

Title	OPALS : optical parallel array logic system
Author(s)	Tanida, Jun; Ichioka, Yoshiki
Citation	Applied Optics. 25(10) P.1565-P.1570
Issue Date	1986-05-15
Text Version	publisher
URL	<a href="http://hdl.handle.net/11094/3319">http://hdl.handle.net/11094/3319</a>
DOI	
rights	

*Osaka University Knowledge Archive : OUKA*

<https://ir.library.osaka-u.ac.jp/repo/ouka/all/>

# OPALS: optical parallel array logic system

Jun Tanida and Yoshiki Ichioka

A new optical-digital computing system called OPALS (optical parallel array logic system) is presented. OPALS can execute various parallel neighborhood operations such as cellular logic as well as parallel logical operations for 2-D sampled objects. The system has the ability to perform iterative operations. OPALS is systemized, centering on the optical logic method using image coding and optical correlation techniques. The concept of array logic plays an important role in designing the OPALS, and new optical techniques for constructing the OPALS are proposed. Using these techniques, a pure optical version of the OPALS can be made.

## I. Introduction

The main motivation of recent studies of optical-digital computing is the increasing interest in developing a new parallel computing system capable of processing large amounts of data at high speed. The advantage of optics is the capability of high-speed parallel processing of structured data; that of digital electronics is flexibility. Thus a parallel computing system making good use of the features of both optics and digital electronics will be a powerful tool in the field of information processing in future. Research on optical computing covers a wide variety of fields; e.g., studies on parallel logical gate devices,<sup>1</sup> optical bistability,<sup>2</sup> seeking of novel concepts of optical-digital computers.<sup>3,4</sup> The paper by Sawchuk and Strand<sup>3</sup> is an excellent review of the field of optical-digital computing.

We developed a method of optically implementing parallel logical operations.<sup>5</sup> The method basically utilized the techniques of image coding and optical correlation by a shadow-casting system. Taking note of the analogy between the optical method of implementing logic gates and the concept of array logic,<sup>6</sup> we have constructed a new optical-digital computing system capable of executing various neighborhood operations.<sup>7,8</sup> Named optical parallel array logic system, or OPALS, this optical-digital computing system consists of parallel processing units based on array logic

implemented by optical and/or electronic techniques. The system can also perform cellular logic<sup>9</sup> with its feedback loop.

To realize the OPALS, however, electronic techniques had to be used for image coding and data feedback. Although such a hybrid system is practical, it cannot be expected to handle massive amounts of data. To overcome this drawback, making a pure optical version of the OPALS that can deal with a large amount of data in parallel has been anticipated.

In this paper we propose one realizable optical version of the OPALS, which might be a prototype for future optical-digital computing systems. First, we illustrate the concepts of optical array logic and the OPALS and clarify the difference between the OPALS and conventional computing systems. Then we present a dynamic coding method and a dynamic correlation technique available for implementing optical array logic. Finally, we show an achievable version of the OPALS.

## II. Concept of Optical Parallel Array Logic System

Earlier we presented a method of optically implementing parallel logic operations by using an image coding technique and optical correlation.<sup>5</sup> Here we summarize that method using a simple example.

Figure 1 is a schematic diagram illustrating the parallel logical operation. Input objects are two discrete binary ones (column 1), coded into a single coded object. Coding is the process of replacing corresponding pixels in the two input objects by a specific code pattern according to the combination of their pixel values. Figure 2 shows the relationship between code patterns and the combination of pixel values. All pixels in the input objects are coded into any one of four code patterns in Fig. 2 (column 2).

The authors are with Osaka University, Department of Applied Physics, Yamadaoka 2-1, Suita, Osaka 565, Japan

Received 25 February 1985.

0003-6935/86/101565-07\$02.00/0.

© 1986 Optical Society of America.

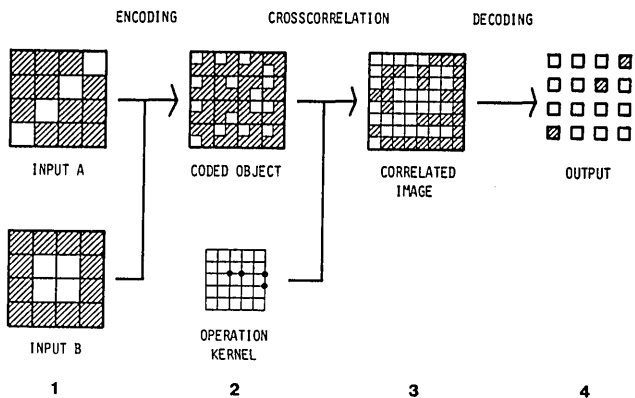


Fig. 1. Illustrative diagram of an optical method of implementing parallel logical operations.

