# SM3 和 RIPEMD-160 杂凑函数的高效量子线路实现

邹剑[1,2]，郭楠生[1,2]，李俊康[1,2]
（1. 福州大学计算机与大数据学院，福建 福州 350108；2. 福州大学网络系统信息安全福建省高校重点实验室，福建 福州 350108）

**摘　要：**量子计算技术的发展对现有密码算法的安全性构成潜在威胁，亟需高效的量子线路实现方案以支撑其安全性评估。为设计 SM3 和 RIPEMD-160 杂凑函数的低 T 深度高效量子线路，采用了如下方法：对于 SM3，通过应用低 T 深度量子加法器以及量子组件的重新排列来优化整体线路；对于 RIPEMD-160，设计其布尔函数的低 T 深度量子实现，优化其压缩函数的分层并行排列，并首次构建其量子线路。结果显示，所提出的 SM3 量子线路在 T 深度和 T 深度–量子比特数乘积成本（T-DW-cost）值上显著优于现有方案，T 深度达到 4 528；首次实现的 RIPEMD-160 量子线路 T 深度为 4 454，为 SM3 和 RIPEMD-160 在量子环境下的安全性评估提供了高效的线路实现方案。

**关键词：**量子线路；SM3；RIPEMD-160；T 深度
**中图分类号：**TP393.0
**文献标志码：**A
**DOI：**10.11959/j.issn.1000–436x.2025246

# Efficient quantum circuit implementations of SM3 and RIPEMD-160 Hash functions

ZOU Jian[1,2], GUO Nansheng[1,2], LI Junkang[1,2]
1. College of Computer and Data Science, Fuzhou University, Fuzhou 350108, China
2. Fujian Provincial Key Laboratory of Network Systems and Information Security, Fuzhou University, Fuzhou 350108, China

**Abstract:** The development of quantum computing technology poses potential threats to the security of existing cryptographic algorithms, making it urgent to develop efficient quantum circuit implementation schemes to support security assessments. To design low-T-depth and efficient quantum circuits for the SM3 and RIPEMD-160 Hash functions, the following methods were employed. For SM3, the overall circuit was optimized by applying low-T-depth quantum adders and reorganizing the quantum components. For RIPEMD-160, a low-T-depth quantum implementation of its Boolean functions was designed, the hierarchical parallel arrangement of its compression function was optimized, and its quantum circuit was constructed for the first time. The results show that the proposed SM3 quantum circuit significantly outperforms existing schemes in terms of T-depth and T-DW-cost, with a T-depth of 4 528. The first-time implemented RIPEMD-160 quantum circuit achieves a T-depth of 4 454. These provide efficient circuit implementation schemes for the security assessment of SM3 and RIPEMD-160 in quantum environments.

**Keywords:** quantum circuit, SM3, RIPEMD-160, T-depth

## 0　引言

量子计算技术的飞速演进，使得对密码算法安全性的系统性评估成为当前信息安全领域的重要议题。这一研究需求的迫切性源于学界的基本共识：具备强大算力的量子计算机能够高效求解现有主流密码算法所基于的数学难题，从而对当前密码体系

收稿日期：2025–08–27；修回日期：2025–12–09
通信作者：邹剑，fzuzoujian15@163.com
基金项目：国家密码科学基金资助项目（No.2025NCSF02012）
**Foundation Item：**The National Cryptologic Science Fund of China (No.2025NCSF02012)

的安全性构成潜在威胁。具体而言，肖尔（Shor）算法[1]可在多项式时间内完成整数分解操作，这对于依托大数分解机制的 RSA （Rivest-Shamir-Adleman）加密算法[2]、椭圆曲线密码算法（ECC, elliptic curve cryptography）[3]而言，无疑构成了安全隐患。格罗弗（Grover）算法[4]作为一种量子搜索方法，能够让传统的密钥穷举搜索效率实现平方根级别的提升，对高级加密标准（AES, advanced encryption standard）[5]、安全哈希算法 2（SHA-2, secure Hash algorithm 2）[6]以及安全哈希算法 3（SHA-3, secure Hash algorithm 3）[7]等算法带来威胁。另外，西蒙（Simon）算法[8]在解决特定函数的周期查找问题上展现出高效性，这也给对称密钥系统的安全带来了不容忽视的潜在挑战。在此背景下，对现有密码算法进行全面的安全性分析，以明确其在量子计算环境下的安全强度，成为一项关键任务。

在量子线路优化这一研究领域，相关研究工作主要围绕着 2 个核心方向展开：一是尽可能减少量子比特的使用数量，二是提升线路的容错性能。Grassl 等[9]针对 AES 的量子线路展开系统性探究，创新性地引入锯齿形（zig-zag）结构，构建出仅需 40 个量子比特的 AES S 盒线路。Almazrooie 等[10]通过优化乘法逆元线路，将 AES-128 所需的量子比特数降至 976 个。2020 年，Langenberg 等[11]采用塔域分解技术，进一步将 AES 的 S 盒线路所需量子比特数缩减至 32 个，同时完成了 AES-128 量子线路的整体设计，整个设计共消耗 864 个量子比特。同年，Zou 等[12]在 ASIACRYPT2020 上提出了宽度为 22 个量子比特的 AES S 盒线路，通过对 zig-zag 结构进行改进，仅用 512 个量子比特便构建出 AES-128 的新型量子线路。Wang 等 [13] 提出一种线性密钥扩展方法，构建出了消耗 400 个量子比特 AES-128 量子线路。Li 等[14]通过自动化工具构建出了仅需 20 个量子比特的 AES S 盒量子线路和 16 个量子比特的 AES S 盒原地实现量子线路，提出了宽度为 270 个量子比特的 AES-128 量子线路。Huang 等[15]提出了 2 种构造向量布尔函数最小宽度实现的方法，并以此实现了 9 量子比特的 AES S 盒量子线路，以及 5 量子比特的 SHA3 χ 函数量子线路。

量子线路深度是衡量量子线路性能的另一重要指标，它指的是线路中量子门的最大层数，同一层内可以包含多个并行运作的量子门。而在量子门中，T 门由于其独特的物理特性，实现成本远高于其他量子门，因此 T 深度，即量子线路中从输入到输出最长的 T 门路径长度，成为评估量子线路运行效率时不可或缺的关键指标。Li 等[16]针对 S 盒这一 AES 中的核心模块，提出了一种 T 深度为 4 的 S 盒线路，有效地将 AES-128 量子线路的整体 T 深度降低到了 80。Huang 等[17]在 2022 年亚洲密码会议（ASIACRYPT2022）上提出了 T 深度为 3 和 4 的 AES S 盒量子线路及其逆线路。他们提出了一种通过线性变换将 S 盒线路转换为其逆量子线路的方法，并以此构建了 T 深度为 60、量子比特数为 374 的 AES-128 量子线路。Lin 等[18]通过对经典线路进行重新排列，提出了 AES S 盒新的量子线路实现方案，该方案在线路宽度和深度代价之间进行了权衡。在同年的 ASIACRYPT2023 上，Liu 等[19]在不增加 T 深度的前提下，将 AES S 盒量子线路所需的辅助比特数减少到 83 个，同时提出了基于共享思想的组合管道架构，仅使用 98 个辅助量子比特就完成了 AES S 盒及其逆变换的量子线路的构建，显著降低了辅助量子比特的消耗。Shi 等[20]提出了一种新型优化深度的贪心算法，为 AES 的列混合变换（MixColumns, mixcolumns transformation）操作找到了受控非门（CNOT, controlled-NOT gate）深度为 10 的线路，同时提出了一种名为压缩流水线结构的新架构来合成 AES 量子线路，构建出 T 深度为 33 的 AES-128 量子线路。Huang 等[15]设计了 T 深度 3 的 AES S 盒的克利福德 + T 门基（Clifford+T, Clifford+T gate basis）紧凑线路，该线路在 T 门数、总深度及 Clifford 门数上均得到了显著优化。

