ATLAS I: A Single-chip ATM switch for NOWs

Manolis G.H. Katevenis Panagiota Vatsolaki
Dimitrios Serpanos Evangelos Markatos

Institute of Computer Science (ICS)
Foundation for Research & Technology – Hellas (FORTH)
P.O.Box 1385, Science and Technology Park,
Heraklio, Crete, GR-711-10 GREECE
markatos@ics.forth.gr
http://www.ics.forth.gr/proj/avg/asiccom.html

Appears in Workshop on Communication and Architectural Support for Network-based Parallel Computing (CANPC 97), San Antonio, Texas, 1997

Abstract. Although ATM (Asynchronous Transfer Mode), is a widely accepted standard for WANs (Wide Area Networks), it has not yet been widely embraced by the NOW community, because (i) most current ATM switches (and interfaces) have high latency, and and (ii) they drop cells when (even short-term) congestion happens. In this paper, we present ATLAS I, a single-chip ATM switch with 20 Gbits/sec aggregate I/O throughput, that was designed to address the above concerns. ATLAS I provides sub-microsecond cut-through latency, and (optional) back-pressure (credit-based) flow control which *never* drops ATM cells. The architecture of ATLAS I has been fully specified and the design of the chip is well under progress. ATLAS I will be fabricated by SGS Thomson, Crolles, France, in 0.5 μm CMOS technology.

1 Introduction

Popular contemporary computing environments are comprised of powerful workstations connected via a high-speed network, giving rise to systems called *workstation clusters* or Networks of Workstations (NOWs) [1]. The availability of such computing and communication power gives rise to new applications like multimedia, high performance scientific computing, real-time applications, engineering design and simulation, and so on. Up to recently, only high performance parallel processors and supercomputers were able to satisfy the computing requirements that these applications need. Although recent networks of workstations have the aggregate computing power needed by these high-end applications, they usually lack the necessary communication capacity. Although there exist several high-speed interconnection networks specifically designed for NOWs, no one of them has clearly dominated (or is likely to dominate) the market yet [2, 15, 16].

Recently, ATM (Asynchronous Transfer Mode), a widely accepted standard for WANs (Wide Area Networks), gains increasing popularity in Local Area

Communications as well. Although, ATM was initially developed for Telecommunications over Wide Area Networks, it can also be efficiently used for communications over LANs, for several reasons, including: (i) ATM provides fixed-size cells that make communication hardware simpler and faster, (ii) ATM cells are small allowing low latency, which is of utmost importance in LANs, and (iii) by using the same ATM equipment both for LANs and WANs, costs will be reduced through mass production.

Although several ATM-based NOWs are in everyday operation around the world, there are several concerns whether ATM is appropriate as an interconnection network for NOWs for the following reasons:

- *Most ATM switches (and interfaces) have high latency today*: In several cases, end-to-end application latency over ATM is similar to the latency observed over more traditional networks, like Ethernet. Thus, most applications that communicate using short messages, will not benefit from such ATM equipment of today significantly.
- *ATM switches drop cells when (even short-term) congestion happens*: ATM was originally developed for voice and image transmission over WANs. In such an environment, it is more important to deliver the information on time (even if it has to be slightly distorted), than to delay the information. For example, if short-term congestion happens during the transmission of live-video image, and some cells are dropped (as a result of the congestion), the human viewers of the video will at most see a short-term distortion in their image, if they notice it at all. However, dropping cells when data (e.g. text) is transferred over LANs is not acceptable. Dropping cells will result in incomplete messages that have to be retransmitted, which increases latency and wastes bandwidth.

In this paper, we present ATLAS I (ATm multi-LAne Switch I), a general-purpose, single-chip gigabit ATM switch, with credit-based flow control and other advanced architectural features. ATLAS I is being developed within the *ASICCOM* (Atm Switch for Integrated Communication, COmputation, and Monitoring) project[1]. ATLAS I was designed to address the above concerns. ATLAS I can be effectively used as a high-speed ATM switch for NOWs because:

- ATLAS I provides cut-through routing, and its latency is well under one microsecond.
- ATLAS I (optionally) provides multi-lane back-pressure (credit-based) flow control: i.e. it *never* drops cells. In back-pressure flow control, an ATM cell is transmitted to the next switch only if there is guaranteed buffer space to store it. Thus, data can be reliably transferred to their destination without wasting throughput and time for retransmissions.

The architecture of ATLAS I has been fully specified and the design of the chip is well under progress. ATLAS I will be fabricated by SGS Thomson, Crolles, France, in 0.5 μm CMOS technology.

[1] the "Gigabit Switching" task project of the European Union *ACTS* (Advanced Communication Technologies and Services) Programme.

