A two-step optical modified signed-digit adder for large-scale 2D data array using digit-decomposition-plane representation

Sabah S. Alsheraidah[1] Mohammed A. Alebadi[2] Alaa A. Al-Saffar[3]

[1]Department of Computer Systems, Basrah Technical Institute, E-mail: sabahhd@yahoo.com.
[2]Department of Computer Engineering, College of Engineering, Basrah University, Email: m.ali@compengbas.net
[3]Department of Electronics, Basrah Technical Institute, E-mail: alaaasaffar@yahoo.com.

Abstract:

In this paper, parallel optical array adder for large-scale 2D Modified Sign-Digit (MSD) data array is proposed and implemented to limit the carry propagation to constant steps. The digit-decomposition-plane (DDP) representation technique is expanded to code the 2D array of the MSD number system. The design is based on the logical formulas which are newly derived according to the fundamental parallel addition algorithm for MSD number system using the features of the DDP coding technique. The optical implementations scheme is based on classical optical elements such as spatial light modulators, beam combiners, beam splitters, mirrors, light source arrays, and light detector arrays. The proposed algorithm and its optical architecture have useful intrinsic characteristics such as ultra-high speed, constant processing time, and parallel computation on large-scale data arrays. The simulation results insure that the proposed arithmetic unit is worked successfully.

جامع إشارة– رقم معدل ضوئي ذو خطوتين لسلسلة بيانات كبيرة القياس ذات بعدين باستخدام تمثيل مستويات فصل الرقم

أ.م صباح سليم الشريدة (المعهد التقني/بصرة)

م. م. محمد عبد علي العبادي (جامعة البصرة– كلية الهندسة)

أ.م. د. علاء عبد الوهاب الصفار (المعهد التقني/بصرة)

الخلاصة:

في هذا البحث تم اقتراح وبناء جامع ضوئي متوازي لسلسلة بيانات كبيرة القياس ذات بعدين لغرض تحديد انتشار الحمل إلى خطوات ثابتة. لقد تم استخدام تقنية تمثيل مستويات فصل الرقم لتشفير سلسلة ذات بعدين متكونة من أرقام بنظام إشارة – رقم المعدل. اعتمد التصميم على المعدلات المنطقية التي اشتقت حديثاً على أساسيات خوارزمية الجمع المتوازي لأرقام نظام إشارة – رقم المعدل وباستخدام مميزات التشفير في تقنية تمثيل مستويات فصل الرقم. مخطط البناء الضوئي اعتمد على الأدوات الضوئية التقليدية مثل مضمنات الضوء التحويلية وجامعات الأشعة وفاصلات الأشعة والمرايا ومصادر ضوئية وكاشفات ضوئية. الخوارزميات المقترحة وبنائها الضوئي لها خصائص مفيدة مثل السرعة الفائقة ووقت معالجة ثابت وحسابات متوازية لكتل بيانية واسعة.

1- Introduction

Digital computer system has been suffered from the carry propagation cases that may appear in the intermediate steps of the arithmetic operations due to its sequential operation. The carry propagation makes the processing time depended on the length of the numbers under processing. Thus, the processing system will take a long computing time when the numbers are large in length, and vise versa. This problem has been solved by proposing many optical techniques which perform the arithmetic operations in parallel manner and constant processing time independent on the length of numbers.

Recently these optical techniques are established using many parallel algorithms in order to perform the arithmetic operations. Some of these parallel algorithms use the redundant binary numbers [1]. The other use a multi-leveled number system based on residue arithmetic algorithm [2]. The common technique in optical parallel arithmetic operations uses the ***"Signed-Digit*** (SD) ***number systems"*** [3]. Alspo, parallel optical logic gates are suggested to process large amounts of data in parallel and in speed of light [4].

Also optical coding schemes are investigated for coding the information and realizing the parallel algorithms such as, symbolic substitution (SS) [5], optical shadow casting (OSC) [6], content addressable memory (CAM) [7], optical correlation [8] and digit-decomposition planes [9].

Optical arithmetic operations of modified signed-digit number system is designed and implemented in parallel computing. The basic operations, which are addition, subtraction, multiplication and division, are more interested than the other operations. The rest arithmetic operations, such as square root, logarithm, exponential, etc., can be determined using the combinations of these basic operations [10].

