Embedded platforms for next-generation autonomous driving systems

PAOLO BURGIO

UNIVERSITY OF MODENA AND REGGIO EMILIA
PAOLO.BURGIO@UNIMORE.IT

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement: 688860

UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

HiPeRT Lab

HERCULES

Future (?) automotive systems

- Cars, in 2016
  - More and more functionalities
  - Massive integration with IoT/environment
  - ..and in 10-15 years?

- Self-driving cars
  - **A**dvanced **D**river **A**ssistance **S**ystems - ADAS
    - "The Holy Graal"

Future (?) automotive systems

- Cars, in 2016
  - More and more functionalities
  - Massive integration with IoT/environment
  - ..and in 10-15 years?

... prototypes exist!

ving cars
ms - ADAS
Holy Graal"

Why aren't we (yet) there?

The four horsemen

1. **Heavy workloads**
   - Sensor-fusion and image-processing
2. **Reduced power consumption**
   - Smaller batteries and renewable power sources
3. **Quickly interact** with the environment
   - Prompt elaboration of sensor data
4. Run **highest criticality** workloads
   - Replacing safety-critical human activities

Driverless systems, today

Expensive: $60k

Bulky: Multiple servers and batteries

Power hungry: up to 5kW!!!

Not marketable!

Climbing the *Power Wall*

- Many-core embedded platforms
  - Hundreds of G(FL)OPs, ~ 10W
  - **C**ommercial-**O**ff-**T**he-**S**helf components
  - COTS

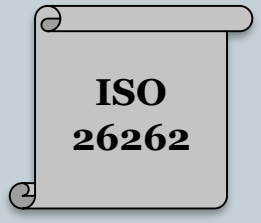
Are they suitable for safety critical/ADAS?

- **Not yet!**

- Too complex architectures
  - *H*s-to-*K*s cores/threads => **Poor programmability**
  - Shared resources => Not analyzable => **Poor predictability**

Hard Real-Time & safety critical systems

- The keyword: **predictability**
  - Provide the correct result….at the correct instant
  - The system must be **simple to analyze**
  - Trade average for **worst case performance**



- Automotive safety standards
  - ASIL-B
  - ASIL-D

- "Real-Time embedded super-computing platform"
  - Predictable + (high-)performance + low power
  - Holy Graal (…for engineers)

Hercules in a nutshell

- High-Performance Real-time Architectures for Low-Power Embedded Systems
  - EU's H2020 ICT-4-2015
  - http://hercules2020.eu/

*"Obtain **an order-of-magnitude** improvement in the cost and power consumption of next generation real-time applications for safety-critical domains."*

...driven by industry...

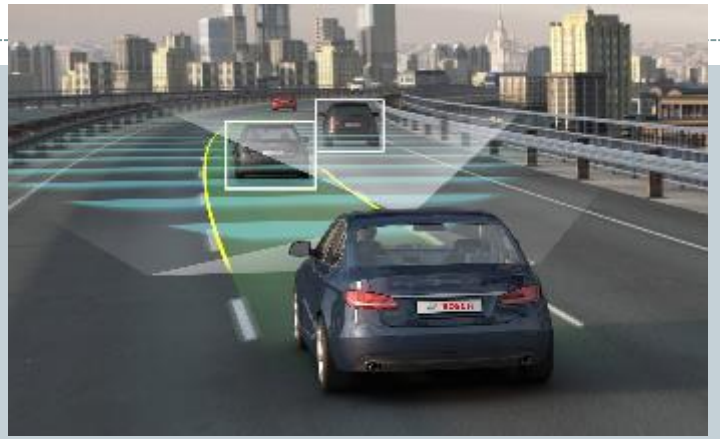
- Project partners
  - Airbus AGI, Magneti Marelli S.p.A., Pitom snc
- **I**ndustrial **A**dvisory **B**oard
  - Provide feedback/advices
  - Porsche, Finmeccanica, BMW, NVIDIA, Continental, Autoliv...

Predictable platform requirements

- "Traditional" embedded designs
  - Not built for predictability
- Custom solutions?
  - Quickly become obsolete
- Use Commercial-Off-The-Shelf SoCs
  - Cheap
  - Robust to hardware and timing faults
  - Fully supported by the hardware provider

- Hercules ➔ Pure software approach
  - Techniques for predictability on top of COTS
  - Scales different platforms/generations
  - Expressive programming model

