AMPERE

A Model-driven development framework for highly Parallel and EneRgy-Efficient computation supporting multi-criteria optimisation

15th Workshop on Virtualization in High-Performance Cloud Computing (VHPC'20, part of ISC 2020)

ISC High Performance
The HPC Event.

Model-based engineering of high-performance embedded applications on heterogeneous hardware with real-time constraints and energy efficiency

Tommaso Cucinotta – Scuola Superiore Sant'Anna, Pisa (Italy)

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

isep Instituto Superior de Engenharia do Porto

Scuola Superiore Sant'Anna
di Studi Universitari e di Perfezionamento

ETH zürich

SYSGO
EMBEDDING INNOVATIONS

THALES

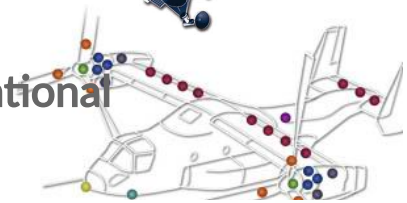
BOSCH

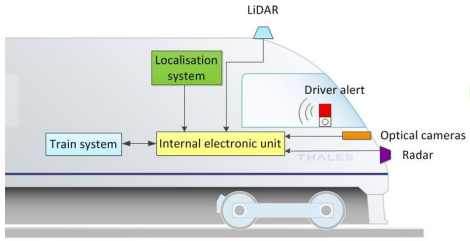
EVIDENCE
EMBEDDING TECHNOLOGY

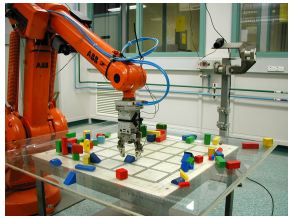
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 871669

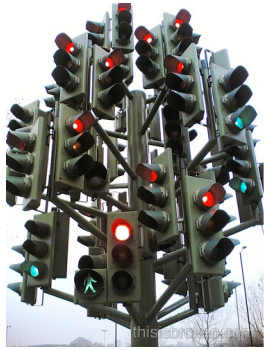
Introduction & Motivations

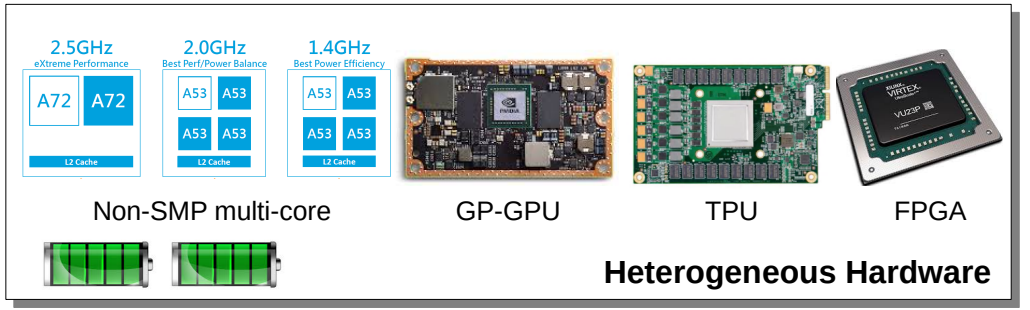
- CPSs have higher and higher computational performance & reliability requirements
- Use of increasingly heterogeneous & interconnected, battery-operated platforms
  - non-SMP multi-core
  - GP-GPU/TPU acceleration
  - FPGA
- Heterogeneous platforms needed in soft and hard real-time use-cases
  - automotive, railways, aerospace, robotics, gaming, multimedia, ...