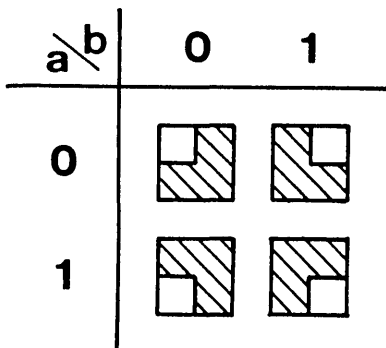


Fig. 2. Code patterns used for the coding process.

The coded object is correlated with an operation kernel consisting of several dots, e.g., light spots, located at specific lattice points. The result is the overlapped version of the coded objects with lateral shifts. The manner of overlapping is shown by a dot pattern. The amount of shift by each dot is set to be half of the amount of the size of the code pattern (column 3). The resultant image is called a correlated image. The signals sampled at the centers of individual pixels in the correlated image give the result of a product term operation for values of a pixel and its neighbors (column 4). A product term operation is defined as a logical operation expressed by a logical product of logical variables  $P_i$  and their negations  $\bar{P}_i$ .<sup>10</sup>

The specific product term operation to be performed is designated by the corresponding operation kernel. Figure 3 indicates examples of relations between operation kernels and corresponding product term operations. The result is expressed by dark-true logic. Dark-true logic is logic that a dark light signal has assigned as true.<sup>1</sup>

Parallel product term operations can be easily implemented using any optical correlation system. As presented in Ref. 5, a shadow-casting system composed of a point source array, an input transparency, and a screen is available for these purposes.

Here it should be noted that any logical operation can be completed by executing an OR operation for results of a sequence of product term operation, be-

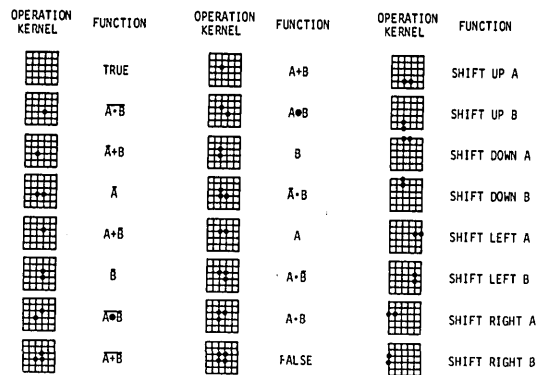


Fig. 3. Examples of operation kernels and corresponding neighborhood operations.

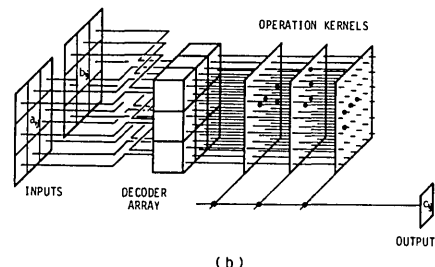
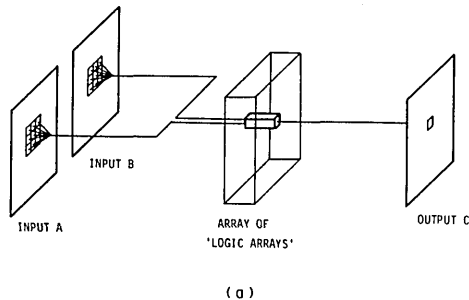


Fig. 4. Concept of optical array logic: (a) optical array logic for two 2-D objects; (b) signal flow in logic array for an element pixel.

cause any logical operation can be divided into several product term operations and an OR operation for them. Unfortunately, the results of individual product term operations are obtained sequentially by dark-true logic. Therefore, we must use several nonlinear devices to hold the intermediate results for the subsequent OR operation.

