A High Speed VLSI Architecture for Scaleable ATM Switches

Paul Shipley
Sherif Sayed
Magdy Bayoumi
The Center for Advanced Computer Studies,
University of Southwestern Louisiana,
Lafayette, Louisiana 70504

Abstract

This paper presents a prototype of a VLSI chip to be used as a building block for an efficiently scaleable ATM switch with a link speed of 622.2 Mb/s. The chip is a 4x4 shared multibuffer ATM switch based on the distributing banyan architecture. It is efficient in storage space like a shared memory switch and scaleable in size like a space division switch. Since the architecture is self-routing, the chip contains all necessary routing control. Special high speed and low power circuitry is used. The chip is implemented in 1.0 micron static CMOS and measures only 25 mm^2 in area.

1. Introduction

The Asynchronous Transfer Mode (ATM) is the leading technology for implementing Broadband Integrated Digital Services Networks (BISDN) and many other general purpose high speed communications networks[1]. ATM's high operating speed requires that the switching be performed in hardware. Currently one of the largest difficulties in implementing ATM systems is the need for an inexpensive, scaleable, efficient ATM switch that has sufficient performance while not losing packets or violating cell sequence. The Distributing Banyan Shared Multibuffer Switch (DBSMS) has been proposed as an efficient and highly scaleable architecture for large ATM switches[2]. The DBSMS architecture has several improvements in both efficiency and scalability over previously proposed architectures[3-9]. This paper presents a prototype chip for the distributing banyan shared multibuffer switching network. This chip is the only building block necessary to build the entire system except for the aligner which precedes the actual switch. The chip is designed to be high speed with a link rate of 622.2 Mb/s per channel with four channels per chip. This data rate is the high end of the ATM standard. While other chips have appeared in the literature with 622.2 Mb/s link rates many of these designs have relied on expensive and power consuming technologies such as ECL and BiCMOS to achieve the necessary speed. By contrast the presented chip uses 1 micron static CMOS so as to facilitate implementation. The chip is compact, measuring approximately 5 mm per side allowing a very low cost of manufacture.

The architecture is highly efficient in buffer space and has a low order of growth. The distributing banyan architecture is superior to previous ATM multistage interconnection networks (MINs) that use switching elements with shared storage because previous architectures of this type have either N^2 growth rates[4,5,6] or have limited bandwidth due to call blocking despite the use of complicated global control [7]. The system is scaleable both in terms of buffering capacity and the number of input channels with the number of chips used varying with both. Furthermore the use of additional distribution layers increases the amount of sharing which lowers cell loss, as illustrated in section 5.

The rest of this paper is organized as follows: section 2 describes the ATM switch architecture that the chip is used to construct, section 3 discusses the architecture of the chip, section 4 discusses the implementation issues of the chip, section 5 presents the simulated performance of the chip and its architecture, and finally section 6 draws the conclusions.

2. System Architecture

The system architecture consists of a set of chips which are interconnected by a distributing banyan network. The term distributing banyan network used to refer to a banyan network which has extra distribution layers added to it to make it non-blocking, figure 1. The banyan network used in this architecture is based on four by four switching elements (SEs) instead of the usual two by two SEs, which allows more sharing of

0-8186-7502-0/96 $5.00 © 1996 IEEE

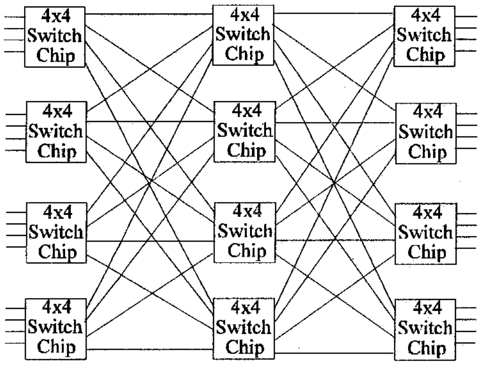

Figure 1. Distributing Banyan Network for a 16 by 16 ATM Switch

buffer space and are thus more efficient than two by two SEs[8]. The distribution layers provide enough extra paths so that the number of paths in the system is equal to the number of channels being routed. They are also used to evenly allocate packets to the next layer of chips and thus increase efficiency by sharing the storage buffers of the chips. This creates the effect of a much larger shared buffer than any of the actual chip buffers.