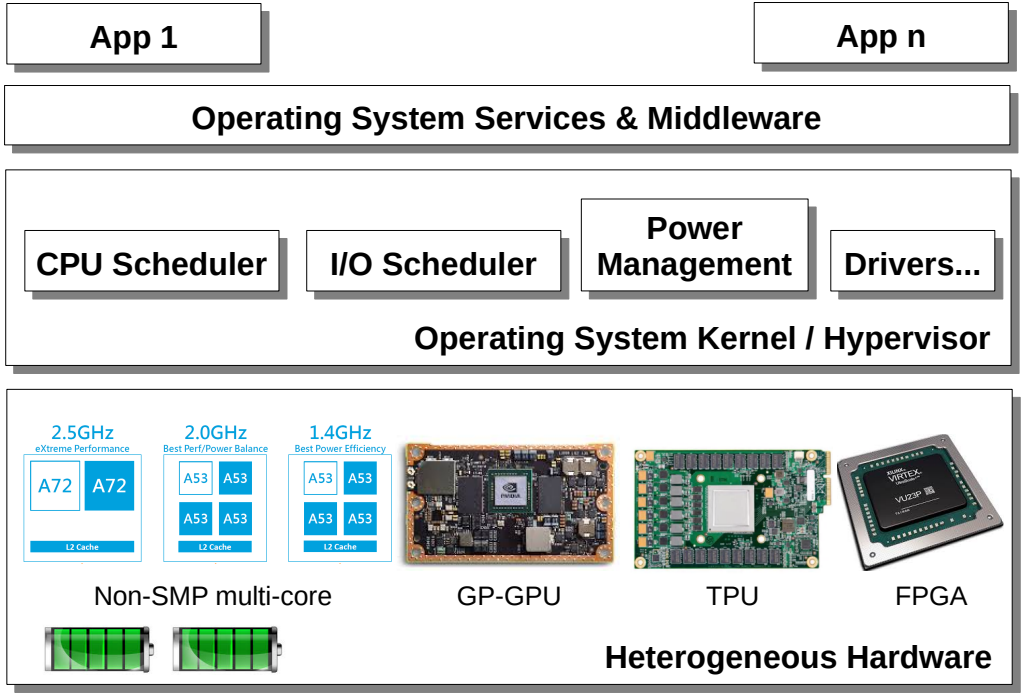






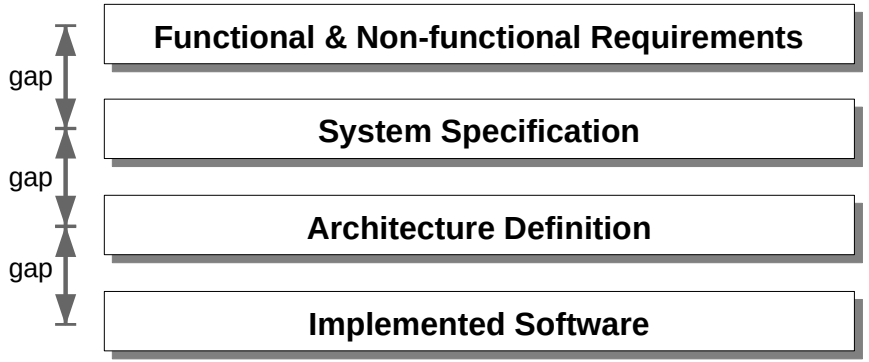
Problems & Challenges

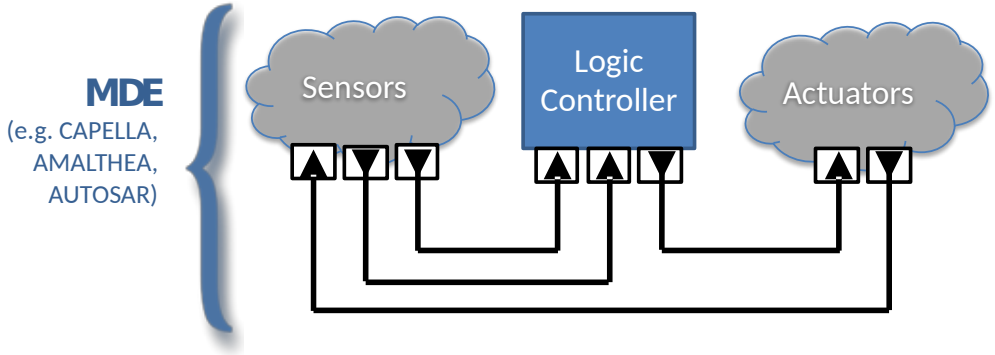
- Development of software for CPSs is cumbersome!
  - Optimum usage of underlying hardware parallelism & acceleration
  - Performance vs energy consumption trade-offs
  - Real-time constraints
  - Safety & certification



MDE & Formalisms in Embedded System Design

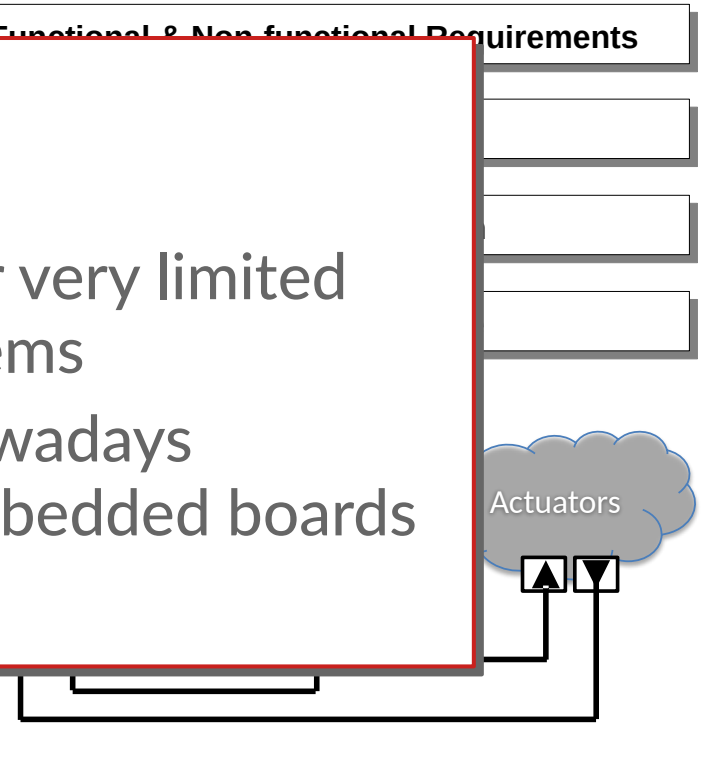
- Model-Driven Engineering (MDE)
  - Fill the gap between high-level specifications and actual system behavior
- MDE embraces
  - Formal specification language(s)
  - Model transformation engine(s)
  - Model refinements & composability
  - Automatic code generator(s)
  - Model verifiability
- => Correctness-by-construction