SM3 密码 [21]是由我国科研人员于 2010 年自主研发的杂凑算法，凭借其高安全性和可靠性，被广泛应用于数字签名、消息认证等诸多关键信息安全场景。在 SM3 的量子线路实现方面，2021 年，Song 等[22]以优化线路宽度为核心目标，最终提出了消耗 2 721 量子比特的 SM3 量子线路。2022 年，Zou 等[23]通过对 SM3 算法的内部逻辑与运算特性进行研究，分别针对量子比特数量和 T 深度，提出了 2 种优化实现方案。在低宽度量子线路设计方面，Zou 等[23]构建了一套仅需 33 个可复用辅助量子位的 SM3 量子线路，而在低 T 深度线路实现中，通过对非线性组件的量子线路优化设计，同时对 SM3 整体量子线路进行设计，使 T 深度达到 25 344。

2025年，Yang等[24]在前人研究成果的基础上持续深耕，进一步优化了非线性组件的低T深度量子实现方案，提出了2种低T深度量子线路方案，T深度分别为7 040和7 136，同时也提出了仅需779个量子比特的低宽度量子线路实现方案。本文扩展了上述工作，提出了一种新型SM3量子线路设计，实现了更低的T深度。表1中展示了SM3量子线路的64轮资源。结果表明，本文的量子线路设计在T深度和T深度–量子比特数乘积成本（T-DW-cost, T-depth and qubit count product cost）值上显著优于前人的工作。其中，T-DW-cost是量子比特数量和T深度的乘积，这个指标综合了量子比特和T深度，能够更全面地反映量子线路的资源消耗情况。

加密散列函数RIPEMD-160[25]作为密码学领域的重要杂凑函数，在杂凑函数的发展历程中占据关键地位。RIPEMD-160的输出长度为160 bit，相较于128 bit的消息摘要算法5（MD5, message-digest algorithm 5）[26]，更长的输出空间显著降低了碰撞攻击的风险。值得注意的是，目前RIPEMD-160在量子线路方面的研究尚属空白，鉴于其在诸多领域的重要应用价值，对其进行量子化探索具有重要意义。为此，本文针对这一现状展开研究，提出了低T深度的RIPEMD-160量子线路，以期为该杂凑函数在量子计算时代的应用与发展提供新的思路。本文所实现的SM3和RIPEMD-160量子线路资源如表1所示。

## 1 预备知识

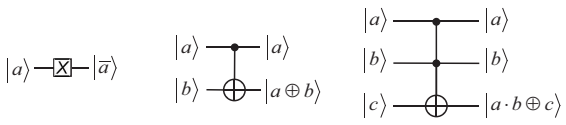
### 1.1 量子线路与量子门

电子计算机以线路和逻辑门为基本构建模块。与之类似，量子计算机的核心组件是量子线路，其由量子比特与量子门构成。本文介绍几种常用的量子门类型，如图1所示。

1) 量子非门（X门）是单量子比特门，仅有一个输入。量子非门可以实现量子线路中的取反（NOT）操作，完成输入量子比特状态的翻转，将$\alpha|0\rangle+\beta|1\rangle$转化成$\beta|0\rangle+\alpha|1\rangle$。量子非门对单个量子态的作用可描述如下：$|a\rangle\rightarrow|a\oplus1\rangle$。

2)受控非门（CNOT门）有2个输入比特，用于在量子线路中实现经典计算机的逻辑线路中的与非门的功能。当控制量子比特的状态为$|0\rangle$时，目标量子比特状态不变；当控制量子比特的状态为$|1\rangle$|，目标量子比特状态取反。受控非门完成以下映射：$|a,b\rangle\rightarrow|a,a\oplus b\rangle$。

3)Toffoli门，也被称为控–控–非门（CCNOT门），是一种三量子位逻辑门。只有当2个控制量子比特为$|1\rangle$|时，它才会对第三个量子比特执行NOT操作。量子算法中经常使用Toffoli门来实现乘法运算。Toffoli门完成的映射可描述如下：$|a,b,c\rangle\rightarrow|a,b,c\oplus ab\rangle$。




(a) X门 (b) CNOT门 (c) Toffoli门

图1 基础量子门

Amy等[27]实现了T深度为3的Toffoli门分解方案，具体实现如图2所示。

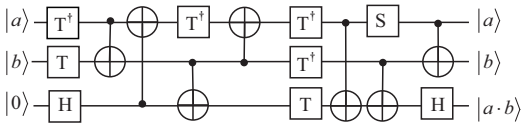



图2 T深度为3的Toffoli门

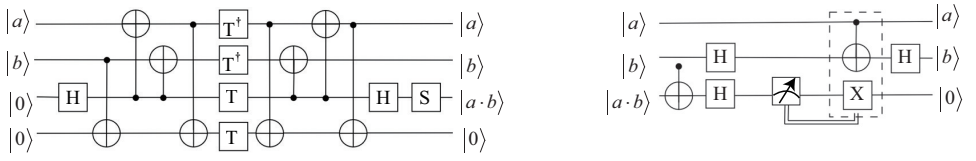
由于降低量子线路的T深度是本文的优化目标，因此引入文献[28]中提出的量子与（QAND, quantum AND）门来替代Toffoli门实现量子乘法。QAND门的T深度为1，QAND†门的T深度为0，如图3所示。

### 1.2 SM3

SM3大致流程可概述为对长度为$l(l<2^{64})$ bit的消息$m$，经过消息填充/扩展和迭代压缩，生成杂凑值，杂凑值长度为256 bit，其流程具体说明如下。

表1 SM3和RIPEMD-160量子线路资源

| 目标算法 | 来源 | T深度 | 量子比特数 | T门 | CNOT门 | 1qcliff门 | T-DW-cost |
|---|---|---|---|---|---|---|---|
| SM3 | 2.2节 | 4 528 | 1 369 | 965 632 | 2 310 528 | 905 984 | $6.19\times10^6$ |
| | 文献[24] | 7 040 | 1 130 | 662 528 | 1 782 656 | 1 201 216 | $7.95\times10^6$ |
| | 文献[23] | 25 344 | 970 | 1 187 072 | 1 370 432 | 420 160 | $2.45\times10^7$ |
| RIPEMD-160 | 3.2节 | 4 454 | 1 868 | 1 220 976 | 2 827 862 | 1 119 906 | $8.32\times10^6$ |




(a) QAND门　　(b) QAND†门

图3　QAND门和QAND†门

1) 消息填充

在对消息进行处理之前，需要先对消息进行填充操作，目的是使消息的长度满足特定的要求，便于后续的分组处理。假设消息$m$的长度为$l$ bit，首先将1 bit“1”添加到消息的末尾。再添加$k$个“0”，$k$是满足$l+1+k\equiv 448(\mathrm{mod}\ 512)$的最小非负整数。最后添加一个64 bit串，该比特串是$l$的二进制表示。填充后消息$m'$的长度为512的倍数。

2) 消息分组与扩展

消息$m'$需要分组，将填充后的消息$m'$按512 bit分组，每个分组用$B^{(i)}$表示$m'=B^{(0)}B^{(1)}\cdots B^{(n-1)}$，其中$n=\dfrac{l+k+65}{512}$。对于分组后的消息$B^{(i)}$，先将$B^{(i)}$分成16个字$W_0,W_1,\cdots,W_{15}$，按算法1进行扩展。

**算法1**　SM3消息扩展算法

**输入**　$W_0,W_1,\cdots,W_{15}$

**输出**　$W_0,W_1,\cdots,W_{67},W_0',W_1',\cdots,W_{63}'$

1)for $j$ = 16 to 67 do

2)　$W_j\leftarrow P_1(W_{j-16}\oplus W_{j-9}\oplus(W_{j-3}\lll 15))\oplus(W_{j-13}\lll 7)\oplus W_{j-6}$

3)end for

4)for $j$ = 0 to 63 do

5)　$W_j'\leftarrow W_j\oplus W_{j+4}$

6)end for

其中$P_1(X)=X\oplus(X\lll 15)\oplus(X\lll 23)$，该算法将生成这些扩展后的消息用于后续的迭代压缩。

3) 迭代压缩

将消息$m'$与扩展后的消息$W$与$W'$共132个字按下列方式迭代。对于分组的消息$B^{(i)}$（0<$i$<$n$−1），运用迭代公式$V^{(i+1)}=\mathrm{CF}(V^{(i)},B^{(i)})$进行迭代，其中CF是压缩函数，$V^{(0)}$为256 bit初始值IV= 0x7380166f 4914b2b9 172442d7 da8a0600 a96f30bc 163138aa e38dee4d b0fb0e4e，迭代压缩的结果为$V^{(n)}$。CF压缩函数的主要流程是利用$A$、$B$、$C$、$D$、$E$、$F$、$G$、$H$8个大小为32bit的寄存器进行操作，在每一轮迭代压缩过程开始前会将$V^{(i)}$的值以大端序存储到以上8个寄存器中，之后进行迭代压缩过程。迭代压缩过程如算法2所示。

