



论文

结构量计算机 —— 三值光学计算机的应用特点

金翊^{①②}, 徐群^{①②}, 欧阳山^{①②*}, 韩越兴^{①②}, 李卫民^{①②}

① 上海大学计算机工程与科学, 上海 200444

② 上海大学高性能计算中心, 上海 200444

* 通信作者. E-mail: ouyangshan@shu.edu.cn

收稿日期: 2015-04-11; 接受日期: 2015-07-14; 网络出版日期: 2016-02-23

国家自然科学基金青年基金 (批准号: 61103054)、上海市自然科学基金 (批准号: 13ZR1416000, 15ZR1415400)、上海市教育委员会科研创新项目 (批准号: 13ZZ074, 13YZ005) 和上海高校青年教师培养资助计划 (批准号: ZZSD13035) 资助

摘要 本文论证了三值光学计算机用硬件并行计算一个简单结构量的技术, 阐述了形成这种强大计算能力的自然根源 —— 三值光电混合处理器具有数据位数众多、按位可重构和加法器无进位过程等天然优势. 讨论了如何利用这些优势将三值光学处理器重构成一个众位数复合处理器, 使其包含多个不同功能的计算器, 以及如何应用复合处理器, 在一条操作指令下用硬件处理具有不同数据类型的多个数据. 显然, 当这些不同类型的数据组成一个结构型数据时, 三值光学计算机就展现出在一条指令下处理一个结构量的能力, 因此称这种计算机为结构量计算机, 并以此来表征它在应用方面的特征. 文中详细介绍了应用三值光学计算机计算一个简单结构量的技术概要, 包括 SZG 文件格式、SZG 文件的自动生成方式、对高级程序语言的扩充技术等, 还举例计算了具有 4 个不同类型项 (分量) 的结构量, 通过分析对 1000 组这种结构量的计算过程, 展示了三值光学计算机在计算结构量方面的强大并行能力.

关键词 结构型数据类型 可重构型处理器 众多数据位数 MSD 加法器 三值 光电混合计算机

1 引言

伴随着计算机功能的不断增强, 它能够处理的数据类型和处理方法越来越贴近人类的复杂活动, 从最简单的二值数据逻辑运算和整型数据加法运算, 到浮点数据的四则运算, 再到矢量数据的矢量运算和解线性方程组, 再到结构型数据和非结构型数据的处理, 日趋复杂与高效. 这个过程中硬件和软件相互依存, 相互配合, 实现了数据类型的不断扩充和处理效率的不断提高. 数据类型的每一次扩充, 都将计算机带入一个崭新的应用领域. 尤其是对结构型数据的处理技术, 它将计算机带入了广阔的管理领域. 然而, 在三值光学计算机问世之前还没有能够直接处理结构型数据的硬件设备^[1~5], 这个时期对结构型数据的处理都是通过软件主导下用硬件逐一处理结构型数据的每一个项 (也称为分量) 来实现, 而每个项都是硬件可以直接处理的简单数据类型.

引用格式: 金翊, 徐群, 欧阳山, 等. 结构量计算机 —— 三值光学计算机的应用特点. 中国科学: 信息科学, 2016, 46: 311–324, doi: 10.1360/N112014-00207

三值光学计算机的应用研究实验系统“上大 11”(SD11) 彰显了这种光电混合计算机的四大特色^[5,6]: 三值、数据位数众多、处理器按位可重构和无进位全加器. 这四大特色使得它可以用硬件来处理结构型数据, 在目前阶段已经能够处理以各种三值(包括二值)逻辑型数据和实数型数据为分量的结构型数据, 本文介绍在一个三值光学处理器中对这类结构型数据的各分量同步进行逻辑运算或加减运算的技术, 并称这类结构型数据为简单结构量. 例如, 在一个 SD11 基本型上, 凡逻辑型数据的总位数不超过 1024 位、实数型数据的二进制总位数不超过 380 位的简单结构量, 均可由三值光学处理器执行一条计算指令来同步计算它的各个分量. 另外, SD11 的光学处理器还具有硬件直接扩充 16 倍的能力, 扩充后它就可以用硬件直接处理更大的简单结构量. 本文介绍这项研究中的主要结果.

2 结构型数据类型

目前最常见的结构型数据当属数据库中的每一个记录, 而经典的结构型数据类型的定义方法当属 C 程序语言中结构型数据类型定义语句: struct 类型名^[7], 用这个语句可以定义一个名为 JINYI 的简单结构量数据类型如下(参考文献[7]中称结构型数据类型为结构体类型):

```
struct JINYI
{
    int a;
    float b;
    int c[20];
}
```

{ } 内的每一行是该结构型的一个项, 或称为“分量”, 或称为“成员”. 其第三项是整型数组, 其余各项都是简单数据类型, 它们都是 C 语言的数据类型, 各项均可进行数值计算或逻辑计算.

3 结构量计算机的特征

结构量是指结构型数据类型的一个具体值或变量. 结构量计算机是指能够直接对结构型数据类型的具体值进行处理的计算机. 在概念上, 它是单指令单数据 (SISD) 计算机和单指令多数据 (SIMD) 计算机的延伸. 就概念而言, C 语言和各类数据库管理语言都以软件方式提供了虚拟形式的结构量计算机, 然而这些虚拟机使用的处理器硬设备多是单指令单数据计算机, 或许也用到单指令多数据计算机, 不过在目前的电子计算机中还没有出现结构量处理器硬设备.

单指令单数据处理器执行一条运算指令只能处理一个简单数据类型的量, 如经典的 8086 CPU 就是一个这样的处理器. 当用众核、多核、GPU 或用多个线程同时分别计算一个向量的多个元素时, 由于每个计算核在执行一条指令时只能处理向量的一个元素, 因此各个核仍是单指令单数据处理器, 这种计算方法是将向量分解成标量来计算, 而不是直接计算向量^[8]. 单指令多数据处理器执行一条运算指令就能处理多个相同数据类型的量, 如 intel-64 架构的处理器, 它在执行属于 AVE 指令集的 ADDPS 指令时可以同时处理 8 个 32 位数的加法, 而执行属于单指令多数据 (SIMD) 类指令的 ADDPS 指令时可以同时处理 4 个 32 位数的加法^[9,10], 即, 它们是典型的 8 元素或 4 元素向量加法处理器; 再如, PowerPC 架构的异构众核处理器 Cell 工作在协处理器 (SPE) 的并行层次模式时, 每个 SPE 最多能同时处理 16 个 8 位的数值型数据, 整个 Cell 可以看成是 16 个元素的向量处理器^[11]. 以此类推, 结构量处理器执行一个运算指令就能处理一个结构量 —— 多个不同数据类型的量组成的复合数据, 目前只有三值光学处理器能够做到这一点.

