Characteristics of Parallel Carry-Free Three-Step MSD Additions

SHEN YUNFU1, WANG ZHEHE1,2, PENG JUNJIE1,2, AND OUYANG SHAN1

1School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China
2College of Computer Science and Technology, Hainan Tropical Ocean University, Sanya 572022, China

Corresponding author: Shen Yunfu (yfshen@shu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFE0117500, and in part by the National Natural Science Foundation of China under Grant 61572305.

ABSTRACT
Since the modified signed digital (MSD) redundant representation was proposed in the 1950s, lots of achievements have been made in MSD arithmetic. By inspecting the processes of the parallel carry-free three-step MSD addition, the transformations for such additions are studied in detail in this paper. The characteristics for parallel carry-free three-step MSD addition are proposed and the correctness is proved. Then seven groups of transformations that have characteristics of three-step MSD addition are presented. These groups of transformations are dual or self-dual, and some of them have simpler forms than the typical transformations consisting of \( T, W, T', W', T_2 \). The general design mode of parallel carry-free three-step MSD additions and its applications in ternary optical computer (TOC) are further proposed. At the same time, single adder (multi-adder) reconstruction mode, processor bits allocation strategy, and light path diagrams are given. The optical experiments of MSD additions for three groups of MSD addition transformations show that the results of these transformations are correct. This work provides the theoretical basis for the design of the ternary optical computer adder.

INDEX TERMS
Ternary optical computer, modified signed digit (MSD) representation, ternary logic transformation, parallel computing, carry-free addition.

I. INTRODUCTION
Addition is the most fundamental arithmetic operation in most processors. But the problem of carrying propagation from the least-significant-bit to the most-significant-bit in conventional binary addition inevitably affects the computing speed. People commonly meet the difficulties in massive parallelism, high speed, and huge data processing in the electronic computer. To solve these problems, mathematicians and computer scientists have long been looking for fast and effective algorithms and have therefore proposed many non-binary number systems [1]–[3], such as the redundant number systems [4]–[6]. O Akbari et al. proposed a fast yet energy-efficient reconfigurable approximate carry look-ahead adder [7].

In the 1950s, A. Avizienis proposed the signed-digit representation [4], which eliminated the carry propagation chains for operands. The redundancy in number representation makes the fast and parallel addition and subtraction possible. Soon afterward, the modified signed-digit (MSD) representation is presented, which belongs to a redundant-number set of radix-2. In 1986, B. L. Drake, R. P. Bocker, et al. applied modified signed-digit representation to optical computing [8], [9] and suggested an optical architecture. It offered fully parallel carry-free arithmetic operations.

Now studies on optical computing algorithms and techniques have been made much progress and the optical computing modes have shown remarkable advantages in meeting the requirements such as parallel, carry-free, and fast computing. According to the value of the radix, the signed-digit representation contains the subsets of MSD number [5], [8]–[11], trinary [2], [3], [12], [13], quaternary [14]–[16] and negabinary signed-digit number [17], and so on. According to the steps of optical computing, the MSD additions are now usually classified in terms of the number of steps involved, i.e., three-step [5], [8]–[11], two-step [12]–[14], [18]–[20] and one-step algorithms [17], [21]–[24]. These two-step additions or one-step additions generally use more complex optical structures or focus on some specific data sets.

The MSD number system with the radix \( r = 2 \) and the digit set \( \{1, 0, 1\} \) is a typical signed-digit system. Since the MSD representation for optical computing was adopted...
TABLE 1. Ternary logic truth tables used in typical TW-MSD addition.

| $T(T_2)$ | $W$ | $T''$ | $W''$ |
|---|---|---|---|
| $a(t')$ | $b(w')$ | $b$ | $w$ |
| $\bar{a}$ | $\bar{b}$ | $\bar{t}$ | $\bar{w}$ |
| $0$ | $1$ | $0$ | $0$ |
| $1$ | $0$ | $1$ | $1$ |

in 1986 [8], and MSD arithmetic algorithms have been carried out deeply and several research directions and branches in optical computers have appeared. Scientists naturally consider using optical components to make optical computers, especially the all-optical computers [25], but the progress in design and development are not rapid. A new computational rule for redundant binary adder generating the intermediate sum and intermediate carry in posibit and inverted encoding of negabits representations replacing the redundant digits are proposed [26]. Since the principle and structure of the ternary optical computer was proposed by Jin Yi et al. [27], [28], lots of work has been done in this field [29]–[32].

Ternary optical computer (TOC) uses two polarized lights with orthogonal polarization states and a null light state to express information, and uses such device as liquid crystal pixel array to rotate the polarization direction of light in order to make light state conversion. It can have millions of processor bits to realize various operations. TOC has such features as easy processor bit scalability, huge processor bits, bitwise allocability, bitwise reconfigurability, parallel computing, and low power consumption and so on, which makes it more suitable for fast processing large batches of data with complex operations. In 2017, a prototype SD16 (short for “ShangDa 2016”, and “ShangDa” means Shanghai University) of the ternary optical processor (TOP) is constructed in Shanghai University, every module has 192 processor bits and has the functions described above. Much progress has been made in the design and development of the ternary optical computer [33]–[36].

In this situation, as early as 2010, the authors began to consider the problem of parallel carry-free MSD addition. By the summer of 2013, Shen Yunfu gave the rules of MSD addition transformation that can satisfy parallel computing. In 2016, many complete sets of transformation rules were formed and applied in the design of actual optical computer processors and one set of the MSD transformations is used to perform carry-free MSD addition in SD16. This work is summarized in this paper.

In this paper, by studying MSD number system with the radix $r = 2$ and the redundancy digit set $\{1, 0, 1\}$ and inspecting the generation of transfer and weight digits in MSD addition, the authors study three-step algorithms realizing MSD addition and obtain several groups MSD addition transformations. After executing these logic transformations, the weight and transfer digits and the final sum digits can be generated in three steps. In Section II, we give a brief introduction to MSD representation and the typical three-step MSD addition. In Section III, by inspecting the typical three-step MSD addition, we give the characteristics for parallel carry-free three-step MSD addition and the corresponding ternary logic transformations with these characteristics. In Section IV, we give the general design mode of parallel carry-free three-step MSD additions and its applications. Then optical experiments of MSD addition transformations are carried out. Finally, we summarize our work in Section V.

II. BRIEF INTRODUCTION TO MSD ADDITION

MSD system was proposed by Avizienis in 1961 [4], for any decimal number $A = a_na_{n-1}\cdots a_1a_0a_{-1}\cdots$ can be represented by an MSD number representation shown in formula (1).

