



—

# SoC – “Have I Done it Right”?

Anil TS

---

# Agenda

- SoC Validation
- Tradeoffs of Verification Methods
- Why Emulation?
- Helium Hybrid



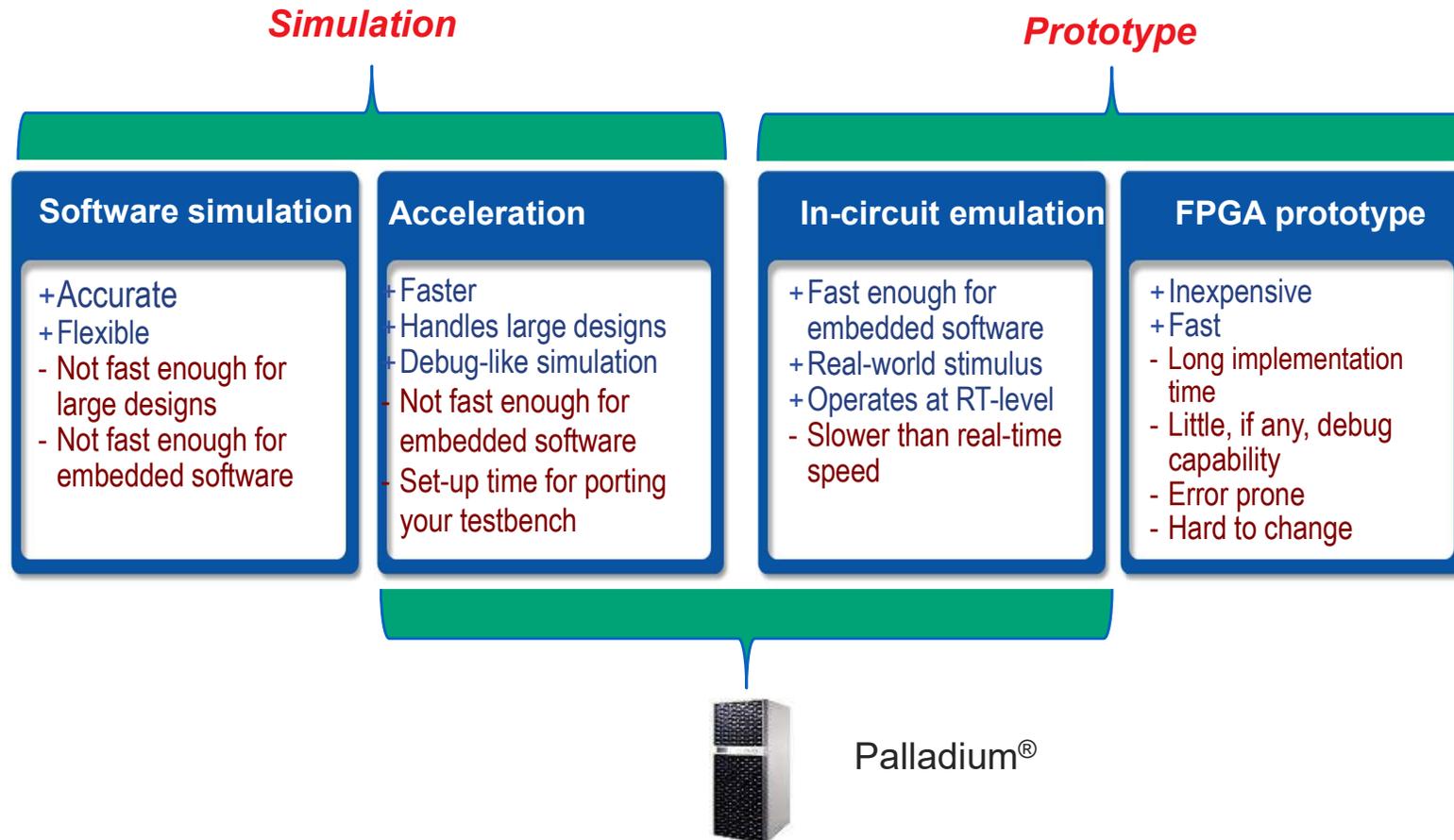
---

## SoC Validation – Do we Have What we Want?

- Need to validate the SoC as per architectural definition
- Will give confidence in silicon
- Early software bring-up
- Quick driver development

