基于 MLWE 的格密码高效硬件实现

崔益军, 姚 衍, 倪子颖, 王成华, 刘伟强

南京航空航天大学电子信息工程学院 南京 中国 210000

摘要 后量子密码的发展已经引起各界的广泛关注, 硬件实现效率是后量子密码最终标准的重要衡量指标之一。其中基于模误差学习问题(Module Learning With Errors, MLWE)的 CRYSTALS-Kyber 格密码是 NIST 第三轮后量子密码标准中最有希望的一种加密方案, 可变的公钥矩阵维度参数 k 将基于 MLWE 的公钥加密方案的安全性扩展到不同级别, 相较于其他格密码方案更具灵活性和安全性。本文首先分析了基于 NIST 第三轮最新参数 q=3329 的 MLWE 的格密码公钥加密方案的算法理论, 并针对其中的核心模块—多项式乘法模块提出了两种不同的硬件实现方式。两种多项式乘法硬件实现方式都是采用基于频率抽取的数论变换(Number Theoretic Transform, NTT)算法,使用 NTT 算法实现多项式乘法降低了传统算法实现的线性复杂度, 在硬件结构上能够面对不同应用场景进行优化, 因此本文针对 NTT 算法中循环计算的核心模块提出了两种不同的优化硬件结构。一是面积和执行时间折中的迭代型 NTT 硬件结构, 二是高性能低时延的多路延时转接(Multi-path Delay Commutator)的流水型 NTT 硬件结构; 并且针对于面积时间均衡的迭代型NTT模块设计了一种整体MLWE硬件实现结构。与已有的先进设计相比, 本文的流水型NTT结构具备更好的速度性能, 在速度上相较于之前的设计分别提升 11.64%和 59.43%。而对于使用迭代型 NTT 的 MLWE 整体实现方案, 本文的设计使用了最少的周期和最小的面积时间乘积 (Area-Time-Product, ATP),其效率比最新发表的工作的硬件效率实现高 2 倍左右。

关键词 后量子密码; 格密码; 数论变换算法; 模误差学习问题

中图法分类号 TN918 DOI 号 10.19363/J.cnki.cn10-1380/tn.2021.11.04

Efficient Hardware Implementation of MLWE Lattice Based Cryptography

CUI Yijun, YAO Kan, NI Ziying, WANG Chenghua, LIU Weiqiang

College of Electronics and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210000, China

Abstract The development of post-quantum cryptography has attracted widespread attention from all walks of life, and its hardware implementation efficiency is also one of the important metrics for the final standard of post-quantum cryptography. Among them, the MLWE-based CRYSTALS-Kyber lattice cipher is the most promising encryption scheme in the Round 3 of NIST post-quantum cipher standards. The variable public key matrix dimension parameter k extends the security of the MLWE-based public key encryption scheme to different levels. Compared with other lattice password schemes, it is more flexible and safer. This paper first analyzes the primes q=3329 in Round 3 of NIST process based on the algorithm theory of the MLWE-based lattice cipher public key encryption scheme and proposes two different implementation methods for the core module—polynomial multiplication module. The two implementation methods are number theoretic transform (NTT) algorithm based on frequency extraction. The use of NTT algorithm to achieve polynomial multiplication reduces the linear complexity, and can be optimized for different application scenarios in the hardware structure. Therefore, this paper proposes two different optimized hardware structures for the core module of cyclic calculation in the NTT algorithm. One is the iterative NTT hardware structure with the compromise of area and execution time, and the other is the pipelined NTT hardware structure with high-performance and low latency multi-path delay commutator (Multi-path Delay Commutator) structure; and designed an overall area-time balanced MLWE hardware implementation structure for the iterative NTT module. Compared with the state-of-the-art designs, the pipelined NTT structure in this paper shows better speed performance, although it has slightly more resources than the previous structure and is 11.64 faster than the previous design. Compared with the previous design, the speed is increased by 11.64% and 59.43%. As for the overall implementation of MLWE using iterative NTT, the design in this paper uses the least cycle and the smallest ATP, and its efficiency is about 2 times higher than the latest hardware implementation.

Key words post-quantum cryptography; lattice-based cryptography; number theoretic transform; module learning with errors

通讯作者: 刘伟强, 教授, Email: liuweiqiang@nuaa.edu.cn。

本课题得到国家重点自然科学基金项目(No. 62134002)和青年自然科学基金项目(No. 62104107)资助。

收稿日期: 2021-09-01; 修改日期: 2021-10-15; 定稿日期: 2021-10-22

1 引言

信息安全在是现代信息社会的建设基础及发展保障, 各种应用信息的传输、交换与存储都是在公认足够安全的密码体制保护下进行的。传统的基于公钥加密方案的安全性依赖于一些数学困难问题, 例如基于大整数分解问题的 RSA(Rivest Shamir Adleman)算法和基于离散对数问题的椭圆曲线密码(Elliptic Curve Cryptography, ECC)算法[1], 这些数学困难问题在经典计算机的计算能力下是难以破解的。但是, Peter Shor 在 1994 年提出了著名的量子算法[2], 在强大的量子计算机问世后可以轻松解决传统公钥加密方案的数学困难问题。虽然量子计算机的发展仍处于初级阶段, 但随着量子计算机技术的成熟, 由于底层依赖的数学问题被解决, 现有的设备中所采用的公钥密码算法将被完全攻破。因此, 亟需研究出能够抵抗量子计算机攻击的新一代密码算法。

1.1 后量子密码标准化进程

后量子密码算法(Post-Quantum Cryptography, PQC)也称抗量子密码算法, 美国国家标准技术研究所 (National Institute of Standards and Technology, NIST)早在 2012 年启动了后量子密码的研究工作, 并于 2016 年 2 月启动了全球范围内的后量子密码标准征集, 相关时间线如图 1 所示。NIST 在制定后量子密码算法的标准时主要聚焦于公钥加密、密钥交换和数字签名。在 NIST 第一轮公布的 69 个初次的候选方案中, 主要包括以下 4 种数学方法构造的后量子密码算法:

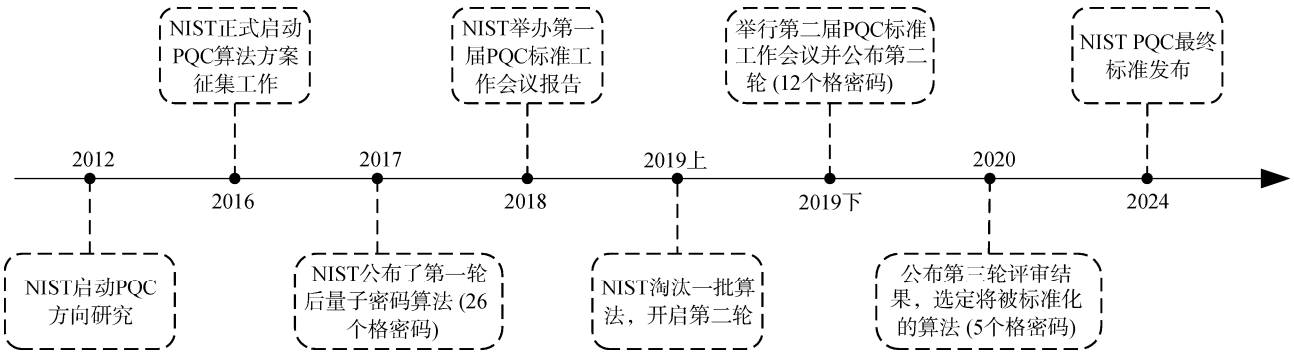

图 1 后量子密码标准的发展时间线

Figure 1 NIST PQC standardization timeline

1) 基于格的密码(Lattice-based Cryptography, LBC)。其通过格上的数学困难问题, 如发现格上 n 个线性无关的短向量构造基于格的后量子密码。格密码以其高效率轻量级公钥而闻名, 使得它成为高速网络应用和物联网应用的首选, 也是标准化过程中最有希望成为最终候选方案的后量子密码算法。

