Technische Universität Braunschweig

Models for Mixed Criticality

Mladen Berekovic

Computer Engineering

TU Braunschweig

EMC2 Workshop @ Hipeac 2016 in Prague

- January 2016
- Targeting all technologies from EMC2
- Will Circulate Call for Papers via email
- https://www.hipeac.net/2016/prague/

Feel Encouraged to submit !

Design Methodologies for Mixed Criticality on MPCs

- Design Tools & Methodologies have to reflect increased complexity
- Introduction of formal methods
  - Worst-Case Execution Time Analysis (WCET)
- Early System Validation of Hardware & Software Components / Integration
- Faster Simulation Methods

Virtual System Prototyping

What is a virtual prototype ?

Demo System on a *Virtual System Platform*

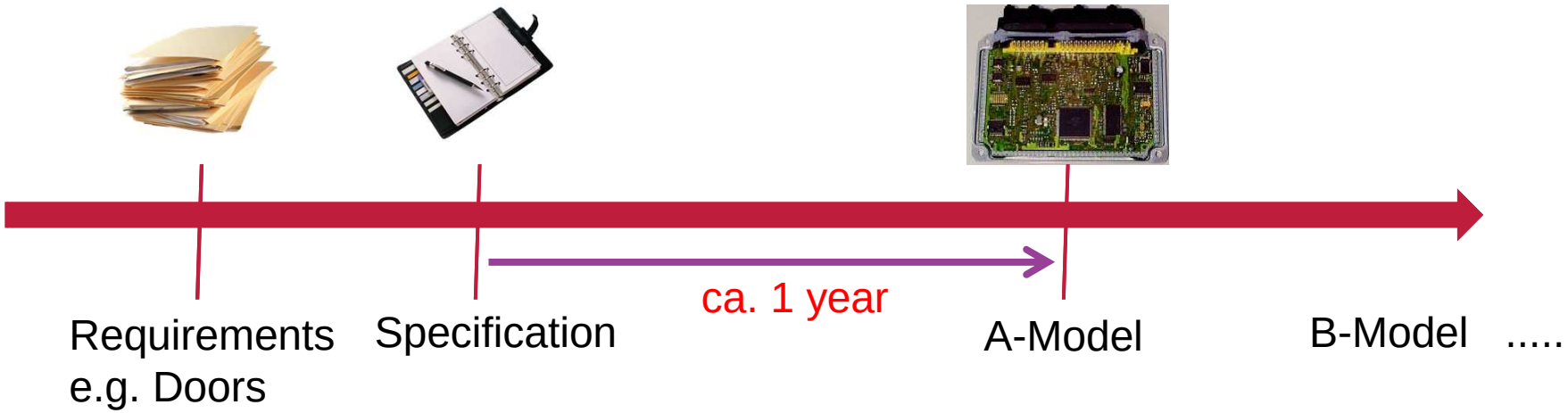
Virtual System Platform (VP)

- Is a fast functional (SW-) Model of an embedded system (e.g. ECU – real or future)
- Consists of a (fast) Instruction-Set Simulator (ISS) and a set of models of chip- and board components (Bus, Memory, etc.)
- Executes the same ***Binaries*** as the (ECU) Hardware
- Runs on Standard Compute Platforms
- Uses **Electronic-System-Level (ESL)** Design-Techniques based on **SystemC** and **TLM** for a fast Hardware-Software Simulation

A Virtual System Platform looks to the OS / Applications like **real** hardware

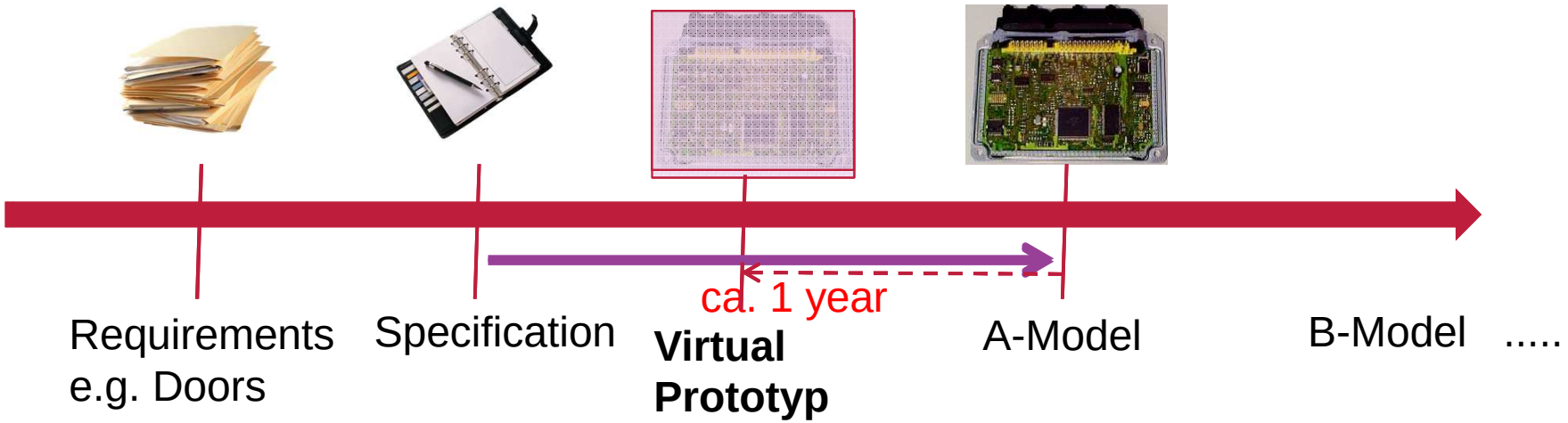
Virtual System prototyping in the Automotive Development Process – OEM as architect and integrator

- Separation of technical and functional integration
  - Tier1: Secures Execution of SW (techn. Integration)
  - OEM: Tests Functionality of a subsystem (funct. Integration)



Virtual System prototyping in the Automotive Development Process – OEM as architect and integrator

- Separation of technical and functional integration
  - Tier1: Secures Execution of SW (techn. Integration)
  - OEM: Tests Functionality of a subsystem (funct. Integration)
- ***Virtual Prototyping***: Early Availability of an „executable spec“ within the supply chain

