High-Throughput Multi-Threaded Sum-Product Network Inference in the Reconfigurable Cloud

TECHNISCHE UNIVERSITÄT DARMSTADT

Micha Ober, Jaco A. Hofmann, Lukas Sommer, Lukas Weber, Andreas Koch

Embedded Systems and Applications Group, TU Darmstadt

Embedded Systems & Applications

Agenda

TECHNISCHE UNIVERSITÄT DARMSTADT

- TaPaSCo in the Clouds
  - Introduction to the TaPaSCo framework
  - Challenges in porting TaPaSCo to Amazon AWS F1
- High-Throughput Sum-Product Network Inference
  - Introduction to Sum-Product Networks
  - FPGA Acceleration Toolflow
  - Optimizations for Amazon AWS F1
  - Evaluation
- Conclusion

Embedded Systems & Applications

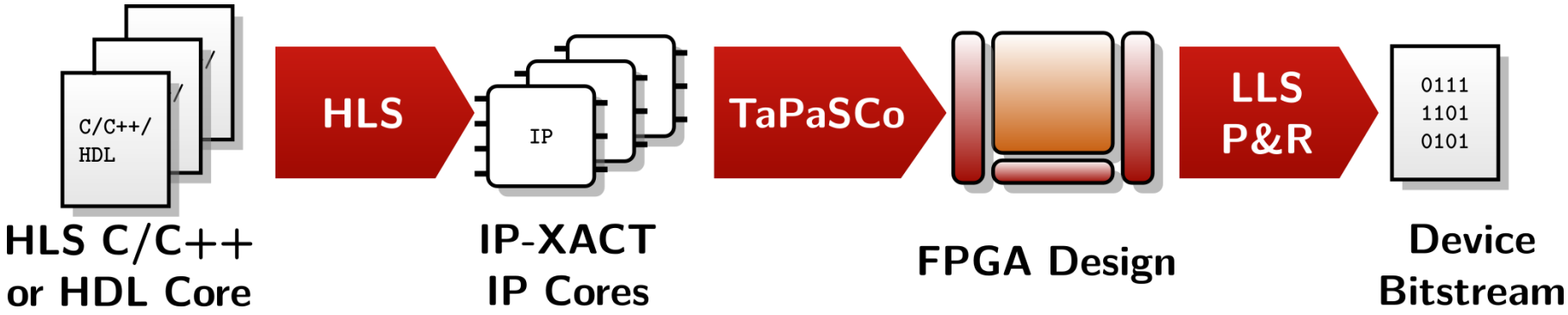
TaPaSCo Framework

TECHNISCHE UNIVERSITÄT DARMSTADT

- Builds complete FPGA SoC-designs from HLS kernels or custom HDL cores
- Automates Design-Space Exploration to determine best system composition
- Supports wide variety of Xilinx platforms
- Includes software API for dispatching compute tasks to FPGA
- Available as free & open-source software

Embedded Systems & Applications

TaPaSCo Design Flow



Core name

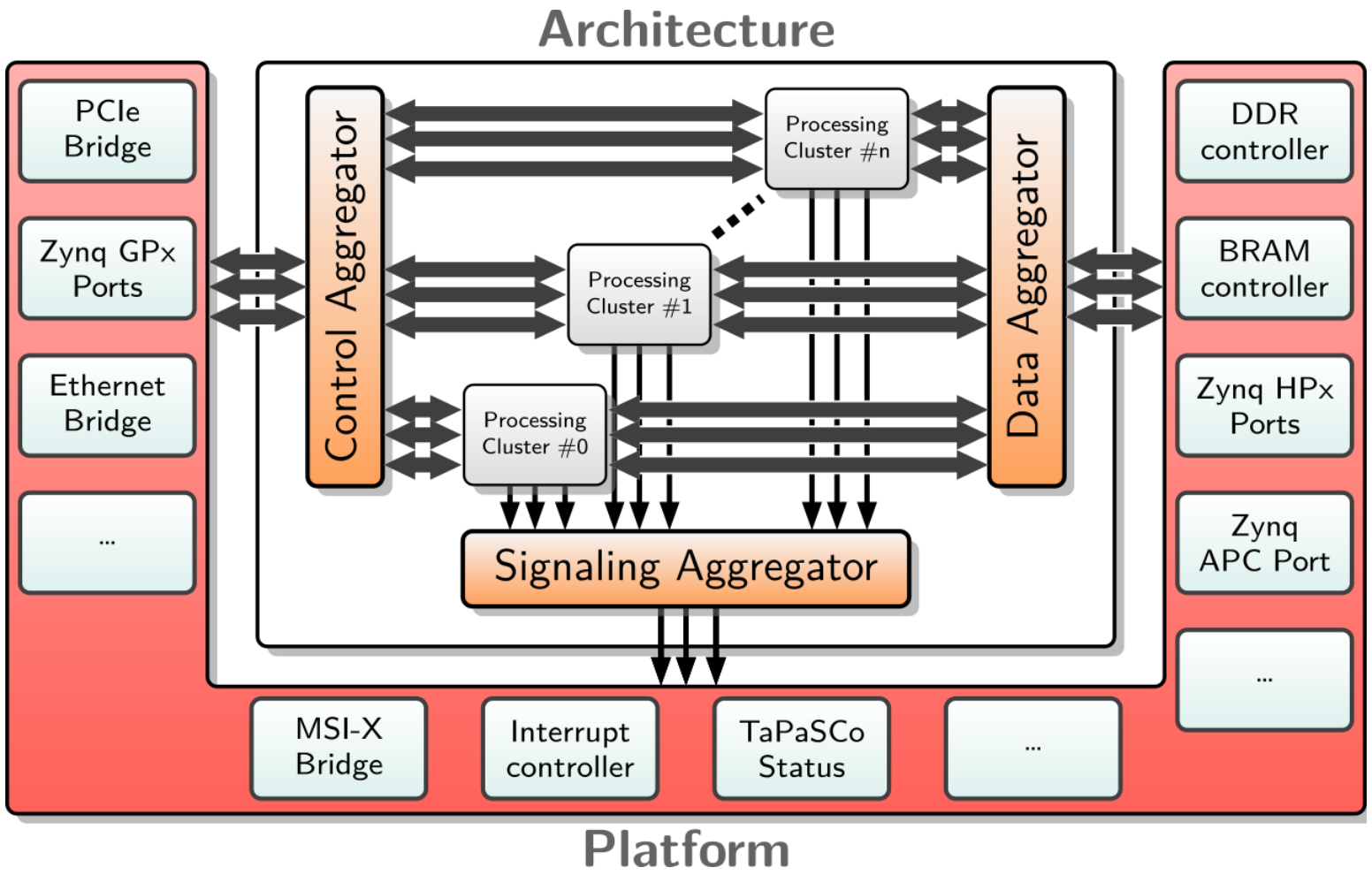
Design frequency

```
tapasco compose [cnn x 2, sobel x 3] @ 100 MHz -p vc709
```

