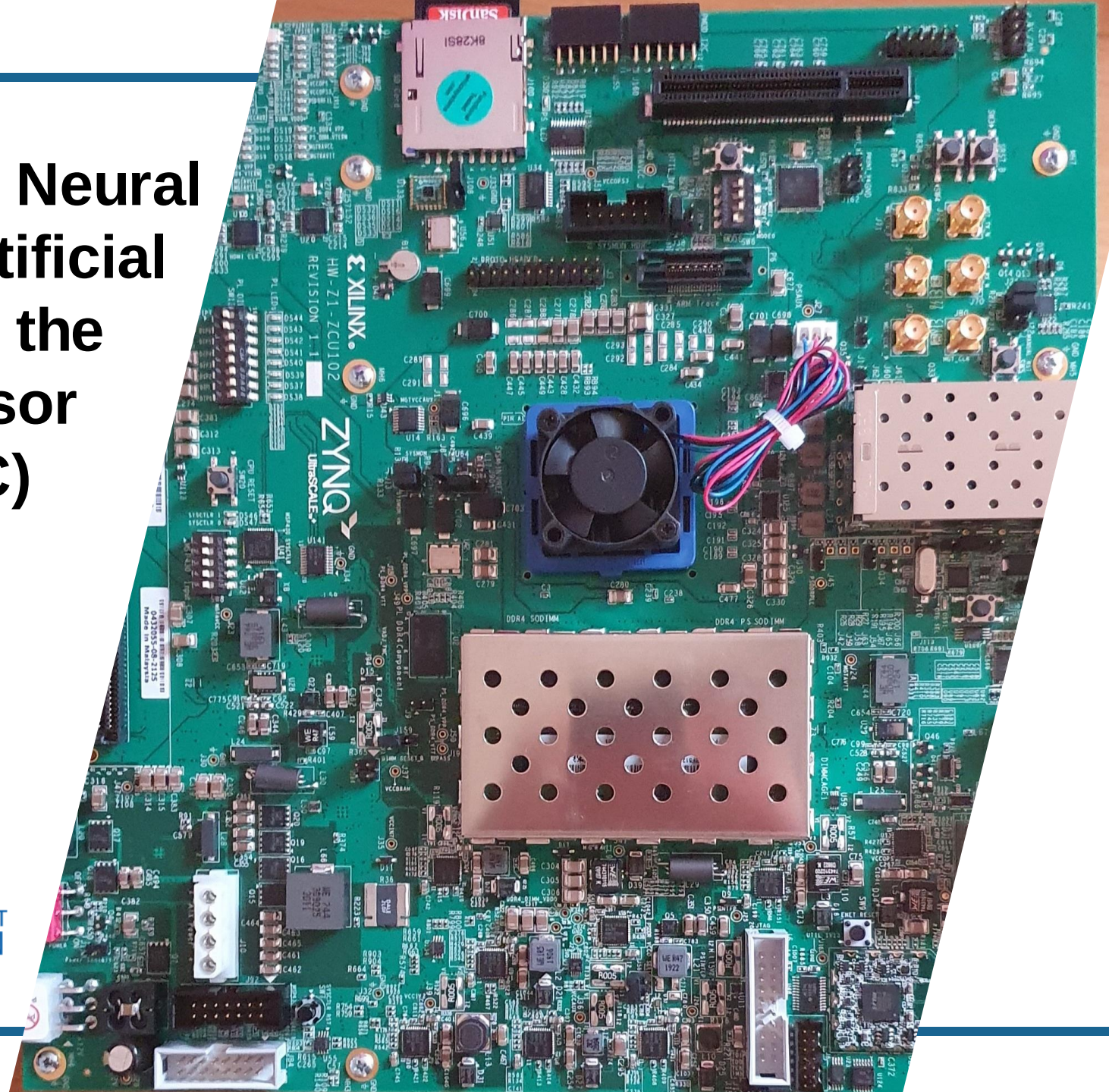
Latency optimized Deep Neural Networks (DNNs): An Artificial Intelligence approach at the Edge using Multiprocessor System on Chip (MPSoC)

