Design and Implementation of Modified Signed-Digit Adder

Junjie Peng, Rong Shen, Yi Jin, Yunfu Shen, and Sheng Luo

Abstract—How to fully apply the characteristics and advantages of light in numerical computation is an important issue that attracts many scholars. Though much research has been done in this field, how to design and realize specific applications or devices is still an issue to be solved. Based on this, we present an architecture and implementation method of modified signed-digit (MSD) optical adder from the point of applicability. In the implementation, we fully consider the different procedures of the MSD addition which including optical logical operation, results decoding, special storage area design, data feedback, control of light path, etc. Meanwhile, we also introduce pipeline mechanism which guarantees that the addition operation is an automatic and continuous process. This is a carry free adder design method which guarantees the addition has high data throughput. It is very suitable to fulfill the large-scale numerical computation. The experiment shows, the MSD adder not only has a reasonable and correct design, but also has high throughput rate, can work efficiently and steadily.

Index Terms—Modeling of computer architecture, system architectures, integration and modeling

1 Introduction

The high speed and multidimensional parallelism characteristics of light make optical computing have many advantages over electrical computing. This has attracted many researchers setting their focuses on the study of optical computer since the electronic computer was invented [1], [2].

However, most of the research on optical computer is mainly focused on the logical calculations. Little work was reported on the numerical computation field. One of the important reasons is that no optical adder with high parallelism had been designed successfully.

Although the design of adder is quite ripe and has been widely used in electronic computers, we cannot just apply the design method in electronic computer to the optical one because of the big difference between them. This encourages a lot of scholars to work on the design of adder from optical aspect. Take trinary number system (MTN) as an example, much research has been done on it, especially on mixed modified signed-digit number system. In 1990, A. K. Ghosh et al. proposed an optical model based on the modified trinary number system (MTN) [3]. They combined logic gates with orthogonal states of polarisation and used shadow casting for logical operations. Maiti et al. also did interesting research on MTN. They proposed a conversion scheme from binary number to its MTN form using all-optical material [4]. In 2008, A. K. Ghosh et al. designed the half-adder and full adder circuits to complete the opto-electronics system [5]. Cherri used Mach-Zehnder interererometer to design circuits for representing the trinary values with the negabinary modified signed-digit number and proposed the all-optical NMSD adder [6].

Also, a lot of research is based on the opticity of light. Zaghloul et al. designed a binary adder using a kind of optical polarized processor. The adder could be used for logical calculation [7]. Chattopadhyay et al. designed circuit for converting numbers between binary, ternary and quaternary logical systems under the help of a polarization converter and a polarization isolator. This kind of circuit will be of great importance for applications in a future multivalued processing system [8]. A. K. Ghosh et al. also put forward their solution to convert three-valued logic to multivalued logic using Savart plate and spatial light modulator (SLM) [9]. In 2009, Chattopadhyay et al. introduced a new conception: Galios field (GF) algebraic expressions for multivalued logic. They developed an adder with four valued logic circuits in all-optical domain [10]. And they used non-linear material-based interferometric switches to design and implement the converter which can transform quaternary-signed digit number to its binary counterpart [11].

Different the study mentioned above, some researchers also used different kinds of light amplifiers to work on adders. Kim et al. designed an optical half adder with semiconductor light intensity amplifier [12]. Iftekharuddin et al. implemented a signed-digit adder using electronically addressable spatial light modulator [13]. Sun et al. from Changchun Institute of Optics put forward a space position based logic encoding strategy based on optoelectronic interconnect network. Directed by the strategy, they built a parallel MSD adder and subtracter [14]. Scaffardi et al. realized a full binary addition based on semiconductor optical amplifier [15]. Han et al. did some experimental study on all-optical half-adder based on semiconductor optical amplifier [16]. Qian et al. from Shanghai Institute of Optics proposed digit-restricted MSD algorithm based on symbol substitution concept [17]. Li et al. presented parallel optical negabinary signed-digit computing algorithm and the optical implementation method [18].

• *The authors are with Shanghai University, Shanghai 200444, China. E-mail: jjie.peng@shu.edu.cn.*

Manuscript received 29 July 2012; revised 06 Nov. 2012; accepted 20 Nov. 2012; published online 9 Dec. 2012; date of current version 29 Apr. 2014. Recommended for acceptance by C. Bolchini. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TC.2012.285

To numerical computation of ternary optical computer (TOC), our team has also done a lot of research from many aspects. TOC is an experimental platform system our team made and is used for researching in optical fields. Wang et al. studied carry-free vector matrix multiplication based on the ternary optical platform we designed [19]. Their work and experiment verified that the ternary optical platform was suitable for numerical computation. He et al. discussed the addition procedure and light path structure, which proved that structure of light path is feasible for addition operations [20]. But they didn't give a scheme on how to store the intermediate data in the process of logic transform and continuous addition solution. Liu et al. focused on a simple MSD addition model and verified the correctness of optical addition with experiment [21]. They discussed the logic transforms of MSD addition. However, they still failed to discuss the issues such as special data area, feedback and continuous work guarantee and so on. Jin et al. analyzed and discussed the principles and architecture concept of MSD adder in TOC. Specifically, they were mainly focused on the basic principle on how to process data with huge number of bits with MSD addition [22]. That is, the data can be combined or cut in the process of MSD addition.

All these research is the theoretical basis and foundation for design and implementation of MSD optical adder, they have demonstrated the feasibility from different aspects of implementation of the optical adder on the optical logic platform. However, they did not realize the adder. This is what this article comes to. From the aspect of practical use, we worked out how to design and implement a MSD optical adder that can continuously work. Also, how the data of addition should be stored and reused is taken into consideration.

Meanwhile, Song et al. proposed a kind of one-step MSD adder using the symmetric encoding form of MSD numbers [23]. However, the adder needs symmetric MSD numbers as input. This means that the augend and addend should be transformed into the symmetric MSD format from other format before the addition process starts. Unfortunately, in this work, the authors didn't discuss the implementation issue of the data format transformation. If the transformation is a serial process, the one-step method will have no advantages over other MSD method.

The rest of this article is arranged as below. Section 2 describes related works of our MSD optical adder. Section 3 presents the whole architecture and design of optical adder based on MSD algorithm. Section 4 shows the specific experiment and analysis. Section 5 makes the conclusion remarks.