Here we discuss the MSD number system of radix $r = 2$. The modified signed-digital redundant representation uses $\{\bar{1}, 0, 1\}$ to present numbers, which is simply a binary system with no sign bit, where the symbol $\bar{1}$ stands for -1.

$$A = \sum_i a_i2^i$$

where $a_i \in \{\bar{1}, 0, 1\}$ and $i$ is an integer. Any decimal number $A$ except for 0 has many MSD representations. For example, $(13)_{10} = (\overline{1101})_{\text{MSD}} = (\overline{1111})_{\text{MSD}} = (\overline{111101})_{\text{MSD}}$.

Let $A$ and $B$ be two MSD numbers $a_na_{n-1}\cdots a_1$ and $b_nb_{n-1}\cdots b_1$ respectively. Then the $n$-bit MSD addition of $A$ and $B$ can be carried out in three steps according to TW transformations in Table 1, and its computing processes are listed in Table 2 [29]. We will call it TW-MSD addition.

III. CHARACTERISTICS FOR PARALLEL CARRY-FREE THREE-STEP MSD ADDITION

A. CHARACTERISTICS OF CARRY-FREE THREE-STEP MSD ADDITIONS

Since the TW transformations mentioned in Section II is only a special case of the ternary logic transformations discussed in this paper. For the sake of generality and distinguish from TW-MSD addition, we use five ternary logic transformations $Y, F, Y', F', S$ instead of $T, W, T', W', T_2$, and the new five transformations are called YFS transformations for short. Let $A$ and $B$ be two MSD numbers $a_na_{n-1}\cdots a_1$ and $b_nb_{n-1}\cdots b_1$ respectively. The mode of carry-free
three-step MSD addition of $A$ and $B$ with five ternary logic transformations $Y, F, Y', F'$, $S$ on $\{0, 1, \bar{1}\}$ is described as follows.

**Step 1:** Apply the $Y$ transformation to the numbers $A$ and $B$ bit by bit to obtain a result denoted by $y$. Append one 0 at the end of $y$ and denote the result with $y_0$. Meanwhile, apply the $F$ transformation to $A$ and $B$ bit by bit to obtain a result denoted by $f$. Add one 0 at the head of $f$ and denote it with $f_0$ too.

**Step 2:** Apply the $Y'$ transformation to $y$ and $f$ bit by bit to obtain the result $y'$. Append one 0 at the end of $y'$ and still denote it with $y'$. Meanwhile, apply the $F'$ transformation to $y$ and $f$ bit by bit to obtain the result $f'$. Add one 0 at the head of $f'$ and denote it with $f'_0$ too.

**Step 3:** Apply $S$ transformation to $y'$ and $f'$ bit by bit to obtain $s$. Now $s$ is the sum of $A$ and $B$. Computing processes of the general MSD addition are described in Table 3.

Next, we discuss the establishment conditions of the YFS transformations to realize the carry-free three-step MSD addition.

**Theorem 1:** Given five ternary logic transformations $Y, F, Y', F', S$ on $\{0, 1, \bar{1}\}$, suppose that they satisfy the following equations (2), (3), and (4)

\[
\begin{align*}
    a + b &= y \times 2 + f \\
    y + f &= y' \times 2 + f' \\
    y' + f' &= s, \quad y' = f' \neq \bar{1} \quad \text{and} \quad y' = f' = \bar{1} (4)
\end{align*}
\]

where $y = Y(a, b), f = F(a, b), y' = Y'(y, f), f' = F'(y, f), s = S(y', f'),$ and $a, b, y, f, y', f', s \in \{0, 1, \bar{1}\}$. Then for any $n$-bit MSD numbers $A = a_n a_{n-1} \cdots a_2 a_1$ and $B = b_n b_{n-1} \cdots b_2 b_1$, the sum of $A$ and $B$ can be done in three steps with the processes described above using transformations $Y, F, Y', F', S$ successively.

Furthermore, if the two cases which $y'_{i-1} = f'_{i} = 1$ and $y'_{i-1} = f'_{i} = 1$ don’t occur in the processes of addition, then the group consisting of five transformations $Y, F, Y', F', S$ is a ternary transformations group of parallel carry-free MSD addition.

**Proof:** The theorem can be proved by mathematical induction on $n$.

Consider any two $n$-bit MSD numbers $A$ and $B, A = a_n a_{n-1} \cdots a_2 a_1$ and $B = b_n b_{n-1} \cdots b_2 b_1$.

If $n = 1$, the sum of $A$ and $B$ is $a_1 + b_1, a_1 + b_1 = y_1 \times 2 + f_1$ at the first step, $0 + f_1 = y'_1 \times 2 + f'_1, y_1 + 0 = y'_2 \times 2 + f'_2$ at the second step, and $s_1 = 0 + f'_1 = f'_1$, $y_1 + f'_1 = s_2$, $s_3 = y'_2 + 0 = y'_2$ at the third step. Here, $y'_1 + f'_1$ has no carry-bit according to the assumption of the theorem. The computing processes are listed as follows:

\[
\begin{align*}
    a_1 &\quad b_1 \\
    y_1 &\quad 0 \\
    0 &\quad f_1 \\
    y'_2 &\quad y'_2 \\
    0 &\quad f'_1 \\
    f'_2 &\quad f'_2
\end{align*}
\]

Here, $s_3 s_2 s_1 = y'_2 s'_2 f'_1$ is the result of the transformations $y'_2 s'_2 f'_1 = 4 \times y'_2 + 2 \times s_2 + f'_1 = 4 \times y'_2 + 2 \times (y'_1 + f'_1) + f'_1 = 2 \times (2 \times y'_2 + f'_2) + 2 \times y'_1 + f'_1 = y_1 \times 2 + f_1 = a_1 + b_1$. The proposition is true when $n = 1$.

Suppose that the proposition is true when $n = k - 1$.

Now let $n = k$. Denote $C = a_n a_{n-1} \cdots a_2 a_1$ and $D = b_n b_{n-1} \cdots b_2 b_1$. We have $A = C \times 2 + a_1, B = D \times 2 + b_1$.

Obviously, $A + B = (C + D) \times 2 + (a_1 + b_1)$. Now we do MSD additions $A + B$ and $C + D$ by applying transformations $Y, F, Y', F', S$ respectively. By induction assumption, $e_n + 2 e_{n-1} \cdots 2 e_1$ is the sum of $C$ and $D$.

