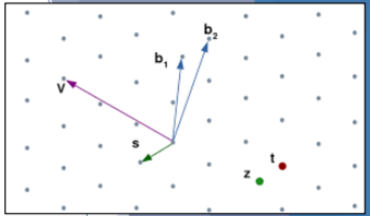
06

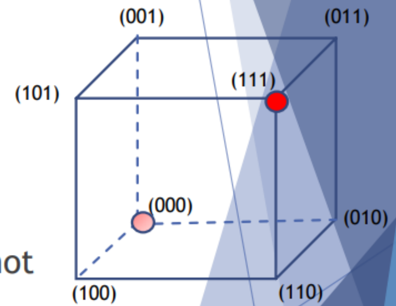
PQC 최적화 II

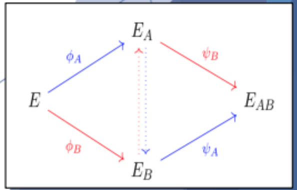
아이소 지니 기반 양자 내성 암호

The KEMs

- The finalists **Kyber, NTRU, SABER** are based on structured lattices
  - *NIST expects to select at most one for standardization*
- **Classic McEliece**, the other finalist, is based on codes
- The alternates **NTRUprime** and **FrodoKEM** are based on lattices
  - **NTRUprime** uses structured lattices, while **FrodoKEM** does not
- The alternates **BIKE** and **HQC** are based on structured codes
- The final alternate **SIKE** is based on isogenies of elliptic curves



