科学通报 2010 年 第 55 卷 第 4-5 期：322 ~ 327
论 文 www.scichina.com csb.scichina.com
《中国科学》杂志社
SCIENCE CHINA PRESS

Shor 整数分解量子算法的加速实现

付向群, 鲍皖苏*, 周淳

解放军信息工程大学电子技术学院, 郑州 450004
* 联系人, E-mail: 2004bws@sina.com

2009-08-24 收稿, 2009-11-28 接受
国家自然科学基金资助项目(批准号: 10501053)

摘要 基于半经典量子 Fourier 变换的实现方法, 提出了整数 k 的 3 元二进制表示生成向量和生成函数概念, 构造了生成函数的真值表, 证明了由其逐比特生成的整数 k 的 3 元二进制表示向量是整数 k 的一种 NAF 表示, 且表示中非 0 元个数的最大值为 $\left\lfloor\left(\lceil\log k\rceil+1\right)/2\right\rfloor$, 并基于此重新设计了 Shor 算法的量子实现线路. 与 Parker 的 Shor 算法量子实现线路相比, 计算资源大体相同(所需的基本量子门数量均为 $O\left(\lceil\log N\rceil^3\right)$, 所需的量子比特数量前者较后者多 2 量子比特), 但实现速度提高了 2 倍.

关键词
Shor 量子算法
半经典量子 Fourier 变换
量子比特
基本量子门
NAF 法

迄今为止, 经典公钥密码体制一般都是建立在难解的数学问题基础之上, 目前国际上广泛认可和使用的三类数学难题是整数分解问题、有限域上离散对数问题和椭圆曲线上离散对数问题. 在经典计算机上, 求解这三个问题的最好算法的计算复杂性要么是亚指数时间的, 要么是指数时间的. 比如: 目前解决整数分解问题的最优经典算法是数域筛法, 其计算复杂性是亚指数时间的, 即

$$O\left(\mathrm{e}^{(1.923+o(1))(\ln N)^{1/3}(\ln\ln N)^{2/3}}\right).$$

1994 年, Shor[1]利用量子计算机强大的并行计算能力, 提出了在多项式时间内求解整数阶的量子计算算法, 并将整数分解问题和离散对数问题归约为求解整数阶问题. Shor 算法的提出对经典公钥密码体制的安全性产生了重要的影响, 促进了量子计算机的研究. 自此以后, 量子算法的优化和实现方法研究引起了国内外学者的广泛关注, 取得了许多重要的研究成果[2~6].

1996 年, Vedral 等人[7]设计了一个量子线路, 该线路需要 $7n+1$ 个量子比特和 $O\left(n^3\right)$ 基本量子门就可以实现模幂运算(n 是所分解整数的比特长), 如果利用 Toffoli 门代替用于存储运算过程中产生的中间态的 n 比特量子寄存器, 可将所需量子比特数降为 $4n+3$, 同年, Beckman 等人[8]对此作了进一步分析, 如果所需的 Toffoli 门数量不受限制, 那么实现模幂运算只需要 $4n+1$ 个量子比特. 1998 年, Zalka[9]给出一个需要 $3n+O\left(\log n\right)$ 个量子比特实现整数分解的算法. 2000 年, Parker 等人[10]基于 Griffiths 等人[11]提出的半经典量子 Fourier 变换, 设计了一个量子线路, 该线路只需要 $n+1$ 个量子比特就可以完成整数分解. 与此同时, 国内学者在此研究领域也取得了卓有成效的研究成果, 2004 年, Long[12]给出了利用经典并行加速量子计算算法的思想, 并于 2007 年基于对偶量子计算机[13]实现了 Shor 整数分解量子算法[14].

到目前为止, 对 Shor 算法实现方法的优化都是基于算法实现所需的量子比特数和基本量子门数量,

英文版见: Fu X Q, Bao W S, Zhou C. Speeding up implementation for Shor's factorization quantum. Chinese Sci Bull, 2010, 55, doi: 10.1007/s11434-010-0039-0

即从算法实现所需的资源考虑, 还没有从提高算法的实现速度方面去研究 Shor 算法的优化. 如何在量子计算模式下提高 Shor 算法的实现速度, 就其应用而言具有重要的意义.

