

# 基于 DNA 下推自动机二进制减法和乘法的实现

程 珍<sup>1)</sup> 黄玉芳<sup>1)</sup> 周 康<sup>2)</sup>

<sup>1)</sup>(华中科技大学控制科学与工程系生物计算实验室 武汉 430074)

<sup>2)</sup>(武汉工业学院数理科学系 武汉 430023)

**摘 要** 提出了基于 DNA 下推自动机二进制减法和乘法的实现方法. 一位二进制借位减法, 是通过预先构造好的 DNA 下推自动机模型在一个试管中以该模型的运行方式自动完成运算.  $m$  位二进制借位减法, 是在一位二进制减法的基础上, 按照从低位到高位顺序, 将低位产生的借位作为高位试管操作中的输入字符串, 从而完成高位的减法运算. 两位二进制乘法中包含移位和加法操作, 在两个试管中分别设计好 DNA 下推自动机模型, 分别完成被乘数与乘数各位的移位操作, 同时结合相应的生物操作, 将其作为另一个试管加法操作中的输入字符串, 则加法操作中产生的结果即为所求. 在此基础上,  $m$  位二进制乘法可通过移位操作的并行性和加法操作的串行性来完成运算. 这些实现方法为 DNA 下推自动机实现基本的算术运算提供了比较完整的运算机制.

**关键词** DNA 下推自动机; 借位减法; 乘法; 移位操作; DNA 编码

**中图法分类号** TP301 **DOI 号**: 10.3724/SP.J.1016.2009.02338

## Implementation of Binary Subtraction and Multiplication Based on DNA Push-Down Automaton

CHENG Zhen<sup>1)</sup> HUANG Yu-Fang<sup>1)</sup> ZHOU Kang<sup>2)</sup>

<sup>1)</sup>(Research Institute of Bio-molecular Computing, Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074)

<sup>2)</sup>(Department of Mathematics and Physics, Wuhan Polytechnic University, Wuhan 430023)

**Abstract** The implementations of binary borrow bit subtraction and multiplication based on DNA push-down automata are proposed. The one bit binary subtraction can be automatically completed through the operations in one tube using the DNA push-down automata model designed in advance. Based on this model,  $m$  bits subtraction can be obtained by putting the borrow bit from the low bit tube to the high bit tube as the input strings. The two bits binary multiplication based on DNA push-down automata model includes shift operations and addition operations, the shift operations between the string representing of the multiplicand and the strands denoting of each bit of the multiplier can be performed in two tubes synchronously combining biology operation, then the result strings are put into another tube as the input strings in the addition operation, finally the output in this operation is the solution. The  $m$  bits binary multiplication can be got by parallel shift operations and serial addition operations. These processes provide relatively complete arithmetic mechanisms for DNA automata.

**Keywords** DNA push-down automata; borrow bit subtraction; multiplication; shift operation; DNA-encoding

## 1 引言

自 Adleman 1994 年创立 DNA 计算模型<sup>[1]</sup>以来, DNA 计算无论是在理论上、应用模型的建立上, 还是实验与检测技术上都取得了惊人的进步. 它的优势是利用 DNA 分子具有海量的存储能力及生化反应的巨大并行性等特点进行计算<sup>[2-3]</sup>. 在 DNA 计算运算系统的研究中, 首先取得的突破性工作是 Guarnieri<sup>[4]</sup>等人于 1996 年建立了 DNA 计算的加法运算模型, 提出了基于 DNA 计算的二进制正有理数的加法运算, 解决了加法的进位问题, 创造性地完成了对 DNA 分子生物运算过程的构造与控制, 同时进行了相应的生物实验操作; 而后他们继续将上述结果扩展到一般的 3 进制乃至  $k$  进制的加法运算<sup>[5]</sup>. 1999 年, Bernard<sup>[6]</sup>等人提出了一种新的 DNA 加法模型, 这种加法运算与 Guarnieri 的工作相比, 其优点是 DNA 计算中的 Boole 逻辑的 DNA 分子中, 输入串与运算串是分离的, 该加法通过对几例实验操作成功显示了具有复杂度为 30 个逻辑门的数字分子计算的可行性. Fujiwara<sup>[7]</sup>等人对 DNA 链进行了逻辑操作和算术操作, 给出了可设定地址的操作程序; 同时, 基于 DNA 计算的二进制数乘法和除法可以通过相应的生物操作程序来实现运算<sup>[8]</sup>. 在实验室中进行分子水平计算机装置的自动操作的实验比较少, 可见文献[9-11]. DNA 计算机的研究具有突破性进展是 2001 年以色列 Benenson 研究小组<sup>[12]</sup>提出了一种可程序实现的、自治的基于生物分子的有限状态自动机模型, 它的所有部件, 包括硬件、软件、输出都是生物分子. 酶包括限制性内切酶和连接酶是分子自动机的硬件, 输入分子和转移分子是分子自动机的软件, 通过选择合适的状态转移分子来最终产生以 DNA 分子的形式代表终止状态的输出分子. 在此基础上, 如何提高分子自动机的计算能力也引起了广泛的研究<sup>[13-14]</sup>. 其后, Cavaliere 等人<sup>[15]</sup>提出了基于 DNA 分子的带有一个栈和两个栈的下推自动机模型, 并利用环形和线性的 DNA 分子和 II 类限制性酶给出了 DNA 下推自动机执行操作的过程. Reif 等人<sup>[16]</sup>提出了自治的基于 DNA 分子自动机的理论设计方案, 并将其拓展到了二维结构的设计. 李汪根等人<sup>[17]</sup>给出了 DNA 自动机二进制加法的实现方法, 在上述基础上, 本文提出了基于 DNA 下推自动机二进制数减法和乘法的实现方法, 与此相关的 DNA 计算的原理可参见文献[18].

