Requirements-Driven Design from Concept to Implementation to Compliance

Abstract

- As electronic design complexities escalate and the ramifications of project failure become more severe, the need for new design techniques becomes more critical. New design techniques are needed not only to describe these designs, but to aid in verification, reusability, reliability, agility and management, while also reducing project risk. Applying a requirements-driven design approach, which is being rapidly adopted for projects with compliance needs such as DO-254, ensures that the verified design performs its intended function before costly commitment to silicon.
- A requirements-driven design approach is beneficial to any design project and vital to all safety/mission/security critical projects as the process will improve efficiency, productivity, quality, and project schedule predictability. However, raising the level of design abstraction above RTL will contribute an entire new dimension of benefits to this design approach. Raising the design description language up from RTL to electronic system level (ESL), which is TLM-based (transaction level modeling), will enable more design to designed, explored, validated, and co-designed with corresponding software and firmware to result in faster design cycles and higher quality projects.
- Automating the traceability and management of the design requirements from the original requirements source through all these levels of design and corresponding verification results provides status at any stage and delivers project visibility as never before, enabling the design team to make informed decisions and adjustments to keep the project on schedule. As this paper will discuss a requirements-driven design process in terms of SoC design, the approach and concepts are applicable to all design – it's just good design practice!

Requirements-Driven Design from Concept to Implementation to Compliance

Valerie Rachko

HDL & ESL Design Creation Marketing Director

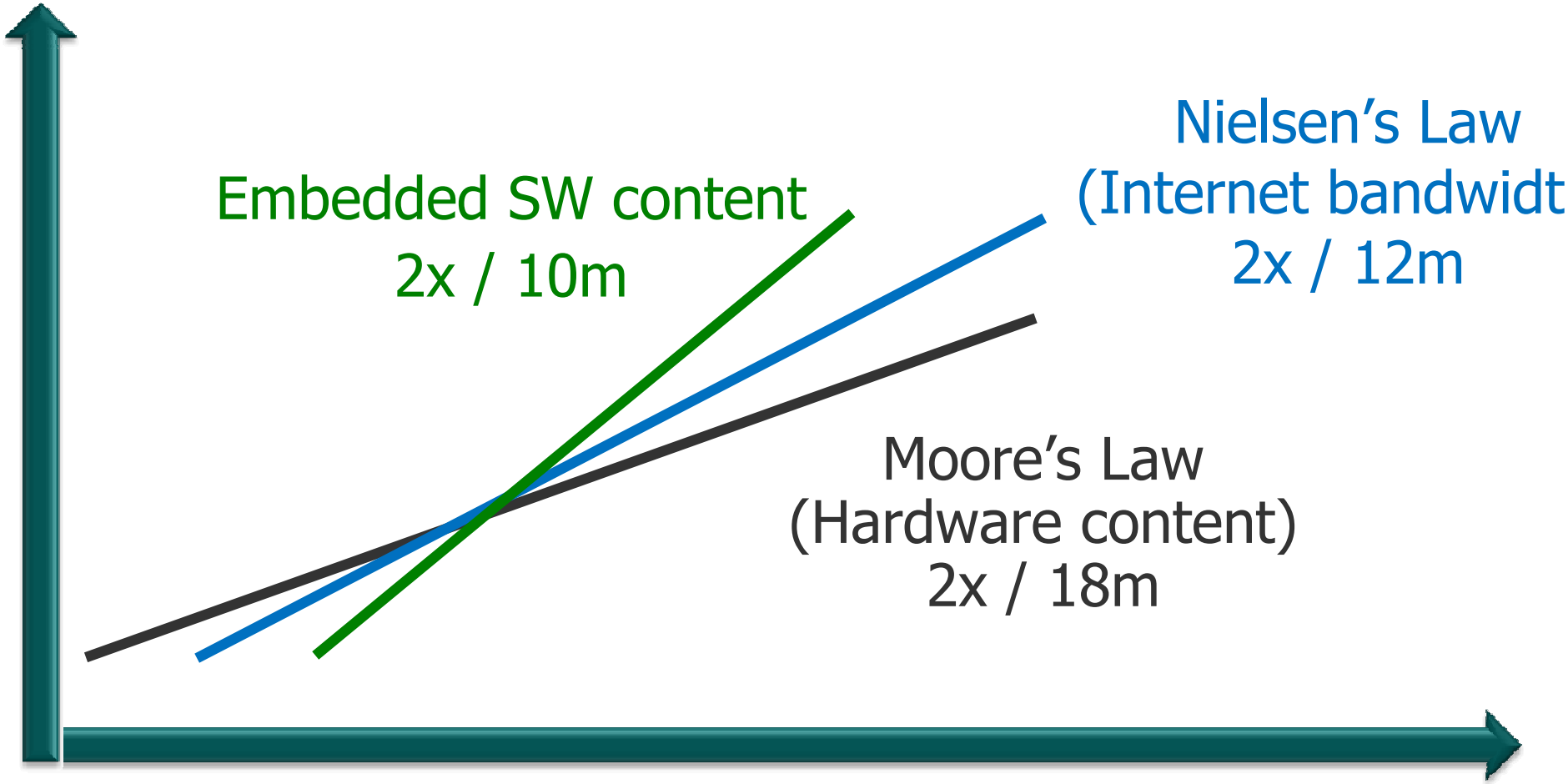
Design Creation & Synthesis Division

Mentor Graphics Corporation

April 2011

Mentor Graphics®

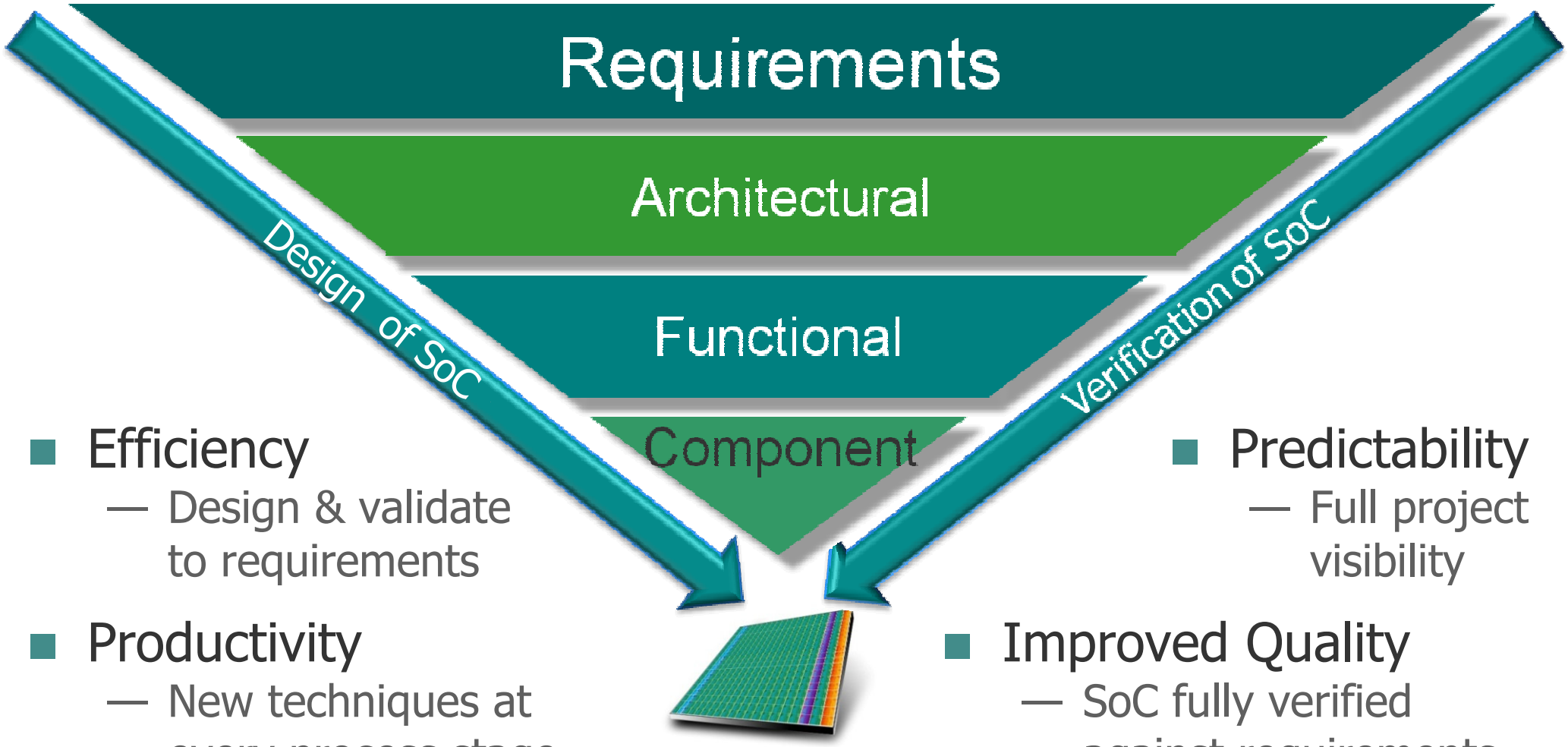
Putting Complexity Into Perspective
Size, Software, Speed



Need new design techniques to effectively & securely reach project goals

Requirements-Driven Design Process

Parallel Design & Verification → Implementation



- Efficiency
  - — Design & validate to requirements
- Productivity
  - — New techniques at every process stage
- Predictability
  - — Full project visibility
- Improved Quality
  - — SoC fully verified against requirements

Who Needs a Requirements-Driven Design Process?

- Any company who cannot afford design failure
  - Mil-Aero
  - Security
  - Medical
  - Automotive
  - Railway
  - Nuclear/Energy
  - Banking Systems
  - Industrial Controls
  - Shipping Systems
  - Critical Infrastructure
- Companies needing to adhere to design compliance regulations & audits
- It's just good design practice!

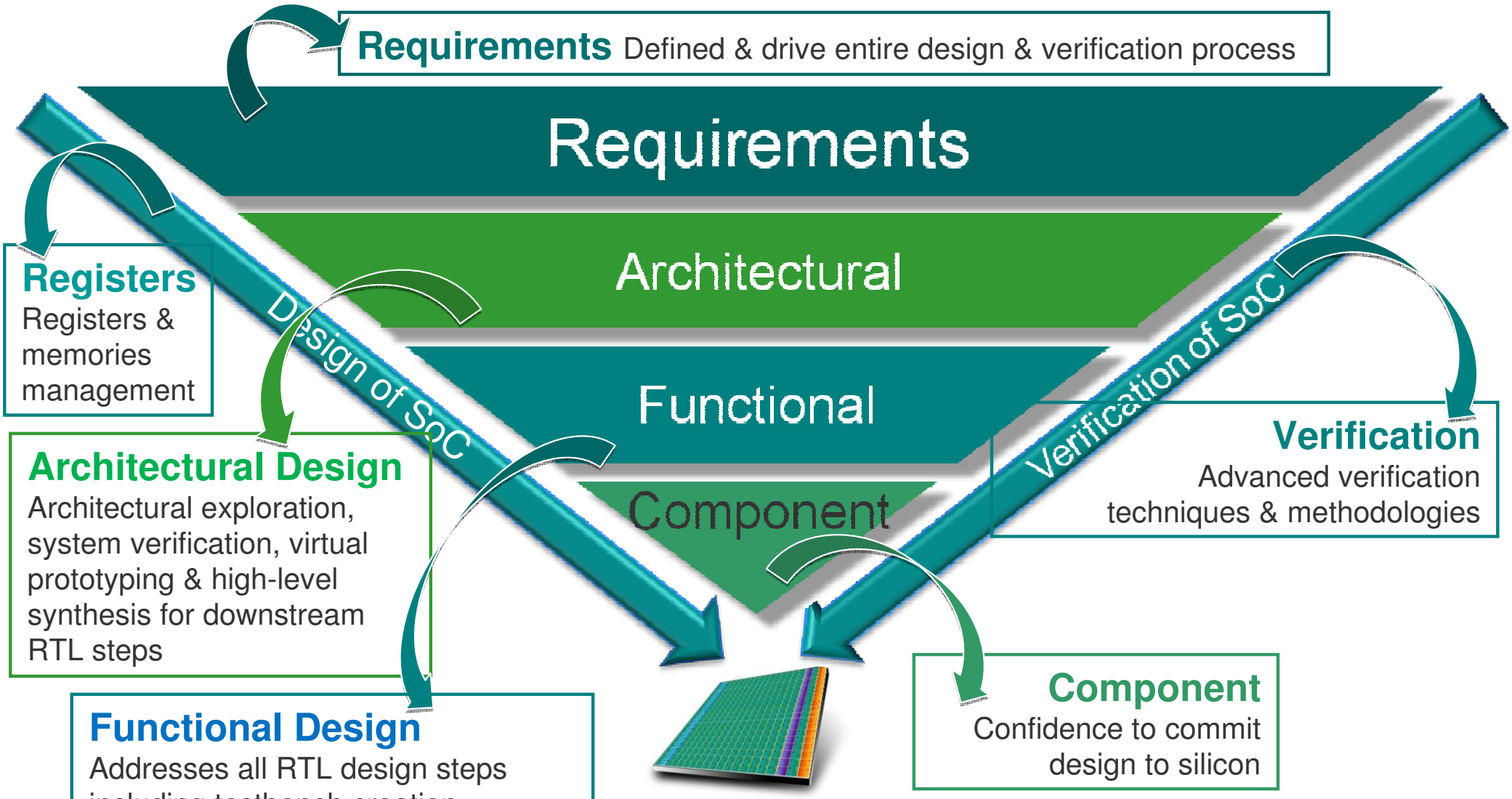
DO-254:
Design Assurance & Requirements

- Requirements define the "intended function" of a device
- CFR Title 14 (Aeronautics & Space) 25.1309:

"The equipment, systems…must be designed to ensure that they perform their intended functions under any foreseeable operating condition."

