





CMALU: Compact Fault-Tolerant Modular Arithmetic Logic Unit for Post-Quantum Cryptography

YoungBeom Kim[1], Malik Imran[2], Zain Ul Abideen[3], Ciara Rafferty[2], Ayesha Khalid[2], Máire O'Neill[2], and Seog Chung Seo[1]

[1]Kookmin University [2]Queen's University Belfast [3]University of Idaho

ICCAD 2026 · San Jose, CA, USA · November 2026

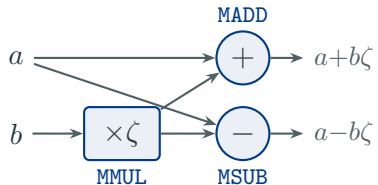
Lattice PQC runs on three modular operations

- ML-KEM, ML-DSA (FIPS 203/204), Falcon (FN-DSA): rings $\mathbb{Z}_q[x]/(x^n+1)$
- NTT-based polynomial multiplication dominates the cost
- Every butterfly needs `MMUL`, `MADD`, `MSUB`

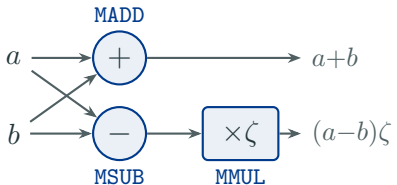
	q	$W = \lceil \log_2 q \rceil$
ML-KEM	3,329	12
ML-DSA	8,380,417	23
Falcon	12,289	14

Only q and W change.

CT butterfly (NTT)



GS butterfly (inverse NTT)



All arithmetic mod q. CT: Cooley–Tukey; GS: Gentleman–Sande.

Compact or protected: prior designs pick one

PQC accelerators

[AMI+22, WZCG22, MCL+23, BNV24]

- DSP multipliers or q-specific reduction
- Mostly modulus- or scheme-specific
- No fault detection

Fault-tolerant

[KKR+23, BSKA25, AAKA25]

- Parity, interpolation, recomputation
- Large area and energy overhead
- [AAKA25]: +3.0–5.7k LUT, 16 DSP

IoT endpoints need

- Small area, no DSP
- One unit for all three schemes
- Detection without extra cycles

Goal: a compact, flexible, fault-tolerant modular ALU.

Threat model: faults in the arithmetic core

Adversary: laser, EM or glitch with cycle-level timing

Fault model [AAKA25]: $V' = V \oplus \Delta$
transient: $\Delta \neq 0$; ST: data-dependent Δ

	flipped bits	duration
SF	1	transient
MF	2–5, any position	transient
BU	2–8, consecutive	transient
RT	arbitrary weight	transient
ST	stuck at 0 or 1	permanent

Attack 1 DFA on the NTT butterfly
A faulty butterfly output O'_t:
$O'_t \oplus O_t$ leaks secret coefficients [RYB+23].

Attack 2 Fault replay on BSM
W cycles per MMUL = W injection windows;
collect N undetected faulty outputs [HPP21].

Attack 3 Montgomery reduction skip
Stuck F = s can make U′ odd:
the right shift is no longer exact [AAKA25].

Scope: operands inside the modular datapath (MMUL, MADD, MSUB).
Memory, Keccak, control flow, FO checks, checker faults, side channels: orthogonal protection.

CMALU: one datapath, three protection modes

`ADD` `SUB` `MUL` `MADD` `MSUB` `MMUL`

one CSA/CLA datapath
+ bit-serial Montgomery, 0 DSP

Mode-0

Unprotected baseline

No detection

1× cycles

Mode-1

Parity prediction +
evenness invariant

Single-bit proof
on the checked word

1× cycles

Mode-2

Mode-1 + recomputation
on the same datapath

Observed SDC: 0%

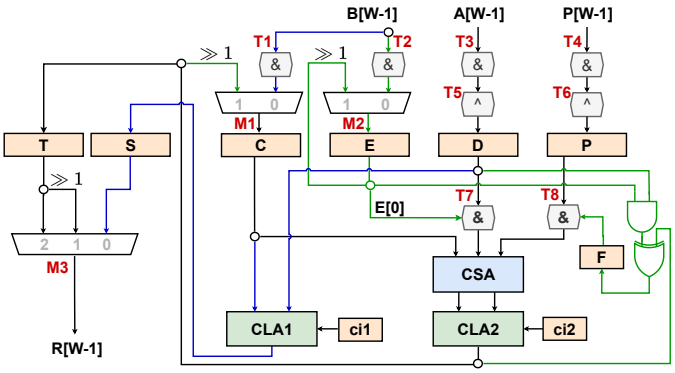
≈1.86× NTT cycles

Selected by a 2-bit `iMODE` input (fixed at synthesis on Ibex).
Validated in an NTT accelerator and an Ibex RISC-V core, on FPGA and ASIC.