MDE & Formalisms in Embedded System Design

- Model-Dri
  - Fill the g
    high-lev
    behavio
- MDE embr
  - Formal s
  - Model t
  - Model r
  - Automa
  - Model v
- => Correctness-by-construction

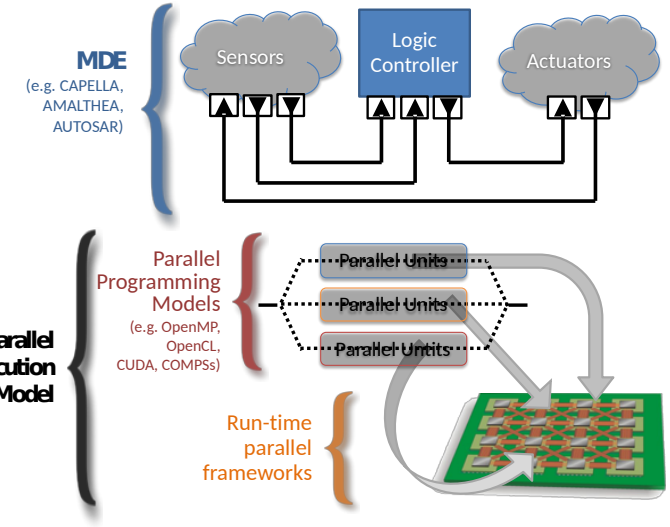


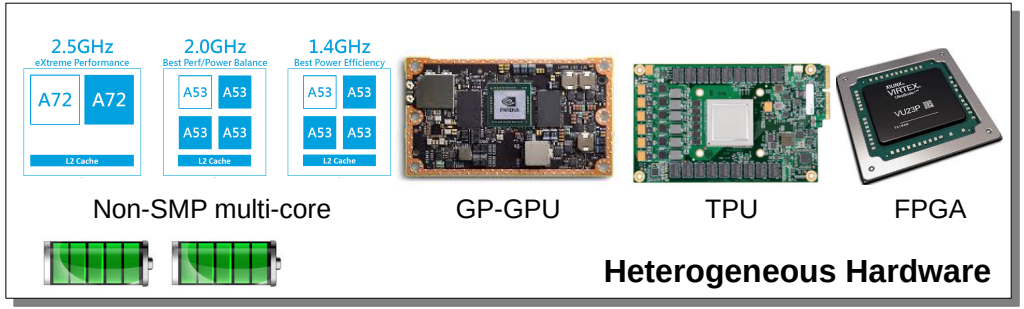
Traditional MDE limitations

- Single-processor systems or very limited support for multi-core systems
- Struggles at coping with nowadays complex heterogeneous embedded boards

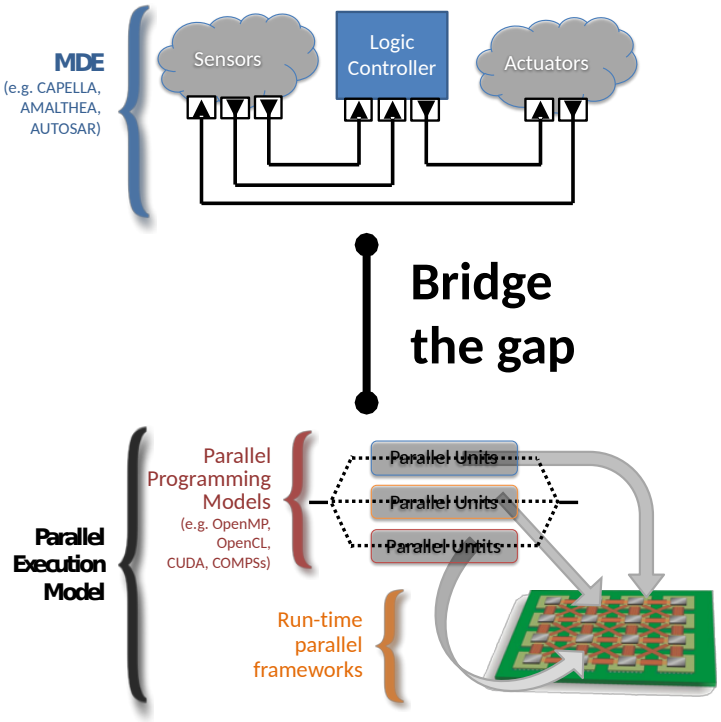
AMPERE Project Goal

- **Fill the gap between**
  - MDE techniques with no/limited parallelism support
  - Parallel-programming models with efficient HW offloading (OpenMP, CUDA, …)
  - Heterogeneity in hardware
- **In presence of non-functional requirements**
  - High-Performance
  - Real-Time Constraints
  - Energy Efficiency
  - Fault Tolerance



