DESIGN, VALIDATION AND FPGA IMPLEMENTATION OF MULTISTAGE TELECOMMUNICATION NETWORKS IN HDL ENVIRONMENT

By

ADESH KUMAR

(SAP ID: 500012583)

COLLEGE OF ENGINEERING

Under the Guidance of

Dr. Piyush Kuchhal

(Associate Professor, College of Engineering, UPES, India)

Dr. Sonal Singhal

(Assistant Professor, Department of Electrical Engineering,
Shiv Nadar University, NCR, G.B Nagar, India)

Submitted

IN PARTIAL FULFILLMENT OF THE REQUIREMENT OF THE DEGREE OF DOCTOR OF PHILOSOPHY

TO

UNIVERSITY OF PETROLEUM AND ENERGY STUDIES

DEHRADUN, INDIA

February, 2014

THESIS COMPLETION CERTIFICATE

This is to certify that the thesis on "**Design, Validation and FPGA implementation of Multistage Telecommunication Network in HDL Environment"** by **Adesh Kumar** in Partial completion of the requirements for the award of the Degree of Doctor of Philosophy (Engineering) is an original work carried out by him under our joint supervision and guidance. It is certified that the work has not been submitted anywhere else for the award of any other diploma or degree of this or any other University.

External Guide

Dr. Sonal Singhal

Internal Guide

Dr. Piyush Kuchhal

Design, Validation and FPGA Implementation of Multistage Telecommunication Networks in HDL Environment

By

Adesh Kumar

Dedicated to

My Parents: Father, Shri Rajendra Kumar, Mother Late. Smt. Rajesh Devi for their blessings and Maa Saraswati, who has given us the divine gift of education

ACKNOWLEDGMENTS

Working on the Ph.D. has been a wonderful and often overwhelming experience. It is hard to say whether it has been grappling with the topic itself which has been the real learning experience, or grappling with how to write papers and proposals, give talks, work in a group, stay up until the birds start singing, and stay focus. In any case, I am indebted to many people for making the time working on my Ph.D. an unforgettable experience.

I am deeply grateful to the honorable Chancellor Dr. S.J Chopra and Vice chancellor Dr. Parag Diwan, Dr. Shri Hari Campus Director, Dr. Kamal Bansal, Director, College of Engineering, to permit me research work in the R & D house in VLSI Design, UPES Dehradun. To work with my guide Dr. Piyush Kuchhal, Associate Professor, CoES, UPES and Dr. Sonal Singhal, Assistant Professor, Shiv Nadar University, NCR, India, has been a real pleasure to me, with heaps of fun and excitement. They have been a steady influence throughout my Ph.D. career that have oriented and supported me with promptness and care, and have always been patient and encouraging in times of new ideas and difficulties.

In addition, I have been very privileged to get to know and to collaborate with many other great people who became friends over the last several years. I learned a lot from them about life, research, how to tackle new problems and how to develop techniques to solve them. From the very beginning of my Ph.D. career, my guides supported me. Thank you for the fun and the encouraging discussions while meeting with you at the University of Petroleum & Energy Studies, Dehradun India.

ADESH KUMAR

CURRICULAM VITAE

ADESH KUMAR **Email:** adeshmanav@ gmail.com

PROFESSIONAL QUALIFICATION

- **Ph.D(Pursuning)** from **University of Petroleum and Energy Studies, Dehradun, India** since **June 2010.**
- **M.Tech (Hons) (Embedded Systems Technology/ ECE)** from **SRM University** Campus, **Chennai** with **9.197 CGPA** in **2008.**
- **B.Tech. (Electronics & Communication Engineering)** from **Veera Engineering College, Bijnor (U.P)** which is affiliated to **U.P. Technical University, Lucknow** with **70.74%** in **2006.**

ACADEMIC QUALIFICATION

- Intermediate **(12th)** with **PCM group** from J.L.N.S. Inter College Satheri, MuzaffarNagar **(U.P. Board)** with **67.00%** in **2002.**
- High School **(10th)** with **Mathematics and Science** group from J.L.N.S. Inter College Satheri, MuzaffarNagar **(U.P. Board)** with **63.16%** in **2000.**

INDUSTRIAL + ACADEMIC EXPERIENCE

- Currently working with University **of Petroleum and Energy Studies, Dehradun as Assistant Professor and Ph. D Scholar** in the department of Electronics and Instrumentation Engineering since June 2010.
- Worked with ICFAI University, Dehradun as Faculty Memeber in "Faculty of Science & Technology" from Dec 2009 to May 2010.
- Worked for TATA ELXSI LIMITED, Bangalore as Senior Engineer in Design and Development for Semicon and Systems group from Nov. 2008 to Oct 2009.

PROJECTS

Projects were completed in **Science and Technology Entrepreneurship Park (STEP), IIT Roorkee** and **CIPL, Roorkee** and are submitted in **Veera College of Engineering, Bijnor** during **B.Tech final year**.

- **Major Project:** Design & Simulation of **16 bits Microprocessor** of my own configuration Using **VHDL**.
- **Minor project:** Design & Simulation of **Traffic Light Controller** IC Using **VHDL**.

M.Tech Thesis was submitted to SRM University, Chennai in M.Tech final Year.

- **Thesis Title:** Simulation & Design of **Time Division Multiplexing (TDM) switch IC for landline and mobile communication** using HDL.

TOOLS

- Xilinx ISE 14.2
- ModelSim 10.1 b
- MATLAB 12.1
- Keil µVision IDE
- SPARTEN-3E, SPARTEN-6 and VIRTEX-5 FPGA

AREAS OF RESEARCH INTEREST

- Network on Chip (NoC) and Reconfigurable Strucure
- System on Chip (SoC) and Multiprocessor system on chip(MPSoC)
- Embedded System and IP based design
- Real Time Operating System (RTOS)
- FPGA Synthesis and Verification

Ph.D PUBLICATIONS

1. **Adesh Kumar,** Piyush Kuchhal, Sonal Singhal **" Network on Chip for DTMF Decoder and TDM Switching in Telecommunication Network with HDL Environment" published in** Advance Computing Conference (IACC), Feb 2013 IEEE 3rd International Conference proceedings by IEEE digital Xplore doi: 10.1109/IAdCC.2013.6514464
2. **Adesh Kumar,** Sonal Singhal, Piyush Kuchhal **"Network on Chip for 3D Mesh Structure with Enhanced Security Algorithm in HDL Environment"** International Journal of Computer Applications (IJCA), USA, (ISBN: 973-93-80871-97-9) Volume 59– No.17 December 20112 (page 6-13)
3. **Adesh Kumar**, Piyush Kuchhal, Sonal Singhal, **"FPGA Implementation of Multistage Telecommunication Network on Chip (NOC) in HDL Environment**" ACM Transactions on Embedded Computing (TECS), USA, Nov 2013 **(Under Review)**
4. Adesh Kumar, Piyush Kuchhal, Sonal Singhal, **"Network on Chip Design and Synthesis for TACIT Network Security in Hardware Description Language Environment"** 2nd International Conference on Advanced Computing, Networking, and Informatics [ICACNI-2014] will be published by "Advances in Intelligent Systems and Computing", Springer [ISBN: 978-81-322-1664-3] **(Under Review)**

ABSTRACT

Telecommunication networks are often based on crossbar switching structure. Designers have utilized the crossbar switches in telecommunication system designs as electromechanical switches. Switch provides a connection between input and output. Non-blocking crossbar switching structure can be implemented using crosspoint switch fabric ICs. High performance nonblocking matrix crosspoint switches are used to avoid the congested path and eliminate the bandwidth problem that only one user can communicate in half duplex mode of communication. Telecommunication switching makes use of multistage crossbar switching to overcome the limitations of individual chips or boards. There are several benefits of multi-stage network. It is scalable and expandable. Multistage networks can be made completely non-blocking with the help of number of devices. It provides multicast and good broadcast features, and builds in reliability and redundancy with no single point of failure in the system. Existing system can be incremented with the addition of one or more modules. Switching elements can also be programmed for flexible crosspoint architecture, allows one connection path without affecting the connections for the other paths. This work focuses on the FPGA implementation of two, three, four and five stages telecommunication networks on chip. The generic architecture scheme is followed to implement the design in a cluster of 2 x 2, 4 x 4. 8 x 8 and 16 x 16 network configurations. The network parameters such as blocking probability, switching capacity and memory utilization are optimized and compared for two, three, four and five stages multistage networks. A comparison of two, three and four stages network is realized in terms of hardware utilization and timing parameters.

For the designing of the larger exchange, the cluster size of N = 8 is found an optimal solution in terms of memory utilization, hardware parameters, switching capacity, switching elements and blocking probability. For N = 8, the switching capacity of the network has been increased from 4, 16, 64, 256, 1024 calls, for single stage, two stages, three stages, four stages and five stages networks respectively. Blocking probability of multistage network increases with network size and decreases with the increment in number of stages, For N = 8, blocking probability is reduced. In case of four stage network the reduction is 28.80 % in comparison with three stage while in five stage network the reduction is 71.04 % in comparison with four stage. Blocking probability in turn improves the Grade of Service (GoS) of the network. The designs are synthesized in Xilinx ISE design suit 14.2 version and function simulation is carried out in Modelsim 10.1b student edition software. Very high speed integrated circuit hardware description language (VHDL) is used to develop the design. The experimental analysis is done with the Virtex- 5 FPGA to communicate the voice signal in multistage networks and validated with 3 KHz voice signal.

EXECUTIVE SUMMARY

Telephone exchanges are automatic and use electronic switching, where all the functions of the network operations are controlled with the help of processors. There are different memories and registers, used to store the information of source, destination subscribers and data. The information is stored using Stored Program Control (SPC) technique, in which, a set of instructions or program is stored in its memory and processor executes the instructions one by one automatically. The structures of telephone exchanges are complex and crossbar switches are larger in size. These factors lead to the traffic congestion in the network. Network switching capacity depends on the number of free links available to communicate among inlets and outlets. A digital network is a natural environment for data communication services. The applications relating to call handling are relating with the direct use of the facilities became available for data applications. Integrated Services Digital Network (ISDN) became available to provide digital access to digital facilities for internet operations.

In the telephone exchanges, the calls are processed by multistage environment. Till today conventional telephone switching is limited to three stage switching specially in India BSNL exchange is an example. The call operations and controlling functions in an exchange, largely depends on the line cards associated with the processors used to store the program. If the card is failed, there is no connection between the inlet and outlet. The traffic congestion and

limited bandwidth is a problem of telephone exchanges. In multistage networks there is the research gap in the implementation of programmable switch. The research can be focused in the implementation of programmable reconfigurable structure with the integration of multiplexing, routing, signaling and NoC topologies. The research is focused on the hardware chip implementation of the multistage telecommunication switching. The switching capacity of the exchange is increased with the implementation of the multistage network with programmable structure. In multistage network, there are alternate paths to provide the availability of the network. The four and five stage switching provide more capacity in comparison to three stage switching.

The execution of the multistage network is done with the consideration of the network cluster size of inlets/ outlets. The designing is done for single stage, two stage, three stage, four stage and five stage switching. The configuration is chosen of (2 x 2, 4 x 4, 8 x 8 and 16 x 16). First the designing of the networks is done in Xilinx ISE software, after that the simulation of each staged network is done with the help of Modelsim 10.1 b tools. The designing of each network depends on the routing scheme and available number of crosspoints. The networks are designed in such a manner that they provide the maximum alternate routes with the shortest path to reach the call from inlets to outlets. In the research work, each case of the routing is tested in simulation environment. The programmable structures provide the feasibility to reconfigure or reprogram the structures.

The concept of DTMF signaling is integrated with the chip which enables the inlets /outlets to communicate each other. In DTMF a key is represented with a combination of two sine waves. Dual tones of DTMF are called row and column frequencies of keypad. DTMF is the global standard for audible tones that represents the digits on a phone keypad. The landline phones which are based on touch tone pad generate the corresponding DTMF tone for a key of dial pad. The landline phone systems can then listen and decode that tone to determine which key was pressed, and thus enables dialing. The concept of TDM is also integrated with the multistage switching chip, which enable to access multiple users. The Principe of TDM switch is based on modulo counter which counts the number of users sequentially and time is allocated for a particular user. The data transfer scheme in the network is analyzed with the help of Virtex – 5 FPGA XC5VLX110T, a Digilent manufactured FPGA, and is validated for the voice signal of 3 KHz. The experimental set up is arranged with FPGA, and tested with different test cases with analog voice signals. The experimental set up and synthesis with FPGA guarantees the voice and data over the multistage networks, in which voice signals of some frequency say 3 KHz or higher frequency is given to FPGA. There is inbuilt Analog to Digital Converter (ADC) in FPGA, which converts the analog data into digital signal and FPGA processes the same signal to the output device via Digital to Analog (DAC) converter. The output device is CRO or Digital Storage Oscilloscope (DSO), which displays the same voice signal. A comparison of hardware parameters is carried out for all the stages. The synthesis report is generated that contains the information for hardware utilization

in terms of No of slices, No of flip flops, No of input LUTs, No. of bounded IOBs and No of gated clocks (GCLKs) used in the implementation of design. Timing analysis is also carried out for the staged network which provides the information of delay, minimum period, maximum frequency, minimum input arrival time before clock and maximum output required time after clock. Total memory utilization required by individual stage is also compared for different stages. The network parameters, blocking probability and switching capacity is optimized with the help of the synthesized results. It is seen that the blocking probability of multistage network increases with network size and decreases as number of stages are increasing.

The network security is also a great issue, when the data transfer is taken place over the internet. There are different algorithms available to provide network security like AES, DES, Triple DES, Triple AES, Kasumi, Blowfish, RSA, RC4, XMODES, but limited to their block size and key size of 128 bits maximum size. The new security algorithm named TACIT is a network security algorithm, in which block size and key size can vary of 'N' bits. If the key size is greater than the block size, TACIT network security can provide better results. The research focuses on the integration of network security, signaling, multiplexing and data communication in multistage networks. The network security is verified with the simulation tools with the block size and key size of 64 bits, 128 bits and 256 bits.

The switching capacity of the network is also calculated with each stage and found to increase with the increment of stages. As the number of stages are increasing, the hardware and memory utilization of the device is also increasing with network cluster configuration (N = 2, 4, 8, 16). Memory utilization for N = 16 is found as 19.01 % greater, for two stage switching in comparison to single stage switching, 38.03 % in three stage switching in comparison to two stage switching, 14.67 % in four stage switching in comparison to three stage switching and 13.93 % in five stage switching in comparison to four stage switching. The validation of voice signal is carried out in all stages of networks. In this design, N = 8, is found as the optimal solution to co-control the programmable telecommunication network. It shows that the switching of the telephone exchange is distributed in the cluster size of 8 users and controlling of exchange is done by cascading the same size of FPGA chips. For N = 16, the hardware utilization is found more. For N = 8, the switching capacity of the network is increased from 4, 16, 64, 256, 1024 calls, for single stage, two stage, three stage, four stage and five stage networks respectively. The research work of network chip implementation of four and five stage telecommunication network is a significant effort towards total digitization and programmable of switching system.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

ASIC	Application Specific Integrated circuits
ADC	Analog to Digital Converter
ADM	Adaptive Delta Modulation
AES	Advanced Encryption Standard
ATM	Asynchronous Transfer Mode
ARQ	Automatic Repeat Query
AM	Amplitude Modulation
ASK	Amplitude Shift Keying
BUFG	Generic Check Buffer
BRAM	Random Access Memory
BiNoC	Bidirectional Channel Network on Chip
BSNL	Bharat Sanchar Nigam Limited
CPLD	Complex programmable logic devices
CDMA	Code division Multiple Access
CGRA	Coarse Grained Reconfigurable Architecture
CRL	Certificate Revocation Unit
CA	Certificate Autority
CLB	Configuration Logic Block
CHC	Call Handling Circuit

CF	Call Forwarding
CAD	Computer Aided Design
CCITT	International Telegraph and Telephone Consultative Committee
CMOS	Complementary Metal Oxide Semiconductor
CE	Cache Element
CMP	Chip Multiprocessor
CPU	Central Processing Unit
DSP	Digital signal processing
DAC	Digital to Analog Converter
DSO	Digital Storage Oscilloscope
DUT	Design Under Test
DCM	Digital Clock Manager
DM	Delta Modulation
DTMF	Dual tone multi frequency
DM	Delta Modulation
DND	Do Not Disturb
DES	Data Encryption Standard
DRAM	Dynamic Random Access Memory
DDR	Double Data Rate
ESS	Electronic Switching System
EAX	Electronic Automatic Exchange

EMAC	Ethernet Medium Access Controllers
EOF	End of File
EDK	Embedded Development Kit
EDA	Electronic Design Automation
FPGA	Field Programmable Gate Array
FM	Frequency Modulation
FSK	Frequency Shift Keying
FIFO	First input First output
FSM	Finite State Machine
FXT	Full Duplex Transreceiver
FDC	Finite Domain Constraints
GPP	Generation Partnership Project
GE	Gigabit Ethernet
GMII	Gigabit Media Independent Interface
GCLK	Gated Clock
GoS	Grade of Service
HDL	Hardware Description Language
HOPE	Hotspot Prevention
HWNoC	Hardwired Network on Chip
HSEC	High Speed Ethernet IP Core
HTSS	Hybrid Telephone Network System

I/O	Input /Output
IOB	Input Output Block
IC	Integrated Circuit
ID	Identification
IP	Intellectual Property
IBUF	Input Buffer
IOBUF	Input Output Buffer
IEEE	Institute of Electrical and Electronics Engineering
IDEA	International Data Encryption Algorithm
ISE I	Integrated System Environment
LUT	Look Up Table
LED	Light Emitting Diode
LCD	Liquid Crystal Display
LSB	Least Significant Bit
MAC	Medium Access Control
MCR	Minimum Configuration Region
MDR	Memory Data Register
MEMS	Micro Electronics Mechanical System
MPEG	Moving Picture Expert Group
MPMC	Microprocessor/microcontroller system
MOSFET	Metal Oxide Semiconductor Field Effect Transistor

MPSoC	Multiprocessor system on chip
MIN	Multistage Interconnection Network
MF	Multi Frequency
M/B	Multicast/ Broadcast
MB	Megabyte
MII	Media Independent Interface
NoC	Network on chip
NPU	Network Processor Unit
NRZ	Non Return to Zero
NI	Network Interface
NDA	Non Disclosure Agreement
OBUF	Output Buffer
OCP-IP	Open Core Protocol (International Partnership)
OGB	Outgoing Bar
OTP	One Time Programmable
OSI	Open System Interconnect
PLD	Programmable Logics devices
PKI	Public Key Infrastructure
PE	Processing Elements
P2P	Point to Point
P&R	Place & Route

PC	Personal Computer
PTP	Peer To Peer
PM	Phase Modulation
PCM	Pulse Code Modulation
PAM	Pulse Amplitude Modulation
PWM	Pulse Width Modulation
PPM	Pulse Position Modulation
PSK	Phase Shift Keying
PIC	Programmable Interface Controller
PLL	Phase Locked Loop
PTSS	Programmable Telephone Software System
PSTN	Public Switching Telephone Network
QFT	Quantitative Feedback Theory
QoS	Quality of Service
RAM	Random Access memory
ROM	Read Only Memory
RCM	Reconfigurable Computing Module
RZ	Return to Zero
RSA	Rivest Shamir Adleman
RTL	Register Transfer Level
RXD	Receive Data

RSU	Road Side Unit
RGMII	Reduced Gigabit Media Independent Interface
SRAM	Static Random Access memory
SPIN	Scalable Programmable Integrated Network
SM	Switch Module
SPC	Stored Program Control
SoC	System on Chip
SHC	Service Handling Circuit
STD	Subscriber Trunk Dialing
STDB	Subscriber Trunk Dialing Bar
SNR	Signal to Noise Ratio
SC	Switching Capacity
SSL	Secured Socket Layer
SGMII	Serial Gigabit Media Independent Interface
SPDIF	Sony/ Philips Digital Interface Format
SAC	Stereo Audio Codec
SW	Switch
SIM	Subscriber Identification Module
SPI	Serial Peripheral Interface
TB	Test Bench
TDM	Time division Multiplexing

TCFR	Test Configurations Functional Region
TXD	Transmit Data
TDM	Time Division Multiplexing
TSS	Telephone switching System
TAPI	Telephony Application Program Interface
UART	Universal Asynchronous Receiver Transmitter
UMTS	Universal Mobile Telecommunication System
USA	United State of America
USB	Universal Serial Bus
UMARS	UnMapped Read Architecture
VLSI	Very Large Scale of Integration
VHDL	Very High Speed Integrated Circuit Hardware Description Language
VANET	Vehicular Ad HoC Network
VGA	Video Graphics Array
VOQ	Virtual Output Queue
XBT	Xilinx Synthesis Technology
ZBT	Zero Bus Turnaround

LIST OF ABBREVATIONS

m	Meter
cm	Centimeter
km	Kilometer
µm	Micometer
nm	Nanometer
b	Bit
B	Byte
kB	Kilo Byte
MB	Mega Byte
bps	Bits per second
kbps	Kilo bits per second
Mbps	Mega bits per second
Hz	Hetrz
KHz	Kilo Hertz
MHz	Mega Hertz
W	Watt
mW	Milli Watt
µP	Microprocessor
µC	Microcontroller

CHAPTER-1

INTRODUCTION

The chapter describes the introduction of digital telephony and programmable telecommunication networks. The need of work and motivation of multistage telecommunication networks is discussed. It also includes the problem statement, objectives, supporting parameters for software, hardware and theoretical background. The methodology and design flow is also discussed in the chapter.

1.1 INTRODUCTION TO DIGITAL TELEPHONY

Digital telephony [1, 2] refers to the telephone exchange environment, in which the telephone operations are controlled using digital signals. US organization Bell Laboratories introduced the first digital computer control switching system named as Electronics Switching System (ESS) No. 1 [42, 96]. Modern computers follow the concept of Stored Program Control (SPC) [2], in which, a set of instructions or program is stored in its memory and processor executes the instructions one by one automatically. All the control functions of the exchange are executed through programs stored in the memory is called stored program control. SPC technique in ESS allowed the new features such as call dialing, call waiting, abbreviated dialing [46] and three way calling facility. In electronic exchange, call processing is handled by the different software, which are used to create and terminate every call by the main system call process. Telecommunication networks are used to route the calls with different stages. As the user dials the number, it is processed by the network. Telecommunication network requires more switching nodes to establish the connections, but achieves significant savings in the number of trunks. A switching network [2, 91] can be made up of single stage or multistage switching [91] blocks which are distributed through intermediate frames. The number of interconnections differs from exchange to exchange.

Telecommunication wired network require routing equipment, bandwidth requirement, set of unique protocols and bandwidth growth rate. For example, the telecom industry has typically increased bandwidth in increments of roughly four (2.5 Gbps to 10 Gbps, now moving to 40 Gbps), while computer networking has done the job in leaps of 10 x (100M, 10G, 100G) [10]. There is the requirement of sophisticated equipment's such as switches [3], routers [1, 5] and transport systems with advanced circuitry to transport data at these bandwidths. For example, a series of line cards are the heart of a metro router. Under a wide set of protocols, line card [46] receives data packets and examines the packets for size, origin, destination along the network. There is the requirement to complete all functions and computations in nanoseconds.

Telecommunication networks demands high speed switching to deploy and create new and novel services. High speed switching does not depend on the service provider's operational infrastructure [8] but it depends on the flexibility of the software architecture. Current telecommunication networks are based on the network structure based on architecture over 30 years in age [42]. The architecture has less computational capabilities and bandwidth is a factor for degree of flexibility. The network switching can be increased using cluster based models where large processors are distributed over the network and to provide distributed services [91]. Dedicated processors used monolithic software [10], which control the response, necessary data and coordinate with the services modules. Traditional communication fabrics [91] have scalability issues in terms of performance and physical circuit design. Networking concept can be brought on chip domain in terms of Network-on-Chip (NoC) paradigm [1] [5], which resolve the concern with a scalable design, at both physical and architectural levels. The NoC system [6] [7] [16] is the combination of different processing blocks which can operate at different clock frequencies and timing requirements. Synchronization is required to communicate among all processing blocks.

The general operation of a switch is to set up and release a connection between two communication entities. Single stage space division network [8] has the limitations that the numbers of crosspoint switches required are prohibitive [91], the no of possible paths are N (N-1) for a square array and N (N-1)/2 for a triangular array [42]. The utility of one crosspoint is that it can be utilized for only one inlet/outlet pair connection. If the connection is failed, there is not any alternative path to communicate with destination subscriber. In a telephone exchange one crosspoint should be utilized to establish more than one connection. It is possible to increase the utilization efficiency of crosspoints, if they are shared for more than one connection. This work focuses on the chip design and synthesis of four and five stage multistage networks. The networks should be realized in full duplex mode and full available network. The reason to opt the four and five stage networks is to increase the switching capacity of telecommunication network in optimal hardware solution. The design is carried out using Xilinx ISE design suit 14.2 version and functional simulation is done in Modelsim 10.1b student edition software.

1.2 NEED & MOTIVATION

There are three basic elements in the communication network, switches, terminals and transmission media. Electronic switching system (ESS) [1] or Electronics Automatic Exchange (EAX) [2] is used to control the switching functions of a central office. In a single stage space division network, the specific connection between two subscribers was established with the help of a crosspoint switching element [91]. If the switching element is failed, a conversion is not possible between two subscribers. In a multistage space division network, the conversion could be established via any one of the many alternative paths or a crosspoint can be utilized to provide more than one connection. A switching element, once allotted, remains dedicated to a connection for its entire duration because a continuous analog speed signal is passed through the switch in space division switching.

Telephone exchanges are automatic and use electronic switching where all the functions of the network operations are controlled with the help of processors. There are different memories and registers, used to store the information of source, destination subscriber and data. The information is stored using SPC technique. Call operations and controlling functions in an exchange, largely depends on the line cards associated with these processors. If any of line cards gets failed, no communication remains intact. It is because that the processor is not configured using programmable chips. Although telephone exchanges are digital, they are not yet fully programmable. Programmable structures help in replacement of the chip. It also provides the possibility to reprogram for the specific operation, rather than changing the processors [28]. Thus the need for the programmable network is eminent.

In addition to the non-programmability, telephone exchanges are complex in structure and crossbar switches are larger in size. These factors lead to the traffic congestion in a network [67]. Network switching capacity depends on the number of route available for inlet to outlets. Till today conventional telephone switching is limited to three stage switching. Increasing number of stages in a conventional network is not the feasible solution as it further adds to the hardware complexity along with larger crossbar switches [17, 91]. This limitation can be overcome by the increasing the number of stages such as utilizing four and five stage with programmable network configurations and switching capacity of a network can be enhanced considerably.

In digital transmission, sampled values of the speech are sent as Pulse Amplitude Modulated (PAM) values or Pulse Code Modulated (PCM) binary words. A sampled value should be transmitted from an inlet to outlet in a few microseconds or even less, through a switching element. If one can provide a dynamic control, where a switching element can be assigned to a number of inlet and outlet pairs, it can be shared by a number of active speech circuits simultaneously in time domain. A switching element can also be made programmable according to the route. It will be greater saving in terms of hardware. Also the integration of multistage switching with time division [46, 91] will improve the efficiency and switching capacity of telephone exchange. A telephone exchange operates 24 hours in a day, 365 days and numbers of years without interruption [5]. This in turn dictates that the exchange should be highly tolerant to control the faults. Early commercial computers were unknown to fault tolerant features [4, 19] and switching engineers had the problem with the development of software and hardware for fault tolerant.

The motivation of digital machines was to reduce manufacturing cost, reduced floor area, low maintenance and simplified expansion. Interoffice transmission was changed to complete digital in mid 1980s and analog switches were replaced to digital. Analog toll [2] and end offices transmission swing to digital conversion cost moved from the associated digital transmission links. The associated trunks with interoffice were already digital. So digital loop carrier system was a cost effective solution in metropolitan applications and digital fiber [14] can be used at the feeder points. Today, all exchanges are based on the SPC, which is an attempt to replace the space division electromechanical switching matrices by semiconductor cross point matrices. Speed, size and cost are three important factors while designing the electronic system. Microprocessor/microcontroller (MPMC) [46)] system can handle sequential operations with high flexibility and use of Field Programmable Gate Array (FPGA) can handle concurrent operations with high speed in small size area. Thus system performance can be enhanced with the combination of these features. The combination of SPC and its implementation in Hardware Description Language (HDL) environment leads to Programmable Telephone Switching System (PTSS) [3]. PTSS can be designed as a combination of stored program control (SPC) and VLSI technology. As already mentioned, that in telephone switching systems [2, 46], it is not feasible to increase the junction or extension lines [42, 46] because of limitation of processor or controlling system as it will not be cost effective and hardware requirement is more. With the implementation of the controlling system for eight lines in a FPGA and then provides the provision to co-control or cascade to the other FPGA of another eight lines will rectify the above issue.

NoCs support efficient on-chip communication [6, 7] potentially leads to NoC based multiprocessor systems characterized by high structural complexity and functional diversity. Multiprocessor System On-Chip (MPSoCs) [8] consists of complex integrated components, which communicates with each other at very high speed rates. A single shared bus or hierarchies of buses are not feasible for intercommunication. Intercommunication requirements of MPSoCs is made up of about hundred cores having poor scalability with their shared bandwidth between all the attached cores, system size and the energy efficient requirements of final products. NoCs are a promising solution to the scalability problem of forthcoming MPSoCs [5, 9, 10]. NoC is an approach for designing the telecommunication subsystem between IP cores in a System on Chip (SoC) [10, 13]. The software and application layer is a very critical aspect on the NoC communication stack [11]. Secured transmission among nodes is possible if it follows the NoC layer protocol [14, 19], mostly physical and network layers. All networked layers follow the pipeline and parallel processing that validate the optimized hardware parameters on chip development.

The network design and routing of the telecommunication system depends on multistage switching architecture followed by two stage, three stage, four stage and five stage structures [91]. A flexible FPGA based NoC design that consists of processors and reconfigurable components can be integrated into a single NoC chip. The blocking probability, switching elements, switching capacity of the crossbar multistage telecommunication network should be optimized for reconfigurable NoC structures. Secured data transmission is an issue, when the data is routed through network. Secured transmission can be guaranteed, if the concept of encryption and decryption can be integrated in the NoC chip. In such a case the data is locked at transmission end and is retrieved at the receiving end. The NoC structure can be synthesized on Xilinx supporting FPGA which is used to implement programmable multistage network [102]. For these reasons, research can be emphasized to increase the switching capacity and to increase the communication lines and reduction in blocking probability in four and five stage multistage networks. The integration of network security algorithm with multistage network chip will enhance the system performance and security [84].

1.3 PROBLEM STATEMENT

On the basis of the above mentioned research gaps there was a stringent requirement to carry out the systematic study on such reconfigurable networks. It formulates and defines the problem statement as "Design, Validation and Field Programmable Gate Array (FPGA) implementation of Multistage Telecommunication Network in Hardware Description Language (HDL) Environment"

Problem statement includes the chip design, modeling, simulation and FPGA synthesis of multistage telecommunication network on chip. VHDL has been chosen as the HDL language.

1.4 OBJECTIVES

The objective of the research work is to design the N-stages network and to optimize the hardware parameters. This work also includes the comparison of the obtained parameters with different stages network parameters. FPGA implementation of N-stages network is carried out to validate the results. VHDL programming is used to implement the hardware of N-stages networks starting from single stage, two stage, three stage, four stage and five stage networks. In particular, following are the key objectives to reduce the blocking probability and to increase the switching capacity.

- To design the network in VHDL environment by optimizing the switching parameters

  Blocking probability

  Increasing the calling / switching capacity

To reduce the existing hardware

- Virtex-5 FPGA synthesis and experimentation on FPGA platform.
- To validate the programmable switching system results with switching parameters
- Integration of network security with multistage network.

1.5 PARAMETERS SUPPORTING TO CHIP DEVELOPMENT

The parameters relating to chip design, modeling, simulation and FPGA implementation [11, 21] and validation are the followings:

- *Synthesis Options Summary:* Synthesis includes the functional simulation and logic optimization. Register Transfer level (RTL) is the chip view, extracted after the modeling of the chip, contains all the possible inputs and outputs, using those pins the chip is developed.
- *VHDL Compilation:* The compilation depends on the approaches used in the designing of the chip weather it is used the top down approach or bottom up approach. It depends on the simulation software used for the chip development. In the research work, the simulation tool is Xilinx ISE Design Suite 14.2.
- *VHDL Analysis*: The VHDL analysis depends on the simulation environment used to check the functionality of the chip developed.
- *Device utilization summary:* Device utilization report gives the percentage utilization of device hardware for the chip implementation. Device hardware includes, logic gates, buffers, multiplexer, decoders, latches, flip flops etc. Synthesis report shows the complete details of device utilization.

- *Timing report and delay time calculation:* Timing details provides the information of delay, minimum period, minimum input arrival time before clock and maximum output time required after clock, and time required to propagate the input to the output.

1.5.1 HARDWARE PARAMETERS

The hardware parameters are realizing in terms of FSM encoding, decoding, combinational and sequential logics development. The brief description of each parameter is discussed sequentially.

- *FSM Encoding Algorithm:* FSM encoding determines the FSM [101] coding technique should be used to configure the design. Possible FSM encoding algorithms are auto, one hot, compact, sequential, and gray. Auto encoding selects the needed optimization algorithms during the synthesis process. In one hot encoding, the numbers of flip flops are based on the number of states. For an example, a 31 sates FSM will have 31 flip flops after synthesis. It is advantageous because only one flip flop is hot or active during one state transition. Therefore, it has low power with each state. Binary encoding takes 'n' flip-flops for 2^n states. Binary coding will take 5 flip flops to cover 31 states, but the disadvantage is that decoding logic is complex. Compact encoding Minimizes the number of state variables and flip flops and is based on hypercube immersion. Sequential encoding consists of identifying long paths and applying successive radix two codes to the states on these paths. Next state equations are minimized. Gray encoding guarantees that only one state variable switches between two consecutive states.

- *FSM Style:* It specifies whether to map the FSM LUTs or block RAM. By default, FSM style is set to LUT.
- *RAM Extraction:* The parameter is relating to the amount of RAM memory utilized in the chip implementation, it is extracted from device utilization report.
- *RAM Style:* The parameters is relating to the type of RAM memory developed, single port RAM, double port RAM and single clocked RAM, double clocked RAM. It can be Auto, distributed or block RAM. In auto XST uses best implementation for each macro. In distributed RAM, memory is distributed in different sizes. In block RAM, entire memory is implemented in different blocks of same size.
- *ROM Extraction:* This parameter relates to the amount of ROM memory utilized in the chip implementation, it is also extracted from device utilization report.
- *ROM Style:* It is related to the ROM type which is developed. It gives the information that weather generated ROM is based on single clock pulse or double clock pulse. ROM memory also can be auto, distribute or block.
- *Mux Extraction:* This parameter relates to the number of multiplexers used in the chip implementation, it is also extracted from device utilization report.
- *Mux Style*: In the chip integration, RTL internal view the logic can be configured using 2 x 1 mux, 4 x 1mux, 8 x 1 mux and 16 x 1 mux, or depends on the logic inputs required to configure the structure.

- *Decoder Extraction:* Decoder is the essential part of the network structure which is utilized in the address selection of the node in NoC design.
- *Priority Encoder:* Selection logic to assign the priority which accepts 2^n inputs and extract the data of *'n'* bits.
- *Shift Register Extraction:* Used to select the number of shift registers used, it depends on the no of flip-flops used to store the information.
- *Logical Shifter Extraction:* Based on shifting operations like left shift, right shift, rotate left and rotate right operations logical shifter is selected.
- *XOR Collapsing:* how many numbers of XOR gates are required, can be found by synthesis report.
- *Resource Sharing*: it contains the information of number of interconnecting logic used for implementing the chip.
- *Multiplier Style:* Used to identify the number of multipliers used in the chip development.
- *Automatic Register Balancing:* Optimal number of buffer registers used to store the data.
- *DSP blocks:* A DSP block provides optimal solution for DSP operations with maximum functions and minimum logic resource utilization. Each DSP block consists of adders, multipliers, accumulators, subtractors and a summation unit. Each DSP block can support a variety of multipliers vary in size (9 x 9, 18 x 18, 36 x 36), operation modes such as multiplication, complex multiplication, multiply-addition and multiply-accumulation.

1.5.2 FPGA PARAMETERS

The parameters related to the hardware synthesis and logic optimization are called FPGA parameters.

- *I/O Buffers:* Add I/O Buffers is the target option for Xilinx Synthesis Technology (XST) to add or not add Input Buffers (IBUF), Output Buffers (OBUF) or Input Output Buffers (IOBUF) on the top level ports. By default, it is enabled because each design requires I/O buffers on the top level port. In a sub module synthesis, it needs to disable because I/O buffer should be in the top level.
- *Global Maximum Fan-out*: The maximum fan-out of an output measures its load-driving capability. For Virtex 5 FPGA device, max fan-out is default 100000. If the value of max fan-out synthesis is changed, it will be applied globally and affect the whole design.
- *Generic Clock Buffer (BUFG):* In Xilinx design, the programmer uses global clock buffers to take advantage of the low-skew, and high-drive capabilities. Whenever an input signal drives a clock signal, FPGA Compiler automatically inserts a generic global clock buffer (BUFG). The Xilinx implementation software automatically selects the clock buffers, which are appropriate for specified design constraints.
- *Register Duplication:* This parameter moves registers through combinatorial logic to evenly distribute the path's delay between registers. It is also called as flip-flop retiming. In the process, both forward and backward retiming are

possible but XST does not perform flip flop retiming. It specifies whether or not the designer want to replicate the register to help control fanout.

- *Slice Packing:* XST currently has a "Slice Packing" switch, help in grouping LUTs into slices during optimization. It does not provide only more accurate timing information for optimization, but it is also passed to implementation for maintain more consistency during mapping.
- *Optimized Instantiated Primitives:* It is the default mode, which allows XST to perform efficient optimization across module boundaries. If the designer needed, it is possible to keep Hierarchy in synthesis.
- *Use Clock Enable:* XST optimizes the logic such that the clock enable signal and reset condition are combined and fed to the reset input of the counter. The property specifies whether or not clock enable pins are utilized by XST. When clock enable is set to auto mode, XST uses dedicated clock enable pins on inferred registers if they provide a benefit to the overall quality of the design. When it is set to yes, clock enable pins are used in flip-flops. When set to No, there is no use of clock enable pins, and the corresponding functionality is implemented in standard logic.
- *Use Synchronous Set:* Synchronous set is synchronized with clock and specifies whether or not synchronous set pins are utilized by XST. When the pin is set to auto mode, XST uses dedicated synchronous set pins on inferred registers if they provide a benefit to the overall quality of the design. When it is set to Yes, synchronous set pins are used in flip-flops. When set to No, there

is no utilization of synchronous set pin, and the corresponding functionality is implemented in standard logic.

- *Use Synchronous Reset:* Synchronous reset is synchronized witch clock and specifies whether or not synchronous set pins are utilized by XST. If reset is done all the contents of flip flops are zero.
- *Pack IO Registers into IOBs: It* controls the IOB flip flop merging capabilities. This option forces all flip-flops connected to pads into the IOB, whatever possible. In auto mode, this option uses the timing offsets and periods specifications to determine if IOB flip-flop merging should be done. In no condition, it prevents IOB flip-flop merging from occurring.
- *Equivalent register Removal:* For the optimization of flip-flops the property is utilized. Flip-flop optimization includes the removal of flip-flops and equivalent flip-flops with constant inputs.

1.5.3 GENERAL PARAMETERS

The general parameters are related to RTL, module hierarchy, netlist generation, memory and slice utilization and are given below:

- *Optimization Goal:* The optimization parameters for the chip are speed, cost, delay and power consumption.
- *Optimization Effort:* Speed Optimizes the design for speed by reducing the levels of logic and area optimizes the space for area by reducing the total amount of logic used for design implementation.
- *RTL Output:* Chip synthesis optimizes the design using minimization and algebraic factoring algorithms. Additional optimizations are tuned to the

selected device architecture and higher multiple optimization algorithms are used to get the best result for the target architecture. If the designer need to optimize the inputs and outputs of the chip, it is possible with the help of RTL.

- *Keep Hierarchy:* This property specifies whether the corresponding design units merged with the rest of the design. It can be No, Yes and Soft. Soft is used when the designers want to maintain the hierarchy through synthesis.
- *Netlist Hierarchy:* Netlist hierarchy controls the form in which the final netlist is generated. It allows the designer to write the hierarchical netlist even if the optimization was done on a fully or partially flattened design.
- *Global Optimization:* Global optimization specifies the global timing optimization goal. There are some properties for global optimization which are All Clock Nets, inpad to outpad, offset in before, offset out after and maximum delay. All Clock Nets optimizes the period of the entire design. inpad to outpad optimizes the maximum delay from input pad to output pad throughout an entire design. Offset in before optimizes the maximum delay from input pad to clock, either for an entire design or for a specific clock. Offset out after optimizes the maximum delay from clock to output pad, either for a specific clock or for an entire design. Global optimization will be also set to maximum delay constraints for paths from staring input to output.
- *Slice Utilization Ratio:* It specifies the area in percentage that XST will not exceed during timing optimization. XST will make timing optimization, if the

area constraint cannot be satisfied. Default, the designer keeps this ratio 100%.

- *BRAM Utilization Ratio:* It specifies the number of BRAM blocks in percentage XST will not exceed during timing optimization. Default, the designer keeps this ratio 100 %.
- *Auto BRAM Packing:* It specifies whether or not XST will try to pack two small single-port BRAMs into a single BRAM primitive, to form dual-port BRAM. If BRAMs are at the same hierarchical level in the design, only then can be packed together.
- *Slice Utilization Ratio Delta:* This constraint defines the percentage of slices XST can use to implement the design or a block of designer's need.

1.5.4 PERFORMANCE PARAMETERS

Performance parameters for a lossy system are the grade of service (GOS) and the blocking probability (P_B) and switching capacity [4, 91]. The parameters depend on the traffic congestion, routing delay and network cluster or population size.

- *Grade of Service (GoS):* In lossy system, the network carries less traffic than the actual traffic offered to the network by the subscribers. The overall traffic is rejected and is an index of the quality of service offered by the network. This is known as grade of service (GoS) [46] and it is the ratio of lost traffic to offered traffic.

$$\text{GoS} = \frac{Lost\,Traffic}{Offered\,Traffic}$$

$$\text{GOS} = \frac{A - A_0}{A} \qquad \text{Equ. 1.1}$$

Where

A_0 = Traffic carried by the network.

A = Offered traffic to the network

- *Blocking Probability (P_B):* Blocking probability is the probability that all the servers in a system are busy. When all the servers are busy, the network will not carry any further traffic and arriving subscriber traffic is blocked. The probability that all the servers are busy may be presented by calls lost.

Blocking Probability (P_B) = Congestion probability

Blocking probability is calculated using Poisson process. The governing equation of a Poisson Process is [11]

$$P_k(t) = \frac{(\lambda t)^k e^{-\lambda t}}{k!} \qquad \text{Equ. 1.2}$$

Where λ = Calls per second, the value of't' is taken from the deice utilization report as the value of minimum time.

- *Switching Capacity:* The total capacity of public switching exchanges corresponds to the maximum number of fixed telephone lines that can be connected. Therefore, this number includes fixed telephone lines already connected and fixed lines available for future connection, including those used for the technical operation of the exchange. The measure should be the actual capacity of the system, rather than the theoretical potential when the system is upgraded or if compression technology is employed. Switching capacity is the capacity of the multistage network when it is full connected and full available network.

1.6 METHODOLOGY &DESIGN FLOW

The methodology comprises the different steps which are carried out for the design and development of chip. The model is shown in figure 1.1.

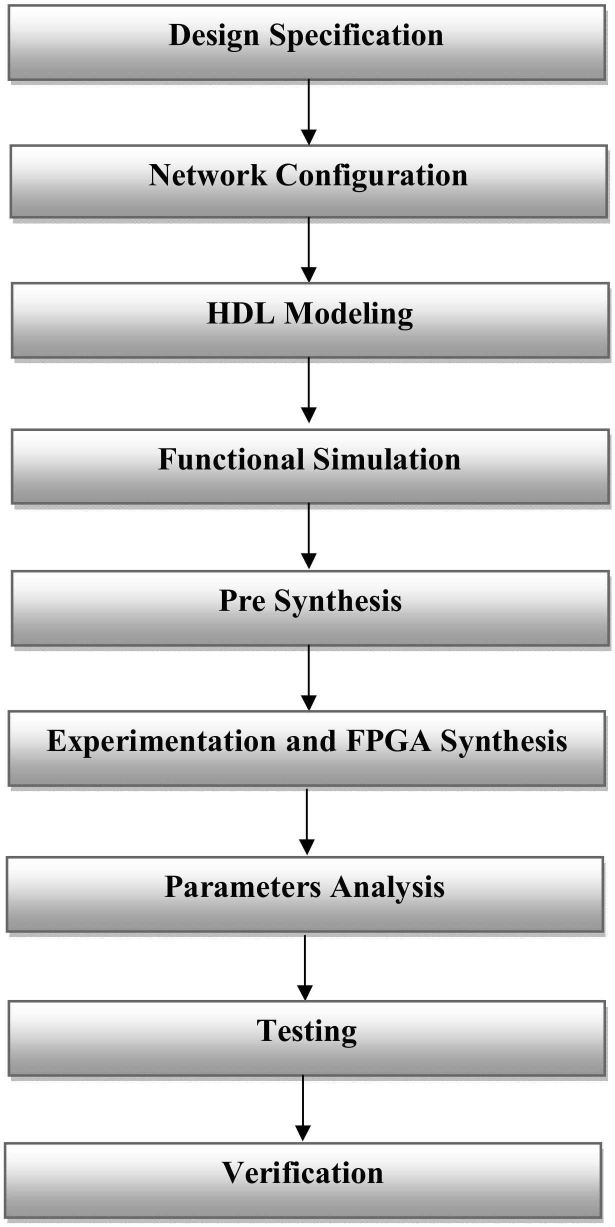


Fig. 1.1 Steps in methodology

- *Design Specification:* In design specifications the designer decides to develop the chip either in top down or bottom up approach. In bottom up approach the circuit is designed using micro module to form a design and top down approach a design is distributed in sub modules. In the multistage networks bottom up approach is utilized to implement the design.
- *Network Configuration:* The designer has to choose the cluster size of the design. In the case of multistage network, the network configuration needs to be decided. For example, for single stage , two stage, three stage, four stage and five stage networks the network configuration is 2 x 2, 4 x 4 , 8 x 8, 16 x 16 or more with respect to inlets and outlets respectively.
- *HDL Modeling:* The designer has to understand the feasibility to design with the supporting languages such as VHDL, Verilog HDL, and System C etc. The designer also decides the modeling of chip and design constructs in data flow, behavioral and structural model.
- *Functional Simulation:* The designed modules are checked according to their functionality and test cases. The functional simulation depends on the test benches developed by the designer, clock frequency and reset circuitry.
- *Pre Synthesis:* Pre synthesis includes the RTL simulation, device synthesis report contains the summary of hardware parameters with combinational and sequential circuit. If the hardware utilization is greater than the 100 % for the configured device, the designer has to redesign and check for the optimized device and timing parameters.

- *Experimentation and FPGA Synthesis:* The experimental setup is arranged to check the functionality of chip with its compatibility and interfacing to FPGA board. The maximum support frequency of FPGA board is analyzed to check the data transfer rate. In the experimental setup inputs can be through switches, and output can be analyzed with the help of LEDs. There is inbuilt ADC and DAC in Virtex 5 FPGA to check the real time functionality.
- *Parameters Analysis:* The FPGA source and target parameters are analyzed with the help of optimized FPGA results. Static timing analysis and device utilization is also analyzed with minim speed grade and memory utilization.
- *Testing:* The synthesized results are tested for the different test cases and combination with the help of LUTs. In the multistage network the inlets and outlets functionality is checked for maximum combination, inert and interexchange communication. The developed chip is also tested for analog input given to FPGA and processed output signals on DSO. The signal characteristics are also tested with the FPGA device compatibility and display unit.
- *Verification:* The Design Under Test (DUT) is verified with timing parameters and test cases. Standard VHDL has all the features necessary to code randomization of stimulus and functional coverage, both are very important while verifying larger, system-level designs. Verification is used to describe testing of a group of logic using a test bench, implemented for every verification level. The most common verification levels are SoC verification level, sub SoC verification level with a group of IP blocks, IP block

verification level building test benches around IP blocks. IP blocks verification allows greater control to stress internal blocks more heavily than a SoC test bench can provide.

1.6.1 VIRTEX- 5 FPGA DESIGN FLOW

The Virtex 5 family provides the most recent and powerful features within Xilinx FPGA families. There are existing five distinct sub-families of Virtex 5 FPGA. Each platform contains a different ratio of features to address the needs of a wide variety of advanced logic designs. Moreover, in addition to the most advanced features, high-performance logic fabric, Virtex 5 FPGAs contain many hard-IP system level blocks, including powerful second generation 25 x 18 DSP slices, 36-Kbit block RAM/FIFOs, and enhanced clock management tiles with integrated Digital Clock Manager (DCM) and Phase Locked Loop (PLL) clock generators with advanced configuration options. It has additional platform dependant features include tri mode Ethernet Media Access Controllers (EMACs), power optimized high speed serial transceiver blocks for enhanced serial connectivity, and high-performance Power PC 440 microprocessor embedded hard core blocks. The research is based on the FPGA flow shown in figure 1.2.

The first step involves understanding of the design requirements, initial design entry, problem decomposition, and functional specifications where correctness by comparing outputs of the VHDL model and the behavioral model are checked. The multistage network is designed with the help of VHDL programming language. The designed may be based on FSM encoding, state diagram analysis. Synthesis involves the conversion of an HDL description to a netlist which is basically a gate level description of the design.

