基于 NCP 门库的一维量子行走可逆逻辑电路

朱皖宁[1],陈汉武[1,3],李志钢[1],阮 越[1,2],王 冬[1],周 刚[1]
(1.东南大学计算机科学与工程学院,江苏南京 210096;2.安徽工业大学计算机学院,安徽马鞍山 243005;
3.东南大学计算机网络和信息集成教育部重点实验室,江苏南京 210096)

摘 要: 本文提出了基于 NCP 门库的一维量子行走可逆逻辑电路设计方案.根据一维量子行走的特点,电路被划分为投掷硬币和 S 操作两个部分;文章详细分析一维量子行走,对其行为数学建模,巧妙利用可控加减电路实现了 S 操作.目前对于量子行走算法的研究多数局限于数学理论和数理解析层面,在量子电路理论层面对量子行走算法的研究为数不多.本文利用原始递归给出了一维量子行走中每一步在量子电路理论层面上的数学表达式;提出的可逆逻辑电路描述了一维量子行走的最基本操作,并且将其使用模块化表示,使一维量子行走算法的研究从理论到实现上前进了一步.

关键词: 一维量子行走; NCP 门库; 可逆逻辑; 可控加减电路; 原始递归
中图分类号: TP387; TN911.73 **文献标识码:** A **文章编号:** 0372-2112 (2013)01-0091-07
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2013.01.017

Reversible Logic Circuit for One-Dimensional Quantum Walk Based on NCP Quantum Gates Library

ZHU Wan-ning[1], CHEN Han-wu[1,3], LI Zhi-gang[1], RUAN Yue[1,2], WANG Dong[1], ZHOU Gang[1]
(1. *School of Computer Science and Engineering, Southeast University, Nanjing, Jiangsu* 210096, *China*;
2. *School of Computer Science, Anhui University of Technology, Ma'anshan, Anhui* 243005, *China*;
3. *Key Laboratory of Computer Network and Information Integration of Ministry of Education, Southeast University, Nanjing, Jiangsu* 210096, *China*)

Abstract: The design proposal of reversible logic circuit for one-dimensional quantum walk based on NCP quantum gates library is presented. According to the features of the one-dimensional quantum walk, this circuit is divided to two parts, one part is quantum coin tossing and the other part is S operation. Besides the work above, this paper thoroughly analyses the one-dimensional quantum walk and builds a mathematical model of the one-dimensional quantum walk and uses controlled add-sub circuit to realize the S operation. At present the researches on quantum walk often limited to the mathematical theory and analysis. Depend on the primitive recursive, Mathematical expression of every step of the one-dimensional quantum walk is given in this paper; the circuit studied in this paper describes element operation of the one-dimensional quantum walk, and make this modular which contribute to the realization for the algorithm of one-dimensional quantum walk.

Key words: one-dimensional quantum walk; NCP quantum gates library; reversible logic; controlled add-sub circuit; primitive recursive

1 引言

随着量子力学在物理学界不断的普及应用,其影响已经深入到各个学科之中.基于量子信息计算理论的量子算法在一些 NP 问题的理论解决方案上显示出良好的性能.1994 年,基于傅里叶变换方法,P W Shor 提出了多项式时间复杂度的大数质因子分解量子算法[1].从此以后高效的量子算法研究就成为了重要的研究课题.1997 年 Grover 又提出了量子搜索算法,在经典的搜索算法上进行了二次加速[2].

关于量子行走的思想是在 1993 年被提出的[3],但是由于一直没有发现很好的算法应用,所以在以后相当长的时间内没有能够引起广泛关注.随着对量子行走的深入研究,越来越多的基于量子行走的算法被提出[4~6].

收稿日期:2012-04-21;修回日期:2012-10-12
基金项目:国家自然科学基金(No.61170321);高等学校博士学科点专项科研基金(No.20110092110024);计算机网络和信息集成教育部重点实验室基金

量子行走开创了量子算法研究的新分支.量子行走是在量子计算机上模仿经典随机行走的算法.利用干涉现象,量子行走能够获得比经典随机行走更高的运行效率.例如黑盒子问题[7,8],使用量子行走对经典的算法进行了二次加速.并且许多量子行走算法都对经典算法实现了多项式加速效果.

量子行走最简单的变化就是在一条无限距离的线上进行双向的随机行走,也可以称之为一维量子行走.每次行走之前都会根据一枚无约束硬币投掷的值来判断向左走还是向右走.这样的随机行走可以推广到更加复杂的维数或者是有限和无限图中,并且在计算机科学领域里已经有了很多有趣的应用.在本文中我们主要考虑一维量子行走.