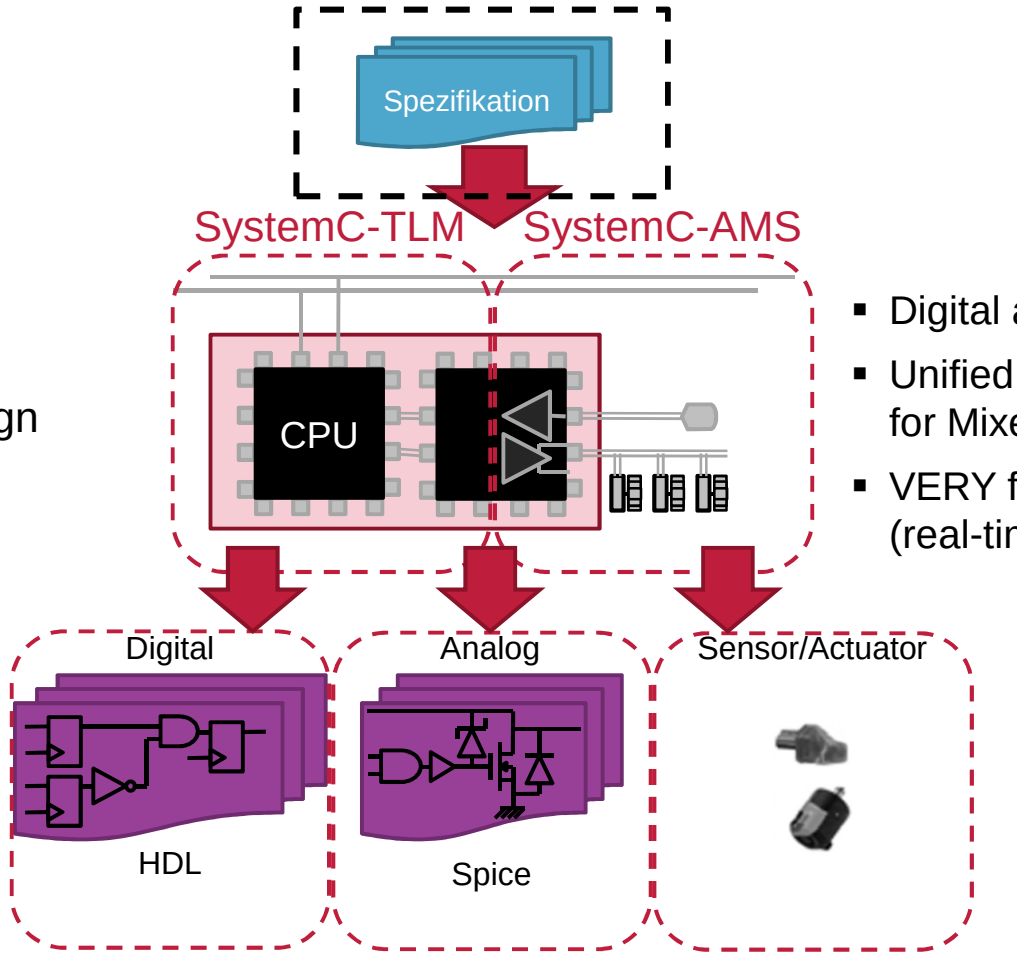


Electronic-System-Level (ESL) Design: Virtual System-Platform with SystemC/TLM

System-Design
Functional Design

Architecture-Design

Implemention



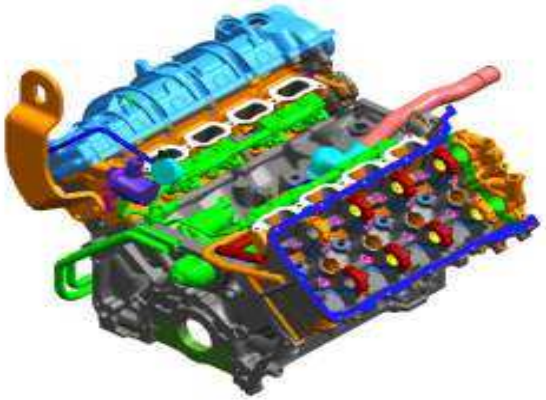
- Digital and Analog Simulation,
- Unified System-Environment for Mixed Signal
- VERY fast Digital-Simulators (real-time feasible).

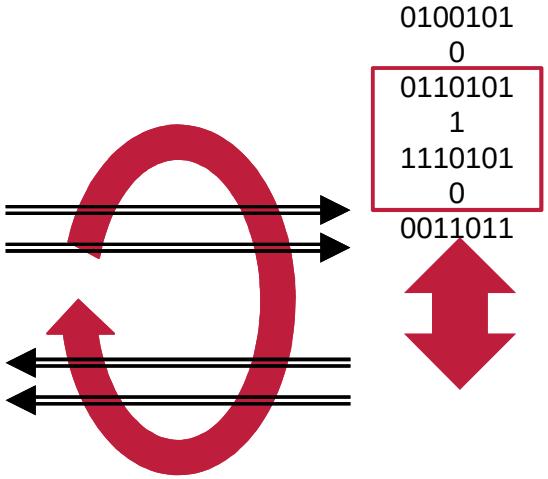
Source: Denso

Concrete: ECU as Virtual System-Platform

Joint Simulation

Engine-Control Model





Microcontroller-Model

Control- & Basic-Software
(Implementierungsverbindung)

New Development,
New Application
(Combination)

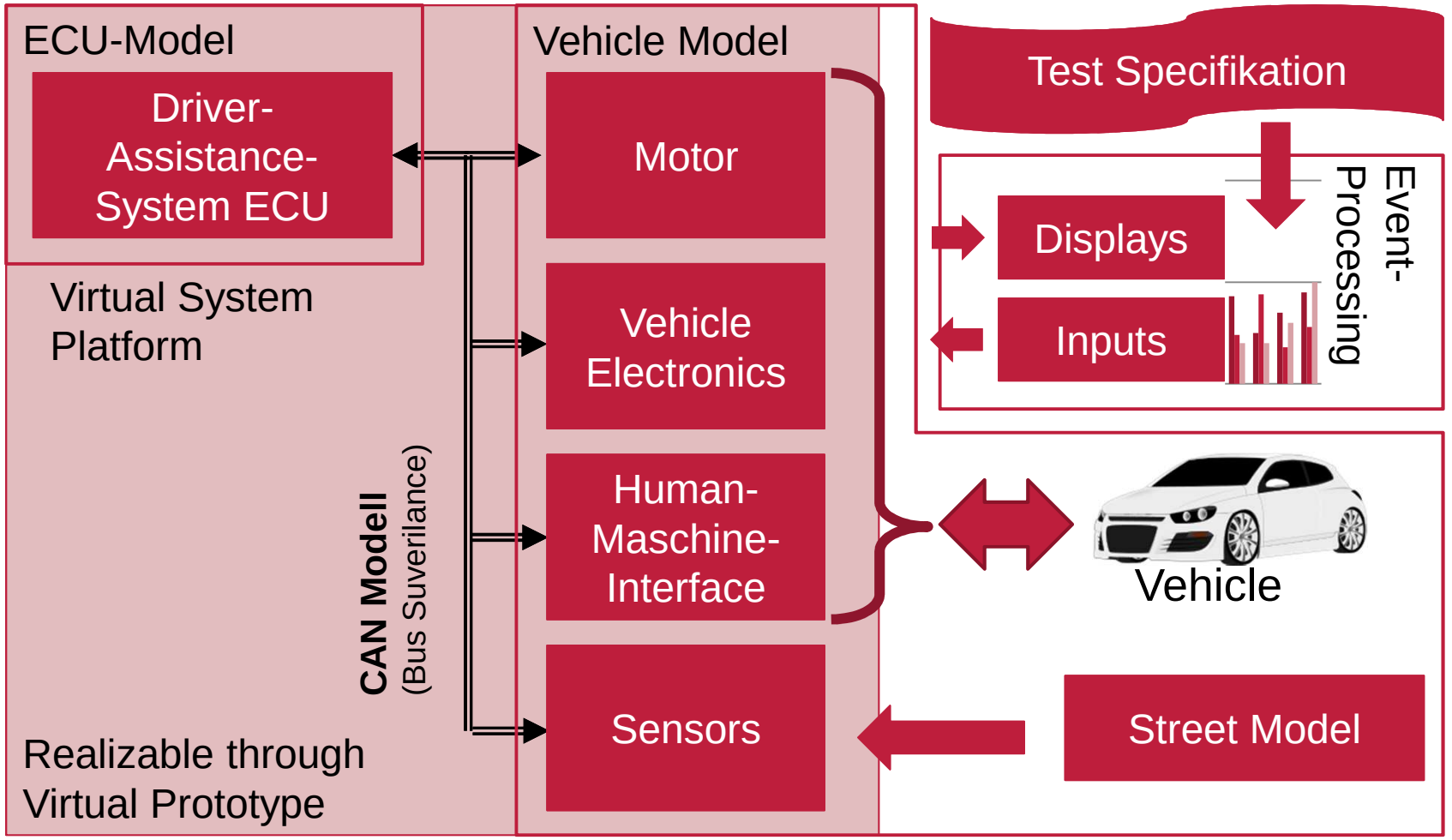
Controler- and Peripheral-HW

Neuanwendung
(Kombination)