本文针对半经典量子 Fourier 变换输入逐比特的特点[11], 提出了整数 k 的 3 元二进制表示生成向量和生成函数概念, 构造了生成函数的真值表, 证明了其逐比特生成的整数 k 的 3 元二进制表示是整数 k 的一种 NAF 表示, 且该表示中非 0 元个数的最大值为 $\left\lceil\left(\lceil\log k\rceil+1\right)/2\right\rceil$, 并基于此重新设计了 Shor 算法的量子实现线路. 与文献[10]中Shor算法量子实现方法相比, 所需的计算资源大体相同(所需的基本量子门数量均为 $O\left(\lceil\log N\rceil^3\right)$, 所需的量子比特数量前者较后者多 2 量子比特), 但实现速度提高 2 倍.

1 基础知识

定义 1 量子 Fourier 变换[2,3] 如果在一组标准正交基 $|0\rangle,|1\rangle,\cdots,|N-1\rangle$ 上的一个线性算子在基态上的作用 U_F 为

$$U_F:|j\rangle\mapsto\frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}\omega_N^{jk}|k\rangle,$$

那么对任意量子态的作用可以表示为

$$U_F:\sum_{j=0}^{N-1}x_j|j\rangle\mapsto\frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}\sum_{j=0}^{N-1}x_j\omega_N^{jk}|k\rangle,$$

称 U_F 为量子 Fourier 变换, 其中"$\mapsto$"表示为该变换是可逆变换.

在文献[10]中, Parker 等人首次将半经典量子 Fourier 变换应用于 Shor 算法实现, 其量子线路如图 1 所示, 其中, H 为 Hadamard 门, $R_j'=\begin{pmatrix}1&0\\0&\phi_j\end{pmatrix}$, $\phi_j=\mathrm{e}^{-2\pi\mathrm{i}\sum_{k=2}^{j}m_{j-k}/2^k}$, U_a 由一个控制比特控制, 其变换为 $cU_a|r\rangle|x\rangle=|r\rangle|a^r x \bmod N\rangle$, 最终观测得到的结果 $c=\sum_{i=0}^{L}2^{L-i}m_i$, $L=2\lceil\log N\rceil+1$, $\lceil\log N\rceil$ 是比 $\log N$ 大的最小整数.

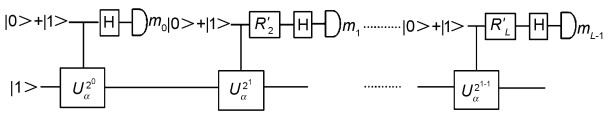

图 1 基于半经典量子 Fourier 变换的 Shor 算法实现线路

由于半经典量子Fourier变换是逐比特输入的[11], 因此, 图 1 中用于控制 U_a 变换的存储控制比特的寄存器可以循环使用, 使得 Shor 算法实现只需 $\lceil\log N\rceil+1$ 个量子比特[10], 并且需要 $O\left(\lceil\log N\rceil^3\right)$ 基本量子门.

2 逐比特输入的整数 k 的 NAF 表示法求解方法

在 Shor 整数分解量子算法中需要运行两次量子 Fourier 变换和一次模幂运算, 文献[9]中指出运行一次量子 Fourier 变换需要 $O\left(n^2\right)$ 基本量子门, 运行一次模幂运算需要 $O\left(n^3\right)$ 基本量子门, 因此模幂运算是 Shor 算法实现中最耗时的.

在经典运算中通常通过研究整数 k 的表示法来提高模幂运算 $\alpha^k \bmod N$ 的运算速度, 如整数 k 的 NAF 法[15]. NAF 表示法之所以能提高模幂运算的实现速度, 关键是整数 k 的 NAF 表示法中的非 0 元的个数比二进制表示法中少. 但是, 在现有的求整数 k 的 NAF 表示法的算法[15]中, k 的所有比特信息是一次输入的, 而在半经典量子 Fourier 变换中输入是逐比特的, 并且对输入前的比特信息不做存储, 否则会增加所需的量子比特数, 因此针对这种逐比特信息获取的整数 k, 利用文献[15]的方法求不出整数 k 的 NAF 表示, 那么如何给出它的最终 NAF 表示?