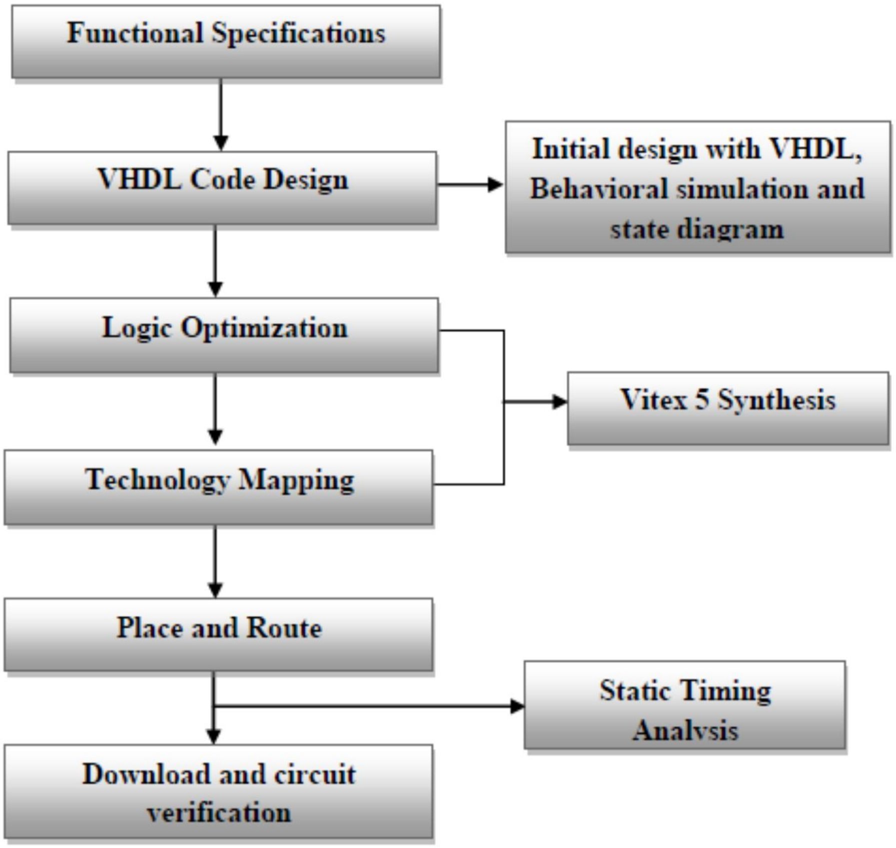

Functional Specifications
VHDL Code Design
Initial design with VHDL, Behavioral simulation and state diagram
Logic Optimization
Vitex 5 Synthesis
Technology Mapping
Place and Route
Static Timing Analysis
Download and circuit verification

Fig. 1.2 Virtex-5 FPGA Design Flow

In the FPGA synthesis, different optimization constraints are applied to the design. In implementation of the chip, the generated netlist is mapped onto Virtex 5 FPGA device's internal structure using technology libraries. Logic optimization process optimizes Boolean expressions into a standard form to optimize area or speed. Technology mapping minimizes blocks to minimum area. The main phase of the implementation stage is place and route, which allocates FPGA resources. The general FPGA resources may such as logic cells, hard core blocks, memory, and connection wires. Thereafter, configuration data are written to a special file by a program called bit stream. For timing analysis, there is a special software checks whether the implemented design satisfies timing constraints specified by the designer. In static timing analysis, the actual delay models are used to estimate the real delay on the chip after routing. Routing is helpful to provide connections between cells to minimize area. After the FPGA synthesis the chip is ready to send to foundries for fabrication unit.

1.7 OUTLINE OF CHAPTERS

- Chapter-1 presents the introduction to digital telephony, need & motivation, problem statement, objectives, research model and research methodology.
- Chapter-2 presents the literature review and contributions by the different researchers on multistage network and network security.
- Chapter-3 presents the multistage networks and routing scheme in multistage networks (single stage switching to five stage switching)
- Chapter-4 presents the 3D NoC, Network security and key management policy, used for secured networks.
- Chapter-5 presents the experimental set up and synthesis environment with different testes cases and analysis. The chapter also includes the display of the experimented data.
- Chapter-6 presents the results and discussion. Results include Xilinx 14.2 ISE chip design synthesis report, functional simulation on Modelsim 10.1 b and FPGA synthesis. FPGA synthesis includes device utilization summary, timing parameters. A comparison of FPGA parameters with single stage, two stage,

three stage, four stage and five stage switching network is also included with the chapter. The chapter also includes the statistical simulated data and analysis with optimized hardware and experimental synthesis work on Virtex - 5 FPGA.

- Chapter-7 presents the conclusion and future scope.

CHAPTER SUMMARY

A telecommunications network is a collection of links, terminal nodes, and any intermediate nodes, which are connected so as to enable telecommunication between the terminals. The nodes are connected together using transmission links. The nodes use circuit switching, packet switching or message switching to transmit the signals through the correct links and nodes to reach the correct destination terminal. Individual terminal in the network usually has a separate address so messages or connections can be routed to the correct recipients. Messages are generated by a sending terminal, routed through the network of links and nodes until they arrive at the destination terminal. In telecommunication networks, the message is routed in multistage switching because in single stage switching there is only one dedicated path to establish the connections between source and destination subscribers. In multistage switching, there are alternative ways, in case of failure of dedicated link. All telecommunication networks are made up of five basic components that are present in each network environment regardless of type or use. These basic components include terminals, telecommunication processors, telecommunication channels, computers, and telecommunication control software. The SPC technique is used to store the data, with the help of processors. The data transfer from one node to another node can be secured if the concept of programmable chips can be integrated with network security algorithms. The switching structures can reduce the hardware used to control the switching operations, with the integration of transmission, routing, signaling and security in programmable chips and make a supervision of control and co control. In the chapter, it has been summarized the need and motivation of programming telecommunication networks with the integration of secured transmission. The general parameters, hardware parameters, theoretical parameters, source and target parameters are also discussed, with the synthesis process on Virtex-5 FPGA.

CHAPTER - 2

LITERATURE SURVEY

The chapter describes the literature survey carried out in understanding the concept of NoC, multistage network, routing schemes, interlink communication and structures to configure a reconfigurable structure. The chapter describes the detailed literature survey based on the different research papers.

2.1 LITERATURE REVIEW

The NoC architecture is an m x n mesh [1] of switches and resources are placed on the slots formed by the switches. It follows a direct layout of the 2D mesh of switches [6, 7] and resources providing physical architectural level design integration. The connection of each switch is with one resource and four neighboring switches, and every resource is connected to one switch. The resources of switches can be a processor core, an FPGA [2], memory, a custom hardware block or any other intellectual property (IP) block, which fits into the available slot and complies with the interface of the NoC [1, 3]. The actual architecture of NoC essentially is the on-chip communication infrastructure comprising the layer relating to data link, physical layer, and the network layer of the Open System Interconnections (OSI) protocol stack model [9]. It can be used to define the concept of a regional hardware unit, which utilizes an area of any number of resources and switches. The concept is allowed to the NoC, to accommodate large resources such as large memory register banks, FPGA resources areas, or special purpose computation resources such as high performance and multiprocessors. NoC architecture can be differentiated based on network topology, flow control schemes, routing methodology, switching and the techniques applied to ensure quality of service for data transmission.

The topology of a NoC specifies the physical organization of the interconnection network. The topological structure defines switches, nodes and links which are interconnected to each other. Topologies for NoCs [12, 13] can be classified into two broad categories: 1) direct network topologies, in which each node or switch is connected to at least one core IP , and 2) indirect network topologies, where the subset of switches or nodes are not connected to any core IP and performs only network operation. Both direct and indirect topology can be regular like meshes, tori, k-ary n-cubes [15] and fat trees or irregular customized application specific topology. Most NoCs implement regular forms of network topology that can be laid out on a chip surface for a 2D plane, for an example, a k-ary 2 cube (where k is the degree of each dimension and 2 is the number of dimensions) commonly known as grid based topologies. Besides the form, the nature of links adds an additional aspect to the topology. The popular NoC topologies are based on k-ary 2-cube networks, the nature of link are the mesh which uses bidirectional links and torus which uses unidirectional links. For the structures like torus, a folding can be employed to reduce long wires. Millberg et al. has NoC presented the NOSTRUM [15, 16], a folded torus is discarded in favor of a mesh with the argument that it has longer delays between routing nodes. Generally, mesh topology makes better use of links and their utilization, while tree based topologies are useful for exploiting locality of traffic.

The NoC switching strategy determines how data flows through the routers in the network. NoCs use packet switching as the fundamental transportation mode. Packet switching is a communications paradigm in which packets are routed between nodes over data links shared with other traffic. Packets are queued or buffered in each queue, resulting in variable delay. This is in contrasts with the circuit switching, other principal paradigm, which sets up a limited number of constant bit rate and constant delay connections between nodes for their exclusive use for the duration of the communication [5]. In packet switching, instead of establishing a path before sending any data, the packets are transmitted from the source and make their way independently to the receiver that is possibly along different routes and with different delays. There are mainly three kinds of switching schemes [5], store and forward, virtual cut through and wormhole switching.

2.2 FINDINGS OF LITERATURE REVIEW

The following listed papers have discussed, in general, the relevance of NoC architecture, multistage cross bar networks, their routing schemes and testing methodology [17] .It has been suggested by many authors that a NoC structure can follow mesh, tree, ring, torus or hierarchical structures. Each NoC structure has Processing Elements (PEs), which can be reprogrammed based on deflection routing. These papers also explain the layered architecture of NoC, Scalable

Programmable Integrated Network (SPIN) on-chip micro network that defines packets as sequences of 32-bit words, with the packet header fitting in the first word. The network uses a byte in the header to identify the destination address, which allows the network to scale up to *'N '*terminal nodes. In programmable multistage networks each stage can be made programmable and reconfigurable structures. The programmable network enhances the switching capacity of the network and also reduces the blocking probability. Both these advantages have been obtained as the source subscribers are programmed for different possible route to overcome the problem of congestion. The network security has also been an issue, when the data is transferred over long distance [2]. Dual Tone Multi Frequency (DTMF) signaling of multistage networks and Time Division Multiplexing (TDM) techniques are surveyed [41].The papers discussed here presents different security algorithm. It is found that TACIT network security can be integrated with programmable chip. The detailed survey on these issues is discussed sequentially, with the support of the papers described here.

2.2.1 Andreas Hansson, Kees Goossens and Andrei Radulescu "A Unified Approach to Mapping and Routing on a Network-on-Chip for Both Best Effort and Guaranteed Service Traffic" Hindawi Publishing Corporation VLSI Design Volume 2007, pp (1-16)

In this paper the problem of mapping cores onto any given NoC topology and statically route the communication between these cores is considered. Group presented the UnMappable Read Architecture (UMARS+) algorithm which integrates the spatial mapping of cores, three resource allocation phases, spatial routing of communication and TDM time slot assignment. As the main contribution they have shown that how the mapping can be fully incorporated in path selection. This allows the formulation of a single consistent objective function that is used throughout all allocation phases. They show how the pruning and the cost metric used in path selection can be extended beyond one channel to capture the nature of virtual circuits. By the incorporation of the traversed path in cost calculations, they derived a metric that reflects how suitable a channel is when used after the channels already traversed. They have shown how a highly flexible turn prohibition algorithm can be used to provide maximum addictiveness in routing of best effort flows. The proposed algorithm is based on the prohibitions on residual resources such that best effort flows can use what is not required by guaranteed service flows. The time complexity of UMARS+ is low and experimental results indicate a run time only 20% higher than that of path selection alone. They applied the algorithm to a Moving Picture Expert Group (MPEG) decoder SoC, improving area 33%, power dissipation 35% and worst-case latency by a factor of four over a traditional waterfall approach.

2.2.2 Hiroaki Morino Thai Thach Bao Nguyen Hoaison Hitoshi Aida Tadao Saito "A Scalable Multistage Packet Switch for Terabit IP Router Based on Deflection Routing and Shortest Path Routing" © 2002 IEEE, IEEE Xplorer, pp (2179-2185)

The paper presents a new scalable multistage packet switch using deflection routing and shortest path routing multistage network. Deflection routing multistage network have advantage of hardware simplicity since switch element has no memory relating to buffers, and variable length packet switching can be easily handled. Moreover, in the methodology proposed by the author, new interconnection method between switch elements, required amount of hardware is reduced compared with conventional switch based on the deflection routing principle. A circuit of 8 x 8 variable length packet switch elements is designed on FPGA, and required amount of hardware to realize a 64 x 64 multistage network is calculated. It is shown that 64 x 64 switches will be implemented within one VLSI chip, and that 10 Tbps is switch is realized by two stage interconnection of the VLSI chips. Multistage network consists of simple deflection routing crossbar switch with no buffer memory inside. Hardware simplicity is the main advantage of the method. Xilinx simulation results show that the proposed method can reduce required amount of hardware of multistage network compared with conventional closed loop shuffle out switch by about 10% under condition that achieving packet loss rate of 5×10^{-7}, for 60% offered load in 64 x 64 switch. Circuits of two types of switch elements are floor planed on FPGA device, Xilinx XCV2600E, and they obtained results that needed number of gates for 8 x 8 switch elements is about 27000 gates. For 12 x 8 switch elements, the numbers of logic gates are 40000.

2.2.3 Paolo Meloni, Igor Loi, Federico Angiolini, Salvatore Carta,“Area and Power Modeling for Networks-on-Chip with Layout Awareness” Hindawi Publishing Corporation VLSI Design, Volume 2007, pp (1-12)

The paper presented a methodology for characterization of NoC switch area and power requirements. The approach, which they have proposed, was based on thorough parameterization on several architectural, deployment, and runtime variables. The area and power models for the Xpipes case study turn out to be very accurate within the limits allowed by the nonidealities of synthesis tools, even when applied to a whole NoC topology with irregular traffic flows. Their experiments show that, at least at the 0.13 µm node, applying the methodology to netlist level devices yields an acceptable approximation of the actual behavior even after placement and routing, but that even greater precision can be achieved, if desired, by applying the same technique at the layout level. They also discussed the tradeoff among accuracy and modeling effort exists, namely, coefficients can be extracted based on a single device instances, by normalization of the synthesis report, or on several of them, by an interpolation process. In their case they have chosen, the rectangular switches to add a smaller amount of information to the training set. For example, when studying the power consumption, a rectangular switch is by design unable to simultaneously feature traffic flows on all of its input and output ports as shown in figure 2.1, and is therefore behaving similarly to a square switch of smaller cardinality. Their preliminary internal testing confirms this property, at least for the Xpipes NoC. Therefore, they simply choose the npi and npo axes for the generation of the training set, and only include 4 × 4, 10 × 10, 16 × 16, and 20 × 20 instances. The work was carried on Xilinx tools and they compared the letlist results.

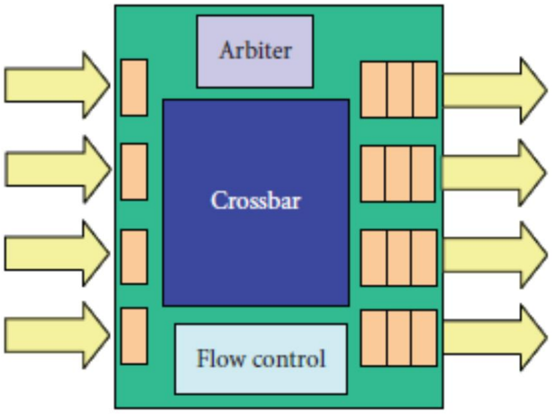


Fig. 2.1 4 x 4 pipes with switch architecture [75]

2.2.4 Ben Soh, Hien Phan, Raghu "A Four Stage Design Approach Towards Securing a Vehicular Ad Hoc Networks Architecture" Fifth IEEE International Symposium on Electronic Design, Test & Applications, Australia, IEEE Computer Society, 2010, pp (177- 282)

In this paper the authors proposed a four stage design approach towards securing Vehicular Ad hoc Networks (VANET) architecture with an improved Public Key Infrastructure (PKI) structure. The new structures define by PKI, helps in whilst achieving the security alongside and keeping the users autonomous. The communication between the central certificate authorities is minimized by employing self authorization by the users. It can be attained by self generation of pseudonyms. The scheme discussed here, will help in providing the security to users when not in coverage with the central certificate authority. The discussed paper also has proposed an efficient way of deploying Certificate Revocation Units (CRL), during revocation scheme which employs car to car forwarding of CRL along with the Road Side Units (RSU). A malicious node is detected only when it is traced back. The Certificate Authority (CA) has to follow back the real holder of a pair of certified pseudonym. Based on the certification issued to CA to work, it is pre determined that the pseudonym and the certificate pair are traceable to the CA. As a node is detected to the CA i.e. to attain the master certificate, it is not denying its authorship later. As soon as the malicious node is detected, the next step for the CA to undertake is to revoke it and distribute this information to every other node. The conclusion of the based is based on that NoC structure is helpful in guiding the mesh topology with the help of their route and address generation scheme to forward the data packets.

2.2.5 Najla Alfaraj, Yang Xu, H. Jonathan Chao "A Practical and Scalable Congestion Control Scheme for High-Performance Multistage Buffered Switches" IEEE 13th international conference on high performance routing and switching 2012, pp (44-52)

The paper explains multistage buffered switches architectures scheme for congestion control which are widely used by the industry such as Cisco, CRS series, Juniper's ex series and Broadcom's switch chip sets. They have proposed Hotspot Prevention (HOPE), which is an effective congestion control scheme, used in the 3 stages close NoC. HOPE proactively regulates traffic destined for each output by estimating the number of their backlogged packets in the network and applying a simple stop and go mechanism to prevent hotspot traffic from jamming the internal links between the stages. In the NoC architecture, the effectiveness of HOPE has motivated the authors to apply it in the multistage buffered switches. The SMs in a multistage buffered switch are separated from each other for a distance up to 100 m, which are different from a NoC, where Switch Modules (SMs) are all on the single chip. The hardware complexity of HOPE can be significantly increased. In the paper, they addressed the implementation challenges when applying HOPE in the 3 stages close network switch. In the particular network case, they proposed a scalable traffic measurement mechanism to approximate the backlogged traffic for each output port by taking advantage of the property of close network that traffic is evenly distributed among central SMs. They have addressed the problem in NoC area, and proposed HOPE [18], an effective and scalable congestion control for the 3 stages close NoC. HOPE monitors and regulates the number of back logged packets in the switch for each output port. The effectiveness and robustness of HOPE have motivated the authors to apply it in the multistage buffered switch. The implementation in NoC was less complicated compared to multistage buffered structure.

2.2.6 Ganghee Lee, Kiyoung Choi, and Nikil D. Dutt, "Mapping Multi-Domain Applications onto Coarse Grained Reconfigurable Architectures" IEEE Transactions on Computer Aided Design on Integrated Circuits and Systems, Vol. 30, No.5, May 2011, pp (637-650)

The paper explains that Coarse Grained Reconfigurable Architectures (CGRAs) have drawn increasing attention due to their performance and flexibility. Moreover, the applications of them have been restricted to domains based on integer arithmetic since typical CGRAs support only integer arithmetic or logical operations.

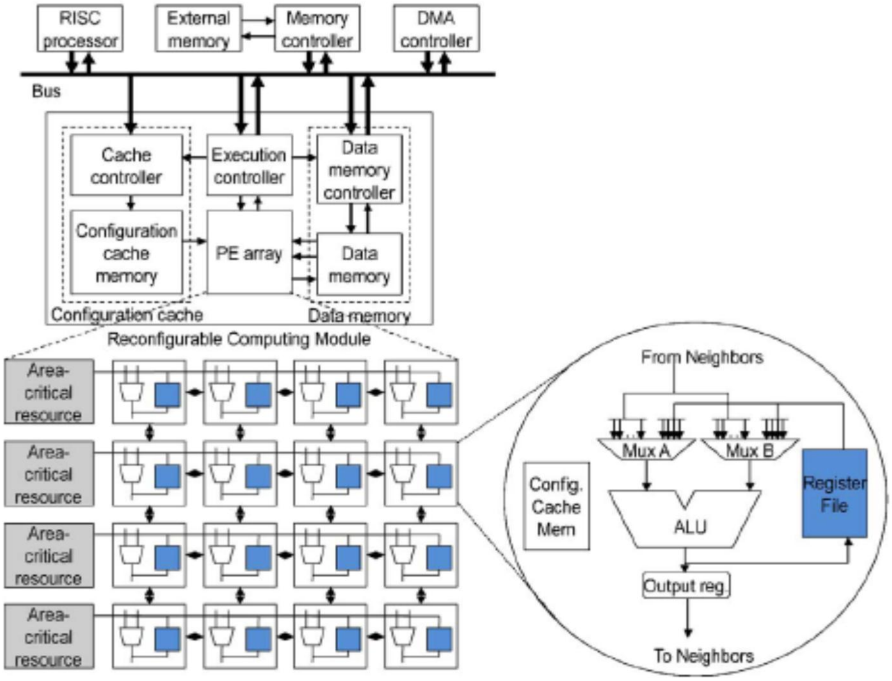


Fig 2.2 Example of reconfigurable structure [32]

The paper introduces approaches to mapping applications onto CGRAs supporting both integer and floating point arithmetic. An optimal formulation is presented using integer linear integrated programming technique. It can present a fast algorithm named Heuristic mapping algorithm. The example of reconfigurable structure is shown in figure 2.2, in which many processing elements are configured in queue as area critical resources. The experiments carried out by them are on randomly generated examples that generate optimal mapping results using Heuristic algorithm for 97% of the examples within a few seconds. It has been observed similar results for practical examples from multimedia and 3D graphics benchmarks. The developed chips and applications mapped on a CGRA show up to 120 times performance improvement compared to software implementations, helpful in demonstrating the potential for application acceleration on CGRAs supporting floating-point operation. The target architecture consists of a Reconfigurable Computing Module (RCM) [17] for executing loop kernel code segments and a general purpose processor for controlling the RCM, and these units are connected with a shared bus. The RCM used in the platform consists of an array of Processing Elements (PEs), several sets of data memory, and a configuration cache memory. CGRA is containing a 4 × 4 reconfigurable array of PEs. Buses are also shared by the PEs like shared functional units. These two sets of memory are used for double buffering. The configuration cache consists of Cache Elements (CEs), in the form of an array of the same size as the array of PEs, i.e., it has an M number of PEs in a column by N number of PEs in a row array of CEs. Each CE has several layers, so the corresponding PE can be reconfigured independently with different contexts. It is noticed that the area critical resources shared by the PEs in the same row are activated through the individual PEs and, thus, need not be explicitly considered for the modeling of the CEs.

2.2.7 Muhammad Aqeel Wahlah, Kees Goossens, "A test methodology for the non-intrusive online testing of FPGA with hardwired network on chip" Microprocessors and Microsystems, Elsevier (2012), pp (1-18)

In the paper, authors have proposed an online test methodology that uses hardwired network on chip as test access mechanism, helpful to conduct tests on a region wise basis. The methodology used to test on it, exhibits a non-intrusive behaviour that means it does not affect the applications on FPGA regions in terms of network configuration, model, programming, and execution. The methodology used for it, possesses approx, 32 times lower fault detection latency as compared to existing schemes, respectively. They presented an online test methodology for FPGAs applied for Hardwired Network on Chip (HWNoC) as test access mechanism. The online test scheme used by them, ensure the non-intrusive behaviour by: (i) invoking test at application startup time (ii) not allowing disrupted execution for already existing applications, and (iii) not restricting the parallel operations of dynamic reconfiguration and test for multiple applications. The authors analyzed the performance and cost of our test methodology for different Test Configurations Functional Regions (TCFRs) of FPGA architecture. The largest TCFR area was 348 MCRs (5568 CLBs) and the smallest one was with 44 Minimum Configuration Regions (MCRs) (704 CLBs). They were able to detect faults in the largest TCFR in 28.8 ms and at the cost of temporal overhead of 0.021 ms and spatial overhead of 82.4 CLBs.

2.2.8 Hyung Gye Lee and Naehyuck Chang, Umit Y. Ogras and Radu Marculescu "On-Chip Communication Architecture Exploration: A Quantitative Evaluation of Point-to-Point, Bus, and Network-on-Chip Approaches" ACM Transactions on Design Automation of Electronic Systems, Vol. 12, No. 3, August 2007, pp (1-20)

The article presents a comprehensive performance evaluation of three on-chip communication architectures targeting multimedia applications. The networks considered were compared and contrast the NoC Point to Point (P2P) and bus-based communication architectures in terms of performance, area, and energy consumption. They focused their implementation for bus, P2P, and NoC based implementations of a real time multimedia application for MPEG-2 encoder targeted as FPGA prototype XC2V3000 and actual video clips, instead of simulation and synthetic workloads. The author concluded that the NoC architecture scales very well in terms of performance, area, energy, and design effort, but P2P and bus-based architectures scale poorly on all accounts except for performance and area, respectively. The performance of the NoC based implementation is found very close to the P2P for the same application. Apart from it, the scalability analysis is based on duplicating the bottleneck module in the MPEG 2 design concludes that the performance of the NoC design scales as well as the P2P. Bus based communication and their FPGA implementation scales much more poorly. If NoCs, are analyzed in terms of area, they scales as well as the bus-based implementation. The P2P implementation does not scale well due to the overhead involved in redesigning the interfaces. In terms of space, NoC scales as well as the bus based implementation. Moreover, the design implementations used for adding new cores to an existing design is much smaller for the NoC case as compared to P2P. The energy consumption of the NoC based architecture and implementation is estimated much smaller than both P2P and bus-based implementations and it scales much quality and better performances with the number of extra modules added to the base design.

2.2.9 Luca Benini, Giovanni De Micheli Networks on Chips "A New SoC Paradigm, IEEE Computer Society, SoC Design", January (2002) pp (70-79)

The paper explains Scalable Programmable Integrated Network (SPIN) on-chip micro network defines packets as sequences of 32 bits words, with the packet header fitting in the first word. The network uses a byte in the header to identify the destination address, which allows the network to scale up to 256 terminal nodes. The routing information carry packet tagging, and the packet payload can be of variable size. A trailer which does not having data, use a checksum for error detection, terminates every packet. SPIN has a packetization overhead of two words. The payload should be significantly larger than two words to amortize the overhead. On chip networks relate closely to interconnection networks for high performance parallel computers with multiple processors, in which each processor is an individual chip. The networks based on multiprocessor interconnection networks, source and destination nodes are physically closed to each other and have high link reliability. Multiprocessors have traditionally designed using multiprocessor interconnections under stringent bandwidth and latency constraints to support effective parallelization. The paper explains hybrid network, multistage networks, direct, indirect networks and micro control networks. SoC have arbitration of communication among nodes, reliability and better performance.

2.2.10 Vasilis F. Pavlidis, Eby G. Friedman "3D Topologies for Networks-on-Chip" IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 15, No. 10, October 2007 pp (1081-1091)

The paper explains the chip integration of 2D and 3D NoC topological structures. NoC Integration in the third dimension introduces a variety of topological structures choices for configurable NoCs. In 3D NoC, each Processing Elements (PE) is on a single axis but possibly different physical plane (3D IC–2D NoC). By the chip integration point of view, a PE can be implemented on only one of the physical planes of the system. Therefore, each 3D NoC contains PEs on every physical planes such that the total number of nodes is $N = n_1 \times n_2 \times n_3$, where n_1, n_2, n_3 are the number of nodes in X, Y and Z directions. The different configurations of 2D and 3D NoC are shown in figure 2.3. A 3D NoC topology is proposed in figure 2.3(c), in which exists one physical plane. Each PE can be integrated in multiple planes, as (2D IC-3D NoC). Based on this integration, It is possible to form a hybrid 3D NoC. In such a NoC system, both the interconnect network and the PEs can span more than one physical plane of the stack (3D IC - 3D NoC). The paper emphasized on the research work towards latency expressions for each of the NoC topologies, with the assumption of a zero load model. The speed and power consumption of 3D NoC are compared to that of 2D NoC, their physical constraints, such as the maximum number of planes that can be vertically stacked and the asymmetry between the horizontal and vertical communication channels of the network. An analysis is done on their performance and analytic model for the zero load latency of individual network that considers the effects of the topology on the performance of 3D NoC is also developed. 3D - 3D network provides zero latency but for crossbar structure, 2D - 3D NoC structure can be preferred for a larger network where power and delay are major constraints.

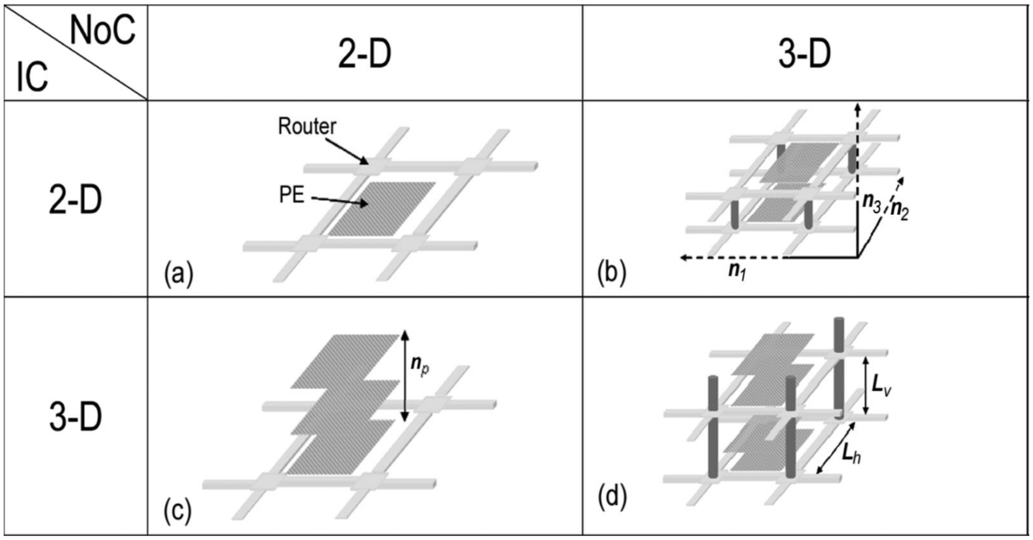


Fig. 2.3 Various NoC topologies [98] (not to scale) (a) 2D IC–2D NoC. (b) 2D IC–3D NoC. (c) 3D IC–2D NoC. (d) 3D IC–3D NoC.

2.2.11 Mike Santarini “FPGA Command Centre Stage in Next Gen Wired Networks” XCell Journal Today, Vol.1 issue 67, (2009), pp (10-14)

The paper explains two types of wired network, one for computing and the second for telecommunications. Wired networks have been separate, their own set of unique protocols, bandwidth requirements routing equipment and rate of bandwidth growth. In telecom industry bandwidth requirement has increased four times (2.5 Gbps to 10 Gbps), and now moving to 40 Gbps). On the other hand, computer networking has done the job in leaps of 10x (100M, 10G, and 100 G). Therefore, a distinguished engineer Goron Brebner from Xilinx noted that during the last wired network retooling a few years ago, a convergence of sorts took place at 10 Gbps. The physical signaling for Ethernet has been converged with signaling for telecom as both network types independently increased their top bandwidth rates. In telecommunication, a line card is a combination of a series of dedicated Network Processor Units (NPUs), a CPU and a number of the highest speed FPGAs available. As a packet arrives at a line card, an FPGA processes the raw data into formats that a given router can read. The NPUs are coordinated by the processor to read and route data, while the FPGAs facilitate some of the communication between the CPU and the NPUs. Future generation wired networks will be transferring internet data, voice, and video simultaneously with the integration of all these into IC package. In June 2008, telecommunications giant Comcast Corp. announced, it had successfully completed a 100 Giga bit Ethernet (GE) technology test over its existing backbone infrastructure between Philadelphia and McLean, Va., using the industry’s first 100GE router interfaces. The system used the same High Speed Ethernet IP Core (HSEC) running on a Virtex-5 Full Duplex Transreceiver (FXT) FPGA board, which is supported by the Virtex-5 platform today. Virtex-5 TXT XC5VTX240T device contains 37,440 logic slices with a total of 239,616 logic cells. The Xilinx FPGA electrically transmitted all ten signals to ten 10 Gbps (Small Form Pluggable Factor) XFP optical transceivers, which converted the signals into the optical domain.

2.2.12 Dr. Rosula S.J. Reyes, Carlos M. Oppus, Jose Claro N. Monje, Noel S. Patron, Reynaldo C. Guerrero, Jovilyn Therese B. Fajardo "FPGA Implementation of a Telecommunications Trainer System" International Journal of Circuits, Systems and Signal Processing, Issue Vol, 2, 2008, pp(87-95).

The paper presents the use of FPGAs in the implementation of both analog and digital modulation that includes Amplitude Modulation (AM), Frequency Modulation (FM), Phase Modulation (PM) , Pulse Code Modulation (PCM), Pulse Width Modulation (PWM), Pulse Position Modulation (PPM), Pulse Amplitude Modulation (PAM), Delta Modulation (DM) , Amplitude Shift Keying (ASK), Frequency Shift Keying (FSK), Phase Shift Keying (PSK) , Time Division Multiplexing (TDM) and different encoding techniques like Non Return to Zero (NRZ) line code, NRZ mark line code, NRZ inversion line code, Unipolar NRZ line code, bipolar NRZ line code, alternate mark inversion line code, and Manchester line code. Moreover, the designing of FPGA can be done to emulate a particular device like an oscilloscope and a function generator. The paper describes the capability of an FPGA to internally generate a low frequency input signal and through the use of a Video Graphics Array (VGA) port, and it is capable to display the signals in an output device. The paper focuses that the use of FPGAs is not limited to the aforementioned applications because of its reconfigurability and reprogramability.

2.2.13 Aye Sandar Win "Design and Construction of Microcontroller Based Telephone Exchange System" World Academy of Science, Engineering and Technology, Vol. 46, 2008, pp (60-67)

The paper demonstrated design and construction of microcontroller based telephone exchange system, connection with Programmable Interface Controller (PIC) 16F877A and DTMF MT8870D. In telephone system, PIC16F877 microcontroller is used to control the call processing. When call is processed, different tones are generated which are dial tone, busy tone and ring tones. The paper demonstrates the eight line telephone systems with full signaling and switching functions similar to those of the central office systems. The eight telephones are connected to the switching devices and common line. In microcontroller system, PIC16F877A microcontroller is utilized to control tone, ring relay and on/off-hook switch when the telephone is used. In the designed system there is tone generator, which is used to get dial tone, ring tone and busy tones. Ringing is generated at the receiving end of the phone being called. Ring relay is used to get tone and ring processes. DTMF signaling technique is based on basis for voice communication control. Each number consists of a combination of two frequencies. DTMF decoder converts the DTMF tones to the binary numbers and sends to the microcontroller. In the signaling and switching operations, transistors and relays are used to switch audio signals and control signals and to decode the DTMF signals. These switches are controlled by powerful software procedures to be implemented.

2.2.14 K. P. Rane, S.V.Patil and A.M.Patil "Efficient combination of Electronics Switching System and VLSI technology" Proceedings of SPIT-IEEE Colloquium and International Conference, Mumbai, India pp (220-225)

The authors have applied the new approach to design high performance Hybrid Telephone Switching System (HTSS) using combination of stored program control (SPC) electronic switching system and VLSI technology. They proposed to separate total digital processor card circuit around processor into two parts. One is service handling circuit (SHC) and another is call handling circuit (CHC). For the good performance of the system, it is very important to increase the call handling speed of system. The speed of the system is increased with the use of SHC and CHC because CHC is designed to work in concurrent manner to handle individual calls of subscribers and SHC helps CHC while any service/facility is to be provided. SHC is having all features of SPC with removed call handling operations. In a general way, each line card includes driver circuit of eight lines. In the same time, different functions are incorporated on the line cards and those perform many switching functions by themselves. The sub unit CHC performs all those functions. Whenever an electronic circuit is designed, the size of the circuit is the very important factor. The authors were able to reduce CHC in to single IC. Therefore, it is possible to use reduced processor card that could be placed in any corner of line cards. They designed CHC to work with eight lines so that it can be placed with each line card. The drive r circuit is carried out using junction card for trunk lines and switching matrix establishes the connection between calling subscriber and called subscriber or between junction line and calling line. The operations relating to call handling are handled by concurrent operations using CPLD and complicated and sequential operations like services handling are handled by Microcontroller/ Microprocessor Systems (MPMC). VHDL codes are designed and tested for different test cases on call handling operations and verified using test benches are designed to act as MPMC for the testing of services like Do Not Disturb (DND), Outgoing Bar (OGB), Call Forwarding (CF), and STD Bar (STDB) facilities.

2.2.15 David Atienzaa, Federico Angiolini, Srinivasan Murali, Antonio Pullinid LucaBeninic, Giovanni De Michelia, "Network-on-Chip design and synthesis outlook" Integration The VLSI Journal Elsevier, Vol. 41, 2008, pp(340-359)

In the article, the authors have presented an overview for NoCs using a complete NoC synthesis flow, design, and a detailed scalability analysis of different NoC implementations for the latest nanometer scale technology nodes. They presented NoC based solutions for the on chip interconnects of Multiprocessor System On Chip (MPSoCs) that illustrate the benefits of competitive application specific NoCs with respect to more regular NoC topologies regarding performance, power and area. Emerging consumer applications demand a very high level of performance in the next generation of embedded devices. Therefore, a new techniques and interconnection mechanisms that can provide solutions for an efficient design of the system complex for the coming embedded architectures are greatly needed. The impact of the target frequency of operation on the area and energy consumption of an example 5 x 5 switch obtained from layout level estimates for 130 nm is presented, and energy values (in power MHz) instead of the total power, so that the inherent increase in power consumption due to an increase in frequency. The paper emphasizes the use of 3D NoC instead of 2D NoC in terms of scalability and reliability, their structure and routing. It is completely based on the review of NoC topological structures.

2.2.16 Jason Cong, Yuhui Huang, and Bo Yuan "A Tree-Based Topology Synthesis for On-Chip Network" Computer Science Department University of California, Los Angeles Los Angeles, USA, IEEE Conference proceedings, (2011)pp (650-658)

The Network on Chip (NoC) interconnect network of future multi-processor system on chip (MPSoC) needs to be efficient in terms of energy and delay. The custom on chip network, which targets a given application, has proved to be more efficient than the regular structure on-chip network design in. The reason is that the communication requirement for each data flow is available in the design time, so the packet latency and power consumption are predictable once the links of networks are determined. The problem of topology synthesis is to determine the number of routers, the location of newly added routers, and the connectivity between them. Power consumption and packet latency are two trades off factors for any Application Specific Integrated Circuit (ASIC) design which are met in the research. The paper has focused more on tree topological NoC implementation in Hardware Description Language (HDL) environment.

2.2.17 Hao Tian, Ajay K. Katangur, Jiling Zhong Yi Pan "A Novel Multistage Network Architecture with Multicast and Broadcast Capability" The Journal of Supercomputing, Springer, Vol. 35, 2006, pp (277–300)

The paper focuses about a Multistage Interconnection Network (MIN), which is composed of several stages of switch elements by which any network input port can be connected to any output port. In the paper, optical MIN is represented and its capability.

It a very important class of interconnecting schemes used for constructing optical interconnections for communication networks and multiprocessor systems. Hybrid structure is used to represent the most commonly implementation approach with guided wave technology. In the structure main basic Switching Element (SE) in hybrid optical MINs, which is also called directional coupler and is typically fabricated on Titanium Diffused Lithium Niobate (Ti:LiNbO3). MIN architecture has multicast capability, used in telecommunication switching which is constructed utilizing a modularization approach for fixed exchange pattern. It consists of an input module, two Point to Point (PTP) modules, and one or more Multicast/Broadcast (M/B) and an output module as shown in figure 2.4.

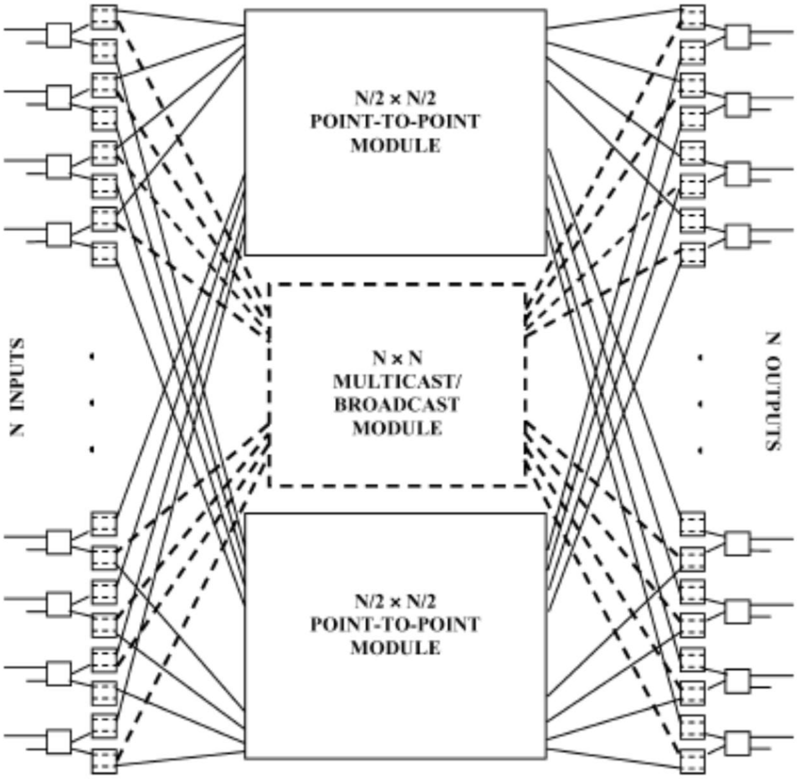


Fig. 2.4 N x N multistage network [35]

The input signal is taken by input to PTP and M/B modules, which are independent to each other. PTP can follow any MIN architecture and M/B provides multicast functions. The comparison is also carried out to indicate that this new architecture with Dilated Benes PTP module has much better performance in terms of system Signal to Noise Ratio (SNR), signal attenuation, the number of switch elements of point-to-point connections than two current multicast MIN architectures, PS/AC and Jajszczyk's networks.

2.2.18 Teijo Lehtonen, Pasi Liljeberg, and Juha Plosila "Online Reconfigurable Self Timed Links for Fault Tolerant NoC" Hindawi Publishing Corporation VLSI Design (2007), pp (1-13)

The paper focuses on fault tolerance design of the communication links in NoC architecture. They proposed link structures that have properties for tolerating efficiently transient, intermittent, and permanent errors. Transient errors can be realized using Hamming coding technique methods and interleaving for error detection and Automatic Repeat Query (ARQ) as the recovery method. Two approaches have been introduced to tackle the intermittent and permanent errors. Time redundancy can be utilized using split transmission approach, while the other structure, introduced are using spare wires, is a hardware redundancy approach. Network communication in the links was based on asynchronous 2-phase signaling and the control signals for ARQ and reconfiguration were incorporated into these control signals. The control lines are used to control the functionality of the network and protected using triple modular redundancy. The developed designs compared against reference designs and simulated. From the simulation results, it has been shown that the performance of NoC decrease when comparing to a design with ARQ, but no reconfiguration structure is larger for the split transmission design than for the spare wire design. Split transmission has latency 31%, throughput 25% and spare wire has latency 15% and throughput 10%. Moreover, the area overhead is larger for the spare wire design (105%) as for the split transmission design (75%).

2.2.19 Wen-Chung Tsai,Ying-ChernglLan,Yu-Hen Hu, and Sao-Jie Chen "Networks on Chips: Structure and Design Methodologies" Hindawi Publishing Corporation., Journal of Electrical and Computer Engineering (2012), pp (1-13).

The paper focuses the layered protocol architecture of NoC. The routing scheme of 5 x 5 crossbars NoC is also discussed in the paper. The typical architecture of a mesh NoC is shown in figure 2.5 which is a combination of multiple segments of wires, Network Interfaces (NI), and routers (R). Each interface can have either source IP or destination IP. The NoC function can be performed into several layers: application layer, transport layer, network layer, data link layer, and physical layers.

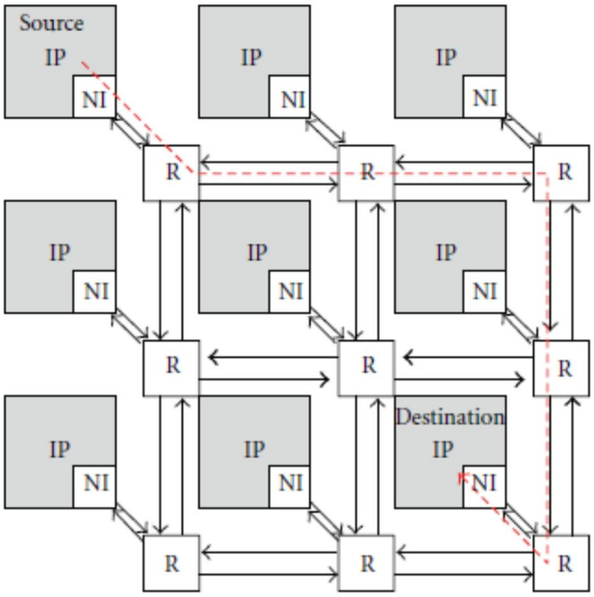


Fig.2.5 Typical NoC architecture in a mesh topology [99]

For every NoC router, it should contain both software and hardware implementations to support functionality of these layers. In the end of the paper, they proposed a novel Bidirectional Channel NoC (BiNoC) backbone architecture. It can be easily integrated into most conventional NoC designs and successfully improve the NoC performance with a reasonable cost and power.

2.2.20 Krishnan Srinivasan, “OCP-IP Network-on-chip benchmarking workgroup Erno Salminen”, Tampere University of Technology Sonics Inc. Zhonghai Lu, Royal Institute of Technology, December 2010, pp (1-5)

Multiprocessor System on Chip (MPSoC) devices integrate multiple processing elements, memories, peripherals, and off chip interfaces into a single silicon chip. It allows higher performance with reasonable power consumption, which is critical in mobile devices but also in many other embedded systems. The highest demand is of efficient parallel processing for the interconnect network that is utilized inside the chip; this is also called a Network on Chip (NoC) .The practical implementation and adoption of the NoC design paradigm faces multiple unresolved issues related to design methodology/technology and analysis of architectures, which are helpful in test strategies and dedicated Computer Aided Design (CAD) tools. Benchmarking has been a long tradition in CPU and complier design. To advance and accelerate the state of the art of the NoC paradigm research and development, the community is in the need of widely available reference benchmarks. Open Core Protocol International Partnership (OCP-IP) is dedicated to proliferating a common standard for intellectual property (IP) core interfaces, or Sockets, that facilitate “plug and play” System on Chip (SoC) design. Implementing complex SoC design more efficient for the widest audience services, OCP-IP provides the tools and services to its members that are necessary for convenient maintenance, implementation and support of the standard OCP socket interface. There are several workgroups each concentrating on a certain topic, used as socket specification, assembling, system level design, co-design, debug, and NoC benchmarking. The article described in the paper presents the goals and deliverables provided by the NoC benchmarking workgroup. The NoC implementation and OCP interface using pipelined structure and parallel processing is an extensive research in the implantation of NoC benchmarking.

2.2.21 Aurel A. Lazar, "Programming Telecommunication Networks" IEEE Network, September 1997, pp (8-19)

In the paper authors have discussed the realization of an open programmable networking environment based on a new service architecture for advanced telecommunication services that overcomes the limitations of the existing networks. The paper investigating a model will help to clarify some of the pertinent issues confronting the telecommunications service industry today as it comes of age. The paper focuses the programmable switch implementation of telecommunication network. They address some of the important QoS, performance, scalability, and implementation issues, fully aware that our work has opened new vistas that call for additional research. The paper exploits the advantages offered by IP and ATM technologies without necessarily suffering their shortcomings. The need for investigating scaling issues through the emulation of complex service scenarios arising in large scale broadband networks is much focused in the research paper. The research paper is a start in the era of programmable telecommunication network and their reconfigurable structures.

2.2.22 Xinmiao Zhang, and Keshab K. Parhi "High-Speed VLSI Architectures for the AES Algorithm", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 12, No. 9, Sep. 2004, pp (957-968)

The paper presents novel high speed architecture method for the hardware implementation of the Advanced Encryption Standard (AES) algorithm used for network security. The paper is an extensive research towards sub pipelined architecture in network security. In order to explore the advantage of sub pipelining further, data is implemented by a combinational logic to avoid the unbreakable delay of Look up Tables (LUTs) in the traditional designs. Encryptor and decryptor both can enjoy of 128 bits of block size at encryption and decryption end respectively, but the key size is possible to keep up to 128, 192 and 256 bits respectively. Using the proposed methodology, a fully sub pipelined encryptor with 7 sub stages in each round unit can achieve a throughput of 21.56 Gbps on a Xilinx XCV1000 e-8 Bg 560 device in non feedback modes, which is faster and is 79% more efficient in terms of equivalent throughput slice than the fastest previous FPGA implementation known to date.

2.2.23 Prosanta Gope, Ashwani Sharma Ajit Singh Nikhil Pahwa "An Efficient Cryptographic Approach for Secure Policy Based Routing (TACIT

Encryption Technique)", Conference Proceedings, IEEE Xplorer, (2011), pp (359-363)

High performance internetworks need the freedom to implement packet forwarding and routing to their own defined policies in a way that goes beyond traditional routing protocol concerns, but there are some administrative issues dictate the traffic be routed through specific path. With the help of policy based routing users can implement policies that selectively cause packets to take different paths. In a staged network it is much needed to route secured data towards destination. Therefore, the secured policy routing methods are applied. The paper focuses compares the different methods available for network security such as DES, Triple DES, AES, Blowfish, RC4, Modes, X-Modes, but these methods are limited to block size and key size. Maximum key size is supported by AES algorithm of 256 bits. In the paper, authors have proposed a new security algorithm, TACIT encryption and decryption, which can be applicable to any network. The greatest advantage of the algorithm is that it can have 'n' bits block size and 'n' bits key size. The hardware chip implementation of the TACIT network security is the future work proposed by the authors and it can provide the best results if key size is considered greater than the block size. The algorithm has been tested on the different text files. It can be implemented in C, C++, C# and Java programming languages.

2.2.24 Nikos Sklavos, Alexabdros Papakonstinou, Spyros Theoharis Odysseas Koufopavlou, "Low-power Implementation of an Encryption/Decryption

System with Asynchronous Techniques", VLSI Design, Taylor and Francis 2002 Vol. 15 (1), pp. (455–468)

The paper has focused, a low power VLSI block encryption system design and implementation. The choice of International Data Encryption Algorithm (IDEA) as the encryption/decryption algorithm ensures the strength of the data encryption operation. The paper presented the two implementations of the system which are synchronous and asynchronous. The only know fast single chip implementation is synchronous design. The synchronous chip has a power consumption of the 1.25 W, while the chip designed by the authors for synchronous version has a power consumption of 58 mW and our asynchronous 41.25 mW in the worst cases of operation. Moreover, their synchronous design has low power dissipation, and the asynchronous design has significantly very low power consumption. Apart from it, with the second implementation of asynchronous design of the encryption/decryption system the total power dissipation is decreased at about 20–40 % in percentage units. The same integrated circuit can be applied as a very fast and low power encryption/decryption device in high speed networks such as multistage telecommunication networks or 2D and 3D NoC.