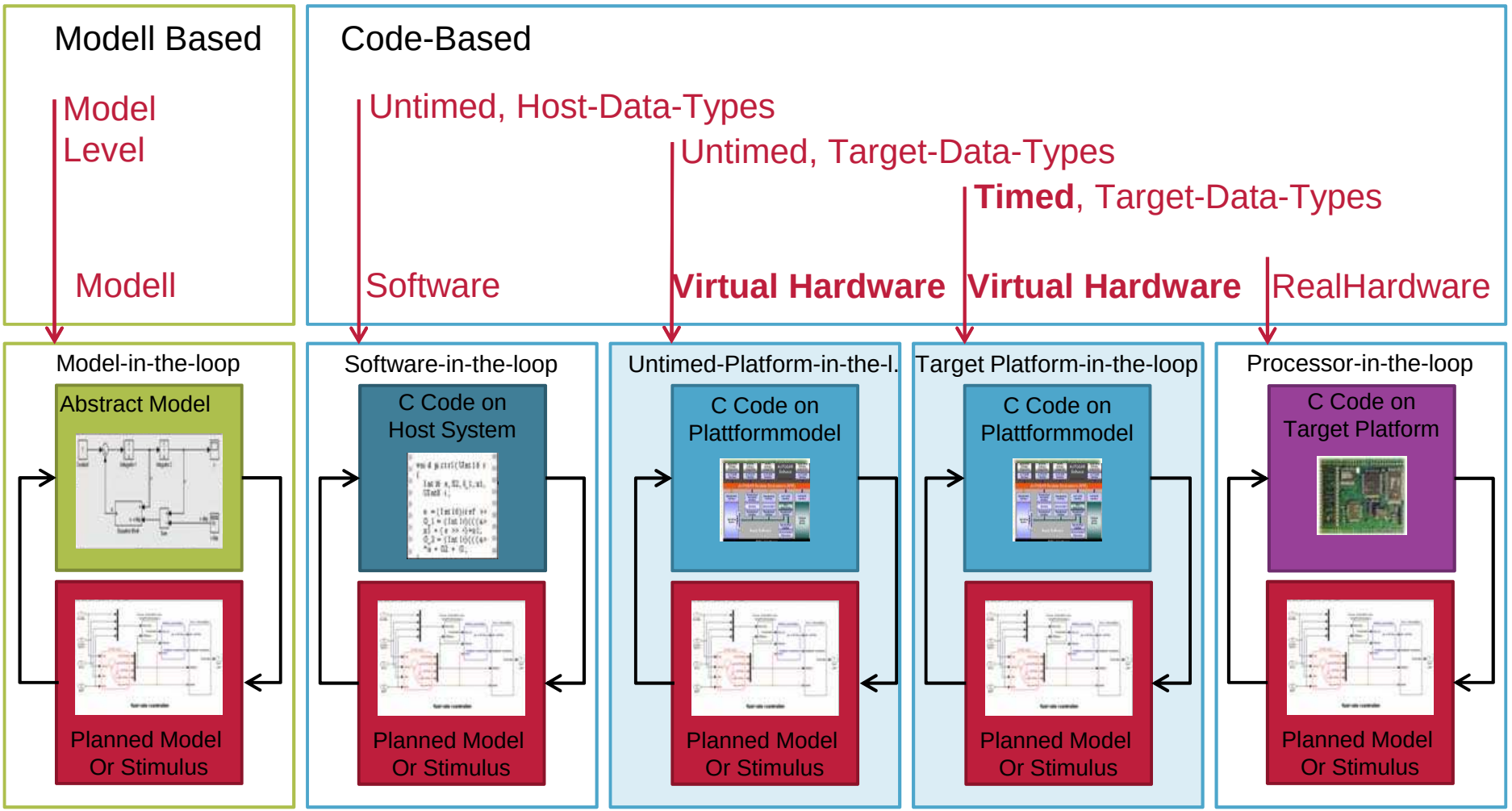
Source: Hitachi

- Complete Software-Simulation of the Hardware/Software-System,
- Hardware-in-the-Loop (**HILS**) ECU replaced with Software-based "Virtual Hardware-in-the-Loop" (**virtual HILS**).

Virtual Hardware-in-the-Loop (vHILS)



ES Software Development with Virtual Prototypes



Source: Hitachi

Software-Design on Virtual System-Platforms (e.G. with Cadence VSP Tool)

Early Software- and System-Development

- High-Performance-Simulation, runs with Production-Software,
- Higher SW-Design-Productivity with Multi-core-HW/SW-Debugging.

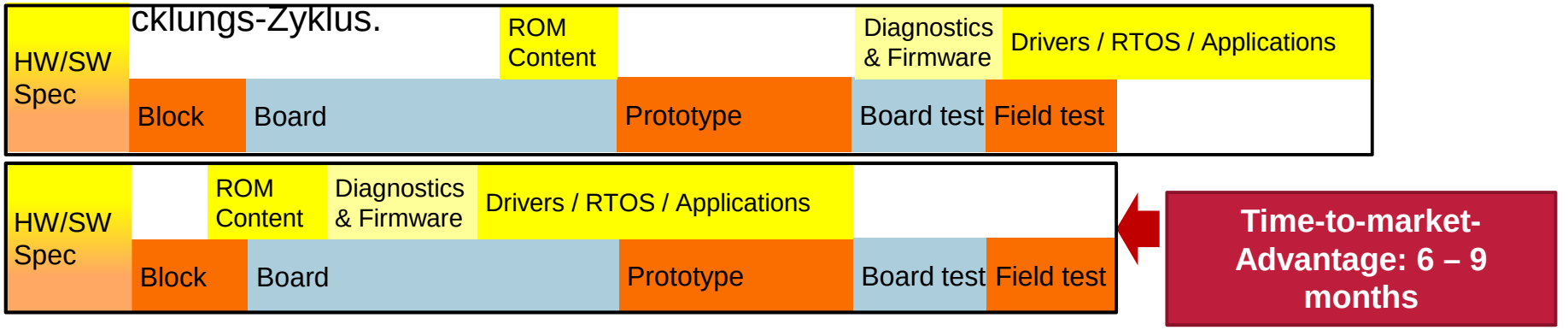
Fast Platform Design, Debugging und Analysis

- HW/SW-Debugging,
- Plattform- and Functional-Profiling, to localise Bottlenecks.

Validation of von Hardware/Software-Integration 1 year <u>before</u> Availability of Hardware

- Unified HW/SW-Simulator for System-Validiation (e.G. with Cadence VSP),
- Enables finer Synchronisation between HW- & SW-Design in the Development cycle

cklungs-Zyklus.



Source: Cadence

Tool-Support for Embedded-Software-Design (ESD) on virtual Platforms

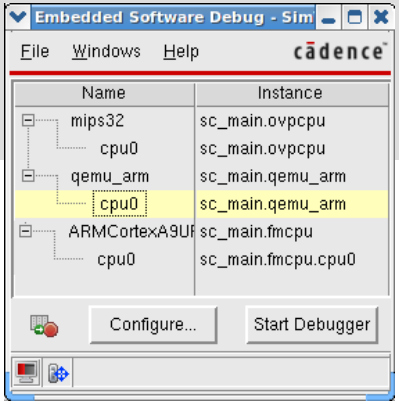
„Programmer's View" of Hardware models only those aspects of hardware that are needed for software debugging. Alle other Hardware Aspects are abstracted from theSW-developer.

