

Advances in Digital Optical Computing

Submitted to: International Journal of Optoelectronics. This appeared as:

Murdocca, M. J., "Digital Optical Computing: Some Advances," *The International Journal of Optoelectronics*, **5**, no. 2, pp. 191-205, (Mar. 1990).

Miles Murdocca

Rutgers University, Hill Center

Department of Computer Science

New Brunswick, New Jersey 08903

(201) 932-3547

Abstract

A current approach in applying photonics to digital computing is to interconnect arrays of optical logic gates in free space with regular patterns, such as perfect shuffles or crossovers. Circuit depths and breadths are only slightly worse than when irregular optical interconnects are used, but the optics are significantly simplified by using regular interconnects. Problems relating to the design of digital circuits are summarized as well as approaches to solving the problems.

Introduction

There is a large disparity between the speed of the fastest electronic switching components and the speed of the fastest digital electronic computers. Figure 1 illustrates the problem, indicating that transistors exist that can switch in 5ps while the fastest computers run at clock rates on the order of a few nanoseconds. What are the causes of this slowdown for increasing hardware complexity, and how can we get the system speeds closer to the device speeds?

Transistor	5ps
Ring oscillator	30ps
Logic gate	120ps
Chip	1ns
System	5ns

Figure 1: *The difference in speed between one of the fastest transistors and one of the fastest computers is a factor of ~1000. (Courtesy of Alan Huang, AT&T Bell Laboratories.)*

Limitations of electronics include [1]:

- Electromagnetic interference at high speed
- Distorted edge transitions
- Complexity of metal connections
- Drive requirements for pins
- Large peak power levels
- Impedance matching effects

Electromagnetic interference arises because the inductances of two current carrying wires are coupled. Sharp edge transitions must be maintained for proper switching but higher frequencies are attenuated greater than lower frequencies, resulting in sloppy edges at high speeds. The complexity of metal connections on chips, on circuit boards, and between system components affects connection topology and introduces complex fields and unequal path lengths. This translates to signal skews that are overcome by slowing the system clock rate so that signals overlap sufficiently in time. Large peak power levels are needed to overcome residual capacitances, and impedance matching effects at connections require high currents which result in lower system speeds.

A technology based on optics offers solutions to these problems if we can exploit the advantages of optics without introducing new complexity or new limitations that render the use of optics ineffective. Advantages of optics include [1]:

- High connectivity through imaging
- No physical contact for interconnects
- Non-interference of signals
- High spatial and temporal bandwidth
- No feedback to the power source
- Inherently low signal dispersion

High connectivity is achieved by imaging a large array of light beams onto an array of optical logic devices. There is no need for physical interconnects unless fibers or waveguides are used, so that connection complexity is simplified and drive requirements are reduced. Optical signals do not interact in free space, which means that beams can pass through each other without interference. This allows for a high density of signals in a small volume. High bandwidth is achieved in space because of the non-interference of optical signals, and high bandwidth is achieved in time because propagating wavefronts do not interact. There is no feedback to the power source as in electronics, so that there are no data dependent loads. Finally, inherently low signal dispersion means that the shape of a pulse as it leaves its source is virtually unchanged when it reaches its destination.

In consideration of these arguments we might conclude that optics is a better technology for digital computing, but we need to support this claim by showing how to apply this technology effectively.

A Brief History of Optical Computing

Digital computing with the use of optical components was considered at least as early as the 1940's by von Neumann [2]. If lasers were available at the time, the first digital computers may well have used optics. In the early 1960's and throughout the 1970's and 1980's, optics was employed for computing Fourier transforms of military images in matched filtering operations. Synthetic aperture radar (SAR) signal processing [3-5] matches images in stored photographic form with input images, at a very high rate. Spectrum analysis is performed with acousto-optic signal processing [6]. Both of these applications are performed optically when bandwidth needs exceed electronic capability.

Studies at IBM [7] showed that digital optics would not surpass digital electronics with technology available in the foreseeable future, and arguments in favor of optical computing were hard to support except in a few niche applications like matched filtering. There were renewed interests in the late 1970's as advances were made in optical transmission and optically nonlinear materials.

Limits of electronic digital circuits became more apparent as the need for communication bandwidth became more severe [8-10], and attention returned to optics. A current survey on the field of optical computing can be found in Feitelson's *Optical Computing* [11]. Other works that cover broad aspects of the field can be found in Refs. [12-16].

There is a great volume of literature on various aspects of optical computing such as optical bistability, optically nonlinear materials, architectures, number systems, methodologies, etc. There is far too much work to categorize it all here, so the focus is on projects that have had the greatest influence on the design of general purpose digital optical computers. Except for neural networks, discussion on analog optical computing is intentionally omitted. Error accumulation and accuracy limit the extent analog computing can enjoy in both electronic and optical technologies. There will always be special applications for analog optics such as signal processing, matrix-vector multiplication, and neural networks so the significance of analog optics is not meant to be lessened here, but the role analog optics is likely to play must be placed in perspective with digital optics.

Huang's symbolic substitution [10], which is a parallel method of binary pattern replacement has had a significant influence on the field of optical computing. Progress has been made in optical implementations [17-21] and in computing methods [8,22-25] based on symbolic substitution. Shadow-casting [26] was introduced at least as early as symbolic substitution in 1983, and offers interesting methods for performing spatial logic. The Digital Optical Cellular Image Processor (DOCIP) approach [27] also enjoys support for its generality and simple hardware requirements. Other work on computing methods [28,29] for specific applications increased awareness in the diversity of techniques made available through optics. The field of neural networks has also spurred optical computing research [30,31] and is currently a rapidly moving area.

There has been progress in guided-wave optics using fibers [32], Ti:LiNbO₃ switching elements [33], new approaches using integrated optics [34], and optical interconnects for Very Large Scale Integration (VLSI) [35].

Ongoing work in numerical methods using optics such as matrix-vector computing [36], optical linear algebra [37], modified signed digit (MSD) arithmetic [38], and signal processing [39] has influenced the need for optical logic devices and manufacturable systems. The success of digital optics depends heavily on advances in optical hardware, and a number of efforts have focused on device design [40-43] and free-space implementations of gate-level interconnects [44-47].

