AN ALTERNATIVE APPROACH OF ALL-OPTICAL BINARY ADDITION WITH RESIDUE ARITHMETIC BY SOME LOGIC OPERATIONS

[1]Partha Pratim Das, [2]Manasi Jana (Das)

[1]School of Applied Science & Humanities, Haldia Institute of Technology
ICARE Complex, HIT, Haldia, Purba Medinipur
West Bengal 721 657, India

[2]Department of Master in Computer Applications
Haldia Institute of Technology, ICARE Complex, HIT
Haldia, Purba Medinipur, West Bengal 721 657, India

Abstract

Residue arithmetic has the traditional and useful advantages in the field of parallel computation. Several different techniques and methods have been proposed by the scientists and researchers from various arenas for this purpose [1, 3, 10].In this communication, the authors propose an alternative methodology of binary addition with some logical operations by the proper exploitation of residue arithmetic. All the operations have been done in parallel without any carry (carry-less). The conceptual implementation of addition with logic operations by residue arithmetic has been proposed. The authors extend this proposal in all-optical domain for the construction of super fast computation.

Keywords: Residue arithmetic, Moduli, Decimal, Binary, Parallel computation, Carry, Logic operations, All-optical domain.

1. Introduction:

Residue arithmetic works basically on the moduli by which the decimal numbers are converted to its residue. Several researchers and technologists have proposed different techniques of residue arithmetic based mathematical and logical functioning for parallel handling of various arithmetic and logic operations [4].

In this communication, the authors proposed a completely different methodology by single moduli and irrespective of choice of moduli (it may be prime or non-prime) to obtain the addition of two decimal numbers by the logical operations (OR and AND) with the help of residue arithmetic. The maximum decimal number that can be handled by this methodology is irrespective of choice of the assumed moduli. Mathematically, we may deal with any decimal integer number between 0 to Z with the moduli m_i (i = 1 to n) by the equation

$Z = [(\Pi\, m_i) – 1]$, where I takes the value from 1 to n.

The maximum decimal integer number that can be handled is limited by the choice of moduli according to the above equation. But in this communication we proposed a completely different approach where we can chose the moduli without any limitation. If we chose the moduli greater than the two decimal numbers that will be added, the addition process will be simpler and it takes less time. Otherwise, it will increase the process complexities and the advantages of residue technique will be then lost.

All the operations proposed in this communication are completely carry-less parallel handling in binary domain.

The authors also extend the idea in all-optical domain for the development of super fast computing device.

2. Binary Addition with Residue Arithmetic by Some Logic Operations

Let us consider two decimal integer numbers X and Y. The sum of these two decimal numbers be Z and we will try to find the value of Z with residue arithmetic by OR, AND logical operation. Now, we consider only one decimal integer moduli m (it may be prime or non-prime, less or greater than X and Y).

Suppose the residues of the decimal integer X and Y under the moduli m are x and y. Now, we convert x and y in binary and operate logical OR and logical AND operation between binary x and y. Suppose the OR and AND operation

between x and y is a and b respectively in decimal. Now, we will make a series starting from a (OR result of the residues of X and Y under moduli
m) and expand the series up to maximum of Z following as a, a + m, a + 2m, a + 3m and so on. Similarly, we will make a series starting from b (AND result of the residues of X and Y under moduli m) and expand the series up to maximum Z following as b, b + m, b + 2m, b + 3m and so on.
From the above two series, we will see that sum of the maximum number arises in OR series and minimum number arises in AND series be the addition result of X and Y (i.e., Z). Consequently, the sum of minimum number arises in OR series and maximum number arises in AND series also be the addition result of X and Y (i.e., Z). This case may arise when m is less than X and Y (i.e., m < X and Y). If there is only one number arises in each of OR and AND series then the sum of these two numbers will be the addition result of X and Y (i.e., Z). This case may arise when m is greater than Z (i.e., m > Z). If there is only one number arises in OR series and more than one number arises in AND series then the sum of the number arises in OR series and the minimum number arises in AND series will be the addition result of X and Y (i.e., Z). This case may arise when m is greater than X and Y but less than Z (i.e., Z > m > X and Y).