定义 2 设布尔函数 $f(x,y,z)=(u_1,u_2)$, 整数 k 的二进制表示形式为 $\left(a_0,a_1,\cdots,a_{[\log k]}\right)_2$, 即 $k=\sum_{i=0}^{[\log k]}a_i2^i$, 令 $c_{-1}=0$, $a_{[\log k]+1}=a_{[\log k]+2}=0$, 记

$$\begin{cases}f(c_{i-1},a_i,a_{i+1})=(b_{1i},b_{2i})\\h(c_{i-1},a_i,a_{i+1})=c_i\end{cases},\tag{1}$$

再令

$$b_i=\begin{cases}-b_{2i}, & b_{1i}=0,\\ b_{2i}, & b_{1i}=1,\end{cases}\tag{2}$$

其中 $i=0,1,\cdots,[\log k]+1$, 则称 $\left(b_0,b_1,\cdots,b_{[\log k]+1}\right)$ 是整数 k 的 3 元二进制表示生成向量, 其对应的整数为

$k'=\sum_{i=0}^{[\log k]+1} b_i 2^i$，$f(x,y,z)$和$h(x,y,z)$称为生成函数.

由定义2可知, 生成函数$f(x,y,z)$和$h(x,y,z)$的结构决定了整数k的 3 元二进制表示生成向量的最终形式.

定理 1 设整数k的二进制表示形式为$(a_0,a_1,\cdots,a_{[\log k]})_2$，$k=\sum_{i=0}^{[\log k]} a_i 2^i$，$k$的3元二进制表示生成向量为$(b_0,b_1,\cdots,b_{[\log k]+1})$，$k'=\sum_{i=0}^{[\log k]+1} b_i 2^i$，生成函数$f(x,y,z)=(u_1,u_2)$和$h(x,y,z)=v$的真值表如表1, 表2所示, 则$(b_0,b_1,\cdots,b_{[\log k]+1})$是$k$的一种 NAF 表示.

证明: 由$f(x,y,z)$和$h(x,y,z)$的真值表可知, 整数k的 3 元二进制表示生成向量$(b_0,b_1,\cdots,b_{[\log k]+1})$中 b_i=0 或 ± 1, $i=0,1,\cdots,[\log k]+1$, 因此要证明$(b_0,b_1,\cdots,b_{[\log k]+1})$是$k$的一种 NAF 表示, 只需证明$k=k'$. 采用数学归纳方法证明.

当整数k是 1 比特时, 显然有$k=k'$.

设整数k是$j+1(j\geqslant 1)$比特时, 结论成立, 即$k=k'$.

如果整数k是$j+2$比特的, 其二进制表示形式为$(a_0,a_1,\cdots,a_j,a_{j+1})_2$. 当$a_{j+1}=0$时, 由假设可知, $k'=k$; 当$a_{j+1}=1$时, 记k''的二进制表示形式为$(a'_0,a'_1,\cdots,a'_j)_2$, k''的 3 元二进制表示生成向量为$(b'_0,b'_1,\cdots,b'_j,b'_{j+1})$, 其中$a'_i=a_i$, $i=0,1,\cdots,j$, 那么由定义 2 和生成函数的真值表可知, $b'_{j+1}=f(c_j,a'_{j+1},a'_{j+2})$, 其中$a'_{j+1}=a'_{j+2}=0$, 故$b'_{j+1}$的取值只能为0或1.

不妨设整数k的 3 元二进制表示生成向量为$(b_0,b_1,\cdots,b_j,b_{j+1},b_{j+2})$, 易知其前$j$比特与整数$k''$的 3 元二进制表示生成向量$(b'_0,b'_1,\cdots,b'_j,b'_{j+1})$的前$j$比特相同, 即$b_i=b'_i$, $i=0,1,\cdots,j-1$.

