TECHNISCHE UNIVERSITÄT DRESDEN

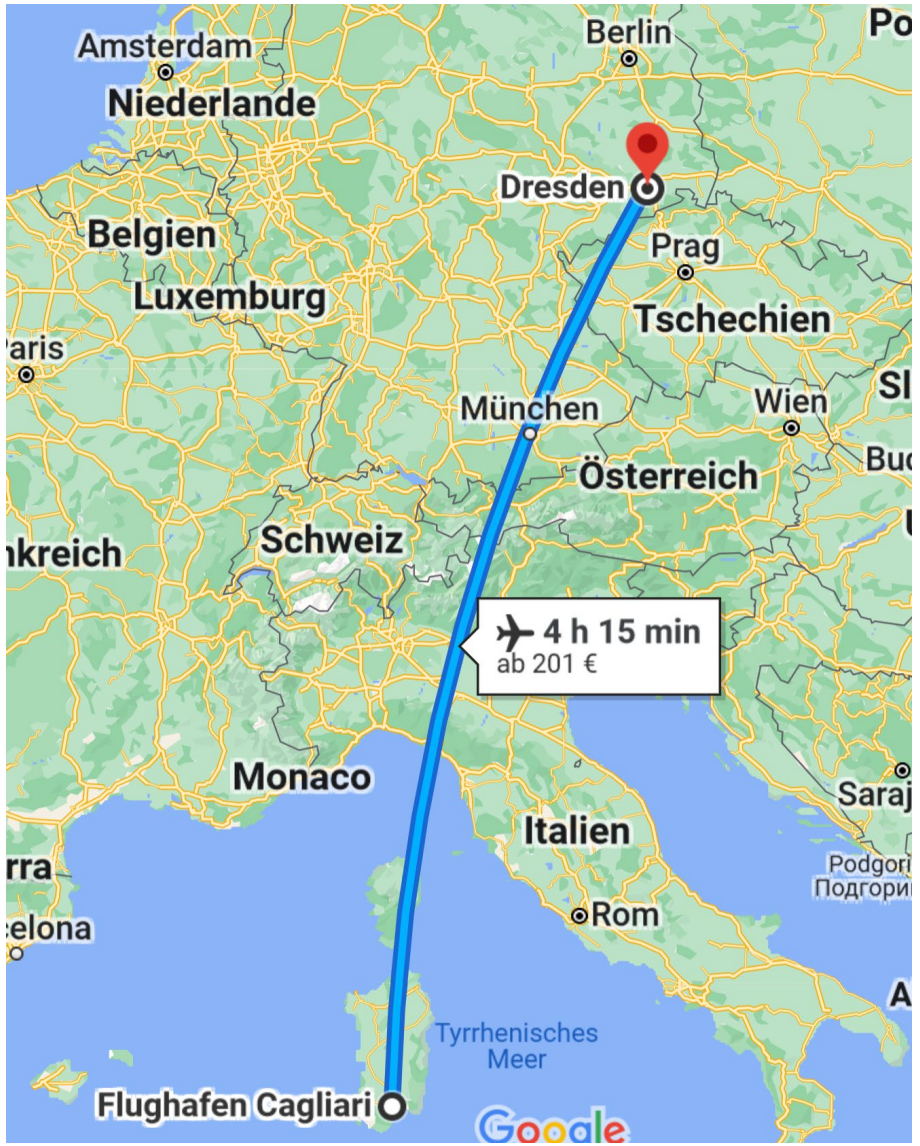
DRESDEN concept

Diana Göhringer
Chair of Adaptive Dynamic Systems
TU Dresden

Adaptive Computing Architectures for CPS

CPS Summer School 2022

Dresden


Amsterdam
Niederlande
Berlin
Dresden
Belgien
Prag
Luxemburg
Tschechien
München
Wien
Österreich
Schweiz
4 h 15 min
ab 201 €
Monaco
Italien
Rom
Tyrrhenisches Meer
Flughafen Cagliari
Google

TU Dresden Campus and Faculty of Computer Science

Chair of Adaptive Dynamic Systems: Research Topics and Projects

ADS

Domain-Specific Computer Architectures
- Processors
- Accelerators
- Network-on-Chip
- Memory

Hardware/Software Co-Design
- Modeling (e.g. UML)
- HW/SW Codesign of applications or middleware (ROS, RTOS)
- High Level Synthesis

Runtime Reconfigurable Systems
- Virtualization
- Self-adaptation
- Dynamic and Partial Reconfiguration

Programming Methods and Tools
- Design Space Exploration
- Application Partitioning and Mapping
- LLVM
- Processing-in-Memory

DAKORE · marie SFB/TRR 196 · 6G-life · CeTI · EUFRATE · ScaDS.AI DRESDEN LEIPZIG

Evaluation with Application Scenarios from Adaptive Dynamic Systems, e.g.
Image and Signal Processing Algorithms from Robotics, AAL, I4.0, Automotive

Outline

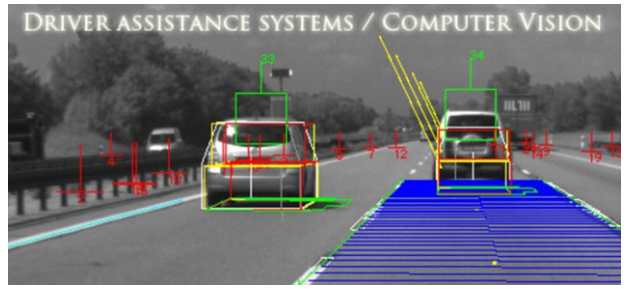
- Introduction and Motivation
- Reconfigurable Domain-Specific Computer Architectures
  - System Overview and Processing Elements
  - Reconfiguration Management
  - Network-on-Chip
- Design / Programming Methodology
  - High Level Synthesis Library: HiFlipVX
  - Application Partitioning and Mapping
- Application Examples in Relation to Funded Research Projects
- Conclusion and Outlook

Introduction and Motivation

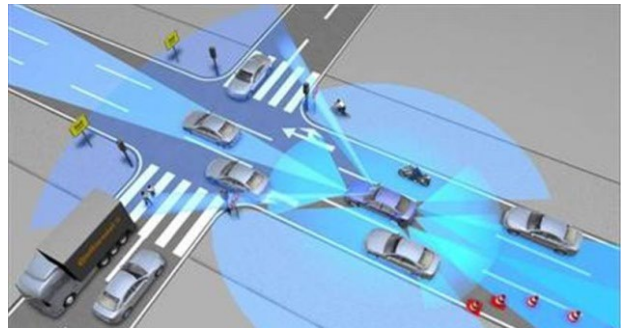
Problem:

- Increasing complexity
- Real-time requirements
- Low power/energy consumption (batteries, cooling)
- Dynamic adaption to changing environments
- → Cannot be solved anymore by increasing the clock frequency, as $P_{dyn} \sim C \times V^2 \times f$

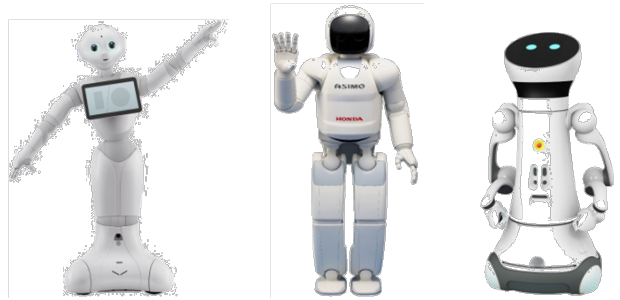
Possible solution: Domain-specific Computer Architectures



http://www.rcs.ei.tum.de/forschung/driver-assistance/

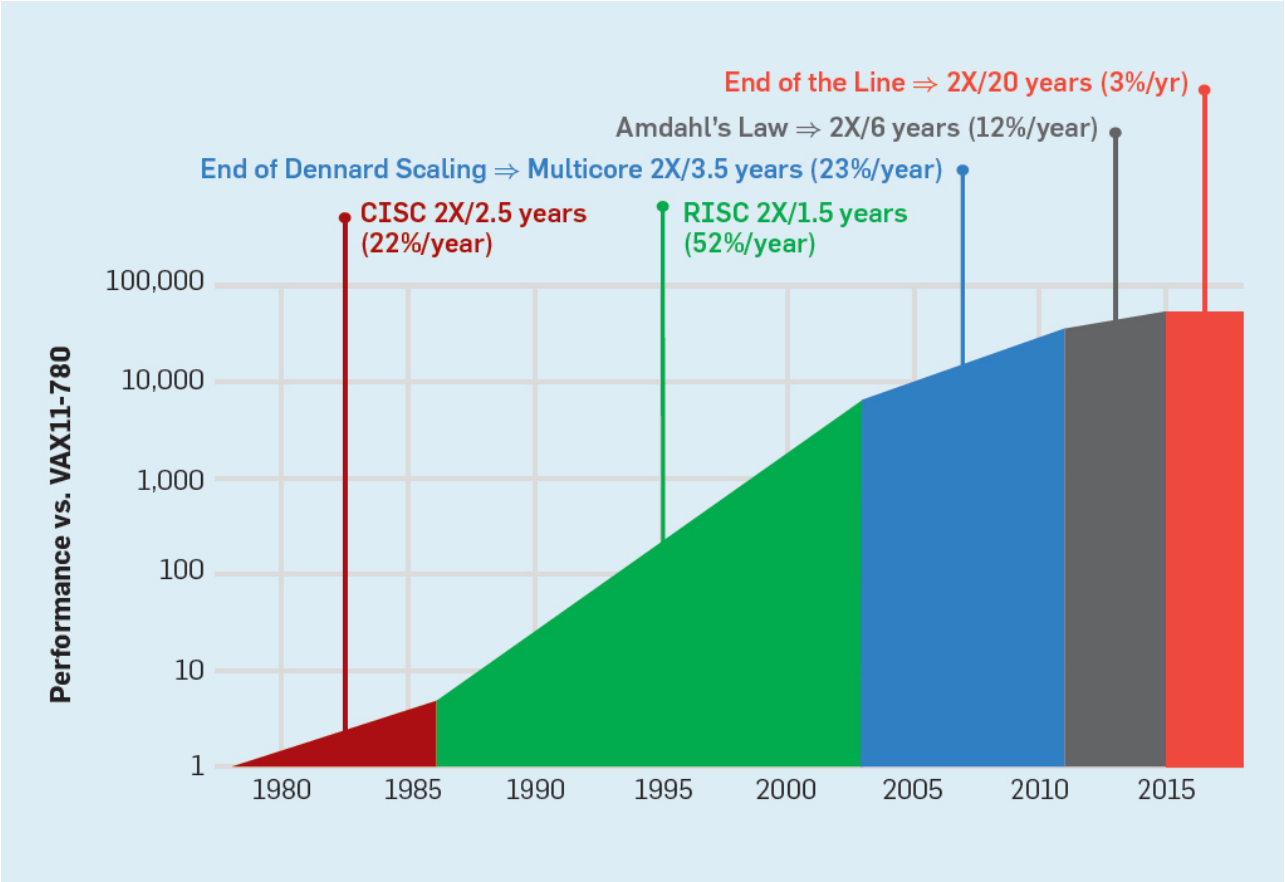


http:// www.asdreports.com/news-10595/key-players-advanced-driver-assistance-systems-adas-market-north-america-20152019



www.ald.softbankrobotics.com asimo.honda.com www.care-o-bot-4.de

Introduction and Motivation



Problem:

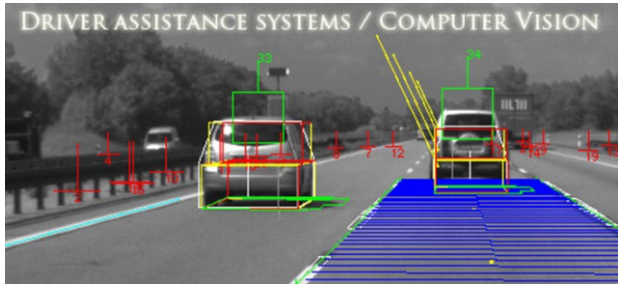
Performance of processor cores reached a plateau

Solution:

Domain-specific Computing Architectures

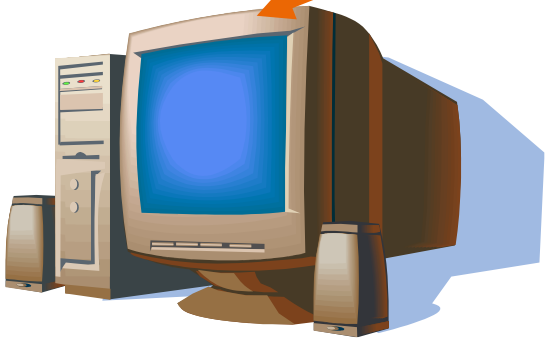
Figure: Norman P. Jouppi, Cliff Young, Nishant Patil, David Patterson: A Domain-Specific Architecture for Deep Neural Networks. Communications of the ACM, September 2018, Vol. 61 No. 9, Pages 50-59

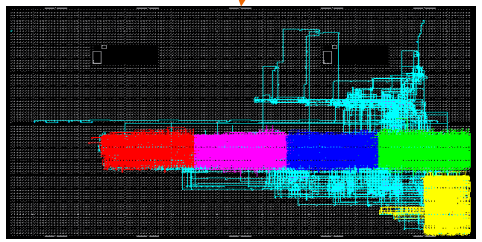
Example: Which architecture to choose?

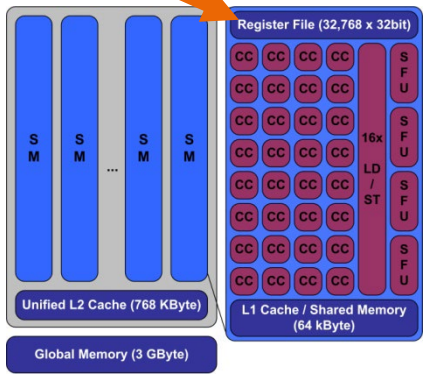


http://www.rcs.ei.tum.de/forschung/driver-assistance/

Applications → Platforms



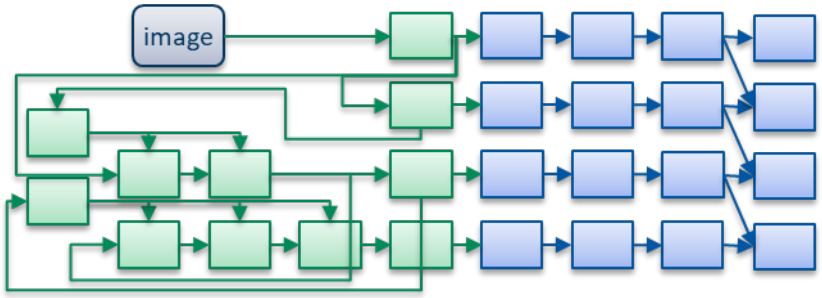




CPU **FPGA** **GPU**

Comparison of CPU, GPU and FPGA using OpenCL and AKAZE

- Evaluated for a 720p (1080p) resolution

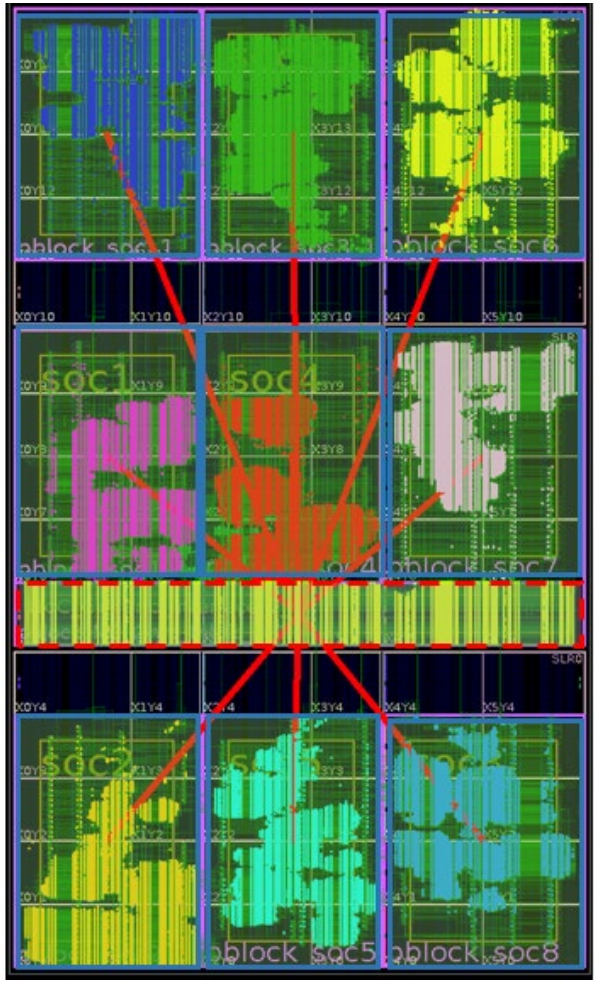


Device	Vendor	Model	Fab (nm)	Bandwidth (GB/s)
FPGA	Xilinx	Virtex-7 XC7VX690T	28	10.67
GPU	NVIDIA	GTX 780	28	288
CPU	Intel	Core-i7 4770k	22	25,6

	Speed Up	Power (watts)	Energy (mJ)
Single Thread	**1** (1)	**23.8**	**1907**
CPU OpenMP	**3.40** (3.42)	**65.7**	**1528**
CPU OpenCL	**5.15** (4.87)	**66.9**	**1041**
GPU OpenCL	**37.58** (42.77)	**222.7**	**475**
FPGA OpenCL	**55.14** (62.77)	**30.5**	**45**

Kalms L, Göhringer D (2017) Exploration of OpenCL for FPGAs using SDAccel and Comparison to GPUs and Multicore CPUs.In Proc. of the International Conference on Field Programmable Logic and Applications (FPL).

