Designing FPGA Accelerators With HLS

Dr. Endric Schubert, Univ. Ulm & Missing Link Electronics
Lisa Halder, Univ. Ulm & Xilinx Labs Ireland

We are
a Silicon Valley based technology company with Offices in Germany. We are Partner of leading electronic device and solution providers and have been enabling key innovators in the automotive, industrial, test & measurement markets to build better Embedded Systems, faster.

Our Mission is
To develop and market technology solutions for Embedded Systems Realization via pre-validated IP and expert application support, and to combine off-the-shelf FPGA devices with Open-Source Software for dependable, configurable Embedded System platforms

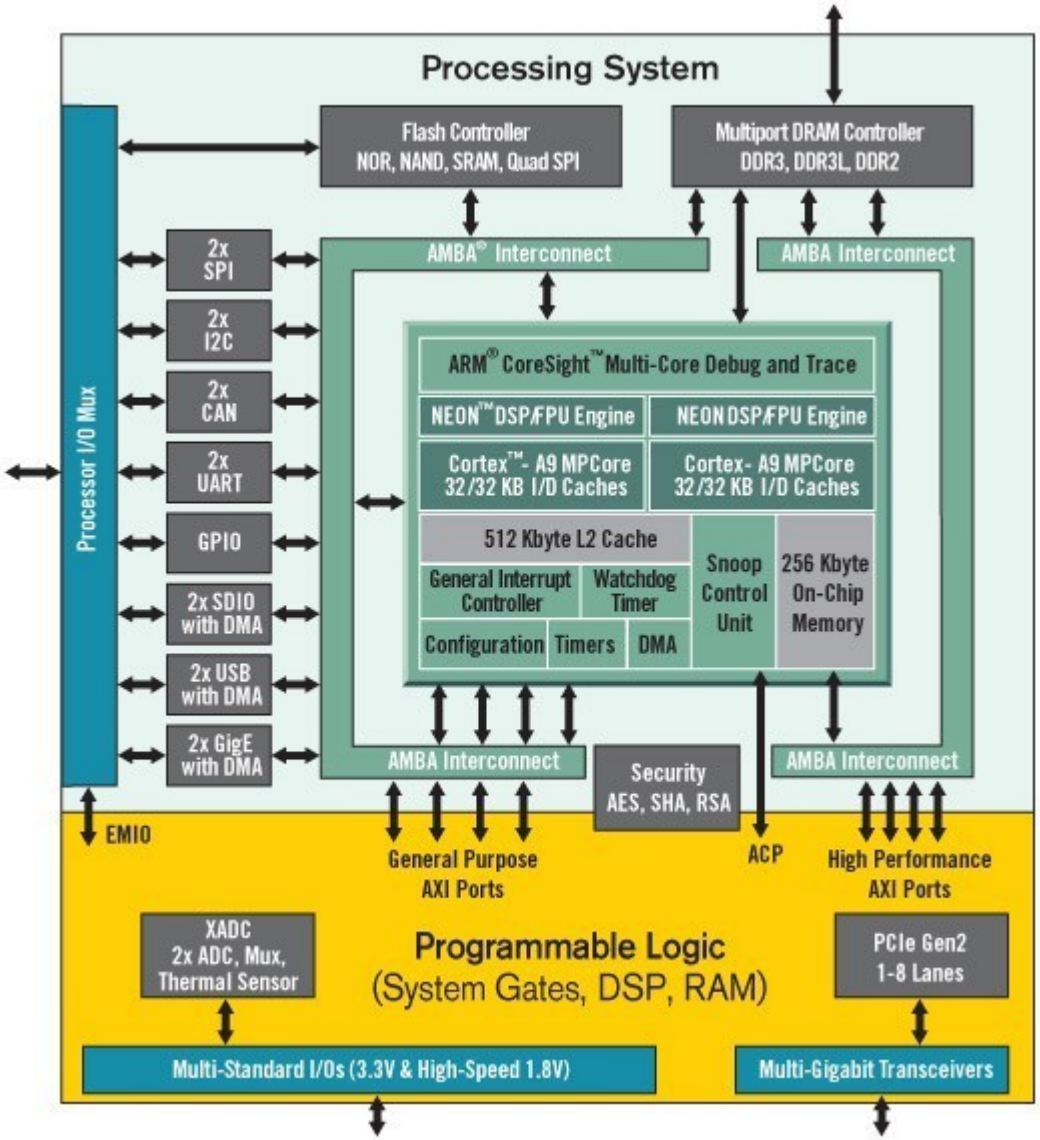
Our Expertise is
I/O connectivity and Acceleration of data communication protocols, additionally opening up FPGA technology for analog applications, and the integration and optimization of Open Source Linux and Android software stacks on modern extensible processing architectures.

Introduction and Motivation

SoC FPGA as (yet) another computer

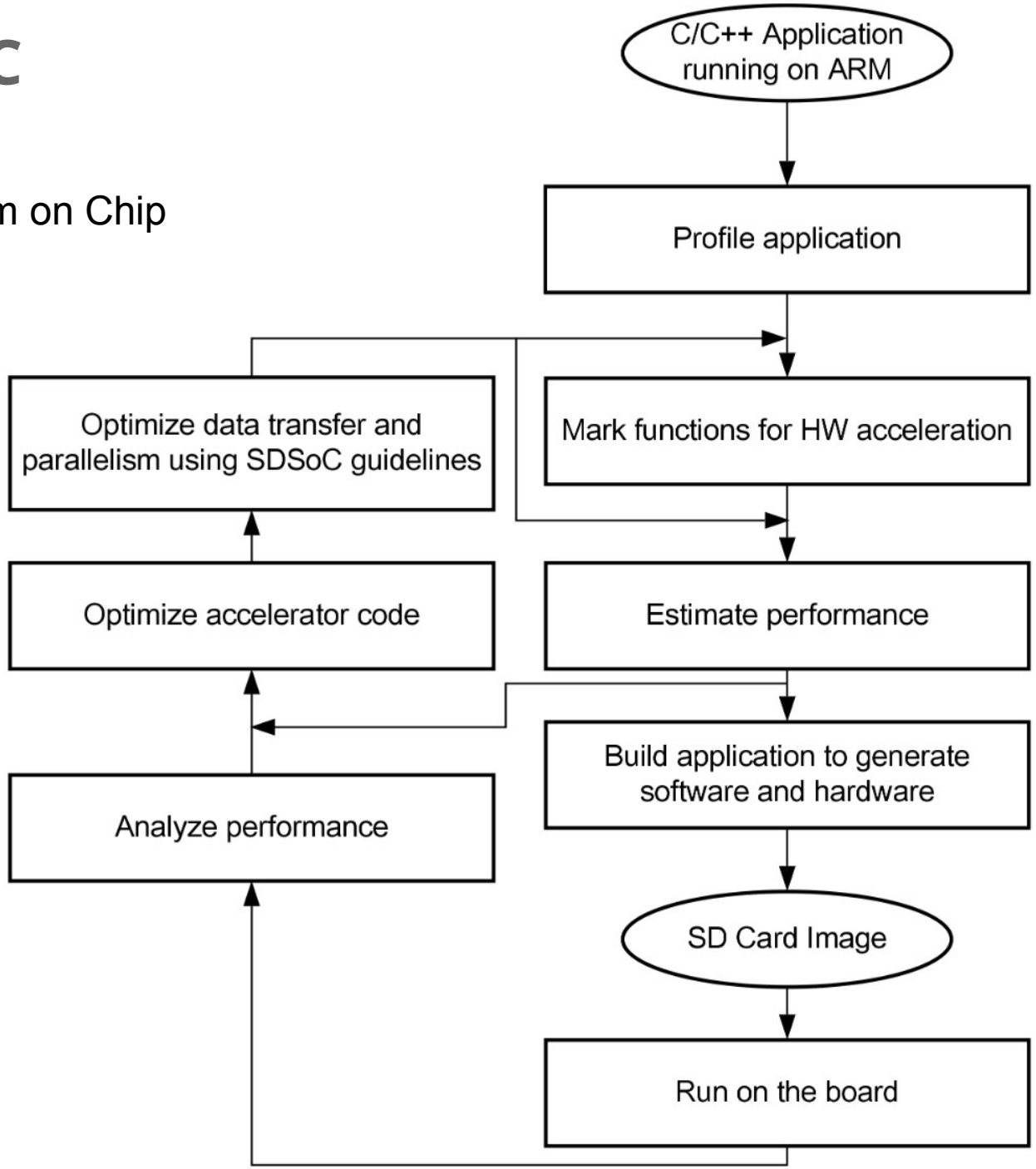
	Intel i7-4770	Xilinx Zynq 7045
Compute	~100 GFLOPS	5 GFLOPS (PS) 778 GFLOPS (PL)
TDP	84 W	<20 W (typ)

