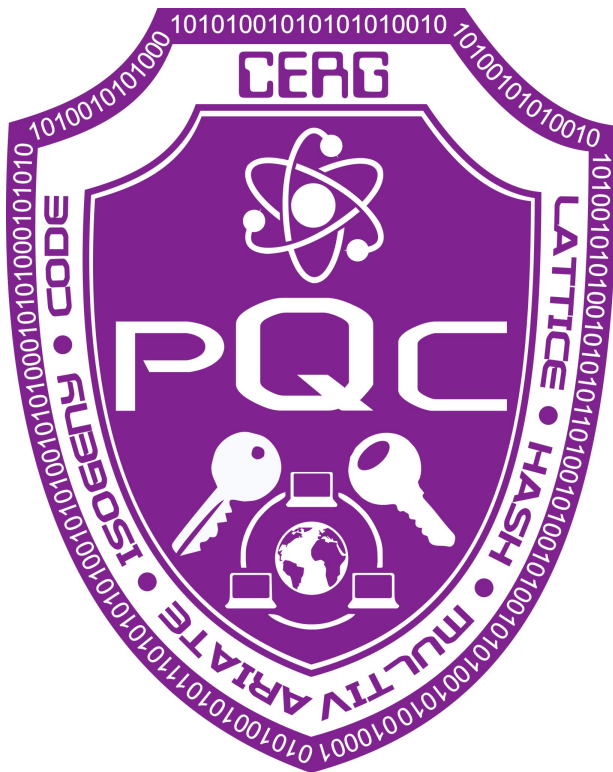
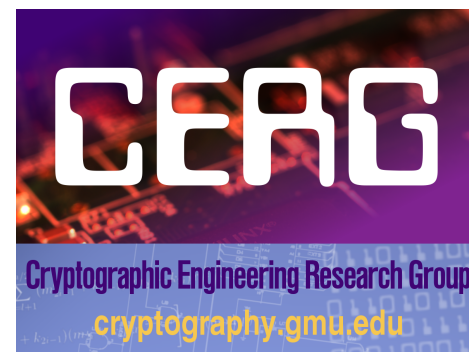


Implementation and Benchmarking of Round 2 Candidates in the NIST Post-Quantum Cryptography Standardization Process Using FPGAs



Kris Gaj

George Mason University



Thank You!

Great thanks to

- Dr. David Hu
- Prof. Ray Cheung

for the kind invitation
to give this talk!

Where is George Mason University?



- East Coast of the U.S.A.
- Near Washington D.C.
- 4 hour drive from New York
- 30 min drive to the Washington Monument, White House, and the U.S. Capitol

Advantages of the Location



National Science Foundation



National Institute of Standards
and Technology



Defense Advanced Research
Projects Agency



Amazon Headquarters 2

CERG: Cryptographic Engineering Research Group



3 faculty members, 8 Ph.D. students,
5 MS students, 7 affiliated scholars

CERG Group Members supporting PQC

Recent Graduate



Farnoud

SW/HW Codesign
RTL Accelerators
Experimental Setup for
Timing Measurements
CAD Tools

PhD Students



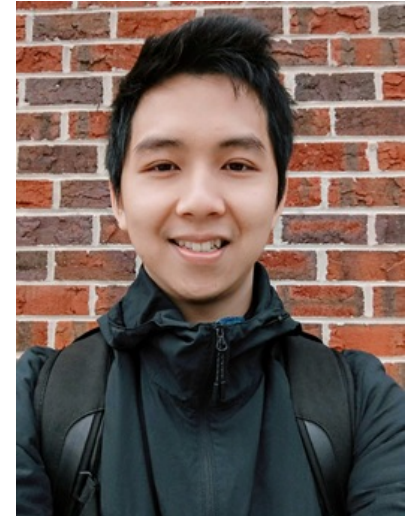
Viet

RTL Design of
HW Accelerators
for Lattice-based
& Code-based PQC



Kamyar

RTL Design of
HW Accelerators
for Lattice-based
PQC
Side-Channel
Analysis
RISC-V Accelerators



Duc

HLS Design of
HW Accelerators
for Lattice-based
PQC
NEON-based SW
implementations

CERG Group Members supporting PQC

PhD Students

Affiliated Scholar

Faculty



Bakry

Experimental Setup
for Side-Channel
Analysis
Lightweight
Architectures



Javad

RTL Design of
HW Accelerators
for Symmetric-based
PQC



Michał

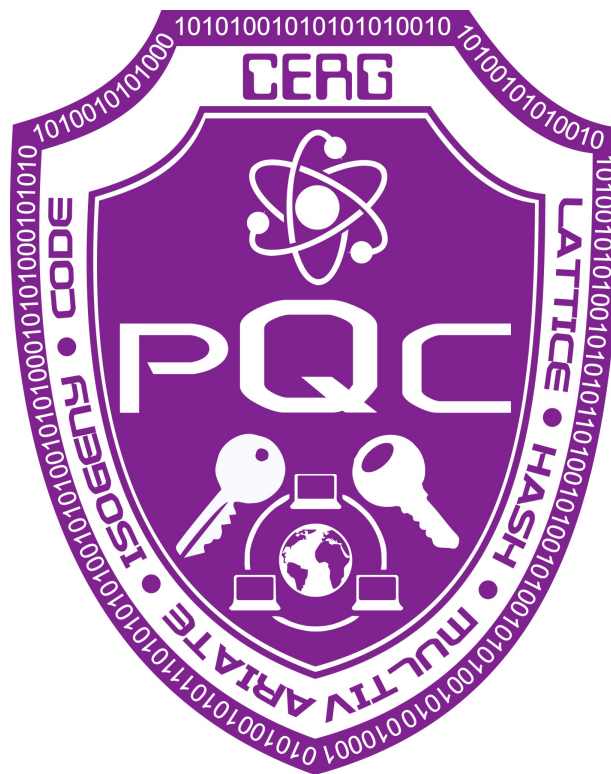
Military University
of Technology in
Warsaw, Poland
RTL Design of
HW Accelerators
for Lattice-based PQC
& Lattice Sieving



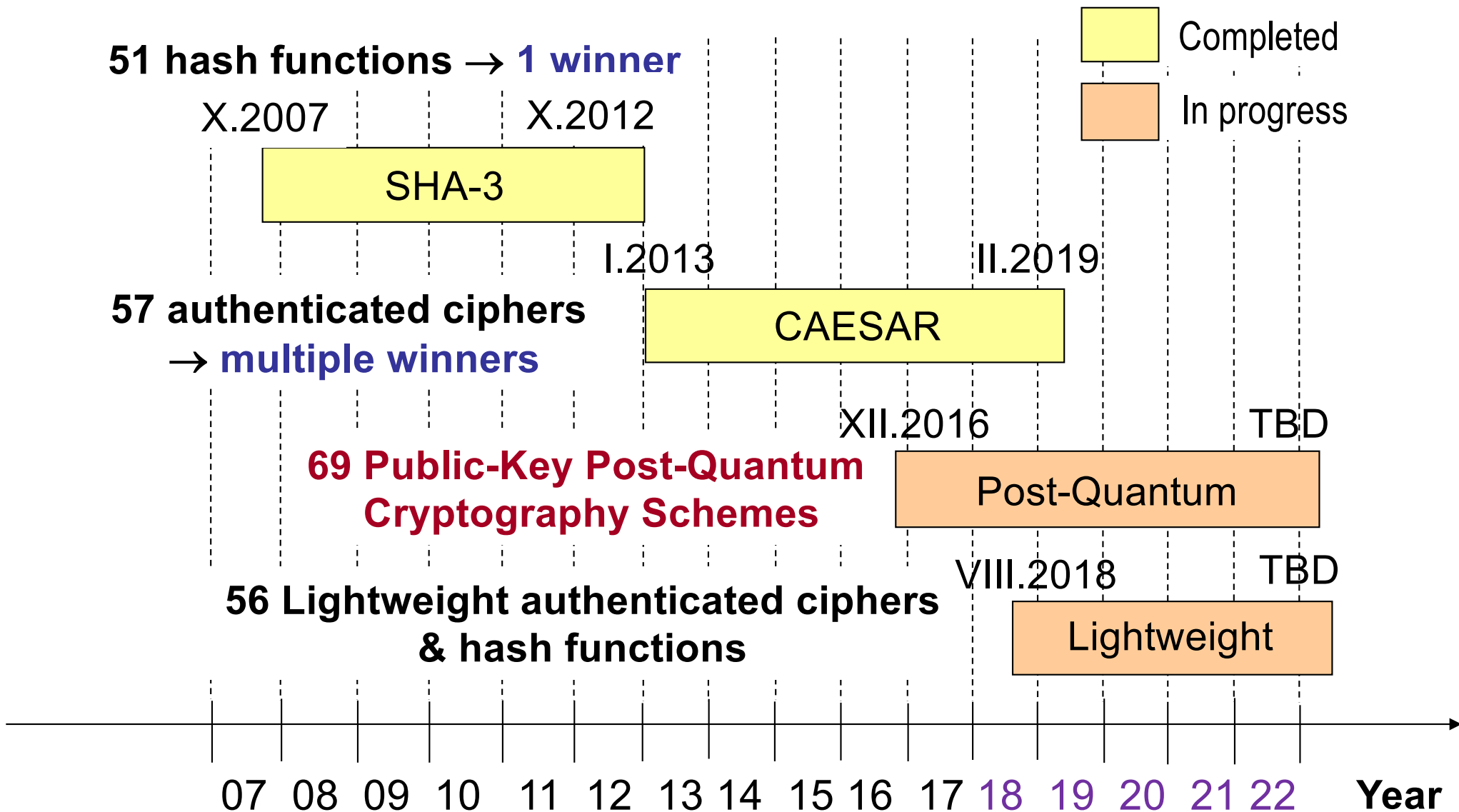
Mike

Sampling
in Hardware

Implementation and Benchmarking of Round 2 Candidates in the NIST Post-Quantum Cryptography Standardization Process Using FPGAs



Cryptographic Contests 2007-Present



Evaluation Criteria

Security

Software Efficiency

μProcessors **μControllers**

Hardware Efficiency

FPGAs **ASICs**

Flexibility

Simplicity

Licensing

U.S. Open of Ciphers



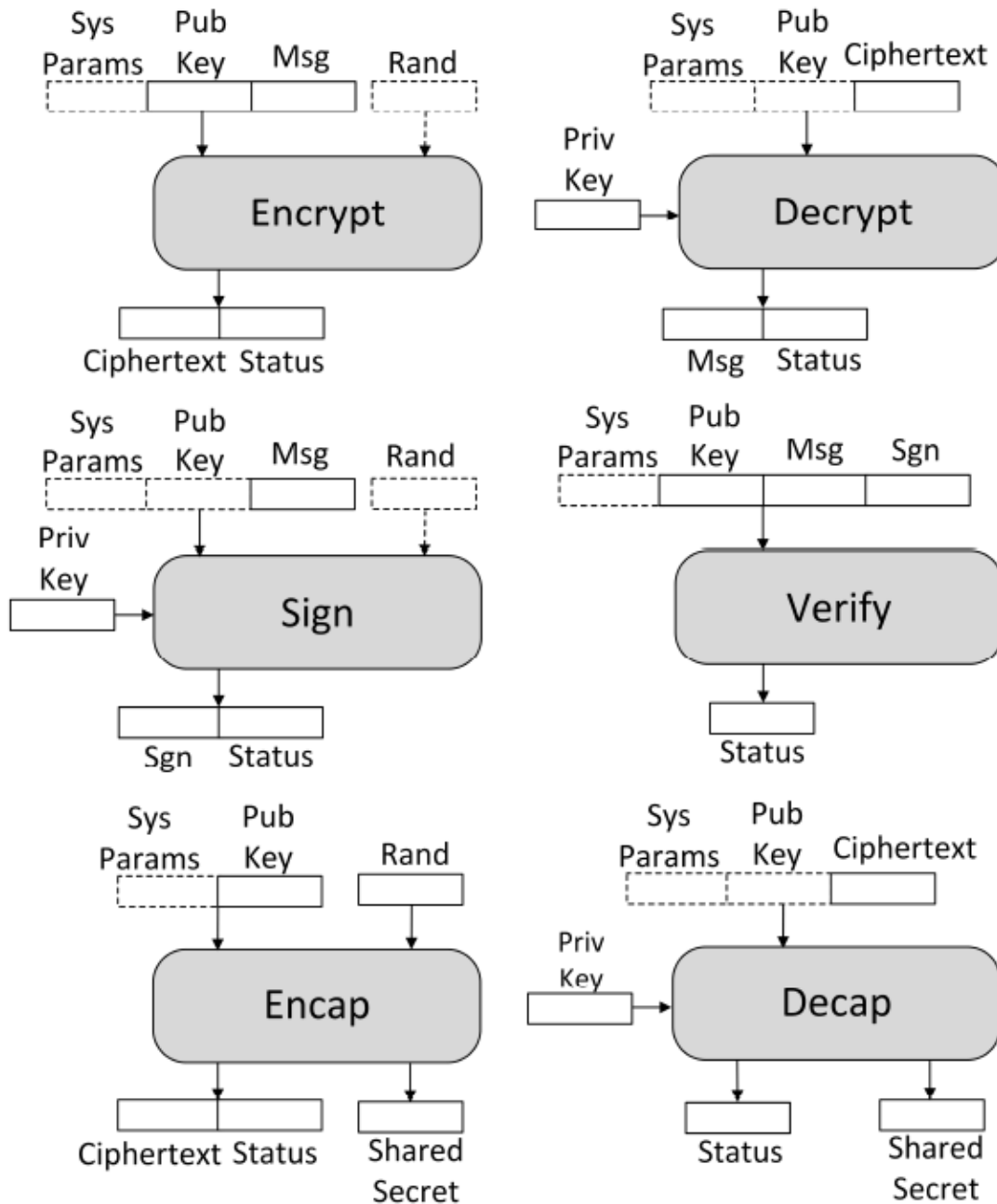
NIST PQC Standardization Process

- **Feb. 2016:** NIST **announcement of standardization plans** at PQCrypto 2016, Fukuoka, Japan
- **Dec. 2016:** NIST **Call for Proposals** and Request for Nominations for Public-Key Post-Quantum Cryptographic Algorithms
- **Nov. 30, 2017:** **Deadline** for submitting candidates
- **Dec. 2017:** Announcement of the **First Round Candidates**
- **Apr. 2018:** The First **NIST PQC Standardization Conference**
- **Nov. 30, 2018:** Deadline for **mergers of similar submissions**
- **Jan. 30, 2019:** Announcement of **candidates qualified to Round 2**

NIST PQC Standardization Process

- **Mar. 15, 2019:** Deadline to submit **tweaks** for Round 2 candidates
 - **April 10, 2019:** Publication of **Round 2 submission packages**
 - **Aug. 22-24, 2019:** Second **NIST PQC Conference**
 - **April 15, 2020:** **Deadline** to submit comments
 - **July 22, 2020:** Announcement of **Round 3** 7 finalists and 8 alternate candidates
 - **July 29, 2020:** **NSA's** Cybersecurity **Perspective**
 - **Spring 2021:** Third **NIST PQC Conference**
 - **2022-2023:** **Draft standard(s)** released for public comments
 - **2024:** First PQC **standard(s)** published
- Focus of this talk
- Reality Check
- Gazing the PQC Crystal Ball

Three Types of PQC Schemes

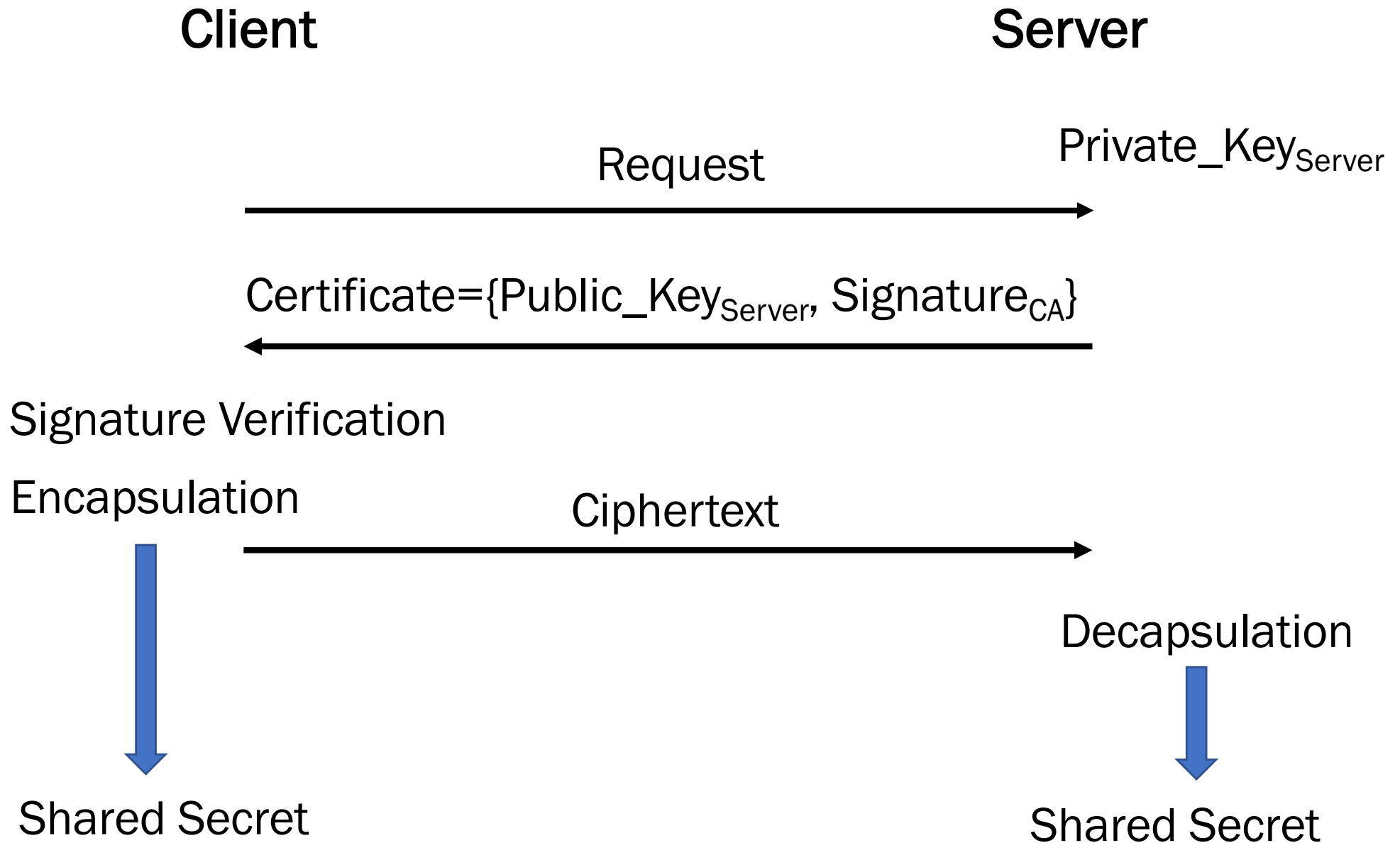


1. Public Key Encryption (PKE)

2. Digital Signature (DS)

3. Key Encapsulation Mechanism (KEM)

Key Establishment Using Long-Term Keys



Five Security Levels

Level	Security Description
1	At least as hard to break as AES-128 using exhaustive key search
2	At least as hard to break as SHA-256 using collision search
3	At least as hard to break as AES-192 using exhaustive key search
4	At least as hard to break as SHA-384 using collision search
5	At least as hard to break as AES-256 using exhaustive key search

Leading PQC Families

Family	Encryption/ KEM	Signature
Symmetric-based		XX
Code-based	XX	X
Lattice-based	XX	X
Multivariate	X	XX
Isogeny-based	X	

XX – high-confidence candidates

X – medium-confidence candidates

Round 1 Candidates

82 submissions, 69 accepted as complete, 5 officially withdrawn
25 Countries, 6 continents, 256 co-authors

Family	Signature	Encryption/KEM	Overall
Lattice-based	5	21	26
Code-based	2	17	19
Multivariate	7	2	9
Symmetric-based	3		3
Isogeny-based		1	1
Other	2	4	6
Total	19	45	64

Round 1 Submissions



12 considered broken, 8 in need of serious tweaks

BIG QUAKE. BIKE. **CFPKM**. Classic McEliece. **Compact LWE**. CRYSTALS-DILITHIUM. CRYSTALS-KYBER. **DAGS**. Ding Key Exchange. **DME**. DRS. DualModeMS. **Edon-K**. EMBLEM and R.EMBLEM. FALCON. FrodoKEM. GeMSS. **Giophantus**. Gravity-SPHINCS. **Guess Again**. Gui. **HILA5**. HiMQ-3. **HK17**. HQC. KINDI. LAC. LAKE. **LEDAkem**. **LEDApkc**. **Lepton**. LIMA. Lizard. LOCKER. LOTUS. LUOV. **McNie**. Mersenne-756839. MQDSS. NewHope. NTRUEncrypt. NTRU-HRSS-KEM. NTRU Prime. NTS-KEM. Odd Manhattan. OKCN/AKCN/CNKE. Ouroboros-R. Picnic. pqNTRUSign. pqRSA encryption. pqRSA signature. **pqsigRM**. QC-MDPC KEM. qTESLA. **RaCoSS**. Rainbow. Ramstake. **RankSign**. **RLCE-KEM**. Round2. RQC. **RVB**. SABER. SIKE. SPHINCS+. **SRTPI**. Three Bears. Titanium. **WalnutDSA**.

Some attack scripts already posted causing **total break** or **serious tweaks**. Many more receiving detailed analysis.

