Reprinted from Applied Optics, Vol. 26, page 2726, July 15, 1987

Digital optical isochronous array processing

George Eichmann, Yao Li, Ping Pei Ho, and R. R. Alfano

An optical isochronous array processing method is proposed. An optical isochronous array processor (OIAP) is a local regularly interconnected processing network that employs an array of identical optical processing elements. In an OIAP, incoming isochronous data are parallel processed in a fashion much like a propagating electromagnetic wavefront. For the various applications, the OIAP processing elements and their interconnections can be different. In this paper, various all-optical OIAP elements are considered. Applications ranging from optical binary number multiplication preprocessing to optical matrix algebra as well as to optical residue arithmetic are presented.

I. Introduction

The three major advantages[1] of optics for the modern day signal processing and computation applications are (1) its ability to process large bandwidth signals at ultrahigh speed; (2) its unguided (free-space) wave propagation property; and (3) its lack of interaction between intersecting beams propagating in a linear medium. A combination of these three salient properties can lead to ultrafast parallel optical signal processing and computing. Using linear optics, a good optical analog parallel processing example is an optical spatial Fourier transform. Other optical analog signal processing methods, such as convolution and correlation, are also available. For numerical computation, however, optical analog processing methods cannot offer in general high numerical precision.[2] To improve the precision, digital calculations need to be used. Thus a parallel ultrafast digital optical computer has long elicited the research interest of optical scientists and engineers.

Among the three parallel computer structures, i.e., vector processor, multiprocessor system, and array processor (AP), the first two are general purpose, while the last belongs to the special purpose computer category. With an AP, data are parallel processed either synchronously with a global system clock or asynchronously in a data-driven fashion.[3] Because these APs offer solutions to a large variety of signal processing problems, there is a considerable interest in the study of their architectures and programming languages. To implement optically an AP, some of its unique features, such as the parallel input–output channels and the spatially local gate interconnections, need to be considered.

In this paper, methods to implement various optical AP (OAP) architectures are proposed. Since these OAPs can process incoming data in an isochronous fashion, that is, whenever the inputs arrive at the same time outputs are generated, the term, optical isochronous array processor (OIAP), is used. Since the IAP is also a locally regularly interconnected network, it is a subset of the systolic AP (SAP). The difference between the two is that in a SAP, the elemental processor is an arithmetic processor (i.e., adder and multiplier), while in an IAP, lower level processors such as a logic gate can be used. With the current optical technology, an OIAP is easier to implement. In the following, for the various OIAP operations, as fundamental processing units regularly interconnected ultrafast nonlinear optical logic elements are proposed. Since some of the processing units can have a femtosecond response, it is possible that these pipelined OIAPs will process data in the picoseconds. The paper is organized as follows: in Sec. II a number of all-optical elemental processing units are briefly discussed. In Sec. III, the use of AND element-based OIAP for optical binary multiplication is described. In Sec. IV, various OIAP-based binary element matrix algebra processors are presented. In Sec. V, fundamental residue mapping units are described, while in Sec. VI, based on these mapping units, an OIAP matrix–matrix residue multiplier is proposed. Finally, Sec. VII summarizes the results of this paper.

The authors are with City University of New York, City College, New York, New York 10031.
Received 17 December 1986.
0003-6935/87/142726-08$02.00/0.

II. Ultrafast OIAP All-Optical Elemental Processing Units

An isochronous wavefront AP is an array of locally interconnected identical processing units. For different data processing applications, the processing unit can be different. In this section, some possible elemental OIAP processing units are described. For the use of an OIAP for arithmetic processing, such as a binary scalar, vector as well as matrix multiplications, an algorithm, the so-called digital multiplication via analog convolution[2] (DMAC) scheme may be utilized. In this approach, analog optics is commonly used to obtain as the first step a mixed-binary output format convolution result. In principle, for the multiplication of two large numbers, compared to traditional shift/add multipliers, the DMAC processor is faster. However, its actual performance is limited by electronic A–D postprocessing.[4] With a current version of DMAC binary convolver, the two fundamental processing units are an AND gate and a summer. The summation operation can easily be implemented via a lens. To perform the AND operation, most approaches adopt a hybrid, either an acoustooptic (AO) or an electrooptic (EO), methodology. To increase the processing speed, an all-optical convolution method must be used. In this section, for the OIAP convolution preprocessing step, several ultrafast all-optical elemental processing approaches are proposed.

Currently, there are a number of techniques available to perform an all-optical AND logic operation. However, among these only those AND elements that have a spatially symmetrical input and output channel format are suitable for an AP. This format will not introduce additional time delay, a delay that tends to slow down the computational speed. As possible symmetric input and output channel optical AND elements, four different ultrafast gates are next briefly considered. A simple AND device is a three-input-beam optical nonlinear etalon[5,6] [see Fig. 1(a)]. The two symmetrical beams A and B are the logic inputs, while the third (middle) input beam R serves as a bias or optical reference. Initially, using the R beam, the etalon is tuned to a low power transmission state. Next, the A and B beam intensities are adjusted so that when all the three beams are on, the total optical power reaches a switching threshold. Above the threshold, the etalon is in a high transmission state switching the signal R from the input to the output. The angle θ is adjusted so that no bistable switching phenomena occur to A and B. This is an important condition since the two off-axis outputs must serve as the inputs to all subsequent OIAP stages. Using a similar geometry, a three (two for logic and one for optical bias) input optical Kerr gate[7] [see Fig. 1(b)] can also act as a symmetric channel optical logic AND element. There are two differences between the Kerr and etalon AND gate. With Kerr AND gate, the logic input and output have different polarization states, and because this device is not bistable, a clear switching threshold does not exist.

