# Optical binary coded ternary arithmetic and logic

George Eichmann, Y. Li, and R. R. Alfano

Two ternary, an ordinary ternary (OT) and a binary balance ternary (BT), number representations to be used for optical computing are discussed. An unsigned OT number is represented by a string of symbols (0, 1, 2), while for the BT, the three logic symbols take on the set {−1, 0, +1}. The BT symbols can represent a signed number. Using a particular binary encoding method, the three ternary symbols are converted to a pair of binary symbols. The binary coded ternary (BCT) representation has two advantages. First, it allows use of the well-developed binary optical components. Second, compared with other optical multiple-valued number encoding schemes, it reduces the number of input–output channels and thus is able to conserve the optical space–bandwidth product. As an example for arithmetic operations, BCT full addition algorithms are given. As examples for multiple-valued logic computing, BCT Post, Webb, and residue logic elements are discussed. Using the two-port Sagnac interferometric switches, optical implementations of various BCT arithmetic and logic operations are described.

## I. Introduction

Recently, there has been a revival of interest in nonbinary, the so-called multiple-valued logic computing. The interest is due to the fact that the increase of logic density can provide additional processed information through each connection.[1,2] It has been indicated that an efficient way to implement multiple-valued logic is to use a mosaic of ultrafast large aperture optical elements.[3] Since for some optical materials the switching time is of the order of picoseconds and most optical beams can intersect without interaction, very high-bandwidth parallel optical signal channels can be established. For these reasons, a combination of multiple-valued computing and ultrafast optical switching may lead to the realization of a future generation of the ultrafast digital optical computer.

For binary optical computing, several number representations have been introduced. These methods include the digital multiplication by analog convolution (DMAC) algorithm[4] and the modified signed digit (MSD)[5] number representations. The DMAC method uses a mixed binary format to represent binary numbers and is suitable for optical multiplications.[6,7] The MSD is a redundant number system of radix two.[8,9] Using the MSD numbers, binary optical arithmetic operations have been performed. For nonbinary arithmetic optical computing, a residue number representation[10,11] has been suggested. Using residue number representation, based on a set of different modulo suboperations, carry-free arithmetic operations are performed. In this paper, a number of ternary representation methods are introduced. Ternary number system is optimum in terms of storage complexity.[2] Two ternary number representations, to be used for optical computing, the ordinary ternary (OT) and the binary balanced ternary (BT) representations, respectively, are discussed.[12] Since in practice an efficient and ultrafast tristable switch is difficult to realize, a practical method to synthesize the three logic states is to use binary encoding techniques. Here each ternary symbol is encoded into a pair of binary symbols. The arithmetic or logic operations thus performed are based on binary elements. In this paper, using binary coded OT (BCOT), and binary coded BT (BCBT) representations, various arithmetic and multiple-valued logic operations are discussed. A number of BCT arithmetic and logic implementation examples, based on the two-port Sagnac interferometric switches, are presented.

The paper is organized as follows, In Sec. II a short discussion of the OT and BT number representations and the corresponding BCT codes is presented. In Sec. III, BCT arithmetic operations are discussed. Addition algorithms for both BCT representations are given. In Sec.IV, using BCT, various Post, Webb, and residue multiple-valued elementary logic operations are discussed. In Sec. V, optical two-port Sagnac interferometric switch (TPSIS) binary logic gates and their interconnections are briefly discussed. In Sec. VI, based on the use of various TPSIS canonical logic

The authors are with City University of New York, City College, New York, New York 10031.

Received 14 March 1986.
0003-6935/86/183113-09$02.00/0.

gates, a number of optical implementation examples of BCT arithmetic and multiple-valued logic operations are presented.

## II. Ternary Numbers and Their Representations

In a radix $N$ ($N > 2$) number system, $N$ usually takes on the set of positive integer $(0, 1, \ldots, N - 1)$ values. When $N$ is odd, a symmetrical set of numbers $[-(N - 1)/2, \ldots, -1, 0, 1, \ldots, (N - 1)/2]$ can also be used.[2] A radix three is called the ternary number system. In the ternary system, the positive integer set (0, 1, 2) is called the OT, while the symmetrical number set (−1, 0, +1) is called the BT set. Using either set, an unsigned ternary number $A$ can be represented by an ordered string of symbols:

$$A = (a_n a_{n-1} \cdots a_0 \ldots a_{1-m} a_{-m}) = \sum_{i=-m}^{n} a_i \times 3^i, \qquad (1)$$

where $a_i$ is either a OT or BT symbol. Since BT is a signed digit form,[8] it is suitable to represent a signed number. In this paper, for both BCT representations, only integer numbers are considered.

To optically encode a nonbinary number system, an optical multistable switch can be used. However, in practice, it is more difficult to implement a fast and efficient optical multistable switch than its binary counterpart. For this reason, there is considerable interest in implementing nonbinary numbers using binary elements. In one approach, in the so-called pulse-position coding technique,[10] the $N$ signal channels represent $N$ logic or number levels. The presence or absence of a binary signal in a specific channel implies a particular logic or number value. Using a 2-D binary switch array, pulse-position coding has been used to perform residue arithmetic.[13] For either OT or BT number representations, this three-channel coding method can be used. In this paper, a new binary coded ternary (BCT) pulse-position coding approach is introduced. In this approach, the three logic or number values are encoded into a pair of binary digits. In turn, a ternary number is decomposed into two binary strings. The advantage of this approach is, that by combining BCT with pulse-position coding, each ternary symbol requires only two signal channels. Also, with this approach, various binary optical arithmetic and logic devices can directly be used to synthesize ternary logic and arithmetic operations.

The binary coded OT (BCOT) set is chosen as

$$\begin{aligned} 0_{OT} &= [0,0]_{BCOT}, \\ 1_{OT} &= [1,0]_{BCOT} = [0,1]_{BCOT}, \qquad (2) \\ 2_{OT} &= [1,1]_{BCOT}. \end{aligned}$$

Here the ternary OT digit is the sum of two binary digits present in the two channels. A clever way to choose the binary coded BT (BCBT), since it is a signed number representation, is to assign the positive (negative) number parts to different signal channels, i.e.,

$$\begin{aligned} -1_{BT} &= [0,1]_{BCBT} = \bar{1}_{BT}, \\ 0_{BT} &= [0,0]_{BCBT} = [1,1]_{BCBT}, \qquad (3) \\ +1_{BT} &= [1,0]_{BCBT}. \end{aligned}$$

Using the BCBT form, a negative number can be obtained from its positive expression by interchanging the channel positions of the two binary symbols. To represent a negative OT number, similar to the binary twos complement representation, a threes complement method can be used.[12]

As an example, the decimal number $47_{10}$ is represented in OT, BCOT, BT, and BCBT forms as

$$47_{10} = (1202)_{OT} = [1101, 0101]_{BCOT}, \qquad (4a)$$

$$47_{10} = (1\bar{1}\bar{1}1\bar{1})_{BT} = [10010, 01101]_{BCBT}. \qquad (4b)$$