基于 DNA 下推自动机的运算模型, 其二进制减法和乘法操作的运算方式类似于电子计算机中的运算系统. 在设计好转移函数分子的基础上, 比较容易完成运算, 相对于 DNA 计算中的运算模型而言, 该模型操作数的编码相对较简单. 一位二进制减法, 只需对 0 和 1 进行编码, 就可以完成相应的操作. 多位二进制减法是在一位二进制减法的基础上进行循环操作完成的. 两位二进制乘法包含移位操作和加法操作, 通过分别设计好相应的转移函数分子, 在不同的试管中完成移位操作和加法操作, 加法操作的结果即为该两个两位二进制数乘法的结果. 在此基础上,  $m$  位二进制乘法可通过移位操作的并行性和加法操作的串行性来完成运算.

## 2 DNA 下推自动机

自动机是实现信息与信息之间转移的一种装置, 是数字计算机的抽象模型, 是一种描述和执行运算的工具. 下面介绍下推自动机的概念、基于 DNA 分子的下推自动机的特性以及文中所涉及的生物操作.

### 2.1 下推自动机

一个下推自动机  $M$  是如下的一个七元组  $(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ , 其中,  $Q$  是一个有穷状态集合,  $\Sigma$  是一个字母表, 称为输入字母表;  $\Gamma$  是一个字母表, 称为栈字母表;  $q_0$  属于  $Q$ , 是初始状态;  $Z_0$  属于  $\Gamma$ , 是一个特殊的符号, 称为栈起始符号;  $F$  包含于  $Q$  是终结状态集合;  $\delta: Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \rightarrow Q \times \Gamma^*$  是  $M$  的动作函数. 一个自动机包含一个有穷控制器、一条输入带和一个读头. 根据自动机的当前状态以及读头的位置, 按照一定的转移规则, 自动机会自动完成一次操作, 进入下一个状态以及移动读头.

下推自动机就是在典型的自动机的基础上带有栈的存储操作, 它由有限状态控制器、输入符号串和栈三部分组成. 在执行操作时, 下推自动机从初始状态出发, 在有限状态控制器的控制下, 根据它的当前状态、输入符号串和栈顶符号通过转移函数进入下一个状态, 同时修改栈顶元素以及决定读头是否移动. 当输入符号串处理结束时, 下推自动机处于可接受状态集  $F$  的某一个状态, 且栈的内容为空, 则表示自动机接受该符号串; 否则自动机不接受该符号串.

在有穷状态的 DNA 自动机中, 酶是分子自动机的硬件, 输入分子和转移分子是分子自动机的软

件,同时还编码了检测分子.设计一个 DNA 自动机的关键在于选择合适的软件分子.当输入分子、转移分子以及所需的酶混合后,DNA 自动机将会通过一系列的杂交、链接、提取、酶切等生物操作来处理输入分子,最终确定是否产生编码 DNA 自动机终结状态的检测性输出分子,从而判断该 DNA 自动机是否接受某个输入字符串.

基于 DNA 的下推自动机则是在此基础上,增加的栈元素也用 DNA 分子表示,通过对输入分子实施一系列的操作,以 DNA 分子的形式产生代表 DNA 下推自动机最终状态的输出分子,同时通过对表示栈的 DNA 片断实施一系列的酶切、绑接和连接等生物操作实现入栈或出栈操作,以修改栈顶元素.

## 2.2 DNA 自动机的相关生物操作

下面我们介绍文中所需要的一些生物操作,这些生物操作包含了对 DNA 链进行处理的一系列生化反应,参见文献[7-8].

(1)混合.给定各含有一定数量 DNA 链的两试管溶液倒入同一试管中得到混合溶液.

(2)复制.通过聚合酶链反应 PCR 将 DNA 串复制.

(3)检测.若给定的试管中包含指定的 DNA 片断,则输出为“是”;否则输出为“否”.

(4)分离.将若干条单链的集合从给定含有一定数量 DNA 链的试管中提取出来.

(5)切割.利用限制酶在某一特定的位置切断 DNA 双链,在切割位点处被切割成两个 DNA 片断.

(6)退火.将含有单链的溶液冷却,互补的单链 DNA 序列重新组合起来,形成双链 DNA.

(7)合成.在一定长度范围内,合成可以将 A、T、C、G 4 个碱基按照预定的顺序排列在 DNA 单链上.

(8)抽提.通过亲和力,将已包含给定序列的 DNA 片断作为子串的 DNA 链提取出来.

(9)连接.由互补序列经退火形成 DNA 双链.

(10)杂交.具有互补粘性末端的两个 DNA 片断经退火后,互补末端会绑接在一起形成双链结构.

## 3 基于 DNA 下推自动机二进制减法

在本节中,首先介绍了基于 DNA 计算一位二进制减法的运算法则,通过构造基于 DNA 下推自动机模型,完成基于该模型一位二进制借位减法的

操作.然后在此基础上,给出了实现两个  $m$  位二进制减法的操作过程.