Sources: Lange, ICMC **May 2018** & pqc-comments@nist.gov

Round 2 Candidates

26 Candidates announced on January 30, 2019

Family	Signature	Encryption/KEM	Overall
Lattice-based	3	9	12
Code-based		7	7
Multivariate	4		4
Symmetric-based	2		2
Isogeny-based		1	1
Total	9	17	26

Round 2 Submissions (announced Jan. 30, 2019)

• Encryption/KEMs (17)

- CRYSTALS-KYBER
- FrodoKEM
- LAC
- NewHope
- NTRU (merger of NTRUEncrypt/NTRU-HRSS-KEM)
- NTRU Prime
- Round5 (merger of Hila5/Round2)
- SABER
- Three Bears

9

- BIKE
- Classic McEliece
- HQC
- LEDAcrypt (merger of LEDAkem/pkc)
- NTS-KEM
- ROLLO (merger of LAKE/LOCKER/Ouroboros-R)
- RQC

7

- SIKE

1

- Lattice-based
- Code-based
- Isogenies

▪ Digital Signatures (9)

- CRYSTALS-DILITHIUM
- FALCON
- qTESLA

3

- GeMSS
- LUOV
- MQDSS
- Rainbow

4

- Lattice-based
- Symmetric-based
- Multivariate

- Picnic
- SPHINCS+

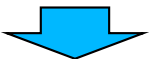




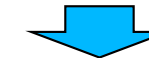



2

NIST Report on the 1st Round: <https://doi.org/10.6028/NIST.IR.8240>



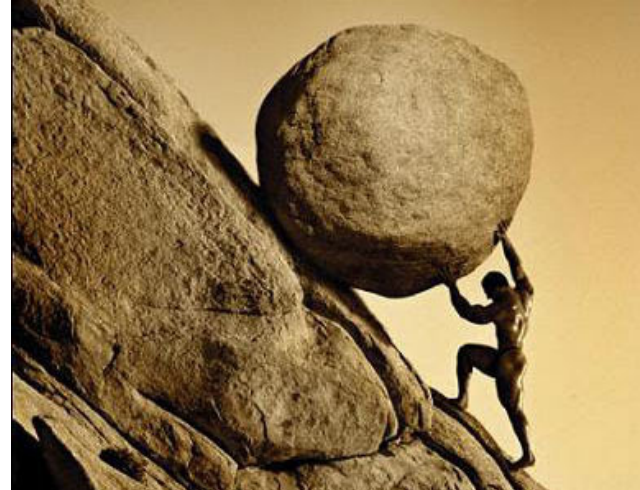
Hardware Benchmarking

Round 2 Candidates in Hardware

	#Round 2 candidates	Implemented in hardware	Percentage
AES	5	5	100%
			
SHA-3	14	14	100%
			
CAESAR	29	28	97%
			
PQC	26	17	65%

Challenges of Post-Quantum Cryptography

- Mathematical complexity
 - Large amount of man-power
 - New types of basic operations
 - Need for random sampling not only from uniform but also from discrete Gaussian and/or other distributions
 - Constant-time implementations
 - Hardware resources required
-
- Need for new SCA (Side-Channel Attack) countermeasures against power and electromagnetic analysis
 - Plug-and-play replacement for current public-key cryptography units
 - Intermediate use of hybrid systems



Major Optimization Targets



High-Speed

- Parallel processing
- Constant-time
- Parametric code



Lightweight

- Small area, power, energy per operation
- Resistance to power & electromagnetic analysis

Lattice-Based KEMs in Pure Hardware

	High-Speed	Lightweight
KYBER	H: GMU, USA	
FrodoKEM	H: PQShield/Bristol, UK + ALaRI, Switzerland	
LAC	H: GMU, USA	
NewHope	H: Tsinghua, China H: IIIT Delhi & IIT Ropar, India + NTU, Singapore & Fraunhofer, Singapore H: GMU, USA	
NTRU		
NTRUPrime		
Round5	H: MUT, Warsaw, Poland & GMU, USA	H: MUT, Warsaw, Poland
SABER	H: U. Birmingham, UK	
Three Bears		

Lattice-Based KEMs: HW & SW/HW

	High-Speed	Lightweight
KYBER	H: GMU, USA SH: Fudan U., China; (VPQC)	SH: Fraunhofer SIT, Darmstadt, Germany SH: TUM/Airbus, Germany (RISQ-V)
FrodoKEM	H: PQShield/Bristol, UK + ALaRI, Switzerland SH: GMU, USA	SH: MIT, USA (Sapphire)
LAC	H: GMU, USA SH: Fudan U., China (VPQC)	
NewHope	H: Tsinghua, China H: IIIT Delhi/IIT Ropar, India + NTU/Fraunhofer Singapore H: GMU, USA SH: TUM, Germany + Delft, the Netherlands SH: Fudan U., China (VPQC)	SH: MIT, USA (Sapphire) SH: Fraunhofer SIT, Darmstadt, Germany SH: TUM/Airbus, Germany (RISQ-V)
NTRU	SH: GMU, USA	
NTRUPrime	SH: GMU, USA	
Round5	H: MUT, Warsaw, Poland + GMU, USA	H: MUT, Warsaw, Poland
SABER	HW: U. Birmingham, UK SH: KU Leuven, Belgium + U. Birmingham, UK SH: GMU, USA	SH: TUM/Airbus, Germany (RISQ-V)
Three Bears		

Isogeny-Based and Code-Based KEMs

	High-Speed	Lightweight
Isogeny-Based		
SIKE	H: FAU & USF, USA SH: Radboud U., the Netherlands + Microsoft Research, USA H: FAU & USF, USA	SH: Radboud U., the Netherlands + Microsoft Research, USA
Code-Based		
BIKE	H: NTU, Singapore + Yale U., USA + CUHK, Hong Kong (key generation) H: Intel, USA (decoder) H: R-U Bochum, Germany	
Classic McEliece/ NTS KEM	H: Yale U., USA + Fraunhofer SIT, Darmstadt, Germany	
LEDACrypt		H: NTU, Singapore + Marche Polytechnic U., Italy
ROLLO		
RQC		

Digital Signatures

	High-Speed	Lightweight
Lattice-Based		
DILITHIUM		SH: MIT, USA
FALCON		
qTESLA		SH: MIT, USA SH: Yale U., USA + MAN T&B SE, Germany + U. Waterloo, Canada + Microsoft Research, USA
Symmetric-Based		
Picnic	H: Graz U.T., Austria + AIT, Vienna, Austria	
SPHINCS+		
Multivariate		
GeMSS		
LUOV		
MQDSS		
Rainbow	H: GMU, USA	

NewHope and CRYSTALS-KYBER

Feature	NewHope	CRYSTALS-KYBER
Underlying Problem	Ring-LWE	Module-LWE
Security Levels	lattice dimension = n L1: $n=512$, L5: $n=1024$	$n=256$, lattice dimension = $k \cdot n$ L1: $k=2$, L3: $k=3$, L5: $k=4$
Modulus q	Prime 12,289	Prime 3,329
Required Hash-based Functions	SHAKE128, SHAKE256	SHAKE128, SHAKE256 SHA3-256, SHA3-512
Sampling	CBD*	CBD*
# Poly-Mult in Encaps	2	$k^2 + k$
# Poly-Mult in Decaps	3	$k^2 + 2k$

* Centered Binomial Distribution (CBD)

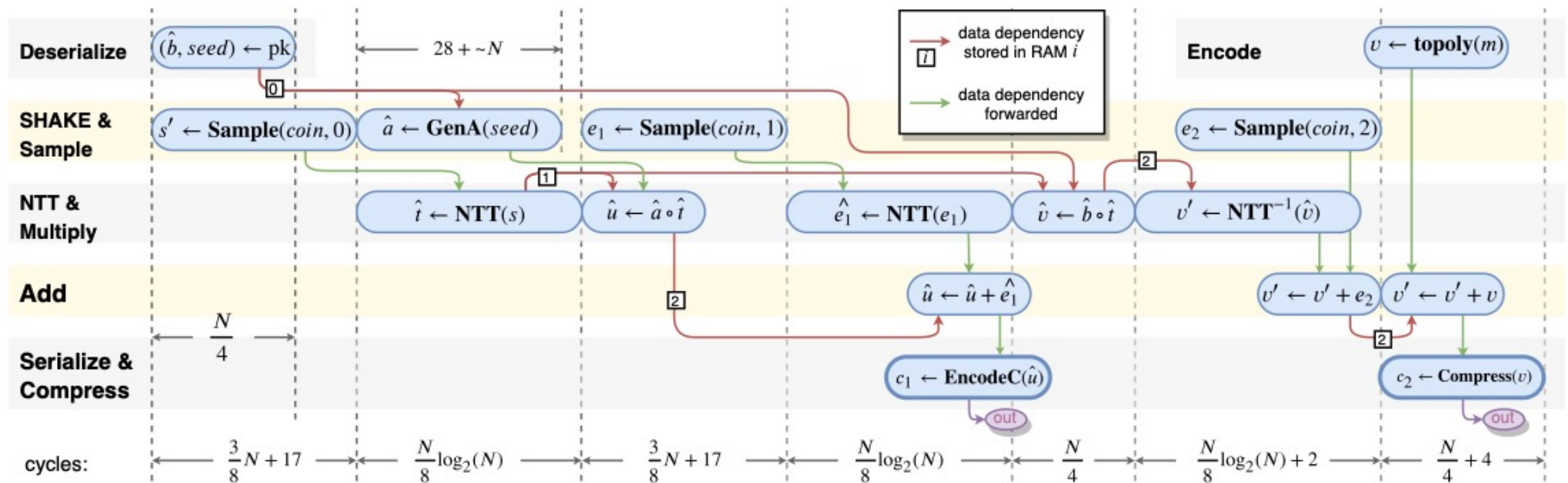
LAC and Round5

Feature	LAC (v3a/v3b)	Round5 (0d/5d)
Underlying Problem	Ring-LWE	Ring-LWR
Error Correcting Code	BCH	None / XEf
Security Levels	lattice dimension = n L1: n=512, L3: n=1024, L5: n=1024	lattice dimension = n L1: n=586/508 L3: n=852/756 L5: n=1170/946
Modulus q	Prime 251 / 256	L1: $2^{13}/2^{10}$, L3: $2^{12}/2^{12}$ L5: $2^{13}/2^{11}$
Required Hash-based Functions	Left up to implementers	L1: SHAKE128, L3, L5: SHAKE256
Sampling	n-ary CBD with fixed Hamming weight	uniform
# Poly-Mult in Encaps	2	2
# Poly-Mult in Decaps	3	3

Common Optimization Method

- Efficient hardware scheduling to perform operations without data dependency in parallel

NewHope Encryption

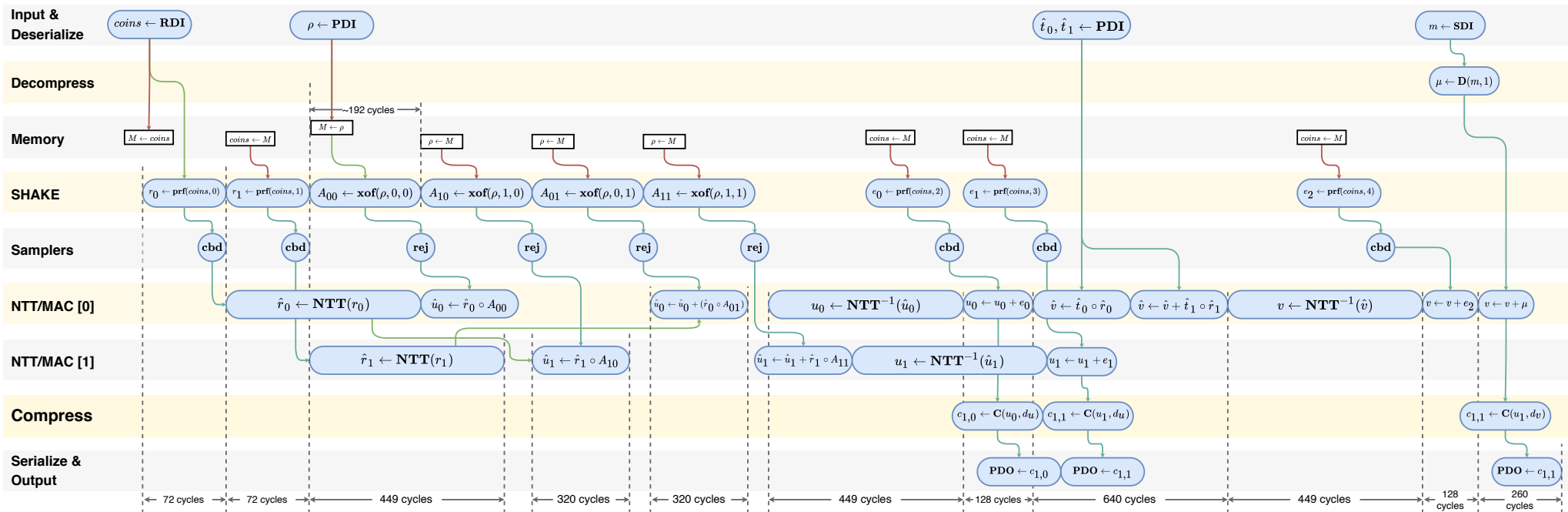


Common Optimization Method



Efficient hardware scheduling to perform operations without data dependency in parallel

CRYSTALS-KYBER Encryption (Security Level 1)

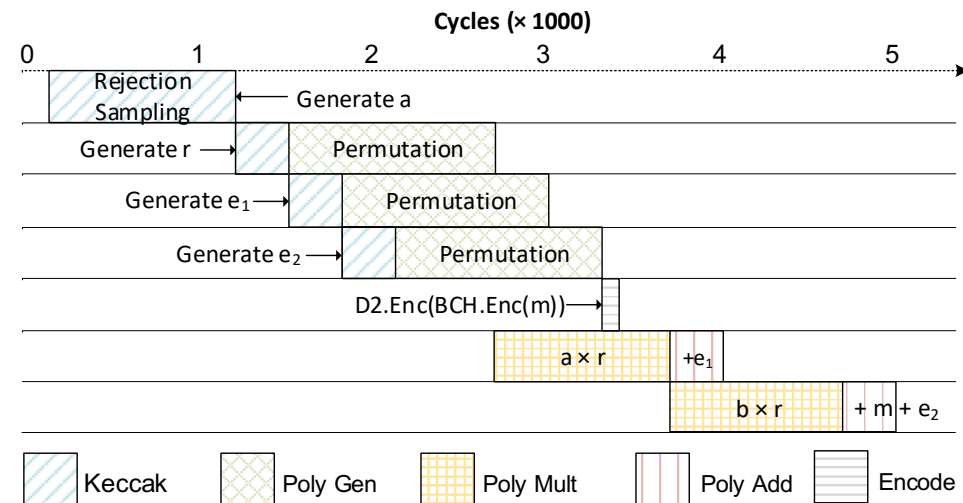


Common Optimization Method

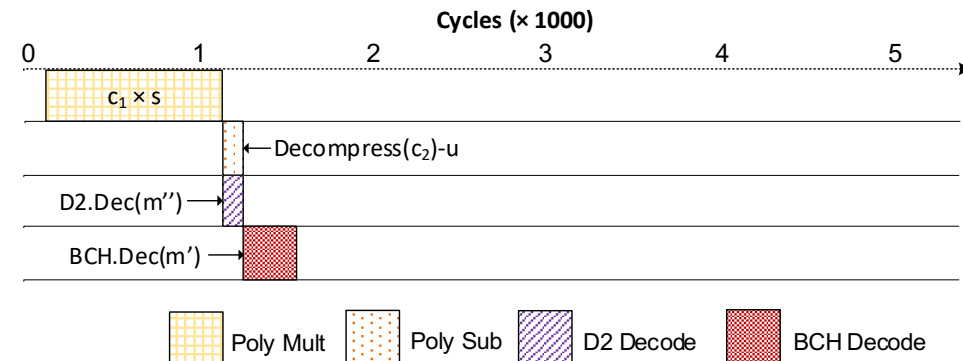


Efficient hardware scheduling to perform operations without data dependency in parallel

LAC Encryption



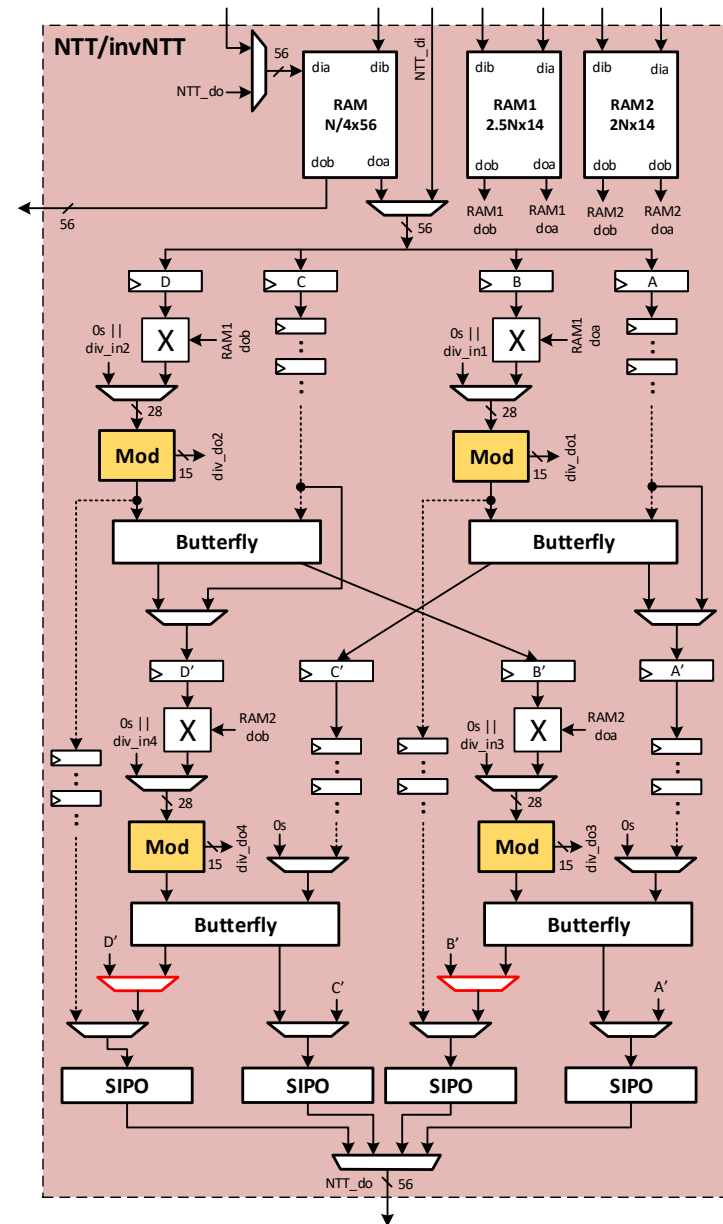
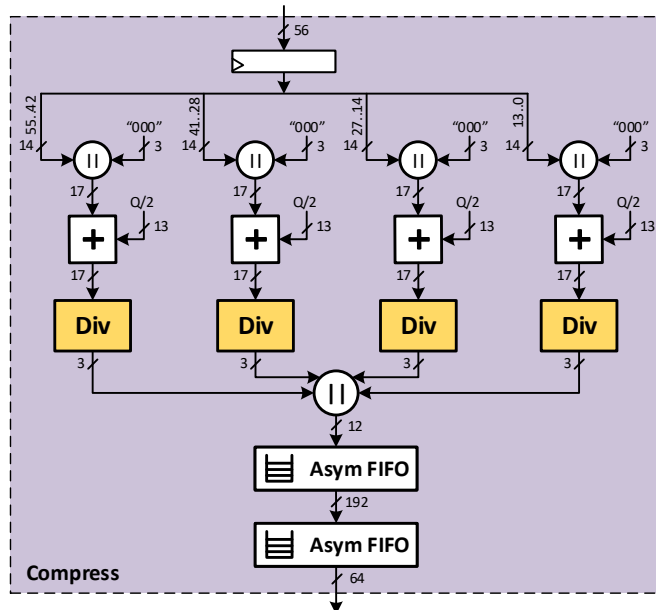
LAC Decryption



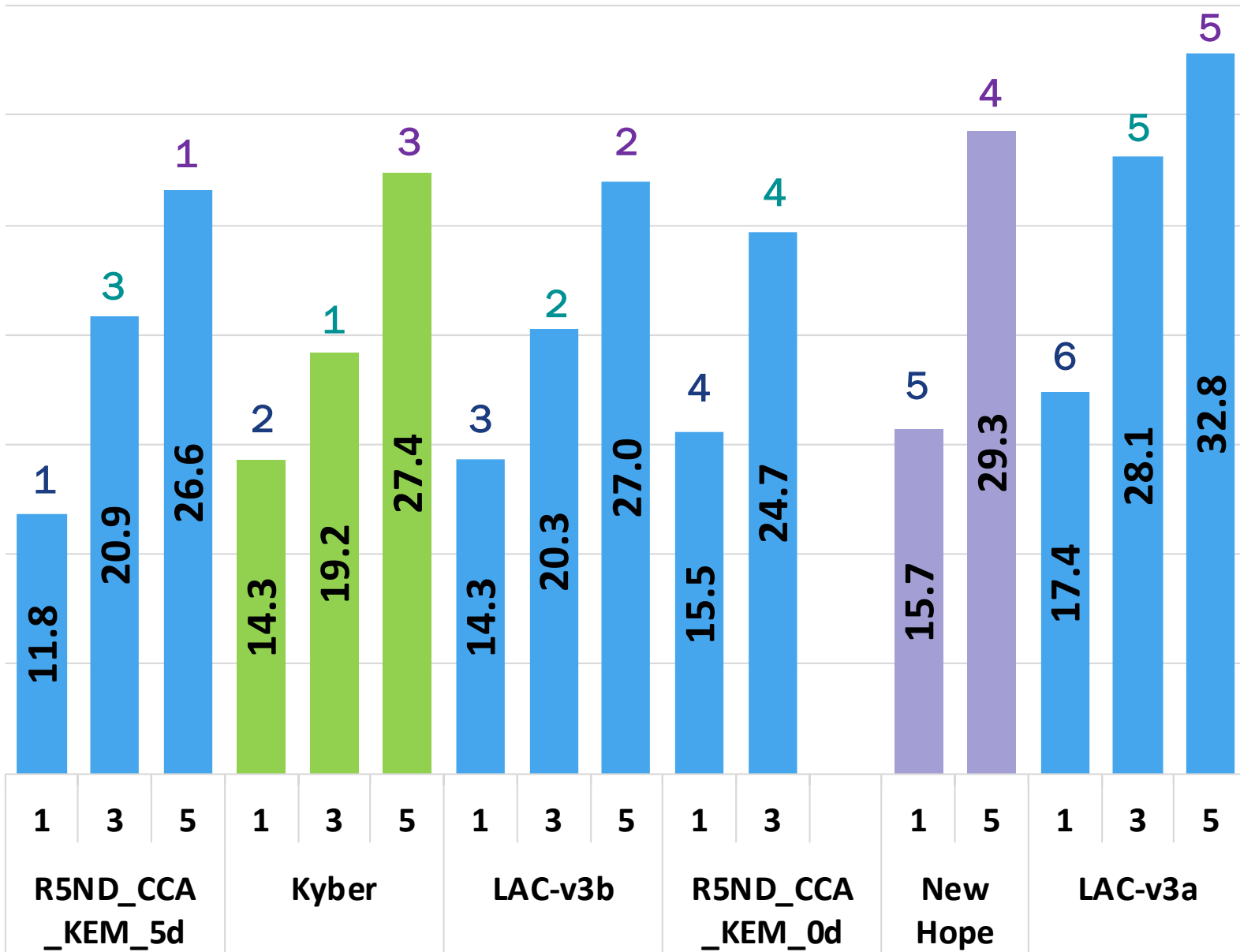
Algorithm-Specific Optimization Methods

NewHope & CRYSTALS-KYBER

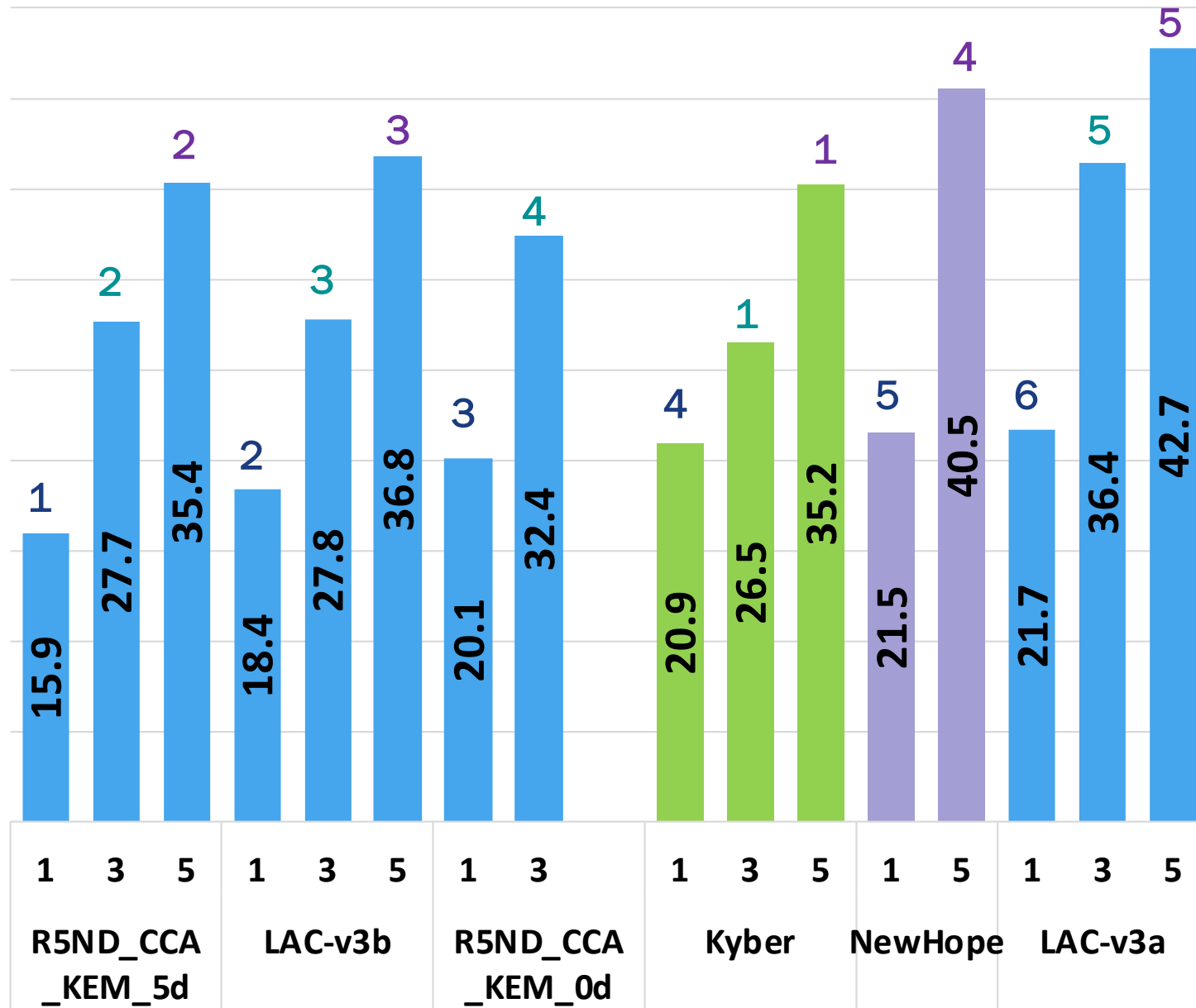
- Number Theoretic Transform (NTT)
- Processing **FOUR** coefficients at a time
- Resource sharing
e.g., use a single module to perform NTT, NTT^{-1} , & pointwise multiplication
- Efficient modular reduction