如果$b'_{j+1}=0$, 则$c_j=h(c_{j-1},a_j,0)=0$, 从而(c_{j-1},a_j)的取值为$(0,0)$和$(0,1)$或$(1,0)$.

当$(c_{j-1},a_j)=(0,1)$时, 可得$(b'_{1j},b'_{2j})=f(c_{j-1},a_j,a'_{j+1})=(1,1)$, 即$b'_j=1$, 此时$(b_{1,j},b_{2,j})=f(c_{j-1},a_j,a_{j+1})=(0,1)$, $c'_j=h(c_{j-1},a_j,a_{j+1})=1$, 因此$(b_{1,j+1},b_{2,j+1})=f(c'_j,a_{j+1},a_{j+2})=(0,0)$, 从而$c'_{j+1}=h(c'_j,a_{j+1},a_{j+2})=1$, $(b_{1,j+2},b_{2,j+2}=f(c'_{j+1},a_{j+2},a_{j+3})=(1,1)$, 其中$a_{j+2}=0$, $a_{j+3}=0$.

同理, 当$(c_{j-1},a_j)=(1,0)$时, $b'_j=1$, $b_j=\bar{1}$, $b_{j+1}=0$, $b_{j+2}=1$, 其中$\bar{1}=-1$; 当$(c_{j-1},a_j)=(0,0)$时, $b'_j=0$, $b_j=0$, $b_{j+1}=1$, $b_{j+2}=0$.

又由于$k''=\sum_{t=0}^{j+1} b'_t 2^t=\sum_{t=0}^{j} a_t 2^t$, 而$\sum_{t=0}^{j+2} b_t 2^t-k''=2^{j+1}$, 即$k'=\sum_{t=0}^{j+2} b_t 2^t=\sum_{t=0}^{j+1} a_t 2^t$, 故$k'=k$.

如果$b'_{j+1}=1$, 则同理可证$k'=k$.

综上所述, $k'=k$, 即$(b_0,b_1,\cdots,b_{[\log k]+1})$就是$k$的一种 NAF 表示. 证毕.

在定理 1 的证明中, 由于

$$\begin{cases} f(c_{i-1},a_i,a_{i+1})=(b_{1i},b_{2i}), \\ f(c_i,a_{i+1},a_{i+2})=(b_{1,i+1},b_{2,i+1}). \end{cases}$$

当$(c_{i-1},a_i,a_{i+1})=(0,0,0)$, $(0,0,1)$, $(1,1,0)$, $(1,1,1)$时, $(b_{1i},b_{2i})=(0,0)$, 即$b_i=0$, 所以$b_ib_{i+1}=0$.

当$(c_{i-1},a_i,a_{i+1})=(0,1,0)$时, 可得$c_i=h(c_{i-1},a_i,a_{i+1})=0$, 又由于$(b_{1,i+1},b_{2,i+1})=f(0,0,a_{i+2})$, 根据生成函数$f$的真值表可知, $(b_{1,i+1},b_{2,i+1})=(0,0)$与$a_{i+2}$的取值无关, 即$b_{i+1}=0$, 所以$b_ib_{i+1}=0$.

当$(c_{i-1},a_i,a_{i+1})=(0,1,1),(1,0,0)$或$(1,0,1)$时, 同理

表 1 $f(x,y,z)$的真值表

x	y	z	u_1	u_2
0	0	0	0	0
0	0	1	0	0
0	1	0	1	1
0	1	1	0	1
1	0	0	1	1
1	0	1	0	1
1	1	0	0	0
1	1	1	0	0

表 2 $h(x,y,z)$的真值表

x	y	z	v
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

可得，$b_i b_{i+1}=0$.

因此，根据定理 1，所求出的整数 k 的 3 元二进制表示生成向量 $\left(b_0,b_1,\cdots,b_{[\log k]+1}\right)$ 中相邻两个分量的乘积为零，即任何两个非零分量不相邻.