Detection covers the modular paths (`MMUL`, `MADD`, `MSUB`); the integer operations share the datapath.

Architecture

Six operations on one CSA/CLA datapath



blue: ADD/SUB/MADD/MSUB, 2 cycles
green: MUL/MMUL, W cycles; black: shared

Bit-serial Montgomery [Mon85]

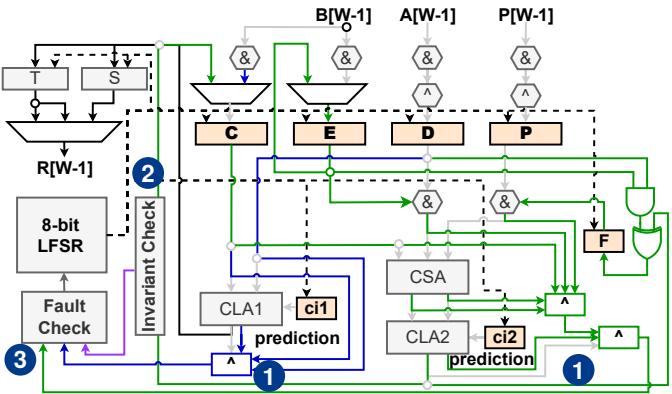
Ensure $R = \mathtt{D} \cdot \mathtt{E} \cdot 2^{-W} \bmod \mathtt{P}$

1: $\mathtt{C} \leftarrow 0$
2: **for** $i = 0$ **to** $W-1$
3: $\quad \mathtt{F} \leftarrow \mathtt{C}[0] \oplus (\mathtt{E}[0] \cdot \mathtt{D}[0])$
4: $\quad \mathtt{U} \leftarrow \mathtt{C} + \mathtt{E}[0] \cdot \mathtt{D} + \mathtt{F} \cdot \mathtt{P}$
5: $\quad \mathtt{C} \leftarrow \mathtt{U} \gg 1$ ($\mathtt{U}$ even)
6: $\quad \mathtt{E} \leftarrow \mathtt{E} \gg 1$
7: **if** $\mathtt{C} \geq \mathtt{P}$: $\mathtt{C} \leftarrow \mathtt{C} - \mathtt{P}$
8: **return** $R \leftarrow \mathtt{C}$

Line 4 on CSA + CLA2:
one bit per cycle, no DSP.
P: any odd modulus.

Algorithm 1 and Figure 1 in the paper. CLA1, CLA2: Kogge–Stone adders.

Mode-1: checkers built from the datapath



1 Parity prediction

- CLA2 every `MMUL` iteration; CLA1 for `MADD`/`MSUB`
- Shadow parity on `D`, `P`, `E`, `C`: `E` tracked as $p \leftarrow p \oplus \mathtt{E}[0]$

2 Evenness invariant

- $\mathtt{U}[0] = 0$ in every iteration

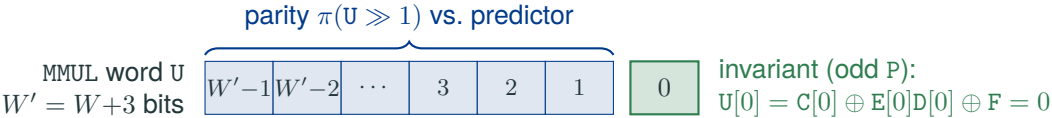
3 Fault check

- Mismatch → LFSR response

Off the critical path:
0 extra cycles.

Figure 2 in the paper. Purple path: invariant check. Fault check → 8-bit LFSR response.

