

# Optimized Architecture and Design of an Output-Queued CMOS Switch Chip

Ronald P. Luijten, François Abel, Mitch Gusat, and Cyriel Minkenberg

IBM Research, Zurich Research Laboratory  
Säumerstrasse 4, CH-8803 Rüschlikon, Switzerland

## Abstract

Traditional improvements in packet switch architecture aimed at increasing switch performance in terms of utilization, fairness and QoS. This paper focuses on improving architecture to achieve implementation feasibility of terabit aggregate data rates while maintaining such performance.

Terabit class shared-memory switch chips are simple in concept but are a challenge to build due to the memory speed requirements and the complexity of wiring needed to connect these memories. Using a property of the combined shared memory and virtual output queuing switch architecture and a property of SRAMs, a new architecture is derived that enables construction of a terabit class switch fabric.

## I. Introduction

In recent years the mainstream switching interest has focused on a number of variations of input-buffered architectures with central control. Although the results are excellent in terms of performance, other architectures also achieve good performance and deserve attention.

This paper focuses on the combined input / output queuing (CIOQ) architecture, which is very scalable owing to distributed control. The traditional output queued switch chip packet buffer is fully shared, which can improve performance [1,2]. The basis for this paper is the shared memory switch fabric [3] enhanced with input adapters employing the virtual output queuing (VOQ) concept [4] as shown in Figure 1.

The switch chip processes fixed-sized packets which are transmitted by an input adapter that sorts the packets per switch output port. Adapter transmission scheduling is performed with a round robin mechanism using grant information provided by the switch. The

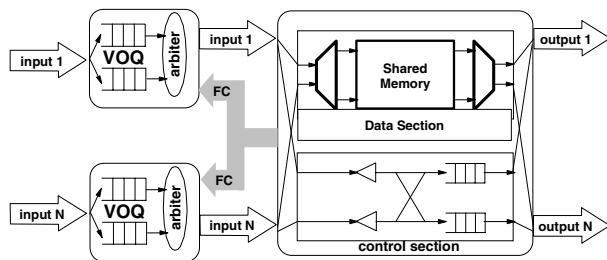
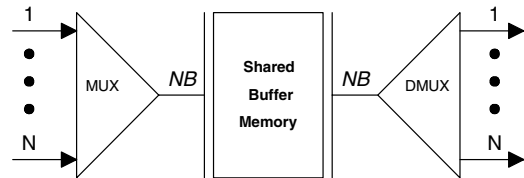


Figure 1: Combined input/output queuing architecture.

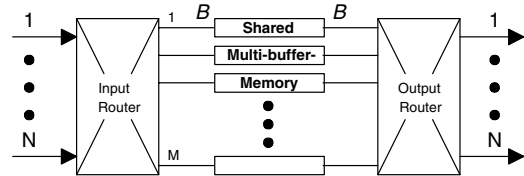
grant-based flow-control mechanism is used to ensure zero packet loss within the fabric. The switch fabric is non-blocking and self-routing and has 32 input and 32 output ports (32x32). When used in speed expansion mode, multiple switch chips can be put in parallel to increase the port speed [3]. This concept allows the memories of each individual switch chip to run at a rate that is a fraction of the external port rate. Speed expansion can be regarded as an implementation of the *parallel packet switch* (PPS) concept described in [5]. When used in port expansion mode, multiple chips can be put together to increase the number of ports [3]. In the literature, this is also referred to as block-cross point queuing: shared distributed buffers [6].

In general for an NxN switch, the shared memory can be regarded as an N-write port and N-read port memory. Because SRAMs do not exist for N>2 at high speeds, another way of constructing the shared memory is required [6,7]. Two common approaches are the use of very wide memory or shared multi-buffer memories as shown in Fig. 2.

Figure 2a shows the implementation of a wide memory approach used by the well-known shared-memory type switch [3,8]. A multiplexer and a demultiplexer are implemented. Given an interface data rate of B bps, the multiplexed data rate is NB at both the input and output sides, thus requiring the shared buffer to sustain an aggregate read and write bandwidth of 2NB. Due to memory cycle time limits imposed by the technology, the bus carrying NB bandwidth becomes very wide. This bus cannot be



(a)



(b)

Figure 2: a) wide memory b) multi-buffer implementation

larger than the packet processed by the switch chip and therefore this approach has limits to its scalability.

