论 文

三值光学处理器的数据位管理理论和技术

金翊*, 欧阳山, 宋凯, 沈云付, 彭俊杰, 刘学民

上海大学计算机工程与科学学院, 上海 200072

* 通信作者. E-mail: yijin@shu.edu.cn

收稿日期: 2012-03-31; 接受日期: 2012-06-19

国家自然科学基金 (批准号: 61073049, 61103054)、上海市重点学科建设项目 (批准号: J50103) 和上海市教育委员会科研创新项目 (批准号: 13ZZ074, 13YZ005) 资助

摘要 数据位众多是三值光学计算机的突出特点之一, 如何有效地管理众多的数据位成为计算机科学领域的新课题. 本文通过继承、更新、增补、归纳和梳理相关的研究成果, 形成了比较系统的数据位管理理论和技术架构. 这个框架包括下列概念和技术: 1) 数据位管理基本单位——“算位”和“算道”, 算位用于标记简单数据类型的数据位, 算道用于标记复合数据类型的数据位. 2) “计算量指标”用于标示一个任务的计算量规模, 对大规模、中规模和小规模的任务, 系统将采取不同的管理策略. 3) “数据位物理形态”指出在数据传送的各个阶段数据位所占用的具体器件. 4) “坏位替换”技术实现了用预留的冗余数据位及时替换失效的数据位, 依靠这项技术, 三值光学计算机具有对数据位软硬件故障的容错能力. 5) “整体重构”技术采用定时重构所有数据位的策略, 显著减少了重构过程对计算过程的打断次数. 6) “数据位分配技术”将对不同规模的任务采用各不相同的规划策略. 另外, 文中还讨论了用这个理论指导设计三值光学计算机第三个实验系统 (SD11) 硬件的情况. SD11 的设计目标是为探索如何应用三值光学计算机提供实验平台. SD11 的光学处理器基本模块有 1024 个数据位, 还有 128 个冗余数据位, 而一个光学处理器可以包含 16 个基本模块, 所以, 一个 SD11 最多可以有 16384 个数据位, 并有 2048 个冗余数据位.

关键词 数据位管理 三值光学计算机 可重构性 数据位分配策略 算位 算道

1 引言

在电子计算机中, 处理器的数据位总是作为整体使用, 没有数据位管理问题. 虽然 64 位或 128 位的媒体运算器可以完成矢量运算, 即可以执行单指令多数据运算, 也只是把一个 64 位或 128 位的运算器作为多个 8/16/32 位的相同小处理器来并行使用, 这相当于把一个 64 位或 128 位的数据宽度分成几个较小的数据宽度来并行使用, 但仍然不能把一个数据位同其他的数据位分开来使用^[1]. 即便是拥有许多运算单元的 GPU, 其每个运算单元的数据位仍然不能被分割, 而且所有运算单元并行地执行相同的操作^[2]. 但是在三值光学计算机的理论和实验系统中^[3~6], 光学运算器众多的数据位数和良好的可重构性两大特点揭示了对数据位进行管理的必要性. 上海大学在构建面向应用研究的新一代三值光学计算机实验系统时, 对数据位管理问题进行了系统研究, 创立了相关的技术、策略和实现方案, 并应用于指导设计新的实验系统. 本文对这些研究进行了系统整理和抽象, 形成了相关的理论.

引用格式: 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, doi: 10.1360/112012-260

2 数据位管理问题的产生

由于光线的空间互不干扰特性, 光学计算机可以采用二维图像处理方式构造运算器, 这奠定了光学计算机数据位数众多的物理基础. 三值光学计算机采用液晶阵列分别构造了百位和千位数据位的实验系统, 目前在建的实验系统数据位数可以扩展到 16384 位^[7]; 另一方面, 依据降值设计理论构造的三值光学处理器具有良好的可重构性^[8,9], 能随意将光学处理器数据宽度中的任何部分构造成任一种三值 (包括二值) 逻辑运算器或加法器, 这为多个用户分享光学处理器或构造包含多种运算的复合运算器奠定了基础. 众多的数据位数和处理器各数据位的可重构性产生了对数据位进行管理的课题, 对这个课题的研究包括: 数据位的组织策略、分配策略、硬件控制技术、重构技术等. 随着三值光学计算机不断完善, 这个课题逐步成为计算机科学中基础研究的一个新方向.

3 数据位管理研究的历史

2003 年, 三值光学计算机还处于萌发期, 作者就注意到理论上这种计算机具有数据位众多的特征, 曾直观地考虑过这些数据位由多用户分享的问题^[10], 从这个角度出发, 提出了算位、算道等基本概念, 并依据这些概念建立了较简单的数据位管理策略, 但当时的这些研究都限于理论推论, 没有实验支持.

2006 年, 在百位量级三值光学计算机原理研究实验系统上, 数据位众多的推测得到证实, 数据位管理随之成为研究课题. 这个时期建立了可行的数据位管理策略^[11]、简单的数据位分配技术^[12,13]和数据位回收技术^[14].

2008 年, 依据降值设计理论建造的三值光学处理器显示出很好的可重构性, 这一特性使千位三值光学计算机结构研究实验系统的数据位和光学处理器的像素位之间出现了一对一和一对二两种映射关系, 要求数据位管理技术必须分别适应这两种映射关系; 同时, 这个实验系统还表明光学计算机更适用于大量数据处理的领域, 在这类应用领域中, 用户进入和退出系统的频度不会太高, 但时间上的随机性很大. 为简化数据位管理策略, 使管理技术实际可行, 研发了数据位到光学处理器像素位的定时映射技术^[15]. 至此基本完成了数据位管理的理论框架、管理策略、管理技术和相关软件结构等工作, 但缺少为管理技术提供支持的底层硬件结构和硬件技术.

