Concepts for numerical optical computers

I. Cindrich
A. Tai
J. R. Fienup
C. C. Aleksoff
Environmental Research Institute of Michigan
Radar and Optics Division
Box 8618
Ann Arbor, Michigan 48107

Abstract. The approach to the design of an optical computer or processor system has typically been analog in nature in the past. Recently design concepts for a numerical optical processor have evolved in which we see digital techniques implemented with optical devices. This paper describes design concepts for a numerical optical processor which is based on the residue number system.

Keywords: optical processing; digital computers; integrated optics; numerical optics.

Optical Engineering 20(4), 639-650 (July/August 1981).

CONTENTS

1. BACKGROUND

Data processor systems based on analog optical devices have been used for several special purpose applications[1-3] which include correlation and Fourier transform processing. More recently the optical implementation of numerical (digital) methods in processor system design has been under study by several investigators.[4-7] Motivation to pursue numerical optical concepts arises from at least two points of view. Given the typical need for performance advancements in processor designs, the electronic digital processor designer often seeks greater speed and packing density while the optical processor designer strives for greater accuracy and programmability. Numerical methods implemented with optical or, more accurately, hybrid optical concepts, though in an early state of study, appear to provide a potential for advancement of these system performance factors.

Numerical optical design concepts are, of course, based on handling data in a quantized and encoded form. A wide range of computing or processing operations is of interest, generally starting with basic arithmetic (+, −, ×, ÷) and building into more complex operations. Included in any study of these topics are the choices of number system and system architecture. Our interest here will be the design of optical processors which are based upon the residue number system. In this paper, we review briefly the residue number system and then go on to examples of numerical optical processor design concepts.

Invited Paper 5064 received May 27, 1980; revised manuscript received Sept. 29, 1980; accepted for publication Oct. 2, 1980; received by Managing Editor Oct. 10, 1980. This paper is a revision of Paper 185-01 which was presented at the SPIE seminar on Optical Processing Systems, May 22-23, 1979, Huntsville, AL. The paper presented there appears (unrefereed) in SPIE Proceedings Vol. 185.

2. RESIDUE NUMBER SYSTEM

The residue number system[8-11] deals with integer numbers only. Its virtue in processor design lies in its cyclic property and the fact that carry operations are not needed when performing arithmetic, thereby allowing a high degree of parallelism in the system architecture. The residue representation for a number X is given by a set of integers as

$$X = (r_1, r_2, \ldots, r_n) \tag{1}$$

Each digit r_i is the residue or least nonnegative remainder obtained when X is divided by a prescribed set of base numbers or moduli, m_1, $m_2, \ldots, m_n$, which are relatively prime integers. Illustrated in Table I is the correspondence between a decimal number X and its residue for moduli 2, 3, and 5. The available range of representation, M, is

TABLE I. Residue Number Representations

X	$r_1(m=2)$	$r_2(m=3)$	$r_3(m=5)$
0	0	0	0
1	1	1	1
2	0	2	2
3	1	0	3
4	0	1	4
5	1	2	0

limited to the product of the moduli being used, i.e.,

$$M = \prod_{i=1}^{n} m_i \,. \tag{2}$$

For moduli 5, 7, and 9, we have a range M = 315 which is equivalent to that of an eight bit binary system. Moduli 31, 32, and 33 on the other hand give M = 32,736 for a range equivalent to fifteen bits. Each residue digit is cyclic as seen in Table I. The residue representation is also cyclic over its range M which means that a number X that is larger than M will be represented modulo M.

Addition is accomplished by adding, modulo m_i, residue digits of common modulus independently (no carry). For example, numbers X and Y in residue form will be $|X|_m$ and $|Y|_m$; the sum of these is $|X|_m + |Y|_m$; and this sum reduced to residue is $||X|_m + |Y|_m|_m$. More specifically with X = 7 and Y = 3, we have for modulus 5 the residue numbers $|7|_5 = 2$ and $|3|_5 = 3$ giving $|7|_5 + |3|_5 = 5$ and the result that $||7|_5 + |3|_5|_5 = |2 + 3|_5 = 0$. This is the residue digit modulus 5 for the sum of 7 and 3. Another example is given below for moduli 2, 3, and 5 with X = 21 and Y = 4.

$$\begin{array}{r} X = (1,\ 0,\ 1) \\ +\ Y = (0,\ 1,\ 4) \\ \hline X + Y = (1,\ 1,\ 0) \end{array} \tag{3}$$

Subtraction is also performed by operating on each modulus independently. A subtraction operation may be converted into an addition operation by transforming the subtractor into its additive inverse. The additive inverse is denoted $|-X|_m$ and is defined by the following relationship:

$$||-X|_m + |X|_m|_m = 0. \tag{4}$$

For modulus seven, the additive inverse of the residue digit $|X|_7 = 5$ will be $|-X|_7 = 2$. An example of subtraction using the additive inverse for the operaton X−Y with X = 8 and Y = 3 is given below for moduli 2, 3, and 5:

$$\begin{array}{r} 8 = (0,\ 2,\ 3) \\ +\ (-3) = (1,\ 0,\ 3) \\ \hline 8 + (-3) = (1,\ 2,\ 0) \end{array} \tag{5}$$

Multiplication can be carried out in any one of several ways. One method consists of multiplication of digits having a common modulus followed by taking the residue of each product. A second approach is to simply perform successive addition of the multiplicand. The homomorphic method provides a third approach, usable only for prime moduli. The multiplication modulo m is accomplished by an addition modulo m − 1, provided that a transformation is appropriately applied before and following the addition step. The transformation is straightforward. A residue number $|X|_m$ upon being transformed will have a value K where K is defined by the relation $|X|_m = |b^K|_m$ with b a prescribed integer. The selection of the base or generator integer b is not arbitrary. It is limited to a subset of integers which works for all residues of the modulus of interest, as described under the topic of index calculus by Szabo and Tanaka.[8] Generator integers satisfying this requirement are called primitive roots. The inverse transformation entails finding $|b^K|_m$ when given K. Examples of the transformed pairs for moduli five (b = 2) and seven (b = 3) are shown in Table II. The transform for the number $|X|_m = 0$ is not defined. However this exception is not troublesome because multiplication with zero will always result in zero. Note that the transformation table for a modulus m will always have only m − 1 entries and the addition step is performed modulo m − 1.

As an example of multiplication using the log-like transformation and modulo m − 1 addition, consider the operation X × Y = Z with X = 4 and Y = 3 using the modulus 5. The forward transform is applied to X and Y, the result is added modulo 4, and then this sum is inverse transformed as shown below:

$$\begin{array}{l} |4|_5 \Rightarrow K = 2 \quad \text{and} \quad |3|_5 \Rightarrow K = 3 \\ \qquad |2+3|_4 = 1 \\ \quad 1 \Rightarrow Z = |2^1|_5 = 2\ . \end{array} \tag{6}$$

The result 2 is the residue digit of modulus 5 for the expected answer 12, i.e., $2 = |12|_5$.

General division is not readily accommodated in the residue system. However, division by an integer with remainder zero can be performed with the approach illustrated above for multiplication except that the transformed divisor is subtracted from the transformed dividend. Though division with remainder zero is of limited applicability, it is useful for the important operation of scaling. Scaling is used to bring a large number into the available range

$$M = \prod_i m_i\ .$$

The range required of the number system is established by the type of arithmetic operation to be performed. For addition, if the largest numbers to be added are N, then the range M should be at least as large as 2N. For multiplication with largest numbers N, the range needed is N^2. Division with largest dividend N requires a range no larger than N. In subtraction, what must be recognized is that a negative number does not have a sign symbol designation. Negative numbers can be handled by assigning the interval 0 to M/2 − 1 to represent positive numbers and M/2 + 1 to M − 1 to represent the negative numbers −M/2 + 1 to −1. Therefore with largest operand N, we require a range M = 2N.