Every single-bit flip of the checked word is caught



Theorem 1 (Mode-1):
any $V' = V \oplus 2^j$ is detected

- MMUL, $j{=}0$: $\mathtt{U}[0] \to 1$, invariant fires
- MMUL, $j{>}0$: $\pi(\mathtt{U} \gg 1)$ flips, parity fires
- MADD/MSUB: the word parity flips
- Stuck F (Attack 3): caught once any $\mathtt{F}_i \neq s$

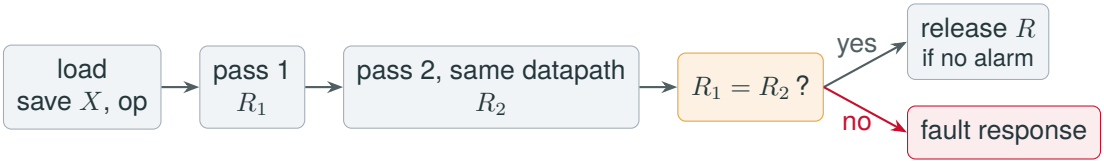
Uniform nonzero-mask model

MMUL miss: $\Delta[0] = 0 \wedge \mathrm{wt}(\Delta[W'{-}1{:}1])$ even

path	$\Pr[\text{miss}]$
MMUL (parity + invariant)	$< 1/4$
MADD/MSUB (parity)	$< 1/2$

Checker-observed word model, fault-free checkers; $\mathtt{U} < 4\mathtt{P} < 2^{W+2}$. Measured coverage: page 12.

Mode-2: recompute on the same hardware, fail closed



Theorem 2: $R_i = \Phi(X) \oplus \epsilon_i$,
comparator fires $\iff$ $\epsilon_1 \neq \epsilon_2$

- One-pass transient: $\epsilon_2 = 0 \neq \epsilon_1$, always detected
- $\epsilon_1 = \epsilon_2$ (e.g. permanent): comparator blind, **Mode-1** still checks
- Cost: $\approx$1.86$\times$ NTT cycles, $+$69 LUT

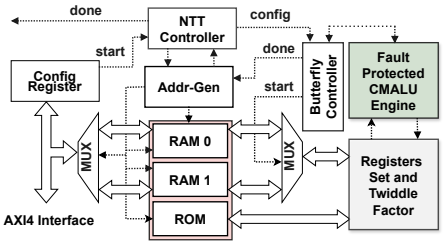
Fault response

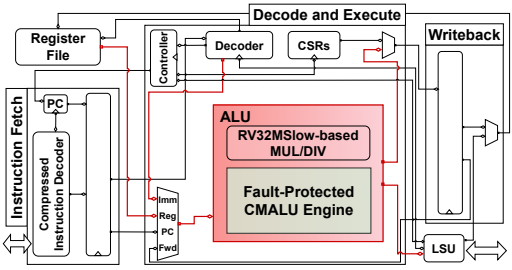
- Online checkers, within one cycle: output $\leftarrow$ LFSR, `FAULT_ABORT`, sticky `FAULT_ALERT`, state $\leftarrow$ LFSR
- Comparator mismatch: output suppression + alert

Assumes fault-free save/compare logic. $+$69 LUT: NTT-top increment over **Mode-1**. LFSR: not masking.

Evaluation

Two integration targets





NTT accelerator (loosely coupled)

- Radix-2 CT/GS butterfly, ping-pong BRAM, twiddle ROM
- One build per scheme: $W = 12$, 23, 14
- Artix-7 at 200 MHz; 65 nm ASIC

Ibex RISC-V core (tightly coupled)

- `CUSTOM-0`: `MMUL`, `MADD`, `MSUB`, `CFG`
- One $W = 23$ unit; `CFG` selects q at run time
- Protection mode fixed at synthesis

Figures 3 and 4 in the paper. Ibex: RV32MSlow, no DSP; Arty A7 at 50 MHz; Nangate45.

NTT accelerator: detection at zero extra cycles

	CMALU LUT	NTT top LUT	DSP	NTT/iNTT cycles	
	Mode-0 → Mode-1	**Mode-0 → Mode-1**		**Mode-0 = Mode-1**	**Mode-2**
ML-KEM (W=12)	244 → 362	473 → 596	0	29,197	54,285
ML-DSA (W=23)	473 → 697	745 → 991	0	44,557	≈1.86×
Falcon (W=14)	298 → 446	553 → 701	0	79,373	≈1.86×

