

# 基于状态向量表示的快速量子仿真算法

陈 雄<sup>1,2</sup>, 陈汉武<sup>1,2</sup>, 刘志昊<sup>1,2</sup>, 李志强<sup>3</sup>

(1. 东南大学计算机科学与工程学院, 江苏南京 211189;

2. 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 211189; 3. 扬州大学信息工程学院, 江苏扬州 225009)

**摘 要:** 本文给出了量子门的符号化表示形式, 将  $n$  量子线路表示成一个控制位符号向量和一个低阶受控量子门的组合, 避免了使用高维矩阵的巨大存储开销. 对无循环格雷码生成算法进行扩展, 提出了  $n$  元  $k$  定位二进制数生成算法. 提出了快速量子仿真算法 FQSA, 将输入状态向量分组, 用同一酉算子对各组进行矩阵向量乘积运算, 从而快速产生输出状态向量. 相比其他通用量子仿真算法, FQSA 节省了存储空间, 并具有最优的时间复杂度. 仿真 QFT 表明, 较当前最好的分治算法, FQSA 极大降低了运行时间, 提高了可仿真量子比特数.

**关键词:** 量子线路; 仿真技术; 状态向量; 格雷码生成; 快速量子仿真算法

**中图分类号:** TP387 **文献标识码:** A **文章编号:** 0372-2112 (2011) 03-0500-05

## A Fast Quantum Simulation Algorithm Based on State Vector

CHEN Xiong<sup>1,2</sup>, CHEN Han-wu<sup>1,2</sup>, LIU Zhi-hao<sup>1,2</sup>, LI Zhi-qiang<sup>3</sup>

(1. School of Computer Science and Engineering, Southeast University, Nanjing, Jiangsu 211189, China;

2. Key Laboratory of Computer Network and Information Integration of Ministry of Education, Southeast University, Nanjing, Jiangsu 211189, China; 3. College of Information Engineering, Yangzhou University, Yangzhou, Jiangsu 225009, China)

**Abstract:** We introduce symbolic representation for quantum circuit, and show how  $n$ -qubit circuit can be described as a control qubit symbolic vector and a lower-order controlled gate, instead of the representation of higher-order unitary matrix. We extend the loopless Gray code binary generation algorithm and propose  $n$  elements  $k$  position binary generation algorithm. Besides, a fast quantum simulation algorithm (FQSA) is presented, it divides a qubit state vector into several operand groups, each group is multiplied by the same unitary matrix, and then the output can be given quickly. In contrast to any other state vector based algorithm, FQSA requires less memory, and has an optimal time complexity. The experimental result of QFT indicates that FQSA is more time-efficient and it can provide more qubits circuit simulation under the same condition.

**Key words:** quantum circuit; simulation technique; state vector; Gray code generation; fast quantum simulation algorithm

## 1 引言

在经典计算机上可对运行在量子逻辑电路<sup>[1]</sup>上的量子算法进行仿真计算. 采用  $2^n$  维量子状态复向量表示  $n$  量子比特信息, 需要  $2^n$  阶方阵来模拟量子状态的演化. 仿真计算可采用特制的微处理器<sup>[2]</sup>进行高性能向量运算, 也可研制更优的仿真算法. 为降低仿真算法的复杂度, 或者用更有效的数据结构代替状态向量表示, 或者采取适当的压缩技术以减少状态向量和酉算子所需存储空间. 针对前者有稳定子形式方法、矩阵乘积态表示 (MPS) 和带权图状态等<sup>[3]</sup>, 这类数据结构可以在经典计算机上有效地仿真特定的电路. 但据我们所知, 所有通用的量子仿真算法都基于状态向量表示.

