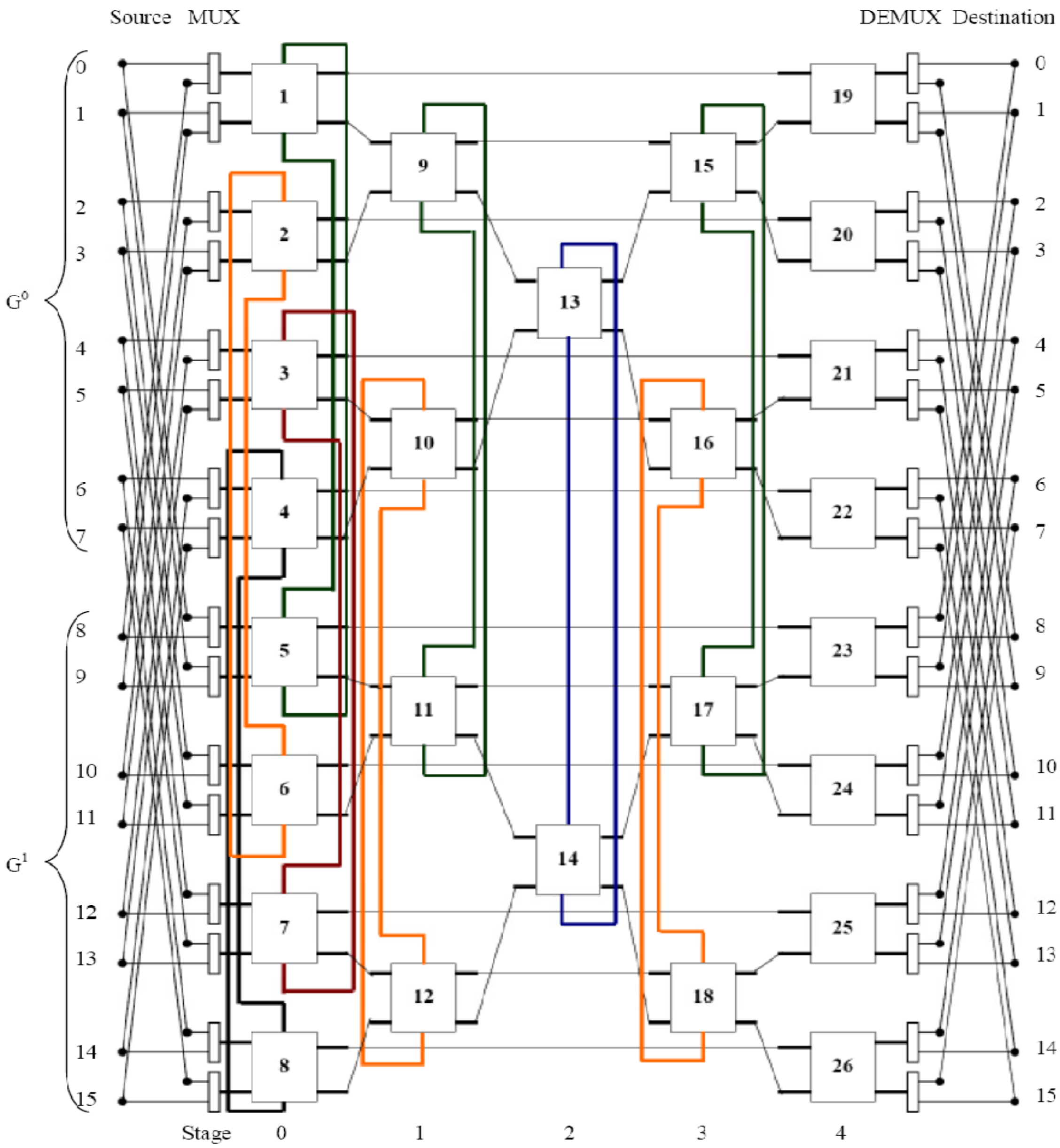

Figure 3.9. A 16 x 16 QTN. Here SEs are renumbered to solve the stability problem.

3.3.7.2.1 *Preference Lists*

Refer algorithm explained in Section (3.3.4) for deriving preference lists for the QTN. The SEs in Figure (3.2) are renumbered and put up again in Figure (3.9). Figure (3.10), shows the preference lists having Ties. All Ties have been solved for QTN.

SE 1	19	5	23	9	15	11	17	13	14	16	18	20	21	22	24	25	26
SE 2	20	6	24	9	15	11	17	13	14	16	18	21	22	23	25	26	
SE 3	21	7	25	10	16	12	19	14	13	17	15	19	20	22	13	24	26
SE 4	22	8	26	10	16	12	18	13	14	15	17	19	20	21	23	24	25
SE 5	23	1	19	11	9	15	14	13	16	18	20	21	22	24	25	26	
SE 6	24	2	20	11	17	9	15	14	13	16	18	19	21	22	23	25	26
SE 7	25	3	21	12	18	10	16	14	13	15	17	19	20	22	23	24	26
SE 8	26	4	22	12	18	10	16	14	13	15	17	18	20	21	23	24	25
SE 9	13	15	16	19	20	21	22	11	14	17	18	23	24	25	26		
SE 10	13	15	16	19	20	21	22	12	14	17	18	23	24	25	26		
SE 11	14	17	18	23	24	25	26	9	13	15	16	19	20	21	22		
SE 12	14	17	18	23	24	25	26	10	13	15	16	19	20	21	22		
SE 13	15	16	19	20	21	22	14	17	18	23	24	25	26				
SE 14	17	18	23	24	25	26	13	15	16	19	20	21	22				
SE 15	19	20	17	23	24												
SE 16	21	22	18	25	26												
SE 17	23	24	15	19	20												
SE 18	25	26	16	21	22												

Figure 3.10. The complete preference lists of the QTN.

3.3.7.2.2 *Reduction of the Ties*

Refer procedure explained in Section (3.3.5). The same is used here to derive the optimal pairs for the QTN. See Figure (3.10) the preference list for the SE 11 and SE12 stands as:

SE 11   14 17 18 23 24 25 26 9 13 15 16 19 20 21 22

SE 12   14 17 18 23 24 25 26 10 13 15 16 19 20 21 22

Both the above cases have been rendered in such a method that SE 14 comes in priority 1 of them as such both can form the optimal pairs. Therefore, to resolve the above conflict it is assumed that the SE 11 lays more emphasis upon considering the switch SE 14 first as it appears before hence it is allocated to it and for switch SE 12, SE 17 comes in the next order of preference and it is compared to all other members in the preference list in which SE 12 seems to have more priority over the switch SE 17 than any other switch hence is allocated to it. Thereby the optimal pairs are as follows:

SE 11 – – – SE 14

SE 12 – – – SE 17

The same procedure can be and is followed for all such cases in case such a collision occurs and a Tie for priority of switches occurs.

3.3.7.2.3 *Deriving Optimal Pairs from the Preference Lists*

Refer procedure explained in Section (3.3.6). The same is used here to derive the optimal pairs for the QTN. See Figure (3.10) the preference lists of QTN MIN for the switch SE 1 stands as:

SE 1 19 5 23 9 15 11 17 13 14 16 18 20 21 22 24 25 26

SE 1 has highest priority been set to SE 19 as such appears first in the priority list and next priority has been set to SE 5 as such appears second and SE 23 as third thereby the rest exists as similar pattern.

As it can be seen in Figure (3.10), that SE 5 has specified as:

SE 5 23 1 19 11 9 15 14 13 16 18 20 21 22 24 25 26

Thus the priority of SE 23 is more on the of SE 5 thus both will exist as a stable matched pair and the set can be stable thus reduce the above list by eliminating the corresponding SE 23 from the list of SE 1 as someone else (SE 5) holds a better proposal for the SE 23 to follow the path and reach to its destination.