2010 年, 开始规划千位三值光学计算机应用研究实验系统——SD11, 这个系统将实现千位数据并行工作, 因此为数据位管理提供底层硬件支持并实现数据位管理的硬件技术成为这个实验系统的主要设计目标之一. 由于数据位管理研究的重要性和新颖性, 它在这一年得到了国家自然科学基金面上项目的资助.

2011 年, SD11 实验系统的硬件设计和关键技术的实验研究已基本完成, 这期间对数据位管理的底层硬件系统进行了充分研究, 得到了相关的硬件设计理论和技术, 并付诸于实践, 形成了比较完整的数据位管理理论和技术架构.

4 数据位管理理论和技术

依靠光线的空间巨并行性, 光学计算机可以拥有很多数据位数的猜想已经在三值光学计算机的两代实验系统中得到证实, 数据位管理随之成为三值光学计算机研究中一个重要的独特方向. 在三值光学计算机第三代实验系统 SD11 的规划、设计和前期实验中, 对数据位管理的理论进行了深入系统的研究, 这个理论在总体上可分为两大部分: 数据位的物理形态和数据位管理技术. 前者以硬件为主体,

后者以软件为主体. 前者的研究始于 2011 年, 在 SD11 的研制中得到关注、抽象和实践, 后者的研究始于 2003 年, 虽然在十余年中不断得到修正, 但在 SD11 的研制过程中才得到硬件的直接支持, 使其原始概念有了切合实际的更新, 其管理策略和技术才进入实际可用阶段.

4.1 数据位的概念

在三值光学计算机中, 目前把数据位的概念表述为: 传送每一位数据的通道在数据总线中的排序; 这组排序符号称为数据位序. 数据位序与数据位的关系类似于内存地址和内存单元的关系.

这个数据位的概念与算术和电子计算机中的数据位概念相关, 但也有明显差别. 算术中的数据位表征各位数在整个数据中的排序位置, 排序值就是该位数的计数权重, 由此构成进位计数制, 但三值光学计算机数据位的排序仅仅是区分各位数据通道的有序符号——数据位序, 没有计数权重的含义. 电子计算机中的数据位与算术中的数据位概念一致, 其更常见的用途是表征二进制数据的总宽度, 如常说的 8 位数据、32 位数据等, 不能随意改变大小, 不能对其中的某一个数据位单独操作, 也不能将一个数据位和不相邻的其他数据位组合在一起, 即便是在位地址空间里, 虽然每个数据位都有一个地址, 但仍很难将不相邻的数据位组合在一起使用. 然而, 在三值光学计算机中, 除数据位序之外, 数据位之间没有其他关联, 因此, 相邻的数据位之间可能毫无关系, 而不相邻的几个数据位也可以任由人们组合在一起使用. 这既给多用户共享整个数据宽度奠定了基础, 又使得数据总线在整体上可以传送不同种类的数据, 形成复合数据总线. 使用好这些新特征正是数据位管理策略的主要任务.

更重要的差别在于: 算术和电子计算机中数据位概念的核心是各位的“计数权重”, 而三值光学计算机数据位概念的核心是“标有序号的数据通道”. 数据通道强调的是数据存在和传送的物理实体以及相关的管理技术, 这与通信领域中“信道”的概念相近. 下文中的“数据位”一词均指三值光学计算机中的数据位概念.

4.2 数据位的物理形态

在三值光学计算机的不同部件中, 数据位表现出不同的物理形态且需要不同的管理技术, n 位数据位的抽象结构如图 1 所示. 其中部的细点划线框内为光学处理器, 它明显地分为上下两条通道, 包含光信号变换器的上通道称为主光路, 下通道中的位锁存器 B 和编码器 B 组成控制光路, 而重构电路和重构命令缓冲区 R_m 构成重构路径; 图中虚线框表示光学器件, 粗实线框表示电子器件, 它们下边角上标注的 1 表示可寻址单元的起始地址 (注意: 内存空间地址起始序号为 0, 二者不同. 数据位序 0 在坏位替换策略中另有用途, 见 4.4 小节), 上边角处标注的小写字母表示寻址空间的末地址; 细实线框为数据位管理策略, 花括号表示管理策略的作用位置; 箭头虚线为光信号线, 箭头实线为 8/16/32 位或 64 位电信号线, 具体值由所使用的电子计算机决定, 箭头宽线为 h 或 n 位的电信号线, h 值由三值光学计算机数据位数 n 和数据的编码格式共同决定.

图 1 中输入存储器和输出存储器位于面向用户的 PC 机存储空间中, 构成三值光学计算机一侧与用户交互数据的通道; 监控存储器位于运行三值光学计算机监控系统的电子计算机的存储空间中, 其中容有系统监控程序、支持监控功能的各种数据结构和各种例程, 这些内容构成监控系统软件, 另外也容有一部分数据位管理软件.

输入数据缓冲区 R_a 和 R_b 、输出数据缓冲区 R_c 、重构命令缓冲区 R_m 、位锁存器 A, B, C 和重构电路构成光学处理器模块中的电路部分. 其中 R_a 和 R_b 分别保存本次运算的两个输入数据 a 和 b 的 n 位数, R_c 保存本次运算结果 c 的 n 位数, R_m 中以命令集形式形成将要重构成的光学运算器映像.