以下分析这种逐比特求出的整数 k 的 NAF 表示中非 0 元个数的最大值.

定理 2 设整数 k 的二进制表示形式为 $\left(a_0,a_1,\cdots,a_{[\log k]}\right)_2$，$k=\sum_{i=0}^{[\log k]} a_i 2^i$，则由定理 1 求出整数 k 的 3 元二进制表示生成向量 $\left(b_0,b_1,\cdots,b_{[\log k]+1}\right)$ 的非 0 元个数最大值为 $\left[\left(\lceil\log k\rceil+1\right)/2\right]$.

证明：根据定理 1，求得的 $\left(b_0,b_1,\cdots,b_{[\log k]+1}\right)$ 是将 $\left(a_0,a_1,\cdots,a_{[\log k]}\right)_2$ 中连续两个或两个以上 1 的情况作如下变换

$$\underbrace{111\cdots111}_{j}\ \Rightarrow\ \bar{1}\underbrace{00\cdots001}_{j-1}\quad (j\geqslant 2).$$

如果 $\left(a_0,a_1,\cdots,a_{[\log k]}\right)_2$ 中含有 i 个 1，$i<\left[\left(\lceil\log k\rceil+1\right)/2\right]$，则 $\left(b_0,b_1,\cdots,b_{[\log k]+1}\right)$ 中含有少于 $\left[\left(\lceil\log k\rceil+1\right)/2\right]$ 个非零元素.

如果 $\left(a_0,a_1,\cdots,a_{[\log k]}\right)_2$ 中含有 $\left[\left(\lceil\log k\rceil+1\right)/2\right]$ 个 1，且不存在“$\underbrace{111\cdots111}_{j}$”$(j\geqslant 2)$ 形式，则当 $\lceil\log k\rceil$ 为奇数时，$\left(a_0,a_1,\cdots,a_{[\log k]}\right)_2=(1010\cdots01)_2$，当 $\lceil\log k\rceil$ 为偶数时，$\left(a_0,a_1,\cdots,a_{[\log k]}\right)_2=(1010\cdots10)_2$ 或 $(0101\cdots01)_2$，所以 $\left(b_0,b_1,\cdots,b_{[\log k]}\right)=(a_0,a_1,\cdots,a_{[\log k]})_2$ 且 $b_{[\log k]+1}=0$，即含有 $\left[\left(\lceil\log k\rceil+1\right)/2\right]$ 个非零元素.

如果 $\left(a_0,a_1,\cdots,a_{[\log k]}\right)_2$ 中含有 $\left[\left(\lceil\log k\rceil+1\right)/2\right]+1$ 个 1，则必定存在“111”形式，这些比特位在 $\left(b_0,b_1,\cdots,b_{[\log k]+1}\right)$ 中对应的数为 $\bar{1}001$，即 $\left(b_0,b_1,\cdots,b_{[\log k]+1}\right)$ 中含有至多 $\left[\left(\lceil\log k\rceil+1\right)/2\right]$ 个非零元素.

同理，如果 $\left(a_0,a_1,\cdots,a_{[\log k]}\right)_2$ 中含有 i 个 1，$i>\left[\left(\lceil\log k\rceil+1\right)/2\right]+1$，则 $\left(b_0,b_1,\cdots,b_{[\log k]+1}\right)$ 中含有至多 $\left[\left(\lceil\log k\rceil+1\right)/2\right]$ 个非零元素.

综上，$\left(b_0,b_1,\cdots,b_{[\log k]+1}\right)$ 中非 0 元素个数的最大值为 $\left[\left(\lceil\log k\rceil+1\right)/2\right]$. 证毕.

定理 1 中给出的生成函数 $f(x,y,z)$ 和 $h(x,y,z)$ 能否在量子计算机实现，直接关系到逐比特输入的整数 k 的 NAF 表示法求解方法能否利用量子线路实现.