2.2.25 Nikos Chrysos, Lydia Y. Chen, Cyriel Minkenberg, Christoforos Kachrismand Manolis Katevenis "End-to-end congestion management for non-blocking multistage switching fabrics" ACM digital Library, (2010), pp (1-2)

Scalable network routers and high performance computer interconnects are encountered as packet switched networks. As these networks scale to larger port counts, and their utilization increases, helpful in congestion management becomes indispensable. There are some technology constraints rule out monolithic buffer less switches with centralized schedulers, and proposed buffered multistage switching fabrics with distributed control. The controlling of multistage network is discussed in the paper. There is an arbiter, located in central scheduling unit and it can be distributed across switches and fibers. The network was considered of 64 x 64 size, three stage network. The authors depicted the average delay of packets = 2.5 x bursty arrival time and each request can be granted under 128 segments of Virtual Output Queue (VOQ).

2.3 RESEARCH GAPS

As seen from the literature survey that the various articles have presented the concept of multistage networks, NoC structures, routing schemes. It also describes HOPE, VANET, UMARS+, Xpipes NoC, 2D-3D NoC switching, buffering techniques and scalable structures. In multistage networks there is the research gap in the implementation of programmable switch. The research can be focused in the implementation of programmable reconfigurable structure with the integration of multiplexing, routing, signaling and NoC topologies. TACIT is the only security algorithm that can have ‘n’ bits block size and ‘n’ bits key value. All other existing algorithms such are limited to 128 bits block size and 256 bits of key size. TACIT algorithm is not integrated yet with any network, as a hardware chip. Mesh topology is the best topology for the scalable network because it has the maximum routing in comparison to other existing topologies and the integration of the topology with 2D and 3D network can enhance the performance of the network. In the multistage network the congestion can be controlled using programmable switches such as programmable three stage network, programmable four stage network programmable five stage network. As an example, in India, Bharat Sanchar Nigam Limited (BSNL) exchanges are following three stage digital network structures which are not programmable. Programmable structures means entire exchange environment is configured using FPGA or ASIC. For a good system the call handling capacity should be more. The operations relating to call handling are handled by concurrent operations using CPLD and complicated and sequential operations like services handling are handled by Microcontroller/ Microprocessor Systems (MPMC). In traditional Telephone switching systems, it is not possible to increase the extension or junction lines because of limitation of processor or controlling system. Designing the system with higher processors is then not cost effective and is complicated too. Solution for this problem is to implement the controlling system for eight lines in a FPGA and make provision to co-control or cascade to the other FPGA of another eight lines so that it can have as many as possible lines. It is also possible because available FPGAs are not much costly but the programmable switching system is a challenge. Programmable multistage networks like four and five stage will enhance the switching capacity of the network and reduce the blocking probability. By the literature survey, it is clear that multistage network concept of four stage is also utilized in optical communication. Therefore, there is an ubiquitous requirement of development, study and reconfiguration of IP based multistage telecommunication system

CHAPTER SUMMARY

The chapter explains the literature carried out the findings of various research papers support to carry out the research work on multistage crossbar NoC used for telecommunication switching and reconfigurable programmable NoC. The most important reasons for using NoC architectures are their promise for scalability and programmable network capability. Telecommunication traffic characteristics have been long recognized as playing a major part in multicore systems design. The traffic is passed by multistage network and packets are routed with shortest path under maximum available network. These effects have important consequences for the design of on-chip multimedia systems since self-similar processes have properties which are completely different from traditional short range dependent or Markovian processes that have been traditionally used in system-level analyses. From the literature review, it has been subsequently reported that even the traffic generated by programmable cores consists of multiple program phases. Moreover, the research in this area is behind and lacking due to NoC benchmarks. There are two reasons. First reason is that the applications suitable for NoC platforms are typically very complex. It is common for applications to be partitioned among tens of processes or more in order to allow for evaluations of scheduling, portioning and mapping, etc. Some general purpose Chip Multiprocessors (CMPs) are originally designed for shared memory multiprocessors that can be used to share the communication among processor and I/Os. Second, with the comparison to traditional research areas like physical reason design, where the design constraints are static, the NoC research requires detailed information about the dynamic behavior of the system, which is very hard to obtain even using detailed simulation or prototyping. As a consequence result, most researchers and designers still rely on synthetic traffic patterns such as uniform random, scheduling, bit-permutation traffic, to stress test a network design. The network design of the telecommunication depends on multistage environment. It can follow the two stage, three stage, four stage and five stage structures. The ability of the network to efficiently disseminate information depends largely on the underlying topology. The simplicity and regularity of mesh structures makes design approaches based on such a modular topologies very attractive and much applicable in telecommunication network, the data routed in it is in packet form. Performance analysis of networks largely depends on various simplifying assumptions on the network or traffic characteristics such as uniform traffic vs. bursty traffic and typically assumes deterministic routing due to the difficulty in handling the more general problem. NoC architecture follow mesh, tree, torus and hierarchical structure, depends on the applications is under design. A flexible FPGA based NoC design that consists of processors and reconfigurable components can be integrated into a single NoC chip. The blocking probability, switching elements, switching capacity of the crossbar multistage telecommunication network can be optimized using reconfigurable NoC. The NoC structure can be synthesized on Xilinx supporting FPGA, used to implement programmable multistage network.

CHAPTER-3

MULTISTAGE NETWORKS

The chapter describes the need of multistage networks over single stage network. Multistage networks are realized in two stage, three stage, four stage and five stage telecommunication networks. The number of crosspoints, links, switching capacity is calculated for each staged network. The routing scheme of each staged network is discussed with maximum possible routes, to estimate full connected and full available network.

3.1 INTRODUCTION TO MULTISTAGE NETWORKS

In the staged network an inlet is connected directly to an outlet through the single crosspoint. In single stage network [42] each individual cross point can be used to connect to inlet and outlet pair. Therefore, the possible number of inlet and outlet pairs is equal to N (N-1)/2 in a triangular arrays and N (N-1) for a square array [91]. For a large single stage network, the number of crosspoint switches is prohibitive and a specific crosspoint is needed for specific connection. The data is mostly circulated a number of times in the network, if source subscriber want to establish a connection with destination subscriber. The major problem in the single stage network is the connection establishment between source and destination, no availability of data, if any crosspoint fails. There is no alternate path [6, 15] to route the call towards destination. The association of large numbers of crosspoints on inlets and outlets leads to capacitive loading on the message path. For a single stage larger exchange crosspoints are very insufficiently utilized [6]. As an example, in a square switch, only one crosspoint is used in each column or row, but all lines are active. To enhance the switching capacity and efficiency of telephone exchange, it is necessary that any crosspoint can be usable for more than one potential connection. If particular inlets want to communicate to outlet, there is more than one path or alternate paths are available to establish the connection and crosspoints are to be shared, so that blocking will not occur. Alternate paths reduce the blocking probability and also protect the exchange to be failure. Sharing of crosspoints for alternate paths through the switch can be overcome by considering the concept of multistage switching networks.

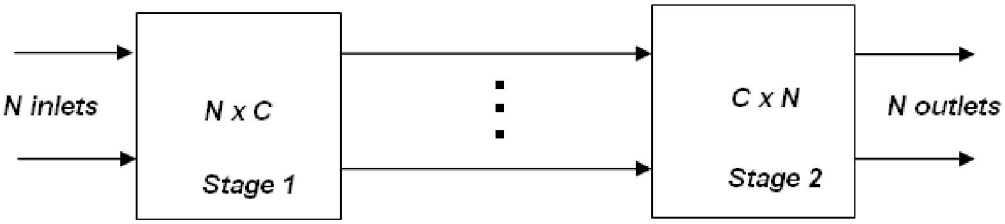


Fig 3.1 $N \times N$ network representation using two stage of $N \times C$ and $C \times N$ [91]

A single stage network can be configured using equivalent multistage networks [42, 46]. A single stage $N \times N$ network having the switching capacity of C connections can be realized by a two stage network of $N \times C$ and $C \times N$ stages as shown in the figure 3.1. Each connection requires two switching elements. In first stage, any of the N inlets can be connected to any of the C outputs. Similarly, in second stage, *C* inputs of the second stage can be connected to any of the *N* outlets. So, there are *C* alternative paths to establish connections between inlet/outlet. In such a case, each stage is having *NC* switching elements and the network is said full available or full connective network.

3.2 TWO STAGE NETWORK

Considering a two stage network where 8 inlets can communicate with 8 outlets as shown in figure 3.2(a). Eight inlets and outlets are configured as (000) USER, (001) USER 1, (010) USER 2, (011) USER 3, (100) USER 4, (101) USER 5, (110) USER 6, (111) USER7, can communicate with each other in full duplex mode. Inlets [(000)….. (111)] can communicate with any outlets [(000)….. (111)]. There is only one dedicated path, for Inlet (000) to communicate with any output subscriber, as shown in figure 3.2(b)

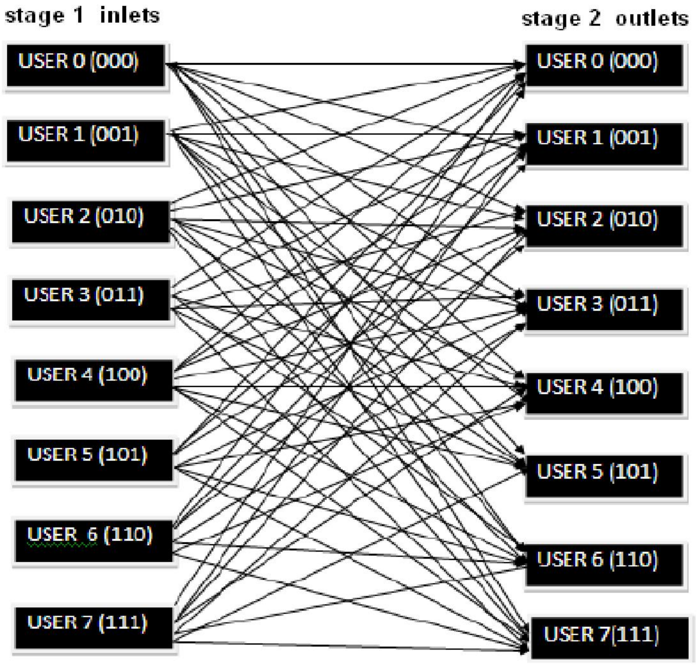


Fig 3.2(a) Dedicated paths for two stage switching network (8 x 8)

Similarly, if any other input subscriber wants to communicate with any destination subscriber, there is a single path to connect or vice versa. If any link fails, there is no communication between the particular inlet and outlet. For a larger exchange, if the inlets and outlets are more, the numbers of associated crosspoints are also large. Therefore, the methodology is adopted to reduce the number of crosspoints and to enhance the switching capacity of the switching system. Dividing the architecture into smaller sized blocks, is called switching matrices. The same architecture of 8 x 8 configurations is divided into 4 blocks at inlet and outlets having 2 input and output subscribers associated with each block.

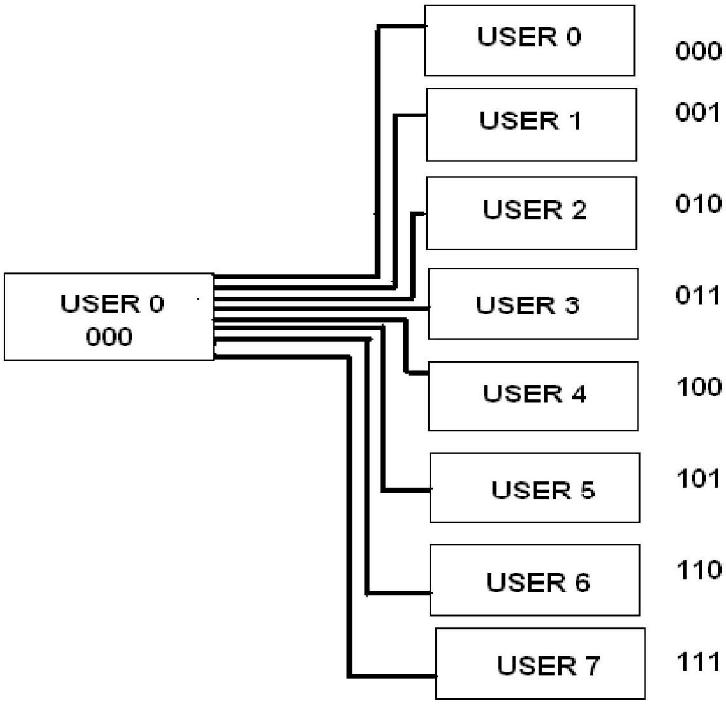


Fig 3.2(b) Routing of one inlet to all outlets

The generic architecture of two stage network with *M* inlets and *N* outlets *(M* × *N)* is shown in the figure 3.3(a) [91]. *M* inlets are divided into *a* blocks having *x* inlets each block such that *M* = *x.a.* Similarly, the outlets are divided into b clock having y outlets with each block such that $N = y.b$. In the first stage the network configuration is $x \times b$, and in second stage the network configuration is $a \times y$, under full availability of the network. Total number of switching elements can be calculated

$$S = xba + yab \qquad \text{Eqn. (3.1)}$$

Substituting the values of M and N, $M = x.a$ and $N = by$. Total numbers of switching elements are given by

$$S = Mb + Na \qquad \text{Eqn. (3.2)}$$

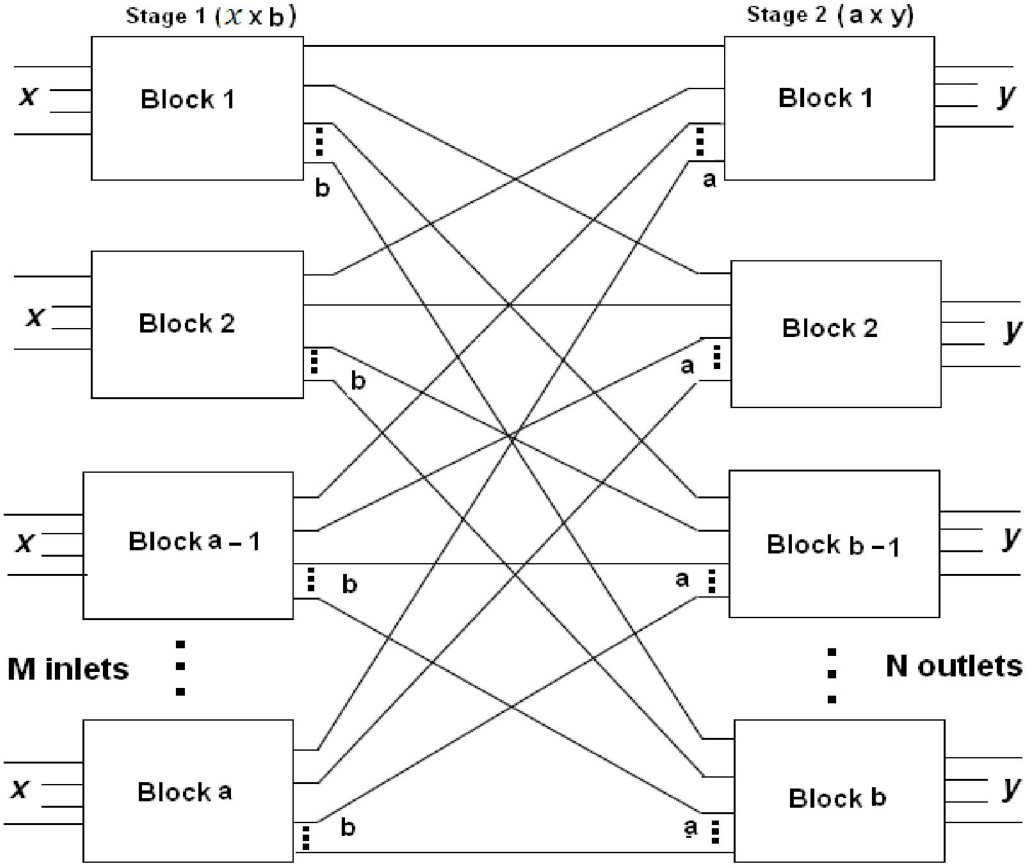


Fig 3.3(a) Two stage generic network with blocks [91]

Switching capacity of the network can be found with the possible number of links between first and second stage and ensures the maximum numbers of simultaneous calls supported by the network.

$$Switching\ Capacity = ab \qquad \text{Eqn. (3.3)}$$

As a particular case, 8 × 8 two stage network configuration is shown in figure 3.3 (b). Here, M = 8 inlets and N = 8 outlets are divided into a = b = 4 blocks with x = y = 2 inlets and outlets with each block. Here the inlets are outlets can be structured as

$$\mathrm{M} = \begin{bmatrix} \{\mathrm{M}_0(000), \mathrm{M}_1(001)\}, \\ \{\mathrm{M}_2(010), \mathrm{M}_3(011)\}, \\ \{\mathrm{M}_4(100), \mathrm{M}_5(101)\}, \\ \{\mathrm{M}_6\ (110), \mathrm{M}_7\ (111)\} \end{bmatrix}$$

$$\mathrm{N} = \begin{bmatrix} \{\mathrm{N}_0(000), \mathrm{N}_1(001)\}, \\ \{\mathrm{N}_2(010), \mathrm{N}_3(011)\}, \\ \{\mathrm{N}_4(100), \mathrm{N}_5(101)\}, \\ \{\mathrm{N}_6\ (110), \mathrm{N}_7\ (111)\} \end{bmatrix}$$

The routing scheme of two stage network is shown in table 1. In two stage network the number of crosspoints between the inlets and outlets are reduced but the problem is associated with the utilization of common link in a particular block. In such a case in block 1, if M_0 is already communicating with either N_0 or N_1, then M_1 cannot communicate with N_1 or N_0 at the same time because the path $M_0\ (000) \rightarrow X_0 \rightarrow A \rightarrow A \rightarrow X_0 \rightarrow N_0\ (000)$ or $M_0(000) \rightarrow X_0 \rightarrow A \rightarrow A \rightarrow X_1 \rightarrow N_1(001)$ is already busy. Similarly if M_1 is already busy with either N_0 or N_1 and M_0 want to communicate with N_1 or N_0, the connection will not establish among them.

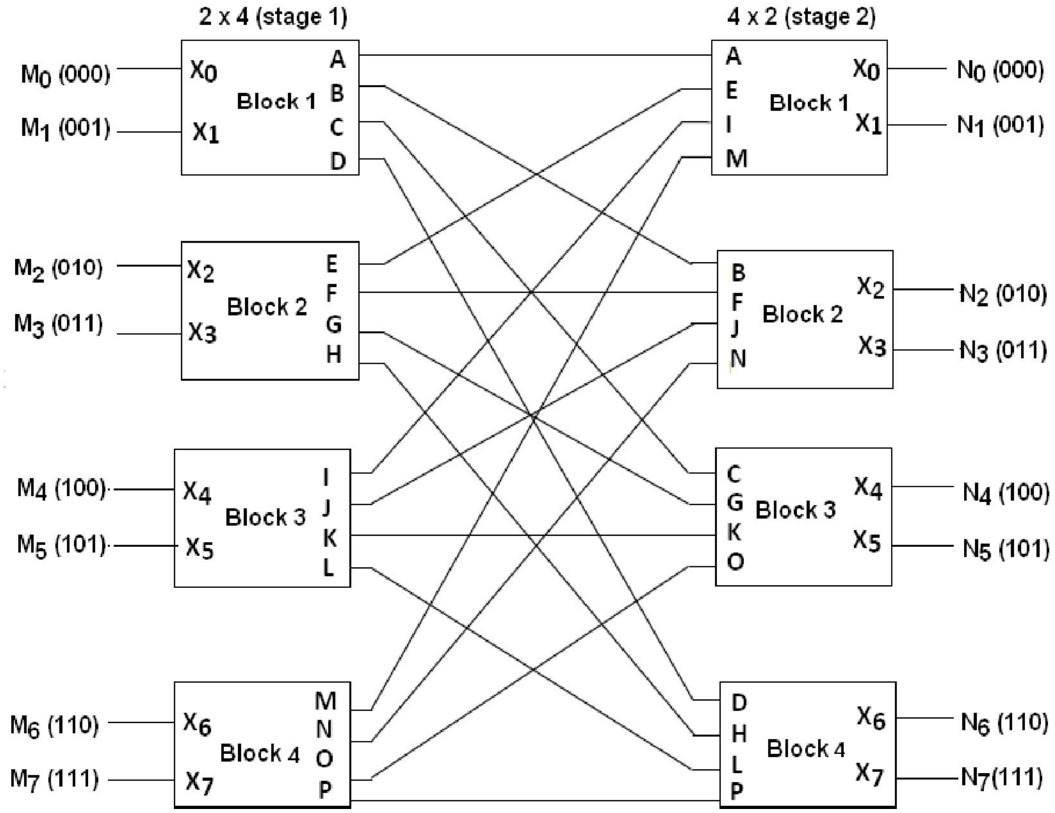


Fig 3.3(b) Two stage generic network (8 x 8)

Table 3.1 Routing scheme of two stage network (8 x 8)

Inlet	**Outlet**	**Routing**
M_0(000)	N_0(000)	M_0(000) → X_0→ A→A→X_0→N_0(000)
	N_1(001)	M_0(000) → X_0→ A→A→X_1→N_1(001)
	N_2(010)	M_0(000) → X_0→ B→B→X_2→N_2(010)
	N_3(011)	M_0(000) → X_0→ B→B→X_3→N_3(011)
	N_4(100)	M_0(000)→ X_0→ C→C→ X_4→N_4(100)
	N_5(101)	M_0(000)→ X_0→ C→C→ X_5→N_5(100)
	N_6(110)	M_0(000)→ X_0→ D→D→ X_6→N_6(110)
	N_7(111)	M_0(000) → X_0→ D→D→ X_7→N_7(111)

M_1(001)	N_0(000)	M_1(001) → X_1→ A→A→X_0→N_0(000)
	N_1(001)	M_1(001) → X_1→ A→A→X_1→N_1(001)
	N_2(010)	M_1(001) → X_1→ B→B→X_2→N_2(010)
	N_3(011)	M_1(001) → X_1→ B→B→X_3→N_3(011)
	N_4(100)	M_1(001) → X_1→ C→C→ X_4→N_4(100)
	N_5(101)	M_1(001) → X_1→ C→C→ X_5→N_5(100)
	N_6(110)	M_1(001) → X_1→ D→D→ X_6→N_6(110)
	N_7(111)	M_1(001) → X_1→ D→D→ X_7→N_7(111)
: :	: :	: :
M_7(111)	N_0(000)	M_7(111) → X_7→ M→M→X_0→N_0(000)
	N_1(001)	M_7(111) → X_7→ M→M→X_1→N_1(001)
	N_2(010)	M_7(111)→ X_7→ N → N →X_2→N_2(010)
	N_3(011)	M_7(111)→ X_7→ N→ N→ X_3→ N_3(011)
	N_4(100)	M_7(111) → X_7→ O→O→ X_4→N_4(100)
	N_5(101)	M_7(111) → X_7→ O→O→ X_5→ N_5(101)
	N_6(110)	M_7(111) → X_7→ P → P→ X_6→N_6(110)
	N_7(111)	$M_{7(}$111) → X_7→ P→ P→ X_7→ N_7(111)

Two stage network provides alternate paths for establishing the connection, in comparison with single stage network. The number of crosspoints in two stage network is S = Mb + Na = 8 × 4 + 8 × 4 = 64 and switching capacity = ab = 4 × 4 =16. This result shows that, the network can support 16 calls simultaneously. The two stage network may be of blocking nature, if $a \times b$ calls are in progress in full available condition and *(ab + 1)*th call arrives to the network, because calls are uniformly distributed with switching matrices. The solution of the problem or making the network with full availability is to use three or more stages.

3.3 THREE STAGE NETWORK

Three stage $N \times N$ network is considered and is shown in figure 3.4(a). Here N inlets are divided into a blocks having x inlets ($N = a \times x$) and N outlets are also divided into a blocks having x outlets ($N = a \times x$). The three stage network is realized in switching matrices of size $x \times b$ in first stage, $a \times a$ in second stage and $b \times x$ in third stage. Network has b alternative paths to reach any outlets of third stage. The total number of crosspoints can be calculated

$$\mathrm{S} = axb + ba^2 + bxa \qquad \text{Equ. 3.4)}$$

$$S = 2\ axb + ba^2 \qquad \text{Equ. (3.5)}$$

Putting the value of ($N = a \times x$) in the equation, the number of cross points are given as

$$S = b(\ 2N + a^2) \qquad \text{Equ. (3.6)}$$

The switching capacity of three stage network can be calculated, when the network is fully available.

$$Switching\ Capacity = aba = a^2b \qquad \text{Equ. (3.7)}$$

Three stage network (8 × 8) is shown in figure 3.4(b). The switching capacity of three stage network = $a^2b = 4^2 \times 4 = 64.$ It suggests that three stage (8 × 8) network is capable to support 64 calls simultaneously. The numbers of crosspoints for three stage (8 × 8) network are calculated using equation (3.5).The number of crosspoints $S = 2\ axb + ba^2 = 2 \times 4 \times 2 \times 4 + 4 \times 4^2 = 64 + 64 = 128$.

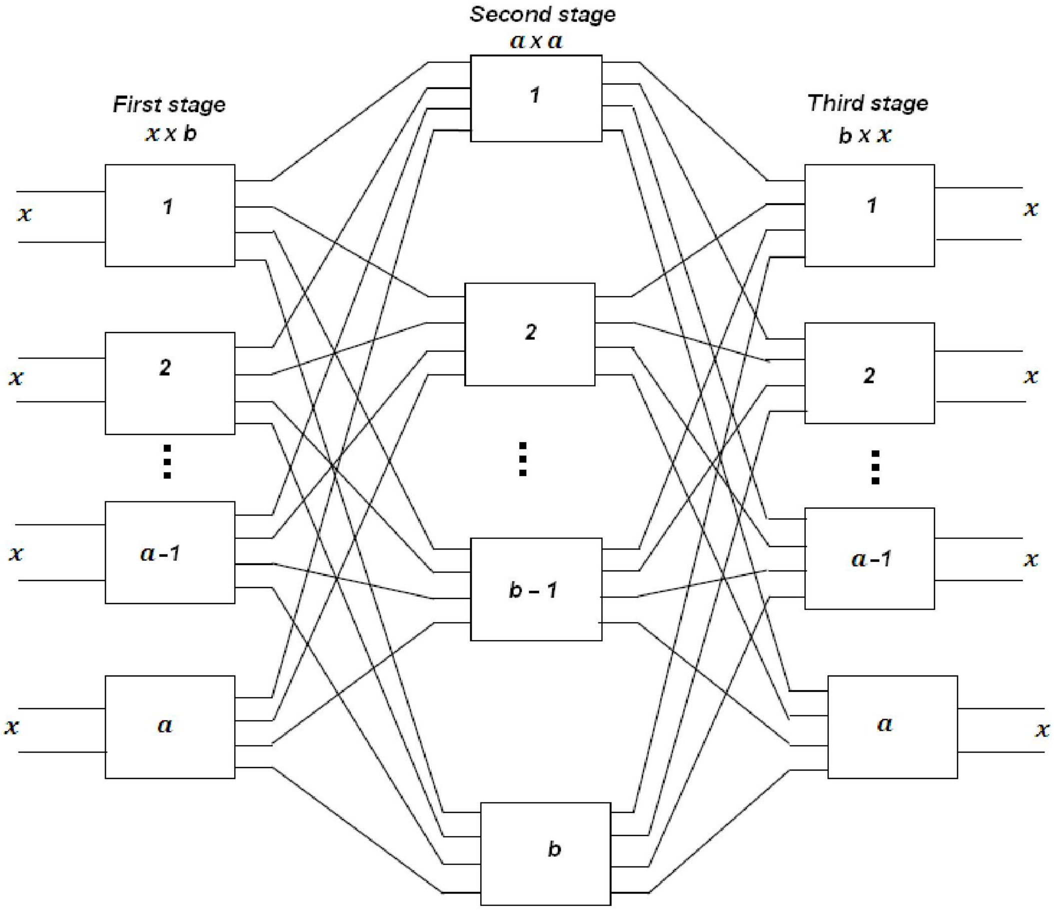


Fig 3.4(a) Three stage switching [11]

The routing scheme of the network can be understood using table 3.2. Let user *M_0(000)* want to communicate with *N_0(000),* there exists alternative paths to establish the connections. First path *$M_0(000) \rightarrow X_0 \rightarrow Y_0 \rightarrow P_0 \rightarrow T_0 \rightarrow Y_0 \rightarrow X_0 \rightarrow N_0(000)$* is through intermediate stage in block 1. Second path *$M_0(000) \rightarrow X_0 \rightarrow Y_1 \rightarrow Q_0 \rightarrow U_0 \rightarrow Y_1 \rightarrow X_0 \rightarrow$ \$N_0(000)$* is through intermediate stage in block 2. Third path *$M_0(000) \rightarrow X_0 \rightarrow Y_2 \rightarrow R_0 \rightarrow V_0 \rightarrow Y_2 \rightarrow X_0 \rightarrow$*

$N_0(000)$ is through intermediate stage in block 3 and fourth path is *$M_0(000) \rightarrow X_0 \rightarrow Y_3 \rightarrow S_0 \rightarrow W_0 \rightarrow Y_3 \rightarrow X_0 \rightarrow N_0(000)$* through intermediate stage in block 4. Similarly, all the subscribers are having four alternative paths. The advantage of three stage network is that, if any path in the network is busy, there are alternative paths for routing the calls.

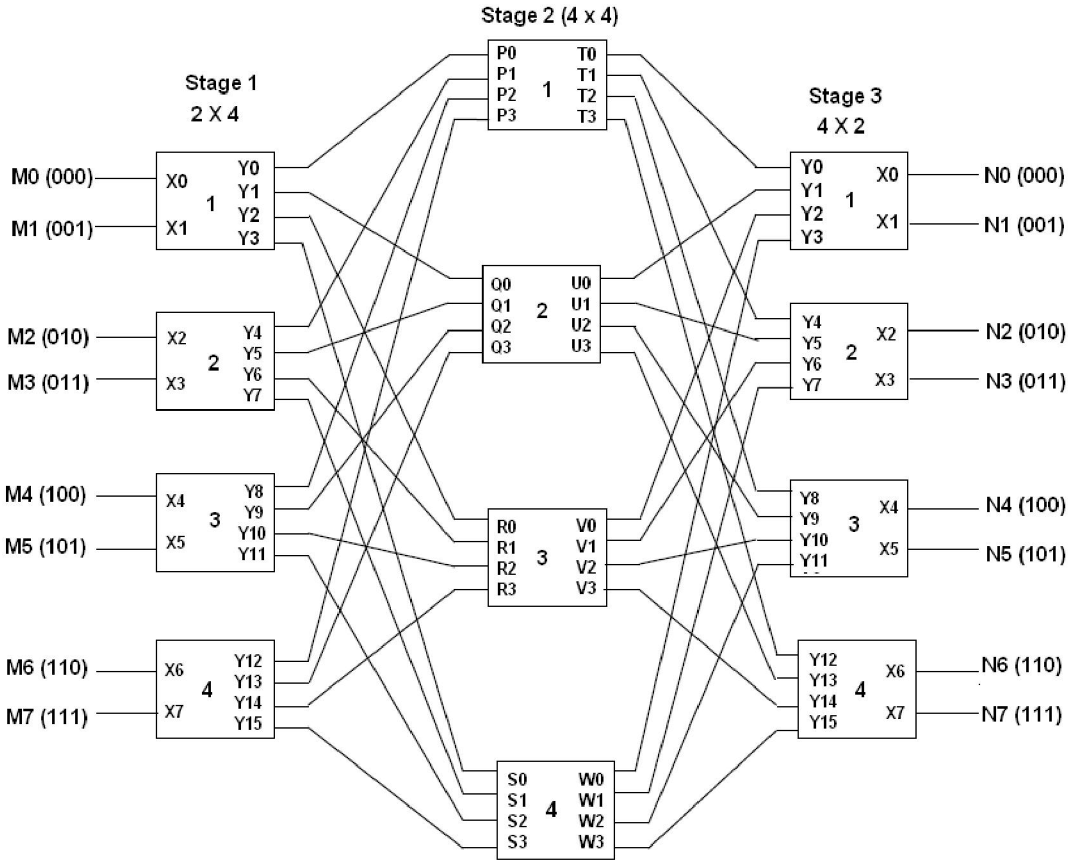


Fig. 3.4(b) Three stage switching network (8 x 8)

Table 3.2 Routing scheme of three stage network (8 x 8)

Inlet	Outlet	Routing
M_0 (000)	$N_0(000)$	$M_0(000) \rightarrow X_0 \rightarrow Y_0 \rightarrow P_0 \rightarrow T_0 \rightarrow Y_0 \rightarrow X_0 \rightarrow N_0(000)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_1 \rightarrow Q_0 \rightarrow U_0 \rightarrow Y_1 \rightarrow X_0 \rightarrow N_0(000)$

		$M_0(000) \rightarrow X_0 \rightarrow Y_2 \rightarrow R_0 \rightarrow V_0 \rightarrow Y_2 \rightarrow X_0 \rightarrow N_0(000)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_3 \rightarrow S_0 \rightarrow W_0 \rightarrow Y_3 \rightarrow X_0 \rightarrow N_0(000$
	$N_1(001)$	M_0 $(000) \rightarrow X_0 \rightarrow Y_0 \rightarrow P_0 \rightarrow T_0 \rightarrow Y_0 \rightarrow X_1 \rightarrow N_1(001)$
		M_0 $(000) \rightarrow X_0 \rightarrow Y_1 \rightarrow Q_0 \rightarrow U_0 \rightarrow Y_1 \rightarrow X_1 \rightarrow N_1(001)$
		M_0 $(000) \rightarrow X_0 \rightarrow Y_2 \rightarrow R_0 \rightarrow V_0 \rightarrow Y_2 \rightarrow X_1 \rightarrow N_1(001)$
		M_0 $(000) \rightarrow X_0 \rightarrow Y_3 \rightarrow S_0 \rightarrow W_0 \rightarrow Y_3 \rightarrow X_1 \rightarrow N_1(001)$
	$N_2(010)$	$M_0(000) \rightarrow X_0 \rightarrow Y_0 \rightarrow P_0 \rightarrow T_1 \rightarrow Y_4 \rightarrow X_2 \rightarrow N_2(010)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_1 \rightarrow Q_0 \rightarrow U_1 \rightarrow Y_5 \rightarrow X_2 \rightarrow N_2(010)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_2 \rightarrow R_0 \rightarrow V_1 \rightarrow Y_6 \rightarrow X_2 \rightarrow N_2(010)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_3 \rightarrow S_0 \rightarrow W_1 \rightarrow Y_7 \rightarrow X_2 \rightarrow N_2(010)$
	$N_3(011)$	$M_0(000) \rightarrow X_0 \rightarrow Y_0 \rightarrow P_0 \rightarrow T_1 \rightarrow Y_4 \rightarrow X_3 \rightarrow N_3(011)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_1 \rightarrow Q_0 \rightarrow U_1 \rightarrow Y_5 \rightarrow X_3 \rightarrow N_3(011)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_2 \rightarrow R_0 \rightarrow V_1 \rightarrow Y_6 \rightarrow X_3 \rightarrow N_3(011)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_3 \rightarrow S_0 \rightarrow W_1 \rightarrow Y_7 \rightarrow X_3 \rightarrow N_3(011)$
	$N_4(100)$	$M_0(000) \rightarrow X_0 \rightarrow Y_0 \rightarrow P_0 \rightarrow T_2 \rightarrow Y_8 \rightarrow X_4 \rightarrow N_4(100)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_1 \rightarrow Q_0 \rightarrow U_2 \rightarrow Y_9 \rightarrow X_4 \rightarrow N_4(100)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_2 \rightarrow R_0 \rightarrow V_2 \rightarrow Y_{10} \rightarrow X_4 \rightarrow N_4(100)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_3 \rightarrow S_0 \rightarrow W_2 \rightarrow Y_{11} \rightarrow X_4 \rightarrow N_4(100)$
	$N_5(101)$	$M_0(000) \rightarrow X_0 \rightarrow Y_0 \rightarrow P_0 \rightarrow T_2 \rightarrow Y_8 \rightarrow X_5 \rightarrow N_5(101)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_1 \rightarrow Q_0 \rightarrow U_2 \rightarrow Y_9 \rightarrow X_5 \rightarrow N_5(101)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_2 \rightarrow R_0 \rightarrow V_2 \rightarrow Y_{10} \rightarrow X_5 \rightarrow N_5(101)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_3 \rightarrow S_0 \rightarrow W_2 \rightarrow Y_{11} \rightarrow X_5 \rightarrow N_5(101)$

	$N_6(110)$	$M_0(000) \rightarrow X_0 \rightarrow Y_0 \rightarrow P_0 \rightarrow T_3 \rightarrow Y_{12} \rightarrow X_6 \rightarrow N_6(110)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_1 \rightarrow Q_0 \rightarrow U_3 \rightarrow Y_{13} \rightarrow X_6 \rightarrow N_6(110)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_2 \rightarrow R_0 \rightarrow V_3 \rightarrow Y_{14} \rightarrow X_6 \rightarrow N_6(110)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_3 \rightarrow S_0 \rightarrow W_3 \rightarrow Y_{15} \rightarrow X_6 \rightarrow N_6(110)$
	$N_7(111)$	$M_0(000) \rightarrow X_0 \rightarrow Y_0 \rightarrow P_0 \rightarrow T_3 \rightarrow Y_{12} \rightarrow X_7 \rightarrow N_7(111)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_1 \rightarrow Q_0 \rightarrow U_3 \rightarrow Y_{13} \rightarrow X_7 \rightarrow N_7(111)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_2 \rightarrow R_0 \rightarrow V_3 \rightarrow Y_{14} \rightarrow X_7 \rightarrow N_7(111)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_3 \rightarrow S_0 \rightarrow W_3 \rightarrow Y_{15} \rightarrow X_7 \rightarrow N_7(111)$
: :	: :	: :
$N_7(111)$	$N_7(111)$	$M_7(111) \rightarrow X_7 \rightarrow Y_{12} \rightarrow P_3 \rightarrow T_3 \rightarrow Y_{12} \rightarrow X_7 \rightarrow N_7(111)$
		$M_7(111) \rightarrow X_7 \rightarrow Y_{13} \rightarrow Q_3 \rightarrow U_3 \rightarrow Y_{13} \rightarrow X_7 \rightarrow N_7(111)$
		$M_7(111) \rightarrow X_7 \rightarrow Y_{14} \rightarrow R_3 \rightarrow V_3 \rightarrow Y_{14} \rightarrow X_7 \rightarrow N_7(111)$
		$M_7(111) \rightarrow X_7 \rightarrow Y_{15} \rightarrow S_3 \rightarrow W_3 \rightarrow Y_{15} \rightarrow X_7 \rightarrow N_7(111)$

3.4 FOUR STAGE NETWORK

Further extension of the stages results in better switching capacity as shown in figure 3.5(a).Considering a four stage $N \times N$ network, in which N inlets are divided into a blocks having x inlets ($N = a \times x$) and N outlets are divided into a blocks having x outlets ($N = a \times x$). The four stage network is realized with switching matrices of size $x \times b$ in first stage, $a \times b$ in second stage, $b \times a$ in third stage and $b \times x$ in fourth stage. The total number of crosspoints can be calculated

$$S = axb + ab^2 + ab^2 + bxa \qquad \text{Equ. (3.8)}$$

$$S = 2\ axb + 2ab^2 \qquad \text{Equ. (3.9)}$$

Putting the value of ($N = a \times x$) in the equation 3.9, the number of cross points are

$$S = 2(Nb + ab^2) \qquad \text{Equ. (3.10)}$$

The switching capacity of four stage network can be calculated, when the network is fully utilized.

$$Switching\ Capacity = abba = a^2b^2 \qquad \text{Equ. (3.11)}$$

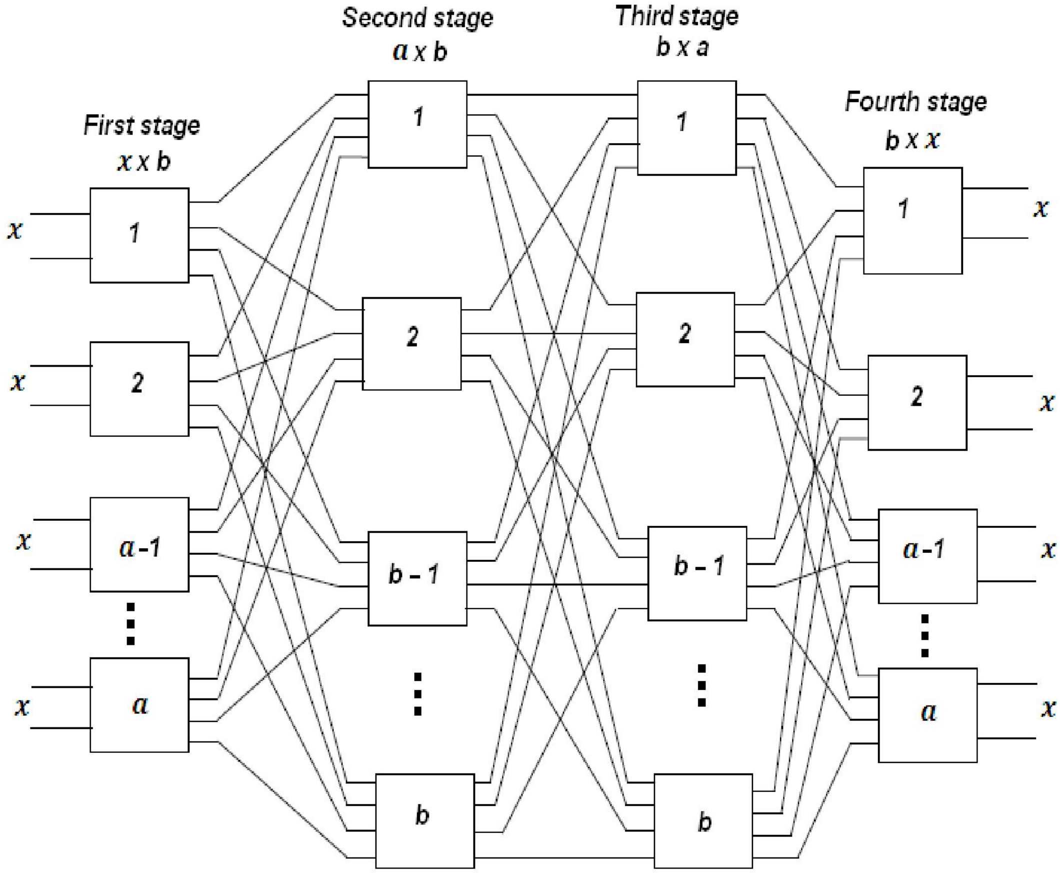


Fig 3.5(a) Four stage switching

In the 8 × 8 four stage networks shown in figure 5(b), $Switching\ Capacity = a^2b^2 = 4^2 \times 4^2 = 256.$ It means, the four stage (8 × 8) network is capable to support 256 calls simultaneously. The numbers of crosspoints for four stage (8 × 8) network are calculated using equation (3.9). The number of crosspoints $S = 2\ axb + 2ab^2 = 2 \times 4 \times 2 \times 4 + 2 \times 4 \times 4^2 = 64 + 128 = \ 192$. The routing scheme of the network can be understood using table 3.3. From the table it is clear that if any inlet wants to communicate to outlet, the call can be routed by 16 alternate paths. Therefore, the network cannot be blocked. The five stage switching provides more alternate paths in comparison to three and four stage networks.

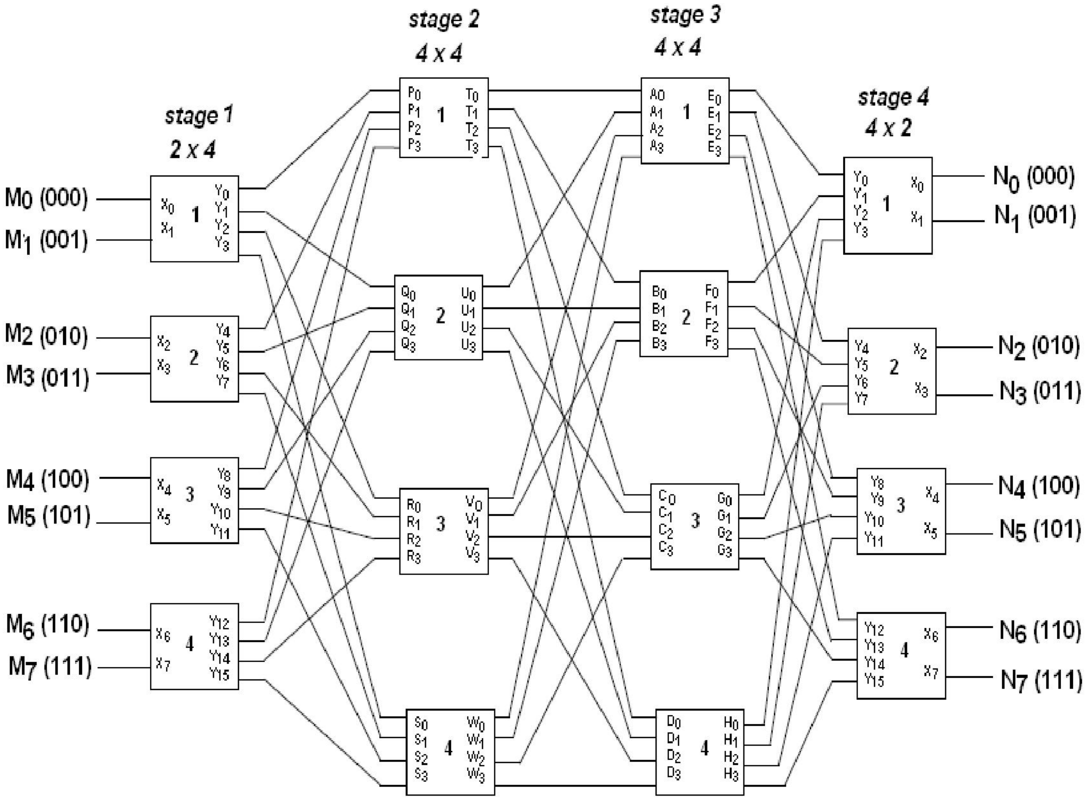


Fig 3.5(b) Four stage switching network (8 x 8)

Table 3.3 Routing scheme of four stage network (8 x 8)

Inlet	Outlet	Routing
$M_0(000)$	$N_0(000)$	$M_0(000) \rightarrow X_0 \rightarrow Y_0 \rightarrow P_0 \rightarrow T_0 \rightarrow A_0 \rightarrow E_0 \rightarrow Y_0 \rightarrow X_0 \rightarrow N_0(000)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_0 \rightarrow P_0 \rightarrow T_1 \rightarrow B_0 \rightarrow F_0 \rightarrow Y_1 \rightarrow X_0 \rightarrow N_0(000)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_0 \rightarrow P_0 \rightarrow T_2 \rightarrow C_0 \rightarrow G_0 \rightarrow Y_2 \rightarrow X_0 \rightarrow N_0(000)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_0 \rightarrow P_0 \rightarrow T_3 \rightarrow D_0 \rightarrow H_0 \rightarrow Y_3 \rightarrow X_0 \rightarrow N_0(000)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_1 \rightarrow Q_0 \rightarrow U_0 \rightarrow A_1 \rightarrow E_0 \rightarrow Y_0 \rightarrow X_0 \rightarrow N_0(000)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_1 \rightarrow Q_0 \rightarrow U_1 \rightarrow B_1 \rightarrow F_0 \rightarrow Y_1 \rightarrow X_0 \rightarrow N_0(000)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_1 \rightarrow Q_0 \rightarrow U_2 \rightarrow C_1 \rightarrow G_0 \rightarrow Y_2 \rightarrow X_0 \rightarrow N_0(000)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_1 \rightarrow Q_0 \rightarrow U_3 \rightarrow D_1 \rightarrow H_0 \rightarrow Y_3 \rightarrow X_0 \rightarrow N_0(000)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_2 \rightarrow R_0 \rightarrow V_0 \rightarrow A_2 \rightarrow E_0 \rightarrow Y_0 \rightarrow X_0 \rightarrow N_0(000)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_2 \rightarrow R_0 \rightarrow V_1 \rightarrow B_2 \rightarrow F_0 \rightarrow Y_1 \rightarrow X_0 \rightarrow N_0(000)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_2 \rightarrow R_0 \rightarrow V_2 \rightarrow C_2 \rightarrow G_0 \rightarrow Y_2 \rightarrow X_0 \rightarrow N_0(000)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_2 \rightarrow R_0 \rightarrow V_3 \rightarrow D_2 \rightarrow H_0 \rightarrow Y_3 \rightarrow X_0 \rightarrow N_0(000)$
		$M_{0(}000) \rightarrow X_0 \rightarrow Y_3 \rightarrow S_0 \rightarrow W_0 \rightarrow A_3 \rightarrow E_0 \rightarrow Y_0 \rightarrow X_0 \rightarrow N_0(000)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_3 \rightarrow S_0 \rightarrow W_1 \rightarrow B_3 \rightarrow F_0 \rightarrow Y_1 \rightarrow X_0 \rightarrow N_0(000)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_3 \rightarrow S_0 \rightarrow W_2 \rightarrow C_3 \rightarrow G_0 \rightarrow Y_2 \rightarrow X_0 \rightarrow N_0(000)$
		$M_0(000) \rightarrow X_0 \rightarrow Y_3 \rightarrow S_0 \rightarrow W_3 \rightarrow D_3 \rightarrow H_0 \rightarrow Y_3 \rightarrow X_0 \rightarrow N_0(000)$
: :	: :	: :
M_7 (111)	$M_7(111)$	$M_7(111) \rightarrow X_7 \rightarrow Y_{12} \rightarrow P_3 \rightarrow T_0 \rightarrow A_0 \rightarrow E_0 \rightarrow Y_0 \rightarrow X_0 \rightarrow N_0(000)$: $M_7(111) \rightarrow X_7 \rightarrow Y_{15} \rightarrow S_3 \rightarrow W_3 \rightarrow D_3 \rightarrow H_3 \rightarrow Y_{15} \rightarrow X_7 \rightarrow N_7(000)$

3.5 FIVE STAGE SWITCHING

To enhance the switching capacity further extension is done to five stage switching network. A five stage [91] 8 x 8 network is shown in the figure 3.6(a), where 8 inlets can communicate with 8 outlets in full duplex mode. Inlets are represented with their addresses as {M_0 (000), M_1 (001)}, M_2 (010), M_3 (011), M_4 (100), M_5 (101), M_6 (110), M_7 (111)} and outlets are represented with their addresses as {N_0 (000), N_1 (001)}, N_2 (010), N_3 (011), N_4 (100), N_5 (101), N_6 (110), N_7 (111)}. In five stage switching network all inlets and outlets are configured in different blocks. Each block is having two inlets/outlets.

