

一种实现三值逻辑电路的 DNA 计算模型^{*})

朱翔鸥 刘文斌 陈丽春 吴桂初

(温州大学计算机科学与工程学院 温州 325035)

摘要 近年来,随着生物计算和量子计算研究的深入,多值逻辑电路的各种实现成为一个热门的研究方向。发夹结构是 DNA 分子一种特殊杂交方式的产物,具有结果稳定、特异性强的优点。本文首次提出了一种利用 DNA 分子来实现多值逻辑电路的方法,用 DNA 分子的多发夹结构来表示三值逻辑的值,并给出“与”运算和“或”运算的计算模型,该模型适合应用于大规模的多值逻辑电路。

关键词 DNA 计算,多值逻辑,发夹结构

A DNA Computing Model for Three-Valued Logical Circuit

ZHU Xiang-Ou LIU Wen-Bin CHEN Li-Chun WU Gui-Chu

(College of Computer Science and Engineering, Wenzhou University, Wenzhou 325035)

Abstract In recent years, because biological computing and quantum computing are deeply studied, the implementation of multi-valued logical circuit is becoming an interesting field for researchers. On the other side, Hairpin Formation is the product of DNA molecules in special hybridization way. It has such advantages as stability in results and strong in deviationism. This paper puts forward a new method of implementing the multi-valued logical circuit. This method use Hairpin formation to represent the values of three-valued logic, and it also give the computing model for “AND” and “OR” operations. This model can be used multi-valued logical circuits which are large scale and low energy consumption.

Keywords DNA computing, Multi-valued logic, Hairpin formation

1 引言

1921 年美国逻辑学家 E. L. Post 首次提出对任意基数都保持功能完备的多值逻辑代数系统^[1],自 20 世纪 70 年代初期,多值逻辑的研究开始引起国际计算机学术界的重视。但多值逻辑在电子科学技术和计算机科学技术中的应用,目前还远没有二值逻辑那么普遍,这是由于多值逻辑电路的复杂性和电路实现的困难性。然而,自然界中许多事物之间存在着多值的逻辑问题,例如电机的状态有“正传”、“停止”和“反转”,一个系统的状态有“正常”、“异常”和“故障”,选举投票有“同意”、“弃权”和“反对”等。多值逻辑的各种状态可以用不同的电压或电流来表示,也可以用 DNA 分子的各种状态来表示。

1994 年,美国南加州大学的 Adleman 博士通过 DNA 的生化实验,解决了有向图的哈密尔顿路问题(HPP)^[2],开创了 DNA 分子计算的新纪元。目前,人们在研究 NP-完全问题的 DNA 计算模型的同时,也在开始探索新的潜在的应用^[3]。构建基于 DNA 分子的逻辑电路就是其中一个研究方向。例如 2004 年,刘文斌等提出了一个基于吡啶二聚物的诱导发夹结构的逻辑“与非”门的 DNA 计算模型^[4]。本文通过一种特定发夹结构的 DNA 分子来表示三值逻辑值,并完成三值逻辑“与”和“或”的逻辑运算和输出过程。数学上已经证明,任何一个布尔电路都可以在多项式时间转化为标准的层状结构,并且同一层中的逻辑门均为相同形式,“与”阵列和“或”阵列

是常用的逻辑阵列。衡量布尔电路规模通常的两个指标为:电路中所包含的逻辑门的数目和电路中逻辑门的层数,对于多值逻辑电路也是一样的。图 1 给出了一个由与门(AND)和或门(OR)组成的 3 层逻辑电路的示意图。显然,由于 DNA 分子巨大的并行性,用 DNA 分子来构成多值逻辑的阵列电路是有显著意义的。

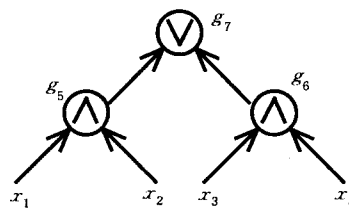


图 1 $f(x_1, x_2, x_3, x_4) = (x_1 \wedge x_2) \vee (x_3 \wedge x_4)$ 的逻辑电路

2 三值逻辑运算的定义

三值逻辑是最简单的多值逻辑,也是最重要的多值逻辑系统,通过三值逻辑系统可以更深入地了解多值逻辑系统的理论与应用。多值逻辑理论体系众多,目前主要有 Post 代数、Vranesic-Lee-Smith 代数、Allen-Givone 代数、模代数、T 门算子代数和对称三值代数等。下面讨论三值逻辑的“与”和“或”逻辑运算。