The Tegra SoC Family



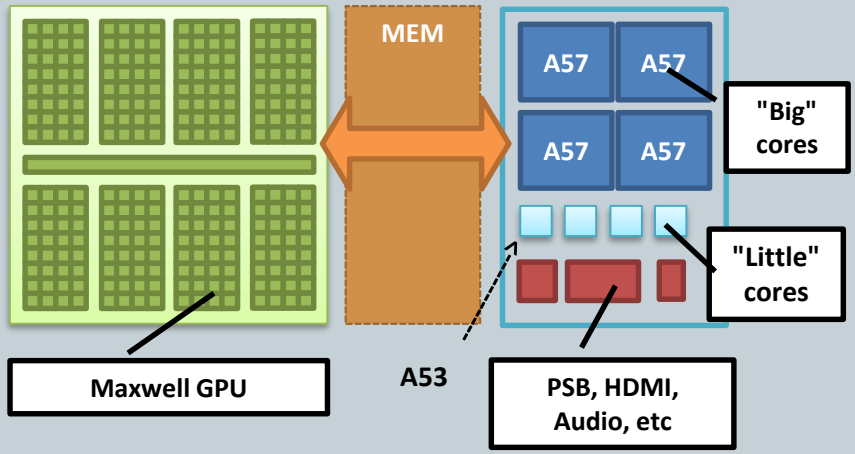
- Explicitly targets automotive market
- Tegra X1: Big.LITTLE + Maxwell GPU
  - 8 + 256 cores
  - 1TFLOP, 15 W
  - **Not** certified ISO 26262

DISCLAIMER

- Drive PX 2
  - Big.LITTLE + Maxwell GPU
  - **Certified** ASIL-B ISO 26262
  - Q1, 2017

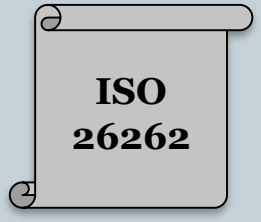
Big.BIGGER

- Cooperation UNIMORE/NVIDIA



Hercules ADAS platform

1. Safety microcontroller subsystem
   - For **legacy** highly critical computation
   - ASIL-D compliant



2. General Purpose multi-core "host"
   - Big.LITTLE – like
   - **Control-driven** "less critical" computation

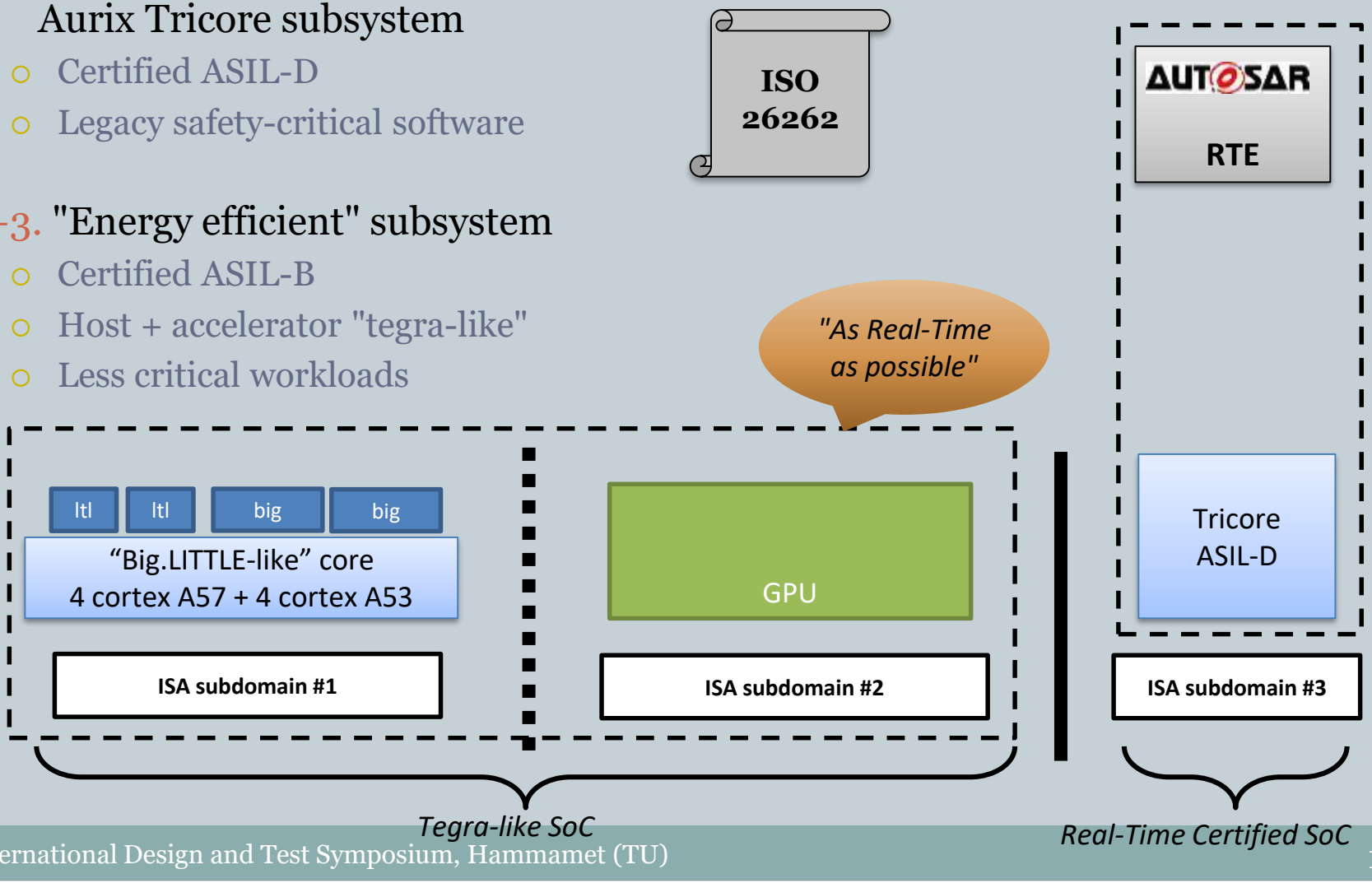
3. Many-core accelerator
   - GPU for **massively parallel, data-driven computation**
   - Energy-efficient subsystem