2) 基于编码的密码(Code-based Cryptography, CBC)。其通过使用错误纠正码对加入的随机性错误进行纠正和计算, McEliece 是一种经典的基于编码的后量子密码, 即使其公钥尺寸过大但凭借非常小的密文, McEliece 在后量子密码标准化过程中始终有自己的一席之位。

3) 多变量多项式的密码(Multivariate Polynomial Cryptography, MPC)。其通过在有限域上具有多个变量的二次多项式方程组构造新的密码方案, 提供了快速的签名和验证, 在后量子密码标准化进程中 Rainbow 作为该类密码方案的代表, 被选为最终标准的数字签名**最终候选方案**。

4) 基于哈希签名(Hash-based signatures, HBS)的密码。其不依赖于某一个特定的哈希函数构造数字签名, 但考虑到其较其他三类后量子密码密钥过长且成本较大, 在标准化进程中逐渐被淘汰。

在 NIST 公布的通过第一轮筛选的 69 项草案中, 基于格的密码方案多达 26 种。在接下来的两年中, NIST 根据候选算法的抗侧信道攻击性、实现性能、成本和其他特性对其进行评估。2019 年, NIST 选择了 26 种算法进入第二轮进行更深层次的分析, 其中有将近一半为格密码算法。2020 年 7 月, 第三轮后量子密码标准入围名单公布[3], 除了 7 种最终被考虑标准化的决赛方案, 此外还有 8 个备选方案也将进入第三轮的考察中。其中在 7 个**最终候选方案**入围的公钥加密和密钥交换中, 基于格的协议有 NTRU、

CRYSTALS-Kyber 和 SABER ; 入围数字签名的格密码方案有两种分别是 DILITHIUM 和 FALCON。由后量子密码标准化进程的发展, 可以观察到格密码方案的多样性以及成为最终标准后量子密码方案的潜力性。后量子密码算法的硬件实现性能也是成为最终标准的衡量指标, 格密码能够在硬件平台上的高效实现也是其优势之一, 但是在物联网技术的飞速发展的环境下, 如何实现轻量级高性能的格密码并使其适用于物联网芯片是推动后量子密码标准化进程的研究重点。

1.2 国内外研究现状

在众多的后量子密码方案中, 基于格的方案凭着自身众多优异的特点, 成为了最有潜力的后量子密码方案。2005 年, Regev[4]首次提出了基于错误学习问题(Learning With Errors, LWE)的可严格证明理论安全性的格密码方案, 相比较于传统的公钥加密方案, 基于 LWE 问题的公钥加密方案结构更简单, 运算速度更快, 但随安全系数正比增长的公钥长度限制了实用性。为了解决这个问题, Lyubashevsky 于 2010 年提出了环域上的误差学习(Ring-LWE, RLWE)问题[5], RLWE 在实现效率和公钥长度的优越性使其在提出后备受学术界的关注, 也有相关研究人员将其应用到物联网设备中。但是基于 RLWE 的格密码方案结构化程度最高, 也就意味着其安全性受到了影响, 因此在 NIST 公布的第三轮后量子密码标准中基于 RLWE 的格密码方案全部落选。综合 LWE 和 RLWE 的特点, Brakerski 和 Langlois 等人[6]在 2015 年提出了基于模误差学习问题(Module-LWE, MLWE)的格密码方案, MLWE 问题相比较于 RLWE 方案具有更复杂的代数结构和更高的安全性, 同时具备比 LWE 方案更高的性能。事实上, 在第三轮后量子密码标准中, 公钥加密方案 CRYSTALS-Kyber 以及数字签名方案 DILITHIUM 都是基于 MLWE, SABER 方案也是基于 MLWE 问题的变体模带舍入学习(Module Learning With Rounding, MLWR)困难问题[7]。由此可见, 基于 MLWE 的格密码在后量子密码算法中的具备主导地位。本文的主要内容便是探讨基于 MLWE 的格密码算法的高效硬件实现方案。

随着 NIST 对后量子密码标准化的推进, 关于格密码的硬件实现研究也步入了快速发展的轨道。在 NIST 第三轮 PQC 密码方案公布之前, R-LWE 公钥加密方案是被广泛研究的格密码方案。2012 年, 文献[8]中首次采用了快速数论变换(Number theoretic transforms, NTT)对 RLWE 密码进行了硬件设计, 并与 LWE 的矩阵运算进行了性能对比。2014 年, Sujoy[9]对 RLWE 加密方案优化提出了最经典的紧凑型设计, 优化 NTT 算法的预处理和后处理时间、数据存储和读取方式以及旋转因子的存储等方面, 在面积、频率和执行时间上都有较大的优势。为了追求更高的性能, 研究者在核心的多项式乘法模块对 NTT 算法进行了不同的优化设计, 文献[10]中 DU 采用了 4 个蝶形单元并行, 合理安排存储单元中的存储顺序得到了高速的多项式乘法模块。清华大学李树国教授团队于 2019 年提出了一种多路径且高并行度的 NTT 算法[11], 该算法受 Stockham FFT 算法启发, 通过大量蝶形单元并行来提升 NTT 模块的运算速度。这些硬件优化为第三轮格密码候选方案的实现提供了可参考的设计方案, 文献[12]在紧凑型 RLWE 设计方案的基础上提出了 q=7681 的 MLWE 的高效硬件实现, 但该方案不能兼容最新参数。在 2021 年, Mojtaba[13]提出了将基于 K^2-RED 算法的模 3329 约减算法无寄存器延迟的硬件单元应用到 4 个并行计算的 NTT 蝶形单元从而实现高速设计。文献[14]首次提出了 Kyber 密钥交换协议的完整硬件实现, 但硬件资源利用率较低, 同时执行的时间周期也较长。Xing[15]在 CHES2021 提出了一种时序安排紧密, NTT 单元采用基于时域和基于频率的结合型蝶形单元, 有关多项式的加减法操作也复用该蝶形单元完成, 该设计相较于文献[14]在面积时间上都具有很大的超越, 实现了轻量级的格密码硬件实现, 但是在频率上还可以进一步优化。总的来说, 如何在满足性能和资源上的不同需求条件下, 选择最适宜的格密码方案硬件实现方案是目前学术界亟待解决的问题。

1.3 本文主要内容和章节安排

目前已提出的格密码硬件结构在性能上存在硬件资源、速度和功耗不均衡, 兼容性和安全性低等问题, 迫切需要在核心多项式乘法模块进行大量深入细致的基础研究。本文在 Kyber 第一轮参数q=7681的基础上对第二轮和第三轮的q =3329 的 NTT 算法提出了不同的硬件设计, 一是面积成本和执行时间折中的迭代型 NTT 设计, 二是高性能的多路延时转接的流水线 NTT 设计, 并且针对于迭代型 NTT 模块设计了一种整体 MLWE 硬件实现结构。

本文其他内容安排如下: 第 2 节介绍了格密码算法理论基础, 分析了新参数的多项式乘法的改变; 第 3 节详细介绍了本文提出的基于 MLWE 的格密码公钥加密方案的硬件结构; 第 4 节给出了本文提出的两种不同的硬件实现方案的实验数据分析和对比;

第 5 节总结了本论文的主要工作。

2 格密码算法理论基础

目前, 基于 MLWE 问题的后量子密码算法以 CRYSTALS-Kyber 最具有代表性, 该方案对大多数应用来说, 具有很好的综合性能。本文也将在 CRYSTALS-Kyber 算法理论基础上对 MLWE 问题进行硬件上实现的研究。