Shared multi-buffer switch architectures [9,10,11], as shown in Fig. 2b, store a packet within a single bank rather than being spread among multiple banks as is customary in other interleaved memory implementations. In this approach, a separate connection is provided from any input and any output to every bank, thus removing the need for a multiplexer stage and the fast shared memory. Therefore, the multi-buffer technique ensures a better scalability of the shared buffer throughput with increasing link throughput and port counts for constant packet size. The challenge of a shared multi-buffer approach is the wiring required in the input and output routers shown in Fig. 2b.

The remainder of this paper is organized as follows: Sec. II describes the wiring challenge and study for a shared memory switch chip. In Sec. III a new architecture that solves the wiring problem is derived. In Sec. IV the performance aspects of the new architecture are examined. Sec. V. shows an implementation sizing and Sec. VI presents the conclusions.

## II. Placement and routing study

Table 1 shows the essential features of a shared memory switch chip that is targeted for implementation in a 0.11  $\mu$  CMOS process.

Number of ports	32
Total number of shared memory packet locations	1024 packets
Packet size	64 Byte
Port speed	8 Gbps
Maximum speed expansion factor	4
Technology	0.11 $\mu$ CMOS (Cu-11)

Table 1: targeted switch chip

The shared memory is implemented with the shared multi-buffer approach as proposed by [3]. A placement and routing study is required to demonstrate feasibility of the switch chip given the wiring challenge.

Each of the buffers in this approach holds a fixed-sized packet and has an input bandwidth of 8 Gbps and also 8 Gbps output bandwidth. These 1024 memories are each implemented by a Growable Register Array (GRA) as the memory is too small to be efficiently implemented with an SRAM. A GRA uses multiple standard technology library latches combined into one larger register and is optimized for size. To minimize the wiring, the GRA's are run at the maximum speed of 2nSec cycle time possible for the CMOS technology, requiring only 16-bit-wide buses between the memories and the input / output ports. The magnitude of the wiring problem is illustrated in Fig. 3.

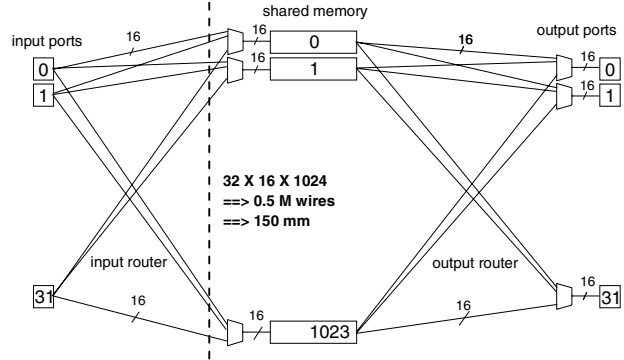


Figure 3: Multi-buffer wiring.

Half a million wires pass the cross section between the input port and the shared memory as a result of 1024 memories, each connecting to 32 input ports through a 16-bit-wide bus. A similar result is obtained at the output router. These wires cross a bisecting surface in front of the memories, shown as a dashed line in Fig. 3. With the 0.11  $\mu$  CMOS metal pitch, using seven levels of metal, of which four can be used for global wiring ( 2 in each direction) this results in a surface length of 150 mm, far beyond a practicable die size when using a straightforward placement.

A full placement and routing job has been performed for the data section of a shared memory switch in the 0.11  $\mu$  CMOS technology, also known as Cu-11 (Table 2). The required silicon area is 87 mm<sup>2</sup>, of which only 25 mm<sup>2</sup> is used by the memories. The placement, illustrated in Fig. 4, shows a regular layout of the GRA's, muxes and decoders, with the logic for the input and output router sprinkled between them. The empty space is required for the wiring (not shown). This result is obtained by careful preplacement of the memories causing a structured meandering of the bisecting surface.

This result was only obtained after a significant effort to rewrite the VHDL and develop new scripts for the placement, the details of which are beyond the scope of this paper. Although the result is encouraging, the lesson learned is that this switch design point is the maximum that is feasible in Cu-11 technology.

While the routing study was in its final stage, new marketing requirements dictated that the port speed

Technology name	SA27e	CU-11
Drawn feature size	0.15 $\mu$ m	0.11 $\mu$ m
L <sub>eff</sub>	0.11 $\mu$ m	0.08 $\mu$ m
Supply voltage	1.8 V	1.5 V
2 input NAND delay	33 nS	27 nS
Metal levels	6	7
Metal 2 - 5 pitch	0.56 $\mu$ m	0.4 $\mu$ m
Design system	Standard cell	
Standard cell size	3.76 $\mu$ m <sup>2</sup>	1.92 $\mu$ m <sup>2</sup>

Table 2: CMOS technology features.

