COPY

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

2

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	9/15/89	FINAL 15 July 1988 - 14 July 1989

4. TITLE AND SUBTITLE

PARALLEL MEMORY ADDRESSING USING COINCIDENT OPTICAL PULSES

5. FUNDING NUMBERS

AFOSR-88-~~198~~ 0198

6. AUTHOR(S)

DONALD M. CHIARULLI RAMI G. MELHEM
STEVEN P. LEVITAN

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

UNIVERSITY OF PITTSBURGH
Mineral Industries Building
Pittsburgh, PA 15260

8. PERFORMING ORGANIZATION REPORT NUMBER

none

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFOSR, Bolling AFB, D.C. 20332 Bld 410

NE

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

AFOSR-TR- 89-1707

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE, DISTRIBUTION UNLIMITED

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This research was a preliminary investigation of the applicability of coincident optical pulse techniques to hybrid electronic-optical computing systems. The results of the investigation are focused in two areas. First, it was determined that the technological constraints for pulse generation, detection, and synchronization do not substantially restrict the applicability of the technique. This conclusion was based on the results of pulse coincidence experiments. Second, the application of the technique is not restricted to memory addressing structures as had been originally proposed. It was demonstrated that coincident pulse methods can be applied to general multiprocessor interconnections. In this context they can be used to provide the functionality and performance of fully interconnected systems while using low cost and low complexity optical structures.

14. SUBJECT TERMS

Optical Computing
Optical - Electronic Hybrid Systems
Memory Addressing

15. NUMBER OF PAGES

20 + appendix

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unclassified

89 12 20 0 1

PARALLEL MEMORY ADDRESSING USING COINCIDENT OPTICAL PULSES

Donald M. Chiarulli
Dept. of Computer Science
University of Pittsburgh
Pittsburgh, PA 15260

Rami G. Melhem
Dept of Computer Science
University of Pittsburgh
Pittsburgh, PA 15260

Steven P. Levitan
Dept. of Electrical Engineering
University of Pittsburgh
Pittsburgh, PA 15260

AFOSR-TR- 89-1702

September 15, 1989

Final Report for period 15 July 1988 - 14 July 1989
AFOSR-88-198.

Prepared for
AIR FORCE OFFICE OF SCIENTIFIC RESEARCH
Building 410
Bolling AFB, DC 20332-6448

Summary

This research was a preliminary investigation of the applicability of coincident optical pulse techniques to hybrid electronic-optical computing systems. The results of the investigation are focused in two areas. First, is was determined that the technological constraints for pulse generation, detection, and synchronization do not substantially restrict the applicability of the technique. This conclusion was based on the results of pulse coincidence experiments. Second, the application of the technique is not restricted to memory addressing structures as had been originally proposed. It was demonstrated that coincident pulse methods can be applied to general multiprocessor interconnections. In this context they can be used to provide the functionality and performance of fully interconnected systems while using low cost and low complexity optical structures.

1. Objectives

The following objectives were met by this research:

1 Experiments were performed which demonstrate the feasibility of the technique of coincident pulse addressing, current technology and "off the shelf" components.

2 Experimentally determined limits on pulse width, synchronization and detection were used to project practical values for system parameters such as scalability, topology, and performance.

3 New structures were developed which extend the technique from the environment of memory addressing to more general problems in multiprocessor interconnections.

4 An analysis was performed to evaluate the practicality of the technique when extended to two dimensional structures.

2. Significant Accomplishments

2.1. Coincident Pulse Experiments

The goal of these experiments was to test the basic concept of pulse coincidence using commercial components and multi-mode glass fiber waveguides. To do this, several pulser and detector circuits were built. Experiments were performed with varying techniques for generating and detecting optical signals.

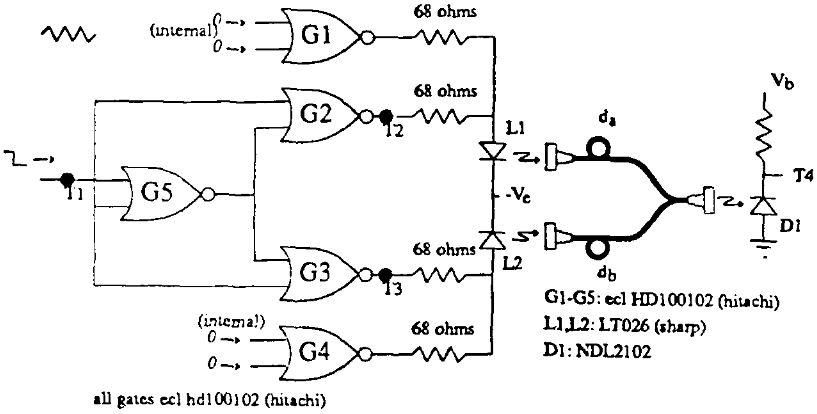


Figure 1: (a) Experimental Circuit

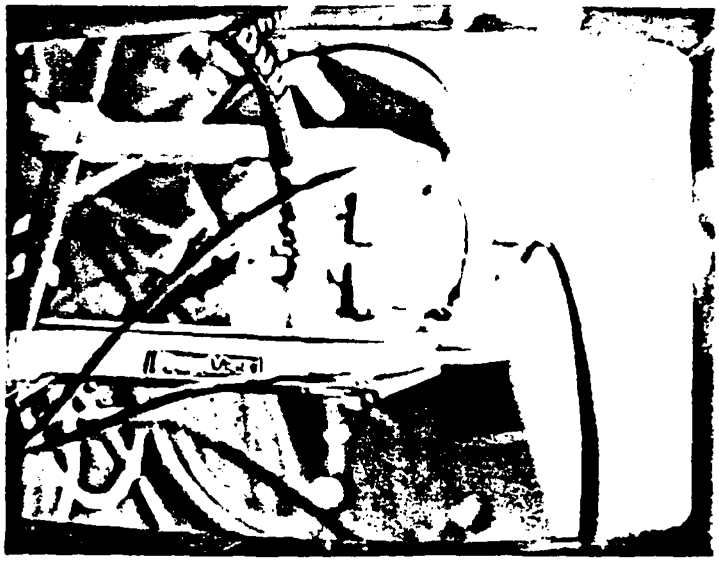


Figure 1: (b) Circuit Implementation

The experimental circuit shown in Figure 1 was used to generate coincident pulse streams, as shown in the oscilloscope traces in Figure 2. The top trace in each of these two displays is a pulse edge from a Tektronix model PG501 pulse generator used as a system trigger, and monitored at test point T1 in the schematic. The second and third traces are the electronic signals used to modulate, via G2 and G3, two laser diodes, each biased near threshold by gates G1 and G4. The traces are the outputs of these gates monitored at test points T2 and T3 respectively. The bottom two traces are stored waveforms recorded at T4 from the results of two separate experiments.

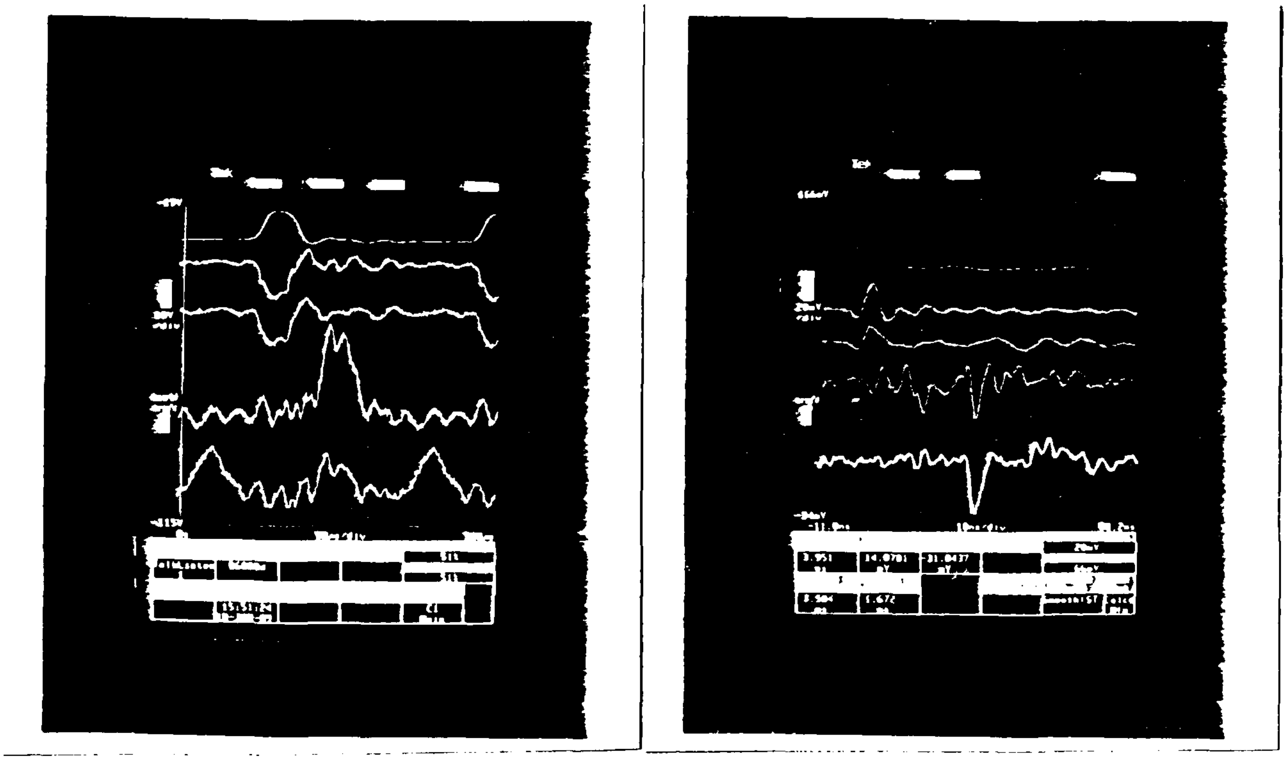


Figure 2: Coincidence Traces (a) positive pulses (b) negative pulses

In the first experiment, the optical path length d_a from laser diode, L1, to the detector is 3 meters, and the path d_b from the second laser diode, L2, to the detector is 6 meters. The resulting trace clearly shows the detector output with each pulse separated in time proportional to the difference in optical path length. In the second experiment, the optical path length of d_a and d_b are both 6 meters. Here the trace shows a single coincident pulse of amplitude sufficient to be easily discriminated from the two separate pulses shown in the previous experiment. This experiment was run for both positive optical pulses as shown in Figure 2a and for and negative optical pulses as shown in Figure 2b and achieved comparable results, using pulse widths as low as 3.9 nanoseconds.

2.2. Evaluation of Technology Based Constraints

A number of constraints imposed on coincident pulse structures by available technology have been identified. These constraints can be categorized as follows:

1) The scale of any implementation is determined by the smallest pulse that can be generated and detected using commercially available components and support circuitry appropriate for computing applications. As shown above, pulsers and detector circuits which operate at 4 nsec. are feasible. In the experiments the pulse was primarily limited by the 250mhz bandwidth available for the trigger input to the pulser. However, the components from which the circuit was constructed, 100K series ECL, Sharp LT026 laser diodes, and NEC NDL2102 avalanche photodiodes, are specified to operate in excess of 1 GHz.

2) The percentage of overlap for coincident pulses must be within tolerances determined by the detection bandwidth. The percentage of overlap is constrained by the synchronization limits on pulse generation and by variations in optical path length. Although synchronization of the pulse generators appeared to be a significant problem, the results of the experiments have shown that the detection of coincidence is more affected by the signal-to-noise ratio at the detector than by degree of overlap.

Thus, provided that sufficient optical power can be delivered to the point of coincidence, overlapping pulses can be easily discriminated above a threshold at which single pulses are eliminated. This allows a relatively high tolerance for degree of pulse overlap.

3) The scale of coincident pulse structures is constrained by the optical power that can be delivered to individual detectors sites. In [1] it has been shown that using non-reciprocal couplers, with coupling ratios of up to 95/5, linear systems with up 64 taps are feasible. Two dimensional topologies can support up to 4096 taps, and higher dimensional topologies are also possible.

2.3. Extensions to General Multiprocessor Interconnections

As part of this research, several new optical switching structures which are appropriate for multiprocessor interconnection applications have been studied. In general, multiprocessor interconnection structures can be classified as either broadcast systems, which are typically used in shared memory implementations, or point-to-point systems which support direct communication between processors or between processors and memory. It has been suggested by Levitan[3] that from a computational complexity standpoint, it is desirable to additionally support multicasting and simulcasting modes of communication. These modes are not widely implemented in electronics due to the complexity of their implementation. However, using optical techniques such structures can be realized efficiently.

The implementation of these structures exploits two properties of optical signals: unidirectional propagation and predictable path delays. These properties have allowed the use of the relative path length between two signals as a system timing mechanism. Further, the relation between time and space within a waveguide allows the positionally encoding information which normally requires complex decoding structures.

Table 1 is a classification of communication structures based on varying levels of connectivity and capabilities for a set of transmitting processors. A multicasting structure is one in which a transmitter sends a single message to a specific subset of m receivers where $m \leq n$, the number of processors in the system. Unlike broadcasting where all receivers actively interpret every message, multicasting provides that only the intended receivers interpret the message. This requires that some of the work in interpreting a message destination is done by the communications subsystem rather than by using resources in unintended receivers. Simulcasting, by our definition, is the concurrent transmission of n unique messages by a single transmitter to each of the n receiving processors. Multicasting and simulcasting may be generalized to the case where n transmitters are each multicasting (or simulcasting) concurrently. These cases are referred to as $n-way$ multicasting and $n-way$ simulcasting, respectively.

Number of Senders	Number of Receivers per Sender	Message Type per Sender	Comments
1	1		point to point
1	m	same msg.	multicast
1	n	same msg.	broadcast
1	n	different msgs.	simulcast
n	1		permutation/complete
n	m	same msg.	n-way multicast
n	n	same msg.	n-way broadcast
n	n	different msgs.	n-way simulcast

Table 1, Interconnection Structures for n Communicating Processors
(m less than n)

The point to point, broadcast, and completely interconnected structures have been implemented with different degrees of success in multiprocessor systems. Multicasting, simulcasting, and the *n*-way structures, have not been extensively examined because of the hardware complexity of their realization. In this research, several specific applications of coincident pulse techniques which realize multicasting and simulcasting have been developed [2.

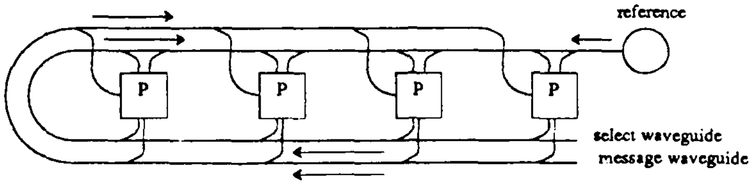


Figure 3 - 1-dimensional Multicasting Interconnection

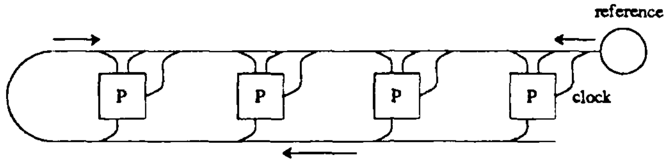


Figure 4 - 1-dimensional Simulcasting Interconnection

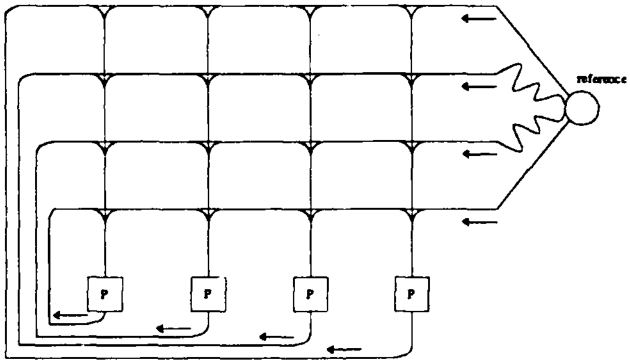


Figure 5 - 1-dimensional Array Simulcasting Interconnection

Figure 3 is an example of a 1 to *m* multicasting structure. This figure shows a bus interconnected multiprocessor with separate optical interconnections for address and data. The unique feature of this structure is the use of coincident pulse techniques in the implementation of the address bus. In each cycle, one transmitting processor places on the select waveguide a positionally encoded set of destination address bits followed by an *n*-bit message on the message waveguide. Simultaneously, a reference pulse propagates in the opposite direction in the select waveguide and coincides with the the destination address bits at each of the destination processors. Thus as the data propagates through the message waveguide, it is read only by those processors for which coincident pulse selections have been made.

A simple modification of the previous example results in the 1 to *n* simulcasting structure of Figure 4. In this case, the message waveguide has been removed and the interpretation of the address waveguide has been changed. Specifically, the select pulses in the address waveguide are now considered one bit data messages. The destination address of each message is positionally encoded by the relative position of the data bit in the select pulse train. Thus, a "one" is transmitted as a select pulse and a "zero" is the absence of a select pulse in the position corresponding to that receiver. The entire array is globally clocked to provide a strobe which moves the select information into a data latch at each receiver. This may be implemented using a separate copy of the reference pulse as a clock input, tapped as shown in Figure 4, with a small internal delay to allow for electronic propagation in the data latch. In both of these structures, A control arbitration mechanism is assumed to exist [1] such that only one processor is allowed to transmit in a given cycle.

Both multicasting and simulcasting can be extended to n-way structures. One method, is to allow all transmitters to transmit at the same time, and ensure that the optical path length between adjacent transmitters is greater than the length of the select pulse train, D, where $D \geq 2nd$, and d is the optical path length between any two adjacent receivers. For multicasting, this restricts the length of the message to be less than D. A second method for extending multicasting removes this restriction at the expense of more complex bus arbitration hardware. As discussed in [1] a control mechanism can be implemented such that a group of messages from an arbitrary subset of transmitters can be pipelined onto the data bus in a single cycle.

A second method to extend the 1 to n simulcasting structure of Figure 4 to an n-way simulcasting structure, is shown in Figure 5. In this example, an array of select waveguides connects individual busses attached to each processor. The unique feature of this structure is that the coincidence points are no longer detectors, but rather passive couplers which merge the coincident pulses into the receiving bus for each respective processor. Selection within each row of the array operates as in the previous structure relative to the transmitting processor attached to that row. The receiving busses are arranged in columns, perpendicular to the direction of propagation of the reference pulses in the transmitting busses. Therefore, the reference pulses arrive at all sites along a receiving bus simultaneously. The resulting data pulse train on the receiving bus is thus formed by coupling-in the message bits at specific optical path distances corresponding to the vertical separation of selection points. Each receiving bus thus contains an n-bit pulse train consisting of one bit from each of the transmitting processors. An advantage of this structure is that there is no need for any arbitration. Only a simple clocking mechanism is needed to delimit bus cycles.

2.4. Two Dimensional Arrays

By generalizing the propagation of pulses in one dimension to the propagation of linear wavefronts moving through a series of parallel waveguides, we can construct two dimensional structures. Hence, the method of addressing a location by programming the intersection of pulses may be generalized to addressing a location in a two-dimensional array by programming the intersection of wavefronts.

Consider 2-dimensional arrays similar to the one shown in Figure 6. An array of size n is composed of $\sqrt{n} \times \sqrt{n}$ cells separated by a distance $d = \tau c_g$ in both the vertical and the horizontal directions. The coincidence mechanism is the same as the linear example, except that the coincidence of three optical signals is required. Specifically, a reference wavefront generated by the reference diode L_{ref}, a select pulse train L_{col}, each traveling horizontally and in opposite directions, and another select pulse train, L_{row}, traveling vertically.

The optical signal generated by each source is decoupled from the source fiber by a star connection into $\sqrt{n}$ signals that travel through the array in parallel waveguides. Since the optical path length of all legs in the star will be equal, the wavefront will arrive at all locations in a single row (or column) simultaneously. For example, an optical pulse generated by L_{ref} and directed horizontally through the array will simultaneously arrive at all locations in column j. Similarly, any pulse generated by L_{col}, will also arrive, simultaneously, at all the cells in column j, and any pulse generated by L_{row} will arrive, simultaneously, at all the cells in

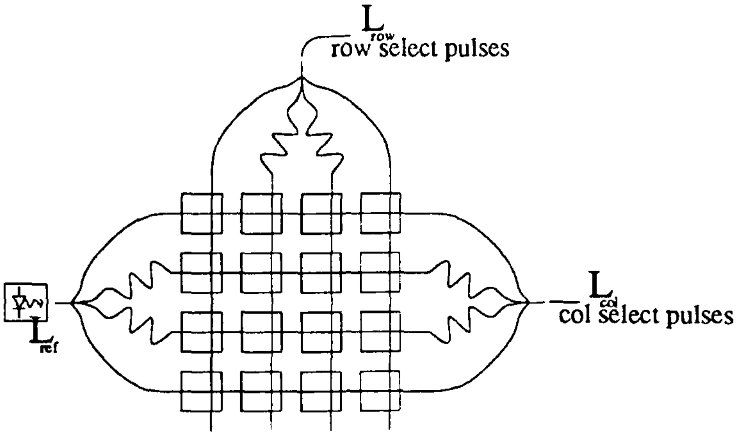


Figure 6: A Two Dimensional Structure

row i.

In order to derive the equations that govern the intersections of three wavefronts, assume, as in the case of the linear array, that L_{ref} generates a pulse of duration τ at time t_{ref}, and that L_{col} and L_{row} generate pulses at times t_{col} and t_{row}, respectively. If the timing of L_{col} is such that

$$t_{ref} - t_{col} = (\sqrt{n} - 1 - 2(j-1))\,\tau \qquad (1)$$

then, the two wavefronts generated by L_{ref} and L_{col} will meet at column j of the array. In order to select a particular cell i,j in that column, the third wavefront, namely the one generated by L_{row}, should be crossing row i when the other two wavefronts meet at column j. This may be accomplished by timing L_{row} such that

$$t_{ref} - t_{row} = (j - i)\,\tau \qquad (2)$$

In other words, to address a certain location i,j, the column number j is encoded as $t_{ref}-t_{col}$ and the difference, $j-i$, between the column number and the row number is encoded as $t_{ref}-t_{row}$. From (1) and (2), it may be shown that

$$-(\sqrt{n} - 1)\,\tau \le t_{ref}-t_{col} \le (\sqrt{n} -1)\tau$$

and

$$-(\sqrt{n} - 1)\,\tau \le t_{ref}-t_{row} \le (\sqrt{n} - 1)\,\tau$$

and hence, the latency time, σ, is

$$\sigma = 2\sqrt{n}\;\tau \qquad (3)$$

Using the above scheme, it is possible to encode the addresses of all of the n cells in the column and row pulse trains during a single cycle. In the one-dimensional case, the cycle time was directly proportional to the size of the array. This was because each cycle needed to provide an optical time-base slot for each location. In the two-dimensional case, cycle time is proportional to the square root of the number of cells. The price pa.J for this reduction in cycle time is the potential for overlap in parallel selection. This results from a requirement that corresponding select bits in each of the select waveguides be uniquely paired such that only the coincidence of paired bits are considered to be appropriate selections. Coincidences occurring from the intersections of non-paired bits will be referred to as *shadows*.

For example, if the two cells (i,j) and (l,k) are selected during the same bus cycle, then a shadow will appear at cell $(l+j-k,j)$ as shown in Figure 7a. This is because the selection of position j in the column

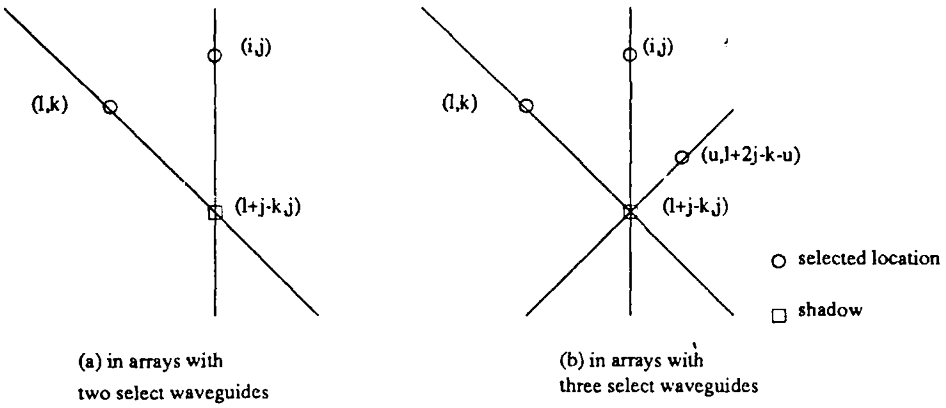

(i,j)
(l,k)
(l+j-k,j)
(a) in arrays with two select waveguides
(i,j)
(l,k)
(u,l+2j-k-u)
(l+j-k,j)
(b) in arrays with three select waveguides
○ selected location
□ shadow

Figure 7: Wavefront Intersections to Cause Shadow Selections

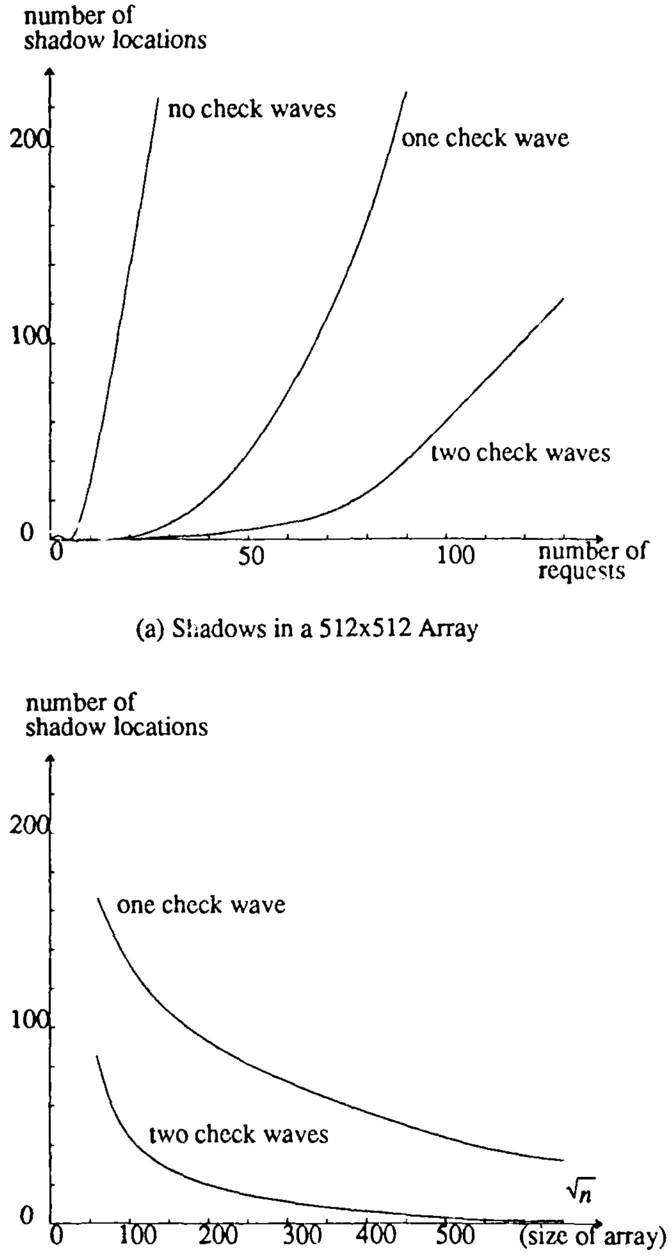
select train causes a coincidence with the reference wavefront at every cell along that column. This partial coincidence pattern is referred to as a *trace*. Similarly, the selection of bit $k-l$ in the row select pulse train will cause a partial coincidence trace with the reference wavefront along the diagonal passing through cell (l,k) as shown in Figure 7a. Therefore, cell $(l+j-k,j)$ which resides at the intersection of these two traces will see a coincidence of the reference wavefront with each of the select wavefronts, and hence, will be falsely selected.

Since shadows occur at the intersections of two traces corresponding to the vertical and horizontal select wavefronts, the number of such intersections can be reduced by the addition of a third select wavefront, referred to as the check wavefront. With this, a valid selection occurs on the coincidence of the reference wavefront with select bits in each of the three wavefronts. In this case, shadows are generated at the intersection of the two traces mentioned and the additional trace corresponding to the check wavefront in Figure 7b. As shown by the simulation results in Figure 8a, this greatly reduces the number of shadows generated in the array. Using the same argument, by the addition of a fourth wavefront, a second check wavefront, it is possible to reduce even further the occurrence of shadows.

In the above scheme the select and check pulse trains are each of length $2\sqrt{n}$. Thus, the total number of bits transmitted to the array in a single cycle is of order $O(\sqrt{n})$. However, the following proposition shows that the number of bits required to distinguish a unique set of parallel selections is n.

> **Proposition:** *The minimum number of bits required to uniquely select an arbitrary set of cells from a collection of size n is n.*
>
> **Proof:** For a set of cells S of size n, the size of the power set $P(S)$ is 2^n. Using a binary encoding, the enumeration of $P(S)$ requires $\log_2 2^n = n$ bits per code word. Assume that a binary encoding scheme exists which can address all possible subsets of S using a code of length less than n-bits. Such a scheme would result in an enumeration of less than 2^n subsets. Therefore there must exist in that encoding at least one code which corresponds to more than one element in $P(S)$. Hence, such a system does not uniquely address all subsets. []

This result introduces a restriction on the application of 2-dimensional structures. If n bits are to be used as the proposition implies, n bits must either transmitted to the array serially, in which case latency is comparable to the 1-dimensional case, or additional waveguides must be added, thus increasing the hardware complexity.



