



Unifying ESL and RTL Design and Verification

Grant Martin

Fellow, Cadence Berkeley Labs

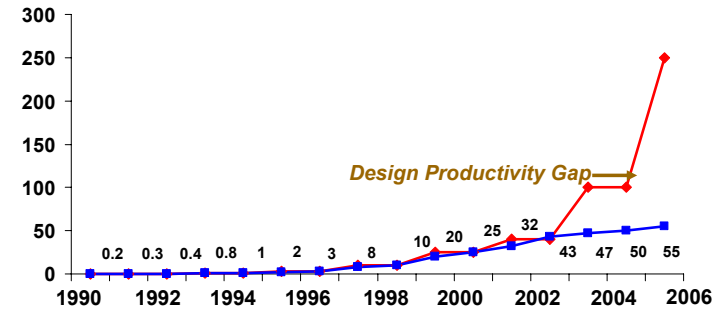
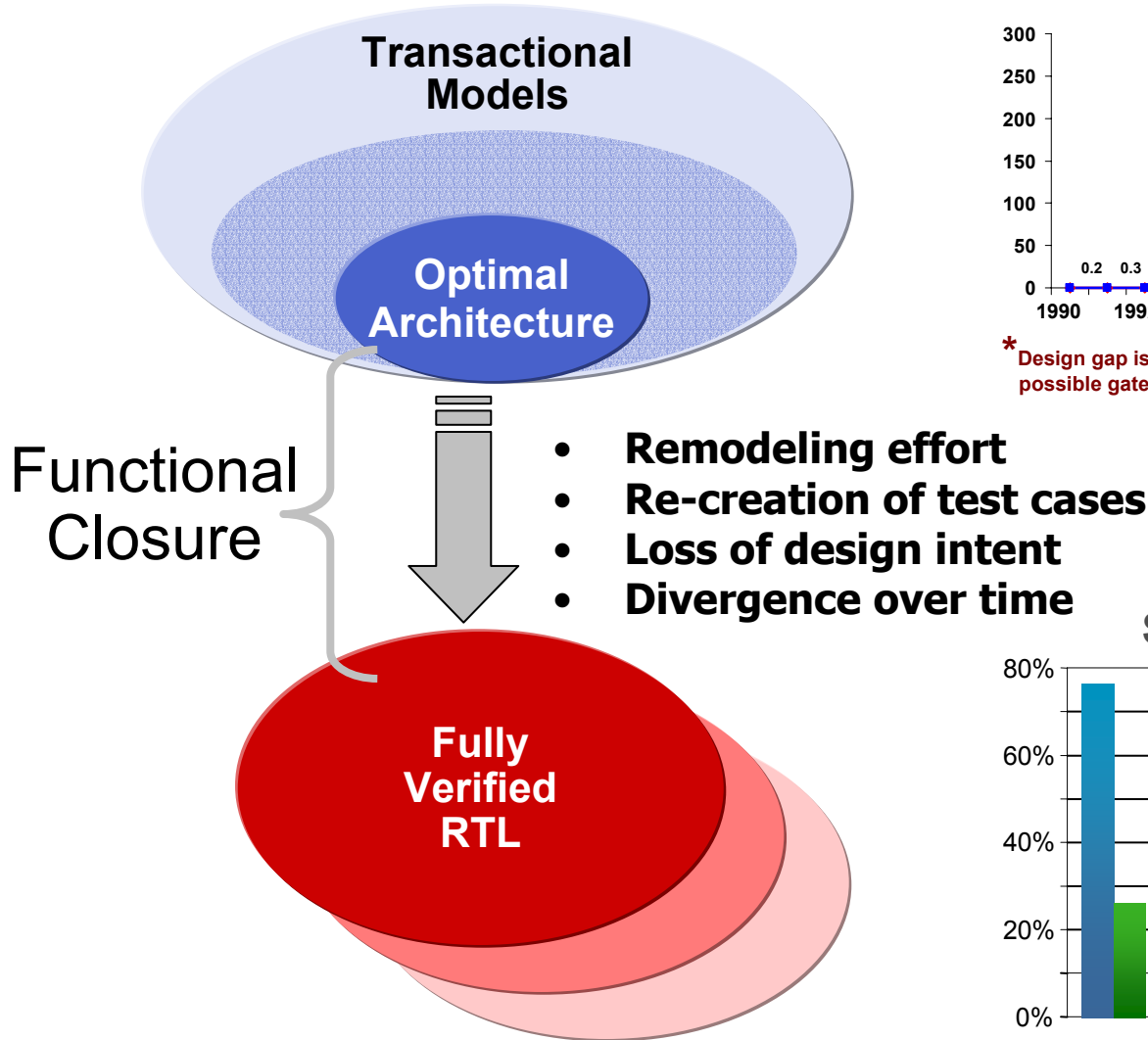
CoWare System Level Design Symposium

Lugano, Switzerland: 3 November 2003: 14:15-15:00

The Vision of the Alliance

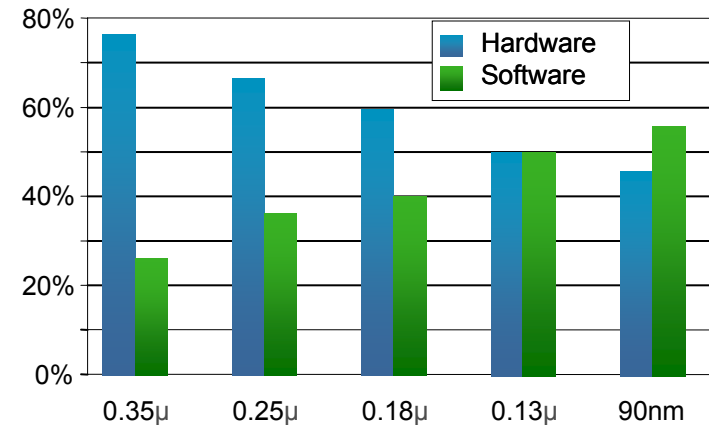
- Cadence and CoWare announced, on 8 September 2003, a new alliance that will provide a complete solution for system-level design and RTL implementation of nanometer-scale chips.
- Vision: To unify the design and verification of hardware and embedded software by
 - Leveraging SystemC
 - Integrating customer proven technology from CoWare's ConvergenSC system-level design product family and the Cadence Incisive verification platform, and
 - Optimising development of IP models for the combined solution.

