OUKA
The University of Osaka
Institutional Knowledge Archive

Title	Optical parallel array logic system. 2 : A new system architecture without memory elements
Author(s)	Tanida, Jun; Ichioka, Yoshiki
Citation	Applied Optics. 1986, 25(20), p. 3751-3758
Version Type	VoR
URL	https://hdl.handle.net/11094/3024
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

Optical parallel array logic system. 2: A new system architecture without memory elements

Jun Tanida and Yoshiki Ichioka

An optical parallel array logic system—OPALS—is a new type of optical parallel digital computing system. The OPALS optically implements array logic in parallel using techniques of image coding and 2-D correlation for the coded image with a pointwise function called an operation kernel. The OPALS can execute any logical neighborhood operation for binary images in parallel. In this paper we present a new version of the OPALS that needs no memory elements. To construct this OPALS, we consider two useful techniques, i.e., an optical sequential logic technique and multiplex correlation based on wavelength multiplexing. Use of pipelined processing together with the wavelength multiplexing technique gives the promise of eliminating memory devices from the system. Consequently, the new version of the OPALS can be systematized with simpler architecture compared with that of the OPALS presented in the previous paper. Computer simulation verifies the appropriateness of operations of the new version of the OPALS.

I. Introduction

An optical computing technique utilizing the great potential of light in information processing offers an excellent means of large capacity and high-grade processing. Light reads out signals in a 2-D plane and processes them in parallel, so that large capacity of processing is possible. Typical examples of high-grade processing in optics are 2-D correlation and 2-D Fourier transformation.[1] These waste much computing time when electronic computers are utilized. Recently, much effort has been devoted to research in optical computing.[2–11]

In the previous paper we presented a general-purpose optical–digital computing system called the OPALS (optical parallel array logic system).[11] The system effectively utilizes the advantages of the parallel nature of light in information processing and in data transfer. The OPALS can execute parallel array logic[12] using the techniques of both image coding and correlation for the coded image with a pointwise function called an operation kernel. Any logical neighborhood operation for binary images can be executed in parallel on the OPALS.

The OPALS that we proposed before needs three types of spatial light modulator for latching images to be processed, coding the images, and executing OR operation for intermediate processed results. When considering devices and techniques available today, composition of such a version of the OPALS becomes necessarily complicated to make up for imperfections in the performance of these devices. For example, we must use many imaging systems to connect the devices, so that the whole system cannot but be huge. Also, cascadability of the devices must be considered. In addition, from the point of view of a computational architecture, use of many spatial light modulators restricts total performance of the OPALS because they prevent information carriers from running free. Namely, the more spatial light modulators that are placed in the processing cycle, the more time is wasted as the response time of the modulators, and processing speed is reduced.

To improve the total performance of the OPALS, two spatial light modulators for latching and OR operation should be eliminated from the system. According to the discussions of Huang[2] and Jenkins *et al.*,[4] it is better to use one-cycle delay by a feedback system (Fig. 1) than to use special memory elements for latching data in optical computing systems. This one-cycle delay can be accomplished by the period that the light signal propagates through a feedback loop and by the response time of logical devices. In this paper we call this technique optical sequential logic. Using the one-cycle delay technique, the spatial light modulator for OR operation can also be eliminated. As a result, only spatial light modulators for coding images remain in the OPALS.

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

Received 23 January 1986.
0003-6935/86/203751-08$02.00/0.

© 1986 Optical Society of America.

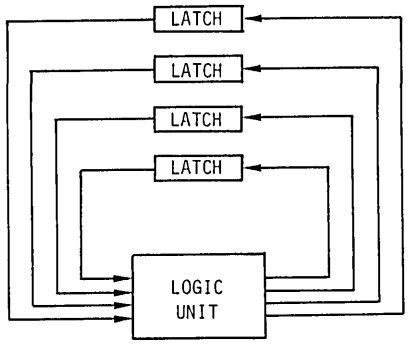


Fig. 1. Feedback system having memory function. Latching elements are used to hold data during one cycle of the logic unit.

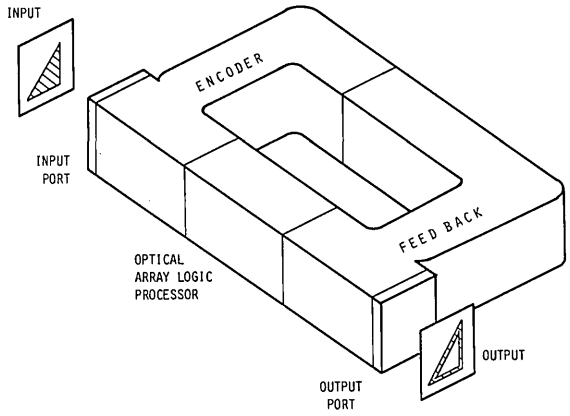


Fig. 2. Schematic diagram of the OPALS. An image to be processed is put into the input port of the system, coded by an encoder, and processed by an optical array logic processor. The result is put out from the output port of the system or fed back to the encoder.

In this paper we propose a new architecture of the OPALS without memory elements. First, we describe a processing algorithm on the OPALS and discuss how to eliminate memory elements from the system. Then we present a multiplex correlation technique capable of optically realizing the OPALS without memory elements. Finally, we show an architecture of a possible version of the OPALS without memory elements. The appropriateness of the operations on the new version of the OPALS is verified by some computer simulation experiments.