(a) Shadows in a 512x512 Array

(b) Shadows Generated by 50 Requests

Figure 8: Simulation Results for the Incidence of Shadows

Alternatively, the number of concurrent accesses may be restricted to some number m such that $m \ll n$. As shown in Figure 8 the incidence of shadow selections is dependent on the size of the array relative to the number of requests and the number of wavefronts in the selection structure. Therefore, 2-dimensional arrays are most appropriate in computational structures where the number of potential receiving sites is much greater than the number of transmitting sites. This would be the case in the design of an m-ported memory of size n. For example, simulation results show that a 256k memory implemented in a $512x512$ square array with two select and two check wavefronts can be operated as a 50 ported memory with an average of 4.6 shadow locations for 50 simultaneous requests. In a memory application these few shadows would appear as extra read requests which should be discarded. The problem of shadows for write requests does not exist when writes are restricted to a single request per cycle. This results in a concurrent-read-exclusive-write model for shared memory multiprocessors.

In applications where the restriction $m \ll n$ cannot be met, a structure can be provided which eliminates shadows by the transmission of $O(n)$ bits of selection information into the array. In this structure, shown in Figure 9, the $\sqrt{n}$ row select waveguides are kept distinct. On each waveguide a unique pulse train of $2\sqrt{n}$ bits is transmitted. Each train contains only the select information for access on that row. There is no longer a need for the vertical waveguides. The row waveguides share a common set of reference pulses generated as in the previous 2-dimensional example. Thus, each row pulse train will have all the information (and only the information) for selections on that row. In effect each row is an independent linear structure of size $\sqrt{n}$.

One structure for the generation of the row pulse trains is shown in Figure 10. In this figure a set of m transmitters are each connected to a linear structure where the coincident points on those structures are optical repeaters, which detect pulse coincidence and re-transmit into the row select pulse trains. Unlike the linear structures, only a single pulse is allowed to travel in each direction through the structure. In addition, the timing of both pulses is varied in order to achieve both the appropriate row location and relative timing of the pulse coincidence.

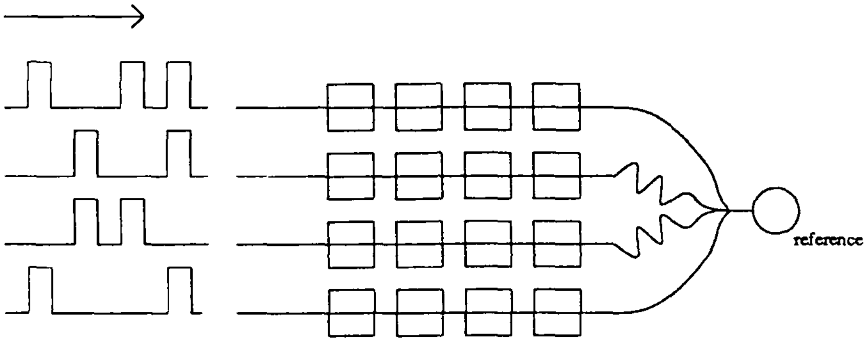


Figure 9: Parallel Access 2-dimensional Selection Structure

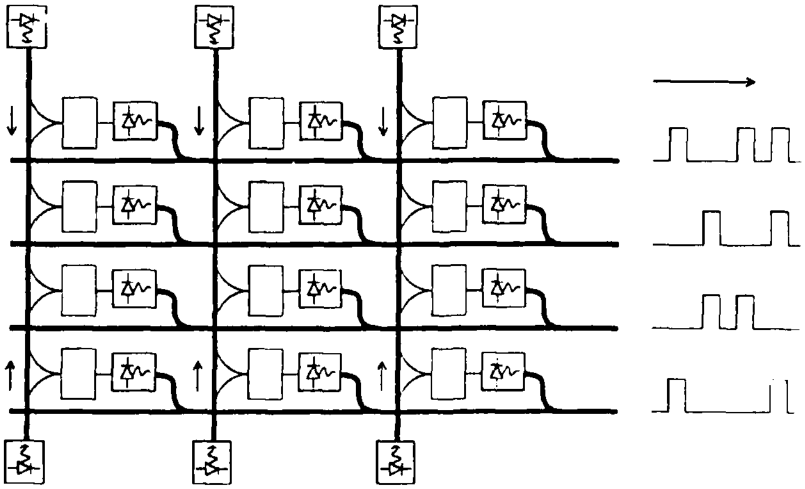


Figure 10: Parallel Address Encoder

In order to derive the equations governin.: the relative timing of the pulses generated at a transmitter k, $1 \leq k \leq m$, for the selection of location i,j (see Figure 11), assume that the reference pulse is fired at time t_{ref}

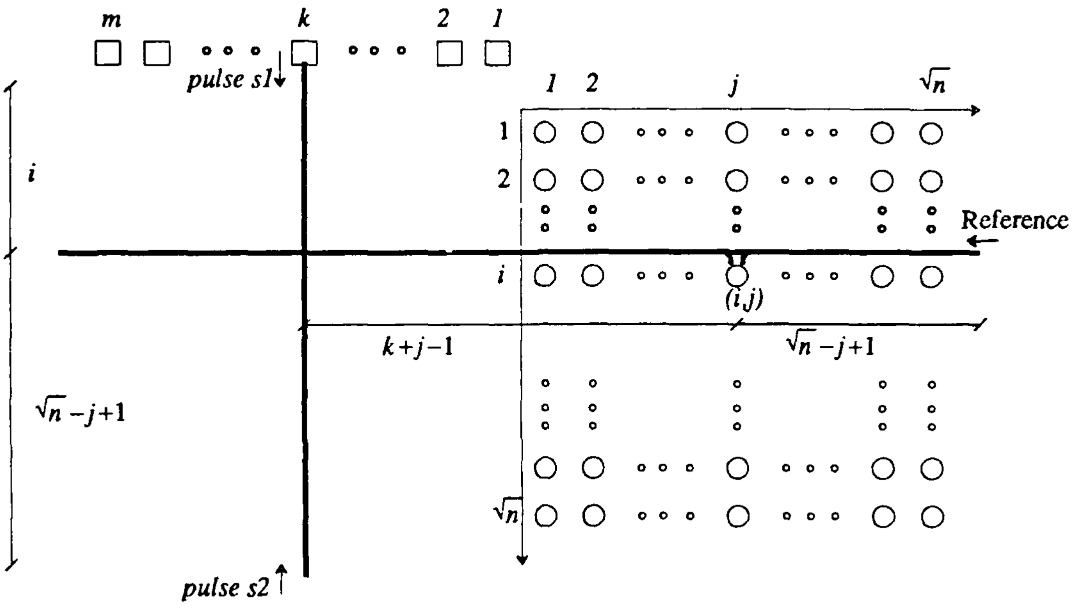


Figure 11: Pulse Path Lengths for the Memory Structure of Figures 9 and 10

and the two select pulses are fired at time t_{s1} and t_{s2}. Given that the reference pulse will be at location i,j at time $t_{ref}+(\sqrt{n}-j+1)\tau$, the two select pulses should be at that location at the same time. That is:

$$t_{s1} + (i+k+j-1)\,\tau + \tau_{rep} = t_{ref} + (\sqrt{n}-j+1)\tau$$

$$t_{s2} + (\sqrt{n}-i+k+j)\,\tau + \tau_{rep} = t_{ref} + (\sqrt{n}-j+1)\tau$$

Where, τ_{rep} is the delay introduced by the optical repeater circuit shown in Figure 10. Therefore, relative to the reference pulse, for selecting an address i,j, the select pulses must be fired at times:

$$t_{s1} = t_{ref} + (\sqrt{n}-2j+2-i-k)\,\tau - \tau_{rep}$$

$$t_{s2} = t_{ref} + (i-k-2j+1)\,\tau - \tau_{rep}$$

In this manner up to m selections, one per processor, can be made simultaneously to a memory of n cells. The worst case latency is $(3\sqrt{n}+m)\tau+\tau_{rep}$, which is the time for a pulse to travel from a transmitter to a selected cell. However, using pipelining, new selections can be generated with a cycle time of $2\sqrt{n}\,\tau$.

All the structures presented in this report are based on two simple components, the 1D and the 2D selection arrays. Current research supported by AFOSR* is to determine whether structures built from these components will substantially outperform their electronic counterparts. To this end, prototypes of the two component structures are being constructed and will be configured as described above.

References

1. D. Chiarulli, S. Levitan, and R. Melhem, "Asynchronous Control of Optical Busses in Closely Coupled Distributed Systems," *Journal of Parallel and Distributed Computing (to appear).*

2. S. Levitan, D. Chiarulli, and R. Melhem, "Coincident Pulse Techniques for Multiprocessor Interconnection Structures," Technical Report 89-22, Dept. of Computer Science, University of Pittsburgh,

*) Coincident Pulse Techniques for Hybrid Optical-Electronic Computer Systems, AFOSR 89-0469

Pittsburgh, PA, 1989.

3. Steven P. Levitan, "Measuring Communication Structures in Parallel Architectures and Algorithms," in *The Characteristics of Parallel Algorithms*, ed. L. Jamieson, E D. Gannon, R. Douglass, pp. 101-137, MIT Press, Cambridge, MA, 1987.

3. Publications

Archived Journals

D. M. Chiarulli, R. Melhem, and S. P. Levitan, "Asynchronous control of optical busses for distributed multiprocessors" (to appear) *Journal of Parallel and Distributed Computing.*

R. Melhem, D. Chiarulli, and S.P. Levitan, "Space multiplexing of optical waveguides in a distributed multiprocessor" *The Computer Journal, British Computer Society*, vol 32, no 4, 1989.

S. P. Levitan, D. Chiarulli, and R. Melhem "Coincident Pulse Techniques for Multiprocessor Interconnection Structures" (submitted), *Applied Optics* feature issue on optical computing.

Conference Proceedings and Technical Reports

D. M. Chiarulli, R. Melhem, and S. P. Levitan, "Optical bus control for distributed multiprocessors" Technical Report #88-2, Department of Computer Science, University of Pittsburgh, Pittsburgh, 1988.

S. P. Levitan, D. Chiarulli, and R. Melhem "Coincident Pulse Techniques for Multiprocessor Interconnection Structures" Technical Report #89-22, Department of Computer Science, University of Pittsburgh, Pittsburgh, 1989.

4. Personnel and degrees awarded

Principle Investigators

Donald Chiarulli, Assistant Prof., Co-PI
Rami Melhem, Associate Prof., Co-PI
Steven Levitan, Assistant Prof., Co-PI

Graduate Students

David Carson, Graduate Assistant,
Angelos Leventoplous - Graduate Assistant,
Zicheng Guo - Graduate Assistant,

Theses and Dissertations

Greg Owens, M.S. May 1988, Thesis: "A simulation of Data Collection Topologies in an Optically Addressed Memory".

5. Other supporting Material

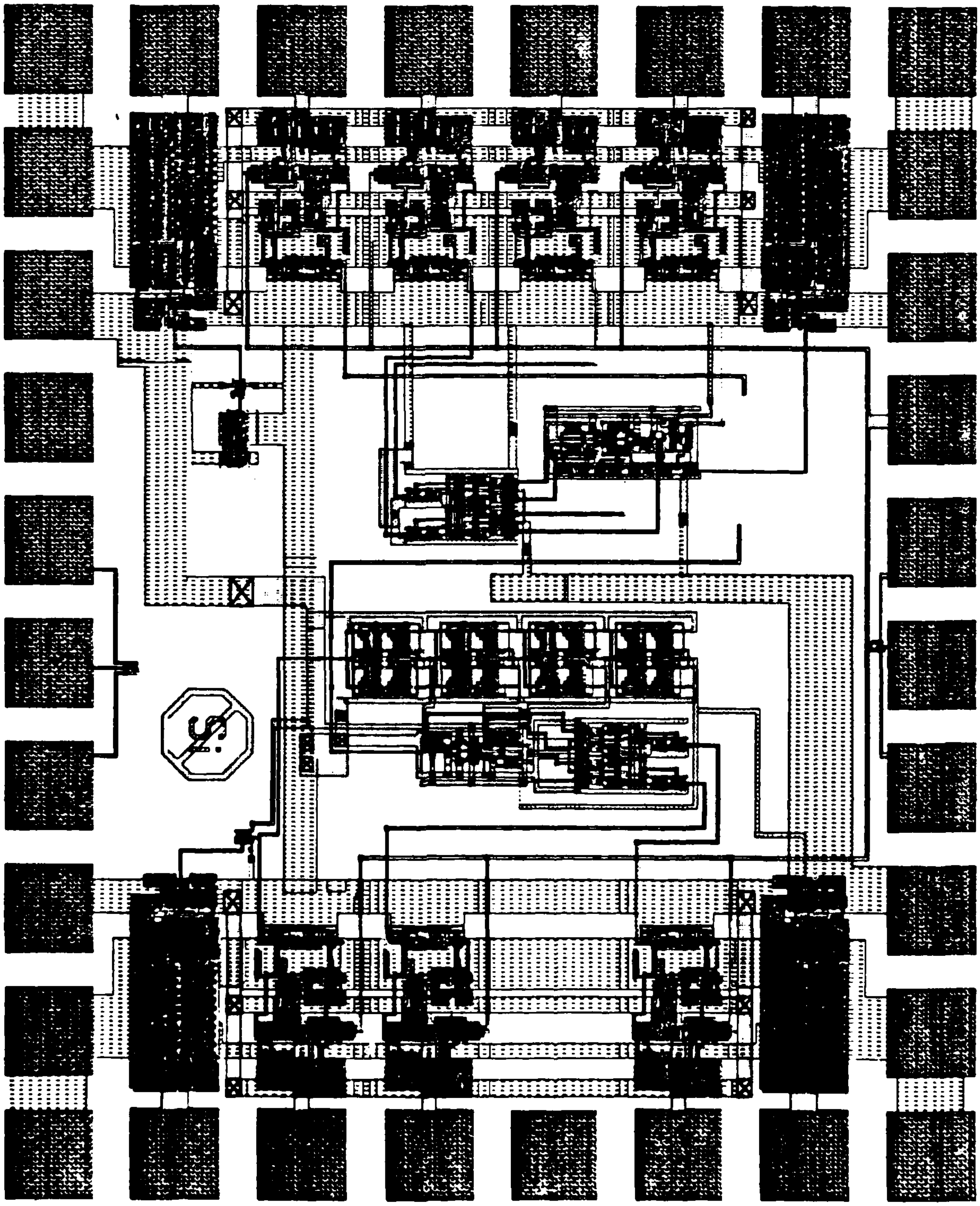
A DARPA sponsored GaAs IC design course was held at University of California, Santa Barbara during July and August 1989. The purpose of the course was to provide an intensive introduction to the design of GaAs digital ICs using the foundry services which will be made available through MOSIS/ISI. Levitan (one of the PI's of the grant) participated in the design course. The course emphasized full custom design of GaAs digital ICs. The MAGIC layout editor, SPICE, and associated verification tools were utilized for the design projects.

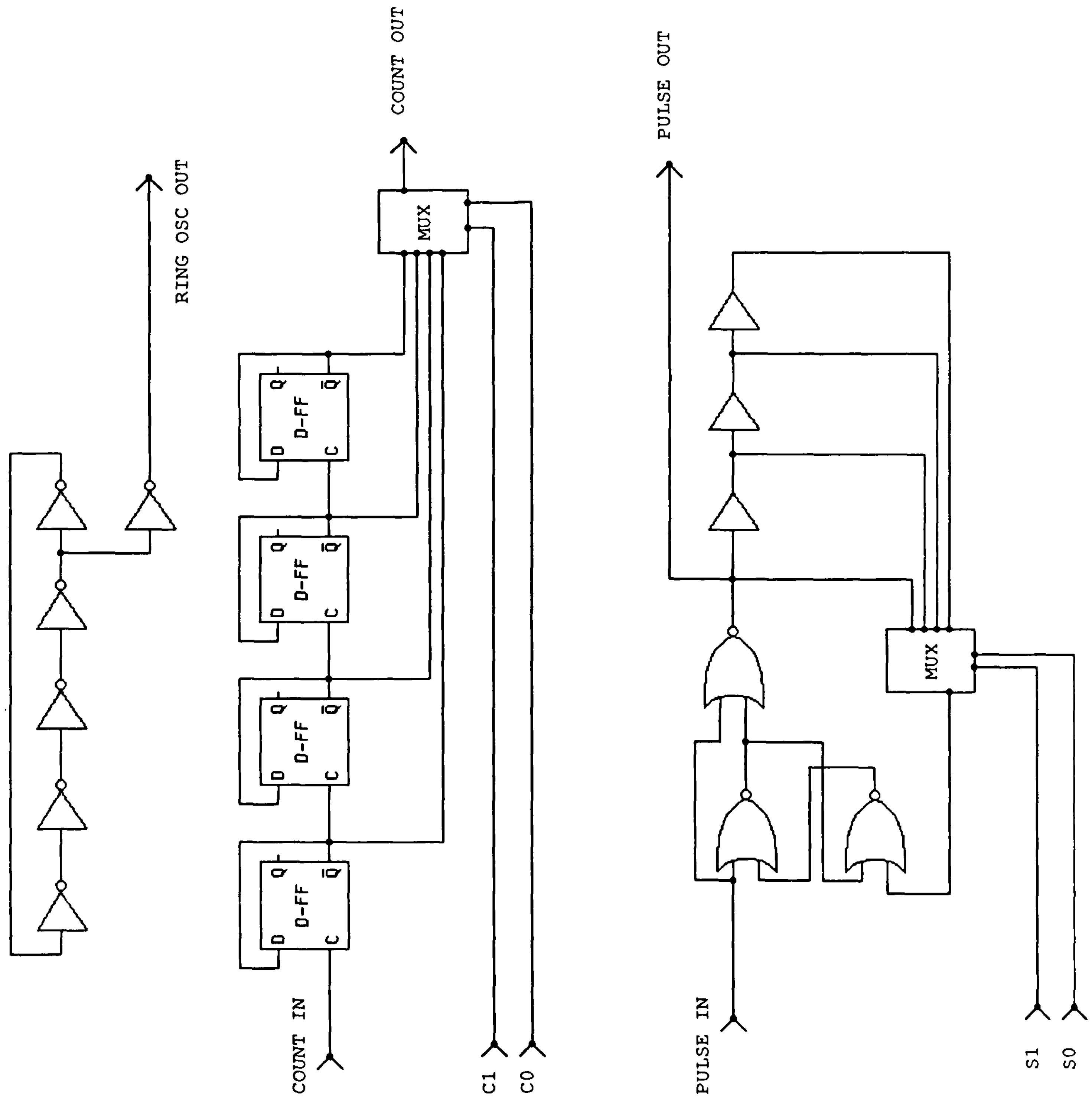
The course covered the following topics:

1. Introduction to GaAs devices and models
2. Logic design principles and examples
3. Interconnection design rules
4. Test procedures for GaAs digital ICs
5. Layout and simulation tools
6. Layout design rules and the MOSIS interface
7. A design project

Projects from the course were submitted for fabrication to MOSIS and will be made available for testing in packaged form in the fall of 1988. The project chosen for Levitan's group was a simple digital function generator. The current version of this design is shown in figures 12 and 13. This circuit is designed to provide high speed electronic pulses, in the .8ns to 4ns range, driven from a high speed counter. This circuit will provide a cheap method of providing pulses into the optical system.

The significance of this relative to the sponsored research is that GaAs technology is the natural medium through which coincident pulse systems would be implemented. The high speed, low power, and electro-optic characteristics of GaAs will in the near term provide a mechanism for high speed electronic modulation using pulsers such as the one shown. In the longer term is will be the basis for an integrated electronic/optical solution which will provide for high speed electronic computation and high bandwidth optical communications on the same device.




RING OSC OUT
COUNT OUT
MUX
D-FF
D
Q
C
Q̄
COUNT IN
C1
C0
PULSE OUT
MUX
PULSE IN
S1
S0

Appendix: Reprints and Preprints of Related Publications

1) Using Coincident Optical Pulses for Parralel Memory Addressing.

2) Optical bus control for distributed multiprocessors.

3) Space multiplexing of optical waveguides in a distributed multiprocessor.

4) Coincident Pulse Techniques for Multiprocessor Interconnection Structures.

Using Coincident Optical Pulses for Parallel Memory Addressing

Donald M. Chiarulli, Rami G. Melhem, and Steven P. Levitan

University of Pittsburgh

Common-bus, shared-memory multiprocessors are the most widely used parallel processing architectures. Unfortunately, these systems suffer from a memory/bus bandwidth limitation problem. For the designer of a hybrid optical/electronic supercomputer, an immediate temptation is to replace the shared electronic bus with an optical analog of higher bandwidth. To make that replacement is only a partial solution. The true bottleneck in such systems is in the address-decoding circuits of shared memory units.

In this article we propose a new memory structure that provides for parallel access in a multiprocessor environment. The proposed system has two advantages. First, it distributes the address-decoding circuitry to each of the requesting units on a common bus, thus eliminating the bottleneck of centralized decoding of encoded memory addresses. Second, it allows for parallel fetches of memory data with a level of parallelism limited only by the ratios of optical to electronic bus bandwidths and the dimensionality of the memory array.

By distributing the address-decoding function to the requesting units on an optical bus, this new memory structure eliminates the bottleneck of centralized decoding in multiprocessor environments.

In a conventional electronic memory circuit, like the one shown in Figure 1, an incoming memory address is divided into row and column addresses, each decoded separately. The selected memory location is the intersection of the select lines generated by the row and column decoders. In common-bus multiprocessors these decoders have traditionally been a performance-limiting bottleneck. Each decoder can process only a single encoded address, thus limiting memory access to a single location. Memory interleaving techniques,[1] which subdivide the memory space into regions, each in a separate memory unit, are commonly applied in an attempt to make parallel some subset of memory accesses. More recently, sophisticated cache memory systems,[2] which physically reproduce portions of shared memories in a local store, have been developed. Both systems have obvious limitations. Interleaved systems impose an ordering in which parallel accesses to a shared memory can be made, and cache memories rely on the locality of memory references for each processor and require a large overhead to support cache coherence.

Our solution is to distribute the address-decoding function to the requesting devices, thereby breaking contention for monolithic address decoders. This solution requires the abandonment of conventional encoded addresses as a mechanism for conserving bus bandwidth. Rather, we use the high bandwidth of optics to time-multiplex fully decoded addresses into an optical "select" pulse train. Using a technique based on the coincidence of optical pulses, we can directly apply the optical

0018-9162/87/1200-0048$01.00 © 1987 IEEE

select pulse train to a memory array to address one or more cells. Effective parallelism is possible in this system because of the differential between optical and electronic bandwidths. Within a single electronic memory access cycle, N parallel memory references are possible, where N is limited only by the ratio of optical to electronic bandwidths. For a fixed bandwidth ratio the size of memory that can be constructed is further determined by the dimensionality of the memory structure. The technique requires no active optical or electro-optical switching devices. It uses only the mature technologies of optical sources, waveguides, and photodetectors. In the two-dimensional form, the system is well adapted to an integrated optics implementation.[3,4]

The addressing mechanism, which we call *optical pulse delay modulation*, is based on the use of time delays between optical pulses. The optical pulses are propagated through waveguides in several directions through the memory array. By appropriately adjusting the delays, we can make these pulses coincide at specific memory cells. This coincidence is detected by photodetectors at the addressed locations, thereby selecting those locations for memory access.

Our primary interest is in the application of this addressing mechanism to two-dimensional, multiported memory modules. Such structures are composed of horizontal and vertical waveguides with n memory cells located at the intersecting points. With proper cell layout we can access up to $\sqrt{n}$ memory cells concurrently by sending a sequence of pulses in the horizontal and vertical waveguides. In a multiprocessor environment a sequence of pulses, each corresponding to a distinct memory reference, is generated by independent address decoders located at each of the processing units. Thus, the address-decoding function is completely distributed to the requesting processors, and there is no address-decoding circuitry at the memory unit.

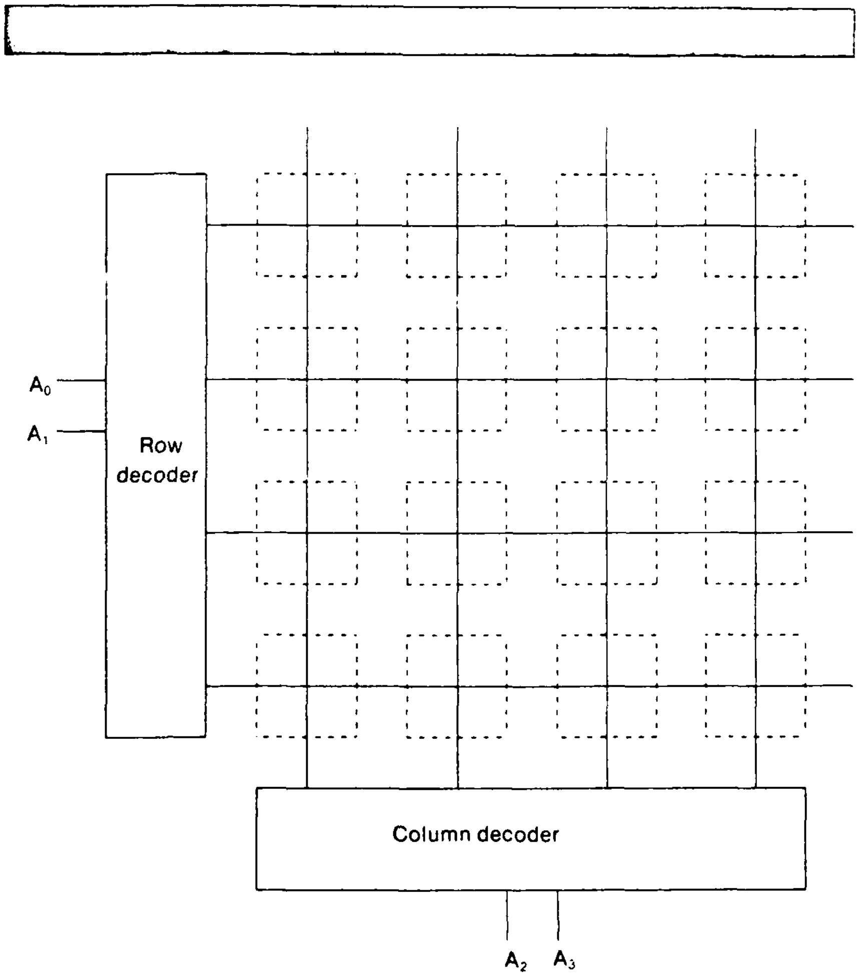


Figure 1. Conventional electronic memory structure.

A one-dimensional memory array

In this section we introduce the technique of pulse delay addressing, using a one-dimensional memory array as an example. This example is not ideal because both the hardware complexity and access time grow in proportion to the size of the memory. This is not the case for the higher-dimensional structures presented in later sections.

Optical pulse delay addressing. As shown in Figure 2, a memory module is composed of n cells $C_1, \ldots, C_n$, each storing one bit of information. The select signal for each cell C_k is an electronic pulse at the output of a photodetector D_k. The photodetector generates the logical OR of two incident optical signals, denoted in Figure 2 by s_1 and s_2.

The signals s_1 and s_2 travel in opposite directions along an optical path, which can be either an optical fiber or a planar waveguide in an integrated optical device.[5] Photodetectors are placed at fixed distance intervals d along the optical path, and two laser diodes, L_1 and L_2, are coupled to each end. Both laser diodes are normally *on* and the circuits of all detectors normally generate a logic *one*.

Assume that two dark pulses of duration τ are transmitted, one from L_1 and the other from L_2 at times t_1 and t_2, respectively. These pulses represent "dark spots" propagating at speed c_g (the speed of light in the waveguide). By carefully selecting the delay between t_1 and t_2, we can make the two dark spots meet at exactly one detector. This detector will then turn off, generating a logic *zero* of duration τ. The distance d between any two detectors is chosen to be equal to d =

τc_g, the propagation distance corresponding to the pulse duty cycle. The delay $t_1 - t_2$ is also chosen such that it is an even multiple of d. More specifically, if

$$t_1 - t_2 = (n - 1 - 2(k - 1))\tau \qquad (1)$$

then the two dark spots will meet at detector D_k, thus addressing cell k. For example, when $n = 5$, if L_2 generates its dark pulse 2τ seconds before L_1 generates its pulse, then Equation 1 gives $k = 2$; that is, the two pulses meet at D_2. Similarly, if L_2 generates its pulse 2τ seconds after L_1 generates its pulse, then the two pulses meet at D_4. Clearly, we choose the middle cell by generating the' two pulses simultaneously—that is, by having $t_1 = t_2$. Therefore, the address of the cell is encoded by means of the delay $t_1 - t_2$. In this view, the pulse generated by L_1 may be defined as a reference pulse, and the pulse generated by L_2 becomes a select pulse. In the remaining discussion the terms t_{ref}, L_{ref}, t_{sel}, and L_{sel} will refer to t_1, L_1, t_2, and L_2, respectively

The memory access time is determined by the maximum delay needed to address any cell in the array. From Equation 1, it is clear that for $k = 1, \ldots, n$, we have

$$-(n-1)\tau \le t_{ref} - t_{sel} \le (n-1)\tau \qquad (2a)$$

from which we find that the memory access time, σ, is given by

$$\sigma = 2n\tau \qquad (2b)$$

Note that Equation 2a indicates that the select pulse occurs within $n\tau$ before or after the reference pulse.

