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 8, 2022

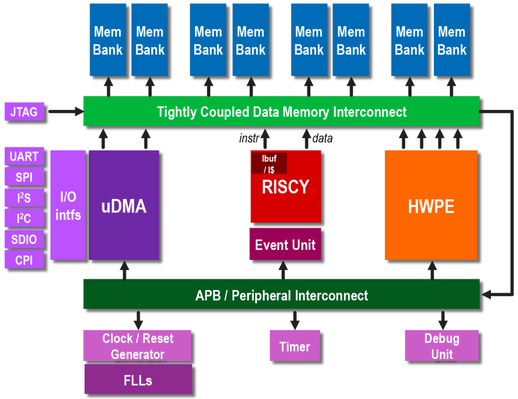
Motivation

- 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

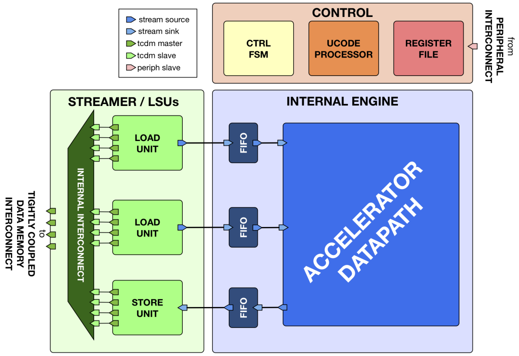
Recap I: Setup

- 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



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



Project Goal

- Implement an AES HWPE
  - Encryption only
  - 128 bit keys
  - Only straightforward electronic code book (ECB) mode
    - This is not state of the art!
- Compare to reference software

What can be gained from implementing AES in hardware?

Reference Software Implementation

- Software reference implementation to assess gains
- `tiny-AES-c` available open source [2]
- Lightweight and portable C implementation
- Code size 2.1 kB (compiled for PULPissimo)
- Extremely simple API (`AES_init_ctx()`, `AES_ECB_encrypt()` sufficient)
- Simple demo application encrypting N words & checking results

Verilog AES IP cores

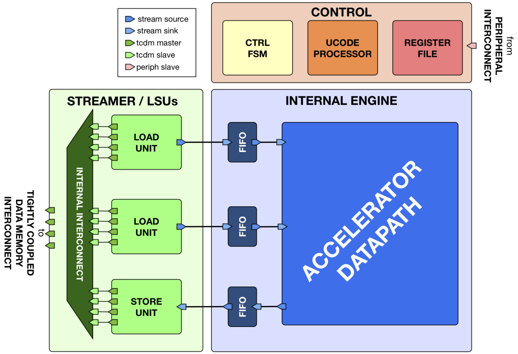
Core	tiny_aes	aes_128 [3]	secworks_aes
Cycles/Op	1	12	4
Latency (cylces)	21	12	14
LUTs	4588	487	3327
Registers	4474	402	2990
BRAM Tiles	68	5	0
Max. Frequency [MHz]	375.9	180.5	124.8
Decryption	no	no	yes
AES 256	no	no	yes

Resource consumption and clock frequency on Nexys4DDR with Vivado 2020.2

aes_128 selected due to minimal resource consumption and sufficient performance

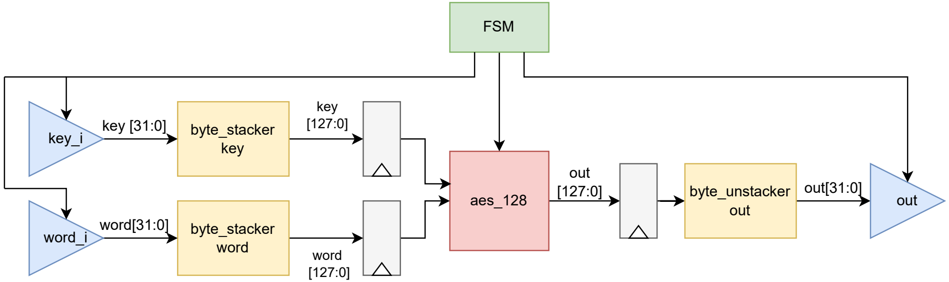
HWPE Operation

- Input data written to shared memory by CPU
- Control registers set by CPU (from software)
  - Source and destination pointers
  - Number of bytes to be loaded
  - Start operation
- HWPE FSM controls data ports and data path (interleaving):
  - Stream of input words fetched from shared memory
  - Data path computation (AES)
  - Stream of output words written to shared memory