量子行走在量子算法中有着非常巨大的研究意义:第一,量子行走是一种利用量子信息态叠加性的通用量子算法,也就是说任何可以在量子计算机上解决的问题都可以使用量子行走来解决[9],即量子行走可以有效率地模拟任意量子算法.第二,量子行走的计算能力强大,可以充分体现量子算法的并行性.第三,量子行走的研究可以给我们提供量子算法复杂度的分析工具[10].

1985 年 Deutsch 提出了量子图灵机(QTMs)[11]作为量子计算机的数学计算模型,1989 年 Deutsch 又提出了量子电路[12]作为量子计算机的物理实现模型.直到现在,以上的两个模型仍然一直在用于研究量子计算机的各种问题.QTMs 模型被用于对一个编程计算的量子计算机进行建模,而且作为量子算法的数学模型来研究其效率.而量子电路则主要作为量子计算机的物理实现模型来进行研究.从量子计算机的两种模型可以看出,对于量子计算机算法的研究不仅要在数学理论层面进行,还需要在量子电路理论层面进行.例如 P W Shor 的大数质因子分解算法已经有了详细的量子电路实现方法,Grover 的量子搜索算法也有初步的电路图解.当前人们对于量子行走的研究,还局限于数学理论层面,没有进行量子电路理论研究.这样无论是对量子行走继续深入研究,还是对量子行走进行仿真测试都很难顺利的进行.本文基于这一难点,着重研究了一维量子行走通过量子可逆逻辑电路进行表示的方法.

量子行走算法是由一组酉算子复合而成,酉算子对应了可逆门,因此量子行走算法可以由一组可逆门级联而成.量子可逆逻辑综合将可逆门由一组门库综合而成,常用的门库有 NCT 门库,NCV 门库和 NCP 门库.常用的可逆逻辑综合算法都是基于全局状态进行计算,例如:真值表法、置换群法和 Reed-Muller 展开式法等.这些算法都需要预先获得所有逻辑电路输入输出的真值才能计算,因此对于量子比特数较大的可逆逻辑电路综合问题很难求解.而一维量子行走是在量子比特数较大的一个空间里进行,所以很难用常用可逆逻辑综合算法来进行可逆逻辑综合.本文提出一种通过原始递归对全局电路进行递归分解的方法,给出了一维量子行走电路的数学模型,由 NCP 门库级联构造而成的一维量子行走电路,并且将电路进行模块化,从而解决了一维量子行走可逆逻辑电路综合问题.由电路模块直接生成量子行走算法,可以使算法由硬件提速,拥有更高的效率;在大规模生产芯片时有效降低成本;凸显量子行走算法背后的数学原理,使以后的研究道路更加宽阔.

2 背景知识

2.1 一维量子行走简介

在一维坐标下的经典随机行走是这样的一个过程:从任意的起点 N 开始,每次行走之前都会进行一次概率判定,假设以概率为 p 向左走$(N-1)$,则有以概率为 $1-p$ 向右走$(N+1)(0\leqslant p\leqslant 1)$.当 p 取 0 或者 1 时就丧失了随机性质.特别的,在研究过程中取 $p=\frac{1}{2}$.在一次判定后进行的一次行走可以称之为一步,经过 m 步以后的行走将会使移动到某些坐标点上的概率远大于其他的坐标点,从而获得算法的解.

从经典的随机行走进行直接的改造获得量子行走,就必须要定义一个操作,使得在一步中有一定的概率向左,又有一定概率向右,同时还有一定概率原地不动.定义酉算子 U 如下:

$$U\circ|N\rangle=a|N+1\rangle+b|N\rangle+c|N-1\rangle \qquad (1)$$

其中 a,b,c 满足归一化:$a^2+b^2+c^2=1$.

然而这样定义的 U 操作,如果要保证其随机性质,则在物理上无法实现.因为这样的操作必然不是酉操作[10].可以由以下数学公式定义 U 算子的性质:

若要使 U 算子为酉操作,则必然要满足以下 3 个公式之一:(排他性操作)

(1) $a=1,b=0,c=0$

(2) $b=1,a=0,c=0$

(3) $c=1,a=0,b=0$

也就是说,若要使 U 操作为酉操作,则会使量子行走丧失随机性从而变的毫无实际意义.在文献[10]中提出了 H 行走的概念,将一步行走分为两个部分 $U=S\circ C$,从而能够定义酉操作 U,使得量子行走成为可能.

首先定义一个量子行走硬币 $C=H_c$ 来确定量子行走的概率.定义量子行走硬币 $H_c\in(|0\rangle,|1\rangle)$,将 H_c 放在输入值,即坐标值的后面,表示如下:$|N,H_c\rangle$,其中 N 为坐标值,用 n 位二进制数来表示,$N=2^n$.

