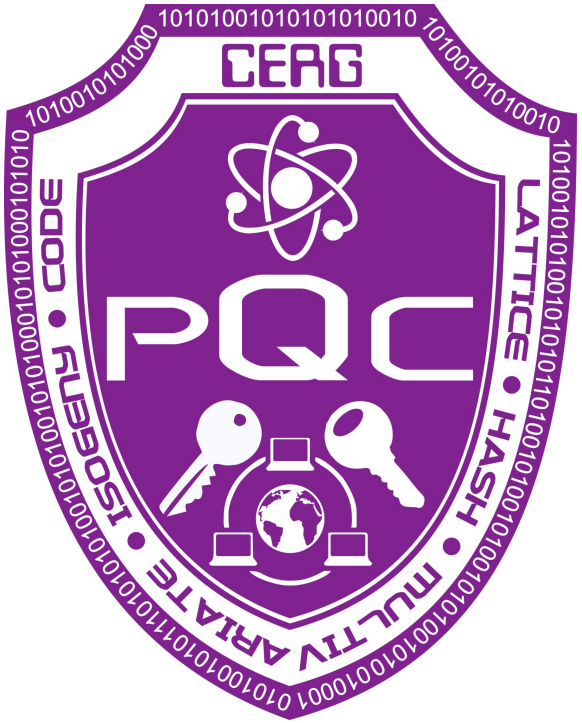
Implementation and Benchmarking of Round 2 Candidates in the NIST Post-Quantum Cryptography Standardization Process Using FPGAs



Kris Gaj

George Mason University

Thank You!

Great thanks to

- Dustin Moody
- Daniel Apon

for the kind invitation
to give this talk!

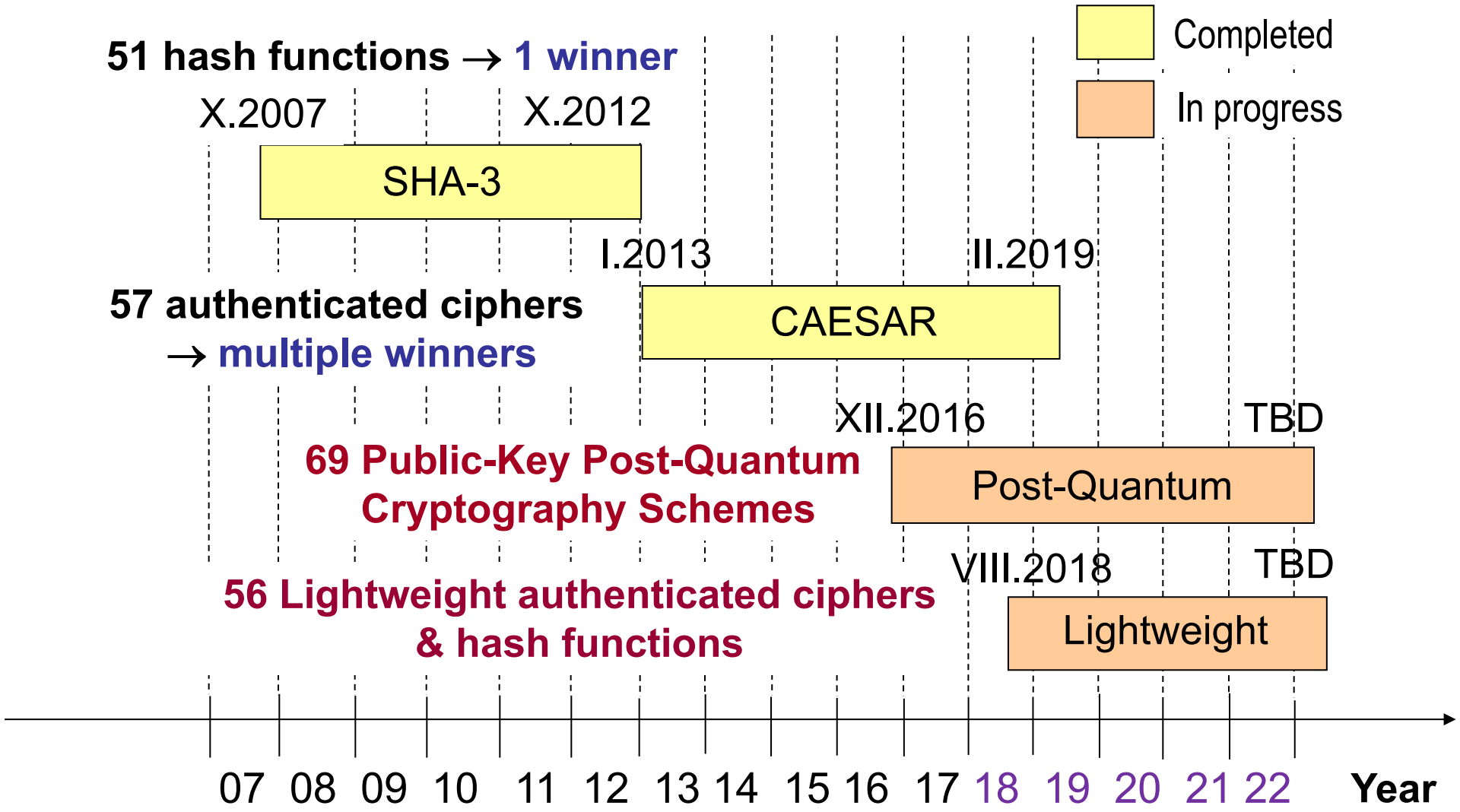
CERG: Cryptographic Engineering Research Group

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

CERG
Cryptographic Engineering Research Group

CERG
Cryptographic Engineering Research Group

Cryptographic Contests 2007-Present



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

Bakry

Experimental Setup for Side-Channel Analysis

Lightweight Architectures

Javad

RTL Design of HW Accelerators for Symmetric-based PQC

Affiliated Scholar

Michał

Military University of Technology in Warsaw, Poland

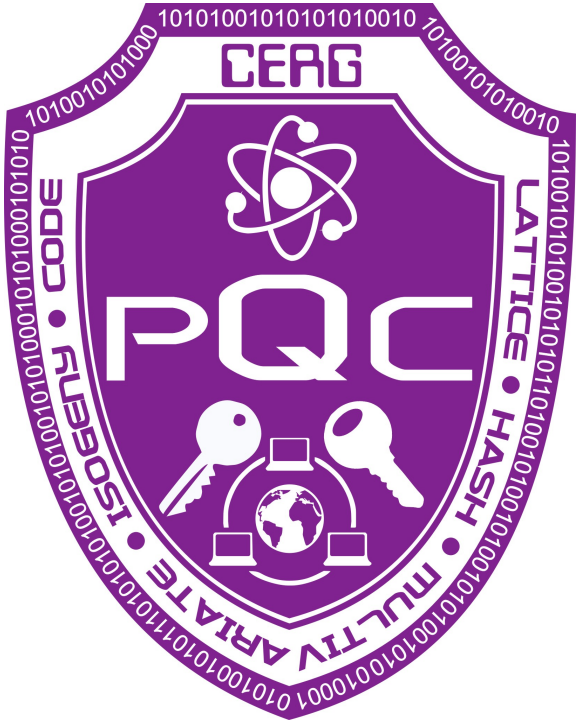
RTL Design of HW Accelerators for Lattice-based PQC & Lattice Sieving

Faculty

Mike

Sampling in Hardware

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



Evaluation Criteria

Security

Software Efficiency

µProcessors µControllers

Hardware Efficiency

FPGAs ASICs

Flexibility

Simplicity

Licensing

Talk Based on GMU Round 2 Report

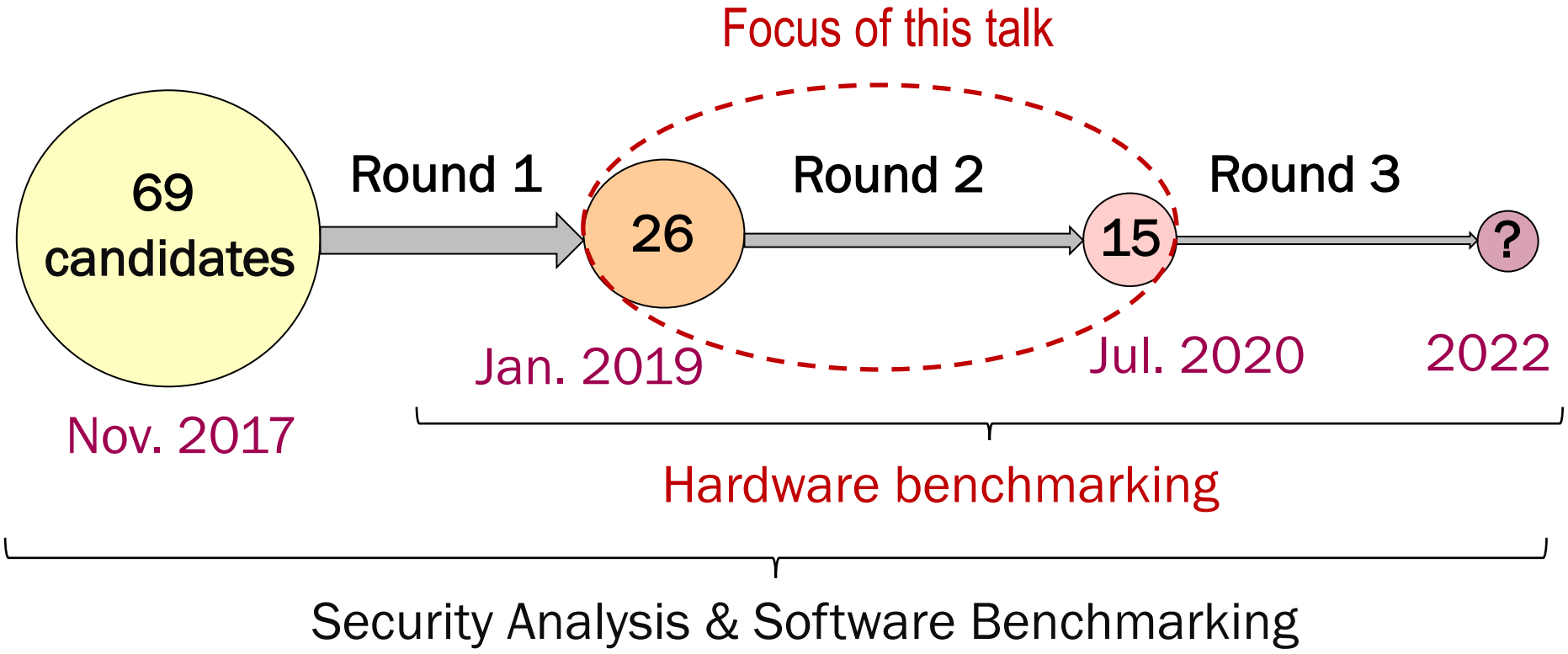
Cryptology ePrint Archive: Report 2020/795

"Implementation and Benchmarking of Round 2 Candidates in the NIST Post-Quantum Cryptography Standardization Process Using Hardware and Software/Hardware Co-design Approaches,"

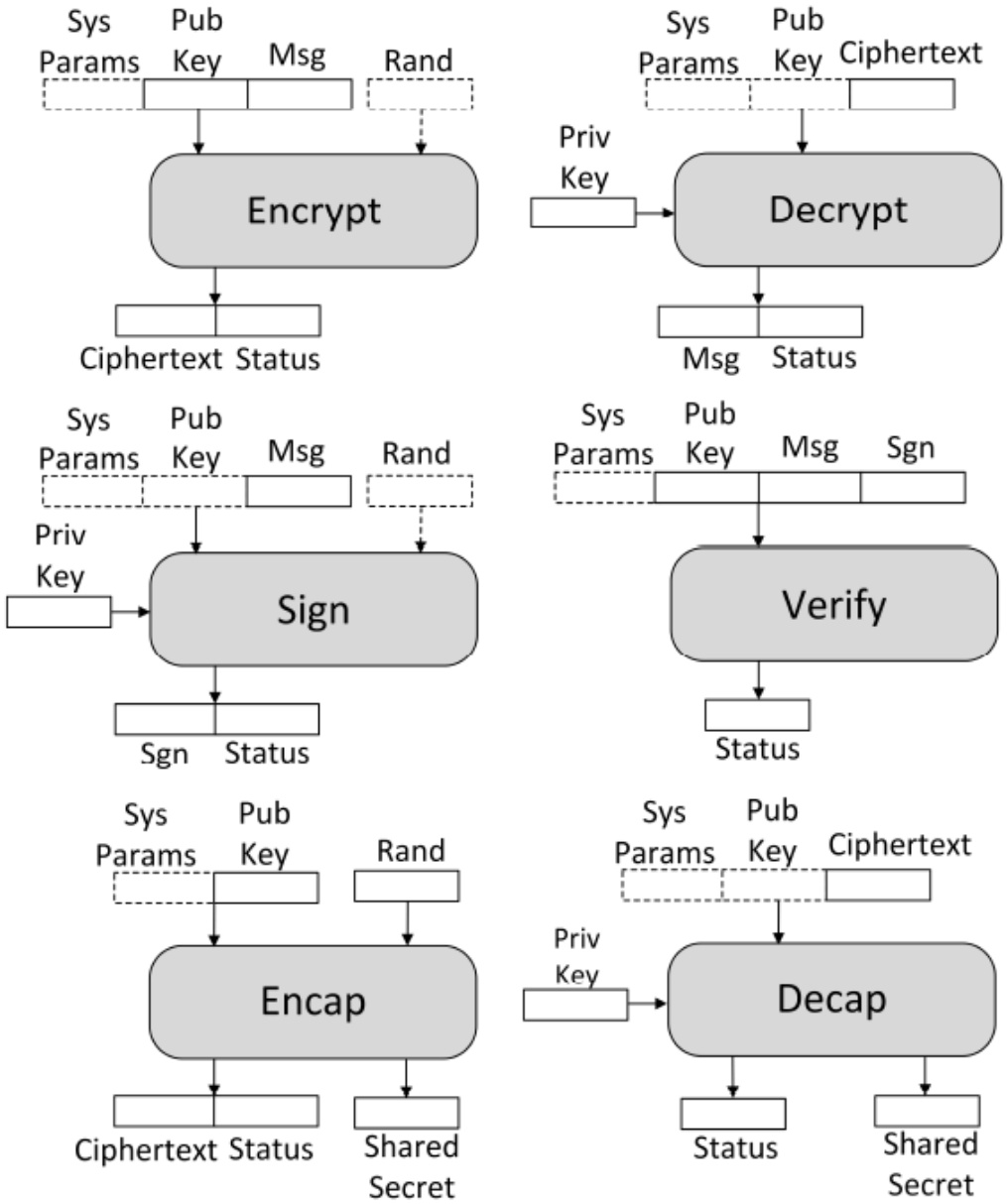
by Viet Ba Dang, Farnoud Farahmand, Michał Andrzejczak, Kamyar Mohajerani, Duc Tri Nguyen, and Kris Gaj

- 86 pages
- Extensive literature review (16 pages, 11 tables)
- New unpublished results from GMU
- Focus on methodology and rankings
- No details of hardware architectures (to be included in the follow-up conference/journal papers)

NIST PQC Standardization Process



Three Types of PQC Schemes



1. Public Key Encryption (PKE)

2. Digital Signature (DS)

3. Key Encapsulation Mechanism (KEM)

Round 2 Submissions (announced Jan. 30, 2019)

- Lattice-based
- Code-based
- Isogenies

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

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

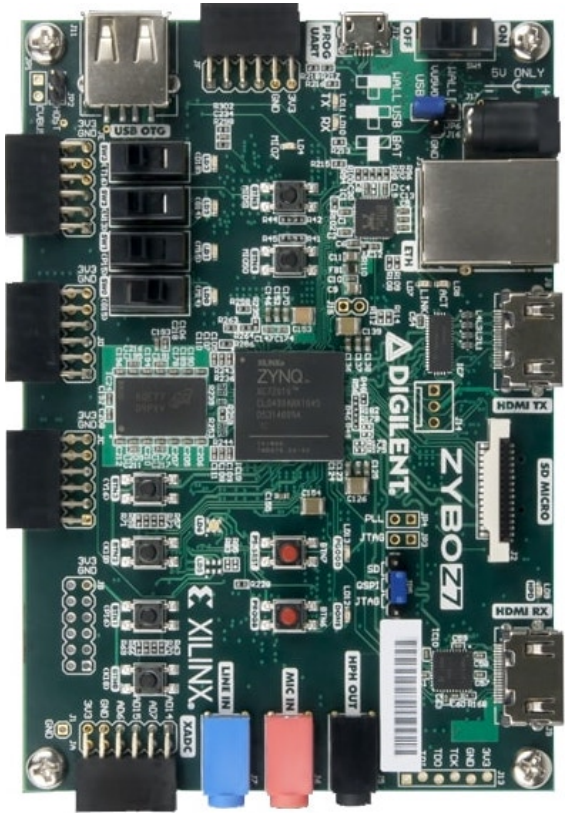
Sources: Moody, PQCrypto May 2019

Five Security Levels

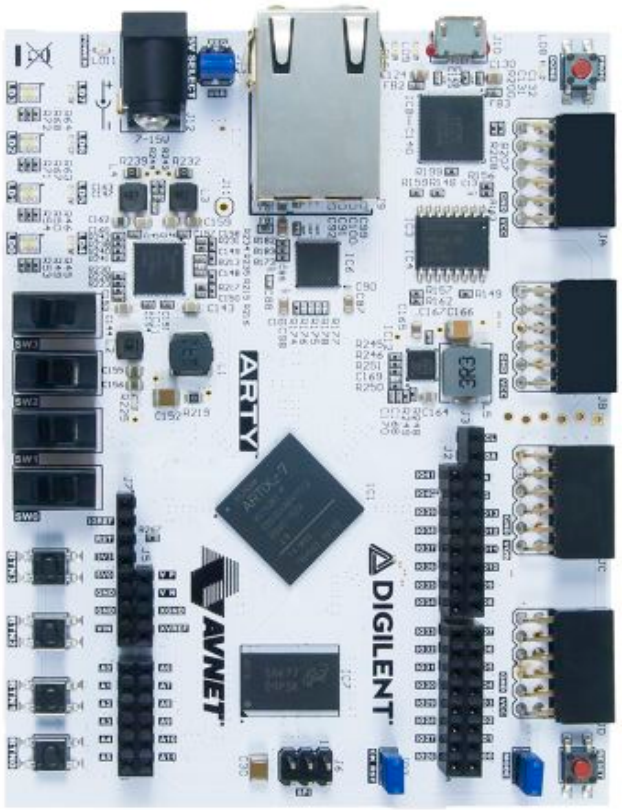
Level	Security Description
1	At least as hard to break as AES-128 using exhaustive key search
2	At least as hard to break as SHA-256 using collision search
3	At least as hard to break as AES-192 using exhaustive key search
4	At least as hard to break as SHA-384 using collision search
5	At least as hard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


Software,
Hardware,
Software/Hardware
Benchmarking

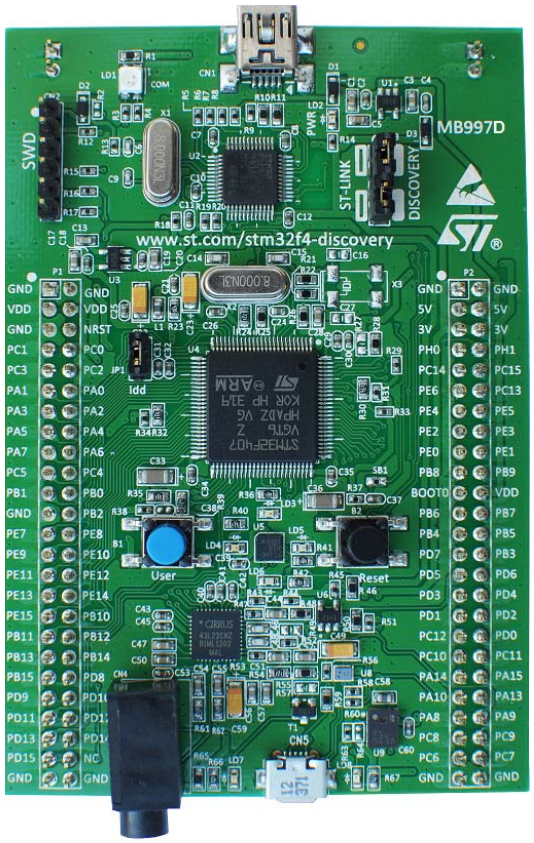
Software, Hardware, or Software/Hardware?



A



B

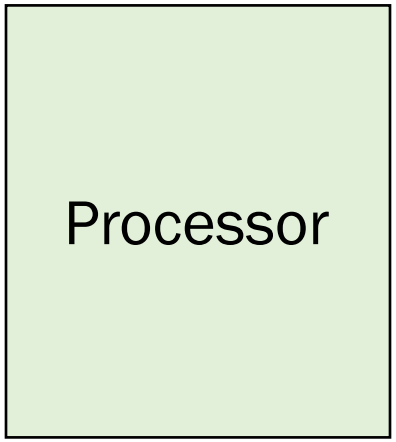


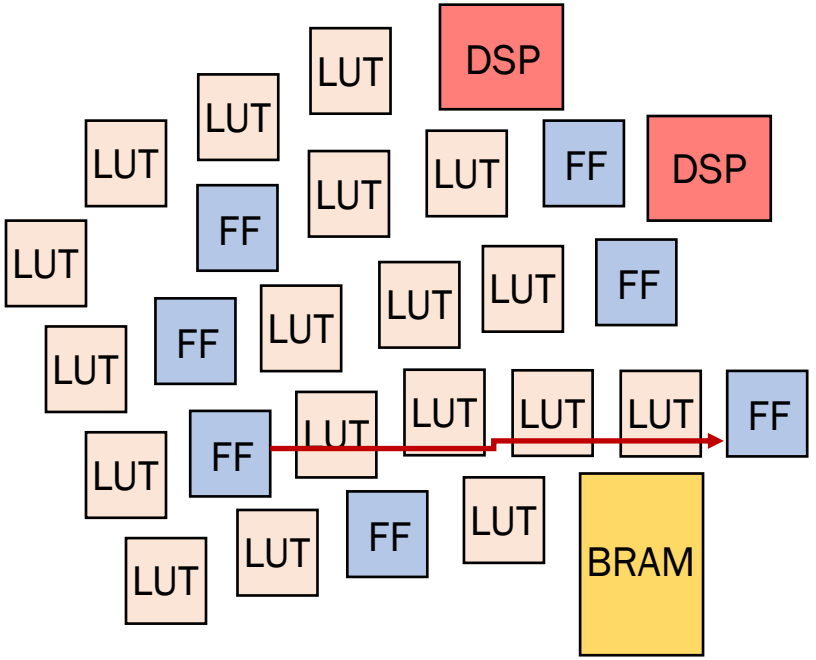
C

What NIST wants

- Performance (hardware+software) will play more of a role
  - More benchmarks
  - For hardware, NIST asks to focus on Cortex M4 (with all options) and Artix-7
    - pqc-hardware-forum
  - How do schemes perform on constrained devices?
  - Side-channel analysis (concrete attacks, protection, etc…)
- Continued research and analysis on **ALL** of the 2nd round candidates
- See how submissions fit into applications/procotols. Any constraints?