The parallelism in this addressing scheme comes from the fact that within time σ it is possible to address more than one cell by sending a series of pulses from L_{sel}, one for each memory reference. Each of these pulses will intersect with the reference pulse at the desired detector. In other words, parallel memory references are positionally distinguishable in a pulse train generated by a series of select pulses.

Before we describe how this memory can be incorporated into a shared-memory multiprocessor, two design issues at the interface between the electronic processing units and the optical system must be resolved. First, a system for generating a series of optical pulses corresponding to

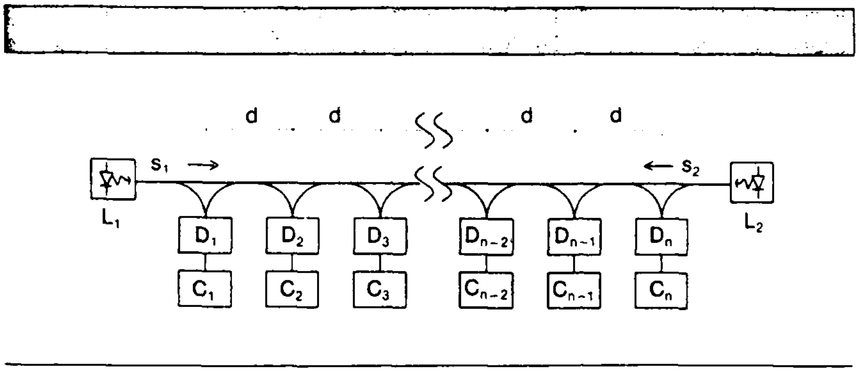


Figure 2. Linear memory structure.

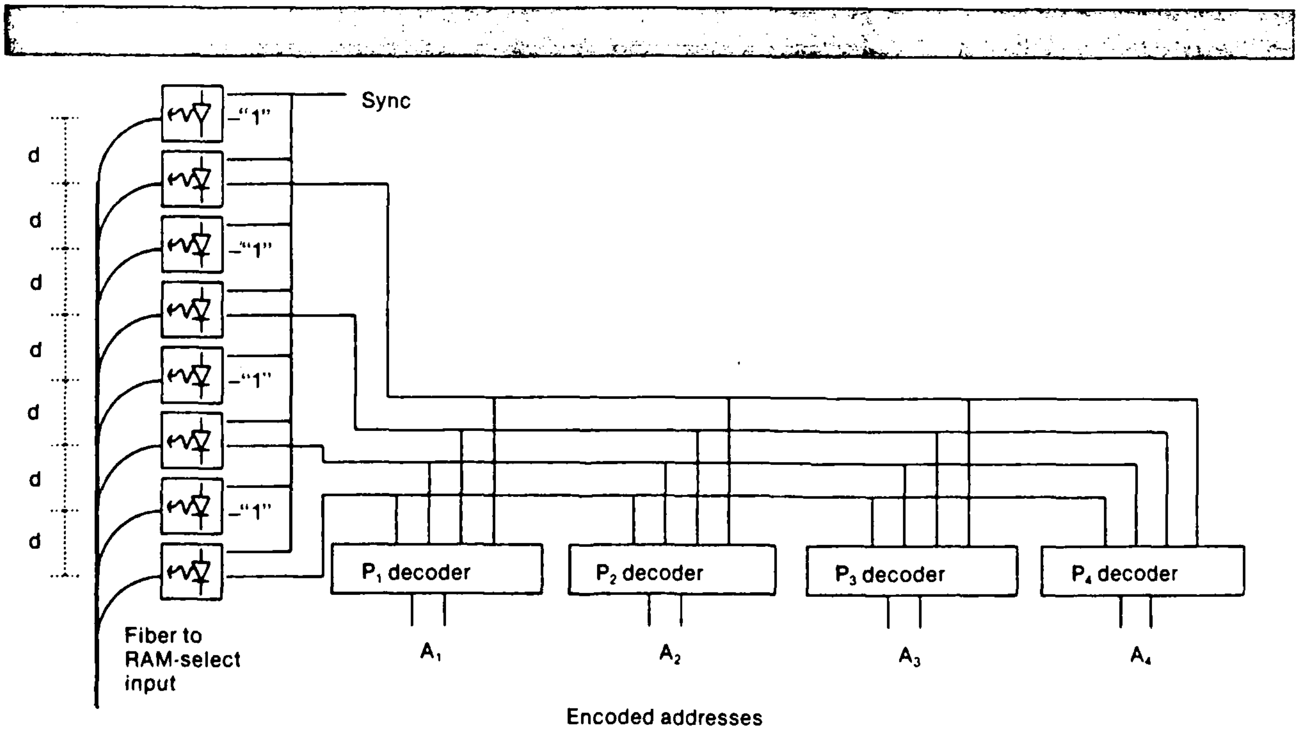


Figure 3. Distributed address generator.

the select pulse train must be specified. Second, the memory must allow data stored in the referenced locations to be returned to the requesting processors within a single processor memory cycle.

Address generation. One of the major advantages of the proposed memory organization over conventional systems is the removal of the address-decoding function from the memory unit and the distribution of this function to the requesting processors. More specifically, each of the processors is assumed to generate normal encoded addresses when referencing the shared memory. These addresses are decoded locally by an address decoder attached to each processor. The decoded addresses are electronically ORed onto a select bus common to all the processors. In the one-dimensional case the select bus consists of *n* lines, each controlling a laser diode pulser (see Figure 3). As will be explained later, the size of the select bus in the two-dimensional case reduces to a more manageable $2\sqrt{n}$ lines controlling $2\sqrt{n}$ laser diode pulsers.

Returning to the linear case, the optical pulse train containing the memory requests is generated by $2n$ laser diode pulsers spaced at incremental distances *d* in optical path length from the memory. To reconcile the difference between optical and electronic bandwidths, a single edge in the electronic time base (sync1 in Figure 3) controls the activation of all the pulsers such that all the optical pulses are generated simultaneously. If the duration of each optical pulse is equal to τ, then the select pulse train will be confined to $2n$ time slots, each with duration τ. Since proper time multiplexing of select pulses is not possible with the select pulsers constantly on, the *n* select pulsers connected to the electronic select bus are separated by *n* pulsers modulated by a fixed *one* signal (see Figure 3). Thus, all the slots will contain an optical pulse except those slots corresponding to requested memory addresses. More specifically, a dark spot (no pulse) at slot $2i-1$ corresponds to a request for memory location *i*.

In addition to the select pulse train, a dark reference pulse must be generated. Alternatively, the reference pulse is a series of $2n$ light pulses, with a single dark pulse at position *n*. Such a pulse train can be produced by a single laser diode that is normally *on* and is pulsed *off* for duration τ upon the reception of the synchronization pulse sync1. For the dark pulse in the reference train to coincide with pulse slot *n* in the select pulse train, the optical path from the reference diode to the memory should equal the optical path from the diode generating the middle pulse in the select train to the memory.

The above address generation scheme is crucially dependent on the simultaneous pulsing of several laser diodes. In an integrated optics environment, such synchronization problems can be avoided by the use of a single optical pulse source of duration τ and a series of electro-optic switches. In such an implementation each laser diode in Figure 3 is replaced by a directional coupler, which "couples in" the pulse at various optical path lengths. Synchronization problems are replaced in this system by a new set of issues relating to optical power distribution. We will discuss this and other issues relating to an integrated optics implementation in later sections.

Parallel memory read. The other issue at the interface between the electronic processing units and the optical system is a mechanism for returning the electronically stored data from the memory to the processors. The data is returned, on an optical bus, in a pulse train that consists of *n* slots, one for each memory location. Therefore, parallel accesses are positionally distinguishable in the read pulse train. When this pulse train arrives at a processor that has issued a read request for the *i*th position of the store, this processor will find the requested data in the *i*th slot of the data pulse train.

One method for generating the read data pulse train is to use a structure similar to the one depicted in Figure 4. In this structure, *n* laser diodes are placed on the optical data bus, separated by an optical distance *d*. When a specific memory location *k* is addressed, an electronic signal is generated from the photodetector as described earlier. The data at location *k* is assumed to be stored electronically and is used to modulate the *k*th laser diode only if location *k* is addressed. A synchronization signal, sync2, is used to synchronize the output of light (positive) pulses of duration τ from the selected memory locations that store a *one*. The difference in the optical path lengths between the laser diodes ensures the correct generation of the data pulse train. A similar technique, using detectors at fixed distances *d* and latching the pulse train at each processor interface, is used to demultiplex the pulse trains.

A two-dimensional memory structure

With the above mechanism it is possible to address all the *n* memory locations in one processor memory cycle. For the one-dimensional case this represents a sequential read of the entire store and requires a ratio of electronic to optical time bases equal to the size of the memory. Even for the most optimistic assumptions about achievable optical pulse widths, this structure will be inadequate and wasteful. However, applying a similar addressing mechanism to two-dimensional memory arrays reduces the required ratio of electronic to optical time bases to $\sqrt{n}$, where at

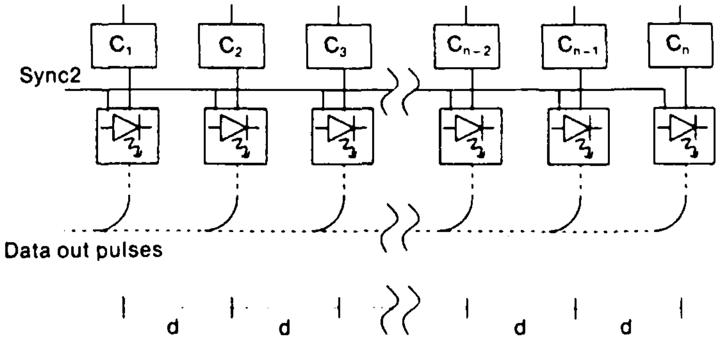


Figure 4. Generation of the data pulse train.

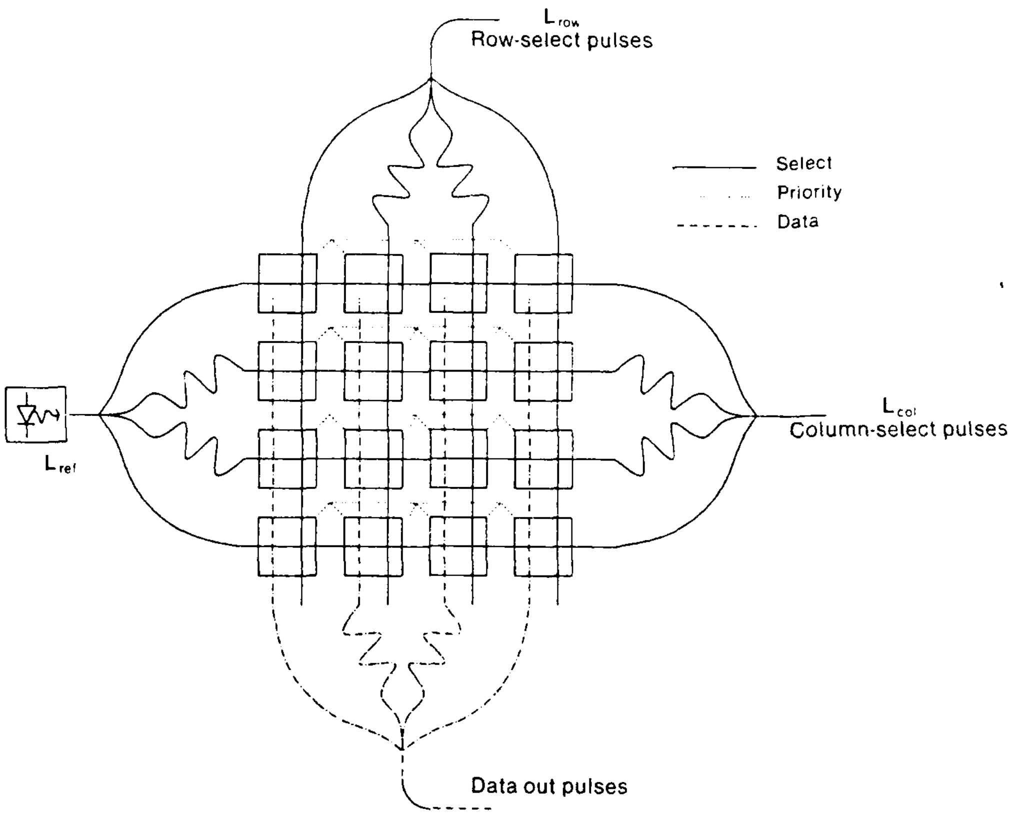


Figure 5. Two-dimensional memory structure.

most $\sqrt{n}$ memory locations can be addressed in one cycle. This allows for the construction of reasonable-size shared memories.

Coincident wavefront addressing. In the two-dimensional case we generalize the propagation of dark spots in one dimension to the propagation of linear dark wavefronts moving through a series of parallel waveguides. Hence, the method of addressing a location by programming the intersection of two dark spots can be generalized to addressing a location in a two-dimensional array by programming the intersection of three dark wavefronts. The literature on systolic and wavefront arrays (for example, H.T. Kung[6] and S.Y. Kung et al.[7]) suggests many possible ways for propagating and programming the intersection of wavefronts. Here we present a simple propagation scheme that can be used in two-dimensional memory addressing.

Consider two-dimensional memory arrays similar to the one shown in Figure 5. An array of size n is composed of $\sqrt{n} \times \sqrt{n}$ photodetector/cell units $DC_{i,j}$, $i,j = 1, \ldots, \sqrt{n}$, separated by a distance $d = \tau c_g$ in both the vertical and the horizontal directions. The structure of a DC unit is identical to the linear example, except that the input to the photodetector generates the logical OR of three optical signals: specifically, a dark reference wavefront generated by the reference diode L_{ref}, a select pulse train generated from a distributed set of column address decoders, L_{col}, both traveling horizontally in opposite directions, and a select pulse train generated by a distributed set of row select decoders, L_{row}, traveling vertically.

The optical signal generated by each source is decoupled from the source fiber by a "squid" connection into $\sqrt{n}$ signals that travel through the array in parallel waveguides. Since the optical path length of all legs in the squid will be equal, the wavefront will arrive at all locations in a single row (or column) simultaneously. For example, an optical pulse generated by L_{ref} and directed horizontally through the array will be simultaneously incident at locations $DC_{i,j}$, $i = 1, \ldots, \sqrt{n}$, in column j. Similarly, any pulse generated by L_{col} will arrive at all the cells in a specific column simultaneously, and any pulse generated by L_{row} will arrive at all the cells in a specific row simultaneously.

To derive the equations that govern the intersections of three wavefronts, assume, as in the case of the linear array, that all three laser diodes are *on* and that L_{ref} generates a dark pulse of duration τ at time t_{ref}. Also, L_{col} and L_{row} generate dark pulses at times t_{col} and t_{row}, respectively. If the timing of L_{col} is such that

$$t_{ref} - t_{col} = (\sqrt{n} - 1 - 2(j - 1))\tau \qquad (3)$$

then the two dark wavefronts generated by L_{ref} and L_{col} will meet at column j of the array. To select a particular cell $DC_{i,j}$ in that column, the third dark wavefront, namely the one generated by L_{row}, should be crossing row i when the other two wavefronts meet at column j. This can be accomplished by timing L_{row} such that

$$t_{ref} - t_{row} = (j-i)\tau \qquad (4)$$

In other words, to address a certain memory location i,j, the column number j is encoded as $t_{ref} - t_{col}$, and the difference, $j-i$, between the column number and the row number is encoded as $t_{ref} - t_{row}$. From Equations 3 and 4 it can be shown that

$$-(\sqrt{n}-1)\tau \leq t_{ref} - t_{col} \leq (\sqrt{n}-1)\tau$$

and

$$-(\sqrt{n}-1)\tau \leq t_{ref} - t_{row} \leq (\sqrt{n}-1)\tau$$

and, hence, the memory access time, o, is

$$o = 2\sqrt{n}\tau \qquad (5)$$

As in the linear case, parallel accesses are made possible by the generation of multiple pulses in the row and column select signals. For example, Figure 6a shows the pulse trains for the selection of memory locations (2,2), (1,4), and (4,4) in the 16-location memory array of Figure 6b. For these three locations, $t_{ref} - t_{col}$ should equal -1, 3, and 3, respectively, and $t_{ref} - t_{row}$ should equal 0, 3, and 0, respectively. The locations of the wavefront resulting from these trains at times 0, 5τ, and 7τ are shown in Figures 6b, 6c, and 6d. It is clear from the intersection of the dark fronts in these figures that location (2,2) is selected at time 5τ and locations (1,4) and (4,4) are selected at time 7τ.

With the above scheme it is possible to encode the addresses of all the n memory cells in the column and row pulse trains during a single memory access cycle. However, for a time-multiplexed memory read such as the one proposed earlier, the length of the return data pulse train, and hence the total read time, would grow linearly with memory size. To prevent this and to facilitate a pipelined implementation, we chose the maximum length of the read pulse train to be $2\sqrt{n}$, the length of the select pulse trains. This is actually an advantage of the two-dimensional structure. More specifically, in the one-dimensional case, the memory cycle time was directly proportional to the size of the store. Each cycle needed to provide an optical time base slot for each location. In the two-dimensional case, access time and the possible number of parallel accesses are proportional to the square root of the number of locations in the store. This is a far more realistic scenario for constructing a shared-memory multiprocessor system. The price paid for this reduction in access time is the potential for conflicts in parallel memory accesses.

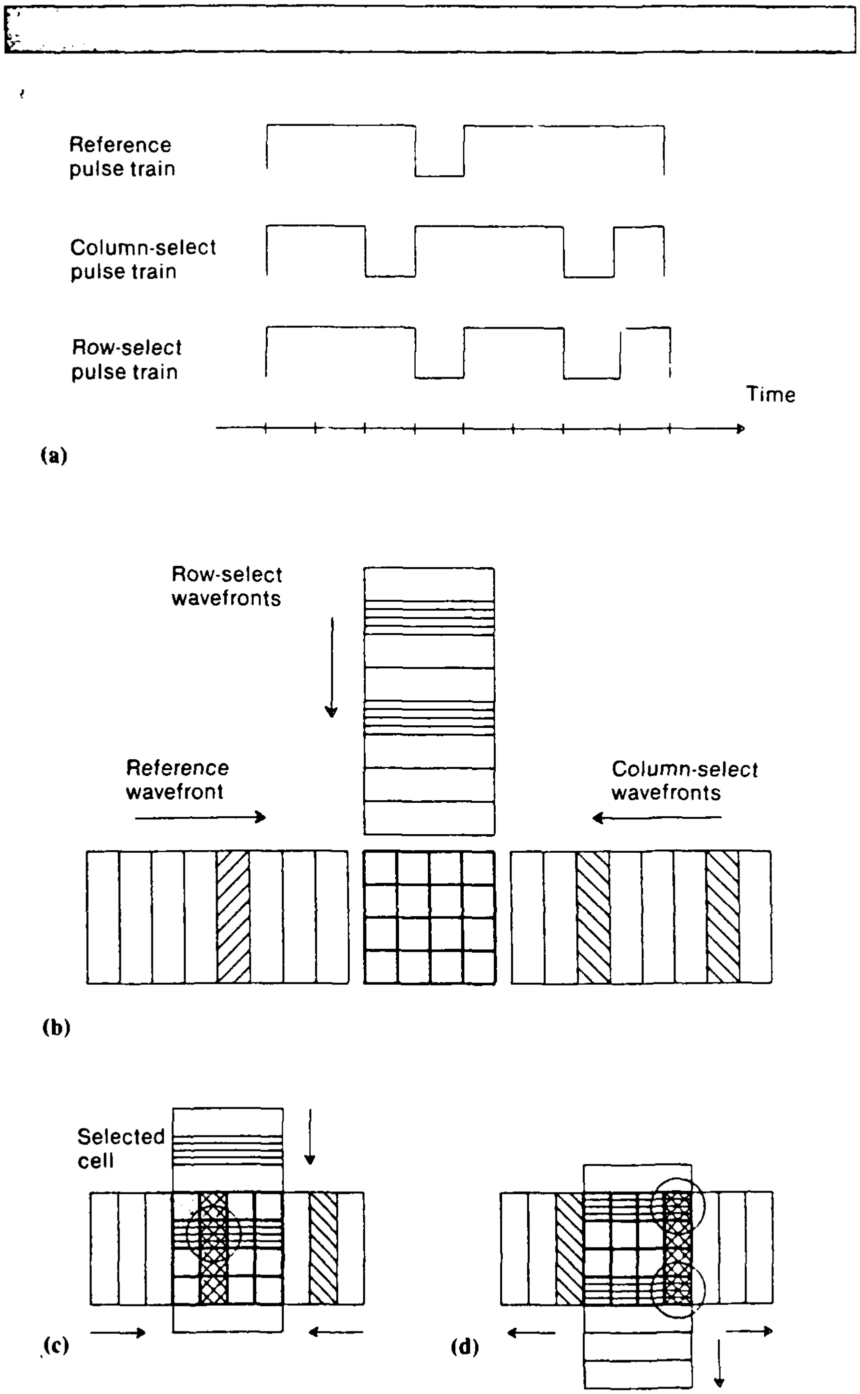


Figure 6. Programming the intersection of wavefronts, showing the select and reference pulse trains (a), the wavefronts at time 0 (b), and the wavefronts at times 5τ and 7τ (c and d).

Resolution of conflicts in memory access. Two policies can be applied to limit the number of memory references during a given cycle to $\sqrt{n}$. The first policy is to allow only $\sqrt{n}$ addresses to reach the memory module during the cycle, and the second is to allow as many as n addresses to reach the memory but return only the con-

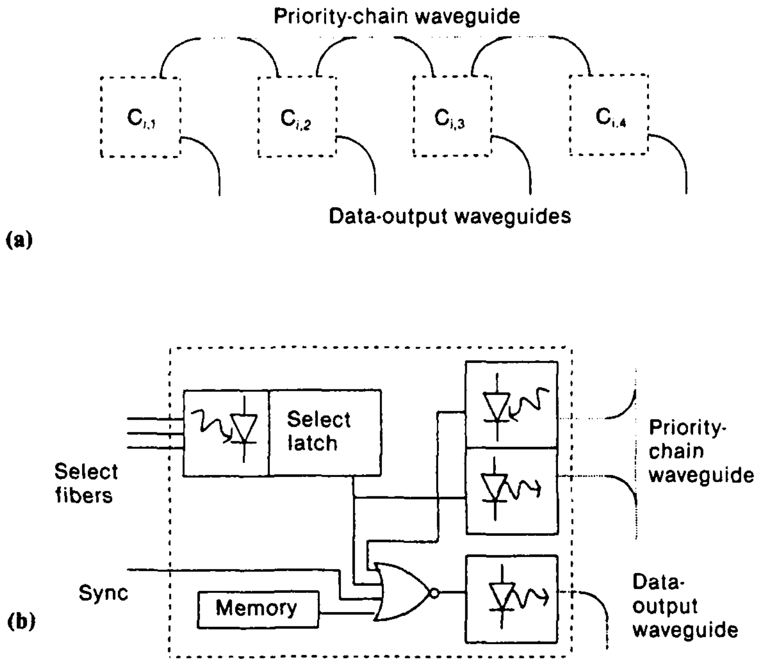


Figure 7. Conflict resolution strategy, with a prioritized memory row (a) and a typical memory cell (b).

tents of $\sqrt{n}$ of these addresses. The first policy requires active optical switching devices to resolve conflicts in the incoming select pulse trains. To avoid the need for such devices, we choose to implement the second policy, which allows full addressing and prioritizes referenced locations for conflict resolution.

The same data collection circuitry used in the linear array is used to collect data for each column of the two-dimensional array. One waveguide is dedicated to the collection of the contents of the addressed cells in each column. The signals in the $\sqrt{n}$ waveguides are merged into a single waveguide (denoted "dataout pulses" in Figure 5), which returns the data to the processors. The lengths of the waveguides are adjusted such that the optical paths between any two cells i,j and i,j' in the same row, i, and the merging point are equal.

With this collection mechanism, the content of any referenced cell i,j in column j will appear in the $(\sqrt{n} - i + 1)$th time slot on the waveguide of column j. However, when the $\sqrt{n}$ pulse trains corresponding to the $\sqrt{n}$ columns are merged in the data output waveguide, the data produced by any two cells i,j and i,j' in the same row and different columns will collide. Since we have elected not to provide a mechanism for preventing conflicting addresses in the input pulse train, conflict resolution must be built into the memory array. That is, requests for memory references in the same row can be allowed, but only one request should be satisfied. Two problems arise for such a system. First, a mechanism to allow only one cell per row to output its data must be devised. Second, a method is needed for announcing which of the conflicting requests has been satisfied.

We start by resolving the second issue. The discussion to this point has assumed a single-bit memory. In fact, memory locations will contain an entire word of W bits, stored electronically and returned in parallel on W optical data out lines. If each memory location is tagged with its column number, then each processor can read the column address along with the data and use the data only if the address coincides with its request. If not, the processor must reissue the memory request. For a memory of size n, $\log(\sqrt{n})$ tag bits are needed, which increases the number of bits stored in each memory location from W to $W + \log(\sqrt{n})$.

The first of the conflict resolution issues is the more difficult. To prevent bus contention between conflicting requests, we have chosen a priority system based on an optical priority chain, like the one shown in Figure 7. Figure 7a shows a single row of a two-dimensional array. The optical distance from each cell in this row to the data output waveguide is equal, hence any parallel accesses within this row will conflict. To avoid this conflict, only one of the optical sources along this row can be allowed to generate data. The horizontal waveguide connecting all cells in the row forms an optical priority chain to resolve these conflicts.

Figure 7b shows a diagram of a typical memory cell. The optical OR output from the pulse-sensing photodetector sets the select latch. This output gates the contents of the memory cell through the three-input, electronic output control NOR gate. The third input to this NOR gate is the priority control. This signal is the output of a photodetector, which senses select signals for higher priority cells indicated optically on the priority chain waveguide. The local select signal is also used to turn on a laser diode, which injects light into the priority chain waveguide to indicate its selection to lower priority cells. Finally, the synchronization signal, sync2, ensures that all data out pulsers are activated simultaneously.

From the above description, it is clear that every memory read cycle is divided into three stages. In the first stage the select pulse trains are generated and propagated through the memory array. The minimum time required to complete this stage is equal to $2\sqrt{n}\tau$.

In the second stage the read operation is propagated through the memory cell electronics. Note that in Figure 5 the priority chain waveguides are parallel to the reference pulse waveguides. In this arrangement the wavefront that encodes current priority propagates through the priority chain waveguides during the first stage of the pipeline. The priority is delayed relative to the reference pulse by time

$$t_{pd} = t_s + t_l$$

where t_s is the switching delay of the detector/latch circuit and t_l is the turn-on time for the priority-out laser diode. At each cell in the rightmost column of Figure 5, which consists of the last cells to see the reference pulse and thus the lowest in the priority chain, the priority input signal arrives at the output control gate at time $t_{pd} + t_d$ relative to the arrival of the reference pulse (t_d is the response time of the priority-in detector). Since the select signal arrives at the output control gate in time t_s, the critical timing path for the second stage of the pipeline is

$$t_{pd} + t_d + t_g$$

where t_g is the output control gate switching time. By noting that t_l and t_d must be less than or equal to the pulse width τ, we can place the lower bound on second-stage pipeline delay at

$$t_s + t_g + \tau$$

Finally, in the third stage of the pipeline, data is returned from the memory to the requesting processors in a pulse train of $\sqrt{n}$ bits. Thus, the minimum time required for the third stage is $\sqrt{n}\tau$. Assuming that the memory size and the ratio of electronic to optical bandwidths is sufficient to satisfy

$$t_s + t_g < (2\sqrt{n} - 1)\tau$$

the longest of the stages is the first. With this three-stage pipeline the total memory cycle length is $6\sqrt{n}\tau$. Since we are accessing the memory in a pipelined fashion, and each stage can process $\sqrt{n}$ references, the effective memory bandwidth limit is $1/(2\tau)$ words per second.

Finally, we should mention that it is possible to support $2\sqrt{n}$ memory references per cycle, rather than merely $\sqrt{n}$, by rearranging the data collection waveguides of Figure 5. If the data collection waveguides are run diagonally, $2\sqrt{n}$ waveguides can be accommodated at the price of a more complex and unevenly distributed conflict resolution scheme.

Organization of memory modules. In the above discussion, we concentrated on select and read mechanisms, assuming a one-bit memory. In an $n \times W$-bit memory module we would reproduce W copies of the memory cell, the output control NOR gate, and the data output pulser of Figure 7b. For reading, only one decoder-select plane and one priority chain waveguide are necessary for each of the $W + \log(\sqrt{n})$-bit words in an n-word array. The only optical system that must be scaled with the number of bits in the word is the read output pulsers. In the same manner as an electronic memory, we must provide a separate return path for each bit.

Memory write control

For a conventional memory, support for write operations would require an additional control signal and a secondary data path for incoming data. Merely providing these additional signals in a parallel memory will not be adequate, since the issue of resolving mixed parallel reads and writes must also be resolved. The three possibilities for resolving this issue trade off write access time for optical circuit complexity:

(1) *Exclusive write:* In this solution we eliminate the possibility of mixed read/write operations and conflicting write operations. By implementing an external arbitration mechanism, we allow only exclusive write access to the memory. Once a single processor is selected, it can perform writes to the memory using conventional electronics. Although the system requires no additional optics, it results in nonparallel writes. If the ratio of writes to reads for the shared memory structure is relatively low, then exclusive write access represents a viable low-complexity alternative.