然后在量子行走的第一部分,投掷量子行走硬币

C.方法是对 H_c 位做 Hadamard 操作,因此使用量子硬币 H_c 的一维量子行走又称为 Hadamard 行走. H_c 经过变换后得到的叠加态如式(2)所示:

$$\frac{1}{\sqrt{2}}(|N,0\rangle \pm |N,1\rangle) \tag{2}$$

如式(2)表示的 qubit 状态,等于将 H_c 投掷一半概率为 $|0\rangle$,一半概率为 $|1\rangle$.

最后在量子行走的第二部分,定义 S 操作如下:

$$\begin{cases} S\circ|N,0\rangle = |N+1,0\rangle \\ S\circ|N,1\rangle = |N-1,1\rangle \end{cases} \tag{3}$$

式(3)所定义的 S 操作实现了以概率控制向左或者向右行走.将两个部分联合起来操作,就成为一维量子行走的一步.

2.2 NCP 门库和 Hadamard 门简介

NCP 门库包含 3 个基本门:NOT 门,Control-NOT 门,Peres 门. 其中 Control-NOT 门又称为 Feynman 门,是 Feynman 在 1985 年提出的基本可逆门[13],后文简称 FG. Peres 门是 Peres 在 1985 年提出的基本可逆门[14],后文简称 PG.这 3 个门构成了通用门,即可以综合 01 电路中的所有可逆逻辑.

如图 1~3 所示的 NCP 门库中三个基本电路的电路图和表达式在后文中即可构造出低开销的一维量子行走电路.本文设量子电路从上到下分别为第一位、第二位、……、第 n 位,左边为输入右边为输出.

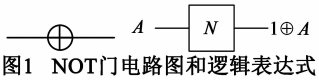

图1　NOT门电路图和逻辑表达式

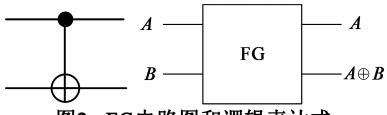

图2　FG电路图和逻辑表达式

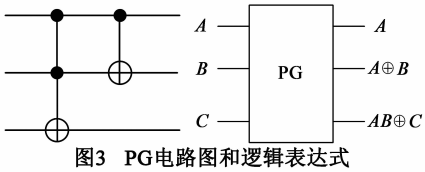

图3　PG电路图和逻辑表达式

Hadamard 门可以实现 qubit 的 X 基和 Z 基转换,将 $|0\rangle$ 和 $|1\rangle$ 变成以概率 1/2 进行叠加的叠加态 $\frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$,在后文中简称为 H 门,如图 4 所示.

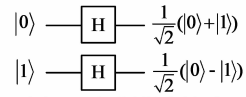

图4　H门逻辑表达式

3　一维量子行走可逆逻辑综合

量子行走的量子态所在空间 H 由坐标位置空间 H_L 和硬币空间 H_C 构成, $H = H_L \otimes H_C$. 量子行走一步操作是定义在希尔伯特空间的一个酉算子 U.假定量子行走算法的初态为 ψ_0,那么进行一次行走可以定义为 $\psi_1 = U\psi_0$.当算法需求行走 m 步时, $\psi_m = U^m\psi_0$.其中 $\psi_m = \{|N,g\rangle | |N\rangle \in H_L, |g\rangle \in H_C\}$.可以很明显的看出,量子行走算法是一个迭代的过程,从可逆逻辑综合的角度上来说,只需要实现一步行走算子 U,通过多次重复算子 U 就可以实现任意的量子行走.根据第 2.2 节所述,算子 U 需要分成硬币算子 C 和行走算子 S 两部分,定义为 $U = S\circ C$.量子行走可逆逻辑综合的关键就是分别实现硬币算子 C 和行走算子 S 的可逆逻辑电路,然后将其组合为量子行走电路.

对于本文所讨论的一维量子行走的 U 操作,有两个限定条件:

(1)由于一维行走只存在两个方向,因此硬币空间为二维.

(2)由于一维行走的位置是向正负两个方向不断延伸,因此坐标位置的改变必须是对坐标位置二进制数的整体进行计算,而不是单单只计算其中的某一位(图上的量子行走只改变坐标位置的其中一位二进制数).

因此硬币空间选取为 $H_C = \{|0\rangle, |1\rangle\}$.实现量子硬币 C 算子需要根据具体要解决的问题来定.一般来说是一个规模为输入的多项式复杂度的 ORACLE 电路复合一个硬币翻转算子 $C = O\circ C'$(在附录中以 Hadamard 行走举例,硬币算子初始化为 $|0\rangle$,然后让其通过 H 门,即 C = H 门).

