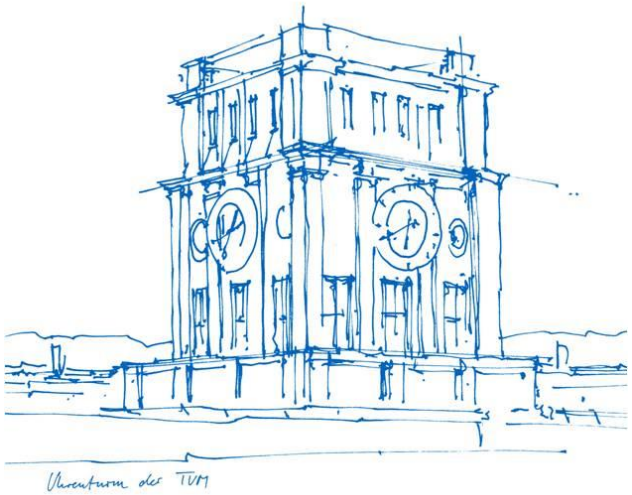
A Detailed Report on the Overhead of Hardware APIs for Lightweight Cryptography

Patrick Karl and Michael Tempelmeier

Technical University of Munich
Department of Electrical and Computer Engineering
Chair of Security in Information Technology

October 16, 2020



Exemplary API compliant implementations

API	Design	LUT	FF
CAESAR	Ascon128 [1]	1595	818
	SpoC-64 [2]	2136	876
LWC	Ascon128 [2]	1802	539
	SpoC-64 [2]	1565	728
	Gimli [3]	946	235

Exemplary API compliant implementations

API	Design	LUT	FF
CAESAR	Ascon128 [1]	1595	818
	SpoC-64 [2]	2136	876
LWC	Ascon128 [2]	1802	539
	SpoC-64 [2]	1565	728
	Gimli [3]	946	235

- What do absolute numbers tell us?

Exemplary API compliant implementations

API	Design	LUT	FF
CAESAR	Ascon128 [1]	1595	818
	SpoC-64 [2]	2136	876
LWC	Ascon128 [2]	1802	539
	SpoC-64 [2]	1565	728
	Gimli [3]	946	235

- What do absolute numbers tell us?
- Common API → fair comparison?

Exemplary API compliant implementations

API	Design	LUT	FF
CAESAR	Ascon128 [1]	1595	818
	SpoC-64 [2]	2136	876
LWC	Ascon128 [2]	1802	539
	SpoC-64 [2]	1565	728
	Gimli [3]	946	235

- What do absolute numbers tell us?
- Common API → fair comparison?
- What about different API implementations?

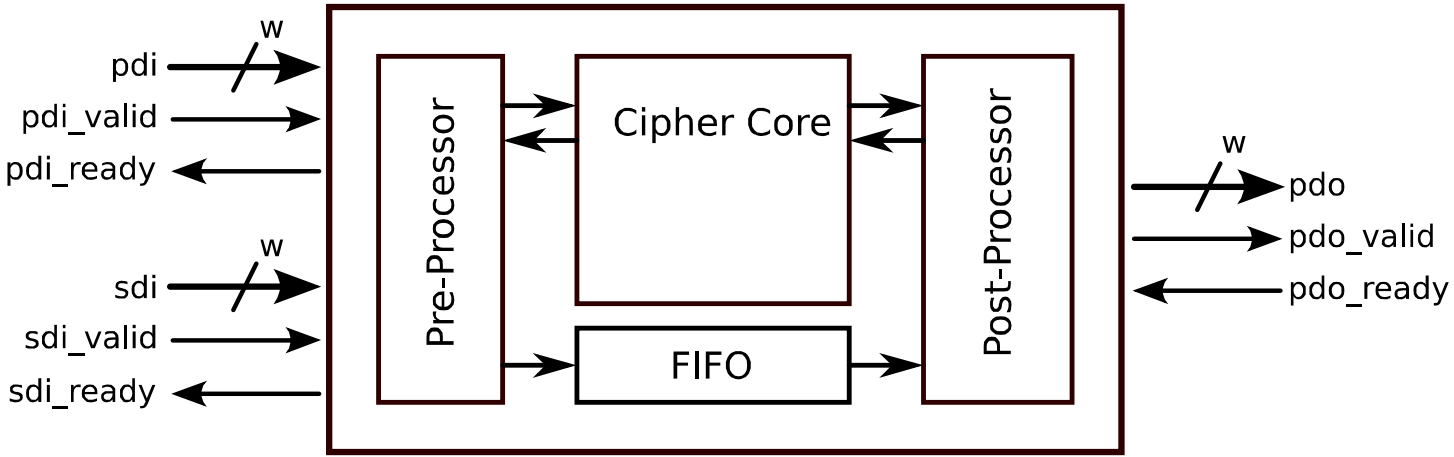
Why?

