应用于格密码的可重构多通道数论变换硬件设计

刘冬生 赵文定* 刘子龙 张 聪 刘星杰

(华中科技大学光学与电子信息学院 武汉 430074)

摘 要：针对不同格密码体制带来的数论变换参数多样性，以及数论变换的性能优化设计，该文提出一种基于随机存取存储器(RAM)的可重构多通道数论变换单元。在数论变换单元设计中，在按时间抽取的基础上改进多通道架构，并提出一种优化地址分配方法。最后基于Xilinx Artix-7现场可编程逻辑门阵列(FPGA)平台进行原型实现，结果显示，所设计的数论变换单元消耗的资源为1744 Slices, 16 DSP，完成1次多项式乘法的时间为2.01 μs (n=256), 3.57 μs(n=512), 6.71 μs(n=1024)和13.43 μs(n=2048)，支持256～2048的不同参数n和13～32 bit模q的可重构配置，工作频率最高可达232 MHz。

关键词：格密码；多项式乘法；数论变换；硬件实现

中图分类号：TN918.2; TN402 **文献标识码**：A **文章编号**：1009-5896(2022)02-0566-07

DOI: 10.11999/JEIT210114

Reconfigurable Hardware Design of Multi-lanes Number Theoretic Transform for Lattice-based Cryptography

LIU Dongsheng ZHAO Wending LIU Zilong ZHANG Cong LIU Xingjie

(*School of Optical and Electronic Information, Huazhong University of Science and Technology, Wuhan* 430074, *China*)

Abstract: The performance of number theoretic transformation in lattice-based cryptography is insufficient, and the number theoretic transformation parameters are different. A Random Access Memory (RAM)-based reconfigurable multi-lanes number theoretic transform is proposed. In the design of number theory transformation unit, the multi-lanes architecture is improved on the time decimation operation architecture, and an optimized address allocation method is proposed. The number theory transform unit is implemented on Xilinx artix-7 Field Programmable Gate Array (FPGA) platform. The results show that the resource consumed by the unit is 1744 slices and 16 DSP, and the time to complete a polynomial multiplication is 2.01 μs (n=256), 3.57 μs (n=512), 6.71 μs (n=1024) and 13.43 μs (n=2048). The unit supports reconfigurable configurations of 256～2048 parameters n and 13～32-bit modulus q, and the maximum operating frequency is 232 MHz.

Key words: Lattice-based cryptography; Polynomial multiplication; Number theoretic transform; Hardware implementation

1 引言

传统的公钥密码体制所基于的数学难题，如RSA(Rivest-Shamir-Adleman)和椭圆曲线密码(Elliptic Curve Cryptography, ECC)，早在1997年便被证明在量子计算下毫无安全性可言[1]。因此传统公钥密码体制构建的信息安全系统及各种应用将面临着严峻的安全问题，甚至是被完全破解的危险。因而能够抵御量子攻击的下一代密码方案及其实现技术，即后量子密码成为学界研究的热点。

2019年9月，谷歌宣布实现“量子霸权”：一台53量子比特的量子计算机在200 s内执行的运算任务需要经典计算机运行10000年[2]。2020年7月，美国国家标准与技术研究所宣布了后量子密码的第3轮候选方案。预计在2023年前后，后量子密码标准将被确立。同年12月，中国信息协会发布《量子安全技术白皮书(2020)》[3]，从多个角度对量子时代下的信息安全进行了阐述。在后量子密码标准的

收稿日期：2021-02-01；改回日期：2021-06-10；网络出版：2021-06-22
*通信作者： 赵文定 zhaowend@foxmail.com
基金项目：国家自然科学基金(61874163)；国家科技重大专项基金(2017ZX01032-101)；中央高校基本科研基金(HUST: 2018KFYYXJJ056)

Foundation Items: The National Natural Science Foundation of China (61874163), The National Science and Technology Major Project (2017ZX01032-101), The Fundamental Research Funds for the Central Universities (HUST: 2018KFYYXJJ056)

评选中，基于格难题的公钥密码，即格密码方案处于优势地位：在现有的7种候选方案中，CRYSTALS-KYBER[4]、数论研究单元算法(Number Theory Research Unit, NTRU)和SABER都是基于格的后量子密码方案，因此格密码有着重大的研究与应用价值。

