

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

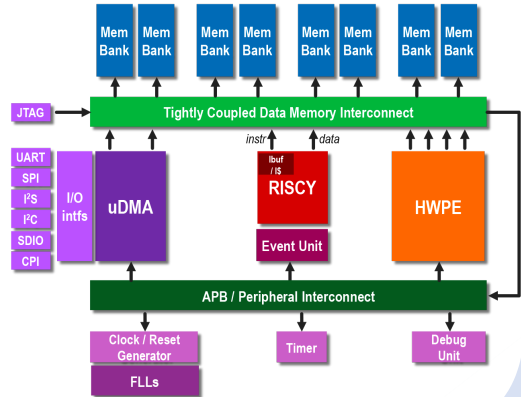
384.178 SoC Design Lab

Severin Jäger

11.01.2022

Recap I: Setup

- PULPissimo SoC [1]: Single-core microcontroller with full set of peripherals
- Open hardware (ISA, RTL)
- Focused on energy efficiency, not performance
- Hardware processing engine (HWPE) allows accelerator integration



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 Software Implementation

- Software reference implementation to assess gains
- `tiny-AES-c` available open source [2]
- Lightweight and portable C implementation
- Code size 2.4 kiB (compiled for RISC-V)
- Extremely simple API (`AES_init_ctx()`, `AES_ECB_encrypt()` sufficient)



Open-Source AES Cores

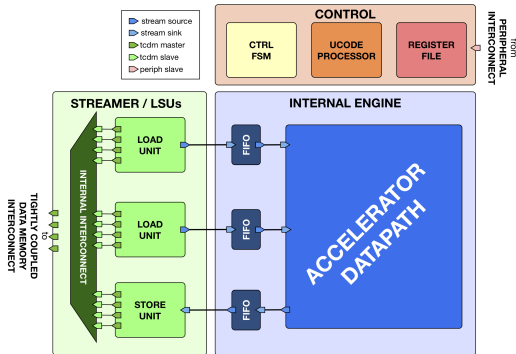
Core	tiny_aes [3]	aes_128 [4]	secworks_aes [5]
Cycles/Op	1	12	4
Latency (cycles)	21	12	14
LUTs	4588	487	3327
Registers	4474	402	2990
BRAM Tiles	68	5	0
Max. Frequency [MHz]	375.9	180.5	124.8
Decryption	no	no	yes
AES256	no	no	yes

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



Pulp Hardware Processing Engine (HWPE)

- Data exchange via shared L2 memory (no DMA or the like required)
- Control flow defined via peripheral interconnect (memory-mapped)
- Pointers and parameters exchanged, then autonomous operation
- Can be integrated into PULPissimo as well as larger clusters in the PULP ecosystem
- Template available open source



Challenges

- PULP documentation nice, but far from complete
 - Dozens of repositories with countless dependencies
 - Hardly any support available besides official documentation available
- Questasim on ICT EDA server, other tools on my machine
 - Great hassle to get scripts working
- Large design: NexysDDR4 85 % utilised with bare PULPissimo
 - Place & Route takes ages
- NexysDDR4 JTAG unit not supported \implies FPGA demo only possible with external JTAG programmer (and several software workarounds)



Next steps

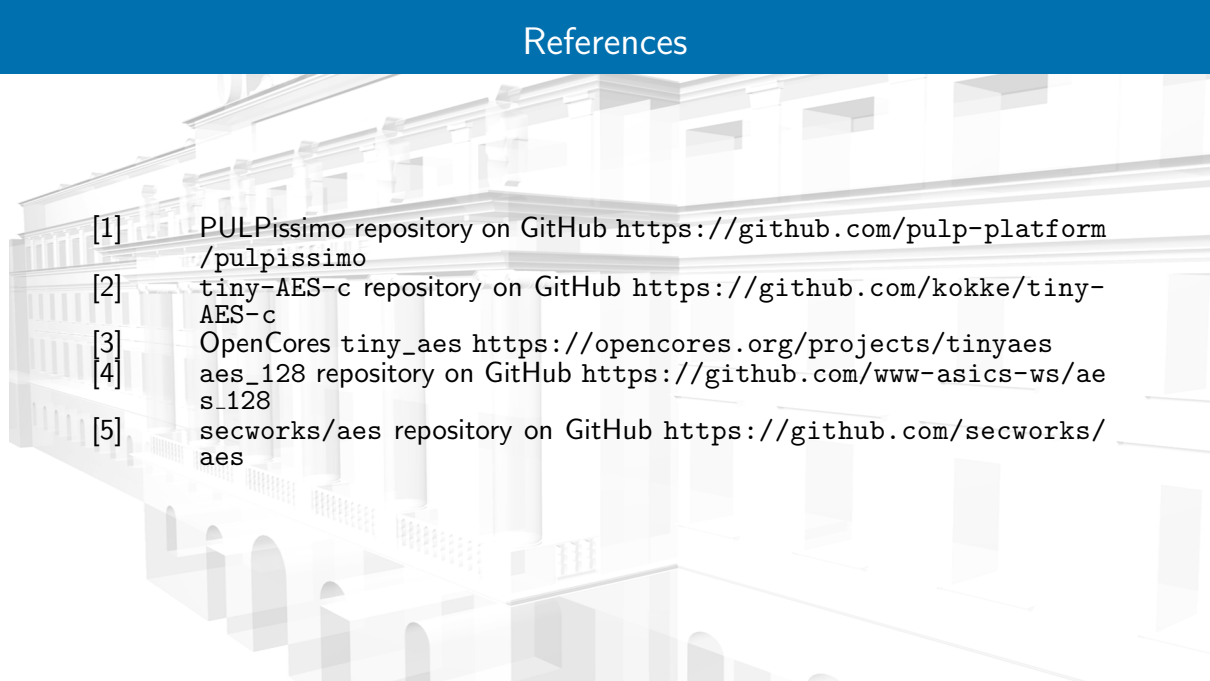
- Run compiled code on PULPissimo
 - Fix remaining tooling problems
- Integrate AES core (at least aes_128) into HWPE template
- Create demo application
- Compare HW & SW implementation on PULPissimo

Possible extensions (for consecutive project)

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



References

- 
- [1] PULPissimo repository on GitHub <https://github.com/pulp-platform/pulplissimo>
 - [2] tiny-AES-c repository on GitHub <https://github.com/kokke/tiny-AES-c>
 - [3] OpenCores tiny_aes <https://opencores.org/projects/tinyaes>
 - [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>