For addition operation, three-step parallel algorithm is investigated for MSD numbers [6, 9] for limiting the carry propagation to only three steps. In order to reduce the computing time, two-step and one-step addition algorithms are also proposed by *Cherri* [5] for

MSD number to approach to the carry-free addition algorithms.

In this work, we design and implement a two-step optical parallel MSD adder. This proposed MSD adder can be used to process large-scale 2D MSD data array in parallel fashion and constant computing time. DDP representation will be applied to code large MSD array.

2- Signed-Digit Number Systems

Signed-digit number systems (SD) are widely used in optical systems. SD number systems overcame the carry propagation or limit it in constant steps [5].

The decimal number can be represented in SD number form as:

$$D = \sum_{i=0}^{n-1} x_i r^i \qquad (1)$$

Where;

D : the decimal number,

x_i : the i-th digit of SD number,

$x_i \in \{-\alpha, -(\alpha-1), \dots, (\alpha-1), \alpha\}, \alpha \leq r-1.$

r : the radix of SD number system,

n : number of digits in SD number.

The radix **r** can take many values; each one makes some changes in the SD number system. The common used values of **r** are {2, 3, and 4}. Table (1) shows the three SD number systems which have been widely used.

[illegible]	[illegible]	[illegible]	[illegible]
Modified Signed-Digit number system (MSD)	2	1	$\{\bar{1},0,1\}$
Trinary Signed-Digit number system (TSD)	3	2	$\{\bar{2},\bar{1},0,1,2\}$
Quaternary Signed-Digit number system (QSD)	4	3	$\{\bar{3},\bar{2},\bar{1},0,1,2,3\}$

Table (1): The three SD number systems.

Note that, -1, -2, and -3 are denoted by $\bar{1}$, $\bar{2}$, and $\bar{3}$, respectively. It is noted that, any SD number can be represented in more than one form, for example, the decimal number $(19)_{10}$ can be represented in SD form as:

$$(19)_{10} = (10011)_{MSD} = (1010\bar{1})_{MSD} = (10\bar{1}1)_{MSD} = (1\bar{1}\bar{1}1)_{MSD}$$

$$(19)_{10} = (0201)_{TSD} = (1\bar{1}01)_{TSD} = (1\bar{1}1\bar{2})_{TSD}$$

$$(19)_{10} = (01\bar{1}\bar{1})_{QSD} = (0103)_{QSD} = (02\bar{3}\bar{1})_{QSD} = (12\bar{3}\bar{1})_{QSD}$$

It is clear that the SD numbers arc redundant. This redundancy feature will make SD numbers very powerful to implement an optical parallel arithmetic unit with carry-free or carry-limited processing [11]. A higher radix **r** of SD number systems gives a large range of numbers with fewer SD digits comparing with the binary number system that will require a large number of binary digits to offer a large range of binary numbers.

3- Two-Step MSD Addition Algorithm

The two-step addition of two n-digit MSD numbers is executed by examining the values of (x_i, y_i) digits with (x_{i-1}, y_{i-1}) digits.

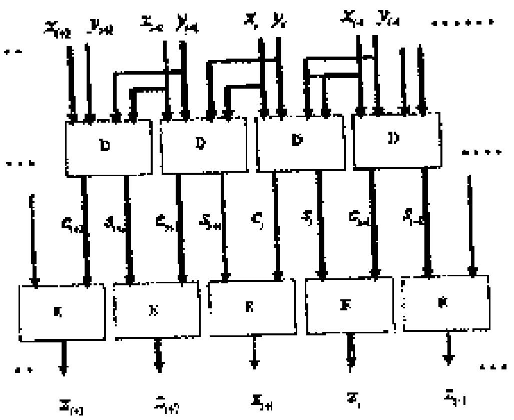


Fig. (1) : Block diagram of two-step MSD adder.

