SoC – “Have I Done it Right”?

Anil TS

cadence®

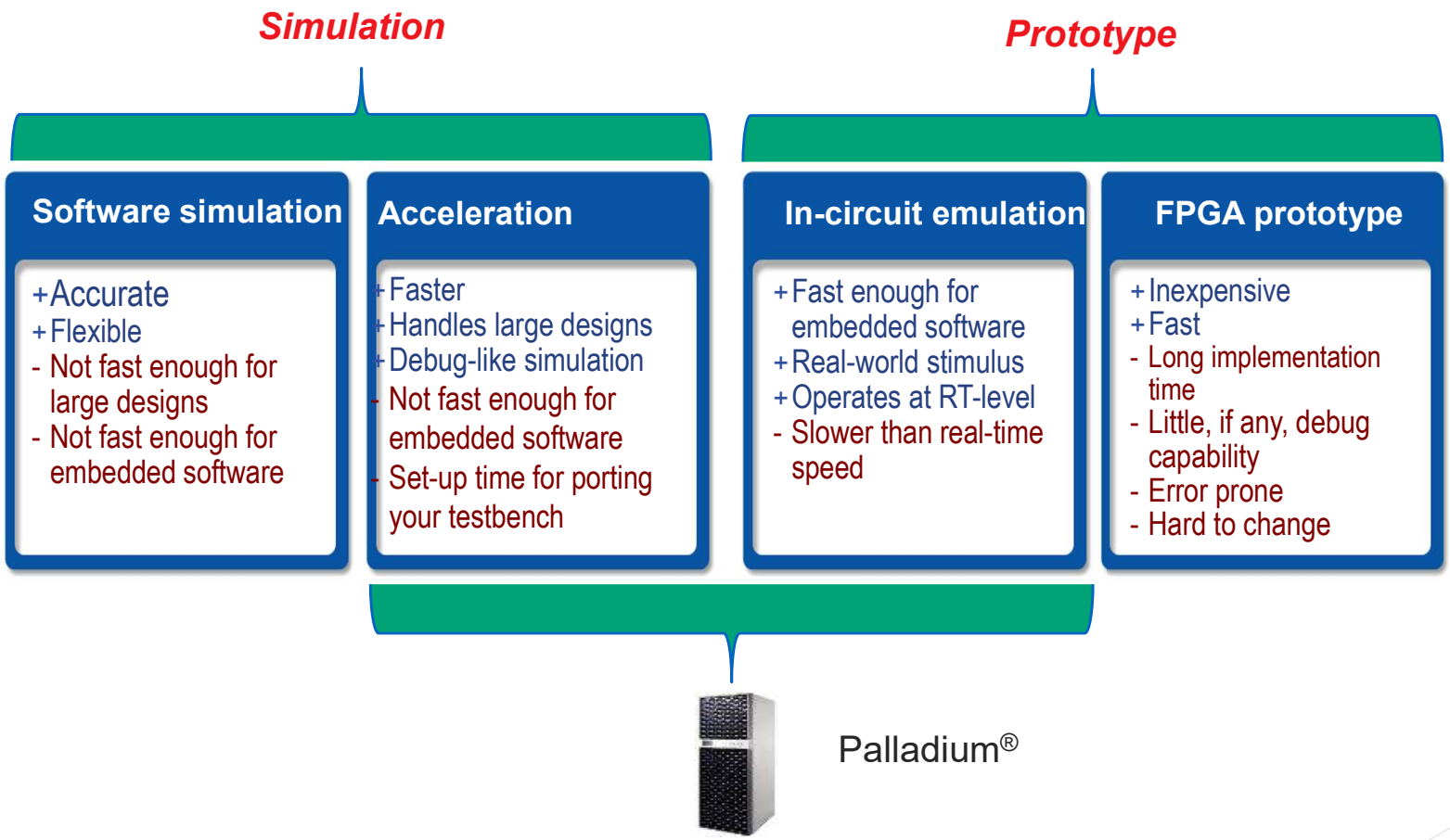
Agenda

- 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

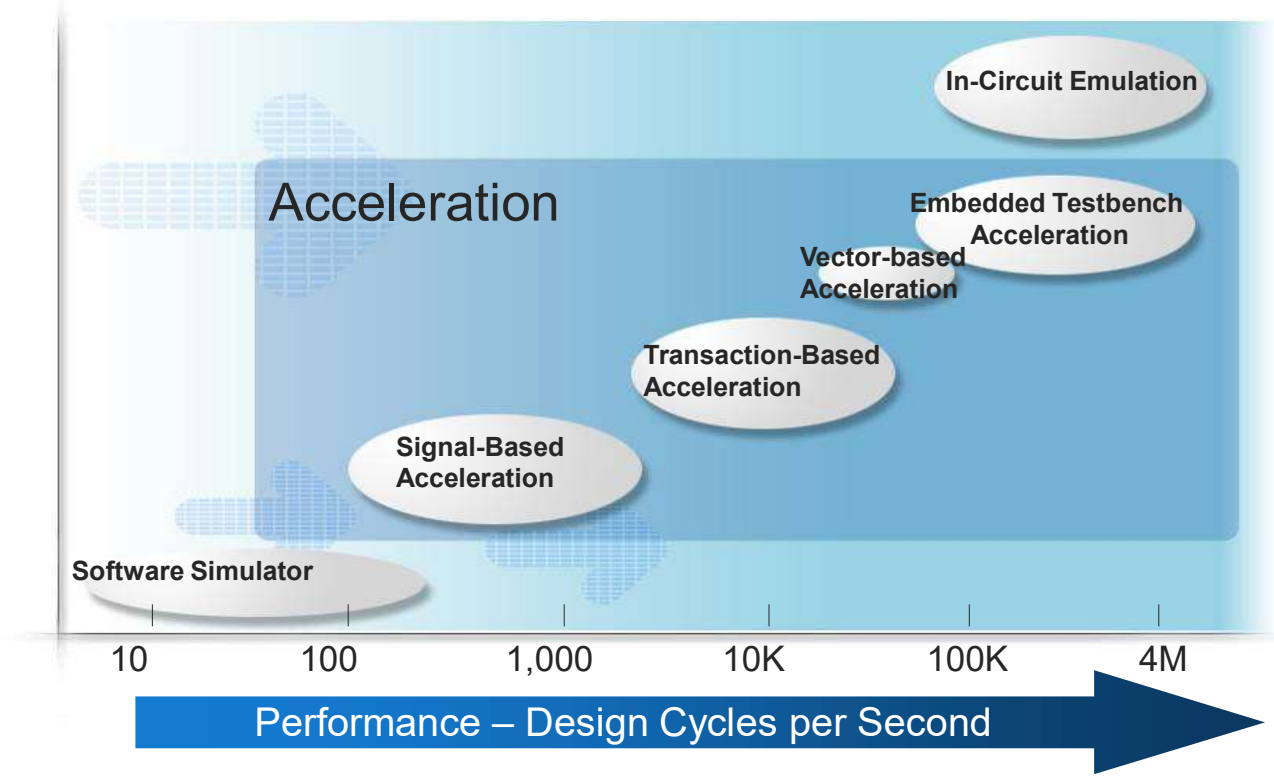
Tradeoffs of Verification Methods