Only recently have general purpose digital optical techniques come about that are tailored for technology available today while providing sufficient extensibility for future systems. The methods described here provide a foundation for digital design techniques that promote regularity in structure and simplicity in design methods, while providing sufficient extensibility to meet the needs of larger and faster systems.

Digital Design in Optical Computing

In order to make an all-optical computer, we need optical logic gates, optical interconnects, supporting hardware, and a means for putting it all together. We can consider one contending optical logic gate, the optical logic etalon (OLE) [41] shown in Figure 2. In this device sample, there are several thousand devices that are a few microns in diameter, spaced $10\mu\text{m}$ apart. Light enters the tops of the devices and exits through the bottoms. The devices behave as NOR gates with fan-in and fan-out of two, and achieve an operating rate of 30ps at room temperature.

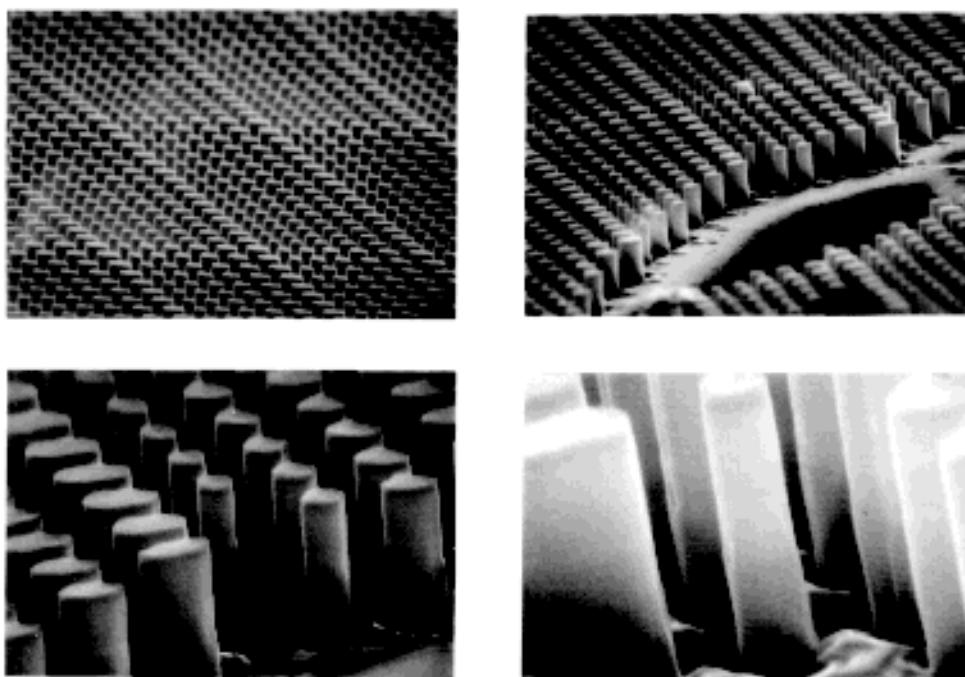


Figure 2: Arrays of optical logic etalons (OLEs), also called microresonators. Devices are a few microns in diameter on $10\mu\text{m}$ centers. Logic gates are etched about $8\mu\text{m}$ deep. Operating speed is 30ps at room temperature, and the logic operation is two-input, two-output NOR. Input beams enter the tops and output beams emerge on the bottom sides, through the substrate. (Courtesy of Jack Jewell, AT&T Bell Laboratories.)

OLEs can be interconnected using free space as the connection medium. Optical components such as beam-splitters, gratings, and mirrors can be used to implement a gate-level interconnect. A functional layout of the crossover interconnect [45] is shown in Figure 3 and an optical implementation is shown in Figure 4. A two-dimensional input image is split into two identical copies. One copy is imaged onto a mirror where it is reflected back through the system to the output plane. The second copy is permuted according to the period of the prism array and is combined with the output image. The connectivity achieved on one pass through this setup is shown in Figure 3b. The interconnect is customized by setting masks in the image planes that block light at selected locations.

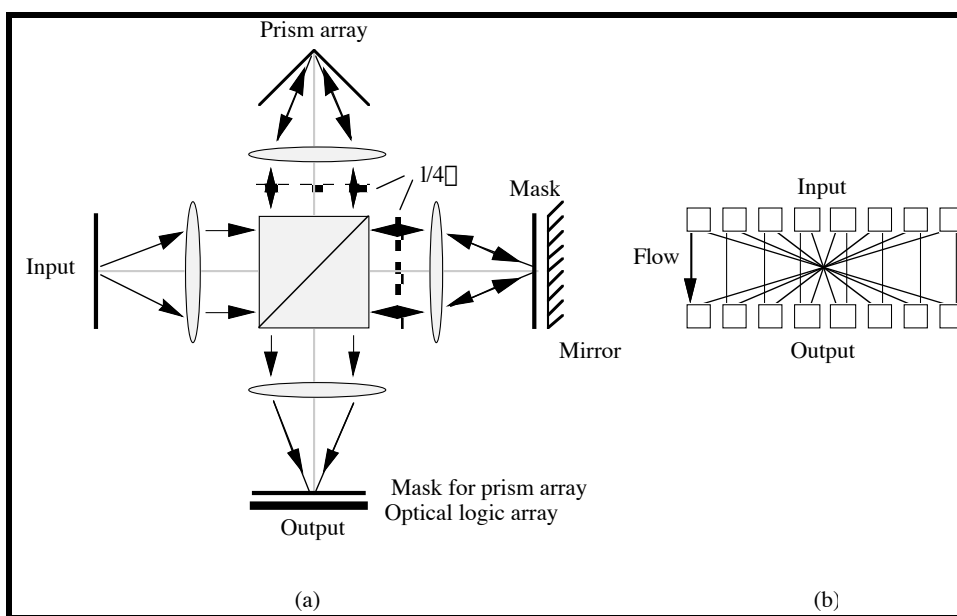


Figure 3: (a) Schematic of an implementation of one stage of the crossover interconnect. A two-dimensional input image is split into two identical copies. One copy is imaged onto a mirror where it is reflected back through the system to the output plane. The second copy is permuted according to the period of the prism array and is combined with the output image. The interconnect is customized by setting masks in the image planes that block light at selected locations. (b) Connectivity achieved for one row of data passing through the crossover stage shown in panel (a).

