



# PQ.V.ALU.E: Post-Quantum RISC-V Custom ALU Extensions on Dilithium and Kyber

---

Konstantina Miteloudi, Joppe Bos, Olivier Bronchain, Björn Fay, Joost Renes

CARDIS - November 16, 2023

## ► Context

- Quantum computing will threaten traditional Public Key Cryptography.
- Shift to Post-Quantum cryptography.
- NIST standardizes: CRYSTALS-Kyber (KEM) and CRYSTALS-Dilithium (Digital Signatures).

## ► Challenges in implementation

- Resource-constrained devices:
  - IoT, sensors, healthcare, automotive processors.
  - Limited computational capabilities, energy resources, memory.

- ▶ Custom ALU
  - Lightweight ALU for NTT computations in Dilithium and Kyber.
  - Integrated into a 4-stage pipeline 32-bit RISC-V processor.
- ▶ ISA Extension
  - Ten new instructions for modular arithmetic and NTT butterfly operations.
- ▶ Efficiency
  - Over 80% reduction in cycle count compared to optimized assembly.
  - No decrease in specific microprocessor's operating frequencies.

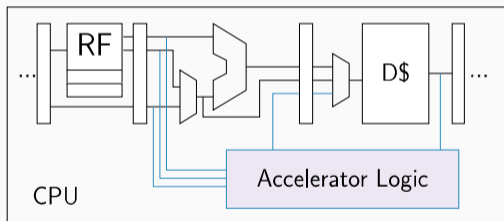
# Hardware accelerators

## ► Custom Extensions

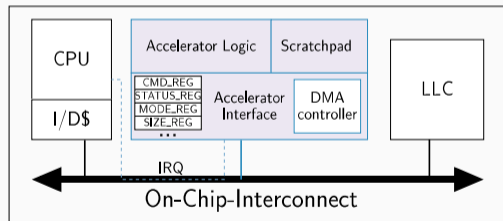
- Tailored instructions for specific applications.

## ► Need for Efficiency

- HW/SW co-design strategies for performance.

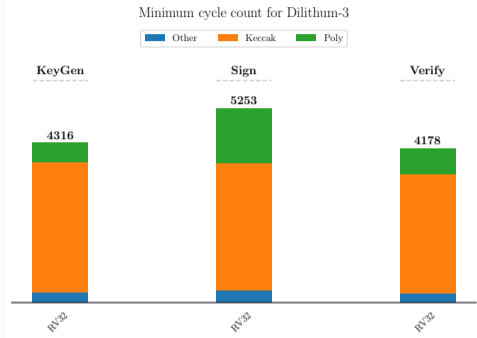


Integrated directly into the processor.(TCA)

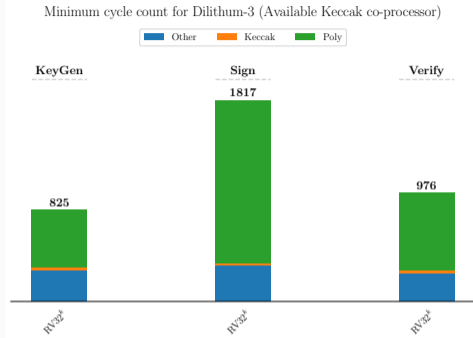


Added as peripherals to the processor.(LCA)

# Dilithium Profiling



RISC-V assembly

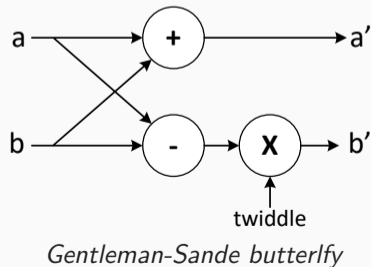
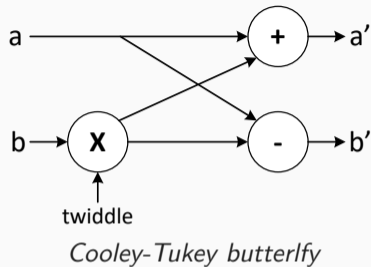
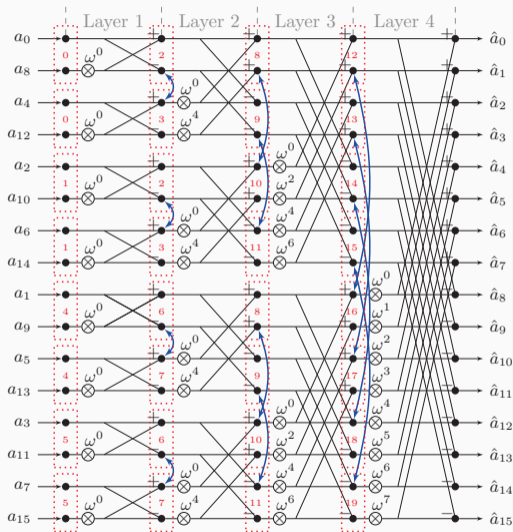