SOC FPGA has 4x more compute with ¼ the power dissipation!



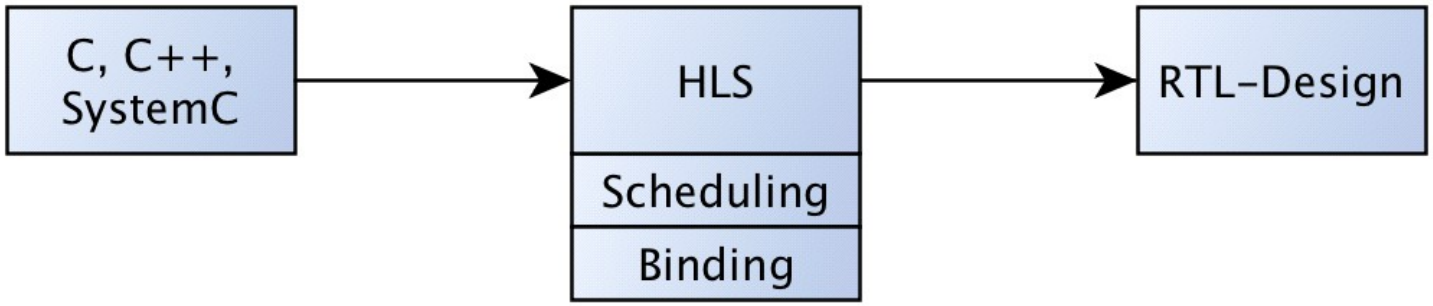
Intro SDSoC

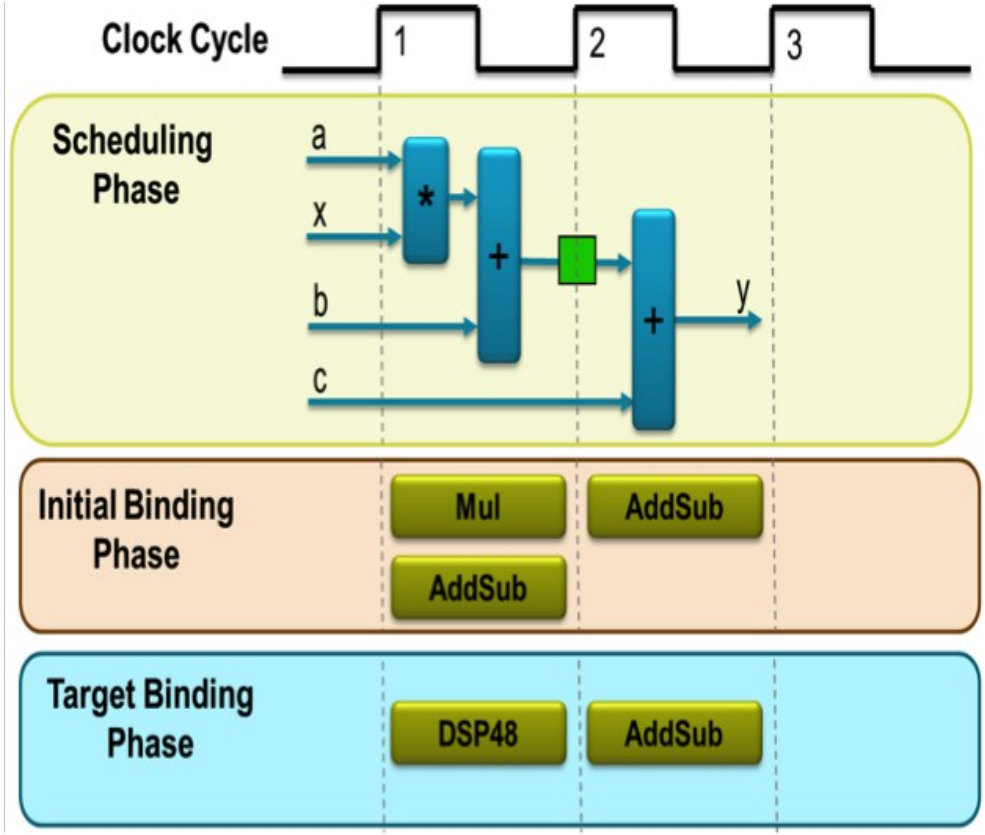
- SDSoC – Software Defined System on Chip
- IDE for developing heterogeneous embedded systems using the Zynq7000 platform
- transformation of C/C++ Code into complete hardware/software systems
- References:
  UG1027-intro-to-sdsoc.pdf
  UG1028-sdsoc-getting-started.pdf