- Programmer's View of Hardware
  - Processor- und Peripheral-Registers „viewing" and trace,
  - Memory View,
  - Interrupt/Reset Trace,
  - "Waveform" type view von interrupts, resets, HW-Register-accesses,
  - Peripherals accessible through a processor.
- Processor/Core Sensitive GUI Views
- "One-click" Launch & Debugging of several ESD-Executables on a single VP
- Full ESD-Debug of Multi-core und Multi-Prozessor
- Non-intrusive ESD Trace, Analysis & Profiling finds more problems than ESD- Debug
  - Line, Function coverage, Memory Allocation Trace … .

Cadence VSP Eclipse

Source: Cadence

Tools for the ESD Design on virtual Platforms



*Integrated by Cadence in VSP

Source: Cadence

Performance of Virtual Platform Simulation
Camera Demonstrator

Mixed FPGA/VP-Simulation:

- Simulation speed 13MHz
- High Accuracy at full testability

VP-Demonstration:

- Street sign detecion
- Almost in real-time

Embedded World 2013 (Cadence)

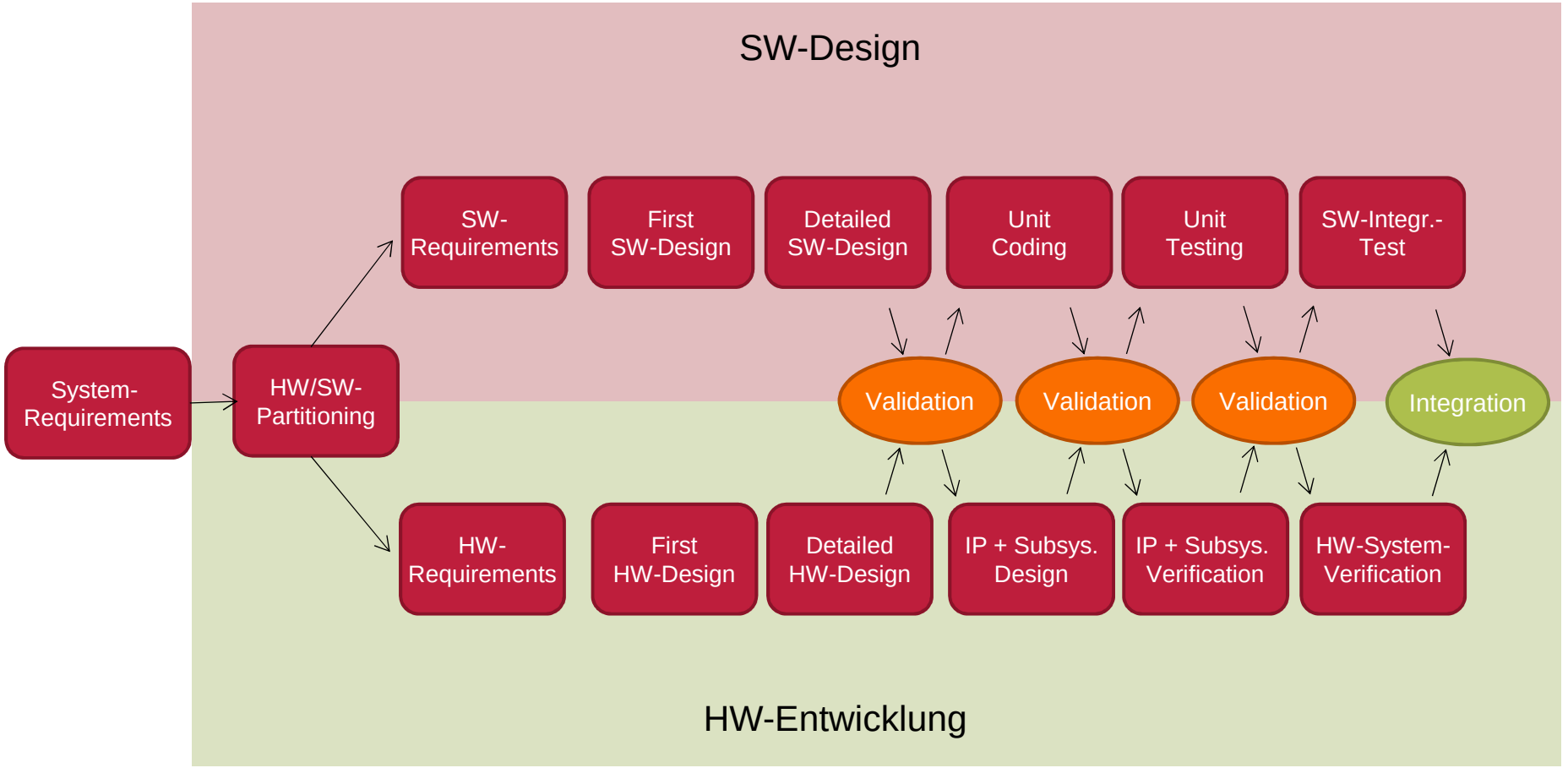
- http://goo.gl/T2BjZ (Youtube)

Realization of Virtual System-Platforms with Electronic-System-Level Design Methodology (ESL)

- **ESL** is a standard that is based on **SystemC/TLM** (IEEE 1666-2005) (C/C++ based) for fast Hardware-Simulation.
- HW und SW are developed in a **single** language (C++) instead of C/C++/Java and Verilog.
- **ESL** is basde on the simplified modeling of HW/SW-Systems from a communication perspective (**TLM**: Transaction-Level Modeling).
- A System-Modeling on Transaction Level (**TLM**), delivers a speedup of the HW/SW-System Simulator by a factor of of 1000-10000. This enables **virtual prototyping** on a workstation without real hardware („Executable Spec“).
- A System-Modeling on Transaction-Level (**TLM**), is possible on 3 levels of detail that can also be mixed (mixed-mode simullation) for verification and validation purposes
  - LT (loosely timed): Abstract model for fast software verification,
  - AT (approximately timed): 90% Cycle-true, 90% speed
  - CT (cycle-true): Fully detailed hardware model
- The high Abstraction level together with the high simulation speed enables prototyping and design space exploration of complex hardware/software systems such as ECUs.

HW/SW-Integration in the system design

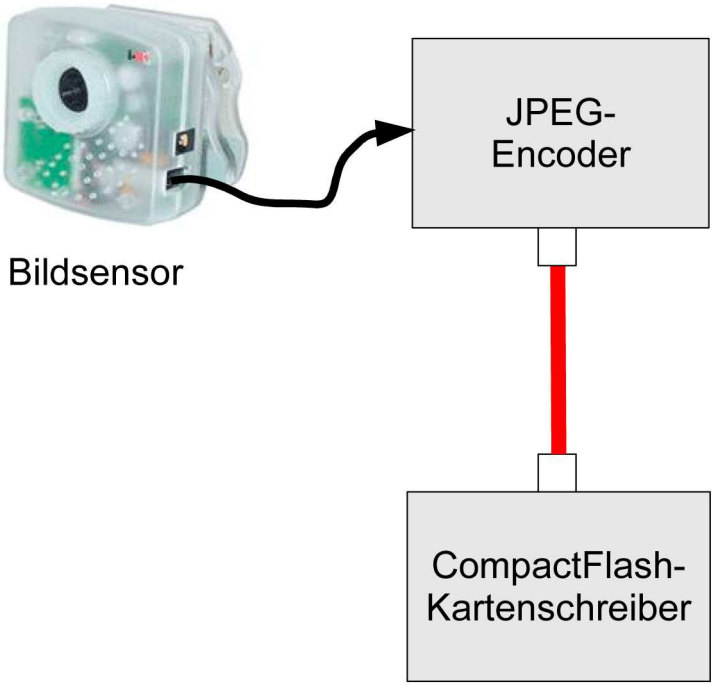
Validation of the HW/SW-Integration as part of the development cycle