- DO-254 is a means to show compliance to this rule
- DO-254 & other similar standards
  - Objectives involving requirements & ensuring that the design meets them!
- Establishing a requirements-driven design flow is key to design assurance

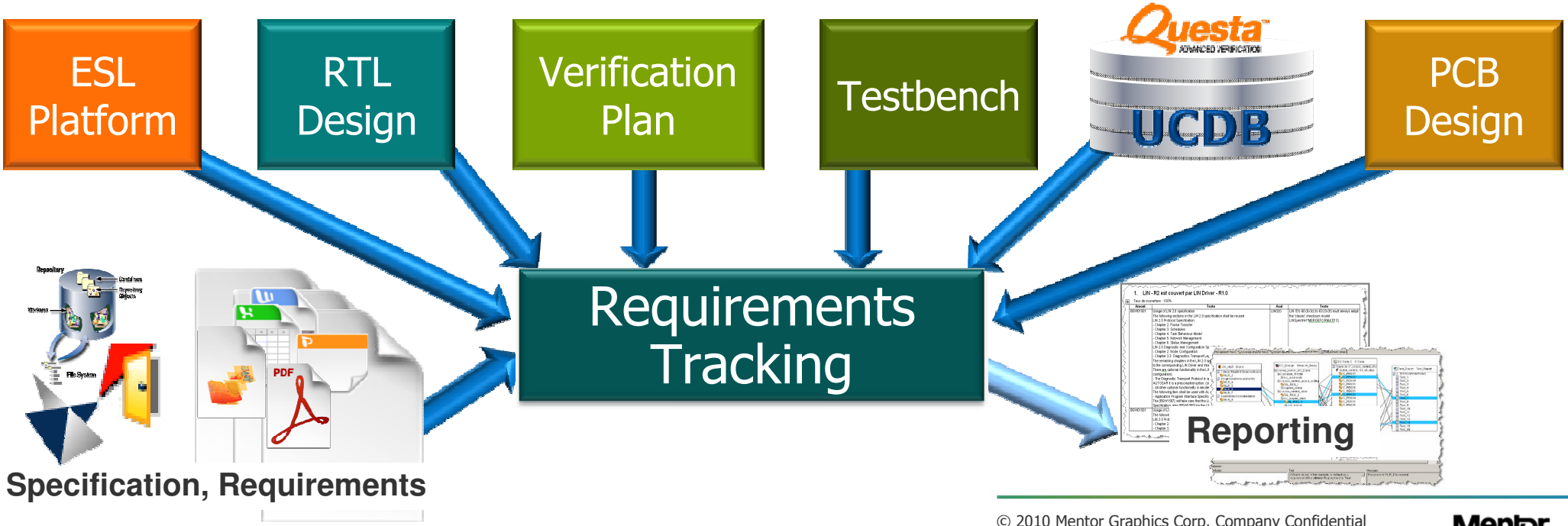
Requirements-Driven Design Process



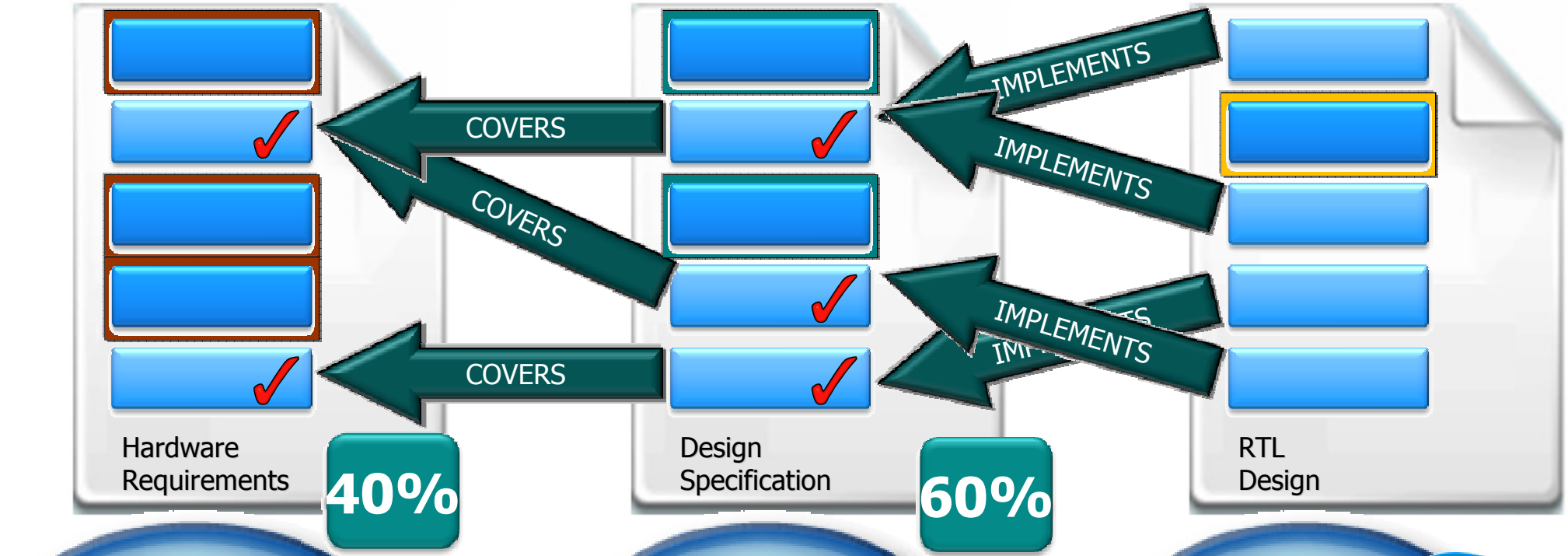
Requirements Tracking

Requirements

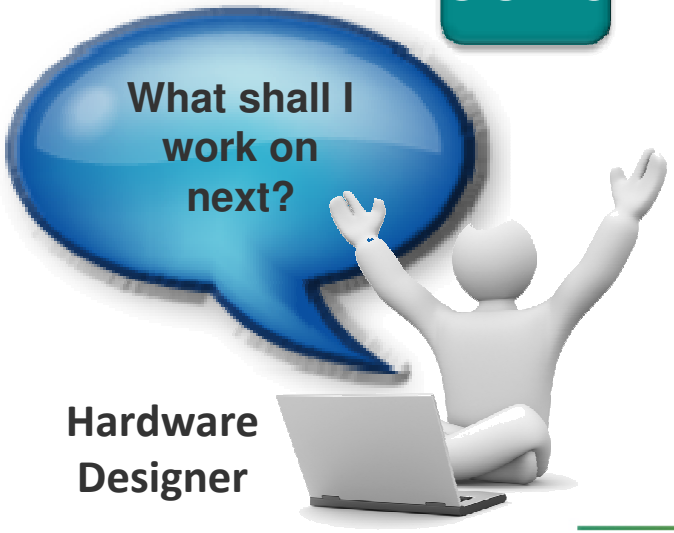
- Track requirements through all levels of design & verification
- Manage requirements for ECOs, volatility, & automate reports
- Ensure all requirements are designed-in & fully verified, indicating simulation complete