结构型数据类型有两个重要特征, 一是它的具体结构由用户根据问题来确定, 处理器制造商不可能事先知道; 另一个是它的数据总位数多, 且不能事先确定. 前一个特征要求结构量处理器必须做到硬件随时可重构, 考虑到系统的安全性, 一般会采用由用户提交重构要求, 由监控系统 (或操作系统) 的底层模块按用户要求重构处理器的做法. 后一个特征源于结构型数据中包含有多个数据项, 每个项都有各自的数据位数, 因此结构型数据的总数据位数远远大于简单类型数据的位数, 而且所包含的每个项的数据位数以及总数据位数都不能事先确定, 这个特征要求结构量计算机的处理器必须有很多的数据位数, 且可以随意划分成组, 即系统的数据总线必须很宽, 而且数据总线要便于分割成多个相互独立的部分. 众所周知, 目前电子计算机的数据位数不能任意改变, 而且所有数据位的计算功能相同, 虽然允许用户只使用其中的一两个字节, 但剩余的部分不能由其他任务同时使用, 只能“空闲”, 而且用户使用的各数据位也必须进行相同的计算, 因此, 电子计算机的硬件无力满足对结构量计算机的要求. 然而, 三值光学计算机的第一个整机系统就可以达到 1024×16 位、每一位的计算功能都可以根据用户的需要来重构、每一位都可以分配给不同的任务使用, 这些特征使得它的硬件完全满足对结构量计算机的要求, 可以用这种计算机实现执行一条操作指令就完成对一个结构量的计算——构成硬件层面上的结构量计算机.

4 三值光学计算机成为结构量计算机的基础

4.1 三值和数据位数众多的物理根源

“三值”特点来源于光有三个稳定的物理状态适合用于在计算机中表达信息. 这三个物理状态是: 暗光态、水平偏振光亮态和垂直偏振光亮态^[2]. 处理器的物理学本质可以叙述为: 用自身稳定的、传输时保持不变的、易于用器件改变的物理状态集来表达信息, 通过改变物理状态完成信息处理. 用这个观点考察光的物理特性, 就自然地找到这三个适合于计算机处理器中使用的光状态^[1].

“数据位众多”特点源于光线的互不干扰性. 光线相遇后互不干扰这一物理特性不仅使得多束光线可以在同一空间中相互交叉穿越而互不干扰, 更使得很多条信号光线可以不加隔离地排列在一起构成平面图像, 从而使得对大量光学信息的处理可以按平面图像变换的方式进行. 一般而言, 平面图像上的几个像素才能组成光学信息的一位^[12], 但光学阵列器件往往具有很多像素, 这造就了光学计算机的数据位数众多之特色. 例如, 目前用液晶点阵屏构造的三值光学处理器实验系统就可以拥有 16384 位的数据位数, 远远超过了电子计算机的数据位数^[6, 13~15]. 使用这种处理器的三值光学计算机应用研究实验系统——SD11 将是第一台可进行应用研究的众位数计算机系统. 另一方面, 随着液晶响应时间已经达到纳秒量级, 三值光学计算机的工作速度将会接近 GHz^[16], 接近目前主流电子计算机的主频.

4.2 按位可重构的理论基础和处理器结构

“按位可重构”特点源于三值光学逻辑处理器的构造规律——降值设计理论. 这个规律指出: 若表示信息的 n 个物理态中包含有 D 态, 则 $n^{(n \times n)}$ 个两输入的无进 (借) 位 n 值运算器可以按照固定步骤, 从 $n \times n \times (n-1)$ 种最简基元中组合不超过 $n \times (n-1)$ 个最简基元而成. 其中 D 态是与任何一个其他物理态 λ 进行物理叠加后的结果仍是 λ 态的物理状态; 最简基元是只有一种输入数据组合使其输出为非 D 态, 其他的输入数据组合都使其输出 D 态的 n 值逻辑运算器^[17, 18].

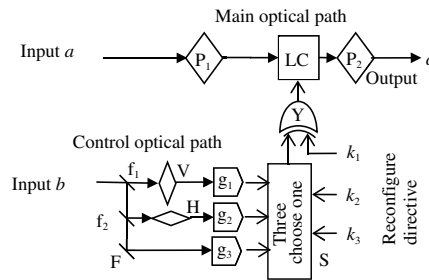


图 1 三值光学计算机基元的统一结构

Figure 1 Uniform structures of TOC's basic operation units

将这个理论应用于三值光学计算机, 则最简基元有 18 个, 组合其中的不超过 6 个就可以构成 19683 个三值逻辑运算器中的任一个. 进一步研究发现, 三值光学计算机的 18 个最简基元有相同的光学结构, 如图 1 所示. 这个结构中有两个光路, 输入数据 a 进入的是主光路, 它由两个偏振片 (P_1 和 P_2) 夹一个液晶像素 (LC) 形成三明治结构. 另一个输入信号 b 进入的是控制光路^[19,20]. 各基元之间的差别在于: 主光路中液晶的静态旋光性相反; 或两个偏振片的偏振取向不同; 或控制光路的特性不同. 控制光路有三种类型, 它们的输入由两个半反射镜 f_1 和 f_2 以及全反射镜 F 对输入光信号 b 分束而成, 它的上分路中有垂直偏振片 V , 则只有 b 为垂直偏振光时, 光电管 g_1 才会输出高电平; 类似地, 它的中间分路中有水平偏振片 H , 则只有 b 为水平偏振光时, 光电管 g_2 才会输出高电平; 而它的下分路中没有偏振片, 则只要 b 为亮, 无论是水平偏振光或垂直偏振光, 光电管 g_3 就会输出高电平. 根据重构指令的 k_2 和 k_3 位的值, 三选一器件 S 从 g_1 , g_2 和 g_3 的输出信号中选出一个, 并将选出的信号送入异或门 Y . 当重构指令的 k_1 位是 1 时, Y 将 S 输出的信号取反, 当 k_1 是 0 时, 不取反. Y 的输出信号用于控制主光路中液晶 LC 的旋光性^[21]. 依据偏振片 P_1 和 P_2 的偏振方向, 主光路被分成四个类型: P_1 是垂直偏振片而 P_2 是水平偏振片的主光路被称为 VH 型, P_1 是水平偏振片而 P_2 是垂直偏振片的主光路被称为 HV 型, P_1 和 P_2 都是垂直偏振片的主光路被称为 VV 型, P_1 和 P_2 都是水平偏振片的主光路被称为 HH 型.

至此可知, 只要将图 1 所示微小单元汇集足够多的数量, 就可以制成一个三值光学处理器. 根据降维设计理论, 在构造一个具体的逻辑运算器时, 首先是选定各类型主光路的像素, 然后给出这些像素的重构命令中的 k_1 , k_2 和 k_3 位即可^[22,23]. 计算任务完成之后, 逻辑运算器将释放占有的所有微小单元, 以便它们被再使用.

