

Design of a Single-Chip ATM Switching Element

Martin Siegel Ralf Medow Thomas Theimer
Institute of Communications Switching and Data Technics
University of Stuttgart
Seidenstraße 36, D-7000 Stuttgart 1, F.R.G.

Thomas Schwederski
Institute for Microelectronics Stuttgart
Allmandring 30, D-7000 Stuttgart 80, F.R.G.

Abstract

Communication networks based on ATM are expected to satisfy the growing demand for new, high-speed telecommunication services. One of the challenges of the ATM technique is the design of a switch architecture which fulfils the performance requirements and can still be implemented efficiently. This paper describes the design of a switching element for ATM which can be integrated on a single VLSI chip. After summarizing the main characteristics of possible architectures the most promising alternative is identified, taking performance considerations as well as implementation aspects and technological limitations into account. The paper presents a hardware design of that selected architecture. One complex part, the buffer control logic, is discussed in more detail.

Introduction

The rapid technological evolution of the last decade has stimulated the demand for new telecommunication services exceeding the capabilities of today's public networks [21]. This has led to the development of new network architectures supporting various services with widely differing requirements: high speed data communication such as the interconnection of supercomputers and the transport of high quality video signals for medical applications and video conferencing are just two examples [22].

As a basis for future broadband networks, the Asynchronous Transfer Mode (ATM) has now been widely accepted due to its inherent flexibility with respect to the integration of new services with characteristics which may even be unknown today [4]. ATM is a packet oriented transmission and switching principle which employs small, fixed size information units called cells. The characteristic features of ATM are high speed fibre optic links for trunk and subscriber lines, simplified link protocols without sophisticated error correction and flow control mechanisms, and high capacity switching nodes with a total throughput in the range of hundreds of Gb/s.

An ATM switching node must be able to process several millions of cells per second with only a few microseconds delay. This implies a switch architecture where all cell related functions are implemented in hardware, without involving any software control. In recent years, several approaches for ATM switches have been proposed in the literature [1], and a number of prototypes have already been realized to demonstrate the advantages and the feasibility of ATM [3,7,23].

The design of an ATM switch architecture must take performance aspects as well as technological constraints into account. To obtain an economic solution, a reasonable compromise between the required hardware and the resulting performance must be found. This compromise strongly depends on the available technology, which is a limiting factor for parameters like processing speed, chip area and pin count.

Key elements in the architecture of an ATM switch are the internal cell buffers which are required to solve contentions of cells, which are intended for the same resources. The organization and the placement of these buffers have a significant influence on performance, circuit complexity and processing speed. Therefore, a careful selection of the buffering concept is very important for an efficient implementation using a single CMOS VLSI-chip.

In this paper, we describe the design of an ATM switching element under technological constraints. From a larger number of possible solutions, we identify two alternatives which are best suited to fulfil our requirements: a switching element with combined output and input buffers and an architecture with general input buffers where several cells can be read out simultaneously. In a second step, a detailed simulation study is used as a basis for the final selection of the optimum architecture, which is then described in detail including the implementation aspects leading to a realization of the switching element on a single VLSI chip.

2 Switching Element Architectures

The switch fabric of an ATM switching node is usually a multistage network constructed of several elementary switching elements with a typical size of 2 to 32 inlets/outlets. The actual switching function is based on an internal destination address which has to be derived from the information contained in the cell header. Each virtual channel is uniquely identified by a VCI/VPI (Virtual Channel/Path Identifier) routing field, which is used as an index for a translation table to determine the internal routing information and the new VCI/VPI field for the outgoing link. This translation is required at the entrance of the exchange, and in some architectures it is also performed in each stage. Throughout this paper we assume that the translation is *not* a function of the switching elements, and that the internal routing information has already been added to the cells.

Since ATM is based on a statistical multiplexing scheme, several cells may contend for the same output port within a switching element. Consequently, buffers are required to avoid excessive cell loss in case of such collisions. Several possibilities for the placement and the organization of these buffers are known from the literature, and their main characteristics are summarized below.