The switch architecture can be scaled to arbitrary sizes with the required number of chips being given by:

$$\tfrac{N}{S}(E+2\lceil \log_2 N \rceil - 1)$$

where N is the number of channels in the entire switch, S is the number of channels per chip, and E is the number of extra distribution channels added to reduce cell loss. For a 16 channel switch a minimum of 12 chips would be needed and large switches with 256 or even 1,024 channels are possible if multiboard packing is used to accommodate several hundred chips.

In order to determine the order in which cells arrive, it is necessary to associate a time stamp field with every packet that enters the system. This field must have one additional bit beyond the number of bits required to enumerate the maximum possible number of packets that can be in the system at once. The number of required time stamp bits is equal to one more than log base two of the total number of cell buffers in the system. For a sixteen by sixteen switch, ten time stamp bits should be sufficient since fewer than 512 cell buffers are necessary to achieve reasonable cell loss levels.

The interconnection lines between the chips transmits both packet data and control information. Each channel has four data lines that link it with a chip in the next layer of chips. Sixteen data lines are used to lower the speed of the motherboard lines from 622 MHz to 40 MHz. There are also four control lines associated with each channel that indicates to the transmitting chip how many packets the destination chip has in storage.

There are two ways to lower cell loss rates with this architecture, either the size of the shared multibuffer used on each chip can be enlarged or more distribution layers can be added. Enlarging the buffers on each chip can greatly reduce cell loss rates, but increases the area of the chip as additional packet buffers and more complicated control logic is needed. Since the cost of a chip climbs rapidly with the area of the chip, this method can only be used to a certain extent. For simplicity, the prototype chip has 16 packets of storage.

By contrast, adding more distribution layers increases the number of chips required but not the size of each chip. Additional layers cause only a linear increase in the cost of the system. However, the delay experienced by the cells is slightly increased by the additional layers. The physical size of the system and the circuit board complexity would be increased. As a result of these factors the optimum configuration has to be determined by consideration of both the number of packet buffers in each chip and the number of distribution layers.

Each packet entering the system has an extra header associated with it beyond the 53 bytes of the actual packet. These fields consist of the time stamp field, the routing field, and a priority field. The total number of bits required for a sixteen by sixteen switch would be 10 for the timestamp, 4 for the routing field, and perhaps 2 for the priority field which would make a total of 15 bits. This is less than 4% of the 424 bits in the actual packet.

3. Chip Architecture

Each chip contains a four by four shared multibuffer switch. The shared multibuffer switch is comprised of a 4 to 16 crossbar connecting the four input channels to the storage buffers, the N buffers packet-sized storage buffers, a 16 to 4 crossbar connecting the buffers to the output channels, and the associated control circuitry, figure 2.

The chips in the distribution layer send out packets to the next layer according to the following rule: the oldest packet in the chip is sent to the emptiest chip in the next layer. This rule accomplishes two functions. Firstly, it keeps the packets in the proper sequence by reordering the packets at each stage. Secondly, it ensures that the next layer of chips has an even distribution of packets which allows them to act as one large shared buffer. Since shared buffers are much

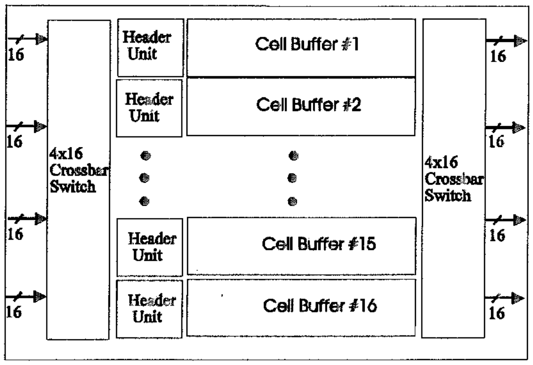

Fig. 2 Diagram of a Shared Multibuffer Switching Node

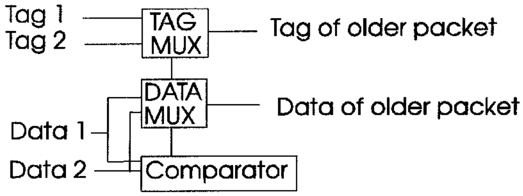

Fig. 3 Comparator Tree Building Block Unit.

