J Comput Electron (2012) 11:405–413
DOI 10.1007/s10825-012-0420-0

Quadruple-valued logic system using savart plate and spatial light modulator (SLM) and it's applications

Amal K. Ghosh · Animesh Bhattacharya · Amitabha Basuray

Published online: 6 September 2012

Abstract With the demanding scenario of communication and optical computing technology the trinary and quadruple valued logic systems are the most important ones in the many valued logic system. Different techniques are being proposed day-by-day to implement the multi-valued logics (MVL). In our previous papers we have proposed the modified trinary number (MTN) systems using savart plates and spatial light modulators (SLM). In this paper we have communicated the quadruple valued logic system using di-bit concept and their implementations to meet up the tremendous needs of speeds by exploiting the advantages of savart plates and spatial light modulators (SLM) in the optical tree architecture (OTA).

Keywords SLM · Quadruple · Di-bit

1 Introduction

During the last thirty years due to the needs of tremendous operational speed and processing a number of data, many new ideas are being floated in the field of computing. These include exploration of implementation of optical processor for switches in one hand and on the other hand the logical development from binary to multivalued logic are also being included in their field of activities. Though the major attraction for optical processors lies in the parallel operation but it was also felt that it is possible to implement multivalued logic in optical system using the polarization states of light beam along with the presence or absence of light [1]. The parallelism of optical beam could not be properly utilized using cascaded single-bit operating units therefore a signed digit number system was initiated with the pioneering works of Avizienis [2]. The carry free operation was also suggested using a modified signed digit [3–5] or modified trinary [6] system. The demand for implementations of such gates has also extended the activities in the field [7–12].

However, Lukasiewicz [13] who initiated the use of ternary logic based on three states has modified it later [14–17] with an idea that four states logic is a much better proposition. This paper is an extension of the modified trinary system [18–21] to a quadruple-valued logic system along with its implementation. In the implementation the different states are represented with a dibit representation using presence and absence of light of two orthogonal polarization states of light beam.

2 Quadruple valued logic representations and the system

The four-state representations of the quadruple valued logic system may be classified as the true, partly true, partly false and the false. In this case we have considered these four states explicitly as {0, 1, 2, 3} and their di-bit representations

A.K. Ghosh (✉)
Netaji Subhash Engineering College, Techno City, Garia, Kolkata 700152, India
e-mail: amal_k_ghosh@rediffmail.com

A. Bhattacharya
Department of Applied Electronics & Instrumentation Engineering, Netaji Subhash Engineering College, Techno City, Garia, Kolkata 700152, India
e-mail: animeshb_17@yahoo.com

A. Basuray
Department of Applied Optics & Photonics, University of Calcutta, 92, A.P.C.Road, Kolkata 700009, India
e-mail: abasuray@rediffmail.com

Table 1 Quadruple-valued logic system

Logical state	Represented by	Dibit representation	State of polarization
False/Wrong information	0	00	No light
Partial information	1	01	Vertical polarization
Partial information (complement of 1)	2	10	Horizontal polarization
True/Complete information	3	11	Presence of both the horizontal & vertical polarization

as $\{00, 01, 10, 11\}$. It is to be noted here that the four valued system with states $\{0, 1, 2, 3\}$ does not satisfy the basic field conditions whereas as a dibit representation of the form $00 \to 0$, $01 \to 1$, $10 \to 2$ and $11 \to 3$ may be used to represent a four valued logic where the basic two valued logic are applicable. As four is not a prime number, it cannot be considered as a field nevertheless this can be included in Galois Field GF(k^r), where k is a prime number and r is a positive integer [22]. The logical states, their representations and corresponding dibit representations and the state of polarization is given in the Table 1.

3 Truth tables based on di-bit representation

The basic logical operations with dibit representation as mentioned in the earlier section may be expressed in the following fashion. In the present system the normal logical gates e.g., OR, AND, NOT, XOR, NAND, NOR and XNOR may be represented bit-wise. The truth table for these conventional bit wise logic gates are represented in Table 2.

It is interesting to note at this point that the addition and multiplication are not simple bit-wise XOR and AND operations, these operations are performed in bit serial fashion. This is apparent from the truth table given in Table 3. In binary system the XOR gate is also the modulo-2 gate and thus gives the addition, which is not true in case of dibit logic gates based on binary logic for each bit as in such cases, XOR operation is not the modulo-4 gate. Similarly, AND gate defined in Table 2 does not represent either the multiplication logic or generates the carry bit. Then four-valued logic system calls for a more number of gates and the mathematical equations are to be developed using bit serial fashion. The most important mathematical gates i.e., the addition gate and the multiplication gates may be defined in the following fashion.