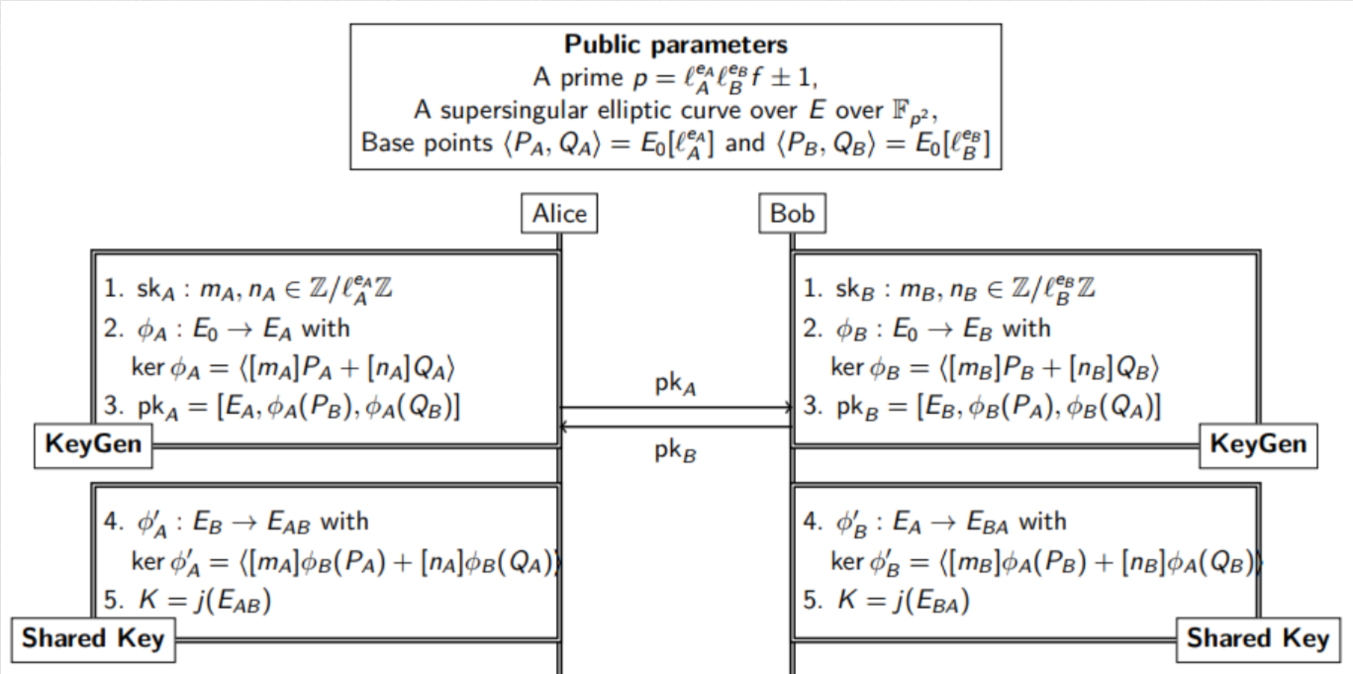




아이소 지니 기반 양자 내성 암호

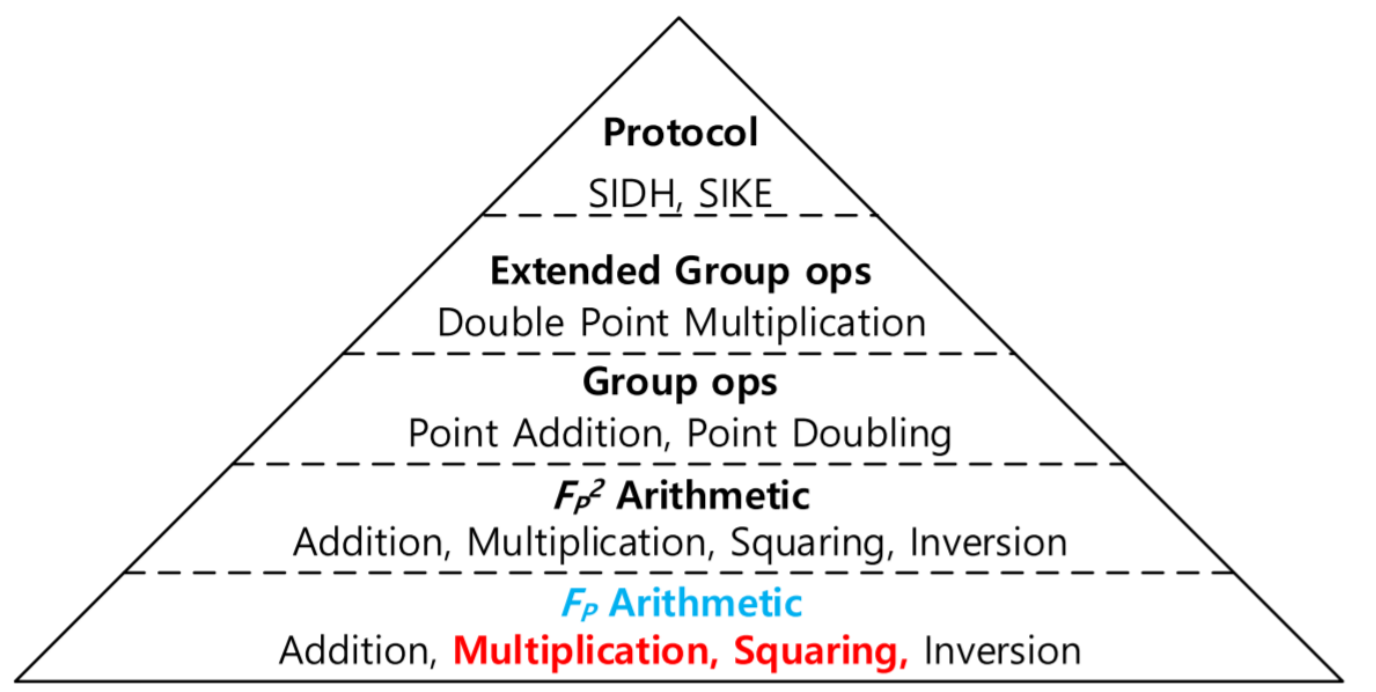
Supersingular Isogeny Diffie–Hellman (SIDH)