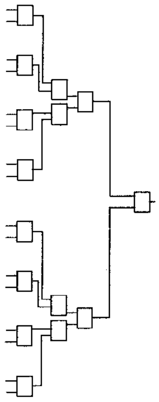

Fig. 4. Comparator Tree For Switch with 16 Buffers

more efficient than unshared buffers this greatly enhances efficiency. The chips in the routing layers must examine all of the packets in its buffers and for each output channel send out the packet with the oldest timestamp. This is necessary to preserve cell sequence. The control logic for cell sequence is done with a comparator tree built out of a magnitude comparator module, figure 3. The comparator tree, figure 4, examines the registers which hold the timestamps for the packets and finds the one with the oldest timestamp. The comparator tree allows the routing to be decided in S clock cycles where S is the number of output channels. The comparator requires only N-1 modules for N cells of storage. For S output channels and N cells of storage space, the amount of time required for this control logic to decide the routing is:

$$k * S * \lceil \log_2 N \rceil$$

where k is the combined delay of an individual comparator and multiplexor. Thus the amount of time required grows linearly with increasing number of channels per chip but logarithmically with increasing cell storage per chip. For the prototype chip the value of N is 16 and S is 4 so the time required is 16*k. Since the cell time is 682 Ns at 622.2 Mb/s, k must be substantially less than 682/16 or 43 Ns. Since this represents approximately 3 nanoseconds per gate this is easily achieved using 1 micron CMOS technology.

The chip's control logic consists of the comparator tree, already discussed, and the input and output control units. The input control unit, figure 5, decides in which of the buffers each incoming packet should be stored. The output control unit, figure 6, uses the comparator tree to determine the address of the packet to be sent out. If the chip is in a distribution layer, then the output control unit uses a much smaller comparator tree to determine which chip in the next layer is the emptiest. This emptiest chip is the destination the packet should be sent to. If the chip is in a routing layer then the control logic only has to check to see if the destination chip has room for additional packets. Both the input and output control units decide the control for one channel per clock period, taking a total of four clock periods. The partial results are stored in working registers until the next cell time arrives when the register's contents are transferred to the shadow register which is connected to the control lines of the crossbar switches. The shadow registers are used so that the control for the next set of packets can be determined without disturbing the routing of the current packets. Since the packet headers are stored while the rest of the packet is still being stored, the control units are able to decide the control logic during the 27 clock cycles required to shift the packets into the chip. This large timing margin would allow the switch to be expanded with more storage and more channels.

4. Implementation Issues

The chip achieves high speed primarily through parallelism. Processing the data streams 16 bits in parallel allows the chip and its inputs to run at 40.3 MHz which is easily achieved in 1.0 micron static CMOS. The intra-chip control is performed largely in parallel to avoid bottlenecks.

