

Develop Parallel Arithmetic Operations for Binary Modified Signed-Digit System Using Two-Step Algorithm

Wijdan Yassin A. Al Karim

College of computer sciences and mathematics
Mousel University

Received on: 25/10/2009

Accepted on: 16/5/2010

الملخص

نظام الحساب الضوئي يتوقع أن يكون من أنظمة معالجة المعلومات القوية لأنه يأخذ الفائدة الكاملة من الضوء مثل، التوازي، السرعة العالية، واستيعاب المعلومات الكبير. لذلك الكثير من أنظمة العد الضوئية طورت من قبل العديد من الباحثين لغرض استغلال طبيعة التوازي الفطرية للضوء لتطوير عمليات حسابية متوازية. في هذا البحث استخدمنا نظام عد الأرقام المؤشرة الثنائي المحدث وطريقة تشفير ملائمة لتوليد جدول حقيقة ثنائي مبسط لعمليات الجمع و الطرح للحصول من خلاله على خوارزمية متوازية متكونة من خطوتين. في هذه الخوارزمية سلسلة التحميل المتولدة خلال عملية الجمع وسلسلة الاستعارة المتولدة خلال عملية الطرح سوف توقف بعد تطبيق هذه الخطوتين. عملية الجمع و الطرح يتم إكمالها بدون الاعتماد على طول الأرقام المستخدمة في أي معامل. إضافة إلى ذلك يشمل البحث تقليص المعلومات المخزونة الخاصة بجدول الحقيقة لعمليات الجمع والطرح المستخدمة في المعالجة المنطقية.

ABSTRACT

An optical computing system is expected to be a powerful information processing system that takes full advantage of optics, such as parallelism, high speed, and large information capacity. Therefore many suitable optical number systems have been developed by many research to exploit the inherent parallelism of optics to developed parallel arithmetic operations .

In this paper we used binary modified signed-digit (BMSD) number system and available recoding method to generate a simplified binary MSD addition/subtraction truth table to obtained a parallel two-step algorithm in which the carry chain was generated during the addition operation and the borrowing chain was generated during the subtraction operation that will be stopped after two steps, the addition and subtraction operations will be completed independent of the number of digits in each operand. Also this paper involve limitation for the minterms that used in the logical processing for the addition/subtraction truth table.

1. Introduction :

Digital optical computing has stimulated a great deal of interest during the last two decades. The objective is to develop efficient computing algorithms and architectures that can exploit the inherent parallelism and 3-D massive no interfering interconnections of optics [4].

High-speed and parallel processing has been an attractive subject in computing science for along time. In Boolean logic-based electronic digital computers, binary addition is a key arithmetic operation. The addition algorithms however must have a carry propagation that will severely limit the calculating speed [5].

To avoid or limit the carry propagation and to reduce the memory size of content-addressable memory (CAM), many of number systems have been developed such as , residue-number, redundant-binary (RB) number representation ,and modified signed-digit (MSD) number system [7].

2. Explicative idea for the number systems employed:

All number systems have their own advantages and limitations. Using a residue number system(RNS) , arithmetic operations such as addition, subtraction, and multiplication can be done without carries but the number and the size of the truth table increase greatly as the numerical range involved becomes large [1,4,7] . The redundant signed-digit number system is another alternative number representation which generates limited-carry propagation arithmetic. In this number system parallel arithmetic operation are possible in a constant operation time and with few carry propagation steps but the information storage density of the number system is limited which often result in complex circuits [1].

The most widely investigated representation is the modified signed-digit (MSD) number system , in which the carry propagation can be confined in two adjacent digit position and thus addition or subtraction of arbitrary-length operands can be completed in definite steps . As a result, three step, two step, and single-step algorithms have been suggested [6].

3. Olden arithmetic approaches:

Mirsalehi and Gaylord had implemented MSD addition and subtraction methods using content-addressable memory (CAM) ,even though this method improves the speed of operation but it requires an increasingly more complex CAM hologram , for every single output bit the minterms need to be stored in fifty-six CAM holograms [2].

Li and Eichmann have introduced a two-step condition symbolic MSD operation where only up to twelve holograms are needed for each output bits [9].