The (x_i, y_i) and (x_{i-1}, y_{i-1}) combinations are called the current pair CP and the next lower order pair NLOP of the addend $X(x_{n-1}, x_{n-2}, \ldots\ldots, x_1, x_0)$ and augend $Y(y_{n-1}, y_{n-2}, \ldots\ldots, y_1, y_0)$ numbers, respectively. Figure (1) presents the block diagram of two-step MSD adder [10], where boxes D and E represent single-digit MSD adders for the first-step and second-step, respectively. Equation (2) explains the operation of this block diagram.

step 1: $$x_i + y_i = 2c_i + s_i \qquad (2)$$

step 2: $$s_i + c_{i-1} = z_i$$

where,

x_i , y_i : i-th MSD digits of X and Y,

s_i , c_i : i-th MSD digits of intermediate sum and carry,

z_i : i-th MSD digit of the final result,

i : index of digit position.

Table (2): First step computational rules of MSD adder.

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
G_1	$(\bar{1},\bar{1})$	Don't care	0	$\bar{1}$
G_2	$(0,\bar{1}),(\bar{1},0)$	Both nonnegative Other wise	$\bar{1}$ 1	0 $\bar{1}$
G_3	$(0,0),(1,\bar{1}),(\bar{1},1)$	Don't care	0	0
G_4	$(0,1),(1,0)$	Both nonnegative Other wise	$\bar{1}$ 1	1 0
G_5	$(1,1)$	Don't care	0	1

Table (3): Second step computational rules of MSD adder.

[illegible]	[illegible]	[illegible]
H_1	$(1,0),(0,1)$	1
H_2	$(1,1),(1,\bar{1}),(0,0),(\bar{1},1),(\bar{1},\bar{1})$	0
H_3	$(\bar{1},0),(0,\bar{1})$	$\bar{1}$

Haung et. al. [9] classified the nine (X_i, Y_i) combinations into five groups (G_1 –G_5) as shown in the first and second columns of Table (2). In the first step, CPs and NLOPs are examined to know the groups G_i they belong to, and depending on these groups s_i and c_i will be generated. *Awwal* and *Iftekharuddin* [12] simplified the computational rules of the first step as presented in Table (2). In the second step, s_i will be added to c_{i-1} (carry shifted one position to the left) according to the three computational rules (H_1 – H_3) of the second step as shown in Table (3) in order to find z_i. The final result of adding two n-digit MSD numbers will be (n+1)-digit MSD number during only two steps.

4- Digit-Decomposition-Plane Representation (DDP)

DDP representation had been proposed by *Huang* and *et.al* [9]. It is an extension for bit-plane representation method [13]. DDP representation can be applied to code large 2D data arrays of SD numbers. It can be explained by applying it to 2D MSD numbers array as follow:

i- If (i,j)-th digit in the 2D MSD numbers array equal to 1, so the corresponding (i,j)-th pixel of the DDP-1 plane will be transparent (white) and the corresponding pixels (i,j)-th of DDP-0 and DDP- $\bar{1}$ planes will be opaque (dark).

ii- If the digit equal to 0, then the corresponding pixel of DDP-0 plane will be transparent and the corresponding pixels of DDP-1 and DDP-$\bar{1}$ planes will be opaque. Otherwise, if the digit equals to $\bar{1}$, the corresponding pixel of DDP-$\bar{1}$ plane will be transparent and the corresponding pixels of DDP-1 and DDP-0 planes will be opaque.

Note that the DDP planes will be equal to the number of weights that MSD number system consisted. Figure (2) shows an example for DDP coding method for a 2D MSD numbers array (3 DDP planes).

In order to implement the DDP scheme, the DDP planes can be represented as Spatial Light Modulators (SLMs) with pixel resolution fulfill the SD array dimensions.

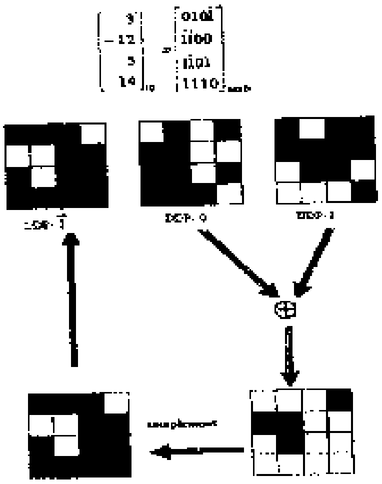


Fig. (2): Complement feature of DDP representation.