既然一个三值光学处理器拥有许多微小单元, 就可以在这个处理器的不同区域上构造出不同的逻辑运算器^[24], 如果每个逻辑运算器恰好与一个简单结构量的每一个分量所实施的逻辑运算一致, 则这个包含有不同逻辑运算器的三值光学处理器就成为一个逻辑型简单结构量处理器. 当运行一条操作指令时, 这个三值光学处理器就能够对相应的结构量的各个逻辑型分量进行同步计算 —— 对逻辑型结构量进行计算. 有关三值光学处理器重构性理论、结构和技术的细节请参考文献^[17,19,21~24].

4.3 无进位的三值光学并行加法器

在加法运算中, 进位是由一个位到相邻高位的串行过程, 这个过程花费的时间正比于数据的位数, 在电子计算机的萌芽期, 这个问题就被注意到^[25]. 1950 年就有用冗余计数方法在加法中消去进位过程的报道^[26,27], 后来, 这种冗余计数方法形成了 MSD 方法^[28]. 在 MSD 方法中, 通过对数据的每一位并行实施 4 个三值逻辑变换来完成二进制加法. 这 4 个三值逻辑变换列在表 1 中, 称为 T , W , T'

表 1 MSD 加法器使用的 T, W, T' 和 W' 变换
Table 1 T, W, T' and W' transforms used in MSD addition

T transform				W transform				T' transform				W' transform			
$b \text{ or } w' \setminus a \text{ or } t'$	$\bar{1}$	0	1	$b \setminus a$	$\bar{1}$	0	1	$w \setminus t$	$\bar{1}$	0	1	$w \setminus t$	$\bar{1}$	0	1
$\bar{1}$	$\bar{1}$	$\bar{1}$	0	$\bar{1}$	0	1	0	$\bar{1}$	$\bar{1}$	0	0	$\bar{1}$	0	$\bar{1}$	0
0	$\bar{1}$	0	1	0	1	0	$\bar{1}$	0	0	0	0	0	$\bar{1}$	0	1
1	0	1	1	1	0	$\bar{1}$	0	1	0	0	1	1	0	1	0

和 W' . 而加法运算被分成 5 步.

第 1 步, 对数据 a 和 b 的每一位实施变换 T 和 W. T 变换的结果记为 u , W 变换的结果记为 w .

第 2 步, 在 u 的后边加一个 0, 记为中间结果 t .

第 3 步, 对 t 和 w 的每一位实施变换 T' 和 W'. T' 变换的结果记为 u' , W' 变换的结果记为 w' .

第 4 步, 在 u' 的后边加一个 0, 记为中间结果 t' .

第 5 步, 对 t' 和 w' 的每一位实施变换 T. 得到加法的最终结果 c .

虽然科学家早已发现通过四个三值逻辑变换器可以实现并行的二进制加法运算. 但是常见的电子计算机只有两个值: 0 和 1, 或者说高电平和低电平, 所以电子计算机无法使用结构简单且并行运算的 MSD 加法器, 而在普通加法器中使用结构复杂的先行进位技术, 在多于 100 位的加法器中使用结构简单但耗时长长的串行进位技术.

对三值光学计算机而言, 数据位数高达千位量级, 无法忍受串行进位加法器中进位过程造成的长延时, 也不能构造出千位先行进位器件, 因而必须创建新的无进位的全加器. 2009 年作者利用三值光学逻辑运算器的按位可重构性, 在其数据位的不同区段上分别构造出二进制 MSD 加法器所需要的 4 个三值逻辑运算器 T、W、T' 和 W', 由此构造成了二进制 MSD 并行加法器. 同时给出了提高这种加法器使用效率的数据编辑技术和流水计算技术^[29~31]. 随后 Peng 等^[32] 对这种加法器结构进行了实验和改进, 而沈云付等^[33] 提出的一步式 MSD 加法器只需要一个时钟脉冲就能完成运算, 这和逻辑运算器的运算时间相同. 有关三值光学计算机 MSD 二进制并行加法器的理论、结构和技术的细节请参考文献^[29, 30, 33].

构建三值光学 MSD 加法器的基础有三个, 一是三值光学计算机有三个稳定的物理状态, 二是三值光学逻辑处理器的按位可重构性, 三是三值光学处理器的数据位数很多. 这三个“天然而成”的基础决定了三值光学计算机使用二进制 MSD 加法器为“必然”. MSD 加法器将按位可重构特性从三值光学逻辑运算器扩大到三值光学加法器, 使三值光学处理器可以被重构为具有三值逻辑运算器、二值逻辑运算器、加法器和减法器的复合处理器, 在执行一条运算指令时, 它能完成对简单结构量的各类型分量的同步处理, 虽然目前三值光学计算机能够运算的结构量还受到很大限制, 但它是第一款能够用硬件直接处理结构量的计算机, 为用户简洁明了地掌握三值光学计算机在应用方面不同于电子计算机的特点, 且明晰它与电子计算机的传承, 我们概括三值光学计算机的应用特点为——结构量计算机.

5 结构量计算机的使用方式

虽然三值光学计算机目前还只能计算简单结构量, 但这已经超越了电子计算机的能力, 同时为处理更复杂的结构量奠定了思想和技术基础. 对于简单结构量, 每一个分项的数据类型都和这个分项所

使用的运算器相一致, 于是, 当用户定义了每个分项所使用的运算器时也就给定了这个分项的数据类型, 因此, 在应用三值光学计算机处理简单结构量时不需要再明确给出各分量的数据类型。

三值光学处理器的按位可重构和数据位数众多之特点使它成为结构量计算机, 但重构处理器需要花费数个时钟周期, 如果重构一次处理器后仅计算几对数据, 则重构处理器所花费的时间就和计算数据所花费的时间相当, 如果重构一次处理器后计算大量的数据, 则重构处理器的时间就可忽略不计。因此, 三值光学计算机更适合处理大量数据的相同运算。于是, 使用三值光学计算机时要把对处理器的重构要求和大量初始数据同时送至三值光学处理器。再考虑到一个运算任务很少用到数千位的处理器, 往往需要将三值光学处理器分享给几个计算任务, 因此在程序运行时才将运算规则和原始数据送给 CPU 的方式不适合三值光学处理器。为适应三值光学处理器的特点, 研究者设计了“运算数据文件”交互方式, 这种交互方式的特征是将用户的计算要求和大量的原始数据打包成一个专用格式文件, 称为三值光学计算机运算数据文件——SZG 文件, 存放于用户端外存上, 文件名为 *.szg, 星号是用户给定的文件名, .szg 是专用后缀。在应用程序中, 当需要用三值光学处理器计算数据时, 使用专门扩充的编程指令将 *.szg 文件送入三值光学计算机, 然后等待三值光学计算机送回的计算结果文件——*-r. szg。