The scheme was extended further by Cherri and Karim which introduce an additional input bit with each pair of input bits only up to nine reference hologram are needed for each output bit and only four bits are needed in this template where as the template in Li and Eichmann method requires to five bits [2].

4. Research objective:

In this paper we have to extend the scheme provided by Cherri and Karim by utilizing recoding method for the input number to convert it to an efficient MSD representation in order to limit the carry and barrow propagation in the addition and subtraction operations. Based on the recoding method we attempted to develop a truth table which can apply parallel addition / subtraction operations in two-steps algorithm and fewer number of minterms needed to be stored in CAM hologram.

5.The modified signed-digit (MSD) number system :

The MSD number representation allows parallel arithmetic operation with reduced number of carry propagation steps. It's referred to as a redundant number because more than one representation is possible for any number in this system.

For radix-2 MSD number system, a given decimal integer number D either positive or negative can be represented by an n-digit as in eq(1) :

$$D = \sum_{i=1}^{n-1} X_i 2^i \quad \dots(1)$$

Where the X_i is a member of the set $\{\bar{1}, 0, 1\}$, here $\bar{1}$ denotes (-1). So the MSD negative number is the complement of the MSD positive number [1,5].

For example, number $(9)_{10}$ has many binary MSD representation so, (01001) , $(1\bar{1}01\bar{1})$, $(011\bar{1}\bar{1})$, and $(1\bar{1}001)$. These representations are given by applied encoding method based on truth table shown in table (1).

Table (1): truth table for binary MSD representation

Decimal representation	Binary MSD representation
1	01 1 $\bar{1}$
0	00
-1	0 $\bar{1}$ $\bar{1}$ 1

6. Basic concept of the recoding MSD method :

The redundant property of MSD number system can be used to simplify the MSD addition rules. Let $x = x_{n-1} x_{n-2} \dots x_0$, be an n-digit MSD number, then when x is recoded according to the recoding method, it produced an (n+1) digit MSD number $z = z_n z_{n-1} z_{n-2} \dots z_0$, note that the MSD number x and the recoding number z are numerically equal. Furthermore the MSD digits of the recoded number z satisfy the condition $(z_i \times z_{i-1} \neq 1)$ [1, 3].

As shown in table (1), number $(1)_{10}$ has two MSD representation (01) and $(1\bar{1})$ respectively, we can eliminate the representation (01) during the recoding operation and use $(1\bar{1})$ representation only. These operations will limit the appearance of consecutive (11) and $(1\bar{1})$ series from the recoded number (addend / augend) operand which can make the carry propagation during the addition operation [3].

Thus when number $(9)_{10}$ is recoded according to this method it will take the MSD representation $(1\bar{1}01\bar{1})$ only. This representation was given by divided number $(9)_{10}$ into radix-2 as shown in fig(1), then when the remainder is equal to 1 it will map into $(1\bar{1})$ MSD representation, where bit $\bar{1}$ is called a sum and bit 1 is called a carry. Note that bit 1 will be shifted one step below. Also when the remainder is equal to 0 it will map into (00). These sum and carry bits are summed to give the final recoding MSD representation number.

Decimal number	$\div 2$	Decimal remain	Binary carry	MSD sum	Final result
9	4	1		$\bar{1}$	$\bar{1}$
4	2	0	1	0	1
2	1	0	0	0	0
1	0	1	0	$\bar{1}$	$\bar{1}$
			1		1

Fig.(1): recoding method

7. Binary MSD addition operation :

In the MSD addition operation , the number are first represented in MSD representation and then to-be-added pair of numbers will be mapped into another intermediate pair of numbers called primary transfer and weight digits in a way that the latter numbers when added , do not result in any carry . These transfer and weight are used to generate secondary set of transfer and weight digits , which in turn generate bits that are summed to give the final result , therefore the MSD addition operation requires four(three)steps to obtained the final carry-free addition [8].

Li and Eichmann combined the first two (three) steps by incorporating additional information from the less significant pair of bits, the resulting two-step algorithm [9].

