“Interfacing concept for different levels of abstraction in IP-based SoC design“

by

Sven Verdenhalven & Holger Kühl
sci-worx GmbH
Hanover, Germany
www.sci-worx.com

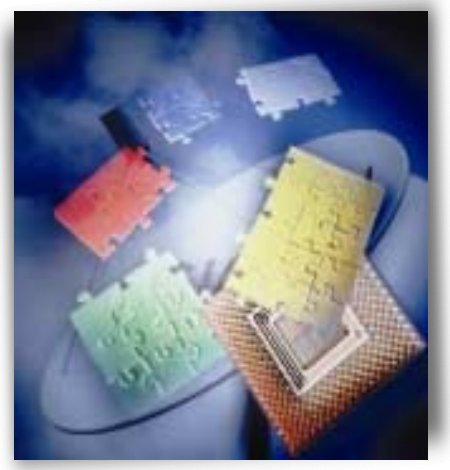
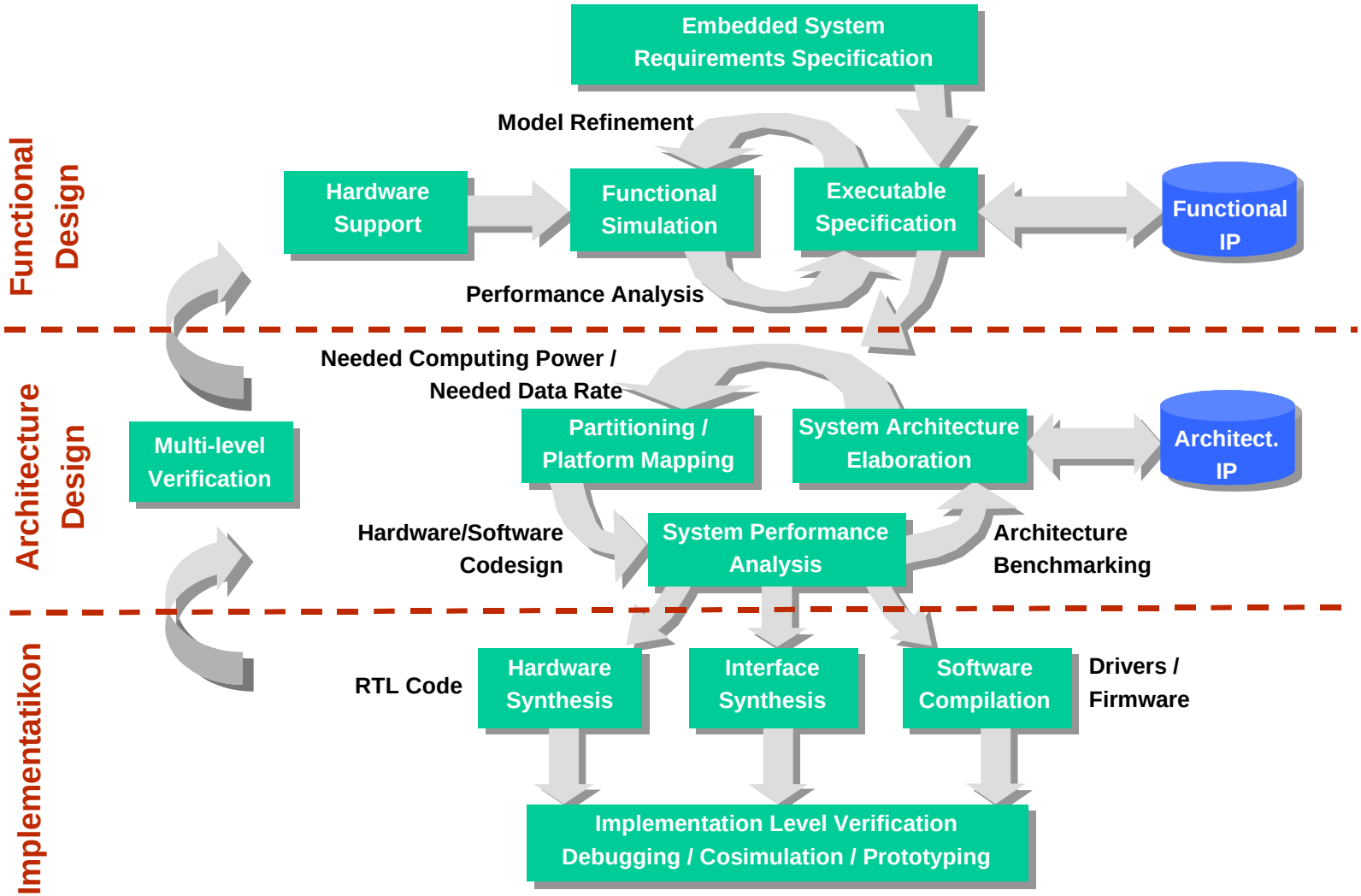