The Problem: Functional Modelling Closure is needed to unify ESL and RTL Design and Verification

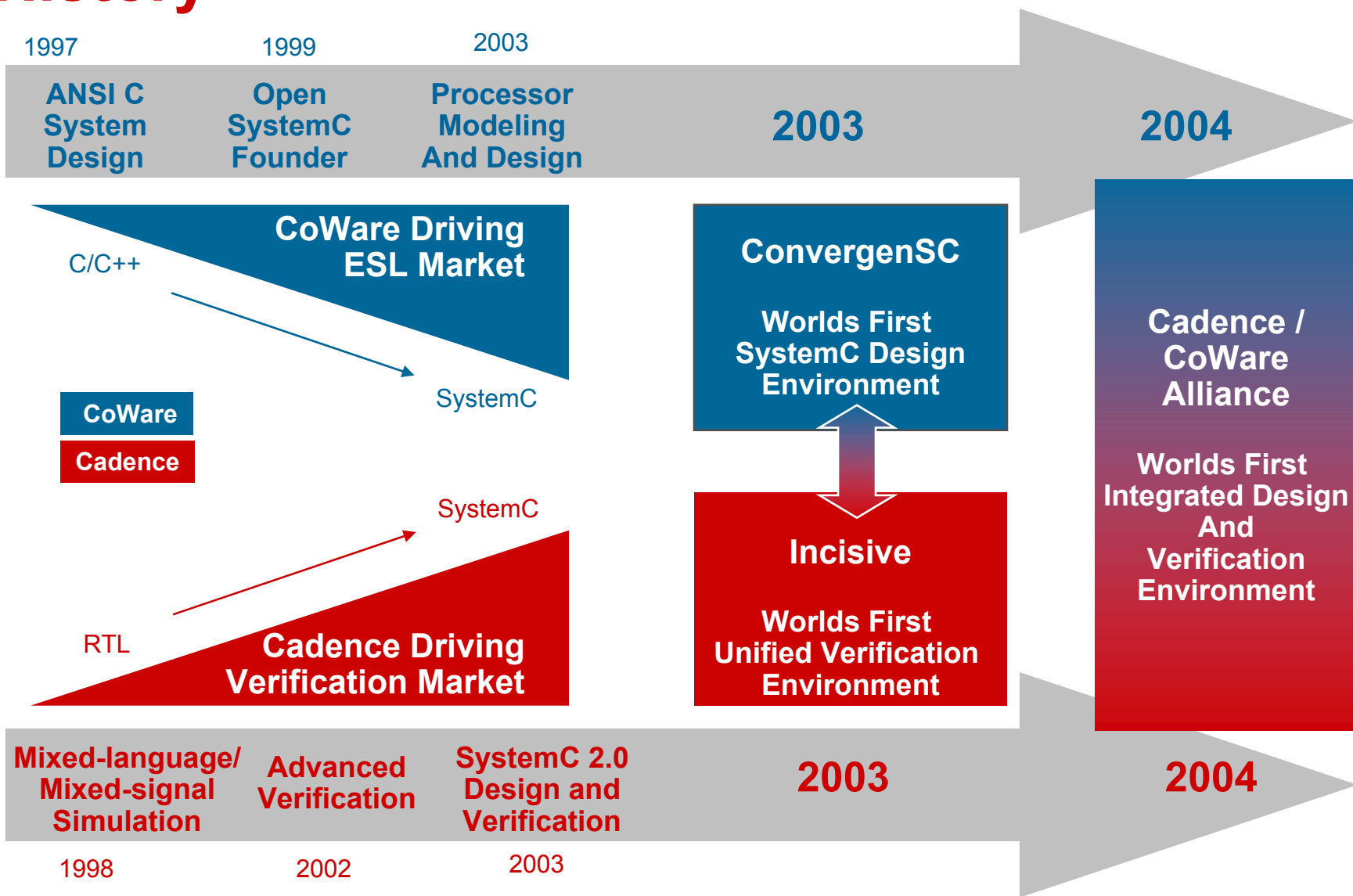


* Design gap is the difference between available gates and possible gates per silicon process.

SoC HW and SW Cost



History

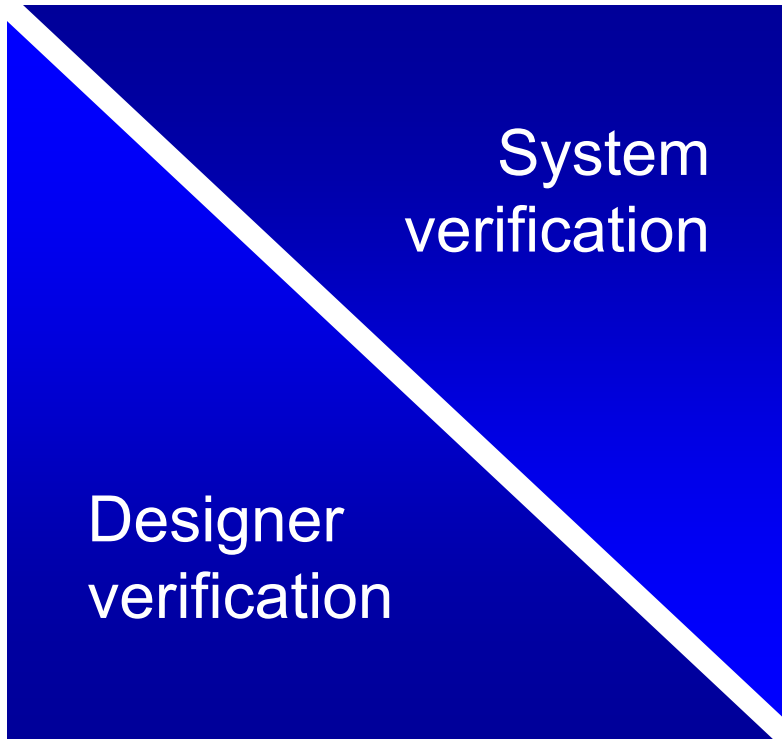


Cadence and CoWare are uniquely positioned to unify ESL and RTL design and verification