迄今主要的仿真算法均采用基于 BDD 的结构来表示酉算子和状态向量, 如 QMDD<sup>[4]</sup> 和 QuIDD<sup>[5,6]</sup>, 仿真中

用到的矩阵向量乘法可以在  $O(2^{2n})$  时间内完成. 此类算法可以有效地仿真具有特定属性的状态向量和量子线路, 如输入为  $n$  量子比特均衡叠加态的 Grover 搜索算法的线路. 但在仿真非平凡线路时, BDD 结构不能有效压缩状态向量和酉算子, 算法的复杂度无法降低. Samoladas 提出使用投影算子符号化地描述量子线路的方法<sup>[7]</sup>提高仿真算法的效率, 但该算法仍然基于 BDD 结构. 最近, 薛希玲等人<sup>[8]</sup>提出了基分治策略的通用仿真算法, 该算法将量子线路抽象为嵌套的受控酉运算, 层层解析量子线路来确定线路上的受控量子门对状态向量的作用, 将门电路递归分解. 在含单量子门和二量子门的线路仿真的最坏时间为  $O(n2^{2n})$ , 相比 BDD 结构算法, 分治算法的时间复杂度具有指数级降低, 但其最坏情形下仍未达到最好的时间复杂度.

在量子门线路的表示上, 本文根据线路结构中量子

连线的状态定义量子门,仅存储相应的受控酉矩阵,避免了高阶酉算子的构造和存储.通过分析量子门的一般形式,得出量子门对输入状态向量的作用实质上是受控酉算子对  $2^n$  维输入状态向量分组作用的结论.提出了  $n$  元  $k$  定位的二进制元组生成算法,快速量子仿真算法 FQSA 基于元组生成算法进行高效的组遍历,每组进行相同受控运算,更新  $2^n$  维输入状态向量中的对应的概率幅,从而快速完成仿真计算.本文分析了 FQSA 的时空开销.最后,通过对 QFT 进行仿真计算,将 FQSA 与目前最好的分治算法进行对比,证明了本文仿真算法的性能优于上述各仿真算法.

## 2 预备知识

采用状态向量表示量子状态为便于后文描述,首先描述了  $n$  元  $k$  定位的二进制元组生成问题.对无循环格雷二进制码生成算法<sup>[9,10]</sup>进行扩展,提出了  $n$  元  $k$  定位的二进制元组生成算法.

### 2.1 二进制元组生成问题

**定义 1** 设  $n$  为二进制数的位数,给定  $k$  个非负整数  $p_0, p_1, \dots, p_{k-1}$ , 满足  $p_0 < p_1 < \dots < p_{k-1} < n$ . 集合  $P = \{p_0, p_1, \dots, p_{k-1}\}$  称为  $k$  定位集,用于指定二进制数的  $k$  个位.

**定义 2** 令  $\bar{P} = \{0, 1, \dots, n-1\} - P$ , 称  $S(n, k) = \{(b_{n-1}b_{n-2}\dots b_0)_2 \mid b_j = 0, \forall j \in \bar{P}\}$  为  $n$  元  $k$  定位二进制数集. 称  $S(n, k)$  为  $n$  元全定位二进制集合,也称二进制  $n$  元组集合<sup>[9,10]</sup>. 本定义中,集合  $S(n, k)$  中的每个二进制数元素的由  $\bar{P}$  指定的  $n-k$  个位皆为 0. 也可将此  $n-k$  个位指定为 0 或 1 的定值.

**定义 3** 设  $\text{baseline}_j$  为二进制整数  $\text{baseline}$  的第  $j$  位,并且对  $\forall j \in P$ , 满足  $\text{baseline}_j = 0$ . 则称  $S(n, k, \text{baseline}) = \{(b_{n-1}b_{n-2}\dots b_0)_2 \mid b_j = \text{baseline}_j, \forall j \in \bar{P}\}$  为带起点  $\text{baseline}$  的  $n$  元  $k$  定位二进制数集.