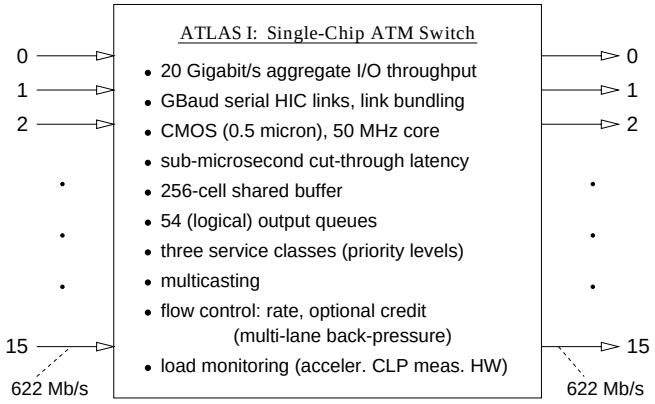

Fig. 1. ATLAS I chip overview

2 Description of ATLAS I

The ATLAS I chip is a building block for high speed ATM networks, for both wide and local areas. ATLAS I based systems can participate in general ATM WANs or LANs as nodes, or they can be used to build complete networks or subnetworks of larger ATM configurations. With appropriate interfaces, they can interoperate with third vendors' equipment that complies to the standards, and they offer the additional advantages of credit flow control when operating within appropriate environments. The ATLAS I chip is a full ATM switch, since it provides all necessary functionality required by the standards, except for management. Although management functions are not provided on-chip, ATLAS I includes the necessary support for a management processor to be either directly attached to it or accessed through the network itself; the resulting configuration provides full ATM switch functionality.

ATLAS I is a single-chip 16× 16 ATM switch, with unidirectional, point-to-point, serial links, running at 622.08 Megabits per second, each.

Figure 1 illustrates the main characteristics and features of this chip, which are explained below.

20 Gbps Aggregate I/O Throughput: the 16 incoming links can handle a sustained throughput of 16 × 622.08 Mb/s, i.e. an aggregate throughput of 9.95 Gigabits/second. The 16 outgoing links, operating in parallel with the incoming links, offer an aggregate output throughput of another 9.95 Gb/s.

Link Bundling: ATLAS I is configurable so that its links can operate either as individual independent links of 622 Mbps each, or as pairs of links operating at 1.25 Gbps (the pair), or as quadruples of links operating at 2.5 Gbps (the quad), or as octets of links operating at 5.0 Gbps (the octet). Furthermore, mixtures of the above configurations, for different links are acceptable.

CMOS Technology: ATLAS I will be implemented in CMOS technology which offers the advantages of high density and lower cost (compared to BiCMOS or GaAs). The high speed (serial) links will operate in CMOS, using BULL's "STRINGS" technology [25].

Low Latency: ATLAS I provides cut-through routing – the head of a cell can depart from the switch before its tail has entered the switch. Short latency is of crucial importance for several distributed and parallel computer applications running on NOWs, as well as in building switching fabrics that offer low overall delay. The ATLAS I cut-through latency is considerably shorter than one microsecond. This represents more than an order of magnitude improvement relative to today's ATM switch boxes. No valid comparison can be made to the latency of unbuffered crosspoint chips that are used to build switching fabrics, since, unlike ATLAS I, these chips are not complete ATM switches.

Cell Buffering, Shared Queueing: ATLAS I is not merely a crosspoint switching matrix -it contains much more than that, starting with an on-chip cell buffer for 256 ATM cells (110 Kbits). Within this shared space, 54 (logical) output queues are maintained; output queueing offers the best switching performance. Moreover, for non-backpressured traffic, memory space is shared among all lines, which is the queueing structure that gives the best utilization of the available buffer space.

Three Priority Levels / Service Classes: ATLAS I supports, via distinct queues, three service classes, at a different level of priority each. These distinct logical queues are also organized *per output*, thus eliminating head-of-line blocking; they all share the same physical buffer space – the shared cell buffer. The top two priorities are intended for policed traffic, while the low priority is intended for non-policed (flooding) traffic. The top priority is non-back-pressured, while the two bottom ones are (optionally) back-pressured – see discussion on flow control, below. The top priority class is designed for voice and other similar real-time traffic, where dropping cells is preferable over delaying cells during congestion periods. The middle-priority class is intended for policed data and other similar traffic, where we wish to offer certain performance guarantees to the user, and we also wish that cells are not dropped. The lowest priority class is appropriate for best-effort data.

Multicasting: Any entry in the translation table of ATLAS I can specify a mask of output links to which a corresponding incoming cell will be multicast. The VP and VC numbers of all outgoing cells, for a given incoming cell, are all identical to each other.