### 3.1 基于 DNA 下推自动机一位二进制减法

在这里我们只考虑被减数比减数大的情形.表 1 是基于 DNA 计算的一位二进制减法的运算法则,通过构造基于 DNA 下推自动机来实现这个运算过程,同时给出该下推自动机的各个组成部件的 DNA 分子编码方式.

表 1 一位二进制借位减法的运算法则表

被减数	减数	前位的借位	差	产生的借位
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

初始状态集合为  $\{0,1\}$ ,分别用  $S_0, S_1$  代表在初始状态时,对应被减数分别为 0,1 的情况.终止状态的集合为  $\{0,1\}$ ,分别用  $S_0, S_1$  代表对应减法的运算结果,即差为 0,1 的情况.输入符号的集合为  $\{0,1\}$ ,代表减数分别是 0,1 的情况.栈顶符号的集合为  $\{0,1\}$ ,分别代表产生的借位为 0,1 的情况.用  $\delta(Q,a,Z)=(T,A)$  表示当前状态是  $Q$ ,栈顶符号是  $Z$ ,当前输入符号是  $a$ ,在控制器的控制下,自动机的状态转变为  $T$ ,栈顶符号变为  $A$ .所设计的 DNA 下推自动机的所有可能状态转移函数的集合是

$$\begin{aligned} \delta(S_0, 0, 0) &= (S_0, 0), & \delta(S_0, 0, 1) &= (S_1, 1), \\ \delta(S_0, 1, 0) &= (S_1, 1), & \delta(S_0, 1, 1) &= (S_0, 1), \\ \delta(S_1, 0, 0) &= (S_1, 0), & \delta(S_1, 0, 1) &= (S_0, 0), \\ \delta(S_1, 1, 0) &= (S_0, 0), & \delta(S_1, 1, 1) &= (S_1, 1). \end{aligned}$$

下面来描述基于 DNA 下推自动机的各个组成部分 DNA 分子的编码方式以及相应的生物操作过程.图 1 为一位二进制减法的试管体系.

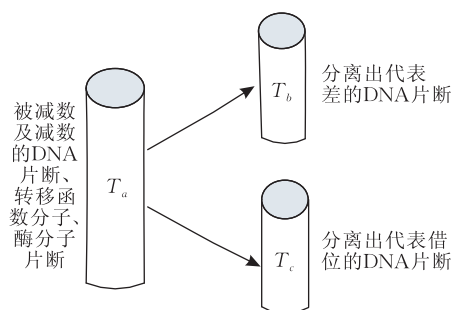


图 1 一位二进制减法的试管体系

所构造的自动机的输入符号串,为代表被减数

及减数所组成的 DNA 片断以及栈元素中代表低位减法中产生借位的 DNA 片断. 将以上预先合成的 DNA 片断按照自动机的规则合成在一起. 其中, 栈的元素在合成片断的左边, 右边是输入符号串, 中间是 FokI 酶的识别位点, 通过其来实现对栈和输入符号串进行生物操作. 在合成的 DNA 片断的右端包含一个 4bp 的粘性末端, 用于连接从低位减法操作中产生的借位片断. 同时预先设计好转移函数分子, 它的粘性末端刚好与 FokI 酶切后的片断的末端互补. 在自动机运行结束后, 加入检测分子, 同时利用内切酶 BsmFI 酶来进行酶切操作, 以检测出代表差的片断以及借位的片断, 加入 SmaI 酶进行分离操作, 即可得到代表差的 DNA 片断和代表借位的 DNA 片断. 相应的操作步骤如下:

1. 先预先合成好被减数和减数的 DNA 片断以及栈顶元素为 0 的片断和代表借位的 DNA 片断, 中间是 FokI 酶的识别位点, 在试管  $T_a$  中对这几个片断进行连接操作, 同时加入预先设计好的代表转移函数分子的 DNA 片断以及 T4 连接酶.

2. 在试管  $T_a$  中进行自动机的操作. 首先, 合成好了的 DNA 片断经 FokI 酶进行了酶切操作, 将合适的转移函数片断与酶切下的片断进行杂交. 然后将杂交后的片断通过 T4 连接酶连接起来, 在 FokI 酶的作用下, 继续进行酶切操作, 直到没有合适的转移函数片断与酶切下的片断杂交.

3. 加入检测分子, 检测分子与最后酶切下的片断经连接后形成报告分子.

4. 报告分子经 BsmFI 酶酶切后, 可以得到代表差的片断和代表借位的片断. 同时加入 SmaI 酶, 将酶切后的片断进行分离, 就可以得到代表差的 DNA 片断和代表借位的 DNA 片断, 并分别放在另外两个试管  $T_b$ ,  $T_c$  中. 图 1 所示为两个一位二进制减法的试管体系.

### 3.2 基于 DNA 下推自动机的 $m$ 位二进制减法

$m$  位二进制的减法, 则是在一位二进制数减法的基础上, 进行循环操作得到的. 这里也只考虑被减数比减数大的情形. 利用  $m$  个类似的试管按照从低位到高位顺序组成串行计算方式, 得到低位试管运算结果的同时, 分离出代表差的 DNA 片断和代表借位的 DNA 片断, 然后将低位减法操作产生的代表借位的 DNA 片断作为高位下推自动机的一个输入符号串, 即能完成高位的减法操作.