Encapsulation Time on Artix-7 [μ s]



Decapsulation Time on Artix-7 [μ s]



Rankings & Ratios on Artix-7

Encapsulation

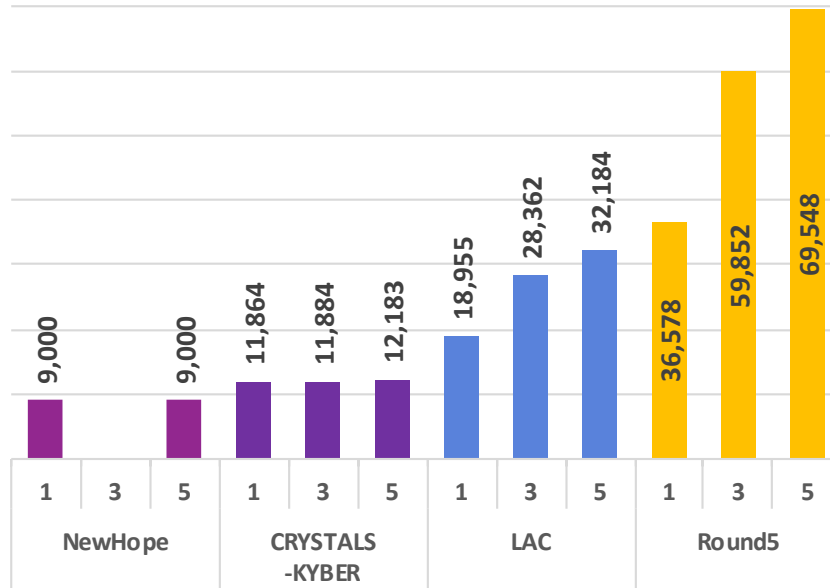
Level 1			Level 3			Level 5		
	Exe[us]	Ratio		Exe[us]	Ratio		Exe[us]	Ratio
Round5_5d	12.2	1.00	Kyber	19.9	1.00	Round5_5d	27.6	1.00
Kyber	14.8	1.21	LAC-v3b	21.2	1.07	LAC-v3b	28.1	1.02
LAC-v3b	14.8	1.21	Round5_5d	21.6	1.09	Kyber	28.4	1.03
Round5_0d	16.0	1.31	Round5_0d	25.6	1.29	NewHope	30.3	1.10
NewHope	16.3	1.34	LAC-v3a	29.1	1.46	LAC-v3a	33.9	1.23
LAC-v3a	17.9	1.47						

Decapsulation

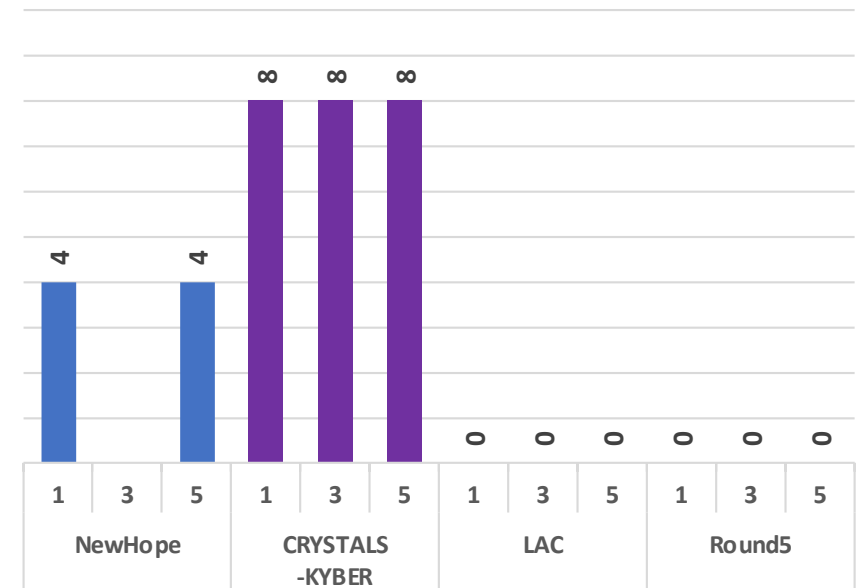
Level 1			Level 3			Level 5		
	Exe[us]	Ratios		Exe[us]	Ratio		Exe[us]	Ratio
Round5_5d	16.3	1	Kyber	27.2	1.00	Kyber	36.2	1.00
LAC-v3b	18.9	1.16	Round5_5d	28.4	1.04	Round5_5d	36.4	1.01
Round5_0d	20.6	1.26	LAC-v3b	28.7	1.06	LAC-v3b	37.9	1.05
Kyber	21.4	1.31	Round5_0d	33.2	1.22	NewHope	41.5	1.15
NewHope	22.0	1.35	LAC-v3a	37.4	1.38	LAC-v3a	43.8	1.21
LAC-v3a	22.2	1.36						