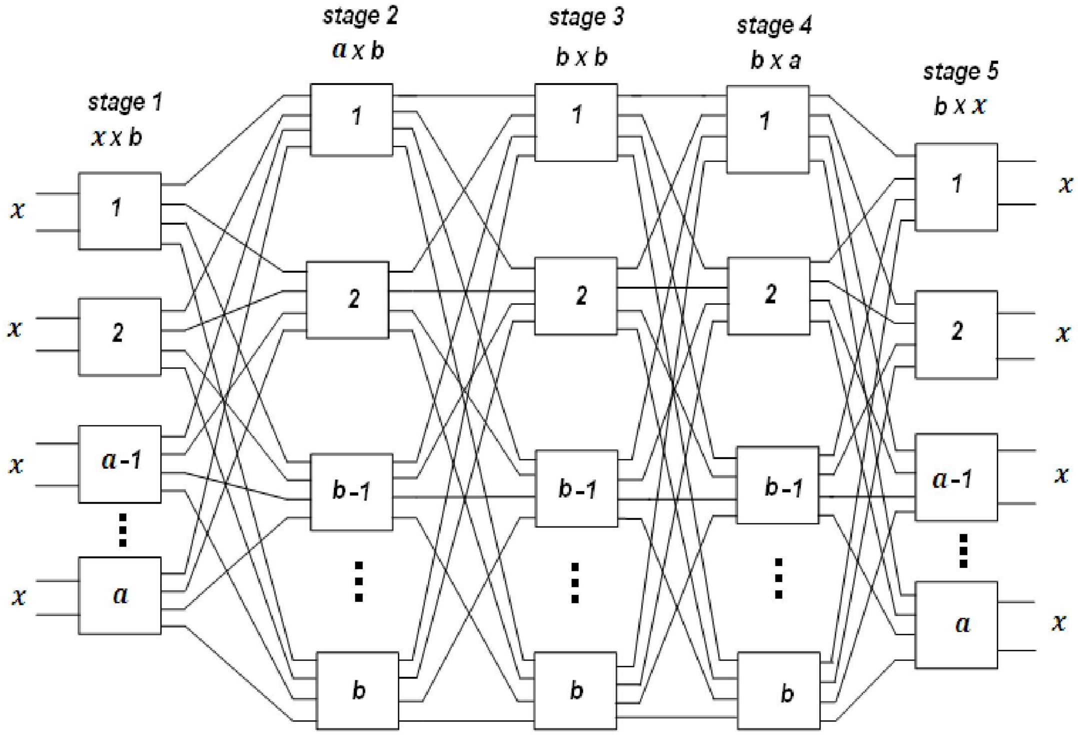


Fig 3.6(a) Five stage switching network (8 x 8)

` In 8 x 8 five stage structure inlets are outlets are configured in four blocks as inlet and outlet block matrix.

$$Inlet\ block\ matrix = \begin{bmatrix} \{\mathrm{M}_0(000), \mathrm{M}_1(001)\} \\ \{\mathrm{M}_2(010), \mathrm{M}_3(011)\} \\ \{\mathrm{M}_4(100), \mathrm{M}_5(101)\} \\ \{\mathrm{M}_6\ (110), \mathrm{M}_7\ (111)\} \end{bmatrix}$$

$$Outlet\ block\ matrix = \begin{bmatrix} \{\mathrm{N}_0(000), \mathrm{N}_1(001)\} \\ \{\mathrm{N}_2(010), \mathrm{N}_3(011)\} \\ \{\mathrm{N}_4(100), \mathrm{N}_5(101)\} \\ \{\mathrm{N}_6\ (110), \mathrm{N}_7\ (111)\} \end{bmatrix}$$

In inlet block matrix $\{\mathrm{M}_0(000), \mathrm{M}_1(001)\}$ represents first block,$\{\mathrm{M}_2(010), \mathrm{M}_3(011)\}$ second block, $\{\mathrm{M}_4(100), \mathrm{M}_5(101)\}$ third block and $\{\mathrm{M}_6(110), \mathrm{M}_7(111)\}$ fourth block. Similarly outlets are grouped in four blocks $\{\mathrm{N}_0(000), \mathrm{N}_1(001)\}$ first block, $\{\mathrm{N}_2(010), \mathrm{N}_3(011)\}$ second block, $\{\mathrm{N}_4(100), \mathrm{N}_5(101)\}$ third block and$\{\mathrm{N}_6(110), \mathrm{N}_7(111)\}$ fourth block.

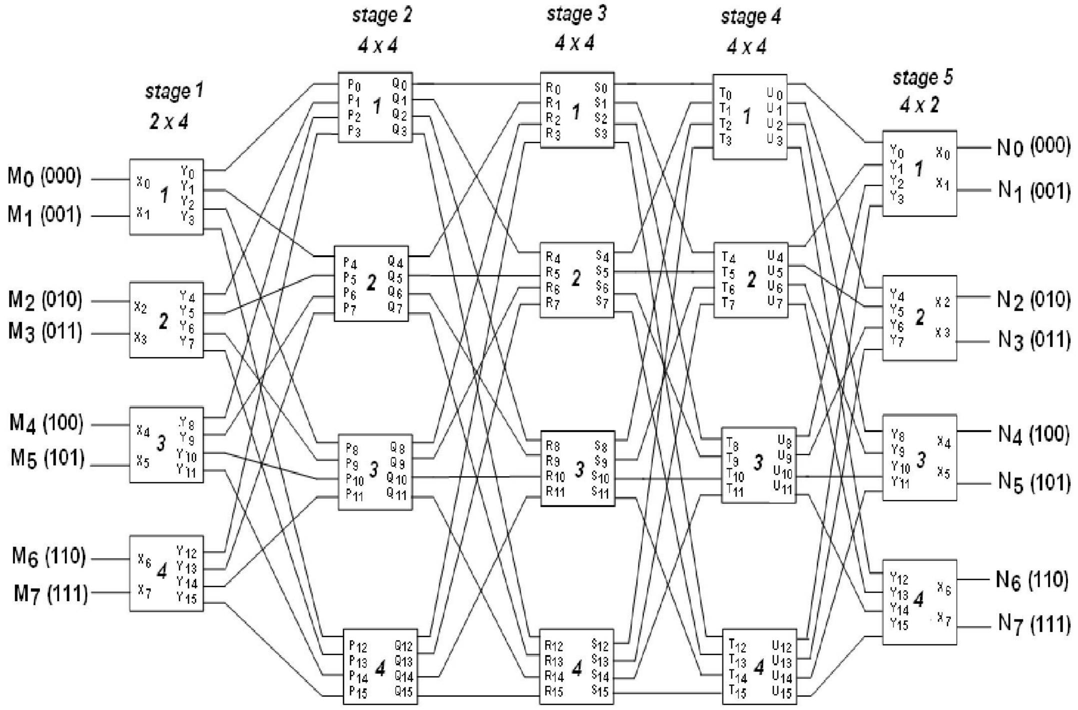


Fig 3.6(b) Routing structure of five stage switching network (8 × 8)

In the M × N structure shown in figure 3.6 (b), M inlets are divided into a blocks having x inlets such that (M = a × x) and N outlets are also divided into a blocks having x outlets such that (N = a × x). Hence inlets $M = a \times x = 4 \times 2 = 8$ outlets $N = a \times x = 4 \times 2 = 8$ .The value of a represents the number of blocks in first and fifth stage, x represents the value of inlets/outlets for each block of first and fifth stage. The intermediate stage, second third and fourth stages are having b number of blocks. The value of b is taken such that $a = b = 4$. The five stage network is realized using switching matrices of size $x \times b$ in first stage, $a \times b$ in second stage, $b \times b$ in third stage, $b \times a$, fourth stage and $b \times x$ in fifth stage under full availability of network. Network has a alternative paths to reach from first to second stage, b alternative paths to reach from second to third stage, b alternative paths to reach from third to fourth stage and a alternative paths to reach from fourth to fifth stage.

Switching Capacity [19] of the network is the parameter which defines the maximum number of call supported by the network, in case of fully available network [19] and numbers of crosspoints [18][29] are the total number of possible maximum paths under fully connected network. The total number of crosspoints can be calculated

$$S = axb + ab^2 + b^3 + ab^2 + bxa \qquad \text{Equ. (3.12)}$$

$$S = 2\,axb + 2ab^2 + b^3 \qquad \text{Equ. (3.13)}$$

Putting the value of (M = N = a × x) in the equation, the number of crosspoints are given as,

$S = 2\ (Nb + ab^2 + \ b^3) = 2\ (Mb + ab^2 + \ b^3)$ Equ. (3.14)

The switching capacity of five stage network can be calculated, when the network is fully utilized.

$Switching\ capacity = abbba = a^2b^3$ Equ. (3.15)

In the $8\ \times 8$ five stage network, the number of crosspoints are calculated as $S = 2\ axb + 2ab^2 + b^3\ = 2 \times 4 \times 2 \times 4 + 2 \times 4 \times 4^2{+}4^3 = 256$ and $Switching\ capacity = a^2b^3\ = 4^2 \times\ 4^3 = 1024.$ It means five stage network can support maximum 1024 calls simultaneously using 256 paths under the condition of full available network. The routing architecture of $8\ \times 8$ five stage network is shown in the figure 3.6(b) and possible routing scheme is listed in the table 3.4. Considering that M_0 inlet wants to communicate with N_0 outlet, there are 64 alternative paths to establish connections. Similarly it can communicate with any outlet (N_1 …N_7) associated with 64 connections corresponding to each outlet. All inlets (M_0 … M_7) can communicate with all outlets (N_0…N_7) in similar way.

Table 3.4 Routing scheme of five stage switching network (8 × 8)

Inlet	Outlet	Routing
000	000	M_0(000) – X_0 – Y_0– P_0 – Q_0 – R_0 – S_0 – T_0 –U_0 –Y_0– X_0 – N_0(000) M_0(000) – X_0 – Y_0– P_0 – Q_0 – R_0 – S_1 – T_4 –U_4 –Y_1– X_0 – N_0(000) M_0(000) – X_0 – Y_0– P_0 – Q_0 – R_0 – S_2 – T_8 –U_8 –Y_2– X_0 – N_0(000) M_0(000) – X_0 – Y_0– P_0 – Q_0 – R_0 – S_3 – T_{12} –U_{12} –Y_3– X_0 – N_0(000)

		$M_0(000) - X_0 - Y_0 - P_0 - Q_1 - R_4 - S_4 - T_1 - U_0 - Y_0 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_0 - P_0 - Q_1 - R_4 - S_5 - T_5 - U_4 - Y_1 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_0 - P_0 - Q_1 - R_4 - S_6 - T_9 - U_8 - Y_2 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_0 - P_0 - Q_1 - R_4 - S_7 - T_{13} - U_{12} - Y_3 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_0 - P_0 - Q_2 - R_8 - S_8 - T_2 - U_0 - Y_0 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_0 - P_0 - Q_2 - R_8 - S_9 - T_6 - U_4 - Y_1 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_0 - P_0 - Q_2 - R_8 - S_{10} - T_{10} - U_8 - Y_2 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_0 - P_0 - Q_2 - R_8 - S_{11} - T_{14} - U_{12} - Y_3 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_0 - P_0 - Q_3 - R_{12} - S_{12} - T_3 - U_0 - Y_0 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_0 - P_0 - Q_3 - R_{12} - S_{13} - T_7 - U_4 - Y_1 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_0 - P_0 - Q_3 - R_{12} - S_{14} - T_{11} - U_8 - Y_2 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_0 - P_0 - Q_3 - R_{12} - S_{15} - T_{15} - U_{12} - Y_3 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_1 - P_4 - Q_4 - R_1 - S_0 - T_0 - U_0 - Y_0 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_1 - P_4 - Q_4 - R_1 - S_1 - T_4 - U_4 - Y_1 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_1 - P_4 - Q_4 - R_1 - S_2 - T_8 - U_8 - Y_2 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_1 - P_4 - Q_4 - R_1 - S_3 - T_{12} - U_{12} - Y_3 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_1 - P_4 - Q_5 - R_5 - S_4 - T_1 - U_0 - Y_0 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_1 - P_4 - Q_5 - R_5 - S_5 - T_5 - U_4 - Y_1 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_1 - P_4 - Q_5 - R_5 - S_6 - T_9 - U_8 - Y_2 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_1 - P_4 - Q_5 - R_5 - S_7 - T_{13} - U_{12} - Y_3 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_1 - P_4 - Q_6 - R_9 - S_8 - T_2 - U_0 - Y_0 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_1 - P_4 - Q_6 - R_9 - S_9 - T_6 - U_4 - Y_1 - X_0 - N_0(000)$

		$M_0(000) - X_0 - Y_1 - P_4 - Q_6 - R_9 - S_{10} - T_{10} - U_8 - Y_2 - X_0 - N_0(000)$
		$M_0(000) - X_0 - Y_1 - P_4 - Q_6 - R_9 - S_{11} - T_{14} - U_{12} - Y_3 - X_0 - N_0(000)$
		$M_0(000) - X_0 - Y_1 - P_4 - Q_7 - R_{13} - S_{12} - T_3 - U_0 - Y_0 - X_0 - N_0(000)$
		$M_0(000) - X_0 - Y_1 - P_4 - Q_7 - R_{13} - S_{13} - T_7 - U_4 - Y_1 - X_0 - N_0(000)$
		$M_0(000) - X_0 - Y_1 - P_4 - Q_7 - R_{13} - S_{14} - T_{11} - U_8 - Y_2 - X_0 - N_0(000)$
		$M_0(000) - X_0 - Y_1 - P_4 - Q_7 - R_{13} - S_{15} - T_{15} - U_{12} - Y_3 - X_0 - N_0(000)$
		$M_0(000) - X_0 - Y_2 - P_8 - Q_8 - R_2 - S_0 - T_0 - U_0 - Y_0 - X_0 - N_0(000)$
		$M_0(000) - X_0 - Y_2 - P_8 - Q_8 - R_2 - S_1 - T_4 - U_4 - Y_1 - X_0 - N_0(000)$
		$M_0(000) - X_0 - Y_2 - P_8 - Q_8 - R_2 - S_2 - T_8 - U_8 - Y_2 - X_0 - N_0(000)$
		$M_0(000) - X_0 - Y_2 - P_8 - Q_8 - R_2 - S_3 - T_{12} - U_{12} - Y_3 - X_0 - N_0(000)$
		$M_0(000) - X_0 - Y_2 - P_8 - Q_9 - R_6 - S_4 - T_1 - U_0 - Y_0 - X_0 - N_0(000)$
		$M_0(000) - X_0 - Y_2 - P_8 - Q_9 - R_6 - S_5 - T_5 - U_4 - Y_1 - X_0 - N_0(000)$
		$M_0(000) - X_0 - Y_2 - P_8 - Q_9 - R_6 - S_6 - T_9 - U_8 - Y_2 - X_0 - N_0(000)$
		$M_0(000) - X_0 - Y_2 - P_8 - Q_9 - R_6 - S_7 - T_{13} - U_{12} - Y_3 - X_0 - N_0(000)$
		$M_0(000) - X_0 - Y_2 - P_8 - Q_{10} - R_{10} - S_8 - T_2 - U_0 - Y_0 - X_0 - N_0(000)$
		$M_0(000) - X_0 - Y_2 - P_8 - Q_{10} - R_{10} - S_9 - T_6 - U_4 - Y_1 - X_0 - N_0(000)$
		$M_0(000) - X_0 - Y_2 - P_8 - Q_{10} - R_{10} - S_{10} - T_{10} - U_8 - Y_2 - X_0 - N_0(000)$
		$M_0(000) - X_0 - Y_2 - P_8 - Q_{10} - R_{10} - S_{11} - T_{14} - U_{12} - Y_3 - X_0 - N_0(000)$
		$M_0(000) - X_0 - Y_2 - P_8 - Q_{11} - R_{14} - S_{12} - T_3 - U_0 - Y_0 - X_0 - N_0(000)$
		$M_0(000) - X_0 - Y_2 - P_8 - Q_{11} - R_{14} - S_{13} - T_7 - U_4 - Y_1 - X_0 - N_0(000)$
		$M_0(000) - X_0 - Y_2 - P_8 - Q_{11} - R_{14} - S_{14} - T_{11} - U_8 - Y_2 - X_0 - N_0(000)$
		$M_0(000) - X_0 - Y_2 - P_8 - Q_{11} - R_{14} - S_{15} - T_{15} - U_{12} - Y_3 - X_0 - N_0(000)$

		$M_0(000) - X_0 - Y_3 - P_{12} - Q_{12} - R_3 - S_0 - T_0 - U_0 - Y_0 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_3 - P_{12} - Q_{12} - R_3 - S_1 - T_4 - U_4 - Y_1 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_3 - P_{12} - Q_{12} - R_3 - S_2 - T_8 - U_8 - Y_2 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_3 - P_{12} - Q_{12} - R_3 - S_3 - T_{12} - U_{12} - Y_3 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_3 - P_{12} - Q_{13} - R_7 - S_4 - T_1 - U_0 - Y_0 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_3 - P_{12} - Q_{13} - R_7 - S_5 - T_5 - U_4 - Y_1 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_3 - P_{12} - Q_{13} - R_7 - S_6 - T_9 - U_8 - Y_2 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_3 - P_{12} - Q_{13} - R_7 - S_7 - T_{13} - U_{12} - Y_3 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_3 - P_{12} - Q_{14} - R_{11} - S_8 - T_2 - U_0 - Y_0 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_3 - P_{12} - Q_{14} - R_{11} - S_9 - T_6 - U_4 - Y_1 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_3 - P_{12} - Q_{14} - R_{11} - S_{10} - T_{10} - U_8 - Y_2 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_3 - P_{12} - Q_{14} - R_{11} - S_{11} - T_{14} - U_{12} - Y_3 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_3 - P_{12} - Q_{15} - R_{15} - S_{12} - T_3 - U_0 - Y_0 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_3 - P_{12} - Q_{15} - R_{15} - S_{13} - T_7 - U_4 - Y_1 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_3 - P_{12} - Q_{15} - R_{15} - S_{14} - T_{11} - U_8 - Y_2 - X_0 - N_0(000)$ $M_0(000) - X_0 - Y_3 - P_{12} - Q_{15} - R_{15} - S_{15} - T_{15} - U_{12} - Y_3 - X_0 - N_0(000)$
	001	$M_0(000) - X_0 - Y_0 - P_0 - Q_0 - R_0 - S_0 - T_0 - U_0 - Y_0 - X_1 - N_1(001)$: $M_0(000) - X_0 - Y_3 - P_{12} - Q_{15} - R_{15} - S_{15} - T_{15} - U_{12} - Y_3 - X_1 - N_1(001)$
	010	$M_0(000) - X_0 - Y_0 - P_0 - Q_0 - R_0 - S_0 - T_0 - U_1 - Y_4 - X_2 - N_2(010)$: $M_0(000) - X_0 - Y_3 - P_{12} - Q_{15} - R_{15} - S_{15} - T_{15} - U_{13} - Y_7 - X_2 - N_2(010)$

	011	$M_0(000) - X_0 - Y_0 - P_0 - Q_0 - R_0 - S_0 - T_0 - U_1 - Y_4 - X_3 - N_3(011)$: $M_0(000) - X_0 - Y_3 - P_{12} - Q_{15} - R_{15} - S_{15} - T_{15} - U_{13} - Y_7 - X_3 - N_3(011)$
	100	$M_0(000) - X_0 - Y_0 - P_0 - Q_0 - R_0 - S_0 - T_0 - U_2 - Y_8 - X_4 - N_4(100)$: $M_0(000) - X_0 - Y_3 - P_{12} - Q_{15} - R_{15} - S_{15} - T_{15} - U_{14} - Y_{11} - X_4 - N_4(100)$
	101	$M_0(000) - X_0 - Y_0 - P_0 - Q_0 - R_0 - S_0 - T_0 - U_2 - Y_8 - X_5 - N_5(101)$: $M_0(000) - X_0 - Y_3 - P_{12} - Q_{15} - R_{15} - S_{15} - T_{15} - U_{14} - Y_{11} - X_5 - N_5(101)$
	110	$M_0(000) - X_0 - Y_0 - P_0 - Q_0 - R_0 - S_0 - T_0 - U_3 - Y_{12} - X_6 - N_6(110)$: $M_0(000) - X_0 - Y_3 - P_{12} - Q_{15} - R_{15} - S_{15} - T_{15} - U_{15} - Y_{15} - X_6 - N_6(110)$
	111	$M_0(000) - X_0 - Y_0 - P_0 - Q_0 - R_0 - S_0 - T_0 - U_3 - Y_{12} - X_7 - N_7(111)$: $M_0(000) - X_0 - Y_3 - P_{12} - Q_{15} - R_{15} - S_{15} - T_{15} - U_{15} - Y_{15} - X_7 - N_7(111)$
001	000 : 111	$M_1(001) - X_1 - Y_0 - P_0 - Q_0 - R_0 - S_0 - T_0 - U_0 - Y_0 - X_0 - N_0(000)$: $M_1(001) - X_1 - Y_3 - P_{12} - Q_{15} - R_{15} - S_{15} - T_{15} - U_{15} - Y_{15} - X_7 - N_7(111)$
010	000 : 111	$M_2(010) - X_2 - Y_4 - P_1 - Q_0 - R_0 - S_0 - T_0 - U_0 - Y_0 - X_0 - N_0(000)$: $M_2(010) - X_2 - Y_7 - P_{13} - Q_{15} - R_{15} - S_{15} - T_{15} - U_{15} - Y_{15} - X_7 - N_7(111)$
011	000	$M_3(011) - X_3 - Y_4 - P_1 - Q_0 - R_0 - S_0 - T_0 - U_0 - Y_0 - X_0 - N_0(000)$

	: 111	: $M_3(011) - X_3 - Y_7 - P_{13} - Q_{15} - R_{15} - S_{15} - T_{15} - U_{15} - Y_{15} - X_7 - N_7(111)$
100	000 : 111	$M_4(100) - X_4 - Y_8 - P_2 - Q_0 - R_0 - S_0 - T_0 - U_0 - Y_0 - X_0 - N_0(000)$: $M_4(100) - X_4 - Y_{11} - P_{14} - Q_{15} - R_{15} - S_{15} - T_{15} - U_{15} - Y_{15} - X_7 - N_7(111)$
101	000 : 111	$M_5(101) - X_5 - Y_8 - P_2 - Q_0 - R_0 - S_0 - T_0 - U_0 - Y_0 - X_0 - N_0(000)$: $M_5(101) - X_5 - Y_{11} - P_{14} - Q_{15} - R_{15} - S_{15} - T_{15} - U_{15} - Y_{15} - X_7 - N_7(111)$
110	000 : 111	$M_6(110) - X_6 - Y_{12} - P_3 - Q_0 - R_0 - S_0 - T_0 - U_0 - Y_0 - X_0 - N_0(000)$: $M_6(110) - X_6 - Y_{15} - P_{15} - Q_{15} - R_{15} - S_{15} - T_{15} - U_{15} - Y_{15} - X_7 - N_7(111)$
111	000 : 111	$M_7(111) - X_7 - Y_{12} - P_3 - Q_0 - R_0 - S_0 - T_0 - U_0 - Y_0 - X_0 - N_0(000)$: $M_7(111) - X_7 - Y_{15} - P_{15} - Q_{15} - R_{15} - S_{15} - T_{15} - U_{15} - Y_{15} - X_7 - N_7(111)$

CHAPTER SUMMARY

A single stage network has limited number of crosspoints and switching capacity. The major drawback of single stage switching is that, if any link fails, there is not any alternate path to establish the call between inlet and outlet. Single stage network can be configured using equivalent multistage network and the switching capacity of the exchange is increased. In this chapter, two stage, three stage, four stage and five stage switching networks are discussed. Their routing scheme, calculations relating to switching capacity, no of crosspoints, analyzed. It is found that, for an 8 x 8 network the switching capacity is increased with increment in number of stages are analyzed, when the network is in full utilization state. It is found that, two stage network supports 16 calls, three stage 64 calls, four stage 256 calls and five stage 1024 calls. Therefore, a multistage network is called full connected and full available network and can be configured in small clusters ICs based on network support structures.

CHAPTER-4

MULTISTAGE 3D NETWORK SECURITY

Network on chip (NoC) uses 2D and 3D structures, and reconfigurable architectures with topological structures. NoC can have bus, ring, star, tree, mesh, torus, hierarchical, omega topological structures. Similarly, multistage network follow the concept of 2D and 3D multidimensional structures. The chapter includes the details of 2D and 3D NoC, with mesh topological switching, in which source node has the maximum routes to connect the destination node, which are N(N-1)/2. This chapter also explains the need of multistage network security, existing security algorithms and new security algorithm named TACIT encryption and decryption algorithm, supporting to multistage multidimensional NoC. The encoding and decoding of the tones of source and destination subscribers in multistage NoC, using Dual Tone Multi frequency (DTMF) technique is also discussed. To control the network congestion, NoC integration is included with time division multiplexed switch to enable the inlets and outlets in duplex mode, in inter and intra exchange telecommunication environment.

4.1 3D MESH NETWORK

Network on Chip (NoC) is the latest approach to overcome the limitation of bus based communication network [1-3]. NoC is a set of routers employed in a network, in which different nodes are interconnected with their cores and can communicate with each others. In a network data comes in packets and sent to the destination with IP via routers and links [24]. When a packet reaches to its destination address, it means it is switched [25] to the IP attached to the router. On chip communications among different networks is possible using interconnection network topology [35] [36], switching, routing, queuing, flow control [71] and scheduling. The research is going on three dimensional topological structures, network on chip design. The idea of NoC is derived from distributed computing and large scale computer networks. There are different routing techniques used in NoC design, and considerations to meet high throughput. Due to constraints on hardware and memory resources utilization, the routing methods for NoC should be very simple, and can have 2D and 3D network configurations.

To understand the behavior of 3D NoC structure, first it is necessary to understand the behavior of 2D NoC structure. 2D NoC follows the crosspoint technology which allows addressing any node at any time [51]. A crosspoint switch is a switch connecting multiple inputs to multiple outputs in a matrix form. The 2D NoC architecture is an m × n mesh of switches [70] and resources are placed on the slots formed by the switches. For an m x n architecture there are m nodes on X axis and n nodes on Y axis respectively. For the design and implementation of switching network, 8 × 8 architecture is considered in the work. With 8 × 8 switching mesh network 64 nodes can be addressed at one time. To address the 64 nodes, 3 bits were required individually for both the axes

(2^n = 8). Three bits row address is assigned to nodes on X axis. Similarly 3 bit address is assigned on the nodes on Y axis. Addressing and node selection scheme is described in the functional table 4.1. It is evident from table 4.1 that if the row address is 000 and column address is 110, node N_6 is selected. Similarly any node could be selected based on node address table having row address and column address as shown in figure 4.1.

Table 4.1 Node address generation scheme in 2D structure

Row Address	Column Address	Destination Node
000	000 : 111	Node 0 : Node 7
001	000 : 111	Node 8 : Node 15
010	000 : 111	Node 16 : Node 23
011	000 : 111	Node 24 : Node 31
100	000 : 111	Node 32 : Node 39
101	000 : 111	Node 40 : Node 47
110	000 : 111	Node 48 : Node 55
111	000 : 111	Node 56 : Node 63

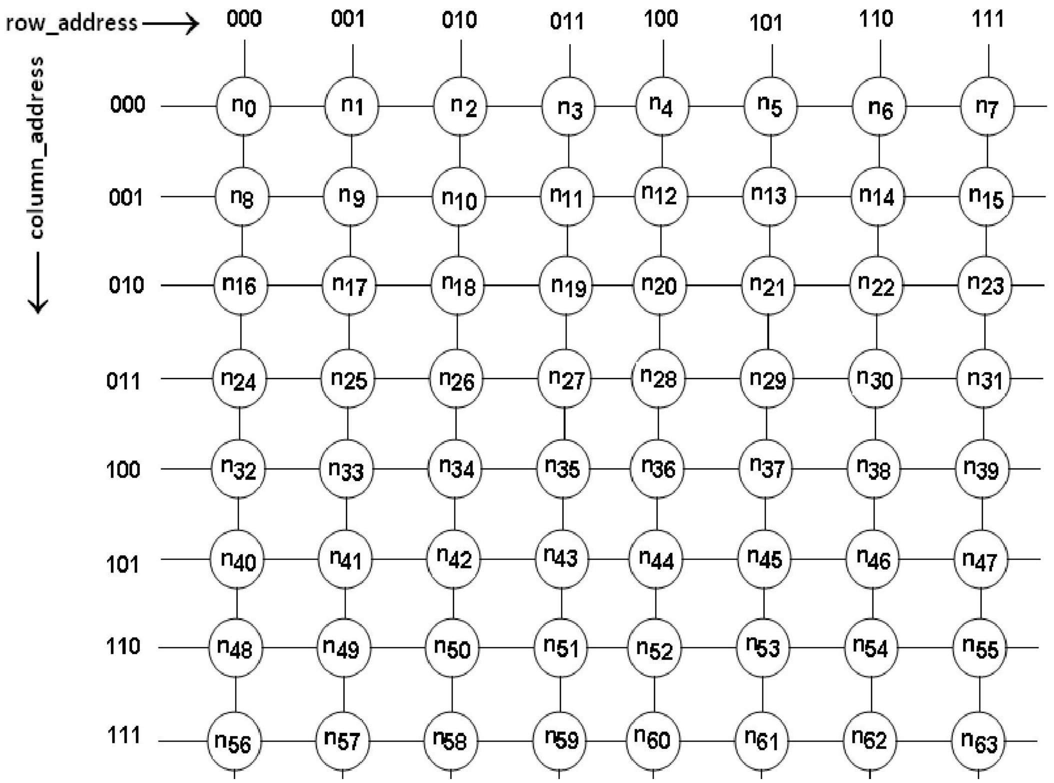


Fig. 4.1 2D crosspoint topological (8 × 8) structure

In 3D NoC architecture nodes are configured in X, Y and Z directions. In 2D NoC, sometimes there are the chances of not receiving the exact data sent by the sending node [42]. It results into an erroneous transmission. 3D NoC stacking of silicon layers has emerged as a promising direction for scaling [52]. In 3D stacking, a design is spited into multiple silicon layers, which are stacked on the top of each other. The 3D stacking technology has several major advantages including smaller footprint on each layer, shorter global wires, and ease of integration of diverse technologies, as each could be designed as a separate layer. Another advantage of 3D structure is that signal recovery is fast in comparison to 2D because the signal is recovered in one extra dimension. The 3D NoC architecture assures the true data transmission. Figure 4.2 shows the $8 \times 8 \times 8$

switching mesh network. The functionality of the 3D NoC architecture is described in table 4.2. Since it was not possible to identify the nodes in one direction, an alternative approach was adopted to identify the nodes in 3D NoC architecture. 3D topological structure was broken into parallel 2D structures like XY axis, YZ axis, ZX axis. In the structure, row address, column address and third address represent the addresses of the nodes in X, Y and Z axes respectively.

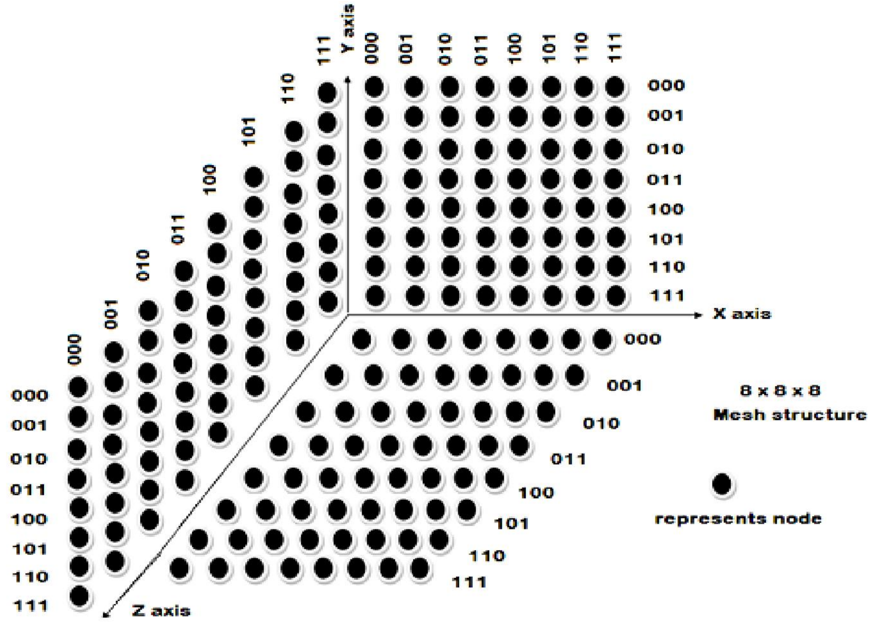


Fig. 4.2 3D network structure for (8 x 8 x 8) switching structure

For the chip implementation, 8 × 8 × 8 network is divided into three 8 × 8 2D NoCs, one in XY direction, second in YZ direction, and third in ZX direction. The 8 × 8 2D NoC configuration in XY direction has been assigned 3 bits for row addresses on X axis and 3 bits for column address on Y axis. The 8 × 8 2D NoC configuration in YZ direction has assigned 3 bits for column addresses on Y axis and 3 bits for third address on Z axis Similarly, The 2D NoC configuration 8 × 8 in XZ direction have assigned 3 bits for row addresses on X axis and 3 bits for third address on Z axis. The address generation scheme of 3D NoC is shown in the table 4.2. As a specific example, let the node no. 62 needs to be identified. The node detection will be realized by the address in XY direction (X = Row address (110), Y = Column address (010)), YZ direction (Y = Column address (110), Z = Third address (010)) and ZX direction (Z = Row address (110), X = Third address (010)) will be transmitted.

Table 4.2 Node address generation scheme in 3D network structures

XY Dimensional		**YZ Dimensional**		**ZX Dimensional**		**Destination node selection**
Row address	**Column address**	**Column Address**	**Third address**	**Row address**	**Third address**	
000 : 000	000 : 111	000 : 000	000 : 111	000 : 000	000 : 111	Node 0 : Node 7
001 : 001	000 : 111	001 : 001	000 : 111	001 : 001	000 : 111	Node 8 : Node 15
010 : 010	000 : 111	010 : 010	000 : 111	010 : 010	000 : 111	Node16 : Node 23
011 : 011	000 : 111	011 : 011	000 : 111	011 : 011	000 : 111	Node 24 : Node 31
100 : 100	000 : 111	100 : 100	000 : 111	100 : 100	000 : 111	Node 32 : Node 39
101 : 101	000 : 111	101 : 101	000 : 111	101 : 101	000 : 111	Node 40 : Node 47
110 : 110	000 : 111	110 : 110	000 : 111	110 : 110	000 : 111	Node 48 : Node 55
111 : 111	000 : 111	111 : 111	000 : 111	111 : 111	000 : 111	Node 56 : Node 63

4.2 NETWORK SECURITY

Multistage networks route data in packet form via different paths. A high performance multistage network is needed to implement packet forwarding and routing to own defined path. When the network provides the internet services to the different networks, then security of the network is utmost, because the network can communicate over any untrusted network [47]. In this scenario, network security is a major concern in different services such as data storage, secure data distribution and internet services [61]. Cryptographic [84] mechanism forms a foundation on network security [57], which help in implementation of security system based networks. There are encryption and decryption cryptographic algorithms. These algorithms suggest the ways by which it is possible to transfer secured data over multistage networks [91].Encryption is the process of converting plain text or unhidden text to a cipher text or hidden text, to secure against thieves under key management policy [84]. In encryption, [30] the data is locked at one end by the sender with the help of key and routed over network. Decryption is the process to retrieve the same text from the cipher text at another end. In decryption, same data is received, when the receiver is breaks the encrypted data with the help of key. The encryption and decryption process is shown in figure 4.3.

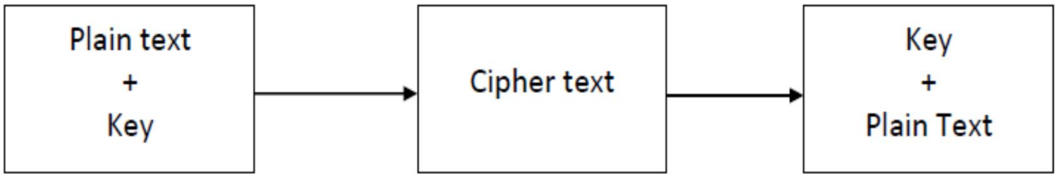


Fig. 4.3 Encryption-decryption process

The key size is a very important aspect to secure the data, long key size means the data is more secured. Encryption algorithms are very important for cryptography with an approach of key management because there are different algorithms that offer different degree of security based on key size. There are couples of encryption and decryption algorithms which are already proposed. A comparison of these techniques is shown in table 4.3. Table shows the various cryptographic techniques and their features on the basis of type, key size, and block size. It can be seen that from this comparison table that TACIT encryption technique has a unique independent approach by having a new key distribution system along with mathematical foundation [8]. The main advantage of TACIT logic is that, it can processes 'N' bits blocks and 'N' bits key size. This approach may be good if the block size is less than the key size [71]. The algorithm may be implemented in any languages, which support unicode system facility like VHDL, Verilog HDL, Java, C#, System C, .Net, etc.

Table 4.3 Comparison of various encryption algorithms on the basis of key size and block size [84].

Algorithm	**Key size (bits)**	**Block size (bits)**	**Features**
DES	64	64	Most common, Not strong enough
Triple DES	192	64	Modification of DES, Adequate security
AES	Variable	128	Replacement of DES, Excellent

	(192 /256)		security, limited key size
Kasumi Encryption core	128	64	Designed for Third Generation Partnership Project(3GPP), used in Universal Mobile Telecommunication System UMTS, limited to 64-bits word
Blowfish	448	64	Excellent security, No. of bits are variable ranging from 16-448 bits.
RSA	1024	128	Asymmetric algorithm, speed is low
RC4	Variable (40 or 128)	Variable (32,64,128)	Fast stream chipper in Secured Socket Layer (SSL), more memory is required since they work on large chunk of data (stream) instead of block cipher.
X-MODES	32	32	Enhanced security level & faster.
TACIT Encryption	n-bits	n-bits	Good for small size of packets

4.2.1 DATA ENCYPTION LOGIC FOR TACIT ALGORITHM

The TACIT encryption logic [21] for data communication between two nodes of NoC is presented with the help of following algorithm. The corresponding flowchart of the algorithm is shown in figure 4.4.

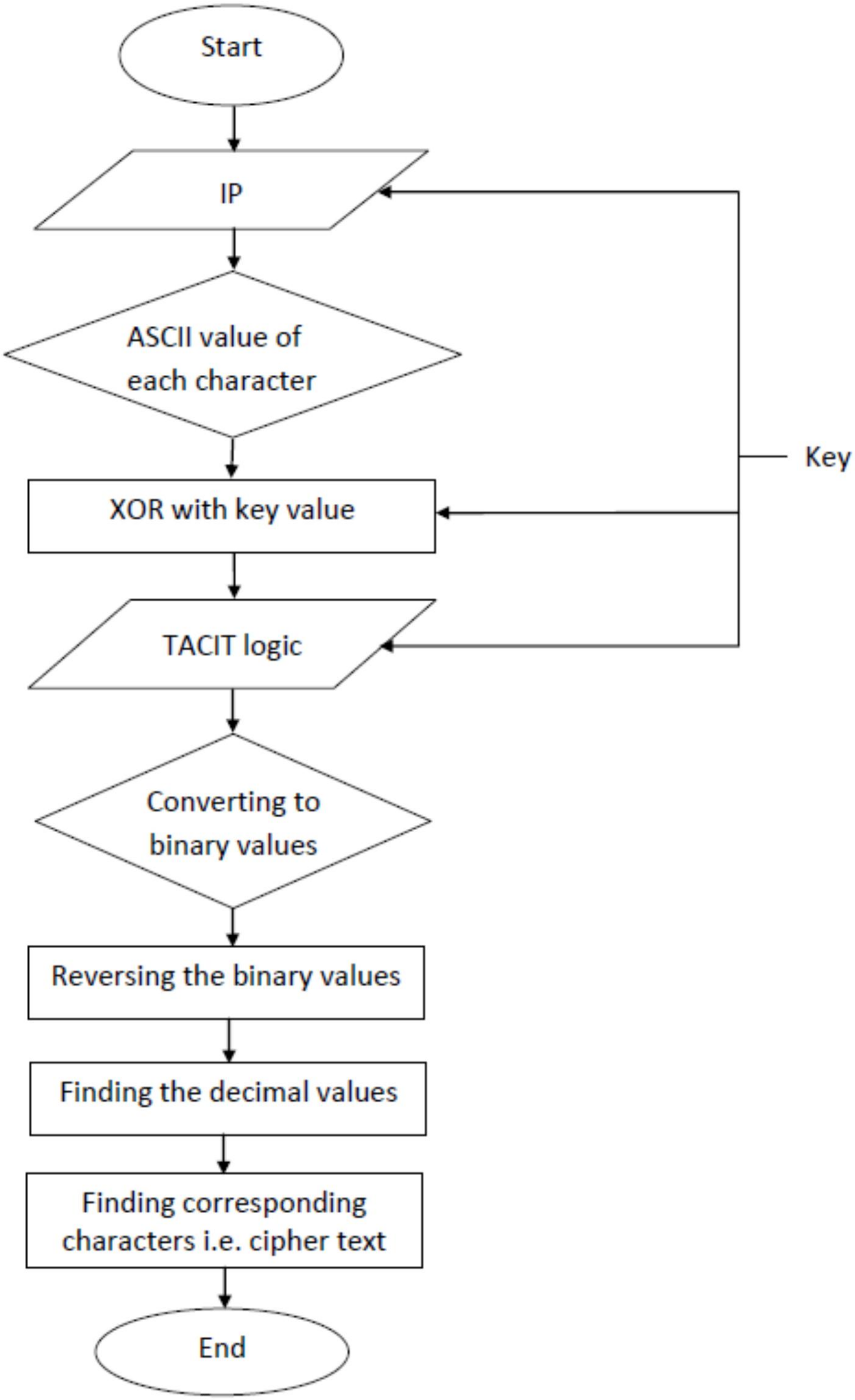

Start
IP
ASCII value of each character
XOR with key value
TACIT logic
Key
Converting to binary values
Reversing the binary values
Finding the decimal values
Finding corresponding characters i.e. cipher text
End

Fig. 4.4 Data encryption logic for TACIT Algorithm [84]

Step 1: Text file content is read and position of the character is shuffled by using initial permutation approach using key value.

Step 2: Read the character from the text file corresponding to the text and get the ASCII value of that character.

Step 3: Perform XOR operation with the specific n-bit key value.

Step 4: A secure tacit logic has been introduced (i.e. n^k xor k^k along with some specific operations; where n is the value computed from step 3).

Step 5: Convert the value into binary one.

Step 6: Perform reverse operation on the binary string.

Step 7: Corresponding decimal value is found.

Step 8: The Unicode character corresponds to the decimal value is formed which is none other than the cipher text.

Step 9: Continue step 1 to 7 for the next characters of the file until End of File (EOF) is reached.

4.2.2 DATA DECRYPTION LOGIC FOR TACIT ALGORITHM

The decoding of same data is done at receiving end. The text which is encoded at transmitting end using TACIT encryption technique is converted into cipher text. The decryption algorithm [84] decodes the cipher text with the same key at the receiving end that follows the steps listed below. The corresponding flow chart of the decryption algorithm is shown in figure 4.5.

Step 1: Read the first character from the cipher text and get the corresponding decimal value of it.

Step 2: The corresponding binary value is evaluated and make the reverse of it.

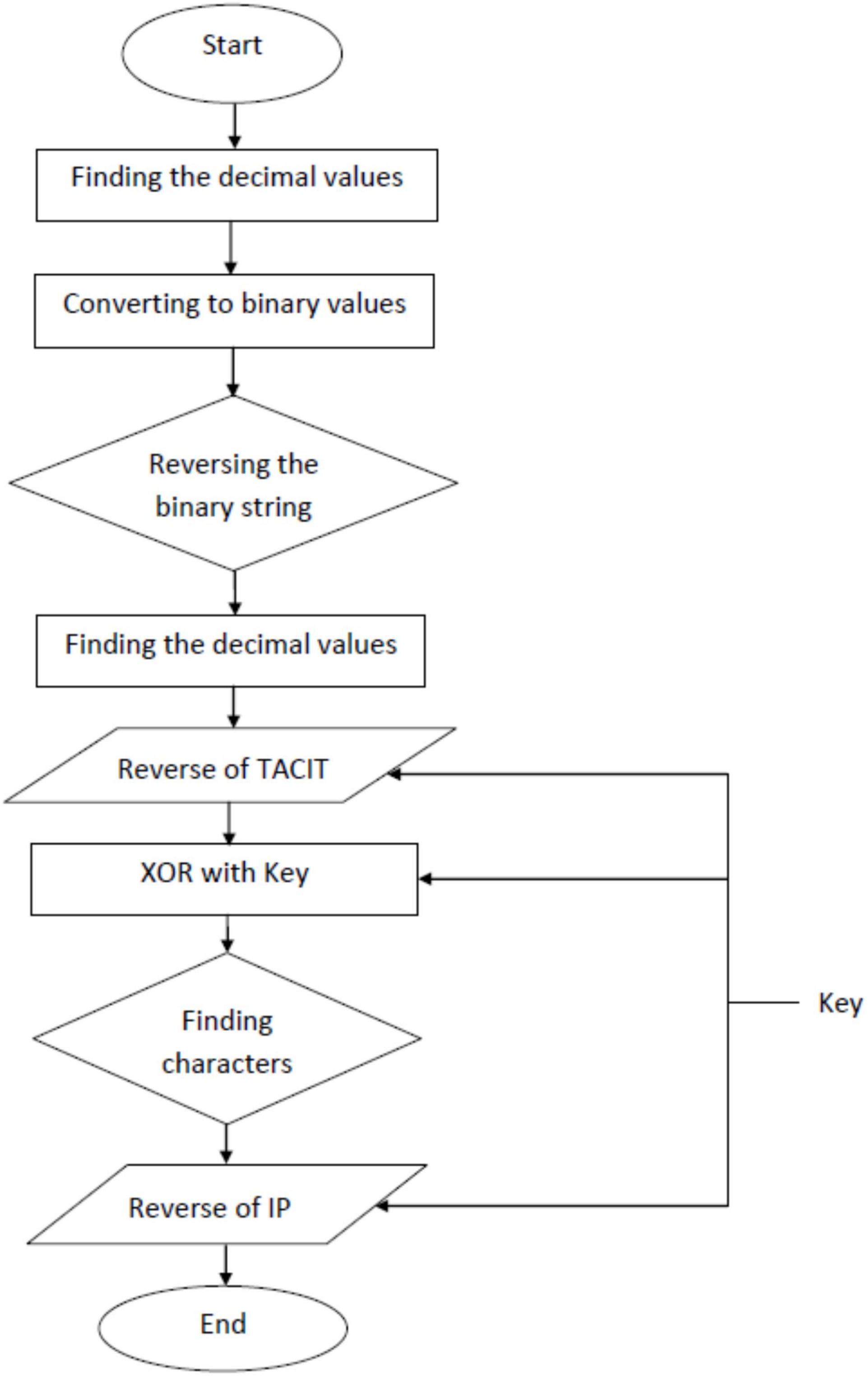

Start
Finding the decimal values
Converting to binary values
Reversing the binary string
Finding the decimal values
Reverse of TACIT
XOR with Key
Finding characters
Reverse of IP
End
Key

Fig. 4.5 Data decryption logic for TACIT Algorithm [84]

Step 3: Inverse of the tacit logic is applied.

Step 4: Perform XOR with n-bit key value.

Step 5: The character corresponds to it is determined.

Step 6: Now reshuffling is done using key value.

Step 7: Repeat the steps (1 to 6) till the end.

4.2.3 KEY MANAGEMENT POLICY

In secured network sender and receiver utilizes the same key but the distribution of key is a challenging task. A new technique is introduced to understand the key distribution policy [7, 84], considering that the key distribution takes place between A and B. At first, it is introduced two lookup tables of different dimensions, one dimensional and *m* x *n* dimensional along with a hash table shown in table 4.4, which is containing several hash functions.

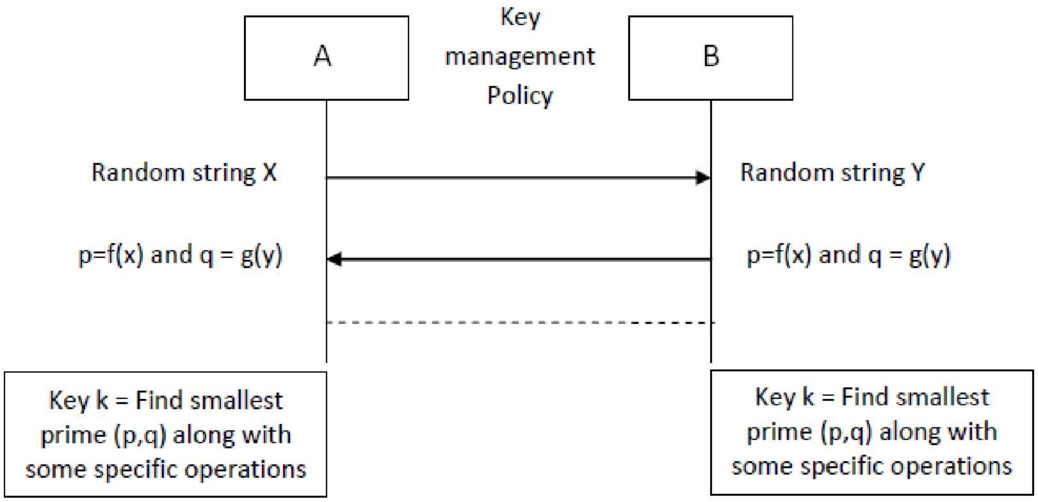

A
Key management Policy
B
Random string X
Random string Y
p=f(x) and q = g(y)
p=f(x) and q = g(y)
Key k = Find smallest prime (p,q) along with some specific operations
Key k = Find smallest prime (p,q) along with some specific operations

Fig 4.6 key distribution system

The first lookup table [48] only contains the numerical values and the second one contains some alphanumeric characters. Now, generate a random number at sender's end within a specified range say 0 to 9 and add this number in a code sequence which signifies a specific hash function includes from the hash table. After it, generate a sequence of random numbers on the basis of locations of the second lookup table and fetch the corresponding alphanumeric characters from the second lookup table on the basis of location chosen by the random number. Therefore, it is needed to add all those alphanumeric characters in the code sequence which forms a string *X*. Similarly, the string *Y* is generated from the receiver end.