RISC-V assembly + Keccak co-processor

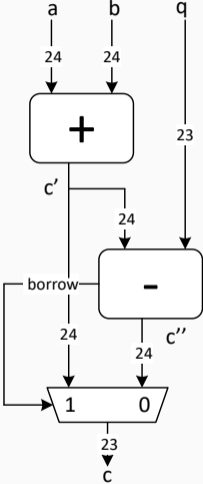
## ► Dominant factors

- Keccak is a significant portion of the runtime.
- Polynomial operations.

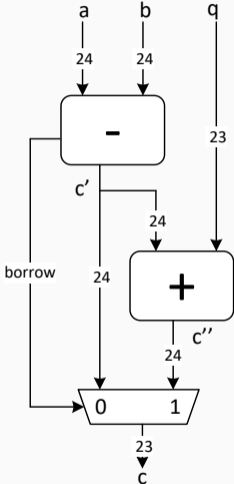
# Number-Theoretic Transform (NTT) and butterfly operations



# Modular addition and subtraction

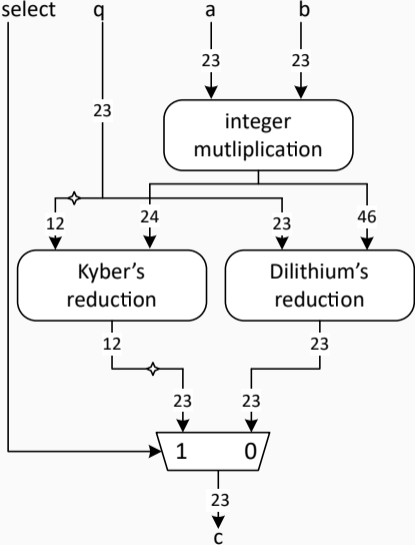


*Modular Addition*



*Modular Subtraction*

# Modular multiplication







---

## Algorithm 2 Barrett Reduction in Kyber

---

**Input:**  $0 \leq x < 3329^2$ ,

**Output:**  $z = x \bmod 3329$

1:  $t \leftarrow 5039 \cdot x$

2:  $t \leftarrow t \gg 24$

3:  $t \leftarrow (t \ll 11) + (t \ll 10) + (t \ll 8) + t$

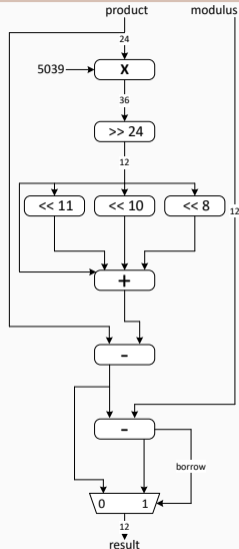
4:  $z \leftarrow x - t$

5: **if**  $z \geq 3329$  **then**

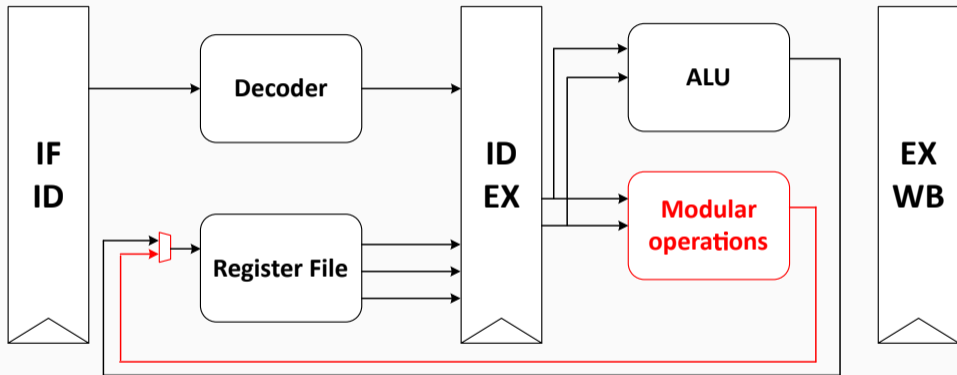
6:      $z \leftarrow z - 3329$

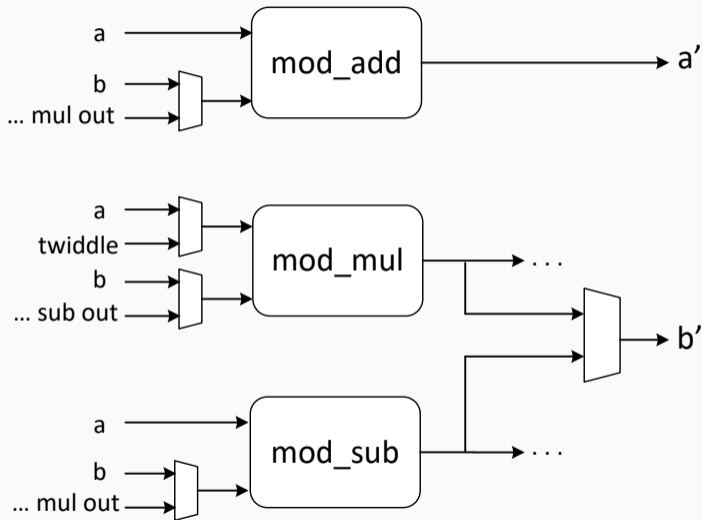
7: **return**  $z$

---

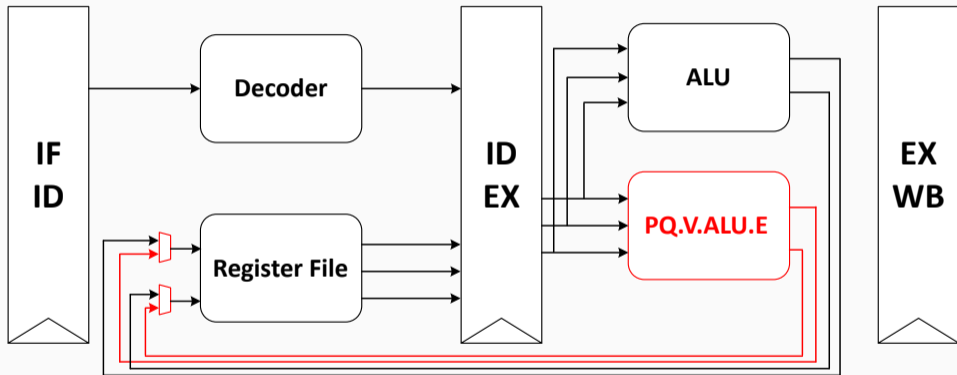


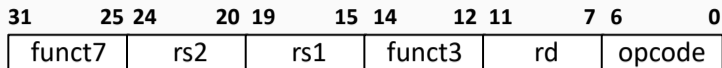
## Modified datapath





## Modified datapath





opcode	funct3	funct7	operation name
1110111	001	0000000	pq.mod_add_dil
1110111	010	0000000	pq.mod_sub_dil