2 Related Works

2.1 Modified Signed-Digit Number System and Its Addition

MSD number was first proposed by Avizienis in 1961 [24]. And Bocker et al. provided the MSD number system based on MSD number and first applied it to optical computing in 1986 [25]. In the MSD number system, any number x can be represented by Eq. (1):

$$x = \sum_i x_i 2^i, \quad x_i \in \{\bar{1}, 0, 1\}, \tag{1}$$

TABLE 1
Truth Tables for T, W, T′, W′ Transformations

a	b	T	W	T′	W′
$\bar{1}$	$\bar{1}$	$\bar{1}$	0	$\bar{1}$	0
0	0	0	0	0	0
1	1	1	0	1	0
$\bar{1}$	0	$\bar{1}$	1	0	$\bar{1}$
0	1	1	$\bar{1}$	0	1
1	$\bar{1}$	0	0	0	0
$\bar{1}$	1	0	0	0	0
0	$\bar{1}$	$\bar{1}$	1	0	$\bar{1}$
1	0	1	$\bar{1}$	0	1

where $\bar{1}$ denotes -1. Different from other number system, in MSD, a number (except 0) has more than one forms of representations. For example,

$$(4)_{10} = (100)_{2\,\text{or}\,\text{MSD}} = (1\bar{1}00)_{\text{MSD}} = (1\bar{1}\bar{1}00)_{\text{MSD}},$$

and

$$(-4)_{10} = (\bar{1}100)_{\text{MSD}} = (\bar{1}1100)_{\text{MSD}}.$$

From the example, it can be seen that the MSD number system has the following characteristics:

- The opposite number of any MSD number can be obtained by applying NOT operation on each bit of number.
- If the highest bit is positive, then the MSD number is positive, vice versa. So when presenting a negative number, we need not an extra sign bit.
- MSD is a number system with redundancy.
- MSD treats positive and negative numbers as the same while its addition and subtraction are carry-free. This means that a number in this number system can be expressed with different forms.

Considering the real application requirement of the TOC, we use MSD number system as the foundation when discussing the design of MSD adder.

MSD addition is a carry-free addition based on MSD number system with five steps of operations. Each operation is a logical transformation. These transformations are T, W, T′, W′ and T2. Step 1 and 5 are the same logical operation. And these logical operations follow the truth table as shown in Table 1 [19].

Because transformation T and W are independent, they can be executed in parallel. For the same reason, transformation T′ and W′ can also processed in parallel. In Fig. 1, we apply the parallelism of the logical transformations and five logical calculations can be implemented and simplified into three steps. The detailed logical operations are described in Section 3.4.2. For more details about principles of MSD adder, we recommend the interested readers refer to the literature [22], [26].

2.2 Framework of Ternary Optical Computer

From system aspect, the architecture of TOC can be divided into light source, encoder, processor, decoder and monitor system, as shown in Fig. 2.

Encoder mainly answers for encoding natural light from source into three-valued optical signals which serve as the input of the system. Processor focuses on execution optical

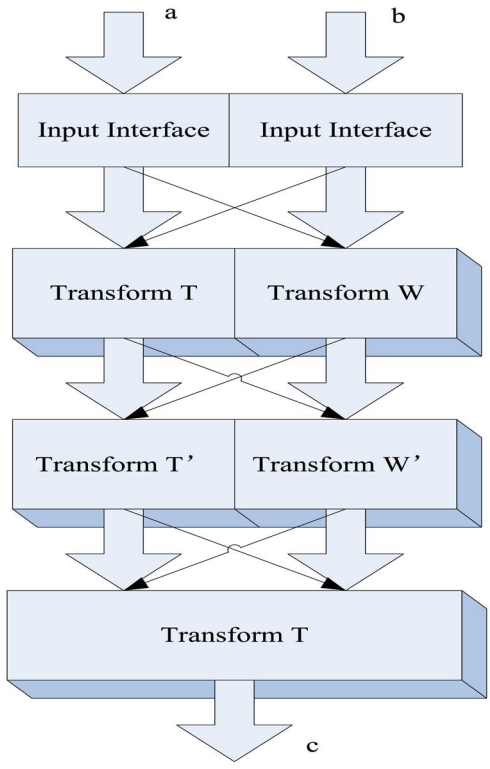

Fig. 1. Procedures of the MSD addition.

operations. The structure of encoder and processor is shown as Fig. 4 and they are kind of LCD-polarizer structure. These two parts are both consist of liquid crystal arrays (L1, L2, and L3) and polarizers (VP1, VP2, VP3, VP4, VP5, HP1, and HP2). Polarizers include two kinds: VP means vertical polarizer and HP means horizontal polarizer, thus the corresponding lights can pass through. Decoder consists of a series of photoreceptors which convert optical signals to electronic ones. The converted signals are more suitable for output and display on the electronic computer. The monitor system monitors the encoder, processor and decoder while doing some verifications. All of these parts are realized in the hardware control component of TOC as shown in Fig. 3. The upper part of the figure is for encoder and processor. The below part is for decoder. And the monitor system uses a bridge to connect these two parts to monitor and examine their data. Details of TOC are presented in literature [27]–[29].

The process of addition involves encoder, processor, decoder and the monitor system. The details of designing the optical adder and the relations among these modules will be discussed in Sections 3 and 4.

Functionally, the processor of the TOC consists of two parts: one is the main light path (MLP) and the other is control light path (CLP). Both of these two parts are consist of liquid

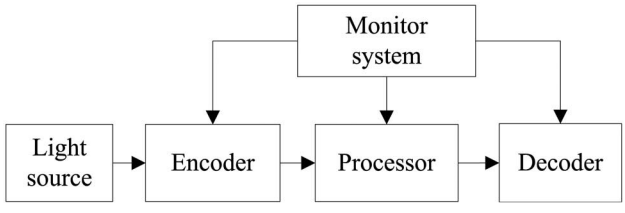

Fig. 2. Architecture of ternary optical computer.

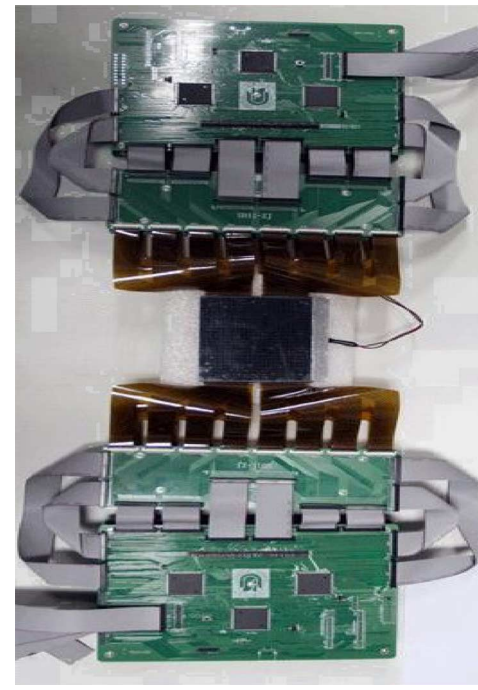

Fig. 3. The hardware control component of TOC.

crystal array, but they have different polarizers on them. Liquid crystal array of the main light path is divided into four areas (HH, HV, VH, VV), and that of control one is divided into two areas (H, V), as shown in Fig. 5.

3 Design of Modified Signed-Digit Optical Adder

3.1 System Framework and Structure

Considering the working process, the MSD optical adder consists of seven modules as it shows in Fig. 6.