图 1 三值光学计算机数据位概念和物理形态抽象图

Figure 1 The data bits concept and physical form of ternary optical computer

A 和 B 保存输入数据的编码, 同时扩充输入数据位总线的物理线数目, C 保存解码器输出数据的编码, 同时缩减输出数据位总线的物理线数目。

重构电路是实现处理器重构性的核心, 包括重构锁存器和控制信号通道两部分。重构锁存器每个字节中保持着对光学处理器一个数据位的当前重构命令, 整个重构锁存器以命令集形式给出光学处理器的当前映像。重构锁存器每个字节中的重构命令确定控制信号通道中的一个数据位通道允许通过的信号属性, 当编码器 B 输出光信号中某一位的属性符合控制信号通道中对应数据位的允许通过属性时, 编码器 B 的这一位输出就被连接到光信号变换器中对应位液晶单元的控制端, 于是编码器 B 的这一位输出就控制光信号变换器对编码器 A 这一位输出的光信号进行变换; 若编码器 B 某一位输出的光信号属性不符合控制信号通道对应数据位的允许通过属性, 则编码器 B 这一位的输出信号就不能通过控制信号通道, 于是光信号变换器对编码器 A 这一位输出光信号的变换就不受编码器 B 输出信号的控制。通过向重构锁存器中按字节写入重构命令, 软件完成重构光学处理器的任务。

编码器 A 和 B 、光信号变换器和解码器构成光学处理器模块中的光学部分。编码器 A 和 B 分别按位锁存器 A 和 B 中的编码值产生对应的三态光信号。光信号变换器根据控制信号通道送来的信号完成对编码器 A 输出光信号的变换, 生成运算结果的光信号。解码器解读运算结果的光信号, 生成运算结果的电信号编码, 然后将编码转换成数据形式送入输出数据缓冲区 R_c 。

三个位锁存器 A 、 B 和 C 面向光学处理器的端口为 h 位宽度, 并且每位连接一条独立的位线, 即以位为单位传送数据; 而面向电子计算机的端口采用电子计算机的数据总线结构, 即通过地址线寻址, 以字节为单位传送数据。这种结构将对光学处理器 n 个数据位的寻址转换成对位锁存器的位寻址, 进而转换成对位锁存器的字节寻址, 从而将光学处理器数据位寻址问题转换成常规的锁存器按字节寻址问题。在重构电路模块中, 重构锁存器起相同的作用。

由图 1 可见, n 个数据位在三值光学计算机的不同器件中和器件之间传送时的物理形态差别很大。在光信号变换器、编码器 A 和 B 中, 它的物理形态是 n 个液晶像素位¹⁾; 在解码器中, 它的物理形态是 n 个光电转换单元和 n 个阈值判断电路; 在控制信号通道中, 它是 n 条连接线, 若光信号变换器是

1) 像素位是使用三值光学处理器像素的基本单位, 包含一组液晶像素。详见参考文献 [9] 第 2239 页

光控器件, 则这 n 条连接线是光连接线, 若光信号变换器是电控器件, 则这 n 条连接线就是电连接线; 在光信号变换器与编码器 A 之间、光信号变换器与解码器之间、控制信号通道和编码器 B 之间都是 n 束三态光信号线; 在重构锁存器和重构命令缓冲区 R_m 中都是 n 个字节的半导体存储单元; 在位锁存器 A, B 和 C 中是 h 位的半导体存储单元, 其中 $h = Kn$, 由于 A, B, C 都是二值锁存器, 所以 K 是一位三值数据对应的二值编码位数, 在三值光学计算机中 $K=2$; 在位锁存器 A, B 和编码器之间、位锁存器 C 和解码器之间都是 h 条并行的电连接线; 在输入数据缓冲区 R_a , R_b 和输出数据缓冲区 R_c 中, 数据位的物理形态是 j 个字节的半导体存储器, 其中 $j = h/8 = Kn/8$; 在三个位锁存器和对应的数据缓冲区之间都是所用电子计算机数据总线宽度的电连接线; 数据位在输入存储器、输出存储器和监控存储器中没有物理形态, 这三个存储器中的数据都依照电子计算机的技术存在和使用。

图 1 表明: 用户在把原始数据送入输入存储器和从输出存储器取出计算结果的操作就是电子计算机中的访存操作. 因此, 用户在编程时只需要有“数据位数为 n ”的概念, 并以这个概念编写 n 位数据并行的程序, 而具体编程过程仍保持使用电子计算机的习惯。

当原始数据进入输入存储器后, 三值光学计算机数据位管理程序将运用数据位分配策略和系统监控软件中的坏位替换策略, 将输入数据 a 和 b 分别地分批送入输入数据缓冲区 R_a 和 R_b , 这个过程中虽然没有数据位的物理实体形态, 但已经用到了数据位概念和一些管理策略. 每批数据都以电子计算机的数据总线宽度为单位顺序送入 R_a 和 R_b , 并呈现出相应的数据位物理形态. 在 R_a 和 R_b 中, 数据的每一位都被转换成编码器需要的编码形式, 然后将编码通过电子计算机数据总线以字节为单位分别送入位锁存器 A 和 B , 虽然这个传送过程仍然是以字节为单位, 但对三值光学计算机而言, 这是以 n 位为单位的一次数据传送, 只是采用了以电子计算机数据总线宽度为单位的“串行”方式而已. 从位锁存器 A, B 输出, 经过编码器、控制信息通道、光信号变换器和解码器, 直到进入位锁存器 C , 数据都以 n 位并行方式传送. 数据从位锁存器 C 进入输出数据缓冲区 R_c 的过程又一次采用了以电子计算机数据总线宽度为单位的“串行”方式, 并在 R_c 中将数据从解码器的编码表示转换成电子计算机的数字表示形式. 最后通过电子计算机的数据总线将数据存入输出存储器, 完成 n 位数据的一次运算. 在具体的实现中, 三个数据缓冲区可以位于输入存储器或输出存储器中, 使它们之间的传送变成存储器内部搬移. 事实上, 输入存储器和输出存储器也往往位于同一个存储设备上。

