VLSI SUPPORT FOR VOICE OVER INTERNET PROTOCOL SCHEDULING AND BUFFERING IN HIGH SPEED PACKET SWITCHED NETWORK

Ms.Jyothi Noubade
PhD, Scholar, Dept. Of Electronics, Mewar University,
Email: jyothinoubade12@gmail.com
Dr. Rajendra B. Konda
Department of Electronics, Smt. V.G.College for Women, Gulbarga,
Email: rbkonda@yahoo.com,
Prof.Baswaraj
Department Of Master of computer Applications,CMRIT, Bangalore-37
Email: basawarajnb@gmail.com

----------**ABSTRACT**----------
This paper presents a number of new approaches for designing fast, scalable queuing structures in VLSI for very high speed packet-switched networks. Such queuing structures are necessary for implementing packet buffers in switches and routers that have multi Gigabit-per-second (Gbps) ports. The paper addresses the design of two specific queue architectures: balanced parallel multi-input multi-output (MIMO) buffers and systolic parallel priority queues (PPQ) .A methodology for the systematic design of order-preserving parallel MIMO buffers is presented. The MIMO buffer employs an arithmetic-free systolic routing network and bank of parallel FIFO buffers to yield a load-balanced realization with increased bandwidth. Using this methodology we derived scalable parallel buffer structures that can be designed to match the rate of ultra high-speed links using current memory technology that uses moderate clock rates. A small prototype of the MIMO buffer attains a rate of 10.6 Gbps which is more than adequate to support a Sonet OC-192 link. The combined use of pipe lined architecture and dynamic CMOS circuits resulted in significant reduction in design complexity and substantial performance gains in speed and area.

Keyword:- Very Large Scale Integration, Voice Over Internet Protocol, Parallel Priority Queue, Complementary Metal Oxide Semi Conductor ,Multi Input Multi Output , First In First Out .

I. INTRODUCTION

This research aims to explore numerous modern approaches for designing fast, scalable queuing structures in VLSI for very high speed packet-switched networks. Such queuing structures are necessary for implementing packet buffers in switches and routers that have multi Gigabit-per-second (Gbps) ports, capable of carrying VoIP, in order to improve the QoS. The research addresses the design of two VoIP-specific queue architectures: balanced parallel multi-input multi-output (BPMIMO) buffers, and systolic parallel priority queues (SPPQ).

We describe a method for monitoring Voice over IP (VoIP) applications based upon a reduction of the ITU-T's E-Model to transport level, measurable quantities. The demand for bandwidth is soaring and today's networks lack the ability to scale network back- bone capacities at aggressive rates to accommodate this increased demand for bandwidth. The explosive growth of Internet users, the increased user demand for bandwidth, and the declining cost of technology have all resulted in the emergence of new classes high-speed distributed IP- router architectures with packet-forwarding rates on the order of Terabits per second (Tb/s)2.

This recent and unprecedented growth in bandwidth availability, poses a heavy load on the under- lying infrastructure, and there is an immediate need for backbone switch/routers that can scale to support these demands. There are a broad range of architectures and techniques being proposed, and several enhanced router architectures3 have been introduced to meet the demand of the ever- growing bandwidth requirements of the Internet. Vendors have been able to introduce routers that offer a range of Tables of non-blocking switching capacity", Similarly, current commercial VOIP switches support link speeds reaching up to 10 Gbps (over Sonet OC-192),and experimental switches are supporting link speeds of 40 Gbps [1], [2]. Implementations of these networks require the development of new hardware architectures using advanced VLSI technology that are capable of handling large amounts of traffic with throughputs on the order of Gbps. Moreover, supporting quality-of-service (Quos) in packet networks requires deploying more sophisticated queuing policies in the switching or routing nodes. The first-come first-serve (or FIFO) discipline is no longer adequate in networks that provide service differentiation. Providing QoS guarantees in both cell- and packet-based networks requires the use of a scheduling algorithm in the switches and network

interfaces. These algorithms need to be implemented in hardware in a high-speed network. At minimum, some level of priority scheduling must be supported.

In more advanced applications, buffers with rate adaptation features must be deployed between the switch and external links.

II. Buffer Management for High Speed Links

The ultra high data rates of optical fiber links cannot be matched directly by the microelectronic interfaces and components deployed in today's packet switches. Techniques such as serial-to- parallel conversion and byte staggering (or multiplexing) are normally employed at the optoelectronic interface to bring the "parallel" data rate down so that it can be handled by VLSI circuits running at a typical clock rate of up to few hundred MHz. The process is depicted in Fig. 1.1 .