在诸多格密码方案中，多项式生成与多项式乘法运算是通用的核心算子，占用了绝大多数的时间资源开销[5]。以基于环误差学习(Ring-Learning With Error, Ring-LWE)格难题而构造的密码方案[5]为例，多项式乘法运算占整个加解密时间的80%以上，对格密码的实现与应用有着显著影响。在现有的研究中，数论变换(Number Theoretic Transform, NTT)、快速傅里叶变换(Fast Fourier Transform, FFT)和School-book都能够用于实现多项式乘法运算[5]。在时间方面进行比较，FFT和NTT的计算复杂度均为$O(n\log n)$，而School-book多项式乘法的计算复杂度为$O(n^2)$[5,6]，随着n增大，School-book所消耗的时间会以指数上升；在运算方面进行比较，FFT涉及复数和浮点运算，NTT只在整数域中进行运算。综上所述，在格密码的多项式乘法运算中，采用NTT来计算模内多项式乘法，更易于硬件的高效实现，能够提升格密码的整体运算性能。

目前对NTT的研究主要集中在以下3个方面：

(1) 降低资源消耗与功耗：通过混合基与多通道延迟换向器，以及单口存储器合并多个内存部分，实现面积较小的NTT大整数乘法[7]。通过NTT非耦合结构，分别使用按时间抽取(Decimation In Time, DIT)与按频率抽取(Decimation In Frequency, DIF)的蝶形运算单元来实现低功耗的NTT与数论逆变换(Inverse Number Theoretical Transform, INTT)运算[8,9]。

(2) 降低运算时间、减小内部时延：通过脉动阵列，将多个蝶形运算单元串行组合起来，以快速完成连续多个NTT运算[6,10]。通过多通道技术，优化多通道地址分配方法与整体架构，以实现低时延[11,12]。

(3) 提高吞吐量：通过存储优化算法，对于单通道与多通道分别设计，以取得更高的吞吐量[13–15]。

本文从模乘算法和数论变换算法出发，研究基于随机存取存储器(Random Access Memory, RAM)的可重构模乘运算单元与多通道NTT架构。本文首先介绍了数论变换算法与多项式乘法，然后提出了基于RAM的模乘单元与多通道蝶形运算单元地址生成的硬件设计，进一步实现了支持4个不同参数n和15个不同参数q可重构NTT运算单元，最后给出了仿真验证的结果。

2　负包裹卷积与数论变换算法

在格密码方案中，NTT通过负包裹卷积算法计算多项式环$\mathbb{Z}_q[x]/\langle x^n+1\rangle$上的多项式乘法运算[5,12,14]。负包裹卷积算法。对多项式$\boldsymbol{a}$和$\boldsymbol{b}$进行多项式乘法获得多项式$\boldsymbol{c}$有5个步骤：(1) $\widehat{\boldsymbol{a}}(x)=\boldsymbol{a}(x)\cdot\omega_{2N}^{x}$；$\widehat{\boldsymbol{b}}(x)=\boldsymbol{b}(x)\cdot\omega_{2N}^{x}$；(2)$\boldsymbol{A}(x)=\mathrm{NTT}(\widehat{a}(x))$；$\boldsymbol{B}(x)=\mathrm{NTT}\left(\widehat{\boldsymbol{b}}(x)\right)$；(3) $\boldsymbol{C}(x)=(\boldsymbol{A}(x)\odot\boldsymbol{B}(x))$；(4) $\widehat{\boldsymbol{c}}(x)=\mathrm{INTT}(\boldsymbol{C}(x))$；(5) $\boldsymbol{c}(x)=\widehat{\boldsymbol{c}}(x)\cdot\omega_{2N}^{-x}$；其中$\odot$表示点乘。NTT运算如式(1)所示

$$X_k=\sum_{n=0}^{N-1}x_n\omega_N^{nk}\ (\mathrm{mod}\ q)\tag{1}$$

n点数论变换逆运算(INTT)如式(2)所示

$$x_k=N^{-1}\sum_{n=0}^{N-1}X_n\omega_N^{-nk}\ (\mathrm{mod}\ q)\tag{2}$$

g为素数q的原根

$$\omega_N^n=g^{\frac{(q-1)n}{N}}\ (\mathrm{mod}\ q)\tag{3}$$

NTT以$g^{(q-1)n/N}$取代了ω_N，所有运算均为模运算，与FFT相比，不需要进行复数或浮点数运算，因而降低了硬件电路的设计难度与资源消耗。基2 DIT-NTT算法实现见表1，采用库利-图基算法将时间复杂度从$O(n^2)$降低到$O(n\lg n)$，运算的核心为蝶形运算，而整个蝶形运算中消耗1/2以上资源的是模乘运算，模乘运算的设计与优化影响着整个负包裹卷积多项式乘法的效率。