A third candidate is an optical phase-conjugate AND gate[8] [see Fig. 1(c)]. The beams A and B act as the two logic inputs, while the beam R is a bias (reference)

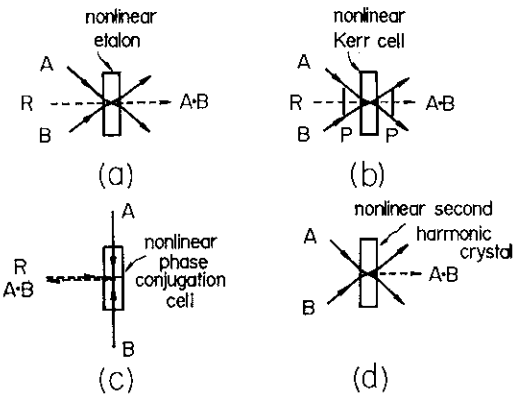


Fig. 1. Four all-optical AND candidates as elemental processing elements for OIAP. A, B, and R, two logic and a bias input beams; P, linear polarizer.

source. Due to the nonlinear phase conjugation via degenerate four wave-mixing effect with respect to the reference beam the logic AND output beam travels in a counterpropagating direction. There are other possible all-optical six-port logic AND gate structures, i.e., structures with two logic and one power supply inputs. However, from an OAP implementation point of view, these elements are not ideal, because with an array a large number of optical bias (reference) channels must be established. For this reason, it is better to use a gate that uses only two symmetric inputs without the need of a bias beam. An optical second harmonic generation (SHG) or parametric wave generation via nonlinear optical three-wave mixing[9,10] AND device does not require a bias beam. A SHG gate has also the potential for a femtosecond response. In Fig. 1(d), a SHG-based AND gate is shown. Two symmetrical identical frequency and polarization logic input beams, denoted A and B with an angular separation θ, are directed into the SHG crystal. When the angle is adjusted so that the so-called 90° phase-matching condition[9] is satisfied, in the bisecting input angular direction, a second harmonic (SH) output signal is generated. One advantage of this structure is that, since only a small part of the input energy is converted to a SH output, most of the fundamental input power passes through the gate allowing the outputs to be used as inputs to feed subsequent AND gates. A second advantage is that a number of such AND gates can optically be interconnected on a single SHG crystal. Thus, except for the change in the output frequency, a SHG-based AND gate is a good candidate for an elemental device.

Next, an AND gate-based OIAP operation is discussed. In Fig. 2, a schematic OIAP network where the intersections indicate the AND element placements is shown. The outputs from these elements are marked by dashed lines. To guarantee the isochronous arrival of the two optical signals at each intersection, using either a holographic grating or a composite prism,[11] both input wavefronts are tilted at an angle θ (see Fig. 2). The output of each logic AND gate is directed to subsequent logic or memory device for further processing. Using an array of such all-optical

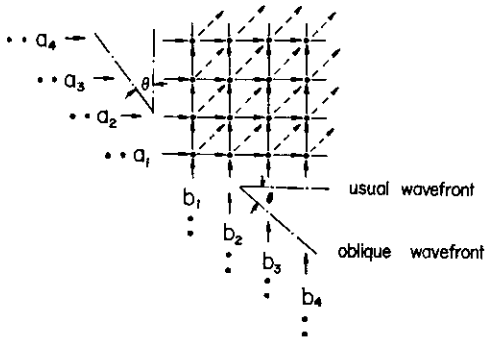


Fig. 2. Schematic of a rectangular OIAP. a_i and b_j are parallel isochronous incoming data, θ is an oblique wavefront angle for synchronizing the input data. An intersection indicates the presence of an identical elemental processing unit generating a dashed line output.

AND gates, an OIAP network can be constructed. For multistage operation, since it is necessary to convert the SH signal back to its fundamental frequency, parametric frequency down (PFD) conversion is needed (see Fig. 3). Using a parametric wave-mixing process with a strong fundamental frequency third harmonic (TH) beam, the SH signal can be converted to the fundamental and amplified. With a KDP crystal, the process has been experimentally demonstrated.[12] In the next sections, the utilization of an AND gate-based OIAP to the implementation of various optical digital arithmetic operations is discussed.