4.3 数据位的管理单位

2003 年, 考虑到应该让多用户分享光学计算机的众多数据位, 应该用一组数据位构成复合运算器. 假设各个数据位和系统中的多种运算器之间通过称之为光路选择器的混洗开关阵列建立联系, 以改变数据位的运算功能、尽量提高数据位管理效率等为出发点, 作者曾提出数据位分组管理思想, 定义了算位、算道、算位类型和算道类型等数据位管理单位. 在三值光学处理器的可重构性被证实和应用以后, 必须重新审视这 4 个基本定义。

现今的三值光学计算机中只有一个位数众多的可重构光学处理器, 没有多个功能固定的运算器. 通过对光学处理器的重构, 每个数据位都可以改变运算功能, 因此光路选择器已不复存在. 但这没有改变“每个数据位的运算功能可以更改”这个基本概念, 只是更改运算功能的手段从依靠光路选择器来调换不同功能的处理器更新为依靠处理器本身的可重构性. 同时, 多用户分享众多数据位和构造复合运算器的追求没有变, 提高数据位管理效率的要求没有变. 由于这些前提没有改变, 算位、算道、算位类型和算道类型的基本概念和定义就不会有根本性改变, 但这些概念本身的一些细节和基于这些概念的一些管理策略必须适时更新. 这 4 个定义目前表述为:

算位: 一个用户的一个基本数据类型所获得的位序相连的一组数据位. 这组数据位的运算规则称为算位类型.

算道: 有限个算位的一个排列. 其各算位运算规则的排列称为算道类型.

算位与“字节”概念相近, 但可取任意值, 它们与系统的基本数据类型对应, 是分割数据位序的基本单位. 而算道则对应于复合数据类型和相应的复合运算器. 面对复杂问题, 用户常常使用自定义的复合数据类型, 如: C 程序中的“结构”类型、数据库中的“记录”、C++ 程序中的“对象”等, 它们本质上是系统提供的多个基本数据类型的一个排列. 而三值光学处理器可以在每一位上构造需要的运算器, 于是就可以对应于用户的复合数据类型构造出相应的复合运算器, 这时就要用算位来描述每一个基本数据类型的数据位数, 用算道描述复合运算器和相应的数据位数. 当用户使用复合数据类型时, 三值光学计算机就给该用户分配一组算位——算道, 再按算道类型将此算道重构成一个复合运算器, 用此复合运算器计算该用户的复合数据.

可见, 算位和算道概念用于根据用户的数据宽度来分配数据位数, 算位类型和算道类型用于根据用户的运算要求将光学处理器中对应于算位和算道的像素位重构成具体的运算器.

4.4 坏位替换技术

三值光学计算机数据位众多, 在不同的器件中数据位的物理形态各不相同, 这导致数据位的物理实体制作比较复杂, 很难保证所有的制作都精准无误. 另一方面, 三值光学计算机的数据位之间没有关联性, 尤其是采用了 MSD 加法器, 其加法运算的各数据位之间也不存在关联性. 这意味着数据位的逻辑排序可以与数据位物理实体的顺序无关. 用户程序和数据位分配策略中都使用数据位的逻辑排序——数据位序, 这个时期数据本身存在于输入存储器中. 直到数据进入输入数据缓冲区 Ra, Rb, 或重构命令进入 Rm, 数据位才落实到具体的数据位物理实体, 而当计算结果离开计算结果缓冲区 Rc 时, 数据位又脱离了具体的数据位物理实体, 数据本身进入存储器实体. 数据位落于和脱离数据位实体的操作由系统监控软件中的坏位替换策略和技术来完成, 对用户和上层策略不可见.

坏位替换技术有 3 个主要操作, 一是及时发现不正常的数据位实体, 二是用预留的正常数据位实体备份替换不正常的实体, 三是当数据位实体备份用完时通知管理员降低系统的数据位总数 n 的值, 以延长系统整体上的使用寿命.

实现坏位替换策略的核心是对每个数据位物理实体当前状态作记录, 记录表的概念结构如表 1, 共有 3 列 $n+m$ 行, m 是物理实体的冗余量, 数字 0 用于表示实体不正常. 表的第一列为数据位序, 从 1 开始按增序列出, 直到 n , n 以后为冗余的或被替换的数据位序号. 第二列为实体序, 按与数据位序的对应排列, 实际中总是尽量与数据位序号相同, 如表中第一行和第 n 行所示. 第 3 列是替换指针, 标注用于替换的冗余序号. 如果第 p 号实体不正常, 则在第 p 行的第二列中写 0, 同时第 p 行的第 3 列上给出替换实体的序号, 这个序号一定大于 n 值, 用 $n+i(i \leq m)$ 表示, 同时第 $n+i$ 行的第一列标出发生替换的数据位序号 p . 实践中第 2 列可以隐去.

坏位替换成功与否决定于表 1 中对实体状态记录的是否正确, 因此要保证及时发现不正确的实体位. 这项技术称为数据位实体测试, 其概要为²⁾: 1) 将标准数据送入 Ra 和 Rb; 2) 将光学处理器重构为预定的典型运算器; 3) 将 Rc 中的计算结果同理论预测值对比, 以发现不正常的数据位实体; 4) 将表 1 第 2 列中不正常实体序的位置清零; 5) 将光学处理器重构为另一个预定的典型运算器, 重复 3) 和 4) 的操作, 直到所有预定的典型运算器都被测试; 6) 考察表 1 中实体序号大于 n 的行, 若其第二列

2) 第 4.2 小节和图 1 中的 n 在这里与 $n+m$ 对应

表 1 数据位序与实体序对应表

Table 1 Correspondence between the data bit sequence and the entity sequence

Data bit sequence	Entity sequence	Replacement pointer
1	1	
.....		
p	0	$n + i$
.....		
n	n	
p	$n + i$	
.....		
0	0	
$n + m$	$n + m$	

为 0, 则将其第一列清零, 如表 1 中倒数第 2 行所示.

