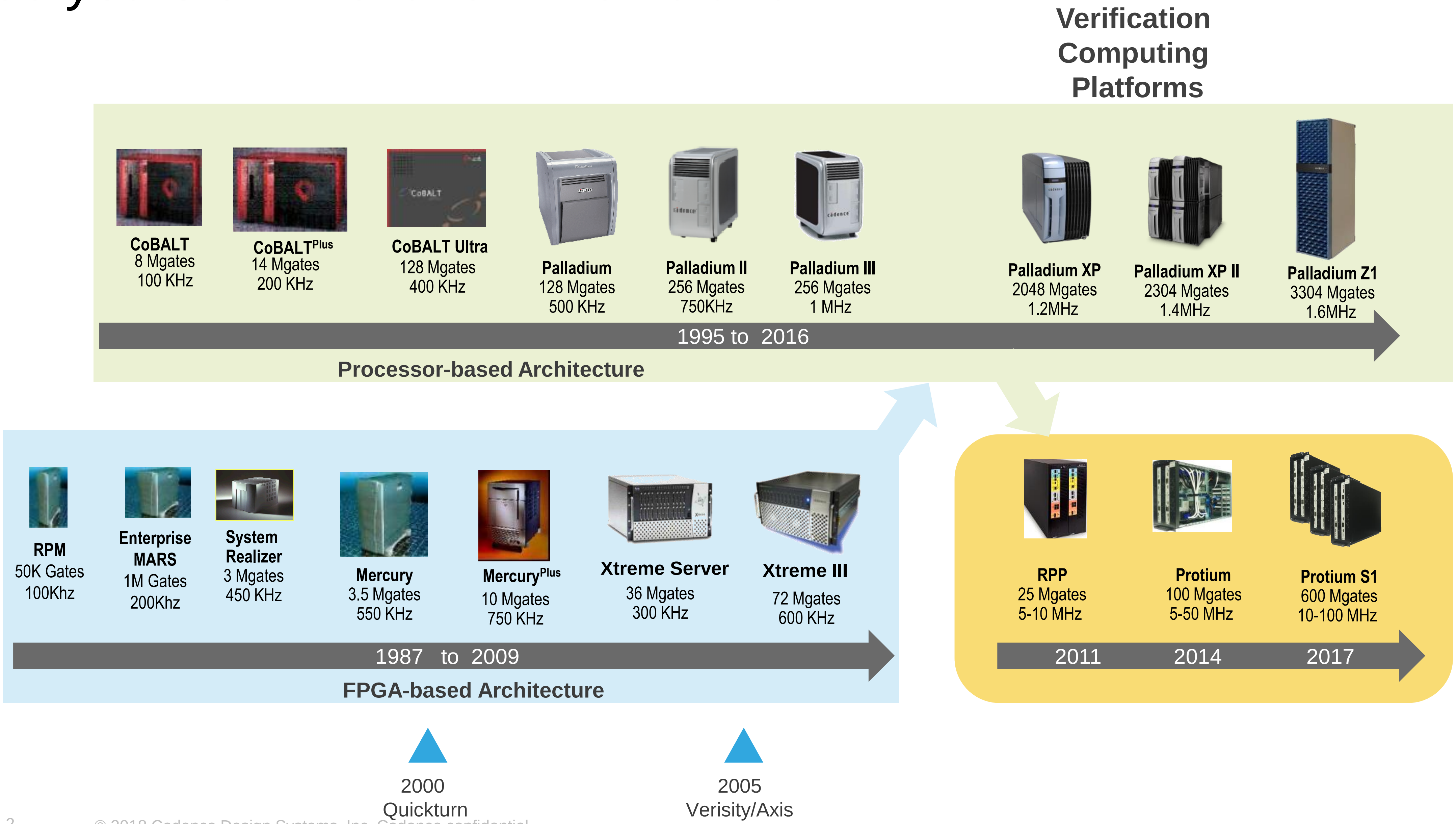
Protium S1 – FPGA-based prototyping made easy