- "FPGA Benchmarking of Round 2 Candidates..." by GMU [4]

- 24 submissions:
  - 13 using unmodified dev. Package
  - 8 using modified dev. Package
  - 3 not using dev. Package

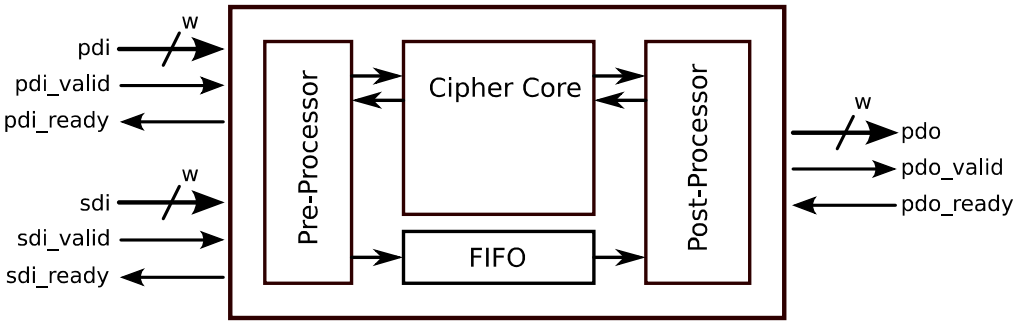
→ What does that mean for comparison?

API compliant Development Packages



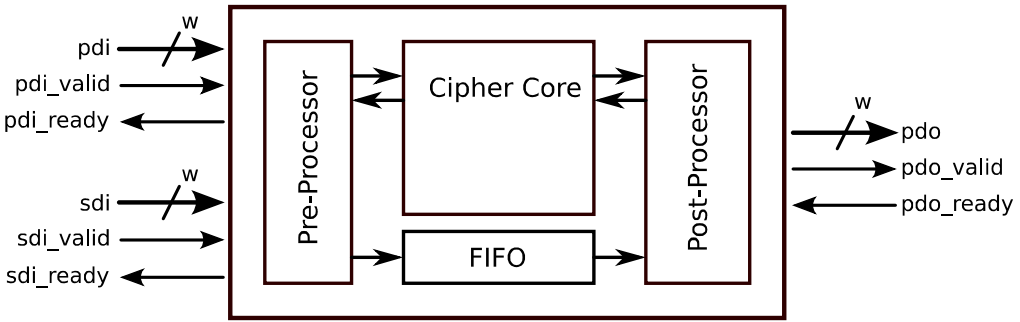
API compliant Development Packages

- LWC exclusive features:
  1. Hash support
  2. Extended width conversion
  3. Multi-Segment messages



API compliant Development Packages

- LWC exclusive features:
  1. Hash support
  2. Extended width conversion
  3. Multi-Segment messages

