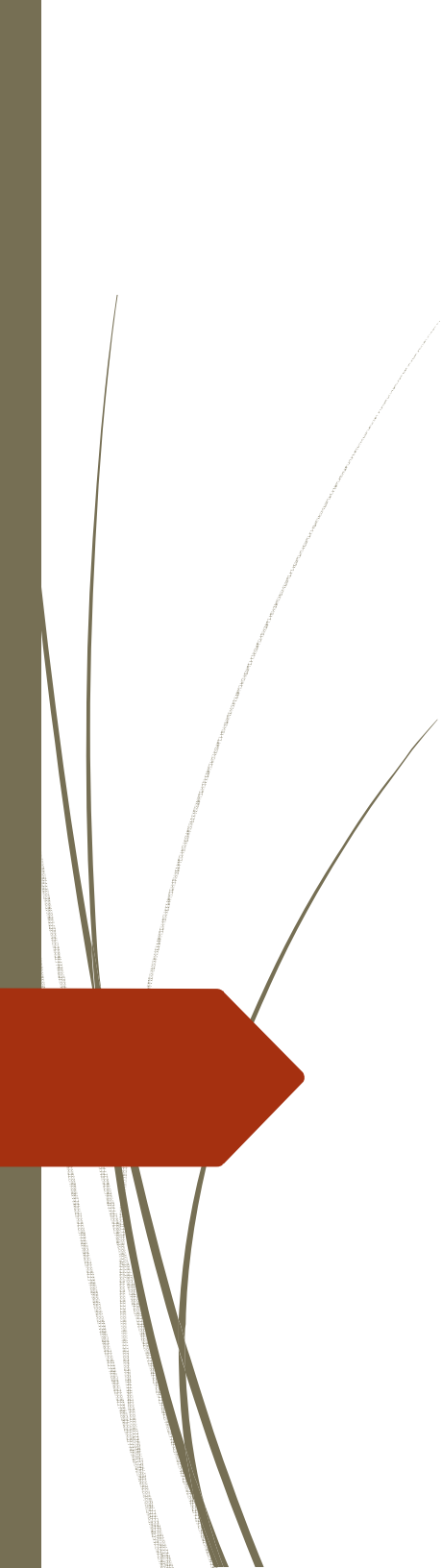


Hardware Functional Verification – Present and Future

HVC '2013

Yuval Caspi

Agenda

- Evolution of Functional Verification
- Languages and Methodologies Timeline
- Advanced Techniques Description and Trends
- Short Case Study
- 4 Challenges in Verification
- Future Verification Trends

“testing shows the presence of bugs not their absence”

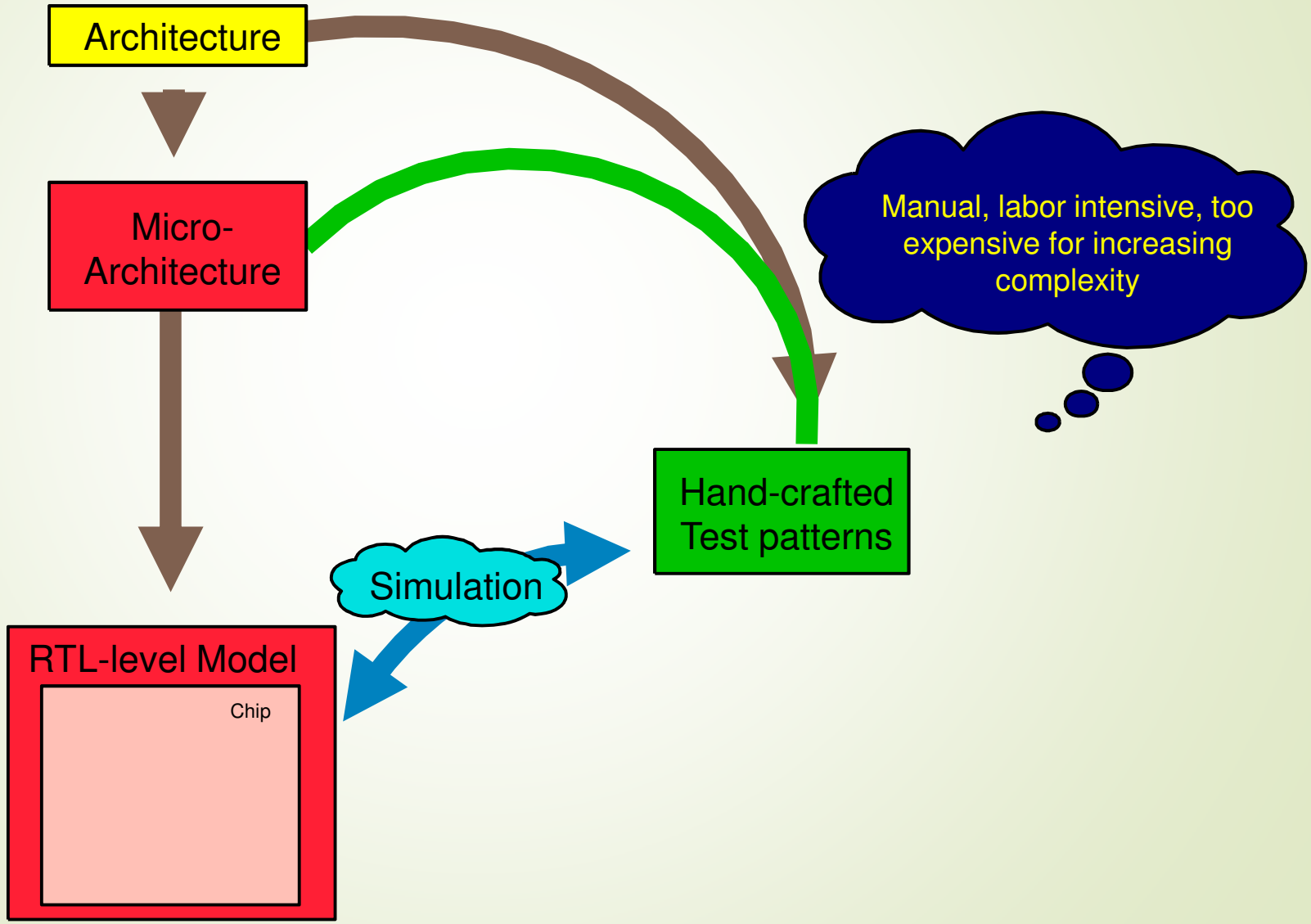
Edsger W. Djikstra

Verification Evolution is driven by SOC Design

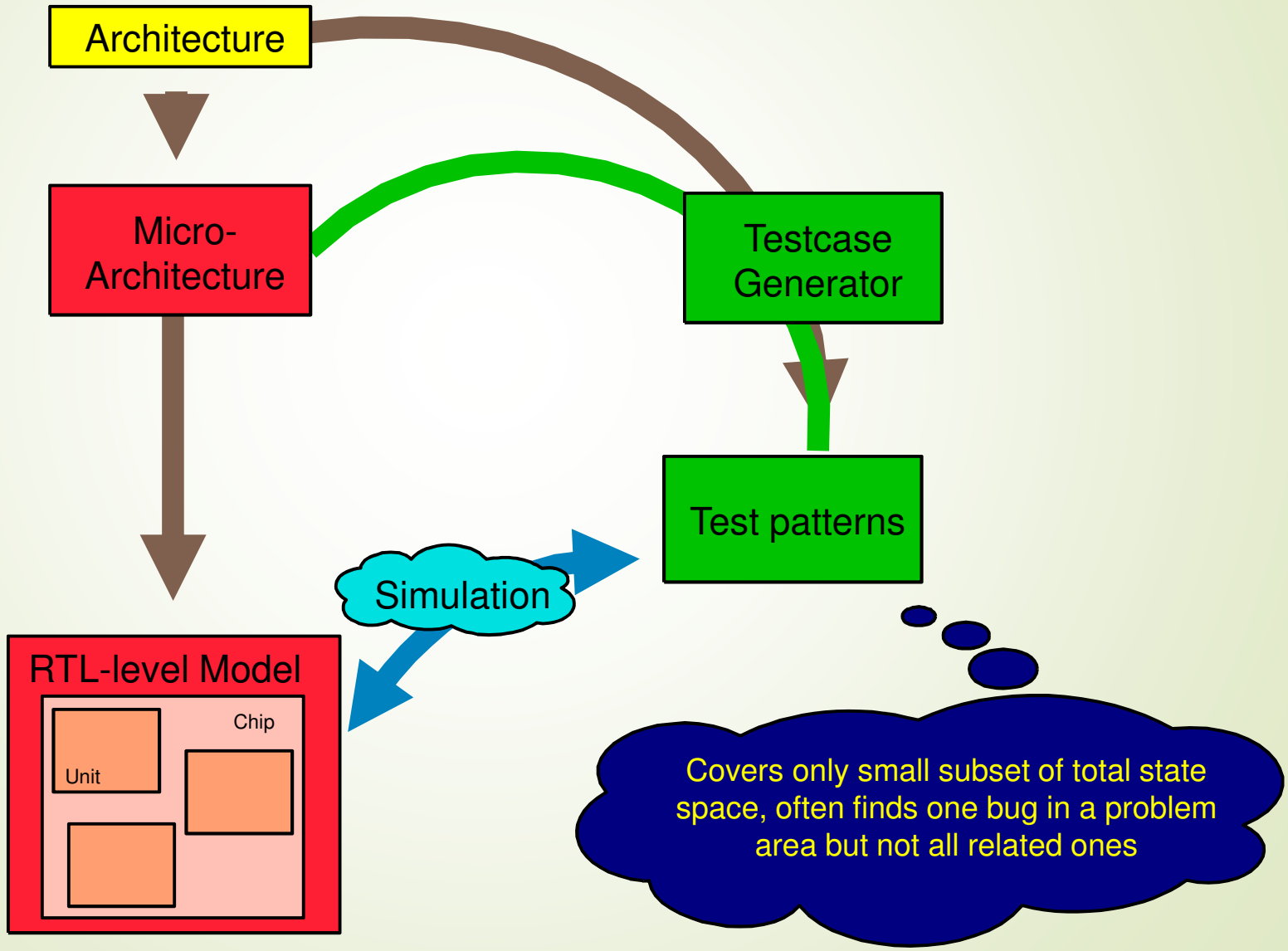
- Increasing Complexity
  - Continuous increase in number of IP's and embedded processors
    - 2006: 30-40 IP's, 1 CPU
    - 2011: 80+ IP's, 6+ CPU's
    - 2014: 120 IP's, 20+ CPUs?
- Exploding State spaces
- Increasing number of functions

