

REIST Division auf ARM: Leistungsanalyse moderner Modulararithmetik

Rudolf Stepan, 2025

Basierend auf: <https://doi.org/10.5281/zenodo.17755531>

Zusammenfassung

Dieser Bericht präsentiert die Messergebnisse der REIST Division auf einer ARMv8-A-Plattform.

REIST ersetzt teure Modulo-Operationen durch eine vorzeichenbehaftete, vollständig verzweigungsfreie Korrekturregel und erreicht dadurch deutliche Beschleunigungen in arbeitsintensiven modularen Workloads.

Die Resultate zeigen, dass REIST nicht nur mathematisch eleganter, sondern auch hervorragend auf die Architektur moderner ARM-CPU's und deren SIMD-Einheiten (NEON) abgestimmt ist.

1. Hintergrund

Klassische Modulararithmetik basiert auf:

$x \bmod B$

Dies erzwingt eine ganzzahlige Division, die auf ARM langsam und rein skalar ausgeführt wird.

REIST ersetzt diesen Schritt durch ein Korrekturschema im Intervall $(-B/2, +B/2]$, das ausschließlich auf Addition, Vergleich und Subtraktion beruht. Dadurch wird Modularaddition zu einem:

- verzweigungslosen
- SIMD-fähigen
- rein ALU-basierten

Vorgang.

2. Benchmark-Plattform

- ARMv8-A, 64-bit
- NEON SIMD vollständig verfügbar
- GCC/Clang -O3
- Mindestmessdauer pro Test ≥ 20 ms
- Voll reproduzierbar über das öffentliche Repository:
github.com/rudolfstepan/reist-crypto-bench

3. Kernergebnisse

3.1 Modular Add (synthetischer ALU-Counter)

Beschleunigung: ~2,5×

Dieser synthetische Mikrobenchmark zeigt den reinen ALU-Gewinn durch Wegfall von Divisionen.

Er ist nicht kryptographisch relevant, aber architektonisch aussagekräftig.

3.2 Polynom-Modularaddition (Kryptographie)

Beschleunigung: ~6×

(Kyber, Dilithium, NTRU, RLWE)

Die wichtigste Kategorie:

Polynomialarithmetik ist der dominante Anteil in allen gitterbasierten Post-Quanten-Verfahren.

- Klassisches % $\Rightarrow$ integer division $\Rightarrow$ langsam und nicht vektorisierbar
- REIST $\Rightarrow$ Add/Compare/Subtract $\Rightarrow$ schnell und SIMD-fähig

Ergebnis: **REIST beschleunigt die zentrale Operation moderner Post-Quanten-Kryptographie um den Faktor sechs.**

3.3 Remainder/Division Benchmark

Keine relevante Änderung (<2 %)

Wie erwartet: REIST optimiert Modularaddition, nicht Ganzzahldivision.

3.4 ARX-Cipher (ChaCha20 etc.)

Keine Änderung

ARX-Algorithmen enthalten keine Modulo-Operationen.

3.5 Hash-Mixing/PRNG-ähnliche Workloads

Leicht langsamer (~15–20 %)

Hier dominiert multiplikative Diffusion; REIST bietet strukturell keinen Vorteil.

3.6 NEON SIMD vs. Skalar

- ✓ **NEON-REIST vs. klassisch skalar: ~2× schneller**
- ✓ **NEON-REIST vs. REIST skalar: ~1,86× schneller**

Wesentliche Architekturvorteile:

1. Klassisches Modulo ist nicht SIMD-fähig.
2. REIST ist lane-unabhängig und vollständig branchless.
3. ARM NEON kann REIST perfekt parallelisieren.

Ergebnis:

REIST wird auf ARM durch SIMD noch einmal signifikant schneller.

