

# Optimization Techniques for Floating-point Division of Ternary Optical Computer

Qun XU \*, Lei WANG

\* 2011769998@qq.com

School of Computer Engineering and Science, Shanghai University, Shanghai, 200444, China

Pages: 463–477

**Abstract:** In order to increase the iterative speed of division and enhance data-bits' utilization efficiency of the optical processor, the theory and implementation of the floating-point division had been researched and optimized. The SRT algorithm was chosen to simplify the iterative process. Using the two-step M+B adder reduced the steps of addition operation. Due to the feature of asymmetric structure, the speed of ternary optical processor had been improved. The computational accuracy was increased by allocating the data-bits of optical processor based on bits. Simulation experiments prove the practicality and high efficiency of the improved floating-point division routine. Furthermore, the results of comparative calculation show that its ability in terms of the large-value data processing and high precision computation has been largely enhanced. The research on optimal techniques provides new technical routes for improving the efficiency of floating-point division.

**Keywords:** Floating-point division, Optical computer, SRT algorithm, Iteration speed

## 1. Introduction

In the four fundamental operations, the research on addition, subtraction, multiplication and their implementation techniques in arithmetic unit is relatively mature, which has made significant progress. In contrast, since the limited executive speed and complicated design, the research on division develops slowly which is a difficult point in processor design (Jin, 2003; González, 2015). Moreover, scientists generally believe that the frequency of division usage is relatively low which does not require much attention (Oberman, 1997; Jin, 2011). However, research has shown that the strategy and optimization of division can lead to degradation in performance in many application environments. Thus, there is an urgent need to design a high efficient division combined with the features of the processor (Jin, 2010).

Traditional electronic computers are restricted by CPU with a fixed number of bits. When researchers design a divider, many problems occur, such as high complexity of hardware implementation, long delay of calculation, and poor parallelism. However, relying on a large amount of data-bits and its bitwise reconfiguration, ternary optical computer (TOC) shows its strong abilities when processing big-value data and complicated data, which has become a powerful tool for improving division (Shen, 2014). TOC expresses information by adopting no light and two perpendicular polarization states. It processes

information by using liquid crystal array and polarizers together (Russinoff, 2013). The principle and structure of TOC make its application characteristics different from that of the electronic computers —one is that there are a number of data-bits in ternary optical computer, which can be allocated based on bits; the other is its bitwise reconfiguration that any part of the data width in optical processor can be constructed as a ternary logical calculus according to the calculation requirements (Shen, 2010). TOC adopts MSD numerical system to establish MSD adder, giving full play to these two major advantages of “ternary” and “a large number of data-bits” in numerical calculation. On the basis of decrease-radix design principle, TOC can construct adders according to the user’s needs at any time. These application features of TOC lay the theoretical foundation for the efficient implementation of the floating-point division routine.

In 2014, we learned the application characteristics of TOC and established a high-speed division theory and implementation of a modified signed digit (MSD) iterated algorithm, presenting strict verification in a simulated environment. However, in the further research on the implementation scheme of the algorithm, we found that: if more efficient Sweeney Robertson Tocher (SRT) algorithm is used, the operational steps of division can be simplified, thereby the speed of calculation can be improved (Yan, 2008); if the two-step MSD adder is used instead of three-step MSD adder in literature (Xu, 2016), the calculation speed of division of ternary optical computer can be greatly improved; if in consideration of the asymmetric structure of ternary optical processor, the original data is allocated to control light path rationally so that the operational speed of ternary optical processor can be enhanced; if original data and calculating are represented by significant digit, and the data-bits of optical processor is accordingly distributed, then the computed value can be ensured to meet user’s requirements, which improves the processing ability of the floating-point division routine in terms of the large value data and high precision arithmetic. The work discussed the principles of the four important improvements and the corresponding division calculation process.

## 2. Using SRT division algorithm to improve the algorithmic model of division routine

The research on SRT division is originated from 1950s. Initially, it was proposed by Sweeney, Robertson and Tocher, greatly improving the computational efficiency of floating-point division at that time. By setting different internal parameters, the SRT algorithm can flexibly change the implementation plan, reducing the hardware complexity of the division operation. It is one of the most widely used algorithmic models among the modern processors.

The division operation can be reflected by  $N=D\times Q+W$ , where  $N$  represents dividend MSD value with  $n$  significant digits;  $D$  represents the divisor MSD value with  $m$  significant digits;  $Q$  represents the ultimate quotient MSD value with  $F$  significant digits;  $W$  represents the ultimate remainder;  $F$  represents the maximum iterations.

