A Highly Scalable Switching and Routing Chipset

Andreas Kirstädter
Corporate Technology, Siemens AG, Otto-Hahn-Ring 6
Munich, 81730, Germany

Matthias Heink and Ulrich Fiedler
Infineon Technologies, 1730 North First Street
San Jose, CA 95112, USA

Christoph Heer
Infineon Technologies, Otto-Hahn-Ring 6
Munich, 81730, Germany

Abstract

The steeply rising usage of the Internet for multimedia applications requires large and scalable router architectures to handle the exploding amounts of traffic. This paper presents a novel system architecture for packet switches and routers with Ethernet interfaces. It consists of packet protocol processing chips with more than 20 Ethernet/Fast-Ethernet/Gigabit-Ethernet interfaces and central crossbar-based switching chips used for interconnecting these protocol chips. While each crossbar switching chip alone provides a throughput of nearly 100 Gbps the crossbar switching chips themselves can also be cascaded in ring structures allowing a total system throughput in the order of several hundred Gbps. The overall architecture operates in a virtual output queuing mode supported by a sophisticated internal messaging.

Keywords: Switching, Routing, Crossbar, Contention Resolution, Quality of Service

1 Introduction

The last years have seen a steep rise in the available transmission capacities for voice and data networks. To fulfill the promise of an all-integrating high-bandwidth support for everyone switches and routers are needed with throughputs in the multi-gigabit and terabit per second range. They not only have to support the common routing and switching protocols but they also have to be easily scalable and have to properly support the Quality of Service (QoS) architectures and protocols defined in the last years.

At such high speeds the complete data plane of the network protocols has to be implemented in hardware. Since the processing capabilities of the single packet protocol processing chips (P3Cs) in a high-speed switch or router are limited (mostly by memory bandwidth) it is necessary to use several of them in a parallel but coupled operation mode. Several alternatives exist for connecting the single chips in the system:

- Passive backplanes are severely limited in their aggregated throughput because of their bus architecture.
- Active backplanes can either be built by using shared-memory switching chips from the ATM world or by using crossbar architectures.

Since shared-memory switching chips themselves are severely limited by the bandwidth of the internal memory a cascading architecture has to be used that requires a large number of chips to stay internally non-blocking.

On the other hand the architecture of a crossbar-based switching chip (CSC) allows a fully parallel operation. Thus the total throughput of a CSC is only limited by the number of available package pins and the transmission technique used to interconnect them with the P3Cs around it. Since a crossbar itself does not contain internal buffers, transmission conflicts have to be avoided by so-called contention resolution algorithms that do a look-ahead transmission control to prevent collisions at the output ports of the crossbar.

As soon as several P3Cs are interconnected two basic internal communication problems have to be solved in the system:

- First, it is necessary to assure that each of the P3Cs has the same state of information regarding the routing and switching information expressed in the corresponding tables.
- The second need for an information exchange arises from the distributed buffering in larger architectures: often the P3C that received the packet (also called ingress P3C) will not be identical to the egress P3C that will finally forward the packet downstream to the next hop switch or router. Thus the egress P3C has to somehow get the packet from the ingress P3C at the right point in time. This requires an effective and efficient information exchange between both chips.

Since the space of this paper is limited it only explained the basic architectural aspects of the chipset and any implementation details are out of its scope. Chapter 2 describes the overall system architecture selected for the scalable switching chipset and the basic packet flow within it. Chapter 3 then explained the internal communication protocol for the information exchange between the connected P3Cs. In Chapter 4 the architecture and the basic operation of the P3Cs is explained. In Chapter 5 the implementation of a CSC together with the communication between the P3Cs and the CSCs is shown. Finally, Chapter 6 gives an overview about the most important system features of the chipset and the basic chip data.

2 System Architecture and Overall Data Flow

To allow an nearly arbitrary scaling of switching and routing systems constructed using this chipset the single P3Cs can be interconnected in several different architectures (for the sake of clarity the central microcontroller - used for processing control plane protocols - and its interconnections between with the P3Cs are not shown):

In the stand-alone mode (Figure 1) a single P3C allows the switching/routing between any of its 24 Ethernet/Fast Ethernet and 2 Gigabit Ethernet ports. It can also be configured to have 5 Gigabit Ethernet ports.