构造一维量子行走的电路重点在于构造 S 操作.分析如公式(3)所定义的 S 操作,实际上就是将硬币位 g 作为控制端,当 g 为 $|0\rangle$ 时,对 n 位二进制数 N 做 +1 操作;当 g 为 $|1\rangle$ 时,对 n 位二进制数 N 做 −1 操作.所以 S 操作的本质是一个 n 位的受控加减器.

量子加法器和减法器已经被提出,但是用一位控制端控制加法和减法变换的加减器到现在还没有.如果直接将加法器和减法器加以融合,用一位控制端来进行切换,其开销也会非常巨大.由于 S 操作实际上只做 +1 和 −1 的操作,因此并不需要实现复杂的加法和减法,这就使得低开销受控加减器能够实现.

4　S 操作的可逆逻辑构造

设输入 $A_n = (a_n \cdots a_2 a_1)$ 为一个 n 位 2 进制数,对其进行 S 操作,就是要实现 n 量子位受控的 +1 和 −1 操作.实现电路的方法通常为递归法,这里更加严格的可以使用原始递归函数来生成可控加减函数.

定义 1　原始递归　设 g 是 2 元函数, k 是一个常数,函数 h 由下述等式给出:

$$\begin{cases} h(0)=k \\ h(t+1)=g(t,h(t)) \end{cases}$$

则可以称 h 是由 g 经过原始递归运算得到的.

下面对 +1 操作电路和 −1 操作电路分别进行分析.

4.1 +1 操作电路实现

假设一元函数 Add 为所求的 +1 函数.首先看对 1 位 2 进制数 A_1 进行 +1 操作,其本质就是做模 2 加 1,当且仅当 A_1 的值为 1 时,进位为 1,否则进位为 0.公式表示如下:

$$\mathrm{Add}(|c_1,A_1\rangle)=|c_1\oplus a_1,a_1\oplus 1\rangle \tag{4}$$

在式(4)中 c_1 表示 A_1 的进位,初始化为 0.由此可知,当 2 进制数只有 1 位时是可以使用函数 Add 进行运算.n 位 2 进制数可以由 1 位 2 进制数和二元函数 Add_0 原始递归而成.假设 $\mathrm{Add}(|c_n,A_n\rangle)$可以运算 n 位 2 进制数 A_n 的 +1 运算,其中 c_n 为 n 位 2 进制数 A_n 的进位,初始化为 0.对 $n+1$ 位 2 进制数 $A_{n+1}=(a_{n+1}a_n\cdots a_2a_1)$做 +1 运算就可以用一个二元函数 Add_0来进行原始递归:

$$\begin{aligned}&\mathrm{Add}(|c_{n+1},A_{n+1}\rangle)\\&=\mathrm{Add}_0(|c_{n+1},a_{n+1}\rangle,\mathrm{Add}(|c_n,A_n\rangle))\end{aligned} \tag{5}$$

问题的关键就是在于求得二元函数 Add_0的函数表达式.由于低位的 n 位数都已经计算好,因此不用再次运算,只需要计算第 $n+1$ 位数 a_{n+1}和进位 c_{n+1}的值.对 a_{n+1}和 c_n 做模 2 加法就可以得到第 $n+1$ 位的值,当且仅当 a_{n+1}和 c_n 都为 1 时,c_{n+1}才为 1,否则为 0.同样的,将 c_{n+1}初始化为 0,可以用数学公式表示如下:

$$\begin{aligned}&\mathrm{Add}_0(|c_{n+1},a_{n+1}\rangle,\mathrm{Add}(|c_n,A_n\rangle))\\&=|c_{n+1}\oplus(a_{n+1}\wedge c_n),a_{n+1}\oplus c_n,A_n\rangle\end{aligned} \tag{6}$$

由式(5)和式(6)可知,假设 n 位二进制数可以用函数 Add 来运算,那么 $n+1$ 位二进制数同样可以用 Add 来运算.因此 Add 函数可以由 1 位二进制数和二元函数 Add_0原始递归而成:

$$\begin{cases}\mathrm{Add}(|c_1,A_1\rangle)=|c_1\oplus a_1,a_1\oplus 1\rangle\\ \mathrm{Add}(|c_{n+1},A_{n+1}\rangle)\\ \quad=\mathrm{Add}_0(|c_{n+1},a_{n+1}\rangle,\mathrm{Add}(|c_n,A_n\rangle))\end{cases} \tag{7}$$

下面展示如何用 PG 构造式(4),将第一位作为辅助位输入为$|1\rangle$,第二位变量输入为$|a_1\rangle$,第三位辅助位输入为进位,初始化为$|0\rangle$,构造如图 5.

如图 5 所示的电路图可以明显看见,将变量$|a_1\rangle$输入后,在电路的第二和第三位上输出了式(4)中所得的结果(由于 c_1 初始化为 0,可得 $0\oplus a_1=a_1$).