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



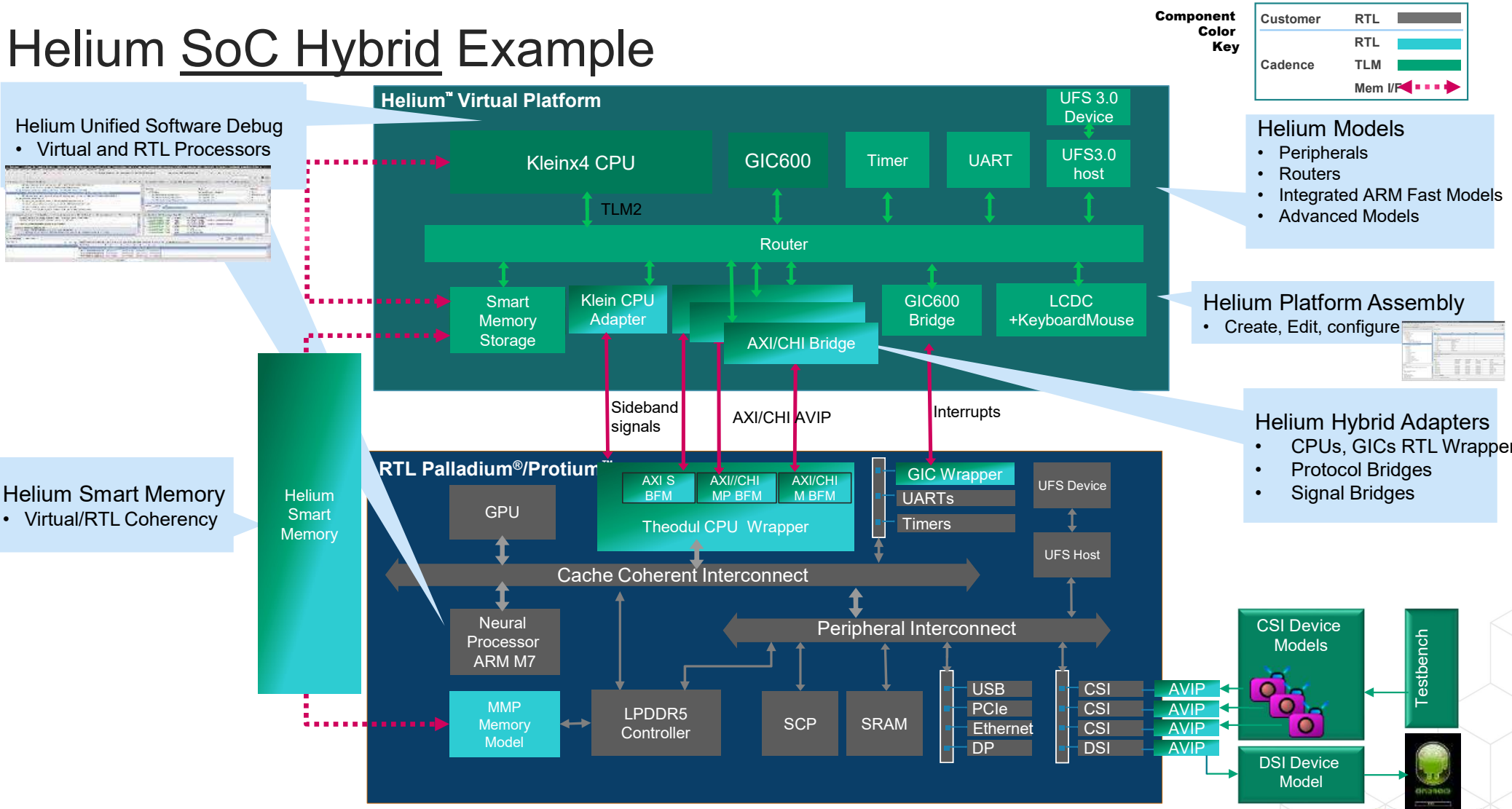
Verification Engines

SDK OS Simulation	Virtual Platform	HDL Simulation	Acceleration Emulation	FPGA Prototype	Prototyping Board
• Highest speed • Ignores HW	• Almost @ speed • HW Functionally accurate	• Best HW debug • HW accurate, too slow for SW	• Good SW and HW Debug • Moderate speed for SW	• Fast SW execution • Harder HW debug	• Real-time speed • Difficult HW debug

