2023 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

SAN JOSE, CA, USA
FEBRUARY 27-MARCH 2, 2023

A Wholistic Approach to Optimizing Your System Verification Flow

Ross Dickson - Lance Tamura - Michael Young

Cadence Design Systems

cadence®

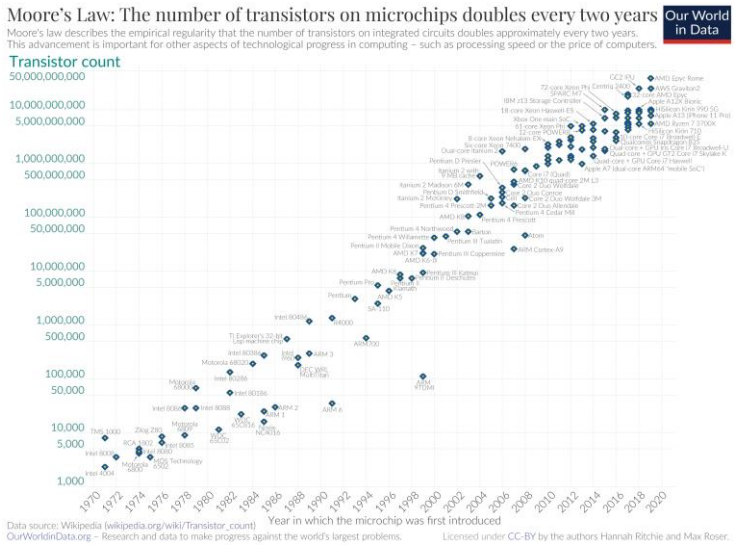
accellera SYSTEMS INITIATIVE

Agenda

- Introduction
- Reviewing The Verification Flow
- Reviewing Current Solutions
  - Hardware, Interfaces, and Extensions
- Why Should We Change?
- A Comprehensive Solution
- Summary and Conclusion

Moore’s Law – Costs, Effort, and Graphs go up and to the right

- Verification Costs and Effort
  - Track the growth in transistor count
  - Increase exponentially
- This trend shows no signs of ending



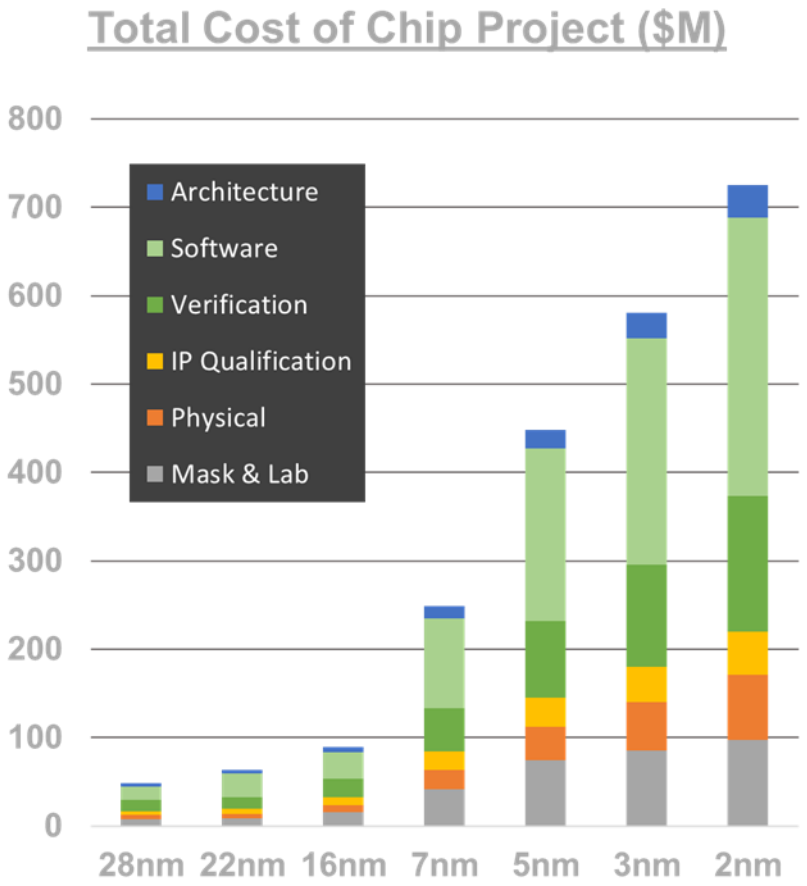
Max Roser, Hannah Ritchie Via Wikipedia
https://ourworldindata.org/uploads/2020/11/Transistor-Count-over-time.png

New Tools Have Always Been the Solution

- Every technology driven increase in complexity has been followed by a new tool generation
  - Delivering an exponential increase in productivity
- Net result
  - Cost and Effort only increase linearly

Now with Software

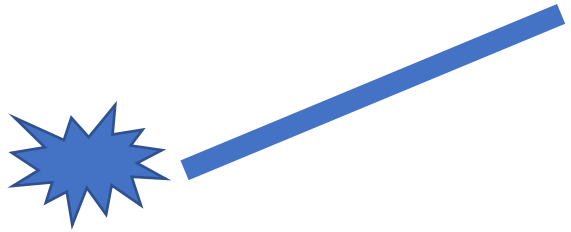
- Software is an increasingly important part of our products
- Software complexity at least as fast as the transistor count
  - Now we have two exponentials
- This trends has also been true for decades



IBS Projection: Cost per Node

Stop Focusing on Filling Gaps

- Accept that designing and verifying complex modern systems will always take work
- Realize that our industry and conference aren’t likely to go away any time soon
- Spend some time looking for the most effective way to get our jobs done and most of our time actually doing our jobs