(2) *Full parallel read/write:* For full parallel optical writes it is not necessary to provide a separate optical write data bus. By combining control and data information in lieu of a data bus, we can provide fully parallel nonconflicting writes to any of the n locations in the store. In this technique each memory bit sees two bits of select information in each cycle—one bit from the read select optical circuit already described and a second from a *per-bit* local copy of that selection circuit used for write control. These two bits encode four states: read, write a zero, write a one, and do nothing (no select). Thus, by reproducing a second address selection structure at each bit in the word and by judiciously selecting code assignments for the four states, we limit the cost of this system to the addition of one optical selection plane for each bit.

This technique allows fully mixed reads and parallel writes. Any processor can write to any word with no conflict restrictions on the rows or columns of write addresses. However, there is no conflict resolution mechanism to prevent two processors from writing to the same address. As with a multiported electronic memory, such operations would generate unpredictable results. We assume that these mutual exclusion issues would be addressed by appropriate software.

(3) *Bit-serial parallel write:* At the cost of lengthening the overall write time, we can reduce the overhead for the full parallel read/write system to one additional optical selection plane by using a bit-serial approach. In this system a designated cycle initiates a W-bit serial write. The select signals generated by the read selection circuit during this cycle are latched separately and held for the duration of the serial write. Each subsequent cycle uses the write selection circuit to serially process each bit in the word. Meanwhile, parallel reads are still possible, concurrent with the serial write. A global counter/decoder circuit, to define the "current bit" as one of the W data bit planes of the memory, is necessary. It is the only additional overhead for this system.

Extensions and future research

We have concentrated in this presentation on the details of a two-dimensional memory array because of its suitability to integrated optical implementations. Nothing in the design prevents the linear wavefronts in two-dimensional arrays from being generalized to planar waves in three-dimensional arrays. In general, for m-dimensional memory arrays the access time is reduced to the mth root of the memory size. Hence, for a fixed bandwidth in the electronic system, the bandwidth requirements for the optical system are substantially reduced. This reduction is gained at the price of a corresponding reduction in the number of locations that can be referenced in parallel during a single electronic cycle. Like the access time, this number is also reduced to the mth root of the memory size.

Extensions of this technique can be applied to other two-dimensional switching structures. For example, the control information for a crossbar switch can be encoded into pulse trains and used to address specific switches in the crossbar. Pulse delay encoded control information can be prepended onto incoming packets to combine routing information and data without the need for optical-to-optical switching devices.

The question is: Can such a memory be built, and will its size and performance make it suitable for integration in computer systems of the next decade and beyond? The following issues must be examined:

(1) *Scalability:* The scale of the physical device is directly related to the optical pulse width. To minimize the physical spacing of detectors, very short pulses are required. For instance, reducing detector spacing to a scale that will allow monolithic integration will require picosecond pulse widths. Specifically, one-picosecond pulses will allow a detector spacing of 200 to 300 micrometers, depending on the refractive index of the optical medium. Current commercial technology for pulsing laser diodes in discrete devices provides for pulses on the order of 100 picoseconds. Recent research has produced optical pulses as short as eight femtoseconds.[8,9] In such rescarch the common technique for pulse duration measurement is to split the pulse into two optical paths and detect coincidence when the paths are recombined at varying optical path lengths. On the basis of this trend we expect that the necessary pulse widths for an integrated optical implementation will be available in the near future. Meanwhile, we are currently using commercial discrete devices and optical fibers to examine scalability issues.

(2) *Detection limits:* A second limit on usable pulse duration is the detector technology for coincident light pulses. A two-dimensional memory requires that as many as three dark pulses are to be detected as they overlap. Even assuming the existence of photodetectors of sufficient bandwidth for single pulse detection, what degree of overlap is required to generate the optical OR of three pulses? Extended to multidimensional structures, fan-in limitations of this type become even more critical.

(3) *Fabrication versus physical limits:* As the system scales down in size and up in speed, what limits will be reached first—the fabrication limits of the technology or the physical limitations of the optical systems?

(4) *Clocking issues:* While there is little doubt that sufficiently narrow pulses can be generated by the electro-optics, the precision to which multiple pulses can be synchronized is an important question. Two coincident pulses must be timed to arrive with a precision of ±10 percent of their pulse width to allow for at least 80 percent overlap. This means that the electronic components must gate the optical signals with a constant delay that is precise to the optical time base. Clock distribution issues have been studied extensively in both electronic and optical domains. We believe that the required precision can be achieved by electronic circuitry. As an alternative, optical clock distribution techniques such as those proposed by Clymer and Goodman[10] can be applied.

(5) *Select latch response:* The electronic latches at the detector sites must respond to the selection pulses from optical detectors. These pulses will be no longer than the duration of the coincident pulses. This is a limitation on the speed of the optical system.

(6) *Waveguide decoupling technology:* In a two-dimensional design it is necessary to split incoming optical signals into parallel row and column waveguides. Detectors at each intersection must couple out sufficient optical power for detection, without significantly degrading the optical signal. Such highly asymmetric, single-mode directional couplers have been developed for optical fiber and are commercially available.[11] Several other techniques for low-power output coupling have been examined by Jackson et al.[12] Further work is needed to apply these techniques in an integrated optic environment.

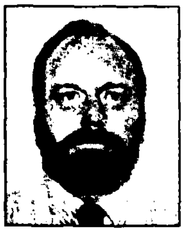
In summary, we have presented a system that distributes the address-decoding function to the requesting units on an optical bus. In this system, addresses become optical pulse trains, and by arranging the optical paths, we provide a selection mechanism based on the coincidence of these pulses. In the coming year we plan to begin construction of a 64 × 16-bit register file based on this research, using discrete optical devices and fiber waveguides. This register file will be used as shared memory in a prototype eight-node multiprocessor.□

Acknowledgments

This research has been supported, in part, under Office of Naval Research contract N00014-85-K-0339.

References

1. T.C. Chen, "Overlap and Pipeline Processing," in *Introduction to Computer Architecture*, Harold Stone, ed., Science Research Associates, Palo Alto, Calif., 1975.
2. F. Briggs and K. Hwang, *Computer Architecture and Parallel Processing*, McGraw-Hill, New York, 1984.
3. J. Neff, "Major Initiatives for Optical Computing," *Optical Engineering*, Jan. 1987, pp. 2-9.
4. *IEEE Spectrum*, special issue on optical computers, Aug. 1986.
5. D.L. Lee, *Electromagnetic Principles of Integrated Optics*, John Wiley and Sons, New York, 1986.
6. H.T. Kung, "Why Systolic Architectures?" *Computer*, Jan. 1982, pp. 37-46.
7. S.Y. Kung et al., "Wavefront Array Processor: Language, Architecture, and



Donald M. Chiarulli is an assistant professor of computer science at the University of Pittsburgh. As his dissertation project at Louisiana State, he was responsible for the design and construction of the Factoring Machine, a reconfigurable VLIW machine for factoring large numbers. Prior to that, he served four years as president of Datanet Services, Inc. His current research interests include hybrid optical/electrical computer architecture, optical interconnects, VLSI design, and parallel computation.

Chiarulli received a BS degree in physics from Louisiana State University in 1976, an MS degree in computer science from Virginia Polytechnic Institute in 1979, and a PhD in computer science from Louisiana State University in 1986.

Rami G. Melhem has been with the faculty of the University of Pittsburgh's Computer Science Department since 1986. His research interests include the systematic design and verification of large computational networks, specialized architectures for scientific problems, and the application of novel techniques and optical technology to parallel computing systems.

Melhem received a BE in electrical engineering from Cairo University, Egypt, in 1976, an MS in mathematics/computer science from the University of Pittsburgh in 1981, and a PhD in computer science from the University of Pittsburgh in 1983.

Steven P. Levitan is the Wellington C. Carl Assistant Professor of Electrical Engineering at the University of Pittsburgh. Previously, he worked for Xylogic Systems, designing hardware for computerized text-processing systems, and for Digital Equipment Corporation on the silicon synthesis project. His research interests include parallel computer architecture, parallel algorithm design, VLSI design, and computer architectures for image understanding.

Levitan received the BS degree in 1972 from Case Western Reserve University; he received the MS in 1979 and the PhD in 1984, both in computer science, from the University of Massachusetts, Amherst.

Readers may write to the authors at the University of Pittsburgh, Dept. of Computer Science, 322 Alumni Hall, Pittsburgh, PA 15260.

Applications," *IEEE Trans. Computers*, Nov. 1982, pp. 1054-1066.

8. C.V. Shank, "The Role of Ultrafast Optical Pulses in High Speed Electronics," in *Picosecond Electronics and Optoelectronics*, G. Morou, D. Bloom, and C. Lee, eds., Springer-Verlag, New York, 1985.
9. J.G. Fujimoto, A.M. Weiner, and E.P. Ippen, "Generation and Measurement of Optical Pulses as Short as 16 fs.," *Applied Physics Letters*, May 1984, pp. 832-834.
10. B.D. Clymer and J.W. Goodman, "Optical Clock Distribution to Silicon Chips," *Proc. SPIE*, Vol. 625, 1986, pp. 134-138.
11. V.J. Tekippe and W.R. Wilson, "Single-Mode Directional Couplers," *Laser Focus*, May 1985.
12. K.P. Jackson et al., "Optical Fiber Delay-Line Signal Processing," *IEEE Trans. Microwave Theory and Technologies*, Mar. 1985.

Technical report 88-2

Optical Bus Control for Distributed Multiprocessors

Donald M. Chiarulli
Steven P. Levitan
Rami G. Melhem

Department of Computer Science
The University of Pittsburgh
Pittsburgh, PA 15260

Optical Bus Control for Distributed Multiprocessors

Donald M. Chiarulli

Steven P. Levitan

Rami G. Melhem

University of Pittsburgh
Pittsburgh, PA 15260

ABSTRACT

Optical interconnections offer a potential for gigahertz transfer rates in an environment free from capacitive bus loading, crosstalk, and electro-magnetic interference. Device technology in electro-optics has matured to a point where small, low power, and low cost devices exist which are suitable for use in bus level implementations. Therefore, the realization of physically distributed, bus interconnected multiprocessors is now possible.

In this paper we propose a bus arbitration mechanism suitable for a large optical bus structure heavily populated with asynchronous bus masters. It is a two level arbitration system with incoming requests batched on a demand basis and serviced in a linear priority order within each batch. Low priority requests cannot be starved in high contention environments. For an optical bus of fixed length, propagation delay is bounded by the end to end transit time and is independent of the number of devices attached. In addition, as contention for the bus increases and batch sizes become larger, the time overhead paid for bus control decreases. Thus, a performance improvement is achieved dynamically under high contention conditions.

This research has been supported in part by a grant from the Air Force Office of Scientific Research under contract number AFOSR 86-0198.

Optical Bus Control for Distributed Multiprocessors

Donald M. Chiarulli

Steven P. Levitan

Rami G. Melhem

University of Pittsburgh
Pittsburgh, PA 15260

1. Introduction

With the introduction of new materials, such as GaAs, the switching time of electronic devices will soon outstrip the bandwidth of the electronic communications channels which interconnect them. It is inevitable that optical communications will be adopted as an alternative. We recognize three potential targets of the application of this technology: first, computer to computer interconnects over optical local area networks (these systems are already a commercial reality [10,23]); second, optical busses used to connect components within a single system; third, inter-chip interconnections via optical channels. We have chosen to examine the issues in the application of optical technology to the second category, system bus interconnections. It is our belief that many of our results will carry over to inter-chip interconnection technology as research in hybrid electro-optical monolithic devices advances.

Our work builds on both traditional electronic bus design techniques and more recent work on optical high speed local networks (HSLNs). However, we must distinguish between the applications and requirements of HSLNs and those of system busses. Busses, as opposed to networks, have a number of particular requirements beyond high bandwidth: Bus level transactions occur with a volume of distinct messages per source which is higher than is typically experienced in network environments. Moreover, messages are small (word size) in bus environments. In addition, the size, cost, and circuit complexity of controllers at the bus level is more tightly constrained since these controllers must be reproduced at the bus interface of each device. Latency, the actual time from a transmission request to the arrival of data at a receiver, is critical at the bus level. Specifically, the two components of latency, control overhead and transmission time, represent areas of substantial differences between optical and electronic busses.

To date, optical networks have been proposed in a variety of configurations including star, ring, and backbone systems. Control structures for these systems fall into three categories. Fibernet [15, 18], for example, is a centralized optical network based on a passive "star" coupler. This system, and Fibernet II [21] are representative of the first category which use a carrier sense multiple access (CSMA/C⌐\ control protocol [19]. Fibernet II was designed for more than 100 computers with a radius of greater than 2.5 km. However, since compatibility with electronic ethernet systems was a primary concern, these systems resort to electronic collision detection and their performance is bounded by the speed and complexity of centralized electronic control. Demand assignment multiple access (DAMA) networks such as EXPRESSNET and FASTNET [17, 24, 25] form the second category. Also known as access and defer systems, these networks have controllers which monitor their inputs and outputs on unidirectional fiber-optic waveguides to simultaneously transmit and detect network activity. If, during a transmission, activity is sensed from the upstream direction of the waveguide, the controller aborts the transmitted message in deference to the upstream controller. More recently, optical technology has been applied to the third category, token rings. The 80 Mbps fiber ring network from Proteon [3], the 100 Mbps Fiber Distributed Data Interface (FDDI) ring [13, 20], and the Lookahead Network [7] are examples of this control structure. All of these systems are designed for use in HSLN applications, where relatively long packets of information are sent for each transaction. Message latency is amortized over the length of the packets.

Rather than networks, we are interested in building closely coupled systems. By closely coupled we mean that the resources of the system are available via a single bus level operation without any I/O transfers, in a manner transparent to both systems and application level software. Of the three categories of control structures listed above, each has a disadvantage which makes it inappropriate for bus applications. Optimistic systems such as the CSMA/CD systems and optical access and defer protocols perform well under low request rates but degrade severely as contention increases. For CSMA/CD this results in unacceptable overhead for collision detection and retransmission for the high message volume environment of a system bus interconnection. Token passing schemes are better suited to high volume environments. However, token based access always forms a logical ring. Control must pass in a predetermined order to all controllers regardless of the actual number of pending requests. Optical DAMA schemes have better overhead performance than CSMA/CD but they require special characters to be transmitted for synchronization. The recognition and handling of these control characters results in controller complexity which is unsuitable for system bus level implementation. In addition to controller complexity, subtle differences between controllers on the same bus must be supported in order to designate which controller will generate the required synchronization characters.

We introduce in this paper a new system designed specifically to meet the requirements of system busses. We have chosen a topology similar to the DAMA schemes described above. This allows us to retain the low overhead performance of these systems. However, we have eliminated the need for synchronizing characters, have made all controllers identical, and have reduced the complexity of the controllers to a few gates.

Our belief that such optical designs can be implemented comes from the current direction of electro-optical device research. As these devices follow the path of electronics to smaller, lower cost, and lower power requirements, they can be expected to migrate to the internal circuitry of computer systems [5,6,8,9,11,12]. We therefore present an electro-optical high speed distributed bus interconnection technique. Specifically, we will present a control structure suitable for optical implementation which is asynchronous, decentralized, and capable of supporting parallel bus transfers.

1.1. Unique Aspects of Optical Busses

The first temptation in the design of an optical bus is to simply make a technological substitution of optics for electronics. Such a solution makes little contribution beyond increased bandwidth. In addition, the performance of such an implementation will inevitably be bound by the speed of the electronic components attached to the bus. More importantly, the use of optics enables the construction of the sam: bus, physically distributed at distances on the order of a kilometer. However, for any technology, optics or electronics, the assumption of exclusive access to the bus resource limits throughput to a function of the end-to-end transmission time for information on the bus. End-to-end transmission times for optical signals are not inherently shorter than for electronic signals. Therefore, unless the additional bandwidth of optics can be used to support parallelism in bus transfers, we cannot expect the throughput of such a distributed system to be substantially higher than an electronic design. Thus it is imperative that new designs be specified which address the issue of exclusive bus access.

A key characteristic of optical interconnections in the solution of this problem is the ability to pipeline the transmission of signals through a channel. Electronic busses are restricted to a single transmission per unit time which propagates in both directions from the source. This characteristic requires that electronic systems introduce directional amplifiers between adjacent bus sources to achieve pipelining. These amplifiers produce unpredictable delays which make any large scale distributed implementation impractical. Optical channels are inherently directional and have predictable delay per unit length. A pipeline of signals can therefore be created by the synchronized directional coupling of each signal at specified optical path lengths along the channel. Thus, it is possible to support temporal parallelism in the form of multiple transactions on the bus simultaneously.

1.2. New constraints

Computer architectures based on temporal parallelism along the interconnection channels will fundamentally change the design of multiprocessor systems. For example, by abandoning exclusive bus access, we must now base our bus arbitration mechanisms on the allocation of temporal/spatial slots rather than the entire bus. In addition, the freedom from inductive-capacitive effects loosens the constraints on the total length of an optical bus and thus gives us the ability to build distributed systems which are closely coupled.

The new constraints on the size and scale of these systems are latency and available optical power. Latency limits the total length of the bus while optical power distribution limits the number of devices attached to the bus. Using temporal parallelism, latency can be amortized over the number of parallel messages active on the bus. Optical power, on the other hand, represents a more difficult problem. With current commercially available optical couplers, we see a practical limit of about 100 processors on a single bus. We expect that the technology for low split ratio optical couplers will continue to evolve. However, other solutions to this problem exist. These include the segmentation of the bus into several local busses, the insertion of high speed repeaters, or the adaptation of special fiber tapping configurations such as the stretched or bypass configurations proposed by Nessehi, Tobagi, and Marhic[17] to limit the number of fiber taps

Finally, the effect of such distribution and scaling implies a non-centralized and asynchronous control structure. This is because the physical distribution of system components will result in clock skew, even for an optically transmitted clock. Thus asynchronous control is mandatory. Moreover, latency arguments make the performance of any centralized control system unacceptable.

In this paper we implicitly assume an optical-bus interconnected multiprocessor with optical data connections of sufficient bandwidth to justify closely coupled interactions. We begin in Section 2 by presenting a simple priority chain control structure. This structure is presented in both its electronic and optical versions to familiarize the reader with the transition to optics. In Section 3, we present our new control structure. Also in Section 3, we present an example controller implementation and a timing analysis for the design. In Section 4, we present experimental data derived from a discrete event simulation of a 64 processor model. Through this model we demonstrate the performance of the control mechanism relative to bus overhead.

2. A Synchronous Bus Control Structure

Synchronous electronic control structures, such as the one shown in Figure 1a, are well understood and commonly implemented. We present a discussion of synchronous structures in this section to provide a starting point at which a familiar electronic structure can be compared with a functionally equivalent electro-optical implementation. One such implementation is shown in Figure 1b. With an assumption of synchronous operation we require that any bus grant operations occur in concert with the designated edge of a global clock signal. We further assume that bus request signals from each device are stable and valid before this clock edge occurs. Once granted, a bus cycle proceeds for a fixed interval, defined by the clock period, and cannot be preempted by an incoming request. In both figures, bus requests, which are signaled electronically from each device to the corresponding bus controller, are labeled req_i to req_{i+n}. This is defined to be ascending priority order. In both the electronic and optical versions, the priority signal propagates from left to right.

For the electro-optical system, the key component of the priority chain is a 2×2 non-reciprocal (or asymmetric) optical coupler. In such a coupler, shown in Figure 2, an optical signal presented at either input port is reproduced at both output ports with the output optical power divided between the outputs A' and B' such that $P_A' \gg P_B'$. Such a coupler is said to be non-reciprocal if the ratio of power coupled out on each output does not depend on the ratio of power coupled in from either input. We can express the optical power at each output for a non-reciprocal coupler as

$$P_A' = (1-H)(P_A + P_B - \varepsilon)$$

$$P_B' = H(P_A + P_B - \varepsilon)$$

where H is the coupling ratio and ε is a combined coupler loss term including excess loss, and directionality. Several techniques for fabricating both reciprocal[4, 14, 22] and non-reciprocal couplers[26], have been demonstrated experimentally and several commercially available non-reciprocal couplers[1, 2] have been recently introduced.

These couplers are a critical technology for the construction of large scale optical bus systems. The ability to distribute optical power to a large number of detector sites along a bus will ultimately determine the scale to which such systems can be built. In Table 1 we have calculated the percentage of input optical power presented to a detector at the 64th, 128th and 256th tap on an optical bus ignoring excess loss and assuming a uniform coupling ratio for all taps. For large scale implementations, the assumption of uniform coupling ratio for all taps is unrealistic. Since optical busses are undirectional, it is likely that downstream taps would have increasing coupling ratios thus allowing a larger

Ratio	tap	power	tap	power	tap	power
1%	64	.525	128	.276	256	.0763
2%	64	.274	128	.075	256	.0058
5%	64	.037	128	.0014		
10%	64	.0012				

Table 1 - Percent of Input Power: Various Ratios and Taps

percentage of the optical power remaining in the fiber to reach the detector. All of this does not preclude the use of other techniques to increase optical power distribution. Provided that timing relationships of signals will not be affected (i.e. all signals in a single fiber), repeaters can be introduced. Further, stretched structures and bypass structures[17] can be implemented to reduce the number of taps required.

Returning now to the example bus configuration of figure 1b, couplers, of the type shown in figure 2, between each controller function as an optical *OR*. Thus, at any coupler between controller i and $i+1$ (ascending priority), the optically encoded priority bus signal enters on port A and is combined with the optical priority-output signal from stage $i+1$ which is connected to port B. The combined signal at output A' continues on to the priority input (port A) of the next stage coupler. Output B' is the priority-in signal for controller i.

When an active req_i signal is asserted electronically by one or more devices to the corresponding bus controllers, the priority-out laser diode at each controller is activated to inject light into the priority waveguide. Simultaneously the active request is ANDed with the inverse of the priority waveguide state, which is sensed (and electronically inverted) by the priority-in photodetectㄱr. This combination of a local request and no active higher priority requests forms bus grant which is latched on the next clock edge. Our assumption that all request signals are stable and valid before the clock transition assures that only one such bus grant will be issued. This is true provided that sufficient time is allowed in the clock cycle for propagation of the optical signals through the entire length of the priority waveguide and for one optical to electronic conversion time.

An analysis of the hardware and time delay complexity of the optical structure and its electronic equivalent yields the following. The hardware complexity of both grow linearly with n, the number of bus devices. However, in the optical structure, a fixed overhead is paid at each controller for conversion devices between the optical and electronic domains. This conversion buys a substantial improvement in the time delay complexity for the optical implementation. Specifically, the electronic structure pays a time delay of one gate delay/controller attached to the bus. The propagation delay for optical signals is proportional only to the end to end optical path length. Optically encoded priority is broadcast by each controller to all other controllers of lower priority. Thus the time delay for propagating priority between any two controllers is equal to the transit time of light through the waveguide for a distance corresponding to the physical separation of the devices plus one optical to electronic conversion. This physical separation will only be related to the number of intervening devices in the case of non-distributed systems.

3. Distributed Asynchronous Control

We have already stated that the primary constraints on the construction of a large scale optical bus are latency and power distribution. Turning now to the latency problem we recognize two components of bus latency, control overhead and signal transmission time. We will assume an environment in which the devices attached to the bus are heterogeneous. That is to say that the requirements for bus message length are non-uniform throughout the requesting devices. Further the physical distribution of system devices will introduce unacceptable clock skew. Based on these specifications it is obvious that the synchronous structure of the previous section is inadequate. An alternative which supports a decentrallized and asynchronous bus control mechanism is the only solution which will provide adequate performance.

More formally, the requirements for this control structure are: 1) A mechanism must be provided whereby an active controller may place a message on the bus, once granted, without preemption by or contention with another bus message. 2) Multiple requests held pending by any subset of the bus controllers be resolved appropriately without starvation of any controller. 3) Any pair of requests which arrive within the signal propagation time between the two controllers at which the request are made must be considered simultaneous. The resolution of this conflict must be free of any possible deadlock or contention.

Our system model consists of a collection of devices each connected to a corresponding bus controller. Communication between device and controller consists of two electronic signals shown in figure 3. *Bus_request* is asserted by the device when a bus transfer is required and *bus_grant* is asserted by the controller when the bus is available. Both *bus_request* and *bus_grant* are held active for the duration of the bus transfer.

In addition to the electronic *bus_request* and *bus_grant* signals in figure 3, three optical signals, one output and two input, support an optical control bus. The output signal, *request*, indicates a pending request at the corresponding controller. The *priority* input signal reflects the state of the *request* outputs of all higher priority controllers. *Request* and *priority* are coupled to a common waveguide which we will refer to as the *priority* waveguide. This is identical to the structure used previously in the synchronous bus. The difference in this new controller is in the interconnection of the third optical signal, *ack*. All controllers tap the *ack* input signal from a common waveguide which we refer to as the *ack* waveguide. The *ack* input eliminates the need for centralized clocking which characterized the synchronous example. It does so by defining a time period during which optical outputs can be generated by a controller. Logically, a high level (light) on *ack* indicates that the bus is busy and that a set of one or more transfers are in progress. During this time no new *request* signals can be asserted by any controller regardless of the state of *bus_request* from the corresponding device. Conversely a low level (no light) on *ack* indicates that the bus is either idle or that control arbitration is in progress. During this time, all controllers are free to assert *request*. However, no controller may issue *bus_grant* to its device until *ack* is once again high and priority has been resolved via the a low input on *priority*. Thus, the high-going transition of *ack* assures the stability of the *priority* inputs by defining a time frame in which the *request* signals of higher priority controllers can be asserted.

To support a distributed, self clocking system, we must configure *ack* as a feedback signal. The particular feedback mechanism depends upon the topology chosen. Several such topologies are shown in Figure 4. To operate successfully the *ack* feedback mechanism must insert an appropriate delay between the output of *request* by a controller and the issuance of *bus_grant* based on the state of the optical *priority* input. The direction in which the *ack* signal propagates relative to *priority* in each of these feedback structures classifies the structures as fixed delay timed, as is the case for structures 4(a) and 4(b), or self timed, as is the case for structure 4(c) and 4(d). In the fixed delay timed structures, an electronic delay element must be included in the controller between *bus_request* and *bus_grant* with a delay greater than the worst case propagation delay in the system. This delay corresponds to the time for *request* to propagate to the most distant controller via the priority waveguide and for a potential *ack* from that controller to propagate back. If we define the transit time for an optical signal to travel between the two most distant controllers in the system as τ, this represents a delay equal to 2τ for the optical control system to generate a bus grant.

For the remainder of this paper we will concentrate on self timed structures, specifically the spiral interconnect structure shown in Figure 4(c) and the equivalent folded structure of 4(d). The feedback signal *ack* is generated in the spiral by connecting the *priority* waveguide at the lowest priority controller to the *ack* waveguide at the highest priority controller. Self timing, which results from the propagation of *ack* in the same direction as *priority*, eliminates the need for explicit delay elements in each controller. Specifically, a single round trip propagation of a *request* signal is indicated at each controller by a transition on the *ack* input. This edge is a reference against which the validity of *priority* can be guaranteed.

An additional advantage of the spiral arrangement is a reduction of the propagation delay to a single end to end propagation delay τ as compared to the 2τ propagation delay which characterized the delay timed structures. For large and distributed systems however, even a single end to end transit time can greatly reduce effective bus bandwidth. Moreover, we will show in the example below that the arbitration overhead for multiple requests is further reduced to the optical signal transit time between the two controllers, provided that the two controllers assert *request* during the same interval of *ack* low.

Note that for the spiral structure (as well as the other delay timed structures) of figure 4(c), the physical separation of the devices attached to the bus need not be equal. Thus if a spiral topology for the devices is not convenient, the more general folded waveguide structure of figure 4(d) can be implemented at a cost of an additional round trip delay.

In the case of multiple requests, a low level on *ack* causes the optical assertion of *request* into the *priority* waveguide by each controller which holds a pending *bus_request* from its device. A transition on *ack* follows as the *priority* waveguide feeds back into the *ack* waveguide from the highest to lowest priority controller. As this transition arrives at each of the active controllers, only the highest priority controller sees the appropriate grant condition of *priority* low and *ack* high. All others see *priority* high and hold their respective *request* outputs high. Only a single *bus_grant* is issued by the highest priority controller. At the end of the transfer at the highest priority controller, the removal of *request* will lower the *priority* input of the next higher priority controller. *Priority* at all lower priority controllers and *ack* at all controllers will remain high, held by the *request* output of the controller succeeding in priority order to the previously active controller.

Operating the controllers in this fashion has two very desirable side effects. First, the control delay for arbitration of multiple requests is now proportional to the optical length between the two highest priority asserting controllers. This delay approaches τ only in the worst case, that is for exactly two requests generated by the highest and lowest (most distant) controllers. For any other combination of requests, the delay requirement must be less that τ. In addition, for a high contention environment, where the number of pending requests is large, the control overhead will become less, as the requests are grouped more closely on the bus. Secondly, since there is no transition on *ack* until all of the currently pending requests have been satisfied, the control mechanism is no longer a strict priority system. At each transition of *ack*, that is to say each time the bus goes from idle to active, all pending requests at that time form a batch. No new requests arriving after that transition can be asserted on *request* nor can they be granted until all requests in the current batch are serviced. Priority based resolution of requests occurs only within a batch and new requests are equally eligible for inclusion in the next batch.