This two-steps algorithm is much simpler by Cherri and Karim as shown in table (2-a). The examination of table (2-a) reveals that when two MSD bits numbers $x_i y_i$ are summed the less significant bits $x_{i-1} y_{i-1}$ must be examined to generate the transfer (T) and weight (W) bits respectively. Notice that when $x_i y_i = (01)$ or (10) and the less significant bits are equal to the set, $set1 = \{11, 01, 10, 00\}$, then we take $(\bar{1}\bar{1})$ representation . If the less significant bits are equal to the set, $set2 = \{\bar{1}\bar{1}, 11, 01, \bar{1}0, \bar{1}\bar{1}\}$ we take (01) representation. Likewise, in the case of $x_i y_i = (\bar{1}0)$ or $(0\bar{1})$ and the less significant bits $x_{i-1} y_{i-1}$ are equal to set1, will take $(0\bar{1})$ representation, and if $x_{i-1} y_{i-1}$ are equal to set2, will take $(\bar{1}\bar{1})$ representation respectively [2].

From table (2-a) notice that, it have two halves. Each half consisting of three identical columns. Therefore, it is suitable to introduce a flag bit f_i , which has a value either 1 or 0 to denote less significant pair of bits. This reduces table (2-a) into a new table shown in table (2-b), this table consist of three input bits only.

Table (2-a): Rules of MSD addition operation for ref.2

$x_i y_i \setminus x_{i-1} y_{i-1}$				$T^+ W^+$		
	11	10/01	00	$1\bar{1}/\bar{1}1$	$\bar{1}0/0\bar{1}$	$\bar{1}\bar{1}$
11	1 0	1 0	1 0	1 0	1 0	1 0
10/01	1 $\bar{1}$	1 $\bar{1}$	1 $\bar{1}$	1 $\bar{1}$	1 $\bar{1}$	1 $\bar{1}$
00	0 0	0 0	0 0	0 0	0 0	0 0
$1\bar{1}/\bar{1}1$	0 0	0 0	0 0	0 0	0 0	0 0
$\bar{1}0/0\bar{1}$	0 $\bar{1}$	0 $\bar{1}$	0 $\bar{1}$	$\bar{1}1$	$\bar{1}1$	$\bar{1}1$
$\bar{1}\bar{1}$	$\bar{1}0$	$\bar{1}0$	$\bar{1}0$	$\bar{1}0$	$\bar{1}0$	$\bar{1}0$

Table (2-b): reduced rules for table (2-a)

$x_i y_i \setminus f_i$		$T^+ W^+$	
	1	0	
11	1 0	1 0	
10/01	1 $\bar{1}$	1 $\bar{1}$	
00	0 0	0 0	
$1\bar{1}/\bar{1}1$	0 0	0 0	
$\bar{1}0/0\bar{1}$	0 $\bar{1}$	0 $\bar{1}$	
$\bar{1}\bar{1}$	$\bar{1}0$	$\bar{1}0$	

8. Binary MSD subtraction operation:

In a similar way the subtraction operation truth table can be introduced as shown in table (3-a). When two MSD number $x_i y_i$ are subtracted to generate the primary T^- and W^- step, first we must examine the less significant bits $x_{i-1} y_{i-1}$. Table (3-a) reveals that when $x_i y_i = 0\bar{1}$ or 10 and the less significant bits $x_{i-1} y_{i-1}$ are equal to the set, $set3 = \{00, 10, 0\bar{1}, 1\bar{1}\}$ then $(1\bar{1})$ representation must be taken. If the less significant bits $x_{i-1} y_{i-1}$ are equal to the set, $set4 = \{10, 01, 1\bar{1}, \bar{1}1\}$, then (01) representation must be taken.

Likewise, in the case of $x_i y_i = (\bar{1}0)$ or (01) and the less significant bits $x_{i-1} y_{i-1}$ are equal to set3, $(0\bar{1})$ representation must be taken, and if the less significant bits $x_{i-1} y_{i-1}$ are equal to set4, $(1\bar{1})$ representation only will be taken. Also we can note that table (3-a) consists of two halves, therefore it is possible to reduce it using a flag bit F to produce new table as shown in table(3-b).