Volker Wegner, Staff Application Engineer

Tensilica Day
Hannover, Germany
February 07, 2018

30 years of innovation in emulation



Verification Computing Platforms

CoBALT
8 Mgates
100 KHz

CoBALT Plus
14 Mgates
200 KHz

CoBALT Ultra
128 Mgates
400 KHz

Palladium
128 Mgates
500 KHz

Palladium II
256 Mgates
750KHz

Palladium III
256 Mgates
1 MHz

Palladium XP
2048 Mgates
1.2MHz

Palladium XP II
2304 Mgates
1.4MHz

Palladium Z1
3304 Mgates
1.6MHz

1995 to 2016

Processor-based Architecture

RPM
50K Gates
100Khz

Enterprise MARS
1M Gates
200Khz

System Realizer
3 Mgates
450 KHz

Mercury
3.5 Mgates
550 KHz

Mercury Plus
10 Mgates
750 KHz

Xtreme Server
36 Mgates
300 KHz

Xtreme III
72 Mgates
600 KHz

1987 to 2009

FPGA-based Architecture

RPP
25 Mgates
5-10 MHz

Protium
100 Mgates
5-50 MHz

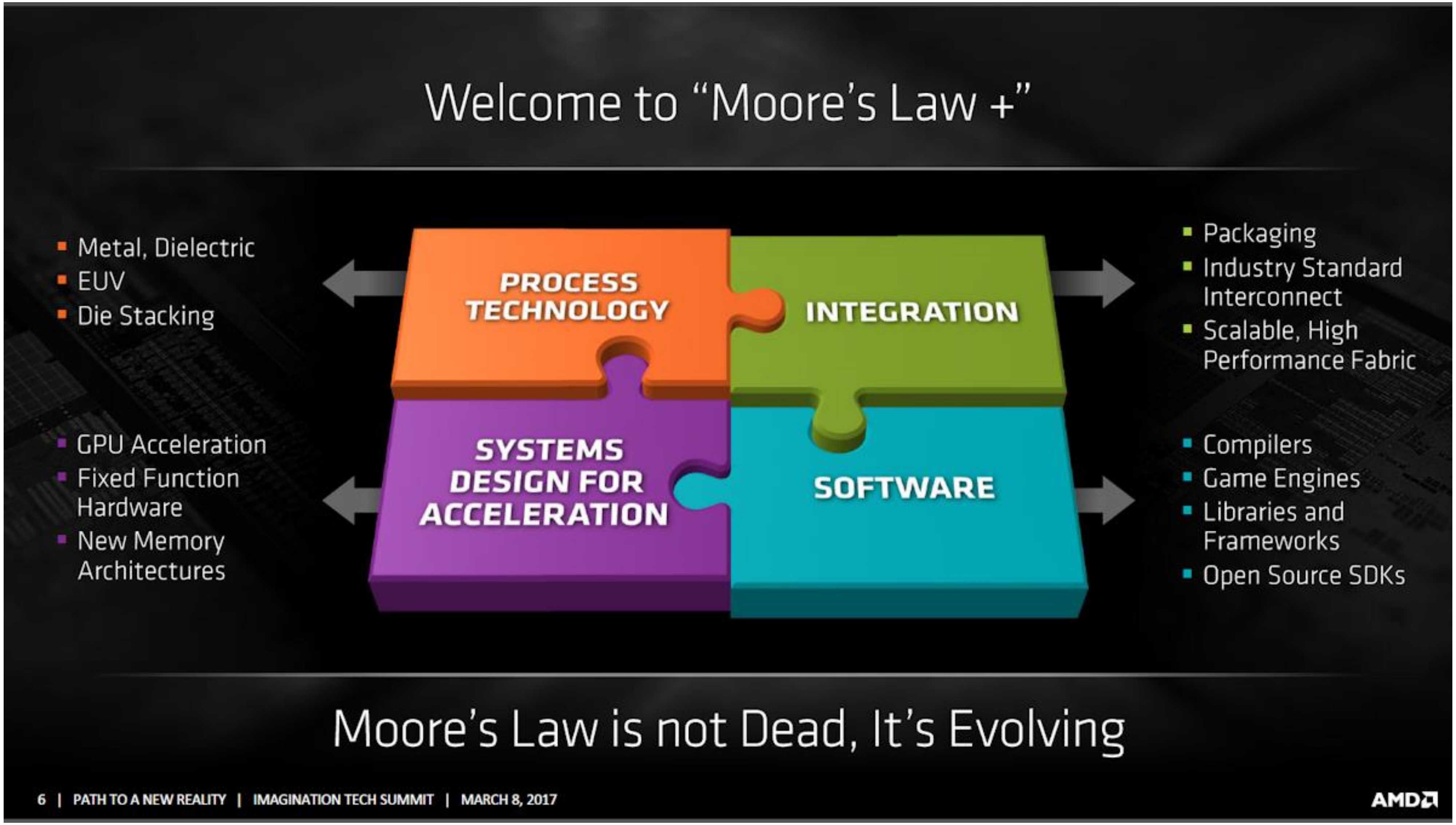
Protium S1
600 Mgates
10-100 MHz

2011 2014 2017

2000
Quickturn

2005
Verisity/Axis

Moore’s Law is evolving - It’s about the software ☺



Welcome to “Moore’s Law +”