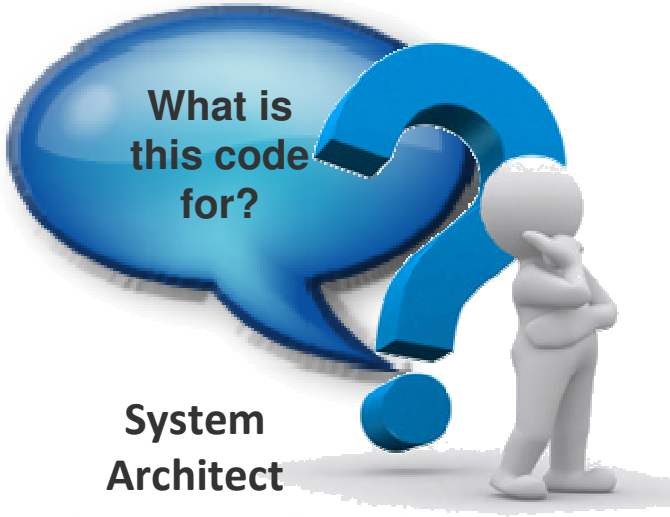
Trace, Manage, & Document Requirements & Their Changes



Certification Authority



Hardware Designer



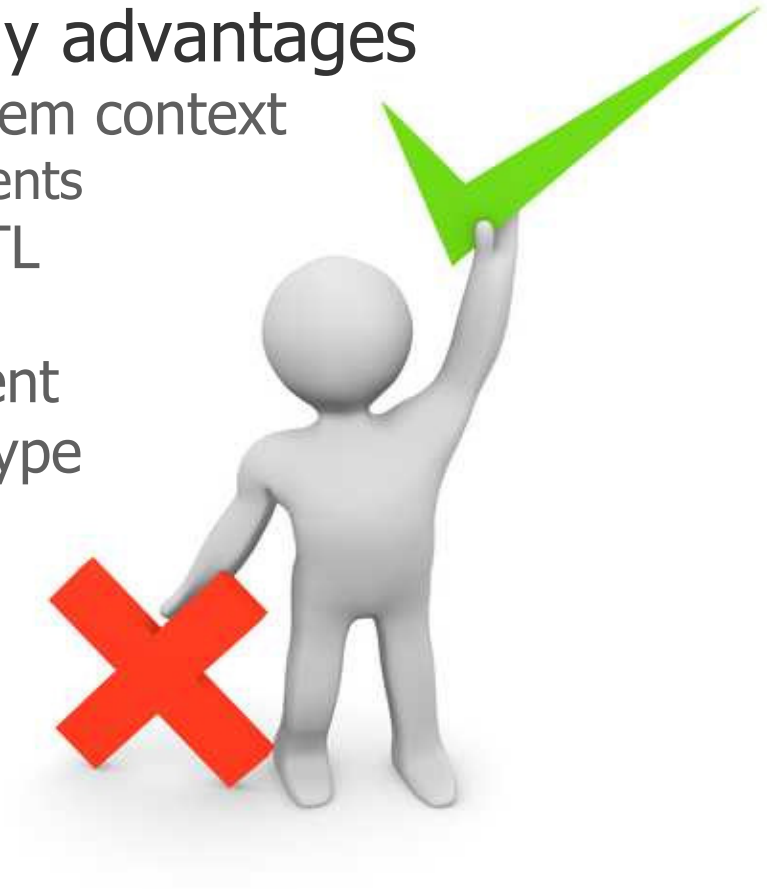
System Architect

Architectural Design

Architectural

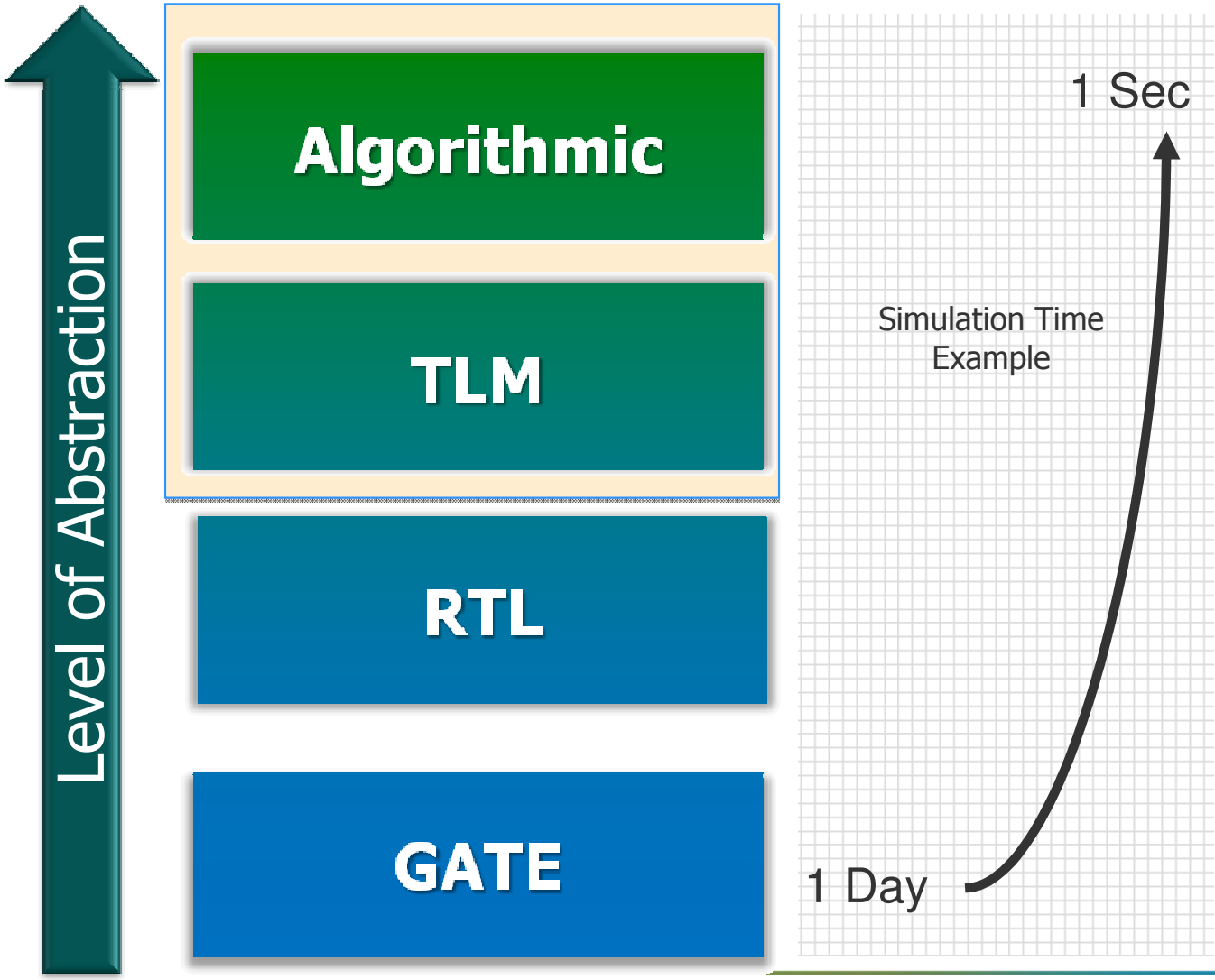
ESL - Electronic System Level

- Abstraction above RTL provides many advantages
  - Simulate HW functional spec in the system context
    - First stage of designing to the requirements
  - Explore architecture tradeoffs before RTL
    - Performance, Power, bandwidth, etc
  - Enable Software integration, development & validation before RTL – Virtual Prototype
  - Reduce manual RTL coding via high level synthesis
  - Reduce RTL verification effort
    - Find bugs early
    - Correct-by-construction RTL