- CoWare ConvergenSC
 - Transaction-level IP development: processor, bus, and peripherals
 - SoC architectural exploration and performance analysis
 - Platform design, configuration, and optimization
 - Early HW/SW verification with fast, accurate SystemC transactional prototypes
- Cadence Incisive
 - Unified verification methodology and verification IP
 - HDL analysis with static and dynamic assertions
 - Testbench generation
 - Mixed language simulation with Acceleration-on-Demand
 - Comprehensive debugging and comprehensive coverage
- Driving enabling technologies into open standards (SystemC, IEEE Verilog, PSL/Sugar, SPIRIT, etc.)

Cadence Language Strategy

Support all standards for all designs



- Verilog & VHDL
- SystemC & SCV
- SystemVerilog
- PSL, OVL, & SVA
- Verilog/VHDL-AMS

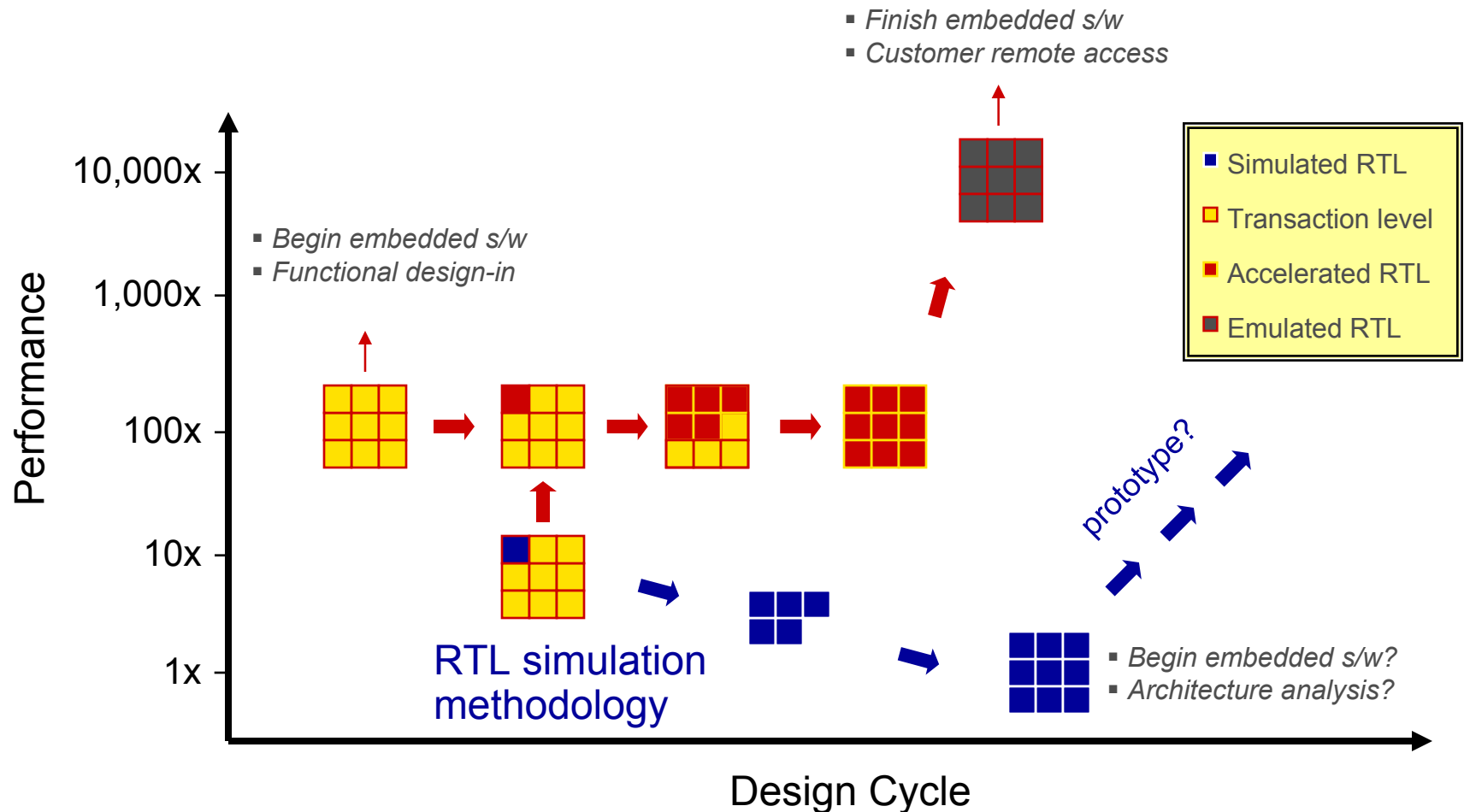
Incisive

Encounter

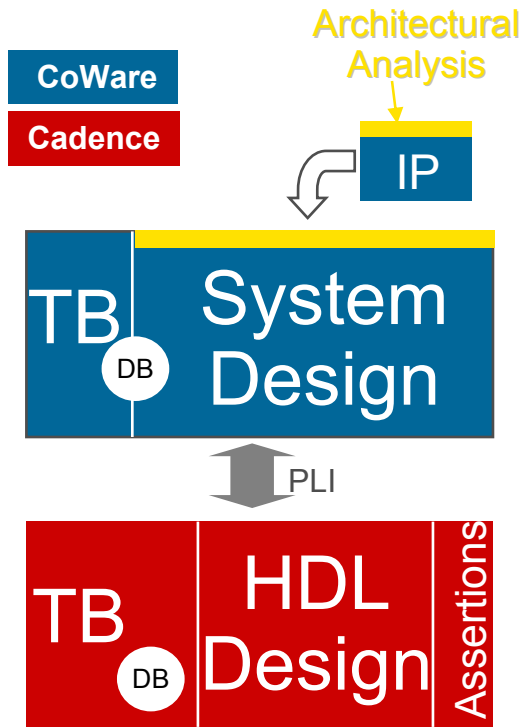
Virtuoso

Incisive single-kernel architecture supports all languages natively

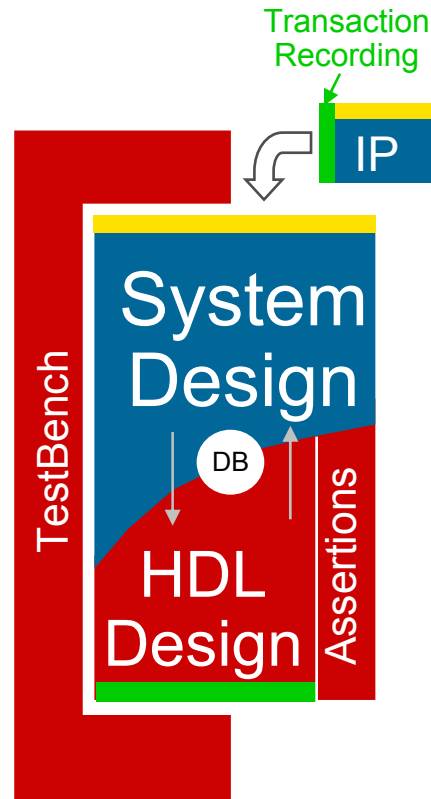
Incisive delivers 100x Full-Chip Performance throughout the verification flow



