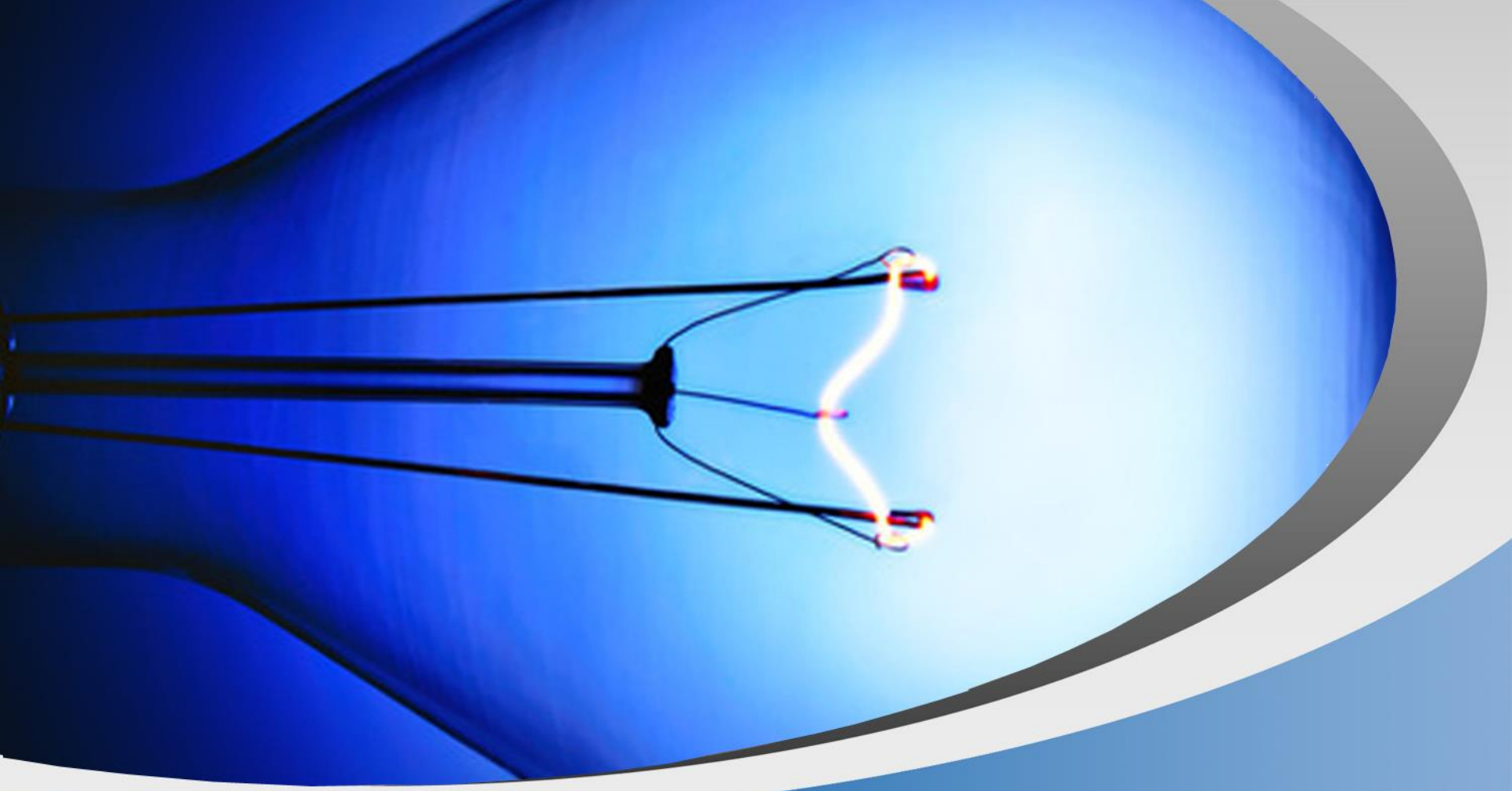


The Next Generation ESL Design Flow around the SystemC AMS Standard

Martin Barnasconi
NXP Semiconductors, SystemC AMSWG chair

accellera
SYSTEMS INITIATIVE

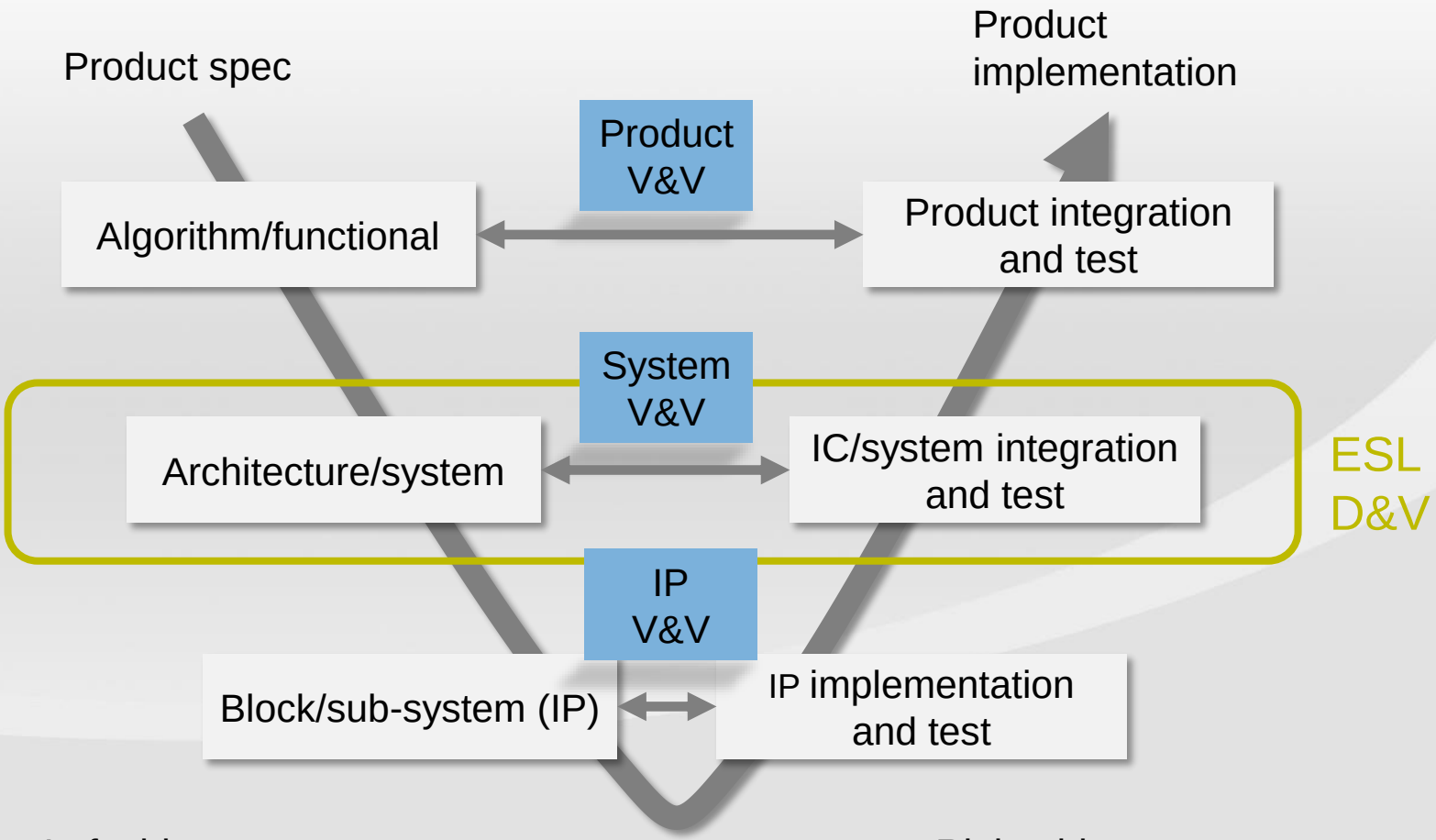
Outline

- Why do we need ESL design?
- From Traditional V-model to Concurrent Engineering
- Why SystemC AMS extensions?
- 10+ Years of SystemC AMS Standardization
- SystemC AMS Features
- Next Generation ESL Design Flow – introducing UVM
- Summary

Why do we need ESL design?

- **Growing complexity of mixed-signal ICs and systems**
  - Integrated microcontrollers (and thus firmware)
  - Interaction with analog environment (e.g. sensors, wireless communication)
  - Smarter system partitioning and analog-digital interaction to comply with low power and die size constraints
- **Compliancy to functional safety and security standards**
  - Introduction of fault injection methods to verify system robustness (ISO26262)
  - Avoid design and specification vulnerabilities (ITU-T security standards)
- **More emphasis on electronic system-level (ESL) design and verification**
  - Design: executable specification of the system (reference model)
  - Verification: earlier and more robust testing of functionality against the specification using the reference model