  ... but ...
  - Reduced timeframe
  - Same Resources

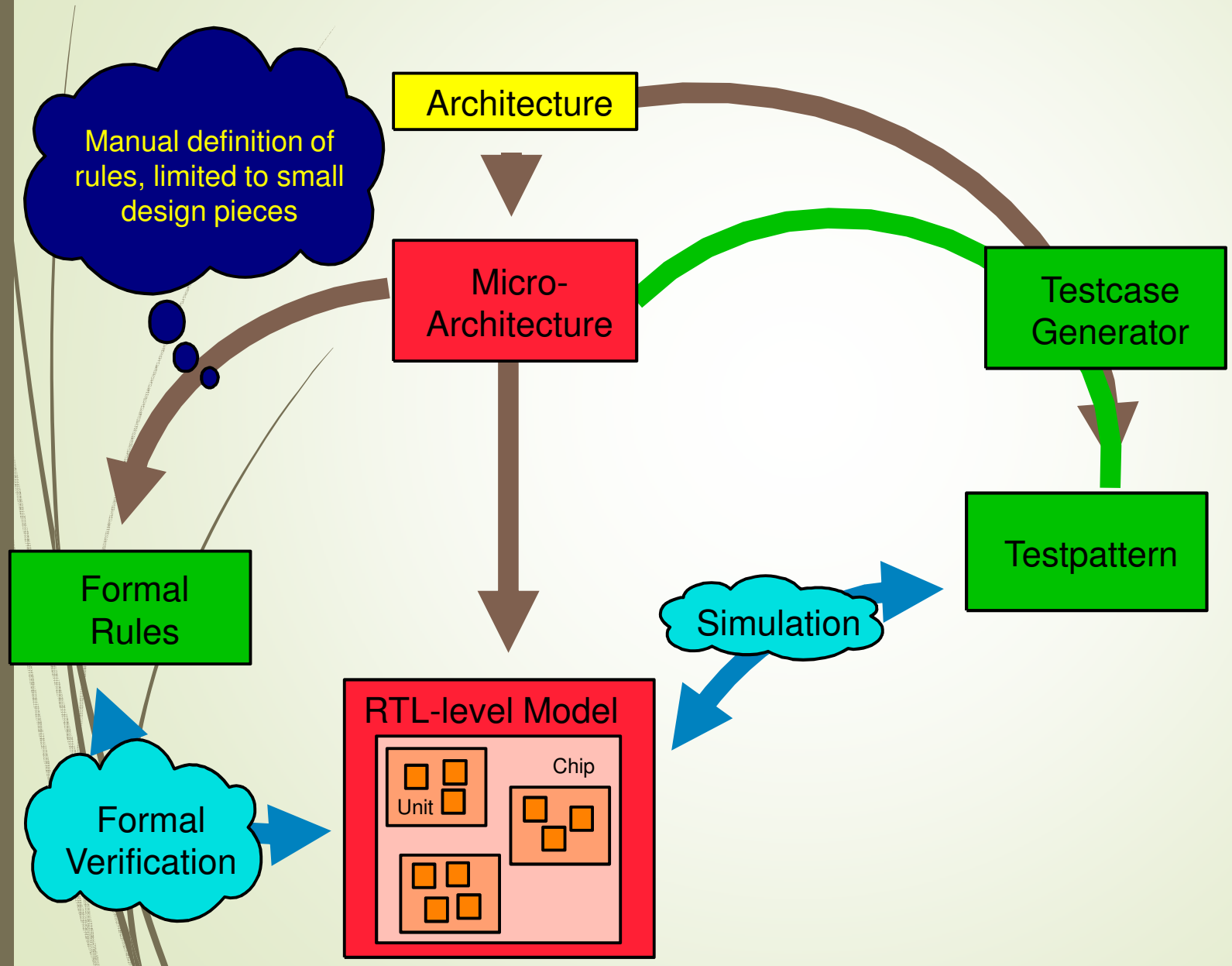
Evolution of Functional Verification



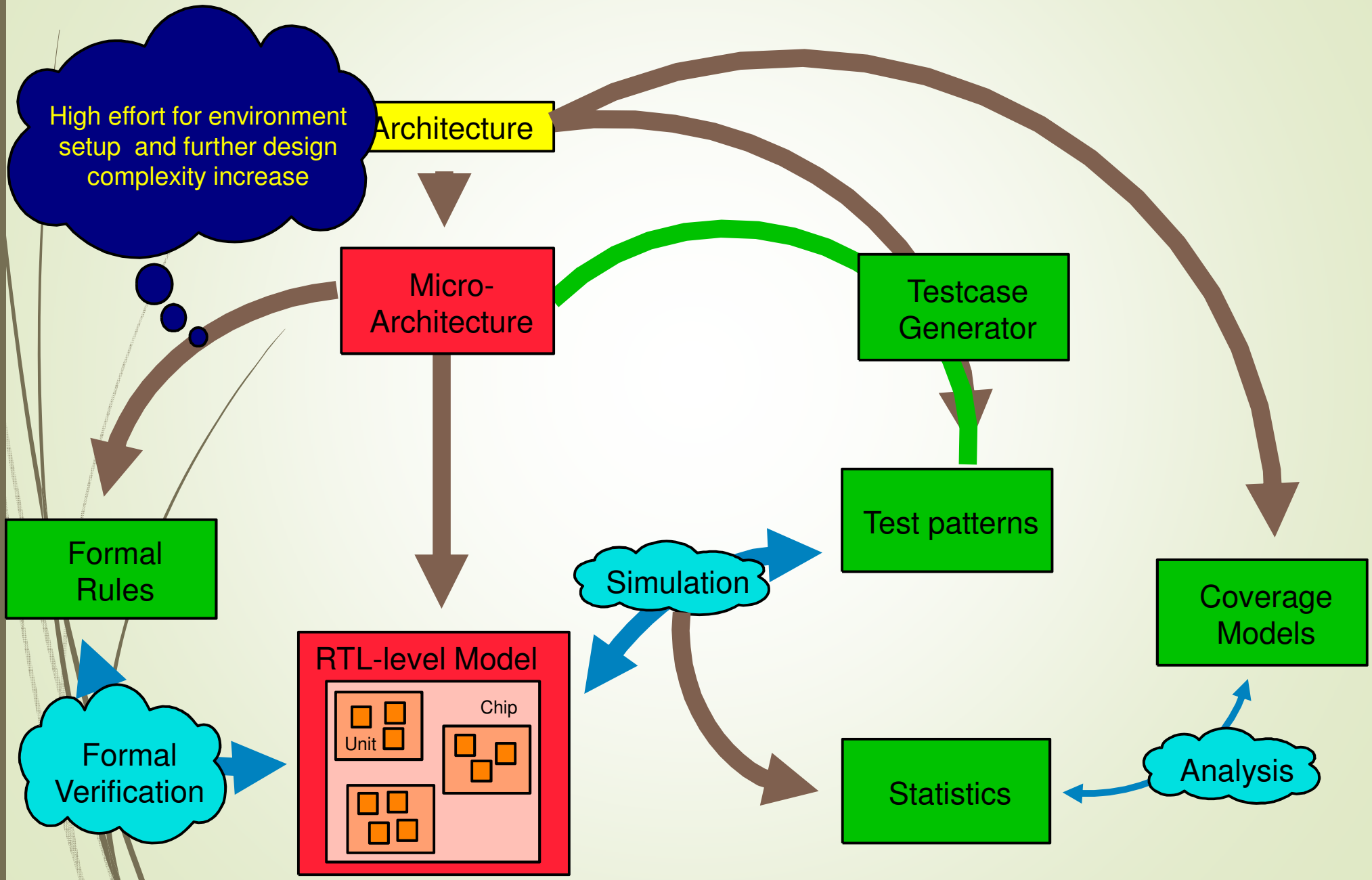
Evolution of Functional Verification



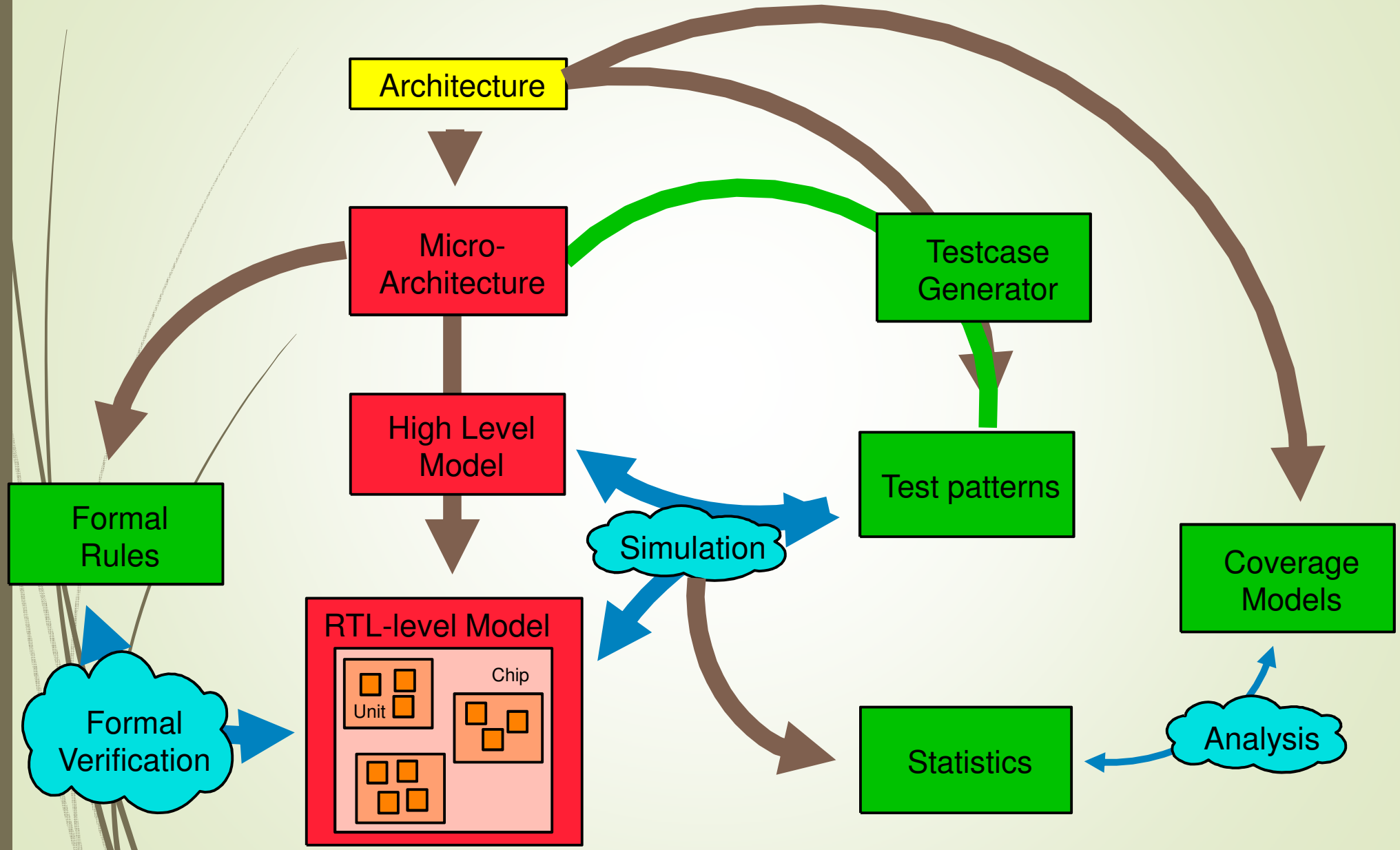