为了及时更新表 1, 每当系统重启、受到较强震动、受到较强的电源波动、发现计算错误时都要立刻自动进行本项测试, 即使在非常好的工作环境中, 每隔一段时间, 如一小时, 也会自动测试一次, 参考文献 [16] 以“坏点剔除”的词目给出了针对光学处理器的数据位实体测试流程和实验.

每当测试完成后, 即可对表 1 前 n 行中第 2 列为 0 的行进行替换操作: 设第 p 行的第 2 列为 0, 在序号大于 n 的范围内找到一个第一列的值大于 n 的行, 设其行序为 $n + i$; 将 $n + i$ 拷贝到第 p 行的第 3 列, 再将 p 值拷入第 $n + i$ 行的第一列.

当替换操作在表 1 中找不到可用的冗余行时, 表示系统的可用数据位已经达不到 n 位, 这时监控程序会停止系统工作, 并通知系统管理员目前可用数据位的最大值. 如果管理员同意降低数据位数, 他将在重启系统时更改 n 值, 或检修、更换部件. 任何设备随着时间推移都会老化和局部损坏, 利用降低 n 值的方法, 可以延长系统剩余性能的使用寿命.

在将数据位序落于和脱离数据位实体时, 都按照表 1 第 1 列和第 2 列的对应关系操作, 保证每个数据位序都在正常的实体上运行.

依靠坏位替换策略可以大幅度降低对设备制造精良度的要求, 从而降低制造成本并提高设备的可用性.

4.5 整体重构技术

为了提高系统的整体效率, 作者在 2009 年提出了对光学处理器的定时重构策略³⁾, 并开发了相应的技术^[15]. 这个策略有两个特征: 1) 仅当运算时间达到预定的时长 T_d 的整数倍时才能对光学运算器进行重构; 2) 每次重构都对光学处理器的全部数据位进行.

然而, 在 2011 年完成的用户编程输入界面中, 用户送入的数据总量被实时统计, 于是当原始数据进入三值光学计算机系统时, 数据位管理程序就能知道本次计算需要多少次操作. 基于此, 定时重构策略演变为目前的整体重构策略: 仅当运算次数达到预定值 N_y 时才能对光学运算器的整体 (即全部数据位) 进行重构. 实现这个策略只需要做到 3 点: 1) 在运算次数达到预定值 N_y 时重构命令才能生

3) 文献 [15] 中称为像素定时映射策略

效; 2) 重构命令的操作把重构命令缓冲区 R_m 中的重构命令集送入重构锁存器; 3) R_m 中的重构命令集是光学运算器下一次将被重构成的整体状态的映像。

由于重构是对光学处理器的整体统一进行, 隐含着在重构时自动收回全部数据位, 于是系统不再需要回收数据位的策略和技术。

由于每次重构操作都是把 R_m 的全部内容写入重构锁存器, 因此每次重构的用时都一样长。 R_m 和重构锁存器之间的数据总线越宽, 重构过程就越快。

整体重构技术需要的硬件有: R_m 寄存器、重构锁存器、预定值 N_y 计数器和重构命令标志位。主要操作步骤为: 数据位分配技术将其生成的光学处理器重构命令集逐步写入 R_m ; 当命令集在 R_m 中写入完成后, 数据位管理系统发出允许重构命令, 使重构命令标志位有效; 当 N_y 计数完成且重构命令标志位有效时, 执行重构命令; 当重构操作执行完毕, 系统更新 N_y 计数器的预定值, 然后启动计数器并使重构命令标志位无效; 无效的重构命令标志位会通知数据位分配技术更新 R_m 的内容; 若 N_y 计数完成, 而重构命令标志位无效, 表示系统中已经没有待计算的数据, 则光学处理器进入休眠状态。可见, 重构命令标志位是协调数据位分配技术和整体重构技术的关键, 它由允许重构命令置 1, 由启动 N_y 计数器的命令清零, 被重构命令查看, 因此它是协调指令执行次序的硬件设施, 虽然它与数据位管理有关, 但不直接关联数据通道, 故在图 1 中没有标示出来。另外, 系统的初始化程序通过设置重构命令标志位和设置 N_y 计数器为计数完成状态来启动第一次重构过程。

4.6 数据位分配技术

数据位分配和回收技术是最早被研究的数据位管理课题, 曾建立了静态分组、静态分配和回收策略与技术^[10~14]。然而, 随着整体重构技术的确立和算位、算道概念的更新, 这些策略和技术已经有了诸多改变。目前条件下, 数据位回收策略已不再需要, 数据位分配策略演变为生成 R_m 中重构指令集的策略。

数据位分配策略追求的目标在于两点: 1) 使尽量多的计算任务在可接受的时间内完成; 2) 使系统资源得到尽量高的利用率。前一目标要求对计算量不同的任务使用不同的分配策略, 这就需要一个指标来判断各任务的计算量规模。后一目标也需要一个指标来判断资源利用率, 考虑到目前与数据位有关的最贵重资源是光学处理器, 因此这个指标目前应以光学处理器中数据位的利用率为基础。显然, 资源利用率越高, 完成计算的用时就越短, 所以这两个指标有内在联系, 我们以资源利用率为基础来统一设立。

表征光学处理器利用率的直观方法是平均单位时间内数据位的有效使用比例。它包含一次运算中有效使用的数据位数目占总数据位数的比例和处理器重构用时占总工作时间的比例两个方面。设运算器重构操作需要用 q 个计算周期, 两次重构之间完成 N_y 次计算, 数据位总数为 n , 有效使用的数据位数为 m' 。则数据位的有效使用比例 B 为

$$B = m' \times N_y / ((N_y + q) \times n), \quad (1)$$

其中 $m' \times N_y$ 为光学处理器在本次重构后完成的计算量。相应地将任务的算道宽度 m 与数据个数 N 之积称为该任务的计算量。

从 (1) 式可见, 当 N_y 不远远大于 q 时, 利用率会被 q 制约, 这表明计算量很少的任务不适合在三值光学计算机上运行。

当 N_y 远远大于 q 时, 利用率就决定于 m'/n 的值. 实际中, 很少有任务的算道宽度 m 接近 n , 这意味着提高 m' 值的途径在于把不同用户的算道连接起来, 使各算道的宽度之和形成的 m' 接近于 n , 这体现了多用户分享数据位数的思想. 然而, 将不同的算道连接后, 意味着 R_a , R_b 和 R_c 中不同位上的值分属于不同的任务, 于是必须花费额外资源来整理用户数据. 另外, 各算道要计算的数据量不可能完全相同, 当数据量较少的算道完成工作后, 它所占用的数据位进入无效操作, 直到计算量最大的同伴算道完成工作, 这又造成一种浪费.

