Optical versus electronic bus for Address-transactions in future SMP architectures

Wissam Hlayhel (a), Daniel Litaize (a), Laurent Fesquet (b), Jacques H. Collet (b),

(a) IRIT, 118 route de Narbonne, 31062 Toulouse Cedex 4, France.(b) LAAS-CNRS, 7 avenue du Colonel Roche, 31077 Toulouse Cedex 4, France.

Abstract

The fast evolution of processor performance necessitates a permanent evolution of all the multiprocessor components, even for small to medium-scale symmetric multiprocessors (SMP) build around shared busses. This kind of multiprocessor is especially attractive because the problem of data coherency in caches can be solved by a class of snooping protocols specific to these shared-bus architecture. But the bandwidth demand, especially for the addresses, is becoming so important that a technological step must be considered. Optical communications are becoming mature, and bring a huge information bandwidth through the implementation of optical busses. This paper is focused on the address bandwidth needed by shared-bus SMP without suggesting a complete solution. We show that an optical address bus can fulfill the bandwidth demand of future SMPs contrarily to standard electronic busses.

1. Introduction

Symmetric multiprocessors (SMP) dominate the server market and are becoming common in that of desktops. A global physical address space and a symmetric access to the whole main memory by any processor offers increased flexibility and programming facilities as a variety of programming models can be efficiently used. Each processor has its own hierarchy of caches and a shared bus is mostly used to interconnect the processors and the memory modules in small to medium-scale multiprocessors. In the shared-bus architecture, all the access requests to data in memory can be observed by all the modules, allowing each processor to snoop the bus for maintaining the coherency of data across the caches. The main scaling limit of this architecture originates in the limited bandwidth of the shared bus which cannot follow the increasing demand due to the fast evolution of the processor performances [1]. Architectural solutions must be found to increase the bandwidth. For instance, splitting the address and data phases of the transactions enables a better use of busses. The data bandwidth can be increased by enlarging data busses, by multiplying their number, by connecting them to the memory modules through multiplexors or crossbars. Scaling the data bandwidth is easier than scaling what is called the snoop bandwidth, linked to address phase. Multiple-address busses can be also used in a static way, each bus being allocated to an address range. But the bandwidth remains insufficient because current electrical busses which connect several tens of printed boards have an operation frequency lower than 100 MHz and because no dramatic frequency increase is expected. Thus, a new emerging technology based on optical busses could be the solution to overcome the present limitation. We stress that transmitting data at 5-10 GHz is usual in the telecommunication world because transmissions in optical fibers or wave guides are free of load adaptation, reflection or capacity problems and are mainly limited by the complexity of the opto-electronic interfaces. The high-speed electronic circuits of the interface reduce essentially the multiplexor and demultiplexor which can operate nowadays in the 10-GHz frequency range [2]. In this paper we only consider the address phase and the use of an optical shared-bus is analyzed in this context. As we will see in section 3, the structure of this bus is intermediate between a shared bus and a ring.

Considering an ideal memory, an ideal data network, we used the SPLASH-2 suite for studying the bandwidth needed by the address bus of a SMP, possibly including up to 64 Instruction-Level Parallelism (ILP) processors. The paper is organized as follows: Sections 2-3 respectively summarize the state of the art of shared-bus architectures and the operation conditions of optical busses. Section 4 introduces the simulation parameters. Simulation results are discussed in section 5. We show in section 6 that the bandwidth requirements of the next-generation ILP SMPs¹ could be fulfilled by optical busses. The conclusion includes some insights for future investigations.

2. Shared-Bus Architectures

The processor speed, which has been increased by a factor ten in a few years, has induced a huge evolution of bandwidth needs as can be shown by analyzing the

¹ We abbreviate with the notation "ILP SMP" the expression "SMP built with ILP processors".