Evolution of Functional Verification



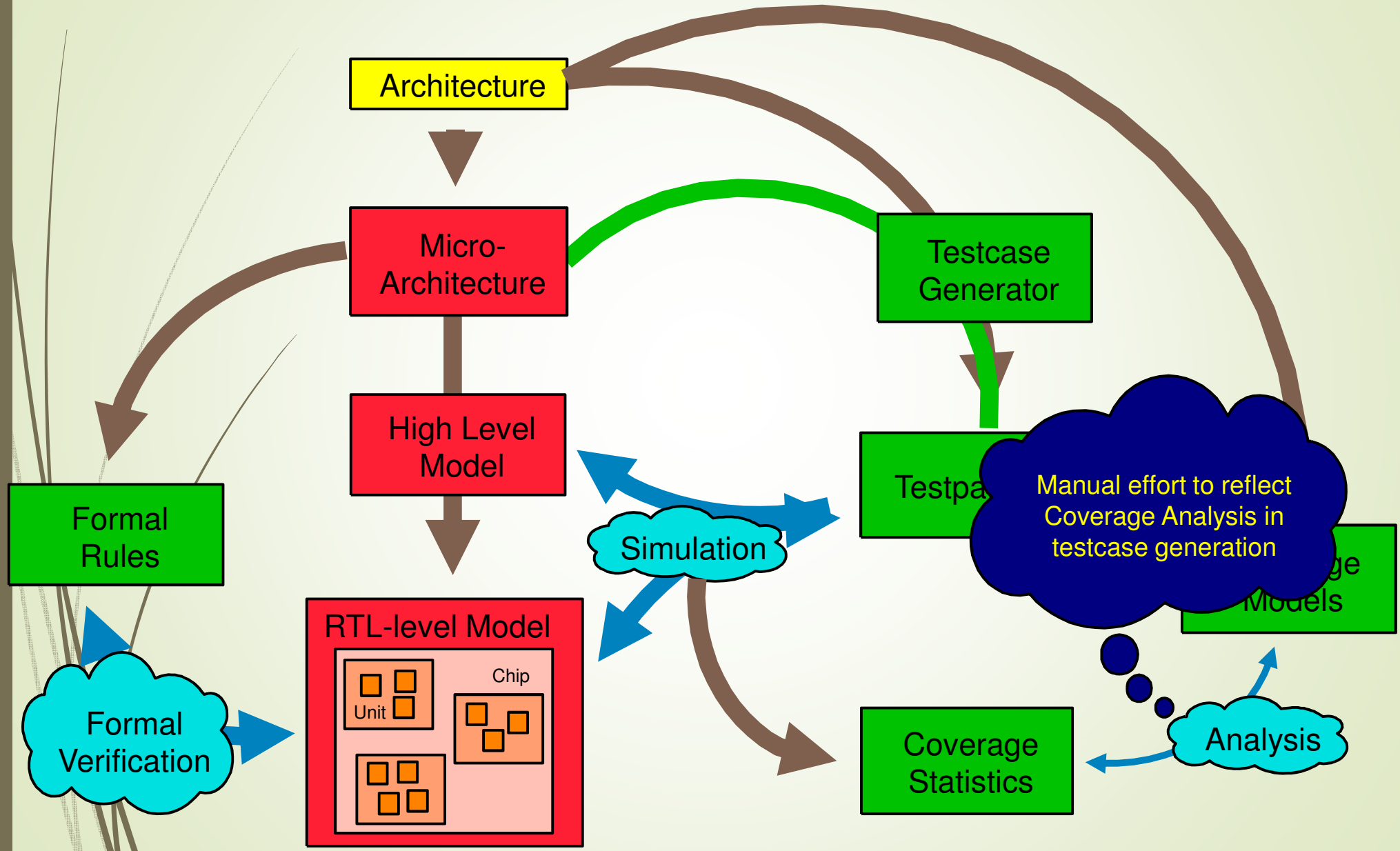
Evolution of Functional Verification



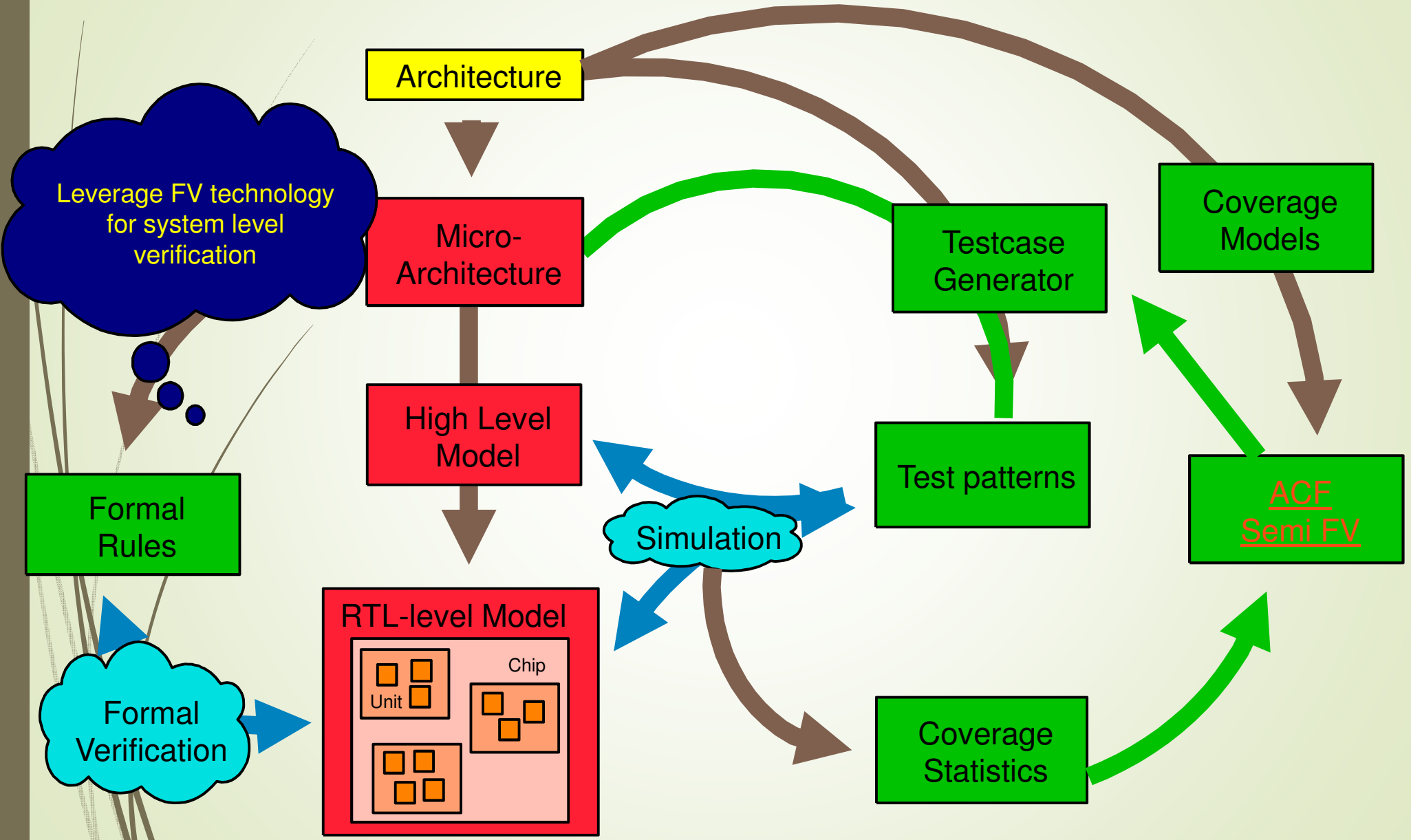
Evolution of Functional Verification



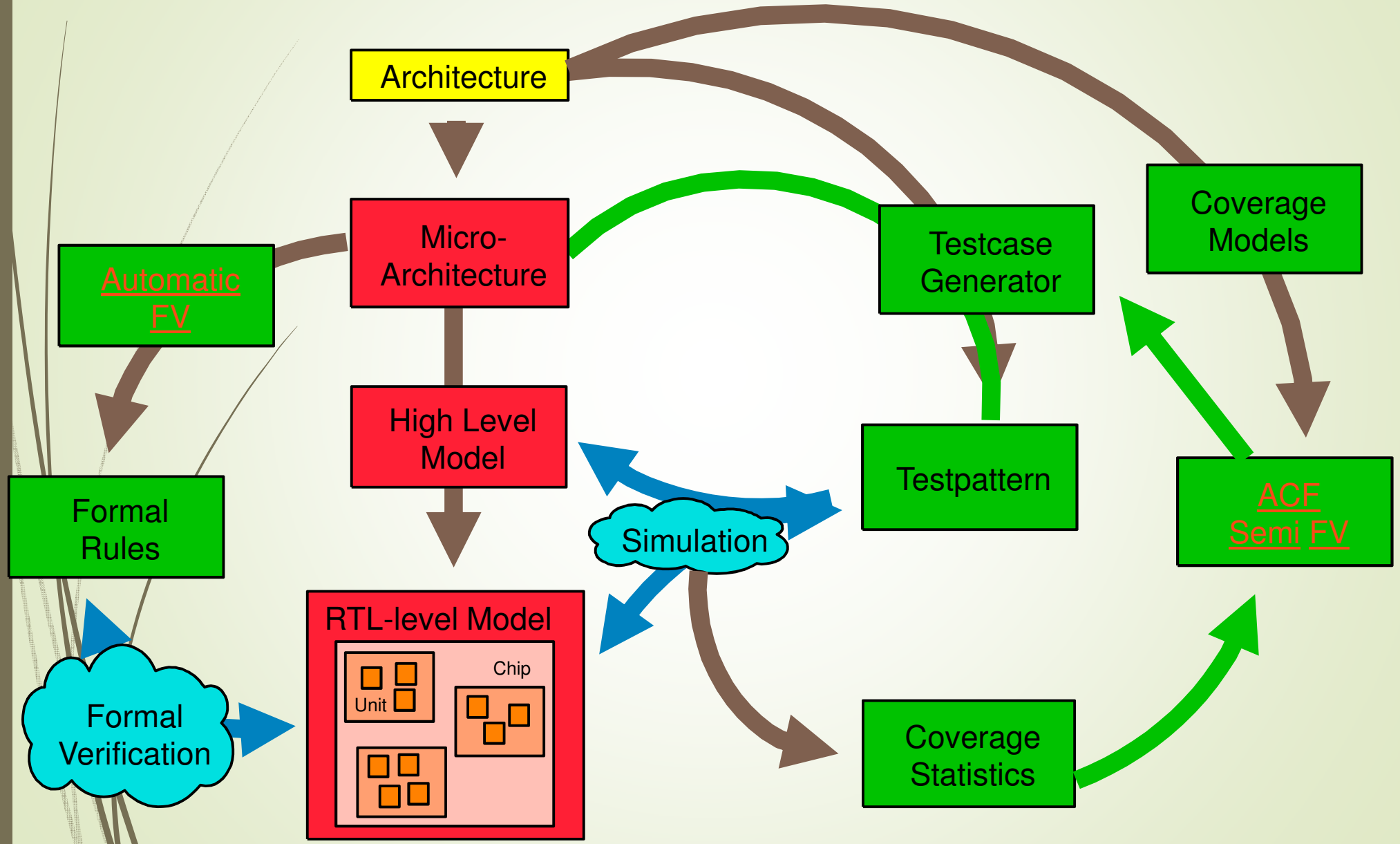
Evolution of Functional Verification



