SYMNET: an optical interconnection network for scalable high-performance symmetric multiprocessors

Ahmed Louri and Avinash Karanth Kodi

We address the primary limitation of the bandwidth to satisfy the demands for address transactions in future cache-coherent symmetric multiprocessors (SMPs). It is widely known that the bus speed and the coherence overhead limit the snoop/address bandwidth needed to broadcast address transactions to all processors. As a solution, we propose a scalable address subnetwork called symmetric multiprocessor network (SYMNET) in which address requests and snoop responses of SMPs are implemented optically. SYMNET not only has the ability to pipeline address requests, but also multiple address requests from different processors can propagate through the address subnetwork simultaneously. This is in contrast with all electrical bus-based SMPs, where only a single request is broadcast on the physical address bus at any given point in time. The simultaneous propagation of multiple address requests in SYMNET increases the available address bandwidth and lowers the latency of the network, but the preservation of cache coherence can no longer be maintained with the usual fast snooping protocols. A modified snooping cache-coherence protocol, coherence in SYMNET (COSYM) is introduced to solve the coherence problem. We evaluated SYMNET with a subset of Splash-2 benchmarks and compared it with the electrical bus-based MOESI (modified, owned, exclusive, shared, invalid) protocol. Our simulation studies have shown a 5–66% improvement in execution time for COSYM as compared with MOESI for various applications. Simulations have also shown that the average latency for a transaction to complete by use of COSYM protocol was 5–78% better than the MOESI protocol. SYMNET can scale up to hundreds of processors while still using fast snooping-based cache-coherence protocols, and additional performance gains may be attained with further improvement in optical device technology. © 2003 Optical Society of America

OCIS code: 200.4650.

1. Introduction

Symmetric multiprocessors (SMPs)[1–3] dominate the server market as the most prevalent form of parallel processing commercially available. In SMPs, each address request is broadcast to all processors/memory modules by use of a shared bus. This address request is snooped by all the processors enabling a simultaneous update or an invalidation of cache blocks,[1,4,5] thereby maintaining the cache's coherency with low latency. As the number of processors grows in the system, contention to acquire the bus also increases. The evolution of faster processors further aggravates the situation because the shared bus cannot run at speeds comparable to that of faster processors. As the bus speed increases, the processor boards connected to the bus behave as stubs resulting in reflections[3] of bus signals. Other fundamental problems, such as impedance mismatch, stray capacitance, ΔI noise, and crosstalk[3,6] significantly affect the speed improvements of the shared bus. These problems limit the operating speed of the shared buses to 150 MHz[2] in current systems. Therefore the bus speed and the coherence overhead limit the rate at which address requests can be broadcast to all the processors/memory modules.[7,8] This in turn limits the number of processors that can share the bus by affecting the scalability of SMP systems.[8] This address rate/bandwidth is the main scaling limit, which cannot keep pace with the increasing demands of faster and multiple processors, on the scalability of shared-bus based SMPs.

A. Recent Electrical Solutions

To increase the address bandwidth, several techniques have been introduced. These techniques in-

The authors are with the Department of Electrical and Computer Engineering, University of Arizona, Tucson, Arizona 85721. E-mail address for A. Louri is louri@ece.arizona.edu.

Received 12 February 2003; revised manuscript received 4 March 2003.
0003-6935/03/173407-11$15.00/0

© 2003 Optical Society of America

clude split-transaction buses,[9] multiple-address buses,[8] physically separate address and data subnetworks,[8] and moving from physically shared buses to logical buses that are implemented as point-to-point links.[8] More aggressive and expensive solutions with address repeaters and multiple crossbars have been adopted to increase the address bandwidth by using a combination of snooping and directory cache-coherence protocols in the Fireplane[2] design from Sun MicroSystems. While directory protocols[1] are more scalable than snooping protocols, the drawbacks are the higher unloaded latency due to the directory indirection and the higher storage overhead required for maintaining the directory information. For example, the pin-to-pin latency in the Fireplane with only snooping protocols is 197 ns when the memory is on the same board and is 240 ns when the memory is on a different board. By adding the directory protocol, the average pin-to-pin latency of a transaction increases to 330 ns (38%) when snooping is done in different domains in the Fireplane[2] model. New architectures, such as Timestamp Snooping[10] and Asynchronous Caches[3] extend the address bandwidth and provide an alternative solution to bus-based SMPs. Hybrid architectures, such as Multicast Snooping[11] and Bandwidth Adaptive Snooping Hybrid[12] improve the address bandwidth by switching between snooping and directory protocols. While hybrid protocols[2,11,12] lower the address-bandwidth requirement for smaller configurations, they are still limited by the directory overheads for larger configurations as explained earlier. Snooping protocols are faster than directory protocols because they obtain data quickly without indirection that results in low latency for the network. By use of contemporary electrical interconnect technology, it remains a big challenge to have a large number of processors, and at the same time to implement fast, pure snooping cache-coherence protocols.

B. Optical Interconnects for Address-Bandwidth Limitation

One technology that can provide high communication bandwidth, low latency, and scalability is optical-interconnection technology.[6,7,13] The recent advances in optical-interconnect devices and packaging techniques such as multidimensional arrays of vertical cavity surface emitting lasers (VCSELs) with low-drive currents (<1 mA), low-drive voltages (<2 V), low-threshold currents (<10 μA), high-speed devices (>1 GHz), uniform device characteristics over the array, and high device yield (98%); arrays of photodetectors (PDs) with high bandwidth (>1 GHz), low noise (<20 $f\mathrm{W}/\sqrt{\mathrm{Hz}}$), high drive capability, and large dynamic range; and waveguide optics[14,15] are making optical interconnects a serious and potentially viable interconnect technology for parallel computing. The data transmission rate of a VCSEL is approximately 3–5 Gb/s. An array of such VCSELs enables address transmission with data rates in excess of 200–300 Gb/s.[15–17] This could satisfy the bandwidth demands of future SMPs. Two unique properties of optics, namely unidirectional propagation and predictable path delays[18] are exploited in this paper to significantly reduce latency and increase the address bandwidth. Optical pulses can coexist on the same optical line without interference if they are sufficiently separated. This enables multiple address requests to propagate within the same waveguide/fiber simultaneously. It can be argued that in electronic pipelined address buses, there could be several address requests in different phases of address translation, such as bus arbitration, address transmission, or waiting for the snoop response. Our contention is that, in electronic SMPs, only a single address request is transmitted on the physical address bus at any given point in time. Optically, we can easily have more than one address request in propagation simultaneously as discussed above. These advantages provide us with the impetus to look at optical technology to develop scalable SMPs with hundreds of processors while still using snooping cache-coherence schemes.

This paper proposes an integrated solution to solve the address bandwidth requirements of large, scalable SMPs and to still use fast snooping protocols to maintain cache coherence with low latency with optical technology. An address subnetwork, called optical symmetric multiprocessor network (SYMNET) with parallel optical interconnects is proposed with one-to-many communications. Parallel optical interconnects provide a higher bandwidth-density product as compared with serial interconnects that provide a higher bandwidth-distance product.[14] SYMNET not only has the ability to pipeline address requests, but also multiple address requests from different processors that can propagate through the address subnetwork simultaneously. The simultaneous insertion of multiple address requests complicates cache coherence. We have introduced a modified snooping-coherence protocol, called coherence in SYMNET (COSYM) and verified its correctness by use of several transient states. The use of transient states is not a new concept because it has been widely documented, but the transient states in our architecture are used to solve the write atomicity along with the snoop response requirements. SYMNET with the COSYM protocol is compared with electrical bus-based systems with the MOESI protocol with Splash-2[19] benchmarks.