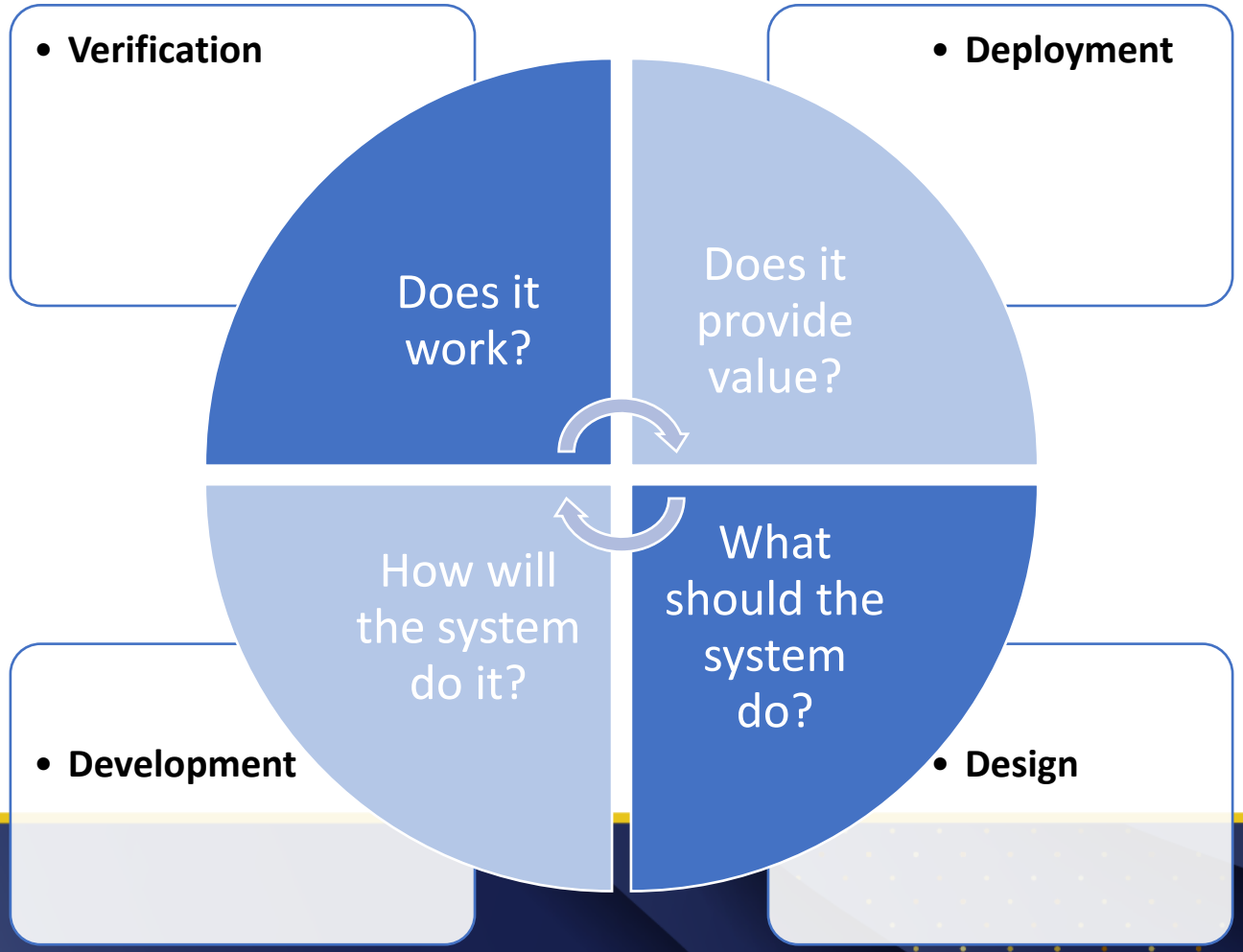
2023 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

SAN JOSE, CA, USA
FEBRUARY 27-MARCH 2, 2023

End Keynote, Begin Content

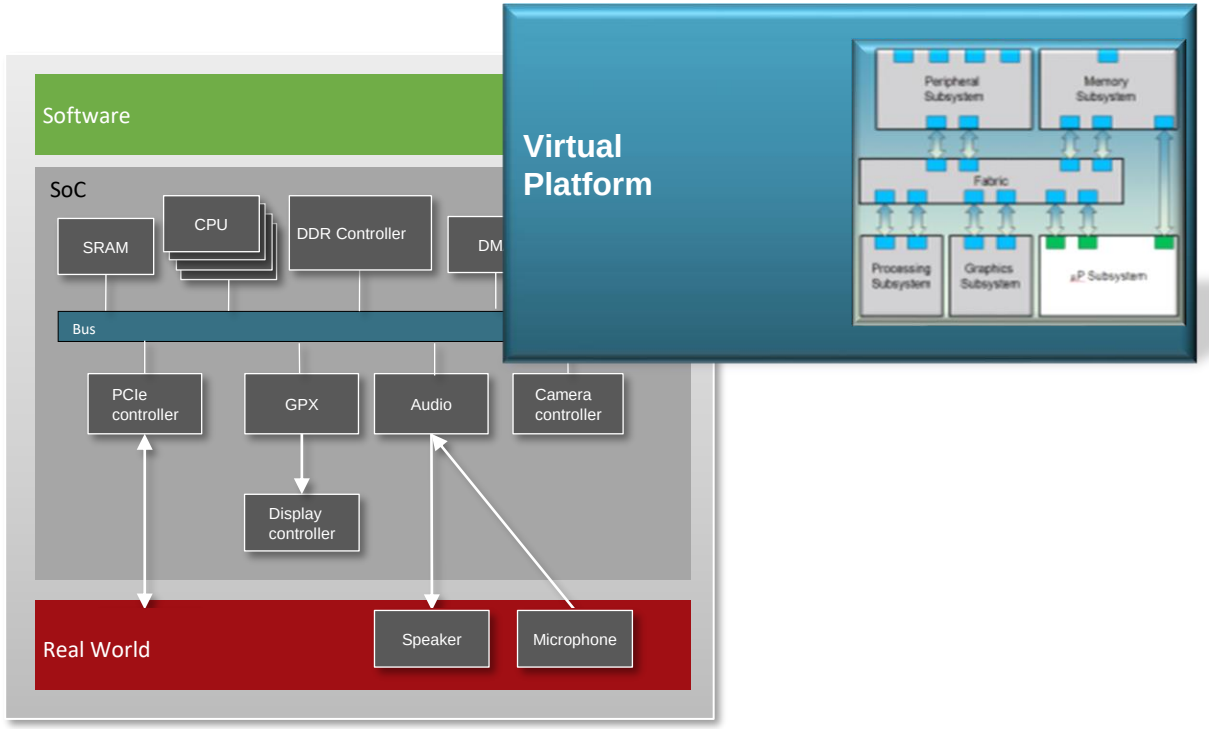
accellera SYSTEMS INITIATIVE

Reviewing The Key Question of the Verification and Design Flow



Does it Work? The Traditional Approach

- Decompose the system into manageable components
- Re-use tests, optimize for runtime
- Generally independent of the design and software teams
- Delay system testing until first silicon, or “shift left” with unrelated tools

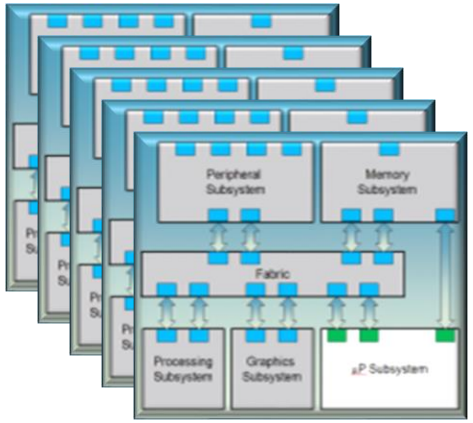


Does it provide Value?