Flow Control: ATLAS I features advanced flow control capabilities. Besides ATM Forum standard EFCI, *optional* credit flow control is additionally provided.

Multi-lane (VP/VC-level) credit (back-pressure) flow control, like what ATLAS I offers, has many, important advantages over rate flow control – the current ATM Forum standard. Low priority traffic always "fills in" the available throughput, and cells are never dropped, resulting in full utilization of the (expensive) link capacity; transmission can start right away at top speed, being automatically throttled down to the correct level, and retransmissions are not needed, resulting in the minimum possible transmission time for a given message.

IEEE Standard 1355 "HIC" Links: The ATLAS I links run ATM on top of IEEE Std. 1355 HIC/HS as the physical layer. Although ATM today is most frequently run on top of SDH/SONET (STM1/STM4), ATM is a network-level protocol that is independent of physical layer; ATM has already been run on top of at least UTP-3, T1/T3, E1/E3, STM1/STM4, and FDDI-physical. HIC (IEEE 1355) is our link protocol of choice, because ATLAS I is optimized for short-distance links, as found inside large ATM switch boxes for WANs and in local and desktop area networks. Although HIC needs a higher signaling (Baud) rate than SDH/SONET for a given effective (data) bit rate, HIC is much simpler than SDH/SONET to implement in hardware; in fact, 16 SDH/SONET interface circuits would not fit on the single chip of ATLAS I. Second, all cells on an SDH/SONET link must be aligned on cell-time boundaries (idle periods last for an integer number of cell times), whereas cells on a HIC link can have arbitrary byte-aligned starting times. This reduces the average (cut-through) latency of the switch (under light load) by half a cell time (340 ns), which is important in local area applications. Third, for short distance links and correspondingly small buffer spaces, the transmission of a flow control credit cannot cost as much as an entire cell, neither can it be delayed until it gets bundled with other credits inside a cell; SDH/SONET leaves no room for credit transmission outside cells, while HIC/HS allows credits to be encoded in a straightforward way, on arbitrary byte-aligned times, using dedicated control character(s). To yield a net SONET/OC-12 rate of 622.08 Mb/s (1.4128 Mcells/s), HIC/HS uses a signaling rate between 0.91 and 1.05 GBaud (1.4128 Mcells/s $\times$ 54 to 62 characters/cell $\times$ 12 b/character), depending on whether single-lane and/or multi-lane back-pressure is enabled. This high data rate requirement results mainly from HIC/HS's encoding of each byte using 12 bits. The advantage of this encoding, however, is that no external crystals and no internal PLLs (expensive circuits) are needed for each incoming link of every ATLAS I chip; in this tradeoff situation, HIC/HS is clearly the option to be chosen, given the optimization of the switch chip for short links.

Load Monitoring: ATLAS I includes hardware support for the accelerated measurement of the cell loss probability (CLP) of its real traffic, for non-back-pressured classes of service. In order to accelerate the measurement, this hardware simulates a set of buffers of smaller size than the real chip buffer, loaded by a (different) subset of the real VCs each, and measures their (increased) CLP. The measurement is completed in software, by running a sophisticated extrapolation algorithm that computes the real CLP from the simulated CLP's [6].

3 Credit-Based Flow Control

In switch-based networks, contention for outgoing links of switches may lead to buffer overflows; these are handled in either of two ways. Some networks allow packets/cells to be dropped, and use *end-to-end* congestion control protocols [18, 26]. Other networks use *credit-based (back-pressure) flow control*, on a hop-by-hop basis, so as to never drop packets. ATLAS I supports both methods. In networks (or service classes) where cell dropping is allowed, ATLAS I provides ATM Forum standard EFCI [26]. In addition, ATLAS I implements optional, multi-lane credit flow control, which has many important advantages as explained below.

We believe that credit-based flow control will be mostly useful in high-performance applications running on top of a Network of Workstations. Parallel applications provide the underlying interconnection network with lots of bursty traffic. Many-to-one communication patterns that are very common in parallel programs, can easily lead to severe network contention. Although rate-based flow control has been proposed to regulate flow in ATM networks, it works at its best only when the traffic sources behave in a repeated predictable way (e.g. voice/image transmission). In the presence of unpredictable bursty data traffic, rate-based flow control algorithms drop cells leading to messages arriving incomplete. To ensure reliable data delivery, programmers are forced to use a high-level flow control protocol (e.g. TCP/IP) on top of the hardware-provided rate-based flow control. Unfortunately, such high level protocols take large amounts of CPU resources, require the expensive intervention of the operating system kernel, and may waste even more bandwidth by retransmitting a whole message even when a single ATM cell is dropped. This situation can soon lead to thrashing, since messages that arrived incomplete will be retransmitted, leading to more cell drops, which will lead to more message retransmissions, etc. On the contrary, credit-based flow control delivers all cells of a message reliably[2], relieving higher levels of the burden of implementing an expensive software flow control protocol. Thus, credit-based flow control reduces (even eliminates in some cases) the software intervention and all its associated overheads, while at the same time allows full utilization of the network's bandwidth.