evolution of some shared busses over the last five years. The XDbus from SUN [3] is a 64-bit multiplexed address-data bus running at 40 MHz (10 slots) which involves a split-transaction protocol. Each bus transaction requires 2 cycles for an address transfer and 9 cycles for a cache block transfer. The raw bandwidth is 320 MB/s, but the effective bandwidth is only 34 of this value. In the SGI Power Challenge [4], the splittransaction protocol is enhanced to take into account the presence of two busses clocked at 47.6 MHz (13 slots), one of 40 bits dedicated to the address transfer and another one of 256 bits used for the transfer of cache blocks. The two busses are synchronized and enable to transfer in 5 cycles the address or the data. The busses have the possibility to support up to 8 outstanding read requests. The raw bandwidth is 1.5 GB/s but the effective bandwidth reduces to 4/5 of this value. The Sun Enterprise 6000 [5] is also a non-multiplexed splitbus running at 83.5 MHz with 256 bits for the data and 41 bits for the address. This high operation frequency (for a shared bus) is due to a special backplane design (so-called "Centerplane" technology) where cards are connected by both sides (8 slots per side). The raw bandwidth is 2,6 GB/s and the number of outstanding read requests that can be in progress at the same time is 112 (up to 7 from each board, each processing board containing two processors). In recent SMP designs, such as the G30 machine from IBM or the 10000 from SUN [5], the data network is a crossbar, but the address network remains a shared bus used for maintaining coherency. Above a certain number of processing elements, multiple address busses are needed. For example, two XDbus were implemented in the SparcCenter 2000 to attain a maximum number of 30 processors and four in the Enterprise 10000 to attain 64 processors. The most recent solutions for data transmission seem to be some kind of crossbar switch. With a smart design of this switch, direct cache to cache blocks transfers may become feasible without updating the main memory. The open problem at the moment is the address/snoop bandwidth, that could be solved by means of a technological step, as presented in the next paragraph.

3. Optical Bus Architecture

The optical bus [6,7,8,9,10,11,12] is a new emerging technology that enables to increase the bandwidth of busses. The operation of an optical bus is based on the two following paradigms:

1. All the processors can simultaneously insert (optical) addresses into the bus, thus dropping the constraint of access serialization and internode arbitration encountered in the operation of standard electronic busses. Notice however the price to pay for this easing of the bus access conditions: Inserting pulses in a multi-point line implies that the spatial extension of the pulses remains shorter than the distance separating two access points. As the propagation time of the light in optical fibers is 50 ps/cm, it is clear that connecting processors separated by a few centimeters requires optical pulses with a duration of the order of one hundred ps, or shorter! This is in fact the capacity of the new optical technologies to easily generate such pulses that enables considering a multi-access bus.

2. The addresses (i.e., the optical cell) freely propagate along the optical bus. There is no transient storage of the optical address at each node. On the contrary, each node gets (on the fly) a copy of the bypassing cell and this copy is processed without delaying the propagation along the bus. We must emphasize the underlying physical reasons that enable this operation mode: A multipoint optical line, even operating at several tens of GHz, as suggested by the pulse duration previously considered, exhibits almost no impedance mismatch problems, no reflections troubles when compared to the electronic solutions. Moreover, there is no capacity effects in the line and the noise immunity is extremely high. Bit error rates lower that 10⁻¹⁶ have been reported in telecommunications even for propagation distances over several hundred meters at several Gb/s.

Figure 1 shows the architecture of a typical folded optical bus connecting 3 processors to 3 memory banks (mem1, mem2, mem3).



Figure 1: Structure of a folded optical bus. D is an optical delay line.

The different nodes are linked by a ribbon of optical fibers. The width of the optical bus (not shown for clarity in the figure) is that of the address plus several control bits including one for the clock and one for the presence detection. The role of the optical clock is twice: 1) It provides a time reference for each node; 2) It is used to locally generate the new optical address cell that is inserted in the bus (this point will be clarified in the next figure). The fundamental consequence is that one single optical pulse source (the clock generator) creates all the optical pulses emitted in the bus avoiding several critical problems related to the synchronization, jitters and the wavelength dispersion. Another advantage is that the parallel propagation of optical pulses in fibers is skew-free for the bus length of a few meters we are considering.

The address insertion process is the following: Each node permanently snoops the presence bit. If no presence bit is detected at clock time, one address may be inserted simultaneously to a presence bit. This process requires to add a small optical delay in the line (see block D in Figure 1) to compensate the trigger time of the local electronic circuits. We stress that:

- Driving transmissions typically between 1 and 5 GHz/s is easy. Thus, a 32-bit wide optical ribbon may provide an equivalent address transfer rate, typically 2 orders of magnitude larger than that of electrical busses.
- The bus is folded so that no cell extraction is required! Cells are automatically killed when they reach the end of the folded bus. Each node gets a copy on the fly.

We show in Figure 2, a draft of the internal node structure, especially the optical interfaces which are monolithic optical circuits with light propagating in optical guides.