Traditional Design Flow – V model

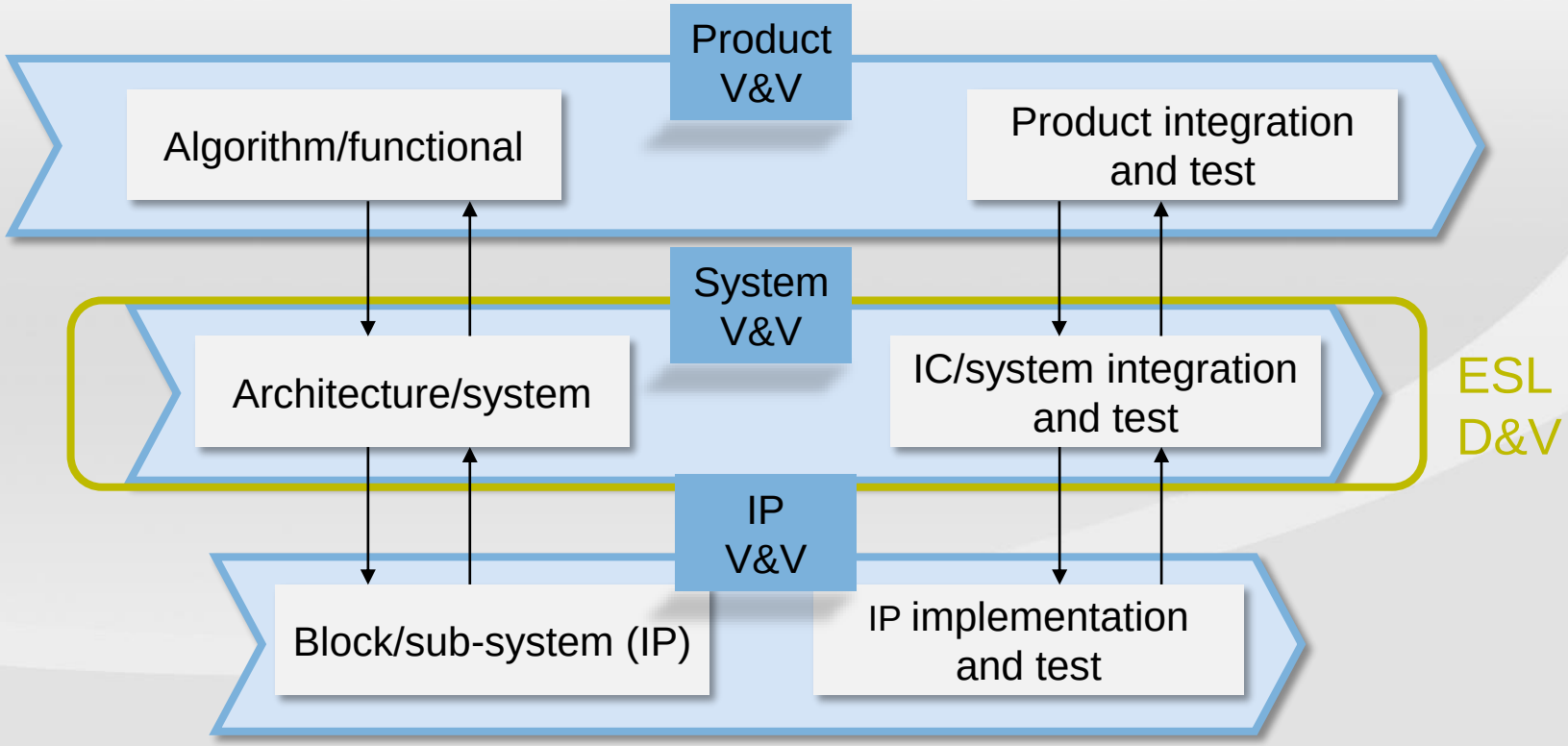


Left side:
Specification and requirements definition

Right side:
Integration & hardware prototyping

accellera SYSTEMS INITIATIVE

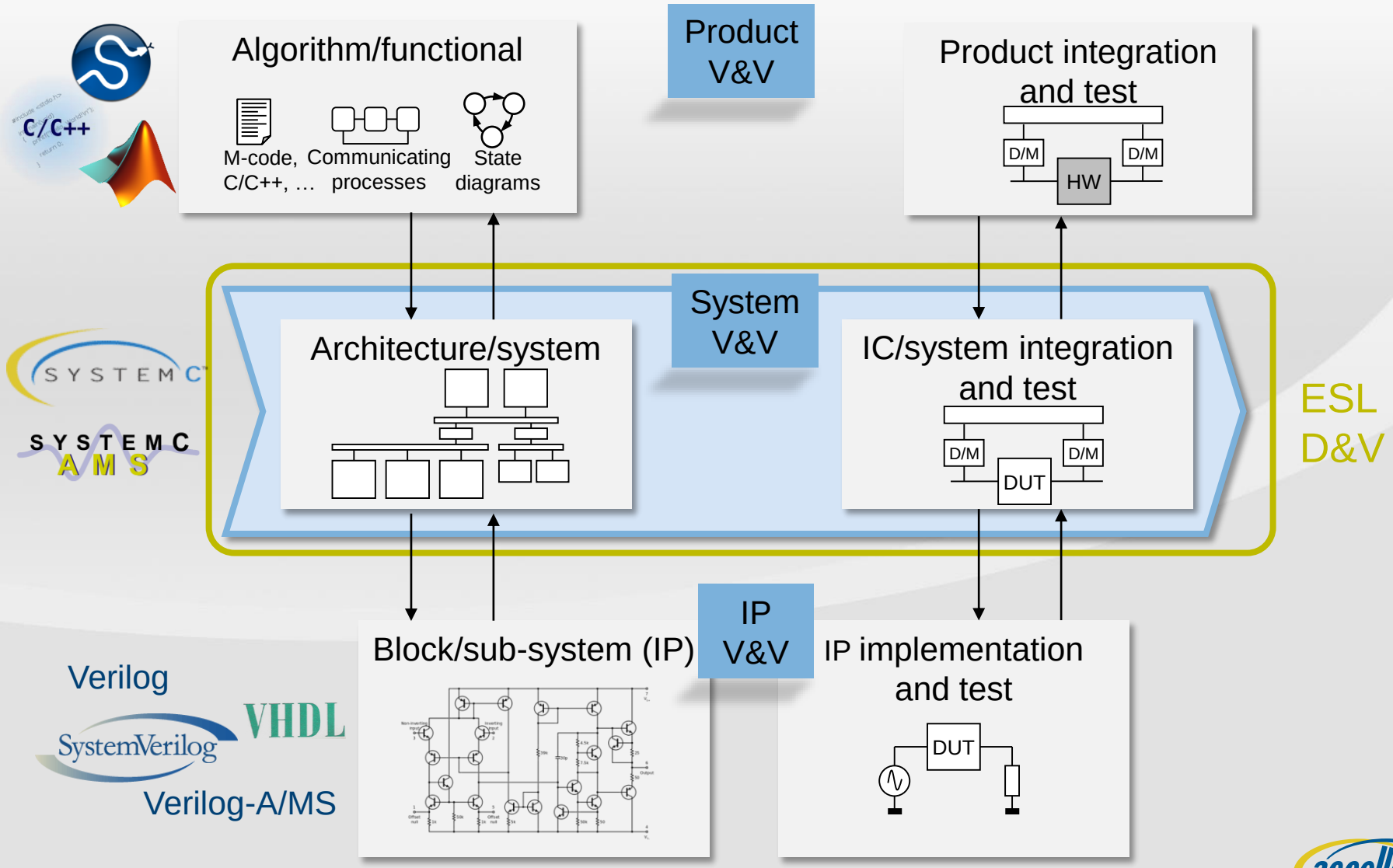
From V-model to Concurrent Engineering



Need for an **integral** system-level design and verification approach for heterogeneous systems