II. OPALS and Its Processing Algorithm

The OPALS is a general-purpose optical–digital computing system for binary image data. Figure 2 is a schematic diagram of it. The signal flow in the system is as follows. An image to be processed is put into the system through the input port and processed by some logical neighborhood operations. The resultant image is put out from the system through the output port or it is fed back for the next processing step. When an auxiliary image is required by specific processing, it can be put in through the input port, and logical operations can be performed between the fed-back and the auxiliary images.

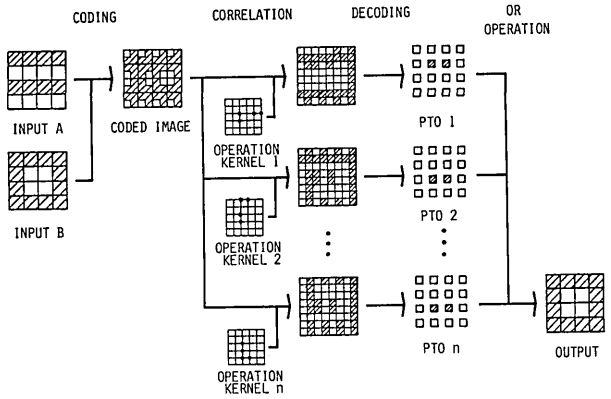


Fig. 3. Procedure of processing by optical array logic according to the expression (output) = $PTO_1[A,B] + PTO_2[A,B] + \ldots + PTO_n[A,B]$, where PTO and + mean product term operation and an OR operator, respectively.

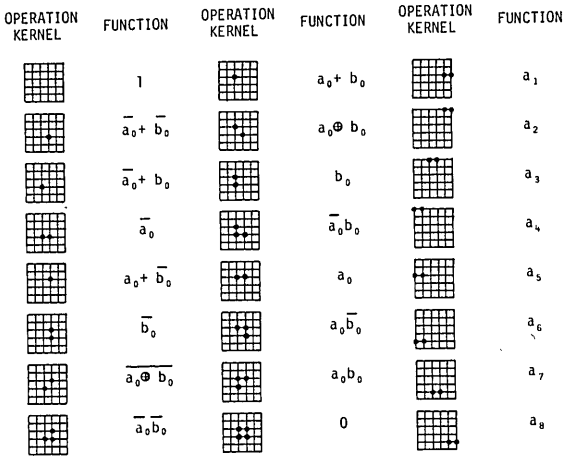
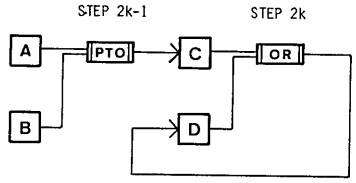


Fig. 4. Examples of operation kernels. a_1–a_8 in the function column shows identifiers of eight neighboring pixels of the center pixel a_0. They are numbered counterclockwise through a_1–a_8.

Processing on the OPALS is based on the concept of optical array logic described in Ref. 10. Optical array logic is executed by two processing steps, i.e., coding of images to be processed and a combination process of the coded image. Any logical neighborhood operation can be carried out by optical array logic. A sort of operation in optical array logic is selected by an operation kernel used for 2-D correlation with the coded image. An operation kernel determines the combination mode of the coded image, so that a series of operation kernels defines the processing program to be executed. Figure 3 indicates the processing procedure based on the concept of optical array logic and Fig. 4 shows some examples of operation kernels.

In optical array logic, a logical operation is divided into two suboperation steps: (1) product term operations for input images and (2) product sum (OR operation) for the product terms (the result of the product term operations). A product term operation is defined as a logical operation expressed by a logical product of logical variables P_is and their negations $\bar{P}_i$.[13] It has



a

STEP	REG. A	REG. B	REG. C	REG. D
1	a	b	PTO 1	
2	a	b	–	PTO 1
3	a	b	PTO 2	PTO 1
4	a	b	–	PTO 1+PTO 2
⋮	⋮	⋮	⋮	⋮
2n-1	a	b	PTO n $\sum_{k=1}^{n-1}$ PTO k	
2n	a	b	– $\sum_{k=1}^{n}$ PTO k	

b

Fig. 5. Processing algorithm of the OPALS with the original architecture: (a) signal flow among registers; (b) status of the registers at individual steps of processing consisting of n product terms. In this case $2n$ steps are needed.

been proved that any logical operation can be expressed by the logical sum of several product term operations.[13]

As shown in Fig. 3, a product term operation is achieved by 2-D correlation for a coded image with an operation kernel and decoding process. Decoding is merely sampling every other pixel of the correlated image. Here, decoded dark and bright signals (pixels) express logical values true (1) and false (0), respectively (dark–true logic). After executing the required n product term operations for the same image, OR operation is carried out for the results obtained by these product term operations. Since the results of product term operations should be interpreted by dark–true logic, the OR operation is the operation providing true logic when any one of its operands is dark logic. The result of the OR operation is that of the desired logical operation consisting of n product terms. Using the same procedure, any logical operation can be executed by optical array logic.