- Solving customer support is out of scope today

What should the (next) system do? Traditionally

- Start with the previous design
- Turn the crank
- Traditionally independent of Verification



How will the system do it?

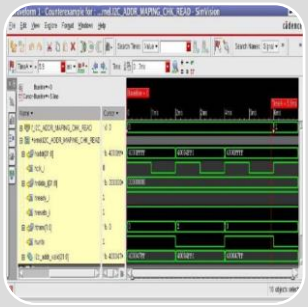
- Learn more at implementation focused session at one of Cadence's Live Technology and Product shows

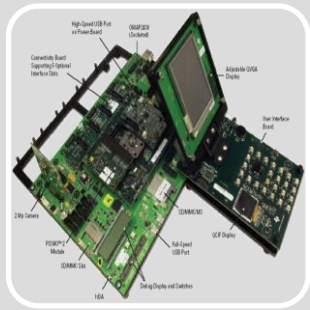
Agenda

- Introduction
- Reviewing The Verification Flow
- Reviewing Current Solutions
  - Hardware, Interfaces, and Extensions
- Why Should We Change?
- A Comprehensive Solution
- Summary and Conclusion

There is no “one size fits all” engine





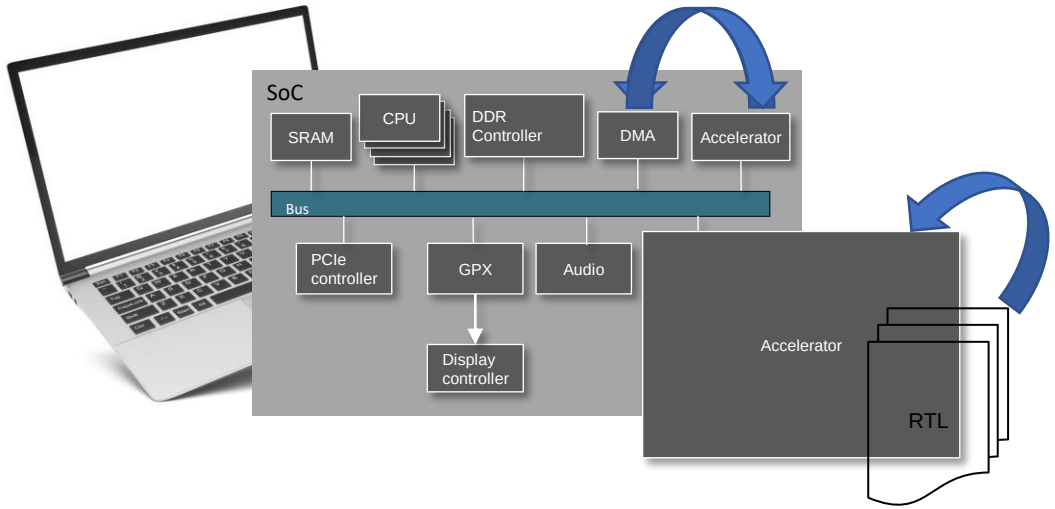
SDK OS Simulation	Virtual Platform	Formal Analysis	HDL Simulation	Emulation	Enterprise Prototype	Prototype Board
•Speed •HW Detail	•SW Debug •HW Detail	•Great for IP •No software execution	•HW debug •Speed	•RTL accurate •HW debug •Fast Compile •MHz range •Cost	•RTL accurate •Up to 10 MHz •OK to debug •After RTL is stable	•10s of MHz •RTL accurate, but capacity may limit design content •Limited debug •Long dev time

Earlier availability

Better accuracy

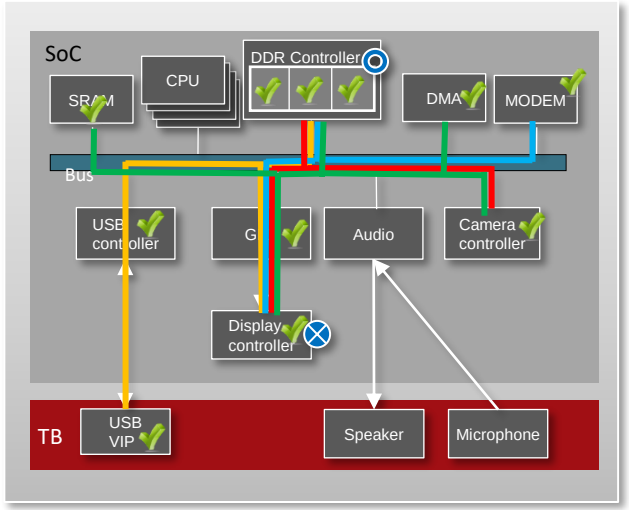
Hardware Verification Focuses on RTL

- HW architectural spec -> implementation-spec -> RTL
- HW verification checks RTL
- RTL tested vs implementation spec



Limits Of Hardware Verification

- 100% HW verification is almost never achieved
  - Complexity often overwhelms schedule and resources
  - Functional coverage may be limited
  - RTL line coverage checks that all logic is touched – but without context
  - RTL code linting can avoid syntax and sematic issues
- HW verification does not directly check architectural spec for correctness/completeness