- Metal, Dielectric
- EUV
- Die Stacking

- GPU Acceleration
- Fixed Function Hardware
- New Memory Architectures

PROCESS TECHNOLOGY

INTEGRATION

SYSTEMS DESIGN FOR ACCELERATION

SOFTWARE

- Packaging
- Industry Standard Interconnect
- Scalable, High Performance Fabric

- Compilers
- Game Engines
- Libraries and Frameworks
- Open Source SDKs

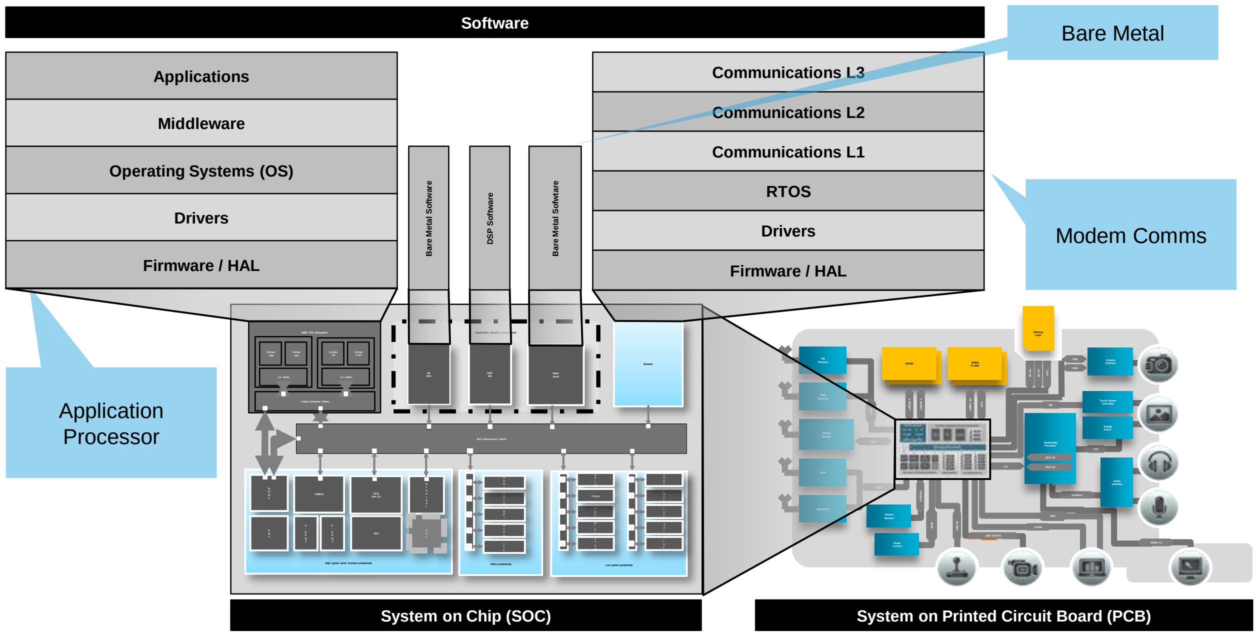
Moore’s Law is not Dead, It’s Evolving

6 | PATH TO A NEW REALITY | IMAGINATION TECH SUMMIT | MARCH 8, 2017

AMD

System & chip design in 2017 - 2020

- Time-to-market
- Development cost reduction
- Multi-core design and verification complexity
- Integration of new designs and derivatives
- Software stack development
- Hardware-software convergence
- More than 80% re-use
- **More than 60% of effort in software**



Software

Bare Metal

Applications

Middleware

Operating Systems (OS)

