The Memory Structures of ATLAS I, a High Performance, 16x16 ATM Switch Supporting Backpressure.

Dionisios Pnevmatikatos, Georgios Kornaros, George Kalokerinos, and Chara Xanthaki

Institute of Computer Science (ICS)
Foundation for Research and Technology - Hellas (FORTH)
Science and Technology Park of Crete, Vassilika Vouton,
P.O.Box 1385, Heraklion, Crete, GR 711 10 Greece

{pnevmati, kornaros, george, xanthaki}@ics.forth.gr

Abstract

We present the overall structure of ATLAS I, emphasizing the memory use and requirements. We categorize these requirements in functionality and bandwidth and present the solutions we used in the first implementation of ATLAS I in a 0.35μ CMOS technology. This implementation can serve as a starting point in the design of future switches with functionality similar to ATLAS I.

1. Introduction

ATLAS I is a general-purpose, single-chip ATM switch with advanced architectural features. It is being developed within the ASICCOM[1] project. Figure 1 presents an overview of ATLAS I; it is a highly integrated 16x16 switch, with point-to-point serial links running at 622 Mbits/s each. Using link bundling, ATLAS I can also be configured as 8x8 at 1.25Gbps/link, or 4x4 at 2.5 Gbps/link, etc. The links run ATM on top of IEEE Std. 1355 “HIC/HS” [HIC95] as physical layer, using the BULL “STRINGS” GBaud serial-link transceiver [MCLN93].

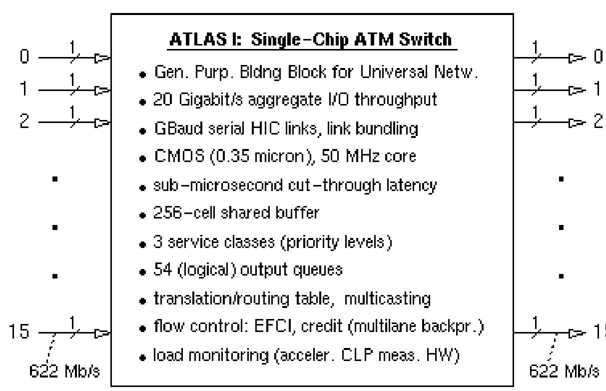

Figure 1: *ATLAS I chip overview*

Internally, ATLAS I operates as a crossbar, with a 256-cell shared buffer and implements three levels of priority, each level having its own queues. Fifty-one output queues (3×16 outputs plus 1 management port) and 3 multicast queues are (logically) maintained in the shared buffer. ATLAS I provides (optional) credit-based flow control (backpressure), using a protocol that resembles QFC [QFC95]. A cell in a backpressured service class (priority level) is never dropped. It can only be transmitted if it acquires both a buffer-pool credit for its outgoing link (indicating that sufficient buffer space exists at the receiver side) and a flow-group credit for its connection. A flow group is merely a set of connections that are flow-controlled together; ATLAS I supports up to 4096 flow groups per link. Credits are transmitted on the links using 1 control and 3 data characters. Our evaluation of the credit-based flow control has shown its superior performance, especially in isolating connections from bursty and hot spot traffic [KaSS98], justifying its implementation cost. These advantages are underlined by other researchers that propose and implement multilane backpressure for ATM switches [QFC95][OzSV95][KuBC94].

The control section of ATLAS I is shared among all links and the entire switch operation is pipelined and functions at the rate of one event per clock cycle. Asynchronous events such as cell arrivals are serialized and handled one at a time. ATLAS I operates in a period of 33 clock cycles: 16+16 to service input and output events and one extra cycle for switch management functions. The clock cycle of the switch core is 20 nanoseconds, and was determined to be fast enough to handle the full bandwidth of the serial-link transceivers. ATLAS I will be fabricated in a 0.35 micron CMOS process, inside a 225 mm^2 die, by SGS Thomson, Crolles, France. More details on the internal organization of the chip and our experiences of the design can be found in [KaSV96], [KKVK97], and [Vats98].

This paper systematically presents the main memory structures of ATLAS I. These structures are more influenced by functionality rather than implementation choices, and can serve as a reference point in the design of

[1] Funded by the European Union ACTS (Advanced Communication Technologies and Services) Programme.