There are two apparent features of DDP coding method. The first one is that any DDP plane is the complement of the superimposing plane of the other planes. The superimposing plane is obtained by accumulating the bright and dark pixels in one plane. Figure (2) explains this feature. The second one is that if all DDP planes are superimposed, the result will be a totally transparent plane. In this work, these two features have been used successfully to simplify the optical implementation and also to check the errors that may appear in the intermediate and final results during simulation operation.

4- Design Parallel Optical Two-Step MSD Array Adder

The addition of the two 2D MSD arrays A and B is performed. The 2D SD data array will be presented as M×N×n. When DDP representation is applied to MSD array A, then three DDP planes will be generated and named as A1, A0, and $A\bar{1}$. Similarly, MSD array B can be decomposed to three DDP planes, which are called B1, B0, and $B\bar{1}$.

The main objective is to derive logic combination formulas from Tables (2) and (3). Then the DDP planes of the intermediate and final results for both the first and second steps of addition operation are calculated.

- **First-step:**

In Table (2), the sum s_i that has value 1 can be obtained according to the following conditional statement:

$$\begin{aligned}&IF\ [(x_i, y_i) = (\bar{1},0)\ OR\ (0,\bar{1})\ OR\ (1,0)\ OR\ (0,1)]\\&\quad AND\ [(x_{i-1}, y_{i-1}) = (1,1)\ OR(1,0)\ OR\ (0,1)\\&\quad OR\ (1,1)\ OR\ (1,1)] \qquad THEN\ [s_i = 1]\end{aligned} \tag{3}$$

Because $x_i = 1, 0, \text{and}\ \bar{1}$ ($y_i = 1, 0, \text{and}\ \bar{1}$) are represented as transparent pixels in A1, A0, and $A\bar{1}$ and (B1, B0, and $B\bar{1}$) planes, respectively, in the i-th pixel position of the j-th row. Therefore, equation (3) can be rewritten in form of sum of product terms as:

$$\begin{aligned}s1_{ji} = [&a\bar{1}_{ji} * b0_{ji} + a0_{ji} * b\bar{1}_{ji} +\\&a1_{ji} * b0_{ji} + a0_{ji} * b1_{ji}] *\\&[a1_{ji-1} * b1_{ji-1} + a\bar{1}_{ji-1} * b0_{ji-1} +\\&a0_{ji-1} * b\bar{1}_{ji-1} + a1_{ji-1} * b\bar{1}_{ji-1} +\\&a\bar{1}_{ji-1} * b1_{ji-1}]\end{aligned} \tag{4}$$

where,

$a1_{ji}, a0_{ji}, a\bar{1}_{ji}$ and $b1_{ji}, b0_{ji}, b\bar{1}_{ji}$:

(j, i)-th pixel of DDP-1,0,$\bar{1}$ of array A and B.

$s1_{ji}$: (j, i)-th pixel of DDP-1 of the intermediate sum array.

* and + : logic AND and OR, respectively.

j, i : indices of rows and columns of DDP plane.

In general, equation (4) is applied to all numbers include in the DDP planes of A and B arrays, and generate all the intermediate sums s_is that have values 1 in parallel. Generalizing equation (4) to the DDP planes of A and B arrays the following equations have been derived:

$$\begin{aligned}S1 = [&A\bar{1} * B0 + A0 * B\bar{1} + A1 * B0 +\\&A0 * B1] * [A'\bar{1} * B'\bar{1} + A'\bar{1} * B'0 +\\&A'0 * B'\bar{1} + A'1 * B'\bar{1} + A'\bar{1} * B'1]\end{aligned} \tag{5}$$

The symbol (′) in the right top of the DDP planes of A and B arrays denotes the shifting one position to the left for each number is included in arrays A and B. Equation (5) can be simplified to:

$$\begin{aligned}S1 = [&A0 * (B1 + B\bar{1}) + (A1 + A\bar{1}) * B0]\\&* [A'\bar{1} * (B'1 + B'0 + B'\bar{1})\\&+ (A'1 + A'0) * B'\bar{1}]\end{aligned} \tag{6}$$

Using the complement and superimposing features of DDP representation then:

$$S1 = [A0 * \overline{B0} + \overline{A0} * B0] * [A'\bar{1} + \overline{A'\bar{1}} * B'\bar{1}] \qquad (7)$$

The symbol $(\overline{\quad})$ on the top of the DDP planes means binary complement of the pixel is included in them.