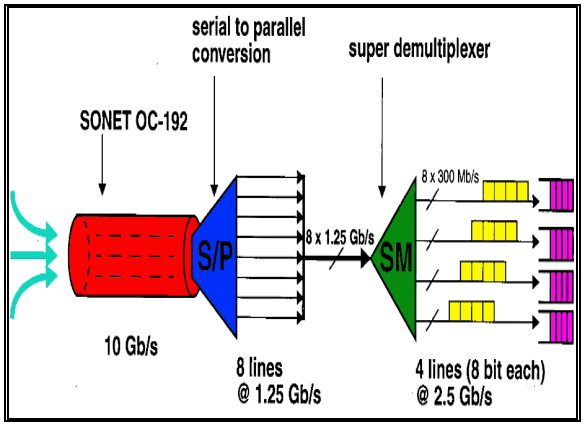

Fig. 1.1: Serial-to-parallel conversion and byte staggering at the op to-electronic interface.

For an OC-192 link that supplies data at a rate of about 10 Gbps to a switch line-card with high-speed input buffers (and byte-wide inputs) which are triggered by 300 MHz clocks, i.e. each buffer has an input rate of 2.5 Gbps. Because low-cost memories are still limited in speed, a 2.5 Gbps buffer will likely be constructed from an interleaved, or a wide data-word, memory structure. Although faster memory technologies can be employed, such as RDRAM [3], this would normally increase the buffer cost significantly and would not be the ideal solution for the large capacity buffers normally associated with high-speed links.

II. PACKET SCHEDULING:

Providing QoS guarantees over packet-switched networks, such as the Internet, relies crucially on the scheduling and buffer management capabilities of the network switches and routers. Packet- switched networks with efficient packet-scheduling mechanisms can support the diverse QoS Requirement of various real time Applications Multimedia sessions may have QoS specifications that include bounded end-to-end delay, bounded delay jitter, bounded cell-loss rates, and guaranteed minimum link bandwidth. Scheduling is required to order packets, for example, to satisfy delay bounds of real-time sessions or to enforce fair bandwidth allocation for sessions sharing a link.

Packet-switched networks exploit a variety of scheduling schemes to provide the QoS guarantees for each connection. Priority-based scheduling is one of the most widely used techniques in packet switches and routers. In this scheme, packets are organized in different priority levels and served according to their priorities. Normally a packet is assigned an attribute which becomes its priority number and it is appended to the packet as an additional field. Packets in a buffer are then ordered and delivered according to their priority value by the scheduler For example, the priority number may be a finish-time computed by a weighted fair queuing (WFQ) algorithm [4], [5], or a class index assigned by a class-based queuing (CBQ) algorithm [6], [7], etc. In any case, accessing the highest-priority packet requires the use of a priority queue structure. By definition a priority queue is a queuing structure for which the desuetude operation always removes the highest (or lowest) priority element in the queue. Arbitrating between a large numbers of packets on a high- speed link requires an efficient hardware implementation of a priority queue. In high-speed net-works, the main challenge is maintaining the correct operation of the priority queue at very high link speeds, e.g. 2.4 to 40 Gbps [1].

The first paragraph under each heading or subheading should be flush left, and subsequent paragraphs should have a five-space indentation. A colon is inserted before an equation is presented, but there is no punctuation following the equation. All equations are numbered and referred to in the text solely by a number enclosed in a round bracket (i.e., (3) reads as "equation 3"). Ensure that any miscellaneous numbering system you use in your paper cannot be confused with a reference [4] or an equation (3) designation.

III. Balanced Parallel MIMO Buffer:

This paper presents a methodology for the systematic design of order-preserving MIMO buffers. Using this methodology we derive scalable parallel FIFO buffer structures that can be designed to match the rate of ultra high-speed links using cur- rent memory technology that uses moderate clock rates. Chapters 3 and 4 develop the design Concept and VLSI implementation for a parallel MIMO buffer structure which can be used in ultra high-speed network interfaces and similar applications. The buffer is capable of inputting and/or outputting multiple packets while maintaining their FIFO (first-in first-out) order. These two chapters focus on buffer design for fixed-size packets, such as VOIP cells (see Appendix A), but the design strategy can be extended to variable-length packets. Some applications of the MIMO buffers

IV. Systolic Parallel Priority Queue

The paper also generalizes the priority queue concept to a parallel priority queue (PPQ) which can be scaled to meet the requirements of ultra high-speed links using standard CMOS technology. Chapters 5 and 6 present an area-

efficient, systolic design of the PPQ for VLSI implementation. The proposed PPQ is rate-adaptive in the sense that the PPQ operates correctly even when the queue input rate and output rate are different [21]. This is an important feature because in practice the output rate of the queue is controlled by the available link bandwidth which may vary (or even become zero) independent of the packet arrival rate. This decoupling of the input and output packet flow rates is a distinguishing feature of our PPQ concept and has not been addressed in previous literature [22].