同样,使用 PG 还可以构造出式(6),将低位的进位$|c_n\rangle$作为辅助位放在第一位输入,将第 $n+1$ 个分量$|a_{n+1}\rangle$作为第二位的输入,将高位的进位$|c_{n+1}\rangle$作为辅助位放在第三位上输入并初始化为$|0\rangle$,构造如图 6.

如图 6 所示的电路图明显可得式(6)在高位所求的值(第三位输出,由于进位初始化为 0,可得 $0\oplus(a_{n+1}\wedge c_n)=a_{n+1}\wedge c_n$),低位不发生变化.将图 5 和图 6 所示的电路图进行综合即可得式(7)所求的 Add 函数电路如图 7 所示.

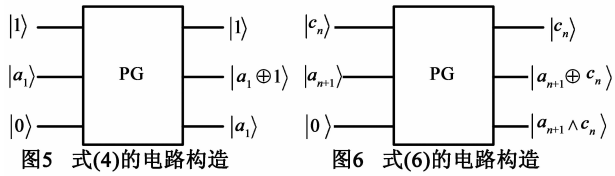

图5 式(4)的电路构造

图6 式(6)的电路构造

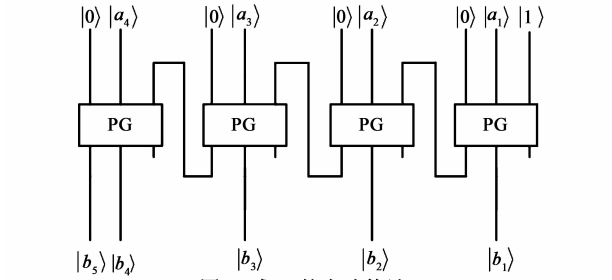

图7 式(7)的电路构造

从如图 7 所示的电路图可见,输入为 4 量子比特的二进制数 $A=(a_4a_3a_2a_1)$和 5 个辅助位,输出为 $B=(b_5b_4b_3b_2b_1)$,较短的输出线表示垃圾位.根据上述电路分析可知,当二进制数位数为 n 时,就需要 n 个 PG 门,$n+1$ 个辅助位;产生 n 个垃圾位.在实际使用中,因为可以确定输入变量所需要的空间大小,输出变量也应该和输入变量所需要的空间大小相等,所以在输出 B 中,$|b_5\rangle$作为判断溢出错误位,当$|b_5\rangle$为$|1\rangle$时,即表示发生了溢出错误.

4.2 −1 操作电路实现

类似 +1 操作,定义一元函数 Sub 为 −1 函数.同样的 Sub 函数可以由 1 位二进制数的 −1 运算原始递归而来.分析 1 位二进制数情况下的 −1 运算:对于 $A_1=a_1$ 来说,对 a_1 做减法,实际还是进行模 2 加 1 运算,同时为了表现是否需要向高位借位,还需要一个辅助借位 l_1,将借位 l_1 初始化为 0.当且仅当 a_1 为 0 时 l_1 才为 1.数学公式表现如下:

$$\mathrm{Sub}(|l_1,A_1\rangle)=|l_1\oplus(a_1\oplus 1),a_1\oplus 1\rangle \tag{8}$$

从式(8)可以看出,在 1 位二进制数下可以构造 Sub 函数,类似 Add 函数,下面介绍如何从 1 位二进制数递归构造 n 位二进制数的 Sub 函数.

设 Sub 函数可以计算 n 位二进制数的 −1 运算 $\mathrm{Sub}(|l_n,A_n\rangle)$,其中借位 l_n 初始化为 0.那么对 $n+1$ 位二进制数 $A_{n+1}=(a_{n+1}a_n\cdots a_2a_1)$做 −1 运算可以用二元函数 Sub_0来进行构造:

$$\mathrm{Sub}(|l_{n+1},A_{n+1}\rangle)$$

$$= \mathrm{Sub}_0(|l_{n+1}, a_{n+1}\rangle, \mathrm{Sub}(|l_n, A_n\rangle)) \tag{9}$$

类似 Add 函数,低位的 n 位都已经计算好,因此 Sub_0不用再次计算,只需要计算 l_{n+1}和 a_{n+1}的值即可.第 $n+1$ 位的值为 $a_{n+1}\oplus l_n$,l_{n+1}初始化为 0,当且仅当 a_{n+1}为 0 并且 l_n 为 1 时,l_{n+1}取值为 1.Sub_0函数用公式表示如下:

$$\begin{aligned}&\mathrm{Sub}_0(|l_{n+1}, a_{n+1}\rangle, \mathrm{Sub}(|l_n, A_n\rangle))\\ &= |l_{n+1}\oplus((a_{n+1}\oplus 1)\wedge l_n), a_{n+1}\oplus l_n, A_n\rangle\end{aligned} \tag{10}$$