- Following the steps in encryption and decryption technique, exchanging *X* and *Y* between the sender and receiver end takes place.
- At this stage both sender and receiver are familiar with *X* and *Y*. Both will calculate *p* and *q*, from *X* and *Y*, after applying specific operations on defined hash table.
- Now for finding the key *k*, the lowest prime number between *p* and *q* is to be used along with some specific techniques.
- In this case, if introducer gets *X* and *Y* by using possible combination of *X* and *Y,* he cannot get the key.

  Here,

  m = no. of lower case alphabetic character

  n = no. of numerical character

  u = no. of upper case alphabetic characters v = no. of special characters

Table 4.4 Hash function table

n	**Hash Functions**
0	$m^n - m.n$
1	$m^u + (m + u)$
2	$m^v - (u + v)$
3	$n^u + (v.m)$
4	$n^v + (n.m$
5	$n^m - m$
6	$u^m - m$
7	$u^n + (n + m - u)$
8	$u^v + (n + m + v - u)$
9	$m.n.v + (m.u)$

For an example Let X = 0mp#*DH@8976LjhR and Y = 7jkhLOUY%^&)678. After applying hash functions at both sender and receiver from the table has been defined, the value of p = 1004 and q = 259. The smallest prime number is 263 and the calculated value of key is 311.

4.3 DTMF SIGNALING

In telecommunication switching, the network is designed to carry voice signal [2] [8]. The network sends the voice signal to the phone company central office for the phone number to which it is intended to call. Moreover, it may connect to a long-distance carrier distinct from local service provider or an international call. DTMF generator sends the dialed number in the network and the same number is decoded on the receiving end based on the frequencies of touch keypad. It may connect user to some service such as entering the credit card number or account number, or to respond to certain questions by pressing buttons on the telephone keypad. The earlier versions of telephones used to have rotary type dials, which are now obsolete. All the landline and mobile phone handsets use pushbutton keypads, and the voice intercommunication occurs between two touch tone dial telephones as shown in figure 4.7.

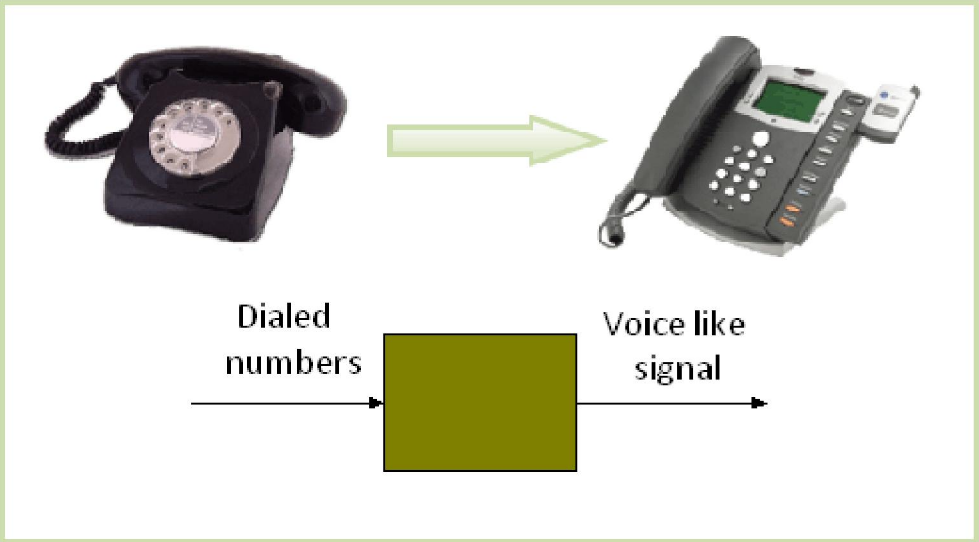


Fig. 4.7 Intercommunication between two touch keypad

It converts sequences of numerical digits into signals that will easily traverse through circuits which are designed for voice. DTMF signaling converts decimal digits (and the symbols '*' and '#') into sounds that share enough essential characteristics with voice to easily traverse circuits designed for voice. It is discussed an illustration of DTMF. As a phone number is in the entry box, and then type return, the heard sound is the DTMF signal for the number which is entered. The number can include any of the digits {0, 1, ... ,9} plus the symbols '*' and '#'. The discussed applet also understands the special symbol ',' (a comma), which will produce a pause of approximately one second. The DTMF coder is thus a function that maps a phone number into a voice like signal. Let *Digits* = {0, 1,..., 9, '*', '#'} are representing the set of digits that a telephone keypad can produce. Let *Indices* = {1,..., *N*} are an *N* digit phone number. Therefore, phone number is a function of digits and all sequences of digits are valid phone numbers.

$$\textit{Phone Number}: \textit{Indices} \rightarrow \textit{Digits}. \qquad \text{Equ. 4.1}$$

Therefore, the set of all valid *N* digit phone numbers is a subset of the set of '*N'* digit sequences,

$$\textit{Phone Numbers} \subset [\textit{Indices} \rightarrow \textit{Digits}] \qquad \text{Equ. 4.2}$$

And sound is function

$$\textit{Sound}: \textit{Time} \rightarrow \textit{Pressure}. \qquad \text{Equ. 4.3}$$

The set of all the sounds is a function space

$$\textit{Sounds} = [\textit{Time} \rightarrow \textit{Pressure}] \qquad \text{Equ. 4.4}$$

The system is a function that maps a function *Phone Number* into a function *Sound*. Thus, DTMF signaling system is a function

$$\textit{DTMF}: \textit{Phone Numbers} \rightarrow \textit{Sounds}. \qquad \text{Equ. 4.5}$$

Both the domain and the range of this function are sets of functions.

4.3.1 DTMF TOUCH KEYPAD

DTMF is a system used for identifying the keys or the number dialed. The early telephone systems used pulse dialing or loop disconnect signaling, and replaced by Multi Frequency (MF) dialing. DTMF is a multi frequency tone dialing system used by the push button keypads in telephone and mobile sets to convey the number or key dialed by the caller. DTMF is enabling the long distance signaling over telephone lines of dialed numbers in voice frequency range. It has eliminated the need of telecom operator between the caller and the callee and evolved automated dialing in the telephone switching centers.

In DTMF a key is represented with a combination of two sine waves. Dual tones of DTMF are called row and column frequencies of keypad. DTMF is the global standard for audible tones that represents the digits on a phone keypad. The landline phones which are based on touch tone pad generate the corresponding DTMF tone for a key of dial pad. The landline phone systems can then listen and decode that tone to determine which key was pressed, and thus enables dialing. It is known as "Touchtone" phone formerly a registered trademark of AT&T. The Telephony Application Program Interface (TAPI) provides a way for a program to detect DTMF digits. DTMF signaling transformed each digit into a pair of tones. There are four frequencies associated with the four rows, and four frequencies are associated with the four columns. Each pressed key then specifies two frequencies. Resulting DTMF signal for that key is the sum of two sinusoidal waves, one at each frequency. As an example, the digit '4' translates into a sound with two frequencies, one at 770 Hz. and the other at 1209 Hz. The touch keypad is shown in figure 4.8 and the frequency generation scheme with each key in upper and lower band is listed in the table 4.5.

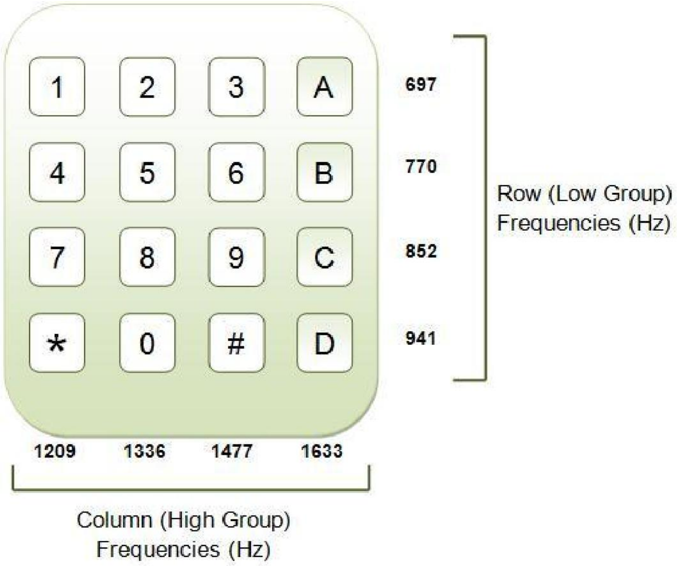

1
2
3
A
697
4
5
6
B
770
7
8
9
C
852
*
0
#
D
941
Row (Low Group) Frequencies (Hz)
1209
1336
1477
1633
Column (High Group) Frequencies (Hz)

Fig. 4.8 Touch keypad [3]

Table 4.5 Frequency generation of touch keypad

Digit	Low Frequency	High Frequency
1	697 Hz	1209 Hz
2	697 Hz	1336 Hz
3	697 Hz	1477 Hz
A	697 Hz	1633Hz
4	770 Hz	1209 Hz
5	770 Hz	1336 Hz
6	770 Hz	1477 Hz
B	770 Hz	1633Hz
7	852 Hz	1209 Hz
8	852 Hz	1336 Hz
9	852 Hz	1477 Hz

C	852 Hz	1633Hz
0	941 Hz	1209 Hz
*	941 Hz	1336 Hz
#	941 Hz	1477 Hz
D	941 Hz	1633Hz

4.3.2 BINAY ENCODING OF TOUCH KEYPAD

The row frequencies are called lower group frequencies and column frequencies are high group frequencies [2]. It prevents misinterpretation of the harmonics. The frequencies of DTMF are chosen such that no frequency has harmonic relationship with the others. At the same time mixing of frequencies should result in frequencies that could mimic another valid tone. High group frequencies are slightly louder than low group frequencies to compensate high frequency roll off of voice audio systems. Table 4.5 shows the dialing scheme for the DTMF signaling.

The level of each of these two signaling frequencies is within the range, -27 dB to -5 dB, and the difference in level of these two signaling frequencies is not more than 6 dB. Most DTMF decoders can process at least 10 tones per second under the worst conditions to meet the signal strength. Therefore, DTMF can easily convey 40 (10 x 4) bits or 5 bytes of data per second which is good for ac communication modem [10]. It can operate nearly 600 times faster 28,800 bits per second. Also, it is not necessary, that the numbers and symbols on the keypad always match their equivalent binary values. For an example, ‘0’ on the keypad is represented in DTMF by a decimal value of “10” or binary value of “1010”. Similarly, ‘D’ on the keypad is represented in DTMF by a decimal value of 0 or binary value of “0000”. The binary code corresponding to the symbols in a DTMF keypad is listed in table 4.6.

Table 4.6 Binary coding scheme for each key

Decimal value	**Binary value**	**Keyboard symbol**
0	0000	D
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	0
11	1011	*
12	1100	#
13	1101	A
14	1110	B
15	1111	C

4.4 TIME DIVISION SWITCHING

In multistage electronic switching [2, 3] the control functions are performed by a computer or a processor. Hence these systems are called Stored Program Control (SPC). The switching system may be either space division switching or time division switching. Time division switching may further be classified analog or digital switching. Digital switching [46] is again divided into two parts, space division and time division. If the values are stored, and transferred to the output at a later time interval, the technique is called time switching. In this design, the data coming in through the inlets are written into the data memory, and later read out to the appropriate outlets. The incoming and out coming data is usually in serial form, whereas the data are written into and read out of the memory in parallel form. It therefore, becomes necessary to perform serial to parallel conversion, and parallel to serial conversion at the inlets and outlets respectively [42] [91]. For convenience, in and data out parts of the Memory Data Register (MDR) are shown in figure 4.9, separately for the data memory. Since there is only one MDR, a gating mechanism is necessary to connect the required inlet/outlet to MDR [11] [12]. This is done by the in-gate and out-gates units. The information is not transferred in real time, and is first stored in the memory and later transferred to the outlet. There is a time delay between the acquisition of a sample from an inlet and its delivery to the corresponding outlet. This switching system can be controlled in following three ways [9] [11], sequential write/\random read, random write/sequential read and random write/random read.

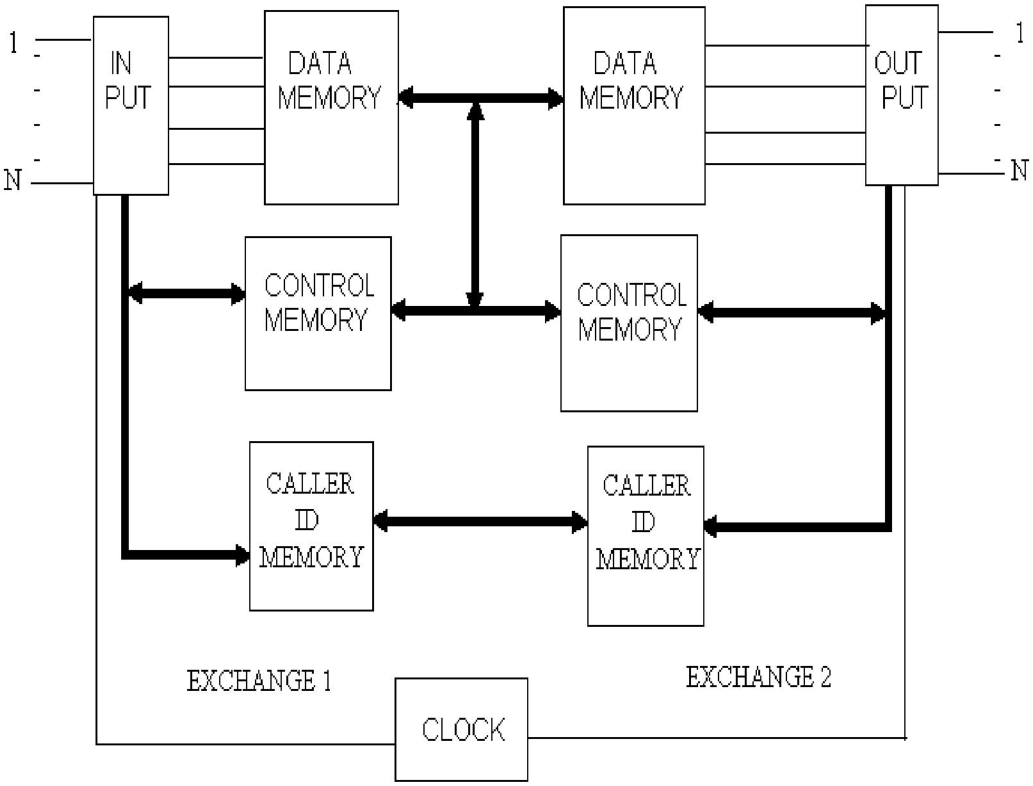


Fig. 4.9 Principle of time division switching

It has the features of caller ID facility. inter and intra exchange communication, 8 bits data transfer, inband signaling, synchronization clock, reset and dual way communication.

4.4.1 TDM INTEREXCHANE COMMUNICATION

The data is sent in packets and routed to the destination subscriber. This is shown with the help of 16 bits data format listed below to communicate the 16 subscribers between two exchanges. Each exchange is containing 8 subscribers, as the data format shown in figure 4.10, First bit in MSB, is the enable bit E, which is used to enable the subscriber to communicate, next bit in the frame, is used to designate the intercommunication or intracommunication. The value of I = 0, for intracommunication and I = 1, for Inter communication. Three bits are assigned for source subscriber and three bits are allocated for destination subscriber addresses. $D_2D_1D_0$ are 3 bits address for destination subscribers and $S_2S_1S_0$ are the address bits for source subscribers.

E	I	D2	D1	D0	S2	S1	S0	X	X	X	X	X	X	X	X

Fig.4.10 Packet format for inter and intra communication

- *Inter-Exchange:*8 Users in each exchange can communicate with each other.
- *Intra-Exchange:* 8 Users in one exchange can communicate with each other.
- *Data Memory:* Used to store the 8 bits data from D_0 to D_7 (8 locations) represented by XXXXXXX in the data format.
- *Control Memory:* Used to store the information of destination subscribers (8 locations).
- *Caller ID memory*: Used to store the information of source subscribers (8 locations).

4.4.2 WORKING SPECIFICATINS OF TDM

The switching among inlets and outlets is taken place under the two phases.

Phase 1: Input Subscribers in both the exchanges are scanned sequentially. It takes 8 clock cycles to scan 16 subscribers in order to know their status, that is if they want to transmit or not. This is called a sequential scanning. The data is to be transmitted is stored in data memory in sequential order. The information relating to the called subscriber is stored in control memory in sequential order and caller id number is stored in the caller id memory in the same way. Thus this system is a sequential write.

Phase 2: When all the scanning is done, the location of the data memory is read according to the corresponding location of the control memory. For example, if first location of data memory has data '*d'* and corresponding location in control memory is 3, this means that 'd' will be communicated to the 3rd user of the exchange, and the system is random read.

To decide where the data will be communicated, a bit is decided in the opcode as 'I' bit. If 'I' =1, then it is interexchange, i.e. the read out data will be given to user of other exchange. Thus communication between the subscriber of two exchanges can be made possible and hence the name interexchange. If 'I' = 0, then it is called as intraexchange, i.e. the read out data, will be given to the user of the same exchange. Thus communication between subscribers of the same exchange is made possible, and hence the name intraexchange. The exchange between caller id memories is done only if the particular user is enabled. Same is the case with data memory. A particular user is enabled if its opcode 16th bit is 1 and disabled if it is 0. So caller must be enabled and called must be disabled in order to make a call successful. The data packets contain the information of source subscriber, destination subscriber, enable bit and 8 bits of data transfer. When communication is taken place from source node to destination node, the entire packet of data will not transfer to the destination, only the 8 bits of data packet are transferred to the destination subscriber which are indicated by (0 to 7), as discussed in data packet format.

CHAPTER SUMMARY

To overcome the bus based communication network, 2D and 3D mesh topological structures are used with multistage networks. In the chapter, the 2D and 3D networks with 8 x 8 and 8 x 8 x 8 network configurations respectively are discussed. The address generation scheme of 2D network structure is based on row address and column address, to form a reconfigurable NoC. 3D network structure is configured in XY, YZ and ZX, structured 2D NoCs. The idea of 3D NoC is derived from distributed computing and large scale computer networks. The chapter describes the details on multistage network security, to protect the data when it is transferred over long distance, especially for internet. There are different algorithms available to provide network security like AES, DES, Triple DES, Triple AES, Kasumi, Blowfish, RSA, RC4, XMODES, but limited to their block size and key size of 128 bits maximum size. The new security algorithm is TACIT network security algorithm, in which block size and key size can vary of 'N' bits. If the key size is greater than the block size, TACIT network security can provide better results. The key management or key distribution policy is also discussed to find the key value using smallest prime number and hash function table. The telephone has to communicate with the phone company central office the phone number to which it is intended to call. The call processing is done trough DTMF touch tone dial phone, which has lower band and upper band frequencies corresponding to each dialed number. After dialing of the number, time division multiplexing technique is used to allocate the time for different users. The Principe of TDM switch is based on modulo counter which counts the number of users sequentially and time is allocated for a particular user. It has the features of caller ID facility inter and intra exchange, 8 Bit data transfer, inband signaling, synchronization clock, reset and dual way communication. The addresses of calling and called subscribers can be identified using the IP address in data packet format. After enabling the particular subscriber, it is possible to transfer ‘N’ bits data for intra and intercommunication, among telephone exchanges.

CHAPTER-5

EXPERIMENTAL AND SYNTHESIS ENVIRONMENT

The chapter explains the synthesis process carried out for the chip implementation on FPGA. It includes the synthesis on Digilent Virtex-5 FPGA supporting Xilinx Integrated System Environment (ISE). The chapter also describes the exponential set up used to verify the functionality of synthesized chip.

5.1 SYNTHESIS TOOL

The synthesis process is carried out on Xilinx Virtex – 5 XC5VLX110T [101] Digilent manufactured FPGA as shown in figure 5.1. It has two Xilinx XCF32P [101] platform flash ROMs for storing large device configurations of 32 MByte each, 64 bits wide 256 Mbyte DDR2 modules compatible with Embedded Development Kit (EDK) supported IP and software drivers. It has in board 32-bit synchronous Zero Bus Turnaround (ZBT) SRAM and Intel P30 Strata Flash [24]. It supports 10/100/1000 tri-speed Ethernet PHY supporting [24] [102] Media Independent Interface (MII), Gigabit Media Independent Interface(GMII), Reduced Gigabit Media Independent Interface (RGMII), and Serial Gigabit Media Independent Interface (SGMII), Universal Serial Bus (USB) host and peripheral controllers, programmable system clock generator [102]. It has Stereo Audio

Codec (SAC) 97 with line in, line out, headphone, microphone, and Sony/Philips Digital Interface Format (SPDIF) digital audio jacks, RS-232 port, 16 x 2 character LCD, I/O devices and ports [24].

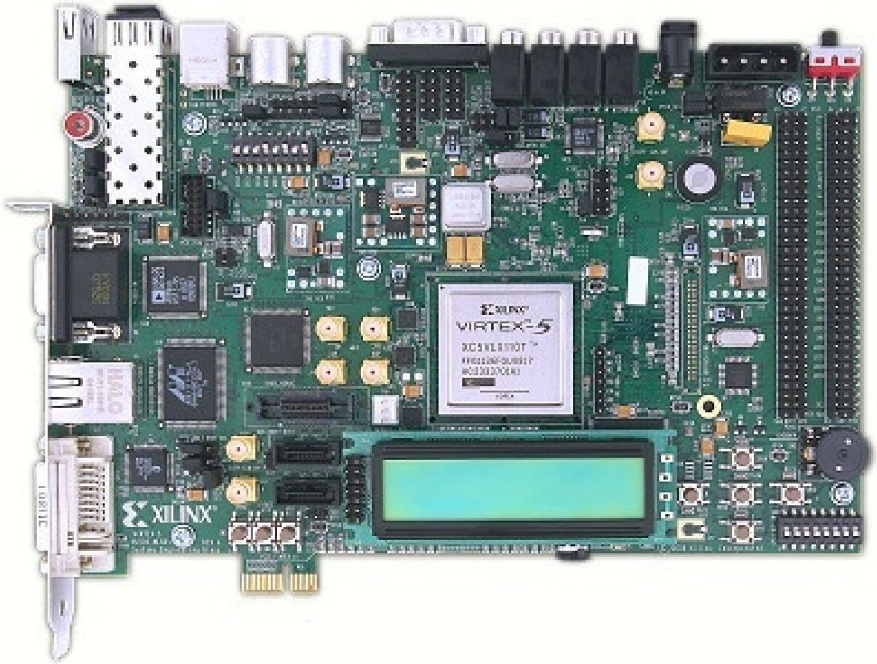


Fig. 5.1 Pictorial view of FPGA Virex -5 FPGA [102]

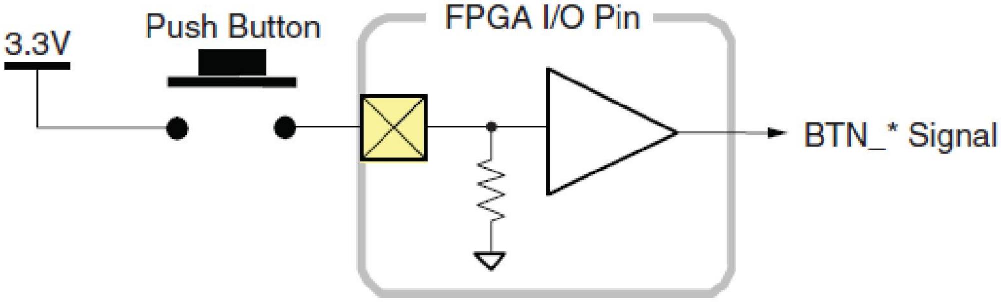


Fig. 5.2 Push button of FPGA [102]

Pressing a push button connects the associated FPGA pin to 3.3 V, as shown in figure 5.2. It uses an internal pull down resistor within the FPGA pin to generate a logic low, when the button is not pressed. Table 5.1 shows the method to specify a pull down resistor within the User Constrained file (UCF). Push buttons work on the principle of bouncing and debouncing to make and release the contacts with the board. The push buttons are also used for pin assignment for Reset, innode_address, out_node_address, including the I/O pins assignment for input and output data on I/O switches and LEDs respectively.

Table 5.1 UCF pin details in FPGA synthesis

Input Logic	**Pins**
Clk<>LOC	E6
Reset<> LOC	E9
Innode_address<2>LOC	DS22
Innode_address<1>LOC	DS21
Innode_address<0>LOC	DS20
Outnode_address<2>LOC	DS25
Outnode_address<1>LOC	DS24
Outnode_address<0>LOC	V10
NET "SW<8>" LOC	AC24
NET "SW<7>" LOC	AC25
NET "SW<6>" LOC	AE26
NET "SW<5>" LOC	AE27
NET "SW<4>" LOC	AF26
NET "SW<3>" LOC	AF25
NET "SW<2>" LOC	AG27

NET "SW<1>" LOC	U25
NET "LED<7>" LOC	AE24
NET "LED<6>" LOC	AD24
NET "LED<5>" LOC	AD25
NET "LED<4>" LOC	G16
NET "LED<3>" LOC	AD26
NET "LED<2>" LOC	G15
NET "LED<1>" LOC	L18
NET "LED<0>" LOC	H18

5.2 EXPERIMENTAL SET UP

The block diagram of an experimental set up is shown in figure 5.3. The experiment is carried out to validate the data transfer among inlets/outlets using Virtex -5 FPGA. Two 9-pin RS-232 [101] ports assist in the transmission of serial data to and fro from the FPGA board. 50 MHz clock oscillator is the system clock provides the clock signal to the various events taking place within the FPGA and the various programs that require clock for their working. A Digital clock manager [7] [8] can also be used to reduce the frequency of the system clock. This feature is useful for the task which needs smaller clock frequency [21]. On board USB based FPGA [9] download and debug interface is also present in the Virtex-5 kit where the programmable file is dumped into the FPGA via the USB based download cable. This feature is beneficial in the testing of the programs. There are 8 LEDs on the board, which glow based on the logic ‘High’ and logic ‘Low’ of output data, to justify the correctness of data transfer. Hence the LEDs can be interfaced to show the output of a single bit. Four slide switches and four push button switches are used to give the inputs to the FPGA board. They can also act as the reset switches for the various programs. The Kit also has four outputs, SPI based on board Digital to Analog Converter (DAC), which is interfaced to give the analog output to the digital data values. Two inputs, SPI based [7] [23] Analog to Digital Converter (ADC) with programmable gain preamplifier converts the real world analog signals into digital values.

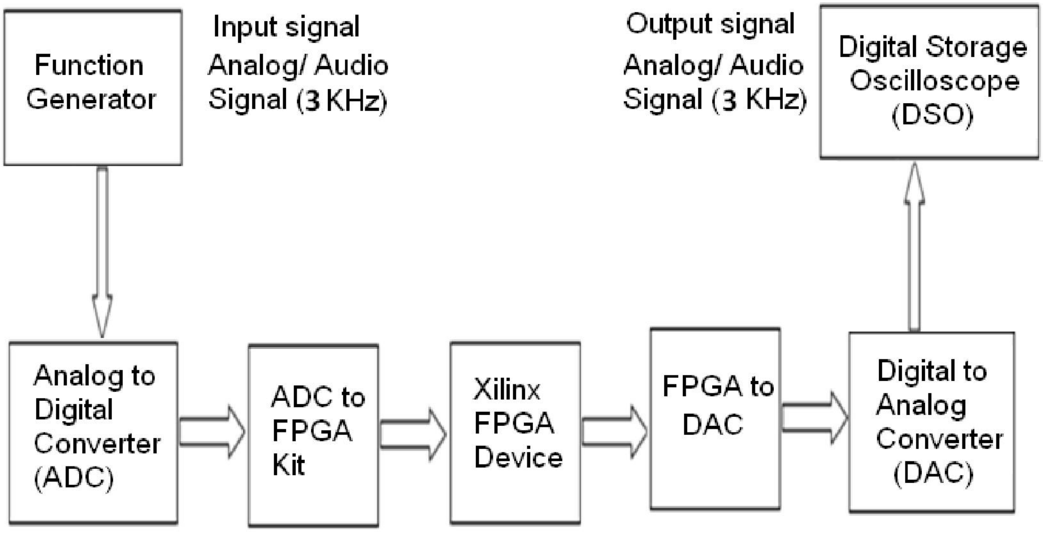

Function Generator
Input signal Analog/ Audio Signal (3 KHz)
Output signal Analog/ Audio Signal (3 KHz)
Digital Storage Oscilloscope (DSO)
Analog to Digital Converter (ADC)
ADC to FPGA Kit
Xilinx FPGA Device
FPGA to DAC
Digital to Analog Converter (DAC)

Fig. 5.3 Block diagram of experimental set up

The complete experimental setup and functional flow is shown in figure 5.4 and 5.5 respectively. An analog signal of audio frequency of 3 KHz is generated with the help of function generator. The analog signal is converted to digital signal using in built ADC in FPGA. The output of ADC is internally connected to FPGA kit. The synthesized program is then loaded into FPGA and is checked for the data transfer among inlets/outlets with the change in their addresses. The data transferred can be seen over FPGA kit using LEDs or LCD. The output of FPGA is given to Digital to Analog converter (DAC) and converted signal is displayed by Digital Storage Oscilloscope (DSO) as shown in figure 5.6. The displayed signal is of same frequency of 3 KHz. It shows that the data transferred over FPGA is correct and also suggest that the network on chip structure is feasible to validate the results.

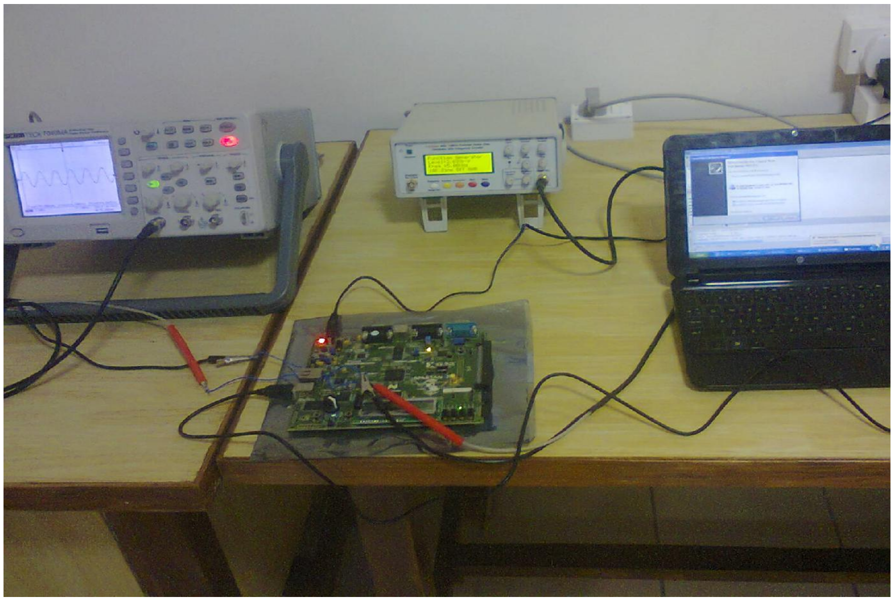


Fig. 5.4 Experimental set up

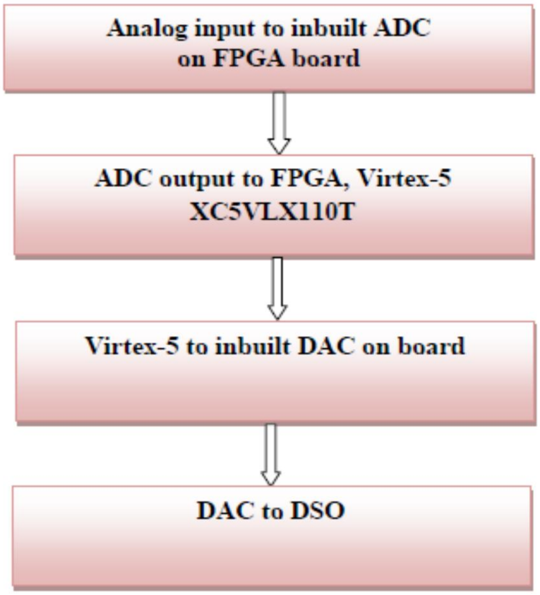

Analog input to inbuilt ADC on FPGA board
ADC output to FPGA, Virtex-5 XC5VLX110T
Virtex-5 to inbuilt DAC on board
DAC to DSO

Fig. 5.5 Flow of synthesis on FPGA board

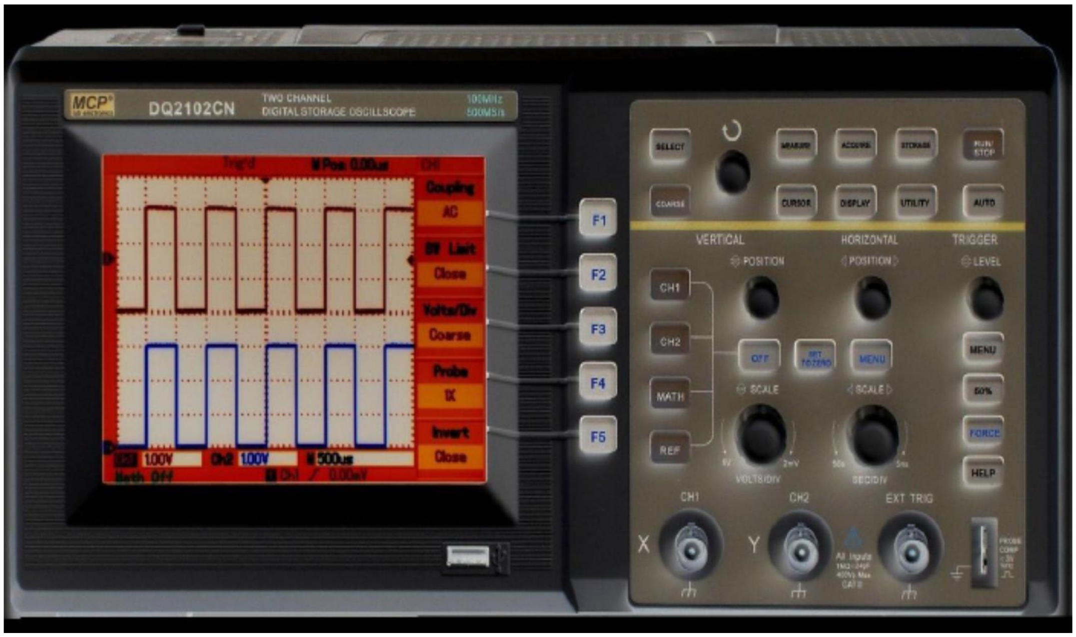


Fig. 5.6 ADC output on FPGA

5.2.2 VERIFICATION TEST CASES

The data transfer among inlets and outlets have been verified in single stage, two stage, three stage, four stage and five stage networks. It can be understood with the help of different test cases listed below.

Test Case 1: Single Stage Switching

Clk = Clk, Reset = 1/0, innode_address = “000”, outnode_address = “010”, input data is given on inlet using switches M_0 = “11001100”, Outlet data of N_2 is flashed on LEDs, N_2 = “11001100”.

Test Case 2: Two stage Switching

Clk = Clk, Reset = 1/0, innode_address = “001”, outnode_address= “011”, input data is given on inlet using switches M_1 = “00001111”, Outlet data of N_3 is flashed on LEDs, N_3 = “00001111”.

Test Case 3: Three Stage sSwitching

Clk = Clk, Reset = 1/0, innode_address = “011”, outnode_address = “111”, input data is given on inlet using switches M_3 = “10001000”, Outlet data of N_7 is flashed on LEDs, N_7 = “11001100”.

Test Case 4: Four Stage Switching

Clk = Clk, Reset = 1/0, innode_address = “111”, outnode_address = “001”, input data is given on inlet using switches M_7 = “11011000”, Outlet data of N_1 is flashed on LEDs, N_1 = “11011000”.

Test Case 5: Five Stage Switching

Clk = Clk, Reset = 1/0, innode_address = “010”, outnode_address = “110”, input data is given on inlet using switches M_2 = “00000011”, Outlet data of N_6 is flashed on LEDs, N_6 = “00000011”.

CHAPTER SUMMARY

The chapter describers the synthesis and verification process carried out on Design Under Test (DUT). The experimental set up supports the testing environment to validate the results. Different test cases have been tested using the experimental set up. Multistage switching among inlets and outlets is verified in single stage, two stage, three stage, four stage and five stage switching structures. The synthesis process is carried out on Xilinx Virtex – 5 XC5VLX110T Digilent manufactured FPGA board with Xilinx ISE 14.2. The data transfer from inlet is passed through the ADC channel, given to FPGA board, converted to DAC and displayed on DSO, which verifies the correct data transfer.

CHAPTER-6

RESULTS & DISCUSSIONS

The chapter explains the Xilinx simulation and syntheis results. It presents RTL view, internal schematics, device utilization and timing parameters. The functional simulation and data transfer scheme of the each staged switching system is done in Modelsim 10.1 b and corresponding results are discussed in the chapter. The chapter explains the details of simulation flow graphs and Register Transfer Level (RTL) details as chip view, internal schematics of single stage, two stage, three stage, four stage and five stage switching system. The synthesis results as device utilization and timing summary are also discussed with the comparison of each stage. The target device onVirtex-5 FPGA is, xc5vlx20t-2-ff323 and the results are carried out based on the same device synthesis and timing report.

6.1 RTL VIEW OF STAGED NETWORKS

RTL view of the chip is a top view representation depicting its pins details and input/ output logic. The possible inputs and ouputs used in the development of the chip are represented with their RTL view. Figure 6.1(a) and (b) represent the chip view and internal schematic of single stage switchng network. Similarly, the RTL view and internal schematics of two stage, three stage, four stage and five stage switching networks are shown in figure 6.2(a) & (b), 6.3(a) & (b), 6.4(a) & (b) and 6.5(a) & (b). The details of the pins of the staged network is discussed in table 6.1. In all the figures, inputs M_0, M_1, M_2, M_3, M_4, M_5, M_6, and M_7, clk, reset, and outputs N_0, N_1, N_2, N_3, N_4, N_5, N_6, and N_7 are kept common because the network structure is assumed of same congiguration in all the staged networks. All the chips are synchronised with the help of clock signal and reset. The data transfer is taken place on the rising edge of the clock pulse and all inlets and outlets working can be checked on the data packet arrival on the destination inlets and outlets.

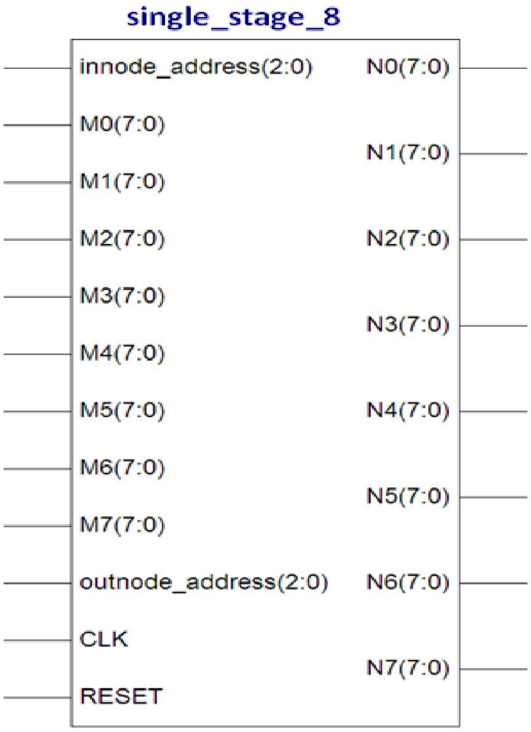


Fig. 6.1 (a) RTL view of single stage switching (8 x 8)

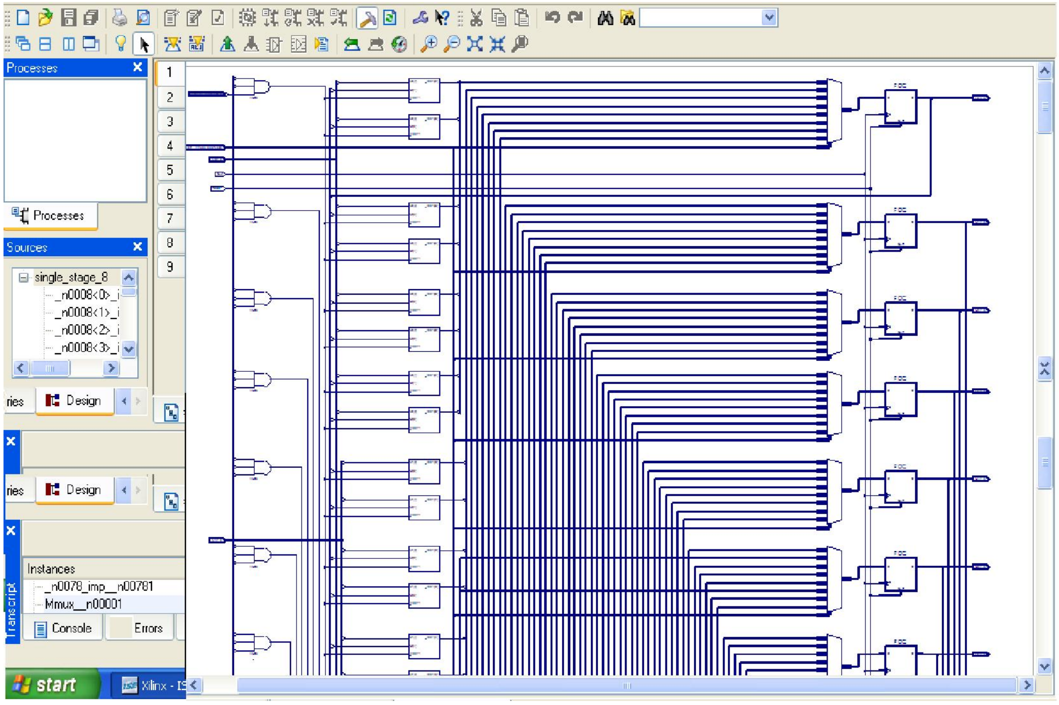


Fig. 6.1(b) internal schematic of single stage switching (8 x 8)

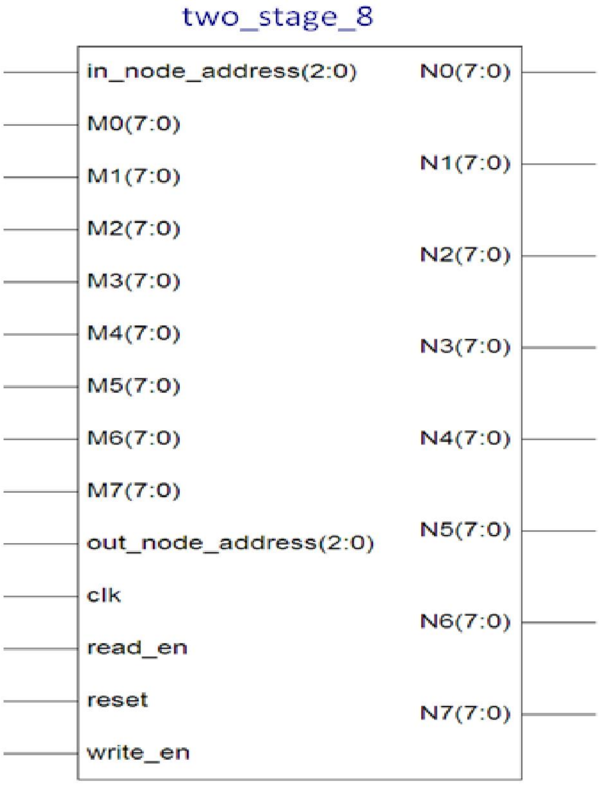


Fig. 6.2 (a) RTL view of two stage switching (8 x 8)

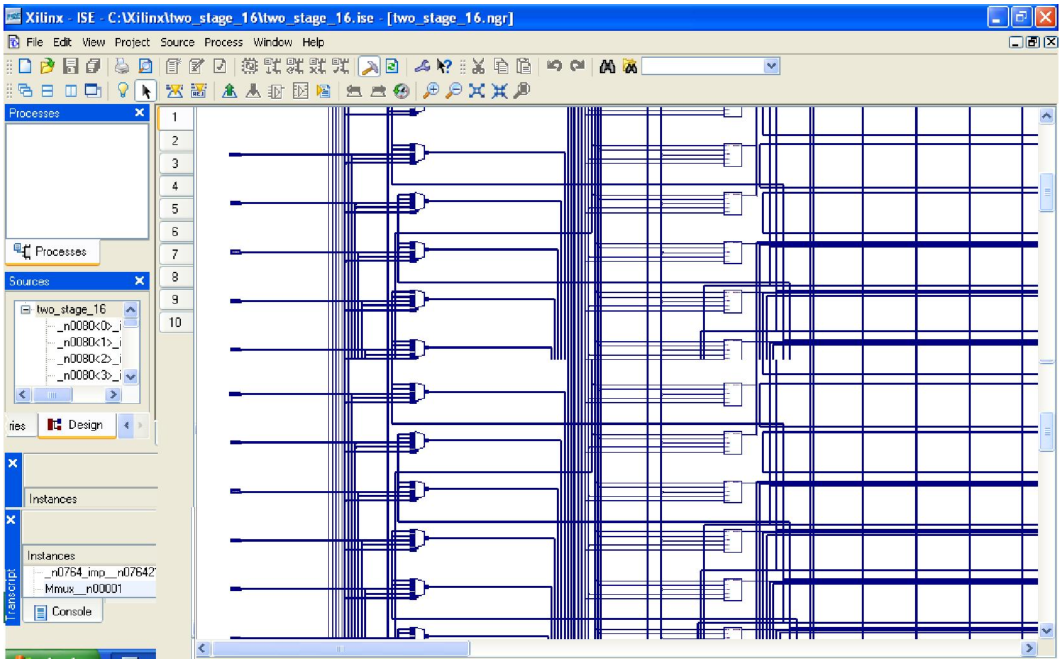


Fig. 6.2(b) internal schematic of two stage switching (8 x 8)

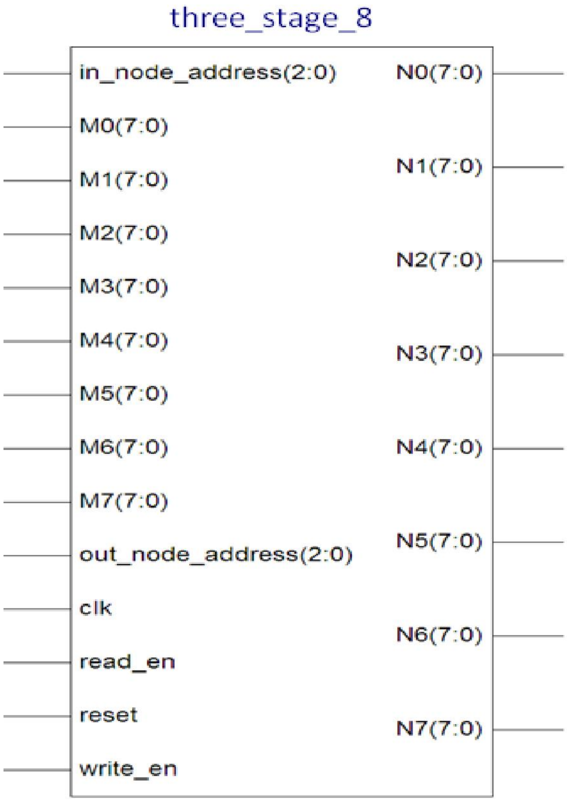


Fig. 6.3 (a) RTL view of three stage switching (8 x 8)

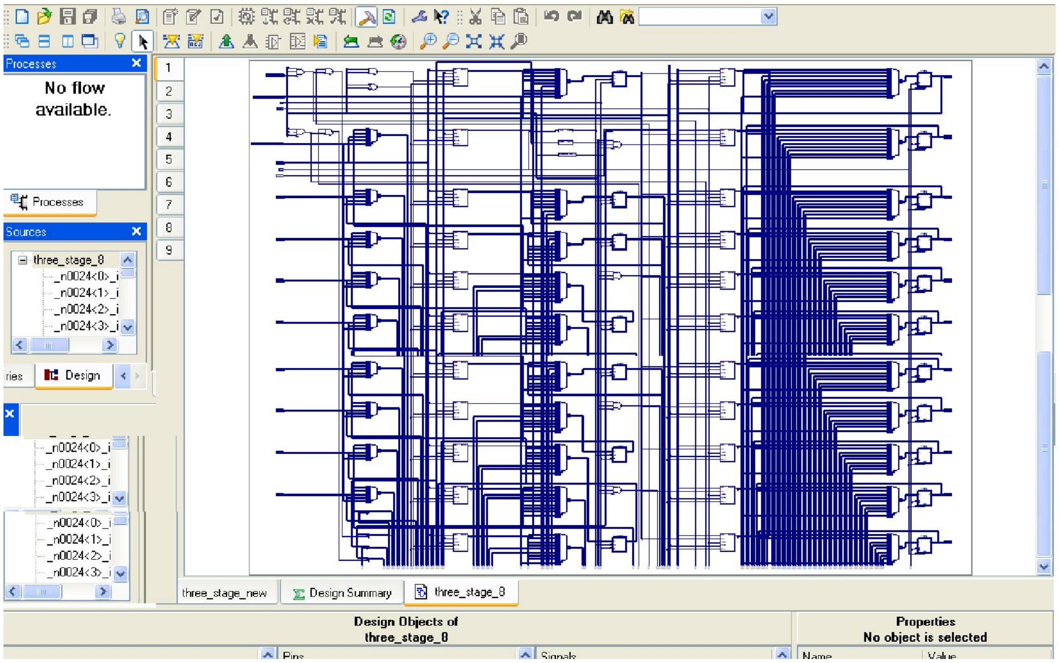


Fig. 6.3(b) internal schematic of three stage switching (8 x 8)