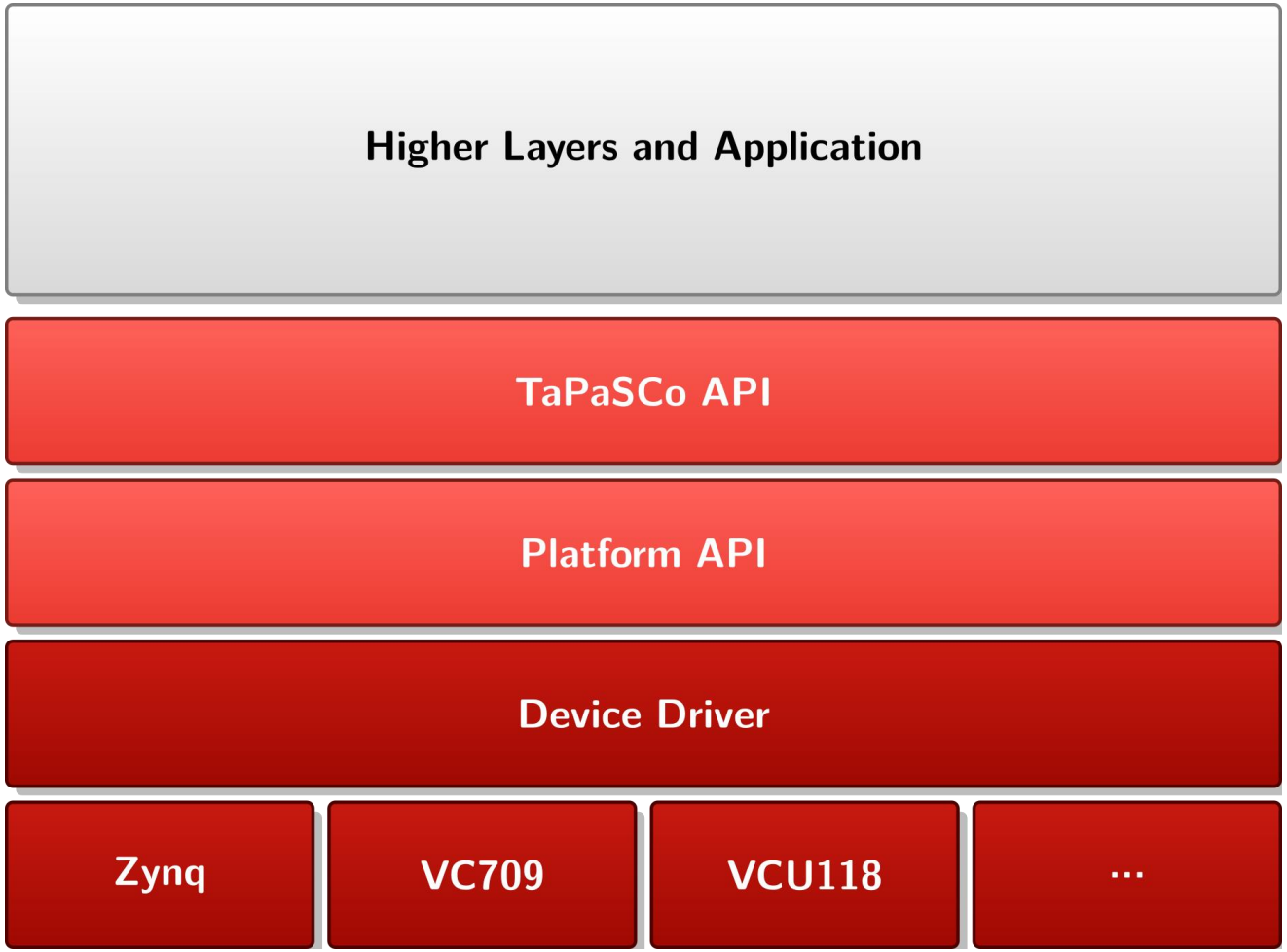
Core count

Platform

TaPaSCo Architecture



TaPaSCo Software API



TaPaSCo Software API – Example

```
Tapasco tapasco;
auto a_wrapped = makeWrappedPointer(a.data(), a.size());
auto b_wrapped = makeWrappedPointer(b.data(), b.size());

auto job = tapasco.launch(SIMPLE_HLS_ID,
  makeInOnly(a_wrapped), makeOutOnly(b_wrapped));
job();
```

Wrap information about data-transfer

Launch FPGA execution

Provide information about data-transfer direction

TaPaSCo Platforms

TECHNISCHE UNIVERSITÄT DARMSTADT

Datacenter

- Xilinx Alveo U250
- Xilinx Virtex UltraScale+ VCU1525
- Xilinx Virtex UltraScale+ VCU118
- Xilinx Virtex UltraScale VCU108
- Digilent NetFPGA SUME
- Xilinx Virtex VC709

Edge Devices

- Xilinx Zynq UltraScale+ MPSoC ZCU102
- Xilinx Zynq SoC ZC706
- AVNET ZedBoard
- Digilent Pynq-Z1

Embedded Systems & Applications

TaPaSCo in the Cloud

- Amazon deploys Xilinx VU9+ FPGAs in AWS EC2 F1 instances
- Most of the FPGA logic freely programmable, all interfaces routed through fixed *Shell* provided by Amazon

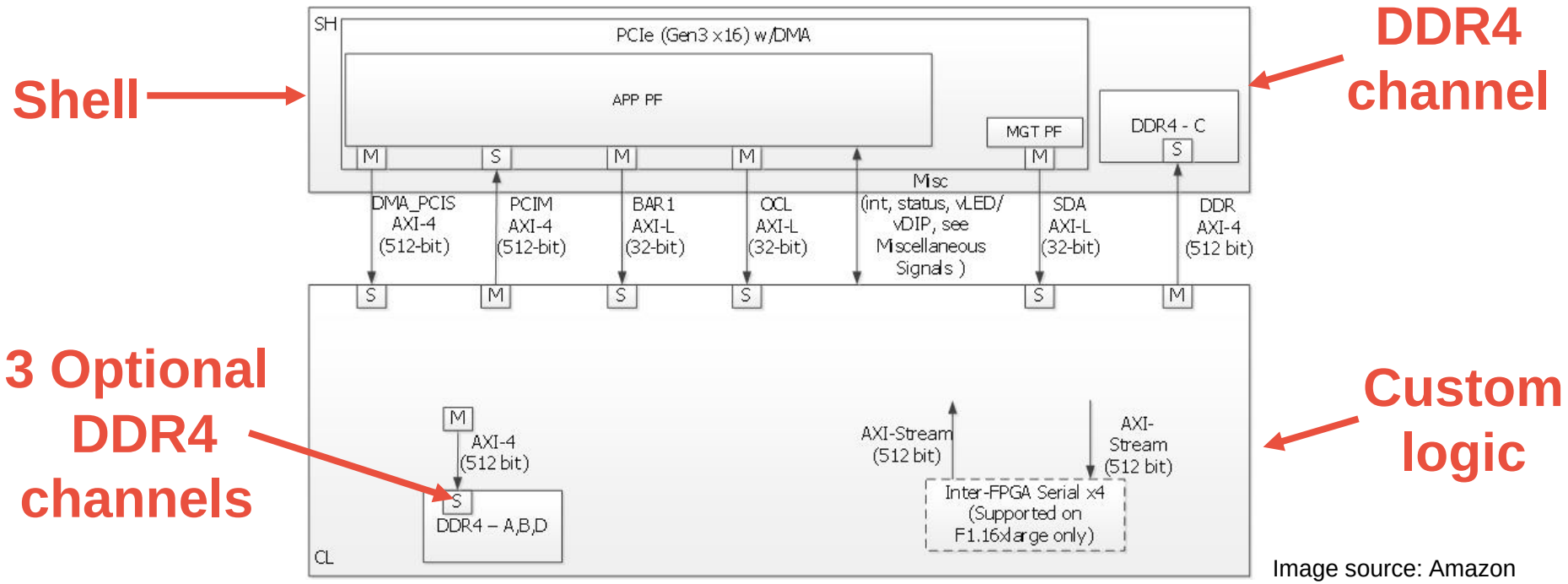


Image source: Amazon

TaPaSCo in the Cloud - Challenges

TECHNISCHE UNIVERSITÄT DARMSTADT

- *Shell* provides only a few frequencies, TaPaSCo supports arbitrary design frequencies
  - Include custom clock controller in programmable logic
- DMA engine in *Shell* provides only limited throughput
  - Replace with TaPaSCo‘s own DMA engine
- *Shell* provides only 16 interrupts, not enough for TaPaSCo architecture
  - Include custom interrupt controller for translation
- Memory controllers for 3 DDR channels have to be placed in custom logic
  - Carefull timing necessary

Embedded Systems & Applications

TaPaSCo in the Clouds – Conclusion

- Completely automated toolflow to generate SoC-design from HLS code or custom HDL core for Amazon AWS EC2 F1 FPGA instances
- Generates ready-to-use Amazon FPGA Image (AFI)
- Supports up to four independent memory channels
- Easy-to-use software API for interfacing with FPGA accelerator
- Open-source available!