TABLE II. Log-like Transformation Examples

$\lvert x\rvert_5 = \lvert 2^K\rvert_5$	K	$\lvert x\rvert_7 = \lvert 3^K\rvert_7$	K
1	0	1	0
2	1	2	2
3	3	3	1
4	2	4	4
		5	5
		6	3

2.1. Encoding

Data made available to a residue arithmetic optical processor may be in analog or digital form. Provisions for encoding the input data into residue representation will depend upon the optical devices employed in the processor design. With optical devices which are cyclic, the analog input data can be encoded to residue representation in a direct manner, whereas noncyclic devices will require a more extensive encoding step. When encoding is needed, one approach is to convert the analog input to an intermediate binary form and then proceed with residue encoding of the binary data. For example, conversion of the binary number 110 for modulus 3 proceeds as follows:

$$110_3 = ||1 \times 2^2|_3 + |1 \times 2^1|_3 + |0 \times 2^0|_3|_3 = 1\ . \tag{7}$$

2.2. Decoding

Conversion of a residue number to a weighted number system representation, such as the decimal numbers, is a more complex operation than encoding. Residue to mixed radix conversion is often used and has the advantage that familiar residue arithmetic operations are employed. The algorithm for this decoding is described with the flow diagram of Fig. 1 for a residue number of four digits (r_1, r_2, r_3, r_4) corresponding to moduli 2, 3, 5, and 7. The operations involved are a sequence of subtractions and multiplications which result in coefficients of the equivalent mixed radix number (a_1, $a_2 \ldots a_N$). For this example, the decoded result would be $X = a_1 + a_2(2) + a_3(2 \times 3) + a_4(2 \times 3 \times 5)$. Multiplication in this procedure is performed using the multiplicative inverse denoted

$$|1/K|_{m_i}$$

which is defined by the relation

$$|K \times |1/K|_{m_i}|_{m_i} = 1\ .$$

Conversion from residue to mixed radix form is useful not only for decoding but also for the important operations of sign detection, magnitude comparison and overflow detection. The conversion from mixed radix back to the residue system is straightforward. Consider starting with the mixed radix representation $X = (a_1, a_2, a_3, a_4)$ for

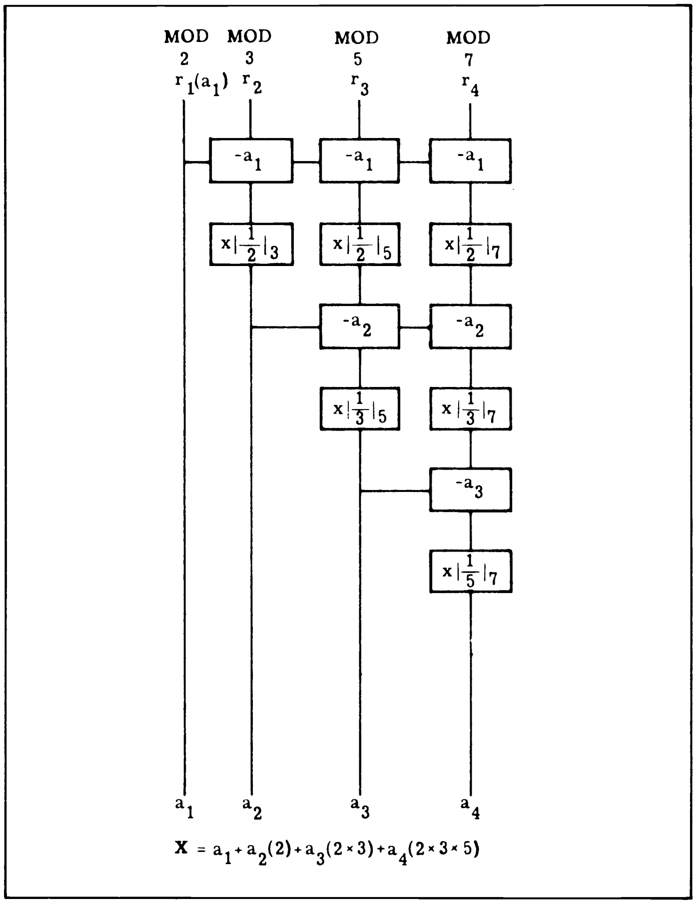

Fig. 1. Residue to mixed radix conversion.

moduli 2, 3, 5, and 7. The residue digit r for modulus m_i is obtained with the following expression:

$$r = |a_1|1|_{m_i} + a_2|1\times 2|_{m_i} + a_3|1\times 2\times 3|_{m_i} + a_4|1\times 2\times 3\times 5|_{m_i}|_{m_i} \tag{8}$$

2.3. Scaling

Overflow may be a more demanding consideration in residue arithmetic than in conventional arithmetic which utilizes a weighted number system. Detection of overflow is not as automatic as with weighted number systems and it is desirable to avoid occurrence of overflow.[12-14] This would require a periodic down-scaling of the residue numbers. To do this, division operations would be necessary. As pointed out earlier, general division cannot be carried out easily and scaling by an arbitrary factor would not be practical. One can, however, scale a residue number by a factor equal to the value of one of the moduli or a product of two or more moduli. For example, for a system with moduli 2, 3, 5, and 7, if we are to scale down a number X = 191 = (1, 2, 1, 2) by a factor of 7, we proceed as follows: since the divisor 7 is also a modulus, the corresponding residue $|X|_7 = 2$ would be equal to the remainder when the number X is divided by 7. Therefore, $X - |X|_7$ is always divisible by 7 and the homomorphic approach can be applied for the division operation. However, for modulus 7, the divisor is equal to 0 and division by 0 is not defined. The general approach is to proceed with the division while ignoring modulus 7. The residue of the quotient for modulus 7 is then obtained using the extension of base procedure[8] which is essentially a residue to mixed radix conversion.

With some of the basic properties of residue number systems in hand, we go on in the following sections to describe examples of optical implementation concepts.

3. BASIS FOR NUMERICAL METHODS

Fundamental to the optical implementation of a numerical processor is the use of devices which provide numerical control of a light beam or wave. Phase, polarization, position, and intensity of a light beam are the physical properties which may be considered for implementation.

Consider the control of light wave phase as a light beam passes through an electro-optic modulator depicted functionally in Fig. 2.

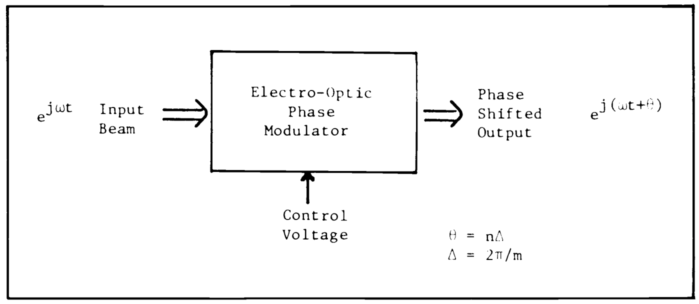

Fig. 2. Cyclic phase operation.

Since the phase of the light wave is inherently cyclic modulo 2π, then by providing control of the phase shift in increments of Δ, with $\Delta = 2\pi/m$ and m the desired residue modulus, the phase of the emergent light wave will serve as a residue number representation. For example, for modulus 5, we make $\Delta = 2\pi/5$. As the phase is changed incrementally, we progress through Δ, 2Δ, 3Δ, and 4Δ and then start to repeat modulo 2π since we have equivalence between 0 and 5Δ, 2Δ and 6Δ, etc. With an input (control voltage) that is continuous rather than quantized, the optical phase shift modulator device may be designed to provide a quantized response, analogous to approaches under development with polarization modulation schemes.[5,15] Otherwise, quantization must be provided at the point of detection or in the applied control voltage itself.

In addition to electro-optic devices, others are available based on acousto-optics, thermo-optics, and material deformation (optical path length modulation) for the control of the phase of a light wave.

Rather than altering the phase of light wave, the polarization angle which is also inherently cyclic can be used for residue number representation.[4,15] Choice of approaches for realizing quantized control is similar to that discussed above for phase control.

Position control of a light beam, or mapping, for residue number representation is depicted in Fig. 3. In this example, optical switches steer (or deflect) the input beam into selectable exit paths. Each of the six optical switches has two output ports which are selectable by a control voltage. The switches either pass the entering light beam undeflected or steer the beam to an alternate path. Any one of the input positions can be switched to any of the three output positions using simple switching logic in which the switches are activated a row at a time. A fixed set of light beam paths can also be useful where the programmability provided by optical switches is not needed. The light beam paths may be open or confined (e.g., optical wave guides, fibers, stacked diffraction gratings, etc.). Switching devices such as optical wave-guide couplers, acousto-optic diffraction cells and fiber optics couplers may be used for position or path control. The beam paths are discrete and the switches are generally controlled with binary control signals.

Instead of individual two state switches for beam position control, devices having multiple output positions may also be used. Some acousto-optic beam deflector designs, for example, can provide at least 10^3 discernible output positions which might serve in place of a set of two-position optical switches.

Use of the intensity level of a light beam for residue number representation may be realized by the direct control of a light source or by external control of the emitted beam. External control can be achieved by analyzing a light wave whose phase or polarization angle has been modulated as noted above. Direct control of the light source is also possible (laser, light emitting diode (LED), etc.). However, in

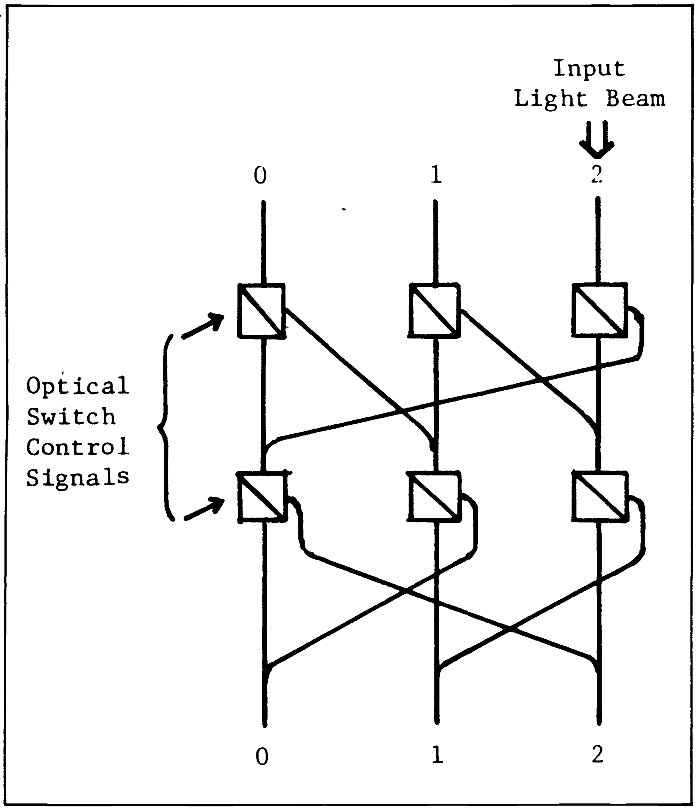

Fig. 3. Light beam position mapping for modulus 3 programmable operation.

