

一种改进的三值光学计算机 MSD 无进位加法器的原理与实现

潘 磊 沈云付

(上海大学计算机工程与科学学院高性能计算中心 上海 200072)

摘 要 三值光学处理器是三值光学计算机中最重要的器件,人们对此进行了许多研究。目前,基于 MSD 的三值光学计算机加法器主要是将加数和被加数改写为 MSD 数,进行 T 、 W 变换,再进行 T' 、 W' 变换,然后进行 T 变换,实现无进位加法运算。针对这一种计算方法进行改进,提出了以先进行 T 、 W 变换,再进行 T' 、 W' 变换,然后进行 W' 变换的方式进行无进位加法运算的方法。理论和实验证明了此方法的可行性与正确性。通过这种方式,从某种角度看可以减少光学加法器中基元的数量,降低实现的难度,为设计光学加法器提供了的新思路。

关键词 三值光学处理器,三值光学计算机加法器,MSD

中图分类号 TP332,TP3-0 **文献标识码** A

Implementation and Principle of Improved MSD Carry-free Adder for Ternary Optical Computer

PAN Lei SHEN Yun-fu

(School of Computer Engineering and Science, High Performance Computing Center, Shanghai University, Shanghai 200072, China)

Abstract The most important device in the ternary optical computer (TOC) is ternary optical processor, and many studies in this field have been done. So far, the TOC adder based on MSD mainly rewrites the addend and summand to MSD format. After that, T , W transformations are first applied respectively, then the outputs of them are as the input of T' , W' transformations, at last T transformation is used to achieve carry-free addition operation. In this paper, the above addition algorithm was improved, and a new computing method was put forward. According to this method, for the two MSD data, first use T , W transformations, then use T' , W' transformations, at the third step use W' transformation to achieve carry-free addition operation. The feasibility and validity of this method are proved theoretically and experimentally. In this way, in some sense, it can reduce the number of basic elements in optical adder, which will decrease the design difficulty, and provide a new approach for the design of the optical adder.

Keywords Ternary optical computer (TOC), TOC adder, MSD

1 引言

加法是所有算术运算中最基本的运算之一;减法是加法的逆运算;乘法可以通过加法来实现;除法是乘法的逆运算,实际上也可以通过加法来实现。在计算机系统中,当加法运算的问题解决时,其他的算术运算也会迎刃而解,所以说加法运算器对于三值光学处理器而言至关重要,它的好坏将直接影响处理器的性能。

由于光具有空间巨并行性的特性,因此很多实现宽数据位加法的光学方法得以产生。国内方面,1994 年上海光机所刘立人等人采用“混合二进制的补码数值”表达信息,实现了光学运算中的加法无进位,减法通过加法完成^[1]。1996 年上海光机所李国强等人提出“光学负二进制”运算,只需两步即可完成光学加法运算并且无进位^[2]。2000 年南开大学物理学院光子学中心的李玉栋等人 and 复旦大学生理和生物系的李庆国等人完成了利用 bR 膜实现全光二值逻辑操作^[3]。

2000 年金翊等人提出以无光态和偏振方向相互正交的两个偏振光态(垂直偏振光和水平偏振光)表示信息、以液晶和偏振片构造光学处理器的三值光学计算机概念和结构^[4]。2005 年金翊等提出了进位直达并行加法器原理^[5]。2007 年,严军勇等人提出了降值设计理论^[6]。2010 年李梅等人提出了一种实现 MSD 加法的光学方法^[7]。考虑到进位直达并行加法器在构造上的困难,以及 MSD 数加法器无进位过程的优点,金翊等人基于降值设计理论并采用数据剪辑技术和流水计算技术建立了三值光学计算机的 MSD 加法器理论^[8],目前基于 MSD 的光学加法器的设计正在进行之中,已取得了一些成果。2010 年底金翊等提出了一种可重构的三值光学处理器,并利用小规模 FPGA 芯片实现了新设计的重构电路^[12]。

本文在已有研究的基础上,对 MSD 加法的光学方法加以改进,提出了一种新的三值光计算机 MSD 并行无进位加法原理和实现方法。

到稿日期:2011-01-27 返修日期:2011-04-21 本文受上海市重点学科建设项目(J50103)与国家自然科学基金项目(61073049)资助。

潘 磊(1988-),男,硕士生,主要研究方向为三值光学计算机,E-mail:pl885763@qq.com;沈云付(1960-),博士,副教授,主要研究方向为软硬件形式化方法、模型检查、三值光学计算机可靠性等。