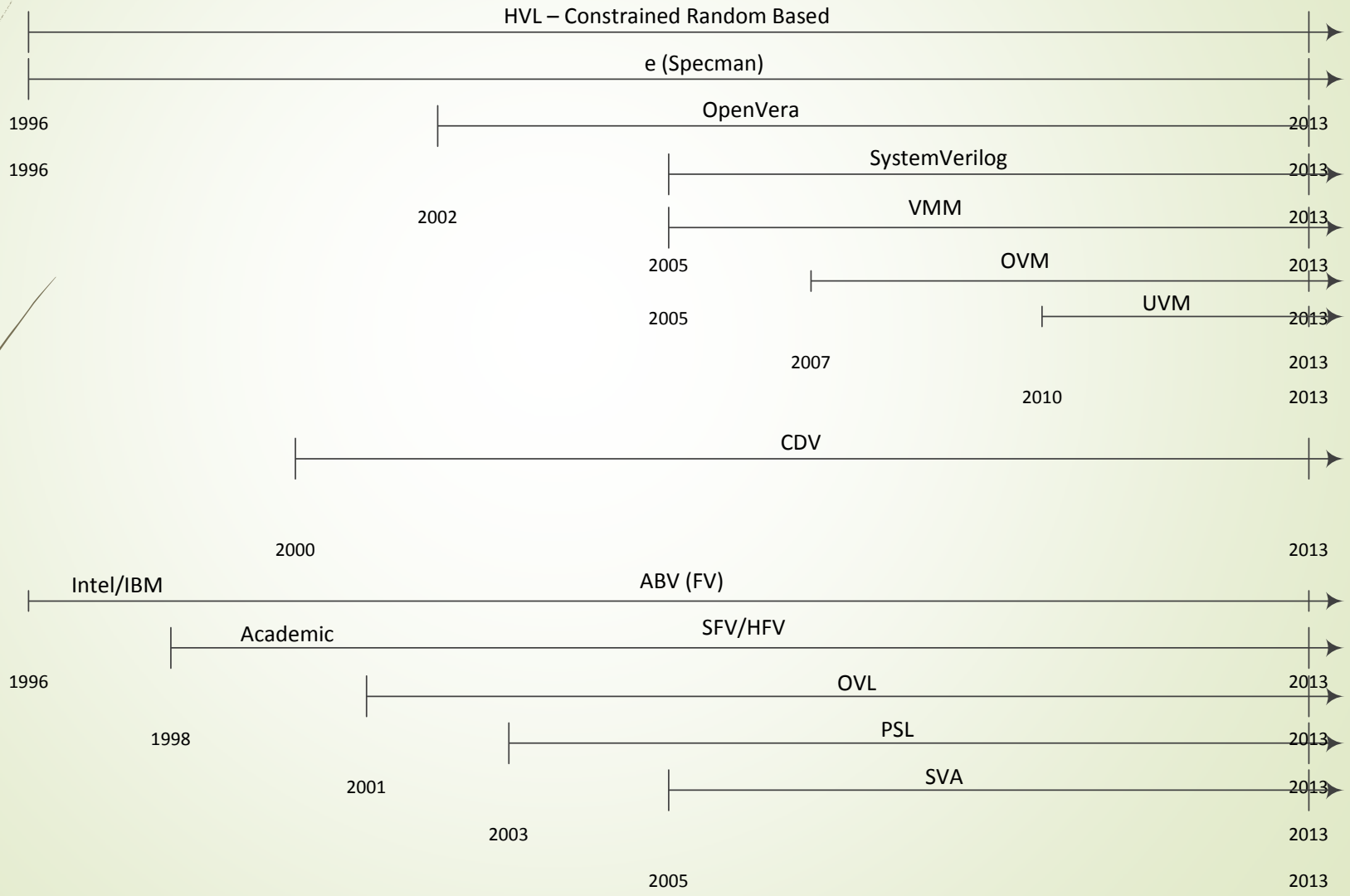
Evolution of Functional Verification



Evolution of Functional Verification



Languages and Methodologies Timeline



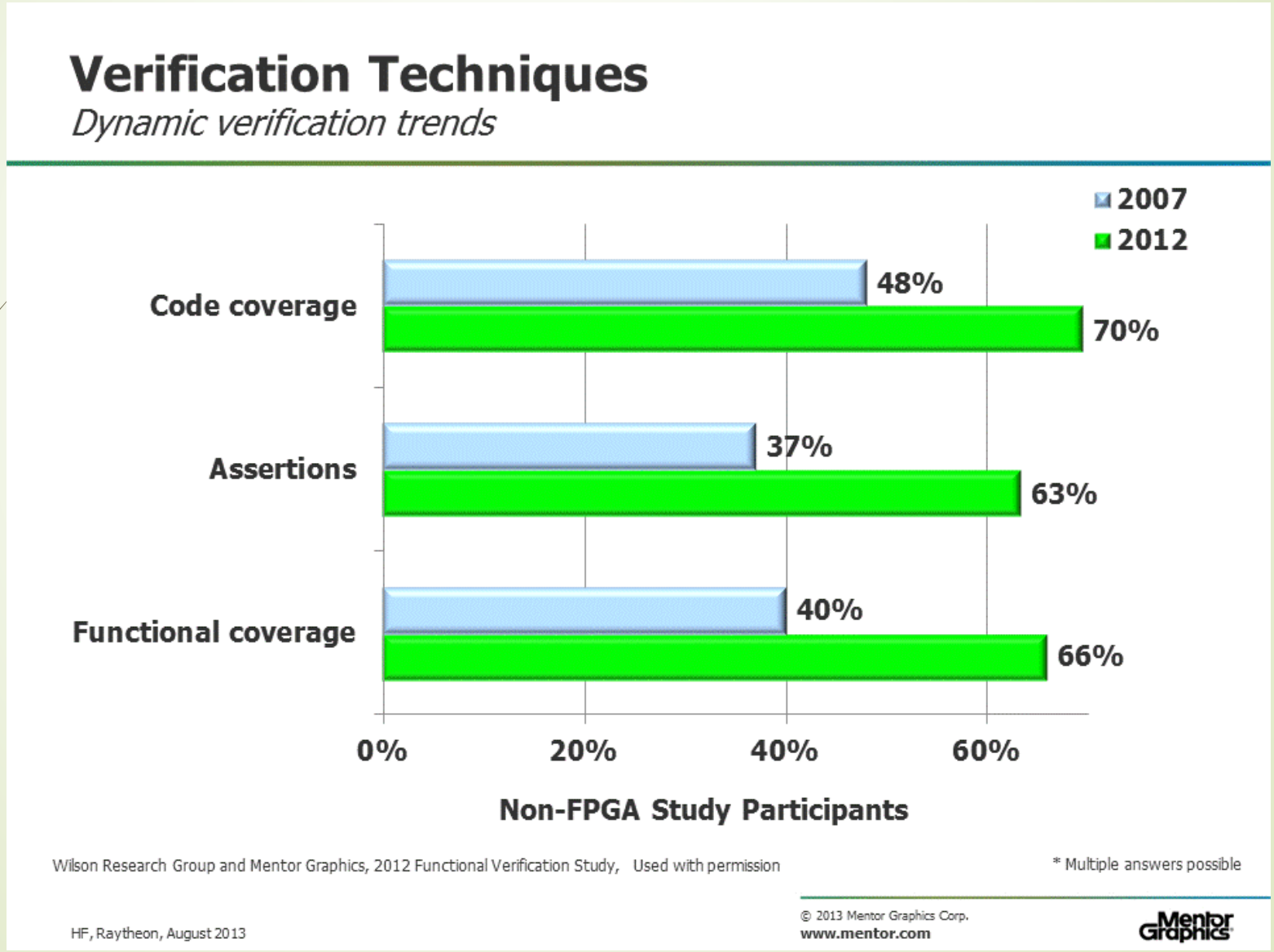
Advanced Techniques Description and Trends