综合上述考虑, 我们按任务的计算量将任务分为大、小和微 3 类, 根据各类任务的特点分别采用不同的数据位分配策略. 当任务的计算量 (任务的算道宽度 m 与数据个数 N 之积) $> 1000 [n/m]$ 时称为大任务 ($[n/m]$ 表示取商的整数部分, 其值是将 n 个数据位分割成 m 宽度算道的个数), 当任务的计算量 $< 100 [n/m]$ 时称为微任务, 当 $100 [n/m] < \text{任务的计算量} < 100 [n/m]$ 时称为小任务. 若任务的计算量在 $1000 [n/m]$ 附近时, 倾向于定为大任务, 也可定为小任务; 若任务的计算量在 $100 [n/m]$ 附近时, 倾向于定为微任务, 也可定为小任务.

大任务的数据位分配策略比较简单. (1) 在处理器上构造 $[n/m]$ 个 m 宽度的算道, 相当于把光学处理器分成 $[n/m]$ 个相同的复合运算器; (2) 在 R_a , R_b 和 R_c 中只使用 1 到 $[n/m]$ 的位, 相应地在数据位空间中也只使用 1 到 $[n/m]$ 位序空间; (3) $N_y = 1 + \text{任务的数据个数} / [n/m]$. 这种分配策略造成的数据量浪费有两部分, 一是 $n - [n/m]m$ 个数据位始终没有用到, 二是最后一次计算时数据个数不足 $[n/m]m$ 个. 如果任务需要的不是复合运算器, 而是基本类型运算器, 则 m 就成为算位宽度, 由于算位内的数据位之间没有排列关联性, 可以按位分开, 于是可以将光学处理器构造成为 n 位的简单类型运算器, 这时数据位的利用率达到最高状态.

小任务的数据位分配策略较为复杂. 首先, 将数据个数相近的任务归为一组, 然后将同组任务的算道拼连在一起, 当算道的总位数接近 n 时, 调度这些任务同时运算, 他们共享数据位数. 这时的 N_y 选取该组任务中数据个数最多的值. 如果小任务足够多, 算道拼连后的总宽度就可能很接近 n , 而且各任务的数据量也相差无几, 这时光学处理器的利用率就很高. 这个分配方案的复杂性在于对 R_a , R_b 和 R_c 中数据的整理. 一个可行的方案是给 R_a 和 R_b 各配备一个规模适度的前置缓冲区, 将组内各任务的数据按其算道的拼连次序在前置缓冲区中拼装好, 然后再调度这组任务进行运算, 待结果进入输出存储器后, 再对其拆分, 形成各任务的结果. 这个方案是用电子计算机的资源耗费换取光学处理器的资源节约.

微任务不在光学处理器中运行, 会被调度到协同工作的电子计算机上模拟执行.

大任务或小任务数据位分配完成的标志是生成了算道拼连图像和 N_y 值, 它们被排入 R_m 等待队列, 当轮到其使用光学处理器时, 系统按相应的算道拼连图像将其包含的每个数据位的重构命令写入 R_m , 将 N_y 值写入 N_y 计数器, 等待光学处理器整体重构操作的发生.

5 数据位管理理论在 SD11 中的实践

5.1 SD11 中数据位寻址总线的设计和实现

SD11 数据位数的设计目标为千位量级, 而图 1 中给出的各锁存器和缓冲区都使用常规半导体器件, 且由电子计算机访问, 所以它们的位数应以 2 的整数幂为好. 综合这两点考虑, 设计 SD11 基本型的数据位数 $n=1024$ 位. 于是 3 个位锁存器 A , B 和 C 都有 $h=2048$ 位, 从面向电子计算机的端口看, 都为 256 个字节; 相应地, 3 个数据缓冲区 R_a , R_b 和 R_c 也都为 256 个字节, 而重构命令缓冲区 R_m

图 2 SD11 数据位实现结构图

Figure 2 The implementation diagram of SD11 data bits

和重构锁存器各为 1024 个字节。

另一方面, SD11 选用的液晶模块有 576 个像素, 于是光学处理器的每个模块可有 576 个数据位, 故此, SD11 基本型使用两个这样的液晶模块构造一个光学处理器, 可有 1152 个数据位, 将其中的 128 个数据位用于冗余备份, 保证 1024 个有效数据位。考虑到数据位的冗余备份, 设计位锁存器 A 、 B 和 C 都为 2304 位, 组成 288 个字节, 则输入数据缓冲区 R_a 和 R_b 、输出数据缓存区 R_c 也为 288 个字节, 而重构命令缓冲区 R_m 和重构锁存器各为 1152 个字节。按照图 1, 对位锁存器 A 、 B 、 C 和数据缓冲区 R_a 、 R_b 、 R_c 按字节寻址时需要 9 条地址线, 而对重构命令缓冲区 R_m 按字节寻址时, 需要 11 条地址线。