Tegra-like

The Hercules ADAS platform

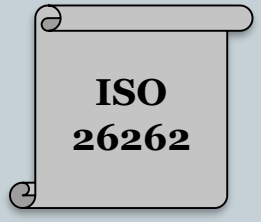
1. Aurix Tricore subsystem
   - Certified ASIL-D
   - Legacy safety-critical software

2+3. "Energy efficient" subsystem
   - Certified ASIL-B
   - Host + accelerator "tegra-like"
   - Less critical workloads



Hercules programming models

- Tackle programmability wall
  - Hide platform complexity
  - Layered approach
  - Portability across platforms/families



AUTOSAR

1. AUTOSAR
   - Legacy safety code
   - Certified at highest safety standard ASIL-D

2. OpenMP
   - *De-facto* standard for SMP, and HW accelerators (OMP 4.x)

OpenMP

3. CUDA/OpenCL
   - CUDA *de-facto* for NVIDIA GPUs
   - OpenCL *"de-iure"* standard for accelerators

NVIDIA CUDA

OpenCL

Hercules programming models

- Tackle programmability wall
  -
  -
  - s/families

ISO 26262

AUTOSAR

1.
   -
   - tandard ASIL-D
2.
   - and HW accelerators (OMP 4.x)

OpenMP

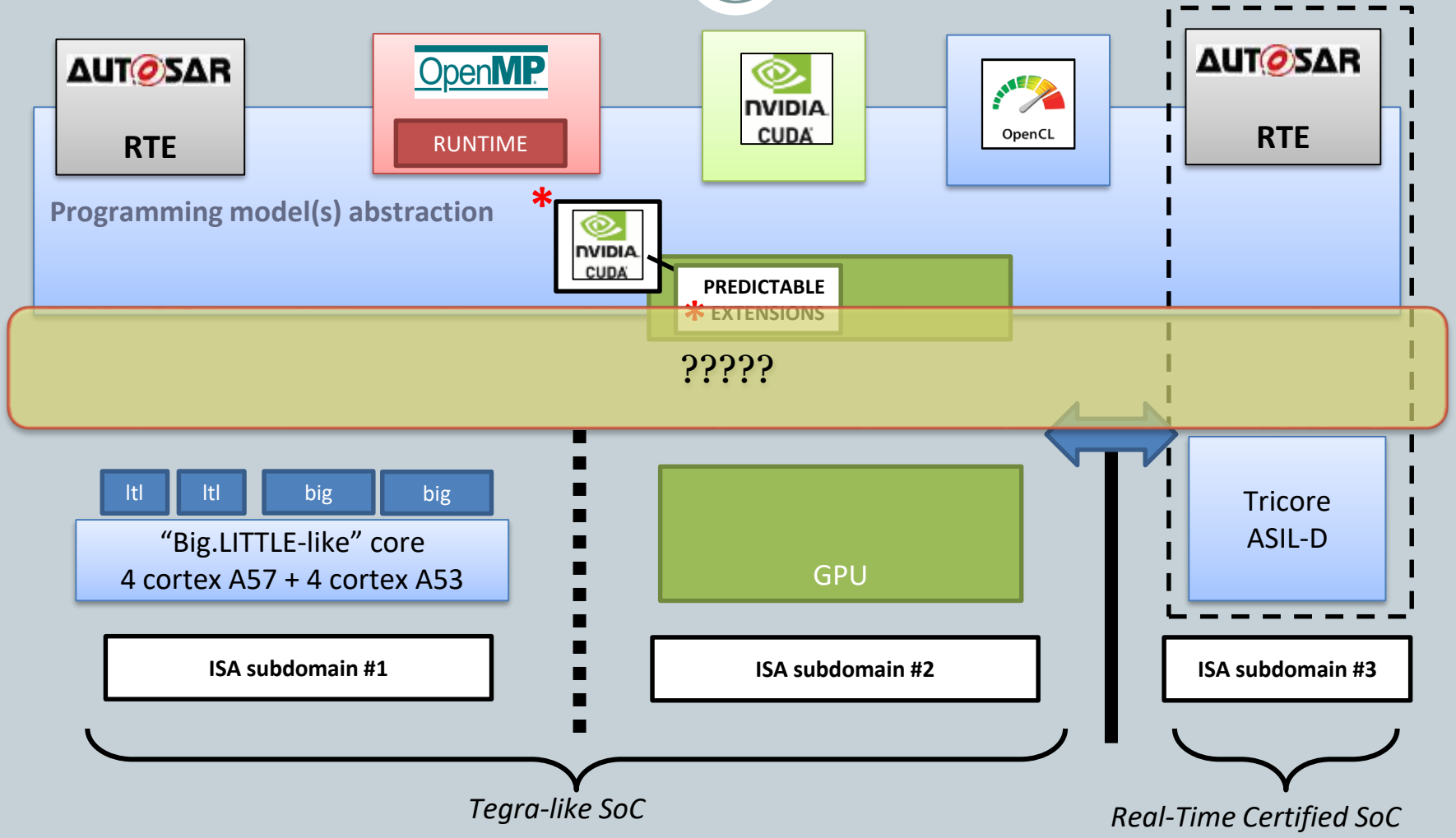
3. CUDA/OpenCL
   - CUDA *de-facto* for NVIDIA GPUs
   - OpenCL *"de-iure"* standard for accelerators

No predictability guarantees!