Furthermore, credit-based flow control allows message transmission to start at maximum (available) transmission rate, being automatically throttled down to the correct level only if and when necessary. This results in minimum transmission time for a given message. On the contrary, rate-based flow control has an initial ramp-up delay (usually a few round-trip times) before it is able to achieve maximum transmission rate. Thus, a message transmitted using rate-based flow control will suffer some initial overhead, which is especially high for short messages. Given that traffic in NOWs consists mostly of unpredictable bursts of short messages, we can easily see that credit-based flow control is more appropriate for this kind of traffic than rate-based flow control.

[2] Except in case of line transmission noise, which is very rare in today's LAN technologies

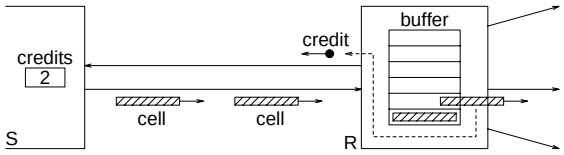

Fig. 2. Back-pressure (credit) flow control: the credit count says how much buffer space is available in the downstream switch.

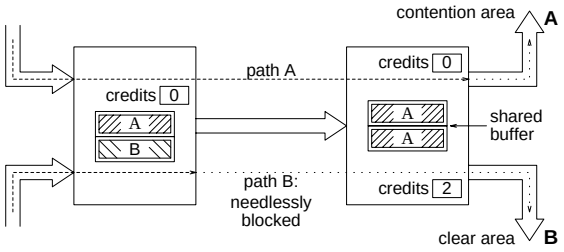

Fig. 3. Indiscriminate (single-lane) back-pressure and buffer sharing: a burst of cells destined to A has filled all available buffer space in the downstream switch, and was then stopped by back-pressure due to contention further down the path to A. As a result, a cell destined to B is needlessly blocked in the upstream switch; poor link utilization results.

3.1 Back-Pressure (Credit) Flow Control

Credit flow control, also called back-pressure, or hop-by-hop window, or virtual channel (circuit) flow control, dates back to Tymnet [27], GMDNET [14], and Transpac [29], in packet switched networks; see also the survey [13]. Under credit flow control, a cell is only transmitted to the downstream neighbor if the transmitter knows that buffer space is available for it at the receiver. This is implemented as illustrated in figure 2, using a count of available receiver buffer space. This "credit count" is decremented every time a cell departs and is incremented when a credit token is received; credits are sent back every time a unit of buffer space becomes available. For a given link peak throughput, the receiver buffer space that is necessary and sufficient is equal to this *throughput* times the *round-trip time* (RTT) of data and credits.

In the "single-lane" back-pressure described above, there is *no selectivity* of which cells (VCs) it is desirable to stop and which ones to let go. This creates a behavior similar to *head-of-line blocking* in input queueing [19], leading to performance degradation. Also, it is not possible to implement multiple levels of priority with single-lane back-pressure. An analogy in every-day life is single-lane streets (where the name came from): cars waiting to turn left block cars headed straight. If one uses the single-lane credit flow control described above, and *indiscriminate sharing* of buffer space, then bursty traffic creates the equivalent effect to head-of-line blocking, as illustrated in figure 3.

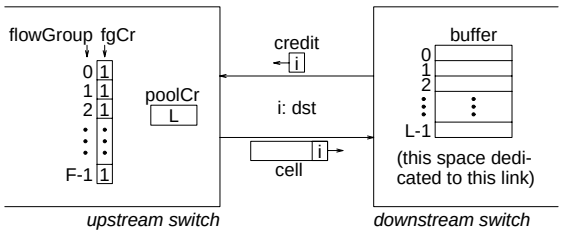

Fig. 4. ATLAS I back-pressure: F flow groups sharing L lanes on a link. A cell can only depart if it can get a pool credit (so as to ensure that there is buffer space for it at the other end) *and* the (single) credit of its flow group (so as to ensure that identically destined cells do not consume excessive buffer space).