TECHNISCHE UNIVERSITÄT DARMSTADT

Case-Study

SUM-PRODUCT NETWORK INFERENCE

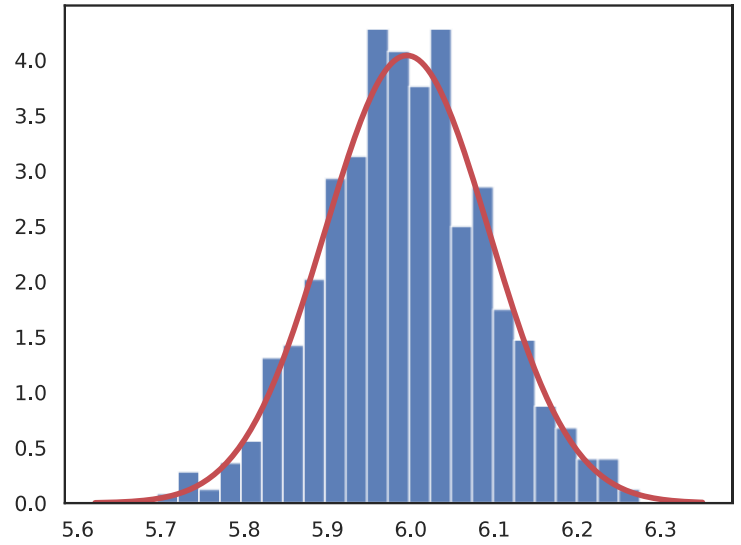
Embedded Systems & Applications

Sum-Product Networks

- ML technique from the class of probabilistic models
- Capture joint probability over a set of random variables
- Advantage over NN: Exact inference, express uncertainty over output
- Advantage over other PGM: Tractable inference in linear time wrt. network size
- Three kinds of nodes in DAG:
  - Sum nodes
  - Product nodes
  - Leaf nodes

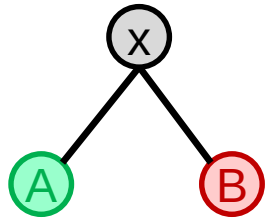
Sum-Product Networks – Leaf Nodes

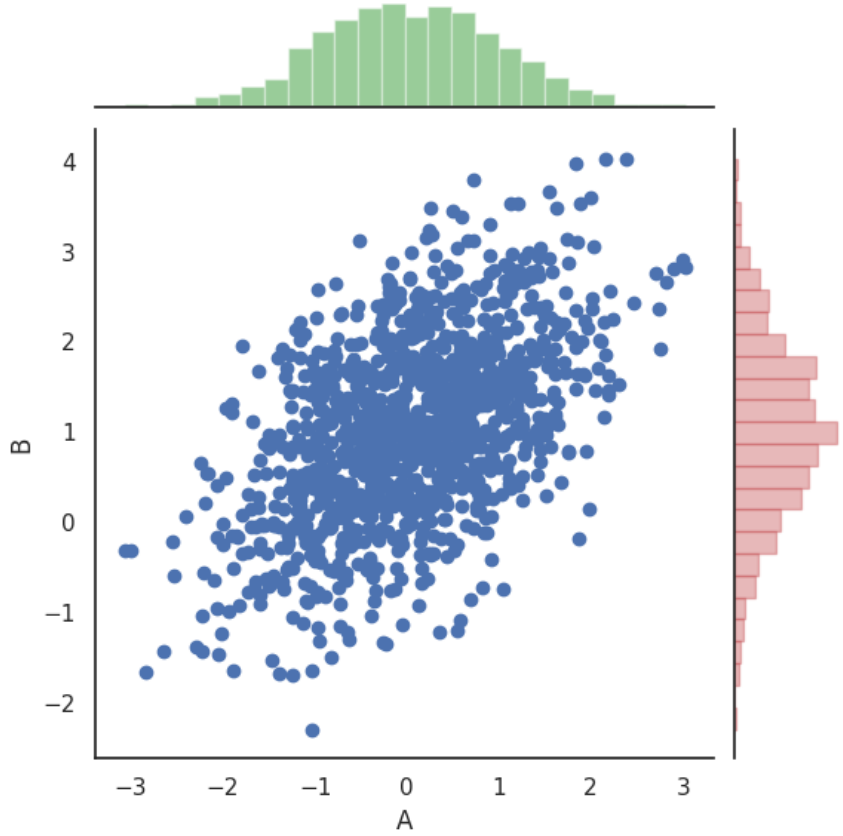
- Capture univariate distributions, e.g., Gaussian, Poisson;
- Queried with evidence (input value) to obtain probability value
- Can be represented efficiently using histograms



Sum-Product Networks – Product Nodes

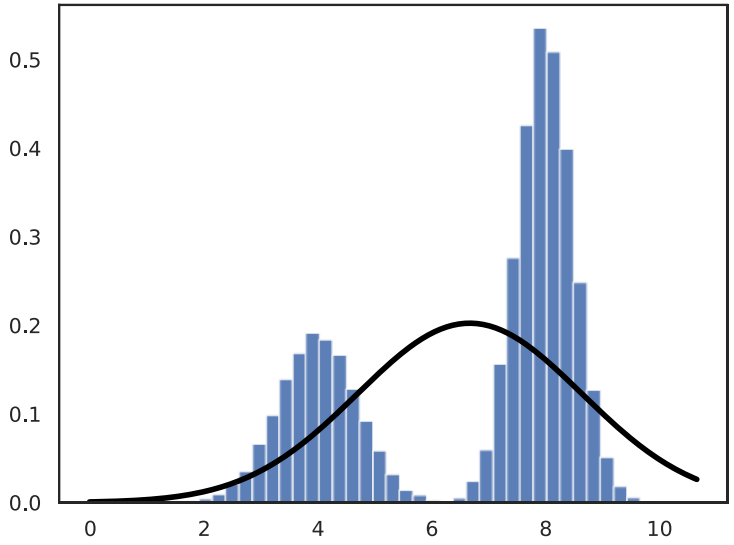
- Factorization over independent random variables
- Multiply probability value from child nodes to obtain result
- Domain knowledge might be required to determine independence





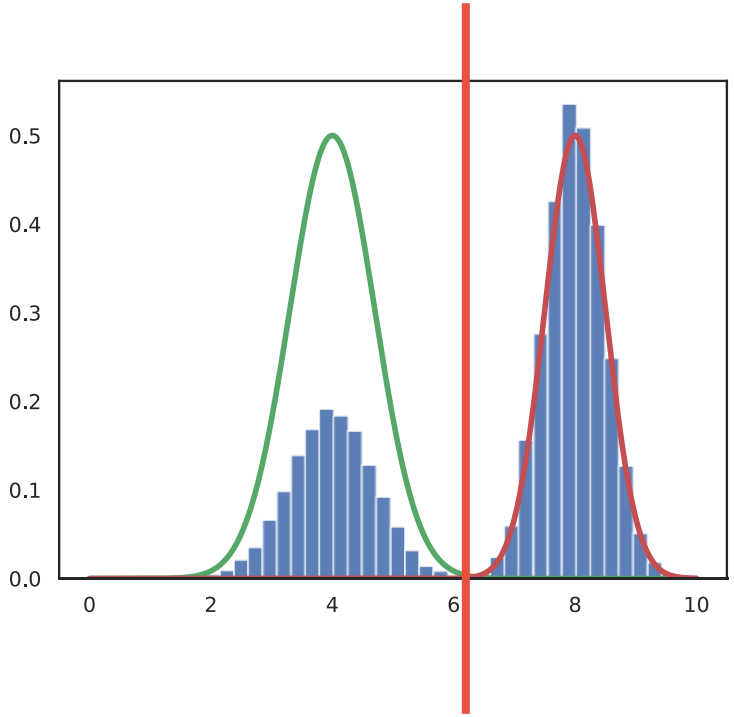
Sum-Product Networks – Sum-Node

- Mixture of two distributions over the same set of random variables



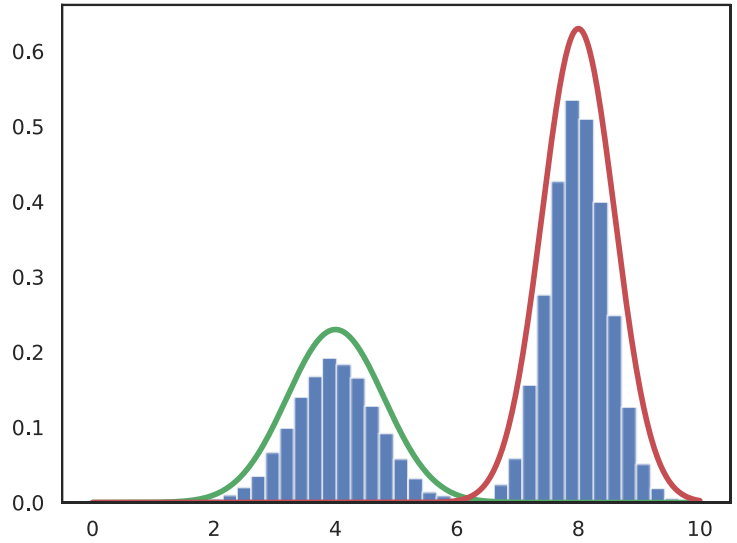
Sum-Product Networks – Sum-Node

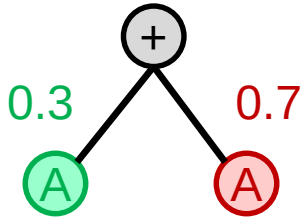
- Mixture of two distributions over the same set of random variables
- Cluster and split samples, e.g. kNN-clustering