NVIDIA CUDA

OpenCL

The Hercules stack

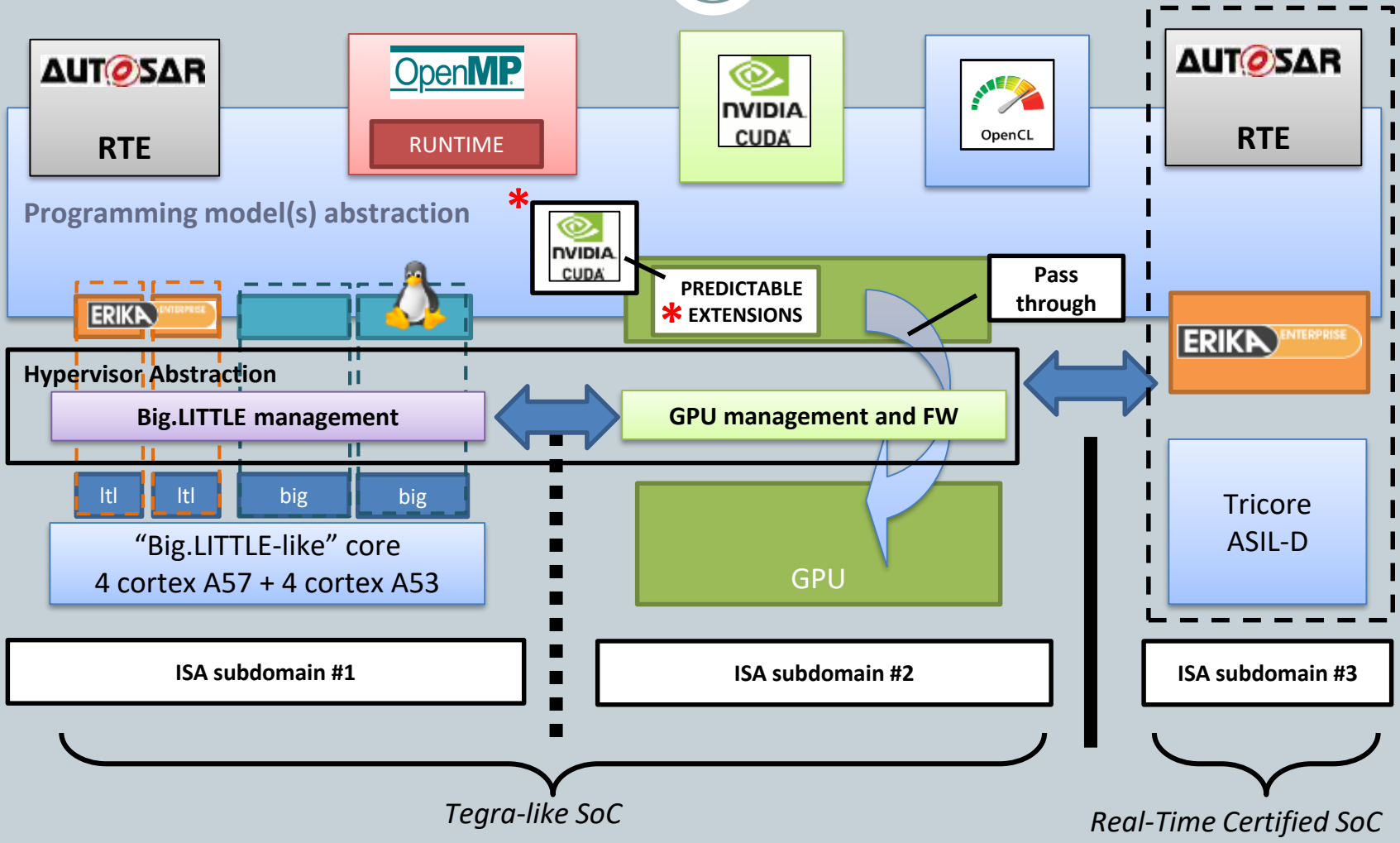


An integrated system

- An application will be composed by
  - A **static**, safety critical part, running on AUTOSAR RTE
  - A **dynamic** part, running on Linux and the GPUs
    - *"As Real-Time as possible"*
  - Integrated in the same framework

- Idea: use an Hypervisor
  - To decouple HW and OS/Apps
  - Simple, clean design
    - ➔ Analyzability