The floating-point division routine of ternary optical computer uses the SRT algorithm as the mathematical model, and the calculation steps are as follows:

Step1: Preprocessing operation—determine the sign of the quotient and take the absolute value of the operand, and then perform a shift to ensure the original data to satisfy  $\frac{1}{2} \leq N < 1$ ,  $\frac{1}{2} \leq D < 1$ , and record the number of bits and direction of the shift;

Step2: Iterative initialization—assign 0 to the loop control variable  $i$ , and assign normalized N to the initial value  $W_o$  of the partial remainder, and calculate the max iterations F according to the users' requirements;

Step3: Parallel implementation of the four group of MSD adds operation;

Send  $\frac{1}{2}$  and  $W_{i-1}$  to MSD adder and MSD subtractor, and write the results in the variables  $W1$  and  $W2$ ; send D and  $W_{i-1}$  to MSD adder and MSD subtractor, and write the results in variables  $W3$  and  $W4$ ;

Step4: Choose the quotient  $Q_i$  and calculate the partial remainder  $W_i$ . Scan the most significant digit of  $W1$  and  $W2$ , and write in the variables  $W1Value$  and  $W2Value$ , then implement the following optical judgment rule:

When  $W1Value$  is 0 or  $\bar{1}$ ,  $Q_i=\bar{1}$  and  $W_i=2W3$ ;

When  $W1Value$  is 1 and  $W2Value$  is 1,  $Q_i=0$ ,  $W_i=2W_{i-1}$ ;

When  $W2Value$  is 0 or  $\bar{1}$ ,  $Q_i=1$  and  $W_i=-2W4$ .

Step5: Add 1 to  $i$ , and repeat Step3~Step4 until  $i=F-1$ ;

Step6: Processing the results. Connect all the quotient bit one by one according to the form of  $Q_1Q_2\dots Q_F$ , and adjust the decimal position and the symbol of the quotient, then the quotient in a correct form can be obtained. The algorithm flow chart is shown in Figure 1.

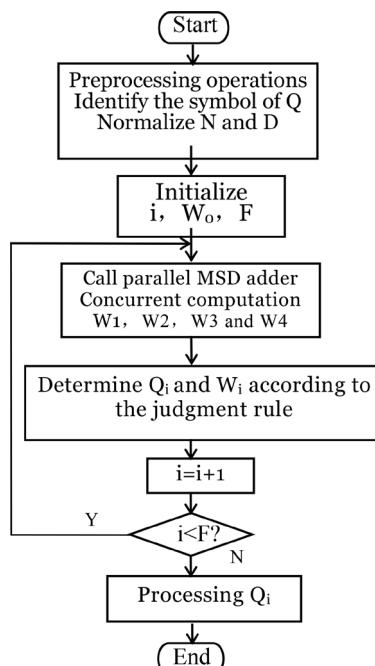


Figure 1 – SRT algorithm flow chart of floating point division routine

Taking  $N=(10\bar{0}0010\bar{0}1\bar{0}10\bar{0}110\bar{0}1\bar{0}0)$ <sub>MSD</sub> =  $(6497970)_{10}$ ,  $D=(101001101000101010001010)$ <sub>B</sub> =  $(10914442)_{10}$  for an example to demonstrate the calculating process of SRT method, the subscript represents the digital system. After preprocessing, N shifts 23 bits to right and is adjusted to  $1.0\bar{0}0010\bar{0}1\bar{0}10\bar{0}110\bar{0}1\bar{0}00$ , D shifts 24 bits to right and becomes  $0.101001101000101010001010$ , initialize  $W_0=N$ .

The first iteration:

$$W_1=10.\bar{1}0\bar{1}0\bar{1}1\bar{1}\bar{1}0\bar{1}0\bar{1}0\bar{1}1\bar{1}00;$$

$$W_2=00.0\bar{0}0\bar{1}1\bar{0}0\bar{1}0\bar{1}1\bar{0}1\bar{1}00\bar{0}0;$$

$$W_3=10.\bar{1}0\bar{1}0\bar{1}01\bar{0}1\bar{0}100000\bar{1}100\bar{1}0;$$

$$W_4=00.0\bar{0}1000001\bar{1}0000\bar{1}101\bar{1}010\bar{1}0;$$

According to the judgment rule,  $W_2\text{Value}=\bar{1}$ , so  $Q_1=1$ ,  $W_1=-2W_4$ ;

The second iteration:

$$W_1=0001.0\bar{1}000000\bar{1}00010\bar{0}100\bar{1}01\bar{1}00;$$

$$W_2=0001.\bar{1}100001\bar{1}10000\bar{1}1\bar{1}1\bar{1}0\bar{1}00;$$

$$W_3=0001.00\bar{1}010\bar{1}0000100001\bar{1}0000\bar{1}0;$$

$$W_4=0001.\bar{1}0\bar{1}0100\bar{1}0001\bar{1}1010\bar{1}010\bar{1}0;$$

According to the judgment rule, when  $W_1\text{Value}$  is 1 and  $W_2\text{Value}$  is 1, then  $Q_2=0$ ,  $W_2=-2W_1$ ;

After iterative calculation in turn for 18 times is finished, each quotient can be connected in the order of  $Q_1, Q_2, \dots, Q_{18}$ . Shifting quotient one bit to right, we can obtain that  $Q=(0.10011000100\bar{1}00101)_{MSD}=(0.59535\dots)_{10}$ .