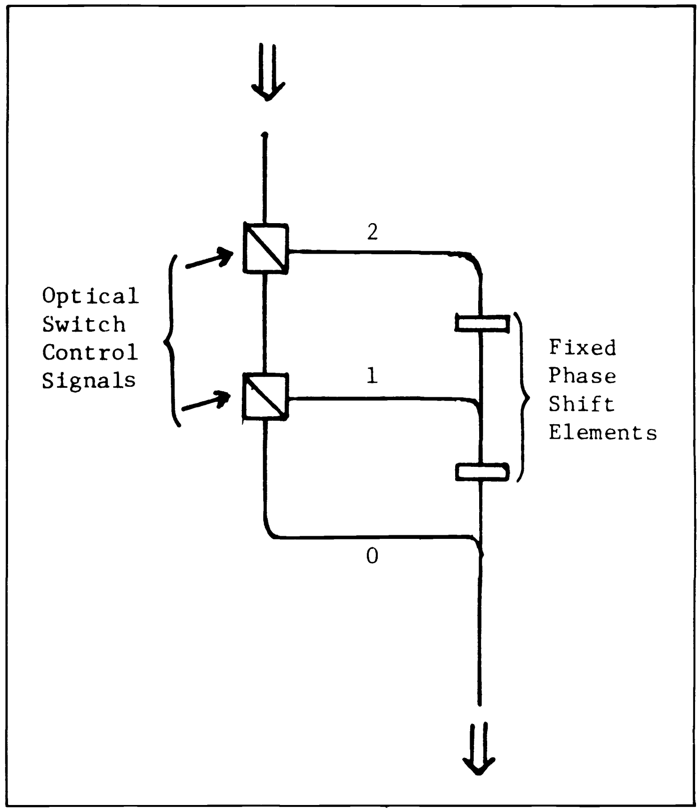

Fig. 4. Mapping and phase control combined for modulus 3 operation.

this case, the cyclic property is not inherent and must be provided by the control signal itself.

Combined use of more than one property of a light beam is also possible. As an example, phase and position control are shown in Fig. 4. This approach provides for residue number representation as an incremental phase shift as mentioned earlier. However, for this case, phase shift elements of a fixed and passive type are used with their selection made by choice of beam path. Instead of the phase shift elements of Fig. 4, the use of optical attenuation elements would provide beam intensity control while the use of polarization rotator elements would give the residue representation as polarization angles.

Recall that the range of numerical representation in residue form has two salient range properties, the range of individual moduli m_1, $m_2, \ldots, m_n$ and the total range M. In the case of an optical phase shift device, for example, the phase modulator must have a modulation capability over the full range M if the modulator control voltage is to be a continuous variable. If, on the other hand, the modulator control voltage is encoded to correspond to the range of an individual modulus, then the phase shift modulator device need only respond over the more limited range of that modulus.

Having introduced the notion of quantized and cyclic control of light beam properties for residue number representation, we present in the next section examples of numerical optical processor design concepts.

4. PROCESSOR DESIGN CONCEPTS

The design of a numerical optical processor can vary considerably with choice of optical hardware, computer architecture, and interface and control assumptions. We consider in this section two approaches for basic computing tasks which can be extended to more complex calculations. The approaches differ in the choice of the physical property of a light beam that is being used for residue number representation. One approach employs phase control of a light wave with its inherent cyclic property and the other utilizes light beam position control or mapping.

4.1. Cyclic phase implementation

As an example of cyclic device, we take the case of spatial phase modulation which can be implemented with such devices as acousto-optic spatial phase modulators.[16] We start with the basic set of components shown in Fig. 5 which consists of two modulators and the means for introducing collimated light waves into each of them. The collimated beams 1 and 2 originate from a laser diode light source directed through a collimating lens and a beam splitter grating G. This arrangement serves as an interferometer which provides an interference or fringe pattern at its output.[17] It will operate with light sources of modest coherence. The spatial frequency (carrier) in the modulators is twice the grating frequency in G. If the modulators

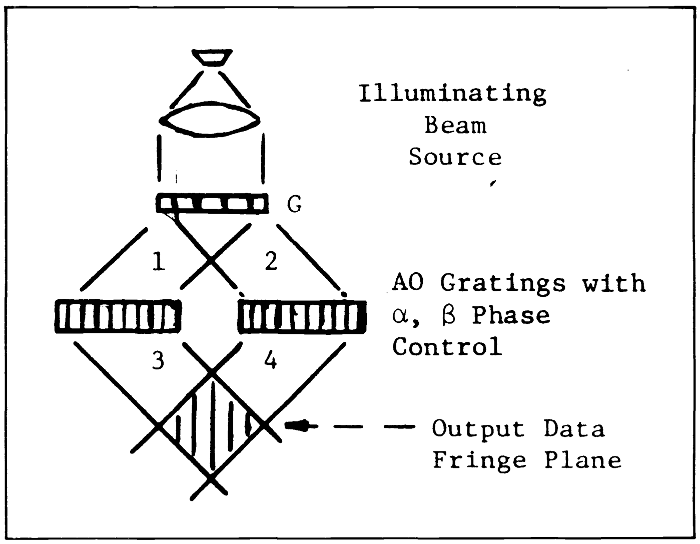

Fig. 5. Cyclic phase operation with a grating interferometer using acousto-optic gratings.

have a sinusoidal spatial modulation of optical index along their length (X-dimension) of the form $\cos(\omega x + \alpha)$ and $\cos(\omega x + \beta)$, then the diffracted first order light waves 3 and 4 can be written as

$$e^{-j(\omega x + \alpha)} \text{ and } e^{+j(\omega x + \beta)} \; . \tag{9}$$

Assuming for convenience that these waves have unity amplitude, then the interference or fringe pattern at the detector plane will be of the form

$$1 + \cos\left[\omega x + (\alpha + \beta)\right] \; . \tag{10}$$

Thus the phase of the fringe pattern output is the sum of the input or modulator phases α and β. Phases α and β would be entered into the acousto-optic modulators as equivalent residue numbers for a particular modulus m_i. Since the accumulated output phase is cyclic, the output sum will have the desired residue property of being cyclic modulo m_i.

Rather than using a detector at the output, two other possibilities exist. An optical transducer or memory may be used at the output plane which records the sinusoidal fringe pattern and then acts as a diffraction grating containing the output phase $(\alpha + \beta)$. When illuminated, it would serve to readout the summation data $(\alpha + \beta)$ as the phase of the diffracted output beams for use in another cascaded computing element possibly of the same type. A second possibility for handling the output avoids the use of a detector or a transducer by simply allowing the output waves 3 and 4 to continue and become inputs to another set of acousto-optic modulator elements as shown in Fig. 6. This figure shows a succession of such cascaded modulators. The phase of fringe pattern output for the set would be the accumulated sum $\sum(\alpha_i + \beta_i)$, i.e., the output fringe pattern is of the form

$$\cos\left[\omega X + \sum_{i}^{N} (\alpha_i + \beta_i)\right] \; . \tag{11}$$

With this device, we can realize a computing module capable of addition, subtraction, and multiplication (through successive addition). For addition or subtraction of two residue numbers, we need only two modulators. With multiplication, the number of modulator elements in this type of computing module must equal the largest multiplier which, for modulus m_i, will be $m_i - 1$. Multiplying two numbers $X \times Y$ is done by entering a phase $\alpha = \beta = X$ in all modulator elements and having the total number of modulators equal to Y.

Considerable reduction in complexity of the multiply operation could be realized by use of the homomorphic approach for multiplication. Recall that with this method a log-like transformation of the multiplier and multiplicand is required followed by modulo $m_i - 1$ addition and then an inverse transformation to determine the answer. However, convenient methods for homomorphic multiplication using the cyclic phase approach have not yet been devised. Such a transformation is readily accomplished with a mapping implementation approach using, for example, light beam position control.

The time required to perform the single summation $\alpha + \beta$ is quite short, being simply the propagation time from the acousto-optic (AO) cell to the output fringe detection plane, which would be a few picoseconds for small integrated optic configurations. Clearly, however, overall cycle time of such a unit is the characteristic of importance and it is comprised of the set-time of an AO device, the light propagation time noted before, and the output fringe phase detection time. At the present state of the art, the AO cell set-time capability is about 0.1 to 10 μsec which is quite modest for computing operation of interest here. Another design concern with this approach at present is the combined speed and accuracy achievable in performing electronic detection of the phase of the output fringe pattern.