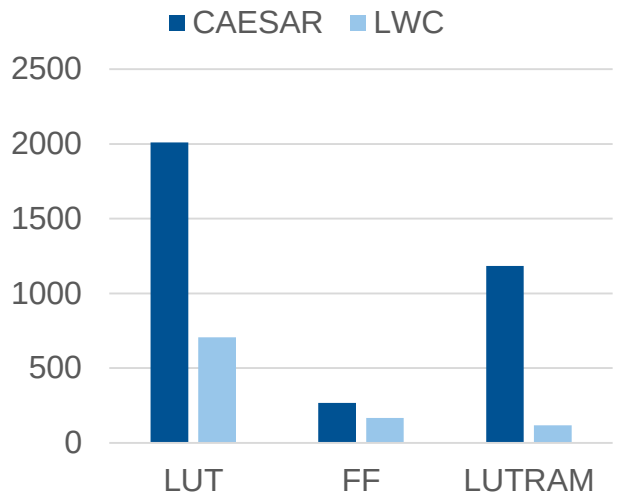


→ Benchmark: common interface

→ Pre-/PostProcessor, FIFO included!

Resource Comparison of CAESAR and LWC

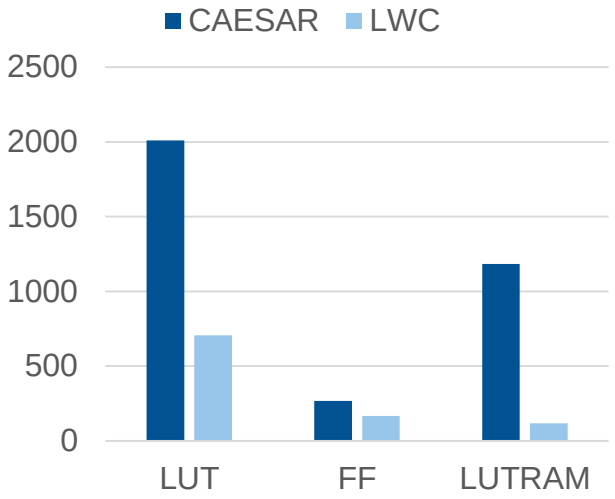
- LWC outperforms CAESAR



E.g. 32-bit designs with default configurations

Resource Comparison of CAESAR and LWC

- LWC outperforms CAESAR
- Exception: 8-bit design with minimized FIFO
- Feature cost constant (e.g. hash, multi-segment)



E.g. 32-bit designs with default configurations

CAESAR FIFO configuration

- Optional tag buffering
- E.g. 32-bit implementation, 128-bit tag → 8 entries sufficient

CAESAR FIFO configuration

- Optional tag buffering
- E.g. 32-bit implementation, 128-bit tag → 8 entries sufficient
- Default: 1024 entries

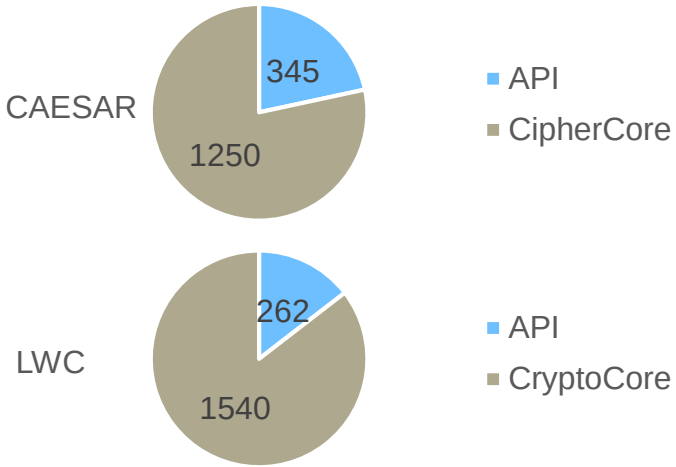
Exemplary API compliant implementations

API	Design	LUT	FF
CAESAR	Ascon128 [1] [1]	1595	818
	SpoC-64 [2] [2]	2136	876
LWC	Ascon128 [2]	1802	539
	SpoC-64 [2]	1565	728
	Gimli [3]	946	235