C. Related Work

Optical bus-based multiprocessor systems by use of the coincident-pulse technique provide optical solutions to the problems of bus design in areas of address bus arbitration, device addressing, and data transfer,[18] however, the problem of cache coherence is not discussed. The photobus smart pixel interconnection system for shared-memory multiprocessors uses optical buses for broadcasting the address requests, but arbitration is implemented with electronic buses leading to the buffering of address requests at the smart-pixel VLSI chip.[20] The Berkeley cache-coherence protocol is used in the photobus architecture.[1] The constraints of access arbitration is eliminated in the U-bus[7] design for SMPs. The U bus extends the address bandwidth, but a new coherence protocol must be designed to maintain consistency across the caches. In the Speed[21] architecture, write requests are broadcast with the snooping protocol and read requests are unicast with the directory protocol. The I-Speed coherence protocol used for this architecture implements a single owner for dirty blocks to preserve the consistency of caches. Lightning network[22] uses directory cache-coherence protocols in which all transactions are completed in a single hop and is constructed as a tree configuration with a wavelength partitioner at each level of the tree. Optical networks discussed so far employ serial links to transmit address requests and data responses between the source and the destination by use of wavelength division multiplexing (WDM) technology. Moreover, directory cache-coherence protocols are used to maintain the coherency, which increases the latency as discussed above. The optical solutions so far have not been able to integrate fast, pure-snooping cache-coherence protocols nor to significantly improve the address-bandwidth demands to scale the architecture.

2. SYMNET Address Subnetwork

The proposed optical symmetric multiprocessor network, SYMNET, is shown in Fig. 1. Figure 1(a) shows SYMNET consisting of the processing elements/memory modules and an interconnection network. The interconnection network consists of two subnetworks: address and data subnetworks. The address and data subnetworks are separated, which reduces the design complexity and enables the design of large scalable SMPs. Scalable data subnetworks have been studied elsewhere,[23] therefore this paper focuses only on the address subnetwork. The address subnetwork consists of two components: a transport part capable of transmitting multiple address requests and a control part that ensures collisionless transmission of these address requests. The transport part in SYMNET is implemented with bidirectional couplers/splitters and the control part is implemented with an optical token. In what follows, we describe the SYMNET address subnetwork and then explain how the architecture is implemented.

The address subnetwork follows a two-level hierarchical architecture design. The first level consists of grouping a few processors on the boards by use of intra-board interconnections, and the second level consists of interconnecting these boards by using inter-board interconnections. The inter-board and intra-board interconnections are constructed with a bidirectional Y-splitter/coupler combination. Time division multiple access (TDMA) protocol is used as a control mechanism to achieve mutual exclusive access to the transport part. Several TDMA protocols, such as preallocation-based protocols,[24] reservation-based protocols with preallocated reservation control,[22] and token-based TDMA protocols[21] have been reported. In this paper we consider an optical token-based TDMA protocol with preallocation to prevent collision of address requests.

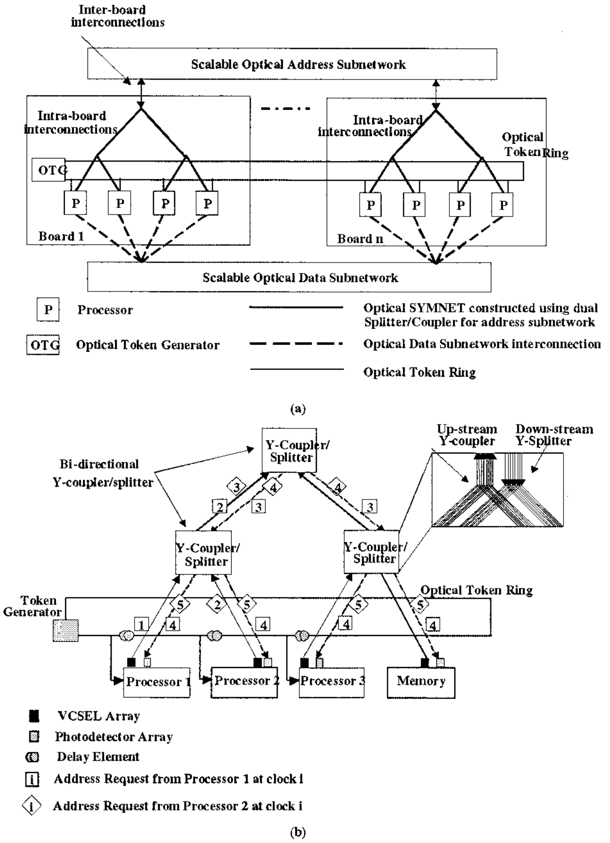
Inter-board interconnections
Scalable Optical Address Subnetwork
Intra-board interconnections
Intra-board interconnections
Optical Token Ring
OTG
P
Board 1
Board n
Scalable Optical Data Subnetwork
P Processor
OTG Optical Token Generator
Optical SYMNET constructed using dual Splitter/Coupler for address subnetwork
Optical Data Subnetwork interconnection
Optical Token Ring
(a)
Y-Coupler/Splitter
Bi-directional Y-coupler/splitter
Up-stream Y-coupler
Down-stream Y-Splitter
Y-Coupler/Splitter
Y-Coupler/Splitter
Optical Token Ring
Token Generator
1
2
3
4
5
Processor 1
Processor 2
Processor 3
Memory
■ VCSEL Array
▨ Photodetector Array
Delay Element
i Address Request from Processor 1 at clock i
i Address Request from Processor 2 at clock i
(b)

Fig. 1. (a) Proposed optical symmetric multiprocessor network (SYMNET) in which the processors are connected to two subnetworks; address subnetwork and data subnetwork, (b) overview of a single board in SYMNET. The processors/memory modules are connected to bidirectional Y-couplers by optical waveguides/fibers. The letter i denotes a variable.