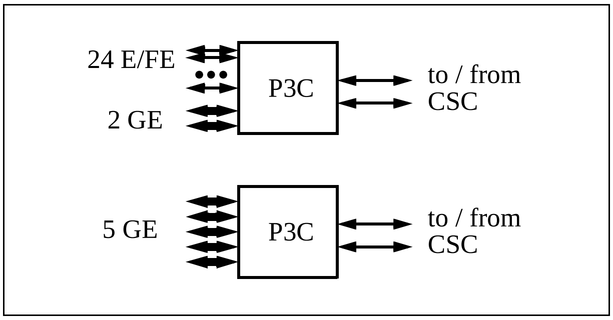

Figure 1: P3C in stand-alone mode.

For small systems several P3Cs can be interconnected via their crossbar interfaces to a ring (Figure 2).

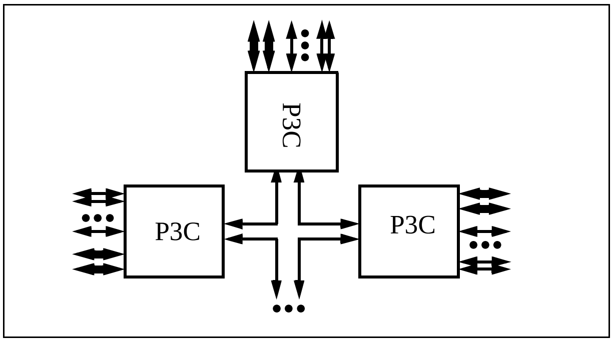

Figure 2: P3Cs cascaded in a ring.

Up to 12 P3Cs can be interconnected using a single CSC to build an internally completely non-blocking system (Figure 3).

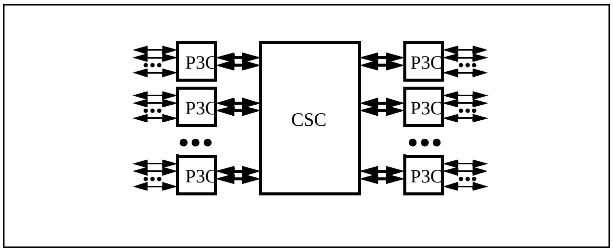

Figure 3: P3Cs cascaded over a CSC.

For even larger systems three 3 CSCs themselves also can be interconnected to a ring (Figure 4). Since a part of each CSC's ports are consumed for building up the ring, up to 6 P3Cs can be attached to each CSC allowing systems with up to 18 P3Cs.

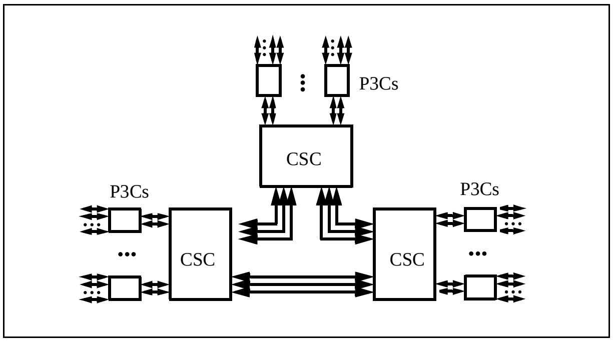

Figure 4: P3Cs cascaded over a ring of CSCs.

For the flow of the data packet through the system two alternatives have to be separated:

- So called "local" packets can be handled completely by a single P3C. This is the case when the ingress and egress ports for this packet are located on the same P3C.
- If the ingress and egress ports are located on different P3Cs these "remote" packets are handled by several P3Cs.

The latter alternative makes it necessary to take some decisions concerning the system-internal buffering model and the communication between the P3Cs. In principle, the arriving packet could be either queued at the ingress chip, at the egress chip or at both. Since the identification and classification of the arriving packet (layer 2 and layer 3 protocol processing) consumes some time it makes sense to queue the packet already at the ingress chip. On the other hand the correct processing of the QoS requirements for the single packet flows in the system can only be provided when the packet throughput of each single system output is observed: Only at the system outputs the correct decision can be taken concerning which packet to forward further downstream at which point in time. Normally this would require another buffering of the packets in the egress chip to have them readily at hand when the scheduled departure time arrives. To avoid a double buffering of packets we decided to combine our physical input buffering approach with a messaging protocol distributing the information about the presence of packets and allowing the fast delivery of packets between ingress and egress chips. This protocol will be explained below in Chapter 3. Thus we get a novel architecture combining a physical input buffering with a logical output buffering as shown in Figure 5.