3　NTT硬件设计

3.1　基于RAM的可重构模乘单元

蝶形运算单元(Butterfly Arithmetic Unit, BAU)是NTT运算的核心。本文设计了用于BAU的基于

表 1　基2 DIT-NTT算法

输入：$\boldsymbol{A}(x),\omega_N\in R_q,N$
输出：$\boldsymbol{A}(x)$
(1) **for** $m=1$ to $N/2$ by $m=2m$ **do**
(2) **for** $i=0$ to N–1 **do**
(3) **for** $j=0$ to N–1 **do**
(4) $u=\boldsymbol{A}[k+j]$
(5) $t=\boldsymbol{A}[k+j+m]\times\omega_N$
(6) $\boldsymbol{A}[k+j]=(u+t)\mathrm{mod}\ q$
(7) $\boldsymbol{A}[k+j+N]=(u-t)\mathrm{mod}\ q$
(8) **end for**
(9) **end for**
(10) **end for**
(11) **return** $A(x)$

RAM的模乘器(RAM-based Modular Multiplier, RMM)如图1所示。该模乘器支持最高32 bit与32 bit的乘法，并可通过可重构操作改变模q的值，以实现不同模下的模乘运算，通过以下3步实现：

(1) 更新RAM内的值：Ref_din[31:0], Ref_addr[9:0]和Ref_ramchoose[1:0]用于为RMM中的两个双端口RAM提供数据与控制信号以进行更新。通过将乘法器MUL的积截取后作为地址信号输入RAM，能够直接得到RAM中存入的计算机预先计算好的求模结果。

(2) 改变积的截取方式：根据Ref_q_choosed [3:0]信号，IO_Ctrl模块分解MUL_out[63:0]，将其按位传送给RAMs和MOD_q(对输入的数据进行判断是否)。例如，当q为32 bit时，MUL_out [63:0]将被分解为{[63:56], [55:48], [47:40], [39:32], [31:0]}；当q为14位时，信号被分解为{8'd0, {2d'0, [27:22]}, [24:14], [13:0]}。

(3) 采用位截取模加法器：ADD0, ADD1, ADD2和ADD3对输入的信号进行求和，并将结果分为两路，一路减去模q、一路直接输出，根据减法运算中的进位信号选择一路，最后根据Ref_q_choosed [3:0]信号对数据按照模的位数截取、高位置0，输出模加结果。

为了达到高性能与资源开销的平衡，根据模q的最大位数，选择RAM的数量，以实现资源与性能的适配。因而本设计采用两个双口RAM实现，如图1所示。如使用1个双口RAM构成模运算器，则不需要模加法器，但需要消耗$32\times2\times2^{16}$位的内存容量。因而本设计采用4个模加法器，消耗内存容量为$32\times4\times2^{8}$位，显著减少了内存消耗，做到了较好的平衡。

根据所提出RMM结构可以实现任意q的模运算，随着q位的增加，RAM消耗的资源将呈指数级增长。相比而言，针对特定参数优化的模乘器在资源消耗方面具有明显的优势，但不适用于多参数可重构设计。

3.2 多通道蝶形运算地址生成的硬件设计

位反转(bit-reversed)能够简化DIT-NTT的地址生成。然而在多通道DIT-NTT的硬件实现中，地址的生成会变得十分复杂。因而本文设计了多通道蝶形运算地址架构，并采用Block RAM(BRAM)存储数据以取得更高的速度与利用效率。对于d通道DIT-NTT(长度n)，则$2d$为BRAM数量。DIT-NTT运算有3个步骤：

(1)数据应按照顺序存入BRAMs中，例如$\{0, 1, \cdots, n/d-1\}$于BRAM0, $\{n/d, n/d+1, \cdots, 2n/d-1\}$于BRAM1, $\cdots$, $\{(d-1)\times n/d, \cdots, n-1\}$于BRAM$d$–1。

(2)如图2所示，所有的BRAM被分为两部分，低位部分地址为0～$k/2$–1 ($k=n/d$)，高位地址部分为$k/2$～n–1。然后，BRAM0的一个端口和BRAM_$d/2$的端口连接到BAU0的a和b端口；BRAM0的b端口和BRAM_$d/2$的b端口连接到BAU1的a和b端口。BAU2的输入端口连接BRAM1的a端口和BRAM_$d/2$+1的a端口，依此类推。BAUd–1对应于BRAM_$d/2$–1和BRAM_d–1。