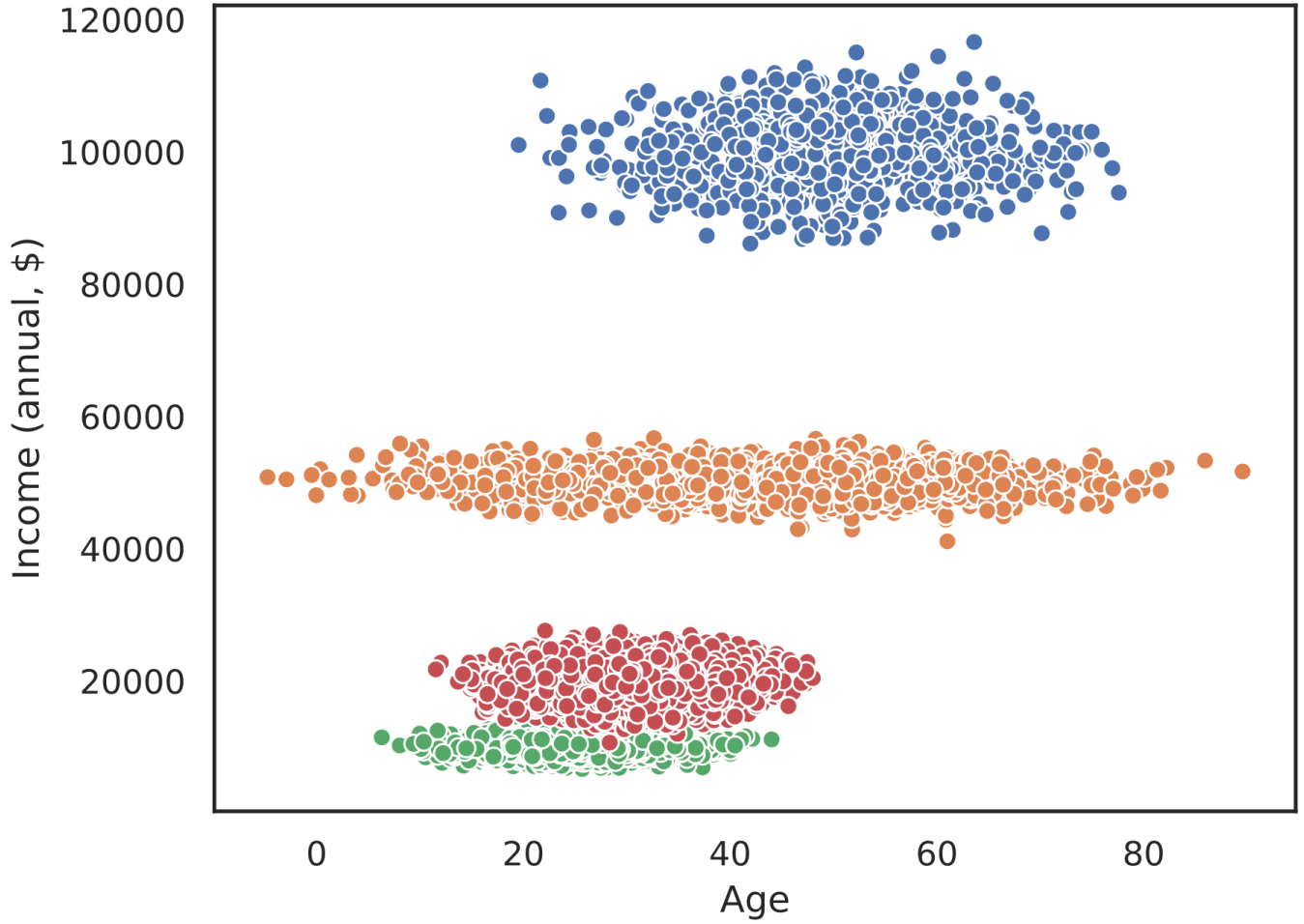
Sum-Product Networks – Sum-Node

- Mixture of two distributions over the same set of random variables
- Cluster and split samples, e.g. kNN-clustering
- Associated weight corresponds to relative size of the cluster