Table 1: *Summary of types of memories required per switch function*

Switch Module	Type of Memory	Memory Functionality
Input & Output Links	Cell elastic buffer / Clock domain isolation	Queues or Dual ported memories
Cell Header Processing	Table for routing & header translation	Single ported memories: 4K×28 & 8K×23bit
Credit Handling	Table of credits	Read-Modify-Write memory, 4K×16bit
Cell Buffering	Cell buffer (256 cells)	27 Banks of 256×16bit single ported memory
Miscellaneous	Queues	Dual ported memories and glue logic
Cell Queuing and Management	Quad ported memory for multiple Queue Head & Tail Pointers	Full-custom implementation: dual ported memory w/ CAM features, quad ported memory, etc

future ATM switches offering functionality similar to ATLAS I. The rest of this paper is organized as follows. In section 2 we describe the ATLAS I memory requirements. In sections 3-5 we present the most important memory structures of the switch: the Cell Buffer, the Credit Table and the Cell Queues, and we conclude in section 6.

2. The Switch Memory Structures

The overall operation of ATLAS I consists mainly of handling events: these events are cell and credit arrivals and completion of cell transmission. On a cell arrival, the cell header is looked-up in the routing and translation table to determine the destination link; then the credit table is consulted to determine whether a credit for that output link and flow group is available. If there is an available credit, then the cell is enqueued in the ready list of the output link, otherwise the cell is enqueued to a *creditless* list. Meanwhile, the cell data are stored in the cell buffer. On credit arrival, the creditless list is checked to identify possible cells that wait for this credit, the first of which will be scheduled for transmission; otherwise the credit is stored in the Credit Table. The completion of a cell transmission is an internal event of ATLAS I, which makes the corresponding output link ready to transmit the next cell. When a link becomes idle, its ready queue is checked and if it is not empty the first cell becomes eligible for transmission.

In the above brief description of the ATLAS I operation, it is evident that several memory structures are consulted during the processing of a cell. Table 1 presents the main memory structures of the switch, their use and implementation. ATLAS I uses queues in three contexts. Across the (different) clock domains of the serial input links and the switch core, queues provide elastic buffering for the incoming cells to absorb the frequency differences of the clocks. Within the switch core, queues decouple the timing of the control with respect to the actual arrivals of the incoming cells. Small queues were implemented using dual ported memories: these compiled memories are fast and with the addition of some semi-custom glue logic provide and easy, fast and reasonably compact queue implementation. Other small structures, such as small register files, were implemented either using FFs, or small compiled memories and are not discussed further. We concentrate on the memories that together account for more than half of the total chip area. These blocks are:

- Routing and Translation: ATLAS I supports routing (switching) based on the VP, VC, or combinations of bits from the VP and VC fields of the header using a routing table of size 8K entries. Each entry specifies a service class and an outgoing link bit mask (to support multicast). ATLAS I also offers a 4K-entry VP/VC translation table; each entry specifies a new 28-bit VP/VC value. The bandwidth requirements from these two tables are relatively small (one access per incoming cell), and the use of compiled static RAM is more than adequate to meet them. Header translation has a significant cost (112Kbits) and is only necessary for large networks; we believe that the cost outweighs the advantages of header translation.
- The Cell Buffer provides space to hold up to 256 cells and the cell size is 55 bytes (53 for the ATM cell plus 2 bytes for the optional routing tag). In ATLAS I the cell buffer integrates the crossbar necessary for the switching functions of the chip.
- Credit-Based flow control is split in two components: (i) the credit table storing the credits for each output link and each flow group, and (ii) the creditless cell list that provides searching capabilities when a credit arrives in the switch.
- Lastly, ATLAS partitions the 256 cells in the cell buffer into fifty-one logical output queues (3×16 outputs plus 1 management port) and 3 multicast queues. Three levels of priority are supported, each level having its own queues. These queues are implemented as linked lists of cell buffer pointers.

The following sections describe in detail each of the Cell Buffer, the Credit Table and the Cell Queue Management blocks of ATLAS and their implementation.

3. The Cell Buffer

The main function of a switch is to buffer and route the cells to the appropriate output link. The cell buffer is therefore a central part of the switch, and must be compact and fast. While the ATLAS I maximum cell size is 55, the