For addition gate,

if $\quad a_j a_i + b_j b_i = c_j c_i,$

then $\quad c_i = a_i \text{ XOR } b_i$

and $\quad c_j = (a_i \text{ AND } b_i) \text{ XOR } (a_j \text{ XOR } b_j)$

where " $+$ " stands for addition.

Table 2 Truth tables for (a) OR, (b) AND, (c) NOT, (d) XOR, (e) NAND, (f) NOR and (g) XNOR gates

B \ A	00	01	10	11
00	00	01	10	11
01	01	01	11	11
10	10	11	10	11
11	11	11	11	11

(a)

B \ A	00	01	10	11
00	00	00	00	00
01	00	01	00	01
10	00	00	10	10
11	00	01	10	11

(b)

A	$\bar{A}$
00	11
01	10
10	01
11	00

(c)

B \ A	00	01	10	11
00	00	01	10	11
01	01	00	11	10
10	10	11	00	01
11	11	10	01	00

(d)

B \ A	00	01	10	11
00	11	11	11	11
01	11	10	11	10
10	11	11	01	01
11	11	10	01	00

(e)

B \ A	00	01	10	11
00	11	10	01	00
01	10	10	00	00
10	01	00	01	00
11	00	00	00	00

(f)

B \ A	00	01	10	11
00	11	10	01	00
01	10	11	00	01
10	01	00	11	10
11	00	01	10	11

(g)

Similarly, the multiplication gate

$a_j a_i \cdot b_j b_i = d_j d_i$

then $\quad d_i = a_i \text{ AND } b_i$

and $\quad d_j = (a_i \text{ AND } b_j) \text{ XOR } (a_j \text{ AND } b_i)$

The corresponding truth tables for the addition and multiplication are given in Tables 3(a) and 3(b) respectively. It is to be noted that addition is obtained by modulo-4 gate.

Thus the logical and mathematical operations over GF(2^m) field can be subdivided in different classes. Some operations are bit-wise but for others it is in bit serial fashion, in which some information are carried over from the results of earlier bit.

Table 3 Truth tables for (a) addition and (b) multiplication

B \ A	00	01	10	11
00	00	01	10	11
01	01	10	11	00
10	10	11	00	01
11	11	00	01	10

(a)

B \ A	00	01	10	11
00	00	00	00	00
01	00	01	10	11
10	00	10	00	10
11	00	11	10	01

(b)

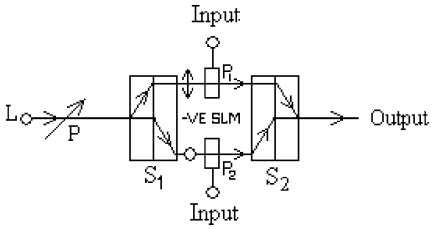

Fig. 1 Basic building block

4 Implementation of quadruple valued logics using opto-electronics systems

In optics, using two orthogonal states of polarization as well as that of absence and presence of light at a time we can generate four-state logic system using dibit representation. In this chapter the logic used has been extended to incorporate dibit representation in quadruple valued system.

4.1 The basic building block

The basic building block to implement the logical operations in quadruple valued logic system is shown in Fig. 1. Light from a laser source L after passing through the polarizer P is polarized at an angle 45° with respect to the two crystal axes and incident on the savart plate S_1 as shown in Fig. 1. The light incident on S_1 is splitted into two orthogonal components and comes out of S_1 with a spatial shift between them. The electrically addressable negative SLMs—P_1 and P_2 are then used for the controlling of two components of inputs beam. The nature of the negative SLM is such that it is transparent when there is no electric voltage applied on it and it becomes opaque when an electric voltage is applied on it. The property of positive SLM is just reverse. Hence the input may be considered as in the form of dibit (two bits) representation.

The second savart plate S_2 is then re-unites the two polarized beam for further operations. Various logic gates as mentioned in Table 2 and Table 3 may be implemented by combining this very basic module.