## III. Arithmetic Computing Using BCT Representations

Before presenting a detailed description of the BCT arithmetic operations, three auxiliary BCT logic functions are discussed. The logic functions are the mutually exclusive equivalent pair (MEEP), the minor pair (MP), and the negation pair (NP) operations. To insure a unique BCBT output after some arithmetic operations [see Eq. (3)], the MEEP function is employed to convert the resultant [1,1] to its equivalent [0,0] form while leaving other pairs [1,0] and [0,1] alone. On the other hand, to insure a unique arithmetic result in the BCOT representation, the MP function is used to convert a [0,1] to a [1,0] pair while leaving other pairs [0,0] and [1,1] alone. Both the MEEP and the MP functions satisfy the addition relation

$$f_1 \text{“+”} f_2 = x_1 \text{“+”} x_2, \qquad (5)$$

where "+" denotes the arithmetic sum, and $x_i(f_i)$ are the MEEP or MP inputs (outputs), respectively. In the final auxiliary BCT logic function, the NP takes a positive number string pair and converts it to its corresponding negative number string pair. The NP function realization is different for the two different, the BCBT and BCOT, representations. The results for the Boolean logic design for the BCT MEEP and MP as well as for the BCBT NP are summarized in Table I.

Using these three auxiliary BCT logic operators and a binary full adder, BCT arithmetic operation can be

**Table I. Various Ternary Logic Functions and their BCOT Representations, $x$ and $f$, the Function Input and Output**

| | | | |
|---|---|---|---|
| MEEP | $f_1 = X_1 \cdot \overline{X_2}$ | $f_2 = \overline{X_1} \cdot X_2$ | BCT |
| MP | $f_1 = X_1 + X_2$ | $f_2 = X_1 \cdot X_2$ | BCT |
| NP | $f_1 = X_2$ | $f_2 = X_1$ | BCBT |
| $[A_1, A_2]$ | $f_1 = \overline{(\overline{(X_{21} \text{“+”} X_{12})} \cdot X_{22})} \cdot X_{11}$<br>$f_2 = (X_{12} \text{“+”} X_{21}) \cdot \overline{X_{21}}$ | | BCOT |
| | $f_1 = (X_{11} \text{“+”} X_{21}) \cdot \overline{X_{21}}$<br>$f_2 = \overline{(X_{11} \text{“+”} X_{21})} \cdot X_{11} + X_{12}$ | | BCBT |

performed. Using BCT addition, other arithmetic operations, such as subtraction and multiplication, can also be implemented. For example, using the addition algorithm, with an additional NP operation, the subtraction of two BCT numbers can be performed. Using an NP operator, first, the subtrahend is negated. A BCT adder is then used to add the negated subtrahend to the minuend. Using a number of BCT addition and shift operations, BCT multiplication of two numbers can also be performed. For the two BCT inputs (number strings) represented as

$$(x_1)_T = [x_{11},x_{12}]_{BCT} \qquad (x_2)_T = [x_{21},x_{22}]_{BCT}, \tag{6}$$

the BCT arithmetic or logic function $f(\ )$ can be obtained as

$$f = [f_1,f_2]_{BCT}, \tag{7}$$

where

$$f_1 = g(x_{11},x_{12},x_{21},x_{22}), \qquad f_2 = h(x_{11},x_{12},x_{21},x_{22}),$$

and $g(\ )$ and $h(\ )$ are either binary arithmetic or logic functions. Since the simultaneous use of four input strings can lead to a complicated operation, the use of two BCT partial additions is preferred.[12] For a BCT partial addition, the fact that

$$(x_1)_T \text{"+"} (x_2)_T = [x_{11},x_{12}]_{BCT} \text{"+"} [x_{21},0]_{BCT} \text{"+"} [0,x_{22}]_{BCT} \tag{8}$$

is used. Thus, for the full addition of two BCT number strings, two BCT partial addition operations are performed. Since each BCT partial addition involves only three number strings, it is simpler to process.

To add two BCOT numbers, first, using the MP operation, the given string pair $x_{11},x_{21}$ is replaced by the string pairs $y_{11},z_{21}$. Then another MP replacement is performed to convert $x_{12},z_{21}$ to $y_{12},y_{21}$. This operation[12] replaces all unwanted [0,1] with the desired [1,0] pairs. Next, the first BCOT partial addition is performed:

$$A = [A_1, A_2]_{BCOT} = [y_{11},y_{12}]_{BCOT} \text{"+"} [y_{21},0]_{BCOT}, \tag{9}$$

where $A_1$ and $A_2$ are mixed (arithmetic and logic) binary functions. To save space, $A_1$ and $A_2$ definitions[12] are given in Table I. To unify the partial result, another MP operation is used.

Using the partial result $[A_1,A_2]$ and the remaining $[0,x_{22}]$ string, by repeating the previous partial addition steps, the full BCOT addition is obtained:

$$\begin{aligned} S &= x_1 \text{"+"} x_2 \\ &= [A_1,A_2]_{BCOT} \text{"+"} [0,x_{22}]_{BCOT} \\ &= [A_1,A_2]_{BCOT} \text{"+"} [x_{22},0]_{BCOT}. \end{aligned} \tag{10}$$

Notice that in the last line in Eq. (10), an interchange between the strings in the two signal channels is indicated. This operation is guaranteed by the OT representation.

As a numerical example, consider the BCOT addition of the two decimal numbers $x_1 = 59_{10}$ and $x_2 = 19_{10}$. In this case, the BCOT numbers are $x_1 =$ [1011,1001] and $x_2 =$ [0101,0100]. The first BCOT partial addition is

$$[A_1,A_2]_{BCOT} = [1011,1001]_{BCOT} \text{"+"} [0101,0000]_{BCOT}. \tag{11a}$$

This expression, after the two consecutive MP replacements, becomes

$$[A_1,A_2]_{BCOT} = [1111,1001]_{BCOT} \text{"+"} [0001,0000]_{BCOT}. \tag{11b}$$

After some binary arithmetic and logic operations, the values $A_1$ and $A_2$ are valued as

$$[A_1,A_2]_{BCOT} = [1110,1010]_{BCOT}. \tag{11c}$$

Since this is a MP, the result $[A_1,A_2]$ remains the same after the next MP operation. In the second part of the full addition, the partial addition

$$S = [S_1,S_2]_{BCOT} = [1110,1010]_{BCOT} \text{"+"} [0100,0000]_{BCOT} \tag{11d}$$

is performed. The MP replacements give

$$S = [S_1,S_2]_{BCOT} = [1110,1110]_{BCOT} \text{"+"} [0000,0000]_{BCOT}. \tag{11e}$$