**性质 1**  $S(n, k)$  包含  $2^k$  个  $n$  位二进制数.

**证明** 由  $|\bar{P}| = |\{0, 1, \dots, n-1\} - P| = n - |P| = n - k$ , 则  $|S(n, k)| = 2^n / 2^{n-k} = 2^k$ .

**性质 2** 生成集合  $S(n, k)$  的时间复杂度至少为  $O(2^k)$ .

**证明** 生成  $k$  定位的二进制数集合包含产生该集合和逐一输出其中的  $2^k$  个元素两种操作,但不要求按元素数值大小有序输出.由于至少需要将  $S(n, k)$  中的  $2^k$  个数全部输出,故其时间复杂度不可能低于  $O(2^k)$ .

### 2.2 二进制 $n$ 元组生成算法

Bitner<sup>[10]</sup>等人提出一种按格雷二进制码顺序产生全定位二进制数的高效算法,Knuth<sup>[9]</sup>称其为无循环格雷二进制码生成算法.令  $(b_{n-1}b_{n-2}\dots b_0)_2$  为  $n$  元组,算法需一个辅助的焦点指针数组  $(f_0, f_1, \dots, f_n)$ .

#### 算法 1 无循环格雷二进制码生成算法

输入:  $(b_{n-1}b_{n-2}\dots b_0)_2$  和辅助数组  $(f_0, f_1, \dots, f_n)$ ;

输出:  $n$  元组集合.

```

{
  for ( $j=0; j < n; ++j$ )  $\{b_j=0; f_j=j;\}$ 
   $f_n=n$ ;
  while (1)  $\{/*$  生成  $n$  元组  $*/$ 
    访问  $n$  元组  $(b_{n-1}b_{n-2}\dots b_0)_2$ ;
     $j=f_0; f_0=0$ ;
    if ( $j < n$ )  $\{f_j=f_{j+1}; f_{j+1}=j+1; b_j=1-b_j;\}$ 
    else break;
  }
}
```

算法证明参见文献[9,10]. 算法 1 每次改变二进制数  $(b_{n-1}b_{n-2}\dots b_0)_2$  中的一位,且均获得一个未访问的二进制数. 由于没有使用递归算法,产生新的二进制数仅需  $O(1)$  时间,故称其为无循环的格雷二进制码生成算法. 算法 1 在 while(1) 循环体中最多进行 5 次赋值运算和一次比较运算,因此,该算法的时间复杂度为  $O(2^n)$ .

### 2.3 $n$ 元 $k$ 定位二进制元组生成算法

给定  $k$  定位集  $P = \{p_0, p_1, \dots, p_{k-1}\}$ , 对算法 1 进行扩展,得到  $n$  元  $k$  定位二进制数集合生成算法.

#### 算法 2 $n$ 元 $k$ 定位二进制元组生成算法

输入:  $(b_{n-1}b_{n-2}\dots b_0)_2$ ,  $P = \{p_0, p_1, \dots, p_{k-1}\}$ , 和辅助数组  $(f_0,$

$f_1, \dots, f_k)$ ;

输出:  $n$  元  $k$  定位二进制数集合.

```

{
  for ( $j=0; j < n; ++j$ )  $b_j=0$ ;
  for ( $j=0; j < k; ++j$ )  $f_j=j$ ;
   $f_k=k$ ;
  while (1)  $\{/*$  生成  $n$  元  $k$  定位元组  $*/$ 
    访问  $n$  元组  $(b_{n-1}b_{n-2}\dots b_0)_2$ ;
     $j=f_0; f_0=0$ ;
    if ( $j < k$ )  $\{f_j=f_{j+1}; f_{j+1}=j+1; b_{p_j}=1-b_{p_j};\}$ 
    else break;
  }
}
```

