ATLAS I: A General-Purpose, Single-Chip ATM Switch with Credit-Based Flow Control

Manolis Katevenis†, Dimitrios Serpanos††, and Panagiota Vatsolaki†

Address all correspondence to Manolis Katevenis at:
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
E-mail: katevenis@ics.forth.gr Tel.: +30 (81) 391664 Fax: +30 (81) 391661

URL: file://ftp.ics.forth.gr/tech-reports/1996/1996.HOTI.ATLAS_I_ATMswitchChip.ps.gz

ABSTRACT: ATLAS I is a 16x16 single-chip ATM switch, featuring 20 Gbit/s aggregate I/O throughput, sub-microsecond cut-through latency, gigabaud serial links, link bundling, rate-based and optional credit-based flow control, 256-cell shared buffer, three service classes, 54 output queues, multicasting, and load monitoring support. It is being implemented in 0.5 micron CMOS technology. Its intended use is in systems ranging from wide area (WAN) to LAN and embedded (desktop – DAN) networking, supporting a mixture of services from real-time, guaranteed quality-of-service to best-effort, bursty and flooding traffic, in a range of applications from telecom to multimedia and multiprocessor NOW. We present an overview of the ATLAS I chip architecture, emphasizing flow control, short-distance link optimizations, and queue model. Then, we describe its internal organization and give its implementation status.

KEYWORDS: *general-purpose single-chip ATM switch, VLSI switch / router, credit-based flow control, multi-lane back-pressure, wormhole routing, gigabaud serial links, ATM LAN / NOW, hardware queue management.*

1. Introduction

Asynchronous Transfer Mode (ATM) [LeBo92] has been adopted as the main high speed technology in both wide and local area networks. This paper presents *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] – the "Gigabit Switching" task project of the European Union *ACTS* (Advanced Communication Technologies and Services) Programme.

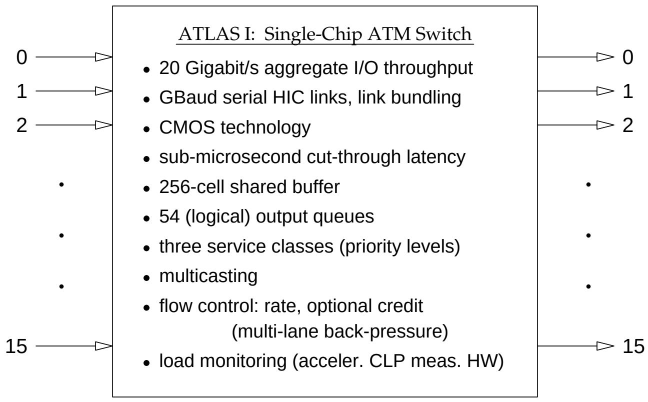

Figure 1: ATLAS I chip overview

ATLAS I is a single-chip 16×16 ATM switch, with point-to-point serial links running at 622 Mbits/s each. It is a general-purpose building-block chip for high-throughput and low-latency networks, providing a number of features and mechanisms, as illustrated in figure 1. Its intended use is in systems ranging from wide area (WAN) to metropolitan (MAN), local (LAN), and desktop (DAN) area networking. It is appropriate for supporting a mixture of ser-

† Institute of Computer Science, FORTH, Heraklion, Crete Greece; also with the Dept. of Computer Science, University of Crete.

†† IBM, T.J.Watson Research Center, Yorktown Heights, NY USA; work performed at FORTH while on leave of absence from IBM.

[1] The ASICCOM Consortium consists of industrial partners (INTRACOM, Greece; SGS THOMSON, France and Italy; BULL, France), telecom operators (TELENOR, Norway; TELEFONICA, Spain), and research institutes (FORTH, Greece; SINTEF, Norway; Politecnico di Milano, Italy; NCSR Democritos, Greece).

IEEE Hot Interconnects IV Symposium Proceedings, Stanford, Ca., USA, 15-17 August 1996

vices ranging from real-time, guaranteed-quality-of-service to best-effort, bursty and flooding traffic. These networks can provide a range of services from telecom to multimedia and multiprocessor NOW (network of workstations) applications.

The following sections discuss particular aspects of the chip, starting with the links and the uses of the chip in systems (section 2), and continuing with flow control (section 3), operation and queueing model (section 4), and the internal organization and a brief outline of implementation (section 5).

2. Gigabaud Serial Links

ATLAS I is the key component of a chipset of building blocks for high-speed wide and local area ATM networks. Three chips are needed to build large systems:

- Switch (ATLAS I)
- End-System Interface (ESI)
- Long-Link Interface (LLI)

The ATLAS I chip is optimized for short links, such as inside switching fabrics and in local and desktop area networks. Components that are only needed for long links were excluded from the switch chip, so that their cost is only paid where necessary. The switch is also optimized for efficient use of the limited buffer memory that can fit inside a single chip.

2.1 IEEE Standard 1355 HIC/HS Links

To implement a multi-gigabit-per-second switch chip, one has to solve first the problem of providing that much bandwidth through the chip's periphery (pins). ATLAS I uses *STRINGS,* the high-speed parallel-serial conversion macrocell that has been developed by BULL [MCLN93]. The CMOS version of this macrocell is capable to operate at Gigabaud speed on the serial side, which is the signaling frequency required to provide 622 Mb/s of data rate, given the 8b/12b encoding used by the circuit.

The ATLAS I links run ATM on top of IEEE Std. 1355 HIC/HS, the protocol of the STRINGS macrocell, as the physical layer. Although ATM today is most frequently run on top of SDH/SONET (STM1/STM4), ATM is a network technology that is independent of physical layer, and has been run at least on UTP-3, T1/T3, E1/E3, STM1/STM4, and FDDI-physical. Although HIC needs a higher signaling (Baud) rate than SDH/SONET for a given effective (data) bit rate, we preferred HIC for three reasons.

First, 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 (section 3) 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 a dedicated control character.