The method described above is based on the concept of array logic.<sup>6</sup> Array logic is a technique of constructing any logic function using a circuit element with array structure like an electronic memory device (called a logic array). However, the optically implemented logic is different from array logic in electronics because of its parallelism, namely, any logical operation for a pixel and its neighbors is executed spatially with the help of the array logic concept, and this operation can be executed independently for all pixels in the input object in parallel. Hence we named this parallel logical operation optical array logic.<sup>8</sup>

Figure 4 illustrates the optical array logic concept. Figure 4(a) shows the optical array logic for two 2-D

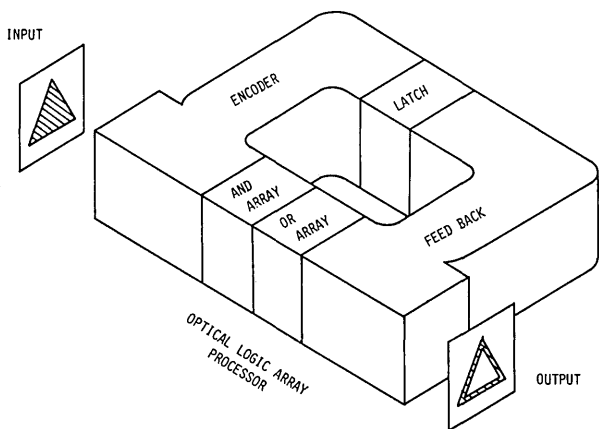


Fig. 5. Conceptual diagram of the optical parallel array logic system.

objects in which one logic array element is exhibited. Figure 4(b) depicts the spatial signal flow in the logic array for an element pixel. An operation kernel in Fig. 1 corresponds to a pattern of nodes in an optical logic array.

Optical array logic is useful for defining and executing parallel neighborhood operations for 2-D objects. In optical array logic, any neighborhood operation can be uniquely defined by a set of operation kernels; it is simply implemented in parallel using an optical correlation technique together with an appropriate image coding method,<sup>8</sup> so that a parallel computing system constructed on the basis of optical array logic will be an efficient system with a simple architecture. The system centers around a parallel processing unit executing parallel neighborhood operations with optical array logic. The OPALS is the optical parallel digital computing system designed on this idea.

Figure 5 is a conceptual diagram of the OPALS. The system, consisting of a loop processor and an input/output port, can execute iterative processing for a binary object. The main functions implementable on the OPALS are logical neighborhood operations. Functions of operations can be changed during iteration.

The OPALS can also implement cellular logic,<sup>9</sup> which is becoming popular in digital image processing. Cellular logic is an operation which converts structured data  $P(i,j)$  into new structures  $Q(i,j)$  pixel by pixel according to the values of a pixel and its nearest neighbors in the original data where  $(i,j)$  express pixel coordinates. To execute cellular logic, processing arrays and large memory elements are required; therefore, practical electronic systems for implementing this logic become generally sophisticated and expensive. Furthermore, it is difficult to increase the number of processing points once the system is constructed.

The OPALS has the ability to overcome these drawbacks. Since a simple optical method is used to implement logical operations in the OPALS, optical cellular logic systems can be constructed much more inexpensively than electronic ones. In addition, it is not diffi-

cult to make a system capable of processing images consisting of 1000 by 1000 pixels if spatial light modulators with high resolution are employed.

Unfortunately, the OPALS needs a coding process in the loop processor, so that high throughputs of the system cannot be expected unless a skillful coding technique is developed. In Sec. III a version of the OPALS is presented, in which a dynamic coding method using spatial light modulators and a dynamic optical correlation technique are utilized.

### III. Implementation of the Optical Parallel Array Logic System

The unavoidable drawback for achieving the OPALS is the necessity of image coding.<sup>7,8</sup> Although two coding techniques have already been presented,<sup>7</sup> both techniques still have problems from the point of view of practical use. They are limitations of spatial resolution, necessity for a transparent type of spatial light modulator, lack of efficient methods of feedback of intermediate processed signals, etc. To solve these problems, we consider a dynamic coding method using liquid crystal light valves (LCLVs) such as Hughes LCLVs and an optical correlation technique using a multifocus imaging system. Then we propose an optical version of the OPALS as one example of optical-digital computing systems.

#### A. Dynamic Coding Method

Image coding required in the OPALS is the process of converting each pair of pixels at the corresponding positions in two input objects into any one of four code patterns in Fig. 2 according to a combination of their values. For convenience, we explain the process of dynamic coding for a pair of pixels,  $a_{ij}$  and  $b_{ij}$ , in two input objects. Code patterns are obtained from linearly polarized light in a cell which is modulated by LCLVs; here, we call the square area around a pixel a cell.