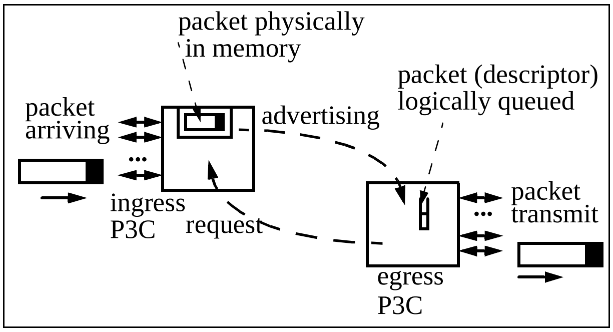

Figure 5: Buffering concept and messaging.

For the transmission to the egress P3C the ingress P3C will segment the data packets into cells of the same length (see also the description in Chapter 5 covering the communication between the P3Cs and the CSCs). This cell format is used regardless whether the P3Cs are coupled in a small ring or interconnected using a CSC or a ring of CSCs.

The basic packet and cell flow in a configuration of several P3Cs connected via a CSCs is shown in Figure 6.

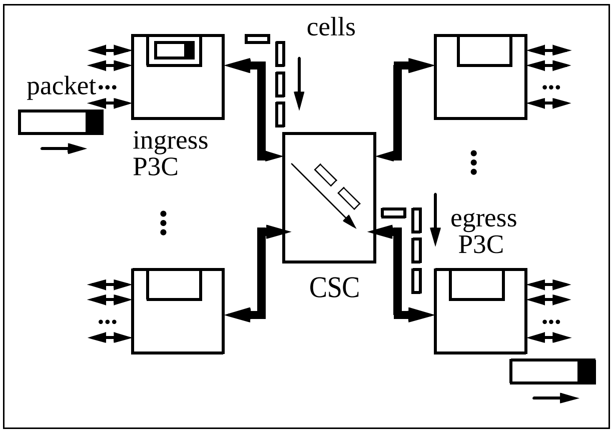

Figure 6: System architecture and data flow over the crossbar.

The ingress P3C analyzes the packet header and stores the packet in its internal packet memory. When the packet transmission time has arrived the ingress P3C segments the packet into cells of constant size and adds a self-routing header to them. These cells are then transmitted to the CSC that switches them to the egress P3C. The egress P3C then re-assembles the cells to the original packet and transmits it onto the outgoing link.

3 Information Exchange Between the P3Cs

Protocol Outline

As explained above the scheduling of the packets is done in the egress P3Cs in order to be able to respect the QoS requirements of the single packet transmissions. Therefore the ingress chips that physically buffer the packets and the egress chips that queue and schedule the pointers have to exchange information about the presence of packets and the requested packet transmissions. This communication is handled via the following protocol steps:

- At the reception of a packet the ingress P3C analyzes its header and determines via table lookup (layer 2 and/or layer 3) the output port on which this packet has to leave the system. Additionally, the packet is classified regarding its QoS requirements.
- Having completely stored the packet in its packet memory the ingress chip informs the egress chip about the presence of this packet. For this purpose it transmits to the egress chip an advertising message containing a pointer to the packet's location in the packet memory and the number of the queue where the egress chip has to queue this pointer in its own scheduling system in order to provide the correct QoS to the packet's transmission. The scheduling rates for certain packet flows have beforehand been configured by the control plane protocols running on the microcontroller.
- The egress chip now does the usual scheduling of packets that are destined to leave the system at one of its output ports. The only difference to common architectures is that for remote packets this queuing and scheduling operates on the basis of packet pointers instead of scheduling packets physically present in the egress chips. Thus another buffering operation on the egress chip is avoided.
- As soon as the pointer for a certain packet has reached the top of queue position for a certain output the egress chip requests the packet itself from the data memory of the ingress chip via a request message.
- Upon the reception of a request message the ingress chip starts transmitting the packet (segmented into cells) out of its own packet memory over to the egress chip.
- The egress chip then re-assembles and sends out the packet without further analyzing its header information.

Both the advertising and the request message are themselves also transferred within cells that are switched by the CSC. Cells transporting the packet segments and those transporting the protocol messages use the same cell format.

Learning and aging messages for the automatic build-up of the switching tables are also transferred within cells over the CSC interconnection structure.

Protocol Optimization

For short packets it would be rather inefficient to always use a complete cell for transporting a single advertising message and another cell for only transporting the corresponding request message. This would lead to a large waste of bandwidth. Therefore every P3C groups several of the advertising and request messages going to the same other P3C into message blocks that then are transferred using a cell. A message cell is transmitted over the CSC as soon as it is full or the grouping timer has run down.