2.1 Input Buffers

A natural way to solve the output contention is to provide a FIFO queue at each input of the switching element. Thus, an arriving cell can be delayed until the desired output port is available. The advantages of this architecture are the simple buffer control and modest speed requirements, but it is well known that the performance is degraded by the 'head of the line' blocking effect: the cell which is waiting at the head of a FIFO queue is blocking all cells stored later, even if they are directed to idle output ports [9].

Several enhancements have therefore been proposed to overcome this limitation [9,16,17,20,24]. 1. the buffer organization can be changed from FIFO to random access, so that the whole buffer may be scanned by the control logic to find a cell which can be transmitted. This improves the performance, but increases also the complexity of the buffer control. 2. the input buffers may be provided with multiple outlets, so that several cells can be read out and transmitted to different outlets simultaneously. Although this keeps the buffer control simple, the speed of the buffer access has to be increased to support this mechanism. 3. both previous enhancements can be combined in a general input buffer with random access and multiple outlets. This architecture offers an almost optimum performance, but requires complex buffer management as well as internal speed increase.

2.2 Output Buffers

Optimum performance can be achieved by an output buffered switching element, where cells are only queued at the output ports [9]. However, the speed of the buffer access is the most critical part of this architecture. If N is the number of input ports of a switching element, up to N cells may arrive simultaneously which are all destined for the same outlet. Consequently, the size of the switching element is limited by the available memory bandwidth, making an output buffered element difficult to implement using today's CMOS technology.

2.3 Central Memory

To minimize the memory requirements of an output buffered switching element, all buffers may share the same physical memory space. Although the total buffer space can be reduced significantly [15], the speed of the memory access must even be higher compared to physically separate output buffers. This can only be realized using a very high degree of internal parallelization; in extreme cases all bits of a cell must be processed in parallel.

2.4 Crosspoint Buffers

A very simple switching element is obtained by dividing each output queue into N physically separate small buffers, one per input/output pair [12]. Consequently, the speed of the memory access is comparable to a pure FIFO input buffer, but the total memory requirements are increased because there is no sharing between the buffers. This architecture is attractive due to its high degree of modularity, but it is not feasible to integrate larger switching elements on a single VLSI chip.

2.5 Output and Input Buffers

A significant reduction of the speed requirements can also be achieved by limiting the simultaneous access to an output buffer. From a performance point of view it is sufficient to allow only two cells to be written to one output buffer at the same time [18,19]. However, additional input buffers are necessary to avoid internal cell losses. The combination of output and input buffers is a compromise between the complexity of the control logic, the processing speed and the resulting performance.

3 Evaluation of the Designs

The objective is to find a switching element architecture that fulfils all functional requirements and is implementable on a single chip. Functional requirements are a transmission speed of 600 Mb/s and a cell loss rate of less than 10^{-10} per switching element at a load of 80%. The mean value and the standard deviation of the delay should be as small as possible. Optimum performance can be achieved by using output buffers of unlimited length [9,16,17,19]. For this case, formulas for the mean delay vs. load and standard deviation vs. load functions have been derived in [14].

To be implementable on a single chip, the architecture has to meet technological constraints like limited chip area, pin number and processing speed. In this paper, these limits are considered to be very tight to be able to use available technologies for implementation (e.g. sea of gates) and to be prepared for even higher speed requirements, e.g. 2.4 Gb/s trunk lines (Figures 4 and 9 in [6]). One of the consequences of limited processing speed is that memory cannot be accessed at an arbitrary high rate. Furthermore, it will be shown in Section 4 that the major portion of the chip is needed for buffer memory. The chip area limit thus implies that buffers have to be small.

Design decisions to be made are:

- buffer placement and access method
- arbitration strategy
- buffer size
- switching element size.

These are interdependent (e.g. buffer placement influences necessary buffer size, buffer size influences switching element size, switching element size influences severity of access method drawbacks etc.), so that the following does not show a sequential decision process, but is a presentation of results.

Simulations were performed using a program for event driven behavioural simulation at cell level. Simulated networks were a 64-input 3-stage network constructed of 4 input switching elements (later on referred to as "network") and a single 8 input switching element for comparison with the results of [16] (later on referred to as "single switching element"). In both cases a Bernoulli arrival process and random routing (packets are directed equiprobably to all network outputs) are assumed, and conflicts are resolved randomly unless stated otherwise. A backpressure mechanism was used to avoid cell loss within the switch fabric. Thus, a cell can only be transmitted if the buffers in the following stage are able to accept it.