User interaction module is used for receiving input from users which serves as the operands of the system. This module is implemented and controlled with PC which can fully exert the advantages of electronic computer.

Encoding control module collects operands and decoding feedback information from the main light path, encodes and sends them to two operation modules as input. For the details about encoding, please refer to literature [29]. Operation control module transfers the control signals to the main light path operation module while receiving data from control light path decoding feedback module. In order to use the characteristic of parallelism of optical computation, we chose FPGA as controller to realize encoding control module and operation control module in our design.

The main light path operation module tries to execute operations under given control signals under the help from the control light path operation module. Of course, all the operations are logical ones. Physically, both the main light path operation module and the control light path operation module have the same structure, consisting of liquid crystal and polarizer. The main light path decoding feedback module and control light path decoding feedback module answer for decoding the corresponding results from two different paths and send them to the encoding control module and operation control module. When the addition operation is over, the

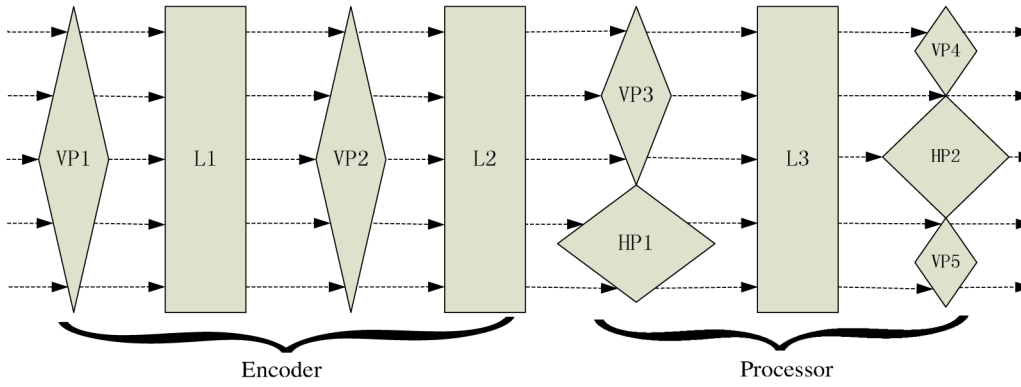

Fig. 4. Encoder and processor structure.

main light path decoding feedback module also sends the final results out as output to the user interaction module. All feedback modules are implemented on ARM9.

The addition process works as the following steps and it is realized by LCD-polarizer structure. First, get the operands from encoder and send them to the main light path and control light path as input. The calculation result is decoded and sent to a special data storage area where the data will be calculated and feedback to the main light path and control light path. This process will last for five times until all the logic transformation T, W, T′, W′ and T2 have finished. The final result obtained after transformation T2 is the result of addition. We use light to carry the information of operands and the result.

As the addition needs three steps even with execution in parallel, continuity is very important. That is the immediate results obtained from the last step should be used as input of the next step. To guarantee the addition process work smoothly, we design a special data area. This area is used to store the information of encoding control module, the decoded immediate data, and organizing feedback data which can guarantee the feedback information can be drawn automatically in the whole process of addition. In addition, it also needs a proper and correct mapping method between feedback area and encoding area to guarantee data can be transferred consistently and correctly.

3.2 Design of Special Data Storage Area

Special data storage area is of great importance for continuous and efficient work of MSD adder. To clearly explain the design, we first present the design of the special encoding data storage area, then discuss about the intermediate results transition in Section 3.3, and finally give the description of the design of the feedback storage area in Section 3.4.

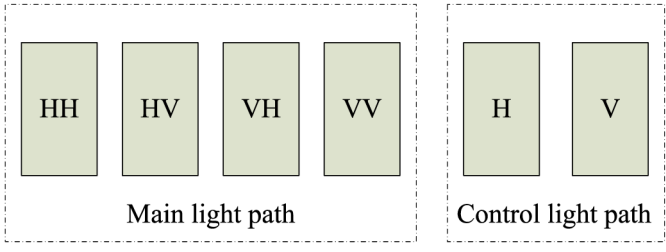

Fig. 5. Optical processor light path division.

In our TOC, ARM9 is used to receive encoding data and handle feedback data and Friendly ARM Mini 2440 is chosen as the development tool. It has two kinds of startup ways: NOR startup and NAND startup. What we use is NAND startup.

We have used two 64 MB sdrams and joined them together as 128 MB data storage area. The addresses of the two sdrams start from 0x30000000 and 0x38000000 respectively. Take the sdram on *nGCS6* as an example, the allocation scheme of 64 MB is shown as Table 2.

To guarantee the exactness of operations, in the design, download area is not used, but we start allocating the address from 0x31000000.

3.2.1 Main Light Path Encoding Data Storage Area

Now the LCD of the processor has 576 pixels, that is, it can process 576 bits data once. Considering the characteristic of light, the number of the data-bit that can be processed each time will be increased in the future. In the design, we have fully considered the future extension and left five address lines for this purpose. This indicates that the maximum number of data-bit the future system can process will be 32 times bigger than that of today's. With the structure we put forward, the system can be used in the future without any change for the special data storage area.

To avoid the extra calculation of addresses, we separate the four areas to extend. To implement this, we allocate the 32 times space for each area, and then divide them respectively. The addresses are continuous and linear. This guarantees the

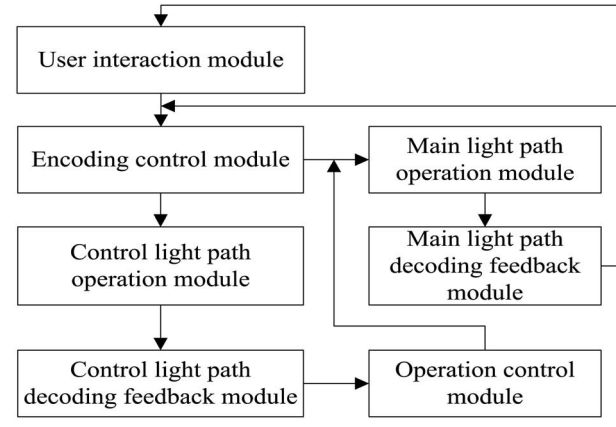

Fig. 6. MSD optical adder modules flowchart.

TABLE 2
Default Storage of 64 MB ARM Sdram

Start address	End address	Length	Function
0x30000000	0x30ffffff	16MB	Download area, cacheable
0x31000000	0x33feffff	47MB	Non-cacheable area
0x33ff0000	0x33ff47ff	18KB	Heap & rw area
0x33ff4800	0x33ff7fff	14KB	User stack area
0x33ff8000	0x33fffeff	31KB	Not used area
0x33ffff00	0x33ffffff	256byte	Exception & ISR vector table

minimum cost is spent for addressing. Fig. 7 shows the address division strategy.

