Optical Interconnect Between Cache and Main Memory

M. Klein*, W. Paul*, J. Preiss*, G. Renz†, M. Scholl†

*Saarland University, Lehrstuhl für Rechnerarchitektur

†DLR Stuttgart, Institut für Technische Physik

Abstract

The implementation of optical I/O at integrated memory modules will lead to increased bandwiths in the near future. In a feasibility study modules are fabricated and tested in an experimental arrangement of a processor board where the cache and the main memory are interconnected by optical links. This permits to realize very broad and fast memory busses.

1 Introduction

Progress in bonding techniques and miniaturization of optical sensors and emitters have opened the perspective to provide highly integrated modules of roughly the size of a silicon chip with considerable numbers of optical inputs and outputs. Table 1 compares the bandwidth achievable by optical inputs and outputs with the bandwidth achievable by electrical pins at the time of this writing (February 2000). Here bandwith is defined as the product of the number of pins and the data rate transmitted over one pin. The optical bandwidth is around 10 Gbit/s and has roughly caught up with the electrical bandwidth; but in the near future the achievable frequency and the number of optical I/O's are both expected to grow by an order of magnitude.

	pins	frequency	bandwidth
electrical	100	100 MHz	10 Gbit/s
optical	10	1 GHz	10 Gbit/s

Table 1: Electrical pins vs. optical I/O

This exciting perspective fuels a large and diverse number of research projects. On one hand, the basic technologies (lasers, sensors, bonding, optical interconnect of modules) are being pushed by physicists [KZP99, KG97, HKBH]. Industry is developping technologies to embed optical fibres, splitters etc. into printed circuit boards [LJVN99]. Computer scientists, finally, are developping CAD tools [Fey99] as well as architectures exploiting the benefits of almost unlimited bandwidth between chips [LSDB99].

Implementing such an architecture with the help of the basic technologies is not completely straightforward. In particular, one has to deal with the following non standard issues:

- As long as the number of optical I/O's of a module is not in the thousands one will try to operate the laser/sensor pairs at least an order of magnitude faster than "ordinary" high density gates. This requires an interface of low

density/high speed gates between the optical devices and the slower high density logic.

- The coupling of optical sensors and their amplifiers often shows high pass behaviour. This coupling breaks down if consecutive ones are received for a longer time. Thus one has to artificially interleave zeros into long runs of ones.
- Operating lasers at 320 MHz (with runs of at most 4 consecutive ones) we have measured error rates around 10^{-12} in pure optical data transmission. This would produce an error on each optical fiber roughly every hour. In order to lower the failure rate one has to use an error detecting or correcting code.
- Synchronization of the high speed logic throughout different modules is impractical. This in turn makes the synchronization between high speed logic and the slower high density logic nontrivial.

In order not to burden hardware designers with the above issues it is desirable to develop building blocks for the optical data transmission between chips, which interface in each module directly to the slower high density logic. In this paper we sketch specification and design of such a device. We also report on a joint project between DLR Stuttgart and Saarland University, where we use this device for a prototypical interconnection of a cache chip with a (small) main memory.

2 Integrated memory device with high I/O bandwidth

Optical I/O provides very high bandwidth between chips. With very high bandwidth, one can very quickly copy a block of data with contiguous addresses from a random access memory M located in one chip into a memory M' located in a second chip. This situation is ubiquitous in modern computer systems. A non-exclusive list of examples is shown in table 2.

M	M'
main memory system	buffer memory of an I/O device
buffer memory of I/O devices	buffer memory of I/O devices (in switches)
main memory system	frame buffer of graphics controller
DRAM	L2 cache
L2 cache	L1 cache (on CPU chip)

Table 2: Example applications

This motivates the development of *highly integrated* devices as shown in figure 1. A random access memory M with a very wide internal data bus (d_{int} bits), an address generator for the generation of contiguous addresses and possibly more logic (CPU, cache control,...) is integrated with one or more communication interfaces $I_1, I_2, \ldots$ Each communication interface has to perform the following four tasks:

- serialization of d_{int} electrical signals from the memory bus into d_{opt} outgoing optical signals using high speed logic and lasers.
- Logic for error detection and for breaking up long runs of consecutive ones.
- parallelization of d_{opt} incoming signals using optical receivers and high speed logic.