The use of the Table I BCOT $A_1,A_2$ and MP operations reduces this to

$$S = [1110,1110]_{BCOT} = (2220)_{OT} = 78_{10}. \tag{11f}$$

The BCBT full addition algorithm, for two BCBT number strings $[x_{11},x_{12}]$ and $[x_{21},x_{22}]$, is given as follows. First, the number strings $x_{11},x_{21}$ are replaced using a MP logic operator with the strings $y_{11},y_{21}$. Next, the first BCBT partial addition is performed as

$$A = [A_1,A_2]_{BCBT} = [y_{11},x_{12}]_{BCBT} \text{"+"} [y_{21},0]_{BCBT}, \tag{12}$$

where the $A_1,A_2$ expressions can be found in Table I. To remove the ambiguity due to possible [1,1] pairs, a MEEP function is used. Unlike the BCOT full addition, the BCBT full addition cannot be obtained by two straightforward BCBT partial additions. This is so because

$$[0,x_{22}]_{BCBT} = NP([x_{22},0]_{BCBT}) \neq [x_{22},0]_{BCBT}. \tag{13}$$

Thus, to obtain the final sum, an additional NP operation is required. The BCBT full addition of the two BCBT numbers

$$\begin{aligned} S = [S_1,S_2]_{BCBT} &= [A_1,A_2]_{BCBT} \text{"+"} [0,x_{22}]_{BCBT} \\ &= NP([A_2,A_1]_{BCBT} \text{"+"} [x_{22},0]_{BCBT}). \end{aligned} \tag{14}$$

Here, as a BCBT full addition example, the previous numerical example of the sum of the decimal numbers $59_{10}$ "+" $19_{10}$ is given. The corresponding number strings now are $x_1 =$ [10100,01011] and $x_2 =$ [01001,00100]. In the first step, the MP replacement leads to

$$\begin{aligned} [A_1,A_2]_{BCBT} &= [11101,01011]_{BCBT} \\ &\quad \text{"+"} [00000,00000]_{BCBT}. \end{aligned} \tag{15a}$$

After the BCBT $A_1,A_2$ binary arithmetic and logic operations, the result becomes

$$[A_1,A_2]_{BCBT} = [11101,01011]_{BCBT}. \tag{15b}$$

To unify the result, the use of the MEEP operator generates

$$[A_1,A_2]_{BCBT} = [10100,00010]_{BCBT}. \qquad (15c)$$

In the second part of the operation, the partial addition

$$[B_1B_2]_{BCBT} = [00010,10100]_{BCBT}$$
$$\text{“+”}\ [00100,00000]_{BCBT} \qquad (15d)$$

is performed. Repeating the described steps yields

$$[B_1,B_2]_{BCBT} = [00110,10100]_{BCBT}. \qquad (15e)$$

This expression reduces after a MEEP operation to

$$[B_1,B_2]_{BCBT} = [00010,10000]_{BCBT}. \qquad (15f)$$

Finally, an NP function is used to obtain the final full addition result:

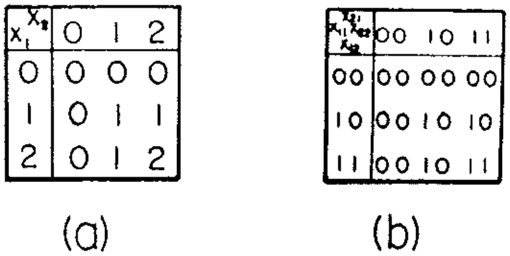
$$[S_1,S_2]_{BCBT} = NP([B_1,B_2]_{BCBT})$$
$$= [10000,00010]_{BCBT} = (100\bar{1}0)_{BT} = 78_{10}. \qquad (15g)$$

## IV. Binary Coded Ternary Logic Computing

In this section, BCT logic computing is discussed. For both BCT number representations, the logic design follows the binary logic design method. First, a ternary truth table that defines a specific ternary logic function is established. Next, this truth table is transformed to its BCT form. Using Boolean algebra expressions and minimizing them using Karnaugh maps, the simplified two-output-channel binary logic functions are obtained. For both BCOT and BCBT logic elements, an identical procedure is used. To explain the technique, without loss of generality, the BCOT representation is used. Various canonical BCOT Post, Webb, as well as ternary residue logic functions are discussed.

To implement all possible multiple-valued logic functions, a set of multiple-valued logic primitives is required. In the Post logic system, this set may include a minimum (MIN), maximum (MAX), succession (SUC), negation (NEG), as well as a number of literal (LIT) multiple-valued logic operators.[2] The SUC, NEG, and LITs are unary operators that act on a single input and generate a single output, while the MIN and MAX act on multiple inputs to yield a single output. To obtain logic completeness, the unary operators are used to augment the multiple-input MIN and MAX operators.

To generate a BCOT Post logic representation, as an example, implementation of a Post MIN ternary multiple-valued logic function is given. In Figs. 1(a) and (b), based on the OT MIN definition (see Table II) and the use of the BCOT codes, the OT and BCOT truth tables are tabulated. To enforce a unique logic output, we let $1_{OT}$ = [1,0]. The BCOT truth table includes two ($f_1$ and $f_2$) binary output channels. After this step, using Boolean algebra expressions and minimizing them with the aid of a Karnaugh map, the two binary output logic functions are



| $x_1$ \ $x_2$ | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 2 | 0 | 1 | 2 |

(a)

| $x_{11},x_{12}$ \ $x_{21},x_{22}$ | 00 | 10 | 11 |
|---|---|---|---|
| 00 | 00 | 00 | 00 |
| 10 | 00 | 10 | 10 |
| 11 | 00 | 10 | 11 |

(b)

Fig. 1. Truth tables for ternary Post logic MIN function. (a) and (b) correspond to the OT and the BCOT formats, respectively.

$$f_1 = x_{11} \cdot x_{21}, \qquad (16a)$$

$$f_2 = x_{12} \cdot x_{22}, \qquad (16b)$$

where the dot stands for binary AND operation. Using this method, other BCOT Post logic functions are obtained. The results are summarized in Table II.

Similar to the binary case, in a multiple-valued logic system there also exist a number of universal logic elements. A universal logic element is that element with which all possible multiple-valued logic functions can be generated.[14] An example of a multiple-valued universal logic element, the analog of binary NAND gate, is the Webb operator.[2] Using the previously described method, the BCOT Webb functions can also be obtained (see Table II).

The BCOT logic design method can also be applied to obtain the ternary-valued residue logic elements.[13] For example, from the ternary residue addition and multiplication truth tables (see Table II), a pair of the BCOT truth tables can be obtained. Again, using the previously described method, the logic reduced BCOT output functions are obtained. In Table II, the BCOT residue addition and multiplication logic functions are also included.

## V. Two-Port Sagnac Interferometric Switches

For the BCT optical computing, various binary optical logic gates may be used. It has been shown that a nonlinear Sagnac interferometer (SI) may be used as an ultrafast optical switch.[15,16] Based on the SI, a new device called the two-port Sagnac interferometric switch (TPSIS) can be formed. Two distinct properties of this switch are its retroreflection-free autostabilization property and its two-port switching capability. Using TPSISs, various canonical optical logic gates can be constructed. Here the implementations of an optical binary NOT, AND, OR, XOR gates are briefly summarized.