根据定理 1 中 $f(x,y,z)$ 和 $h(x,y,z)$ 的真值表，可以求出其布尔函数表达式为

$$f(x,y,z)=\left(x'yz'\oplus xy'z',x'y\oplus xy'\right),$$
$$h(x,y,z)=xy'z\oplus yz\oplus xyz',$$

其中 $y'=1+y$，$z'=1+z$，$\oplus$ 是模 2 加运算. 因此 $f(x,y,z)$ 和 $h(x,y,z)$ 的量子实现线路可以按如图 2、图 3 方式设计，其中 M 是乘法运算，A 是模 2 加运算，N 是非运算，M,A 和 N 运算的具体量子线路可参见文献[7]. 从图 2, 3 中可以看出，实现 $f(x,y,z)$ 需要 3 次乘法运算、3 次非运算和 1 次模 2 加运算，$h(x,y,z)$ 需要 3 次乘法运算、2 次非运算和 1 次模 2 加运算，而实现这 3 个运算所需的基本量子门数量均为 $O(1)$，即 $f(x,y,z),h(x,y,z)$ 可由规模为 $O(1)$ 基本量子门实现.

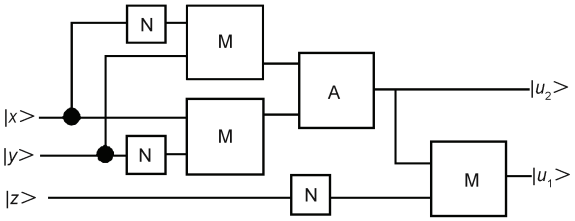

图 2 $f(x,y,z)$ 的量子实现线路图

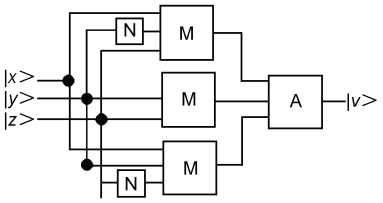

图 3 $h(x,y,z)$ 的量子实现线路图

3 新的 Shor 整数分解量子算法实现量子线路及分析

基于逐比特输入的整数 k 的 NAF 表示法求解方法，本文设计的 Shor 整数分解量子算法的量子线路如图 4 所示.

图 4 的相关说明:

(1) N 是所要分解的整数;

(2) H 是 Hadamard 门[10]; $R'_j=\begin{pmatrix}1 & 0\\ 0 & \phi'_j\end{pmatrix}$ 且 $\phi'_j=\mathrm{e}^{-2\pi i\sum_{k=2}^{j}m_{j-k}/2^k}$, $U_\alpha^{u2^i}$ 盒的上方两个输入 u_1,u_2 是控制 u 的值, 其中 $u=\begin{cases}u_2, & u_1=1,\\ -u_2, & u_1=0;\end{cases}$

(3) 假设 $U_\alpha^{u2^i}$ 黑盒的左端输入是 A, 那么输出的结果是 $A\cdot\alpha^{u2^i} \bmod N$, 其中 $\alpha^{\pm 2^i} \bmod N$ 在经典计算机上预计算;

(4) 图 4 中刚开始输入的 4 个量子寄存器中, $|1\rangle$ 是 $\lceil\log N\rceil$ bit, 其余都是 1 bit;

(5) ⊐是探测器, 最终观测到的结果为 $m=\sum_{i=0}^{L}2^{L-i}m_i$;

(6) $|s_i\rangle=\begin{cases}|0\rangle+|1\rangle, 0\leqslant i\leqslant L-1\\ |0\rangle, \quad L-1<i\leqslant L+1\end{cases}$, 图 4 中的量子态 $|0\rangle+|1\rangle$ 实际上是量子态 $\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$, 是 $|0\rangle$ 态通过 Hadamard 变换得到的, 此处简写为 $|0\rangle+|1\rangle$.