In many real applications, it is desirable to decouple the input process to the PPQ from the output Process. Specifically, the PPQ outputs may be intermittently blocked from sending cells due to Link congestion. This process is usually called output rate-adaptation, and it requires significant modifications to the conventional architectures. To provide rate-adaptation capability to our PPQ, we strategically employ data steering blocks, between pipeline stages of PPQ, which are con- trolled by a global "go/stop" signal.

In non-systolic architectures, data insertion is normally performed over a global bus connected to the inputs of all the modules in the queue [23], as it will be explained in Chapter 2. This creates a bus loading problem, which adds to the hardware costs (buffers), and decreases the maximum operating speed of the queue clock [24]. In a systolic PQ implementation, the clock can be faster, but technology limitations can limit the clock speed. This problem is more serious when dealing with ultra-fast data links operating at 10 Gbps (e.g. Sonet OC-192) to 40 Gbps (e.g. Sonet OC-768) rate. In this paper we employ a parallel processing approach to solve this scalability problem for Systolic PQs. Our approach embodies a combination of hardware design with more theoretical parallel processing concepts. The combined use of a systolic structure, and dynamic CMOS circuits.

V. VLSI IMPLEMENTATION OF THE PARALLEL MIMO BUFFER

This develops VLSI implementation of the parallel multi-input multi-output (MIMO) buffer structure (presented in Chapter 3) which can be used in ultra high-speed network interfaces and similar applications. The combined use of pipelined architecture and dynamic CMOS circuits resulted in significant reduction in design complexity and substantial performance gains in speed and chip area. We demonstrate our design approach by showing a balanced MIMO buffer design in 0.5-l1m CMOS technology operating at about 330 MHz [8]. At this clock rate, and using an 8-bit wide data-path, we show the design of a 4-input 4-output buffer with a throughput of about10.6 Gbps. Such a buffer is capable of interfacing to a Sonet OC-192 link with 10 Gbps bandwidth. In the next section, we elaborate on VLSI design strategy on the implementation of the buffer. .

VI. VLSI DESIGN STRATEGY

Two different design strategies have been used to implement our MIMO buffer. The first strategy involved developing a synthesizable VHDL model of the buffer. Our VHDL model is pararnetrized in the number of ports of MIMO buffer as well as the data-path width so that buffers of different sizes can be synthesized from the same VHDL code. A portion of the VHDL code that describes an SWT is shown in Fig. 4.1. The VHDL model has been also used for timing and functional verification. However, because our main target was to operate the MIMO buffer at the maximum possible clock rate, our second design strategy was focused on transistor-level circuit development.

```
port  ( clk : in STD_LOGIC; en : in STD_LOGIC; i_in:
in STD_LOGIC;
d_n, d_w:  in STD~LOGIC_VECTOR
(XBAR_IN_WIDTH-1  downto 0);
p_in  : in STD_LOGIC; bar:    out STD_LOGIC; i_out:
out STD_LOGIC;
d_s, d_e: out STD_LOGIC_VECTOR
(XBAR_IN_WIDTH-1 downto 0);
p_out  : out STD_LOGIC);
end XBAR_CELL;
architecture  RTL of XBAR_CELL  is signal checked,
switch:  STD_LOGIC; signal latch_switch  :
STD_LOGIC;
begin
logic : process  (i_in, p_in, checked,  d_n, d_w)
begin
if (i_in = '1') then
Checked <=  '0';
switch <=  '0';
else
if ((p~in = '1') and (checked = '0')) then if ((d_~ (0).=
'0') and (d_w(0) =   '1')) then
swit~h <=  '1';
else
switch <=  '0';
end if;
Checked <=  '1';
end if;
end if;
end process logic;
```

Fig. Part of the VHDL code

VII. THE PPQ RETIMED DESIGN WITH OUTPUT RATE-CONTROL

Output rate-adaptation can be also incorporated in the retimed PPQ using the modified design of Fig. 7.1 for P PQ (m,1). Depending on the value of the INHIBIT signal (0 or 1), the structure of the PPQ is reconfigured using MUXs and DMUXs. For larger size PPQs,

