②

计算机科学 1996 Vol. 23 №. 1

5-9

基于符号替换逻辑的光学数字计算

罗金平　陈书明　周兴铭
（国防科技大学计算机系　长沙 410073）

TP302.2

A
摘　要　本文综述了利用符号替换逻辑进行光学数字计算的原理和技术途径，分析探讨了基于符号替换逻辑光学数字计算技术的发展前景。

关键词　符号替换逻辑，光学数字计算

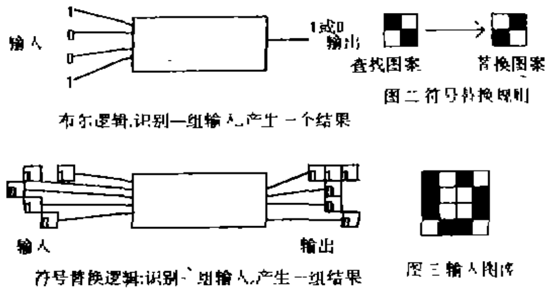
布尔逻辑　数字逻辑

一、前　言

传统的电子数字计算技术的进一步大跨度发展已在诸多方面受到自身固有特性的制约，如何利用光的内在并行性进行高速的数字计算曾一度成为众多的计算机科学家所共同关注的焦点。据估算，每秒1亿个浮点结果的Cray-1计算机，其性能仅与1000×1000点阵、帧频率为每秒3200帧的空间光调制器相当[1]。光学数字计算为计算机系统性能的大幅度提高提供了诱人的前景。光的内在并行性、极高的时空带宽积、和较小的电磁干扰等特性使其成为信息传输和数字计算的极好载体。本世纪六十年代初，世界上许多著名大学及研究机构，如AT&T贝尔实验室和日本的NTT，就已开始了光传输和光计算方面的研究，并取得了丰硕的研究成果。近年来，我国华中理工大学、南开大学、长春光机所、上海光机所等单位进行了这方面的研究，取得了一定的成绩。光作为信息传输载体，其优越性已得到肯定；作为数字计算载体，其优越性的利用还有待进一步的研究努力。已有的研究提出了多种光学数字计算技术，如光学投影数字逻辑、光学符号替换数字逻辑、光学偏振数字逻辑等。在众多的技术中，符号替换逻辑不论在理论上还是在实际实现上，都是相对完善的一种。本文介绍了符号替换逻辑的基本原理；利用符号替换逻辑进行简单二进制算术运算的途径；改进的带符号数算术运算的方法；再编码的MSD数算术运算；对称式再编码的MSD数算术运算；符号替换逻辑的光学实现途径。最后分析探讨了利用符号替换逻辑进行光学数字计算的发展前景。

二、符号替换逻辑的基本原理

符号替换逻辑于1983年被提出，其操作分两步：首先是识别操作，即在给定的图像中识别指定的图案；然后是替换操作，即用指定的图案替换所识别到的图案。符号替换逻辑与传统的布尔逻辑的不同在于布尔逻辑将一组二进制输入变换为一个输出，符号替换逻辑则将一组输入变换为一组输出，而且其输出还与输入数据的空间位置有关；从此意义上讲，符号替换逻辑是布尔逻辑的超集，见图一。

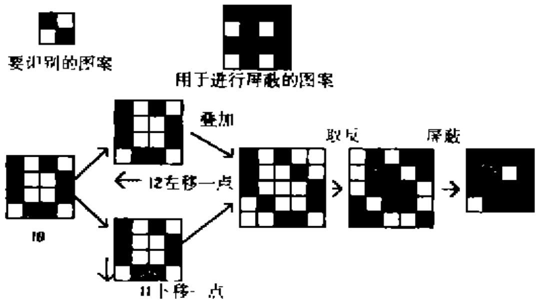


图一　两种逻辑的比较

下面举例详细说明符号替换逻辑的基本步骤。假定要执行图二所示的逻辑规则，即在给定的输入图像中查找图二中左边所示的图案，用图二中右边所示的图案加以替换。设输入的图像如图三所示，符号替换操作如下：

1. **识别指定图案**

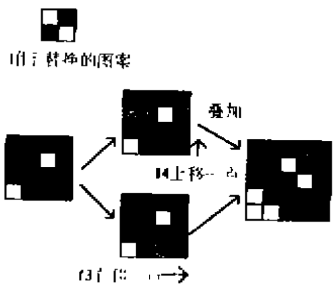
识别过程见图四所示。假定图案的坐标原点在图案的左下角。由于指定的查找图案中有二个白点，两个白点分别位于要识别的图案的左下角和右上角，故需产生输入图像的两份拷贝I1和I2，将I1下移一点和I2左移一点的图像叠加，然后取反，利用屏蔽图像进行屏蔽后所得图像上的白点即表示所要识别的图案在输入图像上的位置，至此识别过程完成。