Software vs. Hardware

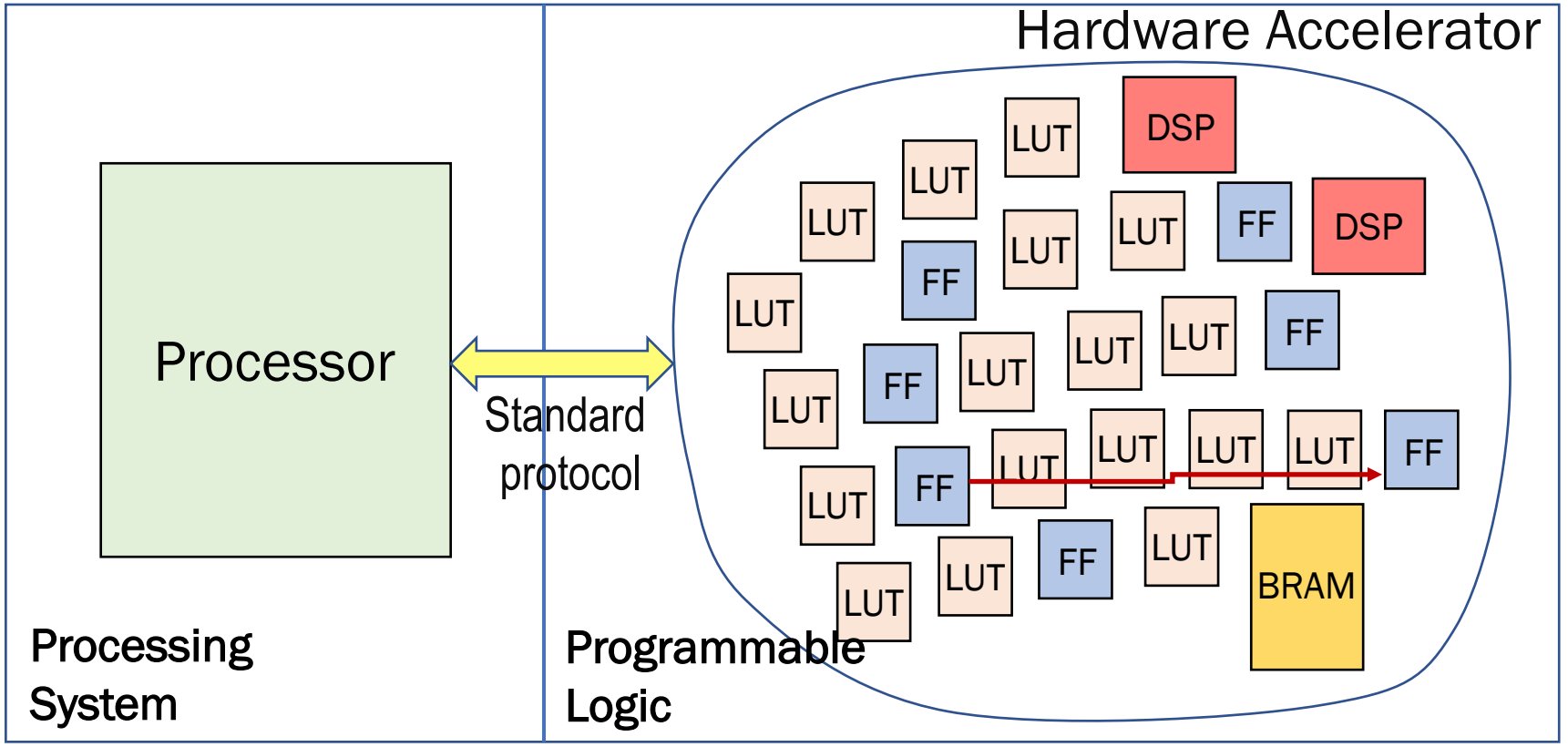




- Program composed of a sequence of assembly language instructions
- Instruction set and thus assembly language varies depending on a processor
- Clock period independent of an application
- Time measured in clock cycles

- Circuit composed of arbitrary number of arbitrarily connected basic components
- Two most-popular languages, VHDL and Verilog , common for all modern HW platforms
- Clock period strongly dependent on an application
- Time measured in units of time [μs, ms, s]

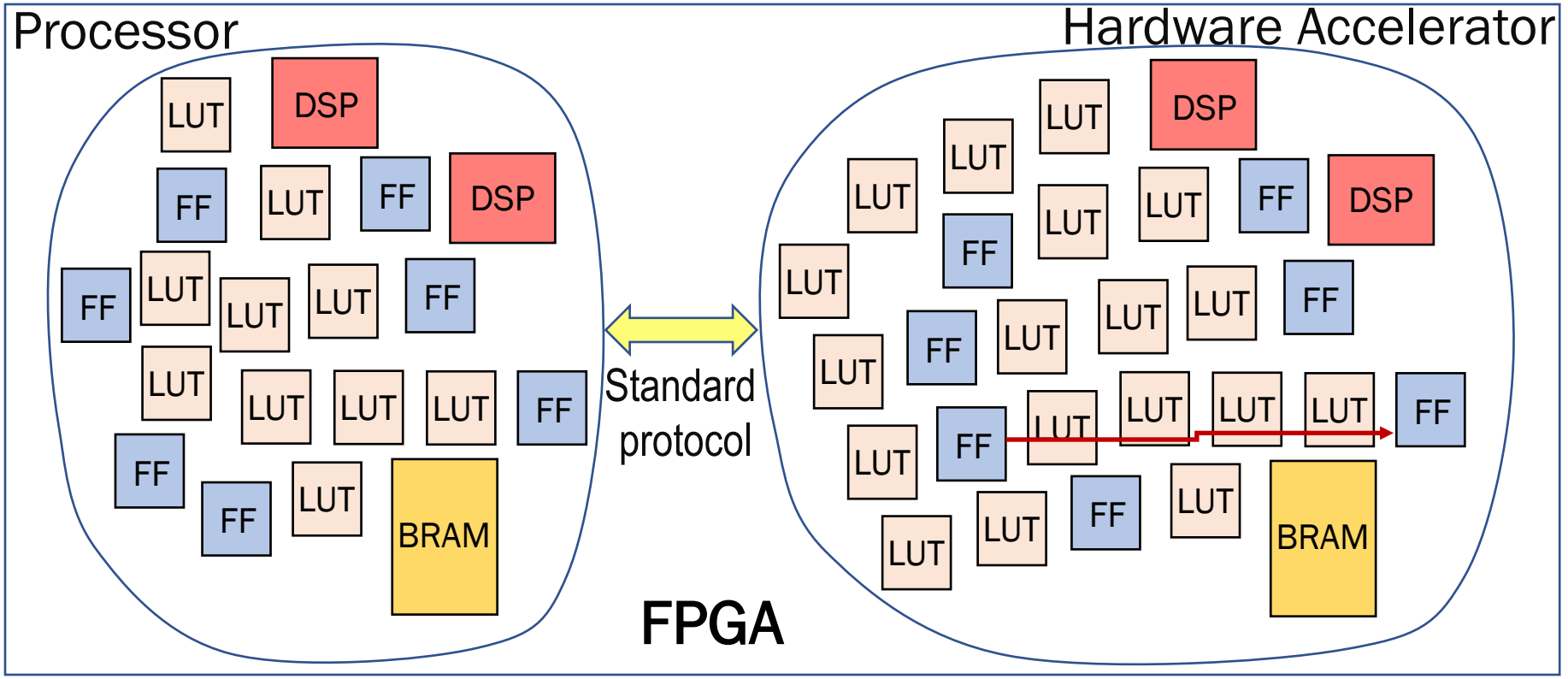
Software/Hardware with Hard Processor Cores



SoC FPGA

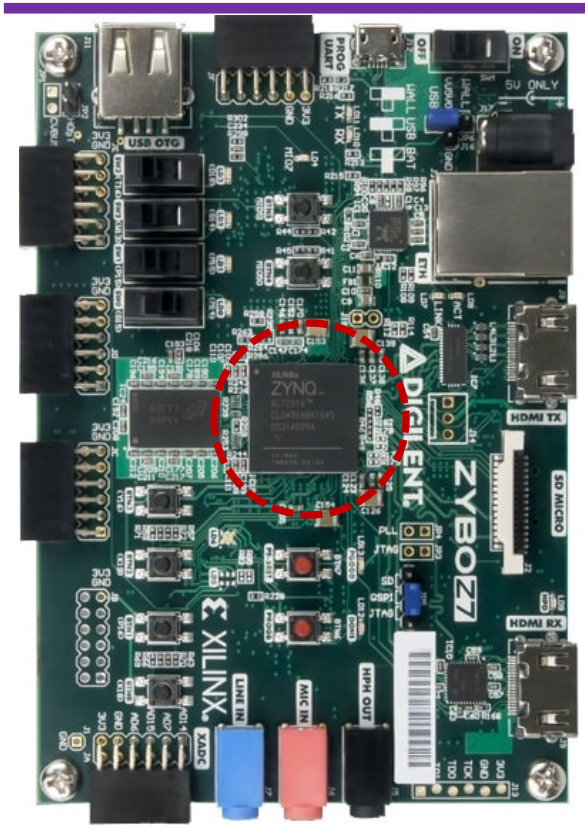
- Processor and hardware accelerator located on the same chip, often called SoC FPGA, such as Zynq UltraScale+, Zynq 7000, etc.
- Popular processors: ARM Cortex-A53, ARM Cortex-M9
- Processor clock speed > 1 GHz, independent of application
- Hardware accelerator clock speed < 400 MHz, dependent on application

Software/Hardware with Soft Processor Cores



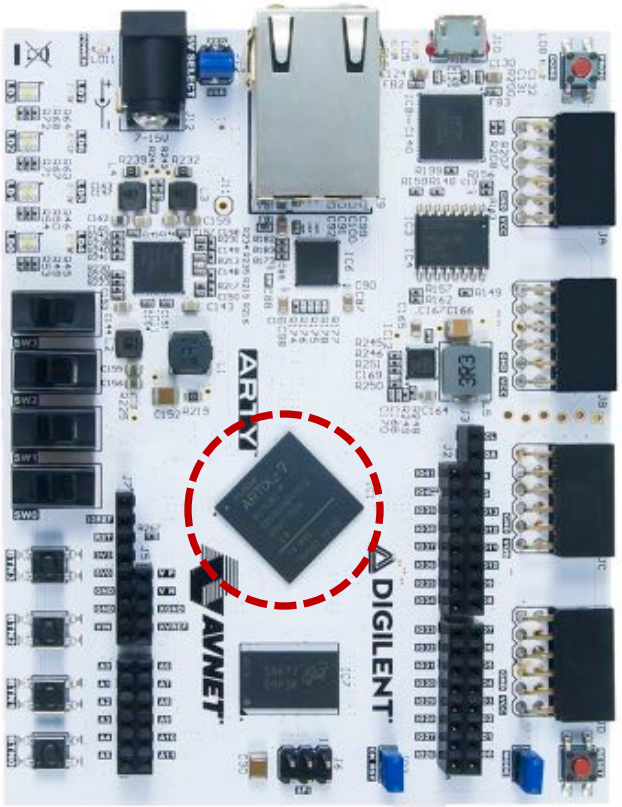
- Processor and hardware accelerator located on the same chip, e.g., the same FPGA, such as Artix-7, Virtex-7, UltraScale+, etc.
- Popular processors: RISC-V, MicroBlaze
- Processor clock speed < 400 MHz, independent of application
- Hardware accelerator clock speed < 400 MHz, dependent on application

Software, Hardware, or Software/Hardware?



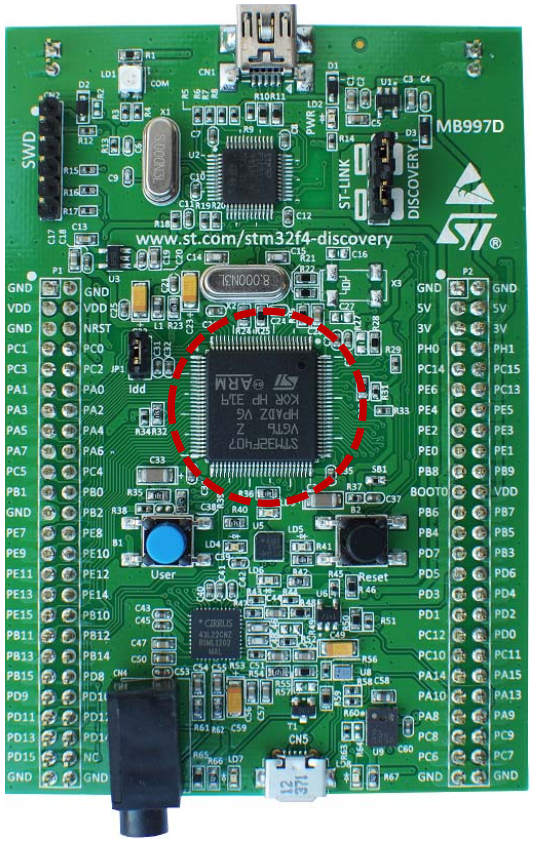
SoC FPGA

Software/Hardware with ARM Cortex



Boards containing FPGA

Hardware Benchmarking
Software/Hardware with RISC-V



Microcontroller

Software Benchmarking

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 PKE/KEMs

	High-Speed		Lightweight	
	HW	SW/HW	HW	SW/HW
KYBER	1+1	1+1		3
FrodoKEM	1	1		1
LAC	1	2+1		
NewHope	2+1	2+1		3
NTRU		1		
NTRUPrime		1	1	
Round5	1	1	1	
SABER	2+1	1+1		1
Three Bears				
TOTAL	6 candidates	8 candidates	2 candidates	4 candidates
TOTAL	8 out of 9 candidates		5 out of 9 candidates	

1 : Designs by GMU

Lattice-Based PKE/KEMs

	High-Speed	Lightweight
KYBER	H: Nanjing U. of Aero- and Astronautics, China + U. Arkansas, USA + ShanghaiTech U., China H, SH: GMU, USA SH: Fudan U., China; (VPQC)	SH: MIT, USA (Sapphire) 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, SH: GMU, USA SH: TUM, AIRBUS, Germany SH: Fudan U., China (VPQC)	
NewHope	H: Tsinghua, China H: IIIT Delhi/IIT Ropar, India + NTU/Fraunhofer Singapore H, SH: 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)

Lattice-Based PKE/KEMs

	High-Speed	Lightweight
NTRU	H, SH: GMU, USA	
NTRUPrime	H, SH: GMU, USA	H: TU Hamburg, NXP, Germany
Round5	H, SH: MUT, Warsaw, Poland + GMU, USA	H: MUT, Warsaw, Poland
SABER	H: U. Birmingham, UK, H: Tsinghua, China H, SH: GMU, USA SH: KU Leuven, Belgium + U. Birmingham, UK	SH: TUM+Airbus, Germany (RISQ-V)

Isogeny-Based and Code-Based PKE/KEMs

	High-Speed		Lightweight	
	HW	SW/HW	HW	SW/HW
Isogeny-based				
SIKE	2	1		1
Code-based				
BIKE	3			
Classic McEliece/ NTS KEM	1			
HQC	1			
LEDACrypt			1	
ROLLO				
RQC				
TOTAL	4 candidates	1 candidate	1 candidate	1 candidates
TOTAL	5 out of 8 candidates		2 out of 8 candidates	

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	
HQC	H (HLS): HQC Team	
LEDACrypt		H: NTU, Singapore + Marche Polytechnic U., Italy
ROLLO		
RQC		

Digital Signatures

	High-Speed		Lightweight	
	HW	SW/HW	HW	SW/HW
Lattice-based				
DILITHIUM				1
FALCON				
qTESLA				2
Symmetric-based				
Picnic	1			
SPHINCS+				
Multivariate				
GeMSS				
LUOV				
MQDSS				
Rainbow	1			
TOTAL	2 candidates	0 candidates	0 candidates	2 candidates
TOTAL	4 out of 9 candidates		2 out of 9 candidates	

Digital Signatures

	High-Speed	Lightweight
Lattice-Based		
DILITHIUM		SH: MIT, USA (Sapphire)
FALCON		
qTESLA		SH: MIT, USA (Sapphire) 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	

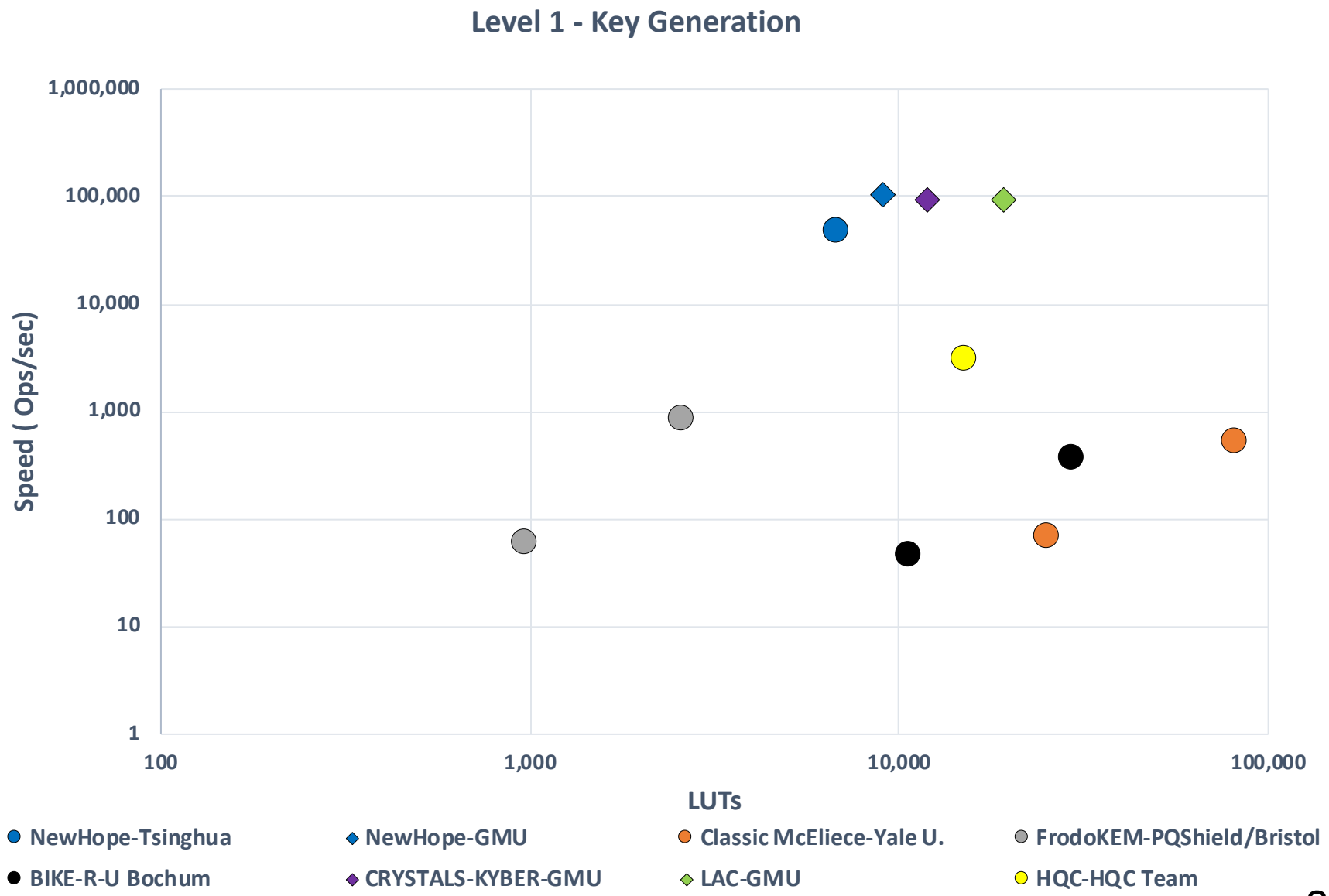
Round 2 Candidates in Hardware

	#Round 2 candidates	Implemented in hardware	Percentage
AES	5	5	100%
SHA-3	14	14	100%
CAESAR	29	28	97%
PQC	26	15	58%

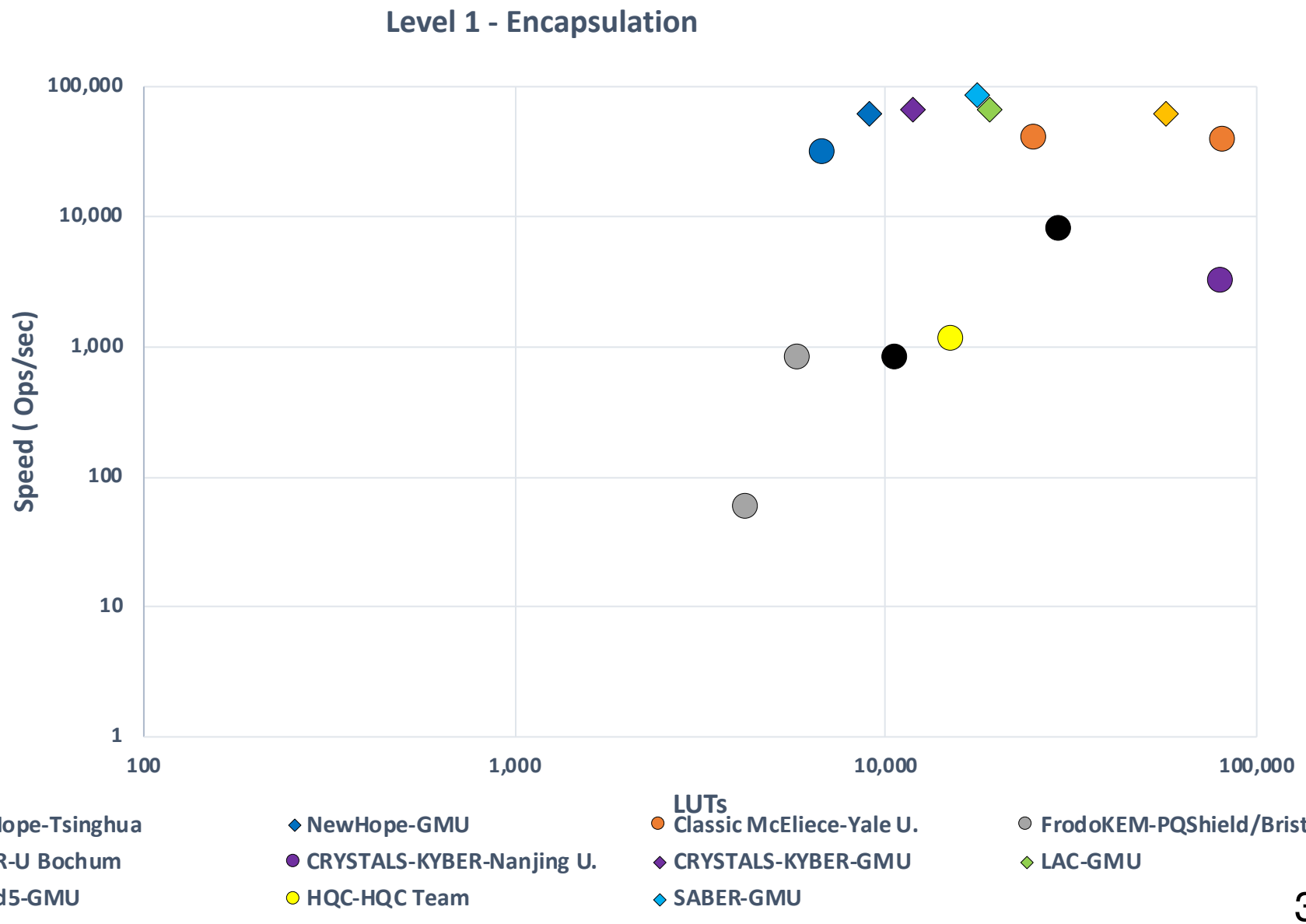
Why so few?

- 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
- Challenges in publishing of results

