CRYSTALS–Kyber NTT 양자회로 구현

https://youtu.be/N6-Q4q-Qko4

IT융합공학부 송경주

NTT quantum circuit for CRYSTALS-Kyber

- CRYSTAL-Kyber: lattice의 Module-LWE의 어려움을 기반으로 한 KEM.

- CRYSTALS-Kyber 다항식 곱셈 = $Z_q[x]/(x^n + 1)$
  → 구현된 CRYDTALS-Kyber NTT 양자회로 : $Z_{3329}[x]/(x^{256} + 1)$ 상에서의 곱셈

- n-bit 다항식 a와 b에 대한 곱셈 계산 복잡도
  - School-book multiplication : $O(n^2)$ computational complexity
  - NTT multiplication : $O(n\log n)$ computational complexity

Related work

[CRYSTALS–Kyber]

- CRYSTALS–Kyber is an IND–CCA2–secure KEM with the hardness of Modul–LWE on a lattices.
- The Kyber cipher, designed to be robust in the post–quantum era, is one of the finalists of the post–quantum cryptography project conducted by NIST.
- Security is based on the hardness of resolving learning–with–errors (LWE) problems for module lattices.

Related work

[Number theoretic transform(NTT)]

- Discrete fourier transform (DFT) performs transformation on finite N complex number fields instead of continuous interval $(-\infty, \infty)$ of FT.

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi i k n / N}$$

- The Number Theory Transform (NTT) is a generalization of the discrete Fourier transf orm (DFT) domain to integer fields.
- It uses the n−th primitive root of unity based on a quotient ring instead of the complex field of DFT.
- When performing multiplication on two n−bit length polynomials,
  - School−book multiplication : $\mathbf{O}(\mathbf{n}^2)$ computational complexity
  - NTT multiplication : $\mathbf{O}(\mathbf{n}\log\mathbf{n})$ computational complexity

NTT quantum circuit for CRYSTALS-Kyber

- CRYSTALS-Kyber NTT

CRYSTALS-Kyber NTT parameter : $Z_{3329}[X]/(X^{256}+1)$

Quantum circuit is as follow :
NTT quantum circuit = NTT sub ∘ Montgomery reduce ∘ fmul

- $\boldsymbol{fmul}$: It multiplies the NTT input and the $\boldsymbol{zetas}$ value. In detail, it is divided into $\mathbf{fmul_1}$ and $\mathbf{fmul_2}$.
- **Montgomery reduce** : It performs montgomery reduce multiplication.
- **NTT sub** : It performs addition and subtraction for Montgomery reduction result and input.

NTT quantum circuit for CRYSTALS-Kyber

- In NTT quantum circuit
  - Negative numbers: Represented in qubits using two's complement.
  - NTT quantum circuits are performed using Montgomery reduction.
  - $\mathbf{32 \times n}$ qubits are used to store the coefficients of CRYSTALS−Kyber.
  - The original input must be used by the last NTT subfunction, so the function proceeds while holding the input.

NTT quantum circuit for CRYSTALS-Kyber

- **fmul**
  - The inner operation of the **fmul** function is to multiply input and **zeta**.
  - Since **zetas** is a fixed constant, the number of qubits is reduced by performing input addition equal to the **zetas** size without assigning a value to the qubit.
  - The **fmul** function is different in the way it operates in the first NTT cycle and other cycles(C).

  - $\mathbf{fmul_1}$

    The $\mathbf{fmul_1}$ function operates when C = 1.
  - $\mathbf{fmul_2}$

    The $\mathbf{fmul_2}$ function operates when C ≥ 2.

NTT quantum circuit for CRYSTALS-Kyber

- **fmul**
  - Since the input must retain its original value, the function result is stored in 32−qubit temp.
  - Both $fmul$ functions use CNOT gates to store input values in te mp and perform multiplication.
  - In the $fmul$ function, the sign of the input and zeta is checked.
  - If the sign is the same, the result is positive, and if the sign is diff erent, the result is negative.
  - As a result, the value of $(input \times zetas)$ is stored in temp qubit.