Let us assume that LCLVs with liquid crystal molecules in a 45° twisted nematic configuration can be used in the coding process.<sup>1,2</sup> This device can act as a reflection type of halfwave plate controlled by a 2-D image signal. When no light signal is incident on the write side of a LCLV, the readout light (linearly polarized light) preserves its polarization after reflection, as shown in Fig. 6. On the other hand, when the write light signal illuminates the write side of the LCLV, polarization of the reflected readout light is perpendicular to that of the incident readout light (linearly polarized light).

Figure 7 depicts the coding process for an element cell. The procedures of cell coding are as follows: First, linearly polarized light (column 1) passes through filter  $a$ . The lower half of the cell is a half-wave plate and the upper half is an isotropic glass. Polarization of light beams transmitted through upper and lower halves of the cell are perpendicular to each other (column 2). Second, the linearly polarized light is incident on the readout side of LCLV  $b$ . Assuming that one pixel signal,  $a_{ij}$ , is supplied to the write side of

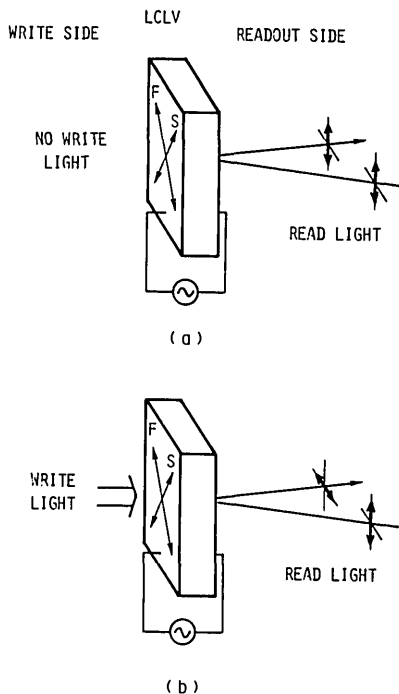


Fig. 6. Function of liquid crystal light valve: (a) no light is incident on the write side; (b) write light is incident on the write side.

LCLV *b*, the direction of polarization of the reflected light rotates according to the pixel value of  $a_{ij}$  (column 3). The polarized light beam is converted into an intermediary code pattern through analyzer *c* which transmits vertically polarized light alone (column 4).

The light passing through analyzer *c* is modulated by the second filter *d* and the second LCLV *e* in the same manner as for filter *a* and LCLV *b* (columns 5 and 6). Filter *d* consists of two parts: the right half is a half-wave plate and the left half is an isotropic glass. The LCLV *e* is controlled by the signal  $b_{ij}$ . As a result, modulated light beams describing any one of four code patterns in Fig. 2 are obtained through analyzer *f* according to a combination of values of the pixels  $a_{ij}$  and  $b_{ij}$  (column 7).

This coding process can be executed for all cells in the input object in parallel. In this case, gratings consisting of strips of halfwave plates must be used as filters *a* and *d*. The imaging system utilized must be adjusted so that the spatial resolution of processed signals matches that of the LCLVs.

### B. Optical Correlation by a Multifocus Imaging System

Code patterns obtained by the dynamic coding method are self-luminous. Therefore, a multifocus imaging system is used for the optical correlation instead of a shadow-casting system. Figure 8(a) shows one example of such an imaging system, consisting of an assembly of small pieces of prisms, an optical shutter array, and an imaging lens. As shown in Fig. 8(b), the switching state of the optical shutter corresponds to an operation kernel designating a product term operation. Thus the product term operations can be

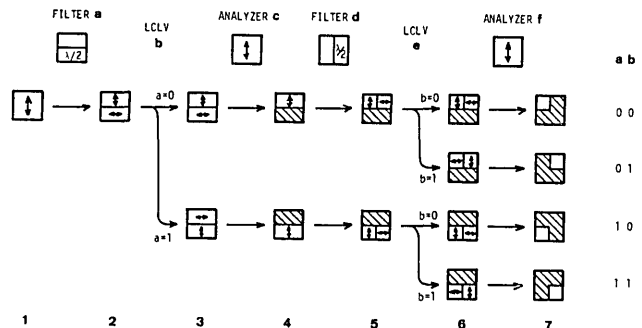


Fig. 7. Dynamic coding process for an element pixel.