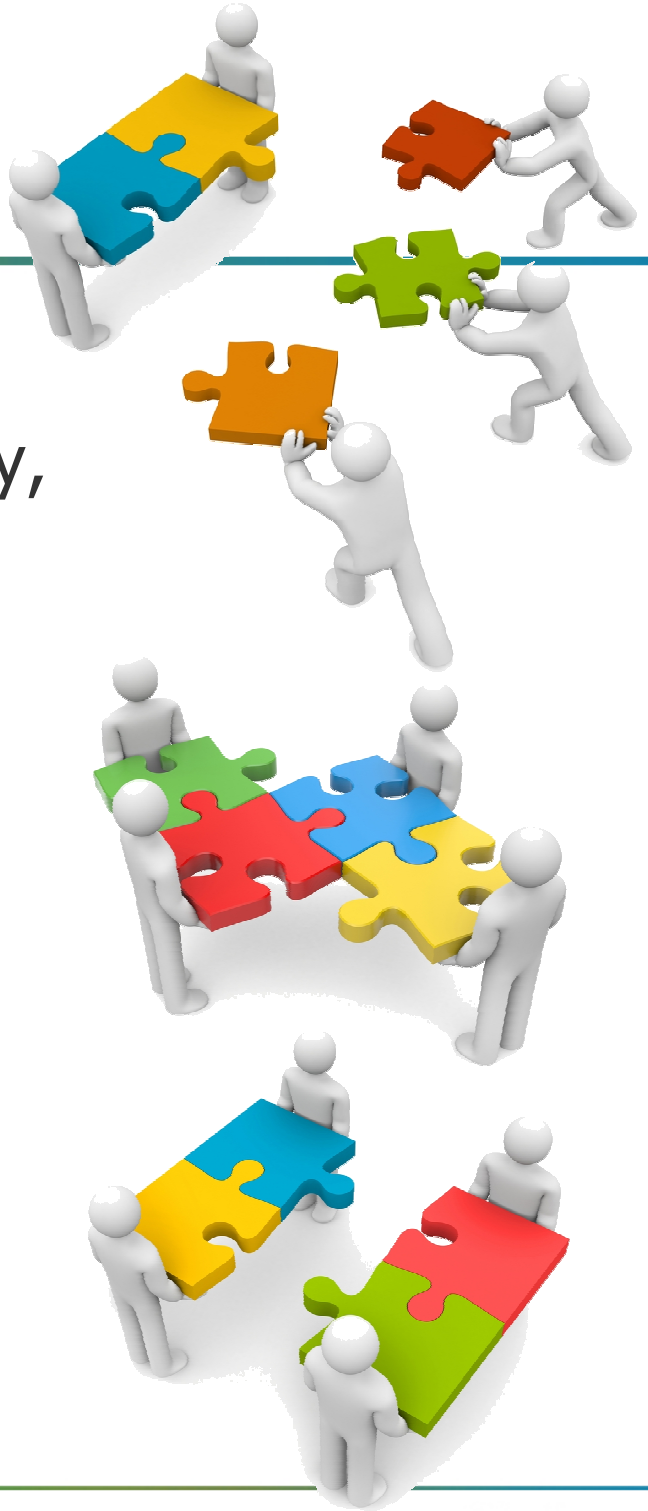
Why is ESL Design & Verification Important?

- Speed in coding the design, Speed in verifying the design
  - SystemC
  - TLM-2.0
- Co-development of firmware & software
- Maximum power optimization
- Full SoC/system verification



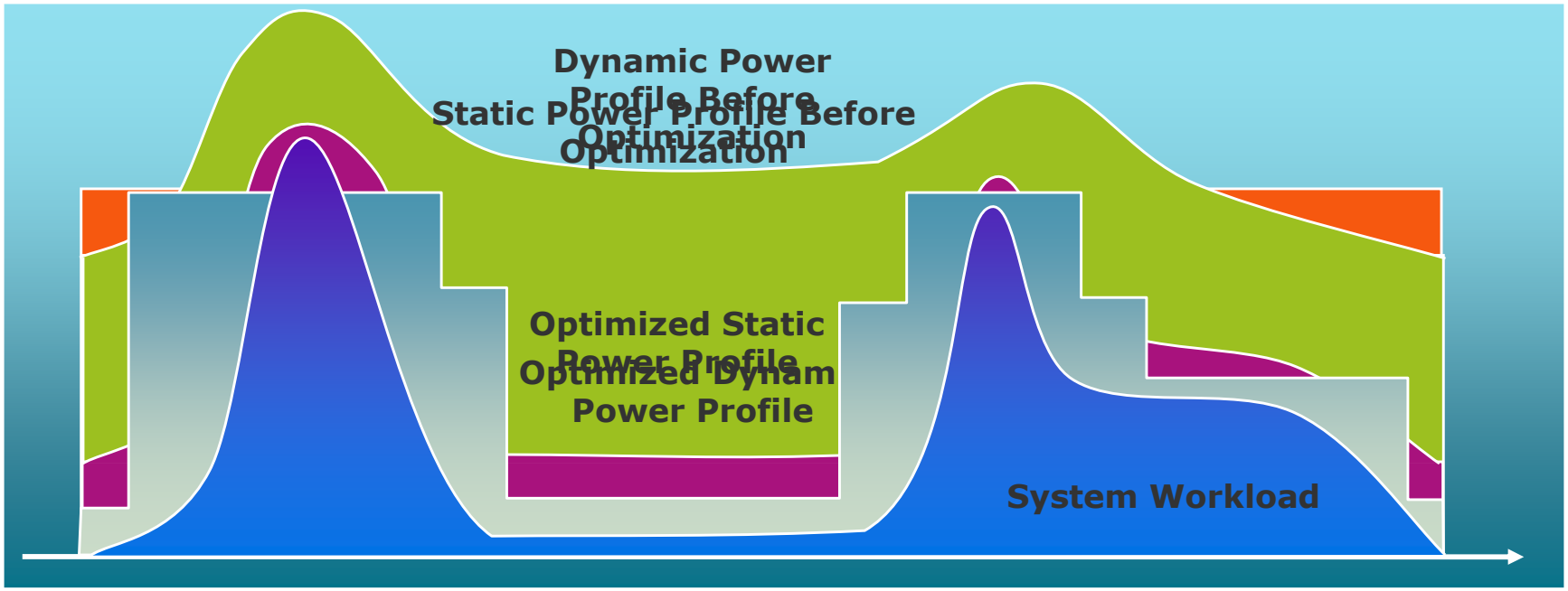
Architectural Exploration

- Permits full chip "what if?" analysis
- Facilitates trade-offs for best functionality, performance & power
  - Hardware & software
  - IP, processor, peripherals, busses, etc.
- Delivers a virtual prototype for firmware & software development
- Enables full chip verification before any RTL code is created
- = Conceptual Design
  - Part of DO-254 design process