Drivers

Firmware / HAL

Bare Metal Software

DSP Software

Bare Metal Software

Communications L3

Communications L2

Communications L1

RTOS

Drivers

Firmware / HAL

Modem Comms

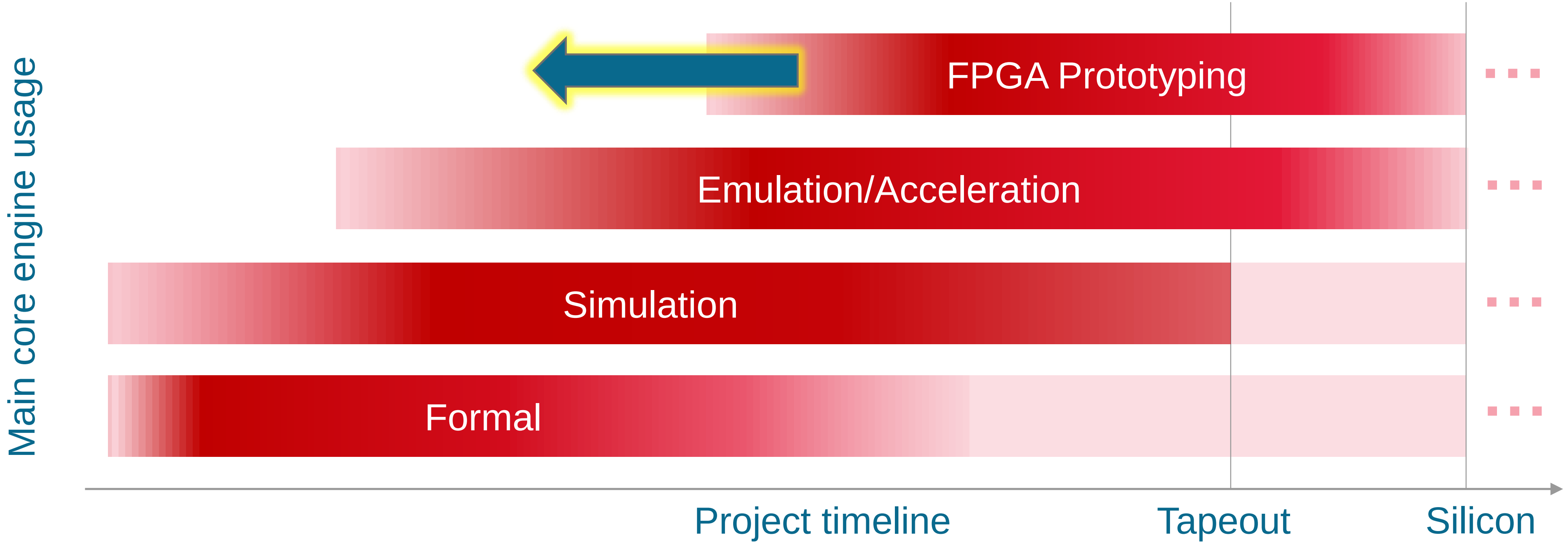
Application Processor

System on Chip (SOC)

System on Printed Circuit Board (PCB)

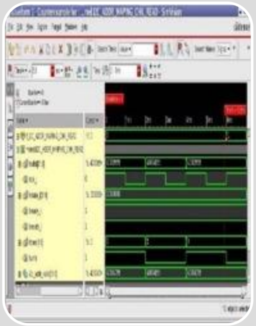
Customers Need the Fastest Engines

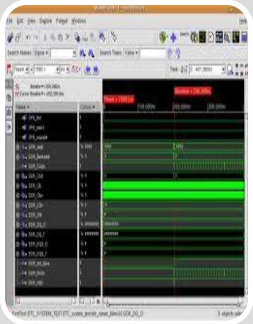
- **Ever-increasing verification** requirements driven by growing hardware and **software complexity**
- **Fast time to results** is essential to ensure projects can **meet schedules**
- **Right tools for the right job**: Combination of formal, simulation, emulation, and FPGA prototyping

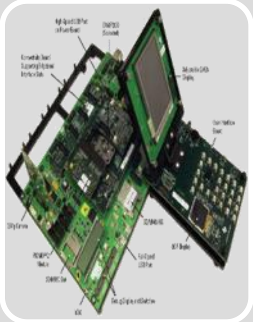


There is no “One Size Fits All”

Verification and software platforms need to interoperate







SDK OS Simulation

- Highest speed
- Earliest in the flow
- Ignore hardware

Virtual Platform

- Almost at speed
- Less accurate (or slower)
- Before RTL
- Great to debug (but less detail)
- Easy replication

Formal Analysis

- Non-scalable
- Exhaustive
- Early RTL
- Great for IP
- No SW execution

HDL Simulation

- KHz range
- Accurate
- Excellent HW debug
- Broadly available
- Mixed-abstractions
- Limited SW execution

Acceleration Emulation

- MHz Range
- RTL accurate
- After RTL is available
- Good to debug with full detail
- Expensive to replicate