Resource Utilization on Artix-7

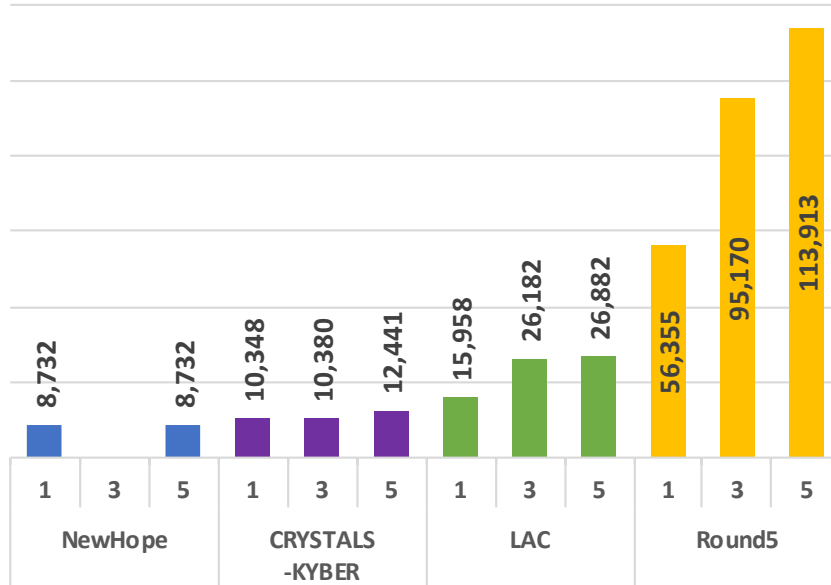
LUT



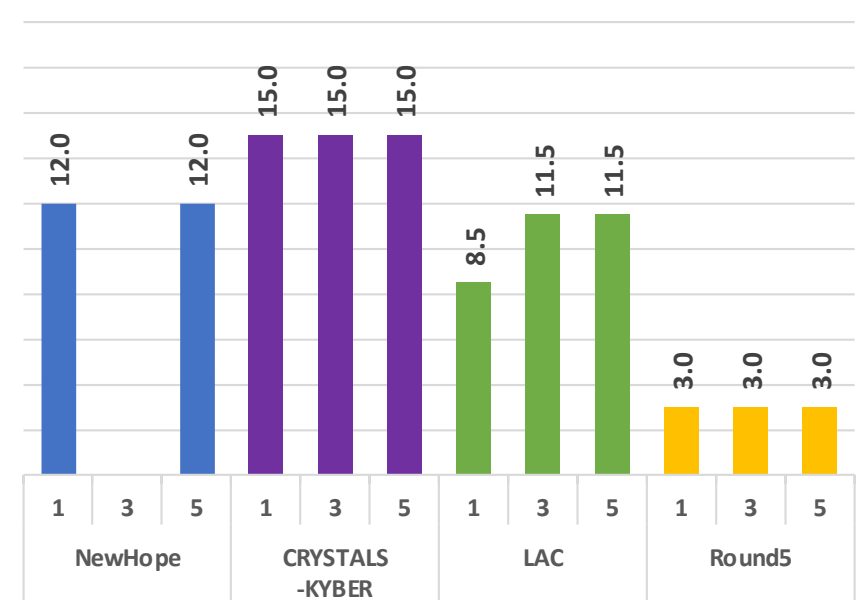
DSP



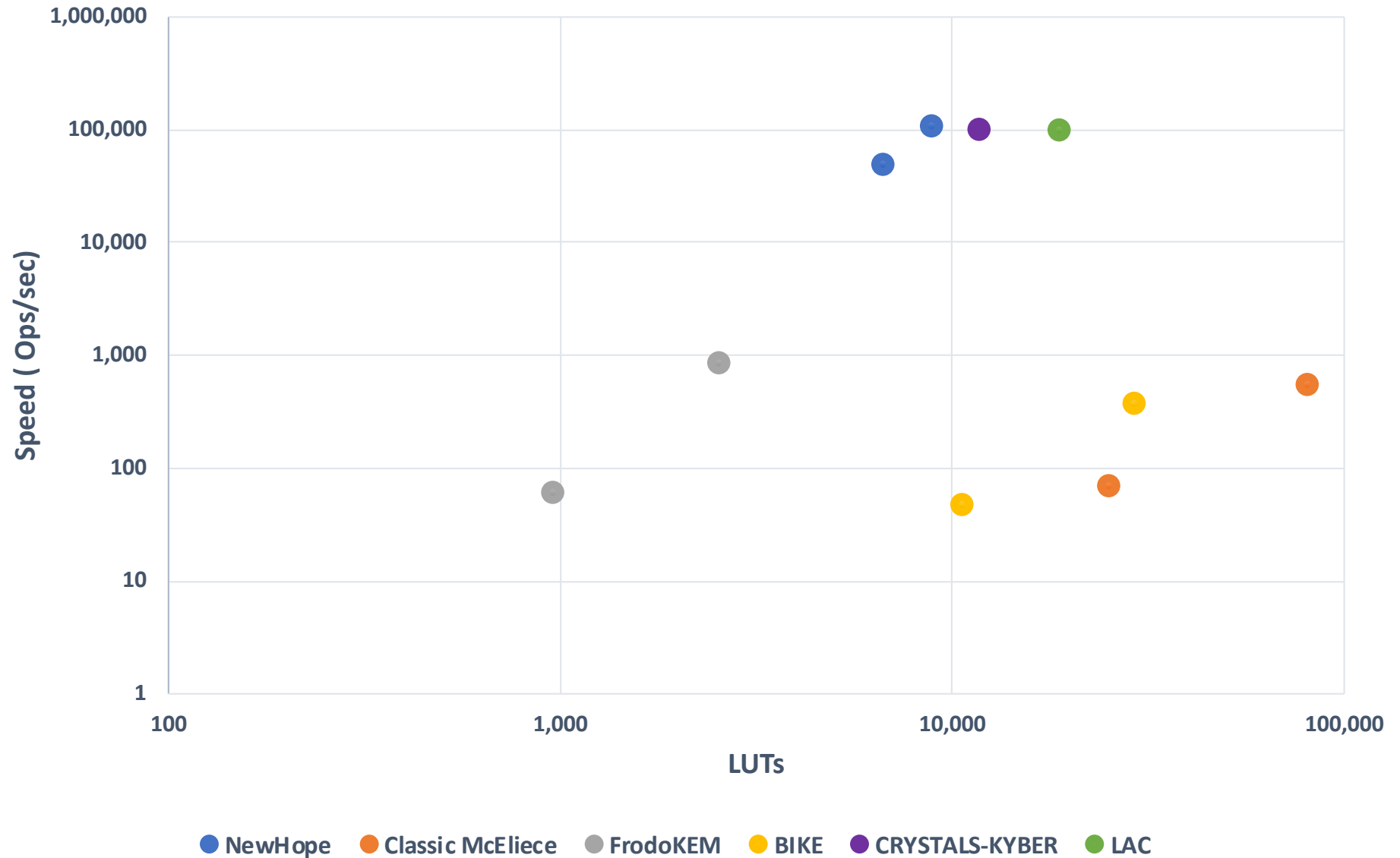
FF



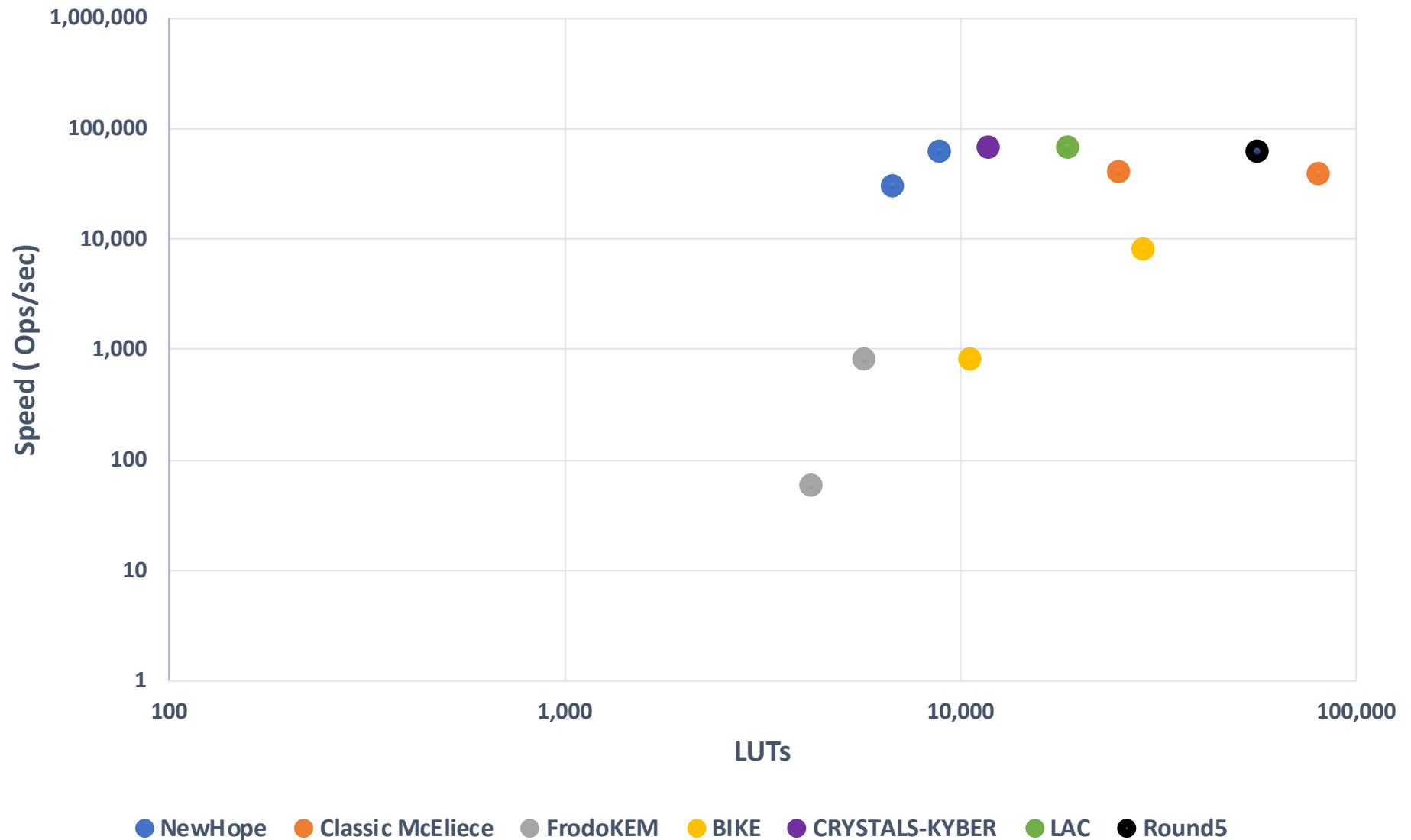
BRAM



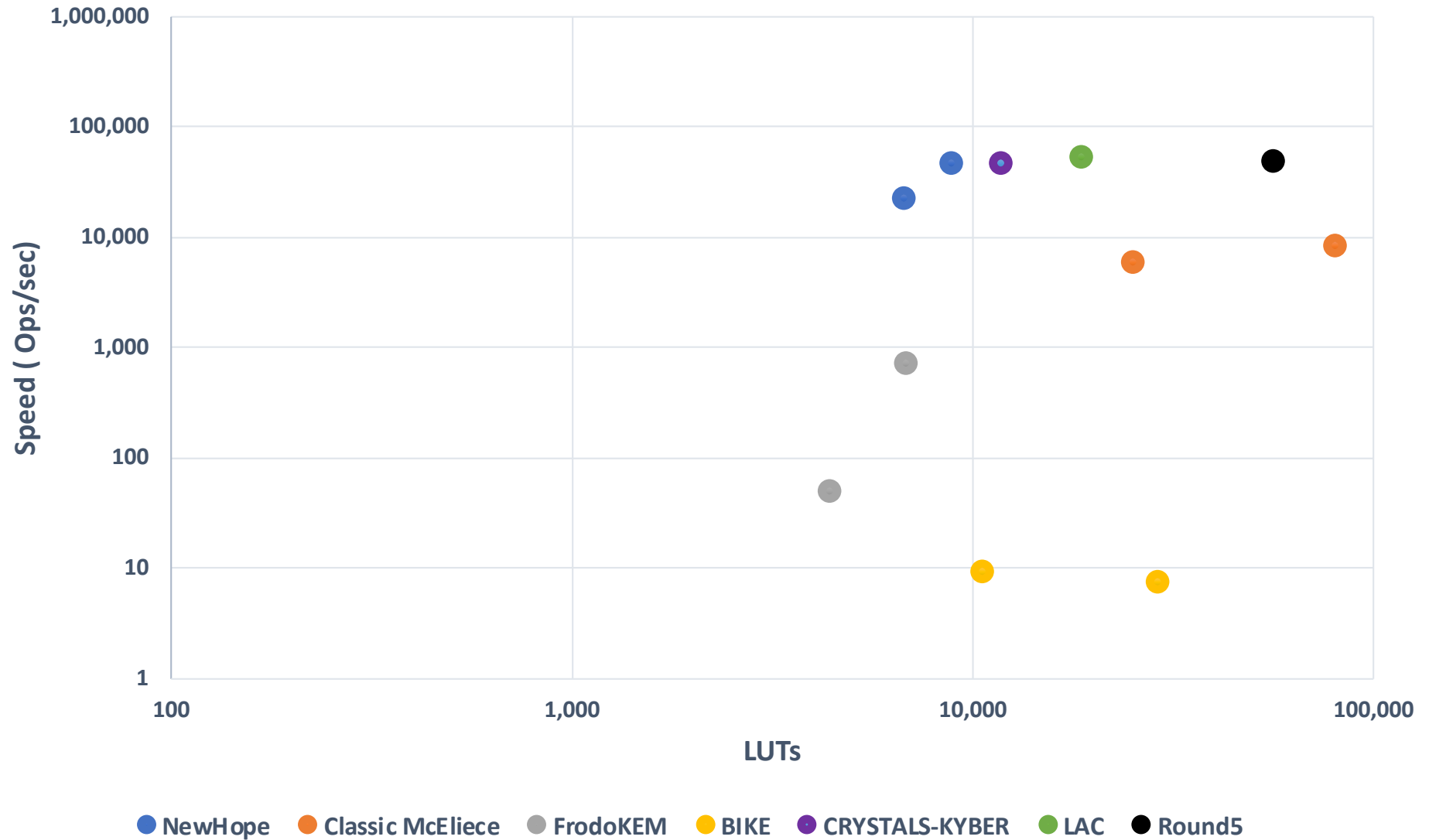
Level 1: Key Generation on Artix-7



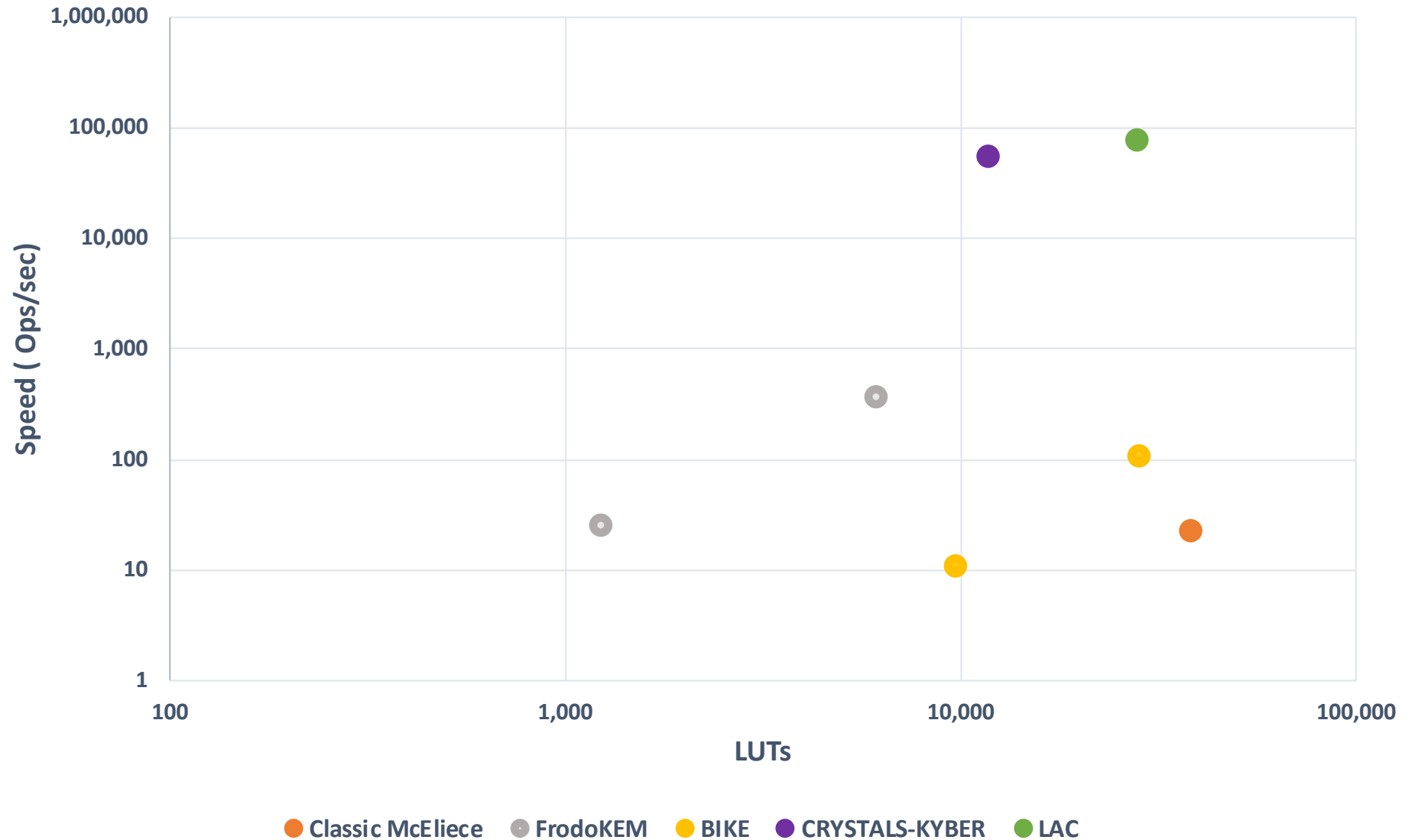
Level 1: Encapsulation on Artix-7



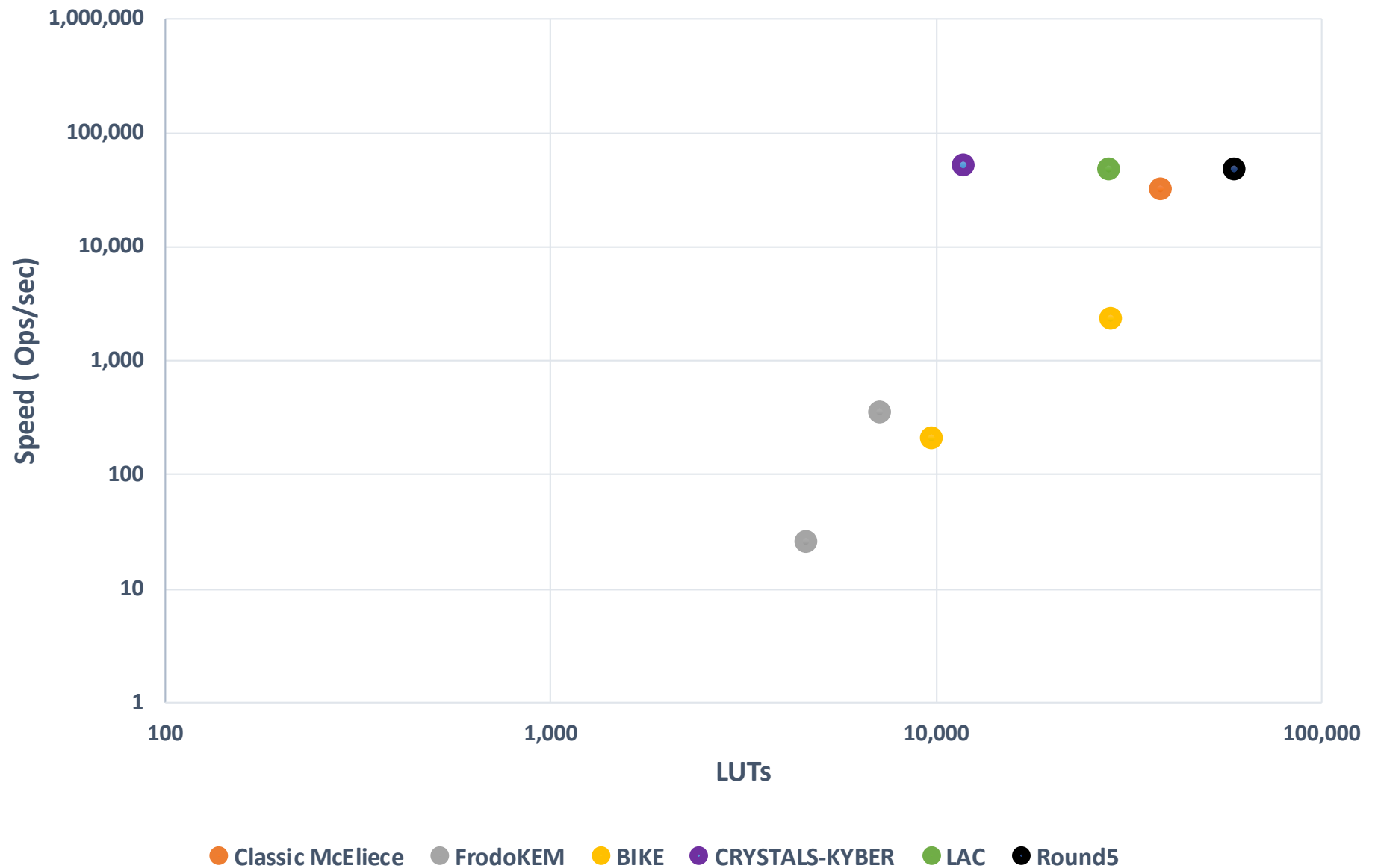
Level 1: Decapsulation on Artix-7



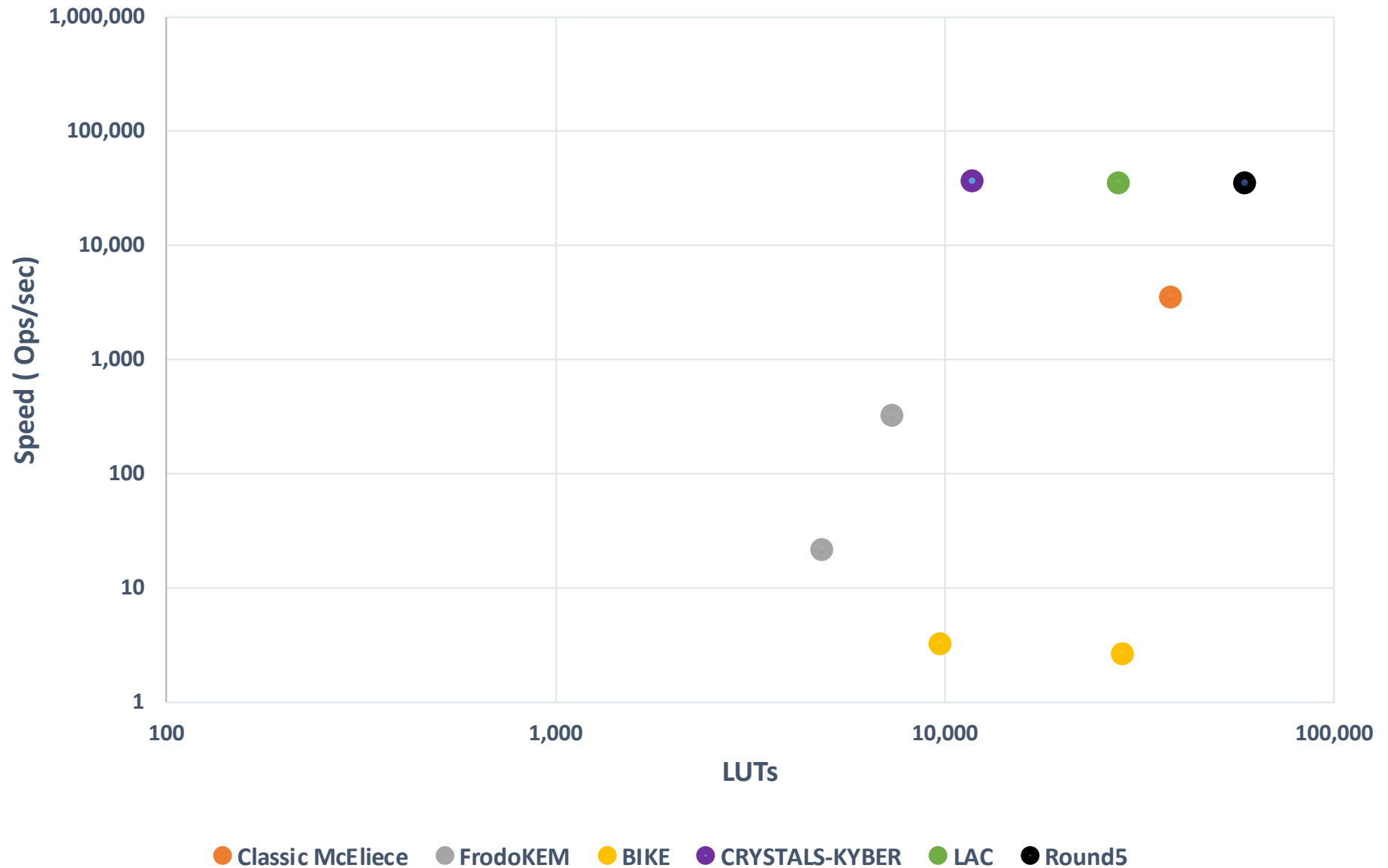
Level 3: Key Generation on Artix-7



Level 3: Encapsulation on Artix-7



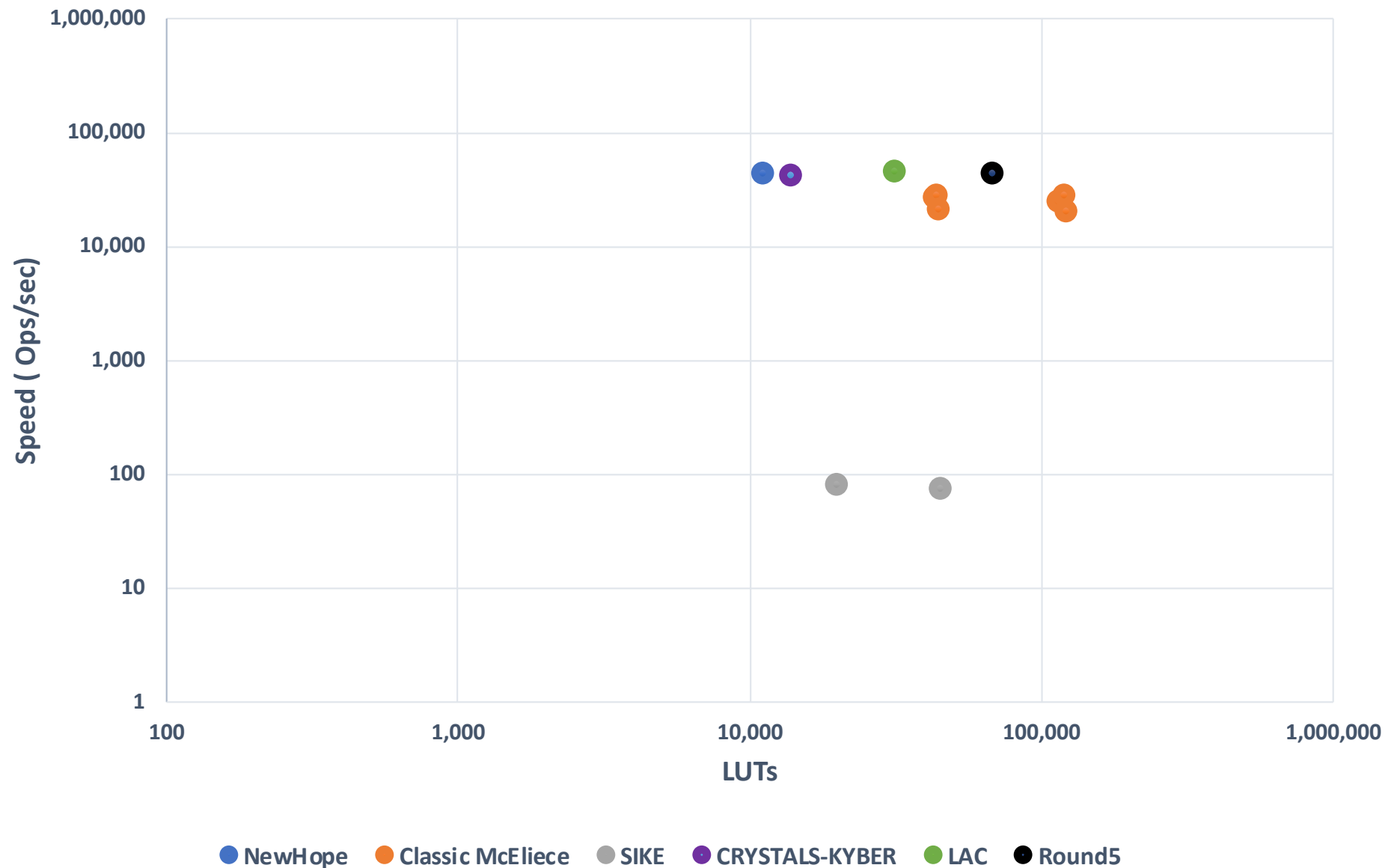
Level 3: Decapsulation on Artix-7



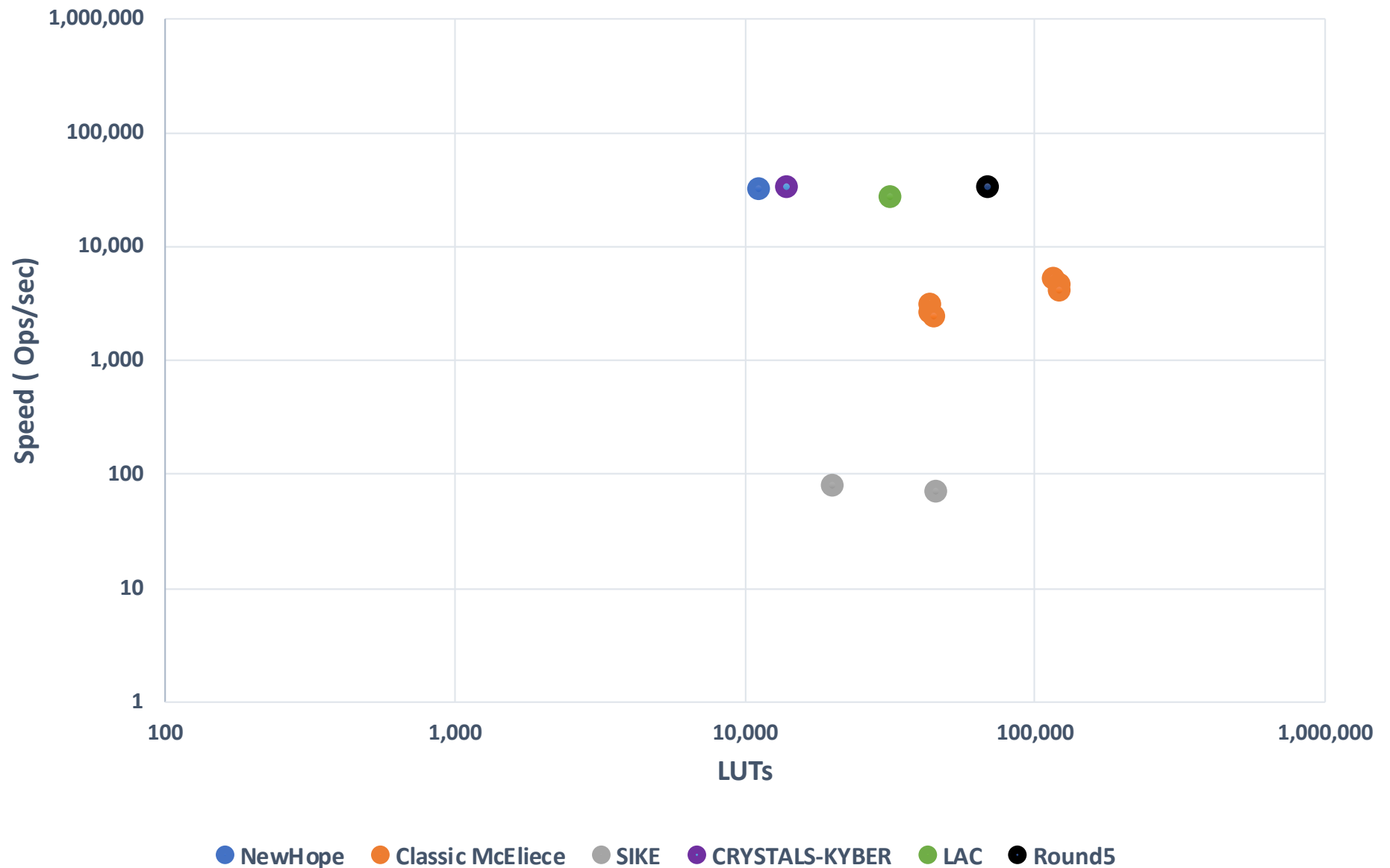
Level 5: Key Generation on Virtex-7



Level 5: Encapsulation on Virtex-7



Level 5: Decapsulation on Virtex-7





Hardware Design Conclusions

Conclusions for Hardware Implementations

- CRYSTALS-KYBER, LAC, NewHope, and Round5 comparable in terms of speed
- CRYSTALS-KYBER & NewHope superior in terms of resource utilization
- FrodoKEM and SIKE about 2 orders of magnitude slower for all operations
- BIKE and Classic McEliece about 2 orders of magnitude slower for key generation and decapsulation



Software/Hardware Co-design

Software/Hardware Codesign

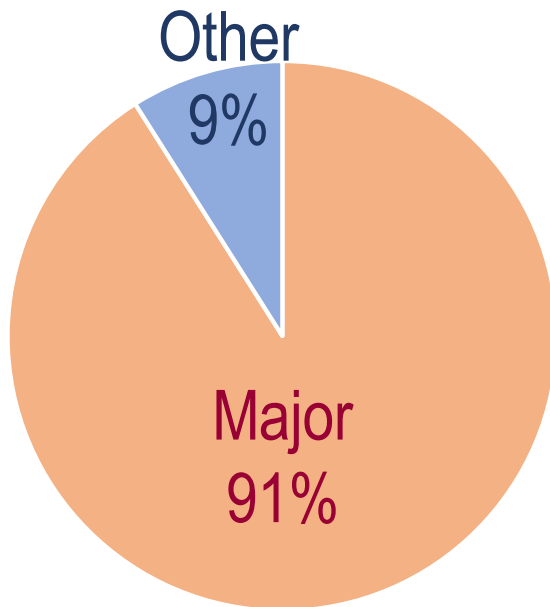
Software

Hardware

**Most time-critical
operation**

SW/HW Co-design: Motivational Example 1

Software

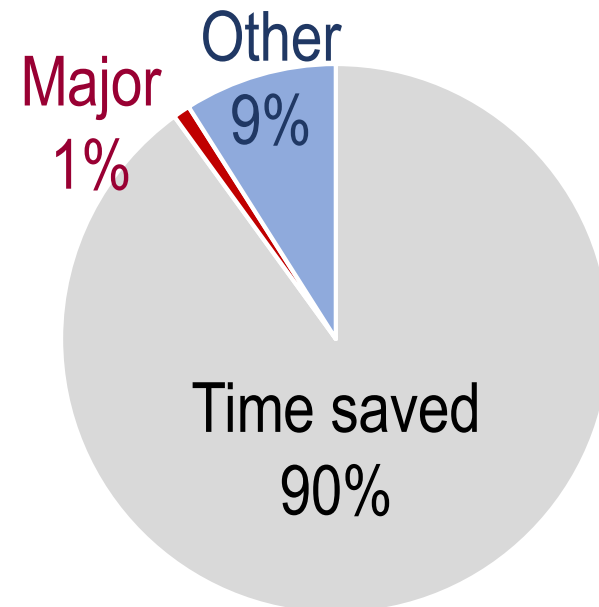


91% major operation(s)
9% other operations

speed-up ≥ 100



Software/Hardware



~1% major operation(s) in HW
9% other operations in SW

Total Speed-Up ≥ 10

SW/HW Co-design: Advantages

- Focus on a few (typically 1-3) major operations, known to be easily parallelizable
 - ☆ much shorter development time (at least by a factor of 10)
 - ☆ guaranteed substantial speed-up
 - ☆ high-flexibility to changes in other operations (such as candidate tweaks)
- Insight regarding performance of future instruction set extensions of modern microprocessors
- Possibility of implementing multiple candidates by the same research group, eliminating the influence of different
 - ☆ design skills
 - ☆ operation subset (e.g., including or excluding key generation)
 - ☆ interface & protocol
 - ☆ optimization target
 - ☆ platform

SW/HW Co-design: Potential Pitfalls

- Performance & ranking may strongly depend on features of a particular platform
 - ☆ Software/hardware interface
 - ☆ Support for cache coherency
 - ☆ Differences in max. clock frequency
- Performance & ranking may strongly depend on the selected hardware/software partitioning



First step, not the ultimate solution!