The value of the grouping timer is critical for the performance of the complete switching / routing system:

- The system will be slowed down if we wait too long for the collection of messages. In this case a complete usage of the capacities of the output ports could no longer be guaranteed.
- The message grouping itself becomes inefficient if the time-out value is chosen too small: This would lead to the collection of only a few messages per message cell and cell transmission bandwidth would be wasted by the signalling.

Extensive system simulations [3] were carried out to provide guidelines for the timer values. They showed that the correct setting of the grouping timers depends on both the load pattern and the system topology:

- The grouping gets more effective if the communication of the considered P3C is limited to a few of the other P3Cs. Then the message cells fill rapidly with messages and they are mostly sent out because of being full (instead of waiting for the timer to run down).
- On the other hand, if the P3C exchanges messages with many other chips in the system the grouping gets more ineffective and the timer determines the transmission of message cells and it has to be chosen carefully to minimize the signalling bandwidth while still guaranteeing wire-speed operation of the overall system..

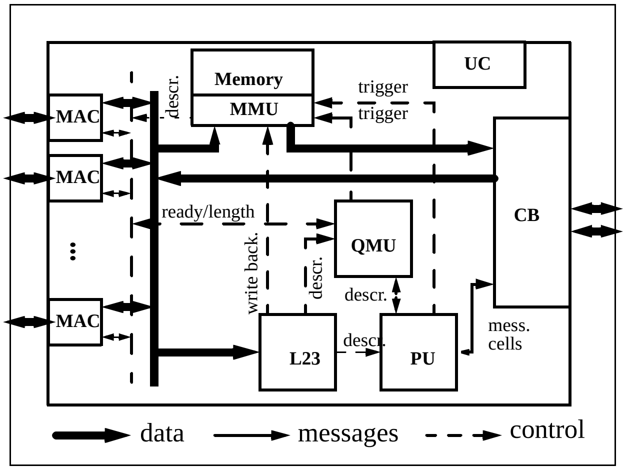

Figure 7: Internal architecture of a P3C.

4 Implementation of a P3C

Internal Architecture

Each P3C contains the units necessary for packet reception, transmission, and header analysis, as shown in Figure 7.

The corresponding MAC cores and the layer 2 and layer 3 processing unit (L23) are connected to the packet memory controller MMU and the packet queuing controller QMU. Additionally we implemented a crossbar interface unit CB that is used both for interconnecting the P3Cs with each other (in a ring) and for connecting them to a CSC. The information exchange between the P3Cs (advertising and request messages) is handled by the so called protocol unit PU.

The operation of the P3C can be configured by a microcontroller that also handles the complex control path protocols. For this purpose the corresponding UC interface unit connects to all the other blocks (these connections are not shown in Figure 7).

Basic Operation

As long as sufficient packet memory space is available each MAC core holds a descriptor from the MMU indicating a free buffer space in the packet memory. When a new packet arrives at an ingress port it will be stored at the corresponding memory location and a new descriptor will be supplied by the MMU. At the same time its header is forwarded to the L23 and analyzed there. If the packet's CRC is valid and its transmission is not suppressed by any control or management protocol the following alternatives have to be observed:

- If the packet is a local packet its descriptor will be immediately appended to the queue in the QMU that corresponds to the destination port and QoS information previously determined in the L23.
- If the packet is a remote packet its descriptor will be handed over to the PU of the ingress chip that forms an advertising message out of it and appends it to the message cell destined to the corresponding egress chip. After the PU of the egress chip has read the advertisement out of the message cell the descriptor will be appended to the corresponding queue in the QMU of the egress chip.

In the transmit direction the actions are again started by the MAC cores. The transmission of data packets over the connectivity structure (ring of P3Cs, CSC, ring of CSCs) is very time critical. Therefore the MACs have been equipped with small output buffers that are also used for the re-assembly of cells to packets. The forwarding of a new packet via the de-queuing of the corresponding descriptor therefore can already be triggered by a MAC that itself is currently still transmitting the previous packet. For that purpose a MAC is always provided by the QMU with the packet size of the next packet to be de-queued (i.e. having the highest priority). As soon as the MAC has sufficient space available in its output buffer it will signal this fact (of being "ready") to the QMU. The QMU then de-queues the top of queue descriptor from the highest priority queue that is non empty. The further handling then again depends on the locality of this packet:

- If the descriptor belongs to a local packet it is handed over to the local MMU ("trigger" signal) that then initiates a transmission of the packet itself out of the packet memory over to the MAC core that sends it out on the port.
- A descriptor pointing to a remote packet is handed over to the PU that forms a request message out of it and appends it to the message cell destined to the corresponding ingress chip. Over there, the ingress chip's PU reads the request message out of the message cell and triggers the MMU to start the transmission of the packet out of the packet memory over the CB (segmented into cells) to the egress P3C. There cells of the packet will arrive over the CB and will be immediately handed over to the egress MAC that re-assembles them to the original packet that is then sent out on the port.