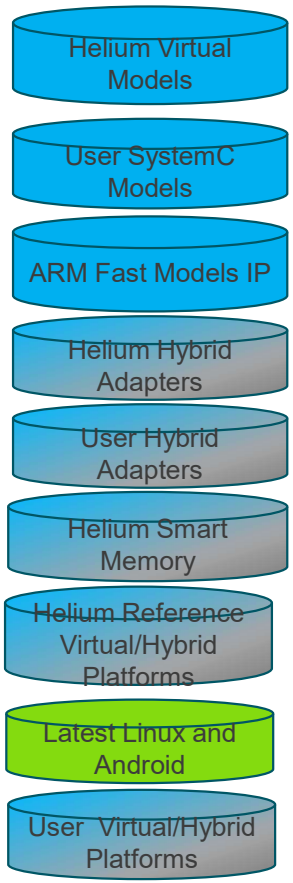
Earlier availability

Better accuracy

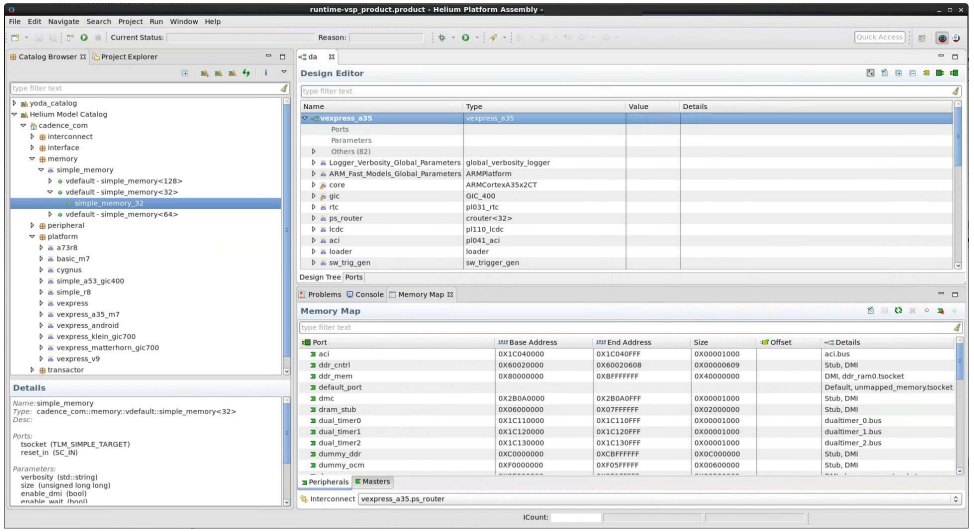
Helium SoC Hybrid Example

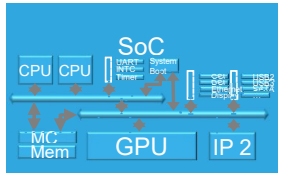


Helium Virtual and Hybrid Platform Assembly

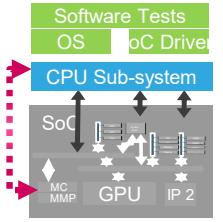


Model and Platform Library





Virtual Platforms

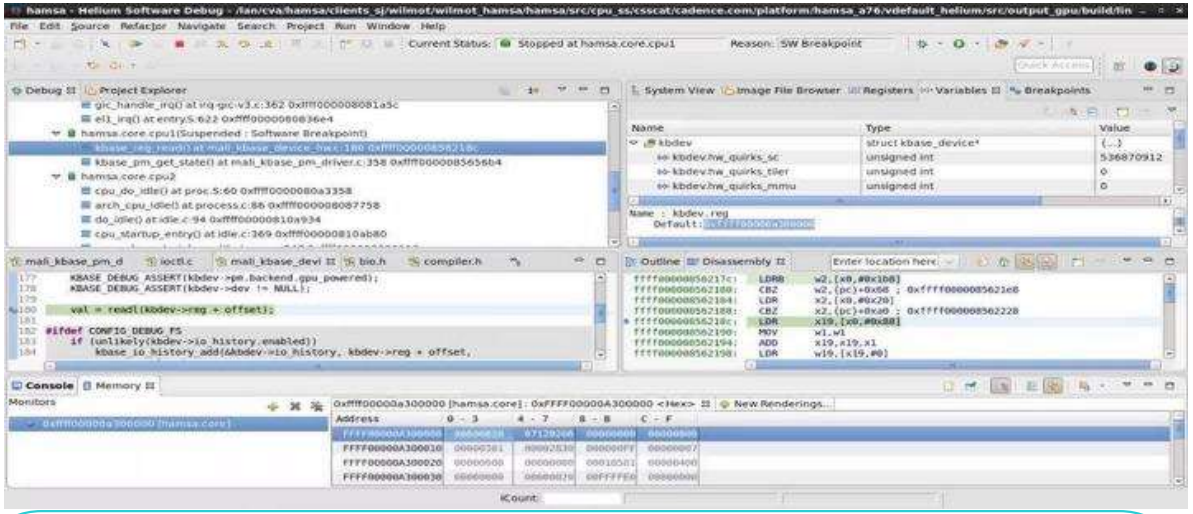


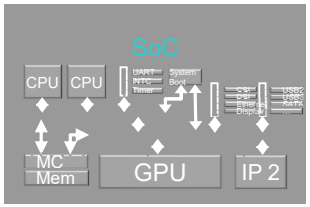
Hybrid Platforms

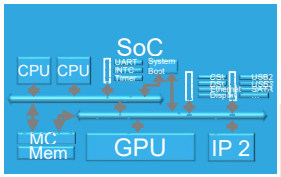
Library Manager

- 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
- eclipse
- Model library browser

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
- Live RTL IP Register Values
- Dynamic Disassembly
- Register View

eclipse

cadence