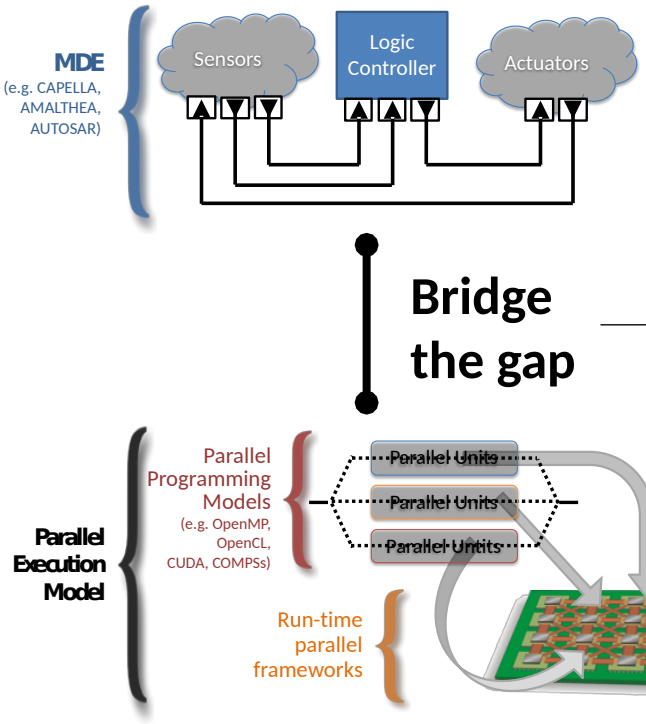


AMPERE Vision



1. **Synthesis methods** for an efficient generation of parallel source code, while keeping non-functional and composability guarantees
2. **Run-time parallel frameworks** that guarantee system correctness and exploit the performance capabilities of parallel architectures
3. **Integration** of parallel frameworks into MDE frameworks

AMPERE Vision



AUTOSAR
- SW-C
- Runnables
- Client-server
- ASIL

AMALTHEA
- Performance
- Tasks
- Scheduling
- Platform

CAPELLA
- Functional components
- Allocation of resources
- Data models
- View points
- validation rules

Meta-model Driven Abstractions
Components, Communications, Timing Characteristics, IntegrityAassurance, ...

Model Transformation Engine

Meta-parallel Programming Abstraction
Parallelism, Synchronization, Data Dependencies, Data Attributes, ...

OpenMP
- Task construct
- Dependencies
- Parallel construct

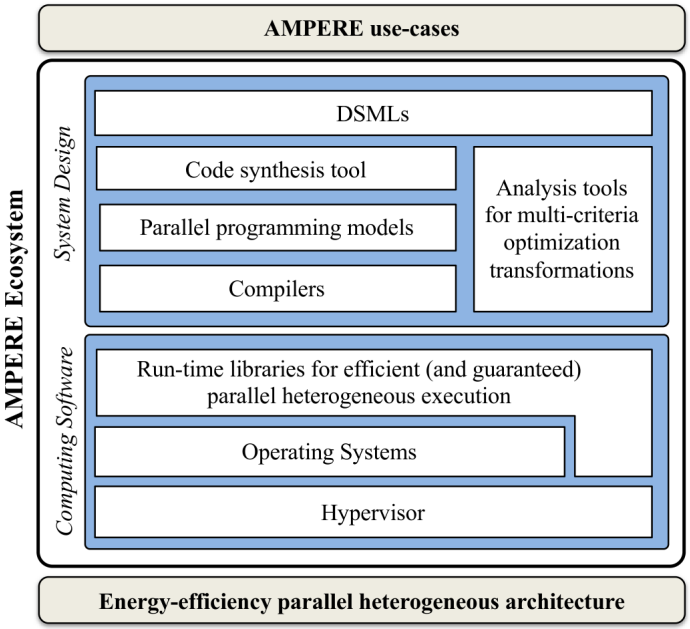
OpenCL
- clgetDeviceId
- clCreateBuffer
- __kernel_exec

COMPSs
- Compute resource
- Data movements
- Task annotations

Parallel Run-Time Frameworks

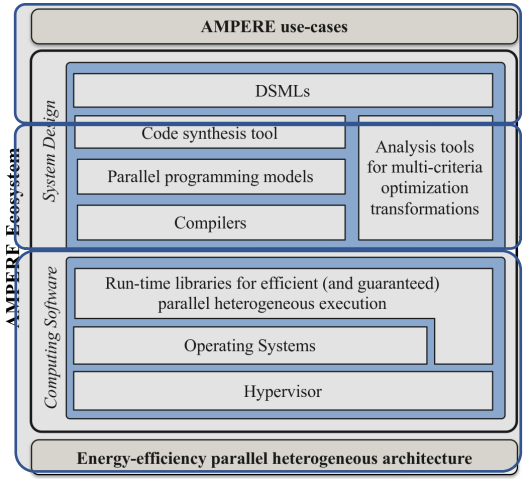
AMPERE MDE Framework

AMPERE Software Architecture



Software Layer	Tool	Owner (License)
DSMLs	**AUTOSAR**	AUTOSAR (Proprietary)
	AMALTHEA	BOSCH (Open-source)
	CAPELLA	TRT (Open-source)
Parallel programming models	**OpenMP**	OpenMP ARB (Proprietary)
	CUDA	NVIDIA (Proprietary)
	OpenCL	Khronos (Proprietary)
	COMPSs	BSC (Open-source)
Artificial Intelligence	TensorFlow	Google (Open-source)
Code synthesis tools	Synthesis tools	AMPERE (Open-source)
Analysis and testing tools	NFP analysis	AMPERE (Open-source)
Compilers and hardware synthesis tools	Mercurium	BSC (Open-source)
	GCC/LLVM	GNU/LLVM (Open-source)
	Vivado	Xilinx (Proprietary)
Run-time libraries	GOMP	GNU-GCC (Open-source)
	KMP	LLVM (Open-source)
	Vivado	Xilinx (Proprietary)
Operating systems	Linux	Linux-Foundation (Open-source)
	ERIKA Enterp.	EVI (Open-source/commercial)
Hypervisors	PikeOS	SYSGO (Proprietary)

AMPERE Software Development Workflow Overview



System description
- Components/communication
- Functional/NFP
- Etc.