设 $L = \{0, 1, 2\}$ 是三值逻辑的值集合, $x, y \in L$, 定义:

(1)“与”运算(取小) $x \wedge y = \min(x, y)$

^{*})基金项目:国家自然科学基金(No. 60403002),浙江省自然科学基金(Y106654)。朱翔鸥 硕士,副教授,研究方向为生物分子计算及生物信息学;刘文斌 博士,副教授,研究方向为计算生物及 DNA 分子计算;陈丽春 硕士,实验师,研究方向为计算生物和计算机应用;王向红 硕士,教授,研究方向为计算生物及生物信息学。

(2)“或”运算(取大) $x \vee y = \max(x, y)$

三值逻辑的“与”和“或”运算结果见真值表 1 和真值表 2。

表 1 “与”运算真值表

$x \wedge y$		y		
		0	1	2
x	0	0	0	0
	1	0	1	1
	2	0	1	2

表 2 “或”运算真值表

$x \vee y$		y		
		0	1	2
x	0	0	1	2
	1	1	1	2
	2	2	2	2

3 多发夹结构

发夹结构是一种在 DNA 分子或 RNA 分子中经常出现的二级结构,当 DNA 分子或 RNA 分子上存在相互互补的序列,在适当的条件下它们就会杂交从而形成发夹结构。尽管在 DNA 计算中我们希望尽量避免“发夹”结构的产生,但是通过精心设计的发夹结构却有很多特殊的应用,如文[4]的与非门计算模型,文[5,6]的可满足问题的计算模型。此外,由于“发夹”具有结构简单,易于控制等优点,已经广泛地应用于生物传感器和分子的自组装等方面^[7,8]。

多发夹结构如图 2 所示,由两条不完全匹配的 DNA 链部分杂交,从而形成多发夹结构。两条 DNA 链分别由五部分组成,短链由“A,B,C,D,E”组成,长链由“A',B',C',D',E'”组成。其中“A,C,E”和“A',C',E'”完全匹配,“B,D”和“B',D'”不匹配,“B',D'”的长度比“B,D”长,“B,D”和“B',D'”之间有足够大的移位距离。为了保证三值逻辑电路的实现,长链和短链的五个部分分别设计不同的长度。其中,“A,A',B”部分最短,都为 5mer,“B',D'”为 10mer,“C,C'”为 15mer,“D”为 20mer,“E,E'”为 25mer。当长链和短链的末端一起固定在表面上,两条 DNA 链就会形成如图 2 的双发夹结构,其中“B'”部分形成的发夹叫小发夹,“D'”部分形成的发夹叫大发夹。

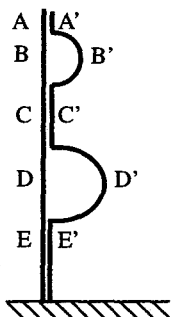


图 2 多发夹结构示意图

4 三值逻辑电路的 DNA 分子实现

4.1 逻辑值的 DNA 编码

要实现多值逻辑运算,首先要确定多值逻辑值的表示方法。图 3 是用 DNA 分子的多发夹结构的三种状态表示三值

逻辑值。具体的表示方法如下:

(1) 当固定在表面上的长短双链的“A,C,E”和“A',C',E'”杂交形成氢键而相对固定,而“B,D”和“B',D'”不匹配无法杂交,形成两个大小发夹,此时表示逻辑值“0”;