Figure 2: Structure of the interface of the optical bus with the electronic node

We may distinguish:

- The demultiplexer stage in the upper branch which directly transmits the optical pulses to photodiodes interfaced to a serial/parallel converter. This approach enables to slow down the operation frequency of the electronic circuits, except for the electronic serial/parallel conversion which must operate at the optical bus frequency that may be as high as several GHz (10 GHz as showed in [2]).
- The multiplexer stage, in the lower branch which operates as follows: The clock pulse is split and simultaneously injected into all the optical switches. The clock and the presence bits control the electronic interface, or in other words, the encoding of the address bits which are emitted by the different switches. This way, all the optical multiplexers of the different nodes work with one single optical generator which is the optical clock.

The multiplexer stage includes Y splitters and optical switches. The switch may be an integrated modulator or an All-Optical Amplifiers (AOA). All these basic

elements have been already demonstrated in numerous laboratories, mostly based on the InP technology for optical communications at $1.55 \,\mu m$. 1-to-16 splitter/amplifiers have been demonstrated [see for instance refs. 13, 14, 15, 16, 17, 18 and the numerous references therein]. Optical modulators may operate at several tens of Gb/s. The control of AOA is limited to 5-10 GHz for physical reasons.

Let us consider an example to show the specificity of the optical bus operation. We assume that the bus operates at 2 GHz with 200 ps pulses [19]. In terms of spatial description, this means that the optical pulses propagate along the bus with an extension of 4 cm and separated by 10 cm (the velocity of light in fibers is 0.2m/ns). The **optical** distance that separates two access points (i.e., 2 multiplexers in Figure 2) must also equal 10 cm (or a multiple of that distance) to ensure the simultaneous insertion of addresses at clock times. Notice that the physical distance which separates two nodes may be shorter. The 10 cm we consider here may be split, for instance, in 2 cm for the physical internode distance and 8 cm (i.e., 400 ps) for the intranode delay (bloc D in Figure 1). As the internode propagation time is 0.5 ns, the latency for 64 nodes will be about 32 ns !!! This value should not be compared with that of an electronic bus because in our example up to 64 address transactions could be simultaneously in progress whereas only one transaction is possible for an electronic bus after the arbitration process.

As addresses can be checked by all the processors, some kind of modified snooping protocol ought to be considered. Even if the structure of the bus resembles that of a ring, a major difference is that cells are not latched in a node. It is therefore not possible to check some directory before forwarding the request, as done in classical ring architectures [20, 21]. An optical ring could also be used to transfer addresses and data blocks as suggested in [11]. But neither ring architecture nor data coherency are the topics of this communication. The question investigated in the next sections is: How might be used the address bandwidth provided such an optical bus by the future ILP SMPs?

4. Experimental Methodology

Previous studies have investigated the effect of the bandwidth in the distributed shared-memory multiprocessors [22] and compared the bandwidth provided by different networks for this kind of architecture [23]. Most of the studies considered previous-generation microprocessors. Recent studies on ILP multiprocessors demonstrate that the parallel efficiency for these architectures is lower than that with previous-generation multiprocessors. Thus, ILP multiprocessors need a greater bandwidth and effective latency reducing techniques [24]. But to our knowledge, no study has so far evaluated the address bandwidth required to build efficient small to medium-scale ILP SMPs. We describe in this section our methodology to investigate the bandwidth needed for transmitting address requests². Basically, we model a physically shared-memory ILP symmetric multiprocessor assuming that the network and the memory are contentionfree. This assumption is useful to capture the maximum number of requests that can be emitted per cycle³ and to let the processors exploit their maximum ILP capabilities.

4.1 Simulation Environment

We modeled the SMP architecture with an instruction-driven simulator that we developed because most of those available when we started this study were designed for distributed-memory architectures and/or did not support ILP processors. It simulates in detail, cycle by cycle, the instructions of workloads and emulates the I/O UNIX functions to verify correctness of results. We describe in the following the memory hierarchy models.

Processor: The processor is a generic ILP processor implementing the state of the art of advances: dynamic scheduling, non-blocking reads, register renaming and speculative execution. The processor supports 8-way issues per cycle. The configuration of functional units and issue/retire buffers were taken from [25]. We did not assume a single-cycle latency for all functional units because they contribute to stall time when the network and the memory are contention-free. Thus, our latencies are realistic contrarily to simplifying assumptions used in some other studies to speed-up simulation.