In Fig. 2, a schematic diagram of an optical TPSIS is shown. The two input and output beams are denoted $I_1$, $I_2$ and $O_1$, $O_2$, respectively. The beam $S$ is the optical switching signal. The optical components are a nonpolarizing 50/50 beam splitting ratio beam splitter (BS), a polarizing beam splitter (PBS) that trans-

Table II. Auxiliary Functions Needed for BCT Arithmetic Operations. $[A_1,A_2]$ Functions are in a Mixed Format Where Binary Full Additions Followed by Several Logic Operations need to be Performed

| | | | | |
|---|---|---|---|---|
| Minimum | $MIN(x_1,x_2)$ | $=\begin{cases}X_1 & \text{if } X_1 \le X_2\\ X_2 & \text{if } X_1 > X_2\end{cases}$ | $f_1 = X_{11} \cdot X_{21}$ | $f_2 = X_{12} \cdot X_{22}$ |
| Maximum | $MAX(x_1,x_2)$ | $=\begin{cases}X_1 & \text{if } X_1 \ge X_2\\ X_2 & \text{if } X_1 < X_2\end{cases}$ | $f_1 = X_{11}+X_{21}$ | $f_2 = X_{12} + X_{22}$ |
| Succession | $SUC(x_1)$ | $= (x_1 + 1) \bmod 3$ | $f_1 = \overline{X_{12}}$ | $f_2 = X_{11} \cdot \overline{X_{12}}$ |
| Negation | $NEG(x_1)$ | $= (2 - x_1) \bmod 3$ | $f_{11} = \overline{X_{12}}$ | $f_{12} = \overline{X_{11}}$ |
| Literals | ${}^{a}x^{b}$ | $= \begin{matrix}2 & \text{if } a \le x \le b\\ 0 & \text{otherwise}\end{matrix}$ | ${}^{0}X_1^{0} = {}^{0}X_2^{0} = \overline{X_1}$ | ${}^{0}X_1^{1} = {}^{0}X_2^{1} = \overline{X_2}$ |
| | | | ${}^{0}X_1^{2} = {}^{0}X_2^{2} = 1$ | ${}^{1}X_1^{1} = {}^{1}X_2^{1} = X_1 \cdot \overline{X_2}$ |
| | | $(a,b) \in 0,1,2$ | ${}^{1}X_1^{2} = {}^{1}X_2^{2} = X_1$ | ${}^{2}X_1^{2} = {}^{2}X_2^{2} = X_2$ |
| Summation (mod 3) | $SUM(x_1,x_2)$ | $\begin{array}{c\|ccc} X_1 \backslash X_2 & 0 & 1 & 2\\ \hline 0 & 0 & 1 & 2\\ 1 & 1 & 2 & 0\\ 2 & 2 & 0 & 1\end{array}$ | $f_1 = X_{11} \oplus X_{21} + X_{12} \cdot X_{22} + X_{11} \cdot \overline{X_{12}} \cdot \overline{X_{22}}$ | $f_2 = X_{12} \cdot \overline{X_{21}} + \overline{X_{11}} \cdot X_{22} + X_{11} \cdot \overline{X_{12}} \cdot X_{21} \cdot \overline{X_{22}}$ |
| Product (mod 3) | $PRO(X_1,X_2)$ | $\begin{array}{c\|ccc} X_1 \backslash X_2 & 0 & 1 & 2\\ \hline 0 & 0 & 0 & 0\\ 1 & 0 & 1 & 2\\ 2 & 0 & 2 & 1\end{array}$ | $f_1 = X_{11} \cdot X_{21}$ | $f_2 = X_{11} \cdot \overline{X_{12}} \cdot X_{22} + X_{21} \cdot \overline{X_{22}} \cdot X_{12}$ |
| Webb | $WEB(X_1,X_2)$ | $\begin{array}{c\|ccc} X_1 \backslash X_2 & 0 & 1 & 2\\ \hline 0 & 1 & 0 & 0\\ 1 & 0 & 2 & 0\\ 2 & 0 & 0 & 0\end{array}$ | $f_1 = f_2 + \overline{X_{11}} \cdot \overline{X_{21}}$ | $f_2 = X_{11} \cdot X_{21} \cdot \overline{X_{12}} \cdot \overline{X_{22}}$ |

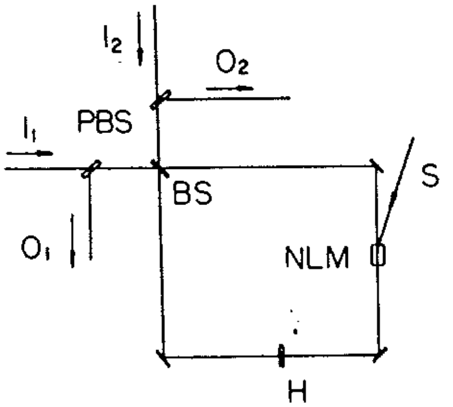


Fig. 2. Optical TPSIS schematic diagram, $I_1$ ($I_2$) and $O_1$ ($O_2$), the first (the second) input and output signal channels; $S$, inducing beam; BS, nonpolarizing 50/50 splitting ratio beam splitter; PBS, polarizing beam splitter; NLM, nonlinear material; $H$; polarizing halfwave plate.

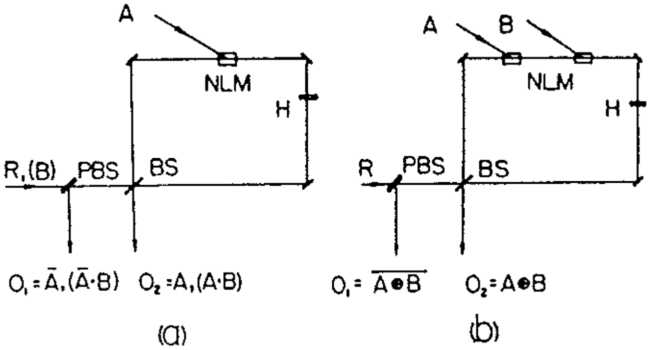


Fig. 3. Schematic diagrams of binary TPSIS gates. (a) NOT and AND functions and (b) XOR and its complement functions. —, ·, and ⊕ denote NOT, AND, and XOR operations.

mits a linear and reflects its orthogonally polarized beam counterpart, and a halfwave plate $H$ whose fast axis is oriented at an angle 45° with respect to the input polarization direction resulting in an orthogonal linear polarization between the input and output beams. To avoid optical phase conjugation, the nonlinear material (NLM) is placed asymmetrically in the interferometer.[16] Similar to an electrooptic waveguide interferometric switch, when the optical signal pulse $S$ is off, the signal is guided from $I_1$ ($I_2$) to $O_1$ ($O_2$) channels with an orthogonal polarization forced by the $H$ plate. When the optical signal $S$ is on and polarized parallel to either input $I_1$ or $I_2$ beam polarization, there exists due to the Kerr effect an intensity induced $\pi$ phase imbalance for the two counterpropagating beams arriving at the BS. This intensity-induced $\pi$ phase imbalance switches the signal from $I_1(I_2)$ to $O_2(O_1)$. Thus, using a TPSIS with a threshold detecting device, optical channel switching is performed.