  - Centralized manager for shared resources
    - Memory, caches, I/O… and the GPU!

The Hercules stack



Linux on the Big cores

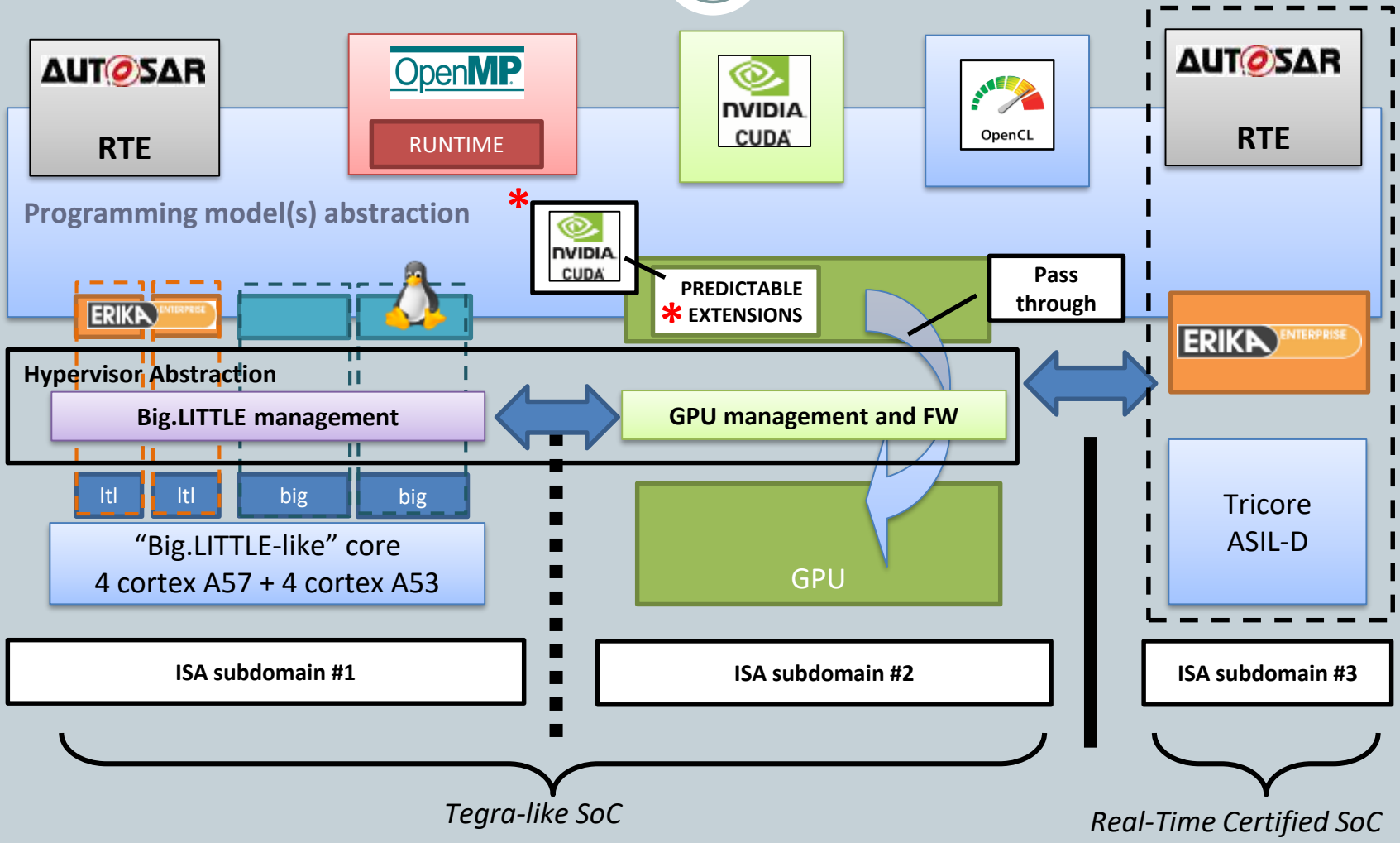
- Natively supports targeted programming models
- Runs on ARM A57
- SCHED_DEADLINE
  - Developed by Evidence, mainline since Kernel 3.10 (TBC :P)
  - Earliest Deadline First -- EDF scheduler + resource reservation
  - Will be enhanced with power-aware scheduling algorithms