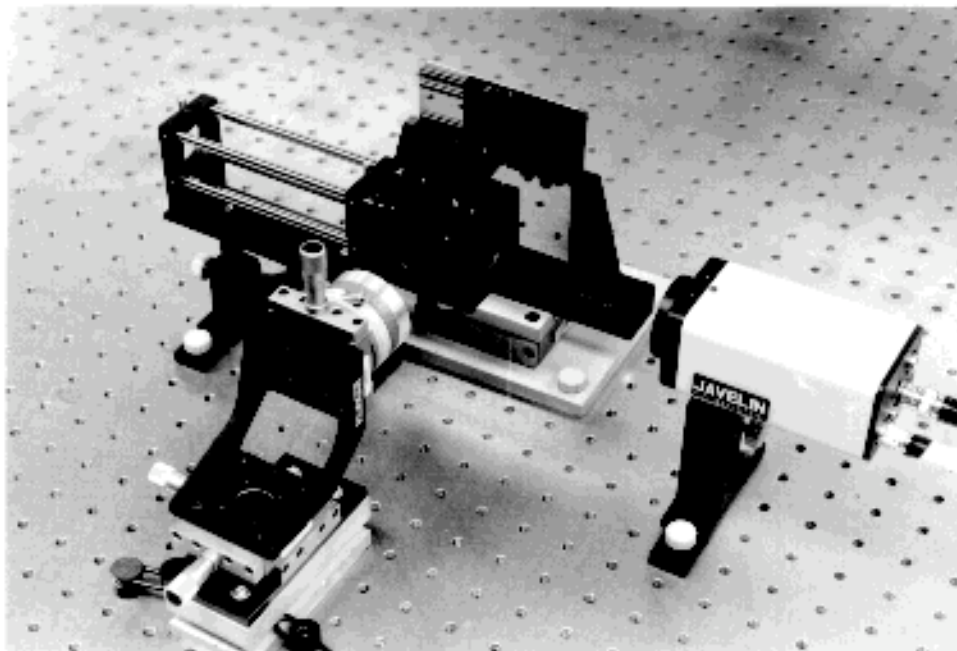


Figure 4: *Optical implementation of the crossover interconnect. The prism array is held in the mount closest to the reader and the mirror is held in the mount opposite the Javelin camera. The beam-splitter is a cube of glass in the center of the structure, between the camera and the mirror. The input image enters the system on the far side of the optical bench opposite the prism array, and the camera captures the output image during experiments. (Courtesy of Jürgen Jahns, AT&T Bell Laboratories.)*

A conceptual layout of an optical computer based on arrays of optical logic devices and crossover interconnects is shown in Figure 5. Optical signals travel orthogonal to the device substrates, through alternating crossover stages and logic stages. The system is fed back onto itself and an input channel and an output channel are provided allowing for a conventional model of a digital circuit. Feedback is imaged with a single row vertical shift so that data spirals through the system, allowing a different section of each mask to be exposed on each pass. This architecture lends itself to an efficient optical implementation [48] which motivates the development of digital design techniques based on this model.

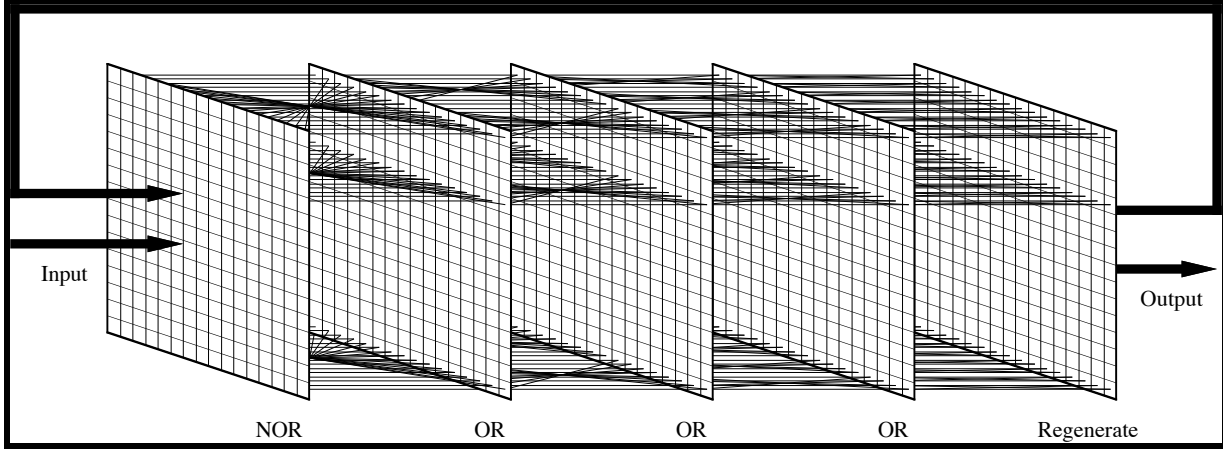


Figure 5: Arrays of optical logic gates are interconnected with crossovers.

The object of digital design for this model is to map the functional description of a digital circuit onto a regular structure. The positions of the logic gates are fixed, all gates have fan-in and fan-out of two, all devices on the same array perform the same function such as OR or NOR, and the interconnects between arrays are crossovers. The only choices the circuit designer has are the positions of the inputs, the positions of the outputs, and the configurations of the masks that block unwanted connections. Algorithmic techniques for mapping functions onto this model are described in Ref. [49], but for the discussion here we can assume we have an automated tool that does the mapping for us.