从图 4 中可以看出, 新的 Shor 整数分解量子算法实现量子线路需要 $\lceil\log N\rceil+3$ 量子比特. 因为运算一次 f, h 需要 $O(1)$ 基本量子门, 所以新的量子线路中完成 $L+1$ 次 f 和 $L+1$ 次 h 的运算需要 $O(L)$ 基本量子门. 而实现 $L+1$ 次模幂运算需要 $O\left(\lceil\log N\rceil^3\right)$ 基本量子门, 量子 Fourier 变换需要 $O\left(\lceil\log N\rceil^2\right)$ 基本量子门, 因此, 新的量子线路整体需要 $O\left(\lceil\log N\rceil^3\right)$ 基本量子门. 与文献[10]相比, 所需的基本量子门均为 $O\left(\lceil\log N\rceil^3\right)$, 但所需量子比特数前者较后者多 2 量子比特.

对于 x 的所有取值 $p_1,p_2,\cdots,p_{2^L-1}$, 由于量子计算机计算的并行性, 即 $\alpha^{p_1} \bmod N,\alpha^{p_2} \bmod N,\cdots,\alpha^{p_{2^L-1}} \bmod N$ 是同时计算出来的, 因此, $\alpha^{p_1} \bmod N$, $\alpha^{p_2} \bmod N,\cdots,\alpha^{p_{2^L-1}} \bmod N$ 中最耗时的模幂运算的运行时间算就是整个模幂运算 $\alpha^x \bmod N$ 的运行时间.

在现有的公开文献中, Shor 算法中模幂运算 $\alpha^x \bmod N$ 的实现是按如下方式进行的.

先将 x 展开为二进制表示

$$x=x_{L-1}2^{L-1}+x_{L-2}2^{L-2}+\cdots+x_0,$$

然后计算

$$\alpha^x \bmod N=\left(\alpha^{2^{L-1}}\right)^{x_{L-1}}\left(\alpha^{2^{L-2}}\right)^{x_{L-2}}\cdots(\alpha)^{x_0} \bmod N,$$

当 $x_j=1$ 时, 就乘上 α^{2^j} (α^{2^j} 是预计算的, 在经典计算机上完成[10]), 否则, 就乘上 1, 因此, 最多用 L 步就能计算出 $\alpha^x \bmod N$.

如果整数 k 用二进制表示, 那么 $\alpha^{2^{L-1}} \bmod N$ 的运行时间就被用来刻画 Shor 整数分解算法中模幂运算的运行时间. 实现一个乘法运算需要 n 次加法运算(加法运算是两个 n 比特的数相加, $n=\lceil\log N\rceil$), 实现一次模幂运算需要 L 次乘法运算[9], 因此用于实现 Shor 算法模幂运算需要 nL 次两个 n 比特数的加法运算, 需要预计算 $\alpha \bmod N,\alpha^2 \bmod N,\cdots,\alpha^{2^{L-1}} \bmod N$.

如果整数 k 用 3 元二进制表示, 那么根据定理 2, 当逐比特输入求出的幂指数 x 的 3 元二进制表示中含非 0 元个数为 $\lceil(L+1)/2\rceil$ (记 x' 的 3 元二进制表示向