HWPE Datapath

- Based on HWPE MAC example
- 32 bit wide memory interfaces
- (Un-)stackers for 128 bit AES words
- Pipelined operation
- Ready/valid handshake between all stages (forward & backward dependencies)



HWPE Software

- Utilising device driver from HWPE MAC example
- Write test data ($N \times 128$ bit keys and words) to memory
- Define HWPE registers
    - Bytecode defining operation
    - Source and destination pointers
    - Number of bytes to be loaded and stored
- Start HWPE operation (via register)
- Wait for HPWE done event
- Validate results

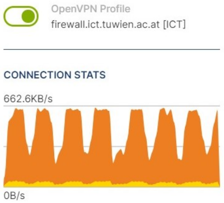
Results

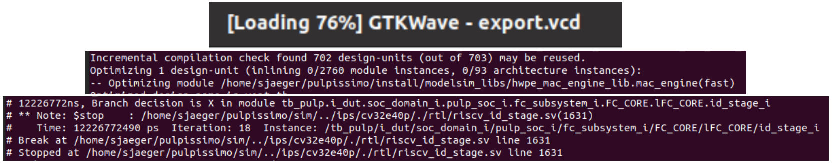
	AES Software	AES HWPE
Runtime $N = 32$	2876 μs	10 μs
Relative speed-up	1	287.6
Runtime further encryption	61 μs	280 ns
Code size	9816 bytes	9140 bytes

- Greatly reduced runtime leads to more CPU sleep time $\implies$ energy savings
- Code size reduced
- No area estimation due to synthesis issues

Challenges I

- Simulator (QuestaSim) only available on ICT EDA server
  - Nasty setup including several bash scripts
  - Slow VPN connection $\Longrightarrow$ terribly slow mounted file system
- PULPissimo is a **huge** design $\Longrightarrow$ long build & simulation times, hard debugging
  - Hardware change to vcd takes at least 10 minutes
  - vcd file sizes are quickly 10 GB
- PULP is an academic project $\Longrightarrow$ little documentation





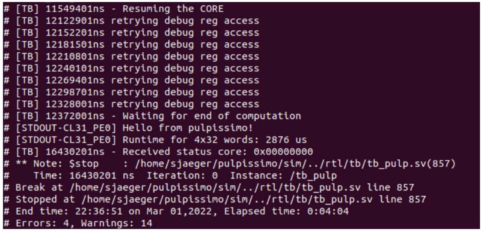
```
[Loading 76%] GTKWave - export.vcd
```

```
Incremental compilation check found 702 design-units (out of 703) may be reused.
Optimizing 1 design-unit (inlining 0/2760 module instances, 0/93 architecture instances):
-- Optimizing module /home/sjaeger/pulpissimo/install/modelsim_libs/hwpe_mac_engine_lib.mac_engine(fast)
```

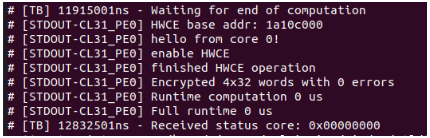
```
# 12226772ns, Branch decision is X in module tb_pulp.i_dut.soc_domain_i.pulp_soc_i.fc_subsystem_i.FC_CORE.lFC_CORE.id_stage_i
# ** Note: $stop    : /home/sjaeger/pulpissimo/sim/../ips/cv32e40p/./rtl/riscv_id_stage.sv(1631)
#    Time: 12226772490 ps  Iteration: 18  Instance: /tb_pulp/i_dut/soc_domain_i/pulp_soc_i/fc_subsystem_i/FC_CORE/lFC_CORE/id_stage_i
# Break at /home/sjaeger/pulpissimo/sim/../ips/cv32e40p/./rtl/riscv_id_stage.sv line 1631
# Stopped at /home/sjaeger/pulpissimo/sim/../ips/cv32e40p/./rtl/riscv_id_stage.sv line 1631
```