SW/HW Co-design Classification

Loosely Coupled
HW Accelerators



Hard Processor
Cores

- Cortex-A53
- Cortex-A9



Soft Processor
Cores

- RISC-V

Tightly Coupled
HW Accelerators



Soft processor
cores

- RISC-V

Advantages of Loosely and Tightly Coupled Accelerators

Loosely Coupled	Tightly Coupled
Standard interfaces (AXI, TileLink)	Low data transfer rate overhead
Ease of development	Lower amount of hardware resources
Portability	Crypto agility
More flexible at the hardware development stage	More flexible in the post-silicon phase

Our Focus in Round 2

Loosely Coupled
HW Accelerators



Hard Processor
Cores

- Cortex-A53
- Cortex-A9



Soft Processor
Cores

- RISC-V

Tightly Coupled
HW Accelerators

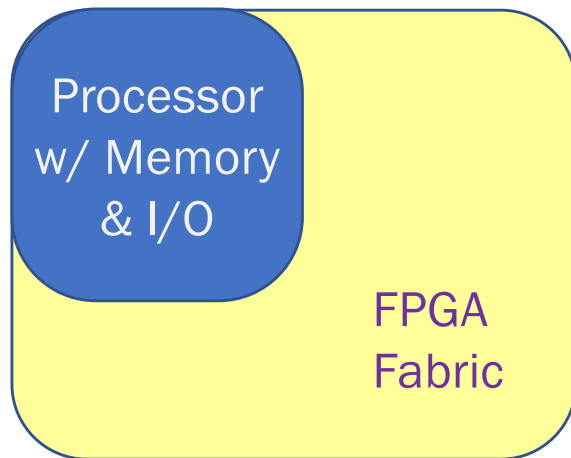


Soft processor
cores

- RISC-V

Two Major Types of Platforms

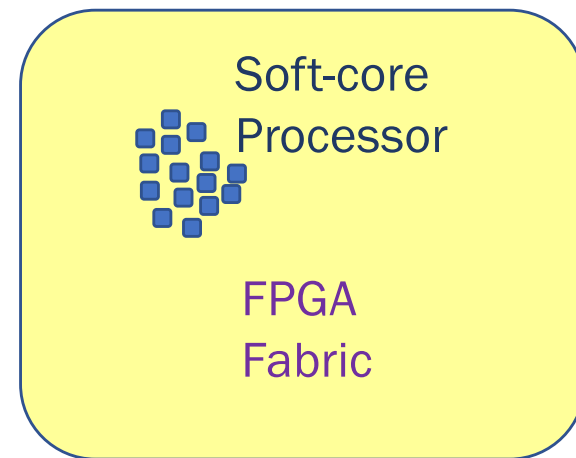
System on Chip (SoC) FPGA



Examples:

- Xilinx Zynq 7000 System on Chip (SoC)
Zynq UltraScale+ MPSoC
- Intel Cyclone V SoC
Stratix 10 SoC FPGAs,

“Traditional” FPGA



Examples:

Xilinx Artix-7, Virtex-7,
Virtex UltraScale+
Intel Cyclone 10 LP,
Stratix 10

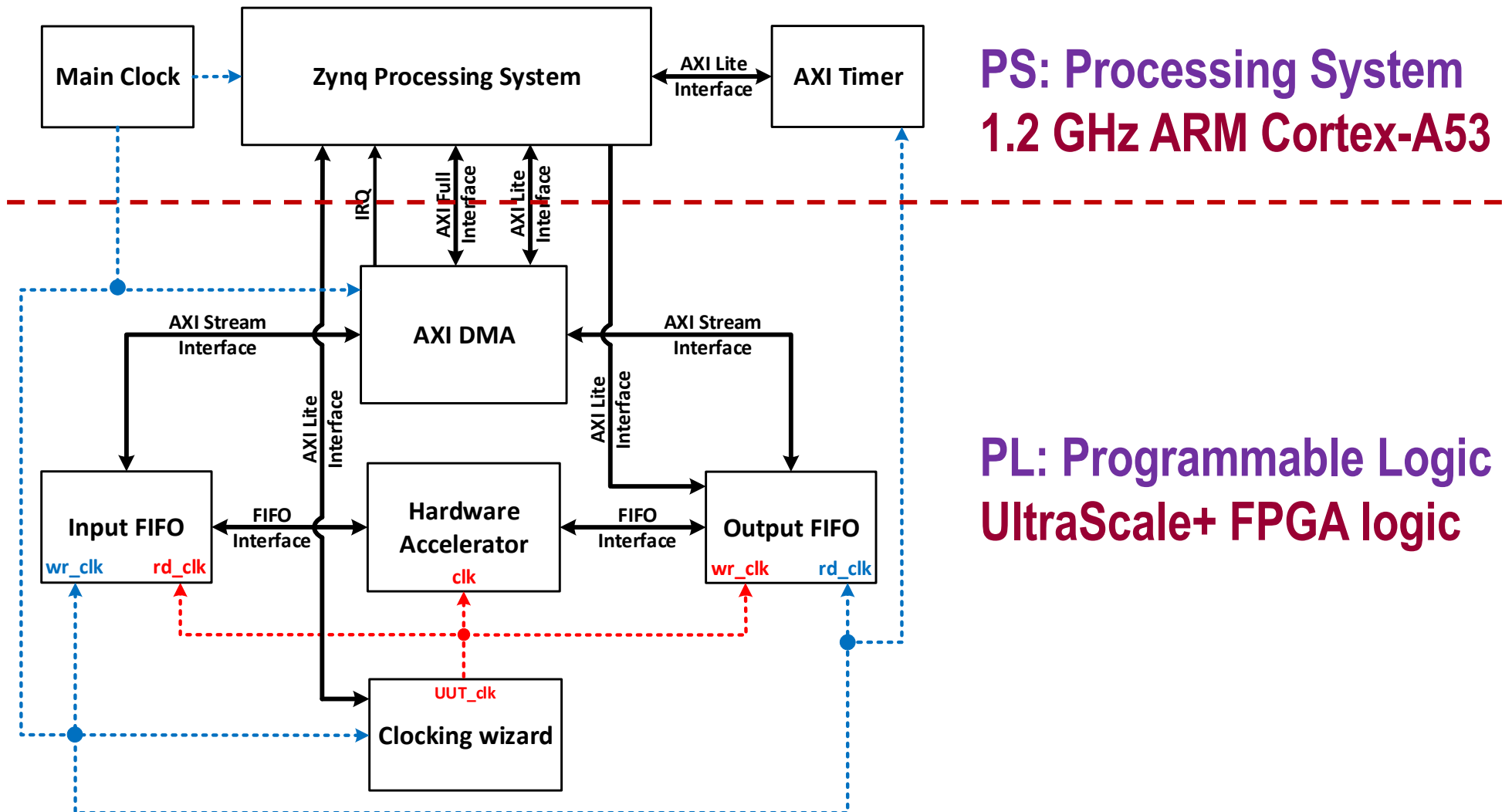
Two Major Types of Platform

Feature	Hard Processor Cores	Soft Processor Cores
Processor	ARM	RISC-V
Clock frequency	>1 GHz	max. 200-450 MHz
Portability	similar FPGA SoCs	various FPGAs, SoC FPGAs, and ASICs
Loosely-coupled accelerators	Yes	Yes
Tightly-coupled accelerators	No	Yes
Ease of design (methodology, tools, OS support)	Easy	Dependent on a particular soft-core processor and tool chain



Platform & Experimental Setup

Xilinx Zynq UltraScale+ MPSoC





Our Case Studies

SW/HW Codesign: Case Study

12 Key Encapsulation Mechanisms (KEMs)

representing

8 out of 9 Round 2 Lattice-Based KEMs

LWE (Learning with Error)-based:

FrodoKEM

RLWR (Ring Learning with Errors)-based:

NewHope, LAC (3a/3b)

RLWR (Ring Learning with Rounding)-based:

Round5 (0d/5d)

Module-LWE-based:

CRYSTALS-KYBER

Module-LWR-based:

Saber

NTRU-based:

NTRU

- NTRU-HPS
- NTRU-HRSS

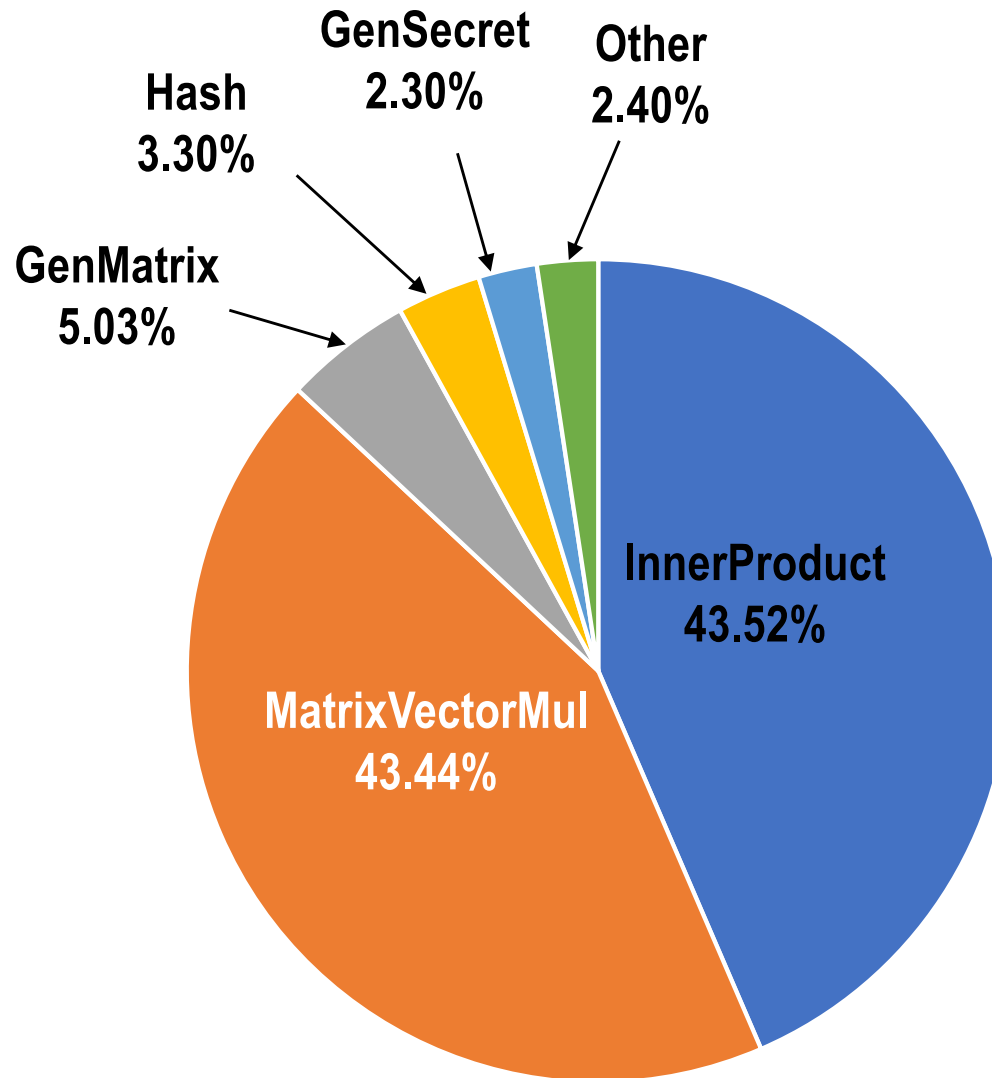
NTRU Prime

- Streamlined NTRU Prime
- NTRU LPRime



Methodology

SW/HW Codesign: Step 1 Profiling



LightSaber Decapsulation

SW/HW Co-design: Step 2 SW/HW Partitioning

Top candidates for offloading to hardware

From profiling:

- Large percentage of the execution time
- Small number of function calls

From manual analysis of the code:

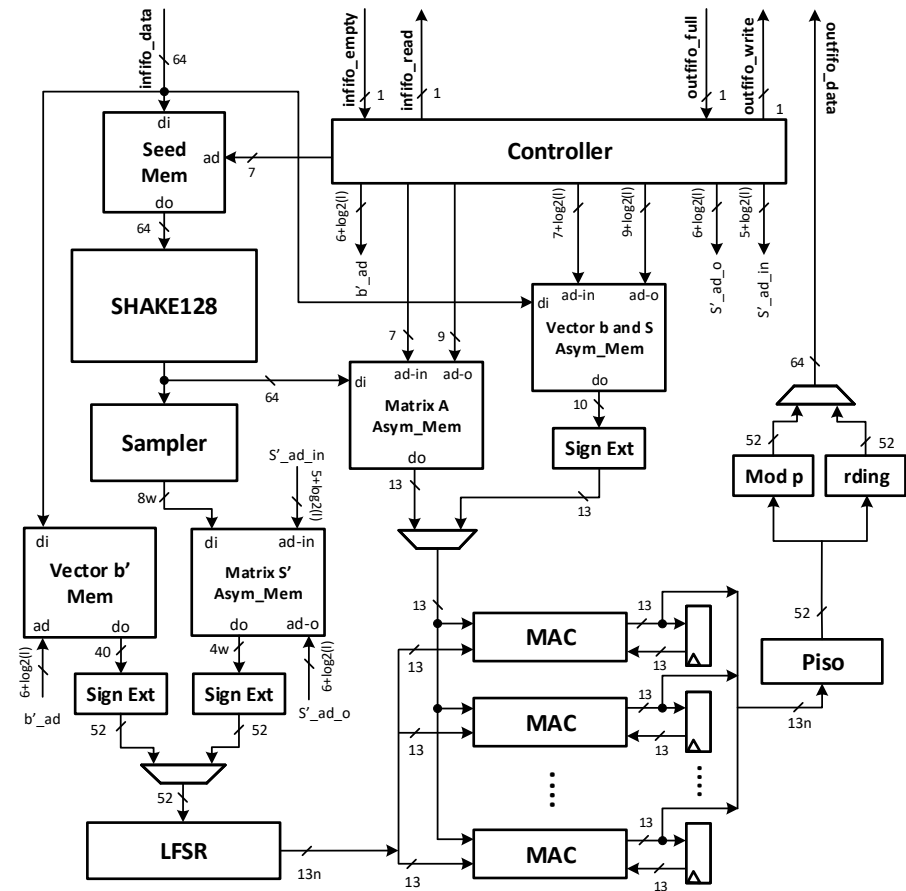
- Small size of inputs and outputs
- Potential for combining with neighboring functions

From knowledge of operations and concurrent computing:

- High potential for parallelization

Operations Offloaded to Hardware

- Major arithmetic operations
 - Polynomial multiplications
 - Matrix-by-vector multiplications
 - Vector-by-vector multiplications
- All hash-based operations
 - (c)SHAKE128, (c)SHAKE256
 - SHA3-256, SHA3-512



Hardware accelerator
of Saber

Saber Decapsulation

Functions offloaded to hardware highlighted in yellow

Algorithm 8 Pseudocode of Saber.KEM.Decaps ($sk = (s, z, pkh), pk = (seed_A, b), c$) [46]

```
1:  $m' = \text{Saber.PKE.Dec}(s, c)$ 
2:  $(\hat{K}, r') = g(pkh, m')$ 
3:  $c' = \text{Saber.PKE.Enc}(pk, m'; r')$ 
4: if  $c = ct'$  then
5:   return  $K = H(\hat{K}', c)$ 
6: else
7:   return  $K = H(z, c)$ 
8: end if
```

Algorithm 9 Pseudocode of Saber.PKE.Enc ($pk = (seed_A, b), m \in R_2; r$) [46]

```
1:  $A = \text{gen}(seed_A) \in R_q^{l \times l}$ 
2:  $(\hat{K}, r') = g(pkh, m')$ 
3: if  $r$  is not specified then
4:    $r = u(\{0, 1\}^{256})$ 
5: end if
6:  $s' = \beta_\mu(R_q^{l \times l}; r)$ 
7:  $b' = ((As' + h) \bmod q) \gg (\epsilon_q - \epsilon_p) \in R_p^{l \times l}$ 
8:  $v' = b'^T (s' \bmod p) \in R_p$ 
9:  $c_m = (v' + h_1 - 2^{\epsilon_p - 1} m \bmod p) \gg (\epsilon_q - \epsilon_T) \in R_T$ 
10: return  $c := (c_m, b')$ 
```

Algorithm 10 Pseudocode of Saber.PKE.Dec ($sk = s, c = (c_m, b')$) [46]

```
1:  $v = b'^T (s \bmod p) \in R_p$ 
2:  $m' = ((v - 2^{\epsilon_p} - \epsilon_T c_m + h_2) \bmod p) \gg (\epsilon_p - 1) \in R_2$ 
3: return  $m'$ 
```

SW/HW Co-design: Step 3 Accelerator Design

Target: Minimum Execution Time

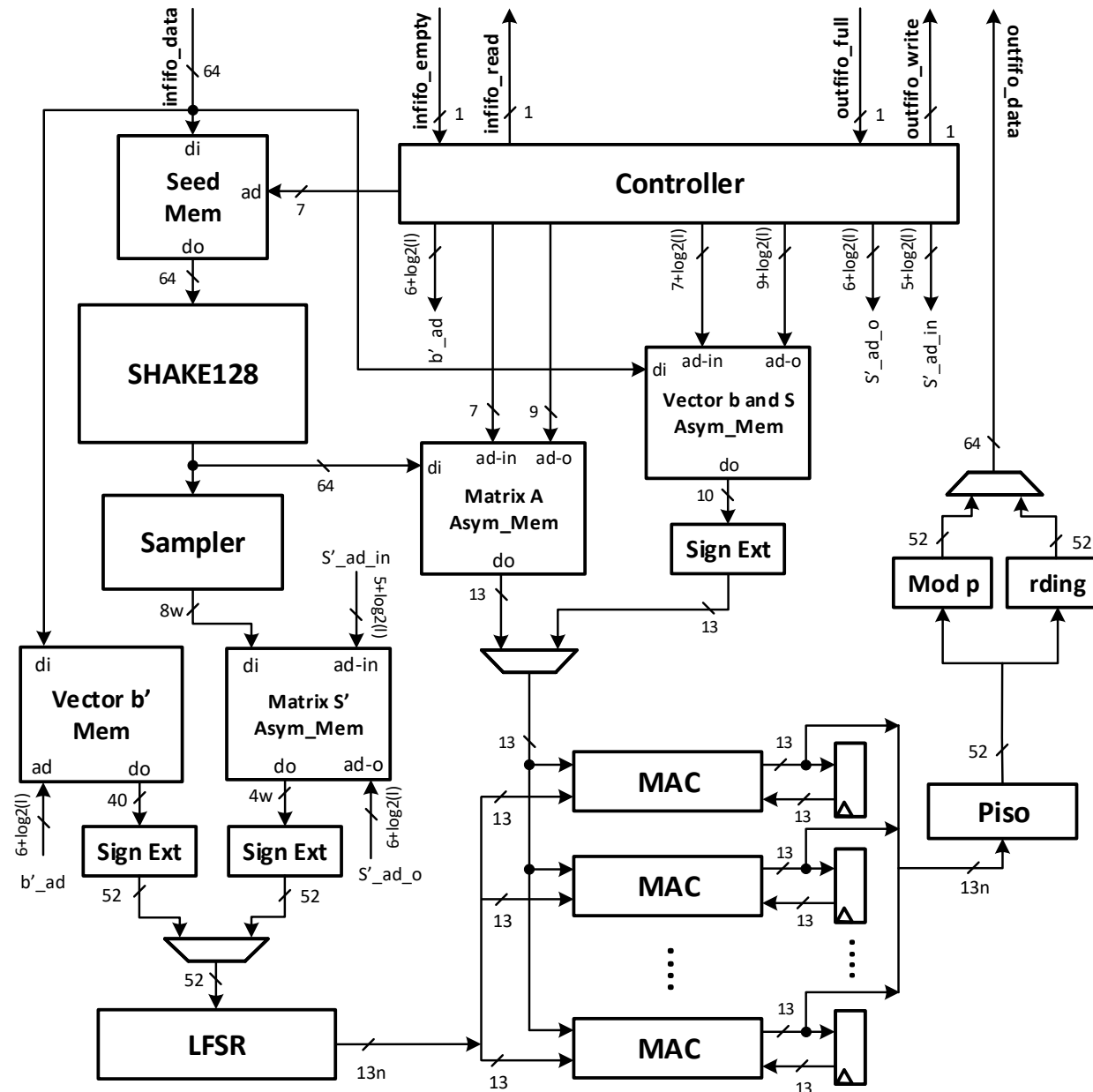
Hardware:

- Register-Transfer Level methodology with VHDL or Verilog
 - ★ Block diagram of the Datapath
 - ★ Algorithmic State Machine (ASM) chart of the Controller

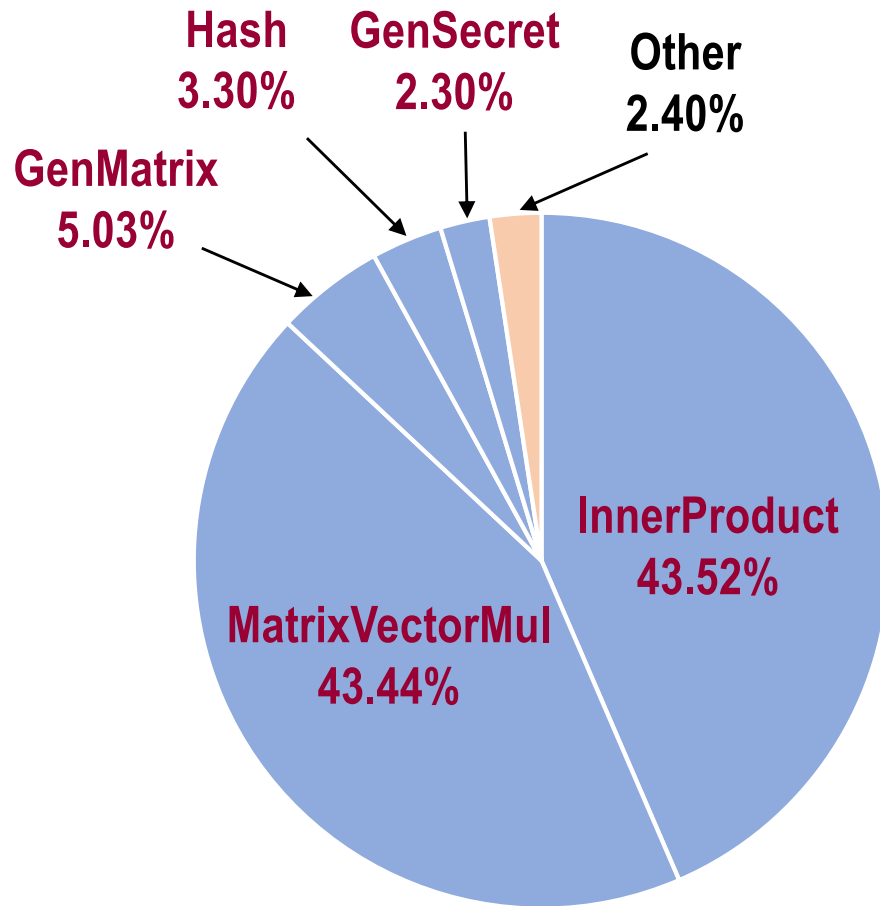
Software:

- Input/Output transfers
- Transfer of control between the processor and the accelerator

Detailed hierarchical block diagrams developed for the entire hardware accelerator