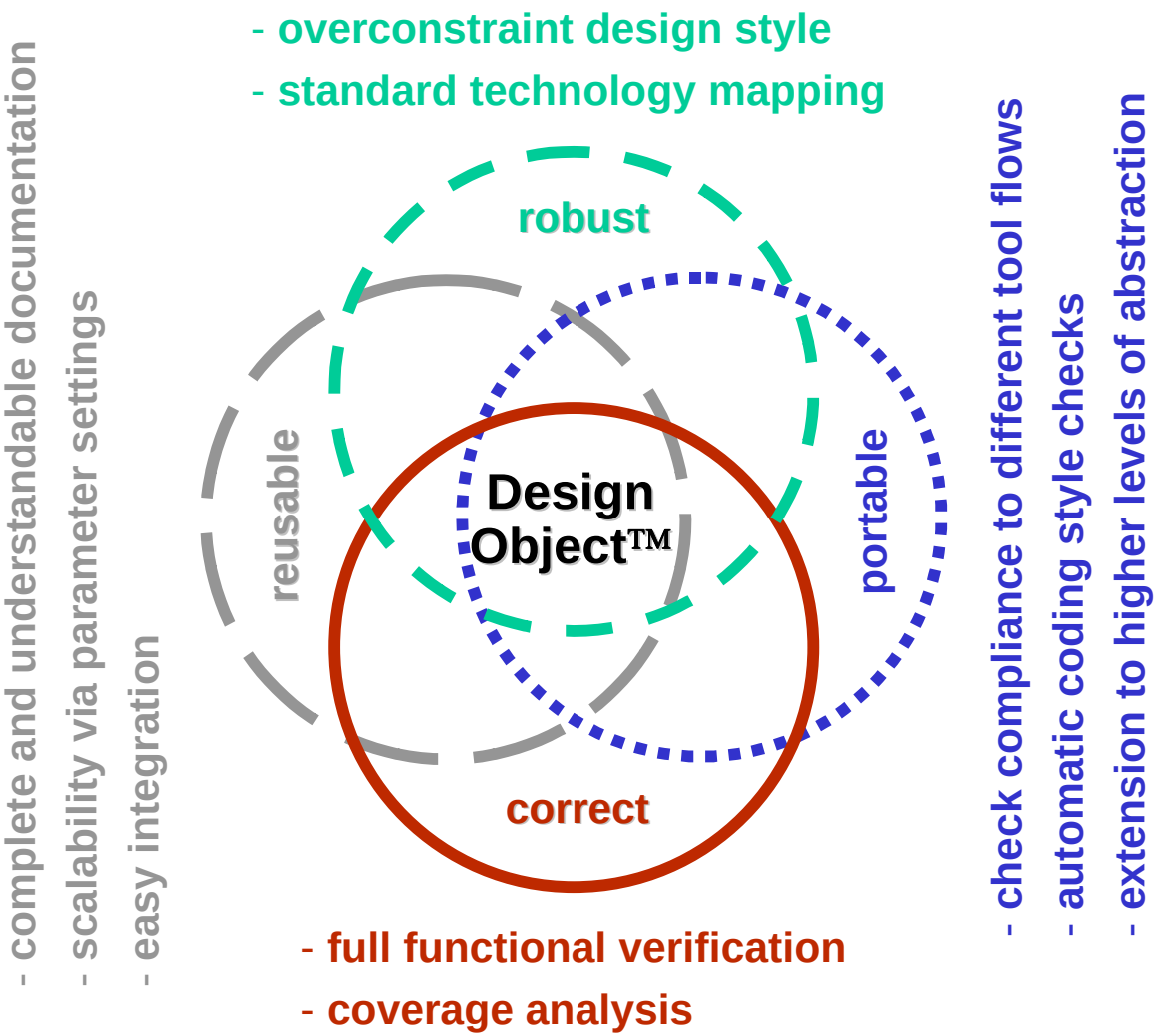
Table of Contents I

Table of Contents II

System-level Design Flow



IP Core Design Challenges



SCIWORX

Table of Contents

Standardization Activities in System-level Design

gain compatibility by standarization of:

- (open) modeling languages and extensions
  - systemC, Superlog, Java, UML-RT, SDL, e-Language, OpenVera, Testbuilder,..
- documentation structure
  - explaining documents format templates, ease understanding and utilization
- deliverables data formats
  - source code, models, specifications, reports, scripts,..
- architecture and interfacing
  - test structures, on-chip bus standards,..