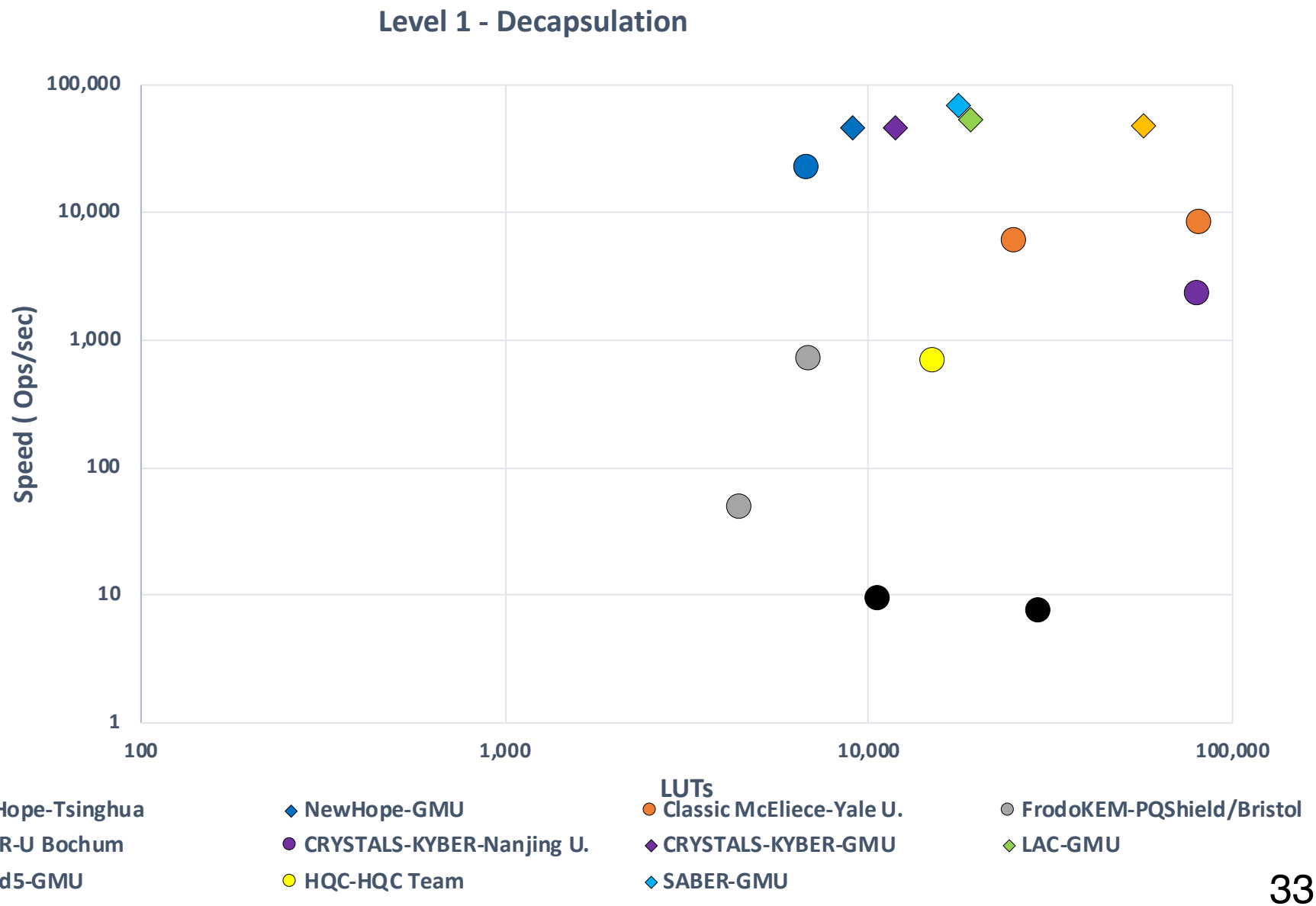
Level 1: Key Generation on Artix-7



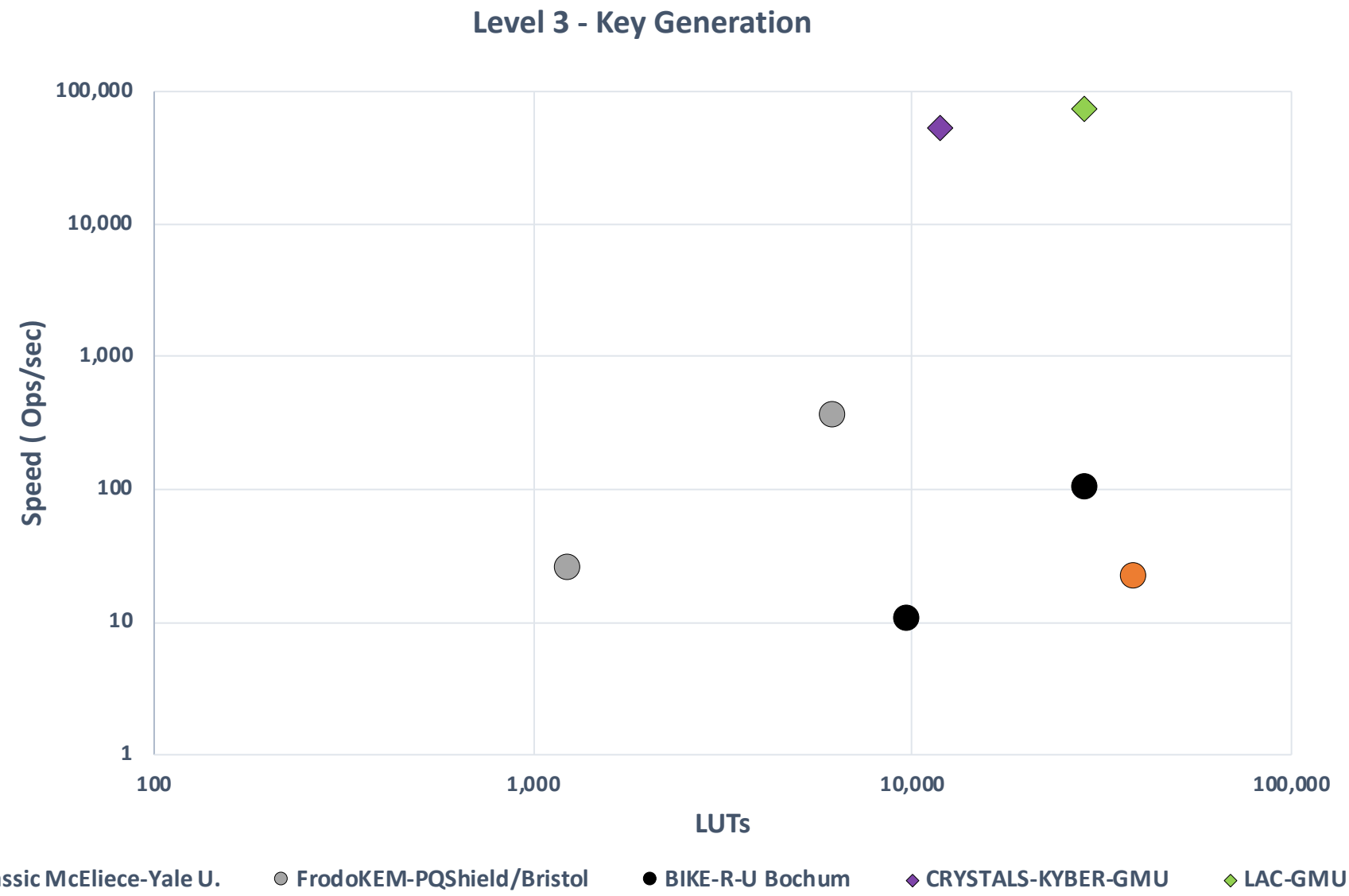
Level 1: Encapsulation on Artix-7



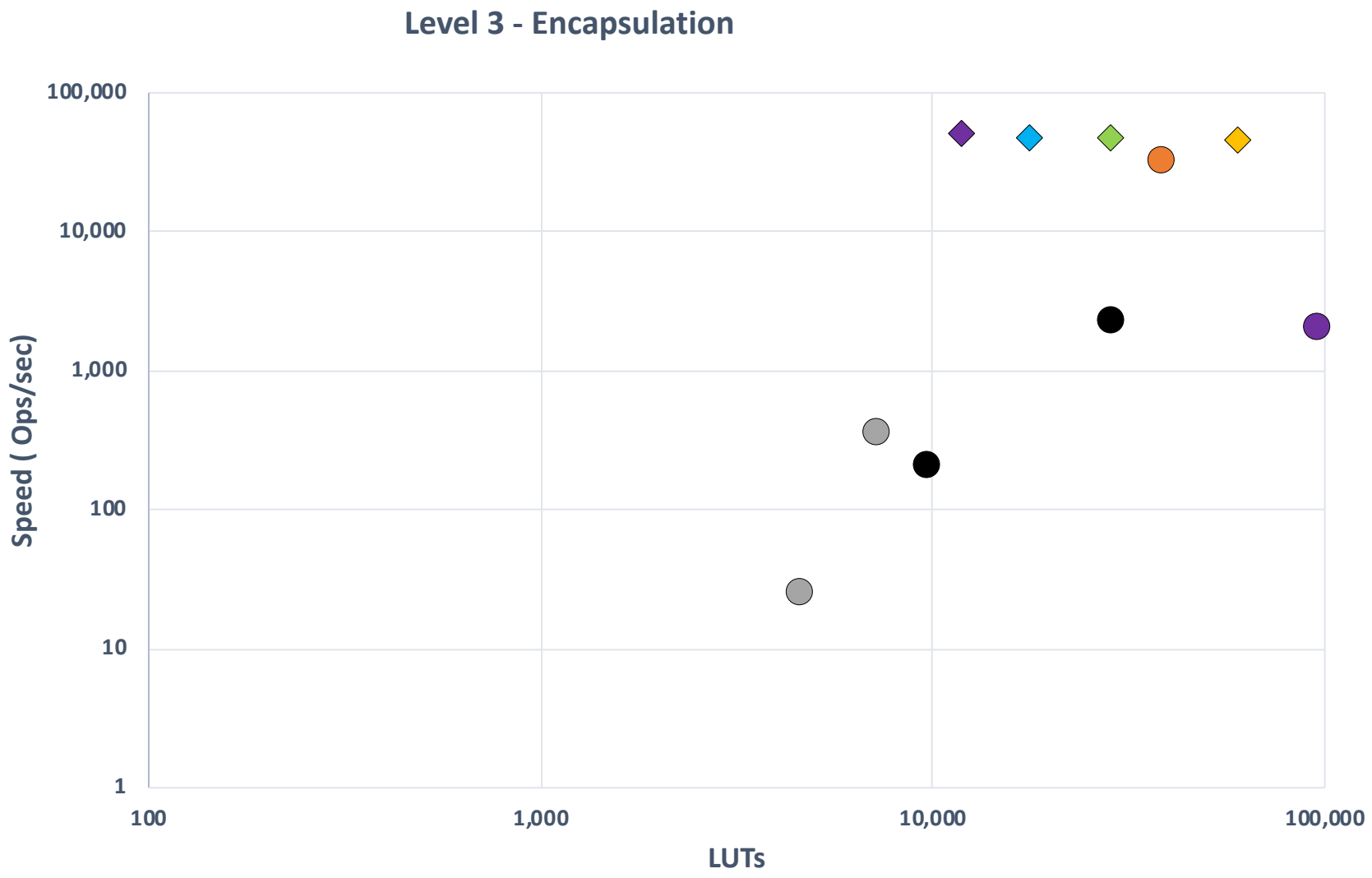
Level 1: Decapsulation on Artix-7



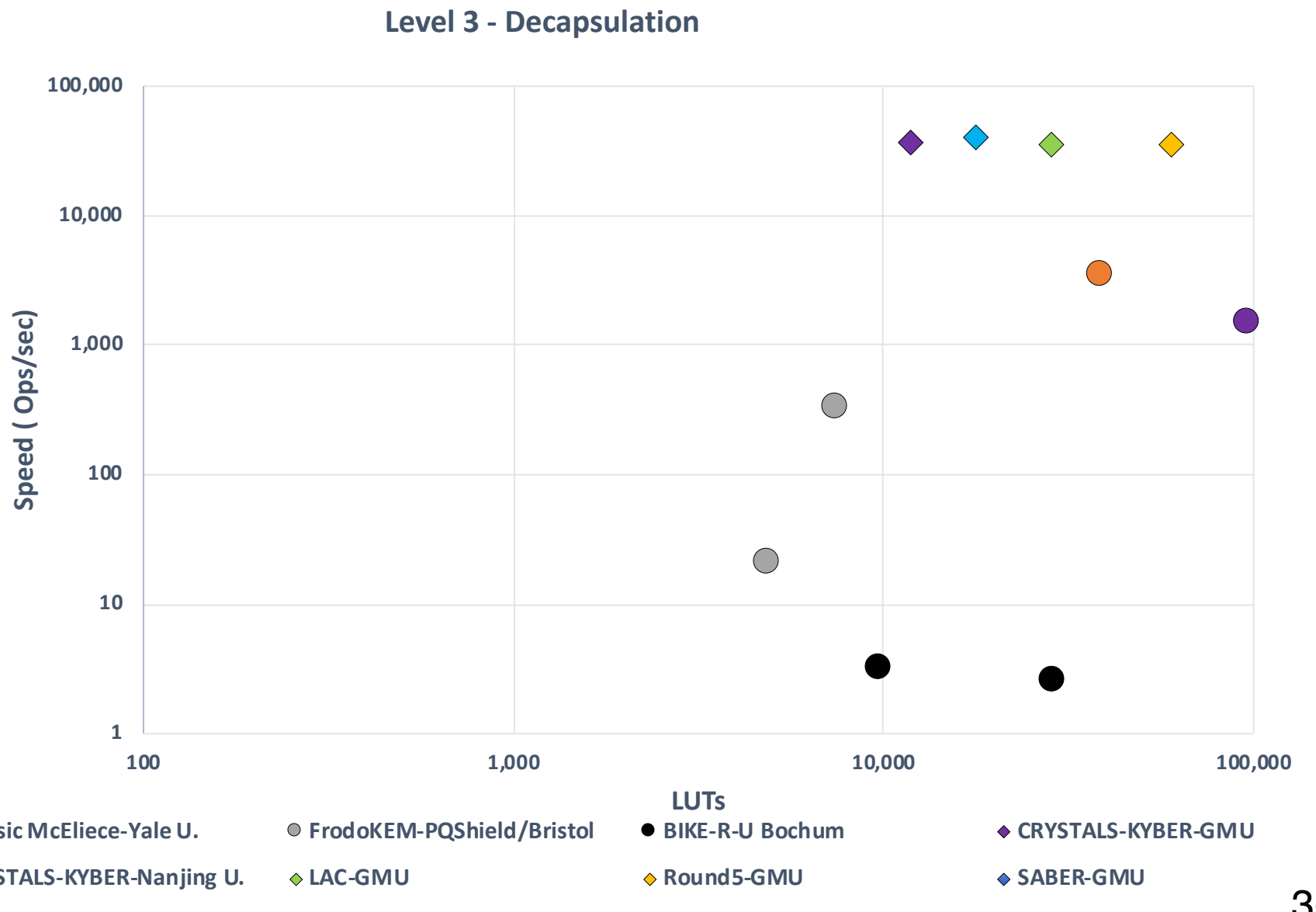
Level 3: Key Generation on Artix-7



Level 3: Encapsulation on Artix-7

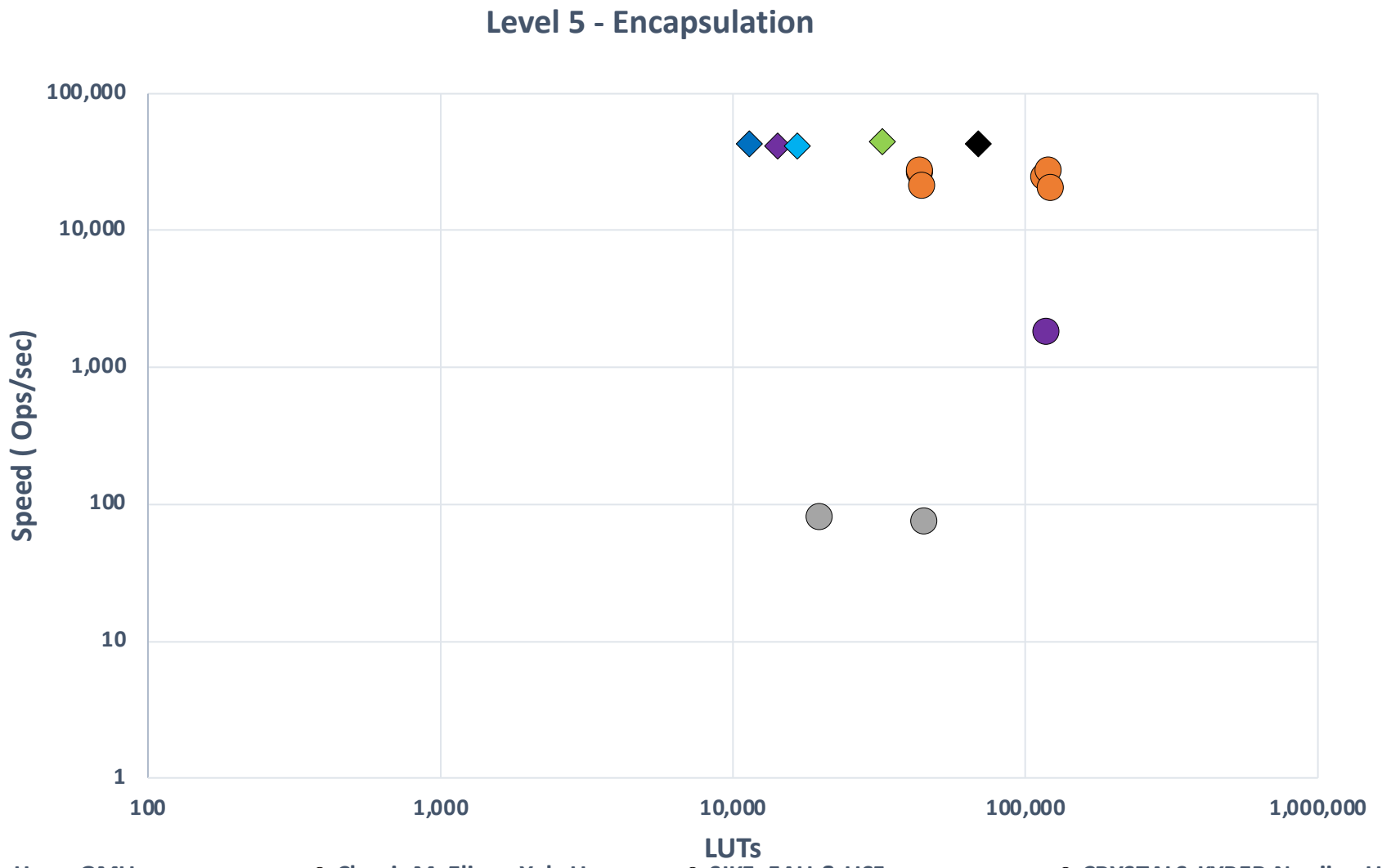


Level 3: Decapsulation on Artix-7

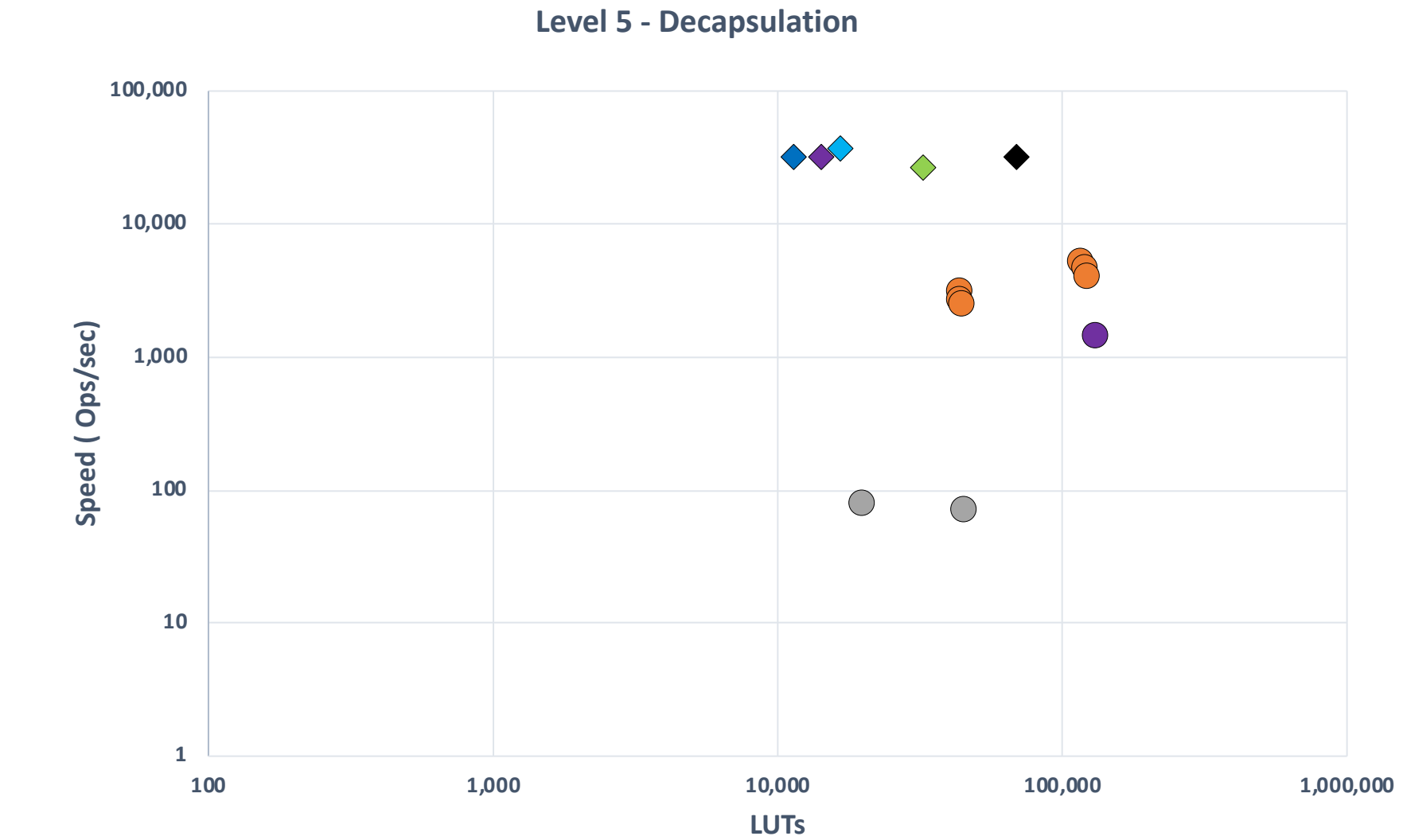


Level 5: Key Generation on Virtex-7

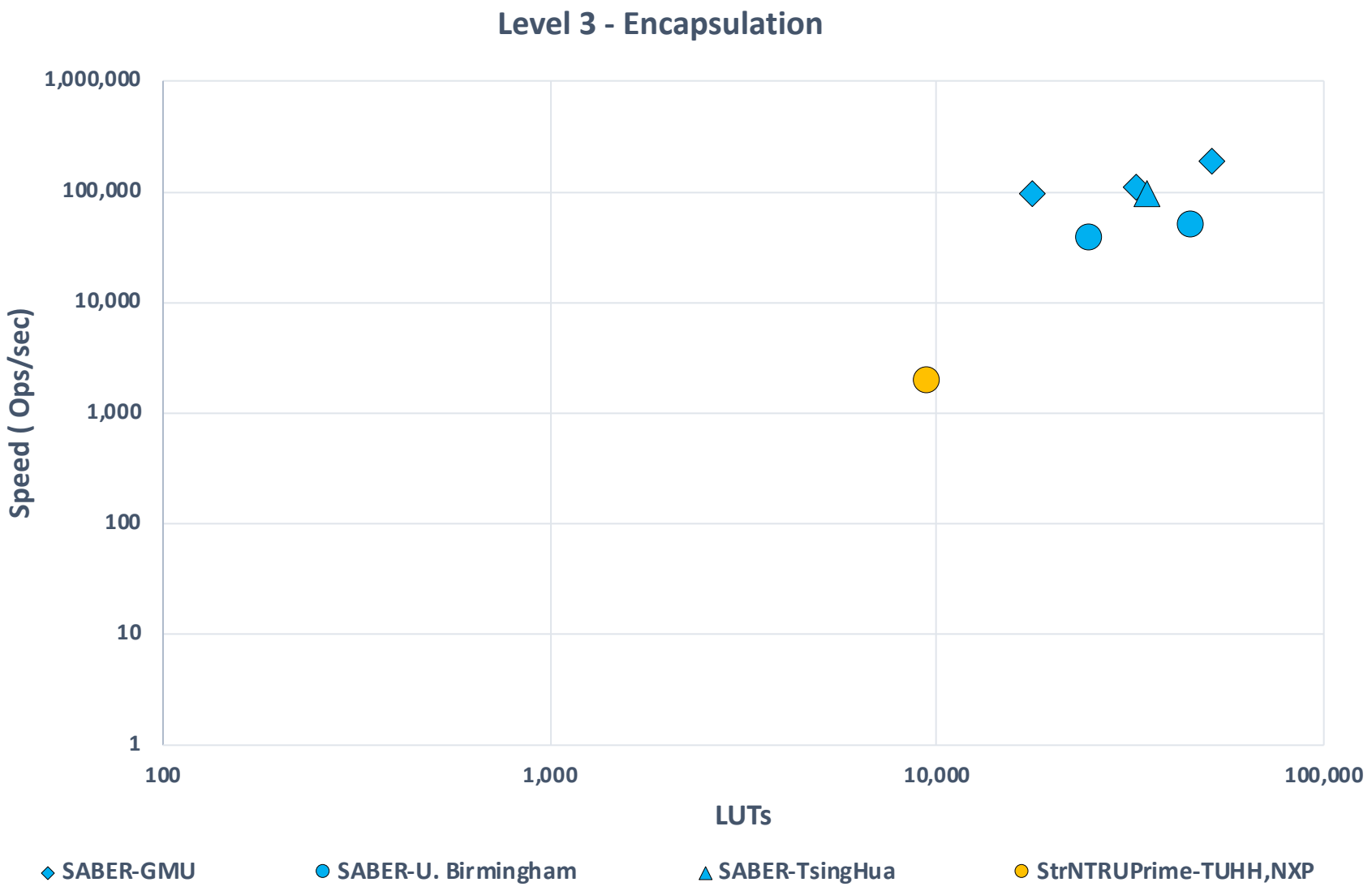
Level 5: Encapsulation on Virtex-7



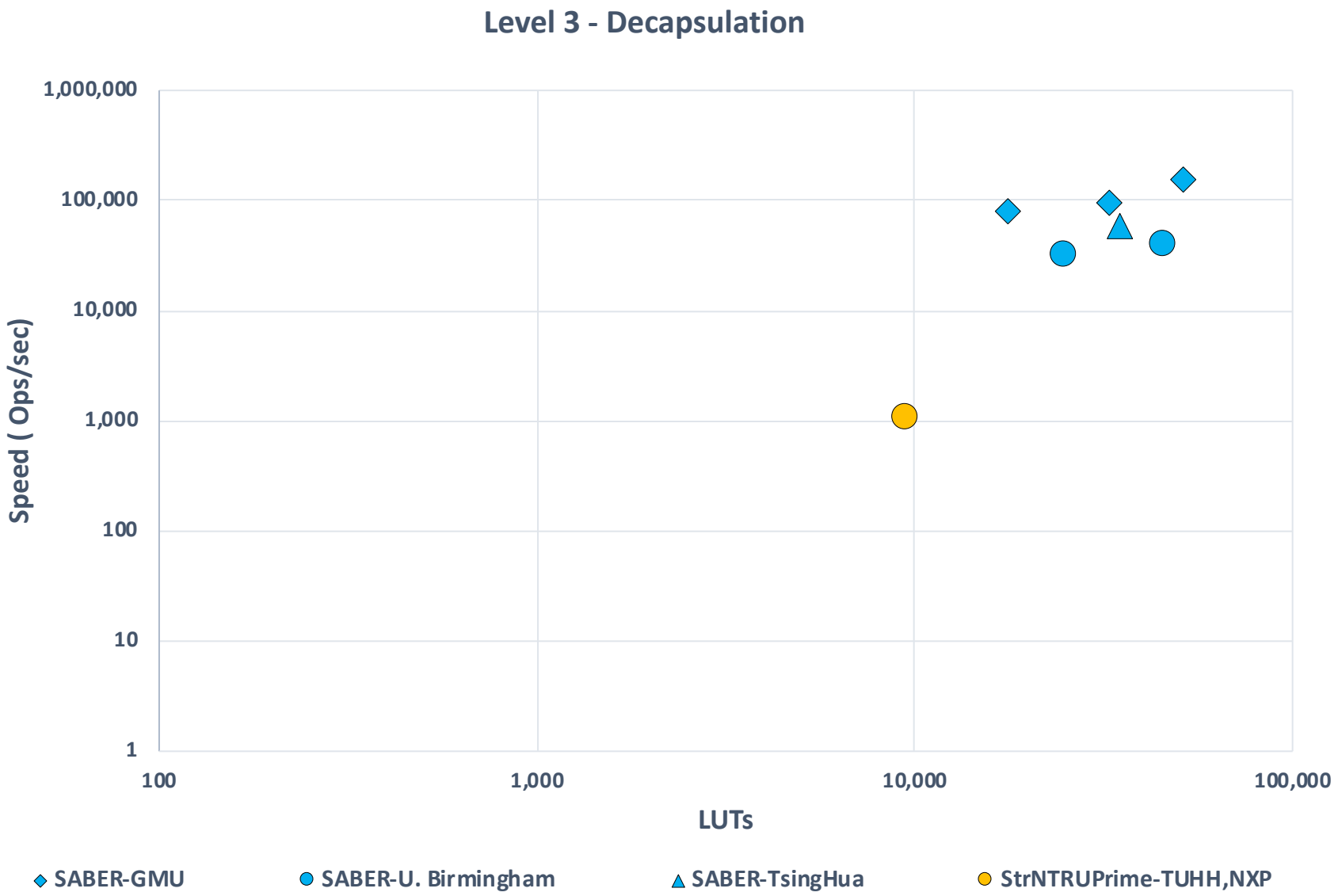
Level 5: Decapsulation on Virtex-7



Level 3: Encapsulation on Zynq UltraScale+



Level 3: Decapsulation on Zynq UltraScale+



Hardware Design Conclusions

Conclusions for Hardware Implementations

- CRYSTALS-KYBER, LAC, NewHope, Round5, and SABER (all lattice-based) comparable in terms of speed
- Among them, NewHope & CRYSTALS-KYBER the best in terms of resource utilization
- BIKE and HQC (code-based), FrodoKEM (lattice-based), SIKE (isogeny-based) about 2 orders of magnitude slower for all operations
- Classic McEliece (code-based) comparable in terms of encapsulation, about 1 order of magnitude slower for decapsulation, about 2-3 orders of magnitude slower for key generation

GMU Hardware Designs

HW Design: Case Study

7 Key Encapsulation Mechanisms (KEMs)

representing

5 out of 9 Round 2 Lattice-Based KEMs

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

NewHope, LAC, and Round5

Feature	NewHope	LAC (v3a/v3b)	Round5 (0d/5d)
Underlying Problem	Ring-LWE	Ring-LWE	Ring-LWR
Error Correcting Code	None	BCH	None / XEf
Security Levels	lattice dimension = n L1: n=512, L5: n=1024	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 12,289	Prime 251 / 256	L1: $2^{13}/2^{10}$, L3: $2^{12}/2^{12}$ L5: $2^{13}/2^{11}$
Required Hash-based Functions	SHAKE128, SHAKE256	Left up to implementers	L1: SHAKE128, L3, L5: SHAKE256
Sampling	CBD*	n-ary CBD with fixed Hamming weight	CBD*
# Poly-Mult in Encaps	2	2	2
# Poly-Mult in Decaps	3	3	3