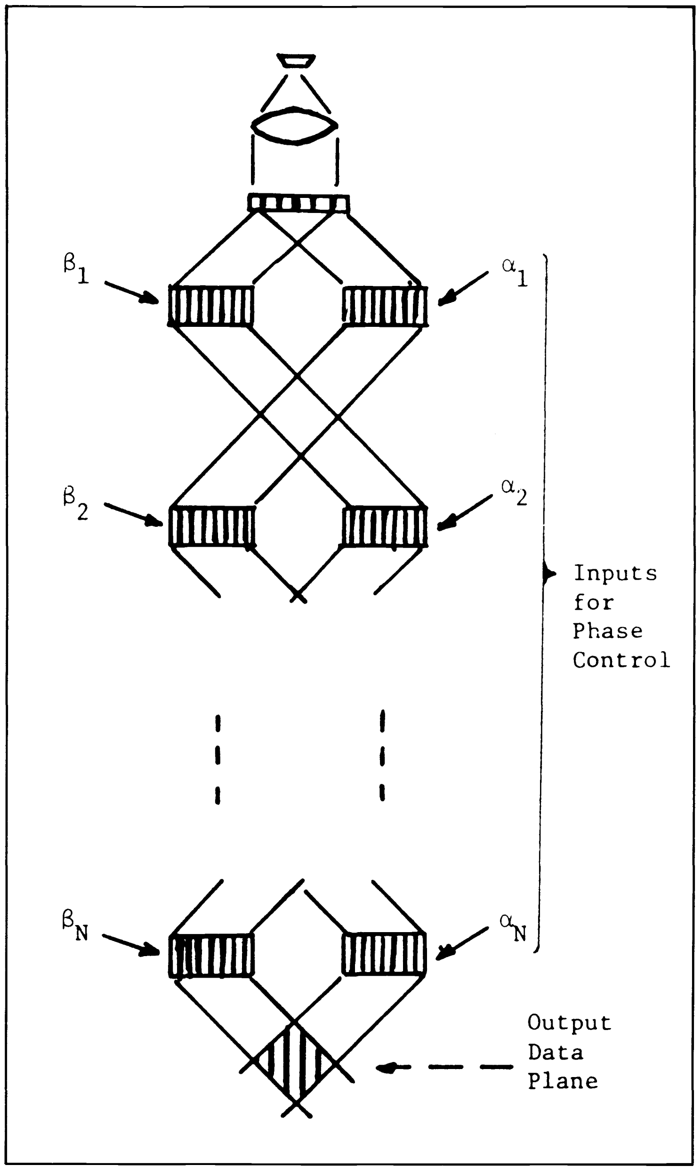

Fig. 6. Cyclic phase operation with cascaded acousto-optic elements for 2N inputs.

4.2. Mapping with beam position control

Position control or mapping of a light-beam path provides an attractive means for residue number representation and computing operations. At the present state of hardware development, mapping appears to offer a more versatile approach to numerical optical processor design than is available with cyclic phase or polarization devices. This is due mainly to the lack of fully developed methods for quantization and the comparative complexity of performing such operations as multiplication with cyclic devices which employ phase or polarization control. In this section, we describe a programmable computing module[6] for basic arithmetic which uses light beam position mapping. The design concept offers considerable versatility in use and a good potential for interconnecting a large set of such modules for more complex computing operations.

To demonstrate the design concept, we will use directional coupler wave guide switches for the implementation of beam path control. The directional coupler is one of the better-developed integrated optical devices and it allows flexibility in optical circuit design.[18-20] We shall briefly describe the optical coupler wave guide switch and then proceed to formulate its use in a basic addition operation which will be extended into a multipurpose arithmetic module. It should be noted that there are other components that

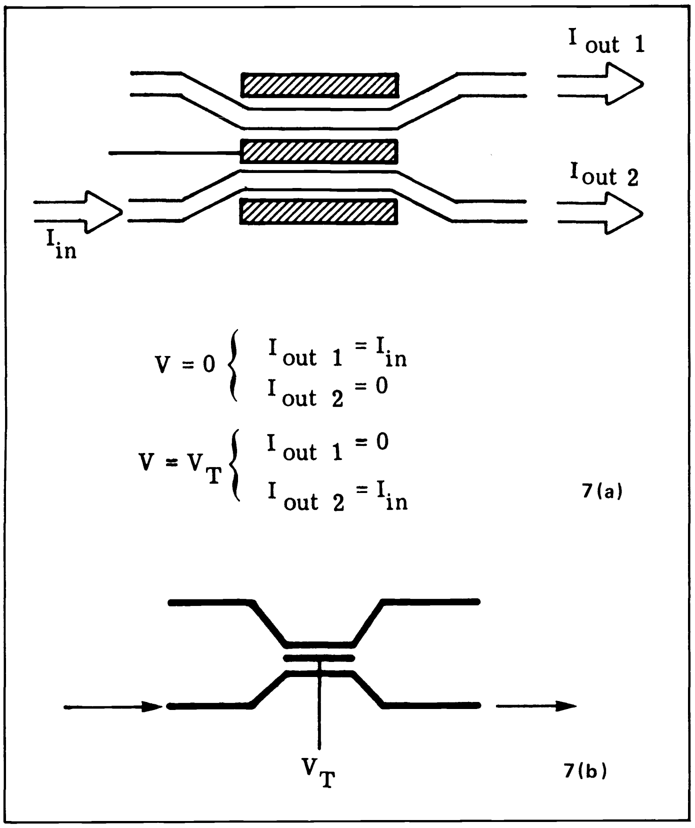

Fig. 7. (a) Directional coupler wave guide switch (not to scale) (b) schematic representation.

would be good candidates for the implementation of a mapping configuration. The wave guide coupler is but one example.

A directional coupler is schematically shown in Fig. 7. Two wave guides are placed physically close to each other such that, in the absence of an applied electric field, the wave guides are synchronous. That is, a light wave propagating in one wave guide will be coupled to the adjacent one producing a switch in light path.[18-20] When an appropriate voltage V_T is applied to the electrode, the synchronism between the wave guides is broken and the light propagation will remain in the wave guide originally excited as illustrated in Fig. 7(a) For simplicity, the coupler wave guide switch from here on will be represented as shown in Fig. 7(b).

Using the wave guide switches, one possible implementation of a modulo 5 adder is shown in Fig. 8. With this design, the electrode voltages of all of the coupler wave guide switches are initially set at V_T. The light wave injected into the input of the adder will therefore propagate inside the same wave guide through the adder. To program the device for the +2 operation, for example, the electrode voltage of the corresponding row of couplers is changed to 0. Thus, when the light propagation reaches that particular set of coupler wave guide switches, the light wave will be coupled into the adjacent wave guide, changing the optical path. The electrode voltages are maintained at constant levels of V_T or 0 by connecting the electrodes to a set of S-R flip flops. The adder can be programmed by sending an electric pulse to the "S" input of the appropriate flip flop, triggering it to change state. Alternatively, we could let the initial electrode voltage of all the couplers be 0 and program the adder by changing the electrode voltage of a particular row of coupler switches to V_T. However, we generally find that it is easier to trace the light path with the former design, and, to make the devices easier to study, we shall make use of the former design in this paper. We shall also use the term "on" to describe the state where coupling occurs at the coupler switch and term "off" for the state where the light propagation will remain in the same wave guide.

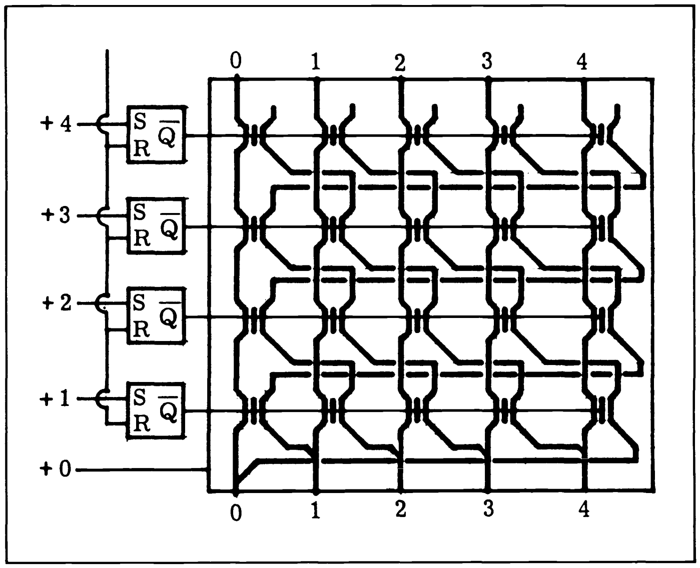

Fig. 8. Implementation of modulo 5 adder (not to scale).

Subtraction can be performed with the use of the additive inverse as described previously. There is a fixed one-to-one correspondence between a residue number and its additive inverse. The additive inverse transformation can therefore be implemented by a fixed map. And by adding this transformation map to an adder, one can convert it into a subtractor as shown in Fig. 9 for modulus 5.