FPGA Prototype

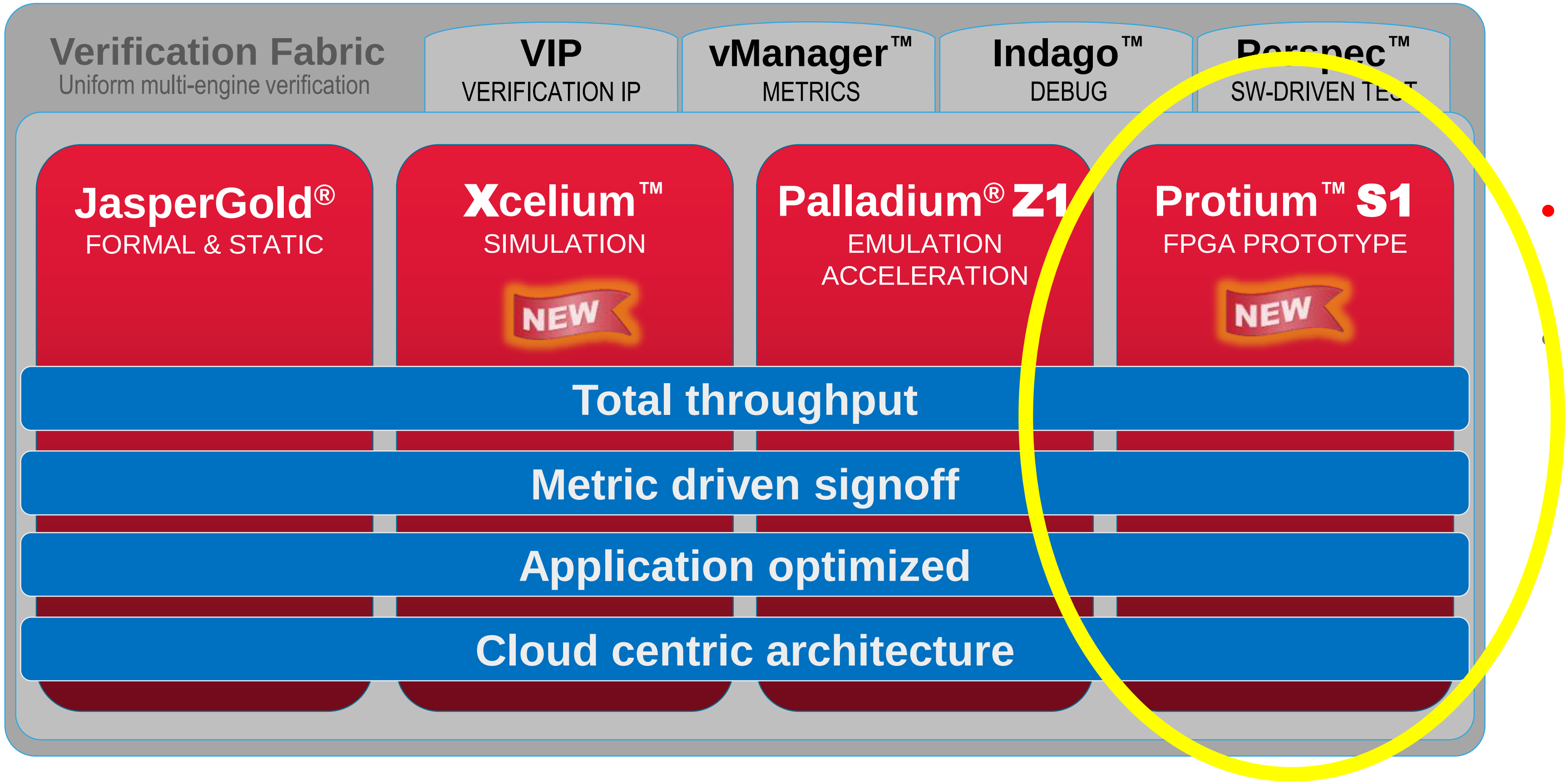
- 10’s of MHz
- RTL accurate
- After stable RTL is available
- OK to debug
- More expensive than software to replicate

Prototyping Board

- Real time speed
- Fully accurate
- Post Silicon
- Difficult to debug
- Sometimes hard to replicate