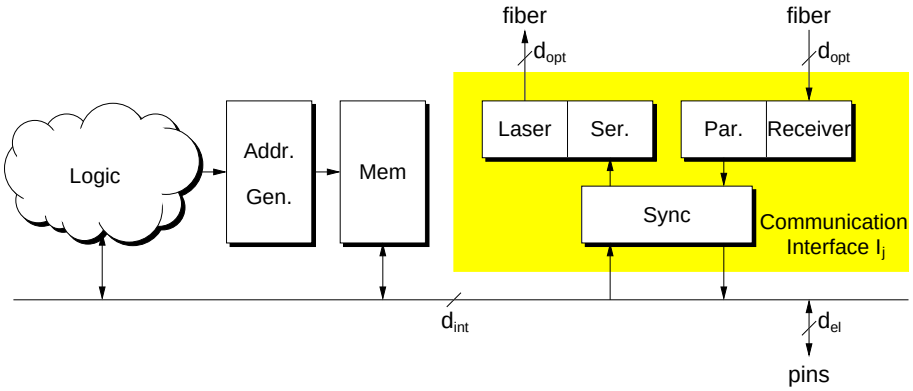

Figure 1: Data paths of the proposed device

- synchronization between internal and external signals.

Presently, the coupling of on-chip memories with optical networks is usually achieved with *external* transceivers like the Siemens Paroli chip family [Sie]. This unfortunately leaves an interface of electrical pins between the chip containing the memory and the transceiver, thus limiting both the width d_{int} and the frequency achievable on the memory bus. In order to overcome the bandwidth limitation of electrical pins, it is essential that the communication interfaces are integrated into the same module as the chip containing the memory.

3 Implementing the Communication Interfaces

Recall that d_{int} is the width of long messages x to be serialized into d_{opt} sequences of messages y each cantaining $e = d_{int}/d_{opt}$ bits of the original message plus extra bits for overhead.

We proceed roughly in the following way:

3.1 Error Detection

Message x is coded with a Hamming code [Ham50] into a message $h(x)$ with a length of at most $D = d_{int} + \lceil \log d_{int} \rceil + 1$ bits. Hamming codes permit the detection of single and double errors.

With an error rate p for single errors, the probability that an entire message x is transmitted with an undetected error is at most

$P \leq 1 - ((1-p)^D + D \cdot p \cdot (1-p)^{D-1} + \binom{D}{2} \cdot p^2 \cdot (1-p)^{D-2})$

if errors occur independently.

3.2 Breaking Runs of Ones

Message $h(m)$ is broken into e messages y of equal length $\ell = \lceil D/e \rceil$. Zeros are inserted after every 4^{th} bit of each message y. Then all messages y are simultanously serialized using high speed logic.

Note that a) computation of the Hamming code and insertion of zeros can be done in ordinary high density logic, b) errors in the transmission of the inserted zeros are trivial to detect *if* one knows, when a transmission starts and c) the combined overhead in all messages y is $0.2 \cdot (d_{int} + \log d_{int})$.

3.3 Serialization

The obvious way to serialize n bits is by a shift register as indicated in figure 2. This requires n flipflops and $n-1$ multiplexors of fast and expensive logic.

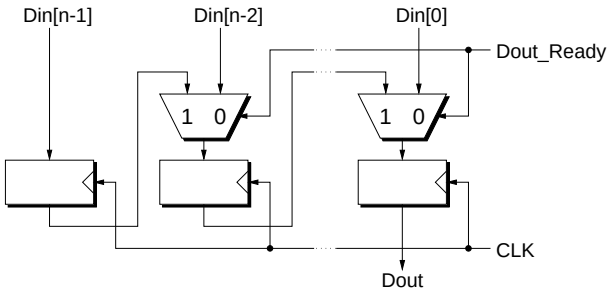

Figure 2: Implementation of a sequencer using a shiftregister

A considerably cheaper way is to use a tree of fast multiplexors controlled by a $\log n$–bit counter as indicated in figure 3. This requires only $n-1$ fast multiplexors and a fast $\log n$–bit counter.

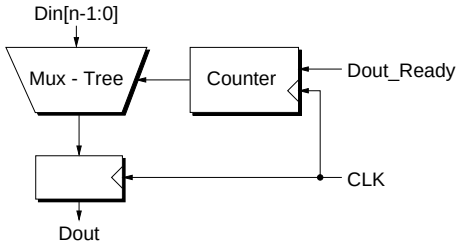

Figure 3: Implementation of a sequencer using a muxtree

3.4 Synchronization

We use in each direction e optical data lines and 1 optical clock line ($d_{opt} = e + 1$), which are synchronized and generated by the sender. Data bits change at the rising edge of the clock. Thus the inverted falling edge of the clock line can be used at the receivers end in order to clock data into a high speed register.