图 2 所示为基于 DNA 下推自动机  $m$  位二进制减法的操作图, 其将被减数与减数对应位相减, 并作为一个下推自动机的模型在一个试管中完成运算. 从图 2 可看到, 将每个自动机运算的结果片断分别用检测分子检测出来, 从而可以读出片断对应的二

进制数. 其中,  $B_0, B_1, \dots, B_{n-2}$  分别表示低位向高位的借位, 最后一位没有借位.  $D_0, D_1, \dots, D_{m-1}$  分别表示从低位向高位每次自动机运行后分离出差的结果.

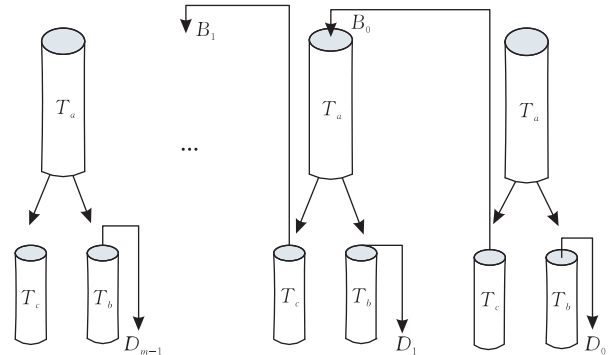


图 2 两个  $m$  位二进制减法的试管体系

## 4 基于 DNA 下推自动机的二进制乘法

基于 DNA 计算二进制乘法, 主要是进行移位操作和加法操作. 下面按照这两个操作来进行设计 DNA 下推自动机模型. 首先介绍基于 DNA 下推自动机两个两位二进制乘法, 然后在两个两位二进制乘法基础上, 介绍两个  $m$  位二进制乘法. 将移位操作的并行计算和加法操作的串行计算相结合, 经过循环操作, 即可得相应乘法的结果.

### 4.1 基于 DNA 下推自动机两位二进制乘法

在两个两位二进制乘法中, 对应的移位操作和加法操作均分步骤分别在不同的试管中进行.

#### (1) 移位操作

初始状态的集合设为  $Q$ , 令  $Q = \{00, 01, 10, 11\}$ , 终止状态的集合设为  $F$ , 令  $F = \{00, 01, 10, 11\}$ , 输入符号的集合为  $\Sigma = \{0, 1\}$ . 栈符号集合  $\Gamma = \{10, 11, 01, 12\}$ , 同时把栈中的两个字符当作一个整体来看, 栈中字符是用来记录移位运算中输入符号在乘数中的位置以及所得结果在另外两个数中的存储位置. 初始状态集合  $\{00, 01, 10, 11\}$  分别代表在初始状态时被乘数分别对应为 00, 01, 10, 11 的情况; 终止状态集合  $\{00, 01, 10, 11\}$  分别代表在终止状态时, 被乘数分别与乘数的个位与十位相乘的结果; 输入符号的集合  $\{0, 1\}$  分别代表乘数的个位与十位分别为 0, 1 的情况. 栈顶符号的集合中, 10 代表输入的是乘数的个位, 11 代表输入的是乘数的十位, 01 代表将被乘数与乘数的个位所得的结果放在第 2 个数的个位与十位, 12 代表将被乘数与乘数的十位所得的结

果放在第 3 个数的十位与百位. 在移位操作中, 所设计的下推自动机模型所有可能状态转移函数的集合是:

$$\begin{aligned} \delta(01, 0, 10) &= (00, 01), & \delta(01, 0, 10) &= (00, 01), \\ \delta(10, 0, 10) &= (00, 01), & \delta(11, 0, 10) &= (00, 01), \\ \delta(10, 1, 11) &= (10, 12), & \delta(11, 1, 11) &= (11, 12). \end{aligned}$$

对于两个两位二进制数的乘法, 如果乘数的某一位是 0, 则所得的结果为 00. 如果乘数的某一位是 1, 则将被乘数的 DNA 片段复制到相应的位子, 结合相应的生物操作, 预先设计好被乘数的 DNA 片段以及分别代表 0 和 1 的 DNA 片段. 现将代表被乘数的 DNA 片段与代表乘数个位的 DNA 片段放在一个试管中按照下推自动机模型完成运算, 在所得结果片断的左边与代表 0 的片断进行连接操作, 此为加法操作中的一个输入串. 把代表被乘数的 DNA 片段与代表乘数十位的 DNA 片段在第 2 个试管中得到相应的结果, 按照相应的状态转移函数操作, 实现了移位操作, 即将所得结果向左移了一位, 这是通过设计好的状态转移函数片断完成的操作, 同时在所得结果的片断右边与代表 0 的片断进

行连接操作. 这两个过程是并行完成的, 此为第 3 个试管中加法操作的另一个输入串, 加法操作的过程比较简单, 即可得两个两位二进制数乘法的结果.