采用运算数据文件交互方式后, 三值光学处理器可以在一个短时间内收到多个用户的运算请求, 统筹收到的运算请求就能较好地发挥三值光学处理器数据位众多和按位可重构之优势。这意味着三值光学处理器端具有一个任务管理软件, 这个软件负责解读用户发来的 SZG 文件, 并安排光学处理器计算用户的数据, 然后把计算结果以文件方式返回给各个用户。

5.1 运算数据文件格式

为提高众多数据位的利用率, 三值光学计算机采用运算数据文件 (*.szg 文件) 方式将用户在应用程序中的运算需求和原始数据送往光学处理器, 也用同样的方式将运算结果 (*-r.szg) 送回应用程序。实现这个技术的核心是 SZG 文件格式, 图 2 给出了 SZG 文件格式的第 2 版 SZGSJWJGS-jy2013。显然, 随着三值光学计算机功能的不断提高, SZG 文件格式也会不断更新, 考虑到这种必然的持续发展过程, 2012 年定义 SZG 文件格式第 1 版 SZGSJWJGS-sk2012 时就在文件头部分设置了很多预留位^[34]。SZG 文件的第 1 版证明了这种以运算数据文件为核心的运算请求方式在三值光学计算机应用程序中的适用性, 为针对三值光学计算机特点, 成功地扩充 C 程序语言和扩充 MPI 语句提供了技术支撑; 第 2 版则比较好地定义了 SZG 文件中对简单运算器和结构型运算器的编码, 为用户使用 SZG 文件处理简单运算、复合运算、简单结构量和数据库记录型的结构量奠定了基础; 正在研究的第 3 版则要定义常见运算的具体编码、设定结果变量名、用户要求的结果数据位数、更完善的结构型数据类型定义方式等, 为第三方自由开发运算例程建立技术支撑。

图 2 中 1)~23) 项是 SZG 文件的文件头 (file head) 部分, 该部分记录了这个 SZG 文件的总体情况和必要的参数, 它是三值光学计算机的任务管理模块、数据位管理模块和处理器重构模块为这个任务服务的桥梁。在头文件中 1)~6) 项是文件参数部分 (file parameter), 它描述这个 SZG 文件的整体情况, 包括: 信息页数 (sum of information pages; 表示该文件的大体规模)、优先级 (priority)、IP 地址 (IP address; 用户的通信地址)、文件名 (file name; 用户标识) 和字符与内码的映射 (map between input character and inner coder; 将用户选定的字符与三值光学计算机处理系统使用的标准编码对应), 并为今后发展预留了 6 位 (reserved 1)。8)~11) 项是一个信息页的参数区 (information page parameter), 它给出该信息页所描述的具体运算, 有数据类型 (data type; 哪一类运算)、数据指示块数 (sum of data indicators; 本信息页包含了多少种运算), 具体的数据指示块 (data indicator) 和两个预留部分 (reserved

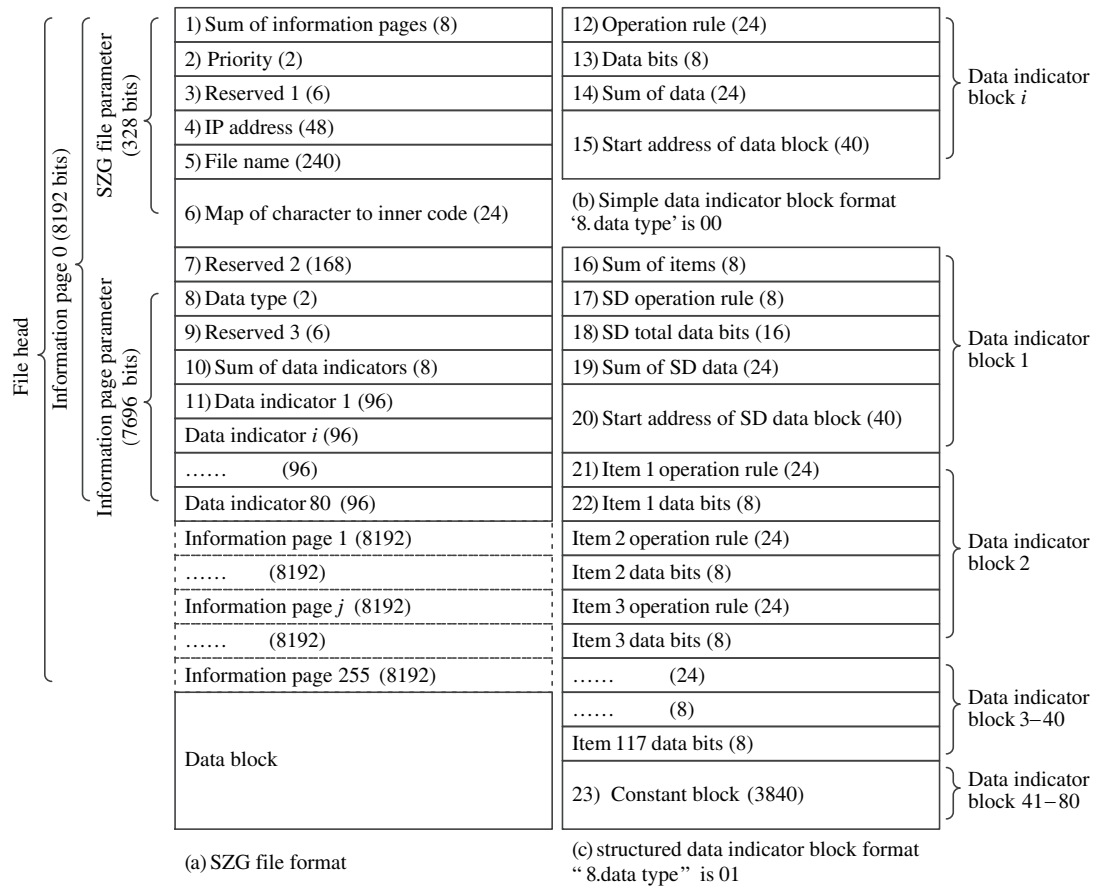


图 2 SZG 文件格式 -SZGSJWJGS-jy2013 版
Figure 2 SZG file format-version SZGSJWJGS-jy2013

2 和 reserved 3). 12)~15) 项给出了一个简单运算的数据指示块结构, 有运算规则 (operation rule)、数据位数 (data bits; 所描述数据中的最多位数, 这是为该运算分配数据位资源的依据)、数据个数 (sum of data; 该运算面对的具体数据量, 这是估算这次运算的任务量的依据) 和数据起始地址 (start address of data block; 该次运算的数据在 SZG 文件中的偏移地址). 16)~23) 项给出结构型数据的格式, 它使用一个信息页来表达结构量的各个项, 其中有项数 (sum of items; 这个结构型数据有多少个项)、结构运算规则 (SD operation rule; 对结构型数据的整体处理, 如排序、查找、投影等)、结构数据总位数 (SD total data bits; 各项的位数之和, 为结构型数据分配数据位资源的依据)、结构数据总数 (sum of SD data)、数据起始地址 (start address of SD data block)、第 i 项的运算规则 (item i operation rule; 对结构型数据中某一项的运算规则, 如加、减、逻辑运算等)、第 i 项的数据位数 (item i data bits) 和常数区 (constant block; 结构型数据整体处理时需要的常数, 如: 查找目标、排序的关键字名称、投影操作的项名称等). 显然, 第 17) 和 18) 项定义对一个数据库型结构量的处理, 第 20) 和 21) 项定义一个简单结构量的第 1 个项, 而“第 i 项的运算规则”和“第 i 项的数据位数”定义了简单结构量的第 i 个项. 本文讨论的内容只涉及对简单结构量的定义.