综上所述,Sub 函数可以由式(8)、(9)原始递归而来,用公式表示如下:

$$\begin{cases}\mathrm{Sub}(|l_1, A_1\rangle) = |a_1\oplus 1, a_1\oplus 1\rangle\\ \mathrm{Sub}(|l_{n+1}, A_{n+1}\rangle)\\ \quad = \mathrm{Sub}_0(|l_{n+1}, a_{n+1}\rangle, \mathrm{Sub}(|l_n, A_n\rangle))\end{cases} \tag{11}$$

下面展示如何利用 PG 和 NOT 门来构造式(8),将第一位输入辅助借位$|1\rangle$,第二位输入当前变量$|a_1\rangle$,第三位输入借位$|l_1\rangle$,初始化为 0.构造如图 8 所示.

从如图 8 所示的电路可以看出输出的第一位为垃圾位,第二位为所求变量 a_1 的值,第三位为借位 l_1 的值.下面继续使用 PG 和 NOT 门构造式(10)的高位.将第一位输入低位的借位$|l_n\rangle$,第二位输入为第 $n+1$ 个分量$|a_{n+1}\rangle$,第三位输入高位借位$|l_{n+1}\rangle$,初始化为 0.构造电路图如图 9 所示.

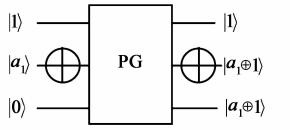

图8　式(8)的电路构造

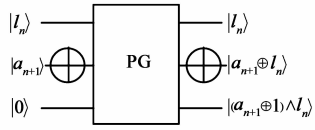

图9　式(10)的电路构造

从如图 9 所示的电路中可以看出输出的第一位为垃圾位,第二位为所求的第 $n+1$ 个分量,第三位为高位的进位.将图 8 和图 9 进行综合即可得 Sub 函数的电路图如图 10 所示.

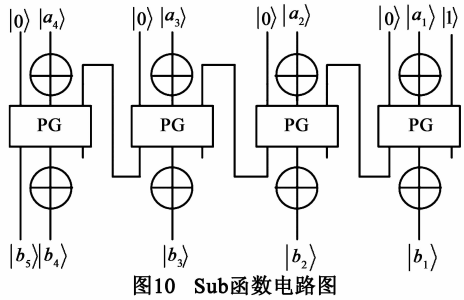

图10　Sub函数电路图

从如图 10 所示的电路图可见,输入为 4 量子比特的二进制数 $A=(a_4a_3a_2a_1)$和 5 个辅助位,输出为 $B=(b_4b_3b_2b_1)$,较短的输出线表示垃圾位.根据上述电路分析可知,当二进制数位数为 n 时,就需要 n 个 PG 门,$n+1$ 个辅助位,产生 n 个垃圾位.类似 Add 函数电路,当$|b_5\rangle$为$|1\rangle$时表示了向高位借位,即表示发生了溢出错误.

4.3　S 和 C 操作电路综合

将 Add 电路和 Sub 电路用量子硬币 C 来进行控制切换,就可以完成所需要构造的 S 操作.观察图 7 和图 10,区别只在于对 PG 的第二位输入和输出加上了 NOT 门,那么只需要用量子硬币 C 作为控制端,PG 的第二位输入和输出作为受控端加上 FG,就可以完成两者的结合.

分析图 11 所示的电路,为了能够让每个分量都受到控制,每个分量都需要将控制端 C 进行两次扇出.每次扇出都需要 1 个辅助位和一个 FG.构造一个能对 n 量子比特$A_n=(a_na_{n-1}\cdots a_2a_1)$进行 S 操作的电路,就需要 $\Theta(n)$个 PG,$\Theta(4n)$个 FG,$\Theta(3n)$个辅助位,产生了 $\Theta(3n)$个垃圾位,因此图 11 所示的电路 cost 为 $\Theta(8n)$.

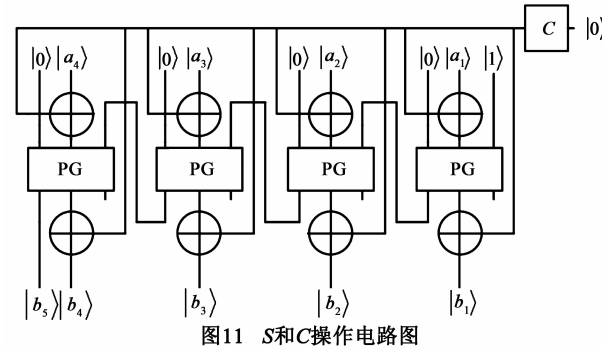

图11　S和C操作电路图

将辅助位和垃圾位固定,只进行 n 量子比特输入和 $n+1$ 量子比特输出,即可构造出一步量子行走 U 门.每经过一次 U 门运算,即可视作一维量子行走走了一步.