4.2 OR gate

The circuit diagram of OR Gate is shown in Fig. 2. The polarized parallel beam coming from the Laser source L

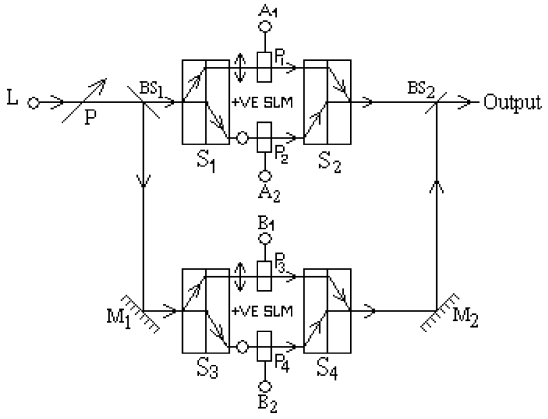

Fig. 2 OR gate

through polarizer P is incident on the beam splitter BS_1—where it is splitted into two directions as shown. One part is incident on the savart plate S_1 and the other part on the mirror M_1. The savart plate S_1 splits the beam into two orthogonal components—the p-polarization and the s-polarization. The input A (combination of A_1 and A_2) controls the positive SLMs P_1 and P_2 and accordingly the p-polarization and s-polarization come out of P_1 and P_2 and they recombined by the savart plate S_2 and incident on the beam splitter BS_2. On the other hand the ray reflected by the mirror M_1 is incident on the savart plate S_3 and by similar process it is also spatially modulated by the positive SLMs P_3 and P_4 depending on the input B (combinations of B_1 and B_2). Then the rays are re-united again by S_4 and after reflection by the mirror M_2 it is incident on the beam splitter BS_2. The output finally comes out from BS_2. We are considering the beams which are orthogonally polarized to each other so there will be incoherent superposition of intensity thus phase shift does not produce any effect.

For example, say A = 00 (i.e. $A_1 = 0$ and $A_2 = 0$), then the components of the light beam splitted by S_1 will be obstructed by P_1 and P_2 and hence the output of S_2 will be dark i.e., absence of p and s-polarizations. Say, the second input B = 11 (i.e., $B_1 = 1$ and $B_2 = 1$), then both the p and s-polarizations will pass through P_3 and P_4 and so the emerging ray from S_4 will contain both the p and s-polarizations. Hence the final output from BS_2 is 11—which follows the truth table for OR gate. Similarly, it is valid for other input combinations also as per Table 2(a).

4.3 AND gate

Figure 3 is the circuit diagram for the AND Gate. The polarized light beam incident on the first savart plate S_1 is splitted into the two orthogonal components with a spatial shift and modulated by the positive SLMs P_1 and P_2 by means of the input A (combination of A_1 and A_2) and they reunited again

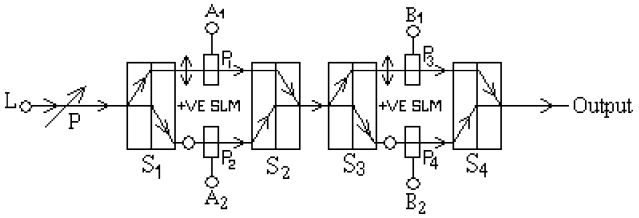

Fig. 3 AND gate

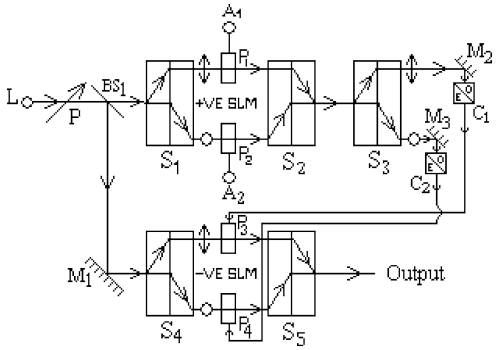

Fig. 4 NOT gate

at S_2. The emerging ray is then incident on the second building block and modulated again by the positive SLMs P_3 and P_4 as per the input B (combination of B_1 and B_2). Depending on A and B the output will occur according to the AND Gate truth table shown in Table 2(b).

4.4 NOT gate

The output obtained from this gate is simply the complement of the input. The circuit diagram is shown in Fig. 4. The savart plate S_1 decomposes the incident beam into two orthogonally polarized beams with a spatial shift. The only input A (combination of A_1 and A_2) is applied through the SLMs P_1 and P_2 and accordingly the essential components of the polarized beams are present in the light coming out of the savart plate S_2. Now it is allowed to pass through the savart plate S_3, where, the opto-electrical converters (O/E) C_1 and C_2 are used in the path of the rays to convert the light signal into electric voltage—which are then feed to the positive SLMs P_3 and P_4 to control the polarized components of the beam coming out of the savart plate S_4. The savart plate S_5 is then used to re-unite the final components present and to get the final output, which is simply the complement of the input A. Here, the output follows the truth table of NOT Gate as shown in Table 2(c).