3.1 Buffer Placement and Access Method

Considering the objectives mentioned earlier, some of the possibilities listed in Section 2 can be ruled out immediately:

- FIFO input buffers are undesirable because of their poor performance [9].
- Output buffers are disadvantageous because they require high memory speed. This disadvantage becomes more severe with increasing size of the switching element.
- The central memory architecture is even worse because nearly twice the memory speed as for output buffers is needed.
- Crosspoint buffers can be implemented using low speed memory, but a lot of memory is needed because no buffer sharing at all is possible. Crosspoint buffers thus violate the chip area constraint and are therefore undesirable.

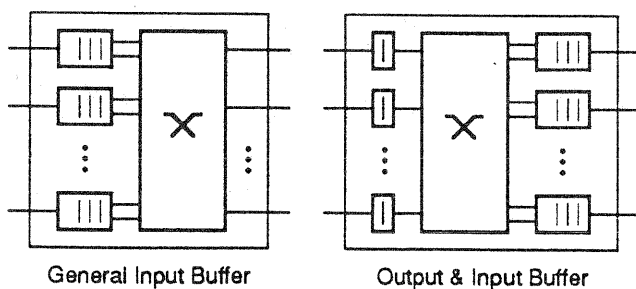


Figure 1: Buffer Architectures

Only the architectures with general input buffers and reduced speed output buffers with additional input buffers are left for consideration (see Figure 1). Both still offer some choices.

With general input buffers the question is how many cells can be taken out simultaneously, and from which places within the buffer they can be taken. According to [20] nearly ideal performance can be reached if up to two cells may be taken out from arbitrary places within the buffer every cell clock cycle. Simulations confirmed these results; reducing the number of cells to be taken out simultaneously to one leads to a severe performance degradation, raising the number of cells to be taken out

simultaneously to 4 yields a marginal performance improvement only.

The basic concept of output buffer combined with input buffers is to reduce the memory speed requirement by limiting the number of cells that may be written into one output buffer simultaneously. A limit of one, which would be the easiest to implement, gives a performance as poor as FIFO input buffers [19]. The optimum limit seems to be 2 according to [18,19]. The input buffers required to hold cells rejected by the output buffers may be as short as one buffer place if backpressure is used. To

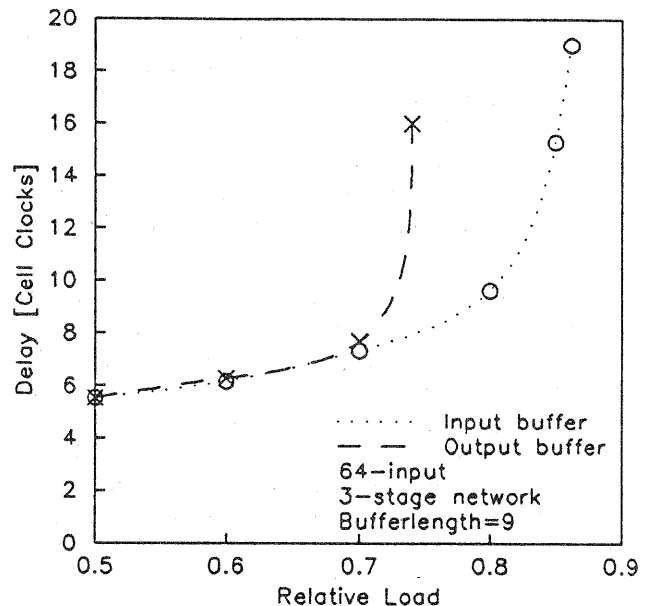


Figure 2: Comparison of Input and Output Buffer.

determine which of the two alternatives should be preferred, networks of both switching element types with various buffer lengths were simulated. Figure 2 shows the results for a total buffer length of 9, which are typical for other buffer lengths as well. It can be seen that the possible throughput when using input buffers is higher than when using output and input buffers. Figure 3 shows the network delay vs. load for input buffers of various lengths. The maximum throughput increases with the buffer length and can even exceed 90% if the buffers are sufficiently long. The simulation results can be summarized as follows: With output and input buffers the buffers need to be larger for a given throughput than with general input buffers. Consequently, given the constraints listed above, general input buffers are superior to a combination of input and output buffers.