Denote the result $s_3 s_2 s_1 = y'_2 s'_2 f'_1$ according to the processes of MSD addition by applying transformations $Y, F, Y', F', S$. Now it only needs to prove that $s_3 s_2 s_1 = e_n + 2 e_{n-1} \cdots e_1 e_2 \times 2 + (a_1 + b_1)$.

We list the processes of MSD additions $A + B$ and $C + D$ in Table 4.

Obviously, we have $s_i = e_i$ for $i = 4, 5, \cdots, n + 2$.

Now it only needs to prove that $s_3 s_2 s_1 = e_3 e_2 \times 2 + (a_1 + b_1)$ is correct, which is to prove (5).

\[
s_3 \times 2^2 + s_2 \times 2 + s_1 = e_3 \times 2^2 + e_2 \times 2 + (a_1 + b_1) (5)
\]
Denote the right side of (5) by \( V \). By the condition of the theorem, obviously, \( e_2 = 0 + f''_2 = f''_1 \). Since the cases \( y''_2 + f''_1 = 2 \) and \( y''_2 + f''_1 = -2 \) do not appear according to the condition of the theorem, the equation \( e_3 = y''_2 + f''_1 \) always holds. Similarly, \( s_3 = y'_2 + f'_1, s_2 = y'_1 + f'_2, \) and \( s_1 = 0 + f'_1 \). We also have \( a_1 + b_1 = y_1 \times 2 + f_1, f_1 = y'_1 \times 2 + f'_1, y_1 + f_2 = y'_2 \times 2 + f'_2, f_2 = y''_2 \times 2 + f''_2 \). We have the following relations.

\[
V = e_3 \times 2^2 + e_2 \times 2 + (a_1 + b_1)
\]

\[
= e_3 \times 2^2 + (e_2 + y_1) \times 2 + f_1
\]

\[
= e_3 \times 2^2 + (e_2 + y_1) \times 2 + (y'_1 \times 2 + f'_1)
\]

\[
= (y''_2 + f''_1) \times 2^2 + (f''_2 + y_1) \times 2 + y'_1 \times 2 + f'_1
\]

\[
= f'_1 \times 2^2 + [(y''_2 \times 2 + f''_1) + y_1] \times 2 + y'_1 \times 2 + f'_1
\]

\[
= f'_1 \times 2^2 + (y''_2 + f''_1) \times 2 + y'_1 \times 2 + f'_1
\]

\[
= (y''_2 + f''_1) \times 2^2 + (y'_1 + f'_2) \times 2 + f'_1
\]

\[
= s_3 \times 2^2 + s_2 \times 2 + s_1
\]

Thus, we have proved that \( V = s_3 \times 2^2 + s_2 \times 2 + s_1 \). That is, the proposition is true when \( n = k \).

Hence it is obvious that the theorem is proved by the induction.

**B. THE TERNARY LOGIC TRANSFORMATIONS WITH CHARACTERISTICS OF THREE-STEP MSD ADDITION**

In this section, we will find the ternary logic transformations which satisfy the condition in Theorem 1. Now we inspect the processes of carry-free MSD addition.

For any two MSD numbers \( A = a_0a_{n-1}a_{n-2} \cdots a_2a_1 \) and \( B = b_nb_{n-1}b_{n-2} \cdots b_2b_1 \), we have that

\[
a_i + b_i = y_i \times 2 + f_i, \quad i = 1, 2, \cdots, n \tag{6}
\]

\[
y_{i-1} + f_i = y'_i \times 2 + f'_i, \quad i = 1, 2, \cdots, n, n + 1, \tag{7}
\]

\[
y_0 = 0, \quad f_{n+1} = 0 \tag{8}
\]

Now consider a general addition mode on the number set \( \{1, 0, 1\} \) based on Table 5.

| \( a \) | \( b \) | \( b \) | \( 1 \) |
|---|---|---|---|
| 1 | (1) | (0,1) or (1,1) | (0,0) |
| 0 | (0,1) or (1,1) | (0,0) | (0,1) or (1,1) |
| 1 | (0,0) | (0,1) or (1,1) | (0,0) |

Considering various combinations of all the values in Table 6 by splitting the transfer digit and weight digit, we can obtain 16 possible transformation groups, each of which consists of two transformation tables. But some of them do not satisfy the MSD addition requirements of Theorem 1.

For any five transformations \( Y, F, Y', F' \), \( S \) on \( \{1, 0, 1\} \), suppose that they satisfy conditions (2), (3) and (4) and the additional condition that \( y''_{i-1} = f'_i = 1 \) or \( \bar{1} \) don’t occur in the processes of MSD addition in the theorem. Now we discuss when the situations \( y''_{i-1} = f'_i = 1 \) or \( y''_{i-1} = f''_i = \bar{1} \) appear from the last step to the first step of the processes of three-step MSD addition.

Since \( 1 = 0 \times 2 + 1 = 1 \times 2 + \bar{1} \), and \( \bar{1} = 0 \times 2 + \bar{1} = \bar{1} \times 2 + 1 \), we need to consider all combinations by checking the processes of the computing steps in Table 3. According to condition (7), \( y''_{i-1} \) is associated with the equation \( y_{i-2} + f_{i-1} = y''_{i-1} \times 2 + f''_i \) and \( f''_i \) is associated with the equation \( y''_{i-1} + f_i = y'_i \times 2 + f'_i \). So we need to check the four values.
we denote them and y. Transformation tables from combinations

\textbf{TABLE 7.} Characteristics of Parallel Carry-Free Three-Step MSD Additions

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|}
\hline
Y & b & 1 0 1 \\
\hline
a & 1 & 1 1 0 \\
0 & 1 & 0 1 0 \\
1 & 0 & 1 1 1 \\
\hline
\end{tabular}
\begin{tabular}{|c|c|c|}
\hline
F & 1 0 1 \\
\hline
a & 1 & 0 1 0 \\
0 & 1 & 0 0 1 \\
1 & 0 & 0 1 1 \\
\hline
\end{tabular}
\begin{tabular}{|c|c|c|}
\hline
Y' & f & 0 0 0 \\
\hline
y & 1 & 0 0 0 \\
0 & 1 & 0 0 1 \\
1 & 0 & 0 1 0 \\
\hline
\end{tabular}
\begin{tabular}{|c|c|c|}
\hline
F' & 1 0 1 \\
\hline
y & 1 & 1 0 0 \\
0 & 1 & 0 1 0 \\
1 & 0 & 1 1 0 \\
\hline
\end{tabular}
\end{tabular}
\end{table}

A. \(P=(Y', F')\) with \((v, x)=((0, 1)\) and \((t, w)=((1, 1))\)

B. \(Q=(Y', F')\) with \((v, x)=((1, 1)\) and \((t, w)=((0, 1))\)

By condition (6), we have \(a_i + b_i = y_i \times 2 + f_i, a_i-1 + b_i-1 = y_i-1 \times 2 + f_i-1\). So \(y_i-1, f_i, f_i-1, y_i-2\) are associated with \(a_i, b_i, a_i-1, b_i-1, a_i-2, b_i-2\).