SE 1 19 5 – – – 9 15 11 17 13 14 16 18 20 21 22 24 25 26 and it becomes;

SE 1 19 5 9 15 11 17 13 14 16 18 20 21 22 24 25 26

Similarly, the nodes SE 17 and SE 13 occur higher in the priority list of switches SE 13 and SE 14 thereby the above two switches are eliminated from the list of SE 1 and similarly the final list of optimal set is:

SE 1 19 5 9 11 16 18

Thus, the final pair becomes SE 1 and SE 19, which is stable in nature. Based on this assumption and analysis the following Figure (3.11) {which shows all the optimal pairs} has been compiled.

$$
\begin{array}{l}
(1,19),\ (2,20),\ (3,21),\ (4,22),\ (5,23), \\
(6,24),\ (7,25),\ (8,26),\ (9,13),\ (11,14) \\
(15,17),\ (16,18),\ (17,15),\ (8,16),\ (10,12), \textit{and} \\
(12,10)
\end{array}
$$

Figure 3.11. The optimal pairs, which have been short–listed from the QTN preference lists.

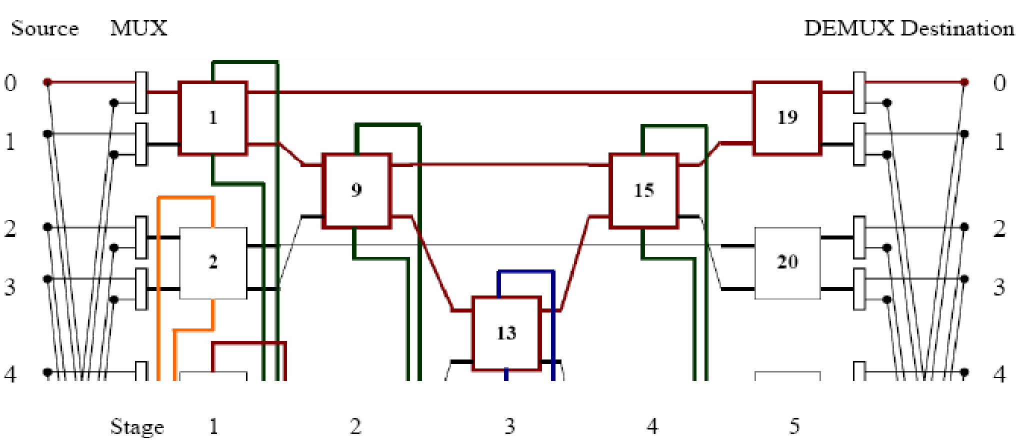

Figure 3.12. The partial cut way part of QTN.

Example 3.6. See Figure (3.12) and Table (3.2) for all possible routes and path–lengths. In this particular example a request is routed from source 0 to destination 0 i.e. source 0000 to destination 0000.

Table 3.2. The routing table of QTN.

Routes	Path–length
SE 1 – SE 19	2
SE 1 – SE 9 – SE 15 – SE 19	4
SE 1 – SE 9 – SE 13 – SE 15– SE 19	5

Explanation: In this example (Table (3.2)), all the possible paths from the source to destination are listed. To route a request from a given source to given destination can have possible route and possible path–lengths. In this particular example, there have three paths from one source to destination. The first path (SE 1 – SE 19) is termed as the primary path, whereas the path (SE 1 – SE 9 – SE 15 – SE 19) is termed as secondary path and finally the path (SE 1 – SE 9 – SE 13 – SE 15 – SE 19) is the express path used, when the primary and secondary path are busy. The respective path–lengths at all the paths mentioned are 2, 4, and 5. Since QTN is an irregular MIN, therefore the 50% of the path–lengths are passed at the path–length of 2 and rest at path–lengths of 4, and 5.

3.3.7.3 *ASEN*

It is known that the ASEN are regular networks and there is no need to give the path length algorithm, as the path length remains constant on all the routes (may be primary, secondary, or express).

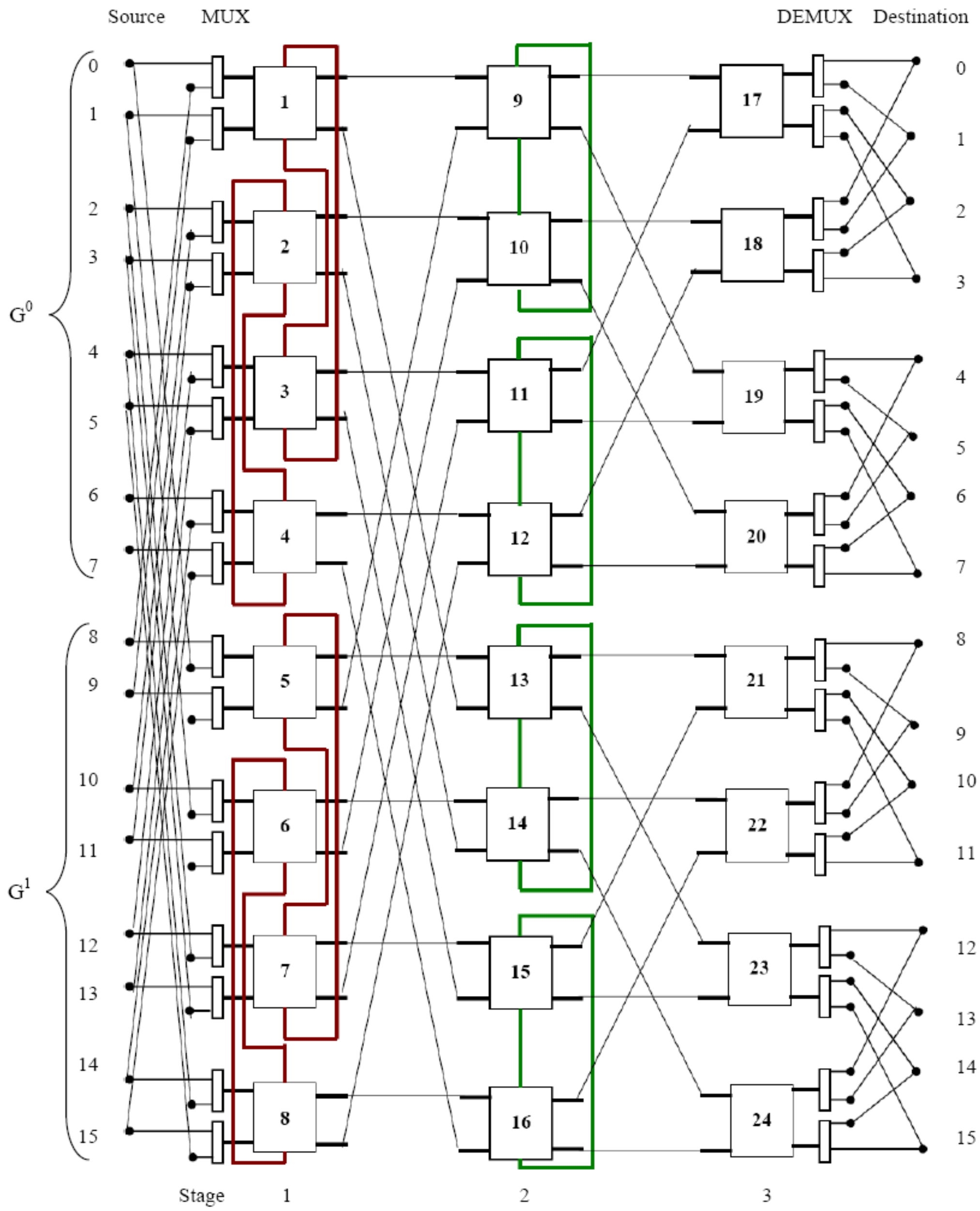

Figure 3.13. A 16 x 16 ASEN. Here SEs are renumbered to solve the stability problem.

3.3.7.3.1 *Preference Lists*

Refer algorithm explained in Section (3.3.4) for deriving preference lists for the ASEN. The SEs in Figure (3.3) are renumbered and put up again in Figure (3.13). Figure (3.14), shows the preference lists having Ties. All Ties have been solved for ASEN.