2.1 MLWE 公钥加密方案

MLWE 问题集成了 LWE 问题的矩阵运算和 RLWE 的多项式乘法运算, 在介绍基于 MLWE 的公钥加密方案之前, 先简要说明一下 MLWE 问题的定义。本文涉及的数学符号有 Z_q 表示为模为 q 的整数环, 在该数域上的数据大小不会超过模 q, 多项式环域用 $R_q = Z_q[x]/(x^n+1)$ 表示, n 为多项式的最高次项。单个多项式不作特别标注, 向量多项式粗体小写字母标注, 矩阵多项式粗体大写字母标注。U 表示均匀分布, β_η 表示中心二项分布, k 表示公钥矩阵的维数, 用 $x \leftarrow \beta_\eta$ 表示根据中心二项分布采样得到的数据。那么一组 MLWE 样本可表示如下:

$$(\boldsymbol{A}, b = \boldsymbol{A}^T\boldsymbol{s} + e) \in R_q^{k\times k} \times R_q \tag{1}$$

其中$\boldsymbol{s} \leftarrow \beta_\eta(R_q^k), \boldsymbol{A} \leftarrow U(R_q^{k\times k}), e \leftarrow \beta_\eta(R_q)$

基于 MLWE 问题构建的公钥加密方案不仅可以具有 LWE 问题的安全等级也有 RLWE 问题的高效率, MLWE 采用 RLWE 中的环多项式替换 LWE 问题矩阵中的整数环参数, 并增加了可变参数 k 控制公钥多项式矩阵的大小, 在 Kyber 中, 改变 k 是将安全性扩展到不同级别的主要机制。

分析 NIST 官方公布的 CRYSTALS-Kyber 的密码方案, 可以得到简化版的基于 MLWE 的格密码公钥加密方案, 其中涉及到的加密和解密过程如算法 1 所示:

算法 1 基于 MLWE 的公钥加密方案

输入: 均匀分布的公钥$\boldsymbol{A} \leftarrow U$和$\boldsymbol{t}$; 二项分布的私钥$\boldsymbol{s} \leftarrow \beta_\eta^k$; 输入信息$m \leftarrow \{0,1\}^{256}$

输出: 密文$\boldsymbol{u}, v$; 解密信息m'

1. 采样错误项$\boldsymbol{e_1}, \boldsymbol{r} \leftarrow \beta_\eta^k$; $e_2 \leftarrow \beta_\eta$
2. 编码输入信息 $\bar{m} = encode(m) = m[i]\frac{q-1}{2}:0;$
3. 计算密文$\boldsymbol{u} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{r} + \boldsymbol{e_1}$, $v = \boldsymbol{t}^{\mathrm{T}}\boldsymbol{r} + e_2 + \bar{m}$;
4. 计算解密中的密文信息$c = v - \boldsymbol{s}^{\mathrm{T}}\boldsymbol{u}$;
5. 解密信息$m' = decode(\bar{m})$

$$= m'[i] > \frac{(q-1)}{4} \&\& \, m'[i] < \frac{3(q-1)}{4} ? 1 : 0;$$

本文所采用的是轻量级 Kyber512 的参数, 它的安全等级相当于 n=512 的 RLWE 方案。由于所有误差项的相加也是一个多项式向量, 因此误差多项式的乘积在解密过程中不断增加, 为了保证正确的解密率, Kyber 降低了噪声参数 η, 并在 NIST 官方文件中给出了详细的证明[3]。在基于 MLWE 的公钥加密方案中, 涉及的最复杂的最占用硬件资源与计算时间的就是多项式向量和矩阵运算, 如当 k=2 时计算密文的具体表达式如下:

$$\begin{aligned}\boldsymbol{u} &= \boldsymbol{A}^{\mathrm{T}}\boldsymbol{r} + \boldsymbol{e_1} \\ &= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \begin{bmatrix} e_{11} \\ e_{12} \end{bmatrix} \\ &= \begin{bmatrix} a_{11}r_1 + a_{21}r_2 + e_{11} \\ a_{12}r_1 + a_{22}r_2 + e_{12} \end{bmatrix}\end{aligned} \tag{2}$$

因此, 要实现高效的 MLWE 公钥加密方案必须在多项式乘法上进行优化设计。

2.2 环多项式乘法算法分析

对于 MLWE 密码算法, 所有多项式都是在 $R_q = Z_q[x]/(x^n+1)$ 条件下, 所以在这样的约束情况下, 多项式乘法公式计算结果要约减去超过最高次项 n 之后的系数, 具体计算公式为

$$\begin{aligned} ab &= \left[\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} a_i b_j x^{i+j}\right] mod < x^n + 1 > \\ &= \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} (-1)^{\left[\frac{i+j}{n}\right]} a_i b_j x^{(i+j)^{mod\, n}} \end{aligned} \tag{3}$$

公式法计算多项式乘法的算法时间复杂度为 $O(n^2)$, 且在硬件实现时如果采用单个乘法单元实现虽然占用资源较少, 但运算耗时较长[16], 如果采用并行结构, 节省了运算时间相应的消耗的资源又太多[17], 所以在 Kyber 公布的官方文件中并没有使用传统的公式法计算方式, 而是采用了 NTT 变换算法, 但是对 NTT 的具体操作流程进行了新的定义。

在 Kyber 的 NIST 第一轮公布的标准中素数模 7681 是满足$q - 1 = 2^8 \times 30 = 2^9 \times 15$是有 256 次和 512 次原根的, 定义$\psi$为旋转因子的模平方根 $\psi^2 mod\ q = \omega$ 以通过常规的 NTT 算法去加速多项式乘法计算, 时间复杂度仅为 $O(n\log n)$, NTT 正变换和逆变换定义如下:

NTT 正变换:

$$X_m = \sum_{k=0}^{n-1} x_k \psi^{(2m+1)k} = \sum_{k=0}^{n-1} (x_k \psi^k) w^{mk} mod\ q \tag{4}$$

NTT 逆变换:

$$x_k = \frac{1}{n}\sum_{m=0}^{n-1} X_m \psi^{-(2m+1)k} = \psi^{-k} \cdot \frac{1}{n}\sum_{m=0}^{n-1} X_m w^{-mk} mod\ q \quad (5)$$

如果采用该计算公式并不会简化多项式乘法的计算, 通过旋转因子的消去引理和折半引理可以将原来的 n 点序列进行奇数偶数拆分, 得到 $n/2$ 的序列两个新序列, 再依次对新序列进行分解, 知道分解成两点之间直接的运算, 称之为蝶形运算, 以 8 点的 NTT 蝶形运算为例, 可将上述运算过程用 3 级每级 4 个蝶形运算单元表示, 如图 2 所示。

图 2 表示的是基于频率抽取的(Decimation-in-Frequency, DIF) NTT 的蝶形运算单元, 该类型的 NTT 计算不需要对输入序列进行重排序, 只需要在第一级进行前处理后, 先计算加减单元, 再对减法单元乘上相应次方的旋转因子, 采用 DIF NTT 算法在硬件实现上既可以采用迭代型 NTT 结构也可以利用 DIF NTT 蝶形运算单元数据地址的规律采用 MDC 流水型 NTT 结构。

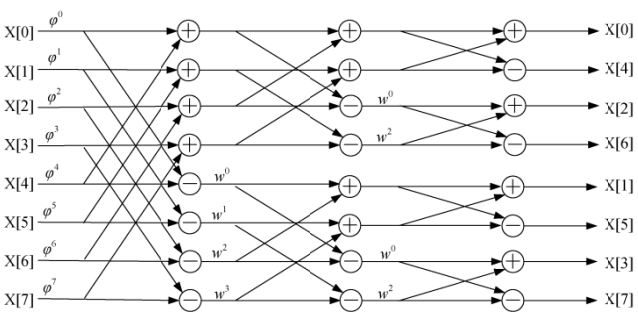

图 2 8 点 DIF 的 NTT 蝶形图

Figure 2 Butterfly diagram for *n*=8 DIF-NTT

第三轮 Kyber 算法中, 将模 q 的值减小到 3329, 该素数只有 256 次原根而不存在 512 次原根, 所以不可以直接采用公式(4)和(5)计算, 参考 NIST 的官方文件, 在环多项式域上的$X^{256}+1$可以分解为 128 个平方项多项式乘积, 分解表达式如下:

$$X^{256}+1 = \prod_{i=0}^{127}\left(X^2 - \psi^{2i+1}\right) \quad (6)$$

用 128 个 1 次多项式向量重新定义 q=3329 时的 NTT 变换,

$$\mathrm{NTT}(f) = \hat{f} = \left(\hat{f}_0 + \hat{f}_1 X, \hat{f}_2 + \hat{f}_3 X, \cdots, \hat{f}_{254} + \hat{f}_{255} X\right)$$

