A High-Performance Hardware Implementation of the LESS Digital Signature Scheme

Luke Beckwith[1,2], Robert Wallace[1], Kamyar Mohajerani[1], and Kris Gaj[1]

(1) Cryptographic Engineering Research Group (CERG) at George Mason University

(2) PQSecure Technologies

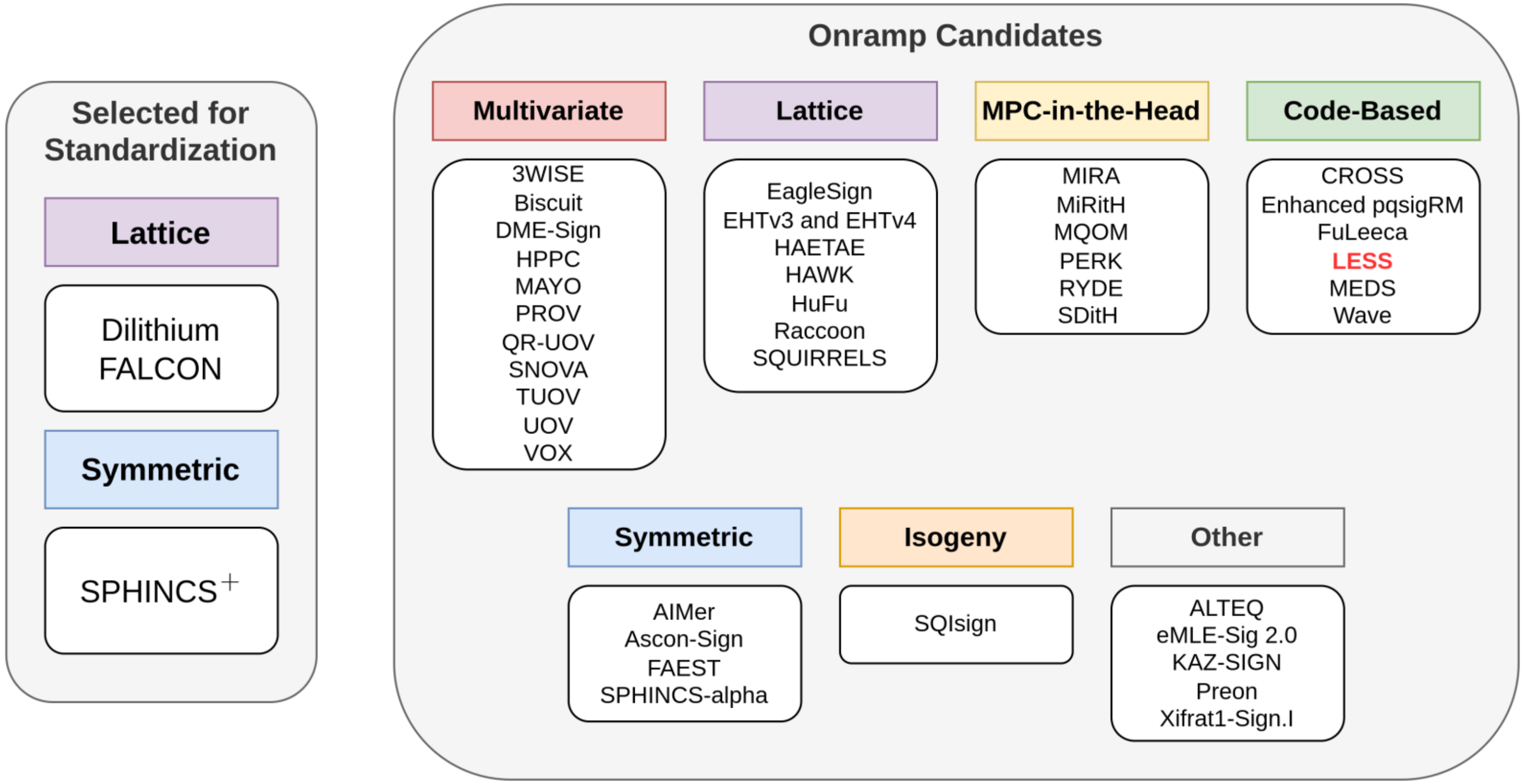
Outline

- Brief overview of PQC status
- Introduction to LESS
  - Mathematical background
  - Algorithm details
  - Parameters
- Hardware architecture
  - Top-level structure
  - Details of RREF implementation
- Results and comparison

PQC Signatures

Winners:
- 3 algorithms
- 2 types of cryptography

New Candidates:
- 40 algorithms
- 7+ types



LESS

- LESS (Linear Equivalence Signature Scheme):
  - Code-based algorithm based on the difficulty of the linear equivalence problem
  - Constructed using Fiat-Shamir
  - Main elements are large matrices with elements in F_q
- Core Operation: RREF(RREF(Generator) x Monomial Matrix)

$$\underbrace{\begin{bmatrix} 1 & 1 & 0 & 0 & 5 \\ 0 & 0 & 1 & 0 & 6 \\ 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{RREF(Generator)} \times \underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6 \\ 0 & 2 & 0 & 0 & 0 \end{bmatrix}}_{Monomial\ Matrix} = \begin{bmatrix} 3 & 3 & 0 & 1 & 0 \\ 0 & 5 & 2 & 0 & 0 \\ 0 & 4 & 0 & 0 & 6 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \xrightarrow{RREF} \begin{bmatrix} 1 & 0 & 0 & 5 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Background - RREF

Reduced Row Echelon Form (RREF):

A matrix is said to be in RREF if:

1. Rows with only zeros are at the bottom of the matrix
2. The leftmost non-zero (leading) entry of each row is to the right of the leading entry of all rows above it
3. All leading entries are 1
4. Each column containing a leading 1 has zeros in all other entries

The leading entries are also referred to as "pivots" and the corresponding columns as "pivot columns"

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 5 \\ 0 & 0 & 1 & 0 & 6 \\ 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Example of a matrix in RREF.
Pivots are in green.

Background – Monomial Matrix

Monomial Matrix:

A monomial matrix is a combination of a scalar matrix and a permutation matrix.

Each column and row have only one non-zero entry which is in F_q^*. The set of monomial matrices is referred to as M_n.

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6 \\ 0 & 2 & 0 & 0 & 0 \end{bmatrix}$$

Background – Generator Matrix and LEP

Generator Matrix:

A generator matrix is a matrix whose rows form the basis for a linear code. So, for generator G of code C, the codeword c of message m is calculated by:

$$c = mG$$

Two generator matrices are said to be ***linearly equivalent*** if there exist a monomial matrix Q and an invertible matrix S, such that

$$G' = SGQ$$

$$\begin{bmatrix} 1 & 0 & 0 & 5 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Linear Equivalence Problem: Given G’ and G, it is difficult to find Q

LEP Sigma Identification Protocol

Prover

Verifier

Public Data: G

Secret Key: Q

Public Key: $G' = RREF(GQ)$

User Keypair: $[Q, G']$
Trivial Keypair: $[I_k, G]$

Commitment
$\tilde{Q} \leftarrow M_n$
$h = H(RREF(G\tilde{Q}))$

$\xrightarrow{h}$

Challenge
$b \leftarrow 0, 1$

$\xleftarrow{b}$

Response
If b is 0: $\mu \leftarrow \tilde{Q}$
if b is 1: $\mu \leftarrow Q^{-1}\tilde{Q}$

$\xrightarrow{\mu}$

Verify
If b is 0: $h == H(RREF(G\mu))$
if b is 1: $h == H(RREF(G'\mu))$

Difficulty can be increased by performing multiple rounds or by using multiple keypairs

LESS Key Generation (Simplified)

$$\underbrace{\begin{bmatrix} \quad G_i \quad \end{bmatrix}}_{k \times n} = \boxed{RREF}\left(\underbrace{\begin{bmatrix} G_0 \in F_q^{k\times n} \end{bmatrix}}_{k \times n} \times \underbrace{\begin{bmatrix} Q_i \xleftarrow{\$} M_n \end{bmatrix}}_{n \times n} \right) \Bigg\} \text{ For } i \in [1, s-1]$$

Public Key

$s - 1$ RREF Operations

Public Data (in RREF)

Secret Key

- Each keypair is an instance of LEP
- Multiple keypairs can be used to lower number of rounds needed
  - First keypair is trivial keypair (I_k, G_0)
  - $s - 1$ additional keypairs generated

LESS Sign Part 1 (Simplified)

Commitment

Public Data (in RREF)

Random Monomial

Step 1: Calculate commitment matricies

t RREF Operations

$$\left\{ \underbrace{\left[\ \tilde{G}_i\ \right]}_{k \times n} = \boxed{RREF}\left(\underbrace{\left[G_0 \in F_q^{k\times n} \right]}_{k \times n} \times \underbrace{\left[\tilde{Q}_i \xleftarrow{\$} M_n \right]}_{n \times n} \right) \right\} \text{ For } i \in [0, t-1]$$

Step 2: Hash commitments and message

$$d \leftarrow HASH(\tilde{G}_0 || \ldots || \tilde{G}_{t-1} || M)$$

LESS Sign Part 2 (Simplified)

ω non-zero entries

Step 3: Parse challenge

$$x_0, \ \ldots \ x_{t-1} \in [0, s-1] \leftarrow CSPRNG(d, S_{t,\omega})$$

Step 4: Calculate responses

$$\left\{ \underset{n \times n}{\begin{bmatrix} rsp_i \end{bmatrix}} \leftarrow \underset{n \times n}{\begin{bmatrix} Q_{x_i}^{-1} \end{bmatrix}} \times \underset{n \times n}{\begin{bmatrix} \tilde{Q}_i \end{bmatrix}} \ \ or \ \ \underset{n \times n}{\begin{bmatrix} \tilde{Q}_i \end{bmatrix}} \right\} \text{For } i \in [0, t-1]$$

Secret Key

When $x_i \neq 0$

When $x_i = 0$

$$signature = (d, rsp_0, \ldots, rsp_{t-1})$$

LESS Verify (Simplified)

$$signature = (d, rsp_0, \ldots, rsp_{t-1})$$

Step 1: Parse challenge

$$x_0, \ \ldots \ x_{t-1} \leftarrow CSPRNG(d, S_{t,\omega})$$

Step 2: Recalculate commitments

$$\underbrace{\left[\ \bar{G}_i \ \right]}_{k \times n} = \boxed{RREF}\Big(\underbrace{\left[\ G_{x_i} \ \right]}_{k \times n} \times \underbrace{\left[\ rsp_i \ \right]}_{n \times n} \Big) \quad \text{For } i \in [0, t-1]$$

t RREF Operations

Public Key

Step 3: Hash commitments and message

$$d' \leftarrow HASH(\bar{G}_0||\ldots||\bar{G}_{t-1}||M)$$

Step 4: Check if commitments matched

$$d' == d?$$

LESS Parameters

NIST Security Level	Parameter Set	Code Parameters		
		n	*k*	*q*
1	*LESS-1b*	252	126	127
	LESS-1i			
	LESS-1s			
3	*LESS-3b*	400	200	127
	LESS-3s			
5	*LESS-5b*	548	274	127
	LESS-5s			

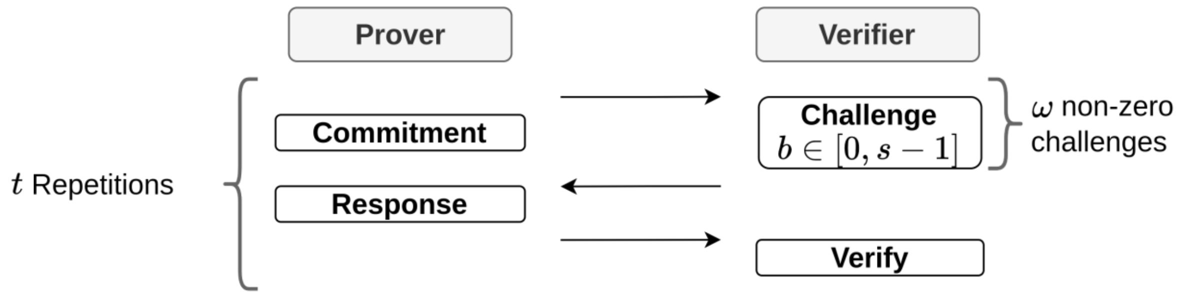
$$\left.\overbrace{\begin{bmatrix} \qquad G_i \qquad \end{bmatrix}}^{n\text{ Columns}}\right\} k\text{ Rows}$$

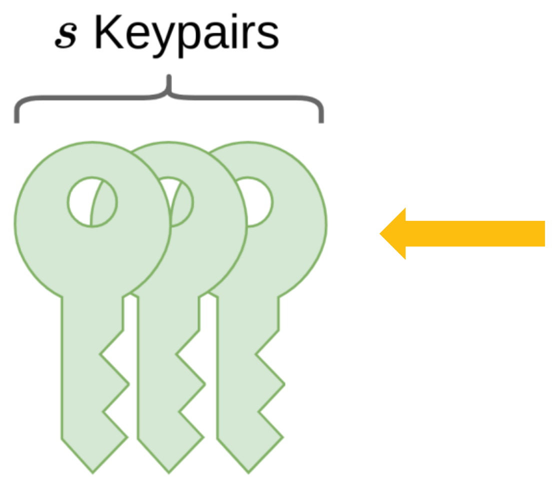
$$G_i[a,b] \in F_q$$

s → “Short” - minimize Sig size
b → “Balanced” - minimize PK + Sig size
i → “Intermediate” – in between b and s

LESS Parameters

NIST Security Level	Parameter Set	Protocol Parameters		
		t	ω	s
1	*LESS-1b*	247	30	2
	LESS-1i	244	20	4
	LESS-1s	198	17	8
3	*LESS-3b*	759	33	2
	LESS-3s	895	26	3
5	*LESS-5b*	1352	40	2
	LESS-5s	907	37	3





The first keypair is $[I_k, G_0]$

LESS Parameters

In LESS:

- Signature size $\nearrow$ when $k\nearrow$, $t\nearrow$ and $\omega\nearrow$
- Public key size $\nearrow$ when $n\nearrow$, $k\nearrow$, and $s\nearrow$
- Secret key size is independent of parameters

In this architecture:

- Area $\nearrow$ to $n \nearrow$
- Keygen Cycle Latency $\nearrow$ when $k\nearrow$ and $s\nearrow$
- Sign/Verify Cycle Latency $\nearrow$ when $k\nearrow$and $t\nearrow$