## (2) 加法操作

在不同于上述试管中进行加法操作, 输入串即为在第 1 步进行的移位操作后的结果, 同时把栈顶元素置为 0, 然后按照加法操作中设置的状态转移函数来实现状态的转移, 加法操作完成后所得的结果即为两个两位二进制乘法的结果.

$$\begin{aligned} \delta(S_0, 0, 0) &= (S_0, 0), & \delta(S_0, 0, 1) &= (S_1, 0), \\ \delta(S_0, 1, 0) &= (S_1, 0), & \delta(S_0, 1, 1) &= (S_0, 1), \\ \delta(S_1, 0, 0) &= (S_1, 0), & \delta(S_1, 0, 1) &= (S_0, 1), \\ \delta(S_1, 1, 0) &= (S_0, 1), & \delta(S_1, 1, 1) &= (S_1, 1). \end{aligned}$$

图 3 是基于 DNA 自动机两位二进制乘法试管体系. 其中  $S_0, S_1, S_2, S_3$  分别为加法操作中分离出代表个位的和、十位的和、百位的和、千位的和的 DNA 片段,  $C_0$  为加法操作中代表进位的栈元素 DNA 片段. 可利用检测分子分别检测出  $S_0, S_1, S_2, S_3, C_0$  所对应的片断代表的二进制数, 则  $C_0 S_3 S_2 S_1 S_0$  为两位二进制乘法的运算结果.

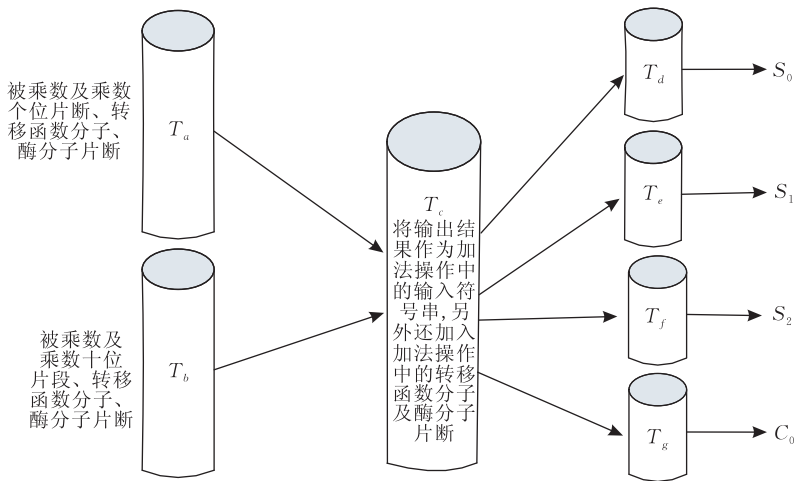


图 3 两位二进制乘法的试管体系

下面, 我们描述基于 DNA 自动机各个组成部分的 DNA 分子的编码方式以及相应的生物操作过程.

1. 所设计的自动机进行如下操作, 预先合成好代表 0, 1, 2 的编码, 利用它们的编码合成好代表被乘数和乘数的编码, 转移函数分子以及代表结果位置的栈顶元素的编码. 在试管  $a$  中加入被乘数与乘数个位的片断、转移函数分子集合、栈元素集合片断, 中间是 FokI 酶的识别位点, 通过它实现对栈和输入符号串进行生物操作. 同时, 预先设计好转移函数分子的编码, 它的粘性末端刚好与 FokI 酶酶切后的片断的末端互补. 在自动机运行结束后, 首先经 FokI 酶进行了

酶切操作, 将合适的转移函数片断与酶切下的片断进行杂交, 将杂交后的片断通过 T4 连接酶连接起来, 在 FokI 酶的作用下, 继续进行酶切操作, 直到没有合适的转移函数片断与酶切下的片断杂交.

2. 在试管  $a$  中加入检测代表结果的检测分子, 将代表 0 的片断与代表结果的片断在其左边进行连接操作. 将连接操作后的片断代表的二进制数读出来, 并按照加法操作中的编码方式重新对 0, 1 编码, 将编码好的片断放入试管  $c$  中.

3. 在试管  $b$  中放入被乘数与乘数十位的片断, 加入转移分子以及栈元素集合片断, 中间是 FokI 酶的识别位点. 在该试管中发生酶切操作, 类似于试管  $a$  中的反应方式.

4. 在试管  $b$  中加入检测代表结果的检测分子, 将代表 0

的片断与代表结果的片断在其右边进行连接操作. 将连接操作后的片断代表的二进制数读出来, 并按照加法操作中的编码方式重新对 0, 1 编码, 把该编码好的片断放入试管  $c$  中.

5. 在试管  $c$  中加入转移函数分子及酶分子进行加法操作. 加法操作过程详见文献[17].

6. 将结果进行分离操作, 得到代表加法操作中三位的的结果以及代表进位的片断. 通过检测分子, 分别检测这些片断代表的二进制数, 从而读出相应的运算结果.

#### 4.2 基于 DNA 下推自动机 $m$ 位二进制乘法

对于  $m$  位二进制数相乘, 类似于两位二进制数相乘. 当乘数增加了一位时, 在进行并行移位操作时, 只需多用一个试管完成被乘数的 DNA 片断与乘数的这一位 DNA 片断的移位运算, 这个过程是并行计算的, 同时结合生物操作, 将代表被乘数的片断与乘数每一位的片断分别在  $m$  个试管中按照乘数从低位到高位顺序进行移位操作, 即在第 1 个试管中所得结果片断的左边通过连接酶连接预先设计好的代表  $m$  个 0 的 DNA 片断, 在第 2 个试管中所得结果片断的右边通过连接酶连接代表 0 的 DNA 片断, 在其左边连接代表  $(m-1)$  个 0 的 DNA 片断, 如此类推, 将代表被乘数的 DNA 片断与乘数每一位的 DNA 片断的运算结果作为新的试管中加法操作中输入的 DNA 片断, 然后进行相应的加法操作, 即可得到  $m$  位二进制相乘的运算结果.