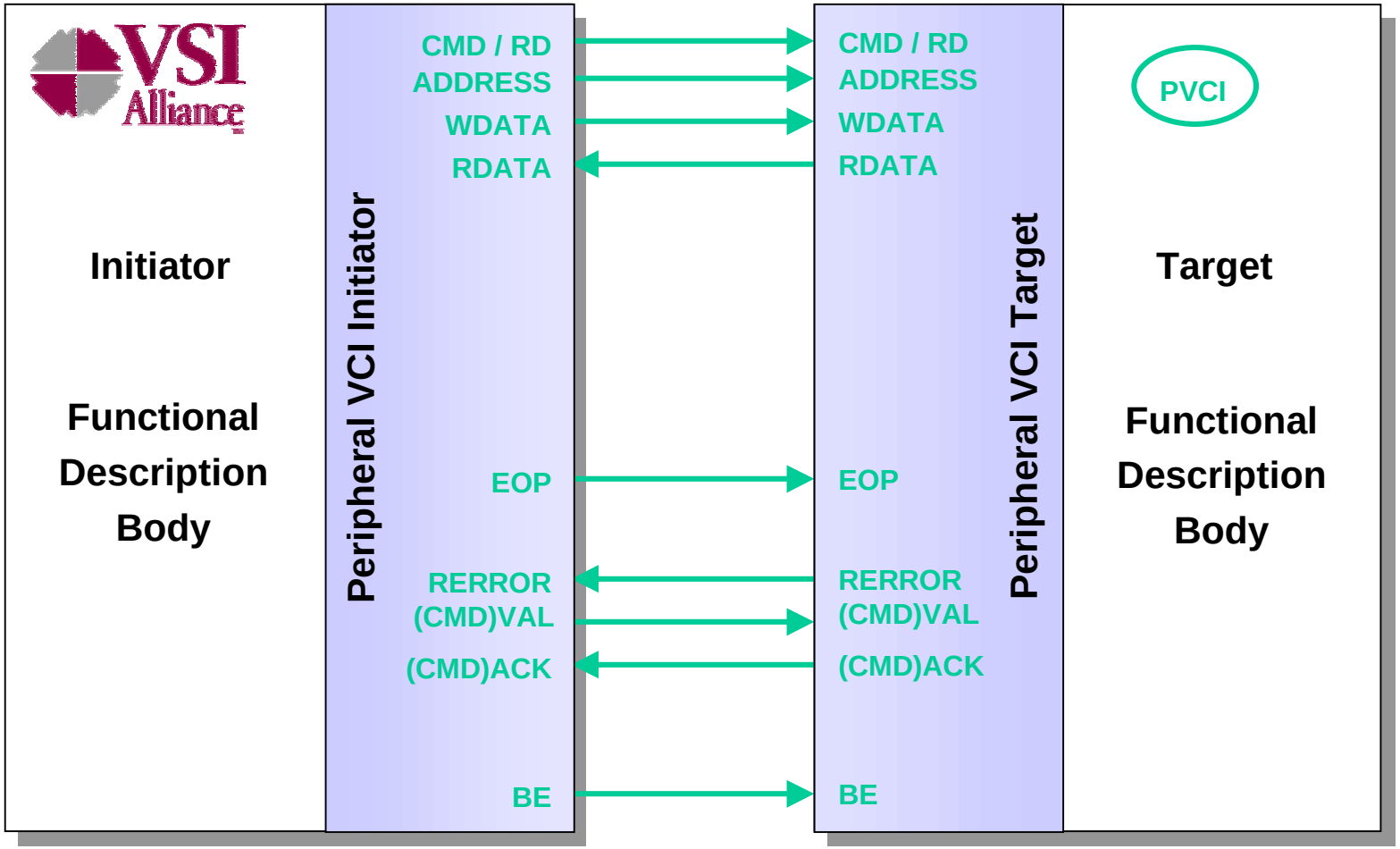
Virtual Socket Interface Alliance

www.vsi.org

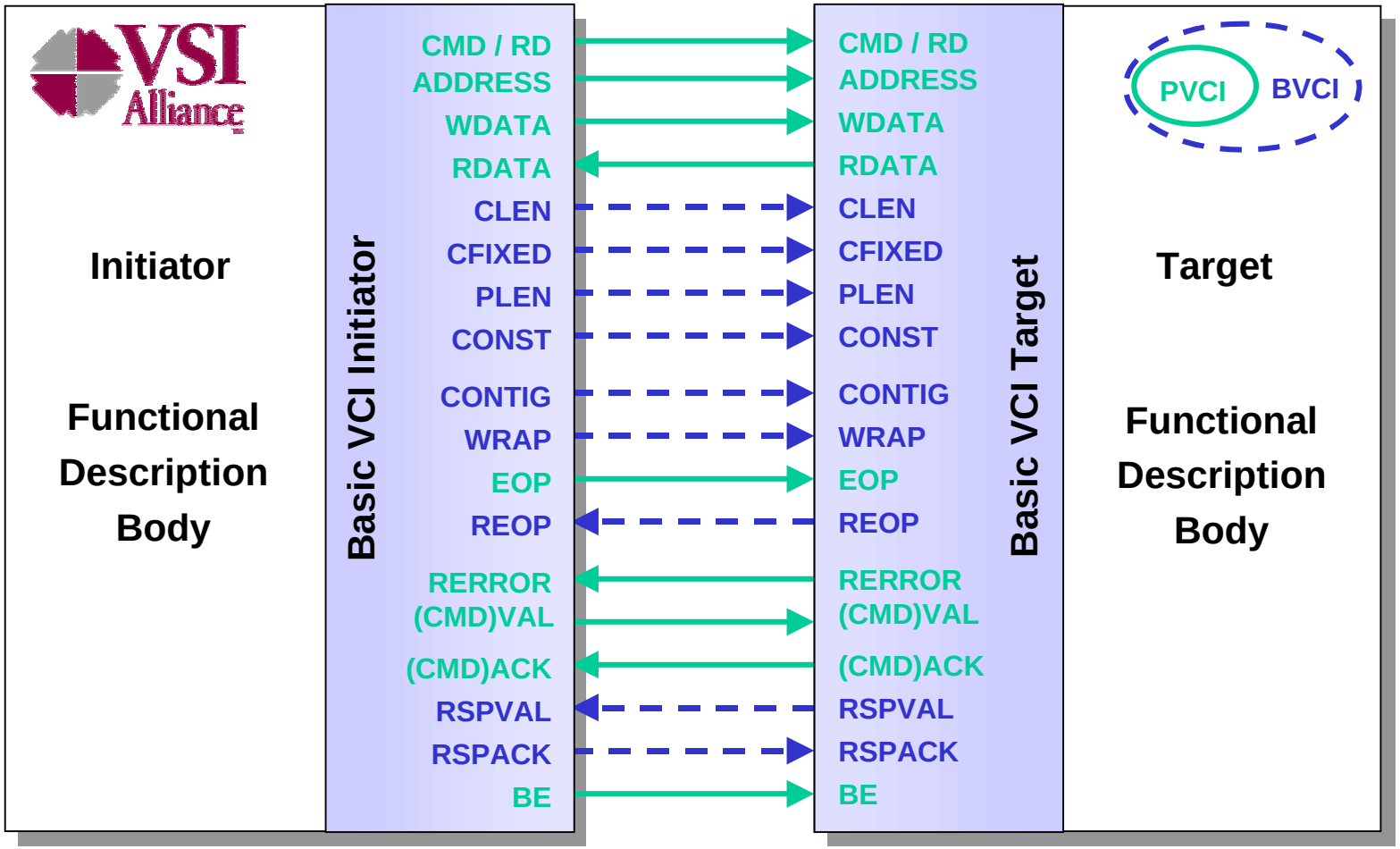
- founded in September 1996
- purpose: “encouragement of virtual component transfer and SoC integration”
- currently about 175 member companies
- different development working groups
  - system-level design, on-chip bus, VC quality, AMS, IP protection, VC transfer, impl. Verification, functional verification, manufac. test
- established different specifications, among those:
  - Deliverables document (DD 2.x)
  - System-Level Interface Behavioral Documentation Standard Version 1 (SLD 1 1.x)
  - Virtual Component Interface Standard Version 2 (OCB 2 2.x)