The improvements of SRT algorithm in terms of iterative division routine are as follows:

1. After the preprocessing, all data-bits of the dividend become the initial value of the partial remainder, eliminating the continuous operation, which will simplify each iterate operation.
2. The range of MSD value is  $\{\bar{1}, 0, 1\}$ , which can naturally represent the redundant quotient, thus eliminating the converting part of the electronic computer and reducing the complexity of the hardware design.
3. Introducing the selected constant  $\pm \frac{1}{2}$  to improve the judging condition of choosing  $Q_i$  and  $W_i$ . The selected constant is a fixed value, and then we can introduce the two-step adder and speed up the calculating process by taking advantage of the asymmetric structure of the ternary optical processor, which will be discussed in the following.

### 3. Using two-step MSD adder to improve the iteration speed

Among the characteristics of the ternary optical computer, the working speed of the adder has the biggest impact on the calculation speed of division. Since there is no carry in the

adder adopting redundant digital system, the addition of each data bit is synchronous parallel processed. The total duration of the addition has been determined, which is irrelevant to the number of the digits of the data. These features are suitable for the addition operation that is associated with numerous data bits. Thus, TOC uses a binary MSD adder, which is a redundant counting method. MSD adder uses  $\{i, 0, 1\}$  to indicate  $-1$ ,  $0$  and  $1$ , complying with the principle of binary. It hides the carry operation in the redundant representation in the process of addition, so that the adder avoids the serial process of carry operation, achieving the parallel addition. The ternary optical computer has three physical statuses saying the information. By using  $0$ ,  $i$  and  $1$  to represent the non-optical status and two other polarization status, the MSD adder can be naturally formed. In 1970s, mathematicians have concluded that the MSD addition can be done by four ternary logical calculus in three steps. The steps and time series are shown in the Figure 2. The completion of an addition operation takes 3 clock cycles. Based on the three-step adder, literature presents a two-step adder that uses 3 ternary logical calculus in two steps. The computation requires 2 clock cycles, and the steps and time series are shown in Figure 3. Two-step adder is a parallel adder that restricts the input symbols. It requires an input of MSD binary number  $M$ , another input of conventional binary number  $B$ , and the output is a MSD value. Thus, the two-step adder is also called  $M+B$  adder.

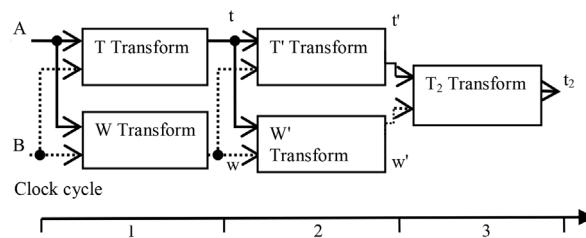


Figure 2 – Operation steps and the time sequential routine of three-step addition

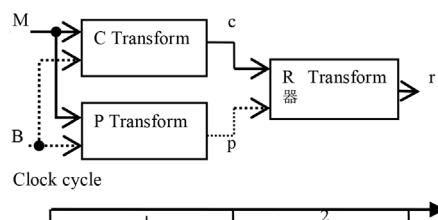


Figure 3 – Operation process and time sequential routine of  $M+B$  addition

By comparing the two MSD adders, it can be concluded that: after  $M+B$  adder is replaced by floating-point division routines, iterative computation of  $F$  times costs  $2F$  clock cycles. Considering the data preparation and quotient arrangement, the total duration of a division operation is  $2F+2$  clock cycles, saving 30% calculation time compared to that of three-step adder which is  $3F+2$  clock cycles; the  $M+B$  adder needs to construct the

data-bits of optical processor to 3 ternary logical converters C, P and R, while the three-step MSD adder needs to construct the data-bits of optical processor to corresponding 5 ternary logical converters T, W, T', W' and T2. Thus, using M+B adder can save about 40% of the data-bits resources.

#### 4. Using the asymmetric structure of ternary optical processor to improve the response speed

Ternary optical processor has an asymmetric structure. As is shown in Figure 4, SY1 and SY2 are optical generators with three states—the controlled optical paths and main optical paths respectively. CG represents the reconfigurable circuit;  $P_1$ ,  $P_2$  and  $P_3$  are polarizers; Lc is the liquid crystal display pixels; g is the photosensitive.