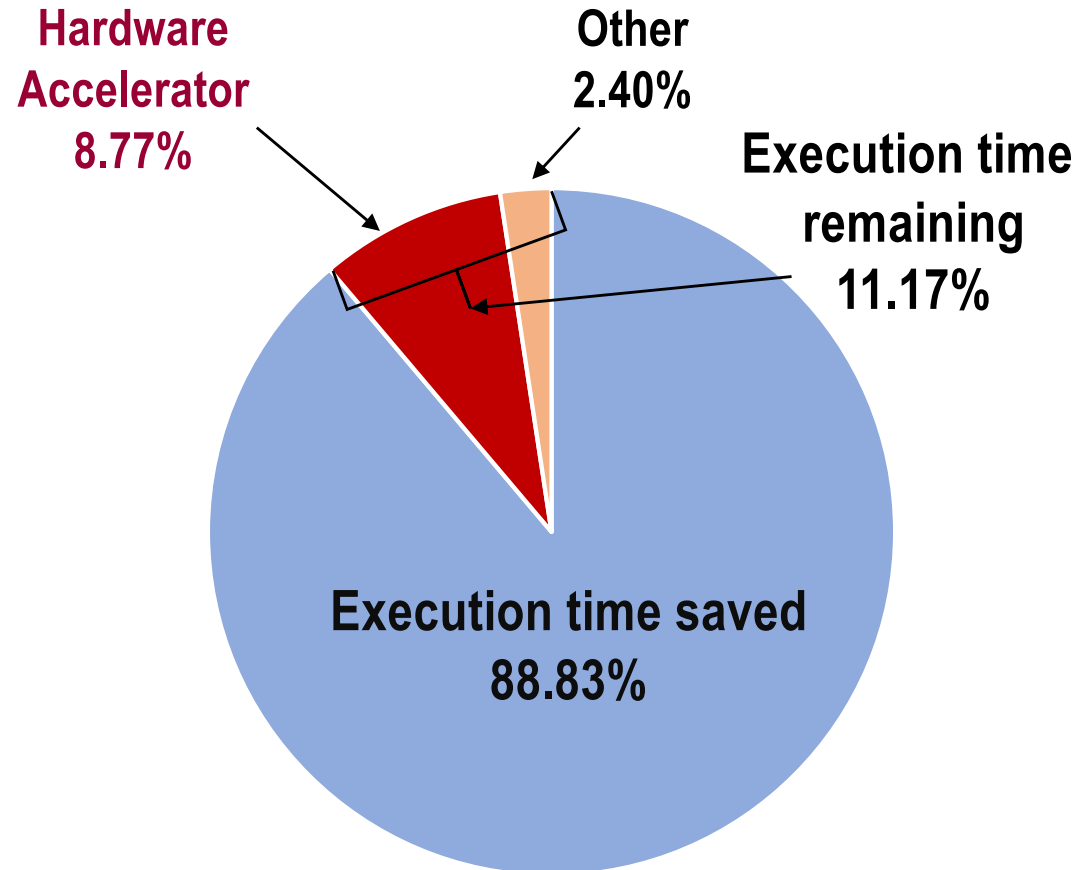


LightSaber Decapsulation



Execution time of functions
to be moved to hardware
97.60%

Execution time of functions
remaining in software
2.40%



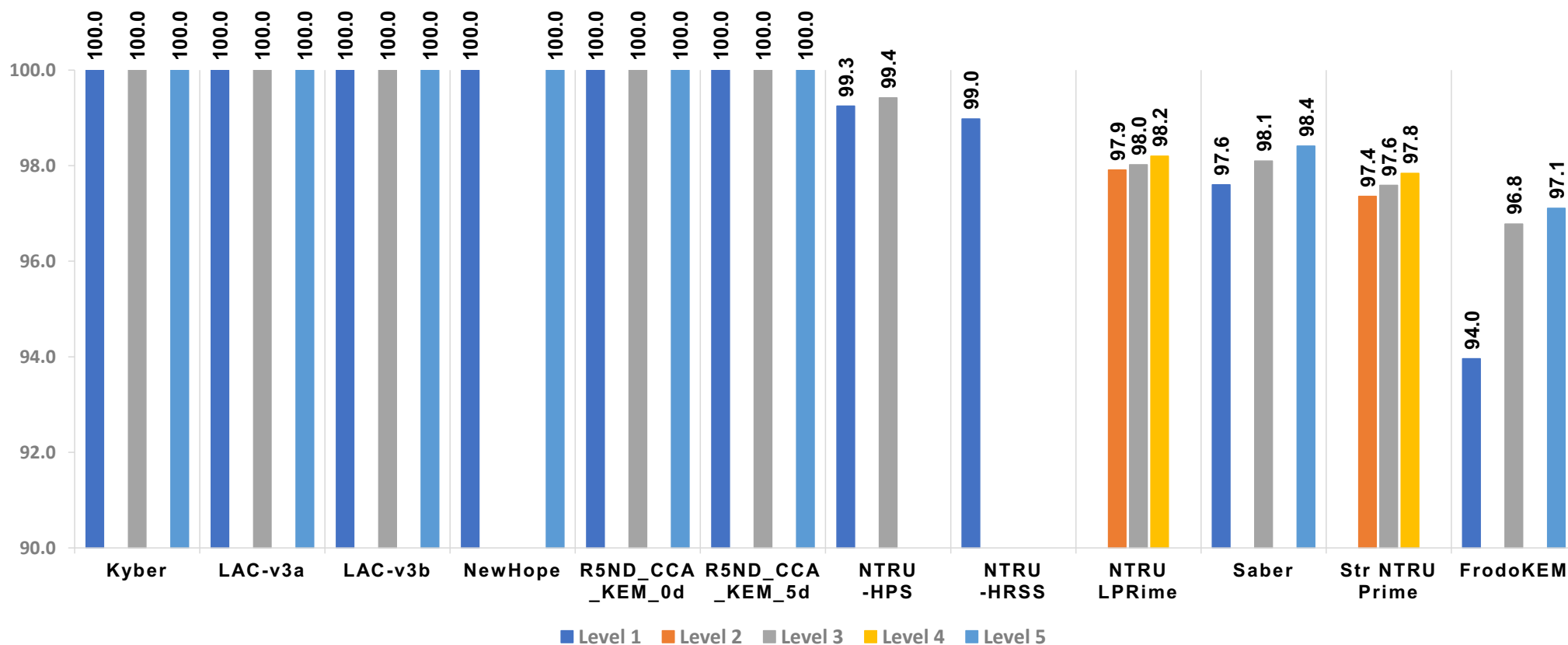
$$\text{Accelerator Speed-Up} = 97.60 / 8.77 = 11.1$$