Quelle: Cadence

HW/SW-Design with Transaction-Level Modellierung (TLM)

- **Separation** of Function und Communication (Orthogonality)
- Function und Communication each adequately abstract
- Refine Component, Communikation as coarse transaction of abstract Data packages
- Abstract Component, but precise Bus-Protocol (e.g. Bus-data traffic estimation)
- **Virtual System-Platformen** (VP), Simulation of Function and Communication
- Virtual Prozessors are **binary- and register compatible**
- Hybrid (TLM/RTL) Model für cycle-true simulation possible (SystemC and Verilog joint simulations possible; hybrid mixed-mode-Simulation)



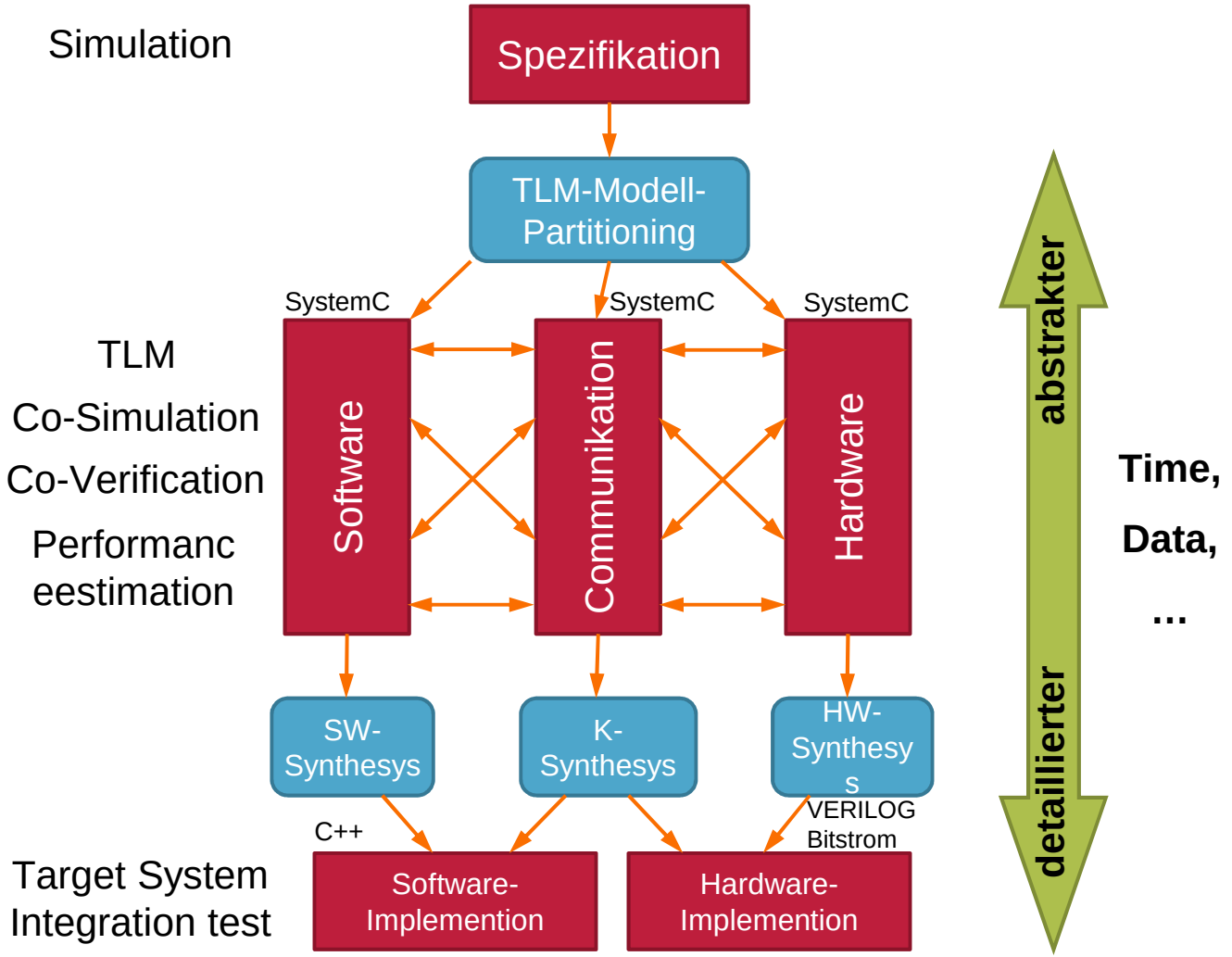
System-Realization with SystemC and TLM

Speci-
fication

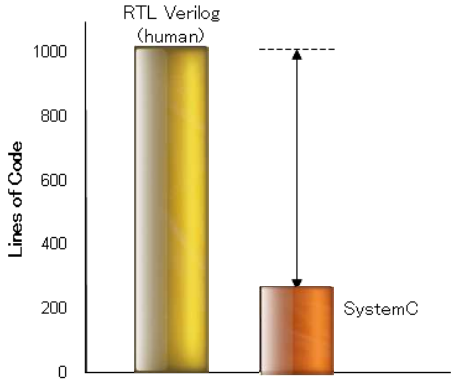
SW-
Modeling

Target-Code-
Generation

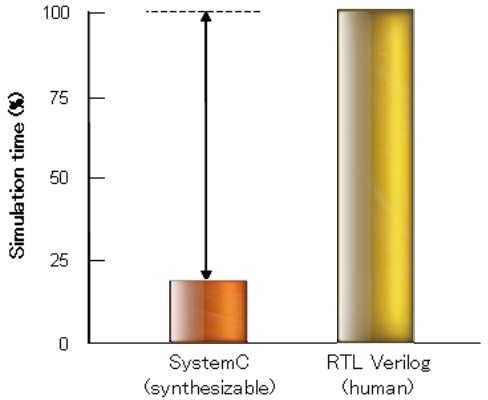
Virtual-
Platform-
Generation



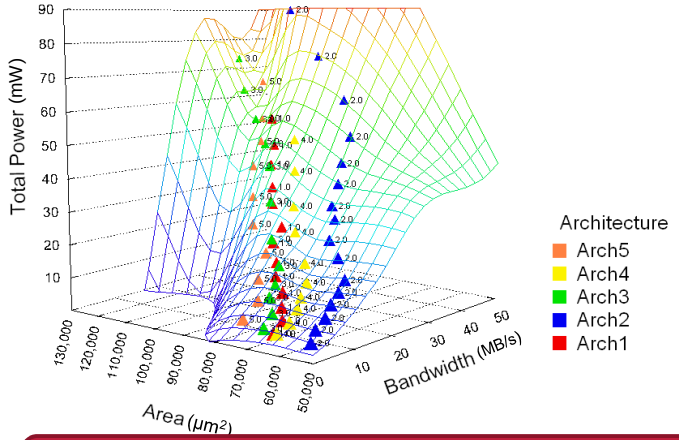
TLM-based Module-Design accelerates Time-To-Market



~3 x kompact Design, efficiency increase

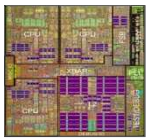


5–10 x faster Verification



Faster Design-Space-Exploration:
"Do we reach this with 200 MHz?"