The basic building block of the SYMNET address subnetwork is shown in Fig. 1(b). The address subnetwork is constructed with bidirectional Y-couplers, which provides two-way address transmission [see the inset in Fig. 1(b)]. The upstream Y-couplers are used for combining the address requests from the processors. After reaching higher levels, this address request is rerouted through the downstream Y-splitters that enable broadcasting of the address requests to all the processors and memory modules. It should be noted that the broadcasting allows a request to reach all processors and memory at the same time. This is a very useful feature for the cache-coherence protocol design to be discussed below. The optical token is a single pulse generated by the token generator. It provides a time reference for insertion of address requests into the subnetwork by each individual processor. The optical token is tapped by the processor, which triggers the electronic interface to drive the address request. The token is delayed by using a delay element, which provides sufficient time to drive the electronics and also ensures that the address requests from successive processors are transmitted without collision. The address requests from different processors are pipelined, which allows multiple requests to propagate through the address subnetwork. By use of the properties of optics,[18] namely unidirectional propagation and predictable path delays, it is possible in SYMNET to transmit multiple address requests simultaneously on the same waveguide/fiber. This is in contrast to all electrical shared-bus solutions, where only a single address request is transmitted on the physical address buses at any given point in time. These address requests move up the hierarchy and then are retransmitted back to all processors and memory simultaneously. This ensures that different requests from different processors are serialized in the global order of requests needed to maintain memory consistency.[1]

The optical clock and the token generator are synchronized, thus successive processors receive the token every clock cycle. As shown in Fig. 1(b), in cycle 1 as indicated by the open and numbered squares, the optical token is received by processor 1, which transmits an address request. During cycle 2, when this address from processor 1 is in propagation at the next level of the address subnetwork, the token is received by processor 2, which can transmit an address request. This is shown in the shape of an open and numbered diamond in Fig. 1(b). In cycle 3, the address request from processor 1 is being rerouted with the downward Y-splitter and at the same time the address request from processor 2 has moved up the address subnetwork. The optical token is now received by processor 3, which can transmit an address. In cycle 4, the address request from processor 1 has reached all the processors, thereby the address request is broadcast to all the processors simultaneously. In cycle 5, the address request from processor 2 has reached all the processors. Broadcasting the address request results in simultaneous reception of the request by all the processors/memory modules enabling snooping of the same request, after which appropriate coherence action is taken as dictated by the snooping protocols.

3. SYMNET Implementation Details

In this section, we analyze a possible implementation of optical SYMNET for address request propagation using parallel optical interconnects, such as VCSEL/Photodetector arrays, integrated arrays of 2 × 1 Y-splitters and 1 × 2 Y-couplers, and polymer waveguides/ribbons.

A. Components Required for SYMNET

- Parallel optical interconnects: The key component of SYMNET is the VCSEL/PD arrays (transceiver arrays) capable of transmitting at data rates in excess of 3 Gb/s per channel,[15,16] which results in providing aggregate data rates of several hundreds of Gb/s.[17] The low-cost linear arrays of VCSEL offer a number of advantages[15] over conventional edge-emitting laser diodes. High-performance GaAs- and InGaAs-based selectively oxidized or proton implanted top-emitting, bottom-emitting VCSEL arrays emitting at 780 nm to 980 nm have been widely reported in the literature.[15,25] Even commercially, several optical component manufacturers such as Xanoptix, TerConnect, Agilent, Corona, Cielo, Emcore, Infineon, and Picolight[17] have developed one- and two-dimensional transceiver arrays with geometries of (1 × 12, 4 × 12, 6 × 12) operating at the lower spectrum of 850/980-nm wavelength. Therefore parallel optical interconnects for short distances are viable options for developing an optical interconnection network.
- Polymer waveguides/ribbons: Optical polymers are increasingly considered as highly versatile elements that can be readily transformed into single-mode, multimode, and micro-optical waveguide/fiber structures because they exhibit excellent thermal stability, low optical loss, humidity resistance, low birefringence, flexibility, mechanical robustness, and have demonstrated capability in a variety of demanding applications. Acrylate-based polymers,[26] developed by Allied Signals, have shown optical loss less than 0.1 dB/cm at 0.8 μm. The low loss in these polymers makes them an attractive material for constructing the 2 × 1 couplers, 1 × 2 splitters, and for routing optical pulses from VCSELs to these couplers/splitters in the SYMNET interconnection network. The ability of acrylate-based polymer to be fabricated on a variety of substrates makes it suitable to be directly interfaced to micro-optical elements, such as micro-optical mirrors, 45 deg micro-reflectors, micro-optical lenses, and also to fiber-to-waveguide interconnect structures. Optical waveguides constructed by use of other techniques, such as photolithographic techniques, reactive ion etching, excimer laser ablation, and molding and embossing techniques[14,26] have been reported.
- Arrays of couplers/splitters: The optical pulses from the VCSELs are routed through polymer waveguides to arrays of integrated 2 × 1 couplers. These couplers can easily be constructed with optical polymer waveguides,[26,27] and these couplers are further connected to the next 2 × 1 coupler to construct the address subnetwork.

B. Technology for Integrating Electronic and Photonic Components

Optical interconnects based on complementary metal-oxide semiconductor CMOS/VCSEL technology[28,29] have been widely proposed for high-performance computing applications. The approach followed in our design is the most widely used hybrid integration[25] with flip-chip bonding of optoelectronic (OE)-VLSI components. The VCSEL/PD arrays can be fabricated on a GaAs substrate such that the devices are designed to be back-side emitting because of the desire to flip-chip bond them to CMOS driver circuits. The *n* contact and *p* contact should then be on the top surface of the wafer to facilitate electrical connectivity with CMOS circuits. GaAs substrate can then be selectively etched leaving the VCSEL/PD contact pad on the backside of the wafer and all optical sources/detectors on the other side of the wafer. The VCSELs and PDs can now be integrated onto the CMOS driver with flip-chip bonding[28] and substrate removal[29] techniques. The passive alignment of VCSELs to waveguides by placing the alignment pedestals on the assembly surface at the locations in reference to the optoelectronic component fiducial marks have been demonstrated,[15] which showed efficient coupling between a waveguide and VCSEL array. VCSEL-waveguide coupling by use of 45 deg mirrors has also been demonstrated,[30] where the mirror loss was estimated to be 0.2 to 0.8 dB at 0.83 μm.

The functional details of SYMNET are shown in Fig. 2. On a cache miss, the address request is forwarded to the address-port controller by the cache controller. The optical token is generated by a high-powered, high-frequency (10 GHz) laser source. This optical signal is split, such that one part of the signal is received by the address-port controller and the other part, is delayed by the delay element implemented by use of a fiber loop. When the token is received, the address-port controller forwards the address request to the transmitter integrated circuit (IC) that drives the VCSEL arrays. The optical pulses emitted from the VCSEL arrays propagate through integrated polymer 2 × 1 upstream couplers, reach the root of the address subnetwork, and is transmitted back by use of 1 × 2 downstream splitters. The address bits encoded as light pulses are eventually detected by the photodetector after amplification and are returned back to the address-port controller from the CMOS receiver IC. The received address request is then forwarded to the cache controller for processing of the request. In Fig. 2, the interconnection between two successive processors n and $n + 1$ on a system board is shown. Few processors, say 2–4 could be combined at the intra-board level. These boards could then be connected to several other boards and a hierarchy of interconnection can be constructed. SYMNET provides easy addition of boards without significantly altering the existing architecture, which facilitates scalability of the address subnetwork.