$$\text{Total Speed-Up} = 100 / 11.17 = 9.0$$

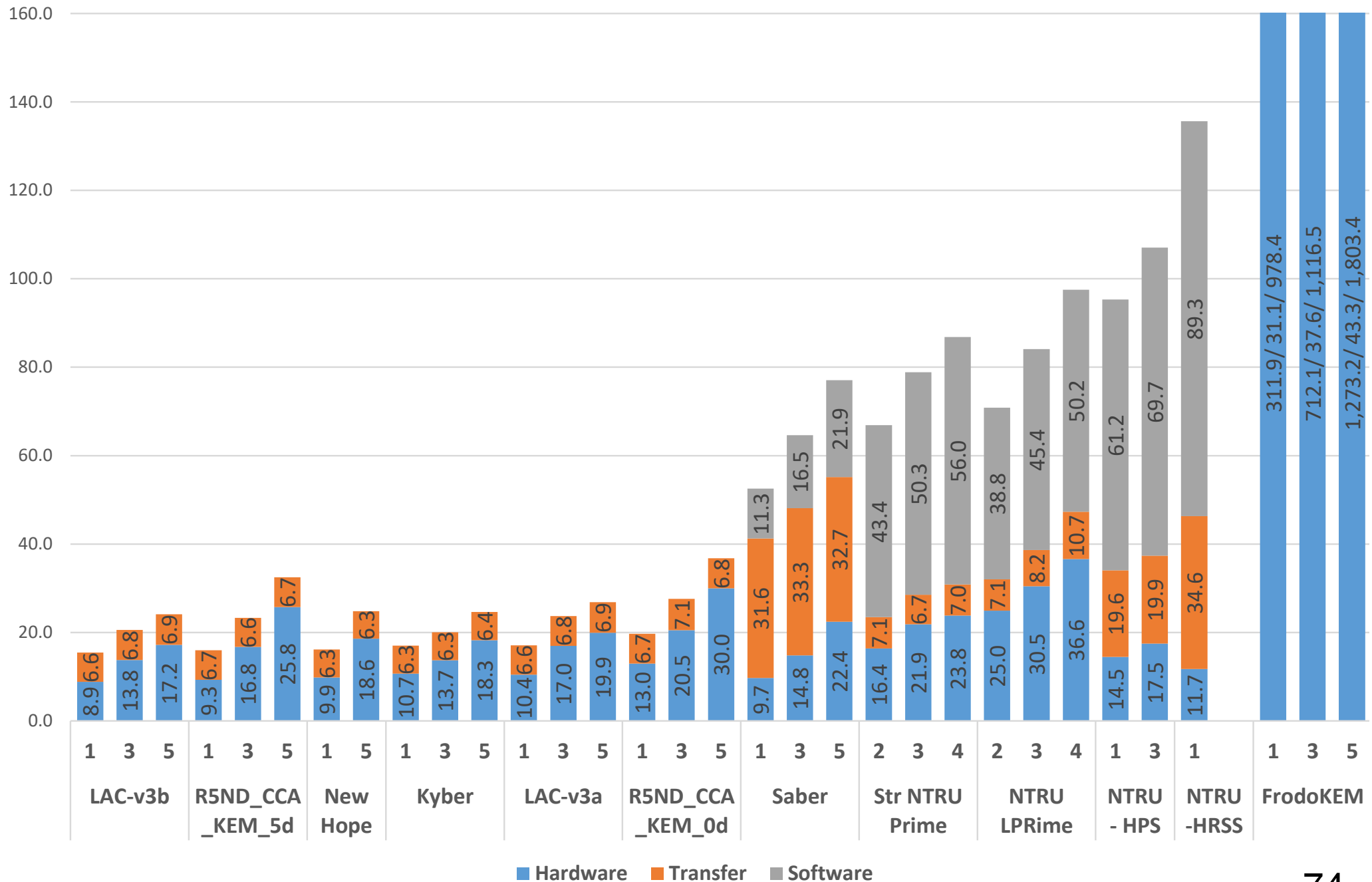


Results

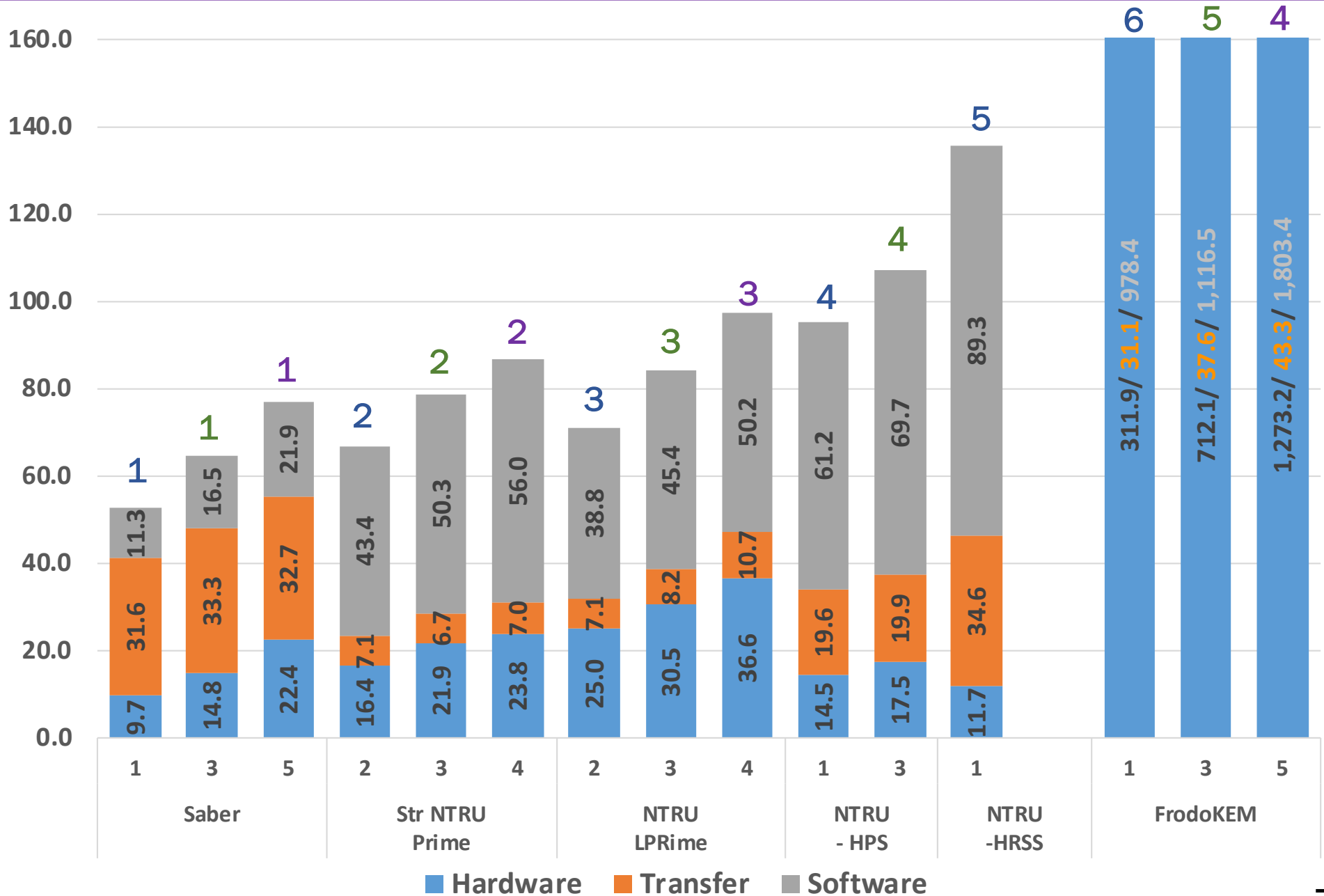
SW Part Sped up by HW[%]: Decapsulation



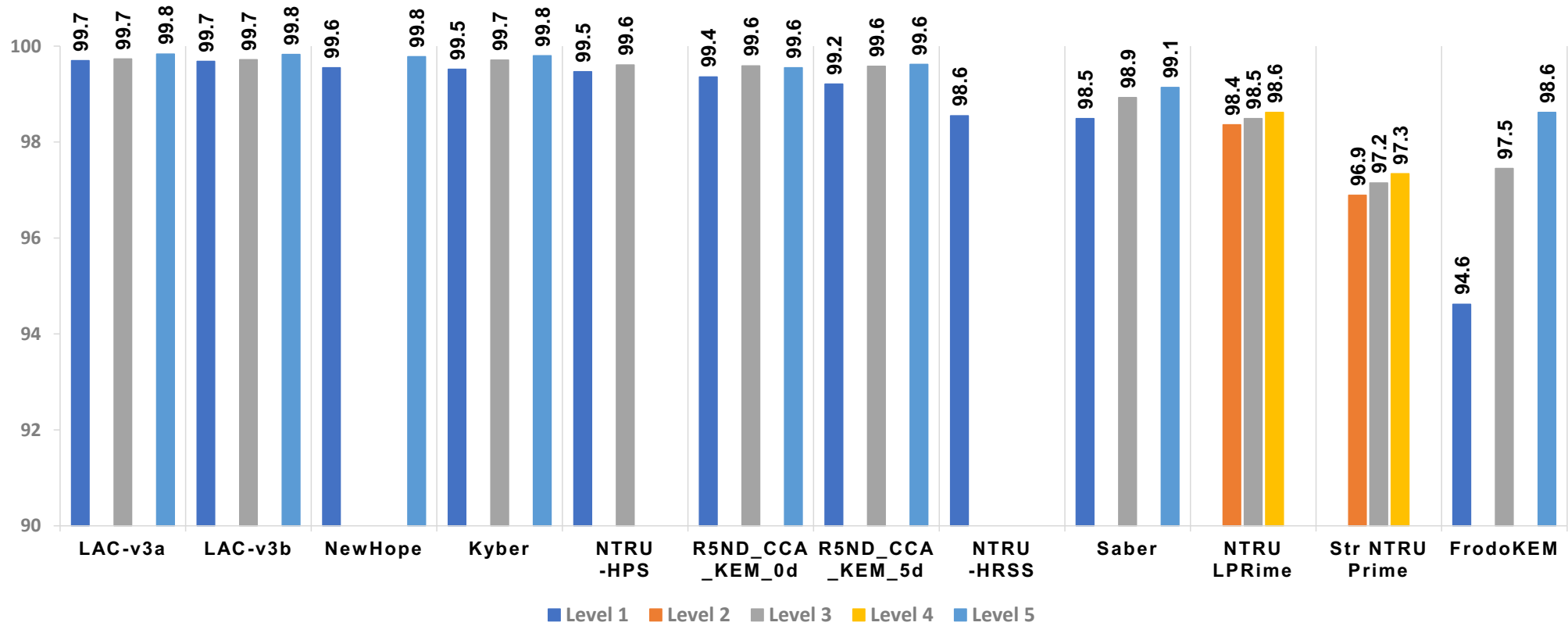
Round2 KEMs: SW/HW Results for Decaps



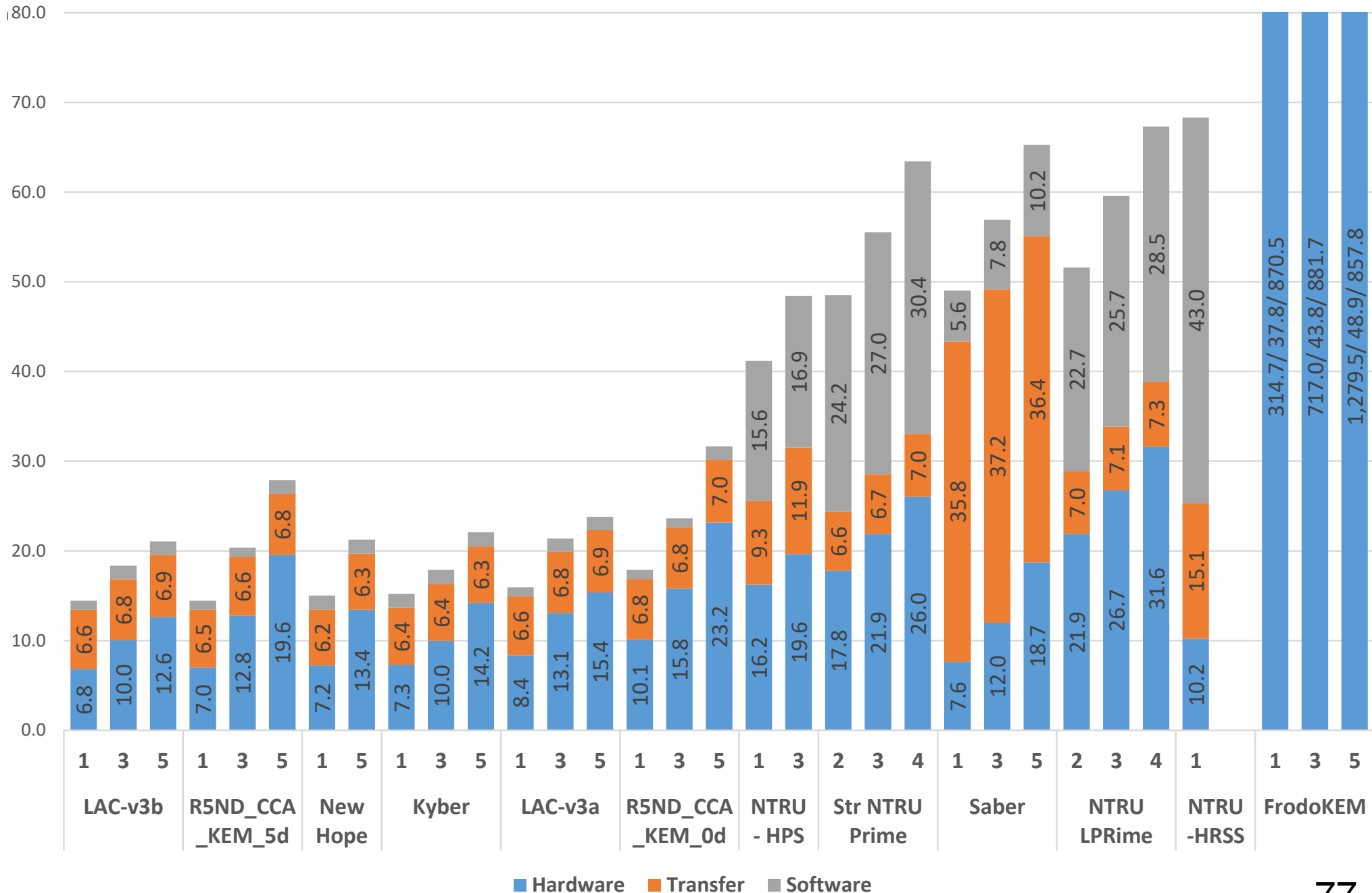
Round2 KEMs: SW/HW Results for Decaps



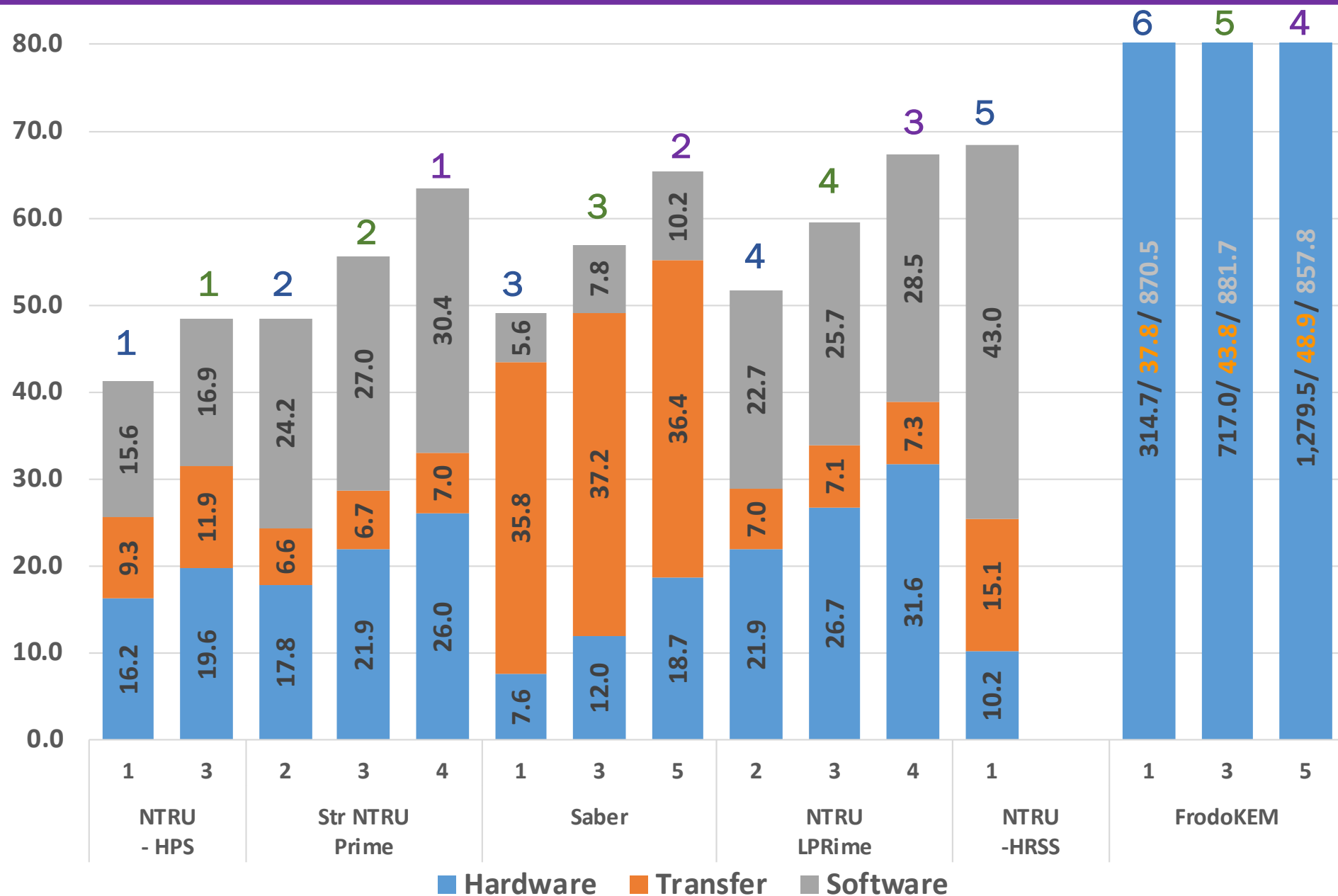
SW Part Sped up by HW[%]: Encapsulation



Round2 KEMs: SW/HW Results for Encaps

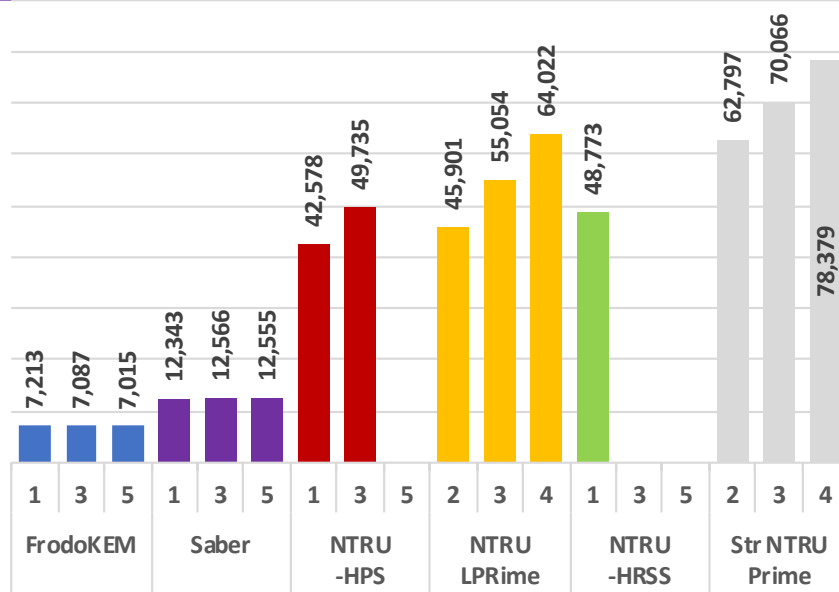


Round2 KEMs: SW/HW Results for Encaps

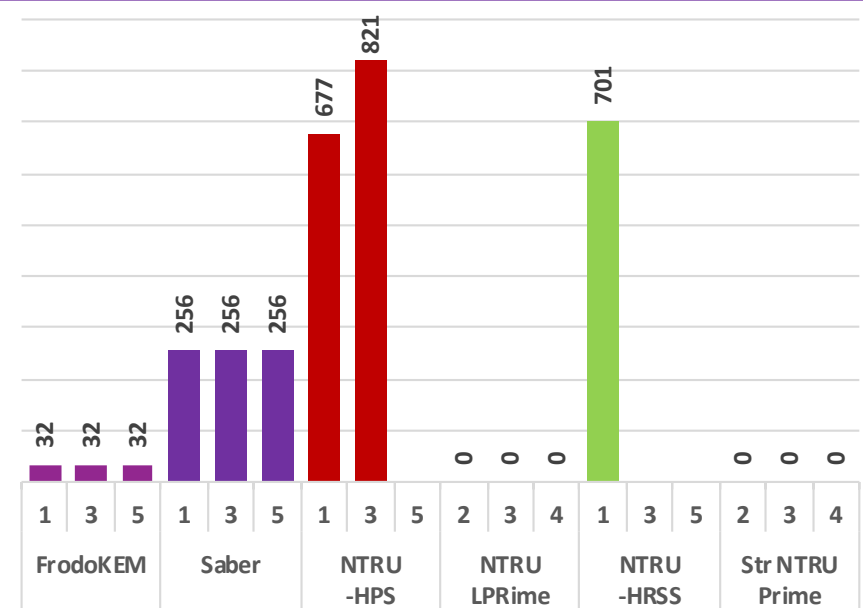


Resource Utilization on Zynq UltraScale+

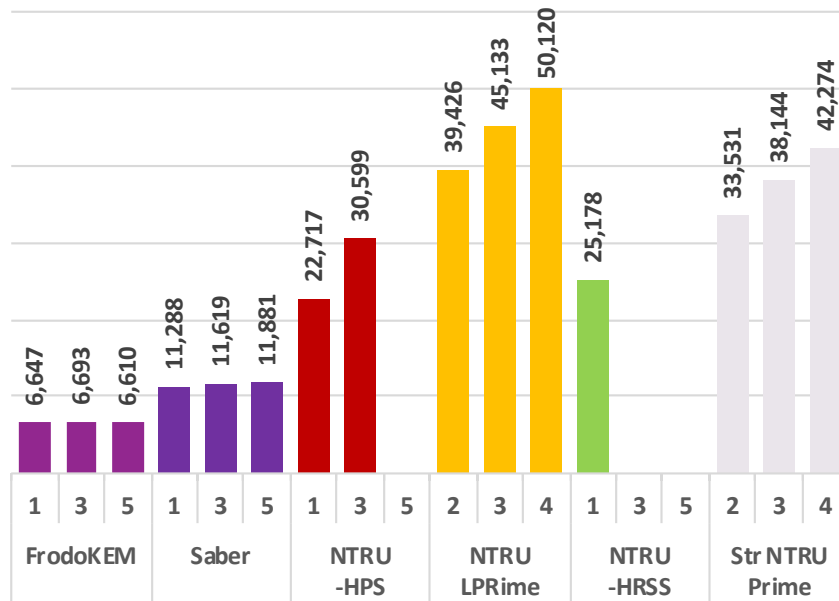
LUT



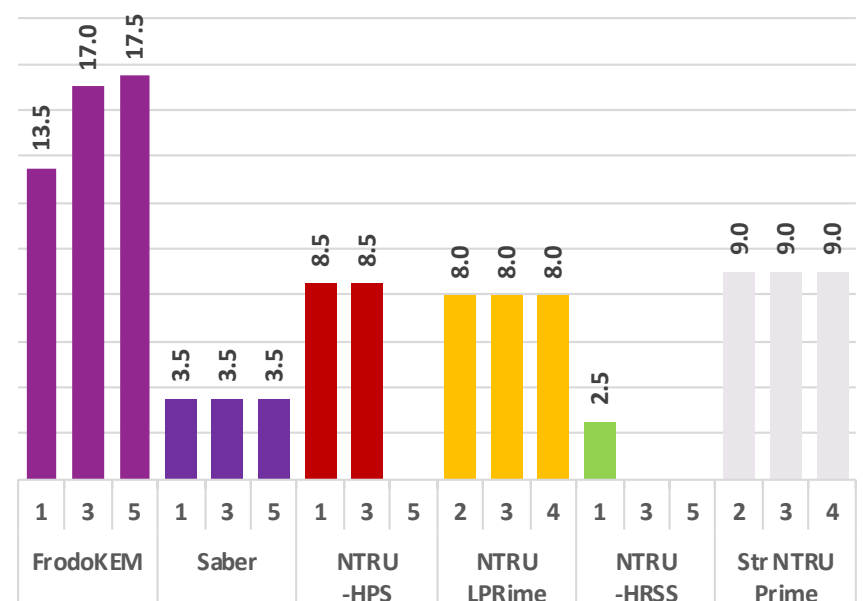
DSP



FF



BRAM





SW/HW Co-Design Conclusions

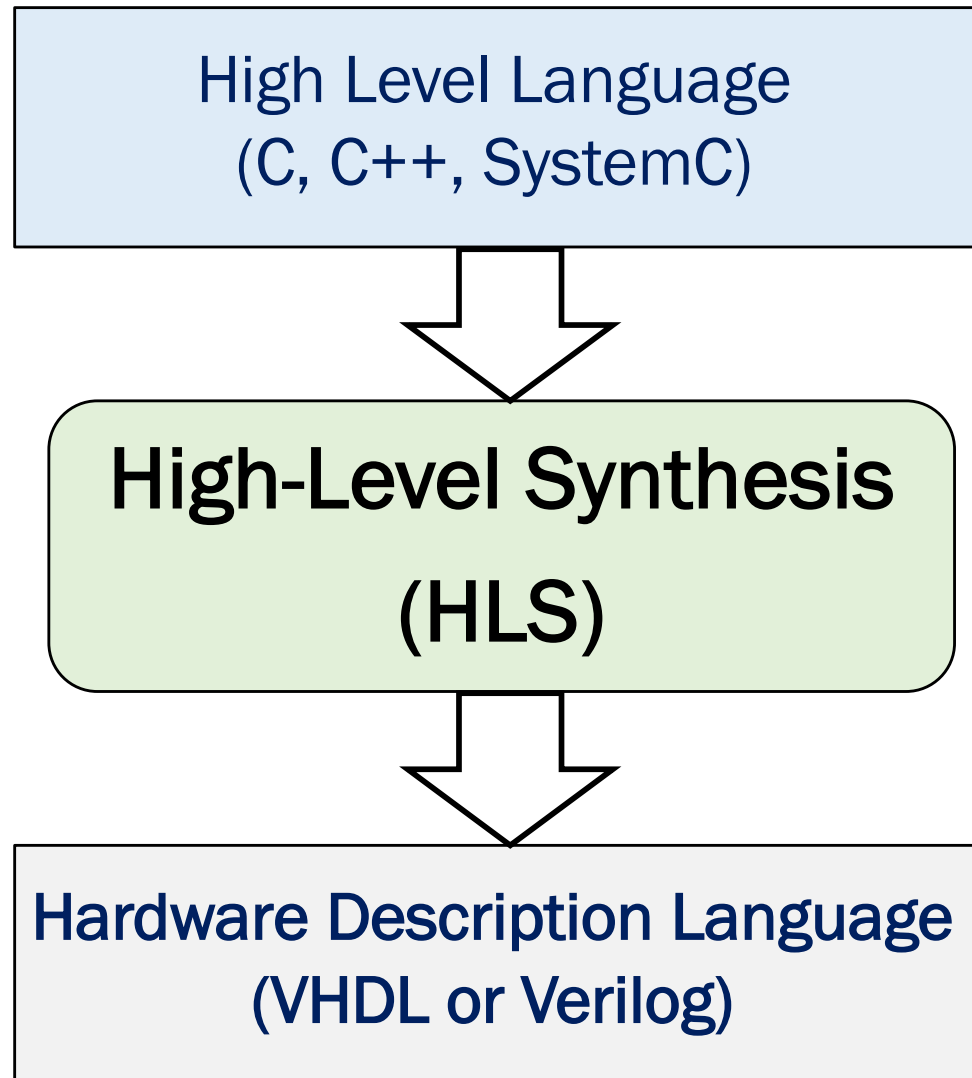
SW/HW Co-design: Conclusions

- ⦿ Unless all operations offloaded to hardware, limited insight on ranking of pure hardware implementations
- ⦿ FrodoKEM much slower than other lattice-based KEMs
- ⦿ Concerns regarding resource utilization:
 - ☆ NTRU-HPS and NTRU-HRSS : large number of DSP units
 - ☆ Streamlined NTRU Prime and NTRU LPrime : large number of LUTs (but no DSP units)
- ⦿ In SABER & FrodoKEM resource utilization almost independent of the security level
- ⦿ Very significant step toward the development of full hardware implementations



High-Level Synthesis

High-Level Synthesis (HLS)



Popular HLS Tools

Commercial (FPGA-oriented):

- Vivado HLS: Xilinx – selected for this study
- FPGA SDK for OpenCL: Intel

Academic:

- **Bambu:** Politecnico di Milano, Italy
- **DWARV:** Delft University of Technology, The Netherlands
- **GAUT:** Universite de Bretagne-Sud, France
- **LegUp:** University of Toronto, Canada

Case for HLS in Crypto Competitions

- All submissions include **reference implementations in C**
- **Development time** potentially **decreased several times**
- **All candidates** can be **implemented by the same** group, and even the same **designer**, reducing the bias
- Results from High-Level Synthesis could have a **large impact in early stages of the competitions** and help narrow down the search (saving thousands of man-hours of cryptanalysis)
- Potential for quickly **detecting suboptimal code written manually**

GMU Case Studies

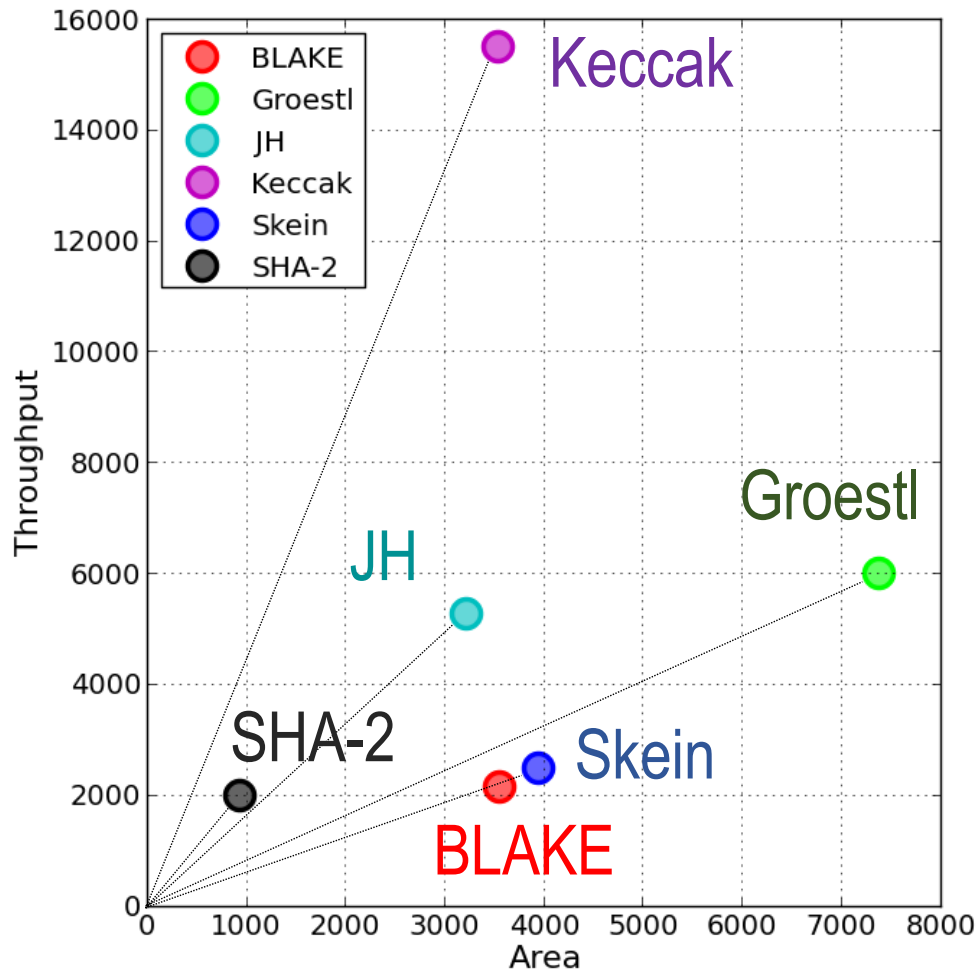
- **5 Final SHA_3** Candidates + SHA-2
Applied Reconfigurable Computing,
ARC 2015, Bochum, Apr. 2015
- **16 Round 3 CAESAR** Candidates
+ AES-GCM
Field Programmable Technology
Conference, Melbourne, Dec. 2017



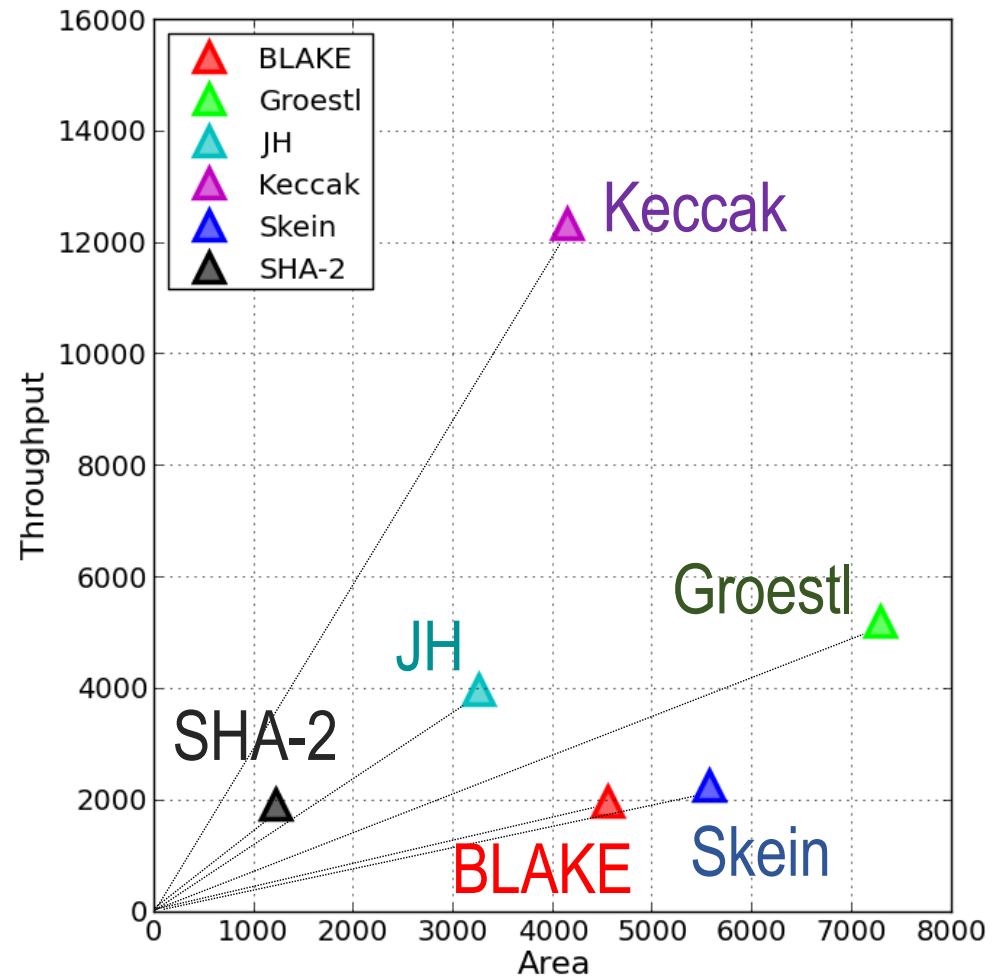
Ekawat Homsirikamol
a.k.a “Ice”

HLS vs. Manual: SHA-3 Candidates Revisited

Altera Stratix III FPGA



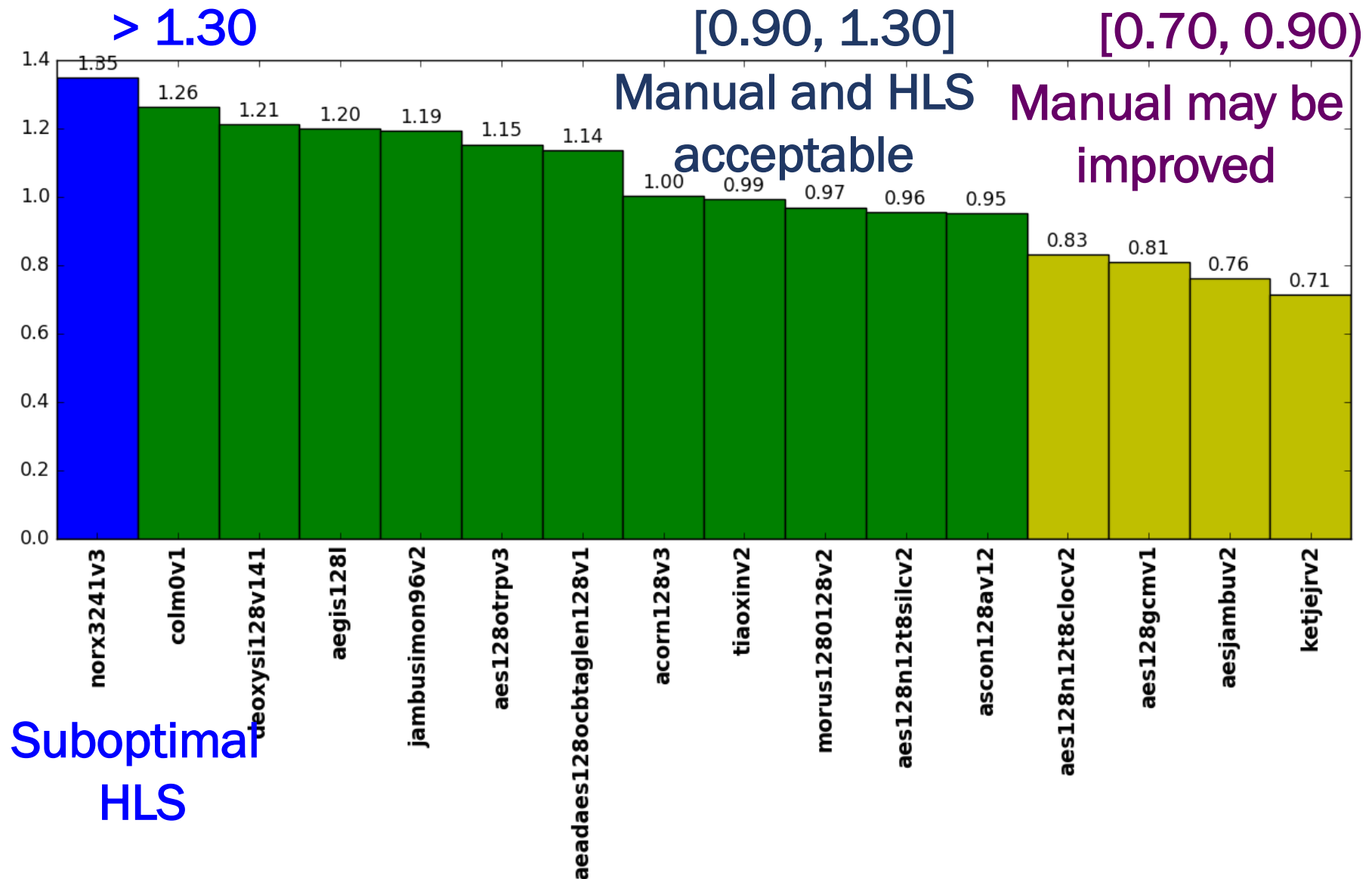
Manual



HLS

HLS vs. Manual: Round 3 CAESAR Candidates

Throughput Manual / Throughput HLS for Xilinx Virtex-7



Transformation to HLS-ready C/C++ Code

1. Interface mapping
2. Addition of HLS Tool directives (pragmas)
3. Hardware-driven code refactoring

Sources of Productivity Gains

- Higher-level of abstraction
- Focus on datapath rather than control logic
- Debugging in software (C/C++)
 - Faster run time
 - No timing waveforms



Software/Hardware Codesign with HLS

Software

HLS-Generated Hardware
**Most time-critical
operation**

3 Lattice-Based Key Encapsulation Mechanisms (KEMs) representing 2 NIST PQC Round 2 Submissions 1 NIST PQC Round 1 Submission

- CRYSTALS-KYBER
 - Round 2 (R2)
 - Round 1 (R1)
- NewHope
 - Round 2 (R2)

Major Findings

Almost identical number of clock cycles

Identical number of DSP units

Identical number of BRAMs
(except of 40% increase in Kyber R2)

Overhead: Clock Frequency [MHz]

Algorithm	RTL	HLS	HLS/RTL
1: NewHope	476	454	0.95
5: NewHope	476	455	0.96
1: Kyber R1	500	455	0.91
3: Kyber R1	500	455	0.91
5: Kyber R1	500	455	0.91
1: Kyber R2	500	455	0.91
3: Kyber R2	500	416	0.83
5: Kyber R2	500	416	0.83

Clock Frequency reduced by 17% or less

Overhead: LUTs

Algorithm	RTL	HLS	HLS/RTL
1: NewHope	1,040	1,181	1.14
5: NewHope	842	1,110	1.32
1: Kyber R1	2,185	2,788	1.28
3: Kyber R1	3,318	4,205	1.27
5: Kyber R1	4,363	5,562	1.27
1: Kyber R2	2,040	2,325	1.14
3: Kyber R2	3,054	5,379	1.76
5: Kyber R2	4,055	7,111	1.75

#LUTs increased by 14%-76% or less

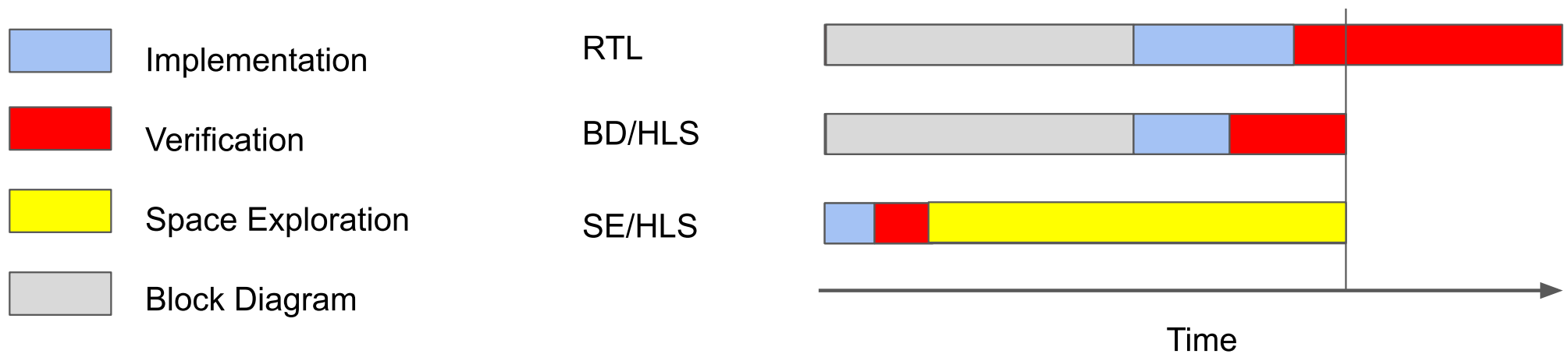
Comparison to the Previous Work in HLS

NTT only

K. Kawamura, M. Yanagisawa, and N. Togawa,
“A loop structure optimization targeting high-level synthesis of fast
number theoretic transform,”
in Int. Symposium on Quality Electronic Design, ISQED 2018.