FPGA Architectures



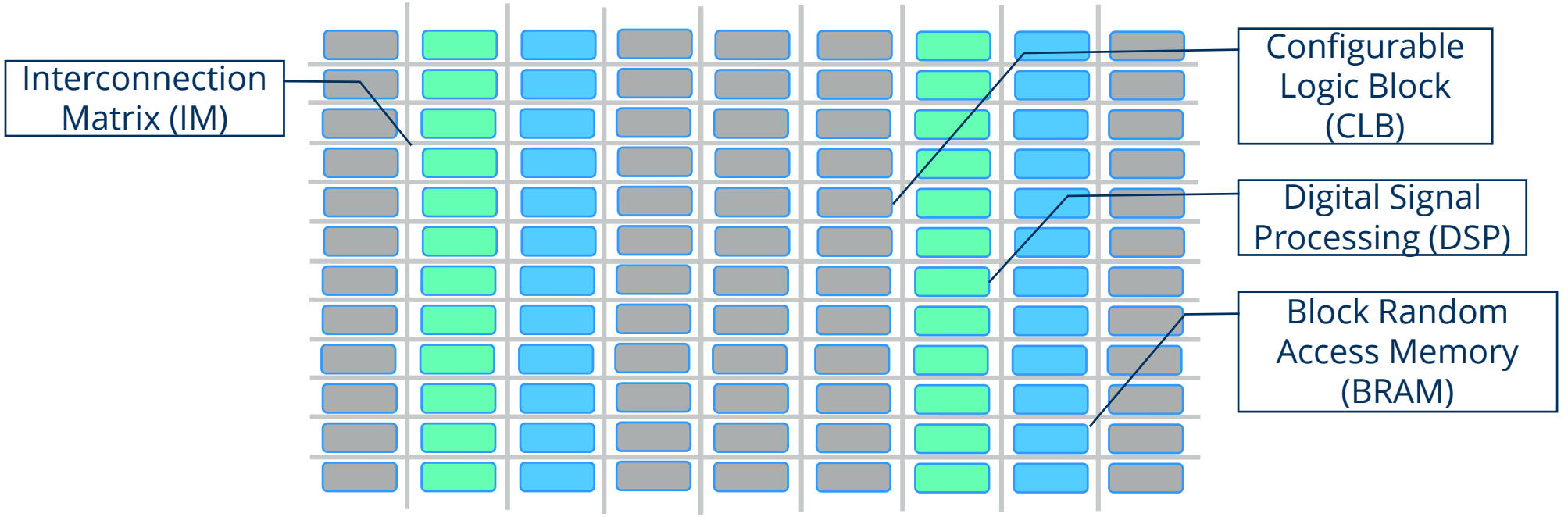
Pros:

- Very flexible → You built your own hardware/processor!
- Can be adapted at design- and runtime → exploiting dynamic and partial reconfiguration
- Low-level parallelization
- Low power consumption compared to multi-cores

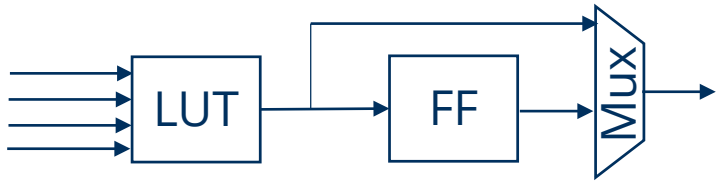
Cons:

- Difficult to program:
  - VHDL, Verilog → Not used by application engineers
  - C-to-FPGA tools (e.g. VivadoHLS)
    → only for accelerators
    → only a subset of ANSI-C, C++ is supported

FPGA Architectures



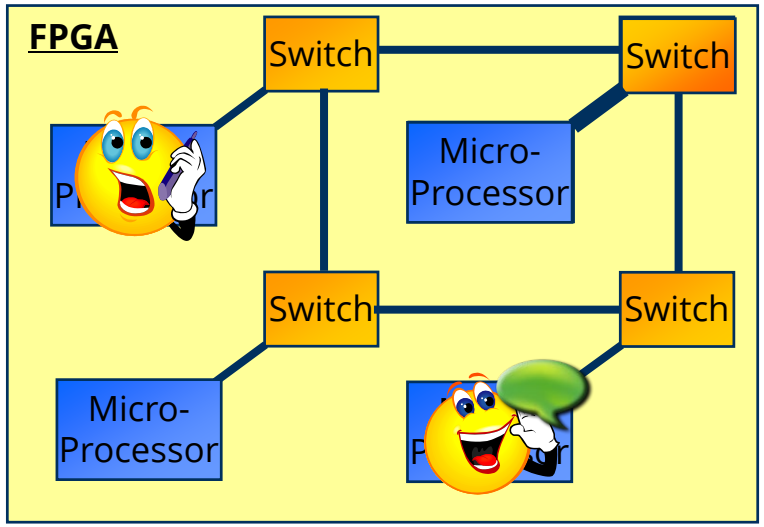
- Major FPGA (Xilinx) blocks are CLBs, DSPs and BRAMs connected by an IM
- CLBs contain Multiplexer (Mux), Look-Up-Tables (LUTs) and Flip-Flops (FFs) providing a configurable logic:
- DSPs are optimized for multiply and add operations
- BRAMs are optimized for memory operations



Dynamic and Partial Reconfiguration

➢Manipulation of a fraction of the configuration data → the remaining hardware architecture stays operative and unaffected

➢For multi-cores → a processing element and its infrastructure can be substituted without disturbing the rest of the multi-core platform



→ "Computing in time and space" : area utilization as well as the time variant content of the hardware device is run-time adaptive

Introduction and Motivation

Problem:

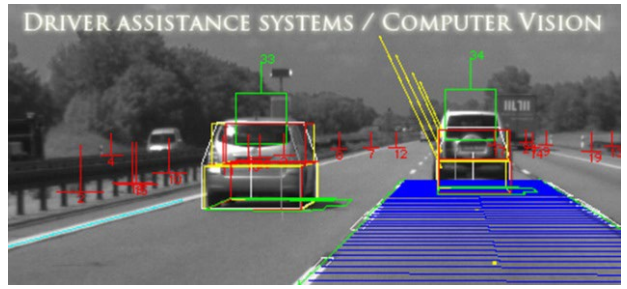
- Increasing complexity
- Real-time requirements
- Low power/energy consumption (batteries, cooling)
- Dynamic adaption to changing environments
- → Cannot be solved anymore by increasing the clock frequency, as $P_{dyn} \sim C \times V^2 \times f$

Possible solution: Domain-specific Computer Architectures

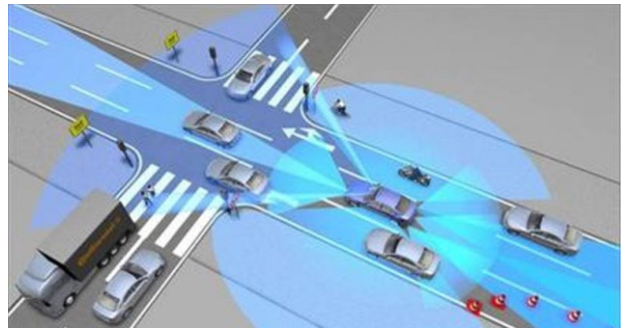
- How to support the dynamic behavior? Is migration of software tasks sufficient?

Better solution: Reconfigurable Domain-specific Computer Architectures

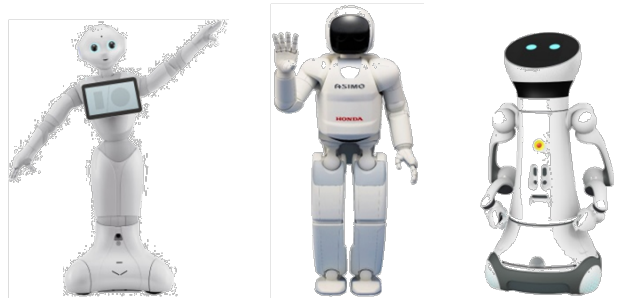
- Runtime adaptation of hardware and software
- Energy-efficient solution for each application phase



http://www.rcs.ei.tum.de/forschung/driver-assistance/

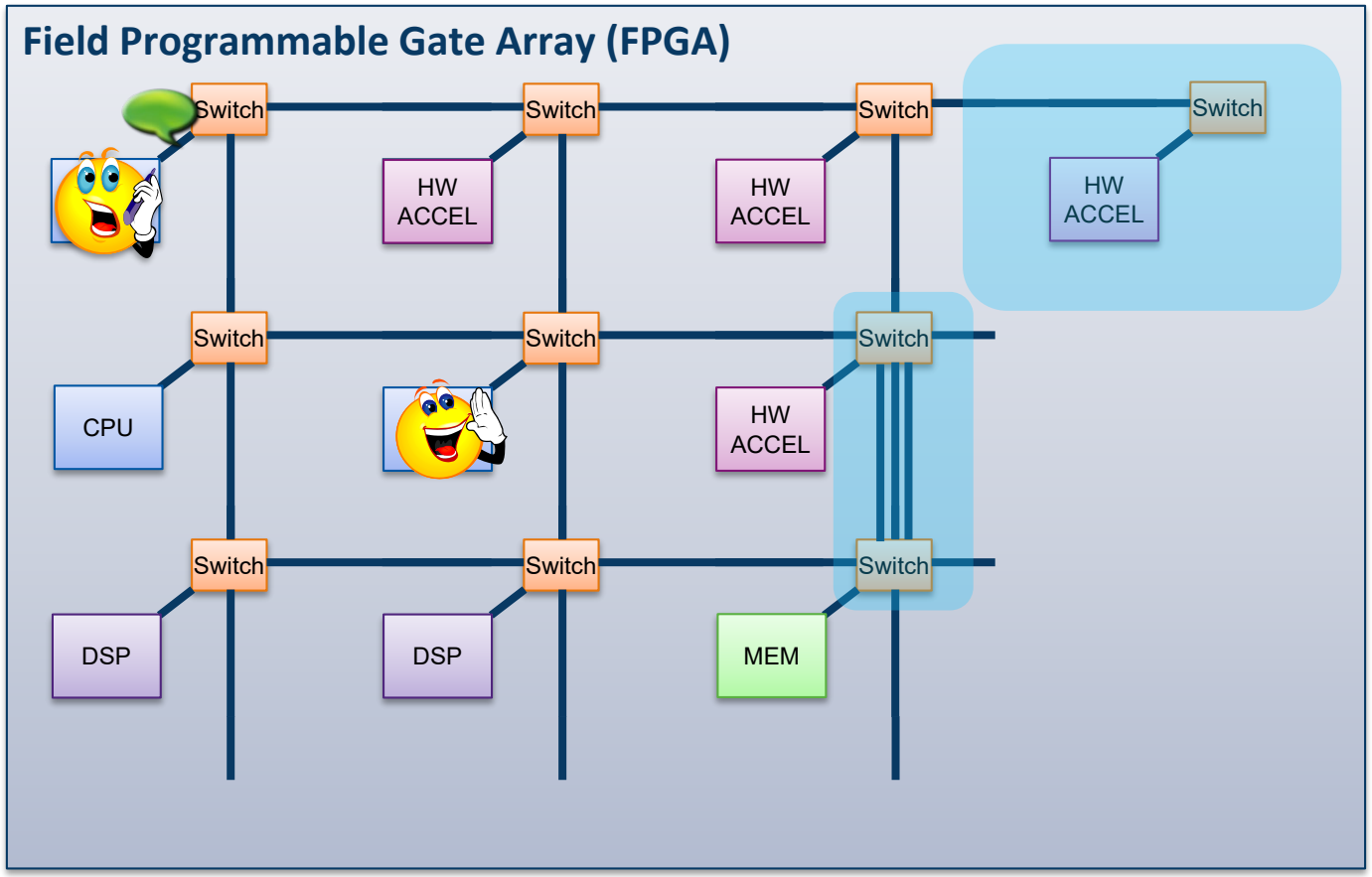


http:// www.asdreports.com/news-10595/key-players-advanced-driver-assistance-systems-adas-market-north-america-20152019



www.ald.softbankrobotics.com asimo.honda.com www.care-o-bot-4.de

Reconfigurable Domain-Specific Computer Architectures



Advantages:

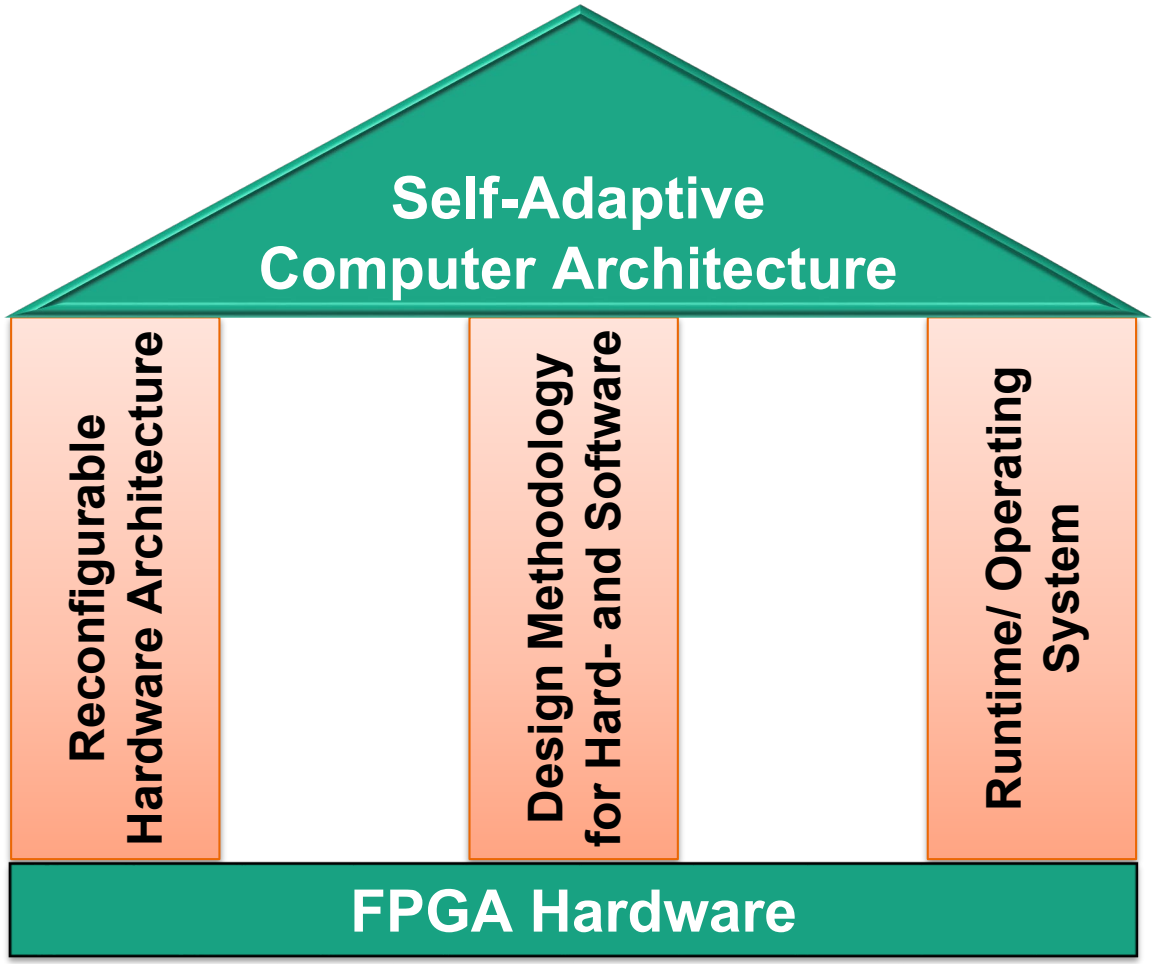
- **Highly flexible → Runtime adaption of hardware and software** to application requirements
- **High performance / watt → Application-specific architecture at design- and runtime**
- **More functionality** on a **small chip** by hardware time multiplexing → **Reduced costs & power/energy**
- Improving **fault tolerance**

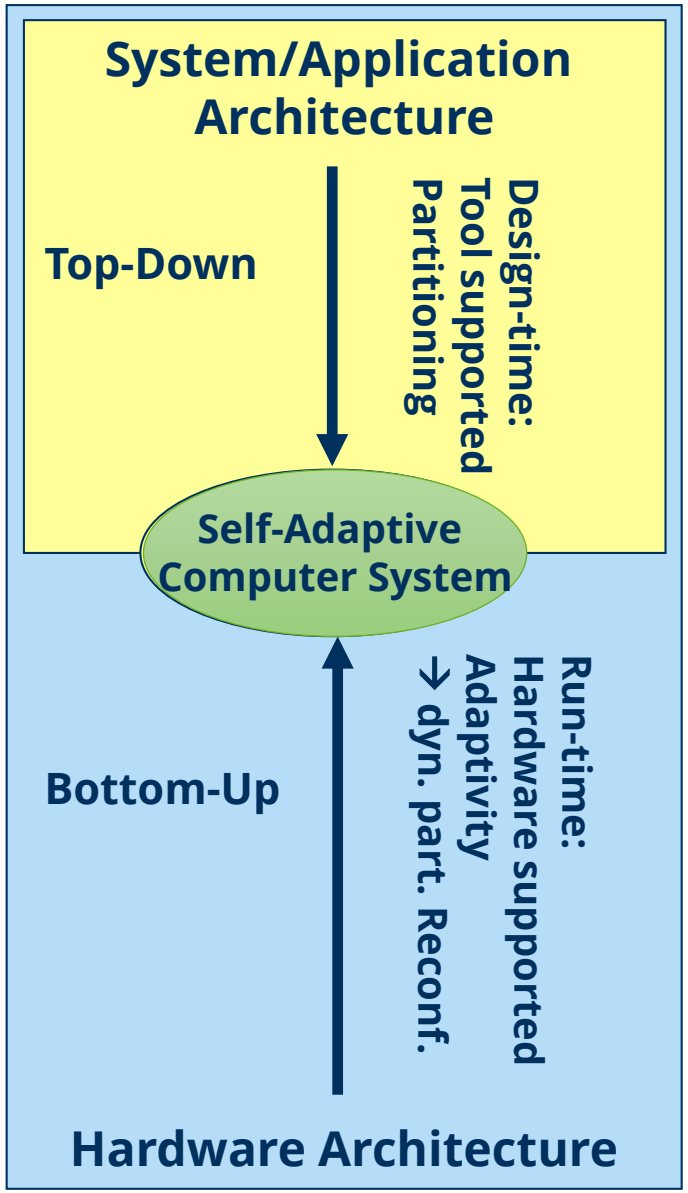
Towards Self-Adaptive Domain Specific Computer Architectures

- Challenges are e.g.:
  - Number and type of processing elements (PEs)
  - Type of communication infrastructure (Bus, Network-on-Chip (NoC), etc.)
  - Application description
  - Application partitioning and mapping
  - Which components need to be runtime adaptive?
  - How to decide at runtime, if and how the systems needs to be adapted? Who makes this decision?

→ Huge design space: Hardware and application mapping have to be managed over time