[1] HeaderFifo: 4 x 24-bit

[2] HeaderFifo: 512 x 32-bit

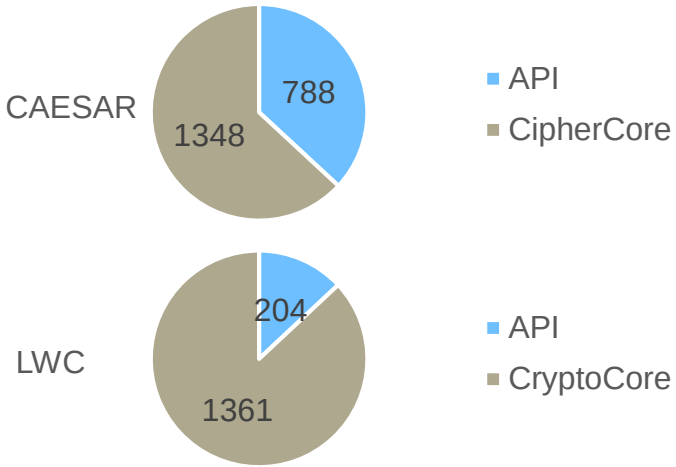
CAESAR Ascon128 [1] / LWC Ascon128 [2]



- CryptoCore by different designers

→ Ideally: multiple designs per cipher

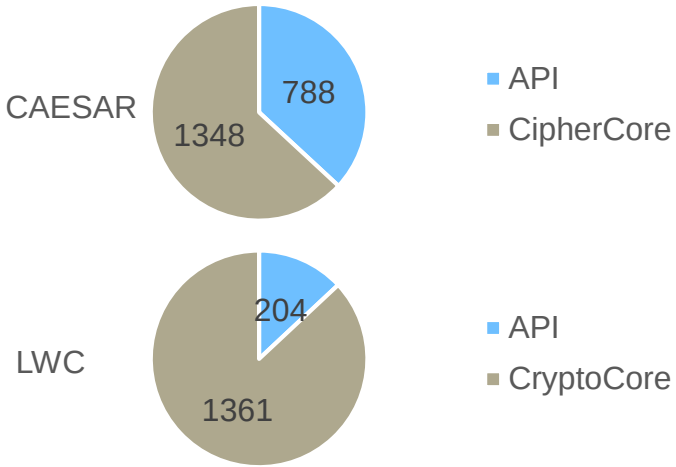
CAESAR SpoC-64 [2] / LWC SpoC-64 [2]



- FIFO: 512 vs. 4 entries

→ FIFO dominates

CAESAR SpoC-64 [2] / LWC SpoC-64 [2]

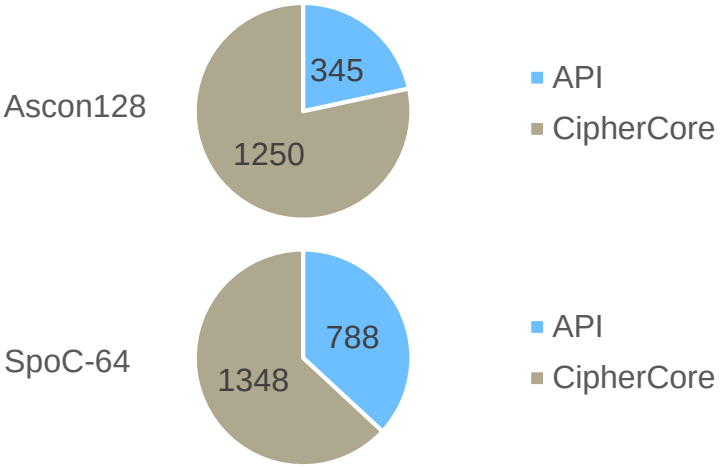


- FIFO: 512 vs. 4 entries

→ FIFO dominates

→ Removing FIFO? Critical Path!