ERIKA Enterprise on the LITTLE cores

http://erika.tuxfamily.org

- RTOS OSEK/VDX certified
- Support to AUTOSAR API under dev.
- Open-source license allowing static linking of closed source code (GPL + Linking Exception)
- Typical footprint around 2-4KB Flash
- Used in automotive applications and research projects
- Support for AURIX tricore

The Hercules stack



Benefits of virtualization

- Isolation between the subsystems
  - Freedom from interference

- Implement predictability guarantees
  - System more analyzable
  - Hypervisor manages shared resources
    - (such as the GPU)

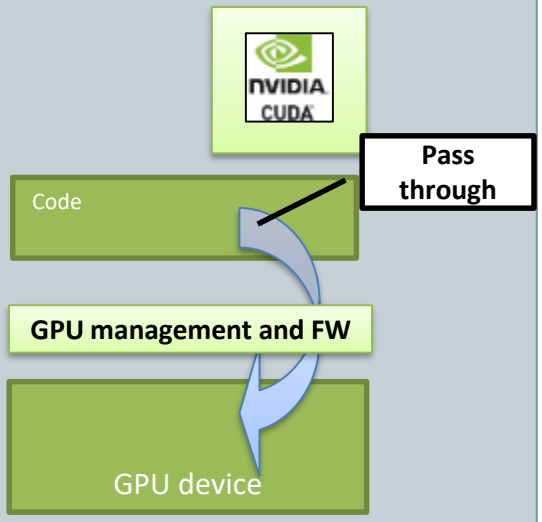
- Supports multiple operating systems
  - RT-Linux on Big cores
  - Erika Enterprise on LITTLE cores
  - AUTOSAR RTE for high safety requirements

ERIKA ENTERPRISE

AUTOSAR

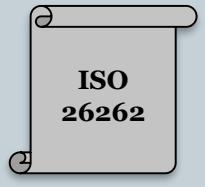
Requirements for an hypervisor

- Host "Big.LITTLE" side:
  - JailHouse / RT-Xen / NOVA / Xvisor ...and others
- Accelerator/GPU side:
  - Pass-through minimizes the overhead of GPU virtualization
  - NVIDIA's native Vibrante (200% closed source)



Key points

- Few KLOCs of code
  - Ease ISO 26262 certification
- Open source license whenever possible
- Customizable with innovative scheduling techniques
  - PRedictable Execution Models