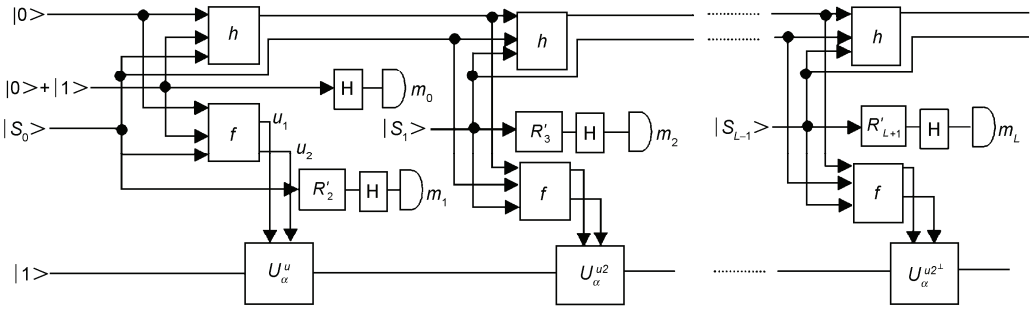

图 4 基于逐比特输入的整数 k 的 NAF 表示法的 Shor 整数分解量子算法实现线路

量中非 0 元素的个数为$\left[(L+1)/2\right]$)时，$\alpha^{x'} \bmod N$的运行时间用来刻画 Shor 整数分解算法中模幂运算的运行时间，因此，用于实现 Shor 算法模幂运算需要$n\left[(L+1)/2\right]$次的两个 n 比特数的加法运算，需要预计算$\alpha^{\pm 1} \bmod N, \alpha^{\pm 2} \bmod N, \cdots, \alpha^{\pm 2^{L+1}} \bmod N$.

对于函数 f, h 而言，均需要 12 次加法运算(两个 3 比特的数相加)，而非运算的运行时间与加法运算相比，可以忽略不计，因此新量子线路中运行 $L+1$ 次 f 和 $L+1$ 次 h 需要 $24L$ 次的两个 3 比特数的加法运算，与模幂运算相比，其时间可以忽略不计. 考虑到在量子线路中 H 门和 R_l' 门与模幂运算是并行的，它们是 1 比特门，实现它们比模幂运算省时，所以新的量子线路整体所需时间可以用模幂运算的运行时间来刻画.

综上，与文献[10]相比，新的量子线路的运行速度提高约 2 倍，预计算增加约 1 倍.

4 结论

本文针对半经典量子 Fourier 变换的实现方法，提出了整数 k 的 3 元二进制表示生成向量和生成函数概念，并构造了生成函数的真值表，通过这种生成函数求出的 3 元二进制表示生成向量是一种新的 NAF 表示，根据该生成向量重新设计了 Shor 算法的量子实现线路，与文献[10]相比，其所消耗的资源大体相同，实现速度提高了 2 倍. 在经典公钥密码体制中有许多提高模幂运算实现速度的方法，如何将这些方法与 Shor 量子算法相融合，有待进一步研究.

参考文献

1 Shor P W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J Comput, 1997, 26: 1484—1509
2 Childs A M, Dam W V. Quantum algorithms for algebraic problems. Arxiv: quant-ph/08120380v1, 2008
3 方细明，朱熙文，冯芒，等. 核磁共振实现量子分立 Fourier 变换. 科学通报, 2000, 45: 140—145
4 何雨果，孙吉贵. 基于 Haar 小波的多尺度分析量子电路. 科学通报, 2005, 50: 2314—2316
5 Mehring M, Müller K, Averbukh I S, et al. NMR experiment factors fumbers with Gauss sums. Phys Rev Lett, 2007, 98: 120502
6 Mahesh T S, Rajendran N, Peng X H, et al. Factoring numbers with the Gauss sum technique: NMR implementations. Phys Rev A, 2007, 75: 062303
7 Vedral V, Barenco A, Ekert A. Quantum networks for elementary arithemetic operations. Phys Rev A, 1996, 54: 147—153
8 Beckman D, Chari A N, Devabhaktuni S, et al. Efficient networks for quantum factoring. Phys Rev A, 1996, 54: 1034—1063
9 Zalka C. Fast versions of Shor's quantum factoring algorithm. Arxiv: quant-ph/9806084v1, 1998
10 Parker S, Plenio M B. Efficient factorization with a single pure qubit and logN mixed qubits. Phys Rev Lett, 2000, 85: 3048—3052
11 Griffiths R B, Niu C S. Semiclassical fourier transform for quantum computation. Phys Rev Lett, 1996, 76: 3228—3232
12 Long G L, Xiao L. Parallel quantum computing in a single ensemble quantum computer. Phys Rev A, 2004, 69: 052303
13 Long G L. General quantum interference principle and duality computer. Commun Theor Phys, 2006, 45: 825—844
14 Wang W Y, Shang B, Wang C, et al. Prime factorization in the duality computer. Commun Theor Phys, 2007, 47: 471—473
15 Solinas J A. Efficient arithmetic on Koblitz curves. Designs Codes Cryptography, 2000, 19: 195—249