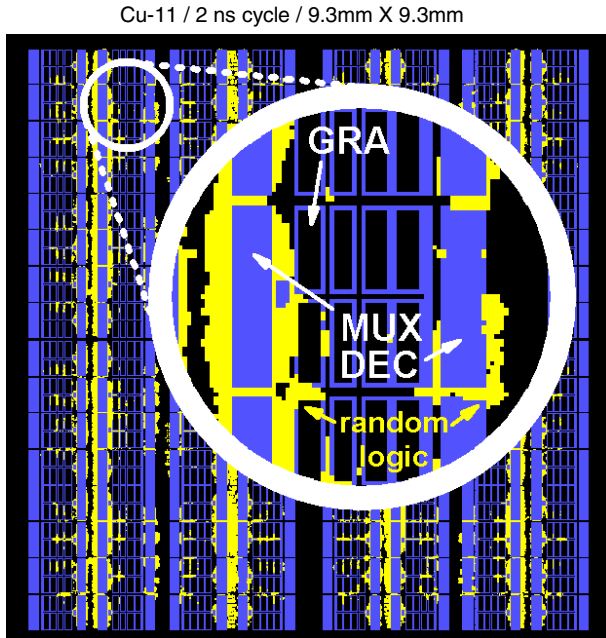


Figure 4: Layout of data section with detail enlargement.

needed to be increased twofold to 16 Gbps. Increasing the chip speed by using a 1 nSec cycle time is not feasible with the Cu-11 technology and increasing the bus width from 16 to 32 bits is not considered feasible based on the results of the routing study.

In order to meet the new challenge, a change is required at the architecture level.

### III. A new architecture

#### III.a Exploiting the CIOQ architecture

The performance of shared memory switch fabrics has been greatly enhanced with the addition of VOQ in the input adapters and has been reported in [4,9,13]. To develop the new architecture we focus on another feature of the VOQ combination with shared memory. Analysis has shown that full sharing of the memory is no longer required [9]. Under bursty traffic conditions sharing of the output buffer can even lead to significant performance degradation due to buffer monopolization [9,14,15,16]. When memory is not fully shared, there is no need for a full connectivity between the memory and in- or output ports. The immediate benefit is the reduced wiring complexity.

To partition the memory three options were studied: (a) use on-chip port expansion, (b) partition memory per output port, and (c) partition memory per input ports.

Port expansion [3] allows construction of a  $kN \times kN$  switch using  $k^2 N \times N$  switches. This is practical for small values of  $k$ . Option (a) is not very attractive in that it does not really reduce the wiring problem significantly. A  $32 \times 32$  fabric is built with four fabrics of  $16 \times 16$ , each

having a memory of one quarter of the  $32 \times 32$  fabric. The wiring problem for the input router has now been reduced from wiring one input router with 1 million wires ( $1024 \text{ memories} \times 32 \text{ inputs} \times 32 \text{ bits}$ ) to wiring four input routers with 128 thousand wires each ( $256 \times 16 \times 32$ ), thus only reducing the complexity by a factor of 2 but introducing many new long wires.

Option (b) eliminates the output router, but multicast is not elegantly solved in that a packet must be replicated before being written into the memory. Furthermore flow control is significantly more complex because each input needs to receive flow control information related to  $N$  different memories. Option (c), shown in Fig. 5, is analyzed in more detail below.

This architecture has the 1024 memories partitioned over the 32 inputs, each input connecting to 32 memories only through a simple redrive network. This scheme has eliminated the input router and thereby has significantly reduced wiring complexity. The flow control mechanism as seen from an input adapter remains unchanged. The switch chip maintains a total count of packets across all partitions for a given output port. A threshold determines the activation of the grant system as in [9]. As the flow control is unchanged and still maintains a global view, the name *virtual shared memory* is proposed for this architecture.