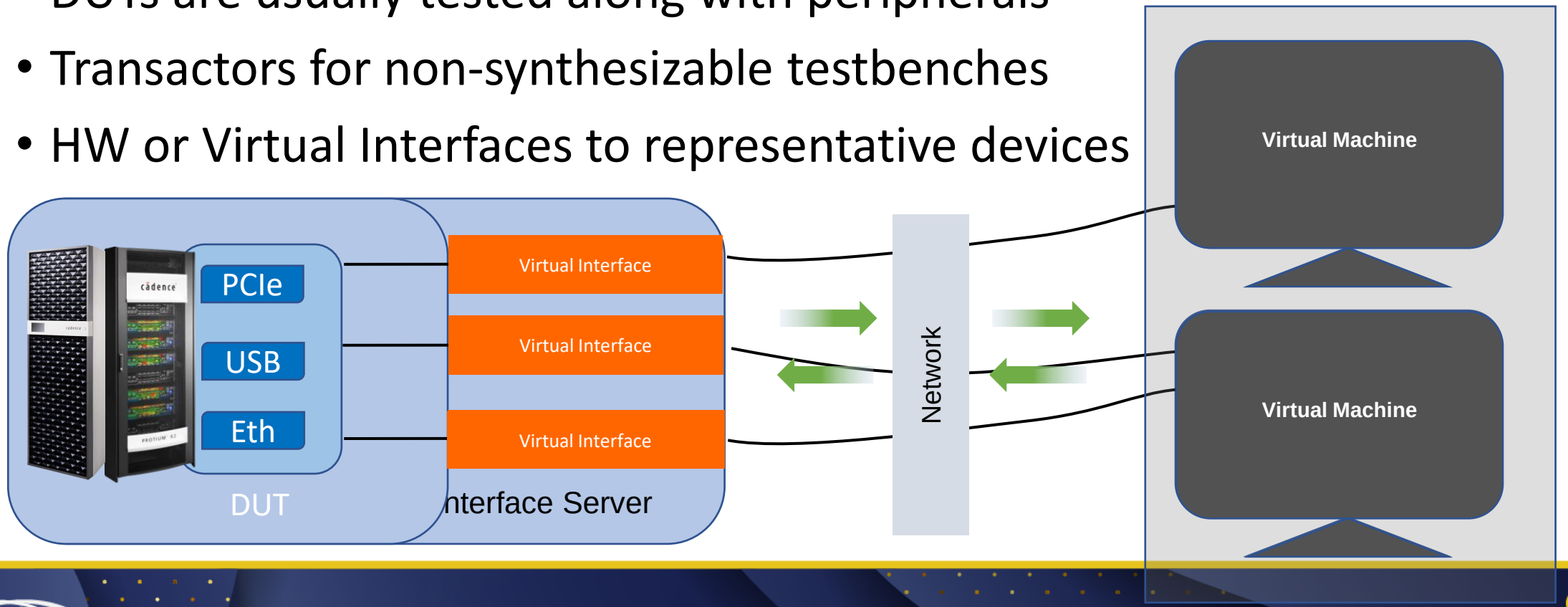
Addressing Complexity Using Hardware Assisted Verification Platforms

- There are many opinions on what role hardware assisted verification platforms play in the verification and design lifecycle
- Given the costs and expenses, it must be front and center in our planning

- Because it's the only thing fast enough to run the software
- And the only practical way to run large designs

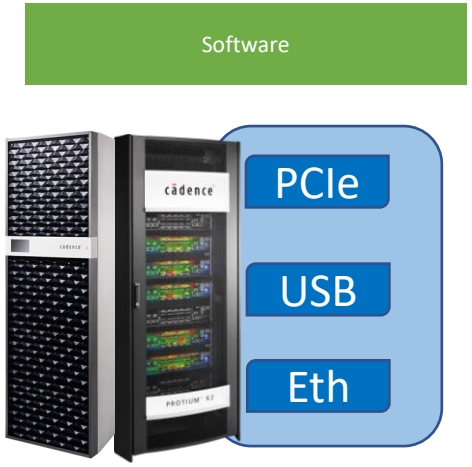
Expanding HW Platforms with Interfaces

- DUTs are usually tested along with peripherals
- Transactors for non-synthesizable testbenches
- HW or Virtual Interfaces to representative devices



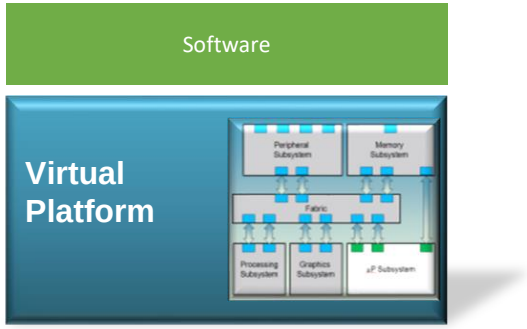
HW Platforms Used For SW Development

- HW Speed is essential, both for runtime and platform development
  - Simulation -> Emulation -> Hybrid -> FPGA
- Usages include
  - Driver and Firmware Development and Test
  - Performance measurement
- Assumes HW is correct
  - Aim is to test SW
  - Often accidentally catch important bugs



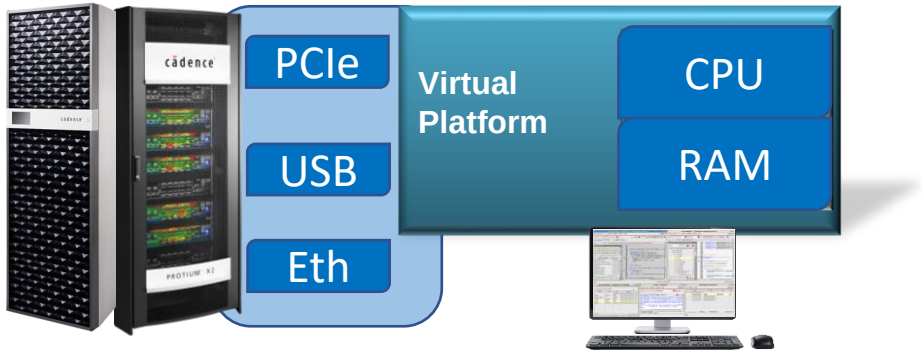
Virtual Platforms for SW Development

- Assumes the HW is someone else's problem
- Assumes architectural spec matches the RTL
- Usage not generally considered Verification

