Superposition in Optical Computing

B. Keith Jenkins
Signal and Image Processing Institute MC-0272
University of Southern California, Los Angeles, California 90089-0272

and

C. Lee Giles
Air Force Office of Scientific Research/NE
Bolling AFB, D.C. 20332-6448

ABSTRACT

The design of an optical computer must be based on the characteristics of optics and optical technology, and not on those of electronic technology. The property of optical superposition is considered and the implications it has in the design of computing systems is discussed. It can be exploited in the implementation of optical gates, interconnections, and shared memory.

INTRODUCTION

Fundamental differences in the properties of electrons and photons provide for expected differences in computational systems based on these elements. Some, such as the relative ease with which optics can implement regular, massively parallel interconnections are well known. In this paper we examine how the property of superposition of optical signals in a linear medium can be exploited in building an optical or hybrid optical/electronic computer. This property enables many optical signals to pass through the same point in space at the same time without causing mutual interference or crosstalk. Since electrons do not have this property, this helps to shed more light on the role that optics could play in computing. We will separately consider the use of this property in interconnections, gates, and memory.

INTERCONNECTIONS

A technique for implementing optical interconnections from one 2-D array to another (or within the same array) has been described [Jenkins et al, 1984]. It utilizes two holograms in succession (Fig. 1). The holograms can be generated by a computer plotting device. The idea is to define a finite number, M, of distinct interconnection patterns, and then assemble the interconnecting network using only these M patterns. The second hologram of Fig. 1 consists of an array of facets, one for each of the M interconnection patterns. The first hologram contains one facet for each input node, and serves to address the appropriate patterns in the second hologram.

It is the superposition property that makes this interesting. Note that many different signal beams can pass through the same facet of the second hologram at the same time without causing mutual interference. (All of these signals merely get shifted in the same direction and by the same amount.) This feature decreases the complexity of both holograms -- The first because it only has to address M facets, the second hologram because it only has M facets. Let N be the number of nodes in the input and output arrays. The complexity (number of resolvable spots) of each hologram can be shown to be proportional to NM, with the proportionality constant being approximately 25 [Jenkins et al., 1984].

Using this as a model for interconnections in parallel computing, a comparison can be made between the complexity of these optical interconnections with those of electronic VLSI for various interconnection networks. Results of this have been given in [Giles and Jenkins, 1986]. It is found that in general the optical interconnections have an equal or lower space complexity than electronic interconnections, with the difference becoming more pronounced as the connectivity increases.

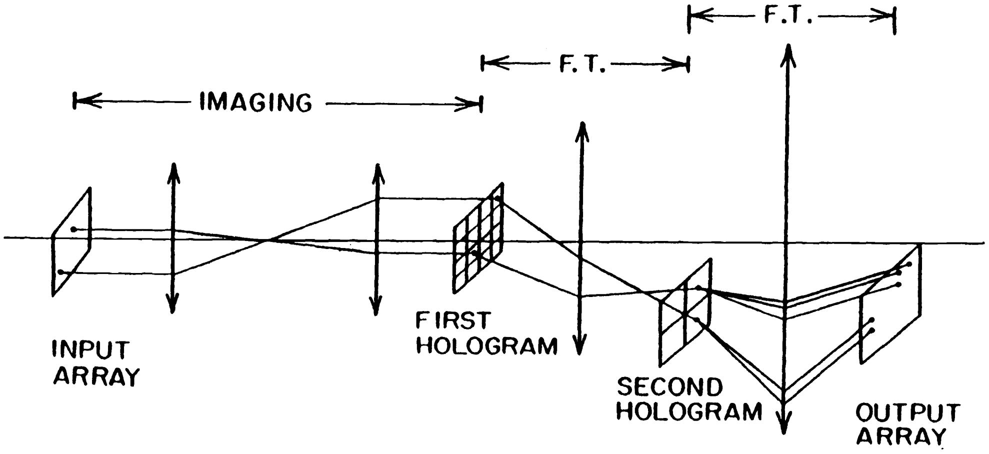

Fig. 1. Optical holographic system for interconnections.

SHARED MEMORY

The same superposition principle can be applied to memory cells, where many optical beams can read the same memory location simultaneously. This concept could be useful in building a parallel shared memory machine.

For this concept, we first consider abstract models of parallel computation based on shared memories. The reason for this approach is to abstract out inherent limitations of electronic technology (such as limited interconnection capability); in designing an architecture one would adapt the abstract model to the limitations of optical systems. These shared memory models are basically a parallelization of the Random Access Machine.