2.2 Link Bundling

Different switch configurations are desirable for different applications. On one hand, for high-end switching fabrics we prefer building blocks with higher-speed individual links, even if the number of links has to be reduced, because higher-speed links allow better statistical multiplexing, while the demand for more links can always be satisfied by building larger fabrics. Thus, a 4×4 switch with 2.5 Gbps links is preferable for such applications. On the other hand, for line-multiplexer applications, and perhaps for gigabit-LANs, we prefer building blocks with more links, even if the individual link speed is (relatively) lower. For such applications, a 16×16 switch with 622 Mbps links is often preferable.

ATLAS I satisfies all these different preferences, by providing *link bundling.* The chip is configurable so that its I/O wires can operate either as individual independent links of 622 Mbps each, or as pairs operating at 1.25 Gbps (the pair), or as quadruples operating at 2.5 Gbps (the quad), or as octets operating at 5.0 Gbps (the octet). Furthermore, an ATLAS I chip is able to operate with its links bundled in mixtures of the above configurations. Link bundling can operate in two ways: *(i)* bit (or byte) interleaving, where odd bits go on one wire and even bits on the other, in a two-wire bundle, or *(ii)* cell interleaving, where each cell is sent on any one of the wires that happen to be available at the time when the cell is ready to depart. The first method offers lower transmission time per cell, but it represents a major departure from the normal (unbundled) mode of operation. The second method is much easier to implement, because the datapath of the switch operates identically under all bundling configurations; the only difference is in the control of the switch −in scheduling which cell goes over which link. ATLAS I uses this second method.

All wires that constitute a bundled link must depart from one switch chip and arrive at *one* other switch chip. In other words, a bundled link cannot be routed through parallel switch chips operating in a ‘‘bit-sliced’’ fashion, because in-order delivery of the cells in individual VCs, as required by ATM, would not be guaranteed. Such a network topology and mode of operation would remind adaptive routing, but, unfortunately, adaptive routing gives out-of-order delivery.

2.3 Gigabit LAN and DAN Configurations

ATLAS I is a building block for high-throughput, low-latency local and desktop area networks carrying client/server, multimedia, and/or multiprocessor traffic. Cost reduction is a major consideration in such systems, so the number of chips should be as small as possible. It is envisioned that the network will be built by placing the switch chips at locations that are convenient depending on the wiring organization of the system or building (in desktop boxes, ceilings, wiring closets, hubs, etc). Switch chips will be connected among themselves or to *End-System Interface (ESI)* chips via coaxial cable, for distances up to a few meters, or via opto-electronic converters and fiber, for longer distances. ESI chips are needed as network adapters for end user systems; on the network side they must provide HIC/HS link connectivity. The ESI chip is outside the scope of the ASICCOM project, but a subset of its functionality is provided by the demonstrator system for ATLAS I which is being built by the ASICCOM partners.

The simplest ASICCOM-based ATM networks consist of ATLAS I switches connecting end-users via ESI chips, using coaxial-cable HIC/HS links. ATLAS I chips placed near end-systems can serve as multiplexors/demultiplexors of traffic, or even to connect the processor to its peripheral devices in DAN configurations. The controller (manager) of the network can be shared among many switches; any of the interconnected end-systems can be the controller −ATLAS I allows management operations to flow into the switch through regular network links− or else the controller can be a dedicated microprocessor that is attached to the control/monitoring port of an ATLAS I chip (section 5.6).

ATLAS I ‘‘islands’’ like the above can be connected to each other and to third-party and wide area ATM networks as shown in figure 2; the latter connections usually need to be over SDH/SONET as physical layer. A conversion chip, the *Long Link Interface (LLI),* is needed in that case, as illustrated in the right part of figure 2.

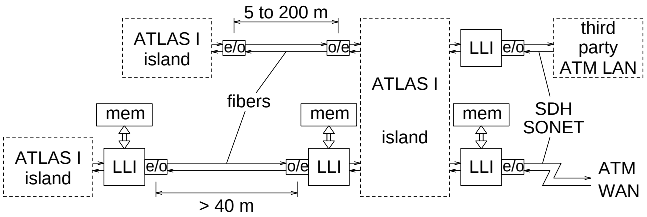

Figure 2: ATLAS I and LLI in Gigabit LAN configurations

The LLI serves a dual purpose: provide physical layer conversion, and interface the short switch links to long length network links. Because of the increased cost of long links, it is desirable to use an efficient coding scheme such as SDH/SONET (8 bits per byte) on them. The LLI also provides a number of other functions that are needed at the external interfaces of switching fabrics or at the end-points of long links: large translation tables, large cell buffer memory, flow control format conversion, and policing and charging control. In order to provide large tables and buffering, the LLI chip includes the capability of adding external RAM chips to it. In figure 2, LLIs are used for physical protocol conversion in the upper right part, for increased buffer memory in the lower left part[2], and for both purposes in the lower right part. The LLI chip is outside the scope of the ASICCOM project, but a subset of its functionality is provided by the demonstrator system for ATLAS I which is being built by the ASICCOM partners.

2.4 WAN Switching System Configurations

Large switching nodes for ATM WANs can be built using ATLAS I and LLI chips, for example as shown in figure 3 (control microprocessor(s) are omitted for clarity). Such large switches can be attached to other vendors' switches to provide a wide-area multivendor interoperable network. The switching fabric can have any desired topology, as e.g. bilateral banyan, fat tree, etc. The ATLAS I chips in the fabric can also be used as multiplexers/demultiplexers. External links of speeds lower than 622 Mbps can also be readily interfaced to an ATLAS I link: in the incoming direction, the ATLAS link will be underutilized; in the outgoing direction, the ATLAS link can be throttled using HIC/HS (single-lane) back-pressure.