Consider mapping the next-state (s) and output (z) functions for a dual-rail serial adder onto a system with arrays that are 16 gates wide, two gates tall, and four arrays deep:

$$\begin{aligned}
 s_{t+1} &= \bar{x}y s_t + x\bar{y} s_t + xy\bar{s}_t + xy s_t \\
 \bar{s}_{t+1} &= \bar{x}\bar{y}\bar{s}_t + \bar{x}\bar{y} s_t + \bar{x}y\bar{s}_t + \bar{x}y s_t \\
 z_{t+1} &= \bar{x}\bar{y} s_t + \bar{x}y\bar{s}_t + x\bar{y}\bar{s}_t + xy s_t \\
 \bar{z}_{t+1} &= \bar{x}\bar{y}\bar{s}_t + \bar{x}y s_t + xy\bar{s}_t + x\bar{y} s_t
 \end{aligned}$$

These equations can be transformed into OR-NOR form to match the operating mode of OLEs as shown below:

$$\begin{aligned}
 s_{t+1} &= \overline{\overline{\overline{\overline{\bar{x}y s_t + x\bar{y} s_t + xy\bar{s}_t + xy s_t}}} \\
 \bar{s}_{t+1} &= \overline{\overline{\overline{\overline{\bar{x}\bar{y}\bar{s}_t + \bar{x}\bar{y} s_t + \bar{x}y\bar{s}_t + \bar{x}y s_t}}} \\
 z_{t+1} &= \overline{\overline{\overline{\overline{\bar{x}\bar{y} s_t + \bar{x}y\bar{s}_t + x\bar{y}\bar{s}_t + xy s_t}}} \\
 \bar{z}_{t+1} &= \overline{\overline{\overline{\overline{\bar{x}\bar{y}\bar{s}_t + \bar{x}y s_t + xy\bar{s}_t + x\bar{y} s_t}}}
 \end{aligned}$$

An automated layout program tackles the problem in two parts. First the minterms are generated for the AND stage of a programmable logic array (PLA), and then functions are generated by combining minterms in the OR stage. Figure 6 shows the mapping for the AND stage implemented in OR-NOR form, which generates all unminimized terms of three variables. Each row of logic gates corresponds to one row of a 16x2 array organized into a system of the form shown in Figure 5.

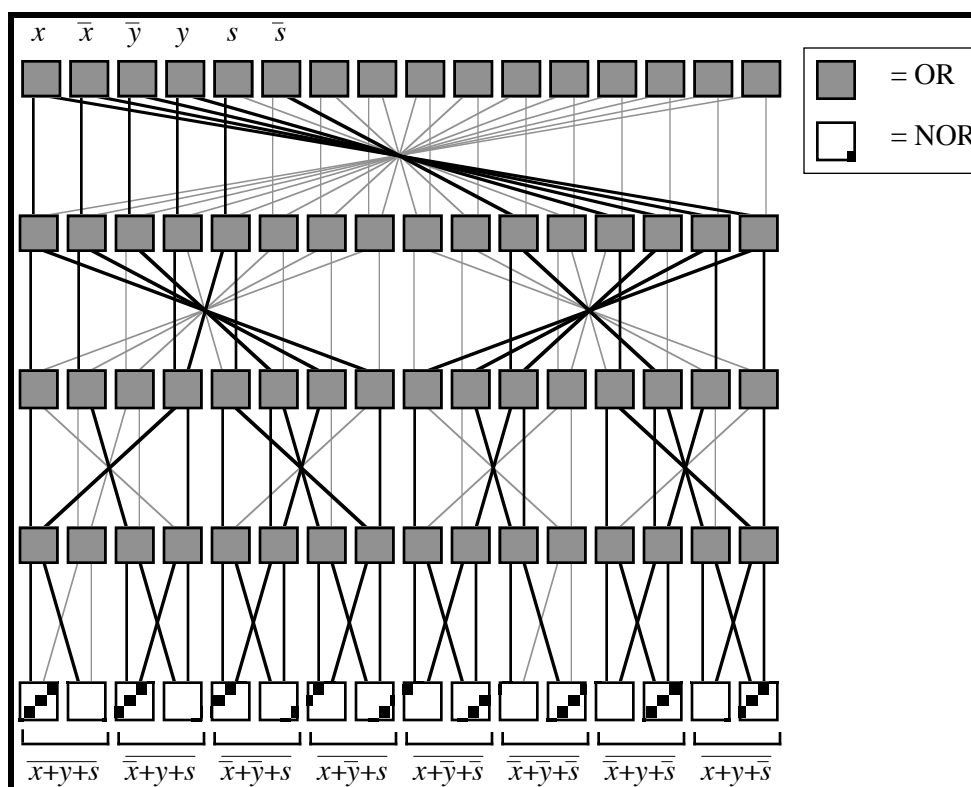


Figure 6: Generation of all three-variable unminimized terms with crossover interconnects and two-input, two-output OR and NOR gates. The bottom stage of logic devices performs logical NOR because an OR-NOR implementation of the AND stage of a PLA matches the operating mode of OLEs. Inputs are provided at the top and two outputs for each term are produced at the bottom. Connections that are masked are dimmed.

The AND stage is implemented on one pass through a four stage system of the form shown in Figure 5. On the feedback path from the last stage back to the first, optical signals are imaged with a single row vertical shift so that new portions of the masks are used on the second pass through the OR stage implemented in OR-NOR form, which is shown in Figure 7.