1110111	011	0000000	pq.mod_mul_dil
1110111	100	0000000	pq.ct_btrfly_dil
1110111	101	0000000	pq.gs_btrfly_dil
1110111	001	0000001	pq.mod_add_kyb
1110111	010	0000001	pq.mod_sub_kyb
1110111	011	0000001	pq.mod_mul_kyb
1110111	100	0000001	pq.ct_btrfly_kyb
1110111	101	0000001	pq.gs_btrfly_kyb

## Butterfly with custom assembly (1/2)

```
.macro montgomery a1, ah, qi, q
    mul \a1, \a, \qi
    mulh \a1, \a1, \q
    sub \a1, \ah, \a1
.endm

.macro ct_butterfly a, b, qi, q, zeta,
    tmp
    mul \tmp, \zeta, \b
    mulh \b, \zeta, \b
    montgomery \tmp, \b, \qi, \q
    sub \b, \a, \tmp
    add \a, \a, \tmp
.endm
```

(a) Cooley-Tukey, RV32

```
.macro montgomery a1, ah, qi, q
    mul \a1, \a, \qi
    mulh \a1, \a1, \q
    sub \a1, \ah, \a1
.endm

.macro gs_butterfly a, b, qi, q, zeta,
    tmp
    sub \tmp, \a, \b
    add \a, \a, \right
    mul \b, \zeta, \tmp
    mulh \tmp, \zeta, \tmp
    montgomery \b, \tmp, \qi, \q
.endm
```

(b) Gentleman-Sande, RV32

## Butterfly with custom assembly (2/2)

```
.macro ct_butterfly a, b, z, tmp
    pq.mod_mul \tmp, \z, \b
    pq.mod_sub \b, \a, \tmp
    pq.mod_add \a, \a, \tmp
.endm
```

