Parallel optical logic using optical phase conjugation

George Eichmann, Yao Li, and R. R. Alfano
City University of New York, City College, New York, New York 10031.
Received 3 July 1986.
0003-6935/87/030194-03$02.00/0.
© 1987 Optical Society of America.

Nonlinear optical phase conjugation (OPC) offers solutions to many problems in real-time optical signal and image processing. In an analog mode, using OPC, optical signals (images) can be processed in parallel. Also, using OPC, digital logic implementation has been suggested. The optical logic variables are represented by either the beam on/off or its orthogonal polarization states. Using these representations, a number of binary optical logic elements such as AND, XOR, NOT, have been described. In this Letter, a new method to perform parallel digital optical logic that combines OPC with parallel logic generation techniques is proposed. This technique is suitable to implement optically all sixteen binary logical operations.

To perform parallel optical logic, Bartelt et al. proposed coherent theta modulation, while Ichikawa and Tanida (I-T) and Yatagai suggested incoherent geometric optical shadowcasting (OSC) methods. In the following, the use of both OSC schemes is discussed. In both the I-T and Yatagai approaches the logic encoding is identical, with the difference being in how the different logic operations are performed. While I-T uses different LED source patterns, Yatagai uses a switchable operation mask to obtain different logic operations. In both methods, the optical beams must pass three consecutive, either a source and two input or two input and one output mask. This triple-multiplication implies an optical triple-product operation. To perform an optical triple-product operation, one may use acousto-optic Bragg cells, nonlinear third harmonic generation, as well as the nonlinear OPC. Here, we only discuss an OPC parallel optical logic processing method.

An OPC technique based on the use of Yatagai’s scheme is described first. In Fig. 1, a typical OPC experimental setup is shown. Three input beams generated from the same laser, labeled $E_A$, $E_B$, and $E_C$, are collimated into a cubic $x^{(2)}$ nonlinear material (NLM). The beams $E_A$ and $E_B$ are mutually phase-conjugated. The third beam $E_C$ serves as the probe. The nonlinear interaction of the three beams in the NLM generates a polarization source that radiates a fourth beam:

$$E_D = x^2 E_A \cdot E_B \cdot E_C^*$$  \hspace{1cm} (1)

where the * stands for complex conjugation. In analogy to Yatagai’s parallel logic geometry, the two encoded logic input masks $T_A$ and $T_B$ (the encoding and operation schemes for both Yatagai’s and I-T’s methods are summarized in Table I) are inserted into the path of beams $E_A$ and $E_B$, respectively, while the operation mask $T_C$ is placed on the $E_C$ beam. The phase-conjugate signal $E_D$ separated out by a beam splitter, is the logic output. This output beam possesses the same properties as Yatagai’s arrangement. Since this OPC geometry is no longer collinear, both input and output beams can be separated either spatially or directionally. This separation allows for the optical interconnection of various stages of parallel logic processors. These processors are needed to perform multiple-instruction multiple-data (MIMD) parallel processing. Also, to generate a phase-conjugate signal, as long as $E_A$ and $E_B$ beams counter-
Table I  Optical Encoding (Black/White Code) Techniques for Either the Yatagai- or I-T-Type Optical Parallel Logic Processing

<table>
<thead>
<tr>
<th>coded inputs</th>
<th>logic function</th>
<th>Yatagai operation</th>
<th>I-T input LED pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Both methods use identical input logic variable encoding (see left side). To implement different logic operations, coded operation masks (for Yatagai’s method) and coded input LED patterns (for the I-T method) are shown on the right.

Fig. 1. OPC implementation of a Yatagai-type parallel logic processor: NLM, cubic nonlinear material; BS, beam splitter; $E_A$, $E_B$, and $E_D$, collimated input beams; $E_0$, the phase-conjugate output beam; $T_A$ and $T_B$, two coded input masks; and $T_C$, the logic operation mask.

Fig. 2. OPC implementation of an I-T-type parallel logic processor: $L_1$, $L_2$, and $L_3$, three equal-focal-length Fourier transform lenses; BS, beam splitter; NLM, nonlinear material; $E_A$, $E_B$, and $E_C$, collimated input beams; $E_0$, output beam; $T_A$ and $T_B$, two coded input masks superposed in the plane $A$; $T_C$, a mask containing a central $\delta$-function placed in the plane $D$; $T_D$, a logic operation mask placed in the plane $C$.

Fig. 3. Correlation of the overlapped inputs with four shifted $\delta$-functions: $D$, width of coded input variable mask; $d_x$ and $d_y$, displacements of the $i$th $\delta$-function; (left) overlapped inputs containing four possible illuminated areas (see the left side of Table I); (middle) four shifted $\delta$-functions; (right) the correlation result containing nine possible illuminated areas.

$$E_0 = \chi^{(2)} E_A \otimes E_B \star E_C,$$

where $\otimes$ and $\star$ denote correlation and convolution operations, respectively. To obtain the required multiplication operation for the I-T-type parallel OPC logic, the two logic input masks $T_A$ and $T_B$ are superimposed and placed on the $E_A$ beam. In analogy to the true optical source array, a corresponding source mask $T_C$ is inserted into the path of the $E_C$ beam with four transparent dots representing four displaced Dirac $\delta$-functions. Since no additional correlation is required, the $E_C$ beam mask has a single-on-axis dot representing a central $\delta$-function. To obtain the correct correlation function, the $E_C$ beam $\delta$-function displacements $d_x$ and $d_y$, where $i = 1, 2, 3,$ and $4$, must be chosen as

15 January 1987 / Vol. 16, No. 2 / APPLIED OPTICS 195
\[ |d_{11}| = |d_{22}| = D/4, \]

where \( D \) is the input pixel size. Since the convolution of a function with a \( \delta \)-function shifts that function after the OPC correlation/convolution operation, the needed I-T-type parallel OPC logic is obtained. To insure the correct correlation/convolution result, the NLM cell should be made thin enough to enclose only the optical Fourier spectra of the three interacting beams. \(^8\) \(^9\) For clarity, in Fig. 3, the 2-D correlation of a square and four properly displaced \( \delta \)-functions is illustrated. The leftmost box contains four sub-squares, each of which represents a possible transparent area. When all four middle box \( \delta \)-functions are on, the correlation result, shown in the rightmost box, contains nine possible illuminated areas. With this method, using sixteen possible on/off combinations of the four \( \delta \)-functions, the sixteen two-variable binary logic operations can be performed. In the I-T method, because the input and output areas are not identical, in general, it is difficult to cascade two or more of these processors. For this reason, a conventional (black/white encoded) I-T OSC method is classified as single-instruction multiple-data (SIMD) processing. \(^4\) To perform parallel MIMD operations, another encoding scheme, such as the use of polarization encoding, \(^10\) is needed. In polarization encoding, the two orthogonal linear polarizations are used as two binary states. The thus encoded logic inputs, after a particular OSC manipulation, can generate two sets of orthogonal output patterns representing two different logic operations.

To summarize: the use of a real-time OPC triple-product device to generate coherent optical parallel logic operations is described. A NLM can be used as a major interconnection device that connects logic inputs to different output ports where different logic operations can be performed. The use of both the Yatagai and I-T-type parallel OPC logic implementation schemes is discussed. When both input signals are generated in real time, i.e., by two spatial light modulators, fast real-time parallel logic processing of 2-D data can be performed.

This work was supported in part by a grant from the Air Force Office of Scientific Research No. 84-0144.

References