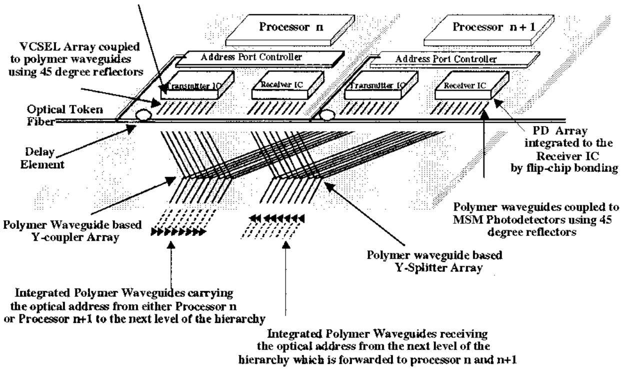
Processor n
Processor n + 1
VCSEL Array coupled to polymer waveguides using 45 degree reflectors
Address Port Controller
Address Port Controller
Transmitter IC
Receiver IC
Transmitter IC
Receiver IC
Optical Token Fiber
Delay Element
PD Array integrated to the Receiver IC by flip-chip bonding
Polymer waveguides coupled to MSM Photodetectors using 45 degree reflectors
Polymer Waveguide based Y-coupler Array
Polymer waveguide based Y-Splitter Array
Integrated Polymer Waveguides carrying the optical address from either Processor n or Processor n+1 to the next level of the hierarchy
Integrated Polymer Waveguides receiving the optical address from the next level of the hierarchy which is forwarded to processor n and n+1

Fig. 2. Shows the interconnection between processor n and n+1 along with the address port controller, optical token ring, delay element, VCSEL, and photodetector arrays.

4. Cache Coherence

Coherence and consistency[1] are different aspects of memory-system behavior, both of which are critical for executing correct shared-memory programs. In shared-memory systems, if a processor modifies the shared-memory location, then this information should be propagated to all caches existing in the multiprocessor system either by invalidating or updating the shared location, thus every read should return the latest write to it. Coherence in SYMNET, called COSYM, is modified from the popular MOESI[5] protocol. In SYMNET, an address request is issued on a cache miss. This request is inserted into the address subnetwork when the optical token is received by the processor. The address request traverses through several Y-splitters and couplers, and then becomes detectable to all processors simultaneously. In SYMNET there is a fixed latency between the time when the address request is inserted into the address subnetwork and when this request becomes detectable to all processors. This is in contrast to electrical bus-based SMPs,[4] where the address request becomes detectable immediately after the request is broadcast on the bus. Therefore in electrical SMPs, the cache controller is aware of all previous outstanding requests in the system at the time of inserting the new request. In SYMNET, at the time of inserting an address request, the cache controller is unaware of other requests propagating through the address subnetwork affecting the same cache block for which the request is inserted. COSYM protocol handles all race conditions that arise because of the simultaneous propagation of multiple-address requests by use of several transient states. In what follows, we discuss the snoop response requirement and the implementation of COSYM protocol.

A. Design Space for Snooping Protocols Implemented Optically

In the electrical networks, the snoop response is implemented by use of two wired-OR lines,[1] shared and owned. The processors sharing the block assert the shared line if the block is in the shared state, or the owned line if the block is in any of the following states: E, M, or O. The shared snoop line could be asserted by more than one processor. In an optically interconnected multiprocessor system, if more than one pulse is inserted into the network as a snoop response by multiple processors, a collision of snoop responses from several processors result in an erroneous response being received by the requestor as they operate at the same wavelength. Therefore the constraint for the snoop response in our architecture is that it should be a single response from a single processor. To achieve this, we maintain an owner for every cache block shared. The owner is responsible for providing the snoop response. In the case of a dirty block, the owner is the most recent processor that wrote to that block. In the case of clean block, there could be several processors sharing the block.

Table 1. Cache Controller Transient States

Transient State	Description
IE-ads*	Invalid, issued read request, has not inserted the address request into the network
IE-ads	Invalid, inserted read request, yet to receive address, snoop, or data response
IE-ds	Invalid, issued read request, received address, yet to receive snoop, and data response
IE-d	Invalid, issued read request, received address and snoop response (low), yet to receive data response
IS-ads*	Invalid, issued read request, yet to receive address, data, and snoop response
IS-ds	Invalid, issued read request, received address, yet to receive snoop and data response
IS-d	Invalid, issued read request, received address and snoop response (high), yet to receive data response
IO-ds	Invalid, issued either read or write request, received address, yet to receive snoop and data response
IO-d	Invalid, issued either read request or write request, received address and snoop, yet to receive data response
IM-ad*	Invalid, issued or/and inserted write request, yet to receive address response and data response
IM-d	Invalid, issued write request, received address, yet to receive data response
S/O,M-a	Owned/shared, issued write request, yet to receive the address request
II-d*	Invalid, next state is known to be Invalid, yet to receive data response

To determine a single owner, MOESI protocol is modified such that if a read miss request is issued to an E block, the block is upgraded to O instead of S, and this makes the processor that was initially in an E state, the owner of the shared block. This does not change any other protocol constraints. Reads from the processor can still be satisfied and writes will still require an invalidation transaction to be issued. In the COSYM protocol, a single snoop-response (high or low) signal can determine all the relevant information required as follows:

Snoop high: A dirty block exists, memory need not respond to the requestor, and if it was for a read request, the block is loaded in S state.

Snoop low: No dirty block exists, memory responds with the data to the requestor, and if it was for a read request, the block is loaded in E state.

B. Implementation of the COSYM Protocol

States and Events: The stable states in COSYM have the same function as in the MOESI[5] protocol. Table 1 describes the transient states used in COSYM protocol. This representation is similar to the table-based method adopted in the multicast[11] protocol. The active/inactive status indicates whether the transient state reacts to incoming address requests. In Table 1, inactive status is indicated by means of an asterisk (*) next to the transient state. The cache controller reacts to two kinds of requests, issued either from the processor or from the interconnect. The address requests issued by the cache controller in SYMNET are read miss, write miss, and upgrade/invalidation requests. The cache controllers make transitions based on their current state and current events. The events that cause the transitions are an address request being issued from the processor, the request being inserted into the interconnect, the request becoming detected, receiving high/low snoop response for the request, and finally receiving the data.

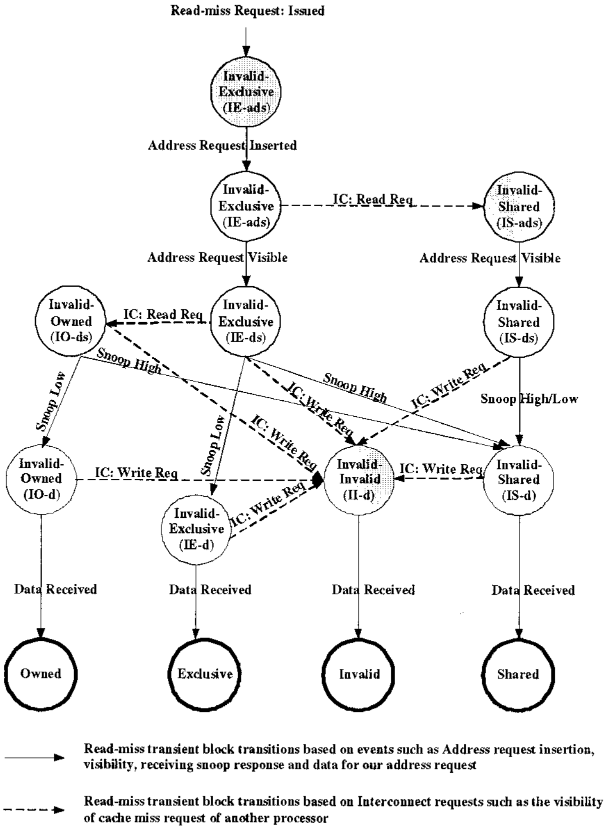

Fig. 3. COSYM cache-coherence protocol using state diagram for all the transient states when a read miss occurs. The various events that cause the transition are issuing of address requests, inserting address requests, visibility of address requests, and receiving snoop signals and data. The transitions caused by address requests made by other processors are indicated IC:Read/Write Req. The arrow from one transient state to another indicates either the event that caused the transition or another processor's request received from the interconnect.