其中常数项和一次项系数计算表达式为:

$$\hat{f}_{2i} = \sum_{j=0}^{127} f_{2j}\psi^{(2i+1)j} = \sum_{j=0}^{127} f_{2j}\psi^{j}\psi^{2ij} \quad (7)$$

$$\hat{f}_{2i+1} = \sum_{j=0}^{127} f_{2j+1}\psi^{(2i+1)j} = \sum_{j=0}^{127} f_{2j+1}\psi^{j}\psi^{2ij} \quad (8)$$

可以观察到新型 NTT 变换中奇数项和偶数项计算是符合常规的 128 点的 NTT 变换, 将$\psi = 17$为前处理的常数, ψ^2=289为新的旋转因子, 将 256 点的 NTT 变换拆分为常规的 2 个 128 点 NTT 变换, 在得到 NTT 变换后的数据后, 对相应的多项式经过 NTT 变换后的 2 个 256 个系数不再是点对点相乘, 再进行 NTT 逆变换, 这种特殊的逐点乘法(Point-Wise Multiplication, PWM)是通过下列计算得到的:

$$\hat{h}_{2i} + \hat{h}_{2i+1}X = \left(\hat{f}_{2i} + \hat{f}_{2i+1}X\right)\left(\hat{g}_{2i} + \hat{g}_{2i+1}X\right) mod\left(X^2 - \psi^{2i+1}\right) \quad (9)$$

化简该计算式可得:

$$\hat{h}_{2i} = \hat{f}_{2i}\hat{g}_{2i} + \hat{f}_{2i+1}\hat{g}_{2i+1}\cdot\psi^{2i+1} \quad (10)$$

$$\hat{h}_{2i+1} = \hat{f}_{2i}\hat{g}_{2i+1} + \hat{f}_{2i+1}\hat{g}_{2i} \quad (11)$$

为了在硬件实现时减少乘法单元的消耗, 利用偶数项的乘法结果可以将总共的乘法的次数从 5 次减少到 4 次:

$$\hat{h}_{2i+1} = \left(\hat{f}_{2i} + \hat{f}_{2i+1}\right)\left(\hat{g}_{2i} + \hat{g}_{2i+1}\right) - \left(\hat{f}_{2i}\hat{g}_{2i} + \hat{f}_{2i+1}\hat{g}_{2i+1}\right) \quad (12)$$

所以当模 q 参数更改为 3329 时, 对于基于 NTT 的环多项式乘法只需要对多项式的奇偶项分别进行 128 点的 NTT 正逆变换, 并对完成变换后的 NTT 域上的两个多项式进行特殊的 PWM 计算, 基于 NTT 算法的q=3329 的多项式乘法算法流程如算法 2 所示, 其中 128 点的 NTT 算法采用的是基于 DIF 的迭代型 NTT 算法[18]。

算法 2 基于 NTT 的多项式乘法算法

输入: 环域上的多项式 a 和 b

输出: 环域上多项式的乘积 c

1. $for\left(i=0; i<\frac{n}{2}; i=i+1\right) do$
2. $a_{odd}[i] = a[2i+1], a_{even}[i] = a[2i]$
3. $b_{odd}[i] = b[2i+1],\ b_{even}[i] = b[2i]$
4. $end\ for$
5. $\hat{a}_{odd} = NTT_{128}(a_{odd}), \hat{a}_{even} = NTT_{128}(a_{even})$
6. $\hat{b}_{odd} = NTT_{128}(b_{odd}), \hat{b}_{even} = NTT_{128}(b_{even})$
7. $for\left(i=0; i<\frac{n}{2}; i=i+1\right) do$
8. $s_0[i] = \hat{a}_{odd}[i] + \hat{a}_{even}[i]$
9. $s_1[i] = \hat{b}_{odd}[i] + \hat{b}_{even}[i]$
10. $m_0[i] = \hat{a}_{odd}[i] \times \hat{b}_{odd}[i]$
11. $m_1[i] = \hat{a}_{even}[i] \times \hat{b}_{even}[i]$
12. $end\ for$
13. $\hat{c}_{odd} = PWM0(m_0, m_1, s_0, s_1)$
14. $\hat{c}_{even} = PWM1\left(m_0, m_1, \psi^{2i+1}\right)$
15. $c = \left(INTT(\hat{c}_{odd}), INTT(\hat{c}_{even})\right)$

3 基于 MLWE 的格密码的硬件实现

不同于其他的基于格的密码, MLWE 中的多项

式乘法可以使用 NTT 来进行计算, 这无疑大大减少了多项式乘法中的计算时间。本设计基于 Kyber 第二轮的参数 q= 3329, 对于 MLWE 设计了不同类型的 NTT 模块, 分别为流水型 NTT 模块和迭代性 NTT 模块, 并且针对于迭代型 NTT 模块设计了一种整体 MLWE 硬件实现结构。

3.1 流水型 NTT 硬件结构设计

在 Kyber 提交给 NIST 的第二轮文档中, Kyber 的参数从 7681 缩小为 3329, 使得 Kyber 的 NTT 变化发生了很多改变。NTT 变换的点数没有变化, 依旧是 256 个点, 但是新的协议中 256 个点中的奇偶点数在完成 NTT 变换时完全分开操作, 相当于两个独立的 128 点进行 NTT 运算。对于新的参数来说, 原根为 17。

同 FFT 运算一样, NTT 变换分为时域变换和频域变换。在时域变换中, NTT 每轮计算的两个点中总会有一个需要与旋转因子进行相乘, 而在频域变换中, 每轮 NTT 运算中的一半数据在完成蝶形变换前并不需要进行乘法, 因此非常有利于进行 NTT 的流水线设计。本文采用 NTT 的频域变换, 使用基 2 多路径延迟转接(Radix 2 multi-path delay commutator, R2MDC)对 Kyber 中的 NTT 变化进行设计。

本设计采用n=128, 对于每次 NTT 变化的中奇偶项分开计算, 因此每轮 128 点的 NTT 变换需要进行 7 轮计算, 连续的两个 128 点的 NTT 计算构成一次 256 点的 NTT 计算。对于 R2MDC 结构来说, 每级流水中都包含一个基 2 的蝶形单元结构、一个模乘单元和一个路径选择器。

3.1.1 基 2 蝶形单元结构

在基 2 蝶形单元结构中每次对于输入的两个数据进行计算, 如图 3 所示, 若每次输入基 2 蝶形单元架构中的数据分别为a_i和$a_{i+\frac{n}{2}}$, 之后分别计算这两个输入数据的模加和模减。模加和模减的结构如图 3(b)和图 3(c)所示, 模加和模减架构都包含一个加法器、一个减法器、一个比较器和一个二选一选择器。对于模加器, 先计算输入数据a和b的和s, 再计算s和模q的差, 然后判断s与q的大小关系, 若s大于q则选择输出$s-q$,若s不大于q则输出s。同理, 对于模减器, 先将输入数据a和b相减, 同时对相减结果加上模q, 并且判断a是否大于b。若a大于b, 则输出$a-b$, 否则输出$a-b+q$。对于模减结果, 将会在下一个周期通入模乘单元进行计算。