(3)每一轮结束后，输入Four_BRAMs和输出

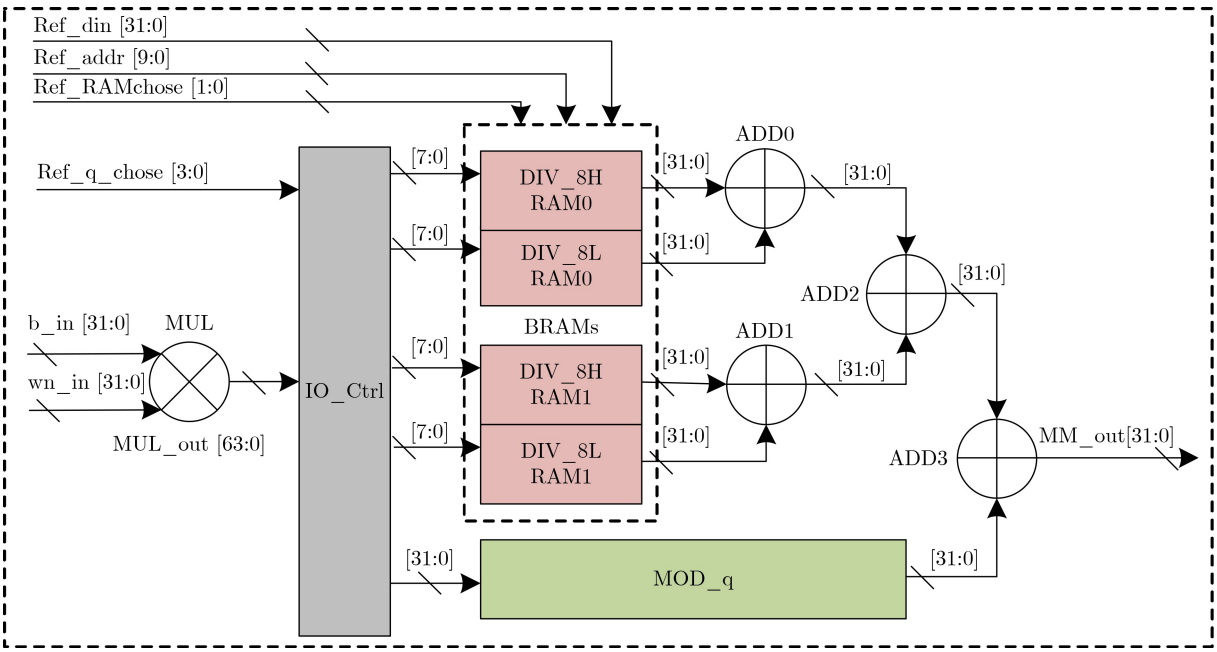

图 1 基于RAM的模乘器结构图

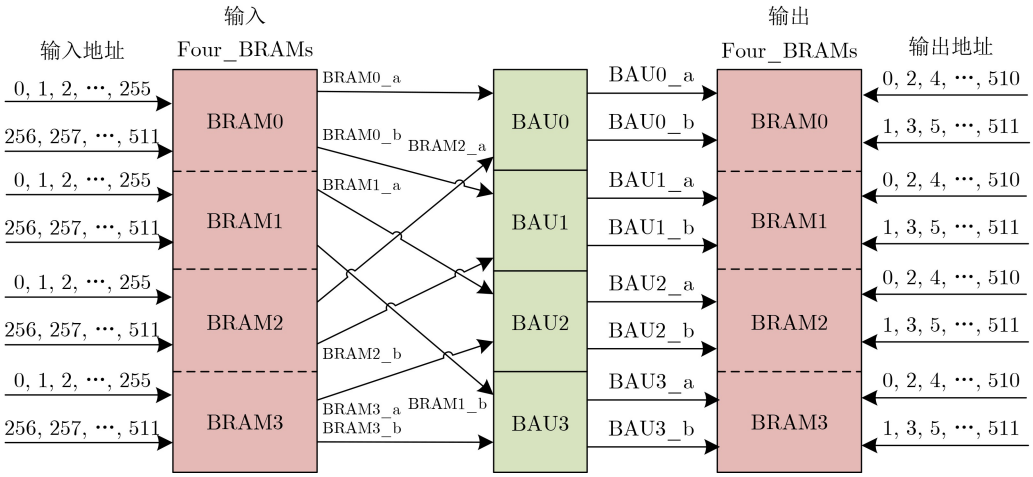

图 2 4通道NTT数据流向图

Four_BRAMs进行交换。在所有回合结束后，NTT结果将被存储到BRAM中，并以地址顺序递增的方式存储。

根据上述步骤，数据的存储顺序与正常数论变换的结果相同，解决了多通道NTT运算过程中，地址生成变复杂的问题，使得DIT在地址生成上与DIF同样简单。优化数据地址分配方法的代价是旋转因子的排列变复杂。NTT和INTT的旋转因子都需要按照位反转规则进行排列。例如，当n=8,4通道时，经典DIT-NTT的排列旋转因子为$\{\omega_2^0,\omega_4^0,\omega_4^1,\omega_8^0,\omega_8^1,\omega_8^2,\omega_8^3,0\}$，将其变为$\{0,\omega_2^0,\omega_4^0,\omega_4^1,\omega_8^0,\omega_8^2,\omega_8^1,\omega_8^3\}$，以便于地址的生成和正确的计算。旋转因子可通过在计算机上按特定规律生成，因此在硬件上不会产生额外的过资源开销。

3.3 可重构NTT结构

高性能、可重构的多通道NTT结构如图3所示。BAUs包括RMM、位截取模加法器和位截取模减法器以及用于执行NTT蝶形运算的寄存器阵列。Ctrl包括状态机和一些控制信号的生成和反馈。ADDR生成Four_BRAMs和Wn_BRAMs的地址。NTT可重构设置将在多项式乘法之前进行：

(1) RAM中的数据更新。通过Ref_din[31:0], Ref_addr[11:0]和Ref_ramchoose[3:0]，连接到Wn_BRAMs与BMM中的RAM，对其进行数据更新。Ref_ramchoose[3:0]从低到高，分别为RMM中的RAM0, RAM1与Wn_BRAM0, Wn_BRAM1的读写控制信号。

(2) 参数n与q的设置：Ref_n[1:0]和Ref_q[3:0]分别按表2的顺序设置参数n和q。n传递至状态机；模q传递至RMM、模加/减器中，参与运算或对根据模q的位数对信号取最低位。