Challenges II

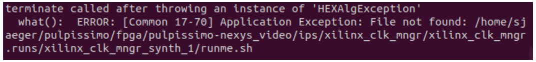
- Errors are normal (e.g. due to different QuestaSim version)
- HWPE template is extremely powerful (620 bit control signal) $\Longrightarrow$ hard to understand and modify
  - Solution: Leave control logic as it is, find suitable software configuration
- AES HWPE is faster than PULPissimo timer resolution $\Longrightarrow$ runtimes from vcd
- Provided scripts do not like me



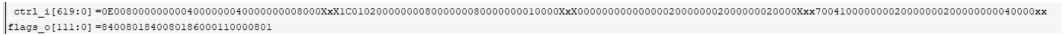
```
# [TB] 11549401ns - Resuming the CORE
# [TB] 12122901ns retrying debug reg access
# [TB] 12152201ns retrying debug reg access
# [TB] 12181501ns retrying debug reg access
# [TB] 12210801ns retrying debug reg access
# [TB] 12240101ns retrying debug reg access
# [TB] 12269401ns retrying debug reg access
# [TB] 12298701ns retrying debug reg access
# [TB] 12328001ns retrying debug reg access
# [TB] 12372001ns - Waiting for end of computation
# [STDOUT-CL31_PE0] Hello from pulpissimo!
# [STDOUT-CL31_PE0] Runtime for 4x32 words: 2876 us
# [TB] 16430201ns - Received status core: 0x00000000
# ** Note: $stop    : /home/sjaeger/pulpissimo/sim/../rtl/tb/tb_pulp.sv(857)
#    Time: 16430201 ns  Iteration: 0  Instance: /tb_pulp
# Break at /home/sjaeger/pulpissimo/sim/../rtl/tb/tb_pulp.sv line 857
# Stopped at /home/sjaeger/pulpissimo/sim/../rtl/tb/tb_pulp.sv line 857
# End time: 22:36:51 on Mar 01,2022, Elapsed time: 0:04:04
# Errors: 4, Warnings: 14
```



```
# [TB] 11915001ns - Waiting for end of computation
# [STDOUT-CL31_PE0] HWCE base addr: 1a10c000
# [STDOUT-CL31_PE0] hello from core 0!
# [STDOUT-CL31_PE0] enable HWCE
# [STDOUT-CL31_PE0] finished HWCE operation
# [STDOUT-CL31_PE0] Encrypted 4x32 words with 0 errors
# [STDOUT-CL31_PE0] Runtime computation 0 us
# [STDOUT-CL31_PE0] Full runtime 0 us
# [TB] 12832501ns - Received status core: 0x00000000
```



```
terminate called after throwing an instance of 'HEXAlgException'
  what():  ERROR: [Common 17-70] Application Exception: File not found: /home/sj
aeger/pulpissimo/fpga/pulpissimo-nexys_video/ips/xilinx_clk_mngr/xilinx_clk_mngr
.runs/xilinx_clk_mngr_synth_1/runme.sh
```



```
 ctrl_i[619:0] =0E008000000000004000000040000000008000XxX1C010200000000800000000800000000010000XxX000000000000000020000000200000000020000Xxx70041000000002000000020000000040000xx
flags_o[111:0] =8400801840080186000110000801
```

Summary and Outlook

- AES HPWE integrated into PULPissimo
  - Speed-up of 287.6 achieved
  - Code size reduced
  - Likely significant energy savings
- Possible extensions
  - Investigate different AES implementations or other algorithms
  - ASIC implementation of HWPE to analyse area and power

References

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

[2] tiny-AES-c repository on GitHub https://github.com/kokke/tiny-AES-c

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