Transient state diagram: The transient diagram of the COSYM protocol in case of a cache miss are shown in Figs. 3 and 4. In Fig. 3 for read-miss case, the states indicated with a gray shade implies that the protocol is not reactive to requests yet. The white open circles, which are not circled in bold, indicate the states in which the protocol reacts to incoming transactions. The text associated with the arrow from one state to another transient state indicates the event that caused the transition. The

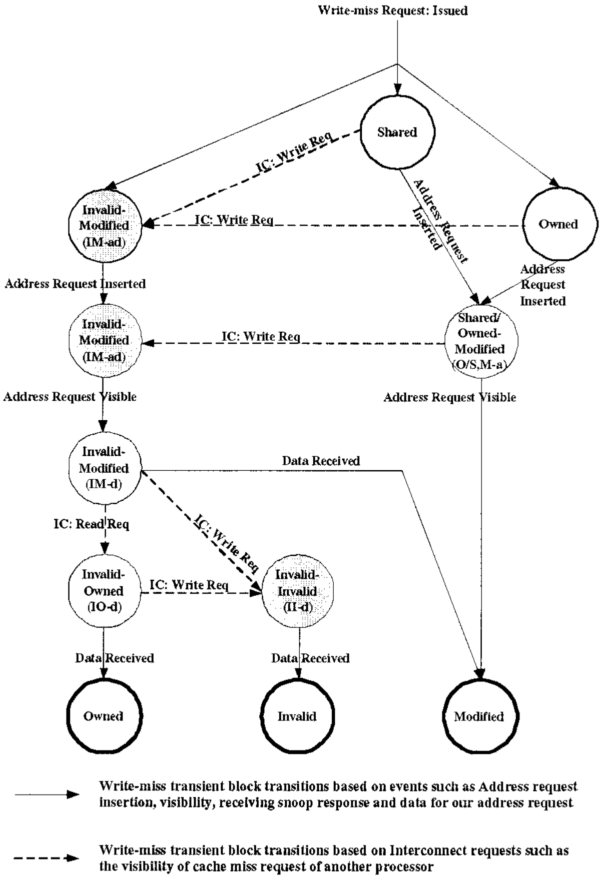

Fig. 4. COSYM cache-coherence protocol using state diagram for all the transient states when a write miss occurs. Otherwise, same as in Fig. 3.

transitions due to cache-miss requests issued from the interconnect are shown with a dotted line. Each transient state is indicated in the following manner:[11] ⟨present state⟩-⟨next state⟩-⟨abbreviation- a/d/s⟩. For example, when a read miss occurs, the transient state is indicated as: ⟨Invalid-Exclusive (IE-ads)⟩. Invalid (I) indicates the present state, the next state is Exclusive (E) and ads stands for the pending address, data and snoop response. When the read-miss request is inserted, the cache is reactive to other requests issued to the same block. When the address request is detected, the state changes to ⟨Invalid-Exclusive (IE-ds)⟩, which indicates that the data and the snoop signal are pending. The other state reachable from IE-ads is IS-ads and IE-ds. IS-ads indicates that the processor has detected a read-miss request issued by another processor that has sequential priority before its own request. When the address request is detected, the transition takes place to either IE-ds or IS-ds depending on the previous state. If the block is in IE-ds and a read-miss request from another processor is detected, the block transits to IO-ds. All write-miss requests from the interconnect will result in the block being downgraded to the II-d state. When the snoop signal is received, depending on whether it is high or low, the next possible states are IE-d, IO-d, or IS-d. Finally, when the data is received, the block makes the transition to E, S, O, or I states, depending on the previous transient state.

In case of a write miss as shown in Fig. 4, the block may be in I, S, or O states. If a write miss occurs, the block transits to IM-ad, indicating the pending address and data. The snoop signal is not relevant to the issuing processor because it only needs the data to write to the block. The snoop signal is still provided by the owner, if one exists, to the memory controller. Once the write-miss request is detected, it is serialized in the global order of requests and transits to IM-d, which indicates that the data is not received. Any intermediate read request causes the block to change to IO-d state. When the data is received, depending upon the previous state, the block is loaded in M, O, or I state. If a write miss occurs and the cache does have a valid block (O/S), it issues an invalidation request and transits to O/S,M-a state. The data is already valid and the processor waits only for the invalidation request to be detected. When the cache block is in the O/S,M-a state, if a write request is issued from the interconnect, the block transits to IM-ad, which indicates that both data and address are pending, and the controller reacts in a similar fashion as explained.

5. Performance Evaluation

A. Simulation Methodology and Architectural Assumptions

We have chosen Limes (Linux Memory Simulator),[31] an event-driven execution simulator to evaluate the performance of SYMNET with electrical bus-based systems considering realistic delays for address and data transactions. Limes models a single level cache and a blocking bus. We have extended the simulator to implement a two-level cache with a split-transaction bus by merging or delaying conflicting requests for the electrical system. We assume that the data network and memory access are contention free to maximize the effects of the limited-address bus bandwidth. We compare SYMNET with the traditional electrical bus-based SMPs implementing the MOESI protocol with a subset of Splash-2[19] suite benchmarks. The electrical SMP considered for comparison is similar in design to the Gigaplane[9] and StarFire[8] models.

Benchmarks: In this study, we use eight benchmarks, which cover a spectrum of memory sharing and access patterns from the Splash-2 suite,[19] namely FFT with input data set 64 k points; LU with 256 × 256, 16 × 16 block; Ocean with 130 × 130; Radix with 1M integers, 1024 radix; , Water-nsquared with 512 molecules; FMM with 16 k particles; Barnes-hut with 1k particles, and Cholesky with tk14.0, to evaluate the performance of COSYM and MOESI protocols. We varied the number of processors from 2 to 32 to evaluate the performance of SYMNET. Unfortunately, owing to the complexities of full-system simulation, we were unable to simulate for more than 32 processors for some applications. However, applications such as FFT, LU, Radix, Water and Ocean showed similar trends when simulated for 64 processors.

Processor/cache parameters: Each node of the simulated network contains a 1 GHz processor and has two cache levels, namely L1 and L2. The L1 cache is a 16 kbyte direct mapped, with a 32 byte block size and a write-through policy. The L2 cache is 64 kbyte, 4-way set associative with a 32 byte block size and a write-back policy. Both the caches implement an LRU (least recently used) replacement policy. The access time to L2 cache is 4 cycles. The processor and the cache parameters are kept constant while simulating both electrical and optical networks. All first-level cache read/write hits are assumed to take one processor clock cycle (pcc). Throughout this evaluation, we have considered pcc's as the base time unit for all measurements.