- 양자 내성 암호 중 키크기가 가장 작음 (TLS 바로 적용 가능)
- 연산 속도는 양자내성암호 중 가장 느림



아이소 지니 기반 양자 내성 암호 – 핵심 연산자

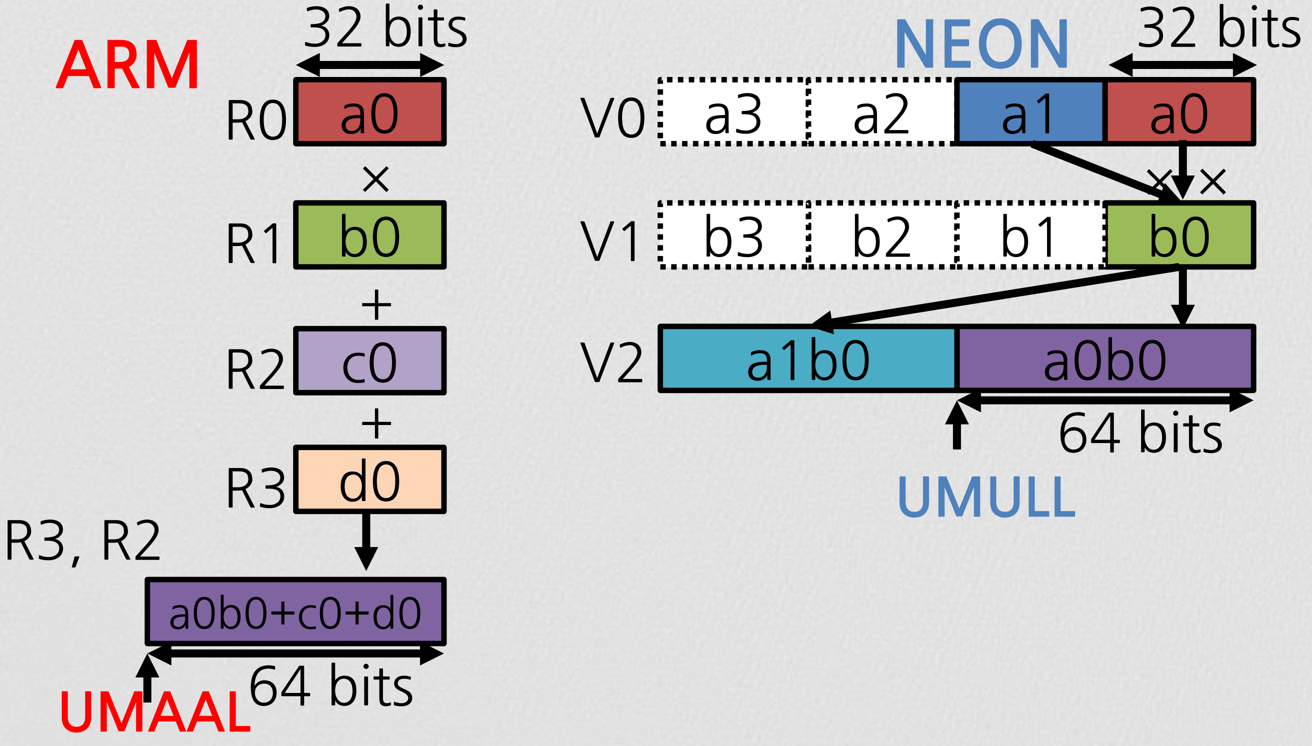
- 최하위 유한체 연산이 전체 시스템의 성능 결정



아이소 지니 기반 양자 내성 암호

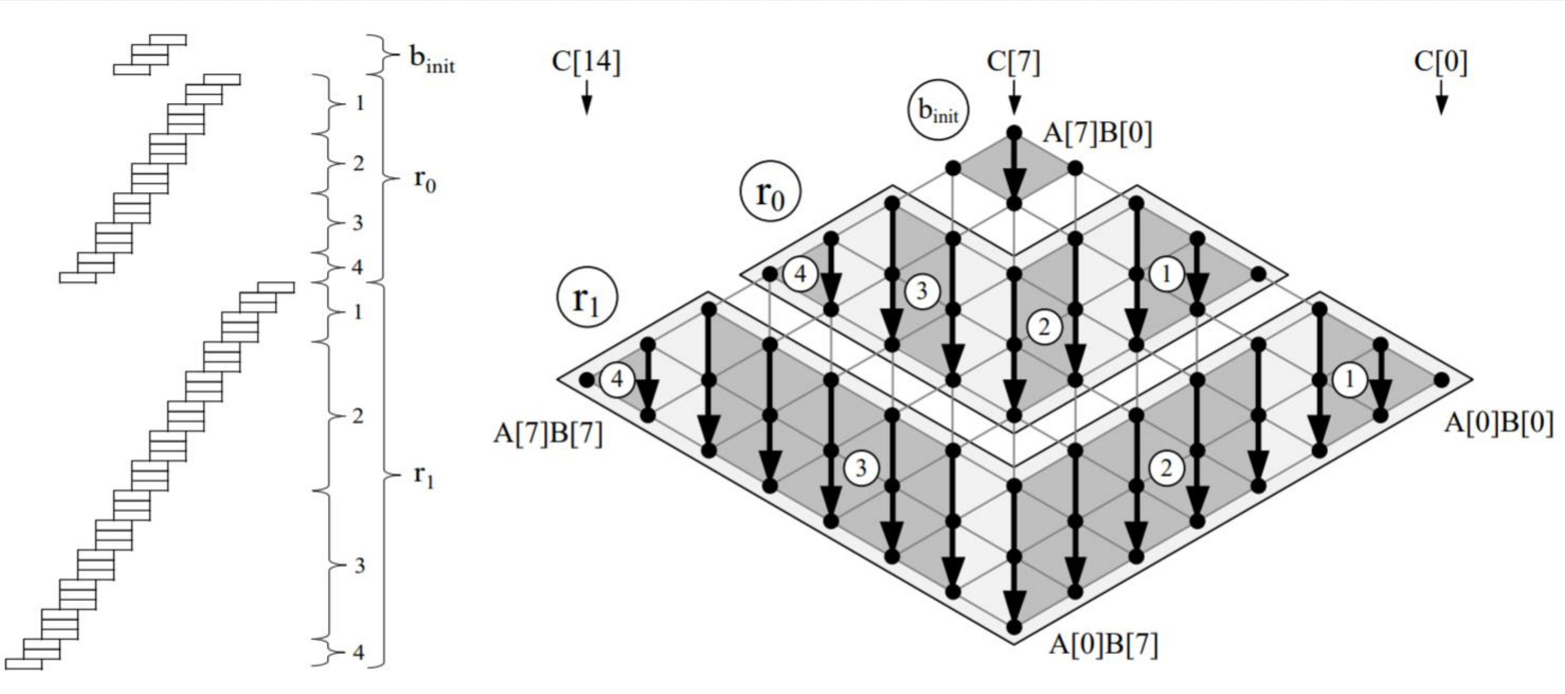
Supersingular Isogeny Diffie-Hellman (SIDH)