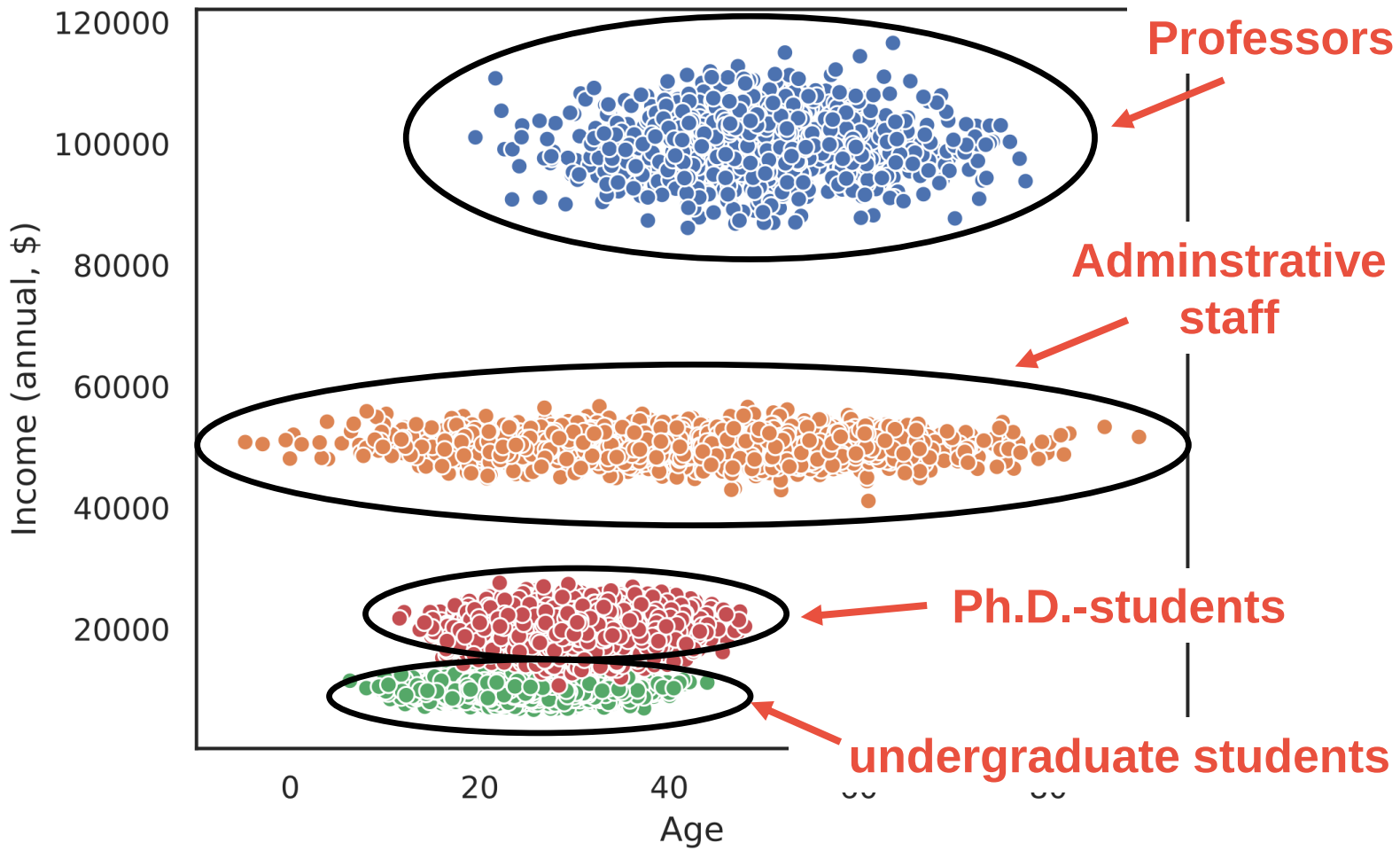




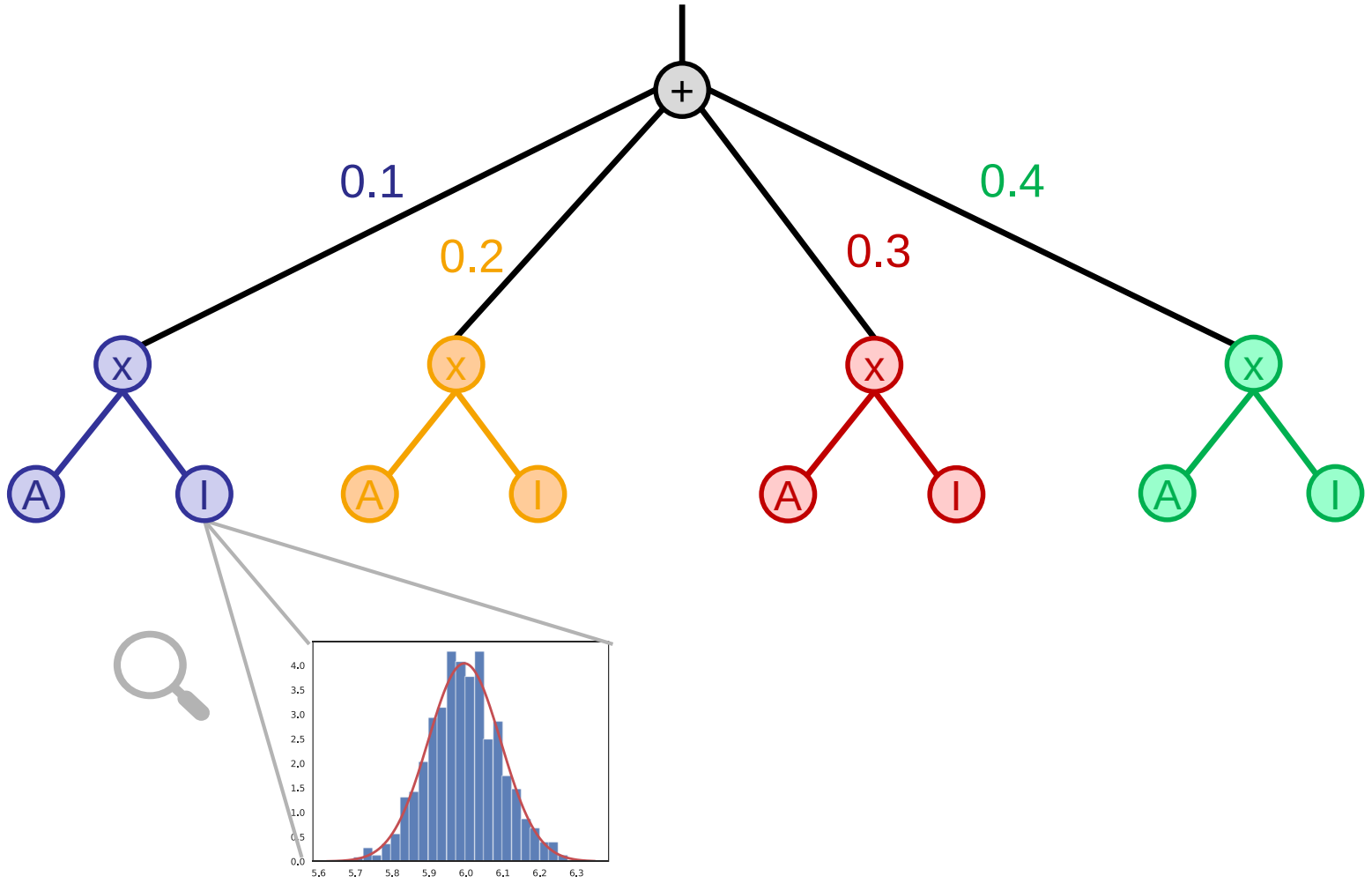
Sum-Product Networks – Example



Sum-Product Networks – Example



Sum-Product Networks – Example Network



Sum-Product Networks - Inference

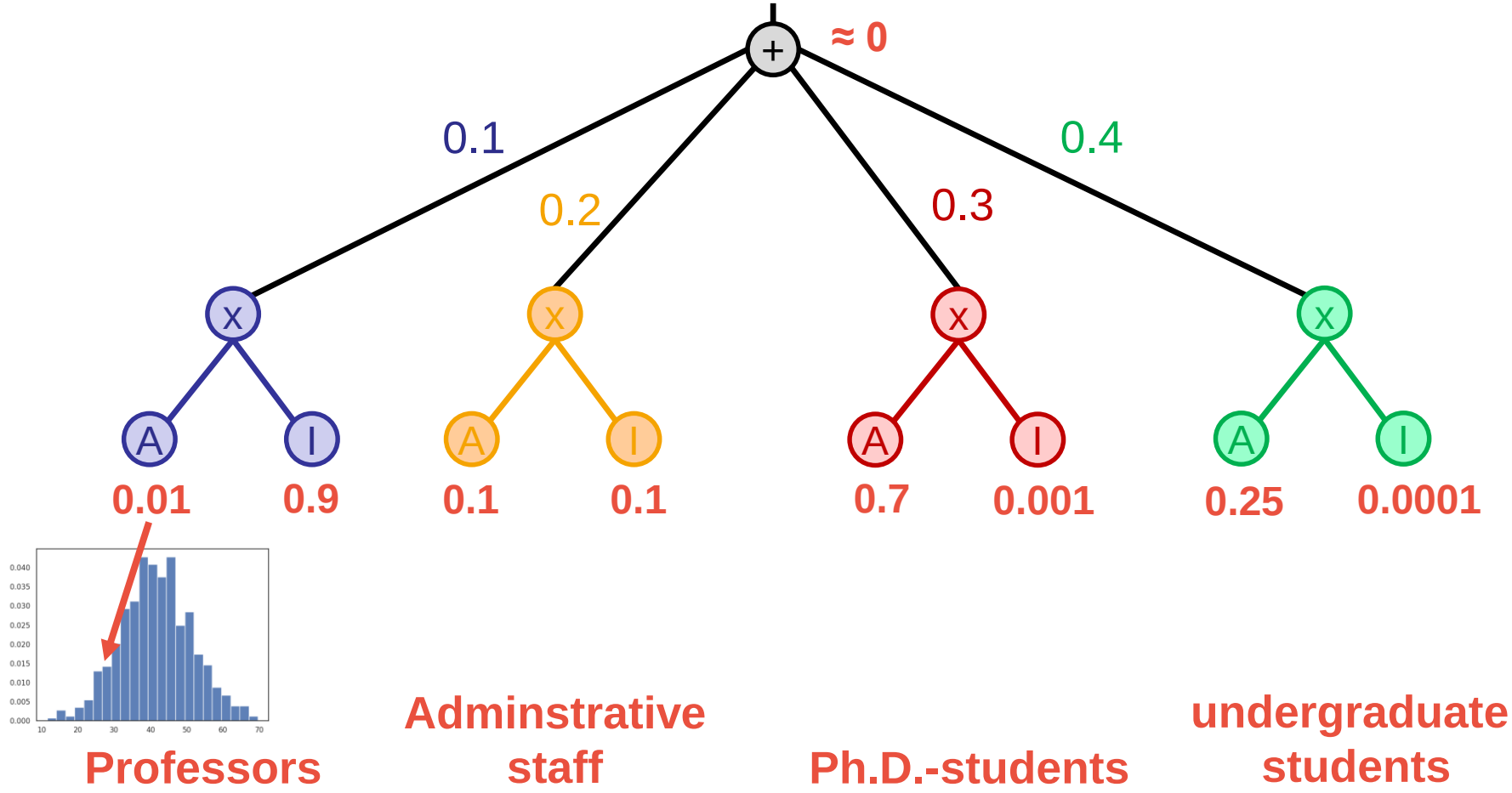
- Answer probabilistic queries & solve ML tasks
  - Probability of earning 100k$ at age 27: **P(A=27, I=100k$)**
  - Probability of earning 150k$: **P(I=150k$)** – marginalization
  - Add label {student, Ph.D.-student, admin, professor} as input variable, do classification based on information about age and income