„Meet-in-the-Middle“ Solution

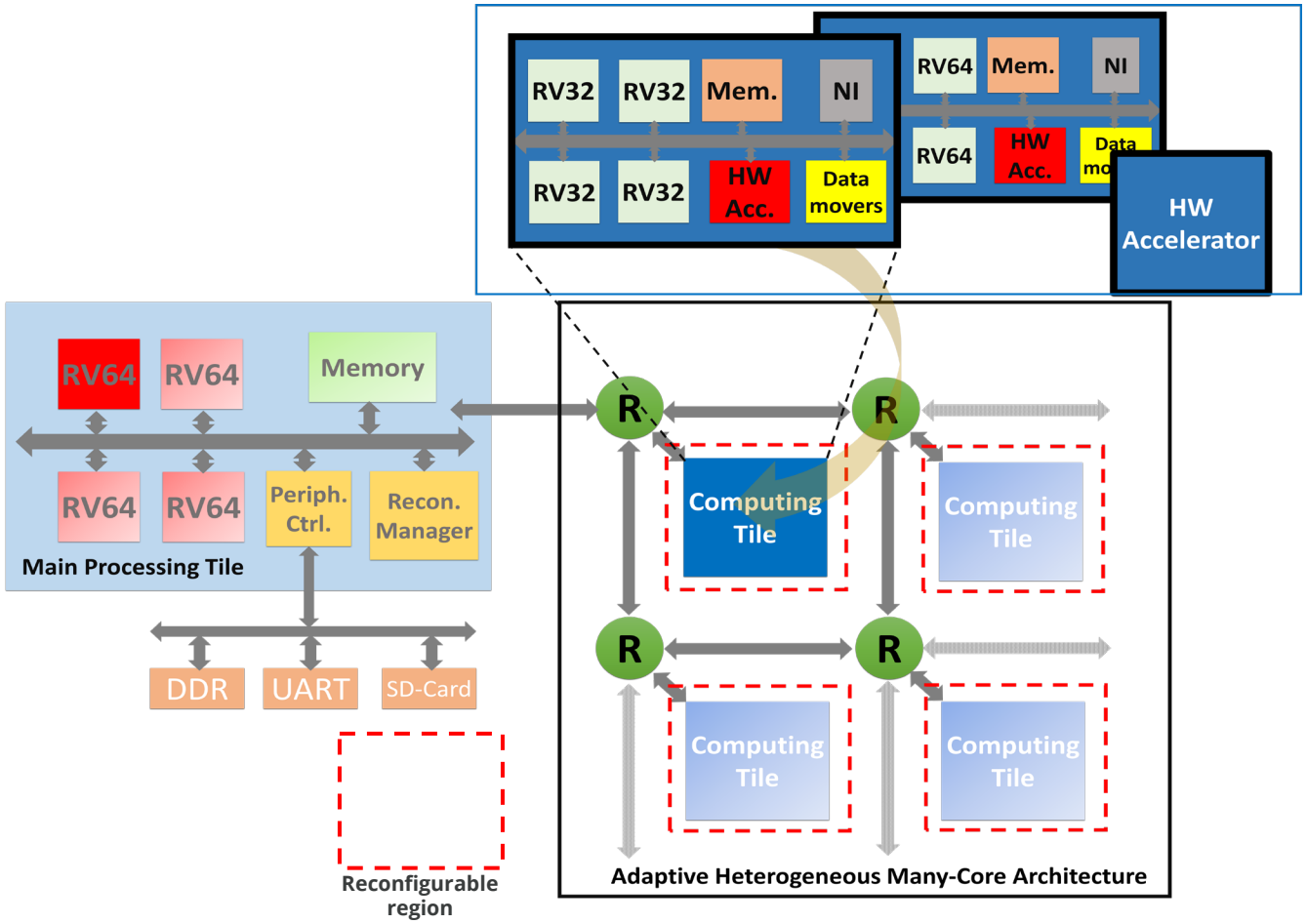




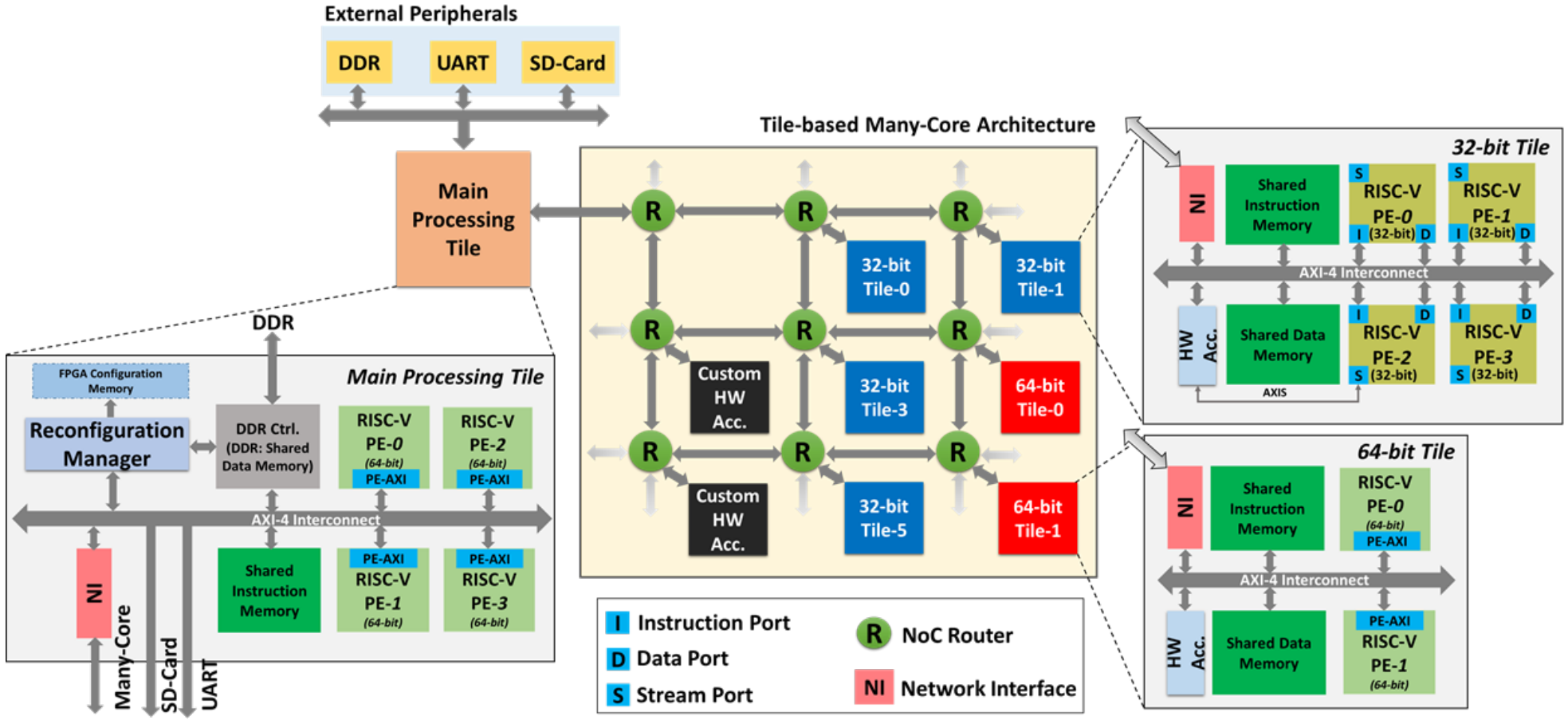
Outline

- Introduction and Motivation
- Reconfigurable Domain-Specific Computer Architectures
  - System Overview and Processing Elements
  - Reconfiguration Management
  - Network-on-Chip
- Design / Programming Methodology
  - High Level Synthesis Library: HiFlipVX
  - Application Partitioning and Mapping
- Application Examples in Relation to Funded Research Projects
- Conclusion and Outlook

Reconfigurable Domain-Specific Computer Architecture

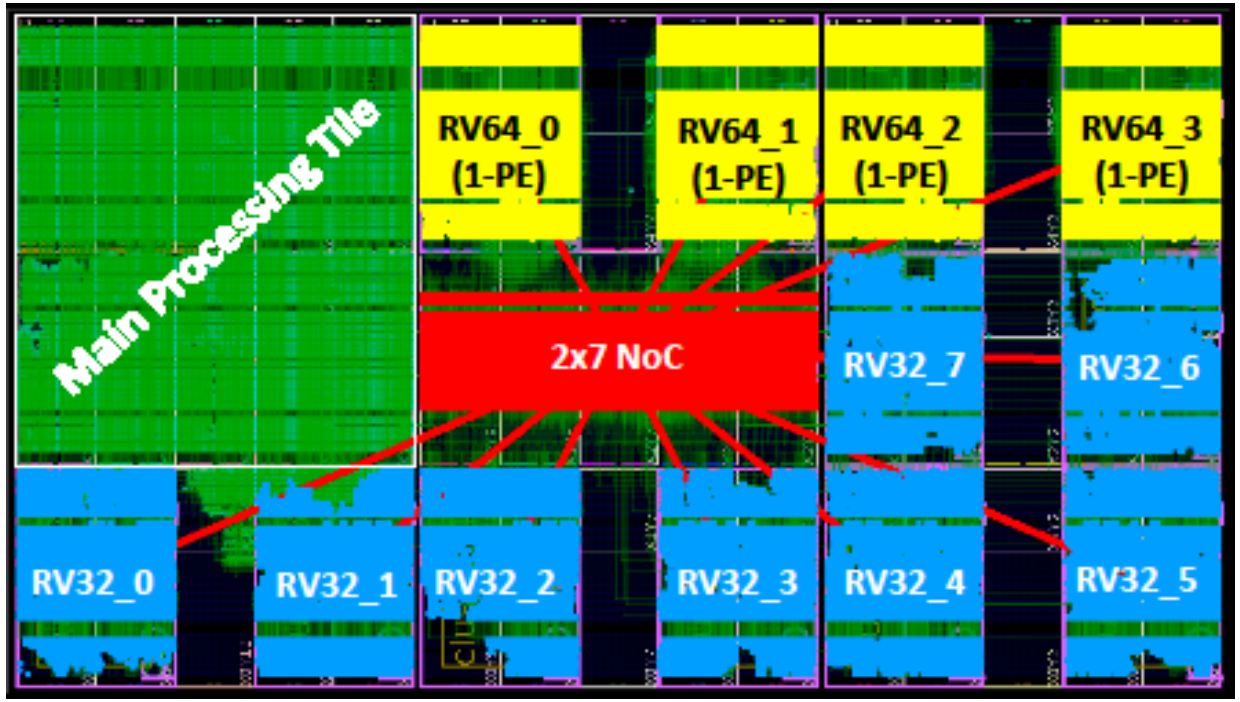


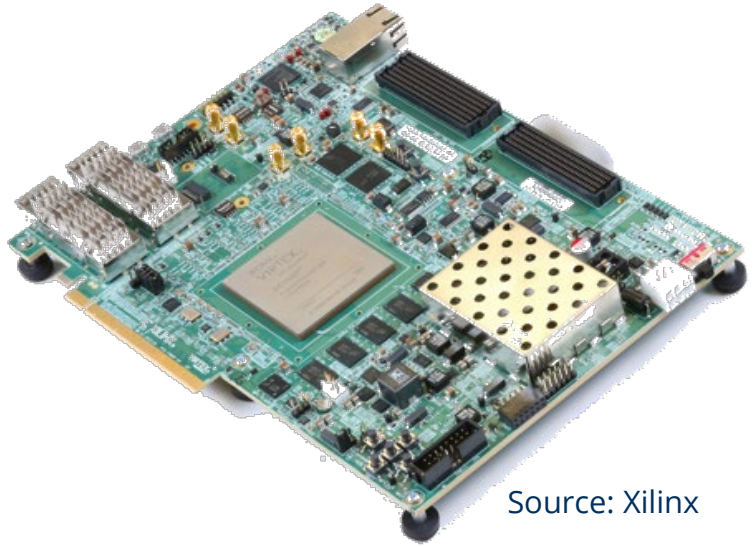
Reconfigurable Domain-Specific Computer Architecture



Kamaleldin A, Göhringer D (2022) AGILER: An Adaptive Heterogeneous Tile-based Many-Core Architecture for RISC-V Processors. IEEE Access.

Reconfigurable Domain-Specific Computer Architecture

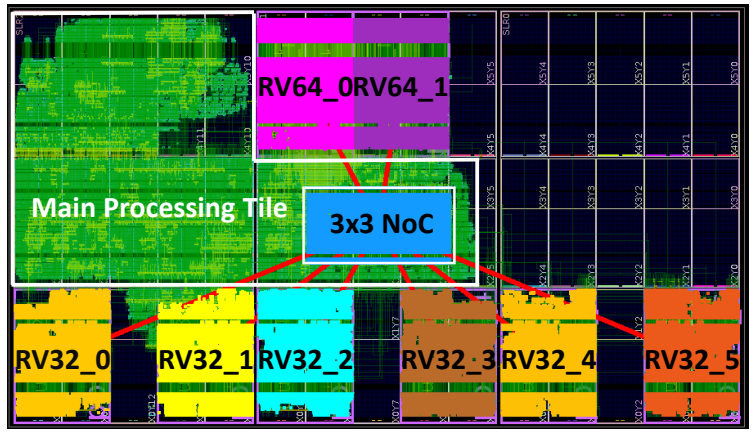


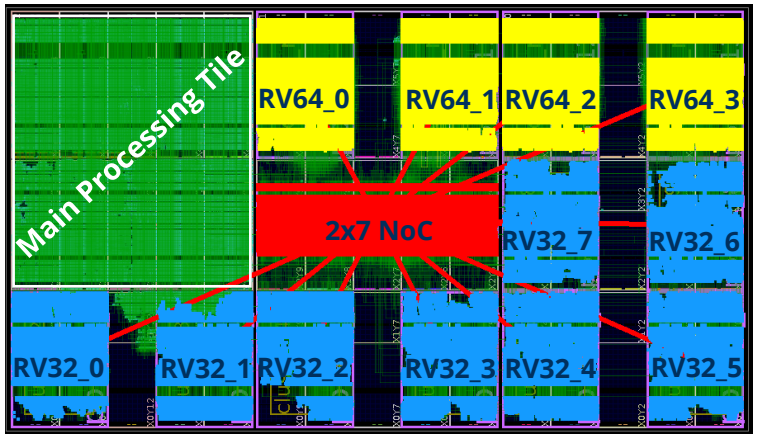


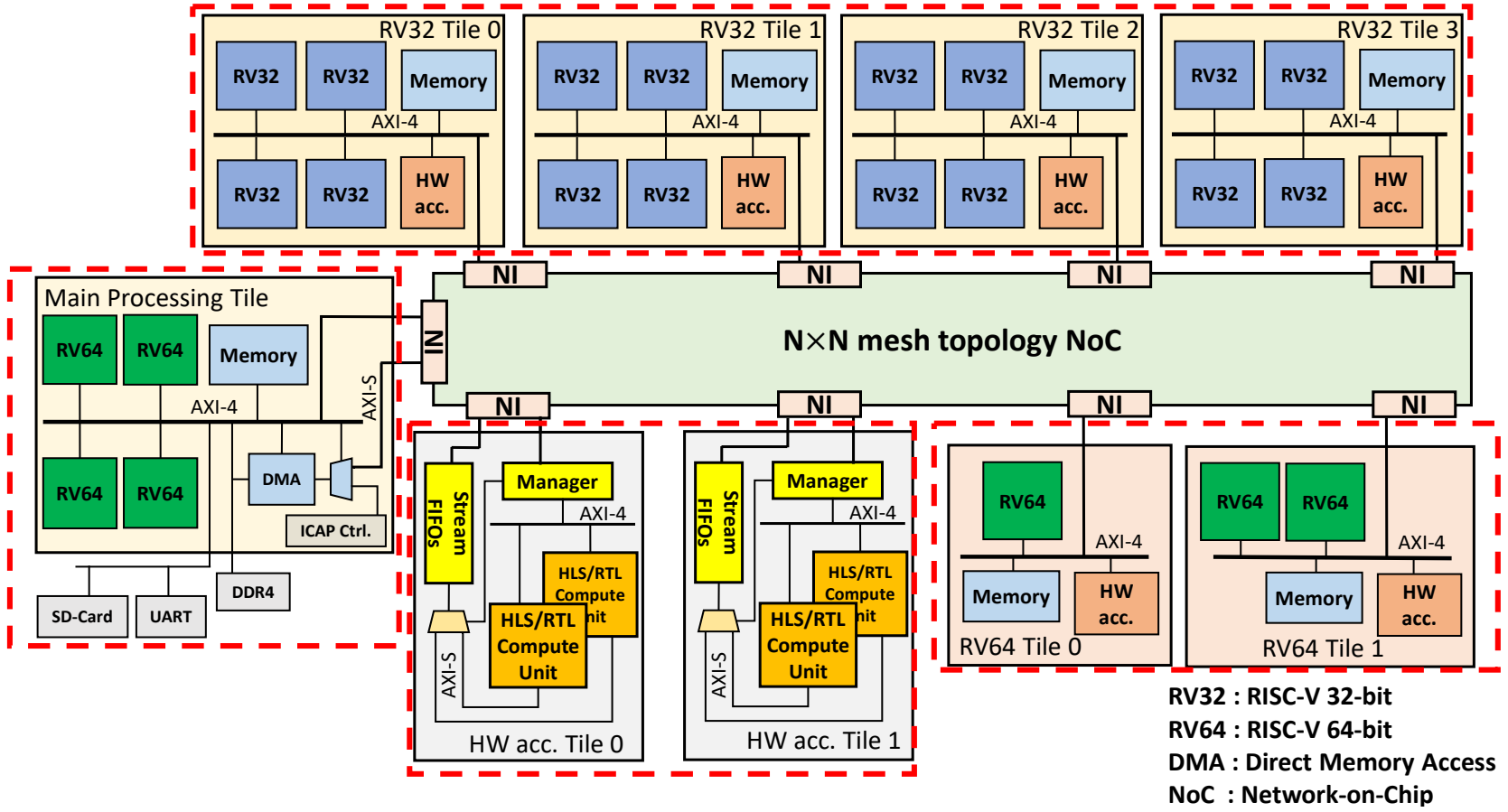
Source: Xilinx

Kamaleldin A, Göhringer D (2022) AGILER: An Adaptive Heterogeneous Tile-based Many-Core Architecture for RISC-V Processorss. IEEE Access.