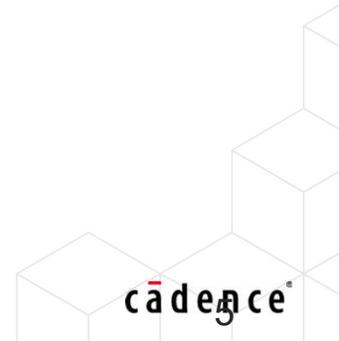


# Tradeoffs of Verification Methods

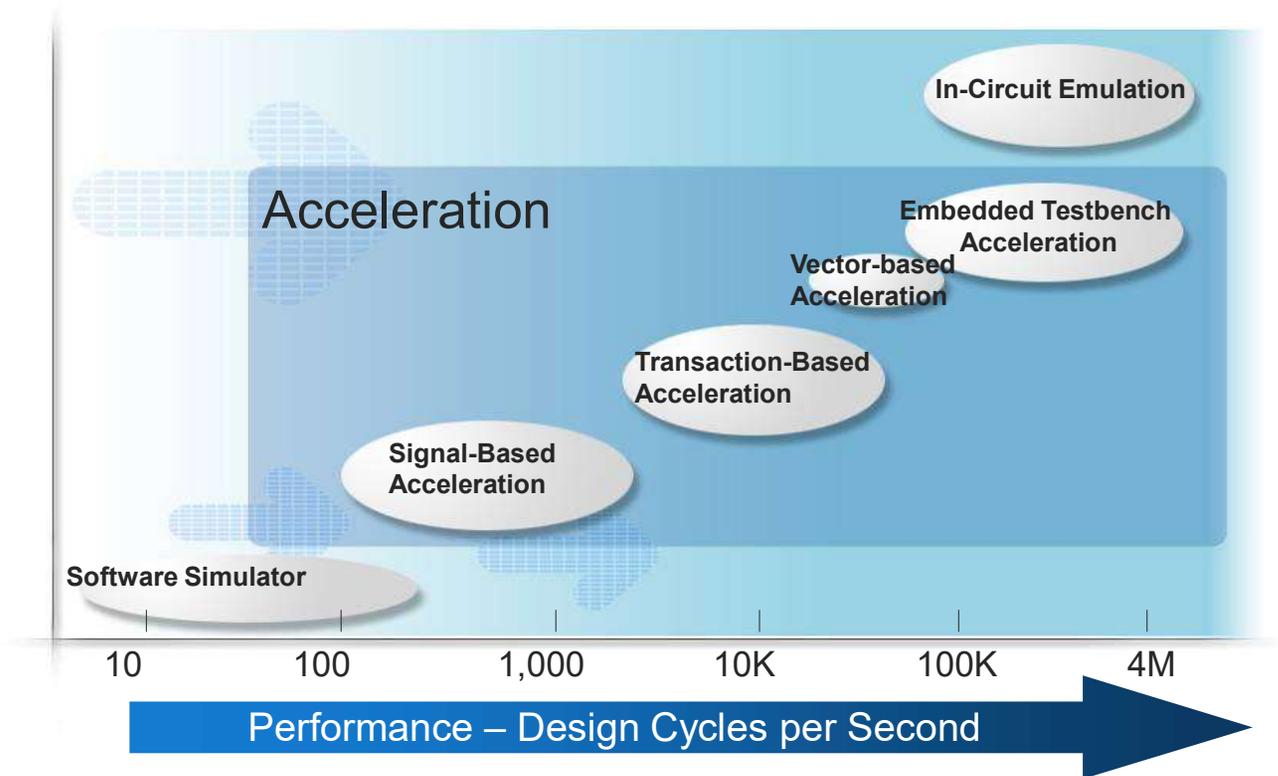


# Why Emulation and Acceleration?

- Much faster than simulation, especially for large designs
- Verifies functionality in the actual system environment
- Finds functional system problems before silicon
- Allows concurrent development of hardware, software, and systems
- Enables efficient post-silicon debug
- Acceleration eliminates the barriers to in-circuit emulation
  - Can be done with the incomplete design, or just parts of the design
  - Does not need external target hardware
- Can be as fast as emulation
  - However, getting the best performance requires an appropriate testbench