Platform description
- Accel. devices
- Cores/clusters
- Memory model
- Etc.

Model

Meta MDE abstraction

Code Synthesis + Multi-criteria Optimization
- Performance
- Time-predictability
- Energy-efficiency
- Resiliency

Meta PPM abstraction

Compiler
(Correctness + Refined Parallel Structure)

Parallel code (e.g. OpenMP, CUDA graphs)

Resource Allocation
(i.e., mapping/scheduling)
- Monitoring
- Dynamic resource allocation

Run-time framework + OS + Hypervisor

Execution Profile

AMPERE Use-Cases

Obstacle Detection and Avoidance System (ODAS)

- ADAS functionalities based on data fusion coming from tram vehicle sensors

Predictive Cruise Control (PCC)

- Extends Adaptive Cruise Control (ACC) functionality by calculating the vehicle's future velocity curve using the data from the *electronic horizon*
- Improve fuel efficiency (in cooperation with the powertrain control) by configuring the driving strategy based on data analytics and AI

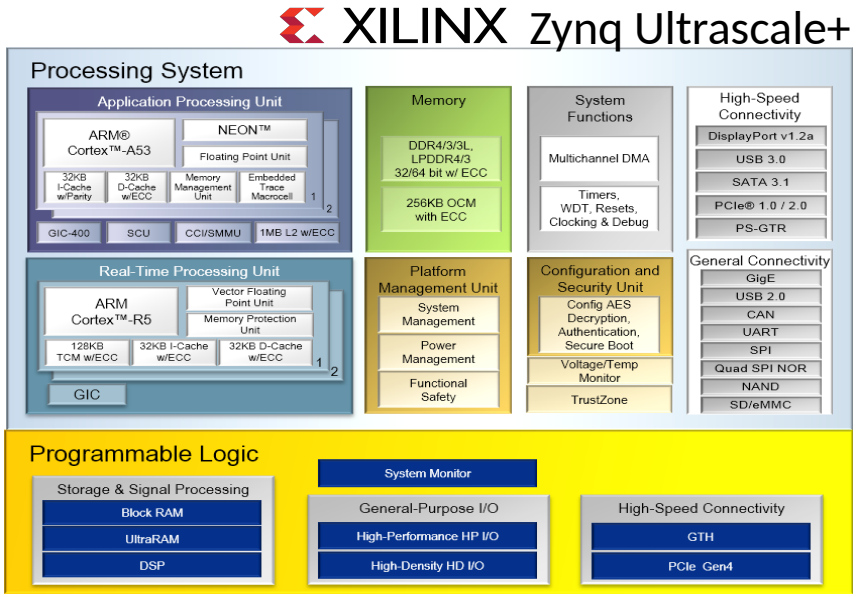
FPGA System-on-Chip (SoC)

FPGA-based system-on-chips are a very promising solution to enable **predictable** HW acceleration of complex computing workloads

- Multiprocessors can host multi-OS software systems
- FPGA fabric can be used to deploy HW accelerators

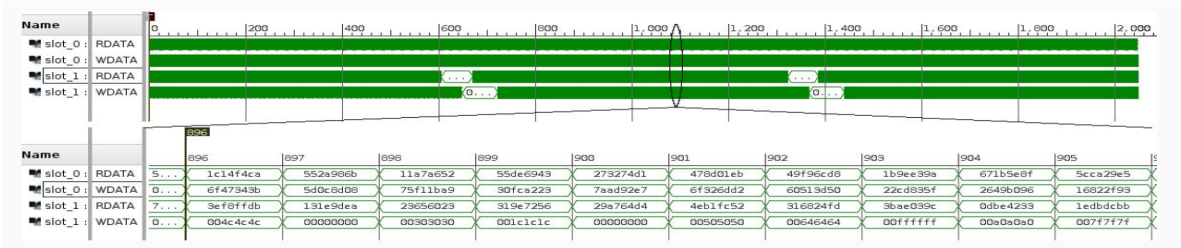
Asymmetric multiprocessor

Large FPGA fabric



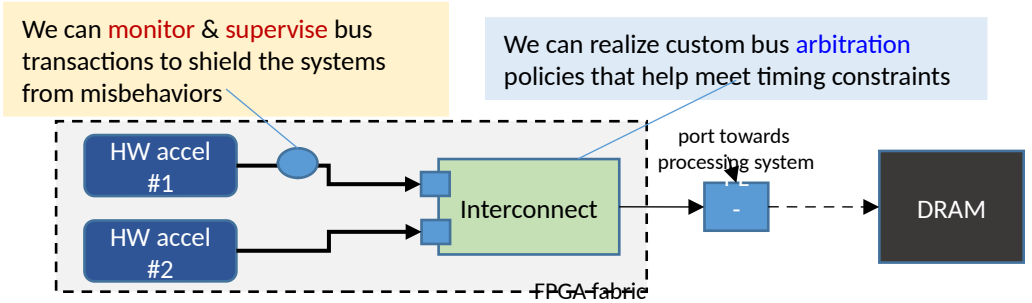
HW Accelerators on FPGAs

- Programmable logic exhibits very regular, clock-level behavior (differently from other HW accelerators, e.g., GP-GPUs)
- Internal control logic of several HW accelerators is typically based on state machines