accellera SYSTEMS INITIATIVE

The Mixed Level & Language Challenge



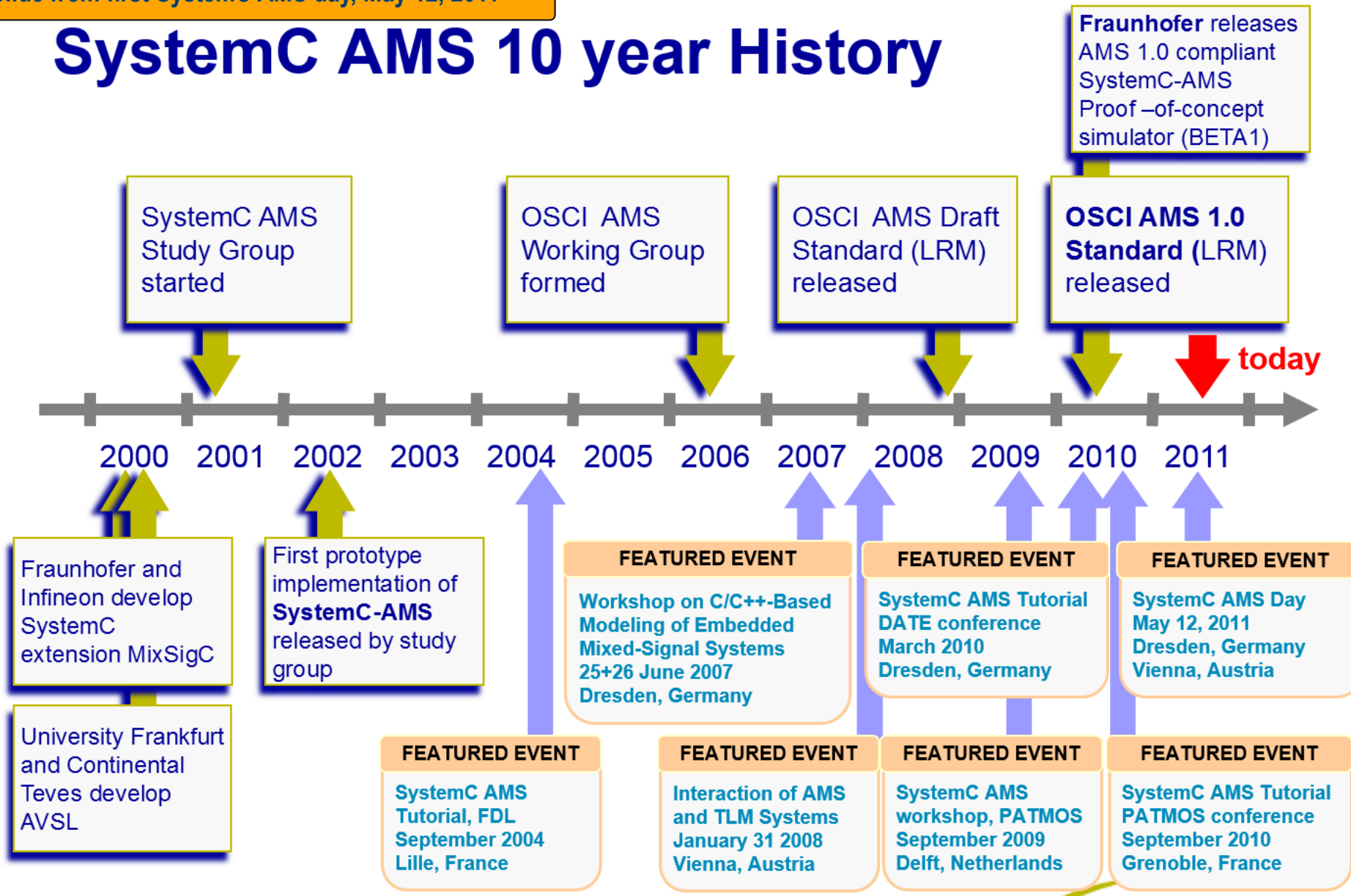
accellera SYSTEMS INITIATIVE

Why SystemC AMS extensions?

- **Unified and standardized modeling language to design and verify embedded mixed-signal architectures**
  - *Abstract AMS model descriptions* supporting a design refinement methodology, from functional/algorithm down to implementation views
  - Enabling *tool-independent exchange* and reuse of AMS models and building blocks
  - System-level language for analog *and* digital signal processing
- **Facilitate the creation of mixed-signal virtual prototypes**
  - Integration of abstract AMS/RF subsystems in combination with digital HW/SW subsystems
- **Foundation for development of AMS system-level design tools**
  - AMS language constructs and semantics defined as C++ class library built on top of IEEE Std 1666-2011 (SystemC LRM)

Slide from first SystemC AMS day, May 12, 2011

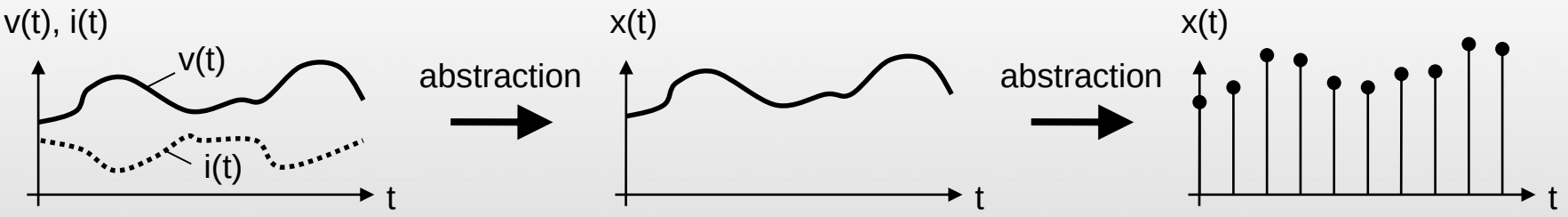
SystemC AMS 10 year History



SystemC AMS advantages

- **SystemC, thus C++ based**
  - The power of C++
  - Object oriented – modular and easy extendable
  - AMS class libraries available for basic building blocks (analog primitives)
  - Tool independent / EDA-vendor neutral
- **Modeling in multiple abstractions using one simulator**
  - No need for complex multi-kernel/co-simulation
  - No difficult APIs
  - Converter models and ports are part of the language
  - Allows abstraction along four axis
    - structure, behavior, communication and time/frequency
- **Transparent modeling platform**
  - Access to simulation kernel to ease debugging and introspection