可见,算法 2 的结束依赖于循环体中数组  $f$  的状态,数组  $f$  的初始化和变化规则同算法 1. 由于此处  $f$  为  $k+1$  维数组,因此,算法 2 将经过  $2^k$  次访问  $(b_{n-1}b_{n-2}\dots b_0)_2$  后跳出循环,并产生  $2^k$  个不同的  $n$  位二进制数. 由于  $p_j \in P$  且  $|P| = k$ , 算法 2 每次循环仅改变  $(b_{n-1}b_{n-2}\dots b_0)_2$  中的第  $p_j$  位,由算法 1 的正确性可知,算法 2 将在  $P$  指定的  $k$  位上以格雷码顺序产生  $n$  元  $k$  定位二进制数集合. 根据性质 2, 算法 2 的时间复杂度达到  $O(2^k)$ . 由定义 3 可知,产生  $S(n, k, \text{baseline})$  只需在算

法 2 中将  $(b_{n-1}b_{n-2}\cdots b_0)_2$  初始化为 baseline, 再进行与算法 2 相同的计算.

### 3 量子线路

利用微观粒子状态表示的信息称为量子信息, 其基本单位是量子比特, 与经典信息不同, 量子比特能以叠加态形式存在, 任何单量子比特均可表示为一个二元向量, 如  $|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$ , 其中  $\alpha$  和  $\beta$  为复数, 满足归一化条件:  $|\alpha|^2 + |\beta|^2 = 1$ .  $n$  量子逻辑门可用  $2^n$  阶复数矩阵表示, 将处于叠加态的  $n$  量子比特送入量子线路进行酉变换, 可实现量子比特的更新.

#### 3.1 通用量子线路的符号表示

一个通用的  $n$  量子比特线路中的每条线有且仅有如下一种状态: 线状态,  $|1\rangle$  控制状态,  $|0\rangle$  控制状态, 以及门作用状态, 分别用 Line, One, Zero 和 Gate 表示, 并对应简写为 L, O, Z 和 G. 其中, 状态 L 表示线上既无控制点也无门, 状态 O 表示当该线上的量子比特为  $|1\rangle$  时进行受控运算, 状态 Z 表示当该线上的量子比特为  $|0\rangle$  时进行受控运算, 状态 G 表示该线上有量子门. 在  $n$  量子比特线路中可以没有控制线状态, 但至少包含一个门作用线. 由于目前除交换门外, 量子门都处于相邻的连线上, 而对交换门的仿真仅需将输入状态向量中的对应值交换即可. 因此, 在仿真计算中, 可先假定量子门分布于线路中相邻连线上, 并单独处理交换门情形, 为简化起见, 本文不考虑此种情况.

由此, 一个  $n$  量子门可表示为  $(s_{n-1}, \cdots, s_1, s_0; U)$ , 其中,  $s_i$  表示线路图中由下至上(下同)第  $i$  条量子连线的状态,  $U$  表示线路中的作用酉矩阵. 例如, 图 1 所示的 5 量子比特线路可表示为  $(O, L, G, Z, O; X)$ , 其中  $X$

$$= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

符号表示法简洁直观, 在仿真计算程序中可以直接使用, 它无需将  $n$  量子线路转换为  $2^n$  阶复数方阵进行存储. 例如, 图 1 所示的门线路采用本表

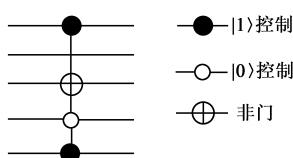


图1 门线路(O,L,G,Z,O;X)

示法仅需存储 4 个复数和 5 维符号向量, 若保存高阶酉矩阵则需存储  $2^5 = 32$  阶复数方阵, 并且需要额外计算出该酉矩阵. 即便采用稀疏矩阵存储仍无法优于本方法.