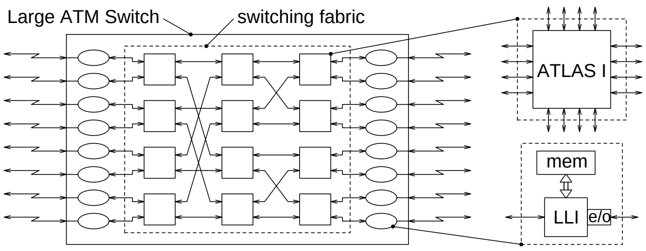

Figure 3: Large ATM switch for SDH WANs

The external links of the switching system will run ATM on SDH/SONET and will use optical fiber. As in the case of ATM LANs, the internal HIC/HS links are interfaced to these external links using LLI chips and electro-optical converters. As above, LLIs can have external RAM attached to them, which is used for various of the functions that these interfaces perform besides physical layer conversion: charging, policing, flow control conversion (e.g. credit-rate conversion, or credit ‘‘bundling’’ inside cells), large VP/VC translation tables, cell buffering (because of the long length of the link).

---

[2] when the length of a link exceeds a few tens of meters, the cell-credit round-trip delay exceeds the cell time so a single back-pressured VC is no longer able to saturate the link by itself.

3. Credit-Based Flow Control

ATLAS I features advanced flow control capabilities. Besides ATM Forum standard rate-based *EFCI* (Error Forward Congestion Indicator) [Ohsa95], *optional credit-based* flow control is additionally provided. Multi-lane (VP/VC-level) credit-based (back-pressure) flow control, like what ATLAS I offers, has many important advantages over rate-based flow control. 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; in case of contention, the available network capacity is fairly shared among all competing flows [HaGa86] [Kate87] [KaSC91].

We believe that the battle between rate and credit based flow control may be far from over [RaNe95], and that credit-based flow control may become very popular in the future. ATLAS I will be ready when that happens. At the moment, credit flow control is optional on ATLAS I, so ATLAS-based standard-ATM networks can disable it. On the other hand, gigabit ATM LANs (including multiprocessor ''NOW'' and desktop area networks), where credit flow control is quite popular, can take advantage of ATLAS's credit features. Also, large ATM switches that use ATLAS I as building block for their switching fabric can use internally ATLAS's credit flow control in order to provide the good performance of output queueing at the low cost of input buffering; at the interfaces to the external links, the credit information can be converted to standard rate information.

3.1 Flow Groups

The credit flow control protocol of ATLAS I is adapted to 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 and 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 employeed to appropriately guide the multiplexing (selection) of cells at their point of entry into the flow group. In ATLAS I, up to 4096 flow groups can exist, per link. ATLAS I chips can *merge* multiple incoming flow groups into each outgoing flow group.

3.2 The ATLAS I Credit Protocol

The credit flow control protocol of ATLAS I is reminiscent of, but simpler than those of [OzSV95] and [KuBC94], since it is adapted to single-chip hardware implementation and optimized for short-distance links. Each flow group can have at most one credit, for one cell's worth of buffer space. Given that the round-trip time on links up to several meters long, including switch logic, is no more than one cell time (about 680 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 flow control 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 *pool credit* (*poolCr*) count per output port, 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.*

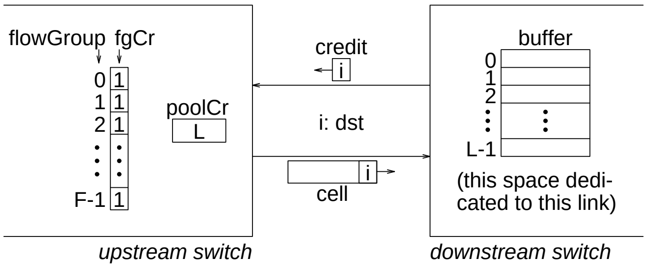

Figure 4: the ATLAS I credit-based flow control protocol

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 such a cell departs, $fgCr[i]$ and $poolCr$ are decremented by 1, each (i.e. $fgCr[i] \leftarrow 0$). When that cell leaves the down-stream 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. $fgCr[i] \leftarrow 1$). 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 ATM versus Wormhole Routing

The credit flow control protocol used in ATLAS I has similarities to the multi-lane back-pressure [Dally90] used in wormhole routing. Wormhole Routing [DaSe87] is an interconnection network architecture that is frequently used in multiprocessors. IEEE standard 1355, which is used for the ATLAS I links, uses single-lane back-pressure. However, multi-lane back-pressure performs much better than single-lane back-pressure [Dally90], so all proposals to add credit flow control to ATM have considered multi-lane protocols exclusively. When credit flow control is enabled in ATLAS I, its links run an extension of the IEEE standard 1355, which implements multi-lane credits.

Credit-flow-controlled ATM has similarities to wormhole routing: their hardware complexity is comparable, they both use credits, and they both transport packets after segmentation into fixed size *cells* or *flits.* Traditionally, the size of wormhole flits was a few bytes, while the ATM cell size is 48 bytes plus a 5-byte header. However, as technology evolves, the flit size is pushed to higher values, because the throughput-round-trip-time product increases (especially in NOW's), and because of limitations on the clock frequency of the switch control circuits, which must operate at the flit rate.

We have studied the performance of credit-flow-controlled ATM, using the credit protocol of ATLAS I, and we compared that to the performance of multi-lane wormhole routing [KaSS96]. Our simulation used detailed and realistic switch models, which operate at clock-cycle granularity. The results of our study show that ATM with credits performs significantly better than wormhole routing, because ATM uses lanes more efficiently: ATM provides high throughput and low latency with much less buffer space than that required by wormhole routing; also, ATM demonstrates little sensitivity to bursty traffic, and, unlike wormhole, it is fair in terms of latency in hot-spot configurations.

Our conclusion is that ATM, which was originally conceived for wide area networking and was subsequently also used in the local area, is an appropriate technology for multiprocessor interconnection networks as well, provided it is extended with credit-based flow control; perhaps, then, ATM deserves the title of universal network architecture. We come to this conclusion considering that credit-flow-controlled ATM performs much better than wormhole routing –the most popular multiprocessor network architecture– and considering that local area networks of workstations (NOW) gradually displace traditional multiprocessors [ACPN95]. This explains the general-purpose nature of the ATLAS I single-chip switch, as claimed in section 1.