Seyed Nima Omidsajedi, Rekha Reddy, Jianming Yi,
Jan Herbst, Christoph Lipps and Hans Dieter Schotten

25. VDE/ITG Fachtagung Mobilkommunikation
3-4 November 2021
Osnabrück



TECHNISCHE UNIVERSITÄT
KAISERSLAUTERN

Motivations

- **Increasing** the number of **mobile devices** significantly in the last decades.
- Significant data transmission between an end device and Cloud, arise problems:
  - **Reliability** and **Bandwidth restriction**.
- The need for **computing** a large portion of data **near to the client-side**.
- Demands for **smart** and **secure** platforms.
- Demands for **low latency** computational units.
- **Energy-efficient** design solutions for mobile devices.

Goals

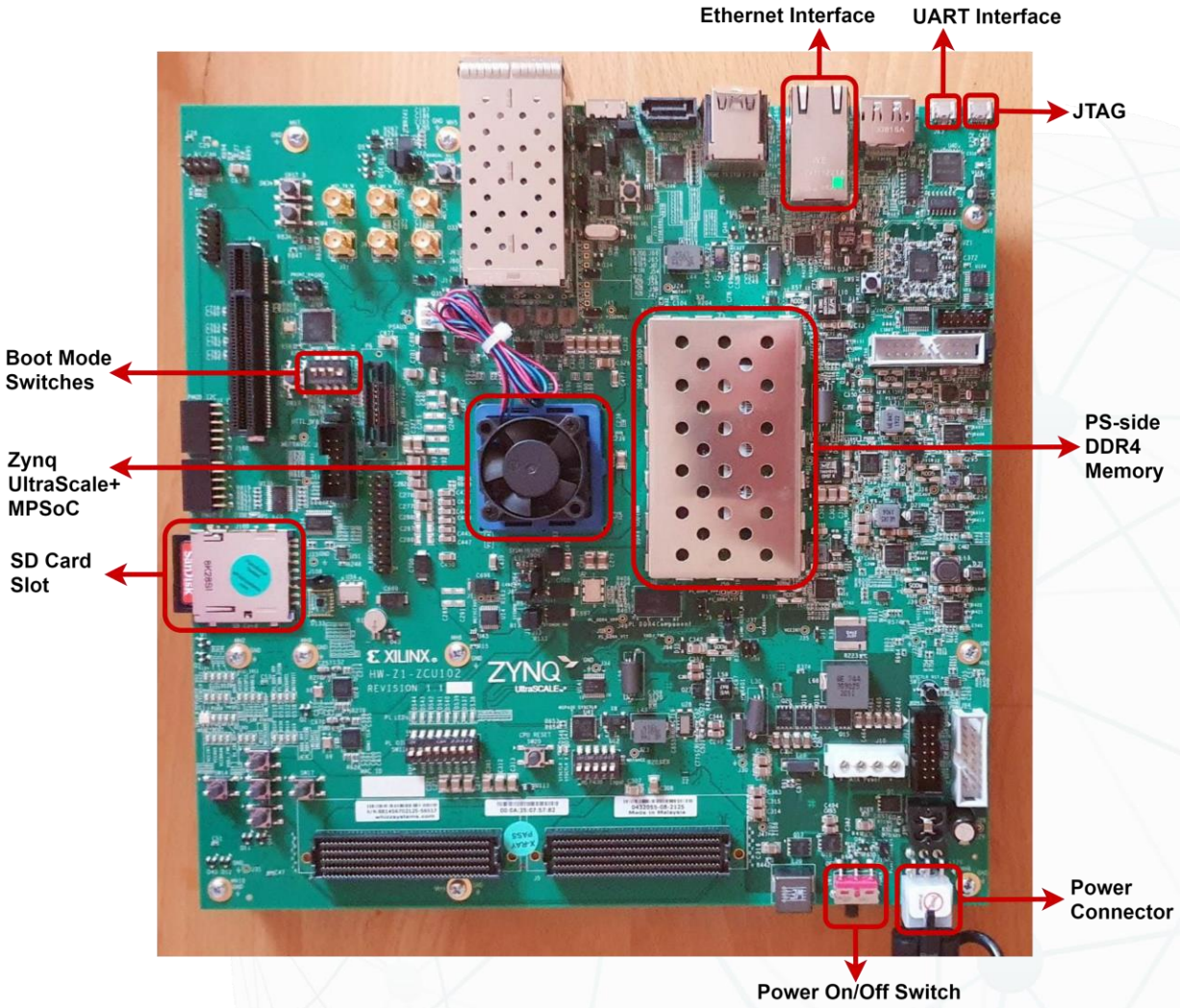
- Design a **low latency** and **energy-efficient** AI solution using Edge devices.
- Using **embedded FPGAs** as a hybrid computation tool for implementing DNNs.
- **Comparing** the implementation results on an **Edge device vs GPU Cluster**.
- Propose an **expandable** hardware accelerator design for selected Edge devices:
  - Devices with Zynq / Zynq UltraScale+ architectures.

