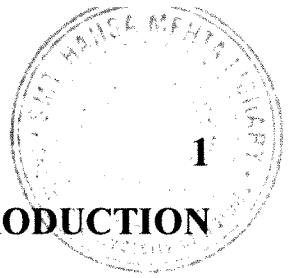
CHAPTER 1

INTRODUCTION



1 INTRODUCTION

1.1 INTRODUCTION

Explosive growth in the field of computers, internet and ICE (Information, Communication and Entertainment) increases the bandwidth and speed demand. The provision of broadband services to end users will sustain internet growth for many years to come. In meeting the needs of the next generation networks, today's networking products need to become more sophisticated than ever. They must not only be able to switch and route at wire speed, but they must process a vast array of new services such as Voice over IP (VoIP), Multiprotocol Label Switching (MPLS), streaming audio and video, multi-media message services and Virtual Private Networks (VPNs). Slow, general-purpose processors (GPP) or expensive-to-develop ASICs simply cannot meet this new market's dual requirements of performance and flexibility because development cycle of ASIC (18 months) is too long to accurately hit fast moving market requirements and GPP can not satisfy performance requirement because it is not optimized for the networking application. Inability to dynamically revise features during development imposes a high degree of risk. The race to increase speed of data transfer in Internet has created a specialized corner in the microprocessor market. Internet (i.e. network) traffic consists of packets from thousands of different flows, each requiring relatively less processing. As a result, processors can parallelize much of the workload since there are no interdependencies among packets of different flows. It is therefore crucial to develop highly scalable processing systems that can meet the demands for faster links and more complex processing.

In fourth generation network system, network processor subsystem, switching fabric and line cards are the key components as shown in figure 1.1.

1. Line cards: They offer a physical interface and can be connected to different network media, such as coaxial cable, twisted pair cable and optical fibers.

2. Network Processor subsystem: It interfaces to the physical data links, decodes incoming traffic, and applies traffic shaping, classification, modification, forwarding, prioritizing and other function like segmenting IP packets into fixed-length cells (such as ATM cells) at the input sides, and assembling the cells into IP packets at the output side.

3. Switching Fabric: Actual data transmission across ports occurs on the switch fabric, which includes a cross point, a scheduler, and buffers. The cross point is a configurable interconnecting element that dynamically establishes links between input and output ports [18].