While it is clear that the implementation of a batch service discipline is effective in reducing control overhead, it is even more important in dealing with the other component of bus latency, message transmission time. Consider an optical bus system consisting of the control waveguides described here and a separate data waveguide on which bus messages are transmitted. In a conventional control mechanism, a controller which obtains a bus grant assumes exclusive access to the entire data bus. Alternatively, an optically transmitted message propagates unidirectionally with a predictable propagation rate in the data waveguide. Therefore, within a batch the low going edge of *request* in the control waveguide corresponds to *the last bit of the* transmitted message in the data waveguide. Assuming that the optical path lengths of the data and control waveguides are equal, this edge will arrive at the next controller which holds a pending request just as the last message bit propagates past that device in the data waveguide. Thus, the new message can be inserted into the data waveguide immediately after the first. In practice this would require adjusting the timing of the low going *request* signal forward relative to the last message bit to account for the electronic delay in each controller. The limit on the actual proximity of the message is therefore based on the tolerances for the timing specifications and not on the actual delay itself. Relative to message transmission latency, this means that while the end-to-end propagation delay of any single message is still limited by the optical path length, the message throughput for the bus is increased by a factor corresponding to the average size of a batch. As contention for the bus increases, so does the average batch size. With increasing batch size, both components of bus latency are effectively reduced; control overhead by the reduction in control signal propagation time, and message latency by multiplexing several messages into a single arbitration cycle.

4. Controller Design

The contróller circuit shown in Fig. 5 is a direct implementation of the control unit of Fig. 3. In both figures, three optical signals form the control bus: *request* is an optical output signal from diode L1 which indicates that the device has a message for the current batch, *priority* is an optical input signal at detector D2 which indicates that higher priority batch transmissions are in progress, and *ack* is an optical input signal at detector D1 which frames the batch formation cycles. Two electronic signals are also included to form the interface between the controller and the corresponding device: *bus_request* is an electronic input signal used to initiate a bus transfer, and *bus_grant* is an electronic output signal used by the controller to signal the device to seize a bus slot. It is assumed that the attached device will hold *bus_request* active for the duration of its bus activity and will signal the end of the bus transfer by lowering the *bus_request* input.

To analyze this circuit we begin by examining the operation of the *request* output signal at L1. This signal must be asserted on an active *bus_request* input in conjunction with *ack* low. Once asserted, *request* should remain active until a high-to-low transition of *bus_request*. The former of these two conditions is satisfied by gate G1 whose output controls the set input to the SR flip-flop which drives L1. Thus, with *ack* low, a low-to-high transition on *bus_request* drives the (S) input low and the (R) input high, setting the flip-flop. The latter condition of removing *request* at the end of the bus cycle is similarly satisfied by the high-to-low transition of *bus_request*.

In order for *bus_grant* to be asserted to the device, the following conditions must be met. First, the corresponding *bus_request* must be latched by the mechanism above. This indicates the placement of the request into the current batch. Second, there must be no higher priority requests in service, indicated by *priority* low. Third, the *ack* input must be high. All three of these conditions are satisfied by the inputs to G2 which drives the *bus_grant* output.

At the end of batch formation, indicated by the high-going transition of *ack*, it is assumed that the *priority* input at each controller in the new batch is stable and valid at the input of G2. However, special care must be taken to ensure that no signals are currently in transit through the *request* waveguide from higher priority controllers in the batch. This is possible if a transition on *bus_request* at a higher priority controller arrived simultaneously with the batch termination transition of *ack* at that controller.

To be more specific, the latching of *bus_request* is under the control of *ack* at each of the higher priority controllers. It can be assumed that, when an *ack* transition arrives at the lower priority controller, the same transition has already been seen by all higher priority controllers. Since the optical path length of the *priority* waveguide is assumed to be equal to the optical path length of the *ack* waveguide between any two controllers, any differences in the timing path for these two signals must be due to differences between the electronic timing paths. These paths are from *ack* to *request*, through G1, the SR flip-flip and L1 in the higher priority controller, and from *ack* to *bus_grant*, through D1 and G2 in the lower priority controller. Based on these paths, a new signal in the *priority* waveguide could trail the corresponding transition in the *ack* waveguide by a delay equal to no more than one gate delay (G1), one latch time, and one laser diode turn-on time (L1). We refer to this delay as τ_{req}. To adjust for this delay and ensure that *priority* is stable on the transition of *ack*, we add a delay element equal to τ_{req} on the *ack* signal in the *bus_grant* timing path at the input to G2. However, it is important to note that this delay is inserted to extend the delay between any pair of controllers. Since the delay is not in the optical path for *priority* and *ack*, it does not extend the end-to-end propagation delay of the bus. The timing analysis given below will further clarify the need for this delay element.

Consider any pair of controllers A and B, such that controller A is the lower priority controller and controller B is the higher priority controller. Figure 6 shows a timing diagram for *bus_request*, *request*, *priority*, *ack*, and *bus_grant* at each of the two controllers. Assume that the bus is initially idle (*ack* low) with no requests pending, and that the device attached to controller A raises the *bus_request* line at time t_0. With the bus idle, the optical *request* signal enters the *priority* waveguide at time $t_0+\tau_{req}$. After time τ_{ack}, corresponding to the round trip transit time through the *priority* and *ack* waveguides and one photodetector turn on time, the *ack* input becomes active at controller A. It is at this time $t_0+\tau_{req}+\tau_{ack}$ that controller A is prepared to generate a *bus_grant*. However, if $\tau_{B,A}$ is the transit time for an optical signal between controllers B and A for both *ack* and *request*, it is possible for a *bus_request* at controller B issued any time up to $t_0+\tau_{req}+\tau_{ack}-\tau_{B,A}$ to be in transit along the *request* waveguide, delayed oy a maximum of τ_{req} relative to *ack*. For this reason, *priority* cannot be considered valid for purposes of *bus_grant* until $t_0+2\tau_{req}+\tau_{ack}$ and a delay element corresponding to a τ_{req} delay must be added to *ack* at the input to G2, the *bus_grant* AND.

Turning now to the control latency introduced by this circuit, note that the delay time $2\tau_{req}+\tau_{ack}$ is the timing delay from bus idle to the first bus grant of a batch. Within a batch, arbitration between bus grants is considerably faster. Referring again to Fig. 6, if the device attached to controller B lowers *bus_request* at time t_1, this signal will appear optically at the *priority* input of controller A at $t_1+\tau_{req}+\tau_{B,A}$. If time τ_{grant} is the gate delay for G2, the total time for arbitration of requests within a batch becomes $\tau_{req}+\tau_{B,A}+\tau_{grant}$. By definition $\tau_{B,A}$ must be less than τ_{ack}. Moreover, in large batches

τ_{BA} will be substantially less. Thus, the bus arbitration system will actually perform better in situations of high contention where large batches can be expected.

5. Simulation Results

We have conducted a discrete event simulation study on an 64 processor model arranged as in figure 4c. For simplicity of the model, the processors are arranged such that the optical path distance between each pair of adjacent processors is equal. While this topology is more restrictive than can be supported in general, it provides a convenient model for performance measurements which is independent of other parameters such as the number of processors.

Two parameters in the model determine the level of bus contention: average next request delay and average transfer length. Average next request delay, τ_{nrd}, is the period that any processor will wait before issuing its next bus request after completion of a bus transfer cycle. Average transfer length, τ_{trans}, is the period a processor will hold the bus (slot size) once a bus grant is issued. For this simulation we have chosen a fixed average value for τ_{trans} and the actual length used for each simulated transfer was randomly generated within a small range bounded by $\tau_{trans}/2$. To simulate various levels of bus contention, τ_{nrd} was varied in each simulation. We began with a relatively low demand environment and incrementally increased demand, by a proportional decrease in τ_{nrd}, until bus saturation. In the final saturated test, new requests arrive at each processor more often than the average transfer length.

Figure 7(a) and 7(b) show clearly the reduction in overhead with increased contention. In figure 7(a) we recognize three possible bus states at any time instant: *idle*, *busy*, and *overhead*. The bus is *idle* when no bus requests are pending and no transfers are in progress. The *busy* state is defined to be the period when a bus slot has been granted to a processor and the requested bus transfer is in progress. *Overhead* occurs between the termination of a busy state and the next grant, if a request is currently pending, or the time from request to grant if the bus is currently idle. We have accounted for and plotted in figure 7(a) the percentage of total time the bus spends in each of the three states vs. increasing bus demand. The uppermost plot, *busy*, increases as expected as larger numbers of bus requests are serviced. The lower-most plot, *idle*, shows the corresponding decrease in bus idle time with increasing demand. The middle plot, *overhead*, initially increases with increasing bus traffic until all idle bus cycles have been exhausted. At this level of demand, where $\tau_{nrd}/n < \tau_{trans}$ (n is the number of processors), one or more new requests will always arrive during each bus transfer. In a fixed overhead system this would be defined as the point of bus saturation. The expected behavior of the *busy* and *overhead* plots would be as shown by the dotted lines of Fig. 7(a). In the protocol we have proposed, it is at this point that batching becomes a dominate effect and control overhead begins to decrease proportionally to further increases in the level of demand. The decreasing overhead trace in this region corresponds to additional bus capacity provided by overhead reduction. It continues to decrease until actual bus saturation, where $\tau_{nrd} < \tau_{trans}$. At this point each new batch contains requests from all n processors.

Figure 7(b) shows the same effect relative to individual request-grant transactions by the controller. In this plot, the average overhead required to service a bus request is plotted for the same incremental increases in bus demand. In the low contention region, this trace remains close to the round trip delay time for the control signals. As batching becomes dominant the average control overhead decreases dramatically until saturation where average overhead reaches its minimum.

It is clear from this simulation that as contention for the bus is increased, a dynamic reduction is achieved in control overhead.

6. Conclusions

In the future, the successful application of electro-optical technology to computing systems will not be limited to network interconnections. As technology produces smaller, faster, and cheaper devices, the ability to produce integrated optical systems will allow the design of hybrid systems using electronics for computations and optics for communications. It is important to isolate the time dependent bottlenecks that will occur at the optical-electronic interfaces.

In this paper we have shown how to remove one of those bottlenecks, the arbitration and control of an asynchronous, distributed bus. Our technique allows arbitration of bus requests to be resolved in a time bounded linearly by the length of the bus and independent of the number of devices on the bus. Using the structures described in this paper, we have introduced a batched service policy which supports demand based bus access to batches with linear priority resolution within each batch. The linear priority resolution mechanism used within a batch results in a reduction of control overhead as batch size increases. Thus, the average overhead time spent in control per bus transaction decreases with increasing bus contention. In addition, this mechanism supports the multiplexing of several messages spatially separated on the bus[16]. Therefore, message latency is amortized over a large number of messages.

We see the application of this research in designs for physically distributed multiprocessor systems, electro-optical crossbar interconnected systems, and fault tolerant "non-stop" systems. The current trend is to implement distributed applications over LAN's. As this trend continues, common bus systems connected via fiber optic buses will provide an environment for such distribution with minimal communications overhead and with transparency in systems and applications software. In electro-optical crossbar interconnected systems, the resolution of contending requests is a critical design issue which may be addressed by this research. Finally, the electrical isolation provided by optically interconnected systems provides an ideal environment for fault tolerant systems where any component may be independently brought up or down for service without effecting overall system operation.

References

1. *Gould Fiber Optics Components*, Gould Electronics, 6711 Baymeadow Drive, Glenn Burnie, MD..
2. *Canstar Fiber Optic Couplers*, Canstar, 3900 Victoria Park Avenue, North York, Ontario.
3. *The LOCALNetter Designer's Handbook*, Architecture Technology Corporation, November 1985.
4. R. A. Bergh, G. Kotler, and H. J. Shaw, "Single-Mode Fibre Optic Directional Coupler," *Electronics Letters*, vol. 16, pp. 260-261, 1980.
5. D. Chiarulli, R. Melhem, and S. Levitan, "Using Coincident Optical Pulses for Parallel Memory Addressing," *IEEE Computer*, vol. 20, no. 12, pp. 48-58, 1987.
6. J. Goodman, F. Loenberger, S. Kung, and R. Athale, "Optical Interconnections for VLSI Systems," *Proceedings of the IEEE*, vol. 72, no. 7, pp. 850-866, July, 1984.
7. A. Goyal, "Lookahead Network," *IEEE Transactions on Communications*, vol. COM-33, no. 11, pp. 1160-1170.
8. D. Hartman, M. Grace, and R. Richard, "An Effective Fiber Optic Electronic Coupling and Packaging Technique Suitable for VHSIC Applications," *IEEE Journal of Lightwave Technology*, vol. LT-4, no. 1, pp. 73-81, January, 1986.

9. P. R. Haugen, A. Husain, and L. D. Hutcheson, "Directions and Development in Optical Interconnect Technology," *SPIE: Optical Computing*, vol. 625, pp. 110-116, 1986.

10. J. Hecht, "Bell Labs Transmits 8 Gbits/s Over 68Km," *Lasers and Applications*, May, 1986.

11. S. Hokanen, A. Tervonen, H. von Bagh, and M. Leppihalme, "Ion Exchange Process for Fabrication of Waveguide Couplers for Fiber Optic Sensor Applications," *Journal of Applied Physics*, vol. 61, no. 1, January, 1987.

12. R. Hunsperger, "Waveguide Fabrication Techniques in GaAs/GaAlAs," *SPIE: Integrated Optical Circuit Engineering*, vol. 517, pp. 9-14, 1984.

13. S. Joshi, "High Performance Networks: A Focus on the Fiber Distributed Data Interface (FDDI) Standard," *IEEE Micro*, June, 1986.

14. B. S. Kawaski, K. O. Hill, and R. G. Lamont, "Biconical-Taper Single-Mode Fiber Coupler," *Optics Letters*, vol. 6, no. 7, pp. 327-328.

15. R. Kelly, J. Jones, V. Bhatt, and P. Pate, "Transceiver Design and Implementation Experience in an Ethernet-Compatible Fiber Optic Local Area Network," in *INFOCOM 84*, 1984.

16. R. Melhem, D. Chiarulli, and S. Levitan, "Space Multiplexing of Waveguides in Optically Interconnected Multiprocessor Systems," *The Computer Journal.* (to appear)

17. M. Nassehi, F. Tobagi, and M. Marhic, "Fiber Optic Configurations for Local AREA Networks," *IEEE Journal on Selected Areas in Communications*, vol. SAC-3, no. 6, pp. 941-949, Nov. 1985.

18. E. G. Rawson and R. M Metcalfe, "Fibernet: Multimode Optical Fibers for Local Computer Networks," *IEEE Transactions on Communications*, July, 1978.

19. J. Reedy and J. R. Jones, "Methods of Collision Detection in Fiber Optic CSMA/CD Networks," *IEEE Journal on Selected Areas of Communications*, vol. SAC-3, no. 6, pp. 890-896, November, 1985.

20. F. Ross, "FDDI: A Tutorial," *IEEE Communications Magazine*, May, 1986.

21. R. Schmidth, E. Rawson, R. Norton, S. Jackson, and M. Bailey, "FIBERNET II: A Fiber Optic Ethernet," *IEEE Journal on Selected Areas in Communications*, vol. SAC-1, no. 5, pp. 702-710, Nov. 1983.

22. S. K. Sheem and T. G. Giallorenzi, "Single-Mode Fiber Optical Power Divider: Encapsulated Etching Technique," *Optics Letters*, vol. 4, pp. 29-31, 1979.

23. William Stallings, *Local Networks,* MacMillan, 1987.

24. F. Tobagi, F. Borgonovo, and L. Fratta, ''Expressnet: A High-Performance Integrated-Services Local Area Network,'' *IEEE Journal on Selected Areas in Communications,* vol. SAC-1, no. 5, pp. 898-912, 1983.

25. F. Tobagi and M. Fine, ''Performance of Unidirectional Broadcast Local Area Networks: Expressnet and Fasnet,'' *IEEE Journal on Selected Areas in Communications*, vol. SAC-1, no. 5, pp. 913-926, 1983.

26. T. H. Wood and M. S. Whalen, ''Effectively Nonreciprocal Evanescant Wave Optical Fiber Directional Coupler,'' *Electronics Letters*, vol. 21, no. 5, pp. 175-176.

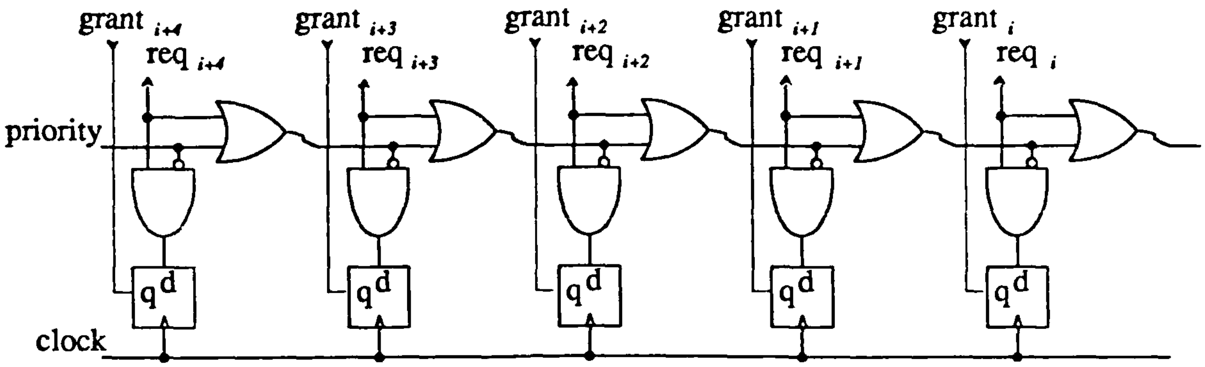


Figure 1a - Electronic synchronous priority chain

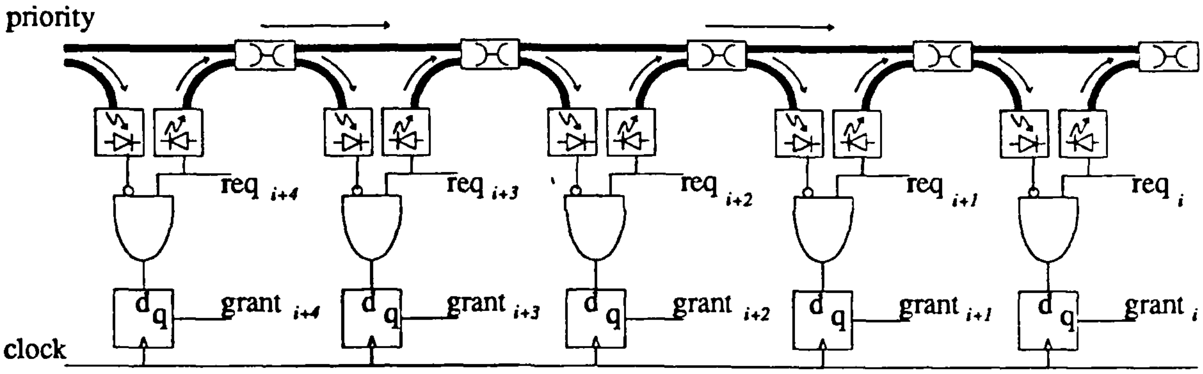


Figure 1b - Optical synchronous priority chain

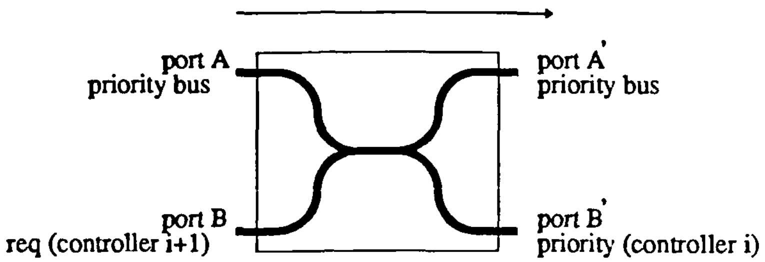


Figure 2 - 2x 2 Optical Coupler

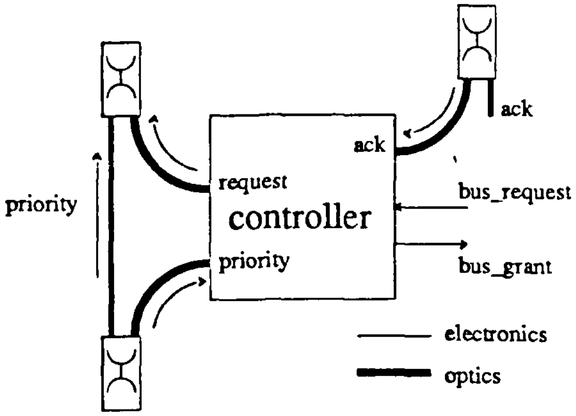


Figure 3 - Controller External Connections

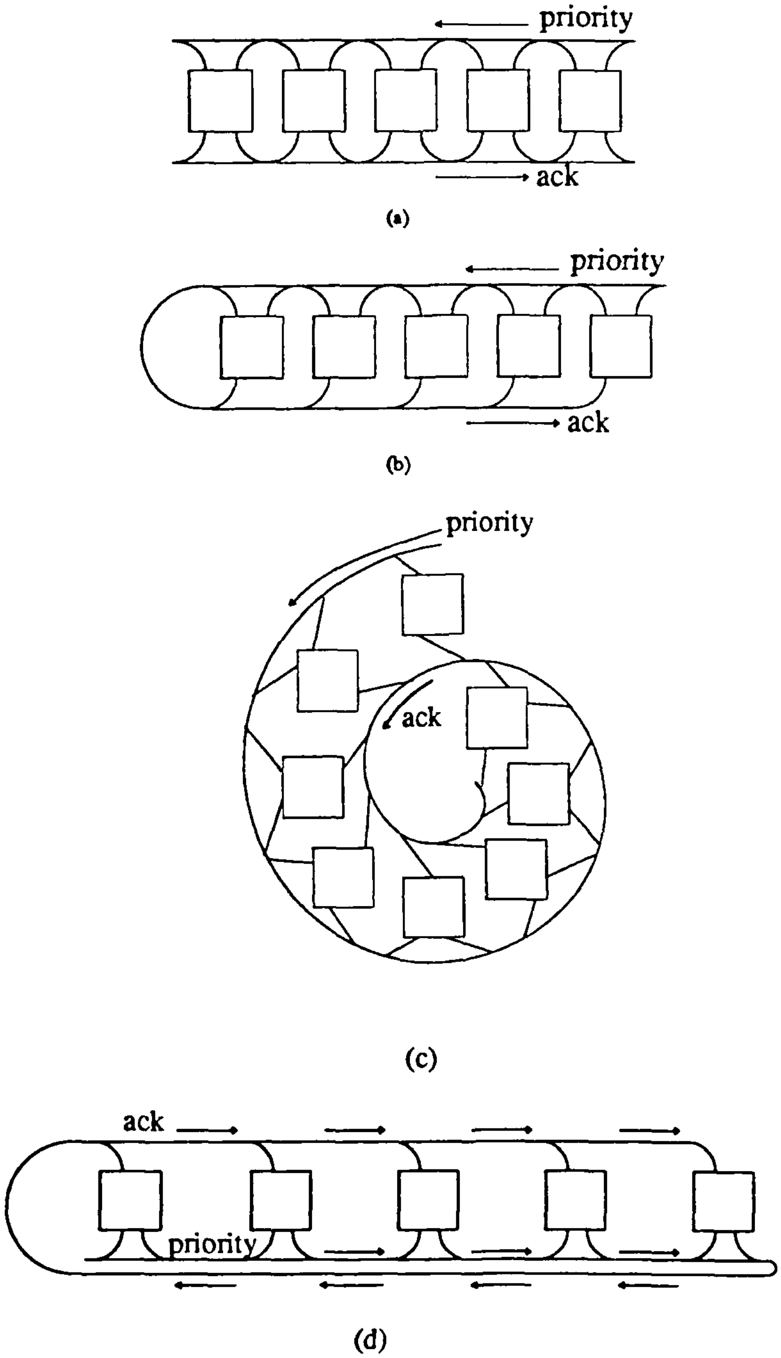


Figure 4 - Ack waveguide feedback structures: (a) dual waveguide (b) linear feedback, (c) spiral feedback (d) folded waveguide

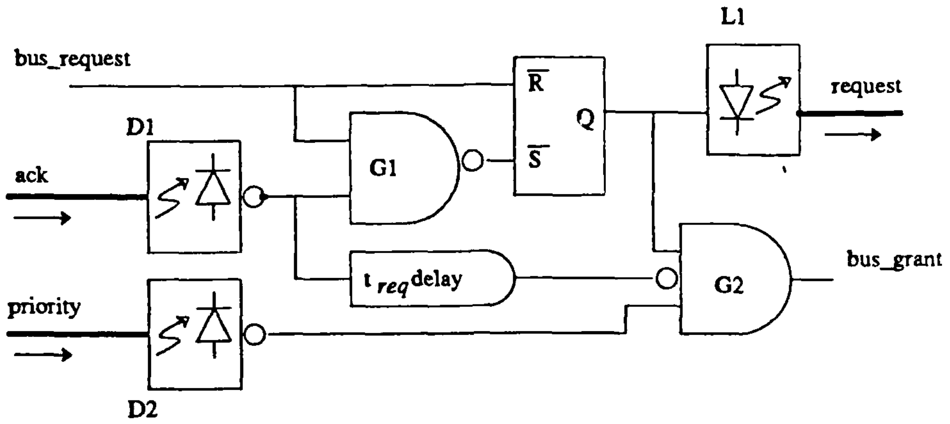


Fig. 5 - Example Controller Circuitry

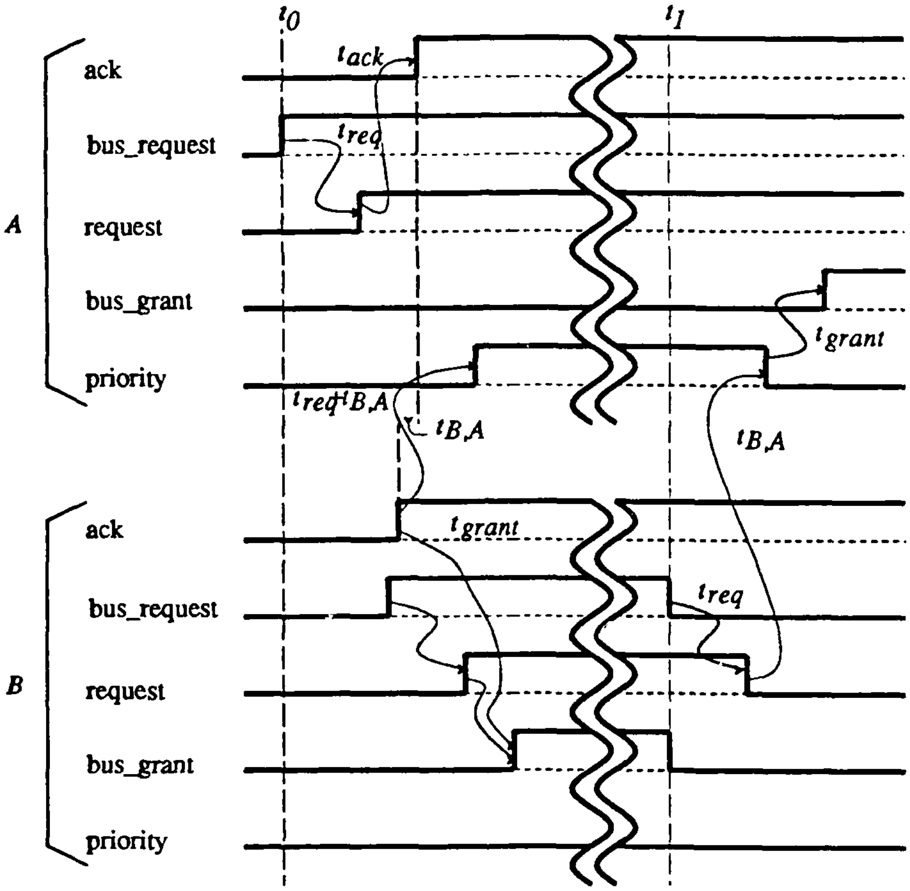
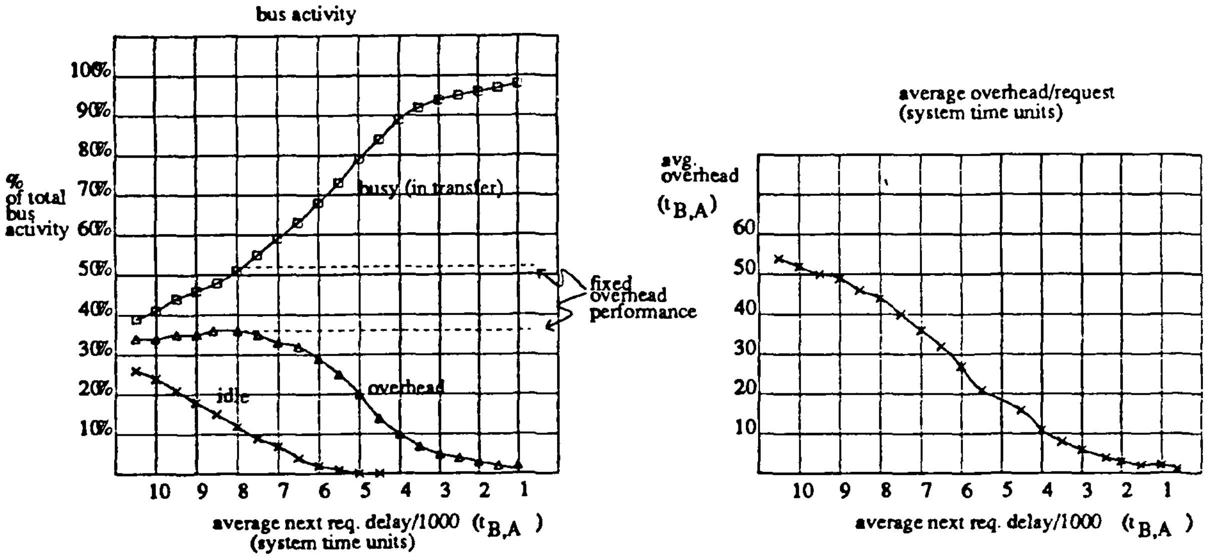


Fig. 6 - Simultaneous Request Timing



(a)

(b)

Fig. 7 - Simulation Results

Space Multiplexing of Waveguides in Optically Interconnected Multiprocessor Systems*

R. G. MELHEM‡, D. M. CHIARULLI‡ AND S. P. LEVITAN§

‡ *Department of Computer Science, the University of Pittsburgh, Pittsburgh, PA 15260, USA*

§ *Department of Electrical Engineering, the University of Pittsburgh, Pittsburgh, PA 15260, USA*

Optical waveguides allow for enhanced bandwidth, loosened loading constraints and large physical distribution of computing resources. Moreover, optics enjoys a unique property that is not shared with electronics, namely the unidirectional propagation of signals. It is this property that is exploited in this paper to increase the effective bandwidth of optical buses. Specifically, a space-multiplexing technique for pipelining messages on optical buses is introduced and analysed. It is shown that pipelined buses support arbitrary routeing permutations in synchronous systems with only linear hardware complexity. Further, a bus arbitration protocol which extends the technique to asynchronous systems is presented. The pipelining of control and data signals represents a significant departure from the conventional exclusive access discipline which characterises bus-interconnected multiprocessors. By relaxing the exclusive access requirement, space multiplexing can support the design of large-scale, distributed, tightly coupled multiprocessor systems.