Algorithm 2 $fmul$ multiplication for $C \geq 2$

Data: *zeta* , *r*, *check*(1-qubit)

check ← CNOT(*r*[*length*(*r*)-1], *check*)
for (i=0 **to** $length(r) - 1$) : *check* ← CNOT(*r*[*length*(*r*)-1], *check*)

if $zeta \neq 1$ **then**
  if $zeta \geq 0$ **then**
    X(*check*)
    if *check*=*1* **then**
      for (i=0 **to** $-zeta + 1$) : **Dagger**: *temp* ← **add**(*r*, *temp*)
    end
    X(*check*)
    if *check*=*1* **then**
      for (i=0 **to** $-zeta - 1$) : *temp* ← **add**(*r*, *temp*)
    end
  end
else
  X(*check*)
  if *check*=*1* **then**
    for (i=0 **to** $-zeta - 1$) : *temp* ← **add**(*r*, *temp*)
  end
  X(*check*)
  if *check*=*1* **then**
    for (i=0 **to** $zeta + 1$) : *temp* **Dagger**: ← **add**(*r*, *temp*)
  end
end
return *temp*

NTT quantum circuit for CRYSTALS-Kyber

Algorithm 1 *fmul* multiplication for $C = 1$

Input: $zeta$, r

```
for i=0 to length(r) do
    temp[i] ← CNOT(r[i], temp[i])
end for

if zeta ≠ 1 then
    if zeta < 0 and input < 0 then
        for i=0 to −zeta + 1 do
            Dagger : temp ←add(r, temp)
        end for
    end if

    if zeta < 0 and input > 0 then
        for i=0 to −zeta − 1 do
            temp ←add(r, temp)
        end for
    end if

    if zeta ≥ 0 and input > 0 then
        for i=0 to zeta − 1 do
            temp ←add(r, temp)
        end for
    end if

    if zeta ≥ 0 and input < 0 then
        for i=0 to zeta + 1 do
            Dagger : temp ←add(r, temp)
        end for
    end if
end if

return temp
```

NTT quantum circuit for CRYSTALS-Kyber

- **Montgomery reduce**

  This function performs montgomery reduction multiplication on the **input×zeta**.

- Algorithm 3 shows the operation of the Montgomery reduce quantum circuit.
- In the for loop, Q is q = 3329 and $QINV$ is the inverse of Q mod R (= 216) : −3327.
- Since Q and $QINV$ are known values, the result of multiplyi ng by the corresponding size is obtained without allocating qubits to store the values of Q and $QINV$.
- Finally, index values [0] to [15] are discarded through 16-bit left shift and the values of indexes [16] to [31] are returned.

Algorithm 3 Montgomery reduce

Input: $a, temp_1, temp_2$

for i=0 to $-QINV$ **do**
  Dagger: $tmp_1[0:16] \leftarrow$ **add**$(a[0:16], tmp_1[0:16])$
end for

for i=0 to Q **do**
  $tmp_2[0:32] \leftarrow$ **add**$(tmp_1[0:32], tmp_2[0:32])$
end for

Dagger: $a[0:32] \leftarrow$ **add**$(temp_2[0:32], a[0:32])$

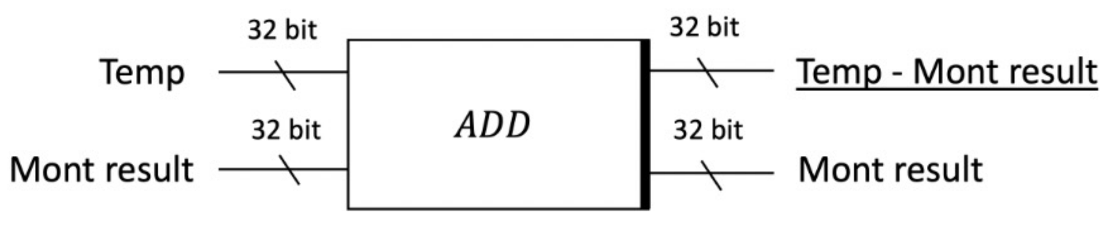
return $a[16:32]$

NTT quantum circuit for CRYSTALS-Kyber

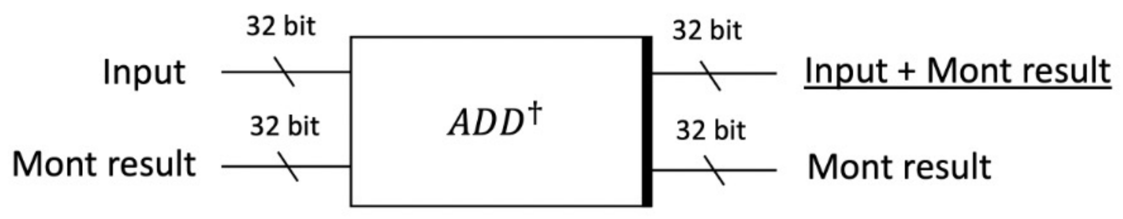
- **NTT sub**
  - The NTT sub function performs addition and subtraction between the montgomery reduce result and the input having the corresponding index, and operates as $\mathbf{NTT_{sub1}}$ and $\mathbf{NTT_{sub2}}$ in detail.