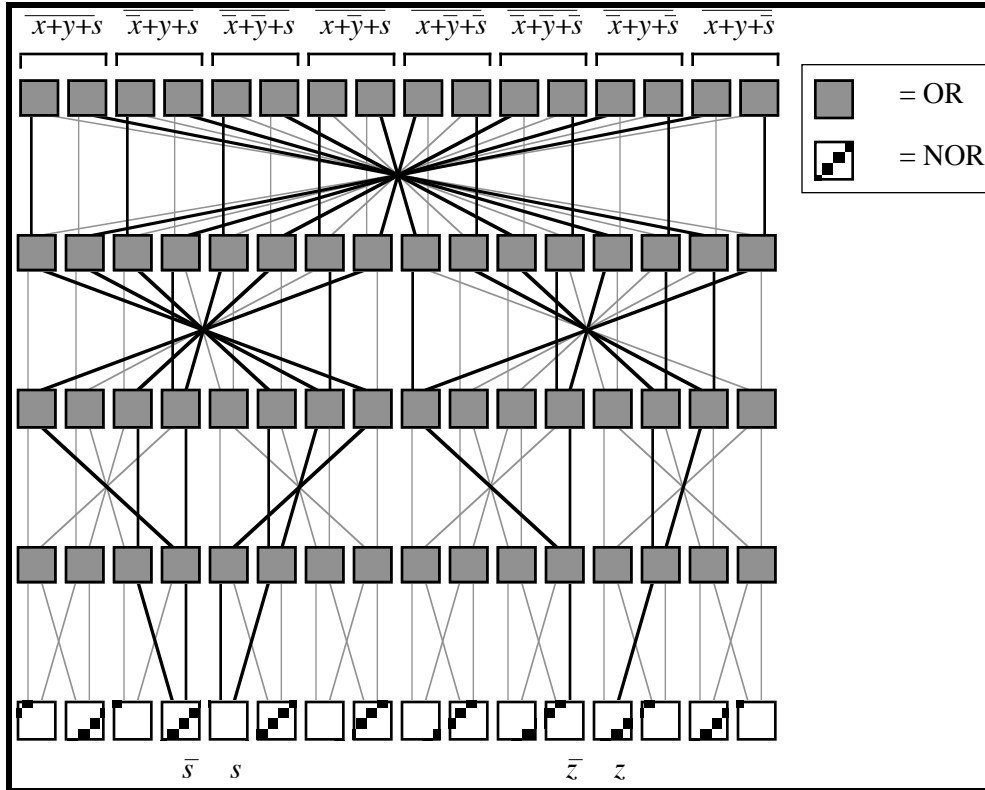


Figure 7: Mapping terms to state (s) and output (z) functions for a dual-rail serial adder in the OR stage of a PLA.

The result of this design process is a set of masks that block light at selected locations that are placed in the image planes of the crossover interconnects. In terms of hardware complexity, it can be shown by properties of binary trees that the AND stage is as shallow as it can be, given the fan-in and fan-out constraints and the fact that two copies of each minterm are generated. The OR stage is one level deeper than optimal given the same constraints. The circuit cannot be any narrower, again applying properties of binary trees. A more complete complexity analysis is given in Ref. [49], but we can conclude from this much of the discussion that we have not paid a high price in the complexity of the target machine by maintaining strict regularity in logic arrays and interconnects.

Regularity pays off in a number of ways. Except for the logic devices, all of the optical components are easily fabricated with existing technology. This should be compared with competing approaches favoring holography [50] and guided wave [51] methods. Every optical path can be made equal to every other within femtosecond delays, so that gate-level pipelining is feasible. For high throughput applications such as packet switching [52] and signal processing, gate-level pipelining can be exploited so that the bit rate is determined by the speed of the

switching devices, as opposed to the latency through a circuit as is the case for VLSI technology.

Regularity pays off for fault avoidance. It is difficult to test each transistor of a VLSI circuit after the circuit is fabricated because of the difficulty of injecting and detecting test signals, and because of device interaction on the chip. For the optical computing model presented here, there is virtually no interaction on the chip so that optical signals can be injected and detected orthogonal to the device substrate, making it easy to detect faults before an array is placed in service. After the faults are cataloged, the positions and orientations of the arrays are determined so that faults line up with masked connections. If such a match cannot be made, then the masks can be modified to route signals around the faults.

Regularity pays off for two other considerations in circuit design: tiling and partitioning. Tiling is the process of laying out subcircuits so that there is a minimum of wasted space, and so that no intolerable latencies are introduced. Padding may be required to equalize delays through adjacent circuits, but this is easily accomplished by adding straight-through levels. Only a finite amount of information can be passed through a practical lens, something on the order of 10,000 signals for current benchtop experiments. A large computer will need several million signals so we are forced to partition a large computer into a number of lens subsystems. We can do this without affecting circuit depth or breadth to any large degree by making use of properties of $\log_2 N$ networks. The basic idea is that a large crossover network can be implemented as a crossover of smaller crossover networks.

The OLE devices are spaced just a few microns apart on an array yet optical signals travel several centimeters through the crossover interconnect in order to interact on the next array of devices. For subnanosecond speeds this latency is intolerable for many applications, particularly those applications that cannot make use of gate-level pipelining. It is not sufficient to measure chip area in describing circuit complexity, free-space volume must be considered as well. A solution to the latency problem is to use a simpler split+shift interconnect that can be implemented with flat optics that reduce array spacings. The concept is shown in Figure 8 for an array spacing of a few millimeters. The circuit layout has more restrictions because only splits and shifts are allowed, and masking is performed in the device image planes rather than in the mirror and prism array planes of the crossover. A dual-rail serial adder implemented for this microoptics model [54] is shown in Figure 9. The circuit is about 25% wider and 50% deeper than the adder circuit shown in Figures 6 and 7, but the gate-to-gate latency has been reduced.