**算法2**　迭代压缩算法

**输入**　$V_0,W_0,W_1,\cdots,W_{67},W_0',W_1',\cdots,W_{63}'$

**输出**　$V_n$

1)for $j$=0 to 63 do

2)　SS1 ← $((A\lll 12)+E+(T_j\lll j))\lll 7$

3)　SS2 ← SS1⊕$(A\lll 12)$;

4)　TT1 ← $\mathrm{FF}_j(A,B,C)+D+\mathrm{SS2}+W_J'$;

5)　TT2 ← $\mathrm{GG}_j(E,F,G)+H+\mathrm{SS1}+W_J$;

6)　$D\leftarrow C$;

7)　$C\leftarrow B\lll 9$;

8)　$B\leftarrow A$;

9)　$A\leftarrow$ TT1;

10) $H\leftarrow G$;

11) $G\leftarrow F\lll 19$;

12) $F\leftarrow E$;

13) $E\leftarrow P_0(\mathrm{TT2})$;

其中$P_0(X)=X\oplus(X\lll 9)\oplus(X\lll 17)$，$\mathrm{FF}_j$与$\mathrm{GG}_j$为布尔函数，$\mathrm{Con}_j$为常量。其中所用到的布尔函数和常量$\mathrm{Con}_j$如下。

$$\mathrm{FF}_j(X,Y,Z)=\begin{cases}X\oplus Y\oplus Z, & 0\leqslant j\leqslant 15\\ (X\wedge Y)\oplus(X\wedge Z)\oplus(Y\wedge Z), & 16\leqslant j\leqslant 63\end{cases}\tag{1}$$

$$\mathrm{GG}_j(X,Y,Z)=\begin{cases}X\oplus Y\oplus Z, & 0\leqslant j\leqslant 15\\ (X\wedge Y)\oplus(\neg X\wedge Z), & 16\leqslant j\leqslant 63\end{cases}\tag{2}$$

$$\mathrm{Con}_j=\begin{cases}\text{0x79cc4519}, & 0\leqslant j\leqslant 15\\ \text{0x7a879d8a}, & 16\leqslant j\leqslant 63\end{cases}\tag{3}$$

### 1.3　RIPEMD-160

RIPEMD-160是一种杂凑函数，设计是为了应对RIPEMD算法在安全性上可能存在的隐患，提供更高的安全性和可靠性。它能够将任意长度的输入消息转换为一个160 bit的固定长度输出。

RIPEMD-160的流程如图4所示，下面将对RIPEMD-160的具体流程进行详细介绍。

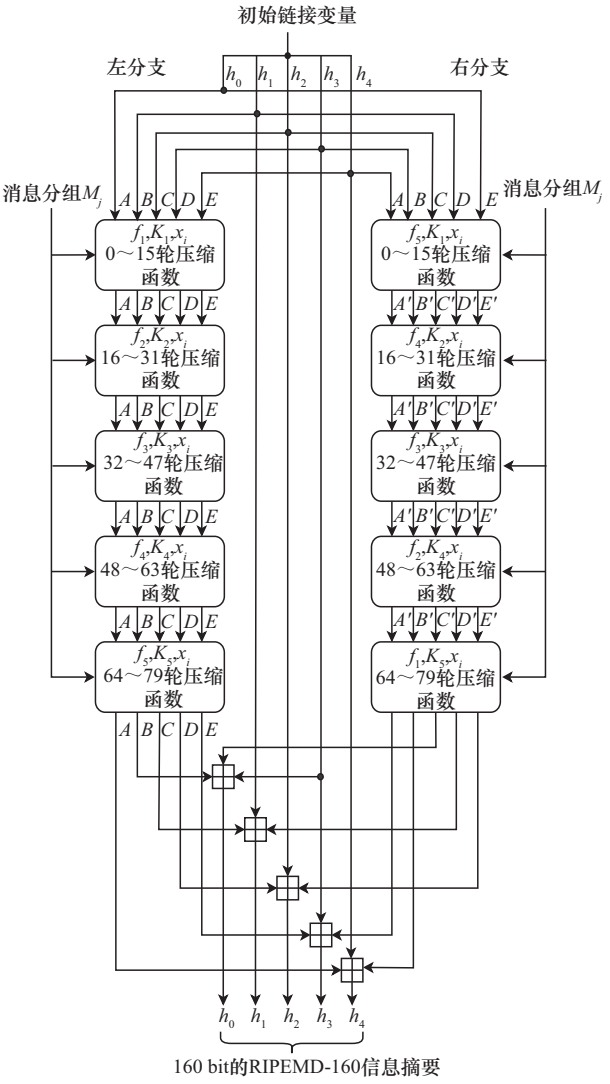



图4　RIPEMD-160流程

1) 消息填充与分组

RIPEMD-160的消息填充流程与SM3类似。首先，在消息的末尾添加一个“1”bit。接着，添加若干个“0”bit，使填充后的消息总长度$l$满足$l \equiv 448(\mathrm{mod}512)$。最后，在填充后的消息末尾添加一个64 bit的整数，该整数表示原始消息的长度（以bit为单位）。经过填充后的消息被分割成若干个512 bit的分组，设分组个数为$N$，分别记为$M_0$，$M_1$，$M_2$，$\cdots M_{N-1}$。每个512 bit的分组会被进一步划分为16个32 bit字，每个字写为$X[i]$。

2) 初始化链接变量

RIPEMD-160使用左右两分支各5个32 bit的链接变量来保存中间结果和生成最终的消息摘要，这些链接变量的初始值分别为：$h_0$=0x67452301，$h_1$=0xefcdab89，$h_2$=0x98badcfe，$h_3$=0x10325476，$h_4$=0xc3d2e1f0。初始化左右2条分支的变量：左分支$A=h_0$、$B=h_1$、$C=h_2$、$D=h_3$、$E=h_4$，右分支$A'=h_0$、$B'=h_1$、$C'=h_2$、$D'=h_3$、$E'=h_4$。

3) 压缩函数处理

压缩函数是RIPEMD-160的核心部分，其压缩函数独特之处在于双并行处理线，各含80步（5轮×16步/轮）。在2轮处理结束后，需要将处理前的链接变量与处理后的中间结果进行相加，得到新的链接变量值，作为下一个消息分组处理的初始链接变量。

左分支压缩函数单步计算过程如下：计算中间值$T = \mathrm{rol_}s[j]\left(A + f(B,C,D) + X\left[r(j)\right] + K(j)\right) + E$，其中$\mathrm{rol_}s[j]$表示循环左移$s(j)$位。加号表示为模$2^{32}$加法运算。$X\left[r(j)\right]$代表输入的16个输入字块中的第$r[j]$块。$s[j]$、$r[j]$的值已事先定义好，$K(j)$为32 bit固定轮常量。$f$函数在每一轮中有不同的定义，具体为

$$\begin{cases} f_1(B,C,D) = B \oplus C \oplus D, 0 \leqslant j \leqslant 15 \\ f_2(B,C,D) = (B \wedge C) \vee (\neg B \wedge D), 16 \leqslant j \leqslant 31 \\ f_3(B,C,D) = (B \vee \neg C) \oplus D, 32 \leqslant j \leqslant 47 \\ f_4(B,C,D) = (B \wedge D) \vee (C \wedge \neg D), 48 \leqslant j \leqslant 63 \\ f_5(B,C,D) = B \oplus (C \vee \neg D), 64 \leqslant j \leqslant 79 \end{cases} \tag{4}$$

右分支压缩函数与左分支在布尔函数的应用顺序、消息子分组的选取顺序、轮常数的取值、循环左移的移位量存在不同。

4) 生成最终消息摘要

当所有的消息分组都经过压缩函数处理后，最后得到的链接变量的值连接起来，就构成了160 bit的RIPEMD-160消息摘要。结果生成的过程如式(5)所示。

$$\begin{cases} h_0 = h_1 + C + D' \\ h_1 = h_2 + D + E' \\ h_2 = h_3 + E + A' \\ h_3 = h_4 + A + B' \\ h_4 = h_3 + B + C' \end{cases} \tag{5}$$