Performance results (see the performance section below) show excellent performance for a variety of traffic. This is mainly attributable to the fact that the memory size is no longer required to be large to achieve performance and the memory is only used for contention resolution. This architecture has each input scheduler at the input adapter make an independent decision of which packet to transmit from its set of virtual output queues depending on the state of these queues and the globally transmitted grant information from the switch fabric. Contention arises when multiple inputs make the decision to send a packet to the same output at the same time. Therefore the memory partition sizing depends on the number of input ports and the latency of the flow control system.

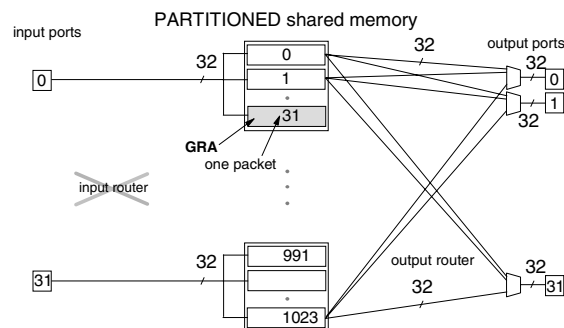


Figure 5: Shared memory partitioned per input port.

Another view of the system is that adapters should move the head of their virtual output queues into the switch fabric but not more than only the head of those queues. As soon as the head of the queue is stored inside the switch fabric the switch has a global view of the traffic requests from all the adapters.

### III.b Exploiting SRAM properties

In the previous section a property of the VOQ / shared memory combination was used to develop a new architecture that enables elimination of the input router. This section uses an SRAM property to eliminate the wiring of the output router.

The fairly low density GRA memories are each replaced with an SRAM that holds 32 packets instead of one. Each SRAM connected to one input port stores the same packet at the same address location, which results in keeping the same total packet storage. This redundant use of memory may at first appear wasteful but the objective is to minimize the total switch fabric die area and not the memory area. The tradeoff is that the freed-up area of the output router is invested in SRAM. The resultant implementation, called *distributed packet routing switch* (DPRS) is shown in Fig. 6. This implementation is also very suitable for multicast.

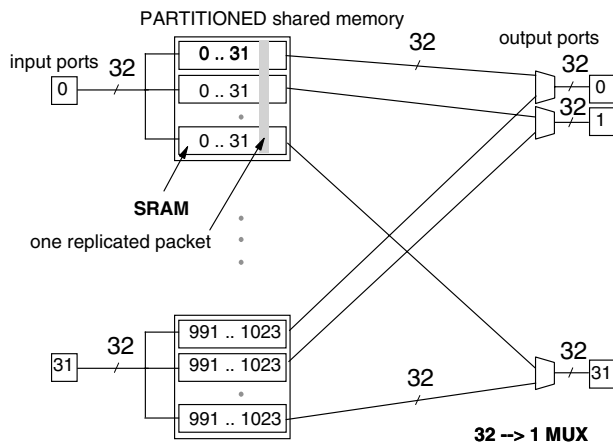


Figure 6: DPRS architecture.

The most important effect of replacing the GRAs with SRAMs is that the output router has largely been moved into the SRAMs. The output router of Fig. 5 is nothing but a very large multiplexer that is built up from AND and OR gates and buffers for driving the long wires. SRAMs not only store information at very high density, but also incorporate a very efficient multiplexer. These large multiplexers are made with bit-lines and sense amplifiers, which is a much denser technology than a standard cell-based approach.

Observe that the architecture has not changed between the switch fabric of Fig. 5 and 6. The only difference is in

implementation and both implementations possess the same performance characteristics.

The resultant output router of Fig. 6 is of low complexity and can be easily wired. In Cu-11 technology, the total SRAM size is around 100 mm<sup>2</sup>, which is only 10% larger than the total data section of Fig. 3, but with twice the bandwidth. An additional benefit is that the place and wire task has become very simple and predictable.

For the 32x32 chip all 1024 memories are written to at each clock cycle under full load. This may lead to excessive power consumption, which may be reduced by using a write enable gating on the SRAMs. A unicast packet only goes to one output, and only the SRAM connected to this output is written to. The buffers to the other SRAMs are still reserved. In case of multicast the SRAMs corresponding to the multicast destination are written to.

Note that this result is closely related to the architecture of a buffered crossbar. The main difference is that with the buffered crossbar a memory management control unit is required at each crosspoint memory, where DPRS needs only one controller per input port. With a 32x32 fabric this requires 32 times fewer controllers. The flow control of this architecture is also much simpler than that of a buffered crossbar as each input adapter only sees one memory. A further optimization could be made by finding a design point in between the two extremes of DPRS and buffered crossbar.