2 MSD 数的加法

2.1 MSD 数

MSD(Modified Signed-Digit)数于 1961 年由 A. Avizienis 等人首次提出^[9],1986 年 B. LDrake 等人首先将其引入到光计算中来^[10],并研究了它的逻辑结构和光学实现。MSD 表示对应的数字系统由 $\{0, 1, \bar{1}\}$ 组成,是一种冗余的不带符号位的二进制,其中 $\bar{1}$ 表示数 -1 。对任意实数 A ,其 MSD 表达式为

$$A = \sum_i a_i 2^i \quad (1)$$

式中, i 为整数, $a_i \in \{0, 1, \bar{1}\}$ 。任意一个(除 0 外)MSD 数都可以具有多种表达形式。

2.2 MSD 数加法

MSD 数无进位并行加法的方法有很多,但考虑到编码规则以及在光学计算机上实现的难易程度,现在采用的是利用 T, W, T', W' 这 4 种变换来完成 MSD 加法^[11],这几个变换如表 1 所列。

T	$\bar{1}$	0	1	W	$\bar{1}$	0	1
$\bar{1}$	$\bar{1}$	$\bar{1}$	0	$\bar{1}$	0	1	0
0	$\bar{1}$	0	1	0	1	0	$\bar{1}$
1	0	1	1	1	0	$\bar{1}$	0
T'	$\bar{1}$	0	1	W'	$\bar{1}$	0	1
$\bar{1}$	$\bar{1}$	0	0	$\bar{1}$	0	$\bar{1}$	0
0	0	0	0	0	$\bar{1}$	0	1
1	0	0	1	1	0	1	0

在已有的研究中,进行 MSD 数加法的具体步骤如下^[8]。

首先进行数据准备。如果两个 MSD 数的位数不同,则在位数较小的最高位前补 0 以使其位数相等。设 $a = a_{n-1} a_{n-2} \dots a_1 a_0$ 与 $b = b_{n-1} b_{n-2} \dots b_1 b_0$ 为 2 个 MSD 数。然后按下述 3 个步骤进行变换。

第 1 步 将输入数据 a 和 b 的对应位作为输入,进行 T 变换,并将各位结果合并后得到中间值 t ,在 t 后面补一位数码 0,仍记为 t ;同理,对数据 a 和 b 经 W 变换得到中间值 w ,在 w 前面补一位数码 0,仍记为 w 。

第 2 步 以中间值 t 和 w 作为输入,经 T' 变换得到第二组中间值 t' ,在 t' 后面补一位数码 0,仍记为 t' ; t 和 w 经 W' 变换得到第二组中间值 w' ,在 w' 前面补一位数码 0,仍记为 w' 。

第 3 步 以中间值 t' 和 w' 作为输入经 T 变换得到最终结果 c 。此时的 T 变换也记作 T_2 。

经上述 3 个步骤,结果 c 是 a 与 b 的和。

2.3 改进的 MSD 数加法

通过对上述计算过程和对 MSD 4 个变换表的深入研究,本节提出一种对上述 MSD 加法的一种改进算法,具体步骤如下。

第 1 步 将输入数据 a 和 b 的对应位作为输入进行 T 变换,并将各位结果合并后得到中间值 t ,在 t 后面补一位数码 0,仍记为 t ;同理,对数据 a 和 b 经 W 变换得到中间值 w ,在 w 前面补一位数码 0,仍记为 w 。

第 2 步 以中间值 t 和 w 作为输入,经 T' 变换得到第二组中间值 t' ,在 t' 后面补一位数码 0,仍记为 t' ; t 和 w 经 W' 变换得到第二组中间值 w' ,在 w' 前面补一位数码 0,仍记为 w' 。

第 3 步 对第 2 步的结果 t' 和 w' 再进行 W' 变换,即可得

到其两个数 MSD 的和。

上述算法与前述算法的不同就是第 3 步,即在改进的 MSD 数加法中,用 W' 变换代替 T 变换。为了证明此算法的正确性,下面将通过数学验证和计算机验证这两种方法进行证明。

2.3.1 数学验证

从上述两种计算方法看,关键是证明在将 MSD 数据 a 和 b 先经过 T, W 变换、再经过 T', W' 变换后,最后一步经过 T 变换或者经过 W' 变换所得的结果是相同的。

