# Self-routing Crossbar Switch with Internal Contention Resolution

[a]Christoph Heer, [b]Andreas Kirstädter, [a]Christian Sauer

[a]Infineon Technologies, Corporate Development
[b]Siemens AG, Corporate Technology
Otto-Hahn-Ring 6, D-81730 München

christoph.heer@infineon.com

## ABSTRACT

Network components for switching and routing systems are needed for the increasing demand of higher bandwidth. Especially the local area networks (LAN) see a strong move from fast Ethernet to gigabit Ethernet transmission. Therefore future router and switch architectures have to be scalable for higher bandwidth. The presented crossbar switch allows building up active backplanes and to connect up to 12 line cards with up to 12 FE ports each. With its aggregated bandwidth of 24 Gbit/sec the crossbar it's the central switching element of this system.
The circuit runs at 125 MHz and consumes about 2,5 W. The overall area of the pad-limited design is given by 64 mm² while the core area for the switching matrix, control logic and memories is about 25 mm².

## 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. At such high speed the network protocols can only be implemented in hardware. Since the processing capabilities of the single chips 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 are again 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. Crossbar switching architectures on the other hand operate fully in parallel. Thus the total throughput of a crossbar chip is only limited by the number of available package pins and the technique used to interconnect them with the protocol processing chips. Since a crossbar 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.
This paper describes a crossbar chip developed for an Ethernet switching and routing chip set. The next chapter shortly explains the overall system architecture. Then the internal block diagram of the crossbar switch is shown followed by a description of the interaction between the crossbar chip and the port chips around it. Another chapter explains the different operation modes and the test features of the chip that can also be used for system monitoring and as a load pattern generator. The paper concludes with a short description of the overall chip data.

## SYSTEM ARCHITECTURE

Figure 1 shows the overall system architecture and the data flow. The crossbar chip is connected by LVDS [1] links with a data rate of 2 Gbit/s to a number (up to 12) of port chips that do the layer 2,3,4-protocol processing. The crossbar itself operates in time-slots: the port chips segment the data packets for transmission over the crossbar into fixed-size cells with an overall length of 70 bytes (see the cell format in figure 3). Thus a crossbar timeslot has a length of 280ns.

0-7803-7057-0/01/$10.00 ©2001 IEEE. 

Therefore the way of a data packet across the system comprises the following actions:

- reception by an ingress port chip,
- protocol processing in the ingress port chip,
- packet buffering in the ingress port chip,
- segmentation into cells,
- transfer of the cells over the crossbar,
- re-assembly of the cells in the egress port chip,
- protocol processing in the egress port chip, and
- transmission of the packet to its destination.

Note: the crossbar chip operates self-routing; a field in the cell header contains the ID of the destination chip.

The protocol processing in the ingress chips consumes strongly varying amounts of time – especially if higher data communication layers are involved. Therefore it is necessary to buffer the incoming data packets in the ingress chips. In order to avoid as much as possible any further propagation delays in the system it has been decided to implement the crossbar chip itself without any internal buffering capabilities.

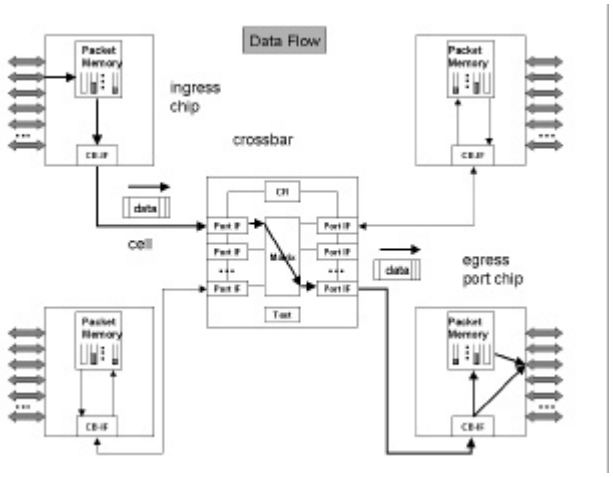


Figure 1: system architecture

Therefore cell transmission conflicts that arise when several cells from different ingress chips that are destined to the same egress chip arrive at the crossbar at the same timeslot cannot be resolved by internal buffers in the crossbar. These situations have to be avoided in advance by an efficient and fair contention resolution mechanism (e.g. [2], [3]).