The TPSIS can be viewed as a binary logic INVERTER or a NOT operator. In Fig. 3(a), the two identically polarized optical inputs $A$ and $R$, where $A$ is an input logic and $R$ is an interferometric reference beam, are shown. The $A$ beam low- and high-energy states are represented by two positive (bright true) logic values: zero and one, respectively. When $A$ is a zero, the input $R$ is retroreflected and separated by the TPSIS to channel $O_1$. When $A$ is a one, its output is switched to channel $O_2$. Therefore, for the binary input $A$, the channels $O_1$ and $O_2$ display logic variables $\bar{A}$ and $A$, where the bar represents the logic inversion. Similarly, when $R$ is a second binary logic variable $B$, channels $O_1$ and $O_2$ display the logic functions $\bar{A} \cdot B$ and $A \cdot B$. When a BS is used to combine two beams, this combination yields an optical OR element. When neither of the two input beams is on, the zero logic output (low-energy state) occurs.

Next, a TPSIS implementation of an XOR gate is considered. In Fig. 3(b), two cascaded NLMs are placed in the TPSIS loop. The NLMs are interrogated by the two logic inputs $A$ and $B$. For identical logic values, the corresponding results for the two input channels are logic one for $O_1$ and logic zero for $O_2$. For opposite logic values, the logic values zero (one) for channels $O_1(O_2)$ are obtained. Thus, at the output channels $O_2(O_1)$, the logic functions XOR and its complement are generated.

In principle, the TPSIS gates are cascadable. Here two TPSIS interconnection requirements are discussed. First, because of the orthogonal polarization states between the TPSIS input and output signals, the interconnection of two TPSISs requires polarization matching. Using a matching pair of TPSISs, polarization matching can be performed. Second, for TPSIS interconnections, a cascade of power or a fan-out is required. In a TPSIS, the output optical power is derived from the $R$ beam power. Since it is assumed there is no energy exchange between the two counterpropagating beams,[17] the output energy state does not depend on the inducing beam power. Thus, for a high $\chi^3$ nonlinear material, the inducing beam power can be less than the $R$ beam power. Similar to an optical transistor,[18] the TPSIS output power is larger than the inducing beam power and is able to drive several subsequent gates. In this paper, to fulfill the power condition, it is assumed that an efficient ultrafast NLM is available.

## VI. BCT Optical Computing and Multiple-Valued Logic Using TPSIS

In this section, using the BCT representations and with TPSIS as the binary optical elements, optical implementations of various BCT arithmetic and logic operations are discussed. Other approaches, however, with different optical switch elements, such as the bistable devices[19] and the liquid-crystal light valve spatial light modulators,[20] are also possible.

First, optical implementations of ternary arithmetic operations are discussed. For the BCT addition, three BCT auxiliary logic elements (MEEP, MP, and NP) as well as a binary full adder are required. To generate the MEEP function, two parallel binary AND functions are used. These two AND functions [see Fig. 3(a)] can easily be obtained using two TPSISs. In Fig. 4(a), the TPSIS MEEP implementation is depicted, where the left (right) TPSIS generates the BCT $f_1(f_2)$ output. Similarly, in Fig. 4(b), using a TPSISs as an AND element and a BS as an OR element, an optical BCT MP is implemented. The NP operator is different in the two, the BCOT and BCBT, representations. Because there exist a number of ways to represent negative numbers in BCOT, the discussion of an optical BCOT NP is omitted. The BCBT NP, on the other hand, is uniquely defined. For the BCBT NP generation, the interchange of the input signal channels will lead to the desired output. This can be accomplished with passive optical elements, such as prisms, mirrors, gratings, and waveguides.

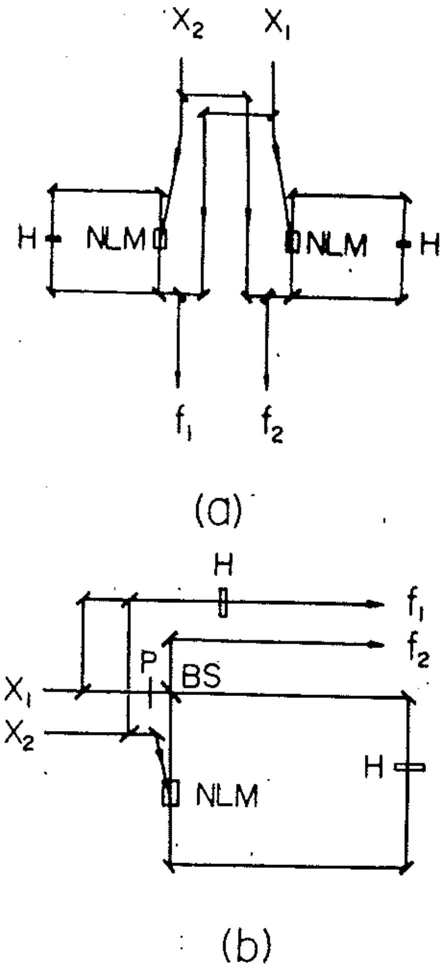


Fig. 4. (a) TPSIS MEEP and (b) TPSIS MP function generations, $P$, linear polarizer; $x_1$,$x_2$ and $f_1$,$f_2$ are the two input and output BCT channels.

To perform BCT arithmetic addition, two BCT partial adders are needed. The key element in a BCT partial adder is a binary full adder. The binary full adder logic functions are

$$S_i = A_i \oplus B_i \oplus C_i, \tag{17a}$$

$$C_{i+1} = (A_i \cdot B_i) \oplus (B_i \cdot C_i) \oplus (C_i \cdot A_i), \tag{17b}$$

where $\oplus$ denotes the binary logic XOR function, $A_i$ and $B_i$ are $i$th bits of two input binary sequences, $S_i$ and $C_{i+1}$ are the sum and carry generated from the $i$th bit addition outputs. In Fig. 5, based on the described TPSIS AND and XOR, a schematic diagram of a binary optical full adder is shown. Due to the parallel pro-

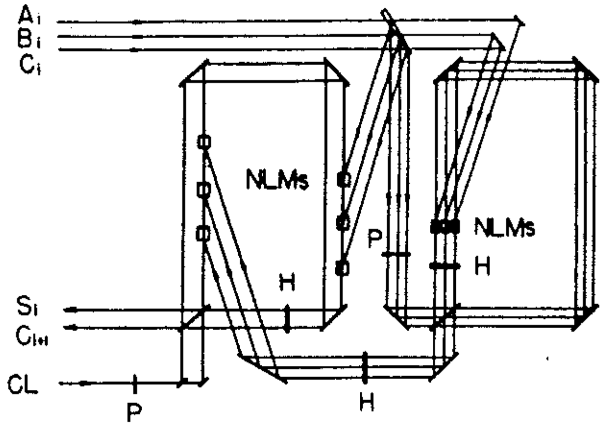


Fig. 5. TPSIS binary full adder. $A_i$ and $B_i$, two input signals ($i$th bit), $C_i$ ($C_{i+1}$), $i$th ($i$ + 1th) carry, $S_i$, $i$th output of summation; CL, clock.