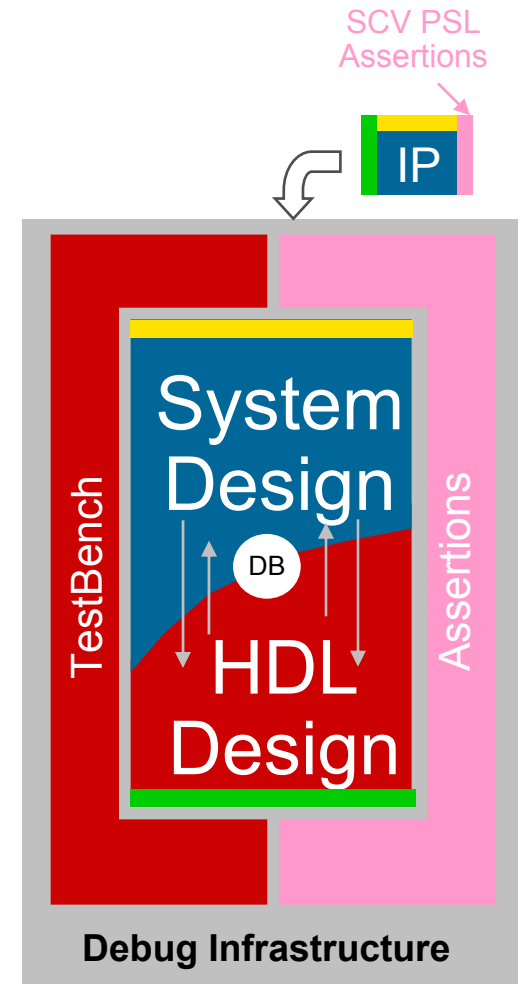
Objectives: Unified Simulation



Pre-Alliance
August 2003

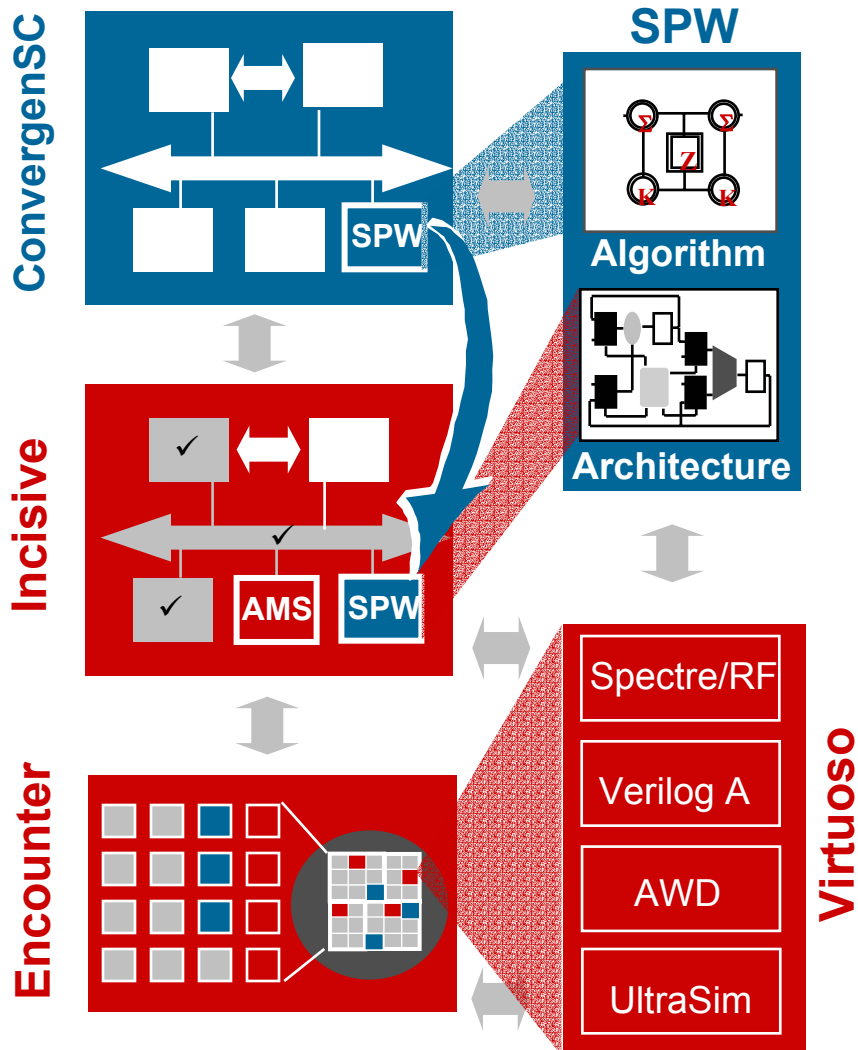


Post-Alliance
2004



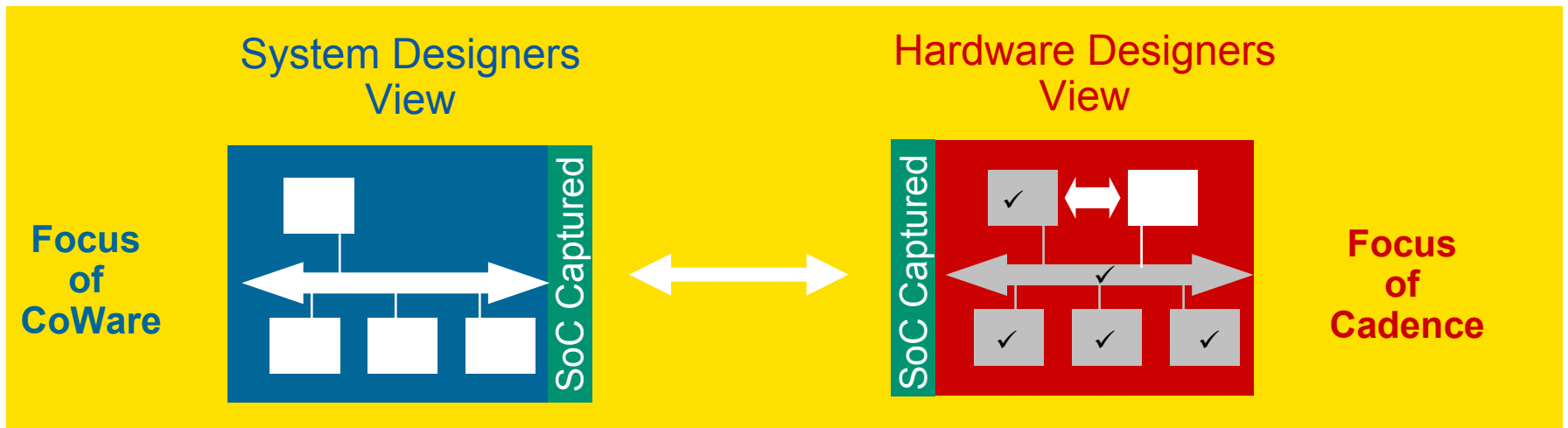
Future
2004+

Objectives: Unified Algorithmic Environment



- SPW has been and will continue to be the heart of the signal processing solution...tightly integrated throughout the flow CDN/CW flow
- Incisive Links
 - Transactional models simulates directly as part of the FVP
 - Co-simulation between SPW and Incisive running HDL
- Virtuoso Links
 - SPW AMS co-simulation (Verilog-A/AMS model, spectre netlist ...)
 - Spectre extracted model runs natively within SPW
 - Testbench integration for AMS Designer
- Encounter Links
 - Enhanced code generation for low power and datapath optimization