考察 T 与 W' 变换表。对照 T 变换和 W' 变换的真值表,可以发现两者之间的不同点仅在于 T 变换中 1 与 $\bar{1}$ 变换的结果是 $1, \bar{1}$ 和 $\bar{1}$ 变换的结果是 $\bar{1}$,而在 W' 中这两个变换结果均为 0。因此只需证明在 MSD 加法运算中经 T' 变换后的结果和经 W' 变换后的结果(也就是完成上述步骤的第 2 步后)在同一位上不相同或者全为 0,也就是经 T' 和 W' 变换后的两数在同一位上不会出现同为 1 或者同为 $\bar{1}$ 的情况。在这种情况下要证明 T 变换和 W' 变换实际上是等价的。

反证法:假设输入的 MSD 数据 a 和 b 经第 1 步再经第 2 步的 T' 和 W' 变换后的两数在第 n 位($n > 1$)上数值相同且全为 1(或 $\bar{1}$),那么会有表 2 所列的情况。

表 2 MSD 加法运算 T' 和 W' 变换两种结果

位	n						
T' 变换	x	...	1(或 $\bar{1}$)	x	...	x	θ
W' 变换	θ	...	1(或 $\bar{1}$)	x	...	x	x

其中 x 为 $\{0, 1, \bar{1}\}$ 中的任意数, n 代表的是第 n 位($n > 1$), θ 为填充的 0。

根据表 1 中 T' 变换真值表,可知只有当 1 与 1 (或 $\bar{1}$ 与 $\bar{1}$) 的情况下 T' 变换的结果才为 1 (或 $\bar{1}$),所以若经 T' 变换后第 n 位为 1 成立,那么只有当第一步中完成 T, W 变换后的两数的第 $n-1$ 位全为 1 (或 $\bar{1}$)。因为前两步中的 T 和 T' 变换都要向前进一位,所以上一步的第 $n-1$ 位决定了 T 和 T' 变换的第 n 位。根据表 1 中的 W' 变换真值表,可以推出只有当 1 (或 $\bar{1}$) 与 0,或者 0 与 1 (或 $\bar{1}$) 的情况下 W' 变换的结果才为 1 (或 $\bar{1}$),所以若经 W' 变换后第 n 位为 1 成立,那么只有当第 1 步中完成 T, W 变换后的两数的第 n 位为 1 (或 $\bar{1}$) 与 0,或者 0 与 1 (或 $\bar{1}$) 这两种情况,如表 3 所列。

表 3 MSD 加法运算 T 和 W 变换两种结果

位	n		n-1					
情况 1 T 变换	x	...	1(或 $\bar{1}$)	1(或 $\bar{1}$)	x	...	x	θ
W 变换	θ	...	0	1(或 $\bar{1}$)	x	...	x	x
位	n		n-1					
情况 2 T 变换	x	...	0	1(或 $\bar{1}$)	x	...	x	θ
W 变换	θ	...	1(或 $\bar{1}$)	1(或 $\bar{1}$)	x	...	x	x

对这两种情况进行分析,发现它们的一个共同点是经 T, W 变换后的第 $n-1$ 位都为 1 (或 $\bar{1}$),根据表 1 中的 T, W 变换真值表有如下推论:经 W 变换后第 $n-1$ 位为 1 (或 $\bar{1}$),那么被加数和加数的 $n-1$ 位必须为 0 与 $\bar{1}$ (或 1),或者 $\bar{1}$ (或 1) 与 0;若被加数和加数的 $n-1$ 位为 0 与 $\bar{1}$ (或 1),或者 $\bar{1}$ (或 1) 与 0,则根据 T 变换真值表其所对应的值为 $\bar{1}$ (或 1),即 T 变换后第 n 位为 $\bar{1}$ (或 1),而这与以上两种情况的第 n 位分别为 1 (或 $\bar{1}$) 或 0 矛盾,所以假设错误。

上述分析^[1],没有考虑也不必考虑 W 变换的第 n 位以及 T 变换的第 $n-1$ 位。理由是 W 变换的第 n 位是由被加数和

加数的第 n 位决定的,被加数和加数的第 n 位不影响 T 变换的第 n 位,但决定 T 变换的第 $n+1$ 位。 T 变换的第 $n-1$ 位是由被加数和加数的第 $n-2$ 位决定的,被加数和加数的第 $n-2$ 位不影响 W 变换的第 $n-1$ 位,但决定 W 变换的第 $n-2$ 位。