- 모듈러 곱셈이 SIDH 프로토콜 상에서 가장 연산 부하가 높음



곱셈기 최적화 (Operand Caching)

- Operand에 대한 메모리 접근 횟수를 연산 순서를 바꾸어서 최적화 시킨 기법



곱셈기 최적화 (Operand Caching w/ UMAAL)

- UMAAL 명령어 셋을 통해 Column-wise 곱셈 연산 최적화 가능

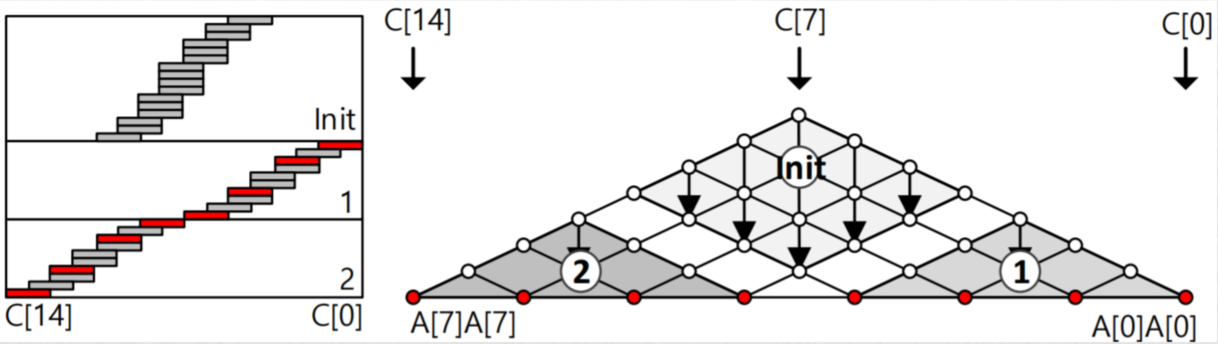
```
@ k = 6
@ r5 and r4 hold R_6, R_7 respectively
@ r6, r7, r8 hold A[3], A[4] and A[5] respectively
@ r9, r10, r11 hold B[3], B[1], B[2] respectively
MOV   r12, #0
MOV   r3,  #0
UMLAL r5,  r12, r8, r10 @ A5 B1
ADDS  r4,  r4,  r12
ADC   r3,  r3,  #0
MOV   r14, #0
UMLAL r5,  r14, r7, r11 @ A4 B2
ADDS  r4,  r4,  r14
ADC   r3,  r3,  #0
MOV   r12, #0
UMLAL r5,  r12, r6, r9 @ A3 B3
ADDS  r4,  r4,  r12
ADC   r3,  r3,  #0
@ r5 holds AB[6], r4 holds R_7, @ r3 holds R_8
```

```
@ k = 6
@ r3, r4, r12 and r5 hold R_6[0,1,2,3]
@ r6, r7, r8 hold A[3], A[4] and A[5] respectively
@ r9, r10, r11 hold B[3], B[1], B[2] respectively
UMAAL r3, r4,  r8, r10 @ A5 B1
UMAAL r3, r12, r7, r11 @ A4 B2
UMAAL r3, r5,  r6, r9  @ A3 B3
@ r3 holds (partially) AB[6]
@ r4, r5 and r12 hold partial products for k = 7
```

- UMLAL rHI, rLO, a, b → rHI:rLO := rHI:rLO + a * b
- UMAAL rHI, rLO, a, b → rHI:rLO := rHI + rLO + a * b
  - Carry가 발생하지 않음