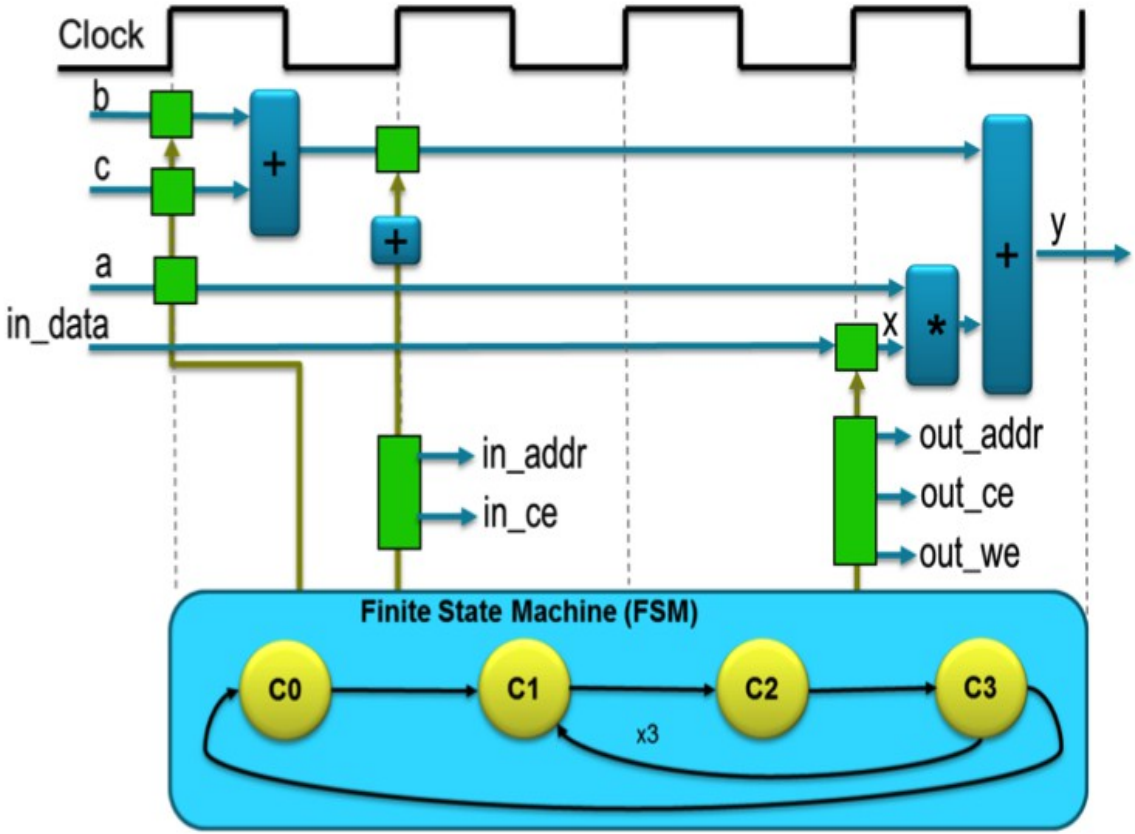


http://www.xilinx.com/support/documentation/sw_manuals/xilinx2015_4/ug1027-sdsoc-user-guide.pdf

Concept of High-Level Synthesis







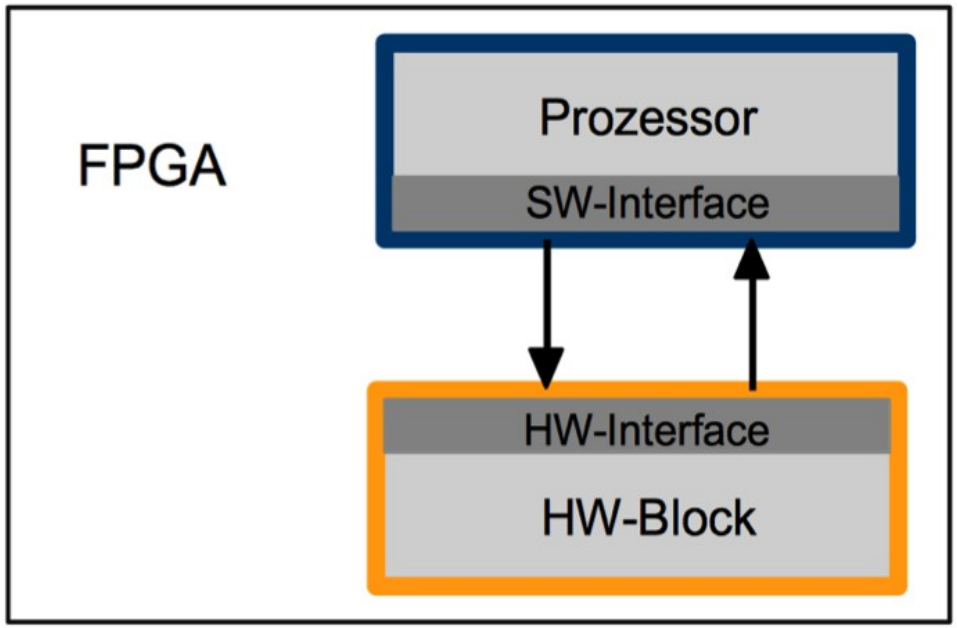
Concept of Interface Synthesis

RTL-Designflow:

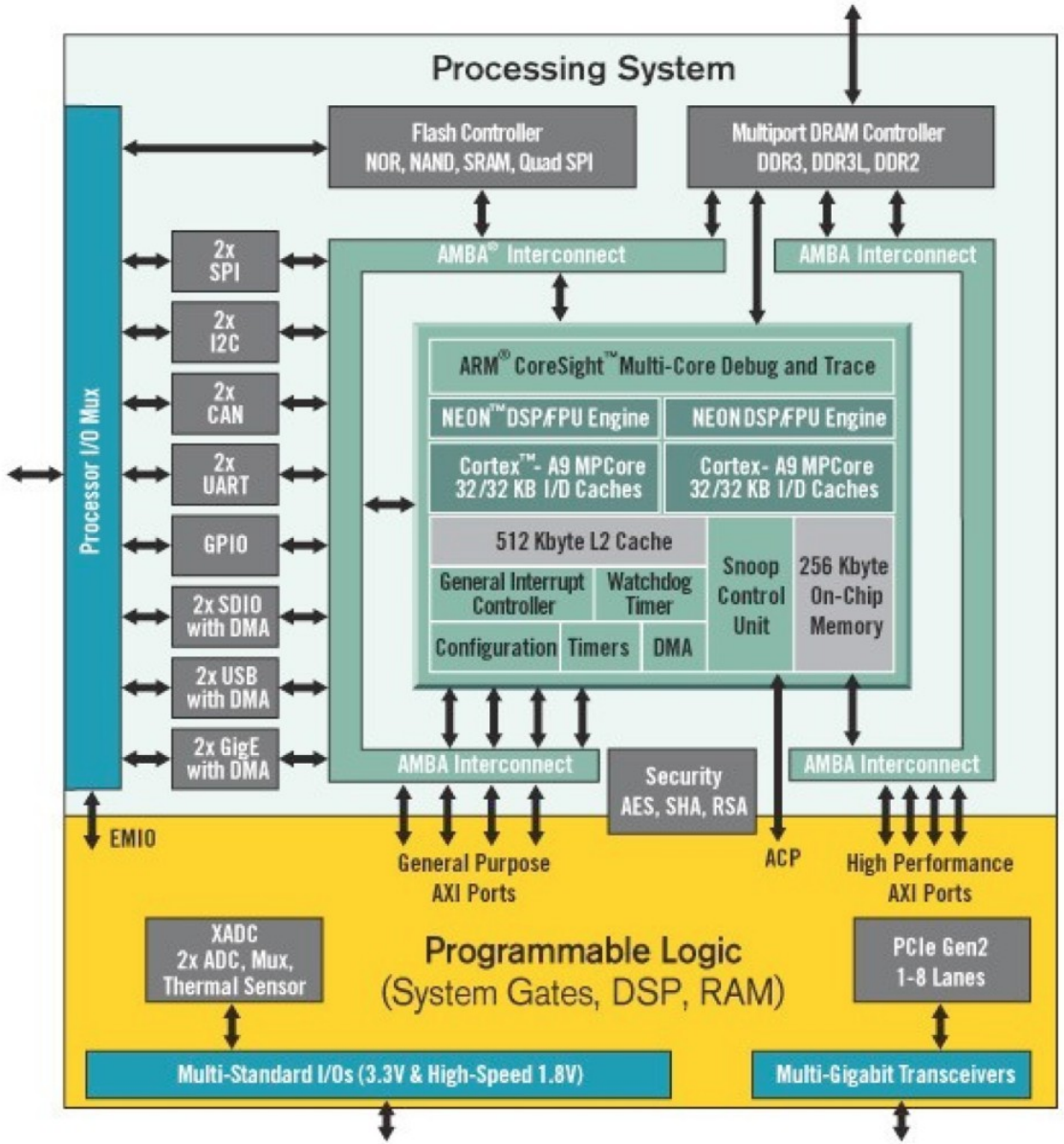
- Use of implemented IP-Cores
- Integration of the IP-Core manually
- Implementation of SW and HW for controlling the communication manually