A further observation is that the data section of this switch scales directly with Moore's law as SRAMs (also DRAMs) scale with this law.

Finally, the DPRS implementation features a low pin-to-pin latency when the buffers are empty as the input- and output router pipeline stages have been eliminated. Besides using DPRS in communication switches and routers, this feature enables using DPRS for use also in computer interconnect applications.

## IV. VLSI Implementation sizings

The sizing described in this section is for a complete switch chip: data section, control section and I/O's. The control section is largely similar to a previous design performed in an older technology [9]. Where applicable the VHDL of this design is sized for the targeted 0.11µm CMOS process. The design methodology is a standard-cell approach using compilable SRAMs that are part of the design system library.

The DPRS architecture uses 1024 SRAM memories each storing 32 packets of 64 bytes. In this 32x32 memory arrangement packets are replicated 32-fold. A further optimization is possible where 4 memories of a 2x2 substructure are physically grouped together into one four-port SRAM operating at a cycle time of 2 nSec

and with a data path of 32 bits. This results in a 16×16 memory arrangement only requiring a 16-fold replication of the packets. As a result the total memory size is reduced by a factor of two, which outweighs the SRAM area increase going from a 1-read / 1-write to 2-read / 2-write port memory. Logically this is still operating as a 32×32 switch.

The area required for the 256 four-port memories is 80mm<sup>2</sup> for the 0.11 μm process. Note that no custom SRAMs are used for this sizing and a further area improvement is possible at the expense of custom SRAM design time and cost.

Figure 7 shows the estimated area of the chip in the 0.11 μm process, broken out for packet memory, control logic and I/O macros. For comparison the same chip running at half speed (8 Gbps per port) is sized for the previous 0.15 μm CMOS generation (SA-27e). In this case the memories are also built with four-port SRAMs, bus width remains constant but the cycle time is 4 nSec.

The 0.15 μm sizing result shows that a 32×32 DPRS implementation, even at 8 Gbps port speed, is not feasible as the required die size is larger than 18×18mm<sup>2</sup>, which is not commercially acceptable. Using the 0.11 μm process the 32×32 DPRS implementation running at 16 Gbps per port requires a 15×15 mm<sup>2</sup> die, which is feasible.

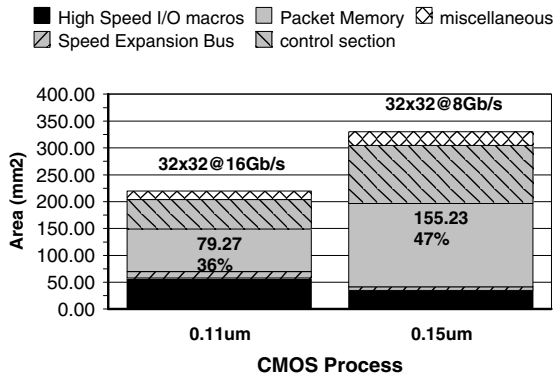


Figure 7: Switch chip sizing estimate.

Using 4 chips in speed expansion, a 32×32 switch fabric is obtained that runs at 64 Gbps per port, sufficient to support OC-768 with switch header overhead and speed escalation, resulting in a fabric with an aggregate line throughput in excess of 1 Terabit per second.

### V. Performance Results

The performance of the traditional fully-shared memory switch element has been studied extensively in [4,9,13]. In this section the effect of the proposed partitioning per input is evaluated by means of simulation and compared to the traditional system. To this end, two different traffic models are employed:

- Correlated, bursty arrivals with geometrically distributed burst sizes. A burst is a sequence of consecutive packets from an input to the same output (Bursty/*B*, *B* = average burst size), and
- "IP"-like traffic, with a burst-size distribution based on the Internet backbone traffic measurements in [17].

A 32×32 system with *M* = 1024 total packet locations is simulated with traffic destinations uniformly distributed over all outputs. The proposed partitioning per input (32 packet locations per input) is compared to the traditional full sharing (every input can access all 1024 locations). In order to prevent output-queue lockout (sometimes also referred to as buffer monopolization or buffer hogging), the output-queue thresholds are always set to  $OQT = M/N$ , with *N* the number of switch outputs. This ensures that no output's contention resolution can interfere with that of any other output. Figure 8 shows delay-throughput curves for Bursty/10, Bursty/30, Bursty/100, and IP traffic. Each data point was obtained by simulating up to 10 runs of 2 million packets each. 95% confidence intervals on the throughput are within 1.25% of the measured value.

