CTIDH on Cortex-M4 with FPU

- Goal: Implement $\mathbb{F}_p$ arithemtic using floating point registers to reduce memory accesses
  - understand Montgomery reduction[1]
  - understand available x86 implementation[2]
  - port x86-assembly code generating scripts[3] to Arm Cortex-M4
  - improve current code using floating point registers on the M4 (or other tricks you can find)

[1] `https://cacr.uwaterloo.ca/hac/about/chap14.pdf`
[2] e.g. `http://ctidh.isogeny.org/high-ctidh-20210523/fp512.S.html`
[3] `http://ctidh.isogeny.org/high-ctidh-20210523/autogen.html`

Toom-Cook polynomial multiplication in velusqrt isogeny formulas

- Goal: Implement velusqrt isogeny formulas using Toom-Cook instead of Karatsuba
  - understand velusqrt isogenies[4]
  - understand available C implementation using Karatsuba[5]
  - Replace Karatsuba by Toom-Cook
  - Compare performance for various isogeny degrees

[4] http://velusqrt.isogeny.org/velusqrt-20200616.pdf
[5] http://velusqrt.isogeny.org/software.html

Primes for B-SIDH

- Goal: Search for special B-SIDH primes
  - understand smoothness requirement of $p-1$ and $p+1$ in B-SIDH [6]
  - check methods for finding B-SIDH primes[7]
  - implement search for special B-SIDH primes, e.g., unbalanced primes such that one of $p-1$ and $p+1$ has a *very* small smoothness bound

[6] https://eprint.iacr.org/2019/1145.pdf

[7] e.g. https://eprint.iacr.org/2019/1145.pdf or https://eprint.iacr.org/2020/1283.pdf