III. Digital Multiplication Using an OIAP

In this section, the computation of both fixed and floating point binary number scalar multiplications, using various OIAP preprocessing networks, is described. In either case, the OIAP is to be used as a convolution preprocessor. Since the convolution results are represented in the mixed-binary format, to complete the operation it must be followed by either an electronic or optical A–D converter[13,14] and a shift/add (S/A) array. In terms of numerical complexity, the implementation of a highly accurate electronic A–D converter is as difficult as the multiplier itself.[4] Thus a useful DMAC processor must employ both a highly accurate ultrafast optical convolver and a A–D postprocessor. In the following, the ultrafast optical implementation of the first part of a DMAC processor is proposed. For our discussion, irrespective of the practical input crossing angle inside the crystal, the schematics are drawn with all the input channel beams to intersect perpendicularly.

The magnitude of the product of two N-bit binary numbers,

$$A = \sum_{i=0}^{N-1} a_i 2^i \qquad B = \sum_{j=0}^{N-1} b_j 2^j,$$

can be expressed in two steps as[4]

$$|P| = |AB| = \sum_{i=0}^{2(N-1)} C_i 2^i, \tag{1a}$$

where

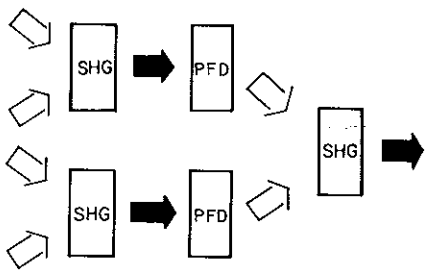


Fig. 3. Staged SHG-based OIAP network with nonlinear parametric frequency-down (PFD) conversion devices as interconnection elements.

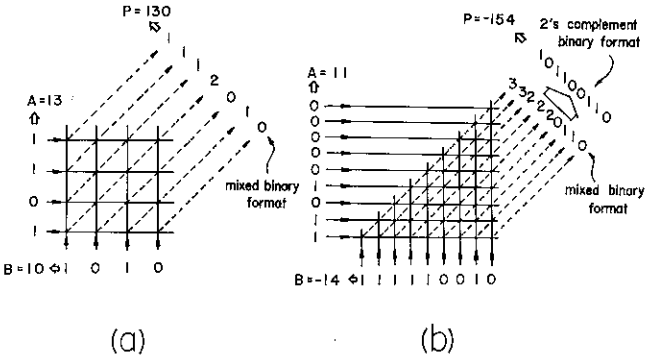


Fig. 4. Two OIAP's for binary (a) unsigned and (b) two's complement signed multiplication preprocessing. Parallel spatially encoded inputs enter from the left and the bottom part of the diagram. Outputs generated at each AND gate beam intersection are used to form the mixed-binary multiplication result.

$$C_i = \sum_{k=0}^{i} a_k b_{i-k} \qquad (0 \leqslant k \leqslant N-1) \tag{1b}$$

is the ith digit weight of resulting product in a mixed-binary format.[2] Equation (1b) can also be interpreted as a ith digit result from the two number sequence convolution. To obtain this multiplication, using the DMAC algorithm the binary convolution indicated in Eq. (1b) needs to be first performed. This convolution in a real-time scheme uses sequential electronic signals to drive two cascaded acoustooptic (AO) Bragg cells.[15,16] Using integrating lenses, the deflected wavefronts are collected at their corresponding detectors. Because of the serial input, the convolution of two N-bit words requires $2N - 1$ computation cycles. The cycle time is mainly determined by the speed of the acoustic wave. To increase the convolution speed, next, an OIAP parallel digital convolution scheme is proposed.

For implementing a two 4-bit binary number parallel multiplication, consider a 4×4 optical AND gate array [see Fig. 4(a)]. At each beam intersection, an optical logic AND element is placed. The input bit spacings D_A and D_B are identical. The seven parallel channel AND gate outputs, indicated by dashed lines, represent the seven output digits. For example, to multiply two unsigned decimal numbers $P = A \times B = 13 \times 10$, first, each decimal number is converted into its corresponding binary format. In this case, they are $A = 1101$ and $B = 1010$. These bits are then spatially

optically encoded as light pulses to be launched into the convolution network. The number of cumulative AND output pulses in seven output channels yields the mixed-binary number convolution result 1112010. This mixed-binary number corresponds to the decimal number 130.

To include sign information, a two's complement binary (TCB) representation[17,18] can be used. In a TCB representation, an additional 0 (1) sign bit in front of the most significant bit (MSB) represents plus (minus) sign information. For a positive number, it simply places a zero in front of the MSB in its unsigned binary form. To obtain a negative number, its positive counterpart is first complemented, and then to it a one is added. For example, the signed decimal numbers $A = 11$ and $B = -14$ have as their TCB representations the numbers $A = 01011$ and $B = 10010$. It has been shown[18] that to multiply two 5-bit TCB numbers, both nine input and output channels are required. For this reason, four zeros (one) are inserted between the sign and MSB of $A(B)$. As an example, to multiply these TCB numbers, in Fig. 4(b), a SHG-based TCB multiplier is shown. The network takes inputs A and B and generates in parallel the mixed binary output $P = 332220110$. To convert this number back to its TCB format, the least significant (LSB) bit is first divided by two, and the quotient is added to the next bit and so on. The result is $P = 101100110$ representing the decimal number -154. Since this TCB multiplication network uses the lower triangular half of the previous unsigned binary multiplication network, two SHG-based implementations can be used. Either only part of the unsigned number multiplication network output (corresponding to lower triangular array) channels is used or the upper triangular half crystal is replaced by an index-matching liquid.