Table (3-a): Rules of MSD subtraction operation for ref.2

$x_i y_i \setminus x_{i-1} y_{i-1}$		$T^- W^-$				
	00	10/0 $\bar{1}$	1 $\bar{1}$	$\bar{1}0/01$	$\bar{1}1$	$\bar{1}\bar{1}/11$
$\bar{1}\bar{1}/11$	0 0	0 0	0 0	0 0	0 0	0 0
00	0 0	0 0	0 0	0 0	0 0	0 0
10/0 $\bar{1}$	1 $\bar{1}$	1 $\bar{1}$	1 $\bar{1}$	0 1	0 1	0 1
01/ $\bar{1}0$	0 $\bar{1}$	0 $\bar{1}$	0 $\bar{1}$	$\bar{1}1$	$\bar{1}1$	$\bar{1}1$
1 $\bar{1}$	1 0	1 0	1 0	1 0	1 0	1 0
$\bar{1}1$	$\bar{1}0$	$\bar{1}0$	$\bar{1}0$	$\bar{1}0$	$\bar{1}0$	$\bar{1}0$

Table (3-b): reduced rules for table (3-a)

$x_i y_i \setminus f_i$		$T^- W^-$	
		1	0
$\bar{1}\bar{1}/11$		0 0	0 0
00		0 0	0 0
10/0 $\bar{1}$		1 $\bar{1}$	1 $\bar{1}$
01/ $\bar{1}0$		0 $\bar{1}$	0 $\bar{1}$
1 $\bar{1}$		1 0	1 0
$\bar{1}1$		$\bar{1}0$	$\bar{1}0$

9. The proposed method:

9-1. Binary MSD addition operation:

As we explained previously, the recoding method was able to generate a recoded MSD numbers x and y such that the combination of $x_i x_{i-1}$ and $y_i y_{i-1} = 11$ or $\bar{1}\bar{1}$ will never occur. The benefit of this feature will be found during the addition operation. When two recoded MSD numbers are summed to generate the primary transfer (T^+) and weight (W^+) step, a carry was obtained when the pair of bits $x_i y_i = 01$ or 10 and the less significant bits $x_{i-1} y_{i-1}$ are equal to (11) , then to avoid this carry propagation $(1\bar{1})$ representation must be used. Otherwise will take (01) representation for all other cases of the less significant bits as shown in table (4-a). This method denotes the incorporation

of all the sets of less significant bits (set1 and set2) that appear in table (2-a) under one representation except the probability of (11) will take another representation.

Also the carry can be obtained when the pair of bits $x_i y_i = 0\bar{1}$ or $\bar{1}0$ and the less significant bits $x_{i-1} y_{i-1}$ are equal to ($\bar{1}\bar{1}$), then ($\bar{1}\bar{1}$) representation must be taken. Otherwise ($0\bar{1}$) representation will be taken for all other cases of the less significant bits. This means incorporate all the sets of the less significant bits (set1 and set2) that appear in table (2-a) under one representation except the probability of ($\bar{1}\bar{1}$) that will take another representation.

Then, when applying the addition operation rules on the truth table (4-a) using summed two recoded MSD numbers to give the T^+ and W^+ digits, and then summed the T^+ and W^+ bits to generate the final result, we obtain fast and correct results as shown in the following example:.

$$1\bar{1}010\bar{1}\bar{1}\bar{1}0 = (154)_{10}$$

$$1\bar{1}10\bar{1}01\bar{1} = (89)_{10}$$

$$010\bar{1}00010\bar{1} = W^+$$

$$00010\bar{1}0000 = T^+$$

$$01000\bar{1}010\bar{1} = (243)_{10}$$

Table (4-a): modification of table (2-a) using the proposed addition rules

$x_i y_i \setminus x_{i-1} y_{i-1}$	$T^+ \quad W^+$		
	11	$\bar{1}\bar{1}$	otherwise
11	1 0	1 0	1 0
10/01	0 1	0 1	1 $\bar{1}$
00	0 0	0 0	0 0
$\bar{1}\bar{1}/\bar{1}\bar{1}$	0 0	0 0	0 0
$\bar{1}0/0\bar{1}$	0 $\bar{1}$	$\bar{1}$ 1	0 $\bar{1}$
$\bar{1}\bar{1}$	$\bar{1}$ 0	$\bar{1}$ 0	$\bar{1}$ 0