Parameter Meaning

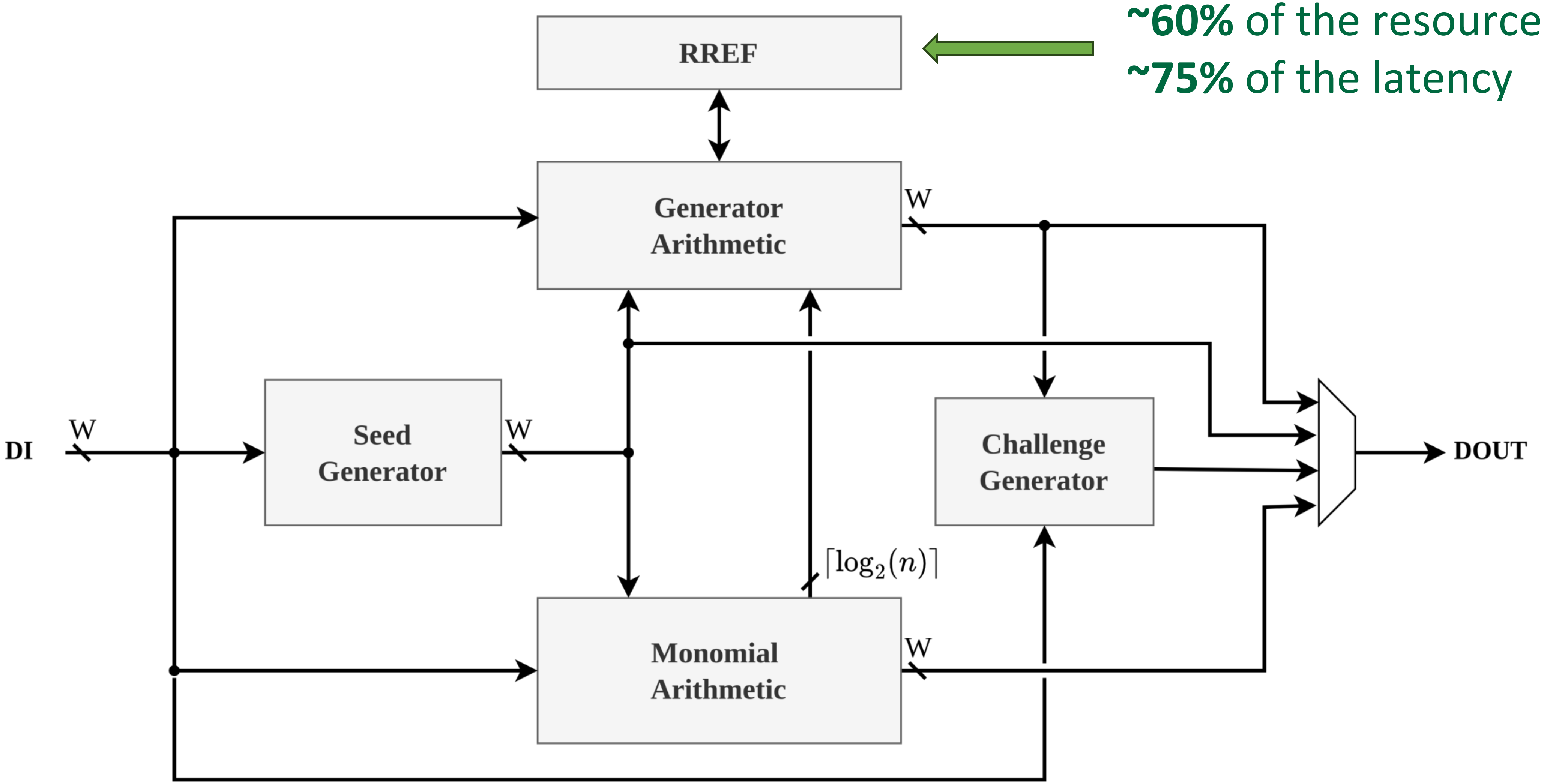
Code Parameters

n: Generator columns
k: Generator rows
q: Generator element modulus

Protocol Parameters

t: Protocol repetitions
ω: Non-zero challenges
s: Number of keypairs

Top-Level Architecture



~60% of the resources
~75% of the latency

RREF

Operation to convert an input matrix to RREF

$$\begin{array}{cc} & n \\ k & \begin{bmatrix} 2 & 2 & 3 & 3 & 1 & 4 & 3 \\ 3 & 3 & 1 & 5 & 1 & 4 & 3 \\ 5 & 3 & 1 & 2 & 2 & 2 & 6 \end{bmatrix} \end{array} \longrightarrow \begin{bmatrix} 1 & 0 & 0 & 0 & 6 & 0 & 4 \\ 0 & 1 & 5 & 0 & 3 & 1 & 2 \\ 0 & 0 & 0 & 1 & 6 & 3 & 4 \end{bmatrix}$$

G RREF(G)

All arithmetic operations performed modulo $\boldsymbol{q} = 7$

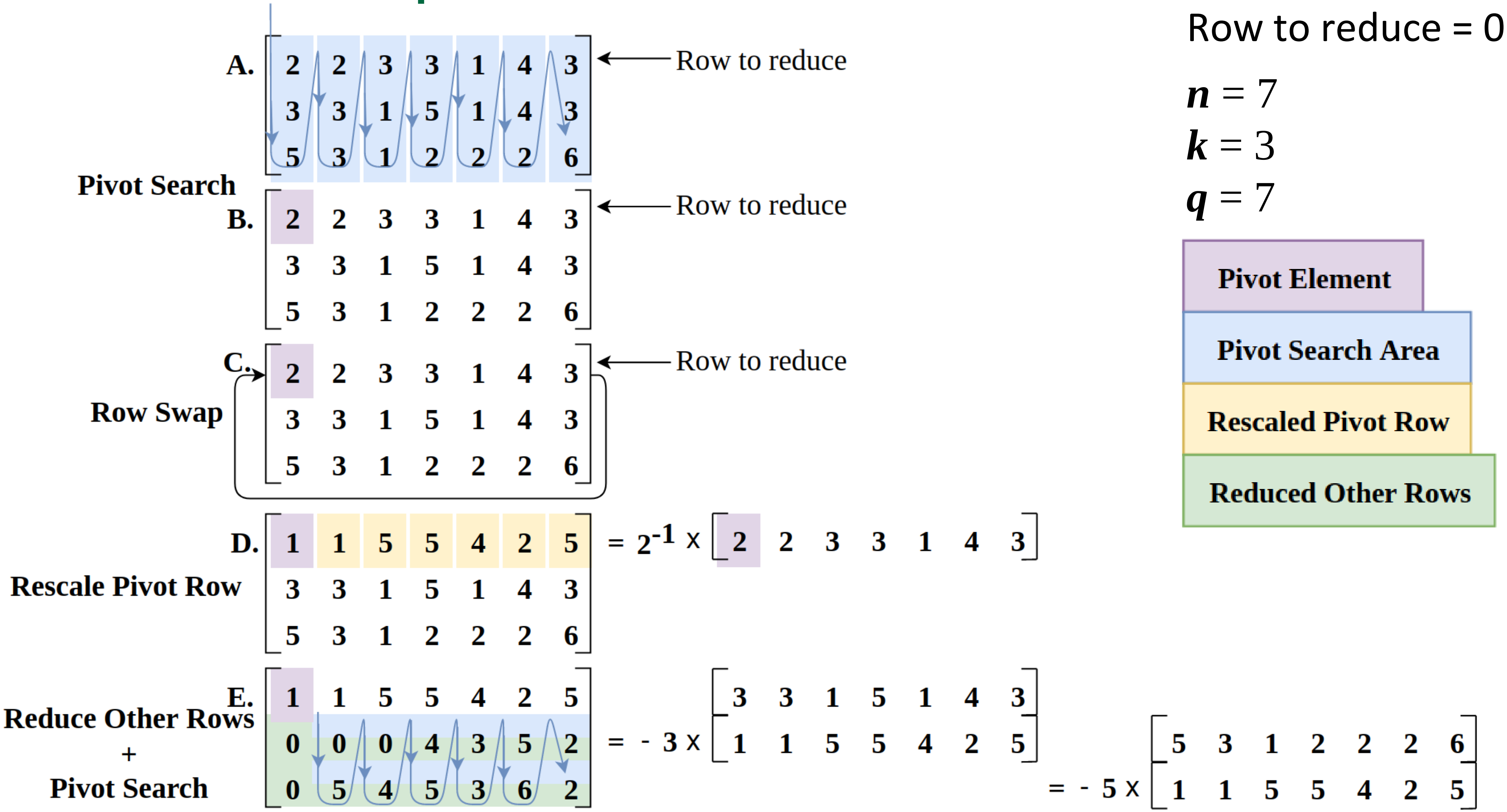
RREF – Example

Row to reduce = 0

$\boldsymbol{n} = 7$

$\boldsymbol{k} = 3$

$\boldsymbol{q} = 7$



RREF – Example

Row to reduce = 1

$\boldsymbol{n} = 7$

$\boldsymbol{k} = 3$

$\boldsymbol{q} = 7$

Pivot Element

Pivot Search Area

Rescaled Pivot Row

Reduced Other Rows

Pivot Search Results

F.
$$\begin{bmatrix} 1 & 1 & 5 & 5 & 4 & 2 & 5 \\ 0 & 0 & 0 & 4 & 3 & 5 & 2 \\ 0 & 5 & 4 & 5 & 3 & 6 & 2 \end{bmatrix} \longleftarrow \text{Row to reduce (row 2)}$$

Row Swap

G.
$$\begin{bmatrix} 1 & 1 & 5 & 5 & 4 & 2 & 5 \\ 0 & 0 & 0 & 4 & 3 & 5 & 2 \\ 0 & 5 & 4 & 5 & 3 & 6 & 2 \end{bmatrix} \longleftarrow \text{Row to reduce (row 2)}$$

H.
$$\begin{bmatrix} 1 & 1 & 5 & 5 & 4 & 2 & 5 \\ 0 & 5 & 4 & 5 & 3 & 6 & 2 \\ 0 & 0 & 0 & 4 & 3 & 5 & 2 \end{bmatrix} \longleftarrow \text{Row to reduce (row 2)}$$

I.
Rescale Pivot Row
$$\begin{bmatrix} 1 & 1 & 5 & 5 & 4 & 2 & 5 \\ 0 & 1 & 5 & 1 & 2 & 4 & 6 \\ 0 & 0 & 0 & 4 & 3 & 5 & 2 \end{bmatrix}$$
$$= 5^{-1} \times \begin{bmatrix} 0 & 5 & 4 & 5 & 3 & 6 & 2 \end{bmatrix}$$

J.
Reduce Other Rows
+
Pivot Search
$$\begin{bmatrix} 1 & 0 & 0 & 4 & 2 & 5 & 6 \\ 0 & 1 & 5 & 1 & 2 & 4 & 6 \\ 0 & 0 & 0 & 4 & 3 & 5 & 2 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 1 & 5 & 5 & 4 & 2 & 5 \end{bmatrix} = -1 \times \begin{bmatrix} 0 & 1 & 5 & 1 & 2 & 4 & 6 \end{bmatrix}$$
$$\begin{bmatrix} 0 & 0 & 0 & 4 & 3 & 5 & 2 \end{bmatrix} = -0 \times \begin{bmatrix} 0 & 1 & 5 & 1 & 2 & 4 & 6 \end{bmatrix}$$

RREF – Example

Row to reduce = 2

$\boldsymbol{n} = 7$

$\boldsymbol{k} = 3$

$\boldsymbol{q} = 7$

Pivot Element

Pivot Search Area

Rescaled Pivot Row

Reduced Other Rows

Pivot Search Results

K. $\begin{bmatrix} 1 & 0 & 0 & 4 & 2 & 5 & 6 \\ 0 & 1 & 5 & 1 & 2 & 4 & 6 \\ 0 & 0 & 0 & 4 & 3 & 5 & 2 \end{bmatrix}$ ← Row to reduce

Row Swap

L. $\begin{bmatrix} 1 & 0 & 0 & 4 & 2 & 5 & 6 \\ 0 & 1 & 5 & 1 & 2 & 4 & 6 \\ 0 & 0 & 0 & 4 & 3 & 5 & 2 \end{bmatrix}$ ← Row to reduce

Rescale Pivot Row

M. $\begin{bmatrix} 1 & 0 & 0 & 4 & 2 & 5 & 6 \\ 0 & 1 & 5 & 1 & 2 & 4 & 6 \\ 0 & 0 & 0 & 1 & 6 & 3 & 4 \end{bmatrix}$

$= 4^{-1} \times \begin{bmatrix} 0 & 0 & 0 & 4 & 3 & 5 & 2 \end{bmatrix}$

Reduce Other Rows

N. $\begin{bmatrix} 1 & 0 & 0 & 0 & 6 & 0 & 4 \\ 0 & 1 & 5 & 0 & 3 & 1 & 2 \\ 0 & 0 & 0 & 1 & 6 & 3 & 4 \end{bmatrix}$

$\begin{bmatrix} 1 & 0 & 0 & 4 & 2 & 5 & 6 \end{bmatrix}$

$= -4 \times \begin{bmatrix} 0 & 0 & 0 & 1 & 6 & 3 & 4 \end{bmatrix}$

$\begin{bmatrix} 0 & 1 & 5 & 1 & 2 & 4 & 6 \end{bmatrix}$

$= -1 \times \begin{bmatrix} 0 & 0 & 0 & 1 & 6 & 3 & 4 \end{bmatrix}$

O. $\begin{bmatrix} 1 & 0 & 0 & 0 & 6 & 0 & 4 \\ 0 & 1 & 5 & 0 & 3 & 1 & 2 \\ 0 & 0 & 0 & 1 & 6 & 3 & 4 \end{bmatrix}$

RREF – Algorithm

- Four major operations:
  1. *Pivot Search*
  2. *Row Swap*
  3. *Rescale Pivot Row*
  4. *Reduce Other Rows*
- Opportunities for parallelization:
  1. Arithmetic performed on entire row
  2. *Pivot Search* while *Reduce Other Rows*
  3. Row operations in *Reduce Other Rows* are independent of each other
- Constant-time implementation

Input: Matrix $G \in \mathbb{Z}_q^{k \times n}$
Output: Matrix $G \in \mathbb{Z}_q^{k \times n}$

for rtr $\in [0, k-1]$ **do**

Pivot Search

- vld_piv $\leftarrow 0$
- **for** col $\in$ [rtr, $n-1$] **do**
  - **for** row $\in$[rtr, $k-1$] **do**
    - **if** $(G[\text{row}][\text{col}] > 0)$ *and* (vld_piv $= 0$) **then**
      - piv_row $\leftarrow$ row
      - piv_col $\leftarrow$ col
      - vld_piv $\leftarrow 1$

swap_row(G[rtr], G[piv_row])