#### 3.2 多量子比特门作用

对于  $n$  量子比特线路  $(s_{n-1}, \cdots, s_1, s_0; U)$ ,  $|\varphi\rangle = \sum_{i=0}^{2^n-1} a_i |i\rangle$  为任意  $n$  量子比特. 引入 4 类状态线集, 状态线集中的每个元素都是 0 到  $n-1$  的整数, 标识一

条量子连线. 设线路中 G 状态线集为  $S_G = \{g, g+1, \cdots, g+m-1\}$ ; O 状态线集为  $S_O = \{o_0, \cdots, o_{\bar{o}-1}\}$ , 其中  $\bar{o} = |S_O|$ ,  $o_0 < \cdots < \bar{o} < n$ ; Z 状态线集为  $S_Z = \{z_0, \cdots, z_{\bar{z}-1}\}$ , 其中  $\bar{z} = |S_Z|$ ,  $z_0 < \cdots < \bar{z} < n$ .  $S_G, S_O, S_Z$  两两不相交, 且有  $m + \bar{o} + \bar{z} \leq n$ . 设  $S_L$  为线路中 L 状态线集, 有  $S_L = \{l_0, \cdots, l_{k-1}\}$ , 其中  $k = |S_L| = n - (m + \bar{o} + \bar{z})$ , 且  $l_0 < \cdots < l_{k-1} < n$ .

设门线路  $(s_{n-1}, \cdots, s_1, s_0; U)$  对应的  $2^n$  阶酉矩阵为  $W$ . 令  $|i\rangle = |b_{n-1}, \cdots, b_1, b_0\rangle$  为  $n$  量子比特基态, 其中  $(b_{n-1}b_{n-2}\cdots b_0)$  为  $i$  的二进制数. 引入布尔表达式

$$b(i) = b_{o_0} \wedge \cdots \wedge b_{o_{\bar{o}-1}} \wedge \neg b_{z_0} \wedge \cdots \wedge \neg b_{z_{\bar{z}-1}} \quad (1)$$

显然, 仅当  $b_{o_0}, \cdots, b_{o_{\bar{o}-1}}$  全为 1 且  $b_{z_0}, \cdots, b_{z_{\bar{z}-1}}$  全为 0 时  $b(i)$  为 1, 否则,  $b(i)$  为 0. 对于量子线路而言, 当  $b(i)$  为 1 时进行受控运算. 令

$$C = \{i | b(i) = 1, 0 \leq i < 2^n\} \quad (2)$$

$$\bar{C} = \{i | 0 \leq i < 2^n\} - C \quad (3)$$

$$CG = \{i | i \in C \wedge b_{g+m-1} = 0 \wedge \cdots \wedge b_g = 0\} \quad (4)$$

$$\begin{aligned} W|\varphi\rangle &= W\left(\sum_{i=0}^{2^n-1} a_i |i\rangle\right) = \sum_{i=0}^{2^n-1} a_i (W|i\rangle) \\ &= \sum_{i=0}^{2^n-1} a_i (W|b_{n-1}, \cdots, b_1, b_0\rangle) \\ &= \sum_{i=0}^{2^n-1} a_i (|b_{n-1}, \cdots, b_{g+m}\rangle \otimes (U^{b(i)} |b_{g+m-1}, \cdots, b_g\rangle) \\ &\quad \otimes |b_{g-1}, \cdots, b_0\rangle) \\ &= \sum_{i \in C} a_i (|b_{n-1}, \cdots, b_{g+m}\rangle \otimes (U |b_{g+m-1}, \cdots, b_g\rangle) \\ &\quad \otimes |b_{g-1}, \cdots, b_0\rangle) + \sum_{i \in \bar{C}} a_i |b_{n-1}, \cdots, b_1, b_0\rangle \\ &= \sum_{i \in CG} \sum_{j=0}^{2^m-1} a_{i+j \cdot 2^g} (|b_{n-1}, \cdots, b_{g+m}\rangle \otimes (U |j\rangle) \\ &\quad \otimes |b_{g-1}, \cdots, b_0\rangle) + \sum_{i \in \bar{C}} a_i |b_{n-1}, \cdots, b_1, b_0\rangle \\ &= \sum_{i \in CG} |b_{n-1}, \cdots, b_{g+m}\rangle \otimes \left(\sum_{j=0}^{2^m-1} a_{i+j \cdot 2^g} U |j\rangle\right) \\ &\quad \otimes |b_{g-1}, \cdots, b_0\rangle + \sum_{i \in \bar{C}} a_i |b_{n-1}, \cdots, b_1, b_0\rangle \quad (5) \end{aligned}$$