Electrical simulation parameters: In electrical SMPs, the address bandwidth is affected by several factors, such as the bus speed, coherence protocol, and the number of address buses. In electrical bus-based SMPs,[2,8,9] the processor clock is always a fraction of the system clock rate. For example, in the StarFire model, UltraSparc2 is clocked at 250 MHz, whereas the system bus is clocked at 83.3 MHz.[8] This implies that the system clock is approximately 1/3rd the cycle of the processor clock. With a 1 GHz simulated processor, and an improved system clock, we assume that the system clock runs at 1/6th the cycle of processor clock. In the Gigaplane[9] architecture, it takes 2 cycles to broadcast a single address request. With the above assumption, it takes 12 pcc to broadcast a single-address request in our simulated address bus. This single-address request per cycle (RPC) is denoted in all results as (RPC = 1). In the StarFire[8] architecture, processors snoop up to two address requests per cycle using four address buses. This case is simulated where each processor receives 2 address requests per cycle and is accomplished by reducing the number of cycles required to broadcast an address request to 6 pcc. This two address requests per cycle, is denoted in all results as (RPC = 2). The data network is contention free and is implemented by use of a crossbar for both RPC = 1 and RPC = 2 cases. The number of cycles required for data transfer is fixed at 2 electrical network cycles irrespective of whether the memory or some cache responds as in the StarFire[8] design. This results in 24 pcc for data transfer in our simulated network for both RPC = 1 and RPC = 2 cases.

Optical simulation parameters: In SYMNET, the optical token is implemented such that the optical signal, generated by a laser source, is split as shown in Fig. 2. One part of the optical signal is detected by the address-port controller and the other part is delayed at the delay element implemented with a fiber loop. Now, we calculate the delay D in transmitting an address request into the address subnetwork by the address port controller. This delay should account for the signal detection, optoelectronic conversion and the time to advancement of address pulses driven by VCSEL arrays. The delay D is given by

$$D = \frac{S_p}{v_c} + 2.O_e + G_d + \frac{b}{mV_d}, \tag{1}$$

where S_p is the distance of separation between the delay element at processor n and the detector at processor $n + 1$, v_c is the velocity of light in fibers, O_e is the latency of optoelectronic conversion, G_d is the gate delay faced by the token at the address port controller, m is the number of parallel links, b is the number of address bits (including one bit for snoop response), and V_d is the VCSEL data rate. O-E conversion takes place when the optical signal is detected by the address-port controller and E-O conversion takes place when the address bits encoded as optical pulses are driven by the VCSEL array. It is assumed that a single gate delay is detected by the address port controller when it receives the token. Assuming that $S_p = 4$ cm, $v_c = 2 \times 10^8$, $O_e = 75$ ps, $G_d = 0.2$ ns, $V_d = 3$ Gbs and with $m = b$, D is estimated to be 0.88 ns. The optical token should be detected by the next processor with a delay greater than 0.88 ns to prevent the collision of address requests. Therefore, the other part of the optical signal at the delay element should be delayed by more than 0.88 ns. Adding guard time to D, we assume the delay to be 1 ns. Considering 1 ns as the required delay, we can estimate the length of the delay element to be 20 cm [$=(2 \times 10^8) \times 1$ ns]. The delay element is implemented by use of a fiber loop 20 cms in length. Therefore the time taken by each processor to insert its address request is estimated to be 1 ns or 1 pcc. The delay encountered by an address transaction to be detected is equivalent to the number of stages in the address subnetwork. This is assumed to be twice the logarithm of the number of processors connected in the address subnetwork. The snoop response also takes a similar number of cycles. The delay in data transfer for the optical network depends on the data rate of current multiwavelength VCSEL arrays. At 5 Gb/s VCSEL data rate, to transmit a 32-byte block takes approximately 52 ns [$=(32 \times 8)/(5 \times 10^9)$], and this corresponds to 52 pcc. The data network considered for SYMNET is the SOCN[23] (scalable optical crossbar-connected interconnection network). SOCN is an optical crossbar constructed by use of VCSEL/PD arrays and a diffraction grating. The data subnetwork for SYMNET is also assumed to be conflict free.

B. Simulation Results

We determined the execution time and the average delay for a cache-miss transaction for SYMNET and the electrical bus-based SMP varying from 2 to 32 processors.

Normalized execution time: Figure 5 displays the normalized execution time for a varying number of

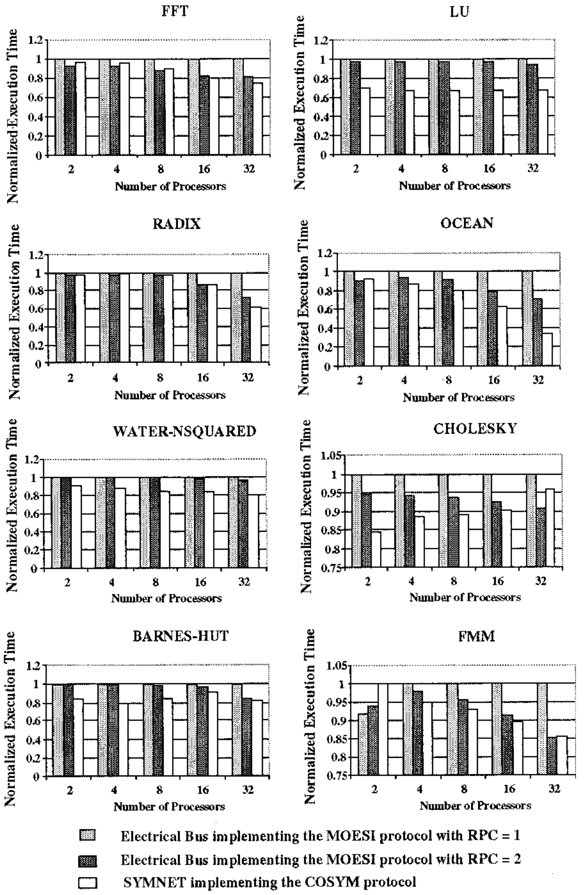

Fig. 5. The Normalized Execution time for processors varying from 2 to 32 for all Splash-2 benchmarks is shown. The execution time for the electrical bus implementing the MOESI protocol with a single address request per cycle (RPC = 1), two address requests per cycle (RPC = 2) and the SYMNET address subnetwork implementing COSYM protocol is shown. Normalized execution time is calculated by considering the maximum number of simulated cycles for a given application and given number of processors. The remaining two cases for a given number of processors are normalized to this maximum value.