Because a pixel relates to 2 bytes data, each area has $576/8 \times 2 = 144$ bytes (one time). We notice that one LCD of the processor has 576 pixels, however only 512 pixels are used in the real applications. With the extension, the largest size is $24 \times 24 \times 32/4 = 4608$ bytes (32 times). And all the extended areas are allocated with the same manner. Following we will only discuss one area with 144 bytes as an example.

Focusing on one area and not considering the extension, we divide the optical processor into five small partitions: T, W, T′, W′, T2, so-called calculation subareas. In each subarea, the number of LCD point must be times of 8. Since $144/5 = 28$, we choose 24 bytes in each small partition as working area. And leave $144 - 24 \times 5 = 24$ bytes for non-redundant area (8 bytes) and redundant area (16 bytes). The redundant area is also divided into five calculation areas, and left 3 bytes.

Fig. 7 shows the extension of the calculation area where we use different sizes for the these areas. Assume we enlarge N times, each areas are then X bytes, $1 < N \leq 32$. It can infer that $X = 24 \times 24 \times N/4$ bytes (N times). Each calculation area is of size $Y = X/5 = 24 \times 24 \times N/4/5$ bytes, and Y should be times of 8.

Take HH area as an example. Each calculation area is of size $Y = 24 \times 24 \times N/4/5$ bytes, and finally left pixels' length is $X - Y \times 5$ bytes among which the non-redundant area contains $X - Y \times 5 - (X - Y \times 5 - 1)/8 \times 8$ bytes, the redundant area contains $(X - Y \times 5 - 1)/8 \times 8$ bytes.

Actually, addresses calculation is done at the initialization. And there is only little cost for the calculation.

3.2.2 Control Light Path Encoding Data Storage Area

Control path is implemented with another mini 2440 embedded development board. Fig. 8 shows the storage and the main difference between control and main light path is that control light path only has data of two areas: H, V.

3.3 Intermediate Results Transition

TOC uses horizontal, vertical polarized light and dark state (H, V and W), three states of light, to express the ternary information, recorded as 1, -1, 0. Special data storage area contains these ternary data. To fully apply advantages of electronic computer, ternary information should be expressed with two values. This is the reason why we need to obtain the transition intermediate results.

Decoder of MSD optical adder is aimed at decoding each area's results based on the states of liquid crystal and polarizer. Two bits are needed for representing one bit ternary information. We have two different ways to detect the intermediate results and transit them as shown in Fig. 9. S represents the source light, WP is the abbreviation of Wollaston prism, D, D1, D2 are the photodetectors, E is a kind of electrooptic modulator, such as LCD, and A is analyzer, such as horizontal polarizer. In (a), we use WP to seperate light into a pair of orthogonal linearly polarized light (V and H lights). If D1 and D2 output no signals, V and H lights should be dark state, thus S is dark state and it means it's 0 in ternary. If D1 outputs signal while D2 does not, it means S should be vertical polarized light, that is -1 in ternary and vice versa. However, (b) shows another scheme of transition. Assuming that A is

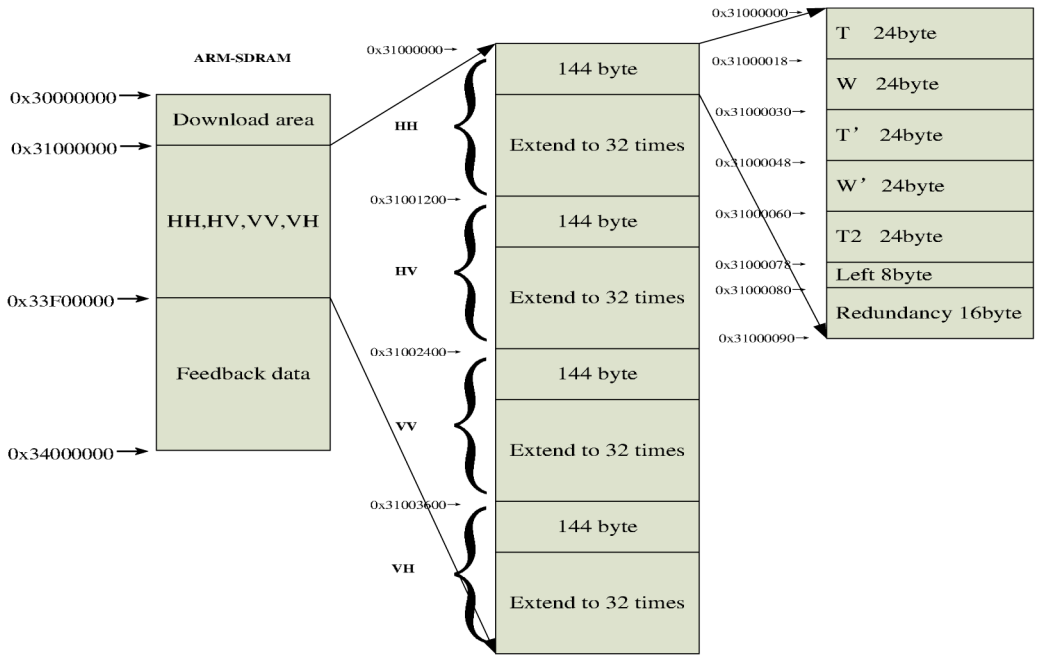

Fig. 7. Main light path encoding data storage.

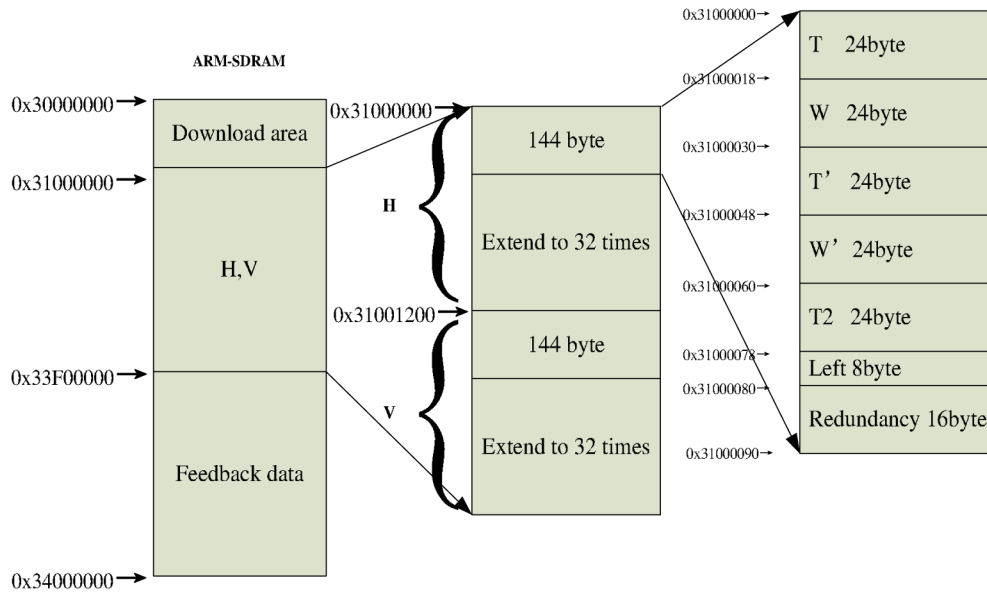

Fig. 8. Control light path encoding data storage.

vertical polarizer, if we add control signal to E in the latter half of signal cycle and make it turn light polarized direction for 90 degree, then we observe the result. If D always outputs no signal in one signal cycle, it means we receive 0 in ternary. But if we have output in the first half of cycle, we get vertical polarized light while having output in the latter one indicates it's horizonal polarized light.