With extra clock lines the presence or absence of transmitted data is trivially indicated by the activation of the clock signal. Errors in the clock line can easily be detected, because on each transmission the line has to be toggled $5\ell/4$ times in consecutive cycles of the high speed logic.

In each module the high speed logic and the slower high density logic are operated asynchronously. This requires in each module two bits, namely

1. data out ready (for the serializer) and
2. data in ready (from the parallelizer)

that are exchanged between high speed logic and high density logic asynchronously. This is done with the usual trick of clocking the signal into two consecutive flipflops which are both clocked by the clock which reads the asynchronous signals [KP95, p239].

4 Chips for Cache and Main Memory

The construction of a cache chip with an I/O–interface as described above is reasonably straight forward. One makes the following two modifications in an existing cache design (e.g. the design in [MP00]):

1. Address bus and data bus, which are usually connected to electrical pins are instead connected to communication interfaces.

2. The bus protocol between cache and main memory is implemented by automata. Protocol and automata have to be modified such that detected errors are dealt with, e.g. by the retransmitting the faulty data.

A corresponding design change has to be made on the side of the main memory.

As part of a joint project between DLR Stuttgart and Saarland University a single chip is under development, which can be configured both as a cache chip and as a plain memory chip. The point of the project is to develop a working prototype of a system using the communication interfaces described above. Putting large memories on such an experimental chip leads to no further insight and increases the production cost dramatically. Therefore, each chip contains only 8Kbit of SRAM. In the cache chip this memory is arranged as 64 cache lines of length 128 Bit.

With 4 such chips a (tiny) main memory capable of storing a total of 256 cache lines can be built.

In our design messages between the cache and main memory have the form $m = (r, a, d)$, where r is a read/write bit, a is the adress of a cache line in main memory (here only 8 bits long), and d is a cache line. Thus the message m is $1+8+128 = 137$ bits long. The correspondig hamming code has only $\lceil \log 137 \rceil = 8$ extra bits. Thus we have $D = 145$ and with $p = 10^{-12}$ the probability of an undetected error is less than $0.5 \cdot 10^{-30}$.

Messages x of length 145 broken into $e = 10$ messages y of length $\lceil 145/10 \rceil = 15$ operating at 320 MHz. The bandwidth in each direction is roughly $10 \cdot 320 Mbit/s = 3.2 Gbit/s$.

For the prototype we are using the 0.8 μm BiCMOS technology from AMS [AMS], which permits CMOS gates and SRAMs as well as high speed ECL gates on a single chip. The development is done as part of the Europractice project using the CADENCE design tools.

5 Multichip Module

Theoretically, the technology of choice to fabricate a module with optical I/O is to fabricate a GaAs chip containing lasers, sensors and amplifiers and to use flip chip bonding between the silicon chip and the GaAs chip. Practically, this technology is too costly for an experimental device.

We are using discrete lasers, sensors and amplifiers which are combined with the cache/memory chip to a module as indicated in figure 4.

6 Optical Interconnect between Cache and Main Memory

In a prototype system a module with a cache chip as well as modules with 4 memory chips are combined with a processor and a host computer as indicated in figure 5.

On the way from the cache to the memory chips, the optical signals have to be broadcast. On the way from the main memory chips the transmited signals

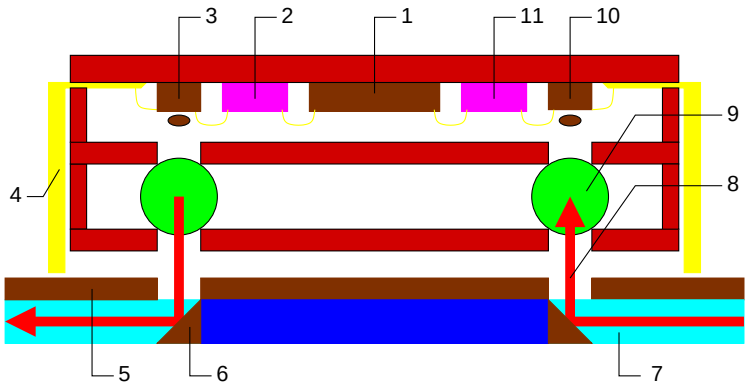

Figure 4: Opto-electronic multichip module

1: Cache chip with sequencer/parallelizer, 2: Laser driver, 3: VCSEL
4: El. feed cable, 5: El. contact plane, 6: Mirror, 7: Opt. linkage plane
8: Opt. signal, 9: Ball lens, 10: PIN diode, 11: Amplifier