## 2　SM3的低T深度量子线路实现

### 2.1　低T深度量子加法器

在SM3和RIPEMD-160中，模加器是不可或缺的基础组件。为了实现低代价的量子加法器，目前

学界学者做了如下研究工作。Cuccaro 等[29]提出了一种新的量子串行进位加法器，其特点是只消耗 1 个辅助比特，但需要 61 个 Toffoli 深度。Draper 等[30]提出了量子超前进位加法器，显著降低了量子加法器的 Toffoli 深度。Wang 等[31]在文献[30]提出的量子超前进位加法器的基础上进行研究，进一步降低了量子模加器的 Toffoli 深度。出于减少量子比特消耗的考量，由于文献[31]所提出的量子加法器非原地实现，因此本文同时使用文献[31]所提出的量子加法器和文献[30]的量子加法器。在后文中统一称文献[31]所提出的量子加法器为加法器 1，称文献[30]所提出的量子加法器为加法器 2。为了降低 T 深度，本文将这 2 种加法器中的 Toffoli 门统一替换成 QAND 门。

加法器 1 探索了经典计算中算术线路在量子线路中的应用，通过采用斯克兰斯基（Sklansky）并行前缀树结构[32]，引入辅助量子位实现输入比特的并行计算。以下概述实现模 $2^4$ 加法器的步骤，该步骤描述了加法器 1，具体实现如图 5 所示，其中，$a_0$、$a_1$、$a_2$、$a_3$和$b_0$、$b_1$、$b_2$、$b_3$分别为加法器 1 的 2 个输入的低位到高位，$s_0$、$s_1$、$s_2$、$s_3$为加法器 1 的输出。

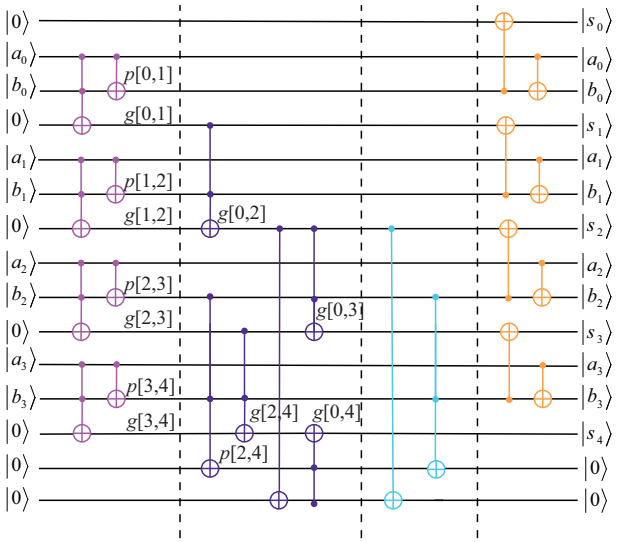



图 5　模 $2^4$ 量子超前进位加法器

加法器 1 的运行过程可分为以下 4 个步骤。

步骤 1：计算每位的生成信号 $g$ 与传播信号 $p$。在超前进位加法器中，$g$ 信号表示当前比特在加法运算中是否需要向高位生成进位，$p$ 信号表示当前比特是否需要向高位传播进位。

步骤 2：采用 Sklansky 量子前缀树结构计算 $g$ 信号值和 $p$ 信号值的传播。

步骤 3：使用 QAND$^\dagger$ 门重置中间变量的辅助量子位。

步骤 4：通过得到的每一位进位 $g[0,n]$ 计算最终结果。为了降低 T 深度，本文将文献[30-31]中的加法器中的 Toffoli 门用 QAND 门和 QAND$^\dagger$ 门替换，量子加法器资源消耗如表 2 所示。其中，文献[31]的加法器的计算需要 2 个 32 bit 的加数参与，需要 32 个量子比特来存储计算结果，需要 179 个可复位的辅助量子比特（包括 QAND 和 QAND$^\dagger$ 门以及其中并行计算所需要的辅助量子比特）。

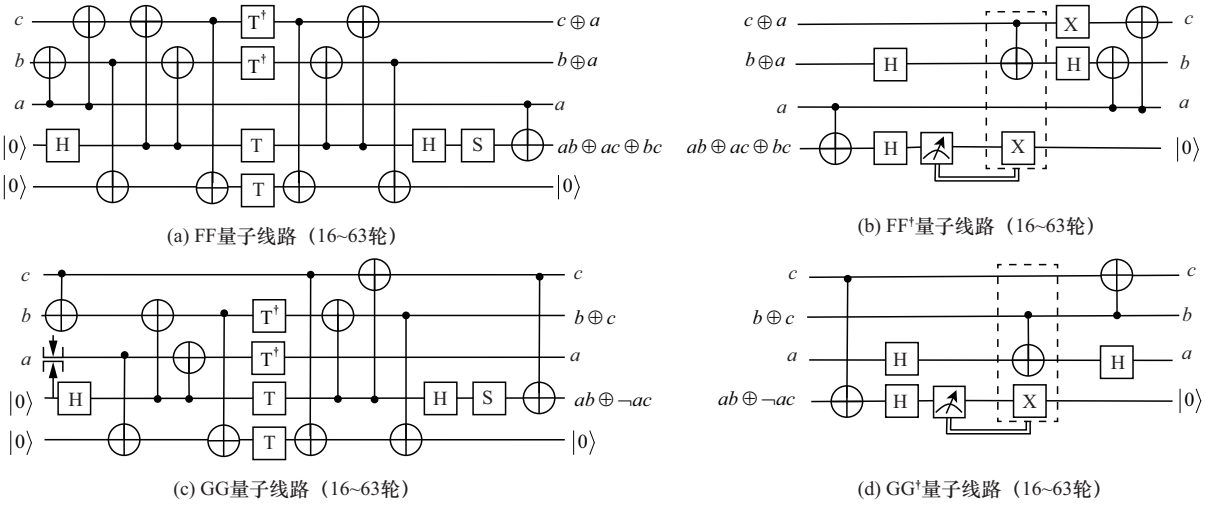
**表 2　　模 $2^{32}$ 量子模加器资源对比**

| 来源 | T 深度 | 量子比特数 | T 门 | CNOT 门 | 1qcliff 门 |
|---|---|---|---|---|---|
| 文献[29] | 183 | 61 | 427 | 157 | 58 |
| 文献[30] | 22 | 117 | 1 016 | 2 663 | 1 840 |
| 文献[31] | 8 | 275 | 1 088 | 2 468 | 845 |

## 2.2　SM3 量子线路整体实现

在每轮的消息扩展与迭代压缩量子线路中，首先需要计算当前轮的扩展字 $W_j = P_1(W_{j-16} \oplus W_{j-9} \oplus (W_{j-3} \lll 15)) \oplus (W_{j-13} \lll 7) \oplus W_{j-6}$。其中 $P_1$ 部分使用了 Zou 等[23]实现的就地量子实现，需要消耗 68 个 CNOT 门，除 $P_1$ 外的消息拓展线路使用了 4 个 32 bit 的 CNOT 运算，消耗 4×32=128 个 CNOT 门，在共计 64 轮的消息扩展部分共需消耗 16×32=512 量子比特。同时，并行执行 FF 和 GG 以得到 $\mathrm{FF}_j(A,B,C)$ 和 $\mathrm{GG}_j(E,F,G)$，如式(1)、式(2)所示，当 $0 \leqslant j \leqslant 15$ 时，FF 与 GG 不需要消耗 T 门，此时 FF、GG 消耗 3×2×32=192 个 CNOT 门；当 $16 \leqslant j \leqslant 63$ 时，FF 与 GG 分别为 MAJ 与 CH 函数。本文使用了 Yang 等[24]所设计的多数门（MAJ, majority gate）的非就地结构的量子线路 FF 及其逆线路 FF$^\dagger$，其中 FF 的 T 深度为 1，消耗 1 个辅助比特，FF$^\dagger$的 T 深度为 0，不消耗辅助比特。对于布尔函数 CH，本文使用 Zou 等[23]给出的非就地结构的量子线路 GG，其 T 深度为 1，消耗 1 个辅助比特。对于逆线路 GG$^\dagger$，其 T 深度为 0 且不消耗辅助比特。这 2 种设计在 T 深度消耗上均较低。各需要 4×32=128 个 T 门（包括 T$^\dagger$ 门），在此过程中消耗一个 T 深度。FF、GG 共消耗(11+10)×32=672 个 CNOT 门。本节所用到的 FF、FF$^\dagger$、GG、GG$^\dagger$分别如图 6(a)~(d)所示。