The previously described multiplication methods use, both for the input and output, a fixed point binary representation. Recently, based on the DMAC algorithm, an optical floating point binary multiplication scheme, also known as a flixed point number multiplication,[19] has been suggested. Based on the SHG AND gate array, next, the operation of a flixed point unsigned binary multiplier is described. Assume, as an example, the numbers to be multiplied are $A = 7/2$ and $B = 5/32$. The corresponding floating point binary representations are $A = (0.111)\,2^{+2}$ and $B = (0.101)\,2^{-2}$. For the use of 3-bit mantissas and the two exponents, $+2$ and -2, with the flixed point binary representation, each number is represented by seven binary digits, i.e., $A = 11.10000$ and $B = 00.00101$. These digits represent the parallel inputs to the previously described SHG-based unsigned binary convolver network. At the convolver output, the train of pulses $P = 0000112110000$ is generated. After setting the decimal point[19] that can be performed electronically the final result is $P = 0.011211$, representing the correct fraction 35/64. Since, with the flixed point multiplication technique, the OIAP is used for handling mantissas, the method can directly be extended to perform a signed floating-point TCB multiplication.

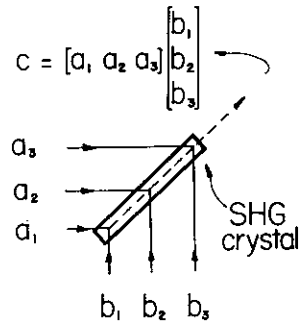


Fig. 5. OIAP 3-D vector inner product processor. Three AND gate outputs are aligned in a single output channel.

IV. Algebraic Processing Using SHG-Based OIAP Networks

The normal method to multiply two binary numbers requires three operations: a logic AND, a shift, and an arithmetic sum operations. With the DMAC scheme, the multiple shift and sum operations are bypassed by using a mixed-binary representation. The mixed-binary representation allows the successive addition of several numbers before a final A–D conversion. More complicated algebraic operations such as matrix algebra can also be decomposed into several multiplications and additions that can then be performed in parallel. For performing digital optical matrix algebra, several DMAC-based architectures have been proposed.[15–20] In this section, using an OIAP, various optical binary algebraic operations, such as vector–vector, matrix–vector, as well as matrix–matrix multiplications, are described.

Given two N-dimensional (N-D) column vectors

$$\mathbf{A} = \begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ \cdot \\ a_N \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ \cdot \\ b_N \end{bmatrix}, \tag{2}$$

the inner or scalar (dot) product of the two vectors is defined as

$$C = \mathbf{A}^T\mathbf{B} = [a_1 a_2 \ldots a_N] \begin{bmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ \cdot \\ b_N \end{bmatrix} = a_1 b_1 + a_2 b_2 + \ldots a_N b_N. \tag{3}$$

Assuming all $a_i(b_j)$, where $i, j \in (1,2,\ldots,N)$ are either a zero or one, Eq. (3) can be implemented with N number of AND gates and a summer. As an example, in Fig. 5 the inner product of two 3-D vectors is considered. Two 3-bit parallel inputs are directed to a SHG crystal. The three intersection AND gates are aligned so that their outputs can be routed into a single channel. Using a time-integrating output detector, the detected inner product result is in a mixed binary form. This result can then be converted using an A–D device to its binary form.

The vector outer product of two N-D binary vectors, **A** and **B**, is defined as the matrix

$$[C] = \mathbf{A}\mathbf{B}^T = \begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ \cdot \\ a_N \end{bmatrix} [b_1 b_2 \ldots b_N] = \begin{bmatrix} c_{11} & c_{12} & \ldots & c_{1N} \\ c_{21} & c_{22} & \ldots & c_{2N} \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ c_{N1} & c_{N2} & \cdots & c_{NN} \end{bmatrix}, \quad (4)$$

where $c_{ij} = a_i b_j$. To perform this vector outer product multiplication, N^2 AND gates are needed. In Fig. 6, a schematic network illustrating the multiplication of two 3×3 vectors is shown. To expand the input light dots into either horizontal or vertical light bars to cross and overlap at the SHG plate, two additional cylindrical lenses are employed. The SH signals emanating from the nine intersections are considered as the outer product outputs.