If we remove the examination of the less significant bits $x_{i-1} y_{i-1}$ from table (4-a) and use (01) representation in the case of $x_i y_i$ are equal to (10/01) numbers, also we use ($0\bar{1}$) representation in the case of $x_i y_i$ are equal to ($\bar{1}0/0\bar{1}$) numbers. Then apply the addition rules on this table, we can note that, this change essentially has no effect on the final result. The reason is belong to the recoding method which not permit the series of bits ($\bar{1}\bar{1}$) and (11) to appear in the recoded numbers. Based on this observation it is appropriate to reduce table (4-a) to introduce a new table shown in table (4-b). Note that table (4-b) consist of two input bits only for each output T^+ and W^+ bits. Instead of three input bits used in table (2-b) for the earlier method by cherri and karim. Thus if we applied the proposed method on the previous example we note the same result as shown below:

$$\begin{aligned} 1\bar{1}010\bar{1}\bar{1}\bar{1}0 &= (154)_{10} \\ 1\bar{1}10\bar{1}01\bar{1} &= (89)_{10} \end{aligned}$$

$$\begin{aligned} 010\bar{1}00010\bar{1} &= W^+ \\ 00010\bar{1}0000 &= T^+ \end{aligned}$$

$$01000\bar{1}010\bar{1} = (243)_{10}$$

Table (4-b): reduced rule for table (4-a)

$x_i y_i$	$T^+ \quad W^+$
11	1 0
10/01	0 1
00	0 0
$\bar{1}\bar{1}/\bar{1}\bar{1}$	0 0
$\bar{1}0/0\bar{1}$	0 $\bar{1}$
$\bar{1}\bar{1}$	$\bar{1}$ 0

9-2. Binary MSD subtraction operation:

In the similar method, when two recoded MSD numbers $x_i y_i$ are subtracted to generate the primary transfer T^- and weight W^- digit step, the borrow is generated when $x_i y_i = 0\bar{1}$ or 10 and the less significant bits $x_{i+1} y_{i+1}$ are equal to $(\bar{1}\bar{1})$, then we must take $\bar{1}\bar{1}$ representation to avoid this borrow. Otherwise, less significant bits will take 01 representation as shown in table (5-a). This method denote to incorporate all the sets of the less significant bits (set3 and set4) which appear in table (3-a) under one representation except the probability of $(\bar{1}\bar{1})$ will take another representation.

Also the borrow is generated when $x_i y_i = \bar{1}0$ or 01 and the less significant bits are equal to $\bar{1}\bar{1}$, then will take $\bar{1}\bar{1}$ representation. Otherwise, the less significant bits will take $(0\bar{1})$ representation. This method incorporates all the sets of the less significant bits (set3 and set4) that appear in table (3-a) under one representation except the probability of $(\bar{1}\bar{1})$ will take another representation.

This means that we can give the complement of y_i and y_{i-1} bits from table (4-a) to find the subtraction truth table (5-a). Then when we apply subtraction rules into the truth table (5-a) using subtract two recoded MSD number to give T^- and W^- digits step and then summed the T^- and W^- bits to generate the final result we obtain a fast and correct result as shown in this example:

$$\begin{aligned} 1\bar{1}10\bar{1}10\bar{1}00 &= (364)_{10} \\ 1\bar{1}10\bar{1}\bar{1}\bar{1} &= (45)_{10} \end{aligned}$$

$$\begin{aligned} 01\bar{1}\bar{1}10000\bar{1}\bar{1} &= W^- \\ 00010000000 &= T^- \end{aligned}$$

$$01\bar{1}010000\bar{1}\bar{1} = (319)_{10}$$

Table (5-a): modification of table (3-a) using the proposed subtraction rules

$x_i y_i \setminus x_{i-1} y_{i-1}$	$\begin{array}{ c c } \hline T^- & W^- \\ \hline \end{array}$		
	$\bar{1}1$	$1\bar{1}$	otherwise
$\bar{1}\bar{1}/11$	0 0	0 0	0 0
00	0 0	0 0	0 0
$10/0\bar{1}$	0 1	$1\bar{1}$	0 1
$01/\bar{1}0$	$\bar{1}1$	$0\bar{1}$	$0\bar{1}$
$1\bar{1}$	1 0	1 0	1 0
$\bar{1}1$	$\bar{1}0$	$\bar{1}0$	$\bar{1}0$