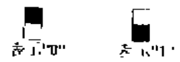


图四 在给定的图像中识别指定的图案

2. 用指定的图案进行替换

替换过程见图五所示。由于替换图案中有两个白点，两个白点分别位于用于替换的图案的左上角和右下角，故将识别过程所产生的结果产生两个拷贝I3和I4，将I3右移一点和I4上移一点所产生的图案叠加即可得到最终的替换结果。





图五 用指定的图案进行替换 图六 "0"与"1"的表示

以上所述的仅是一条符号替换规则的执行过程，实际完成一定的功能需要许多的替换规则，相应必须产生输入图像的多份拷贝，并行完成替换规则，实现指定的功能。

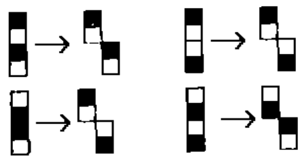
三、二进制算术运算

1. 符号替换逻辑中"0"与"1"的表示

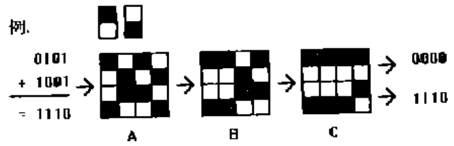
在符号替换逻辑与投影逻辑中，可以用光的有无表示"1"与"0"，如用有光表示"1"，用无光表示"0"；但这种表示中，"1"与"0"所含能量不相等，在实际实现时有一定的问题，如"0"取反为"1"时，不能从无光状态转换为有光状态。可以用图像中的双点表示逻辑上的"0"与"1"，见图六。这种方式的优点是"1"与"0"所含能量相等，在"1"与"0"相互转换时不会遇到能量上的问题。

2. 利用符号替换逻辑实现二进制算术运算

利用符号替换逻辑可以实现二进制加法。加法规则见图七。图八是利用图七所示规则进行两个四位二进制数相加的例子。图中图案A是两个输入的四位二进制数的编码，图案B是第一次利用图七中的规则对图案A进行替换的结果，图案C是第二次进行替换的结果，最后从图案C的下一行中即可得到两数相加的结果。



图七 二进制加法的符号替换规则



图八 4位二进制数加法举例

其它的算术运算如减、乘、除，可以通过适当的编码，再利用上述的加法规则实现。

四、改进的带符号数加减运算

在改进的带符号数(MSD：Modified Signed-digit)数字表示中，任何数X都可利用$X_{n-1}X_{n-2}\cdots X_1X_0$表示，其中$X_i\in\{1,0,\bar{1}\}$，$0\leqslant i\leqslant n-1$(注：$\bar{1}=-1$)。用这种方式可进行并行的无进位和借位的加减运算，这种运算可3步得到加法结果，4步得到减法结果，所需的运算步数与数的位数无关。表一是在MSD数加减运算中所用到的符号替换规则。加法首先用规则一进行符号替换，其次使用规则二进行符号替换，最后再用规则一进行符号替换即可得到加法结果，减法首先使用规则三求补，然后进行与加法相同的三次替换。下面举例说明。

例 MSD数加法运算

$10\bar{1}\,\bar{1}1010\,\bar{1}01\,\bar{1}$ 1433

$+1\,\bar{1}\,\bar{1}100\,\bar{1}10110$ +758

$=01\,\bar{1}001\,\bar{1}00100\,\bar{1}1$ =2191(十进制)

被加数：$10\bar{1}\,\bar{1}1010\,\bar{1}01\,\bar{1}$

加 数：$1\,\bar{1}\,\bar{1}100\,\bar{1}10110$

第一次使用规则1进行替换得到：

$1\,\bar{1}\,\bar{1}01001\,\bar{1}11\,\bar{1}\varnothing$

$\varnothing0100\,\bar{1}00\,\bar{1}1\,\bar{1}01$

表一　MSD 数加减运算符号替换规则

规则1　加[illegible]规则

0 0 → 0 0　　1 1 → 0 1　　0 1 → 1 1　　1 1 → 1 0

$\bar{1}$ 0 → $\bar{1}$ 1　　0 $\bar{1}$ → 1 1　　$\bar{1}$ 1 → $\bar{1}$ 0　　1 0 → 1 0　　1 0 → $\bar{1}$ 0