- Coverage-Driven Verification (CDV)
- Formal Verification
  - Active R & D
  - Assertion-Based Verification (ABV)
  - Semi Formal Verification (SFV)
  - Automatic Formal Verification

Coverage-Driven Verification

- Scenarios are monitored by coverage directives
  - Need **much less** test cases
- Unified Coverage accumulation use coverage of simulation and formal tools
- Automatic Coverage Feedback (ACF)
  - Automatic analysis of coverage results alters the test constraints to hit coverage holes

Coverage Techniques Trends



Formal Verification – Active Research and Development

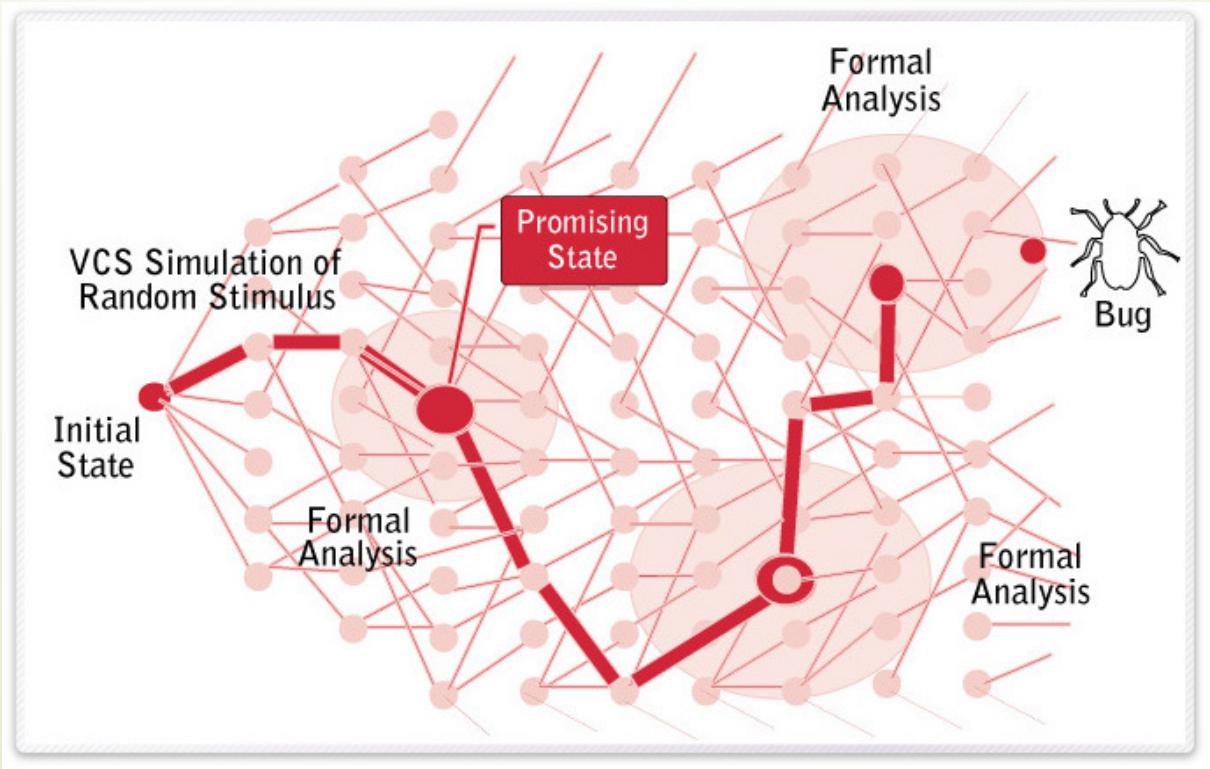
- Assertion-Based Verification
  - Model Checking
  - Bounded Model Checking (BMC)
  - SAT Solvers
  - Symbolic Simulation
  - All Commercial tools use some or all of these techniques
- Semi-Formal Verification
  - IBM SixthSense, Synopsys Magellan
- Automatic Formal Verification
  - Jasper Gold, Cadence Incisive

Assertion-Based Verification

- Bus Protocols
- Block level properties
  - Designers play a significant role!
  - Invariants (always, never, etc.), Arbiter, FIFO, Complex properties
- Arithmetic functions
- Hierarchical verification by using abstractions (black-boxing)
  - Inter block handshake protocols

Semi-Formal Verification

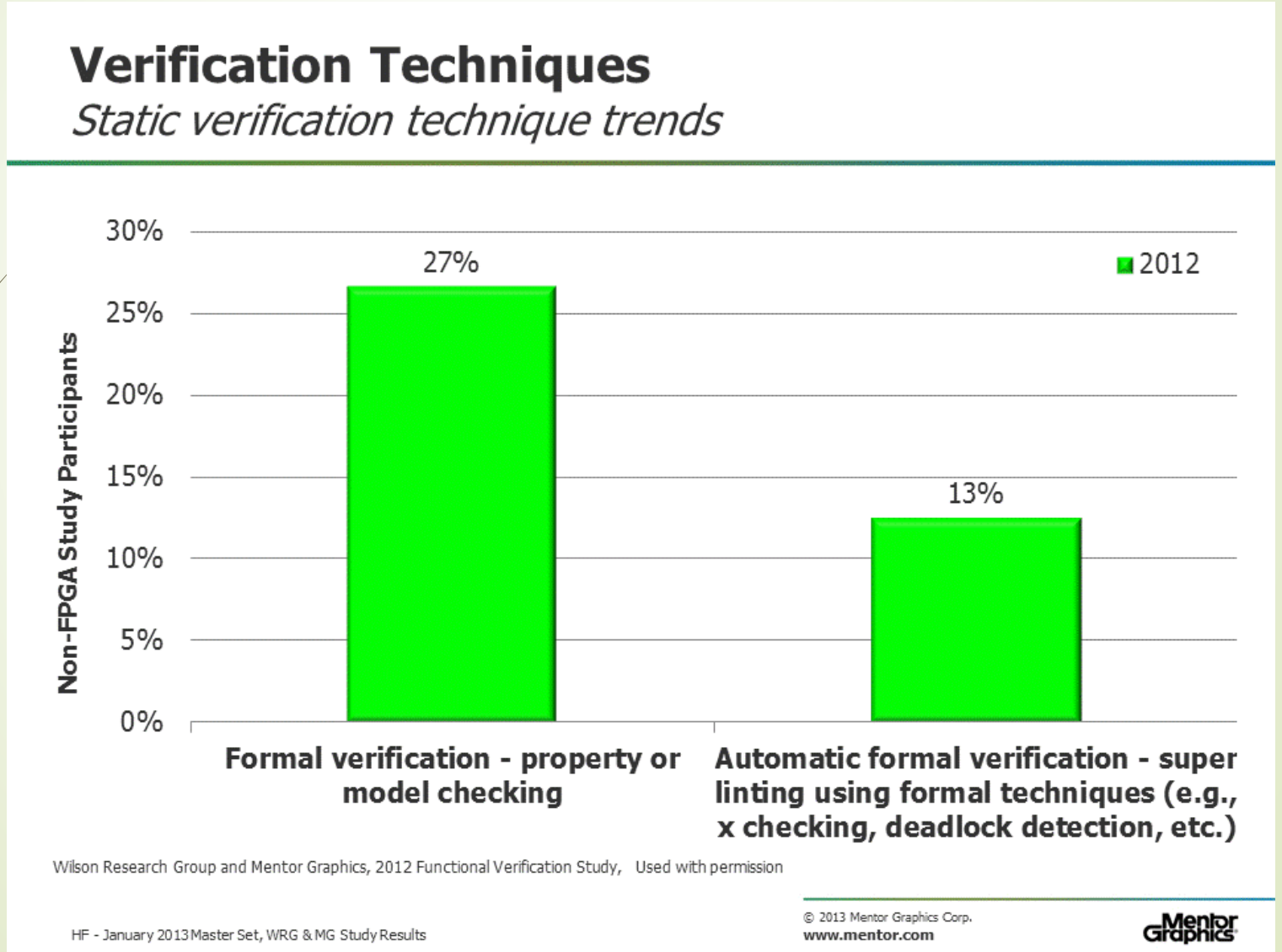
- Using formal-techniques aiming at uncovering bugs (finding a counter-example) instead of achieving a complete proof of all the properties
- Critical for "hard-to-get" bugs
- Key for scaling formal techniques to large designs



Automatic Formal Verification

- Connectivity path verification (at the full-chip level)
- X propagation
- Deadlock detection
- Security violations