Multiplication can be implemented directly by using fixed maps (m_i of them) for the operations of $\times 0$, $\times 1$, $\times 2, \ldots, \times(m_i - 1)$. Alternatively, one can make use of a homomorphic approach where a modulo m_i multiplication is converted into a modulo $m_i - 1$ additive operation. A $\log_b$ K-like forward transform is first per-

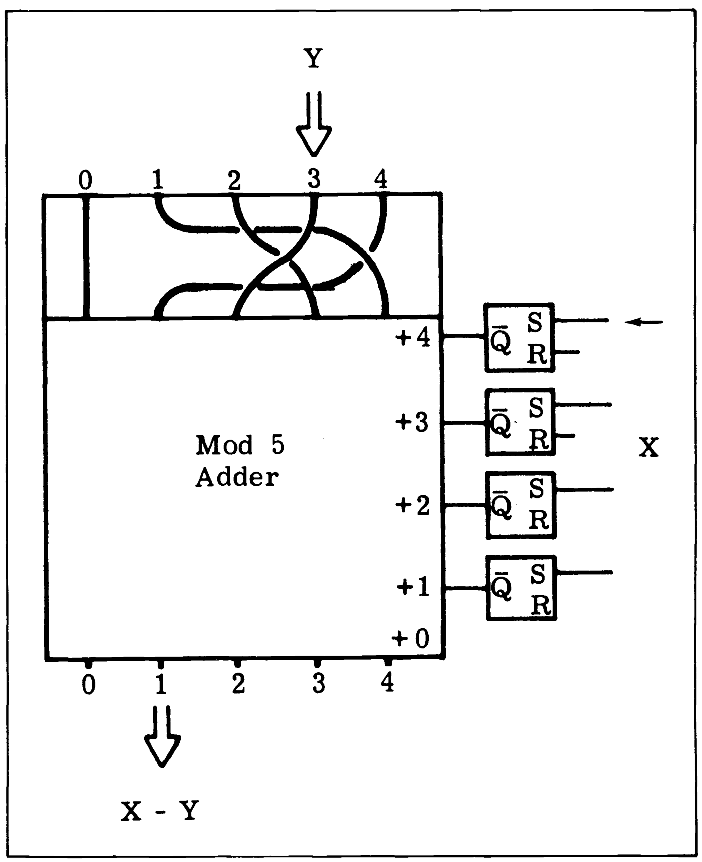

Fig. 9. Converting an adder for subtraction operation.

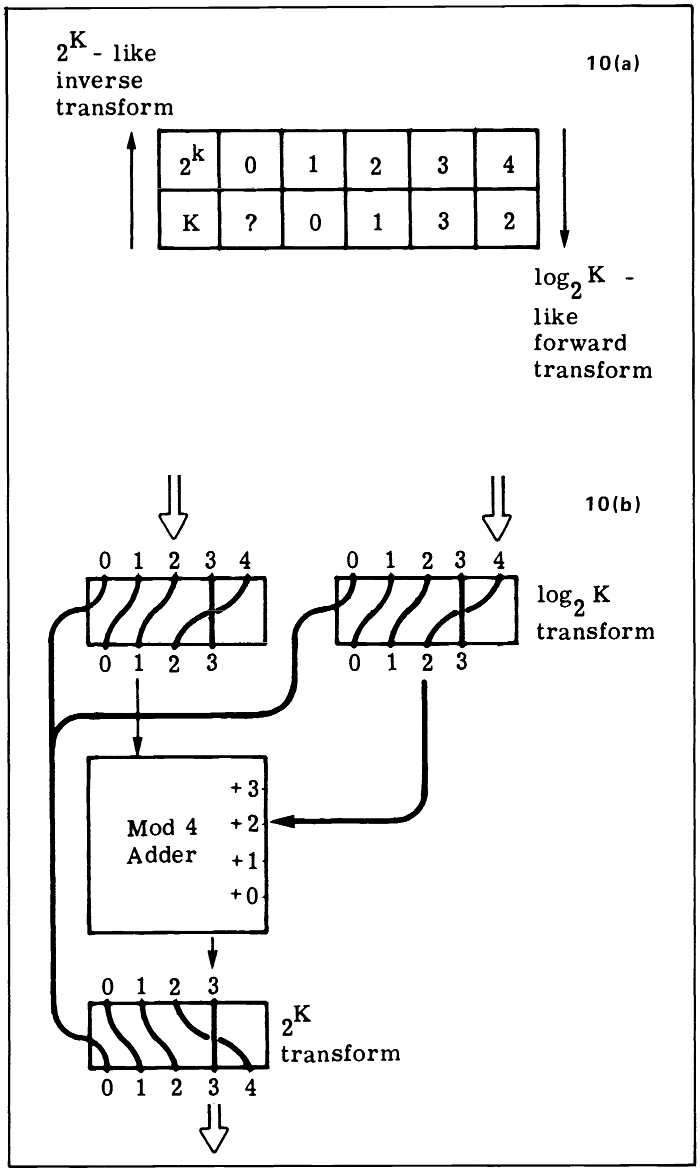

Fig. 10. (a) Transform table for modulus 5 and (b) modulo 5 multiplication using the homographic approach.

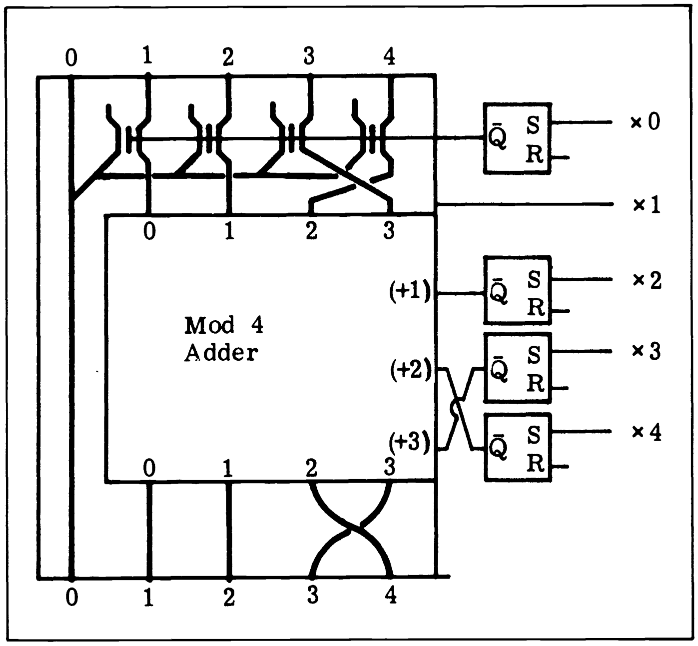

Fig. 11. Implementation of a modulo 5 multiplier.

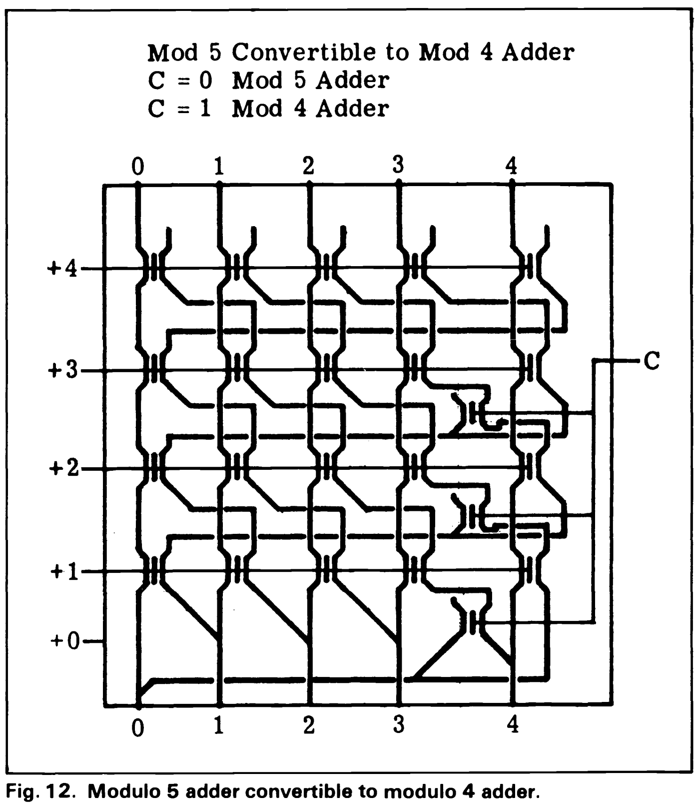

Fig. 12. Modulo 5 adder convertible to modulo 4 adder.

formed on the operands. A modulo $m_i - 1$ addition is then performed and the sum is inverse transformed by a b^K-like transform to obtain the product of the two original numbers. The transform table for modulus 5 and the process is illustrated schematically in Fig. 10. Although the $\log_b$ K-like transformation for the value 0 is not defined, it is known that if either the multiplier or the multiplicand is 0, the product is 0. A modulo 5 multiplier is shown in Fig. 11 using this homomorphic approach. We note that for a modulo 5 multiplication, a modulo 4 addition is performed. Thus, in order to convert a modulo 5 adder into a modulo 5 multiplier, the modulo 5 adder should be designed in such a way that it can be easily converted into a modulo 4 adder. This can be achieved with the design shown in Fig. 12. While the concept can be applied to an adder of any modulus, we should note that this homomorphic approach can be used only if the modulus is prime.