规则2　加法规则

0 0 → 0 0　　1 0 → 0 1　　0 1 → 1 1　　1 1 → 1 0

1 0 → 0 $\bar{1}$　　0 0 → $\bar{1}$ $\bar{1}$　　1 1 → $\bar{1}$ 0　　$\bar{1}$ 0 → 1 0　　1 0 → $\bar{1}$ 0

规则3　[illegible]规则

0 0 → 0 0　　1 1 → 0 0　　0 0 → 1 1　　1 1 → 1 1

$\bar{1}$ 0 → $\bar{1}$ 0　　0 0 → $\bar{1}$ 1　　$\bar{1}$ 1 → $\bar{1}$ 1　　$\bar{1}$ $\bar{1}$ → $\bar{1}$ 1　　1 1 → $\bar{1}$ 1

第二次使用规则 2 进行替换得到：

00000000 $\bar{1}$1000∅

∅1 $\bar{1}$001 $\bar{1}$01000 $\bar{1}$1

第三次使用规则 1 进行替换得到：

01 $\bar{1}$001 $\bar{1}$00100 $\bar{1}$1 结果

0 $\bar{1}$100 $\bar{1}$100 $\bar{1}$001 $\bar{1}$结果的补码

以 MSD 数的加减运算为基础，同样可以完成乘除运算，实现 MSD 数复杂的运算功能。

五、再编码的 MSD 数算术运算

已有研究证明，只要在 MSD 数中无两个连续的"1"或"－1"，则可以进行无进位传递的加法运算[5]。这样，对 MSD 数的加法运算可分两步进行：首先将 MSD 数编码成 RMSD（Recoded Modified Signed-Digit）；然后利用 RMSD 数加法规则进行符号替换，即可一步完成加法运算。表二是将 MSD 数编码成 RMSD 数的编码表，表三是两个 RMSD 数相加的加法操作真值表。

例　2 个 RMSD 数相加运算。MSD 被加数：011 $\bar{1}$1 $\bar{1}$ $\bar{1}$1；MSD 加数：10 $\bar{1}$1 $\bar{1}$ $\bar{1}$10。由于 K 位 MSD 数转换为 RMSD 数时变为 K+1 位，故在上述 MSD 数前加一"0"；另外为了使用表二进行再编码，在每个 MSD 数后加 3 个"0"，以上 MSD 数变成以下形式：

MSD 被加数：0011 $\bar{1}$1 $\bar{1}$ $\bar{1}$1000

MSD 加　数：010 $\bar{1}$1 $\bar{1}$ $\bar{1}$10000

表二　MSD 数再编码为 RMSD 数的再编码表

$x_i x_{i-1} x_{i-2} x_{i-3} z_i$			
$\bar{1}$ $\bar{1}$ $\bar{1}$d0	$\bar{1}$ $\bar{1}$0 $\bar{1}$0	$\bar{1}$ $\bar{1}$011	$\bar{1}$ $\bar{1}$1d1
$\bar{1}$ $\bar{1}$001	$\bar{1}$0 $\bar{1}$d1	$\bar{1}$00d $\bar{1}$	$\bar{1}$01d $\bar{1}$
$\bar{1}$1 $\bar{1}$d $\bar{1}$	$\bar{1}$10 $\bar{1}$1	$\bar{1}$1000	$\bar{1}$1010
$\bar{1}$11d0	0 $\bar{1}$ $\bar{1}$d $\bar{1}$	0 $\bar{1}$0 $\bar{1}$ $\bar{1}$	0 $\bar{1}$000
0 $\bar{1}$010	0 $\bar{1}$1d0	00dd0	111d0
11010	110 $\bar{1}$ $\bar{1}$	11 $\bar{1}$d $\bar{1}$	11000
101d $\bar{1}$	100d $\bar{1}$	10 $\bar{1}$d $\bar{1}$	1 $\bar{1}$1d1
1 $\bar{1}$011	1 $\bar{1}$001	1 $\bar{1}$0 $\bar{1}$0	1 $\bar{1}$ $\bar{1}$d0
011d1	01011	01001	010 $\bar{1}$0
01 $\bar{1}$d0			

注：表中 $x_i x_{i-1} x_{i-2} x_{i-3}$ 表示 MSD 数中的相邻 4 位，z_i 表示编码后 RMSD 数中的第 i 位

利用表二所示的再编码表，以上 MSD 数变成以下 RMSD 数：RMSD 被加数：01 $\bar{1}$1 $\bar{1}$010 $\bar{1}$；RMSD 加数：010 $\bar{1}$010 $\bar{1}$0。为了使用表三所示的 RMSD 数加法操作真值表，应在以上所得的 RMSD 数后再加一个"0"，即：