From these graphs it is clear that the proposed partitioning of the memory per input has no significant negative impact on delay-throughput characteristics. A small penalty is paid in the form of slightly increased delay throughout the load range compared to the case where the memory is fully shared among all inputs and outputs (*M* = 1024). At high utilization (> 85%) the difference in delay becomes more pronounced, although maximum throughput remains at almost the same level.

The reason that performance with dedicated memories per input is slightly worse than with a memory shared by all inputs is that it can happen that all packets that are destined to a given idle output reside in input queues that are connected to switch inputs that have filled their switch memories completely, thus leading to loss of throughput, even though the output is available. This sort of input blocking does not occur with a fully shared

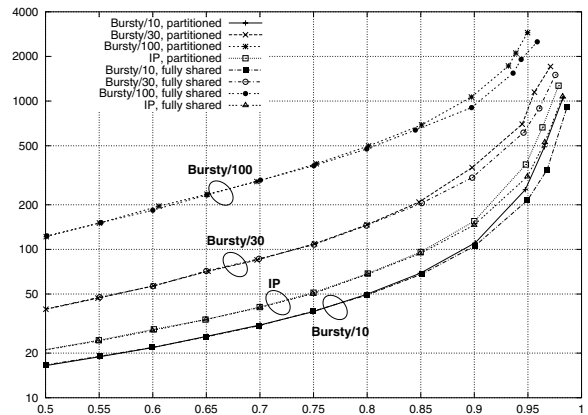


Figure 8: partitioned vs. shared memory performance.

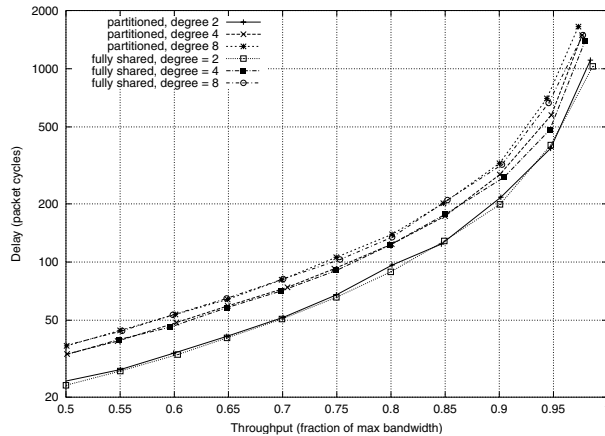


Figure 9: performance for non-uniform traffic.

memory. Especially at high loads with bursty traffic it can occur that a memory is filled completely with packets destined to just a few outputs, blocking newly arriving bursts from accessing the switch. However, the global output-queue grant information will prevent busy outputs from monopolizing the input memories, so that overall throughput performance remains high.

Because real traffic patterns are rarely uniformly distributed also so-called *low-degree* traffic patterns [18] are employed, which are characterized by each input having traffic only for a limited subset of  $k$  outputs ( $k \ll N$ ), and conversely each output receiving traffic from only  $k$  inputs;  $k$  is referred to as the *degree* of the distribution. To save on simulation time, a  $16 \times 16$  system is simulated with  $k$  equal to 2, 4, and 8. Figure 9 shows the corresponding delay-throughput curves with Bursty/30 traffic, comparing the fully-shared memory with the per-input partitioned memory. It is observed that also with non-uniform destination distributions there is very little difference in performance between the proposed partitioned and traditional shared memory switch architectures.

## VI. Conclusions

A new packet switch architecture and implementation has been derived that enables building terabit class switch chips. This new result is based on the property that full sharing of the switch memory is no longer required when combining VOQ with a shared-memory fabric. Furthermore using the built-in multiplexers of SRAMs, complex global chip wiring has been significantly reduced. It is observed that the substantial increase in density in CMOS technology evolution enables this new architecture. A sizing study demonstrates that a  $32 \times 32$  switch chip running at 16 Gbps per port can be built in a  $0.11 \mu\text{m}$  CMOS process. Performance of the new architecture is compared to the previous design point and shows excellent performance for various bursty and IP traffic conditions, with both uniform and non-uniform output port distributions.

