Technische Universität Braunschweig

HW/SW Co-Verification SoC Validation Platform

AO6025 - Final Presentation

Thomas Schuster, Rolf Meyer 06.12.12

ESA Technical Officer: Luca Fossati

Outline

PART I

- Overview
- Survey of Tools & Techniques
- High-Level Modeling of SystemC IP
- Verification of TL models
- Power Modeling
- VP Development and Validation
- Proof-of-Concept VP
- High-Level DSE demonstration
- Ongoing/future work

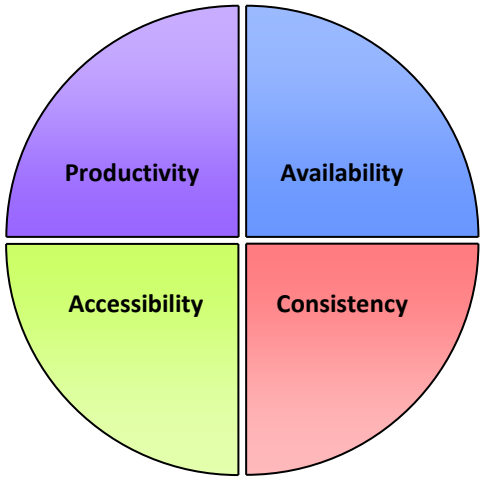
PART II

Demo

Overview

Intended use of activity output

- Enable ESA to develop new Virtual Platforms based on a set of key SystemC IP simulation models
- Distribute platform and models to contractors without being subject to any fee or restrictions
- Improve future SoC Design efficiency:
  - *Enable SW development before HW is ready*
  - *Gradual system refinement from TLM to RTL*
  - *Benefit from early architecture exploration*

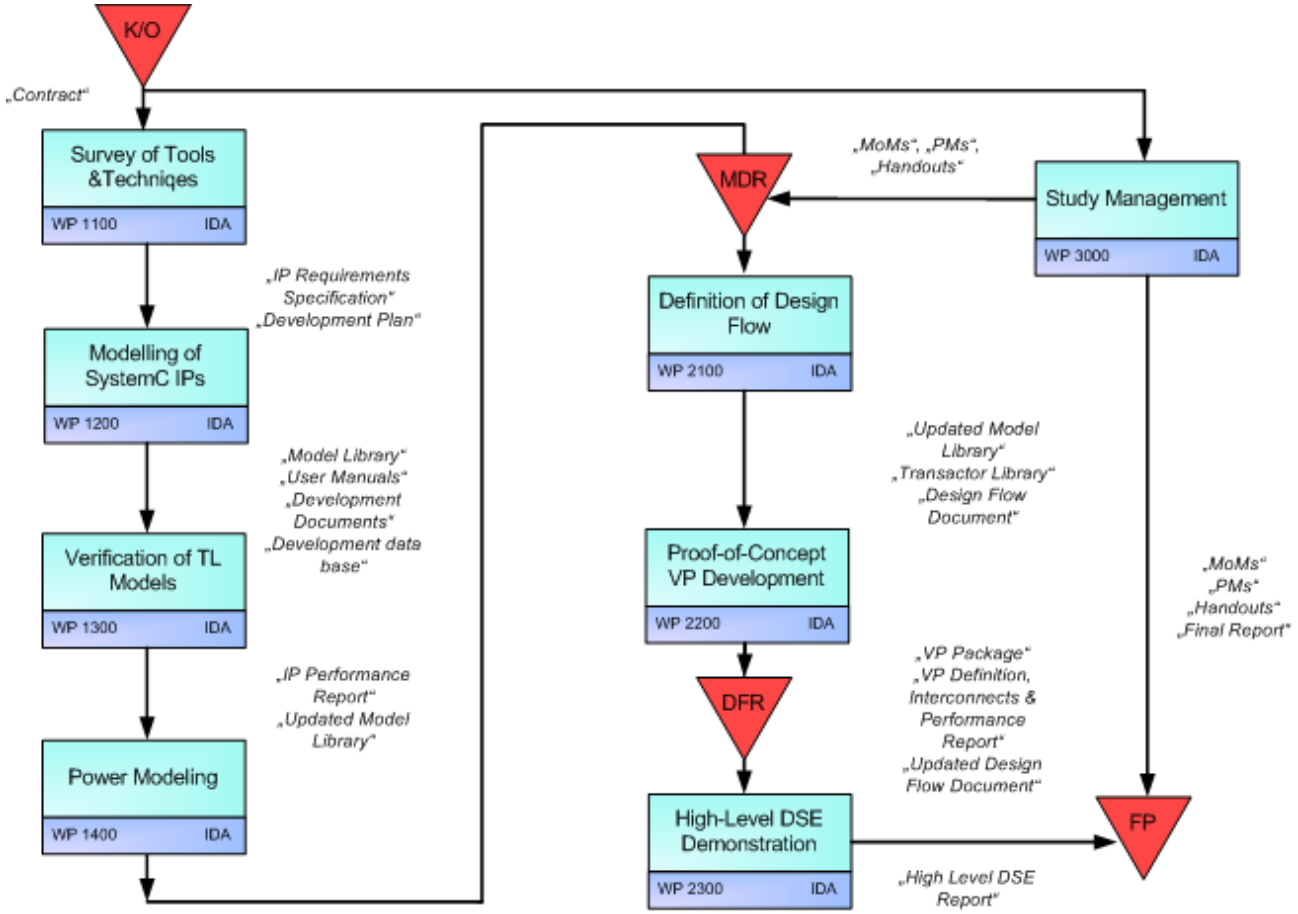


Advantages of virtual platforms

Objectives
Study Objectives

- The definition of a Design Flow
- High-level modeling of key IPs in SystemC TLM2.0
- Functional validation and timing accuracy analysis
- Development of Virtual Platform infrastructure
- Development of a Proof-of-Concept Virtual Platform (VP)
- Simulation performance and accuracy analysis of the proof-of-concept VP

Overview
Study logic



Survey of Tools & Techniques (WP1100)

Platform utilizes GreenSoCs TLM infrastructure

Explored commercial and open-source Virtual Platform solutions.