Received October 1988, revised December 1988

1. INTRODUCTION

There are three fundamental constraints which bound bus interconnections in electronic systems: limited bandwidth, capacitive loading, and cross-talk caused by mutual inductance. Optical systems provide both an opportunity and a challenge to redesign our traditional multiprocessor solutions free of these limitations. Although direct technology substitution may alleviate, to some extent, the communications bottleneck in computer systems, there are obvious limitations to such substitution. For example, any interface between electronics and optics lowers the speed at that interface to the speed of electronics. Even though optical pulses as short as a few femto-seconds may be generated and detected,[4,12] such short pulses may not be used to transmit data on an optical bus, since no existing electronic circuit at the transmitting or the receiving end of the bus can match that speed. In other words, the speed of electronics puts bounds on the transmission speed of optical buses.

Another limitation concerns the end-to-end propagation time of long buses. Due to the absence of self-inductance or -capacitance, long optical buses may be constructed without the need for signal repeaters. Several designs of optical communication networks have already been constructed taking advantage of this property.[1,9,11] However, the control overhead associated with the networking environment is relatively high and cannot support efficient distributed multiprocessing. The major requirement for multiprocessing is that many short messages are transmitted with low overhead. Specifically, the assumption of exclusive access to the bus resource limits throughput to a function of the end-to-end transmission time for the signals on the bus, irrespective of the length of the messages. End-to-end transmission times for optical signals are not inherently shorter than for electronics.

A unique property of optics provides an alternative to exclusive bus access. Namely, the ability in optics to pipeline the transmission of signals through a channel. In electronic buses, signals propagate in both directions from the source. Thus, for directional propagation in pipelines, electronic systems introduce directional amplifiers between adjacent bus sources. These amplifiers cause unpredictable delays, which make any large-scale distributed implementation impractical. On the other hand, optical channels are inherently directional and have predictable delay per unit length. This allows a pipeline of signals to be created by the synchronised directional coupling of each signal at specified locations along the channel. This property, which has been used to parallelise access to shared memory[2] and to minimise the control overhead in networking environments,[16] is applied in this paper to optimise the use of optical buses in multiprocessor systems.

We present a technique for space multiplexing of optical channels in distributed tightly coupled multiprocessors. In the next section we introduce the concept of pipelined optical buses as applied to synchronous processor arrays. We show that by pipelining messages on a single bus we may realise arbitrary routeing permutations. We provide a detailed example of embedding tree interconnections in linear arrays of processors. In Section 3 we continue in the more general framework of asynchronous multiprocessors. The primary concern in such an environment is the realisation of a distributed arbitration mechanism for pipelined buses. We introduce such a mechanism in Section 3.2, and study its performance in Section 3.3. In Section 3.4 we further analyse the technique in the context of large shared-memory multiprocessor systems.

2. SPACE MULTIPLEXING OF WAVEGUIDES IN SYNCHRONOUS PROCESSOR ARRAYS

Multistage networks have been studied extensively[13] as a means of making processor–processor and processor–memory interconnections. In the context of synchronous processor arrays, however, an $n \times n$ multistage network with $\log n$ stages may not realise arbitrary permutations in

* This work is, in part, supported under the Air Force Office of Scientific Research contract AFOSR-88-0198.

† To whom correspondence should be addressed.

a single pass. For example, it has been shown in Ref. 17 that the Omega network[8] needs at least three passes in order to realise arbitrary permutations, and that, alternatively, a network with three log$n$ stages may be used. The hardware complexity of such multistage networks is clearly of the order of $O(n \log n)$. In this section we shall demonstrate that a single optical bus, which has $O(n)$ hardware complexity, may be used to realise arbitrary routeing permutations, as well as many-to-one and one-to-many routeings.

2.1 Pipelining messages on optical buses

Consider a linear array of n nodes connected by a single optical bus (waveguide) that is also connected to a host as shown in Fig. 1. Each node may inject optical signals into the waveguide through a directional coupler. The injected signals propagate from left to right and may be read by any subsequent node on the waveguide. As would be the case in electronics, the bus of Fig. 1 may be used as a medium for broadcasting messages from the host to the nodes. However, because of the directionality of signal propagation, the same bus may also be used to transmit messages from node 1 to node 2, from node 2 to node 3 and, in general, from node i to node $i+1$, $i = 1,\ldots,n-1$, simultaneously.

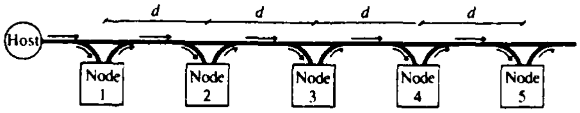


Figure 1. An optically interconnected linear array.

Assuming that each message transmitted between any two nodes consists of b bits, and that each bit is represented by a light pulse of duration w (sec), the concurrent transmission of the $n-1$ messages described above may be accomplished if the following criteria are met.

(1) All transmissions are synchronised to start simultaneously. This may be enforced by the use of a synchronisation signal that arrives at all the nodes at the beginning of each transmission cycle.

(2) The length of the optical path on the waveguide between any two consecutive nodes (d in Fig. 1) is larger than bwc_g, where c_g is the velocity of light in the waveguide. For example, if $c_g = 2 \times 10^8$ m/sec, and 10-bit messages are transmitted at 10 GHz, d should be larger than 20 cm. This minimal optical path length can be reduced by decreasing the message length via parallel transmission on multiple waveguides, by decreasing c_g via the use of waveguides with higher refractive indices, or by shortening the pulse width. Here we note that d is the optical path length between any two nodes, which is not necessarily equal to their physical separation. Also, d does not have to be the same for every pair of adjacent nodes.

Conditions 1 and 2 guarantee that the signals corresponding to two different messages do not physically overlap at any point on the waveguide (hence the term 'space multiplexing'). If the first of the b bits in each message is a start-of-message indicator that is always set to one, a receiving node may pull-off the $b-1$ bits following the start-of-message bit, and ignore any following signal up to the initiation of the next transmission. With this scheme every node may send a message to its right neighbour, simultaneously, on the same waveguide. Note, however, that the initiation of consecutive transmission should be separated by at least nd/c_g sec, where n is the number of nodes in the array. Clearly, the size of the array should be such that the value of nd/c_g is compatible with the computation speed in the nodes. For example, if $d = 10$ cm and $n = 50$ nodes, transmission may be initiated every 25 nsec.

2.2 Realisation of arbitrary permutations

Nearest-neighbour connections are not the only point-to-point communications that may be supported by the single waveguide of Fig. 1. In fact, using space multiplexing, point-to-point messages between any pairs of nodes may be transmitted simultaneously as long as their paths do not intersect in space and time. To be more specific, let $m_{i,dest(i)}$ be a message that is sent from node i to node $dest(i)$, and let $M = \{m_{i,dest(i)};\ 1 \leqslant i < n\}$ be a set of such messages for some one-to-one function $dest$. If conditions 1 and 2 above are satisfied and $dest$ is a strictly increasing function, that is $dest(i) > i$, then all the messages in M may be transmitted on the waveguide simultaneously without causing any signal overlap.

Let $S = \{i : m_{i,dest(i)} \in M\}$ and $D = \{j : m_{i,j} \in M\}$ be the sets of source and destination nodes, respectively, for messages in M. If all the nodes in S initiate transition simultaneously, then a node j in D, where $j = dest(i)$ for some i in S, may have to skip a few messages before reading the message $m_{i,j}$ intended for it. Specifically, j has to skip a number of messages equal to the number of nodes between i and j that are in S. That is:

$$skip(j) = \sum_{l=i+1}^{j-1} \phi(l) \qquad (1)$$

where

$$\phi(l) = \begin{cases} 1 & \text{if } l \in S \\ 0 & \text{otherwise} \end{cases}$$

Hence, by using a single waveguide, it is possible to send messages from any node i to any destination $dest(i) > i$. In order to support communications from i to some destination $dest(i) < i$ a second waveguide may be used as shown in Fig. 2. Clearly, the two directional waveguides in Fig. 2 may support the simultaneous transmission of any set of messages M for any permutation function $dest$. Specifically, M can be partitioned into two set M_1 and M_2 such that M_1 contains the messages with $dest(i) > i$, and M_2 contains the messages with $dest(i) < i$. Messages in M_1 are transmitted on the left-to-right waveguide and messages in M_2 are transmitted on the right-to-left waveguide. The sets of source nodes, S_1, S_2, and destination nodes, D_1, D_2 may be defined for M_1 and M_2, respectively, and at each destination node $j \in D_1 \cup D_2$, the number of messages that

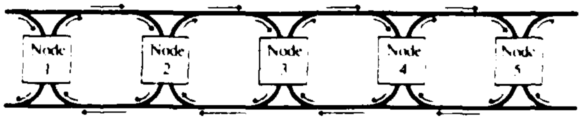


Figure 2. A dual waveguide system.

have to be skipped on a particular waveguide before reading $m_{i,dest(i)}$ may be determined by an expression similar to (1).

The dual waveguide system of Fig. 2 may be viewed as a communication network that may realise any permutation. Given a specific permutation *dest*, a single register, SKIP, may be used at each node j to store information about message reception; the sign of SKIP may indicate whether j is in D_1 or D_2, and its magnitude may indicate the number of messages to be skipped before reading the appropriate message. With this, setting and changing of the interconnection patterns may be accomplished at run-time by programming the values of the SKIP registers at the nodes. Compared to cross-bar switches or multistage interconnection networks, our system uses less hardware (linear with n), eliminates switch delays, and may be re-configured by programming registers that are local to the processors.

The same communication capabilities of the dual waveguide system may be obtained in the folded waveguide system of Fig. 3. In that system each node writes its message on track 1 of the waveguide and senses any signal on the waveguide through a photo-detector coupled to track 2. At the reception of the synchronisation signal, each node puts its message (b bits) on track 1 of the waveguide. The n messages form a train which travels on track 2, thus allowing each node to read the message that is destined to it. As in the single-waveguide system, a register SKIP may be used at each node to indicate the number of messages to be skipped before reading the appropriate message. In this case,

$$skip(j) = \sum_{l=i+1}^{n} \phi(l),$$

where $\phi(l)$ is as defined in (1). Note that the folded waveguide system uses less couplers and photodetectors than the dual-waveguide system at the expense of doubling the optical length of the bus. A new round of communication may be initiated on the folded waveguide every $2nd/c_g$ secs.

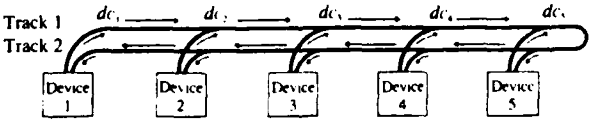


Figure 3. A folded waveguide system.

In addition to allowing arbitrary permutations, a pipelined optical bus may allow many-to-one communications if the appropriate number of SKIP registers are provided. For example, if two registers are provided in each node, each node may receive up to two messages in each bus cycle. Moreover, if the length of the optical path d between two adjacent nodes is increased to $Kbwc_g$, where, as before, bwc_g is the length of a single message, then each node may send up to K messages in each bus cycle, and thus one-to-many communications may also be allowed. This flexibility may be applied to realise several logical interconnections. We demonstrate this capability by an example.

2.3 Tree-interconnections on pipelined buses: an example

Consider a multiprocessor system which consists of $n = 2^L - 1$ processors logically connected in a complete binary tree structure. If a breadth-first numbering is used to identify the processors, the tree connection implies that a processor j should be connected to its parent, processor $\lfloor j/2 \rfloor$, and to its children, processors $2j$ and $2j+1$ (see Fig. 4(a) and Fig. 5(a) for examples). If the n processors are connected by a dual-waveguide system similar to the one shown in Fig. 2, the left-to-right waveguide may support messages from parent processors to children processors, and the right-to-left waveguide may support messages from children processors to parent processors.

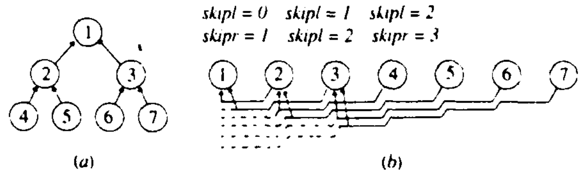


Figure 4. Simultaneous transmission from children processors to parent processors.

First, we illustrate the many-to-one communication capability by assuming that each processor is to send a message to its parent. Clearly, a processor $j, j < 2^{L-1}$, will receive one message from each of its two children, and hence two skip registers are needed at each node. Let $skipl(j)$ and $skipr(j)$ be the number of messages that a node j has to skip before reading the messages addressed to it by its left child (processor $2j$) and right child (processor $2j+1$), respectively. Each of the $2j-j-1$ processors between processor $2j$ and processor j sends a message to its parent, and hence processor $2j$ will have to skip $j-1$ messages before reading the message of its left child. That is

$$skipl(j) = j-1 \quad j = 1, \ldots, 2^{L-1}-1$$

Similarly, w· find that

$$skipr(j) = j \quad j = 1, \ldots, 2^{L-1}-1$$

The values of *skipl* and *skipr* are shown in Fig. 4(b) for the case $L = 3$.

In order to illustrate the one-to-many communication capability, we assume that each processor (except the leaf processors) is to send out two messages, the first to its left child and the second to its right child. If each processor writes its two messages consecutively on the right-to-left waveguide, and the length of the optical path between any two processors is larger than $2bwc_g$, where b is the number of bits per message, the messages will not overlap on the waveguide. Let $skip(j)$ be the number of messages that a processor j has to skip before reading the message sent to it by its parent, namely processor $\lfloor j/2 \rfloor$, where $\lfloor j/2 \rfloor$ is the largest integer smaller than $j/2$.

If j is even, then there are $j/2-1$ processors between processor j and its parent, $j/2$. Denote by S_j the set containing these processors. Now, if j is not a leaf processor each processor in S_j will write two messages on the waveguide, and j will have to skip these messages before reading its message. That is, $skip(j) = 2(j/2-1) = j-2$. However, if j is a leaf processor $j-2^{L-1}$ of the processors in S_j are also leaf processors that will not write any messages on the waveguide. In this case, j will have to skip only $2(j/2-1-j+2^{L-1}) = 2^L-j-2$ messages before reading its message.

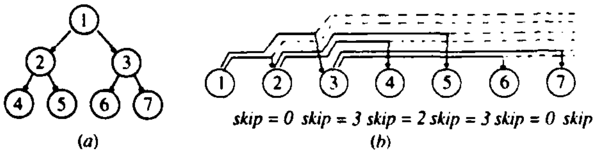


Figure 5. Simultaneous transmission from parent processors to children processors.

If j is odd, its parent is $(j-1)/2$ and the set S_j contains $(j-1)/2$ processors. Again, if j is not a leaf, each processor in S_j will write two messages on the waveguide. In addition to skipping these messages, processor j will have to skip one more message because its parent, $(j-1)/2$, writes the message destined to its left child, $j-1$, before the one destined to its right child j. Hence, in this case, $skip(j) = j$. However, if j is a leaf processor the 2^L-j-2 leaf processors in S_j will not write any message on the waveguide resulting in $skip(j)-j$. In summary, we have

$$skip(j) = \begin{cases} j-2 & \text{if } j \text{ is even and } j \text{ is not a leaf processor} \\ j & \text{if } j \text{ is odd and } j \text{ is not a leaf processor} \\ 2^L-j-2 & \text{if } j \text{ is even and } j \text{ is a leaf processor} \\ 2^L-j & \text{if } j \text{ is odd and } j \text{ is a leaf processor} \end{cases}$$

The values of *skip* are shown in Fig. 5(*b*) for a tree with three levels.

Finally, it should be noted that an analysis similar to the one presented above may be applied if a folded waveguide system is used. Also, pipelined buses may be used to realise other logical interconnection structures such as the barrel switch networks, the shuffle/exchange networks and the hypercube network.[6]

3. MESSAGE PIPELINING IN ASYNCHRONOUS MULTIPROCESSORS

In the previous section, space multiplexing was applied to computational models which assume a fixed communication pattern at each cycle. Systolic arrays and SIMD multiprocessors are clear examples of such models. However, the same dual-waveguide and folded-waveguide configurations shown in Figs 2 and 3 may also be used to multiplex messages between arbitrary sources and destinations. Clearly, this may be accomplished if each message includes the address of its destination, and if individual messages are framed by appropriate delimiters. As described earlier, each of the n devices may put its message on the bus during the same bus cycle, provided that the bits of different messages do not overlap. That is, provided that all devices start the transmission simultaneously and that the length d, of the optical path between any two devices satisfies $d \geqslant d_{min} = wb_{max}c_g$, where b_{max} is the number of bits in the longest message (address + data + delimiters). For a given d_{min}, this condition represents an upper limit on the length of the messages that may be transmitted on the bus.

For asynchronous MIMD multiprocessor systems, it may not be possible to synchronise the simultaneous transmission of messages to the accuracy required in optical systems, especially if the transmitting devices are physically separated, as is the case in distributed multiprocessor systems. In order to resolve this problem, a *grant* signal may be issued by a central controller and propagated through the processor array on a separate waveguide. For example, in the folded-waveguide configuration, the grant signal may be propagated in a direction opposite to that of track 1 of the message waveguide (see Fig. 6). The arrival of the grant signal at device i initiates the transmission of the message from that device. Because the *grant* signal and the signal on track 1 of the message waveguide propagate in opposite directions, the bus cycle becomes $2nd/c_g$ and the condition on the inter-device optical path length d becomes $d \geqslant 0.5\,wb_{max}c_g$.

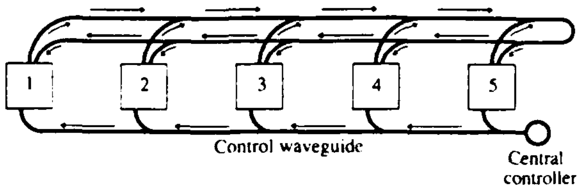


Figure 6. Synchronising transmission in physically distributed systems.

So far we have described a multiplexing method in which a train of n slots, one for the message transmitted by each device, is pipelined on the waveguide. This is quite acceptable provided that the messages fit in the designated slots, and provided that a central controller assumes the charge of issuing the synchronisation signal or the grant signal. In distributed asynchronous multiprocessor systems these conditions are not generally satisfied, and different techniques must be applied to arbitrate the access to the bus. A distributed arbitration technique is described in the next section.

3.1 Distributed control of pipelined optical buses

In addition to pipelining the data signals on the bus, the unidirectional propagation of optics may be applied to pipelining the control signals on the bus. The arbitration mechanism described in this section relies on the directional propagation of signals on a control waveguide that is folded into three tracks as shown in Fig. 7. This three-track waveguide is similar to the one used in Refs. 15 and 16 to pipeline modulated data signals, except that it is used exclusively for control signals. The mechanism results in a batch priority queue protocol.[5] In brief, all devices that request the bus while the bus is idle form a batch, and requests in the batch are serviced in linear priority until all requests are satisfied. Requests that arrive while a certain batch is being serviced have to wait until all the requests in that batch are serviced and then form a new batch.

Normally, the control waveguide is at logic zero (no light). A device which wants to use the bus may assert a *request* signal on track 1 of the control waveguide only if a logic zero is read on the *ack* input which is coupled to track 3 of the control waveguide. If *ack* is high, this is considered as an indication that a batch has already been formed, and hence the device has to wait until that batch is serviced before it may assert its request. As will be explained later, this will be indicated by a high-to-low transition on *ack*.

Clearly, the end of a batch formation period is signalled at each device by a rising edge on *ack* (low-to-

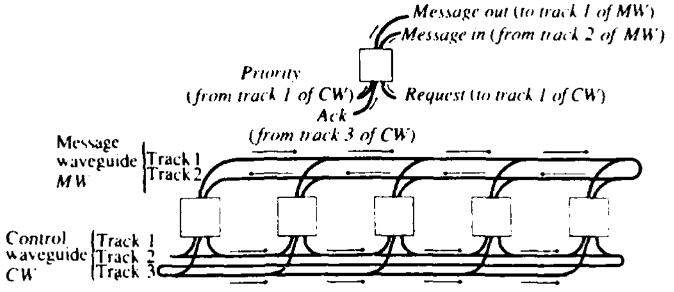

Message out (to track 1 of MW)
Message in (from track 2 of MW)
Priority (from track 1 of CW)
Request (to track 1 of CW)
Ack (from track 3 of CW)
Message waveguide MW
Track 1
Track 2
Control waveguide CW
Track 1
Track 2
Track 3

Figure 7. Distributed control of pipelined optical buses.

high transition). This edge, which travels towards the right on track 3 of the control waveguide, is the feedback of a signal that is generated on track 1 of that waveguide by some device asserting a request signal.

A device is included in the current batch as soon as it asserts *request*. However, it is not granted control of the bus until it reads both a one on *ack* (end of batch formation) and a zero on the *priority* input, which taps track 1 of the control waveguide. With this scheme, after the formation of a batch, the devices within the batch will be granted control of the bus in the order at which they are connected to the control waveguide. Hence, in Fig. 7, the leftmost device requesting the bus is granted the bus first. Here we note that the only function of track 2 of the control waveguide is to cause the feedback signal on track 3 to travel in the same direction as the request signal on track 1. As proved in Ref. 3, this ensures that the arbitration mechanism handles simultaneous requests correctly.

After a device i is granted the bus (*ack* = 1 and *priority* = 0), it sends its message on the message waveguide, and after sending the last bit of the message it relinquishes the bus by lowering the *request* line. This will cause a low-going edge to travel on track 1 of the control waveguide. The next requesting device, say j, will receive that edge at the same time when the last bit of the message sent by device i passes through the coupler dc_j. In other words, when device j is granted control of the bus upon the high-to-low transition on *priority*, it may immediately start to transmit its own message on the message waveguide. The two messages from device i and device j, say m_i and m_j, will thus be pipelined on the message waveguide. The spacing (separation) between m_i and m_j depends on the time required for processing the control signal within the controller of device j.

Since electronic delays cannot be predicted to the accuracy of optics, it is impossible to eliminate completely the separation between messages in the pipeline. However, such separation can be reduced if the *request* signal of device i is lowered before the transmission of the last bit of m_i by a time period $\tau_{e\ min}$ equal to the minimum electronic delay expected in a controller. The actual separation between messages will therefore vary within the specified tolerances of a small number of gates in the control circuit. Specifically, if $\tau_e = \tau_{e\ min} + \Delta_e$ is the actual time consumed by device j to process the control signal, then m_i and m_j will be separated on the waveguide by a distance of $\Delta_e c_g$. This, of course, assumes that the time to transmit m_i is larger than $\tau_{e\ min}$. If this is not the case, then the separation between m_i and m_j becomes $(\tau_e - b_i w) c_g$, where b_i is the number of bits in m_i and $1/w$ is the baud rate used for transmission. In order to simplify future analysis, we assume that the separation between messages is $\tau_e c_g$, where τ_e is a delay due to few electronic gates (for a precise estimation of τ_e we refer to Ref. 3).

With the above scheme, all the messages transmitted by devices in a particular batch will be pipelined into a train that will travel on track 2 of the message waveguide and thus will be seen by every device. It is the responsibility of each device to recognise its address within each message and to read the messages that are addressed to it.

When the rightmost device in the batch finishes sending it message and lowers its *request* line, the corresponding low-going edge will travel on the control waveguide all the way to track 3 of that waveguide. The detection of this high-to-low transition on the *ack* input of a device is interpreted by that device as a signal of completion of the current batch, and thus as a permission to assert *request* if the device was waiting to request the bus.

3.2 Control overhead

In the above bus arbitration scheme there is an overhead associated with forming a batch and with signalling the termination of a batch. For a given batch we use the term 'batch initiation' to refer to the instant of time at which the first device in the batch asserts its request signal. We also use the term 'batch termination' to refer either to the instant of time at which all devices are notified that the service of the current batch has been completed, or to the initiation of the next batch, whichever is first. Note that in the case of high bus contention the bus is never idle, and thus batch termination is always due to the initiation of the next batch. Now, we may define *batch formation* overhead as the time interval between batch initiation and the instant when the highest priority device in the batch starts the transmission of its message. Similarly, we may define *batch release* overhead as the time interval between the instant at which the lowest priority device in the batch finishes the transmission of its message and batch termination.

In order to estimate the overhead associated with batch control in a high-contention environment, we assume that device r is the lowest-priority (rightmost device) in some batch and that device f is the leftmost device that is waiting for the termination of that batch to request the bus. In this case, by tracing the control signals on the control waveguide it is easy to see that the *batch release* overhead is equal to $2\tau_{1,n} - \tau_{f,r}$, where $\tau_{i,j}$ is

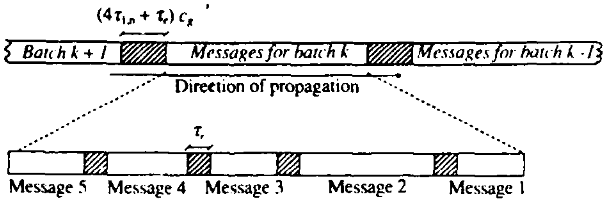


Figure 8. A snapshot of the pipelined messages on track 2 of the message waveguide.

the signal propagation delay between devices i and j, and thus $\tau_{1,n}$ is the bus end-to-end propagation delay. Given that f is the highest-priority device in the next batch, the *batch formation* overhead for that batch is equal to $2\tau_{1,n}+\tau'_e$, where $2\tau_{1,n}$ is the time for the request signal asserted by f on track 1 of the control waveguide to wrap around and reach the *ack* input of f, and τ'_e is the logic delay from the time the *ack* input of f goes high to the time f starts writing its message on the bus. This logic delay is of the same order as the logic delay τ_e discussed earlier, and it will be assumed, for simplicity, that $\tau'_e = \tau_e$.

Hence, the time interval between the instant at which r finished the transmission of its message until the instant at which f starts its transmission is $4\tau_{1,n}-\tau_{f,r}+\tau_e$. This interval creates a spatial separation on the message waveguide between the two trains of messages corresponding to the two batches. Specifically, if, as in Fig. 7, the direction of propagation on the message waveguide is identical to that of tracks 1 and 3 of the control waveguide, then, as shown in Fig. 8, the spatial separation between the two trains of messages is equal to $(4\tau_{1,n}+\tau_e)c_g$.

3.3 Performance analysis

An important measure of performance of any bus control protocol is its efficiency, defined as the ratio $\eta = T_d/(T_c+T_d)$, where T_d is the time spent for transmitting data on the bus and T_c is the time spent in controlling the bus. This ratio is particularly important if the bus is subject to high contention. That is, if the bus is never idle; an assumption that we will maintain throughout this section.

The optical bus protocol described in the previous section utilises the property of undirectional propagation of optical signals in two different ways. First, by pipelining control signals, the control overhead of $4\tau_{1,n}+\tau_e$ is only paid for each batch rather than for each message. Secondly, due to the pipelining of data signals, each device holds the bus only while writing a message on the bus and there is no need to wait for the signals to reach their destination after each transmission. Assuming that N_{av}, $1 \leqslant N_{av} \leqslant n$ is the average number of devices included in each batch, and that b_{av} is the average number of bits in a message, then, over a time span required to process a given number of batches, the efficiency of the pipelined bus protocol is

$$\eta_{\text{pipe}} = \frac{N'_{av}\beta_{av}}{4\tau_{1,n}+\tau_e+N_{av}(\beta_{av}+\tau_e)}, \tag{2}$$

where $\beta_{av} = b_{av}w$ is the average time length of a message. If the same number of messages are transmitted on an electronic or an optical bus, using a protocol which does not exploit the unidirectional propagation of signals, the upper bound on the efficiency is

$$\eta_{\text{non pipe}} = \frac{N_{ab}\beta_{av}}{N'_{av}(2\tau_{1,n}+2\tau_e+\beta_{av})}, \tag{3}$$

where it is assumed that the minimum arbitration time for each message is $\tau_{1,n}+2\tau_e$, for some logic delay τ_e. It is also assumed that the physical location of the receiver is not known to the sender, and hence each sender should not relinquish the bus unless it is certain that the message reached the receiver. This requires, at least, an average time of $\tau_{1,n}$ per message. Note that the efficiency of non-pipelined buses does not reach the bound (3) in practice.