MPSoC Configurations (as Edge device)

Processing System (PS):	
APU	Quad-core ARM Cortex-A53
RPU	Dual-core ARM Cortex-R5
GPU	Mali-400

Programmable Logic (PL):	
System Logic Cells	600 K
Block RAM (BRAM)	32.1 MB
DSP Slices	2520

Memory types:	
PS-Side DDR4	4GB 64-bit
PL-Side DDR4	512MB 16-bit



GPU Cluster Configurations (as Cloud)

The implementation is tested over 2 different GPU servers (GTX 1080Ti, RTX A6000) with 2 and 4 GPUs along with 2 CPUs.

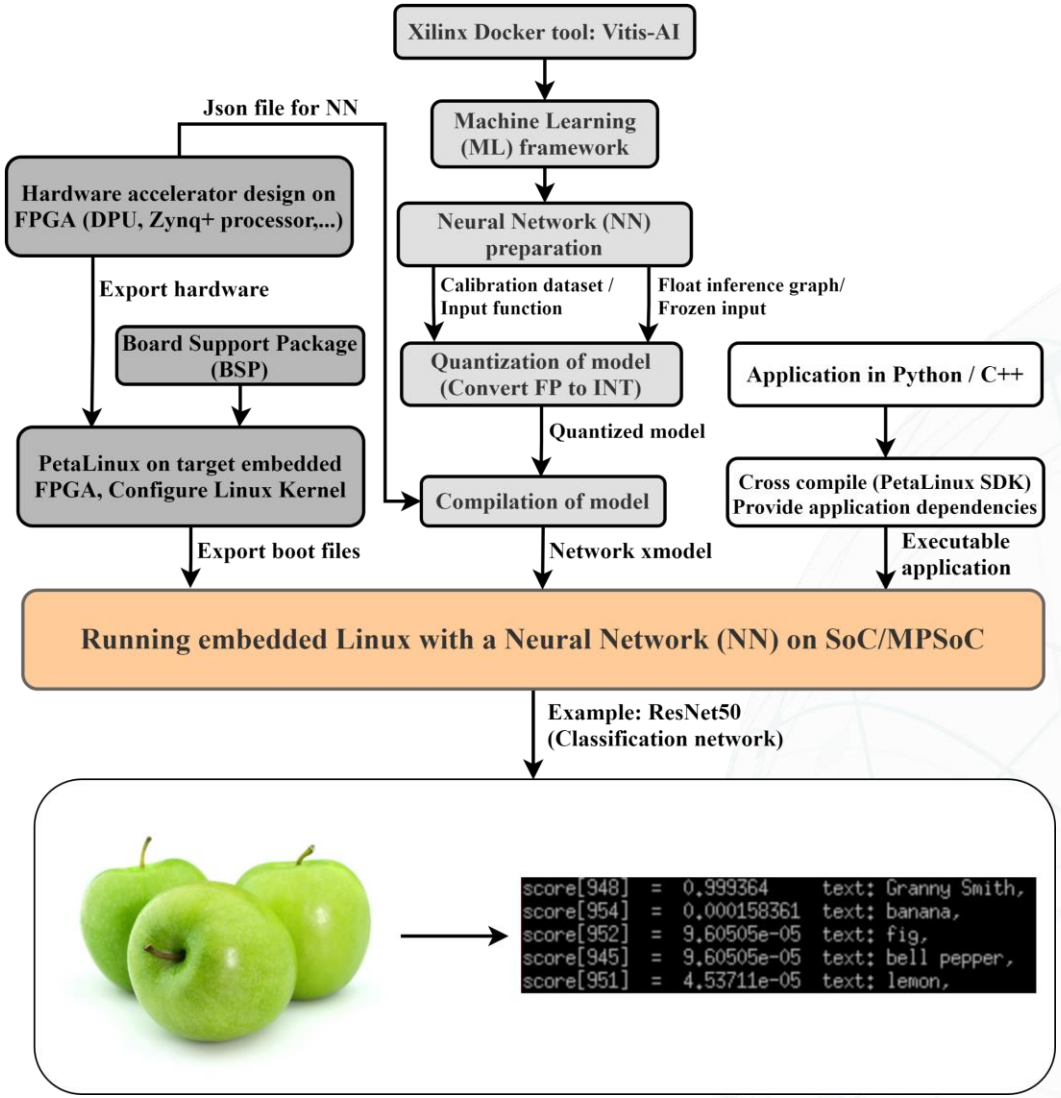
GTX 1080Ti:	
Architecture	Pascal
GPU Memory(GB)	11
GPU per node	8
CPU per GPU	5-9