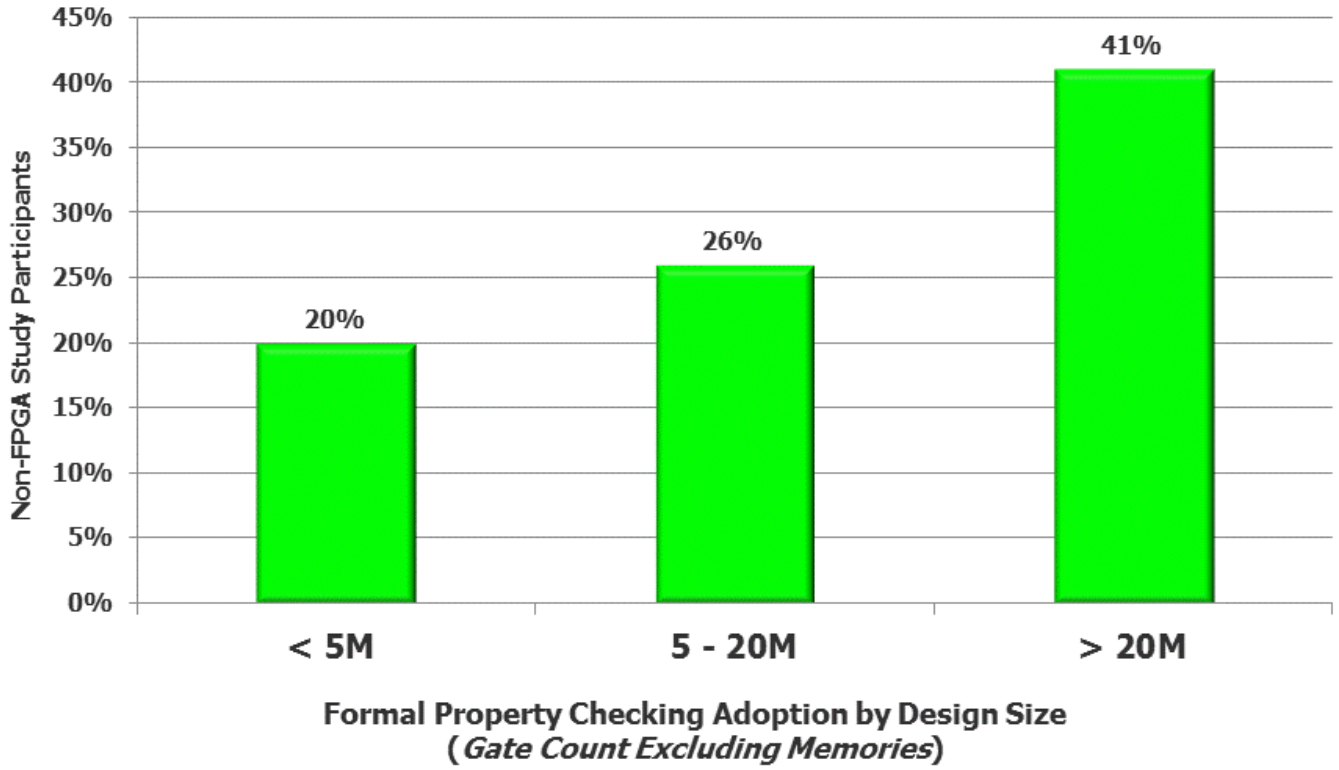
Formal Verification Status



Formal Verification Status - #2

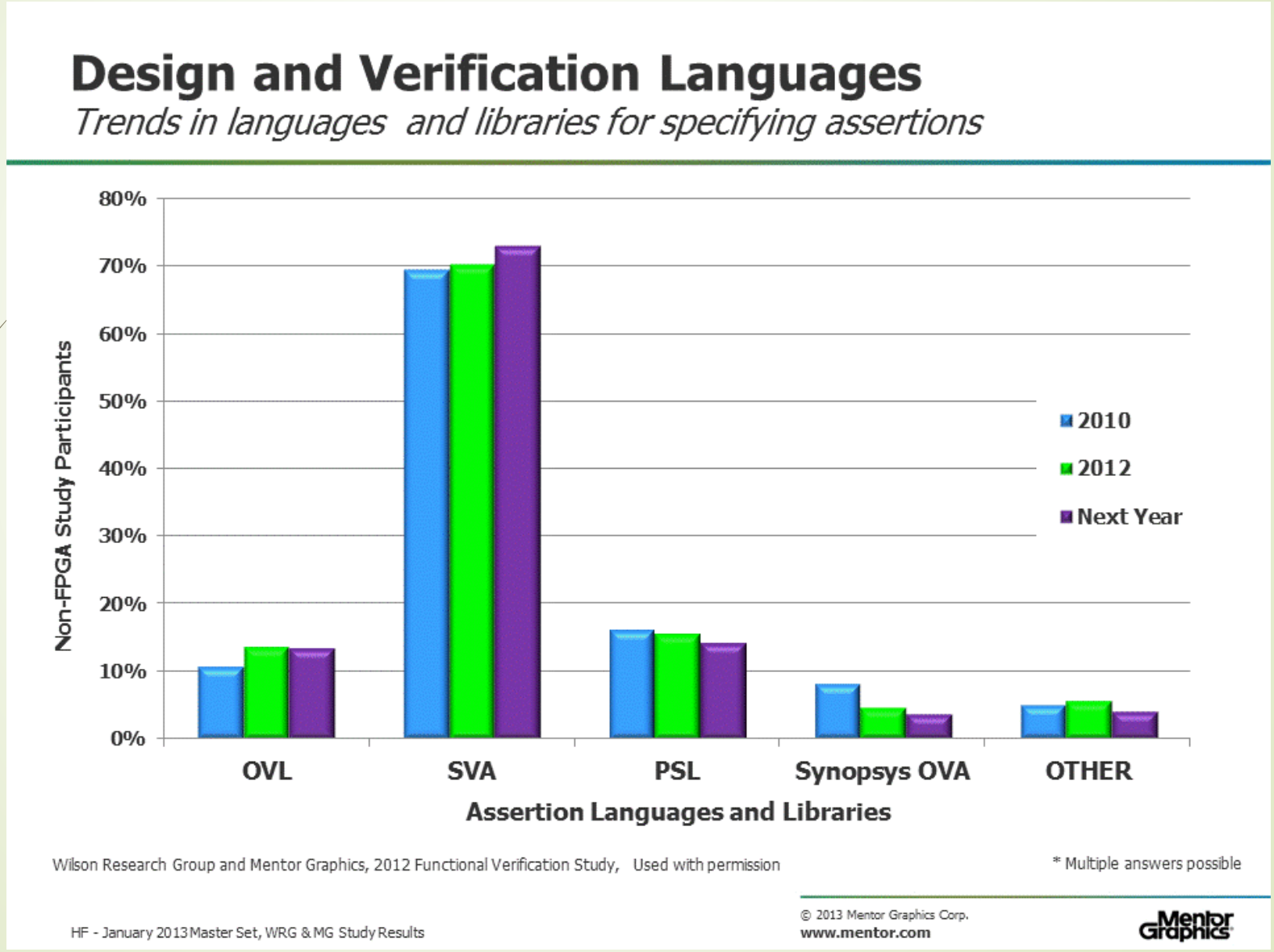
Verification Techniques

Formal property checking adoption by design size



Wilson Research Group and Mentor Graphics, 2012 Functional Verification Study, Used with permission