SE 1	9	13	3	11	15	17	19	21	23
SE 2	10	14	4	12	16	18	20	22	24
SE 3	11	15	1	9	13	17	19	21	23
SE 4	12	16	2	10	14	18	20	22	24
SE 5	13	9	7	15	11	21	23	17	19
SE 6	14	10	8	16	12	22	24	18	20
SE 7	15	11	5	13	9	23	21	17	19
SE 8	16	12	6	14	10	22	24	18	20
SE 9	17	19	10	18	20				
SE 10	18	20	9	17	19				
SE 11	17	19	12	18	20				
SE 12	18	20	11	17	19				
SE 13	21	23	14	22	24				
SE 14	22	24	13	21	23				
SE 15	21	23	16	22	24				
SE 16	22	24	15	21	23				

Figure 3.14. The complete preference lists of the ASEN.

3.3.7.3.2 *Reduction of the Ties*

Refer procedure explained in Section (3.3.5). The same is used here to derive the optimal pairs for the ASEN. See Figure (3.14), the preference list for the SE 10 and SE12 stands as:

SE 10   18 20 9 17 19

SE 12   18 20 11 17 19

Both the above cases have been rendered in such a method that SE 18 comes in priority 1 of them as such both can form the optimal pairs. Therefore, to resolve the above conflict it is assumed that the SE 10 lays more emphasis upon considering the switch SE 18 first as it appears before hence it is allocated to it and for switch SE 12, SE 20 comes in the next order of preference and it is compared to all other members in the preference list in which SE 12 seems to have more priority over the switch SE 20 than any other switch hence is allocated to it. Thereby the optimal pairs are as follows:

SE 10 – – – SE 18

SE 12 – – – SE 20

The same procedure can be followed for all such cases in case such a collision occurs and a Tie for priority of switches occurs.

3.3.7.3.3 *Deriving Optimal Pairs from the preference Lists*

Refer procedure explained in Section (3.3.6). The same is used here to derive the optimal pairs for the ASEN. See Figure (3.14), the preference lists of ASEN for the switch SE 1 stands as:

SE 1 9 13 3 11 15 17 19 21 23

SE 1 has highest priority been set to SE 9 as such appears first in the priority list and next priority has been set to SE 13 as such appears second and SE 3 as third and vice versa.

As it can be seen in Figure (3.14), that SE 5 has specified as:

SE 5 13 9 7 15 11 21 23 17 19

Thus the priority of SE 13 is more on the list of SE 5 thus both will exist as a stable matched pair and the set can be stable thus the above list is reduced by eliminating the corresponding SE 13 from the list of SE 1 as someone else (SE 5) holds a better proposal for the SE 13 to follow the path and reach to its destination in minimum path length.

SE 1 9 – – – 3 11 15 17 19 21 23 and it becomes;

SE 1   9 3 5 7 19 21

Similarly, the nodes SE 21 and SE 23 occur higher in the priority list of switches SE 13 and SE 11 thereby the above two switches are eliminated from the list of SE 1 and similarly the final list of optimal set is:

SE 1   9 3 5 7

Thus, the final pair becomes SE 1 and SE 9, which is stable in nature. Based on this assumption and analysis the following Figure (3.15) {which shows all the optimal pairs} has been compiled.

(1,9), (2,10), (3,11)
(4,12), (5,13), (6,14)
(7,15), (8,16), (9,17)
(10,18), (11,19), (12,20)
(13,21), (14,22), (15,23), *and*
(16,24)

Figure 3.15. The optimal pairs, which have been short–listed from the ASEN preference lists.

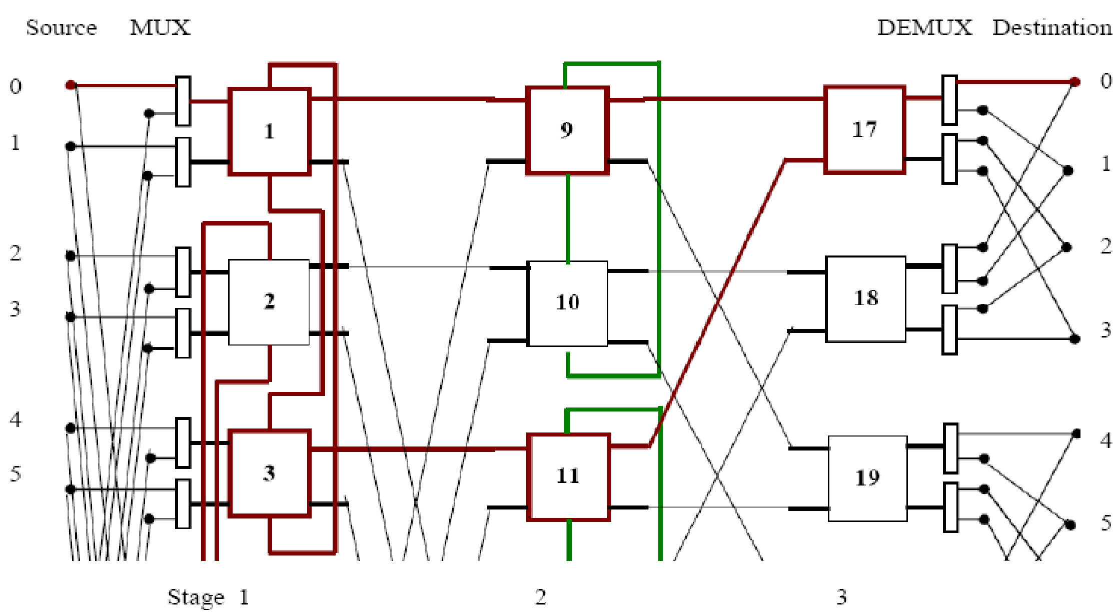

Figure 3.16. The partial cut way part of ASEN.

Example 3.7. See Figure (3.16) and Table (3.3) for all possible routes and path–lengths. In this particular example a request is routed from source 0 to destination 0 i.e. source 0000 to destination 0000.

Table 3.3. The routing table of ASEN.

Routes	Path–length
SE 1 – SE 9 – SE 17	3
SE 1 – SE 3 – SE 11 – SE 17	3

Explanation: In this example (Table (3.3)), all the possible paths from the source to destination are listed. To route a request from a given source to given destination can have possible route and possible path–lengths. In the particular example, there are two paths from one source to destination. The first path (SE 1 – SE 9 – SE 17) is termed as the primary path, whereas the path (SE 1 – SE 3 – SE 11 – SE 17) is termed as the express path used, when the primary path is busy. The respective path–lengths at all the paths mentioned are 3 only. Since ASEN is a regular MIN, therefore it is always have a constant path–length on all the routes.

3.3.7.4. *ABN*

It is know that the ASEN is regular networks and there is no need to give the path length algorithm, as the path length remains constant on all the routes (may be primary, secondary, or express).