If we remove the examination of the less significant bits $x_{i-1} y_{i-1}$ from table (5-a) and use (01) representation in the case where $x_i y_i$ are equal to $(10/0\bar{1})$ numbers, also we use $(0\bar{1})$ representation in the case where $x_i y_i$ are equal to $(01/\bar{1}0)$ numbers. Then applying the subtraction rules on this table we can note that, this change essentially has no effect on the final result. The reason belongs to the recoding method which do not permit the series of bits (11) and $(\bar{1}\bar{1})$ to appear in the recoded numbers. Based on this observation it is suitable to reduce table (5-a) to introduce a new table shown in table (5-b) which consist of two entries only instead of three entries was used in table(3-b) for the work of chirre and karim. Then if we apply the proposed method to the previous example we can note the same result as shown bellow:

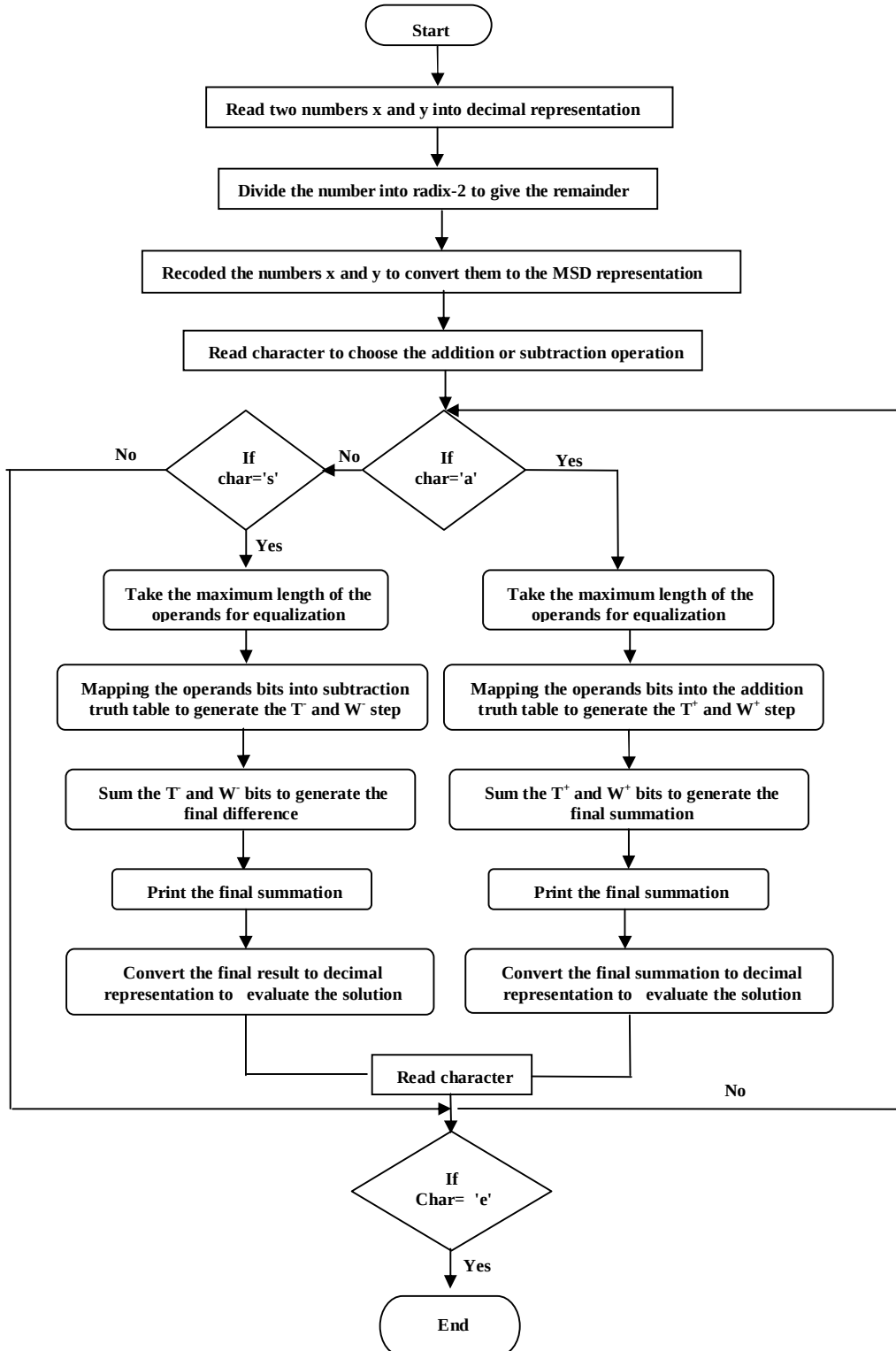
$$\begin{aligned} 1\bar{1}10\bar{1}10\bar{1}00 &= (364)_{10} \\ 1\bar{1}10\bar{1}1\bar{1} &= (45)_{10} \end{aligned}$$