4. Operation and Queueing Model

Figure 5 shows the service classes supported by the ATLAS I chip. Three classes are provided, differentiated by their level of priority. Whenever a cell of a higher priority class exists and is ready to depart over a link, such a cell will always be selected, regardless of the possible presence of lower priority cells. The classes provided are the interesting combinations of two parameters: policed or non-policed traffic, and back-pressured or non-back-pressured connections. This reflects the desire for the hardware to provide *mechanisms rather than policies:* the service classes are defined in terms of hardware behavior, in a parameterized, orthogonal way; their use by the various kinds of traffic is left to be decided by the network manager. The network manager takes other factors too in consideration when making this decision, such as buffer allocation, VP/VC set-up method, policing and admission control algorithms, etc.

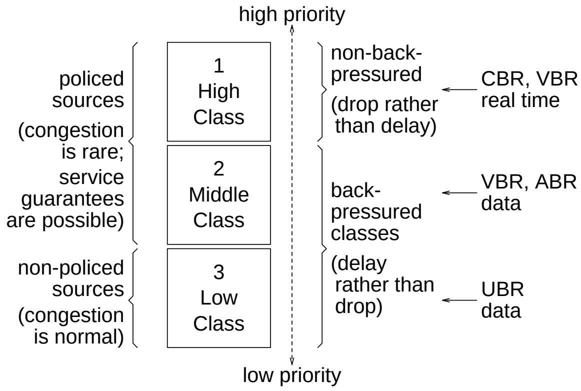

Figure 5: The three service classes of ATLAS I

Figure 6 shows the logical model of the ATLAS I operation. For simplicity, only 3 of the 16 input and output links of the switch are shown. The bodies of incoming cells are stored in the shared data buffer, unless it is determined that the cell should be dropped (see below). The address where each body is stored is (pseudo-) random; this address is used as an identifier (pointer) for the cell in the control (management) section of the chip. The control section looks at the incoming cell headers one at a time; FIFO order is observed among the headers of cells that arrived through links that are bundled together. Each header first goes through the (shared) translation/routing table (4 K entries), which determines the service class and the bit-mask (16 bits) of outgoing link(s).

Cells that are allowed to depart from the switch are called *ready;* when contention for the outgoing link(s) allows it, they depart. The identifiers of (pointers to) the ready cells are maintained in *Ready Cell Queues (RQ).* There are three sets of ready queues, one per service class. For each service class, there is one ready queue per outgoing link for unicast cells, plus one queue shared among all links for the multicast cells; these latter queues are shown with dotted lines in figure 6. ATLAS I has 17 outgoing links (the 17th link is internal – see section 5.6), so there are 18 ready queues per service class, for a total 54 ready queues for all classes. The cells in each queue are served (i.e. they depart) in FIFO order. Selection among the queues is made according to priorities.

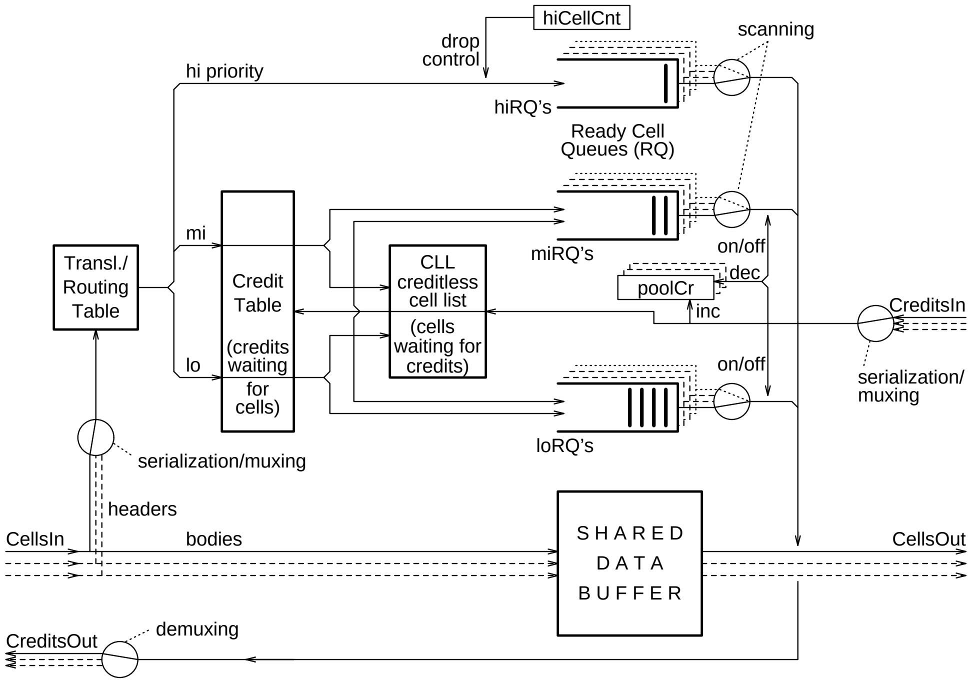

Figure 6: ATLAS I queue model (only 3 of the 16 links shown, for clarity)

Cells that belong to the high service class are not subject to credit flow control (back-pressure); thus, they are either enqueued directly to the proper ready queue, or, on buffer overflow, they are dropped. All high-class cells share *one common partition* of the data buffer. As it is known, this shared-queueing buffer allocation results in the minimum possible cell loss probability for a given buffer memory space [HlKa88]. The size of the data buffer partition that is dedicated to high-class cells is determined by the network manager, and can be changed at run-time. Note that this is only a *logical* partition, i.e. high-class cells are physically stored anywhere in the data buffer, but their total number is controlled so as to not exceed a given upper limit.