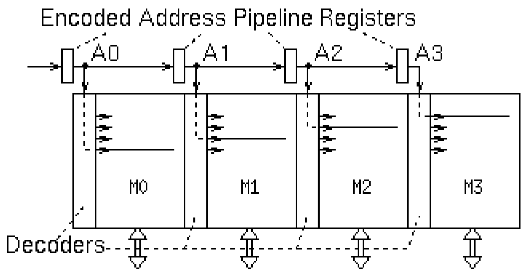

Figure 2: *Pipelined Memory Operation*

cells are transferred to and from the I/O links in quantities of 16 bits, which suggests a banked implementation using 27 16-bit wide banks. The actual implementation is that of a *pipelined memory* [KaVE95]. In a pipelined memory, each bank operates independently. The control signals form a pipeline, reflecting the fact that after the first 16-bit word of a cell is read from the first bank, the next word is read from the exact same location in the second bank, and is sent to the same output link as was in the first cycle. The same property holds for incoming cells that are written in the cell buffer. Figure 2 illustrates the operation of a pipelined memory. In the figure we only show the pipelined control of the memory banks; the control of the crossbar that connects the banks to the output links is also pipelined in a similar manner. A key feature of the pipelined memory is that cell cut-through is achieved without any additional control logic or datapaths. Immediately after the first word of a cell is written in the cell buffer, it can be read for output (if the corresponding output link is idle); the remaining words will be written and read with the same timing, achieving a *cell* cut-through latency of one cycle.

The cell buffer lends itself nicely to a full-custom implementation. Since the addresses of each stage flow in a pipelined fashion, the use of decoders in each stage is not necessary if we latch the already decoded address from the previous stage; a decoder is only necessary for the first stage, saving space. In ATLAS I however, we opted for a semi-custom implementation using compiled memories to minimize the design time, at the expense of increased chip area. Since the control of the ATLAS I switch is pipelined and can initiate only one action per cycle, the cell buffer needs to initiate a single read or write operation per cycle, and therefore single-ported synchronous memory banks are sufficient to cover the bandwidth requirements. The latency of the individual banks is also very short due to their small size: the cycle time under nominal conditions is roughly 5 nanoseconds.

The pipelined memory is by no means the only possible implementation for the cell buffer. An interleaved cell buffer with multiple independent banks is used in many ATM switches such as the Knockout Switch [YeHA87] and the PRIZMA architecture [DeEI95]. Wide memories have also been used in the IBM "Vulcan" switch [Stun94] and in VLSI ATM switches [Koza91][Shob91]. However, due to its simplicity, the pipeline memory achieves the same functionality with a reduced cost in area. An extensive comparison of alternative structures for cell buffering can be found in [KaVE95].

4. The Credit Table

Credit handling is a key feature of the ATLAS I switch. When backpressure is enabled for a link, every time a cell is scheduled for transmission (and therefore an entry in the cell buffer is freed)[2], a credit for the corresponding flow group is sent to the link from which the cell originated. In this way, the sender is notified that a new cell in this flow group can be transmitted. Notice that credits are "generated" when cells are scheduled for transmission, and therefore up to 16 credits may be produced in the 33-cycle period of the switch. While multiple (possibly all 16) of these credits may refer to the same upstream neighbor, in ATLAS I we allow up to two credits to be transmitted every 33 cycles. If more than two credits are generated for a single link, they are queued for later transmission. This queuing is sufficient since at steady state at most one cell is received per 33 per link, creating at most one credit. A short credit turn-around time guarantees that, even with single bit credits, a single connection can fully utilize the link bandwidth.

In a more general case, where the credit range is between 0 and N, a credit cable must perform a "Read-And-Decrement if not zero" for incoming cells, and a "Read-And-Increment" for incoming credits. For ATLAS I with single bit credits, the corresponding operations become: (i) a "Read-And-Clear if not zero" for every incoming cell (up to 16 every 33 cycles) and (ii) "Set" for every incoming credit (up to two credits per output link every 33 cycles). The total number of operations is 48 writes (32 for credits and 16 for the clear part of Read-And-Clear) and 16 reads. Also, consulting the credit table must be done immediately to reduce the latency of a cell inside the switch. As mentioned earlier, the credit table is organized as a 4Kx16-bit structure. Each of the 16 bits corresponds to an output link. Credits set a particular bit in the 16-bit word, while Read-And-Clear operations may read and clear multiple bits to support multicasting.

It is evident from the above discussion, that a simple memory structure cannot meet these requirements; however, the number of operations suggests a dual ported memory. Such a memory can be retrofitted with a custom update-buffer to provide the required read-and-clear functionality. Our first implementation [Vats98] uses 16 1-

[2] For multicast operations the entry if freed and a credit is generated only after the cell is transmitted to *all* the links in its output mask.

bit memories to provide immediate access to the read operations (so as to minimize the cell latency) on one memory port. The corresponding update operations are scheduled in later idle cycles, while the set operations from credit arrivals use the second memory port. The update buffer is searchable since a memory entry is stale when an entry in the update buffer for that location exists. The size of the update buffer is 16 entries since this is the maximum number of cells that may arrive within 33 cycles. This implementation suffers from a limitation of the available compiled memories: the minimum width of these memories is 2 bits while we need a 1-bit wide memory to provide bit-wise control on the set and clear operations. This approach results in a 50% utilization of the area but was used to reduce the design time and simplify the implementation of the ATLAS I prototype.