Multiple Heterogeneous Many-Core Configurations

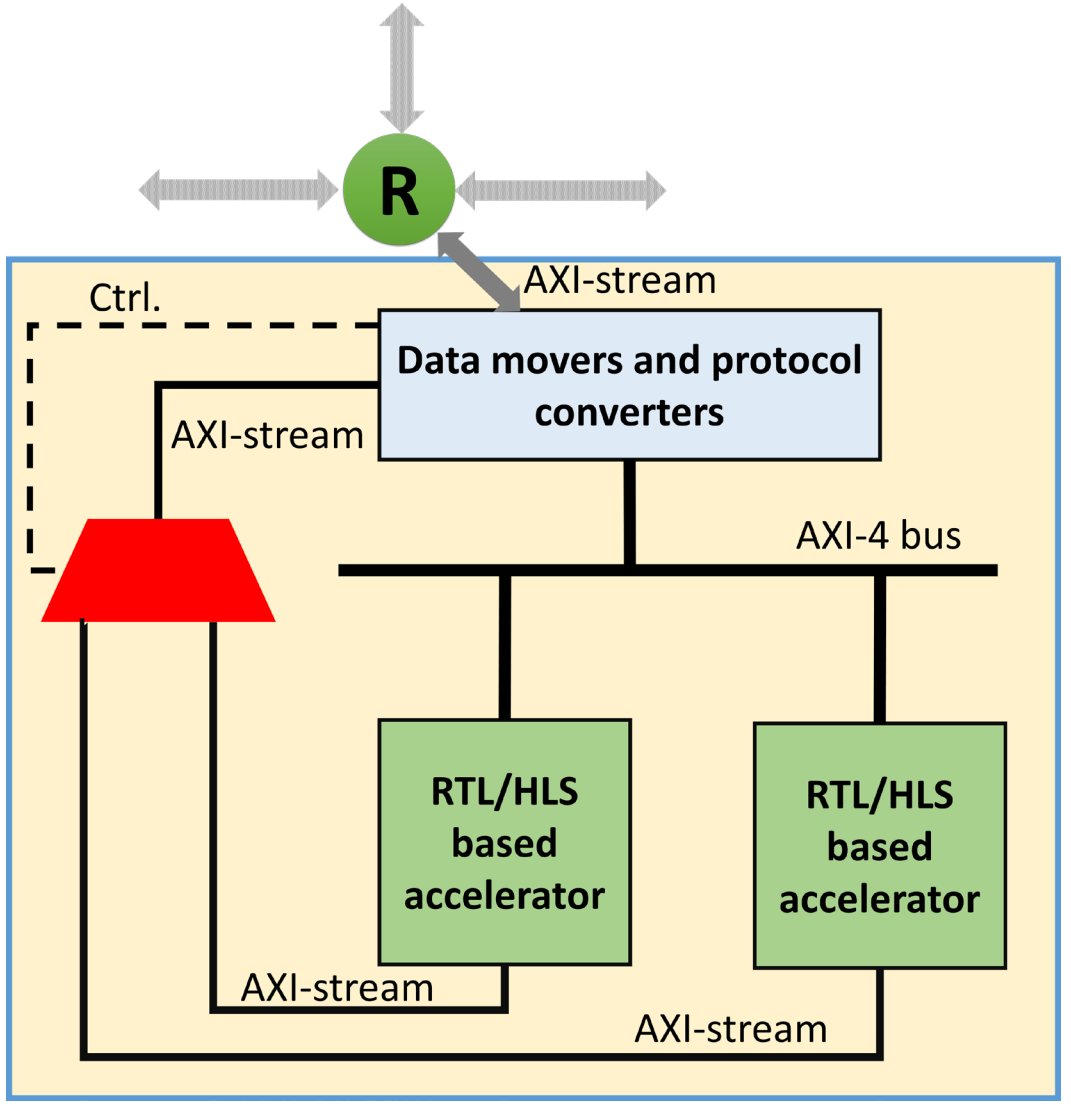






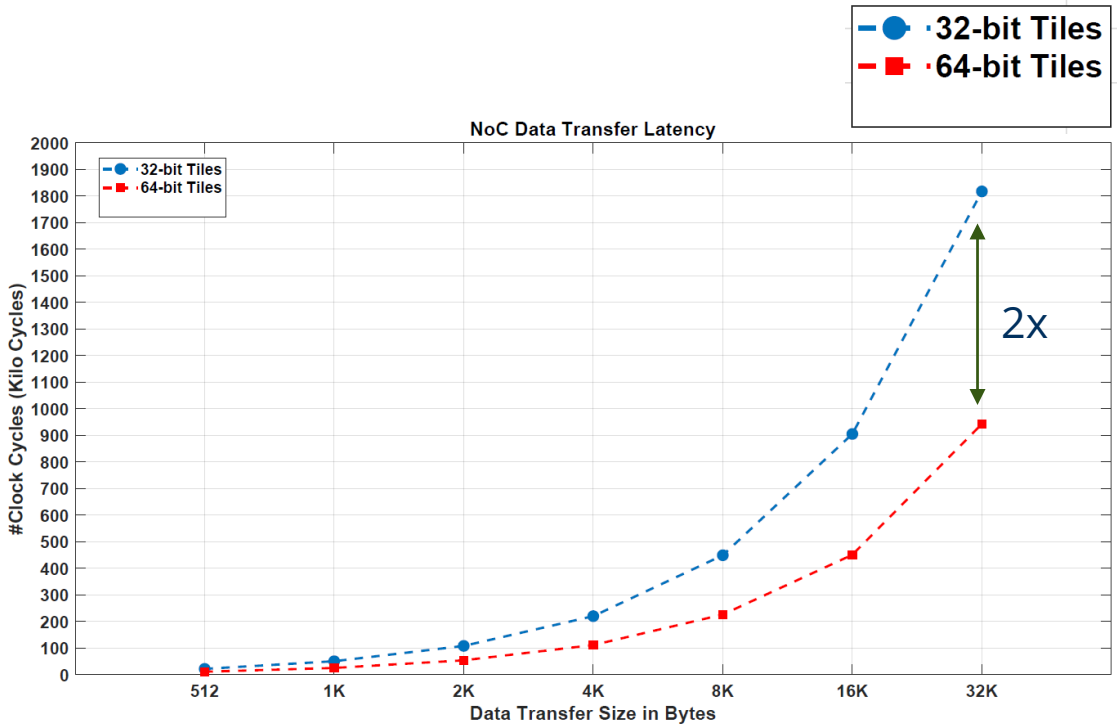
HW Accelerator Tile

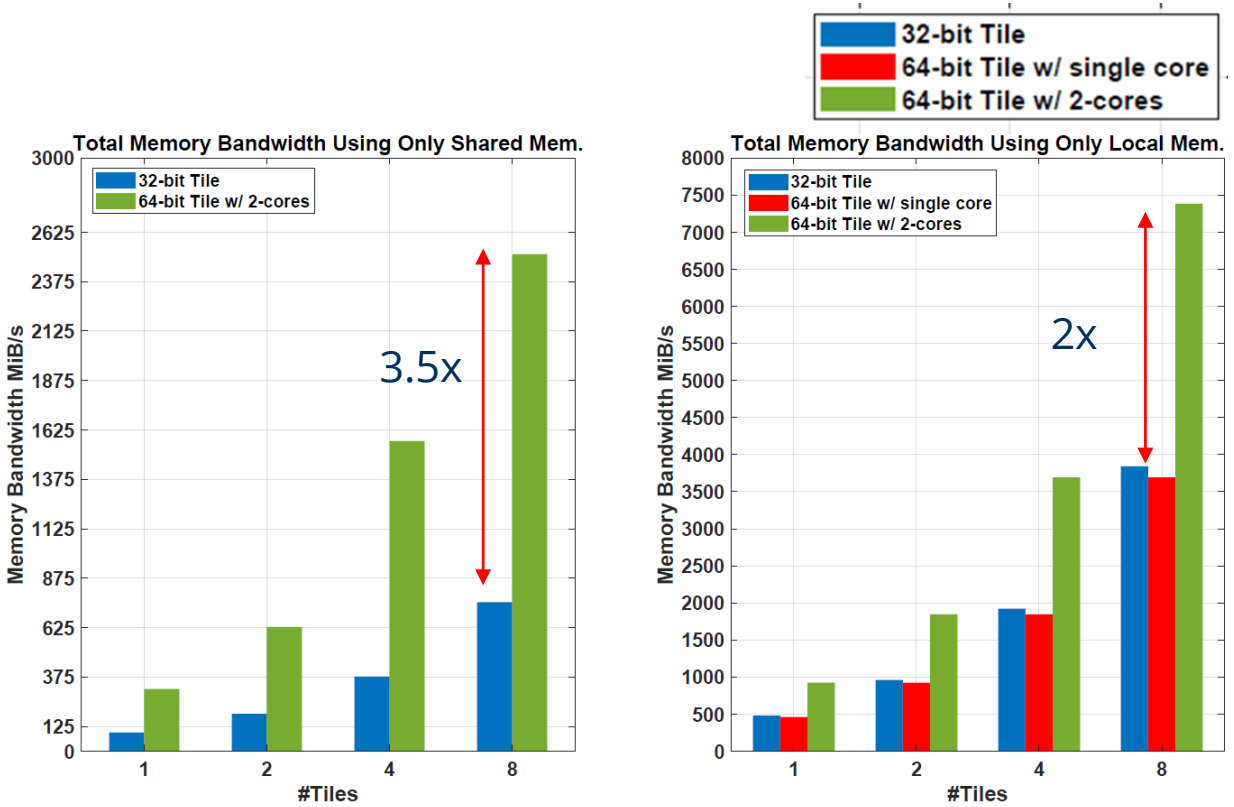
- Specialized loosely-coupled accelerators implemented using RTL /HLS with AXI-4 or AXI-S interfaces
- The tile provides access to other computing tiles through NoC
- Data movers (e.g. DMA, FIFOs) are responsible to handle data movement between the NoC and internal accelerators
- Software drivers for data exchanging and communication are essential for HW/SW co-design



HW Accelerator Tile

Performance Evaluation



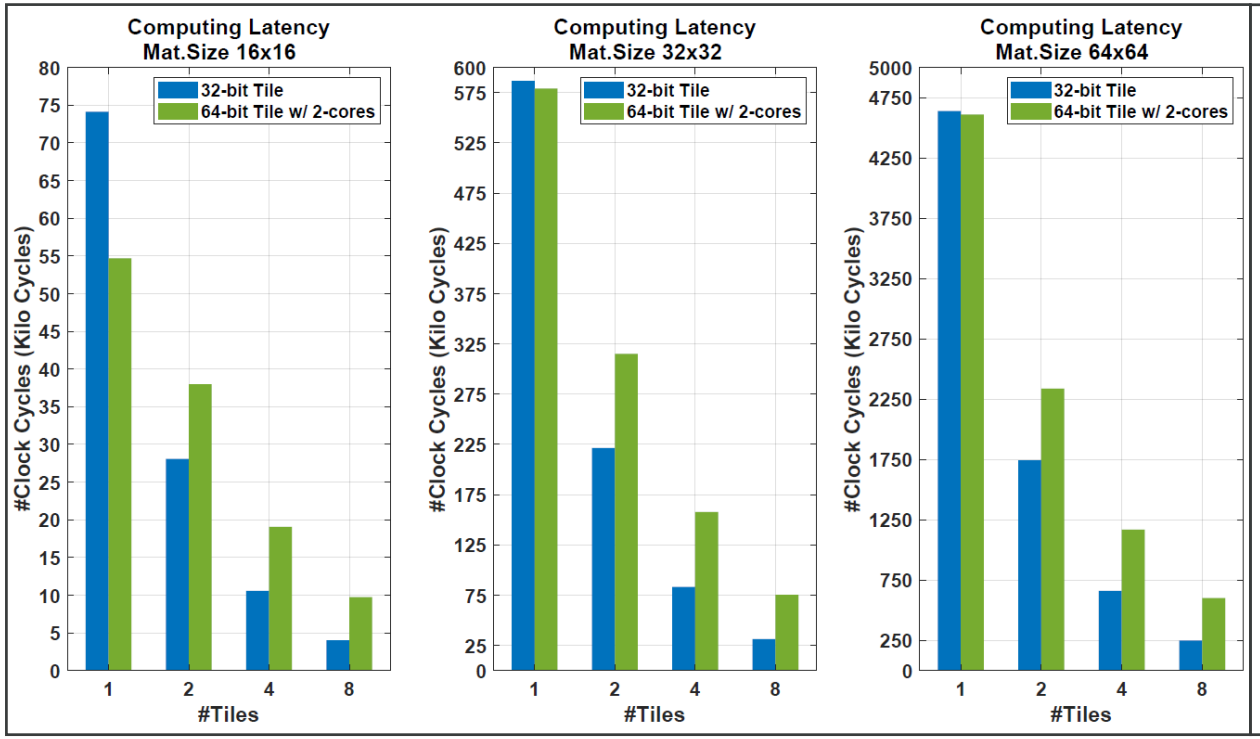


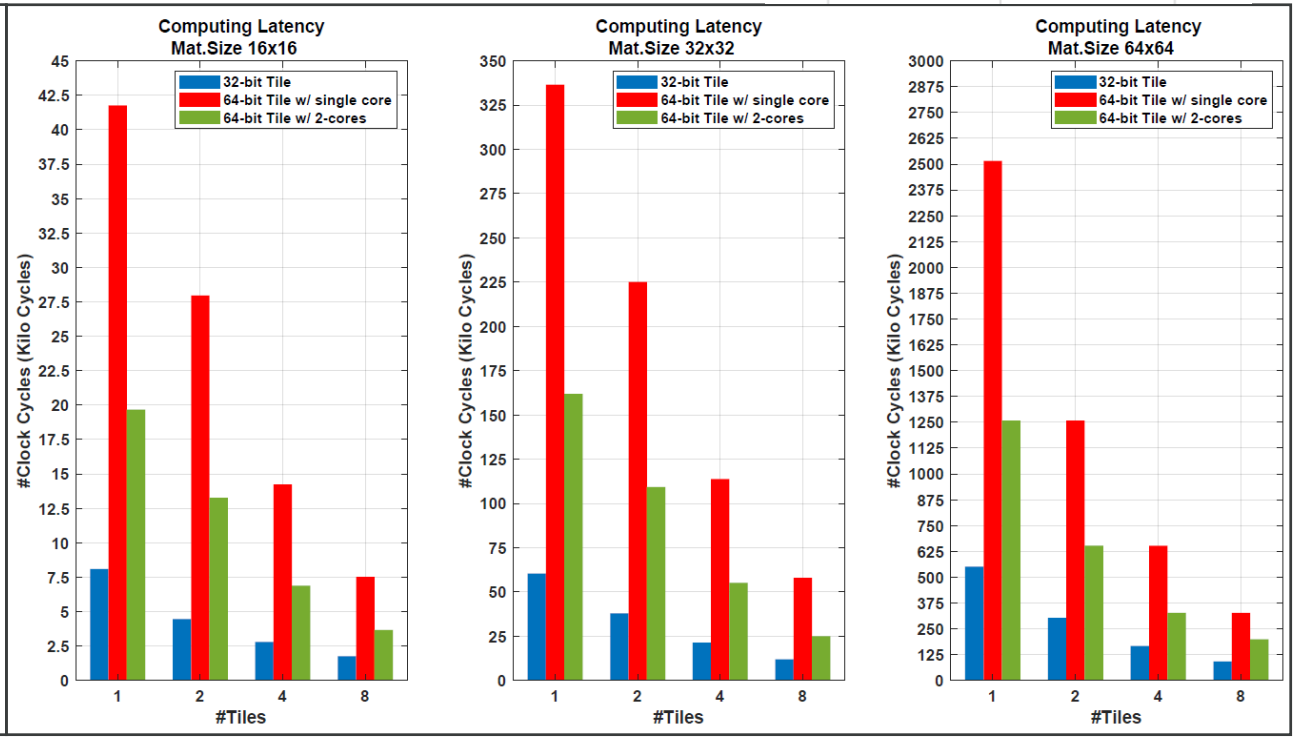
Data transfer latency between heterogeneous tiles (lower is better)

Achievable memory bandwidth for heterogeneous tiles (higher is better)

Kamaleldin A, Göhringer D (2022) AGILER: An Adaptive Heterogeneous Tile-based Many-Core Architecture for RISC-V Processorss. IEEE Access.

Performance Evaluation





Matrix multiplication using shared memory

Matrix multiplication using local memory

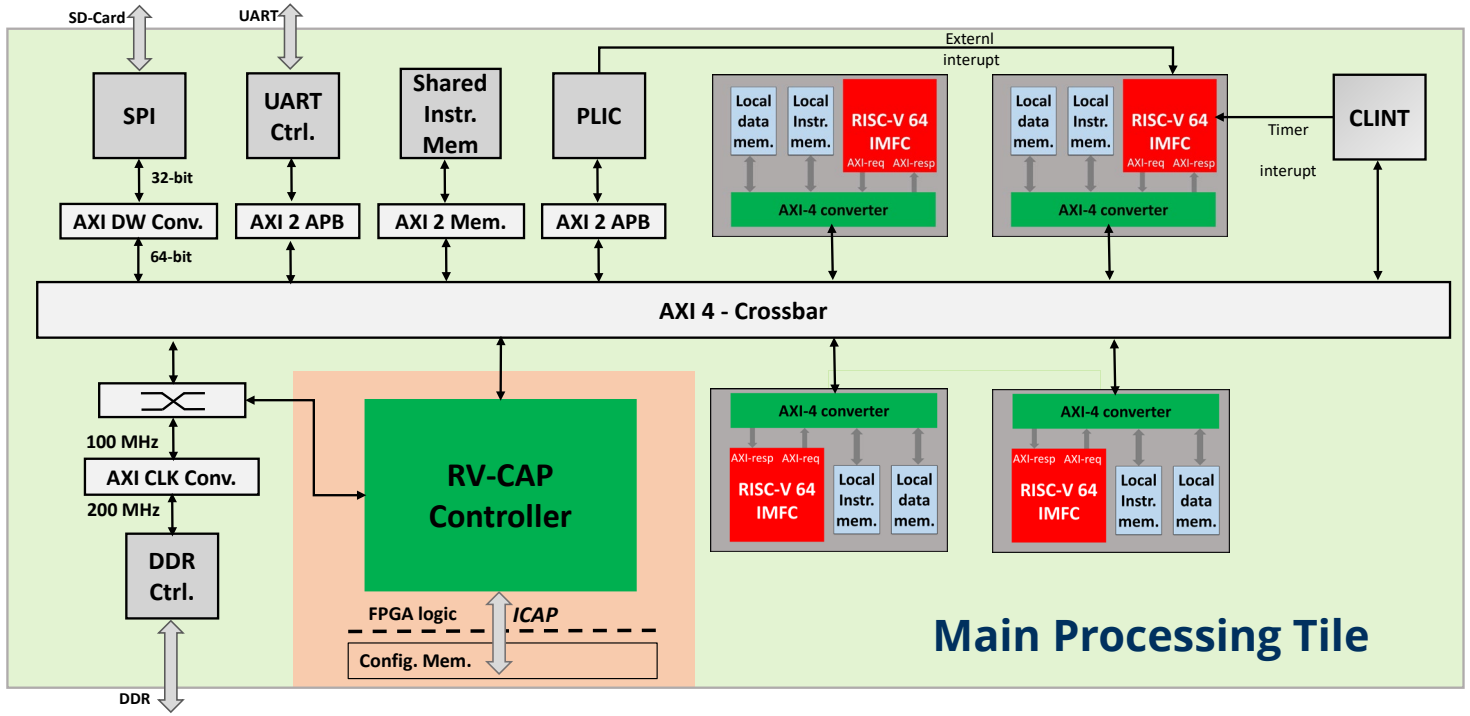
Kamaleldin A, Göhringer D (2022) AGILER: An Adaptive Heterogeneous Tile-based Many-Core Architecture for RISC-V Processorss. IEEE Access.

Outline

- Introduction and Motivation
- Reconfigurable Domain-Specific Computer Architectures
  - System Overview and Processing Elements
  - Reconfiguration Management
  - Network-on-Chip
- Design / Programming Methodology
  - High Level Synthesis Library: HiFlipVX
  - Application Partitioning and Mapping
- Application Examples in Relation to Funded Research Projects
- Conclusion and Outlook

Internal Reconfiguration Management

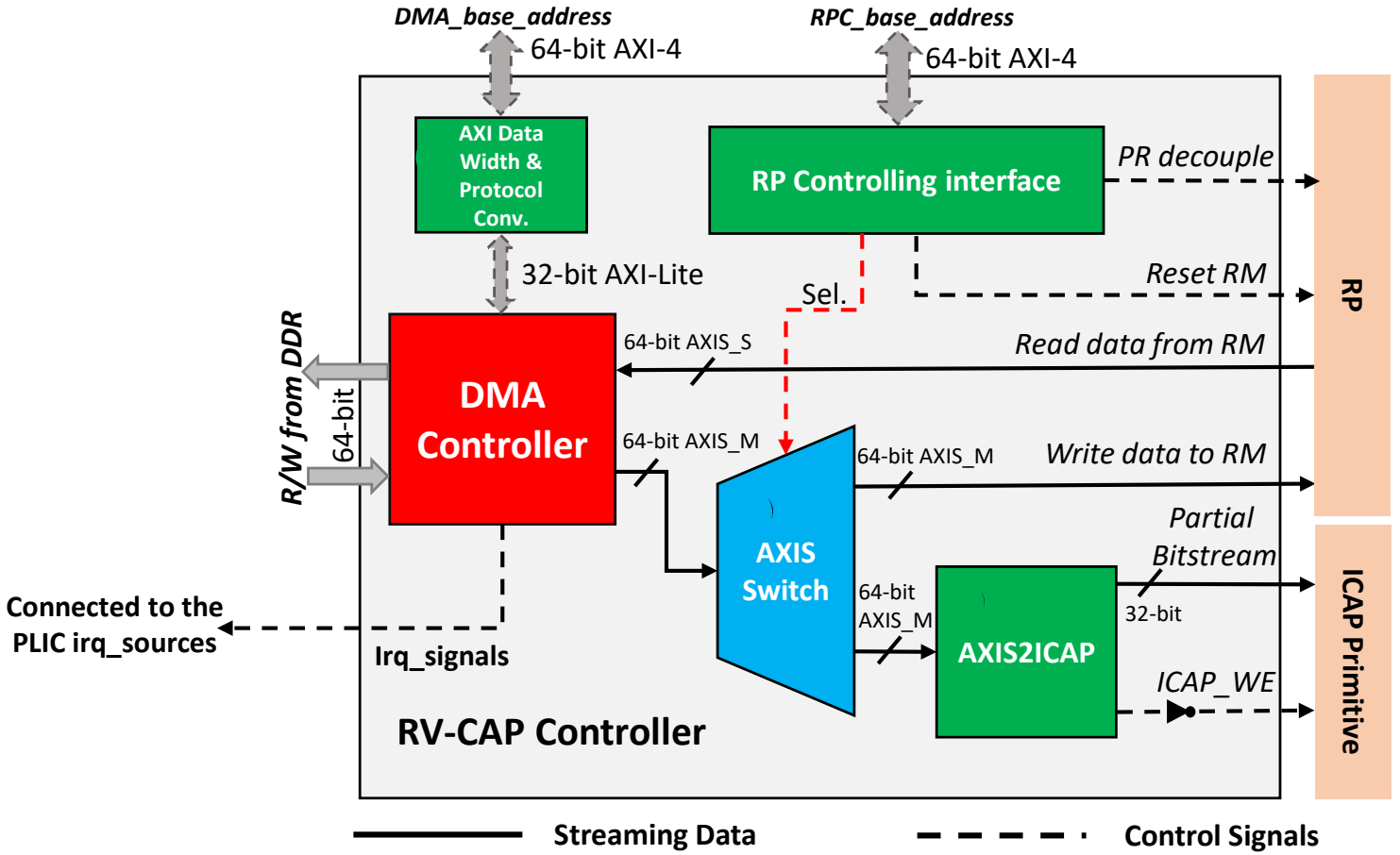
- The internal reconfiguration management (RV-CAP) is implemented inside the main processing tile
- Dynamic and partial reconfiguration controlling for RISC-V PE
- High-speed reconfiguration for different RP sizes is achieved



Charaf N, Kamaleldin A, Thümmler M, Göhringer D (2021) RV-CAP: Enabling Dynamic Partial Reconfiguration for FPGA-Based RISC-V System-on-Chip. In Proc. of the IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), pp. 1-8.

DPR Controller

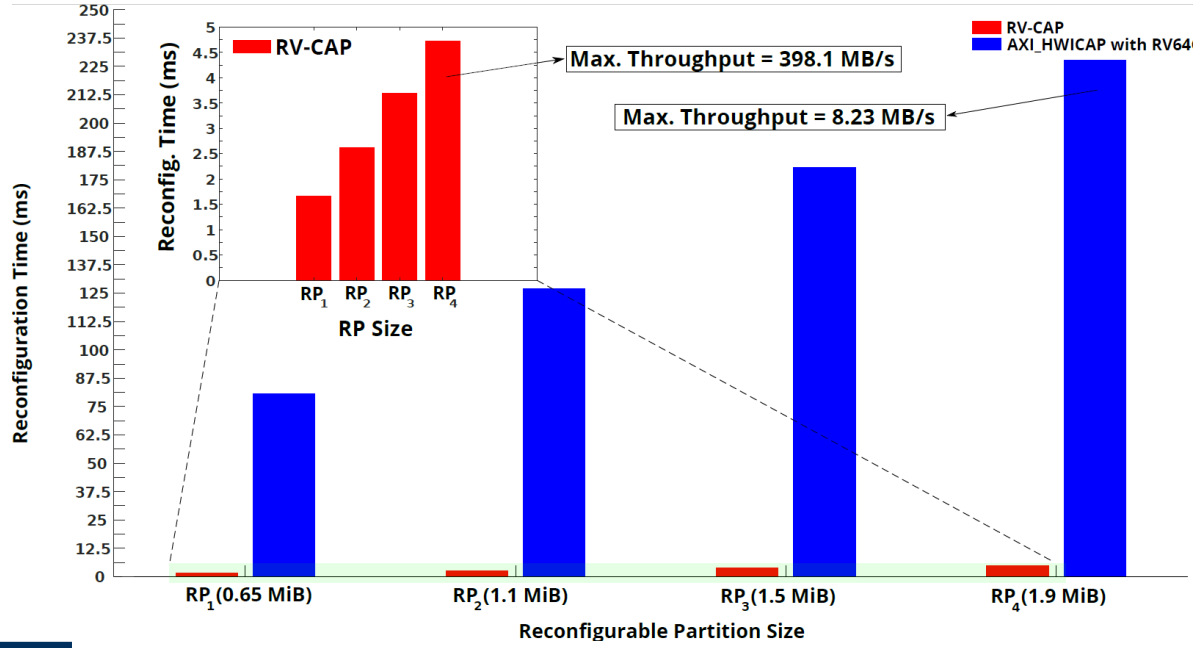
- RV-CAP is the first DPR controller for RISC-V based SoC
- Fully software controlled through one of RISC-V cores inside the main processing tile
- Partial bitstreams are transferred from external DDR to ICAP primitive using DMA
- Optionally, data can be streamed to a reconfigurable partition directly connected to the main processing tile



Charaf N, Kamaleldin A, Thümmler M, Göhringer D (2021) RV-CAP: Enabling Dynamic Partial Reconfiguration for FPGA-Based RISC-V System-on-Chip. In Proc. of the IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), pp. 1-8.