初始状态的集合为被乘数 ( $m$  位); 终止状态的集合为被乘数 ( $m$  位, 当乘数的某一位为 1 时) 以及 0 ( $m$  位, 当乘数的某一位为 0 时); 输入符号集合为 0, 1 (乘数的某一位为 0 或者 1); 栈元素集合为表示被乘数与乘数的某一位所得结果移位后所放的位置. 两个  $m$  位二进制数乘法如图 4 所示, 在试管  $T_a, T_b, \dots, T_{m-1}$  中, 分别表示被乘数的 DNA 片断与乘数各位的 DNA 片断按照乘数的低位到高位顺序以自动机的方式并行完成移位运算, 同时结合如上

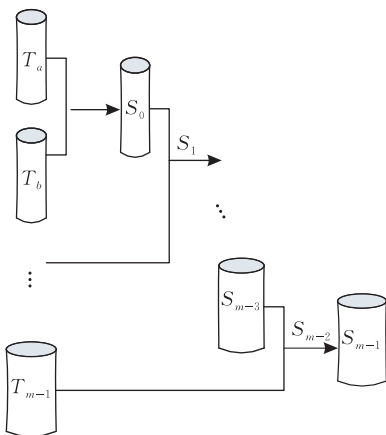


图 4  $m$  位二进制数乘法的试管体系

所述的生物操作, 从而得到加法操作中的输入串.  $S_0$  表示  $T_a, T_b$  中移位运算后前两个输入串完成加法运算所得和的片断, 同时再与第 3 个试管中移位操作后的输入串作和的运算得到  $S_1$ , 如此类推, 即可得到最后一个试管中为所有加法操作中输入串的和的片断  $S_{m-1}$ , 利用检测分子, 检测该片断代表的二进制数, 从而读出相应的运算结果.

## 5 编码实例及复杂度分析

DNA 限制性内切酶特异性结合于一段称为限制性酶识别序列的特殊 DNA 序列之内或其附近的特异位点上, 并在此切割双链 DNA. 它可分为 3 类. 在 DNA 计算中, I 类和 III 类限制性酶都不常用. II 类限制修饰系统是由两种酶分子组成的二元系统, 其中一种常用的限制性内切酶, 它切割某一特异性的脱氧核苷酸系统. DNA 被限制性内切酶切割后, 在切割处可以形成粘性末端和平头末端两种不同的结构. 不同的限制性内切酶所形成的粘性末端往往是不同的. 有相同的粘性末端而来源不同的 DNA 片断之间可以通过互补配对的碱基重新形成氢键. II 类限制修饰系统在本文中, 起着实现状态转移的作用.

FokI 酶的识别位点为 5'-GGATG-3', 剪切位点是 (9, 13). BsmFI 酶的识别位点为 5'-GGGAC-3', 剪切位点是 (10, 14). SmaI 酶的识别位点为 5'-ATTTAAAT-3', 剪切位点是识别位点内部 1/2 位置.

### 5.1 二进制减法的编码实例

为了验证基于 DNA 自动机的二进制串行减法的可行性, 我们给出了一位二进制减法运算过程中所需分子的编码. 表 2 为 0, 1 及结束时终止符 T 的编码, 表 3 为转移函数分子的编码, 表 4 为检测分子的编码.

表 2 0, 1 及结束时终止符 T 的编码

0 编码	1 编码	T 编码
GACCGA	CGCAGC	TATGCA
CTGGCT	GCGTCG	ATACGT

表 3 转移函数分子的编码

转移函数	对应转移函数分子的编码
$\delta(S_0, 0, 0) = (S_0, 0)$	GATCCGGATGTGA CTAGGCTACACTTGGC
$\delta(S_0, 0, 1) = (S_1, 1)$	GATCCGGATGTG CTAGGCTACACCTC

(续 表)

转移函数	对应转移函数分子的编码
$\delta(S_0, 1, 0) = (S_1, 1)$	GATCCGGATGTG CTAGGCCTACACCGTC
$\delta(S_0, 1, 1) = (S_0, 1)$	GATCCGGATGTGA CTAGGCCTACACTCGTC
$\delta(S_1, 0, 0) = (S_1, 0)$	GATCCGGATGTGA CTAGGCCTACACTGTCG
$\delta(S_1, 0, 1) = (S_0, 0)$	GATCCGGATGTGA CTAGGCCTACACTGGCT
$\delta(S_1, 1, 0) = (S_0, 0)$	GATCCGGATGTGA CTAGGCCTACACTGGCT
$\delta(S_1, 1, 1) = (S_1, 1)$	GATCCGGATGTG CTAGGCCTACACGTCG

表 4 检测结果的检测分子的 DNA 编码

终止状态	栈顶元素	对应检测分子的编码
$S_0$	0	GACCGATATGCATGGTCCCATTTAAATGGGAC CTATACGTACCAGGGTAAATTTACCTG
$S_0$	1	CAGCGATATGCATGGTCCCATTTAAATGGGAC CTATACGTACCAGGGTAAATTTACCTG
$S_1$	0	GACCGATATGCATGGTCCCATTTAAATGGGAC CTATACGTACCAGGGTAAATTTACCTG
$S_1$	1	CAGCGATATGCATGGTCCCATTTAAATGGGAC CTATACGTACCAGGGTAAATTTACCTG

5.2 二进制乘法的编码实例

为了验证基于 DNA 自动机的二进制串行乘法的可行性,我们给出了两位二进制乘法运算过程中所需分子的编码.表 5 为 00,01,10,11,12,0,1 及结束时终止符 T 的编码,表 6 为转移函数分子的编码,表 7 为检测分子的编码.