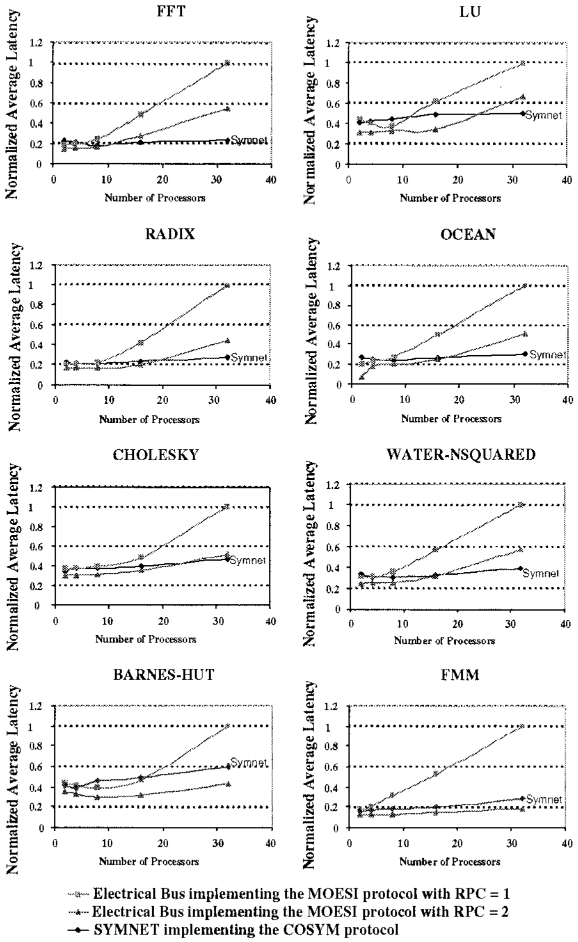

Fig. 6. The Normalized Average latency of a transaction for processors varying from 2 to 32 for all Splash-2 benchmarks is shown. The average latency for the electrical bus implementing the MOESI protocol with a single address request per cycle (RPC = 1), two address requests per cycle (RPC = 2) and the SYMNET address sub-network implementing COSYM protocol is shown. Normalized average latency is calculated by considering the maximum value of average latency for a given application and dividing all the remaining cases with the maximum value.

processors for different applications. Normalized execution time is calculated by considering the maximum number of simulated cycles for a given application and given number of processors. The remaining two cases for a given number of processors are normalized to this maximum value. For FFT, COSYM shows a 25% improvement over the MOESI protocol for RPC = 1 and an 8% improvement for RPC = 2 running for 32 processors. For the LU application, the improvement in execution time is around 30–35% for both the cases. The best improvement is visible for Ocean application, where the improvement is over 66% for RPC = 1 and over 52% for RPC = 2 running for 32 processors. The improvement in performance for Radix is 38% for RPC = 1 for 32 processors. Cholesky and Water-nsquared applications show lower improvement in performance for COSYM protocol, with Cholesky showing an improvement of 5% for RPC = 1 and water shows an improvement of 16–19% for both cases. COSYM protocol improves execution time by 20% and 15% for Barnes-Hut and FMM applications for 32 processors, respectively.

Normalized average latency: Figure 6 shows the normalized average delay for a transaction to be completed for both the electrical and the optical case. The delay in completing a transaction was calculated from the time the cache miss request was received by the L2 cache to the time the data was received by the L2 cache for each processor. The ratio of the total number of transactions to the total time consumed for all processors was used to determine the average delay. This delay was then normalized by considering the maximum average delay for a given application and then dividing all the remaining cases with this value. The delay for the single address request per cycle case was higher than the two address requests per cycle case as expected. Most applications showed lower delay for MOESI protocol for smaller configurations. As the number of processors increased, COSYM outperformed both electrical cases. This is directly attributed to the saturation of the electrical bus, as the number of processors increases in the interconnect, the delay to acquire the bus also increases, thereby increasing the latency for a transaction to complete. The COSYM model with a faster address interconnect and a data crossbar provides much better performance for all the cases. The reduction in latency for FFT with the COSYM protocol is as high as 76% for RPC = 1 and 57% for RPC = 2. For LU, the reduction in latency for COSYM protocol ranged from 51% for RPC = 1 to 25% for RPC = 2. Barnes-Hut and FMM showed slightly better performance for the RPC = 2 condition than COSYM protocol. COSYM still outperformed Barnes-Hut and FMM for RPC = 1 condition by as much as 40% and 85%, respectively.

These simulation results clearly indicate that the proposed optical SYMNET with COSYM as the coherence protocol provide much better support for scalable SMPs than their electrical counterparts. We modelled realistic delays for current electrical systems and for our proposed optical interconnect system making our simulation results significant. Our simulation studies have shown a 5–66% improvement in execution time for COSYM as compared with MOESI for various applications. Simulations have also shown that the average latency for a transaction to complete by use of COSYM protocol was 5–78% better than the MOESI protocol. With theoretical power budget analysis, it has been shown that SYMNET can scale up to 128 processors.[32] This is a significant improvement considering the largest pure snoop electrical SMP can support 64 processors.[8]

6. Conclusion

In this paper, we studied the primary limitation of address bandwidth in SMPs. As a solution, we propose a parallel optical interconnect based on symmetric multiprocessor network (SYMNET) and a modified cache-coherence protocol called COSYM. SYMNET improves execution time and reduces the latency by pipelining multiple address requests from different processors simultaneously. By use of the modified Limes simulator, we simulated SYMNET implementing the COSYM protocol and compared it with the electrical bus-based MOESI protocol with Splash-2 benchmarks from 2 to 32 processors. Our simulation studies have shown a 5–66% improvement in execution time for COSYM as compared with MOESI for various applications. Simulations have also shown that the average latency for a transaction to complete by use of COSYM protocol was 5–78% better than the MOESI protocol. The simulation results provide encouragement that SYMNET has the potential to match the bandwidth needs of future SMPs. Parallel optical interconnects and integrated waveguide technology makes SYMNET a viable solution for SMPs with distinct cost and performance advantages over traditional electronics. Greater improvements in terms of bandwidth, latency, and scalability can be expected with further improvement in optical device technology.

Future research will involve techniques to further increase the address bandwidth by use of wavelength division multiplexing (WDM). Each wavelength will be controlled by a separate optical token, which propagates through the token ring. Different tokens are assigned to different memory address spaces and transmission to a particular address space is issued at the wavelength assigned to that address space. These techniques are being pursued to further improve the performance of SYMNET in terms of bandwidth and latency.

This research project was sponsored by NSF grant no. CCR-0000518. We thank D. Litaize and J. H. Collet for their comments on this work.