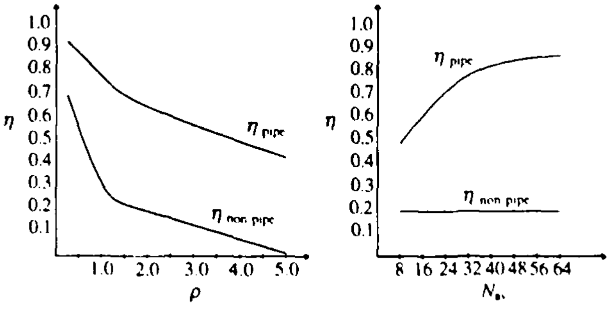
By comparing (2) and (3), it becomes clear that the efficiency of pipelined buses is always larger than that of non-pipelined buses as long as $N_{av} > 1$ – that is, as long as each batch contains more than one request. Moreover, pipelining the bus becomes more advantageous when the arbitration overhead is dominated by the end-to-end propagation delay – that is, when $\tau_{1,n} \gg \tau_e$. For example, if sub-nanosecond ECL electronics is used for the control logic and the bus is longer than a few metres, then $\tau_{1,n} \gg \tau_e$, and the effect of τ_e in (2) and (3) becomes very small. In this case, η_{pipe} and $\eta_{\text{non pipe}}$ may be approximated to

$$\eta_{\text{pipe}} = \frac{N_{av}}{2\rho+N_{av}} \tag{4}$$

and

$$\eta_{\text{non pipe}} = \frac{1}{(2\rho+1)}, \tag{5}$$

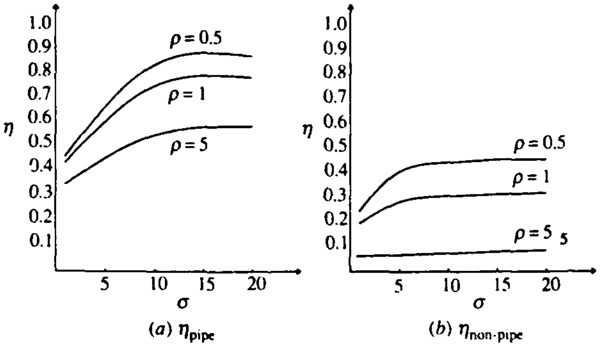
where $\rho = \tau_{1,n}/\beta_{av}$ is the ratio of bus length to message length. In Fig. 9(a), Equations (4) and (5) are plotted for a bus connecting $n = 64$ processors, assuming that $N_{av} = n/4$. The plot shows that pipelined buses may be efficiently used even for values of ρ larger than unity. For such ρ, the length of the messages is smaller than the end-to-end delay and the overhead associated with non-pipelined buses becomes prohibitive. In Fig. 9(b), we fix ρ at $\rho = 2$ and show the effect of N_{av} on the efficiency of pipelined buses. As expected, the efficiency of the pipelined protocol increases when the bus becomes busier and the batches become larger.



(a) $N_{av} = 16$ (b) $\rho = 2$

Figure 9. Bus efficiency for $\tau_{l,n} \gg \tau_e$.

Pipelining the bus reduces the control overhead, even for short buses in which the logic delay τ_e may not be neglected. In order to illustrate this, we consider Equations (2) and (3) with $n = 64$ and $N_{av} = n/2$ and we plot, in Fig. 10, the bus efficiency against the ratio ($\sigma = \beta_{av}/\tau_e$) for different values of ρ. If the same electronic technology is used for both the controller circuit and the transmitter circuits, it is reasonable to assume that the width of the pulses used for transmission is approximately equal to one logic level delay in the controller circuit. With this assumption, $\sigma = b_{av}/g$, where b_{av} is the average number of bits in a message and g is the number of logic levels in the controller circuit (in the controller design described in Ref. 3, $g = 5$). We have chosen to use σ instead of β_{av} in Fig. 10 because σ is a measure of the length, in bits, of the messages transmitted on the bus, and is independent of the electronic technology used in the system.



(*a*) η_{pipe} (*b*) $\eta_{non\text{-}pipe}$

Figure 10. Bus efficiency as a function of σ.

3.4 Application to distributed common bus multiprocessors

The results of the previous section indicate that optical buses may be used for the construction of distributed multiprocessor systems with relatively large physical separations. Specifically, pipelining the signals on the bus provides high message throughput and thus large bandwidth, even for relatively short messages. In order to illustrate this point, we consider a 64-multiprocessor system with $N_{av} = n/2 = 32$, $b_{av} = 32$, $g = 5$, and we assume a 1 nsec delay per logic level. For such a system, $\sigma = 6.4$ and $\rho = 0.104L$, where L is the length of the bus in metres. By substituting these values in (3) we find that, in order to maintain a bus efficiency of more than 0.5, the physical length of the bus should be less than 3.4 m. From (2), this same efficiency may be obtained for pipelined buses of length up to 64 m. The ability of pipelined optical buses to support distributed multiprocessors increases as the number of processors increases. For example, with $N_{av} = 128$, $\sigma = 6.4$, $\rho = 0.104L$ and a 50% bus efficiency, pipelined optical buses of length up to 259 m may be used. This clearly allows for the construction of distributed multiprocessor systems across buildings in university campuses or industrial sites.

Pipelining the messages on the bus, however, does not shorten the delay required for delivering any particular message. This delay remains proportional to the length of the path travelled by that message. For shared-memory multiprocessor systems such delay is especially critical, since it represents a performance bottleneck. In the remaining part of this section we shall analyse in some detail the memory-response time in shared-memory systems interconnected by pipelined optical buses.

Consider a system of n processors, each connected to a local memory such that the n local memories form a global memory space that is addressable by any processor in the system. The processors are connected by an optical bus that is used to transmit memory requests to non-local memories. Also, memory modules receiving read requests from remote processors use the bus to send back the content of the requested memory locations. For simplicity, it will be assumed that messages are of some fixed length, say b bits. This is a reasonable assumption, since only three types of message are needed for 'read requests', 'write requests' and 'returned data', and since separate message waveguides may be used for address, data and processor identification. If, as before, the bus cycle time, τ_{cycle}, is defined as the time for servicing the bus requests in one batch plus the associated control overhead, then

$$\tau_{cycle} = 4\tau_{1,n} + \tau_e + N_{av}(\beta + \tau_e), \tag{6}$$

where $\beta = bw$.

The memory-response time for a read operation, denoted by τ_{fetch}, is the time elapsed between the instant when a processor issues a memory read request and the instant it receives the addressed data. For a non-local memory read, the value of τ_{fetch} depends on the location of the processor relative to the addressed memory module, and on the status of the bus at the time of the request. In order to compute the worst response time, we consider a processor P, which issues a read request to a module M just after the formation of a batch. In this case, P has to wait for one complete bus cycle before it can even request control of the bus. Thus the read request will be transmitted on the bus during the following bus cycle (call this cycle 2). If P is the lowest-priority device on the second batch, M will receive the request at the end of cycle 2 and may not have time to fetch the data before the formation of the third batch. Thus the data will be sent back to P during cycle 4, and if M has a low priority the data will reach P at the end of cycle 4. In other words, τ_{fetch} may be as high as $4\tau_{cycle}$.

The above analysis assumes that memory contention within M may be resolved in a τ_{cycle} time. That is, if a given batch contains k read requests, $k \leqslant n$, for locations in M, then all the k requests will be fetched during cycle 3, stored in a queue, and sent back during cycle 4. If this is not possible some requests will not be ready during cycle 4, and the response time for these requests will be greater than $4\tau_{cycle}$. This is a memory-contention problem, which is typical for all shared-memory multiprocessor systems. In our system such contention will only delay memory access to M, but unlike multistage interconnection networks will not affect request to other memory modules. Specifically, for multistage networks, this situation, referred to as a hot spot[10], creates a saturation tree which affects all memory requests, including those that do not address M.[7]

By carefully following the path of the different control signals in the pipelined bus, it becomes clear that, even if P is the lowest-priority device and the read request to M is the last message in batch 2, there is a period of $2\tau_{1,n}$ between the instant when M receives the read request and the instant when M receives the rising edge on *ack* indicating the formation of batch 3. Hence, if M can fetch the required location in a time less than $2\tau_{1,n}$ the requested data may be sent back on the bus during cycle 3. For relatively long buses the above condition is usually satisfied and thus the upper bound on τ_{fetch} is $3\tau_{\text{cycle}}$. The lowest bound on τ_{fetch} is obtained when P is the lowest-priority device and M is the highest-priority device. In this case it takes at least $\tau_{1,n}$ time units for the request of P to reach M. The next batch, which includes the reply of M, requires $2\tau_{1,n}$ time units for its formation, and an additional $\tau_{1,n}$ is needed for the data to travel back from M to P. In other words,

$$4\tau_{1,n} \leqslant \tau_{\text{fetch}} \leqslant 3\tau_{\text{cycle}}. \qquad (7)$$

For a distributed 64-processor system connected by a 100 m-long bus, if $N_{\text{av}} = 32$, $b = 32$, $w = 1$ nsec and $\tau_e = 5$ nsec, we have $4\tau_{1,n} = 1.32$ μsec and $N_{\text{av}}(\beta + \tau_e) = 1.18$ μsec. Hence, from (6) and (7), the memory-access time for non-local memory is between 1.32 and 7.5 μsec. This is comparable to the memory-access time in multiprocessor systems that use multistage interconnection networks.[14] However, the use of optical buses allows for a separation of 100 m between the processors in the system. In addition, a bus structure has the obvious advantage of using hardware that is linear with the number of processors.

4. CONCLUDING REMARKS

In this paper we have presented a technique for space multiplexing of signals on optical waveguides. This technique uses the unique transmission characteristics of optical signals to construct message pipelines in the communication channels of multiprocessors. We have shown that it is possible to realise a variety of interconnection structures for both synchronous and asynchronous systems. The achievable performance is comparable to that of multistage networks and the hardware complexity is linear with the number of processors. Further, the use of optics allows for the system to be distributed over distances that are larger than those feasible in electronics.

Thus we have shown that is is possible to implement large-scale distributed and tightly coupled multiprocessors. We foresee these systems as an alternative to multistage interconnection networks for implementing the next generation of supercomputers.

REFERENCES

1. R. Allan, Low-loss tapping opens doors to optical network buses. *Electronic Design* (Oct. 1984).
2. D. Chiarulli, R. Melhem and S. Levitan, Using coincident optical pulses for parallel memory addressing. *IEEE Computer*, **20** (12), 48–58 (1987).
3. D. Chiarulli, S. Levitan and R. Melhem, *Asynchronous control of optical buses in closely coupled distributed systems.* Technical Report 88-2, Department of Computer Science, the University of Pittsburgh (1988).
4. J. Fujimoto, A. Weiner and E. Ippen, Generation and measurement of optical pulses as short as 16 fs. *Applied Physics Letters*, **44** (9) (1984).
5. D. Gustavson, Introduction to the fastbus. *Microprocessors and microsystems.* **10**, (2), 77–85 (1986).
6. K. Hwang and F. Briggs. *Computer Architecture and Parallel Processing*, McGraw-Hill, Maidenhead (1984).
7. M. Kumar and G. Pfister, The onset of hot spot contention. *Proc. Int. Conf. Parallel Processing*, pp. 28–34 (1986).
8. D. Lawrie, Access and alignment of data in an array processor. *IEEE Transactions on Computers*, **C-24** (12), 1145–1155 (1975).
9. M. Nassehi, F. Tobagi and M. Marhic, Fiber optic configurations for local area networks. *IEEE Journal on Selected Areas in Communications*, **SAC-3** (6), 941–949 (1985).
10. G. Pfister and V. Norton, Hot spot contention and combining in multistage interconnection networks. *Proc. Int. Conf. Parallel Processing*, pp. 790–797 (1985).
11. R. Schmidt, E. Rawson, R. Norton, S. Jackson and M. Bailey, FIBERNET II: a fiber optic ethernet. *IEEE Journal on Selected Areas in Communications*, **SAC-1** (5) 702–710 (1983).
12. C. Shank, The role of ultrafast optical pulses in high speed electronics. In *Picosecond Electronics and Opto-electronics*, edited G. Morou, D. Bloom and C. Lee. Springer, Heidelberg (1985).
13. H. J. Siegel. *Interconnection Networks for Large-Scale Parallel Processing: Theory and Case Studies.* Lexington Books, Lexington, Mass (1984).
14. R. Thomas, R. Gurwitz, J. Goodhue and D. Allen, *Butterfly Parallel Processor Overview.* BBN Report no. 6148. Bolt Beranek and Newman, Cambridge, Mass. (1986).
15. F. Tobagi and M. Fine, Performance of unidirectional broadcast local area networks: Expressnet and Fastnet. *IEEE Journal on Selected Areas in Communications*, **SAC-1** (5), 913–926 (1983).
16. F. Tobagi, F. Borgonovo and L. Fratta, Expressnet: a high-performance integrated-services local area network. *IEEE Journal on Selected Areas in Communications*, **SAC-1**, (5) 898–912 (1983).
17. C. Wu and T. Feng, Universality of the shuffle exchange network. *IEEE Trans. on Computers*, **C-30** (5), 324–331 (1981).

Technical report 89-22

Coincident Pulse Techniques for Multiprocessor Interconnection Structures

Steven P. Levitan
Donald M. Chiarulli
Rami G. Melhem

Department of Computer Science
The University of Pittsburgh
Pittsburgh, PA 15260

Coincident Pulse Techniques for Multiprocessor Interconnection Structures

Steven P. Levitan

Donald M. Chiarulli

Rami G. Melhem

University of Pittsburgh
Pittsburgh, PA 15260

ABSTRACT

We present several optical interconnection structures which support communication requirements unique to multiprocessor systems, namely, broadcasting, multicasting, simulcasting and multiport memory access. The structures are based on guided-wave time division multiplexed channels and use coincident pulse techniques to optically demultiplex individual bits at selected destinations. We describe one dimensional and two dimensional structures which are appropriate for processor to processor interconnections and for processor to memory interconnections, respectively.

June 8, 1989

This research is supported in part by a grant from the Air Force Office of Scientific Research under contract #AFOSR-88-198

Coincident Pulse Techniques for Multiprocessor Interconnection Structures

Steven P. Levitan

Donald M. Chiarulli

Rami G. Melhem

University of Pittsburgh
Pittsburgh, PA 15260

1. Introduction

In this paper we present several optical switching structures which are appropriate for multiprocessor interconnection applications. Multiprocessor interconnection structures have extensively studied[1,2,6,7,23] and range from bus interconnected systems through a variety of permutation networks. In most cases the communication modes supported by these structures can be classified as either broadcast systems, which are typically used in shared memory implementations, or point-to-point systems which support direct communication between processors or between processors and memory. It has been suggested by Levitan[13] that it is desirable to additionally support multicasting and simulcasting modes of communication. These modes are not widely implemented in electronics due to the complexity of their implementation. However, using optical techniques such structures can be realized efficiently.

To implement these structures we exploit two properties of optical signals: unidirectional propagation and predictable path delays. These properties have allowed us to use the relative path length between two signals as a system timing mechanism. Further, the relation between time and space within a waveguide allows us to positionally encode information which normally requires complex decoding structures.

Both free space and guided wave structures for multiprocessor interconnections have been examined previously. For example, optical crossbar switches and multi-stage networks in free space have been proposed in[3,8,9,20,21,25]. Optical fiber space division switching (SDS) systems using arrays of electro-optic switching elements have been demonstrated[10]. Wavelength division switching (WDS) systems[26] and more recently systems using combinations of WDS, SDS and TDS[15,24] have also been proposed. Time division switching (TDS) systems in a variety of switch and memory configurations for pulse interchange have been studied[16,22,27,28]. Fiber based structures for digital circuitry[11] and interconnection structures[19] have also been proposed.

In our research, we have concentrated on guided wave TDM systems. For example, in [5], we have shown the application of TDM techniques to the implementation of parallel memories. In [4], we have provided a solution to decentralized bus arbitration for optically interconnected and physically distributed multiprocessors. In [12], we have extended this work and addressed the bus latency problem by amortizing end-to-end propagation delay over a large number of concurrent bus transfers, and shown that arbitrary interconnection permutations may be realized with systems of hardware complexity linear in the number of processcrs.

Time division multiplexing schemes have traditionally been limited by the bandwidth ratios of the multiplexed optical signals to the circuitry at the optical-electronic interface. While tapped delay line structures can be used to encode multiplexed pulse trains, high speed demultiplexing at the receivers represents a significant problem. The optical self routing networks demonstrated by Prucnal[17, 18] and the coincident pulse logic technique in [5] have demonstrated that such demultiplexing can take place in the optical domain. This is accomplished by timing the arrival of a select data pulse within a pulse train and a reference timing pulse such that the two are uniquely coincident in both time and space at a particular detector. In this context it is possible to reexamine several aspects of multiprocessor design and find new solutions to classical problems such as interconnection complexity and parallel memory addressing.

Our presentation is organized as follows. In Section 2, we present a simple linear coincident pulse logic structure. In Section 3, we show the use of the linear structure in several multiprocessor interconnection applications. In Section 4, we describe a two dimensional coincident structure suitable for multi-port memory applications. Finally, in Section 5, we give concluding remarks.

2. Coincident Pulse Logic

In this section, we introduce the concept of coincident pulse logic with an example of pulse delay addressing in one-dimensional arrays. The array shown in Figure 1 is composed of n cells $C_1, \cdots, C_n$. Each cell C_k is uniquely addressable with an electronic pulse at the output of the photodetector D_k. The photodetector generates a voltage proportional the sum of the two incident optical signals, denoted in Figure 1 by s_1 and s_2.

The signals s_1 and s_2 travel in opposite directions along an optical path. Photodetectors are placed at fixed distance intervals d along the optical path, and two laser diodes, L_1 and L_2, are coupled to each end.

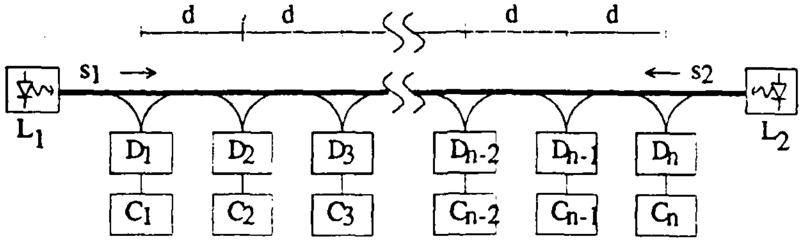


Figure 1: A Linear Structure

Assume that two pulses of duration τ are transmitted, one from L_1 and the other from L_2 at times t_1 and t_2, respectively. These pulses propagate at speed c_g (the speed of light in the waveguide). By carefully selecting the delay between t_1 and t_2 the pulses can be made to meet at exactly one detector. The distance d between any two detectors is chosen to be equal to $d = \tau c_g$, the propagation distance corresponding to the pulse width. The delay $t_1 - t_2$ is also chosen such that it is an even multiple of d. More specifically, if

$$t_1 - t_2 = (n - 1 - 2(k-1))\,\tau \qquad (1)$$

then, the two pulses will meet at detector D_k, thus addressing cell k. For example, when n=5, if L_2 generates its pulse 2τ seconds before L_1 generates its pulse, then (1) gives k=2, that is the two pulses meet at D_2. Similarly, if L_2 generates its pulse 2τ seconds after L_1 generates its pulse, then the two pulses meet at D_4. Clearly, the middle cell is chosen by generating the two pulses simultaneously, that is having $t_1 = t_2$. Therefore, the address of the cell is encoded using the delay $t_1 - t_2$. In this view, the pulse generated by L_1 is treated as the reference pulse and the pulse generated by L_2 becomes a select pulse. In the remaining discussion, the names t_{ref}, L_{ref}, t_{sel}, and L_{sel}, will refer to t_1, L_1, t_2, and L_2 respectively.

The worst case selection time is determined by the maximum delay needed to address any cell in the array. From (1), it is clear that for $k=1, \cdots, n$, we have

$$-(n-1)\,\tau \le t_{ref} - t_{sel} \le (n-1)\,\tau \qquad (2a)$$

from which we find that the worst case latency σ, is given by

$$\sigma = 2\,n\,\tau \qquad (2b)$$

Note that equation (2a) indicates that the select pulse occurs within $n\,\tau$ before or after the reference pulse. Within time σ it is possible to address more than one cell by sending a series of pulses from L_{sel}, one for each selection. Each of these pulses will intersect with the reference pulse at the desired detector. In other words, parallel selections are positionally distinguishable in a pulse train generated by a series of select pulses.

The key to the coincident pulse technique is the appropriate temporal positioning of the pulses in the select pulse train. Figure 2a shows a simple mechanism for low bandwidth applications which allow for direct electronic modulation of a single optical source. In this mechanism a shift register is loaded with a select bit pattern and the output is clocked into an optical pulse train.

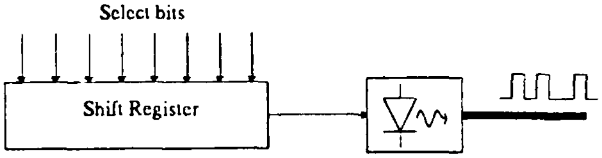


Figure 2a: Single Source Pulse Train Generator

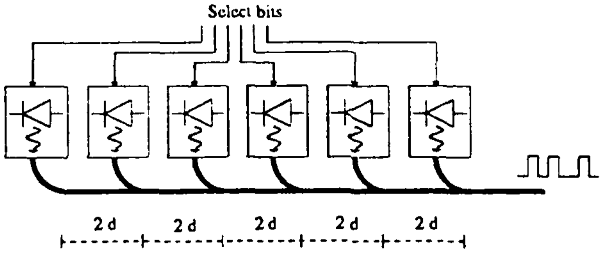


Figure 2b: Linear Coupled Pulse Train Generator

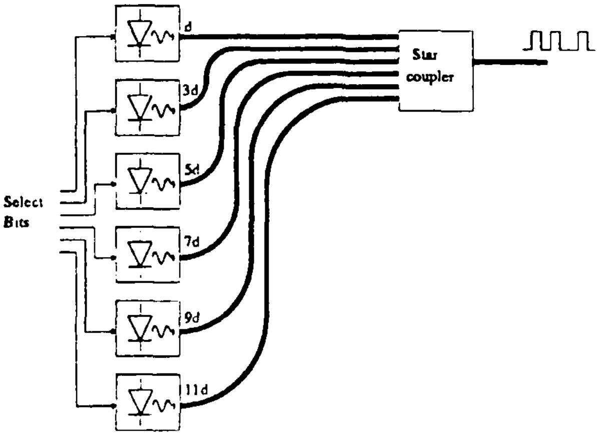


Figure 2c: Star Coupled Pulse Train Generator

Alternatively, in higher bandwidth applications spatial separation may be used to control the optical path length of various pulses that are multiplexed on a single fiber to create a pulse train. Two possible delay line structures for generating the optical pulse train containing the select pulses are shown in Figure 2b and Figure 2c. In Figure 2b, n laser diode pulsers are spaced at incremental distances $2d$ along an optical delay line. In Figure 2c, a star coupler joins the output of n laser diode pulsers at incremental path lengths $2d$. In order to reconcile the optical to electronic bandwidth difference, a single edge in the electronic time-base controls the activation of all the pulsers such that all the optical pulses are generated simultaneously. If the duration of each optical pulse is equal to τ, then the select pulse train will be confined to $2n$ time slots, each with duration τ.

The above address generation scheme is crucially dependent on the simultaneous pulsing of several laser diodes. Such synchronization problems can be avoided by using a single optical pulse source of duration τ, and a series of electro-optic switches. Each laser diode is replaced by an electro-optic switch which "couples in" the pulse at various optical path lengths to create a pulse train. However, this solution trades the synchronization problem for a power distribution problem[14].

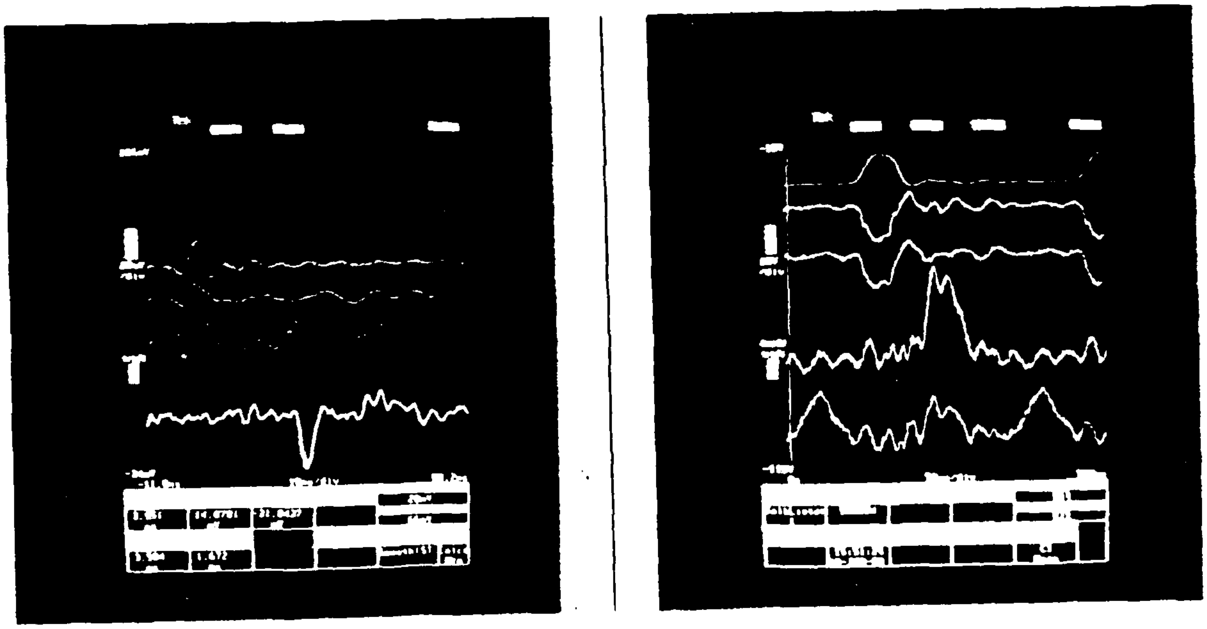


Plate 1 - Coincidence traces (a) positive pulses, (b) negative pulses

Another key component to the success of coincident pulse structures is the ability to discriminate between coincident pulse . nd non-coincident pulse waveforms at the output of a photodetector. In Plates 1a and 1b, we show typical waveforms generated by a coincident pulse structure which was tested using both positive and negative pulse inputs. The top trace of each of the two displays is a pulse edge used as the system trigger. The second and third traces are electronic signals used to modulate the laser diodes, each biased near threshold. The bottom two traces are stored waveforms from the results of two separate experiments. In the first experiment the optical path length from the first laser diode was twice the path length from the second laser diode. The resulting trace clearly shows the detector output with each pulse separated in time proportional to the difference in the optical path length. In the second experiment the optical path length from both laser diodes is equal. Here, the trace shows a single coincident pulse of amplitude sufficient to be easily discriminated from the two separate pulses shown in the previous experiment. Since it is apparent from this experiment that both positive and negative pulse coincidence can be easily discriminated, the choice between the two appears to be a trade-off between shot noise in the detector versus output stability and power consumption in the optical pulsers.

By combining the pulser and and detector structures illustrated in Figures 1 and 2 we can provide a general purpose mechanism for m out of n selection. These structures can be used for constructing a variety optical interconnection systems and permutation networks for interconnecting processors in a computational system, as shown in the next section. Alternatively, as discussed in Section 4, by placing memory at each detector cell, the structures can be used as parallel memory.

3. Application to Multiprocessor Interconnections

Table 1 is a classification of communication structures based on varying levels of connectivity and capabilities for a set of transmitting processors. A multicasting structure is one in which a transmitter sends a single message to a specific subset of m receivers where $m \leq n$, the number of processors in the system. Unlike broadcasting where all receivers actively interpret every message, multicasting provides that only the intended receivers interpret the message. This requires that some of the work in interpreting a message destination is done by the communications subsystem rather than by using resources in unintended receivers. Simulcasting, by our definition, is the concurrent transmission of n unique messages by a single transmitter to each of the n receiving processors. Multicasting and simulcasting may be generalized to the case where n transmitters are each multicasting (or simulcasting) concurrently. We refer to these cases as n*−way* multicasting and n*−way* simulcasting, respectively.

The point to point, broadcast, and completely interconnected structures have been implemented with different degrees of success in multiprocessor systems. Multicasting, simulcasting, and the n-way structures, have not been extensively examined because of the hardware complexity of their realization. In this section we outline several specific applications of coincident pulse techniques to realize multicasting and simulcasting using

Number of Senders	Number of Receivers per Sender	Message Type per Sender	Comments
1	1		point to point
1	m	same msg.	multicast
1	n	same msg.	broadcast
1	n	different msgs.	simulcast
n	1		permutation/complete
n	m	same msg.	n-way multicast
n	n	same msg.	n-way broadcast
n	n	different msgs.	n-way simulcast

Table 1, Interconnection Structures for n Communicating Processors (m less than n)

the linear structure of the previous section.