CPU config:	
Model	Intel(R) Xeon CPU E5-2630 v4
CPU MHz	2856.359

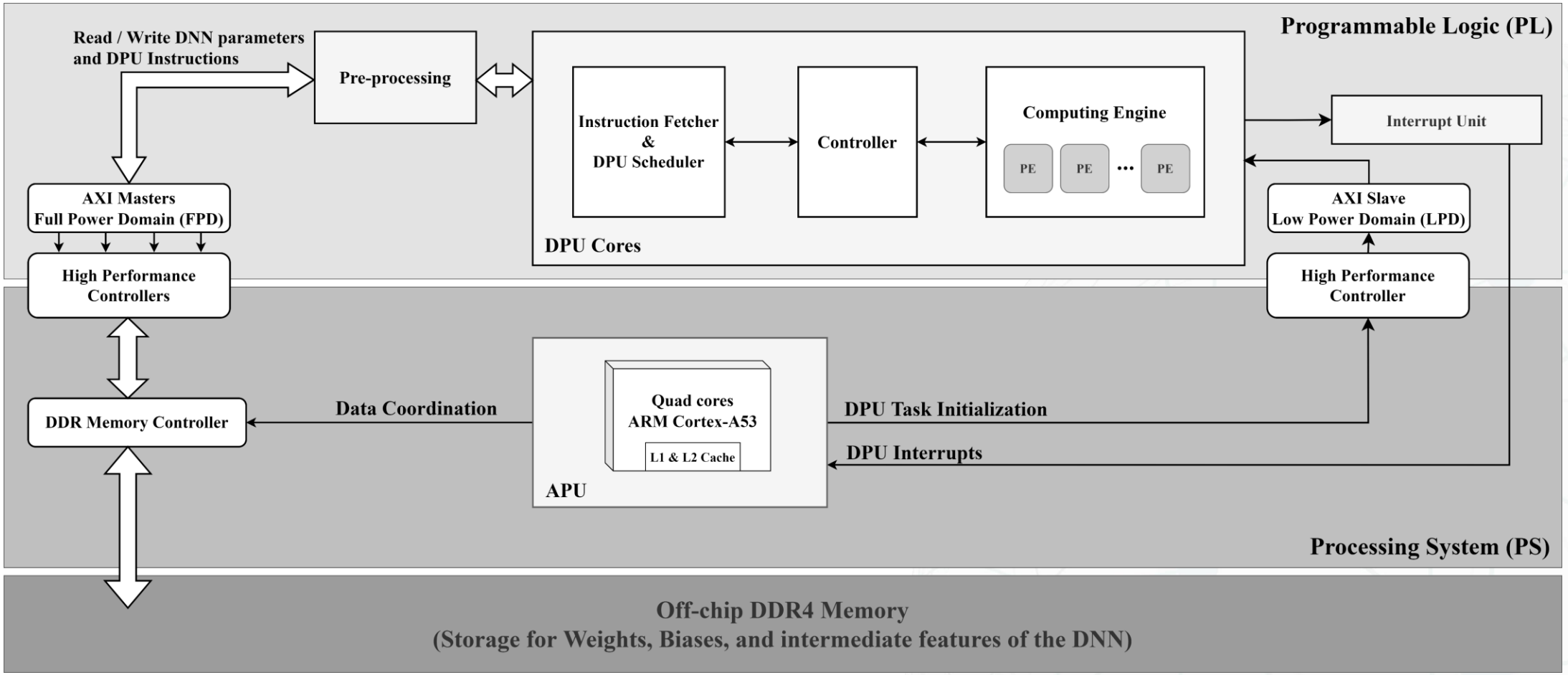
RTX A6000:	
Architecture	Ampere
GPU Memory(GB)	48
GPU per node	8
CPU per GPU	12