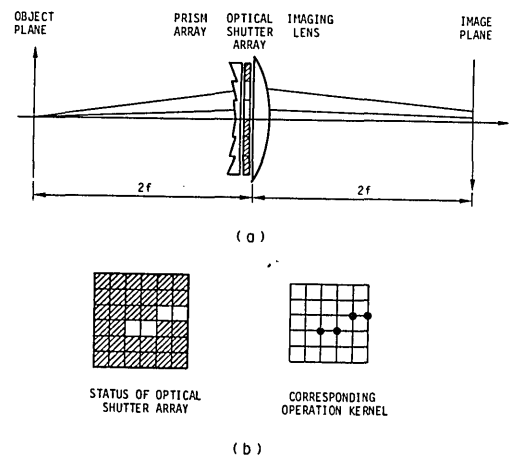


Fig. 8. Optical correlation by a multifocus imaging system: (a) optical setup; (b) function of optical shutter array.

dynamically changed by controlling the state of the shutter array.

The point spread function of the multifocus imaging system is given by

$$h(x,y) = \sum_{m=-2M}^{2M+1} \sum_{n=-2N}^{2N+1} w_{mn} \delta(x - md, y - nd), \quad (1)$$

where  $2M + 1$  and  $2N + 1$  are pixel dimensions of both sides of a neighborhood area, the various  $w_{mn}$  indicate the switching state of the optical shutter,  $\delta(x,y)$  is the Dirac delta function, and  $d$  is a half of the cell size.

In this system, the light signal departing from a specific pixel in the object plane can propagate through free space to the pixels in the image plane designated by Eq. (1). This free space propagation is performed in parallel for every cell in the object. Thus the multifocus imaging system can act as a kind of space-invariant interconnection system, which connects all pixels in the object plane to those in the image plane by a specific interconnection mode. The interconnection mode is dynamically reconfigurable by controlling the switching state of the shutter array, that is, it is changeable by programming. This programmability is one of the most important functions in the implementation of the OPALS presented in this paper.

### C. Latch Devices

As discussed in Sec. II, the OR operation in optical array logic is implemented by a processing sequence: detecting dark signals, inverting its state, and latching it. The result of this operation depends on the input signals history, namely, once a dark signal is put in, the result changes into true, otherwise it holds the initial state, or false. Figure 9(a) shows a function table for the desired operation, and Fig. 9(b) is the circuit that carries out this operation for an element pixel. In the table, output is expressed by bright-true logic.<sup>1</sup> The CLR signal is used to initiate the output state of the operation; in electronics this circuit is called an S-R type flip-flop.<sup>11</sup>

Although the dynamic coding method can produce code patterns dynamically, the state of input objects during a sequence of product term operations must be preserved. For this purpose, latch devices holding input objects are also needed; this device is called a D-type flip-flop.

It should be noted that these flip-flops must be arranged to hold 2-D structures in the OPALS, namely, a number of flip-flops have to operate independently to keep individual pixel data in the object in parallel. At present, we do not know how to achieve such parallel flip-flops. Key techniques or parallel devices available for such a purpose will be produced by future investigations on optical memory devices and optical parallel sequential logic.

### D. Achievable Version of the OPALS

Figure 10 represents a proposed version of the OPALS; this system can be constructed using the components described in the previous sections. A binary object is fed to the loop processor through an input port. After encoding the object, product term operations are executed by a multifocus imaging system. When the sequence of product term operations is completed, the output signals stored in the S-R type flip-flop device, or OR device, provide the result of one parallel neighborhood operation. This image can be monitored through a beam splitter.

For iterative processing, the output image is fed back to the D-type flip-flop device and changes its state. Then the data stored in the D-type flip-flop device are transferred to LCLV *b*. The D-type flip-flop device still holds them during the next sequence of product term operations. If the subsequent operation requires another input object with the fed-back image, the second object can be fed to the system from the input port, synchronizing the feedback timing. The result of the iteration is obtained when a stated criterion is fulfilled. It is obvious from Fig. 10 that we need not take account of light attenuation during iterative processing.