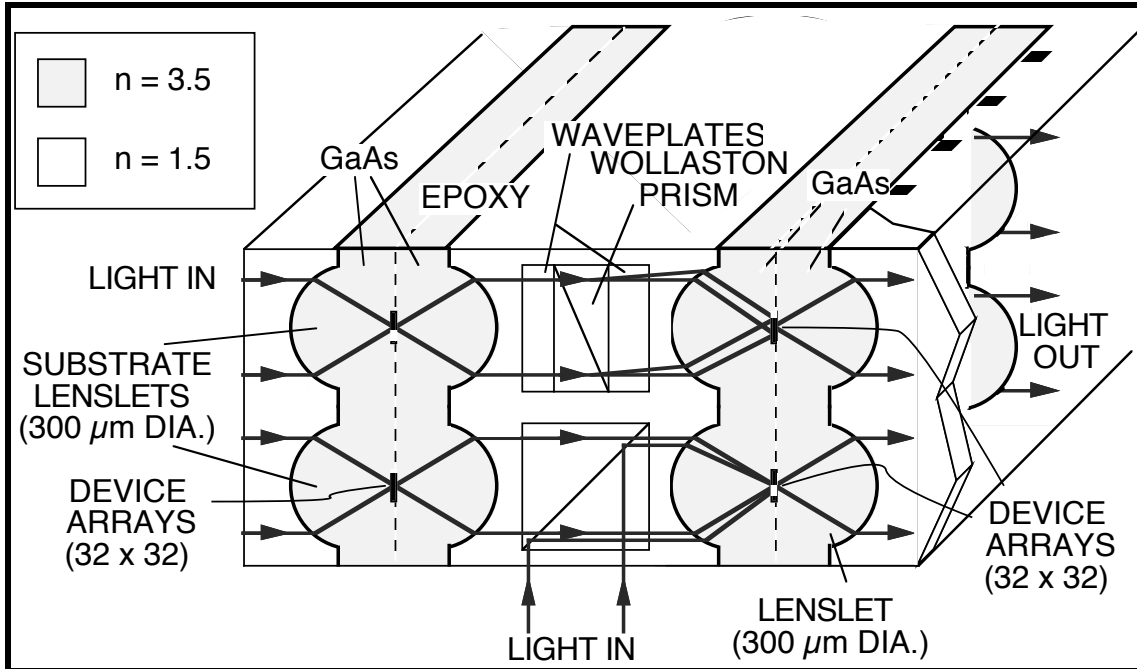


Figure 8: *Microoptics approach for implementing low latency digital optical circuits. Arrays of devices are placed between fabricated lenses that image signals through glass components onto the next arrays of devices. Epoxy binds the structure together. Device spacings are a few millimeters or less. (Courtesy of Jack Jewell, AT&T Bell Laboratories).*

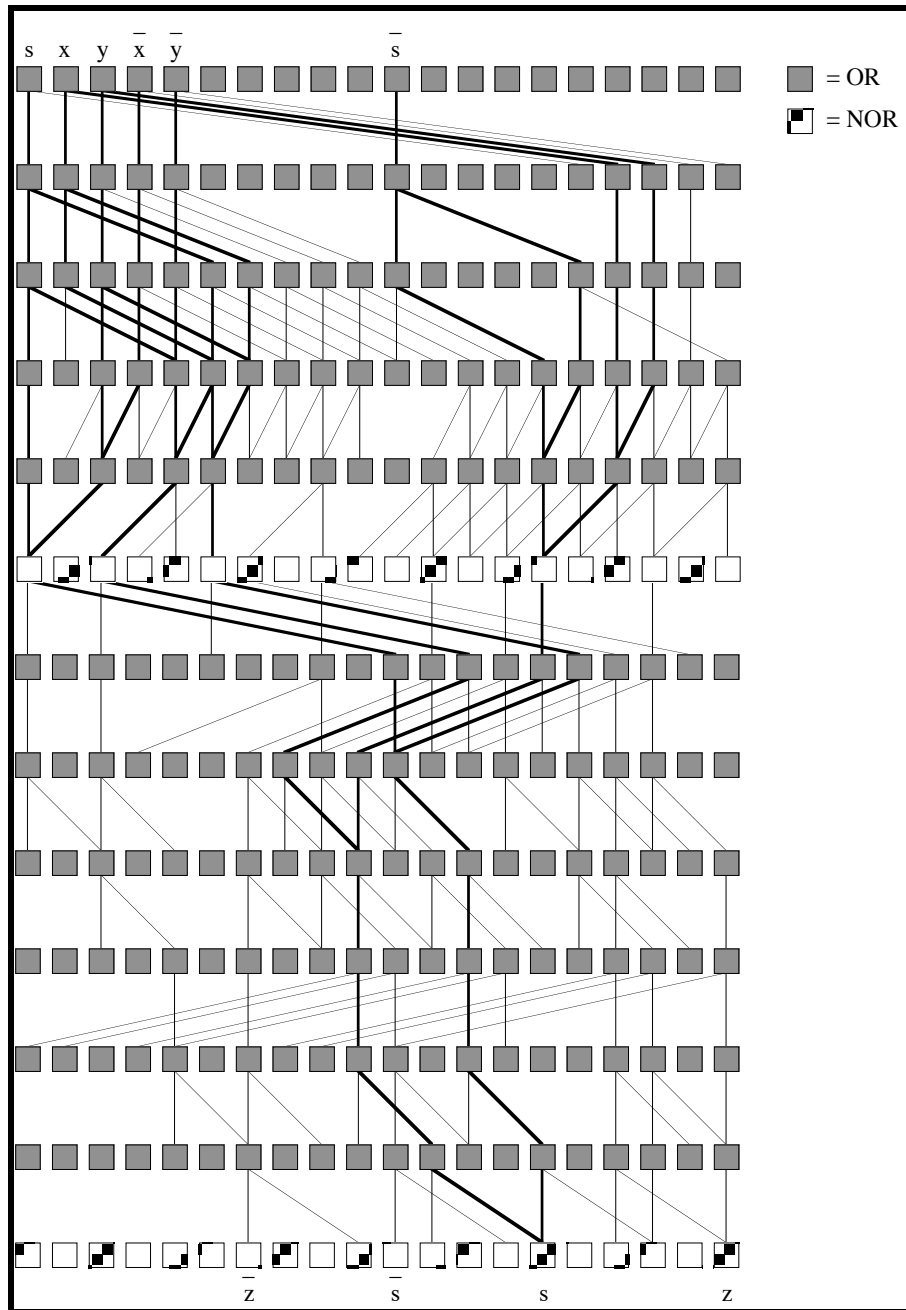


Figure 9: A dual-rail serial adder is mapped onto a circuit making use of split, shift, and combine interconnects. x and y are the streams to be added, z is the output stream, and s is the carry. All logic devices perform the same function at each stage (OR or NOR) and all devices have fan-in and fan-out of two. Masking is allowed in the device image planes only, so that each logic device has either two outputs or no outputs. Connection paths for function s are highlighted.