HLS-Designflow:

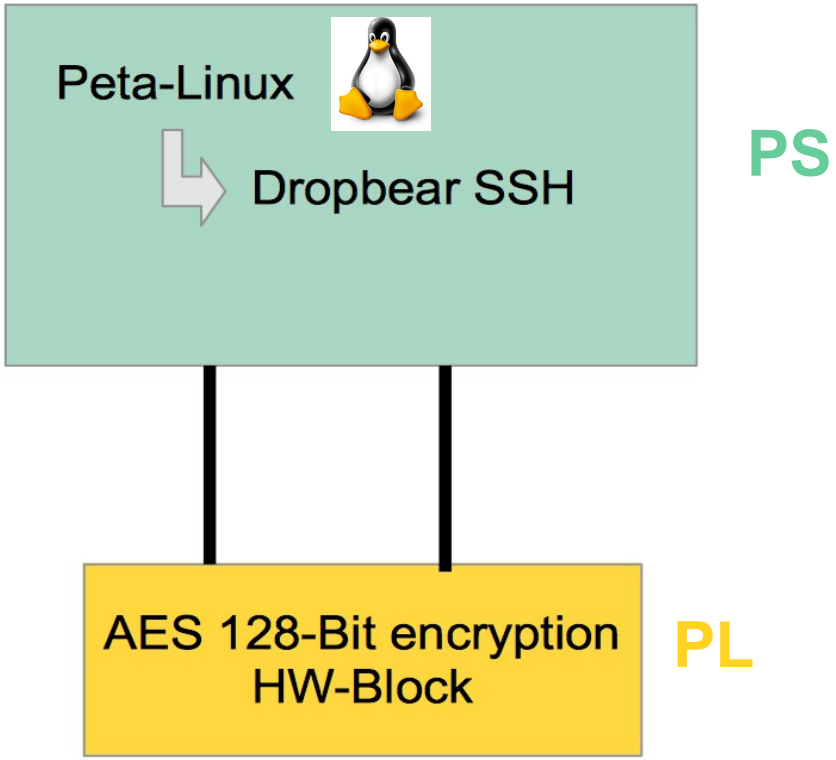
- Generation of HW-blocks out of source code by High-Level-Synthesis tools
- Automatic integration of the HW-Block through Interface-Synthesis
- Generation of HW and SW for controlling the communication



Design Example

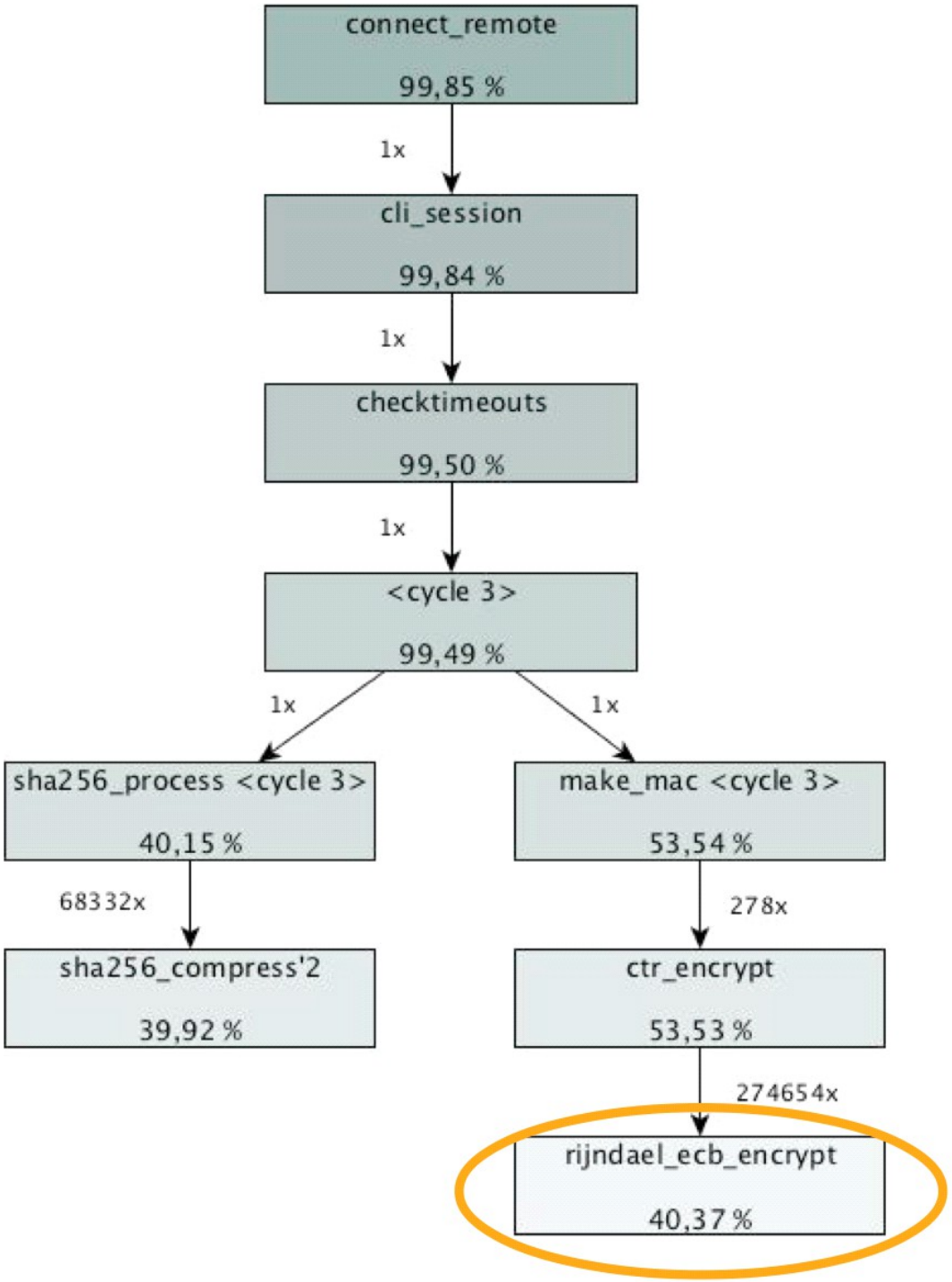


Zedboard



Profiling

- Profiling tool: valgrind - callgrind
- analyzation of data transfer through ssh via Dropbear SSH
- 128-Bit AES encryption
- Percentage of AES encryption „rijndael_ecb_encrypt“ very high (40,37%)
- AES implementation out of Dropbear SSH