对于式(5)中的  $\sum_{j=0}^{2^m-1} a_{i+j \cdot 2^g} U |j\rangle$ , 令  $\mathbf{a}_i = (a_i, a_{i+2^g}, \cdots, a_{i+(2^m-1) \cdot 2^g})'$ ,  $\mathbf{U} = (u_{ij})_{2^m \times 2^m} = (\mathbf{u}_0, \mathbf{u}_1, \cdots, \mathbf{u}_{2^m-1})'$ , 其中,  $\mathbf{u}_i = (u_{i0}, u_{i1}, \cdots, u_{i(2^m-1)})'$ . 则有

$$\begin{aligned} \sum_{j=0}^{2^m-1} a_{i+j \cdot 2^g} U |j\rangle &= \sum_{j=0}^{2^m-1} a_{i+j \cdot 2^g} \sum_{t=0}^{2^m-1} u_{jt} |t\rangle = \sum_{j=0}^{2^m-1} \sum_{t=0}^{2^m-1} a_{i+j \cdot 2^g} u_{jt} |t\rangle \\ &= \sum_{t=0}^{2^m-1} \left(\sum_{j=0}^{2^m-1} a_{i+j \cdot 2^g} u_{jt}\right) |t\rangle = \sum_{t=0}^{2^m-1} \mathbf{a}_i' \mathbf{u}_t |t\rangle \quad (6) \end{aligned}$$

#### 3.3 快速量子仿真算法

据式(5), 输入状态向量中集合  $\bar{C}$  对应的量子比特基态的概率幅无须更新. 由式(2)、(3)可得  $|C| =$

$2^n/2^{\hat{o}+\hat{z}} = 2^{n-\hat{o}-\hat{z}} = 2^{k+m}$ ,  $|CG| = |C|/2^m = 2^k$ . 令  $\text{baseline} = \min CG$ , 由定义 3 可知  $CG = S(n, k, \text{baseline})$ . 根据式(5)、(6), 可将集合  $C$  对应的量子比特基态的概率幅分成  $2^k$  组, 对每组中的  $2^m$  个概率幅值与酉矩阵  $U$  作乘法运算, 进行更新状态向量. 结合算法 2, 基于  $2^n$  维状态向量的快速更新算法 FQSA 描述如下:

### 算法 3 基于状态向量的快速量子仿真算法 FQSA

输入: 门线路  $(s_{n-1}, \dots, s_1, s_0; U)$  和量子比特状态向量  $(a_0, a_1 \dots$

$a_{2^n-1})$ ;

输出: 更新后的状态  $(a_0, a_1 \dots a_{2^n-1})$ .

```
{
    baseline = 0; SL.SetEmpty(); k = 0; g = 0;
    for (j = 0; j < n; ++j) {
        if (Line = sj) { SL.Insert(j); ++k; }
        else if (One = sj) baseline = baseline + 2j;
        else if (Gate = sj) {
            if (m > 0) return ERROR;
            g = j; ++j; ++m;
            while (j < n && Gate = sj) ++j; ++m;
            if (j = n) break;
        } else if (sj! = Zero) return ERROR;
    }
    if (0 = m) return ERROR;
    (bn-1 bn-2 ... b0)2 = 0;
    for (j = 0; j < k; ++j) fj = j;
    fk = k;
    while (1) {
        i = baseline + (bn-1 bn-2 ... b0)2;
        ai = (ai, ai+2g, ..., ai+(2m-1)2g)';
        ai = Uai; j = f0; f0 = 0;
        if (j < k) { fj = fj+1; fj+1 = j+1; bj = 1 - bj; }
        else break;
    }
}
```