* Centered Binomial Distribution (CBD)

CRYSTALS-KYBER and SABER

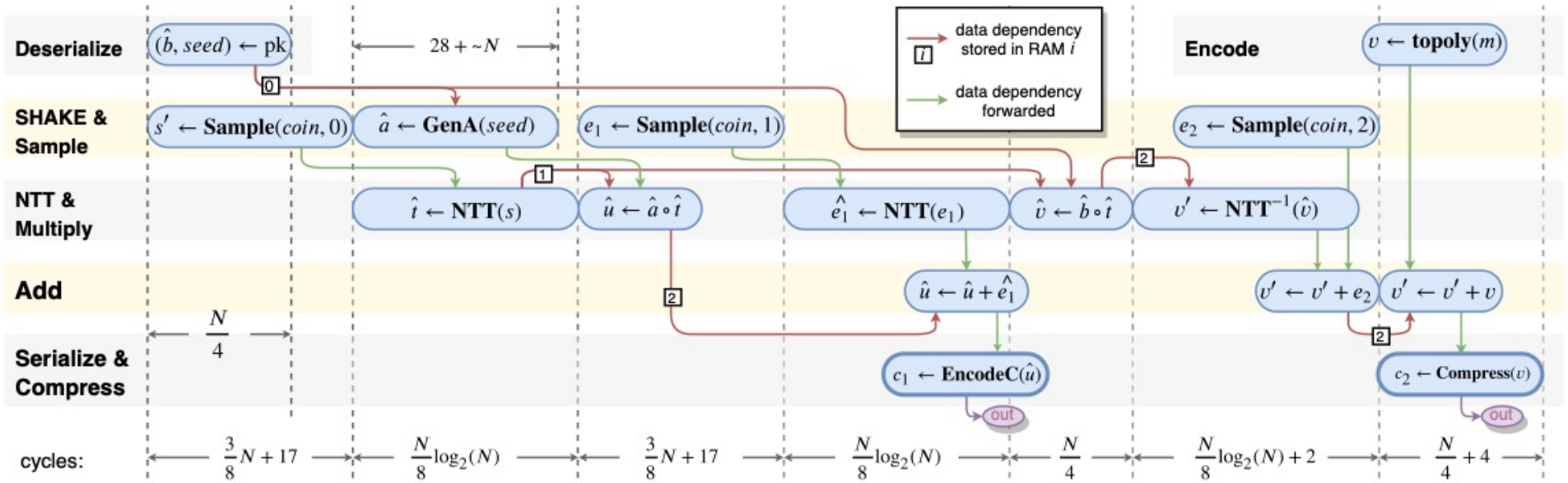
Feature	CRYSTALS-KYBER	SABER
Underlying Problem	Module-LWE	Module-LWR
Security Levels	n=256, lattice dimension = k*n L1: k=2, L3: k=3, L5: k=4	n=256, lattice dimension = l*n L1: l=2, L3: l=3, L5: l=4
Modulus *q*	Prime 3,329	2^{13}
Required Hash-based Functions	SHAKE128, SHAKE256 SHA3-256, SHA3-512	SHAKE128, SHA3-256, SHA3-512
Sampling	CBD*	CBD*
# Poly-Mult in Encaps	$k^2 + k$	$l^2 + l$
# Poly-Mult in Decaps	$k^2 + 2k$	$l^2 + l$

* Centered Binomial Distribution (CBD)

Common Optimization Method

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

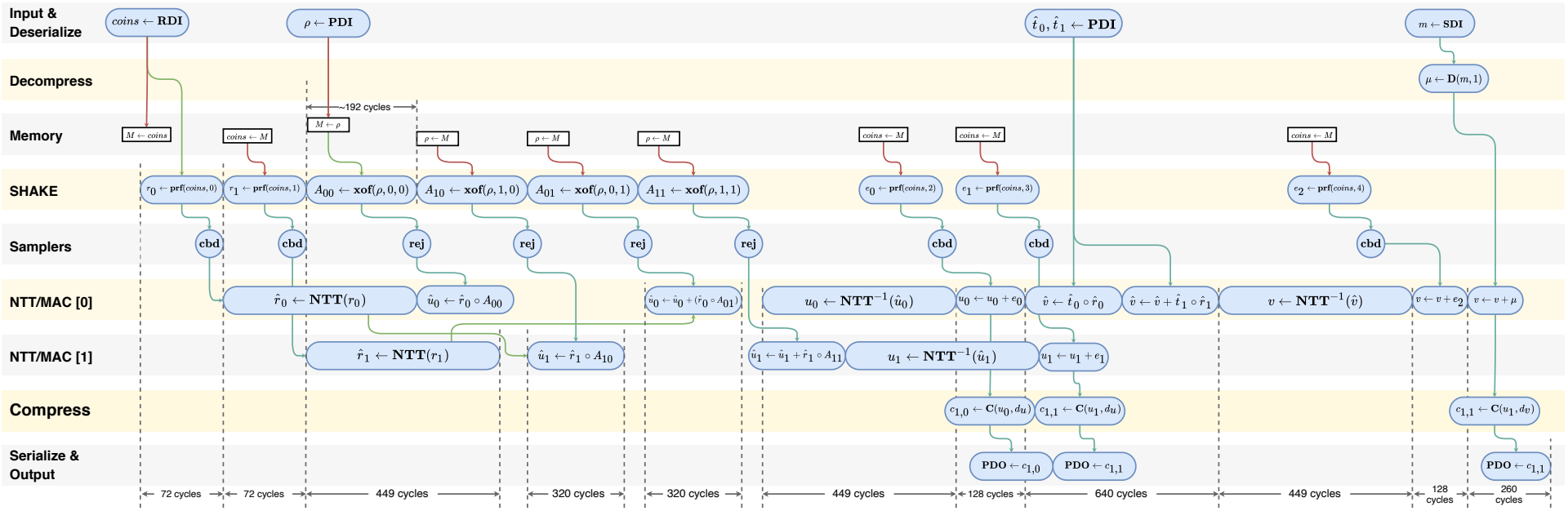
NewHope Encryption



Common Optimization Method

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

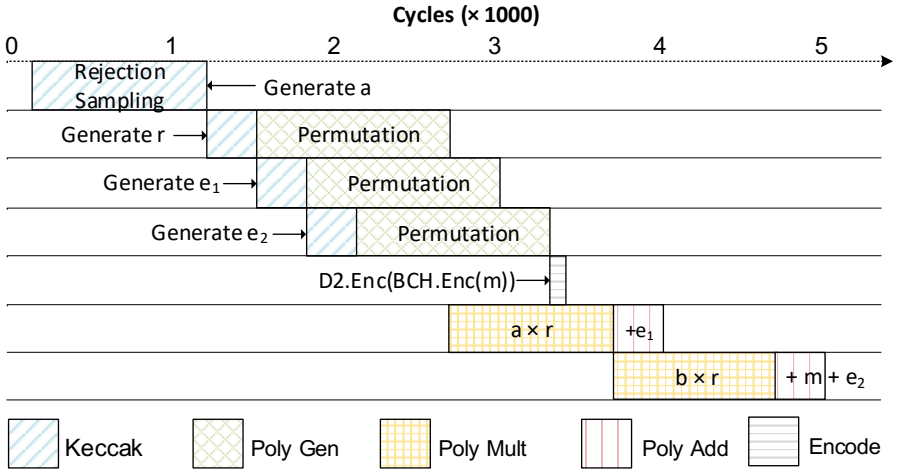
CRYSTALS-KYBER Encryption (Security Level 1)



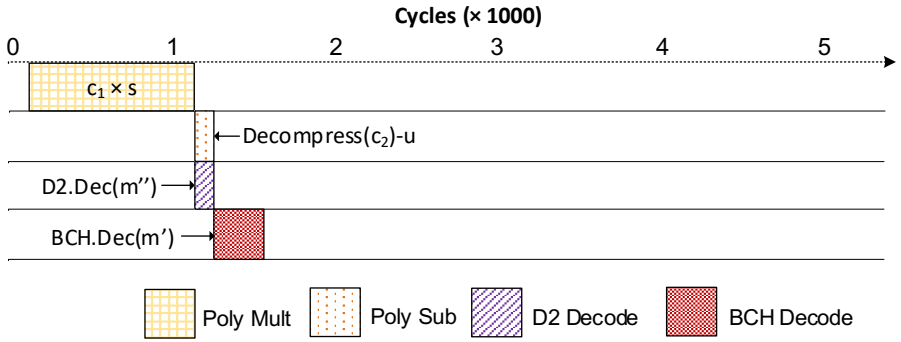
Common Optimization Method

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

LAC Encryption



LAC Decryption

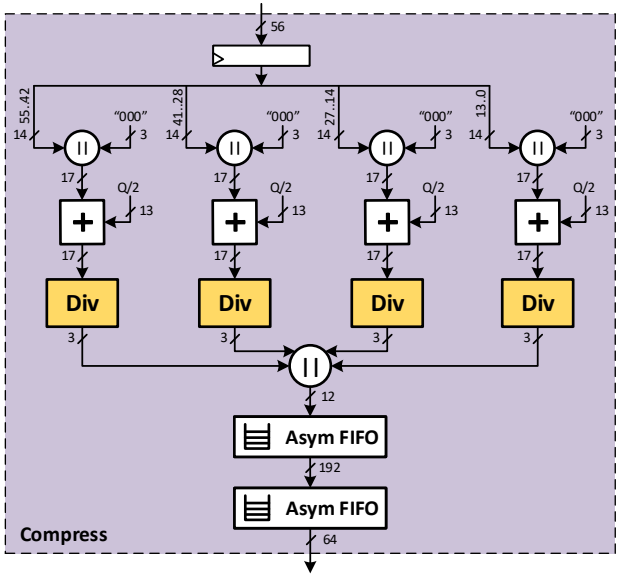


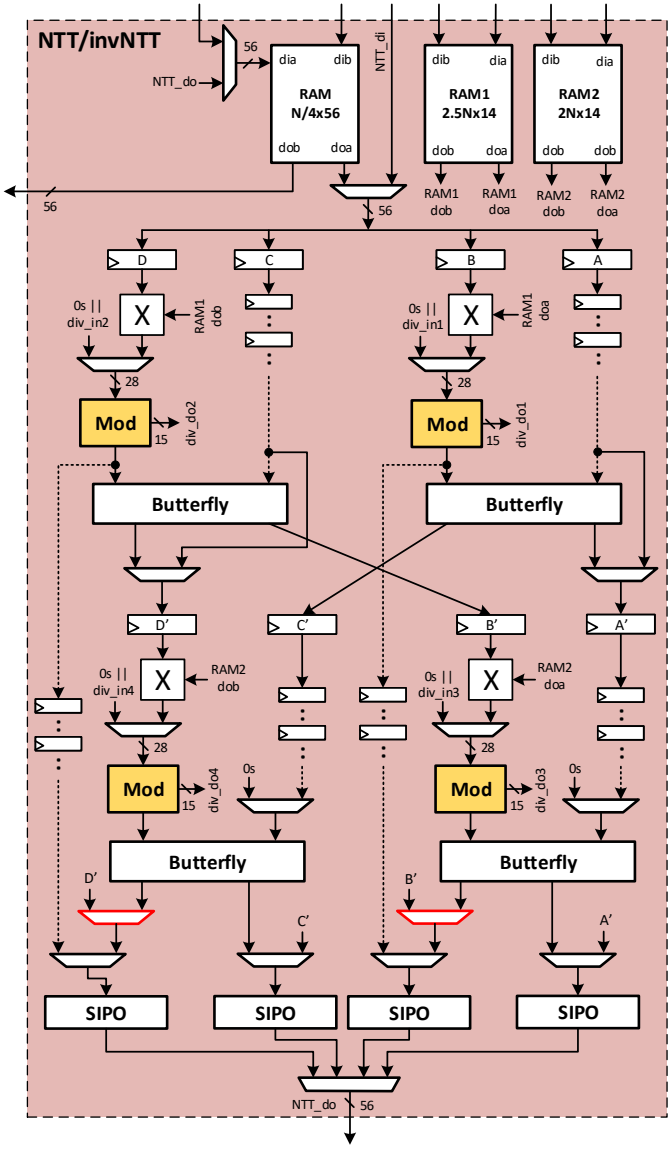
Algorithm-Specific Optimization Methods

NewHope & CRYSTALS-KYBER

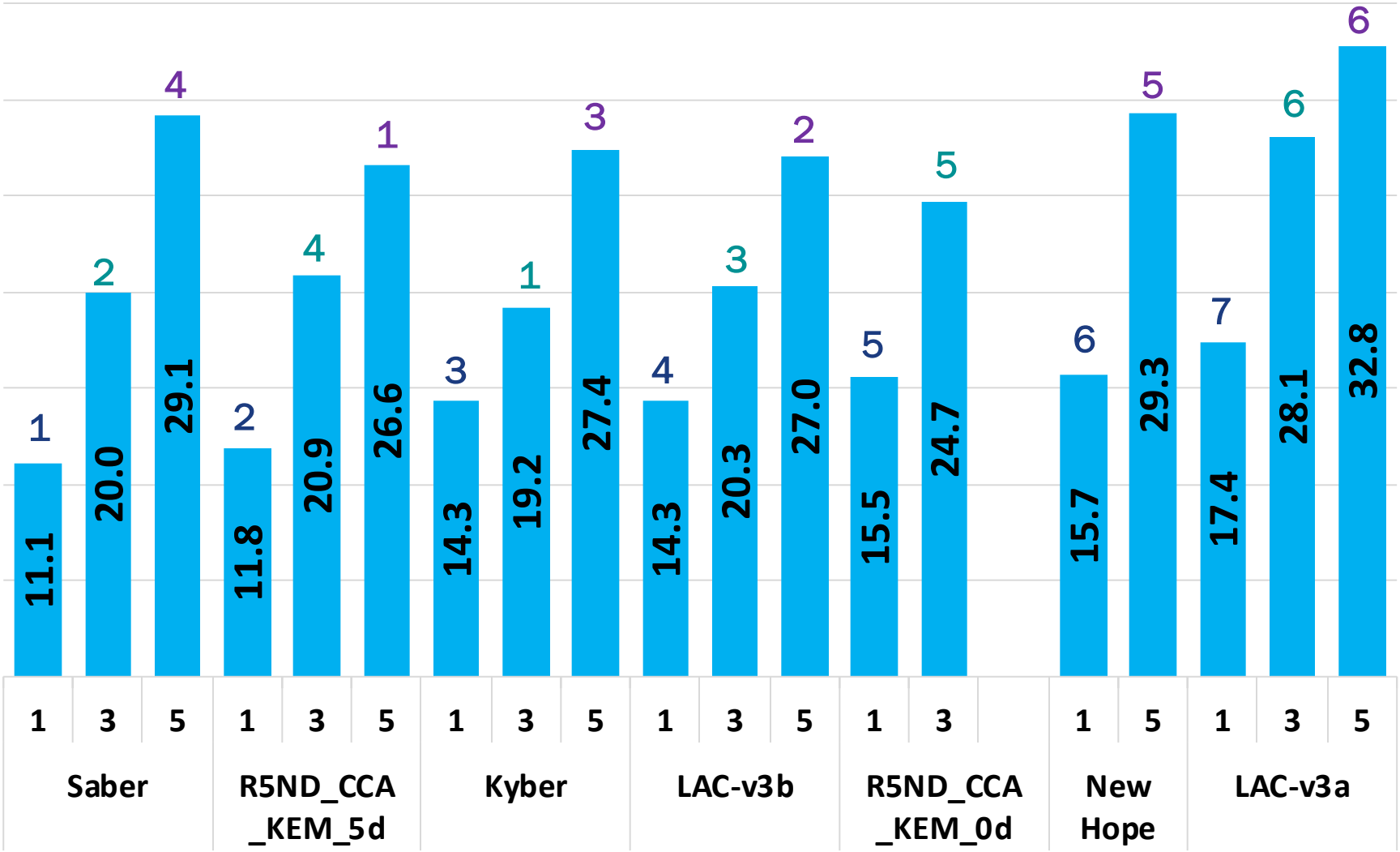
- Number Theoretic Transform (NTT)
- Processing FOUR coefficients at a time
- Resource sharing

  e.g., use a single module to perform NTT, NTT^{-1}, & pointwise multiplication
- Efficient modular reduction

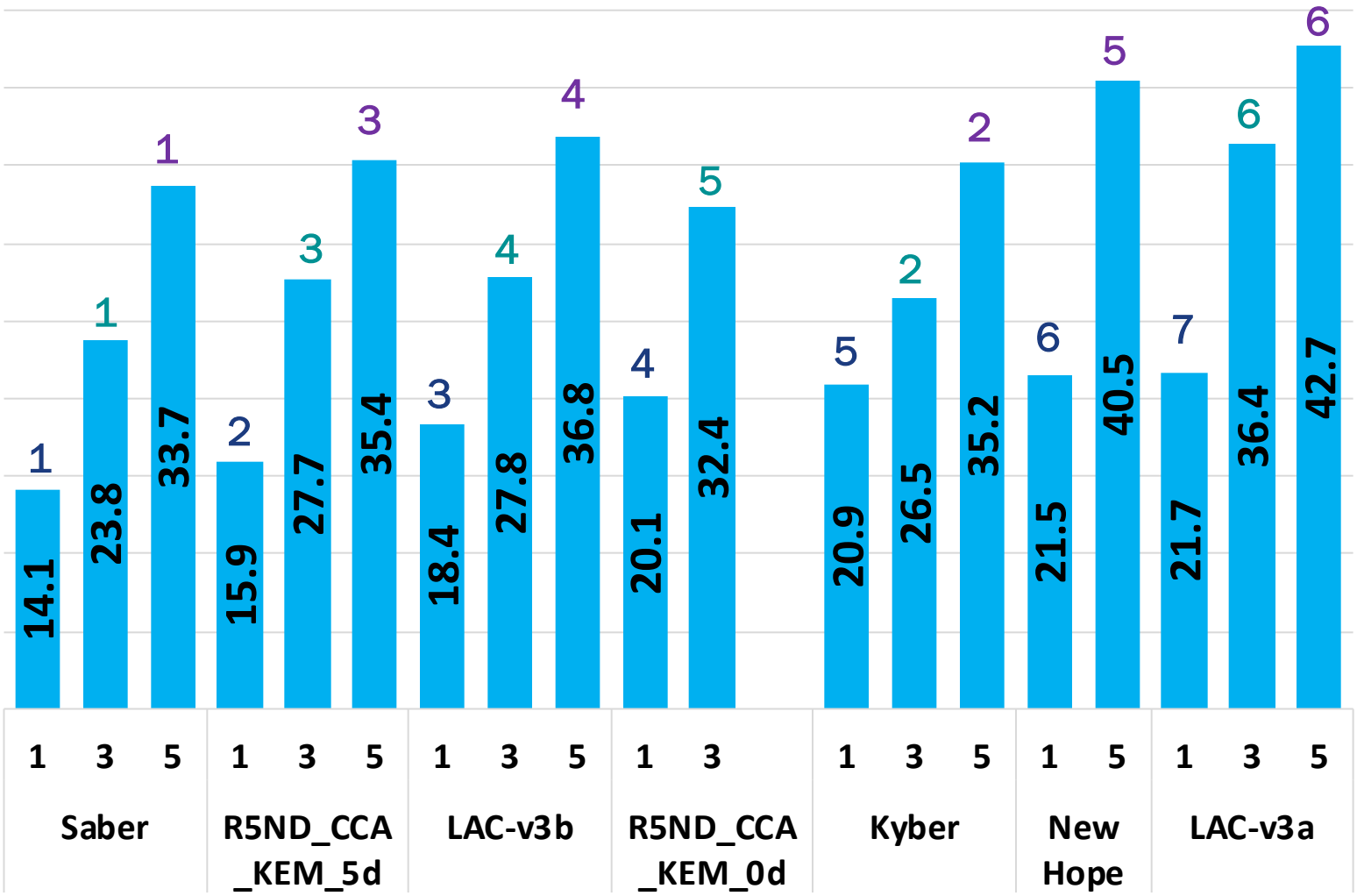




Encapsulation Time on Artix-7 [µs]



Decapsulation Time on Artix-7 [µs]



Rankings & Ratios on Artix-7

Encapsulation

Level 1			Level 3			Level 5		
	Exe[us]	Ratio		Exe[us]	Ratio		Exe[us]	Ratio
LightSaber	11.6	1.00	Kyber	19.9	1.00	Round5_5d	27.6	1.00
Round5_5d	12.2	1.05	Saber	20.8	1.05	LAC-v3b	28.1	1.02
Kyber	14.8	1.28	LAC-v3b	21.2	1.07	Kyber	28.4	1.03
LAC-v3b	14.8	1.28	Round5_5d	21.6	1.09	FireSaber	30.1	1.09
Round5_0d	16.0	1.38	Round5_0d	25.6	1.29	NewHope	30.3	1.10
NewHope	16.3	1.41	LAC-v3a	29.1	1.46	LAC-v3a	33.9	1.23
LAC-v3a	17.9	1.54						

Decapsulation

Level 1			Level 3			Level 5		
	Exe[us]	Ratios		Exe[us]	Ratio		Exe[us]	Ratio
LightSaber	14.6	1.00	Saber	24.5	1.00	FireSaber	34.6	1.00
Round5_5d	16.3	1.12	Kyber	27.2	1.11	Kyber	36.2	1.05
LAC-v3b	18.9	1.29	Round5_5d	28.4	1.16	Round5_5d	36.4	1.05
Round5_0d	20.6	1.41	LAC-v3b	28.7	1.17	LAC-v3b	37.9	1.10
Kyber	21.4	1.47	Round5_0d	33.2	1.36	NewHope	41.5	1.20
NewHope	22.0	1.51	LAC-v3a	37.4	1.53	LAC-v3a	43.8	1.27
LAC-v3a	22.2	1.52						

Rankings & Ratios on Artix-7

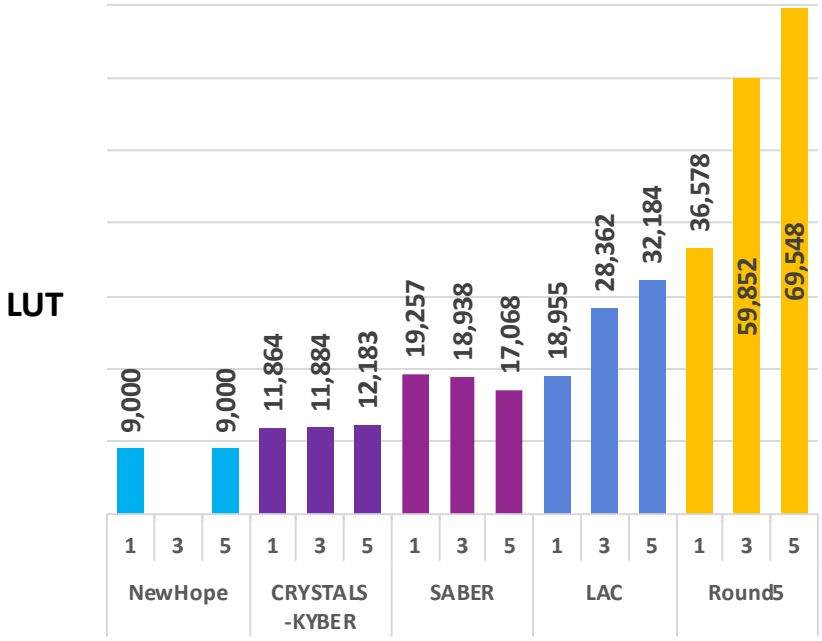
Encapsulation

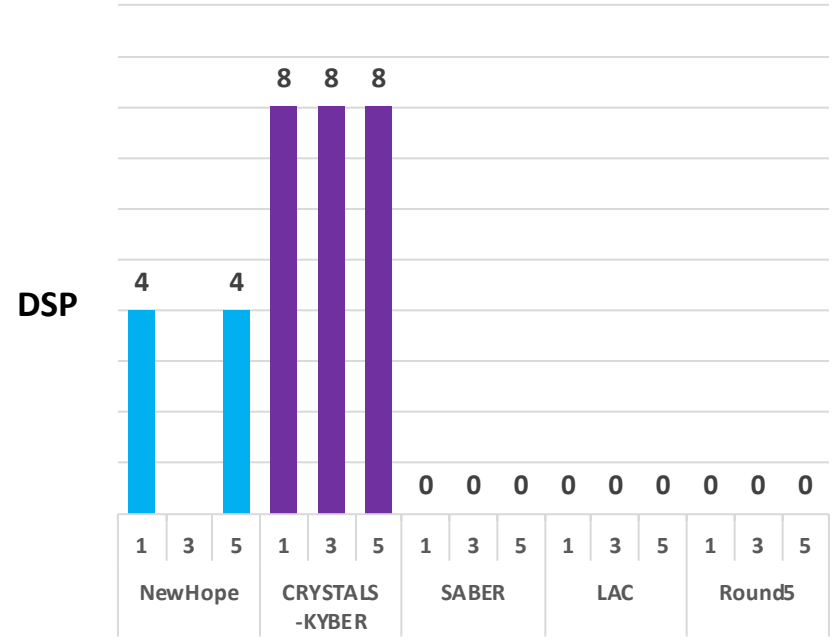
Level 1			Level 3			Level 5		
	Exe[us]	Ratio		Exe[us]	Ratio		Exe[us]	Ratio
LightSaber	11.6	1.00	Kyber	19.9	1.00	Round5_5d	27.6	1.00
Round5_5d	12.2	1.05	Saber	20.8	1.05	LAC-v3b	28.1	1.02
Kyber	14.8	1.28	LAC-v3b	21.2	1.07	Kyber	28.4	1.03
LAC-v3b	14.8	1.28	Round5_5d	21.6	1.09	FireSaber	30.1	1.09
Round5_0d	16.0	1.38	Round5_0d	25.6	1.29	NewHope	30.3	1.10
NewHope	16.3	1.41	LAC-v3a	29.1	1.46	LAC-v3a	33.9	1.23
LAC-v3a	17.9	1.54						