Now, we list 8 basic transformation conditions. For brevity, we denote them:

\(A : Y(0, 1) = 1, F(0, 1) = 1; A' : Y(0, 1) = 0, F(0, 1) = 1;\)

\(B : Y(0, 1) = 1, F(0, 1) = 1; B' : Y(0, 1) = 0, F(0, 1) = 1;\)

\(C : Y(1, 0) = 1, F(1, 0) = 1; C' : Y(1, 0) = 0, F(1, 0) = 1;\)

\(D : Y(1, 0) = 1, F(1, 0) = 1; D' : Y(1, 0) = 0, F(1, 0) = 1;\)

Consider the combination of basic conditions, we have the four basic combinatorial relations:

\((A, C'), (A', C), (B, D')\) and \((B', D)\).

We then deduce the new results that \(Y'(0, 1) = 0\) and \(F'(0, 1) = 1\) are derived from \((A, C')\) or \((A', C)\), and \(Y'(0, 1) = 0\) and \(F'(0, 1) = 1\) are derived from \((B, D')\) or \((B', D)\).

The incompatible cases may occur in the following two computing modes:

\[
\begin{array}{c}
* a_i & a_{i-1} & a_{i-2} \\
* b_i & b_{i-1} & b_{i-2} \\
\end{array}
\]

\[
\begin{array}{c}
y_i & y_{i-1} & y_{i-2} & * \\
* & f_i & f_{i-1} & f_{i-2} \\
\end{array}
\]

\[
\begin{array}{c}
y'_i & 1 & * & * \\
* & 1 & f'_{i-1} & * \\
\end{array}
\]

\[
\begin{array}{c}
* & s_i & * & * \\
\end{array}
\]

and

\[
\begin{array}{c}
* a_i & a_{i-1} & a_{i-2} \\
* b_i & b_{i-1} & b_{i-2} \\
\end{array}
\]

\[
\begin{array}{c}
y_i & y_{i-1} & y_{i-2} & * \\
* & f_i & f_{i-1} & f_{i-2} \\
\end{array}
\]

So there are some other combinations that are conflicted. There are four conflicting combinations: \((A', B'), (A', D'), (C', D')\) and \((B', C')\). The counter examples are 000 + 110, 010 + 100, 110 + 000 and 100 + 010 respectively. The details are omitted here.

Now we discuss the possible combinations of the above four basic combinations. According to the above discussions, we obtain the following all possible compatible cases:

\[ABCD, A'BCD, AB'CD, ABC'D, ABC'D, A'BD'CD, A'BC'D.\]

The corresponding transformation tables are presented in Tables 7-13.

Here the symbol * and # can be any digit from \((1, 0, 1)\). The letters labeled \(t, w, v, x\) are optional which satisfy the corresponding relations \(1 = t \times 2 + w\) and \(1 = v \times 2 + x\), that is, \((t, w) = (1, 1)\) or \((0, 1)\) and \((v, x) = (1, 1)\) or \((0, 1)\). By selecting the values of \(t, w, v, x\) in Table 7, we will obtain four groups of transformations \(Y'\) and \(F'\).

The transformations \(T, W, T', W'\) of TW-MSD adders are just special cases of Table 7, where \((t, w) = (0, 1)\) and \((v, x) = (0, 1)\).

Furthermore, we will obtain another nine forms of transformation \(S\) by selecting the values \# and * from \((1, 0, 1)\). This will provide great convenience in designing ternary optical adders.

Definition: (i) Two ternary logic truth tables \(M\) and \(N\) are said to be dual if one table becomes another by exchanging the symbols 1 and \(\bar{1}\) in one table.

(ii) \(M\) is said to be self-dual if \(M\) remains unchanged when the symbol 1 and \(\bar{1}\) are exchanged each other in \(M\). (iii) Given two sets \(\tau\) and \(\tau'\) of transformations with same number, \(\tau\) and
\[ \tau' \text{ are said to be dual if there is a one to one relation from } \tau \text{ to } \tau' \text{ so that the corresponding transformations are dual. Especially, a set of transformations } \tau \text{ is said to be self-dual if any transformation in } \tau \text{ is self-dual. By the above definition, we have the following:} \]

1. The set of transformations \( T, W, T', W' \) is self-dual.
2. The set \( \{Y', F'\} \) of transformations with \((v, x) = (\bar{1}, 1)\) and \((t, w) = (1, \bar{1})\) in Table 7 are self-dual.
3. The set \( P = \{Y', F'\} \) of transformations obtained with \((v, x) = (0, \bar{1})\) and \((t, w) = (0, 1)\) in Table 7 are dual, as shown in Table 7A and 7B.

The dual carry-free MSD addition transformations are usually regarded as the same.

Since \((t, w) = (1, \bar{1})\) or \((0, 1)\) in Table 12, we have two simplified forms listed in Table 12A and Table 12B.

Similarly, we have the simplified forms 13A and Table 13B of Table 13.

The transformations in Table 12 and Table 13 are also dual when \((t, w) = (0, 1)\) and \((v, x) = (0, \bar{1})\), or \((t, w) = (1, \bar{1})\) and \((v, x) = (\bar{1}, 1)\). That is, transformations in Table 12A and Table 13A are dual and transformations in Table 12B and Table 13B are dual.

Next, we prove that the cases \(y_i' = f_i' = 1\) and \(y_i' = \bar{1}\) mentioned in Theorem 1 cannot appear during the processes of three-step transformations \(Y, F, Y', F', S\) for any group of transformations successively. The correctness of TW-MSD addition has been recognized by scholars. But for other transformations listed in Table 8-13, due to paper limitations, we can only list the following conclusions.
TABLE 12. Transformation tables from combinations $AB'CD'$.