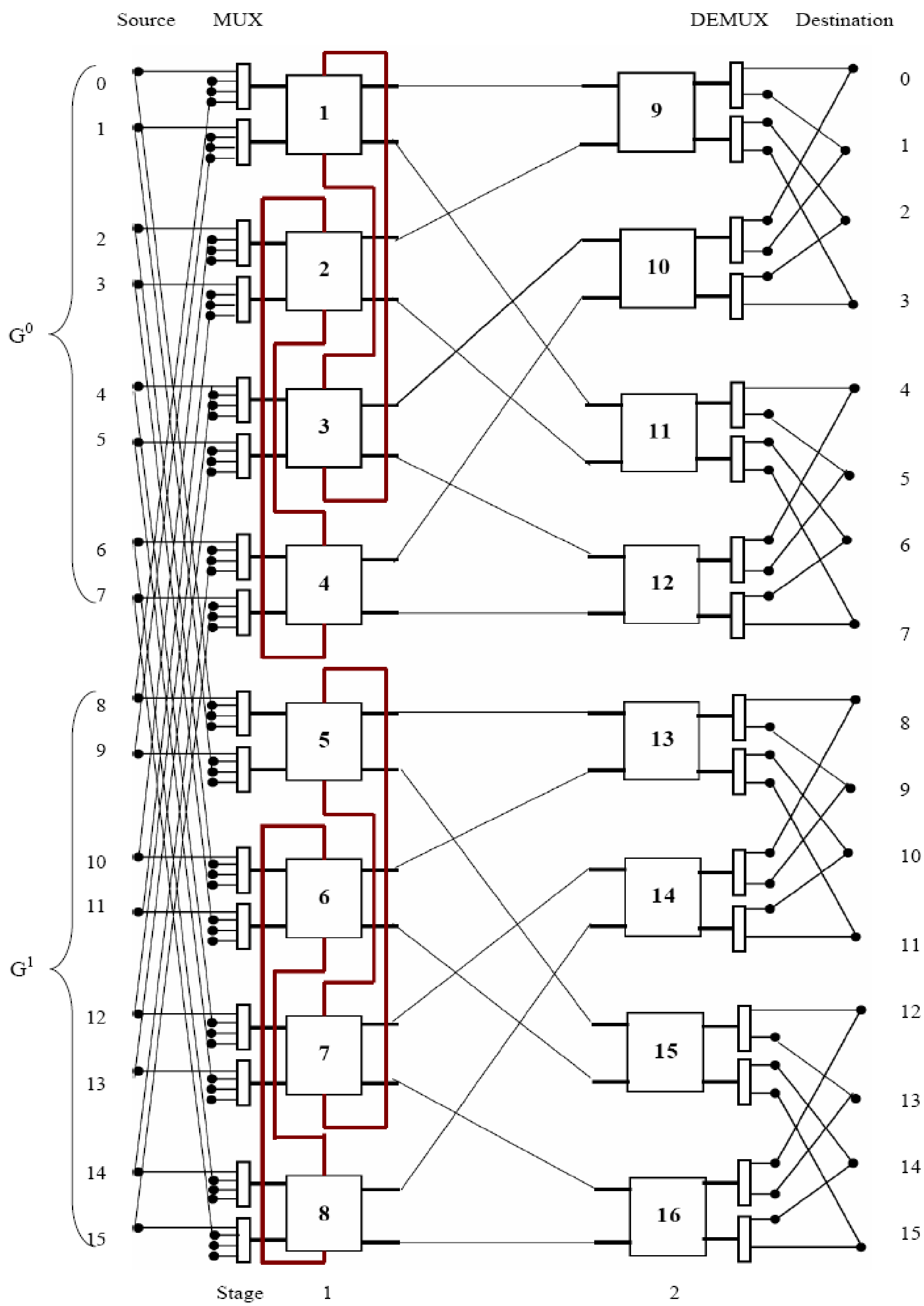

Figure 3.17. A 16 x 16 ABN. Here SEs are renumbered to solve the stability problem.

3.3.7.4.1 *Preference Lists*

Refer algorithm explained in Section (3.3.4) for deriving preference lists for the ABN. The SEs in Figure (3.4) are renumbered and put up again in Figure (3.17). Figure (3.18), shows the preference lists having Ties. All Ties have been solved for ABN.

SE 1	9	11	3	10	12	5	13	15
SE 2	9	11	4	10	12	6	13	15
SE 3	10	12	1	9	11	7	14	16
SE 4	10	12	2	9	11	8	14	16
SE 5	13	15	7	14	16	1	9	11
SE 6	13	15	8	14	16	2	9	11
SE 7	14	16	5	13	15	3	10	12
SE 8	16	14	6	13	15	4	10	12

Figure 3.18. The complete preference lists of the ABN.

3.3.7.4.2 *Reduction of the Ties*

Refer procedure explained in Section (3.3.5). The same is used here to derive the optimal pairs for the ABN. See Figure (3.18) the preference list for the SE 5 and SE 6 stands as:

SE 5   13 15 7 14 16 1 9 11

SE 6   13 15 8 14 16 2 9 11

Both the above cases have been rendered in such a method that SE 13 comes in priority 1 of them as such both can form the optimal pairs. Therefore to resolve the above conflict it is assumed that the SE 5 lays more emphasis upon considering the switch SE 13 first as it appears before hence it is allocated to it and for switch SE 6, SE 15 comes in the next order of preference and it is compared to all other members in the preference list in which SE 6 seems to have more priority over the switch SE 15 than any other switch hence is allocated to it. Thereby the optimal pairs are as follows:

SE 5 – – – SE 13

SE 6 – – – SE 15

The same procedure can be and is followed for all such cases in case such a collision occurs and a Tie for priority of switches occurs.

3.3.7.4.3 *Deriving Optimal Pairs from the Preference Lists*

Refer procedure explained in Section (3.3.6). The same is used here to derive the optimal pairs for the ABN. See Figure (3.18), the preference lists of ABN for the switch SE 1 stands as:

SE 1   9 11 3 10 12 5 13 15

SE 1 has highest priority been set to SE 9 as such appears first in the priority list and next priority has been set to SE 11 as such appears second and SE 3 as third and vice versa.

As it can be seen in Figure (3.18), that SE 3 has specified as:

SE 3   10 12 1 9 11 7 14 16

Thus the priority of SE 10 is more on the list of SE 3 thus both will exist as a stable matched pair and the set can be stable thus the above list reduced by eliminating the corresponding SE 10 from the list of SE 1 as someone else (SE 3) holds a better proposal for the SE 10 to follow the path and reach to its destination in minimum path length.

SE 1   9 11 3 – – – 12 5 13 15 and it becomes;

SE 1   9 11 3 12 5 13 15

Similarly, the nodes SE 11 and SE 12 occur higher in the priority list of switches SE 1 and SE 4 thereby the above two switches are eliminated from the list of SE 1 and similarly the final list of optimal set is:

SE 1   9 3 5 13 15 7

Thus, the final pair becomes SE 1 and SE 9, which is stable in nature. Based on this assumption and analysis the following Figure (3.19) {which shows all the optimal pairs} has been compiled.

$$(1,9),\ (2,11),$$
$$(3,10),\ (4,12),$$
$$(5,13),\ (6,15)$$
$$(7,14),\ \textit{and}\ (8,16)$$

Figure 3.19. The optimal pairs, which have been short–listed from the ABN preference lists.

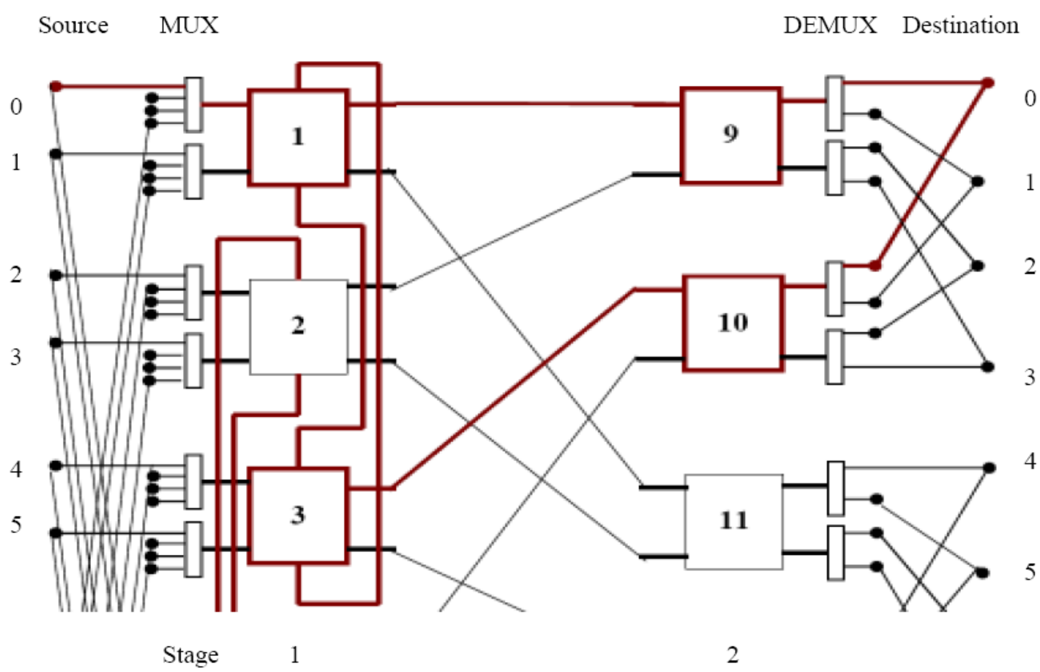

Figure 3.20. The partial cut way part of ABN.