为了适应在完善 SZG 文件过程中对其格式的适当补充, 图 2 中共预留了 180 位. 但预留位的多少和分段则主要考虑使图 2 中各字段符合当前电子计算机系统以 32 位或 64 位为字长, 以 2048 位或

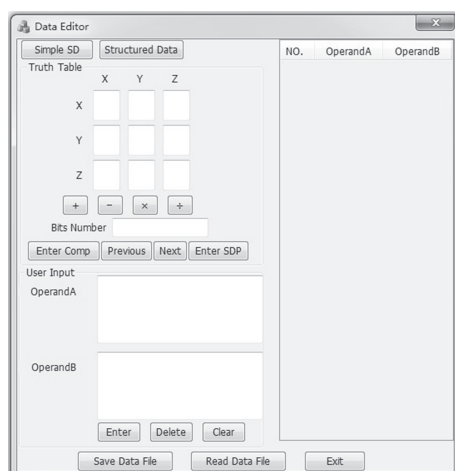


图 3 用户输入数据界面

Figure 3 Inputting data assist interface

4096 位为页长的划定, 以便 SZG 文件的生成和解析易于在电子计算机中完成, 从而用电子计算机作为支撑设备. 其中考虑到字段 1) 和 2) 共 10 位, 故预留字段 3) 为 6 位, 使 1), 2) 和 3) 正好 16 位, 同时与 4) 合并为 64 位字长; 考虑到字段 8) 为 2 位, 故预留字段 9) 为 6 位, 以与 8) 合并为 8 位字节长度, 同时使 8), 9) 和 10) 合并为 16 位字长; 再考虑到每个数据指示块为 96 位, 从字段 8) 到数据指示块 80 共计 7696 位, 故安排字段 7) 预留 168 位. 字段 7) 一方面使信息页 0 正好为 8192 位, 是 4096 页长的 2 倍, 是 2048 页长的 4 倍, 便于生成 SZG 文件和解析 SZG 的电子计算机系统以页为单位进行内存调用; 另一个方面 7) 的前 24 位与 5) 和 6) 合并成 9 个 32 位, 最后 16 位与 8), 9) 和 10) 合并成 32 位, 以方便 32 位或 64 位字长计算机使用.

5.2 SZG 文件的自动生成技术

对一般用户而言, 写出符合 SZG 格式的运算数据文件比较困难, 为减轻使用三值光学计算机的难度, 三值光学计算机团队开发了自动生成 SZG 文件的小软件 —— SZGWJSCRJ-gh2012, 这个软件为用户提供了输入界面, 用户只需在这个界面上输入所需的运算规则和原始数据, 点击 “Enter” 按钮后, 该软件自动将用户输入信息生成相应的 SZG 文件, 并按用户给出的存储路径和文件名保存之^[35]. 随着运算数据文件格式的更新, 这个软件也在逐步升级, 对应于 SZGSJWJGS-jy2013 版本的 SZG 文件自动生成软件输入界面如图 3 所示, 其主要操作有 4 步.

(1) 在界面左上部分第 1 行选按 simple SD 键. 则 SZG 文件生成软件会在图 2 中的 “(8) 数据类型” 字段填入对应于简单结构数据类型的标识码 “01”.

(2) 在界面的左上部分逐一输入简单结构量各个分量的运算规律和数据位数, 每输完一个分量按一下 Enter Comp 键, 所有分量输完后, 按一下 Enter SDP 键. 当第 1 次按 Enter Comp 时, SZG 文件生成软件会在图 2 中的 “(21) 项 1 运算规则” 和 “(22) 项 1 数据位数” 两个字段填入用户刚输入的信息, 当第 i 次按 Enter Comp 时, SZG 文件生成软件会在 “项 i 运算规则” 和 “项 i 数据位数” 两个字段填入用户刚输入的信息.

(3) 用户在这个界面的左下部连续地输入结构量数据, 各分量用逗号分开, 各结构量用分号分开. 输入的每一个数据都在界面的右部显示, 并可以修正.

(4) 按下 Enter 键后, SZG 文件生成软件把输入的原始数据填入图 2 中的数据区, 由此创建成一个符合 SZGSJWJGS-jy2013 格式的三值光学计算机数据文件 $X:\backslash j\backslash *.szg$, 其中 X 是用户指定的存储盘, j 是 X 盘上的文件夹, $*$ 是文件的名称, $.szg$ 是三值光学计算机运算数据文件专用后缀.

5.3 针对三值光学计算机的高级程序语言扩充

将运算规则和原始数据生成 $*.szg$ 文件后, 只需在程序中适时将 SZG 文件送往三值光学计算机即可. 因此, 给一种程序语言扩充一组将 SZG 文件送往三值光学计算机并等待运算结果文件的指令后, 就能在这种语言编写的应用程序中使用三值光学计算机. 一般而言, 这组扩充指令要有以下功能:

- (1) 与三值光学计算机建立通信连接.
- (2) 发送 $*.szg$ 文件给三值光学计算机监控系统的任务管理模块.
- (3) 查询运算结果文件 ($*-r.szg$) 是否已经返回; 或挂起本程序, 等待运算结果文件返回时唤醒.
- (4) $*-r.szg$ 文件返回后继续执行本程序.
- (5) 断开与三值光学计算机的通信连接.

显然, 编写应用程序的过程与生成 $*.szg$ 文件的过程相互独立, 二者可以分别完成. 目前, 对 C 语言和 MPI 并程序语句, 已经完成了针对三值光学计算机的扩充插件^[35,36], C 语言的扩充插件是 SZGCYY-gh2012, 目前它提供 4 个扩充函数:

- (1) void SZG Init(). 建立针对三值光学计算机的扩充指令环境. 它必须是程序中的第一条 SZG 类扩充指令, 且在一个程序中只需运行一次.
- (2) int SZG(char *path). 将地址指针 path 所指的 SZG 文件发送到三值光学处理器.
- (3) int SZG SearchResult(char *path). 查询地址指针 path 所指的 SZG 文件在三值光学处理器上的运算状态. 返回值为“0”表示已经得到运算结果文件 $*-r.szg$; 返回值为“1”表示还没有得到运算结果文件; 返回值为“2”表示在处理该 SZG 文件时三值光学计算机端出现错误, 已经无结果返回.
- (4) void SZG Suspend(). 挂起本程序, 直到运算结果文件返回时被唤醒. 另外, 还要给程序加入头文件 SZG.h. 目前正在对 C++ 和 Fortran 语言进行扩充.

