

论 文

三值光学计算机的 SRT 除法算法和实现技术

徐群^{①②}, 金翊^{①*}, 沈云付^①, 李卫民^①, 韩越兴^①

① 上海大学计算机工程与科学学院,

② 国网青岛市黄岛区供电公司, 青岛 266400

* 通信作者. E-mail: yijin@shu.edu.cn

收稿日期: 2015-05-05; 接受日期: 2015-05-21

国家自然科学基金青年基金项目 (批准号: 61103054), 上海市自然科学基金项目 (批准号: 13ZR1416000), 上海市教育委员会科研创新项目 (批准号: 13ZZ074) 和上海市教育委员会科研创新项目 (批准号: 13ZZ074)

摘要 本文建立了三值光学计算机第一个除法例程的算法, 设计了该例程的操作步骤. 该算法根据三值光学计算机数据位众多、按位可分配和每一位的计算功能可随时配置等特点, 选用 SRT 除法计算方法为数学模型. 该算法设计的后台模块分别与三值光学计算机监控软件中的任务管理模块、三值光学处理器重构模块和三值光学处理器的底层硬件控制软件相融合, 构成了软硬件结合的计算例程. 实现 SRT 除法算法的例程将纳入三值光学计算机的底层软件系统, 供应用程序调用. 文中详细讨论了这个算法需要的三值光学处理器的数据位数、对处理器的重构策略、与三值光学计算机监控软件各部分的协作方式、迭代次数和商的小数点位置, 并对相应例程的工作过程做了详尽描述. 本文还介绍了模拟实验系统的设置和实验用例, 并通过模拟实验证实了实现该算法的例程的有效性. 本文提出用有效数字位数作为确定用户所需数据位资源的依据, 并以此来确定 SRT 计算方法中的迭代次数. 本文描述的工作为开发三值光学计算机其他常用计算例程提供了技术路线和例证.

关键词 三值光学计算机 可重构处理器 SRT 除法 除法例程 MSD 加法器

1 引言

加减乘除是最常用的算术运算, 且减法、乘法和除法都可以转化为加法来实现. 因此在早期的电子计算机中只构造了硬件加法器, 而把减法通过将减数取补码后在加法器中完成, 乘法和除法则用软件将其转换成一系列的加 (减) 法运算, 然后用加法器来完成. 直到电路集成度达到数百万个晶体管后, 才有了用硬件完成乘法的实例. 然而, 用硬件完成除法运算仍然有很多困难, 比如硬件复杂度太高、算法造成的时间延迟太长等. 长期以来, 人们对除法的计算方法和实现过程进行了不断的优化, 先后发明了 Newton-Raphson 计算方法, Talyor 展开法^[1] 和 SRT 除法^[2] 等. 在这些计算方法中, SRT 除法通过设置不同的参数表现出了较灵活的实现方案, 并能藉此降低实现除法的硬件复杂度, 故受到除法器研究者的重视. 然而 SRT 除法是一个迭代过程, 这决定了完全依靠硬件实现 SRT 除法非常困难, 构造一种软硬件结合的实现 SRT 除法算法的例程是常

三值光学计算机 (ternary optical computer, TOC) 具有数据位数众多、数据位数易增加、数据位资源可按位分配和光学处理器 (称为可重构) 等特色^[3]. 例如这种计算机的最

引用格式: 徐群, 金翊, 沈云付, 等. 三值光学计算机的 SRT 除法算法和实现技术. 中国科学: 信息科学, 2014, 44: 1-?, doi: xxxxxxxx

新一代实验系统——SD11的数据位数已经达到1024位,且可扩展至16384位^[4],它的任一个数据位都可以同其他数据位分开使用,并构造成二值(一位)逻辑运算器^[5,6].再以处理器的可重构性和数据位数众多为基础,三值光学计算机构造出了MSD数并行加(减)法器^[7,8],进而建立了乘法例程.这些工作为构建三值光学计算机实现SRT除法算法的例程奠定了基础.

本文介绍利用三值光学计算机数据位数众多、三值光学处理器按位可分配和按位可重构3项特点,结合SRT除法,为这种新型光电计算机实现一款除法例程.同时,该例程具有迭代过程并行度较高和可以将多对数据并行计算的优点,因此它也加速了SRT除法的计算过程.

2 适合三值光学计算机的SRT除法

与电子计算机不同,三值光学计算机将根据用户的需要为其分配运算器的数据位数,这不仅节约计算资源,而且能使计算结果更适合用户的实际需要,减轻用户对计算结果的后期修正,但获得这些优势的代价是用户必须同原始数据一起给出对计算结果的位数需求.鉴于在科学实验、工程测量和实际问题计算中,普遍采用“有效数字”来表示数值数据,三值光学计算机也相应地采用有效数字来表示原始数据和计算结果,并依此给用户分配数据位数.

有效数字一般被定义为所有准确值和最后一位不准确值形成的数值.从仪器上读数据时,对仪器最小刻度之后的值只能估计出一位,所以工程、实验和生活中读到的原始数据都是有效数字.有效数字的最后一位表征了实际数据的精度.有效数字进行数学计算得到的结果也应该为有效数字,即计算结果的最后一位是不准确值,它是原始数据的不准确性所致结果误差的最高位.本文将有效数字来讨论SRT除法算法.