如图 12 所示的 U 门,输入为 n 量子比特的数$A_n=(a_na_{n-1}\cdots a_2a_1)$,输出为 n 量子比特的数 $B_n=(b_1b_2\cdots b_n)$和错误检测位 e,当 e 为 1 时,表示发生了溢出错误;e 为 0 时,输出值有效.

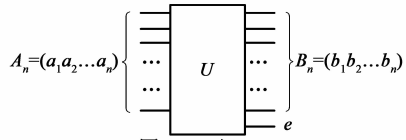

图12　U门

5　总结

量子计算带来了并行计算能力,将许多的指数级复杂度算法降低为多项式级复杂度,因此研究量子计算是计算机理论发展的趋势之一.一维量子行走是当前量子计算的重要研究课题,具有很重要的研究意义:是一种通用的量子算法理论,而且有着量子计算高效的运算能力,更重要的是可以为量子算法的复杂度提供分析工具.在最近几年的量子计算研究中,已经有多篇论文指出量子行走是打开量子计算大门的一条崭新

的大道[6~10].

本文对一维量子行走的具体构造进行了研究.与经典算法不同,量子算法的物理实现模型最终是由一串可逆门级联而成,若不能有开销较小,速度较快的可逆门综合方法来对量子算法进行表示,那么算法也仅仅停留在数学理论层面上.本文对 S 操作在可逆逻辑层面上进行了分析,给出了原始递归递推的数学构造公式.本文最后使用了当前可逆逻辑综合中常用的 NCP 门库对一维量子行走的 S 操作进行了低开销可逆逻辑综合,并将行走的每一步定义为一个 U 门,同时给出了计算溢出时的判断方法.

在文献[10]中,将 C 算子设定为 H 门,就实现了 Hadamard 行走.若需要实现更加复杂的量子算法,则需要根据具体的情况得出 C 算子的具体函数表达式.这也是当前量子算法研究的重点问题之一.在将来的研究工作中将会研究具体量子行走算法的可逆逻辑综合电路.

附录 一维量子行走实例

假设初态 $\psi_0 = |A_4, 0\rangle$,其中 $A_4 = (a_4a_3a_2a_1) = (0111)$,要进行文献[10]中的 Hadamard 行走.显然可知此时在位置 0111 的概率 $p_{0111} = 1$,其余都为 0.现在令硬币算子 C = H 门.每一步行走都先通过硬币算子 H 门,再通过位置算子 S 门.在测量行走最终落在位置上的概率时,只投影到位置空间 H_L 上,即不考虑硬币状态的不同,只要位置向量相同就等于落在同一个点上.

Hadamard 行走第一步的结果为:

$$|01110\rangle \xrightarrow{I^4\otimes H} 1/\sqrt{2}(01110\rangle + |01111\rangle)$$
$$\xrightarrow{S\otimes I} 1/\sqrt{2}(|10000\rangle + |01101\rangle)$$

在位置 1000 的概率为 $p_{1000} = (1/\sqrt{2})^2 = 0.5$

在位置 0110 的概率为 $p_{0110} = (1/\sqrt{2})^2 = 0.5$

Hadamard 行走第二步的结果为:

$$1/\sqrt{2}(|10000\rangle + |01101\rangle)$$
$$\xrightarrow{I^4\otimes H} 1/2\begin{pmatrix} |10000\rangle + |10001\rangle \\ +|01100\rangle - |01101\rangle \end{pmatrix}$$
$$\xrightarrow{S\otimes I} 1/2\begin{pmatrix} |10010\rangle + |01111\rangle \\ +|01110\rangle - |01011\rangle \end{pmatrix}$$

在位置 1001 的概率为 $p_{1001} = (1/2)^2 = 0.25$

在位置 0111 的概率为 $p_{0111} = (1/2)^2 + (1/2)^2 = 0.5$

在位置 0101 的概率为 $p_{0101} = (1/2)^2 = 0.25$

Hadamard 行走第三步的结果为:

$$1/2(|10010\rangle + |01111\rangle + |01110\rangle - |01011\rangle)$$
$$\xrightarrow{I^4\otimes H} 1/2\sqrt{2}\begin{pmatrix} |10010\rangle + |10011\rangle \\ +|01110\rangle - |01111\rangle \\ +|01110\rangle + |01111\rangle \\ -|01010\rangle + |01011\rangle \end{pmatrix}$$
$$\xrightarrow{S\otimes I} 1/2\sqrt{2}\begin{pmatrix} |10100\rangle + |10001\rangle \\ +|10000\rangle - |01101\rangle \\ +|10000\rangle + |01101\rangle \\ -|01100\rangle + |01001\rangle \end{pmatrix}$$
$$= 1/2\sqrt{2}\begin{pmatrix} |10100\rangle + |10001\rangle \\ +2|10000\rangle - |01100\rangle + |01001\rangle \end{pmatrix}$$