Cadence Verification Suite

Technology innovation leadership: Fast, Smart, and Optimized



Verification Fabric
Uniform multi-engine verification

VIP
VERIFICATION IP

vManager™
METRICS

Indago™
DEBUG

Perspec™
SW-DRIVEN TEST

JasperGold®
FORMAL & STATIC

Xcelium™
SIMULATION
NEW

Palladium® Z1
EMULATION
ACCELERATION

Protium™ S1
FPGA PROTOTYPE
NEW

Total throughput

Metric driven signoff

Application optimized

Cloud centric architecture

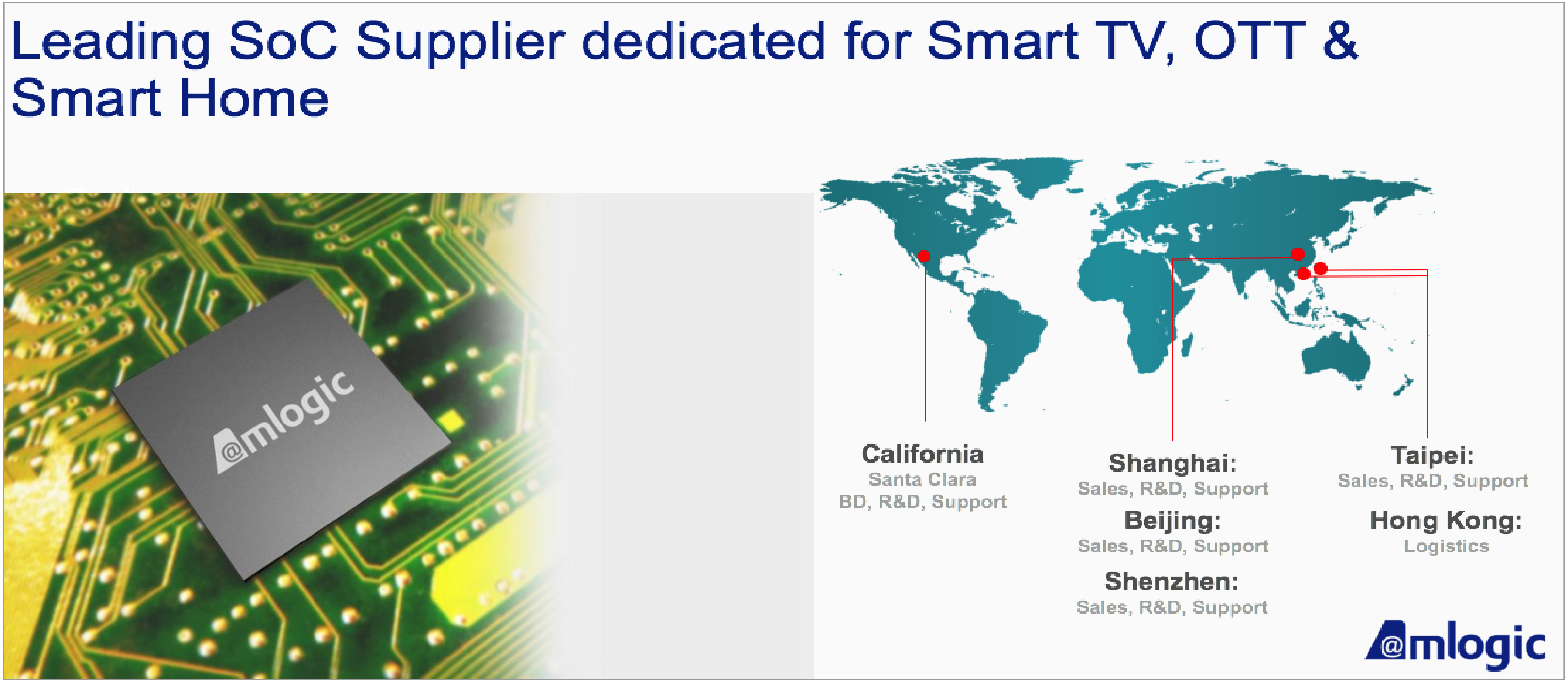
- ***Fast*** **Best-in-class engines**
- ***Smart*** **Flow-driven engine integrations**
- ***Optimized*** **comprehensive solutions**

What is FPGA-Based Prototyping?

- **Primary platform for pre-silicon software development and validation**
  - Maps a digital ASIC, ASSP, SoC design or part thereof into one or more FPGAs

- **Allows software to execute in real world environments**
  - Provides pre-silicon execution speeds in the 10s of MHz
  - Enables connectivity to real peripherals
    - Runs real world traffic flows including interrupts and random events
    - Runs error conditions and handling errata with other system components