FQSA 中初始化部分的时间开销为  $O(n)$ , 由算法 2 可知, while 循环体执行  $2^k$  次, 每次循环中语句  $a_i = Ua_i$  进行  $2^m$  阶方阵和  $2^m$  维向量的乘法运算, 因此, 该语句总时间开销为  $O(2^{2m+k})$ . 令  $c$  为控制位总数, 则  $c = n - m - k$ . 由于通常量子门所占量子连线数很少, 一般为  $m$  为 1 或 2, 因此, 语句  $a_i = Ua_i$  的时间开销为  $O(2^k)$ . 当  $k = 0$  时, 门线路中无  $L$  状态线, 此时, FQSA 具

表 1 仿真 QFT 实验结果(单位: s)

	14	15	16	17	18	19	20	21	...	25	26
分治算法	3.10281	7.08782	16.1233	34.022	89.1036	197.108	435.387	918.56	...	—	—
FQSA	0.02189	0.0497	0.11373	0.27973	3.3145	7.5094	16.0268	32.9016	...	795.344	1554.23

## 5 结论

本文提出了一种通用量子线路的快速仿真算法 FQSA. 引入量子门线路的符号化表示方法替代传统仿

有最好情况时间复杂度  $O(n)$ . 当  $k = n - 1$  时, 线路中无控制位, 此时算法具有最坏时间复杂度  $O(2^n)$ . 就一般的线路而言, 在使用状态向量表示的情况下, 量子线路状态向量大小为  $2^n$ , 量子线路仿真算法复杂度至少为  $O(2^n)$ , FQSA 时间复杂度已达到此下界. 在空间上,  $2^n$  维输入向量带来的存储开销为  $O(2^n)$ .

相比而言, 分治算法<sup>[8]</sup>中包含 ApplyDown 和 ApplyUp 两种操作. 在单量子门或者二量子门情形下, ApplyDown 操作的最坏时间为  $O(2^n)$ , 然而 ApplyUp 的最坏时间为  $O(n2^n)$ . 在 QuidDD 仿真算法<sup>[5]</sup>中, 若酉算子用  $t$  个节点表示, 作用于大小为  $2^n$  的状态向量, 则运行时间的上界为  $O(t^2 2^{2n})$ . Samoladas<sup>[7]</sup>使用 MTBDD 表示状态向量, 其最坏时间复杂度为  $O(|F| 2^{2n})$ , 其中  $|F|$  是布尔投影算子的大小. 在通用量子线路仿真过程中, 线路和输入向量均无特定结构, 将酉算子和状态向量表示成 BDD 图并不能得到有效的压缩, 此时应用 FQSA 的仿真效率将得到很大提高.

## 4 仿真实验

实验仿真量子傅立叶变换(QFT)<sup>[11]</sup>, 其线路如图 2 所示.  $n$  量子 QFT 线路的长度(即包含量子门的个数)为  $n(n+1)/2$ . Hadamard 门的酉算子为  $U_H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ ,  $R_k$  的酉算子为  $U_{R_k} = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{bmatrix}$ . 表 1 给出了在 Pentium 1.87GHz CPU, 768MB 内存的 Windows XP SP3 环境下算法 FQSA 与分治算法的对比实验结果, 其中第一行为量子比特数. 由表 1 可知, FQSA 的运行时间符合给出的分析. 它相比分治算法极大的降低了运行时间. 在 1000s 时间内的同等运行环境中, FQSA 将对 QFT 的可仿真量子比特数从分治算法的 21 提高到 25.