Although, in Fig. 3, n product term operations are shown as being executed concurrently, individual operations are actually executed time sequentially to use spatial light modulators efficiently. To explain this time-sequential operation, we introduce the concept of register. We define a register as a container for an image datum to be processed. This term is also used to identify the operand of a parallel operation, e.g., OR operation for registers A and B.

To execute optical array logic time sequentially, four image registers are used and connected in the manner shown in Fig. 5(a). In this case, as shown in Fig. 5(b), $2n$ operation steps are needed for completing a logical operation consisting of n product terms. Therefore, this processing has disadvantages in processing speed.

If special-purpose spatial light modulators for OR operation are used, processing speed can be reduced. In this case, additional spatial light modulators are required for latching images in registers A, B, and D. Therefore, three types of spatial light modulator for coding, latching, and OR operation must be used for constructing the OPALS. However, various problems arise from the use of many types of spatial light modulator because of the imperfections in the performance of devices developed today. As discussed in Sec. I, system compactness, device cascadability, delay time by many spatial light modulators, etc. must be considered.

III. Pipelined OR Operation

To simplify the architecture of the OPALS, the use of an optical sequential logic technique[2,4] is desirable. Optical sequential logic is a technique to realize any sequential logic using an array of logic gates and an optical feedback system. As shown in Fig. 1, feedbacking outputs of a logic unit (or an array of logic gates) into inputs of a logic unit with one-cycle delay offers a means of sequential logic. The most salient feature of this technique is that no memory device or element is required. Using this technique, one version of the OPALS with a simple architecture can be constructed.

From the processing algorithm in Fig. 5, it is found that the following operations are executed for the registers. They can be expressed by a notation of sequential logic as

$$PTO[A,B] \rightarrow C, \tag{1}$$

$$C + D \rightarrow D, \tag{2}$$

where the capital letters are the identifiers of image registers, $PTO[A,B]$ is a product term operation for registers A and B, and + indicates an OR operator. The arrow indicates the destination of feedbacking, namely, the result of the operation of the left term is fed back into the register indicated by the right term with one-cycle (step) delay.

Operations of latching images in registers A and B are also expressed by

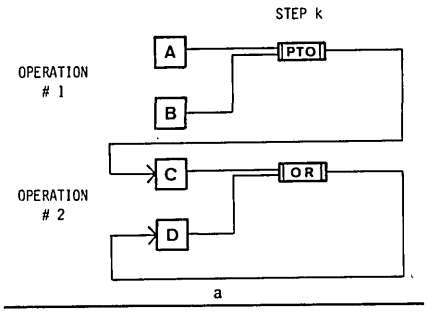
$$A \rightarrow A, \tag{3}$$

$$B \rightarrow B. \tag{4}$$

That is, the contents of registers A and B at the k step are sent to registers A and B themselves at the $k + 1$ step, so that the contents of the registers are substantially preserved even if the registers do not have any memory function. In the following sections we assume that registers have no memory function.

Register A plays a role as an input port as shown in Fig. 2. Thus, we may assume that the contents of register A are latched by another memory element outside of the OPALS. The OPALS can be constructed under this assumption if the three sequential logic operations described by Eqs. (1), (2), and (4) are executed at the same time.

We consider how to implement these operations concurrently. Since the operation according to Eq. (2) needs the result obtained by Eq. (1), operations of Eqs. (1) and (2) cannot be performed simultaneously. However, both operations of Eq. (2) at the k step and



STEP	REG. A	REG. B	REG. C	REG. D
1	a	b	PTO 1	
2	a	b	PTO 2	PTO 1
3	a	b	PTO 3	PTO 1+PTO 2
⋮	⋮	⋮	⋮	⋮
n	a	b	PTO n $\sum_{k=1}^{n-1}$ PTO k	
n+1	-	-	-	$\sum_{k=1}^{n}$ PTO k

b

Fig. 6. Processing algorithm of the OPALS when implementing OR operation in pipelined fashion: (a) signal flow among registers; (b) status of the registers at individual steps of processing consisting of n product terms. $n + 1$ steps are needed for the processing.

Eq. (1) at the $k + 1$ step can be achieved simultaneously if pipelined processing is used. Figure 6(a) shows the processing flow in a pipelined fashion and Fig. 6(b) indicates signal status at each step of a logical operation consisting of n product terms. In this case, $n + 1$ steps of operation are needed for completing the stated processing. When $n > 1$, the number of steps is reduced compared with the case using the algorithm of Fig. 5(b).

The technique described above can be optically achieved using a multifunctional system capable of executing several independent operations concurrently.

IV. Multiplex Correlation

Optical array logic has the capability of performing several logical operations simultaneously. Multiple logical operation can be executed by a multiple correlation technique. This technique consists of multiple correlations by multiplex light carriers with different wavelengths and separation of the results of the multiple correlations by bandpass filters. We call this technique multiplex correlation, which can be utilized on demand.