表 5 00,01,10,11,12,0,1 及结束时终止符 T 的编码

状态	编码
00	GCAGGC CGTCCG
01	ACGGAC TGCCTG
10	GAAGCG CTTCGC
11	GACAGC CTGTCG
12	GCGAAC CGCTTC
0	CAAGCA GTTCGT
1	AAGCCG TTCGGC
T	CATGTA GTACAT

表 6 转移函数分子的编码

转移函数	对应转移函数分子的编码
$\delta(00, 0, 10) = (00, 01)$	CAGTCGGATGTGA GTCAGCCTACACTCGTC
$\delta(00, 1, 10) = (00, 01)$	CAGTCGGATGTG GTCAGCCTACACGTCC
$\delta(00, 0, 11) = (00, 12)$	CAGTCGGATGTG GTCAGCCTACACTCCG
$\delta(00, 1, 11) = (00, 12)$	CAGTCGGATGTGA GTCAGCCTACACTGTCC
$\delta(01, 0, 10) = (00, 01)$	CAGTCGGATGTGA GTCAGCCTACACTCGTC
$\delta(01, 1, 10) = (01, 01)$	CAGTCGGATGTGA GTCAGCCTACACTTGCC
$\delta(01, 0, 11) = (00, 12)$	CAGTCGGATGTGA GTCAGCCTACACTCGTC
$\delta(01, 1, 11) = (01, 12)$	CAGTCGGATGTG GTCAGCCTACACGCCT
$\delta(10, 0, 10) = (00, 01)$	CAGTCGGATGTG GTCAGCCTACACTCCG
$\delta(10, 1, 10) = (10, 01)$	CAGTCGGATGTG GTCAGCCTACACCTTC
$\delta(10, 0, 11) = (00, 12)$	CAGTCGGATGTG GTCAGCCTACACCGTC
$\delta(10, 1, 11) = (10, 12)$	CAGTCGGATGTG GTCAGCCTACACTCGC
$\delta(11, 0, 10) = (00, 01)$	CAGTCGGATGTG GTCAGCCTACACCGTC
$\delta(11, 1, 10) = (11, 01)$	CAGTCGGATGTG GTCAGCCTACACCTGT
$\delta(11, 0, 11) = (00, 12)$	CAGTCGGATGTG GTCAGCCTACACTCCG
$\delta(11, 1, 11) = (11, 12)$	CAGTCGGATGTG GTCAGCCTACACTGTC

表 7 检测结果的检测分子的 DNA 编码

终止状态	栈顶元素	对应检测分子的编码
00	01	ACGGGACATGTATGGTCCCATTTAAATGGGAC CTGTACATACCAGGGTAAATTTACCTG
00	12	CGAAGACATGTATGGTCCCATTTAAATGGGAC CTGTACATACCAGGGTAAATTTACCTG
01	01	ACGGGACATGTATGGTCCCATTTAAATGGGAC CTGTACATACCAGGGTAAATTTACCTG
01	12	CGAAGACATGTATGGTCCCATTTAAATGGGAC CTGTACATACCAGGGTAAATTTACCTG
10	01	ACGGGACATGTATGGTCCCATTTAAATGGGAC CTGTACATACCAGGGTAAATTTACCTG
10	12	CGAAGACATGTATGGTCCCATTTAAATGGGAC CTGTACATACCAGGGTAAATTTACCTG
11	01	ACGGGACATGTATGGTCCCATTTAAATGGGAC CTGTACATACCAGGGTAAATTTACCTG
11	12	CGAAGACATGTATGGTCCCATTTAAATGGGAC CTGTACATACCAGGGTAAATTTACCTG

### 5.3 复杂度分析

基于 DNA 下推自动机  $m$  位二进制减法, 每一位的运算过程中, 包括 DNA 下推自动机的操作及对应的两次分离操作, 因此总共操作次数是  $2m$  次, 那么该操作的时间复杂度为  $O(m)$ . 在运算过程中, 只需要对 0, 1、终止符号以及转移函数分子和检测分子编码. 因此, 需要 DNA 链的条数没有随二进制数的位数增多而增多, 则该算法的空间复杂度为  $O(1)$ .

基于 DNA 下推自动机两个  $m$  位二进制减法运算过程中, 包括移位操作和加法操作, 移位操作有  $m$  次, 同时有连接操作  $(2m-1)$  次, 加法操作  $m(m-1)$  次, 因此该操作的时间复杂度是  $O(m^2)$ . 在此过程中, 在移位操作中, 需要对 0, 1 以及转移函数分子编码, 在加法操作中需要重新对 0, 1 进行编码以及对相应的转移函数分子编码, 可以看到, 需要不超过  $O(m)$  条 DNA 链, 则该算法的空间复杂度为  $O(m)$ .