Selected appropriate infrastructure components:

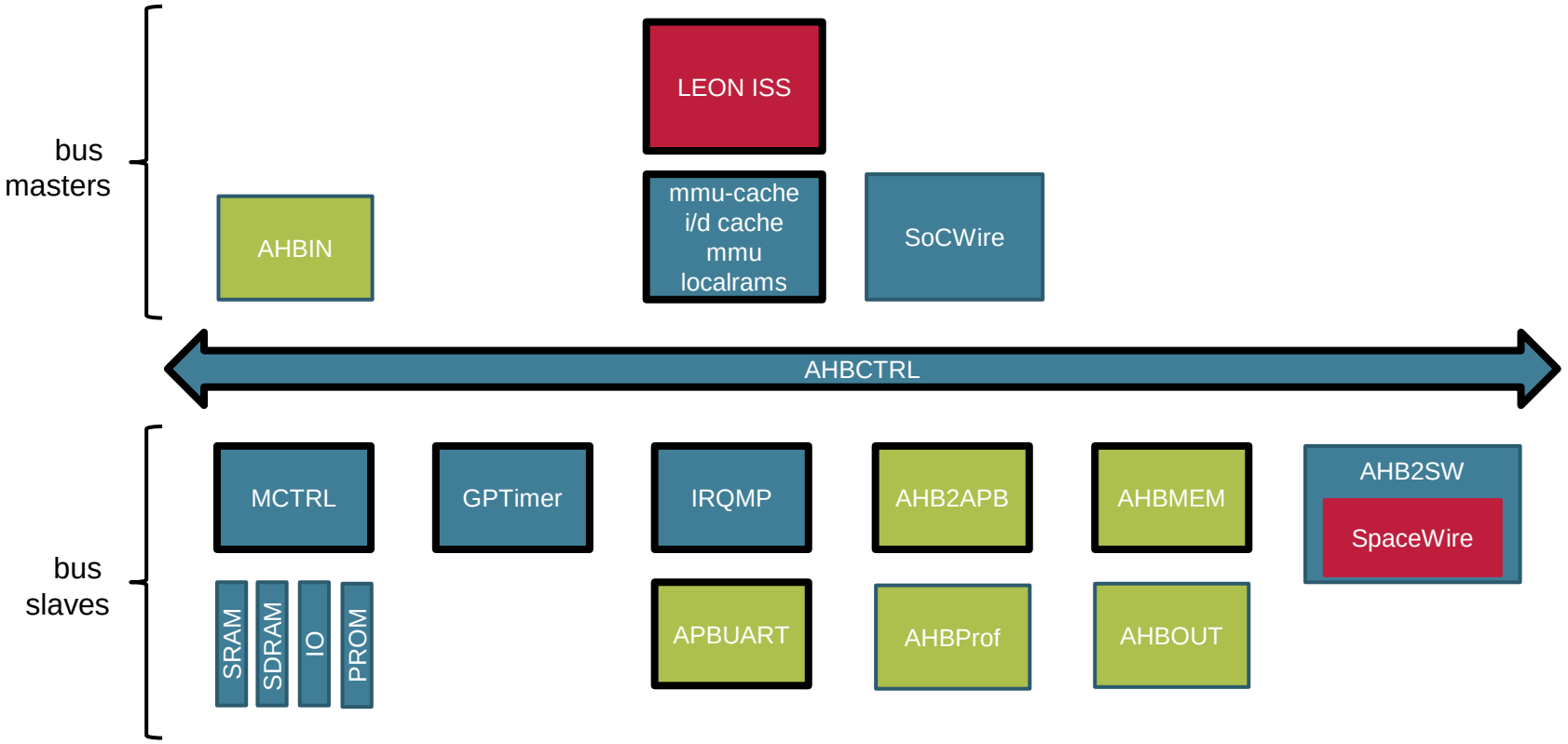
- **GreenReg**
  - Framework for SystemC register modeling
  - Register bank container which can be bound to TLM sockets
  - Bit-field callback functions etc.
- **AMBA Sockets**
  - TLM convenience sockets with build-in memory management
  - Predefined payload extensions for modeling AMBA bus transfers
- **GreenControl**
  - Parameter API for model configuration
  - Debugging, Tracing, runtime-reconfiguration

GreenSocs®

www.greensocs.com

High-Level Modeling of SystemC IPs (WP1200)