在位置 1010 的概率为 $p_{1010} = (1/2\sqrt{2})^2 = 0.125$,

在位置 1000 的概率为 $p_{1000} = (1/2\sqrt{2})^2 + (2/2\sqrt{2})^2 = 0.625$

在位置 0110 的概率为 $p_{0110} = (1/2\sqrt{2})^2 = 0.125$

在位置 0100 的概率为 $p_{0100} = (1/2\sqrt{2})^2 = 0.125$

可以很明显的看到走到第三步时已经出现了量子行走所特有的干涉现象.这个例子很好的验证了组合电路运用 H 门作为硬币算子后,成功的实现了 3 步 Hadamard 行走.

参考文献

[1] P W Shor. Algorithms for quantum computation: discrete logarithms and factoring[A]. Proceedings of the 35th Annual Symposium on Foundations of Computer Science [C]. Los Alamitos, California : IEEE Press, 1994. 124 – 134.

[2] L K Grover. Quantum mechanics helps in searching for a needle in a haystack[J]. Physical Review Letters, 1997, 79(2): 325 – 328.

[3] Y Aharonov, L Davidovich, N Zagury. Quantum random walks [J]. Physical Review A, 1993, 48(2): 1687 – 1690.

[4] A Ambainis. Quantum search algorithms[J]. SIGACT News, 2004, 35(2): 22 – 35.

[5] A Ambainis. Quantum walk algorithm for element distinctness [A]. Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science[C]. USA: IEEE Press, 2004. 22 – 31.

[6] F Magniez, A Nayak, J Roland, M Santha. Search via quantum walk[A]. Proceedings of the 39th Annual ACM Symposium on theory of Computing[C]. USA: ACM Press, 2007. 575 – 584.

[7] A M Childs, R Cleve, E Deotto, E Farhi, S Gutmann, D A Spielman. Exponential algorithmic speedup by quantum walk [A]. Proceedings of the 35th ACM Symposium on Theory of Computing[C]. USA: ACM Press, 2003. 59 – 69.

[8] A M Childs, L J Schulman, U V Vazirani. Quantum algorithms for hidden nonlinear structures [A]. Proceedings of the 45th IEEE Symposium on Foundations of Computer Science [C]. USA: IEEE Press, 2007. 395 – 404.

[9] A M Chids. Universal computation by quantum walk[J]. Physical Review Letters, 2009, 102(18): 180501.

[10] A Ambainis, E Bach, A Nayak, A Vishwanath, J Watrous. One-dimensional quantum walks[A]. Proceedings of the 33rd

ACM Symposium on the Theory of Computing [C]. New York: ACM, 2001. 60 – 69.
[11] D Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer[J]. Proceedings of the Royal Society of London Series A, 1985, 400(1818): 96 – 117.
[12] D Deutsch. Quantum computational networks[J]. Proceedings of the Royal Society of London Series A, 1989, 425(1868): 73 – 90.
[13] Feynman R. Quantum mechanical computers[J]. Foundations of Physics, 1986, 16(6): 507-531. (Originally appeared in Optics News, 1985, 11(2): 11 – 20.)
[14] Peres A. Reversible logic and quantum computers[J]. Physical Review: A, 1985, 32(6): 3266 – 3276.

作者简介

朱皖宁　男,1983 年 1 月生,江苏南京人. 2005 年毕业于东南大学计算机科学与工程学院. 2006 年在东南大学计算机科学与工程学院就读硕士,方向为系统结构专业,2010 年进入东南大学计算机科学与工程学院,现为博士生,从事量子计算与量子可逆逻辑方面的有关研究.
E-mail: granny025@163.com

陈汉武　男,1955 年 11 月生,江苏南京人. 教授,博士生导师. 现从事量子信息与量子计算方向研究.
E-mail: hanwu_chen@163.com

李志钢　男,1985 年 8 月生,山西省阳泉人. 2009 年 7 月毕业于同济大学电子与信息工程学院. 2009 年 9 月进入东南大学计算机科学与工程学院,现为硕博连读生,从事量子计算与量子可逆逻辑方面的有关研究.

阮　越　男,1972 年 8 月生,安徽马鞍山人. 东南大学在职博士生,安徽工业大学讲师,主要研究领域为量子计算和量子算法.