由以上证明可以得到以下结论,经过第二步 T' 和 W' 变换后的两数在同一位上数值不会同为 1 或 $\bar{1}$,所以在 MSD 四则变换加法中的第 3 步可以用 W' 来代替 T 。这个证明表明所述结论是正确的。

2.3.2 程序验证

以下再从程序模拟的角度,对上述改进算法的正确性进行证明,以确保可靠性与可信性。

为简单起见,这里只需验证三位 MSD 数的情况。因为三位 MSD 数每位都有 3 种可能,所以共有 $3^3 = 27$ 种可能。从这 27 种可能中选取被加数和加数,即被加数和加数都有 27 种可能,所以参与验证的数据共有 $27 \times 27 = 729$ 种可能。以下用二维数组保存这些 MSD 数。

表 4 MSD 数 4 个变换核心代码

```

T,W,T',W'对应的 4 个二元三值变换,其中数组 T1,数组 W1 分别代表变换 T',W'。
int T[3][3]={0,1,-1,1,1,0,-1,0,-1};
int W[3][3]={0,-1,1,-1,0,0,1,0,0};
int T1[3][3]={0,0,0,0,1,0,0,0,-1};
int W1[3][3]={0,1,-1,1,0,0,-1,0,0};
int n=0; //全局变量
实现 T,W 变换,其中数组 b 保存 T 变换结果,数组 c 保存 W 变换结果。
void MSD_TransTW(a[][3],b[][4],c[][5]){
for(int i=1;i<=27;i++){
for(int j=1;j<=27;j++){
n++;
b[n][3]=0; //T 变换最低位补 0
c[n][0]=0; //W 变换最高位补 0
for(int k=0;k<3;k++){ //进行 T,W 变换
b[n][k]=T[(a[i][k]+3)%3][(a[j][k]+3)%3];
c[n][k+1]=W[(a[i][k]+3)%3][(a[j][k]+3)%3];
}
}
}
实现 T',W'变换,其中数组 c 保存 T'变换结果,数组 d 保存 W'变换结果。
void MSD_TransT1W1(a[][4],b[][4],c[][5],d[][5]){
int j=0;
for(int i=1;i<=n;i++){
j++;
c[j][4]=0; //T'变换最高位补 0
d[j][0]=0; //W'变换最高位补 0
for(int k=0;k<4;k++){ //进行 T',W'变换
c[j][k]=T1[(a[i][k]+3)%3][(b[i][k]+3)%3];
d[j][k+1]=W1[(a[i][k]+3)%3][(b[i][k]+3)%3];
}
}
}

```

设计的程序含有 2 种不同的验证方法,其设计思路分别为:

1. 第 1 种验证方法是证明在第 2 步 T' 与 W' 变换后的同一位不会出现同是 1 或同是 $\bar{1}$ 的情况。被加数与加数均有 27 种情况。依次选取 27 种被加数中的一种作为 a ,27 种加数的一种作为 b (全部有 $27 \times 27 = 729$ 种情况),分别进行 T 、 W 变换,再进行 T' 、 W' 变换,然后对刚做的 T' 、 W' 变换后的结果进行逐位比较。

2. 第 2 种验证方法是选取对应关系中的全部 729 种可

能,分别进行 T 、 W 变换,再进行 T' 、 W' 变换,最后分别进行 T 、 W' 变换。然后对 T 、 W' 变换后的结果进行逐位比较,查找不相同的数位。

以下程序是在 VC++6.0 编译环境下编写的 C++ 程序,核心代码见表 4,表 5。

表 5 2 种验证算法的核心代码

```

实现 T',W'变换的结果按位比较并统计其中同一位数值不同或全为 0 的对应关系系数。
int MSD_Method1(int a[][5],int b[][5]){
int flag=0;
for(int i=1;i<=n;i++){
for(int k=0;k<5;k++){
if(a[i][k]==b[i][k]&&!(a[i][k]==0)){
cout<<"结论不成立"<<endl;
break;
}
else flag++;
}
}
return flag/5;
}
数组 a 存放的是第 3 步完成 T 变换的结果,数组 b 存放的是第 3 步完成 W' 变换的结果,对数组 a 和 b 进行逐位比较,统计其中对应相等的数组个数。
int MSD_Method2(int a[][5],int b[][5]){
int flag=0;
for(int i=1;i<=n;i++){
for(int k=0;k<5;k++){
if(a[i][k]!=b[i][k]){
cout<<"结论不成立"<<endl;
break;
}
else flag++;
}
}
return flag/5;
}