3.2 Arbitration Strategies

If several cells within different input buffers are destined for the same output port, only one can be transmitted. Possible arbitration strategies to select the one cell to be transmitted are:

- *Random*: Selects a buffer at random.
- *Cyclic*: Scans all buffers cyclically; to avoid unfairness, the first buffer to be polled also varies in a cyclic fashion.
- *Global Time*: Selects the cell with the longest delay within the switch fabric.

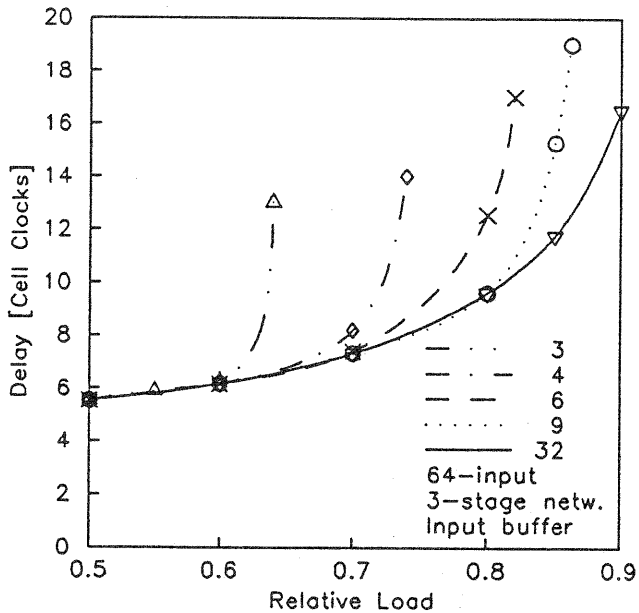


Figure 3: Influence of Buffer Length.

- *Local Time*: Similar to 'Global Time', but based on the delay in the switching element.
- *Longest Queue First*: Selects the buffer with the longest queue.

According to [16], the average delay is hardly influenced by the arbitration strategy, but the standard deviation of the delay is. According to [19], the arbitration strategy influences the required buffer length, but figures are only given for a switching element with FIFO input buffers.

The global time strategy is the best as far as a minimization of the standard deviation is concerned [11], but is rather complicated to implement. A time stamp must be added to every cell, which has to be evaluated whenever a conflict occurs. The complexity makes this strategy unsuitable for implementation. The random and cyclic strategies give about the same standard deviation [19]. It is larger than for all other arbitration strategies [11,16], making both strategies undesirable.

According to [19], the local time strategy and the longest queue first strategy give about the same standard deviation. According to [16] it is close to the ideal behaviour of output buffers. Reference [16], also offers a simple method of implementing the local time strategy without the need to have time stamps on cells. Thus both strategies fulfil the requirements of good performance and easy implementation.

To find out the best of the strategies, a single switching element with large input buffers of the described type and with the arbitration strategies longest queue first, local time and (for comparison purposes) random were simulated. For comparison purposes an equivalent switching element with output buffers was simulated as well. For 90% load Table 1 shows the mean delay, the standard deviation of the delay and the 10^{-4} percentiles of the queue length for a simulation run with $3.6 \cdot 10^6$ cells.

Table 1: Comparison of different arbitration strategies; delays are given in cell clock cycles, confidence intervals refer to 95% confidence level.

	Mean Delay	Std. Dev. of Delay	10^{-4} Perc.
Random	4.18 ± 0.31	6.53 ± 0.43	29
Longest Queue	4.18 ± 0.34	4.55 ± 0.58	10
Local Time	4.23 ± 0.39	4.42 ± 0.29	16
Output Buffer	4.05 ± 0.37	4.13 ± 0.32	34

As predicted by [16] and [19], the mean delay is about the same in all cases. The standard deviation - when using input buffers with the longest queue first or the local time arbitration strategy - is only marginally worse compared to using output buffers. Input buffers with the random arbitration strategy result in the worst standard deviation. The last column of the table is the most interesting one: it shows that by using the longest queue first strategy, queues grow only about half as long as when using the local time strategy and less than one third as long as when using output buffers. Thus, input buffers with longest queue first strategy perform almost as good as output buffers, but the required memory size is drastically reduced.