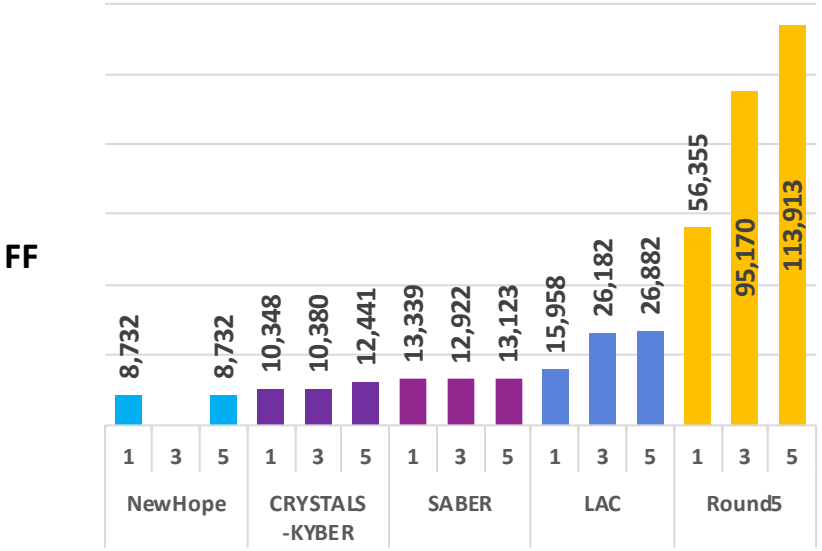
Decapsulation

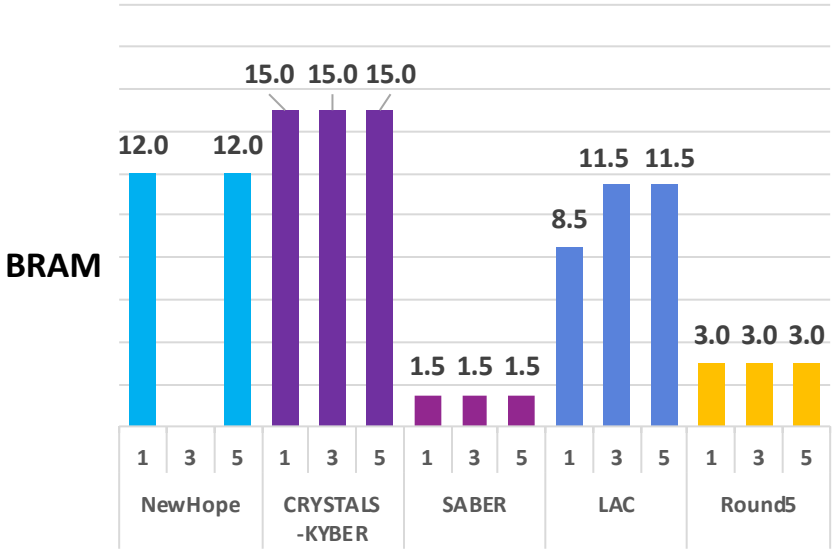
Level 1			Level 3			Level 5		
	Exe[us]	Ratios		Exe[us]	Ratio		Exe[us]	Ratio
LightSaber	14.6	1.00	Saber	24.5	1.00	FireSaber	34.6	1.00
Round5_5d	16.3	1.12	Kyber	27.2	1.11	Kyber	36.2	1.05
LAC-v3b	18.9	1.29	Round5_5d	28.4	1.16	Round5_5d	36.4	1.05
Round5_0d	20.6	1.41	LAC-v3b	28.7	1.17	LAC-v3b	37.9	1.10
Kyber	21.4	1.47	Round5_0d	33.2	1.36	NewHope	41.5	1.20
NewHope	22.0	1.51	LAC-v3a	37.4	1.53	LAC-v3a	43.8	1.27
LAC-v3a	22.2	1.52						

Resource Utilization on Artix-7



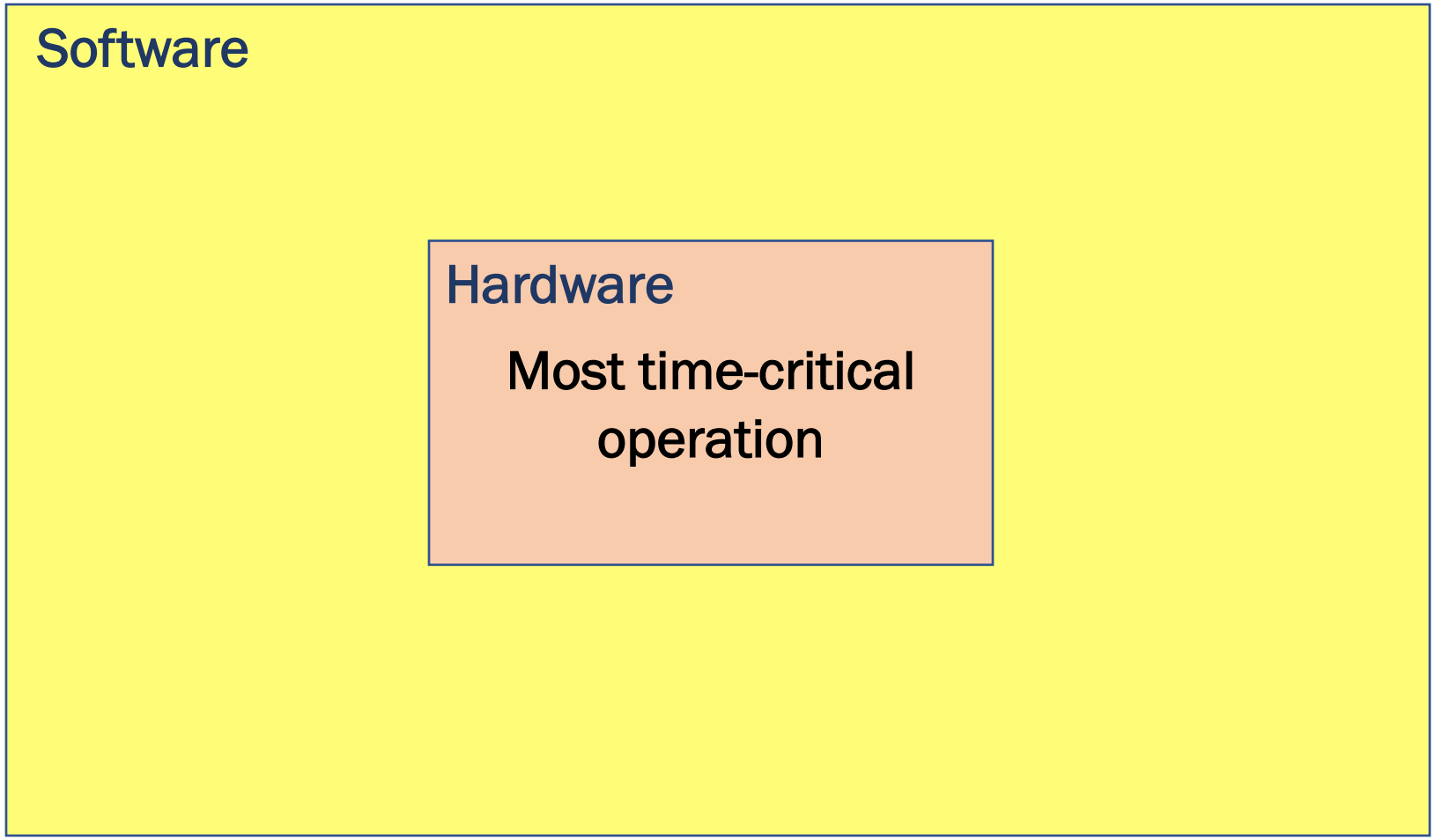






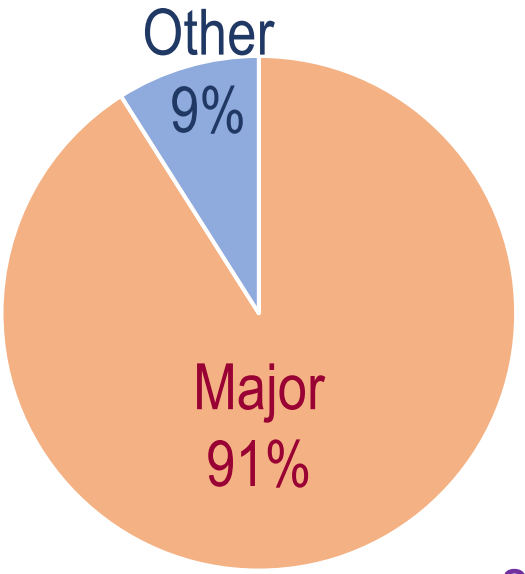

GMU
Software/Hardware
Co-designs

Software/Hardware Codesign

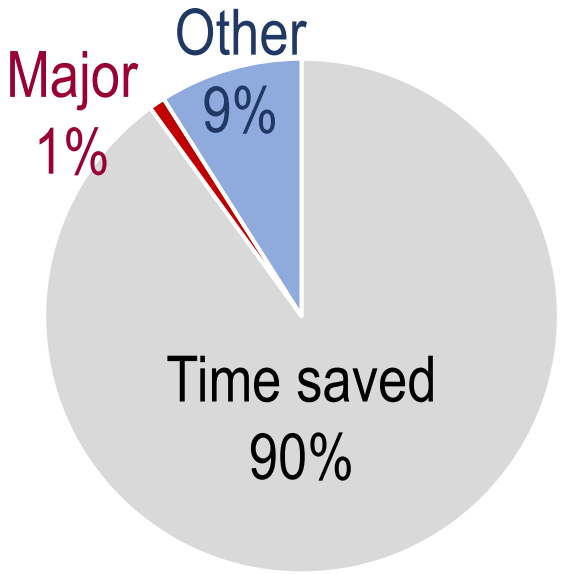


SW/HW Co-design: Motivational Example 1

Software



Software/Hardware



speed-up ≥ 100

91% major operation(s)
9% other operations

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

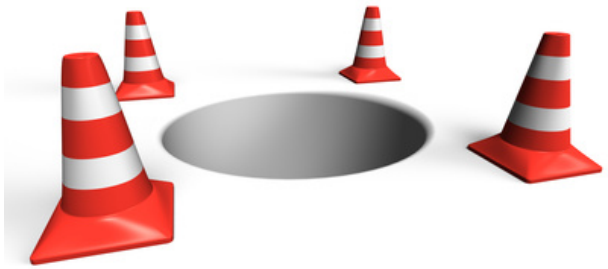
Total Speed-Up ≥ 10

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

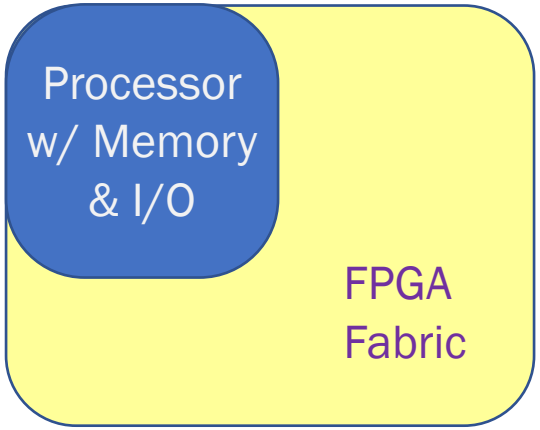


- Performance & ranking may strongly depend on the selected hardware/software partitioning

First step, not the ultimate solution!

Two Major Types of Platforms for SW/HW Co-design

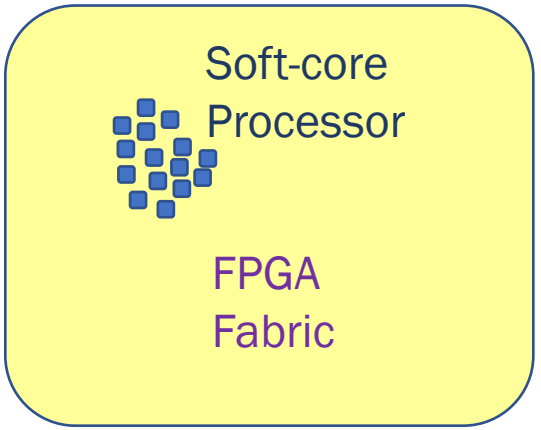
System on Chip (SoC) FPGA



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

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 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:

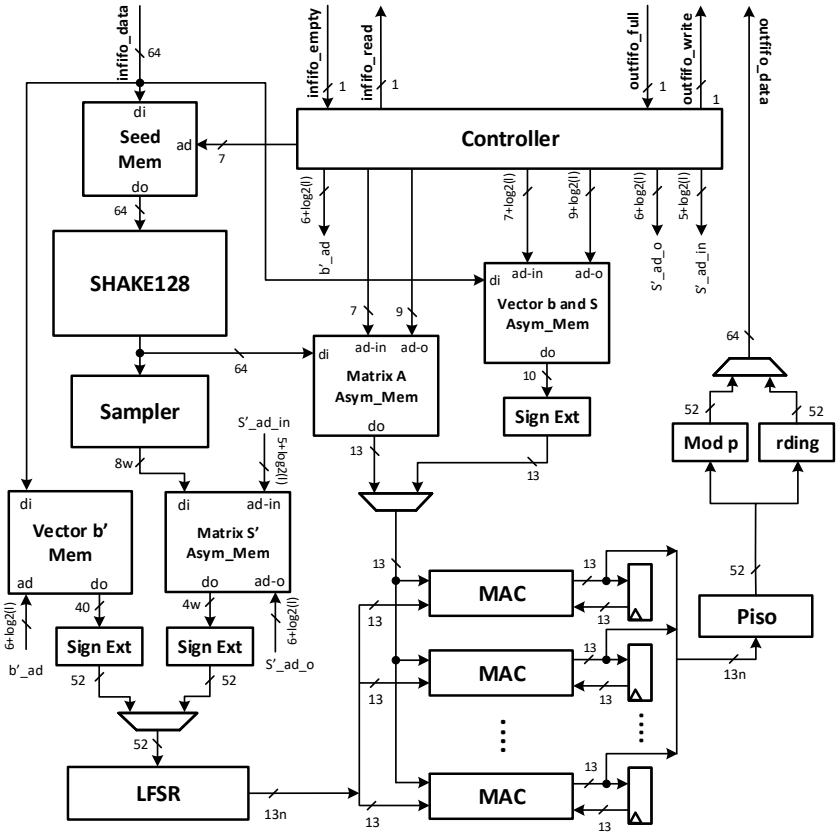
- Small size of inputs and outputs
- Potential for combining with neighboring functions

From knowledge of operations and concurrent computing:

- High potential for parallelization

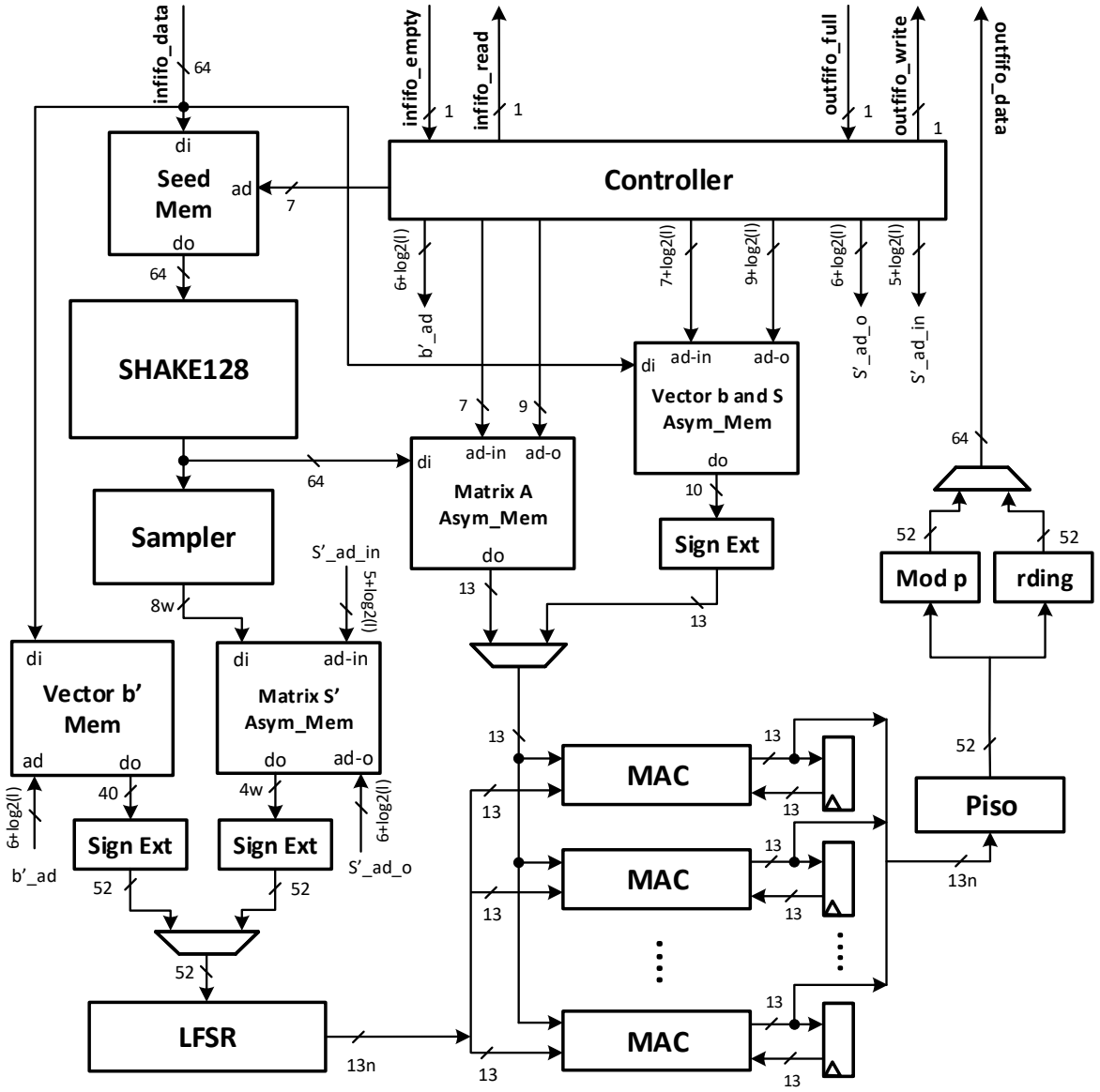
Operations Offloaded to Hardware

- 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



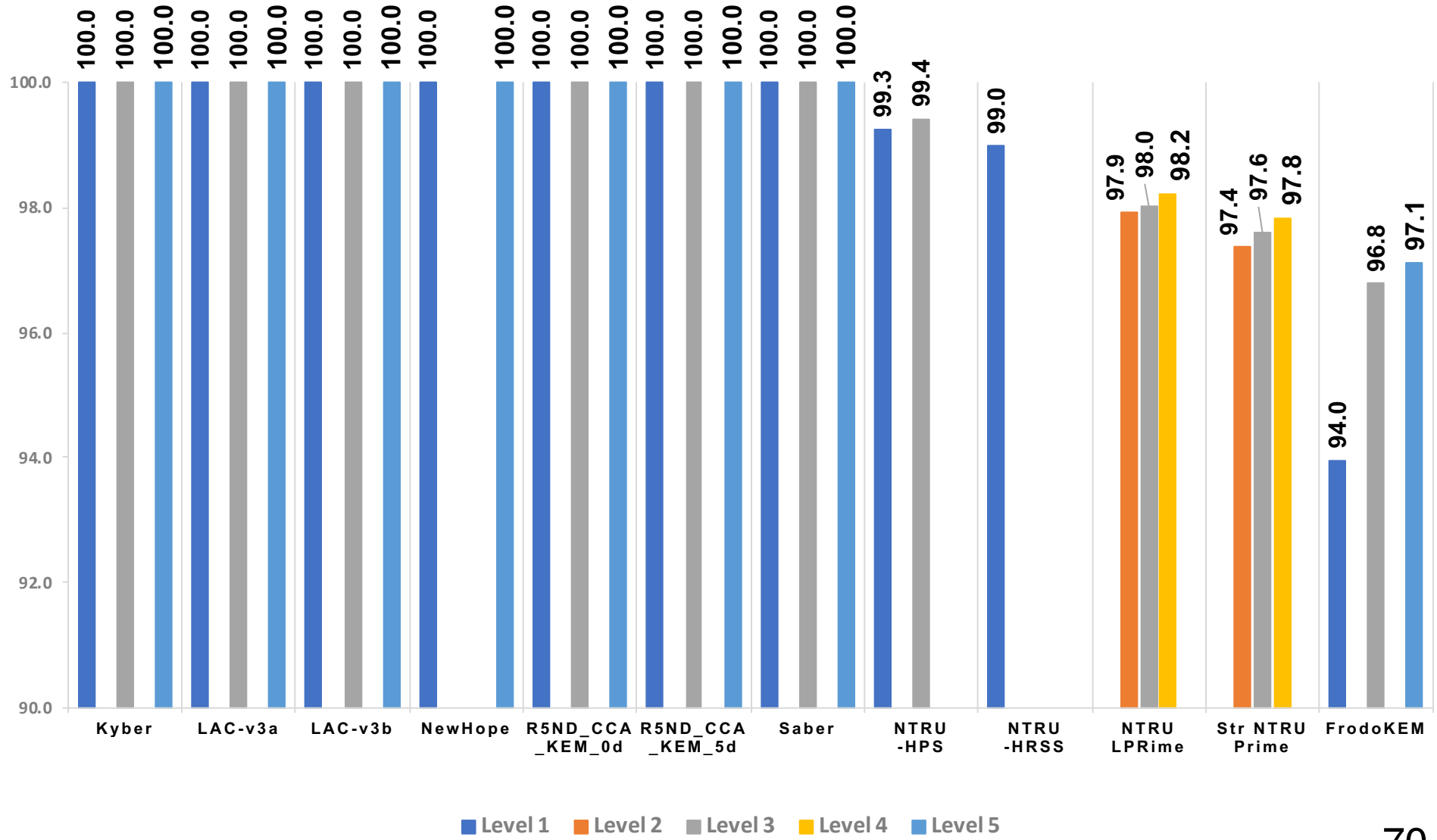
Hardware accelerator of Saber

Detailed hierarchical block diagrams developed for the entire hardware accelerator