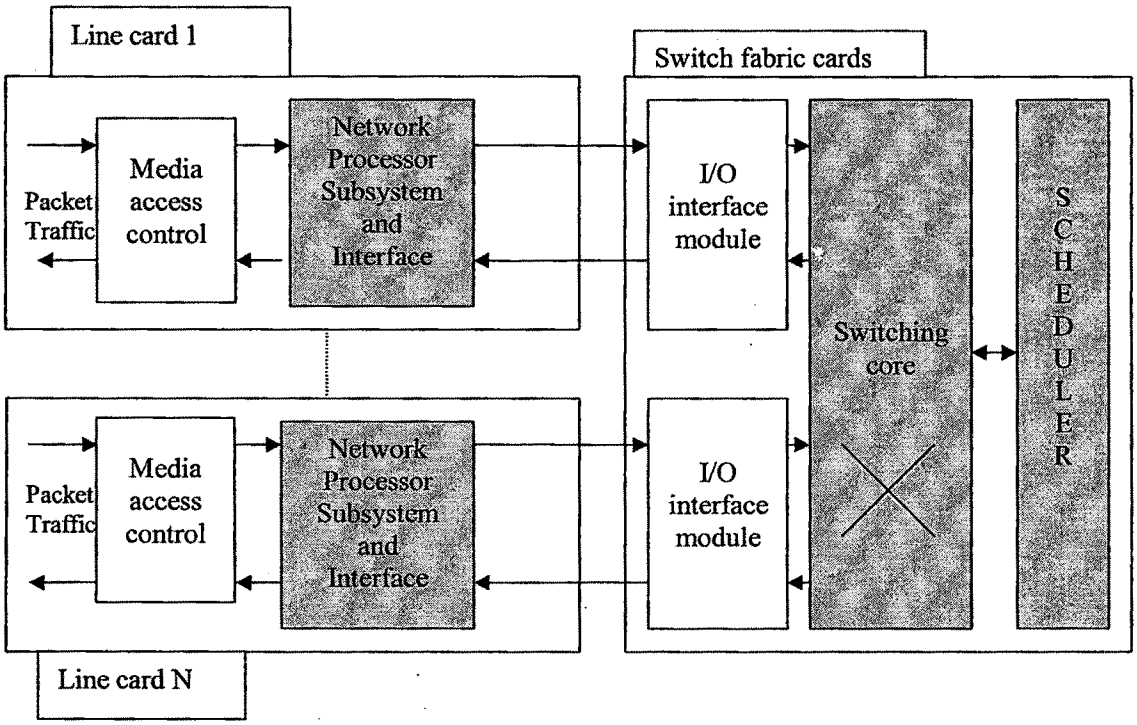


Figure 1.1 Common switch fabric architecture, and interfacing with network processor

Shaded blocks as shown in figure 1.1, are the focus of the thesis. The thesis focuses on implementation of packet processing functions and simulation, implementation and comparison of standard scheduling algorithms for input queued crossbar switching fabric. It also proposes two scheduling algorithms for input queued crossbar switching fabric.

1.2 NETWORK PROCESSOR

Modern VLSI technology is limited by bandwidth rather than arithmetic processing capacity. It is possible to place hundreds of ALUs on a chip. However, they create bandwidth demands on registers, cache, and main memory that is difficult to be satisfied. DSL, Cable Modems, B-band wireless, and cheaper fiber-based access solutions are finally opening up WAN access. At the present growth rate (doubling every four months), aggregate Internet bandwidth will increase 32768x, but in 5 years processing power will increase 8x (doubling 3 times) [27]. Network Processors usually consist of multiple processing units such as CPU cores, micro-engines, and dedicated hardware for compute-intensive tasks such as header parsing, table look-up and encryption/decryption. Together with these, there are also memory units, caches, interconnections, and I/O interfaces. Following a system-on-a-chip (SoC) design method, these resources are put on a single chip and must interoperate to perform packet-processing tasks at line speed.

1.2.1. PACKET ENCRYPTION FUNCTION OF NETWORK PROCESSOR

Network processors are specialized CPUs optimized to support the implementation of network protocols at wire line speed and will become critical components of next-generation networking equipment. The vulnerability of Internet from attacks by intruders and hackers, as well as expansion of private networks over the Internet has made the security aspects as a key element in designing Network Processor architecture. Thus combinations of cryptographic applications and communication applications have become important issues of network processor design. The most important criteria of selection of cryptographic algorithm and its implementation depend on the speed, ease of implementation and security of the packets to be processed by the NPU (Network Processor Unit). We have implemented IDEA (International Data Encryption Algorithm) cryptographic algorithm as a network processor element.

1.2.2. PACKET CLASSIFICATION FUNCTION OF NETWORK PROCESSOR

To classify NPU, two aspects are important. 1. Where in a system NPU fit and 2. Internal architecture type. Different NPU can fit at router, terminal and line card and use combinations of RISC, ASIC, ASIP and reconfigurable architectures. Packet classification is one of the main tasks of NPU. A technique that extracts values from the headers of an incoming packet and uses these values to identify the protocol, used within the packet and to map the packet to flow is called "Packet Classification". Packet Classification can be carried out according to the application and priority like voice, multimedia and video. Packet classification efficiency is evaluated by parameters like search time, storage space requirement, update time, and scalability in the number of header fields used for classification. We realize packet classification in software by C program. We also realize packet classification for router using hardware and packet classification for network terminal using hardware software co-design.

1.2.2.1. SOFTWARE REALIZATION OF PACKET CLASSIFIER

In the case of web traffic, static classification is used, in which values for header fields can be determined a priori. Several classification rules are needed to define web traffic. Conceptually, a classifier needs to compute logical AND of the several conditions specified in the rules. The most conventional implementation of a classifier uses a series of tests to examine one header field at a time. As soon as it finds a field that does not match a specified value, the classifier stops and declares that the packet is not a match. Only after completion of all tests, the classifier asserts that a packet matches. We have written a C program to demonstrate the same.