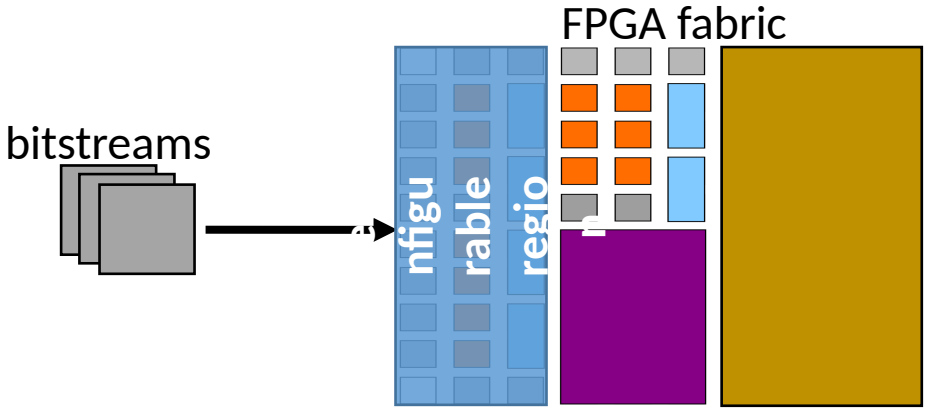
FIR and Sobel filters from Xilinx IP library (screenshot from Vivado 2017.4)

- Possibility to deploy custom **bus logic**
  - Bus/memory contention can be made predictable



Dynamic Partial Reconfiguration

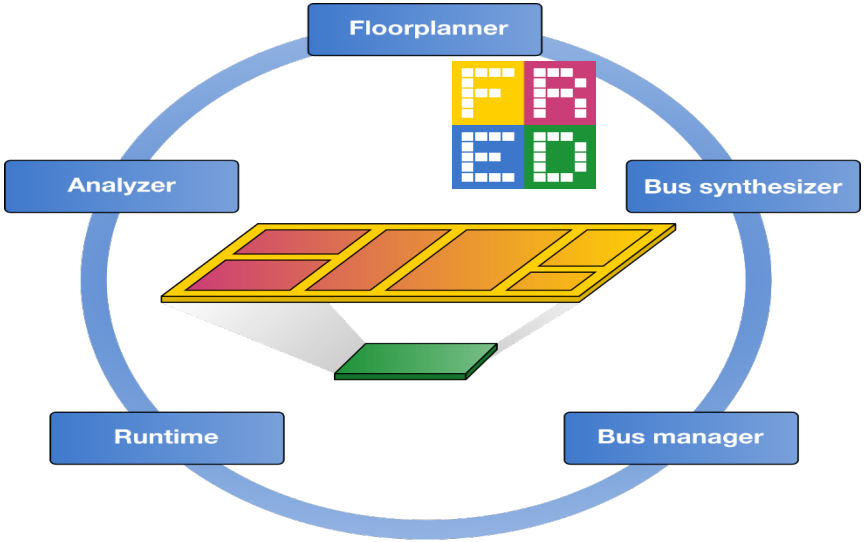
- Modern FPGAs offer dynamic partial reconfiguration (**DPR**) capabilities
- DPR allows **reconfiguring** a portion of the FPGA at **runtime**, while the rest of the device continues to operate
- Is essence, reconfiguration requires programming a memory
  - Simplifying, an image of the FPGA configuration (bitstream) is copied from one memory to another



FRED Framework

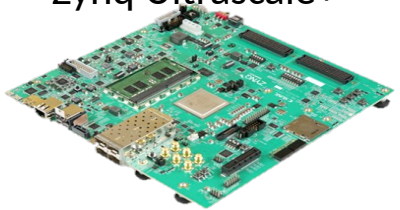
- Enable predictable HW acceleration on FPGA system-on-chips
- Collection of technologies developed at the ReTiS Lab