为了获得最大的吞吐量和最少的时钟周期消耗，两个双口BRAM分别进行读写操作来减少NTT操作所消耗的时钟周期至1/2。初始数据存储在Four_BRAMs_0中。多项式乘法包含以下6个步骤(n=2048)：

(1) Four_BRAMs_0将多项式$\boldsymbol{a}$和$\boldsymbol{b}$分别存储在2048低位地址和2048高位地址。

(2)执行NTT($\boldsymbol{a}$)。NTT($\boldsymbol{a}$)的结果存储在Four_BRAMs_1的低位2048地址中。在这个过程中，Four_BRAMs_0和Four_BRAMs_1都被用来存储中间变量。

(3) 执行NTT($\boldsymbol{b}$)。NTT($\boldsymbol{b}$)的结果存储在4个Four_BRAMs_1的高2048地址中。

(4) Four_BRAMs_1输出NTT($\boldsymbol{a}$)和NTT($\boldsymbol{b}$)到BAU中计算的模内点乘(NTT($\boldsymbol{a}$)与Wn_in相连，NTT($\boldsymbol{b}$)与b_in相连，如图1所示)。结果NTT($\boldsymbol{a}$)$\odot$NTT($\boldsymbol{b}$)将被存储在Four_BRAMs_0的2048低位地址中。

(5) 执行INTT(NTT($\boldsymbol{a}$)$\odot$NTT($\boldsymbol{b}$))。结果$\boldsymbol{c}$将存储在Four_BRAMs_1中。

(6) 执行$\mathbf{inv}\odot\boldsymbol{c}$运算。结果$\boldsymbol{c}$将存储在Four_BRAMs_0中。

对于不同的n和q，q不会改变存储最终结果的BRAM。当n=256或1024时，结果将存储在Four_BRAMs_0中；当n=512或2048时，结果将存储在Four_ BRAMs_1中。所有的模q可以在不需要改变硬件结构的前提下改变，只需要更新相应BRAM中存储的数据即可。本设计可以从13 bit到32 bit实现任意15种模q参数的可重构。该设计在不同参数下具有相同的速度和资源消耗。为了满足最大n和q下NTT运算的需要，当n和q为其他的值时会有一部分存储空间与数据线路的高位被闲置。