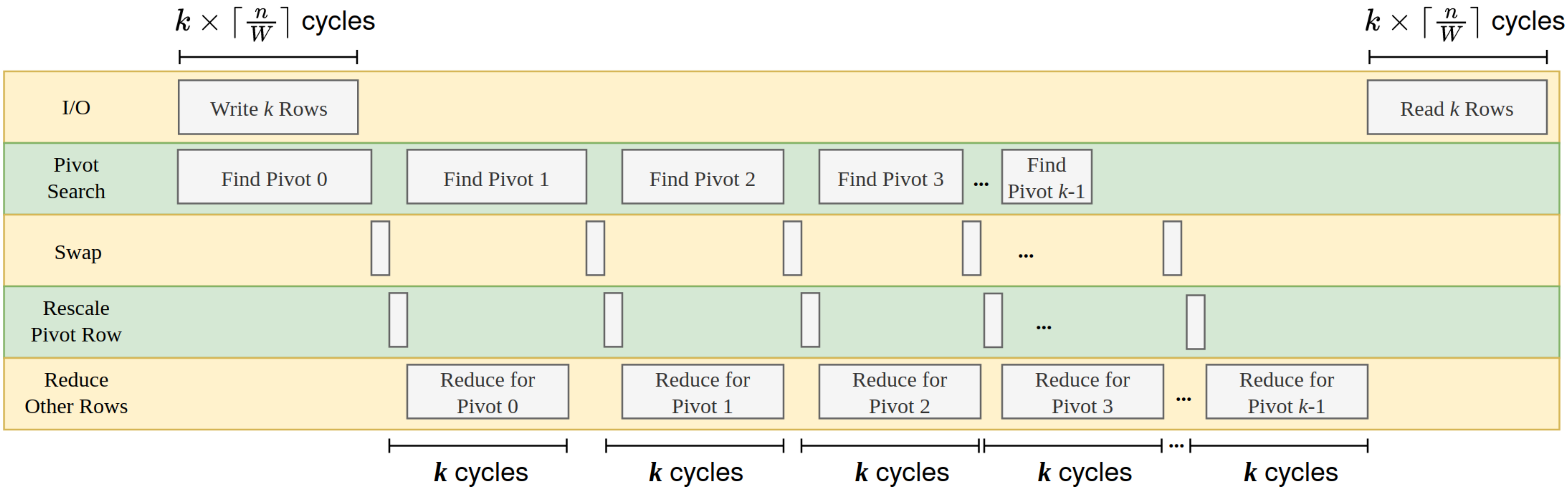
Rescale Pivot Row

- $m \leftarrow G[\text{rtr}][\text{piv\_col}]^{-1} \bmod q$
- **for** col $\in [0, n-1]$ **do**
  - $G[\text{rtr}][\text{col}] \leftarrow m \cdot G[\text{rtr}][\text{col}] \bmod q$

Reduce Other Rows

- **for** row $\in [0, k-1]$ **do**
  - **if** row $\neq$ rtr **then**
    - $m \leftarrow G[\text{row}][\text{piv\_col}]$
    - **for** col $\in$ [piv_col, $n-1$] **do**
      - tmp $\leftarrow G[\text{row}][\text{piv\_col}] \cdot G[\text{rtr}][\text{col}] \bmod q$
      - $G[\text{row}][\text{col}] \leftarrow G[\text{row}][\text{col}] -$ tmp $\bmod q$

RREF – Operational Flow

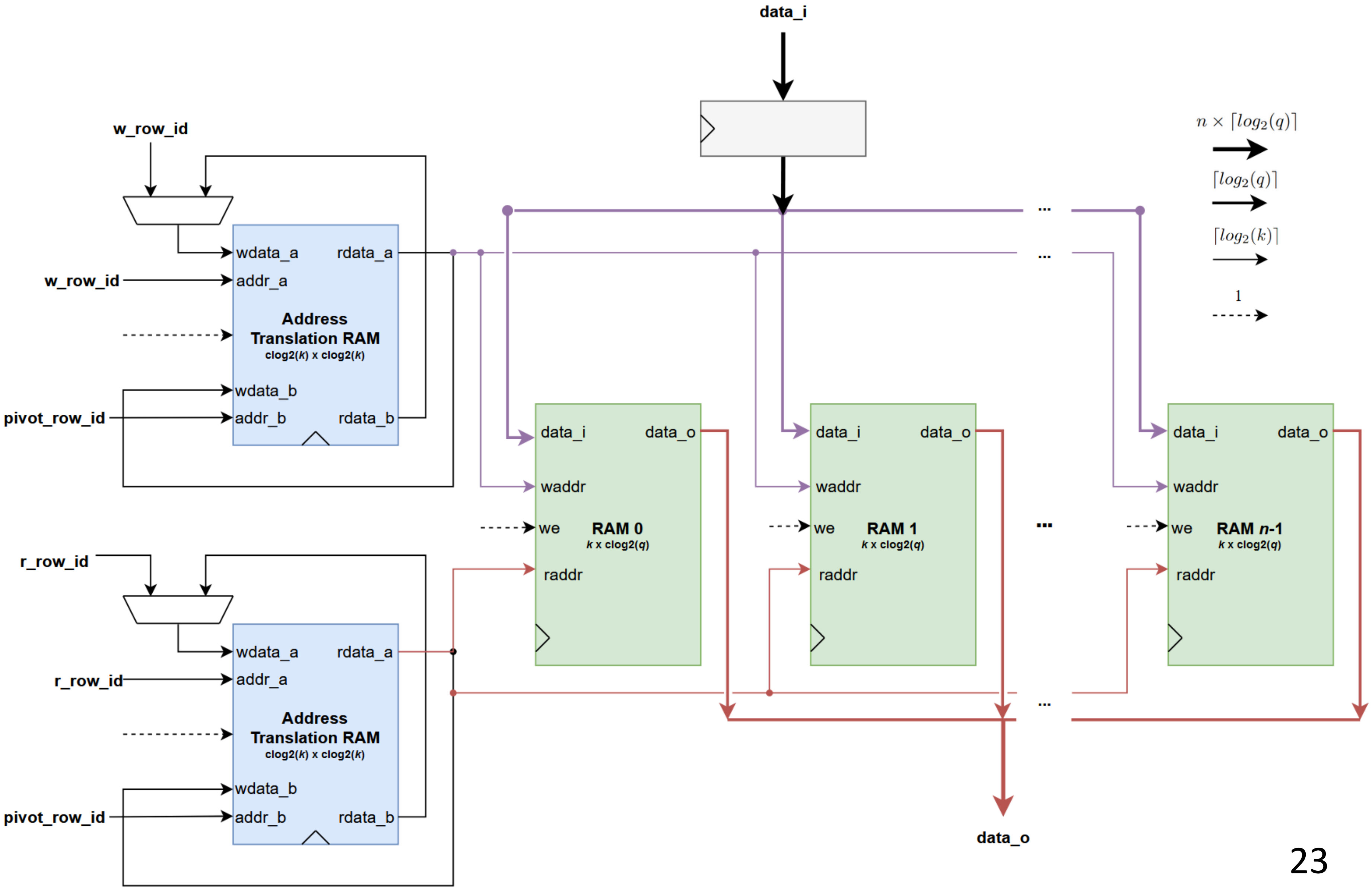


Latency (clock cycles) = $k^2 + 3k + 58$

(including implemented pipeline stages,
not including I/O)

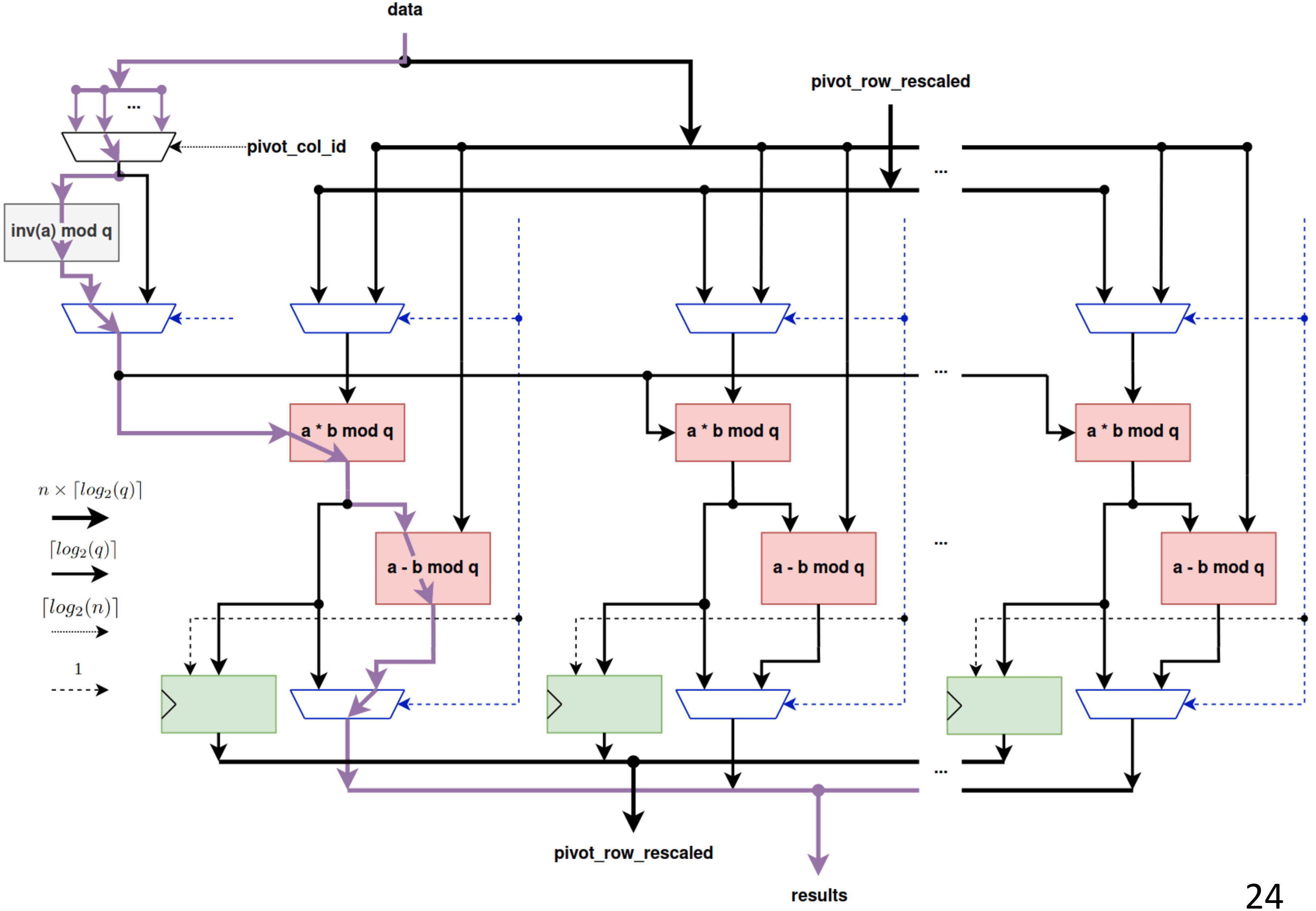
RREF – Column Memory

- ***n* RAMs** to hold one column of the matrix, each
- **Parallel memory units** to access entire row of matrix in one cycle
- **Address translation tables** for constant time conditional row swap
- **Separate input and output** ports

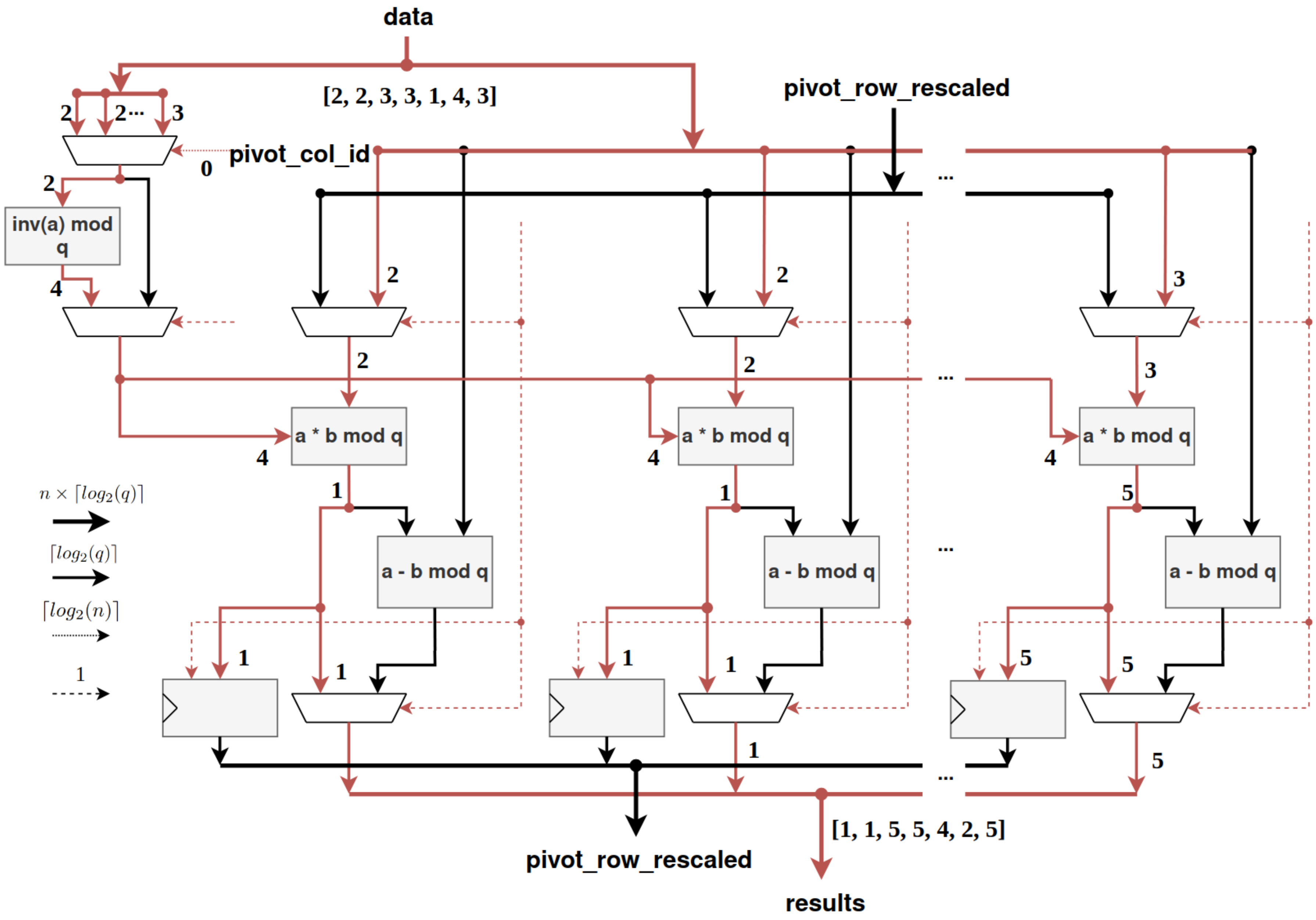


RREF – Row Arithmetic

- **Hardware re-use** for *Rescale Pivot Row* and *Reduce Other Rows*
- **Parallel arithmetic** units to operate on entire row at a time.
- **Long feed forward critical path** – good for pipelining
- **Registers** to hold result from *Rescale Pivot Row* to be used during *Reduce Other Rows*