Caches: Each processor has two cache levels, namely L1 and L2. The L1 cache is direct-mapped, write through with a one-cycle access time. It interleaves the sets of cache with 4 banks [26] to serve up to 4 requests per cycle. The L2 cache is 4-way associative with write-back policy. It is also organized in 4 interleaved banks allowing up to 4 access requests (namely, the misses from L1 and update-responses to other processors). The access time to L2 amounts is of 4 cycles. Both caches are lock-up free and allow 4 pending misses per bank (16 misses at all). They support buffering to solve bank contentions. Cache line size is 64 bytes in L1 and L2. The instruction cache is not modeled in our simulator. We assume that all instructions hit in one cycle to speed up simulation time. This is realistic because we used as workload scientific applications that have a high instruction hit ratio.

The cache coherency protocol is identical to the Illinois Protocol [27], except that cache transfers are only used for dirty data. This restriction is not too severe as, for simplicity, most of the implementations of this protocol do not support this capability. Each bloc has four states: Modified, Exclusive, Shared and Invalid. The Architecture is release-consistent, no

³ We abbreviate the term processor cycle with cycle

ordering of the memory access is forced outside the synchronization zones.

Network and memory: We assume that the network and the memory are contention-free. Thus, the network transmits at each cycle all the requests. The Latencies for memory access and for network data transfer are modeled with constant values. The miss latency depends on where data is fetched from. If it is fetched from the main memory, the latency is the time necessary to retrieve data from memory plus the data transfer latency. If it is retrieved from another dirty block in a cache, the latency is equal to the pending time to access the remote cache (if busy), plus 4 cycles of data fetching, plus the data transfer latency.

We assumed some simplifications concerning coherency. All the requests are fulfilled at each cycle and an atomic bus-like mechanism is implemented to serialize the accesses to the same address.

4.2 Workloads

We used as workload 4 applications of the SPLASH-2 suite [28], namely LU, FFT, Ocean and Radix. These applications are frequently used in studies and less computation time-consuming than other ones in the suite. Table 1 shows the input sizes for each of these applications.

To avoid pollution of results due to the creation of processes and cold misses, we started capturing statistics after the initialization phase of the application. Modeling was executed after the initialization phase is executed on a quicker simulator which only maintained the caches and the context (registers and memory) logically consistent. This way, we divided our simulation time up to a factor 2.

| Application | Input Size | |
|----------------------|------------------------|--|
| LU | 512x512 matrix, bloc 8 | |
| FFT | 64K points | |
| Ocean | 258x258 ocean | |
| Radix | 1M keys, radix 1024 | |
| Table 1: Input sizes | | |

The cache size was related to the input size of applications. As recommended in the reference [28] on the SPLASH-2 suite, primary working-sets fit in the L1 cache, and secondary working-sets do not fit in the L2 cache. Since we simulate different numbers of processors, different cache sizes in each case were considered because the secondary working-sets decrease versus the number of processors when keeping unchanged the input sizes. The size of caches is reported in the following table versus the number of processors.

| Number of processors | L1 cache size | L2 cache size |
|----------------------|---------------|---------------|
| 16 Processors | 32K | 128K |
| 32 Processors | 16K | 64K |
| 64 Processors | 8K | 32K |
| | | |

Table 2: Cache sizes versus the number of processors

² An address request is a request which needs an address transaction. It may be a miss, an invalidation or a flush to memory. Flush from one cache to another without memory update is not considered as (an address request. It is similar to a memory response to a miss. We abbreviate the term "address request" with request.







Figure 3: Distribution of request, 3-17 latency

5. Results of Simulation :

We changed two key parameters, namely the number of processors and the data latency. Varying the first parameter is important to show how the bandwidth requirement scales with the number of processors. The second parameter shows how the difference in the operation frequency between the processors and memory/network components affects the bandwidth requirement.

5.1 Traffic characterization

We studied here the traffic dynamics for each application using the contention-free network. We simulate 16, 32 and 64 processors, first assuming 3 cycles for the data transfer and 17 for the memory fetch. This investigation is useful for determining the bandwidth needed by the applications and for understanding the increase of the execution time induced by any bandwidth reduction. To determine the bus traffic dynamics, our simulator traces and records the number of requests emitted at each cycle during the whole execution of the application. We get therefore millions of points that cannot be simply displayed in a graph. Thus, we chose to partition the data set in adjacent subsets with the same number of points and to display the average request number per subset. This way, we smooth the traffic curves but our tests show that 200 points are generally sufficient to observe the stable burst zones as displayed in figure 3. Increasing the number of points by a factor 10 adds new small narrow peaks to the traffic curves without changing the global behavior. The curves show the irregularity of the request traffic. For FFT, the majority of the requests are concentrated within three burst zones. For Ocean, the requests are distributed in a larger number of zones. These curves also show that the traffic augmentation versus the number of processor is especially important for some applications such as Radix. This effect depends on the scale of the communication pattern and of the working-set of each application.