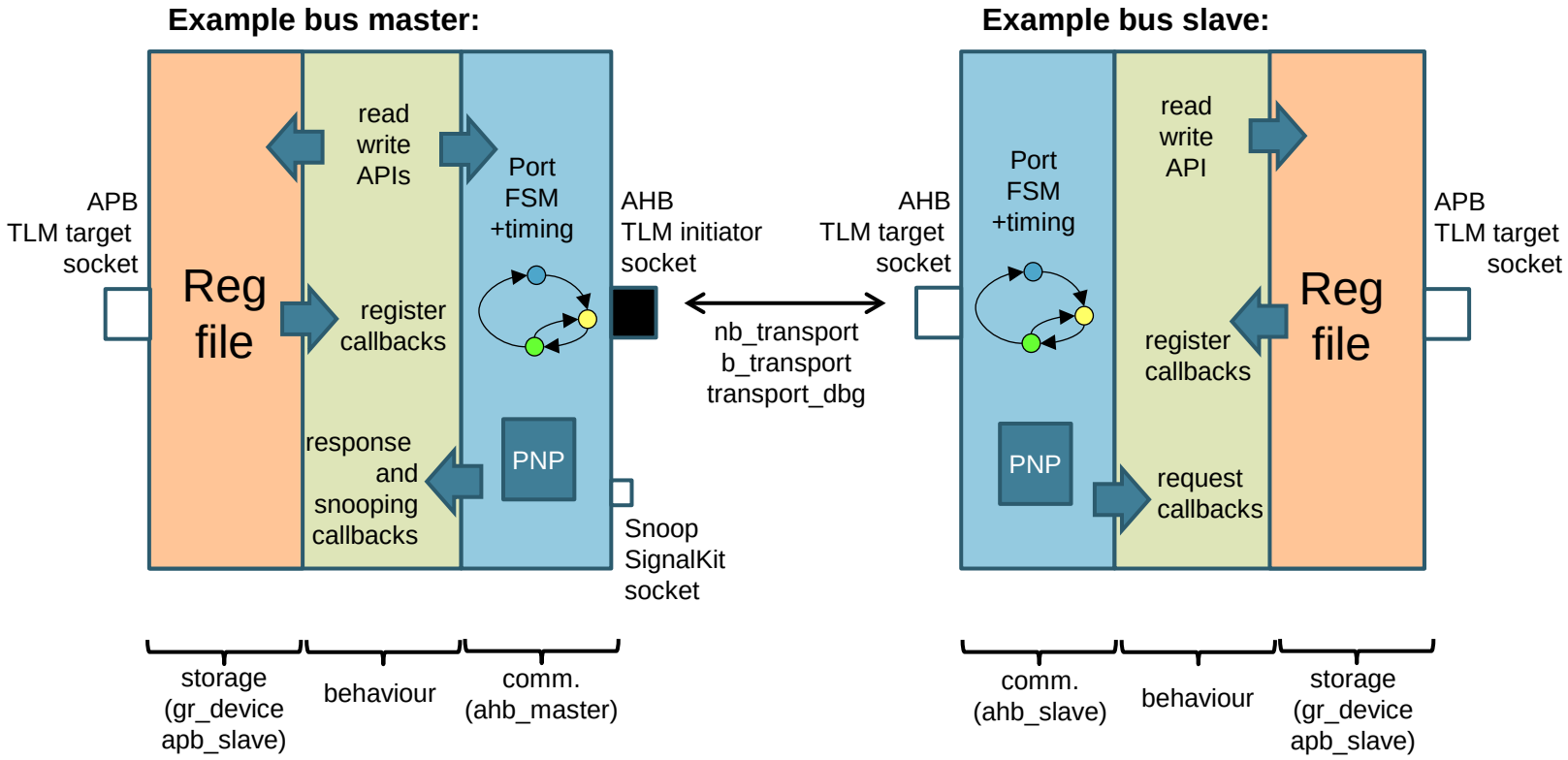
IPs modeled in course of project

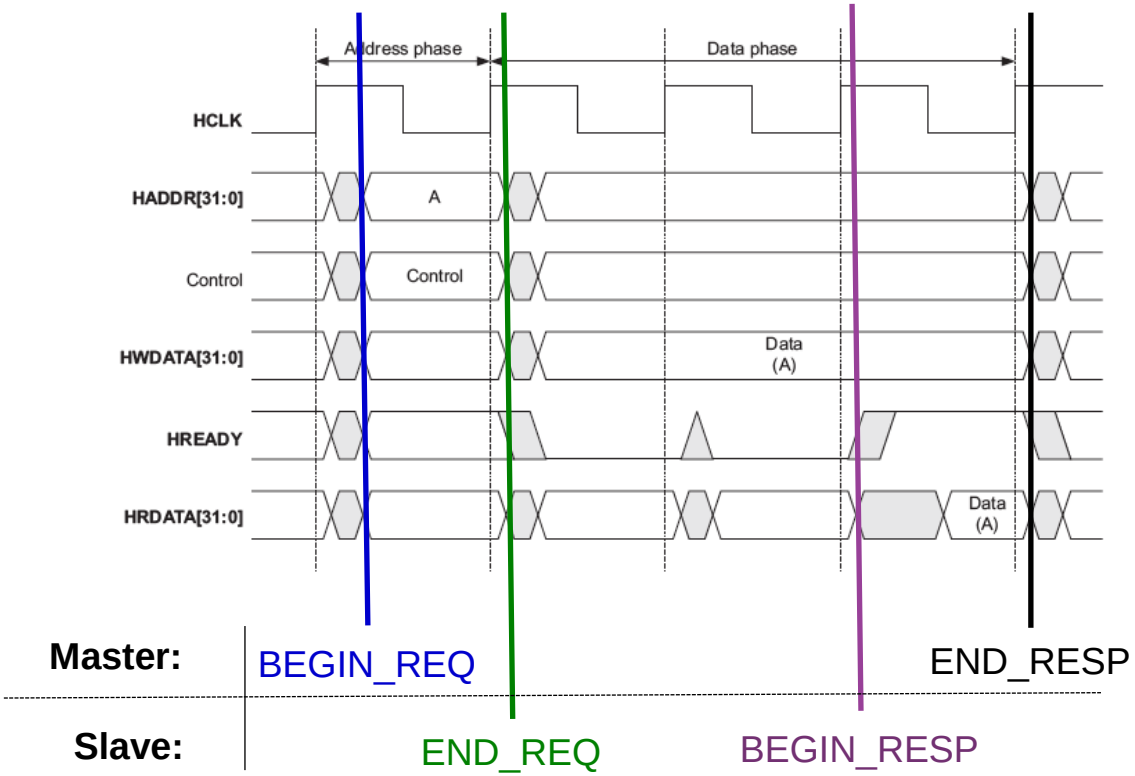


All models available in loosely timed (LT), and approximately timed (AT) flavor of TLM2.0.

- Models provided by ESA (integration only)
- Additionally/secondary models
- Mandatory/Optional IPs
- Models with GRLIB reference

- Library base classes for AHB masters, AHB slaves and APB slaves
- Models inherit library base classes and register file





Example read burst:

BEGIN_REQ
Time master sends the first address

END_REQ
Time slave samples the last address

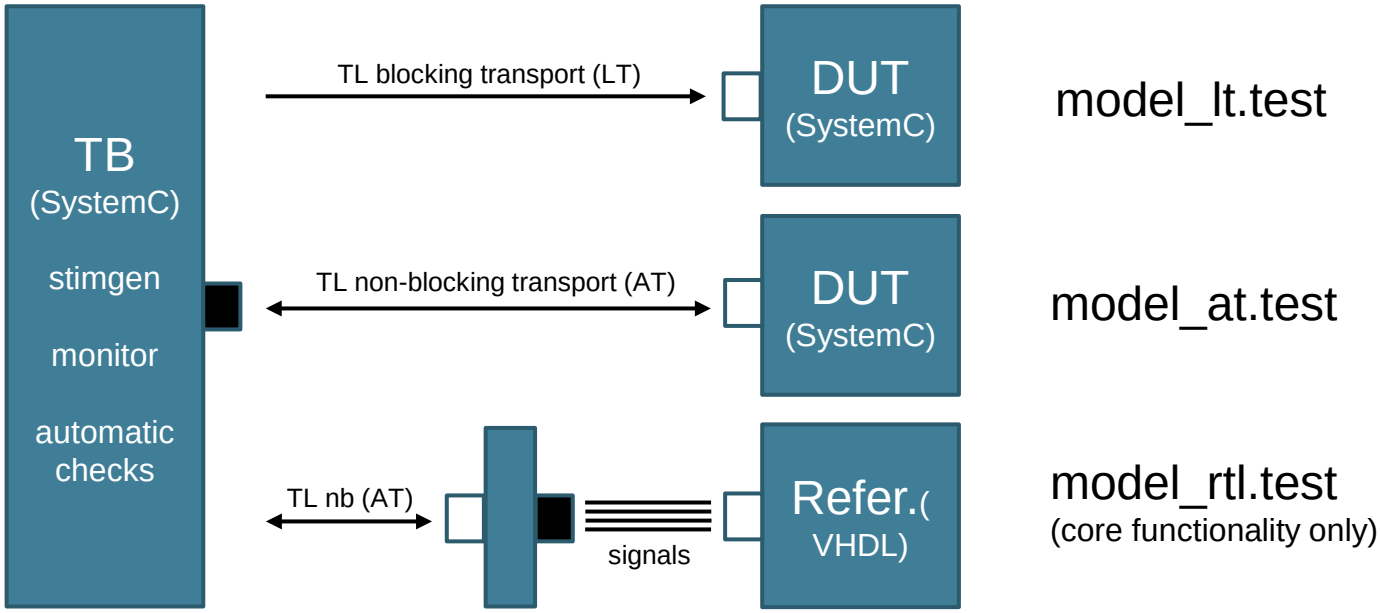
BEGIN_RESP
Time slave sends first data

END_RESP
Time master samples the last data

AHB protocol approximated using 4-phase non-blocking TL communication.

Verification of TL models (WP1300)

- Directed functional testing
- RTL Co-Simulation



Test coverage: ~90% Timing accuracy (AT)*: ~90%

*with respect to available RTL tests

Power Modeling (WP1400)

Concept

Provide indicator about impact of architectural decisions on power consumption.

- RTL reference models synthesized using generic 90nm ASIC library
  http://www.synopsys.com/Community/UniversityProgram/Pages/Library.aspx
- Executed statistical power analysis

Results (Model power inputs):

- Normalized leakage power (static power/area)
- Normalized cell internal power (dynamic power/area/clock)
- Normalized per-access energy for memories

Power estimation:

- Models de-normalize power inputs using construction parameters
  (e.g. clock, number of counters in a timer)
- Reads/writes of memories are counted and multiplied with energy per access.