For example, suppose A = 01 i.e., $A_1 = 0$ and $A_2 = 1$—hence, no voltage is applied at SLM P_1 but a voltage is applied at SLM P_2. As a result, no p-polarization is present from the output of P_1 but s-polarization is available from the output of P_2 and when they recombine at S_2, only s-polarization is available at the output of S_2. As a result when it is passes through S_3 only C_2 of the O/E converter will be activated and correspondingly P_4 will be energized but P_3 will remain at zero voltage. So the ray following the path BS_1, M_1 and savart plate S_4 will be modulated by the negative SLMs P_3 and P_4 such that only the p-polarization is present at the output coming out of S_5, which corresponds to 10. So it is simply the complemented output of the input. For other inputs also it follows the truth table of NOT gate as shown in Table 2(c).

4.5 Exclusive-OR (XOR) gate

The circuit diagram for the exclusive-OR (XOR) gate is shown in Fig. 5. The combination as shown follows the truth table for XOR Gate shown in Table 2(d). The two inputs are respectively A (combination of A_1 and A_2) and B (combination of B_1 and B_2).

4.6 NAND gate

The circuit diagram for the NAND gate is shown in the Fig. 6. The combination as shown follows the truth table for NAND gate given in Table 2(e).

4.7 NOR gate

The circuit diagram for the NOR gate is shown in the Fig. 7. The combination as shown follows the truth table for NOR gate as per Table 2(f).

5 Adder circuit

The adders of quadruple valued logics can also be classified into two categories—

(i) Half Adder and
(ii) Full Adder.

5.1 Quadruple-valued half adder

A half adder has two inputs and two outputs. The outputs are called Sum and Carry(Cout) respectively. The Sum output is the addition of two inputs and Carry is the carry out generated from the addition. The truth table for the half adder is given in the Table 4.

5.1.1 Truth tables of quadruple-valued half adder based on dibit representation

Here we can represent the inputs as well as the outputs in the dibit form. The dibit representations of the two inputs A and B and the outputs S and C are given below:

Inputs: $A = a_j a_i$ and $B = b_j b_i$

Output: Sum (Sum) $= S = s_j s_i$

Carry (Cout) $= C = c_j c_i$

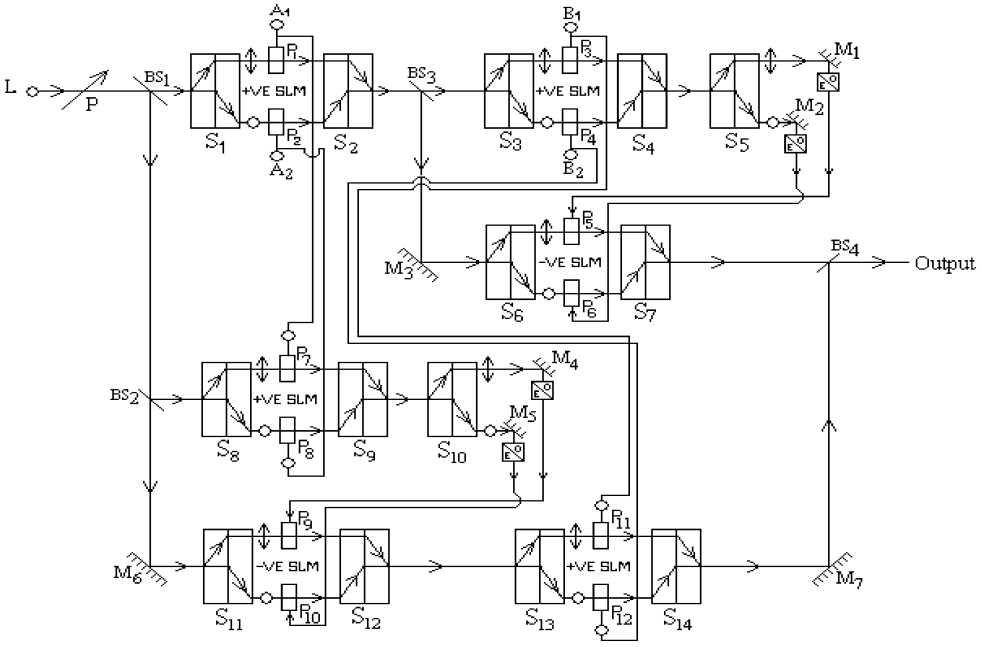

Fig. 5 XOR gate

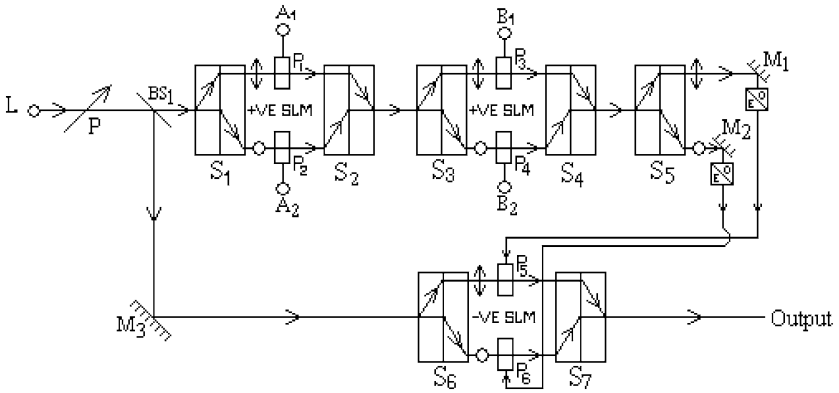

Fig. 6 NAND gate

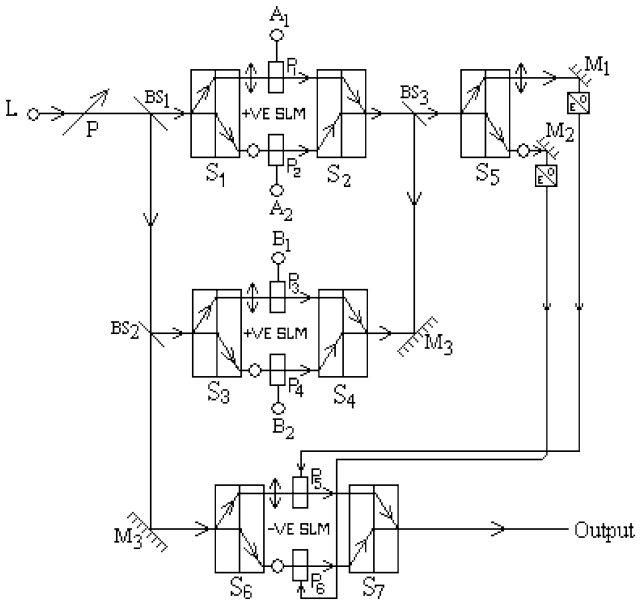

Fig. 7 NOR gate

The truth tables for the Sum and Carry in the dibit form are given in Table 5.

5.1.2 *Expressions for Sum and Carry*

From the truth Table 5 we can write the expressions for the Sum and Carry as follows:

$$s_i = a_i \text{ XOR } b_i$$

$$s_j = (a_i \text{ AND } b_i) \text{ XOR } (a_j \text{ XOR } b_j)$$

$$c_i = (a_j \text{ AND } b_j) \text{ OR } \left[(a_i \text{ AND } b_i) \text{ AND } (a_j \text{ OR } b_j)\right]$$

$$c_j = 0$$

5.1.3 *Block diagram of quadruple-valued half adder*

The block diagram of Half-Adder circuit is shown in Fig. 8. A and B are the two inputs and the outputs are Sum and Carry (Cout).

Table 4 Truth table for half adder

Input		Output	
B	A	Cout	Sum
0	0	0	0
0	1	0	1
0	2	0	2
0	3	0	3
1	0	0	1
1	1	0	2
1	2	0	3
1	3	1	0
2	0	0	2
2	1	0	3
2	2	1	0
2	3	1	1
3	0	0	3
3	1	1	0
3	2	1	1
3	3	1	2

5.2 Quadruple-valued full adder

The truth table for the quadruple-valued full adder is given in the Table 6. It is clear from the truth table that it is accomplished by four half adders and multiplexing their output based on the carry-in. The first half adder is identical to a normal half adder since the carry-in is 0. The second adder with carry-in = 1, the third adder with carry-in = 2 and the fourth adder with carry-in = 3.

5.2.1 Block diagram of quadruple-valued full adder

The block diagram of quadruple valued full-adder circuit is shown in Fig. 9. A and B are the two normal inputs and Cin is the other input and the outputs available from the full-adder circuits are Sum and Carry (Cout). This full adder circuit can be constructed from the half adder circuit and by using the logic gates as discussed earlier in this chapter.

6 Subtractor circuit

The subtractor of quadruple valued logics can also be classified into two categories—

(i) Half Subtractor and
(ii) Full Subtractor.