Similarly, referring to Table (3), $S\bar{1}$ and $S0$ which are DDP-$\bar{1}$ and DDP-0 planes of the intermediate sum array S, and C1, $C\bar{1}$, and C0 which are DDP-1, DDP-$\bar{1}$, and DDP-0 planes of the intermediate carry array C, can be determined according to the newly derived equations as shown below:

$$S\bar{1} = [A0 * \overline{B0} + \overline{A0} * B0] * [\overline{A'\bar{1}} * \overline{B'\bar{1}}] \qquad (8)$$

$$S0 = \overline{A0} * \overline{B0} + A0 * B0 \qquad (9)$$

$$C1 = A1 * B1 + [A1 * B0 + A0 * B1] * [\overline{A'\bar{1}} * \overline{B'\bar{1}}] \qquad (10)$$

$$C\bar{1} = A\bar{1} * B\bar{1} + [A\bar{1} * B0 + A0 * B\bar{1}] * [A'\bar{1} + \overline{A'\bar{1}} * B'\bar{1}] \qquad (11)$$

$$C0 = A0 * B0 + A1 * B\bar{1} + A\bar{1} * B1 + [A1 * B0 + A0 * B1] * [A'\bar{1} + \overline{A'\bar{1}} * B'\bar{1}] + [A\bar{1} * B0 + A0 * B\bar{1}] * \overline{A'\bar{1}} * \overline{B'1} \qquad (12)$$

- **Second-step:**

After DDP planes of the intermediate sum and carry arrays are obtained, the DDP planes of the final results array Z1, Z0, and $Z\bar{1}$ can be calculated depending on the computational rules of Table (3). In this table, $z_{\bar{1}} = 1$ is based on the following equation:

$$\begin{aligned} &IF\ [(s_i = 1)\ AND\ (c_{i-1} = 0)] \\ &\quad OR\ [(s_i = 0)\ AND\ (c_{i-1} = 1)] \\ &\qquad THEN\ z_i = 1 \end{aligned} \qquad (13)$$

or

$$z1_{p} = s1_{p} * c0_{p-1} + s0_{p} * c1_{p-1} \qquad (14)$$

and in plane form,

$$Z1 = S1 * C'0 + S0 * C'1 \qquad (15)$$

Similarly, $Z\bar{1}$ and Z0 can be generated according to the following rules:

$$Z\bar{1} = S0 * C'\bar{1} + S\bar{1} * C'0 \qquad (16)$$

$$Z0 = (S1 + S\bar{1}) * (C'1 + C'\bar{1}) + S0 * C'0 \qquad (17)$$

Finally, three DDP planes Z1, Z0, and $Z\bar{1}$ are obtained, each of them has dimension M×N×(n+1), which are given the 2D MSD array of the final results Z computed in parallel.

Note that, when the number is shifted one position to the left, one zero will be padded to the LSB digit. In DDP representation, a transparent pixel will be placed in LSB pixel position of DDP-0 plane of the number, and a dark pixel will be placed in the LSB pixel position of the DDP-1 and DDP-$\bar{1}$ planes of this number.

5- Optical Implementation

The optical hardware schemes of this parallels MSD array adder can be implemented practically using simple optical tools. The logical AND and OR gates are the main operations in the logical formulas. The principles handled for AND and OR operations will be considered to realize our suggestion for MSD adder [3]. Logical AND can be achieved optically by cascading two DDP planes, each of one array, with same dimensions and pixel resolutions. The light beams will be applied on each pixel of the first plane. These beams are passed through the transparent pixel (1) and blocked by opaque pixel (0). Thereby, four combinations of transparent (1) and opaque (0) pixels of the two DDP planes will be presented: (1× 1), (1× 0), (0× 1), and (0× 0).Only in case (1× 1) the light beam will pass through the two cascaded planes. So, this optical AND can process M×N×n binary AND operations in parallel. Figure (3) shows the optical implementation of the logical AND. A pixel-wise addition in parallel for the corresponding optical logical OR can be realized using beam combiner (BC). BC performs light beams of the two packages passed through two DDP planes and fallen onto the two BC input surfaces. Three pixel-wise addition cases will be gain: (1+0), (0+1), and (0+0). The case (1+1) will never appear because the pixel-wise addition is done between two DDP planes of the same array. So, M×N×n binary OR operations will be done in parallel. Figure (4) presents the optical implementation of the logical OR.