RMSD 被加数：01 $\bar{1}$1 $\bar{1}$010 $\bar{1}$0

RMSD 加　数：010 $\bar{1}$010 $\bar{1}$00

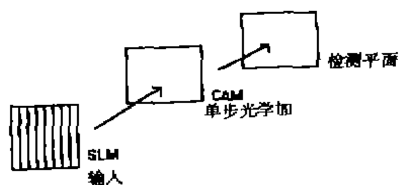
在上述的 RMSD 数上使用表三所示的加法规则即得到以下结果：

RMSD 被加数：01 $\bar{1}$1 $\bar{1}$010 $\bar{1}$0　83（十进制）
RMSD 加　数：010 $\bar{1}$010 $\bar{1}$00　102（十进制）
和：　　　　　10 $\bar{1}$0 $\bar{1}$11 $\bar{1}$ $\bar{1}$　185（十进制）

利用上述办法，可以在一步内得到加法计算结果，其中"0"、"1"、"$\bar{1}$"的光学符号如图九所示，其它的算术运算可以利用上述加法实现。符号替换加法的光学实现见图十所示。

(a)　(b)

图九



注：SLM-空间光调制器　CAM-相联存储器

图十　RMSD 加法的光学CAM实现

表三　RMSD 数的加法操作真值表

$a_ib_ia_{i-1}b_{i-1}s_i$			
$\bar{1}\bar{1}000$	$\bar{1}\bar{1}010$	$\bar{1}\bar{1}100$	$\bar{1}\bar{1}111$
$\bar{1}00d\bar{1}$	$\bar{1}01\bar{1}\bar{1}$	$\bar{1}010\bar{1}$	$\bar{1}0110$
$\bar{1}10\bar{1}0$	$\bar{1}1000$	$\bar{1}11\bar{1}0$	$\bar{1}1100$
$0\bar{1}d0\bar{1}$	$0\bar{1}\bar{1}1\bar{1}$	$0\bar{1}01\bar{1}$	$0\bar{1}110$
$00\bar{1}\bar{1}\bar{1}$	$001\bar{1}0$	$00d00$	11000
$110\bar{1}0$	$11\bar{1}00$	$11\bar{1}\bar{1}\bar{1}$	$100d1$
$10\bar{1}01$	$10\bar{1}\bar{1}0$	$1\bar{1}010$	$1\bar{1}000$
$1\bar{1}\bar{1}10$	$1\bar{1}\bar{1}00$	$01d01$	$011\bar{1}1$
$010\bar{1}1$	$01\bar{1}\bar{1}0$	00111	$00\bar{1}10$
$000d0$	$10\bar{1}11$		

注：a_i 被加数的第 i 位，b_i 加数的第 i 位，s_i 和数的第 i 位

六、对称式再编码的 MSD 数的加减运算

从前面的分析可以看出，利用光学符号替换逻辑实现 MSD 数的加减运算所用步数较少，例如前面分析的方案中可以在 3 步或 1 步即可得到加法结果；但是以上的实现中所用的再编码规则和加法规则过多，这一方面造成硬件成本过高，另一方面给规则并行带来困难，因而减少符号替换规则数量在符号替换逻辑中是一个不可忽视的因素。在 MSD 数的表示中，MSD 数的任一位 $x_i \in \{0, 1, \bar{1}\}$，设 x_i 的补码为"x'_i"，则 $1' = \bar{1}$，$0' = 0$，$\bar{1}' = 1$。可见 MSD 数据表示具有良好的对称性，因而可以利用这种对称性重新设计新的编码规则，从而减少符号替换规则的数量。

从表二可以看出，利用表二再编码得到的 RMSD 数不具备对称性，如输入$\bar{1}\bar{1}00$时，再编码输出为 1，如果输入求补，即输入为 1100 时，按表二得到的再编码输出为 0，而不是 1 的补码 $\bar{1}$，这样可以修改表二所示的再编码规则，得到表四所示的对称式的 MSD 数的再编码规则表。利用表四所示的再编码规则对输入的 MSD 数进行再编码即可得到相应的 SRMSD 数（Symmetrically Recoded Modified Signed-Digit），利用表三所示的符号替换规则即可对再编码得到的 SRMSD 数进行加法运算。