为消除进位过程带来的加法运算延时,三值光学计算机采用了MSD数二进制并行加法器,这决定了三值光学计算机内部的数值数据都是MSD二进制数据.三值光学计算机在其输入界面将用户输入的十进制数据转换成MSD二进制数据,在输出界面又将数据转换成十进制数据.因此在讨论三值光学计算机完成SRT除法的内部操作时,本文使用有效数字的MSD二进制数据.

众所周知,两数相除的运算关系可以用乘法形式表示如下:

改为:MSD数字除法

$$N = D \times Q + W,$$

其中, N 为被除数, D 为除数, Q 为商, W 为余数,余数需要满足条件 $0 < W < D$.

根据除法的数学原理,当 N 与 D 同号时,商为正数,本文用符号位 $sign=1$ 表示;当 N 与 D 异号时,商为负数,用 $sign=-1$ 表示.在这个约定下,本文中只讨论 N 和 D 都是正数的情况而不失一般性.

设被除数 $N(N_n \dots N_{k+1}.N_k \dots N_1)$ 是 n 位有效数字的MSD二进制数,其中有 k 位小数;除数 $D(D_m \dots D_{x+1}.D_x \dots D_1)$ 是 m 位有效数字的MSD二进制数,其中有 x 位小数.同时要求 N 和 D 的值不能超过三值光学处理器可表示的最大值,并不允许 D 的输入值为0.除法运算过程中需要对数据进行必要的调整:①运算前对被除数和除数执行左移 x 位操作,以将除数 D 调整为 m 位的整数,相应地被除数 N 有 $k-x$ 位小数;当 k 小于 x 时,给 N 后补 $x-k$ 个0,并在尾部加小数点;②运算过程足时,在其后部补零;③商数 Q 的小数点位置与调整后的被除数 N 的小数点位置对齐,小数点位置用 P 记录.

SRT除法是一个迭代过程,迭代的次数由商数的二进制有效数字位数 F 决定.而用户给出的商数的有效数字位数是商的十进制数形式的位数,记为 G ,其中前 $G-1$ 位是准确值,最后一位是不准确

值 (估计值). 本例程将从 $G-1$ 来计算商的 MSD 二进制数据的准确值位数 $F-1$, 然后再在最后增加一位不准确值, 形成商的 MSD 二进制数据的有效数字,

$$F = (F - 1) + 1 = \lceil (G - 1) \times \ln 10 \div \ln 2 \rceil + 1.$$

三值光学计算机任务管理软件中的数据位分配模块将根据 F 值为用户分配数据位数, 实现计算机资源的按需分配.

因此, 本文中约定除数 D 、被除数 N 、计算过程中的中间数据、计算直接得到的商数 Q 和余数 W 都为 MSD 二进制 改为: MSD 数字除法 的符号集为 $\bar{1}, 0, 1$ (其中 $\bar{1}$ 表示 -1), 所以 D 、 N 、 Q 、 W 和中间数据都是三值形式的数据; m 、 n 、 x 、 k 和 F 分别为不同数据的有效数字位数, 故它们本身都是十进制数. 在这种情况下, SRT 除法 的第 i 次迭代生成的商位 Q_i 和部分余数 W_i 的计算方法如下, 其中 $i = 1, 2, \dots, F$.

Q_i 的计算公式为式 (1), 其核心是比较前一次迭代的部分余数 W_{i-1} 与除数 D 及其相反数 $-D$ 的大小 ^[9, 10]. 迭代运算结束时, 依次连接各次迭代得到的商位 $Q_1 \dots Q_i \dots Q_F$, 根据 $sign$ 值给商位前端加 改为: 将数字迭代法扩充到 MSD 数值域后, 根据不恢复余数除法的商数确定法则可以写出公式 (1), 计算 Q_i 的核心 商位中加上小数点, 即可获得商数 Q .

$$Q_i = \begin{cases} \bar{1}, & W_{i-1} \leq -D, \\ 0, & -D < W_{i-1} < D, \\ 1, & W_{i-1} \geq D. \end{cases} \quad (1)$$

部分余数 W_i 的计算分两步: 先由式 (2) 确定出中间变量 U_i , 这一步的核心操作仍然是分别比较 W_{i-1} 与 D 、 W_{i-1} 与 $-D$ 的大小; 然后在 U_i 后接续上被除数的第 $(m+i)$ 位得到 W_i . 但 W_0 为被除数 N 的高 m 位, 而将 U_F 调整后得到最终余数 W [11, 12] 改为: [9, 10]

$$U_i = \begin{cases} W_{i-1} + D, & W_{i-1} \leq -D, \\ W_{i-1}, & -D < W_{i-1} < D, \\ W_{i-1} - D, & W_{i-1} \geq D. \end{cases} \quad (2)$$

考察式 (1) 和 (2) 可知: 计算 q_i 和 U_i 的核心操作是将 W_{i-1} 分别与 D 和 $-D$ 比较大小, 这可以通过分别判读 $W_{i-1} - D$ 和 $W_{i-1} - (-D)$ 的正负来实现; 同时, 确定 U_i 的值也要进行 $W_{i-1} + D$ 或者 $W_{i-1} - D$ 运算. 故对于 MSD 数的二进制 SRT 除法 改为: 数字除法 每次 做一次 $W_{i-1} + D$ 和一次 $W_{i-1} - D$ 运算. 设前者的运算结果记为 WA_i , 后者的运算结果记为 WS_i , 显然, 从 WA_i 和 WS_i 的正负性即可确定 W_{i-1} 分别与 D 和 $-D$ 的大小, 进而确定 Q_i 和 U_i . 判断规则如下:

当 WA_i 不大于 0 时, 有 $Q_i = \bar{1}, U_i = WA_i$;

当 WA_i 大于 0, 且 WS_i 小于 0 时, 有 $Q_i = 0, U_i = W_{i-1}$;

当 WS_i 不小于 0 时, 有 $Q_i = 1, U_i = WS_i$.