Table of Contents

Peripheral Virtual Component Interface



Basic Virtual Component Interface



VCI Uni-level Interface Adaptation

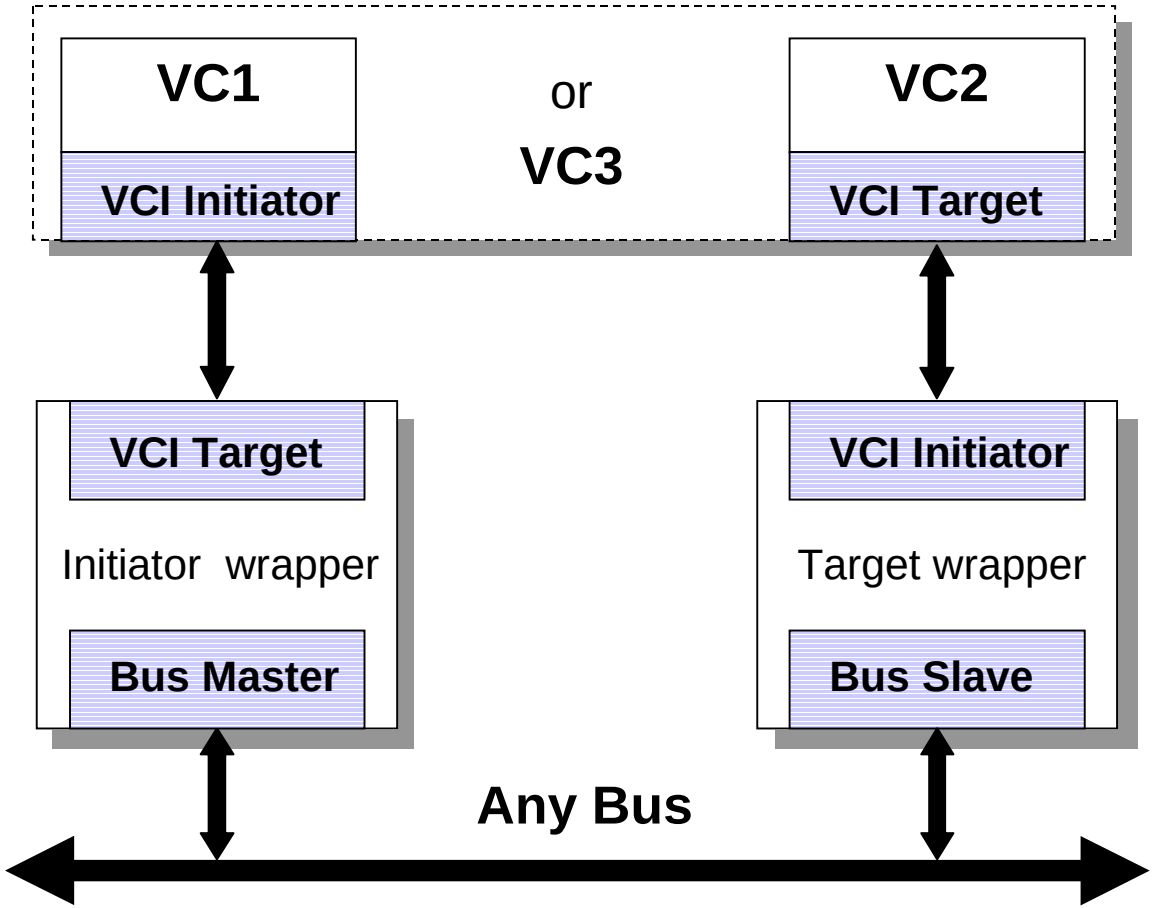


Table of Contents

Investigation Goals and Limits

multi-abstraction level system-IP design

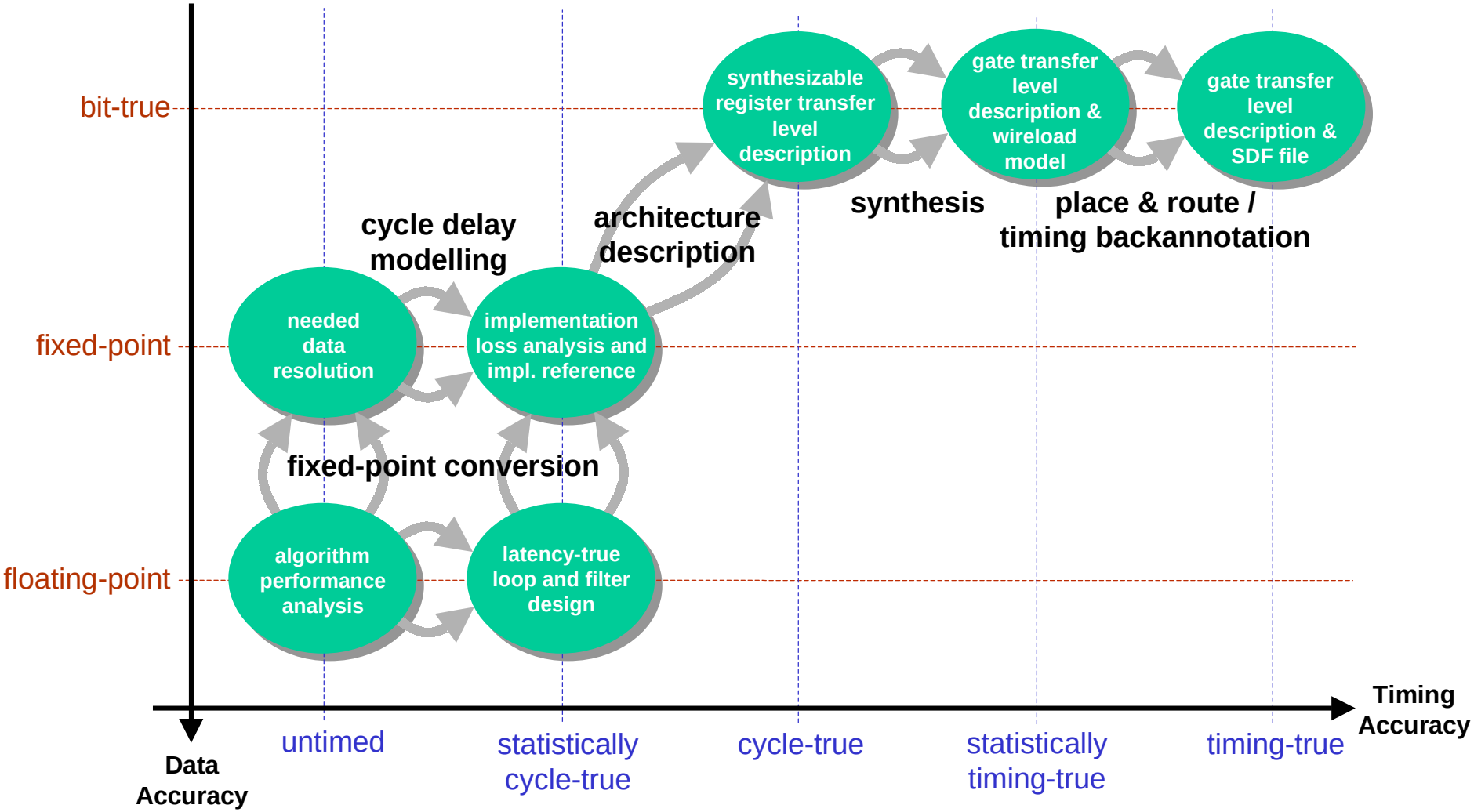
- first goals/interests
  - smooth model transition during functional and architectural design
  - gain VSIA compliance as far as possible
- framework
  - ***Spe****cification and* ***A****lgorithm/****A****rchitecture* ***C****o-Design...*
  - task 3.3 system-IP design
- limits
  - levels over and including RTL
  - initial focus HW module development at different levels of abstraction
- open issues
  - software appliance check
  - synchronization: software function – RTOS – HDL description
  - definition of mandatory used subset of model classes/combinations

System-level Model Classification I

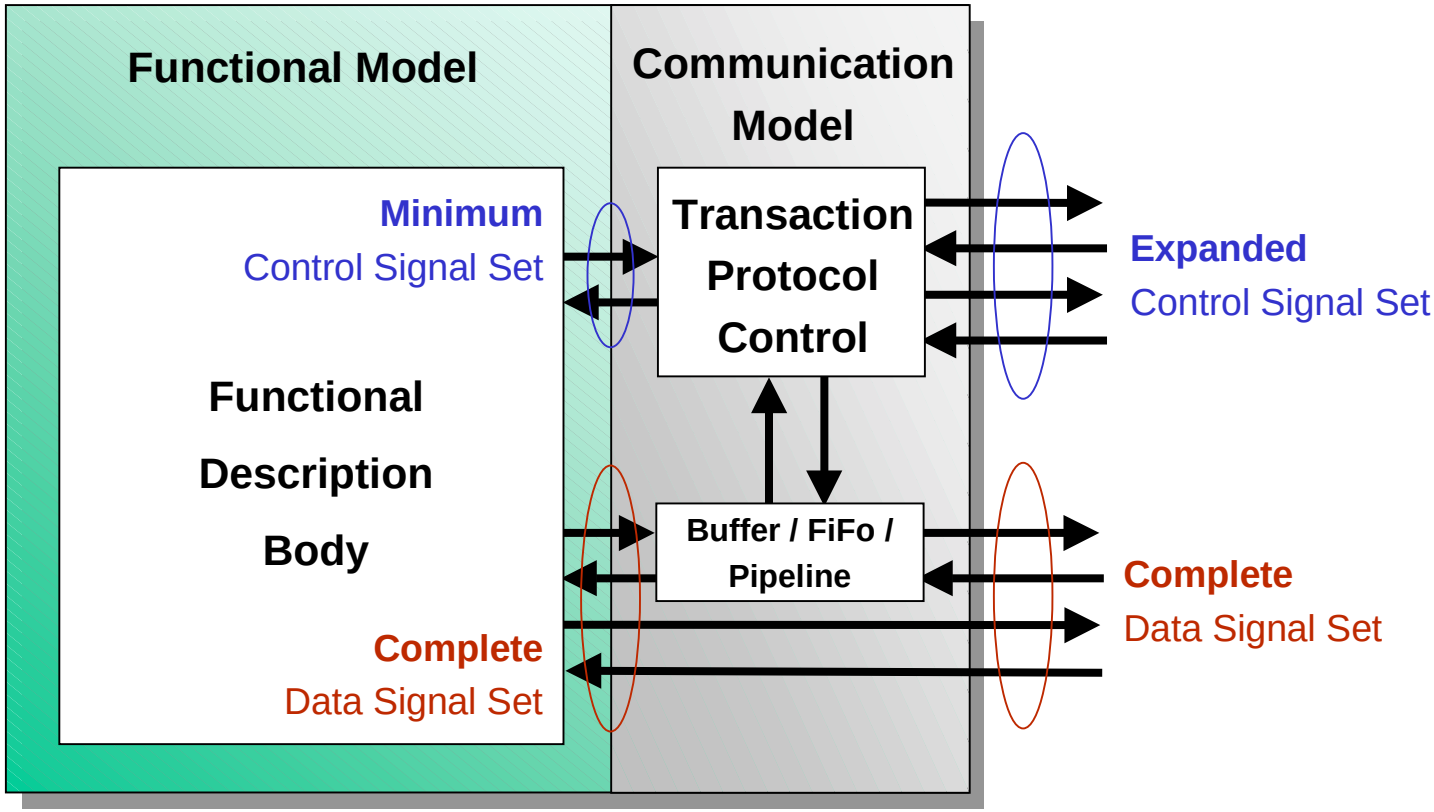
categories:

- data accuracy
  - floating-point / fixed-point, integer / bit-true
- timing accuracy
  - untimed / statistical delay / clock cycle true
- communication accuracy
  - transaction-based / packet-based / cell-based
  - data flow / control flow
  - dedicated point-to-point / bus connection point-to-multipoint
  - blocking / non-blocking
  - data priority / control priority
  - non-buffered / buffer / fifo / pipeline