# Palladium Use Models



# Verification Engines



## SDK OS Simulation

- Highest speed
- Ignores HW



## Virtual Platform

- Almost @ speed
- HW Functionally accurate



## HDL Simulation

- Best HW debug
- HW accurate, too slow for SW



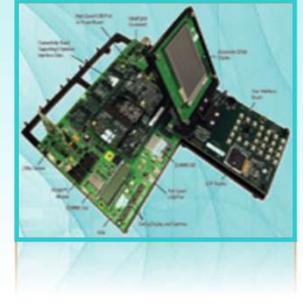
## Acceleration Emulation

- Good SW and HW Debug
- Moderate speed for SW



## FPGA Prototype

- Fast SW execution
- Harder HW debug



## Prototyping Board

- Real-time speed
- Difficult HW debug

← Earlier availability

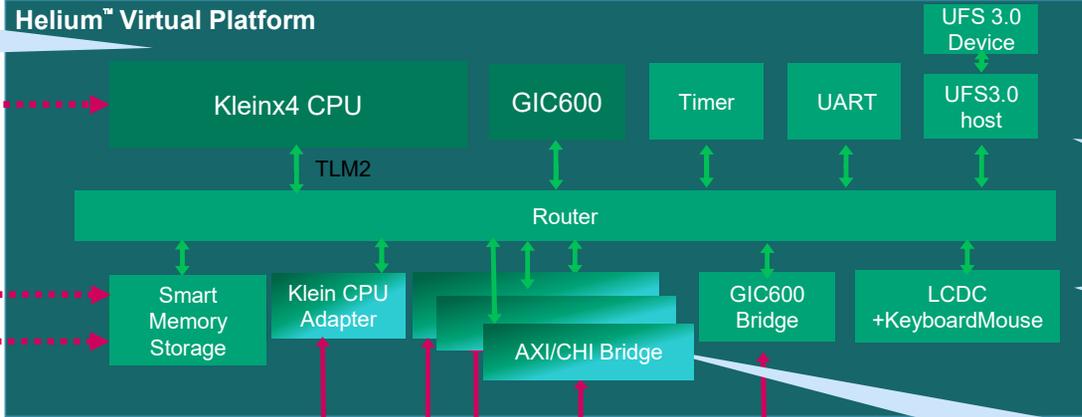
→ Better accuracy