For this purpose the crossbar chip contains a control unit that implements the contention resolution algorithm presented in [2] and that communicates with the transmission controllers in the port chips.

The flow of the contention resolution information between one of the port chips and the control unit in the crossbar is given in figure 2. The single steps in the contention resolution process (repeated during every timeslot) are:

- At the beginning of each cell transmission slot the single port chips communicate the availability of cells to the single crossbar output ports to the crossbar by a bit vector in the cell header of the presently sent cell: the so-called contention resolution request (CRreq).
- The crossbar extracts these bit vectors from the cell headers and feeds this information to its contention resolution unit.
- The contention resolution unit processes (during around 120ns) the CRreq from all inputs and identifies a nearly optimum allocation of contention resolution grants (CRgnt) to the single port chips.
- These CRgnt's are the transmitted back to the requesting port chips within the cell headers of the cells currently leaving the output ports of the crossbar chip.

The port chips then schedule the cells according to the received CRgnt for the transmission in the next timeslot (pipelined operation of contention resolution and data transfer).

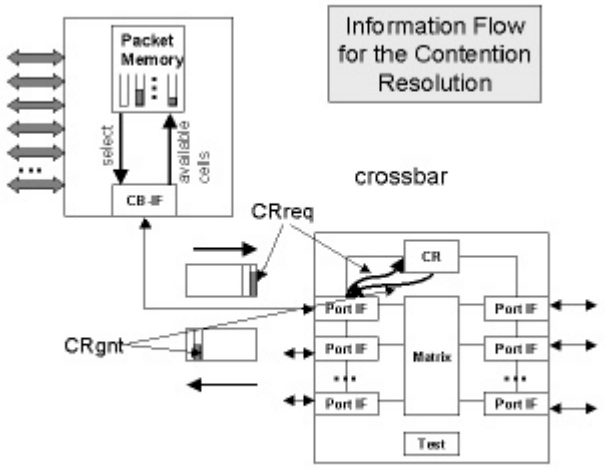


Figure 2: information flow of request and grant

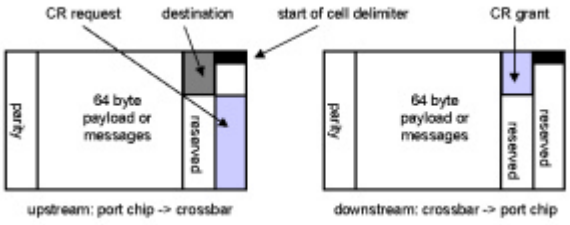


Figure 3: data cell format

## BLOCK DIAGRAM OF CROSSBAR SWITCH

Low voltage differential signaling (LVDS) ports are used for external communication from the crossbar switch to the port chips. The LVDS ports are four bits wide and are running at a clock frequency of 625 MHz.

The crossbar switch itself consists of five major units. Incoming data cells are investigated by 12 port interface modules. The port interface module detects the start of a new cell and extracts cell specific data. The destination of the cell is sent to the switching matrix, the request vector for future data cells is sent to a contention resolution unit. The data cell itself is stored in a local buffer. Outgoing cells are rebuilt at the output of the port interface module. Header and data words are received via the switching matrix. Port specific data, like parity errors detected in the incoming cell, are inserted in the outgoing cell header.

The contention resolution takes the request vectors from the 12 input ports and grants the output resources. The grant is the binary encoded number of the egress chip, reserved to receive the next data cell from the respective ingress port chip. The grant is binary encoded in the header of the outgoing cell and sent back to the respective ingress port chip.

Aim of the central controller of the crossbar switch is the generation of a global synchronization scheme for the whole system. All data cells are sent to the egress port chips synchronously to the master clock signal of the chip. As the latency of the contention resolution as well as the latency of the data path is given, the controller can force a fixed scheduling of all operations inside the circuit.

The fixed internal scheduling of operation only allows for a limited latency of the external port chips. As soon as the start signal for the contention resolution has been set, no further input data and requests can be taken into account. Therefore port chips, which have not sent their data in time or lost synchronization, are taken as not available. A grant will not be sent to these ports nor will they receive a cell with valid data. Instead a cell with empty payload data will be sent, to allow the respective port to resynchronize again.

The switching matrix is a switching array with 12 inputs and 12 outputs of 16-bit data width. The crosspoint switches are set by the destination vector of the incoming data cells as soon as a start signal from the central controller is received. Synchronously to that signal, all internal crosspoints are set. As all destinations have been derived from the grants delivered before by the contention resolution unit, conflicts are avoided.