为了优化 T 深度，本文探索了 SM3 量子线路压




(a) FF量子线路（16~63轮）　(b) FF†量子线路（16~63轮）

(c) GG量子线路（16~63轮）　(d) GG†量子线路（16~63轮）

图6　FF、FF†、GG、GG†量子线路

缩函数中各组件的并行可能性，使用2.1节中提到的模32量子超前进位加法器和上文提到的布尔函数FF与GG。在SM3压缩函数的量子线路中，当计算出$\mathrm{FF}_j(A,B,C)$与$\mathrm{GG}_j(E,F,G)$之后，采用一系列的模加器与循环左移操作，本文根据模加器的并行运算方式将其分为7层，下文中若无具体说明，则默认使用的是2.1节中提到的加法器1。SM3压缩函数的量子线路如图7所示。

第一层：并行计算$W_{j-4}+\mathrm{GG}_j(E,F,G)$、$W_j+\mathrm{FF}_j(A,B,C)$、$E_{j-4}+(A_{j-4}\lll 12)$3个加法。

第二层：计算$(E_{j-4}+A_{j-4}\lll 12)+(T_{j-4}\lll j-4)$，在这之后，将其内容左移7 bit得到SS1，再与$A_{j-4}\lll 12$进行异或操作得到SS2。

第三层：并行执行$\left(W_j+\mathrm{FF}_j(A,B,C)\right)+\mathrm{SS2}$、$\left(W_{j-4}+\mathrm{GG}_j(E,F,G)\right)+\mathrm{SS1}$。

第四层：计算$\mathrm{TT1}=\left(W_j+\mathrm{FF}_j(A,B,C)+\mathrm{SS2}\right)+D_{j-4}$与$\mathrm{TT2}=\left(W_{j-4}+\mathrm{GG}_j(E,F,G)+\mathrm{SS1}\right)+H_{j-4}$，注意此处使用的是原地实现的加法器2，将计算的结果存储在原本存储$D_{j-4}$和$H_{j-4}$的量子比特上。至此已经完成了对中间值TT1、TT2的计算，然而为了实现量子比特复用，减少量子比特消耗，本文还需要逆序执行前面3层加法器操作来对辅助量子比特复位。

在这7层量子线路中，除了第四层中用到的是加法器2，其余都是加法器1。因此共需要12×1 088+2×1 016=15 088个T门、12×2 468+2×2 663=34 942个CNOT门、12×845+1 840×2=13 820个1qcliff门。T深度为6×8+22=70，加法器并行数量最多的为第1层，同时需要3个加法器并行，因此需要179×3=537个可复位的辅助量子比特。

在此之后，通过执行FF†与GG†将FF与GG所用的量子比特进行还原，再经过一些移位交换操作便可得到下一轮迭代压缩的输入。由于FF†与GG†都不含T门，此步骤不需要消耗T深度，当$0\leqslant j\leqslant 15$时，FF†与GG†消耗3×2×32=192个CNOT门；当$16\leqslant j\leqslant 63$时，FF†与GG消耗(4+3)×32=224个CNOT门。

综上，本文设计的SM3量子线路产生最终输出结果需要$j$=16~63共计48轮的FF、GG共128×2×48=12 288个T门（包括T†门），加法器中消耗15 088×64=965 632个T门，共计977 920个T门。消息扩展部分与压缩函数中消耗32×8×64=16 384个CNOT门，FF与GG消耗192×16+672×48=35 328个CNOT门，FF†与GG†消耗192×16+224×48=13 824个CNOT门，P0($X$)和P1($X$)共消耗68×2×64=8 704个CNOT门，加法器部分消耗34 942×64=2 236 288个CNOT，共计2 310 528个CNOT门。加法器部分消耗13 820×64=884 480个1qcliff门。FF与GG及其逆线路共消耗(3+4+3+4)×32×48=21 504个1qcliff门，共计905 984个1qcliff门。线路的消息扩展部分需要16×32=512量子比特，迭代压缩部分A~H寄存器需要8×32=256量子比特，由于在每轮加法深度中最多使用了3个加法器并行，需要537个可复位的辅助量子比特，FF和GG中需要32×2=64个可复位的量子比特来存储结果，共计512 + 256 +

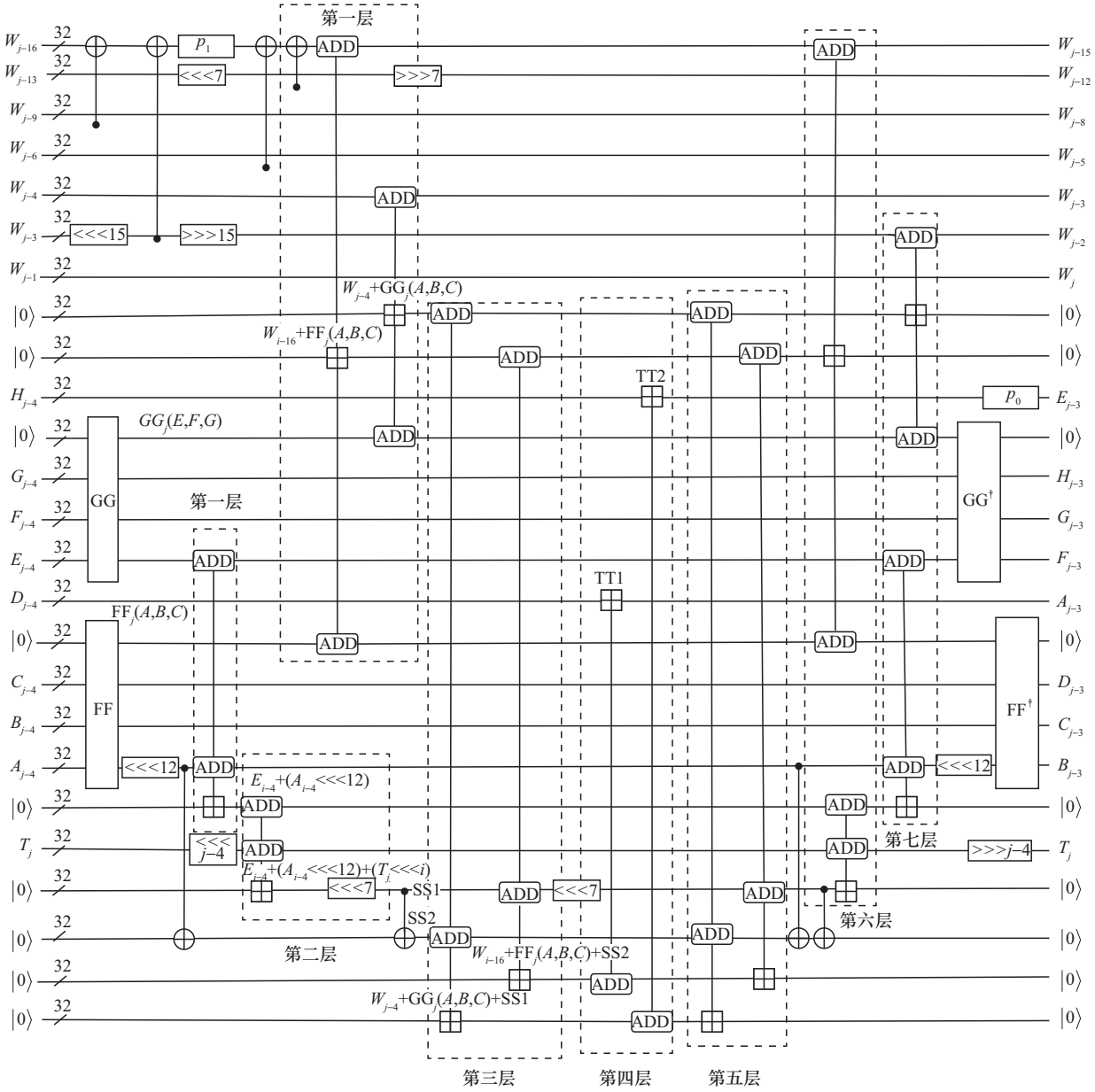



图 7　SM3 压缩函数量子线路

537+64 = 1 369 个量子比特。T 深度为 70×16+(1+70)×48=4 528。T-DW-cost 值为 1 369×4 528=6 198 832。