Performance Evaluation

DPR Controller (RV-CAP)	Resources Utilization		
	LUT	FF	BRAM
PR Cntrl. + AXI modules	420	909	0
DMA	1897	3044	6



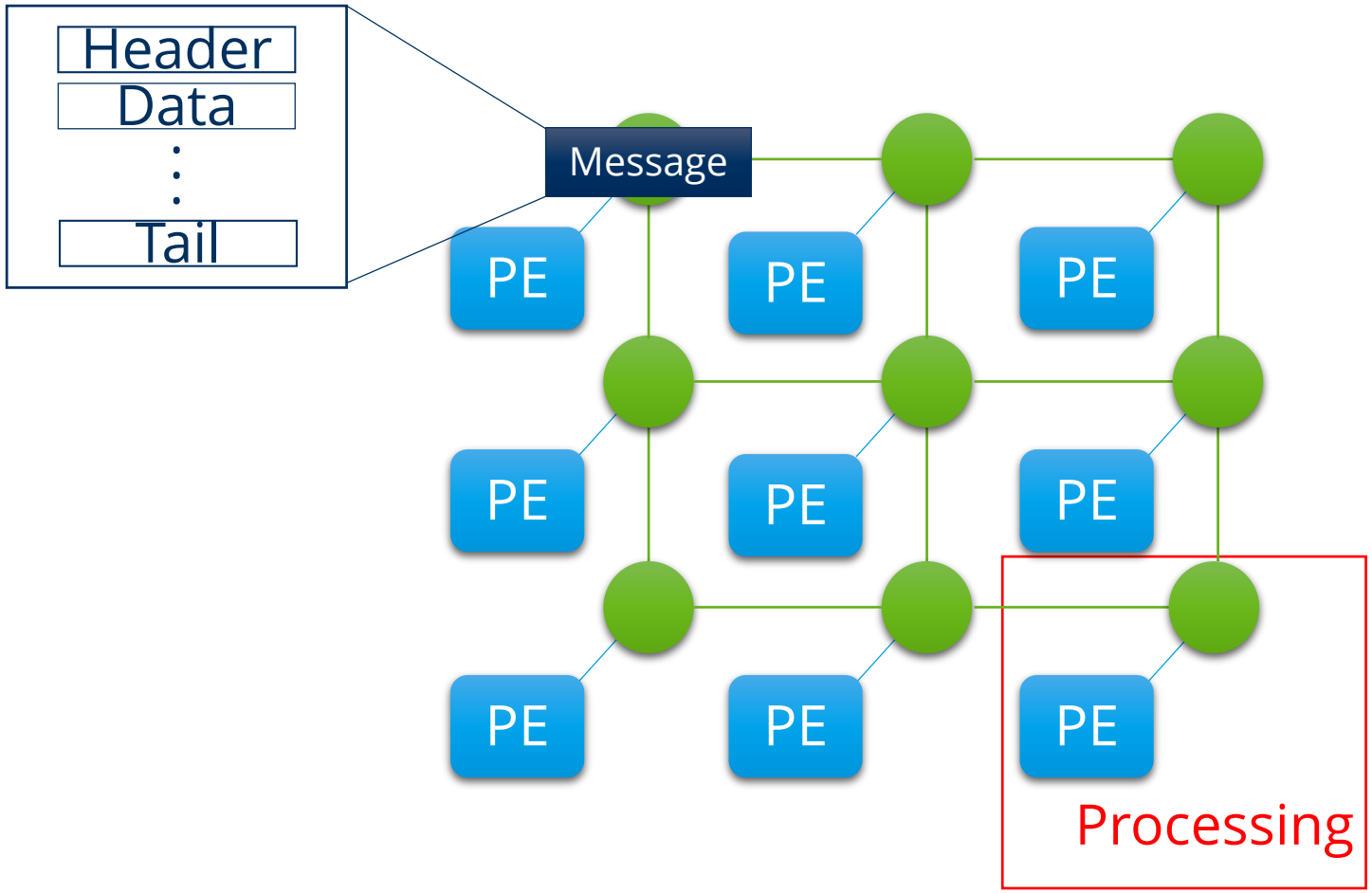
DPR Controller	SoC Processor	Freq. (MHz)	Throughput (MB/s)
Zycap [1]	ARM	100	382
Di Carlo et al. [2]	LEON3	100	395.4
AXI-HWICAP	RV64IMFC	100	8.23
RV-CAP	RV64IMFC	100	398.1

[1] Kizheppatt V, Fahmy S. A. (2014) ZyCAP: Efficient partial reconfiguration management on the Xilinx Zynq, IEEE Embedded Systems Letters.
[2] Di Carlo S, et al., A portable open-source controller for safe dynamic partial reconfiguration on Xilinx FPGAs, FPL 2015.

Outline

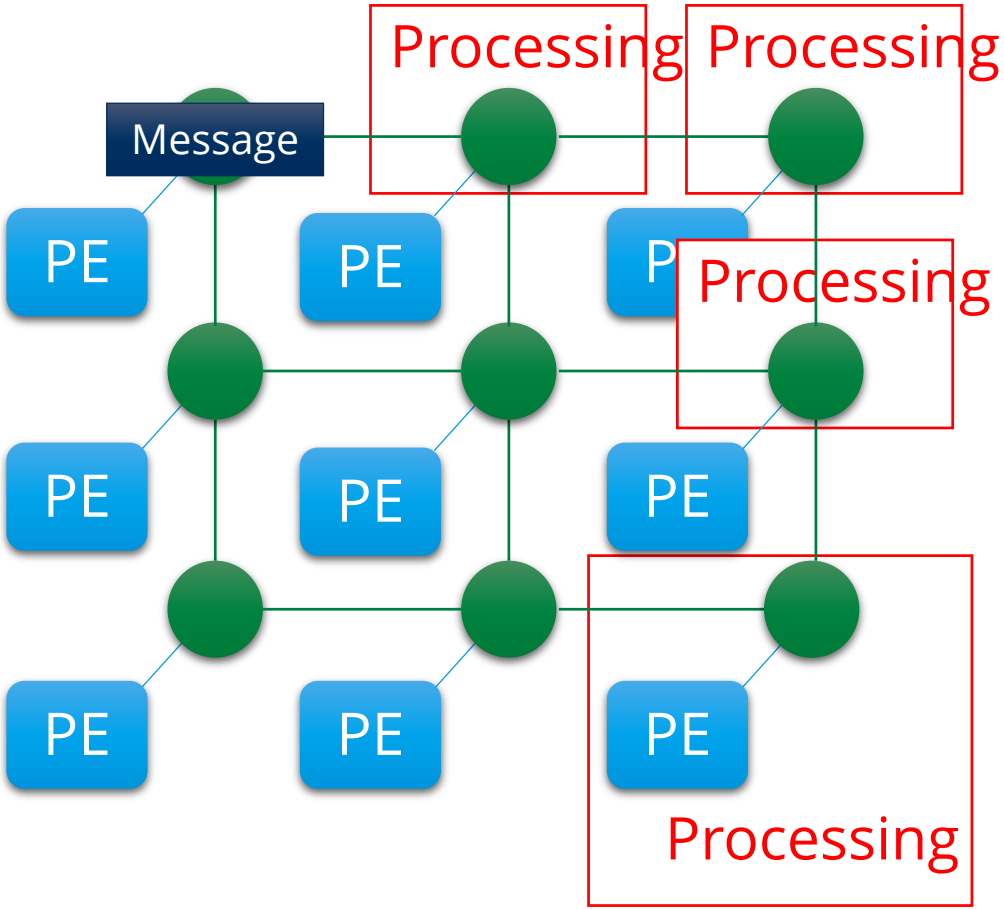
- Introduction and Motivation
- Reconfigurable Domain-Specific Computer Architectures
  - System Overview and Processing Elements
  - Reconfiguration Management
  - Network-on-Chip
- Design / Programming Methodology
  - High Level Synthesis Library: HiFlipVX
  - Application Partitioning and Mapping
- Application Examples in Relation to Funded Research Projects
- Conclusion and Outlook

Processing in Networks-on-Chip



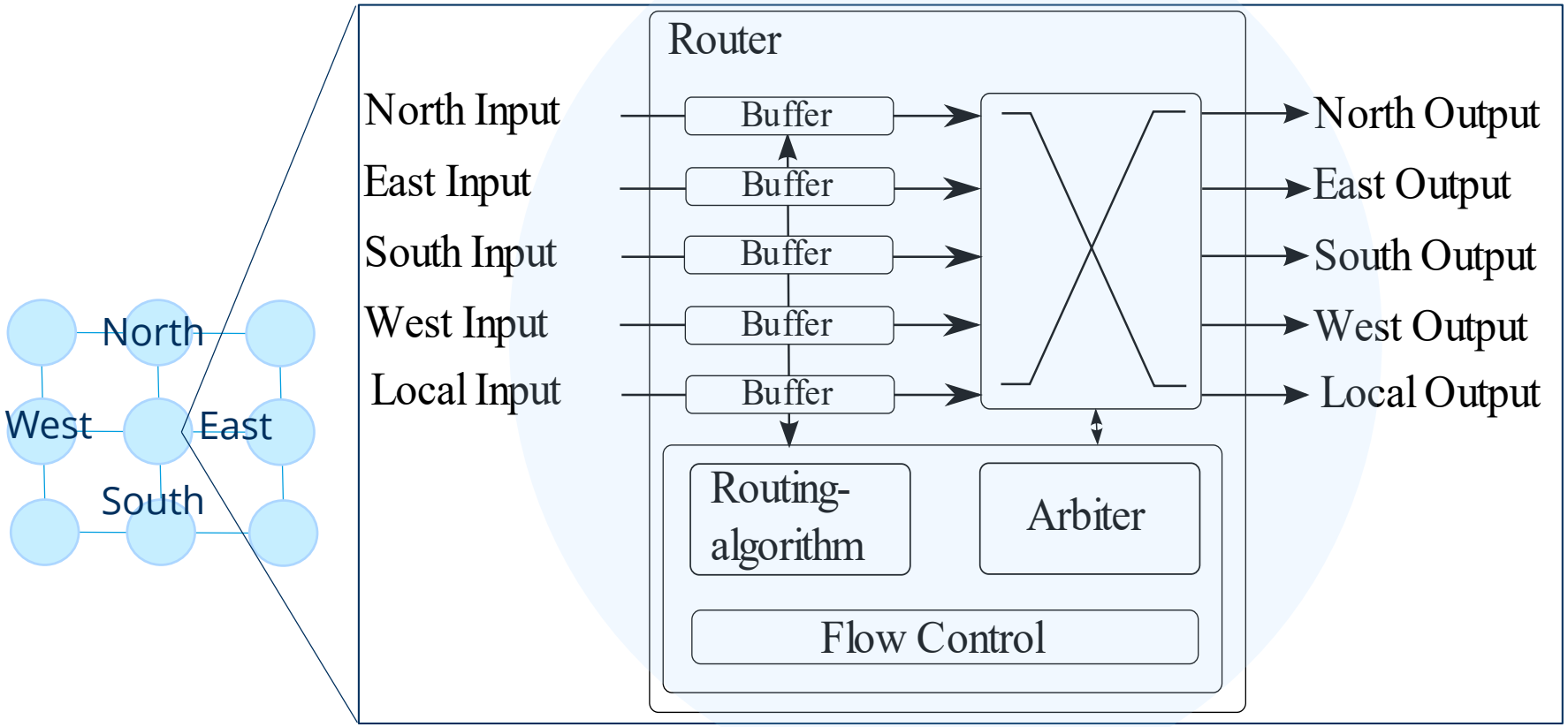
Rettkowski J, Göhringer D (2018) *ASIR: Application-Specific Instruction-Set Router for NoC-based MPSoCs*. Computers, MDPI.

Processing in Networks-on-Chip



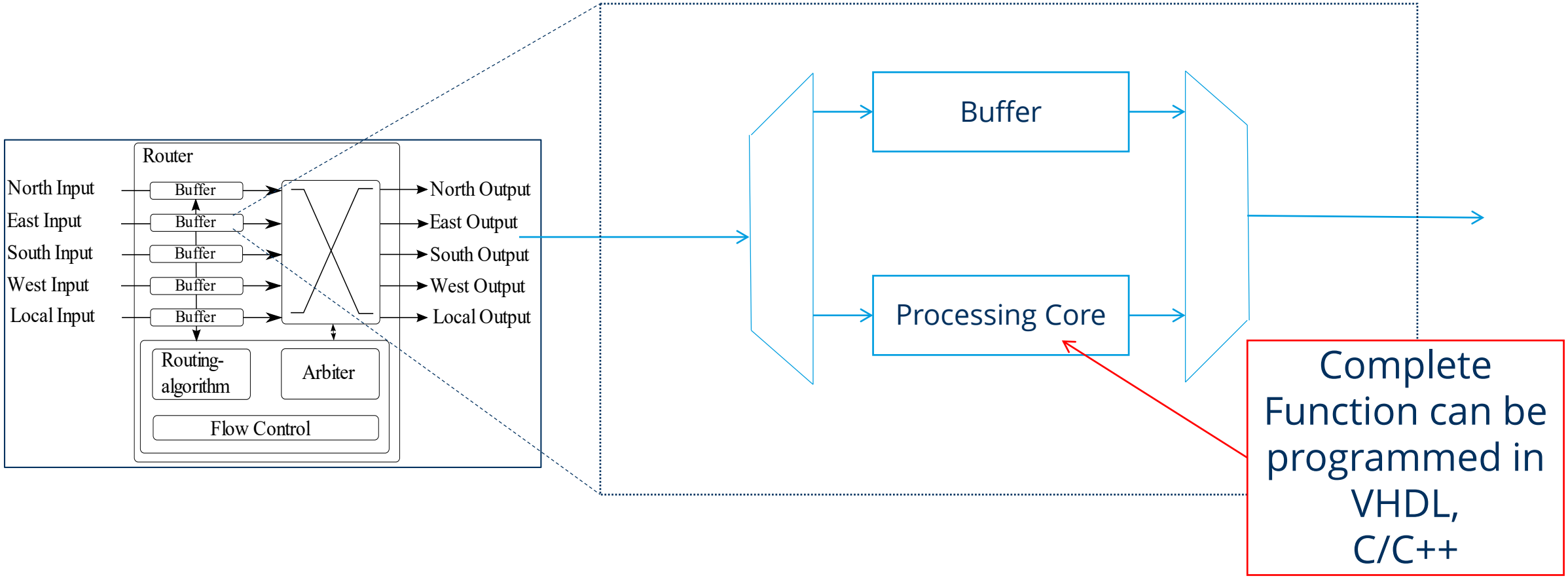
Rettkowski J, Göhringer D (2018) *ASIR: Application-Specific Instruction-Set Router for NoC-based MPSoCs*. Computers, MDPI.

Processing in Networks-on-Chip



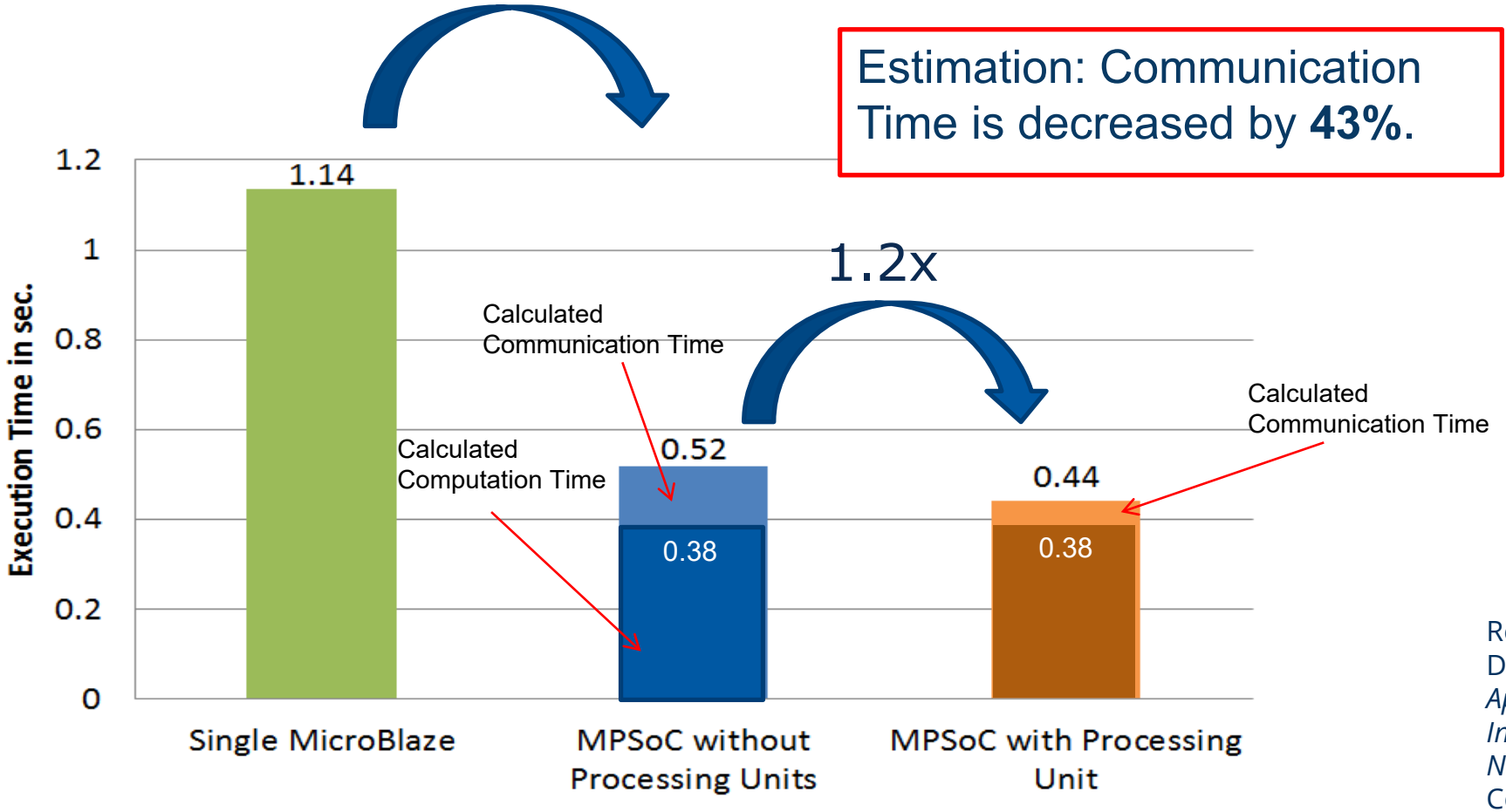
Rettkowski J, Göhringer D (2018) *ASIR: Application-Specific Instruction-Set Router for NoC-based MPSoCs*. Computers, MDPI.

Processing in Networks-on-Chip



Rettkowski J, Göhringer D (2018) *ASIR: Application-Specific Instruction-Set Router for NoC-based MPSoCs*. Computers, MDPI.