1.2.2.2. HARDWARE REALIZATION OF PACKET CLASSIFIER

Rapid growth of the Internet has caused increasing congestion and packet loss at intermediate routers. As a result, some users are willing to pay a high price, for QoS from the network. QoS normally deals with guaranteed bandwidth and guaranteed delay from source to destination with a certain high probability. Thus packets that are part of certain connections may get prioritized treatment over other packets. For real time voice and video transmission, QoS is essential. To maximize their revenue, the ISPs also wish to have the capability of providing differentiated services to its users, while still deploying one common network infrastructure.

To provide such service, flow-aware router is required which performs special processing like filtering packets for security reasons, delivering packets according to a pre-agreed delay guarantee, and treating high priority packets preferentially, on incoming packets. This special processing requires that the router classify incoming packets into one of several *flows* — all packets of a flow obey a pre-defined rule and are processed in a similar manner by the router. Each rule specifies a flow that a packet may belong to based on some criteria on the contents of the packet header. All packets belonging to the same flow are treated in a similar manner. The identified flow of an incoming packet specifies an *action* to be applied to the packet.

Increase in link speeds and traffic volume imposes constraints on packet classification solutions. Ternary Content Addressable Memory (TCAM) is a memory mechanism that supports rapid (i.e. parallel) searching as well as data storage. TCAM is especially helpful in producing a high-speed classification mechanism, due to the fast and deterministic lookup performance afforded by their use of massive parallelism. The TCAM can bind the packet to a specific classification ID thereby distinguishing and isolating the traffic in different flows. Here, we implement the packet classifier using Ternary Content Addressable Memory (TCAM) which produces multi-match classification in VHDL, by using Quartus II tool of Altera to get the chip area requirement.

1.2.2.3. HARDWARE/SOFTWARE CO-DESIGN OF PACKET CLASSIFIER

In a network terminal protocol processing, using hardware software co-design type methodology, incoming frame is classified/processed on the fly i.e. before it is stored in the memory. Hence reducing time and memory requirements and increasing the possibilities to handle higher data rate frames. The architecture deals with classification of single packet or frame. For task that involves several packets, the implemented architecture only provides supporting functions, such as extraction, classification and information pre-processing. The processor architecture is re-configurable; hence classification of newer protocols can be carried out, when required. This architecture can be extended to router also.

1.3. THE SWITCHING FABRIC

Switching fabric serves to interconnect the various functional units of switch and router. It is a critical design component in all high performance switching systems. All switched traffic crosses the fabric, and under heavy network loads, interconnect capacity can easily become the bottleneck, limiting overall throughput. Three basic architectures are commonly used for switches and routers: bus interconnects, shared memory interconnects and crossbar interconnects. Each of these architecture has specific trade-offs in terms of implementation, features and performance. The bus architecture is the simplest of the three. With this architecture, all inputs are connected to all outputs through a common bus. Therefore, the bandwidth of this interconnect fabric is exactly the bandwidth of the bus. Hence bus architecture is not sufficient for more than very few Gbps speed ports. The shared memory architecture is similar to the bus architecture except that traffic coming from the inputs is first stored in a central memory before being read to the outputs. In this case, the bandwidth achieved is half the bandwidth of the data bus to the memory since every packet must be written to and read from the memory. The crossbar architecture enables multiple inputs to be connected to multiple outputs simultaneously, as long as different inputs are connected to different outputs. The bandwidth of a crossbar is the sum of the bandwidth of its inputs, thus providing a very suitable solution for building high performance switching fabrics. It is possible to connect several crossbars together in order to achieve even greater bandwidth and a larger numbers of inputs.

A single- staged, non-blocking switch fabric, as shown in figure 1.2, consists of the following three basic components:

1. Queuing Device – It requires either at input or output to hold those cells, which failed in arbitration process. Normally residing on the line card and responsible for hosting the virtual output queues (VOQ) and managing control processes such as flow control. Queue management is necessary to get low packet loss rates and high throughput.
2. Cross point Switch (Switch Fabric) – A matrix cross-connecting element through which packets traverse and deliver to destined output ports.
3. Scheduler – Core element of the system that arbitrates which input port is to be connected to which output port at any time slot, when there are more than one destined for the same output. (i.e. determines which of the N^2 VOQ's are served in each cell time.)