例：$-3175+1159=-2016$（十进制）。

MSD 运算：

$$\begin{array}{lr} \bar{1}\bar{1}00\bar{1}00111\bar{1}\bar{1}(\mathrm{MSD}) & -3175 \\ +011\bar{1}\bar{1}001\bar{1}111(\mathrm{MSD}) & +1159 \\ \hline =0\bar{1}00000100000(\mathrm{MSD}) & -2016 \end{array}$$

RMSD 运算：

$$\begin{array}{lr} \bar{1}1\bar{1}00\bar{1}010\bar{1}01\bar{1}(\mathrm{RMSD}) & -3175 \\ +01\bar{1}01\bar{1}000100\bar{1}(\mathrm{RMSD}) & +1159 \\ \hline =0\bar{1}00000100000(\mathrm{MSD}) & -2016 \end{array}$$

表四　MSD 数再编码为对称式
SRMSD 数的再编码表

$x_ix_{i-1}x_{i-2}x_{i-3}v_iw_i$			
$\bar{1}\bar{1}\bar{1}d00$	$\bar{1}\bar{1}0\bar{1}00$	$\bar{1}\bar{1}0111$	$\bar{1}\bar{1}1d11$
$\bar{1}\bar{1}0001$	$\bar{1}0\bar{1}d11$	$\bar{1}00d1\bar{1}$	$\bar{1}01d\bar{1}\bar{1}$
$\bar{1}1\bar{1}d\bar{1}\bar{1}$	$\bar{1}10\bar{1}\bar{1}\bar{1}$	$\bar{1}1000\bar{1}$	$\bar{1}10100$
$\bar{1}11d00$	$0\bar{1}\bar{1}d\bar{1}\bar{1}$	$0\bar{1}0\bar{1}\bar{1}\bar{1}$	$0\bar{1}0000$
$0\bar{1}0100$	$0\bar{1}1d00$	$00dd00$	$111d00$
$110d00$	$110\bar{1}\bar{1}\bar{1}$	$11\bar{1}d\bar{1}\bar{1}$	$11000\bar{1}$
$101d\bar{1}\bar{1}$	$100d\bar{1}1$	$10\bar{1}d11$	$1\bar{1}1d11$
$1\bar{1}0111$	$1\bar{1}0001$	$1\bar{1}0\bar{1}00$	$1\bar{1}\bar{1}d00$
$011d11$	010111	010010	$010\bar{1}00$
$01\bar{1}d00$			

注：表中 $x_ix_{i-1}x_{i-2}x_{i-3}$表示 MSD 数中的相邻 4 位，$v_i w_i$ 表示编码后 SRMSD 数中的第 i 位，$v_i w_i$ 是两种不同的编码方案，两种方案均具备对称性

按表四 v_i 列再编码后进行的 SRMSD 数加法运算：

$$\begin{array}{lr} \bar{1}010\bar{1}1010\bar{1}001(\mathrm{SRMSD}) & -3175 \\ +01\bar{1}001000100\bar{1}(\mathrm{SRMSD}) & +1159 \\ \hline =\bar{1}100000100000(\mathrm{MSD}) & -2016 \end{array}$$

按表四 w_i 列再编码后进行的 SRMSD 数加法运算：

$$\begin{array}{lr} \bar{1}1\bar{1}00\bar{1}010\bar{1}01\bar{1}(\mathrm{SRMSD}) & -3175 \\ +01\bar{1}01\bar{1}000101\bar{1}(\mathrm{SRMSD}) & +1159 \\ \hline =0\bar{1}00000100000(\mathrm{MSD}) & -2016 \end{array}$$

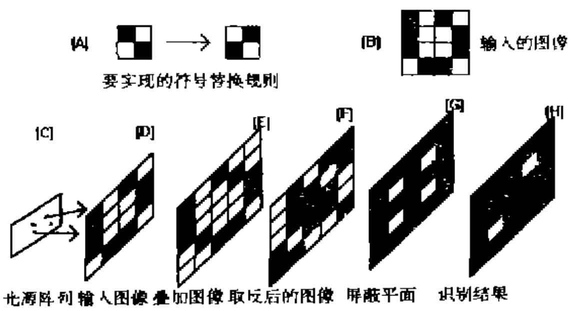
在这种方案中，产生"1"的规则可以由产生"$\bar{1}$"的规则求补得到，故而可以极大程度减少规则数量，便于物理实现。