Power Modeling (WP1400)
Monitoring

- Power monitoring can be enable per IP (constructor parameter)
- Library provides default power monitor
- Default power monitor reports average power consumption at end of simulation (per IP and total):

```
@2130 us: Info:  ***************************************************
@2130 us: Info:  * Power Summary:
@2130 us: Info:  * -----------------------------------------------------
@2130 us: Info:  * Static power (leakage): 1217.16 uW
@2130 us: Info:  * Internal power (dynamic): 1920.78 uW
@2130 us: Info:  * Switching power (dynamic): 898.567 uW
@2130 us: Info:  * -----------------------------------------------------
@2130 us: Info:  * Total power: 2112.73 uW
@2130 us: Info:  ***************************************************
```

*LEON3MP single processor running “Hello World” from SDRAM.

Results are only indications – Power numbers not verified !!

VP Development and Validation (WP2000)

Infrastructure

Developed in the course of this project:

Tailored build system (WAF):

- Host system configuration (env set and checks)
- Centralized control for all build processes and tools
- Automatic unit tests
- Modelsim co-sim integration

SignalKit:

Modeling kit for fast TL signal communication.

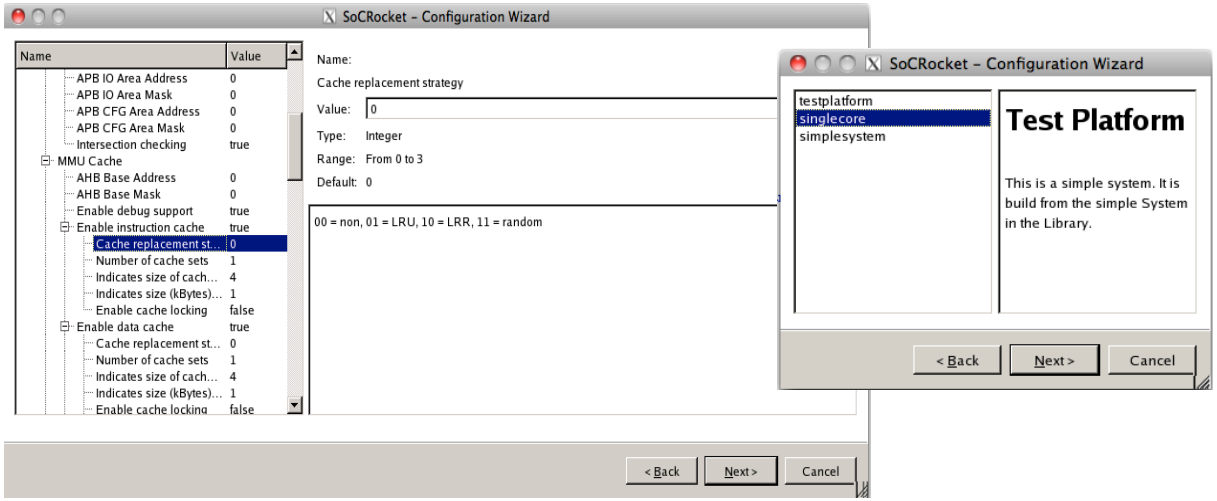
Used for modeling interrupt lines, reset and snooping

Verbosity control:

Debug output configuration and filtering

Configuration Wizard:

Tool for system configuration in terminal or GUI mode.



Doxygen integration

Code coverage analysis (gcov/lcov)

Command line tools

execute, get/set attribute, csv2json

Definition of Design Flow (WP2100)