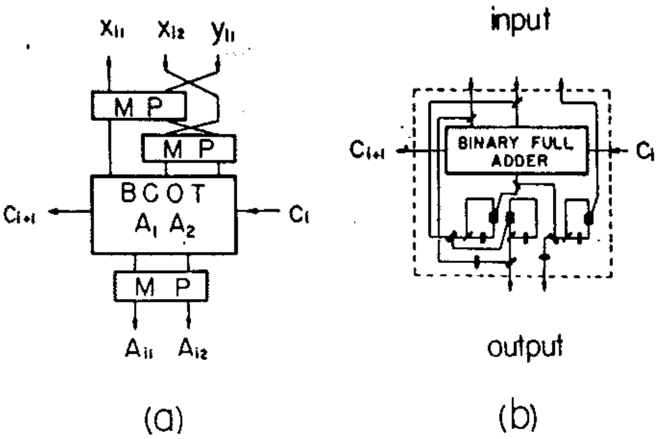


Fig. 6. (a) BCOT partial addition signal flow diagram for adding $i$th digits of $[x_1,x_2]_{\text{BCOT}}$ and $[y_1,0]_{\text{BCOT}}$, all mirror sketches are omitted; (b) BCOT $A_1A_2$ function generation, a binary full adder, three AND, and one OR gates are used.

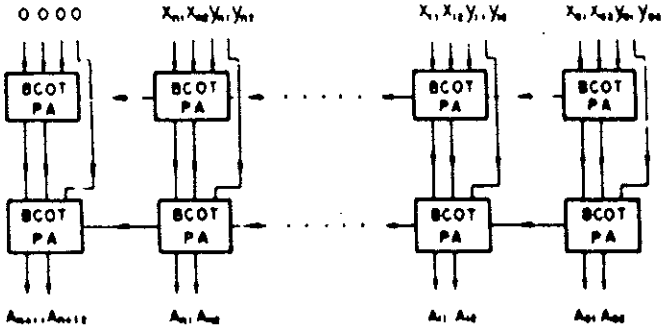


Fig. 7. $N$-digit BCOT adder for two $N$-digit input numbers $x$ = $[x_1,x_2]_{\text{BCOT}}$ and $y$ = $[y_1,y_2]_{\text{BCOT}}$.

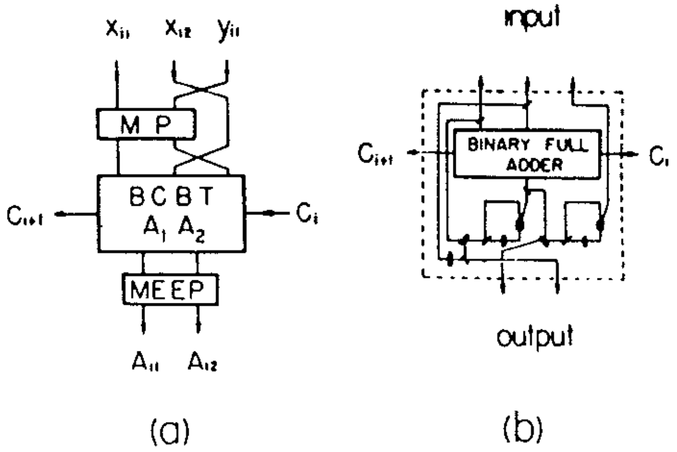


Fig. 8. (a) BCBT partial addition signal flow diagram for adding $i$th digit of $[x_1,x_2]_{\text{BCBT}}$ and $[y_1,0]_{\text{BCBT}}$; (b) BCBT $A_1,A_2$ function generation, a binary full adder, two AND, and one OR gates are used.

cessing property of the TPSIS, only two interferometers need to be employed, one for generation of the ANDs and another for generation of the XOR functions. At the left TPSIS, the sum output $S_i$ [see Eq. (17a)], using a three-input XOR function, is generated. The right-side TPSIS performs the three AND functions: $A_i \cdot B_i$, $B_i \cdot C_i$, and $C_i \cdot A_i$. To obtain the carry $C_i$+1 [see Eq. (17b)], the three optical signals are then guided to the left TPSIS. Finally, the carry can be used for the next bit addition, and the sum is outputed. Several $H$ plates are used for polarization matching. Because of two linear polarizers used for the optical retroreflection isolation, there is no feedback to the input.

To explain the BCT optical full adder implementation, first, consider a BCT partial adder design. Since the optical implementations of the three auxiliary BCT logic operators as well as a binary full adder have already been described, only the block diagram of these elements is indicated. In Fig. 6(a) the BCOT partial adder signal flow diagram is shown. It contains three parts with each corresponding to an algorithm step. The input signals arrive at the top black box that performs the two MP logic functions. The outputs are then injected to the middle box where the partial adder $A_1,A_2$ functions (see Table I) are performed. In Fig. 6(b), this part of the diagram is drawn separately, where in addition to a binary full adder, three TPSIS AND gates are used. To unify the results, the bottom box of Fig. 6(a) is used to perform the final partial addition operation. Based on two such BCOT partial adders, a BCOT full adder can be obtained. In Fig. 7, using an array of $N$ + 1 full adders, an $N$-digit TPSIS BCOT adder is shown. Similarly, in Fig. 8(a), a BCBT partial adder is depicted. Again, three different sections are shown. For clarity, in Fig. 8(b), the middle part that corresponds to the BCBT partial adders $A_1,A_2$ operations (see Table I) is drawn separately. For the full BCBT addition, two BCBT partial adders and an additional BCBT NP operator are used. In Fig. 9, the $N$-digit BCBT adder connection diagram is shown. Using these adders, BCT subtractions can also be performed. However, for the preparation of inputs for the two number subtraction, an additional NP logic element is needed.

Finally, optical implementations of BCT multiple-valued canonical logic functions are discussed. In Fig.

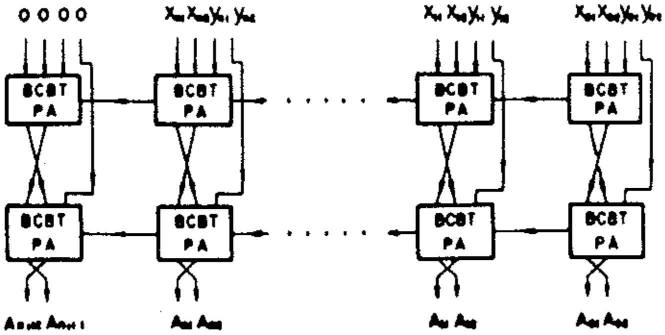


Fig. 9. *N*-digit BCBT adder for two *N*-digit input numbers $x = [x_1,x_2]_{\rm BCBT}$ and $y = [y_1,y_2]_{\rm BCBT}$.