(2) 表面上的长短双链处于大小发夹状态时,溶液中加入“B'”的补链“ $\overline{B'}$ ”,经适当加热解链操作和退火操作,此时“B'”和“ $\overline{B'}$ ”杂交形成双链,由于“B'”和“ $\overline{B'}$ ”杂交的化学自由能变化要远大于“A”和“A'”杂交的化学自由能变化,因此“B'”和“ $\overline{B'}$ ”杂交时就将“A”和“A'”解链。另外,由于“C,E”和“C',E'”杂交的化学自由能变化要大于“B'”和“ $\overline{B'}$ ”杂交的化学自由能变化,因此“C,E”和“C',E'”不会解链,仍然以杂交形式存在。这样就形成了单个大发夹,此时表示逻辑值“1”;

(3) 长短双链处于大小发夹状态或单大发夹状态时,加入“B',D'”的补链“ $\overline{B'}, \overline{D'}$ ”,经解链和退火后,“B',D'”和“ $\overline{B'}, \overline{D'}$ ”分别杂交形成双链,而且有足够大的化学自由能变化来解开“A,C”和“A',C'”双链结构,但“E”和“E'”仍然杂交。此时大小发夹均不存在,用来表示逻辑值“2”。

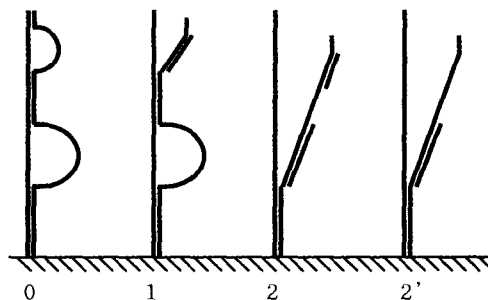


图 3 三值逻辑值的表示方法

4.2 “与”、“或”运算

1. “或”运算

“或”运算中的一个输入变量 $X, X=x, x \in \{0, 1, 2\}$, 逻辑变量 X 的取值由 4.1 节中图 3 的方法来表示, 运算结果 $F = X \vee Y$ 的逻辑值也同样表示。另一个输入变量 $Y, Y=y, y \in \{0, 1, 2\}$, 变量 Y 逻辑值用各种补链来表示, 定义为:

(1) 溶液中加入“B',D'”的补链“ $\overline{B'}, \overline{D'}$ ”, 表示逻辑值“2”;

(2) 仅加入“B'”的补链“ $\overline{B'}$ ”, 表示逻辑值“1”;

(3) 溶液中不加入任何 DNA 链, 表示逻辑值“0”。

这样,“或”逻辑运算 $X \vee Y$ 就容易实现, 对照表 2“或”运算真值表, 运算操作如下:

(1) 当 $X=0$ 时, 固定在表面上的长短双链呈双发夹状态, 溶液中加入“ $\overline{B'}, \overline{D'}$ ”($Y=2$), 经适当解链操作和退火操作, 大小发夹都被打开, 运算结果表示为逻辑值“2”, 从而实现 $0 \vee 2 = 2$ 。当往双发夹状态($X=0$)的溶液中加入“ $\overline{B'}$ ”($Y=1$), 经解链和退火后, 小发夹被打开, 大发夹不变, 运算结果表示为逻辑值“1”, 从而实现 $0 \vee 1 = 1$, 见图 4。如果初始溶液中加入任何 DNA 链, 经解链和退火后, 显然长短双链仍呈双发夹状态, 从而实现 $0 \vee 0 = 0$;

(2) 当 $X=1$ 时, 表面上的长短双链呈单大发夹状态, 如果往溶液中加入“ $\overline{B'}, \overline{D'}$ ”($Y=2$), 经解链和退火后大发夹被打开, 运算结果表示为逻辑值“2”, 实现 $1 \vee 2 = 2$ 。如果溶液中加入“ $\overline{B'}$ ”($Y=1$)或者不加入任何 DNA 链, 经解链和退火后, 长短双链状态不会改变, 实现 $1 \vee 1 = 1$ 和 $1 \vee 0 = 1$, 见图 5;