Example 1: Let us consider X = 25, Y = 22. Thus, Z (X + Y) = 47. Suppose, m = 6[m < X and Y]. Residues of X (= 25) and Y (= 22) under moduli m (= 6) are 1 (1 in binary) and 4 (100 in binary) respectively. Now, we operate OR and AND operation of these two residues under the moduli m (= 6). The logical OR operation of residue 1 (in binary) and 100 (in binary) is 101 (= 5 in decimal). Thus the OR series under moduli m (= 6) will be 5, 11, 17, 23, 29, 35, 41, 47. As the next number of that series will be 47 + 6 = 53, we could not consider it. Because, here we can expand the series up to maximum of Z (= 47).
Similarly, the logical AND operation of residue 1 (in binary) and 100 (in binary) is 0 (= 0 in decimal). Thus the AND series under moduli m (= 6) will be 0, 6, 12, 18, 24, 30, 36, 42. As the next number of that series will be 42 + 6 = 48, we could not consider it. Because, here we can expand the series up to maximum of Z (= 47).
From the above two series we see that sum of the maximum number arises in OR series (= 47) and minimum number arises in AND series (= 0) is the addition result of X and Y (i.e., 47 + 0 = 47 = Z). Consequently, from the two series we see that sum of the minimum number arises in OR series (= 5) and maximum number arises in AND series (= 42) is the addition result of X and Y (i.e., 5 + 42 = 47 = Z).

Example 2: Let us consider X = 25, Y = 22. Thus, Z (X + Y) = 47. Suppose, m = 48[m > Z]. Residues of X (= 25) and Y (= 22) under moduli m (= 48) are 25 (11001 in binary) and 22 (10110 in binary) respectively. Now, we operate OR and AND operation of these two residues under the moduli m (= 48). The logical OR operation of residue 11001 (in binary) and 10110 (in binary) is 11111 (= 31 in decimal). Thus the OR series under moduli m (= 48) will be 31. As the next number of that series will be 31 + 48 = 79, we could not consider it. Because, here we can expand the series up to maximum of Z (= 47).
Similarly, the logical AND operation of residue 11001 (in binary) and 10110 (in binary) is 10000 (= 16 in decimal). Thus the AND series under moduli m (= 48) will be 16. As the next number of that series will be 16 + 48 = 64, we could not consider it. Because, here we can expand the series up to maximum of Z (= 47).
From the above two series we see that sum of only number arises in OR series (= 31) and only number arises in AND series (= 16) is the addition result of X and Y (i.e., 31 + 16 = 47 = Z).

Example 3: Let us consider again X = 25, Y = 22. Thus, Z (X + Y) = 47. Suppose, m = 26[Z > m > X and Y]. Residues of X (= 25) and Y (= 22) under moduli m (= 26) are 25 (11001 in binary) and 22 (10110 in binary) respectively. Now, we operate OR and AND operation of these two residues under the moduli m (= 26). The logical OR operation of residue 11001 (in binary) and 10110 (in binary) is 11111 (= 31 in decimal). Thus the OR series under moduli m (= 26) will be 31 only. As the next number of that series will be 31 + 26 = 57, we could not consider it. Because, here we can expand the series up to maximum of Z (= 47).
Similarly, the logical AND operation of residue 11001 (in binary) and 10110 (in binary) is 10000 (= 16 in decimal). Thus the AND series under moduli m (= 26) will be 16, 42. As the next number of that series will be 42 + 26 = 68, we could not consider it. Because, here we can expand the series up to maximum of Z (= 47).
From the above two series we see that sum of the only number arises in OR series (= 31) and minimum number arises in AND series (= 16) is the addition result of X and Y (i.e., 31 + 16 = 47 = Z).

3. Optical Implementation

The logical operations (OR and AND) of the residues under moduli m may be done by using opto-electronic switching technique or by all-optical switching (using non-linear switching technique) [2, 8, 11]. As there are more delay time to operate logical operations by using opto-electronic switching technique, the authors emphasizes on all-optical methodology by the proper exploitation of non-linear material based switching. There are several research papers from several researchers and technologists of our globe on non-linear based switching technique for logical operations as well as memory units (flip-flops). Thus, if we construct an all-optical logic gates by the proper exploitation of non-linear material (e.g. KH_2PO_4, $LiNbO_3$) based switching technique, we can add decimal integer numbers of infinite domain following the above mentioned methodology [5,6,7, 9]. As the process is all-optical, there is no delay time and hence this may be the door step to construct a super fast computing system.

We know that the refractive index (RI) of some useful linear –nonlinear material composite is defined as:

$n = n_L (1 + n_{NL}I_0)$, where n, n_L and n_{NL} are the refractive indices of linear-nonlinear material composite, linear material and nonlinear material respectively. Here, I_0 is the input light intensity.

Now, we will try to construct an all-optical OR and AND logic gates with the proper exploitation of nonlinear material based switching. The schematic diagram of all-optical OR logic gate is given in figure 1.