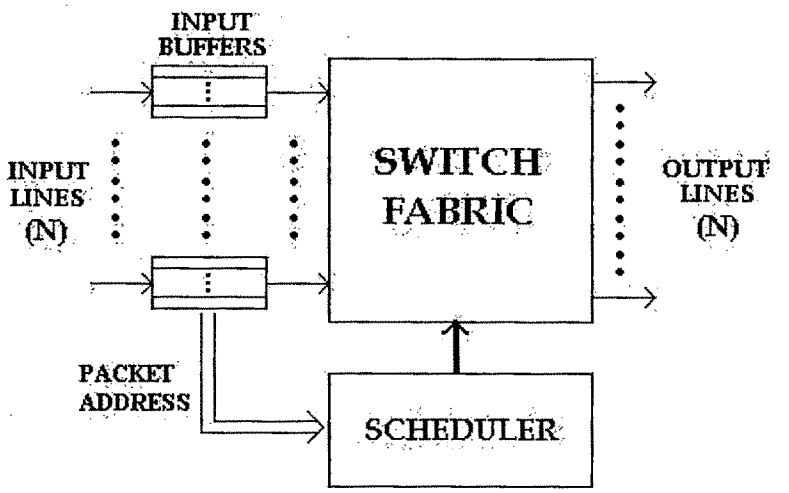


Figure 1.2 General Structure of a Packet Switch with Input Buffers

We assume that the fabric only handles fixed-size data units known as cells. If we relax this assumption, system switches variable-sized packets, by first segmenting variable-sized packets into fixed-size cells at input. Once transferred across the switch, packets are reassembled at output. So segmentation and reassembly (SAR) functionality is required to handle variable-sized packets. Switching fabric decides the efficiency of the network. In interconnection networks, crossbar switch architecture is widely used. Cross-bar switches can store the packet at (1) input (2) output (3) input and output (4) crossbar, depending on that it is respectively classified as (1) Input queuing (2) Output queuing (3) Combined input/output queuing (CIOQ) (4) Cross point crossbar switch. Input queuing has a well-known limitation of low permitted input load, head of line blocking and difficulty in supporting multicasting and broadcasting, while in output queuing full link utilization is used. To achieve a small packet loss rate, output queuing requires fast memory access. In worst case, output queuing requires speedup factor of N for NxN switch. The Knockout switch, tries to overcome the disadvantage of output queuing by using N parallel bus architecture. Disadvantage of Knockout switch is quadratic increase in cross points which can be avoided by using multistage switch like Banyan. Because of regular and repetitive structure, VLSI implementation of Banyan switch is simple. On the other hand, its throughput performance is relatively poor because of its internal blocking and output conflict. Internal blocking problem can be solved by sorting input packets according to the packet destination address by using batcher switch. To eliminate output conflict we can use trap switch.

This thesis considers input queued design for crossbar switching fabric and to remove the HoL problem VOQ is applied. That is, for NxN switch, each input queue is divided into N sub-queues for the N output separately.

1.4. SCHEDULING ALGORITHMS

A key component of any switch fabric is the scheduling mechanism, which governs the functionality of the switch and greatly determines the performance of the system. Therefore it is focus of many researchers.

The scheduling problem in input queued switches is nothing but a bipartite graph-matching problem. Figure 1.3 shows a bipartite graph G with *M* inputs and *N* outputs, together with a matching *W* on the graph. (*M* would be equal to *N* for an *N*×*N* switch with VOQ). Requests from different inputs to various outputs can be seen as edges in bipartite graph. Graph G is bipartite if its nodes are divided into two sets, and each edge has an end in one of the sets. Switch inputs and outputs form the two sets of nodes of the bipartite graph and the edges are the connections required by the queued cells.

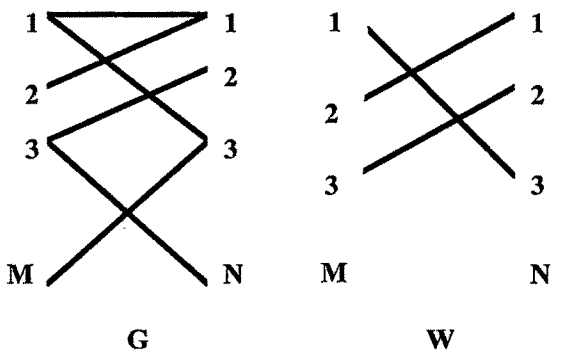


Figure: 1.3 Bipartite graph G and a matching W on it.