Arriving cells that belong to the middle and low service classes must acquire a credit for their desired outgoing link and flow group before they can become ready for departure and get enqueued in the proper ready queue. Multicast cells must acquire credits for their flow group on *all* desired outgoing links before they can become ready. For these cells, control first consults the *Credit Table,* a two-dimensional array of flags; one dimension is the flow group identifier, and the other dimension is the outgoing link identifier. Credits needed by the cell and found in the table are taken from it. If all credits needed by the cell were found and taken, then the cell is ready for departure, and thus it is enqueued to the proper ready queue. Else, the cell is recorded in the *Creditless Cell List (CLL),* where it waits for the remaining credits to arrive.

Incoming credits are examined one at a time. Each of them increments the proper pool credit counter, and then it is given to the creditless cell list (CLL). If there are cells waiting for this credit in that list, one of them is selected (pseudo-) randomly and gets the credit; if this was the last credit that the cell was waiting for, the cell becomes ready and gets enqueued to the proper ready queue. Else, if no cells wait for this credit in CLL, the credit is recorded in the credit table.

On the switch output side, cell departures are initiated one at a time over the outgoing links. For each outgoing link, control considers the three (shared) multicast-cell ready queues and its three (dedicated) unicast-cell ready queues. Middle and low class ready queues are only considered when their pool credits are not exhausted. Most pool credits are shared between middle and low class cells, so figure 6 shows a single pool credit counter per outgoing link. However, a minimum number of pool credits (not shown in figure 6) is reserved for the middle class, so that low class cells can never lock middle class cells out of the buffer. These minimum limits, as well as the number of total pool credits per link are set by the network manager, and can be modified at run-time. Upon cell departure, for each outgoing link, control makes a selection among the appropriate non-empty ready queues, according to priorities. When a unicast cell or the last copy of a multicast cell departs, a credit is sent back to the upstream neighbor where this cell came from.

5. Internal Organization

We present now the internal organization of the ATLAS I switch and its current implementation status. The organization of ATLAS I exploits several of the techniques used in the Telegraphos II switch chip [KVES95]. The datapath of the switch consists primarily of the shared data buffer, implemented as a pipelined memory; the dual-ported, pipelined control section is time-shared among all incoming and outgoing links.

ATLAS I will be fabricated in 0.5 μm CMOS technology. The core of the switch operates under a clock of approximately 50 MHz; the datapath width is 2 bytes per link. The parallel-serial conversion macrocells that interface the core to the external serial links (section 2) operate under separate clocks, of approximately 84 MHz; they handle one byte per clock period. There are elastic buffers between these macrocells and the switch core. Figure 7 shows the block diagram of the switch core.

5.1 Datapath

The datapath of the switch, along the top and right sides of figure 7, consists almost exclusively of the shared data buffer, of size 256 ATM cells. It performs the buffering and the switching (and cut-through) functions. It uses the pipelined memory architecture [KaVE95] [Efth95], which is particularly suitable for VLSI switch buffers. Besides its compactness and speed advantages, this architecture automatically provides virtual cut-through without needing any additional control circuits for that purpose.

The pipelined memory operation is controlled by telling the buffer to initiate one cell write operation (from one of the incoming link latches) or one cell read operation (to one of the outgoing links) in each 20 ns clock cycle. Once initiated, a read/write operation proceeds at the rate of one pipeline stage per clock cycle, in parallel with all other similar on-going operations in different pipeline stages each; each stage is a single-port memory. The datapath width is 16 bits per link.

5.2 Control Organization

Control is the most challenging part of the ATLAS I design. It manages cell and credit arrival and departure events, and keeps track of the buffered cells in multiple queues and in the creditless cell list. To provide the high throughput and specialized functions required, control includes some content-addressable and some multi-ported register files and memories. The control section of ATLAS I is:

- *time-shared* among the 16 incoming and 16 outgoing links; this matches well the pipelined memory organization, where at most one cell read or write operation can be initiated per clock cycle
- *dual-ported:* one port handles cell arrivals and departures; in parallel, the other port handles credit arrivals
- *pipelined:* each port handles events at the rate of one per clock cycle. There are at least 33 clock cycles per cell time; thus, control can handle all traffic on all 16 incoming and 16 outgoing links, even when cells arrive and depart back-to-back. During the 33rd cycle, rare extra events are handled: additional cell arrivals due to clock frequency mismatch, and control/monitoring cells through the 17th, internal port. Credit arrivals are handled in parallel, at the rate of at least 2 per link per cell time.

The first control port consists of a five-stage pipeline, shown in the left and middle part of figure 7 with the stages arranged left to right. This control port handles cell events, at the rate of one per clock cycle; stages three and beyond of this pipeline are time-shared among cell arrivals and cell departures. The *I/O Link Scheduler,* in the second pipeline stage, decides which event will be serviced by the third stage of the cell control pipeline during the next cycle. When there are one or more outgoing links that are idle or about to become idle and whose ready queues are non-empty, the scheduler chooses one of these links and instructs the rest of the control pipeline to dispatch the next highest-priority cell onto that link. We give priority to outgoing links over incoming links, since delaying the outputs causes loss of useful throughput, while delaying the management of inputs has no immediate consequences. At most 16 initiations of cell transmissions can be needed in every 33-cycle interval, so we are guaranteed that at least 17 cycles per cell time remain untaken by the above process. During these remaining cycles, if the routing-table stage contains a header of an arriving cell, the scheduler instructs the rest of the control pipeline to process this incoming cell.

5.3 Cell Arrivals

In the first stage of the cell control pipeline, the *Header Serializer* arbitrates among the up to 17 headers of cells that may appear simultaneously or may have been waiting on the 16 incoming links and on the internal cell injection path (section 5.6). Headers are serviced in arrival order (FIFO order), so that cell order within a connection is maintained for cells arriving through bundled links. In the second stage, the *VP/VC Translation/Routing Table* determines the service class, the outgoing link mask (*outMask*), and other such information for the incoming cell. Since incoming headers were serialized before reaching it, this is a single-port memory.