由于三值光学计算机的数据位数众多、可以按位分配且每一位的运算功能可以被随时重构, 因此, 可以在三值光学处理器的不同数据位数上分别构造出一个加法器 A 和一个减法器 S, 在第 i 次迭代中, 可将 W_{i-1} 和 D 同时送至加法器 A 和减法器 S 进行并行计算, 于是同时获得 WA_i 和 WS_i , 使原本在电子计算机中顺序完成的加法和减法运算优化成并行完成, 从而减少每次迭代的运算时间, 加速

SRT 除法. 改为: MSD 数字除法

3 在三值光学处理器上构造加法器 A 和减法器 S

3.1 加法器 A 和减法器 S 的数据位数分析

第一次迭代运算过程中, W_0 和 D 是两个 m 位的 MSD 二进制数, 送至加法器 A 和减法器 S 上运算, 由 MSD 加 (减) 法器运算规律可知, 运算结果 WA_1 和 WS_1 的数据位数均为 $m+2$ 位的 MSD 二进制数. 根据第 2 节所列的判断规则, U_1 的取值集合为 $\{WA_1, WS_1, W_0\}$, 所以 U_1 的位数最多为 $m+2$ 位的 MSD 二进制数, 于是 U_1 拼接上被除数 N 的余下部分的最高位后生成的 W_1 最多有 $m+3$ 位的 MSD 二进制数. 同理, 每次迭代得到的部分余数的最多位数都比迭代前多出 3 位, 于是, 第 i 次迭代后, 部分余数 W_i 的位数最多为 $m+3i$ 位的 MSD 二进制数. 最后一次迭代时, 将 $m+3F-3$ 位的部分余数 W_{F-1} 和 m 位的除数 D 送至加法器 A 和减法器 S 上运算, 得到的中间变量 U_F 不再执行接续操作, 故最终余数 W 的位数最多为 $m+3F-1$ 位的 MSD 二进制数.

3.2 构造加 (减) 法器需要的光学处理器数据位数量

根据 3.1 节的分析, 在迭代过程中, 加法器 A 和减法器 S 的输入数据从 m 位 MSD 二进制数增加到 $m+3F-3$ 位. 考虑到修改加 (减) 法器的位数需要重构处理器, 这时运算过程必须暂停等待, 为避免计算过程的断续, 本例程采用在运算之前构造出所需最多位数的 A 和 S, 即构造出 $m+3F-3$ 位的加法器 A 和 $m+3F-3$ 位的减法器 S. 又由参考文献 [8] 知, 一个输入端为 $m+3F-3$ 位的 MSD 二进制加法器由 $m+3F-3$ 位的 T 变换器、 $m+3F-3$ 位的 W 变换器、 $m+3F-2$ 位的 W' 变换器和 $m+3F-1$ 位的 T_2 变换器构成, 每一位都要占用光学处理器的一个数据位, 因此, 构造一个 $m+3F-3$ 位的三值光学处理器的 $5(m+3F-3)+4$ 个数据位. 故本例程需要三值

说明：参考文献8在第1节中就有引用，这里是第二次引用，不是顺序错乱

$$V_T = 2(5(m+3F-3)+4) = 10m + 30F - 22. \quad (3)$$

3.3 加法器 A 和减法器 S 的具体构造

为了方便管理和使用三值光学计算机的众多数据位, 参考文献 [13] 定义了数据位序——每一位数据的通道在数据总线中的排序号; 同时还定义了 **改为：MSD 二进制数字除法** 的排列. 其中算位的定义是一个基本数据类型所获得的位序相连的一组数据位. 本文沿用这些定义来描述对数据位的分配和使用. 对于一个除数为 m 位有效数字、商为 F 位有效数字的 **MSD 数二进制 SRT 除法** 例程, 需要占用三值光学处理器的 V_T 个数据位来构造加法器 A 和减法器 S, 因此三值光学计算机监控系统中的数据位管理模块将在空闲数据位中寻找一段连续 V_T 位的区段, 若找到合适的区段, 就将其分配给该 **SRT 除法** 例程使用, 若找不到合适的区段, 就将该任务延后处理. 设分配给 **SRT 除法** 例程的数据位区段的起始数据位序为 h , 则本例程重构这段数据位的策略为: 在该区段的前一半上构造加法器 A, **改为：MSD 数字除法** 器 S, 具体安排见表 1.