3.3 Buffer Size

There are two ways to keep the cell loss rate due to buffer overflow low: by making the buffers very large or by introducing a backpressure mechanism that makes sure that no cells are lost within the network. However, even with backpressure the buffers cannot be made arbitrarily small as can be seen in Figure 3. Furthermore, in front of the network a larger buffer has to be provided to make sure cell loss rate at the network entry is small enough.

Simulations of the network show that for the longest queue first arbitration strategy an input buffer length of 7 places is sufficient to achieve a load of 85%. An extrapolation of the simulation results shows that for a cell loss rate of 10^{-10} without backpressure, buffer lengths of 18 places in the first stage and up to 25 places in later stages of the network are necessary.

3.4 Switching Element Size

In this section size refers to the number of input and output ports of a single switching element. With increasing switching element size the chip area needed for buffer memory increases linearly (because every input has its own buffer) and the number of pins increases almost linearly (because most of the pins are associated with input and output ports as will be shown in Section 4). Thus, the switching element size is limited by the available chip area and the number of available pins. Within these limits it should be as large as possible to minimize the number of chips required for a network, to minimize the wiring needed, and to minimize the overall network delay.

For the remainder of this paper a size of 4×4 is assumed. This size was chosen because the design of a

switching element of size 4x4 already shows the problems with larger switching elements, but the design itself is not too area consuming and complicated yet.

4 Implementation Aspects

4.1 Structure and Operation of a Switching Element

In Section 3, a switching element with general input buffers was found the best architecture for implementation. A block diagram of such a switching element is shown in Figure 4. The function of the blocks will be explained below, while the dimensioning of the internal and external data path width (DPW_i and DPW_e) is discussed in the following subsection. The blocks named

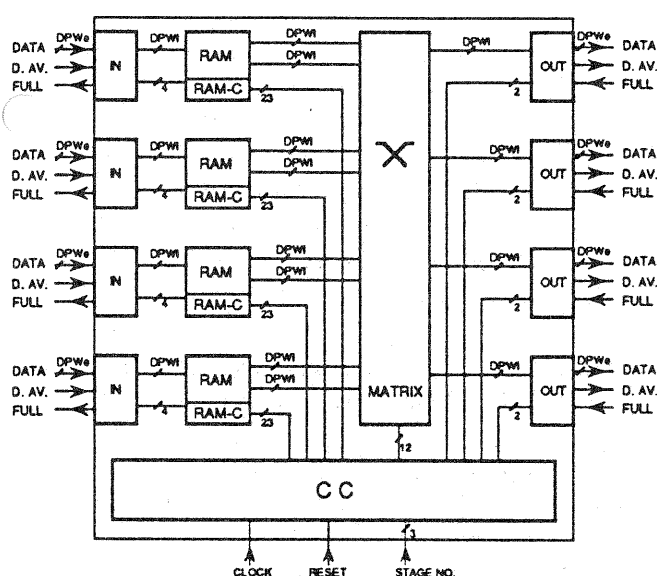


Figure 4: Block Diagram of Switching Element.

IN and OUT contain signal regeneration circuits for input pins and drivers for output pins. If the transmission speed on the links is higher than the processing speed inside the chip ($\Rightarrow DPW_i > DPW_e$), the IN and OUT blocks also perform the necessary serial/parallel and parallel/serial conversions. In addition, the IN block extracts the routing information relevant to the current switching element from the routing tag.

The blocks named RAM are the buffer memories. This RAM has to be triple ported (one write port, two read ports) because it must be possible to place one cell into the buffer and take two cells out of the buffer simultaneously (see Section 3).

The blocks named RAM-C contain the RAM control required to make the buffers act like queues to which cells can be appended and out of which cells can be taken from arbitrary places. For that purpose, the RAM control must keep track of the RAM locations being used, the free locations and the sequence of arrival of those cells currently in RAM. A more detailed description of the function and of one possible implementation is given in Subsection 4.3.

The block named MATRIX is an 8 input 4 output crossbar switch. The block named CC is the central control for the whole switching element. It controls the

internal timing and performs the routing and arbitration functions.