RREF – Row Arithmetic – Rescale Pivot Row



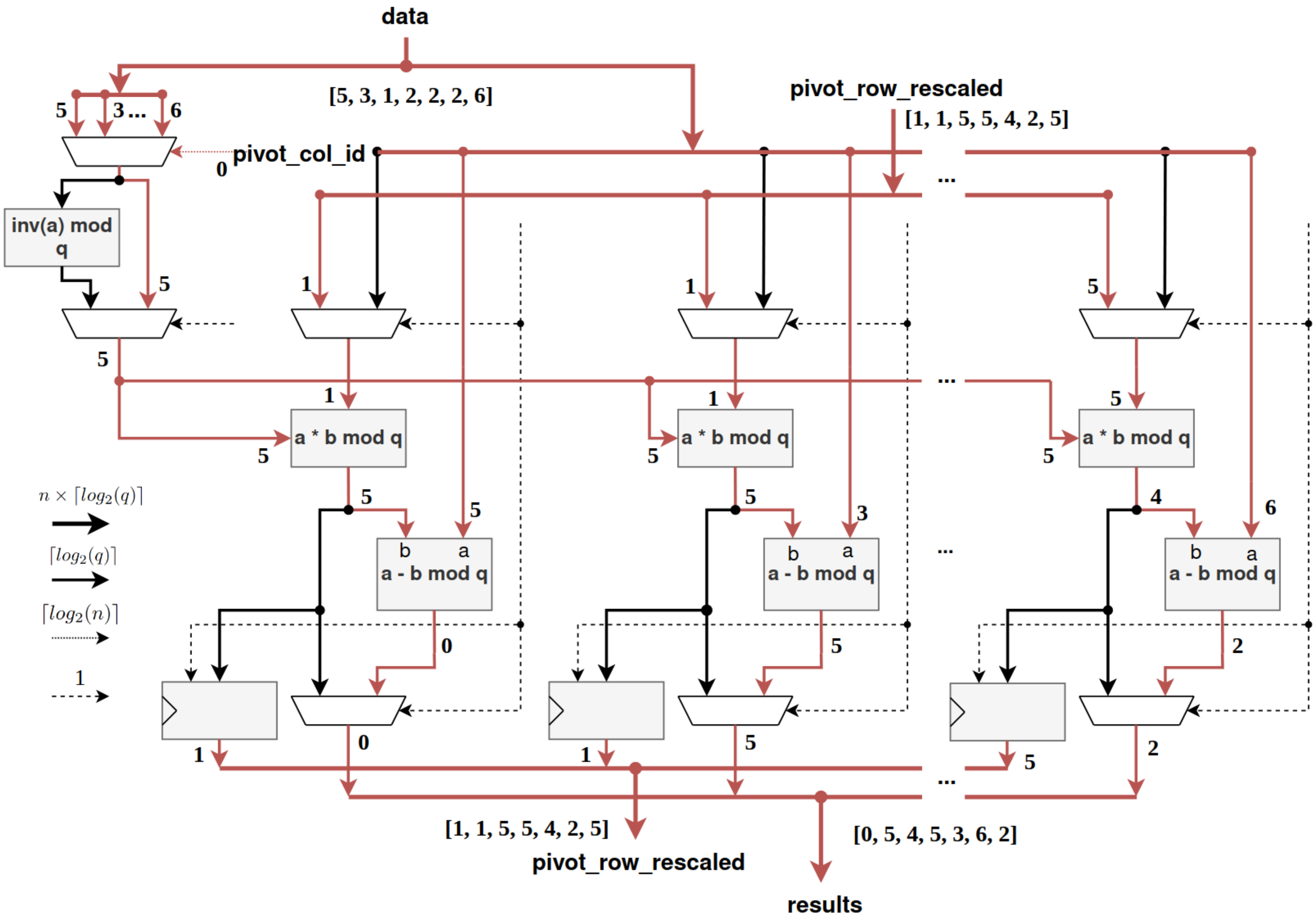
$\boldsymbol{n}=7$
$\boldsymbol{k}=3$
$\boldsymbol{q}=7$

$$\begin{bmatrix} 2 & 2 & 3 & 3 & 1 & 4 & 3 \\ 3 & 3 & 1 & 5 & 1 & 4 & 3 \\ 5 & 3 & 1 & 2 & 2 & 2 & 6 \end{bmatrix}$$

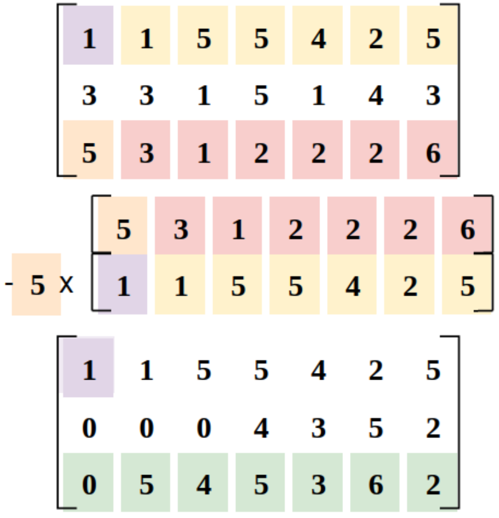
$$2^{-1} \times \begin{bmatrix} 2 & 2 & 3 & 3 & 1 & 4 & 3 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 5 & 5 & 4 & 2 & 5 \\ 3 & 3 & 1 & 5 & 1 & 4 & 3 \\ 5 & 3 & 1 & 2 & 2 & 2 & 6 \end{bmatrix}$$

RREF – Row Arithmetic – Reduce Other Rows



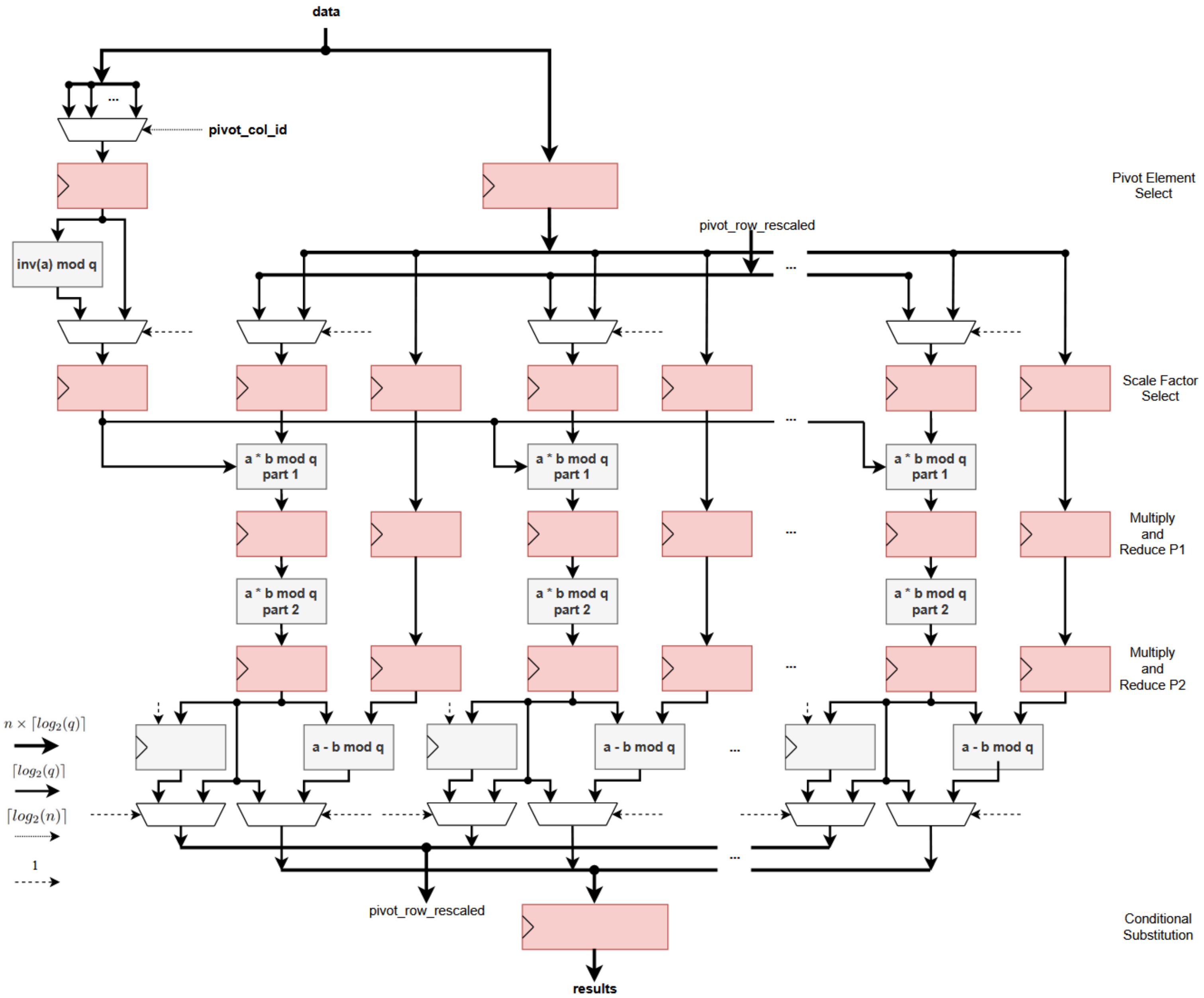
n=7
k=3
q=7



RREF – Row Arithmetic Pipeline

- Enables higher clock frequency
- Generates new result every clock cycle
- Increases flip-flop utilization

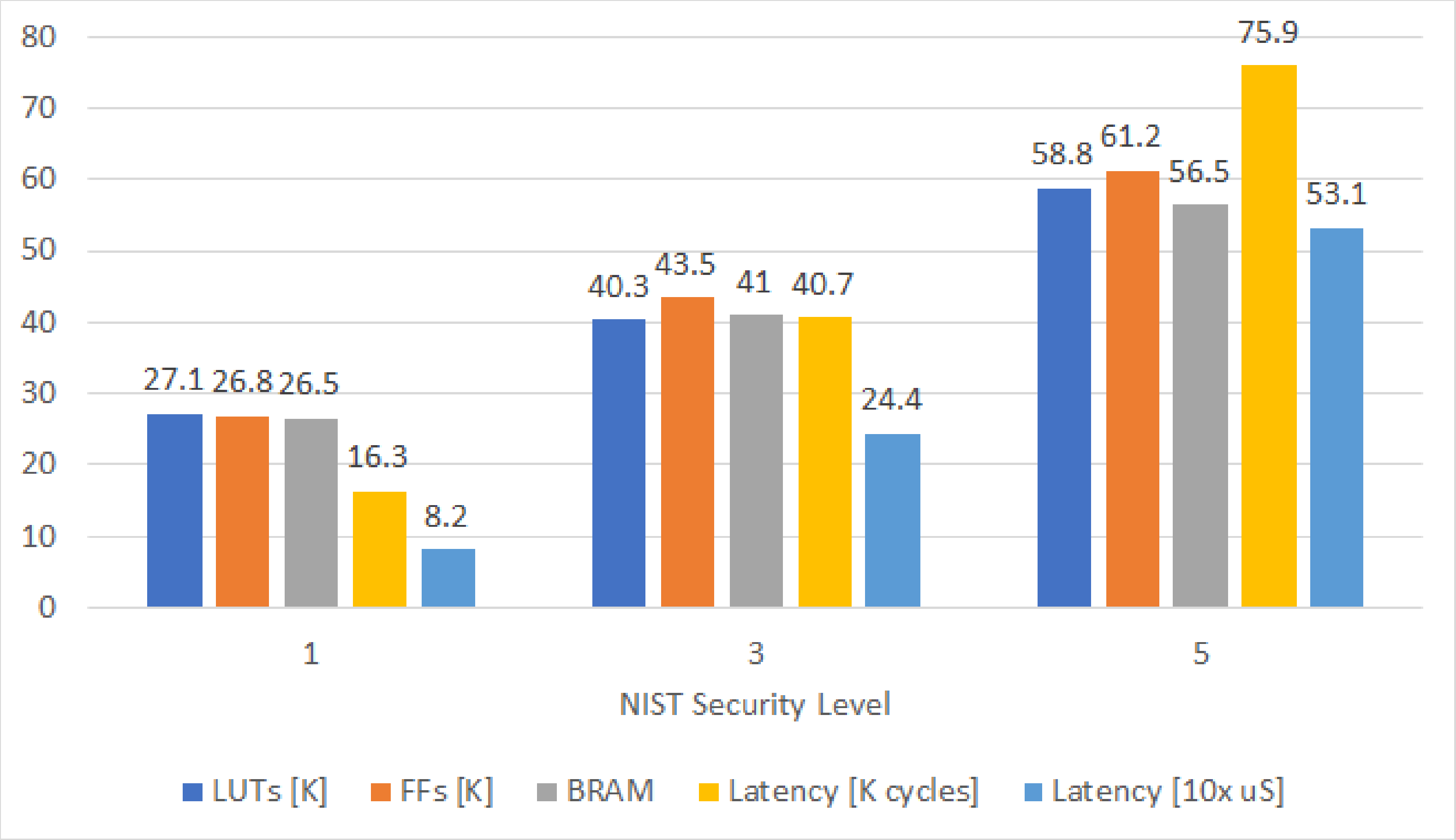
NIST Security Level	n	k	Frequency (MHz)
1	252	126	200
3	400	200	167
5	548	274	142