Abstraction of analog signals



Electrical Linear Networks (ELN)

- Conservative description represented by two dependent quantities, being the voltage v(t) and the current i(t)
- Continuous in time and value
- Analog (linear) solver will resolve the *Kirchhoff's Laws*

Linear Signal Flow (LSF)

- Non-conservative description represented by single quantity x(t), to represent e.g. the voltage or current (not both)
- Continuous in time and value

Timed Data Flow (TDF)

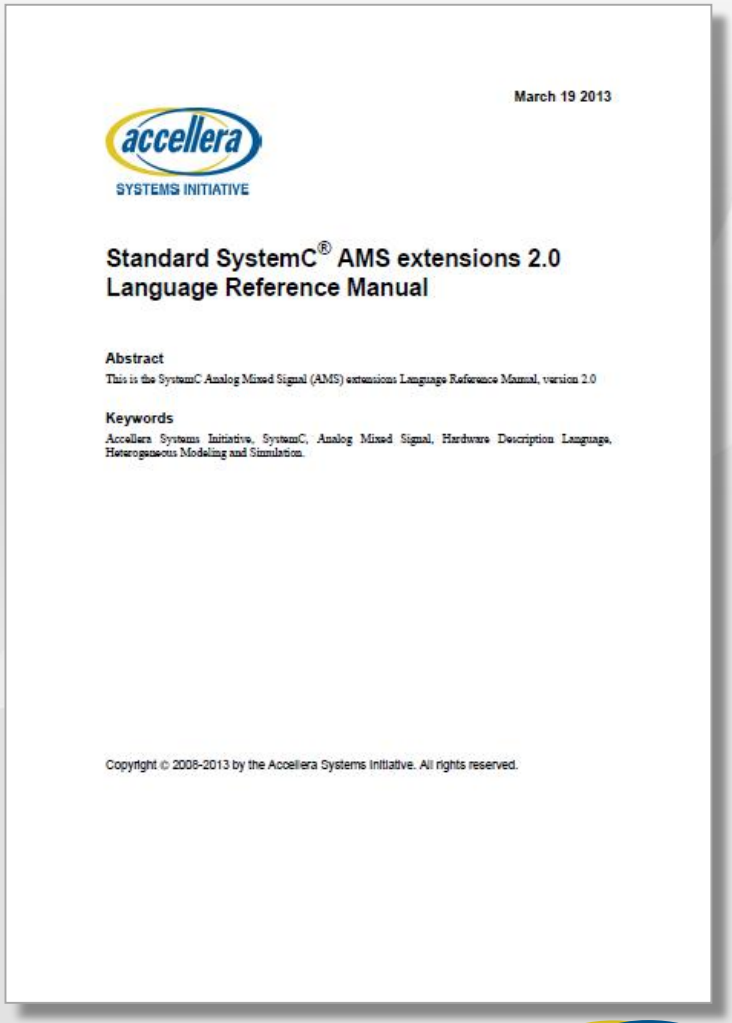
- Non-conservative description represented by single quantity x(t), to represent e.g. the voltage or current (not both)
- Discrete-time samples only, can hold any arbitrary data type

Preferred model abstraction!

accellera SYSTEMS INITIATIVE

SystemC AMS 2.0 Standard

- **SystemC AMS 2.0 Standard defined in a Language Reference Manual (LRM)**
- **Contents**
  - Terminology and conventions
  - Core language definitions
  - Predefined models of computation and analysis types
  - Utility definitions
  - Introduction to the SystemC AMS extensions (Informative)
  - Glossary (Informative)
  - Deprecated features (Informative)
  - Changes between SystemC AMS 1.0 and 2.0 standard (Informative)



March 19 2013

accellera SYSTEMS INITIATIVE

Standard SystemC® AMS extensions 2.0
Language Reference Manual

Abstract

This is the SystemC Analog Mixed Signal (AMS) extensions Language Reference Manual, version 2.0