To implement multiplex correlation, processing parameters such as the number of light carriers and the manner of data arrangement must be determined beforehand. For simplicity, we consider the case using four light carriers with different wavelengths and a data arrangement shown at the right-hand side of Fig. 7. The data to be processed are so arranged that corresponding pixels in four image registers, P, Q, R, and S, are interleaved in the manner shown in Fig. 7. Thus the pixel size of this interleaved datum (or interleaved register) is four times larger than that of each

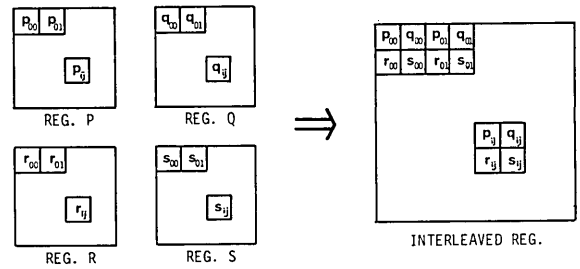


Fig. 7. Data arrangement on spatial light modulators used for multiplex correlation. This arrangement is obtained by interleaving corresponding pixels in four image registers, P, Q, R, and S.

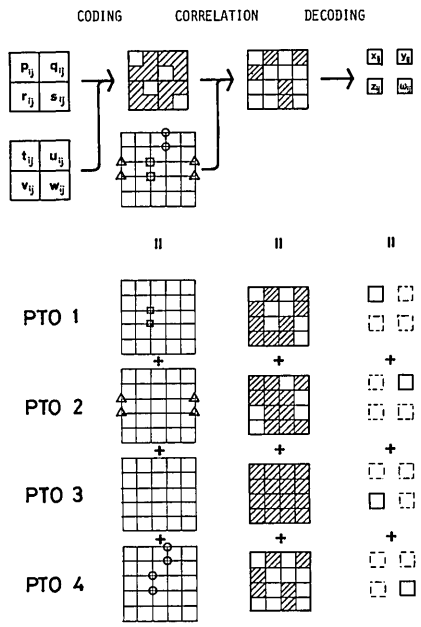


Fig. 8. Schematic diagram of a multiplex product term operation in a logic element. The upper part is the procedure of an actual operation, which can be divided into four independent product term operations as shown in the lower part.

image register. A set of four pixels arranged in square array serve as the pixel element for an image point.

According to the terminology used by Jenkins *et al.*,[8] we define an image point as the resolution element of an image to be processed, while we refer to the resolution element on an image register as a pixel. Also we refer to the set of pixels in an interleaved register associated with an image point as a pixel element.

Figure 8 shows the procedure of multiplex product term operations for a couple of pixel elements. Quadruplex light carriers are employed to achieve multiple correlations of a coded image and four different operation kernels. A correlated image mixed with four different wavelengths of signals is obtained, which can be separated by a decoding mask with bandpass filters. Thus, the following four product term operations are implemented concurrently:

$$PTO_1[P,Q,R,S,T,U,V,W] \rightarrow X, \quad (5)$$

$$PTO_2[P,Q,R,S,T,U,V,W] \rightarrow Y, \quad (6)$$

$$PTO_3[P,Q,R,S,T,U,V,W] \rightarrow Z, \quad (7)$$

$$PTO_4[P,Q,R,S,T,U,V,W] \rightarrow \Omega, \quad (8)$$

where P, Q, R, S, T, U, V, W, X, Y, Z, and Ω are identifiers of registers whose ij pixels are p_{ij}, q_{ij}, r_{ij}, s_{ij}, t_{ij}, u_{ij}, v_{ij}, w_{ij}, x_{ij}, y_{ij}, z_{ij}, and ω_{ij}, respectively; the subscript to PTO means an identifier of a product term operation. These four operations are a set of operations for a couple of pixel elements.

Figure 9 is an example of an optical setup capable of achieving the multiplex logical operation. A coded image is illuminated by an array of point sources. It is assumed that individual point sources can emit light of four different wavelengths independently. Instead of such point sources, four sets of monochromatic point source emitting at four different wavelengths located at near coincident points[14] can also be used for the source. The correlated image is obtained on the screen. The system is set up such that shadowgrams of ij pixels projected by the individual point sources are superimposed on the screen, shifting one another by an amount half of the projected pixel size along the vertical and horizontal directions. A decoding mask with four kinds of bandpass filter is used to separate signals on quadruplex light carriers. Assembly of bright and dark signals detected through the decoding mask represent the result of multiplex logical operations.

It should be noted that the arrangement of output data is equal to that of input data as shown in Fig. 8. Therefore, the output image can be used as one of the input images of the following processing without rearrangement. The multiplex logical operation technique can be utilized together with a feedback technique.

V. Possible System of the OPALS Without Memory Elements

In Sec. III we noted the necessity of concurrent operations of Eqs. (1), (2), and (4) for eliminating memory elements form the OPALS. In Sec. IV we showed the usefulness of the wavelength multiplexing technique for multiplex logical operations. In this section we present a way of composing an actual system for the OPALS.

Although a method of quadruplex operations using light with four wavelengths is mentioned in Sec. IV, careful consideration reveals the fact that triple operations are also sufficient for constructing the OPALS, that is, three concurrent operations of Eqs. (1), (2), and (4) are needed to construct the OPALS. Thus, we assign the four registers of the OPALS to four of twelve registers appearing in Eqs. (5)–(8) in the following manner:

$$A = P, \quad (9)$$

$$B = T = X, \quad (10)$$

$$C = U = Y, \quad (11)$$

$$D = W = \Omega. \quad (12)$$

Registers Q, R, S, V, or Z need not be used, so that they would be available for an extended version of the OPALS in future. It should be noted that each pair of registers, T and X, U and Y, or W and Ω, are assigned to a common register in the OPALS to form a feedback system.