3.1.2 模乘结构

同 FFT 不一样, NTT 都是在整数上进行的计算, 并且数据范围限制在素数域当中, 因此对于每次乘法都需要进行取模运算。对于 Kyber 第二轮的参数来说, 乘法器输入的数据位宽为 12, 因此为了保证计算速度, 每次乘法都需要消耗一块 DSP 单元。2021 年, 文献[15]提出了一种基于巴雷特算法的模乘算法, 该算法使用多次数据的加减法和移位操作对输入的 24 位数据取模, 该算法如算法 3 所示。

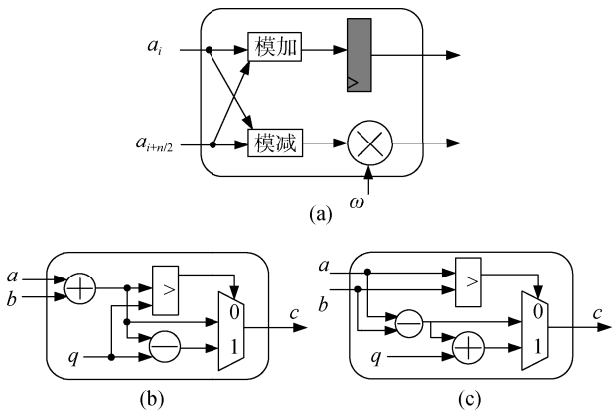

图 3 R2MDC 中蝶形单元结构

Figure 3 Butterfly unit structure in R2MDC

算法 3.修改的巴雷特约减算法[15]

输入: 不超过 24 比特的整数 prod, 模数 $q=3329$

输出: $res = prod \bmod p, quo = \lfloor \frac{prod}{q} \rfloor$

1. $quo = prod_{[23:12]} + prod_{[23:14]} - prod_{[23:18]} - prod_{[23:20]}$
2. $diff = prod_{[14:0]} - (quo + quo_{[3:0]} \ll 11 + quo_{[4:0]} \ll 10 + quo_{[6:0]} \ll 8)$
3. SWITCH（$diff_{[14:12]}$）
4. CASE 0: $q_{mux} = 0$
5. CASE 1: $q_{mux} = -q$
6. CASE 5: $q_{mux} = 3q$
7. CASE 6: $q_{mux} = 3q$
8. CASE 7: $q_{mux} = 2q$
9. DEFAULT: $q_{mux} = 0$
10. $res = diff + q_{mux}$

如图 4 所示, 本设计的模乘结构一共包含 4 级流水线。在第一级结构中, 对于输入的两个 12 位数据使用一个 DSP 单元进行相乘, 然后执行算法 1 中的第 2 步。在第二级架构中, 执行算法中的 3,4 两步, 并且将计算得到的$diff$和q_{mux}传递给下一级。第三级将$diff$和q_{mux}相加得到res, 并且将得到的结果减去模q得到res_1。最后一级中利用res_1的最高位选择输出的结果为res还是res_1。该结构一共包含有 4 级流水线, 这意味着 R2MDC 架构中每级启动时均需要等待 4 个周期, 但是对于整个架构的影响微乎其微。

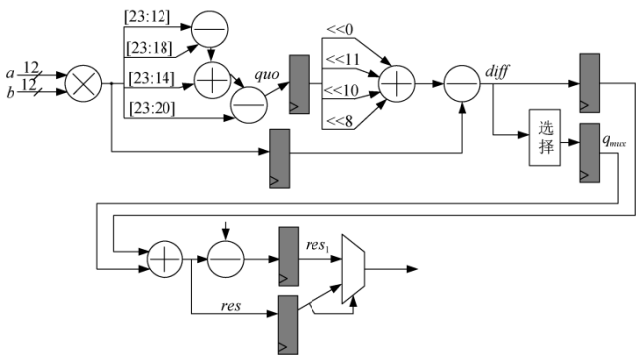

图 4　R2MDC 中模乘单元结构

Figure 4　Module Multiplication unit structure in R2MDC

3.1.3　路径选择器

对于 MDC 架构来说, NTT 在运算时并不是直接对当前来自前级的数据进行计算, 而是一段时间之后的缓存序列, 因此, 设置路径选择器的原因在于配合延时单元, 使得下一级输入至蝶形单元的两个数据正好是匹配的。当数据来到时, 路径选择器 C2 先选择第一条数据路径, 第一组数据先后各输入一半作为后级蝶形单元两条路径上的输入。与此同时, 第二条数据路径上的待处理数据通过延时单元来匹配时序。然后 C2 切换为第二条数据路径, 同样的方式输入到蝶形单元。

3.1.4　R2MDC 整体结构

图 5 展示的是本设计的 R2MDC 的整体实现架构。在计算的开始, 使用两个模乘单元对于输入的数据a_i和$a_{i+\frac{n}{2}}$分别乘以旋转因子。因为旋转因子是需要提前计算并且存储在 FPGA 中的, 但是和输入的数据一样, 位宽为 12 位。在 FPGA 中, 一块 18k 的 BRAM(Block-RAM)深度为 512, 宽度为 32 比特, 因此使用 BRAM 来实现存储会造成大量的资源浪费。在 FPGA 中, 分布式 ROM 使用 LUT 块来组成, 非常适合存储位宽不大的数据, 本设计中使用分布式 ROM 来存储数据。

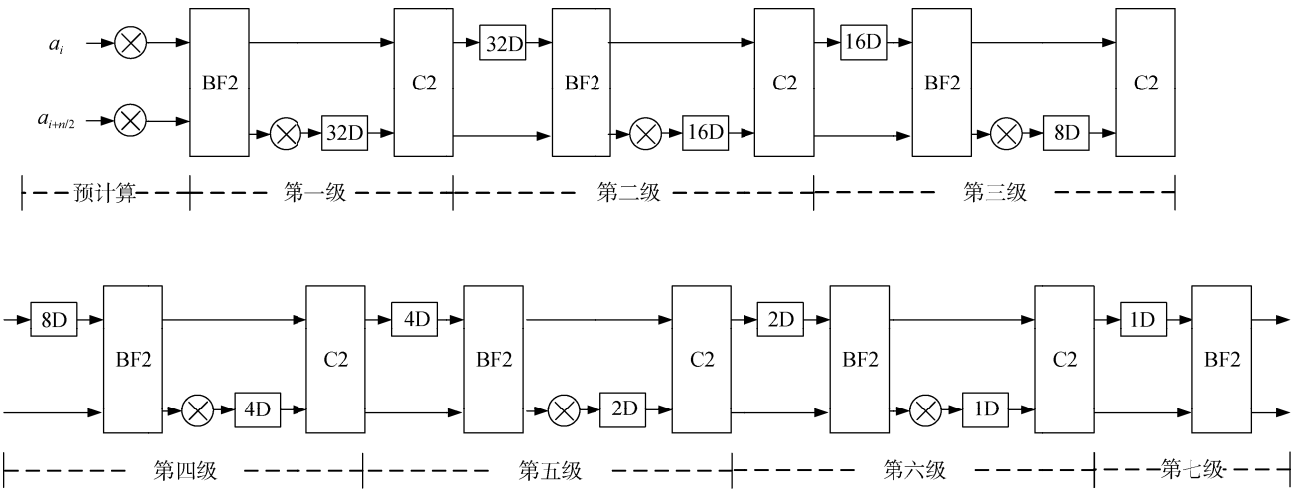

图 5　R2MDC 整体结构

Figure 5　R2MDC overall structure

当预处理结束后, 数据被推送入 MDC 架构中的第一级。对于 MDC 结构来说, 每一级的计算大抵相同, 但是当级数越高后, 每次计算的数据延时单元变少, 直至最后一级中数据需要 1 个周期的延时。并且, 对于每级架构中的预计算的旋转因子的数量也会减少。当所有的数据计算完第一次的蝶形单元模块后, 第 65~128 个点需要与旋转因子相乘, 而前 64 个点的数据不需要相乘, 因此在延时 32 个周期后, a_0和a_{31}一起进入第二级的蝶形单元进行蝶形单元计算。这样, 每一级都如此往复。