- Inference is bottom-up evaluation of the SPN graph with (partial) evidence

- Some queries might require multiple passes, but always linear time

Sum-Product Networks – Example Network

Probability of earning 100k$ at age 27: **P(A=27, I=100k$)**



FPGA Inference Accelerator

TECHNISCHE UNIVERSITÄT DARMSTADT

- Automatic toolflow for FPGA acceleration of SPN inference developed in prior work [TPM2018, ICCD2018]
- Maps SPN graph to fully spatial, pipelined accelerator with AXI4-based, pipelined memory interface
- Throughput-oriented scenario, accelerate inference for batch of input queries
- Turn-key solution, heterogeneous system integration with TaPaSCo

Embedded Systems & Applications

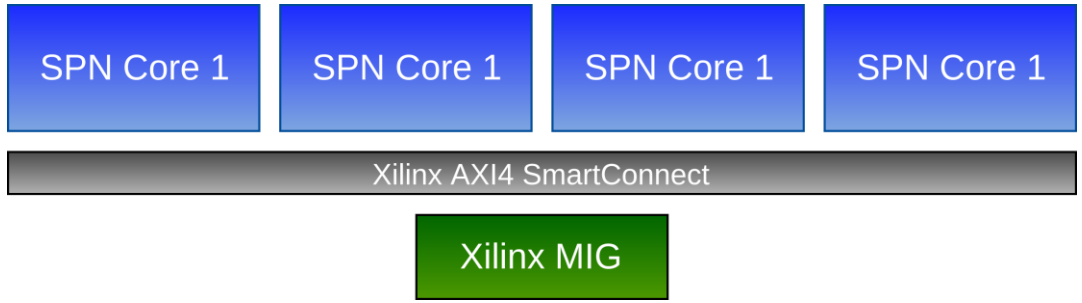
Memory Interface

- Existing memory interface aggressively optimized, occupies bus through long-running AXI4 bursts and many transfers in-flight
  - Potential deadlocks in multi-core scenario
  - No concurrent DMA-transfer between host and FPGA memory possible

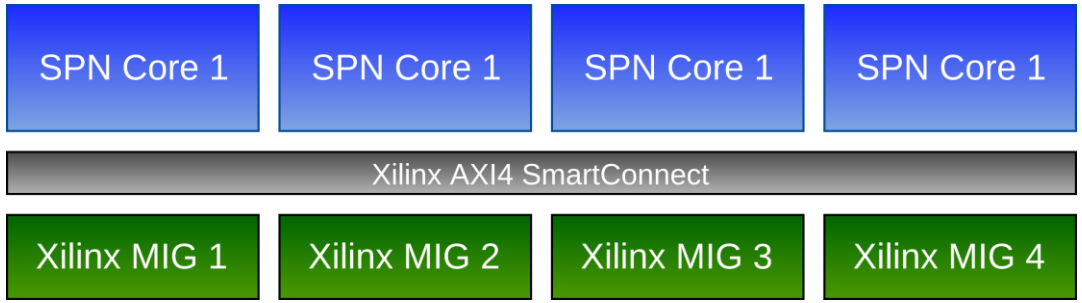
- **Solution:** Complete re-design of the memory interface
  - Strictly limit the number of outstanding transfers
  - Buffer result values, write back block-wise in short-running AXI4 burst transfer

Multi-core Architecture

- Size of VU9P FPGA allows for replication of accelerator
  - Baseline architecture: 1 compute unit, 1 memory channel
  - Multi-core, single memory: Up to 4 compute units, 1 memory channel



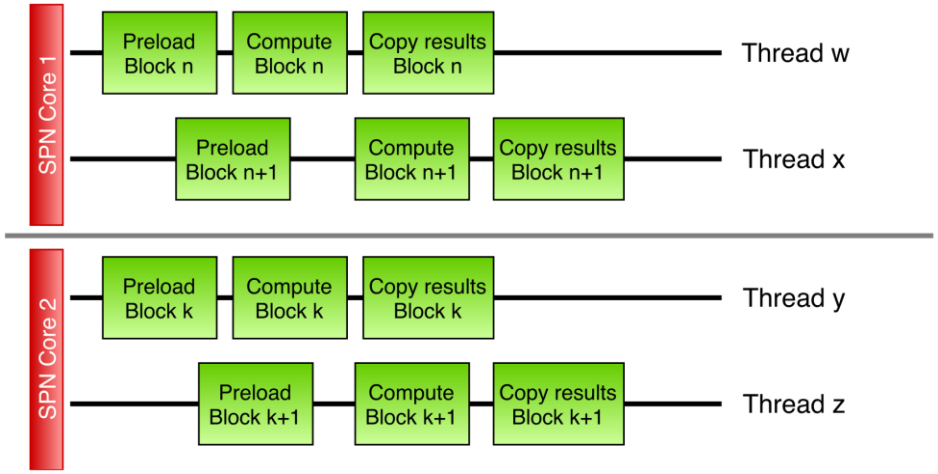
  - Multi-core, multi-memory: Up to 4 compute units, 4 memory channels



Multi-threaded Operation

- Low to moderate computational density for SPN inference, data-transfer overhead significant

- **Solution:** Split computation into blocks, overlap computation and data-transfer to/from host with multiple threads on host-side