A. Simplified Transformation tables of Table 12

| $Y'$ | $F'$ | $Y''$ | $F''$ | $S$ |
|------|------|-------|-------|-----|
| 0    | 1    | 0     | 0     | 0   |
| 1    | 0    | 0     | 1     | 1   |

B. Simplified Transformation tables of Table 12

| $Y'$ | $F'$ | $Y''$ | $F''$ | $S$ |
|------|------|-------|-------|-----|
| 0    | 1    | 0     | 0     | 0   |
| 1    | 0    | 0     | 1     | 1   |

TABLE 13. Transformation tables from combinations $A'B'C'D'$.

A. Simplified Transformation tables of Table 13

| $Y'$ | $F'$ | $Y''$ | $F''$ | $S$ |
|------|------|-------|-------|-----|
| 0    | 1    | 0     | 0     | 0   |
| 1    | 0    | 0     | 1     | 1   |

B. Simplified Transformation tables of Table 13

| $Y'$ | $F'$ | $Y''$ | $F''$ | $S$ |
|------|------|-------|-------|-----|
| 0    | 1    | 0     | 0     | 0   |
| 1    | 0    | 0     | 1     | 1   |

Theorem 2: For any two MSD numbers $A = a_na_{n-1} \cdots a_2a_1$ and $B = b_nb_{n-1} \cdots b_2b_1$, the cases $y'_{i-1} = f'_i = 1$ or $\overline{y}'_{i-1} = f'_i = \overline{1}$ cannot appear after applying the transformations $Y, F, Y', F'$ in Table 8 to $A$ and $B$ successively.

Proof: To shorten the length of this article, the proof is omitted. (Please refer to Appendix A).

Theorem 3: For any two MSD numbers $A = a_na_{n-1} \cdots a_2a_1$ and $B = b_nb_{n-1} \cdots b_2b_1$, the cases $y'_{i-1} = f'_i = 1$ and $\overline{1}$ cannot appear by applying any
set transformations $Y, F, Y', F'$ of Table 9-13 to $a$ and $b$ successively.

Proof: The proof is similar to Theorem 2 and is omitted here. (Please refer to Appendix B).

Theorem 4: All transformations $Y, F, Y', F'$ and $S$ given in Table 7-13 have the characteristics of parallel carry-free three-step MSD addition.

Proof: The theorem is proved by Theorem 1, Theorem 2 and Theorem 3.

IV. GENERAL DESIGN MODE OF PARALLEL CARRY-FREE THREE-STEP MSD ADDITIONS

A. BRIEF INTRODUCTION TO THE TERNARY OPTICAL COMPUTER AND ITS PROTOTYPE SD16

Figure 1 is the construction schematic diagram of TOC. It consists of two parts: an upper computer (a master computer, a common PC) and a lower computer (a slave computer, or the ternary optical processor). The master computer is responsible for task management and communicating with users as well as the slave computer. That is, it generates the reconstruction information and receives information from the optical processor. TOC uses two polarized lights with orthogonal polarization states and a null-light state to express information, and uses such device as liquid crystal pixel array to rotate the polarization direction of light in order to perform light state conversion. It can have millions of processor bits to realize various operations. One of the core units in “Ternary optical processor” is the “Optical state converter”; another is “Reconfigure unit” for reconstructing frame and induction, which realizes the three light states transition. The processor reconstruction makes TOC reconstruct its functions for various operators required by the user. The input numbers are $a$ and $b$, where $a$ is used as the data of main optical path through the Signal Generator of Main Light Path, and $b$ is used as the data of control optical path through the Signal Generator of Control Light Path. The calculated result beam is sent to the TOC decoder by the control system in the slave computer to obtain the user’s binary result data.

Figure 2 is a prototype SD16 developed based on the above construction schematic of TOC. Each of its modules has 192 processor bits and can load up to 64 modules. Therefore, each prototype SD16 has a maximum of 12,228 processor bits. At present, each processor bit can be reconfigured into a bit of any ternary (including binary) logical operator, and some processor bits can be used to construct a parallel carry-free adder. The processor is reconfigurable according to user’s requirements.

In SD16, a liquid crystal board LCD with $24 \times 24$ grids (or pixels) is served as a processor, and the layout of the pixel bits in the liquid crystal is arranged in the form as in Figure 3, where three adjacent pixels in the same row form a processor bit. Here, the LCD board is divided into two parts: the left and right parts. The twelve grids at the bottom of the left part are divided into 4 sections from right to left, which are
numbered as 0, 1, 2, 3 accordingly. Each section forms a processor bit. Every grid in a section can output one of three light status: null light (W), horizontally polarized light (H) or vertically polarized light (V). Hence every LCD board has 192 processor bits. For example, the light status of the four processor bits numbered 3, 2, 1, 0 at the bottom of the image in Figure 3 are (HWW), (WWW), (WVW) and (HWW), which are decoded as MSD number 10101 respectively.

**B. SINGLE ADDER RECONSTRUCTION MODE**

The bits of optical processor can be numbered from 0 to MAX − 1 in some order, where MAX is the number of processor bits. Here MAX = 192. According to the calculation mode of n-bit MSD addition in Section III.A, the MSD adder is designed and configured as follows. Let the position of the adder be start. Then in the processor, the Y transformer is reconstructed from start to start + (n − 1), and the transformer F is reconstructed from start + n to start + 2n − 1. Similarly, transformers Y′, F′ and S are reconstructed from start + 2n to start + 3n, start + 3n + 1 to start + 4n + 1 and start + 4n + 2 to start + 5n + 3 respectively. The configuration of MSD adder with n = 5 bits is shown in Figure 4, where start = 0. Generally, an n-bit MSD adder uses 5n + 4 bits.

**C. MULTI-ADDER RECONSTRUCTION MODE**

It is assumed that t adders need to be configured on the processor. Suppose that the starting position of the i-th adder is pos[i], and the number of adder bits is len[i] and pos[1] = start. If the centralized reconstruction mode shown in Figure 5 is adopted in the processor, then pos[i+1] = pos[i] + 5 × len[i] + 4. The independent bit allocation mode of three MSD adders of k-bit, m-bit and n-bit is shown in Figure 5.

**D. THE APPLICATION OF MSD ADDITION TRANSFORMATIONS TO THE DESIGN OF MSD ADDER**

In the parallel carry-free MSD adder transformations given in Table 7-13, there are several groups of transformations that are relatively simple. Here, we take a group of transformations as an example to further clarify these transformation tables that meet the conditions of the carry-free parallel adder. The following optical path diagram is a schematic diagram of the five transformations in Table 13B. The light path diagrams of Y, F, Y′ and F′, S transformations are shown in Figure 6, 7, 8 respectively.