当q需要更改时，BRAM中的所有数据都需要

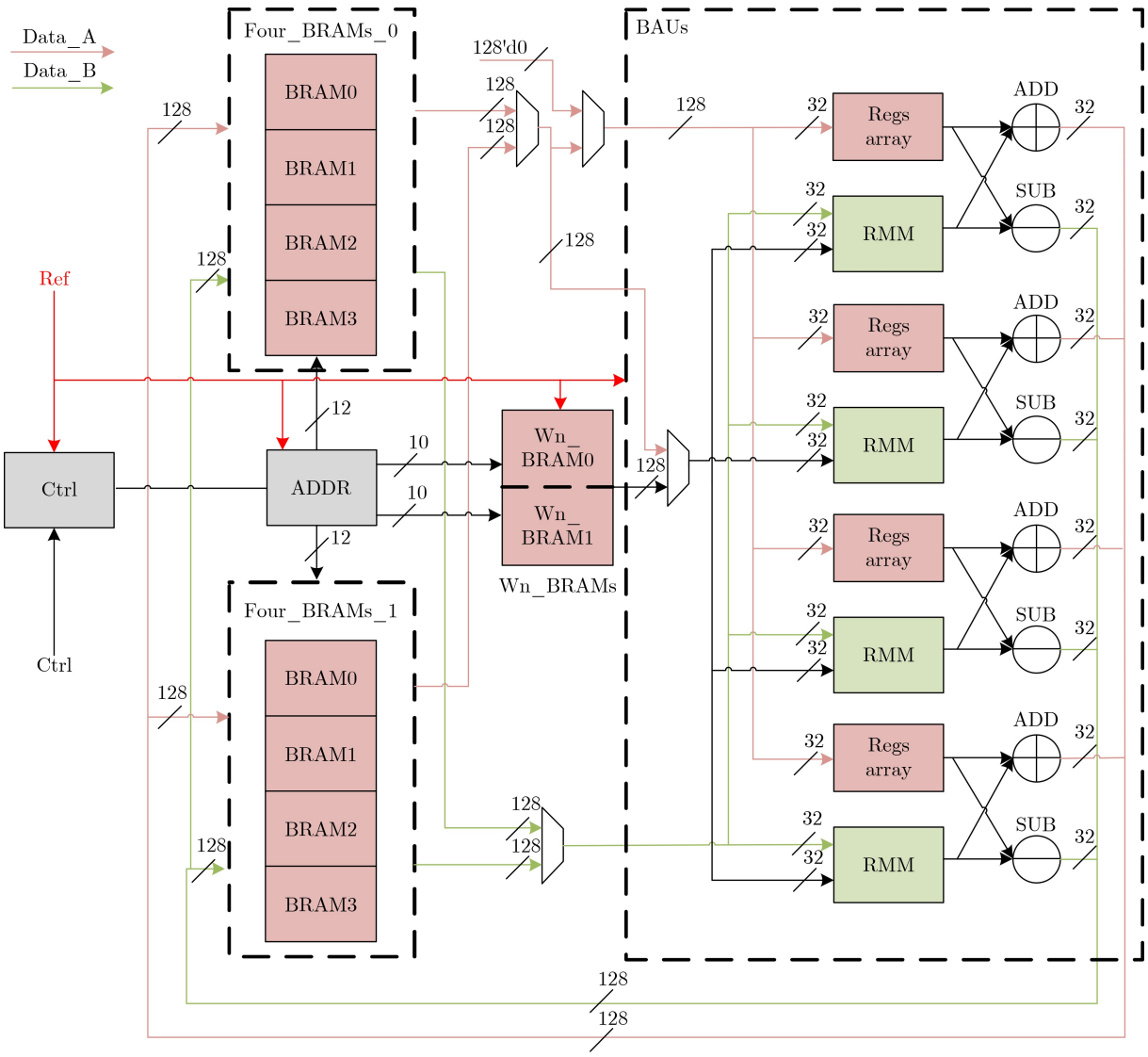

图 3 可重构NTT结构图

表 2 可重构参数表

	n							
	256	512	1024	2048	–	–	–	–
模q	7681	12289	40961	65537	786433	5767169	7640033	23068673
	104857601	16772161	469762049	998244353	1004535809	1998585857	2013265921	–