Space-Exploration (SE) based vs. Block Diagram (BD) based approach

Time spent on particular phases of the development process:



Comparison to Previous Work in HLS

1024-point NTT only

Previous work optimized for area

	BRAMs	DSPs	LUTs	FFs	Cycles
Previous work	11.5	10	21,167	16,402	7,597
Our work	10	4	1,110	1,342	4,776
Ratio	1.15	2.5	19	12	1.6

Previous work optimized for speed

	BRAMs	DSPs	LUTs	FFs	Cycles
Previous work	21.5	19	38,984	30,498	5,291
Our work	10	4	1,110	1,342	4,776
Ratio	2.15	4.75	35	23	1.1

Additional Advantages of HLS vs. RTL

Easy integration of software and hardware
within the Xilinx SDSoC environment

No need for manually developed
Bare Metal or Linux drivers
for the communication between
the microprocessor and hardware accelerator

HLS/SDSoC vs. RTL/Bare Metal

Algorithm	Total SW (μ s)	Total SW NTT (μ s)	%SW NTT	Total SW/HW		Total Speed-up	
				RTL	(μ s) HLS	@Max Freq	
				BM	SDSoC	BM	SDSoC
ENCAPSULATION							
NewHope 1	360.3	199.8	55%	175.2	180.3	2.06	2.00
NewHope 5	737.0	438.1	59%	324.0	332.2	2.27	2.22
Kyber R1-1	389.2	240.9	62%	158.9	161.1	2.45	2.42
Kyber R1-3	582.3	368.3	63%	224.8	228.7	2.59	2.55
Kyber R1-5	826.9	509.4	62%	329.6	334.0	2.51	2.48
Kyber R2-1	328.5	237.8	72%	101.1	103.4	3.25	3.18
Kyber R2-3	533.9	343.0	64%	201.5	205.7	2.65	2.60
Kyber R2-5	785.2	495.4	63%	301.8	306.4	2.60	2.56

**Total SW/HW execution time
increased by 3% or less!**



Reality Check

NIST Announcement on July 22, 2020

Round 3 Candidates

FINALISTS	Encryption/KEM	Lattice-based <ul style="list-style-type: none">CRYSTALS-KYBERNTRUSABER	Code-based <ul style="list-style-type: none">Classic McEliece
	Digital Signature	Lattice-based <ul style="list-style-type: none">CRYSTALS-DILITHIUMFALCON	Multivariate <ul style="list-style-type: none">Rainbow

ALTERNATE	Encryption/KEM	Lattice-based <ul style="list-style-type: none">FrodoKEMNTRU Prime	Code-based <ul style="list-style-type: none">BIKEHQC	Isogeny-based <ul style="list-style-type: none">SIKE
	Digital Signature	Symmetric-based <ul style="list-style-type: none">PicnicSPHINCS+	Multivariate <ul style="list-style-type: none">GeMSS	

NIST Announcement on July 22, 2020

NISTIR 8309

“Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process,”

by Gorjan Alagic, Jacob Alperin-Sheriff, Daniel Apon, David Cooper, Quynh Dang, John Kelsey, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, and Daniel Smith-Tone

available <https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8309.pdf>

No references to papers on hardware implementations.

All decisions based solely on **security analysis**
and (to lower extent) **performance in software**.

NSA's Cybersecurity Perspective on PQC

Lattice-Based Cryptography

“These systems are fairly well-studied in cryptologic literature, and analysis suggests that these systems can be secure when well-parameterized.

We agree with the NIST assessment [...] that these are among the most efficient post-quantum designs.

Based on their history of analysis and implementation efforts, NSA CSD [Cybersecurity Directorate] expects that a NIST-candidate lattice-based signature and a NIST-candidate lattice-based key encapsulation mechanism will be approved for NSS [National Security Systems].”

NSA's Cybersecurity Perspective on PQC

Hash-Based Signatures

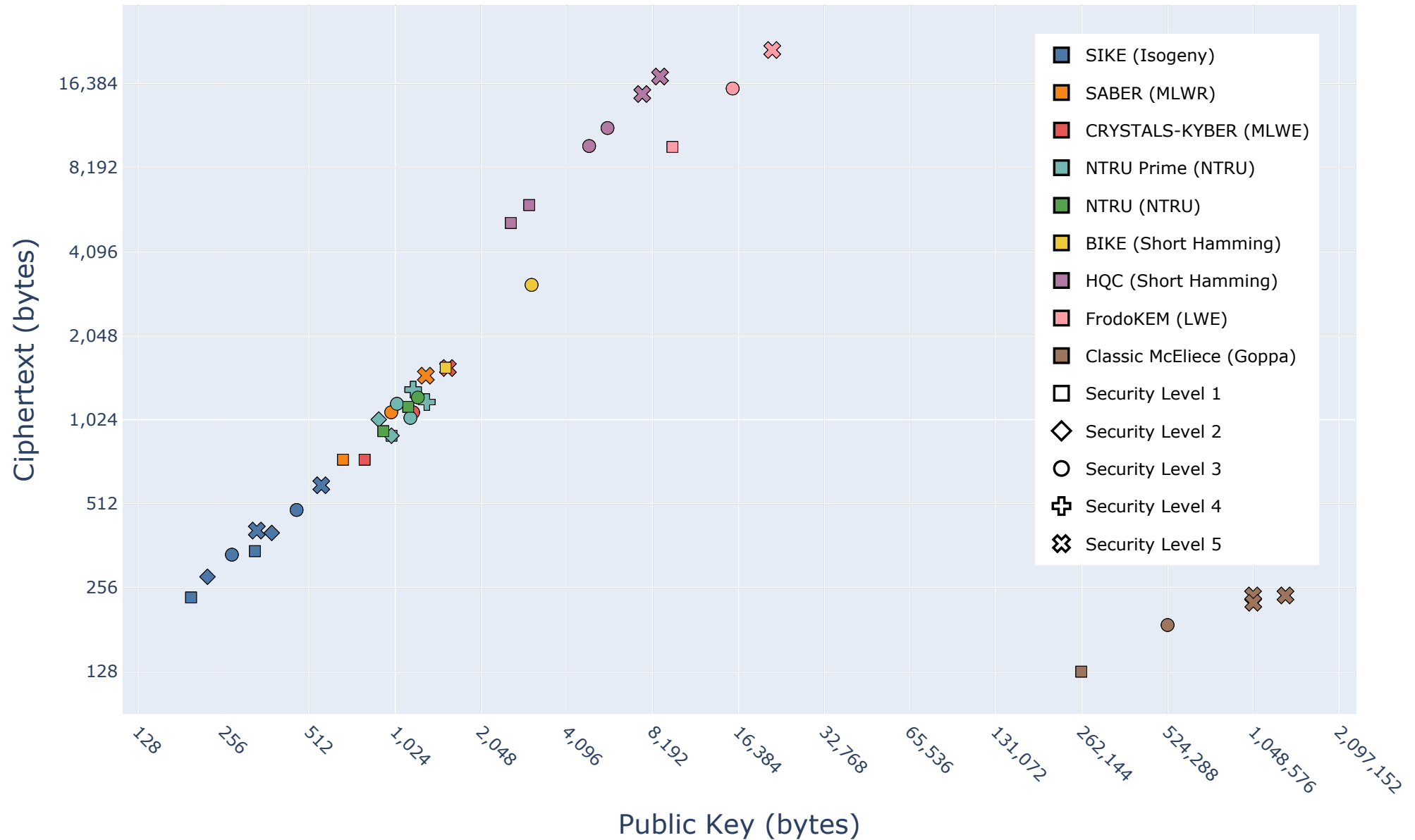
“These systems are also fairly well-studied in cryptologic literature, and analysis suggests that these systems can be secure when well-parameterized. However, the stateful versions have a limited number of allowable signatures per public key and require the signer to maintain an internal state. Because of this, they are not suitable for all applications. NSA CSD expects that the stateful signatures LMS and XMSS will be standardized by NIST in NIST SP 800-208 and approved for NSS solutions for certain niche applications where maintaining state is not a problem.”

NSA's Cybersecurity Perspective on PQC

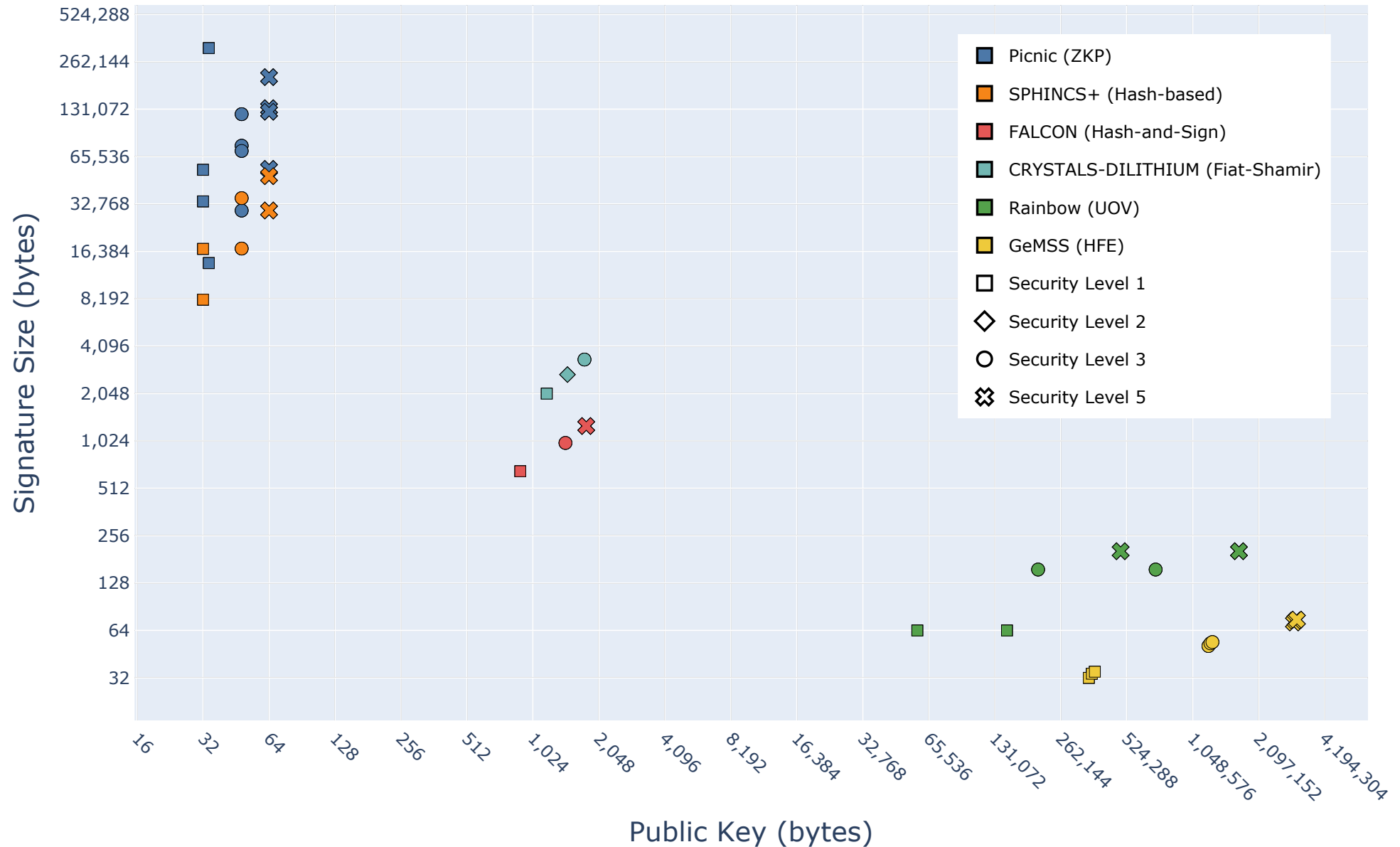
Future


“At the present time, NSA CSD does not anticipate the need to approve other post-quantum cryptographic technologies for NSS usage, but recognizes circumstances could change going forward. A variety of factors—including confidence in security and performance, interoperability, systems engineering, budgeting, procurement, and other requirements—could affect such decisions.”

Round 3 Key Encapsulation Mechanisms



Round 3 Digital Signature Schemes





Gazing
the PQC
Crystal
Ball
(use with caution!)

Candidates to Beat



PQCrypto 2017

The Eighth International Conference on Post-Quantum Cryptography

Utrecht, the Netherlands, June 26–28, 2017

Invited talk (chair: Andreas Hülsing)

09:00 –
10:00

Vadim Lyubashevsky

Standardizing Lattice Crypto and Beyond ([slides](#))

CRYSTALS

Cryptographic Suite for Algebraic Lattices



CRYPTOGRAPHIC SUITE FOR ALGEBRAIC LATTICES

KEM:

CRYSTALS-KYBER

Digital Signature:

CRYSTALS-DILITHIUM

Close Matchups

KEMs

CRYSTALS-KYBER

Module-LWE:
Module Learning
with Errors

SABER

Module-LWR:
Module Learning
with Rounding

NTRU

SVP
Shortest Vector
Problem

Digital Signatures

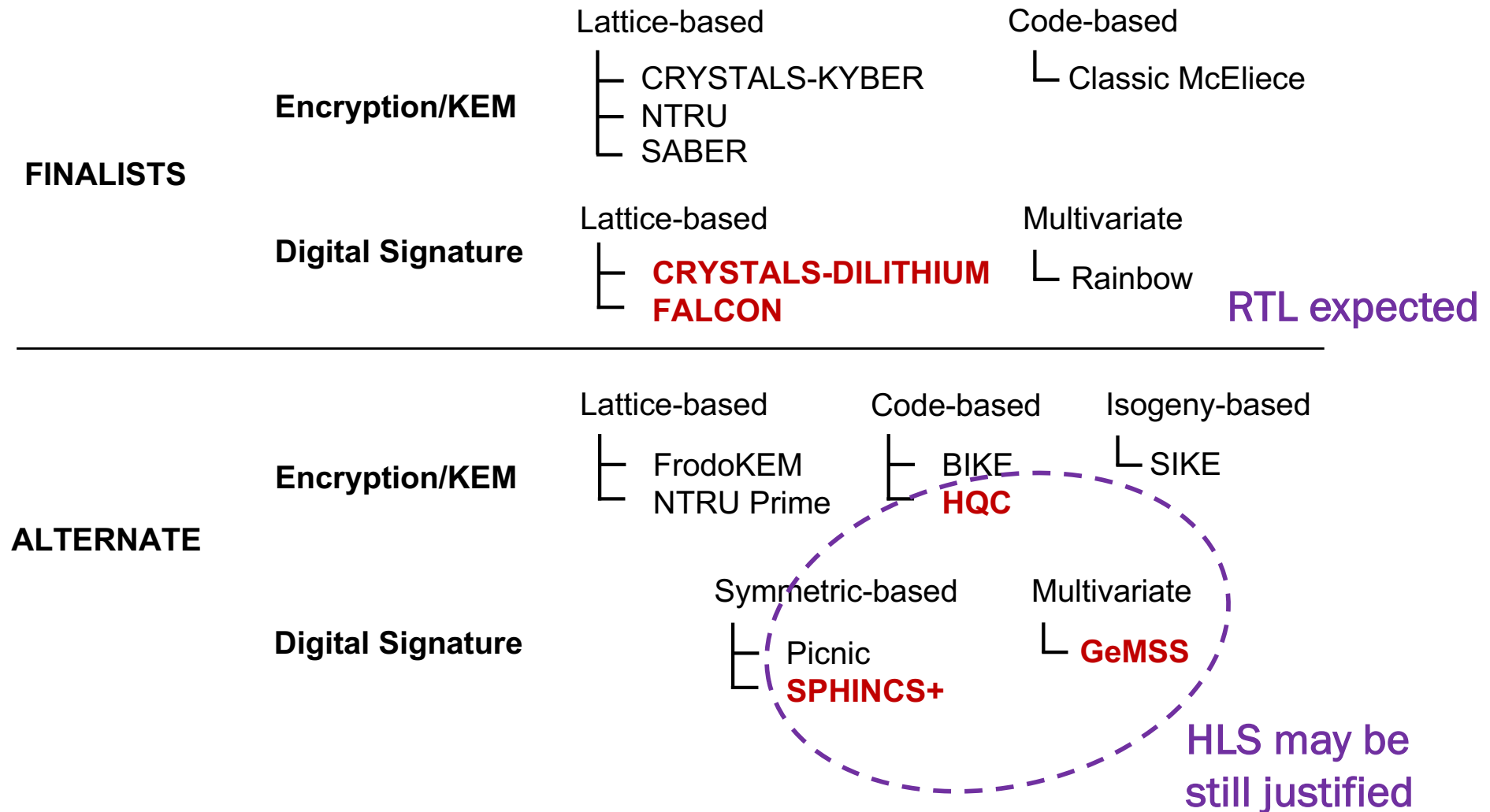
CRYSTALS-DILITHIUM

Fiat-Shamir with aborts
Module-LWE
& Module SIS
(Short Integer Solution)

FALCON

Hash & Sign
SIS
(Short Integer Solution)
over NTRU Lattices

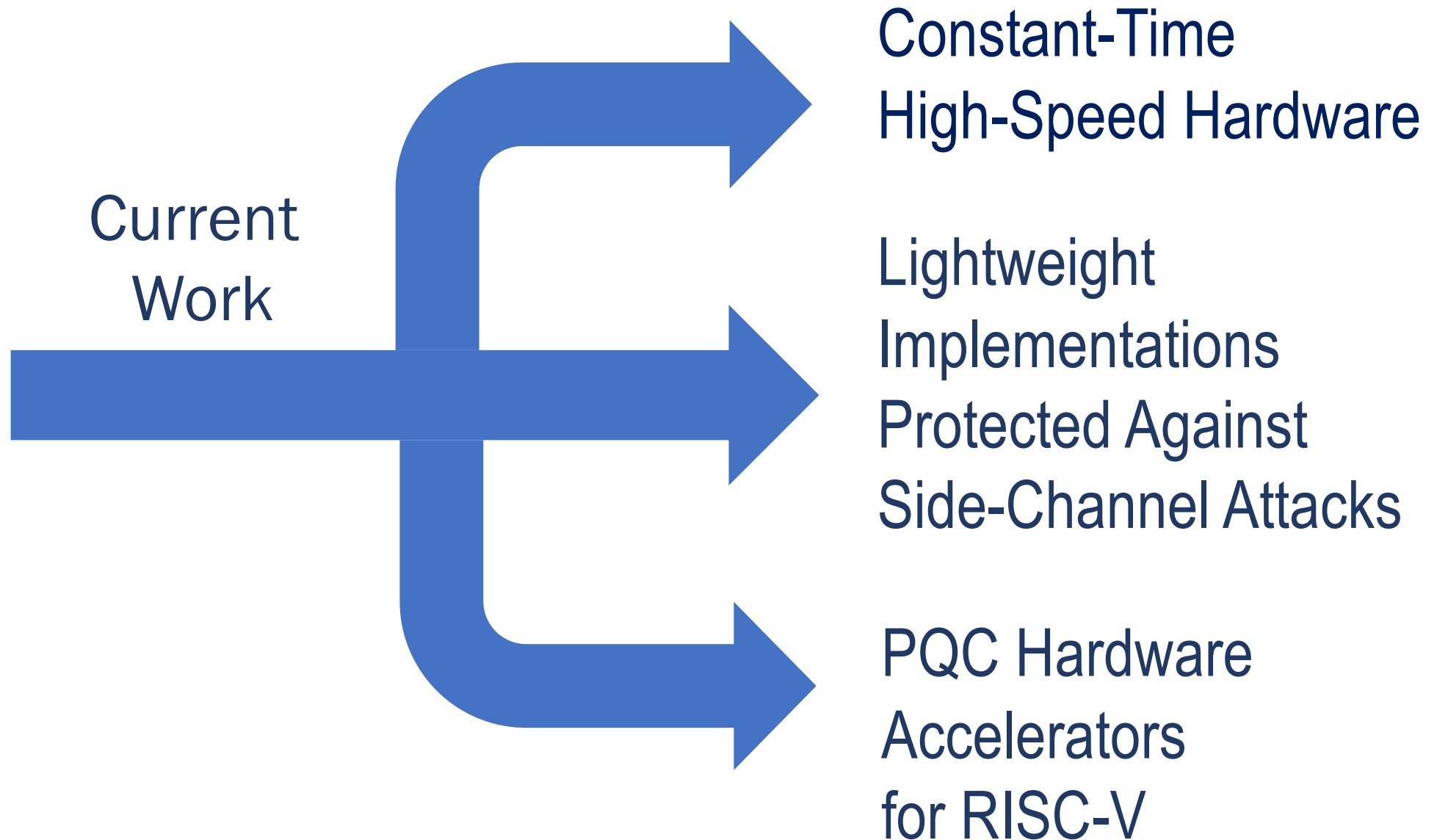
Round 3 Candidates without HW or SW/HW Implementations



Round 3 Candidates in Need of Improved Implementations

FINALISTS	Encryption/KEM	Lattice-based <ul style="list-style-type: none">CRYSTALS-KYBERNTRUSABER	Code-based <ul style="list-style-type: none">Classic McEliece	
	Digital Signature	Lattice-based <ul style="list-style-type: none">CRYSTALS-DILITHIUMFALCON	Multivariate <ul style="list-style-type: none">Rainbow	
ALTERNATE	Encryption/KEM	Lattice-based <ul style="list-style-type: none">FrodoKEMNTRU Prime	Code-based <ul style="list-style-type: none">BIKEHQC	Isogeny-based <ul style="list-style-type: none">SIKE
	Digital Signature	Symmetric-based <ul style="list-style-type: none">PicnicSPHINCS+	Multivariate <ul style="list-style-type: none">GeMSS	

Other Potential Research Directions



Upcoming Milestones

- **Oct. 1, 2020:** Deadline for **updated Round 3 submission packages**
- **Spring 2021:** Third **NIST PQC Conference**
- **Fall 2021:** Deadline for submitting comments
- **2022-2023:** **Draft standard(s)** released for public comments
- **2024:** First PQC **standard(s)** published

PQC Opportunities & Challenges

- Efficient **hardware implementations of Round 3 candidates** in FPGAs and ASICs **sought by NIST** to prove the final candidates' suitability for high-performance applications and constrained environments
- Potential **new standards in** other countries, including **China**
- Likely **Instruction Set Extensions** for multiple major microprocessors
- **First PQC industry trials**
- **Multiple opportunities for collaboration!**

Q&A

Thank You!

Questions?



Comments?

<https://eprint.iacr.org/2020/795>

CERG: <http://cryptography.gmu.edu>

ATHENa: <http://cryptography.gmu.edu/athena>

Choose: PQC