The scheduling problem for NxN crossbar switch with VOQ is defined as follows: for each input port i, there are n request lines going to the scheduler: $R_{i,1}$; $R_{i,2}$.... $R_{i,n}$. $R_{i,j} = 1$ indicates that there is a request from input port i to switch a cell to output port j. Correspondingly, there are n grant lines produced by the scheduler: $G_{i,1}$; $G_{i,2}$.....$G_{i,n}$. $G_{i,j} = 1$ means that the scheduler has granted the request from input port i to switch a cell to output port j depending on the scheduling algorithm. To ensure that each input delivers at most one cell into the crossbar fabric, and that each output receives at most one cell, the conditions $\Sigma^{n}_{j=1} G_{i,j} \leq 1$; for $1 \leq i \leq n$ and $\Sigma^{n}_{i=1} G_{i,j} \leq 1$; for $1 \leq j \leq n$ must hold, respectively [51].

The primary challenges and/or important characteristics of designing scheduling algorithms are:

- **High Throughput:** How to maximize the throughput (Efficiency) of the switch? An algorithm that keeps the backlog low in the VOQ's; ideally, the algorithm will sustain an offered load up to 100% on each input and output and has an extremely low cell loss rate.
- **Quality of service (QoS):** QoS support level so that packets with higher priority are treated accordingly.
- **Latency:** How to minimize the latency and jitter experienced by packets while traversing the switch?
- **Robustness:** Level of robustness to different traffic patterns, regardless of its statistical nature.

- **Fairness:** No input queue should remain unserved indefinitely, i.e. there should not be any starvation.
- **Speed:** To achieve the highest bandwidth switch, it is important that the scheduling algorithm does not become the performance bottleneck; the algorithm should be fast enough to find a match as quickly as possible.
- **Implementation simplicity:** Complexity of hardware implementation should be less and it should have minimum state and communication requirement, preferably possible to implement within a single chip.
- **Scalability:** Whether algorithm is scalable to higher-level switch fabric? (i.e suppose it support 4x4 switch fabric but how easily it can be extended to 32x32 switch or higher .)
- **Support:** 1. Cell sequencing is preserved or not?
  2. Multicast functionality supported or not?

1.5 PREVIOUS WORK

Currently available NPU and Cryptographic processors use DES, 3DES and AES algorithms [43].

Jacob J. Repanshek proposed hardware architecture to classify the protocol fields at line speed [83].

Tomas Henriksson et.al proposed a novel way to build a protocol processor for terminal, by using an array of reconfigurable functional pages which are connected in a deep pipeline and supported by microcontroller [38][41][35].

M. Karol et al.'s simulated and compared input and output queuing, on an NxN non-blocking space-division packet switch and concluded that mean queue lengths are always greater for input queuing than output queuing and output queue saturates as the utilization approaches 100%, but input queue saturates at 58.6% utilization [56].

N. McKeown et al.'s solve head of line (H.O.L) blocking problem by introducing separate FIFO queue for each output at each input known as Virtual Output Queue (VOQ). In that the maximum throughput is 100% for independent arrivals. They also proved that for maximum size matching algorithm throughput is 100% when arrivals are uniform. For maximum weight matching algorithm throughput is 100% for both uniform and non-uniform arrivals [67].

N. McKeown introduced new scheduling algorithm for input queue switch known as iSLIP algorithm, which solved synchronization problem in Round Robin Matching (RRM) scheduling algorithm by changing the way of updating the grant pointer and achieves 100% throughput for uniform traffic. ISLIP outperforms RRM and Parallel Iterative Matching (PIM) algorithm which was based on randomness and developed by DEC system research center [65].