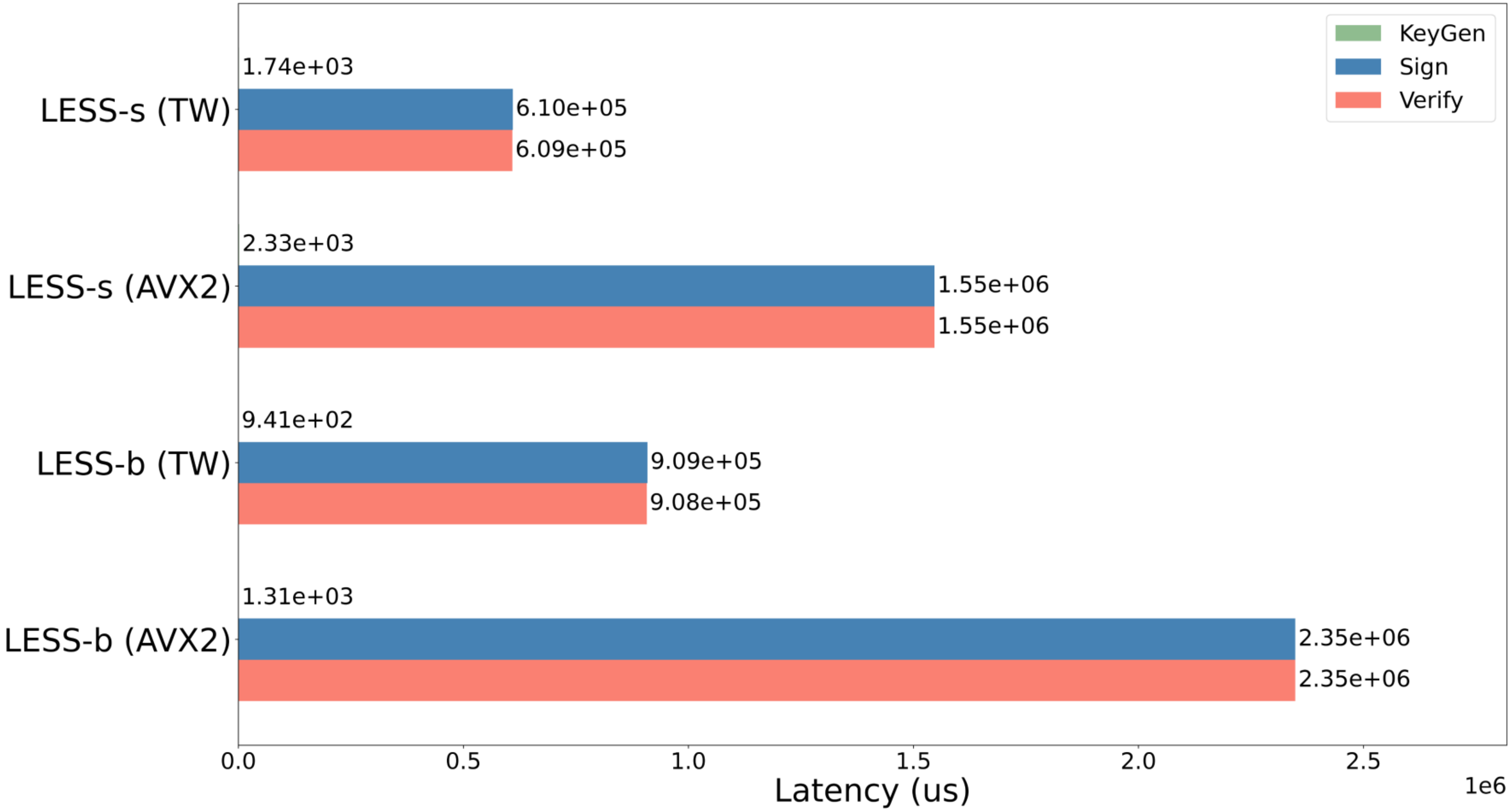
RREF – Results

- Area ~ $\boldsymbol{n}$
- $\boldsymbol{n}\nearrow$ Frequency$\searrow$
- Latency (Cycles) ~ $\boldsymbol{k}^2$

NIST Security Level	n	k	Frequency (MHz)
1	252	126	200
3	400	200	167
5	548	274	142



Improvement Over AVX2 [Level 5]

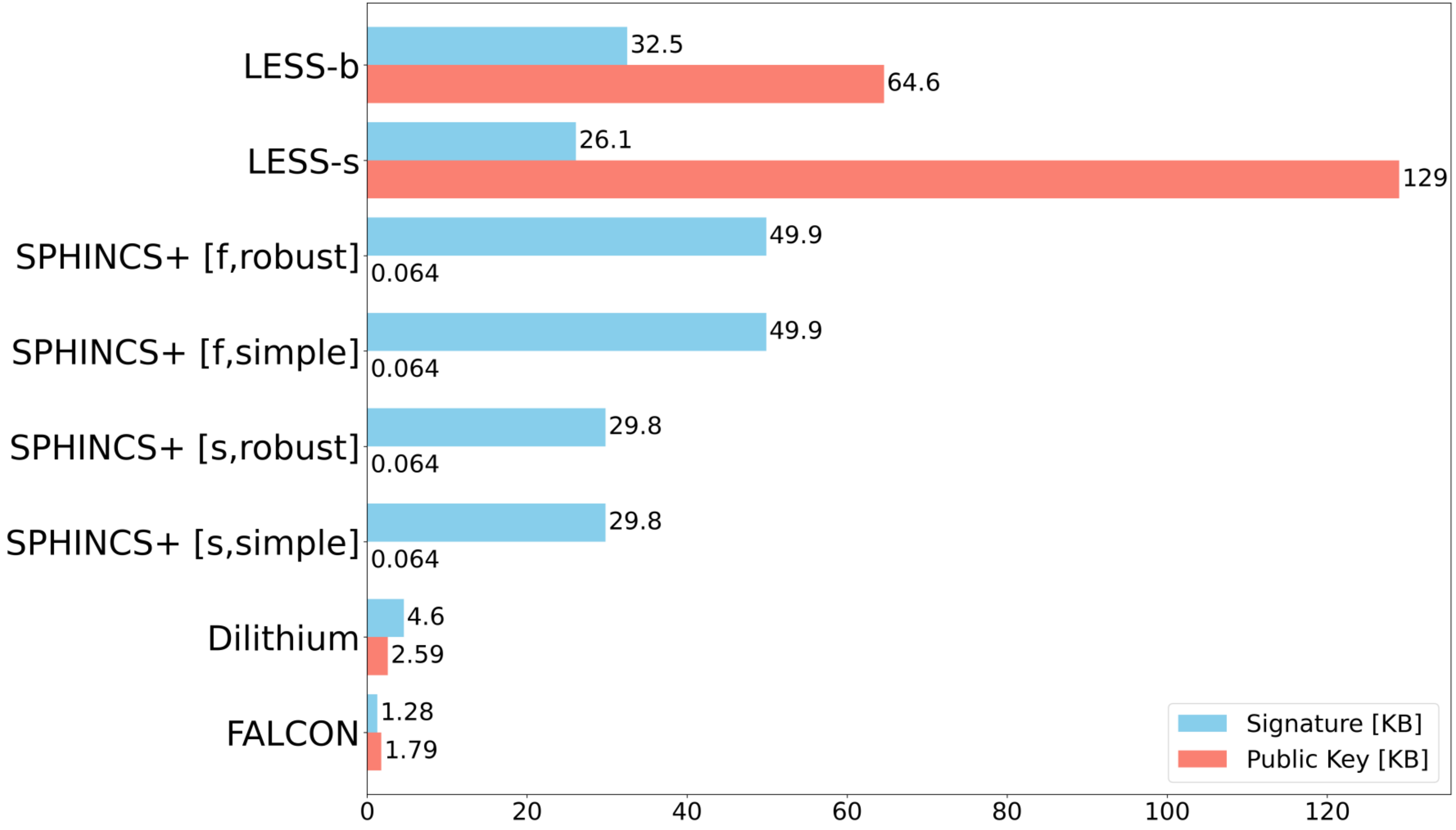


TW (Hardware):
Evaluated on Artix-7

AVX2 (Software):
Evaluated on Ryzen 5 5600G running @3.9 GHz
$27.3\times$ higher frequency than HW

HW is faster by a factor of
- **1.4x** for Keygen
- **2.5x** for Sign and Verify

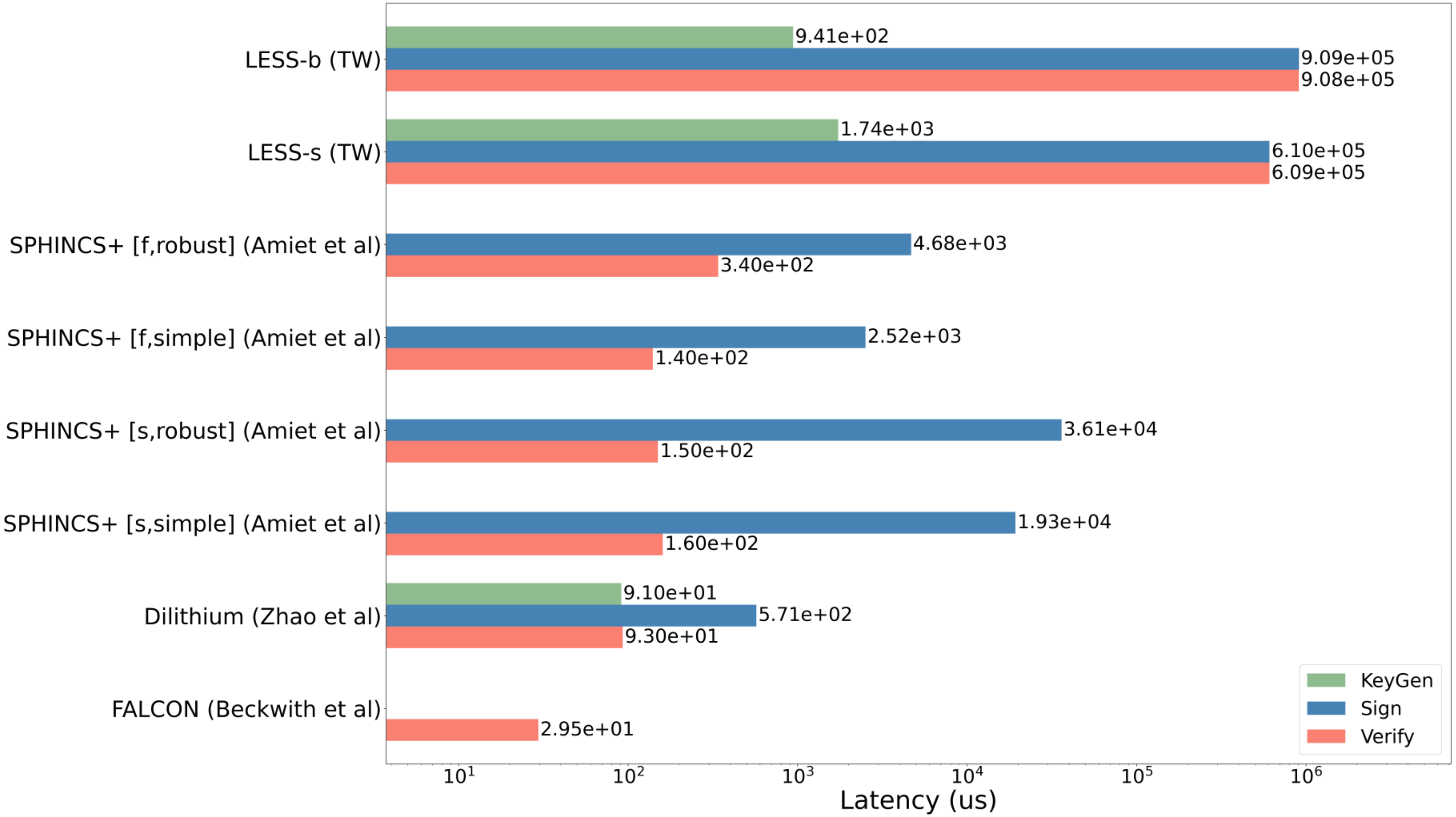
Transmission Cost [Level 5]



Lattices have the smallest sizes

LESS has smaller signatures than SPHINCS$^+$, but much larger public keys

Latency [Level 5]

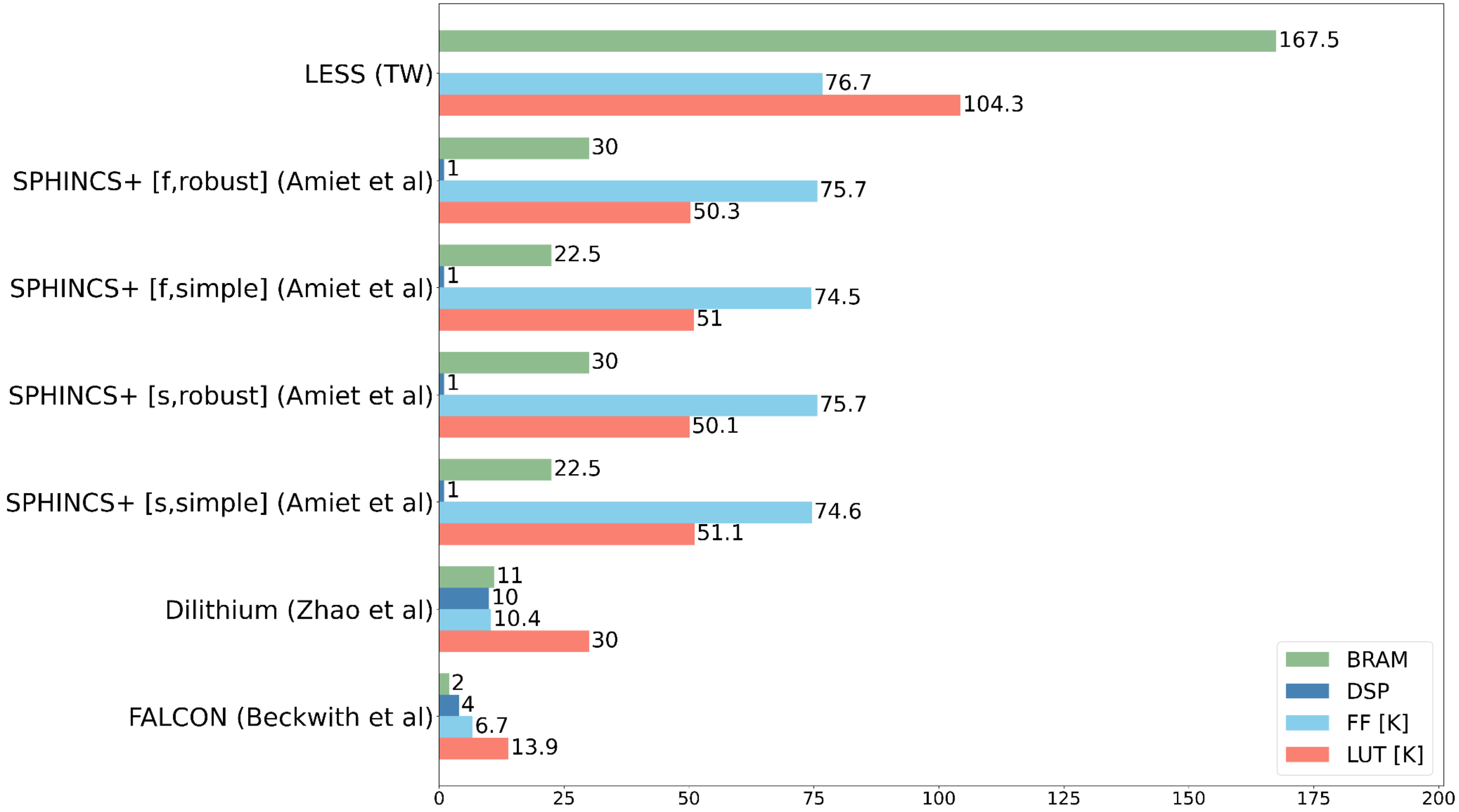


Compared to SPHINCS+

Order of magnitude slower signing

Several orders of magnitude slower for verification

Area [Level 5]