```

2.3.3 实验结果及结论

第 1 种验证的实验结果表明,在第 2 步 T' 与 W' 变换后的同一位不会出现同是 1 或同是 $\bar{1}$ 的情况。

第 2 种验证的实验结果表明,对 729 种所有可能的输入分别进行 T 、 W 变换,再进行 T' 、 W' 变换,然后分别进行 T 、 W' 变换,最后经过 T 或者 W' 变换后的结果是相同的。也间接证明了数学验证方法的正确性与可靠性。

这 2 种验证方法表明最后一步经 T 变换和 W' 变换的结果相同,同时也表明,在 MSD 加法的最后一步,可以将 T 变换替换为 W' 变换。

2.4 实例

表 6 MSD 加法过程

		MSD 加法								
		被加数	1	$\bar{1}$	0	1	$\bar{1}$	0	0	1
		加数	$\bar{1}$	1	1	0	0	$\bar{1}$	0	1
第 1 步	T 变换	0	0	1	1	$\bar{1}$	$\bar{1}$	0	1	θ
	W 变换	θ	0	0	$\bar{1}$	$\bar{1}$	1	1	0	0
第 2 步	T' 变换	0	0	0	0	$\bar{1}$	0	0	0	θ
	W' 变换	θ	0	0	1	0	0	0	1	1
第 3 步	T 变换	0	0	0	1	$\bar{1}$	0	0	1	1
		改进后的 MSD 加法								
		被加数	1	$\bar{1}$	0	1	$\bar{1}$	0	0	1
		加数	$\bar{1}$	1	1	0	0	$\bar{1}$	0	1
第 1 步	T 变换	0	0	1	1	$\bar{1}$	$\bar{1}$	0	1	θ
	W 变换	θ	0	0	$\bar{1}$	$\bar{1}$	1	1	0	0
第 2 步	T' 变换	0	0	0	0	$\bar{1}$	0	0	0	θ
	W' 变换	θ	0	0	1	0	0	0	1	1
第 3 步	W' 变换	0	0	0	1	$\bar{1}$	0	0	1	1

给定 $X=73, Y=-35$, 它们的 MSD 表示分别为 $X_{MSD} = 1\bar{1}01\bar{1}001, Y_{MSD} = \bar{1}1100\bar{1}01$, 过程如表 6 所列, 其中 θ 为填充的 0。

3 改进的加法器的设计

3.1 加法器设计的思想

根据以上分析, 在 MSD 加法中的最后一步用 W' 变换代替 T 变换是可行的。这样就给我们在设计光学加法器方面提供了新的思路和方法。

图 1 所示是第一种 MSD 加法器设计思想。它是根据以上改进的 MSD 加法设计出来的一个加法器原理图, 其中 T, W, T', W' 分别代表进行 4 个变换的区。根据原理图, 用重构电路来设计 T, W, T', W' 这 4 个分区, 让被加数和加数中的一个作为主光路, 另一个作为控制光路。

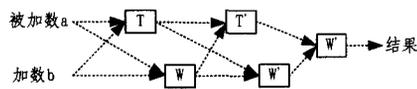


图 1 第一种 MSD 加法器的原理图

图 2 所示是第二种设计思想。我们减少一个 W' 分区形成 4 个分区的加法器结构。对这个加法器而言, 需要增加回溯机制以及同步机制。第一次经 W' 变换的结果需要返回 W' 分区与 T' 分区送来的结果再进行一次 W' 变换, 然后输出最终结果。所谓同步机制就是当有最终结果产生时才允许 T' 分区和 W' 分区的结果向 T' 分区和 W' 分区传送。

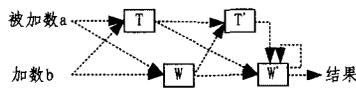


图 2 第二种 MSD 加法器的原理图

3.2 改进后的 MSD 加法在三值光学处理器上的优势

用 W' 变换代替 T 变换有一定的优势。

根据降值设计理论^[6], 把光学运算器分成大小相等的 4 个区域, 即 VV 区、VH 区、HH 区和 HV 区。VV 区两侧均为垂直偏振片; VH 区左侧为垂直偏振片, 右侧为水平偏振片; HH 区两侧均为水平偏振片; HV 区左侧为水平偏振片, 右侧为垂直偏振片。所以在 VV 区和 HV 区只能输出垂直偏振光和无光, VH 区和 HH 区只能输出水平偏振光和无光。