## 6 结论与展望

提出了基于 DNA 下推自动机二进制减法及乘法的实现方法, 给出了相应的运算模型、操作步骤以及复杂度分析. 该模型操作数简单, 具有可行性, 只需对 DNA 下推自动机的硬件分子及软件分子编码好, 即可完成相应的运算操作. 除此之外, 还可以考虑减法运算中, 差是负数的情形, 同时还可将其应用到矩阵运算等, 可以看到该模型的提出对研究 DNA 计算机的运算系统有重大意义.

### 参 考 文 献

- [1] Adleman L M. Molecular computation of solutions to combinatorial problems. *Science*, 1994, 66(11): 1021-1024
- [2] Pan L Q, Xu J, Liu Y C. A Surface-based DNA algorithm for the minimal vertex problem. *Progress in Natural Science*, 2003, 13(1): 81-84
- [3] Pan L Q, Liu G W, Xu J. Solid phase based DNA solution of the coloring problem. *Progress in Natural Science*, 2004, 14(5): 104-107
- [4] Guarnieri F, Fliss M, Bancroft C. Making DNA add. *Science*, 1996, 273: 220-223
- [5] Guarnieri F, Bancroft C. Use of a horizontal chain reaction for DNA-based addition. *DIMACS Series in Discrete Mathematical Theoretical Computer Science*, 1999, 44: 105-111
- [6] Bernard Y, Allen P, Siu L et al. DNA implementation of addition in which the input strands are separate from the operator strands. *Biosystems*, 1999, 52(1): 165-174
- [7] Fujiwara A, Matsumoto K, Chen W. Procedures for logic and arithmetic operations with DNA strands. *International Journal of Foundations of Computer Science*, 2004, 15(3): 461-474
- [8] Fukagawa H, Fujiwara A. Procedures for multiplication and division in DNA computing//*Proceedings of the FCS 2006*, Las Vegas, Nevada, USA, 2006: 95-101
- [9] Chang W L, Guo M Y, Michael Shan-Hui Ho. Fast parallel molecular algorithms for DNA-based computation: Factoring integers. *IEEE Transactions on Nanobioscience*, 2005, 4(2): 149-163
- [10] Winfree E, Liu F et al. Design and self-assembly of two-dimensional DNA crystals. *Nature*, 1998, 394: 539-544
- [11] Mao C, LaBean T H et al. Logical computation using algorithmic self-assembly of DNA triple-crossover molecules. *Nature*, 2000, 407: 93-96
- [12] Benenson Y, Paz-Elizur T, Adar R et al. Programmable and autonomous computing machine made of biomolecular. *Nature*, 2001, 414(22): 430-434
- [13] Shi X L, Zhang Z et al. Improve capability of DNA automaton; DNA automaton with three internal states and tape head move in two directions//*Lecture Notes in Computer Science* 3645. Springer, 2005. 71-79
- [14] Zhang Z, Xu J, Liu J et al. Programmable Pushdown Store Base on DNA Computing. Berlin Heidelberg: Springer-Verlag, 2006: 286-293
- [15] Cavaliere M, Jonoska N, Yegorov S et al. Biomolecule implementation of computing devices with unbounded memory. Berlin Heidelberg: Springer Verlag, 2005, 3384: 35-49
- [16] Peng Y, Turberfield A J, Reif J H et al. Design of autonomous DNA cellular automata. Berlin Heidelberg: Springer-Verlag, 2006, 3892: 399-416
- [17] Li Wang-Gen, Ding Yong-Sheng. Implementation of serial binary carry-save adder based on DNA automata. *Computer Science*, 2006, 33(7): 167-170(in Chinese)  
(李汪根, 丁永生. 基于 DNA 自动机的串行二进制进位加法的实现. *计算机科学*, 2006, 33(7): 167-170)
- [18] Xu Jin, Zhang Lei. DNA computer principle, advances and difficulties (I): Biological computing system and its applications to graph theory. *Chinese Journal of Computers*, 2003, 26(1): 1-11(in Chinese)  
(许进, 张雷. DNA 计算机原理、进展及难点 (I): 生物计算系统及其在图论中的应用. *计算机学报*, 2003, 26(1): 1-11)



**CHENG Zhen**, born in 1982, Ph. D. candidate. Her research interests include DNA computing, DNA automata, graph theory.

**HUANG Yu-Fang**, born in 1982, Ph. D. candidate. Her research interests include DNA computing, DNA automata, graph theory.

**ZHOU Kang**, born in 1965, associate professor. His research interests include DNA computing, operational research, the stability of system.

### Background

This research is supported by the National Natural Science Foundation of China (grant Nos. 60373089, 60674106, 60573190 and 60803113), and program for New Century Excellent Talents in University (NCET-05-0612). There are many research results based on this Foundation, such as the coding of DNA computing, the application of DNA automata

in cryptography and so on. This paper is concerned to the field which is the arithmetic computation of DNA automata in DNA computing. The implementations of binary borrow bit subtraction and multiplication based on DNA push-down automata are proposed. These processes provide relatively complete arithmetic mechanisms for DNA automata.