Progress of Scalar Crypto Support on LLVM

Mainly based on spec v0.8.1 (partly v0.9.0)

Presenter: Xinlong Wu

LLVM MC layer

- Instructions: Fully defined in Scalar Crypto & shared with B extension (bit manipulation)
- All feature sets in Scalar Crypto Specification:
  - Fully defined in Scalar Crypto
    - K, Zkn, Zks, Zkr, Zkne, Zknd, Zknh, Zksed (Zkse&Zksd in 0,8.1), Zksh
  - Shared with B extension:
    - Zkg, Zkb
- What we have done:
  - Almost all Scalar Crypto instructions can be encoded
  - Submitted revision (D98136, request for comments)
- Future work :

[Scalar Crypto Specification](https://github.com/riscv/riscv-crypto/releases)

LLVM MC layer

Instructions	Functional Set	Feature Sets		
		Zkn (RV32)	Zkn (RV64)	Zks (RV32)
aes32dsi	Zknd	✔		
aes32dsmi	Zknd	✔		
aes32esi	Zkne	✔		
aes32esmi	Zkne	✔		
aes64ds	Zknd		✔	
aes64dsm	Zknd		✔	
aes64es	Zkne		✔	
aes64esm	Zkne		✔	
aes64im	Zknd		✔	
aes64ks1i	Zkne		✔	
aes64ks2	Zkne		✔	
sha256sig0	Zknh	✔	✔	
sha256sig1	Zknh	✔	✔	
sha256sum0	Zknh	✔	✔	
sha256sum1	Zknh	✔	✔	
sha512sig0h	Zknh	✔		
sha512sig0l	Zknh	✔		
sha512sig1h	Zknh	✔		
sha512sig1l	Zknh	✔		
sha512sum0r	Zknh	✔		
sha512sum1r	Zknh	✔		
sha512sig0	Zknh		✔	
sha512sig1	Zknh		✔	
sha512sum0	Zknh		✔	
sha512sum1	Zknh		✔	
sm3p0	Zksh			✔
sm3p1	Zksh			✔
sm4ed	Zkse			✔
sm4ks	Zkse			✔
pollentropy	Zkr			
getnoise	Zkr			
clmul, clmulh	Zkg	✔	✔	✔
xperm.n, xperm.b	Zkb	✔	✔	✔
ror, rol, rori	Zkb	✔	✔	✔
roriw	Zkb		✔	
andn, orn, xorn	Zkb	✔	✔	✔
pack, packu, packh	Zkb	✔	✔	✔
packw, packuw	Zkb		✔	
rev.b, rev8 (grevi)	Zkb	✔	✔	✔
rev8.w (grevi)	Zkb		✔	
gorci	Zkb	✔	✔	✔
zip (shfl)	Zkb	✔	✔	✔
unzip (unshfl)	Zkb	✔	✔	✔

Table 2: Feature sets for instructions in the scalar cryptography extension.

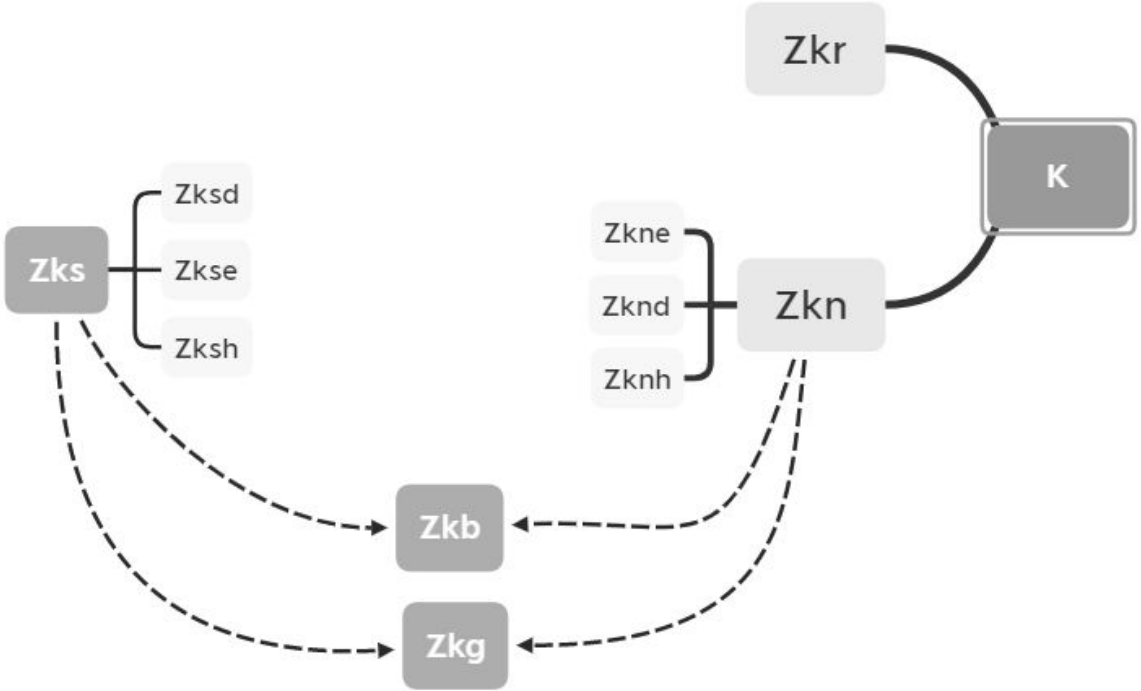
[Scalar Crypto Specification](https://github.com/riscv/riscv-crypto/releases)

LLVM MC layer

Functional Set	Description
`K`	The default scalar cryptography extension, short for `ZknZkr`
`Zkg`	Constant time carry-less multiply for Galois/Counter Mode.
`Zkb`	Bitmanip subset included in the scalar cryptography extension, minus those in `Zkg`.
`Zkr`	Entropy source for seeding random number generators.
`Zkn`	NIST algorithm suite. Short for `ZkneZkndZknhZkgZkb`.
`Zkne`	NIST AES Encryption Instructions.
`Zknd`	NIST AES Decryption Instructions.
`Zknh`	NIST SHA2 Hash function instructions.
`Zks`	ShangMi (SM) algorithm suite. Short for `ZksedZkshZkgZkb`.
`Zksed`	SM4 Instructions.
`Zksh`	SM3 Hash function instructions.

Table 1: Explanation of the feature strings used to refer to the functional sets.

LLVM MC layer



Scalar Crypto Intrinsics

- Scalar Crypto intrinsics document from Markku
- Further work:
  - Try to implement Scalar Crypto intrinsics

[Intrinsics proposal from Markku](https://github.com/rvkrypto/rvkrypto-fips/blob/main/rvkintrin.h)

Thanks for Listening

Related resources

- [Scalar Crypto Specification](https://github.com/riscv/riscv-crypto/releases)
- [Intrinsics proposal from Markku](https://github.com/rvkrypto/rvkrypto-fips/blob/main/rvkintrin.h)
- [Revision: Initially support the K-extension instructions](https://reviews.llvm.org/D98136)
- [PLCT Repository of LLVM (RISCV K-ext)] (https://github.com/isrc-cas/rvv-llvm/tree/riscv-k-extension)
- [wiki page of Crypto extension] (https://wiki.riscv.org/display/TECH/Scalar+Crypto+Standardization+Status+Summary)