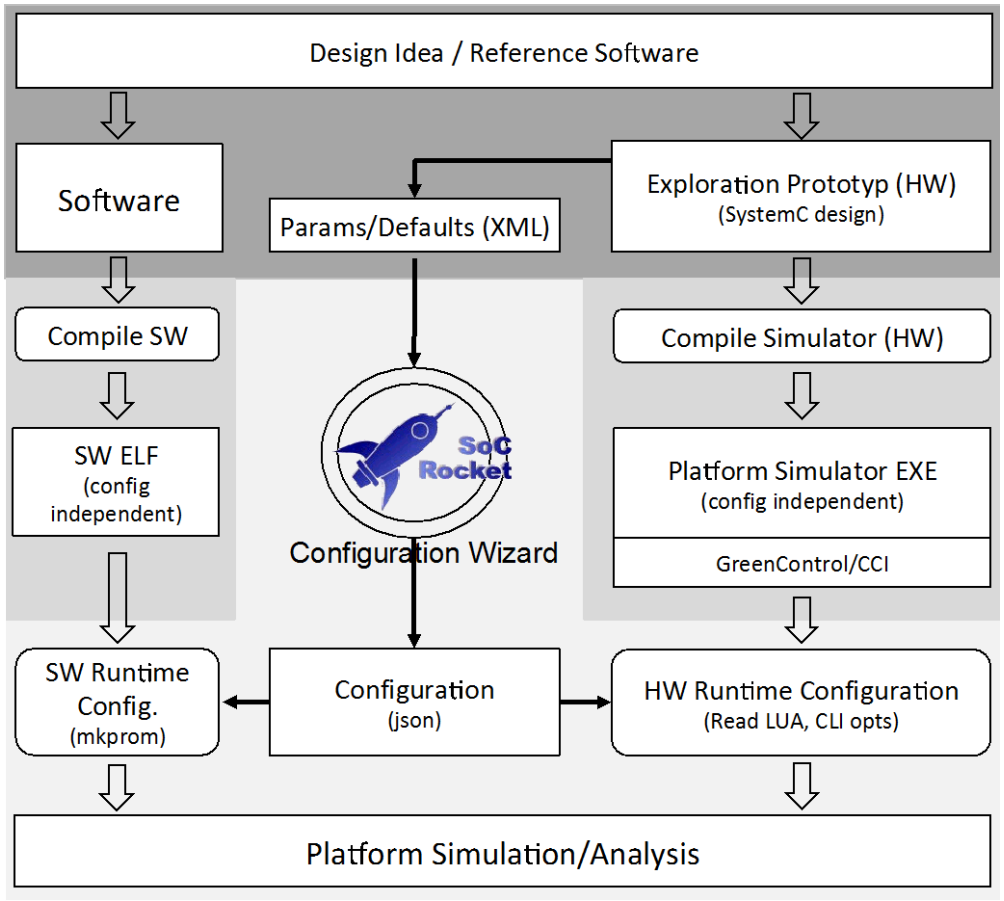
SoCRocket DF / Runtime re-configuration

Path from Design Idea
to VP Prototype

1. HW/SW partitioning
2. Create Exploration Prototype
3. Extract config parameters
4. Compile HW & SW (config independent)
5. Configure HW & SW (configuration wizard)
6. Simulation/Analysis **Design goals met ?**

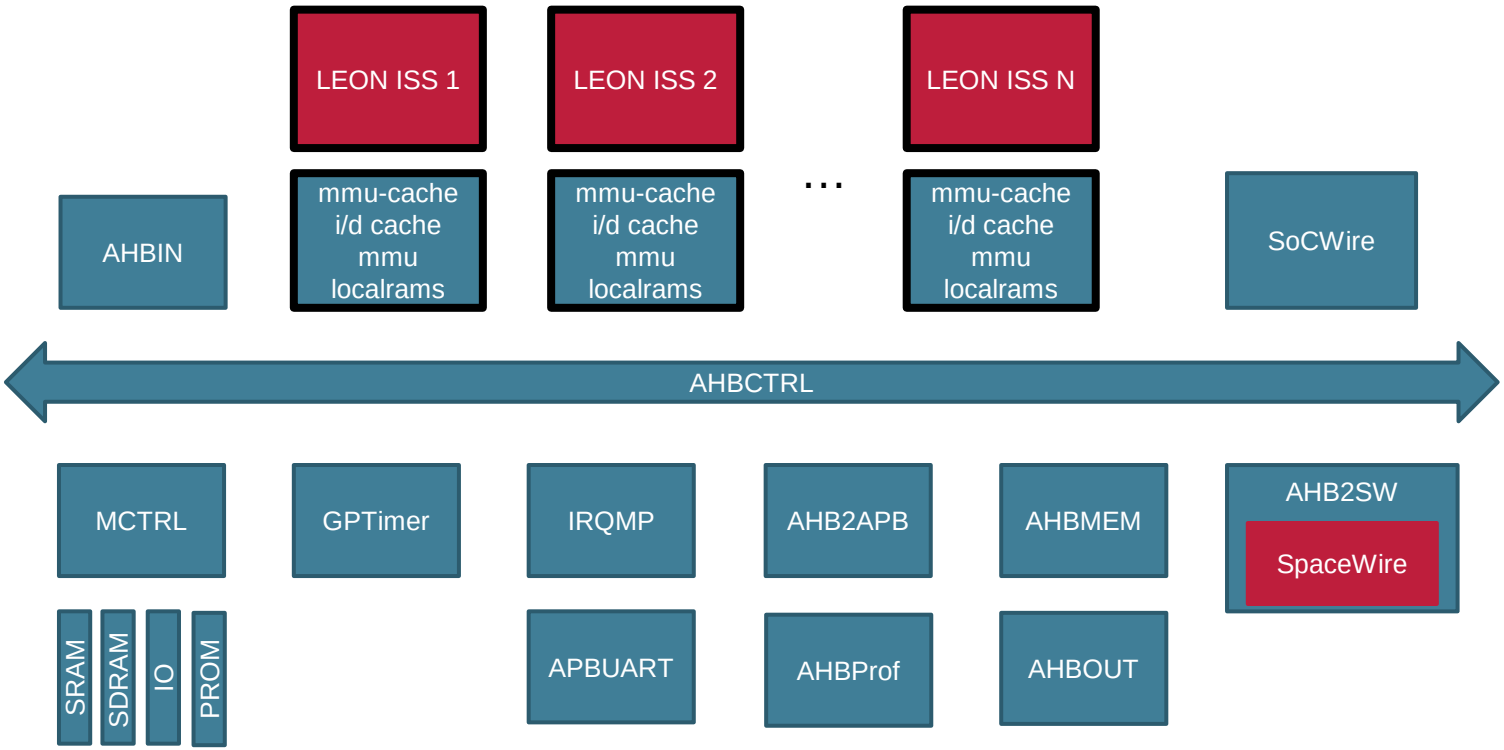
iterate

done



(From SoCRocket Design Flow Report)

Proof-of-concept VP Development (WP2200)
LEON3MP



LEON3MP prototype as shipped with the platform library.

