TECHNISCHE UNIVERSITÄT WIEN
Vienna | Austria

Institut für Computertechnik
Institute of Computer Technology

Integration of an Encryption Accelerator into an Open-Source Low-Power Microcontroller

384.178 SoC Design Lab

Severin Jäger

March 7, 2022

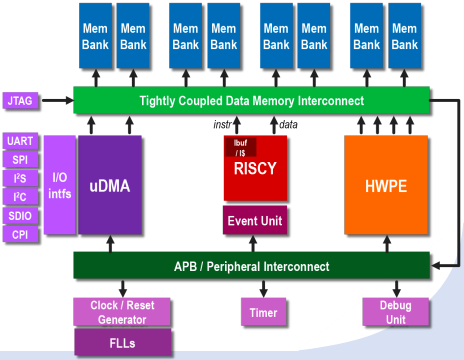
Motivation

- Increasing demand for near-sensor processing (signal processing, simple machine learning)
- Highly energy-constrained edge devices
- Privacy crucial for certain applications (e.g. biomedical devices, surveillance, home automation)
- Open source as gold standard for security - why not in hardware?

Goal: Provide energy-efficient cryptographic support for an open-source low-power microcontrollers

System Overview

- PULP Project (ETHZ, Unibo) develops open-source hardware based on RISC-V
- PULPissimo [1]: Single-core microcontroller with full set of peripherals
- Open ISA, open RTL, (mostly) open tools
- Optimised for minimal energy per operation in signal processing applications (433 MOPS/mW) [2]
- Hardware processing element (HWPW) allows accelerator integration



Goal

Add an efficient cryptographic accelerator as HWPE to PULPissimo

- Efficient in terms of area, energy
- Cryptographic capability: AES encryption (128 bits)
  - Well-established
  - Mature open-source hardware implementations available
  - Only encryption required for sensor nodes

Open-Source AES Cores

Core	`tiny_aes` [3]	`aes_128` [4]	`secworks_aes` [5]
Cycles/Op	1	12	4
Latency (cylces)	21	12	14
LUTs	4588	487	3327
Registers	4474	402	2990
BRAM Tiles	68	5	0
Max. Frequency [MHz]	375.9	180.5	124.8
Decryption	no	no	yes
AES256	no	no	yes

Only Verilog IP considered for easier integration. Resource consumption and clock frequency on Nexys4DDR with Vivado 2020.2

Challenges

- PULP documentation nice, but far from complete
  - Dozens of repositories with several dependencies
- Not all required tools are freely available (Mentor)
- Large design: NexysDDR4 85 % utilised with bare PULPissimo
  - Place & Route takes ages

Roadmap

Next steps

- Set up simulation framework
- Provide software reference implementation of AES
- Integrate AES core (`aes_128`) into HWPE template
- Evaluate HW & SW implementation on PULPissimo

Possible extensions

- Implement other crypto core (e.g. lightweight algorithm)
- Perform ASIC design flow (at least for crypto core) to assess power consumption

References

[1] PULPissimo repository on GitHub `https://github.com/pulp-platform/pulpissimo`

[2] P. D. Schiavone, D. Rossi, A. Pullini, A. Di Mauro, F. Conti and L. Benini, *Quentin: an Ultra-Low-Power PULPissimo SoC in 22nm FDX*, 2018 IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S), 2018.

[3] OpenCores `tiny_aes`, `https://opencores.org/projects/tiny_aes`

[4] `aes_128` repository on GitHub `https://github.com/www-asics-ws/aes_128`

[5] `secworks/aes` repository on GitHub `https://github.com/secworks/aes`