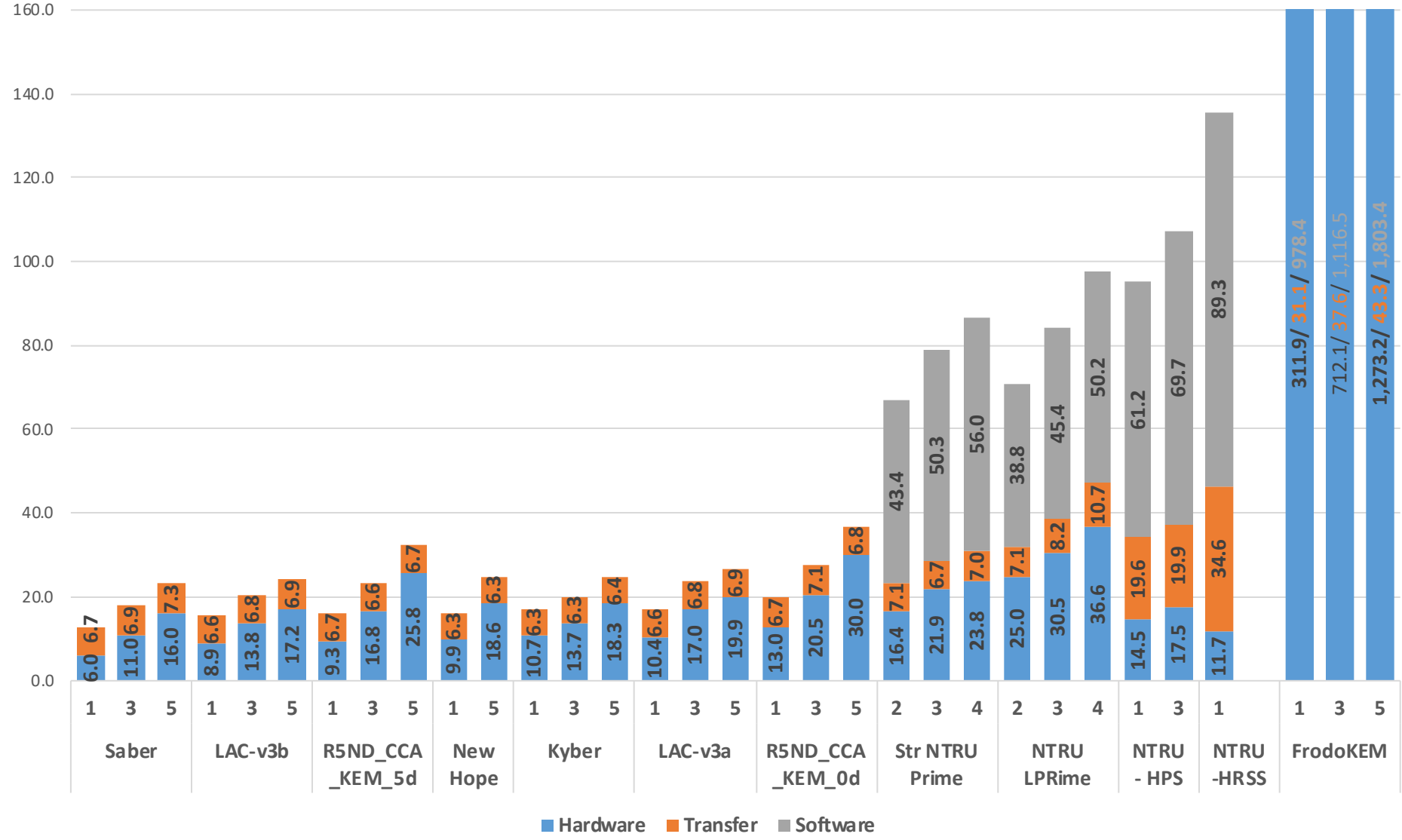


Results

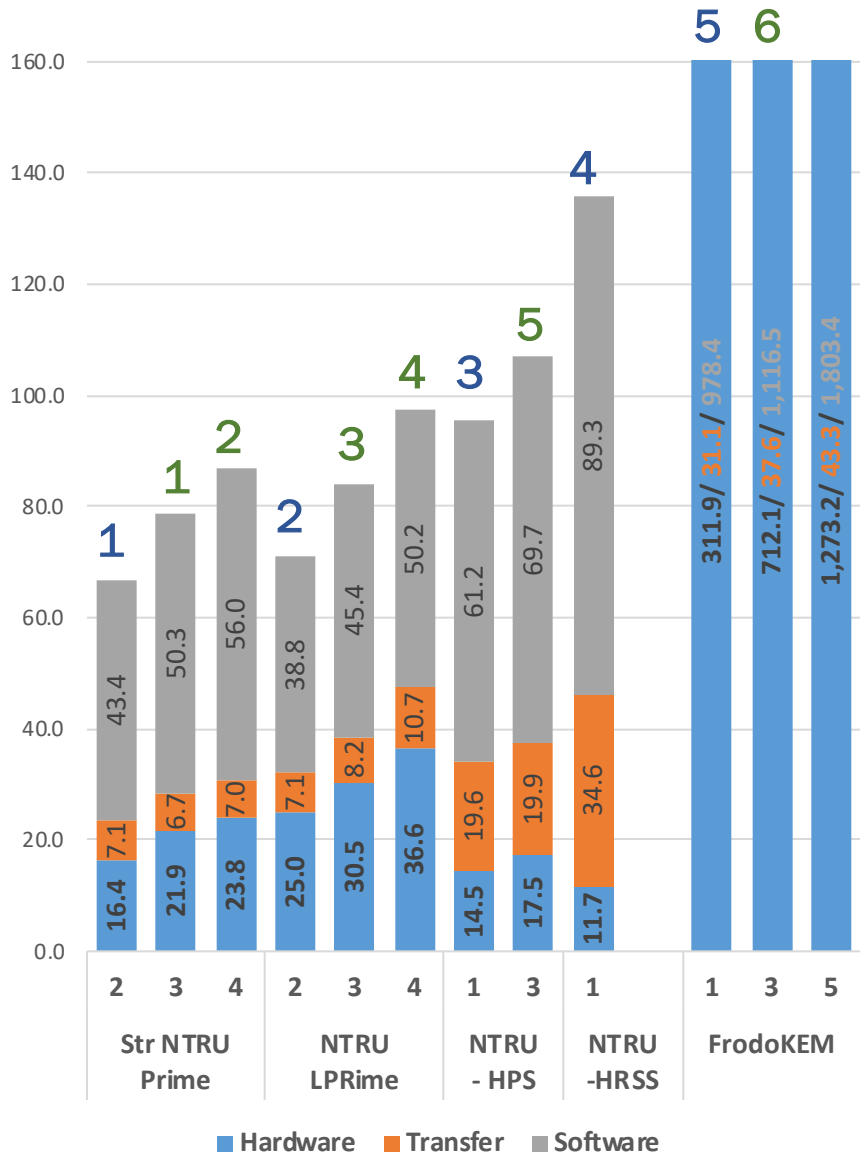
SW Part Sped up by HW[%]: Decapsulation



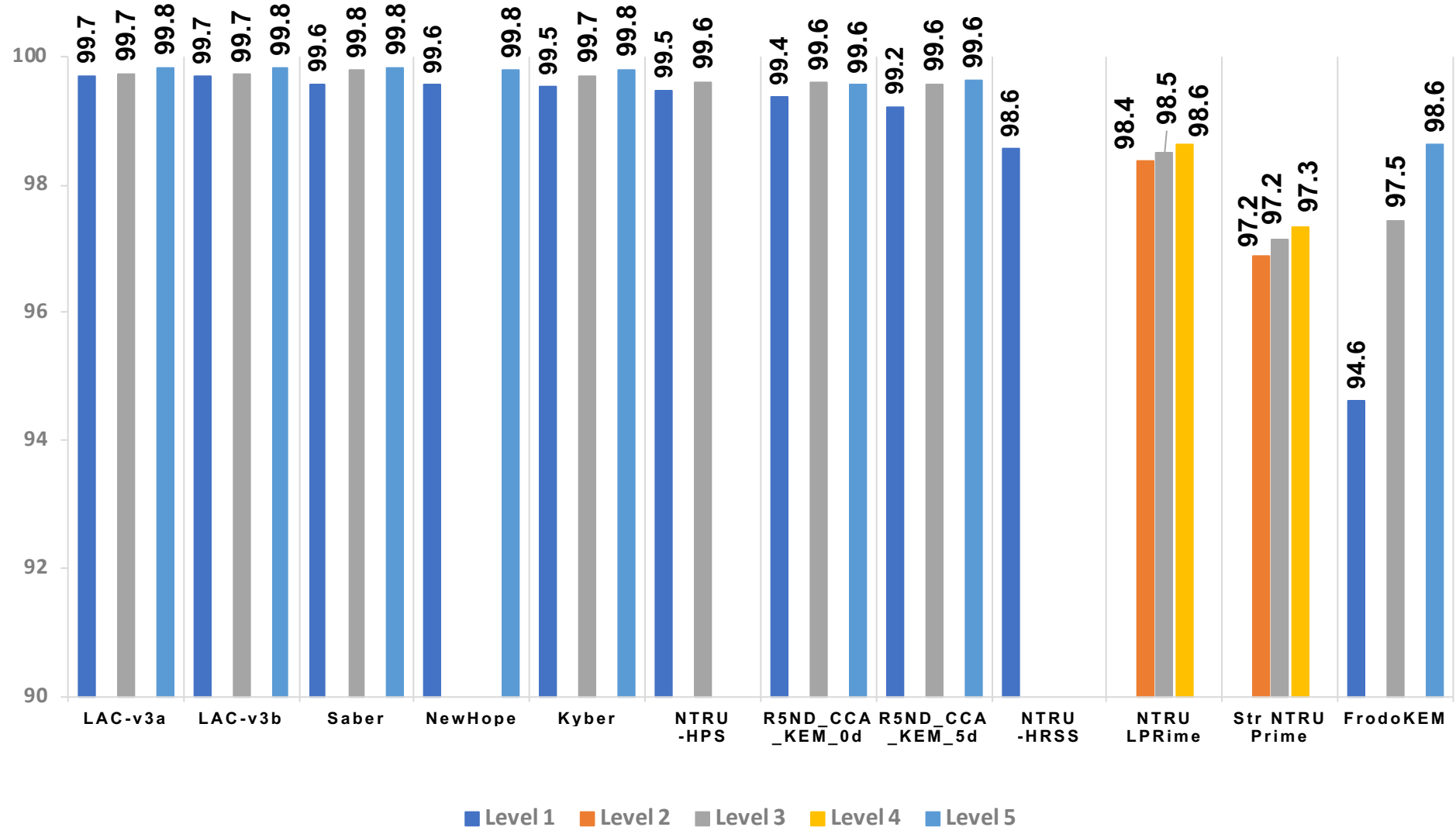
Round2 KEMs: SW/HW Results for Decaps



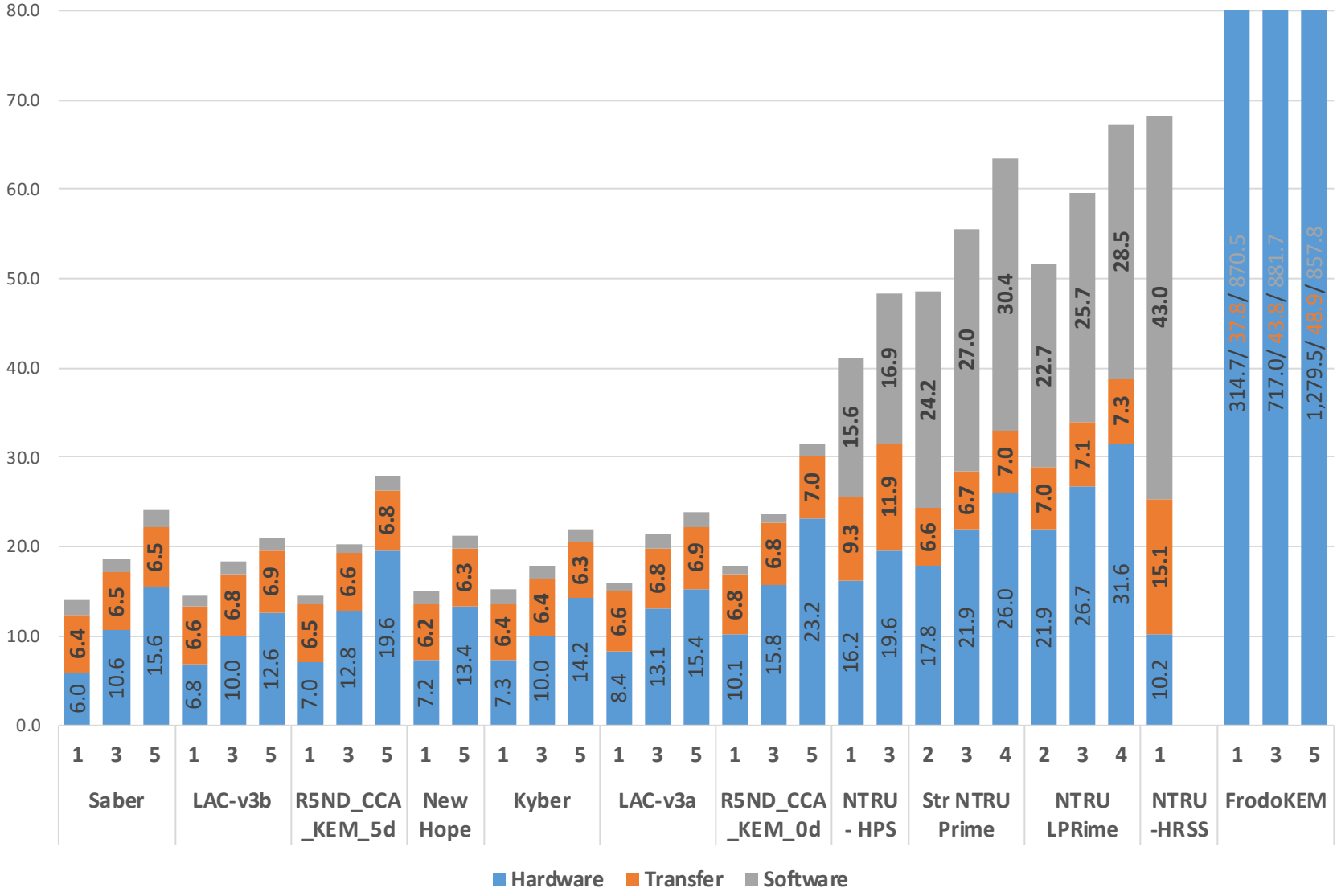
Round2 KEMs: SW/HW Results for Decaps



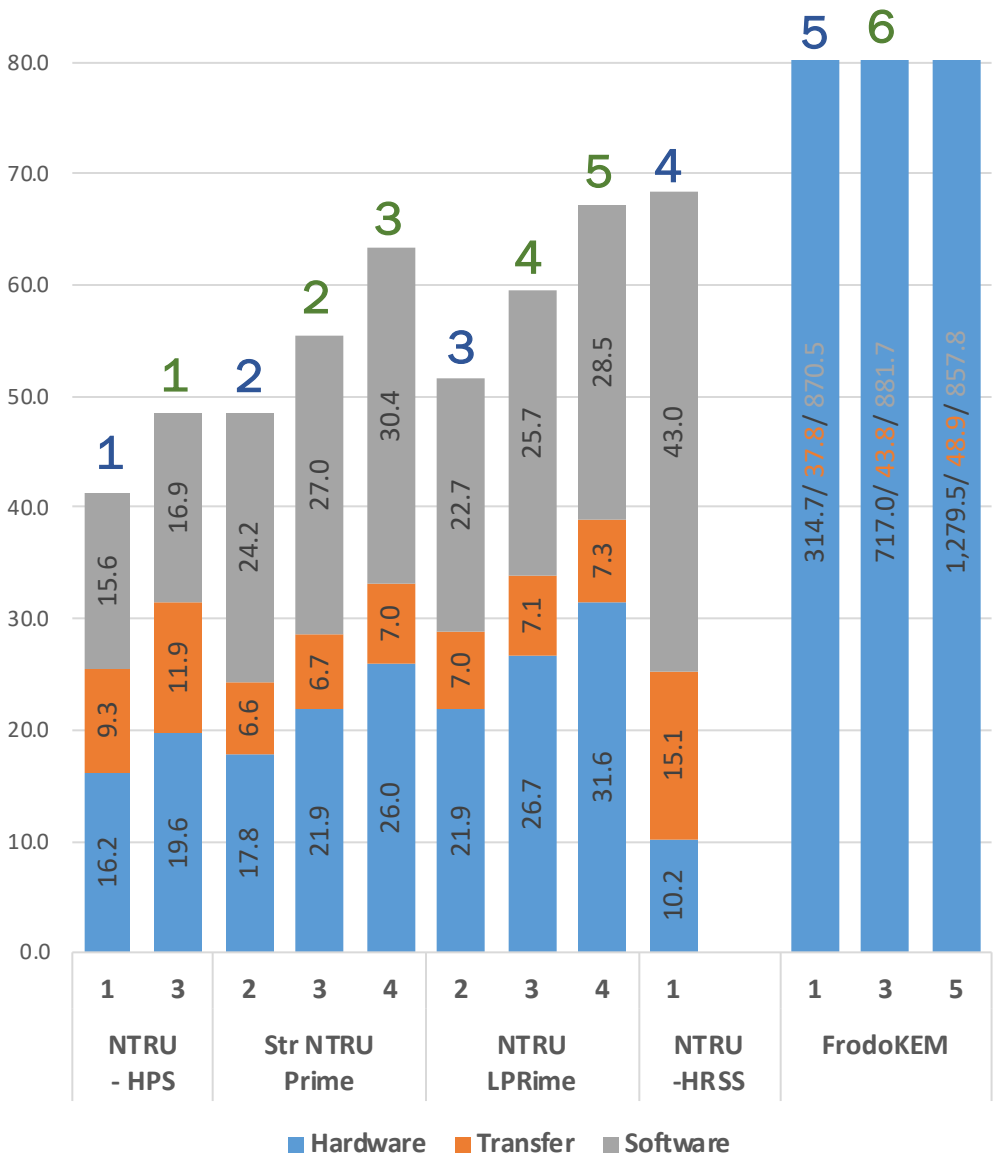
SW Part Sped up by HW[%]: Encapsulation



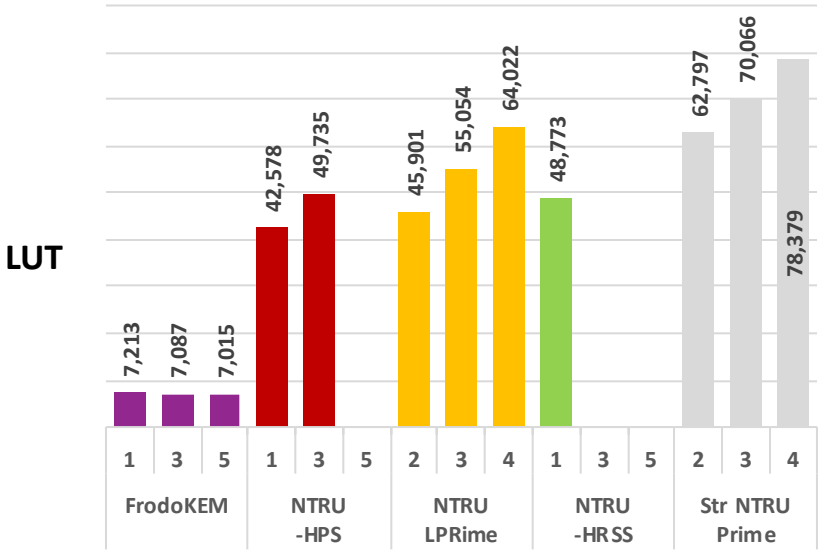
Round2 KEMs: SW/HW Results for Encaps

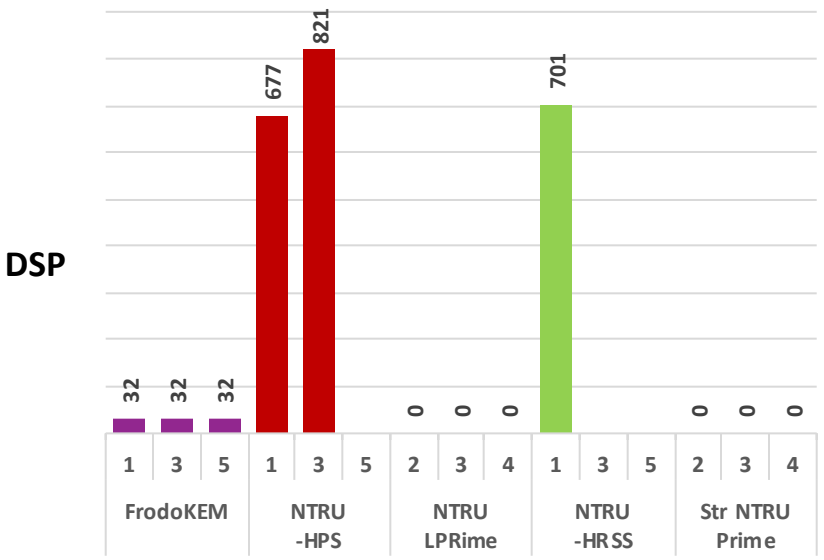


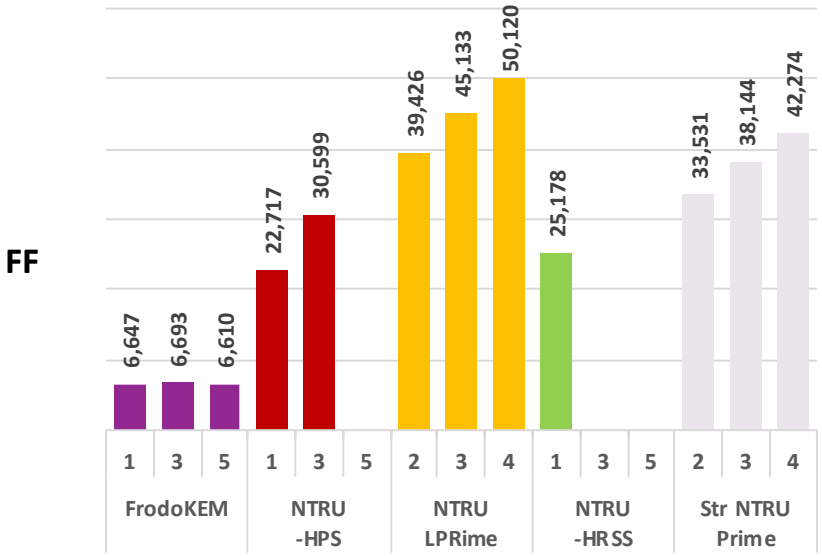
Round2 KEMs: SW/HW Results for Encaps

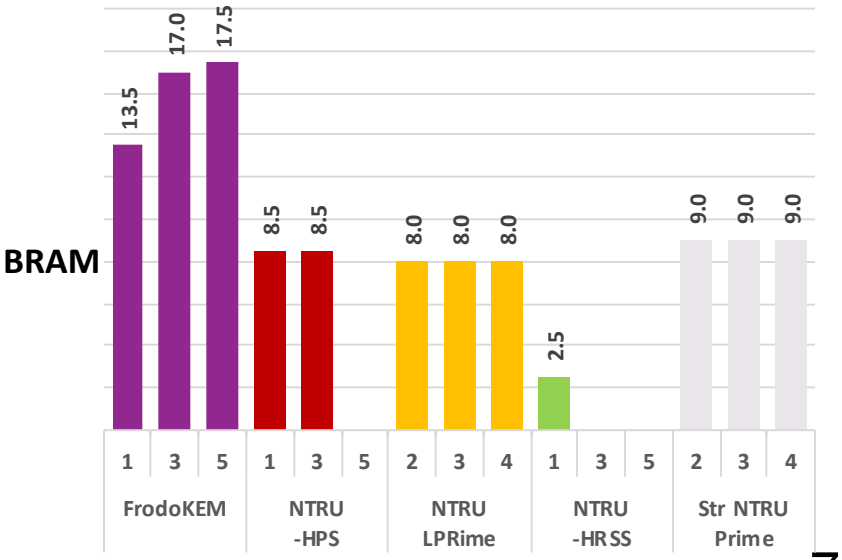


Resource Utilization on Zynq UltraScale+

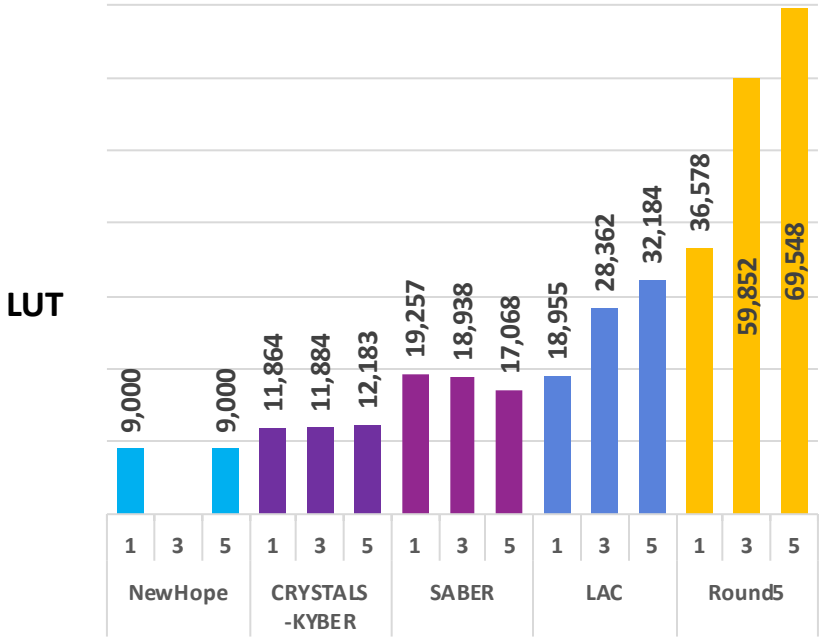


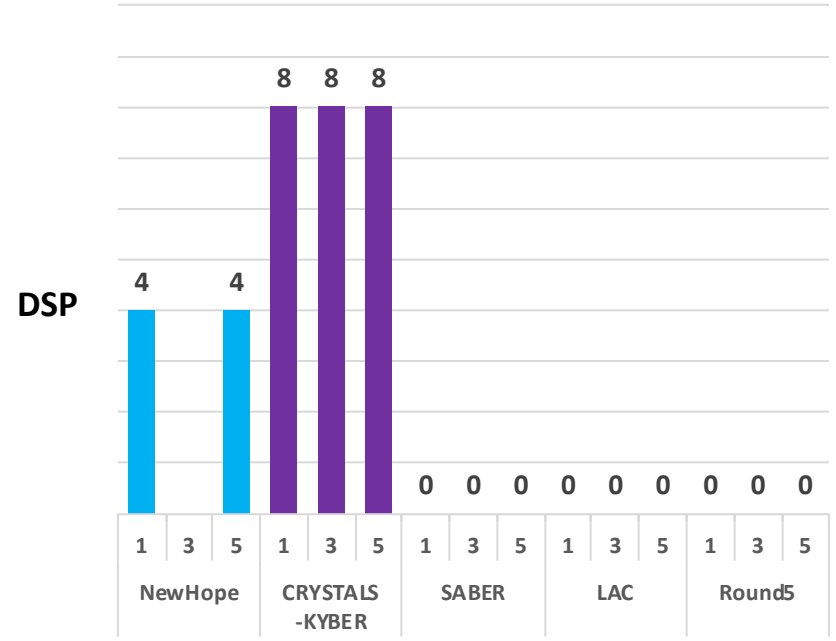


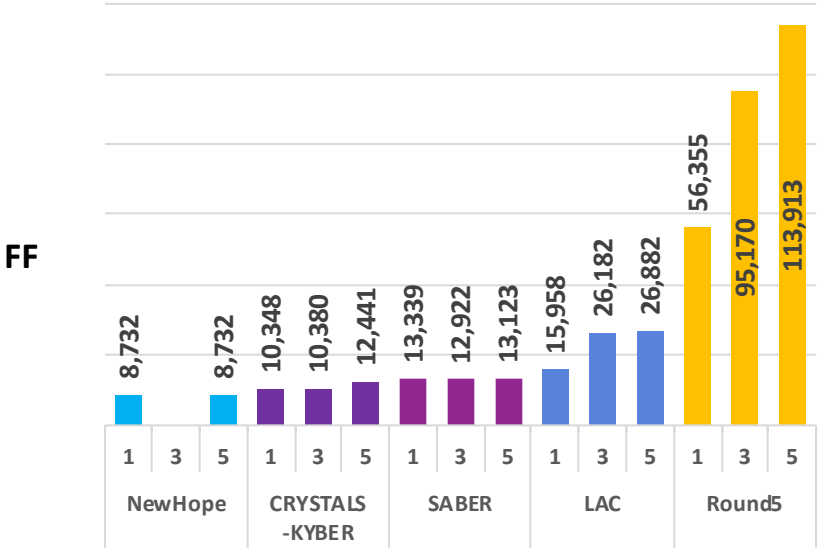


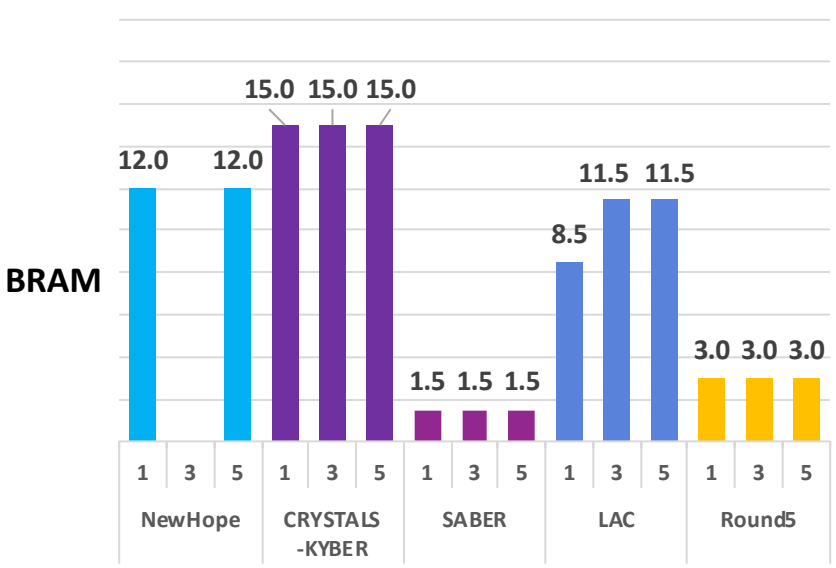


Resource Utilization on Artix-7









GMU SW/HW vs. Intel Xeon E3-1220 v3 (3.1 GHz)