# Helium SoC Hybrid Example

**Component Color Key**

Customer	RTL	
	RTL	
Cadence	TLM	
	Mem I/F	

Helium Unified Software Debug  
• Virtual and RTL Processors



**Helium Models**

- Peripherals
- Routers
- Integrated ARM Fast Models
- Advanced Models

**Helium Platform Assembly**

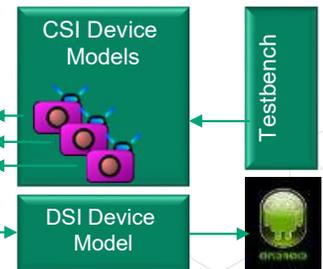
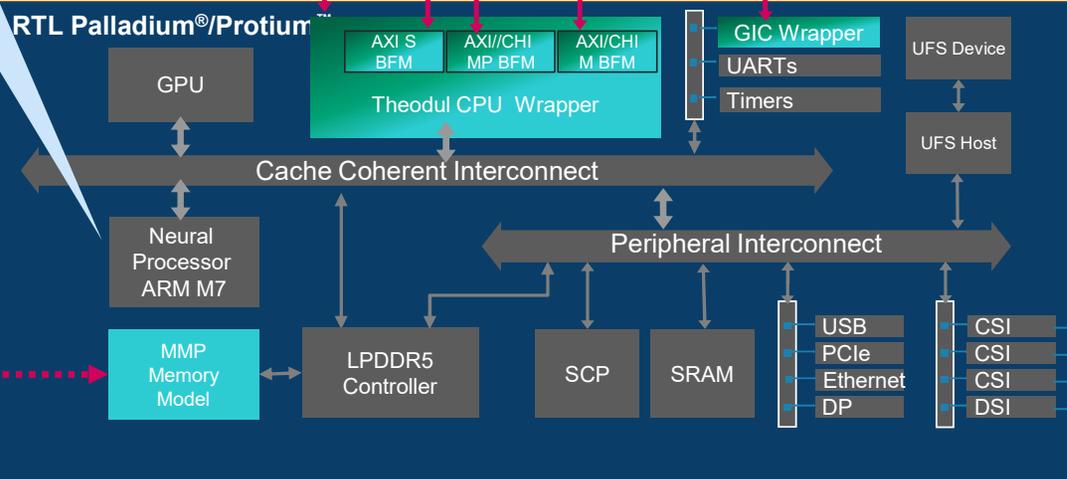
- Create, Edit, configure



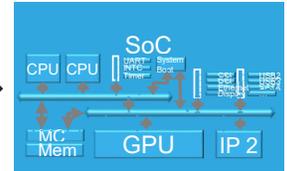
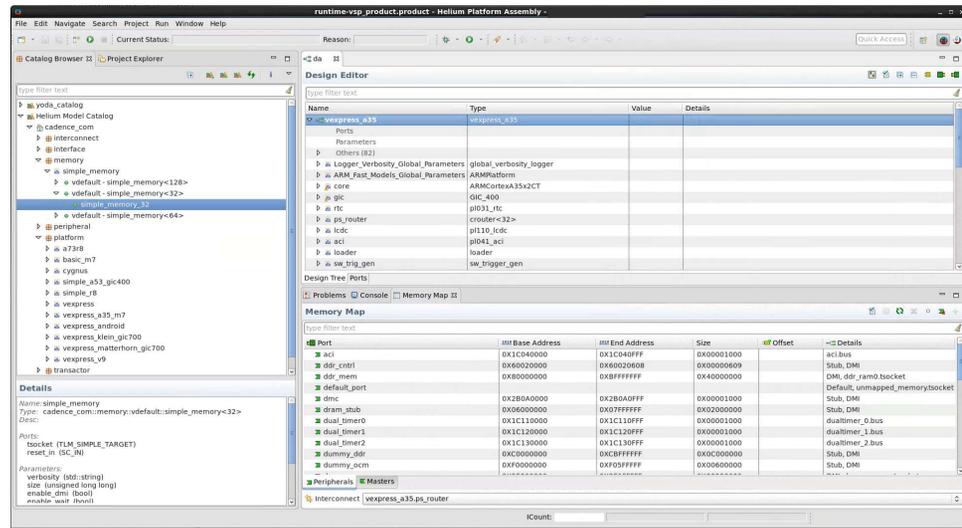
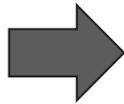
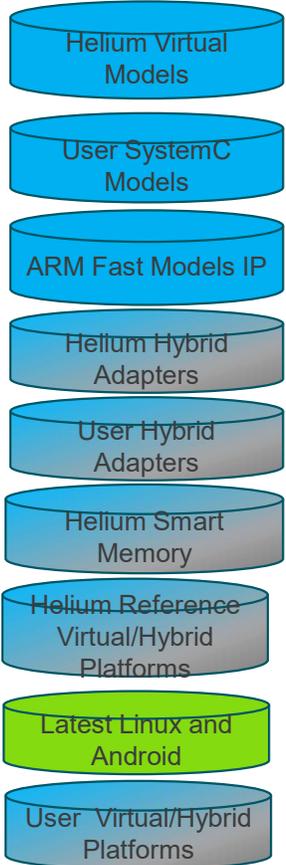
**Helium Hybrid Adapters**