We also simulated the traffic considering a latency of 10 cycles for the data transfer and 40 for memory fetch. we also found a similar traffic behavior, the same alternances of burst zones accompanied by a traffic reduction. The increase of the latency delays the initiation of new requests due to the dependency of data and resources. But the traffic reduction is not proportional to the latency increase because ILP processors may initiate new requests without waiting for the completion of those already in progress.

We shall use the condensed notation 3-17 and 10-40 latency to design the latency conditions of these simulations.

5.2 Bandwidth Requirements

In the contention-free model of the previous paragraph, the network can serve any number of requests per cycle, therefore leading to the minimum execution time for any application. But obviously, this time is expected to increase if the network cannot serve all the requests, or in other words, if the network bandwidth is limited. Thus, we define the bandwidth requirement for an application as the minimum bandwidth which does not increase the execution time by more than 5% with respect to that obtained in the contention-free model. In the following, we shall use the bandwidth notation RxC where R is the number of requests per C cycles. The following tables show the minimum bandwidth that we deduced from simulation for the different applications.

| | 16 Processors | 32 Processors | 64 Processors |
|--|---------------|---------------|---------------|
| LU | 1x10 | 1x2 | 2x1 |
| FFT | 1 | 2x1 | 3x1 |
| Ocean | 1 | 2x1 | 3x1 |
| Radix | 1 | 3x1 | 7x1 |
| T 11 2 D 1 111 1 1 1 2 2 3 171 | | | |

Table 3 : Bandwidth requirement for 3-17 latency

| | 16 Processors | 32 Processors | 64 Processors |
|-------------|---------------|---------------|---------------|
| LU | 1x10 | 1x2 | 1 |
| FFT | 1x2 | 1 | 2x1 |
| Ocean | 1x2 | 1 | 2x1 |
| Radix | 1 | 2x1 | 5x1 |
| m 11 | (B 1 1 1 1 | | 0 10 1 |

Table 4 : Bandwidth requirement for 10-40 latency

Notice that the traffic curves in figure 3 provide an estimate of the minimum bandwidth (in request/cycle) in good agreement with the calculations. For instance, a bandwidth of 1 request/cycle is an estimate of the minimum bandwidth to run Radix with 16 processors.

5.3 Effect of Bandwidth Limitation on Execution-Time

We used a simple analytic model to study how the bandwidth limitation affects the execution time. Suppose we have the traffic distribution shown in figure 4. The natural effect of limiting the bandwidth to B requests/cycle (see fig. 4) is an extension of execution time proportional to the gray area found above the horizontal line B.



This relative extension is estimated with the following formula (Eq. 1):

$$EXT = \sum_{\substack{i=1,N\\AV[i] > B}} (AV[i] - B) / B$$

The summation runs over all the points having an average traffic AV[i] above the limited bandwidth B. N



is the number of averages to represent the traffic as

corresponding to a bandwidth of 0.33 Giga-req/s.

Figure 5: Degradation of the execution time due to bandwidth limitation

(e)

1x2

1x1

2x1

3x1

0%

1x5

1x4

1x3

defined in paragraph 5.1. The relative increase of the execution time (which is determined by the expression 100xEXT/N) is reported in figure 5. The curves show that the increase of the execution time is not a linear function of the bandwidth. For example, Radix in figure 5(c) exhibits a clear inflection point at the abscissa 2x1. The behavior of Ocean is similar in figure 5(c). We simulated some points of these curves using a limited network model. It turns out that our formula (Eq. 1) is very accurate, leading to discrepancies with simulated results ranging from 1 to 2%.

1x4 1x3 1x2

6. A case study :

0%

1x10 1x9 1x8 1x7

1x6 1x5

(d)

We wish to show in this section how optical solutions may mach the bandwidth requirement. We introduce some realistic assumptions on the specifications of the next generation of multiprocessor architectures. We consider a system built with a) 8-issue ILP processors clocked at 333MHz; b) DRAM memories with a latency of 50 ns, so equivalent to 17 processor cycles; c) an electronic bus operating at 100MHz and 3bus cycles necessary for an address transaction. We only consider Ocean which is an application that has great bandwidth needs.