Case study: Amlogic



Leading SoC Supplier dedicated for Smart TV, OTT & Smart Home

California
Santa Clara
BD, R&D, Support

Shanghai:
Sales, R&D, Support

Beijing:
Sales, R&D, Support

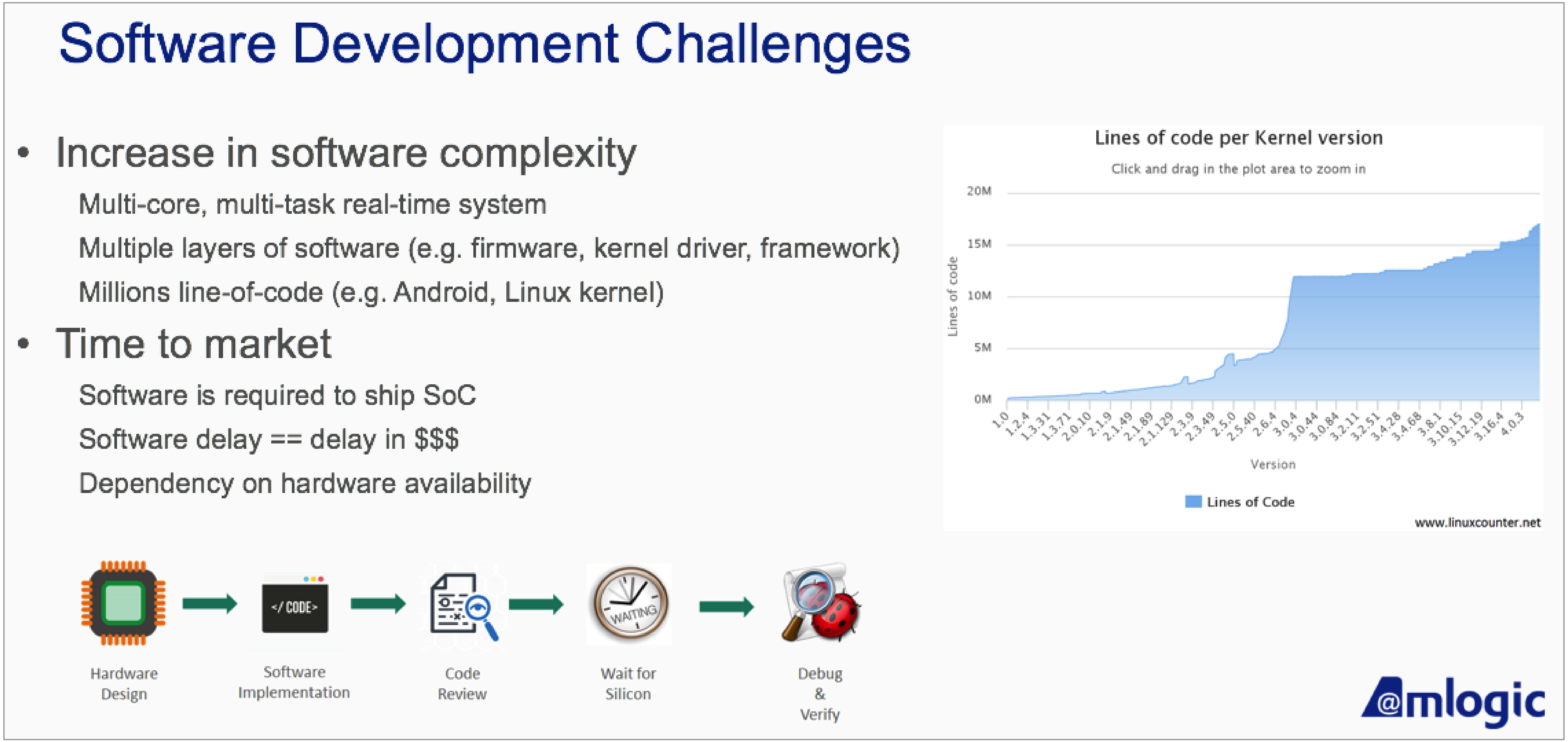
Shenzhen:
Sales, R&D, Support

Taipei:
Sales, R&D, Support

Hong Kong:
Logistics

Amlogic

Case study: Amlogic (cont.)



Software Development Challenges

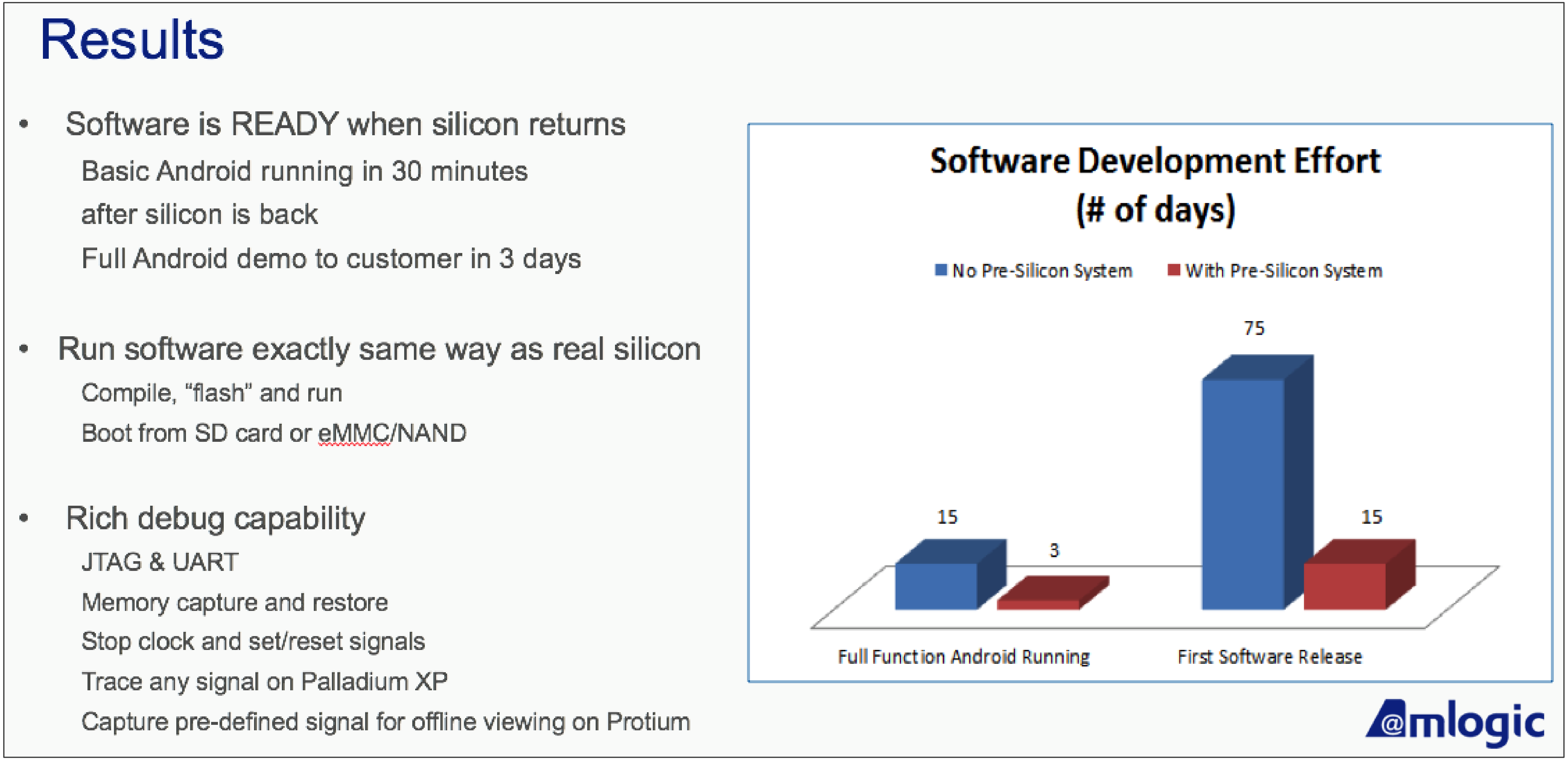
- Increase in software complexity
  - Multi-core, multi-task real-time system
  - Multiple layers of software (e.g. firmware, kernel driver, framework)
  - Millions line-of-code (e.g. Android, Linux kernel)
- Time to market
  - Software is required to ship SoC
  - Software delay == delay in $$$
  - Dependency on hardware availability