Example Image Processing Application (Grayscale conversion, Sobel, Threshold)

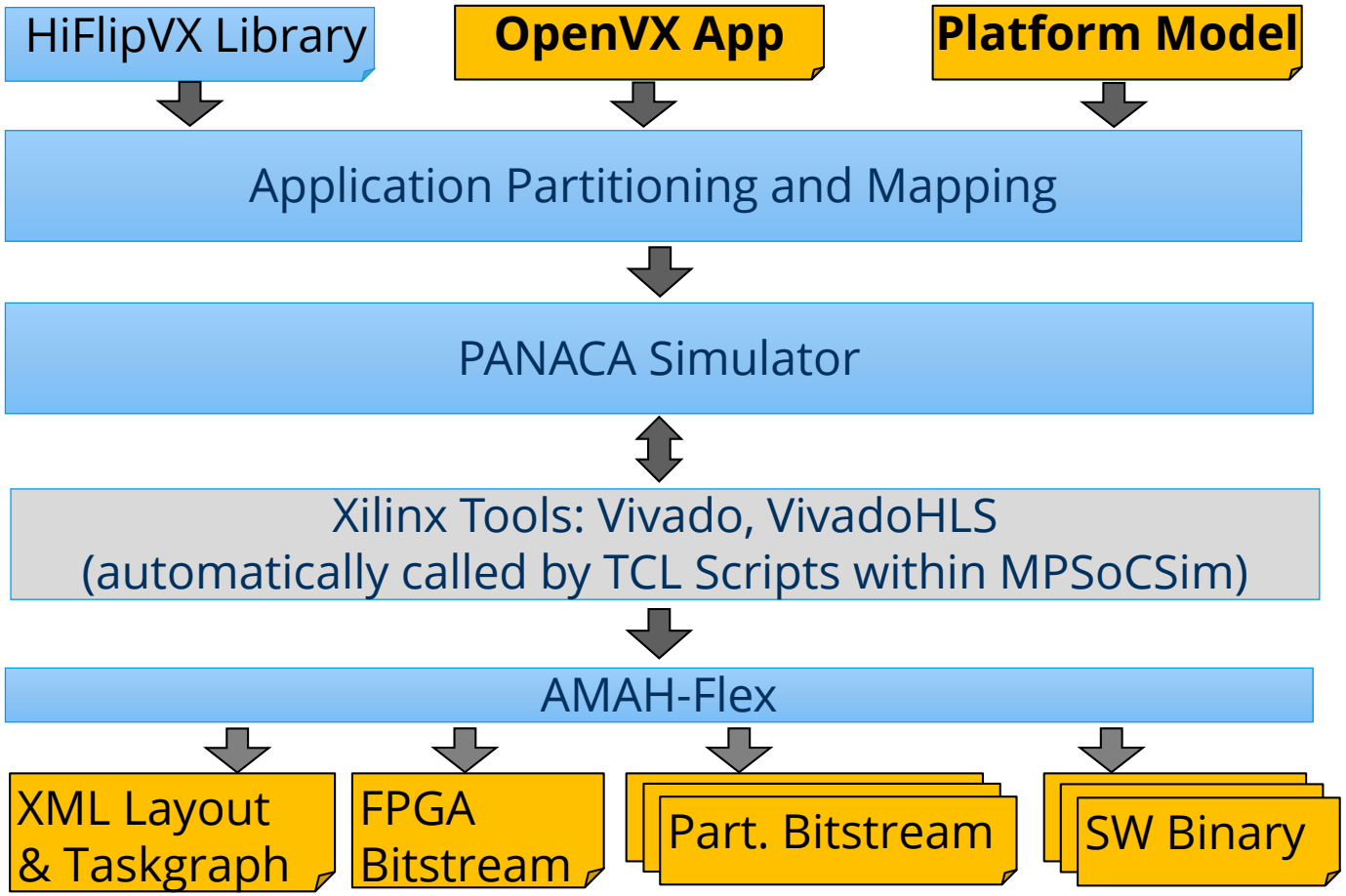


Rettkowski J, Göhringer D (2018) *ASIR: Application-Specific Instruction-Set Router for NoC-based MPSoCs*. Computers, MDPI.

Outline

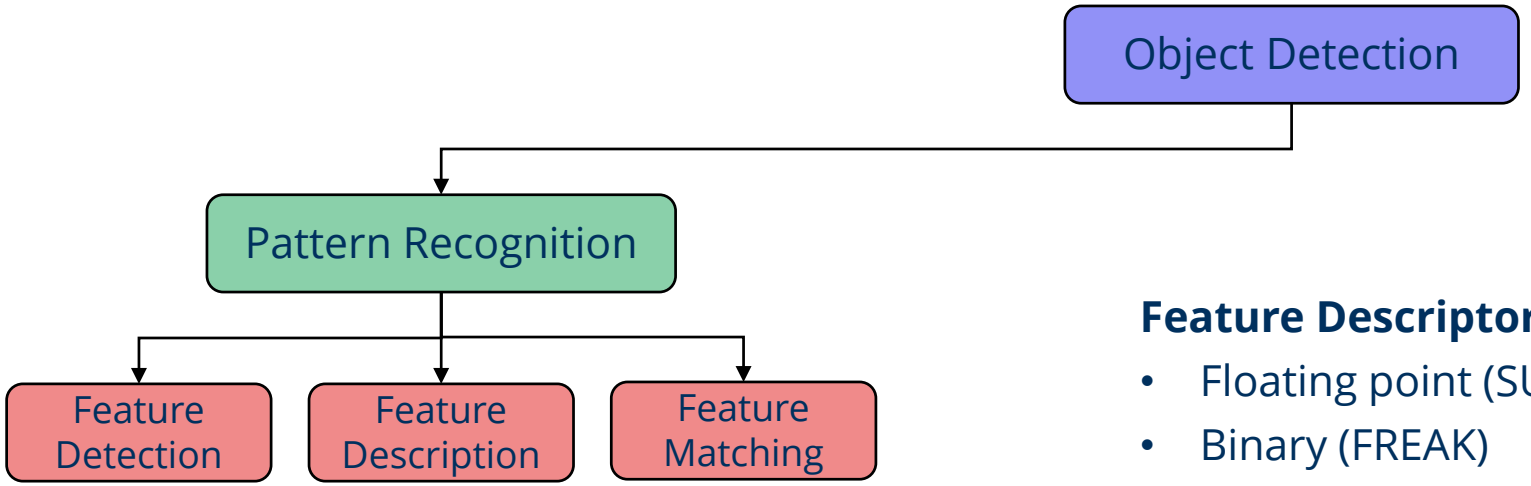
- Introduction and Motivation
- Reconfigurable Domain-Specific Computer Architectures
  - System Overview and Processing Elements
  - Reconfiguration Management
  - Network-on-Chip
- Design / Programming Methodology
  - High Level Synthesis Library: HiFlipVX
  - Application Partitioning and Mapping
- Application Examples in Relation to Funded Research Projects
- Conclusion and Outlook

Design and Programming Methodology



- Kalms L, Rad PA, Ali M, Iskander A, Göhringer D (2021) A Parametrizable High-Level Synthesis Library for Accelerating Neural Networks on FPGAs. Journal of Signal Processing, pp. 1-17.
- Kalms L, Göhringer D (2018) Scalable Clustering and Mapping Algorithm for Application Distribution on Heterogeneous and Irregular FPGA Clusters. Journal of Parallel and Distributed Computing (JPDC).
- Haase J, Groß A, Feichter M, Göhringer D (2022) PANACA: An Open-Source Configurable Network-on-Chip Simulation Platform. In Proc. of the 35th Symposium on Integrated Circuits and Systems Design (SBCCI), pp. 1-6.
- Charaf N, Tietz C, Raitza M, Kumar A,, Göhringer D (2021) AMAH-FLEX: A Modular and Highly Flexible Tool for Generating Relocatable Systems on FPGAs. In Proc. of the International Conference on Field-Programmable Technology (FPT). pp.1-6.

Hardware Accelerators for Image Processing and Machine Learning



Feature Descriptors

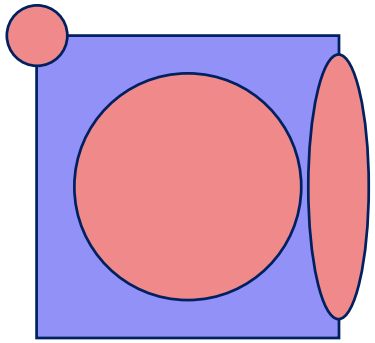
- Floating point (SURF)
- Binary (FREAK)

Feature Matching

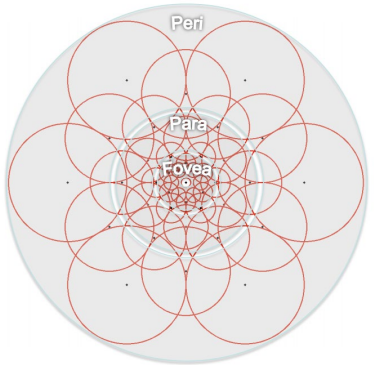
- Object recognition
- Object tracking
- SLAM (Simultaneous Localization and Mapping)

Feature Detectors

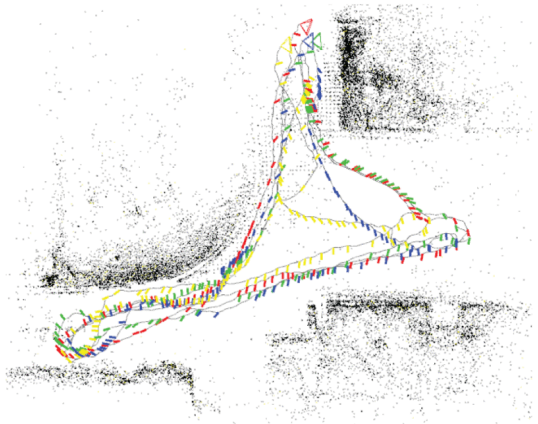
- Corner detector (ORB)
- Edge detector (Canny)
- Blob detector (AKAZE)



Object

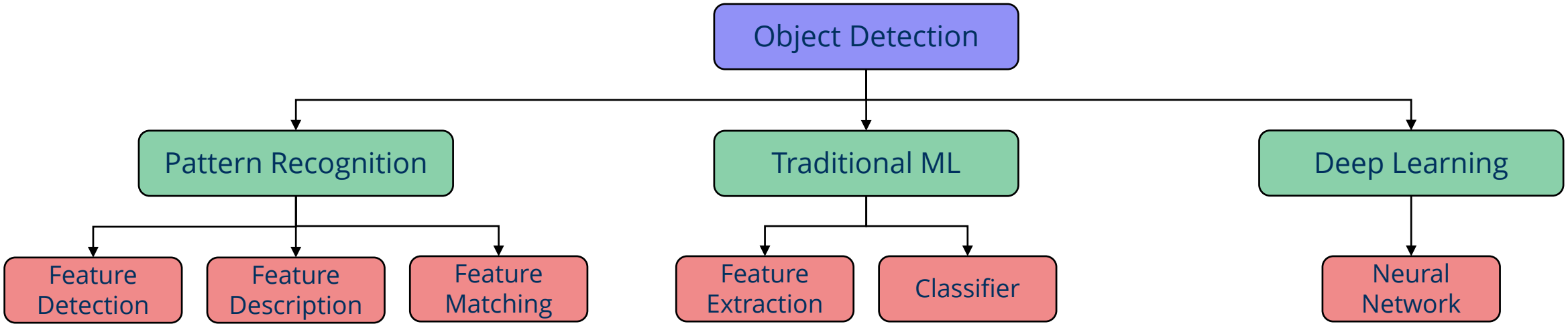


Retina (FREAK)
[Source:] Alahi et al., CVPR 2012



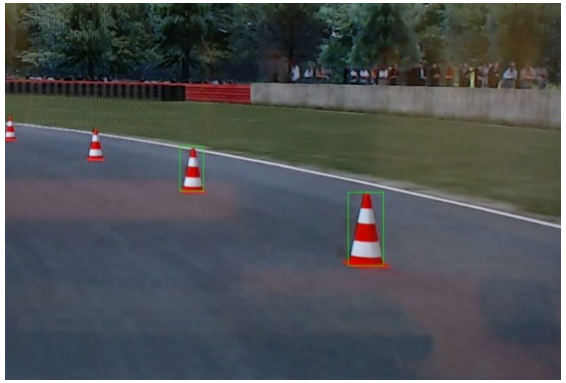
CoSLAM
[Source:] Zou et al., TPAMI 2013

Hardware Accelerators for Image Processing and Machine Learning



Traditional ML

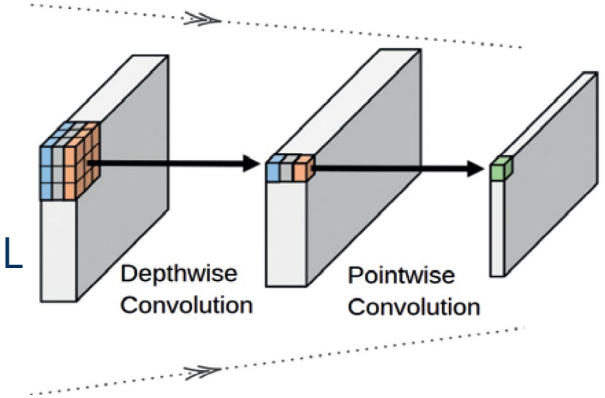
- Extracts features
- Trains classifiers
- E.g., Viola & Jones



Cone detection for autonomous driving

Deep Learning

- Based on neural networks
- No feature engineering
- Outperforms traditional ML
- Needs a lot of data



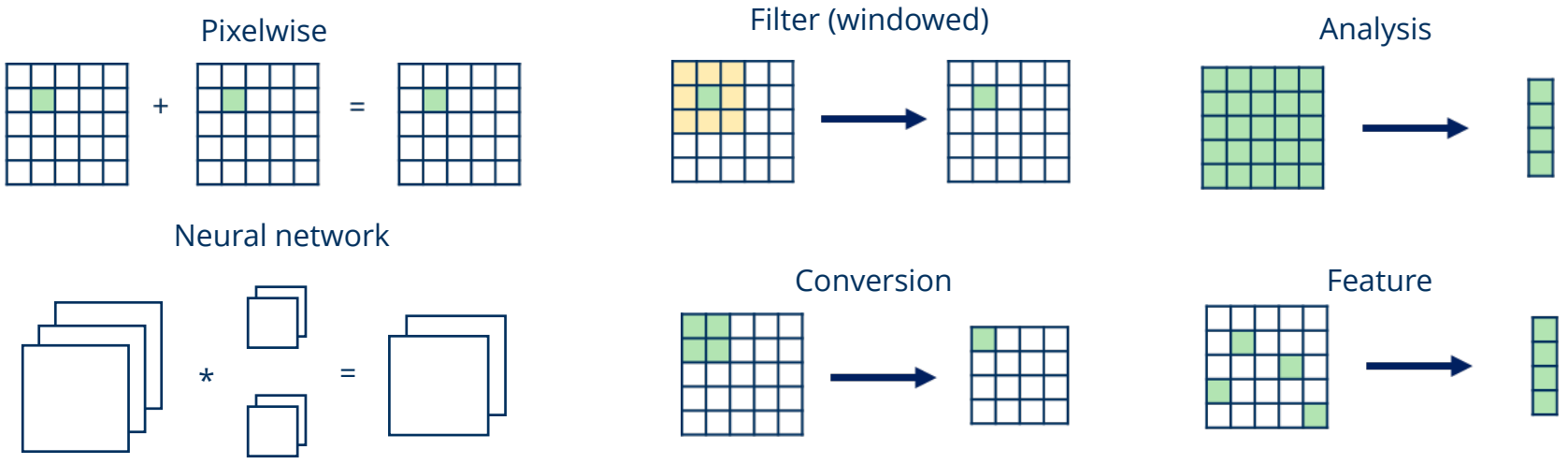
MobileNets V1

Source: Enkvetchakul et al., ASEP 2021

Hardware Accelerators for Image Processing and Machine Learning

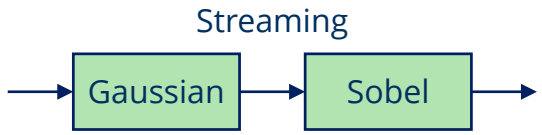
Open-source high-level synthesis FPGA library for image processing (HiFlipVX)

- Includes image processing and neural network functions
- Parametrizable and highly optimized for High-Level Synthesis (HLS)
- Functions are based on the OpenVX specification with some extensions
- Supports auto-vectorization & more data types
- Filters support different kernel sizes
- Portable to other vendors, e.g. AMD-Xilinx and Intel



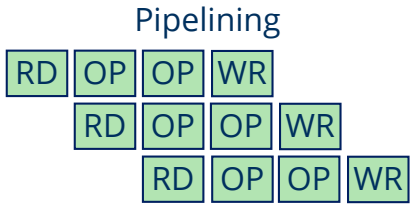
- Dávila-Guzmán M A, Kalms L, Gran Tejero R, Villarroya-Gaudo M, Suárez Gracia D, Göhringer D (2022) A Cross-Platform OpenVX Library for FPGA Accelerators. Journal of System Architecture.
- Kalms L, Rad PA, Ali M, Iskander A, Göhringer D (2021) A Parametrizable High-Level Synthesis Library for Accelerating Neural Networks on FPGAs. Journal of Signal Processing.
- Kalms L, Podlubne A, Göhringer D (2019) HiFlipVX: an Open Source High-Level Synthesis FPGA Library for Image Processing, Int. Symposium on Applied Reconfigurable Computing. Architectures, Tools, and Applications (ARC).

High-Level Synthesis Library - HiFlipVX



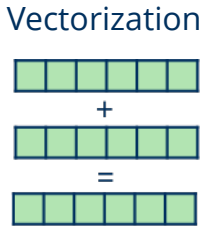
High Parameterizability

- Additional parameters and more options
- To improve flexibility and usability and design space exploration



Performance Optimized

- Use streaming (TLP), pipelining (ILP), vectorization (DLP), buffers and windows
- Vectorization with DVFS to improve energy efficiency

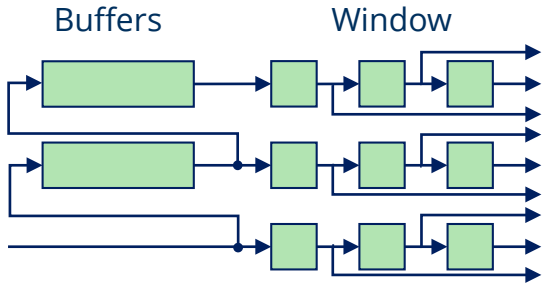


Resource Efficiency

- Consumed in average 39% FFs and 32% LUTs in comparison to xfOpenCV

Portability

- No external/vendor libraries (only XILINX directives)
- Partly ported to Intel FPGAs



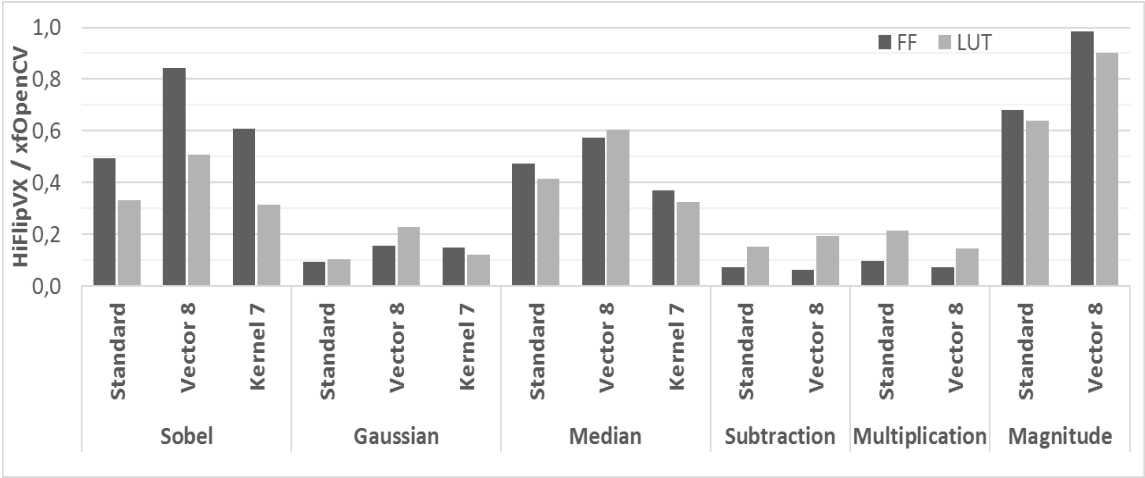
High-Level Synthesis Library - HiFlipVX

Features:

- Good Scalability
- Low resource usage
- Good comparison to related work

Standard configuration is:

- Vectorization: 1
- Kernel size: 1
- Data type: 8-bit



Comparison of HiFlipVX and xfOpenCV

Kalms L, Podlubne A, Göhringer D (2019) HiFlipVX: An Open Source High-Level Synthesis FPGA Library for Image Processing. In Proc. of Applied Reconfigurable Computing (ARC).