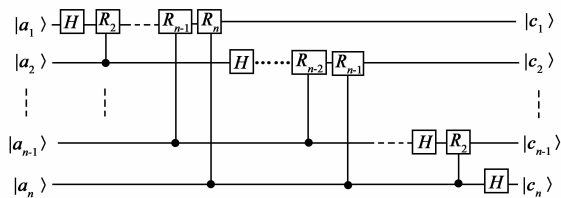


图2  $n$ 量子比特QFT电路逻辑图

真中的高阶酉矩阵表示, 给出了  $n$  元  $k$  定位二进制数生成问题及其算法, 并使用此算法对  $2^n$  维输入量子比特状态向量分组更新, 从而快速完成仿真计算. 对于通用量子线路仿真问题, 我们论证了算法 FQSA 的效率更

高,并具有最优的时间复杂度  $O(2^n)$ . 仿真更多数量量子比特通用线路的瓶颈将是  $2^n$  状态向量耗费的指数级存储开销,研制更高效的通用仿真算法并突破输入状态向量存储的瓶颈将是我們下一步的研究工作.

## 参考文献

- [1] 李志强,陈汉武,等. 四量子可逆逻辑电路快速综合算法[J]. 电子学报,2008,36(11):2081–2089.  
LI Zhi-qiang, CHEN Han-wu, et al. Fast algorithms for 4-qubit reversible logic circuits synthesis[J]. Acta Electronica Sinica, 2008, 36(11): 2081–2089. (in Chinese)
- [2] 宋辉,陆洪毅,等. 银河 TS-1 向量处理及其在量子计算中的应用[J]. 电子学报,2002,30(12):1828–1831.  
SONG Hui, LU Hong-yi, et al. The vector processing of YH TS-1 and its application on quantum computing[J]. Acta Electronica Sinica, 2002, 30(12): 1828–1831. (in Chinese)
- [3] BROWNE D E. Efficient classical simulation of the quantum fourier transform[J]. New Journal of Physics, 2007, 9(147): 1–7.
- [4] MILLER D M, et al. QMDD: A decision diagram structure for reversible and quantum circuits[A]. IEEE International Symposium on Multiple-Valued Logic (ISMVL) [C]. Washington, DC: IEEE Computer Society, 2006. 30–30.
- [5] VIAMONTES G F. Efficient Quantum Circuit Simulation[D]. America: University of Michigan, 2007.
- [6] VIAMONTES G F, et al. Improving gate-level simulation of quantum circuits[J]. Quantum Information Processing, 2003, 2(5): 347–380.
- [7] SAMOLADAS V. Improved BDD algorithms for the simulation of quantum circuits[A]. Proceedings of the 16th Annual European Symposium on Algorithms (ESA) [C]. Heidelberg: Springer-Verlag, 2008. 720–731.
- [8] 薛希玲,陈汉武,等. 量子线路仿真的分治算法[J]. 电子学报,2010,38(2):439–442.

XUE Xi-ling, CHEN Han-wu, et al. Divide and conquer algorithms for quantum circuit simulation[J]. Acta Electronica Sinica, 2010, 38(2): 439–442. (in Chinese)

- [9] D E KNUTH. The Art of Computer Programming, volume 4, fascicle 2, Generating All Tuples and Permutations[M]. New York: Addison Wesley. 2005. 9–12.
- [10] BITNER J R, et al. Efficient generation of the binary reflected gray code and its applications[J]. Communications of the ACM, 1976, 19(9): 517–521.
- [11] NIELSEN M A, CHUANG I L. Quantum Computation and Quantum Information[M]. Beijing: Higher Education Press, 2003. 217–220.

## 作者简介



陈 雄 男,1986 年 10 月出生于湖北汉川. 东南大学计算机科学与工程学院硕士研究生. 主要研究方向为量子计算、量子算法与仿真.  
E-mail: chenxiong0115@163.com



陈汉武 男,1955 年 11 月出生于江苏南京. 现为东南大学教授、博士生导师. 主要研究领域为经典信息论、量子信息与量子计算、数理解析.  
E-mail: hw\_chen@seu.edu.cn

刘志昊 男,1982 年 10 月出生于湖南邵阳. 现为东南大学博士生. 主要从事量子信息和量子通信方面的研究.

李志强 男,1974 年生于江苏姜堰. 博士研究生,副教授. 主要研究方向为量子计算、可逆电路综合与测试.