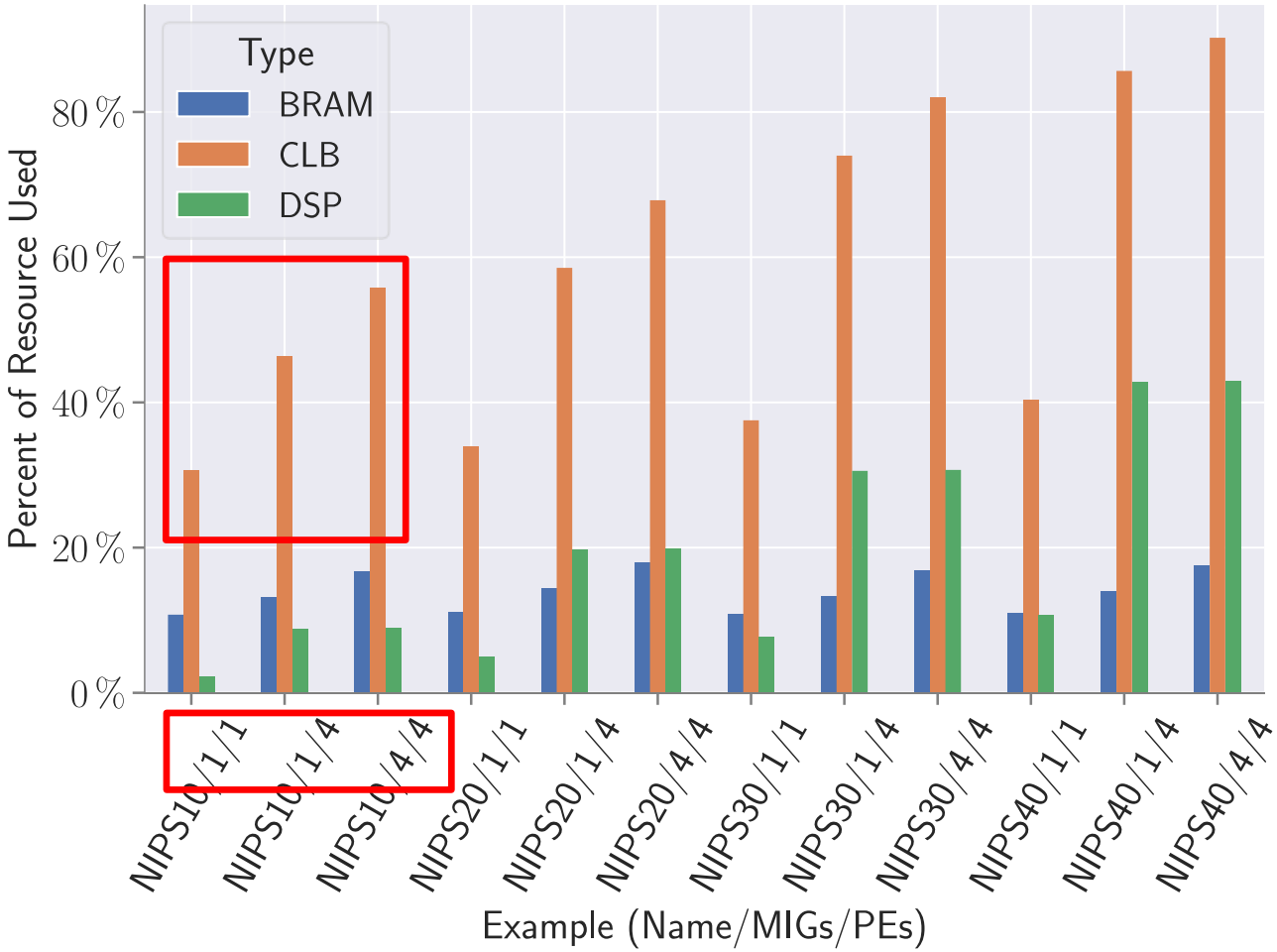
Evaluation

TECHNISCHE UNIVERSITÄT DARMSTADT

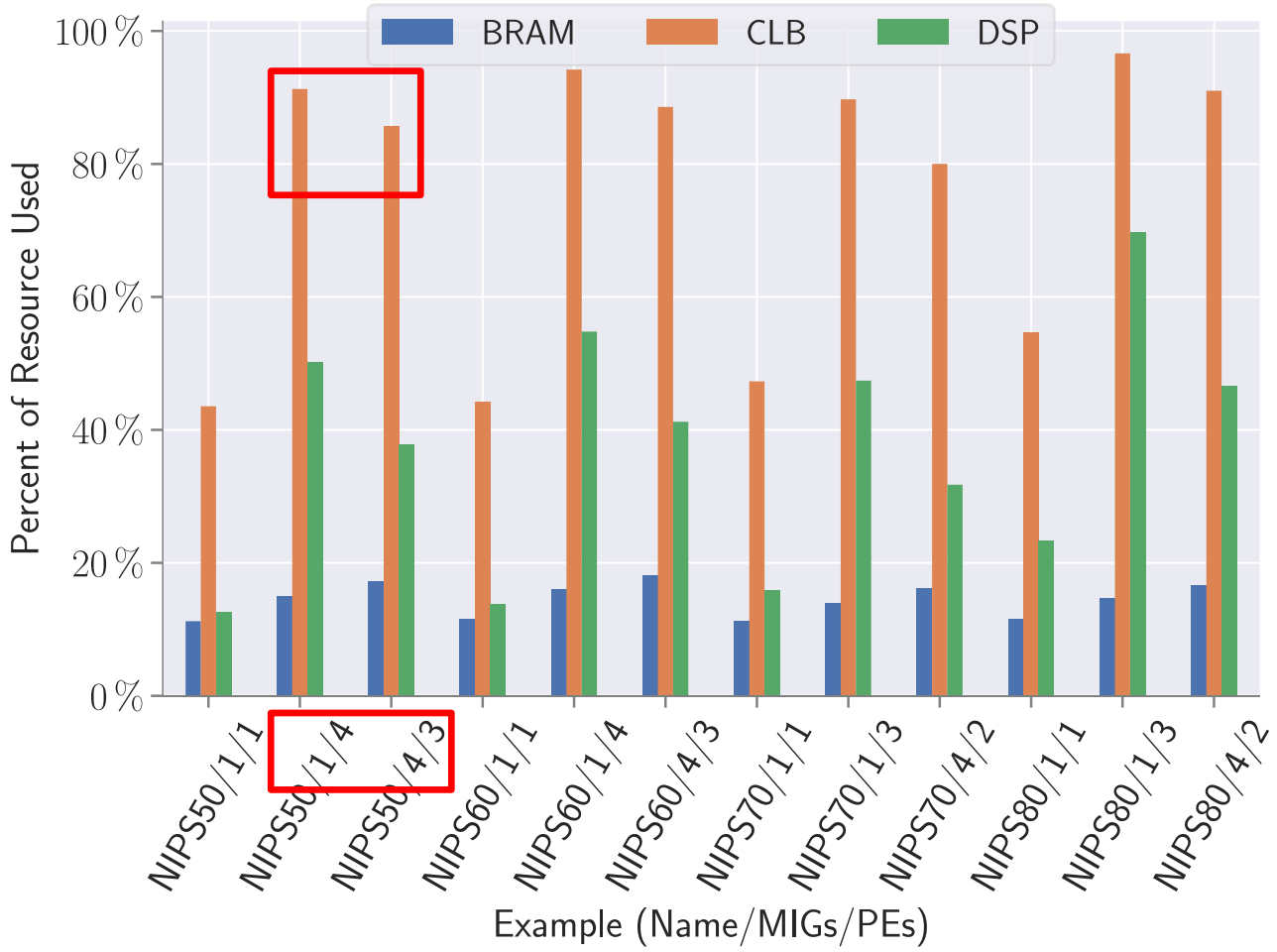
- Evaluation with 8 different benchmarks from the NeurIPS corpus
- FPGA implementation for AWS F1 for all three architectures
- CPU comparison with generated C++ code on 12-core Xeon E5-2680v3
- GPU comparison with optimized CUDA code on Nvidia V100 (AWS EC2)
- Measure end-to-end throughput, including data-transfer from/to host