Y. Tamir and H.C. Chi. [10] discussed and compared FIFO Arbiter (FIFOA), Statically Optimal Arbiter (SOA), Two Step Arbiter (TSA) and Skewed Two Step Arbiter (STSA).They proposed Wave

Front Arbiter (WFA) and Wrapped Wave Front Arbiter (WWFA). In FIFOA, there is at most one request for each row. Multiple requests for each column are scheduled independently using round robin scheduler. SOA checks all the requests and searches for the scheduling so it maximizes the throughput for the next cycle. If there is more than one configuration with equal throughput, one is selected randomly. In Two Step Arbiter (TSA), row and column wise arbitration is done by keeping highest priority cell (1, 1). Skewed Two Step Arbiter (STSA) rotates the top priority to different rows for the different column arbitration. In WFA, scheduling process begins with top priority cell, instead of top priority row. Top priority cell shifts diagonally from the top left to the bottom right corner. If cell (1, 1) has the top priority and a cell performs it operation in T time units, then total scheduling completes after (2n-1) T time units for nxn switch. WWFA differs from WFA in the mechanism used to indicate the top priority cells when the scheduling begins. WFA uses two circular n bit shift registers, while WWFA uses only a single n bit shift register to indicate the wrapped diagonal of top priority cross points.

J. Hurt et. al, [51] proposed slight variation in WFA and WWFA, known as Rectilinear Propagation Arbiter (RPA) and Diagonal Propagation Arbiter (DPA) by modifying architectures. RPA and DPA avoid the combinational feedback problem and outperform iSLIP and also easy to implement in VLSI due to modulo structure.

M. Keyvani [61] describe the methodology, design and VHDL implementation of high speed, symmetric, input buffer 4 x 4 crossbar switch. He implemented switch for Dynamic Allocated Multi Queue DAMQ buffers in three parts (1) Input port module (2) Cross bar fabric module (3) DPA scheduler using Altera's Flex10KE device and Maxplus II software.

Itamar Elhany [19][20], discussed centralized, maximal scheduling algorithm known as GLIMPSE scheduling algorithm, in which each queue within the VOQs is dynamically assigned a weight based on three parameters: 1.Queue occupancy 2. Average waiting time 3. Prescribe QoS class.

1.6 CONTRIBUTIONS OF THE THESIS

This thesis basically involves comprehensive study, simulation and implementation of the following aspects:

- Architecture study of first generation network system to fourth generation network system (i.e. Network Processor)
- Implementation of two packet processing functions of network processor namely packet encryption and packet classification, to understand and evaluate existing methods. In packet encryption, we implement IDEA cryptographic algorithm because it is one of the strongest cryptographic algorithms and we realize packet classification function by three methods: software (C), packet classification for router using hardware (VHDL) and packet

classification for terminal using hardware software co-design (VHDL) to decide the protocol characteristics and distinguish the specific flows.

- Study simulation and implementation of Space division switches like Knockout, Batcher-Banyan-Trap along with input queued crossbar switches. We simulate 4x4 and 8x8 Knockout and Batcher-Banyan-Trap switches and implement 8x8 Knockout and Batcher-Banyan-Trap switches. We simulate 4x4,8x8,16x16,32x32 crossbar switches along with PIM, RRM, iSLIP, RPA, DPA scheduling algorithms and implement 4x4 and 8x8 crossbar switches with above scheduling algorithms.
- Thesis proposes modification of DPA algorithm [51], named modified DPA (m-DPA) for input queued crossbar switches. It also simulates and implements m-DPA algorithm. Modified DPA rotates priority dynamically, (instead of round robin in DPA) based on queue occupancy in each VOQ, to efficiently utilize the buffers.
- Thesis proposes design, simulation and implementation of Dynamic Scheduling Algorithm (DSA), which is a major contribution of thesis. In DSA algorithm each queue in the Virtual Output Queues is dynamically assigned a weight based on (1) Number of cells in the queue at that time (i.e. Queue occupancy) (2) priorities given by the user (QoS). For each time slot all input ports will compete for the output. Depending on the priority, the input port with the highest priority will block the required output port. While the remaining input ports will continue to compete for the unblocked output port.
- We generate data using four traffic models in MATLAB and apply the same data to all the scheduling algorithms (standard as well as proposed scheduling algorithms) and compare all the scheduling algorithms based on throughput (efficiency), average latency and delay variance.
- We implement space division switches like Knockout, Batcher-Banyan-Trap and Crossbar (with RRM, iSLIP, RPA, DPA, m-DPA and DSA scheduling algorithms) using VHDL in Quartus II and compare VLSI area requirements.