### IV. Discussion

The version of the OPALS described in this paper needs several spatial light modulators. When we construct a practical system, the response time of such spatial light modulators will be the most important

INPUT	CLR	OUTPUT
BRIGHT	BRIGHT	NO CHANGE
BRIGHT	DARK	DARK
DARK	BRIGHT	BRIGHT
DARK	DARK	NOT STABLE

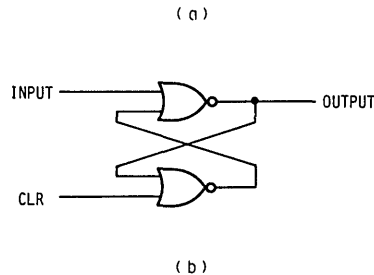


Fig. 9. Operation of an OR device: (a) function table; (b) corresponding circuit.

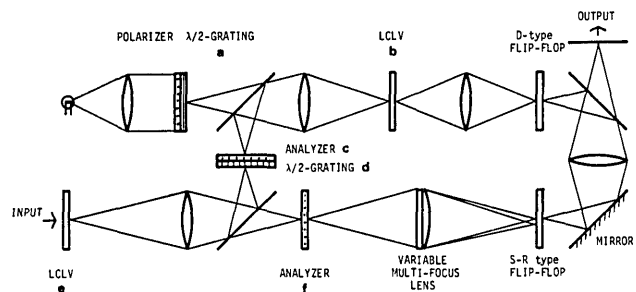


Fig. 10. Example of an optical version of the OPALS.

factor in determining the system's performance. Some comments should be made at this point.

The devices used for product term operations must have the fastest response time because product term operations are the fundamental processing in the OPALS, which is executed most frequently. Thus the method executing fast product term operations holds the key to implementing the OPALS.

However, this system has advantages compared with parallel random logic systems<sup>8</sup> that use spatial light modulators merely as monofunctional gate devices. Namely, the OPALS can execute much more complicated operations within the same time slot than can a random logic system consisting of monofunctional gates. If the same number of spatial light modulators are used for both systems, performance of the OPALS will be much better than that of a parallel random logic system.

From the point of view of a computational architecture, use of latch devices restricts total performance of the OPALS because they prevent information carriers from running free. Thus systems excluding such latch devices are more desirable. Unfortunately, use of the S-R type flip-flop device for OR operations is indispensable in the OPALS. However, if a sequence of product term operations is executed in a very short time and pulse-driven operation is achieved, the D-type flip-flop device can be substituted merely by de-

lay line. If so, system performance will be greatly improved.

Furthermore, although the OPALS is designed for processing binary objects, this system is also applicable to gray-level objects if an appropriate thresholding technique is established. In addition, if an S-R type flip-flop device, which can threshold intensity at any level, is developed, the OPALS will be able to achieve more complicated operations.

## V. Conclusion

We have proposed an example of a pure optical version of the OPALS. However, the OPALS is still a conceptual system, so we may design other versions of the OPALS. Seeking the most preferable and the most achievable system is our future work. We hope the OPALS becomes a prototype for optical-digital computing systems.

## References

1. M. T. Fatehi, K. C. Wasmundt, and S. A. Collins, Jr., "Optical Logic Gates Using Liquid Crystal Light Valve: Implementation and Application Example," *Appl. Opt.* **20**, 2250 (1981).
2. P. W. Smith and W. J. Tomlinson, "Bistable Optical Devices Promise Subpicosecond Switching," *IEEE Spectrum* **18**, (6) 26 (1981).
3. A. A. Sawchuk and T. C. Strand, "Digital Optical Computing," *Proc. IEEE* **72**, 758 (1984).
4. A. Huang, "Architectural Considerations Involved in the Design of an Optical Digital Computer," *Proc. IEEE* **72**, 780 (1984).
5. J. Tanida and Y. Ichioka, "Optical Logic Array Processor Using Shadowgrams," *J. Opt. Soc. Am.* **73**, 800 (1983).
6. H. Fleisher and L. I. Maissel, "An Introduction to Array Logic," *IBM J. Res. Dev.* **19**, 98 (1975).
7. Y. Ichioka and J. Tanida, "Optical Parallel Logic Gates Using a Shadow-Casting System for Optical Digital Computing," *Proc. IEEE* **72**, 787 (1984).
8. J. Tanida and Y. Ichioka, "Optical-Logic-Array Processor Using Shadowgrams. III: Parallel Neighborhood Operations and an Architecture of Optical Digital-Computing System," *J. Opt. Soc. Am. A* **2**, 1245 (1985).
9. K. Preston, Jr., M. J. B. Duff, S. Levialdi, P. E. Norgren, and J. Toriwaki, "Basics of Cellular Logic with Some Applications in Medical Image Processing," *Proc. IEEE* **67**, 826 (1979).
10. A. B. Marcovitz and J. H. Pugsley, *An Introduction to Switching System Design* (Wiley, New York, 1971).
11. M. T. Fatehi, K. C. Wasmundt, and S. A. Collins, Jr., "Optical Flip-Flops and Sequential Logic Circuits Using a Liquid Crystal Light Valve," *Appl. Opt.* **23**, 2163 (1984).