Optical signals that are imaged onto the mirror and prism array of the crossover travel the same optical path lengths. The optical distance from the mirror to the output plane can be adjusted so that there is little or no time skew that develops between the mirror and prism array legs of the crossover. Optical signals in the form of pulse wavefronts are regenerated at each logic level. All of this means that the optical design can be pipelined at the gate level. When the pipeline is filled, an average of only one row of switching components are devoted to each adding operation. Regularity pays off because throughput can be maintained at the bit rate while time multiplexing the hardware at the gate level.

Finally, consider the fixed masks that block light at the outputs of logic devices. We can use an alternative method for NOR logic if we move the masks to the input planes, change opaque areas to transparent and change transparent areas to opaque, and image light through the masks. Instead of blocking light at the outputs of the NOR gates we are applying light to the inputs, which has the same effect of disabling the outputs of the devices. We can take this idea a step further and replace the masks with active arrays of optical logic devices. We now have the opportunity to completely reconfigure the hardware on every time step. How do we make use of this capability? A potential application is high-throughput, decisionless computation such as matrix multiplication. The sequence of operations is independent of the data so that the control sequence is known before runtime. When multiplying is taking place, terms are not being added so that the machine can alternate between add and multiply modes and achieve the same performance it otherwise would have without reconfigurable interconnects but with less hardware. The potential for dynamically changing masks is largely unexplored, but the message is that this new capability may not have been possible if the architecture did not maintain strict regularity.

Conclusion

A digital optical computing model based on arrays of optical logic gates interconnected in free space with regular patterns is argued to be an efficient means for implementing digital circuits. Limitations of electronics and advantages of optics are discussed, and optical components and systems are described that exploit the advantages of optics. Fault avoidance, tiling, and partitioning are traditional concerns in digital electronics that are argued to be simplified for the model described here. Gate-level pipelining and reconfigurable masks are recognized as new capabilities that compensate for higher gate counts when using regular optical interconnects. It is concluded that regular interconnects implemented with free-space optics provide an efficient means for interconnection among arrays of optical logic devices.

References

- [1] Jordan, H., "Report of the Workshop on all-optical, stored program, digital computers," Technical Report, Department of Electrical and Computer Engineering, University of Colorado at Boulder, (May 1988).

- [2] Von Neumann, J., "Nonlinear capacitance or inductance switching, amplifying and memory devices," in *John von Neumann's Collected Works*, New York: MacMillan, **V**, (1963).

- [3] Cutrona, L. J., E. N. Leith, L. J. Porcello, and W. E. Vivian, "On the application of coherent processing techniques to synthetic aperture radar," *Proc. IEEE*, **54**, 1026, (1966).

- [4] Goodman, J. W., *Introduction to Fourier Optics*, New York: MacGraw-Hill, (1968).

- [5] Haney, M. and D. Psaltis, "Real-time programmable acoustooptic synthetic aperture radar processor," *Appl. Opt.*, **27**, 1786, (May 1, 1988).

- [6] Rhodes, W. T., "Acoustooptic signal processing: convolution and correlation," *Proc. IEEE*, **69**, 65, (1981).

- [7] Keyes, R. W. and J. A. Armstrong, "Thermal limitations in optical logic," *Appl. Opt.*, **8**, 2549, (Dec. 1969).

- [8] Huang, A., "Architectural considerations involved in the design of an optical digital computer," *Proc. IEEE*, **72**, 780, (1984).

- [9] Huang, A., "Design for an optical general purpose digital computer," *SPIE 1980 International Optical Computing Conference*, W. T. Rhodes, Ed., **232**, 119, (1980).

- [10] Huang, A., "Parallel algorithms for optical digital computers," *IEEE 1983 10th International Optical Computing Conference*, H. J. Caulfield, Ed., 13, (1983).

- [11] Feitelson, D., *Optical Computing*, The MIT Press, (1988).

- [12] Bell, T. E., "Optical computing: a field in flux," *IEEE Spectrum*, 34, (Aug. 1986).

- [13] Huang, A., "Impact of new technological advances and architectural insights on the design of optical computers," *Proc. SPIE*, J. A. Neff, Ed., **456**, 25, (1984).
- [14] Arrathoon, R., Ed., "Digital Optical Computing," *Proc. SPIE*, **752**, (Jan. 1987).
- [15] Goodman, J. W., "Optical computing, an historical perspective," *IEEE CH2135-2/85*, 282, (1985).
- [16] Sawchuk, A. A., "Digital logic and computing with optics," SPIE vol. 456: *Optical Computing*, J. A. Neff, Ed., 41, (1984).
- [17] Botha, E., D. Casasent, and E. Barnard, "Optical symbolic substitution using multi-channel correlators," *Appl. Opt.*, **27**, 817, (Mar. 1, 1988).
- [18] Brenner, K.-H., A. Huang, and N. Streibl, "Digital optical computing with symbolic substitution," *Appl. Opt.*, **25**, 3054, (1986).
- [19] Brenner, K.-H., "New implementation of symbolic substitution logic," *Appl. Opt.*, **25**, 3061, (1986).
- [20] Mait, J. N. and K.-H. Brenner, "Optical systems for symbolic substitution," Topical Meeting on Optical Computing, Technical Digest Series 1987, (Optical Society of America, Washington, D.C., 1987), **11**, 12, (1987).
- [21] Wang, L., H. M. Chou, H. M. Gibbs, G. C. Giglioli, G. Khitrova, H.-M. Kulcke, R. Jin, H. A. Macleod, N. Peyghambarian, R. W. Sprague, and M. T. Tsao, "Symbolic substitution using ZnS interference filters," SPIE vol. 752: *Digital Optical Computing*, R. Arrathoon, Ed., 14, (1987).
- [22] Johnson, J. L., "Architectural relationships involving symbolic substitution," *Appl. Opt.*, **27**, 529, (Feb. 1, 1987).
- [23] Murdocca, M. J., "Digital optical computing with one-rule cellular automata," *Appl. Opt.*, **26**, 3365, (1987).