Example 3.8. See Figure (3.20) and Table (3.4) for all possible routes and path–lengths. In this particular example a request is routed from source 0 to destination 0 i.e. source 0000 to destination 0000.

Table 3.4. The routing table of ABN.

Routes	Path–length
SE 1 – SE 9	2
SE 1 – SE 3 – SE 10	2

Explanation: In this example (Table (3.4), all the possible paths from the source to destination are listed. To route a request from a given source to given destination can have possible routes and possible path–lengths. In the particular example, there are two paths from one source to destination. The first path (SE 1 – SE 9) is termed as the primary path, whereas the path (SE 1 – SE 3 – SE 10) is termed as the express path used, when the primary path is busy. The respective path–length at all the paths mentioned is 2 only. Since ABN is a regular MIN, therefore it is always have a constant path–length on all the routes.

3.3.8 *Comparisons*

Table 3.5. The comparison of different MINs based on their stability.

MINs	Number of Ties	Number of Optimal Pairs/Total Number of Switching Elements	Maximum Path–Length	Neglected Pairs	Status of the MIN
HZTN	4	16/28	5	4	Low Stable
QTN	6	16/26	5	2	Intermediate Stable
ASEN	4	16/24	3	0	Highly Stable
ABN	3	8/16	2	0	Highly Stable

Based on the analysis of Sections (3.1–3.7) the comparison chart have been made and shown in Table (3.5) and Figure (3.21). From there it is depicted that the regular MINs are highly stable in comparison to the irregular MINs as the neglected pairs (those who are not able to find any stable match) are 0 in the case of regular ASEN and ABN. Therefore, regular MINs are highly stable according to the stable matching algorithm. On the other hand, the neglected pairs in case of QTN are low in comparison to the HZTN. Moreover, their neglected pairs are low in comparison to regular MINs.

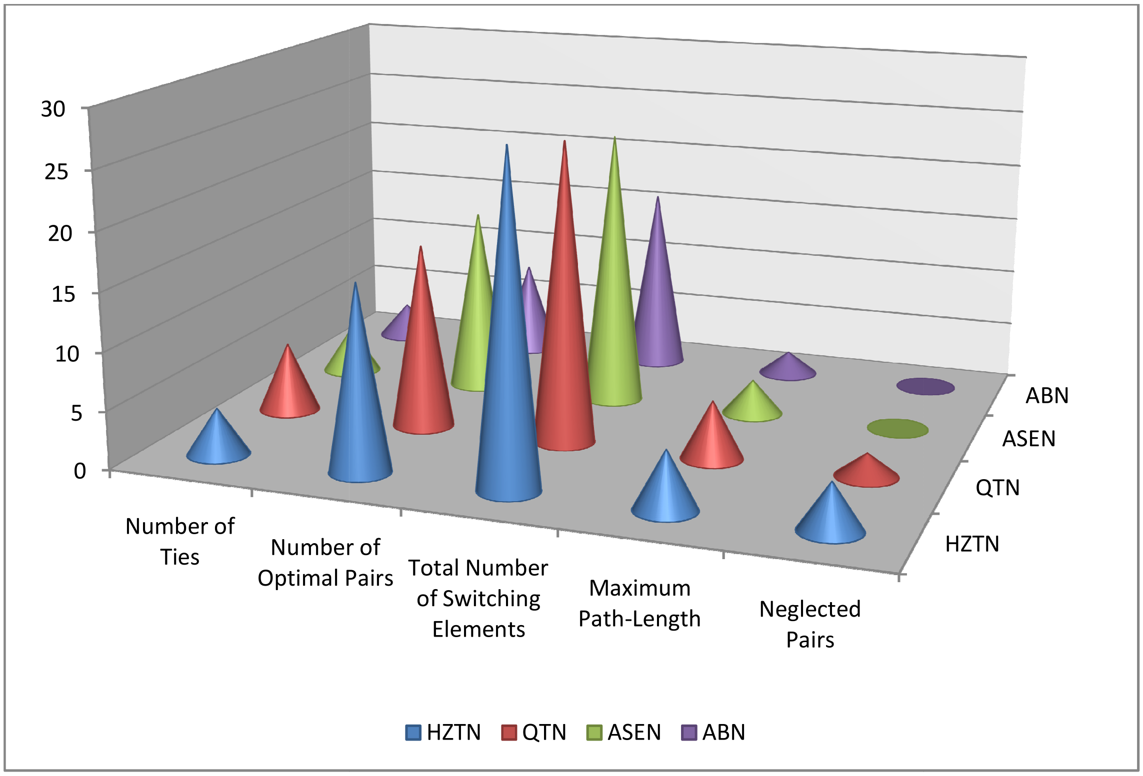

Figure 3.21. The comparison graph of different MINs based on their stability.

3.4 *Results and Discussions*

This chapter explores the relationship between stable matching and MINs stability problem. Specifically stable marriage problem is used as example of stable matching to solve the MINs stability problem. The situations in which the fault–tolerant irregular and regular MINs become unstable have been shown. To counter this problem the appropriate algorithm, procedures, and methods have been designed using the concept of stable marriage. The ties problem of the optimal pairs has been solved. The comparison of the MINs based upon their stability shows that the ASEN and ABN are more or less equally stable as the neglected pairs in both the cases are equal. On the other hand, irregular MINs based upon their stability shows that the QTN is more stable as the neglected pairs in his case are low in comparison to the HZTN. Therefore, QTN is more stable than the HZTN. However, on comparing the irregular and regular MINs in totality upon their stability the regular MINs comes out to be more stable than the irregular MINs.

CHAPTER 5

STOCHASTIC COMMUNICATION FOR NoC

5.1 *Introduction and Motivation*

Systems–on–chip (SoC) designs provide integrated solutions to challenging design problems in the telecommunications, multimedia, and consumer electronic domains. With Deep Sub–Micron (DSM) technology, chip designers are expected to create SoC solutions by connecting different Intellectual Property (IP) using efficient and reliable interconnection schemes known as Networks–on–Chip (NoC). This technique makes a clear distinction between computation (the tasks performed by the IPs) and communication (the interconnecting architecture between the IPs). A NoC is formed by connecting either homogeneous or heterogeneous IPs on a single chip. Since modern NoCs are becoming extremely complex, so there are many challenges in this new area of research.

On–chip wire delays have become more critical than gate delays and recently synchronization problems between IP are more apparent. This trend only worsens as the clock frequencies increase and the feature sizes decrease [25]. However, the problem of high latency, remain an open question and important factor in real time applications [53]. During parallel communication using IoC it subject to new types of malfunctions and failures that are harder to predict and avoid with the current NiP and SoC design techniques.

These new types of failures are impossible to characterize using deterministic measurements. Therefore, in the near future, probabilistic metrics, such as average values and variance will be needed to quantify the critical design objectives such as performance and power. In NoCs and NiP, the IP communicates using probabilistic broadcast scheme called On–chip stochastic communication. This scheme achieves many of the desire features of the future NoC and provides [30]:

1. *Separation between computation and communication.*
2. *Fault– tolerance.*

Despite of these features, low latency is major challenge in modern NoCs. Latency in NoC can be measure by calculating the latency in switch and propagation delay in IoC [54] but it depends on the type of NoC i.e. single NoC or multiple NoC (also known as Networks–in–Package (NiP)). The different NoC topologies are already used in [51] provides different communication structure in NoC [52].