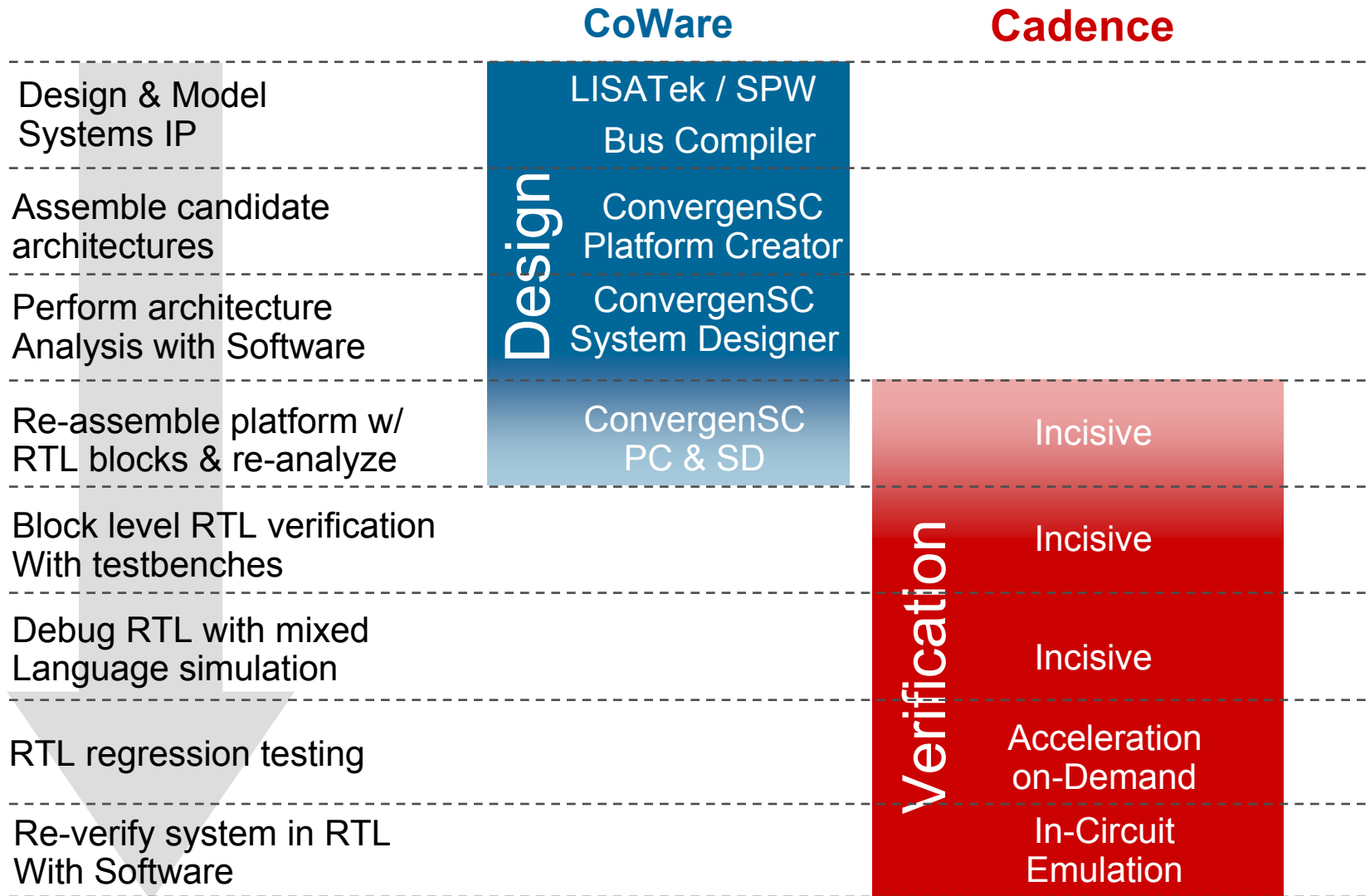
SystemC Simulation and Debug: Uses



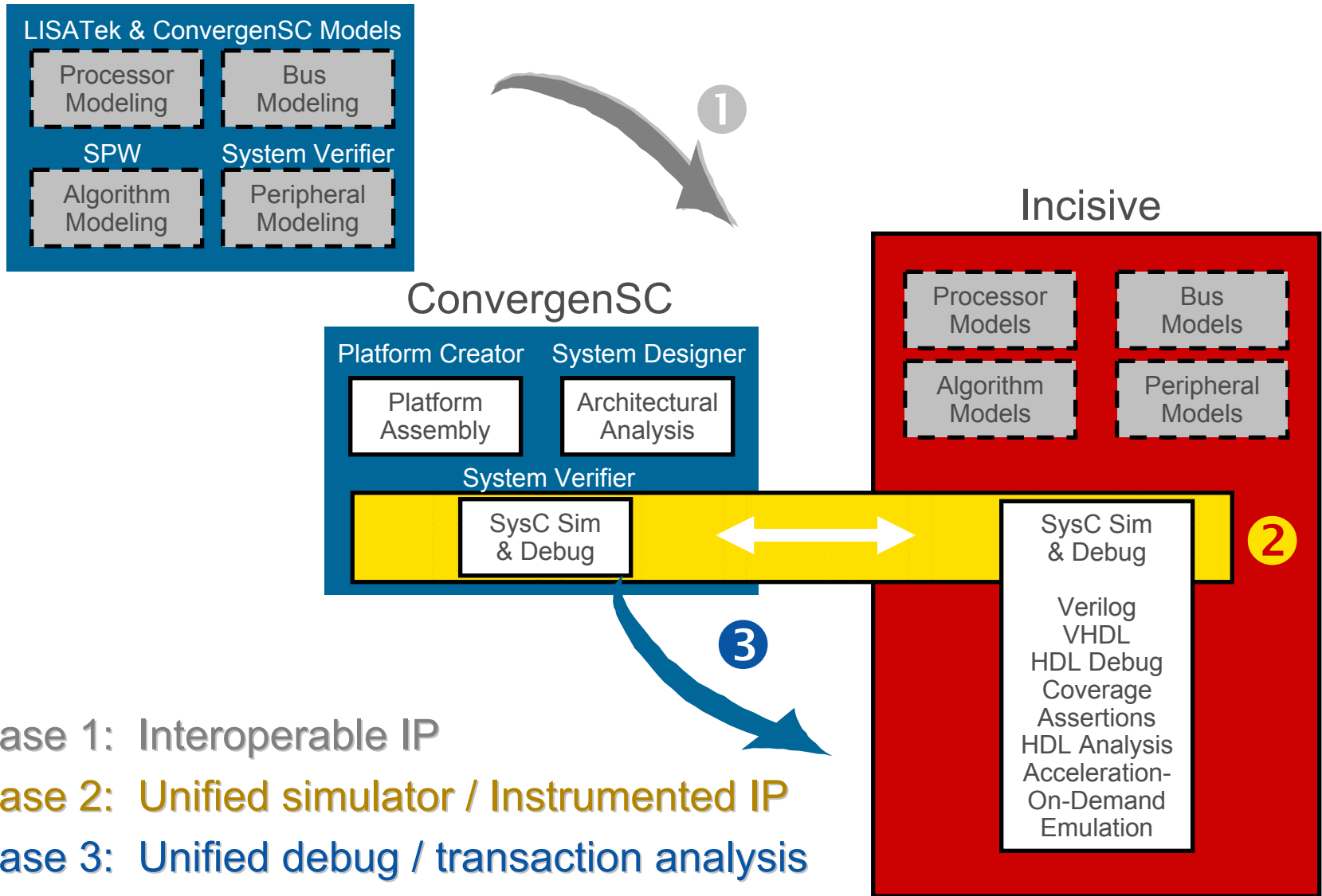
- Stand-alone SystemC block authoring
- SystemC platform creation and analysis
- SystemC based HW / SW coverification
- Top down design dominated by SystemC

- RTL verification with SystemC testbench
- Refinement of a mixed language fixed SoC platform
- Hardware accelerated mixed language coverification
- Bottom up design dominated by RTL and mixed environments