Among our four areas, HH area and VH area output H light or W light, while VV area and VH area output V light or W light. It is the decoder that converts optical signals into electrical ones. Table 3 shows the relationship of the main and control light path decoding signals and H light represents horizontal polarized light while V light represents vertical polarized light.

From the last two columns, it can easily find out that the calculation formula of decoding follows this rule: $first\ bit = HH|HV|VV|VH,\ \ second\ bit = HV|VV$. For more details about decoding theory and technology, we would like to recommend interested readers look up literature [28] for reference.

3.4 Design of Feedback Control

Feedback strategy is the key of MSD optical adder. In order to realize carry-free calculation, it is a need to send back the intermediate data to encoding control module and operation control module as the input of next logical operation. The following is about the design of the feedback data storage area and feedback control strategy.

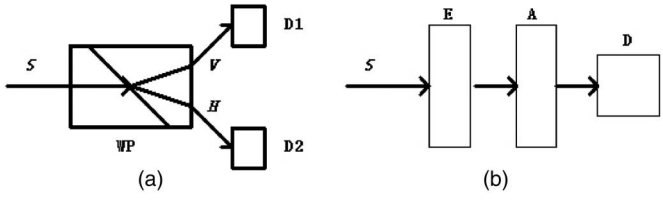

Fig. 9. Two different structure of transition.

3.4.1 Feedback Data Storage Area

Feedback control module of adder has two parts: the main light path feedback module and control light path feedback module when considering data storage.

Main light path feedback: Among the feedback data, one LCD point relates to 2 bytes data, so when we use a LCD with 512 valid pixels, each subarea contains $512/4/5 = 24$ bytes and can extend to 32 times. The address begins from 0x33F00000 as Fig. 10.

The basic principle of division is that we keep corresponding extension space based on the number of extensible address lines, so that the start addresses of each subarea do not change, which will be convenient for later use.

Calculation described in Section 3.3 together with data feedback is very important for the system.

Consider the efficiency of the calculation and transfer, we use subarea as a unit and DMA as data transmission channel. This can much increase the data processing efficiency and highly reduce the times of opening and closing DMA channels. So, we allocate another subarea as the buffer of transfer. Particularly, its size is 24 bytes.

Control light path feedback: Control light path feedback data is the same as that of the main light path, but it only contains feedback data from H, V areas as shown in Fig. 11.

3.4.2 Feedback Control Strategy

According to the rules of operations in MSD digital system, it needs three steps which include five logical operations to complete a carry-free addition. Among these steps except last one, we take the indirect results as the new input of the next

TABLE 3
Relationship of Decoding Signals

Logical symbol	Physical form	Decoding information	
		Main path	Control path
0	No light	0	0
1	H light	1	0
-1	V light	1	1

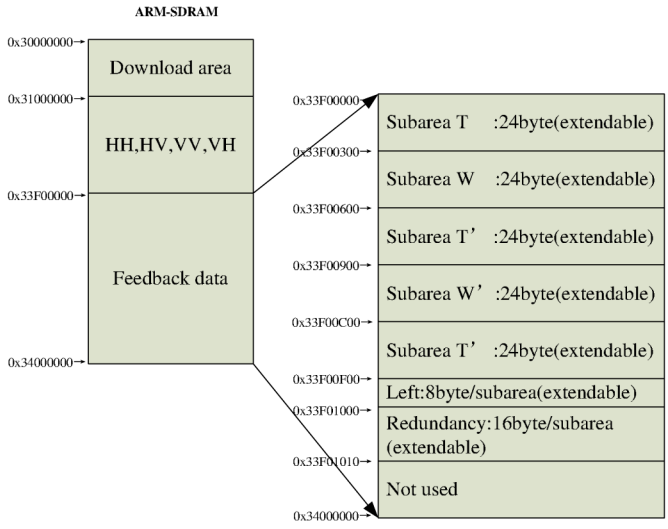

Fig. 10. Main light path feedback data area.

step. That is a feedback strategy, and when we go to the last step, we only send the final results to user interaction module.

Feedback control is carried out as follows and more details please refer to the literature [22]:

- Get operands a, b from encoding control and operation control module.
- Apply T, W to a, b bit by bit. The result of T is u and the result of W is w.
- Append one 0 to the tail of u, and this is t.
- Apply T′, W′ to t, w bit by bit. The result of T′ is u' and the result of W′ is w'.
- Append one 0 to the tail of u', and this is t'.
- Apply T2 to t', w' bit by bit. The result is the final result of the addition.

The adder adopts the line stream structure to the operations, please see Fig. 12. Due to the pipelining process, we are able to let operands A and B do their T, W transformation first. And then operand A and B go to T′, W′ transformation while we can get operand C and D from input and apply T, W transformation on them. In the next step, T2 transformation is used for operand A and B while operand C and D is working on T′, W′ transformation. Finally, only one cycle is need for

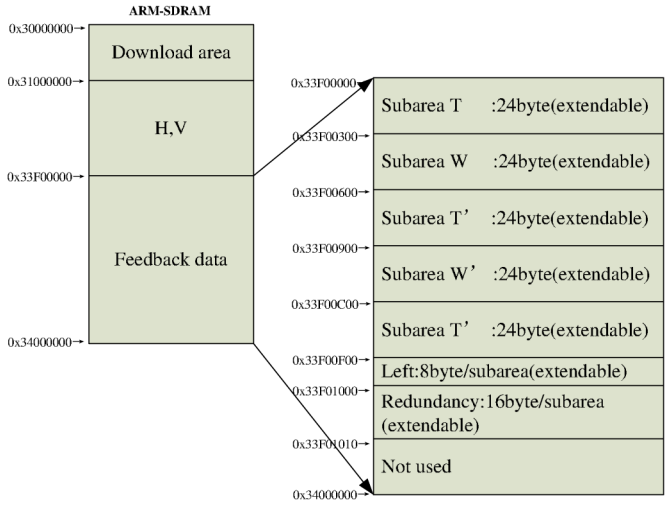

Fig. 11. Control light path feedback data area.