System Workload Power Optimization

- Significant static & dynamic power reduction when optimizing power based on system workload



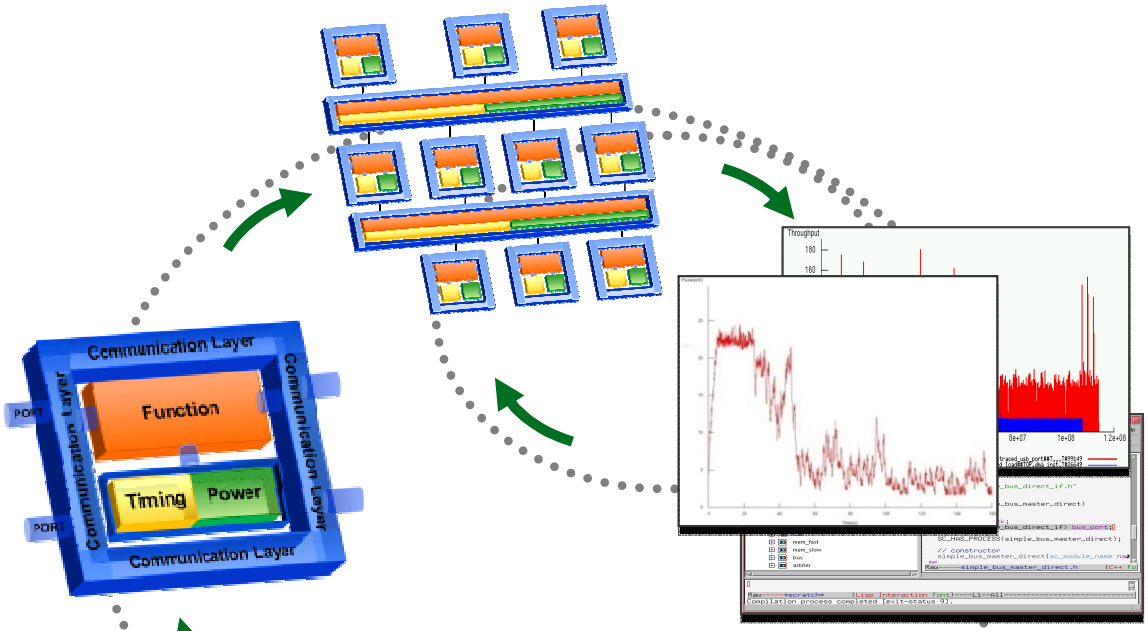
Both dynamic & static power can be optimized

Source: LSI

- Fast simulation of power under real system level scenarios

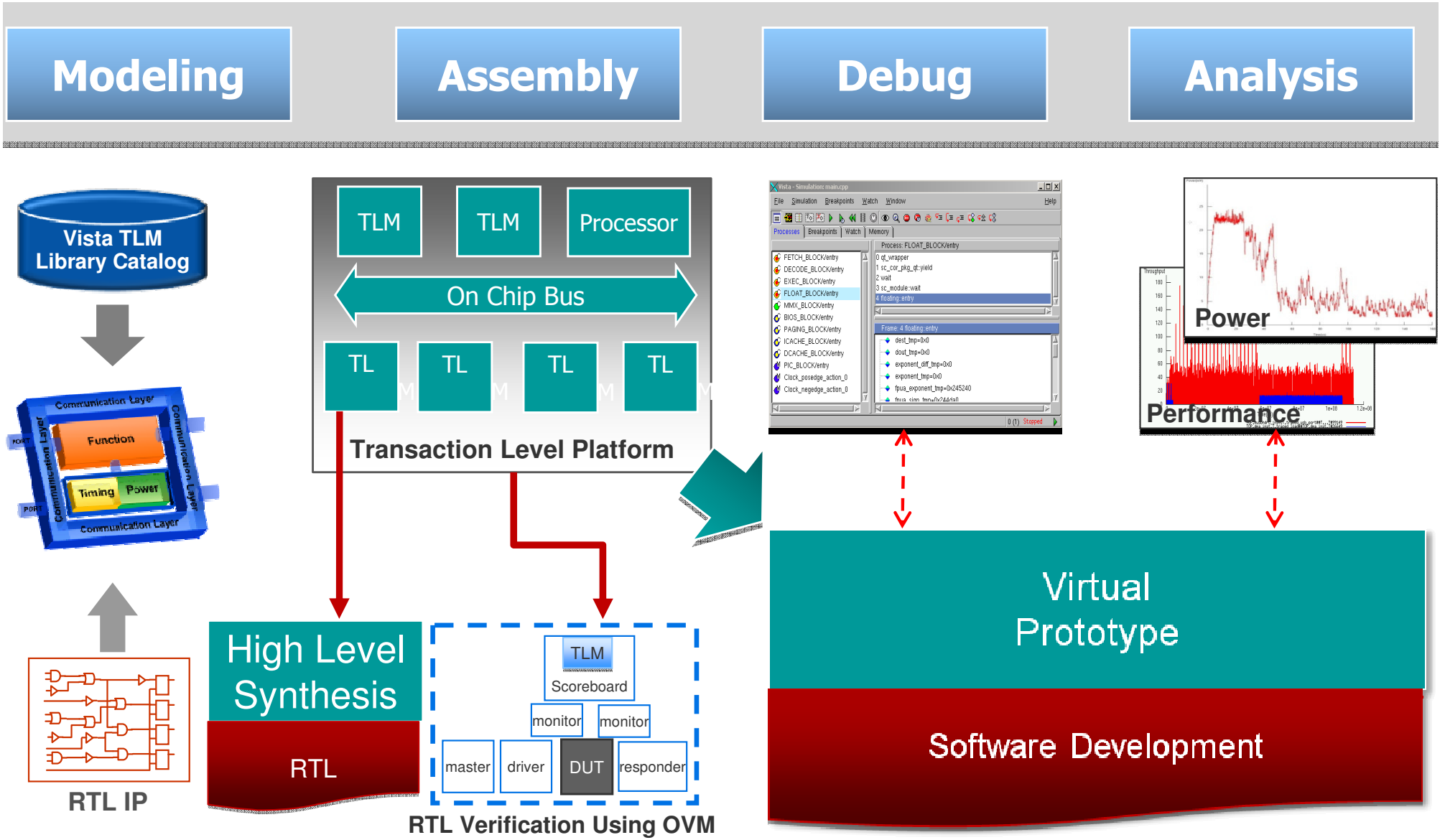
Power & Performance Optimization Flow

- Model TLM Timing & Power via policies
- Assemble transaction level reference platform
- Analyze Timing/Power in a system context
- Modify Timing/Power policies on the platform architecture
- Quickly iterate optimizing for timing & power