With 16 processors, the bandwidth required by Ocean is 1 req/cycle as concluded in Table 3. With our hypothesis, the processor cycle is equal to 3ns, thus Similarly, the bandwidth required is 0.66 Greq./s. and 1 Greq./s. with 32 and 64 processors respectively. These values are lower than the 2 Greq/s possible with optical busses. Even Radix, the most bandwidth-consuming application might be supported with no significant increase of the execution time (less than 10% as shown in figure 5(c)).

(f)

0%

1x3 1x2 1x1 2x1 3x1 4x1 5x1 6x1

Now let us examine how electronic busses degrade performances. With 16 processors and one bus, the provided bandwidth is equivalent to 1 request / 10 processor cycles (one request each 3 x 10 ns). Figure 5(a) shows an increase of the execution time greater than 200%. If we consider 2 busses with 32 processors, the available bandwidth is 1 request / 5 cycles, thus leading to an increase of 250% of the execution time (figure 5(b), at abscissa 1x5). For 4 busses with 64 processors, the degradation reaches 270% and 450% for the Radix application (figure 5(c) between the abscissas 1x2 and 1x3).

As processor performance evolves faster that than of memories, we consider in a second example a memory latency reduced to 40ns and future processors operating at 1 GHz. So, the memory latency is equivalent to 40 processor cycles. In this case, the bandwidth required by Ocean with 64 processors equals 2 Greq./s (see Table 4). An optical bus operating at 2 GHz still satisfies this requirement, but it becomes difficult to also support heavy applications such as Radix which requires 6 Greq./s. However, as the advances in

optoelectronics are permanent [19], we expect that the bandwidth of the optical bus will scale with the processor speed.

7. Conclusion :

With the growing performance of ILP processors, small to medium-scale SMPs will continue to be of high interest, provided that the processor-memory network remains able to match the communication needs. This paper reports quantitative measurements on the bandwidth that will be needed by future ILP SMPs. This study was mainly focused on the address transfer problem considering an ideal memory and data network. The results show the interest for the optical bus which is a technological step adapted for the huge address bandwidth needed by the next generation of ILP SMPs. As billions of address-transactions per second are to be considered, the directory cache-check mechanism must be carefully studied, in conjunction with the cache coherency protocol, which can not be simply adapted from the Illinois protocol. Further investigations are also necessary to solve new problems such as the structure of the network for data transfers and the architecture of the main memory.

References

[1] Albert Yu, "The Future of Microprocessors". MICRO-IEEE, December 1996.

[2] Vitesse SemiConductor Corp., Communication Products Databook 1996, VSC8071/8072 10Gbits/s 16 bits mux/demux chipset]

[3] Pradeep Sindhu, Jean-Marc Frailong, Jean Gastinel, Michel Cekleov, Leo Yuan, Bill Gunning and Don Curry. "XDBus: A High-Performance, consistent, Packet-Switched VLSI Bus". In proceedings of COMCON, Spring 93.

[4] Mike Galles and Eric Williams. Performance Optimisations, Implementation and Verification of the SGI Challenge Multiprocessor. Proceedings of 27th Annual Hawaï International Conference on Systems Sciences, January 1993.

[5] Technical Report:www.sun.com/datacenter/products

[6] M. Feldman, S. Esener, C. Guest, and S. Lee, Comparison between optical and electrical interconnects based on power and speed considerations. Appl. Opt. , 27:742-751 (1988).

[7] J. Jahns and Murdocca ; Crossover Networks and their Optical Implementation Appl. Opt. 27:3155-3160 (1988).

[8] J. Sauer ; A multi-Gb/s optical interconnect SPIE, Proc. Digital Opt. Comp. II 1215:198-207 (1990).