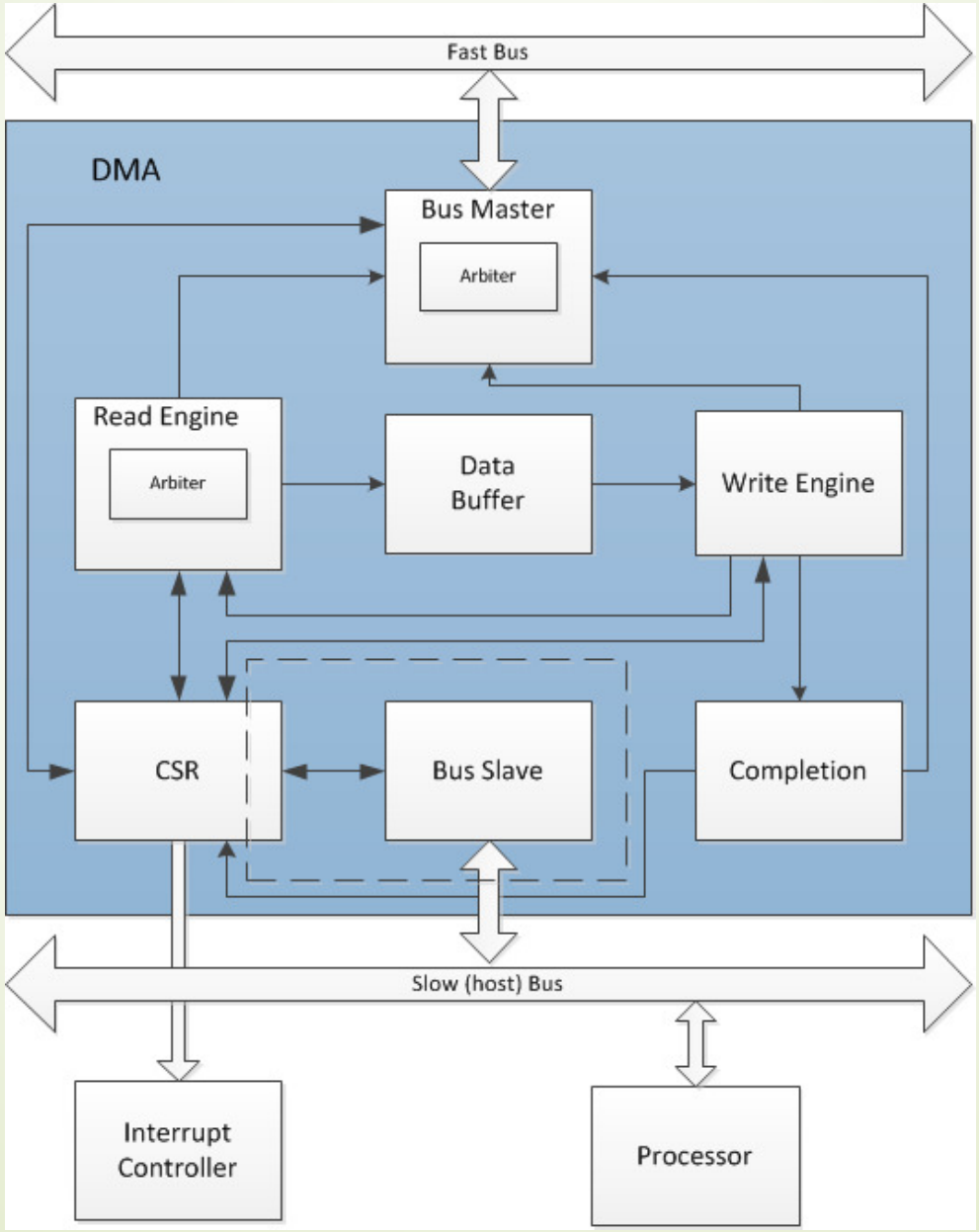
ABV Languages Status



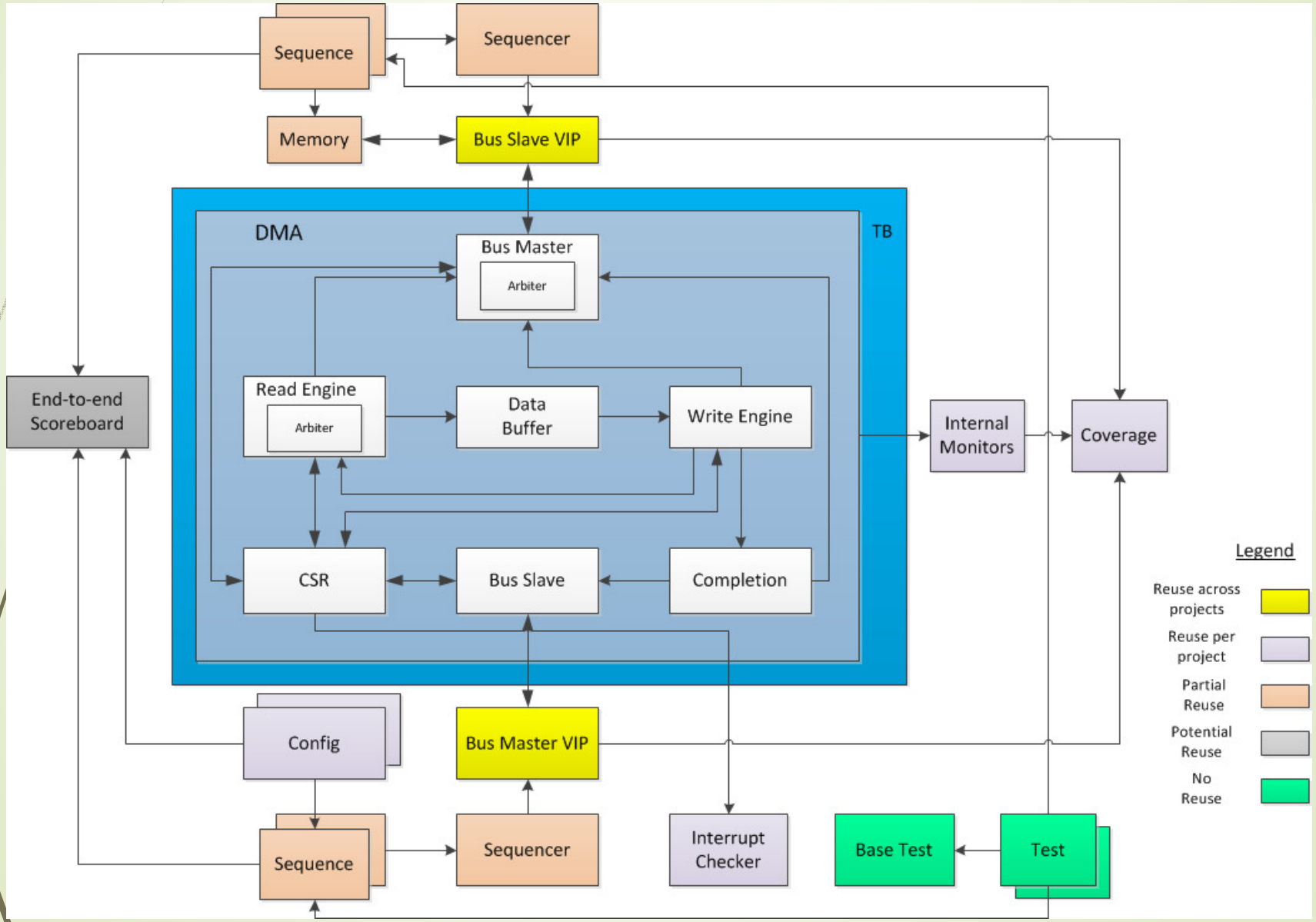
Case Study

- Direct Memory Access (DMA) Core Verification
- 2 Interfaces with different clock speeds
  - Slow interface for host access
  - Fast interface for both descriptors read and the memory access (Data Transfer)
- Multiple active DMA channels with weighted Round-Robin
- Supports Out-of-order response for Data read and write
- Completion notification should be in-order
- Interrupts are used to notify completion and errors

DMA Block Diagram



DMA Verification Environment



DMA Verification Key Points

- Correct Behavior Checking
  - Registers Access
  - Data transfer
  - Completion Interrupts ordering
  - Error Interrupts
  - Performance with all possible slave latencies and OOO responses
- Coverage Directives
  - All control and status register fields exercised
  - Out-of-order is exercised
  - Clock-domain crossing contentions
  - All configuration legal combinations
  - Back-pressure scenarios
  - Performance parameters

Methodology Key Points

- Methodology
  - UVM with in-house guidelines
  - All functionality is implemented in the environment
  - Use Macros (logging and factory)
- Test writing
  - When possible, use only factory overrides

DMA Verification Environment

- Did I forget anything?
- Formal Verification
  - Arbiters
  - FIFOs
  - Bus Interfaces
  - Handshake protocols
- Automatic Coverage Feedback
  - Test cases can be altered with additional constraints extracted by coverage analysis tools
- Semi-Formal Verification
  - Light-houses: Use simulation to reach high risk states and then use formal techniques to cover the state space around it exhaustively

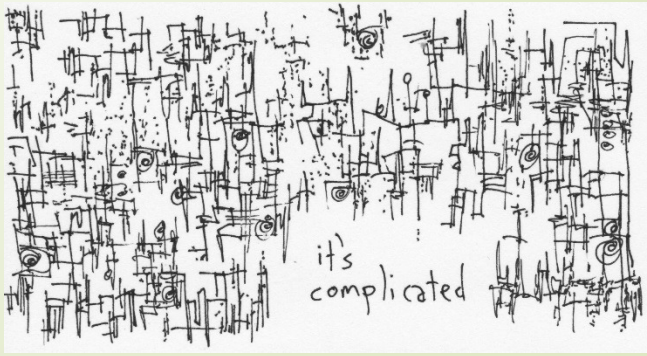
DMA Environment Integration to Higher Hierarchies

- Whenever possible use bus monitors and other interfaces (instead of looking at environment constructs)
  - Coverage/Interrupt checker should obey this guideline
- Host bus interface VIP could be replaced by the actual host
  - Host registers access sequences should be replaced by SW/FW that performs the same functionality as the sequences
  - Configurations are reconstructed by monitoring the host bus
- Fast bus interface VIP could be replaced by the actual slave architecture
  - Affects end-to-end scoreboard – different memory models
- This concludes are DMA case study


4 Challenges in Verification

4 Challenges in Verification

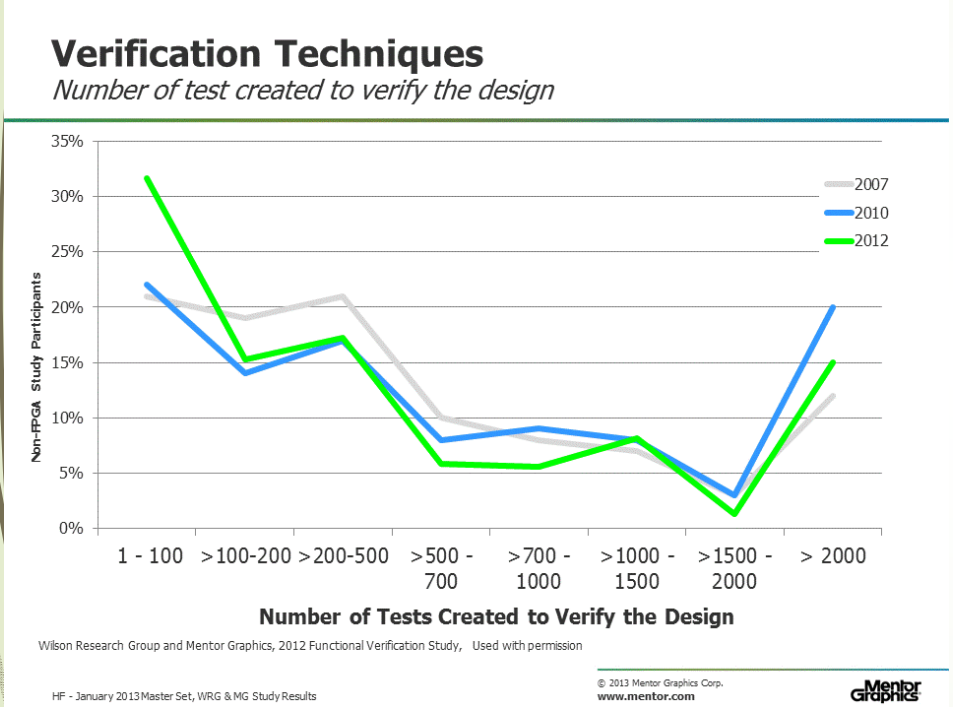
- **Mastering Verification Complexity**
  - Continuous increase in number of IP's and embedded processors
    - 2006: 30-40 IP's, 1 CPU
    - 2011: 80+ IP's, 6+ CPU's
    - *2014: 120+ IP's, 20+ CPU's*
  - The more IP's the higher the risk of late spec & implementation changes
  - Driving towards true Hw/Sw Co-Verification
  - Reuse of verification environments / stimulus from IP-level into big multi-CPU SoC environments
  - Reuse across projects