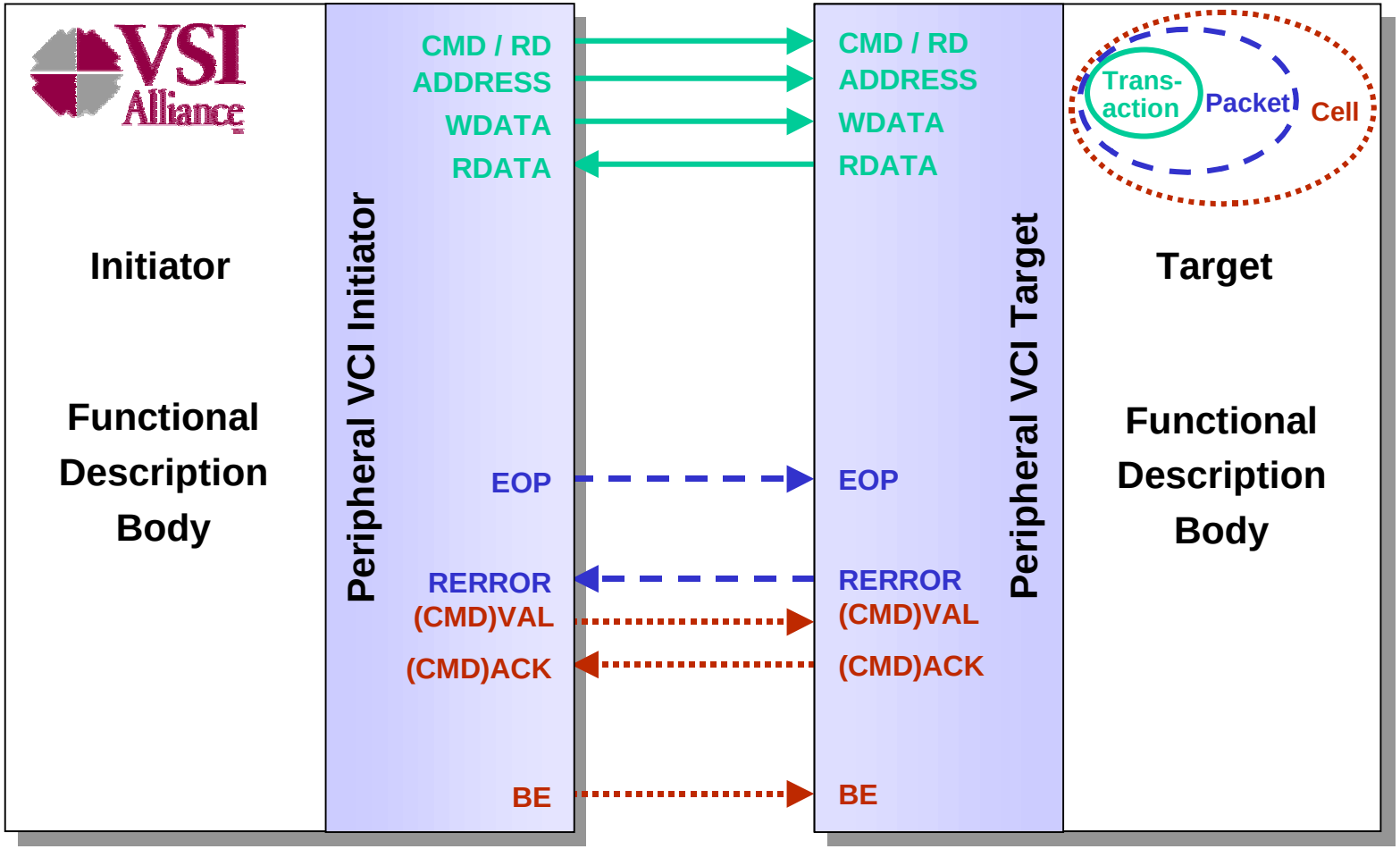
System-level Model Classification II



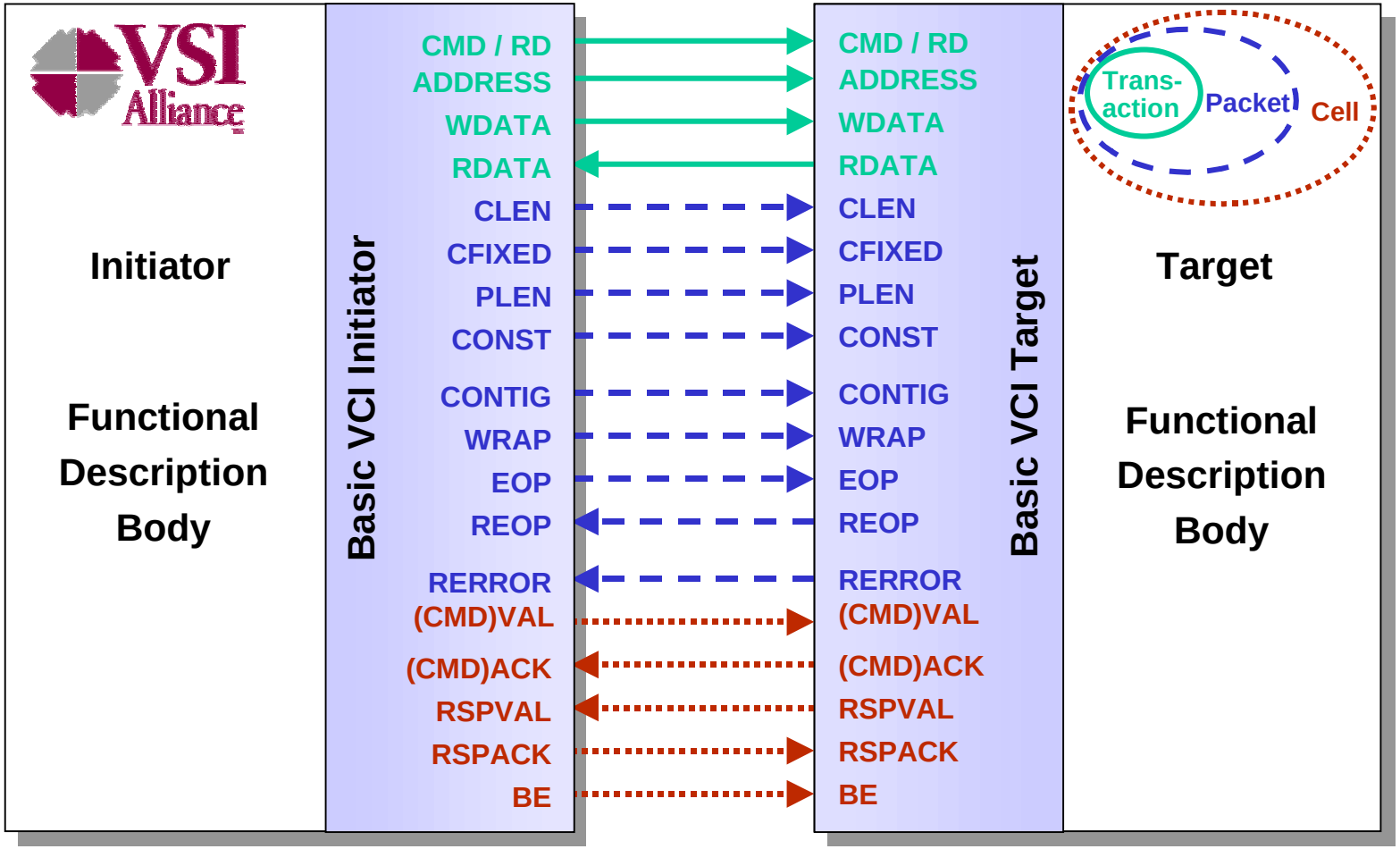
System-level Model Classification III