One feature of this design is that the input, output, and programming controls are all represented spatially in the same way. This allows the interconnection of these devices for sequential operations. The outputs of one module can be connected directly to the inputs of the next module or it can be used to program the map of the next adder as illustrated in Fig. 13. An electrical pulse is sent to the first multiplier to program it to perform

$$x|X|_{m_i}.$$

A light pulse is then injected into the adder at the spatial position corresponding to

$$|Y|_{m_i}.$$

The exit position of the light beam would correspond to

$$|X \times Y|_{m_i}.$$

A fast avalanche photodiode is connected to each of the output wave guides. The exiting light pulse will be detected by the photodiode,

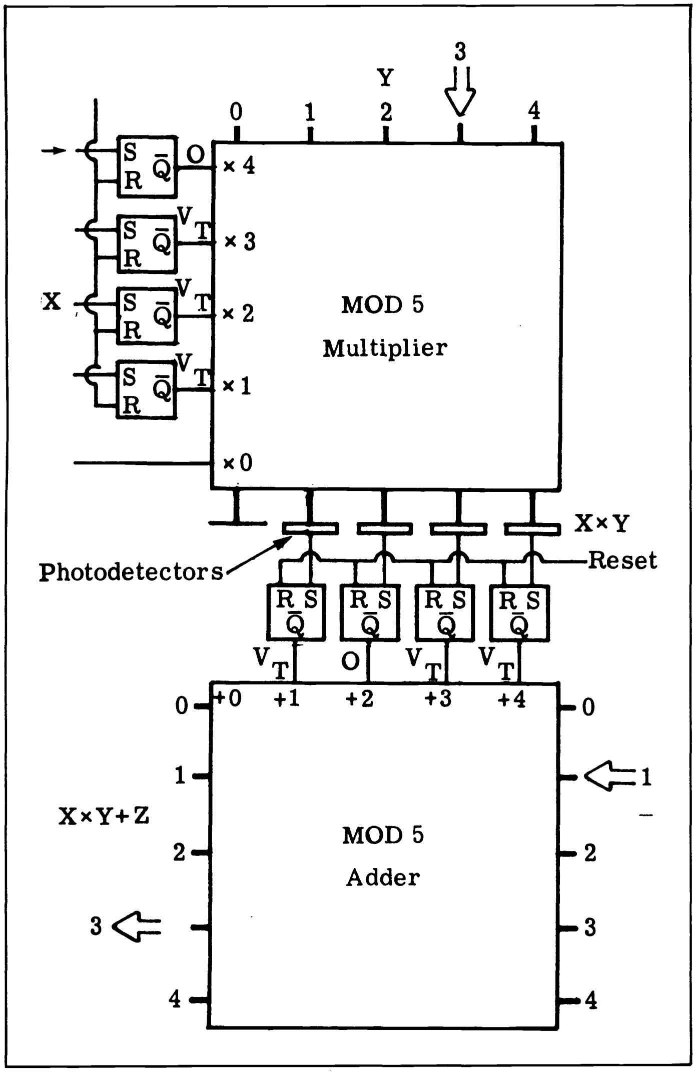

Fig. 13. Interconnection of modulo 5 computation modules.

generating an electric pulse. The electric pulse in turn triggers the corresponding flip flop of the next adder, setting it for the

$+|X \times Y|_{m_i}$

operation. Another light pulse is then injected into the input of the second adder at the position corresponding to

$|Z|_{m_i}$.

The position where the light pulse exits will represent

$|X \times Y + Z|_{m_i}$.

4.3. Multipurpose arithmetic module

With the subunits described above, we can proceed to describe a multipurpose programmable computation module. The module will contain four distinct parts as shown in Fig. 14. Each of these subunits can be turned on and off individually, allowing the different combinations of the subunits to perform various computation operations. However, it is more complicated than simply stacking all the subunits together. Special attention must be paid to the case of $+0$ and $\times 0$ by noting that $X + 0 = X$, $X \cdot 0 = 0$, $0 \cdot Y = 0$, and $X \cdot 1 = X$. Furthermore, the modulus m_i adder must be modified to perform modulo $m_i - 1$ addition and the

$|-K|_{m_i}$

additive inverse transform must be converted into a

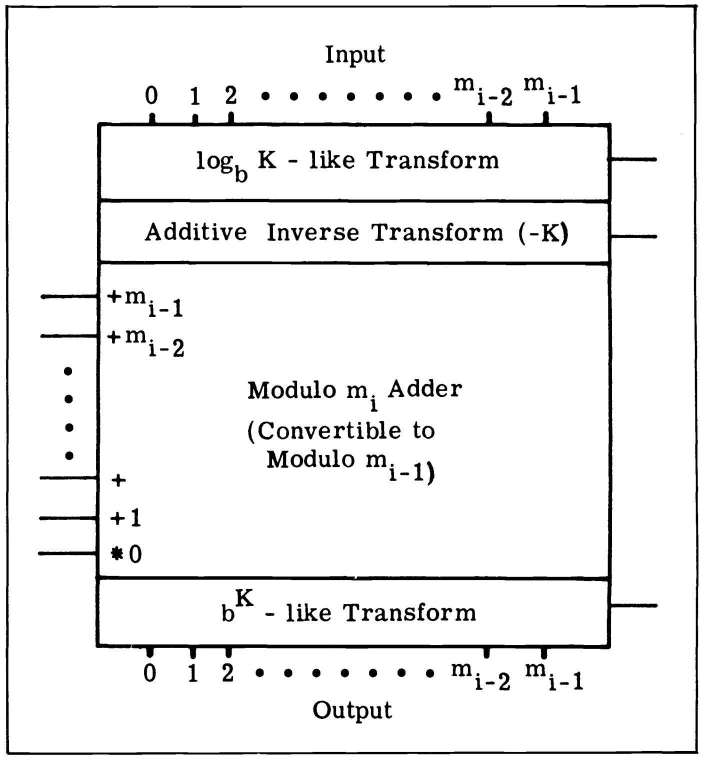

Fig. 14. Conceptual design of programmable multipurpose computation module.

$|-K|_{m_i-1}$

transform when the module is programmed to perform multiplication and division. A possible design of the programmable multipurpose computation module is shown in Fig. 15.

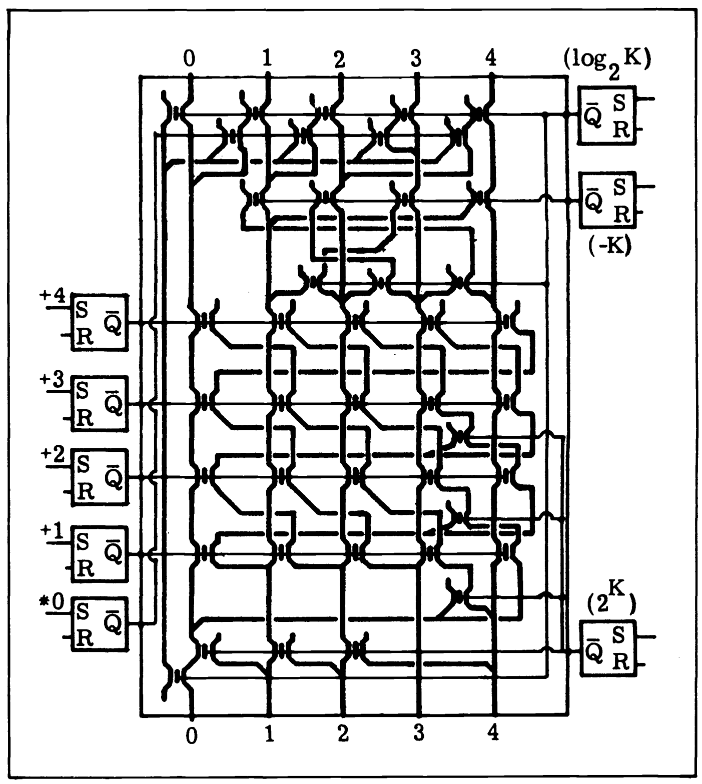

Fig. 15. Implementation of programmable multipurpose computation module.

The multipurpose computation module can be programmed to perform +, −, ×, and ÷ arithmetic operations with simple binary controls. For example, to perform modulo 5 addition, the subunits for $\log_2$ K-like transform, additive inverse transform, and 2^K-like transform are all turned "off." That is, a light pulse injected into any of the m_i input ports will propagate undeviated along the same wave guide through these subunits. With these units "off," the module would be essentially the simple adder shown earlier in Fig. 8. To perform subtraction, the additive inverse transform

$|-K|_{m_i}$

unit is turned "on," changing the light path according to the transform map shown in Fig. 9. We note that while operating in the addition and subtraction modes with the $\log_2$ K-like transform unit off, an input to the "*0" control has no effect on the light path. The position of the exit beam would, therefore, be the same as that of the input beam, performing, in effect, the +0 operation. The programming of the computation module for addition and subtraction operations is illustrated in Figs. 16(a) and 16(b).