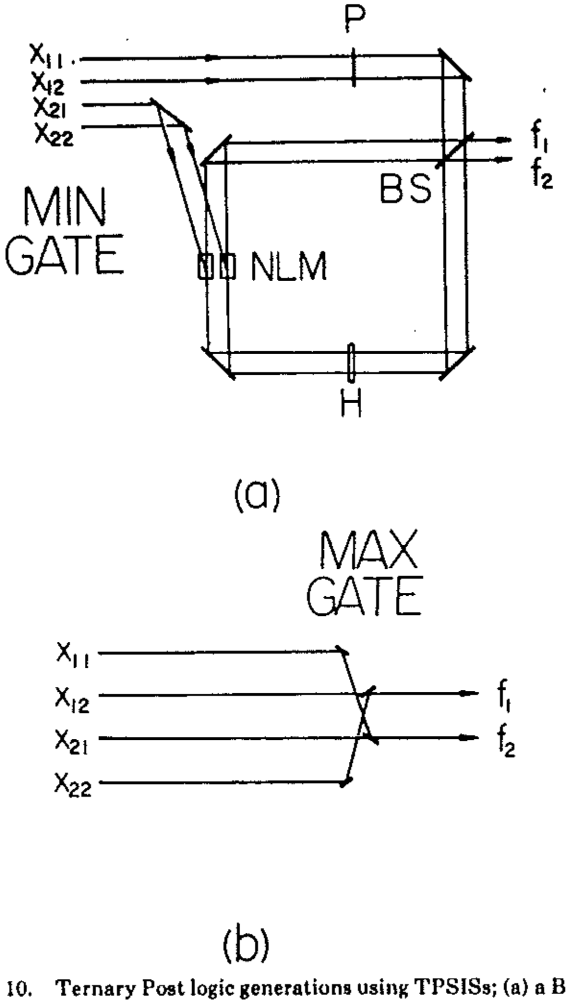


Fig. 10. Ternary Post logic generations using TPSISs; (a) a BCOT MIN gate and (b) a BCOT MAX gate.

10(a), the BCOT MIN function implementation is shown. Two parallel TPSISs simulate the binary AND gates. In Fig. 10(b), a schematic diagram of an optical BCOT MAX function (see Table II for the BCOT output functions) is depicted. Here two mirrors and two BSs are used for the two optical binary OR functions. Similarly, other BCOT logic functions (see Table II), such as the SUC, NEG, and LIT operators, the universal Webb as well as the ternary residue logic operators can also be optically generated.

## VII. Summary

In this paper, TPSIS-based ternary optical computing has been discussed. Two, OT and BT, ternary number representations are used. While the OT is suitable for unsigned, the BT is suitable for both signed and unsigned number representations. To represent a ternary number, binary encoding techniques were discussed where each ternary digit is represented by a binary signal channel pair. Compared with other multiple-valued optical encoding methods, one advantage of the optical BCT is that it conserves the system's space-bandwidth product. Another advantage is that the ternary optical computing elements can be directly synthesized using binary optical switches. As examples of BCT arithmetic computing, both BCT-type addition algorithms were given. Based on the algorithm other BCT arithmetic operations such as subtraction and multiplication can also be performed. Using BCOT numbers, various Post, Webb, and residue multiple-valued logic operations were also discussed. A number of optical TPSIS binary gates implementation examples were presented.

This work is supported in part by a grant from the Air Force Office of Scientific Research. The constructive comments of the referees are deeply appreciated.

## References

1. T. T. Dao, E. J. McCluskey, and L. K. Russell, "Multivalued Logic I²L," IEEE Trans. Comput. C26, 1233 (1977).
2. S. L. Hurst, "Multiple Valued Logic—its Status and its Future," IEEE Trans. Comput. C-33, 1160 (1984).
3. G. Abraham, "Multiple-Valued Logic for Optoelectronics," Opt. Eng. 25, 3 (1986).
4. D. Psaltis, D. Casasent, D. Neft, and M. Carlotto, "Accurate Numerical Computation by Optical Convolution," Proc. Soc. Photo-Opt. Instrum. Eng. 232, 151 (1980).
5. B. L. Drake, R. P. Bocker, M. E. Lasher, R. H. Patterson, and W. J. Miceli, "Photonic Computing Using the Modified Signed Digit Number Representation," Opt. Eng. 25, 38 (1986).
6. R. P. Bocker, S. R. Clayton, and K. Bromley," Electrooptical Matrix Multiplication Using the Twos Complement Arithmetic for Improved Accuracy," Appl. Opt. 22, 2019 (1983).
7. P. S. Guilfoyle, "Systolic Acousto-optic Binary Convolver," Opt. Eng. 23, 20 (1984).
8. A. Avizienis, "Signed-Digit Number Representations for Fast Parallel Arithmetic," IRE Trans. Electron Comput. EC-10, 389 (1961).
9. N. Takagi, H. Yasuura, and S. Yajima, "High-Speed VLSI Multiplication Algorithm with a Redundant Binary Addition Tree," IEEE Trans. Comput. C-34, 789 (1985).
10. A. Huang, Y. Tsunoda, J. W. Goodman, and S. Ishihara, "Optical Computation Using Residue Arithmetic," Appl. Opt. 18, 149 (1979).
11. D. Psaltis and D. Casasent, "Optical Residue Arithmetic: a Correlation Approach," Appl. Opt. 18, 163 (1979).
12. G. Frieder and C. Luk, "Algorithms for Binary Coded Balanced and Ordinary Ternary Operations," IEEE Trans. Comput. 24, 212 (1975).
13. A. Tai, I. Cindrich, J. R. Fienup, and C. C. Aleksoff," Optical

Residue Arithmetic Computing with Programmable Computation Modules," Appl. Opt. 18, 2812 (1979).
14. Z. Kohavi, *Switching and Finite Automata Theory* (McGraw-Hill, New York, 1978).
15. Y. Li, G. Eichmann, and R. R. Alfano, "Pulsed Mode Laser Sagnac Interferometry with Applications in Nonlinear Optics and Optical Switching," Appl. Opt. 25, 209 (1986).
16. G. Eichmann, Y. Li, and R. R. Alfano, "Digital Optical Logic using a Sagnac Interferometer Switch," Opt. Eng. 25, 91 (1986).
17. A. E. Kaplan and P. Meystre, "Directionally Asymmetrical Bistability in a Symmetrically Pumped Nonlinear Ring Interferometer," Opt. Commun. 40, 229 (1982).
18. P. W. Smith and W. J. Tomlinson, "Bistable Optical Devices Promise Subpicosecond Switching," IEEE Spectrum 18, 26, (1981).
19. J. L. Jewell, M. C. Rushford, and H. M. Gibbs, "Use of a Single Nonlinear Fabry-Perot Etalon as Optical Logic Gates," Appl. Phys. Lett 44, 172 (1984).
20. M. T. Fatehi, K. C. Wasmundt, and S. A. Collins, Jr., "Optical Logic Gates Using Liquid Crystal Light Valve: Implementation and Application Example," Appl. Opt. 20, 2250 (1981).