改为：[11]

表 1 数据位序 h 到 $h + V_T$ 的重构策略

Table 1 Reconstruction strategy of data bits from h to $h + V_T$

A		S	
Data bits interval	Type	Data bits interval	Type
$h \sim h+m+3F-4$	T transform	$h+5m+15F-11 \sim h+6m+18F-15$	T transform
$h+m+3F-3 \sim h+2m+6F-7$	W transform	$h+6m+18F-14 \sim h+7m+21F-18$	W transform
$h+2m+6F-6 \sim h+3m+9F-9$	T' transform	$h+7m+21F-17 \sim h+8m+24F-20$	T' transform
$h+3m+9F-8 \sim h+4m+12F-11$	W' transform	$h+8m+24F-19 \sim h+9m+27F-22$	W' transform
$h+4m+12F-10 \sim h+5m+15F-12$	T ₂ transform	$h+9m+27F-21 \sim h+10m+30F-23$	T ₂ transform

4 三值光学计算机完成 SRT 除法的步骤

改为：MSD 数字除法

三值光学计算机是一个新生事物，它在构建光学处理器时，采用最方便易行的器件——电子元件和电子计算机来构造控制器件、运算辅助器件、数据通道和数据存储器件，以此来大幅度降低实现和应用这种新型计算机的难度，并实现与电子计算机优势互补、协同工作。因此，这种计算机在整体上是一种光电混合计算机，只是随着它的光学部件不断完善和增多，其光学成分会不断加大。目前的三值光学计算机在拥有众位数可重构三值光学处理器的同时，还使用电子 CPU 来运行它的硬件监控程序、任务调度程序、用户界面等底层软件，并通过电子 CPU 来使用传统的半导体存储器，而当三值数据出现在半导体存储器或辅助的电子 CPU 中时，使用两位二值数据的编码来表示一位三值数据。

本文所述的除法例程属于三值光学计算机的底层软件，它在运算功能上是一个独立体，因此面向用户时显现为一个完整的例程，但在后台实现上被分解成多个功能模块，这些功能模块分别同三值光学计算机的任务调度模块、处理器重构模块、存储器管理模块、光学部件控制相结合或相配合，通过协调三值光学处理器的硬件资源来完成 SRT 除法运算。

调用三值光学计算机的例程时，不仅要给出用于计算的原始数据，还要给出用于构造处理器的参数。为满足这个要求，三值光学计算机采用了运算数据文件 (*.SZG) 来传送用户运算请求的技术^[14]。于是，当使用本例程时，用户给出的 SZG 文件不仅包含所有的除数和被除数，还将同时包含构造处理器的参数：除数中的最大位数 n 、被除数中的最大位数 m 和商的有效位数 F 。其中 n 和 m 值均可以由计算机软件从用户送入的原始数据中取得，而且 F 、 n 和 m 都不超过三值光学处理器的最大值对应的数据位数，且每个除数的值不为 0，即^[12]时已被排除。


以 SZG 文件为起点，SRT 除法例程的运行过程如下所述。

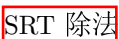

1) 获取并保存处理器构造参数。从 SZG 文件中读出被除数 N 的 MSD 二进制数的最大位数，记入变量 n ；读出除数 D 的 MSD 二进制数最大位数，记入变量 m ；读出用户要求的商数 Q 的十进制有效数字位数 G ，计算对应的商数二进制有效数字位数，记入变量 F 。完成这一步操作的软件模块被融于三值光学计算机的任务调度模块之中。

2) 重构出 MSD 数二进制加法器 A 和 MSD 数二进制减法器 S。完成这一步操作的软件融于三值光学计算机的处理器位数分配和重构模块，其重构操作根据表 1 中给出的数据位要求和重构安排（算道类型）进行^[5,15] ← 改为：[5,13]

3) 将算道分配、迭代次数和原始数据（多对除数 D 和被除数 N ）送到底层控制软件。这一步操作直接调用三值光学计算机监控系统中相应的功能模块来完成，不属本文的研究内容。

4) 定义变量。定义变量 D 和 N ，分别用于保存本次迭代的除数和被除数；定义变量 i ，用于记录

迭代次数; 定义变量 W , 用于记录第 i 次迭代产生的部分余数 W_i ; 定义变量 WA , 用于记录第 i 次迭代过程中 W_{i-1} 与 D 的和; 定义变量 WS , 用于记录第 i 次迭代过程中 W_{i-1} 与 D 的差; 定义变量 U , 用于记录第 i 次迭代产生的部分余数的中间变量 U_i ; 定义变量 Q , 用于记录第 i 次迭代生成的商数 Q_i ; 定义变量 cnt , 用于记录每次迭代要接续的被除数 N 的位序号; 定义变量 $sign$, 用于记录商 Q 的符号; 定义变量 P , 用于记录商 Q 的小数点位置。由于迭代过程在光学处理器中进行, 因此本例程将变量 D 、 N 、 W 、 WA 、 WS 、 U 、 Q 、 cnt 、 $sign$ 、 P 设置在直接控制处理器运行的底层控制软件部分, 其中 D 、 N 、 W 、 WA 、 WS 、 U 和 Q 为表达 MSD 二进制数的三值变量, 其余为常规二值变量。

5) 除法迭代运算. 从底层控制软件获得输入数据 D 和 N 的值, 若完成, 则跳转到第 8) 步. 考虑到迭代过程要反复使用加法器 A 和减法器 S , 本例程将实施这一步骤的软件模块融合于三值光学计算机的底层控制软件中, 该模块使用光学处理器 (A 和 S 部分) 和第 4) 步定义的各变量完成对输入数据的  .

5-1) 确定商 Q 的符号. 当除数和被除数同号时 $sign$ 取 1, 否则 $sign$ 取 -1 .