Software Adjustments

```
#define ECB_ENC  rijndael_ecb_encrypt
int ECB_ENC(const unsigned char *pt, unsigned char *ct,
    symmetric_key *skey)
```

Software Adjustments

```
#define ECB_ENC  rijndael_ecb_encrypt
int ECB_ENC(const unsigned char *pt, unsigned char *ct,
    symmetric_key *skey)
```

modified source code

```
int rijndael_ecb_encrypt_hw(const unsigned char pt[16], unsigned char
   ct[20], uint32_t eK)
```

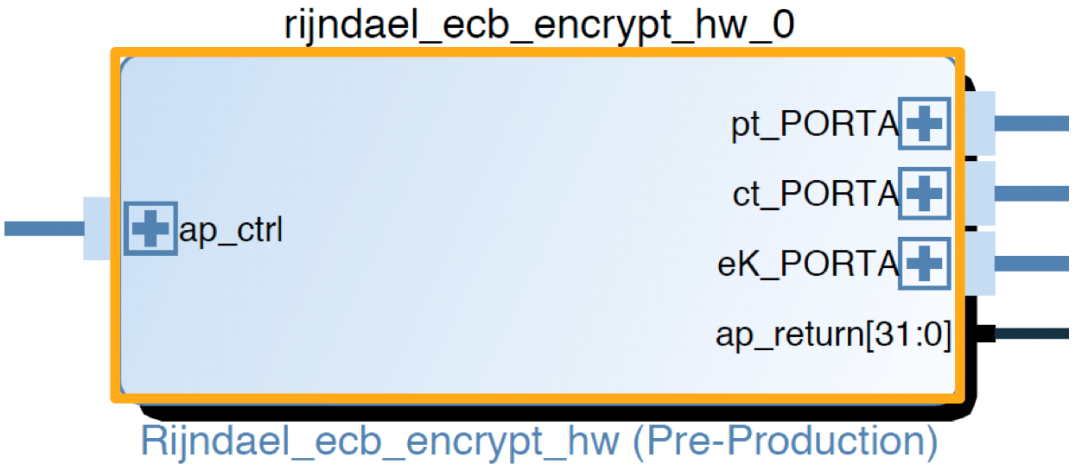
Software Adjustments

```
#define ECB_ENC  rijndael_ecb_encrypt
int ECB_ENC(const unsigned char *pt, unsigned char *ct,
    symmetric_key *skey)
```

modified source code

```
int rijndael_ecb_encrypt_hw(const unsigned char pt[16], unsigned char
   ct[20], uint32_t eK)
```

HLS



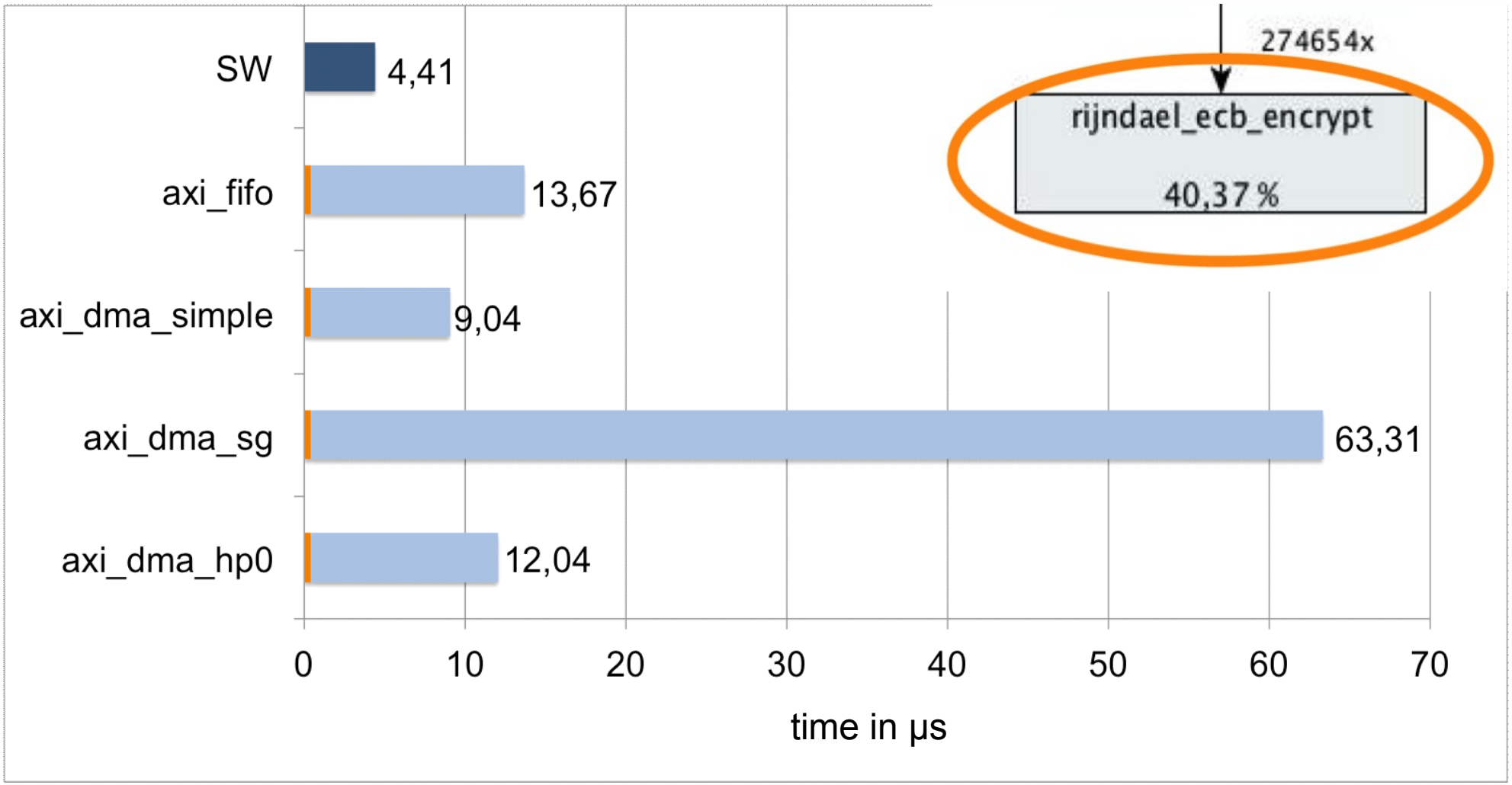
Datamover

- Datamover are generated automatically by SDSoC
- user can influence the generated Datamover by modifiying the dataflow
  e.g. declaring array as contigous memory