The proposed parallel optical two-step MSD adder operation can be explained as following:

1- The input M× N× n DDP-planes of the first step are illuminated by Laser sources to enter the optical system for processing.

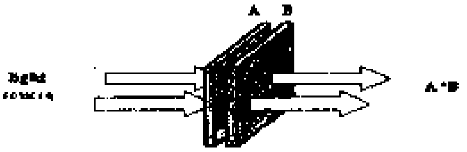


Fig. (3): AND gate optical implementation

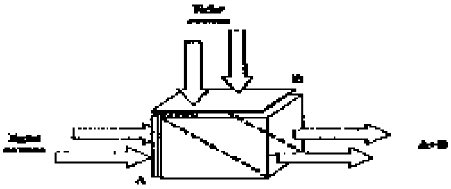


Fig. (4): OR gate optical implementation

2- Light detector arrays (LDAs) in the end of optical scheme of the first step detect the DDP-planes of the intermediate results. that were produced by the first step.

3- These optical signals will be converted to electrical signals and passed to the optical implementation of the second step.

4- The input DDP-planes of the second step is addressed in parallel by these electrical signals to form the sufficient expanded and shifted copies of the intermediate results. Expanding and shifting means that, one pixel is necessary to be padded to the LSB and MSD pixel position of the detected DDP-planes of the intermediate carry and intermediate sum, respectively.

5- Finally, the LDAs detect the optical signals that represent the DDP-planes in the final result array, which are M×N×(n+1) MSD arrays. The above points present the complete optical implementation of the suggested parallel optical two-step MSD adder with the consideration of the following explanations:

BS BC : denotes a beam combiner and beam splitter.

\ : denotes a mirror.

▯ : denotes SLM (DDP plane), LDA, or CMP.

→ : the direction of the light signal emitted by an optical source.

The optical implementation of the two steps 2D MSD array is shown in Figure (5).

6- Simulation Results

Computer programming is used to simulate the optical implementations of the proposed SD adders. C++ program is built and executed as two-step adders for DDP coded 2D MSD arrays. The two 10×2 ×4 MSD data arrays A and B are added by the proposed two-step MSD adder. In Figure (6), the MSD arrays A and B are shown as DDP codes. The DDP planes of the intermediate sum array S and the shifted copies of the DDP planes of the intermediate carry array C are obtained by the optical implementation of the first step and are presented too. The DDP planes of the final result array Z are calculated by the optical implementation of the second step. We can see that, the input MSD arrays A and B are 10×2 ×4, while the output MSD array Z is 10×2× 5.