- CPUs, GICs RTL Wrapper
- Protocol Bridges
- Signal Bridges

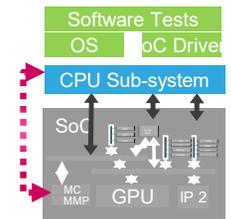
Helium Smart Memory  
• Virtual/RTL Coherency



# Helium Virtual and Hybrid Platform Assembly



Virtual Platforms



Hybrid Platforms

## Model and Platform Library

Library Manager

Manage Software Configurations

Customizable Build and Run Script Generation

Adaptive Port Binding

Intuitive Memory Map

One click updates to models

Smart connectivity

Rule Checker; Model, and Platform Documentation; Reports

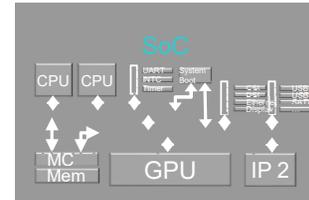
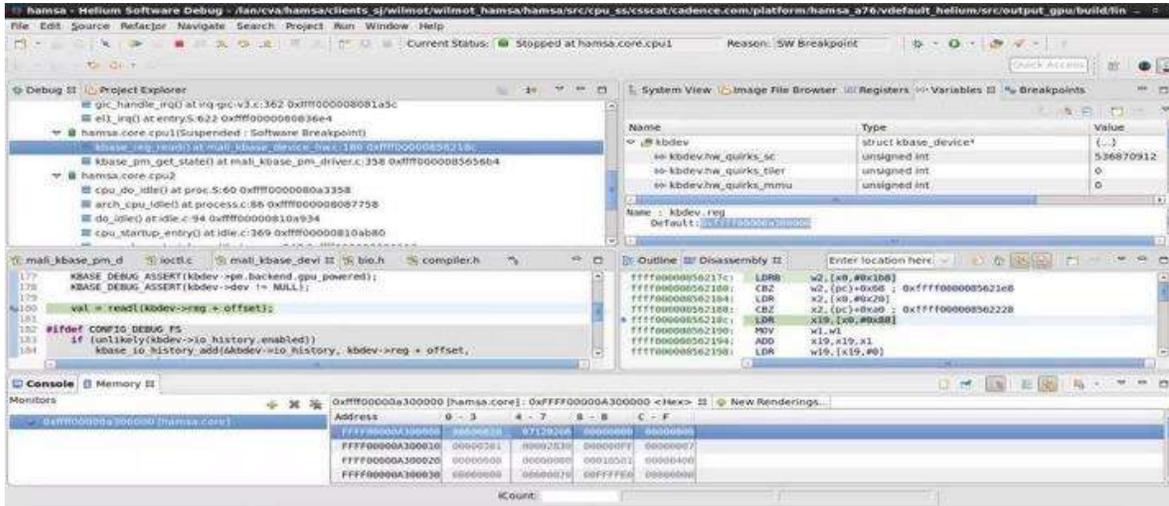
Built-in ARM Fast Model