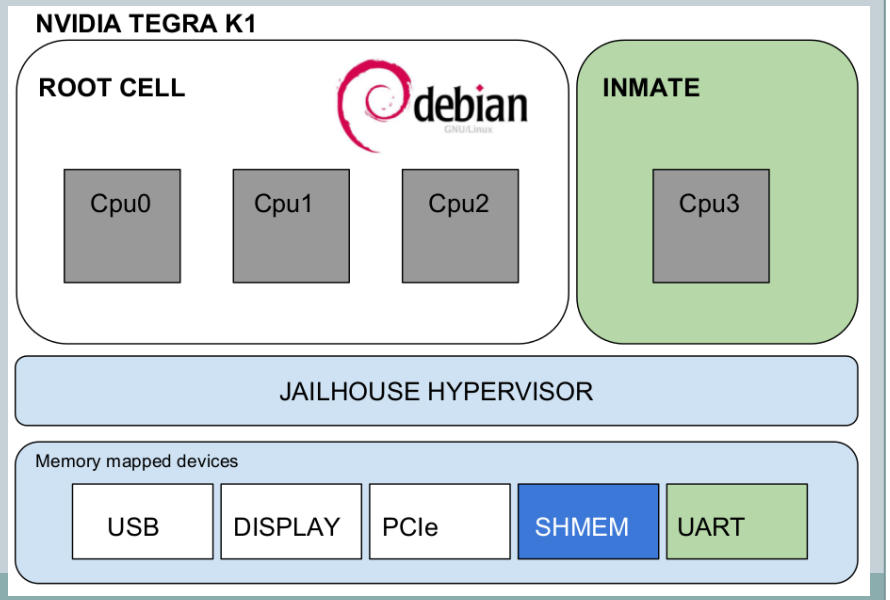


Jailhouse on NVIDIA Tegra X1

- Support for NVIDIA's Linux Kernel 3.10
  - Code is Open Source (GitHub)
- Hypercall from a bare-metal application running on a dedicated core
  - ~twice of the time of a Linux sycall
- Ongoing work
  - Port ERIKA Enterprise on LITTLE cores
  - Already runs on Aurix, Linux already on Big cores
  - Then, focus on the GPU (Vibrante?)

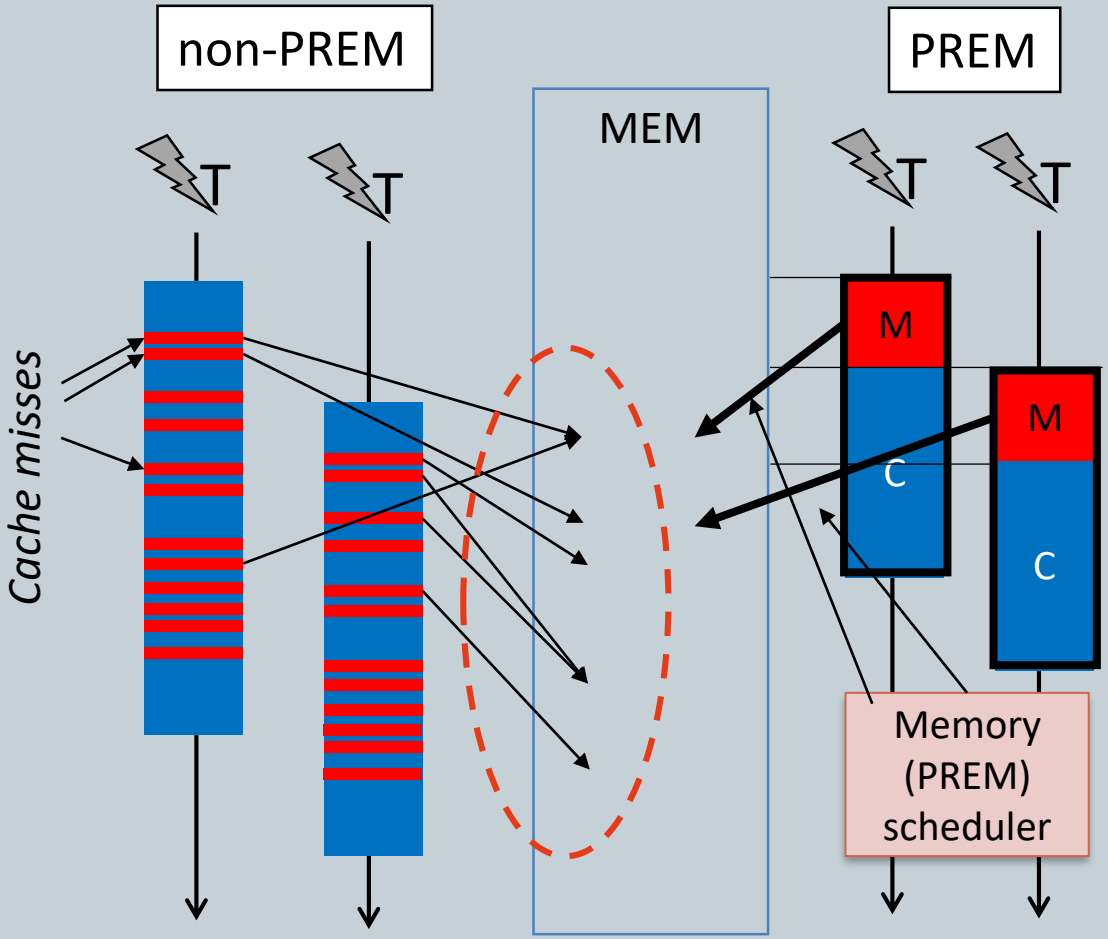
Hercules design for resource management

- Idea: assign devices exclusively to hypervisor cells
  - Ex: USB, Display, PCIe to Root cell, UART to Inmate cell
  - Act as servers, to whom peripherals are assigned
- Novell communication mechanism between cells
  - Featuring shared memory
  - Mailbox-based
  - To enable resource sharing
- Implemented in Tegra K1
  - Older generation than TX1