七、符号替换逻辑的光学实现

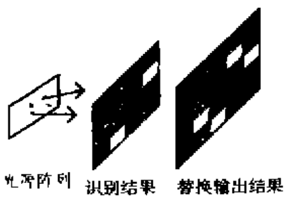
符号替换逻辑分两步：识别和替换，在识别过程中，首先根据要识别的图案将输入图像产生一定数量的拷贝，将拷贝根据识别图案的不同进行相应移动后叠加，叠加后的图像取反屏蔽后即可得到识别结果；在替换的过程中，将识别结果按替换图案的不同产生一定数量的拷贝，然后将所得拷贝进行相应的叠加即可得到替换结果，详细的符号替换原理见本文第二部分。

目前已有多种符号替换逻辑的光学实现途径问世，这里仅介绍利用投影逻辑实现的符号替换逻辑。投影逻辑的详细原理参见文[1]，假设所要实现的符

号替换规则见图十一(A),输入的图像见图十一(B),其实现过程见图十一(C)、(D)、(E)、(F)、(G)、(H)所示。由于所要识别的图案中有两个白点,且两白点在对角线上,故在图十一(C)中,光源阵列中相应有两点发光,从而使图十一(D)中所示的输入图像产生两份拷贝在图十一(E)上叠加,叠加结果取反后得到图十一(F),经图十一(G)屏蔽后得到图十一(H)的输出结果,输出图像中的白点即表示所要识别的图案在图像中的位置。图十二是替换过程的光学投影逻辑实现,在替换的过程中,根据用于替换的图案的不同,图中光源阵列中的不同光源发光,使识别结果的投影在输出平面上叠加从而得到替换结果,以上所述的是一条符号替换规则的执行过程,多条符号替换规则的并行可以参见文[8]。



图十一 识别过程



图十二 替换过程

八、前景

符号替换逻辑从问世至今已经历了 20 多年的发展,这种逻辑从理论上讲是完备的,能极好地利用光的内在并行性进行复杂的逻辑运算,因而这种逻辑被认为在人工智能、光学数字计算等方面具有极好的应用前景,在符号替换逻辑问世至今的 20 余年里,已有多种利用符号替换逻辑进行光学数字计算的方案问世,这些方案的共同特点是能充分利用光的内在并行性和符号替换逻辑的逻辑完备性进行高速的并行计算。从理论上讲,这些方案都能极大程度提高计算并行性,进而提高计算机系统的性能;但是从目前的发展状况看,其物理实现途径离实用还有较大距离。现有的器件水平难以有效地实现光学符号替换逻辑,器件技术过去是光学数字计算技术发展的严重障碍,今后仍然会是光学数字计算技术发展所面临的主要难题之一,另外,目前电子数字技术还在迅速发展,电子数字产品性能仍在大幅度提高,且能很好地满足一般应用的需求,价格低廉,众多的电子数字产品已占领市场并为广大的用户所熟悉和接受,光学数字产品只有具备明显的优势才能进入市场,这样就要求光学数字技术必须有很高的起点,这增加了光学数字技术获得成功的难度。光学计算的研究至今已经历了三十余年的发展,但直到今天仍无较成功的光学数字产品进入市场,人们担心在未来的光计算研究中难以继续寻求科研资助。现在,纯光学数字计算的发展面临严峻的考验,纯光学数字计算的研究暂时进入低谷[9]。

参考文献

[1]T. S. Yu, Francis 等, Optical signal processing, computing and neural networks, 1992

[2]Karl-Heinz Brenner 等, Diggital optical computing with symbolic substitution, Appl. Opt., Vol. 25, No. 18, 1986

[3]Richard P. Bocker 等, Modified signed-digit addition and substraction using optical symbolic substitution, 同[2]

[4]Abdul Ahad S. Awwal, Recoded signed-digit binary addition-substraction using optoelectronic symbolic substitution, Appl. Opt., Vol. 31, No. 17, 1992

[5]Abdallah K. Cherri, Symmetrically recoded modified signed-digit optical addition and substraction, Appl. Opt., Vol. 33, No. 20, 1994

[6]B. Parhami, Carr-free addition of recoded binary signed-digit number, IEEE Trans., Computer 37, 1988

[7]Ahmed Louri, Efficient optical implementation method for symbolic substitution logic based on shadow casting, Appl. Opt., Vol. 23, No. 15, 1989

[8]Ahmed Louri, Parallel implementation of optical symbolic substitution logic using shadow-casting and polarization, Appl. Opt., Vol. 30, No. 5, 1991

[9]H. Scott Hinton 等, Optical computing: introduction by the feature editors, 同[5]