- [24] Murdocca, M. J. and A. Huang, "Symbolic substitution methods for optical computing," SPIE vol. 963: *Optical Computing* 88, Goodman, J. W., P. Chavel, and G. Roblin, Eds., 585, (1988).
- [25] Murdocca, M. J. and B. Sugla, "Design of a symbolic substitution based, optical random access memory," *Optical Computing*, Technical Digest Series 1989, (Optical Society of America, Washington, D. C., 1989), **9**, 92, (1989).
- [26] Tanida, J. and Y. Ichioka, "Optical logic array processor," *IEEE 1983 10th International Optical Computing Conference*, H. J. Caulfield, Ed., 18, (1983).
- [27] Huang, K. S., B. K. Jenkins, and A. A. Sawchuk, "Binary Image Algebra and Digital Optical Cellular Image Processors," Topical Meeting on Optical Computing, Technical Digest Series 1987, (Optical Society of America, Washington, D.C., 1987), **11**, 20, (1987).
- [28] Wherrett, B. S., "All-optical computation: a design for tackling a specific physical problem," *Appl. Opt.*, **24**, 2876, (1985).
- [29] Stirk, C. W. and R. A. Athale, "Sorting with optical compare-and-exchange modules," *Appl. Opt.*, **27**, 1721, (May 1, 1988).
- [30] Abu-Mostafa, Y. S. and D. Psaltis, "Optical neural computers," *Sci. Am.*, **256**, 88, (1987).
- [31] Psaltis, D., D. Brady, and K. Wagner, "Adaptive optical networks using photorefractive crystals," *Appl. Opt.*, **27**, 1752, (May 1, 1988).
- [32] Schaefer, D. H. and J. P. Strong, III, "Tse Computers," *Proc. IEEE*, **65**, 129, (Jan. 1977).
- [33] Hinton, H. S., "Photonic switching technology applications," *AT&T Technical Journal*, **66**, no. 3, 41, (1987).
- [34] West, L. C., "Picosecond integrated optical logic," *IEEE Computer*, 34, (Dec. 1987).
- [35] Goodman, J. W., F. I. Leonberger, S.-Y. Kung, and R. A. Athale, "Optical interconnections for VLSI systems," *Proc. IEEE*, **72**, 850, (July 1984).

- [36] Athale, R. A., "Optical matrix algebraic processors: a survey," *IEEE 1983 10th International Optical Computing Conference*, H. J. Caulfield, Ed., 24, (1983).
- [37] Casasent, D. and A. Ghosh, "Optical linear algebra," *SPIE vol. 388: Advances in Optical Information Processing*, G. Michael Morris, Ed., 182, (1983).
- [38] Li, Y. and G. Eichmann, "Conditional symbolic modified signed digit arithmetic using optical content-addressable memory logic," *Appl. Opt.*, **25**, 1530, (Jun. 15, 1987).
- [39] Rhodes, W. T., "Optical signal processing: Fourier transforms and convolution/correlation" *SPIE vol. 634: Optical and Hybrid Computing*, H. H. Szu, Ed., 57, (1987).
- [40] Gibbs, H. M., *Optical Bistability: Controlling Light with Light*, Academic Press, (1985).
- [41] Jewell, J. L., A. Scherer, S. L. McCall, A. C. Gossard and J. H. English, "GaAs-AlAs monolithic microresonator arrays," *Appl. Phys. Lett.*, **51**, 94, (Jul. 13, 1987).
- [42] Miller, D. A. B., D. S. Chemla, T. C. Damen, T. H. Wood, C. A. Burrus, A. C. Gossard, and W. Wiegmann, "The quantum well self-electrooptic effect device: optoelectronic bistability and oscillation and self-linearized modulation," *IEEE J. Quant. Electron.*, **QE-21**, 1462, (1985).
- [43] Lee, S. H., S. C. Esener, M. A. Title, and T. J. Drabik, "Two-dimensional Si/PLZT light modulators: design considerations and technology," *Opt. Eng.*, **23**, 250, (1986).
- [44] Brenner, K.-H. and A. Huang, "Optical implementation of the perfect shuffle interconnections," *Appl. Opt.*, **27**, 135, (1988).
- [45] Jahns, J. and M. J. Murdocca, "Crossover networks and their optical implementation," *Appl. Opt.*, **27**, 3155, (August 1, 1988).
- [46] Jeon, H.-I. and A. A. Sawchuk, "Optical crossbar interconnections using variable grating mode devices," *Appl. Opt.*, **26**, 261, (Jan. 15, 1987).
- [47] Yeh, P., A. Chiou, and J. Hong, "Optical interconnection using photorefractive dynamic holograms," *Appl. Opt.*, **27**, 2093, (1988).

- [48] Craft, N. C. and M. E. Prise, "Optical systems tolerances for symmetric self-electrooptic effect devices in optical computers," *Optical Computing*, Technical Digest Series 1989, (Optical Society of America, Washington, D. C.), **9**, 334, (1989).
- [49] Murdocca, M. J., A. Huang, J. Jahns, and N. Streibl, "Optical design of programmable logic arrays," *Appl. Opt.*, **27**, 1651, (May 1, 1988).
- [50] Griffith, P. C. and S. A. Collins, Jr., "Efficient holographic optical interconnects using resonated holograms," SPIE vol. 963: *Optical Computing 88*, Goodman, J. W., P. Chavel, and G. Roblin, Eds., 240, (1988).
- [51] Bristow, J., A. Guha, C. Sullivan, and A. Husain, "Optical implementations of interconnection networks for massively parallel architectures," *Optical Computing*, Technical Digest Series 1989, (Optical Society of America, Washington, D. C., 1989), **9**, 18, (1989).
- [52] Murdocca, M. J. and T. J. Cloonan, "Optical design of a digital switch," *Appl. Opt.*, **28**, 2505, (Jul. 1, 1989).
- [53] Murdocca, M. J., *A Digital Design Methodology for Optical Computing*, The MIT Press, (1990).