$$\begin{array}{rcl} 01\bar{1}1\bar{1}0000\bar{1}1 & = & W^- \\ 000000000000 & = & T^- \\ \hline 01\bar{1}1\bar{1}0000\bar{1}1 & = & (319)_{10} \end{array}$$

Table (5-b): reduced rule for table (5-a)

$x_i y_i \setminus x_{i-1} y_{i-1}$	$\begin{array}{ c c } \hline T^- & W^- \\ \hline \end{array}$
$\bar{1}\bar{1}/11$	0 0
00	0 0
$10/0\bar{1}$	0 1
$01/\bar{1}0$	$0\bar{1}$
$1\bar{1}$	1 0
$\bar{1}1$	$\bar{1}0$

To illustrate the flow chart for the proposed method, Fig(2) will explains the steps for the carry-free addition and borrow-free subtraction operations.



Fig(2): flow chart for the proposed addition and subtraction

10. The MSD logic processing :

The MSD logic processing is implemented using the content addressable memory (CAM). In a CAM the truth table inputs which produce the same output logic level are generally grouped together. Then the logical minimization process is applied to reduce the number of minterms (maxterms) in the sum of product (product of sum) expression. The reduced minterm or maxterms are stored in an optical memory [2].

In this paper we obtain a result for the reduced minterms for T^+ , W^+ , T^- , W^- , S^+/D^+ , and S^-/D^- , as shown in table (7). This table produces either a 1 and $\bar{1}$ at the output. We can note that the $T(W)$ function requires two (one) minterms for both addition and subtraction operation compared with table (6) which illustrated the reduce minterms for earlier method by chirre and karim that required three (two) minterms for $T(W)$ function of the addition and subtraction tables. Furthermore, we eliminate the flag bit (F) which appear in table (6).

Table(6): minterms for MSD addition and subtraction operation for ref.7

Function	Output bit	Minterms $X_i Y_i F_i$	Function	Output bit	Minterms $X_i Y_i F_i$
T^+	$\begin{matrix} 1 \\ \bar{1} \end{matrix}$	$\begin{matrix} d_{01}11, 1d_{01}1, 11d_{01} \\ d_{01}\bar{1}0, 1d_{01}\bar{1}0, \\ \bar{1}1d_{01} \end{matrix}$	W^+	$\begin{matrix} 1 \\ \bar{1} \end{matrix}$	$\begin{matrix} 0d_{11}\bar{0}, d_{11}\bar{1}00 \\ 0d_{11}\bar{1}1, d_{11}\bar{1}01 \end{matrix}$
T^-	$\begin{matrix} 1 \\ \bar{1} \end{matrix}$	$\begin{matrix} d_{01}\bar{1}1, 1d_{01}\bar{1}1, \bar{1}1d_{01} \\ \bar{1}d_{01}0, d_{01}\bar{1}10, \bar{1}1d_{01} \end{matrix}$	W^-	$\begin{matrix} 1 \\ \bar{1} \end{matrix}$	$\begin{matrix} d_{11}\bar{1}00, 0d_{11}\bar{1}0 \\ 0d_{11}\bar{1}1, d_{11}\bar{1}01 \end{matrix}$
S^+/D^+	$\begin{matrix} 1 \\ \bar{1} \end{matrix}$	$\begin{matrix} 01, 10 \\ 0\bar{1}, \bar{1}0 \end{matrix}$	S^-/D^-	$\begin{matrix} 1 \\ \bar{1} \end{matrix}$	$\begin{matrix} 0\bar{1}, 10 \\ 01, \bar{1}0 \end{matrix}$
F^+	1	$d_{01}d_{01}$	F^-	1	$d_{01}d_{01}$