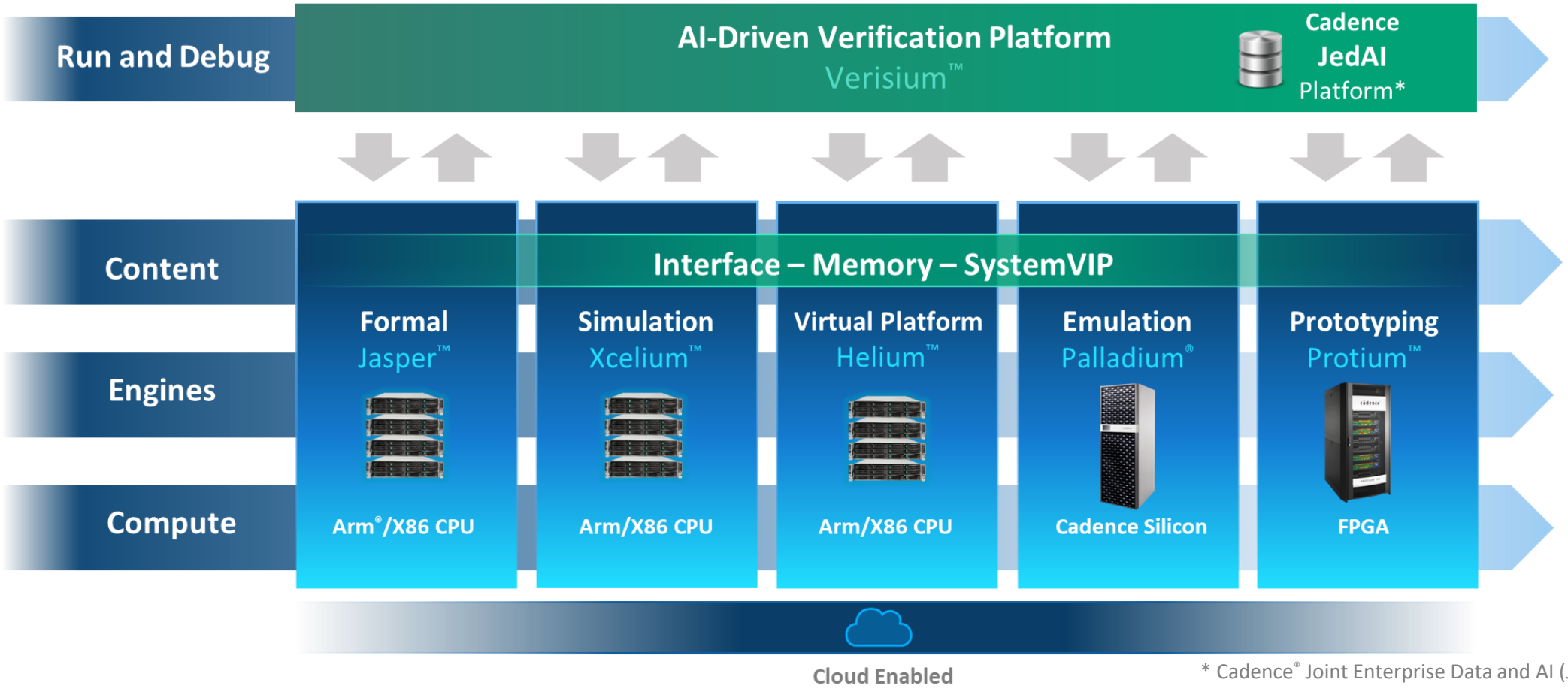


Hybrid HW / Virtual Platforms

- Assume part of the HW is already verified
  - Offload the verified portion (such as the CPU) to a virtual platform
- Usage include both Software Development and Hardware Verification
  - Higher performance improves upon HW for SW Development
  - RTL rather than models enables actual verification and ensures correlation



Cadence Verification Full Flow



* Cadence® Joint Enterprise Data and AI (JedAI) Platform

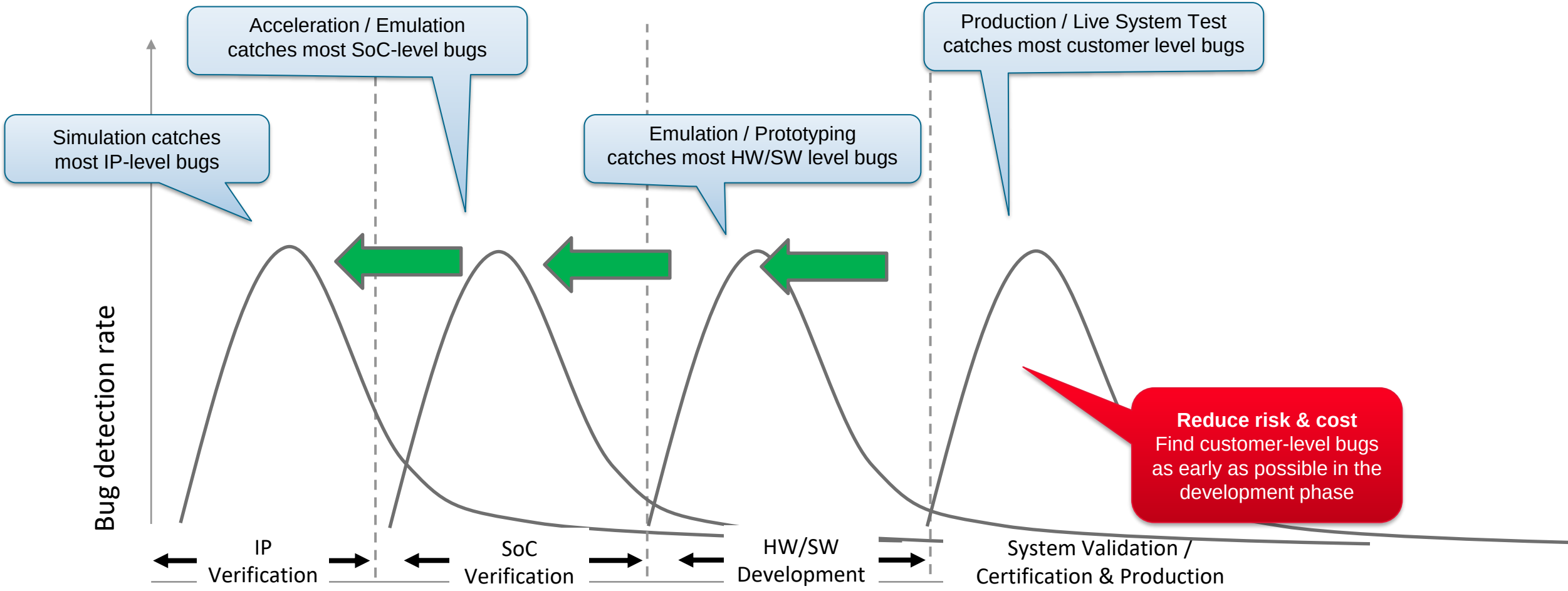
A Verification Based Wholistic Approach

- Unify your existing verification solutions
- Expand the scope of your verification calendar
- Use existing verification solutions for co-development “shift left”