SDSoC Data Mover	Vivado IP Data Mover	Accelerator IP Port Types	Transfer Size	Contigous Memory Only
axi_lite	processing_system7	register		
axi_fifo	axi_fifo_mm_s	bram, ap_fifo, axis	< 300 B	
axi_dma_simple	axi_dma	bram, ap_fifo, axis	< 8 MB	✔
axi_dma_sg	axi_dma	bram, ap_fifo, axis		
zero_copy	accelerator IP	aximm master		✔
axi_dma_2d	axi_dma	bram		✔

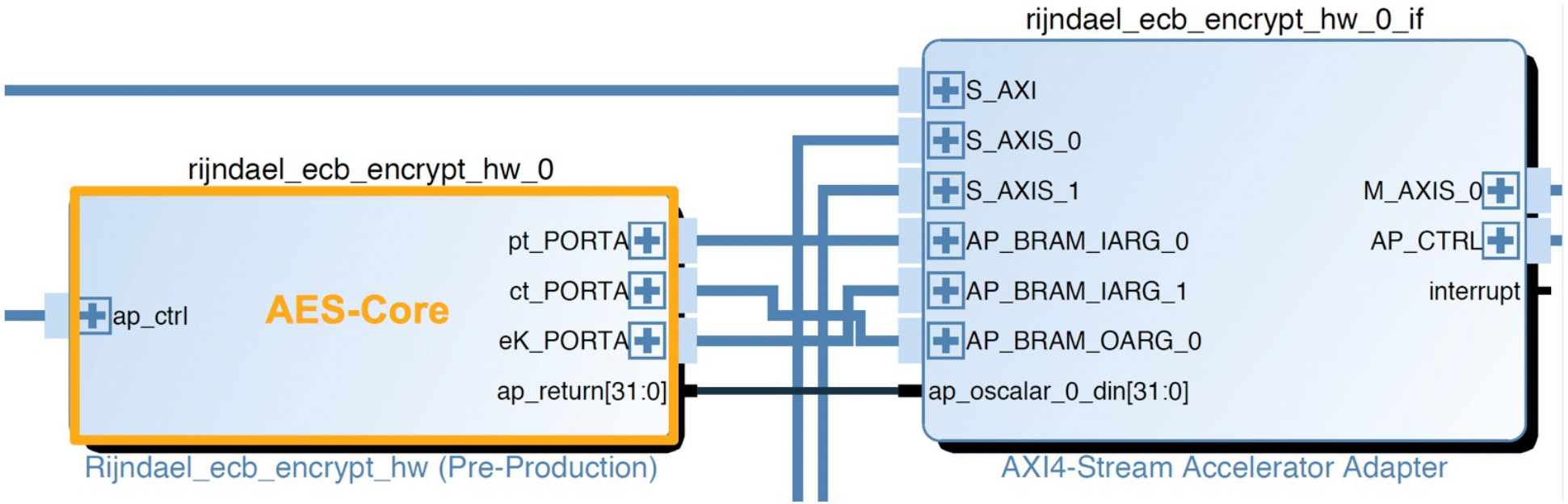
The simple (and inefficient) Way

Encryption of a 16 Byte Array, which corresponds to one call of rijndael_ecb_encrypt