제곱연산 최적화 (Operand Caching + Sliding Block Doubling)

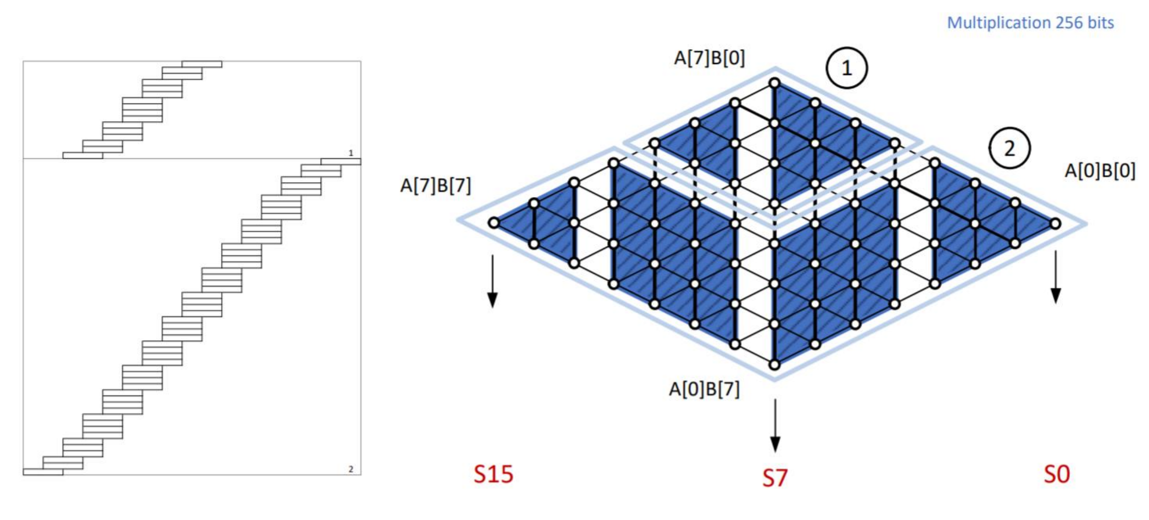
- 제곱 연산의 경우 절반 계산 후 이를 더블링하는 방법으로 수행



곱셈기 최적화 (Operand Caching w/ UMAAL & FPR)

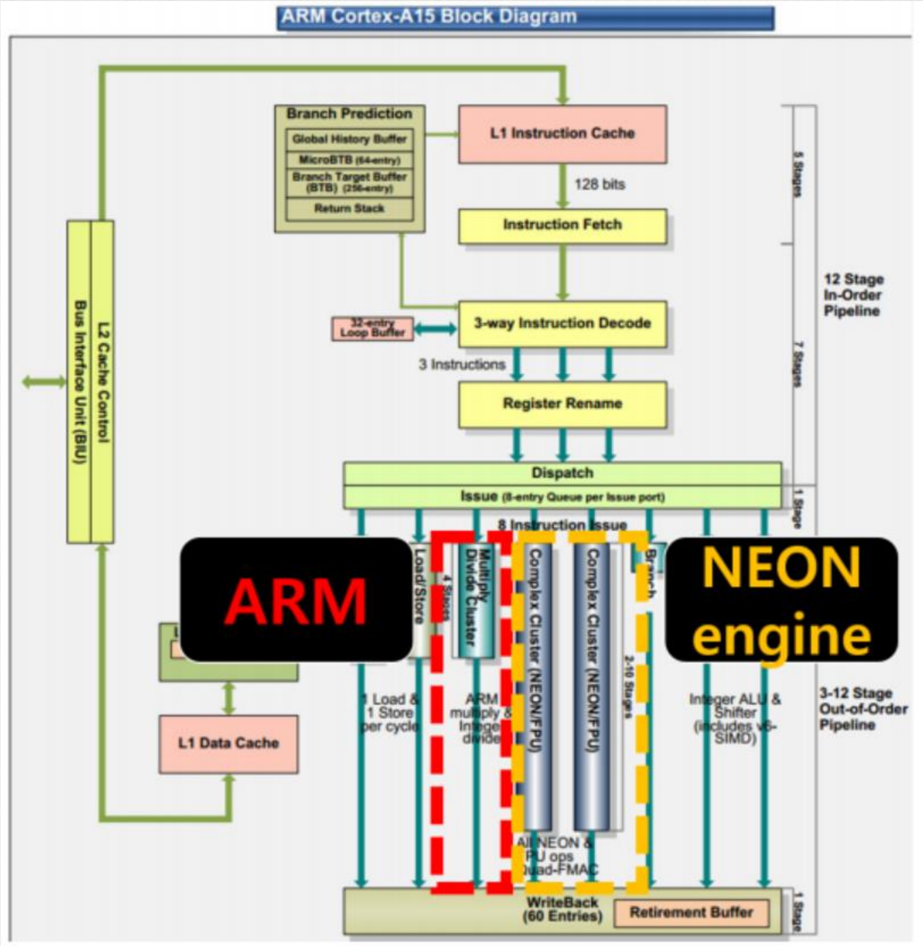
- FPR 을 Caching 공간으로 활용하여 정보에 대한 접근 속도 최적화

```
VLDM R0, {S0-S15}
VLDM R1, {S16-S31}
....
UMAAL R0, R14, R1, R6
UMAAL R0, R12, R2, R7
UMAAL R0, R11, R3, R8
VMOV R6, S6
UMAAL R0, R10, R4, R6
VMOV R6, S10
UMAAL R0, R9, R5, R6
STR R0, [SP, 4*16]
LDR R0, [SP, 4*17]
UMAAL R0, R14, R1, R7
UMAAL R0, R12, R2, R8
UMAAL R0, R11, R3, R6
VMOV R7, S7
UMAAL R0, R10, R4, R7
VMOV R7, S11
UMAAL R0, R9, R5, R7
....
```