当第一组 128 个点全部完成预计算时(需要 64 个周期), 将 256 个点中的剩下 128 个点通入预计算单元中, 这样使得 256 个点完全在 R2MDC 单元中流水线运行。

在 R2MDC 架构中, 蝶形单元和乘法器的利用率由于操作时序上的原因只有 50%, 一共需要$\log_2 N-2$个复数乘法单元, $\log_2 N$个基 2 蝶形单元以及$3N/2-2$个数据延时单元。所有的延时单元采用 FPGA 上的移位寄存器来实现以减少资源的使用。

3.2　迭代型 NTT 硬件结构设计

流水型 NTT 变化会带来速度上的显著提升, 但是同样存在着占用资源过大的问题。在 R2MDC 中每一级的实现均需要一块 DSP 块, 一块 DSP 块相当于 100 个 FPGA 片, 并且 R2MDC 同样存在着运行频率

较低等问题。因此, 本文提出一种仅使用一块 DSP 块来实现的 NTT 结构。

本设计中选用频率抽取 DIF 的方式, 采用 GS(Gentleman-Sande)蝶形变换。这种蝶形变换首先进行数据的加法和减法, 然后对于减法产生的结果与旋转因子ω相乘。因此, 这种方式构建的结构一共包含一个模加器、一个模减器和一个模乘器。此外, 以上结构均使用流水线结构以提高运行频率。

前两个取模操作是在多路复用器的帮助下对模数q进行加减。而模乘器通过 DSP 块进行 12×12 位的乘法运算, 并使用巴雷特约减法将 24 位的乘积减少到 12 位。

本设计将预处理与 NTT 算法结合起来, 并利用旋转因子的模平方根ψ来减少 ROM 的存储量。的平方根ψ来减少 ROM 中的存储量。频域变换中的 NTT 需要在 NTT 计算的最后一级进行比特位翻转, 因此位翻转单元也包含在 NTT 模块中。反转单元也包括在同一个 NTT 模块中。在 NTT 计算结束时进行置换。对于解密过程中的逆 NTT(Inverse NTT), 在 NTT 域中的向量需要转化为常数域中, 然后算法只需要改变旋转因子(将ω变成ω^{-i})和后运算(第 i 个系数的乘法 最后将第 i 个系数乘以$n^{-1}\psi^{-i}$。为了这个目的, 常数因子$\psi^i(0\leqslant i<n)$, $\omega^{-i}\left(0\leqslant i<n/2\right)$和$n^{-1}\psi^{-i}$ $(0\leqslant i<n)$被存储在 ROM 中, 被配置为分布式 ROM。

3.3 基于迭代型 NTT 的 MLWE 的加密处理器

本小节中将基于上一小节提出的迭代型 NTT 来设计一种 MLWE 的硬件实现结构, 该结构选取 Kyber 第三轮的参数, 即 $q=3329$。提出的结构如图 6 所示, 一共包含一个二项分布采样、一个迭代型 NTT 模块和一个控制模块。

3.3.1 二项分布采样

本设计中没有使用离散高斯作为噪声分布, 而是使用$\eta=2$或$\eta=3$的居中二项分布。CRYSTALS-Kyber 采用二项分布是因为实现高斯采样器的方法比较复杂, 而且可能容易受到时间攻击。而二项式取样器只涉及一个步骤, 具有更好的安全性, 在 Newhope[19]的案例中已经证明了其安全性。

一个遵循二项分布的样本可以通过比较两个随机数的汉明距离序列来得到。本设计使用轻量级流密码 Trivium 来生成随机数[20]。产生随机数的数学公式如下:

$$sample=\left(a_1,\cdots,a_\eta,b_1,\cdots,b_\eta\right)\leftarrow\{0,1\}^{2\eta} \tag{13}$$

$$output=\sum_{i=0}^{\eta}a_i-\sum_{i=0}^{\eta}b_i \tag{14}$$

伪随机数发生器(Pseudo Random Number Generator, PRNG)被用来在每个时钟周期产生一个噪声样本, 因此一共需要 $256\times k$个周期来产生一个噪声矢量多项式, 然后将其发送到汉明权重序列比较器, 以获得二项式分布数据。

3.3.2 MLWE 方案总体硬件结构

MLWE 的整体实现方案包括密钥产生、加密和解密。在开始加密和解密之前, 假设公钥和私钥是在 NTT 域中产生的, 并且存储在块存储器(Block RAM, BRAM)中。加密阶段, 使用二项分布采样器产生向量多项式误差$\boldsymbol{e_1},\boldsymbol{r}$和多项式误差$\boldsymbol{e_2}$, 并且写入不同的 BRAM 中: 18Kb BRAM 配置为真双端口 RAM 模式, 深度和宽度分别为 512 和 12(在 Kyber512 的情况下)。NTT 模块每个周期从BRAM_error中读取两个系数作为蝶形单元的输入。*NTT_signal* 控制 NTT 模块的启动和结束。

$\boldsymbol{e_1}$和$\boldsymbol{r}$的系数需要进行 4 次 NTT 变换, 因此当之前的系数处理完成后, 首次读取数据的两个端口地址分别扩展为 256 和 384。BRAM_tmp 单元负责在内循环和外循环中临时存储数据。在所有的向量

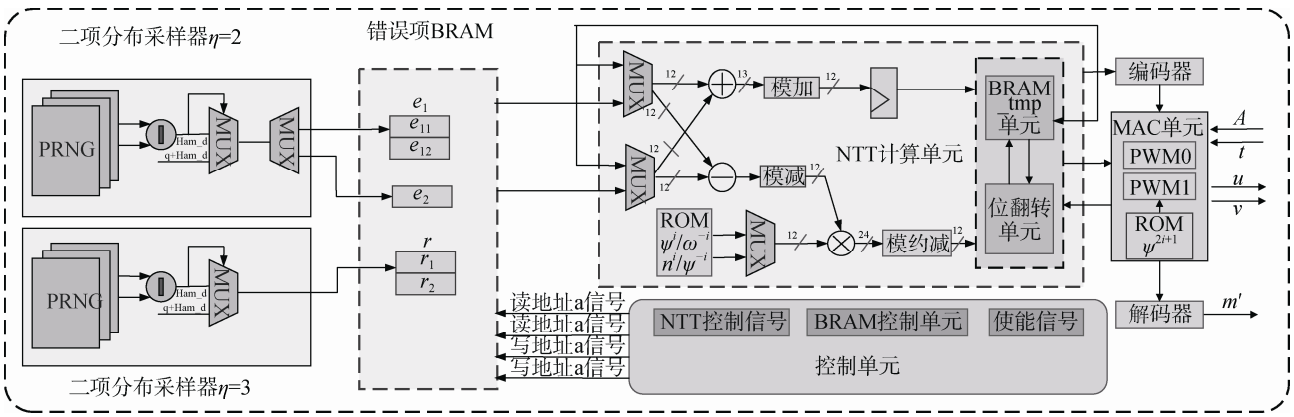

图 6 基于 MLWE 方案的硬件结构

Figure 6 The proposed structure of MLWE scheme

被转换到 NTT 域之后。乘法-累加(Multiply- Accumulate, MAC)单元执行算法 2 中的 PWM0 和 PWM1 操作。本设计利用三个并行 DSP 计算$\hat{h}_{2i}$和$\hat{h}_{2i+1}$中的乘法操作来减少运算周期, 其中ψ^{2i+1}存储在分布式 ROM 中并在加法器和乘法器之间插入流水线寄存器来提高频率和连续数据的处理速度。