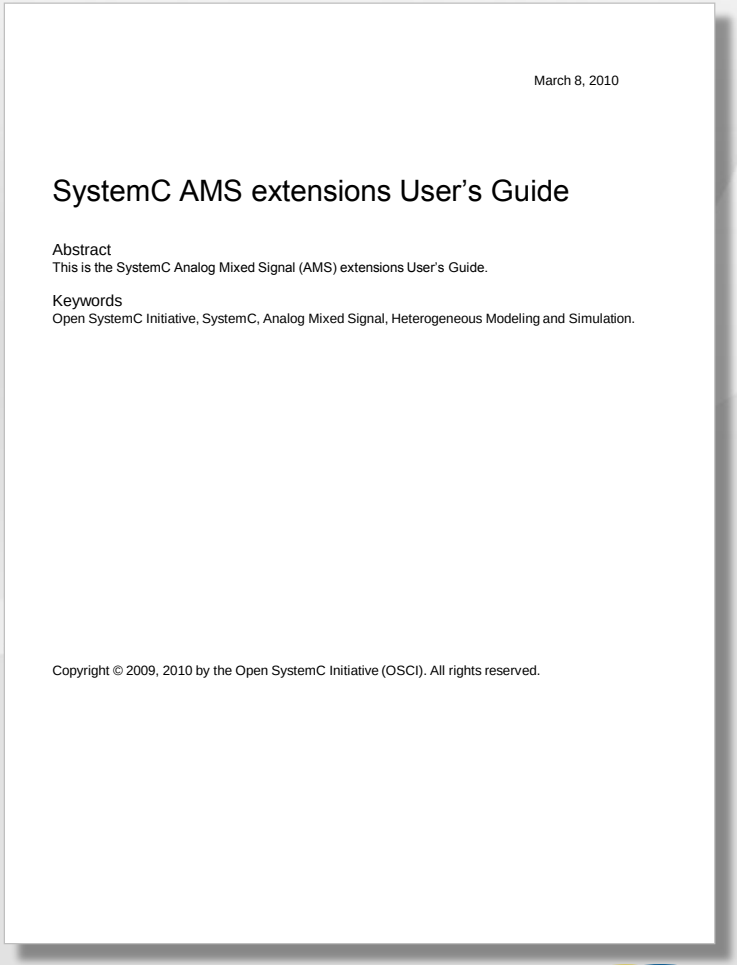
Keywords

Accellera Systems Initiative, SystemC, Analog Mixed Signal, Hardware Description Language, Heterogeneous Modeling and Simulation.

accellera SYSTEMS INITIATIVE

SystemC AMS User's Guide

- **Comprehensive guide explaining the basics of the AMS extensions**
  - TDF, LSF and ELN modeling
  - Small-signal frequency-domain modeling
  - Simulation and tracing
  - Modeling and refinement methodology
  - Many code examples
- **Application examples**
  - Binary Amplitude Shift Keying (BASK)
  - Plain-Old-Telephone-System (POTS)
  - Analog filters and networks
- **Has proven its value**
  - Reference guide for many new users



March 8, 2010

SystemC AMS extensions User's Guide

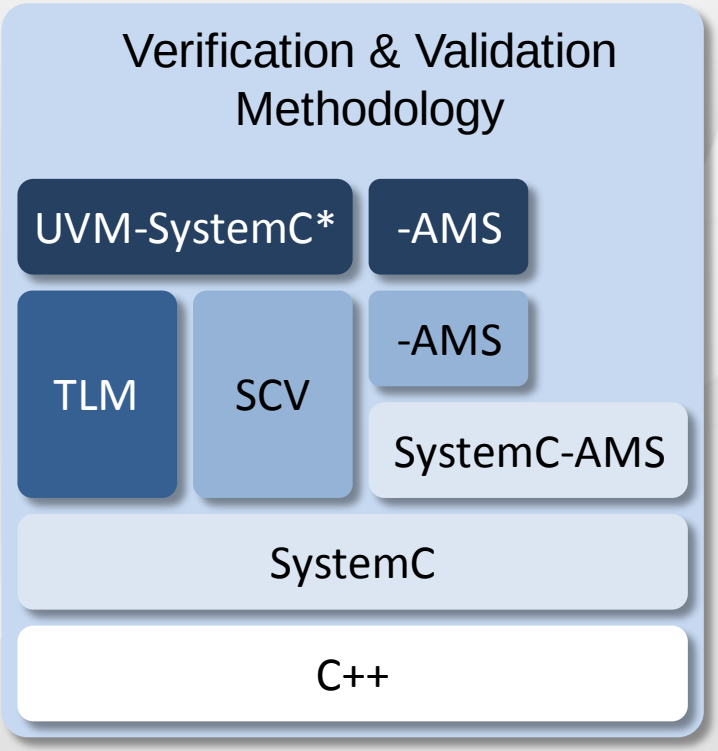
Abstract
This is the SystemC Analog Mixed Signal (AMS) extensions User's Guide.

Keywords
Open SystemC Initiative, SystemC, Analog Mixed Signal, Heterogeneous Modeling and Simulation.

accellera SYSTEMS INITIATIVE

Next Generation ESL Design Flow

- **No structured nor unified verification methodology available for ESL design**
  - The Universal Verification Methodology (UVM) in SystemVerilog is primarily targeting block/IP level (RTL) verification, not system-level
- **Porting UVM to SystemC/C++ enables**
  - creation of more advanced system-level test benches
  - reuse of verification components between system-level and block-level verification
- **Target is to make UVM truly *universal*, and not tied to a particular language**
  - Standardization in Accellera has started