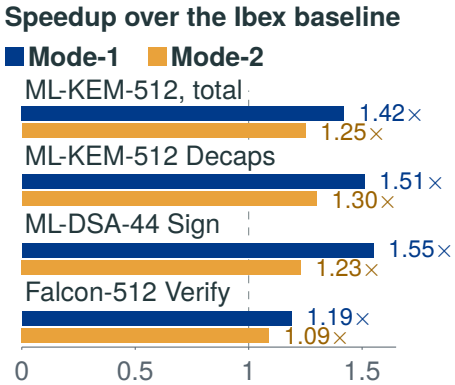
- **Mode-1** cycles equal **Mode-0**; 200 MHz in all modes
- **Mode-1**: CMALU is 60–70% of NTT-block LUTs
- 65 nm ASIC: 970 / 666 / 715 MHz

Energy per transform (ML-KEM, μJ)



Artix-7 (xc7a100t), post place-and-route. Energy: activity-based SAIF.
Mode-2 factor: 54,285 / 29,197 cycles (computed).

Ibex: faster and protected at the lowest eSlice+



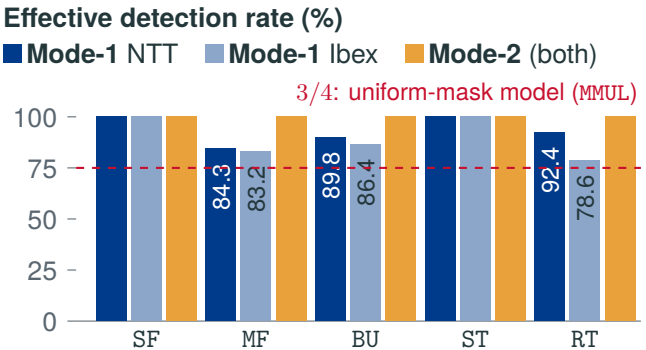
	Added area			Speedup		
	LUT	DSP	eSlice+	App.	NTT	FD
Celik et al.	6,933	0	1,733	1.76×	–	×
Gewehr et al.†	1,791	9	673	1.47×*	–	×
Bevin et al.	354	14	439	1.19×	1.69×	×
Mode-1	**564**	**0**	**141**	1.42×	**2.56**×	✓
Mode-2	651	0	163	1.25×	1.85×	✓

- Lowest eSlice+; only design with fault detection

Speedups from the paper or computed from its cycle counts. App.: ML-KEM-512 total (*encaps only).
NTT: ML-KEM forward NTT; each vs. its own SW baseline. †Our reproduction.
eSlice+ $= \max(\mathrm{LUT}/4 + 128\,\mathrm{BRAM} + 25\,\mathrm{DSP},\ \mathrm{FF}/8)$. Prior designs: [CYKO23, GLM24, BKIO25].

Fault injection: measured coverage in two systems

- Verilator wrapper; 10,000 trials per configuration
- Targets: `C`, `E`, `D`, `P`, `F`
- NTT: ML-KEM forward NTT
  Ibex: full ML-KEM-512 Decaps
- Rates among effective faults (architectural output changed)
- Fault-free baseline runs: no checker fired



SDC: **Mode-1** NTT 4.40%, Ibex 9.78%; **Mode-2** 0.00%

Attack 1: **Mode-2** 100%, SDC 0%

Attack 2: `SF`/`ST` 100% every run

Attack 3: `ST` on `F` 100%

Table 5 in the paper ($W = 12$). n_{eff} = 2,540–6,550 (NTT), 1,680–2,040 (Ibex); Wilson 95% CI $\leq \pm 2.0\%$.

Conclusion

IoT PQC needs small, fault-aware modular arithmetic.

- CMALU: six integer and modular operations on one DSP-free CSA/CLA datapath
- **Mode-1**: parity + Montgomery evenness, single-bit proof, 0 extra cycles; SF/ST 100% in FI
- **Mode-2**: recomputation on the same hardware; observed SDC 0%, $\approx 1.86\times$ NTT cycles
- Ibex: $1.42\times$ ML-KEM-512 with $+564$ LUT, 0 DSP; lowest eSlice+ among compared designs

RTL on GitHub

The Montgomery loop checks itself:
detection from the datapath, not from a duplicate.

Future work: non-linear checks for even-weight multi-bit masks; protection beyond the arithmetic core.

Thank you!