When the link scheduler decides to handle a cell arrival, the outputs of the routing table are given to a number of blocks in the third pipeline stage. The *Credit Table (CT)* contains the flow-group credits, *fgCr*, for all outgoing links (section 3.2, figure 4). It is read via port A to determine the mask of available credits for this flow group on all outputs.

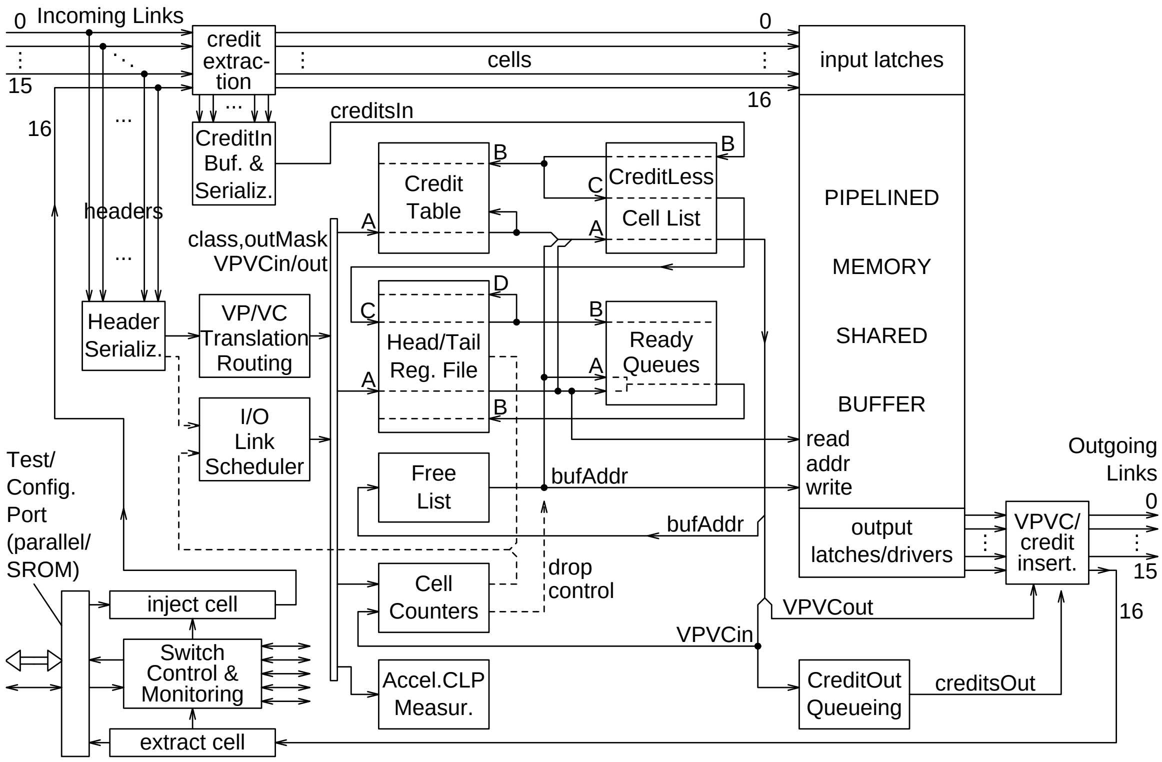

Figure 7: Block diagram of the ATLAS I switch chip core

Cells belonging to back-pressured classes must "consume" the credits that correspond to their *outMask*, so these bits of CT must be cleared. These write accesses could be performed via a separate port of CT, but we economize on the number of ports by keeping these updates in temporary registers, and performing them during cell departure cycles, when port A of CT remains unused; 16 such registers are needed, plus the corresponding bypass paths. If all desired outgoing links for this cell have credits (if the mask of credits is a superset of the *outMask*), then the incoming cell is ready for transmission, so it should be enqueued on the proper ready queue. Else, the cell information, including a mask of the missing credits, must be written into the creditless cell list, as discussed below.

In parallel with the credit table access, in the third pipeline stage, the class and link identifiers are used to index the queue tail pointer table. This table is contained in the *Head/Tail Register File (HTRF),* which holds the head and tail pointers of the 54 ready queues (section 4). The head/tail register file is four-ported; the above tail read access is performed via port A, and it is the first step of possibly enqueueing the incoming cell to the desired ready queue.

The third pipeline stage also contains the *Free List (FL)* block, which keeps track of all empty slots in the shared buffer memory. This block generates the address of an empty slot in the shared data buffer, into which the incoming cell can be written. Another block contains the *Cell Counters (CC)* and their limits, e.g. the pool credit counters (*poolCr*) for the outgoing links, the counts of cells per service class (*hiCellCnt*), etc. For each incoming cell, a check is made to see if it can be accepted or if it has to be dropped due to overflow of the class or buffer partition to which it belongs.

At the end of the third pipeline stage, it has been determined whether the cell will be accepted or dropped; if accepted, we know the buffer address into which it will be written, and whether it will be enqueued into a ready queue or held in the creditless cell list, waiting for one or more credits to arrive. All these will be done in the fourth pipeline stage. Also, at the end of the third stage, if it is determined that the cell will be enqueued into a ready queue, we conclude that that ready queue is non-empty. This information is passed back to the link scheduler; if the corresponding outgoing link is idle or about to become idle, then the scheduler will trigger a cell departure on that link. If that queue was empty before this cell arrival, then the cell to depart is the cell currently arriving, so *virtual cut-through* will occur; this takes place "automatically", because concurrent enqueue and dequeue operations on the same queue are handled correctly by the control pipeline (using bypassing), and because the pipelined memory shared buffer provides automatic cut-through when a cell starts being read shortly after it starts being written into that memory [KaVE95].