Algorithm	median cycles	SW (us)	SW/HW (us)	Ratio
Encapsulation				
Level 1 & 2				
ntruhrss701	26116	8.4	68.3	0.12
ntruhps2048677	35352	11.4	41.2	0.28
kyber512	44404	14.3	15.2	0.94
sntrup653	46620	15.0	48.5	0.31
lightsaber2	67568	21.8	14.0	**1.56**
ntrulpr653	69400	22.4	51.6	0.43
lac128	82684	26.7	15.9	**1.67**
r5nd1kem0d	89500	28.9	16.7	**1.73**
newhope512cca	109040	35.2	15.0	**2.34**
r5nd1kem5d	122492	39.5	13.8	**2.85**
frodokem640shake	4529184	1,461.0	1,223.0	**1.19**
Level 3				
ntruhps4096821	43100	13.9	48.4	0.29
sntrup761	48780	15.7	55.5	0.28
ntrulpr761	72372	23.3	59.6	0.39
kyber768	74040	23.9	17.9	**1.34**
saber2	115948	37.4	18.7	**2.00**
lac192	158628	51.2	21.4	**2.39**
r5nd3kem5d	209572	67.6	19.2	**3.52**
r5nd3kem0d	317244	102.3	21.9	**4.67**
frodokem976shake	9467152	3,053.9	1,642.5	**1.86**
Level 4 & 5				
sntrup857	60668	19.6	63.4	0.31
ntrulpr857	91416	29.5	67.3	0.44
kyber1024	103936	33.5	22.1	**1.52**
firesaber2	175844	56.7	23.7	**2.39**
lac256	188244	60.7	23.8	**2.55**
newhope1024cca	201772	65.1	21.3	**3.06**
r5nd5kem5d	368004	118.7	26.0	**4.57**
r5nd5kem0d	392492	126.6	29.2	**4.34**
frodokem1344shake	16379980	5,283.9	2,186.2	**2.42**

Algorithm	median cycles	SW (us)	SW/HW (us)	Ratio
Decapsulation				
Level 1 & 2				
kyber512	37600	12.1	17.1	0.71
r5nd1kem0d	43000	13.9	19.3	0.72
sntrup653	59324	19.1	66.9	0.29
ntruhps2048677	62004	20.0	95.3	0.21
r5nd1kem5d	63624	20.5	15.7	**1.31**
ntruhrss701	63632	20.5	135.6	0.15
lightsaber2	69508	22.4	14.3	**1.57**
ntrulpr653	82732	26.7	70.9	0.38
lac128	105388	34.0	17.1	**1.99**
newhope512cca	109728	35.4	16.1	**2.19**
frodokem640shake	4494652	1,449.9	1,321.3	**1.10**
Level 3				
sntrup761	59120	19.1	78.9	0.24
kyber768	63916	20.6	20.1	**1.03**
ntruhps4096821	79448	25.6	107.1	0.24
ntrulpr761	85908	27.7	84.1	0.33
r5nd3kem5d	117028	37.8	22.8	**1.65**
saber2	118848	38.3	19.5	**1.97**
r5nd3kem0d	156692	50.5	27.0	**1.87**
lac192	243008	78.4	23.7	**3.30**
frodokem976shake	9380108	3,025.8	1,866.2	**1.62**
Level 4 & 5				
sntrup857	80904	26.1	86.8	0.30
kyber1024	91628	29.6	24.7	**1.20**
ntrulpr857	112116	36.2	97.5	0.37
firesaber2	182136	58.8	24.8	**2.37**
r5nd5kem0d	193228	62.3	35.9	**1.73**
newhope1024cca	206248	66.5	24.8	**2.68**
r5nd5kem5d	209136	67.5	31.7	**2.13**
lac256	377784	121.9	26.9	**4.54**
frodokem1344shake	16312844	5,262.2	3,119.9	**1.69**

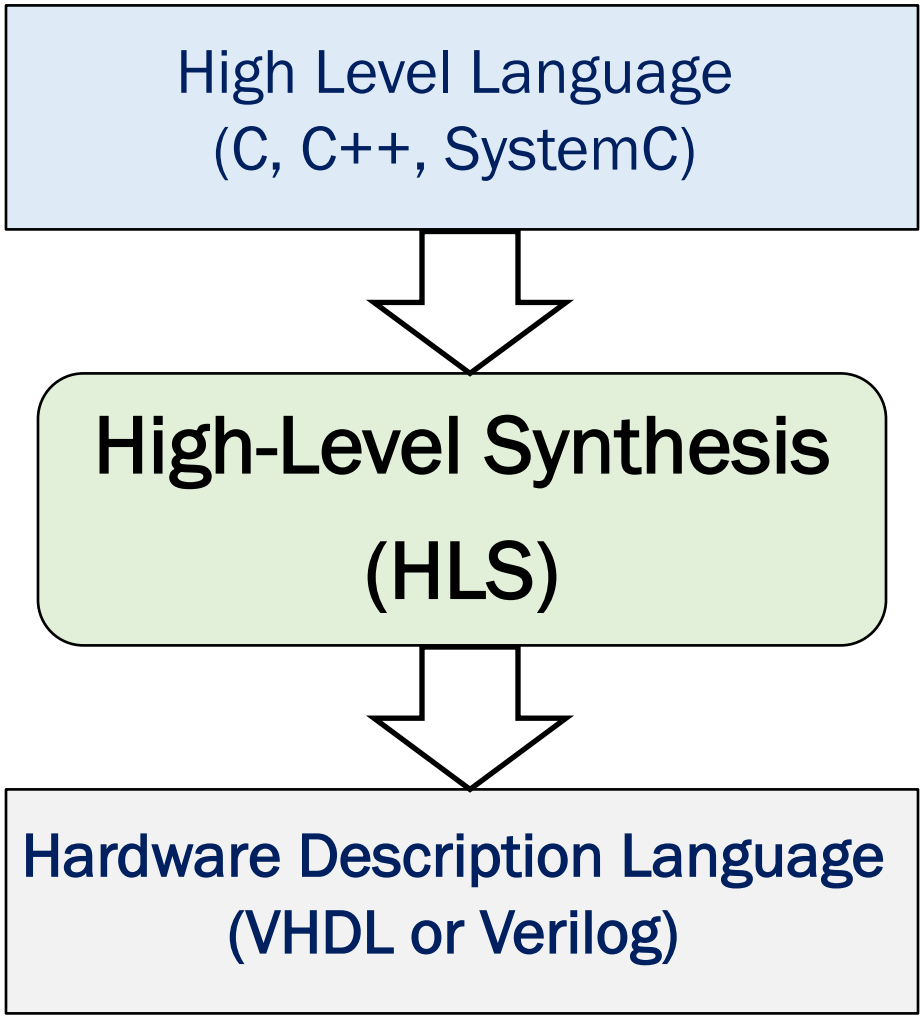

SW/HW
Co-Design
Conclusions

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 NewHope, CRYSTALS-KYBER, SABER & FrodoKEM resource utilization almost independent of the security level
- Important 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

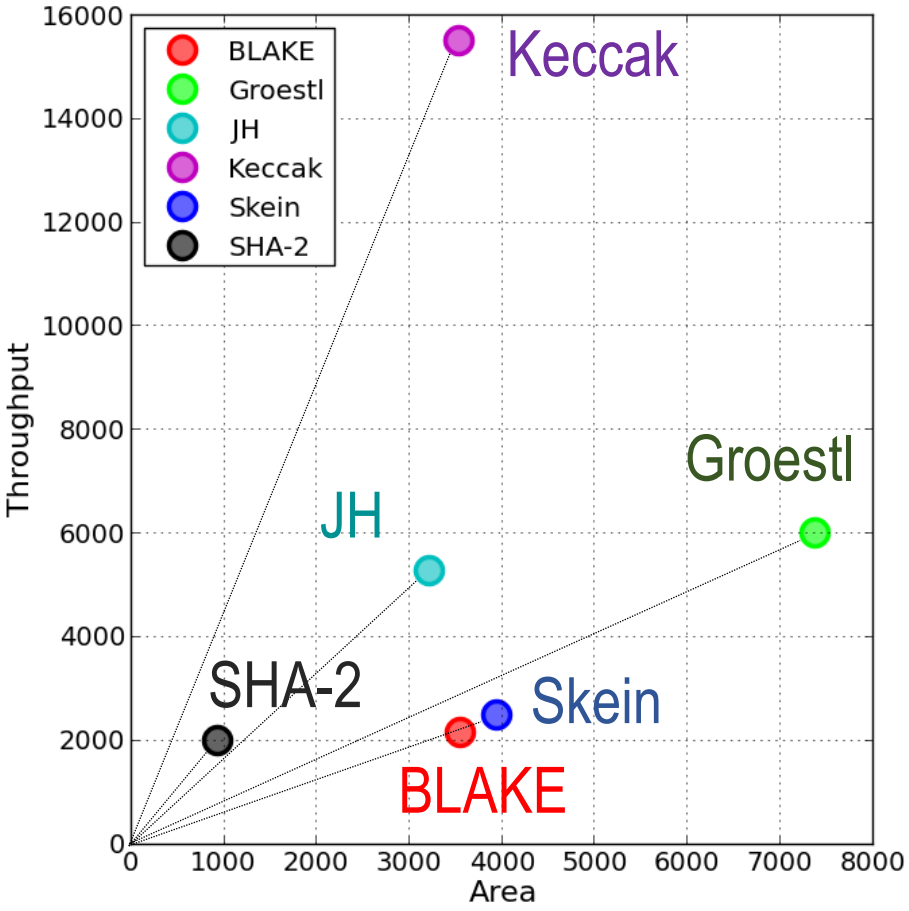
- 5 Final SHA_3 Candidates + SHA-2
  Applied Reconfigurable Computing, ARC 2015, Bochum, Apr. 2015

- 16 Round 3 CAESAR Candidates + AES-GCM
  Field Programmable Technology Conference, Melbourne, Dec. 2017

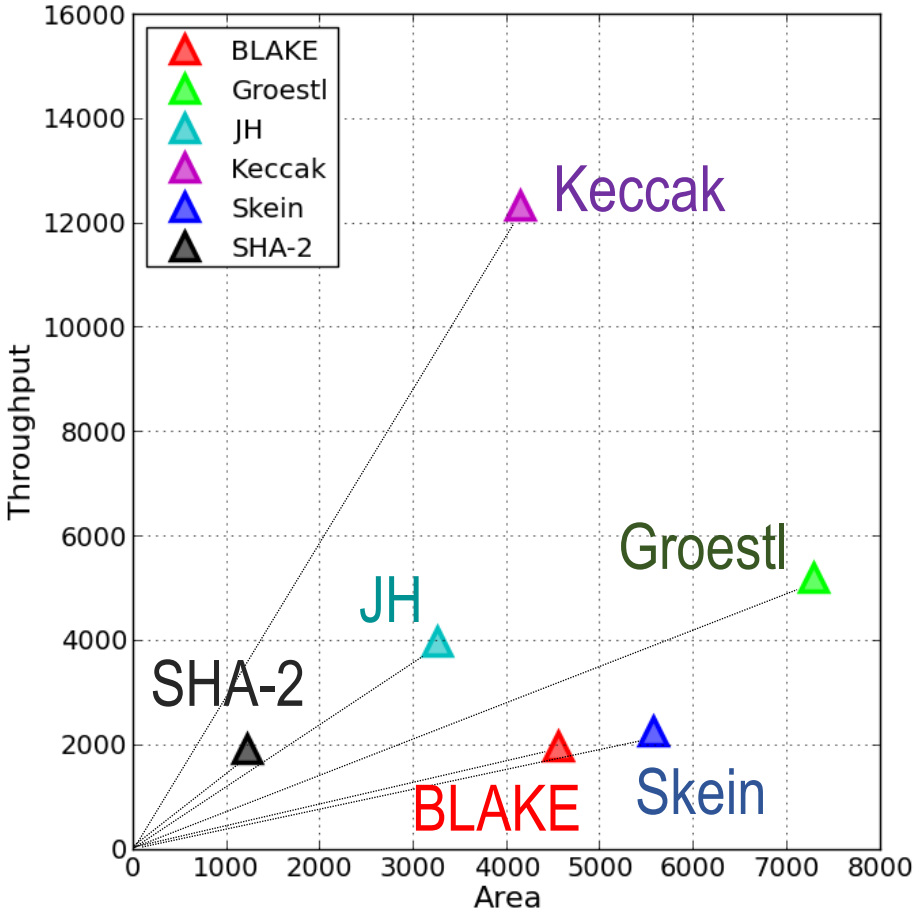
Ekawat Homsirikamol
a.k.a “Ice”

HLS vs. Manual: SHA-3 Candidates Revisited

Altera Stratix III FPGA



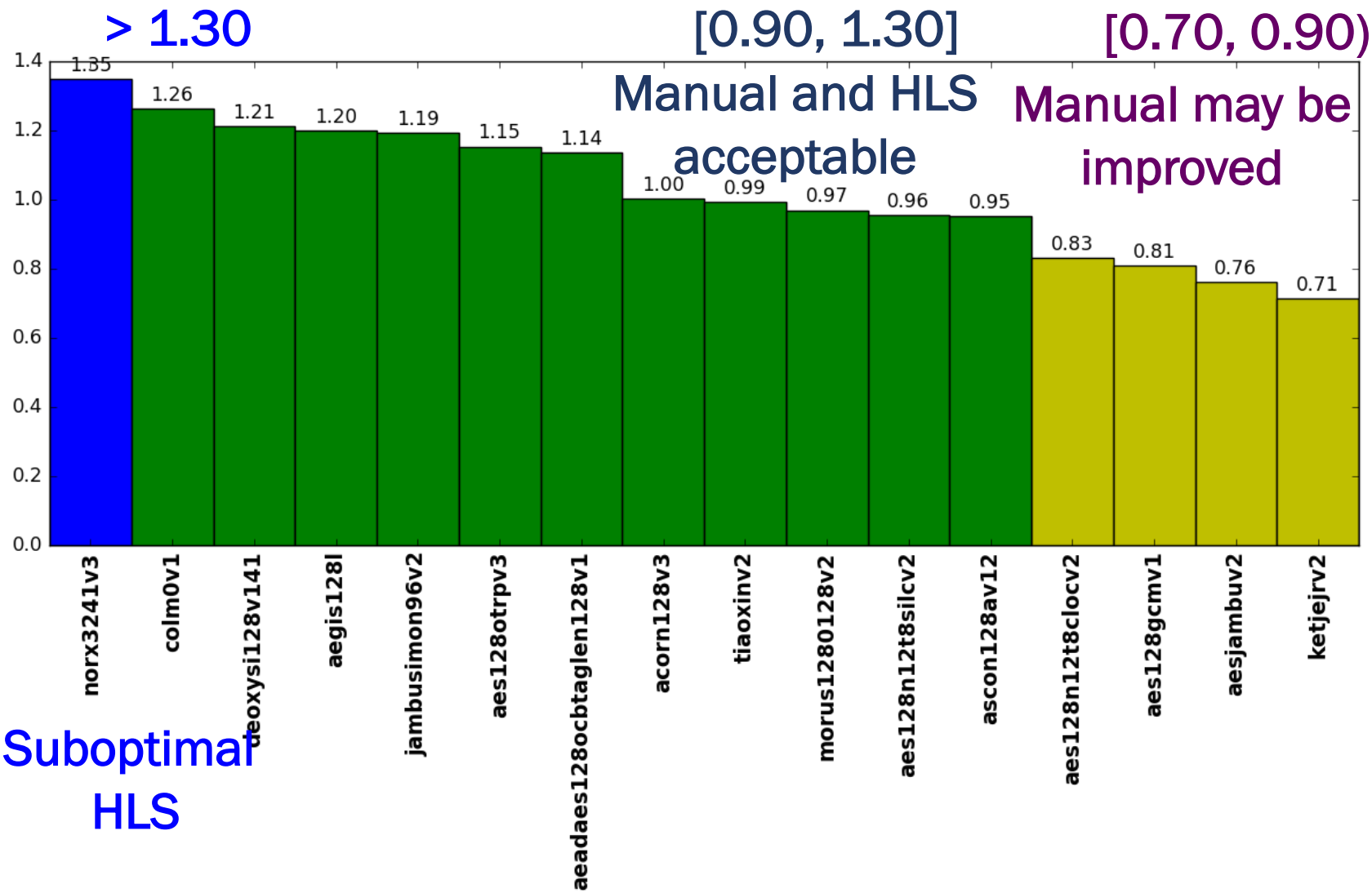
Manual



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

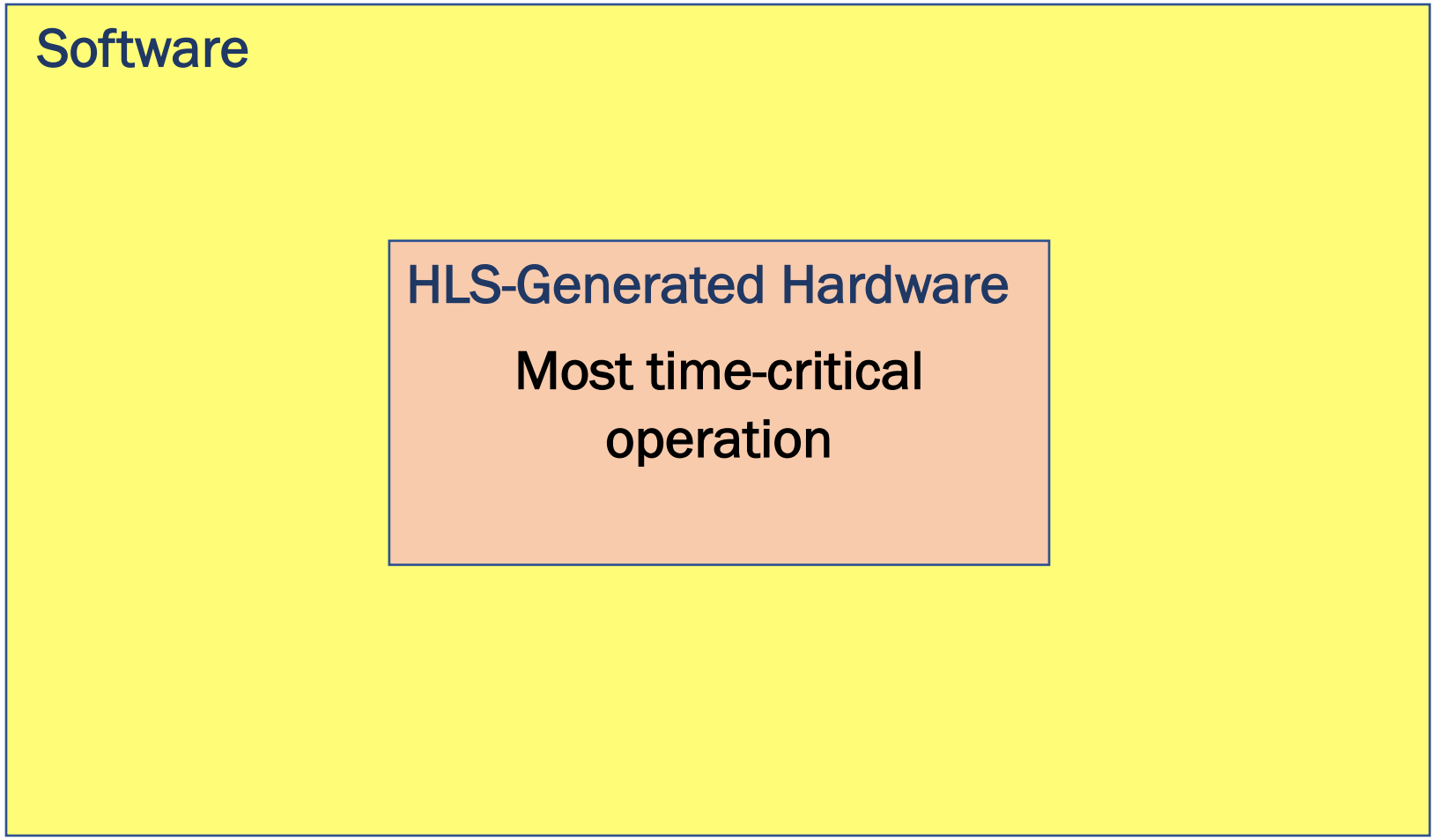
2. Addition of HLS Tool directives (pragmas)

3. Hardware-driven code refactoring

Sources of Productivity Gains

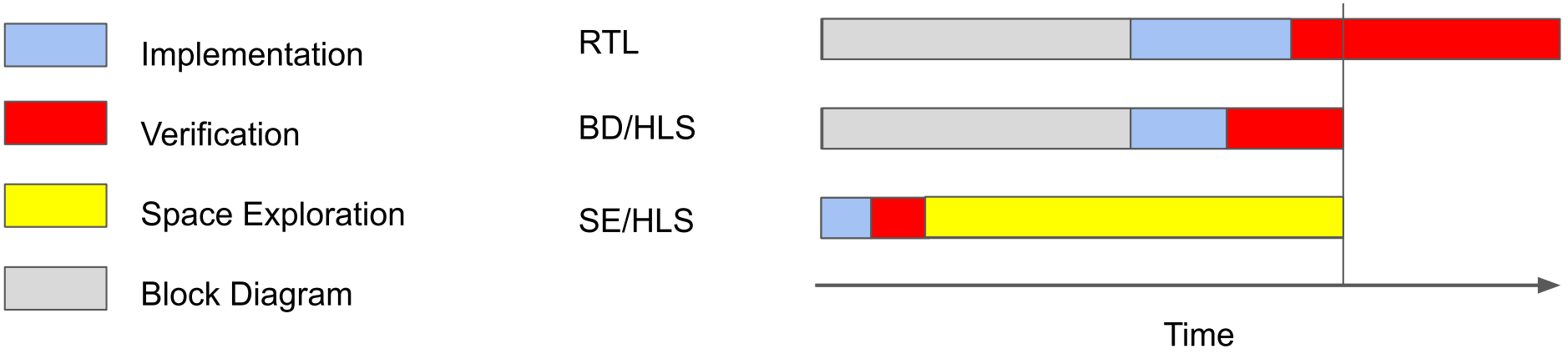
- Higher-level of abstraction
- Focus on datapath rather than control logic
- Debugging in software (C/C++)
  - Faster run time
  - No timing waveforms

Software/Hardware Codesign with HLS



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

Time spent on particular phases of the development process:



SW/HW Co-design: GMU Case Study
Applied Reconfigurable Computing, ARC 2020

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

Round 3

NIST Announcement on July 22, 2020

Round 3 Candidates

FINALISTS

Encryption/KEM

- Lattice-based
  - CRYSTALS-KYBER
  - NTRU
  - SABER
- Code-based
  - Classic McEliece

Digital Signature

- Lattice-based
  - CRYSTALS-DILITHIUM
  - FALCON
- Multivariate
  - Rainbow

ALTERNATE

Encryption/KEM

- Lattice-based
  - FrodoKEM
  - NTRU Prime
- Code-based
  - BIKE
  - HQC
- Isogeny-based
  - SIKE

Digital Signature

- Symmetric-based
  - Picnic
  - SPHINCS+
- Multivariate
  - 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

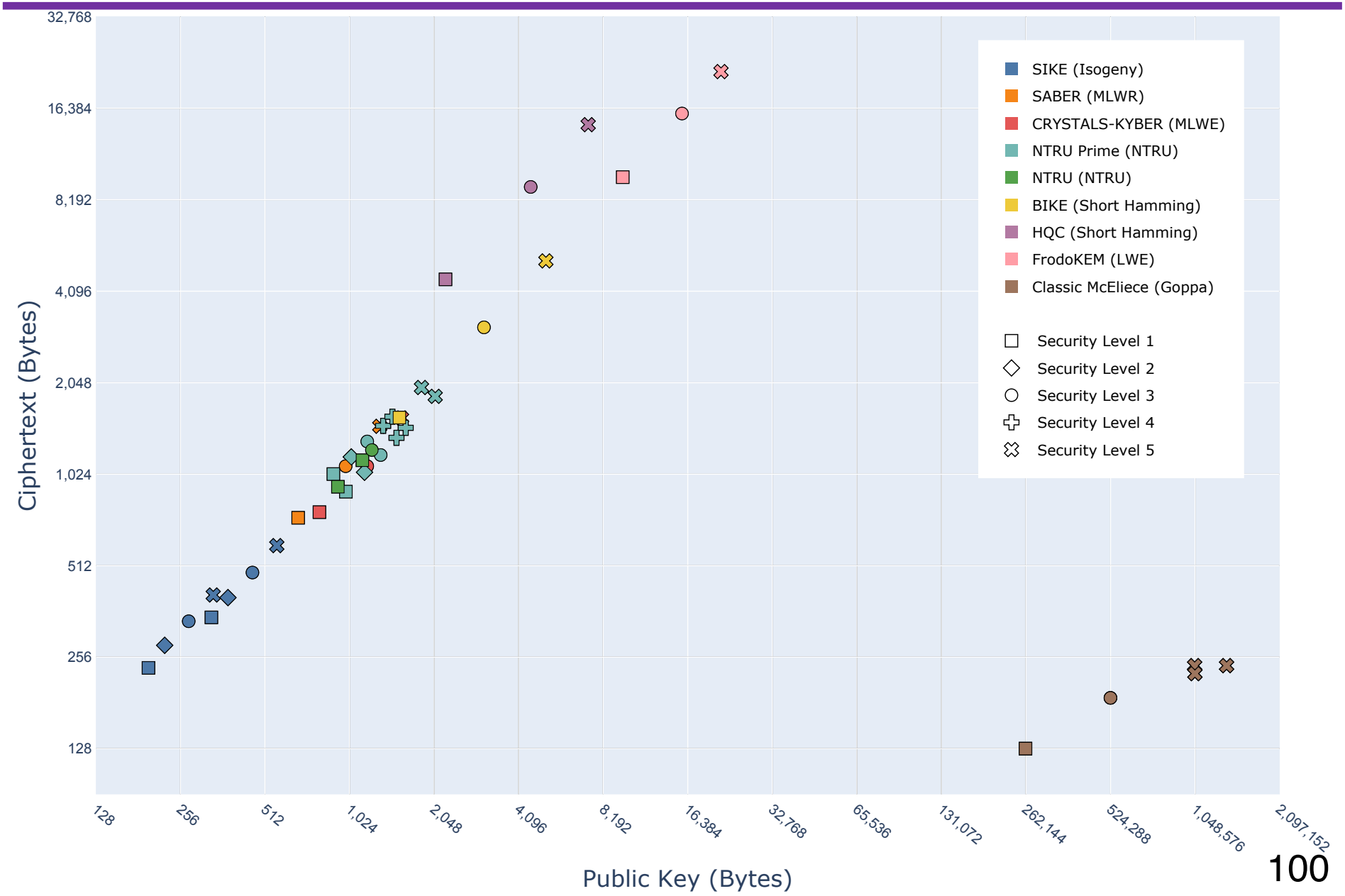
No references to papers on hardware implementations.
All decisions based solely on **security analysis**
and (to lower extent) **performance in software**.