아이소 지니 기반 양자 내성 암호 최신 ARM 프로세서

- ARM과 NEON은 독립적 모듈
- 병렬적 연산 수행 가능



아이소 지니 기반 양자 내성 암호 Karatsuba 곱셈기

2 워드 A와 B에 대한 곱셈

$$(A = A_H 2^{\frac{n}{2}} + A_L,\ B = B_H 2^{\frac{n}{2}} + B_L)$$

기본적인 방법은 4번의 곱셈 필요: O(n^2)

$$A_H B_H 2^n + A_H B_L 2^{\frac{n}{2}} + A_L B_H 2^{\frac{n}{2}} + A_L B_L$$

Karatsuba의 경우 3번의 곱셈만 필요: O($n^{\log_2 3}$)

$$A_H B_H 2^n + ((A_H + A_L)(B_H + B_L) - A_L B_L - A_H B_H) 2^{\frac{n}{2}} + A_L B_L$$

ARM **NEON** **ARM**

ARM이 NEON보다 곱셈 연산 성능이 우수

아이소 지니 기반 양자 내성 암호

Algorithm 1 Unified ARM/NEON multiplication

Input: Two m-bit operands $A = (A_H || A_L)$ and $B = (B_H || B_L)$
Output: $2m$-bit result C

$A_M \leftarrow |A_L - A_H|$ {ARM}
$B_M \leftarrow |B_L - B_H|$ {ARM}

`Interleaved section begin`

$C_L \leftarrow A_L \cdot B_L$ {ARM}
$C_M \leftarrow A_M \cdot B_M$ {NEON}
$C_H \leftarrow A_H \cdot B_H$ {ARM}

`Interleaved section end`

return $C \leftarrow C_L + (C_L + C_H - C_M) \cdot 2^{\frac{m}{2}} + C_H \cdot 2^m$ {ARM}