The matrix–vector product of a $N \times N$ binary matrix [A] and a N-D column vector **B** is defined as

$$\mathbf{D} = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1N} \\ a_{21} & a_{22} & \ldots & a_{2N} \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ \cdot \\ b_N \end{bmatrix}$$

$$= \begin{bmatrix} a_{11}b_1 + a_{12}b_2 + \ldots + a_{1N}b_N \\ a_{21}b_1 + a_{22}b_2 + \ldots + a_{2N}b_N \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ a_{N1}b_1 + a_{N2}b_2 + \ldots + a_{NN}b_N \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \cdot \\ \cdot \\ \cdot \\ d_N \end{bmatrix}. \quad (5)$$

Note that a matrix–vector product can be decomposed into several parallel vector inner product operations. In Fig. 7, by combining with an additional input cylindrical lens, three Fig. 5 type networks, a SHG-based 3-D vector optical matrix–vector multiplier is shown.

Compared to the previously described algebraic operations, an optical matrix–matrix multiplication is more complicated. As an example, consider the multiplication of two 2×2 binary matrices [A] and [B]

$$[E] = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

$$= \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix} = \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix}. \quad (6)$$

There are two methods to evaluate this matrix–matrix product. Using a vector outer product decomposition, the matrices are first decomposed into vectors and then are sequentially entered into a physical vector outer product multiplier. With a properly decomposed synchronized temporal sequence, the previously described outer product processor (see Fig. 6) can be used for the matrix–matrix product generation. Using a vector inner product decomposition in combination with an AND gate-based OIAP provides a faster optical matrix–matrix multiplier. In Fig. 8, a vector inner product-based OIAP matrix–matrix multiplier is shown. Unlike the previous examples, both a 3-D network and unequal input channel spacings are used. In this network instead of entering the data sequentially as in the case of a vector outer product multiplier,[20] a fully parallel input format is used. Because of the parallel format, it possesses an inherently faster multiplication speed. To collect the channelized multiplication results at the output an additional cylindrical lens is used.

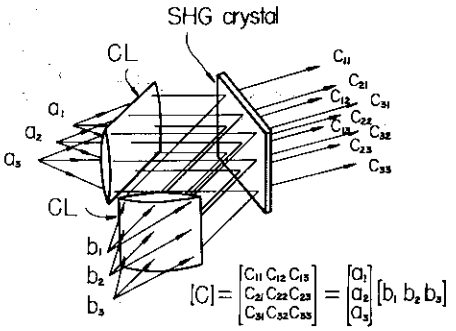


Fig. 6. OIAP 3-D vector outer product processor. In addition to nine SHG AND gates, two input prisms are used for interconnections.

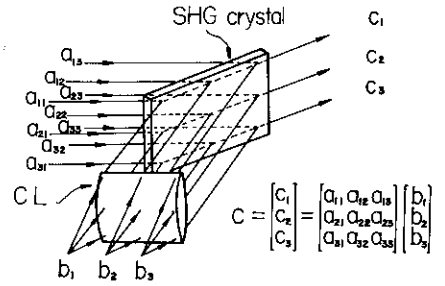


Fig. 7. OIAP matrix–vector multiplier. Three 3-D parallel vector inner product processors together with an input prism are used.

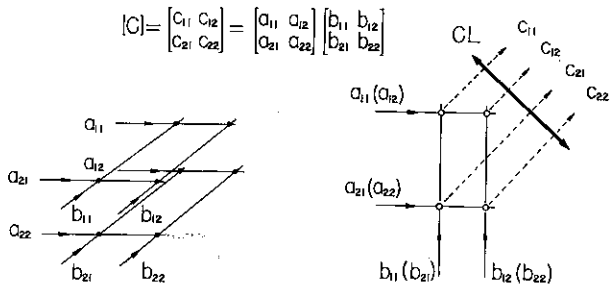


Fig. 8. OIAP matrix–matrix multiplier. Parallel input spacings for [A] and [B] are different so that four output channels (after a lens space integration) are obtained.

V. Fundamental Residue Mapping Units using OIAP

The major attraction of the residue number system is its carry-free arithmetic operation capability.[21,22] In this system, to perform numerical calculation, a set of relative prime integer bases is used. Each base forms a group of independent parallel processing units.

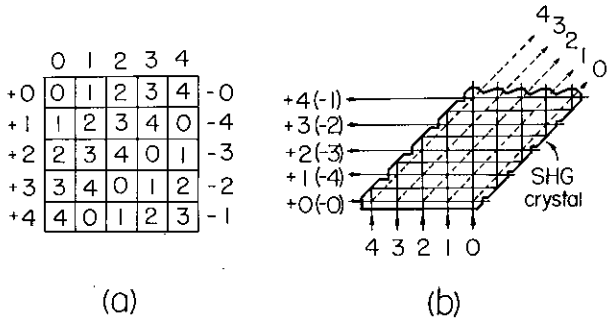


Fig. 9. (a) Mod5 residue addition (subtraction) truth table. (b) SHG-based mod5 residue optical adder implementation. At the interfaces, total internal reflections are used.