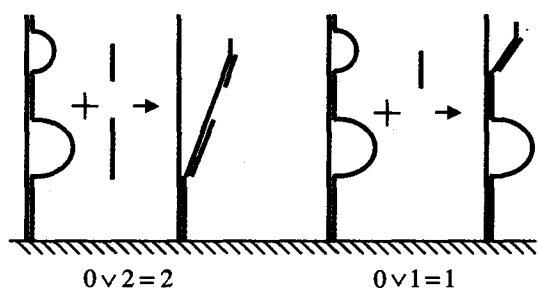


图4 “或”运算示意图1

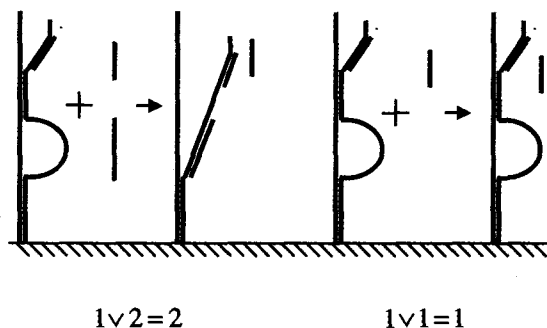


图5 “或”运算示意图2

(3) 当 $X=2$ 时, 长短双链在溶液中均不存在大小发夹, 无论往溶液中加入“ $\overline{B'}$, $\overline{D'}$ ”(Y=2)、“ $\overline{B'}$ ”(Y=1)或不加入任何东西(Y=0), 长短双链状态不会因解链和退火操作而改变, 即实现 $2 \vee 2=2$, $2 \vee 1=2$ 和 $2 \vee 0=2$ 。

2. “与”运算

“与”运算中的输入变量 X 和运算结果的逻辑值 $F=X \wedge Y$ 同样由 4.1 节中图 3 的方法来表示。输入变量 Y 逻辑值定义为:

(1) 溶液中加入“ $\overline{B'}$, $\overline{D'}$ ”段的 DNA 链, 表示逻辑值“0”;

(2) 仅加入“ $\overline{D'}$ ”段的 DNA 链, 表示逻辑值“1”;

(3) 溶液中加入任何 DNA 链, 表示逻辑值“2”。这样, 逻辑运算 $x \wedge y$ 同样有如下操作如下:

(1) 当 $X=2$ 时, 在表面上的长短双链无发夹状态, 如果溶液中加入“ $\overline{B'}$, $\overline{D'}$ ”段的 DNA 链(Y=0), 经加热解链, “ $\overline{B'}$, $\overline{D'}$ ”和“ $\overline{B'}$, $\overline{D'}$ ”形成的双链也解开。然而在退火操作过程中, 解链的“ $\overline{B'}$, $\overline{D'}$ ”段 DNA 链与加入的“ $\overline{B'}$, $\overline{D'}$ ”段 DNA 链相互匹配而杂交形成独立的双链, 游离于溶液中。同时, 固定在表面的长链失去了与“ $\overline{B'}$, $\overline{D'}$ ”段 DNA 链杂交的机会, 于是和附近表面的短链杂交, 形成大小发夹状, 从而实现 $2 \wedge 0=0$ 。如果溶液中加入“ $\overline{D'}$ ”(Y=1), 经加热解链后, 在退火操作过程中, 解链的“ $\overline{D'}$ ”段 DNA 链与加入的“ $\overline{D'}$ ”段 DNA 链杂交形成游离的双链。同时, 固定在表面的长链失去了与“ $\overline{D'}$ ”段 DNA 链杂交的机会, 但仍和“ $\overline{B'}$ ”段 DNA 链杂交, 于是在和附近表面的短链杂交时, 就形成单大发夹状, 实现 $2 \wedge 1=1$, 见图 6。显然, 如果溶液中加入任何 DNA 链, 则解链退火后, 表面上的长短双链仍处于无发夹状态, 从而实现 $2 \wedge 2=2$;

(2) 当 $X=1$ 时, 表面上的长短双链呈单大发夹状态, 如果溶液中加入“ $\overline{B'}$, $\overline{D'}$ ”段的 DNA 链(Y=0), 经加热解链, 大发夹打开, 同时“ $\overline{B'}$ ”和“ $\overline{D'}$ ”形成的双链也解开, 然而在退火操作过程中, 解链的“ $\overline{B'}$ ”段与加入的“ $\overline{B'}$ ”段 DNA 链杂交形成游离的双链。同时, 固定在表面的长链失去了与“ $\overline{B'}$ ”段 DNA 链杂交的机会, 而溶液中也无“ $\overline{D'}$ ”段 DNA 链, 于是和附近表面的短链