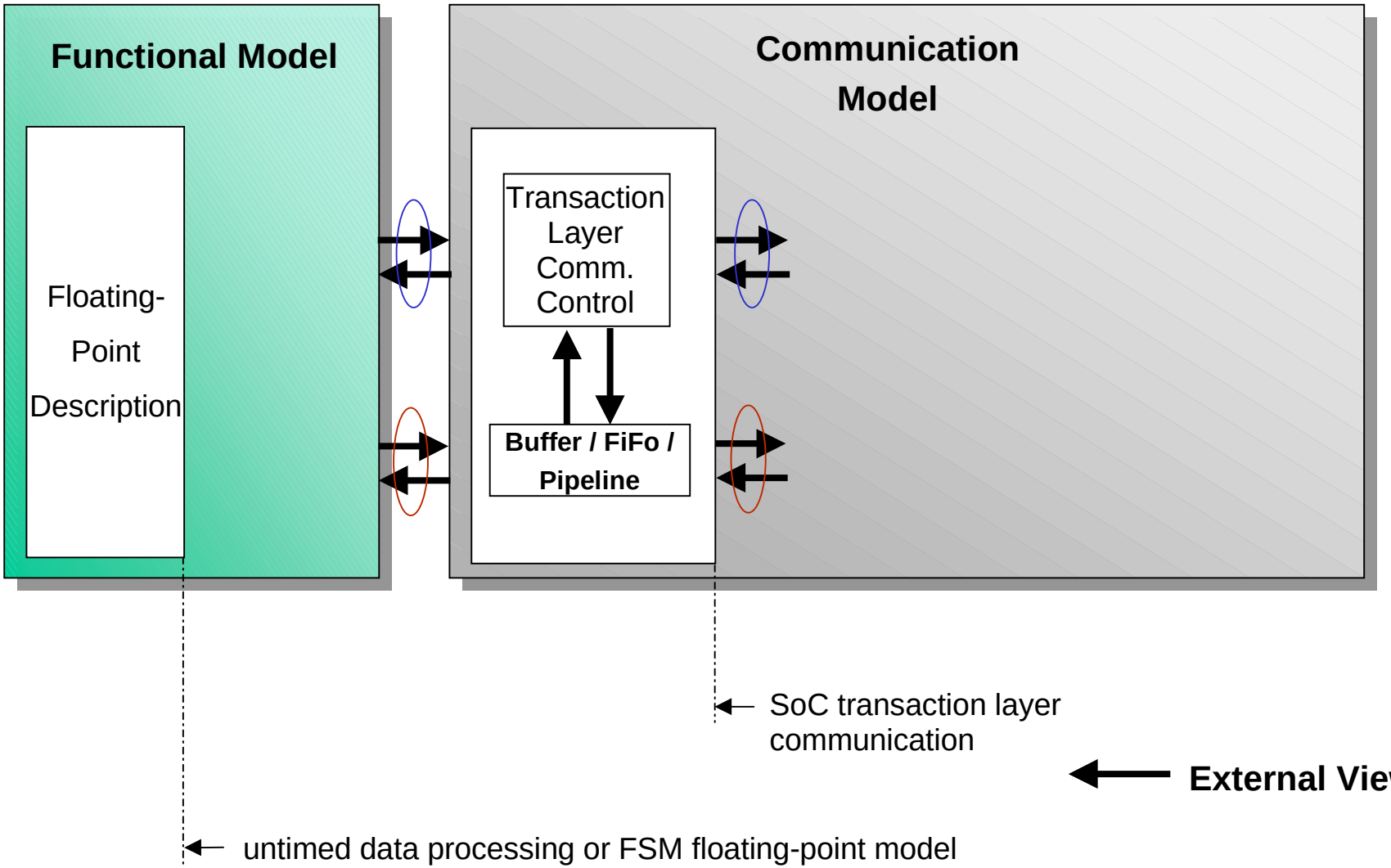
Peripheral VCI at Different Communicat. Levels



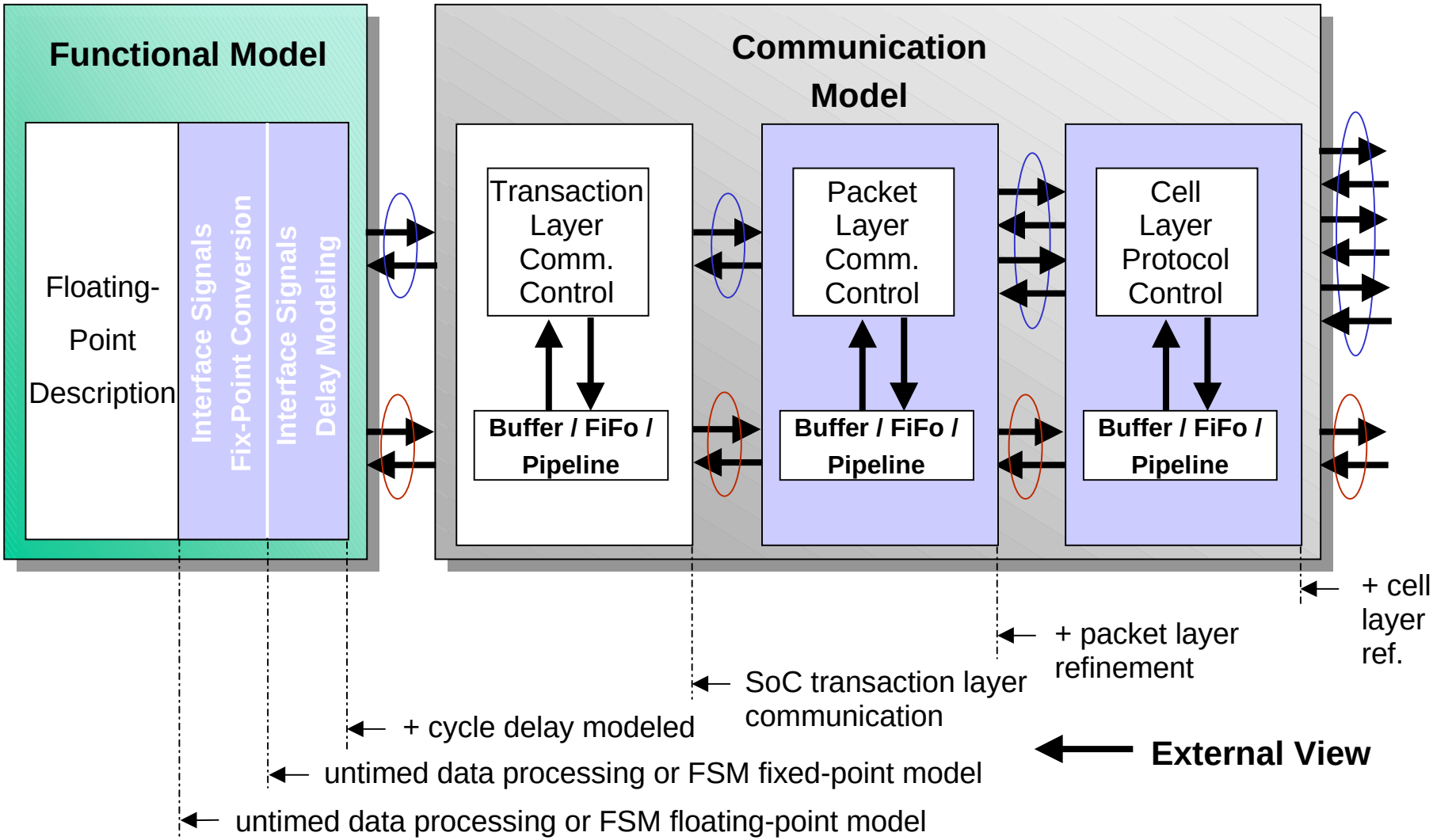
Basic VCI at Different Communication Levels