The resultant partitioned memory switch architecture features low pin-to-pin latency under low load, also allowing its usage in computer interconnect fabrics in addition to communication switches and routers.

## Acknowledgments

The authors thank the logical and physical design teams of the IBM laboratory in Boeblingen, Germany, for performing the layout study.

## References

- [1] I. Iliadis, "Performance of a packet switch with shared buffer and input queueing," in *Proc. Teletraffic and Datatraffic in a Period of Change, ITC-13*, 1991, pp. 911-916.
- [2] I. Iliadis and W.E. Denzel, "Analysis of packet switches with input and output queueing," *IEEE Trans. Commun.*, vol. 41, no. 5, May 1993, pp. 731-740.
- [3] W.E. Denzel, A.P.J. Engbersen and I. Iliadis, "A flexible shared-buffer switch for ATM at Gb/s rates," *Computer Networks and ISDN Systems*, vol. 27, no. 4, Jan. 1995, pp. 611-624.
- [4] C. Minkenberg, and T. Engbersen, "A combined input- and output-queued packet-switch system based on PRIZMA switch-on-a-chip technology," *IEEE Commun. Mag.*, vol. 38, no. 12, Dec. 2000, pp. 70-77.
- [5] S. Iyer, A. Awadallah, N. McKeown, "Analysis of a packet switch with memories running slower than the line-rate", Computer Systems Laboratory, Stanford University.
- [6] M. Katevenis, P. Vatsolaki and A. Efthymiou, "Pipelined Memory Shared Buffer for VLSI Switches," in *Proc. ACM SIGCOMM '95*, Aug. 1995, pp. 39-48.
- [7] N.K. Sharma, "Review of recent shared memory based ATM switches," *Computer Communications*, vol. 22, 1999, pp. 297-316
- [8] P. Andersson, C. Svensson, C, "A VLSI architecture for an 80 Gb/s ATM switch core", *Innovative Systems in Silicon*, 1996. Proceedings., Eighth Annual IEEE International Conference on, 1996, pp. 9-15
- [9] R. Luijten, A. Engbersen, C. Minkenberg, "Shared Memory Switching + Virtual Output Queueing: a Robust and Scalable Switch", *IEEE International Symposium on Circuits and Systems*, Sydney, Australia, May 2001
- [10] H. Kitamura, "A Study on Shared Buffer-Type ATM Switch", *Electronics and Communications in Japan, Part 1*, vol.73, no. 11, 1990, pp. 58-64.
- [11] H. Yamanaka *et al*, "622 Mb/s 8x8 Shared Multibuffer ATM Switch with Hierarchical Queueing and Multicast Functions", *GLOBECOM '93 Conf. Rec.*, Houston, TX, Nov./Dec. 1993, pp. 1488-1495.
- [12] M.G. Hluchyj and M.J. Karol, "Queueing in high-performance packet switching," *IEEE J. Sel. Areas Commun.*, vol. 6, no. 9, Dec. 1988, pp. 1587-1597.
- [13] C. Minkenberg, T. Engbersen and M. Colmant, "A robust switch architecture for bursty traffic," in *Proc. Int. Zurich Seminar on Broadband Commun. IZS 2000*, Zurich, Switzerland, Feb. 2000, pp. 207-214.
- [14] M. Saleh and M. Atiquzzaman, "Buffer occupancy in ATM switches with single hot-spot," *IEE Electronics Letters*, vol. 31, Jan. 1995, pp. 13-15.
- [15] S. Fong and S. Singh, "Analytical modeling of shared buffer ATM switches with hot-spot pushout under bursty traffic," *Proc. GLOBECOM '96*, 1996, pp. 835-839.
- [16] S. Fong, S. Singh, and M. Atiquzzaman, "An improved buffer sharing scheme for ATM switches under bursty traffic," *Proc. Australasian Comp. Sci. Conf.*, 1996, pp. 26-34.
- [17] K. Thompson, G.J. Miller and R. Wilder, "Wide-area internet traffic patterns and characteristics," *IEEE Network - Magazine of Global Information Exchange*, vol. 11, no. 6, Nov.-Dec. 1997, pp. 10-23.
- [18] M.W. Goudreau, S.G. Kolliopoulos and S.B. Rao, "Scheduling algorithms for input-queued switches: Randomized techniques and experimental evaluation," in *Proc. INFOCOM 2000*, Tel Aviv, Israel, Mar. 2000, vol. 3, pp. 1634-1643.