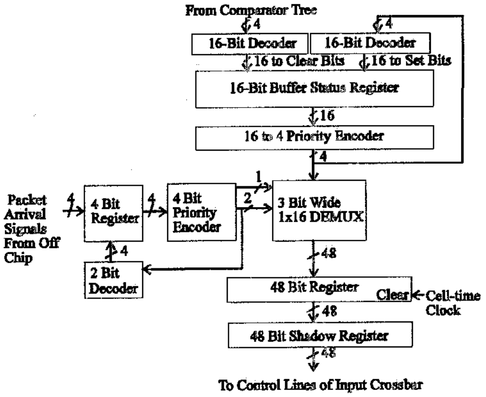

Figure 5. Input Control Unit

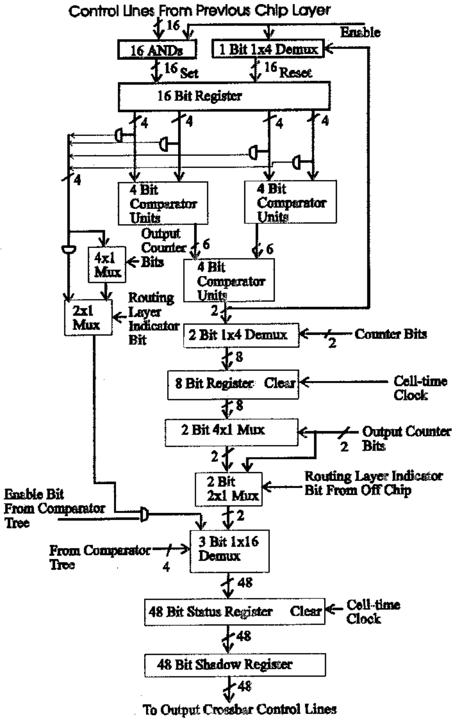

Figure 6. Output Control Unit

There are a number of features in this architecture that make it amenable to low power design. First the architecture works in 16-bit parallel so that the chip can run internally at 40.3 MHz while maintaining a 622.2 MHz line speed. Since the power consumed by CMOS circuitry varies almost linearly with the clock speed this greatly reduces power as compared to a bit serial approach. Also, since each of the multi-buffers are in parallel instead of in series, only a maximum of four cell buffers would need to shift out per cell time, leaving the other buffers idle and consuming almost no power. By shifting in a new packet into a cell buffer

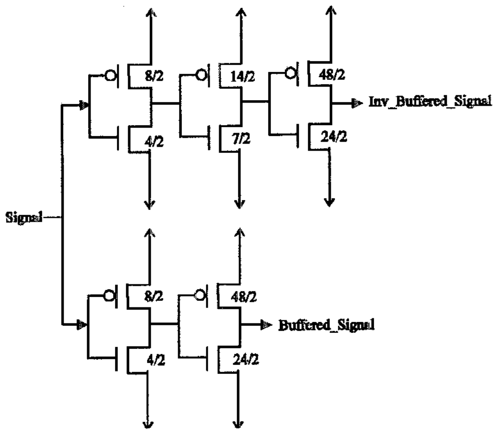

Figure 8. An Amplifier with Equal Delays for the Inverted and Non-Inverted Outputs

that is shifting out an old packet, two packets can be transferred while only shifting one buffer for one cell time.

Low power circuitry is also used. Since the comparator tree comprises the bulk of the control circuitry, the comparator tree is designed with several power conserving features. Comparators which represent empty packet buffers are disabled. Each 10-bit comparator is constructed of three 4-bit comparators. The first 4 bit comparator only enables the second one if the first four bits are equal and the second comparator does the same with the third. This reduces the amount of switching while being much faster than a bitwise comparator. A low power pass transistor XOR gate is used to conserve power in the control circuitry, figure 7. To save some of the power dissipated due to hazards a special buffer was designed which has identical delays for the inverting and noninverting paths, figure 8.

The chip, due to its efficient use of storage space, would use relatively small silicon area. Even using 16-transistor master-slave D flip-flops to store each bit in the cell buffers, the number of transistors required for packet storage would be 112,384 for a chip with 16 cell buffers. The reason shift registers are envisioned for storage is their near-zero access time and high bandwidth. However, high speed static ram could also be used. This would increase the complexity of the control logic slightly but would allow for much greater storage.

The crossbar switches require approximately 9,000 transistors. The control logic requires approximately 6,000 transistors. Since about 112,000 transistors are used for cell storage, the total number of transistors in the chip is therefore approximately 127,000. The number of pins used is about 180 with

128 of these pins being used for the data lines and another 32 used for intra-switch flow control.