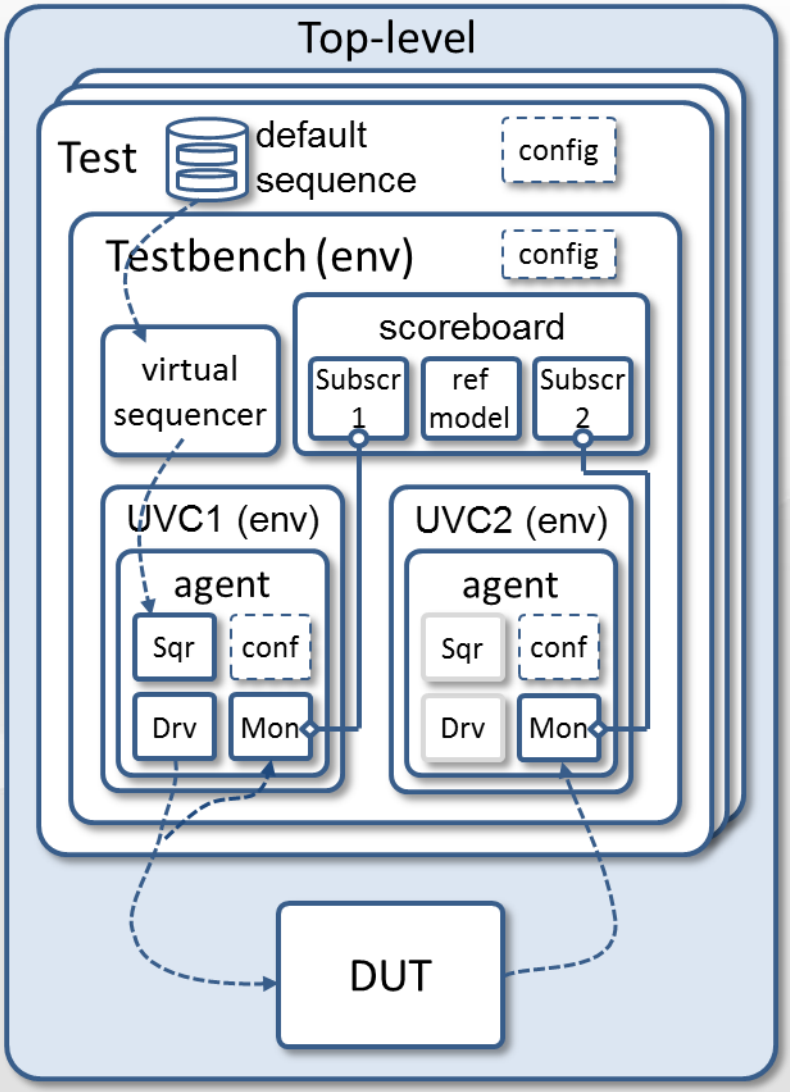


*UVM-SystemC = UVM implemented in SystemC/C++

accellera SYSTEMS INITIATIVE

Universal Verification Methodology

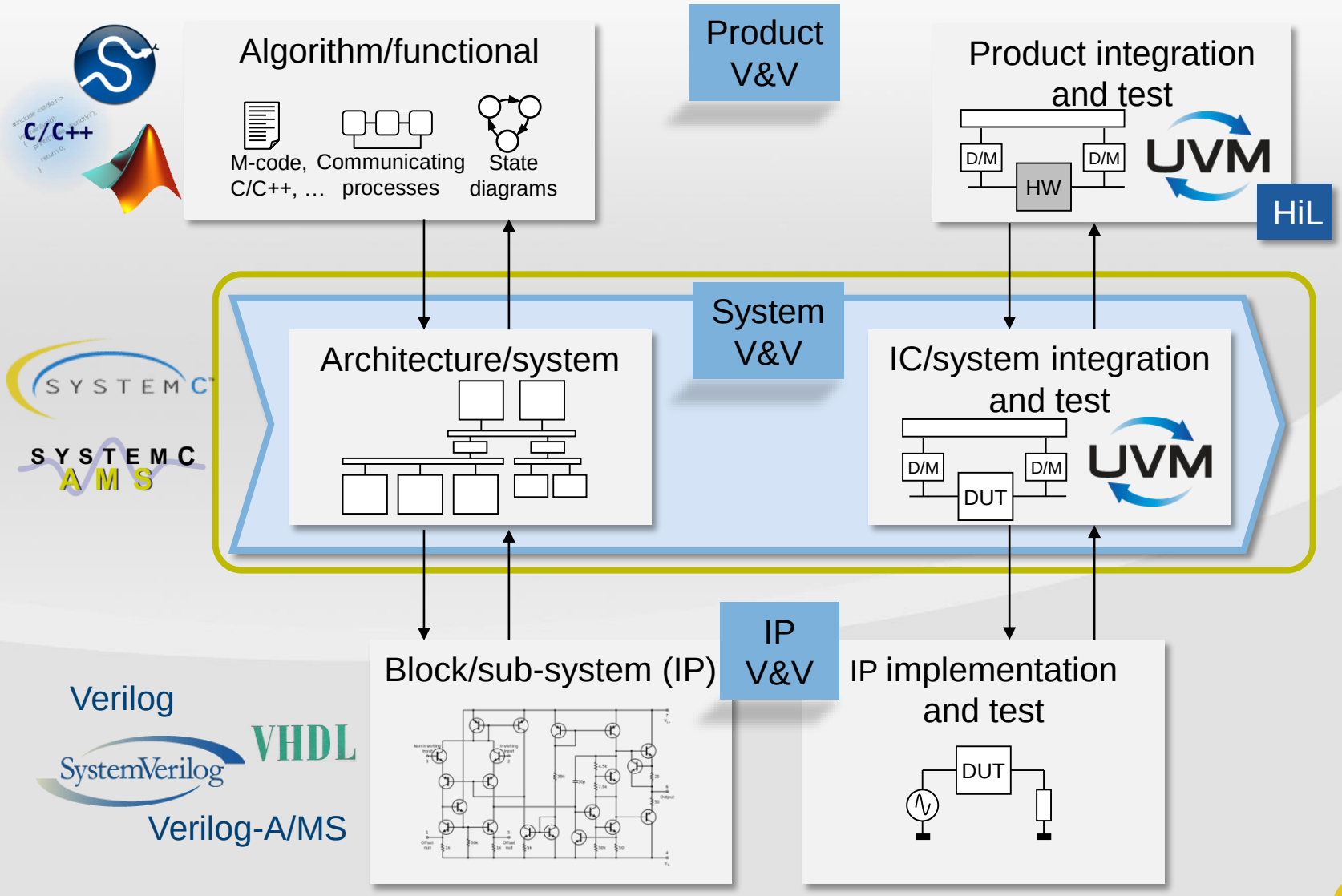
- **Accellera standard to create modular, scalable, configurable and reusable test benches**
  - Based on verification components with standardized interfaces
- **Class library which provides a set of built-in features dedicated to verification**
  - Utilities for phasing, component overriding (factory), configuration, comparing, scoreboarding, reporting, etc.
- **Coverage Driven Verification (CDV)**
  - Introducing automated stimulus generation, independent result checking and coverage collection