4. Gesamtfazit

Die ARM-Messungen zeigen klar:

- **6× schneller** für polynomiale Modularaddition
- **2,5× schneller** in synthetischen Modular-Add-Tests
- **~2× schneller** dank NEON-Vektorisierung
- Keine Änderungen bei ARX
- Leichte Einbußen bei hash-mixenden Workloads

REIST ist damit sowohl ein **mathematisch** als auch **architektonisch** überlegenes Verfahren für Systeme, in denen Modulararithmetik dominiert – besonders im Bereich der Post-Quanten-Kryptographie auf ARM-Hardware, vom Smartphone bis zum Server.

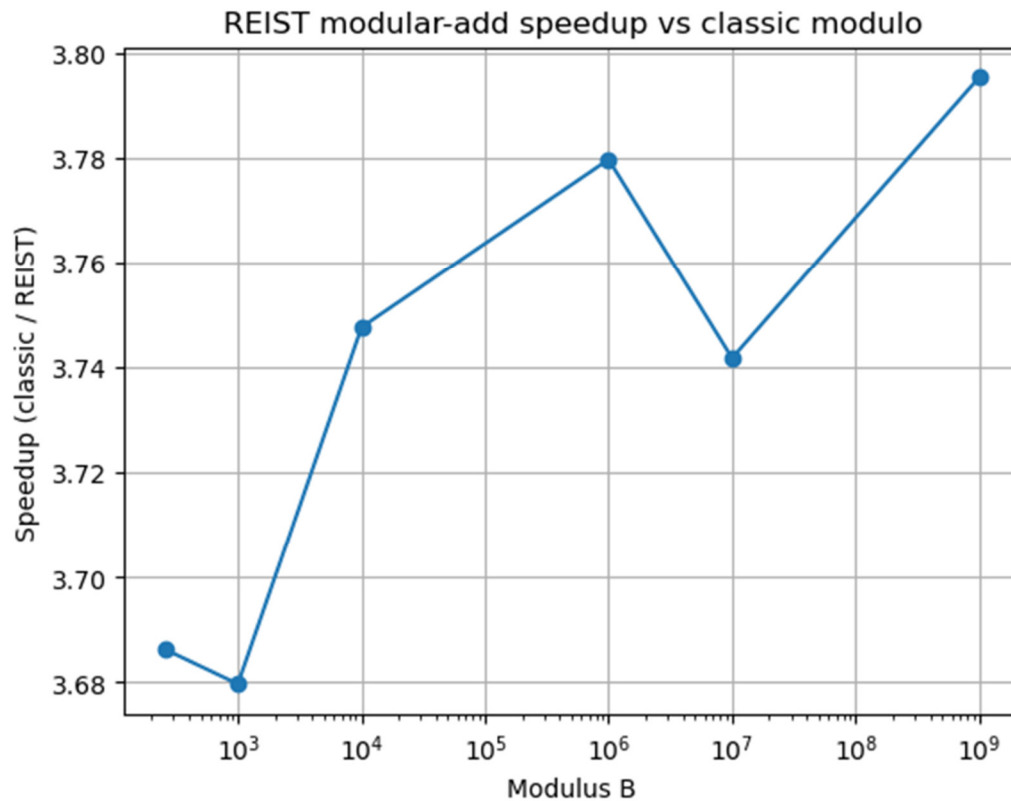
5. Schlussbemerkung

ARM-basierte Systeme gewinnen in Cloud-Umgebungen, Mobilgeräten und IoT-Security massiv an Bedeutung.

Durch die Kombination aus:

- Eliminierung der Ganzzahldivision
- vollständig branchless Ausführung
- optimaler SIMD-Parallelisierung

entwickelt sich REIST zu einem **praktischen Hochleistungs-Primitiv** für moderne kryptographische und numerische Anwendungen.