杂交, 形成大小发夹状, 从而实现 $1 \wedge 0=0$ 。如果溶液中加入“ $\overline{D'}$ ”(Y=1), 经加热解链后, 在退火操作过程中, 解链的“ $\overline{B'}$ ”段仍和固定在表面的长链“ $\overline{B'}$ ”段杂交, 表面上的长短双链杂交后, 呈单大发夹状态仍形成单大发夹状, 实现 $1 \wedge 1=1$, 见图 7。显然, 如果溶液中加入任何 DNA 链, 则解链退火后, 表面上的长短双链仍保持原来状态, 从而实现 $1 \wedge 2=1$;

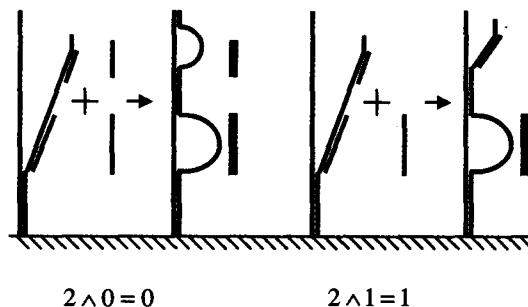


图6 “与”运算示意图1

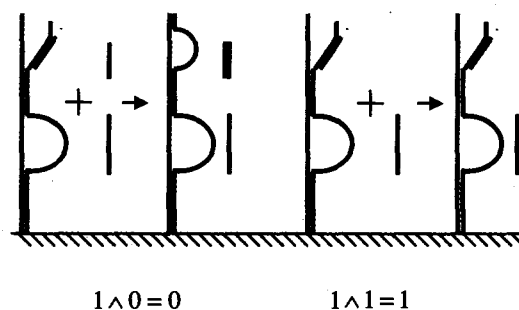


图7 “与”运算示意图2

(3) 当 $X=0$ 时, 表面上的长短双链呈大小双发夹状态, 无论往溶液中加入“ $\overline{B'}$, $\overline{D'}$ ”(Y=0)、“ $\overline{D'}$ ”(Y=1)或不加入任何东西(Y=2), 因为溶液无“ $\overline{B'}$, $\overline{D'}$ ”段 DNA 链, 表面上的长短双链状态解链和退火操作仍形成大小双发夹结构, 即实现 $0 \wedge 2=0$, $0 \wedge 1=0$ 和 $0 \wedge 0=0$ 。

为了避免“与”运算的生物操作中的杂交竞争, 保证解链后的补链“ $\overline{B'}$, $\overline{D'}$ ”要优先与表示变量 Y 的“ $\overline{B'}$, $\overline{D'}$ ”段 DNA 链杂交, 需要在解链后将表面和溶液暂时分离, 使溶液中只剩下补链“ $\overline{B'}$, $\overline{D'}$ ”, 然后往溶液中加入表示变量 Y 的“ $\overline{B'}$, $\overline{D'}$ ”段 DNA 链, 使补链“ $\overline{B'}$, $\overline{D'}$ ”与其优先杂交, 然后再将表面浸入溶液中, 使没杂交的补链“ $\overline{B'}$, $\overline{D'}$ ”与固定在表面上长链的“ $\overline{B'}$, $\overline{D'}$ ”段杂交。

3. 运算结果的输出及传递

运算结果的输出非常关键, 不仅表示本级逻辑电路的输出信息, 而且也是用于下一级逻辑电路的输入信息。上述逻辑电路运算结果的输出很简单, 加热表面上的长短双链使之解链, 游离出的“ $\overline{B'}$, $\overline{D'}$ ”段 DNA 链就是运算结果的输出。显然, 运算结果输出逻辑值表示同“或”运算的输入变量 Y 的表示是一样的, 也就是说, 运算结果的输出可以直接作为下一级“或门”电路的输入。但是对于“与门”输入电路, 运算结果的输出需要一次杂交移除(求补)转化后才能作为输入信号。比较运算结果和“与”运算的输入信号 Y 的 DNA 编码表示, 可知运算结果和“或”运算输入信号 Y 均由长链“ $\overline{B'}$, $\overline{D'}$ ”的补链“ $\overline{B'}$, $\overline{D'}$ ”来表示, “与”运算输入信号 Y 由长链“ $\overline{B'}$, $\overline{D'}$ ”段 DNA 链来表示, 两者的表示刚好互补。