The Random Access Machine (RAM) model [Aho, Hopcroft, and Ullman, 1974] is a model of sequential computation, similar to but less primitive than the Turing machine. The RAM model is a one-accumulator computer in which the instructions are not allowed to modify themselves. A RAM consists of a read-only input tape, a write-only output tape, a program and a memory. The time on the RAM is bounded above by a polynomial function of time on the TM. The program of a RAM is not stored in memory and is unmodifiable. The RAM instruction set is is small and consists of operations such as store, add, subtract, and jump if greater than zero; indirect addresses are permitted. A common RAM model is the uniform cost one, which assumes that each RAM instruction requires one unit of time and each register one unit of space.

Shared memory models are based on global memories and are differentiated by their accessibility to memory. In Fig. 2 we see a typical shared memory model where individual processing elements (PE's) have variable simultaneous access to an individual memory cell. Each PE can access any cell of the global memory in unit time. In addition, many PE's can access many different cells of the global memory simultaneously. In the models we discuss, each PE is a slightly modified RAM without the input and output tapes, and with a modified instruction set to permit access to the global memory. A separate input for the machine is provided. A given processor can generally not access the local memory of other processors.

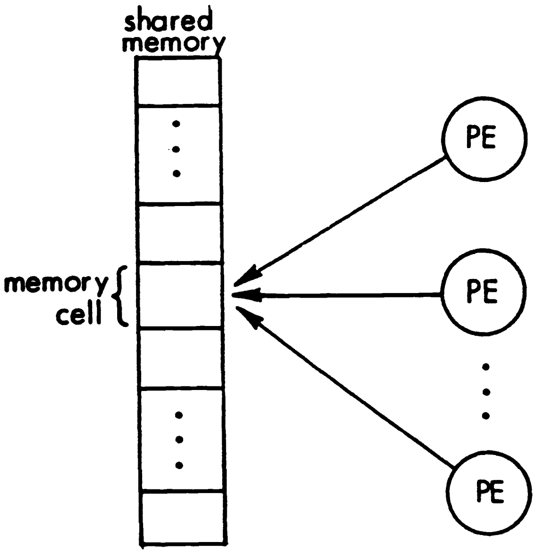

Fig. 2. Conceptual diagram of shared memory models.

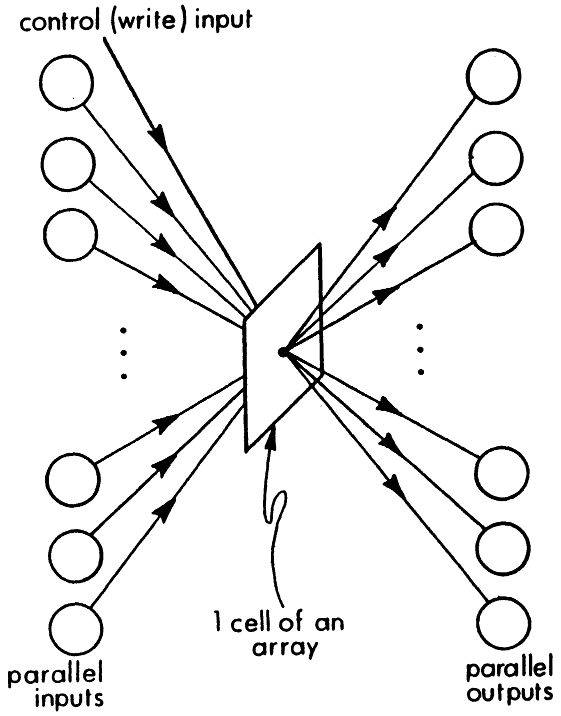

Fig. 3. One memory cell of an array, showing multiple optical beams providing contention-free read access.

The various shared memory models differ primarily in whether they allow simultaneous reads and/or writes to the *same* memory cell. The PRAC, parallel random access computer [Lev, Pippenger and Valiant, 1981] does not allow simultaneous reading or writing to an individual memory cell. The PRAM, parallel random access machine, [Fortune and Wyllie, 1978] permits simultaneous reads but not simultaneous writes to an individual memory cell. The WRAM, parallel write random access machine, denotes a variety of models that permit simultaneous reads and certain writes, but differ in how the write conflicts are resolved. For example, a model by Shiloach and Vishkin (1981) allows a simultaneous write only if all processors are trying to write the same value. The paracomputer [Schwartz, 1980] has simultaneous writes but only "some" of all the information written to the cell is recorded. The models represent a hierarchy of time complexity given by

$$T^{\mathrm{PRAC}} \geq T^{\mathrm{PRAM}} \geq T^{\mathrm{WRAM}}$$

where T is the minimum number of parallel time steps required to execute an algorithm on each model. More detailed comparisons are dependent on the algorithm [Borodin and Hopcroft, 1985].