NTT 域中的密文$\boldsymbol{u}$和v被传送到解密部分, 与密钥 $\boldsymbol{s}$ 参与计算得到解密中的密文 c。解密时不需要产生错误值, 并且只需要一个两次 128 点的逆 NTT 变换, 因此节省很多周期。MAC 单元将经过处理后的输入信息 m 提供给解码器。

4 实验结果分析和对比

本节将以上的设计使用 Vivado 软件进行仿真综合实现, 并且使用 FPGA 运行。在对比方面, 首先将设计的流水型 NTT 和迭代型 NTT 设计与目前最先进的 NTT 设计进行, 之后将由迭代型 NTT 构建的 MLWE 方案与其他完整加密方案实现结果进行对比。

4.1 NTT 实现对比

表 1 展示的是本文提出的流水线 NTT 结构在硬件综合实现后, 资源与时间和目前最先进的结构的对比。本文的流水型结构和对比的结果均使用参数 n=256, q=3329。表 1 以 LUT、FF、DSP、18Kb BRAM 为评估面积资源消耗、频率和周期表示硬件设计的计算速度和时间, 展示了本文提出的设计的实现结果。其中频率的单位为 MHz, 时间的单位为μs。

表 1 不同 NTT 架构 FPGA 实现资源消耗对比
Table 1 Implementation Results for Different NTT Implementation on FPGA

设计	LUT	FF	DSP	BRAM	频率	周期	时间
[13]	801	717	4	2	222	324	1.46
[21]	609	640	2	2	257	490	1.91
[22]	948	444	1	5	190	904	4.76
	2543	792	4	9	182	233	1.28
[15]	1731	1167	2	3	161	512	3.18
迭代型	**821**	**1024**	**1**	**1**	**203**	**1415**	**6.97**
流水型	**1575**	**1383**	**8**	**1**	**179**	**232**	**1.29/0.64**

表 1 中对比的两个设计和本文提出的设计均采用 Artix-7 FPGA 实现。在频率上, 本文的设计略低于文献[13]中的设计, 但在周期上, 本文的流水型 NTT 结构首次启动时间为 104 个周期, 之后的 64 个周期中输出第一组 128 个点的结果, 接着输出第二组的 128 个点的结果, 同样消耗 64 个周期, 因此一次完整的 256 点 NTT 只需要 232 个周期, 这远低于其他结构。由于周期上的优势, 本文的流水型 NTT 运行时间仅为 1.29μs, 在下一次的 NTT 变换中则仅仅需要 0.64μs, 相较于文献[13]提升 56.16%。

在资源上, 由于流水型 NTT 除了最后一级以外, 每级包含一个 DSP 块, 并且预计算包含两个 DSP 块, 一共使用了 8 个 DSP 块。流水型 NTT 在 DSP 块上明显多于其他设计, 但是在 BRAM 资源上, 由于本设计大部分使用了分布式 ROM, 进一步减少了 BRAM 资源的使用, 在 BRAM 资源上的优化也是流水型 NTT 相比于文献[21]和[22]的主要优势, 文献[22]对 NTT 结构采用了 1 个蝶形单元计算但是采用了 5 个 18Kb BRAM 寄存中间数据, 其综合出的面积是大于本文中迭代型 NTT 设计, 而当文献[22]采用 4 个蝶形单元计算时, 周期和流水型几乎一样但是在资源数量利用上远超过流水型 NTT。与文献[15]相比, 本设计使用了更少的 LUT 资源, 减少幅度达到 9.01%。与文献[13]的设计相比, 本设计使用了更多的 LUT 和 FF 资源, 但是在赛灵思 7 系列的 FPGA 中, 一块 DSP 块大致相当于 102 个 FPGA 片, 一块 36K BRAM 块大致相当于 196 个 FPGA 片[16]。因此在 DSP 和 BRAM 的资源上和本设计相差无几。对于迭代型 NTT 来说, 由于 NTT 模块只使用了单个模乘单元, 因此在 DSP 和 BRAM 的使用数量上均显著少于其他的设计, 但因迭代型 NTT 模块中蝶形操作的乘法均复用一个 DSP 模块, 所以整个执行周期比其他设计略高。

4.2 MLWE 整体方案实现对比

本文的 MLWE 密码处理器使用 Kintex-7 (KC705) FPGA 综合实现并使用 Xilinx Vivado 2018.3 实现。表 2 以 Slice/LUT/FF、DSP、BRAM 为资源消耗、频率的单位为 MHz、 Cycles 为加密解密需要的时钟周期数。为了与其他设计进行综合性能比较, 本文提供了一个面积时间乘积 (Area-Time-Product, ATP) 指标来评估执行时间和使用的 Slice 数量之间的权衡, 其计算公式为综合后的 Slice 片数量乘执行时间 μs 得来的, 而 Slice 是在 FPGA 芯片中占到比重最大的重要资源。ATP 的值越小, 设计就越高效, 该设计是一个更折中的综合性能较优的设计。“Op/s”列包含加密和解密处理器每秒可以执行的周期数, 通过周期数除频率计算得到。为了确保公平比较, 本文还选择了不同的 RLWE 公钥加密方案, 其中 n = 512 (RLWE512) 具有相同的安全级别公钥加

表 2 本文提出的 MLWE 方案与其他的基于 FPGA 的设计结果比较

Table 2 FPGA based results comparisons of our proposed MLWE scheme with other designs

设计	模块	Slice	LUT	FF	DSP	BRAM	频率	Cycles	时间	Op/s	ATP($\times 10^3$)
RLWE512 [23]	加密	1887	5595	4760	1	14	251	13769	54.86	18229	103.52
	解密	1887	5595	4760	1	14	251	8883	35.39	28256	66.78
RLWE512 [9]	加密	---	1536	953	1	3	278	13300	47.9	20902	----
	解密	---	1536	953	1	3	278	5800	20.86	47931	---
RLWE256 [24]	加密	402	1254	1046	2	2	280	35478	126.7	7892	50.93
	解密	249	722	558	2	0	290	17732	61.14	16467	15.22
MLWE512 [12]	加/解密	147	442	237	1	3	136	7179	52.92	18897	7.78
MLWE512 [13]	加/解密	3547	10502	9859	8	13	200	2446/ 3754	12.23/ 18.77	81766/53276	43.38/ 66.58
本设计	**加密**	**626**	**1871**	**1726**	**8**	**4**	**256**	**9717**	**37.96**	**26346**	**23.76**
	解密	**594**	**1775**	**1498**	**6**	**4**	**280**	**2808**	**10.03**	**99715**	**5.96**

密方案和具有相似的多项式运算复杂度的 RLWE 的方案, 其中 $n = 256$ (RLWE256)。

与实际的 RLWE 处理器[23]相比, 本设计提出的结构消耗更少的资源和更少的周期次数, 并且从 ATP 参数来看在实现效率方面具有明显的优势。以小面积成本实现的高效的 RLWE512 设计[9]优化了 NTT 模块中的内存访问方案并获得更高的频率, 但是本设计在周期和执行时间方面具有更好的性能, 从而每秒处理更多操作。此设计没有报告消耗的 Slice 数量, 因此无法获得准确的 ATP 值。基于最先进的最快的可用 Schoolbook 多项式乘法的高效 RLWE 设计[24]具有最高频率, 但是, 与 NTT 算法相比, 该设计消耗的时钟周期数依旧过多。Kyber 在 FPGA 上的处理器设计[12]仅包括针对多项式向量的优化 NTT 设计, 文献[13]是 Kyber 整体协议的实现, 而本文提出的 MLWE 加解密适用于需要高安全级别且资源受限设备。