更新以实现新的NTT操作。从理论上讲，该设计可以推广到任意参数n和q的可重构NTT的实现，所消耗的资源仅与最大的n和q有关。与Feng等人[11]提出的多通道Stockham结构相比，本设计具有参数灵活、动态可配置的优点。除此之外，本设计采用DIT-NTT结构，合并了负包裹卷积引入的参数，减少了1次多项式的点乘运算。因此本设计有很强的可拓展性与研究价值。

4 实现结果及对比

本文在Xilinx Artix-7(xca35tftg256)FPGA上实现了可重构多通道NTT设计。对于不同的参数，该设计消耗相同的资源(4780 Luts,1744 Slices,16 DSPs和24 BRAMs)，通过更新设置以及模乘器中的数据实现可重构操作。每个BAU包含6个32 bit模加法器、2个32 bit模减法器和1个32 bit×32 bit乘法器。BAU平均消耗资源为454 Slices。

多项式乘法的实现结果如表3所示。通过对传统NTT的优化，在单通道BAU下，$1.5n\lg n+2n$时钟周期是极限值。本设计与NTT[5], NTT[6], NTT[11], FFT[12]进行了比较。与17 bit NTT的流水线结构[6]相比，该设计实现了32 bit NTT运算与可

表 3　多项式乘法结果比较表

文献	n	q位数	LUTs	Slices	Memory(36 kB)	DSPs	MHz	周期数	时延(μs)	ATP[1] ($\times10^5$)
NTT[5]	1024	30	2317	997	11.5 BRAMs	4	194	21405	110.34	1.100
NTT[5]	2048	57	3846	1310	22.5 BRAMs	16	161	45453	282.32	3.698
NTT[6]	2048	17	2323	–	287820 Bits	22	228.99	10289	44.93	1.965[2]
NTT[11]	512	23	–	18 K	2.5 BRAMs	128	233.1	412	1.77	0.448[3]
FFT[12]	2048	22	–	4406	50 BRAMs	12	208.12	17402	83.615	3.684
本设计	256	32	4780	1744	24 BRAMs/655360 Bits	16	232	1627	7.00	0.122
	512							2798	12.03	0.210
	1024							5251	22.58	0.394
	2048							10455	44.96	0.784

[1]ATP是通过将消耗的Slices乘以时延来计算的。

[2]NTT[3]由于未提供消耗的Slices数量，因此ATP的计算采用LUTs×Time，而本设计对应的ATP为2.149。

[3]为了合理地比较，多通道NTT[7]的ATP计算公式为$(1.5n\times9+2n)\ /\ (n\times9+2n)\ \times1.77\times18\text{k}=0.448\times10^5$。

重构操作。由于模q的数据位数对资源消耗有显著影响，在ATP接近时，本设计的设计效率较NTT[6]而言更高。

与采用FFT结构[12]进行多项式乘法相比，该设计在时间和资源消耗方面具有明显的优势。本设计采用4通道，理论需要$(1.5n\lg n+2n)/4$个时钟周期。值得注意的是，多通道NTT[11]的$(n\lg n+2n)/d$ (d=16)时钟周期是由NTT(常数多项式$\boldsymbol{a}$)的结果预存来实现的。这只能用于带误差的环学习(Ring-LWE)应用，在基于格的后量子加密方案中并不具备普适性。在n=512的情况下，16通道NTT设计[11]消耗了约本设计10倍的资源，延迟时间为$(1.5n\times9+2n)/(n\times9+2n)\ \times1.77\ =\ 2.49$ μs。因此，所提4通道NTT结构具有第2小的ATP。根据实现结果，在n=1024时(时钟周期为4.3 ns)，1次NTT操作的延迟为6.71 μs(256为2.01 μs，512为3.57 μs，2048为13.43 μs)。

5　结论

为了解决NTT时延长与应用于不同加密环境的问题，本文提出一种可重构多通道NTT硬件设计。通过对多通道蝶形运算进行改进，设计了多通道NTT架构与基于RAM的可重构蝶形运算单元，以实现简单高效的可重构NTT运算，并在Xilinx Artix-7 FPGA平台上进行了验证。本设计的最大工作频率为232 MHz，能在6.71 μs内完成多项式长度为1024的NTT运算，并拥有第2小的ATP，具备很高的研究与应用价值。

参 考 文 献

[1] SHOR P W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer[J]. *SIAM Journal on Computing*, 1997, 26(5): 1484–1509. doi: 10.1137/S0097539795293172.