Agenda

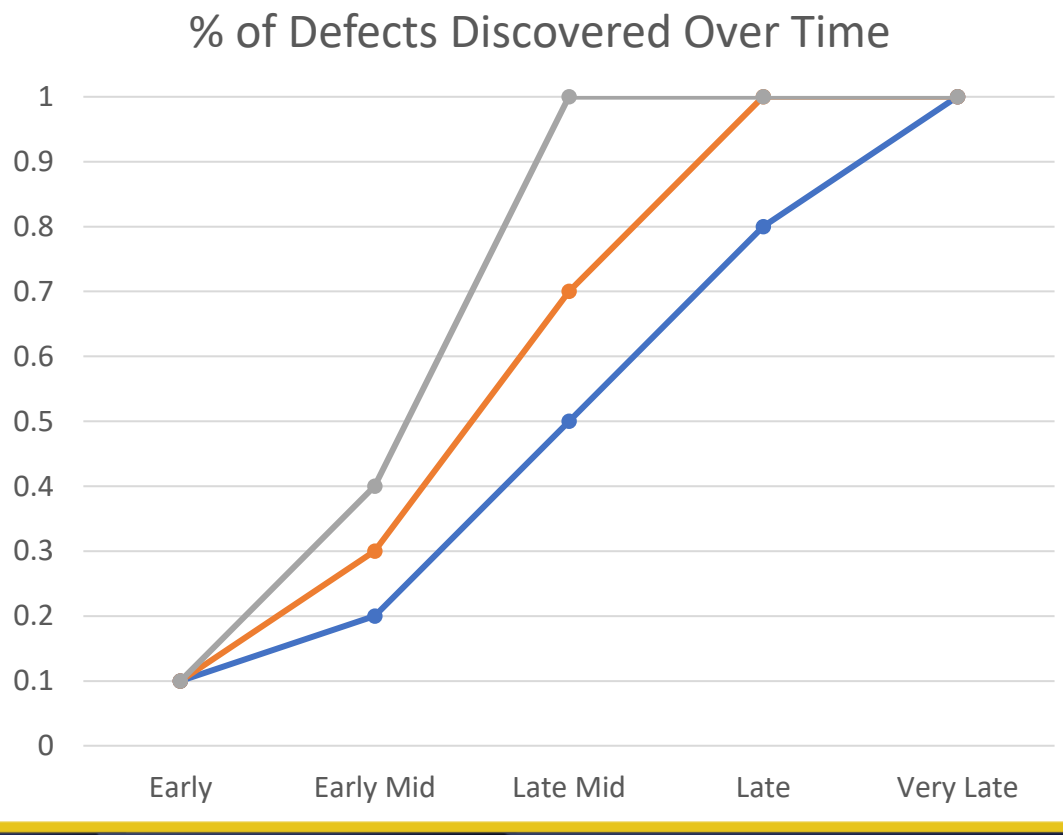
- Introduction
- Reviewing The Verification Flow
- Reviewing Current Solutions
  - Hardware, Interfaces, and Extensions
- Why Should We Change?
- A Comprehensive Solution
- Summary and Conclusion

Bug detection as early as possible



Shift Left vs Up and To The Right

- View cumulative defects over time rather than rate
- All improved flows are about discovering more defects earlier
- The slope also implies acceleration



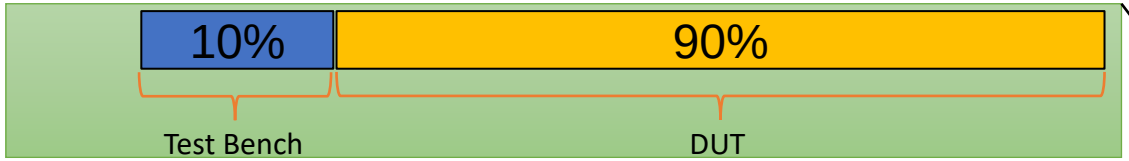
Can We Achieve Acceleration through Parallelism?

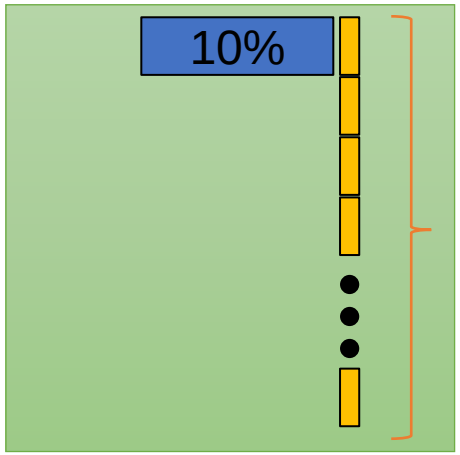
- Acceleration prediction as defined by Amdahl's Law
  - How much of a speedup one could get for a given parallelized task
  - Amdahl's Law is a statement of the maximum theoretical speed-up
  - The actual speed-ups are always less than the theoretical speed-up
- How it works
  - F is the fraction of a program execution that is sequential
  - 1-F is the fraction of a program that can be parallelized
  - P is the degree of parallelization

$$Speedup = \frac{1}{(F + (1 - F)/P)}$$

Its Important to Address the Right Problem

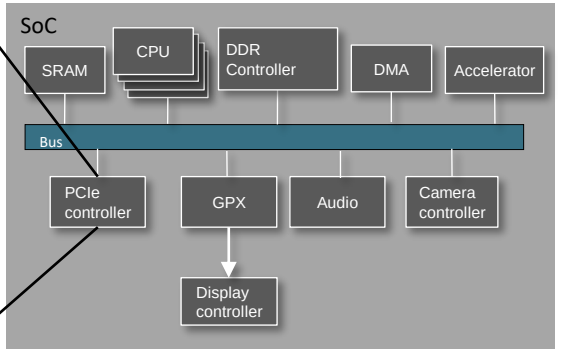
- We are all engineers and can all understand Amdahl's Law
- Too many verification engineers forget about Amdahl's Law or apply too narrowly





Theoretical max

$$\frac{1}{(10\%+(1-10\%)/1024)} \approx 9.91$$



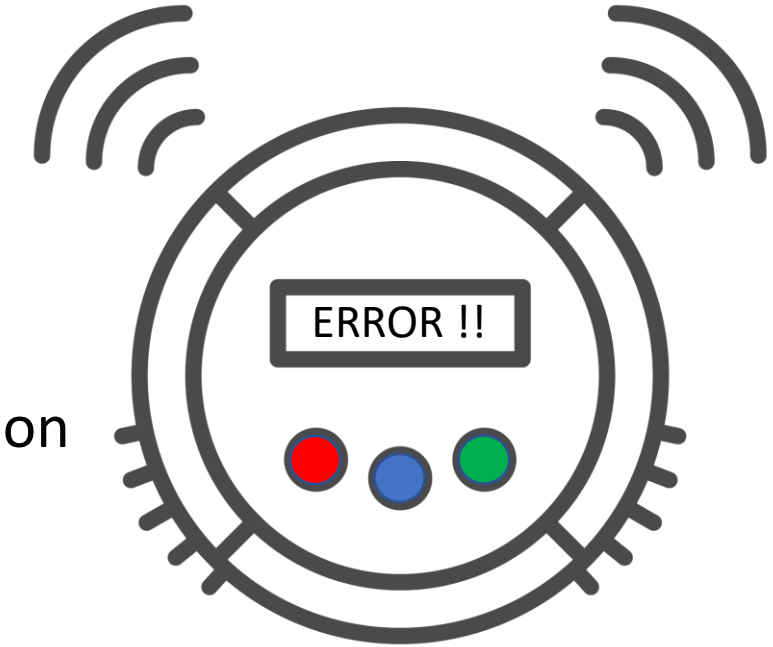
What Have we Learned from Virtual Platforms