Compared to SPHINCS+

2 × more LUTs

Similar number of DSP/FF

6 × more BRAM

Conclusion

- This work represents the first hardware work on the new candidate LESS
- Our implementation running on an Artix-7 FPGA outperforms optimized AVX2 by $\sim 2\times$
- LESS provides smaller signature sizes than SPHINCS^{+}, but at the cost of larger public keys and slower signing/verification

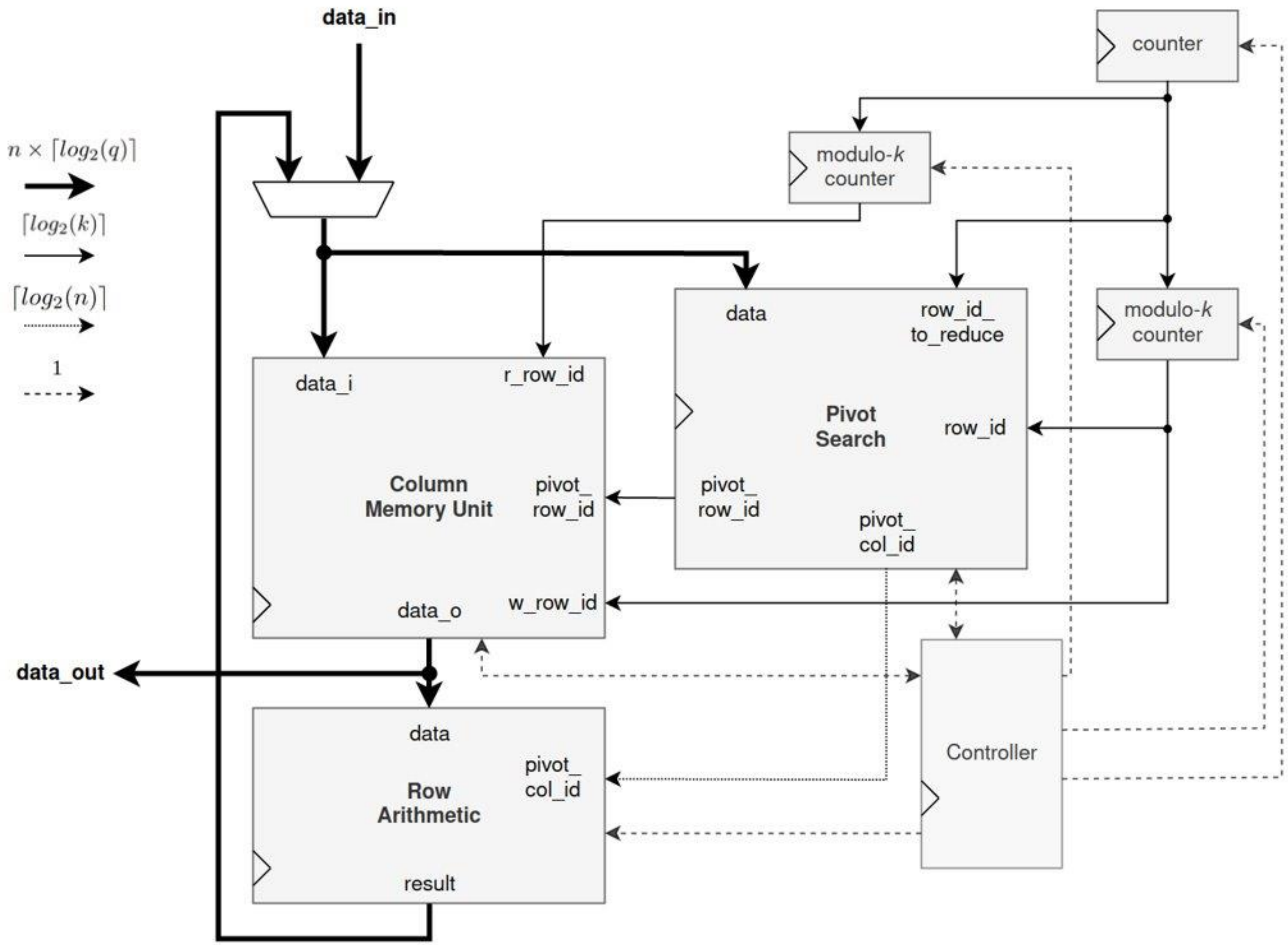
Questions?

https://cryptography.gmu.edu/athena

Page: PQC

RREF – Top-Level Unit

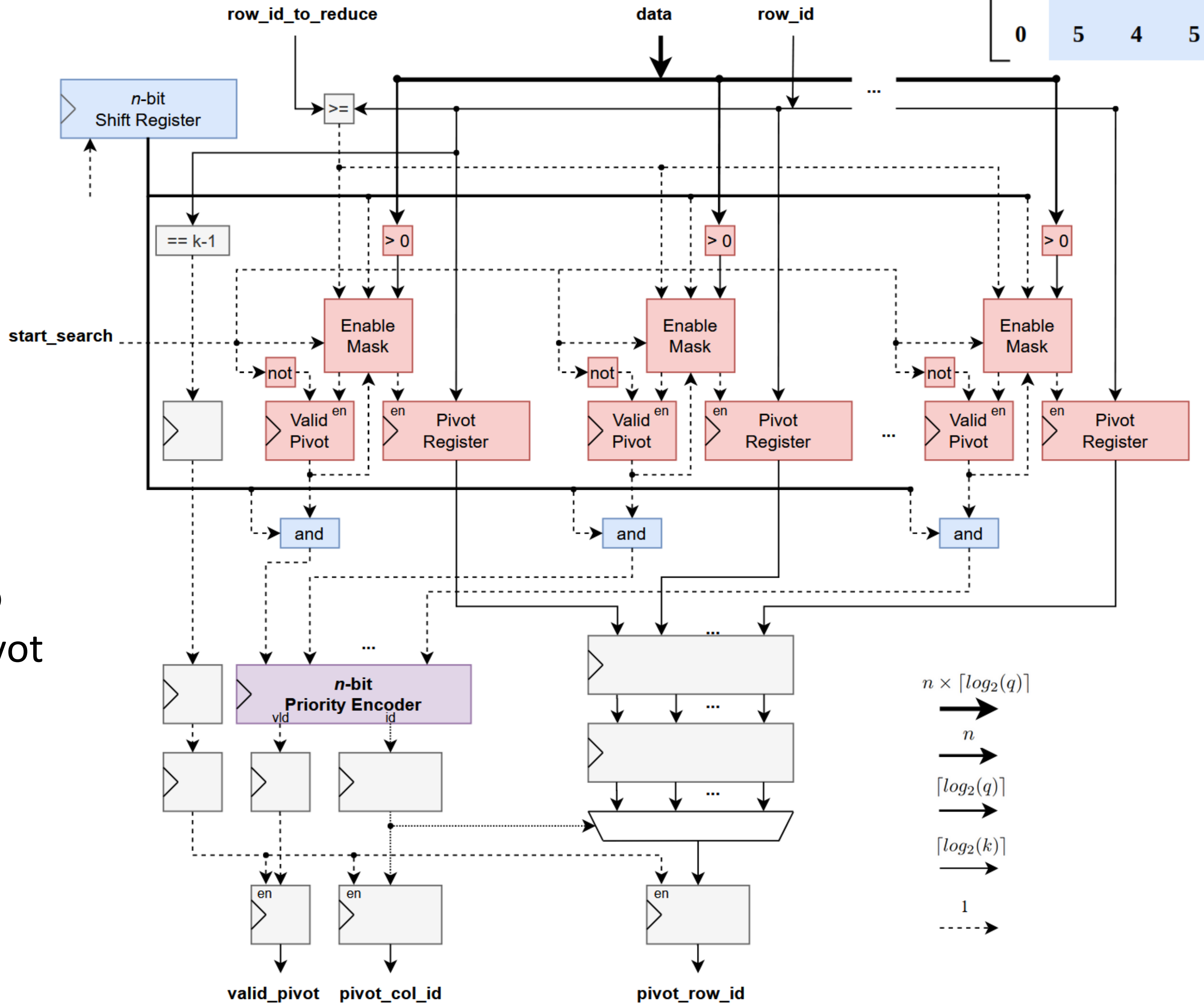
- Operates on entire row of the input matrix at a time
- Constant time *Pivot Search* implementation
- Parallel *Pivot Search* with initialization and *Reduce Other Rows*
- Pipelined arithmetic for increased clock frequency and high throughput



RREF – Pivot Search

$$\begin{bmatrix} 1 & 1 & 5 & 5 & 4 & 2 & 5 \\ 0 & 0 & 0 & 4 & 3 & 5 & 2 \\ 0 & 5 & 4 & 5 & 3 & 6 & 2 \end{bmatrix}$$

- Constant-time search hardware
- **Search area decreases** as algorithm progresses
- **Parallel units** to identify the non-zero element in every column of a row
- **Large priority encoder** to identify "left-most" non-zero element in search area as pivot element



Linear Equivalence Problem

Linear Equivalence Problem (LEP):

Given two matrices $G, G' \in F_q^{k\times n}$ which generate codes C, C', determine if the two corresponding codes are linearly equivalent. That is, does there exist matrices $Q \in M_n$ and $S \in GL(k)$ such that $G' = SGQ$ where $GL(k)$ is the set of invertible matrices.

$$\begin{bmatrix} 0 & 0 & 3 & 0 \\ 0 & 2 & 0 & 0 \\ 2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 4 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 3 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 4 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 3 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix}$$

S

Invertible Matrix

G

Generator Matrix

Q

Monomial Matrix

$G' = SGQ$

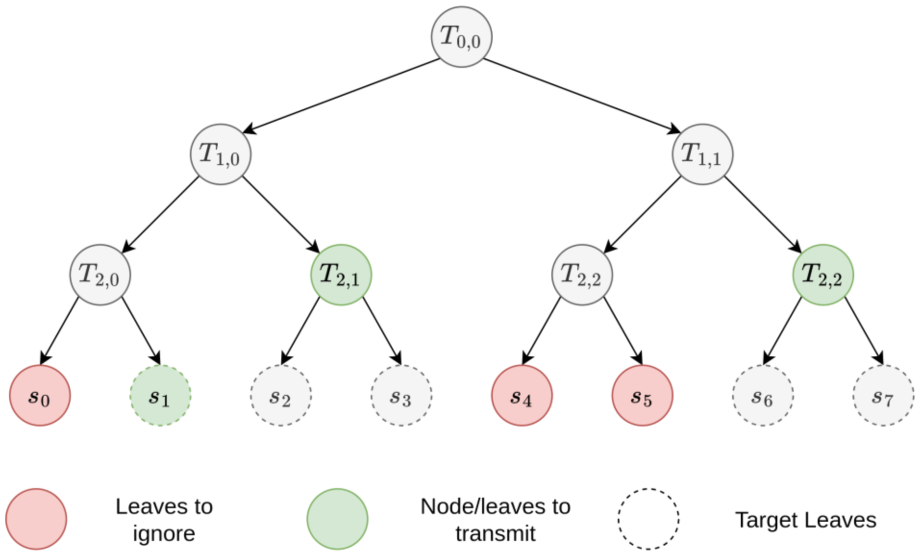
Introduction: LESS Optimizations

Commitment Seed Tree:

- Commitment matrices are sampled using a leaf of a tree as the seed
- Benefit: Rather than sending the seeds of all zero-challenges ($\tilde{Q}$), we can send the path nodes needed to generate them

Information Sets:

- For nonzero-challenges ($Q^{-1} \times \tilde{Q}$), send only the k columns of the monomial which are needed to calculate the pivot columns of the commitment
- Non-pivot columns are minimized and sorted to account for lack of scaling/permuting
- Benefit: Cost of non-zero transmissions is cut in half



Example of path nodes saving transmission cost in seed tree

Computational Bottlenecks:

Conversion to RREF:

- Requires $k^2 * n$ operations
- ~80% of the latency in software

Column Sorting:

- Non-pivot columns are sorted before hashing commitment
- Column-wise sorting requires transposition before and afterwards for optimal performance

Generator Sampling:

- On-the-fly sampling used to reduce BRAM requirement
- K^2 coefficients (up to 75K) coefficients needed

$$\begin{bmatrix} 3 & 3 & 0 & 1 & 0 \\ 0 & 5 & 2 & 0 & 0 \\ 0 & 4 & 0 & 0 & 6 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 & 5 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Conversion to RREF

$$\begin{bmatrix} 0 & 4 & 0 & 1 & 2 \\ 1 & 5 & 1 & 0 & 0 \\ 6 & 4 & 3 & 0 & 6 \\ 8 & 4 & 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 1 & 2 & 4 \\ 1 & 1 & 0 & 0 & 5 \\ 3 & 6 & 0 & 6 & 4 \\ 0 & 8 & 0 & 0 & 4 \end{bmatrix}$$