SystemC → RTL in **10 days.**
vs. manual RTL in **3 month**



I/F-Controller-Modul

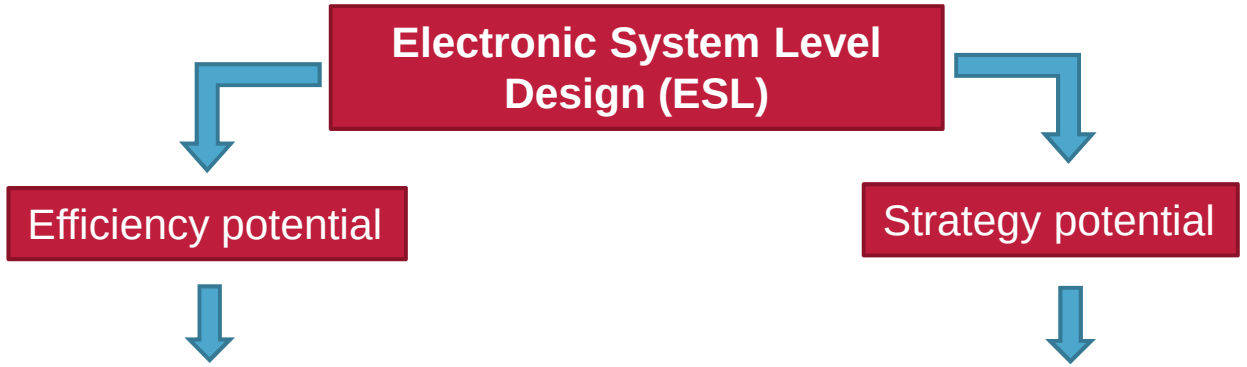
FPGA –reconfig in **1-2 days.**
vs. manual in **2-3 weeks**

Steuergerät

>10 x Productivity-invcrease

Source: Cadence

Advantages of Virtual HILS-System-Platforms with ESL



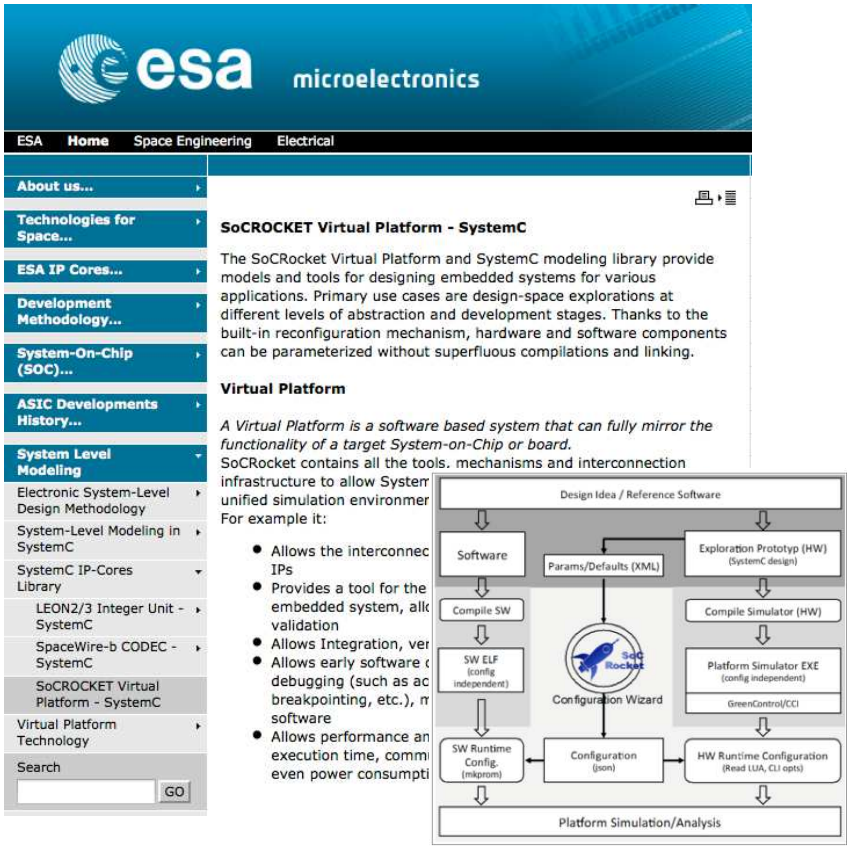
- Already during Hardware design phase software design and fast, virtual debugging in parallel is possible
- In contrast to hardware prototypes, system configuration can be changed with little effort, e.g. to evaluate other busses, protocols or hardware components
- Direct connection of the ESL framework enables early design, testing and debugging of the board-network

- Hardware simulation enables IP protection against external developers. Hardware Modules can be swapped as binaries (encapsulation of IP)
- Error Correction effort in the late development process can be significantly reduced through „virtual hardware in the loop“ methodology
- ESL-Platform can be replicated arbitrarily and used worldwide on servers (no hardware development boards needed). This enables development in parallel
- ESL can be used as a strategic tool for comparison of supplier hardware or software building blocks in board networks
- Integration and evaluation of HW/SW component from different suppliers can be achieve dwithin hours

EMC2 SoCRocket – Virtual Platform

Overview

Cooperation with European Space Agency (ESA) for research in advanced development methodologies for aerospace systems.



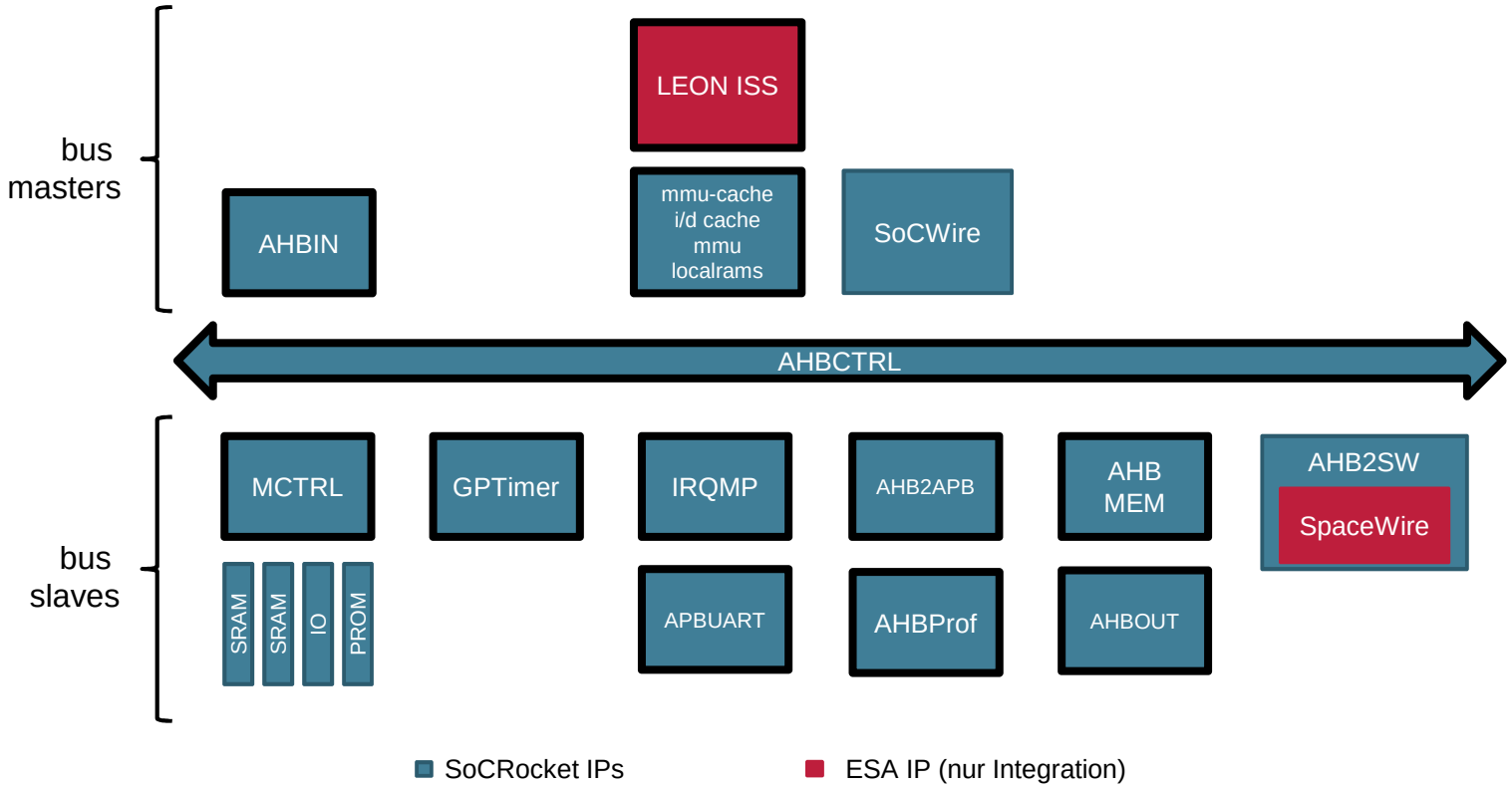
www.esa.int/TEC/Microelectronics/SEMW9XK1QAH_0.html