- Most hardware “defects” found when running software on a platform are actually defects in the specification, often incompleteness
- It’s not fast enough
- You must verify the spec and not just the RTL

A Fundamental Disconnect

- Hardware engineers assume that the goal of Verification is
  - That each line of RTL works correctly
  - That the RTL implements the specification correctly
- Software engineers
  - Don’t believe in the existence of lines of RTL
  - Assumes the Spec has been evaluated for correctness and sufficiency
    - And that the HW is correct and consistent
- Problems arise when:
  - The spec is incomplete
  - The spec is interpreted differently by different people
- Running SW exposes these differences (often unintentionally)

Example of Spec Error



- Spec for a device
  - When the blue button is pressed turn on
  - When the green button is pressed perform the operation
  - When the red button is pressed turn off
  - After some period of inactivity turn off
- Potential problems (exceptions):
  - What happens when you push the blue button but not off?
  - What happens when you push the red button while the operation is ongoing?
  - What happens when you push the green button again?

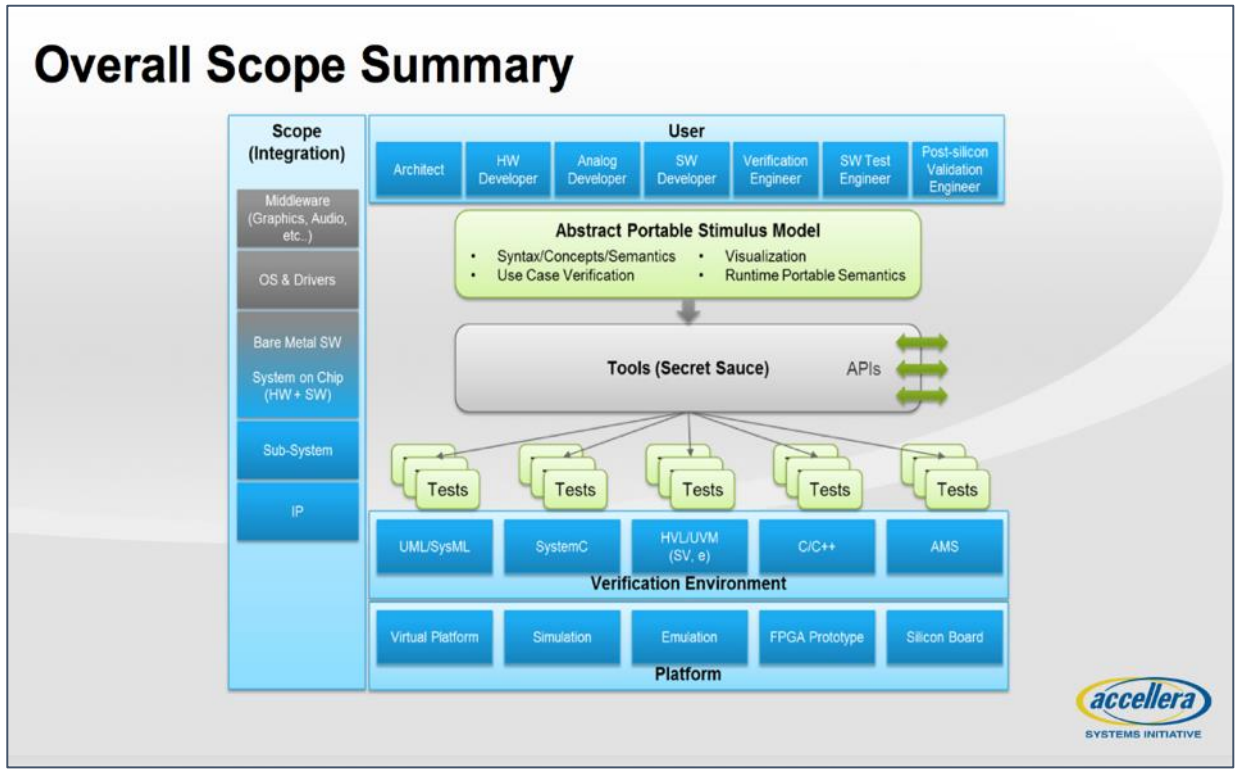
Is System Level Design part of Verification?

- Only it if wasn’t done during the Design phase
- Software is increasingly important to our products
- Software runs on the system, not one component
- We must have a system level design, even if minimal, in order to verify that the software will run

Agenda

- Introduction
- Reviewing The Verification Flow
- Reviewing Current Solutions
  - Hardware, Interfaces, and Extensions
- Why Should We Change?
- A Comprehensive Solution
- Summary and Conclusion

Doesn’t the Portable Stimulus Standard (PSS) solve this?