Model library browser

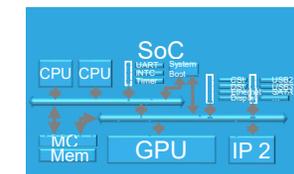
eclipse

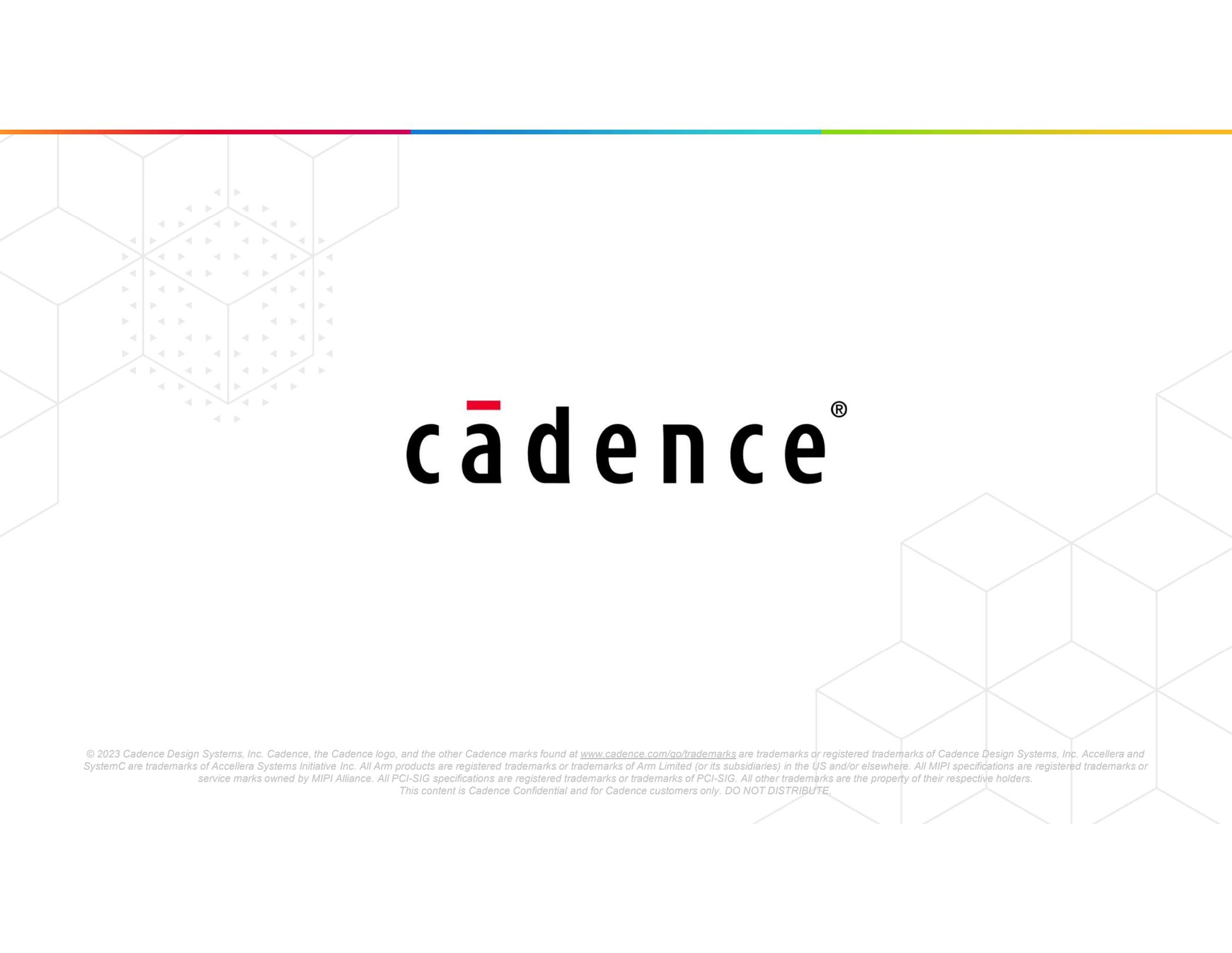


# Helium Unified Software Debug Across Runtime Engines



- Multi-core, multi-processor Debug
- Hardware-Software Command Console
- Mixed Abstraction Software Debug
- Memory View
- Variable View
- Breakpoints
- Core-specific Stack View
- Customized System View
- Powerful TCL scripting with HW and SW visibility
- Linux Kernel Module Debug
- Dynamic Disassembly
- Live RTL IP Register Values
- Register View





# cādence<sup>®</sup>

© 2023 Cadence Design Systems, Inc. Cadence, the Cadence logo, and the other Cadence marks found at [www.cadence.com/go/trademarks](http://www.cadence.com/go/trademarks) are trademarks or registered trademarks of Cadence Design Systems, Inc. Accellera and SystemC are trademarks of Accellera Systems Initiative Inc. All Arm products are registered trademarks or trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All MIPI specifications are registered trademarks or service marks owned by MIPI Alliance. All PCI-SIG specifications are registered trademarks or trademarks of PCI-SIG. All other trademarks are the property of their respective holders.  
This content is Cadence Confidential and for Cadence customers only. DO NOT DISTRIBUTE.