The switching element operates in a pipelined fashion. While cells are being received and transmitted over the input and output ports, the central control determines which of the cells currently in the buffers are to be transmitted to the output ports during the next cell clock cycle. At the end of the current cell clock cycle, the memory control information is updated with those cells to be taken out during the next cell clock cycle. At the beginning of the next cell clock cycle the matrix is set appropriately and the memory control information is updated with the destination of the incoming cells. Now the memory control information reflects the buffer contents at the end of the current cell clock cycle and can thus be used to perform the routing for the following cell clock cycle.

Bearing in mind that the input buffers are logical queues where cells can be appended and taken out from arbitrary places, the routing algorithm can be described as follows:

For all queue places j beginning with the one containing the oldest cell do

For all outputs i do

If output i is available (i.e. the switching element connected to this output can accept cells and no cell has been assigned to this output yet) then

Look for cells destined for output i at place j of all input queues out of which less than two cells have been taken yet

If there are such cells then

If more than one cell has been found then
Select those cell(s) which is (are)
in the longest queue(s)

If there is still more than one cell
(because there are several queues
with the maximum length) then

Select one cell pseudorandomly

Assign the selected cell to output i

4.2 Dimensioning and Implementability

According to [5], an ATM cell is 53 bytes long. However, for the routing tag within the network an additional header of at least two bytes is needed, giving a total cell length of at least 55 bytes. For practical reasons (see below) a cell length of 56 bytes has been selected.

The target technology is CMOS, for reasons see [10]. In this paper the maximum clock frequency is assumed to be 20 MHz. This assumption is rather conservative but makes sure that even circuits containing complex logic (central control, RAM) can be implemented with a currently available process. Future technologies probably will allow higher clock frequencies; some remarks on how to exploit this can be found below.

A transmission speed of 600·56/53 Mb/s and a clock frequency of 20 MHz gives a common data path width ($DPW_i = DPW_e$) of 32 bits and 14 clock cycles per cell clock cycle (14 clocks/cell · 4 bytes/cycle = 56 bytes/cell; this is the reason for selecting a cell size of 56 bytes). With 4 input buffers of 7 places each, this gives a total of 1568 bytes memory on chip. With 34 lines per

port, 8 ports, 5 global control lines (see Figure 4) and at least 2 power supply lines the minimum total pin count is 279 pins.

In the following paragraphs the implementability of the design outlined above using various technologies and layout methods is investigated.

The first approach is to use a currently available sea of gates (e.g., gate forest [2]) technology fabricated in a $2\ \mu\text{m}$ double metal p-well CMOS process. The memory is laid out with the help of a RAM compiler, for everything else a standard cell approach is used. Somewhat optimistic estimates for the required chip area give $111\ \text{mm}^2$ RAM and $61\ \text{mm}^2$ for the remaining logic. Considering that the largest available gate forest master has an area of about $128\ \text{mm}^2$, the switching element would not fit on a single chip. However, it is possible to build a switching element for demonstration and evaluation purposes if cells of only half or one quarter the normal length are used and the data path widths are reduced by the same factor, thus simplifying the data

path and reducing the required memory (i.e. chip area) and pin count considerably. The second approach uses the same design methodology on a gate forest fabricated using a 1 micron process. Because the area of all structures is smaller by a factor of 4 compared to the first approach, the whole switching element will fit onto one chip. However, the switching element cannot be implemented this way, because the required number of 279 pins is too high. A feasible solution is to operate the links between switching elements at a clock frequency of 40 MHz and to reduce the external data path width to $\text{DPW}_e = 16$ bits, resulting in a total pin count of 151. The converters at input and output ports which are necessary to adapt the external to the internal data path width of $\text{DPW}_i = 32$ are the only parts on chip that operate at the higher clock speed. Due to the rather simple design of the converters compared to the other logic on chip, this will not be a problem.

The third approach is a full custom design. This allows for as many as 1 000 000 transistors on a single chip. Triple ported RAM of 1568 bytes needs about 123 000 transistors, all the rest of the switching element about 31 000 transistors. Thus, the entire switching element easily fits onto a single chip as far as area is concerned, and this chip will be much smaller than the possible $100\text{--}150\ \text{mm}^2$. For the pin limitation the same as stated for the second approach holds.