5.4 三值光学处理器对 SZG 文件的运算过程

当 SZG 文件被送入三值光学处理器后, 对其的运算过程可分为 3 大步骤:

- (1) 收到一个 $*.szg$ 文件后, 三值光学计算机的任务管理模块读出这个文件的头部, 并根据用户给出的运算规则来重构光学处理器. 若“8) 数据类型”字段标示为简单结构量数据类型, 则根据各“项 i 运算规则”和“项 i 数据位数”将光学处理器重构成一个简单结构量处理器, 显然, 经过这样重构后的光学处理器已经与用户定义的简单结构量数据类型匹配.
- (2) 简单结构量处理器准备好后, 三值光学计算机监控软件就顺序地将包含在 $*.szg$ 文件中的原始数据送入光学处理器进行计算. 显然, 原始数据的数量越多, 光学处理器的效率越高.
- (3) 计算完 $*.szg$ 文件中的所有原始数据后, 三值光学计算机监控软件将全部计算结果归入 $*-r.szg$ 文件, 并将此计算结果文件返还给用户.

由于重构处理器需要时间, 因此, 当处理的数据量较少时, 使用三值光学处理器会得不偿失.

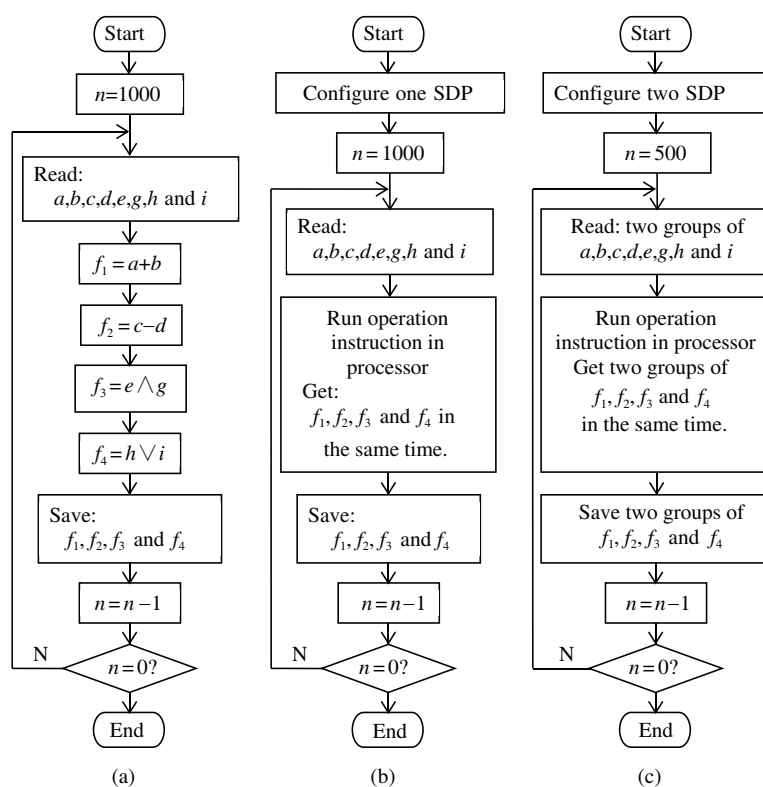


图 4 电子计算机和三值光学计算机处理计算的过程

Figure 4 The operational processes of (a) electronic computer, (b) TOC with one SDP and (c) TOC with two SDP

6 结构量计算机的应用示例

在程序中分别计算几个函数是很常见的情况. 假设需要计算 $f_1 = a + b$, $f_2 = c - d$, $f_3 = e \wedge g$ 和 $f_4 = h \vee i$, 自变量 a, b, c, d, e, g, h 和 i 的位数分别是 32, 32, 16, 16, 5, 5, 7 和 7, 且每一个变量有 1000 个数据. 在电子计算机中, 解决这个问题的算法是: 逐一计算 f_1, f_2, f_3 和 f_4 , 然后重复这个计算过程 1000 次, 如图 4(a) 所示. 显然, 无论使用 CPU、多核、众核或 GPU, 在总数上, 包括加法指令、减法指令、逻辑与和逻辑或在内的运算类指令共要执行 4000 次.

然而, 在三值光学计算机中, 则先定义一个简单结构量数据类型, 记为 S , 定义 S 有 4 个分量, 分别为 32 位整型、16 位整型、5 位逻辑型和 7 位逻辑型. 再定义两个 S 类型的结构变量 A 和 B . A 中包含变量 a, c, e 和 h , B 中包含变量 b, d, g 和 i . 必须说明的是, S, A 和 B 是关于数据类型的定义, 它们体现在通过图 3 所示界面输入数据类型、数据运算规则和数据本身的过程中, 与编写程序的过程无关. 编程者在头脑中完成数据类型定义后, 即可在图 3 所示的输入界面上, 按照 5.2 小节所述的 4 个步骤, 输入 S 的 4 个分量的运算规则和数据位数, 以此告知三值光学计算机为其构造合适的简单结构量处理器. 然后在图 3 所示的输入界面上送入 A 和 B 的 1000 对结构量原始数据, 并生成相应的 SZG 文件, 如 D:\jy.szg. 另一方面, 编制应用程序, 如 C 程序 JY, 并在程序中用扩充指令将 D:\jy.szg 文件送入三值光学处理器.

三值光学处理器接到 D:\jy.szg 文件后, 对这个问题的处理过程如图 4(b) 所示. 首先, 三值光学计算机监控软件的数据位管理模块分配光学处理器的 60 个数据位来构造这个简单结构量处理器, 其前

表 2 三值光学计算机和电子计算机在算法上的性能对比

Table 2 Performance comparison between electronic computer and TOC on algorithm

Performance index	32-bit electronic processor	64-bit electronic processor	A structured data processor	17 structured data processors
Reconstruction time (T_0)	0	0	$T_0 (<10T_y)$	$T_0 (<10T_y)$
Data bits (W)	$4 \times 32 = 128$	$4 \times 64 = 256$	60	1020
Operation cycles (T_y)	$4 \times 1000 = 4000$	$4 \times 1000 = 4000$	1000	$\lceil 1000 \div 17 \rceil = 59$
Amount of resource ($W \times T_y + T_0$)	512000	1024000	$60000 + T_0$	$60146 + T_0$

32 位构造加法器, 跟着的 16 位构成减法器¹⁾, 再 5 位构造“逻辑与”运算器, 最后的 7 位构造“逻辑或”运算器. 然后, 将 A 和 B 的第 1 对结构化原始数据送入这个 60 位的简单结构量处理器, 执行一次运算指令即可得到 f_1, f_2, f_3 和 f_4 的第一组结果. 然后计算第 2 对结构化原始数据, 以此类推, 直至 1000 对结构化原始数据全部计算完, 再把 1000 个结构化的计算结果组成 jy-r.szg 文件, 返回给 C 程序 JY.