Using static RAM instead of shift registers would allow several times the buffer space for the same number of transistors. The problem with using static ram is the need for sufficient speed and the added complexity and it is for these reasons that shift registers would probably be most suitable for a prototype implementation. For a commercial version where greater design resources and more advanced process technologies are available static ram would make the best storage method.

5. Performance Evaluation

The proposed architecture has been tested by computer simulation with very encouraging results. The simulation results indicate that the architecture does maintain cell sequence. They also indicate that the chip enables the architecture to achieve reasonable cell loss rates. The testing regime used was random arrivals at each channel with randomly chosen destination channels for each packet. The simulated switch size is 16 channels.

The test results, as shown in table 1, reveal that three extra layers of chips are needed to lower cell loss below the threshold of testing (10^{-7}). The delay increases by one cell time per extra layer. With three extra layers the total number of chips in the switch would increase to 24.

# of Extra Layers	Cell Loss	Delay
0	1.25xE-4	6.15
1	4.06xE-5	7.15
2	2.34xE-7	8.15
3	<1.0xE-7	9.15

Table 1. Cell Loss and Delay for the System with Different Numbers of Extra Chip Layers

6. Conclusions

The presented chip implements a new architecture that has several major advantages. It is efficient in storage space like a shared memory switch and scaleable in size like a space division switch. The amount of buffering and hence the cell loss can be varied by the addition of extra distribution layers. The architecture is efficient in the number of layers required and thus requires relatively few chips as compared to other space-division switches such as the rerouting banyan network which may require 50 or more layers of switching nodes[9]. Computer simulations have confirmed that the architecture preserves cell sequence and it is efficient in terms of storage space.

The prototype chip is very small with an area of about 25 mm^2 which would allow an extremely low manufacturing cost. Low power features have been incorporated to reduce the power consumption and allow inexpensive plastic packaging. The low cost per chip combined with the low growth rate of the distributing banyan architecture makes the system very cost efficient. Finally, the chip supports 622 Mb/s per channel which is sufficient for even high-end ATM applications.

References:

[1] Jean Yves Le Boudec, "The Asynchronous Transfer Mode: a Tutorial," Computer Networks and ISDN Systems, *The International Journal of Computer and Telecommunications Networking,* vol. 24, no. 4, 1992, pp. 279-309.

[2] Paul Shipley, Magdy Bayoumi, "A Scaleable Multibuffer ATM Switch Architecture", Proceedings of the ISCA International Conference on Parallel and Distributed Computing, 1995, pp. 313-317.

[3] H. Jonathan Chao, Byeong-Soeg Choe, "A Large-Scale Multicast Output Buffered ATM Switch", Globecom 93, pp. 34-41.

[4] A. Jajszczyk and W. Kabacinski, "A Growable Shared-Buffered-Based ATM Switching Fabric", pp. 29-38, Proceedings of Globecom 93.

[5] W.E. Denzel, A.P.J. Engbersen, I. Iliadis, "A Flexible Shared-Buffer for ATM at Gb/s Rates", Computer Networks and ISDN Systems, vol. 27, pp. 611-624, 1995.

[6] T. Kozaki, N. Endo, Y. Sakurai, O. Matsubara, M. Mizukami, K. Asano, "32 x 32 Shared Buffer Type ATM Switch VLSI's for B-ISDN's", IEEE Journal on Selected Areas in Communications, vol. 9, no. 8, pp. 1173-1193, October 1991.

[7] N. Mirfakhraei, "Design of a CMOS Buffered Switch for a Gigabit ATM Switching Network", *IEEE Journal of Solid State Circuits*, vol. 30, no. 1, January 1995.

[8] H. Jonathan Chao, "A Recursive Modular Terabit/Second ATM Switch", *IEEE Transactions on Communications* vol. 42, no. 11, pp 2881-2889, 1991.

[9] Madihally J. Narasimha "The Batcher-Banyan Self--Routing Network: Universality and Simplification", *IEEE Transactions on Communications* vol. 36, no. 10 October 1988.