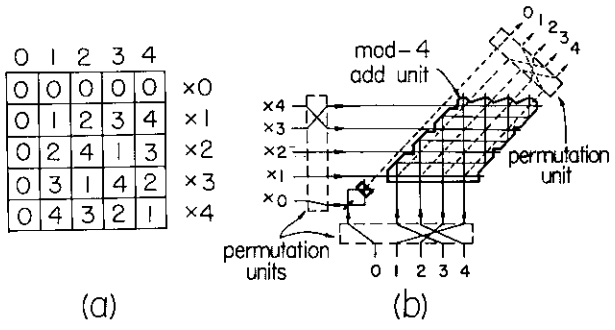


Fig. 10. (a) Mod5 residue multiplication truth table. (b) SHG-based mod5 optical multiplication implementation. Two SHG crystals, one for a mod4 addition and another to handle the zero, are shown. Also, a prepermutation and a postpermutation devices are also used.

To obtain the result in a specific modulo, in each independent residue channel, various mapping functions are performed. By suitably combining the partial results generated from each processing unit the final result is formed. Thus a specific modulo processing unit is the key element to implement a residue-based numerical processor. Using electrooptic waveguide switches, different modulo residue mapping units have been proposed.[23] Next, several SHG-based all-optical residue mapping units are described and then used to implement a residue-based matrix multiplication.

In a mod$N$ residue number system the addition is a circulant operation with the number of integral shifts determined by the value of addend. Similarly, subtraction is also a circulant operation where the subtrahend is added but in an opposite direction. To illustrate this principle, consider a mod5 addition/subtraction truth table [see Fig. 9(a)]. To implement this table, a 2-D optical mapping network must be constructed. In Fig. 9(b), using a SHG crystal, such a OIAP network is shown. For both the summand (minuend) and addend (subtrahend), pulse-position coded inputs are used. The mapping operation is controlled by the summand (minuend) signals. Thus generated signals share five output channels. The crystal is cut and oriented so that desired total internal reflections can be achieved. Here at any given time only one of the five addend (subtrahend) channels contains a signal. In other words, for a pair of input pulses, a SH output can only be generated at a single intersection. Again, using an oblique input isochronous wavefront, the inputs are autosynchronized, and thus no additional delay elements or clocks are needed.

In addition to residue addition/subtraction, using an OIAP, mod$N$ residue multiplication can also be performed. A mod5 multiplication truth table, for example, is shown in Fig. 10(a). Since the multiplication by a zero results in a zero, only operations that map the other numbers, i.e., 1, 2, 3, and 4, are necessary. For this reason, to implement a mod5 multiplication truth table, use of a mod4 adder has been suggested.[23] In general, multiplication in mod$p$, where p is a prime, can be decomposed into addition in mod $(p-1)$ with suitable prepermutation and postpermutation networks. In Fig. 10(b), using a SHG-based OIAP network, an SHG-based residue mod5 multiplication unit is shown. In addition to three input (output) permutation elements, two separate SHG units, one for implementing mod4 add and another for dealing with zeros, are employed. Although a multiplier uses more elements than a corresponding add unit, compared to other methods[23] this is a faster and more compact unit.

VI. Residue-Based OIAP for Matrix Multiplication

In many real-time scientific and engineering problems, it is necessary to solve a large number of algebraic and differential equations. For the solution of these equations, a large amount of matrix manipulations are needed. With a digital optical computer, it is essential to be able to perform fast matrix multiplication. In previous sections, using the DMAC algorithm, an AND element-based binary matrix multiplication preprocessor was described. In this section, using residue arithmetic, an alternative integer matrix–matrix multiplication approach is proposed. Using a residue number system, integer matrix multiplication can be decomposed into a set of parallel relative prime modulo-based residue matrix multiplications. Thus solving a set of linear equations using residue matrix algebra can increase computational speed. Details on the solution of integer-valued linear equations using residue matrix algebra are available.[24,25]

The multiplication of two identical prime modulo-based matrices is similar to the decimal case. In the residue case, both multiplication and addition are evaluated in the specific modulo residue system. A conventional matrix–matrix multiplier that performs the multiplication of two $N \times N$ matrices ([C] = [A][B]) contains a 2-D square array of N^2 identical processing elements, each performing an arithmetic/logic operation that adds to its past contents the multiplication results of two present inputs. For the multiplication of two mod$p$ $N \times N$ matrices, each of the N^2 cells, for example, C_{ij}, where $i,j \in (1,2,\ldots,N)$, executes recursively for $k = 1,2,\ldots,N$,

$$[C_{i,j}^{(k)}] = [C_{i,j}^{(k-1)} + a_{ik}b_{kj}] \bmod p, \tag{7}$$

where $a_{ik}(b_{kj})$ are the ikth (kjth) element of the matrix

[A]([B]). As an example, consider the multiplication of two 4 × 4 mod5 matrices [A] and [B], where

$$[A] = \begin{bmatrix} 1 & 2 & 0 & 3 \\ 4 & 1 & 0 & 2 \\ 3 & 0 & 2 & 4 \\ 2 & 2 & 1 & 0 \end{bmatrix} \qquad [B] = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 2 & 4 & 0 & 2 \\ 3 & 0 & 1 & 3 \\ 2 & 3 & 4 & 1 \end{bmatrix} \text{mod5.} \quad (8)$$