Questions?



github.com/whYBeKim/CMALU-for-PQC

YoungBeom Kim · darania@kookmin.ac.kr

Appendix

Appendix: vs. prior Montgomery fault detection

	Aghapour [AAKA25]	CMALU **Mode-1**	CMALU **Mode-2**
Technique	Partial recomp. + input encoding	Parity + Montgomery evenness	**Mode-1** + full recomp.
Target	2048-bit MP Montgomery	PQC `MMUL`/`MADD`/`MSUB`	same
Added area	+3.0–5.7k LUT, 16 DSP	+118 LUT, 0 DSP	+69 LUT, 0 DSP
Latency	+10–25%	**0%**	+86%
Coverage	94–100% (C model)	`SF`/`ST` 100%, `MF`/`BU`/`RT` 78–92%	all 100%, SDC 0%
Validation	C (GMP) simulation	RTL + processor	same

- Different widths and platforms: a distinct design point, not an apples-to-apples win
- Higher overhead in percent, but under 1/25 of the absolute LUT increase, with 0 DSP

Table 6 in the paper. **Mode-1** area: CMALU block vs. **Mode-0**; **Mode-2**: NTT-top increment vs. **Mode-1**.

Appendix: scope and limitations

- Arithmetic-core countermeasure:
  detects faults once an operand is inside the modular datapath
- Out of scope: checker/comparator faults, common-mode recomputation errors, memory I/O, Keccak, control flow, Fujisaki–Okamoto check bypass
- Even-weight masks with $\Delta[0] = 0$ evade both **Mode-1** checkers;
  Mode-2 closes this for effective one-pass transients
- Theorem 2 does not cover faults with $\epsilon_1 = \epsilon_2$ (e.g. permanent);
  stuck $\mathtt{F}$ still trips the invariant once some $\mathtt{F}_i \neq s$
- Twiddle constant corrupted before capture into $\mathtt{D}$: 0% detection in both modes;
  one parity bit per ROM entry ($+4$ LUT) restores 100% single-bit ROM detection
- Checker logic may leak through power, EM or timing; no side-channel claim

Sections 3.3 and 4.5 in the paper.

Appendix: **Mode-1** detection per attack (theory)

Attack	Model scope	Fault	**Mode-1** per run
1 (DFA)	`MMUL/MADD/MSUB`, checked word	`SF`	1 (Theorem 1)
1	`MMUL`, uniform nonzero mask	`MF/BU/RT`	$> 3/4$
1	`MADD/MSUB`, uniform nonzero mask	`MF/BU/RT`	$> 1/2$
2 (replay)	BSM-based paths	`SF/MF/BU/RT`	= Attack 1
3 (skip)	`MMUL`, correction bit stuck	`ST`	$1 - \Pr(\mathtt{F}_1 = \cdots = \mathtt{F}_m = s)$

- Replay: $\Pr[N \text{ undetected}] = \prod_{r=1}^{N}(1 - p_{\mathrm{det},r})$ for independent runs
- **Mode-0**: no detection.
  Mode-2: any effective one-pass transient (Theorem 2);
  permanent faults fall back to **Mode-1**

Table 1 in the paper; checker-observed word model.

Appendix: FPGA results (Artix-7, post place-and-route)

		LUT		FF		BRAM	Power (W)	
		M0	M1	M0	M1		M0	M1
NTT accelerator, 200 MHz								
ML-KEM	NTT top	473	596	337	341	1	0.106	0.109
	CMALU (W=12)	244	362	128	123			
ML-DSA	NTT top	745	991	502	525	2	0.134	0.143
	CMALU (W=23)	473	697	184	208			
Falcon	NTT top	553	701	374	378	1	0.119	0.128
	CMALU (W=14)	298	446	130	135			
Ibex RISC-V (RV32MSlow), 50 MHz								
Baseline	SoC / core	5,891 / 3,459		5,280 / 1,370		32	0.310	
+CMALU	SoC	6,252	6,473	5,498	5,530	32	0.308	0.310
	core (W=23)	3,815	4,023	1,588	1,620			

Table 2 in the paper; M0/M1 = **Mode-0**/**Mode-1**; power: Vivado vectorless.