5-2) 将除数 D 化为整数. 左移除数 D 和被除数 N 的小数点相同位数, 使除数 D 转化为整数. 将转化后的除数和被除数仍记入 D 和 N 中.

5-3) 截取被除数 N 的高 m 位作为 W_0 , 存入变量 W , i 赋值 1, cnt 赋初值 $m+1$. 当 W_0 中包含被除数的 e 位小数时, 商数过渡值 Q 的前部补 $e-1$ 个 0, 并在最高位的 0 前加小数点, 并令 $P=0$.

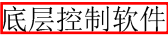
5-4) 计算 WA_i 和 WS_i . 将 W_{i-1} 和 D 同时送入加法器 A 和减法器 S , 得到 WA_i 和 WS_i , 分别存入变量 WA 和 WS ;



5-5) 计算 Q_i 和 W_i . 由 WA_i 和 WS_i , 依据式 (1) 和 (2) 计算出 Q_i 和 U_i , 分别存入变量 Q 和 U ; 在 U_i 后接续上被除数 N 的第 cnt 位得到 W_i , 存入变量 W , 为下次迭代准备输入值. 当 W_i 中第一次包含被除数的小数部分时, 将此时的 i 值保存于变量 P , 以记录商数的小数点位置.

5-6) i 加 1, cnt 加 1, 重复步骤 5-4) 和 5-5), 直到 $i=F$, 迭代运算完成.

5-7) 获得商数 Q . 根据变量 $sign$ 和 P 分别确定商数 Q 的符号和小数点位置, 获得最终商数值.




6) 保存本次除法运算结果. 将输入数据 N 、 D 和计算结果 Q 的值送入计算结果队列.

7) 计算下一对数据. 跳转到第 5) 步. 顺序重复第 5) 和 6) 步, 直到所有数据计算完毕. 当所有数据计算完成时,  不再供给输入数据, 而是送来完成符.

8) 将运算结果队  送任务调度模块. 通知底层控制软件将运算结果队列回送三值光学计算机的任务调度模块  软件的相应功能模块完成, 不属本文研究内容.

9) 将运算结果回送用户. 当所有输入数据都计算完成后, 三值光学计算机的任务调度模块会将全部运算结果回送给用户, 这项工作不属于本文研究内容.

5 模拟实验

鉴于面向应用的三值   1 正在建设中, 故本文在三值光学计算机模拟实验平台上对  例程进行验证. 模拟实验平台硬件配置为, 处理器 Intel(R) Core(TM) i5-2400CPU, 3.10GHz, 4GB 内存, 32 位操作系统, 用 C 语言编写模拟程序, 并用一个整型数据表达 MSD 二进制数的一位.

5.1 实验概述

改为：MSD数字除法

实验目标. 验证本文给出的 SRT 除法例程运行过程的有效性.

实验规划.

(1) 定义变量. 定义整型变量 n 、 m 、 G 和 F ; 定义字符型数组 *inputbuf*, 接收模拟实验输入的被除数和除数各位的值; 定改为：MSD数字除法 UT、D_INPUT 和 Q , 分别记录模拟实验输入的一组被除数、除数的值以及除法例程计算得到的所有商值; 定义一维整型数组 D 、 N 、 W 、 WA 、 WS 和 U , 用于除法的迭代运算. 本步模拟 SRT 除法例程运行过程的第 4) 步 —— 定义变量, 并为模拟 SRT 除法例程运行过程的第 1) 步做准备.

改为：MSD数字

(2) 定义读键盘函数. 从键盘上输入被除数的 MSD 数的位数、除数的 MSD 数的位数和用户要求的商数的有效数据位数 G , 这三个位数值均用十进制数表达, 并分别将前两者送入变量 n 和 m 保存, 根据 G 计算出商数的二进制有效数字位数, 送入变量 F 保存. 本步模拟 SRT 除法例程运行过程的第 1) 步 —— 从 SZG 文件头中解析出变量 n 、 m 和 F 的值. 这里认为 n 、 m 和 F 都不超过三值光学处理器最大值对应的数据位数.

改为：MSD数字除法

(3) 原始数据输入函数. 从键盘输入 20 组 MSD 二进制形式的原始数据, 通过 *inputbuf* 接收, 再将其存入数组 $N_INPUT[i][j]$ 和 $D_INPUT[i][j]$, 其中前一个下标表示输入的第 i 个数据, 后一个下标表示第 i 个数据的第 j 位, 设定每个原始数据不超过 64 位, 每位 MSD 二进制数用数组的一个项表达. 输入数据时保证每个除数都不为 0 值. 本步模拟从 SZG 文件中读取一组原始数据.

(4) 用二维整型数组 $T[3][3]=\{0, 1, -1, 1, 1, 0, -1, 0, -1\}$ 、 $W[3][3]=\{0, -1, 1, -1, 0, 0, 1, 0, 0\}$ 、 $T1[3][3]=\{0, 0, 0, 0, 1, 0, 0, 0, -1\}$ 和 $W1[3][3]=\{0, 1, -1, 1, 0, 0, -1, 0, 0\}$ 分别定义 T 、 W 、 T' 、 W' 变换的真值表, 用一维整型数组 t 、 w 、 $t1$ 、 $w1$ 分别记录变换的结果. 对于两个二位的二进制 MSD 数字 a 和 b 相加, T 变换可以用式 $t[i]=T[(a[i]+3)\%3][(b[i]+3)\%3]$ ($i=1, 2, \dots, p$) 实现, 相应地, 其他三值逻辑变换均可以类似形式定义 [7, 8]. 本步模拟 SRT 除法例程运行过程的第 2) 步 —— 重构三值光学处理器.