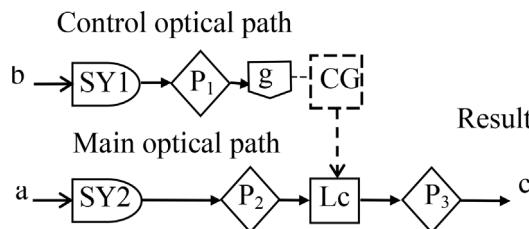


Figure 4 – Asymmetric structure of optical

It consists of two parts: the main optical path and the controlled optical path. The main optical path adopts the structure that places an Lc in between two polarizers -  $P_2$  and  $P_3$ , and the controlled optical path adopts the structure that places the photosensitive tube g behind polarizer  $P_1$ .  $P_1$ ,  $P_2$  and  $P_3$  can be horizontal polarizer or vertical polarizer; Lc can be constant optical rotation controlled liquid crystal unit or constant optically inactive controlled liquid crystal unit. In the controlled optical path, there can be  $P_1$  or input optical signal. The two original data—a and b—are sent to the main optical path and the controlled optical path, respectively. After b passes through the photosensitive tube, the elec-signal can be generated for Lc. This signal will affect the rotation state of Lc. While a passes through  $P_2$ , Lc and  $P_3$  successively, the optical state changes into c, namely the output of signal light. The optical processor transforms the input of optical signals—a and b—into the output c. Thus, the computing time of a signal light is determined by the duration which lights a passes through the optical processor (two polarizers with a liquid crystal pixel in between). The working process is as follows:

T1: Preparation stage—data b and a are sent from storage to SY1 and SY2.

T2: Signal light generating stage—from receiving a and b by SY1 and SY2 to sending the light with three states.

T3: Response time of photosensitive tube —from receiving the light with three states by g to sending corresponding elec-signals.

T4: Response time of reconfiguration circuit —the elec-signals sent by g pass through the reconfiguration circuit CG.

T5: Reaction time of liquid crystal—from receiving the elec-signals by Lc, which is sent by g, to completing the change of optical rotation.

T6: Calculating time—from receiving the light by Lc, which is sent by SY2 to produce the results of calculation c.

The duration which the light passes through the free space and each polarizer can be ignored since it is very short, and the elements are closed to each other. In most of the computing devices, the state of the liquid crystal of main optical path will change with the signals sent from controlled optical path. The calculating speed of the system is determined by the speed of change in liquid crystal state (response frequency of liquid crystal). In terms of the calculation where the main optical liquid crystal state remains unchanged, such as logical computing NOT and comparison operations, then the calculating speed is irrelevant to the response speed of liquid crystal. It is only determined by the duration of which a passes through two polarizers  $P_2$  and  $P_3$  with an Lc in between. In the six stages of the working process of optical processor, T3, T4 and T5 are closely associated with b. When the photosensitive tube detected b, T3 will be produced as the output; only by passing through CG, the elec-signal sent by g can impact Lc, thus generating T4; under the control of elec-signals sent by g, Lc can change its optical rotation, thus generating T5. Although the signal light a sent by SY2 has passed through Lc in the stage of T3+T4+T5, the output is not a signal result which would be abandoned by the system since it has no use.

Only in the T6 stage after T3+T4+T5, we can obtain the calculating result c. If value b remains unchangeable after several times of continuous calculation, then T3, T4 and T5 should be occurred in the first calculation. However, in the following calculation, the value of b will stay the same, then the optical rotation of Lc will be unchanged, which means that Lc has been well prepared. After that, T6 can be occurred after T2 without having to experience T3+T4+T5. Thus, as long as informing the optical processor that the value of b will stay the same in advance, there is no need to consider the duration of T3+T4+T5. Thus, the following operation after the first calculation is completed in T1+T2+T6. T2 and T5 of ternary optical processor are all determined by the response time of liquid crystal. At present, the shortest response time is at  $10^{-9}$  order second, which is approximately several dozen times more than other durations. Thus, the calculation time of optical processor is about  $2 \times T5$ . Therefore, the corresponding calculating speed can be doubled if the value of b remains.

Considering each iteration of SRT algorithm, the input of optical processor are  $W_{i-1}$ , D and  $\frac{1}{2}$ . Among them,  $W_{i-1}$  is the remainder of the  $i-1$  iteration which will change with i. However, D remains unchangeable in a division, and  $\frac{1}{2}$  remains unchangeable in all the divisions in SRT algorithm. Thus we improve the floating-point division routine in the following two aspects:

1. D and  $\frac{1}{2}$ , as the value of b, is sent to the controlled optical path.
2. The optical processor is informed that the signal c can be received after T2 from the second time.

This improvement makes use of the asymmetric structure of ternary optical processor and the M+B adder that consists of ternary logical optical processors. It remains asymmetrical between the controlled optical path and the main optical path in terms of the hardware structure. Therefore, the speed of iteration operation will be doubled from the second iteration after sending D and to the controlled optical path.