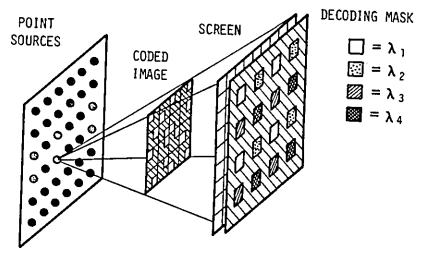


Fig. 9. An Optical setup executing multiplex logical operations. Each point source can emit light with four different wavelengths independently. Four kinds of bandpass filter transmitting light with any one of the four wavelengths are attached to the decoding mask.

To process an image on the OPALS without memory elements the following operational steps are needed:

Step 1: (a) Load the image to be processed into register B from register A, i.e., the input port.
(b) Clear register C.
(c) Set register D.

Step 2: (a) Transfer the contents of register B into register B.
(b) Execute a product term operation for registers A and B and transfer the result into register C.
(c) Execute an OR operation for registers C and D, and transfer the result into register D.

Step 3: Repeat step 2 until the desired product term operations are completed.

Step 4: (a) Transfer the contents of register D into register B with/without inversion.
(b) Clear register C.
(c) Set register D.

Step 5: Return to step 1 if iterative operation is needed.

Some comments are necessary about treatment of register D in this procedure Although OR operation for registers C and D are to be executed in step 2, OR operation is not a product term operation such as Eq. (5). Thus, to implement this operation on the OPALS, we treat the contents of register D by negative logic. As a result, Eq. (2) can be expressed by

$$C + \bar{D} \rightarrow \bar{D}, \quad (13)$$

where the overbar is an inverting operator. In Eq. (13), the operand of the overbar should be interpreted by negative logic. According to Demorgan's theorem (in Ref. 13) Eq. (13) is rewritten as

$$\bar{C}\,D \rightarrow D. \quad (14)$$

Equation (14) is one of the product term operations which can be executed by optical array logic.

Since the contents of register D are interpreted by negative logic, this register is initialized as 1 in steps 1 and 4. After the stated processing, the pixels in regis-

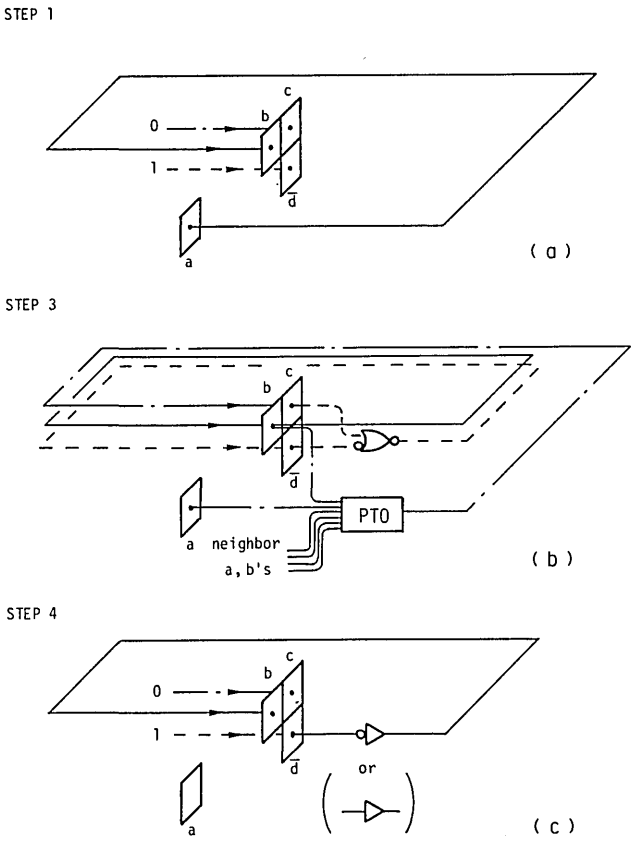


Fig. 10. Signal flow in a logic element at each step in actual processing: (a) step 1; (b) step 2; (c) step 4. Lowercases represent the pixels in the interleaved registers.

ter D, whose values have turned into zeros, indicate those with true logic. Thus, in normal use, the contents of register D are inverted and transferred into register B to execute iterative processing because the contents of register B are interpreted by positive logic. This inversion is optionally disabled.

Figure 10 shows signal flow among pixel elements at steps 1, 2, and 4 of the specific processing. The same processing is made for pixel elements associated with all image points in the image to be processed. Hence, we can carry out any parallel logical operation for 2-D data.