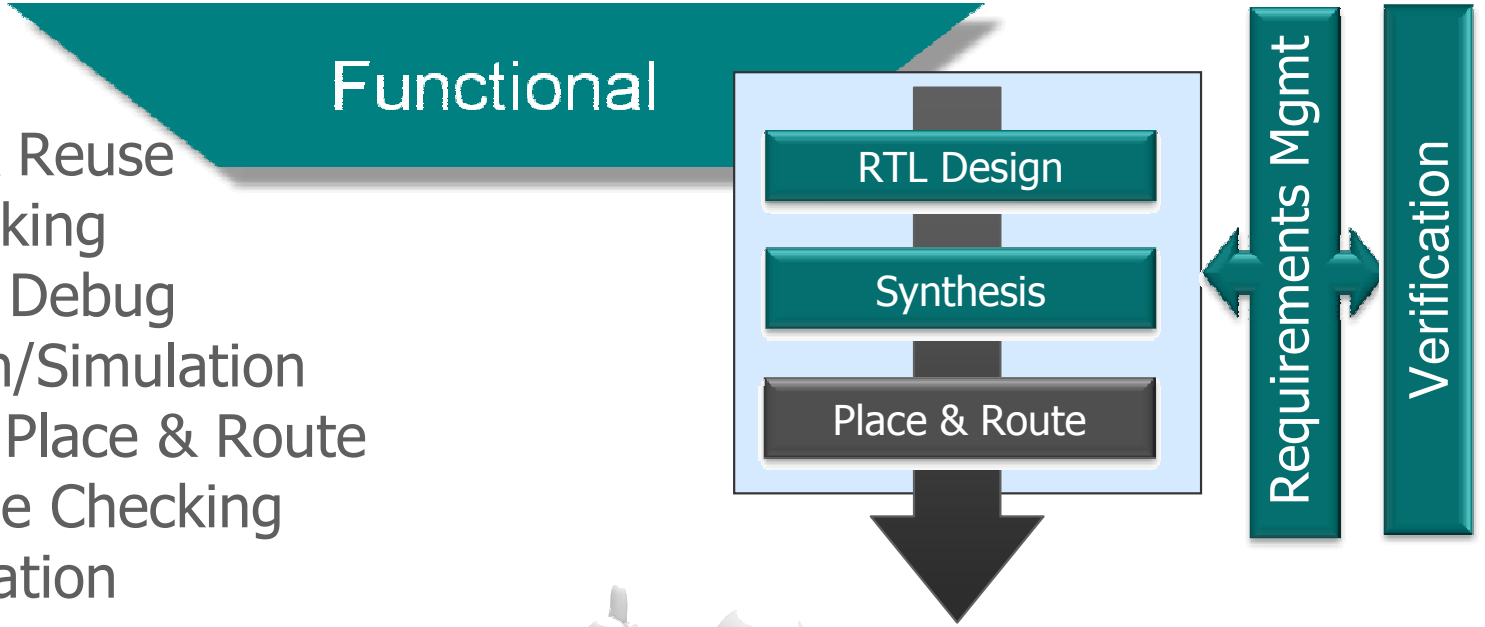
Dynamic Power		
Transaction	Power	Time Interval
USB.READ	0.16mw	200
Clock Tree Power		
0.5mw		
Static (Leakage) Power		
0.2mw		

ESL Flow



Functional Design: RTL → Implementation

- RTL design
  - Creation & Reuse
  - Code Checking
  - Analysis & Debug
  - Verification/Simulation
  - Synthesis, Place & Route
  - Equivalence Checking
  - Documentation
- New changes of significance
  - Requirements tracking through all design phases
  - Automation of many steps
    - Requirements tracking
    - Design Code Checking
    - Documentation
    - Register generation



Automated Design Checking & Documentation

DO-254 TOGETHER

Design Checking

- Coding guidelines to enforce effective & efficient design code
  - Catches violations before simulation, synthesis & production

Automatic Design Checking Significantly ⬇ Costs & ⬆ Quality

  - Ex. 10,000 lines RTL code

Manual Checking	**@200 lines/hr → 50 hrs**
Automatic Checking	**→2 mins 40 sec → 1100x improvement!**

Documentation

- Essential for reuse
- Required in design assurance processes

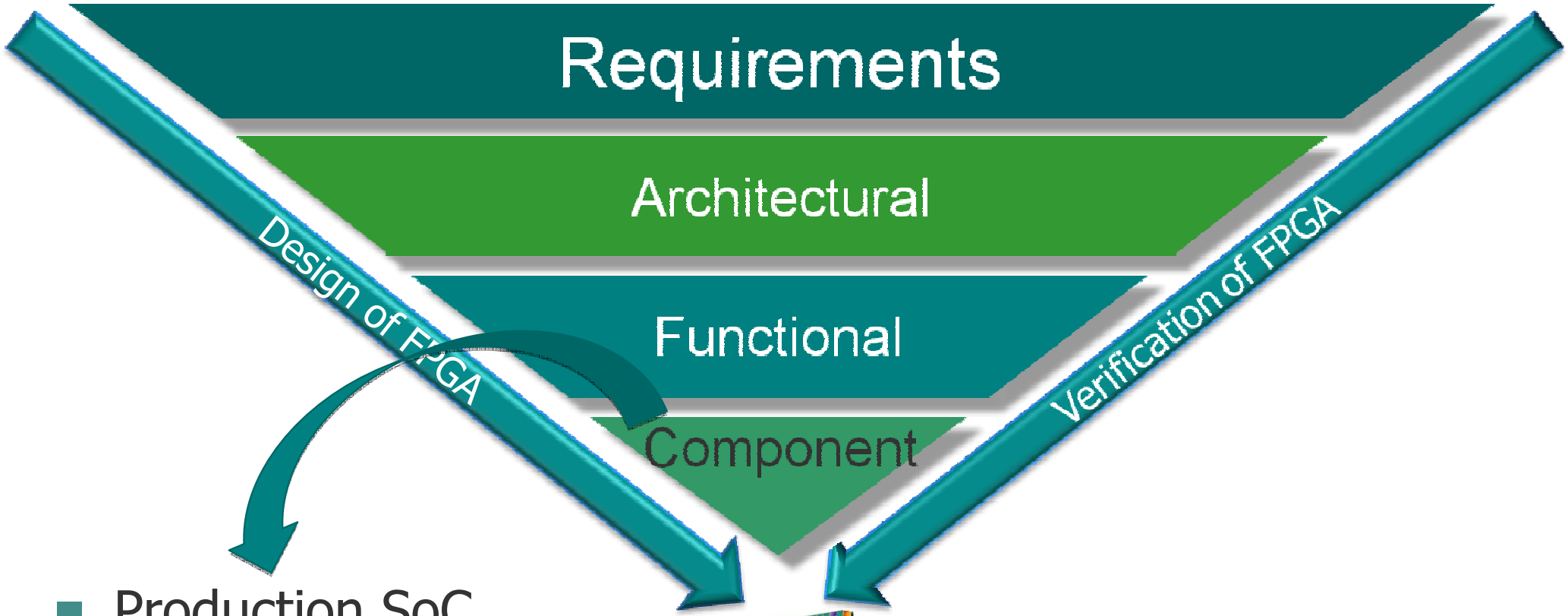
Automated Documentation

- Speeds design review processes
- Includes all design data & design artifacts
- Controlled under version management
- Visualizes RTL code
- Exports to document files
- Snapshots project in HTML