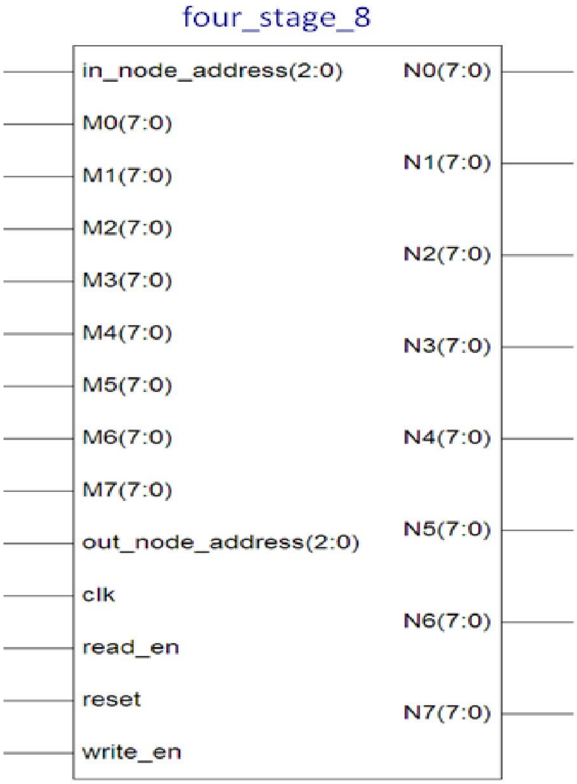


Fig. 6.4 (a) RTL view of four stage switching (8 x 8)

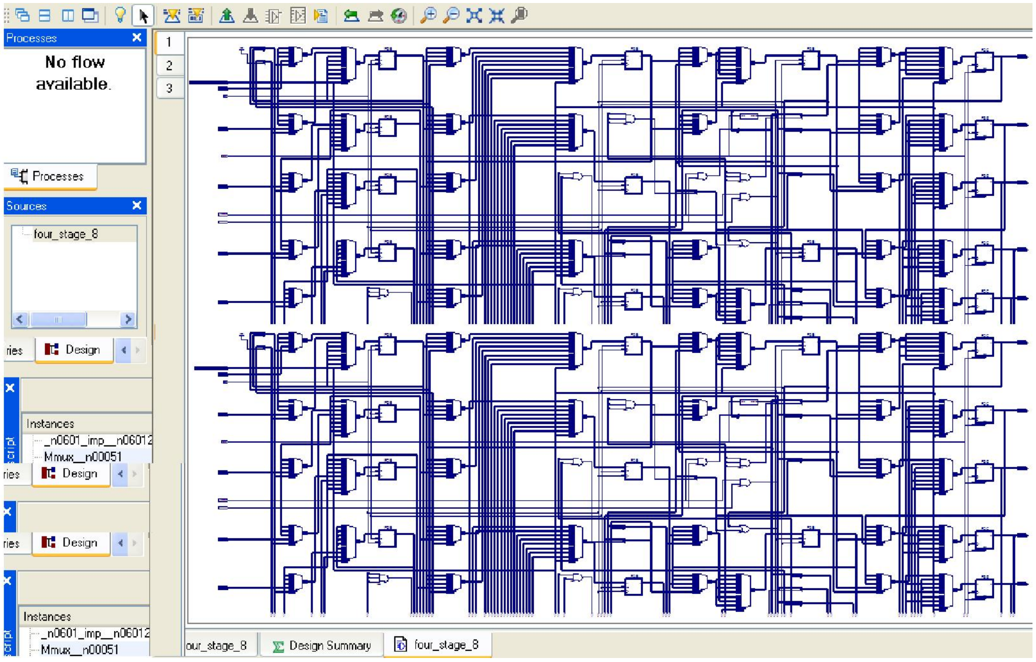


Fig. 6.4(b) internal schematic of four stage switching (8 x 8)

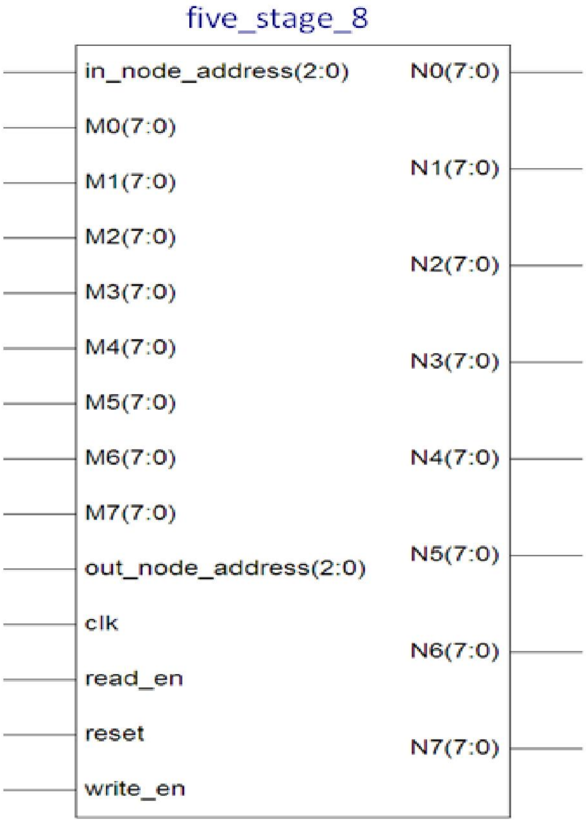


Fig. 6.5 (a) RTL view of five stage switching (8 x 8)

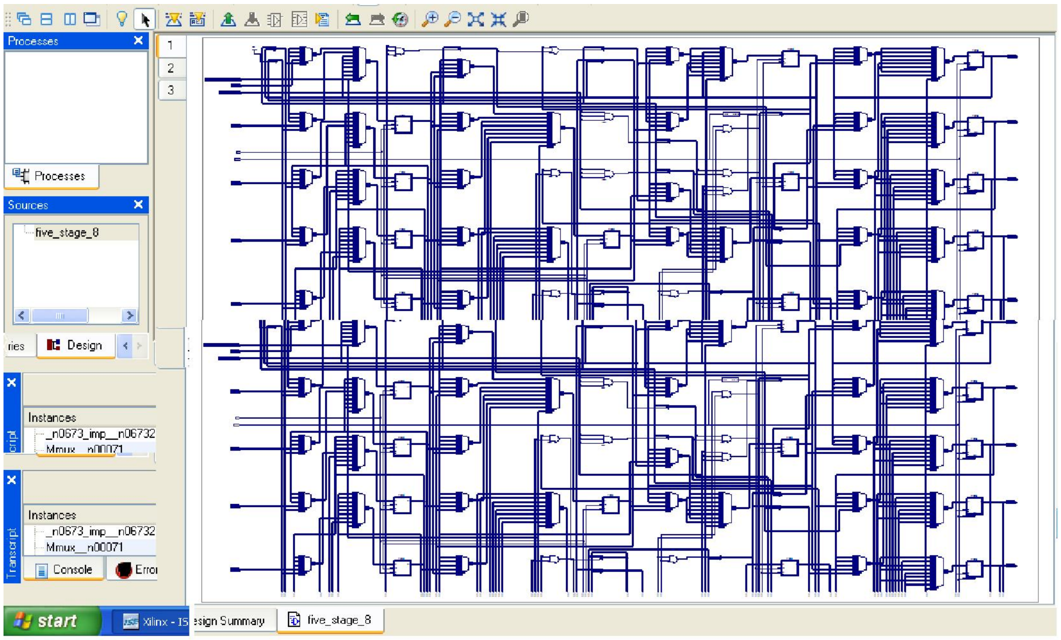


Fig. 6.5(b) internal schematic of five stage switching (8 x 8)

Table 6.1 Pin description of RTL view of multistage NoC

Pins	Functional Description
reset	Used to reset the memory contents zero for synchronization of the components by using clk of std_logic (1 bit)
clk	Default input for sequential logic to work on rising edge of clock pulse of std_logic (1 bit)
in_node_address [2:0]	Address of source subscribers or inlets of std_logic_vector (3 bits)
in_node_address [2:0]	Address of destination subscribers or outlets of std_logic_vector (3 bits)
M_0 [7:0]	Inlet with in_node_address = “000” carrying data of std_logic_vector (8 bits)

M_1 [7:0]	Inlet with in_node_address = “001” carrying data of std_logic_vector (8 bits)
M_2 [7:0]	Inlet with in_node_address = “010” carrying data of std_logic_vector (8 bits)
M_3 [7:0]	Inlet with in_node_address = “011” carrying data of std_logic_vector (8 bits)
M_4 [7:0]	Inlet with in_node_address = “100” carrying data of std_logic_vector (8 bits)
M_5 [7:0]	Inlet with in_node_address = “101” carrying data of std_logic_vector (8 bits)
M_6 [7:0]	Inlet with in_node_address = “110” carrying data of std_logic_vector (8 bits)
M_7 [7:0]	Inlet with in_node_address = “111” carrying data of std_logic_vector (8 bits)
N_0 [7:0]	Outlet with out_node_address = “000” receiving data of std_logic_vector (8 bits)
N_1 [7:0]	Outlet with out_node_address = “001” receiving data of std_logic_vector (8 bits)
N_2 [7:0]	Outlet with out_node_address = “010” receiving data of std_logic_vector (8 bits)
N_3 [7:0]	Outlet with out_node_address = “011” receiving data of std_logic_vector (8 bits)

N_4 [7:0]	Outlet with out_node_address = "100" receiving data of std_logic_vector (8 bits)
N_5 [7:0]	Outlet with out_node_address = "101" receiving data of std_logic_vector (8 bits)
N_6 [7:0]	Outlet with out_node_address = "110" receiving data of std_logic_vector (8 bits)
N_7 [7:0]	Outlet with out_node_address = "111" receiving data of std_logic_vector (8 bits)
read_en	control signal to read data from memory of std_logic (1 bit)
write_en	control signal to write data into memory of std_logic (1 bit)

6.2 SIMULATION RESULTS

The simulation results are taken from the Modelsim 10.1b software which shows the 8 bit data transfer among inlets and outlets. Figure 6.6 (a) (b), (c), (d) and (e) shows the flow diagrams of single stage, two stage, three stage, four stage and five stage data transfer scheme among inlets and outlets. The Modelsim simulation screenshots of corresponding stages are shown in figure 6.7(a), (b), (c), (d) and (e). In the flow diagrams, inlets are represented with $M_0[7:0]$, $M_1[7:0]$, $M_2[7:0]$, $M_3[7:0]$, $M_4[7:0]$, $M_5[7:0]$, $M_6[7:0]$ and $M_7[7:0]$ carrying 8 bits data, and outlets $N_0[7:0]$, $N_1[7:0]$, $N_2[7:0]$, $N_3[7:0]$, $N_4[7:0]$, $N_5[7:0]$, $N_6[7:0]$ and $N_7[7:0]$ also carrying 8 bit data. Inlets and outlets can communicate in full duplex mode. *in_node_address[2:0]* and *out_node_address[2:0]* are the addresses of inlets and outlets. *write_en* and *read_en* are the control signals considered for data writing in intermediate stages and reading out to appropriate outlets. The functional simulation depends on the test inputs in design. c*lk* and *reset* are used for the synchronization.

Step input 1: reset = ‘1’, clk is used for synchronization and then run. The rising edge of clock pulse is applied to check the results on the rising edge of applied clock pulse with 50% duty cycle.

Step input 2: reset = ‘0’, same clk is used for synchronization. Select the address of source and destination node, out_node_address. Force the eight bits value to any inlet.

Step input 3: write_en = ‘0’ and read_en = ‘1’ and run. The desired output on corresponding outlet is achieved. The writing and reading operations are carried to read and write the data on appropriate outlets / inlets.

In full duplex mode, inlets M_0[7:0], M_1[7:0], M_2[7:0], M_3[7:0], M_4[7:0], M_5[7:0], M_6[7:0], M_7[7:0] and outlets N_0[7:0], N_1[7:0], N_2[7:0], N_3[7:0], N_4[7:0]are having the data. Based on the address identification, the data is transferred to the appropriate inlets to outlets or vice versa. In single stage, two stage, three stage, four stage and five stage multistage networks the data packet is routed based on the maximum availability of switching connections. The functional simulation guarantees the successful data transfer in the network and FPGA synthesis results guarantees the hardware feasibility of the simulated chips to fabrication foundries.

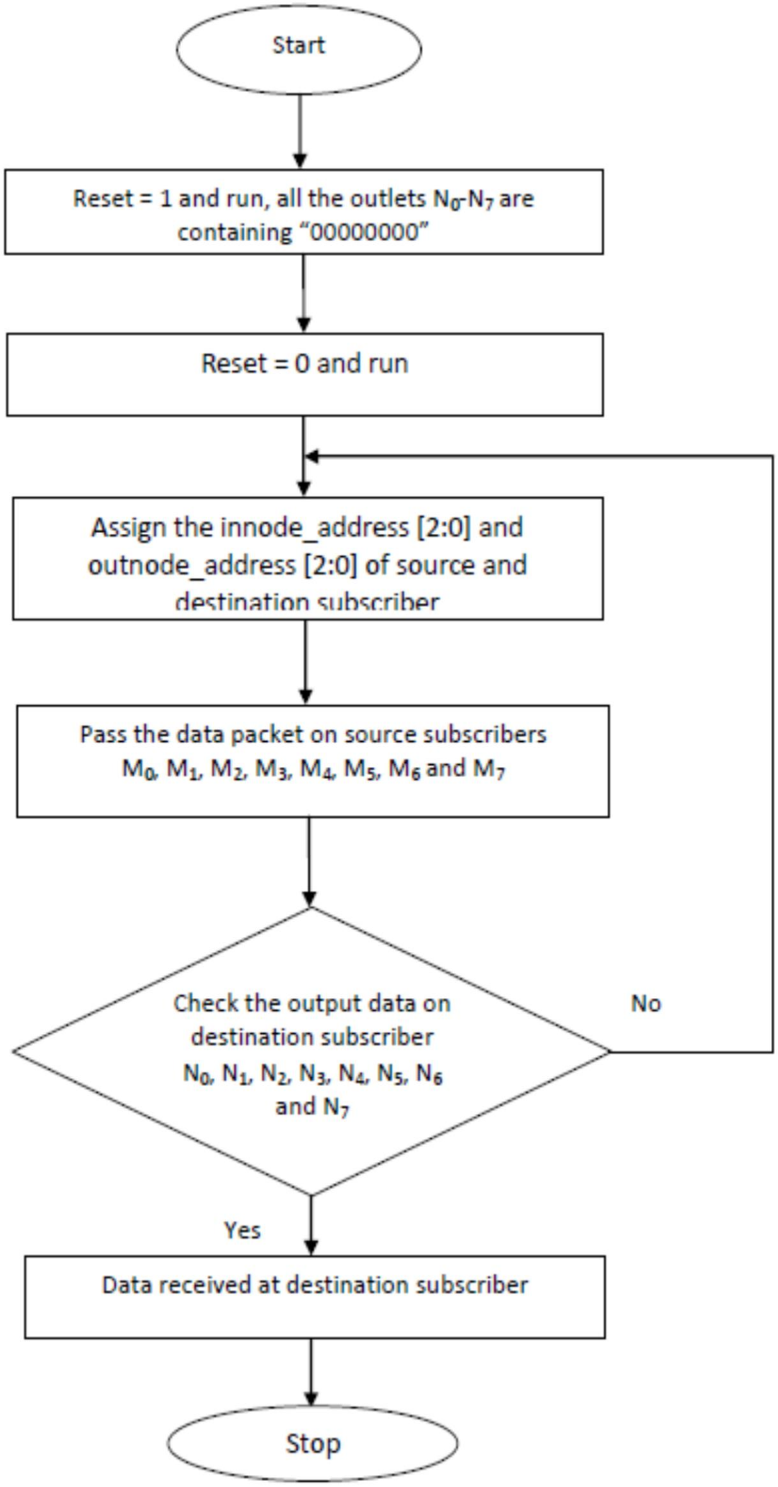

Start
Reset = 1 and run, all the outlets N_0-N_7 are containing "00000000"
Reset = 0 and run
Assign the innode_address [2:0] and outnode_address [2:0] of source and destination subscriber
Pass the data packet on source subscribers M_0, M_1, M_2, M_3, M_4, M_5, M_6 and M_7
Check the output data on destination subscriber N_0, N_1, N_2, N_3, N_4, N_5, N_6 and N_7
No
Yes
Data received at destination subscriber
Stop

Fig. 6.6 (a) Flow chart of single stage switching

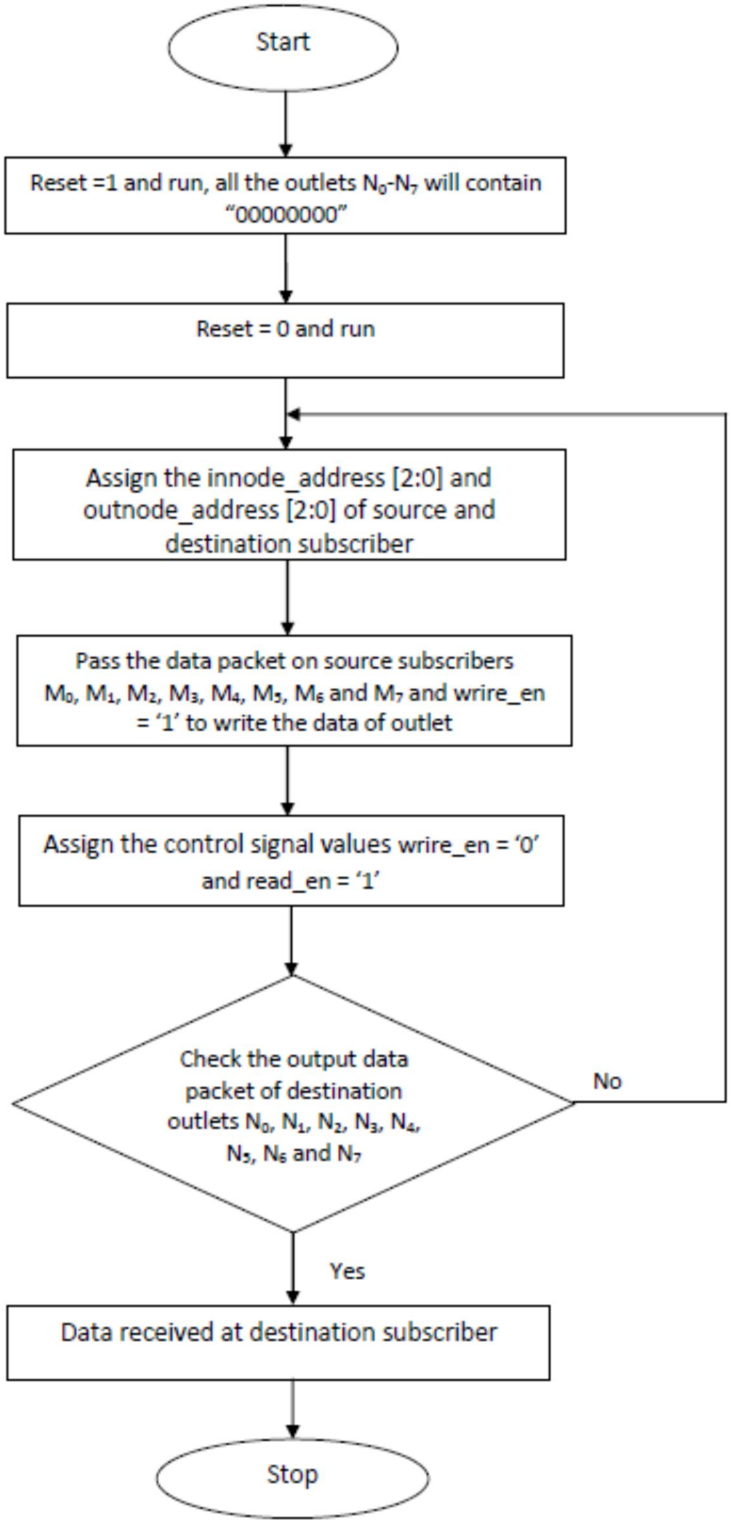

Start
Reset =1 and run, all the outlets N_0-N_7 will contain "00000000"
Reset = 0 and run
Assign the innode_address [2:0] and outnode_address [2:0] of source and destination subscriber
Pass the data packet on source subscribers M_0, M_1, M_2, M_3, M_4, M_5, M_6 and M_7 and wrire_en = '1' to write the data of outlet
Assign the control signal values wrire_en = '0' and read_en = '1'
Check the output data packet of destination outlets N_0, N_1, N_2, N_3, N_4, N_5, N_6 and N_7
No
Yes
Data received at destination subscriber
Stop

Fig. 6.6 (b) Flow chart of two stage switching

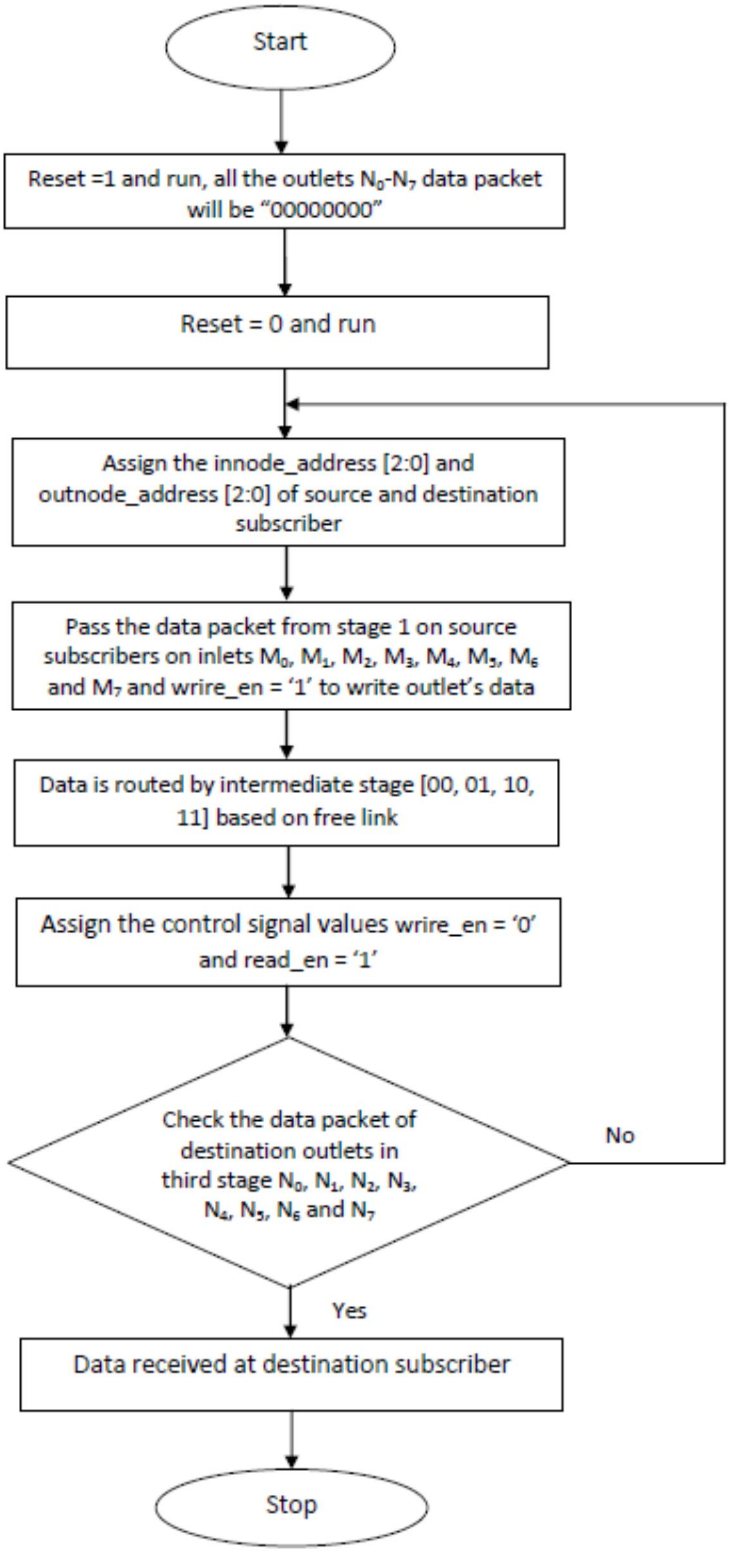

Start
Reset =1 and run, all the outlets N_0-N_7 data packet will be “00000000”
Reset = 0 and run
Assign the innode_address [2:0] and outnode_address [2:0] of source and destination subscriber
Pass the data packet from stage 1 on source subscribers on inlets M_0, M_1, M_2, M_3, M_4, M_5, M_6 and M_7 and wrire_en = ‘1’ to write outlet’s data
Data is routed by intermediate stage [00, 01, 10, 11] based on free link
Assign the control signal values wrire_en = ‘0’ and read_en = ‘1’
Check the data packet of destination outlets in third stage N_0, N_1, N_2, N_3, N_4, N_5, N_6 and N_7
No
Yes
Data received at destination subscriber
Stop

Fig. 6.6 (c) Flow chart of three stage switching

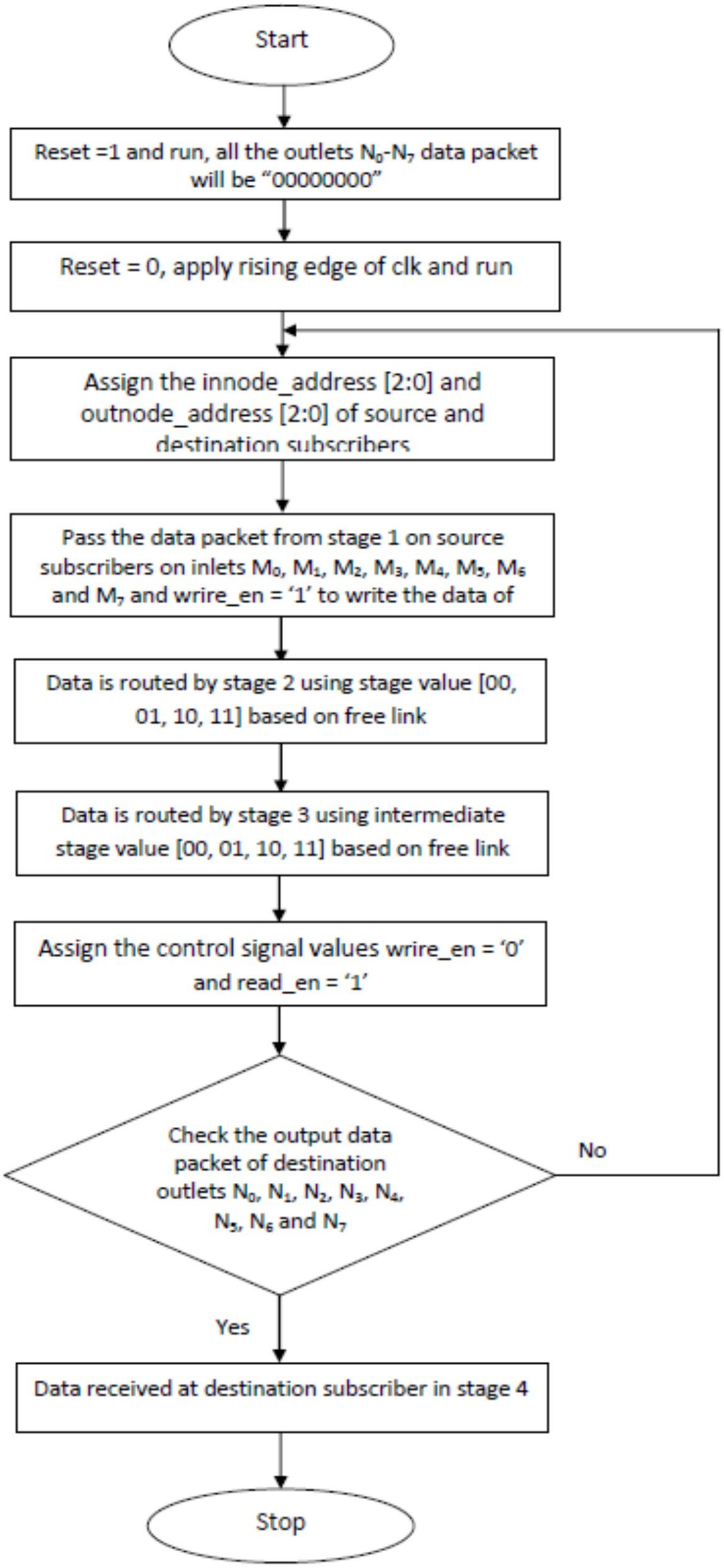

Start
Reset =1 and run, all the outlets N0-N7 data packet will be "00000000"
Reset = 0, apply rising edge of clk and run
Assign the innode_address [2:0] and outnode_address [2:0] of source and destination subscribers
Pass the data packet from stage 1 on source subscribers on inlets M0, M1, M2, M3, M4, M5, M6 and M7 and wrire_en = '1' to write the data of
Data is routed by stage 2 using stage value [00, 01, 10, 11] based on free link
Data is routed by stage 3 using intermediate stage value [00, 01, 10, 11] based on free link
Assign the control signal values wrire_en = '0' and read_en = '1'
Check the output data packet of destination outlets N0, N1, N2, N3, N4, N5, N6 and N7
No
Yes
Data received at destination subscriber in stage 4
Stop

Fig. 6.6 (d) Flow chart of four stage switching

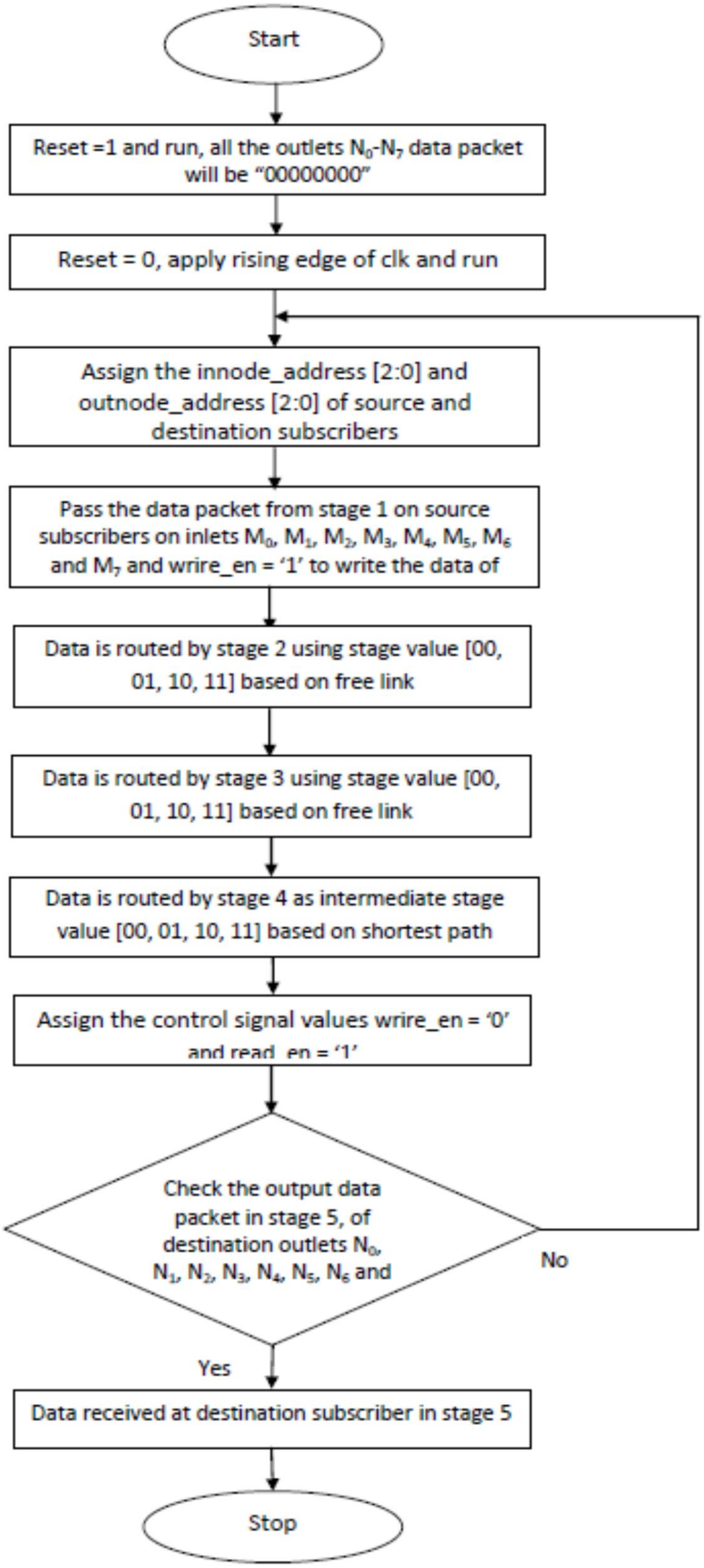

Start
Reset =1 and run, all the outlets N_0-N_7 data packet will be “00000000”
Reset = 0, apply rising edge of clk and run
Assign the innode_address [2:0] and outnode_address [2:0] of source and destination subscribers
Pass the data packet from stage 1 on source subscribers on inlets M_0, M_1, M_2, M_3, M_4, M_5, M_6 and M_7 and wrire_en = ‘1’ to write the data of
Data is routed by stage 2 using stage value [00, 01, 10, 11] based on free link
Data is routed by stage 3 using stage value [00, 01, 10, 11] based on free link
Data is routed by stage 4 as intermediate stage value [00, 01, 10, 11] based on shortest path
Assign the control signal values wrire_en = ‘0’ and read_en = ‘1’
Check the output data packet in stage 5, of destination outlets N_0, N_1, N_2, N_3, N_4, N_5, N_6 and
No
Yes
Data received at destination subscriber in stage 5
Stop

Fig. 6.6 (e) Flow chart of five stage switching

6.2.1 TEST CASES OF FUNCTIONAL SIMULATION

The functional simulation of the developed chip for single, two, three, four and five stage multistage network is verified with the test vectors discussed below.

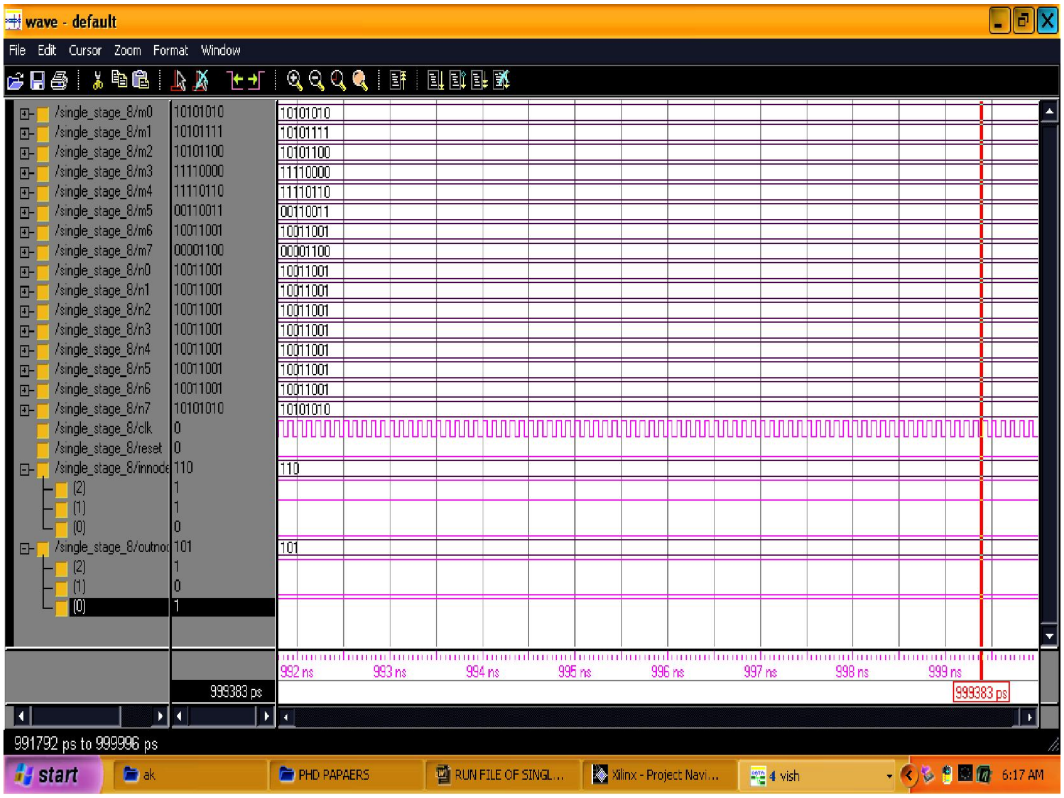
Test case 1: The test case 1 includes the testing inputs of single stage switching network. When reset = ‘1’ then outlets N_0 = “00000000”, N_1 = “00000000”, N_2 = “00000000”, N_3 = “00000000”, N_4 = “00000000”, N_5 = “00000000”, N_6 = “00000000”, N_7 = “00000000”. When reset = ‘0’, the packet data for inlets are M_0 = “10101010”, M_1 = “10101111”, M_2 = “10101100” M_3 = “11110000”, M_4 = “11110110”, M_5 = “00110011”, M_6 = “10001000”, M_7 = “00001100”, innode_address = “110”, outnode_address = “101”. In the simulation, the outlet N_5 gets the data transferred by inlet M_6. It shows that M_6 and N_5 are communicating to each other at positive edge of clock pulse.

Test case 2: The test case 2 includes the testing inputs of two stage switching network. When reset = ‘1’ then outlets N_0 = “00000000”, N_1 = “00000000”, N_2 = “00000000”, N_3 = “00000000”, N_4 = “00000000”, N_5 = “00000000”, N_6 = “00000000”, N_7 = “00000000”. When reset = ‘0’, the packet data for inlets are M_0 = “00001111”, M_1 = “00111100”, M_2 = “01111000”, M_3 = “11110000”, M_4 = “11110011”, M_5 = “01010101”, M_6 = “10101010”, M_7 = “00110000”, in_node_address = “100”, out_node_address = “111”. Control signal write_en = ‘1’ to write the data in outlet and read_en to read the data at same outlet. In the simulation, the outlet N_7 is getting the data transferred by inlet M_4, it means M_7 and N_4 are communicating to each other at positive edge of clock pulse.

Test case 3: The test case 3 includes the testing inputs of three stage switching network. When reset = ‘1’ then outlets N_0 = “00000000”, N_1 = “00000000”, N_2 = “00000000”, N_3 = “00000000”, N_4 = “00000000”, N_5 = “00000000”, N_6 = “00000000”, N_7 = “00000000”. When reset = ‘0’, the packet data for inlets are M_0 = “00001111”, M_1 = “00111100”, M_2 = “01111000”, M_3 = “11110000”, M_4 = “11110011”, M_5 = “01010101”, M_6 = “10101010”, M_7 = “00110000”, in_node_address = “100”, out_node_address = “111”. After the simulation, the outlet N_7 is getting the data transferred by inlet M_4. Control signal write_en = ‘1’ is to write the data in intermediate stage and read_en to read the data at outlet, it means M_7 and N_4 are communicating to each other at positive edge of clock pulse.

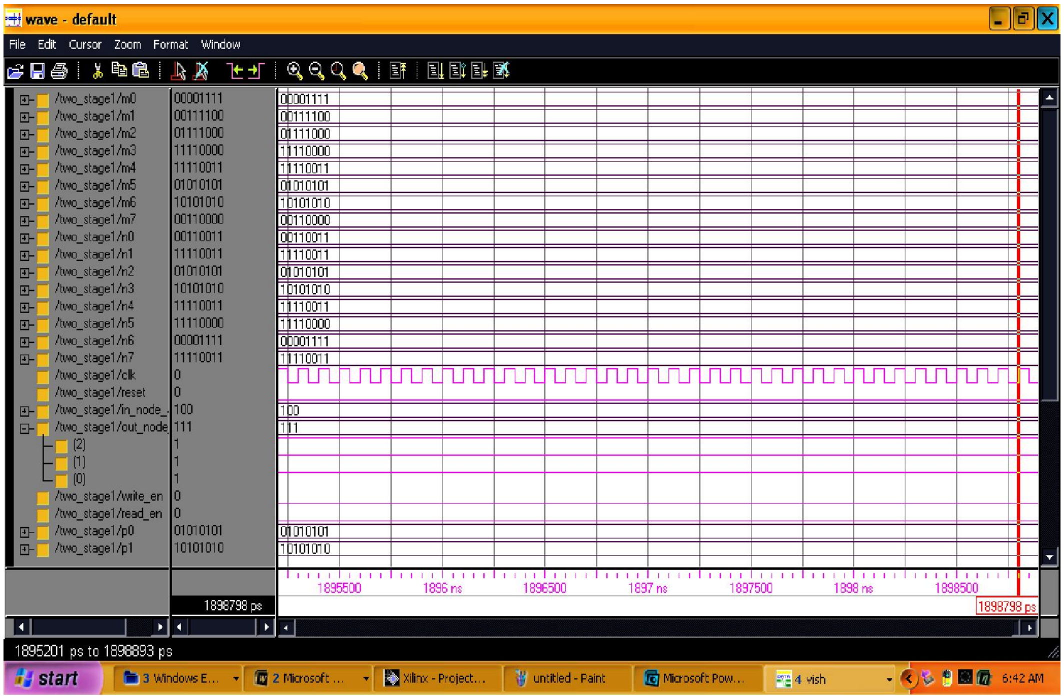
Test case 4: The test case 4 includes the testing inputs of four stage switching network. When reset = ‘1’ then outlets N_0 = “00000000”, N_1 = “00000000”, N_2 = “00000000”, N_3 = “00000000”, N_4 = “00000000”, N_5 = “00000000”, N_6 = “00000000”, N_7 = “00000000”. When reset = ‘0’, the packet data for inlets are M_0 = 11010111, M_1 = “00010000”, M_2 = “0001000”, M_3 = “00010010”, M_4 = “00010011”, M_5 = “00010100”, M_6 = “00010101”, M_7 = “00010101”, in_node_address = “010”, out_node_address = “110”. Control signal write_en = ‘1’ is to write the data in stage 2 and stage 3 from stage 1 and read_en to read the data at outlet of stage 4. After the simulation, the outlet N_6 is getting the data transferred by inlet M_2, it means M_2 and N_6 are communicating to each other at positive edge of clock pulse.

Test case 5: The test case 5 includes the testing inputs of five stage switching network. When reset = ‘1’ then outlets N_0 = “00000000”, N_1 = “00000000”, N_2 = “00000000”, N_3 = “00000000”, N_4 = “00000000”, N_5 = “00000000”, N_6 = “00000000”, N_7 = “00000000”. When reset = ‘0’, the packet data for inlets are M_0 = “00001100”, M_1 = “00001101”, M_2 = “00001110”, M_3 = “00001111”, M_4 = “00010000”, M_5 = “00010001”, M_6 = “00010010”, M_7 = “00010011”, in_node_address = “011”, out_node_address = “100”. Control signal write_en = ‘1’ is to write the data in stage 2 and stage 3 and stage 4 from stage 1 and read_en to read the data at outlet of stage 5. In the simulation, the outlet N_4 is getting the data transferred by inlet M_3, it means M_3 and N_4 are communicating to each other at positive edge of clock pulse.



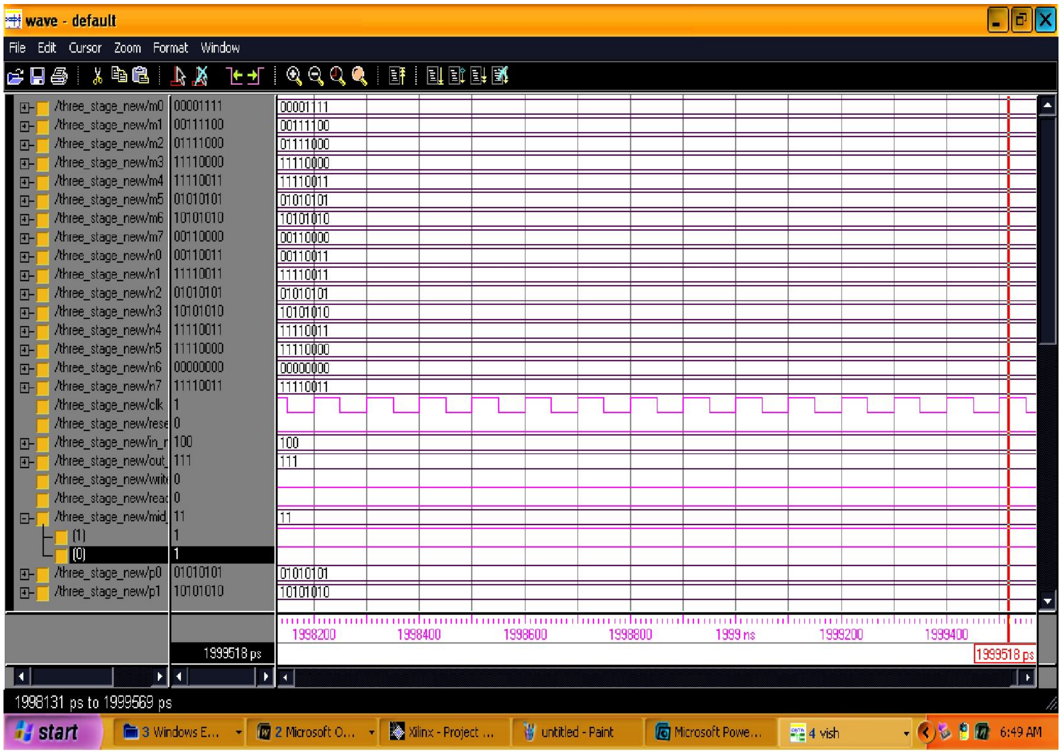
```
/single_stage_8/m0       10101010    10101010
/single_stage_8/m1       10101111    10101111
/single_stage_8/m2       10101100    10101100
/single_stage_8/m3       11110000    11110000
/single_stage_8/m4       11110110    11110110
/single_stage_8/m5       00110011    00110011
/single_stage_8/m6       10011001    10011001
/single_stage_8/m7       00001100    00001100
/single_stage_8/n0       10011001    10011001
/single_stage_8/n1       10011001    10011001
/single_stage_8/n2       10011001    10011001
/single_stage_8/n3       10011001    10011001
/single_stage_8/n4       10011001    10011001
/single_stage_8/n5       10011001    10011001
/single_stage_8/n6       10011001    10011001
/single_stage_8/n7       10101010    10101010
/single_stage_8/clk      0
/single_stage_8/reset    0
/single_stage_8/innode   110         110
   (2)                   1
   (1)                   1
   (0)                   0
/single_stage_8/outnod   101         101
   (2)                   1
   (1)                   0
   (0)                   1
                         999383 ps
991792 ps to 999996 ps
```

Fig. 6.7(a) Modelsim simulation of single stage switching (8 x 8)



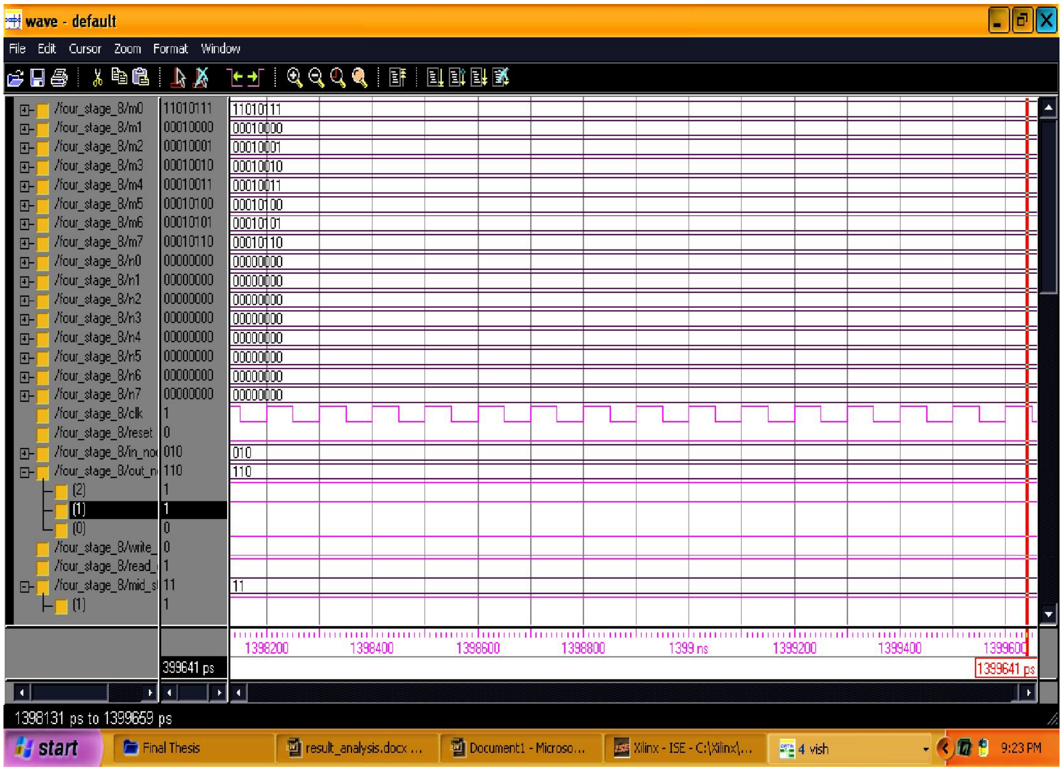
```
/two_stage1/m0          00001111
/two_stage1/m1          00111100
/two_stage1/m2          01111000
/two_stage1/m3          11110000
/two_stage1/m4          11110011
/two_stage1/m5          01010101
/two_stage1/m6          10101010
/two_stage1/m7          00110000
/two_stage1/n0          00110011
/two_stage1/n1          11110011
/two_stage1/n2          01010101
/two_stage1/n3          10101010
/two_stage1/n4          11110011
/two_stage1/n5          11110000
/two_stage1/n6          00001111
/two_stage1/n7          11110011
/two_stage1/clk         0
/two_stage1/reset       0
/two_stage1/in_node_    100
/two_stage1/out_node    111
    (2)                 1
    (1)                 1
    (0)                 1
/two_stage1/write_en    0
/two_stage1/read_en     0
/two_stage1/p0          01010101
/two_stage1/p1          10101010
                        1898798 ps
1895201 ps to 1898893 ps
```

Fig. 6.7(b) Modelsim simulation of two stage switching (8 x 8)



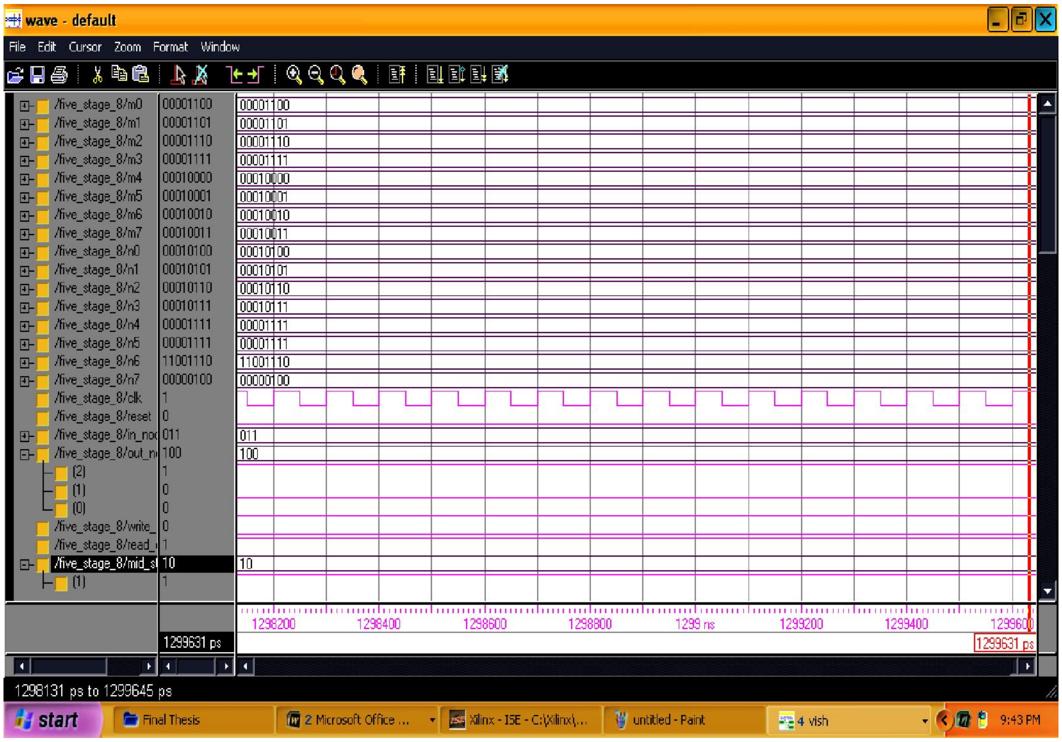
```
/three_stage_new/m0     00001111
/three_stage_new/m1     00111100
/three_stage_new/m2     01111000
/three_stage_new/m3     11110000
/three_stage_new/m4     11110011
/three_stage_new/m5     01010101
/three_stage_new/m6     10101010
/three_stage_new/m7     00110000
/three_stage_new/n0     00110011
/three_stage_new/n1     11110011
/three_stage_new/n2     01010101
/three_stage_new/n3     10101010
/three_stage_new/n4     11110011
/three_stage_new/n5     11110000
/three_stage_new/n6     00000000
/three_stage_new/n7     11110011
/three_stage_new/clk    1
/three_stage_new/rese   0
/three_stage_new/in_r   100
/three_stage_new/out_   111
/three_stage_new/writ   0
/three_stage_new/reac   0
/three_stage_new/mid_   11
    (1)                 1
    (0)                 1
/three_stage_new/p0     01010101
/three_stage_new/p1     10101010
                        1999518 ps
1998131 ps to 1999569 ps
```

Fig. 6.7(c) Modelsim simulation of three stage switching (8 x 8)



```
/four_stage_8/m0      11010111
/four_stage_8/m1      00010000
/four_stage_8/m2      00010001
/four_stage_8/m3      00010010
/four_stage_8/m4      00010011
/four_stage_8/m5      00010100
/four_stage_8/m6      00010101
/four_stage_8/m7      00010110
/four_stage_8/n0      00000000
/four_stage_8/n1      00000000
/four_stage_8/n2      00000000
/four_stage_8/n3      00000000
/four_stage_8/n4      00000000
/four_stage_8/n5      00000000
/four_stage_8/n6      00000000
/four_stage_8/n7      00000000
/four_stage_8/clk     1
/four_stage_8/reset   0
/four_stage_8/in_noc  010
/four_stage_8/out_n   110
    (2)               1
    (1)               1
    (0)               0
/four_stage_8/write_  0
/four_stage_8/read_   1
/four_stage_8/mid_s   11
    (1)               1
                      399641 ps
1398131 ps to 1399659 ps
```

Fig. 6.7(d) Modelsim simulation of four stage switching (8 x 8)



```
/five_stage_8/m0      00001100
/five_stage_8/m1      00001101
/five_stage_8/m2      00001110
/five_stage_8/m3      00001111
/five_stage_8/m4      00010000
/five_stage_8/m5      00010001
/five_stage_8/m6      00010010
/five_stage_8/m7      00010011
/five_stage_8/n0      00010100
/five_stage_8/n1      00010101
/five_stage_8/n2      00010110
/five_stage_8/n3      00010111
/five_stage_8/n4      00001111
/five_stage_8/n5      00001111
/five_stage_8/n6      11001110
/five_stage_8/n7      00000100
/five_stage_8/clk     1
/five_stage_8/reset   0
/five_stage_8/in_noc  011
/five_stage_8/out_n   100
    (2)               1
    (1)               0
    (0)               0
/five_stage_8/write_  0
/five_stage_8/read_   1
/five_stage_8/mid_s   10
    (1)               1
                      1299631 ps
1298131 ps to 1299645 ps
```

Fig. 6.7(e) Modelsim simulation of five stage switching (8 x 8)

6.3 SIMULATION OF 3D MULTISTAGE NETWORK

The RTL view of 3D (8 x 8 x 8) network is shown in the figure 6.8 (a) and its internal schematic is shown in figure 6.8 (b). Table 6.2 explains the pin details of the 3D (8 x 8 x 8) network structure. 3D NoC can be integrated with any staged network.