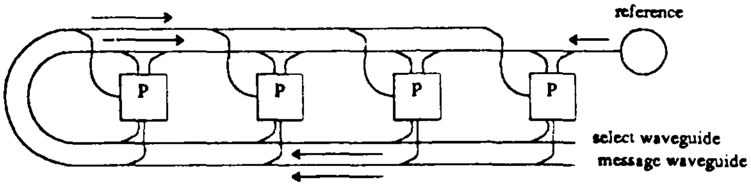


Figure 3 - 1-dimensional Multicasting Interconnection

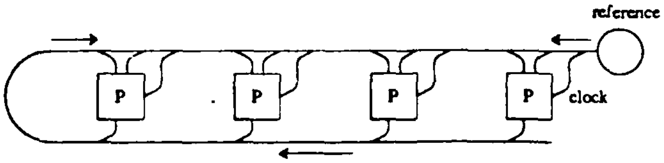


Figure 4 - 1-dimensional Simulcasting Interconnection

Figure 3 is an example of a 1 to *m* multicasting structure. This figure shows a bus interconnected multiprocessor with separate optical interconnections for address and data. The unique feature of this structure is the use of coincident pulse techniques in the implementation of the address bus. In each cycle, one transmitting processor places on the select waveguide a positionally encoded set of destination address bits followed by an *n*-bit message on the message waveguide. Simultaneously, a reference pulse propagates in the opposite direction in the select waveguide and coincides with the the destination

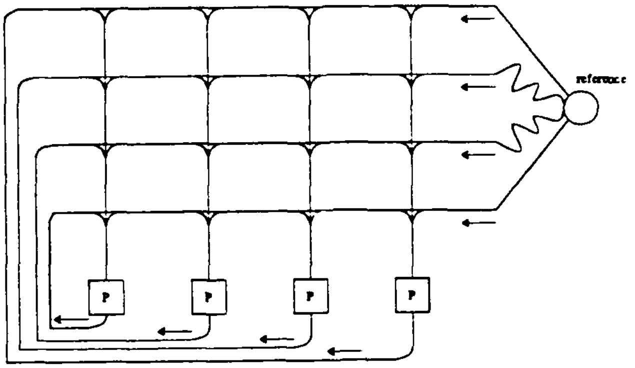


Figure 5 - 1-dimensional Array Simulcasting Interconnection

address bits at each of the destination processors. Thus as the data propagates through the message waveguide, it is read only by those processors for which coincident pulse selections have been made.

A simple modification of the previous example results in the 1 to n simulcasting structure of Figure 4. In this case, we have deleted the message waveguide and changed the interpretation of the address waveguide. Specifically, the select pulses in the address waveguide are now considered one bit data messages. The destination address of each message is positionally encoded by the relative position of the data bit in the select pulse train. Thus, a "one" is transmitted as a select pulse and a "zero" is the absence of a select pulse in the position corresponding to that receiver. The entire array is globally clocked to provide a strobe which moves the select information into a data latch at each receiver. This may be implemented using a separate copy of the reference pulse as a clock input, tapped as shown in Figure 4, with a small internal delay to allow for electronic propagation in the data latch. In both of these structures, we assume that a control arbitration mechanism exists [4] such that only one processor is allowed to transmit in a given cycle.

We can extend both multicasting and simulcasting to n-way structures. One method, is to allow all transmitters to transmit at the same time, and ensure that the optical path length between adjacent transmitters is greater than the length of the select pulse train, D, where $D \geq 2nd$, and d is the optical path length between any two adjacent receivers. For multicasting, this restricts the length of the message to be less than D. A second method for extending multicasting removes this restriction at the expense of more complex bus arbitration hardware. As discussed in [4] a control mechanism can be implemented such that a group of messages from an arbitrary subset of transmitters can be pipelined onto the data bus in a single cycle.

A second method to extend the 1 to n simulcasting structure of Figure 4 to an n-way simulcasting structure, is shown in Figure 5. In this example, an array of select waveguides connects individual busses attached to each processor. The unique feature of this structure is that the coincidence points are no longer detectors, but rather passive couplers which merge the coincident pulses into the receiving bus for each respective processor. Selection within each row of the array operates as in the previous structure relative to the transmitting processor attached to that row. The receiving busses are arranged in columns, perpendicular to the direction of propagation of the reference pulses in the transmitting busses. Therefore, the reference pulses arrive at all sites along a receiving bus simultaneously. The resulting data pulse train on the receiving bus is thus formed by coupling-in the message bits at specific optical path distances corresponding to the vertical separation of selection points. Each receiving bus thus contains an n-bit pulse train consisting of one bit from each of the transmitting processors. An advantage of this structure is that there is no need for any arbitration. Only a simple clocking mechanism is needed to delimit bus cycles.

4. Two Dimensional Arrays

By generalizing the propagation of pulses in one dimension to the propagation of linear wavefronts moving through a series of parallel waveguides, we can construct two dimensional structures. Hence, the method of addressing a location by programming the intersection of pulses may be generalized to addressing a location in a two-dimensional array by programming the intersection of wavefronts. In this section, we present a simple propagation scheme which may be used in 2-dimension selection.

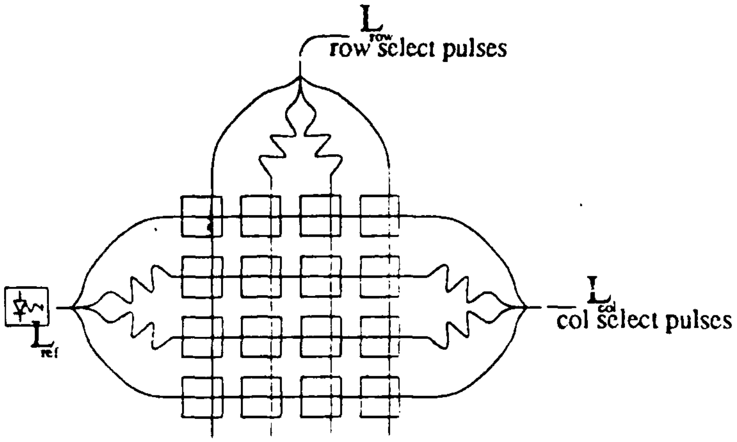


Figure 6: A Two Dimensional Structure

Consider 2-dimensional arrays similar to the one shown in Figure 6. An array of size n is composed of $\sqrt{n} \times \sqrt{n}$ cells separated by a distance $d = \tau \, c_g$ in both the vertical and the horizontal directions. The coincidence mechanism is the same as the linear example, except that we now require coincidence of three optical signals. Specifically, a reference wavefront generated by the reference diode L_{ref}, a select pulse train L_{col}, each traveling horizontally and in opposite directions, and another select pulse train, L_{row}, traveling vertically.

The optical signal generated by each source is decoupled from the source fiber by a star connection into $\sqrt{n}$ signals that travel through the array in parallel waveguides. Since the optical path length of all legs in the star will be equal, the wavefront will arrive at all locations in a single row (or column) simultaneously. For example, an optical pulse generated by L_{ref} and directed horizontally through the array will simultaneously arrive at all locations in column j. Similarly, any pulse generated by L_{col}, will also arrive, simultaneously, at all the cells in column j, and any pulse generated by L_{row} will arrive, simultaneously, at all the cells in row i.

In order to derive the equations that govern the intersections of three wavefronts, assume, as in the case of the linear array, that L_{ref} generates a pulse of duration τ at time t_{ref}, and that L_{col} and L_{row} generate pulses at times t_{col} and t_{row}, respectively. If the timing of L_{col} is such that

$$t_{ref} - t_{col} = (\sqrt{n} - 1 - 2(j-1))\,\tau \tag{3}$$

then, the two wavefronts generated by L_{ref} and L_{col} will meet at column j of the array. In order to select a particular cell i,j in that column, the third wavefront, namely the one generated by L_{row}, should be crossing row i when the other two wavefronts meet at column j. This may be accomplished by timing L_{row} such that

$$t_{ref} - t_{row} = (j - i)\,\tau \tag{4}$$

In other words, to address a certain location i,j, the column number j is encoded as $t_{ref}-t_{col}$ and the difference, $j-i$, between the column number and the row number is encoded as $t_{ref}-t_{row}$. From (3) and (4), it may be shown that

$$-(\sqrt{n} - 1)\,\tau \le t_{ref}-t_{col} \le (\sqrt{n}-1)\tau$$

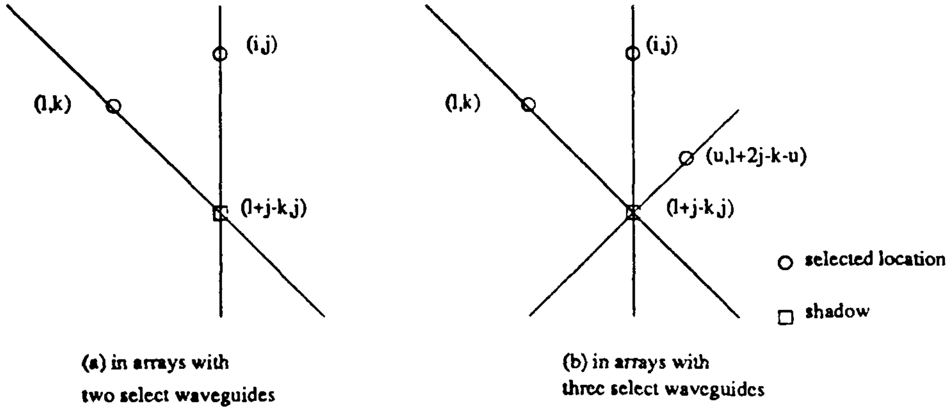
and

$$-(\sqrt{n} - 1)\,\tau \le t_{ref}-t_{row} \le (\sqrt{n} - 1)\,\tau$$

and hence, the latency time, σ, is

$$\sigma = 2\sqrt{n}\;\tau \tag{5}$$

Using the above scheme, it is possible to encode the addresses of all of the n_c cells in the column and row pulse trains during a single cycle. In the one-dimensional case, the cycle time was directly proportional to the size of the array. This was because each cycle needed to provide an optical time-base slot for each location. In the two-dimensional case, cycle time is proportional to the square root of the number of cells. The price paid for this reduction in cycle time is the potential for overlap in parallel selection. This results from a requirement that corresponding select bits in each of the select waveguides be uniquely paired such that only the coincidence of paired bits are considered to be appropriate selections. Coincidences occurring from the intersections of non-paired bits will be referred to as *shadows*.



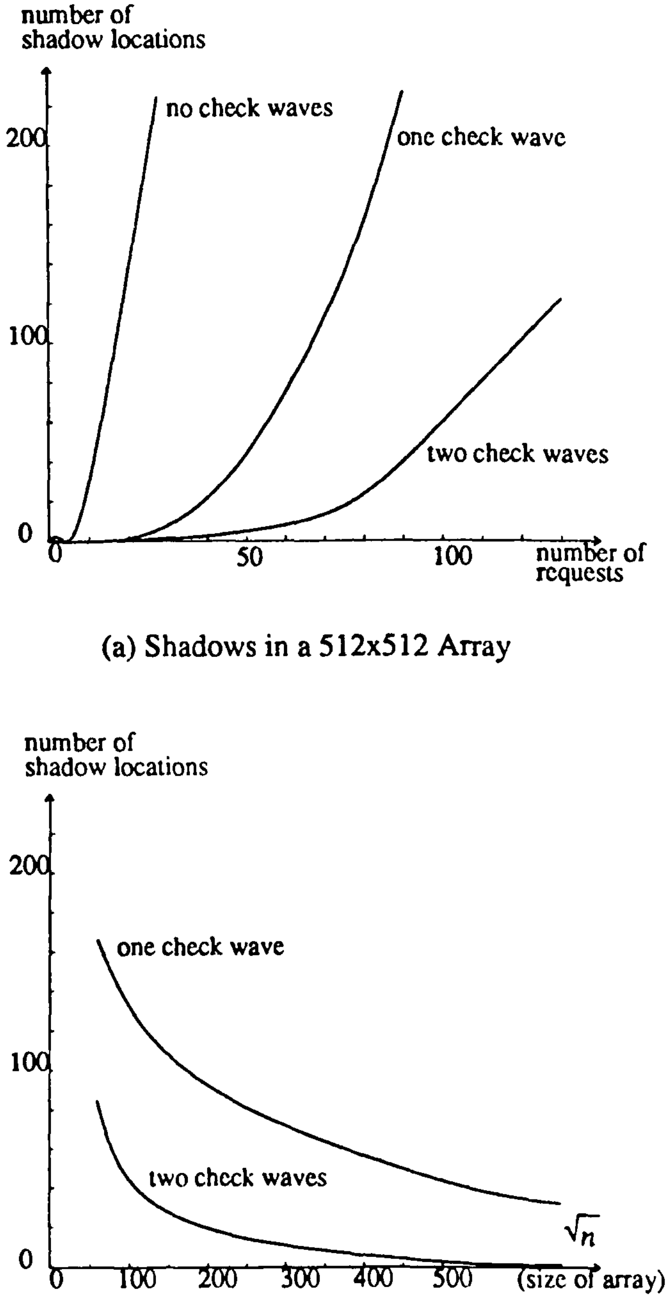
(a) in arrays with two select waveguides

(b) in arrays with three select waveguides

Figure 7: Wavefront Intersections to Cause Shadow Selections

For example, if the two cells (i,j) and (l,k) are selected during the same bus cycle, then a shadow will appear at cell $(l+j-k,j)$ as shown in Figure 7a. This is because the selection of position j in the column select train causes a coincidence with the reference wavefront at every cell along that column. We will refer to this partial coincidence pattern as a *trace*. Similarly, the selection of bit $k-l$ in the row select pulse train will cause a partial coincidence trace with the reference wavefront along the diagonal passing through cell (l,k) as shown in Figure 7a. Therefore, cell $(l+j-k,j)$ which resides at the intersection of these two traces will see a coincidence of the reference wavefront with each of the select wavefronts, and hence, will be falsely selected.

Since shadows occur at the intersections of two traces corresponding to the vertical and horizontal select wavefronts, we can reduce the number of such intersections by the addition of a third select wavefront, referred to as the check wavefront. With this, a valid selection occurs on the coincidence of the reference wavefront with select bits in each of the three wavefronts. In this case, shadows are generated at the intersection of the two traces mentioned and the additional trace corresponding to the check wavefront in Figure 7b. As shown by the simulation results in Figure 8a, this greatly reduces the number of shadows generated in the array. Using the same argument, by the addition of a fourth wavefront, a second check wavefront, it is possible to reduce even further the occurrence of shadows.



(a) Shadows in a 512x512 Array

(b) Shadows Generated by 50 Requests

Figure 8: Simulation Results for the Incidence of Shadows

In the above scheme the select and check pulse trains are each of length $2\sqrt{n}$. Thus, the total number of bits transmitted to the array in a single cycle is of order $O(\sqrt{n})$. However, the following proposition shows that the number of bits required to distinguish a unique set of parallel selections is n.

Proposition: *The minimum number of bits required to uniquely select an arbitrary set of cells from a collection of size n is n.*

Proof: For a set of cells S of size n, the size of the power set $P(S)$ is 2^n. Using a binary encoding, the enumeration of $P(S)$ requires $\log_2 2^n = n$ bits per code word. Assume that a binary encoding scheme exists which can address all possible subsets of S using a code of length less than n-bits. Such a scheme would result in an enumeration of less than 2^n subsets. Therefore there must exist in that encoding at least one code which corresponds to more than one element in $P(S)$. Hence, such a system does not uniquely address all subsets. []

This result introduces a restriction on the application of 2-dimensional structures. If we are to use n bits as the proposition implies, we must either transmit n bits to the array serially, in which case latency is comparable to the 1-dimensional case, or we must add additional waveguides thus increasing the hardware complexity.

Alternatively, we may restrict the number of concurrent accesses to some number m such that $m \ll n$. As shown in Figure 8 the incidence of shadow selections is dependent on the size of the array relative to the number of requests and the number of wavefronts in the selection structure. Therefore, 2-dimensional arrays are most appropriate in computational structures where the number of potential receiving sites is much greater than the number of transmitting sites. This would be the case in the design of an m-ported memory of size n. For example, simulation results show that a 256k memory implemented in a $512x512$ square array with two select and two check wavefronts can be operated as a 50 ported memory with an average of 4.6 shadow locations for 50 simultaneous requests. In a memory application these few shadows would appear as extra read requests which should be discarded. The problem of shadows for write requests does not exist when we restrict writes to a single request per cycle. This results in a concurrent-read-exclusive-write model for shared memory multiprocessors.

In applications where the restriction $m \ll n$ cannot be met, a structure can be provided which eliminates shadows by the transmission of $O(n)$ bits of selection information into the array. In .his structure, shown in Figure 9, the $\sqrt{n}$ row select waveguides are kept distinct. On each waveguide a unique pulse train of $2\sqrt{n}$ bits is transmitted. Each train contains only the select information for access on that row. There is no longer a need for the vertical waveguides. The,row waveguides share a common set of reference pulses generated as in the previous 2-dimensional example. Thus, each row pulse train will have all the information (and only the information) for selections on that row. In effect each row is an independent linear structure of size $\sqrt{n}$.

One structure for the generation of the row pulse trains is shown in Figure 10. In this Figure a set of m transmitters are each connected to a linear structure where the coincident points on those structures are optical repeaters, which detect pulse coincidence and re-transmit into the row select pulse trains. Unlike the linear structure of Figure 1, we allow only a single pulse to travel in each direction through the structure. In addition, we vary the timing of both pulses in order to achieve both the appropriate row location and relative timing of the pulse coincidence.

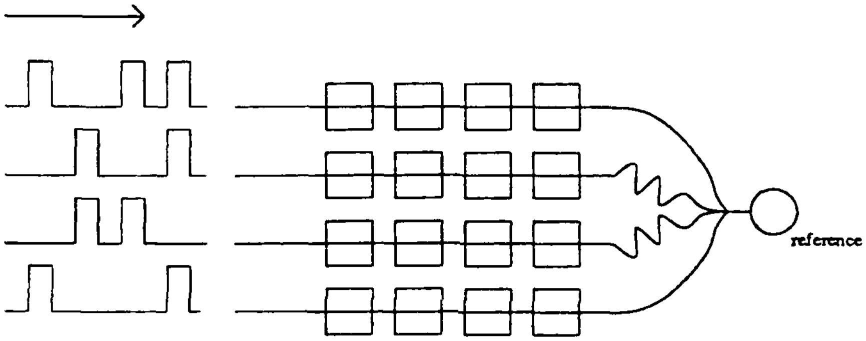


Figure 9: Parallel Access 2-dimensional Selection Structure

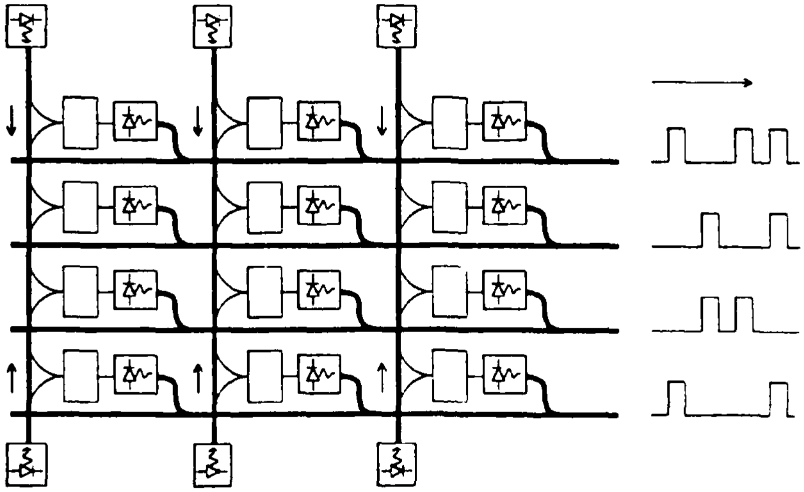


Figure 10: Parallel Address Encoder

In order to derive the equations governing the relative timing of the pulses generated at a transmitter k, $1 \le k \le m$, for the selection of location i,j (see Figure 11), assume that the

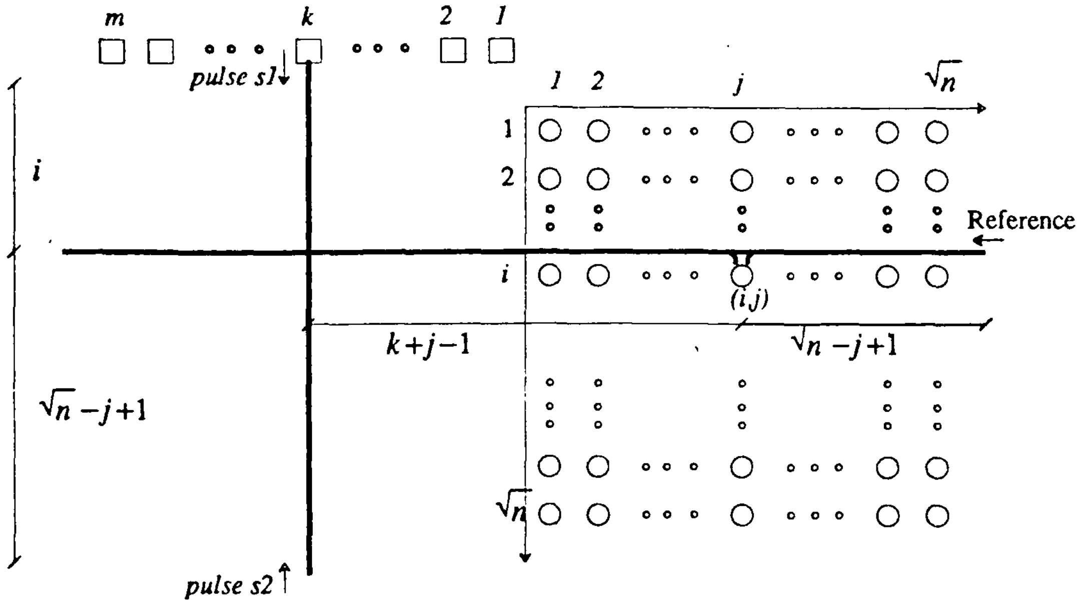


Figure 11: Pulse Path Lengths for the Memory Structure of Figures 9 and 10

reference pulse is fired at time t_{ref} and the two select pulses are fired at time t_{s1} and t_{s2}. Given that the reference pulse will be at location i,j at time $t_{ref}+(\sqrt{n}-j+1)\tau$, the two select pulses should be at that location at the same time. That is:

$$t_{s1}+(i+k+j-1)\,\tau+\tau_{rep}=t_{ref}+(\sqrt{n}-j+1)\tau$$

$$t_{s2}+(\sqrt{n}-i+k+j)\,\tau+\tau_{rep}=t_{ref}+(\sqrt{n}-j+1)\tau$$

Where, τ_{rep} is the delay introduced by the optical repeater circuit shown in Figure 10. Therefore, relative to the reference pulse, for selecting an address i,j, the select pulses must be fired at times:

$$t_{s1}=t_{ref}+(\sqrt{n}-2j+2-i-k)\,\tau-\tau_{rep}$$

$$t_{s2}=t_{ref}+(i-k-2j+1)\,\tau-\tau_{rep}$$

In this manner up to m selections, one per processor, can be made simultaneously to a memory of n cells. The worst case latency is $(3\sqrt{n}+m)\tau+\tau_{rep}$, which is the time for a pulse to travel from a transmitter to a selected cell. However, using pipelining, new selections can be generated with a cycle time of $2\sqrt{n}\,\tau$.

5. Concluding Remarks

In this paper, we have presented several structures based on coincident pulse techniques for multiprocessor interconnections which support multicasting, simulcasting and multiported memory based communications. The multicasting structure is capable of supporting transmission of one message from a single sender to a subset m of n receivers with cycle time $2n\tau$. The simulcasting structure is capable of supporting the transmission of n different one bit messages, concurrently, from a single transmitter to n receivers with the same cycle time. m-way versions of these structures allow concurrent transmissions from m transmitters in a single cycle of length $2n\tau$ for simulcasting and worst case length (where the number of transmitters $m=n$) $2n^2\tau$ for multicasting. However using arbitration techniques a subset of m transmitters may operate concurrently and the cycle length will be reduced to $2nm\tau$.

Two dimensional structures are appropriate for application as m-ported memory and have been presented in two versions. The first version, a concurrent-read-exclusive-write system, where $m \ll n$, has a selection latency equal to $2\sqrt{n}\,\tau$. The second version, which removes any restriction on m, has a latency of $(3\sqrt{n}+m)\tau+\tau_{rep}$ and a cycle time of $2\sqrt{n}\,\tau$.

Based on these results we foresee that *coincident pulse techniques* will have a significant role in the design of hybrid optical-electronic multiprocessors. Moreover, the ability of these systems to support multicasting, simulcasting, and multi-ported memory based communications will have a substantial impact on the performance of fundamental parallel algorithms.

References

1. "Special Issue on Interconnection Networks," *IEEE Computer*, June, 1987.

2. J. L. Baer, *Computer Systems Architectures,* Computer Science Press, Potomac, MD, 1980.

3. J. Bristow, A. Juha, and C. Sullivan, "Optical Implementation of Interconnection Networks for Massively Parallel Architectures," *Technical Digest, Optical Computing Conference*, vol. 9, Optical Society of America, 1989.

4. D. Chiarulli, S. Levitan, and R. Melhem, "Asynchronous Control of Optical Busses in Closely Coupled Distributed Systems," *Journal of Parallel and Distributed Computing (to appear).*

5. D. Chiarulli, R. Melhem, and S. Levitan, "Using Coincident Optical Pulses for Parallel Memory Addressing," *IEEE Computer*, vol. 20, no. 12, pp. 48-58, 1987.

6. P. H. Enslow, *Multiprocessors and Parallel Processing*, Wiley-Interscience, New York, 1974.

7. T. Y. Feng, "A Survey of Inteconnection Networks," *IEEE Computer*, pp. 12-27, December, 1981.

8. A. Hartmann and S. Redfield, "Design Sketches for Optical Crossbar Switches Intended for Large-Scale Parallel Processing Applications," *Optical Engineering*, vol. 28, no. 4, April 1989.

9. P. R. Haugen, S. Rychnovsky, and A. Husain, "Optical Interconnects for High Speed Computing," *Optical Engineering*, vol. 25, no. 10, October 1986.

10. J. Hecht, "Bell Labs Transmits 8 Gbits/s Over 68Km," *Lasers and Applications*, May, 1986.

11. H. Jordan, "A Bit Serial Optical Computer," *Technical Digest, Optical Computing Conference*, vol. 9, Optical Society of America, 1987.

12. R. Melhem, D. Chiarulli, and S. Levitan, "Space Multiplexing of Waveguides in Optically Interconnected Multiprocessor Systems," *The Computer Journal (to appear)*.

13. Steven P. Levitan, "Measuring Communication Structures in Parallel Architectures and Algorithms," in *The Characteristics of Parallel Algorithms*, ed. L. Jamieson, E D. Gannon, R. Douglass, pp. 101-137, MIT Press, Cambridge, MA, 1987.

14. M. Nassehi, F. Tobagi, and M. Marhic, "Fiber Optic Configurations for Local Area Networks," *IEEE Journal on Selected Areas in Communications*, vol. SAC-3, no. 6, pp. 941-949, Nov. 1985.

15. M. Nishio and S. Suzuki et. al., "An Experiment on Photonic Wavelength-Division and Time-Division Hybrid Switching," *Technical Digest, Photonic Switching Conference*, Optical Society of America, 1989.

16. A. Perrier, P. Prucnal, M. Chbat, and Self-Clocked Optical Time Slot Interchanger, *Technical Digest, Photonic Switching Conference*, Optical Society of America, 1989.

17. P. R. Prucnal and D. J. Blumenthal, "12.5 Gbits Fibre-Optic Network using all Optical Processing," *Electronics Letters*, vol. 23, no. 12, June 4, 1987.

18. P. R. Prucnal, D. J. Blumenthal, and P. Perrier, "Self Routine Photonic Swithing Demonstration with Optical Control," *Optical Engineering*, vol. 26, no. 5, May 1987.

19. S. V. Ramanan and H. F. Jordan, "Photonic Architectures for Performing Perfect Shuffle on a Time-Division Multiplexed Signal," OCS Technical Report 89-03, University of Colorado, Boulder, CO.

20. A. A. Sawchuk, B. K. Jenkins, C. S. Raghavendra, and A. Varma, "Optical Crossbar Networks," *IEEE Computer*, p. 50, June, 1987.

21. Y. Sheng and H. H. Arsenault, "Light Effective Perfect Shuffle Using Fresnel Mirrors," *Technical Digest, Optical Computing Conference*, vol. 9, Optical Society of America, 1989.

22. T. Shimoe, S. Kuroyanagi, K. Murakami, H. Rokugawa, N. Mekada, and T. Odagawa, "Expirimental 512 Mbits/s Time Division Photonic Switching System," *Technical Digest, Photonic Switching Conference*, Optical Society of America, 1989.

23. H. J. Siegel, *Interconnection Networks for Large-Scale Parallel Processing: Theory and Case Studies*, Lexington Books, Lexington, MA, 1984.

24. D. W. Smith, P. Healey, and S. A. Cassidy, "Extendible Optical Interconnection Network," *Technical Digest, Photonic Switching Conference*, Optical Society of America, 1989.

25. B. Sugla, "Computing on a Digital Optical Computer Using Regular Interconnections," Technical Report Number 963-0100, AT&T Bell Laboratories, Holmdel, NJ, 1988.

26. S. Suzuki and K. Nagashima, "Optical Broadband Communications Network Architecture Utilizing Wavelength-Division Switching Technologies," *Technical Digest, Photonic Switching Conference*, vol. 13, Optical Society of America, 1987.

27. Thompson, "Optimizing Photonic Variable-Integer-Delay Circuits," *Technical Digest, Photonic Switching Conference*, vol. 13, Optical Society of America, 1987.

28. R. S. Tucker and S. K. Korotky et. al., "4 Gb/s Optical Time-Division Multiplexed System Expiriments using $Ti:LiNbO_3$ Switch/Modulators," *Technical Digest, Photonic Switching Conference*, vol. 13, Optical Society of America, 1987.