In these optical path diagrams, the rectangles marked 1L, LC2 represent optically inactive liquid crystal. It is not in the photon spin state when there is no control signal acting on crystal, and it is photon spin state the control signal acting on the crystal. At this time, if the light directed to the liquid crystal is vertical light, the light transmitted through the liquid crystal becomes horizontal light and vice versa; the diamonds represent vertical polarizers (only allow vertical light to pass through the polarizers) or horizontal polarizers (only allow horizontal light to pass through the polarizers); LD1 and LD2 are photoelectric converters, it can convert optical signals into electrical signals; “Light” represents uniform scattering white light source, it includes three states (null light, vertical light, horizontal light). The grey lines with an arrow are the main light paths and the thin lines with an arrow are the control light paths and the grey lines with an arrow are the light paths.

It generally assumes that value 0 is represented by a null light state, value 1 is represented by vertically polarized light, and value 1 is represented by horizontally polarized light.

**E. OPTICAL EXPERIMENTS OF MSD ADDITION TRANSFORMATIONS**

Here we take a group of transformations as examples to illustrate that the conclusion is correct. We take three sets of carry-free MSD transformations in Table 8, 10 and 13A. The MSD numbers are selected randomly.

For example. Let A = (457526569)10, B = (709383477)10. Choose the MSD representations of A and B as (011101110111111101011001)MSD and (1111101101001111010110111001)MSD respectively.
TABLE 14. Theoretical results of MSD addition $A + B$ with transformations in table 8, 10, and 13A.

| A   | B   | $y$  | $f$  | $y'$ | $f'$ |
|-----|-----|------|------|------|------|
| 01101101011101011101011101011101 | 11101011011101011101011101011101 | 11010111011101011101011101011101 | 11010111011101011101011101011101 | 01010111011101011101011101011101 |

$\alpha + \beta$ 457526589+709383477+1166910346 457526589+709383477+1166910346 457526589+709383477+1166910346

The theoretical results of MSD additions $A + B$ with the transformations in Table 8, 10, and 13A are listed in Table 14.

Next, we first generate reconstruction instructions according to the data with Table 8, 10, 13A and the single adder reconstruction mode, and then send them to the ternary optical processor through the slave computer, respectively, to construct three 30-bit adders. Therefore, every adder requires a total of 154 ($30 \times 5 + 4 = 154$) processor bits, of which processor bits $No.0 \sim No.9$ are allocated to $Y$ transformation, and $No.30 \sim No.59$ are allocated to $F$ transformation, $No.60 \sim No.90$ are allocated to $Y'$ transformation, $No.91 \sim No.121$ are allocated to $F'$ transformation, $No.122 \sim No.153$ are allocated to $S$ transformation. Secondly, the operands $A$ and $B$ are passed through the slave computer in turn send to the ternary optical processor, after three steps of transformation, their final results $s$ and intermediate results $y, f, y', f'$ are all displayed on the LCD board. Here, we only decode the final result $s$ (the rest result are omitted) as follows:

The optical image of addition with Table 8 is shown in Figure 9, and its decoding result $s$ is 0100110101011101010011110111010100 from the high processor bit to the low bit.

The optical image of addition with Table 10 is shown in Figure 10, its decoding result $s$ is 0100110101011101010011110111010100 from the high processor bit to the low bit.

The optical image of addition with Table 13A is shown in Figure 11, its decoding result $s$ is 0100110101011101010011110111010100.

The three decoding results $s$ are all $(1166910046)_{10}$, which are completely consistent with the theoretical results in Table 14.

The three experiments of MSD addition $A + B$ show that the transformations in Table 8, 10, and 13A for MSD additions are correct.
s. yunfu et al.: characteristics of parallel carry-free three-step MSD additions

v. conclusion

recently, great progress has been made in the research of ternary optical processor. the parallel ternary optical adder based on MSD representation has been equipped with the basic module of SD16 system and run in Shanghai University successfully. This progress marks the advent of new processors with a large number of data bits, data-on-demand allocation and processor configuration based computing functions on demand. In this paper, the parallel carry-free three-step MSD addition transformations are studied deeply, and seven groups of transformations for MSD additions with this characteristic are obtained. They are all alternatives which will be used in the design of carry-free adders of the ternary optical computer. Some of the transformations are rather simple, which is shown in Table 12 or 13. Experiments of MSD additions for MSD transformations in Table 8, 10 and 13a show that the results are correct.

in fact, the transformations $Y'$ and $F'$ in Table 12 or 13 can be merged into a transformation $(Y'/F')$, in other word, this two group transformation only need 4 transformation tables respectively, therefore, the MSD adders of one bit using them that will only use 4 processor-bits, which greatly decrease the hard resource with 25% compared to TW-MSD adder with transformations $T, W, T', W', T_2$. Furthermore, the work in this paper provides a theoretical basis for the design of the carry-free adder of the ternary optical computer.

appendix

a. proof of theorem 2

Theorem 2: For any two MSD numbers $A = a_na_{n-1} \cdots a_2a_1$ and $B = b_nb_{n-1} \cdots b_2b_1$, the cases $y'_{i-1} = f'_i = 1$ or $y'_{i-1} = f'_i = \bar{1}$ cannot appear after applying the transformations $Y, F, Y', F'$ in Table 8 to $A$ and $B$ successively.

Proof: Suppose that during the transformations $Y, F, Y', F'$, the intermediate results $y'_{i-1} = f'_i = 1$ or $y'_{i-1} = f'_i = \bar{1}$ appear.

Firstly, we consider the case $y'_{i-1} = f'_i = 1$.

$y'_{i-1} = 1$ is the result of applying transformation $Y'$ to $y_{i-2}$ and $f_{i-1}$. In $Y'$ transformation table there is only one place of value 1 which is corresponding to the input $(y, f) = (1, 1)$ in $Y'$ transformation table. Hence $(y_{i-2}, f_{i-1}) = (1, 1), f_{i-1} = 1$. $f'_i = 1$ is the result of applying transformation $F'$ to $y_{i-1}$ and $f_i$. There are three places of value 1 which are corresponding to input data $(y, f)$. So $(y_{i-1}, f_i) = (0, 1), (0, 1)$ or $(1, 0)$, and $y_{i-1} = 0$ or 1. Thus $(y_{i-1}, f_{i-1}) = (0, 1)$ or $(1, 1)$. But there is no such input data $(a_{i-1}, b_{i-1})$ that $y_{i-1} = Y(a_{i-1}, b_{i-1})$ and $f_{i-1} = F(a_{i-1}, b_{i-1})$ with $(y_{i-1}, f_{i-1}) = (0, 1)$ or $(1, 1)$.