## 5. Improving the computation accuracy by distributing the data-bits of optical processor based on digits

SRT division algorithm has been strictly proved that it is convergent. Each iteration cycle generates a quotient bit  $Q_i$ . With the increasing number of iterations F, the stable part of quotient value will be increased from high position to low position gradually, and the calculation results will tend to be closed to the theoretical value. The electronic computer calculates all the data according to the fixed calculation precision. The calculation process will not consider the actual needs of the users. After completing the calculation process, the results will be output directly and be processed by the user. When the ternary optical computer performs the floating-point division, the MSD significant digit F of the quotient in binary (computational accuracy) is determined by the user's demand. Ternary optical computer is used to calculate the floating-point division, which is different from the electronic computer. In addition, original input data and the significant digit of the quotient will serve as a basis for determining the required data-bit resources. The division operation is completed by allocating data-bits of the optical processor based on the bits, satisfying user's requirements in terms of computational accuracy.

In the input interface of ternary optical computer, users will provide with dividend, significant digits of the divisor n, m and the significant G of the quotient required by the user when the user input the operation request and the original data. G is a digit in decimal system in line with the user habits. However, in the ternary optical computer, the numerical representation and the computation should adopt the MSD binary digit. The division routine should calculate F based on G, thus determining the maximum iterations in the division algorithm. The conversion rule is as follows: It is supposed that G includes the integer part  $G_1$  and the decimal part  $G_2$ , namely  $G=G_1+G_2$ . The conversion of integer part can be obtained by the formula  $2^{F_1}=10^{G_1}$ , thus  $F_1=G_1 \times 1/2$ . The conversion of decimal part  $G_2$  can be obtained based on the formula below:

$$(1/2)^1 + (1/2)^2 + \dots + (1/2)^F = 9 \times ((1/2)^1 + (1/2)^2 + \dots + (1/2)^G)$$

$$\frac{1}{2} = 9 \times \left( \frac{\frac{1}{10} \times \frac{1}{2} (1 - (\frac{1}{10} \times \frac{1}{2})G)}{1 - \frac{1}{10} \times \frac{1}{2}} \right)$$

$(1/2)^F = (1/2)^G$ , namely  $F_2 = G_2 \times \frac{\ln 10}{\ln 2}^{1/2}$ . Thus  $F = F_1 + F_2 = G \times \frac{\ln 10}{\ln 2}^{1/2}$ . The integer portion is

taken upward when determining the iterative number, with a single inaccurate digit added at the end. After processing, the MSD computational accuracy of the quotient is

$$F = [G \times \frac{\ln 10}{\ln 2}^{1/2}] + 1.$$

After determining the calculation accuracy F, the maximum digits of input data that involves the iterations can be determined according to the formula  $q = \max\{n, m\} + 1$ . It is based on the formula  $V = q + 2F - 2$ . Then, the computational scale of M+B adder should be identified. According to the formula  $V_T = 4(3V + 1) = 12p + 24F - 20$ , the total amount of data-bits of the optical processor should be determined. Based on the digits, the required databits of the division operation should be allocated, in order to design the specific optical processor solution for division operation. Taking  $N = 10010101011010101010011_{(MSD)} = 7539921_{(10)}$ ,  $D = 100010010101010100010010_{(MSD)} = 9000210_{(10)}$  for an example, the calculation of five groups is completed by setting different parameters (See Table 1). From a mathematical point of view, the accurate value of 7539921 divided by 9000210 is 0.8377494525127747019 23621..... After analyzing the five groups of numerical examples, it can be found that when completing the division calculation according to the input of significant digit G, the value F will increase as G increases, which means that the iterative number will increase with the increased requirements of the computational accuracy of quotient. Then, the ternary optical computer may increase the data-bits  $V_T$  of the optical processor according to these structural parameters, ensuring that the significant digit G of the calculation result is accurately calculated.

When the input data and the computational accuracy of decimal calculation that is required by user is less than 17 (i.e., the computational accuracy of binary calculations

Example	significant digit	parameter setting	Result Q	
			MSD number	Decimal system
1	q=24 G=6	F=21 VT=772	0.110101100111011011000	0.837749
2	q=44 G=12	F=41 VT=1492	0.1101011001110110110000 0010000100110010001	0.837749452512
3	q=54 G=15	F=51 VT=1852	0.1101011001110110110000 0010000100110010001100 0110000	0.837749452512774
4	q=64 G=18	F=61 VT=2212	0.1101011001110110110000 0010000100110010001100 01100001001011100	0.837749452512774701
5	q=74 G=21	F=71 VT=2572	0.1101011001110110110000 000100001001100100011 0001100001001011100111 0101010	0.837749452512774701923

Table 1 – Comparison of the results and parameters based on bitwise operation

is less than 54), the electronic computer and the ternary optical computer can all output the satisfied result. If the original input data has a large value, or the significant digit G of the quotient that is required to be calculated is big, the ternary optical computer may calculate the total task in real time. The iterative division routine will increase the steps of the iterative cycle. The management process of the data-bits will allocate more data-bits of optical processor to participate the operation. Thus, the ternary optical computer can effectively deal with high accuracy calculation and the significant digit inflation of the input data. However, the current electronic computer does not have the ability. The latest applied experimental system SD11 can independently allocate 16,384 digits for users. According to the formula of  $V_T$  and F, SD11 can deal with the floating-point division algorithm that involves original input of 455 MSD significant digits in binary and the quotient. It ensures the accurate calculation of 136 significant digits in decimal system, and can increase its calculation ability in terms of big value data and high accuracy with the expansion of the data-bits of optical processor. However, the current electronic computer equipped with double precision floating-point supports calculation of 53 conventional digits in binary, only ensuring 15 significant digits in decimal system. The calculation accuracy of ternary optical computer in terms of the division operation is 8.5 times of the accuracy of electronic computer.