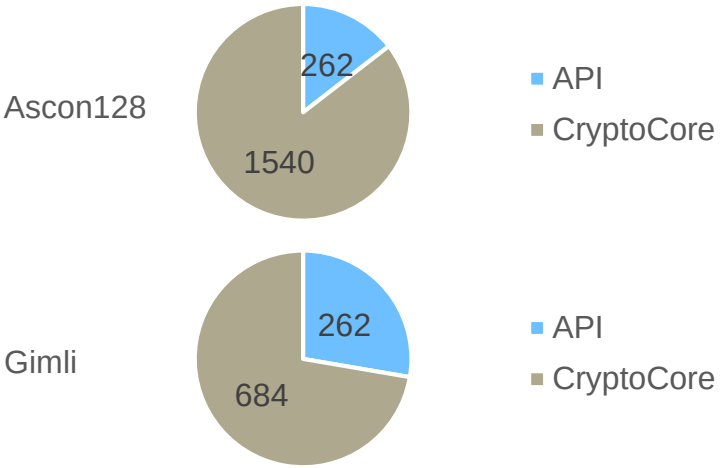
CAESAR Ascon128 [1] / CAESAR SpoC-64 [2]



- CipherCore difference not that huge

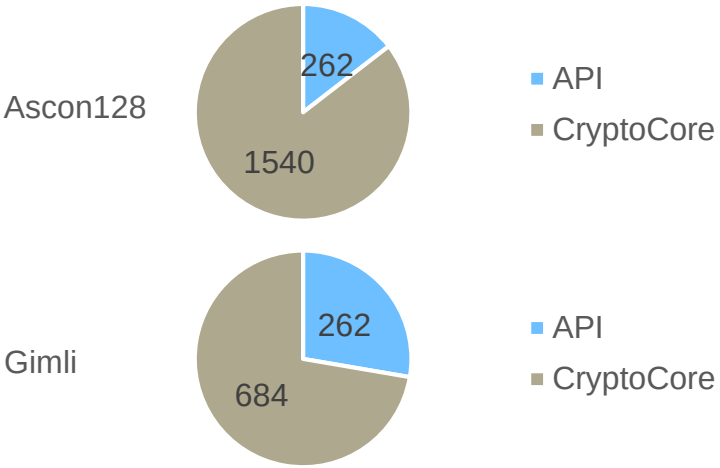
→ FIFO difference (API package)

LWC Ascon128 [2] / LWC Gimli [3]



- Equal assumptions for API implementation

LWC Ascon128 [2] / LWC Gimli [3]



- Equal assumptions for API implementation

→ Improved API implementation + config

→ Fair comparison possible

Conclusion

- Absolute numbers can lead to false impressions
  - ➢ Improvement of LWC over CAESAR package

Conclusion

- Absolute numbers can lead to false impressions
  - ➢ Improvement of LWC over CAESAR package
- Compare ciphers only
  - ➢ Benchmark int. interface, i.e. CryptoCore?

Conclusion

- Absolute numbers can lead to false impressions
  - Improvement of LWC over CAESAR package
- Compare ciphers only
  - Benchmark int. interface, i.e. CryptoCore?
- Ciphers require API
  - Include API implementation?

Thank you for your attention!

References

[1] Accessed:15.1.2020. Institute of Applied Information Processing and Communications (IAIK), Graz University of Technology.
URL: https://github.com/IAIK/ascon_hardware/tree/master/caesar_hardware_api_v_1_0_3/ASCON_ASCON

[2] Accessed: 13.1.2020. Signatures Analysis Laboratory, Virginia Tech.
URL: https://github.com/vtsal?tab=repositories

[3] Accessed: 13.1.2020. Chair of Security in Information Technology, Technical University of Munich.
URL: https://gitlab.lrz.de/tueisec/crypto-implementations/tree/master/LWC/GIMLI

[4] K. Mohajerani et al. FPGA Benchmarking of Round 2 Candidates in the NIST Lightweight Cryptography Standardization Process: Methodology, Metrics, Tools, and Results. Cryptology ePrint Archive, Report 2020/1207. https://eprint.iacr.org/2020/1207, 2020.