Figure 11 shows one example of the operating process of the OPALS simulated by a computer. Processing is parallel extraction of edge points in a given image. This processing is expressed by the following logical operation:

$$d_{ij} = b_{ij}\bar{b}_{i+1,j} + b_{ij}\bar{b}_{i,j-1} + b_{ij}\bar{b}_{i-1,j} + b_{ij}\bar{b}_{i,j+1}. \quad (15)$$

In Fig. 11, the contents of two input interleaved registers (*ILR.IN*1 and *ILR.IN*2) and that of an output interleaved register (*ILR.OUT*) are drawn with multiplex operation kernels (*MOK*) at individual processing steps, which indicate an actual data arrangement dealt with on the OPALS. Since the OPALS executes iterative processing, the contents of *ILR.IN*2 at the $k + 1$ step is equal to the of *ILR.OUT* at the k step. The contents of registers B, C, and D (*REG.B*, *REG.C*, and *REG.D*) composing the interleaved output register are also drawn to facilitate understanding. Capital letters mean 1 and a dot indicates 0 in the individual patterns in Fig. 11.

The number of individual operation kernels expresses the switching status of a point source by a hexadecimal digit. We assume that a point source is capable of emitting light with quadruple wavelengths independently. Thus, each bit of the hexadecimal number corresponds to the status of emitting light with a specific optical wavelength. The least significant bit is wavelength 1, the second bit wavelength 2, the third bit wavelength 3, and the most significant bit is wavelength 4. For example, $2 = (0010)_2$ indicates the point source emitting monochromatic light of wavelength 2, and $9 = (1001)_2$ indicates the source emitting dichromatic light of wavelengths 1 and 4.

Now, we follow a processing sequence. An image to be processed is put in register A (shown in *ILR.IN*1) at step 1, and it is loaded into register B (shown in *ILR.OUT* or *REG.B*) at step 2. Register D (shown in *ILR.OUT* or *REG.D*) is set at the same time. By means of steps 3–6 the product term operations in Eq. (15) are executed. The result of Eq. (15) is obtained in register D at step 7 but the status of the result is inverted. Since the data in register D should be interpreted by negative logic, the result is loaded into register B after status inversion at step 8. The image in register B at step 8 is the result of the operation according to Eq. (15).

In processing steps 2–6, it should be noted that the contents of register B do not change and that register D acts as a latching element. Although we did not simulate any memory elements in this program, the same function as a memory device can be attained by register D. Processing for other applications can be performed, changing a series of operation kernels. Thus, a series of operation kernels shows the processing program for the stated application.

Figure 12 is an example of an optical system achieving the procedure mentioned above, which is composed of the same elements as the previously proposed OPALS[11] except for the memory devices. Use of liquid crystal light valves (LCLVs)[3] for image coding is assumed. A LCLV does not have any data memory function so that this system has no spatial light modulator for latching images.

For multiplex correlation, a multifocus imaging lens with bandpass filters transmitting light with any one of three wavelengths can be used. Figure 13(a) shows an example of an optical element for this purpose, which consists of an assembly of small pieces of prism, a bandpass filter array, an optical shutter array, and an imaging lens. Since the angle of each prismlet determines the amount of lateral shift of the focused image, any operation kernel can be achieved when constructing a prism array consisting of prismlets with desired shapes. Light with a specific wavelength can be selected by bandpass filters.

Figures 13(b) is an example of an address map indicating wavelength and the amount of the image shift