## 6. Experimental verification

### 6.1. Experimental module design

1. Variables are defined. The integer variable n, m, G, F and q, and the numerical arrays inputbuf are defined; the input data of the simulation experiment is recorded; the two-dimensional integer arrays N\_INPUT, D\_INPUT and Q\_OUTPUT are defined; a group of dividend, divisor and the obtained quotient digit are recorded; the integer variables sign and one-dimensional integer arrays D, N, W, U and Q for the iteration of division are defined; the one-dimensional integer arrays W1, W2, W3 and W4 are defined to receive the output of two-step M+B adder and subtractor in the iterative operation of SRT division routine; one-dimensional integer arrays Half is defined and the conventional binary form of the selected constant  $\frac{1}{2}$  is recorded; the integer variables Noffset and Doffset are defined, and the shift digits after the normalization of dividends and divisors are recorded; the W1Index, W1Value, W2Index and W2Value are defined, and the position and the value of the most significant digit of W1 and W2 are recorded respectively;
2. The keyboard reading function is constructed to obtain the structural parameters n, m and G of ternary optical processor. According to n and m, the max digit that involves the iterations can be calculated, and then restore it in variable q; we also calculate the significant binary digit of the corresponding quotient, and then restore it in variable F;
3. The original data input function is constructed. The original data of the simulation experiment is received through inputbuf, and then they are restored into arrays N\_INPUT and D\_INPUT. Among them, the variables restored in N\_INPUT[i][j] are MSD binary digits, while the variables restored in D\_INPUT[i][j] are conventional binary digits.

4. The subfunctions  $C\_trans$ ,  $P\_trans$  and  $R\_trans$  are constructed to simulate  $C$  conversion,  $P$  conversion and  $R$  conversion. The truth table of  $C$ ,  $P$  and  $R$  conversion are defined respectively by using the two-dimensional integer arrays  $C[2][3]=\{0, 0, 1, 0, 1, 1\}$ ,  $P[2][3]=\{-1, 0, -1, 0, -1, 0\}$  and  $R[2][2]=\{0, 1, -1, 0\}$ . We also construct the subfunctions  $C\_trans$ ,  $P\_trans$  and  $R\_trans$ , and record the output of these subfunctions respectively by using one-dimensional integer arrays  $c$ ,  $p$  and  $s$ . In terms of the addition of MSD digits  $m$  with  $MAX\_LENGTH$  bits and the binary digit  $b$ , the  $C$  conversion is implemented by using the formula  $c[i]=C[b[i]][m[i]+1]$ , the subfunction  $P\_trans$  is implemented by using the formula  $p[i+1]=P[b[i]][m[i]+1]$ , the subfunction  $R\_trans$  is implemented by using the formula  $s[i]=R[-p[i+1]][c[i+1]](i=0, 2, \dots, MAX\_LENGTH-1)$ .
5. The function of  $MSD\_Add\_MB$  is constructed to simulate  $M+B$  adder. In terms of an MSD digit and a conventional binary digit, this function will correspondingly call the subfunction  $C\_trans$ ,  $P\_trans$  and  $R\_trans$  in (4) successively in terms of each element of the operand, achieving  $C$  conversion,  $P$  conversion,  $R$  conversion and the  $M+B$  add operation.
6. The function  $MSD\_Sub\_MB$  is constructed to simulate  $M+B$  subtractor. In terms of the subtraction involving a conventional binary digit and an MSD digit, this function will firstly use bit-reversed method and then call  $MSD\_Add\_MB$ .
7. The subfunction  $SRT\_Div$  is constructed to simulate the iterative division.
  - 7.1. The sign of quotient is determined, and the absolute value of operands  $N$  and  $D$  is calculated.
  - 7.2. The shift-functions  $ShiftRight$  and  $ShiftLeft$  are constructed. Through the left or right shift, the dividend and divisor will meet the conditions of normalization. The number of the shifted bits will be recorded in the variables  $Noffset$  and  $Doffset$ .
  - 7.3. Iterative initialization. It is needed to initialize the loop control variable  $I$ , and give  $N$  to the one-dimensional arrays  $W$ .
  - 7.4. The value of the counterparts in one-dimensional arrays  $W$  and  $Half$  is taken as the input parameters, then the  $MSD\_Add\_MB$  and  $MSD\_Sub\_MB$  are called, and one-dimensional integer arrays  $W1$  and  $W2$  are used to record the calculating results. Then the value of counterparts in one-dimensional integer arrays  $W$  and  $D$  should be taken as the input parameters, then we call  $MSD\_Add\_MB$  and  $MSD\_Sub\_MB$ , and use one-dimensional integer array  $W3$  and  $W4$  to record the calculating results.
  - 7.5. The most significant digits of  $W1$  and  $W2$  are scanned, and the variables  $W1Value$  and  $W2Value$  are recorded. Then, the following judgment statement is executed, and the value of quotient  $Q$  and the intermediate variable  $Q$  of the partial remainder are determined:
 