Table (7): minterms for MSD addition and subtraction operation for the proposed method

Function	Output bit	Minterms $X_i Y_i$	Function	Output bit	Minterms $X_i Y_i$
T^+	$\begin{matrix} 1 \\ \bar{1} \end{matrix}$	$\begin{matrix} 11 \\ \bar{1}\bar{1} \end{matrix}$	W^+	$\begin{matrix} 1 \\ \bar{1} \end{matrix}$	$\begin{matrix} 01, 10 \\ 0\bar{1}, \bar{1}0 \end{matrix}$
T^-	$\begin{matrix} 1 \\ \bar{1} \end{matrix}$	$\begin{matrix} \bar{1}1 \\ \bar{1}\bar{1} \end{matrix}$	W^-	$\begin{matrix} 1 \\ \bar{1} \end{matrix}$	$\begin{matrix} 0\bar{1}, 10 \\ \bar{1}0, 01 \end{matrix}$
S^+/D^+	$\begin{matrix} 1 \\ \bar{1} \end{matrix}$	$\begin{matrix} 01, 10 \\ 0\bar{1}, \bar{1}0 \end{matrix}$	S^-/D^-	$\begin{matrix} 1 \\ \bar{1} \end{matrix}$	$\begin{matrix} 0\bar{1}, 10 \\ 01, \bar{1}0 \end{matrix}$

11. Conclusions and Recommendations:

The proposed method deals with binary MSD truth tables for the addition and subtraction operations which can be able to reduce these tables by using two inputs only for each table instead of three inputs in the work of chirre and karim. Also we eliminate the examination of the less significant bits that appear in the earlier tables. Further reduction of optical implementation was obtained by using few minterms to be stored in an optical memory.

Finally, the parallelism inherent in the proposed arithmetic operations is well mapped to that of the optical system. This parallel algorithm offers a strong candidate for future applications in digital optical computing.

REFERENCE

- [1]. A. K. Cherri, M. S. Alam, and A. A. S. Awwal, (1997), "Optoelectronic symbolic substitution based canonical modified signed-digit arithmetic ", Optics & Laser Technology, Vol-29, No-3.
- [2]. Abdallah. K. Cherri and Mohammad. A. Karim, (1988), "Modified signed digit arithmetic using an efficient symbolic substitution", Applied Optics, Vol-27, No-18.
- [3]. Cherri. A. K, (1994), "Symmetrically recoded modified signed-digit optical addition and subtraction ", Applied Optics, Vol-33, Issue-20 .
- [4]. Guoqiang Li, Feng Qian, Hao Ruan, and Liren Liu,(1999), "Parallel optical negabinary signed-digit computing: algorithm and optical implementation", Optical Engineering, Vol-38, No-3.
- [5]. Hongxin Huang, Masahide Ltoh, Toyohiko Yatagai, and Liren Liu, (1996), "Classified one-step modified signed-digit arithmetic and it's optical implementation" , Optical Engineering, Vol-35, No-4 .
- [6]. K. W. Wong, L. M. Cheng, (1994),"Optical modified signed-digit addition based on binary logical operation ", Optics & Laser Technology, Vol-26, No-4.
- [7]. Shuqun Zhang, Mohammad A.Karim, (1999), "Optical arithmetic processing using improved redundant binary algorithms ", Optical Engineering, Vol-38, No-3.
- [8]. Shuqun Zhang, Mohammad A. Karim, (1999), "Programmable modified-signed-digit addition module based on binary logic gates", Optical Engineering, Vol-38, No-3.
- [9]. Y. Li and G. Eichmann, (1987), "Conditional symbolic modified signed-digit arithmetic using optical content-addressable memory logic elements", Applied Optics, Vol-26, Issue-12 .