Next we consider the case $y'_{i-1} = f'_i = \bar{1}$.

There are two places of value 1 in transformation table $Y'$ which are corresponding to input data $(y, f)$, that is, $(y, f) = (1, 1)$ or $(0, 1)$. Hence $(y_{i-2}, f_{i-1}) = (1, 1)$ or $(0, 1)$, and $f_{i-1} = \bar{1}$. Similarly, there is only one place of 1 in transformation table $F'$ which is corresponding to the input data $(y, f)$, that is, $(y_{i-1}, f_i) = (1, 0)$, and $y_{i-1} = \bar{1}$. Hence $(y_{i-1}, f_{i-1}) = (1, 1)$. But there is no such input data $(a_{i-1}, b_{i-1})$ that $y_{i-1} = Y(a_{i-1}, b_{i-1})$ and $f_{i-1} = F(a_{i-1}, b_{i-1})$ with $(y_{i-1}, f_{i-1}) = (1, 1)$.

It shows that the cases $y'_{i-1} = f'_i = 1$ and $y'_{i-1} = f'_i = \bar{1}$ cannot appear in the processes of transformations. The theorem is proved.
B. PROOF OF THEOREM 3

Theorem 3: For any two MSD numbers $a = a_na_{n−1} \ldots a_2a_1$ and $b = b_nb_{n−1} \ldots b_2b_1$, the cases $y′_{i−1} = f′_i = 1$ and $\bar{1}$ cannot appear by applying any set transformations $Y, F, Y', F'$ of Table 9-13 to $a$ and $b$ successively.

Proof: Note that the transformations in Tables 8 and 9, Tables 10 and 11 as well as Tables 12 and 13 are dual respectively. If the theorem is true for one group of transformations, the theorem is also true for dual group of transformations. By Theorem 2, it is true for Table 9. Here we only need to discuss Table 10 and Table 12.

Suppose that during the transformations $Y, F, Y', F'$, the intermediate results $y′_{i−1} = f′_i = 1$ or $y′_{i−1} = f′_i = \bar{1}$ appear.

Firstly we discuss Table 10. Consider the case $y′_{i−1} = f′_i = 1$.

On the other hand, $y′_{i−1} = 1$ is the result of applying transformation $Y'$ to $y_{i−2}$ and $f_{i−1}$. In $Y'$ transformation table there is only one place of value 1 which is corresponding to the input $(y, f) = (1, 1)$ in $Y'$ transformation table, Hence $(y_{i−2}, f_{i−1}) = (1, 1)$. But there is no such input data $(a_{i−1}, b_{i−1})$ that $y_{i−1} = Y(a_{i−1}, b_{i−1})$ and $f_{i−1} = F(a_{i−1}, b_{i−1})$ with $(y_{i−1}, f_{i−1}) = (1, 1)$. It shows that the cases $y′_{i−1} = f′_i = 1$ and $y′_{i−1} = f′_i = \bar{1}$ cannot appear in the processes of transformations. The theorem is proved for Table 13.

ACKNOWLEDGMENT

Prof. Jin Yi would like to thank all Ph.D. candidates and master graduate students of their research center for their kind help and valuable discussions in preparing the article.

REFERENCES

[1] D. S. Phatak and I. Koren, “Hybrid signed-digit number systems: A unified framework for redundant number representations with bounded carry propagation chains,” IEEE Trans. Comput., vol. 43, no. 8, pp. 890–891, Aug. 1994.

[2] A. Awwal, M. Islam, and M. A. Karim, “Modified signed-digit carry-ripple subtracting using optical symbolic substitution,” Opt. Lett., vol. 31, no. 11, pp. 1687–1694, 1999.

[3] M. S. Alam, “Efficient quasi-digit symbolic arithmetic,” Opt. Eng., vol. 19, no. 5, pp. 353–355, 1994.

[4] A. Avizienis, “Signed-digit number representations for fast parallel arithmetic,” IEEE Trans. Electron. Comput., vol. EC-10, no. 3, pp. 389–400, Sep. 1961.

[5] L. R. Cherri and M. A. Karim, “Modified-signed digit arithmetic using an efficient symbolic substitution,” Opt. Lett., vol. 27, no. 18, pp. 3824–3827, 1998.

[6] S. Mukkamala, P. Rathore, and R. P. Peesapat, “Decomplex multiplication using compressor-based BCD to binary converter,” Eng. Sci. Technol., Int. J., vol. 21, no. 1, pp. 1–6, Feb. 2018.

[7] O. Akbari, M. Kamal, A. Azfali-Kusha, and M. Pedram, “RAP-CLA: A reconfigurable approximate carry look-ahead adder,” IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 65, no. 8, pp. 1089–1093, Aug. 2018.

[8] B. L. Drake, R. P. Bocker, M. E. Lasher, R. H. Patterson, and W. J. Miceli, “Photonic computing using the modified signed-digit number representation,” Opt. Eng., vol. 25, no. 1, 1986, Art. no. 250138.

[9] R. P. Bocker, B. L. Drake, M. E. Lasher, and T. B. Henderson, “Modified signed-digit addition and subtraction using optical symbolic substitution,” Appl. Opt., vol. 25, no. 15, pp. 2456–2457, 1986.

[10] B. Ha and Y. Li, “Parallel modified signed-digit arithmetic using an optoelectronic shared content-addressable-memory processor,” Appl. Opt., vol. 33, no. 17, pp. 3647–3662, 1994.

[11] D. Casasent and P. Woodford, “Symbolic substitution modified signed-digit adder,” Appl. Opt., vol. 33, no. 8, pp. 1498–1506, 1994.

[12] M. S. Alam, M. A. Karim, A. A. S. Awwal, and J. Westerkamp, “Optical processing based on conditional higher-order trinary modified signed-digit symbolic substitution,” Appl. Opt., vol. 31, no. 26, pp. 5614–5621, 1992.

[13] A. K. Cherri, M. K. Habib, and M. S. Alam, “Optoelectronic recoded and nonrecoded trinary signed-digit adder that uses optical correlation,” Appl. Opt., vol. 37, no. 11, p. 2153, 1998.

[14] G. Li, L. Liu, H. Cheng, and H. Jing, “Simplified quaternary signed-digit arithmetic and its optical implementation,” Opt. Commun., vol. 137, nos. 4–6, pp. 389–396, May 1997.