In general, none of these shared memory are physically realizable because of actual fan-in limitations. As an electronic example, the ultracomputer [Schwartz, 1980] is an architectural manifestation of the paracomputer that uses a hardwired Omega network between the PE's and memories; it simulates the paracomputer within a time penalty of $O(\log^2 n)$. The current IBM RP3 project is a continuation of the (initial) work on the ultracomputer.

Optical systems could in principle be used to implement this parallel memory read capability. As a simple example, a single 1-bit memory cell can be represented by one pixel of a 1-D or 2-D array; the bit could be represented by the state (opaque or transparent) of the memory cell. Many optical beams can simultaneously read the contents of this memory cell without contention (Fig. 3). In addition to this an interconnection network is needed between the PE's and the memory, that can allow any PE to communicate with any memory cell, preferably in one step, and with no contention. A regular crossbar is not sufficient for this because fan-in to a given memory cell must be allowed. Figure 4 shows a conceptual block diagram of a system based on the PRAM model; here the memory array operates in reflection instead of transmission. The fan-in required of the interconnection network is also depicted in the figure.

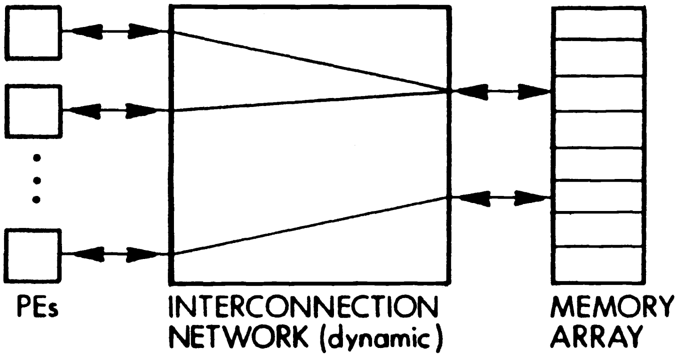

Fig. 4. Block diagram of an optical architecture based on parallel RAM models.

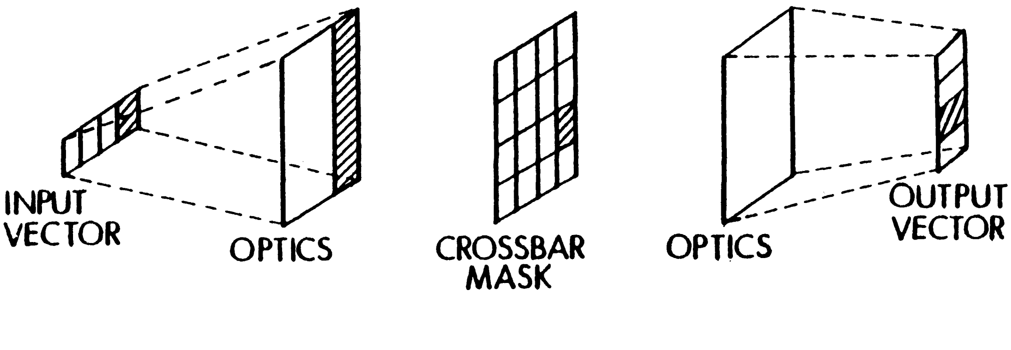

Fig. 5. Example of an optical crossbar interconnection network.

Optical systems can potentially implement crossbars that also allow this fan-in. Several optical crossbar designs discussed in [Sawchuk, et al., 1986] exhibit fan-in capability. An example is the optical crossbar shown schematically in Fig. 5; it is based on earlier work on optical matrix-vector multipliers. The 1-D array on the left could be optical sources (LED's or laser diodes) or just the location of optical signals entering from previous components. An optical system spreads the light from each input source into a vertical column that illuminates the crossbar mask. Following the crossbar mask, a set of optics collects the light transmitted by each row of the mask onto one element of the output array. The states of the pixels in the crossbar mask (transparent or opaque) determine the state of the crossbar switch. Multiple transparent pixels in a column provide fanout; multiple transparent pixels in a row provide fan-in. Many optical reconfigurable network designs are possible, and provide tradeoffs in performance parameters such as bandwidth, reconfiguration time, maximum number of lines, hardware requirements, etc. Unfortunately, most simple optical crossbars will be limited in size to approximately 256 x 256 (Sawchuk, et al., 1986). We are currently considering variants of this technique to increase the number of elements. Possibilities include using a multistage but nonblocking interconnection network (e.g. Clos), a hierarchy of crossbars, and/or a memory hierarchy.