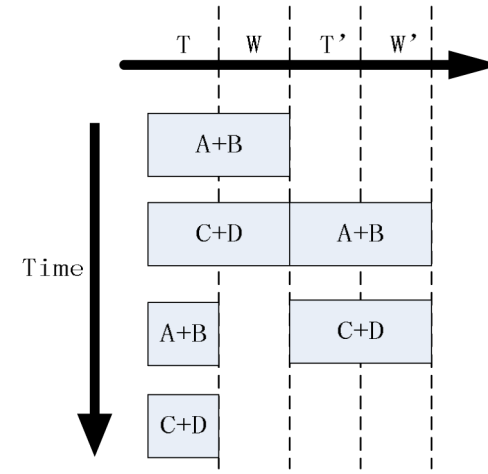

Fig. 12. Continuous working of addition.

two pairs of operands to add. In this procedure, ARM is responsible for delivering feedback data to FPGA, and FPGA realizes the distribution and implantation accordingly through the hardware logical circuit. Mapping between the special data storage area of ARM and data storage of FPGA will be shown in Section 4.2.

4 Experiment and Analysis

Based on hardware structure and storage design of MSD optical adder we discussed above, we have done experiments to verify the validity, effectiveness and reliability of data transfer. First, we focus on reading in encoding data and outputting feedback data, which provides the function of large-scale data transfer. Next, we present the adding results on stroke segment LCD, and this is the essential part to guarantee correctness.

4.1 Verification of Data Transfer

To transfer encoding data to MSD special data store area correctly is the key to our addition process. Considering MSD optical adders outputs are parallel and multi-data bits, we choose DMA to copy data by blocks, so that the data transmission will be high efficiency.

It mainly has these three steps below:

- Assign DMA interrupt handling function, set registers for both the source and destination, the transfer times, transfer mode and other basic parameters.
- Send DMA transfer request signal to CPU, wait for CPU to release the bus, then DMA uses bus to control data transfer between two storage areas.
- After the transfer, end the interrupt request signal.

Among these, core function is *void* DMA_move (int *ch*, int *srcaddr*, int *dstaddr*, int *tc*, int *dsz*, int *tsz*). *Ch* shows passageway number of DMA, *srcaddr* and *dstaddr* represent source address and destination address, *tc* is transfer times, *dsz* shows types of transfer ($dsz = 0$: byte, $dsz = 1$: half word, $dsz = 2$: word, $dsz = 3$: no assigned), *tsz* shows how much data is transferred each time($tsz = 0$: once a time, $tsz = 1$: four a time).

For example, we execute DMA_move (0, 0x31000000, 0x31000000 + 0x800000, 0x800000, 0, 0), that is to say, we use channel DMA0, copy data from 0x31000000H to 0x31800000H

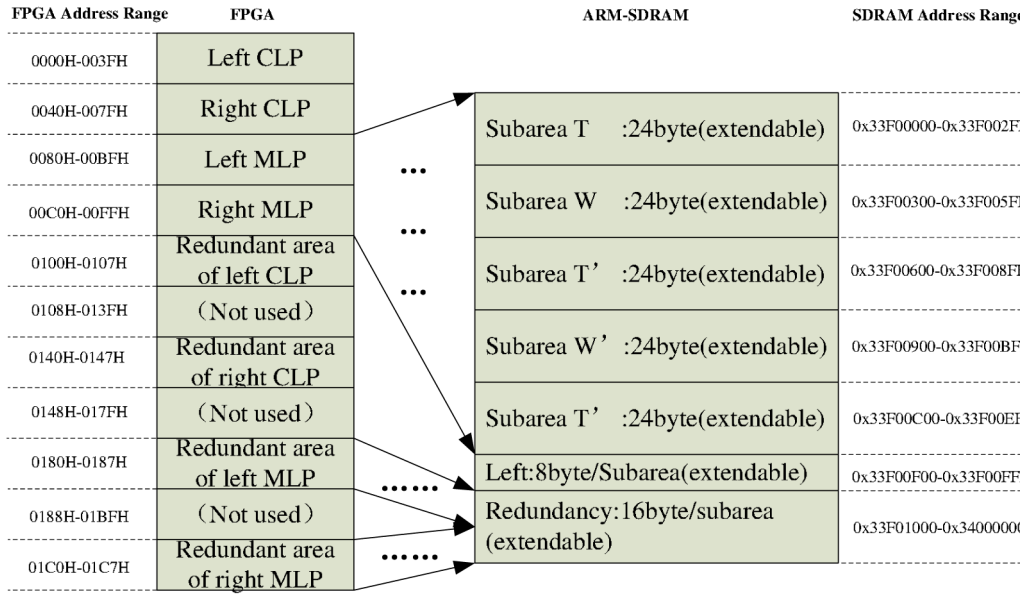

Fig. 13. Mapping from FPGA address to ARM data storage area.

(length is 80000H). Transfer is byte by byte and each time one data type is transferred, totally 80000H times. After transfer, we simply add all the data in source address and in destination address, then compare them to make sure if DMA transfer is correct. If two results don't match, it's failed. After running the code, we have results as below:

Result:

$$\begin{aligned}&\text{DMA0 31000000H} \rightarrow \text{31800000H},\ size = 80000H\\&(tc = 80000H),\ dsz = 0,\ burst = 0\\&\text{DMA transfer done},\ time = 154MS\\&sum0 = fffe0000,\ sum1 = fffe0000\\&\text{DMA transfer OK}\end{aligned}$$

So, we can see clearly that data transfer is valid. For more accuracy, Section 4.2 describes how we use stroke segment LCD to do verification.

4.2 Verification of Addition Process

In order to guarantee our MSD optical adder is well designed and works well, we first need to make sure the feedback strategy is sound. The mapping relationship between ARM and FPGA is shown in Fig. 13. Among all the bits of our multiple bits addition, three continuous bits are chosen for verification. Their addresses on ARM are from 0x33F00111 to 0x33F00113. After mapping to FPGA, addresses are from 191H to 193H.

Stroke segment LCD is chosen for experimental use for three reasons. First, the LCD can change lights' penetrating status by 90 degree if added signals on it, therefore fitting our method of using horizontal and vertical lights. Second, the control of LCD is static, it means that we can use seperate segments for different light path areas easily. Finally, stroke segment LCD has advantages in observing and tracing the experimental data conveniently for human eyes. So, considering these factors, our LCD doesn't have very quick switching speed or real-time reponse against other kinds of LCDs. However, the MSD optical adder does not choose LCD in implementation. For experiment, stroke segment LCD we used is quite cheap and suitable. In real applications, any high performance LCDs can be selected thus producing even better results of our adder.

Here we take two examples to verify the MSD optical adder. First, operand A of these three bits is W, H, W lights,

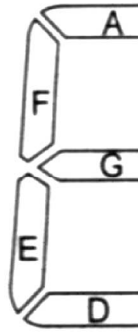

Fig. 14. LCD structure.