6.1 Quadruple-valued half subtractor

A half subtractor has two inputs and two outputs. The outputs are called Difference (D) and Borrow (Bout) respectively. The Difference output is the diffence of two inputs

Table 5 Truth tables for (a) Carry and (b) Sum

B \ A	00	01	10	11
00	00	00	00	00
01	00	00	00	01
10	00	00	01	01
11	00	01	01	01

(a)

B \ A	00	01	10	11
00	00	01	10	11
01	01	10	11	00
10	10	11	00	01
11	11	00	01	10

(b)

Fig. 8 Block diagram of quadruple-valued half adder

Table 6 Truth table for full adder

Input			Output	
C_{in}	B	A	Cout	Sum
0	0	0	0	0
0	0	1	0	1
0	0	2	0	2
0	0	3	0	3
0	1	0	0	1
0	1	1	0	2
0	1	2	0	3
0	1	3	1	0
0	2	0	0	2
0	2	1	0	3
0	2	2	1	0
0	2	3	1	1
0	3	0	0	3
0	3	1	1	0
0	3	2	1	1
0	3	3	1	2
1	0	0	0	1
1	0	1	0	2
1	0	2	0	3
1	0	3	1	0
1	1	0	0	2
1	1	1	0	3
1	1	2	1	0
1	1	3	1	1
1	2	0	0	3
1	2	1	1	0
1	2	2	1	1
1	2	3	1	2
1	3	0	1	0
1	3	1	1	1
1	3	2	1	2
1	3	3	1	3

Table 6 (*Continued*)

Input			Output	
C_{in}	B	A	Cout	Sum
2	0	0	0	2
2	0	1	0	3
2	0	2	1	0
2	0	3	1	1
2	1	0	0	3
2	1	1	1	0
2	1	2	1	1
2	1	3	1	2
2	2	0	1	0
2	2	1	1	1
2	2	2	1	2
2	2	3	1	3
2	3	0	1	1
2	3	1	1	2
2	3	2	1	3
2	3	3	2	0
3	0	0	0	3
3	0	1	1	0
3	0	2	1	1
3	0	3	1	2
3	1	0	1	0
3	1	1	1	1
3	1	2	1	2
3	1	3	1	3
3	2	0	1	1
3	2	1	1	2
3	2	2	1	3
3	2	3	2	0
3	3	0	1	2
3	3	1	1	3
3	3	2	2	0
3	3	3	2	1

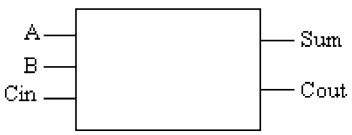

Fig. 9 Block diagram of quadruple-valued full adder

and Borrow is the barrow out generated from the subtraction. The truth table for the half subtractor is given in the Table 7.

Table 7 Truth table for half adder

Input		Output	
B	A	Bout	Difference (D)
0	0	0	0
0	1	1	3
0	2	1	2
0	3	1	1
1	0	0	1
1	1	0	0
1	2	1	3
1	3	1	2
2	0	0	2
2	1	0	1
2	2	0	0
2	3	1	3
3	0	0	3
3	1	0	2
3	2	0	1
3	3	0	0

Table 8 Truth tables for (a) borrow and (b) difference

B \ A	00	01	10	11
00	00	01	01	01
01	00	00	01	01
10	00	00	00	01
11	00	00	00	00

(a)

B \ A	00	01	10	11
00	00	11	10	01
01	01	00	11	10
10	10	01	00	11
11	11	10	01	00

(b)

6.1.1 Truth tables of quadruple-valued half adder based on dibit representation

Here we can represent the inputs as well as the outputs in the dibit form. The dibit representations of the two inputs A and B and the outputs D and Br are given below:

Inputs: $A = a_j a_i$ and $B = b_j b_i$

Output: Difference (D) $= D = d_j d_i$

Borrow (Bout) $= B_r = b_{rj} b_{ri}$

The truth tables for the Difference and Borrow in the dibit form are given in Table 8.