1.7. ORGANIZATION OF THESIS

The thesis is organized in seven chapters as per following details:

Chapter: 2 Introduction and basic architectures of network processors and taxonomy of switching fabrics. It also discusses various methods to integrate switching fabric with network processor.

Chapter: 3 Two packet processing functions are implemented. Packet encryption function of network processor is implemented with VLSI. Packet Classification function of Network Processor is implemented by three methods :Software realization, Hardware realization, and Hardware Software Co-Design realization

Chapter: 4 Discusses four traffic data generation models, which is used as stimuli to different switches. Knockout switch and multistage switch like Batcher Banyan and Batcher Banyan with trap have been simulated for four traffic models. It also discusses VHDL implementation of 8X8 batcher banyan switch, batcher banyan with trap and Knockout switch for different concentrator output in VHDL, using Maxplus II/Quartus II tool of Altera.

Chapter: 5 Reviews scheduling algorithms for input queue switching fabric like PIM, RRM, iSlip, DPA, and RPA. It also Presents MATLAB simulation of above scheduling algorithms for 4x4, 8x8, 16x16, and 32x32 crossbar switches with four different traffic models. It also presents, hardware (VLSI) implementation of above scheduling algorithms, for 4x 4 switch model and an 8x8 switch model.

Chapter: 6 Presents a modification of DPA algorithm (mDPA) and describes MATLAB simulation as well as VHDL implementation of mDPA. It proposes Dynamic Scheduling Algorithm (DSA) and describes MATLAB simulation as well as VHDL implementation of DSA algorithm.

Chapter: 7 Presents comparison of results of various switches including crossbar switches with scheduling algorithms, conclusions, and suggests direction for future research.

1.8. LIST OF PAPERS (PRESENTED/PUBLISHED) BASED ON THIS RESEARCH WORK:

The work described in this project report has been Published /Presented or Accepted for publication as follows:

1. Prof. A.I. Trivedi , Prof S.K. Shah, K. R. Paramar and M.V.Shah " Implementation of cryptographic algorithm as a Network Processor element " Proc. of ninth National

Communication Conference (NCC) 2003, Jointly organized by Telematics Group of all IIT and IISC, January 2003, Chennai, pp 408-412.

2. Mihir V. Shah , Arun B. Nandurbarkar "Review of Various Network Processor Architecture" Presented at National level Seminar SPIN-2003 organised by Gujarat Council on Science and Technology and Goverment Polytechnic, Gandhinagar, 9-01-2003
3. M.V.Shah , S.S.Devar "Memory Cosiderations in Network Processor Architectures" Presented at Emerging Trends organised by IETE, July2003,VADODARA,pp 33
4. Milind S,Shah ,Mihir V.Shah ,and Prof. A.I. Trivedi "Review Of Various Switching Fabric-Network Processor Interface Technology" Presented at Embedded Systems and emerging trends conference- organised by IETE,January 2005, Vadodara, pp 41.
5. Shah M.V., Sharma D.J, Trivedi A.I "Modified DPA Algorithm for High Speed Symmetric Cross Bar Switch" Proc. International Conference on Next Generation Networks, organized by IETE Mumbai Centre, February-2006,Mumbai,cp25.1-cp25.5.
6. Hasmukh Koringa, Mihir V.Shah and Prof.A.I.Trivedi "Matlab simulation and VLSI implementation of ATM knockout switch" *Proc.* National Conference on emerging systems and technologies, June -2006,LDIET,Alwar,pp 9
7. Divyesh R. Keraliya, Mihir V.Shah and Prof.A.I.Trivedi "Implementation of packet classification using TCAM on FPGA" *Proc.* National Conference on emerging systems & technologies, June -2006,LDIET,Alwar,pp 24
8. Mihir V. Shah, Varun Goya, Divyesh Keraliya and Prof. A. I. Trivedi "Implementation and Comparison of different Packet Classification Architectures "*Proc.* National Conference on "Design Techniques for Modern Electronic Devices, VLSI & Communication Systems", May -2007, NIT Hamirpur, pp 349-353,.
9. Mihir V. Shah, Dinesh Sharma, Mehul Patel and Prof. A. I. Trivedi "DSA Algorithm for High speed symmetric crossbar switch" *Proc.* World Congress on Engineering WCE 2007, ICCSE_33, July-2007 ,Imperial college, London.