An alternative approach is to provide true "read-modify-write" support, a solution that can also be generalized for N-bit credits. In that case, the bandwidth requirements are increased because credits too require read-modify-write cycles. The solution is to partition the credit table in smaller pieces and exploit the operation independence to increase memory bandwidth. For each partition, the credit bandwidth is reduced due to the guarantee that at most two credits may arrive per link every 33 cycles. For example, we can partition the credit table in four 8-bit wide pieces, in which case at most 16 credits may arrive at each partition every 33 cycles. Still, all the partitions must be consulted for each incoming cell to handle the possibility of a multicast. As in the previous case, a searchable update buffer is used to buffer operations until memory bandwidth is available. The update buffer now holds credits that must be written to the credit table. For each partition the bandwidth of a 4Kx8-bit wide dual ported memory suffices to handle all the cell and credit traffic. Incoming cells read the memory from one port and update the corresponding entry from the second port in the following cycle. The buffered credits will use the exact same paths for their RMW cycle in the next idle cycle. The required size of the credit buffer is 16 entries, i.e. the maximum number of credits that may arrive before there can be written to the memory. The structure can be further partitioned without increasing the total size of the update buffers: when we split the credit table in N partitions, each partition receives $1/N$ of the total credits, so the combined size of the update buffers remains constant. Further partitioning may be desirable to reduce the power consumption (as each bank is smaller and is used only when a credit for it does indeed arrive) at the expense of increased chip area due to the extra decoders and wiring. The structure of each partition is shown in Figure 3.

5. Cell Queue Management

Cells that are buffered in the cell buffer belong to one of two lists: either to the ready queue for the priority

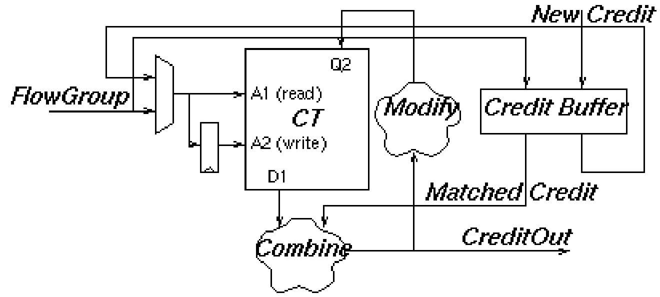

Figure 3: *Credit Table for arbitrary-valued credits*

class/output link combination, or to the creditless cell list. ATLAS I maintains 54 ready queues, implemented via a 54-entry Head/Tail Pointer register file. The Head/Tail Pointer register file may be accessed when a cell arrives (enqueue) or when a cell is scheduled for transmission (dequeue), but also when a credit arrives (enqueue), since a cell waiting for this credit may become ready. Each enqueue or dequeue operations requires two accesses, requiring a quad-ported memory. Such a memory was not available and was laid-out in full-custom VLSI.

The creditless cell list is far more complicated in its operation. As described earlier, the creditless cell list holds cells that are waiting for credits. The structure is shown in Figure 4: when a credit arrives, it must be matched against all cells in the queue to see if there is a match. This matching function is performed in two pipelined steps: first the flow group of the incoming credit is compared to the flow group of all the creditless cells. An exact match means that the flow group corresponding cell(s) is the same. However, the cell may be destined for a different output link than the one the credit came from. In the next cycle the credit input link is compared to the outmask of the cells that matched the flow group. If the input link is part of the outmask, then the corresponding cell is waiting for this credit. Still multiple matches may be found, and a priority enforcer selects the one that will become ready. The following cycle the credit is marked in the Credit Mask memory to record that the credit was received. This memory is required to support multicast in which credits for each destination link arrive independently. The final encoder provides the cell buffer address for the candidate ready cell. More details on the full-custom components of ATLAS I can be found in [KKVK97].