改为：MSD数字除法

说明：参考文献 7, 8 在第 1 节中就有引用, 这里也是第二次引用, 不是顺序错乱。

函数 MSD_Add 模拟 MSD 加法器 A. 对于两个二进制数字相加, 该函数对操作数中的每位依次调用第 (4) 项定义的 T 、 W 、 T' 、 W' 、 $T_2(T)$ 变换. 以此来模拟 MSD 二进制数字的

函数 MSD_Sub 模拟 MSD 减法器 S. 对于两个二进制数字相减, 该函数将减数的每位数取反, 然后调用 MSD_Add 函数. 以此来模拟 MSD 减法.

改为：MSD数字除法

(7) 定义函数 MSD_Div. 本步模拟 SRT 除法例程运行过程的第 5) 步 —— 除法迭代运算.

(7-1) 商数符号赋初值 $sign=1$, 依次扫描一维数组 N 和 D , 用整型变量 $nValue$ 、 $dValue$ 和 $nIndex$ 、 $dIndex$ 分别来记录 N 和 D 最高有效数据位的数值和位置. 根据 $nValue$ 、 $dValue$ 的取值, 对商数符号 $sign$ 进行修正, 并且求得操作数的绝对值. 本步模拟 SRT 除法例程迭代过程的第 5-1) 步.

改为：MSD数字除法

(7-2) 通过调整操作数的小数点所在位置的下标, 将除数调整为整数, 并将被除数作对应调整, 用整型变量 P 记录商数的小数点所在位置的下标. 本步模拟 SRT 除法例程迭代过程的第 5-2) 步、第 5-3) 步和第 5-5) 步中小数点位置的确定.

改为：MSD数字除法

(7-3) 迭代初始化. 给循环控制变量 i 赋初值. 一维整型数组 W 初始化、将一维数组 N 中的元素 $N[nIndex] \sim N[nIndex + m - 1]$ 赋值给数组 W . 定义整型局部变量 $waValue$ 和 $wsValue$, 用于分别记录一维整型数组 WA 和 WS 的最高有效数据位的数值. 本步模拟 SRT 除法例程迭代过程的第 5-3) 步.

改为：MSD数字除法

改为：MSD数字除法

(7-4) 用一维整型数组 W 和 D 的对应项的值为输入参数, 调用 MSD_Add 和 MSD_Sub, 用一维整型数组 WA 和 WS 记录运算结果. 本步模拟 SRT 除法例程迭代过程的第 5-4) 步.

(7-5) 执行下列判断语句, 确定商数 Q 和部分余数的中间变量 U 的值: 当 $waValue \leq 0$ 时, Q 赋值 -1 , 一维整型数组 WA 的值赋值给 U ; 当 $waValue > 0$ 并且 $wsValue < 0$ 时, Q 赋值 0 , 一维整型数组 W 的值赋值给 U ; 当 $wsValue \geq 0$ 时, Q 赋值 1 , 一维整型数组 WS 的值赋值给 U . 将 N 的第 $cnt + i$ 位赋给 U , 并将数组 U 赋值给 W , 作为下一次迭代的输入. 本步模拟实现 SRT 除法算法的例程迭代过程的 5-5) 步.

(7-6) 反复运行步骤 (7-4) 和 **改为: MSD 数字除法** 添加商数的符号, 根据变量 P 添加商数的小数点. 本步模拟 **SRT 除法** 例程迭代过程的第 5-6) 和 5-7) 步.

(8) 定义函数 **Residue** 有值 $Q[i]$ 进制形式数据的计算结果到一维整型数组 $Q[i]$, $i=1,2,\dots,F$. 本步模拟 **SRT 除法** 例程运行过程的第 6) 步——保存除法结果。

(9) 多组除法运算控制变量 j 赋值 1, $j++$, 直到 $j=20$, 反复调用 MSD_Div 和 Result 函数. 本步模拟 SRT 除法例程改为: MSD 数字除法 对多组原始数据逐一进行计算.

(10) 将输入数据 N 、 D 和运算结果 Q 送屏幕显示. 直接显示 N 、 D 和 Q 的 MSD 二进制形式数据 (两位二值数据表示一位三值数据), 同时显示 N 、 D 和 Q 的十进制形式数据. 本步模拟 **SRT 除法** 例程运行过程的第 9) 步 —— 将运算结果返回给用户.

本实验没有对实现 **SRT 除法** 算法的例程运行过程的第 3) 和 8) 步进行模拟, 因**改为: MSD 数字除法**光学计算机监控软件的内部通信过程, 完全由监控软件自身实施, 虽然在逻辑上它们也是本例程的操作步骤, 但在实**改为: MSD 数字除法**步或涉及到二者。

5.2 实例测试

鉴于模拟实验不能做到将所有不同长度的原始数据输入和不同有效位数要求的除法运算一一进行验证,为保证对除法例程功能验证的科学性和严谨性,模拟实验采用如下方案进行测试。

测试实例. 包含随机实例测试和特殊实例测试两部分. 随机实例测试由伪随机函数生成, 而特殊实例测试包含 10 种类型的输入数据, 列举如下: ...111111..., ... 111111..., ...101010..., ... 1 0 1 0 1 0 1..., ...1 1 1 1 1 1..., ... 1 1 1 1 1 1..., ...10 1 0 1 0 1..., ... 1 01 1 01..., ... 1 10 1 10..., ... 1 1 01 1 01...