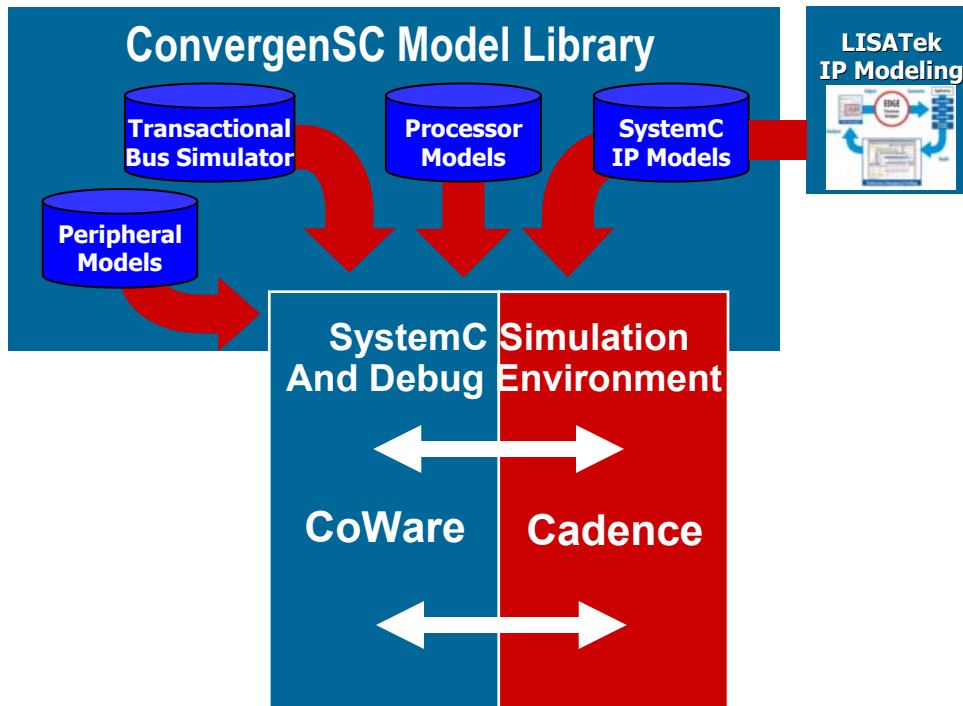
Top Down Flow & Tool Usage



CoWare/Cadence Development Phases



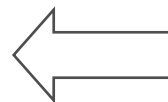
Phase 1: IP Interoperability



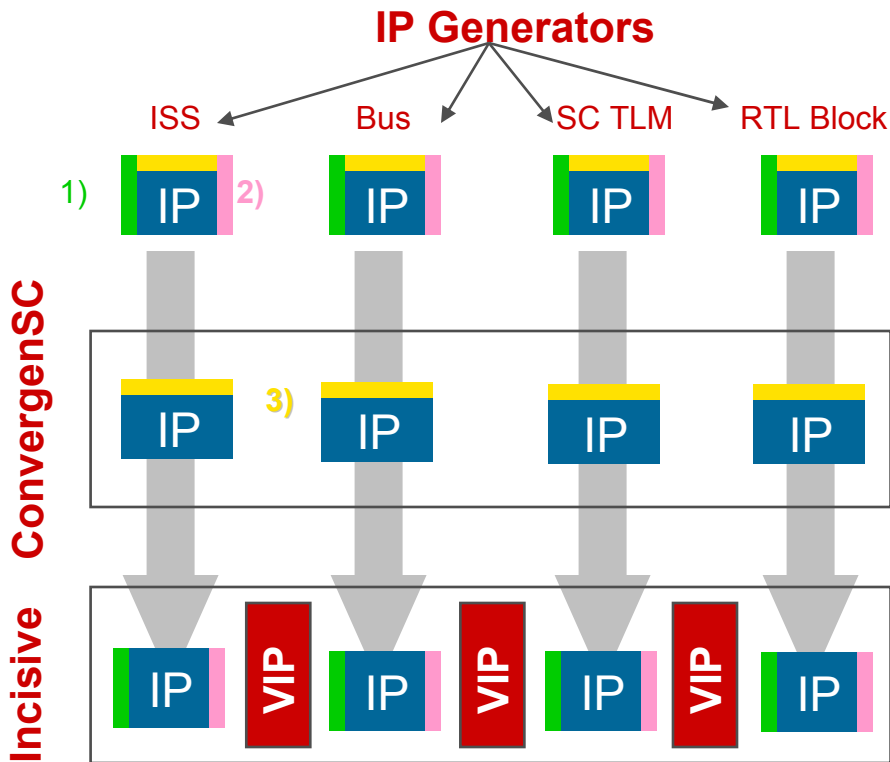
- CoWare IP Models are being re-compiled so that they will work in both ConvergenSC & Incisive SystemC simulators
 - Same library available for design and verification
 - Consistent simulation results
- A unified simulation kernel will allow all models to interoperate without re-compile
 - Less design data maintenance
- Large selection of models available in SystemC

ConvergenSC Model Library Includes:

ARM7TDMI, ARM7TDMI-S, ARM720T
ARM920T, ARM926EJ-S, ARM940T,
ARM946E-S, ARM966E-S
ARM1020E, ARM1020E, ARM1026EJ-S
ARM1136J-S, ARM1136JF-S
MIPS 4Kc, MIPS 20Kc, MIPS 25Kf



Phase 2: Instrumenting IP



1) Instrumented Transaction Recording for Incisive SimVision and Transaction Explorer