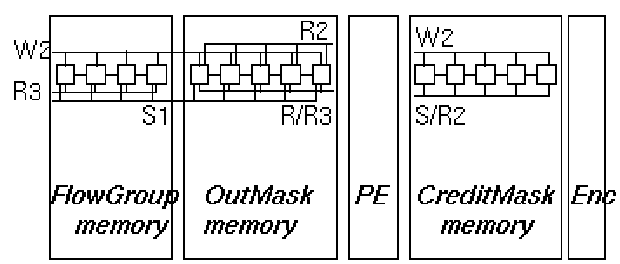

Figure 4: *Creditless Cell List Structure*

6. Conclusions

ATLAS I is an aggressive ATM switch offering large scale integration of 16 bi-directional high-speed serial links along with sophisticated flow control capabilities, large cell buffer space, logical output queuing and support for multiple priority levels, multicasting and load monitoring. More than half of the chip area is consumed by memories. This paper described the functionality and the implementation we chose for these memories in ATLAS I. These memory structures are central to the design of switches with functionality similar to ATLAS I, making the implementations we described in this paper a good starting point for such a design.

References

[DeEI95] W. Denzel, A. Engbersen, I. Iliadis: "A Flexible Shared-Buffer Switch for ATM at Gb/s Rates", *Computer Networks & ISDN Systems 27 (1995)*, pp. 611-624, Elsevier Science B.V.

[HIC95] IEEE Standard 1355-1995, ISO/IEC Standard 14575 DIS: "Standard for Heterogeneous InterConnect (HIC): low-cost, low-latency scalable serial interconnect for parallel system construction," 1995.

[KaSS98] M. Katevenis, D. Serpanos, E. Spyridakis: "Credit-Flow-Controlled ATM for MP Interconnection: the ATLAS I Single Chip ATM Switch," *Proceedings of the* 4th International Symposium on High Performance Computer Architecture, February 1998, pp. 47-56.

[KaSV96] M. Katevenis, D. Serpanos, P. Vatsolaki: "ATLAS I: A General-Purpose, Single-Chip ATM Switch with Credit-Based Flow Control," *Proceedings of the Hot Interconnects IV Symposium*, August 1996, pp.63-73.

[KaVE95] M. Katevenis, P. Vatsolaki, A. Efthimiou: "Pipelined Memory Shared Buffer for VLSI Switches," *Proceedings of the ACM SIGCOMM '95 Conference*, August 1995 pp. 39-48.

[KKVK97] G. Kornaros, C. Kozyrakis, P. Vatsolaki**,** M. Katevenis, "Pipelined Multi-Queue Management in a VLSI ATM Switch Chip with Credit-Based Flow Control," *Proceedings of the 17th Conference on Advanced Research in VLSI (ARVLSI'97),* September 1997, pp. 127-144.

[Koza91] T. Kozaki, N. Endo, Y. Sakurai, O. Matsubara, M. Mizukami, K. Asano: "32x32 Shared Buffer Type ATM Switch VLSI's for B-ISDN's", *IEEE Journal on Selected Areas in Communications*, October 1991, pp.1239-1247.

[KuBC94] H.T. Kung, T. Blackwell, A. Chapman: "Credit-Based Flow Control for ATM Networks: Credit Update Protocol, Adaptive Credit Allocation, and Statistical Multiplexing", *Proceedings of the ACM SIGCOMM '94 Conference*, 31 Aug.-2 Sept., pp. 101-114.

[MCLN93] R. Marbot, A. Cofler, J-C. Lebihan, R. Nezamzadeh: "Integration of Multiple Bidirectional Point-to-Point Serial Links in the Gigabits per Second Range," *Proceedings of the Hot Interconnects I Symposium*, August 1993.

[Shob91] Y. Shobakate, M. Motoyama, E. Shobakate, T. Kamitake, S. Shimizu, M. Noda, K. Sakaue: "One-Chip Scalable 8 * 8 ATM Switch LSI Employing Shared Buffer Architecture", *IEEE Journal on Selected Areas in Communications*, October 1991, pp.1248-1254.

[Stun94] C. Stunkel, D. Shea, B. Abali, M. Denneau, P. Hochschild, D. Joseph, B. Nathanson, M. Tsao, R. Varker: "Architecture and Implementation of Vulcan", *International Parallel Processing Symposium*, April 1994.

[Vats98] P. Vatsolaki, D. Pnevmatikatos, G. Kornaros, C. Xanthaki, G. Kalokerinos, D. Serpanos, M. Katevenis: "Implementation of ATLAS I: a Single-Chip ATM Switch with Backpressure," *Proceedings of the Hot Interconnects VI Symposium*, August 1998.

[YeHA87] Y. Yeh, M. Hluchyj, A. Acampora: "The Knockout Switch: A Simple, Modular Architecture for High-Performance Packet Switching", *IEEE Journal on Selected Areas in Communications*, October 1987, pp.1274-1283.