During the parallel communication between the IPs and among the NoCs with increasing number of PEs, the loss of data packets are inevitable. Therefore, one should aware about loss of data packets during the communication and this exact information is only be provided by the Stochastic Communication. This scheme can be widely used to achieve high adaptivity and support inter–processor communications in parallel computers due to its ability to preserve both communication performance and fault–tolerant demands in such networks.

From starting year 2003, the researchers have already done lots of excellent work in developing, stochastic communication for On–chip or inter (local or communication within the IPs) NoC communication and are reported in [30–32]. However, the comprehensive studies have been reported in [31] and the major drawbacks of these systems are as below:

1. *The communication is only for the broadcast scheme.*
2. *The prediction of the latency is calculated within the groups.*
3. *Probabilistic data tracking is available within the group.*
4. *Designing prospects are worst.*
5. *Stochastic communication technique is available only for inter NoC communication.*
6. *Presence of Livelock.*
7. *Presence of Deadlock.*
8. *Network Unstability.*

The stochastic communication for intra (global or among different NoCs in NiP) NoC communication was ignored and remained out of lime light. To overcome all the previous problems, the interest has been developed for stochastic communication for intra NoC communication and hence, this chapter, proposes an analytical technique for stochastic communication that is suitable for homogeneous as well as heterogeneous NoC. It not only separates the computation and communication in NiP module but also predicts the communication performance. The communication structure in NiP is divided into two zones, one is On–chip (inter or local) communication and other is On–package (intra or global) communication.

In order set up stochastic communication in local and global zone, a compartmental-based stochastic communication technique is used for Application–Specific NoCs, where different IPs is used. These IPs are treated as compartmental IPs. Moreover, a Closed Donor Controlled Based Compartmental Model can represent the flow of data from source IP to Destination IP. From this model, the compartmental matrix is derived that retains the properties of Metzler matrix. The derived compartmental matrix gives us the inter–compartmental flow of IPs that helps us to calculate the transition probability matrix and hence the resultant matrix can be converted into Markov Chain. While handling IP based compartmental models, some models are having feedback and some did not. The models having feedback can be converted into stochastic models using regular markov chains and the others using Absorbing Markov Chains. If the compartmental model is linear then the stochastic model can be easily generated, otherwise, it has to be linearized using Jacobian matrix about the equilibrium points.

The rest of the work is as follows: Section 5.2 discusses the data flow networks in NoC followed by the analysis of compartmental and stochastic modeling of the data flow networks in Section 5.3 and Section 5.4. The stochastic analysis of the NoC communication is discussed in Section 5.5 followed by the simulation and results. Not only the stochastic communication technique proposed for NoC but also for NiP on the similar lines and discussed as a case study in Section 5.7.

5.2 *Data Flow Network in NoC for Stochastic Communication*

The compartmental–based probabilistic data broadcasting, among the IPs is suggested here. This broadcasting is a random process scheme, similar to the randomized gossip protocols [30]. When the data in the form of packets, transmitted from source to destination IP in the grid based square network as shown in Figure (5.1), then IP communicates the data using proposed probabilistic broadcasting scheme. The source IP sends the data packets to the destination IP through its neighbors. It is already known that in homogeneous or heterogeneous NoC, any IP can be used as the source IP or intermediate IP or destination IP. The data can flow, depending upon the requirement in many possible ways. Here, one of the data flow network in Application–Specific NoC is used and that consist of few IP and routers as shown in Figure (5.2).

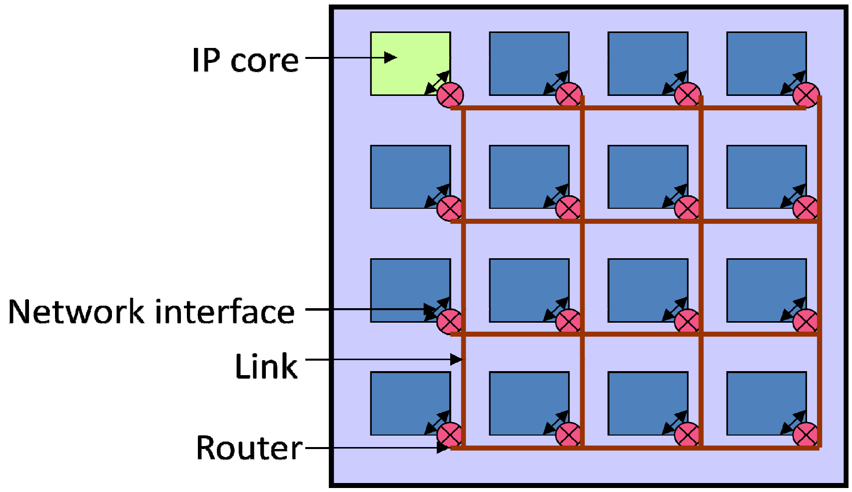

Figure 5.1. Topological illustration of a 4–by–4 grid structured homogeneous NoC.

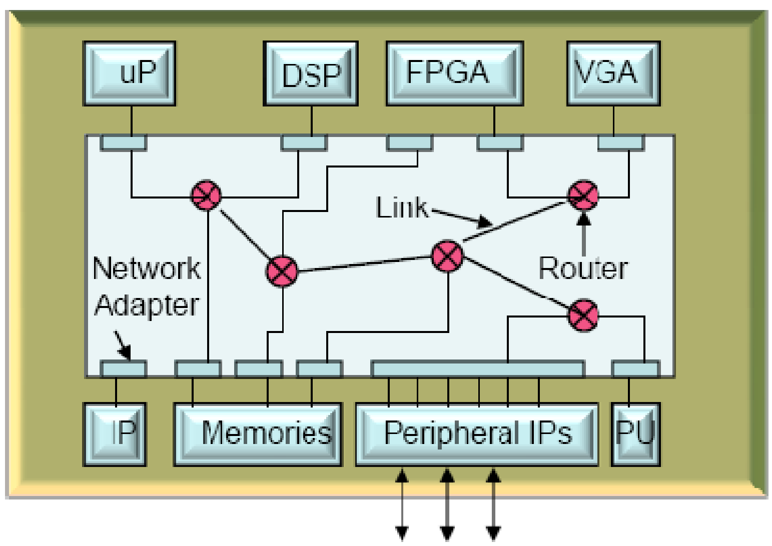

Figure 5.2. Application–Specific NoC.

In Figure (5.2), the Network Adapter or Interface implements the interface by which IPs connect to the NoC. Their function is to decouple computation (the IPs) from communication (the network). Router routes the data according to chosen protocols and implements the routing strategy. Links connect the nodes, providing the raw bandwidth and may consist of one or more logical or physical channels [52].

Consider Figure (5.2) and the following scenario for the inter NoC stochastic communication (more specifically IP to IP communication): If the data has to be sent from Digital Signal Processor (DSP) to Field–programmable Gate Array (FPGA) and Processing Unit (PU) then one of the data flow network from NoC can be extract. There are five compartments in data flow network as shown in Figure (5.3). These compartments are as follows:

1. *Source IP is represented using* (X_1),
2. *Intermediate IP is represented using* $(X_2 \text{ and } X_3)$ *and*
3. *Destination IPs are represented using* $(X_4 \text{ and } X_5)$.