参数设置. 设置除法运算的原始数据输入长度 n 的取值范围为 15~64 位, 用户要求的商数有效数字的取值范围为 4~20 位.

实验分组. 输入被除数和除数 1000 组, 每组包含 20 对原始数据, 同一组内被除数和除数的数据长度以及对商数的有效数据位数要求保持一致.

实验结果检查. 在输出屏上得到各组数据的运算结果, 对照理论上的精确计算值检查每组数据的模拟实验结果, 以此判断模拟实验是否成功.

5.3 结果分析

表 2 是随机选取的 6 组算例的原始数据及其模拟实验值, 表 2 的最后一列是实验结果, 当模拟实验值与理论上精确计算值相同时, 标注结果为“√”, 当模拟实验值与理论上精确计算值不相同时, 标注结果为“×”. 表 3 是将表 2 中的 6 组模拟实验送至电子计算机上进行对比计算的试验结果.

从表 2 中 6 组模拟实验可见, 三值光学计算机能够根据用户对商数**计算精度**的要求输出位数适当、计算精确的结果。事实上, 所有 1000 组实验数据的 20000 对原始数据的实验结果在有效数字准确

值部分都完全相同, 而在有效数字的最后一位 (不准确位) 上的值有差别, 这完全符合有效数字的要求, 可以判定这 20000 次实验的结果都正确.

表 2 实现除法例程的模拟实验结果

Table 2 Outputs of simulation experiment to implement divide routine

Group	Operand	MSD binary system	Decimal system	Parameter	Result
1	Dividend	1-11-11-11-11-11-11-11-11-1	349525	$n=20$	✓
	Divisor	1-11-11-11-11-11-11-11-11-1	1398101	$m=22$	
	Quotient	0.00111111111111111111	0.249998	$G=6$	
2	Dividend	100-1-100101-101-1001-10011-1000-1-10-1	440996467	$n=30$	✓
	Divisor	100-10-101-100011010101-1-10001	28456209	$m=26$	
	Quotient	1111.011111110101001111000	15.497371	$G=8$	
3	Dividend	11-1-10100-1001-1001-1001-10011101101-1001 -1001-11	1267301855779	$n=41$ $m=39$ $G=11$	✓
	Divisor	-1-101-1001-1001-10110-1-100-11000-10010-10 010-100	-393974189684		
	Quotient	-1-1.00-1-10-1-1-10-1-1-1-10-100-1-1-1-1-1-1 00000-1-1-10	-3.2167128936		
4	Dividend	10-1010-101-1-1010100-1001-1001-10010-1001-1 00101-1010-100110	4493182904940 54	$n=50$ $m=50$ $G=15$	✓
	Divisor	100-1-1001-1001-1001-101000-1-1011101-101-10 01001-100-1-10110-1	4597438654140 59		
	Quotient	0.11111010001100011101100001101001101000 0111110000	0.977323079861 832		
5	Dividend	100-1-1001100-100110-10011-100011-1001110-1 00111-1-100110-100010-1000-1	2374486813054 84335	$n=59$ $m=58$ $G=17$	✓
	Divisor	10-1001-10011-1001-1001-10001-100-100110010 -1-10011-10011-1001-100110-1	1106947152388 14795		
	Quotient	10.00100101001000111100001100110011000011 101001100100111	2.14507694241 05587		
6	Dividend	1-1-10010-1001-1001000111-1-1001-1001-1000 11-1001100-10010-1001-10010001-100	1263568731606 926468	$n=63$ $m=64$ $G=19$	✓
	Divisor	101-1-1001-1001-100-1001-10001100-1-100110-1 00110-10011-10001110-10001-1001-10	9837851055387 601954		
	Quotient	0.00100000111000010110100101010101010001 0111001010111101100010	0.128439506198 3526912		

表 3 不同计算平台上算出的商值对比

Table 3 Comparing the quotient by using different computational platform

Group	Ternary optical computer	Electronic computer	Theoretical calculating value
1	0.249998	0.24999982118602304000	0.249999821186023041253...
2	15.497371	15.49737236608010500000	15.49737236608010575126...
3	-3.2167128936	-3.21671289379510170000	-3.21671289379510184268...
4	0.977323079861832	0.97732307986183697000	0.977323079861836944524...
5	2.1450769424105587	2.14507694241055890000	2.145076942410558848622...
6	0.1284395061983526912	0.12843950619835268000	0.128439506198352691407...

从表 3 可见电子计算机始终按照固定精度进行计算, 可以完全保证商数的有效数字只有 15 位, 而

随着商数有效数字位数的增加, 电子计算机计算的商值与精确计算值的差别逐渐变大, 超过 17 位的数值在电子计算机中既不能表示也不再继续运算, 而是显示为 0.

总之, 电子计算机受到位数固定的电子 CPU 限制, 数据位数不是很多, 数据位不可分配并且处理器功能不可改变. 而三值光学计算机目前可以独立分配给用户使用的数据位数已经达到 16384 位, 这些数据位可以根据计算需求弹性分配, 重构成并行加 (减) 法器, 从而减少计算大量除法运算的时间开销, 有效降低大值数据、高精度计算的难度. 因此, 三值光学计算机在处理大量数据、大值数据和高精度运算方面表现出超越电子计算机的能力.