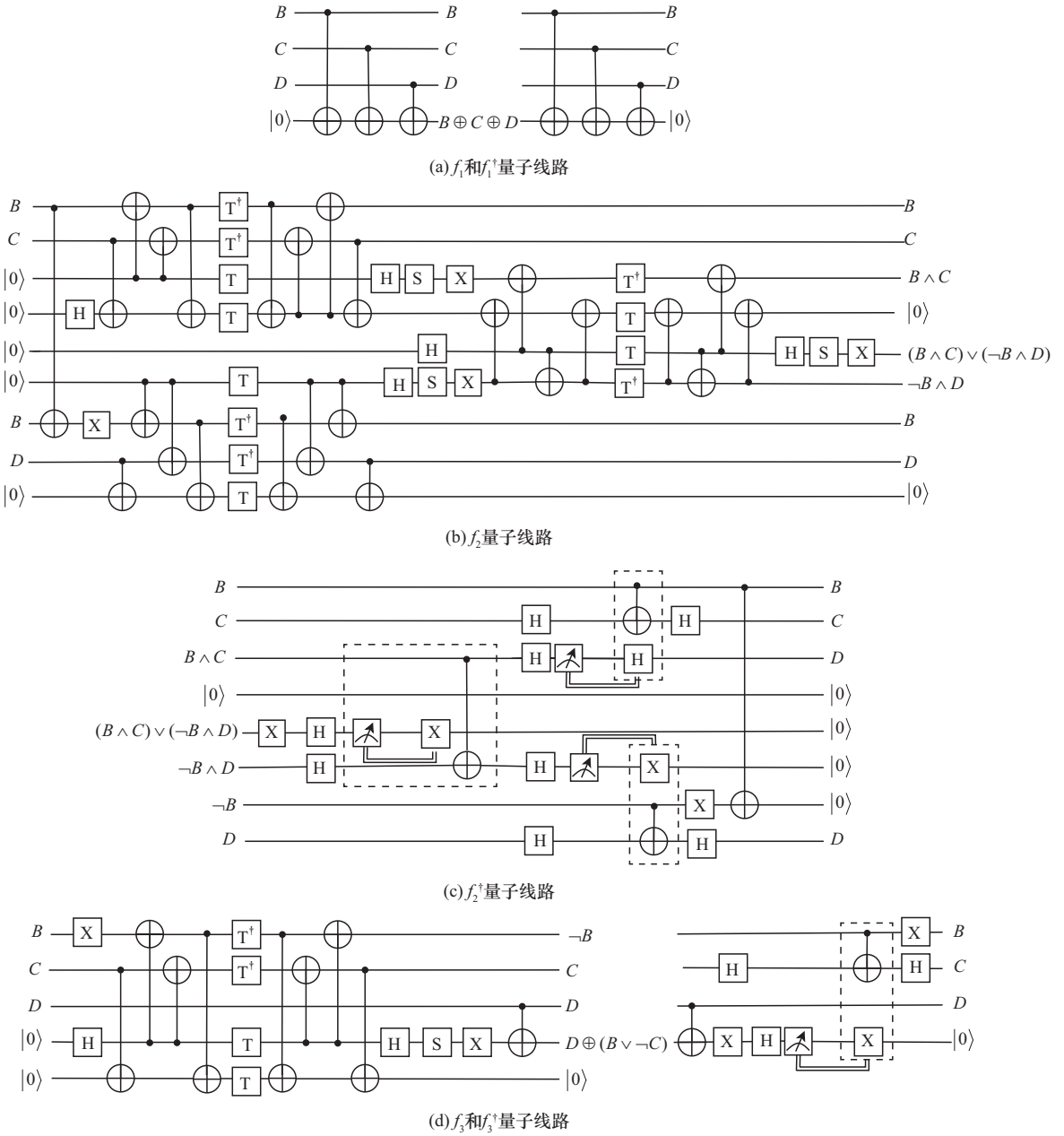
## 3　RIPEMD-160 的低 T 深度量子线路实现

### 3.1　RIPEMD-160 布尔函数的量子实现

RIPEMD-160 压缩函数中的 5 个布尔函数的运算逻辑是整个算法的核心组成部分，如式(4)所示。由于这 5 个函数中存在非线性运算，将它们映射到量子线路时会产生一定的 T 深度。因此，针对 RIPEMD-160 的 5 个布尔函数，本文分别设计了量子线路实现方案，核心目标是在保证函数运算逻辑正确性的前提下，降低 T 深度。以下将详细阐述这 5 个布尔函数的量子线路设计思路，如图 8 所示。

对于 $f_1 = B \oplus C \oplus D$，其中并不存在非线性操作，因此本文只需要使用 3 个 CNOT 门来实现 $f_1$ 及其逆线路。如图 8(a)所示。

对于 $f_2 = (B \wedge C) \vee (\neg B \wedge D)$，根据数字逻辑原理进行设计，$(B \wedge C) \vee (\neg B \wedge D)$ 等价于 $\neg(\neg(B \wedge C) \wedge \neg(\neg B \wedge D))$。据此，先将 $B$ 复制到一个辅助量子比特上，并对该辅助量子比特执行量子非门，得到 $\neg B$，接着分别对 $B$、$C$ 和 $\neg B$、$D$ 执行




(a) $f_1$和$f_1^{\dagger}$量子线路

(b) $f_2$量子线路

(c) $f_2^{\dagger}$量子线路

(d) $f_3$和$f_3^{\dagger}$量子线路

图8　RIPEMD-160布尔函数的量子线路设计

QAND门，将$(B \wedge C)$和$(\neg B \wedge D)$的结果暂存到2个量子比特上，接着对这2个量子比特执行量子非门和QAND门，得到$\neg(B \wedge C) \wedge \neg(\neg B \wedge D)$，最后执行量子非门，至此便实现了$\neg(\neg(B \wedge C) \wedge \neg(\neg B \wedge D))$，即$f_2 = (B \wedge C) \vee (\neg B \wedge D)$。同理，对于其逆线路$f_2^{\dagger}$的实现，先对其执行一个量子非门，得到$\neg(B \wedge C) \wedge \neg(\neg B \wedge D)$，对其执行一次QAND$^{\dagger}$门将其复位，接着分别对$\neg(B \wedge C)$和$\neg(\neg B \wedge D)$执行量子非门和QAND$^{\dagger}$门将其复位。至此，完成了对$f_2$和$f_2^{\dagger}$的量子线路设计，$f_2$消耗2个T深度，$f_2^{\dagger}$不消耗T深度。对于$f_4 = (B \wedge D) \vee (C \wedge \neg D)$，只需要交换$f_2$量子线路的$B$和$D$即可实现。$f_2$量子线路如图8(b)所示，$f_2^{\dagger}$量子线路如图8(c)所示

对于$f_3 = (B \vee \neg C) \oplus D$，同样根据数字逻辑原理进行设计，$(B \vee \neg C) \oplus D$等价于$\neg(\neg B \wedge C) \oplus D$。据此，先对$B$执行量子非门得到$\neg B$，接着对$\neg B$、$C$执行一个QAND门，得到$(\neg B \wedge C)$，接着对$(\neg B \wedge C)$执行量子非门，至此便实现了$\neg(\neg B \wedge$