Column sorted using element-wise comparison

Results

Hardware Comparison Platform:

- Device: Artix-7 FPGAs
- Area: LUTs, FFs, DSP, BRAM
- Performance: Latency in μs

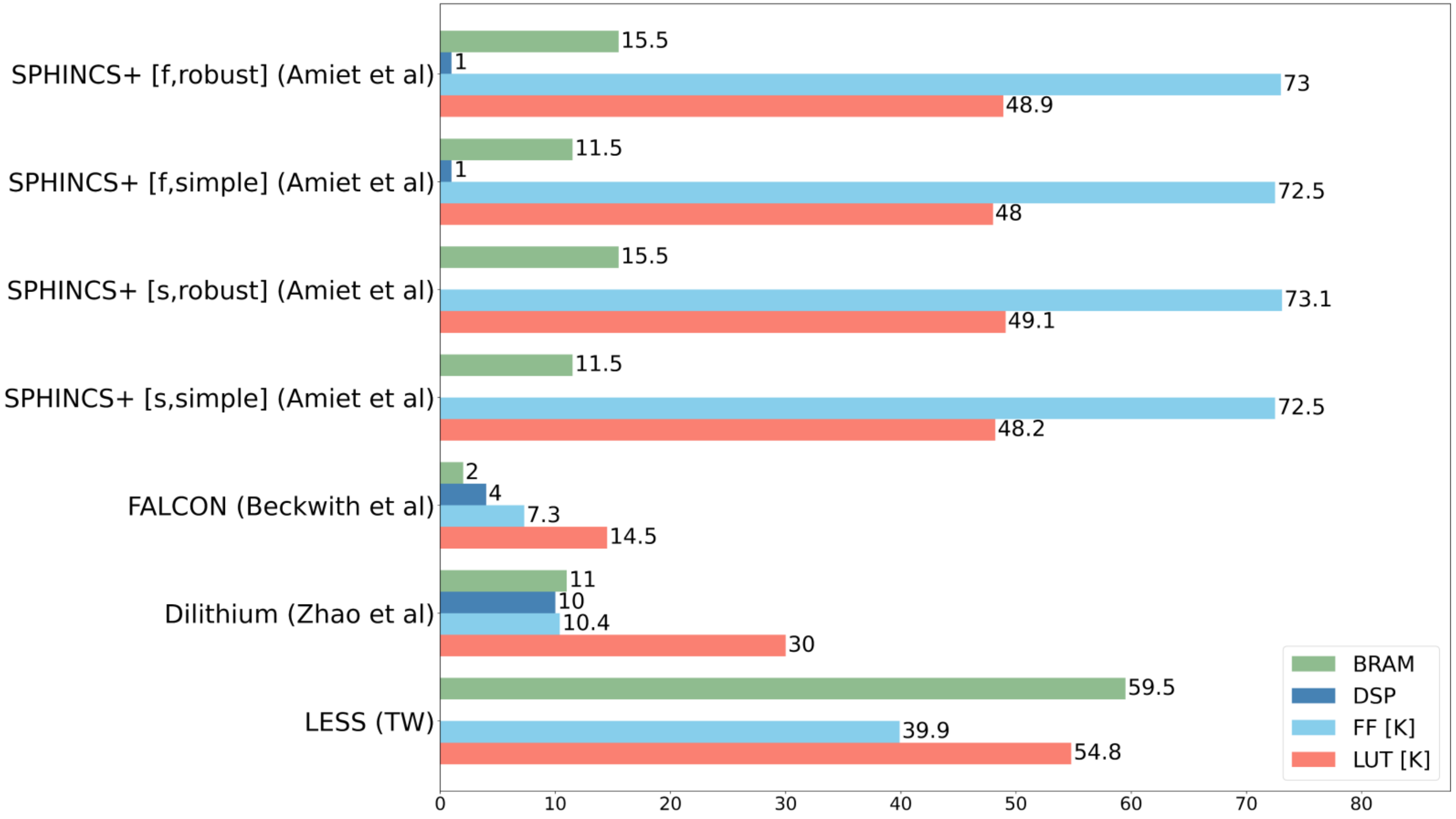
Software Comparison Platform:

- Device: Ryzen 5 5600G
- Implementation: AVX2
- Performance: Latency in μs

Algorithm	Designer	Platform	Parameter Set Selection	Keygen	Sign	Verify
LESS	TW	Artix-7 FPGA	Synthesis	Yes	Yes	Yes
SPHINCS+	Amiet		Synthesis	No	Yes	Yes
Dilithium	Zhao		Runtime	Yes	Yes	Yes
FALCON	Beckwith		Synthesis	No	No	Yes

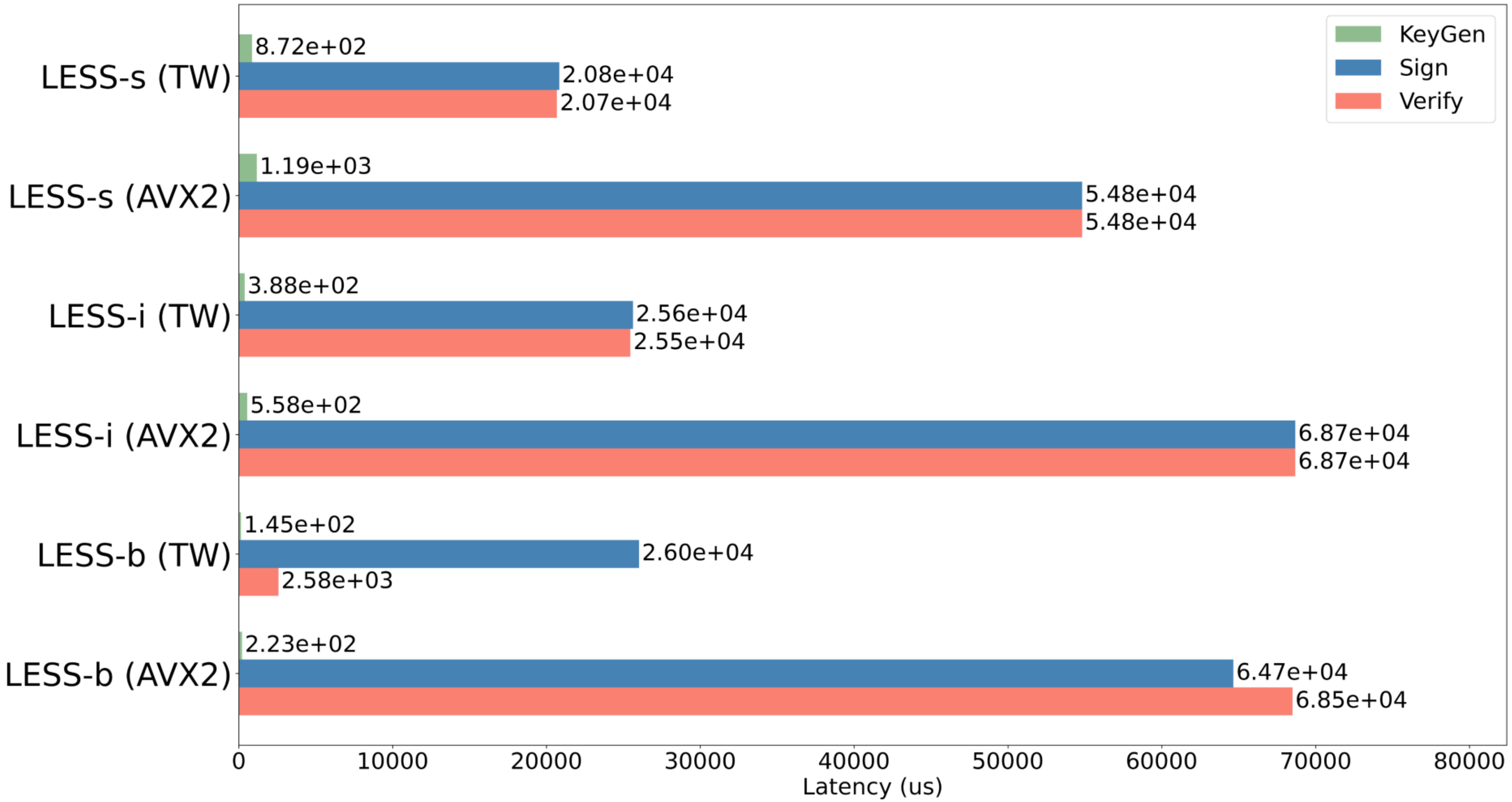
TW → This Work

Area [Level 1]

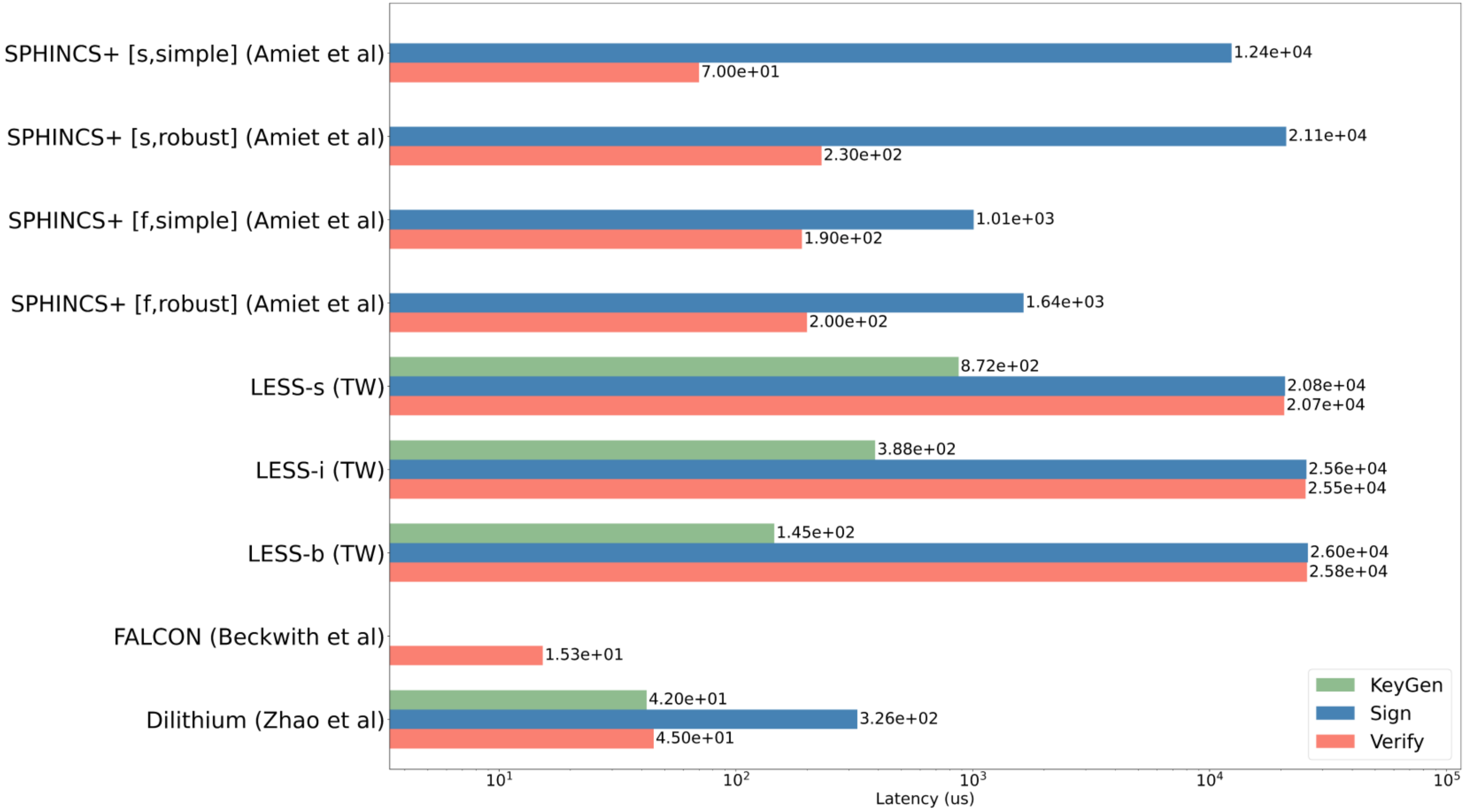


SPHINCS+ [f,robust] (Amiet et al)
15.5
1
73
48.9
SPHINCS+ [f,simple] (Amiet et al)
11.5
1
72.5
48
SPHINCS+ [s,robust] (Amiet et al)
15.5
73.1
49.1
SPHINCS+ [s,simple] (Amiet et al)
11.5
72.5
48.2
FALCON (Beckwith et al)
2
4
7.3
14.5
Dilithium (Zhao et al)
11
10
10.4
30
LESS (TW)
59.5
39.9
54.8
0
10
20
30
40
50
60
70
80
BRAM
DSP
FF [K]
LUT [K]