Mentor Graphics

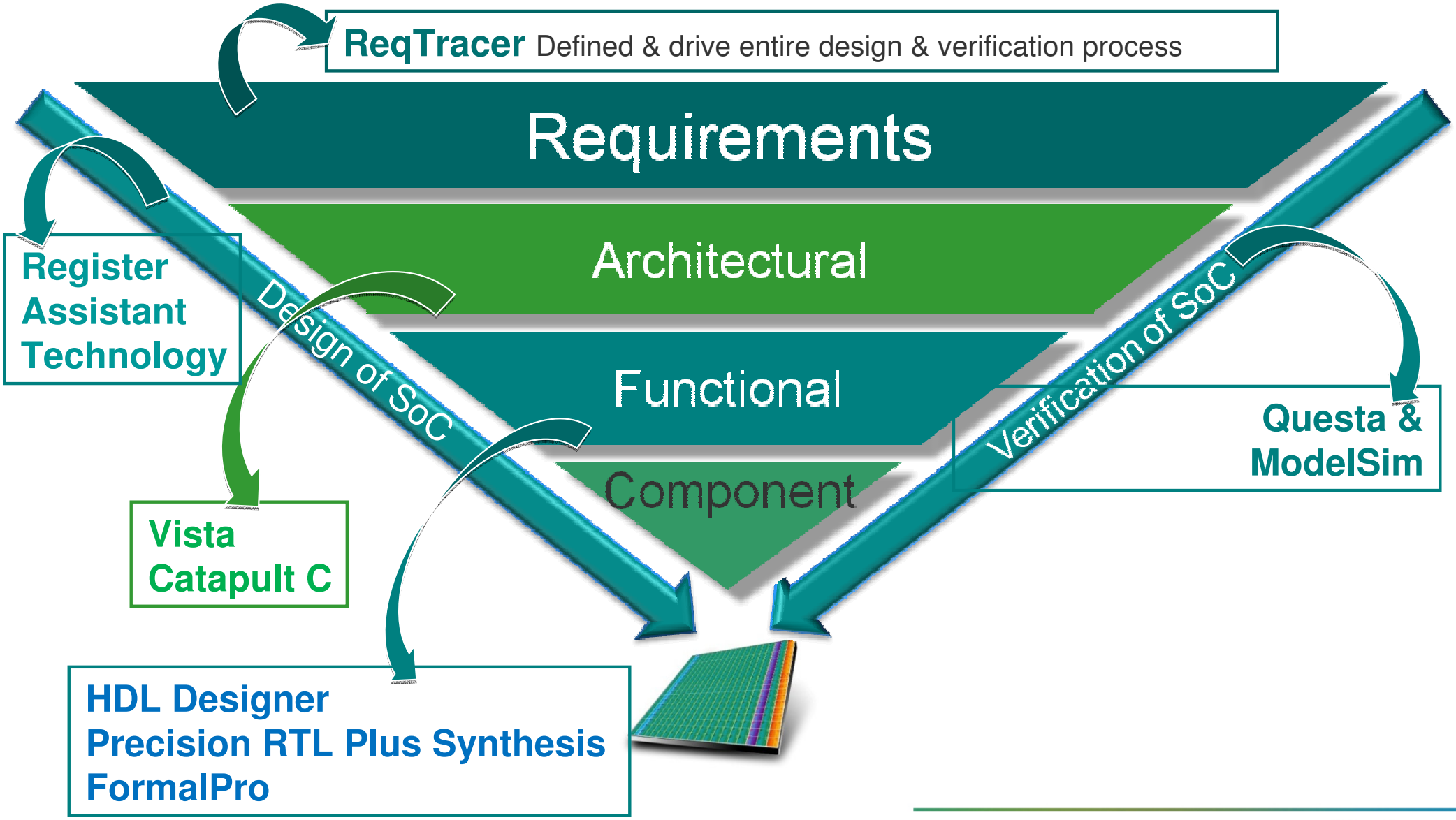
Component

Implementation → Physical Device



- Production SoC
  - — Produce SoC & in-circuit / in-lab testing
  - — Commitment to silicon & production

Mentor Graphics' Requirements-Driven Design Process



In Summary

- Safety/Mission/Security Critical designs require additional techniques to ensure design assurance
- A requirements-driven design process enables
  - Added design & process assurance
  - Documentation of design
  - Repeatable process
  - Compliance to DO-254 & other standards
- ESL delivers many advantages
  - Architectural exploration
  - Virtual prototyping
  - Low-power optimization
  - Full SoC verification
- Many designs today should implement requirements-driven & ESL to improve design quality

www.mentor.com

Register Assistant Overview

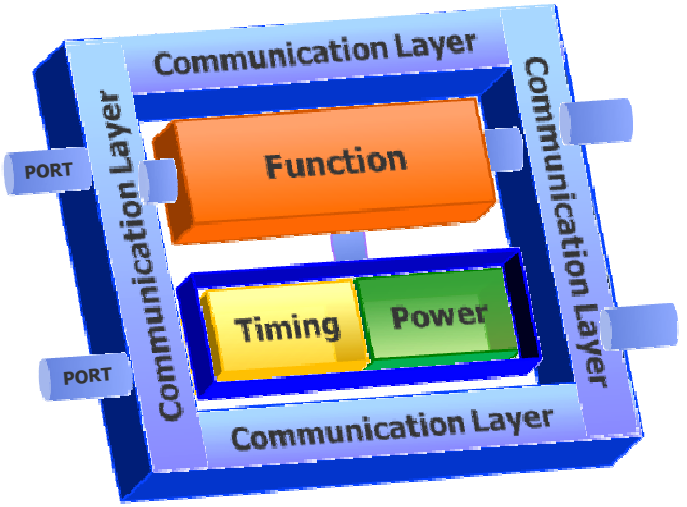
Register/Memory Definition & Management for the Entire Design Process

- Central, Scalable & Extensible Register/Memory Datamodel
  - Enables easy specification of registers
  - Manages register changes
  - Eliminates hand coding & resultant mistakes
  - Executes consistency checks
- Automatically Generates Register Outputs
  - OVM Register Package
  - Documentation
  - Roadmap
    - UVM package generation
    - Synthesizable RTL
    - Full UI
    - ESL
    - IP-XACT 1.5

- Register Input Formats
  - CSV, IP-XACT 1.1-1.4, API via Javascript, XLS
  - Roadmap: spreadsheet, SystemRDL, IP-XACT 1.5, SystemC/C++, full GUI

Vista Scalable TLM: Power Model

- Transaction-level modeling of all power types
  - Static (leakage) power
  - Clock tree power
  - Dynamic power (per transaction)
- Dynamic power is assigned to each transaction type
- Vista TLM power model is
  - Reactive to incoming traffic and inner states
  - Supports voltage and frequency scaling



Power Policies

Dynamic Power		
Transaction	Power	Time Interval
USB.READ	0.16mw	200
USB.WRITE	0.44mw	20
Clock Tree Power		
0.5mw		
Static (Leakage) Power		
0.2mw		