(c) Cooley-Tukey, PQVALUE<sup>1</sup>

```
.macro ct_butterfly a, b, zeta
    pq.ct_btrfly \a, \b, \zeta
.endm
```

(e) Cooley-Tukey, PQVALUE<sup>2</sup>

```
.macro gs_butterfly a, b, zeta, tmp
    pq.mod_sub \tmp, \a, \b
    pq.mod_add \a, \a, \b
    pq.mod_mul \b, \zeta, \tmp
.endm
```

(d) Gentleman-Sande, PQVALUE<sup>1</sup>

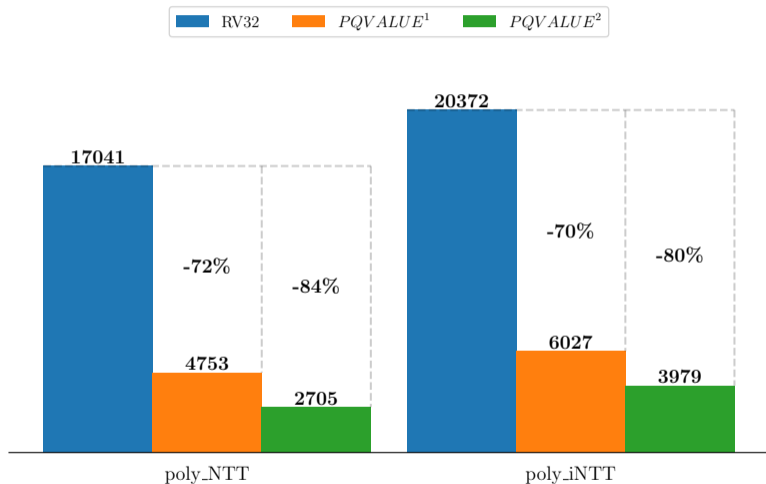
```
.macro gs_butterfly a, b, zeta, tmp
    pq.gs_btrfly \a, \b, \zeta
.endm
```