Table 6.2 Design pins and their functional description for 3D (8 x 8 x 8) NOC

Pins	Functional Description
reset	Used to reset the contents of memory to zero and synchronization with the components by using clk. It is of (1 bit) of std_logic type
clk	Provide the clock pulse to work on rising edge or the falling edge of clock pulse with 50 % duty cycle (1 bit of std_logic)
node_address [5:0]	Address of the source and destination nodes (6 bits of std_logic_vector)
row_address [2:0]	represents address of the nodes in 'X' direction (3 bits of std_logic_vector)
column_address [2:0]	represents address of the nodes in 'Y' direction (3 bits of std_logic_vector)
Third_address [2:0]	represents address of the nodes in 'Z' direction (3 bits of std_logic_vector)
read_en	control signal used to read the data at output port

	(1 bit of std_logic)
write_en	control signal to write data at the output port (1 bit of std_logic)
network_address [1:0]	Selection logic for the network (2 bits of std_logic_vector)
data_in [7:0]	represents input data in the network (8 bits of std_logic_vector)
data_out [7:0]	represents 8 bit output data of the destination node (8 bits of std_logic_vector)

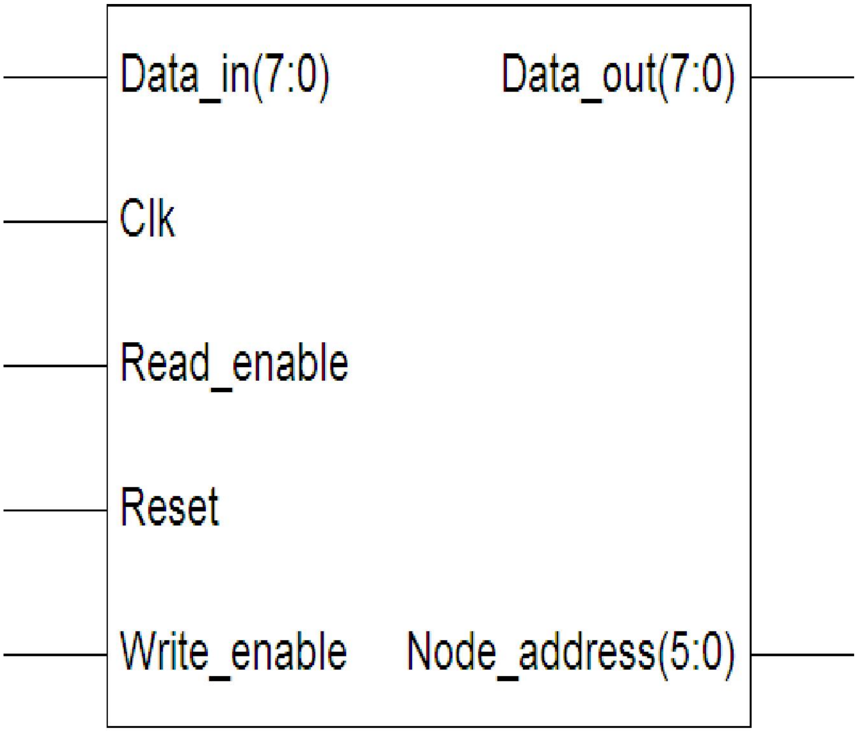
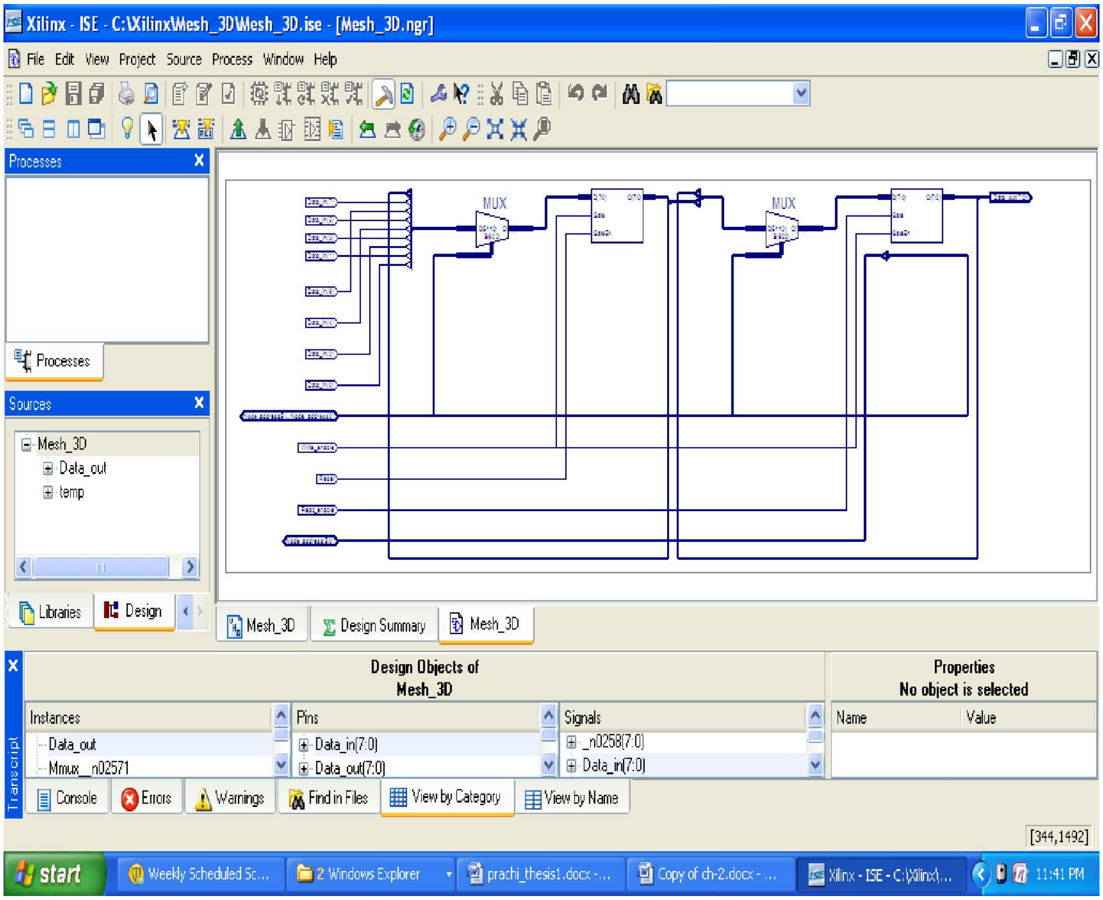


Fig. 6.8 (a) RTL view of 3D (8 x 8 x 8)



Design Objects of Mesh_3D

Instances	Pins	Signals
Data_out	Data_in(7:0)	_n0258(7:0)
Mmux__n02571	Data_out(7:0)	Data_in(7:0)

Properties
No object is selected

Name	Value

Fig. 6.8 (b) internal schematic of 3D (8 x 8 x 8)

The flow diagram of simulation process and intercommunication data transfer in 3D NoC is shown in figure 6.9. Figure 6.10 shows the Modelsim simulated result for the 3D (8 x 8 x 8) NoC for intercommunication, which shows 8 bit data among inlets and outlets. The functional simulation depends on the following steps and output is verified after the completion of these steps.

Simulation Process Sequence

Step 1: reset = *'1', clk* is used for synchronization and then run.

Step 2: reset = *'0',* same *clk* is used for synchronization and provide rising edge

Step 3: Select the address of destination node *Node_address [5:0]* of 6 bits for 8 x 8 x 8 size.

Step 4: Force the value of *row_address* and *column_address* of destination node. For 8 x 8 x 8 NOC *row_address[2:0]* and *column_address[2:0] and third_address[2:0]* are of 3 bits. *row_address[2:0]* and *column_address[2:0]* are for XY axis *column_address[2:0]* and *Third_address[2:0]* are for YZ axis *and Third_address[2:0]and row_address[2:0]* are for ZX axis.

Step 5: Give the eight bits value of data_in. Force write_en =1 and read_en =0 and then run.

Step 6: write_en = ‘0’ and read_en = ‘1’ and run. Desired output on destination is achieved.

When write_en = ‘1’ and read_en = ‘0’, the data is written in temp variable from the source node, when write_en = ‘0’ and read_en = ‘1’, the data is read from the temp variable to destination node. Clk is applied to give the positive edge clock pulse and reset is kept at ‘1’ for the initial state. In the simulated result shown in the figure 6.10 the data_in[7 :0] and data_out[7 :0] are of 8 bits, which are “10101010”, node selection is done for node 16, which depends on the row_address = “001”, column_address = “001” and third_address = “110”. The network_address is default = “00” and clock signal is following the positive clock pulse. The duty cycle of the rising edge clock is kept of 50 %, and its synchronized with reset.

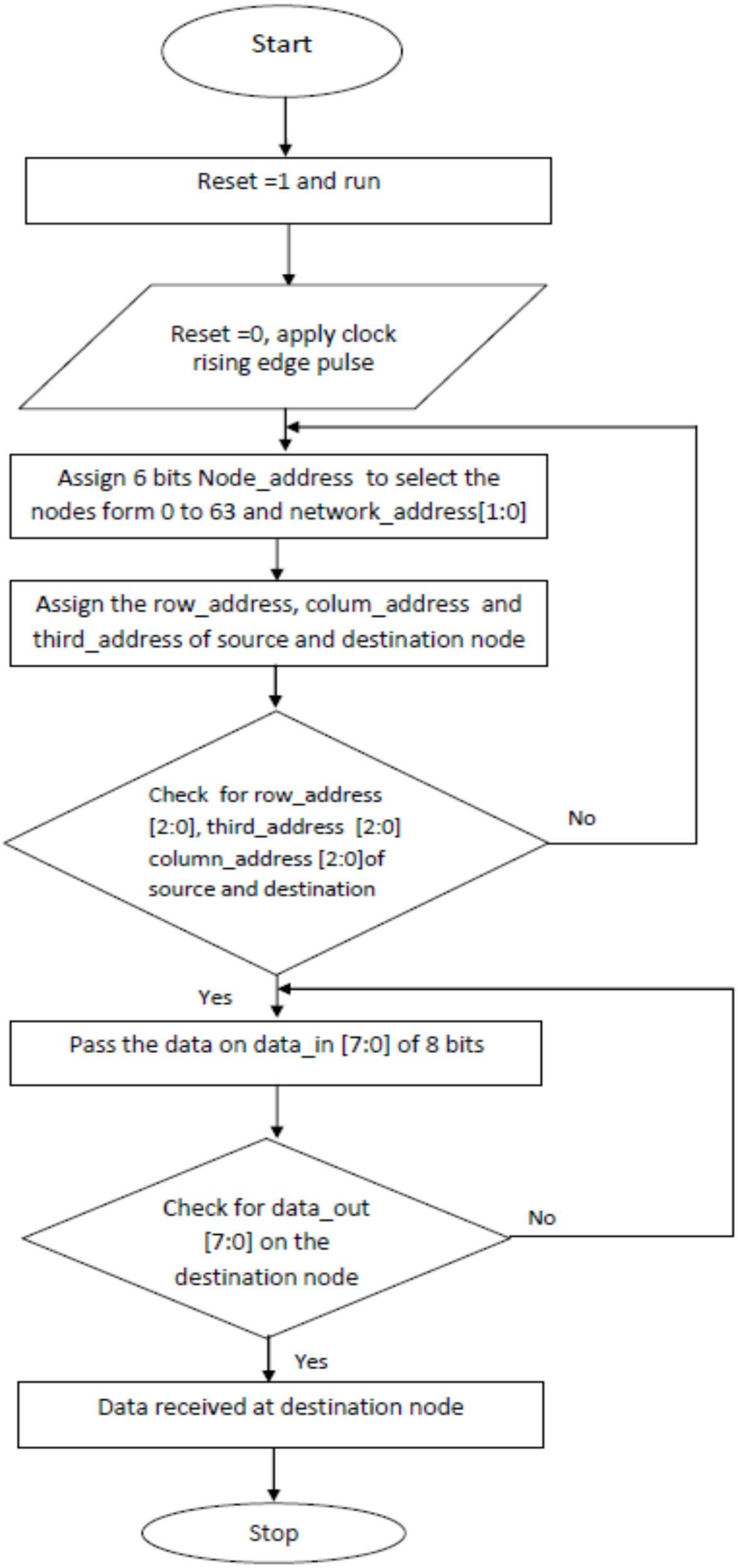
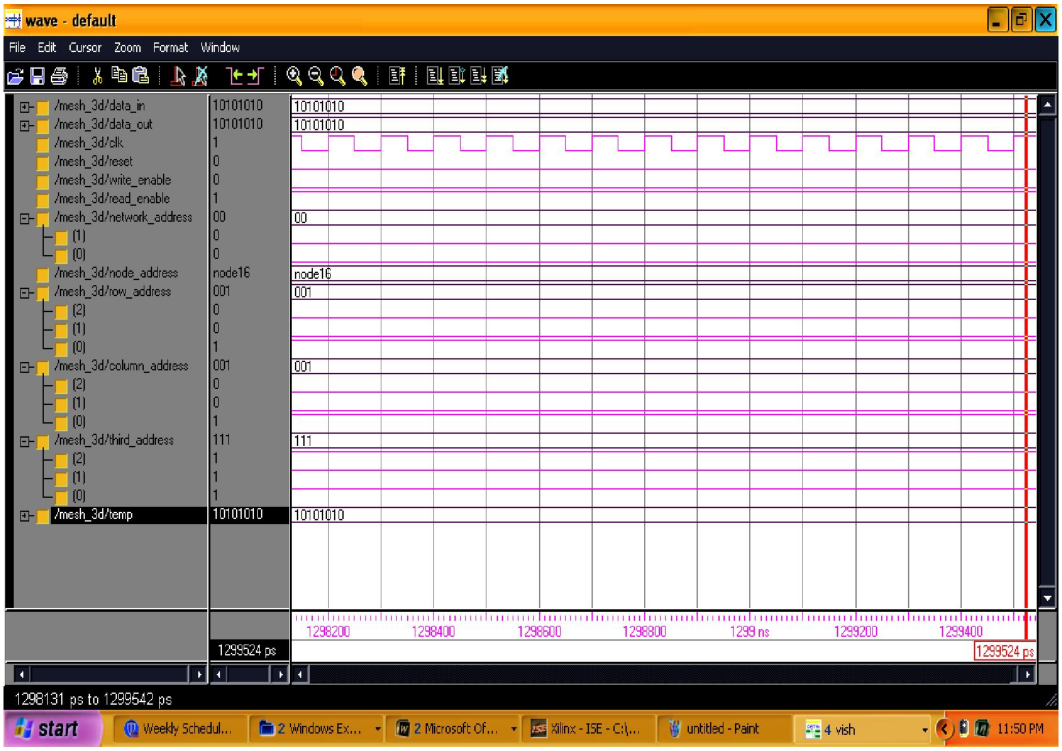

Start
Reset =1 and run
Reset =0, apply clock rising edge pulse
Assign 6 bits Node_address to select the nodes form 0 to 63 and network_address[1:0]
Assign the row_address, colum_address and third_address of source and destination node
Check for row_address [2:0], third_address [2:0] column_address [2:0]of source and destination
No
Yes
Pass the data on data_in [7:0] of 8 bits
Check for data_out [7:0] on the destination node
No
Yes
Data received at destination node
Stop

Fig. 6.9 Flow chart of data transfer in 3D intercommunication



/mesh_3d/data_in	10101010	10101010
/mesh_3d/data_out	10101010	10101010
/mesh_3d/clk	1	
/mesh_3d/reset	0	
/mesh_3d/write_enable	0	
/mesh_3d/read_enable	1	
/mesh_3d/network_address	00	00
(1)	0	
(0)	0	
/mesh_3d/node_address	node16	node16
/mesh_3d/row_address	001	001
(2)	0	
(1)	0	
(0)	1	
/mesh_3d/column_address	001	001
(2)	0	
(1)	0	
(0)	1	
/mesh_3d/third_address	111	111
(2)	1	
(1)	1	
(0)	1	
/mesh_3d/temp	10101010	10101010

1299524 ps

1298131 ps to 1299542 ps

Fig. 6.10 Modelsim simulation of 3D (8 x 8 x 8) network

6.4 SYNTHESIS OF STAGED NETWORKS

The synthesis of single stage, two stage, three stage, four stage and five stage multistage networks include the hardware sub parameters such as no. of registers, no. of flip flops, multiplexers, priority encoder, decoder, shifter, gates, memory extraction, FSM encoding algorithm DSP block, minimum size, automatic register balancing as source options. The source options of synthesized multistage networks are listed in table 6.3. Target FPGA device options include Global maximum fan out, add Generic Clock Buffer (BUFG), register duplication, slice packing, add I/O buffers, use clock enable, use synchronous set/reset, optimize instantiated primitives, equivalent register removal, and optimize instantiated primitives information in terms of used or not.

Table 6.3 Synthesis results as source options of multistage network

Parameters	**Single Stage**	**Two stage**	**Three stage**	**Four stage**	**Five stage**
Automatic FSM Extraction	Yes	Yes	Yes	Yes	Yes
FSM Encoding Algorithm	One-hot	One-hot	One-hot	One-hot	One-hot
FSM Style	LUT	LUT	LUT	LUT	LUT
RAM Extraction	Yes	Yes	Yes	Yes	Yes
RAM Style	Auto	Auto	Auto	Auto	Auto
ROM Extraction	Yes	Yes	Yes	Yes	Yes
Mux style	Auto	No	Auto	Auto	Auto
Decoder Extraction	Yes	No	Yes	Yes	Yes
Priority Extraction	Yes	No	Yes	Yes	Yes
Shift Register Extraction	Yes	Yes	Yes	Yes	Yes
XoR collapsing	Yes	Yes	Yes	Yes	Yes
Resource Sharing	Yes	Yes	Yes	Yes	Yes
Shift Register Minimum Size	2	2	2	2	2
Use DSP Block	Yes	Yes	Yes	Auto	Auto
Automatic Register Balancing	No	No	No	No	No

Table 6.4 Synthesis results as target options of multistage network

Parameters	**Single Stage**	**Two stage**	**Three stage**	**Four stage**	**Five stage**
Add IO Buffers	Yes	Yes	Yes	Yes	Yes
Global Maximum Fan out	100000	100000	100000	100000	100000
Add Generic Clock Buffer(BUFG)	32	16	8	32	32
Register Duplication	Yes	Yes	Yes	Yes	Yes
Slice Packing	Yes	Yes	Yes	Yes	Yes
Optimized Instantiated Primitives	No	No	No	No	No
Use Clock Enable	Auto	Auto	Yes	Auto	Auto
Use Synchronous Set	Auto	Auto	Yes	Auto	Auto
Use Synchronous Reset	Auto	Auto	Yes	Auto	Auto
Pack IO Registers into IOBs	Auto	Auto	Auto	Auto	Auto
Equivalent register Removal	Yes	Yes	Yes	Yes	Yes

The target options are listed in table 6.4 of multistage networks synthesis results as general options of multistage network are listed in table 6.5, which include optimization goal, optimization effort, RTL output, keep hierarchy, netlist hierarchy, global optimization, slice utilization ratio, BRAM utilization ratio, DSP48 utilization ratio, auto BRAM packing and slice utilization ratio delta . Advanced synthesis results of multistage networks are listed in table 6.6, which include the information of 8 bit registers, flip-flops, I/Os, multiplexers, BELs, Look Up Tables (LUT), flip flops/laches, FDCs, BUFGP, clock buffers and input and output buffers (IOB).

Table 6.5 Synthesis results as general options of multistage network

Parameters	**Single Stage**	**Two stage**	**Three stage**	**Four stage**	**Five stage**
Optimization Goal	Speed	Speed	Speed	Speed	Speed
Optimization Effort	1	1	1	1	1
RTL Output	Yes	Yes	Yes	Yes	Yes
Keep Hierarchy	No	No	No	No	No
Netlist Hierarchy	As Optimized	As Optimized	As Optimized	As Optimized	As Optimized
Global Optimization	All Clock Nets	All Clock Nets	All Clock Nets	All Clock Nets	All Clock Nets
Slice Utilization Ratio	100	100	100	100	100
BRAM Utilization Ratio	100	100	100	100	100
DSP48 Utilization Ratio	100	100	100	100	100
Auto BRAM Packing	No	No	No	No	No
Slice Utilization Ratio Delta	5	5	5	5	5

Table 6.6 Synthesis results HDL compilation report

Parameters	**Single Stage**	**Two stage**	**Three stage**	**Four stage**	**Five stage**
8-bit register	64	4	160	160	192
Flip flops	64	4	160	160	192
8-bit 8-to-1 multiplexer	8	19	20	20	24
IOs	136	133	138	138	138
BELS	105	33	750	261	386
LUT3	9	3	8	16	11
LUT5	64	8	242	180	120
LUT6	32	48	211	32	224
Flip Flops/Latches	64	200	160	160	192
FDC	64	4	64	64	64
Clock Buffers	1	1	1	1	1
BUFGP	1	1	1	1	1
IO Buffers	135	132	121	137	137
IBUF	71	68	57	73	73
OBUF	64	64	64	64	64

The source options listed in table 6.3 are same for single stage, two stage, three stage, four stage and five stage, only mux style in two stage and DSP blocks in four and five stage are different. The criteria to choose mux and DSP block depends on network configuration and routing scheme, which is taken by the software based on the program is written.

6.5 DEVICE UTILIZATION AND TIMING SYNTHESIS

Device utilization report gives the percentage utilization [13] of device hardware for the chip implementation. Device hardware includes No of slices, No of flip flops, No of input LUTs, No. of bounded IOBs, and No of gated clocks (GCLKs) used in the implementation of design. Timing [13] details provides the information of delay, minimum period, maximum frequency, minimum input arrival time before clock and maximum output required time after clock. Total memory utilization required to complete the design is also listed for individual stage. The target device is: xc5vlx20t-2-ff323 synthesized with Virtex-5 FPGA. The device utilization and timing parameters of single stage, two stage, three stage, four stage and five stage for network clusters (N = 2, 4, 8, 16) are shown in table 6.7 and 6.8, table 6.9 and 6.10, table 6.11 and 6.12, table 6.13 and 6.14 and table 6.15 and 6.16 respectively.

Table 6.7 Device utilization in single stage switching

Device part	**Device Utilization**			
	N =2	**N = 4**	**N =8**	**N =16**
Number of Slices	9 out of 12480	32 out of 12480 0%	64 out of 12480 0%	128 out of 12480 1%
Number of Slice Flip Flops	2 out of 12480	40 out of 12480 0%	105 out of 12480 0%	493 out of 12480 3%
Number of 4 input LUTs	0 out of 9	32 out of 40 80%	64 out of 105 60%	128 out of 493 25%
Number of bonded IOBs	36 out of 172 20 %	70 out 0f 1 72 40 %	136 out 0f 1 72 79 %	156 out of 172
Number of GCLKs	1 out of 32 3%	1 out of 32 3%	1 out of 32 3%	1 out of 32 3%

Table 6.8 Single stage timing parameters

Timing parameter	Utilization			
	N =2	N=4	N=8	N=16
Minimum period	0.898ns	0.898ns	0.898ns	0.898ns
Maximum Frequency	1113.772MHz	1113.772MHz	1113.772MHz	1113.772MHz
Minimum input arrival time before clock	1.732ns	2.474ns	2.721ns	3.896ns
Maximum output required time after clock	2.826ns	2.830ns	2.830ns	2.830ns
Total Memory usage	102764 kB	113212 kB	128332 kB	139708 kB

Table 6.9 Device utilization in two stage switching

Device part	Device Utilization			
	N =2	N = 4	N =8	N =16
Number of Slices	32 out of 12480 0%	64 out of 12480 0%	192 out of 12480 1%	640 out of 12480 5%
Number of Slice Flip Flops	12 out of 12480 0%	84 out of 12480 0%	297 out of 12480 2%	1588 out of 12480 12%
Number of 4 input LUTs	10 out of 34 29 %	64 out of 84 76%	192 out of 297 60%	640 out of 1588 40%
Number of bonded IOBs	38 out of 172 22%	72 out of 172 41%	138 out 0f 172 80 %	60 out o 172
Number of GCLKs	1 out of 32 3%	1 out of 32 3%	1 out of 32 3%	1 out of 32 3%

Table 6.10 Two stage timing parameters

Timing parameter	Utilization			
	N =2	N=4	N=8	N=16
Minimum period	0.998ns	1.086ns	1.305ns	2.393ns

Maximum Frequency	1013.22MHz	920.938MHz	766.313MHz	417.807MHz
Minimum input arrival time before clock	1.732ns	2.544ns	3.328ns	4.257ns
Maximum output required time after clock	2.826ns	2.830ns	2.835ns	2.854ns
Total Memory usage	112774 kB	124620 kB	145052 kB	166268 kB

Table 6.11 Device utilization in three stage switching

Device part	**Device Utilization**			
	N =2	**N = 4**	**N =8**	**N =16**
Number of Slices	32 out of 12480 0%	64 out of 12480 0%	128 out of 12480 1%	256 out of 12480 2%
Number of Slice luts	11 out of 12480 0%	77 out of 12480 0%	189 out of 12480 1%	585 out of 12480 4%
Number of 4 fully used LUTs flip flop	9 out of 34 26 %	64 out 0f 77 83%	128 out of 189 67%	256 out of 585 43%
Number of bonded IOBs	38 out of 172 22%	72 out of 172 41%	138 out of 172 80 %	168 out of 172
Number of GCLKs	1 out of 32 3%	1 out of 32 3%	1 out of 32 3%	1 out of 32 3%

Table 6.12 Three stage timing parameters

Timing parameter	**Utilization**			
	N =2	**N=4**	**N=8**	**N=16**
Minimum period	0.686ns	1.091ns	1.309ns	1.867ns
Maximum Frequency	1458.045MHz	916.632MHz	763.825MHz	535.733MHz
Minimum input arrival time before clock	2.057ns	2.475ns	2.902ns	4.045ns
Maximum	2.826ns	2.830ns	2.830ns	2.830ns

output required time after clock				
Total Memory usage	123212 kB	144072 kB	166004 kB	229516 kB

Table 6.13 Device utilization in four stage switching

Device part	**Device Utilization**			
	N =2	**N = 4**	**N =8**	**N =16**
Number of Slices	32 out of 12480 0%	80 out of 12480 0%	160 out of 12480 1%	320 out of 12480 2%
Number of Slice luts	27 out of 12480 0%	93 out of 12480 0%	229 out of 12480 1%	649 out of 12480 5%
Number of 4 fully used LUTs flip flop	25 out of 34 73 %	80 out of 93 86%	160 out of 229 67%	320out of 649 49%
Number of bonded IOBs	38 out of 172 22%	72 out of 172 41%	138 out of 172 80 %	168 out of 172
Number of GCLKs	1 out of 32 3%	1 out of 32 3%	1 out of 32 3%	1 out of 32 3%

Table 6.14 Four stage timing parameters

Timing parameter	**Utilization**			
	N =2	**N=4**	**N=8**	**N=16**
Minimum period	1.002ns	1.091ns	1.305ns	1.867ns
Maximum Frequency	944.330MHz	816.632MHz	766.313MHz	535.733MHz
Minimum input arrival time before clock	2.165ns	2.486ns	2.907ns	4.090ns
Maximum output required time after clock	2.830ns	2.830ns	2.830ns	2.830ns
Total Memory usage	142472 kB	155096 kB	186872 kB	263208 kB

Table 6.15 Device utilization in five stage switching

Device part	Device Utilization			
	N =2	N = 4	N = 8	N = 16
Number of Slices	32 out of 12480 0%	96 out of 12480 0%	192 out of 12480 1%	376 out of 12480 3%
Number of Slice LUTs	19 out of 12480 0%	109 out of 12480 0%	386 out of 12480 3%	726 out of 12480 5%
Number of 4 fully used LUTs flip flop	17 out of 34 50 %	96 out of 109 88%	192 out of 386 49%	371 out of 731 50%
Number of bonded IOBs	38 out of 172 22%	72 out of 172 41%	138 out of 172 80 %	168 out of 172
Number of GCLKs	1 out of 32 3%	1 out of 32 3%	1 out of 32 3%	1 out of 32 3%

Table 6.16 Five stage timing parameters

Timing parameter	Utilization			
	N =2	N=4	N=8	N=16
Minimum period	1.012ns	1.191ns	1.786ns	1.848ns
Maximum Frequency	944.330MHz	816.632MHz	559.942MHz	541.155MHz
Minimum input arrival time before clock	2.165ns	2.486ns	3.391ns	4.285ns
Maximum output required time after clock	2.826ns	2.830ns	2.830ns	2.830ns
Total Memory usage	162636 kB	175096 kB	200968 kB	299880 kB

Figure 6.11 shows the memory utilization in first, two, three, four and five stage network. From the graph, it is clear that the memory utilization is also increasing, as the numbers of stages are increasing from single stage to two stage, three stage, four stage and five stage. As the network cluster is increasing from N

= 2 to 4, 8, and 16, memory utilization is increasing in each stage, because the hardware parameters of chip design such as, no of slices , no of flip flops, LUTs, and GCLKs are increasing.

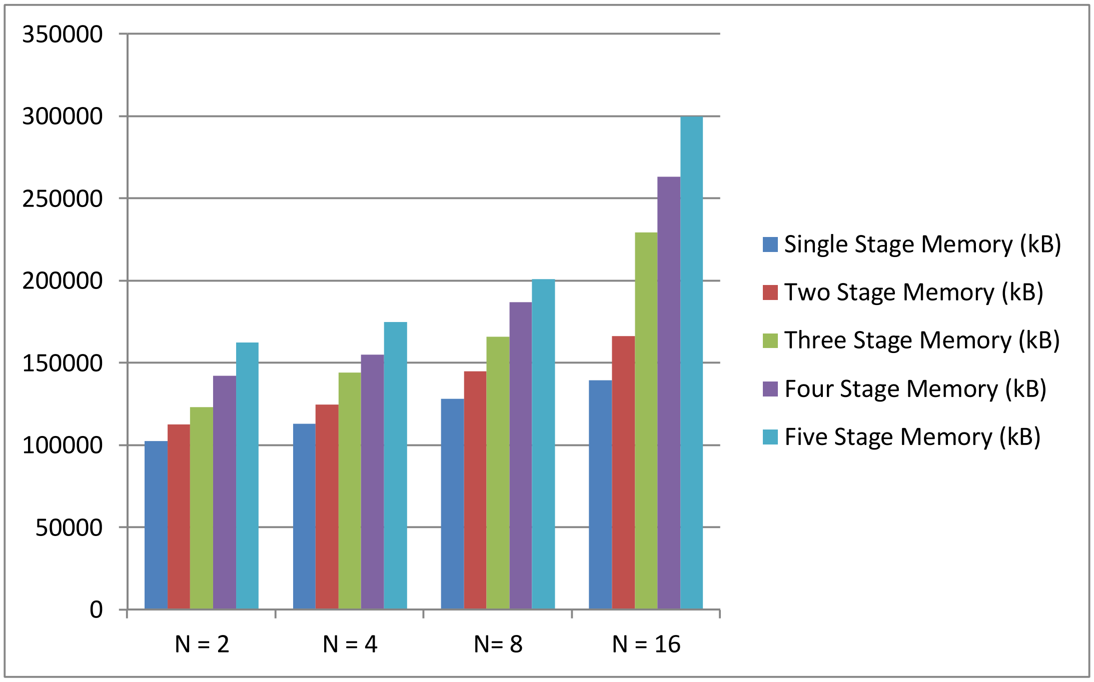


Fig.6.11 Memory utilization in first, two, three, four and five stage network.

For N = 2, there is 9.74 % increment in memory utilization in two stage switching with comparison to single stage switching, 9.25 % in three stage switching with comparison to two stage switching, 15.63 % in four stage switching with comparison to three stage switching and 14.15 % in five stage switching with comparison to four stage switching. Similarly, for N = 4, there is 10.07 % increment in memory utilization in two stage switching with comparison to single stage switching, 15.60 % in three stage switching with comparison to two stage switching, 7.70 % in four stage switching with comparison to three stage switching and 12.90 % in five stage switching with comparison to four stage switching. For N = 8, there is 13.02 % increment in memory utilization in two stage switching with comparison to single stage switching, 14.44 % in three stage switching with comparison to two stage switching, 12.57 % in four stage switching with comparison to three stage switching and 7.54 % in five stage switching with comparison to four stage switching. For N = 16, there is 19.01 % increment in memory utilization in two stage switching with comparison to single stage switching, 38.03 % in three stage switching with comparison to two stage switching, 14.67 % in four stage switching with comparison to three stage switching and 13.93 % in five stage switching with comparison to four stage switching.

The modeling of the multistage switching systems helps to estimate the different parameters support the network designing. These parameters are switching capacity, number of switching elements and blocking probability. The calculated values of these parameters for multistage networks are listed in table 6.17. Switching capacity depends on the cluster size and network configuration. Blocking probability is calculated using Poisson process. The governing equation of a Poisson Process is [91]

$$P_k(t) = \frac{(\lambda t)^k e^{-\lambda t}}{k!} \qquad \text{Equ. 6.1}$$

Where λ = Calls per second, the value of t is taken from the deice utilization report as the value of minimum time.

Table 6.17 Calculations for switching elements, switching capacity and blocking probability of multistage network

Parameters	Size	Switching Elements	Switching capacity	Blocking Probability
Single Stage Switching	N = 2	1	1	0.06433
	N = 4	4	2	0.09777
	N = 8	8	4	0.19681
	N = 16	16	8	0.2491573
Two stage Switching	N = 2	4	1	0.04905
	N = 4	16	4	0.06928
	N = 8	64	16	0.1644
	N = 16	256	64	0.193
Three stage Switching	N = 2	5	1	0.03548
	N = 4	24	8	0.0585
	N = 8	128	64	0.1325
	N = 16	768	512	0.1691
Four stage Switching	N = 2	6	1	0.02387
	N = 4	32	16	0.0325
	N = 8	192	256	0.09433
	N = 16	1280	4096	0.133288
Five Stage Switching	N = 2	7	1	0.0147
	N = 4	40	32	0.0191

	N = 8	258	1024	0.02731
	N = 16	1792	32768	0.0425

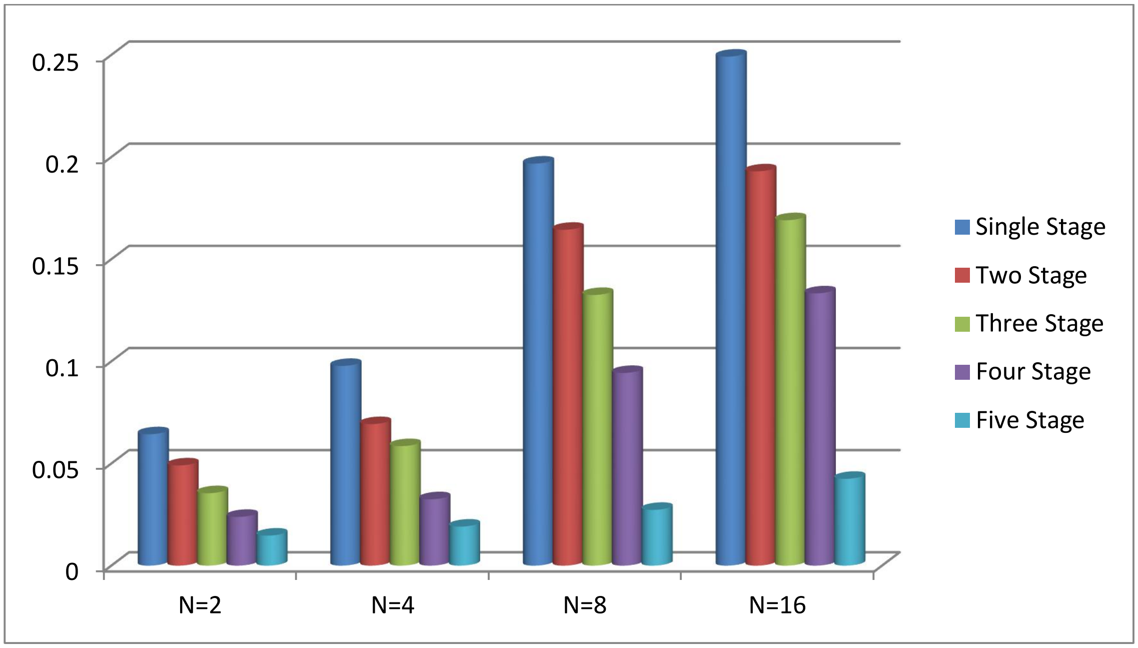


Fig. 6.12 Blocking probability in single, two, three, four and five stage network

The graph shown in figure 6.12 is estimated for blocking probability. It is seen that the blocking probability of multistage network increases with network size and decreases with the increment in number of stages. For N = 2, blocking probability is reduced, in two stage 23.75 % in comparison with single stage, in three stage 27.66 % in comparison with two stage, in four stage 32.72 % in comparison with three stage, in five stage 38.41 % in comparison with four stage. For N = 4 , blocking probability is reduced, in two stage 29.13 % in comparison with single stages, in three stage 15.56 % in comparison with two stage, in four stage 44.44 % in comparison with three stage, in five stage 14.20 % in comparison with four stage. For N = 8, blocking probability is reduced, in two stage 16.46 % in comparison with single stage, in three stage 19.40 % in comparison with two stage, in four stage 28.80 % in comparison with three stage, in five stage 71.04 % in comparison with four stage. For N = 16, blocking probability is reduced, in two stage 22.53 % in comparison with single stage, in three stage 12.38 % in comparison with two stage, in four stage 21.17 % in comparison with three stage, in five stage 68.11 % in comparison with four stage. The main cause of reducing the blocking probability is dependent on the minimum period to route the call. If the time increment is greater than the period of arriving of clock pulse, there are the chances of increasing the blocking probability of multistage networks.

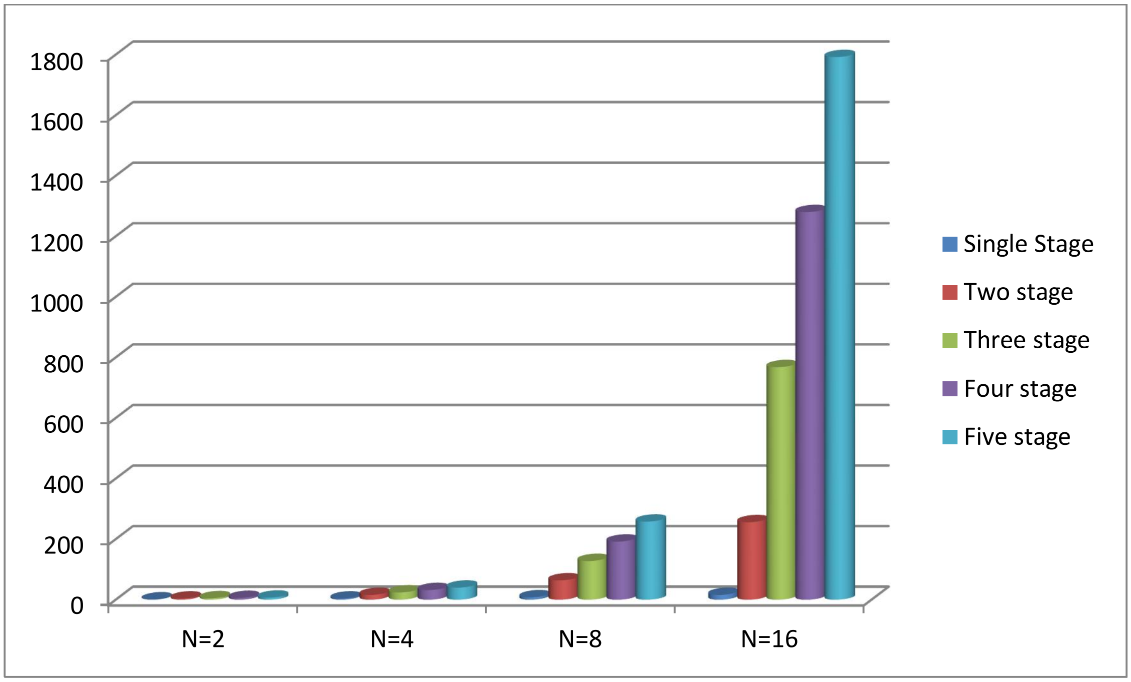


Fig. 6.13 Number of switching elements in single, two, three, four and five stage (N= 2, 4, 8, 16)

Figure 6.13 shows the number of switching elements in single, two, three, four and five stage. From the graph shown in figure 6.13, it is clear that the switching elements of multistage network increases with the increment in the stages. In the same stage, the numbers of switching elements also increases with the increase in the cluster size. For N = 2, there is an increment of switching elements, 75 % in two stage with comparison to single stage, 25 % in three stage with comparison to two stage, 20 % in four stage with comparison to three stage, 16.66 % in five stage with comparison to four stage. Similarly, For N = 4, there is an increment of switching elements, 300 % in two stage with comparison to single stage, 50.00 % in three stage with comparison to two stage, 33.33 % in four stage with comparison to three stage, 25.00 % in five stage with comparison to four stage. For N = 8, there is an increment of switching elements, 700 % in two stage with comparison to single stage, 100 % in three stage with comparison to two stage, 50.00 % in four stage with comparison to three stage, 34.37 % in five stage with comparison to four stage. For N = 16, there is an increment of switching elements, 1500 % in two stage with comparison to single stage, 200 % in three stage with comparison to two stage, 66.66 % in four stage with comparison to three stage, 40.00 % in five stage with comparison to four stage.

Fig. 6.14 shows the switching capacity of one, two, three, four and five stage networks. From the table 6.17 and graph shown in figure 6.14, it can be seen that the switching capacity also increases with the numbers of stages. For N = 2, the switching capacity is same for single stage, two stage, three stage, four stage and five stage networks. For, N = 4, 8 and 16, the switching capacity is increased, in all stages. For N = 4, the switching capacity for single stage, two stage, three stage, four stage and five stage are found as 2, 4, 8, 16 and 32 calls respectively. For N = 8, the switching capacity for single stage, two stage, three stage, four stage and five stage are 4, 16, 64, 256 and 1024 calls respectively. For N = 16, the switching capacity for single stage, two stage, three stage, four stage and five stage are 8, 64, 512, 4096 and 32768 calls respectively.

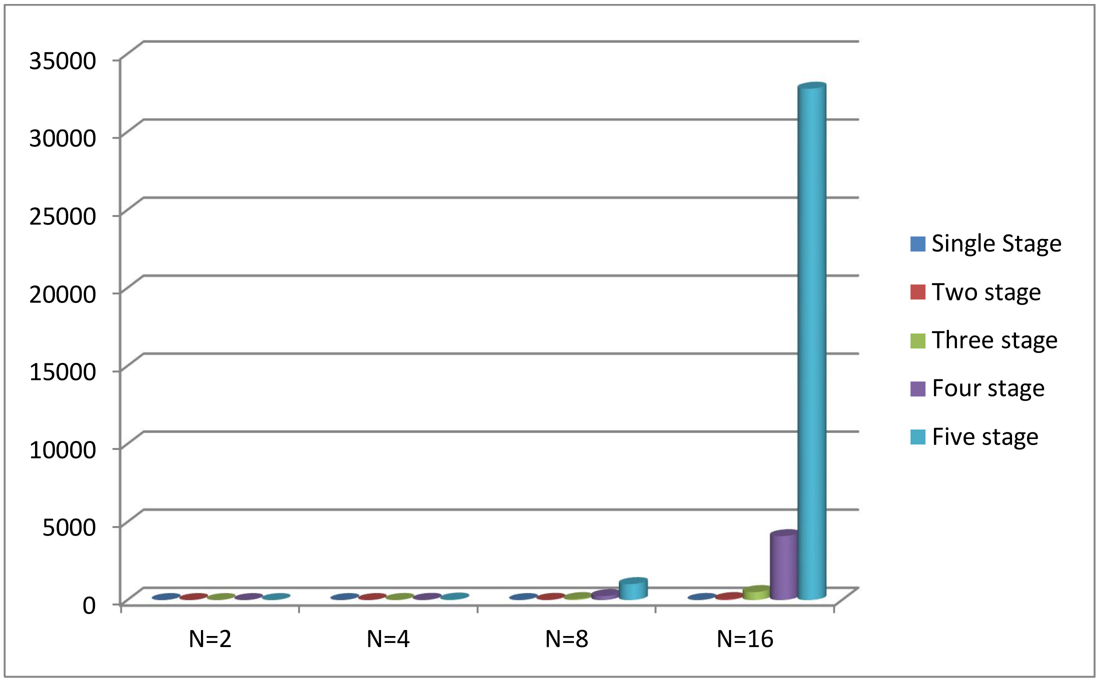


Fig. 6.14 Switching Capacity in single, two, three, four and five stage network (N= 2, 4, 8 and 16)

From the discussion, switching capacity of multistage network can be estimated by mathematical equation,

$$\text{Switching Capacity} = \left[\frac{N}{2}\right]^{S} \qquad \text{Equ. 6.2}$$

Where N = Number of clusters, and S= Number of stages.