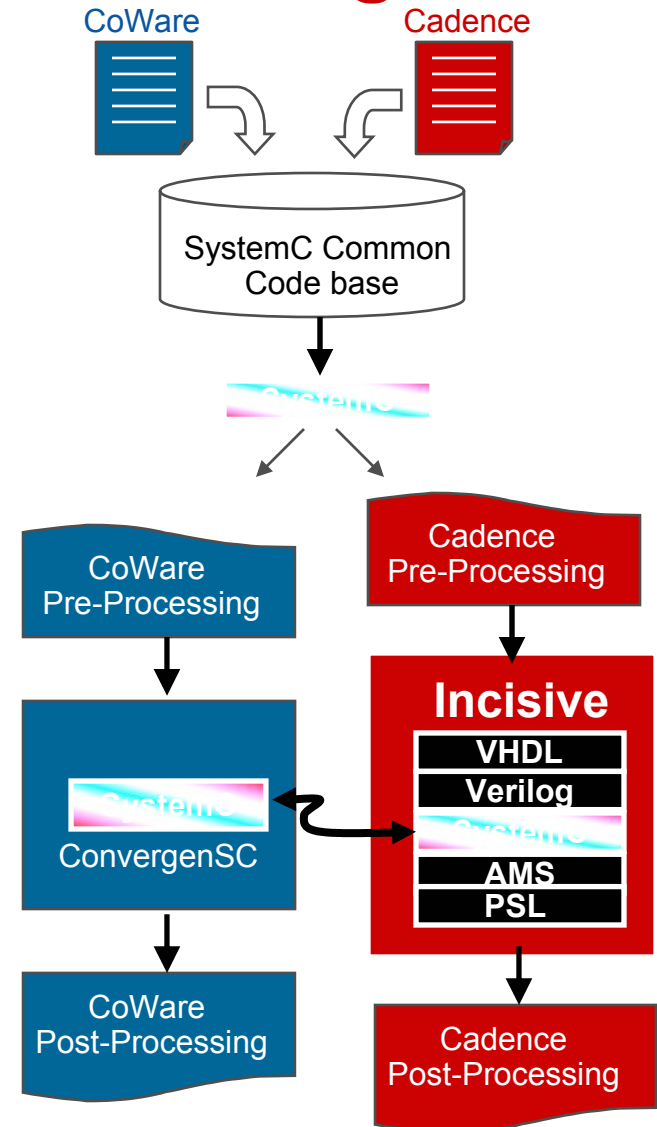
2) Instrumented Assertions for Incisive PSL Sugar Analysis Environment

3) Instrumented Architectural analysis for ConvergenSC

- IP which is created with CoWare tools will be instrumented with advanced features used by both Incisive and ConvergenSC:
 - Maximizing design and verification productivity
 - Unify IP between environments
- Existing Instrumentation includes:
 - Architectural analysis to identify bottlenecks and optimize design
- New IP instrumentation includes:
 - Transaction recording to simplify the simulation analysis
 - Assertions allows capture and measurement of design intent
- Addition of Cadence Verification IP (VIP) to complete the solution

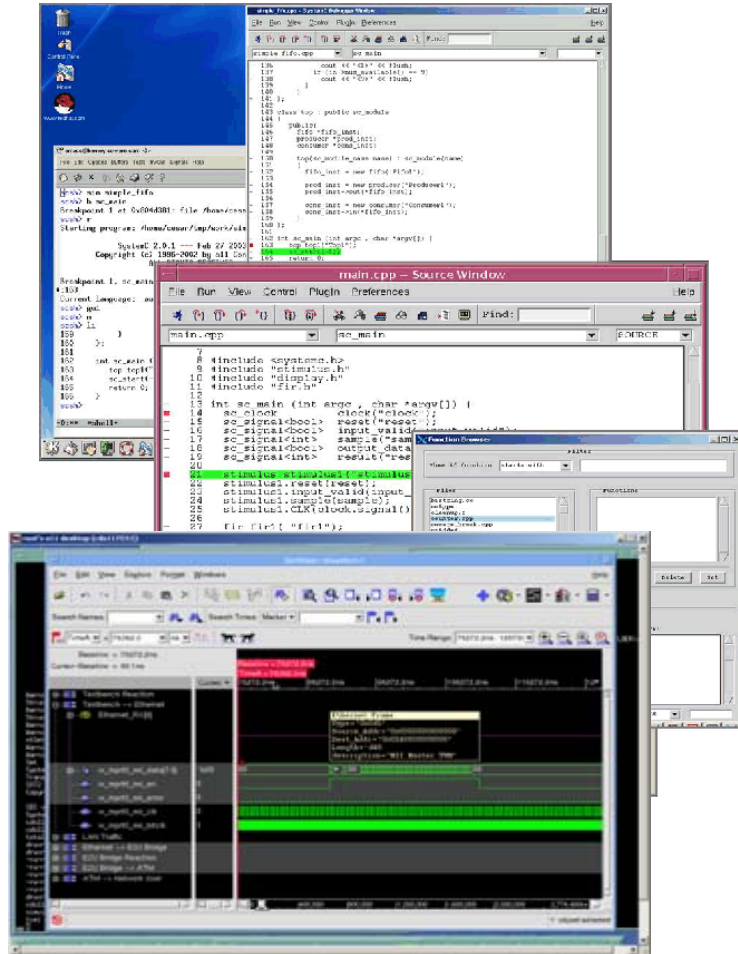
Phase 2: Joint SystemC Execution Engine

- Alliance creates a single code base of SystemC simulation technology
 - Guaranteed interoperability between models
 - Guaranteed consistency of results
- Shared code base maintained by both
 - Technology component integrated into each others respective products
 - Cadence inserts SystemC engine within Incisive and integrates into environment
 - CoWare inserts same SystemC engine within ConvergenSC
- All Incisive v5.2 features preserved
- All ConvergenSC v4.1 features preserved



Phase 3: Unified Debug / Transaction Analysis

Now full system-to-silicon verification



- Objective to provide a common debugging and analysis infrastructure for SystemC (definition Q1'04)
- Common debugger environment enables:
 - Common SystemC source code debugging
 - Multiple debuggers with synchronized breakpoints, single step, etc.
- Users of both environments will check out the necessary licenses depending on the features that are used
 - ConvergenSC analysis features (e.g. bus traffic and contention, cache/memory analysis) will be available from Incisive
 - Incisive's transactions explorer will be available from ConvergenSC

Future Vision: Platform-Based Design

- Designers want an integrated environment for platform based designs
- Platform based design methodology starts with Platform Configuration (for both platform creation and derivative design)
 - This could be a graphical block based design tool with full SystemC 2.0 support
 - Hierarchical design database with XML IP Metadata descriptions
 - Potential future support for SPIRIT
- Continuous refinement methodology and verification analysis are possible in “Incisive”
- Possible future development of technologies and flows