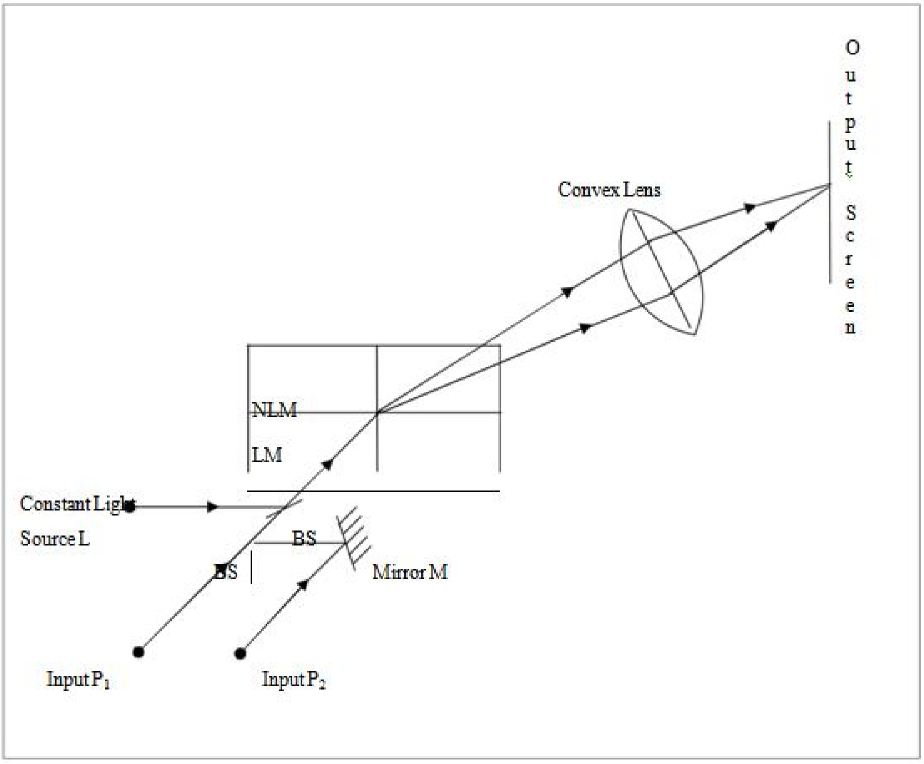

Figure 1: Schematic of All-optical OR Logic Gate

Starting from an initial intensity upon a certain value of optical signal at the inputs we get the output signal intensity as desired. Let P_1 and P_2 are two inputs. L is a constant light source of intensity I. Now we will perform the following operations:

i) When input P_1 = 0(0), input P_2 = 0(0), only the constant light source is present in the input end of intensity I. Then the contribution of the light source of intensity I is extended in the input end A of the linear-nonlinear (L / NL) based system. For this condition, the emergent beam (i.e., output light beam) is not contained within the aperture condition of the convex lens, i.e., there is no output signal on the screen and the output will be zero (0).

ii) When input P_1 = 0(0), input P_2 = 1(I). Then the contribution of two light sources (one is constant light source and another is input P_2) of intensity 2I is extended in the input end A of the linear-nonlinear (L / NL) based system. For this condition, the emergent beam (i.e., output light beam) must be contained within the aperture condition of the convex lens, i.e., output signal on the screen will be present and the output is 1(2I).

iii) When input P_1 = 1(I), input P_2 = 0(0). Then the contribution of two light sources (one is constant light source and another is input P_1) of intensity 2I is extended in the input end A of the linear-nonlinear (L / NL) based system. For this condition, the emergent beam (i.e., output light beam) must be contained within the aperture condition of the convex lens, i.e., output signal on the screen will be present and the output is 1(2I).

iv) When input P_1 = 1(I), input P_2 = 1(I). Then the contribution of three light sources (constant light source, input P_1 and input P_2) of intensity 3I is extended in the input end A of the linear-nonlinear (L / NL) based system. For this condition, the emergent beam (i.e., output light beam) must be contained within the aperture condition of the convex lens, i.e., output signal on the screen will be present and the output is 1(3I). The truth table of all-optical OR logic gate is given in table 1.

Table 1

Input P_1	Input P_2	Output (O/P)
0 (0)	0 (0)	0 (I)
0 (0)	1 (I)	1 (2I)
1 (I)	0 (0)	1 (2I)
1 (I)	1 (I)	1 (3I)

Truth table of all-optical OR logic gate

Following the same procedure, we can construct an all-optical AND logic gate. The schematic diagram will be same as all-optical OR logic gate, but the aperture condition of convex lens will be different. For the all-optical AND logic gate, we will set a convex lens of aperture condition such that output signal will present on the screen when input P_1 = 1(I), input P_2 = 1(I) and the constant light source L is present. For this condition the output will be 1(3I). For the other three cases, i.e., when P_1 = 0(0), P_2 = 0(0), L = 1(I), when P_1 = 0(0), P_2 = 1(I), L = 1(I) and P_1 = 1(I),
P_2 = 0(0), L = 1(I), no output signal is present in the screen. Hence, the outputs are zero (0). The truth table of all-optical AND logic gate is given in table 2.