To make optimum use of the available silicon area, there are basically two ways: either increasing the buffer size up to 25 or 30 places per buffer avoid the backpressure mechanism or keeping the buffer length constant and increasing the switching element size. A switching element with 16 inputs and outputs seems feasible. To keep the pin number below 200, the links between switching elements have to operate at 160 MHz ($\Rightarrow \text{DPW}_e = 4$), a speed that is realistic in a full custom design based on modern technology.

4.3 Memory Control

As mentioned earlier, the central control treats the buffers like queues to which cells can be appended and out of which cells can be taken from arbitrary places.

It addresses cells in the buffers by their position within the queue. However, cells are actually stored in RAM. Thus, one task the RAM control has to perform is the translation from relative position within the queue to actual RAM address for cells that are going to be read out of the buffer. (Actually one cell occupies 14 RAM locations. RAM address here refers to the upper part of the actual RAM address, the lower part is provided by the sequencer in the central control.) Another task is to provide the RAM address of a currently unused memory location into which the next incoming cell can be written. The routing algorithm described above needs information about the destination of all cells in the queue and about the queue length, which must also be provided by the RAM control.

An straightforward way to implement a queue is using a shift register. Thus the RAM control is basically a shift register with one stage per queue place (see Figure 5). Every stage consists of 6 flipflops with multiplexers in front, and the necessary control logic. Three of the flipflops hold the RAM address (ADDR) currently associated with the queue place (= stage number), two flipflops contain the destination (DEST) of the cell currently at the queue place (if there is a cell), and one flipflop holds the information whether there is a cell at this queue place at all (VALID). The behaviour of every stage is determined by control lines (CTRL), the position (POS) - both applied to all stages in parallel, the VALID bits of the stage itself and that of the previous stage and the number of the stage.

5 Conclusion

In this paper we have investigated alternative designs for an ATM switching element. In addition to the usual performance criteria, technological constraints - such as chip area, pin count and feasibility of buffer access schemes - have been taken into account. The best compromise for a single-chip CMOS implementation are input buffers from which two arbitrary cells can be taken out simultaneously with the longest queue first arbitration strategy. This solution requires significantly less memory space than output buffers. A backpressure mechanism further reduces the buffer size to only 7 cells per input queue. The size of the switching element is 4×4 . Simulation confirmed that a maximum throughput of 85% can be achieved.

In the latest CCITT recommendation, which describes cell header functions, one bit in the cell header is reserved for an indication of the buffer access priority [5]. A promising mechanism to implement a buffer access priority is partial buffer sharing [13]. The RAM control described above is well suited for partial buffer sharing because the queue length is always available. However, it should be noted that the partial buffer sharing mechanism is only useful without backpressure.

Currently, a demonstrator version of the chip with reduced cell size is under development, whereby a $2\ \mu\text{m}$ double metal semicustom CMOS environment is used. The functionality is stable and fully simulated. There remain only some VLSI specific problems to be solved.