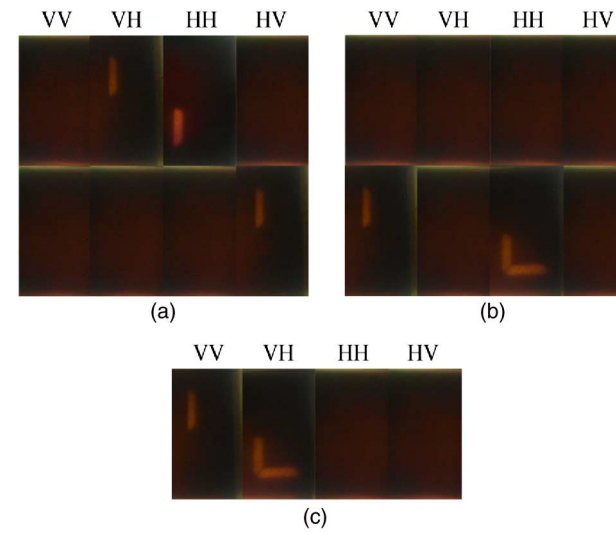

Fig. 15. Outputs of WHW lights adding WHH lights.

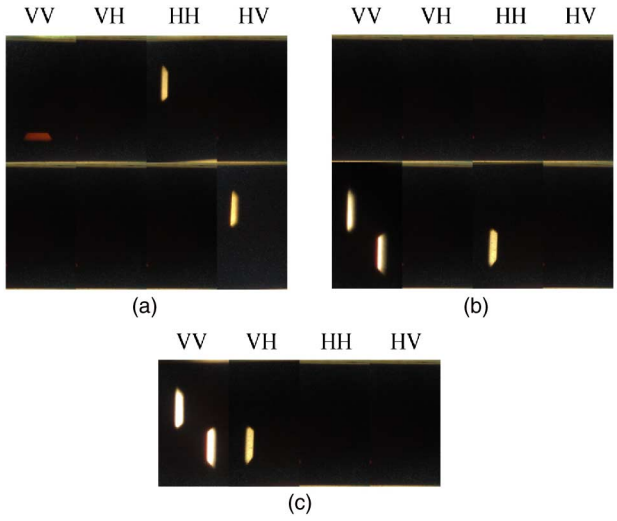

Fig. 16. Outputs of WVWH lights adding WVWW lights.

representing 0, 1, 0 in ternary. And operand B is W, H, H lights, representing 0, 1, 1 in ternary. Each bit is output as a bit of stroke segment LCD. Fig. 14 shows the LCD structure. Among these segments, D, E, F are used for displaying three bits accordingly.

Fig. 15 shows the results of WHW lights adding WHH lights. Each bit lightened shows light penetrates this area. In the figure, one stripe including four areas is an intermediate result shown as (a) and (b), while (c) is the final answer. Each intermediate result except c is needed as the input of next step. Finally, lights penetrate VV, VH area, that is, segment D, E has H light and segment F has V light. So, these three bits output H, H, V lights, representing 1, 1, -1 in ternary. Obviously, $(0\times 4+1\times 2+0\times 1)+(0\times 4+1\times 2+1\times 1)=5$ equals $1\times 4+1\times 2-1\times 1=5$.

The second example shows that the experiment can be repeated and now we use four bits to do verification. Take W, V, W, H lights as operand A, representing 0, -1, 0, 1 in ternary. And operand B is W, V, W, W lights, representing 0, -1, 0, 0 in ternary. Meanwhile segment C is used for representing the highest bit. From Fig. 16, it can be easily recognized that lights penetrate VV, VH area, and segment C, E, F has V, H, V lights. The result is V, W, H, V lights, representing -1, 0, 1, -1 in ternary. Because $(0\times 8-1\times 4+0\times 2+1\times 1)+(0\times 8-1\times 4+0\times 2+0\times 1)=-7$ equals $-1\times 8+0\times 4+1\times 2-1\times 1=-7$. Therefore, these two examples prove that our adder can give out the correct answer of addition, demonstrate the correctness of the design and implementation.

Through the experiments mentioned above, we prove the validity of continuous and parallel handling capacity and effectiveness of data feedback strategy. We also use actual data test on stroke segment LCD. These results show that the design of the structure of the adder is feasible and correct. It can be used to MSD optical adder in the TOC which is very important for the numerical applications.

5 Conclusions

On the basis of optical calculation feature and principles of redundancy addition, this article proposed an overall solution of TOC's MSD adder and its hardware structure. We take full consideration on some key components of MSD optical adder, such as special data store area, data feedback part, light path control system, and combine these components with embedded system to realize the adder which can automatically completes data feedback and keep continuous usage of addition. Finally, the experiments show our adder is correct and reasonable. Furthermore, they illustrate the feasibility of MSD optical adder's architecture and lay a solid foundation for future optical computer applying numerical computation.

Acknowledgments

This work was supported by National Natural Science Foundation of China (61103054, 61073049), Innovation Program of Shanghai Municipal Education Commission under Grant 11YZ09 and Shanghai Leading Academic Discipline Project J50103.

References

[1] Y. M. Li, L. X. Chen, and W. L. She, "Theoretical and experimental investigation on all-optical logic gates based on photoisomerization," *Acta Phys. Sin.*, vol. 56, pp. 5895–5902, 2007.
[2] Q. F. Xu and L. Mical, "All-optical logic based on silicon micro-ring resonators," *Opt. Lett.*, vol. 15, pp. 924–929, 2007.
[3] A. K. Ghosh and A. Basuray, "Trinary optical logic processors using shadow casting with polarized light," *Opt. Commun.*, vol. 79, pp. 11–14, 1990.
[4] A. K. Maiti, J. N. Roy, and S. Mukhopadhyay, "All-optical conversion scheme from binary to its MTN form with the help of nonlinear material based tree-net architecture," *Chin. Opt. Lett.*, vol. 5, pp. 480–483, 2007.
[5] A. K. Ghosh, P. P. Choudhury, and A. Basuray, "Modified ternary optical logic gates and their applications in optical computation," in *Proc. IEEE Innov. Adv. Tech. Syst. Comput. Sci. Softw.*, Univ. Bridgeport, 2008, pp. 87–92.
[6] A. K. Cherri, "All-optical negabinary adders using MachZehnder interferometer," *J. Opt. Laser Technol.*, vol. 43, pp. 194–203, 2011.
[7] Y. A. Zaghloul, A. R. M. Zaghloul, and A. Adibi, "Design of half and full optical binary adders using the polarization optical processor architecture," in *Proc. SPIE*, vol. 6695, 2007, pp. 669509–669509-7.
[8] T. Chattopadhyay, P. Bhowmik, and J. N. Roy, "Polarization encoded all-optical n-valued inverter," *J. Opt. Soc. Amer. B*, vol. 29, pp. 2852–2860, 2012.
[9] A. K. Ghosh and A. Basuray, "Trinary flip-flops using Savart plate and spatial light modulator for optical computation in multivalued logic," *Optoelectron. Lett.*, vol. 4, pp. 443–446, 2008.
[10] T. Chattopadhyay, C. Taraphdar, and J. N. Roy, "Quaternary Galois field adder based all-optical multivalued logic circuits," *Appl. Opt.*, vol. 48, pp. E35–E44, 2009.
[11] T. Chattopadhyay, M. K. Das, J. N. Roy, A. K. Chakraborty, and D. K. Gayen, "Conversion of binary number to its quaternary signed digit form and its all-optical implementation," *IET Circuits Devices Syst.*, vol. 5, pp. 132–142, 2011.
[12] S. H. Kim, J. H. Kim, C. W. Son, G. Kim, Y. T. Byun, Y. M. Jhon, S. Lee, D. H. Woo, and S. H. Kim, "Design and implementation of all-optical half adder using cross gain modulation in semiconductor optical amplifiers," in *Proc. SPIE, Adv. Opt. Quantum Memories Comput. IV.*, vol. 6482, 2007, pp. 64820B–64820B-8.
[13] K. M. Iftekharuddin, A. A. S. Awwal, and M. A. Salam, "Signed-digit adder using electronically addressable spatial light modulator," *Opt. Eng.*, vol. 40, pp. 2442–2445, 2001.
[14] D. Sun, N. X. Wang, L. M. He, Z. H. Weng, D. H. Wang, and R. T. Chen, "Demonstration of an optoelectronic interconnect architecture for a parallel modified signed digit adder and subtracter," *Opt. Eng.*, vol. 35, pp. 1785–1793, 1996.
[15] M. Scaffardi, P. Ghelfi, E. Lazzeri, L. Pot, and A. Bogoni, "Photonic processing for digital comparison and full addition based on semiconductor optical amplifiers," *IEEE J. Sel. Topics Quantum Electron.*, vol. 14, pp. 826–833, 2008.