在三值光学计算机中, 运算类指令只有一个, 它控制整个光学处理器运行一次, 而电子计算机中运算类指令包含多个具体的计算指令, 各计算指令分别控制各自的运算器运行一次. 因此, 三值光学计算机只需要重复执行 1000 次运算指令, 就可以将 1000 对结构化数据全部算完, 即将 4000 对普通数据全部算完.

如果三值光学计算机再分配光学处理器的另外 60 个数据位构造出第二个简单结构量处理器, 则只需要执行运算指令 500 次就可将这 4000 对普通数据算完, 如图 4(c) 所示. 一般而言, 三值光学计算机有几千位, 可以重构出几十个这样的简单结构量处理器, 这时只需要执行几十次运算指令就可完成这个计算任务. 例如, 在面向应用研究的三值光学计算机实验系统 (SD11) 中, 光学处理器的一个基本模块有 1024 位, 可以构造出 17 个这样的结构量处理器, 因此, SD11 只需要执行 59 次运算指令就能完成这个计算任务. 显然, 运算指令重复的次数越少, 计算过程花费的时间和硬件就越少.

归纳上述讨论, 可以从算法层面将三值光学计算机和电子计算机的性能进行大致的对比, 如表 2 所示.

7 结论

光有三个适合于计算机使用的物理状态, 决定了我们研究的“电控制、光运算”计算机有三个值, 被称为三值光学计算机; 光线在传输时互不干扰的物理特性, 决定了三值光学计算机的数据位数可以很多, 而液晶器件巨大的像素数量, 使三值光学计算机真实地拥有了众多的数据位数; 蕴含于降值设计理论的自然规律决定了三值光学处理器的可重构性; 三值和处理器的可重构性又决定了三值光学计算机适合使用二进制 MSD 并行加法器. 将这些源于自然的三值光学计算机特征与人们应用计算机过程中创立的结构型数据类型相结合, 可以得到三值光学计算机是结构量计算机这个新概念. 由于三值光学计算机最先实现了在一条操作指令下用硬件计算一个简单结构型数据, 它堪称是第一台结构量计算机.

另一方面, 结构量计算机全面地概括和表现了三值光学计算机数据位众多、按位可分配和按位可重构的应用特征. 对于程序员而言, 记住结构量计算机比记住三值光学计算机的特征和了解它的原理、

1) 这里为简化描述将加法器和减法器的各位都视为占一位数据位, 这虽然和实际情况有差别, 但用户感觉不到这种差别, 因此这个简化不失一般性.

结构和工作过程更容易. 因此, 结构量计算机这一概念是迅速而高效地应用好三值光学计算机的基础. 随着三值光学计算机的应用不断被开发成功, 结构量计算机这一概念将比三值光学计算机流传更广, 更为应用者所喜爱.

参考文献

- 1 Jin Y. Ternary optical computer principle and architecture. Dissertation for Ph.D. Degree, Xi'an: Northwestern Polytechnical University, 2002 [金翊. 三值光计算机原理和结构. 博士学位论文. 西安: 西北工业大学, 2002]
- 2 Jin Y, He H C, Lv Y T. Ternary optical computer principle. *Sci China Ser F-Inf Sci*, 2003, 46: 145–150
- 3 Jin Y, He H C, Ai L R. Lane of parallel through carry in ternary optical adder. *Sci China Ser F-Inf Sci*, 2005, 48: 107–116
- 4 Jin Y, He H C, Lv Y T. Ternary optical computer architecture. *Phys Scripta*, 2005, 118: 98–101
- 5 Jin Y. Draw near optical computer. *J Shanghai Univ (Natural Science)*, 2011, 17: 401–411 [金翊. 走近光学计算机. 上海大学学报 (自然科学版), 2011, 17: 401–411]
- 6 Jin Y, Ouyang S, Song K, et al. Management of many data bits in ternary optical computers. *Sci Sin Inform*, 2013, 43: 361–373 [金翊, 欧阳山, 宋凯, 等. 三值光学处理器的数据位管理理论和技术. 中国科学: 信息科学, 2013, 43: 361–373]
- 7 Tan H Q. *C Programming*. 3rd ed. Beijing: Tsinghua University Press, 2005. 281–293 [谭浩强. C 语言程序设计. 第三版. 北京: 清华大学出版社, 2005. 281–293]
- 8 Peter N G. NVIDIA's Fermi: the First Complete GPU Computing Architecture. A White Paper of NVIDIA, 2009
- 9 Intel Corporation. Intel Advanced Vector Extensions Programming Reference. 2010
- 10 Intel Corporation. Intel 64 and IA-32 Architectures Software Developer's Manual (Volume 2A). 2010
- 11 Kahle J A, Day M N, Hofstee H P, et al. Introduction to the cell multiprocessor. *IBM J RES DEV*, 2005, 49: 589–604
- 12 Li S F, Jin Y. Mapping technology from components-pixels to data-bits of ternary optical computer. *Comput Eng Design*, 2010, 31: 1077–1080 [李双凤, 金翊. 三值光学计算机数据位与运算部件像素的映射技术. 计算机工程与设计, 2010, 31: 1077–1080]
- 13 Jin Y. Management strategy of data bits in ternary optical computer. *J Shanghai Univ (Natural Science)*, 2007, 13: 519–523 [金翊. 三值光计算机高数据宽度的管理策略. 上海大学学报 (自然科学版), 2007, 13: 519–523]
- 14 Jin Y, Wang X C, Peng J J, et al. Conceptual structure of ternary optical computer and high performance computer merger. *High Perform Comput Tech*, 2010, 6: 1–4
- 15 Zhan X Q, Peng J J, Jin Y, et al. Static allocation strategy of data-bits of ternary optical computer. *J Shanghai Univ (Natural Science)*, 2009, 15: 528–533 [詹小奇, 彭俊杰, 金翊, 等. 三值光计算机数据位资源的静态分配策略. 上海大学学报 (自然科学版), 2009, 15: 528–533]
- 16 Borshch V, Shiyanovskii S V, Lavrentovich O D. Nanosecond electro-optic switching of a liquid crystal. *Phys Rev Lett*, 2013, 111: 107802
- 17 Yan J Y, Jin Y, Zuo K Z. Decrease-radix design principle for carrying/borrowing free multi-valued and application in ternary optical computer. *Sci China Ser F-Inf Sci*, 2008, 51: 1415–1426
- 18 Yan J Y, Jin Y, Zuo K Z. No carry & no borrow n-value operate unit. China Patent, ZL200710041144.1, 2009-10-28
- 19 Jin Y, Wang H J, Ouyang S, et al. Principles, structures and implementation of reconfigurable ternary optical processors. *Sci China Inf Sci*, 2011, 54: 2236–2246
- 20 Jin Y, Ouyang S, Peng J J, et al. Reconfigurable tri-value optical processor. China Patent, ZL201010584129.3, 2012-05-02
- 21 Ouyang S, Jin Y, Zhou Y, et al. Principle and architecture of parallel reconfiguration circuit for ternary optical computer. *J Shanghai Univ*, 2011, 15: 397–404
- 22 Wang H J, Jin Y, Ouyang S. Design and implementation of 1-bit recongrurable ternary optical processor. *Chinese J Comput*, 2014, 37: 1500–1507 [王宏健, 金翊, 欧阳山. 一位可重构三值光学处理器的设计和实现. 计算机学报, 2014, 37: 1500–1507]
- 23 Wang H J, Jin Y, Ouyang S, et al. Experiment of 1-pixel bit reconfigurable ternary optical computer. *J Shanghai Univ*, 2011, 15: 430–436
- 24 Song K, Jin Y, Ouyang S, et al. Reconfigurable ternary optical processor of double rotator structure. *Optics Precision*