Each of the sixteen processing units performs identical arithmetic operations, i.e., mod5-based multiplications and additions. It can be shown that the corresponding matrix multiplication result is

$$[C] = [A][B] = \begin{bmatrix} 1 & 3 & 3 & 2 \\ 0 & 4 & 2 & 4 \\ 2 & 0 & 1 & 0 \\ 4 & 0 & 3 & 2 \end{bmatrix} \text{mod5.} \quad (9)$$

To implement this 4 × 4 residue matrix multiplication using an OIAP the array shown in Fig. 11(a) may be used. Input arrays A and B enter from the left and the top part of the processor. To guarantee the isochronous data arrival, a number of zero's are used. In Fig. 11(b), an individual processing element, a mod5 multiplier/accumulator, is depicted. White and black arrows represent the input and SH beams. The residue multiplier performs on the two present inputs a mod5 multiplication. Its result is first converted from a SH to a fundamental frequency and then added using a mod5 adder to the adder's previous content. The optical delay line is adjusted so that the residue adder previous content arrives isochronously with the present multiplier output. After four recursions, at each element C_{ij} the desired output is generated. Using such residue OIAPs as building blocks, an all-optical matrix residue processor may be constructed.

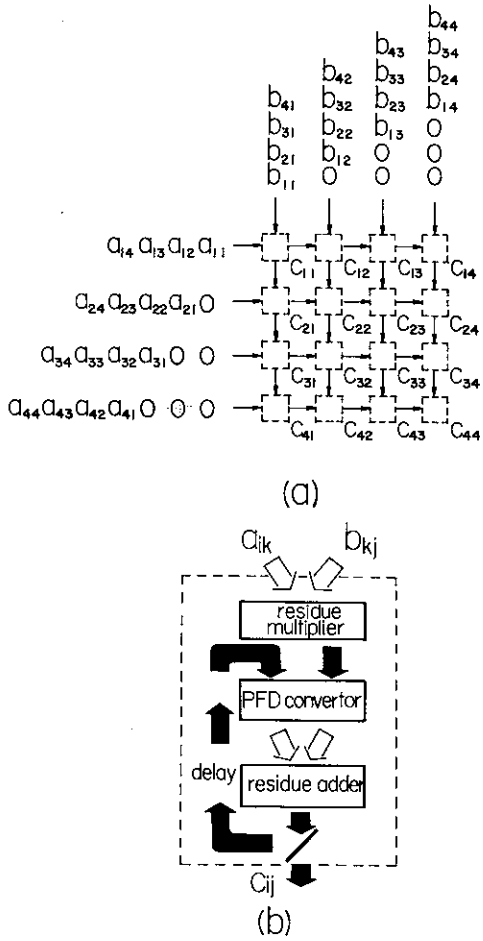
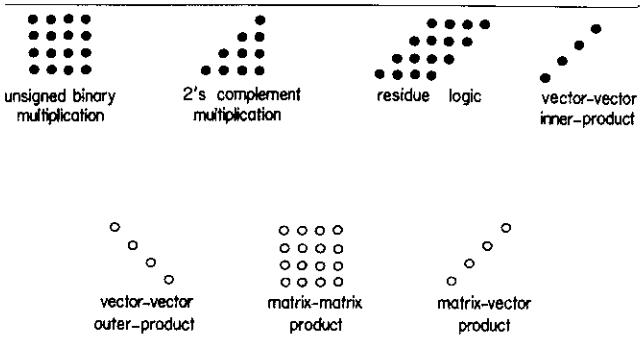


Fig. 11. (a) A 4 × 4 residue matrix–matrix multiplier with sixteen identical arithmetic processing cells C_{ij}. (b) A SHG-based C_{ij} cell that employs a mod4 multiplier and a mod4 adder. Black and white arrows represent the SH and fundamental frequency beams. In addition to the adder and multiplier, a PFD conversion device is needed.

VII. Summary and Conclusion

In this paper, various all-optical array processing methods have been introduced. For OIAP implementations, several nonlinear optical devices, such as a nonlinear etalon, Kerr gate, optical phase conjugator, and SHG device, were described. Among various proposed all-optical AND elements, the SHG AND gate is a preferred candidate. A SHG AND gate can have a femtosecond response. Except for a frequency change, a SHG-based OIAP can be monolithically implemented and thus possesses the potential for optical circuit integration. For multiple-stage SHG-based OIAP operation, using PFD techniques, the doubled frequency can be converted back to its fundamental frequency. The spatially encoded 2-D parallel data are processed in a locally interconnected lattice-type OIAP network. For digital optical arithmetic computing, a binary AND gate-based OIAP is discussed. In principle, for different applications, other elemental logic and arithmetic operators can also be used. In Table I, the use of different AND element OIAP arrays, where the dot and circle represent 2- and 3-D arrays, respectively, is summarized. Parallel data are assumed to enter from both the left and bottom of the

Table I. Summary of the Various Proposed Arithmetic OIAPs



Note: Each dot indicates an elemental processor. The black and white dots are the 2- and 3-D arrays for the different computations. Parallel data enter the processor from both the left and bottom of the diagram.

network. Using these structures with either a DMAC or a residue algorithm, various optical multipliers were described.