3.2 The ATLAS I Multi-Lane Back-Pressure

Multi-lane (VP/VC-level) credit-based flow control solves the above performance problem by selectively allowing unblocked traffic to proceed, while preventing congested connections from occupying all available buffer space thus delaying all other traffic as well. In wormhole routing, multi-lane back-pressure was used to prevent deadlocks [9], or to improve performance [10]; multi-lane wormhole routing has been implemented in hardware in Torus [9] and iWarp [3]. For ATM, VC credit-based flow control extensions have been proposed and studied for long-distance (WAN) links; e.g. see [4, 24]. In case of contention, VC-level credits combined with round-robin servicing allow the fair sharing of the available network capacity among all competing flows [17, 20, 22]. DEC's GIGAswitch/ATM [30] is a commercial ATM switch that implements VC credit flow control. Our earlier switch, Telegraphos [23], also had a simple form of multi-lane back-pressure.

The credit flow control protocol of ATLAS I is reminiscent of, but simpler than those of [4] and [24], since the ATLAS I protocol is adapted to short links and single-chip hardware implementation. ATLAS I credits operate on the granularity of *flow groups*. A flow group is defined as a set of connections, sharing one or more links in their path through the network, whose cells need never or can never overtake each other, i.e. whose cell order is preserved over the length of their common path. Ideally, each flow group on each network link should consist of all connections whose destination is one, common, *single-threaded* device. When this definition leads to an impractically large number of flow groups, either a compromise is suggested (group together connections that go to nearby destinations), or a higher level of back-pressure signals may be employed to appropriately guide the multiplexing (selection) of cells at their point of entry into the flow group.

The ATLAS I chip is optimized for short-distance links; thus, each flow group can have at most 1 credit, for 1 cell's worth of buffer space. [3] Given that the

[3] Although each flow group can have only one ATM cell in each switch, each link may have several flow groups, and thus several cells in each switch.

round-trip time on links up to several meters long, including switch logic, is no more than one cell time (about 700 ns), one credit per flow group is enough. This restriction also simplifies queueing, since there is no need to remember cell order when at most one cell per VC can be present inside the switch at any given time. Figure 4 shows the ATLAS I credit protocol. For each link, the downstream switch (statically) allocates a buffer pool of size L. In order for this buffer space not to overflow, the upstream switch maintains a credit count (per output port), called *pool credit* ($poolCr$), corresponding to the available pool space at the receiver. This buffer is shared among cells according to their flow group. As shown in figure 4, separate credit counts, $fgCr$, are maintained for each flow group, initialized to 1. Since the buffer pool contains L cell slots, and since each flow group is allowed to occupy at most 1 of these slots, L of the flow groups can share this buffer pool at any given time; thus, L is the *number of lanes*.

A cell belonging to flow group i can depart if and only if $fgCr[i] > 0$ (i.e. $fgCr[i] == 1$) and $poolCr > 0$. When it departs, $fgCr[i]$ and $poolCr$ are decremented by 1, each (i.e. $fgCr[i]$ becomes 0). When that cell leaves the downstream switch, a credit carrying its flow group ID, i, is sent up-stream. When credit i is received, $fgCr[i]$ and $poolCr$ are incremented by 1, each (i.e. $fgCr[i]$ becomes 1). The pool credit counts of all upstream neighbors of a switch (for the links leading to this switch) are initialized to values whose sum does not exceed the total buffer space available in the switch.

3.3 Wormhole Routing and ATM

Several developers are skeptical about adopting ATM for network-based parallel computing, since several of its properties are fundamentally different from the familiar wormhole routing that has been successfully used in multiprocessors for several years. For example, wormhole routing usually employs variable size messages that propagate as a string of bits (worm) through the network, not allowing bits of different messages to get interleaved. On the contrary, ATM employs fixed-size cells that may be intermixed with other cells, since each of them carries identification information in its header.

We believe however, that ATM with back-pressure flow control (like that implemented in ATLAS I) is very similar to wormhole routing, and thus such ATM networks are more attractive for use NOWs and cluster-based computers: In wormhole routing, message packets may have variable sizes, but they are all a multiple of a fixed *flit* size. Although the flit size in early parallel processors was 1 byte long, flit sizes in recent wormhole-based interconnection networks range between eight [5, 28], sixteen [12], and thirty two [31] bytes. Extrapolating from this trend, flit sizes in the near future may easily reach up to 48 bytes (which is the size of an ATM cell). Current architecture trends push flit size to higher values for the following reasons:

- The control circuits of a switch must operate at the flit rate (especially if back-to-back single-flit packets are to be handled correctly). Clock rates increase by about 30% each year (see [7] section 1.2.2). Interconnection network

throughput increases 45% each year ([8] page 8). Since network throughput increases faster than clock speed, if flit size remained constant, flit rate would increase faster than clock rate. Thus, the control circuits would have to operate at a rate faster than the clock rate, which would make them very complicated (if implementable at all). Increasing the flit size, decreases flit rate, which allows the switch circuits to operate at rates less or equal to the available clock rate.