$C$)，即($B \vee \neg C$)，最后只需将$D$异或到($B \vee \neg C$)上即可。同理，对于其逆线路$f_2^{\dagger}$的实现，先对($B \vee \neg C$)$\oplus D$和$D$执行一次异或操作得到($B \vee \neg C$)，再对其执行量子非门和一次QAND$^{\dagger}$门将其复位，最后对$\neg B$执行量子非门将其还原。至此，完成了对$f_3$和$f_3^{\dagger}$的量子线路设计，$f_3$消耗1个T深度，$f_3^{\dagger}$不消耗T深度。对于$f_5 = B \oplus (C \vee \neg D)$，只需要交换$f_3$量子线路的3个输入位即可实现。$f_3$和$f_3^{\dagger}$量子线路如图8(d)所示。

### 3.2　RIPEMD-160量子线路整体实现

为了优化T深度，本文探索了RIPEMD-160量子线路压缩函数中各组件的并行可能性，最终获得低T深度的RIPEMD-160量子线路，如图9所示。

在单步压缩函数的量子线路设计中，首先执行的是布尔函数$f$，在这里使用3.1节提出的布尔函数$f_1$~$f_5$的量子实现。其中，一组$f_1$和$f_1^{\dagger}$共需要6个CNOT门，一个可复位的辅助量子比特；一组$f_2$和$f_2^{\dagger}$共需要12个$T$门（包括$T^{\dagger}$门）、29个CNOT门、25个1qcliff门，$f_2$消耗2个T深度，$f_2^{\dagger}$不消耗T深度，$f_4$和$f_4^{\dagger}$的量子线路与其相同。一组$f_3$和$f_3^{\dagger}$共需要4个T门（包括T$^{\dagger}$门，下同）、10个CNOT门、9个1qcliff门。

为了计算出中间值$T = \mathrm{rol_}s[j]\big(A + f(B,C,D) + X\big[r(j)\big] + K(j)\big) + E$，在布尔函数执行完成后，加法操作被排列成5层：第一层：$X\big[r(j)\big] + K(j)$可与$A + f(B,C,D)$的加法并行计算；第二层：执行$\big(A + f(B,C,D)\big) + \big(X\big[r(j)\big] + K(j)\big)$，并对其执行循环左移$s[j]$位；第三层：执行最后的加法操作$T = \mathrm{rol_}s[j]\big(A + f(B,C,D) + X\big[r(j)\big] + K(j)\big) + E$，这一步中采用加法器2，将结果存储在原本存储$E$的量子比特上，此时已经完成中间值$T$的计算，然而为了实现量子比特复用，减少量子比特消耗，还需要逆序执行前面两层加法器操作来对辅助量子比特复位。先对存储$\Big(\big(A + f(B,C,D)\big) + \big(X\big[r(j)\big] + K(j)\big)\Big) \lll S[j]$的量子比特执行循环右移$s[j]$位，并再次执行第二层中的加法线路，最后并行执行第一层中的加法操作$X\big[r(j)\big] + K(j)$与$A + f(B,C,D)$对这2处存储临时结果的量子比特进行复位。在加法器操作之后，需

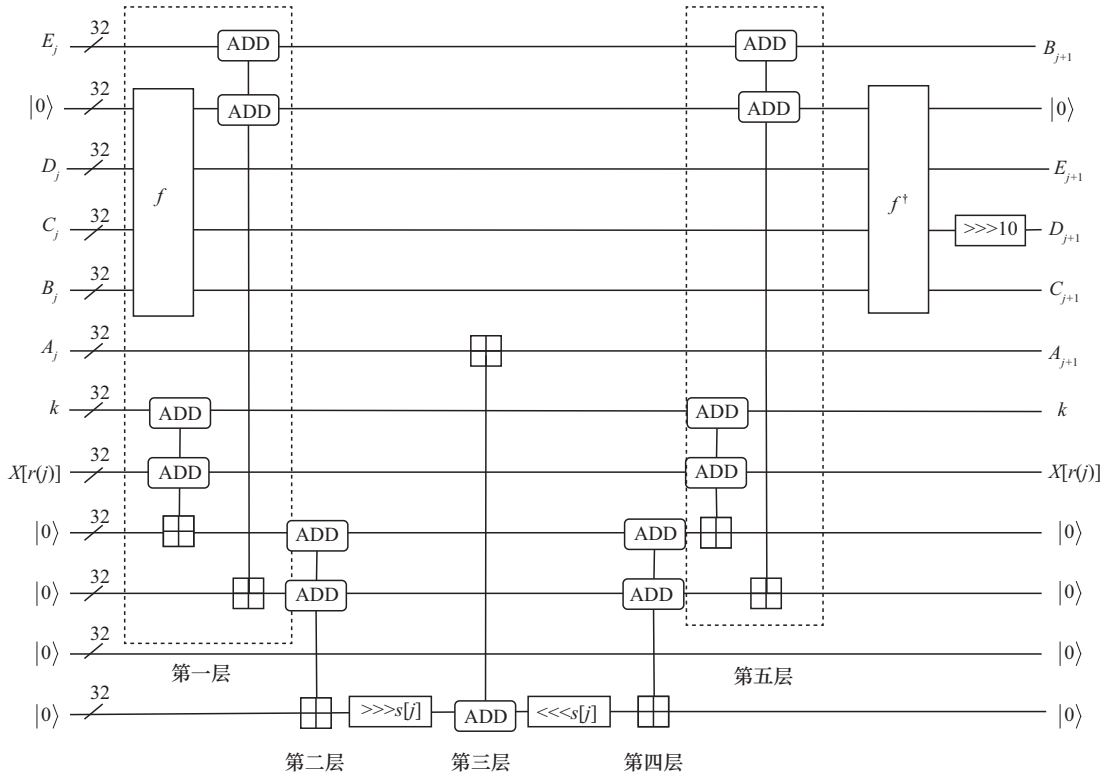



图9　RIPEMD-160第$j$步压缩函数的量子线路设计

要对存储$f(B,C,D)$结果的量子比特执行$f^{\dagger}$操作使其复位，并对$C$进行循环左移10位的操作以得到下一步压缩函数的输入$D$。

下面计算一轮轮函数加法操作所需的量子资源数。注意到，一轮轮函数操作中需要进行6次量子加法器1操作与1次量子加法器2操作，即共消耗1 088×6+1016×1=7 544个T门，2 468×6+2 663×1=1 7471个CNOT门，845×6+1 840×1=6 910个1qcliff门，179×2+32×3=454个可复位的辅助量子比特，T深度为4×8+22=54。

每轮布尔函数的量子资源数可以计算如下，0~15轮的$f_1$和$f_1^{\dagger}$需要6×16=96个CNOT门，需要32个可复位的量子比特用于存储$f_1$的结果；16~31轮的$f_2$和$f_2^{\dagger}$需要12×16=192个T门、29×16=464个CNOT门，25×16=400个1qcliff门，需要32×4=128个可复位的量子比特用于存储$f_2$的结果以及中间的临时结果，T深度为2×16=32。32~47轮的$f_3$和$f_3^{\dagger}$需要4×16=64个T门、10×16=160个CNOT门，9×16=144个1qcliff门，需要32个可复位的量子比特用于存储$f_3$的结果，T深度为16。48~63轮需要的资源与16~31轮相同，64~79轮需要的资源与32~47轮相同。右分支量子线路所需要的资源与左分支相同。

综上，压缩函数的左右分支共需要7 544×80+192×2+64×2=604 032个T门、17 471×80+464×2+160×2=1 398 928个CNOT门，6 910×80+400×2+144×2=553 888个1qcliff门，需要5×32=160个量子比特来存储5个寄存器的初始值，454+128+32=614个可复位的辅助量子比特。因此，左右两分支共需要1 208 064个T门，2 797 856个CNOT门，1 107 776个1qcliff门，160×2+614×2=1 548个量子比特，T深度为54×80+32×2+16×2=4416。除此之外，还需要10×32个量子比特来存储轮常量$K(j)$的值。

为了计算最终的输出结果，还需要执行如式(5)所示的加法运算。采用如图10所示的量子线路设计，需要10个量子加法器1和5个量子加法器2。共需要1 088×10+1 016×2=12 912个T门，2 468×10+2 663×2=30 006个CNOT门，845×10+1 840×2=12 130个1qcliff门，以及179×5=895个可复位的辅助量子比特，小于左右两分支压缩函数所需要的可复位的量子比特，因此无需重复计算，T深度为38。