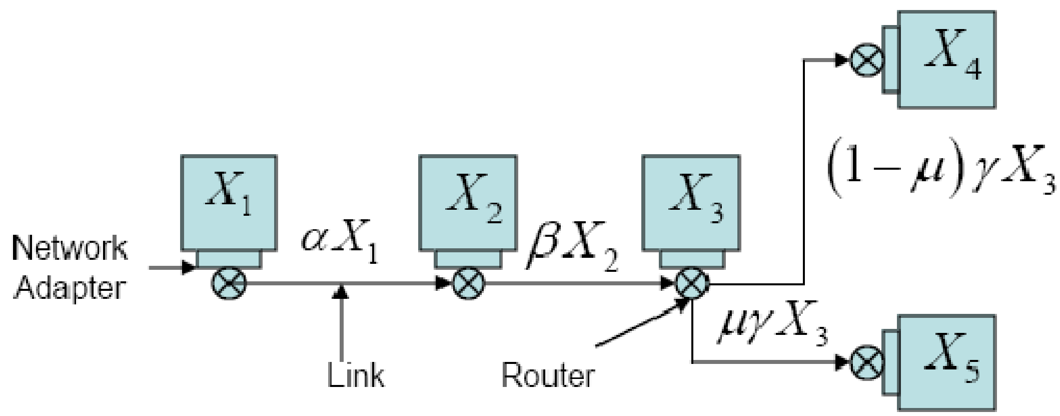

Figure 5.3. Data flow network for stochastic communication.

This model of data flow network is also known as stochastic network and can be used for stochastic modeling by following certain assumptions:

1. *The total number of data packets is constant.*
2. *The model is closed donor controlled based model.*
3. *The model is mass conservative.*

The behavior of data flow model, shown in Figure (5.3), can be described by the following set of differential equations:

$$\frac{dX_1}{dt} = -\alpha X_1 \qquad \text{.....(5.1)}$$

PARTIAL

CHAPTER 6

CONCLUSION

This chapter sums up the entire study and list out further scope of work along with the open problems concern with stable matching and stochastic communication.

6.1 *Conclusion*

The thesis explores the relationship between Stable Matching and MINs stability problems. Specifically Stable Marriage problem is use as an example of stable matching to solve the MINs stability problem. The situation in which the fault–tolerant irregular and regular MINs become unstable is presented. Appropriate algorithms are designed on the concept of stable marriage to prove MINs stable. The Ties problem of the optimal pairs is also solved.

The results are very promising as the comparison (Table (3.5) and Figure (3.21)) of the MINs based on their stability, shows that the fault–tolerant regular MINs are highly stable in comparison to the fault–tolerant irregular MINs because the number of neglected pairs (pairs those who are not able to find any stable match) are 0 in the case for regular ASEN and ABN contrary to 0 in case of HZTN and QTN. When irregular and regular MINs are compared on other aspects such as Number of Ties, Ratio of Number of Optimal Pairs to Total Number of Switching Elements and Maximum Path–Length, it is seen that the regular MINs are more stable as compared to irregular MINs.

The typical modern day application of the MINs includes fault–tolerant packet switches, designing multicast, broadcast router fabrics while SoC and NoC are still in vogue. This thesis for the first time proposes a new method that takes care of global communication between NoCs in NiP. Therefore, the focus shifts from stability of MINs application as IoC in NiP to provide global communication between NoCs. This method sets up intra (global) communication between Application–Specific (heterogeneous or homogeneous) NoCs in NiP. For this, twin $O(n^2)$ fault–tolerant parallel algorithms are proposed. These algorithms allow different NoCs to communicate in parallel using either fault–tolerant irregular PNN or fault–tolerant regular HXN. These two are acting as IoC in NiP and can tolerate faults also.

Using Tables (4.11–4.13) and Tables (4.24–4.26), the results are compare for the NiP using HXN and PNN as IoC has shown 100% efficiency for the single and double pair of communication among the NoCs. However, as the number of pair increases i.e. for quad pair of communication, the NiP using PNN as IoC has shown 75% efficiency whereas the same NiP using HXN as IoC has shown 62.5% efficiency only. The decrease in the acceptance ratio of the packet at the destination NoC is just because of high level of parallelism occurring among the NoC. The NiP architecture as presented provides intra communication between four NoCs only; however, this architecture can be extended upon the requirements. The scalability of the system also depends upon the size of IoC. As, when needed more number of NoCs can be added to NiP architecture however, the size of the IoC that is currently 2 x 2 should be scale to 3 x 3, 4 x 4 etc. according to the requirements. The comparison of IoCs on Cost and MTTR concludes that the HXN has the higher hardware cost than the PNN but the MTTR values of the HXN are low in comparison to the PNN. This signifies that as the size increases the ability to tolerate faults and online repairing of the HXN become higher and faster than the PNN.

The NoC solution brings a networking method to On–chip communication and reports threefold increase in performance over conventional bus systems. A new technique that merges two communications i.e. inter (local) communication (between the IPs in a NoC), and intra (global) communication between NoCs in NiP is propose. This can be achieve through Stochastic Communication between the different IPs, building the compartmental model of IPs, present on the NoC and calculating the latency as well as the transition probabilities of data flow between any two IP. Figures (5.6–5.8 and 5.12–5.15) and Tables (5.1–5.3 and 5.4–5.15), shows that the transient and steady state response of transition probabilities gives us the state of data flow latencies among the different IPs in a NoC and the compartments in NiP. From Figure (5.16), it is seen that the latency of NiP is lower in comparison to the ring and bus topology but it is higher than star topology. Moreover, the latency of inter IP communication in NoC and communication among NoCs in NiP is very low as compared to other topologies. Therefore, the stochastic communication techniques propose in this chapter has produce low latency over any other existing topologies use for setting up inter NoC and NiP communication.

6.2 *Future Scope*

The algorithms and techniques propose and presented in this thesis are of universal nature to solve the stability problems of any class of MINs. The NP–Completeness along with ties problem of the optimal (stable) pairs may be explored further. The modern day application of fault–tolerant MINs in NiP and SoCs are inevitable therefore by using them as an IoC one can designed more faster program construct as the current run time complexity of the twin parallel algorithm is $O(n^2)$ and the same can be reduced to the size of the $O(n \log n)$ or even lesser than that. The loss of data packets during the parallel communication (on large scale) in NiP and SoCs is a performance concern. This problem can be avoided through stochastic communication technique and to improve further with the identification of controllability and absorbability for each NoC as well as IoC. In turn, it will help to design GALS communication.

REFERENCES

[1] Nitin, A. Subramanian, Efficient algorithms to solve dynamic MINs stability problems using stable matching with complete ties, Journal of Discrete Algorithms 6 (3) (September 2008) 353–380, Elsevier.

[2] Nitin, V.K. Sehgal, On path–length and routing–tag algorithm for hybrid irregular multi–stage interconnection networks, IEEE Computer Society Press, July 30–August 1 2007, pp. 652–657.

[3] Nitin, V.K. Sehgal, P.K. Bansal, On MTTF analysis of a fault–tolerant hybrid MINs, WSEAS Transactions on Computer Research 2 (2) (February 2007) 130–138, ISSN 1991–8755.

[4] Nitin, Component level reliability analysis of fault–tolerant hybrid MINs, WSEAS Transactions on Computers 5 (9) (September 2006) 1851–1859, ISSN 1109–2750.

[5] Nitin, Reliability analysis of multi–path multi–stage interconnection network, in: Proceedings of the 10th WSEAS International Conference on Circuits, Systems, Communication and Computers, July 2006, pp. 1018–1023.

[6] Nitin, On analytic bounds of regular and irregular fault–tolerant multi–stage interconnection networks, in: Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications, June 2006, pp. 221–226.

[7] Nitin, A. Subramanian, On reliability analysis of cost–effective hybrid zeta network: a fault–tolerant multi–stage interconnection network, in: Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications, June 2006, pp. 260–265.

[8] A. Subramanian, Nitin, On a performance of multistage interconnection network, in: Proceedings of the 12th International Conference on Advanced Computing and Communication, December 2004, pp. 73–79.

[9] Nitin, D.S. Chauhan, V.K. Sehgal, Two O(n^2) time fault–tolerant parallel algorithm for Inter NoC communication in NiP, Springer–Verlag (May 2008) 267–282, ISBN 978–3–540–79186–7, Invited.

[10] Nitin, V.K. Sehgal, Stochastic communication on application specific networks on chip, Springer–Verlag (September 2007) 11–16, ISBN 978–1–4020–6265–0.

[11] T.Y. Feng, A survey of interconnection networks, IEEE Computer 14 (12) (December 1981) 12–27.