In programming the computation module for multiplication, there are two possible approaches. The module can be connected as a multiplier by rerouting the electrode leads to perform the $\log_2$ K-like transform on the multiplier value (X).The 2^K-like transform unit is turned on to inverse transform the sum as illustrated in Fig. 16(c). With the second approach, both the multiplier (X) and the multiplicand (Y) values are transformed by computation modules, as illustrated in Fig. 16(d). This approach has two advantages. First, the connection of the electrode leads does not have to be changed, allowing the module to be switched back to addition mode when desired. Second, it provides more flexibility in performing division. Observe that the extra coupler switch at the left lower corner in Fig. 15 is necessary for the module to be programmed in this mode. The coupler is turned on together with the $\log_2$ K-like transform unit at the top. When the value of the multiplier X is 0, the "*0" control of the second module is turned on, and the ×0 operation is performed. If the multiplier is 1, its $\log_2$ K-like transform is 0; the purpose of the extra coupler switch is to keep the transformed 0 output of the multiplier from setting the *0 control of the second module. Instead, the coupler switches the light path away from the 0 output port such that the second module would be left undisturbed. The light pulse will exit at the same position as it enters the module, performing the ×1 operation.

The programming of the computation module for the division operation is illustrated in Fig. 16(e). An

$|-K|_{m_i-1}$

additive inverse transform is required for the divisor after the $\log_2$ K-like transform. A

$|-K|_{m_i}$

transform can be changed into a

$|-K|_{m_i-1}$

transform by shifting down the values of the

$|-K|_{m_i}$

transform by 1. Referring back to the module design shown in Fig. 15, the down shifting is performed by the set of three switches at the fourth row. They are turned on together with the $\log_2$ K-transform unit.

There is a very useful feature in the use of computation modules for residue arithmetic that is shared by other implementations using the mapping approach. The operand and the operator are combined in a single representation. For example, to perform addition between an input value and a stored value with a conventional computer, the stored value has to be recalled from storage and entered with the input value into a fixed operator (adder). Implementing residue arithmetic with the computation modules described here allows the stored values to be entered into the module as operators (i.e., +K). The state of the module represents both the operand (K) and the operation (+). The module is therefore functioning simultaneously as the adder and the data storage device. This feature eliminates the need of a separate memory for values such as the coefficients of a reference function in correlation detection operations. Without the access time delay in reading out the stored values, the inputs can be processed at a very high rate, especially for computations that have to be performed repeatedly.

Implementation methods for computing modules of a numerical optical processor which are useful for basic arithmetic operations have been described thus far. In the following section, we discuss the application of the computing module to more extensive computing operations.

5. EXTENDED COMPUTING APPLICATIONS

Extension of the basic numerical optical methods described in previous sections to a broadened range of computing problems will be introduced in this section. We will use mapping of the light beam position as a basis for our discussion; however, it should be kept in mind that a variety of other hardware design concepts might also be considered.

5.1. Polynomial evaluation

To demonstrate how the mapping computation module (described previously) can be interconnected to perform various mathematical calculations, we consider first the evaluation of polynomials. As discussed by Huang et al.,[4] a polynomial may be evaluated using a single fixed map. However, considerable flexibility is achieved if a programmable design is used which can be realized with a collection of basic arithmetic modules and fixed maps. We start with a set of fixed maps for X^n, X^{n-1}, .. X^2 functions which are combined with the computation modules as shown in Fig. 17. To program the modules for the computation of $X^3 + 4X^2 + 3X + 2$, for example, the coefficients 1, 4, and 3 are entered into the multipliers. Light pulses are injected into the inputs of the multipliers at the ports corresponding to the value of input X. The adders would be set by the output of the multipliers for $+(X^3)$, $+(4X^2)$, and $+(3X)$ operations. Another light pulse is then entered into the first adder at input port 2, and the position where the light pulse exits would correspond to the value of $X^3 + 4X^2 + 3X + 2$.

The computation time would be equal to the time needed to set the adder module plus the propagation time through four modules. The propagation time through a single module of 1/2 inch size would be about 50 psec. The set time of the module is the sum of the detection delay of the photodiode, the switching delay of the flip flop, and the switching time of the wave guide coupler. It is possible to achieve a set time under 1.5 nsec for the computation module.[11-18] And if we assume that an additional 1.5 nsec is required for the light pulse to pass through the module and to reset the flip flops, the throughput rate would be about 1/3.2 nsec = 312.5 MHz. Due to the parallelism of the arrangement, the computation time is approximately the same for polynomials of any order.

5.2. Matrix multiplication and transforms

One of the important potential applications of the numerical optical computer is the multiplication of matrices. It can be extended to a number of transform operations such as discrete Fourier transform (DFT), Hadamard transform, etc. We shall examine the general case of matrix multiplication,

$$A_{M\times N}\, B_{N\times P} = C_{M\times P}\,.$$

The coefficients of the master matrix

$B_{N\times P}$

are stored in the modules as multipliers as shown in Fig. 18. The values of the matrix

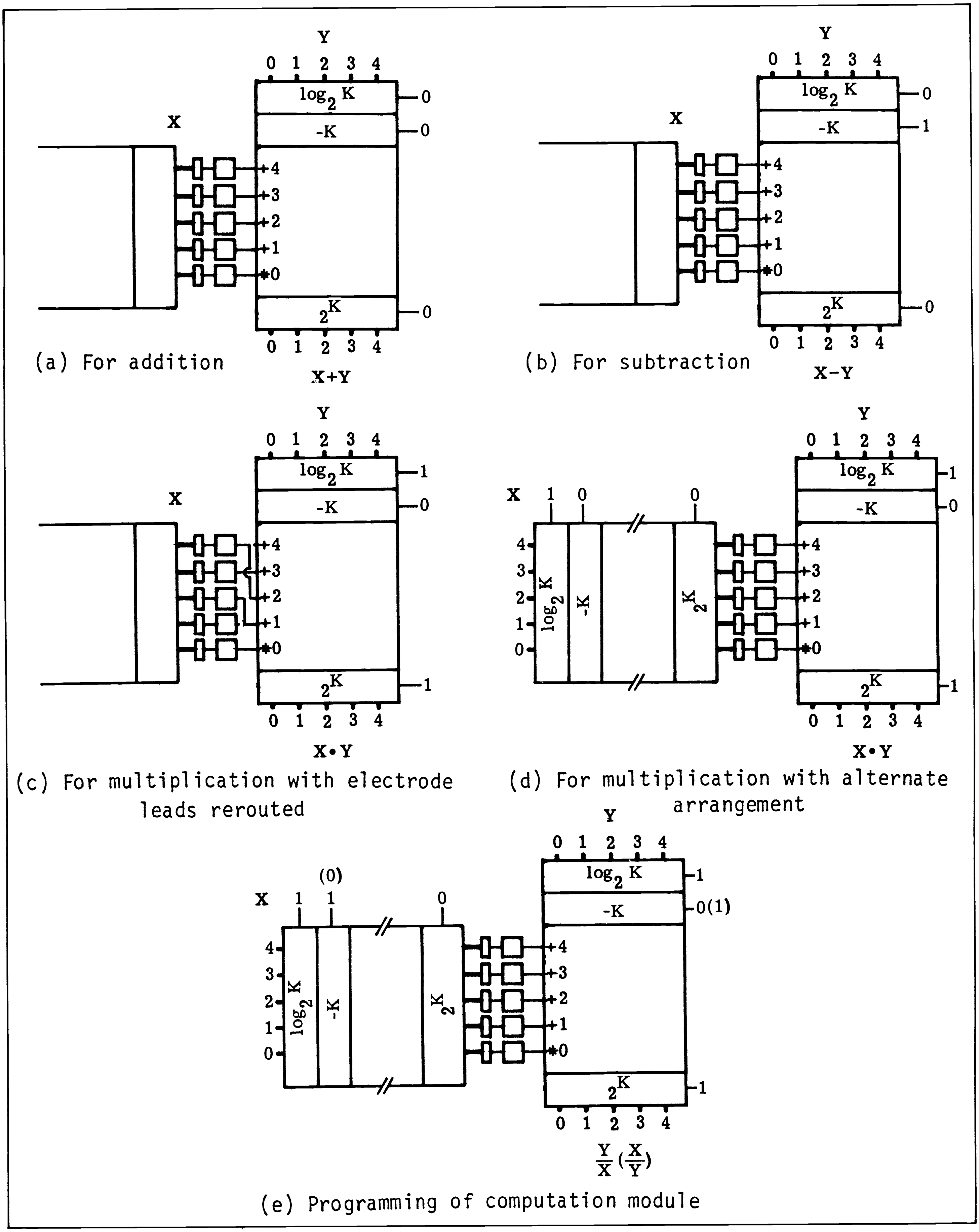

Fig. 16. Programming of computation module.

$A_{M \times N}$

pass through the multipliers row by row setting the corresponding row of adders. Light pulses are entered into the first adder of each row, providing in parallel the values of the first row of C_{1j} at the output. The flip flops are then reset, ready for the entries of the next row of

$A_{M \times N}$.