UVM in SystemC and SystemC-AMS…

- **Brings a system-level verification methodology for embedded systems which include HW/SW and AMS functions**
  - SystemC is the recognized standard for system-level design, and needs to be extended with advanced verification concepts
  - SystemC AMS available to cover the AMS verification needs
- **Enables reuse of tests and test benches across verification (simulation) and validation (HW-prototyping) platforms**
  - This requires a portable language like C++ to run tests on HW prototypes and even measurement equipment
  - Enabling Hardware-in-the-Loop simulation or Rapid Control Prototyping
- **Will be based on standards and open source reference implementations**
  - Leverage from existing methodology standards and reference implementations, aligned with best practices in verification

ESL Design and Verification Flow



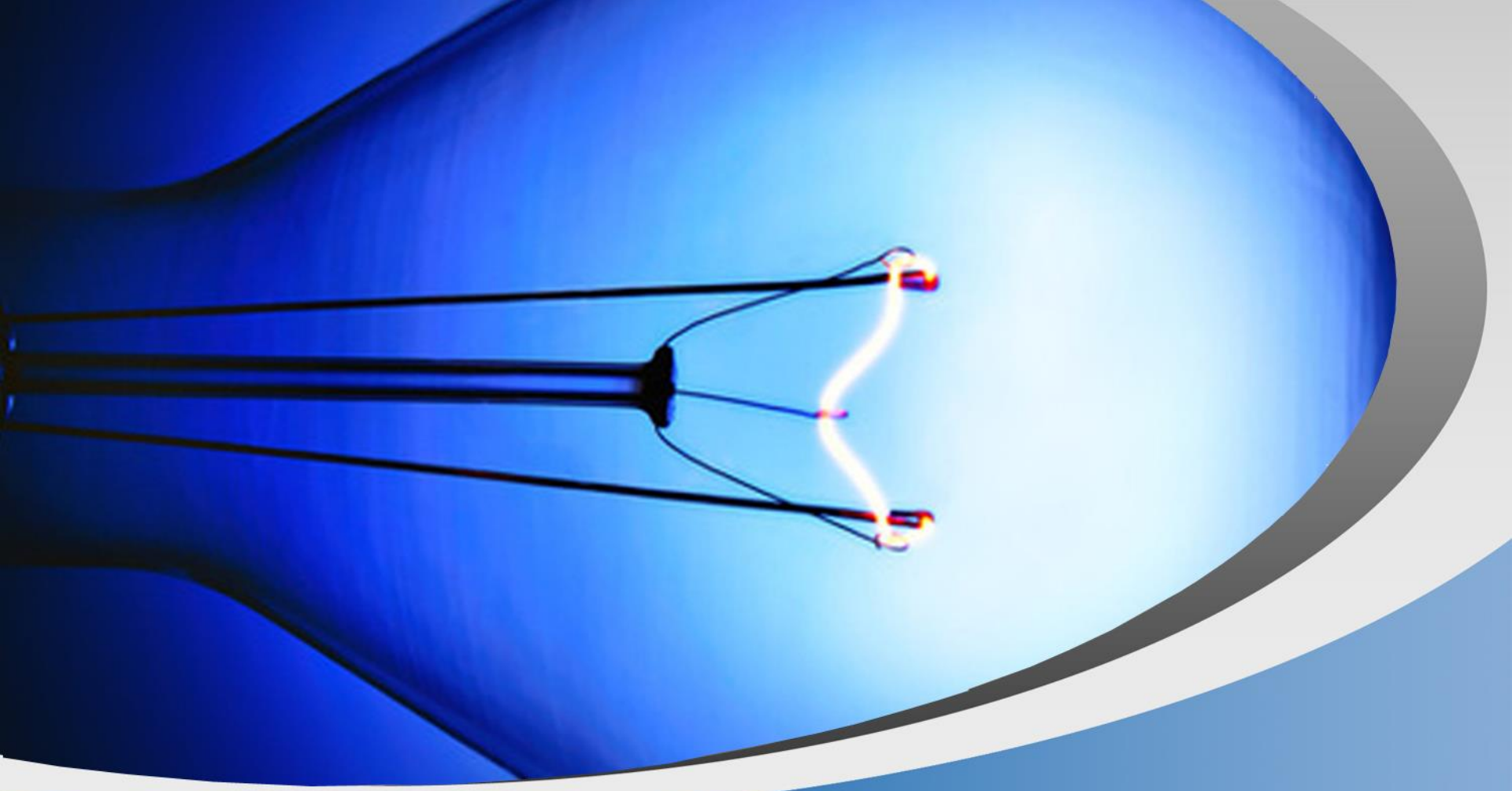
accellera SYSTEMS INITIATIVE

Summary

- **Today's complex mixed-signal systems require an advanced ESL design and verification flow**
  - Design: Executable specification of the system (reference model)
  - Verification: Earlier and more robust testing of functionality against the specification using the reference model
- **SystemC 2.3.1 and SystemC AMS 2.0 are mature standards**
  - Enable system-level modeling of mixed AMS/RF and digital HW/SW systems at different levels of abstraction
- **The next generation ESL design flow will bring UVM in SystemC**
  - Universal Verification Methodology encourages reuse of tests and test benches across verification (simulation) and validation (HW-prototyping) platforms

More information

- **SystemC AMS 2.0 standard and community pages**
  - www.accellera.org/downloads/standards/systemc
  - www.systemc-ams.org
- **SystemC AMS forum**
  - forums.accellera.org/index.php?/forum/13-systemc-ams-analogmixed-signal/
- **SystemC-AMS 1.0 and 2.0 proof-of-concept library**
  - www.eas.iis.fraunhofer.de/systemcamsdownloads
- **UVM-SystemC**
  - www.verdi-fp7.eu/



Thank you

accellera
SYSTEMS INITIATIVE