[9] D.M. Chiarulli, S.P. Levitan, R.G. Melhem, M. Bidnurkar, R. Ditmore, G. Gravenstreter, Z. Guo, C. Qiao, M.F. Sakr, a,d J.P. Teza; Optoelectronic Busses for High-Performance Computing Proceedings of IEEE, 82:1701-1710 (1994)

[10] Z. Guo, R. Melhem, R. Hall, D. Chiarulli, and S. Levitan Array processors with pipelined optical busses J. Parallel Distributed Comput. vol 12, 269-282, (1991)

[11] Laurent Fesquet, Jacques Collet, Low latency optical bus for multiprocessor architecture, International Conference on Applications of Photonics Technology 1996, july 29th-august 1st 1996 Montreal Canada, pages 189-194

[12] Laurent Fesquet, Jacques Collet, Rainer Buhleier, Chapter: Low latency asynchronous optical bus for distributed multiprocessor

systems, Optical Interconnections and Parallel Processing: the interface, editor Kluwer

[13] U. Koren, M.G. Young, B.I. Miller, M.A. Newkirk, M. Chien, M. Zimgibi, C. Dragone, B. Glance, T.L. Kock, B. Tell, K. Brown-Gorbeler, and G. Rraybon 1x16 photonic switch operating at 1.55 • m wavelength based on optical amplifiers and a passive optical splitter. Appl. Phys. Lett. Vol 61, 1613-1615, (1992)

[14] M. Gustavsson, B. Lagerström, L. Thylen, M. JansoStolz, L. Lundgren, A.C. Mörner, M. Rask, and B. Stolz Monolithically integrated 4x4 INGaAsP/InP laser amplifier gate switch arrays; Electron. Lett. Vol 28, p 2223-2225, (1992)

[15] G. Glastre, D. Rondi, A. Enard, E. Lallier, R. Blondeau, and M. Papuchon Monolithic integration of 2x2 switch and optical fibers with 0 dB fiber to fiber insertion loss grown by LP-MOCVD Electronics Letters Vol 29, 124? (1993)

[16] F. Ratovelomanna, N. Vodjdani, A. Enard, G. Glastre, D. Rondi, and R. Blondeau Active Lossless Monotithic One-by-four Splitters/Combinerss using optical Gates on InP IEEE Photonics Technology Letters vol 7, nb 5, 511-513, (1995)

[17] E. Jahm, N. Agrawal, M. Arbert, H.J. Ehrke, D. Franke, R. Ludwig, W. Pieper, H.G. Weber, and C. Weinert 40 Gb/s all_optical demunitplexing using monithically integrated Mach-Zehnder interferometer wiht SOA; Elect. Lett. Oct 1995, 1857-1858

[18] R. Krähenbühl, R. Kyburz, W. Vogt, M. Bachmann, T. Brenner, E. Gini, and H. Melchior Low loss Polarization-Insensitive InP-InGaAsP optical space switches for fiber optical communications; IEEE Phot. Technology Lette. Vol 8, 632-634, (1996)

[19] Transmitters and receivers, FibreSystems, march/april 1997, page 39

[20] Barroso L.A., Dubois M. et al, the Performance of Cache Coherent Ring-Based Multiprocessors. Technical Report: CENG-92-19. Department of Electrical Engineering-Systems, University of Southern California, November 1992.

[21] Z.G. Vranesic, M. Stumm, D.M. Lewis and R. White, Hector: A Hierarchically Structured Shared-Memory Multiprocessor. IEEE computer, January 1991, pp 72-79.

[22] C. Holt, M. Heinrich, J. Pal Singh, E. Rothberg and J. Hennessy —The Effects of Latency, Occupancy and Bandwidth in Distributed Shared Memory Multiprocessors— Technical Report : CSL-TR-95-660, January 1995, Computer Systems Laboratory-Stanford University.

[23] U. Rajagopalan, "the Effect of Interconnection Networks on the Performance of Shared-Memory Multiprocessors". Master's thesis, Rice University, 1994.

[24] V.S. Pai, P. Ranganathan, and S.V. Adve —The Impact of Instruction-Level Parallelism on Multiprocessor Performance and Simulation Methodology— Third IEEE Symposium on High-Performance Computer Architecture 97.

[25] S. Jourdan, P. Sainrat and D. Litaize —Exploring Configuration of Functional Units in an Out-Of-Order Superscalar Processor— ISCA 95.

[26] Gurindar S. Sohi and Manoj Franklin, "High Bandwidth Data Memory Systems for superscalar processors", Computer Sciences Department, University of Wisconsin-Madison. 1991 ACM 0-89791-380-9/91/0003-0053.

[27] Mark Papamarcos and Janak Patel. —A Low Overhead Coherence Solution for Multiprocessors with Private Cache Memories—, ISCA84.

[28] S.C. Woo et al., The SPLASH-2 Programs : Characterization and Methodological Consideration. ISCA95.