In the fourth pipeline stage, if it has been determined that the cell is to be accepted, the address provided by the free list is given to the pipelined memory shared buffer, and a cell write operation is initiated into that buffer slot. Also in the fourth pipeline stage, several pieces of information regarding the incoming cell (when not dropped) are written into the *CreditLess cell List (CLL),* via its port A. Besides a mask of credits that the cell is waiting for, where the name of the list comes from, the *outMask*, the service class, and the incoming and outgoing VP/VC numbers are also recorded in this list; for this reason, actually, an inscription into this list is made for all incoming cells, regardless of whether they have or do not have all the credits that they need for departure.

If it has been determined that the incoming cell is to be enqueued in a ready queue, then this operation is performed in the fourth pipeline stage, into the *Ready Queue (RQ)* memory, which contains the linked lists of the queues; this is a two-ported, 256-word memory. The tail pointer of the proper queue was read from HTRF during the previous cycle; the buffer address of the new cell was also determined during the previous cycle by the free list. If the enqueue operation is enabled (all credits were found), the buffer address is written into the RQ entry pointed to by the tail pointer (unless the queue was empty). The tail pointer of the queue also has to be updated (and the head, if the queue was empty). This is done via port B of the head/tail register file (HTRF), and it could be done during this fourth pipeline stage; however, this port is time-shared with cell departures (dequeue operations), which must update HTRF during their fifth pipeline stage, so that same timing is used for enqueues resulting from cell arrivals.

The Head/Tail Register File, the Ready Queues, the Free List, and the Creditless Cell List form the *Queue Management Block,* the heart of the ATLAS I control. Their architecture, operation, and design are discussed in full detail in [Kozy96].

5.4 Cell Departures

Cell departures are initiated by the same control pipeline that handles cell arrivals. The I/O Link Scheduler decides when to initiate cell departures, over which link, and for which service class. This information is used to index the head/tail register file (HTRF) in the third pipeline stage, giving us (via port A) the head pointer for the ready queue from which the cell must be dequeued.

In the next clock cycle (fourth pipeline stage), several operations proceed in parallel. The cell address that was read from the register file is used in the pipelined memory shared buffer, initiating the read operation of the cell itself. This same address is provided to the ready queue memory; port A of that memory is used to read the next element of the queue, which now becomes the new head. The new head of the queue is written into the head/tail register file during the next, fifth, pipeline stage, using port B of that register file.

In the fourth pipeline stage, the cell address is also used to read the VP/VC identifiers −incoming and outgoing− and the *outMask* of the cell; this information is kept in the creditless cell list (CLL) −independent of the fact that the cell is not creditless any more. The access to the *outMask* portion of CLL is composite: the bit that corresponds to the current outgoing link is cleared (written), while the bits corresponding to the other links are read; for multicast cells, when all these other *outMask* bits are zero, it means that the copy of the cell presently departing is the last copy. When a unicast cell or the last copy of a multicast cell is departing, the cell address is given to the free list (FL) block, signaling that this buffer slot is now empty; also, the cell counters are notified of the cell departure (decremented), and the incoming VP/VC and link identifiers of the cell are used to generate a credit, which is to be sent out on the link that corresponds to this cell's incoming link.

A *Credit-Out Queueing* block is needed for the credits that may be waiting to go out on various outgoing links. The maximum credit rate allowed on a link is 2 credits per cell time. Up to 16 credits may be generated per cell time, but the links over which these credits must be sent may not be distinct from each other; in the worst case, all 16 credits may need to be sent over a single link. Given the limited credit rate on a link, a queue of considerable size is needed. The worst case credit-out queue size is 256 entries (same as the shared cell buffer capacity) total, for all outgoing links.

5.5 Credit Arrivals

The second control port processes incoming credits at the rate of one per clock cycle or 33 per cell time. It consists of a five-stage pipeline, shown in the center of figure 7. On the incoming links, there is a *Credit Extraction* block. The first stage of the credit arrival pipeline is the *Credit-In Serializer:* credits extracted from the incoming links must be buffered and then serialized, so that the control pipeline can handle them one at a time. The second stage of the pipeline consists of a search in the creditless cell list (port B), to see if there are cells waiting for this credit.

Depending on the outcome of this search, the third pipeline stage is one of the following two. If no cell was waiting for this credit, then the credit is written into the credit table (use port B to set the corresponding bit). Else, choose one of the cells waiting for this credit (for fairness reasons we want the choice to be pseudo-random), clear its corresponding credit-waiting flag, and read its other credit-waiting flags and its outgoing link and service class; these are done through port C of the creditless cell list. If all other credit-waiting flags are zero, then this cell has now become ready for transmission, so we enqueue it to the ready queue that corresponds to its outgoing link and service class. This is done in the fourth and fifth stages of the control pipeline, using port C of HTRF to read the queue's tail pointer, and then, in parallel, port D to write the new tail and port B of the RQ memory to write the new linked-list node.

5.6 *Chip Control & Monitoring*

Throughout the switch, there is a test / configuration / control / monitoring bus, called *State Bus,* that can read and write almost all registers, memories, and state of the ATLAS I chip. The only exceptions are "transient" latches, whose contents never stay for more than a few cycles, and always go into other registers where they can be captured; also, the pipelined memory shared buffer is a special case: due to its density, reading and writing for test purposes will be done through the cell injection/extraction path.

The state bus is driven by the *Control & Monitoring* block. Read and write commands are given to this block in two ways: *(i)* through the *parallel/SROM* external port of the chip, intended for connection to a control microprocessor or to an initialization serial ROM, or *(ii)* through ATM cells that arrive to this switch on VCs such that the routing table guides them to the special, internal, 17th port of the pipelined memory, and from there into the *extract cell* register, rather than to a real outgoing link. Read commands for the internal chip state generate a response (and write commands may generate an acknowledgement). These responses are given back in two ways: either through the parallel/SROM external port of the chip, or inside *injected ATM cells* that are routed through the switch, using a special, internal, 17th port of the pipelined memory. These 17th input and output ports have a peak rate quite lower than 1 cell per cell time; they are serviced during some of the "33rd" clock cycles. The injected cells can be routed to any outgoing link of the switch.