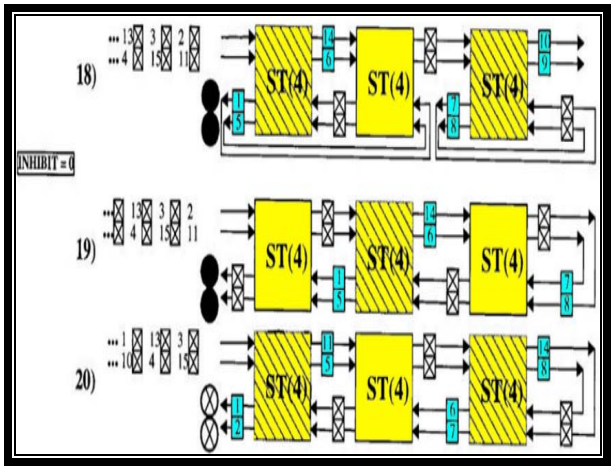

Fig. 7.1 Retimed design of PPQ(m, 1) with inhibit feature

A global INHIBIT signal is employed as before to prevent the PPQ from forwarding data cells through its output. The recursive construction of such a PPQ is illustrated in Fig. 7.2 the dynamic reconfiguration of the retimed PPQ can be illustrated by means of an example. Several steps of the operation of a two inputs, two outputs priority queue of depth 3, i.e. P PQ (2,3)

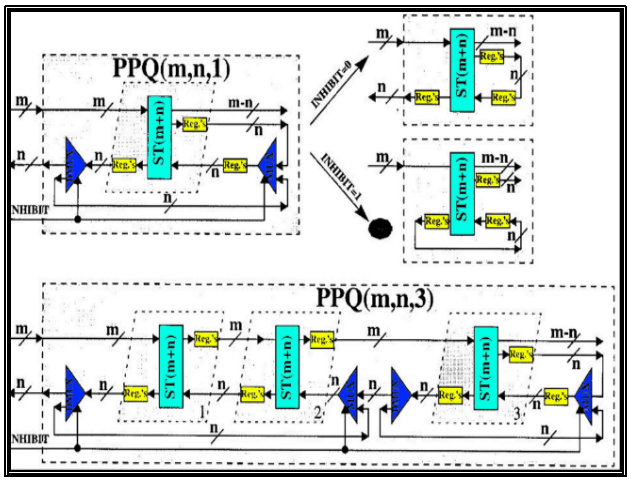

Fig.7.2 PPQS with different depth sizes and inhibit feature

VIII. CONCLUSION

This paper developed a novel rate-adaptation feature to allow different input and output rates for the PPQ [22]. This is an important feature because in practice the output rate of the queue is controlled by the available link bandwidth which may vary (or even become zero) independent of the packet arrival rate. In many real applications, it is desirable to decouple the input process to the PPQ from the output process. Specifically, the PPQ outputs may be intermittently blocked from sending cells due to link congestion. The proposed PPQ in paper is rate-adaptive in the Sense that the PPQ operate correctly even when the queue input rate and output rate are different.

REFERENCES

[1].S. E. Butner, and R. Chivukula, "On the Limits of Electronic VOIP Switching," IEEE Network, vol. 10, no. 6, pp. 26-31, Nov.lDec. 2008.

[2]. S. E. Butner, and D. A. Skirmont, "Architecture and Design of a 40 Gigabit per second VOIP Switch," in Proc. Int'l. Conf. Compo Design, Austin, TX, pp. 352-357, Oct. 2008.

[3]. Richard Crisp, "Direct Rambus Technology: The New Main Memory Standard," IEEE Micro, vol. 17, no. 6, pp. 18-28, Nov.lDec.2007.

[4]. A. Demers, S. Keshav, and S. Shenker, "Design and Analysis of a Fair Queuing Algorithm," ACM Sigcomm '89, Austin, Sep. 1989.

[5].A. K. Parekh, "Generalized Processor Sharing approach to Flow Control in Integrated Services Networks," technical Report LIDS-TH-2089, MIT, Cambridge, MA 02139, Feb.1992.

[6].Dlark S. Shenker, and L. Zhang, "Supporting Real Time Applications in an Integrated Services Packet Network: Architecture and Mechanism", ACM Sigcomm '92, pp. 14-26, Aug. 1992.

[7] S. Floyd, and V. Jacobson, "Link-sharing and Resource Management Models for Packet Networks," IEEE/ACM Trans. on Networking, vol. 3 no. 4, pp. 365-386, Aug. 1995.

[8].M. Kazemi-Nia, and H. Alnuweiri, "VLSI Design of Parallel FIFO Buffers for Gigabit Packet Networks," accepted for publication in the IEEE Trans. on VLSI Systems.