For the switching / routing of multicast packets so-called internal multicast group IDs have been implemented that point to entries in special multicast handling tables located in the egress chips. This approach allows to limit the transmission of multicast packets over the interconnection structure (ring of P3Cs, CSC, or ring of CSCs) saving bandwidth and transmission time: The ingress P3C only sends a single copy to each of the destination egress P3Cs of the corresponding MC group. These egress P3Cs then further send copies of the packet out of the local destination ports on this chip corresponding to the entry in the multicast handling table.

5 Implementation of a CSC

Internal Architecture

As shown in Figure 8. the CSC consists of 24 port interfaces arranged around a central switching matrix. The port interfaces are connected to a contention resolution unit CR and to a central controller (the latter connections are not shown).

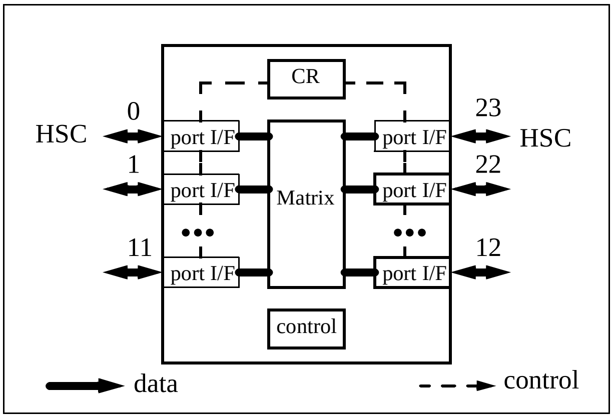

Figure 8: Internal architecture of a CSC

Interfaces and Cell Format

The CSC connects to each of the P3Cs via two High-Speed Channel interfaces (HSC). Externally each HSC operates in each direction 4 differential line pairs running at a speed of 1250 Mbps and having PECL levels. On the external lines a 8B10 coding is used for clock recovery and error detection leaving a net data rate of 4 Gbps (full- duplex) per HSC interface. The message and data cells transferred over the interfaces between the chips always have a payload of 64 bytes, a header of 6 bytes and a CRC of 2 bytes, as shown in Figure 9.

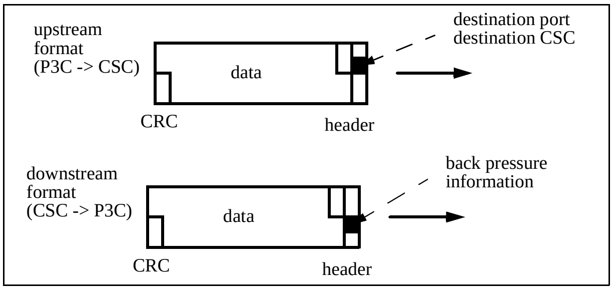

Figure 9: Structure of cells.

The HSC interfaces themselves implement bit error checking and retransmission facilities on the cell level providing a fully reliable transport channel between the P3Cs and the CSC.

Basic Operation

The CSC operates synchronously in time slots. Each 144 nsec from each of the inputs ports a cell can be switched to a destination output port. Data and message cells are not differentiated in the CSC. The P3Cs code the destination output port and the destination CSC (if a ring of CSCs has to be addressed) into the cell headers.

The CSC uses input buffering: Upon the reception of a cell the CSC puts it into its internal input buffer located in the receiving port interface. In order to avoid the well-known head-of-line blocking effect the input buffers of each port interface are divided into sub-buffers corresponding to the destination output port interfaces of the CSC. Each of the sub-buffers has the capacity of a few cells. The transmission of cells over the central matrix is coordinated by the contention resolution unit CR:

- At the beginning of each cell duration the CR receives the information from each port interface concerning which of its sub-buffers contain at least a single cell.
- The CR then runs a contention resolution algorithm (similar to that described in [1] and [2]) and takes a fair decision between all these cell transmission requests.
- The single port interfaces are then informed about the result of this contention resolution process and send at

the beginning of the next cell time slot the corresponding cells into the matrix together with a switching control information to setup the corresponding matrix points.