아이소 지니 기반 양자 내성 암호

Algorithm 1 Unified ARM/NEON multiplication

Input: Two m-bit operands $A = (A_H || A_L)$ and $B = (B_H || B_L)$

Output: $2m$-bit result C

$A_M \leftarrow |A_L - A_H|$ {ARM}

$B_M \leftarrow |B_L - B_H|$ {ARM}

Interleaved section begin

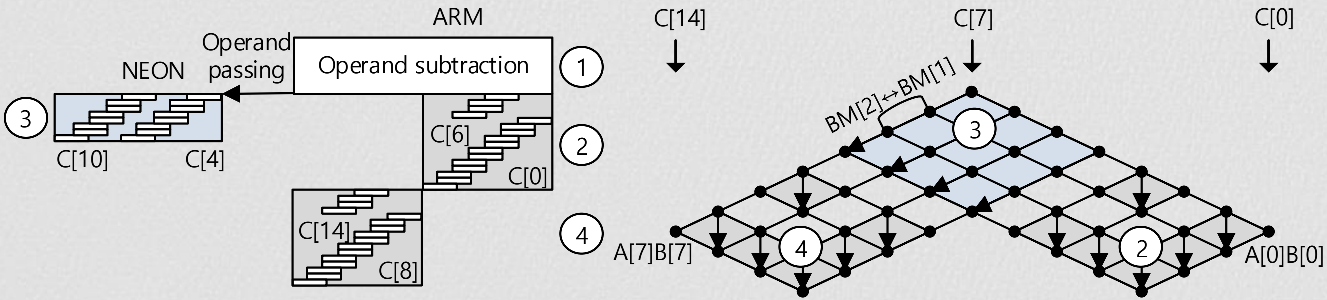
$C_L \leftarrow A_L \cdot B_L$ {ARM}

$C_M \leftarrow A_M \cdot B_M$ {NEON}

$C_H \leftarrow A_H \cdot B_H$ {ARM}

Interleaved section end

return $C \leftarrow C_L + (C_L + C_H - C_M) \cdot 2^{\frac{m}{2}} + C_H \cdot 2^m$ {ARM}



아이소 지니 기반 양자 내성 암호

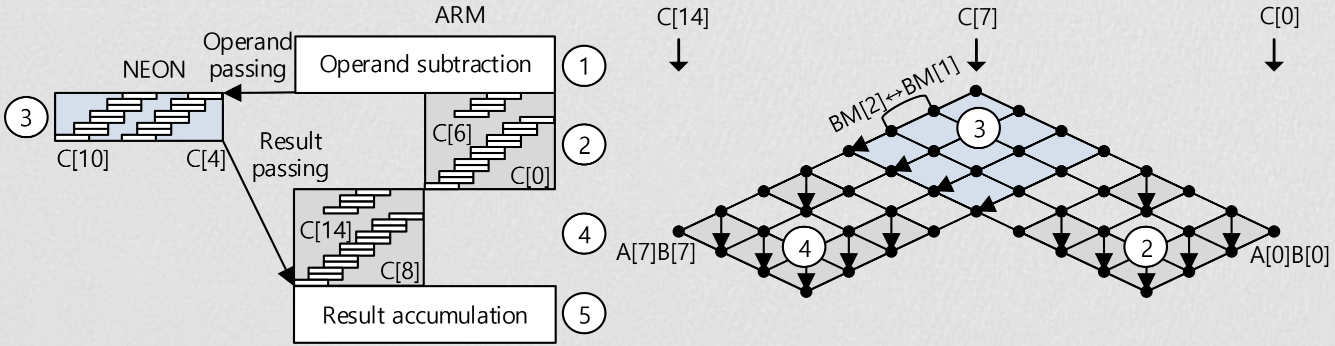
Algorithm 1 Unified ARM/NEON multiplication

Input: Two m-bit operands $A = (A_H || A_L)$ and $B = (B_H || B_L)$
Output: $2m$-bit result C

$A_M \leftarrow |A_L - A_H|$ {ARM}
$B_M \leftarrow |B_L - B_H|$ {ARM}

Interleaved section begin
$C_L \leftarrow A_L \cdot B_L$ {ARM}
$C_M \leftarrow A_M \cdot B_M$ {NEON}
$C_H \leftarrow A_H \cdot B_H$ {ARM}
Interleaved section end

return $C \leftarrow C_L + (C_L + C_H - C_M) \cdot 2^{\frac{m}{2}} + C_H \cdot 2^m$ {ARM}



아이소 지니 기반 양자 내성 암호

Efficient Montgomery reduction: 나눗셈을 곱셈으로..
→ Montgomery-friendly modulus: 절반의 곱셈 생략

- $$p503 = 2^{250}3^{159} - 1$$

0x4066F541811E1E6045C6BDDA77A4D01B9BF6C87B7E7DAF13085BDA2211E7A0ABFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
(in hexadecimal)

- $$p503 + 1 = 2^{250}3^{159}$$

0x4066F541811E1E6045C6BDDA77A4D01B9BF6C87B7E7DAF13085BDA2211E7A0AC00000000000000000000000000000000000000000000000000000000000000
(in hexadecimal)

아이소 지니 기반 양자 내성 암호

Algorithm 5 Unified ARM/NEON Montgomery reduction for SIDH-friendly primes

Input: $\tilde{M} = M + 1 = (\tilde{M}_H \| \tilde{M}_L)$ for an odd m-bit modulus M, the Montgomery radix $R = 2^s$, where $s = wn$ with $w = 32$ and $n = \lceil m/w \rceil$, an operand $T \in [0, M^2 - 1]$, and pre-computed constant $M' = -M^{-1} \bmod R$