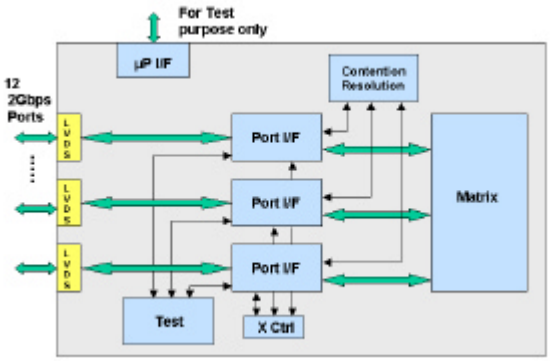


Figure 4: top level crossbar diagram

A test controller, configured via a microcontroller interface, is used to implement various test features for circuit and system analysis. Three 2 kbyte RAMs are implemented to stimulate and to monitor data cells at the input and output ports of the circuit.

## OPERATING MODES

As stated above, the incoming data cells have to be available before a certain central trigger signal for the contention resolution has been set. This scheduling implies strong requirements on the external latency of a data cell. Each data cell has to stay in the crossbar switch for at least the latency of the contention resolution plus the latency of the data in the port interface module (detection of new cell

header and reconstruction of outgoing cell data). Therefore the time budget for the external port chips to close the loop is rather low. To allow for larger external delays, the crossbar switch may be used in the so-called 'late start_of_cell mode'. This mode is statically configured via an external pin of the circuit and allows skipping the latency of the contention resolution. The request vectors are taken from the incoming cells and sent to the contention resolution. But while the grants are computed, the data cells are already sent to their destinations. The grants are sent back to the requesting ports in the next data cell. Therefore the internal delay of the crossbar switch is only given by the internal data path and the time budget for the external port chips is increased.

In addition to the two operation modes described above, the crossbar switch can be used as a fixed multiplexer or as an LVDS repeater. Every pair of port interfaces can be switched together bypassing the internal port interfaces and switching matrix. This fixed connection is only available between neighboring ports. It is configured via an external pin of the circuit. The crossbar switch just works as a LVDS repeater in this mode. Two further external pins may be used to switch the chip into multiplex mode (mux_mode on) and to decide where data cells have to go (mux_select). In this mode the ports 1, 4, 7 and 10 may be switched either to the ports 0, 3, 6, 9 or to the ports 2, 5, 8, 11. Additionally, the ports not used for this static multiplex may be used as a smaller 4-port crossbar switch (the multiplexed ports are not taken into account for the request-grant protocol).

## TESTFEATURES

A test controller with many system level features is included in the crossbar switch. Therefore the crossbar switch may be used to stimulate and monitor a complete system. Data patterns can be stored in 2 Kbytes RAM (1024 words x 16 bit) and sent to any single or any group of output port chips. This feature allows emulating a broadcast mode for the stored patterns. The pattern can be sent once or eternally until the test is stopped via the external microcontroller interface. Another 2 kByte RAM is available to monitor data at one input or output port of the crossbar switch. Data storage may be triggered as soon as a specific pattern has been seen at the input of this observation RAM. All test features are configured via the microcontroller interface. This interface is only accessible in test mode. In addition all output ports can be mirrored to the data bus of the microcontroller interface to monitor the data flow at the speed of the master clock of the crossbar switch.

## CIRCUIT DATA

The crossbar switch is designed for a Siemens 0,25 µm CMOS technology. While the LVDS interface ports are working with 625 MHz, the core clock frequency is 125 MHz. Running at 2.5 V the estimated power consumption of the chip is 2W.

## ACKNOWLEDGMENTS

The authors would like to thank Mr. Charles Bry, Mr. Reinhard Deml and Mr. Matthias Hellwig for fruitful discussions and Mr. Jörg Gliese for the implementation support.

## REFERENCES

[1] TIA/EIA STANDARD, Electrical Characteristics of Low Voltage Diffenrential Signaling (LVDS) Interface Circuits, TIA/EIA-644, March 1996

[2] A. Kirstädter, "Contention resolution for different traffic categories in large input buffered ATM switches", Proceedings of IEEE ATM '97 workshop, Lisboa, May 1997

[3] N. McKeown, M. Izzard, "The Tiny Tera: A Packet Switch Core", IEEE Micro, Jan/Feb. 1997, pp 26-33