$$\boxed{NTT_{sub1} = input - Montgomery\ result}$$

  - In order to sequentially calculate the formula, both the original input and Montgomery result must be maintained after $\mathrm{NTT_{sub1}}$.
  - Since it is not possible to keep all of the calculation targets (input, Montgomery reduce result), the input is stored in temp qubits and the calculation is performed.
  - $\mathbf{NTT_{sub1}}$ stores the subtraction of temp and Mont result in temp, and Mont result is maintained.



(1) NTT_{sub1}



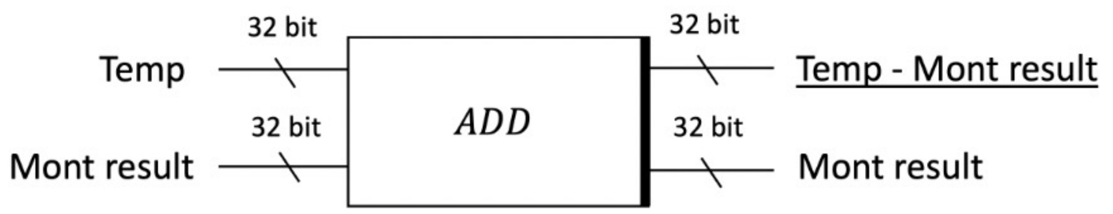
(2) NTT_{sub2}

NTT quantum circuit for CRYSTALS-Kyber

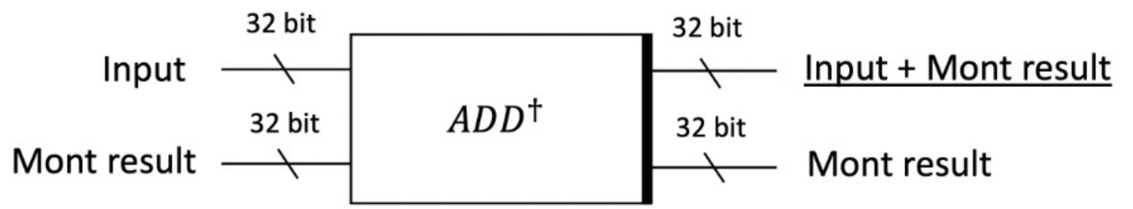
- **NTT sub**
  - $\mathbf{NTT_{sub2}}$ stores the addition of input and Mont result in input.

$$\boxed{NTT_{sub2} = input + Montgomery\ result}$$

  - The results of all operations are sorted according to the NTT array index order.



(1) NTT_{sub1}



(2) NTT_{sub2}

Evaluation

- Development environment: ProjectQ, a quantum programming tool provided by IBM
- Quantum resource estimation : ResourceSimulation provided by ProjectQ

- The NTT quantum circuit operates with three main functions.
- Each function performs an operation as much as a cycle(C).

1. $\mathbf{fmul_1}$ and $\mathbf{fmul_2}$ perform multiplication on input and zeta.
   - The difference between the two functions is that $\mathbf{fmul_2}$ uses more quantum resources than $\mathbf{fmul_1}$ because it has to determine the input sign expressed in two's complement.
   - In $\mathbf{fmul_2}$, the multi−controlled gate is used to determine the operation according to the sign of the input.
2. The Montgomery reduce function uses the most quantum resources because it multiplies large numbers.
3. Since the NTT sub function is a simple addition and subtraction operation for 32−bit qubits, it operates with the least amount of quantum resources.

Table 1: Quantum resource for NTT function. (CCCNOT : 3 qubit multi-controlled gate)

Function	C	Quantum gates				Depth
		CCCNOT	Toffoli	CNOT	X	
$fmul_1$	128	-	48,576	97,943	1	146,488
$fmul_2$	768	97,024	195,564	33	2	292,592
Mont reduce	896	-	306,270	639,184	-	945,438
NTT sub	896	-	124	318	-	379

Thank you :-)

E-mail : thdrudwn98@gamil.com