在数据位管理理论的指导下, 考虑到目前可用元器件的实际情况和 SD11 的设计目标, 从图 1 派生出的 SD11 数据位实现结构如图 2 所示。这个结构在整体上可分为 3 个部分: 位于点直线右侧, 以 PC 为核心的电子计算机部分; 位于点折线左侧, 以光信号变换器为核心的光学处理器部分; 位于两条点线之间, 以 DAB 控制器 (数据位寻址总线, data-bit address bus) 为核心的地址属性变换部分。

与图 1 相比, 图 2 中光学处理器部分增加了解码器 ck 和位锁存器 ck 模块, 二者被设置在嵌入式系统 b 中。增加这个模块的原因是光信号变换器中选用的液晶阵列是电控型元件, 必须把编码器 B 输出的光信号转换成对应的电信号, 才能用于对光信号变换器中液晶像素的控制。显然, 也可以将编码器 B 、解码器 ck 和位锁存器 ck 同时拿掉, 使位锁存器 B 中的电信号通过控制信号通道来控制光信号变换器中液晶阵列各像素的旋光性, 但是, 考虑到 SD11 仍是一个实验系统, 为使其也适合对光控型液晶器件进行实验研究, SD11 中选择了增加解码器 ck 和位锁存器 ck 的结构。

在光学处理器部分, 位锁存器 A 和 B 分别制作在 FPGAb 和 FPGAb 中, 重构电路 (包括重构锁存器和控制信号通道) 制作在 FPGAc 中, 位锁存器 C 和输出数据缓冲区 R_c 设置在嵌入式系统 a 中。FPGAb、FPGAb、FPGAc、嵌入式 a 和嵌入式 b 中都设置有 11 位的数据位寻址总线 (DAB) 和 8 位的数据总线接口, DAB 用于对 FPGA 或嵌入式系统内部的位锁存器或数据缓冲区以字节为单位寻址, 数据总线用于传送字节数据。FPGAb、FPGAb、FPGAc、嵌入式 a 和嵌入式 b 之间用 11 条数据位寻址总线 DAB_{0-10} (粗实线) 和 8 位数据线 (点划线) 相连接, 并通过接插座 0 与地址属性变换部分相连接。FPGAb 和 FPGAb 中的位锁存器只有 288 个字节, 相应地只需要 9 条数据位寻址线, 但为了使它

们能同 FPGA_c 在电路板的层面上互换, 也给它们设计了 11 条数据位寻址线.

PC 中设置了数据缓冲区 Ra, Rb 和 Rm, 并通过 USB 总线与地址属性变换部分相连接.

地址属性变换部分在面向 PC 的一面设计了 USB 接口, 通过 USB 总线与 PC 通信. 从 PC 接收到的数据和即将发送给 PC 的数据都在数据缓冲区中暂存. 面向光学处理器的一面设计了 8 位的数据总线和 15 位的 DAB 总线, 其中 DAB₁₁₋₁₄ 为光学处理器扩充硬件接口, 经过 4-16 译码器, 寻址 16 个接插座, 每个接插座上有 DAB₀₋₁₀ 和 8 位的数据总线, 于是, 每个接插座可接一个像素数目不超过 2048 位的光学处理器硬件. 目前 SD11 使用的光学部件只有 1152 个像素, 其有效数据位数为 1024, 于是, SD11 的有效数据位数最大可扩充到 16384 位, 外加 2048 位的冗余数据位.

5.2 数据位管理软件在 SD11 中的实现

SD11 监控软件的自检程序中有坏位检测模块, 该模块生成并更新表 1 所示的数据位序与实体序对应表, 这是坏位替换技术的基础; SD11 的数据传送程序中有位锁存器内数据迁移模块, 这个模块根据表 1 将位锁存器 A 和 B 中坏位位置上的数据迁移到对应的替换位上, 把锁存器 C 中位于冗余位上的数据迁移到原数据位序上; SD11 的重构处理器程序中有重构命令缓冲区内数据迁移模块, 该模块在 Rm 被写完成后, 允许重构命令发出前, 按表 1 的指示将 Rm 中坏位地址上的数据迁移到替换位上. 这 3 个软件模块配合实现坏位替换技术.

SD11 的用户数据接收程序每收到一个任务立刻计算它的计算量, 根据任务的计算量, 将其列入大任务或小任务群, 而将微任务调入协同工作的 CPU 中; 在大任务群和小任务群中再根据任务的优先级排队. 数据位分配程序按照优先级高的任务先处理, 同等优先级时大任务先处理的策略, 从大任务队列或小任务队列中提取各任务的算道类型, 按照各自的策略排列算道类型成为算道类型帧, 将算道类型帧送入 Rm 等待队列, 完成一次数据位分配.

SD11 的重构处理器程序按照 4.5 小节描述的工作过程和功能模块编制, 并包含 Rm 内数据迁移模块. 配合这个程序的执行, 在监控系统 CPU 的寄存器中设立一个 16 位的 N_y 计数器和 1 位的重构命令标志位, 每从输出数据缓冲区 Rc 读出一个计算结果, 使 N_y 减 1, N_y 减到 0 时启动重构过程. 对重构命令标志位的操作见 4.5 小节, 当它被清 0 时, 这个程序就从 Rm 等待队列中取一个算道类型帧, 然后调用重构例程, 按照算道类型帧将重构命令调入 Rm, 由此在 Rm 中生成光学运算器的重构命令映像, 再调用 Rm 内数据迁移模块进行坏位替换, 最后置位重构命令标志位. 至此完成重构光学处理器的准备工作.

当一个任务所在的算道类型帧被重构处理程序取用后, 这个任务的数据从外部存储器调入输入存储器 (数据管理 CPU 的内存), 如果是一组小任务, 则开始进行数据整理, 待光学处理器重构完成后, 这些数据分批进入 Ra 和 Rb, 并在 Rc 处得到计算结果.