(f) Gentleman-Sande, PQVALUE<sup>2</sup>

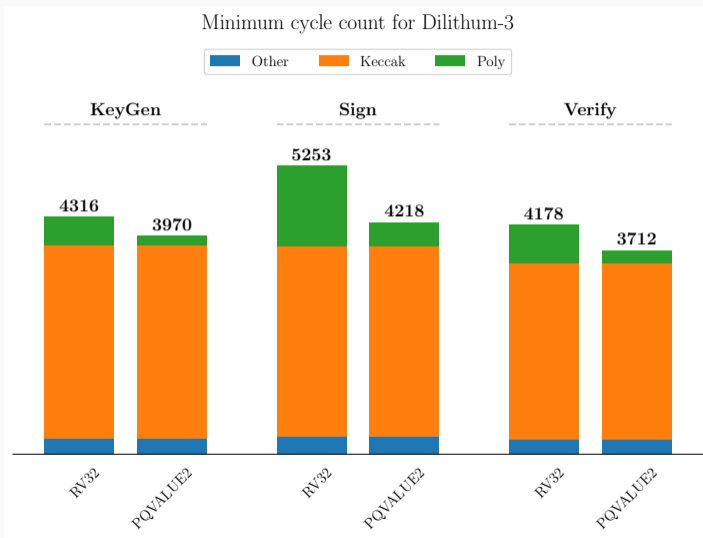


# Cycles for polynomial operations

Cycle counts of polynomial operations in Dilithium

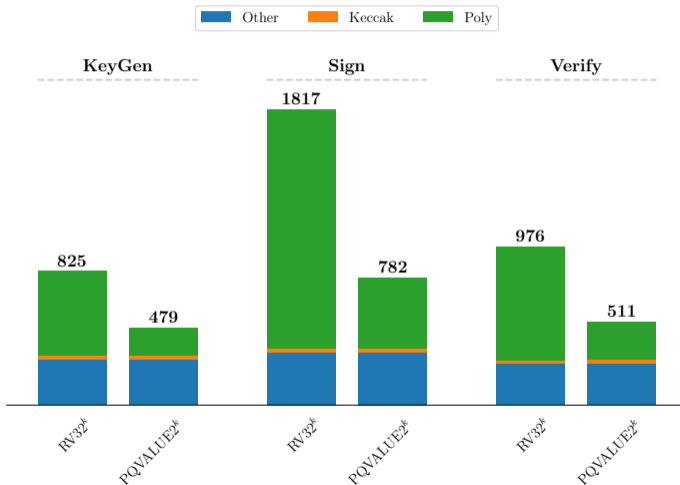


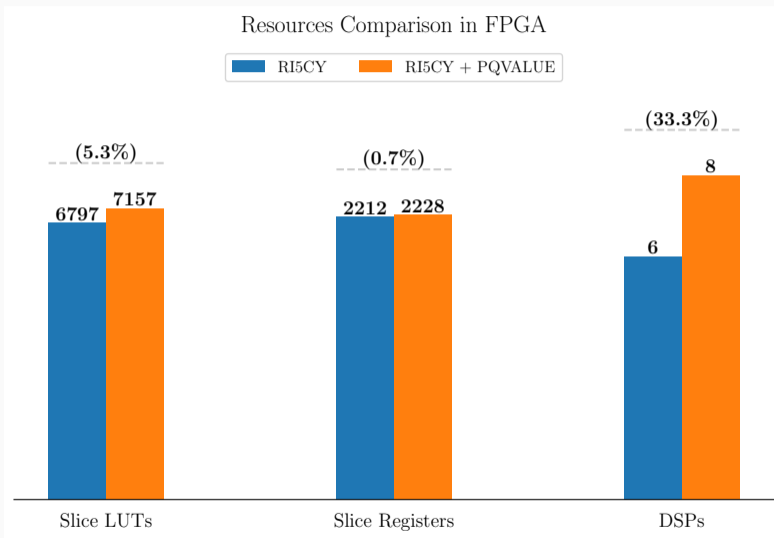
# Cycles for Dilithium per phase



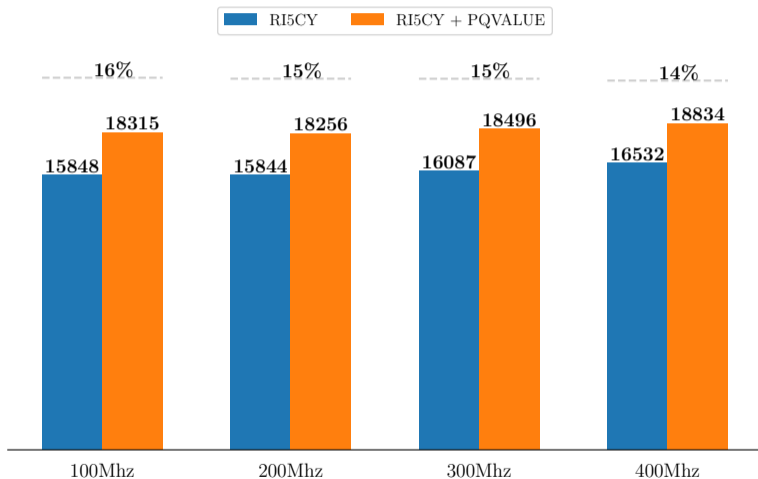
# Cycles for Dilithium with Keccak co-processor

Minimum cycle count for Dilithium-3 (Available Keccak co-processor)





Resources Comparison in ASICs (Total Cell Area in  $\mu m^2$ )



## Size and efficiency comparison of post-quantum ALUs

	Resources				Kyber perf.		
	LUT	Reg.	DSP	BRAM	Core	NTT	NTT <sup>-1</sup>
PQR-ALU [8]	2 908	170	9	0	RI5CY	1 935	1 930
PQ ALU [17]	555	0	15	1	CVA6	18 448	18 448
<b>PQVALUE<sup>2</sup></b>	<b>459</b>	<b>0</b>	<b>2</b>	<b>0</b>	<b>RI5CY</b>	<b>2 577</b>	<b>3 851</b>

[8] Fritzmann, T., Sigl, G., & Sepúlveda, J. (2020). RISQ-V: Tightly Coupled RISC-V Accelerators for Post-Quantum Cryptography. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2020(4), 239–280.

[17] P. Nannipieri, S. Di Matteo, L. Zulberti, F. Albicocchi, S. Saponara and L. Fanucci (2021), "A RISC-V Post Quantum Cryptography Instruction Set Extension for Number Theoretic Transform to Speed-Up CRYSTALS Algorithms," in IEEE Access, vol. 9, pp. 150798-150808.

- ▶ Custom ALU
  - Lightweight ALU for NTT computations in Dilithium and Kyber.
  - Integrated into a 4-stage pipeline 32-bit RISC-V processor.
- ▶ ISA Extension
  - Ten new instructions for modular arithmetic and NTT butterfly operations.
- ▶ Efficiency
  - Over 80% reduction in cycle count compared to optimized assembly.
  - No decrease in specific microprocessor's operating frequencies.

**Thank you :)**