- Switch developers prefer short flit sizes, in order to reduce buffer space requirements. However, there is no benefit from reducing flit size below $(round\ trip\ time) \times Throughput$. Reducing flit size to that value reduces buffer space requirements; further reduction leaves buffer space unaffected, while increasing flit rate, hence control speed. For short low-bandwidth networks, $(round\ trip\ time) \times Throughput$ is a few bits long. For example, for a 100 Mb/sec link that is one meter long, the above value is close to 1 bit. For a modern 2.5 Gb/sec link that is ten meters long, this value is close to 32 bytes.

On the other hand, it is not desirable to increase flit size with no bound, since large flits imply long latencies. Most messages, especially in parallel applications, are short. For example, a cache line is 32-64 bytes long; a "remote write" message is a couple of words long. The prompt delivery and handling of short messages is critical for the completion time of the most applications [11]. Thus, the flit size in wormhole networks will not exceed a small value, probably in the neighborhood of 48 bytes. Thus, a packet in wormhole routing will be a sequence of 48-byte flits stored in switch buffers in the network. This is exactly what an ATLAS I ATM message is: a sequence of 48-byte cells (53 bytes including the header) stored in a sequence of switches from the source node to destination node.

Since latency is of critical importance for short messages, some wormhole networks provide multi-lane wormhole routing [10], which allows several messages to share the same physical link, as long as each of them gets its own *lane*. Thus, although some lanes may be blocked with long and slowly moving messages, some other lanes may be free and allow short messages to get routed to their destination rapidly. To provide similar properties in ATM networks, ATLAS I implements *multi-lane* back-pressured flow control. Each VC (or VP) can occupy at most a single lane; thus, slowly moving VC's (or VP's) leave the remaining lanes free to other traffic.

Although both multi-lane wormhole routing and multi-lane back-pressure ATM base flow control on credits and lanes, they use lanes differently, which is the reason for their significant performance differences. Wormhole dedicates each lane to a single packet until the end of that packet's transmission; if the packet gets blocked, so does the lane. Also, wormhole allows packets with the same destination to simultaneously occupy multiple lanes on the same link; while no advantage is gained relative to using a single lane, the remaining traffic is deprived of the additional lanes. In [21] we compared the performance of these two network architectures, using simulation. Our simulation results show that, for a same buffer size and number of lanes, ATM performs consistently better than

wormhole –by a large margin in several cases. For ATM, saturation throughput approaches link capacity with buffer sizes as small as 4 or 8 cells per link (for 64-port networks), and stays high regardless of the number of lanes. On the other hand, multi-lane ATM switches provide much lower delay than single-lane ones when bursty or hot-spot traffic is present in the network. When the number of lanes is higher than the number of hot-spot destinations, non-hot-spot traffic remains virtually unaffected by the presence of hot spots. For wormhole routing, much larger buffer sizes and many lanes are needed in order to achieve high saturation throughput. When saturation throughput is low, delay characteristics suffer as well, because any given operating point of the network is closer to saturation. Bursty and hot-spot traffic negatively affects wormhole performance to a much larger degree than it does for ATM. [4]

4 Implementation Status

At the time of this writing (December 1996), the architecture of ATLAS I has been fully specified, and the design of the chip is under progress. ATLAS I will be fabricated by SGS Thomson, Crolles, France, in 0.5 μm CMOS technology. The majority of the chip blocks are synthesized using standard cell libraries and macrocell generators of the UNICAD SGS-Thomson design environment, while a few multi-port and CAM blocks are implemented in full-custom. The latter are basically the Free List, which contains 256 flip-flops and a priority encoder, and the Creditless Cell List, which contains a 2-port memory and a 3-port memory with 1 search port each, and a 2-port memory with 1 read-modify port.

The estimated area of the ATLAS I chip is 225 mm^2, as determined by the pre-existing pad frame to be used. The 16 bidirectional serial link interfaces occupy approximately 60 mm^2, and another 10 mm^2 are taken by the remaining pads. The core of the chip is estimated to occupy roughly 120 mm^2. Table 1 presents preliminary estimates of chip area; the largest block in the core is expected to be the shared data buffer, occupying about 25 mm^2 when implemented with compiler-generated memories.

5 Conclusions

Although ATM is a rapidly advancing standard both for voice/image and data communications, it has not been widely accepted by the LAN community yet, since most ATM switches suffer from *high latency*, and *data loss* under bursty traffic.

In this paper we briefly described the architecture of ATLAS I, a single chip ATM switch that is appropriate both for Local Area Networks (NOWs, DANs), and for Wide Area Networks, since it provides sub-microsecond latency and back-pressure flow control that avoid data loss. We have mostly focused on the novel aspects of ATLAS I, which are:

[4] More information ca be found in [21].