Output: m-bit Montgomery product $Z = T \cdot R^{-1} \bmod M$

```
Set Q = (Q_H||Q_L) ← T · M' mod 2^s

Interleaved section begin
T ← T + (M̃_H · Q_L) · 2^(s/2)                {ARM}
Z_4 ← M̃_H · Q_H                              {NEON}
Interleaved section end

Z ← (T + Z_4 · 2^s − Q)/2^s                  {ARM}
if Z ≥ M then
    Z ← Z − M                                {ARM}
return Z
```

$$Q = (Q_H \| Q_L) \leftarrow T \cdot M' \bmod 2^s$$

$$T \leftarrow T + (\tilde{M}_H \cdot Q_L) \cdot 2^{\frac{s}{2}} \quad \{\texttt{ARM}\}$$

$$Z_4 \leftarrow \tilde{M}_H \cdot Q_H \quad \{\texttt{NEON}\}$$

$$Z \leftarrow (T + Z_4 \cdot 2^s - Q)/2^s \quad \{\texttt{ARM}\}$$

NEON $\xleftarrow{\text{Operand passing}}$ ARM

아이소 지니 기반 양자 내성 암호

Algorithm 5 Unified ARM/NEON Montgomery reduction for SIDH-friendly primes

Input: $\tilde{M} = M + 1 = (\tilde{M}_H \| \tilde{M}_L)$ for an odd m-bit modulus M, the Montgomery radix $R = 2^s$, where $s = wn$ with $w = 32$ and $n = \lceil m/w \rceil$, an operand $T \in [0, M^2 - 1]$, and pre-computed constant $M' = -M^{-1} \bmod R$

Output: m-bit Montgomery product $Z = T \cdot R^{-1} \bmod M$

Set $Q = (Q_H \| Q_L) \leftarrow T \cdot M' \bmod 2^s$

Interleaved section begin

$T \leftarrow T + (\tilde{M}_H \cdot Q_L) \cdot 2^{\frac{s}{2}}$ {ARM}

$Z_4 \leftarrow \tilde{M}_H \cdot Q_H$ {NEON}

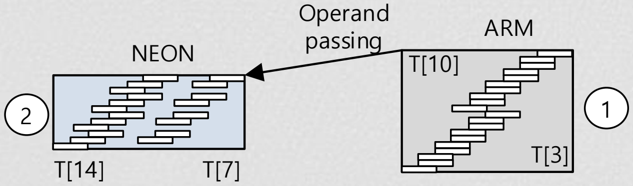
Interleaved section end

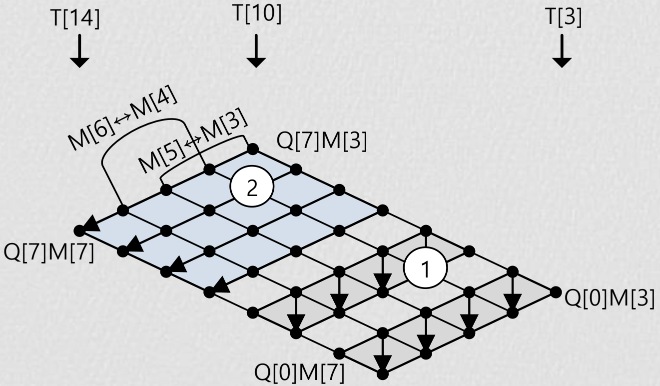
$Z \leftarrow (T + Z_4 \cdot 2^s - Q)/2^s$ {ARM}

if $Z \geq M$ **then**

  $Z \leftarrow Z - M$ {ARM}

return Z





아이소 지니 기반 양자 내성 암호

Algorithm 5 Unified ARM/NEON Montgomery reduction for SIDH-friendly primes

Input: $\tilde{M} = M + 1 = (\tilde{M}_H \| \tilde{M}_L)$ for an odd m-bit modulus M, the Montgomery radix $R = 2^s$, where $s = wn$ with $w = 32$ and $n = \lceil m/w \rceil$, an operand $T \in [0, M^2 - 1]$, and pre-computed constant $M' = -M^{-1} \bmod R$

Output: m-bit Montgomery product $Z = T \cdot R^{-1} \bmod M$

1: Set $Q = (Q_H \| Q_L) \leftarrow T \cdot M' \bmod 2^s$

`Interleaved section begin`

2: $T \leftarrow T + (\tilde{M}_H \cdot Q_L) \cdot 2^{\frac{s}{2}}$ {ARM}

3: $Z_4 \leftarrow \tilde{M}_H \cdot Q_H$ {NEON}

`Interleaved section end`

4: $Z \leftarrow (T + Z_4 \cdot 2^s - Q)/2^s$ {ARM}

5: **if** $Z \geq M$ **then**

6: $\quad Z \leftarrow Z - M$ {ARM}

7: **return** Z