SoCRocket

- SystemC/TLM modeling library with Aerospace IP
- Design Flow for architecture exploration at different levels of abstraction
- Simulation and analysis infrastructure

SoCRocket – Virtual Platform

SystemC/TLM modeling library

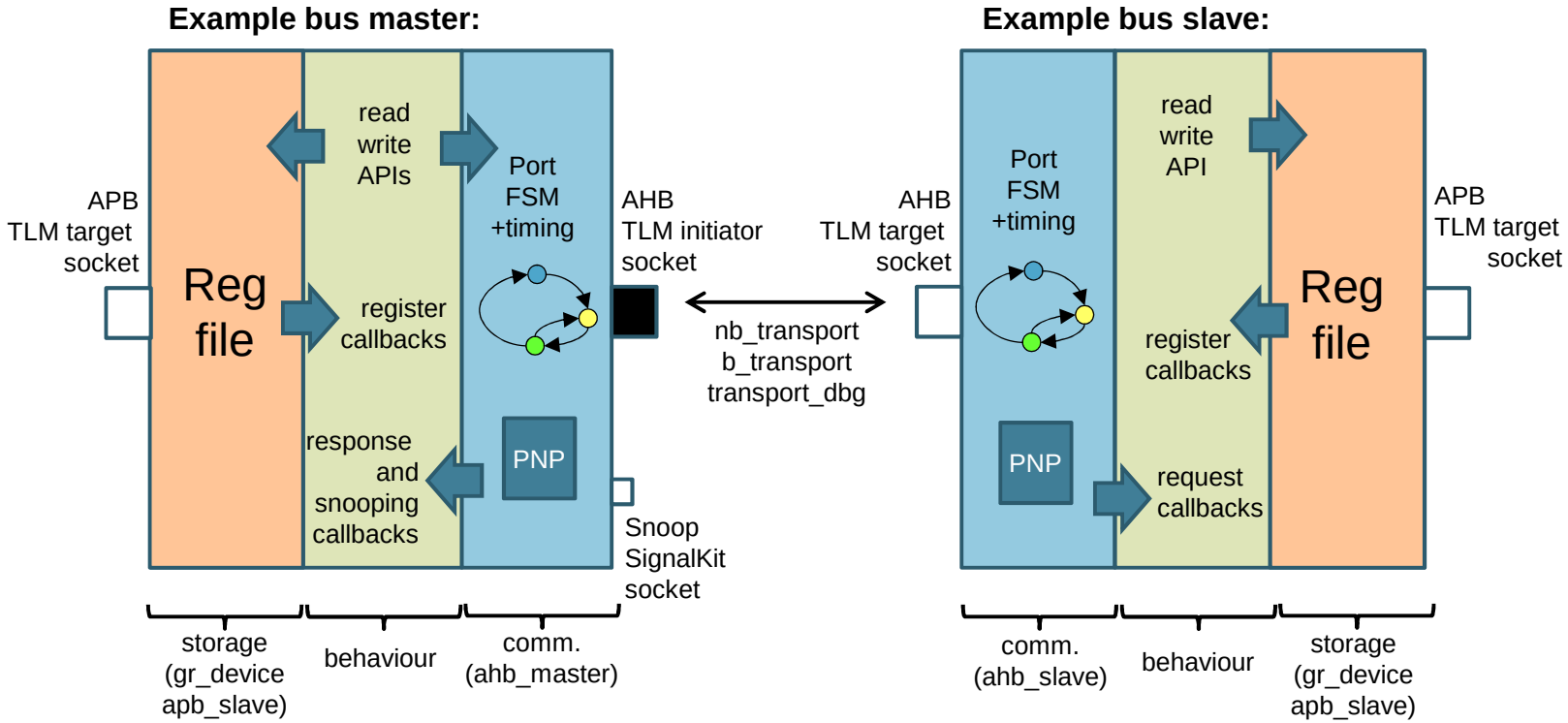


- Contains core components for construction of aerospace SoCs

SoCRocket – Virtual Platform

Component design methodology

- Building blocks for generating new components, based on inheritance from library classes
  (AHB/APB Master/Slave, Signal Master/Slave, ClkDevice, GreenReg, …)
- Support for loosely-timed (LT) and approximately-timed abstraction (AT)