6 结论

三值光学计算机数据位数众多的特征将带来新的算法思路, 同时带来了数据位管理课题. 在三值光学计算机应用研究实验系统的构建过程中, 作者对众多数据位的管理问题进行了比较系统的研究. 在继承、更新和增补了数据位管理方面的基本概念和基本策略的基础上, 本文形成了比较系统的数据位管理理论和技术架构. 作者用这个理论指导了三值光学计算机实验系统 SD11 的硬件设计工作和相关软件的设计, 使这个理论在实践中得到了验证和应用.

参考文献

- 1 Intel. Intel® 64 and IA-32 Architectures Software Developer's Manual. 2010.12. <http://download.intel.com/design/processor/manuals/253665.pdf>
- 2 Peter N. Glaskowsky, NVIDIA's Fermi: The First Complete GPU Computing Architecture, A white paper of NVIDIA. 2009. 9. http://www.nvidia.com/content/PDF/fermi_white_papers/P.Glaskowsky_NVIDIA's_Fermi-The_First_Complete_GPU_Architecture.pdf
- 3 Jin Y, He H C, Lu Y T. Ternary optical computer principle. *Sci China Ser F-Inf Sci*, 2003, 46: 145–150
- 4 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
- 5 Jin Y, He H C, Lu Y T. Ternary optical computer architecture. *Phys Scripta*, 2005, T118: 98–101
- 6 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
- 7 Jin Y. Draw near optical computer. *J Shanghai Univ (Nat Sci)*, 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]
- 8 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
- 9 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
- 10 Jin Y. Ternary optical computer principle and architecture. Dissertation for the Doctoral Degree. Xian: Northwestern Polytechnical University, 2002. 78–81 [金翊. 三值光计算机原理和结构. 博士论文. 西安: 西北工业大学, 2002. 78–81]
- 11 Jin Y. Management strategy of data bits in ternary optical computer. *J Shanghai Univ (Nat Sci)*, 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]
- 12 Zhan X Q, Peng J J, Jin Y, et al. Static allocation strategy of data-bits of ternary optical computer. *J Shanghai Univ (Nat Sci)*, 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]
- 13 Zhan X Q. Static allocation strategy of data-bits of ternary optical computer. Master Thesis. Shanghai: Shanghai University, 2009. 4–34 [詹小奇. 三值光计算机数据位资源的静态分配策略. 硕士论文. 上海: 上海大学, 2009. 4–34]
- 14 Zhang Z Y. Recovery strategy and implementation of data-bits of ternary optical computer. Master thesis. Shanghai: Shanghai University, 2009. 13–37 [张赵云. 三值光计算机数据位回收策略及其实现. 硕士论文. 上海: 上海大学, 2009, 13–37]
- 15 Li S F, Jin Y. A mapping technology from components-pixels to data-bits of ternary optical computer. *Comput Eng Des*, 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]
- 16 Jin Y, Gu Y Y, Zuo K Z. Theory, technology and achievement of TOC's decoder. *Sci China Inf Sci*, 2013, 43: 275–286 [金翊, 顾莹莹, 左开中. 三值光学计算机解码器的理论、技术和实现. 中国科学: 信息科学, 2013, 43: 275–286]

Management of many data bits in ternary optical computers

JIN Yi*, OUYANG Shan, SONG Kai, SHEN YunFu, PENG JunJie & LIU XueMin

School of Computer Engineering and Science, Shanghai University, Shanghai 200072, China

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

Abstract Numerous data bits is one outstanding feature of ternary optical computers. How to effectively manage these data bits is a new topic in computer science. This paper forms a data bits management theory and technical framework through inheriting, updating, supplementing, and searching the research achievements

in this field. The framework includes the following basic concepts and techniques. 1) “Calculating path” and “calculating channel” are the units of data bit management. The calculating path is used for simple data types and the channel for composite data types. 2) “Index of calculation amount” categorizes the scale of a task. Large, medium and small scale tasks will be managed with different strategies. 3) “Physical forms of data bits” identifies the devices occupied by data bits in each link of a data transmission. 4) A technique of “replacing an effectively lost data bit” will replace all disabled data bits with reserved redundancy data bits in a timely manner. Incidentally, the ternary optical computer has tolerant capability for software and hardware faults in data bits. 5) “Overall reconfiguring” reduces the number of reconfiguring actions that interrupt computing, by reconfiguring all the data bits at regular times. 6) “Data bits distribution technique” will employ different schemes for each scale task when distributing data bits. This paper also records the guidelines in the theory for designing a third generation experimental system for a ternary optical computer, named the SD11. The design goal of this experimental system is to research the applications of ternary optical computers. In SD11, a fundamental unit of the optical processor has 1024 data bits and 128 redundant bits, and one optical processor can possess 16 fundamental units in all. Thus, at most, one SD11 can have 16384 data bits and 2048 redundancy bits.

Keywords data bits management, ternary optical computer, reconfigure, allocating data bits, calculating path, calculating channel

JIN Yi was born in 1957. He received his 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. Dr. JIN is a senior member of China Computer Federation and ACM.

OUYANG Shan was born in 1979. He received his Ph.D degree in computer application technology from the School of Computer Engineering and Science, Shanghai University, Shanghai in 2012. His research interests include ternary optical computers and embedded systems.

SONG Kai was born in 1980. He received his master's degree in computer science from the East China Jiaotong University, Nanchang in 2007. Currently, he is a Ph.D. candidate at the School of Computer Engineering and Science, Shanghai University. His research interests include ternary optical computers and embedded systems.

SHEN YunFu was born in 1960. He received his Ph.D. degree in mathematics from Beijing Normal University, Beijing in 1996. Currently, he is a Senior Researcher at Shanghai University. His research interests include formal methods in software and hardware, model checking, and the operators and reliability of ternary optical computers.