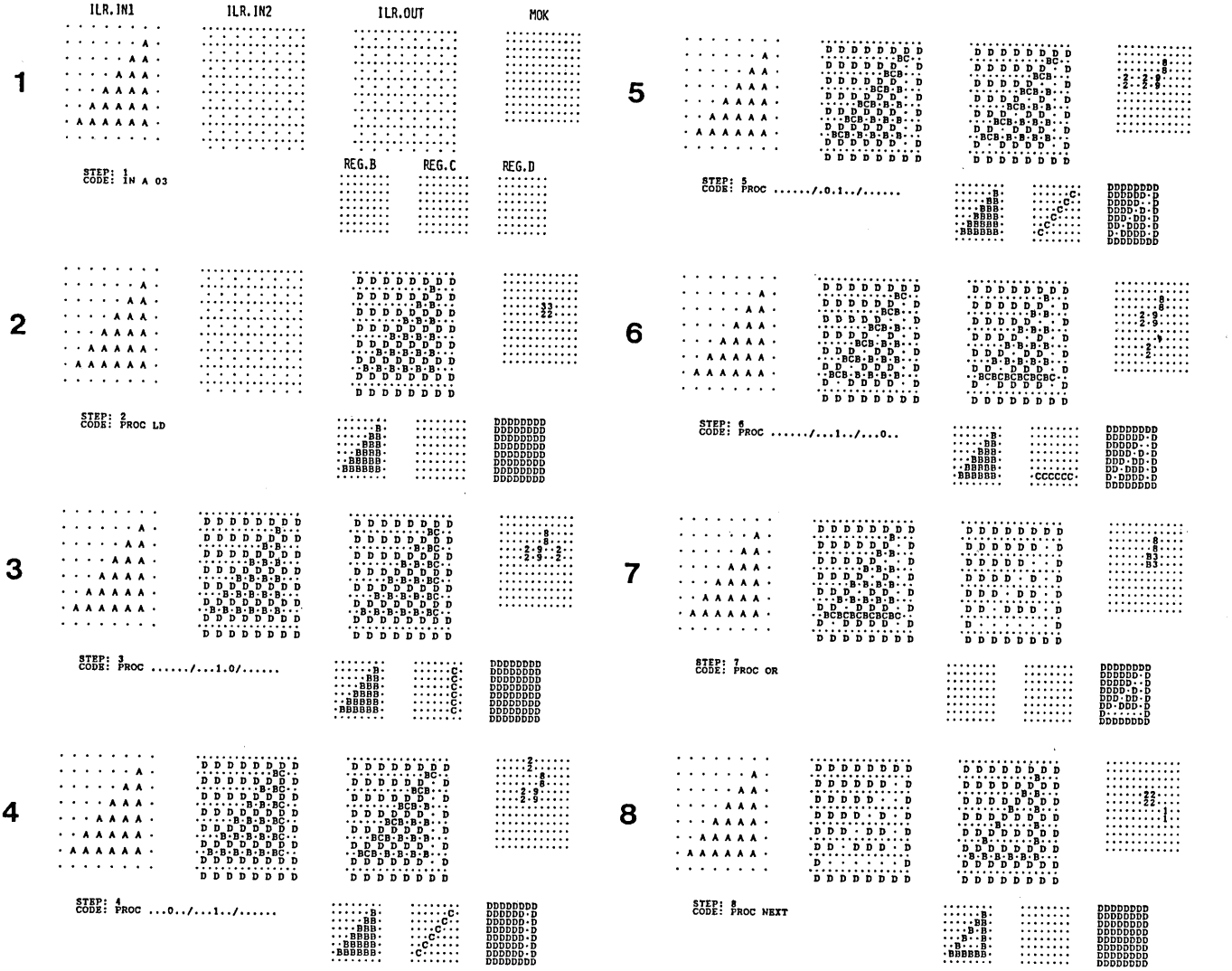


Fig. 11. Example of the operating process of the OPALS simulated by a computer. Processing is to extract edge points in a given image. In pattern blocks of individual processing steps, the contents of two input interleaved registers (*ILR.IN*1 and *ILR.IN*2), that of an output interleaved register (*ILR.OUT*), and the multiplex operation kernel (*MOK*) are drawn. They indicate actual data arrangement in the OPALS. The contents of registers *B*, *C*, and *D* (*REG.B*, *REG.C*, and *REG.D*) are also shown. These separated data are element images forming the image in the output interleaved register.

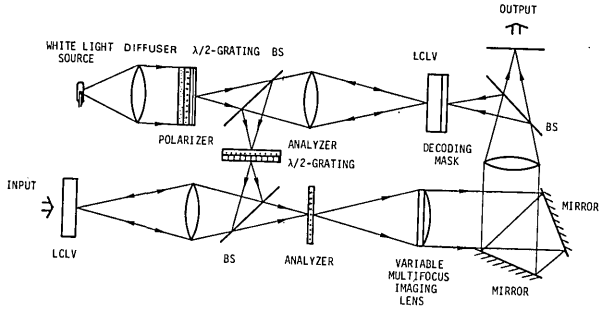
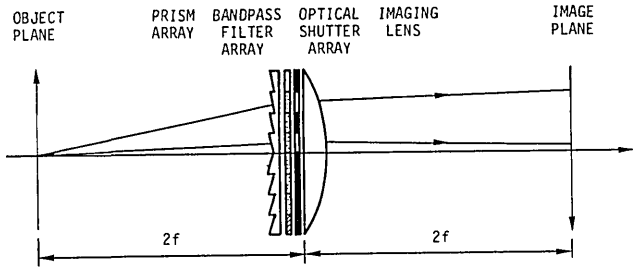


Fig. 12. System setup of the OPALS without memory elements. Two liquid crystal light valves (LCLVs) are used for coding images. A multifocus imaging lens is used for multiplex correlation.

by an array of prismlets. In this address map, the wavelength of light to be passed is indicated by an identifier, and amounts of shift of an image along vertical and horizontal directions are expressed by a 2-D vector designated by the unit of a half-amount of pixel size. The overbar attached to the element of the vector indicates a negative sign. Designing and constructing arrays of prismlets and bandpass filters according to the address map, an OPALS capable of implementing cellular logic[15] can be composed. Namely, all operation kernels needed to execute cellular logic can be realized, controlling the switching status of the optical shutter array. This system has the advantage of compactness of the system setup.

VI. Discussions and Conclusion

We have proposed a new version of the OPALS without memory elements and showed a method of



(a)

#1 (0,0)	#1 (0,1)	#1 (1,0)	#1 (1,1)	#1 (2,$\bar{1}$)	#1 (2,$\bar{2}$)	#1 (3,$\bar{1}$)	#1 (3,$\bar{2}$)
#2 ($\bar{6}$,4)	#2 ($\bar{6}$,3)	#2 ($\bar{5}$,4)	#2 ($\bar{5}$,3)	#2 ($\bar{6}$,0)	#2 ($\bar{6}$,1)	#2 ($\bar{5}$,0)	#2 ($\bar{5}$,1)
#2 ($\bar{6}$,4)	#2 ($\bar{6}$,5)	#2 ($\bar{5}$,4)	#2 ($\bar{5}$,5)	#2 ($\bar{2}$,4)	#2 ($\bar{2}$,3)	#2 (1,$\bar{4}$)	#2 (1,$\bar{3}$)
#2 ($\bar{2}$,0)	#2 ($\bar{2}$,1)	#2 ($\bar{1}$,0)	#2 ($\bar{1}$,1)	#2 ($\bar{2}$,4)	#2 ($\bar{2}$,5)	#2 ($\bar{1}$,4)	#2 ($\bar{1}$,5)
#2 (2,$\bar{4}$)	#2 (2,$\bar{3}$)	#2 (3,$\bar{4}$)	#2 (3,$\bar{3}$)	#2 (2,0)	#2 (2,1)	#2 (3,0)	#2 (3,1)
#2 (2,4)	#2 (2,5)	#2 (3,4)	#2 (3,5)	#2 (0,0)	#2 (0,1)	#2 (1,0)	#2 (1,1)
#4 (0,0)	#4 (0,1)	#4 (1,2)	#4 (1,3)				