(下转第 187 页)

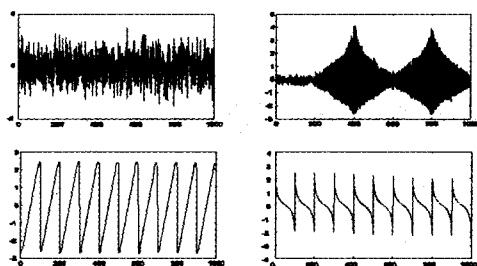


图4 本文算法的分离结果

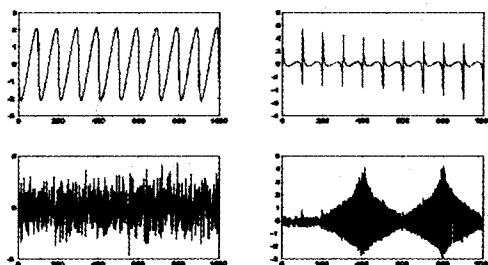


图5 最大熵算法的分离结果

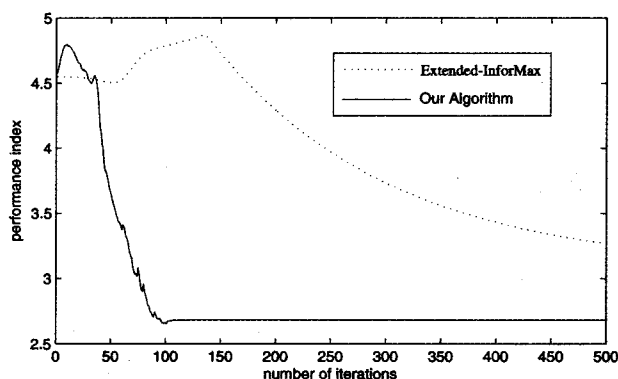


图6 本文的方法和扩展最大熵化算法的收敛曲线

结论 本文提出一种有效的估计评价函数的方法,并把该方法用于盲信号分离算法。该方法不需要源信号的概率密度模型,通过多层神经网络估计任意信号的概率密度函数,并由此估计信号的评价函数。该方法有能力估计任意信号的密度,具有很好的收敛特性,特别适合真实信号的混合。实验验证了其在分离性能和收敛速度上的有效性。

参考文献

- 1 Comon P. Independent component analysis, a new concept? *Signal Processing*, 1994, 36(3): 287~314
- 2 Bell A J, Sejnowski T. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 1995, 7(6):1129~1159
- 3 Nadal J P, Parga N. Non-linear neurons in the low-noise limit: A factorial code maximizes information transfer. *Network*, 1994, 4: 295~312
- 4 Cardoso J F. Blind signal separation; Statistical principles. *Proc IEEE Special Issue on Blind Identification and Estimation*, 1998, 9: 2009~2025
- 5 Lee T W, Girolami M, Sejnowski T J. Independent component analysis using an extended InfoMax algorithm for mixed sub-Gaussian and super-Gaussian sources. *Neural Computat*, 1999, 11(2):417~441
- 6 Amari S I, Chen T P, Cichocki A. Stability analysis of learning algorithms for blind source separation. *Neural Networks*, 1997, 10(8): 1345~1351
- 7 Karvanen J, Eriksson J, Koivunen V. Pearson system based method for blind separation. In: *Proc. Second Int Workshop on Independent Component Analysis and Blind Signal Separation*, 2000. 585~590
- 8 Vlassis N, Motomura Y. Efficient source adaptivity in independent component analysis. *IEEE Trans Neural Networks*, 2001, 12: 559~566
- 9 Boscolo R, Pan H, Vwani P. Roychowdhury. Independent Component Analysis Based on Nonparametric Density Estimation. *IEEE Trans Neural Networks*, 2004, 15(1): 154~161
- 10 Silverman B W. *Density Estimation for Statistics and Data Analysis*. New York: Chapman and Hall, 1985
- 11 Magdon-Ismael M, Atiya A. Density Estimation and Random Variate Generation Using Multilayer Networks. *IEEE Trans Neural Networks*, 2002, 13(3)
- 12 Barlow H B. Unsupervised learning. *Neural Computation*, 1989, 1: 295~311
- 13 Cichocki A, Unbehauen R. Robust Neural Networks with On-line Learning for Blind Identification and Blind Separation of Sources. *IEEE Trans Circuits and Systems-I: Fundamental Theory and Applications*, 1996, 43: 894~906
- 14 Amari S, Cichocki A, Yang H H. A new learning algorithm for blind signal separation. In: *Touretzky D S, Mozer M C, Hasselmo M E, eds. Advances in Neural Information Processing Systems*, Cambridge, MA: MIT Press, 1996, 8(1): 757~7636