Table 2

Input P_1	Input P_2	Output (O/P)
0 (0)	0 (0)	0 (I)
0 (0)	1 (I)	0 (2I)
1 (I)	0 (0)	0 (2I)
1 (I)	1 (I)	1 (3I)

Truth table of all-optical AND logic gate

In all-optical OR and AND logic gate, all the operations are performed in optical domain and real time.

4. Conclusion

Research on residue arithmetic, digital optical processor and its optical implementation was a challenge to the researchers and scientists in the decade of eighties and early of nineties of last century for the construction of super fast computer. Though some theoretical supports were ready, due to the no availability of some optical devices these have no experimental support and remained in the cold room. The authors seem that it has till some importance and can practically establish.

The methodology described in this communication is a completely different approach of binary data addition. It can be said that the conventional restriction of selection of moduli in residue arithmetic has been avoided. The range of dealing of decimal numbers can be increased as we desired. The two series under moduli m (OR and AND series) will be simpler and smaller if we chose the moduli m greater than the addition result of X and Y (i.e. Z). Thus it takes less time for the operation.

Finally, we can get the binary data addition with some logic operations (OR and AND) by the proper exploitation of residue arithmetic. All the process has been made in parallel without carry. We also extend this proposal in all-optical domain for the construction of super fast computation. It is noticed that the X-OR operation of any two binary numbers is nothing but the OR operation of the two said binary numbers minus the AND operation of the two said binary numbers. This noticeable point may help us for further expansion of binary data addition as well as binary data subtraction by logical operation with residue arithmetic in all-optical domain.
For extending the development to achieve a super-fast computing system, first we developed this new methodology. Our next step will be to construct all-optical memory cell (flip-flop) in optical domain, which must run in real time.
The most advantages of this methodology are the super fast capability of addition as it run in optical domain and in real time. The limitation of this methodology is the availability of such non-linear material.

References:

[1] A.Huang, Y.Tsunoda, J.W.Goodman and S.Ishihara, "Optical computation using residue arithmetic," Appl. Opt., 18, 149 (1979).
[2] A.P.Goutzoulis, E.C.Malarkey, D.K.Davies, J.C.Bradley and P.R.Beaudet, "Optical processing with residue LED/LD lookup tables," Appl.Opt., 9, 27, (1988).
[3] S.Mukhopadhyay, A.Basuray and A.K.Dutta, "New technique of arithmetic operation using positional residue system," Appl. Opt., 20, 29, (1990).
[4] S.Mukhopadhyay, P.P.Das et~al, "Use of power based moduli in residue arithmetic," Comm. in Inst., 3 (3), 141 (1995).
[5] T.Yatagi, "Optical space-variant logic gate array based on spatial encoding technique," Opt. Lett., 11, 260 (1986).
[6] J.Tanida and Y.Ichioka, "Optical parallel array logic system," Appl.Opt., 25, 1565 (1986).
[7] J.Tanida and Y.Ichioka, "Optical logic array processor using shadowgrams," Jr. of Optical Society of America, 73, 800 (1983).
[8] G.E.Lohmann and K.H.Brenner, "Space-variance in optical computing," Optik, 89 (3), 123 (1992)
[9] K.W.Wang and L.M.Cheng, "Space-variant optical logic operations based on operation dependent encoding method," Appl.Opt., 33(11), 2134 (1994).
[10]P.P.Das, P.Ghosh and S.Mukhopadhyay, "A new proposal of optical flip-flop using residue arithmetic," SPIE Proc. 4534, 149 (2001).
[11]11. P.P.Das and D.Giri, "A new alternative approach of all-optical flip-flop with non-linear material," Opt.Engg. Vol. 49 (07), 2010.

About the Authors:

Partha Pratim Das received his BS and MS degrees in Physics with first class in 1993 and 1995 from Vidyasagar University. He also received Ph. D. degree in Physics in the year 2003. He is now working as Professor in Physics in the School of Applied Science & Humanities, Haldia Institute of Technology, Haldia, Purba Medinipur, West Bengal – 721 657, India. His research fields include Optical Parallel Computation, Non- linear Optics and contributed more than 30 research papers in reputed International/National standard journals/symposium proceedings. He has delivered several lectures in International/National standard conferences/seminars.

Manasi Jana (Das) received his BS and MS degrees in Computer Science with first class in 1999 and 2002 from Vidyasagar University. He is now working as Assistant Professor in Computer Applications in the Department of Master in Computer Applications, Haldia Institute of Technology, Haldia, Purba Medinipur, West Bengal – 721 657, India. His research fields include Optical Parallel Computation, Neural Networking, and Soft Computing.