6 结束语

改为：MSD数字除法

本文建立了三值光学计算机采用 SRT 除法的一个除法例程, 设计了该除法例程对三值光学处理器的重构方案, 给出了该例程的操作步骤. 通过模拟实验证明了该例程的有效性. 本文通过除法例程的研究, 展示了构造三值光学计算机基本运算例程的思路和方法, 为开展三值光学计算机应用研究奠定了基础, 对于涉及到大量除法运算的问题, 如求解线性方程组的高斯消元算法、破解 RSA 密码的 QS 算法等, 提供了新的计算方案.

致谢 感谢张素兰博士研究生和王康吉吉 (zhe) 博士研究生对本文所做出的贡献.

参考文献

- 1 Oberman S F, Flynn M J. Division algorithms and implementations. IEEE Trans Comput, 1997, 48: 833-854
- 2 Liu H P. Research on high-performance arithmetic for floating-point division and the elementary functions. Dissertation for PhD Degree. Beijing: Institute of Computing Technology Chinese Academy of Sciences, 2003 [刘华平. 高性能浮点除法及基本函数功能部件的研究. 博士学位论文. 中国科学院研究生院 (计算技术研究所), 2003]
- 3 Jin Y, He H C, Lu Y T. Ternary optical computer principle. Sci China Ser F-Inf Sci, 2003, 46: 145-150
- 4 Ouyang S. Design and implementation of ternary optical processor control circuit (In Chinese). Dissertation for PhD Degree. Shanghai: Shanghai University, 2012
- 5 [欧阳山. 三值光学处理器控制电路设计和实现. 博士学位论文. 上海: 上海大学, 2012] ternary optical processors. Sci China Inf Sci, 2011, 54: 2230-2240
- 6 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
- 7 Shen Y F, Jin Y, Peng J J, et al. Simulation implementation of the computational principle of MSD adder for ternary optical computer. High Performance Computing Technology, 2010, 6: 5-10 [沈云付, 金翊, 彭俊杰, 等. 三值光学计算机 MSD 加法器计算原理模拟实现. 上海: 上海大学高性能计算中心, 2010, 6: 5-10]
- 8 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
- 9 Robertson J E. A new class of digital division methods. IRE Trans Electr Comput, 1958, 7: 88-92
- 10 Hooman N. Architectures for floating-point division. Dissertation for Ph.D. Degree. Adelaide: Adelaide University of Australia, 2005
- 11 Atkins D E. The Theory and Implementation of SRT Division. Technical Report UIUCDCS-R-67-230, 1967
- 12 Tocher K D. Techniques of multiplication and division for automatic binary computers. Q J Mech Appl Math, 1958, 11: 364-384
- 13 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]

改为：高性能计算技术

删掉

改为：11

改为 : 12

14 Gao H, Jin Y, Song K. Extension of C language in ternary optical computer. J Shanghai Univ (Nat Sci), 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]

15 Wang H J, Jin Y, Ouyang S. Design and implementation of 1-bit reconfigurable 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]

改为 : 13

改为 : MSD digit division

SRT division algorithm and implementation technique for ternary optical computer

XU Qun^{1,2}, JIN Yi^{1*}, SHEN YunFu¹, LI WeiMin¹ & HAN YueXing¹

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

2 State Grid Huangdao Power Supply Company, Tsingdao 266400, China

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

Abstract In this paper, the first division routine's algorithm of ternary and operation steps of the routine are described. Comparing with the electronic computer, TOC has many data bits, each bit can be used independently and functionality of each bit can be reconfigured at any time. In order that the three characteristics of TOC can play an important role in division, the SRT algorithm (named after D.Sweeney I.E.Robertson and T.D.Tocher) is selected as a mathematic model in author's research. Through kernel modules acting in concert with task management module of TOC's monitoring software, processor reconstruction module and hardware control system, the routine is completed by the cooperation of hardware and software of TOC. Meanwhile, the routine itself is a part of underlying software of TOC and it will be called by other application programs. In the paper, some key problems of the algorithm are discussed, such as the data bit's amount needed by the routine, strategy of reconstructing the optical processor, the collaborate ways of the routine with different parts of monitoring software, iterative times included in the routine and the decimal point position. And the routine's work process is described in detail. Besides, a simulation experiment system and its test cases are introduced, and then hundreds of simulation experiments are done. The experimental results have proved the validity of this routine. Particularly, using the number of significant digits to determine the iterative times included in SRT algorithm is firstly proposed in this paper. The work has provided a technical route and exemplification to other methods of TOC.

改为 : MSD digit division

Keywords ternary optical computer, reconfigurable processors, SRT division, division routine, MSD adder

改为 : MSD Digit Division



XU Qun 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 School of Computer Engineering and Science, Shanghai University. Her research interests include ternary optical computer and basic algorithms.



JIN Yi 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 computer and computer architecture. Dr. JIN is a senior member of China Computer Federation.



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



改为：LI



Li Weimin was born in 1972. He received the Ph.D. degree in control theory from the Donghua University, Shanghai, in 2008. Currently, he is an Associate Professor at Shanghai University. His research interests cover social network analysis, big data and optical computer application.