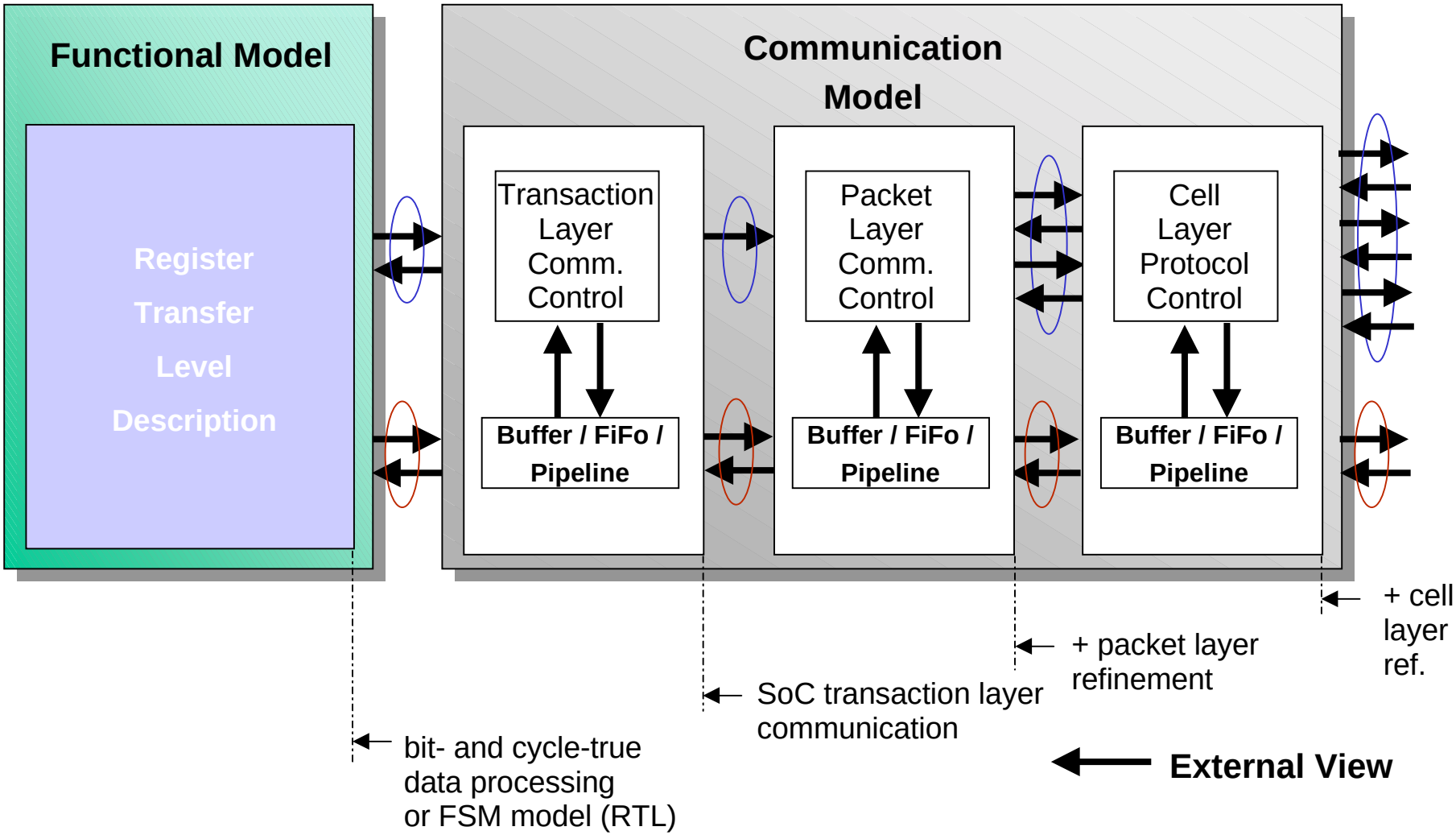
Multi-level Interface Wrapping Example I



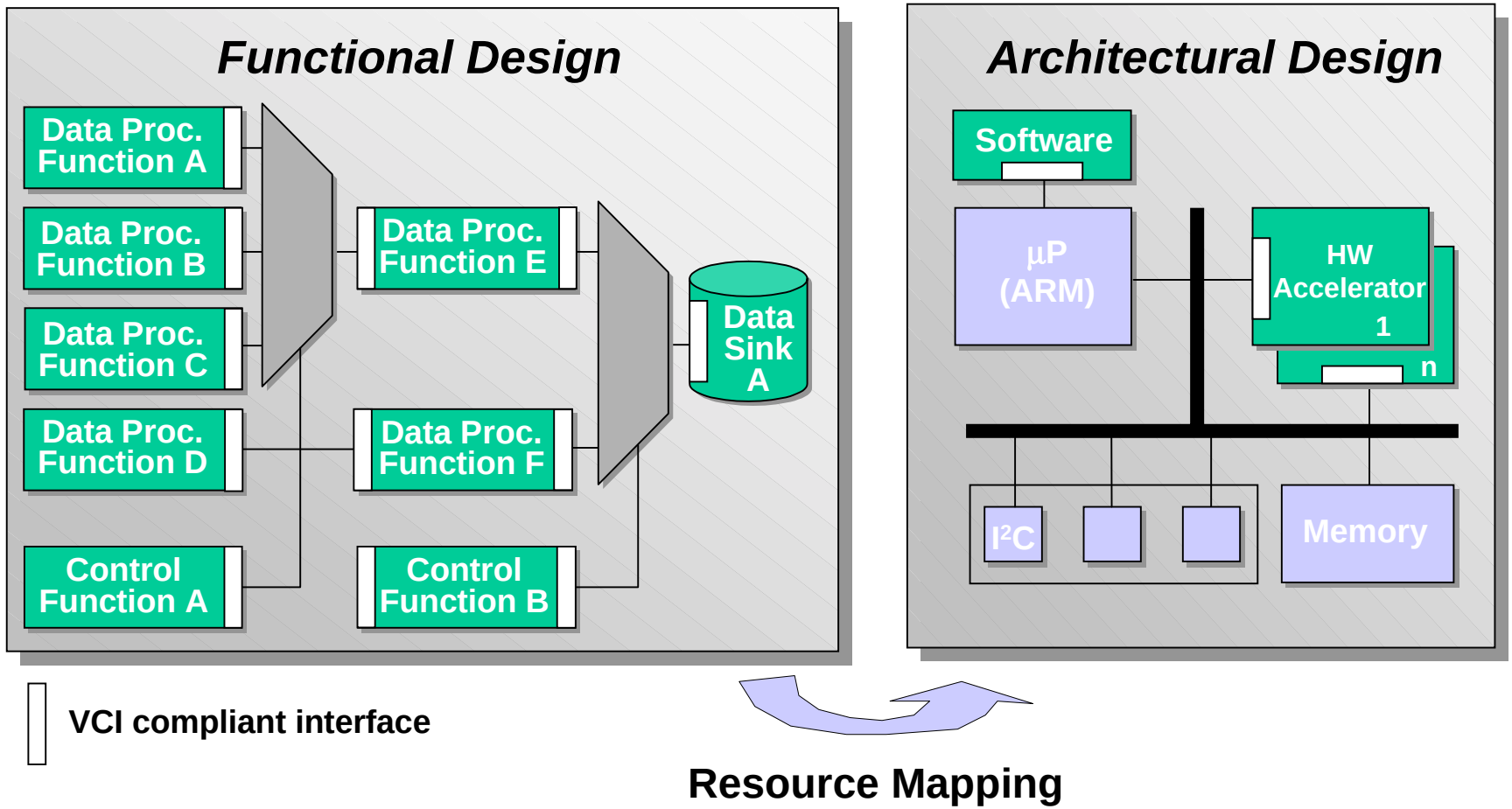
Multi-level Interface Wrapping Example II