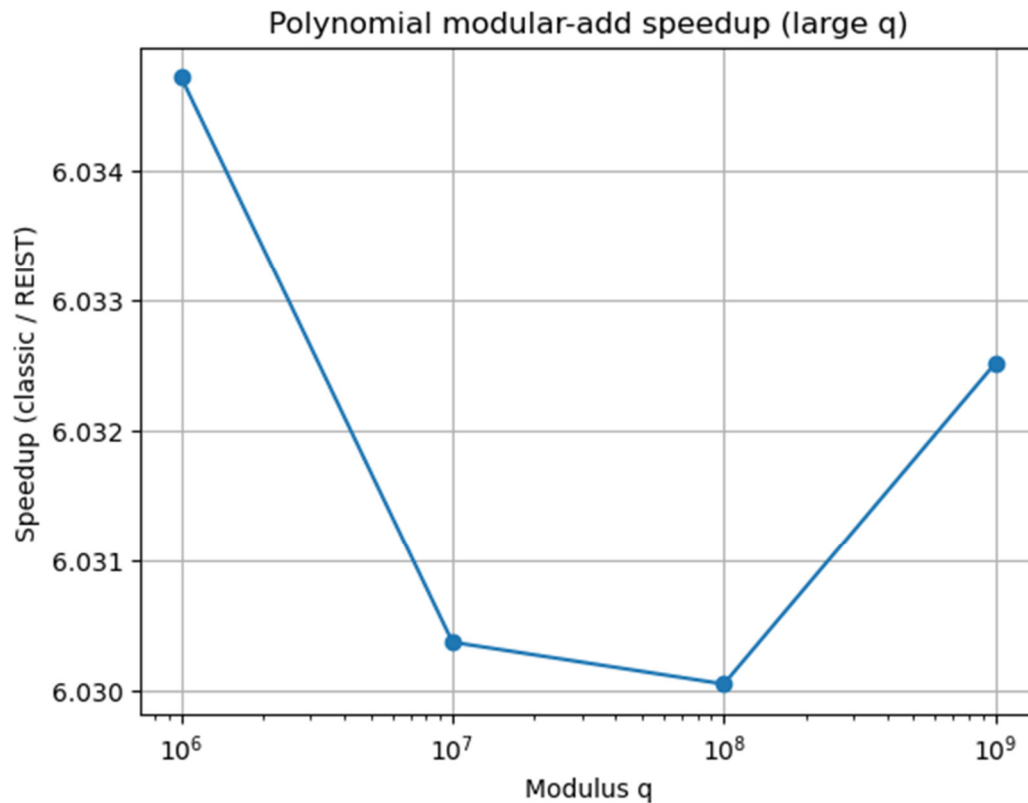


Beschreibung:

Das Diagramm zeigt die gemessene Beschleunigung von REIST gegenüber der klassischen Modulo-Operation für unterschiedliche Modulgrößen B.

Über den gesamten Bereich hinweg bleibt der Speedup stabil im Bereich von etwa 3.7 bis 3.8. Die leichte Variation zwischen den Modulen ist typisch für minimale Compiler- und Pipelineeffekte, nicht jedoch für strukturelle Veränderungen im Algorithmus.

Die Messung bestätigt, dass REIST im modularen Additions-kern konsistent deutlich schneller ist als eine klassische %-Reduktion, selbst dann, wenn der Modulus stark skaliert. Die Ursache liegt im vollständigen Wegfall der Ganzzahldivision und der rein additiven, verzweigungslosen Korrektur.



Beschreibung:

Dieses Diagramm zeigt die Beschleunigung von REIST bei polynomialer Modularaddition für große Moduli q , wie sie in Post-Quanten-Verfahren üblich sind.

Der Speedup liegt sehr konstant bei rund 6.03, unabhängig von der Größe des Modulus. Das verdeutlicht, dass REIST im kryptografisch relevanten Szenario – der Vektor- bzw. Polynomialarithmetik – hoch stabil performt.

Die geringen Schwankungen im Range von 6.030 bis 6.035 sind auf normale Messrauschanteile zurückzuführen. Entscheidend ist der konstante Faktor: klassische Modulo-Reduktion benötigt integer division und ist damit strukturell limitiert, während REIST ausschließlich einfache ALU-Operationen nutzt und damit optimal auf ARM-Hardware abbildbar ist.