This work was supported in part by a grant from the U.S. Air Force Office of Scientific Research 84-0144.

References

1. A. Huang, "Architectural Considerations Involved in the Design of an Optical Digital Computer," Proc. IEEE **72,** 780 (1984).
2. D. Psaltis, D. Casasent, D. Neft, anbd M. Carlotto, "Accurate Numerical Computation by Optical Convolution," Proc. Soc. Photo-Opt. Instrum. Eng. **232,** 151 (1980).
3. S. Y. Kung, K. S. Arun, R. J. Gal-Ezer, and D. V. B. Rao, "Wavefront Array Processor: Language, Architecture, and Applications," IEEE Trans. Comput. **C-31,** 1054 (1982).
4. D. Psaltis and R. Athale, "High Accuracy Computation with Linear Analog Optical Systems: a Critical Study," Appl. Opt. **25,** 3071 (1986).
5. P. W. Smith and W. J. Tomlinson, "Bistable Optical Devices Promise Subpicosecond Switching," IEEE Spectrum (26 June 1981).
6. J. L. Jewell, Y. H. Lee, M. Warren, H. M. Gibbs, and N. Peyghambarian, "3-pJ, 82-MHz Optical Logic Gates on a Room-Temperature GaAs-AlGaAs Multiple-Quantum-Well Etalon," Appl. Phys. Lett. **46,** 918 (1985).
7. E. P. Ippen and C. V. Shank, "Picosecond Response of a High-Repetition-Rate CS_2 Optical Kerr Gate," Appl. Phys. Lett. **26,** 92 (1975).
8. T. R. O'Meara, D. M. Pepper, and I. O. White, in *Optical Phase Conjugation,* R. A. Fisher, Ed. (Academic, New York, 1983).
9. C. Kolmeder, W. Zinth, and W. Kaiser, "Second Harmonic Beam Analysis, a Sensitive Technique to Determine the Duration of Single Ultrashort Pulses," Opt. Commun. **30,** 453 (1979).
10. J. Sauteret, T. Duvillier, and A. Adolf, "Harmonic Generation with Non-collinear Laser Beams; Application to Pulse Stacking," Opt. Commun. **44,** 135 (1982).
11. Zs. Bor and B. Racz, "Group Velocity Dispersion in Prisms and its Application to Pulse Compression and Traveling-Wave Excitation," Opt. Commun. **54,** 165 (1985).
12. R. A. Baumgartner and R. L. Byer, "Optical Parametric Amplification," IEEE J. Quantum. Electron. **QE-15,** 432 (1979).
13. H. F. Taylor, M. J. Taylor, and P. W. Bauer, "Electro-optical Analog-to-Digital Conversion Using Channel Waveguide Modulators," Appl. Phys. Lett. **32,** 559 (1978).
14. R. A. Becker, C. E. Woodward, F. J. Leonberger, and R. C. Williamson, "Wide-band Electrooptic Guided-Wave Analog-to-Digital Converters," Proc. IEEE **72,** 802 (1984).
15. H. J. Caulfield, W. T. Rhodes, M. J. Foster, and S. Horvitz, "Optical Implementation of Systolic Array Processing," Opt. Commun. **40,** 86 (1981).
16. P. S. Guilfoyle, "Systolic Acousto-Optic Binary Convolver," Opt. Eng. **23,** 20 (1984).
17. R. P. Bocker, S. R. Clayton, and K. Bromley, "Electrooptical Matrix Multiplication Using the Two's Complement Arithmetic for Improved Accuracy," Appl. Opt. **22,** 2019 (1983).
18. A. P. Goutzoulis, "Systolic Time-Integrating Acoustooptic Binary Processor," Appl. Opt. **23,** 4095 (1984).
19. R. P. Bocker, M. E. Lasher, W. J. Miceli, R. H. Patterson, and B. L. Drake, "Optical Flixed-Point Arithmetic," Proc. Soc. Photo-Opt. Instrum. Eng. **564,** 150 (1985).
20. R. A. Athale and J. N. Lee, "Optical Processing Using Outer-Product Concept," Proc. IEEE **72,** 931 (1984).
21. S. L. Hurst, "Multiple Valued Logic—Its Status and its Future," IEEE Trans. Comput. **C-33,** 1160 (1984).
22. A. Huang, Y. Tsunoda, J. W. Goodman, and S. Ishihara, "Optical Computation Using Residue Arithmetic," Appl. Opt. **18,** 149 (1979).
23. A. Tai, I. Cindrich, J. R. Fineup, and C. C. Aleksoff, "Optical Residue Arithmetic Computer with Programmable Computation Modules," Appl. Opt. **18,** 2812 (1979).
24. M. Newman, "Solving Equations Exactly," J. Res. Natl. Bur. Stand. Sect. B **71,** 171 (1967).
25. P. R. Beaudet, A. P. Goutzoulis, E. C. Malarkey, and J. C. Bradley, "Residue Arithmetic Techniques for Optical Processing of Adaptive Phased Array Radars," Appl. Opt. **25,** 3097 (1986).