GATES

Since the superposition property of optics only applies in linear media, it cannot in general be used for gates, which of course are inherently nonlinear. However, for important special cases superposition can allow many optical gates to be replaced with one optical switch.

Consider again the situation depicted in Fig. 3, with the aperture being used as a switch or relay. The control beam opens or closes the relay; when the relay is closed (i.e., aperture is transparent), many optical signal beams can independently pass through the relay. If b represents the control beam and a_i the signal beams, this in effect computes $b \cdot a_i$ or $\bar{b} \cdot a_i$, depending on which state of b closes the relay, where $\cdot$ denotes the AND operation (Fig. 6).

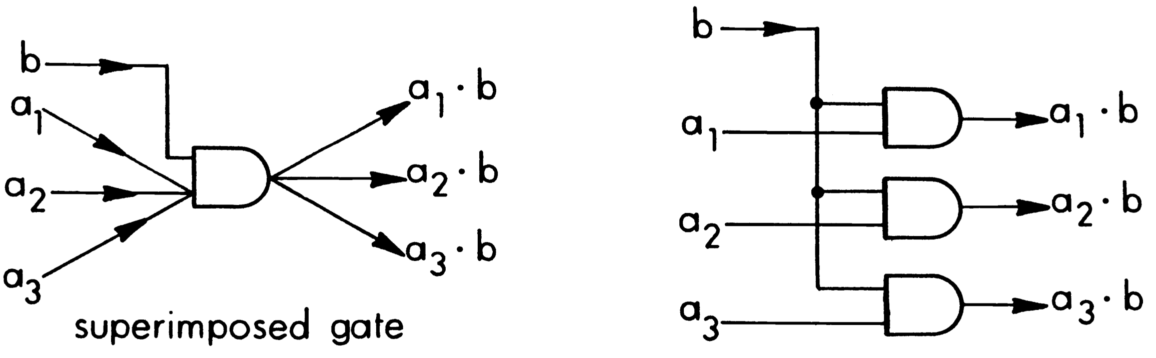

Fig. 6. One optical relay or superimposed gate versus individual gates with a common input.

Using this concept, a set of gates with a common input in a single-instruction multiple-data (SIMD) machine can be replaced with one optical switch or "superimposed gate". An example of this is in the control signals; instead of broadcasting each instruction or control bit to all PE's, a fan-in from all PE's to a common control switch is performed. Thus, for I control bits per instruction word, I superimposed gates could replace NI gates (I per PE). Since for optical or hybrid systems we expect $N \gg I$, this can be a substantial reduction. Fig. 7 shows an example of how this can be incorporated into fixed optical interconnections (such as those of Fig. 1). In the figure there are four PE's laid out on a 2-D array of gates. Each PE sends a signal through one pixel of a transmissive spatial light modulator (SLM). The SLM is electrically addressed, so that the instructions can come from an