[9]. M. Kazemi-Nia, and H. Alnuweiri, "Parallel Buffer Design for High Speed Interfaces," in proc. CCBR '99, pp. 102-113, Ottawa, Nov. 1999.

[10].Y. S. Yeh, M. G. Hluchyj, and A. S. Acampora, "The knockout switch: A simple architecture for high-performance packet switching," IEEE J. on Select. Areas in Com- mun., vol. 5, no. 8, pp. 1274-1283, Oct. 1987.

[11].H. S. Kim, "Design and Performance of Multinet Switch: A Multistage VOIP Switch Architecture with Partially Shared Buffers," IEEEI ACM Trans. on Networking, vol. 2, no.6, pp. 571-580, Dec. 1994.

[12].H. S. Kim, "Multinet Switch: Multistage VOIP Switch Architecture with Partially Shared Buffers," Infocom '93, pp. 4c.3.1 - 4c.3.8, 1993.

[13].K. Eng, M. J. Karol, and Y. Yeh, "A Growable Packet (VOIP) switch Architecture Design Principles and Applications," IEEE Trans. on Communications, vol. 40, no. 2, pp. 423-430, Feb. 1992.

[14].K. Y. Eng, M. J. Karol, and Y. Yeh, "A High-Performance Prototype 2.5 Gbps VOIP Switch For Broadband Applications," Globecom' 92, pp. 111-117, 1992.

[15].K. Y. Eng, M. J. Karol, and Y. Yeh, "Concentrator-based growable packet switch," U. S.Patent 5-256-958, Oct. 26, 1993.

[16] H.J Chao, and B. Choe, "Design and Analysis of a Large-Scale Multicast Output Buffered VOIP Switch," IEEE/ACM Trans. on Networking, vol. 3, no. 2, pp. 126-138, April 1995.

[17].H. J. Chao, "A Recursive Modular Terabit/Second VOIP Switch," IEEE J. on Selec. Areas in Commun., vol. 9, no, 8, pp. 1161-1172, Oct. 1991.
[18]. K. L. E. Law, and A. Leon-Garcia, "A Large Scalable VOIP Multicast Switch," IEEE J. on Selec. Areas in Commun., vol. 15, no. 5, pp. 844-854 June 1997.
[19].I. Widjaja, and A. Leon-Garcia, "The Helical Switch: A Multipath VOIP Switch Which Preserves Cell Sequence," IEEE Trans. on Communications, vol. 42, no. 8, pp. 2618-2629, Aug. 1994.
[20].A. Pattavina, "Nonblocking Architectures for VOIP Switching," IEEE Communication Magazine, vol. 31, no. 2, pp. 38-48, Feb. 1993.
[21]. M. Kazemi-Nia and H. Alnuweiri, "A Parallel Priority Queue (PPQ) with Rate Adaptation for High Speed Networks," in proc. CCBR '99, pp. 90-101, Ottawa, Nov.1999.
[22] M. Kazemi-Nia, and H. Alnuweiri, "A Systolic Parallel Priority Queue (PPQ) with Output Rate-Control for High Speed Networks," submitted for publication in the IEEE Trans. on VLSI Systems, June 1999.
[23] H. J. Chao, and N. Uzun, "A VLSI Sequencer Chip for VOIP Traffic Shaper and Queue Manager," IEEE Journal of Solid-State Circuits, vol. 27, no, 11, pp. 1634-1643, Nov.1992.
[24] S.-W. Moon, J. Rexford, and K. Shin, "Scalable hardware priority queue architectures for high-speed packet switches," IEEE Real-Time Tech. and Applic. Symp., pp. 203-212, June 1997.
[25.]F. T. Leighton, Introductions to Parallel Algorithms and Architectures: Arrays - Trees - Hybercubes, San Mateo, CA: Morgan Kaufmann, 1992.

Biographies and Photographs

Dr R B Konda received his M.Sc, Ph.D in Applied Electronics from Gulbarga University, Gulbarga in the year 1992 and 2009 respectively. He is working as an Assistant Professor in Electronics in Smt. V.G. Degree College for Women, Gulbarga since 1992. He is an active researcher in the field of Microwave Electronics. He has published more than 25 papers in journals and international conference proceedings.

Ms. Jyothi Noubade, She received her Master of Technology in Very Large Scale Integrated from Visvesarya Technological University, Belgaum, in the year 2007. She is working as a Sr.Software Engineer, Cognizant Technological Solutions Private Limited, since 2007. She is an active researcher in the field of VLSI. She has published more than two papers in journals and international conference proceedings.