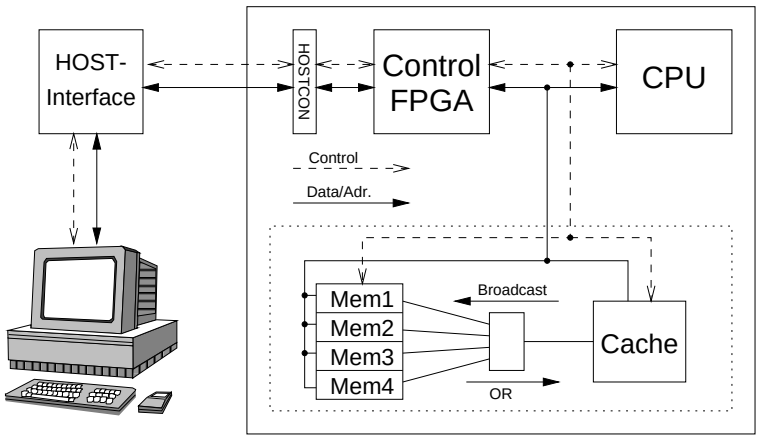

Figure 5: Prototype

are ORed together (memory modules, which are not sending data, transmit zeros). This is achieved with the help of beam splitters ond combiners in an integrated optical system.

7 Acknowledgements

For inspiring discussions the authors thank M. Bosch, C. Jacobi, S. M. Müller and H. Opower.

References

[AMS] AMS – Austria Mikro Systeme International AG. 0.8 μm BiCMOS Process Technology.
`http://www.amsint.com/products/technology/index_b08.html`.

[Fey99] D. Fey. Rechnergestützter Entwurf von Photonischen VLSI-Schaltkreisen mit regulär angeordneten Detektoren. In *Tagungsband Workshop Optik in der Rechentechnik ORT 1999*, pages 88–93, Jena, Germany, Oct 1999.

[Ham50] R. W. Hamming. Error detecting and error correcting codes. *The Bell System Technical Journal*, 29(2):147–160, April 1950. Reprinted in E. E. Swartzlander, *Computer Arithmetic*, Vol. 2, IEEE Computer Society Press Tutorial, Los Alamitos, CA, 1990.

[HKBH] L. Hoppe, J.M. Köhler, H. Bartelt, and B. Höfer. Zweidimensionales Faserarray und Verfahren zu seiner Herstellung. Anmeldung beim Deutschen Patentamt, Nr. 199 25 015.4.

[KG97] A.V. Krishnamoorthy and K.W. Goosen. Progress in optoelectronic-VLSI smart pixel technologie based on GaAs/AlGaAs MQW modulators. In *Int. Journal of Optoelectronics 11*, pages 181–198, 1997.

[KP95] J. Keller and W.J. Paul. *Hardware Design: Formaler Entwurf digitaler Schaltungen*, volume 15 of *Teubner Texte zur Informatik*. Teubner, Stuttgart;Leipzig, 1995.

[KZP99] K. Kieschnick, H. Zimmermann, and P.Seegebrecht. Fast BiCMOS receiver OEIC for short-range optical data transmission. In *Tagungsband Workshop Optik in der Rechentechnik ORT 1999*, pages 57–62, Jena, Germany, Oct 1999.

[LJVN99] S. Lehmacher, J. Jankowski, C. Vavitsas, and A. Neyer. Integration von polymerem Multimode-Wellenleitern in konventionelle Multilayer-Platinen. In *Tagungsband Workshop Optik in der Rechentechnik ORT 1999*, pages 31–33, Jena, Germany, Oct 1999.

[LSDB99] P. Lukowicz, S. Sinzinger, K. Dunkel, and H.D. Bauer. Design of an opto-electronic VLSI/parallel fibre bus. In *J. Opt. A: Pure Appl. Opt. 1 1999*, page 367ff, 1999.

[MP00] S.M. Müller and W.J. Paul. *Computer Architecture: Complexity and Verification*. Springer Heidelberg, 2000.

[Sie] Siemens AG. Parallel Optical Links – PAROLI Family. `http://www.infineon.com/products/fiber/index_paroli.htm`.