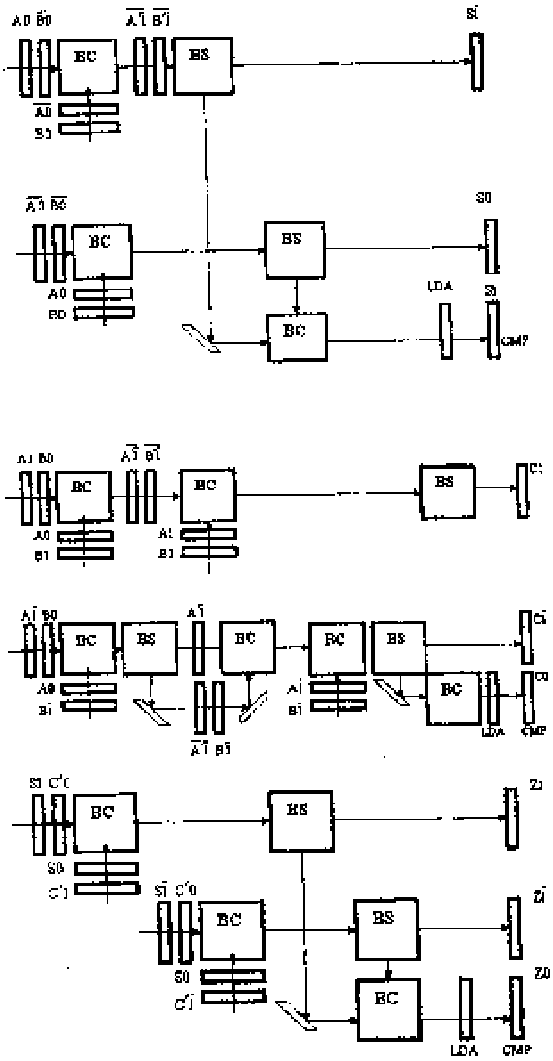


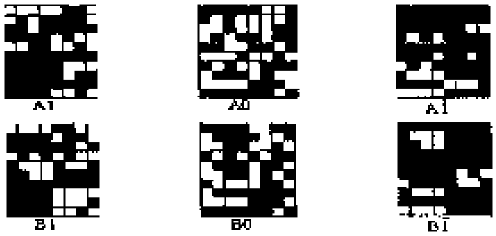
Fig. (5): Optical implementation of MSD adder.

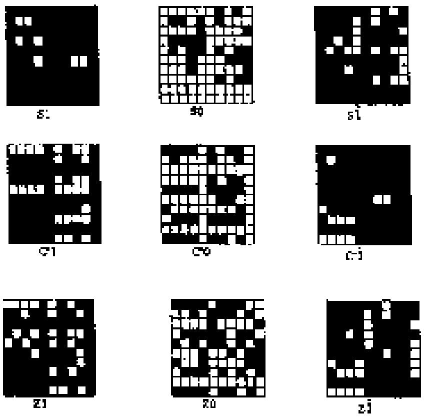
7- Conclusion

The optical implementation of the proposed two-step MSD array adder with DDP representation has been tested here. The two 10×2×4 MSD arrays A and B will be added using this adder. The delay time is caused by the holding time of the input images and the propagation times of light from the source to detector planes.

$$A=\begin{bmatrix} 15 & 9 \\ 10 & 1 \\ 2 & -3 \\ -5 & 11 \\ 12 & 13 \\ -1 & -2 \\ -10 & 7 \\ -7 & 4 \\ 0 & -14 \\ -15 & 13 \end{bmatrix}_{10} = \begin{bmatrix} 1111 & 1001 \\ 1010 & 0001 \\ 0010 & 00\bar{1}\bar{1} \\ 0\bar{1}0\bar{1} & 1011 \\ 1100 & 1101 \\ 000\bar{1} & 00\bar{1}0 \\ \bar{1}0\bar{1}0 & 0111 \\ 0\bar{1}\bar{1}\bar{1} & 0100 \\ 0000 & \bar{1}\bar{1}\bar{1}0 \\ \bar{1}\bar{1}\bar{1}\bar{1} & 1101 \end{bmatrix}_{MSD}$$

$$B=\begin{bmatrix} 15 & 3 \\ -7 & 9 \\ -3 & 2 \\ 10 & 11 \\ 7 & 6 \\ 3 & -5 \\ -8 & -6 \\ -7 & 15 \\ 0 & 14 \\ -15 & 13 \end{bmatrix}_{10} = \begin{bmatrix} 1111 & 0011 \\ 0\bar{1}\bar{1}\bar{1} & 1001 \\ 00\bar{1}\bar{1} & 0010 \\ 1010 & 1011 \\ 0111 & 0110 \\ 0011 & 0\bar{1}0\bar{1} \\ \bar{1}000 & 0\bar{1}\bar{1}0 \\ 0\bar{1}\bar{1}\bar{1} & 1111 \\ 0000 & 1110 \\ \bar{1}\bar{1}\bar{1}\bar{1} & 1101 \end{bmatrix}_{MSD}$$