(上接第180页)

可见,运算结果输出可以正确地传递给下一级门电路信号Y的输入。那么,运算结果又是如何传递给下一级门电路信号X的输入是我们解决的问题。这里只需要增加一次“或”运算 $X=0 \vee F$ 即可,逻辑值“0”是非常容易实现的,就是初始化的状态。不难发现,如果下一级运算是“或”运算,则传递给输入信号X和Y的溶液可以直接混合,无需单独进行输入信号X的传递过程。所以,用多发夹结构的DNA分子表示的多值逻辑电路,采用“与-或”逻辑结构,是最简单的,不仅“与”门运算结果的输出无需求补转化,而且“或”门的所有输入信号都可以同时传递。

结束语 本文首次给出了一个基于DNA分子计算的三值逻辑电路模型,阐述了“与”、“或”运算的生物实现过程。因为这种发夹型DNA计算模型是基于表面方式的,所以随着生物芯片技术的飞速发展,构建这种基于DNA分子的多值逻辑计算芯片将成为可能。由于DNA分子计算本身的优点,该模型用于制造大规模低能耗的逻辑阵列电路是非常适合的,但是模型也有不足之处,(1)“与”运算过程中存在杂交竞争的问题;(2)运算过程中利用加热来解链的方法有待改进,例如文[4]的诱导型“发夹”结构,在表面上加入或清洗吡

啉二聚物,“发夹”闭合与打开非常容易控制。另外,由于DNA分子计算本身的复杂性和不确定性,模型的具体应用有待生物技术的进一步提高。

参考文献

- 1 Post E L. Introduction to a general theory of elementary propositions [J]. *Amer. J. Math.*, 1921, 43:163~185
- 2 Adleman L, et al. Molecular computations to combinatorial problems [J]. *Science*, 1994, 266:1021~1024
- 3 刘文斌,朱翔鸥,王向红,陈丽春. DNA 计算的研究进展[J]. *电子学报*, 2006, 34(11):2053~2057
- 4 Liu Wenbin, et al. A new DNA computing model for the NAND gate based on induced hairpin formation [J]. *BioSystems*, 2004, 77(1-3):87~92
- 5 Cukras A R, Faulhammer D, Lipton R J, Landweber L F. Chess games: A model for RNA-based computation [J]. *Biosystems*, 1999, 52:35~45
- 6 Yin Zhixiang, Zhang Fenyue, Xu Jin. DNA computing based on molecular beacons [J]. *Journal of Biomathematics*, 2003, 4(18): 497~501(in Chinese)
- 7 Nakatani K, Sando S, Saito I. Improved Selectivity for the Binding of Naphthylridine Dimer to Guanine-Guanine Mismatch. *Bioorg [J]. Med. Chem. Soc.*, 2001, 9:2381~2385
- 8 Nakatani K, Sando S, Saito I. Scanning of guanine-guanine mismatches in DNA by synthetic ligands using surface plasmon resonance assay [J]. *Nat. biotechnol.*, 2001, 19:51~55