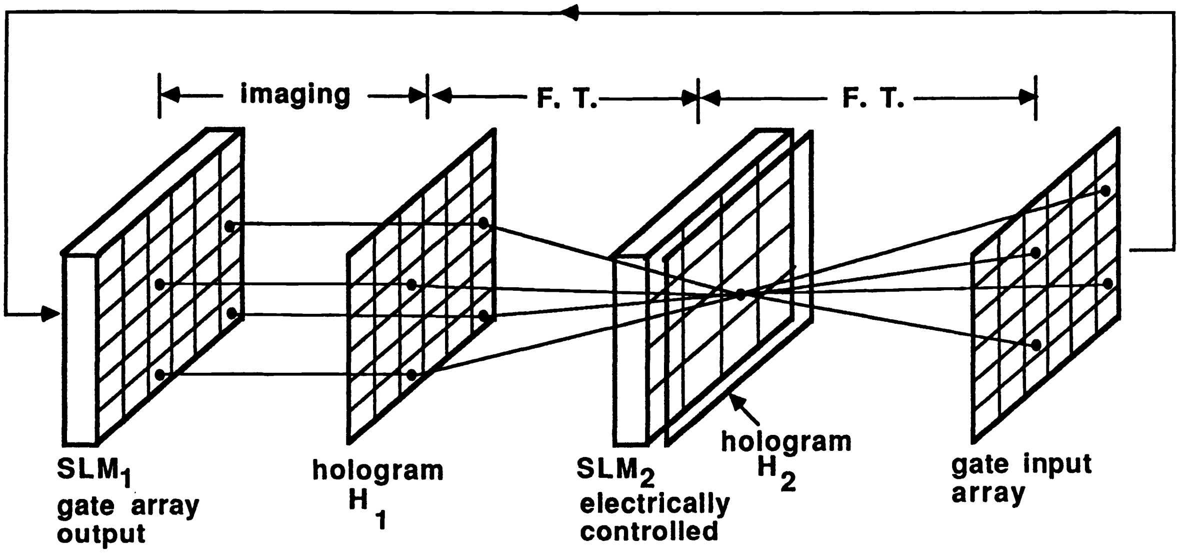

Fig. 7. An optical architecture for the incorporation of superimposed gates for instruction or control bits. The optics are omitted for clarity but are identical to those of Fig. 1. Signals from four gates are shown that fan in to a common control bit.

electronic host. After passing through a common superimposed gate corresponding to the control bit, the signals proceed to the appropriate gate inputs in the gate input array. In this case the second hologram H_2 deflects the signals to the desired gate inputs (gates different from which they came). This optical system is identical to that of Fig. 1 except for the introduction of the SLM for control bits; thus the systems are compatable. Note also that the fanout of each gate in this process is one; a conventional implementation with a large number of PE's would require very high fanout capability or else a tree of gates for each control bit to provide the fanout.

These superimposed gates are not true 3-terminal devices. The common (b) input is regenerated, but the a_i inputs are not. As a result, a design constraint, that these a_i signals do not go through too many superimposed gates in succession without being regenerated by a conventional gate, must be adhered to. This is typically not an issue in the case of control bits. Another consequence is that the total switching energy required for a given processing operation is reduced, because N gates are replaced with one superimposed gate. This is important because it is likely that the total switching energy will ultimately be the limiting factor on the switching speed and number of gates in an optical computer. Other advantages include an increase in computing speed since some of the gates are effectively passive and reduced requirements on the device used to implement the optical gates.

CONCLUSIONS

We have shown that the property of superposition can be exploited in the design of optical or hybrid optical/electronic computing architectures. It can reduce the hologram complexity for highly parallel interconnections, reduce the number of gates in a SIMD system, and permit simultaneous memory access in a parallel shared memory machine, thereby reducing contention problems. Our fundamental reason for studying this is that architectures for optical computing must be designed for the capabilities and limitations of optics; they must not be constrained by the limitations of electronic systems, which have necessarily dominated approaches to digital parallel computing architectures to date.

REFERENCES

Aho, A.V., J.E. Hopcroft and J.D. Ullman, *The Design and Analysis of Computer Algorithms,* Reading, Mass. Addison-Wesley, 1974.

Borodin, A. and J.E. Hopcroft, ''Routing, Merging, and Sorting on Parallel Models of Computation.'' *Journal of Computer and System Sciences,* Vol. 30, pp. 130-145 1985.

Fortune, S., and J. Wyllie, ''Parallelism in Random Access Machines,'' *Proc 10th Annual ACM STOC,* San Diego, California, pp. 114-118, 1978.

Giles, C. L., and Jenkins, B. K., ''Complexity Implications of Optical Parallel Computing,'' *Proc. Twentieth Annual Asilomar Conference on Signals, Systems, and Computers,* Pacific Grove, Calif., to appear, Nov. 1986.

Jenkins, B.K., et al., ''Architectural Implications of a Digital Optical Processor,'' *Applied Optics,* Vol. 23, No. 19, pp. 3465-3474, 1984.

Lev, G., N. Pippenger, and L.G. Valiant, ''A Fast Parallel Algorithm for Routing in Permutation Networks'' *IEEE Trans. on Computers,* Vol. C-30, No. 2, pp. 33-100, Feb 1981.

Sawchuk, A. A., B.K. Jenkins, C.S. Raghavendra, and A. Varma, ''Optical Matrix-Vector Implementations of Crossbar Interconnection Networks,'' *Proc. International Conference on Parallel Processing,* St. Charles, IL, August 1986; also Sawchuk, et al., ''Optical Interconnection Networks,'' *Proc. International Conference on Parallel Processing,* pp. 388-392, August 1985.

Schwartz, J.T., ''Ultracomputers,'' *A.C.M. Trans on Prog. Lang. and Sys.*, Vol. 2, No. 4, pp. 484-521, October 1980.

Shiloach, Y., and Vishkin, U., ''Finding the Maximum, Merging and Sorting in a Parallel Computation Model,'' *J. Algorithms*, pp. 88-102, March 1981.