Part	Area Estimate: mm^2
16 Bidirectional Serial Link intf. & pads	60
Other pads	10
Elastic Buffers, etc.	20
Pipelined Shared Data Buffer	25
Queue Management (full-custom)	10
Credit Extraction, Queueing, Insertion	15
Routing/Translation Table	15
Credit Table	15
Control & Monitoring, other logic	20
Total chip (determined by pad frame)	225

Table 1. Area estimates of the ATLAS-I ATM switch

- *Sub-microsecond latency:* ATLAS I provides cut-through routing –the head of a cell can depart from the switch before its tail has entered the switch. Short latency is of crucial importance for several distributed and parallel computer applications running on gigabit ATM LANs, The sub-microsecond latency of ATLAS I represents more than an order of magnitude improvement relative to today's ATM switch boxes.
- *Back-pressure (credit-based) flow control:* Most ATM manufacturers are reluctant to adopt credit-based flow control, since they believe that it requires large amounts of memory to store ATM cells. Although this is true in WANs, back-pressure requires very little additional memory when used in LAN environments. In this paper we present a novel (but simple) credit-based flow control algorithm, and show that it is feasible to implement it within the limited silicon area of a CMOS single-chip ATM switch.
- *Single-Chip implementation:* Although there exist several single-chip ATM "switches", most of them provide little functionality, and operate mostly as unbuffered crosspoint matrices, or as buffered switches with rather primitive buffer organizations. ATLAS I is the first single-chip ATM switch to provide back-pressure flow control, shared buffering, VP/VC translation, and cut-through capabilities at the same time.

Based on our experience in designing and building ATLAS I, we believe that it is an appropriate switch for Local Area Network configurations, including Networks of Workstations (NOWs), Clusters of Workstations (COWs), and System (Desktop) Area Networks (SANs or DANs). At the same time, by being an ATM switch, ATLAS I is an appropriate building block for large scale ATM Wide Area Networks.

Acknowledgements

ATLAS I is being developed within the "ASICCOM" project, funded by the Commission of the European Union, in the framework of the ACTS Programme. This is a collective effort, with many individuals making valuable contributions. We would like to thank all of them, and in particular Vagelis Chalkiadakis, Christoforos Kozyrakis, Chara Xanthaki, George Kalokerinos, George Dimitriadis, Yannis Papaefstathiou, George Kornaros, Costas Courcoubetis, Luis Merayo, Roland Marbot, Helge Rustad, Vassilios Siris, Kostas Kassapakis, Bjorn Olav Bakka, Geir Horn, and Jorgen Norendal.

References

1. T.E. Anderson, D.E. Culler, and D.A. Patterson. A Case for NOW (Networks of Workstations). *IEEE Micro*, 15(1):54–64, February 1995.
2. N.J. Boden, D. Cohen, and W.-K. Su. Myrinet: A Gigabit-per-Second Local Area Network. *IEEE Micro*, 15(1):29, February 1995.
3. S. Borkar and e.a. Supporting Systolic and Memory Communication in iWarp. In *Proc. 17-th International Symposium on Comp. Arch.*, pages 70–81, 1990.
4. G. Varghese C. Ozveren, R. Simcoe. Reliable and Efficient Hop-by-Hop Flow Control. *IEEE Journal on Selected Areas in Communications*, 13(4):642–650, May 1995.
5. J. Carbonaro and F. Verhoorn. Cavallino: The TeraFlops Router and NIC. In *Proceedings of the Hot Interconnects IV Symposium*, pages 157–160, 1996.
6. C. Courcoubetis, G. Fouskas, and R. Weber. An On-Line Estimation Procedure for Cell-Loss Probabilities in ATM links. In *Proceedings of the 3rd IFIP Workshop on Performance Modelling and Evaluation of ATM Networks*, July 1995.
7. D. Culler, J.P. Singh, and A. Gupta. *Parallel Computer Architecture*. Morgan Kaufmann, 1996.
8. M.D. Dahlin. *Serverless Network File Systems*. PhD thesis, University of California at Berkeley, 1996.
9. W. J. Dally and C. L. Seitz. Deadlock-Free Message Routing in Multiprocessor Interconnection Networks. *IEEE Transactions on Computers*, C-36:547–53, May 1987.
10. W.J. Dally. Virtual-Channel Flow Control. In *Proceedings of the 17th Int. Symposium on Computer Architecture*, pages 60–68, May 1990.
11. T. von Eicken, D. E. Culler, S. C. Goldstein, and K. E. Schauser. Active Messages: A Mechanism for Integrated Communication and Computation. In *Proc. 19-th International Symposium on Comp. Arch.*, pages 256–266, Gold Coast, Australia, May 1992.
12. M. Galles. The SGI SPIDER Chip. In *Proceedings of the Hot Interconnects IV Symposium*, pages 141–146, 1996.
13. M. Gerla and L. Kleinrock. Flow Control: A Comparative Survey. *IEEE Trans. on Communications*, 28(4):553–574, 1980.
14. A. Giessler and e.a. Free Buffer Allocation - An Investigation by Simulation. *Comput. Networks*, 2:191–208, 1978.
15. R. Gillett. Memory Channel Network for PCI. *IEEE Micro*, 16(1):12, February 1996.