CHAPTER SUMMARY

In the chapter RTL views and internal schematics of single stage, two stage, three stage, four stage and five stage multistage networks are discussed. The flow chart of each staged network is discussed based on their functional simulation. The functionality of 3D multistage network is also verified with some test cases. The view synthesis report is based on the hardware utilization in terms of No. of slices, No. of flip flops, No. of input LUTs, No. of bounded IOBs, and No. of gated clocks (GCLKs) used in the implementation of design. Timing details is also carried out for the staged network which provides the information of delay, minimum period, maximum frequency, minimum input arrival time before clock and maximum time required after clock. Total memory utilization, required to complete the design is also listed for individual stage. The target device is: xc5vlx20t-2-ff323 synthesized with Virtex-5 FPGA. The number of switching elements, switching capacity and blocking probability of each stage is also calculated for cluster size (N = 2, 4, 8 and 16). It is concluded, that with the increment of stages and clusters, the hardware utilization is also increasing and blocking probability of the network is decreasing. The equation governing for the switching capacity is estimated based on the calculated values of switching capacity and cluster size.

CHAPTER-7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The hardware chip implementation of the multistage telecommunication network is carried out for single stage, two stage, three stage, four stage and five stage in Xilinx ISE 14.2 successfully. The cluster configurations for the respective stages chosen were, (2 x 2), (4 x 4), (8 x 8) and (16 x 16). The functional simulation of individual stages is done in Modelsim 10.1 b for the different test cases. TACIT network security algorithm is integrated with the chip and functionality tested for the same test cases. The data transfer scheme in the network is analyzed with the help of Virtex – 5 FPGA XC5VLX110T, a Digilent manufactured FPGA, and is validated for the voice signal of 3 KHz. A comparison of hardware parameters is carried out for the all stages. The synthesis report is generated and contains the information for hardware utilization in terms of No of slices, No of flip flops, No of input LUTs, No. of bounded IOBs and No of gated clocks (GCLKs) used in the implementation of design. Timing analysis is also carried out for the staged network which provides the information of delay, minimum period, maximum frequency, minimum input arrival time before clock and maximum output required time after clock. Total memory utilization required by individual stage is also compared for different stages. The network parameters, blocking probability and switching capacity is optimized with the help of the synthesized results. It is seen that the blocking probability of multistage network increases with network size and decreases as number of stages are increasing.

The switching capacity of the network is found to increase with the increment of stages. With the increase in number of stages, the hardware and memory utilization of the device increases with network cluster configuration (N = 2, 4, 8, 16). Memory utilization for N = 16 is found as 19.01 % greater, for two stage switching in comparison to single stage switching, 38.03 % in three stage switching in comparison to two stage switching, 14.67 % in four stage switching in comparison to three stage switching and 13.93 % in five stage switching in comparison to four stage switching.

The network realization of existing telecommunication network has proven advantageous for enhanced switching capacity and reduced blocking probability. As the network cluster increases in size, the hardware utilization also increases. In this design, N = 8, is found as the optimal solution to co-control the programmable telecommunication network. It shows that the switching of the telephone exchange is distributed in the cluster size of 8 users and controlling of exchange is done by cascading the same size of FPGA chips. For N = 8, the switching capacity of the network is increased from 4, 16, 64, 256, 1024 calls, for single stage, two stage, three stage, four stage and five stage networks respectively. The switching capacity of the network is 1024 calls to support fully available network.

Blocking probability of multistage network increases with network size, and decreases with the increment in number of stages. For N = 8, blocking probability is reduced. In two stage network the reduction in blocking probability is 16.46 % in comparison with single stage. In three stage network reduction is 19.40 % in comparison with two stage. In four stage network it is 28.80 % in comparison with three stage and in five stage 71.04 % in comparison with four stage. For N = 16, blocking probability is reduced, in two stage 22.53 % in comparison with single stage. In three stage the reduction is 12.38 % in comparison with two stage. In four stage network it is 21.17 % in comparison with three stage and in five stage 68.11 % in comparison with four stage. From the result it is seen that there is a much higher reduction of blocking probability in four and five stage in comparison to two and three stage respectively. Blocking probability directly relates to the traffic congestion. The grade of service of the network is found to improve with the reduction in the blocking probability.

The data transfer scheme of the multistage networks is tested with their possible routes and addresses. The integration of TACIT network security with two, three, four and five stage networks has given 100 % successful results of data transfer. The voice signal of 3 KHz is transferred from inlets via FPGA and same signal is received at outlet and measured on DSO. The validation of voice signal is carried out in all stages of networks. The research work of network chip implementation of four and five stage telecommunication network is a significant effort towards total digitization and programmable of switching systems and to have co- control using FPGA chips.

7.2 FUTURE SCOPE

Programmable switching structure is an optimal way to implement the telecommunication system for extension lines using Network on chip (NoC) concept in FPGA chips, and make provision to co-control or cascade to the other FPGA for multiplexing the extension lines. The research work is not limited to cluster configuration, it can be made upto N = 32, 64, 128 or more based on the synthesis tools availability and designer's need. The effect of higher number of stages can also be investigated in order to increase the switching capacity. TACIT network security has the advantage that the block size and key size can be of 'N' bits and the integration of security with programmable structure will have best results, especially for secured data transmission in telecommunication network. In future, an additional work should be carried out with added features of network security with other algorithms for encryption and decryption of data transfer among inlets and outlets. The same concept of NoC and programmable switching structures can be integrated with other wireless technologies such as Wimax, WiFi, Bluetooth, and wireless sensor networks.

REFERENCES

[1] A. Mello, L. Tedesco, N. Calazans, and F. Moraes, "Evaluation of current QoS mechanisms in networks on chip," in Proceedings of the International Symposium on System-on-Chip, (SOC' 06), pp. 1–4, Tampere, Finland, November 2006.

[2] Aye Sandar Win "Design and Construction of Microcontroller Based Telephone Exchange System" World Academy of Science, Engineering and Technology, Vol. 46, pp (60-67), 2008.

[3] Aurel A. Lazar, "Programming Telecommunication Networks" IEEE Network, pp (8-19), September 1997.

[4] Andreas Hansson, Kees Goossens and Andrei Radulescu "A Unified Approach to Mapping and Routing on a Network-on-Chip for Both Best-Effort and Guaranteed Service Traffic" Hindawi Publishing Corporation VLSI Design Volume 2007, pp (1-16).

[5] A. M. Rahmani, M. Daneshtalab, A. Afzai-Kusha, S. Safari, and M. Pedram, "Forecasting-based dynamic virtual channels allocation for power optimization of network-on-chips," in Proceedings of the 22nd International Conference on VLSI Design-Held Jointly with 7th International Conference on Embedded Systems, pp. 151–156, New Delhi, India, January 2009.

[6] Al Faruque MA, Ebi T, Henkel J "Run-time adaptive on-chip communication scheme" in Proceedings of IEEE/ACM international conference on computer-aided design (ICCAD'07), San Jose, California, USA, pp (26–31), 2007.

[7] A. Nalamalpu, S. Srinivasan, and W. P. Burleson, “Boosters for driving long onchip interconnects—design issues, interconnect synthesis, and comparison with repeaters,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 21, no. 1, pp. 50–62, 2002.

[8] Beerel P, Roncken “Low power and energy efficient asynchronous design”. Journal of Low Power Electron Vol.3, No. 3, pp (234–253), Dec. 2007.

[9] Ben Soh, Hien Phan, Raghu “A Four-Stage Design Approach Towards Securing a Vehicular Ad Hoc Networks Architecture” Fifth IEEE International Symposium on Electronic Design, Test & Applications, Australia, IEEE Computer Society, pp (177- 282), 2010.

[10] Benini L, De Micheli G “Networks on chips: a new SoC paradigm” IEEE Computer Vol. 35, No.1, pp (70–78), Jan 2002.

[11] Bertozzi D, Benini L, De Micheli G “Error control schemes for on-chip communication links: the energy-reliability tradeoff”. IEEE Trans Computer Aided Des Integer Circuits System vol. 24, No. 6, pp 818–831, 2005.

[12] Bienia C, Kumar S, Singh JP, Li K “The PARSEC benchmark suite: characterization and architectural implications” Princeton University Technical Report TR-811-08, Jan 2008.

[13] Bjerregaard T, Mahadevan S “A survey of research and practices of Network-on-chip” ACM Computer Survey Vol. 38, No. 1, pp–51, 2006.

[14] Bogdan P, Dumitras T, Marculescu R “Stochastic communication: a new paradigm for fault-tolerant networks-on-chip”. Hindawi VLSI design, special

issue on networks-on chip, Vol. 2007, Hindawi Publishing Corporation, USA, pp (1-9).

[15] Bolotin E, Cidon I, Ginosar R, Kolodny A "QNoC: QoS architecture and design process for network on chip", Journal System Architecture (EUROMICRO J) Vol. 50, pp(105–128), Feb. 2004.

[16] C. Grecu, M. Jones, P. P. Pande, A. Ivanov, and R. Saleh, "Performance evaluation and design trade-offs for networkon- chip interconnect architectures," IEEE Transactions on Computers, Vol. 54, No. 8, pp. 1025–1040, 2005.

[17] Carloni LP, McMillan KL, Sangiovanni- Vincentelli AL "Theory of latency in sensitive design", IEEE Trans Computer Aided Des Integer Circuits System Vol. 20, No. 9, pp (1059–1076), Sep. 2001.

[18] Chatterjee S, Kishinevsky M, Ogras UY "Quick formal modeling of communication fabrics to enable verification" In Proceedings of IEEE international high level design validation and test workshop, pp (42–49) June 2010.

[19] Chou C-L, Ogras UY, Marculescu R "Energy and performance-aware incremental mapping for networks-on-chip with multiple voltage levels, IEEE Transactions of Computer Aided Design (TCAD) pp (1866–1879), 2008.

[20] C. J. Glass and L. M. Ni, "The turn model for adaptive routing," Journal of the ACM, vol. 41, no. 5, pp. 874–902, 1994.

[21] D. Park, C. Nicopoulos, J. Kim, N. Vijaykrishnan, and C. R. Das, "Exploring fault-tolerant network-on-chip architectures," in Proceedings of the 2006

International Conference on Dependable Systems and Networks, (DSN '06), pp. 93–104, Philadelphia, Pa, USA, June 2006.

[22] D. Wu, B. M. Al-Hashimi, and M. T. Schmitz, "Improving routing efficiency for network-on-chip through contention aware input selection," in Proceedings of the Asia and South Pacific Design Automation Conference, (ASP-DAC '06), pp. 36– 41, January 2006.

[23] David Atienzaa, Federico Angiolini, Srinivasan Murali,A ntonioPullini dLucaBeninic, Giovanni De Michelia, "Network-on-Chip design and synthesis outlook" Integration The VLSI Journal Elsevier, Vol. 41 , pp(340-359), 2008.

[24] Dr. Rosula S.J. Reyes, Carlos M. Oppus, Jose Claro N. Monje, Noel S. Patron, Reynaldo C. Guerrero, Jovilyn Therese B. Fajardo "FPGA Implementation of a Telecommunications Trainer System" International Journal of Circuits, Systems and Signal Processing, Issue Vol, 2, pp (87-95), 2008.

[25] D. Wentzlaff, P. Griffin, H. Hoffmann et al., "On-chip interconnection architecture of the tile processor," IEEE Micro, vol. 27, no. 5, pp. 15–31, 2007.

[26] D. Bertozzi and L. Benini, "Xpipes: a network-on-chip architecture for gigascale systems-on-chip," IEEE Circuits and Systems Magazine, vol. 4, no. 2, pp. 18–31, 2004.

[27] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, "QNoC: QoS architecture and design process for network on chip," Journal of Systems Architecture, vol. 50, no. 2-3, pp. 105–128, 2004.

[28] E. Nilsson, M. Millberg, J. Oberg, and A. Jantsch, “Load distribution with the proximity congestion awareness in a network- on-chip,” in Proceedings of the Design Automation and Test in Europe Conference, pp. 1126–1127, December 2003.

[29] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, “Routing table minimization for irregular mesh NoCs,” in Proceedings of the Design, Automation and Test in Europe Conference and Exhibition, pp. 1–6, Nice, France, April 2007.

[30] G. M. Chiu, “The odd-even turn model for adaptive routing,” IEEE Transactions on Parallel and Distributed Systems, Vol. 11, No. 7, pp. 729–738, 2000.

[31] Ganguly A et al “Scalable hybrid wireless network-on-chip architectures for multicore systems”, IEEE Trans Computing Vol. 60(, No. 10, pp (1485–1502), 2010.

[32] Ganghee Lee, Kiyoung Choi, and Nikil D. Dutt, “Mapping Multi-Domain Applications onto Coarse-Grained Reconfigurable Architectures” IEEE Transaction on Computer Aided Design of Integrated Circuits and Systems, Vol. 30, No. 5, pp (637-650) , May 2011.

[33] Hiroaki Morino Thai Thach Bao Nguyen Hoaison Hitoshi Aida Tadao Saito “A Scalable Multistage Packet Switch for Terabit IP Router Based on Deflection Routing and Shortest Path Routing” © 2002 IEEE, pp (2179-2185)

[34] Hyung Gye Lee and Naehyuck Chang, Umit Y. Ogras and Radu Marculescu “On-Chip Communication Architecture Exploration: A Quantitative Evaluation of

Point-to-Point, Bus, and Network-on-Chip Approaches" ACM Transactions on Design Automation of Electronic Systems, Vol. 12, No. 3, pp (1-20), August 2007.

[35] Hao Tian, Ajay K. Katangur, Jiling Zhong Yi Pan "A Novel Multistage Network Architecture with Multicast and Broadcast Capability" The Journal of Supercomputing, Springer, Vol.35, 2006, pp (277–300)

[36] H. Kariniemi and J. Nurmi, "Fault-tolerant XGFT network on- chip for multi-processor system-on-chip circuits," in Proceedings of the International Conference on Field Programmable Logic and Applications, (FPL '05), pp. 203–210, August 2005.

[37] H. Ito, M. Kimura, K. Miyashita, T. Ishii, K. Okada, and K. Masu, "A bidirectional- and multi-drop-transmission-line interconnect formulation point-to-multipoint on-chip communications," IEEE Journal of Solid-State Circuits, vol. 43, no. 4, pp. 1020–1029, 2008.

[38] Hu J, Marculescu R "Communication and task scheduling of application-specific networks-on-chi". IEEE Proceedings of computer Digital Tech, pp (643–651), 2005

[39] Hu J, Marculescu R "Energy and performance-aware mapping for regular NoC architectures". IEEE Trans Computer Aided Des Integer Circuits System Vol. 24, No.4, pp (551–562), 2005

[40] Hu J, Ogras UY, Marculescu R "System-level buffer allocation for application specific networks-on-chip router design". IEEE Trans Compote Aided Des Integer Circuits Syst Vol. 25, No. 12, pp (2919–2933), 2006

[41] Jason Cong, Yuhui Huang, and Bo Yuan "A Tree-Based Topology Synthesis for On-Chip Network" Computer Science Department, University of California, Los Angeles Los Angeles, USA, IEEE Conference Proceedings, pp (650-658), 2011.

[42] John. C. Bellamy, Reprint 2011" Digital Switching, Chapter 5 pp 225- 245" Digital Telehony, Wiley India Pvt. Ltd, India.

[43] J. Kim, D. Park, T. Theocharides, N. Vijaykrishnan, and C. R. Das, "A low latency router supporting adaptivity for on-chip interconnects," in Proceedings of the 42nd Design Automation Conference, (DAC '05), pp. 559–564, June 2005.

[44] J. Hu and R. Marculescu, "DyAD—smart routing for networks- on-chip," in Proceedings of the 41st Design Automation Conference, pp. 260–263, June 2004.

[45] J. Howard, S. Dighe, Y. Hoskote et al., "A 48-core IA- 32 message-passing processor with DVFS in 45nm CMOS," in Proceedings of the IEEE International Solid-State Circuits Conference Digest of Technical Papers, (ISSCC '10) San Francisco, Calif, USA, February 2010,

[46] K. P. Rane, S.V.Patil and A. M. Patil "Efficient combination of Electronics Switching System and VLSI technology" Proceedings of SPIT-IEEE Colloquium and International Conference, Mumbai, India pp (220-225)

[47] Krishnan Srinivasan, "OCP-IP Network-on-chip benchmarking workgroup Erno Salminen", Tampere University of Technology Sonics Inc. Zhonghai Lu, Royal Institute of Technology, pp (1-5) December 2010,

[48] G. Ascia, V. Catania, M. Palesi, and D. Patti, "Neighbors-on path: a new selection strategy for on-chip networks," in Proceedings of the IEEE/ACM/IFIP

Workshop on Embedded Systems for Real Time Multimedia, (ESTIMEDIA ’06), pp. 79–84, Seoul, Korea, October 2006.

[49] G. DeMicheli and L. Benini, Networks on Chips: Technology and Tools, Morgan Kaufmann,Waltham, Mass, USA, 2006.

[50] L. Seiler, D. Carmean, E. Sprangle et al., “Larrabee: a manycore x86 architecture for visual computing,” IEEE Micro, vol. 29, no. 1, pp. 10–21, 2009.

[51] Luca Benini, Giovanni De Micheli Networks on Chips: A New SoC Paradigm, IEEE Computer Society, SOC Design, pp (70-79), January (2002).

[52] Lee HG, Chang N, Ogras UY, Marculescu R “On-chip communication architecture exploration: a quantitative evaluation of point-to-point, bus and network-on-chip approaches”. ACM Transactions of Design Automation Electronic System, Vol. 12, No. 3, pp (1–20), 2007.

[53] Liang J, Laffely A, Srinivasan S, Tessier R An architecture and compiler for scalable on-chip communication. IEEE Transactions Very Large Scale Integrated System Vol. 12, No.7, pp (711–726), 2004.

[54] Muhammad Aqeel Wahlah, Kees Goossens, “A test methodology for the non-intrusive online testing of FPGA with hardwired network on chip” Microprocessors and Microsystems, (2012), pp (1-18)

[55] M. T. Schmitz, B. M. Al-Hashimi, and P. Eles, “Iterative schedule optimization for voltage scalable distributed embedded systems,” ACM TECS, Vol. 3, No. 1, pp. 182–217, 2004.

[56] M. Kistler, M. Perrone, and F. Petrini, “Cell multiprocessor communication network: built for speed,” IEEE Micro, Vol. 26, No. 3, pp. 10–23, 2006.

[57] M. Millberg, E. Nilsson, R. Thid, and A. Jantsch, “Guaranteed bandwidth using looped containers in temporally disjoint networks within theNostrum network on chip,” in Proceedings of the Design, Automation and Test in Europe Conference and Exhibition, (DATE ’04), pp. 890–895, February 2004.

[58] M. Li, Q. A. Zeng, and W. B. Jone, “DyXY: a proximity congestion aware deadlock-free dynamic routing method for network on chip,” in Proceedings of the Design Automation Conference, pp. 849–852, July 2006.

[59] M. T. Schmitz, B. M. Al-Hashimi, and P. Eles, “Energy-efficient mapping and scheduling for DVS enabled distributed embedded systems,” in Proceedings of the Conference on Design, Automation and Test in Europe, pp. 514–521, March 2002.

[60] M. Pirretti, G. M. Link, R. R. Brooks, N. Vijaykrishnan, M. Kandemir, and M. J. Irwin, “Fault tolerant algorithms for network-on-chip interconnect,” in Proceedings of the IEEE Computer Society Annual Symposium on VLSI, pp. 46–51, February 2004.

[61] M. D. Harmanci, N. P. Escudero, Y. Leblebici, and P. Ienne, “Providing QoS to connection-less packet-switched NoC by implementing diffServ functionalities,” in Proceedings of the International Symposium on System-on-Chip, pp. 37–40, November 2004.

[62] M. A. Yazdi, M. Modarressi, and H. Sarbazi-Azad, “A load balanced routing scheme for NoC-based systems-on-chip,” in Proceedings of the 1st Workshop on Hardware and Software Implementation and Control of Distributed MEMS, (DMEMS ’10), pp. 72–77, Besan, TBD, France, June 2010.

[63] M. Daneshtalab, A. A. Kusha, A. Sobhani, Z. Navabi, M. D. Mottaghi, and O. Fatemi, “Ant colony based routing architecture for minimizing hot spots in NOCs,” in Proceedings of the Annual Symposium on Integrated Circuits and System Design, pp. 56–61, September 2006.

[64] M. Dall’Osso, G. Biccari, L. Giovannini, D. Bertozzi, and L. Benini, “Xpipes: a latency insensitive parameterized network on- chip architecture for multi-processor SoCs,” in Proceedings of the 21st International Conference on Computer Design, (ICCD ’03), pp. 536–539, October 2003.

[65] Madsen J, Mahadevan S, Virk K, Gonzales M “Network-on-chip modeling for system-level multiprocessor simulation” In: Proceedings of the IEEE international real-time systems symposium, pp (82–92), Dec 2003

[66] Marculescu R, Ogras UY, Peh L, Jerger NE, Hoskote Y “Outstanding research problems in NoC design: system, microarchitecture, and circuit perspectives” IEEE Trans Computer Aided Design Integrated Circuits System Vol. 28, No. 1, pp (3–21), 2009.

[67] M. D. Harmanci, N. P. Escudero, Y. Leblebici, and P. Ienne, “Quantitative modelling and comparison of communication schemes to guarantee quality-of-service in networks-on-chip,” in Proceedings of the IEEE International Symposium on Circuits and Systems, (ISCAS ’05), pp. 1782–1785, May 2005.

[68] M. Ali, M. Welzl, S. Hessler, and S. Hellebrand, “A fault tolerant mechanism for handling permanent and transient failures in a network on chip,” in Proceedings of the 4th International Conference on Information Technology-New Generations, (ITNG ’07), pp. 1027–1032, Las Vegas, Nev, USA, April 2007.

[69] M. Yang, T. Li, Y. Jiang, and Y. Yang, “Fault-tolerant routing schemes in RDT(2,2,1)/α-based interconnection network for networks-on-chip designs,” in Proceedings of the 8th International Symposium on Parallel Architectures, Algorithms and Networks, (I-SPAN ’05), pp. 1–6, December 2005.

[70] Najla Alfaraj, Yang Xu, H. Jonathan Chao “A Practical and Scalable Congestion Control Scheme for High-Performance Multi-Stage Buffered Switches” IEEE 13th international conference on high performance routing and switching, pp (44-52), 2012.

[71] Nigussie E, Lehtonen T, Tuuna S, Plosila J, Isoaho J “High-performance long NoC link using delay-insensitive current-mode signaling”, Hindawi VLSI Des (special issue on networks-on-chip), pp (1–13), 2007.

[72] N. Kavaldjiev, G. J. M. Smit, P. G. Jansen, and P. T. Wolkotte, “A virtual channel network-on-chip for GT and BE traffic,” in Proceedings of the IEEE Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures, pp. 211–216, Karlsruhe, Germany, March 2006.

[73] Ogras UY, Marculescu R “It’s a small world after all: NoC performance optimization via long-range link insertion”. IEEE Transactions Very Large Scale Integrates System, Special Section Hardware Software Co-design System Synthesis Vol. 14, No. 7, pp (693–706), 2006.

[74] Ogras UY, Marculescu R, Marculescu D, Jung EG “Design and management of voltage-frequency island partitioned networks-on-chip” IEEE Trans Very Large Scale Integration System Vol. 17, No.3, pp (330–341), 2009.

[75] Paolo Meloni, Igor Loi, Federico Angiolini, Salvatore Carta,“Area and Power Modeling for Networks-on-Chip with Layout Awareness” Hindawi Publishing Corporation VLSI Design, Volume 2007, pp (1-12)

[76] Pande PP, Grecu C, Jones M, Ivanov A, Saleh R "Performance evaluation and design trade-offs for network-on-chip interconnect architectures" IEEE Transactions Computing Vol. 54, No. 8, pp(1025–1040), Aug. 2005.

[77] P. Bogdan, T. Dumitras, and R. Marculescu, "Stochastic communication: a new paradigm for fault tolerant networks on chip," VLSI Design, vol. 2007, Article ID 95348, 17 pages, 2007.

[78] P. Vellanki, N. Banerjee, and K. S. Chatha, "Quality-of-service and error control techniques for network-on-chip architectures," in Proceedings of the ACM Great lakes Symposium on VLSI, (GLSVLSI '04), pp. 45–50, April 2004.

[79] P. C. Chang, I.W.Wu, J. J. Shann, and C. P. Chung, "ETAHM: an energy-aware task allocation algorithm for heterogeneous multiprocessor," in Proceedings of the 45th Design Automation Conference, (DAC '08), pp. 776–779, Anaheim, Calif, USA, June 2008.

[80] R. Marculescu, U. Y. Ogras, L. S. Peh, N. E. Jerger, and Y. Hoskote, "Outstanding research problems in NoC design: system, microarchitecture, and circuit perspectives," IEEE Transactions on Computer, vol. 28, no. 1, pp. 3–21, 2009.

[81] Mike Santarini "FPGA Command Centre Stage in Next-Gen Wired Networks" XCell Journal Today, Vol.1 issue 67, (2009), pp (10-14)

[82] Nikos Sklavos, Alexabdros Papakonstinou, Spyros Theoharis Odysseas Koufopavlou, “Low-power Implementation of an Encryption/Decryption System with Asynchronous Techniques”, VLSI Design, Taylor and Fransis 2002 Vol. 15 (1), pp. (455–468)

[83] Nikos Chrysos, Lydia Y. Chen, Cyriel Minkenberg, Christoforos Kachrism and Manolis Katevenis “End-to-end congestion management for non-blocking multi-stage switching fabrics” ACM digital Library, (2010), pp(1-2)

[84] Prosanta Gope, Ashwani Sharma Ajit Singh Nikhil Pahwa “An Efficient Cryptographic Approach for Secure Policy Based Routing (TACIT Encryption Technique)”, Conference Proceedings, IEEE Xplorer, (2011), pp (359-363)

[85] S. Kumar, A. Jantsch, and J. P. Soininen, “Network-on-chip architecture and design methodology,” in Proceedings of the International Symposium on Very Large Scale Integration, pp. 105–112, April 2000.

[86] Shim B, Shanbhag “NR Energy-efficient soft-error tolerant digital signal processing.” IEEE Transactions on VLSI Vol. 14, No. 4, pp (336–348), 2006.

[87] Simunic Rosing T, Mihic K, De Micheli G “Power and reliability management of SOCs”. IEEE Transactions on VLSI Vol. 15, pp (391–403), 2007

[88] Srinivasan K, Chatha KS, Konjevod G “Linear programming based techniques for synthesis of network-on-chip architectures “. IEEE Transactions on Very Large Scale of Integration System, Vol. 14, No. 4, pp (407–420), 2006.

[89] Stuijk S, Basten T, Geilen M, Ghamarian AH, Theelen B “Resource efficient routing and scheduling of time-constrained streaming communication on

networks-on-chip”. Journal of System Architect (the EUROMICRO Journal) 54(3–4):411–426, 2008.

[90] Teijo Lehtonen, Pasi Liljeberg, and Juha Plosila “Online Reconfigurable Self-Timed Links for Fault Tolerant NoC” Hindawi Publishing Corporation VLSI Design (2007), pp (1-13)

[91] T. Vishwanathan , 2011 “ Electronics Space Division Switching , Chapter 4, “Telecommunication switching system and networks” India, PHI Publisher, pp 125-140”

[92] Taylor MB, Lee W, Amarasinghe S, Agarwal “A Scalar operand networks “. IEEE Trans Parallel Distributed System (special issue on on-chip networks) Vol. 16, No. 2, pp (145–162), 2005.

[93] T. Bjerregaard and J. Sparso, “A router architecture for connection- oriented service guarantees in the MANGO clockless network-on-chip,” in Proceedings of the Design, Automation and Test in Europe, (DATE ’05), pp. 1226–1231, March 2005.

[94] T. Schonwald, J. Zimmermann, O. Bringmann, and W. Rosenstiel, “Fully adaptive fault-tolerant routing algorithm for network-on-chip architectures,” in Proceedings of the 10th Euromicro Conference on Digital System Design Architectures, Methods and Tools, (DSD ’07), pp. 527–534, Lubeck, Germany, August 2007.

[95] T. Lehtonen, P. Liljeberg, and J. Plosila, “Online reconfigurable self-timed links for fault tolerantNoC,” VLSI Design, vol. 2007, Article ID 94676, 13 pages, 2007.

[96] V. Kianzad, S. S. Bhattacharyya, and G. Qu, “CASPER: an integrated energy-driven approach for task graph scheduling on distributed embedded systems,” in Proceedings of the IEEE 16th International Conference on Application-Specific Systems, Architectures, and Processors, (ASAP ’05), pp. 191–197, July 2005.

[97] Varatkar G, Marculescu R “On-chip traffic modeling and synthesis for MPEG-2 video applications”. IEEE Transaction VLSI, Vol. 12, No. 1, pp (108–119), 2004.

[98] Vasilis F. Pavlidis, Eby G. Friedman “3-D Topologies for Networks-on-Chip” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 15, No. 10, pp (1081-1091), October 2007.

[99] Wen-Chung Tsai, Ying-Cherng, Lan,Yu-Hen Hu, and Sao-Jie Chen “Networks on Chips: Structure and Design Methodologies” Hindawi Publishing Corporation., Journal of Electrical and Computer Engineering pp (1-13), 2012.

[100] W. Wolf, The future of multiprocessor systems-on-chips, in: Proceedings of the 41st Design Automation Conference (DAC’04), June2004, pp. 681–685.

[101]www.digilentinc.com/Press/SalesSheets/XUPV5-datasheet-06.pdf

[102]www.xilinx.com/support/documentation/../xilinx13.../ise_tutorial_ug695.pdf

[103] Xinmiao Zhang, and Keshab K. Parhi “High-Speed VLSI Architectures for the AES Algorithm”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 12, No. 9, pp (957-968), Sep. 2004.

[104] Y. C. Lan, S. H. Lo, Y. C. Lin, Y. H. Hu, and S. J. Chen, “BiNoC: a bidirectional NoC architecture with dynamic self reconfigurable channel,” in

Proceedings of the 3rd ACM/IEEE International Symposium on Networks-on-Chip, (NoCS ’09), pp. 266–275, May 2009.

[105] Zhao D, Wang Y “SD-MAC: design and synthesis of a hardware-efficient collision free QoS-aware MAC protocol for wireless Network-on-Chip” IEEE Transactions Computing TC Vol. 8, pp (1046–1057), 2008

[106] Z. Guz, I. Walter, E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, “Efficient link capacity and QoS design for networkon- chip,” in Proceedings of the Design, Automation and Test in Europe, (DATE ’06), pp. 1–6, March 2006.

APPENDIX – A

FPGA DESIGN

A.1 INTRODUCTION TO FPGA

Field Programmable Gate Arrays (FPGAs) are configurable integrated circuits that can be used to design digital circuits and programmable chips. Hardware description languages such as VHDL or Verilog HDL are used to specify the configuration of FPGA. FPGA offers significant advantages in digital logics and integrated circuits because of the reconfigurability feature as well as non-recurring engineering (NRE) cost. It is not like Application Specific Integrated Circuits (ASICs) where designers do not have the flexibility of design modifications after the chip is manufactured by fabrication unit. FPGA is treated like a blank canvas where the design is "painted" according to the constraints of the designer's needs & the FPGA's capabilities. A FPGA is a device in which the final logic structure can be directly configured by the end user, without the use of an integrated circuit fabrication. FPGA is similar to a Programmable Logic Device (PLD), but PLDs are generally limited to hundreds of gates, FPGAs support thousands or more gates.

FPGA contains a two dimensional arrays of logic blocks and interconnections between logic blocks. The logic blocks and interconnects both are programmable. These logic blocks are programmed to implement a desired Boolean function and interconnects are programmed using the switch boxes to connect the logic blocks. Let us consider, if we want to implement a complex design, then the design is divided into small sub functions and each sub function is implemented using one logic block. Now, to get our desired design (CPU), all the sub functions implemented in logic blocks must be connected and this is done by programming interconnects. The internal structure of an FPGA is shown in the figure A.1. In FPGA designer can put the design according to its capability to reprogram it and the device functions according the program is loaded in the FPGA device.

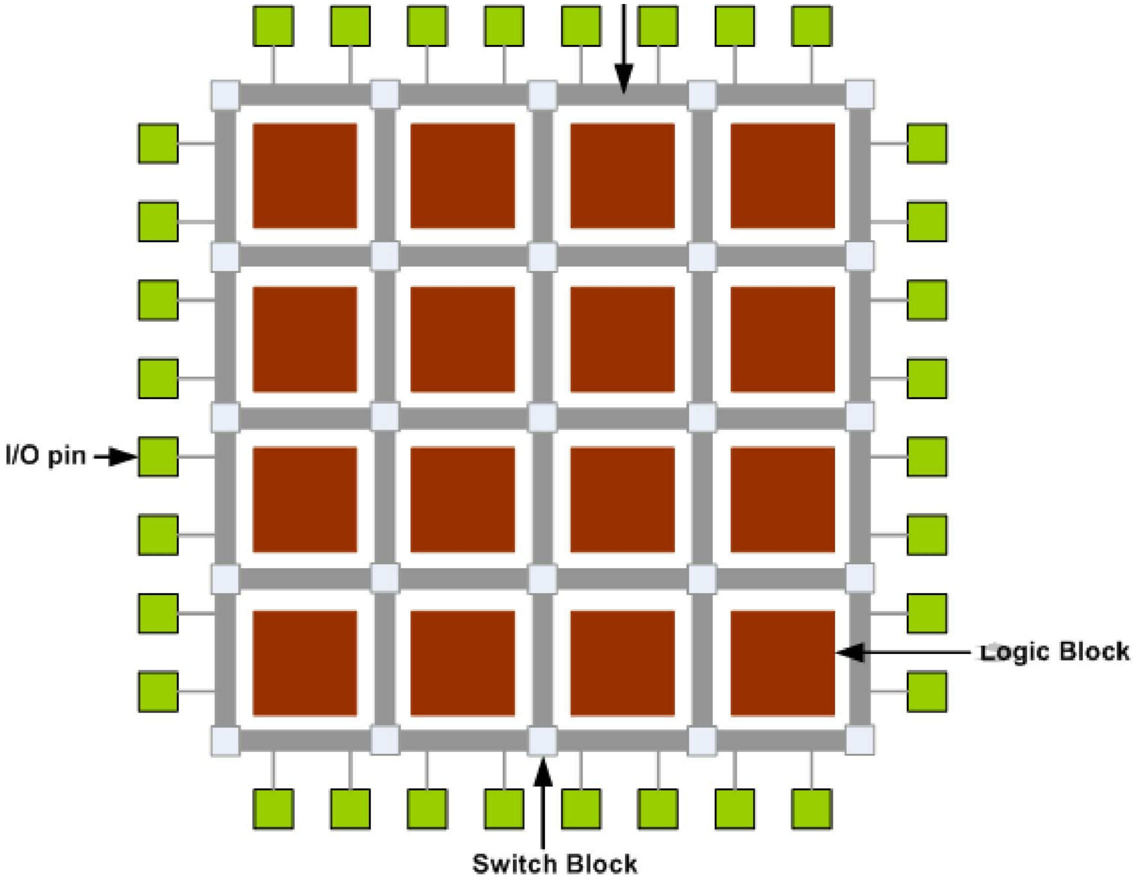


Fig. A.1 FPGA Architecture [102]

FPGAs are alternative approach to the custom ICs, are be used to implement an entire System On one Chip (SOC). The main advantage of FPGA is ability to reprogram. Designer can reprogram an FPGA device to implement a design and this is done after the FPGA is manufactured. It beings the name custom ICs which is due to filed programmability. These ICs are expensive and takes longer time to design so they are useful when produced in bulk amounts. But FPGAs follows the shortest time to market and easy to implement within a short time with the help of Computer Aided Designing (CAD) tools.

Xilinx logic block consists of one Look up Table (LUT) [7] and one Flip Flop. An LUT [7] is used to implement number of different functionality. LUTs handle the input lines to the logic block to go into and enable its port. The output of the LUT gives the result of the logic function that it implements and the output of logic block is registered or unregistered output from the LUT. SRAM [7] [21] is used to implement a LUT. For an example a k-input logic function is implemented using 2^k * 1 size SRAM. Total number of possible different functions for k input LUT is 2^2^k. The main advantage of such architecture is that it supports implementation of so many logic functions, although the disadvantage is unusually large number of memory cells required to implement such a logic block in case number of inputs is large. Figure A.2 shows a 4-input LUT based implementation of logic block of FPGA. LUT based design provides for better logic block utilization. A logic block having k-input LUT can be implemented in number of different ways with trade off between performance and logic density.

An n-LUT can be shown as a direct implementation of a function truth-table. Each of input combination is hold by the input latch that holds the value of the function corresponding to every input based on truth table. For Example: 2 input LUT can be used to implement 16 types of functions like AND, OR, A+ not B.... etc.

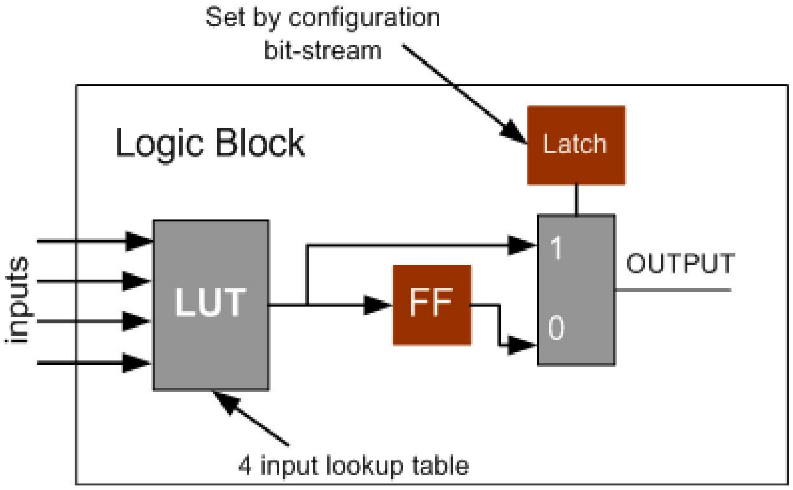


Fig A.2 Xilinx LUT [102]

Table A.1 Truth table for logic design

Inputs		**Output Logic**	
A	B	AND	OR
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

A.1.1 INTERCONNECTS

A wire segment can be described as two end points of an interconnect [8] with no programmable switch between them. There is a sequence of one or more wire segments in an FPGA can be termed as a track. An FPGA has logic blocks, interconnects and switch blocks (Input/output blocks) which are typically interconnected. In FPGA, switch blocks lie in the periphery of logic blocks and interconnect. Switch blocks have wire segments which are connected to logic blocks through switch blocks. Based on the required design, one logic block is connected to another and so on.

Since clock signals are normally routed via special-purpose dedicated routing networks in commercial FPGAs, clock and other signals are separately managed. In the architecture, the locations of the FPGA logic block [8] [9] pins are shown below. Each input of FPGA is accessible from one side of the logic block, although the output pin can connect to routing wires in both the channel to the right and the channel below the logic block. He output pin of each logic block pin can connect to any of the wiring segments in the channels adjacent to it. In the same way, an I/O pad can connect to any one of the wiring segments in the channel adjacent to it. For an example, I/O pad at the top of the chip can connect to any of the W wires (W is the channel width) in the horizontal channel immediately below it. FPGA routing also depends on the number of logic inputs assigned to the LUT.

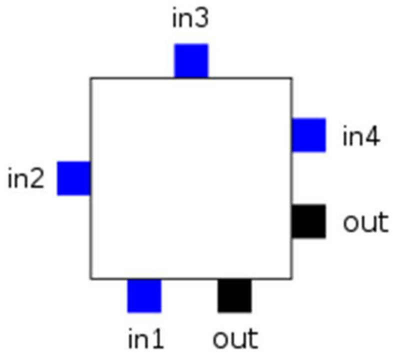


Fig. A.3 Logic block pin locations [102]

Generally, the FPGA routing is not in segment form in which each wiring segment spans only one logic block before it terminates in a switch box. With the help of turning on some of the programmable switches within a switch box, it is possible to construct longer paths. Some FPGA architectures use longer routing lines that span multiple logic blocks for higher speed interconnects. There exists a switch box, whenever a vertical and a horizontal channel intersect. In the architecture, when a wire enters a switch box. In switch matrix, there are three programmable switches that allow it to connect to three other wires in adjacent channel segments. The topology or pattern, of switches used in this architecture is the planar or domain-based switch box topology.

In the switch box topology, a wire in track number one connects only to wires in track number one in adjacent channel segments and wires in track number 2 connect only to other wires in track number 2 and so on. The figure A.4 shown below illustrates the connections in a switch box. Switch box consist of wire segments and programmable structure which can be reconfigured many times.

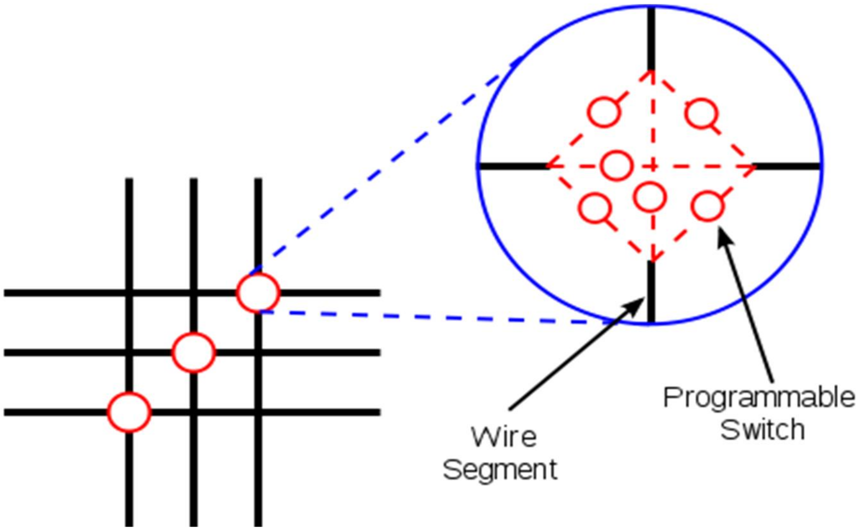


Fig. A.4 Switch box topology [102]

Modern FPGA families expand upon the above capabilities to include higher level functionality fixed into the silicon. The feature of having these common functions embedded into the silicon reduces the requirement of area and gives those functions increased speed compared to building them from primitives. Examples of such logics include logic gates, multipliers, generic DSP blocks, high speed IO logic, embedded processors and embedded memories modules. FPGAs are also widely used for systems validation including pre-silicon validation pr pre synthesis, post-silicon validation or post synthesis, and firmware development. It allows chip fabrication companies to validate their design before the chip is produced in the fabrication plant, reducing the time-to-market.

To shrink the size and power consumption of FPGAs, different vendors such as Tabula and Xilinx have introduced new 3D or stacked architectures [8] following the introduction of its 28 nm 7-series FPGAs [7]. Xilinx revealed that several of the highest-density parts in those FPGA product lines will be constructed using multiple dice in one package, which employs technology development for 3D construction and stacked-die assemblies. The technology stacks several active FPGA dice side by side on a silicon interposer; a single piece of silicon that carries passive interconnects.

APPENDIX –B

SIMULATION TOOLS

The designing of chip and FPGA implementation includes the following software development tools.

B.1 XILINX ISE 14.2

Xilinx [7] [8] has been a semiconductor industry leader at the forefront of technology, market and business achievement. It is a tool to design the IC and to view their RTL (Register Transfer Level) schematic .It is a tool to test the code on FPGA environment and the values of all parameters details required to implement the Chip. The detail of the synthesized results includes the hardware details, utilization of hardware parameters, memory utilization and timing information. Device utilization report gives the percentage utilization [13] of device hardware for the chip implementation. Device hardware includes No of slices, No of flip flops, No of input LUTs, No. of bounded IOBs, and No of gated clocks (GCLKs) used in the implementation of design. Timing [13] details provides the information of delay, minimum period, maximum frequency, minimum input arrival time before clock and maximum output required time after clock

B.2 MODELSIM 10.1 B

Mentor Graphics [9] was the first to combine single kernel simulator (SKS) technology with a unified debug environment for Verilog HDL, VHDL, and System C. The simulation and synthesis combination of industry-leading and native SKS performance with the best integrated debug and analysis environment make Modelsim the simulator of choice for both ASIC and FPGA design. The design platform and standards support in the industry make it easy to adopt in the majority of process and tool flows.

B.2.1 MODELSIM: SIMULATION AND DEBUG

ModelSim EE [9] is the industry-leading, Windows-based simulator for VHDL, Verilog, or mixed-language simulation environments.

B.2.1.1 ModelSim EE Features

- Partial VHDL 2008 support
- It has transaction wlf logging support in all languages including VHDL
- Windows7 a 32 Support
- Secure IP support
- System C option
- RTL and Gate-Level Simulation
- Integrated Debug Environment
- Verilog, VHDL and SystemVerilog Design
- Mixed-HDL Simulation option
- Code Coverage option

B.2.1.2 ModelSim EE Benefits

Model Sim EE has proven the following benefits

- ModelSim EE is a Cost-effective HDL simulation solution,

- It has intuitive GUI for efficient interactive debug,
- Supports to integrated project management simplifies managing project data,
- It is Easy to use with outstanding technical support,
- Support for sign-off for popular ASIC libraries,
- It supports hardware debugging
- Used in functional simulation

B.2.1.3 Simulation and Design Steps

The diagram has shown in figure B.1 the basic steps for simulating a design in ModelSim.

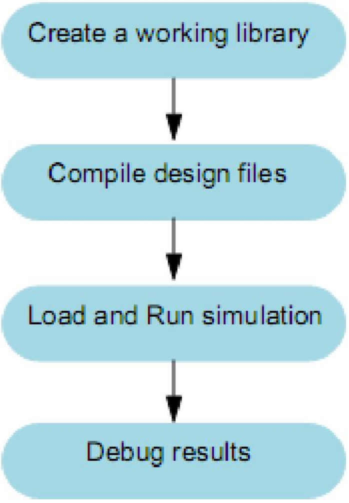


Fig B.1 Chip Design Process Flow [5]

- *Creating the Working Library:* In ModelSim SE, all programs are designed and compiled into a library. Typically start a new simulation in ModelSim by creating a working library called "work," which is the default library name used by the compiler as the default destination for compiled design units.

- *Compiling Design:* After creating the working library, design is being compiled into it. The ModelSim library format is compatible across all supported platforms.
- *Loading and Running the Simulator with the Design:* With the design compiled, we load the simulator with design by invoking the simulator on a top-level module (Verilog) or a configuration or entity/architecture pair (VHDL). Assuming the design is loaded successfully, and the simulation time is set to zero. There is the requirement to enter a run command to begin simulation.
- Debugging: ModelSim's robust debugging environment is used to track down the cause of the problem.

B.3 DESIGN VERIFICATION

The design is developed in Xilinx tool and functionally checked in Modelsim simulation tool. After the simulation, the design is verified Verification can be done at different stages of the process steps.

B.3.1 BEHAVIOURAL SIMULATION (RTL SIMULATION)

The functional simulation is performed before synthesis process to verify RTL (behavioral) code and to confirm that the design is functioning as intended. Behavioral modeling design and simulation can be performed on either VHDL or Verilog HDL designs. In the simulation process, signals and variables are used to pass intermediate values, procedures and functions are traced and breakpoints are set. In fast simulation designer sis allowed to change the HDL code if the required functionality is not met with in a short time period. Timing and resource usage properties are still unknown, since the design is not yet synthesized to gate level.

B.3.2 FUNCTIONAL SIMULATION (POST TRANSLATE SIMULATION)

Functional simulation gives information about the logic operation of the circuit design. Designers can verify the functionality of the design using this process after the Translate process. If the developed code is not meeting the expected functionality, then the designer has to made changes in the code and again follow the design flow steps. Changes in the code and again follow the design flow steps.

B.3.3 STATIC TIMING ANALYSIS (STA)

This can be done after MAP or PAR processes. Timing report relating to post MAP lists signal path delays of the design derived from the design logic. Timing report relating to post Place and Route incorporates timing delay information to provide a comprehensive timing summary of the design.

B.4 IP BASED DESIGN, HARD AND SOFT MACROS

Logical primitives [18] using cell libraries are usually provided by the device manufacturer as part of the service. They will incur no additional cost and their release will be covered by the terms of a Non Disclosure Agreement (NDA) [29] and they will be regarded as intellectual property by the manufacturer. The physical design based on it is predefined so they could be termed "hard macros". What most engineers understand as IP cores, designs purchased from a third party as sub components of a larger ASIC design. These design can be provided as an HDL description [11], or as a fully routed design that could be printed directly onto an ASIC's mask. Many organizations now sell such predesigned cores CPUs, Ethernet, USB or telephone interfaces [88] and larger organizations may have an entire department or division to produce cores for the rest of the organization. There are a lot of functions available in IP design, as a core takes a lot of time and investment to create, its further development cuts product cycle times and reuse dramatically and creates better products.

Additionally, organizations such as open cores are collecting free IP cores paralleling the open source movement in software. Soft macros are often process independent because they can be fabricated on a wide range of manufacturing processes and different manufacturers. Hard core macros are limited and usually further design effort must be invested to migrate to a different process or manufacturer. ASIC design [8] is based on a design flow that uses HDL. Most Electronic Design Automation (EDA) tools used for ASIC flow are compatible with both Verilog HDL and VHDL. In the design flow, the code is synthesized with the help of Xilinx tool. In this process the RTL code is converted into logic gates and flip flops, multiplexers, memory devices etc. The logic gates synthesized will have the same logic functionality as described in the RTL code [102]. In next step, a synthesis tool is required to convert the RTL code to logic gates. Most common tools used in the ASIC industry include Synopsys's Design Compiler, Mentor Graphics, Xilinx and Cadence's Ambit. The synthesis process requires two other input files to make the conversion from RTL to logic gates.