再根据降值理论, 三值光学计算机最简基元为 18 个, 两个最简基元合并成的复合基元为 18 个, 三个最简基元合并成的复合基元为 6 个, 于是三值光学计算机共有基元 42 个, 任意一个三值逻辑运算器包含的基元总数不会超过 6 个。而两输入三值逻辑运算器共有 3^9 个, 也就是说用这 42 个基元中的 1 个, 2 个, 3 个, 4 个, 5 个或 6 个就能实现这 3^9 个逻辑运算。要完成上述加法的 4 个变换所需的各分区基元数见表 7。

表 7 4 个区变换基元数

	V-V	V-H	H-H	H-V
T	2	0	2	0
W	0	2	0	2
T'	1	0	1	0
W'	2	0	2	0

其中, T 变换所需要的 V-V 区和 H-H 区的基元为由两

个最简基元合并的复合基元组成, 其余变换所需基元全为最简基元。改进前的 MSD 加法与改进后的 MSD 加法只有在最后一步不同, 前者最后一步采用 T 变换, 而后者采用 W' 变换。对照表 7, T 变换需要 V-V 区和 H-H 区各 2 个复合基元, 而 W' 变换需要 V-V 区和 H-H 区各 2 个最简基元, 所以二者相比, W' 变换用最简基元替代了 T 变换中的复合基元。这样不仅简化了加法器的复杂程度, 降低了实现难度, 也节约了资源。另外, 从表 7 可知, 原先设计一位 MSD 加法器(最后一步采用 T 变换)一共需要 18 个基元; 图 2 所示的加法器只需要 14 个基元, 与上述相比, 基元数大大减少。如果构造多位的 MSD 加法器, 这样的资源节省是很可观的。因此用 W' 变换代替 T 变换是有优势的, 并且也切实可行。

结束语 本文通过对 MSD 加法的研究, 提出了一种改进的三值光并行无进位加法形式, 用 W' 变换代替最后一步的 T 变换。通过理论证明以及一系列的验证, 证明了此方法的正确性和可靠性, 并且从三值光学计算机的基元角度出发, 说明了用 W' 变换代替 T 变换的优势以及可行性。最后给出了加法器设计的新思路, 为设计光学运算器提供了思路。但是在实现方式上还有待研究, 这也是我们今后要研究的重点问题之一。

参考文献

- [1] 李国强, 刘立人, 邵岚. 两层阵列二的补码复数并行算法及其光学实现[J]. 光学学报, 1995, 15(5): 580-585
- [2] 李国强, 刘立人, 钱家钧, 等. 基于逻辑操作和符号数字表达的光学并行负二进制运算[J]. 中国激光, 1997, 24(7): 660-664
- [3] 李玉栋, 孙骞, 李庆国, 等. 利用 bR 膜实现全光布尔逻辑操作[J]. 中国激光, 1999, 26(12): 1104-1107
- [4] 金翊, 何华灿, 吕养天. 三值光计算机基本原理[J]. 中国科学 E 辑, 2003, 33(1): 111-115
- [5] Jin Yi, He Hua-can, Ai Li-rong. Lane of parallel through carry in ternary optical adder[J]. Science in China (Series F), 2005, 48(1): 107-116
- [6] Yan Jun-yong, Jin Yi, Zuo Kai-zhong. Decrease-radix design principle for carrying/borrowing free multi-valued and application in ternary optical computer[J]. Science in China (Series F), 2008, 51(10): 1415-1426
- [7] 李梅, 何华灿, 金翊, 等. 一种实现 MSD 加法的光学方法[J]. 光子学报, 2010, 39(6): 1053-1057
- [8] Jin Yi, Shen Yun-fun, Peng Jun-jie, et al. Principles and construction of MSD adder in ternary optical computer[J]. SCIENCE CHINA Information Sciences, 2010, 53(11): 2159-2168
- [9] Avizienis A. Signed-digit number representations for fast parallel arithmetic[J]. IRE Trans Electron Comp, 1961, EC-10: 389-400
- [10] Draker B L, Bocker R P, Lasher M E, et al. Photonic Computing Using the Modified Signed-Digit Number Representation[J]. Optical Engineering, 1986, 25: 038-043
- [11] 王先超, 姚云飞, 金翊. 基于三值光计算机的并行无进位加法[J]. 计算机科学, 2010, 37(2): 290-292
- [12] 金翊, 王宏健, 欧阳山, 等. 可重构三值光学处理器的基本结构和实现[J]. 中国科学 E 辑