Performance Counters

- Monitor performance and behavior of simulation models
- Organized in global namespace:

  e.g. Number of transactions routed by AHBCTRL

  top.ahbctrl.performance_counters.total_transactions
- All counters can be accesses at runtime

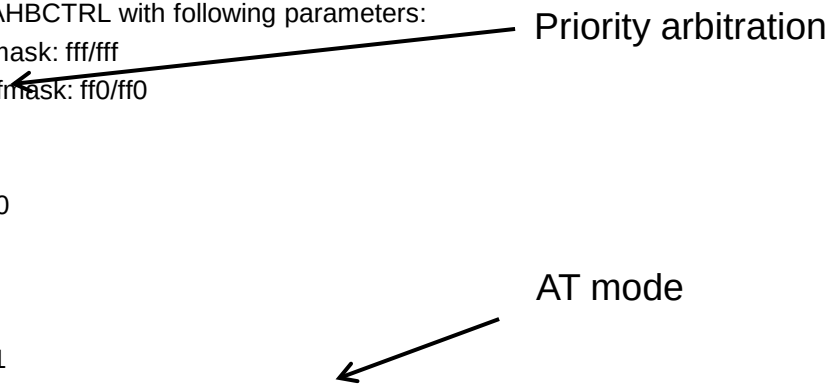
Analysis API (based on GreenControl/AV):

- Read/write counters
- List counters
- Trace changes in logfiles
- Write wave form files (VCD)
- Register callback functions

Configuration Reports

Optionally generated at begin of simulation
(e.g. ahbctrl)

```
********************************************************************************
@0 s /0 (ahbctrl): Info:  * Created AHBCTRL with following parameters:
@0 s /0 (ahbctrl): Info:  * ioaddr/iomask: fff/fff
@0 s /0 (ahbctrl): Info:  * cfgaddr/cfmask: ff0/ff0
@0 s /0 (ahbctrl): Info:  * rrobin: 0
@0 s /0 (ahbctrl): Info:  * split: 0
@0 s /0 (ahbctrl): Info:  * defmast: 0
@0 s /0 (ahbctrl): Info:  * ioen: 0
@0 s /0 (ahbctrl): Info:  * fixbrst: 0
@0 s /0 (ahbctrl): Info:  * fpnpen: 1
@0 s /0 (ahbctrl): Info:  * mcheck: 1
@0 s /0 (ahbctrl): Info:  * pow_mon: 1
@0 s /0 (ahbctrl): Info:  * abstractionLayer (LT = 8 / AT = 4):  4
********************************************************************************
```



Simulation Reports

Optionally generated at end of simulation
(e.g. data cache)

```
**************************************************************************
@246700 ns (mmu_cache.dcache): Report:  * Caching statistic:
@246700 ns (mmu_cache.dcache): Report:  * ------------------
@246700 ns (mmu_cache.dcache): Report:  * Read hits set0: 256
@246700 ns (mmu_cache.dcache): Report:  * Read hits set1: 256
@246700 ns (mmu_cache.dcache): Report:  * Read hits set2: 256
@246700 ns (mmu_cache.dcache): Report:  * Read hits set3: 256
@246700 ns (mmu_cache.dcache): Report:  * Total Read Hits: 1024
@246700 ns (mmu_cache.dcache): Report:  * Read Misses:  1024
@246700 ns (mmu_cache.dcache): Report:  * Read Hit Rate: 0.5
@246700 ns (mmu_cache.dcache): Report:  * Write hits set0: 0
@246700 ns (mmu_cache.dcache): Report:  * Write hits set1: 0
@246700 ns (mmu_cache.dcache): Report:  * Write hits set2: 0
@246700 ns (mmu_cache.dcache): Report:  * Write hits set3: 0
@246700 ns (mmu_cache.dcache): Report:  * Total Write Hits: 0
@246700 ns (mmu_cache.dcache): Report:  * Write Misses: 0
@246700 ns (mmu_cache.dcache): Report:  * Bypass ops:   4096
**************************************************************************
```

Hit rate

High-Level DSE Demonstration (WP2300)

Exploration Space

A Lossless multi spectral and hyper spectral image compression was mapped on the base platform.

(SW executed on top of 

OS)

Explored configurations:

- Number of instruction cache sets (1 .. 4)
- Capacity per icache set (1kB .. 8kB)
- Number of data cache sets (1 .. 4)
- Capacity per dcache set (1kB .. 8kB)
- Number of CPUs (2, 4, 6)