Meetings Calendar continued from page 1564

1986

July

- 16-18 Image Detection & Quality Int. Top. Mtg., Paris P. Chavel, *Institut d'Optique, B.P. 43, 91406 Orsay CE-DEX, France*
- 20-25 Imaging & Photographic Science course, Rochester RIT, P.O. Box 9887, Rochester, NY 14623
- 21-25 Guided-Wave Optical Systems course, Rochester E. Snyder, *Inst. of Optics, U. of Rochester, Rochester, NY 14627*
- 21-25 Synthetic Aperture Radar Technology & Applications course, Ann Arbor Eng. Summer Conf., 200 Chrysler Ctr.- N. Campus, U. of Michigan, Ann Arbor, MI 48109

22-24 Future of Optical Memories Mtg., San Francisco TOC, P.O. Box 14817, San Francisco, CA 94114

28-1 Aug. Fiber Optic Communication Systems, Santa Barbara M. Barnoski, 20200 Sunburst St., Chatsworth, CA 91311

August

4-8 Optical Propagation, Detection, & Communication course, Cambridge Dir. Summer Sessions, Rm. E19-356, MIT, Cambridge, MA 02139

4-8 Contemporary Data Communication Networks: Planning, Management & Computer-Based Design course, Ann Arbor Eng. Summer Conf., 200 Chrysler Ctr.- N. Campus, U. of Michigan, Ann Arbor, MI 48109

4-8 Optical Propagation, Detection, & Communication course, Cambridge Dir. of Summer Sessions, M.I.T., Rm E19-356, Cambridge, MA 02139

4-8 Detection of Infrared Radiation course, Santa Barbara M. Barnoski, 20200 Sunburst St., Chatsworth, CA 91311

4-15 Design & Analysis of Engineering Experiments course, Ann Arbor Eng. Summer Conf., 200 Chrysler Ctr.- N. Campus, U. of Michigan, Ann Arbor, MI 48109

7-9 High Speed Electronics Mtg., Stockholm B. Kallback, RIFA AB, High Speed Electronics, S-163 81 Stockholm, Sweden

11-15 18th Int. Conf. on the Physics of Semiconductors, Stockholm L. Samuelson, U. Lund, Dept. Solid State Physics, P.O. Box 118, S-221 00 Lund, Sweden

11-15 13th Int. Laser Radar Conf., Toronto R. Hoff, 13th ILRC Local Arrangements Committee, 4905 Dufferin St., Downsview, Ontario, M3H 5T4, Canada

11-15 Electrochemical Engineering course, Ann Arbor Eng. Summer Conf., 200 Chrysler Ctr.- N. Campus, U. of Michigan, Ann Arbor, MI 48109

11-15 Non-Ionizing Radiations: Biophysical & Biological Basis, Applications, & Hazards in Medicine & Industry course, Cambridge Dir. of Summer Sessions, Rm. E19-356, M.I.T., Cambridge, MA 02139

12-14 Optical Fiber Measurements course, Vail K. Zimmerman, Off. of Conf. Services, U. of Colorado, Boulder, CO 80310

17-21 80th Ann. Conf. of the Illuminating Engineering Soc. of North America, Boston J. Richards, IES Headquarters, 345 East 47th St., New York, NY 10017

17-22 30th Ann. Int. Symp. on Optical & Optoelectronic Eng., San Diego SPIE, P.O. Box 10, Bellingham, WA 98227

17-22 Diffuse Reflection Symp., Chambersburg F. Barton, 11, P.O. Box 5677, Athens, GA 30613

18-21 Instrumentation & Control Systems course, Baltimore K. Adams, ISA Short Courses, P.O. Box 12277, Research Triangle Park, NC 27709

continued on page 1614