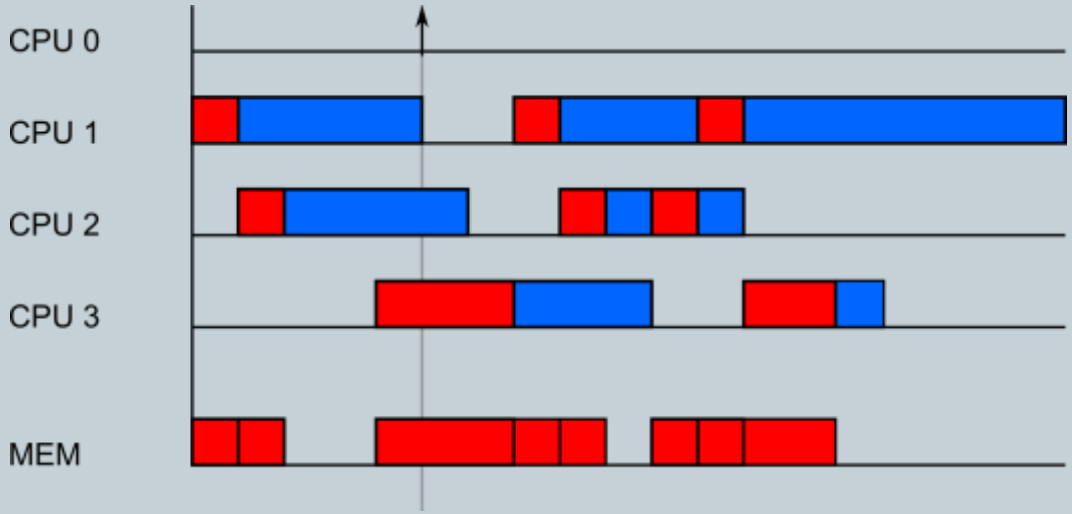


PRedictable Execution Model (PREM)

- Instead of uncontrolled cache misses…
- …split each task onto
  - **M**emory phase
  - **C**omputation phase
- "Predictable intervals"
  - First, memory prefetching
  - No cache misses in the computation phase
  - Non-preemptive computation phase



PRedictable Execution Model (PREM)



- Memory is the scarce resource in many-cores
  - Schedule M phases removes contention
  - Apply also to I/O, GPU..
  - A simple CPU-centric scheduling might not be effective

Preliminary work on PREM

P. Burgio A. Marongiu, P. Valente and M. Bertogna, *A memory-centric approach to enable timing-predictability within embedded many-core accelerators,* RTEST 2015

- Representative embedded multi-core accelerator
  - Texas instrument Keystone II
- Impressive worst case performance gain
  - PREM effective on modern parallel architectures

# Cores/threads	1	2	4	8
No-PREM – Worst (Analytical)	0.026	0.047	0.088	0.170
PREM – Worst (Analytical)	0.010	0.014	0.022	0.038
Speedup	2.6×	3.4×	4.0×	4.5×

PREM in host subsystem

- Code must be PREM-compliant to enable memory-aware task coscheduling
  - Might not always be possible!
  - Bugs might cause application to misbehave

- Memory throttling mechanism
  - Inspired by MemGuard
  - Assign a budget to each VM running on the Hypervisor
  - Use BUS_ACCESS event to count memory R/Ws

H. Yun, G. Yao, R. Pellizzoni, M. Caccamo, L. Sha, MemGuard: Memory bandwidth reservation system for efficient performance isolation in multicore platforms, in: RTAS 2013

Hercules in a nutshell

- Partners

1 (Coordinator)	University of Modena	UNIMORE	Italy
2	Czech Technical University in Prague	CTU	Czech Republic
3	ETH Zurich	ETHZ	Switzerland
4	Evidence Srl	EVI	Italy
5	Pitom snc	PIT	Italy
6	Airbus Gmbh	AB	Germany
7	Magneti Marelli	MM	Italy

- January 2016 – December 2018
- Budget: ~3.3 M

http://hercules2020.eu/

Thank you!

paolo.burgio@unimore.it

UNIMORE
UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA

High-Performance Real-Time Lab