- [13] KRÖNER H.; Comparative Performance Study of Space Priority Mechanisms for ATM Networks", *Proceedings of the IEEE INFOCOM 1990*, San Francisco, pp. 1136-1143.
- [14] KRUSKAL C. P.; SNIR M.; WEISS A.; "The Distribution of Waiting Times in Clocked Multistage Interconnection Networks", *Proceedings of the 1986 International Conference on Parallel Processing 1986*, pp. 12-19.
- [15] KUWAHARA H.; ENDO N.; OGINO M.; KOZAKI T.; "A Shared Buffer Memory Switch for an ATM Exchange", *Proceedings of the International Conference on Communication (ICC89)*, Boston, 1989, pp. 118-122.
- [16] LAMPE D.; "Transfer Delay Deviation of Packets in ATD Switching Matrices and its Effect on Dimensioning a Depacketizer Buffer", *Proceedings of the 9th International Conference on Computer Communication (ICCC88)*, Tel Aviv, 1988, pp. 55-59.
- [17] LUTZ K. A.; "Considerations on ATM Switching Techniques", *International Journal of Digital and Analog Cabled Systems*, Vol. 1, 1988, pp. 237-243.
- [18] OIE Y.; MURATA M.; KUBOTA K.; MIYAHARA H.; "Effect of Speedup in Nonblocking Packet Switch", *Proceedings of International Conference on Communications (ICC89)*, Boston, 1989, pp 410-414.
- [19] RATHGEB E. P.; THEIMER T. H.; HUBER M. N.; "ATM Switches - Basic Architectures and their Performance", *International Journal of Digital & Analog Cabled Systems*, Vol. 23, No. 3, June 1988, pp. 728-735.
- [20] ROTHERMEL K.; SEEGER D.; "Traffic Studies of Switching Networks for Asynchronous Transfer Mode (ATM)", *Proceedings of the 12th International Teletraffic Congress (ITC88)*, Torino, June 1988, Paper 1.3A.5.
- [21] TIMMS S.; "The Dawn of Broadband-The Broadband Phenomenon", *IEEE Network Magazine*, July 1989, pp. 10-15.
- [22] TURNER J. S.; "Design of an Integrated Services Packet Network", *9th Data Communications Symposium ACM Sigcom Computer Communication Review*, Vol. 15, Number 4, Sep. 1985 pp. 124-133.
- [23] HUI J.Y.; ARTHURS E.; "A Broadband Packet Switch for Integrated Transport", *IEEE Journal on Selected Areas in Communications*, Vol. SAC-5, No. 8, October 1987, pp. 1264-1273.
- [24] LEHNERT R.; "On Performance Aspects of Broadband-Switches Using the Asynchronous Transfer Mode Communications Systems", *International Contributions to the Nachrichtentechnisches Kolloquium 1987 at the Technische Universität Braunschweig (FRG)*, Teubner Stuttgart 1988.

Martin Siegel was born in Haslach, F.R.G., in 1959. He received the Dipl.-Ing. degree in Electrical Engineering from the University of Karlsruhe, F.R.G. Since 1987 he is a member of the scientific staff at the Institute of Communications Switching and Data Technics, University of Stuttgart. Since 1988 Siegel also is a permanent visitor scientist at the Institute for Microelectronics Stuttgart. He is currently involved in an European project, where he is responsible for the development of an optimized network management architecture for a highly distributed system. His research interests include VLSI-design, network management and high-speed communication systems, with a special look on efficient implementations and hardware support up to the ISO transport layer.

Ralf Medow was born in Rendsburg, F.R.G., in 1963. He received the Dipl.-Ing. degree in Electrical Engineering from the University of Stuttgart, F.R.G. During the academic year 1987/88 he was at the Georgia Institute of Technology, Atlanta, as a exchange student. He got rewards in several german scientific competitions and got a scholarship of the German National Scholarship Foundation. During his Diploma thesis he did research in the area of ATM switching elements in a cooperation between the Institute of Communications Switching and Data Technics and the Institute for Microelectronics Stuttgart. Medow is currently working on digital control of pneumatic actuators with FESTO.

Thomas H. Theimer was born in 1963 in Germany. He received the Dipl.-Ing. degree in electrical engineering with distinctions from the University of Stuttgart, F.R.G., in 1987. Since then he is a member of the scientific staff at the Institute of Communications Switching and Data Technics at the University of Stuttgart. He works in the field of integrated broadband communication networks, and his research interests include the design and analysis of ATM switch architectures. Currently he is involved in the RACE programme initiated by the Commission of the European Communities. Thomas Theimer is a member of the Communications Society of IEEE.

Thomas Schwederski was born in Datteln, F.R.G., in 1956. He received the Dipl.-Ing. degree in Electrical Engineering from the University of Bochum, F.R.G., the M.S.E.E. 1984 and the Ph.D. degree in 1987, both from Purdue University, U.S.A. From 1984 to 1988 he was working on the design and implementation of the PASM parallel processing system prototype, and on theoretical issues of interconnection network design. Since 1988 Dr. Schwederski is with the Institute for Microelectronics Stuttgart, F.R.G., where he heads the Custom Processor and Test Department. His research interests are VLSI-design, parallel processing, and interconnection networks for parallel computers and ATM switches. Dr. Schwederski is a member of the IEEE, VDE, ITG, Eta Kappa Nu and Phi Kappa Phi.