`http://fred.santannapisa.it/`

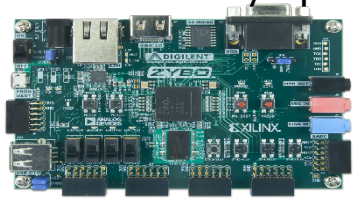


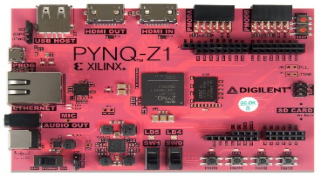
Supported platforms

Zynq Ultrascale+

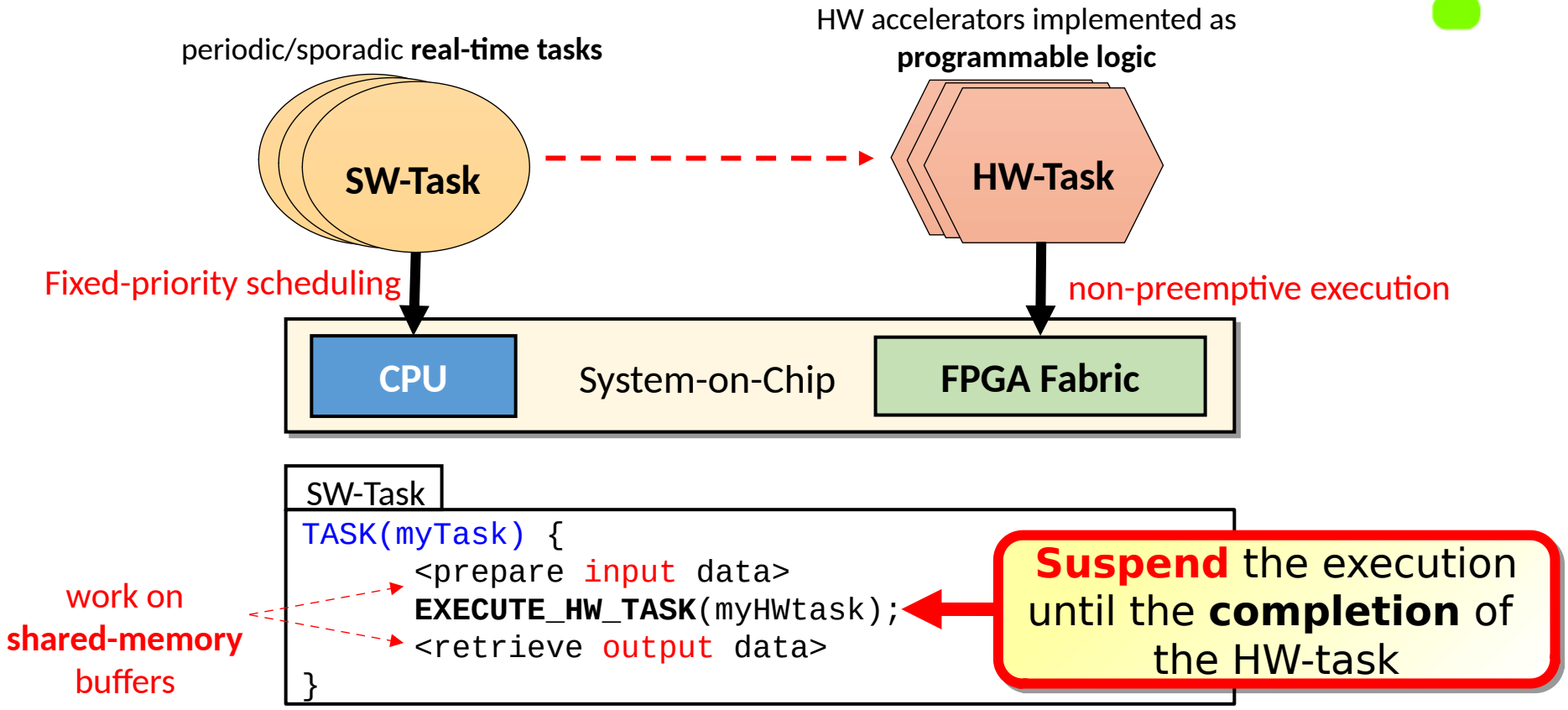


Zynq-7000 series



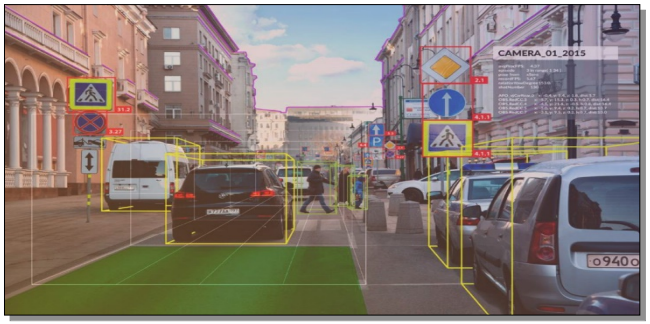


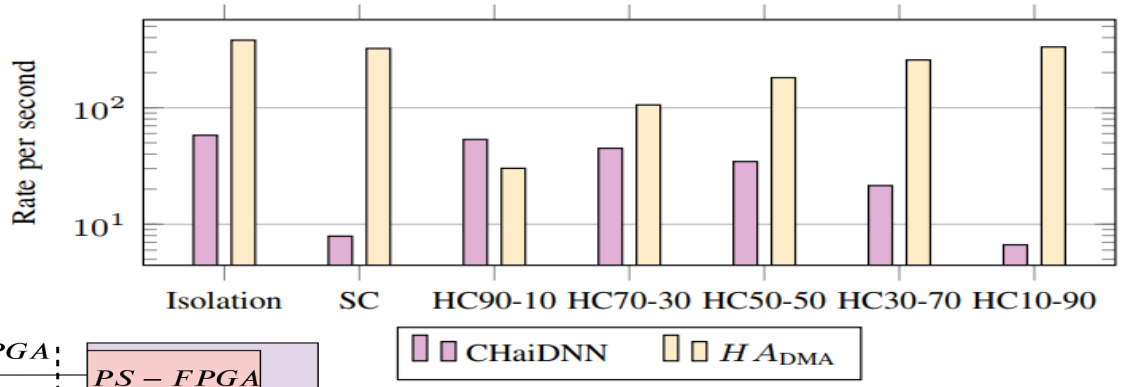
FRED Programming Model

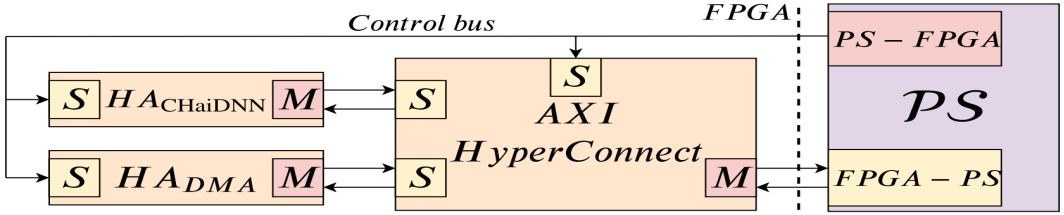


Time-predictable DNN Inference

- **CHaiDNN**: HLS based DNN Accelerator Library for Xilinx Ultrascale+
  - Designed for maximum compute efficiency at 6-bit integer data types (it also supports 8-bit integer data types)
- The inference time in isolation exhibits very little fluctuations
  - The real issue for time predictability is bus/memory contention





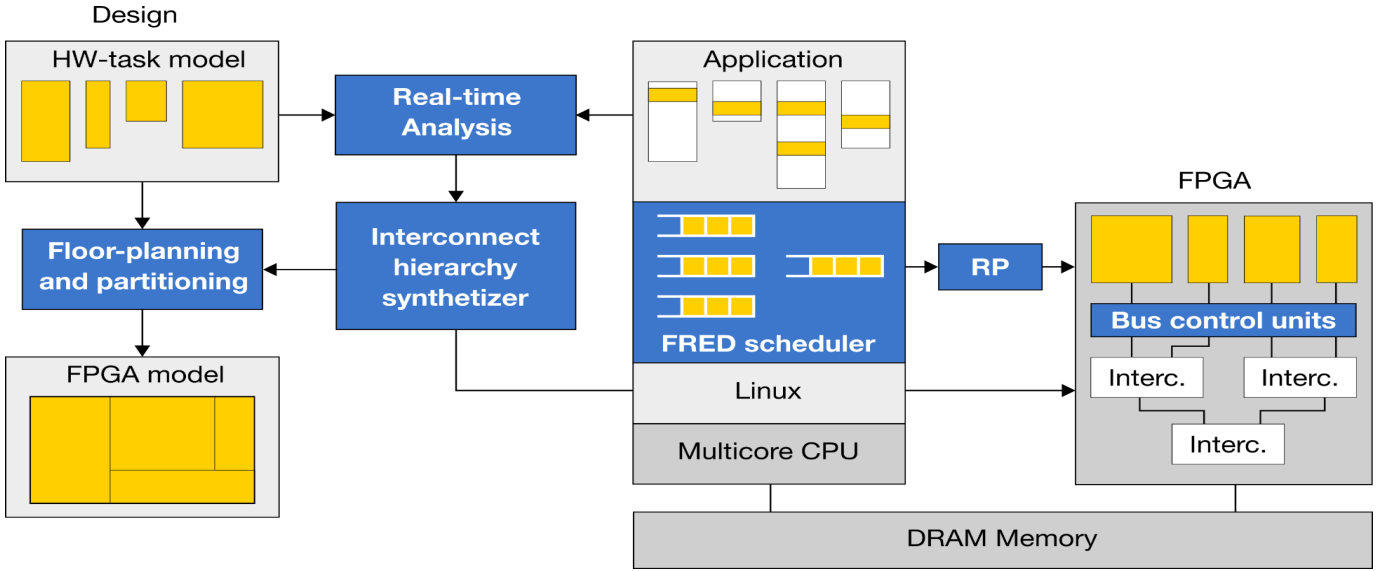


Setup: Xilinx ZCU102 (Ultrascale+), Vivado2018.2, GoogleNet, DMA from Xilinx IP lib

Inside the FRED Framework

The FRED framework is a combination of several technologies:

- Run-time FPGA manager & scheduler for Linux (both C and Python API)
- Bus monitors and budget enforcers
- Automated FPGA floor-planning
- Automatic synthesis of bus interconnections



Conclusions

- AMPERE aims to bridge the gap between MDE and PPM on HHW by
  1. Providing a **development framework** for CPS targeting parallel heterogeneous architectures for an increased productivity compliant with current MDE practises
  2. Providing an **execution framework** for an efficient exploitation of parallel and heterogeneous architectures, fulfilling functional and non-functional constraints
  3. **Integrating** AMPERE software solutions into relevant industrial use-cases (automotive and railway) with HPC and real-time requirements

Thanks for Listening
Any Questions?

https://www.linkedin.com/company/ampere-project

https://twitter.com/ampereproject

www.ampere-euproject.eu

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 871669