References

1. D. E. Culler, J. P. Singh, and A. Gupta, *Parallel Computer Architecture: A Hardware/Software Approach* (Morgan Kaufmann, 1999).
2. A. Charlesworth, "The sun Fireplane SMP interconnect in the Sunfire 3800–6800," in *Hot Interconnects 9*, IEEE Comput. Society, Los Alamitus, Calif., 37–42 (2001) www.hotl.org.
3. F. Pong, M. Dubois, and K. Lee, "Design and performance of SMPs with asynchronous caches," Tech. Rep. HPL-1999-149, Hewlett Packard, HP Laboratories, Palo Alto, Calif. (1999).
4. D. A. Patterson and J. L. Hennesey, *Computer Architecture: A Quantitative Approach*, 2nd ed. (Morgan Kaufman, Los Altos, Calif., 1996).
5. P. Sweazey and A. J. Smith, "A class of compatible cache consistency protocols and their support by the IEEE futurebus," in *Proceedings of the 13th Annual International Symposium on Computer Architecture*, ISCA, 414–423 (1986).
6. D. A. B. Miller, "Rationale and challenges for optical interconnects to electronic chips," Proc. IEEE **88,** 728–749 (2000).
7. J. H. Collet, W. Hlayhel, and D. Litaize, "Parallel optical interconnects may reduce the communication bottleneck in symmetric multiprocessors," Appl. Opt. **40,** 3371–3378 (2001).
8. A. Charlesworth, "Starfire: Extending the SMP envelope," IEEE Micro. **18,** 39–49 (1998).
9. A. Singhal, D. Broniarczyk, F. Cerauskis, J. Price, L. Yuan, C. Cheng, D. Doblar, S. Fosth, N. Agarwal, K. Harvey, E. Hagersten, and B. Liencres, "Gigaplane: A high performance bus for large SMPs," in *Proceedings Hot Interconnects 4*, IEEE Computer Society, Los Alamitos, Calif., 97–112 (1996) www.hotl.org.
10. M. M. Martin, D. J. Sorin, A. Ailamaki, A. R. Alameldeen, R. M. Dickson, C. J. Mauer, M. Plakal, M. D. Hill, and D. A. Wood, "Timestamp snooping: An approach for extending SMPs," in *Proceedings of the Ninth International Conference on Architectural Support for Programming Languages and Operating Systems*, ACM, Cambridge, Mass., November 13–15, 25–36 (2000).
11. D. J. Sorin, M. Plakal, A. E. Condon, M. D. Hill, M. M. Martin, and D. A. Wood, "Specifying and verifying a broadcast and a multicast snooping cache coherence protocol," IEEE Transactions on Parallel and Distributed Systems, **13** (2002).
12. M. M. K. Martin, D. J. Sorin, Mark D. Hill, and David A. Wood, "Bandwidth adaptive snooping," in 8th International Sympo-

sium on High Performance Computer Architecture (HPCA), 224–235 (2002).

13. J. H. Collet, D. Litaize, J. V. Campenhut, C. Jesshope, M. Desmulliez, H. Thienpont, J. Goodman, and A. Louri, "Architectural approaches to the role of optics in mono and multiprocessor machines," Appl. Opt. **39,** 671–682 (2000).
14. Y. Liu, "Heterogeneous integration of OE arrays with SI electronics and microoptics," in *Proceedings of the Electronic Components and Technology Conference*, 864–869 (2001).
15. Y. S. Liu, R. J. Wojnarowski, W. A. Hennessy, J. P. Bristow, Y. Liu, A. Peczalski, J. Rowlette, A. Plotts, J. Stack, M. Kader-Kallen, J. Yardley, L. Eldada, R. M. Osgood, R. Scarmozzino, S. H. Lee, V. Ozgus, and S. Patra, "Polymer optical interconnect technology (point)-optoelectronic packaging and interconnect for board and backplane applications," in *Proceedings of the Electronic Components and Technology Conference*, 308–315 (1996).
16. A. V. Krishnamoorthy, K. W. Goossen, L. M. F. Chirovsky, R. G. Rozier, P. Chandramani, S. P. Hui, J. Lopata, J. A. Walker, and L. A. D'Asaro, "16 × 16 VCSEL array flip-chip bonded to CMOS VLSI circuit," IEEE Photonics Tech. Lett. **12,** 1073–1075 (2000).
17. A. Lindstrom, "Parallel links transform networking equipment," FiberSystems International, 29–32 (2002).
18. D. M. Chiarulli, S. P. Levitan, R. G. Melhem, M. Bidnurkar, R. Ditmore, G. Gravenstreter, Z. Guo, C. Qiao, M. F. Sakr, and J. P. Teza, "Optoelectronic buses for high-performance computing," in Proc. IEEE, **82,** 1701–1710 (1994).
19. C. S. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta, "The splash-2 programs: Characterization and methodological considerations," in *Proceedings of the 22nd Annual International Symposium on Computer Architecture*, ACM, ISCA '95, Santa Margherita Ligure, Italy, 24–37 (1995).
20. P. Lukowicz, "The photobus smart pixel interconnection system for symmetric multiprocessing using workstation clusters," in *6th International Conference on Parallel Interconnects*, IEEE, Anchorage, Al, 106–113 (1999).
21. J.-H. Ha and T. M. Pinkston, "The speed cache coherence for an optical multiaccess interconnect architecture," in *Proceedings of the 2nd International Conference on Massively Parallel Processing Using Optical Interconnections*, IEEE, San Antonio, Tex. 98–107 (1995).
22. P. Dowd, J. Perreault, J. Chu, D. C. Hoffmeister, R. Minnich, D. Burns, F. Hady, Y. J. Chen, and M. Dagenais, "Lightning network and systems architecture," J. Lightwave Technol. **14,** 1371–1387 (1996).
23. B. Webb and A. Louri, "A class of highly scalable optical crossbar-connected interconnection networks (SOCNs) for parallel computing systems," IEEE Transactions on Parallel and Distributed Systems, **11,** 444–458 (2000).
24. K. Bogineni and P. W. Dowd, "A collisionless multiple access protocol for wavelength division multiplexed star-coupled configuration: Architecture and performance analysis," J. Lightwave Technol. **10,** 1688–1699 (1992).
25. A. V. Krishnamoorthy and K. W. Goossen, "Optoelectronic-VLSI: Photonic integrated with VLSI circuits," IEEE J. Sel. Top. Quantum Electron. **4,** 899–912 (1998).
26. L. Eldada and L. W. Shacklette, "Advances in polymer integrated optics," IEEE J. Sel. Top. Quantum Electron. **6,** 54–68 (2000).
27. S. S. Saini, Y. Hu, Z. Dilli, R. Grover, M. Dagenais, F. G. Johnson, D. R. Stone, H. Shen, W. Zhou, and J. Pamulapati, "Integrated 1 × 2 loss-less Y-junction splitter on a passive active resonant coupler platform," in *Conference Lasers and Electro-Optics* 423–424 (2000).
28. R. Pu, E. M. Hayes, C. W. Wilmsen, K. D. Ohoquette, H. Q. Hou, and K. M. Geib, "Comparison of techniques for bonding VCSELs directly to ICs," J. Opt. Soc. Am. A **1,** 324–329 (1999).
29. H. J. J. Yeh and J. S. Smith, "Integration of GaAs vertical cavity surface emitting laser on Si by substrate removal," Appl. Phys. Lett. **64,** 1466–1468 (1994).
30. T. Sakamoto, H. Tsuda, M. Hikita, T. Kagawa, K. Tateno, and C. Amano, "Optical interconnection using VCSELs and polymeric waveguide circuits," J. Lightwave Technol. **11,** 1487–1492 (2000).
31. I. Ikodinovic, A. Milenkovic, V. Milutinovic, and D. Magdic, "Limes: A multiprocessor simulation environment for PC platforms," in *Third Conference on Parallel Processing and Applied Mathematics*, Institute of Mathematics & Computer Science, Technical University of Czestochowa, PPAM '99, Kazimierz Dolny, Poland.
32. A. K. Kodi and A. Louri, "Optical interconnects for large-scale symmetric multiprocessor networks," in Optics in Computing, SPIE, Vol. 1, Taiwan, April 2002, 7–9 (2002).