Appendix: ASIC results

		Area (μm^2)		Freq	Power (mW)	
		M0	M1	(MHz)	M0	M1
NTT accelerator, 65 nm, Cadence Genus						
ML-KEM	NTT top (W=12)	164,435	171,673	970	90.7	103.9
ML-DSA	NTT top (W=23)	276,011	297,721	666	146.2	153.2
Falcon	NTT top (W=14)	367,854	369,469	715	158.2	169.0
Ibex core, Nangate45, Yosys/OpenSTA, W=23						
Baseline	Ibex core	15,864		250	19.13	
+CMALU	Ibex core	19,190	20,116	250	18.52	17.10

- **Mode-1** NTT: 30.10 / 66.90 / 111.01 μs, 3.12 / 10.24 / 18.76 μJ per transform
  Ibex core area: +21.0 / 26.8 / 28.6% (**Mode-0**/**Mode-1**/**Mode-2**)

Table 3 in the paper; M0/M1 = **Mode-0**/**Mode-1**. SRAMs are 90–95% of the NTT ASIC area.

Appendix: PQC cycles on Ibex (10^6 cycles)

ML-KEM		Base	**Mode-1**	**Mode-2**	ML-DSA		Base	**Mode-1**	**Mode-2**
512	KeyGen	2.049	1.577	1.744	44	KeyGen	6.452	5.452	5.940
	Encaps	2.457	1.729	1.956		Sign	20.644	13.362	16.756
	Decaps	3.119	2.070	2.406		Verify	7.099	5.622	6.340
768	KeyGen	3.425	2.593	2.887	65	KeyGen	11.156	9.638	10.383
	Encaps	4.023	2.862	3.230		Sign	29.960	19.632	24.492
	Decaps	4.906	3.322	3.839		Verify	11.702	9.512	10.582
1024	KeyGen	5.362	4.100	4.549	87	KeyGen	18.732	16.417	17.563
	Encaps	6.090	4.437	4.975		Sign	53.780	35.581	44.117
	Decaps	7.195	5.026	5.752		Verify	19.629	16.453	18.017
Falcon-512 Verify		1.289	1.083	1.182	Falcon-1024 Verify		2.561	2.082	2.328

Figure 6 in the paper. Median of 100 runs at 50 MHz, `-O3`; **Mode-0** cycles equal **Mode-1**.
Falcon: Verify only (NTT rare in KeyGen, unused in Sign).

Appendix: fault-injection setup

- Injection into `C`, `E`, `D`, `P`, `F` through a Verilator 4.228 C/C++ wrapper
- 10,000 trials per fault-model configuration; `SF`, `MF`, `BU`, `ST`, `RT`
- NTT accelerator ($W = 12$): CMALU during the CT butterfly of the forward NTT (`MMUL`, `MADD`, `MSUB`)
- Ibex: full ML-KEM-512 Decaps, where polynomial multiplication concentrates; KeyGen and Encaps are dominated by matrix sampling
- `C`, `E`, `F` are injected only during `MMUL` cycles: the `MMUL` bound dominates
- Effective = architectural output changed; rates among effective faults only
- n_{eff} = 2,540–6,550 (NTT), 1,680–2,040 (Ibex);
  Wilson 95% CI $\leq \pm 0.2\%$ (`SF`/`ST`), $\leq \pm 2.0\%$ (`MF`/`BU`/`RT`)

Section 6.3 in the paper.

Appendix: references

AAKA25 Aghapour et al. Partial recomputation fault detection for MP Montgomery mult. TCAD 2025.

AMI+22 Aikata et al. KaLi: a crystal for post-quantum security. TCAS-I 2022.

BKIO25 Bevin et al. Kyber and Dilithium via a single Montgomery reduction ISE. SOCC 2025.

BNV24 Bertels et al. A better Kyber butterfly for FPGAs. FPL 2024.

BSKA25 Bauer et al. A fault-resistant NTT by polynomial evaluation and interpolation. CASCADE 2025.

CYKO23 Celik et al. Implementation of CRYSTALS-Kyber using RISC-V processor. ICECS 2023.

GLM24 Gewehr et al. Hardware acceleration of Kyber with RISC-V ISEs. IEEE Access 2024.

HPP21 Hermelink et al. Fault-enabled chosen-ciphertext attacks on Kyber. INDOCRYPT 2021.

KKR+23 Khan et al. Efficient, error-resistant NTT architectures for Kyber. VLSI-SoC 2023.

MCL+23 Mao et al. Configurable SW/HW co-design of CRYSTALS-Dilithium. ACM TRETS 2023.

Mon85 Montgomery. Modular multiplication without trial division. Math. Comp. 1985.

RYB+23 Ravi et al. Fiddling the twiddle constants: fault injection on the NTT. TCHES 2023.

WZCG22 Wang et al. Efficient implementation of Dilithium on FPGA SoC. TVLSI 2022.