[16] B. C. Han, J. L. Yu, W. R. Wang, L. T. Zhang, H. Hu, and E. Yang, "Experimental study on all-optical half-adder based on semiconductor optical amplifier," *Opt. Lett.*, vol. 5, pp. 161–164, 2009.
[17] F. Qian, and G. Q. Li, "Efficient two-step digit-set-restricted modified signed-digit algorithm based on optoelectronic shared content-addressable memory," *Opt. Eng.*, vol. 44, pp. 125201–125206, 2005.
[18] G. Q. Li, F. Qian, H. Ruan, and L. R. Liu, "Parallel optical negabinary signed-digit computing: Algorithm and optical implementation," *Opt. Eng.*, vol. 38, pp. 403–414, 1999.
[19] X. C. Wang, J. J. Peng, M. Li, Z. Y. Shen, and S. Ouyang, "Carry-free vector-matrix multiplication on a dynamically reconfigurable optical platform," *Appl. Opt.*, vol. 49, pp. 2352–2362, 2010.
[20] H. He, J. J. Peng, Y. P. Liu, X. C. Wang, and K. Song, "Research and design of a MSD adder of ternary optical computer," in *Proc. 3rd Int. Conf. Comput. Intell. Ind. Appl.*, 2010, pp. 265–268.
[21] Y. P. Liu, J. J. Peng, Y. Y. Chen, H. Hui, and H. T. Su, "A new carry-free adder model for ternary optical computer," in *Proc. 10th Int. Symp. Distrib. Comput. Appl. Business Eng. Sci. (DCABES)*, 2011, pp. 64–68.
[22] Y. Jin, Y. F. Shen, J. J. Peng, and G. T. Ding, "Principles and construction of MSD adder in ternary optical computer," *Sci. China Ser. F*, vol. 53, pp. 2159–2168, 2010.
[23] K. Song and L. P. Yan, "Design and implementation of the one-step MSD adder of optical computer," *Appl. Opt.*, vol. 51, pp. 917–926, 2012.
[24] A. Avizienis, "Signed-digit number representations for fast parallel arithmetic," *IRE Trans. Electron. Comp*, vol. EC-10, pp. 389–400, 1961.
[25] 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, pp. 2456–2457, 1986.
[26] M. Li, H. C. He, Y. Jin, and X. C. Wang, "An optical method for MSD addition," *Acta Photon Sin.*, vol. 39, pp. 1053–1057, 2010.
[27] Y. Jin, H. C. He, and Y. T. Lü, "Ternary optical computer architecture," *Phys. Scr.*, vol. 59, pp. 98–101, 2005.
[28] K. Z. Zuo, Y. Jin, T. Xue, and J. Y. Yan, "Tri-state light decoder of ternary optical computer," in *Proc. 3rd Int. Conf. Comput. Educ. Adv. Comput. Technol. New Educ. (ICCSE'08)*, 2008, pp. 442–444.
[29] H. Sun, Y. Jin, and J. Y. Yan, "Research on principle of coder and decoder in ternary optical computer by experiment," *Comput. Eng. Appl.*, vol. 40, pp. 82–83, 2004.

Junjie Peng received bachelor, master and doctor degrees from Harbin Institute of Technology, China in 1999, 2001 and 2005, respectively. He joined the faculty of Shanghai University in January 2008 where he is an associate professor with the School of Computer Engineering and Science. Prior to that, he was a research fellow with Zuse Institue Berlin (ZIB), Germany. Before the ZIB Fellowship, he was a senior researcher with Alcatel Shanghai Bell Co. Ltd. (ASB), Shanghai. He is an author of more than 60 refereed papers appeared in international journals and conference proceedings, an owner of 4 patents and 2 software copyrights, an contributor of 4 submissions to OMA and 3GPP2. Currently, he is a senior member of China Computer Federation (CCF), senior member of China Institute of Electronics (CIE), member of Association for Computing Machinery (ACM), Member of China Institute of Communication Expert Committee of Cloud Computing and SaaS, Expert of Cloud Computing Research Lab, CSIP of Ministry of Industry and Information Technnology, an associate editor of *Frontiers of Computer Science*, an editorial member for three other international journals, a Program Committee member, Session Chair and Registration Chair for more than a dozen of international conferences. He is a referee for more than 30 international journals and conferences. His research interests cover cloud computing, mass storage, embedded system, wireless sensor network, optical computing, etc.

Rong Shen received the bachelor's degree in Shanghai University, China, 2011. He is pursuing the master's degree in the School of Computer Engineering and Science, Shanghai University. His reseach interests include optical computing, ternary optical computer, embedded system, etc.

Yi Jin received the bachelor and doctor degrees in Northwestern Polytechnical University, Xi'an, China, in 1981 and 2003, respectively. He is a professor of the School of Computer Engineering and Science, Shanghai University, China. His research interests focus on optical computer and architecture.

Yunfu Shen received the master's degree and doctorate in Beijing Normal University, China. He is an associate professor with the School of Computer Engineering and Science, Shanghai University, China. His research interests focus on formalization method of software and hardware, model checking, and reliability about optical computer and so on.

Sheng Luo received the bachelor's degree in Shanghai University, China, 2011. He is pursuing the master's degree in the School of Computer Engineering and Science, Shanghai University. His reseach interests include optical computing, wireless sensor network, embedded system, etc.