Multi-level Interface Wrapping Example III



System-on-Chip Example



Portability Securing I

open model interface (IEEE Std 1499-1998) features

- standard IF for model-to-simulator link
- mixed-language model simulation capability
  - VHDL, Verilog, ANSI-C
  - advantage over PLI, Verilog-only standardized
- different model boundary classes and data types
  - port, parameter, viewport
  - 2-/4-state logic, booldean, integer, string, real
- simulator minimum capability definition
  - two-state boundary data class, cycle-based simulation style
- simulator stages of execution
  - bootstrap, elaboration, initialization, simulation, termination

Portability Securing II

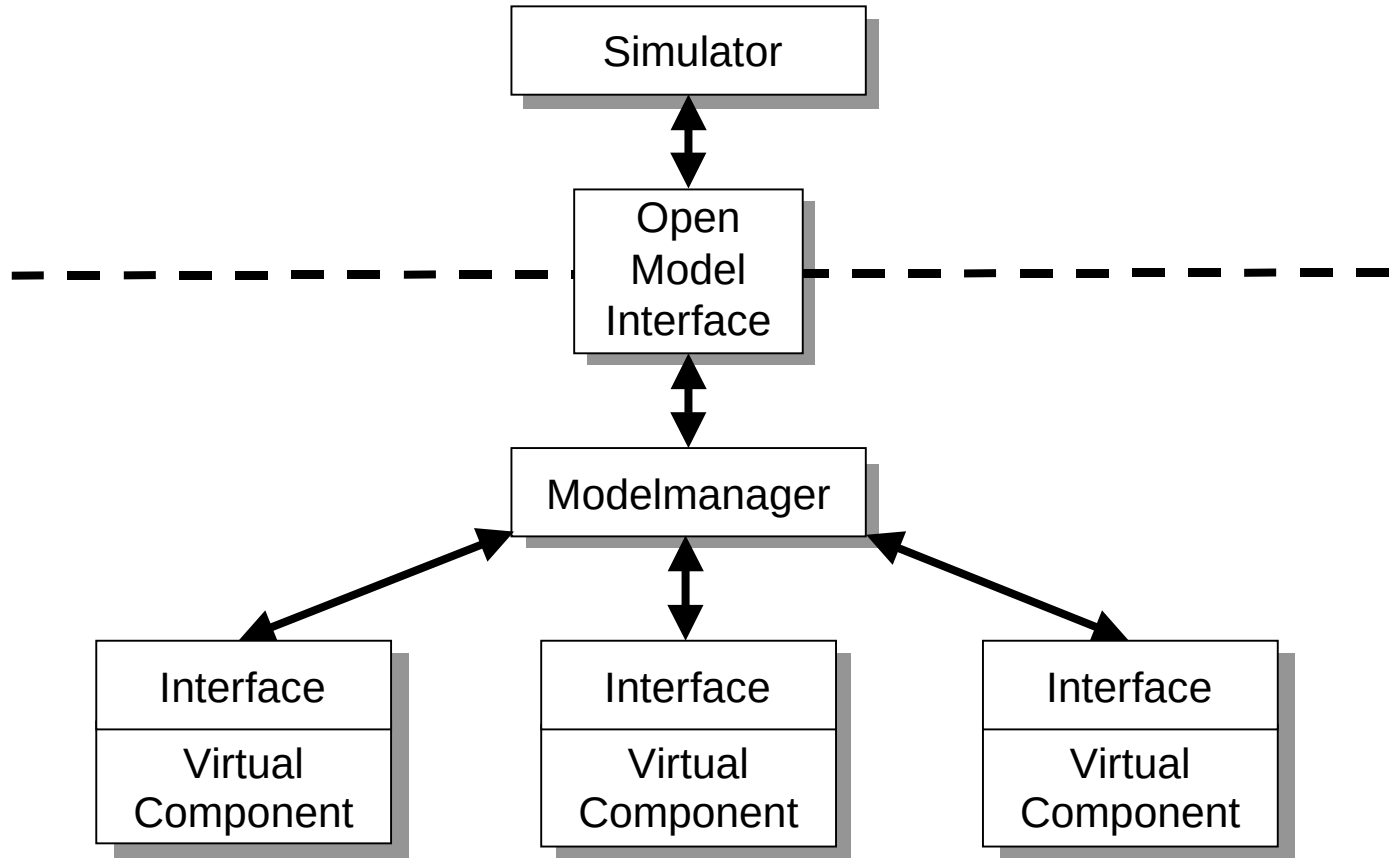


Table of Contents

Outlook

next steps and open issues

- implementation of the concept in SystemC
- check appliance for tools (SystemStudio, VCC,..)
- definition of mandatory used subset of model classes/combinations
- software appliance check
- synchronization: software function – RTOS – HDL description

Conclusion

- Motivation and Goals
  - systen level design flow, IP design challenges
- Development Framework
  - standardization activities, VSI Alliance
- Interfacing on Register Transfer Level (RTL)
  - peripheral VCI, basic VCI, uni-level interface adaptation
- Interfacing on Levels over RTL
  - investigation goals and limits
  - system-level model classification
  - peripheral and basic VCI at different communication levels
  - multi-level interface wrapping example
  - system-on-chip example
  - portability securing, open model interface

References

- [1] *Virtual component interface standard*, version 2.0 (OCB 2 2.0), by the On-chip Bus Development Working Group of the VSIA (www.vsi.org), April 2001
- [2] *System-level interface behavioral documentation standard*, version 1.0 (SLD 1 1.0), by the System-level Design Development Working Group of the VSIA (www.vsi.org), March 2000
- [3] *Model Taxonomy*, version 2.1 (SLD 2 2.1), by the System-level Design Development Working Group of the VSIA (www.vsi.org), July 2001
- [4] *IEEE standard interface for hardware description models of electronic components – Open model interface*, version IEEE-1499-1998, by IEEE (www.ieee.org), Dec. 1998
- [5] *Advanced microcontroller bus architecture specification*, by ARM Ltd. (www.arm.com), Apr. 1997
- [6] *Interface Design Guidelines*, version 1.0, by sci-worx (www.sci-worx.com), Nov. 2001
- [7] *SpeAc project homepage*, http://speac.fzi.de