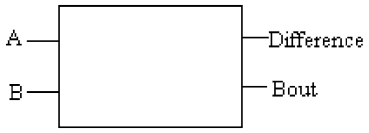

Fig. 10 Block diagram of quadruple-valued half subtractor

6.1.2 *Expressions for sum and carry*

From the truth Table 5 we can write the expressions for the Difference and Borrow as follows

$$B_{ri} = (\bar{b}_j \text{ AND } \bar{b}_i \text{ AND } a_i) + \left[b_j \text{ OR } (\bar{b}_i \text{ AND } a_i)\right] \text{AND } a_j$$

$$B_{rj} = 0$$

$$D_i = \bar{b}_i \text{ XOR } a_i$$

$$D_j = b_i \text{ AND } (b_j \text{ XOR } a_j) + (b_j \text{ AND } \bar{a}_j \text{ AND } \bar{a}_i) + \left[\left(\bar{b}_i \text{ AND } (b_j \text{ XNOR } a_j)\right) \text{ OR } (b_j \text{ AND } a_j)\right] \text{ AND } a_i$$

6.1.3 *Block diagram of quadruple-valued half subtractor*

The block diagram of Half-Subtractor circuit is shown in Fig. 10. A and B are the two inputs and the outputs are Difference and Borrow (Bout).

6.2 Quadruple-valued full subtractor

The truth table for the quadruple-valued full subtractor is given in the Table 9. It is clear from the truth table that it is accomplished by four half subtractors and multiplexing their output based on the borrow-in.

6.2.1 *Block diagram of quadruple-valued full subtractor*

The block diagram of quadruple valued full-Subtractor circuit is shown in Fig. 11. A and B are the two normal inputs and Bin is the other input and the outputs available from the full-subtractor circuits are Difference and Borrow (Bout).

7 Conclusions

In this paper we have discussed the very basic quadruple-valued logic systems and their practical implementations by using the Electro-Optic Technique (EOT) with opto-electronic devices for the fast operation. The dibit representation of this logic helps to implement the system in a simpler manner. In ideal situations the optical devices, mirror, SLM, polarizer etc. in the logic gates have no insertion losses. However, in realistic situations, these losses are important. The polarizer used at the input end absorbs much energy but this polarizer may be avoided if one takes a polarized laser. The main losses are at the switches i.e. at the

Table 9 Truth table for full subtractor

Input			Output	
B_{in}	B	A	Bout	Difference
0	0	0	0	0
0	0	1	1	3
0	0	2	1	2
0	0	3	1	1
0	1	0	0	1
0	1	1	0	0
0	1	2	1	3
0	1	3	1	2
0	2	0	0	2
0	2	1	0	1
0	2	2	0	0
0	2	3	1	3
0	3	0	0	3
0	3	1	0	2
0	3	2	0	1
0	3	3	0	0
1	0	0	0	1
1	0	1	1	2
1	0	2	1	1
1	0	3	1	0
1	1	0	0	0
1	1	1	1	3
1	1	2	1	2
1	1	3	1	1
1	2	0	0	1
1	2	1	0	0
1	2	2	1	1
1	2	3	1	2
1	3	0	0	2
1	3	1	0	1
1	3	2	0	0
1	3	3	1	3
2	0	0	1	2
2	0	1	1	1
2	0	2	1	0
2	0	3	2	3
2	1	0	1	3
2	1	1	1	2
2	1	2	1	1
2	1	3	1	0
2	2	0	0	0
2	2	1	1	3
2	2	2	1	2
2	2	3	1	1
2	3	0	0	1
2	3	1	0	0
2	3	2	1	3
2	3	3	1	2

Table 9 (*Continued*)

Input			Output	
B_{in}	B	A	Bout	Difference
3	0	0	1	1
3	0	1	1	0
3	0	2	2	3
3	0	3	2	2
3	1	0	1	2
3	1	1	1	1
3	1	2	1	0
3	1	3	2	3
3	2	0	1	3
3	2	1	1	2
3	2	2	1	1
3	2	3	1	0
3	3	0	0	0
3	3	1	1	3
3	3	2	1	2
3	3	3	1	1

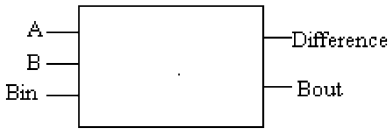

Fig. 11 Block diagram of quadruple-valued full subtractor

SLMs, the losses incurred is not much as the SLMs are used in two modes either it allows the beam or not. The losses incurred are very much comparable to switches used in normal binary photonics switches. The devices discussed here are the passive components—so the bandwidth depends on the switching time of the SLMs and now-a-days very fast optical switches are also available. Moreover, the operating wavelength mainly depends on the selection of Savart plates and SLMs, which are available from visible to IR (Infra-Red) region, but the light must be highly monochromatic for proper performance of the optical components.

The purpose of this study is to explore the quadruple logic system with opto-electronic implementation for the very explicit potential areas like, grey image processing, fuzzy logic implementations, fractal formations and any other emerging areas where the fast operations are needed. Though in optoelectronic implementations, the optical parallelism is sacrificed due to the connections amongst cells, however, the advantages of four-state implementation make it possible to handle more information at a time.