本文提出的设计在加密时, 具有较少的 9717 个周期数和最小的 23.76 的 ATP 值, 并且在解密时需要 2808 个周期和 5.96 的 ATP。与最佳设计相比, 本文提出的单独 MLWE 设计比 RLWE512[9]具有 1.83×的 Op/s 上的提升, 与 RLWE256[24] 相比有 55.4%在 ATP 权衡上的提升。本设计以合理的面积消耗为代价, 是有良好性能的折中硬件设计。

5 结论

本文针对后量子加密中极具前景的 Kyber 加密设计了两种不同的 NTT 结构, 分别为流水型 NTT 结构和迭代型 NTT 结构, 此外基于迭代型 NTT 结构设计出一种 MLWE 的整体实现方案。实验结果表明, 本文的提出的流水型 NTT 结构具备更好的速度性能, 在速度上相较于之前的设计提升 11.64%和 59.43%。同时, 基于迭代型 NTT 的 MLWE 整体实现方案, 在使用了最少的周期和最小的 ATP 时, 其效率比最新的硬件实现高 2 倍左右。在未来的工作中, 将使用流水型 NTT 结构来搭建 Kyber 的完整协议, 并且将所提出的 MLWE 处理器扩展到整个 CRYSTALS-Kyber 算法和其他参数集。

致 谢 感谢南京航空航天大学电子信息工程学院新兴集成电路实验室的各位老师和同学在实验过程中的帮助和支持,感谢审稿专家和编辑老师的指导建议。

参考文献

[1] Miller V S. Use of Elliptic Curves In Cryptography [C]. *In Conference on the theory and application of cryptographic techniques*, 1985: 417–426.

[2] Shor P W. Algorithms for Quantum Computation: Discrete Logarithms and Factoring[C]. *The 35th Annual Symposium on Foundations of Computer Science*, 1994: 124-134.

[3] NIST. Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process. https://csrc.nist.gov/publications/detail/nistir/8309/final. July 2020.

[4] Regev O. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography[C]. *The thirty-seventh annual ACM symposium on Theory of computing - STOC '05*, 2005: 84–93.

[5] Lyubashevsky V, Peikert C, Regev O. On Ideal Lattices and Learning with Errors over Rings[C]. *Advances in Cryptology*, 2010: 1-23.

[6] Langlois A, Stehlé D. Worst-Case to Average-Case Reductions for Module Lattices[J]. *Designs*, *Codes and Cryptography*, 2015, 75(3): 565-599.

[7] D'Anvers J P, Karmakar A, Sinha Roy S, et al. Saber: Module-LWR Based Key Exchange, CPA-Secure Encryption and CCA-Secure KEM [C]. *International Conference on Cryptology in Africa,* 2018: 282–305.

[8] Göttert N, Feller T, Schneider M, et al. On the Design of Hardware Building Blocks for Modern Lattice-Based Encryption Schemes[M].

Cryptographic Hardware and Embedded Systems – CHES 2012. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012: 512-529.

[9] Roy S S, Vercauteren F, Mentens N, et al. Compact Ring-LWE Cryptoprocessor[C]. *International Workshop on Cryptographic Hardware and Embedded Systems*, 2014: 371–391.

[10] Du C H, Bai G Q, Wu X J. High-Speed Polynomial Multiplier Architecture for Ring-LWE Based Public Key Cryptosystems[C]. *2016 International Great Lakes Symposium on VLSI*, 2016: 9-14.

[11] Feng X, Li S G, Xu S F. RLWE-Oriented High-Speed Polynomial Multiplier Utilizing Multi-Lane Stockham NTT Algorithm[J]. *IEEE Transactions on Circuits and Systems II*: *Express Briefs*, 2020, 67(3): 556-559.

[12] Chen Z H, Ma Y, Chen T, et al. Towards Efficient Kyber on FPGAs: A Processor for Vector of Polynomials[C]. *2020 25th Asia and South Pacific Design Automation Conference*, 2020: 247–252.

[13] Bisheh-Niasar M, Azarderakhsh R, Mozaffari-Kermani M. High-Speed NTT-Based Polynomial Multiplication Accelerator for CRYSTALS-Kyber Post-Quantum Cryptography[EB/OL]. 2021

[14] Huang Y M, Huang M Q, Lei Z K, et al. A Pure Hardware Implementation of CRYSTALS-KYBER PQC Algorithm through Resource Reuse[J]. *IEICE Electronics Express*, 2020, 17(17): 20200234.

[15] Xing Y F, Li S G. A Compact Hardware Implementation of CCA-Secure Key Exchange Mechanism CRYSTALS-KYBER on FPGA[J]. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2021: 328-356.

[16] Liu W Q, Fan S L, Khalid A, et al. Optimized Schoolbook Polynomial Multiplication for Compact Lattice-Based Cryptography on FPGA[J]. *IEEE Transactions on Very Large Scale Integration* (*VLSI*) *Systems*, 2019, 27(10): 2459-2463.

[17] Sinha Roy S, Basso A. High-Speed Instruction-Set Coprocessor for Lattice-Based Key Encapsulation Mechanism: Saber In Hardware[J]. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2020: 443-466.

[18] Yao K, Kundi D E S, Wang C H, et al. Towards CRYSTALS- Kyber: A M-LWE Cryptoprocessor with Area-Time Trade-off[C]. *2021 IEEE International Symposium on Circuits and Systems*, 2021: 1-5.

[19] Alkim E, Ducas L, Pöppelmann T, et al. Post-Quantum Key Exchange - a New Hope [J]. *USENIX Security,* 2016: 327-343.

[20] Xing Y F, Li S G. An Efficient Implementation of the NewHope Key Exchange on FPGAs[J]. *IEEE Transactions on Circuits and Systems I*: *Regular Papers*, 2020, 67(3): 866-878.

[21] Zhang C, Liu D S, Liu X J, et al. Towards Efficient Hardware Implementation of NTT for Kyber on FPGAs[C]. *2021 IEEE International Symposium on Circuits and Systems*, 2021: 1-5.

[22] Yarman F, Mert A C, Öztürk E, et al. A Hardware Accelerator for Polynomial Multiplication Operation of CRYSTALS-KYBER PQC Scheme[C]. *2021 Design*, *Automation & Test in Europe Conference & Exhibition*, 2021: 1020-1025.

[23] Pöppelmann T, Güneysu T. Towards Practical Lattice-Based Public-Key Encryption on Reconfigurable Hardware[C]. *Selected Areas in Cryptography*, 2014: 68-85.

[24] Zhang Y Q, Wang C H, Kundi D E S, et al. An Efficient and Parallel R-LWE Cryptoprocessor[J]. *IEEE Transactions on Circuits and Systems II*: *Express Briefs*, 2020, 67(5): 886-890.

崔益军 于 2020 年在南京航空航天大学获得博士学位。现任南京航空航天大学电子信息工程学院讲师。研究领域为硬件安全、物理不可克隆函数。Email: yijun.cui@nuaa.edu.cn

倪子颖 于 2021 年在南京航空航天大学电子与通信工程专业获得硕士学位。现任南京航空航天大学科研助理。研究领域为硬件安全、后量子密码。Email: nzy@nuaa.edu.cn

刘伟强 于 2012 年在英国贝尔法斯特女王大学获得电子工程博士学位。现在南京航空航天大学电子信息工程学院教授、副院长, 国家自然科学基金优秀青年基金获得者。研究领域为信息安全芯片、近似计算芯片、AI 硬件加速等。Email: liuweiqiang@nuaa.edu.cn

姚衍 于 2019 年在华侨大学信息工程专业获得学士学位。现在南京航空航天大学电路与系统专业攻读硕士学位。研究领域为后量子的硬件实现。 Email: yaokan0416 @nuaa.edu.cn

王成华 现任南京航空航天大学电子信息工程学院教授, 国家“万人计划”教学名师。研究领域为信息安全芯片、物理不可克隆函数。Email: chwang@nuaa.edu.cn