[12] K. Hwang, F.A. Briggs, Computer Architecture and Parallel Processing, McGraw–Hill, New York, 1984, ISBN 0–07–066354–8.

[13] G.B. Adams III, D.P. Agrawal, H.J. Siegel, A survey and comparison of fault–tolerant multi–stage interconnection networks, IEEE Computer 20 (6) (June 1987) 14–27.

[14] Special issue on interconnection networks, IEEE Computer 20 (6) (June 1987).

[15] H.J. Siegel, Interconnection Network for Large Scale Parallel Processing: Theory and Case Studies, McGraw Hill, 1990, ISBN 0–07–057561–4.

[16] K. Hwang, Advanced Computer Architecture: Parallelism, Scalability, Programmability, Tata McGraw–Hill, India, 2000, ISBN 0–07–053070–X.

[17] J. Duato, S. Yalamanchili, L.M. Ni, Interconnection Networks: An Engineering Approach, Morgan Kaufmann, San Francisco, CA, 2003, ISBN 1–55860–852–4.

[18] W. Dally, B. Towles, Principles and Practices of Interconnection Networks, Morgan Kaufmann, San Francisco, CA, 2004, ISBN 978–0–12–200751–4.

[19] Nitin, On a fault–tolerant hybrid ZETA MIN, Masters Thesis, Computer Science and Engineering Department, Thapar Institute of Engineering and Technology, Patiala, Punjab, December 2002.

[20] D. Gale, L.S. Shapley, College admissions and the stability of marriage, The American Mathematical Monthly 69 (1962) 9–15.

[21] D. Gusfield, R.W. Irving, The Stable Marriage Problem: Structure and Algorithms, The MIT Press, Cambridge, MA, 1989.

[22] H. Mairson, The stable marriage problem, The Brandeis Review 12 (1992) (online).

[23] M. Sgroi, M. Sheets, A. Mihal, K. Keutzer, S. Malik, J. Rabaey, A.S. Vincentelli, Addressing the system–on–a–chip interconnect woes through communication–based design, in: Proceedings of the 38th Annual ACM IEEE Conference on Design Automation Conference, (2001), pp. 667–672.

[24] L. Benini, G. De Micheli, Networks on chips: a new SoC paradigm, IEEE Computers, 35 (1) (January 2002) 70–78.

[25] L. Kangmin, L. Se–Joong, K. Donghyun, K. Kwanho, K. Gawon, K. Joungho, Y. Hoi–Jun, Networks–on–chip and networks–in–package for high–performance SoC platforms, in : Proceedings of International Conference on Asian Solid–State Circuits, November 2005, pp. 485–488.

[26] L. Benini, G. De Micheli, Networks on Chips: Technology and Tools, Morgan Kaufmann, San Francisco, CA, 2006, ISBN 978–0–12–370521–1.

[27] D.N. Jayasimha, B. Zafar, Y. Hoskote, On–chip interconnection networks: why they are different and how to compare them, Intel Corporation (2006), pp. 1–11.

[28] B. Inwood, Goal and target in stoicism, The Journal of Philosophy 83 (10) (October 1986) 547–556.

[29] S. Asmussen, P.W. Glynn, Stochastic Simulation: Algorithms and Analysis, Springer, 2007, ISBN 978–0–387–30679–7.

[30] T.A. Dumitras, On–chip stochastic communication, Masters Thesis, Department of Electrical and Computer Engineering Carnegie Mellon University, Pittsburgh, Pennsylvania, May 2003.

[31] P. Bogdan, R. Marculescu, A theoretical framework for on–chip stochastic communication analysis, in: Proceedings of 1st International Conference on Nano–Networks, September 2006, pp. 1–5.

[32] P. Bogdan, T.A. Dumitras, R. Marculescu, Stochastic communication: a new paradigm for fault-tolerant networks–on–chip, VLSI Design, Hindawi Publishing Corporation, 2007 (95348) (February 2007), DOI: 10.1155/2007/95348.

[33] R.W. Irving, An efficient algorithm for the stable roommates problem, Journal of Algorithms 6 (1985) 577–595.

[34] G. Polya, R.E. Tarjan, D.R. Woods, Notes on Introductory Combinatorics, Birkhauser Verlag, Boston, MA, 1983.

[35] E.W. Mayr, A. Subramanian, The complexity of circuit value and network stability, in: Proceedings of 4th Annual Conference on Structure in Complexity Theory, 1989, pp. 114–123.

[36] A. Subramanian, A new approach to stable matching problems, SIAM Journal of Computing 23 (4) (August 1994) 671–700.

[37] A. Subramanian, The computational complexity of the circuit value and network stability problems, Doctoral Thesis, Department of Computer Science, Stanford University, Stanford, 1990.

[38] D. Gusfield, Three fast algorithms for four problems in stable marriage, SIAM Journal of Computing 16 (1) (February 1987) 111–128.

[39] R.W. Irving, P. Leather, D. Gusfield, An efficient algorithm for the optimal stable marriage, Journal of Association of Computing Machinery 34 (July 1987) 532–543.

[40] T. Feder, A new fixed point approach for stable networks and stable marriages, Journal of Computer and System Sciences 45 (October 1992) 233–284.

[41] P.K. Bansal, K. Singh, R.C. Joshi, Routing and path length algorithm for a cost–effective four–tree multi–stage interconnection network, International Journal of Electronics 73 (1) (1992) 107–115.

[42] V.P. Kumar, S.M. Reddy, Design and analysis of fault–tolerant multistage interconnection networks with low link complexity, in: Proceedings of 12th International Symposium on Computer Architecture, June 1985, pp. 376–386.

[43] V.P. Kumar, S.M. Reddy, Augmented shuffle–exchange multistage interconnection networks, IEEE Computer (June 1987) 30–40.

[44] P.K. Bansal, R.C. Joshi, K. Singh, On a fault–tolerant multistage interconnection network, Journal of Computers and Electrical Engineering, Elsevier Sciences 20 (4) (July 1994) 335–345.

[45] S.S. Tseng, R.C.T. Lee, A parallel algorithm to solve stable marriage problem, BIT 24 (1984) 308–316.

[46] H.J. Siegel, C.B. Stunkel, Inside parallel computers: trends in interconnection networks, IEEE Computational Science and Engineering 3 (3) (Fall 1996) 69–71, Invited.

[47] D. Tabak, Multiprocessors, Prentice Hall, Englewood Cliffs, NJ, 1990, ISBN 9780136052470.

[48] K.N. Levitt, M.W. Green, J. Goldberg, A study of the data commutation problems in a self–repairable multiprocessor, in: Proceedings of Spring Joint Computer Conference, Book Company, 1968, pp. 515–156.

[49] D.H. Lawrie, Access and alignment of data in an array processor, IEEE Transactions on Computers C–24 (December 1975) 1145–1155.

[50] J.H. Patel, Performance of processor–memory interconnection for multiprocessors, IEEE Transactions on Computers 30 (October 1981) 771–780.

[51] S. Murali, G. De Micheli, SUNMAP: A tool for automatic topology selection and generation for NoCs, in: Proceedings of Design Automation Conference, June 2004, pp. 914–919.

[52] T. Bjerregaard, S. Mahadevan, A survey of research and practices of network–on–chip, ACM Computing Surveys 38 (1) (March 2006) 1–51.

[53] L. Kangmin, L. Se–Joong, Y. Hoi–Jun, Low–power network–on–chip for high–performance SoC design, IEEE Transactions on Very Large Scale Integration Systems 14 (2) (February 2006) 148–160.

[54] K. Kwanho, L. Se–Joong, L. Kangmin, Y. Hoi–Jun, An arbitration look–ahead scheme for reducing end–to–end latency in networks on chip, in: Proceedings of IEEE International Symposium on Circuits and Systems, May 2005, pp. 2357–2360.

[55] G.G. Walter, M. Contreras, Compartmental Modeling with Networks, Birkhauser Verlag, Boston, NY, 1999, ISBN 0–8176–4019–3.