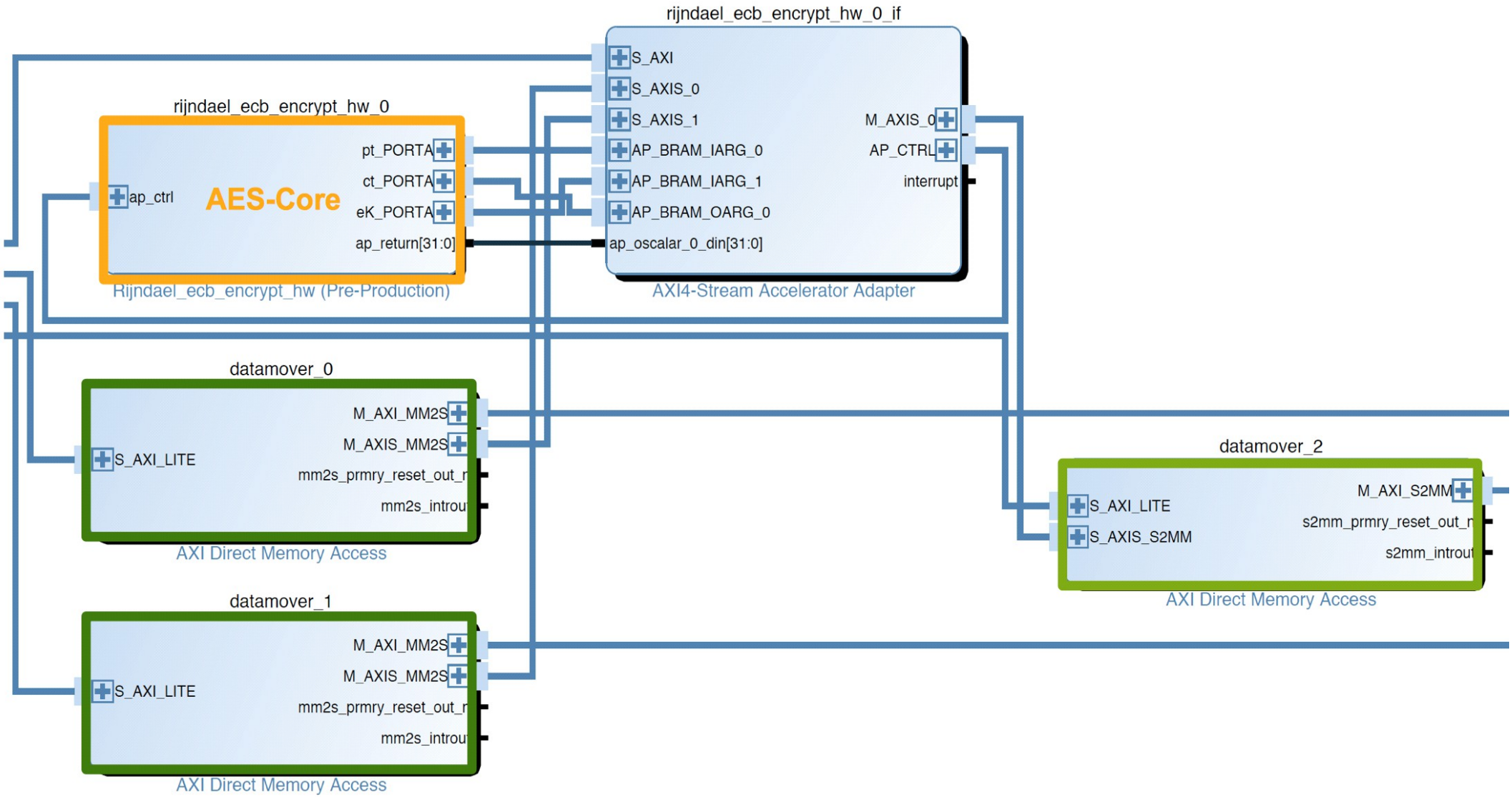


HW-function (AES) takes 0,44 µs for execution

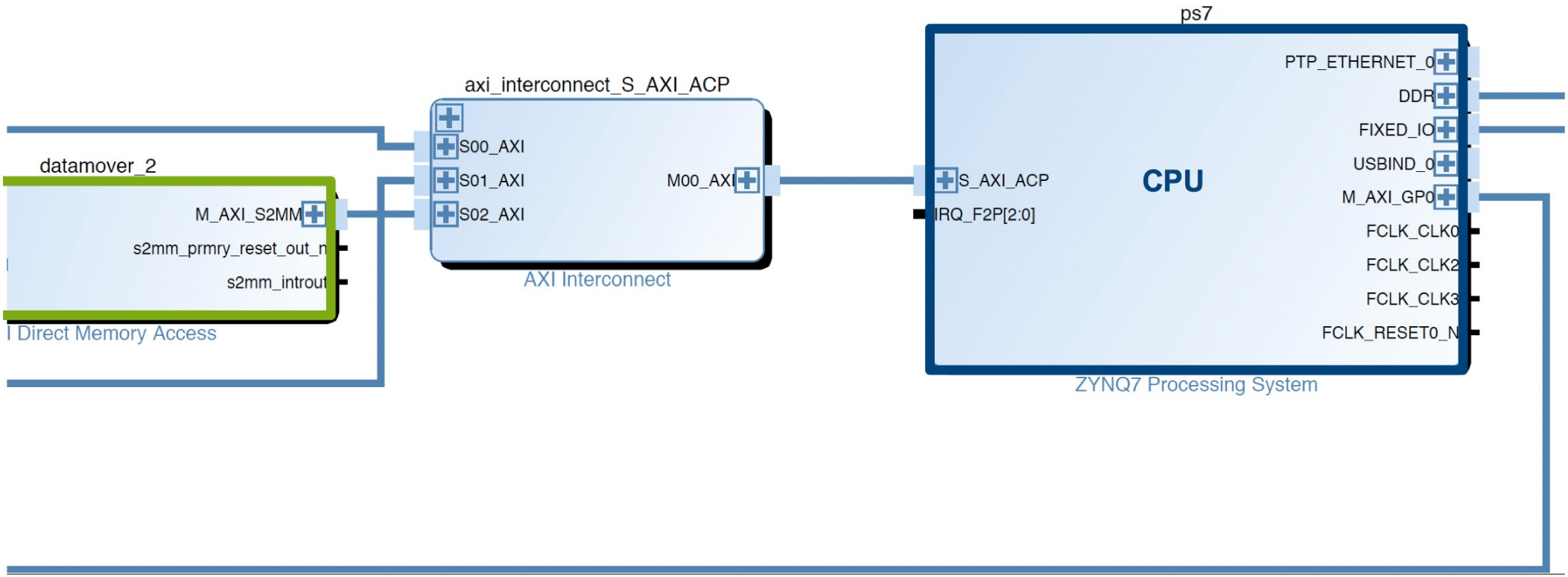
Blockdiagram with axi_dma_simple



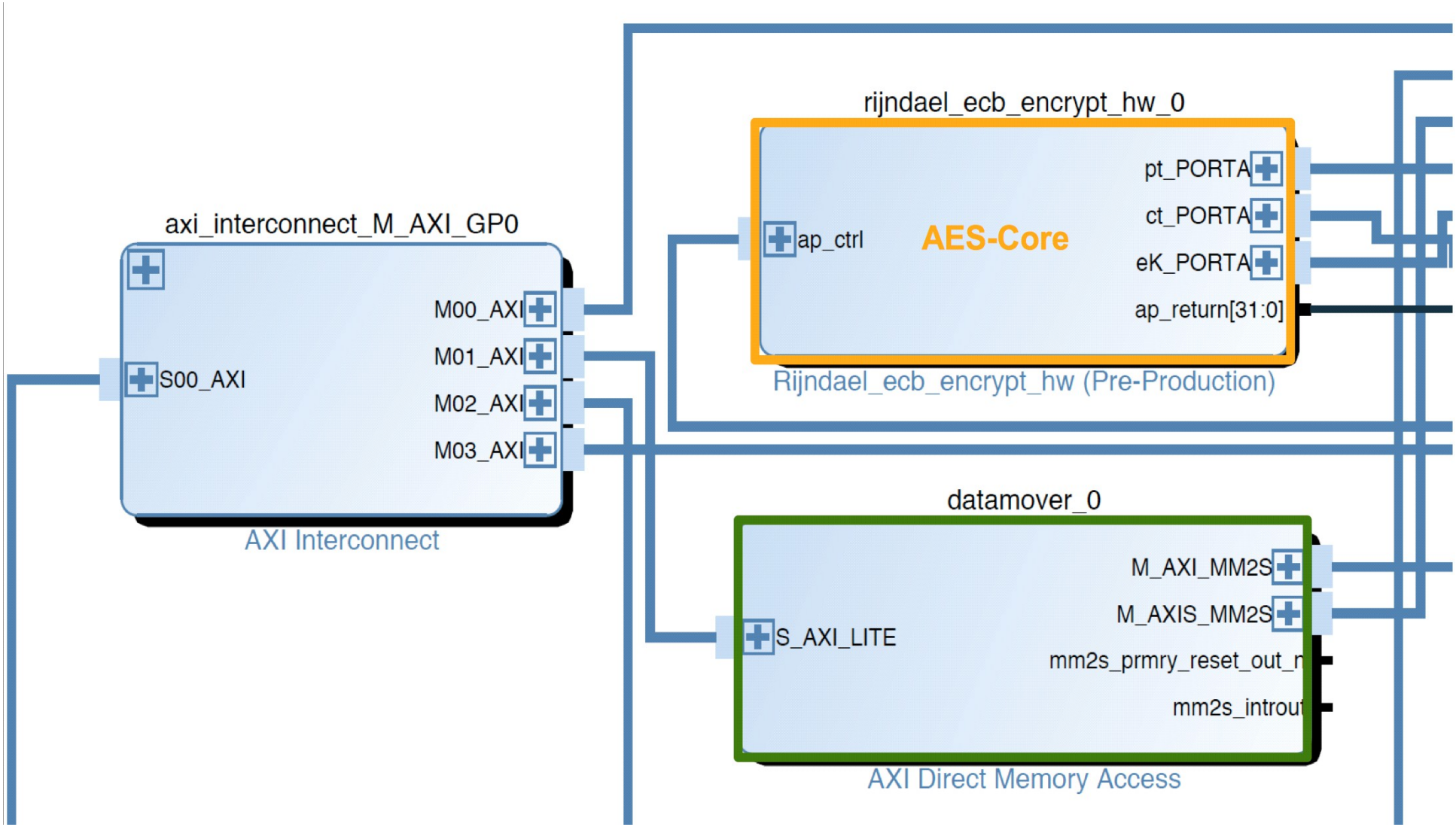
Blockdiagram with axi_dma_simple



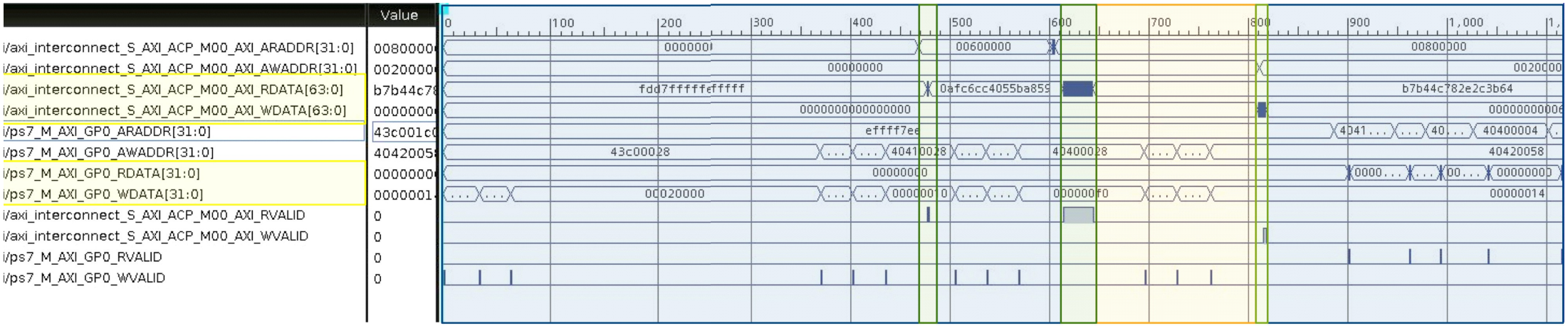
Blockdiagram with axi_dma_simple



Blockdiagram with axi_dma_simple



Analyzation of axi_dma_simple

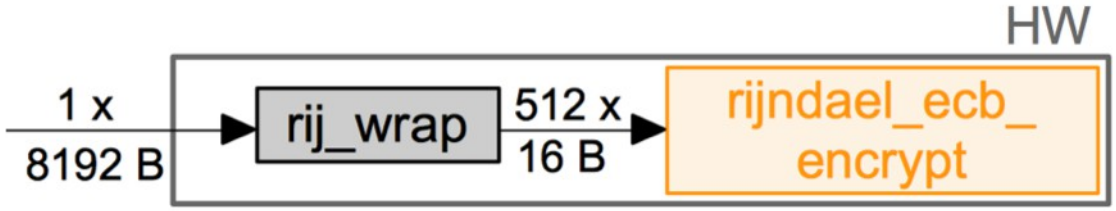


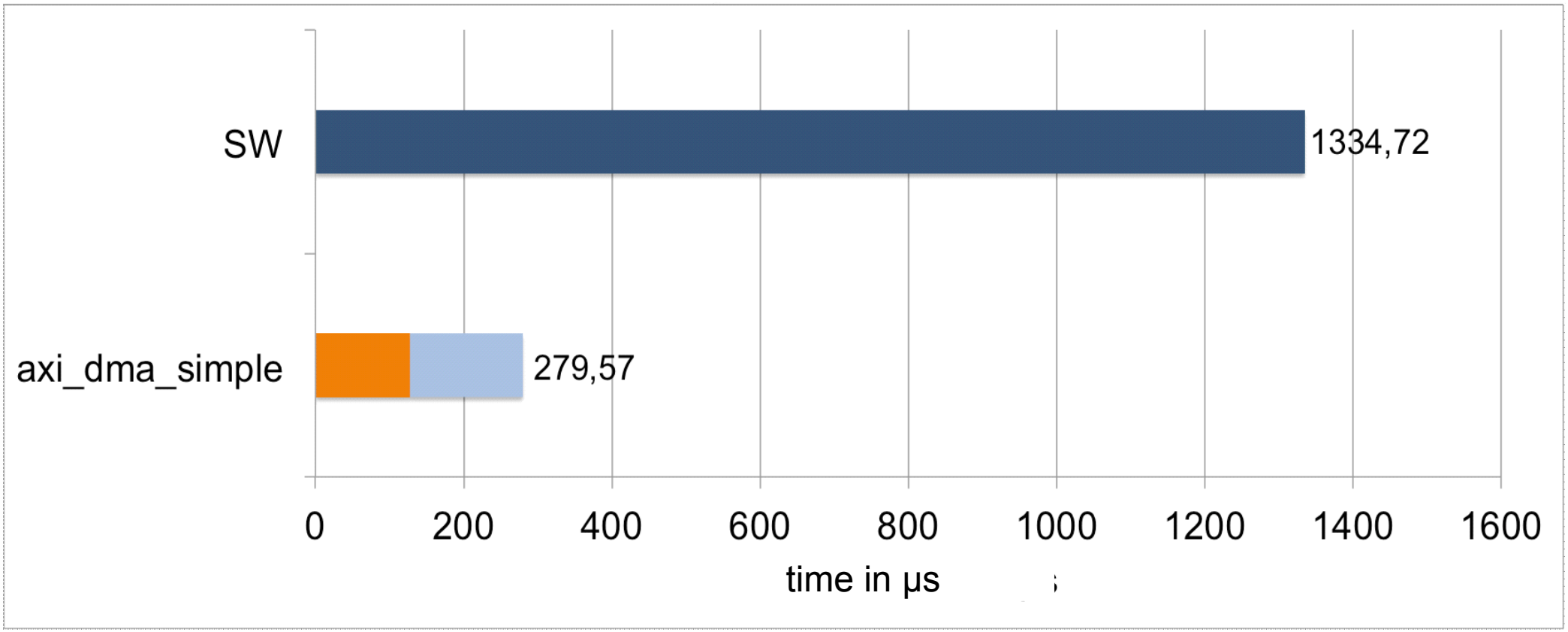
Frequency:	142 MHz	µs
Initialization:	614 HWTZ	4,32 µs
Read access:	31 HWTZ	0,22 µs
HW-function:	166 HWTZ	1,17 µs
Write access:	7 HWTZ	0,05 µs
Waiting time:	82 HWTZ	0,58 µs
Reading of the status register:	218 HWTZ	1,53 µs
Sum:	1118 HWTZ	7,87 µs

Dataflow optimization

Encryption of an 8192 Byte array by rij_wrap
call of rijndael_ecb_encrypt 512 times

optimization of the dataflow
reduction of communication overhead





Conclusion

- Designing FPGA accelerators for (legacy) Software
  - HLS and Interface Synthesis automates migration from SW to HW
  - Xilinx SDSoC a new and efficient methodology for Programmable SoC
  - Not all SW constructs are fully supported (yet) but efficient work-around exist
  - ==> Development times reduced from weeks down to days

- Please visit us at the show:

MLE – Hall 2 Booth 2-421

Xilinx – Hall 1 Booth 1-205