When  $W1Value \leq 0$ , give  $-1$  to  $Q$ , give the value of  $W3$  to  $U$ ;

When  $W1Value > 0$  and  $W2Value > 0$ , give  $0$  to  $Q$ , assign  $U$  to  $W$ ;

When  $W2Value \leq 0$ , assign  $1$  to  $Q$ , and each digit of  $W4$  takes the bit-reversed method, then assign the value to  $U$ .

The array U, as the input parameter, calls the subfunction ShiftLeft and bits once to the left. Then, the array U is assigned to W, as the input of the next iteration.

- 7.6. Repeat (7-4) and (7-5) for F times. The accurate form of the quotient is adjusted according to sign, Noffset and Doffset.
8. The function Result is constructed. The results are restored in one-dimensional integer array Q.
9. The SRT\_Div and Result are called repeatedly to calculate multiple original data one by one.
10. The input N, D and the result Q are displayed on screen; the simulation is used to return the results to the user.

## 6.2. Experimental results

The variation range of original data input length is set as an integer interval [10, 80], and the variation range of the significant digit of the quotient is set as an integer interval. Then the Simulating experiment is repeated for 500 times, and 20 groups of calculations are processed each time. The output will be compared with the theoretical value, which is used to determine whether the computational accuracy of the result is in line with the user's requirement. Figure 1 shows 6 groups of calculation in a random selection of simulation experiment, and its internal parameters are set as follows: m=40, G=19, n=40, F=65, V=169, VT=2032. Seen from Table 1, it can be concludes that the all the number of significant digits of the results are in line with the theoretical value according to the requirements of the user when using the ternary optical computer with the floating-point division routine. The 6 groups of the simulation experiment of floating-point division in Table 2 are transformed into decimal system, and sent to electronic computers for comparison, then the original data and calculating results are shown in Table 3. It can be seen from Table 3 that there are only 15 significant digits are represented and calculated accurately when defining the data by using double precision floating-point division of the electronic computer. Parts of the digits of 16 or 17 bits are accurate, and the digits of more than 17 bits will be shown as 0. All division calculations are calculated in accordance with the predetermined accuracy.

According to the simulation experiment and the parallel calculations, it can be concluded that the quotient obtained from electronic computer and ternary optical computer are in different forms. This difference originates from the different calculating ability in different type of computers. The electronic computer use a finite value for efficient representation and processing the width of data, only relying on the ability of hardware instead of the software. Thus, it has its weakness in dealing with the division operations. However, in the floating-point division routine of the ternary optical computer, the calculation is completed by allocating digits of the optical processor based on bits according to user's requirement. The output will be more closed to the theoretical value in terms of the computational accuracy. Ultimately, ternary optical computers' ability may exceed that of electronic computers.

Example	MSD number	Decimal system	Result
No.1	Dividend N 1.ī010010ī100ī01101ī011ī000ī0 010ī01ī001	0.638353880860449862666428 08914185	
	Divisor D 0.1101000100110001100101100 011001000100101	0.817162883035962295252829 79011536	
	Quotient Q 1.0ī00100000000ī00ī0100 0ī0100100000001000010 000000ī0000000100ī1ī01	0.7811831571311795904	Correct
	Theoretical value 0.7811831571311795904503.....		
No.2	Dividend N 0.īī001ī0011ī00ī0011ī0001ī0010 0ī00110ī00ī	-0.731985025887297524604946 37489319	
	Divisor D 0.1001011000110001100000110 010010000110011	0.586692997300815477501600 98075867	
	Quotient Q 1.0ī00000010011010010010100ī000 ī00ī000ī0ī00 001100010000100001010	-1.247645752130882127	Correct
	Theoretical value -1.247645752130882127991.....		
No.3	Dividend N 0.īī000ī01ī0001ī01ī00100ī0001 ī00011ī001ī1	-0.7635913471140156616456806 6596985	
	Divisor D 0.1001010000110001100100110 000110000110001	0.5788814453853774466551840 3053284	
	Quotient Q 1.0ī0ī000ī0ī00011000ī000001000 01000ī000ī00 0ī0ī000ī0ī00ī0001	-1.319080708495798662	Correct
	Theoretical value -1.319080708495798662666.....		
No.4	Dividend N 0.0110ī011ī0011ī00011000ī100 0ī0011000ī001	0.3537931155265141569543629 8847198	
	Divisor D 0.1000011000110001001000001 01100110001101	0.5241871312146031414158642 2920227	Correct
	Quotient Q 0.10101100110010001010011001 00001000011000 10100101000000ī1ī0ī00000	0.6749366675727692136	
	Theoretical value 0.6749366675727692136655.....		
No.5	Dividend N 0.ī01001ī00100ī0001100ī0001ī ī00011ī00010ī	-0.366322006736481853295117 61665344	
	Divisor D 0.11000111000011001111000001 01010001100111	0.7775411802349481149576604 3663052	
	Quotient Q 0.ī0001000ī0ī0010000100ī0ī00 101ī01010ī000ī000 ī0000010000000ī0100100	-0.471128753110916955959	Correct
	Theoretical value -0.471128753110916955998.....		