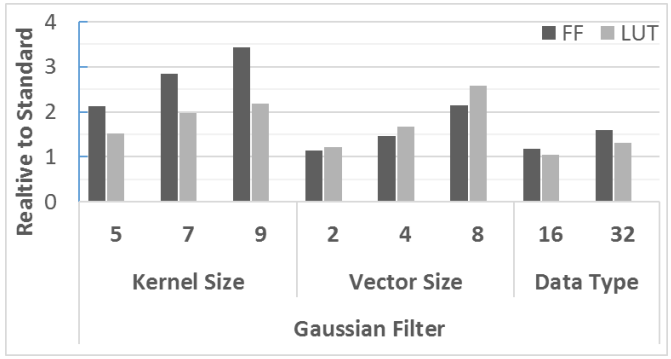
High-Level Synthesis Library - HiFlipVX

Features:

- Good Scalability
- Low resource usage
- Good comparison to related work

Standard configuration is:

- Vectorization: 1
- Kernel size: 1
- Data type: 8-bit



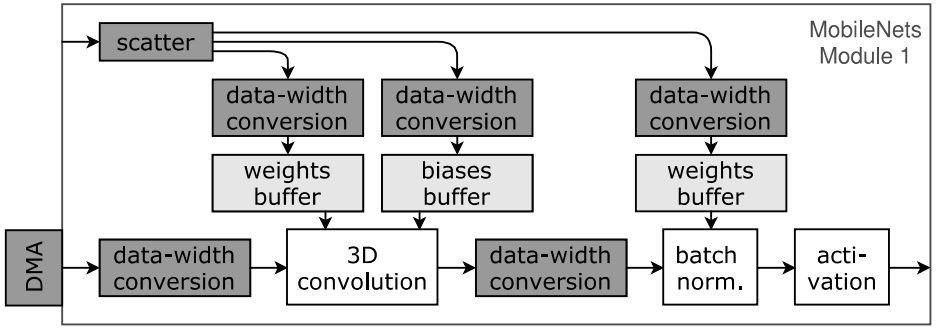
Scalability for different parameters

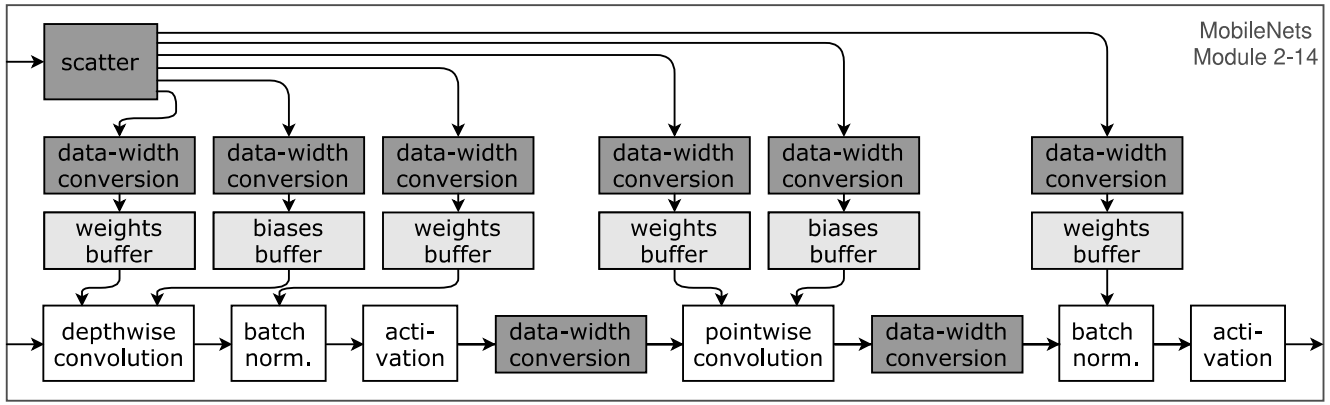
Function	FF	LUT	DSP	BRAM
Box Filter	257	536	2	2
Gaussian Filter	257	624	0	2
Sobel Filter	292	758	0	2
Median Filter	490	1180	0	2
AND, XOR,OR, Add, Subtract	27	171	0	0
Pixel-wise Multiplication	27	156	1	0
Magnitude	345	1106	0	0
Color Convert (RGBX to Gray)	50	258	2	0
Integral	82	389	0	4
Table Lookup (8-bit)	52	293	0	1
Histogram (8-bit)	113	593	0	2

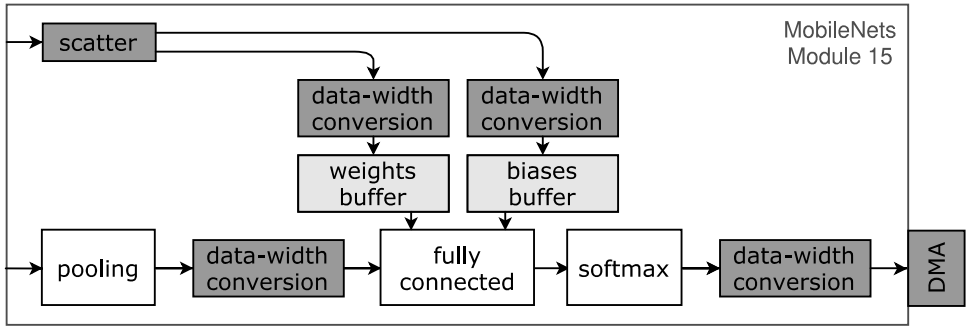
Low resource usage

Kalms L, Podlubne A, Göhringer D (2019) HiFlipVX: An Open Source High-Level Synthesis FPGA Library for Image Processing. In Proc. of Applied Reconfigurable Computing (ARC).

HiFlipVX: Implementation of MobileNets





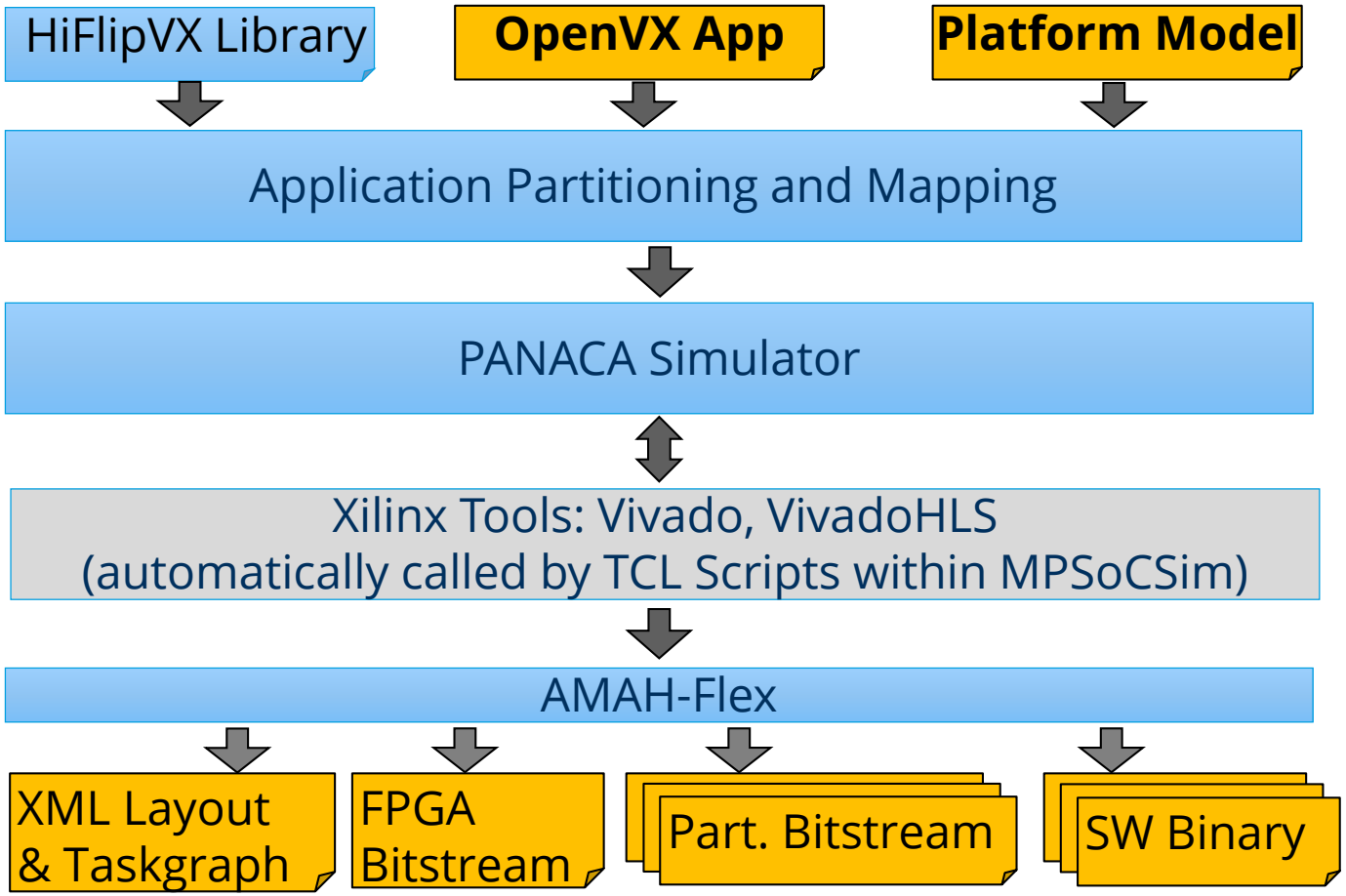


Results for a ZCU104. ARM processor is running at 1.2GHz and FPGA at 200MHz:

	Module 1	Module 2	Module 15
ARM (ms)	34.17	53.22	9.94
FPGA (ms)	0.97	0.90	0.49
Speedup	35.38	58.99	20.32
LUT	11 881	16 914	10 579
FF	13 265	16 660	5773
DSP	237	140	27
BRAM	1	20	263.5

Kalms L, Rad PA, Ali M, Iskander A, Göhringer D (2021) A Parametrizable High-Level Synthesis Library for Accelerating Neural Networks on FPGAs. Journal of Signal Processing, pp. 1-17.

Design and Programming Methodology



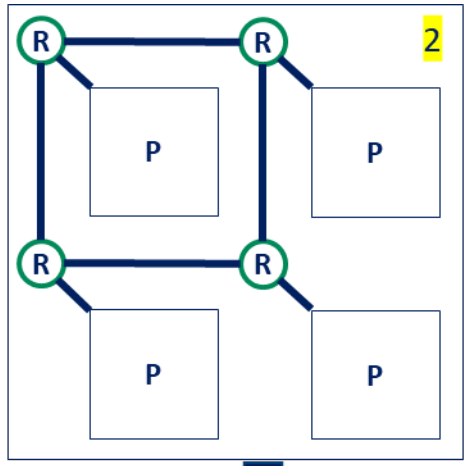
- Kalms L, Rad PA, Ali M, Iskander A, Göhringer D (2021) A Parametrizable High-Level Synthesis Library for Accelerating Neural Networks on FPGAs. Journal of Signal Processing, pp. 1-17.
- Kalms L, Göhringer D (2018) Scalable Clustering and Mapping Algorithm for Application Distribution on Heterogeneous and Irregular FPGA Clusters. Journal of Parallel and Distributed Computing (JPDC).
- Haase J, Groß A, Feichter M, Göhringer D (2022) PANACA: An Open-Source Configurable Network-on-Chip Simulation Platform. In Proc. of the 35th Symposium on Integrated Circuits and Systems Design (SBCCI), pp. 1-6.
- Charaf N, Tietz C, Raitza M, Kumar A,, Göhringer D (2021) AMAH-FLEX: A Modular and Highly Flexible Tool for Generating Relocatable Systems on FPGAs. In Proc. of the International Conference on Field-Programmable Technology (FPT). pp.1-6.

Application Partitioning and Mapping

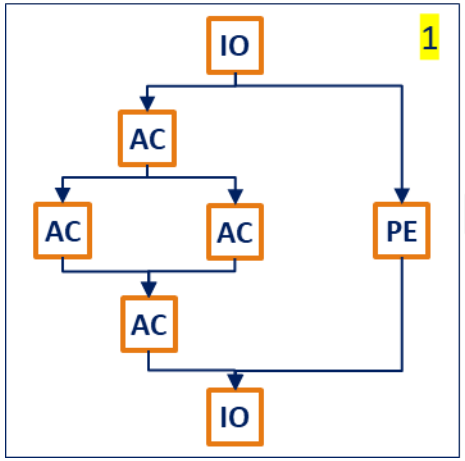
- Clustering & placement in 1 step
- Partitioning (task creation) is done by the developer
- Clustering and mapping are known as NP-hard for irregular graphs
- E.g. 9 partitions and 24 tasks would result in 9^{24} different mapping solutions
- Therefore, we use load balancing and heuristics

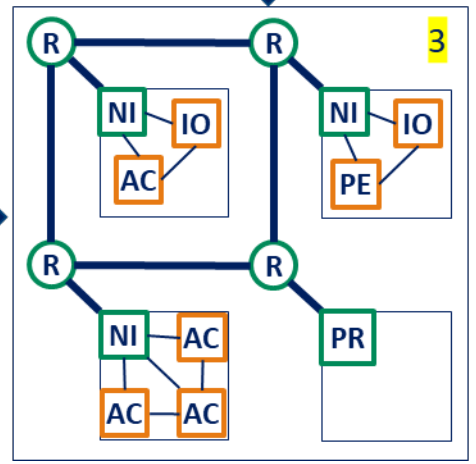
Kalms L, Göhringer D (2018) Scalable Clustering and Mapping Algorithm for Application Distribution on Heterogeneous and Irregular FPGA Clusters. Journal of Parallel and Distributed Computing (JPDC).

Platform model



Task Interaction Graph (TIG)

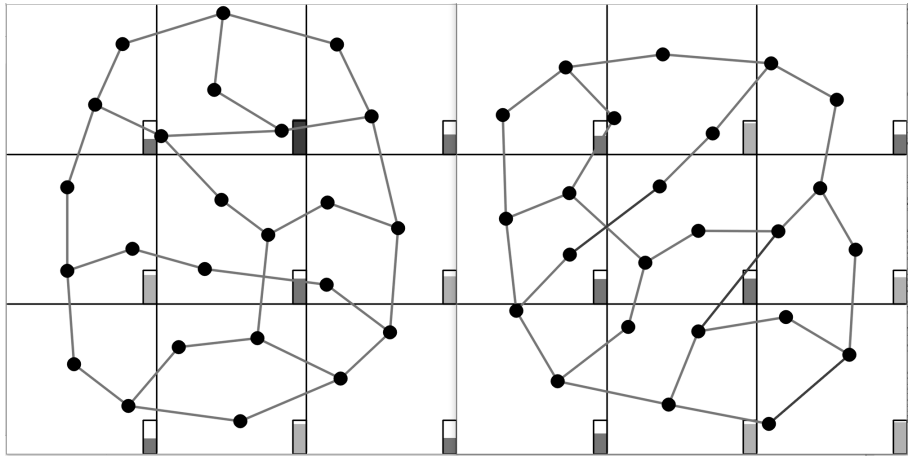




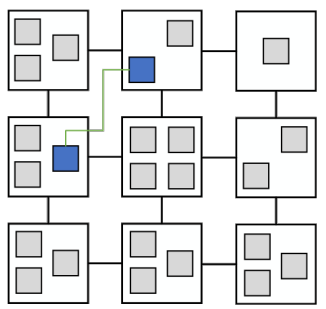
Application Partitioning and Mapping

Algorithm overview:

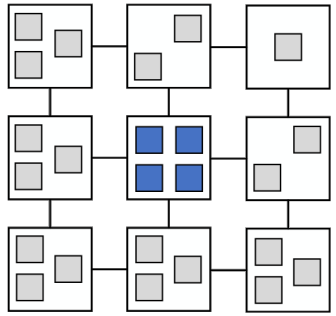
- Load Balancing
  - Connected and unweighted edges are forced together like a rubber
  - All tasks are forced apart from each other based on their capacity utilization
- Optimization
  - Reduce maximum dilation between tasks
  - Reduce maximum partition capacity utilization of clusters
- Used optimization heuristics:
  - Gradient Descent, Taboo Search, Simulated Annealing, Parallelization



Guest graph load balanced into host graph for two threads. Bars show maximum resource usage of partitions.



Dilation Optimization



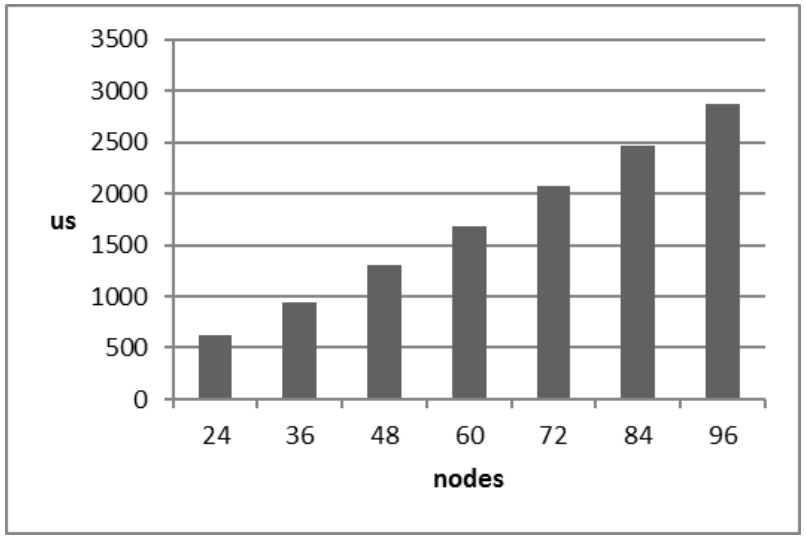
Capacity Utilization Optimization

Kalms L, Göhringer D (2018) Scalable Clustering and Mapping Algorithm for Application Distribution on Heterogeneous and Irregular FPGA Clusters. Journal of Parallel and Distributed Computing (JPDC).

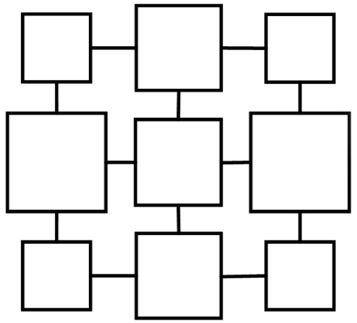
Application Partitioning and Mapping

Evaluation:

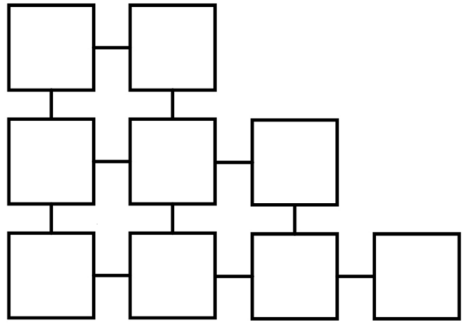
- Support different topologies
- Fast computation for mapping



Computation time of different TIGs for 3 × 3 Partitions that has a homogeneous and regular topology



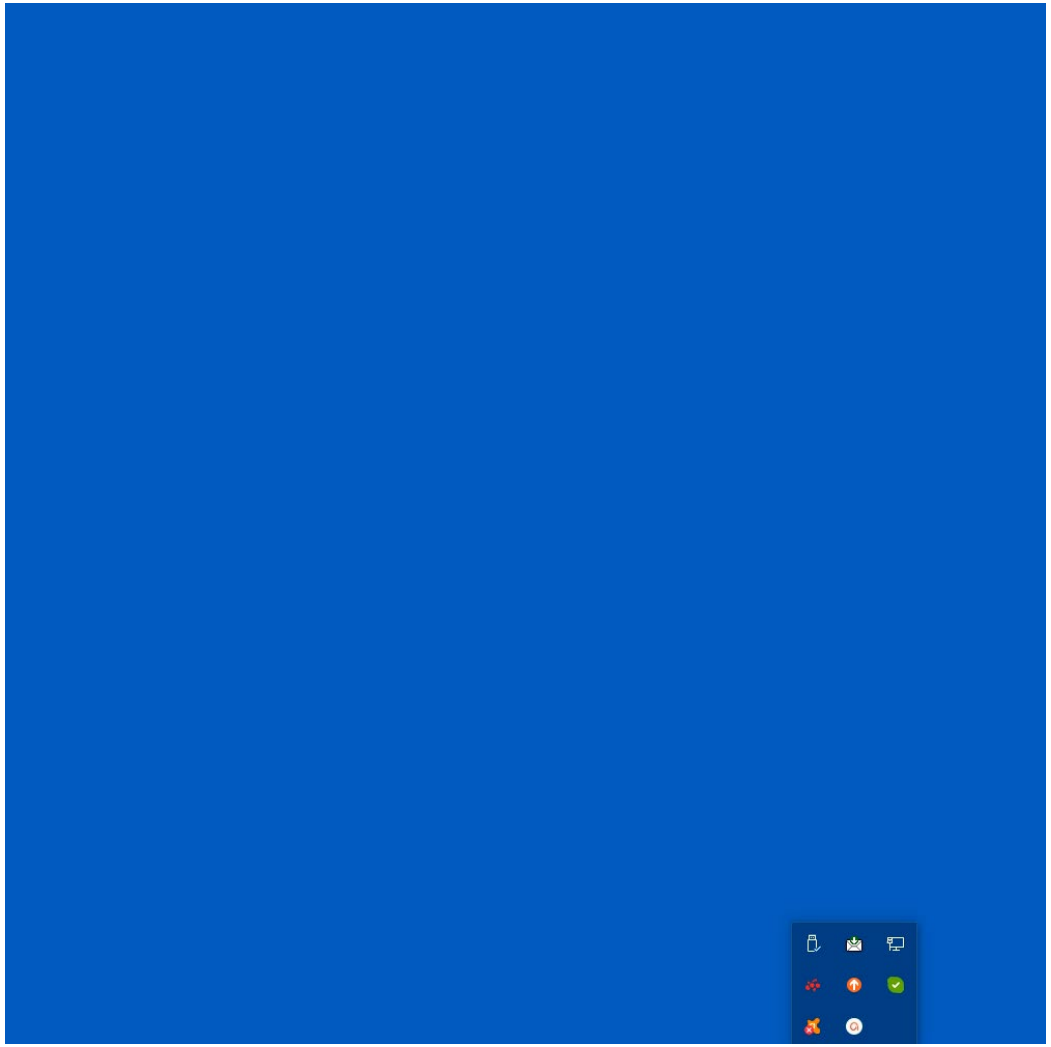
Heterogeneous



Irregular

Kalms L, Göhringer D (2018) Scalable Clustering and Mapping Algorithm for Application Distribution on Heterogeneous and Irregular FPGA Clusters. Journal of Parallel and Distributed Computing (JPDC).