- Eng, 2012, 20: 1890–1898 [宋凯, 金翊, 欧阳山, 等. 双旋光器结构的可重构三值光学处理器. 光学精密工程, 2012, 20: 1890–1898]
- 25 Burks A W, Goldstine H H, Neumann J V. Preliminary Discussion of the Logical Design of an Electronic Computing Instrument. Berlin: Springer, 1946
 - 26 Booth A D. A signed binary multiplication technique. Q J Mech Appl Math, 1951, 4: 236–240
 - 27 Avizienis A. A study of redundant number representations for parallel digital computers. Dissertation for Ph.D. degree. Urbana: University of Illinois, 1960
 - 28 Casasent D, Woodford P. Symbolic substitution modified signed-digit optical adder. Appl Optics, 1994, 33: 1498–1506
 - 29 Jin Y, Shen Y F, Peng J J, et al. Principles and construction of MSD adder in ternary optical computer. Sci China Inf Sci, 2010, 53: 2159–2168
 - 30 Shen Y F, Jin Y, Peng J J. Simulation implementation of the computational principle of MSD adder for ternary optical computer. High Perform Comput Tech, 2010, 6: 5–10
 - 31 Jin Y, Wang X C, Peng J J, et al. Vector-matrix multiplication in ternary optical computer. Int J Numer Analy Model, 2012, 2: 401–409
 - 32 Peng J J, Lu Y P, Jin Y, et al. No carry adder of ternary optical computer. China Patent, ZL201010518342.4, 2010-10-25
 - 33 Shen Y F, Pan L, Jin Y, et al. One-step binary MSD adder for ternary optical computer. Sci Sin Inform, 2012, 42: 869–881 [沈云付, 潘磊, 金翊, 等. 三值光学计算机一种限制输入一步式 MSD 加法器. 中国科学: 信息科学, 2012, 42: 869–881]
 - 34 Song K, Jin Y. Overall plan and design of the task management system of ternary optical computer. J Shanghai Univ, 2011, 15: 467–472
 - 35 Gao H, Jin Y, Song K. Extension of C language in ternary optical computer. J Shanghai Univ (Natural Science), 2013, 19: 280–285 [高桓, 金翊, 宋凯. 针对三值光学计算机的 C 语言扩展. 上海大学学报 (自然科学版), 2013, 19: 280–285]
 - 36 Zhang Q, Jin Y, Song K, et al. MPI programming based on ternary optical computer in supercomputer. J Shanghai Univ (Natural Science), 2014, 20: 180–189 [张茜, 金翊, 宋凯, 等. 在超算集群中使用三值光学计算机的 MPI 编程技术. 上海大学学报 (自然科学版), 2014, 20: 180–189]

Structured data computer—application characteristics of a ternary optical computer

Yi JIN^{1,2}, Qun XU^{1,2}, Shan OUYANG^{1,2*}, Yuexing HAN^{1,2} & Weimin LI^{1,2}

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

2 High Performance Computing Center, Shanghai University, Shanghai 200444, China

*E-mail: ouyangshan@shu.edu.cn

Abstract In this paper, a novel technology for processing simple structured data via the hardware of a ternary optical computer (TOC) is investigated, and the natural sources of this powerful processing ability—massive quantity of data bits of the ternary photo-electronic processor, reconstruction bit by bit, and carry-free adder—are described. It is also discussed that how to use these characteristics to reconstruct a ternary optical processor into a composite one with many data bits and several different functions. Furthermore, this paper details how to apply this kind of composite processor to process vast amounts of different types of structured data in the hardware within one operation instruction. Apparently, when a TOC is applied to process structured data, which is composed of different types of simple data, only one instruction is needed. Thus, its overwhelming superiority is revealed. We call this kind of computer a “structured data computer” for its application features. At the beginning of this paper, a technique outline applying a TOC to process simple structured data is detailed, which includes the SZG file format, a method for generating the SZG file automatically, and the extension for advanced programming languages. Next, an example is given of processing structured data with four components of differ-

ent types through the structured data computer. In conclusion, the strong capacity of the TOC for processing structured data is exhibited by analyzing the computational process of 1000 pairs of structured data.

Keywords structured data, reconfigurable processor, many data bits, MSD adder, ternary, photo-electronic computer



Yi JIN was born in 1957. He received the Ph.D. degree in computer science from the Northwestern Polytechnic University, Xi'an, in 2003. Currently, he is a professor and senior researcher at Shanghai University. His research interests include optical computers and computer architecture.



Qun XU was born in 1985. She received her M.S. degree in computer science from the Qufu Normal University, Rizhao, in 2010. Currently, she is a doctoral candidate in the School of Computer Engineering and Science, Shanghai University. Her research interests include ternary optical computers and basic algorithms.



Shan OUYANG was born in 1979. He received his Ph.D. degree from the School of Computer Engineering and Science, Shanghai University, in 2012. Currently, he is an Engineer at Shanghai University. His research interests include ternary optical computers and embedded systems.



Yuexing HAN was born in 1975. He received his Ph.D. degree in engineering from the University of Electro-Communications, Japan, in 2011. Now, he is working at Shanghai University, and has been a research fellow at Tokyo Institute of Technology, Japan, since 2013. His research interests include molecular robotics, computer vision, and ternary optical computers.