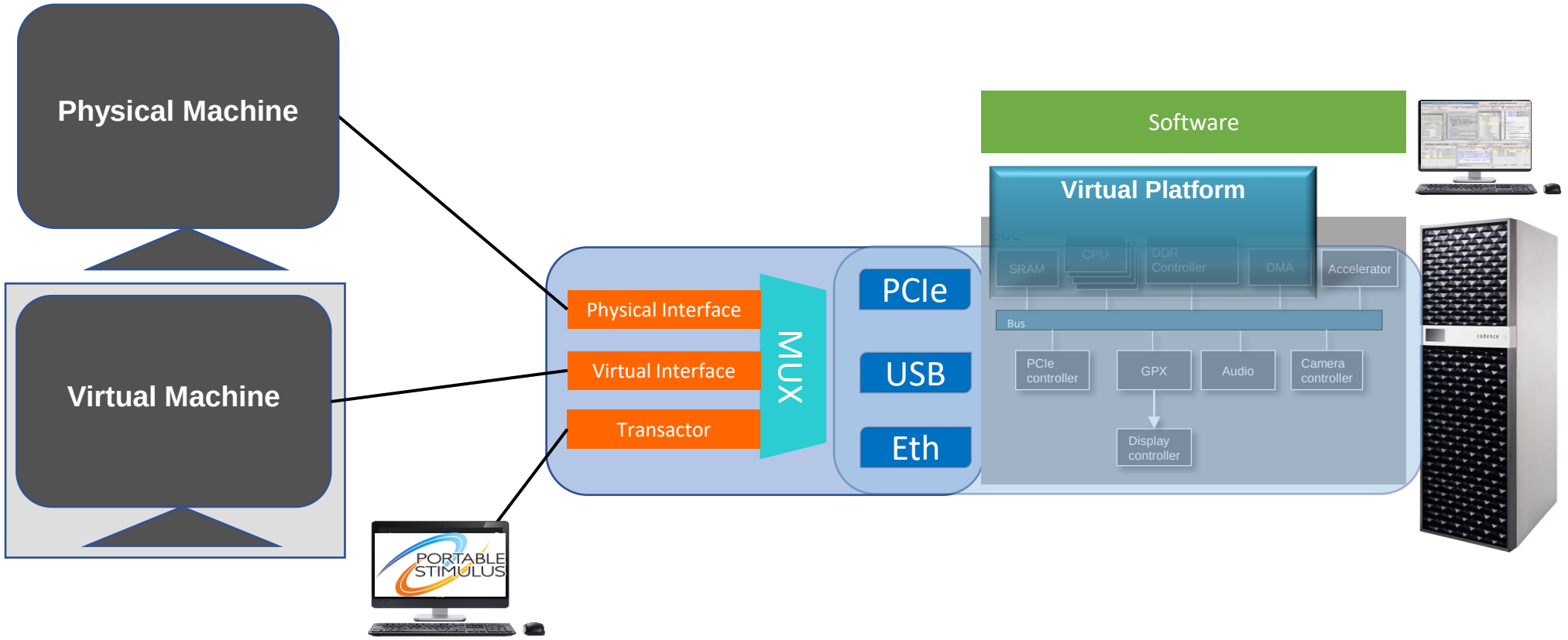


- Accellera standard
  - Single representation of stimulus and test scenarios
- Describe Test Intent and Design Behaviors
  - Use-cases, data flows
  - Legal scenario space, resources, configurations
- Deliver Test Portability
  - Vertical reuse: From IP to SoC
  - Horizontal reuse: from Simulation to Emulation to Post Silicon

Now put all the parts together

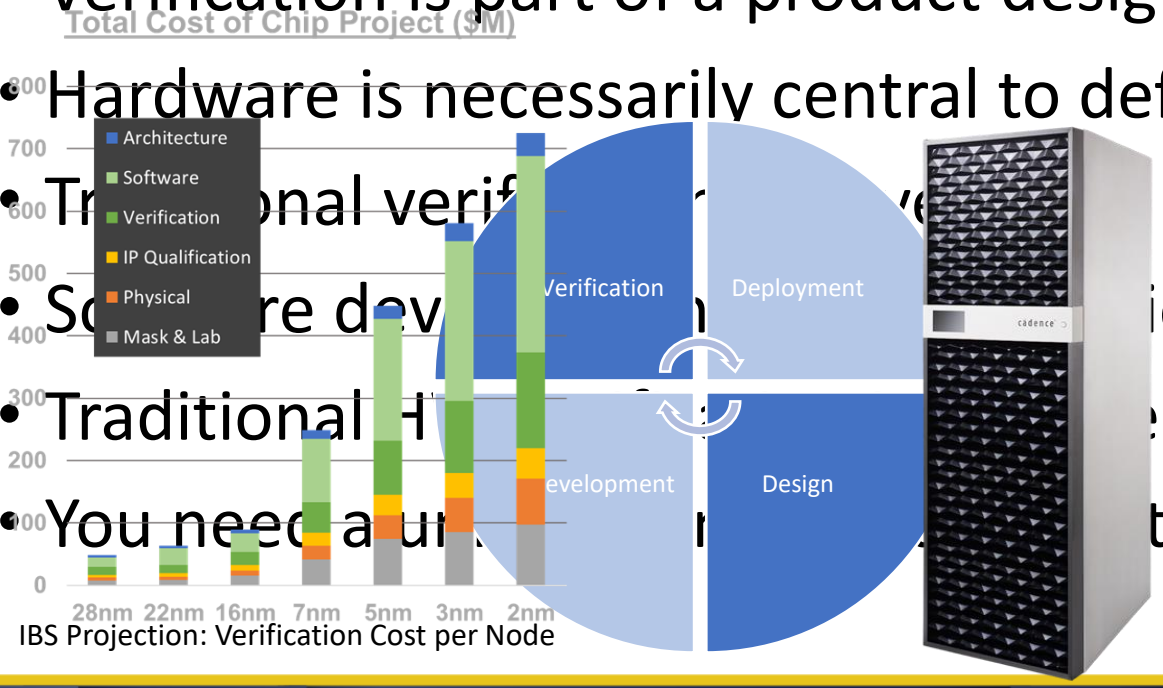
- Acknowledge all aspects of verification
  - Validate that the Architectural Spec is sufficient and consistent
  - Validate that the Implementation Specs are complete and correct
  - Validate that the RTL is correct
  - Validate that the software is designed to the same specs
  - Validate at both unit and system levels
- Apply the right tool to the right job
  - Migrate assets from one tool to the next to avoid rework

A Unified Congruent Wholistic Flow Combining The Best of All Tools

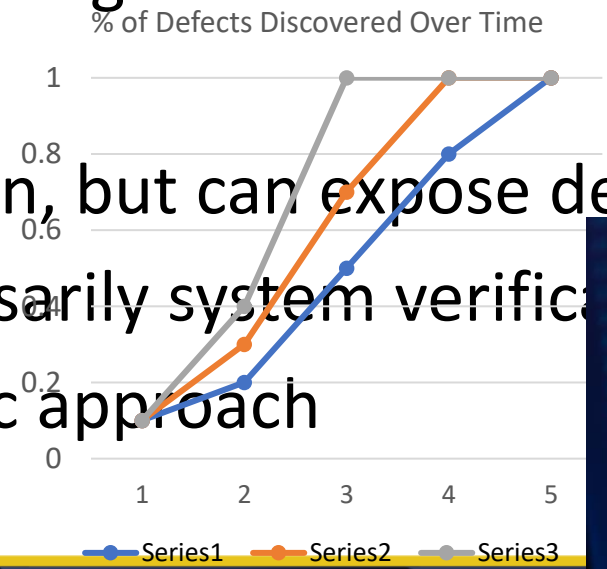


Summary and Conclusion

- The scope of Verification continues to increase
- Verification is part of a product design flow that must include design
- Hardware is necessarily central to defining a verification flow
- Tr…onal veri… …
- So…re dev… …ion, but can expose defects
- Traditional H… …essarily system verific…
- You nee… a u… …tic approach



IBS Projection: Verification Cost per Node



2023 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

SAN JOSE, CA, USA
FEBRUARY 27-MARCH 2, 2023

Questions

Ross Dickson - Lance Tamura - Michael Young

Cadence Design Systems

accellera SYSTEMS INITIATIVE

2023 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

SAN JOSE, CA, USA
FEBRUARY 27-MARCH 2, 2023

Backup

accellera SYSTEMS INITIATIVE