The extremely fast and highly parallel operation of the CR allows it to consider all the transmission requests of all input port interfaces within the CSC at the same time. This virtual output queuing approach achieves the performance of an output queued cell switch. Thus it guarantees a nearly 100% transparent operation of the matrix-CR system and the internal structures of the CSC don’t have to implement any speed-up.

At asymmetric load conditions between the single P3Cs around the CSC some output links might become overloaded and thus certain sub-buffers tend to fill up with cells. To avoid buffer overflows and the loss of cells the CSC signals a back pressure information to the connected P3Cs in the header of downstream cells. The CSC notifies the P3Cs via this back pressure about which input sub-buffers have reached a certain threshold. Each affected P3C will then stop sending cells destined to the corresponding CSC output port. This selective mechanism allows the P3Cs to continue sending cells to any of the non-congested output ports of the CSC. Therefore the overall switching system operates completely free of internal cell losses.

6 System Features and Circuit Data

Each of the P3Cs can be used as a stand-alone switch or router having either 24 Ethernet ports of 10/100 Mbps plus two ports of 1 Gbps or five 1 Gbps ports. Thus it offers a full duplex payload throughput of 5 Gbps (plus 8 Gbps full duplex for traffic and message exchange over the CSCs). It can be cascaded using a ring structure or via the interconnection facilities of the CSC (or a ring of CSCs). Further features of the P3C comprise:

- A very large Unicast and Multicast forwarding database with forwarding rates at full wire speed and automatic self-learning mechanism.
- A special HW-based protocol to keep all system tables updated reducing CPU overhead in management tasks.
- A Flow Engine and a Rule Engine allowing for on-the-fly pattern matching on the first 128 bytes of any received frame header.
- QoS support with eight priority-queues per port and a user programmable Weighted Fair Queueing scheme, bandwidth management, and rate shaping capability.
- Extensive VLAN support complying with the 802.1Q VLAN standard and with dynamic VLAN learning and automatic VLAN classification.
- Standard PCI bus CPU Interface.
- Extensive protocol processing capabilities: The P3Cs are capable of identifying various packet types and configured to send the packet exclusively to the CPU, or to a specific port, or send to the CPU and forward to other switch ports as usual.
- Special HW support for the 802.1D Spanning Tree Algorithm.
- RMON support for 4 basic groups plus capture and filtering support.
- Supported MIBs: MIB-2 Interface MIB, Ethernet MIB, MAU MIB.
- Extensive security options.

Each CSC offers a switching bandwidth of 96 Gbps full duplex for interconnecting up to 12 P3Cs and transferring data and message cells between them. Up to 3 CSCs can be interconnected to a ring where each of the CSCs can connect up to 6 P3Cs. This allows the construction of switching/routing systems with 18 interconnected P3Cs and total payload throughput of up to 90 Gbps. Further features of the CSC comprise:

- Internally blocking-free and completely lossless operation using a novel contention resolution algorithm and an input buffering architecture with back pressure to the P3Cs.
- Collection and distribution of status information concerning the load situation of the protocol units in the connected P3Cs.
- A two-step back pressure mechanism for protecting the protocol units in the connected P3Cs against overload situations.
- High-Speed Channel interfaces providing a lossless cell transfer at data rates of 4 Gbps per port.
- Cell grabbing and insertion for system tests.

Both the P3C and the CSC are designed for a Infineon 0,18 µm CMOS technology. While the HSC interface ports are working with 1250 MHz the core clock frequency is 125 MHz. Running at 2.5 V the power consumption is 7W for the P3C and 6W for the CSC. The P3C will be delivered in a BGA352 package and the CSC in a BGA961 package. First samples will be available soon.

ACKNOWLEDGMENTS

The authors would like to thank all members of the chipset teams for fruitful discussions and for the implementation support.

REFERENCES

[1] A. Kirstädter: “Fair and Flexible Contention Resolution for Input Buffered ATM Switches Based on LAN Medium Access Control Protocols”, IFIP-IEEE Broadband Communications, Montreal, Canada, April 23 -25, 1996.

[2] A. Kirstädter: “Contention Resolution for Different Traffic Categories in Large Input Buffered ATM Switches”, Proceedings of IEEE ATM '97 Workshop, Lisbon, Portugal, May 1997.

[3] W. Klein, Chr. Wincheringer: “Simulation of a High-Speed LAN Switching Chip Set”, Proceedings of SCI2001, Orlando, FL, USA, July 2001