[2] ARUTE F, ARYA K, BABBUSH R, *et al*. Quantum supremacy using a programmable superconducting processor[J]. *Nature*, 2019, 574(7779): 505–510. doi: 10.1038/s41586-019-1666-5.

[3] 赵勇, 戚巍, 徐兵杰, 等. 量子安全技术白皮书(2020)[R]. 2020.

[4] CHEN Zhaohui, MA Yuan, CHEN Tianyu, *et al*. Towards efficient kyber on FPGAs: A processor for vector of polynomials[C]. The 2020 25th Asia and South Pacific Design Automation Conference, Beijing, China, 2020. doi: 10.1109/ASP-DAC47756.2020.9045459.

[5] PÖPPELMANN T and GÜNEYSU T. Towards efficient arithmetic for lattice-based cryptography on reconfigurable hardware[C]. The 2nd International Conference on Cryptology and Information Security in Latin America, Santiago, Chile, 2012. doi: 10.1007/978-3-642-33481-8_8.

[6] RENTERÍA-MEJÍA C P and VELASCO-MEDINA J. Hardware design of an NTT-Based polynomial multiplier[C]. The 2014 IX Southern Conference on Programmable Logic, Buenos Aires, Argentina, 2014: 1–5. doi: 10.1109/SPL.2014.7002209.

[7] YE J H and SHIEH M D. High-performance NTT Architecture for large integer multiplication[C]. 2018 International Symposium on VLSI Design, Automation and Test, Hsinchu, China, 2018: 1–4. doi: 10.1109/VLSI-DAT.2018.8373254.

[8] ZHANG Neng, QIN Qiao, YUAN Hang, *et al*. NTTU: An area-efficient low-power NTT-uncoupled architecture for NTT-based multiplication[J]. *IEEE Transactions on Computers*, 2020, 69(4): 520–533. doi: 10.1109/TC.2019.2958334.

[9] AYSU A, PATTERSON C, and SCHAUMONT P. Low-cost and area-efficient FPGA implementations of lattice-based cryptography[C]. Proceedings of 2013 IEEE

International Symposium on Hardware-Oriented Security and Trust, Austin, USA, 2013.

[10] RENTERÍA-MEJÍA C R and VELASCO-MEDINA J. Lattice-based cryptoprocessor for CCA-Secure identity-based encryption[J]. *IEEE Transactions on Circuits and Systems I*: *Regular Papers*, 2020, 67(7): 2331–2344. doi: 10.1109/TCSI.2020.2981089.

[11] FENG Xiang, LI Shuguo, and XU Sufen. 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. doi: 10.1109/TCSII.2019.2917621.

[12] CHEN D D, MENTENS N, VERCAUTEREN F, *et al*. High-speed polynomial multiplication architecture for ring-LWE and SHE cryptosystems[J]. *IEEE Transactions on Circuits and Systems I*: *Regular Papers*, 2015, 62(1): 157–166. doi: 10.1109/TCSI.2014.2350431.

[13] MERT A C, KARABULUT E, OZTURK E, *et al*. An extensive study of flexible design methods for the number theoretic transform[J/OL]. *IEEE Transactions on Computers*, 2020, 1–15. doi: 10.1109/TC.2020.3017930.

[14] LIU Dongsheng, ZHANG Cong, LIN Hui, *et al*. A resource-efficient and side-channel secure hardware implementation of ring-LWE cryptographic processor[J]. *IEEE Transactions on Circuits and Systems I*: *Regular Papers*, 2019, 66(4): 1474–1483. doi: 10.1109/TCSI.2018.2883966.

[15] KIM S, LEE K, CHO W, *et al*. Hardware architecture of a number theoretic transform for a bootstrappable RNS-based homomorphic encryption scheme[C]. The 2020 IEEE 28th Annual International Symposium on Field-Programmable Custom Computing Machines, Fayetteville, USA, 2020, 56–64. doi: 10.1109/FCCM48280.2020.00017.

刘冬生：男，1979年生，博士，教授，研究方向为能效无线传输技术及芯片设计、后量子密码算法及密码芯片设计.

赵文定：男，1997年生，硕士生，研究方向为数论变换、格密码.

刘子龙：男，1990年生，博士，研究方向为硬件安全.

张　聪：男，1994年生，博士生，研究方向为后量子密码.

刘星杰：男，1997年生，硕士生，研究方向为后量子密码硬件实现.

责任编辑：余　蓉