$$Z=\begin{bmatrix} 11110 & 1\bar{1}100 \\ 0010\bar{1} & 1\bar{1}010 \\ 0000\bar{1} & 0000\bar{1} \\ 0\bar{1}\bar{1}\bar{1} & 10110 \\ 1010\bar{1} & 1010\bar{1} \\ 00010 & 0\bar{1}01\bar{1} \\ \bar{1}00\bar{1}0 & 0001\bar{1} \\ 0\bar{1}\bar{1}\bar{1}0 & 1010\bar{1} \\ 00000 & 00000 \\ \bar{1}\bar{1}\bar{1}\bar{1}0 & 11010 \end{bmatrix}_{MSD} = \begin{bmatrix} 30 & 12 \\ 3 & 10 \\ -1 & -1 \\ 5 & 22 \\ 19 & 19 \\ 2 & -7 \\ -18 & 1 \\ -14 & 19 \\ 0 & 0 \\ -30 & 26 \end{bmatrix}_{10}$$

Fig. (6): Simulation results for MSD adder.

The propagation time can be ignored because of the whole system hardware is small if the optical tools dimensions are take into account. While the SLM response time represents the major operation of the delay addition in response time. The method of algorithm and architecture are independent of the size of the operand arrays. An optical implementation scheme is based on classic optical elements.

References

[1] S. Zhang and M. A. Karim," Optical Arithmetic Processing Using Improved Redundant Binary Algorithm," *Opt.Eng.*, Vol.38, No.3, pp.415-421, March 1999.

[2] C. D. Capps and T. L. Houk," Optical Arithmetic Logic Unit Based on Residue Arithmetic and Symbolic Substitution," *Appl.Opt.*, Vol.27, pp.1682-1687, 1988.

[3] A. K. Cherri and M. S. Alam" Optoelectronic Symbolic Substitution Based Conical Modified SD Arithmetic," *Opt. Laser Technol.*, Vol.29, pp.151-157, 1997.

[4] S. Zhang and M. A. Karim," Programmable Modified SD Addition Module Based on Binary Logic Gates," *Opt.Eng.*, Vol.38, No.3, pp.456-461, March 1999.

[5] A. K. Cherri," Signed-Digit Arithmetic for Optical Computing: Digit Groping and Pixel Assignment for Spatial Coding," *Opt.Eng.*, Vol.38, No.3, pp.422-431, March 1999.

[6]A. W. Al-Saffar," Parallel Optical Computing Using Redundant Number Representation," *Ph.D. thesis,* College of Engineering, Basrah University, May 2001.

[7] M. Alam, A.Cherri, and A. Chatterjea," Symmetrically Recoded Quaternary Signed-Digit Arithmetic Using a Shared Content-Addressable Memory," *Opt.Eng.*, Vol.35, No.4, pp.1141-1149, Apr. 1996.

[8] F. Ahmmed, A. S. Awwal, and G. J. Power," Synthetic Correlation Modified Signed-Digit Processing," *Opt.Eng.*, Vol.38, No.3, pp.449-455, March 1999.

[9] H. Huang," Optical Scalable Parallel Modified Signed-Digit Algorithms for Large Scale Array Addition and Multiplication Using Digit Decomposition Plane Representation," *Opt.Eng.*, Vol.38, No.3, pp.432-440, March 1999.

[10] A. K. Cherri and M. S. Alam," Algorithms for Optoelectronic Implementation of Modified Signed-Digit Division, Square Root, Logarithmic, and Exponential Functions," *Appl.Opt.*, Vol.40, pp.1236-1243, March 2001.

[11] T. Stouraitis and C. Chen," Hybrid Signed-Digit Logarithmic Number System Processor," *Proc.IEEE.*, Vol.140, No.11, pp.205-210, 1993.

[12] A. S. Awwal and K. M. Iftekharuddin," Graphical Approach for Multiple Valued Logic Minimization," *Opt.Eng.*, Vol.38, No.3, pp.462-467, March 1999.

[13] H. Huang and T. Yatagai," Optical Module for Modified SD Computing Based on Bit-Plane Encoding and Pattern Operations," *Opt.Rev.*, Vol.2, No.4, pp.255-260, 1995.