Application Distribution Video



Outline

- Introduction and Motivation
- Reconfigurable Domain-Specific Computer Architectures
  - System Overview and Processing Elements
  - Reconfiguration Management
  - Network-on-Chip
- Design / Programming Methodology
  - High Level Synthesis Library: HiFlipVX
  - Application Partitioning and Mapping
- Application Examples in Relation to Funded Research Projects
- Conclusion and Outlook

DFG SFB/Transregio MARIE (2017 – 2024)

Mobile Material Characterization and Localization by Electromagnetic Sensing

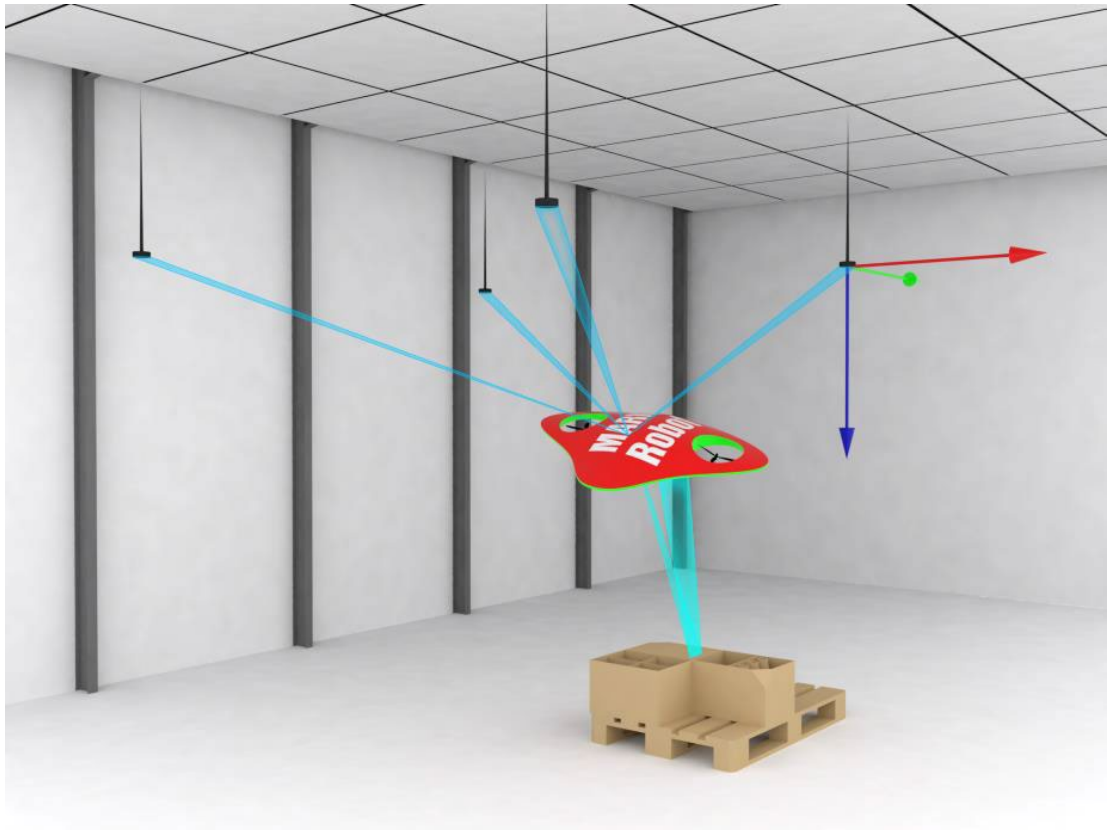
Challenges:

1) THz wave propagation measurement, analyzation and modeling
2) Small sub-mm-wave transceivers
3) Material characterization
4) Material localization

Our Contributions:

- ➢ Domain-specific computing architecture
- ➢ Design and programming methodology

→ For enabling a real-time material map



DFG SFB/Transregio MARIE (2017 – 2024)

Mobile Material Characterization and Localization by Electromagnetic Sensing

Challenges:

1) THz wave propagation measurement, analyzation and modeling
2) Small sub-mm-wave transceivers
3) Material characterization
4) Material localization

Our Contributions:

- ➢ Domain-specific computing architecture
- ➢ Design and programming methodology

→ For enabling a real-time material map

Cluster of Excellence CeTI (2019 – 2025)

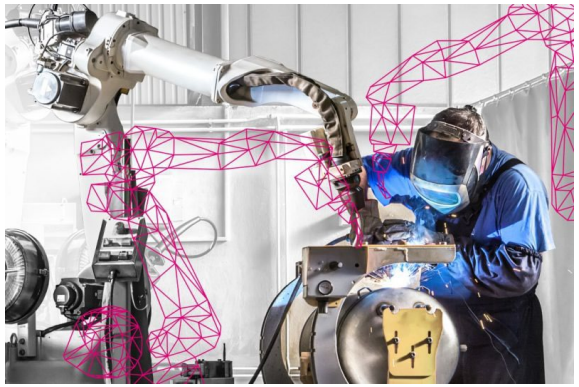
Centre for Tactile Internet with Human-in-the-Loop

Cluster of Excellence CeTI (2019 – 2025)

Centre for Tactile Internet with Human-in-the-Loop

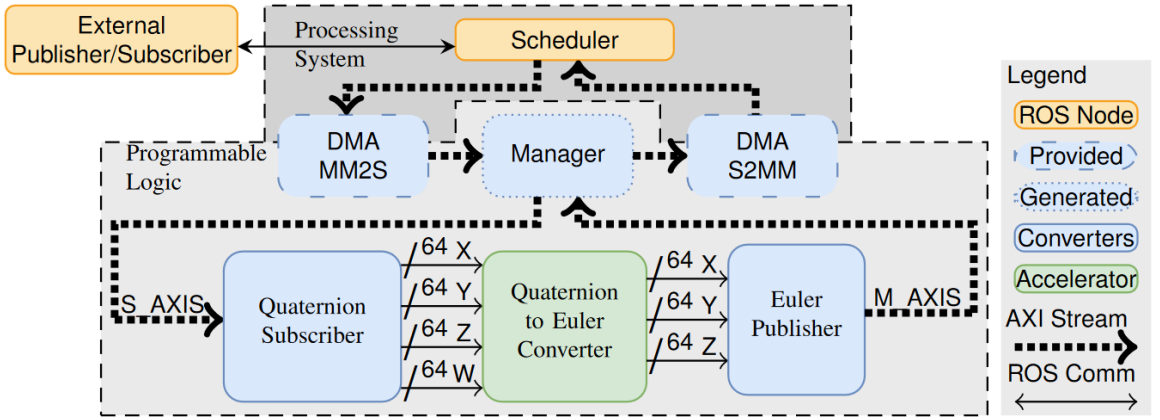
Challenge:

Connect humans with tactile robots to collectively learn millions of manipulation skills.



Our Contributions:

- Reconfigurable computing architecture for robotics
- Model-based tool chain for reconfigurable computing architectures



Computer architecture for Quaternion to Euler conversion with ROS interfaces

Podlubne A, Mey J, Pertuz S, Aßmann U, Göhringer D (2022) Model-based Generation of Hardware/Software Architectures for Robotics Systems. In Proc. of the 32nd International Conference on Field Programmable Logic and Applications (FPL).

Cluster of Excellence CeTI (2019 – 2025)

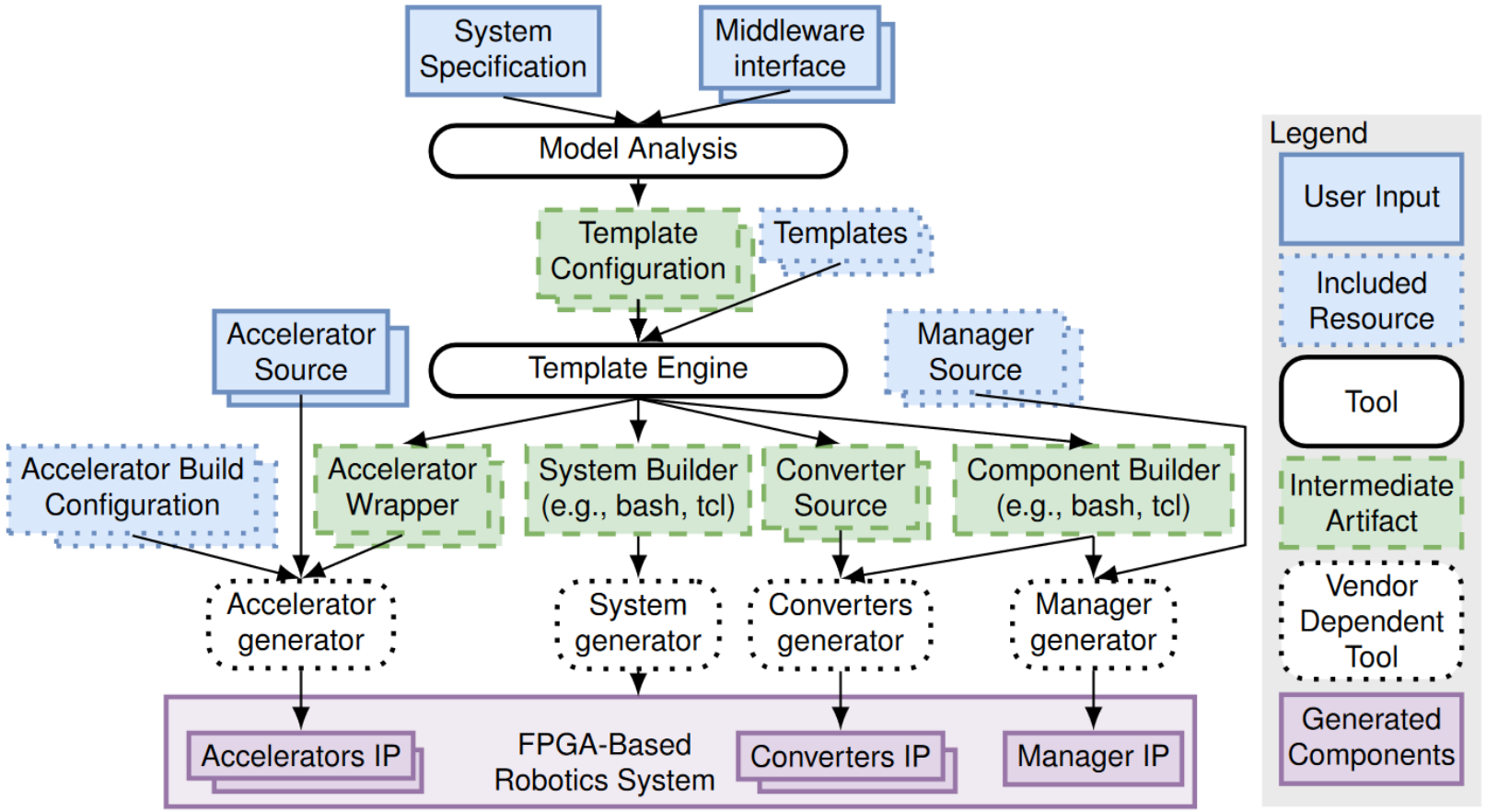
Centre for Tactile Internet with Human-in-the-Loop

Challenge:

Connect humans with tactile robots to collectively learn millions of manipulation skills.

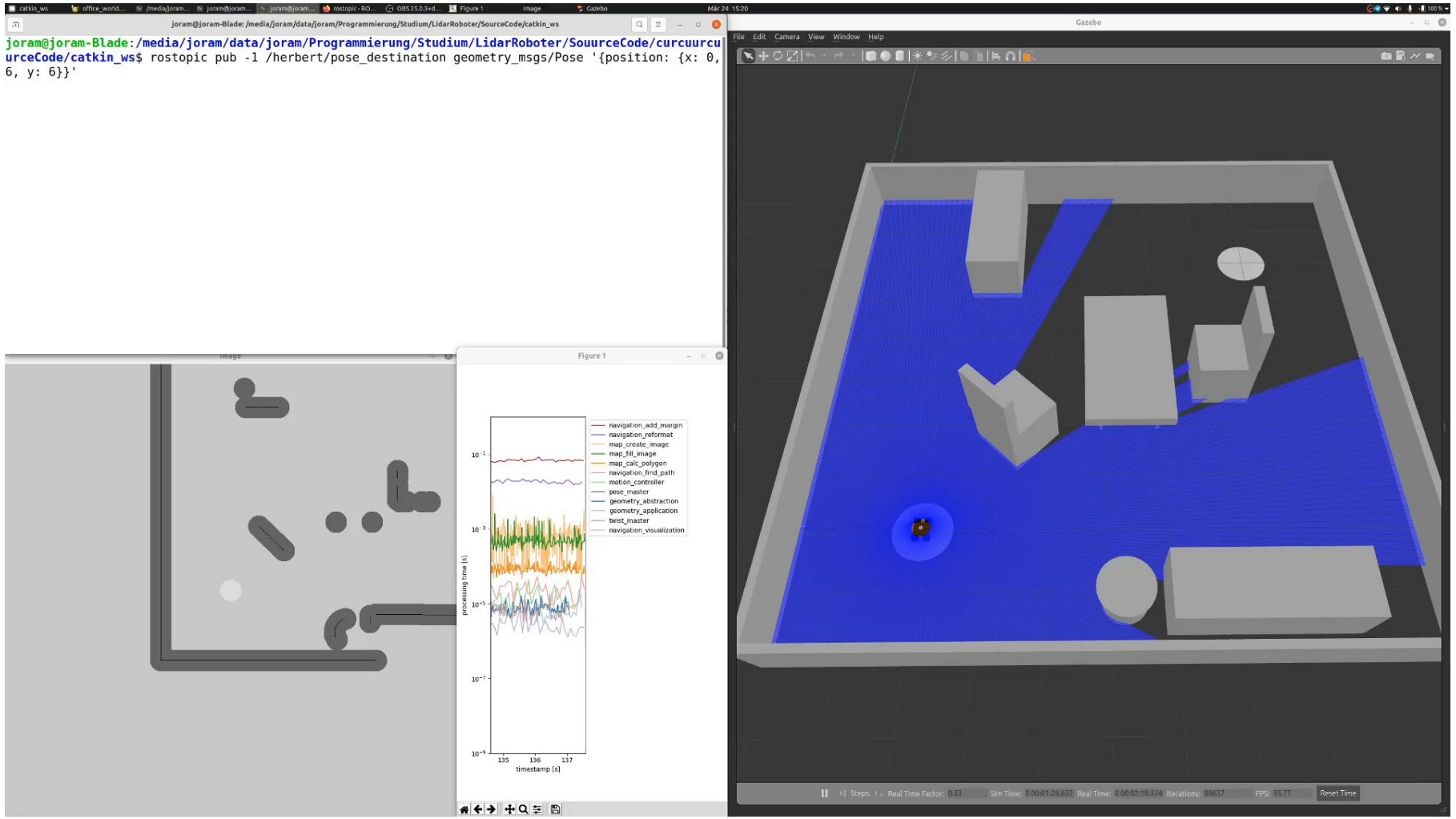
Our Contributions:

- Reconfigurable computing architecture for robotics
- Model-based tool chain for reconfigurable computing architectures



Podlubne A, Mey J, Pertuz S, Aßmann U, Göhringer D (2022) Model-based Generation of Hardware/Software Architectures for Robotics Systems. In Proc. of the 32nd International Conference on Field Programmable Logic and Applications (FPL).

Obstacle Avoidance with LiDAR and ROS

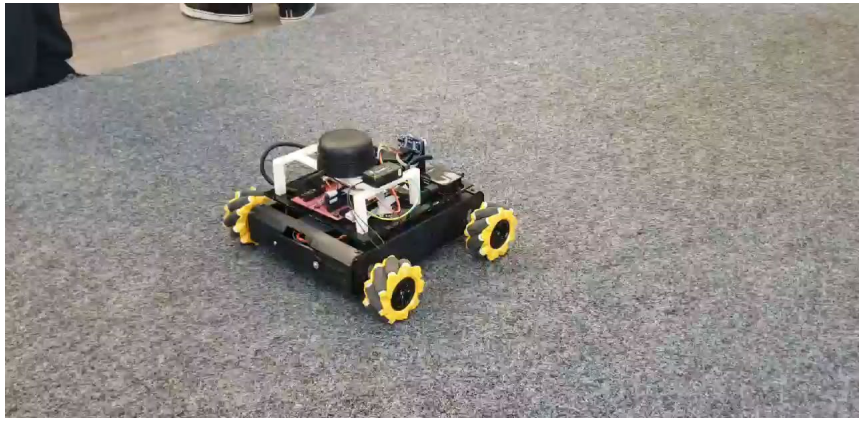


Cluster of Excellence CeTI (2019 – 2025)

Centre for Tactile Internet with Human-in-the-Loop

Our Contributions:

- Reconfigurable computing architecture for robotics
- Model-based tool chain for reconfigurable computing architectures



Cluster of Excellence CeTI (2019 – 2025)

Centre for Tactile Internet with Human-in-the-Loop

Hello Robot Demo @ Hannover Messe 2022:

Outline

- Introduction and Motivation
- Reconfigurable Domain-Specific Computer Architectures
  - System Overview and Processing Elements
  - Reconfiguration Management
  - Network-on-Chip
- Design / Programming Methodology
  - High Level Synthesis Library: HiFlipVX
  - Application Partitioning and Mapping
- Application Examples in Relation to Funded Research Projects
- Conclusion and Outlook

Summary and Outlook

Summary: Reconfigurable FPGA Overlay Architecture

- High flexibility → HW and SW can be adapted at design- and runtime to the application requirements, such as data throughput, real-time, safety
- High performance → Domain-specific architecture
- High energy efficiency → On demand functionality, better area utilization by hardware task multiplexing
- Programming support → Tool support for application partitioning and mapping, HiFlipVX

Outlook:

- Development and support of other/new:
  - Architecture components
  - Design methods and tools, new library functions for HiFlipVX
  - Runtime management systems
- Evaluation with further applications

Thank you! Questions?

Contact:

Prof. Dr.-Ing. Diana Göhringer

Chair of Adaptive Dynamic Systems
TU Dresden, Germany

Email: diana.goehringer@tu-dresden.de
www: http://www.tu-dresden.de/inf/ads