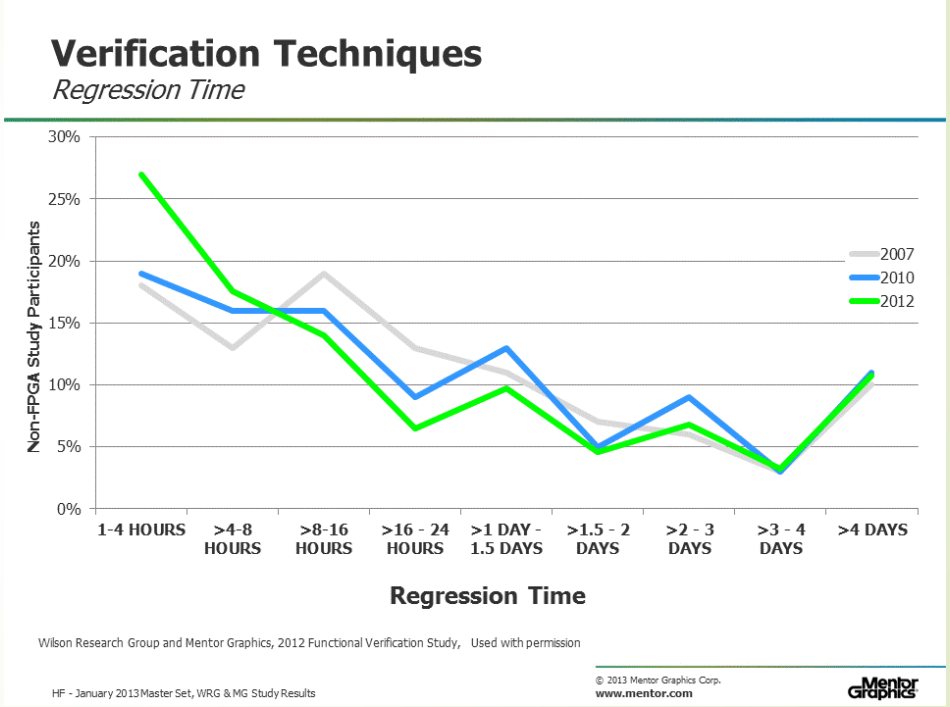


4 Challenges in Verification

- **Completeness**
  - When are we done?
    - Number of tests, Coverage results, Bugs convergence curve
  - Specification/Requirements Based Verification
    - Use e.g. UML (Unified Modeling Language) to write requirements and testplans
    - Is it feasible?
  - Review process
    - How many reviews are conducted?
    - What are the contents of each review?
  - Standard Checklists

Tests and Regressions

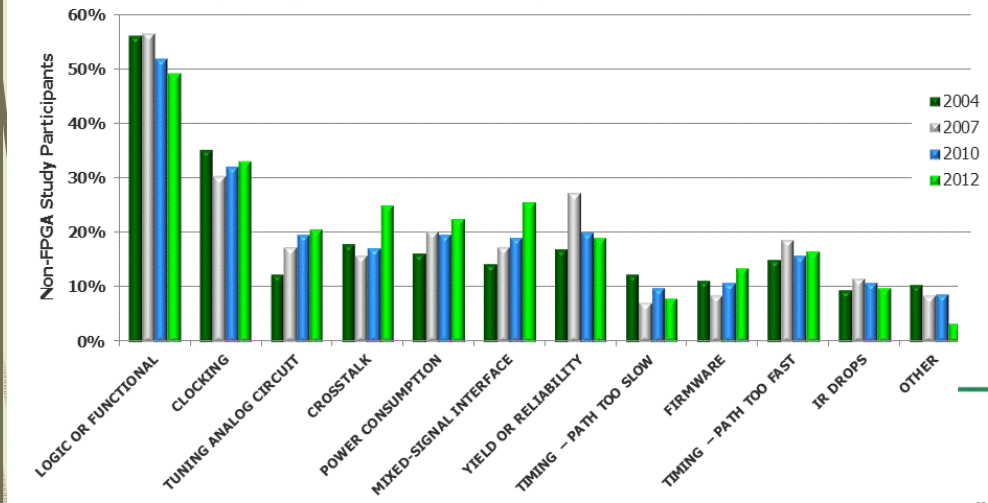




Faults

Effort and Results

Trends: Types of Flaws



Trends in Types of Flaws Resulting in Respins

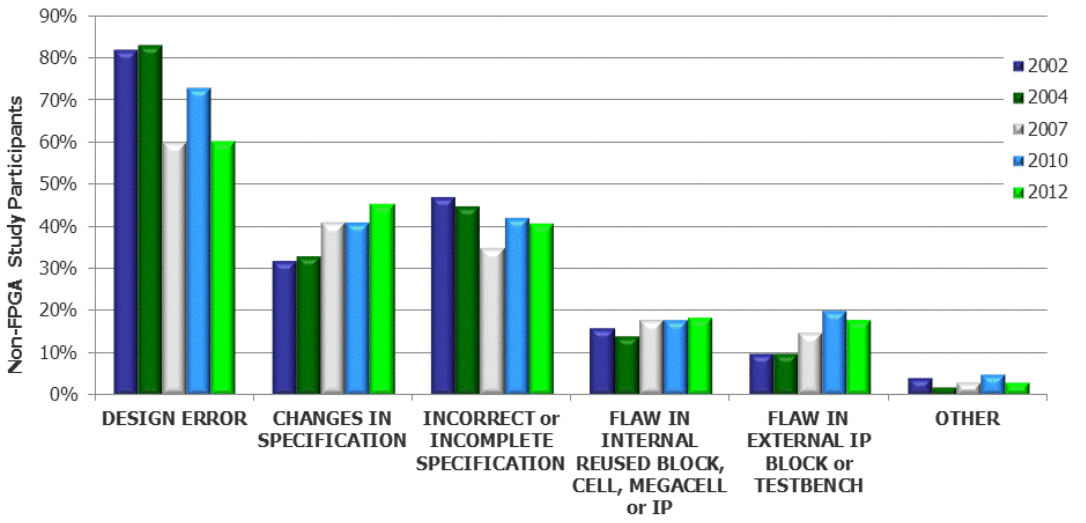
Wilson Research Group and Mentor Graphics, 2012 Functional Verification Study, Used with permission

* Multiple answers po

HF - January 2013 Master Set, WRG & MG Study Results

Effort and Results

Trends: Root Cause of Functional Flaw in Non-FPGA



Root Cause of Functional Flaws

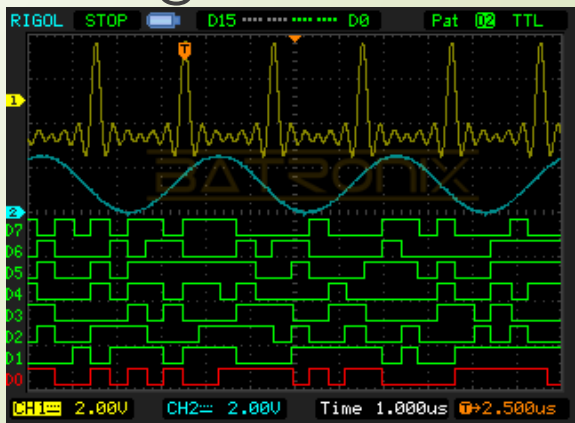
Wilson Research Group and Mentor Graphics, 2012 Functional Verification Study, Used with permission

* Multiple answers possible

HF - January 2013 Master Set, WRG & MG Study Results

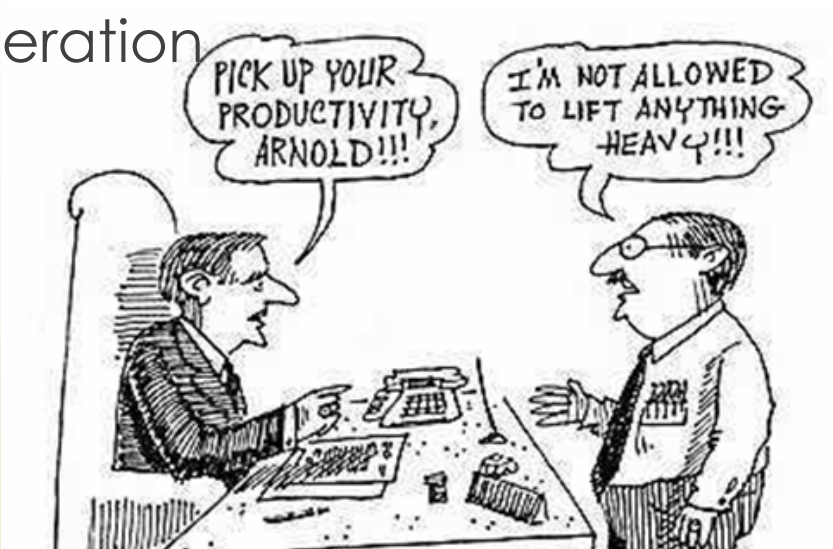
4 Challenges in Verification

- **Mixed Signal Verification**
  - Simulation performance improvement
  - Simulation model creation
  - Unified verification methodology covering analog and digital design parts
  - Verifying logic connectivity including low-power verification
  - Common language between analog and digital/verification engineers
  - Accurate power analysis



4 Challenges in Verification

- **Productivity**
  - Clean integration between formal techniques and simulation
  - Unified coverage database
  - Faster debug
    - New advances in simulators should become ubiquitous
  - Post-Silicon Debug acceleration



Future Verification Trends

- Sharing of Verification IP (Reusability)
  - Integration becomes seamless
  - C Code and HVL sequences are purely interchangeable
- HW/SW Co-verification
- Integration of Analog world into functional verification
- Formally verifying security features, firmware (Bob Bentley Intel)
- Do symbolic analysis of analog circuits (Bob Bentley Intel)
- **Approaching 20 years since the inception of the e Language (1st HVL)**
  - Are we on the verge of a new era?

Summary

- We presented:
  - Functional Verification Flow
  - Present Verification techniques and methodologies
  - A case study of a DMA
  - 4 Challenges in today's Verification activities
  - Future Trends

Thank You