16. D. B. Gustavson. The Scalable Coherent Interface and Related Standards Projects. *IEEE Micro*, 12(2):10–22, February 1992.
17. E. Hahne and R. Gallager. Round Robin Scheduling for Fair Flow Control in Data Communication Networks. In *Proc. IEEE Int. Conf. on Commun.*, pages 103–107, 1986.
18. V. Jacobson. Congestion Avoidance and Control. In *Proceedings of the ACM SIGCOMM '88 Conference*, pages 314–329, 1988.
19. M. Karol, M. Hluchyj, and S. Morgan. Input versus Output Queueing on a Space-Division Packet Switch. *IEEE Trans. on Communications*, COM-35(12):1347–1356, December 1987.
20. M. Katevenis. Fast Switching and Fair Control of Congested Flow in Broad-Band Networks. *IEEE Journal on Selected Areas in Communications*, SAC-5(8):1315–1326, October 1987.
21. M. Katevenis, D. Serpanos, and E. Spyridakis. Credit-Flow-Controlled ATM versus Wormhole Routing. Technical Report 171, ICS-FORTH, Heraklio, Crete, Greece, July 1996. URL: file://ftp.ics.forth.gr/tech-reports/1996/1996.TR171.ATM_vs_Wormhole.ps.gz.
22. M. Katevenis, S. Sidiropoulos, and C. Courcoubetis. Weighted Round-Robin Cell Multiplexing in a General-Purpose ATM Switch Chip. *IEEE Journal on Selected Areas in Communications*, 9(8):1265–1279, October 1991.
23. M. Katevenis, P. Vatsolaki, A. Efthymiou, and M. Stratakis. VC-level Flow Control and Centralized Buffering. In *Proceedings of the Hot Interconnects III Symposium*, August 1995. URL: file://ftp.ics.forth.gr/tech-reports/1995/1995.HOTI.VCflowCtrlTeleSwitch.ps.gz.
24. H.T. Kung, T. Blackwell, and A. Chapman. Credit-Based Flow Control for ATM Networks: Credit Update Protocol, Adaptive Credit Allocation, and Statistical Multiplexing. In *Proceedings of the ACM SIGCOMM '94 Conference*, pages 101–114, 1994.
25. R. Marbot, A. Cofler, J-C. Lebihan, and R. Nezamzadeh. Integration of Multiple Bidirectional Point-to-Point Serial Links in the Gigabits per Second Range. In *Proceedings of the Hot Interconnects I Symposium*, 1993.
26. H. Ohsaki and e.a. Rate-Based Congestion Control for ATM Networks. In *Proceedings of the ACM SIGCOMM '95 Conference*, pages 60–72, 1995.
27. J. Rinde. Routing and Control in a Centrally Directed Network. In *Proc. Nat. Comput. Conf.*, 1977.
28. S. Scott and G. Thorson. The Cray T3E Network: Adaptive Routing in a High Performance 3D Torus. In *Proceedings of the Hot Interconnects IV Symposium*, pages 147–156, 1996.
29. J. Simon and A. Danet. Controle des Ressources et Principes du Routage dans le Reseau Transpac. In *Proc. Int. Symp. on Flow Control in Comp. Networks*, pages 63–75, February 1979. as reported in: L. Pouzin: "Methods, Tools, and Observations on Flow Control in Packet-Switched Data Networks", IEEE Transactions on Communications, Vol. COM-29, No. 4, April 1981, p. 422.
30. R. Souza, P. Krishnakumar, C. Ozveren, R. Simcoe, B. Spinney, R. Thomas, and R. Walsh. GIGAswitch System: A High-Performance Packet-Switching Platform. *Digital Technical Journal*, 1(6):9–22, 1994.
31. B. Zerrouk, V. Reibaldi, F. Potter, A. Greiner, and A. Derieux. RCube: A Gigabit Serial Links Low Latency Adaptive Router. In *Proceedings of the Hot Interconnects IV Symposium*, pages 13–18, 1996.

This article was processed using the LaTeX macro package with LLNCS style