The total computation time is equal to M + 1 set-reset times of the module, and the number of computation modules required is 2NP. For example, to multiply two 10×10 matrices, the computation time would be about 30 + 0.05N nsec if we assume a module set-reset time of 3 nsec and the use of 200 computation modules for each modulus.

6. CONCLUDING REMARKS

We have described design concepts of a preliminary nature for a numerical optical processor based on the residue number system together with a review of the basic aspects of residue arithmetic. A variety of hardware implementations and computer architectures are possible in this field. We have provided ideas in this paper which emphasize use of light beam position mapping in the form of a versatile arithmetic module which can be applied to system designs and more complex mathematical operations. The concepts described provide a potential for improved processor design particularly in

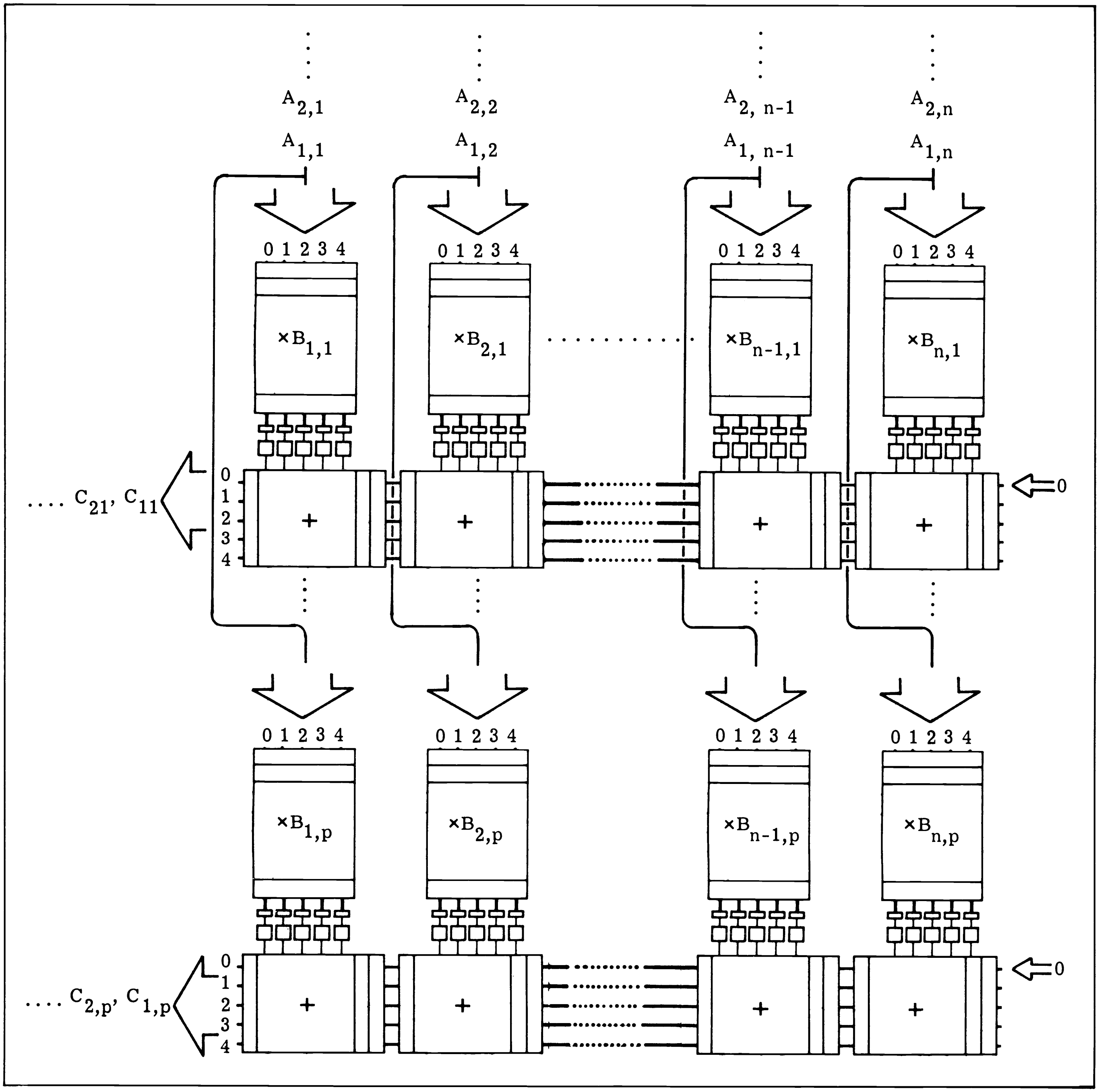

Fig. 18. Matrix multiplication.

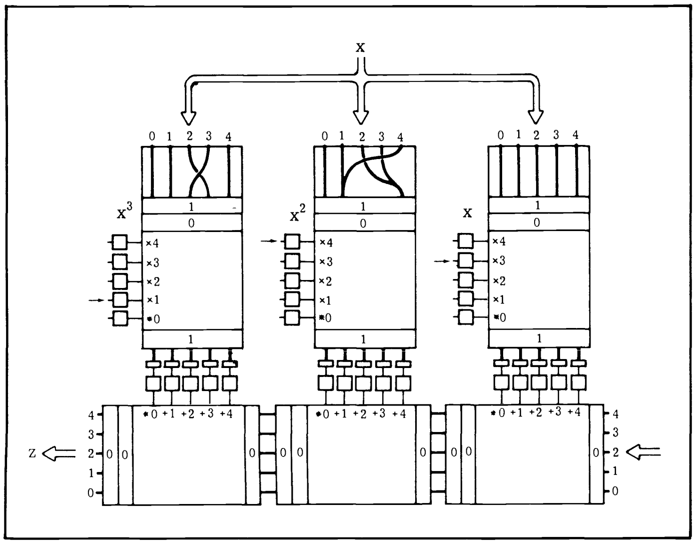

Fig. 17. Programmable arrangement for the evaluation of polynomials.

computing speed and possibly size and power consumption. For practical realization of the mapping approach, it would appear that further development is needed for hardware components of the integrated optics type directed specifically to this application. Of particular importance would be small optical switching devices having fast response, optical wave guides with low loss fabricated for a variety of path geometries, optical-to-electronic-to-optical conversion elements of small size, and, the compact integration of all of these items into a small modular package.

Computing algorithms and architecture warrant continued development efforts for such objectives as the optimization of design for faster operating speed and the minimization of component count.

Other design approaches, particularly those employing phase or polarization control of a light beam, have a potential for exceptional performance provided that certain aspects of the design concept are appropriately developed. This includes means for quantization over a large dynamic range, implementation for simplified approaches to multiplication such as the homomorphic method, means for rapid accurate output detection, and methods for efficient coupling between individual cyclic units.

7. REFERENCES

1. Cindrich, I., A. Klooster, and J. Marks, Proc. SPIE 128, 128 (1977).
2. Fienup, J. R., W. S. Colburn, B. J. Chang, and C. D. Leonard, Proc. SPIE, 118, 21 (1977).
3. Goodman, J. W., *Introduction to Fourier Optics*, McGraw Hill Book Company, 1968.
4. Huang, A., Y. Tsunoda, J. W. Goodman, and S. Ishihara, Appl. Opt. 18, 149 (1979).
5. Collins, S. A., Jr., Proc. SPIE 128, 313 (1977).
6. Tai, A., I. Cindrich, J. Fienup, and C. Aleksoff, Appl. Opt. 18, 2812 (1979).
7. Psaltis, D. and D. Casasent, Appl. Opt. 18, 163 (1979).
8. Szabo, N. S. and R. I. Tanaka, *Residue Arithmetic and Its Application to Computer Technology*, McGraw Hill, New York, 1967.
9. Cheney, P. W., IRE Trans. Elec. Comp., EC-11, 63 (1961).
10. Guffin, R. M., IRE Trans. Elec. Comp., EC-12, 164 (1962).
11. Banerji, D. K., IEEE Trans. Comp., 23, 1315 (1974).
12. Keir, Y. A., P. W. Cheney, and M. Tannenbaum, IRE Trans. Elec. Comp. EC-11, 501 (1962).
13. Garner, H. L., IRE Trans. Elec. Comp. EC-8, 140 (1959).
14. Baisi, F. and P. Maestrini, IEEE Trans. Comp. C-27, 1185 (1978).
15. Okada, M. and K. Takizawa, IEEE J. Quantum Elec. QE-15, 82 (1979).
16. Tsai, C. S., Le T. Nguyen, and C. C. Lee, Digest of Technical Papers Presented at the Topical Meeting on Integrated and Guided Wave Optics, Salt Lake City, TUC3-1 and TUC2-1, 1977.
17. Chang, B. J., ERIM Report 671034-2-X, January 1974.
18. Taylor, H., Appl. Opt. 17, 1493 (1978).
19. Cross, P. S., R. V. Schmidt, and R. L. Thorton, Digest of Technical Papers Presented at the Topical Meeting on Integrated and Guided Wave Optics, Salt Lake City, TUB1-1, 1977.
20. Kaminow, I. P., IEEE Trans. Micro. Theory and Tech. MIT-23, 57 (1975).