5.7 *Implementation Status*

At the time of this writing (July 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. The following table 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.

Preliminary Area Estimates (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

Conclusions

Single-chip ATM switches that provide high throughput, low latency, and advanced architectural features are feasible in today's CMOS technology. They can be flexible enough to become general-purpose building blocks for all kinds of networks, just like microprocessors are building blocks for all kinds of information processing systems. We presented ATLAS I, one such single-chip ATM switch that is being implemented within the European Union ACTS Programme. Its most notable feature is credit-based flow control, an option that offers advanced capabilities to the networks built with this chip. ATLAS I uses high-speed serial link interfaces operating at Gigabaud rates in CMOS, a pipelined memory shared data buffer, and a time-shared, dual-ported, pipelined control engine that manages multiple queues and schedules multiple links at high speed.

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, Evangelos Markatos, Vassilios Siris, Kostas Kassapakis, Bjorn Olav Bakka, Geir Horn, and Jorgen Norendal.

References

[ACPN95] T. Anderson, D. Culler, D. Patterson, and the NOW team: "A Case for NOW (Networks of Workstations)", *IEEE Micro Magazine,* vol. 15, no. 1, February 1995, pp. 54-64.

[DaSe87] W. Dally, C. Seitz: "Deadlock-Free Message Routing in Multiprocessor Interconnection Networks", *IEEE Trans. on Computers,* vol. C-36, no. 5, May 1987, pp. 547-553.

[Dally90] W. Dally: "Virtual-Channel Flow Control", *Proc. of the 17th Int. Symp. on Computer Architecture,* ACM SIGARCH vol. 18, no. 2, May 1990, pp. 60-68.

[Efth95] A. Efthymiou: "Design, Implementation, and Testing of a 25 Gb/s Pipelined Memory Switch Buffer in Full-Custom CMOS", Technical Report FORTH-ICS/TR-143, *ICS, FORTH, Heraklio, Crete, Greece,* November 1995; URL: file://ftp.ics.forth.gr/tech-reports/1995/ 1995.TR143.Design_Implementation_25Gbs_PipelinedMem_Switch_Buffer.ps.gz

[HaGa86] E. Hahne, R. Gallager: "Round Robin Scheduling for Fair Flow Control in Data Communication Networks", *Proc. IEEE Int. Conf. on Commun.,* June 1986, pp. 103-107.

[HlKa88] M. Hluchyj, M. Karol: "Queueing in High-Performance Packet Switching", *IEEE Journal on Sel. Areas in Communications,* vol. 6, no. 9, December 1988, pp. 1587-1597.

[Kate87] M. Katevenis: "Fast Switching and Fair Control of Congested Flow in Broad-Band Networks", *IEEE Journal on Sel. Areas in Communications,* Vol. SAC-5, No. 8, October 1987, pp. 1315-1326.

[KaSC91] M. Katevenis, S. Sidiropoulos, C. Courcoubetis: "Weighted Round-Robin Cell Multiplexing in a General-Purpose ATM Switch Chip", *IEEE Journal on Sel. Areas in Communications,* Vol. 9, No. 8, October 1991, pp. 1265-1279.

[KVES95] M. Katevenis, P. Vatsolaki, A. Efthymiou, M. Stratakis: "VC-level Flow Control and Shared Buffering in the Telegraphos Switch", *Proceedings of the Hot Interconnects III Symposium,* Stanford Univ., CA, USA, Aug. 1995. URL: file://ftp.ics.forth.gr/tech-reports/1995/1995.HOTI.VCflowCtrlTeleSwitch.ps.gz

[KaVE95] M. Katevenis, P. Vatsolaki, A. Efthymiou: "Pipelined Memory Shared Buffer for VLSI Switches", *Proceedings of the ACM SIGCOMM '95 Conference,* Cambridge, MA USA, Aug. 1995, pp. 39-48. URL: file://ftp.ics.forth.gr/tech-reports/1995/ 1995.SIGCOMM95.PipeMemoryShBuf.ps.gz

[KaSS96] M. Katevenis, D. Serpanos, E. Spyridakis: "Credit-Flow-Controlled ATM versus Wormhole Routing", Technical Report FORTH-ICS/TR-171, *ICS, FORTH, Heraklio, Crete, Greece,* July 1996; URL: file://ftp.ics.forth.gr/tech-reports/1996/ 1996.TR171.ATM_vs_Wormhole.ps.gz

[Kozy96] C. Kozyrakis: "The Architecture, Operation, and Design of the Queue Management Block in the ATLAS I ATM Switch", Technical Report FORTH-ICS/TR-172, *ICS, FORTH, Heraklio, Crete, Greece,* July 1996; URL: file://ftp.ics.forth.gr/tech-reports/ 1996/1996.TR172.QueueManagement.ps.gz

[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,* London, UK, 31 Aug. - 2 Sep. 1994, ACM Comp. Comm. Review, vol. 24, no. 4, pp. 101-114.

[LeBo92] J. LeBoudec: "The Asynchronous Transfer Mode: A Tutorial", *Computer Networks and ISDN Systems,* vol. 24, no. 4, May 1992.

[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", *Hot Interconnects I Symposium,* Stanford Univ., CA, USA, Aug. 1993.

[Ohsa95] H. Ohsaki, e.a.: "Rate-Based Congestion Control for ATM Networks", *Computer Communication Review,* ACM SIGCOMM, vol. 25, no. 2, April 1995, pp. 60-72.

[OzSV95] C. Ozveren, R. Simcoe, G. Varghese: "Reliable and Efficient Hop-by-Hop Flow Control", *IEEE Journal on Sel. Areas in Communications,* vol. 13, no. 4, May 1995, pp. 642-650.

[RaNe95] K. Ramakrishnan, P. Newman: "ATM Flow Control: Inside the Great Debate", *Data Communications,* June 1995.