[15] S. Dubey, R. Rani, S. Kumari, and N. Sharma, “VLSI implementation of fast addition using quaternary signed digit number system,” in Proc. IEEE Int. Conf. Emerg. Trends Comput., Commun. Nanotechnol. (ICETCCN), Mar. 2013, pp. 654–659.

[16] P. Harreesh, C. K. Chakravathi, and D. T. Rao, “Fast multiplication with partial products using quaternary signed digit number system,” in Computational Intelligence in Data Mining, vol. 2, H. S. Behera and D. P. Mohapatra, Eds. New Delhi, India: Springer, 2016, pp. 437–446.

[17] A. K. Cherri and H. A. Kamal, “Efficient optical negabinary modified signed-digit arithmetic: One-step addition and subtraction algorithms,” Opt. Eng., vol. 43, no. 2, pp. 420–426, 2004.

[18] R. S. Fathy, A. A. W. Alsaffar, and M. S. Alam, “Optical two-step modified signed-digit addition based on binary logic gates,” Opt. Commun., vol. 208, nos. 4–6, pp. 263–273, Jul. 2002.
[19] Y. Shen, B. Jiang, Y. Jin, O. Shan, and J. Peng, “Principle and design of ternary optical accumulator implementing M-k-B addition,” Opt. Eng., vol. 53, no. 9, Sep. 2014, Art. no. 095108.

[20] F. Qian, G. Li, H. Ruan, H. Jing, and L. Liu, “Two-step digit-set-restricted modified signed-digit addition– subtraction algorithm and its optoelectronic implementation,” Appl. Opt., vol. 38, no. 26, pp. 5621–5630, 1999.

[21] H. Huang, M. Itoh, T. Yatagai, and L. Liu, “Classified one-step modified signed-digit arithmetic and its optical implementation,” Opt. Eng., vol. 35, no. 4, pp. 1134–1140, 1996.

[22] S. Zhang and M. A. Karim, “One-step optical negabinary and modified signed-digit adder,” Opt. Laser Technol., vol. 30, nos. 3–4, pp. 193–198, Apr. 1998.

[23] Y. Shen and L. Pan, “Principle of a one-step MSD adder for a ternary optical computer,” Sci. China Inf. Sci., vol. 57, no. 1, pp. 1–10, Jan. 2014.

[24] Y. Shen, L. Pan, Y. Jin, J. Peng, and B. Jiang, “One-step binary MSD adder for ternary optical computer,” Scientia Sinica Informatica, vol. 42, no. 7, pp. 869–881, 2012.

[25] A. S. Al-Zayed and A. K. Cherri, “Improved all-optical modified signed-digit adders using semiconductor optical amplifier and Mach–Zehnder interferometers,” Opt. Laser Technol., vol. 42, no. 5, pp. 810–818, Jul. 2010.

[26] R. K. Barik, B. K. Bhoi, and M. Pradhan, “An efficient redundant binary adder with revised computational rules,” Comput. Electr. Eng., vol. 72, pp. 224–236, Nov. 2018.

[27] J. Yi, H. Huaan, and L. Yangtian, “Ternary optical computer architecture,” Phys. Scripta, p. 98, 2005, doi: 10.1238/physica.topical.118a00098.

[28] Y. Jin, H. Wang, S. Ouyang, Y. Zhou, Y. Shen, J. Peng, and X. Liu, “Principles, structures, and implementation of reconfigurable ternary optical processors,” Sci. China Inf. Sci., vol. 54, no. 11, pp. 2236, 2011.

[29] Y. Jin, Y. Shen, J. Peng, S. Xu, G. Ding, D. Yue, and H. You, “Principles and construction of MSD adder in ternary optical computer,” Sci. China Inf. Sci., vol. 53, no. 11, pp. 2159–2168, Nov. 2010.

[30] J. Peng, R. Shen, Y. Jin, Y. Shen, and S. Luo, “Design and implementation of modified signed-digit adder,” IEEE Trans. Comput., vol. 63, no. 5, pp. 1134–1143, May 2014.

[31] Q. Xu, X. Wang, and C. Xu, “Design and implementation of the modified signed digit multiplication routine on a ternary optical computer,” Appl. Opt., vol. 56, no. 16, pp. 4661–4669, 2017.

[32] J. Peng, X. Wei, X. Zhang, Y. Shen, and Y. Fu, “Implementation of parallel FFT algorithm on a ternary optical computer,” Scientia Sinica Informatica, vol. 47, no. 7, p. 846, 2017.

[33] P. Junjie, F. Youyi, Z. Xiaofeng, K. Shuaui, and W. Xinyu, “Implementation of DFT application on ternary optical computer,” Opt. Commun., vol. 410, pp. 424–430, Mar. 2018.

[34] S. Zhang, J. Peng, Y. Shen, and X. Wang, “Programming model and implementation mechanism for ternary optical computer,” Opt. Commun., vol. 428, pp. 26–34, Dec. 2018.

[35] J. Jiabao, S. Yunfu, O. Shan, P. Junjie, and W. Xianchao, “The application of SJ-MSD adder to mean value filtering processing,” Optik, vol. 206, Mar. 2020, Art. no. 164271.

[36] S. Zhang, Y. Shen, and Z. Zhao, “Design and implementation of a three-lane CA traffic flow model on ternary optical computer,” Opt. Commun., vol. 470, Sep. 2020, Art. no. 125750.

SHEN YUNFU was born in 1960. He received the Ph.D. degree in science from Beijing Normal University, Beijing, in 1996. He is currently a Professor with Shanghai University. His research interests include reliability and operation units for ternary optical computers, formalization methods for software and hardware, and model checking.

WANG ZHEHE was born in 1980. He is currently pursuing the Ph.D. degree with the School of Computer Engineering and Science, Shanghai University. He is currently a Lecturer with Hainan Tropical Ocean University. His main research interests include storage architecture, file systems, embedded systems, and ternary optical computers.

PENG JUNJIE was born in 1977. He received the Ph.D. degree in engineering from the Harbin Institute of Technology, Harbin, in 2006. He is currently a Professor. His research interests include cloud computing, mass storage, dependable computing, wireless sensor networks, and optical computing.

OUYANG SHAN was born in 1979. He received the Ph.D. degree in engineering from Shanghai University, Shanghai, in 2012. He is currently a Lecturer with Shanghai University. His research interests include optical computer and architecture.

* * *