CPU config:	
Model	AMD EPYC 7F72 24-Core
CPU MHz	2854.292

DNN implementation process on MPSoC (as Edge device)



Hardware accelerator design using FPGA-ARM (Edge)



ResNet50 DNN implementation results (on Edge)

Power, Latency and Throughput of the Edge device (ZCU102):

DPU Cores/Arch	Power (W)	Latency (S)	Throughput (Img/S)
1/B4096	9.97	22.59	88.5
2/B4096	16.29	12.40	161.2
3/B4096	22.86	9.73	205.4
4/B2304	21.98	11.68	171.1

DPU architecture naming:

- The maximum number of operations in 1 Clock Cycle.
- 1/B4096: 1 DPU core with 4096 peak operations.

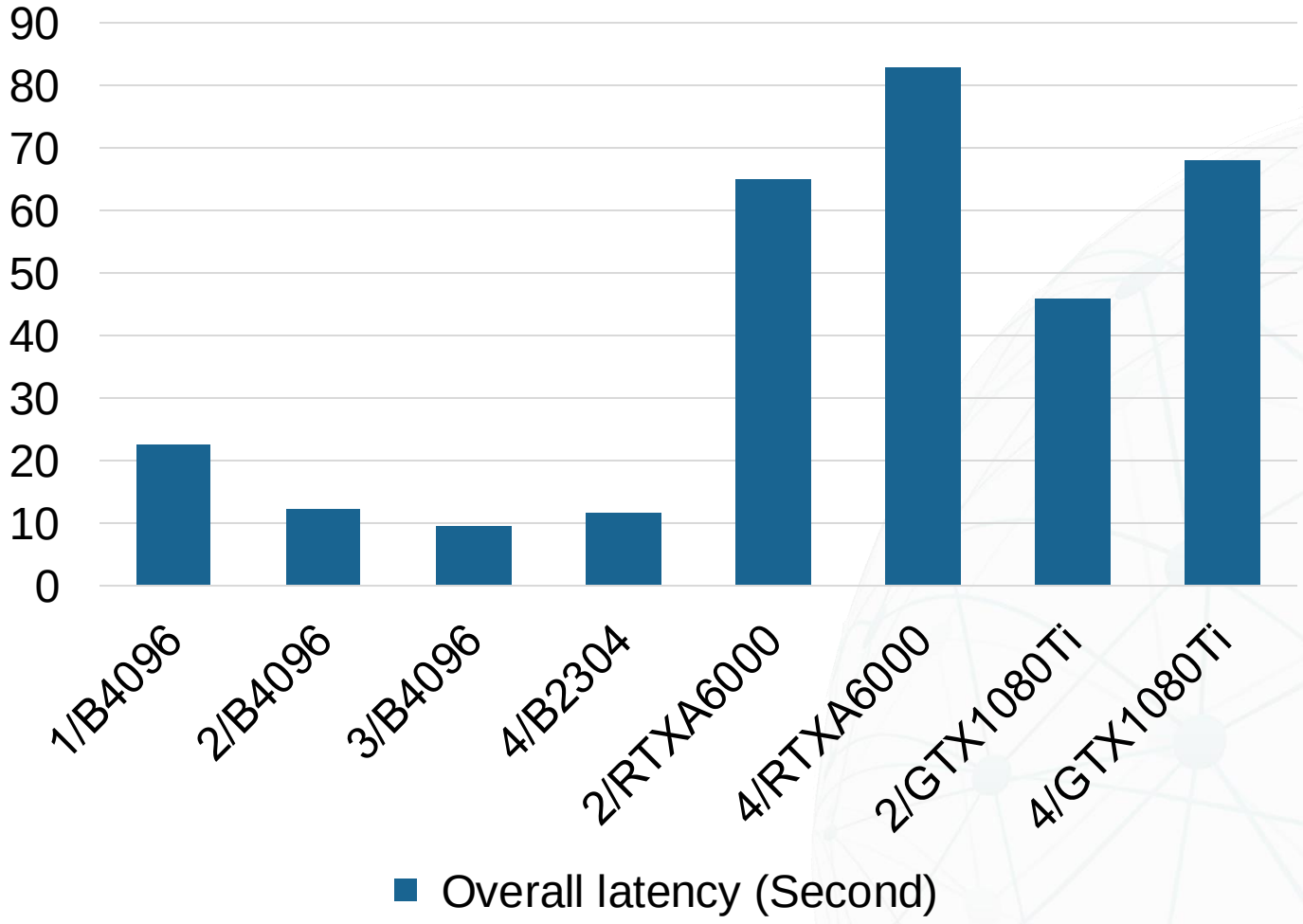
ResNet50 DNN implementation results (on Cloud)

Power, Latency and Throughput of the GPU Cluster:

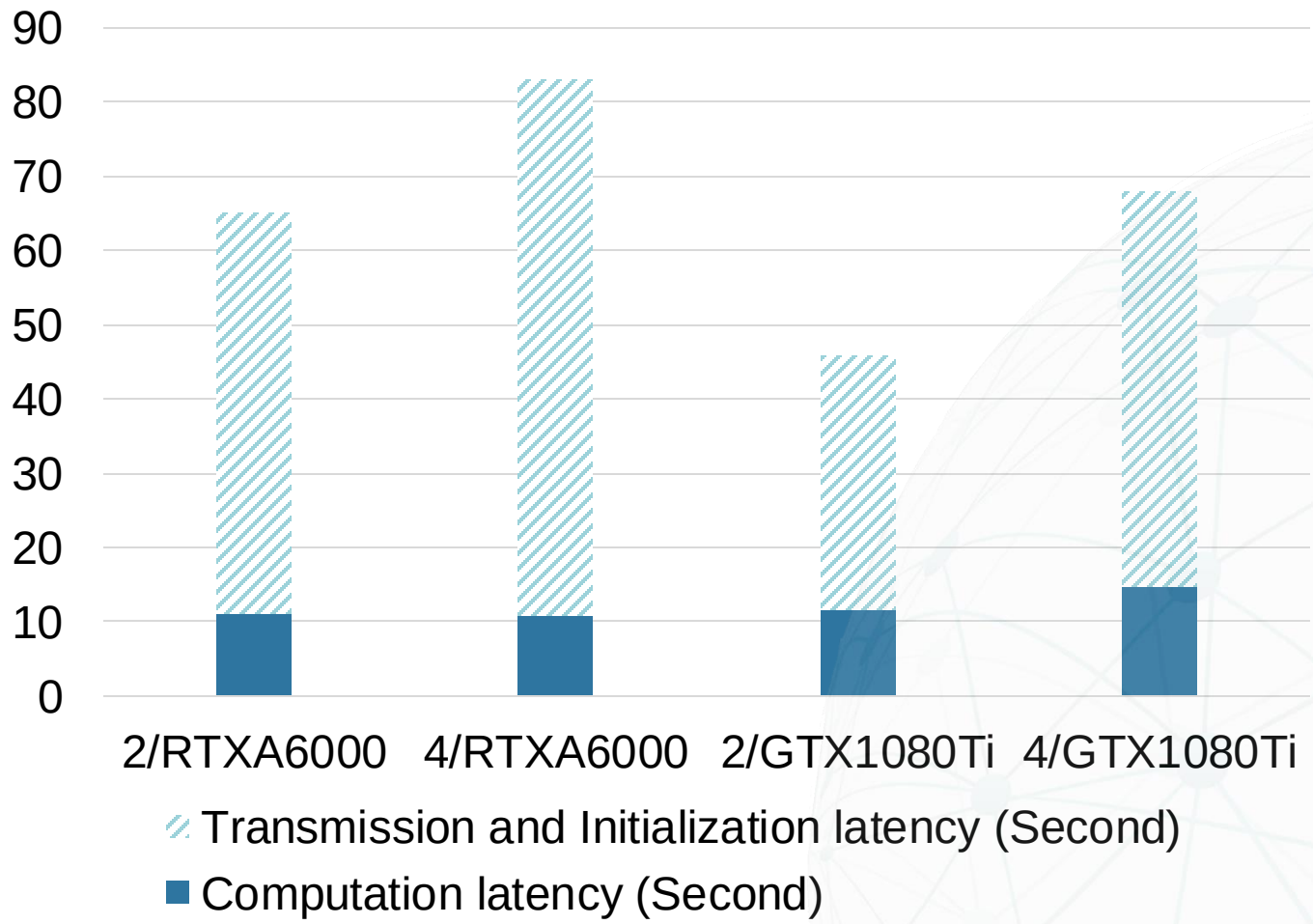
Num/GPU	Power (W)	Latency (S)	Throughput (Img/S)
2/RTX A6000	215.8	65	180
4/RTX A6000	174.4	83	187
2/GTX 1080Ti	95.9	46	175
4/GTX 1080Ti	138.4	68	137

- GPU clusters can achieve higher throughput, but with a much higher power consumption.
- Increasing the batch size → Higher GPU utilization → Higher throughput:
  - Batch size from 8 to 200.
  - 4/GTX 1080Ti: Throughput from 137 to 751.49 Img/S → Power from 138.4 to 747 Watts
  - 4/RTX A6000: Throughput from 187 to 1886.13 Img/S → Power from 174.4 to 899 Watts

Comparison result (Edge vs Cloud)



Comparison result (Edge vs Cloud)



Conclusion

- Proposed a DNN implementation strategy on Edge devices using MPSoCs.
- Depicting superiority of using Edge devices vs Cloud for AI inferences:
  - Latency, Energy efficiency.
- Using FPGA logics and ARM processors for designing a hardware accelerator:
  - Utilizing Xilinx Deep learning Processing Unit (DPU) soft IP.
  - Built an expandable design for selected Zynq/ Zynq UltraScale+ devices.

Future Work

- Implementing multiple DNNs on SoCs/MPSoCs.
- The possibility of using Xilinx Versal boards.
  - ➢ With dedicated AI engines instead of soft IP cores.
- DNN algorithm optimization for FPGA implementation.

Question & Answer

THANK YOU

{Seyednima.Omidsajedi; Rekha.Reddy}@dfki.de