NSA's Cybersecurity Perspective on PQC
July 29, 2020

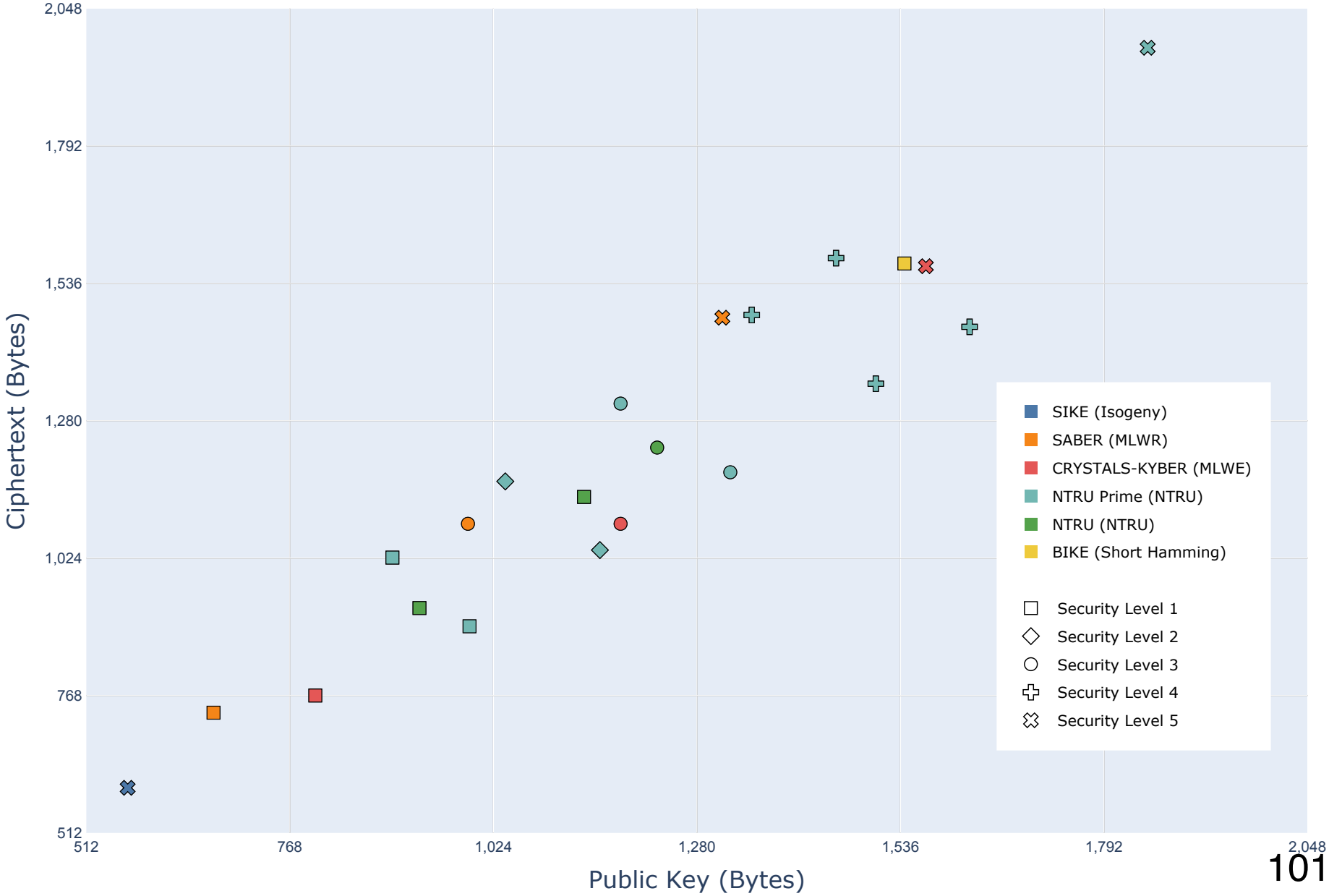
- Strong preference for Lattice-Based Cryptography
  - "fairly well-studied"
  - "secure when well-parameterized"
  - "among the most efficient"
- Lattice-based KEM and digital signature scheme to be approved for National Security Systems (NSS)
- Stateful signature schemes, LMS and XMSS,
  - "have a limited number of allowable signatures per key"
  - "require the signer to maintain an internal state"

  to be approved for NSS solutions for certain niche applications
- NSA CSD does not anticipate the need to approve other PQC schemes for NSS usage
  - "circumstances could change"

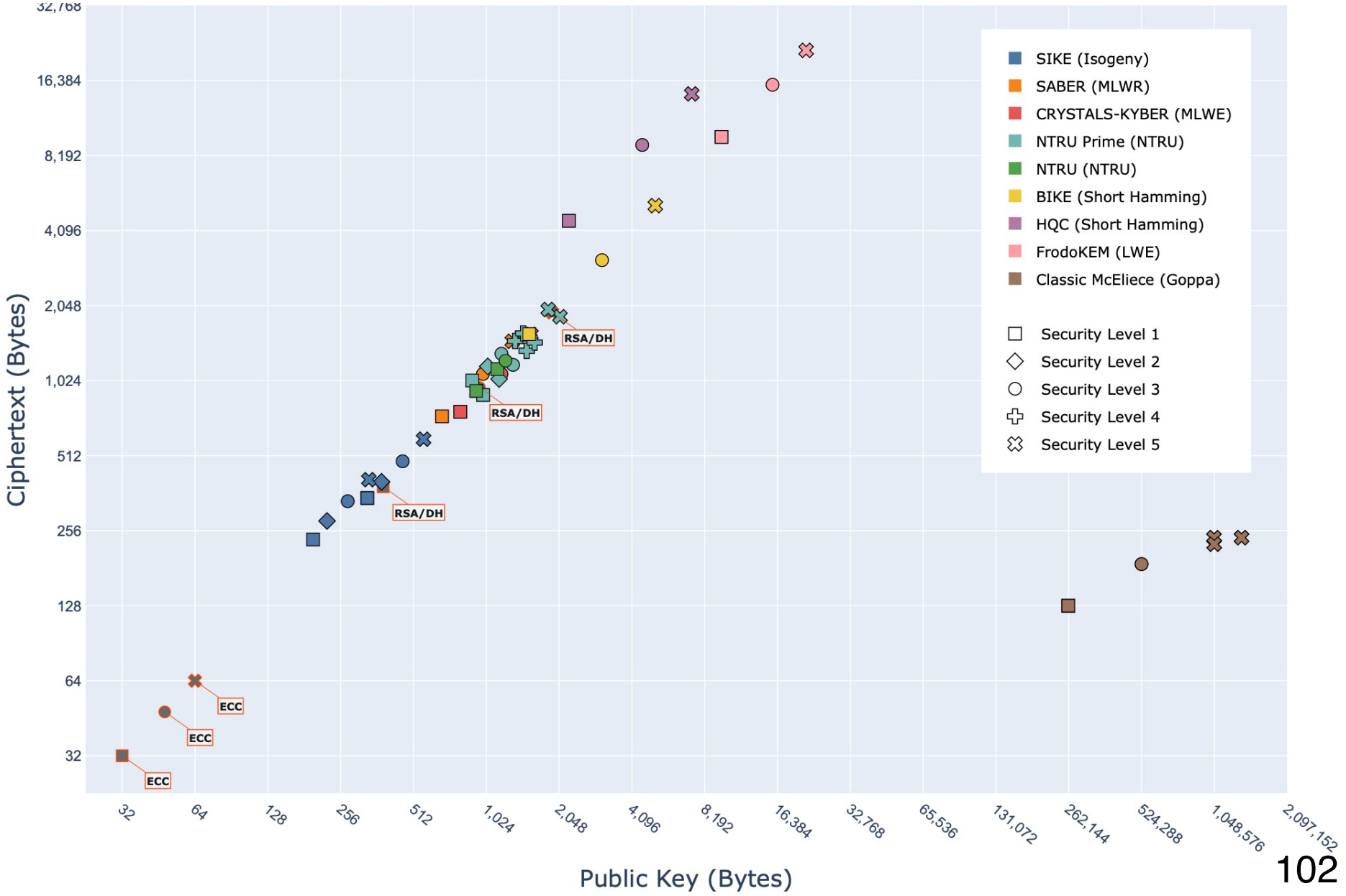
Round 3 Encryption/KEMs



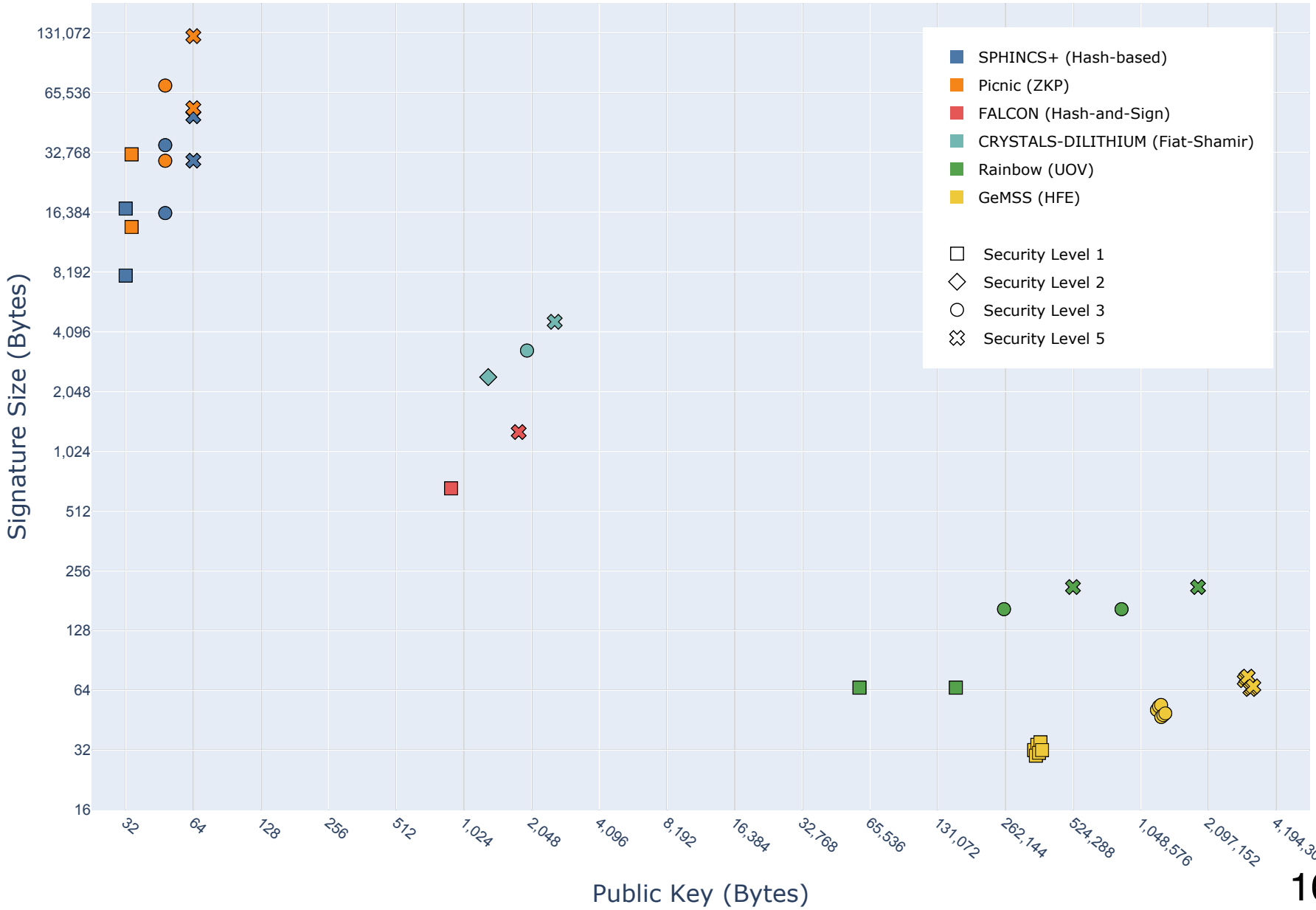
Round 3 Encryption/KEMs



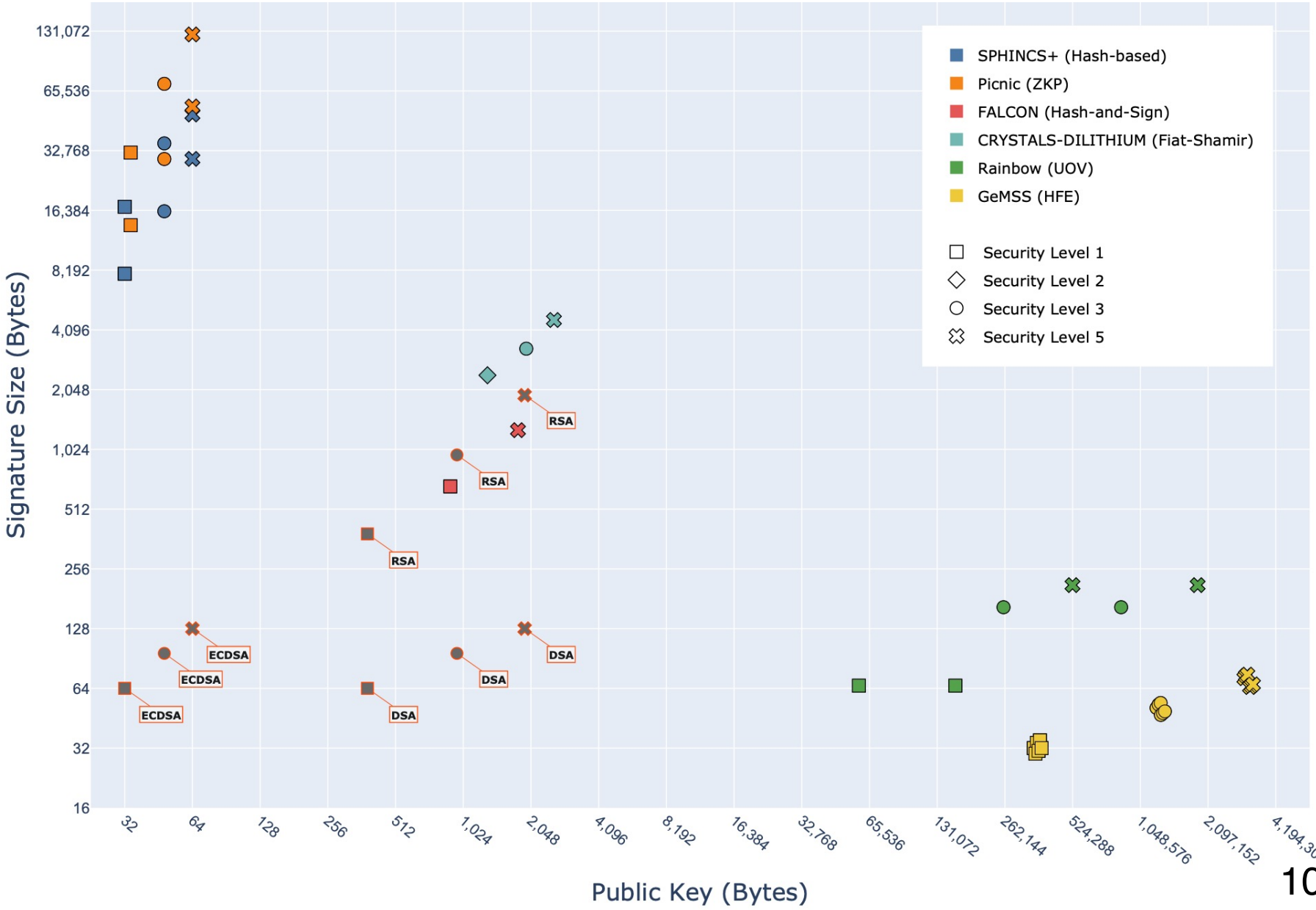
Round 3 Encryption/KEMs + Classical PKE



Round 3 Digital Signature Schemes



Round 3 + Classical Digital Signature Schemes



Close Matchups

KEMs

CRYSTALS-KYBER

Module-LWE:
Module Learning
with Errors

NTRU

SVP
Shortest Vector
Problem

SABER

Module-LWR:
Module Learning
with Rounding

Digital Signatures

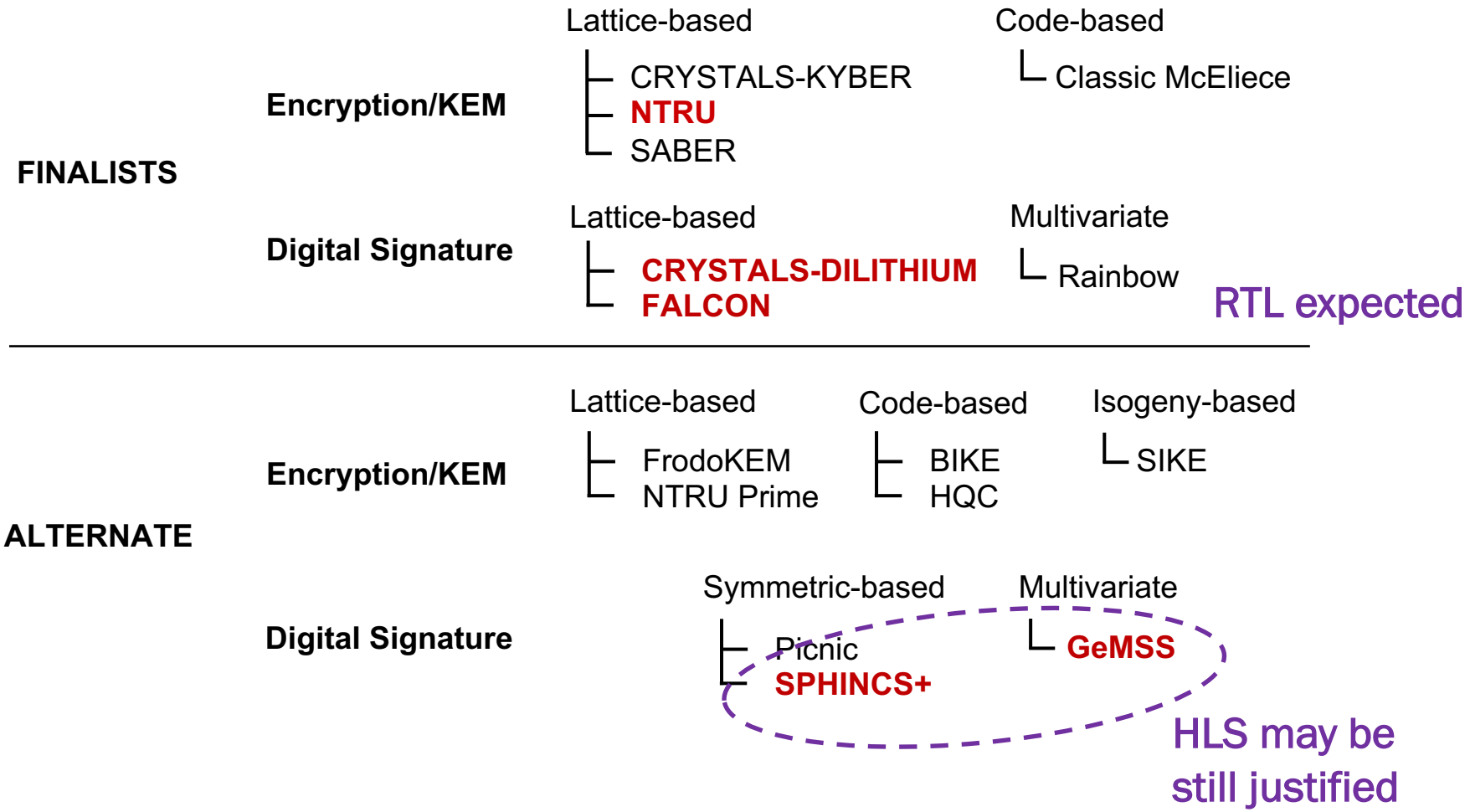
CRYSTALS-DILITHIUM

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

FALCON

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

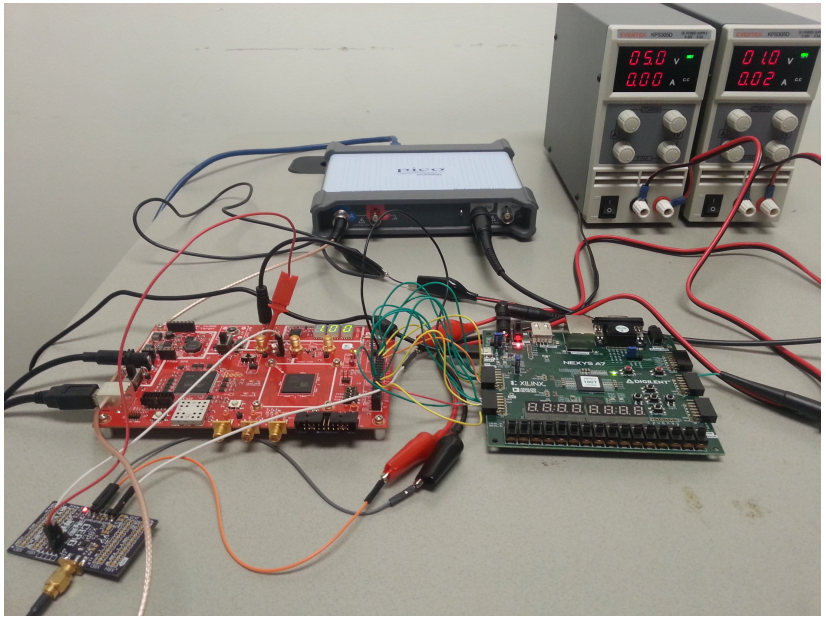
Round 3 Candidates without HW Implementations

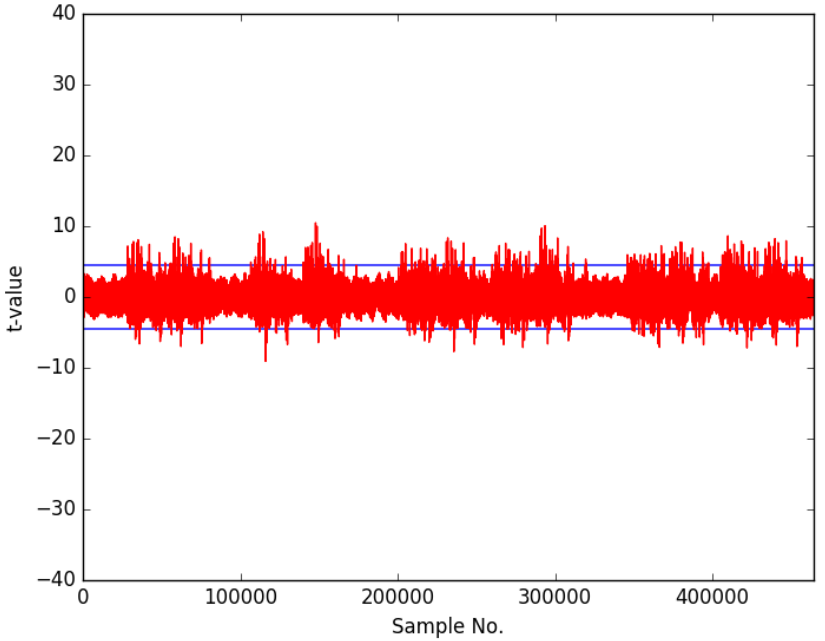


Future Work Directions

- More focus on hardware implementations vs. software/hardware implementations
- More focus on comparisons across families, rather than within the same family
- More hardware platforms to focus on
- Optimized software implementations targeting vector instructions of embedded processors, such as RISC-V and ARM
- Investigation of lightweight implementations protected against side-channel should be conducted by multiple groups, serving interchangeably as attackers and defenders
- Trade-offs among speed, area, power, energy, and resistance against side channel attacks

Evaluation of SCA-protected implementations





- Practical experiments
- Information leakage assessments
- Actual attacks
- Development and evaluation of various SCA countermeasures

Instead of Conclusions...

Topics for Discussion

Priorities

- Hardware vs. SW/HW with ARM vs. SW/HW with RISC-V
- IND-CPA PKE vs. IND-CCA KEM vs. Digital Signatures
- High-speed vs. lightweight
- Finalists vs. alternates

Need for a common FPGA platform

Need for a common Hardware API

Role for High-Level Synthesis

How to make software/hardware benchmarking fair?

Defining speed-up vs. software

Reliable ways of evaluating resistance against SCA

Reproducibility of results vs. publication cycle length

Publication standards: Mathematical vs. engineering improvements

Written report, online database of results, both?

Arguments against using only Artix-7

1. Low-cost family. Not suitable for high-speed implementations.
2. Traditional FPGA, not an SoC FPGA. Suitable only for SW/HW co-designs with ”soft” processor cores, such as RISC-V.
3. Unsuitable for HLS designs.
4. Relatively old FPGA family, released by Xilinx in 2010.
5. It is not customary to base ranking of candidates in cryptographic contests on results obtained for a single family of a single vendor.
6. Multiple reviewers of papers devoted to implementations of Round 2 PQC candidates treated the NIST’s choice of Artix-7 as an absolute requirement!

Recommended FPGA Platforms

1. For lightweight hardware implementations and lightweight software/hardware implementations based on soft processor cores:

   Xilinx Artix-7 and Intel Cyclone 10 LP.

2. For lightweight software/hardware implementations based on hard processor cores:

   Xilinx Zynq 7000-series and Intel Cyclone V SoC FPGAs.

3. For high-speed hardware and high-speed software/hardware implementations:

   Zynq Xilinx UltraScale+ and Intel Stratix 10 SoC.

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