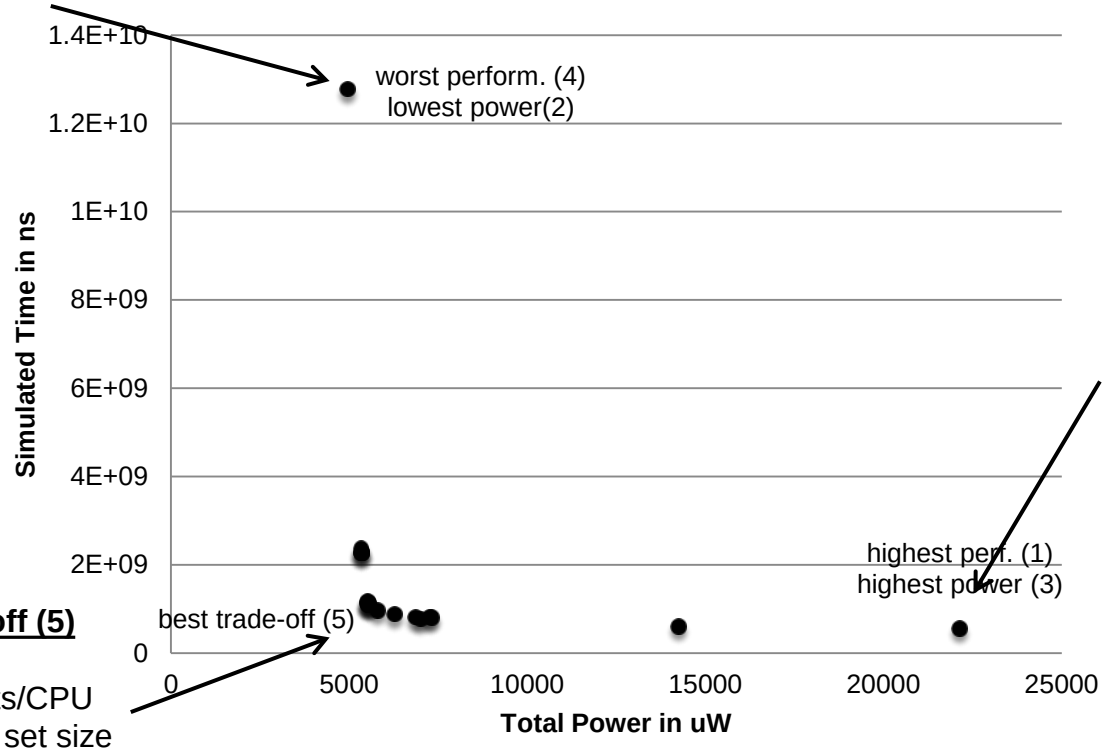
Optimization parameters:

- Simulation time
- Average power consumption

High-Level DSE Demonstration (WP2300)

Energy vs. Performance Trade-off

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

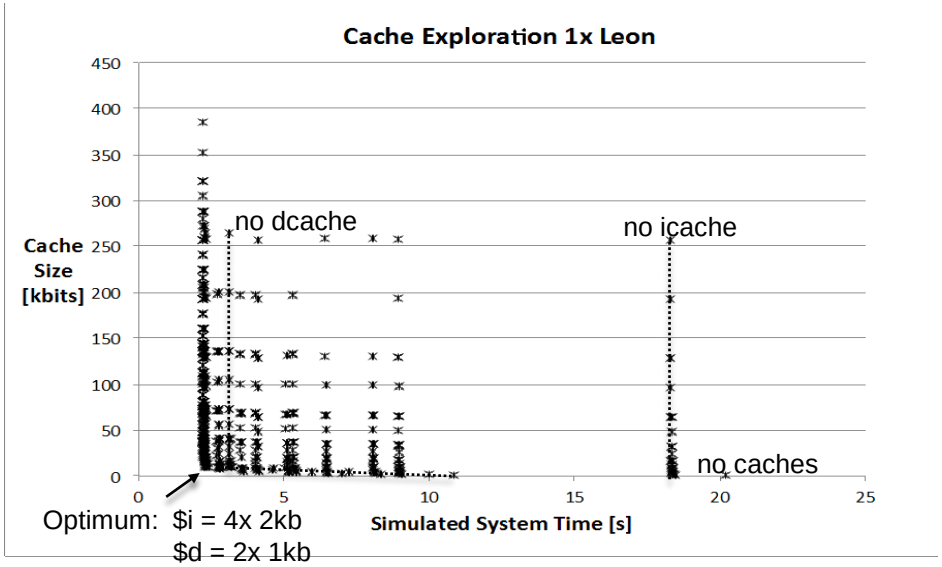


Highest perf., highest power (1, 3)
6x CPU
4 dcache sets/CPU
8 kB dcache set size
3 icache sets/CPU
4 kB icache set size

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

Ongoing/future work

- Automatic design space exploration

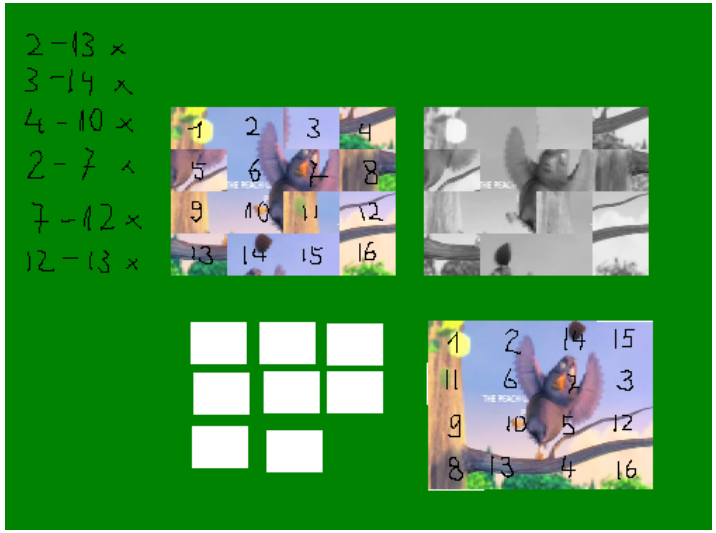


- Reliability modeling
- Teaching Integration (Curriculum SystemC/TLM)
- Development of CCSDS File Delivery Protocol IP Core (TLM & RTL)

PART II
SoCRocket demo

What we are doing with our students in the lab:

- Reordering of scrambled images received from Satellite



- Solution using software shuffling (on LEON ISS)
- Development of “shuffle” hardware accellerator
- Working with SoCRocket and RTEMS

Technische Universität Braunschweig

Thank you for your attention!

Question ??