Since there is chlorine in the oxidizer, but not in the binder, its presence maps the oxidizer distribution. The SEM images are analyzed with the SMFA, first to obtain the projection of the chlorine concentration along the axis, and then to obtain the spatial-frequency distribution of the fluctuations in that distribution. Finally, by taking the reciprocals of the frequencies of peaks in the spatial frequency distribution, characteristic dimensions related to the sample are obtained.

From tabulations of the characteristic dimensions for a number of samples, it appears that the characteristic dimensions correspond approximately to the range of particle sizes in the samples. The characteristic dimensions for the coarse samples ranged from 45 to 398 μm; those for the fine samples, from 8 to 381 μm.

This work was done by Leon D. Strand, Norman S. Cohen, and Miguel A. Hernan of Caltech for NASA's Jet Propulsion Laboratory. Refer to NPO-16487.

## Ion-deposited polished coatings

A process has been developed to provide a highly polished and adherent dense coating to a substrate that is free of voids, contaminants, and inclusions. A broad-beam ion source is used to sputter-polish while a vapor or sputter deposition is occurring simultaneously on the substrate surface. Prior techniques removed material from the surface, exposing subsurface voids and contaminants. No simultaneous fill of voids occurred and therefore the surface was not free of defects since these were removal processes.

The new process consists of using a broad-beam ion source in an evacuated chamber to ion-clean a rotating surface that allows the grazing incidence of the ion beam. This sputter cleans off absorbed gases, organic contaminants, and oxides of the mirror surface. In addition to the cleaning, surface protrusions will be sputter-etched away.

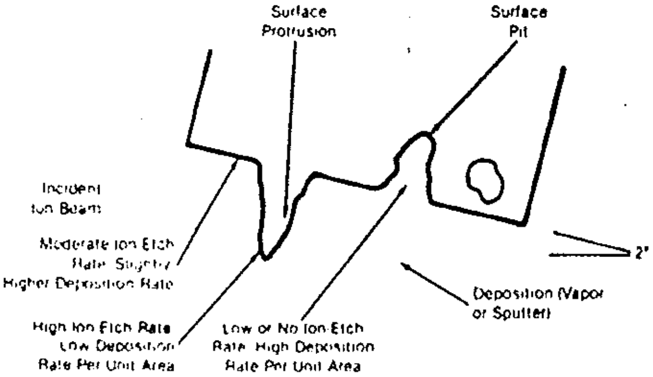


Fig. 14. Grazing-incidence ion polishing is conducted simultaneously with deposition by normally incident vapor or a sputtering beam.

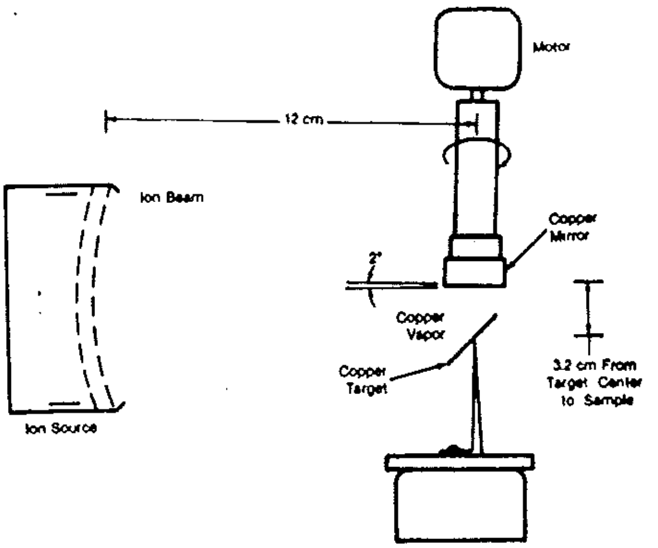


Fig. 15. Ion sputter polishing and deposition give a smooth deposited and polished surface, in a manner analogous to that of the repeated painting and sanding of wood.

Once the surface has been ion-cleaned, a variety of vacuum-deposition techniques can be used to deposit adherent coatings on the cleaned mirror surface. Figure 14 depicts grazing-incidence ion polishing with simultaneous, normally incident vapor or sputter deposition. The simultaneous sputter polishing and vacuum deposition must be done in a manner so that the sputter-etch rate is slightly lower than the deposition rate on the mirror surface. This allows pits or depressions in the mirror surface to fill in quickly because the grazing-incident ion beam does not have a view factor to sputter-etch these areas which are receiving near-normal incident vapor or sputter deposition. At the same time, protrusions, bumps, and other convex surface asperities have a sputter-etch rate higher than surrounding smooth flat surfaces. This is due to the relative angles at which the sputter-etching ions and the deposition material arrive.

Figure 15 depicts the processes of simultaneous ion sputter, polishing, and deposition. The process leads to a smooth deposited and polished surface in a similar manner analogous to the macroscopic process of multiple painting and sanding operations for wood finishing. The net deposition resulting quickly becomes a smooth mirror-surface deposit that slowly builds up free from surface irregularities, such as defects, voids, or protrusions. The process is stopped when the desired deposit thickness is achieved. Both the ion-beam-sputter polishing and the deposition should be stopped simultaneously when the desired coating thickness is achieved.

Eleven other adaptations to the process are the following:

(1) The mirror material can be metals, metal carbides, metal oxides, certain polymers, and most inorganic compounds.

(2) The deposition can be produced by vapor deposition from electron-beam evaporators, heated boats, filaments, or by ion-beam, dc or rf sputtering from the same or another deposition system.

(3) The substrate surface angle $\theta$ can be varied from 0° to ≥10° with respect to the incident ions, depending on the degree of polishing desired.

(4) The surface can be cooled or heated as desired to improve the quality of the coating.

(5) The ion energy can be varied from a few hundred to a few thousand electronvolts.

(6) The sputtering ions can be a variety of gas species, such as Ar, Xe, Ne, and N.

(7) The duration of initial sputter-etch cleaning can be varied from seconds to hours, depending on the thickness desired.

(8) The ion beam can be of various diameters, from centimeters to meters, depending upon the substrate diameter.

(9) The deposited coating can be a variety of materials, such as metals, metal oxides, and metal nitrides. The deposited materials can also be a different material than the substrate.

(10) The rotation rate of the substrate can be varied as desired to produce the best-quality surface coating.

(11) Further coatings can be applied by the same process if one wants to switch materials for the outer substrate surface.

The process is particularly adaptable to the polishing of various substrates for optical or esthetic purposes.

This work was done by Bruce A. Banks of Lewis Research Center. Refer to LEW-13545. Further information may be found in NASA TM-81679 [N81-19278/NSP] "Simultaneous Ion Sputter Polishing and Deposition," available from NTIS for $7.00 prepaid.

*continued on page 3131*