References

1. Lohmann, A.W.: Polarization and optical logic. Appl. Opt. **25**, 1594–1597 (1986)
2. Avizienis, A.: Signed-digit number representation for fast parallel arithmetic. IRE Trans. Electron. Comput. **EC-10**, 389–400 (1961)
3. Drake, B.L., Bocker, R.P., Lasher, M.E., Patterson, R.H., Miceli, W.J.: Photonic computing using the modified-signed-digit number representation. Opt. Eng. **25**, 38–43 (1986)
4. Bocker, R.P., Drake, B.L., Lasher, M.E., Henderson, T.B.: Modified signed-digit addition and subtraction using optical symbolic substitution. Appl. Opt. **25**(15), 2456–2457 (1986)
5. Cherri, A.K., Karim, M.A.: Modified signed digit arithmetic using an efficient symbolic substitution. Appl. Opt. **27**(18), 3824–3827 (1988)
6. Datta, A.K., Basuray, A., Mukhopadhyay, S.: Arithmetic operations in optical computations using a modified trinary number system. Opt. Lett. **14**, 426–428 (1989)
7. Cherri, A.K.: Designs of optoelectronic trinary signed-digit multiplication by use of joint spatial encodings and optical correlation. Appl. Opt. **38**, 828–837 (1999)
8. Alam, M.S., Karim, M.A., Awwal, A.A.S., Westerkamp, J.J.: Optical processing based on conditional higher-order trinary modified signed-digit symbolic substitution. Appl. Opt. **31**(26), 5614–5621 (1992)
9. Fyath, R.S., Alsaffar, A.A.W., Alam, M.S.: Optical two-step modified signed-digit addition based on binary logic gates. Opt. Commun. **208**(4), 263–273 (2002)
10. Fyath, R.S., Alsaffar, A.A.W., Alam, M.S.: Nonrecoded trinary signed-digit multiplication based on digit grouping and pixel assignment. Opt. Commun. **230**(1–3), 35–44 (2004)
11. Basuray, A., Mukhopadhyay, S., Ghosh, H.K., Dutta, A.K.: A tristate optical logic system. Opt. Commun. **85**, 167–170 (1991)
12. Lin, S., Kumazawa, I.: Optical fuzzy image processing based on shadow casting. Opt. Commun. **94**, 397–405 (1992)
13. Łukasiewicz, J.: O logice trojwartosciowej. Ruch Filoz. **5**, 169–171 (1920) [English translation in: Łukasiewicz [15]]
14. Lukasiewicz, J.: Aristotles Syllogistic from the Stand Point of Modern Formal Logic. Oxford, London (1951)
15. Lukasiewicz, J.: Selected Works, Borkowski, L. (ed.). North-Holland, Amsterdam and PWN, Warsaw (1970)
16. Wojcicki, R., Malinowski, G. (eds.): Selected Papers on Łukasiewicz Sentential Calculi. Ossolineum, Wroclaw (1977)
17. Lukasiewicz, J., Tarski, A.: Untersuchungen uber den Aussagenkalkul. C. R. Soc. Sci. Lett. Vars. **23**(iii), 30–50 (1930)
18. Ghosh, A.K., Choudhury, P.P., Basuray, A.: Modified trinary optical logic gates and their applications in optical computation. Presented in the Third International Conference on CISSE 2007, IEEE, University of Bridgeport, 3–12 December 2007 (2007) and published in the proceeding on Innovations and Advanced Techniques in Systems, Computing Sciences and Software, pp. 87–92. Springer (2008)
19. Ghosh, A.K.: Parity generator and parity checker in the modified trinary number system using savart plate and spatial light modulator. Optoelectron. Lett. **6**(5), 325–327 (2010)
20. Ghosh, A.K., Basuray, A.: Binary to modified trinary number system conversion and vice-versa for optical super computing. Nat. Comput. **9**(4), 917–934 (2010). doi:10.1007/s11047-009-9166-4, November 2009, Special issue "Optical Supercomputing"
21. Ghosh, A.K., Bhattacharya, A., Raul, M., Basuray, A.: Trinary arithmetic and logic unit (TALU) using savart plate and spatial light modulator (SLM) suitable for optical computation in multi-valued logic. Opt. Laser Technol. **44**(5), 1583–1592 (2012)
22. Kalay, U., Perkowski, M.A., Hall, D.V.: Highly testable Boolean ring logic circuits. In: ISMVL 1999, pp. 268–274 (1999)