Embedded Systems & Applications

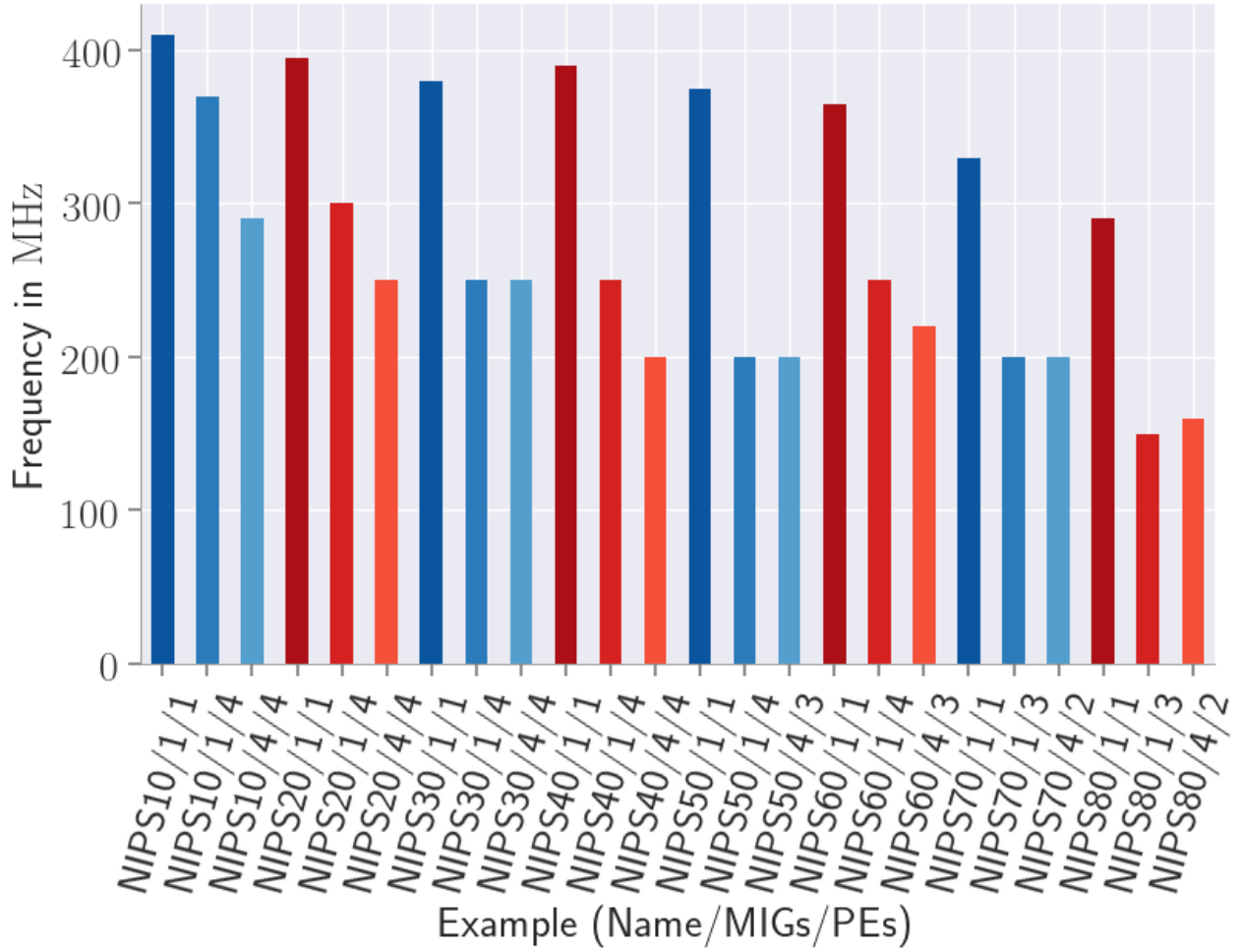
FPGA Implementation Results



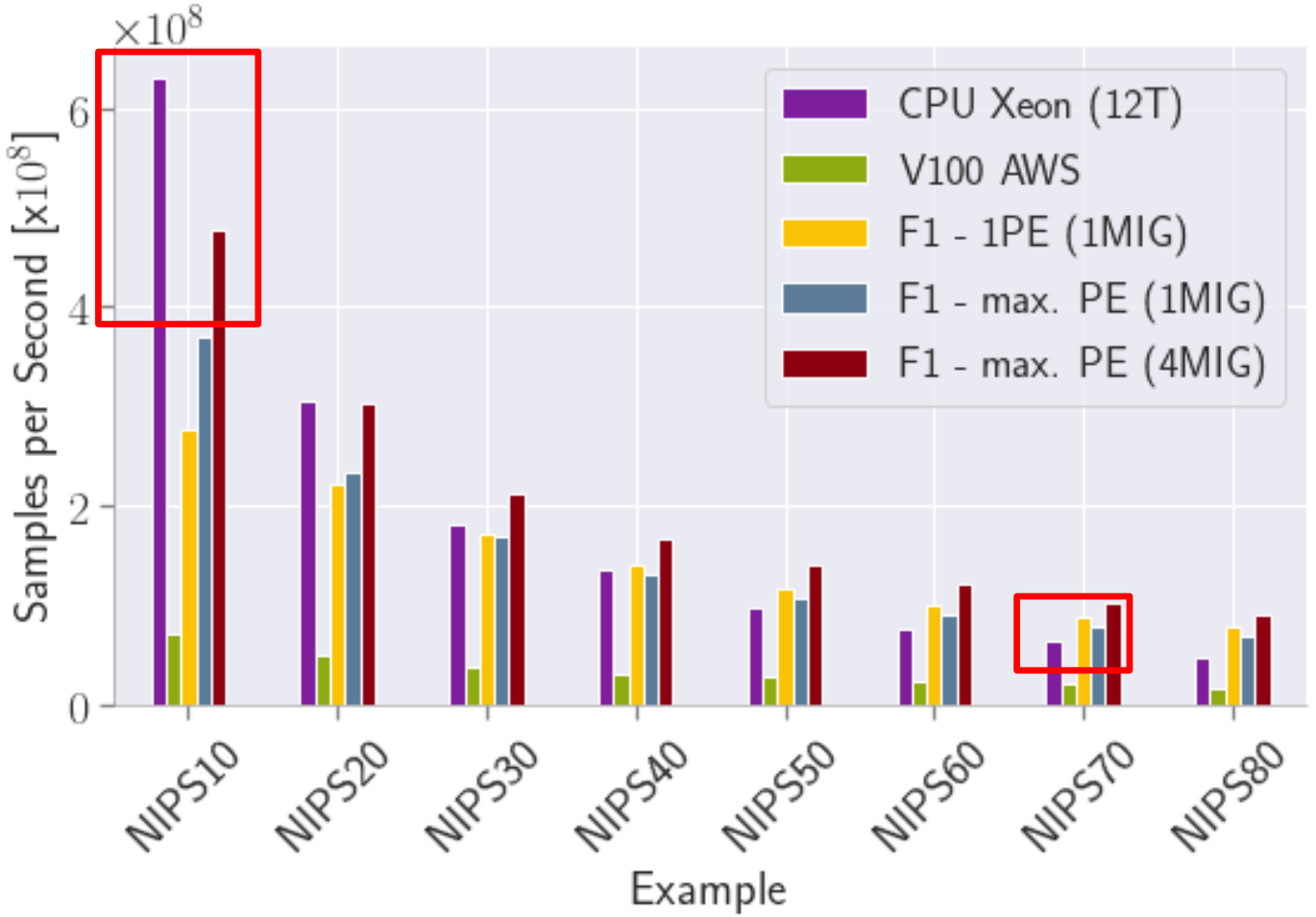
FPGA Implementation Results



FPGA Implementation Results



Performance Comparison



Conclusion

TECHNISCHE UNIVERSITÄT DARMSTADT

- Case study demonstrates ease-of-use of the TaPaSCo design flow to generate heterogeneous accelerator system for the Amazon AWS EC2 F1 instances
- Multi-core architecture with multi-threaded software interface significantly improves throughput for SPN inference
- Up to 1.9x speedup over 12-core Xeon CPU
- Up to 6.6x speedup over Nvidia Tesla V100 GPU – due to low arithmetic intensity

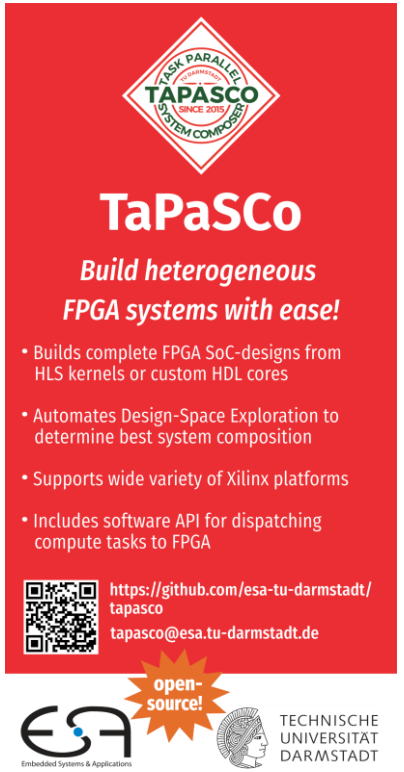
Embedded Systems & Applications

TECHNISCHE UNIVERSITÄT DARMSTADT

Start to build your own AWS F1 accelerator system using TaPaSCo!

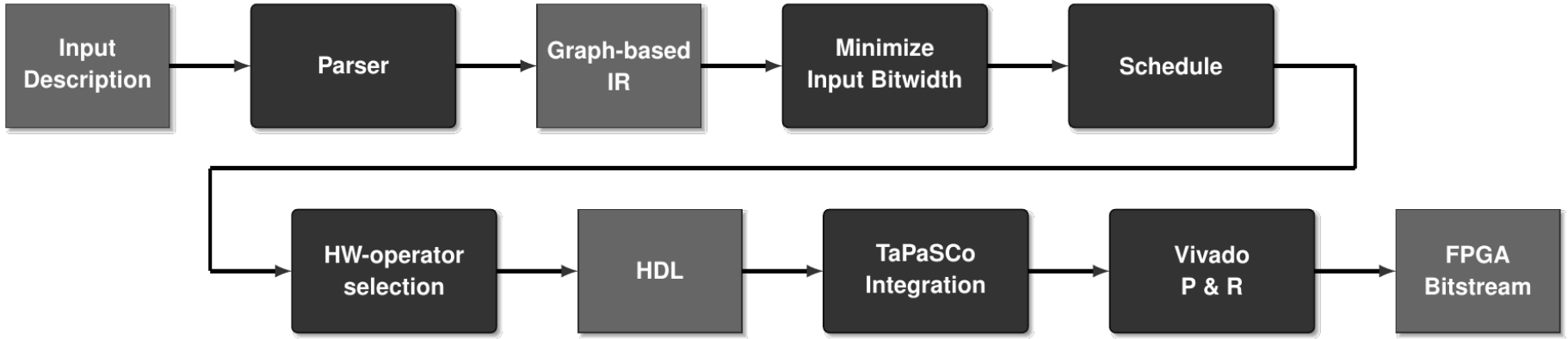
Download TaPaSCo from Github:

`github.com/esa-tu-darmstadt/tapasco`



Embedded Systems & Applications

Existing FPGA Acceleration Toolflow



Existing FPGA Accelerator Core