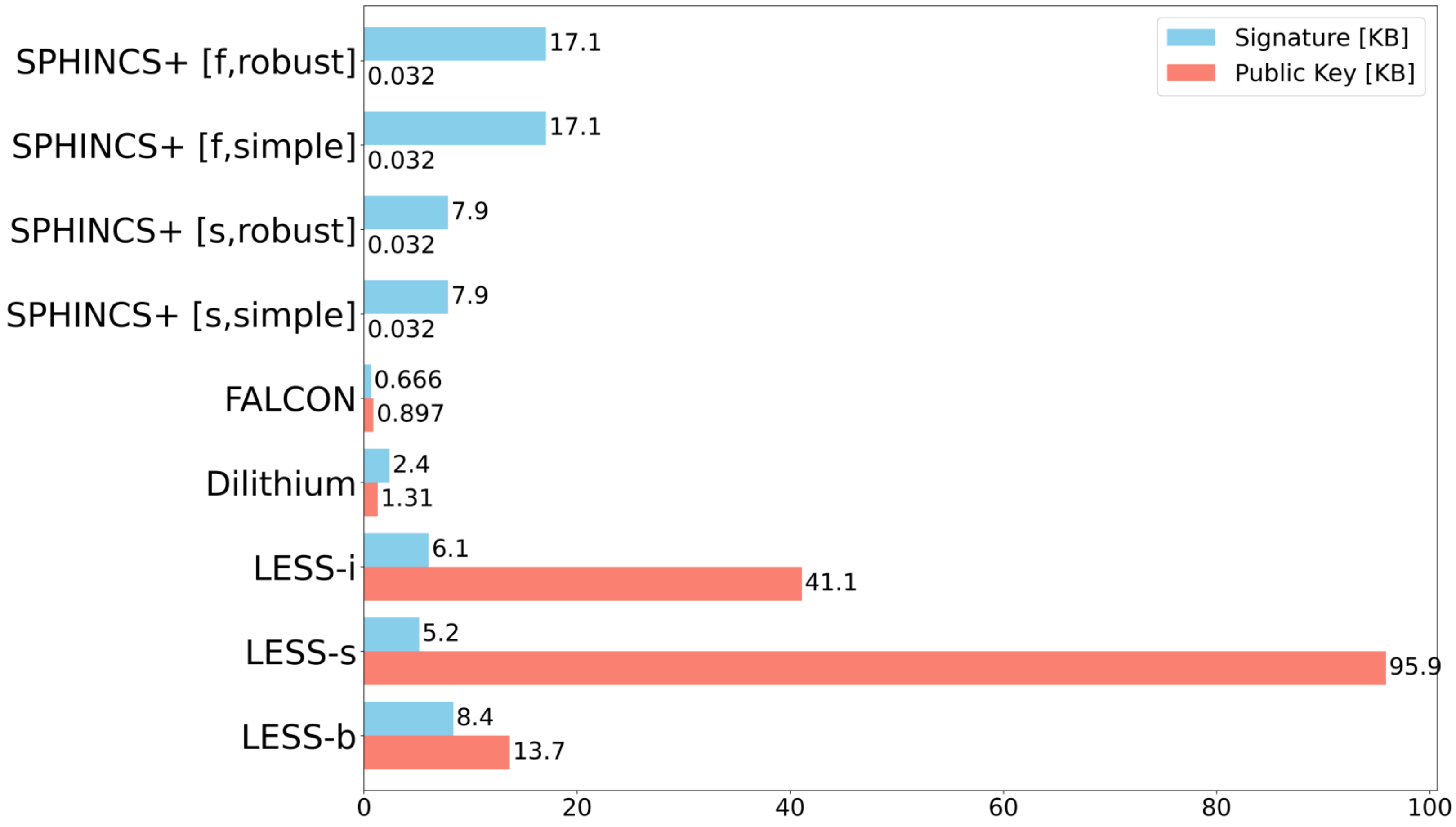
AVX2 Comparison [Level 1]



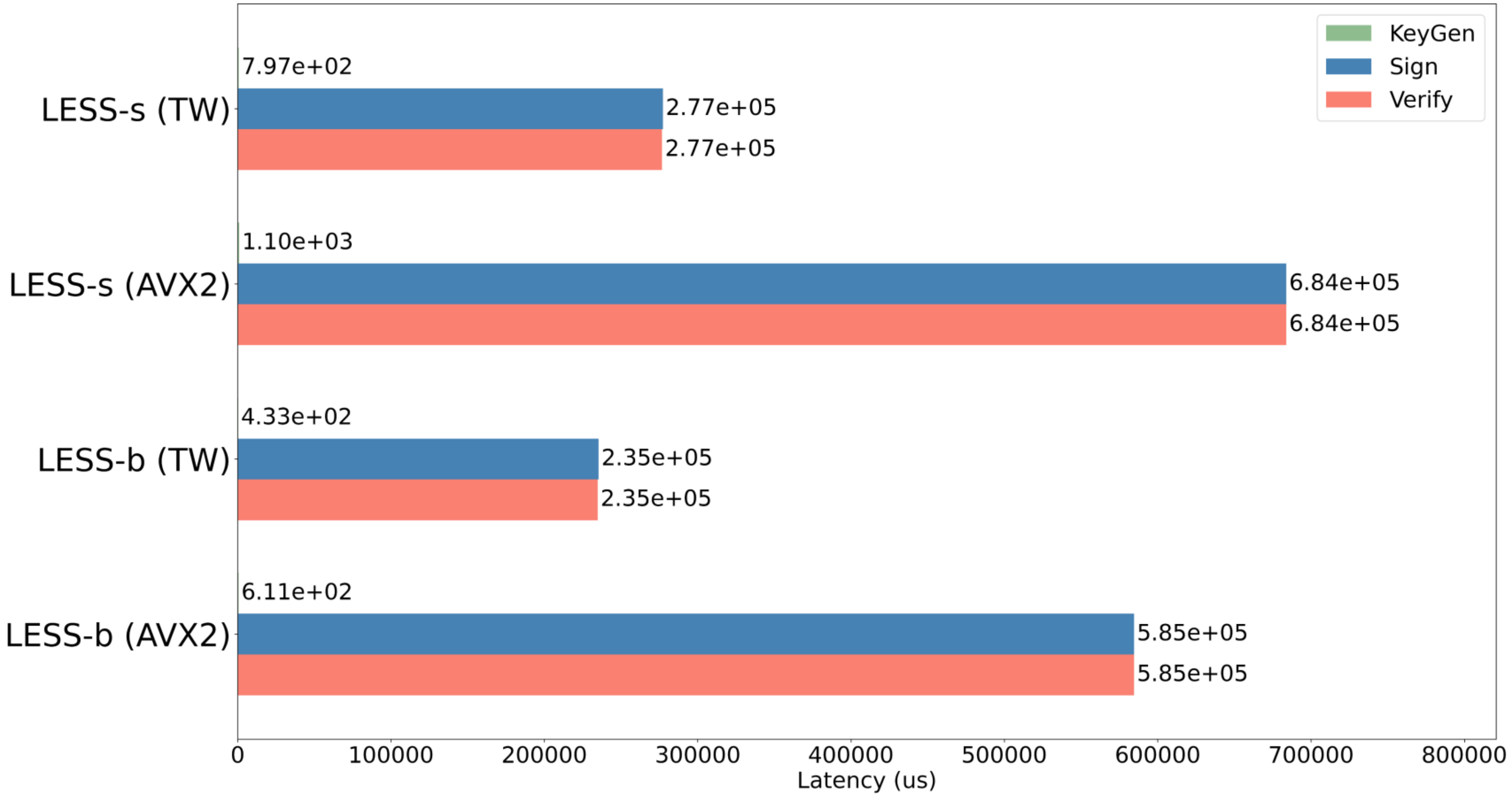
Latency [Level 1]



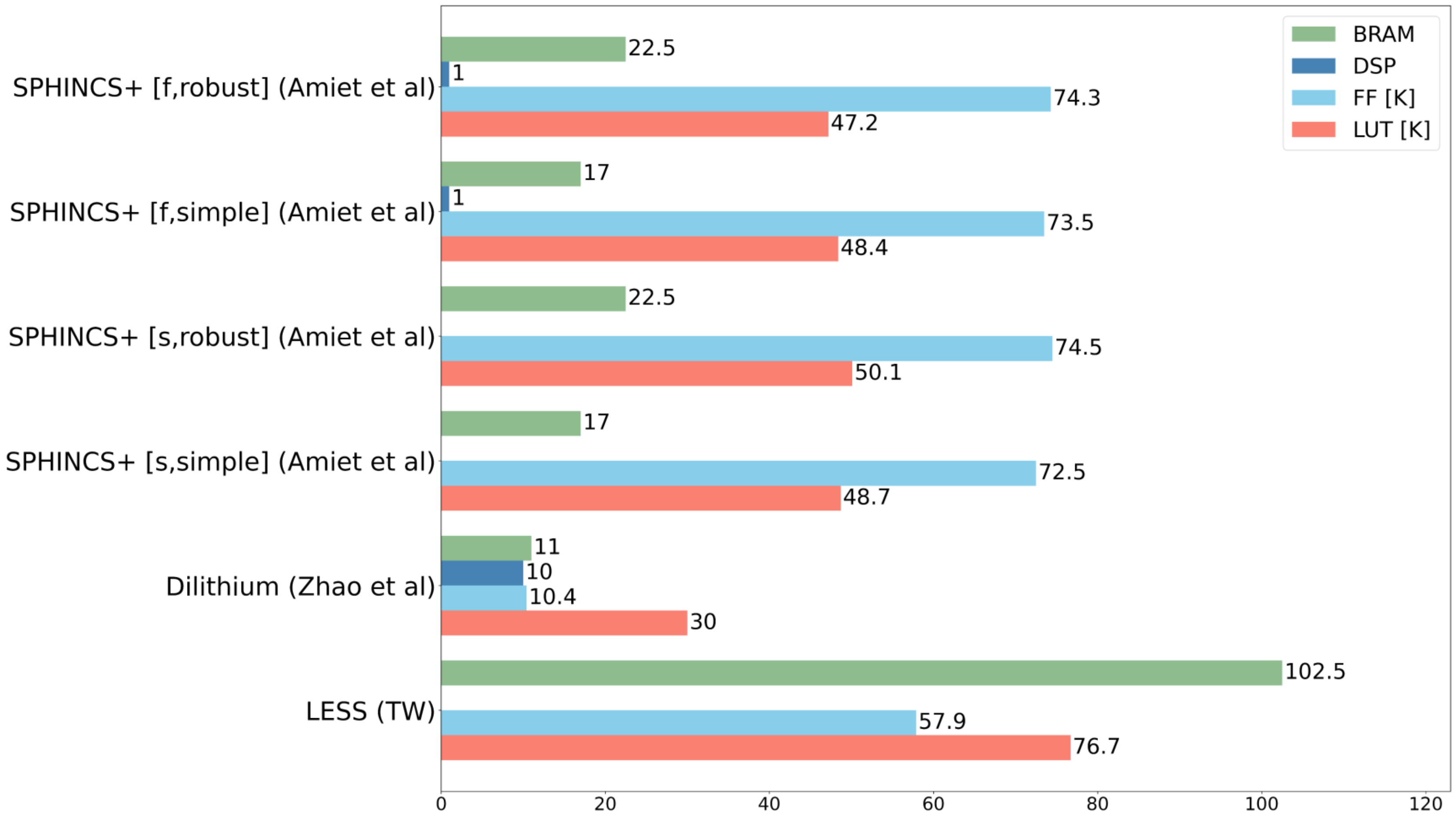
Transmission Cost [Level 1]



AVX2 Comparison [Level 3]

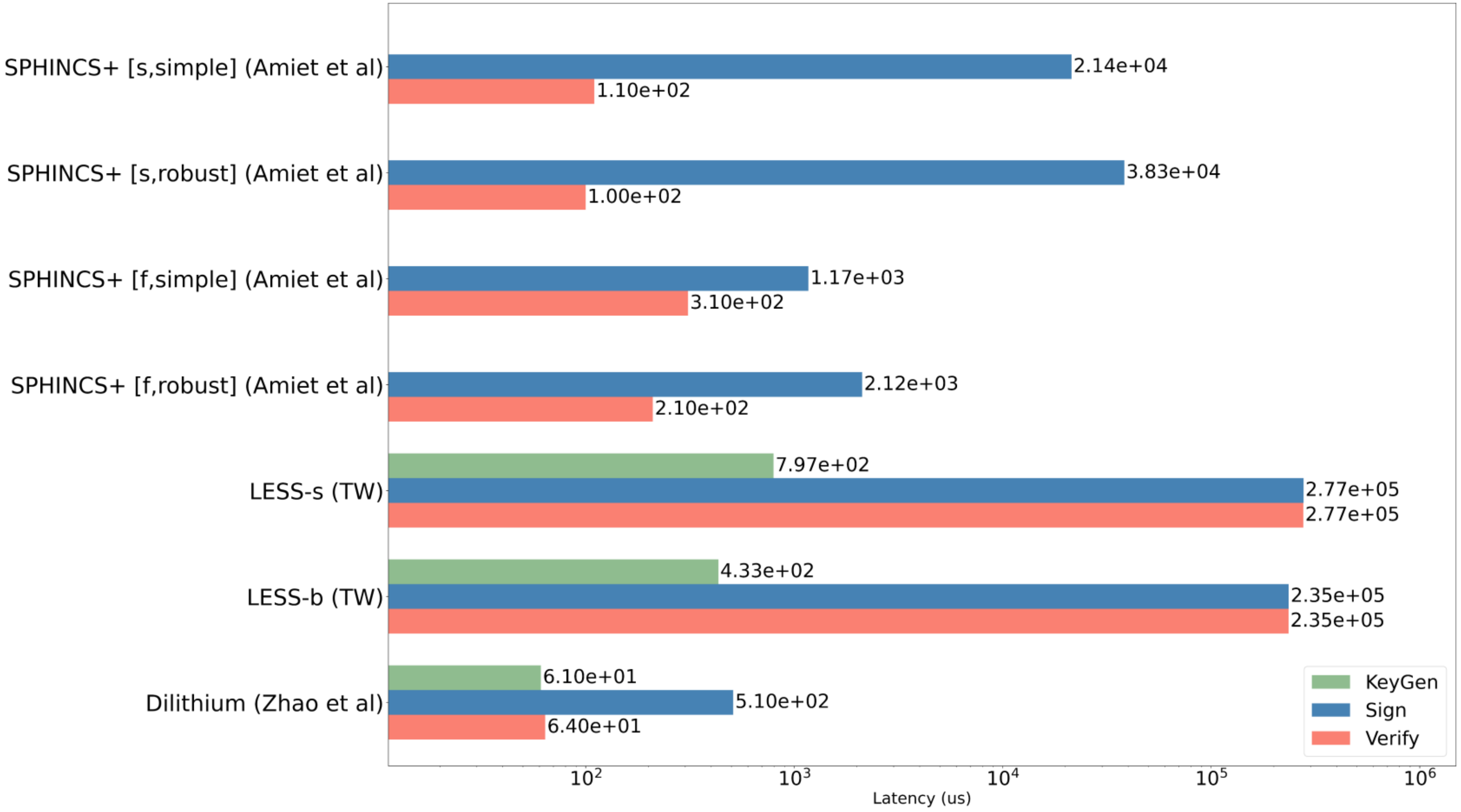


Area [Level 3]



BRAM
DSP
FF [K]
LUT [K]
SPHINCS+ [f,robust] (Amiet et al)
22.5
1
74.3
47.2
SPHINCS+ [f,simple] (Amiet et al)
17
1
73.5
48.4
SPHINCS+ [s,robust] (Amiet et al)
22.5
74.5
50.1
SPHINCS+ [s,simple] (Amiet et al)
17
72.5
48.7
Dilithium (Zhao et al)
11
10
10.4
30
LESS (TW)
102.5
57.9
76.7
0
20
40
60
80
100
120

Latency [Level 3]



Transmission Cost [Level 3]