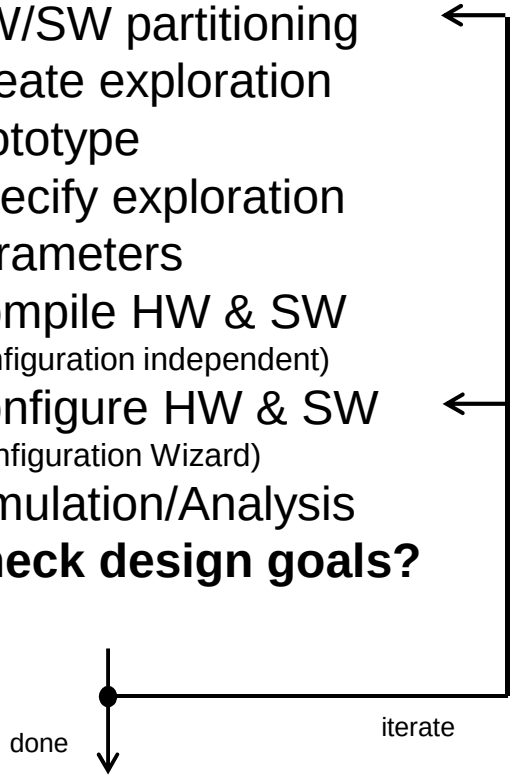


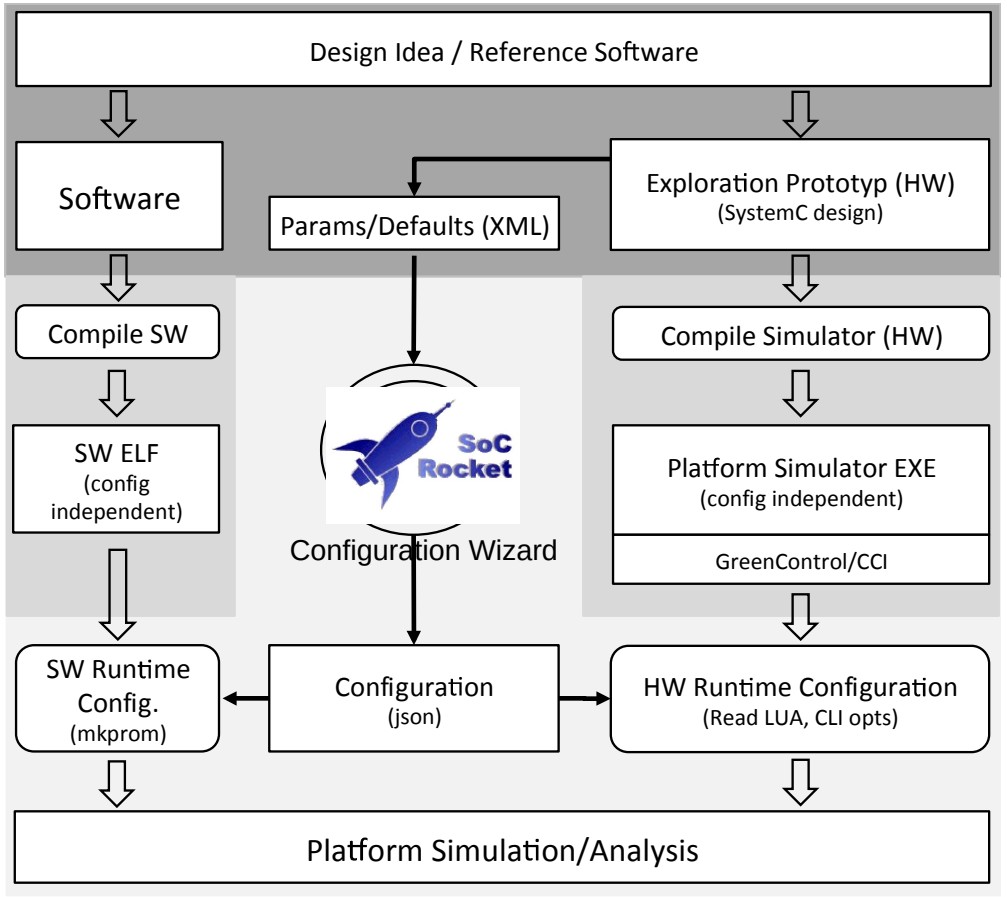
SoCRocket – Virtual Platform

Design Flow for construction of Virtual Prototypes

Path to prototype:

1. HW/SW partitioning
2. Create exploration prototype
3. Specify exploration parameters
4. Compile HW & SW (configuration independent)
5. Configure HW & SW (Configuration Wizard)
6. Simulation/Analysis **Check design goals?**





SoCRocket – Virtual Platform

High-Level DSE Demonstration

Use case:

Lossless multi-spectral / hyper-spectral image compression

Software environment:

RTEMS Real-Time OS

Exploration parameters:

- Number of icache banks (1 .. 4)
- Capacity of icache banks (1kB .. 8 kB)
- Number of dcache banks (1 .. 4)
- Capacity of dcache banks (1kB .. 8 kB)
- Number of CPUs (2, 4, 6)

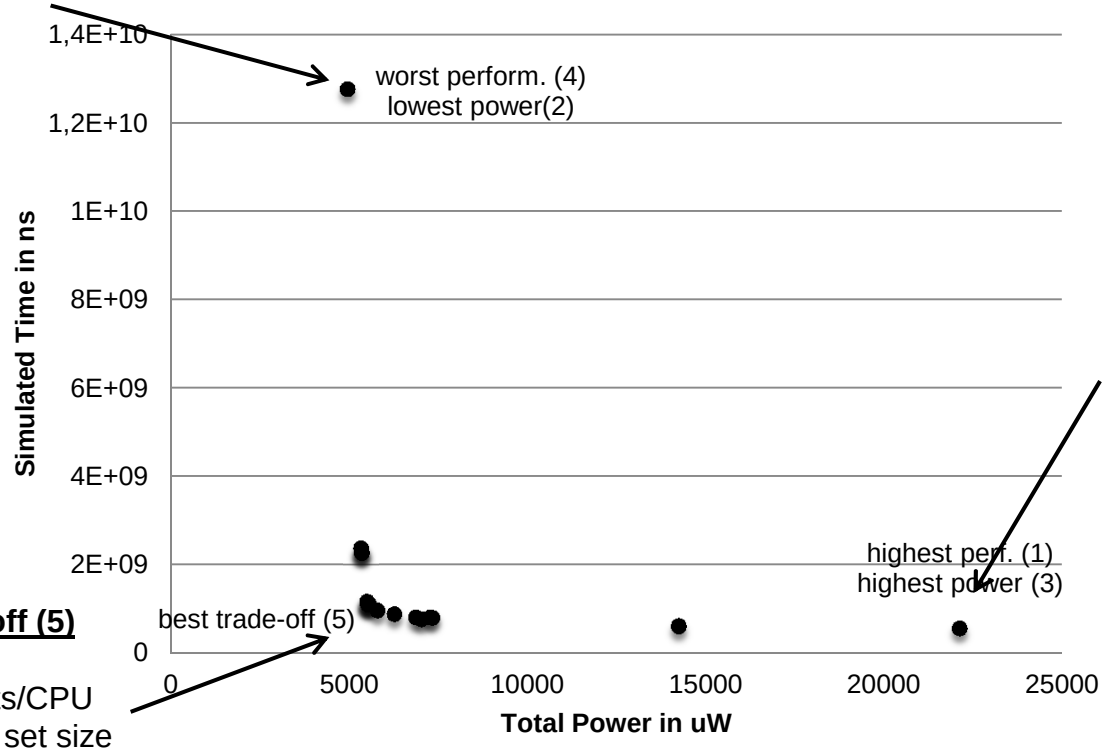
Optimization goal:

- Simulation time
- Average power consumption

SoCRocket – Virtual Platform

Power vs. Performance

Lowest perf., lowest power (2, 4)
2x CPU
1 dcache sets/CPU
1 kB dcache set size
1 icache set/CPU
1 kB icache set size



Highest perf., highest power (1, 3)
6x CPU
4 dcache sets/CPU
8 kB dcache set size
3 icache sets/CPU
4 kB icache set size

Best trade-off (5)
2x CPU
4 dcache sets/CPU
1 kB dcache set size
3 icache sets/CPU
4 kB icache set size

Conclusion

- ESL/TLM design methodology for HW/SW-Systems enables register or binary compatible hardware simulation models that run in real-time.
- Early Availability of prototypes enables early SW- Design and System integration on the target platform.
- Integration and System-Validation at OEM (virtual HILS) can begin before availabilty of HW (A-Muster).
- vHILS offers an early option to Validiate parts of the Reqirements („Lastenheft“) before the A-Muster.
- vHILS enables a quick evaluation of ECU-Implementation variants (HW/SW): CPU, Memory, Performance, Load, Costetc.

Conclusion

- New C++ based Modeling Technology that focuses on communication
- Single language for hardware and software eases hardware/software co-design
- Abstract Models offer faster path to modifications, easier maintainability
- Fully binary compatible VP models at different levels of granularity
- Simulation speeds close to real-time make use of VPs in design space exploration possible (“Virtual Hardware Platform in the Loop”)
- Enables replacement of ES Hardware System Design Platforms with pure Software Solutions
- New Methodologies for non-functional requirements validation, e.g. Fault Injection, reliability, power, ….