Example	MSD number	Decimal system	Result
No.6	Dividend N 0.10 $\bar{1}011000\bar{1}001010011001100\bar{1}$ 00011100 $\bar{1}01\bar{1}$	0.335229482899194408673793 07746887	Correct
	Divisor D 0.11001000010010010110111001 11001111001011	0.782370475072639237623661 7565155	
	Quotient Q 0.100 $\bar{1}00100\bar{1}00010101000001$ 000001000 $\bar{1}0000000$ 100 $\bar{1}0101001111101010\bar{1}00$	0.4284792097606571944	
	Theoretical value 0.42847920976065719445661.....		

Table 2 – The simulation results of floating-point division routine

example	Original input data	Calculation result	Theoretical value
No.1	N 0.63835388086044986000	-0.78118315713117958000	0.7811831571311795904503.....
	D 0.81716288303596230000		
No.2	N -0.73198502588729752000	-	-
	D 0.58669299730081548000	1.24764575213088210000	1.247645752130882127991.....
No.3	N -0.76359134711401566000	-	-
	D 0.57888144538537745000	1.31908070849579870000	1.319080708495798662666.....
No.4	N 0.35379311552651416000	-0.67493666757276927000	0.6749366675727692136655.....
	D 0.52418713121460314000		
No.5	N -0.36632200673648185000	-0.47112875311091695000	-0.471128753110916955998.....
	D 0.77754118023494811000		
No.6	N 0.33522948289919441000	-0.42847920976065718000	0.42847920976065719445661.....
	D 0.78237047507263924000		

Table 3 – Comparative division calculation of the electronic computer

## 7. Conclusions

In the work, the complete operation of ternary optical computer of floating-point division routine is studied in every detail, proposing the method to improve the calculation speed and reduce the hardware costs. Through the simulation experiments, the availability and high efficiency of this division routine is confirmed. It is proved that this routine can process big value data and high precision computations. The research has accumulated experience of standard and general design method for establishing the calculating routine of ternary optical computer. It also lays the foundation for establishing more basic calculating routine, such as power function and trigonometric function. It not only contributes to enhancing the calculating ability of ternary optical computers in terms of improving the time complexity, but also demonstrates the superiority of the ternary optical computer in the field of high performance computing.

## References

Jin, Y., He, H., Lu, Y. (2003). Ternary Optical Computer principle. *Sci China*, 46, 145–150.

González, M., González, L. (2015). The co-creation as a strategy to address IT governance in an organization. *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, (14), 1–15.

Jin, Y., Wang, H., Ouyang, S. (2011). Principles, structures and implementation of reconfigurable ternary optical processors. *Sci China Ser F-Inf Sci*, 54, 2236–2246.

Jin, Y., Shen, Y., Peng, J. (2010). Principles and construction of MSD adder in ternary optical computer. *Sci China Ser F-Inf Sci*, 53, 2159–2168.

Oberman, S.F., Flynn, M. (1997). Division algorithms and implementations. *IEEE Transactions on Computers*, 48, 833–854

Russinoff, D. (2013). Computation and formal verification of SRT quotient and square root digit selection tables. *Computers IEEE Transactions on*, 62(5), 900–913.

Shen, Y., Jin, Y., Peng, J. (2010). Simulation implementation of the computational principle of MSD adder for ternary optical computer. *High Performance Computing Technology*, 6, 5–10.

Shen, Y., Jiang, B. (2014). Principle and design of ternary optical accumulator implementing M-k-B addition. *Opt. Eng*, 53(9), 1–8.

Xu, Q., Jin, Y., Shen, Y. (2016). MSD iterative division algorithm and implementation technique for ternary optical computer. *Sci China Ser F-Inf Sci*, 46(4), 539–550.

Yan, J., Jin, Y., Zuo, K. (2008). Decrease-radix design principle for carrying/ borrowing free multi-valued and application in ternary optical computer. *Sci China Ser F-Inf Sci*, 51, 1415–1426.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.