(b)

Fig. 13. Multifocus imaging lens used for multiplex correlation: (a) optical setup; (b) address map of the prism array, where the number on the top in each square cell shows the wavelength identifier to be passed and the 2-D vector on the bottom indicates amounts of shift of the focused image along the vertical and horizontal directions expressed by the unit equal to a half-amount of projected pixel size. The overbar attached to the element of the vector indicates a negative sign.

systematization. This system fully uses the great potential of light in parallel processing together with the wavelength multiplexing technique for its operation. Although these types of optical–digital processor have already been proposed,[2,4] their fundamental concept is rather different from that of the OPALS presented here.

The former systems assume use of fundamentally fixed circuit lines or optical paths among logic gates. Functions are controlled by optical signals provided from outside the system. On the other hand, the OPALS dynamically changes circuit lines or optical paths themselves among logic gates and can more easily execute various kinds of processing than the former. However, it is still difficult to conclude which system is better for practical applications. The OPALS has advantages in the ratio of the number of active gates to that of all gates in a spatial light modulator, while the earlier systems have advantages in great tolerance for control of timing and fabrication. Since in the OPALS optical paths must be changed in a very short time, control of the system should be rigid. More studies are needed for solving this problem.

Multiplex correlation presented in this paper can fully utilize the great potential of light in information processing. However, to achieve efficient multiplex correlation, it is necessary to develop new types of active device such as a light source capable of emitting multiple wavelengths of light independently and an optical shutter with variable transmitting characteristics for light at specific wavelengths. Unfortunately, devices with such desirable functions and performance have barely been developed. However, it appears that such useful devices may appear in the near future because they are key devices in optical computing. Although technical difficulties remain in the course of making an actual system of the new OPALS without memory elements, the system will be a powerful tool in parallel processing when these problems are solved.

References

1. J. W. Goodman, *Introduction to Fourier Optics* (McGraw-Hill, New York, 1968).
2. A. Huang, "Design for an Optical General Purpose Digital Computer," Proc. Soc. Photo-Opt. Instrum. Eng. **232,** 119 (1980).
3. 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).
4. B. K. Jenkins, A. A. Sawchuk, T. C. Strand, and B. H. Soffer, "Sequential Optical Logic Implementation," J. Opt. Soc. Am. **72,** 1721A (1982); B. K. Jenkins, A. A. Sawchuk, T. C. Strand, R. Forchheimer, and B. H. Soffer, "Sequential Optical Logic Implementation," Appl. Opt. **23,** 3455 (1984).
5. J. Tanida and Y. Ichioka, "Optical Logic Array Processor Using Shadowgrams," J. Opt. Soc. Am. **73,** 800 (1983).
6. Y. Ichioka and J. Tanida, "Optical Parallel Logic Gates Using a Shadow-Casting System for Optical Digital Computing," Proc. IEEE **72,** 787 (1984).
7. H. O. Bartelt, A. W. Lohmann, and E. E. Sicre, "Optical Logical Processing in Parallel with Theta Modulation," J. Opt. Soc. Am. A **1,** 944 (1984).
8. B. K. Jenkins, P. Chavel, R. Forchheimer, A. A. Sawchuk, and T. C. Strand, "Architectural Implications of a Digital Optical Processor," Appl. Opt. **23,** 3465 (1984).
9. D. Psaltis and N. Farhat, "Optical Information Processing Based on an Associative-Memory Model of Neural Nets with Thresholding and Feedback," Opt. Lett. **10,** 98 (1985).
10. J. Tanida and Y. Ichioka, "Optical-Logic-Array Processor Using Shadowgrams. III. Parallel Neighborhood Operations and an Architecture of an Optical Digital-Computing System," J. Opt. Soc. Am. A **2,** 1245 (1985).
11. J. Tanida and Y. Ichioka, "OPALS: Optical Parallel Array Logic System," Appl. Opt. **25,** 1565 (1986).
12. H. Fleisher and L. I. Maissel, "An Introduction to Array Logic," IBM J. Res. Dev. **19,** 98 (1975).
13. A. B. Marcovitz and J. H. Pugsley, *An Introduction to Switching System Design* (Wiley, New York, 1971).
14. S. Kozaitis and R. Arrathoon, "Shadow Casting for Direct Table Look-Up and Multiple-Valued Logic," Appl. Opt. **24,** 3312 (1985).
15. 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).