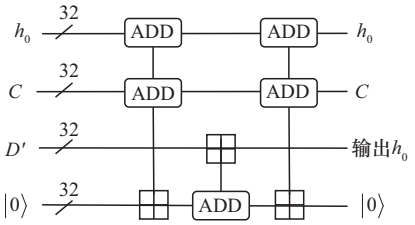



图10 RIPEMD-160输出计算的量子线路设计

综上，本文所设计的RIPEMD-160量子线路产生160 bit输出结果共需要1 208 064+12912=1 220 976个T门，2 797 856 +30 006=2 27 862个CNOT门，1 107 776+12130=1 119 906个1qcliff门，168个量子比特，T深度为4 416+38=4 454。T-DW-cost为1 868×4454=8320 072。

## 4 结束语

本文面向量子计算对密码算法带来的安全评估需求，研究了SM3与RIPEMD-1602种杂凑函数的高效量子线路实现问题。首先，通过引入低T深度量子加法器并对量子组件进行重新排列，提出了一种新型低T深度SM3量子线路，该量子线路在T深度与T-DW-cost上均优于现有方案。其次，针对RIPEMD-160，首次构建了其布尔函数的低T深度量子实现，并优化压缩函数的分层并行结构，成功实现其量子线路，T深度仅为4 454，显著降低了资源消耗。在未来的工作中，我们将对量子比特数展开优化，通过改进量子组件复用策略、优化线路布局等方式，进一步降低资源开销。同时，将本文提出的低T深度量子线路设计方法推广至更多杂凑函数与对称密码算法，探索其在不同类型密码方案量子实现中的适配性与有效性，为密码算法的量子安全性评估提供更全面的技术支撑。

## 参考文献：

[1] SHOR P W. Algorithms for quantum computation: discrete logarithms and factoring[C]//Proceedings of the 35th Annual Symposium on Foundations of Computer Science. Piscataway: IEEE Press, Aug. 2002: 124-134.

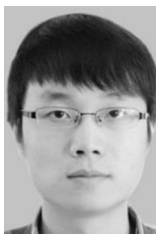
[2] MILANOV E. The RSA algorithm[J]. RSA laboratories, 2009, 1(11): 1-11.

[3] KOBLITZ N, MENEZES A, VANSTONE S. The state of elliptic curve cryptography[J]. Designs, Codes and Cryptography, 2000, 19(2): 173-193.

[4] GROVER L K. A fast quantum mechanical algorithm for database search[C]//Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing - STOC'96. New York: ACM Press, 1996:

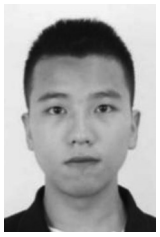
212-219.
[5] SELENT D. Advanced encryption standard[J]. Rivier Academic Journal, 2010, 6(2): 1-14.
[6] PENARD W, VAN WERKHOVEN T. On the secure hash algorithm family[J]. Cryptography in Context, 2008: 1-18.
[7] DWORKIN M J. SHA-3 standard: permutation-based hash and extendable-output functions[R].2015.
[8] SIMON D R. On the power of quantum computation[J]. SIAM Journal on Computing,1997, 26(5): 1474-1483.
[9] GRASSL M, LANGENBERG B, ROETTELER M, et al. Applying Grover's algorithm to AES: quantum resource estimates[C]//Proceedings of the 7th International Workshop on Post-Quantum Cryptography. Berlin: Springer, 2016: 29-43.
[10] ALMAZROOIE M, SAMSUDIN A, ABDULLAH R, et al. Quantum reversible circuit of AES-128[J]. Quantum Information Processing, 2018, 17(5): 112.
[11] LANGENBERG B, PHAM H, STEINWANDT R. Reducing the cost of implementing the advanced encryption standard as a quantum circuit[J]. IEEE Transactions on Quantum Engineering, 2020, 1: 2500112.
[12] ZOU J, WEI Z H, SUN S W, et al. Quantum circuit implementations of AES with fewer qubits[C]//International Conference on the Theory and Application of Cryptology and Information Security. Berlin: Springer, 2020: 697-726.
[13] WANG Z G, WEI S J, LONG G L. A quantum circuit design of AES requiring fewer quantum qubits and gate operations[J]. Frontiers of Physics, 2022, 17(4): 41501.
[14] LI Z Q, GAO F, QIN S J, et al. New record in the number of qubits for a quantum implementation of AES[J]. Frontiers in Physics, 2023, 11: 1171753.
[15] HUANG Z Y, ZHANG F X, LIN D D. Constructing quantum implementations with the minimal t-depth or minimal width and their applications[C]//Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2025: 155-185.
[16] LI Z Q, CAI B B, SUN H W, et al. Novel quantum circuit implementation of advanced encryption standard with low costs[J]. Science China Physics, Mechanics & Astronomy, 2022, 65(9): 290311.
[17] HUANG Z Y, SUN S W. Synthesizing quantum circuits of AES with lower T-depth and less qubits[C]//International Conference on the Theory and Application of Cryptology and Information Security, Berlin: Springer, 2022: 614-644.
[18] LIN D, XIANG Z J, XU R Q, et al. Optimized quantum implementation of AES[J]. Quantum Information Processing, 2023, 22(9): 352.
[19] LIU Q, PRENEEL B, ZHAO Z, et al. Improved quantum circuits for AES: reducing the depth and the number of qubits[C]//International conference on the theory and application of cryptology and information security. Berlin: Springer, 2023: 67-98.
[20] SHI H T, FENG X T. Quantum circuits of AES with a low-depth linear layer and a new structure[C]//International Conference on the Theory and Application of Cryptology and Information Security. Berlin: Springer, 2024: 358-395.
[21] 王小云, 于红波. SM3 密码杂凑算法[J]. 信息安全研究, 2016(11): 983-994.
WANG X Y, YU H B. SM3 cryptographic hash algorithm[J]. Journal of Information Security Research, 2016(11): 983-994.
[22] SONG G, JANG K, KIM H, et al. Grover on SM3[C]//International Conference on Information Security and Cryptology. Berlin: Springer, 2021: 421-433.
[23] ZOU J, LI L J, WEI Z H, et al. New quantum circuit implementations of SM4 and SM3[J]. Quantum Information Processing, 2022, 21(5): 181.
[24] 杨婷玉, 张莎莎, 向泽军, 等. SM3 算法的低T深度与低宽度量子优化实现[J]. 中国科学: 物理学 力学 天文学, 2025, 50: 013002.
YANG T Y, ZHANG S S, XIANG Z J, et al. Optimized quantum implementations of SM3 with low T-depth and low width[J]. Scientia Sinica (Physica, Mechanica & Astronomica), 2025, 50: 013002.
[25] DOBBERTIN H, BOSSELAERS A, PRENEEL B. RIPEMD-160: a strengthened version of RIPEMD[C]//International Workshop on Fast Software Encryption. Berlin: Springer, 1996: 71-82.
[26] RIVEST R. The MD5 message-digest algorithm[R]. 1992.
[27] AMY M, MASLOV D, MOSCA M, et al. A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2013, 32(6): 818-830.
[28] JAQUES S, NAEHRIG M, ROETTELER M, et al. Implementing Grover oracles for quantum key search on AES and LowMC[C]//Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2020: 280-310.
[29] CUCCARO S A, DRAPER T G, KUTIN S A, et al. A new quantum ripple-carry addition circuit[J]. arXiv Preprint, arXiv: 0410184, 2004.
[30] DRAPER T G, KUTIN S A, RAINS E M, et al. A logarithmic-depth quantum carry-lookahead adder[J]. Quantum Information & Computation, 2006, 6(4): 351-369.
[31] WANG S Y, MONDAL A, CHATTOPADHYAY A. Optimal toffoli-depth quantum adder[J]. ACM Transactions on Quantum Computing, 2025, 6(3): 1-16.
[32] SKLANSKY J. Conditional-sum addition logic[J]. IRE Transactions on Electronic Computers, 1960, 9(2): 226-231.

[作者简介]



**邹剑**（1985– ），男，福建福州人，博士，福州大学副教授、硕士生导师，主要研究方向为对称密码分析、量子计算。

**郭楠生**（2001– ），男，回族，福建泉州人，福州大学硕士生，主要研究方向为线路实现、S盒优化。



**李俊康**（2000– ），男，福建宁德人，福州大学硕士生，主要研究方向为线路实现、S盒优化。