Lines of code per Kernel version

Click and drag in the plot area to zoom in

www.linuxcounter.net

Case study: Amlogic (cont.)



Results

- Software is READY when silicon returns

  Basic Android running in 30 minutes

  after silicon is back

  Full Android demo to customer in 3 days

- Run software exactly same way as real silicon

  Compile, “flash” and run

  Boot from SD card or eMMC/NAND

- Rich debug capability

  JTAG & UART

  Memory capture and restore

  Stop clock and set/reset signals

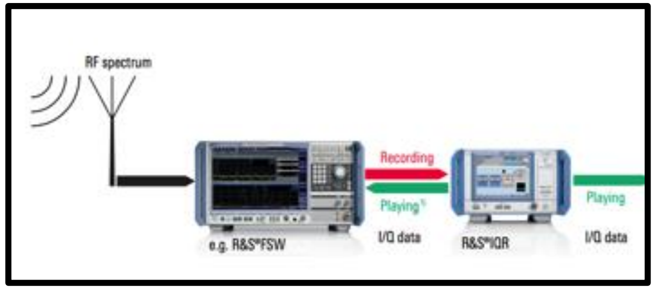
  Trace any signal on Palladium XP

  Capture pre-defined signal for offline viewing on Protium

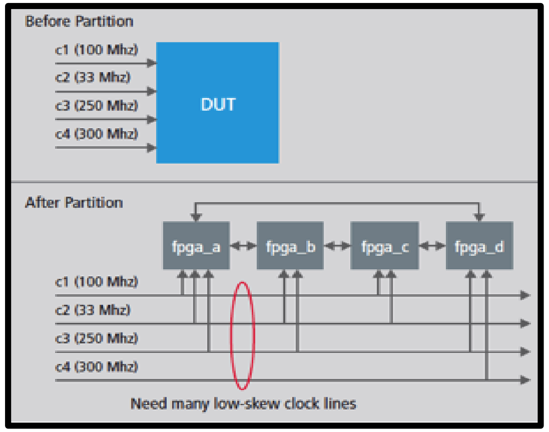
Software Development Effort
(# of days)
🟦 No Pre-Silicon System
🟥 With Pre-Silicon System
15
3
75
15
Full Function Android Running
First Software Release
Amlogic

FPGA-Based Prototyping is Hard to Do …

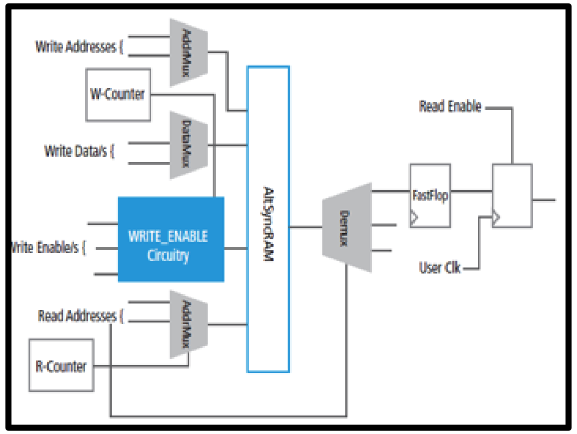
Interfaces

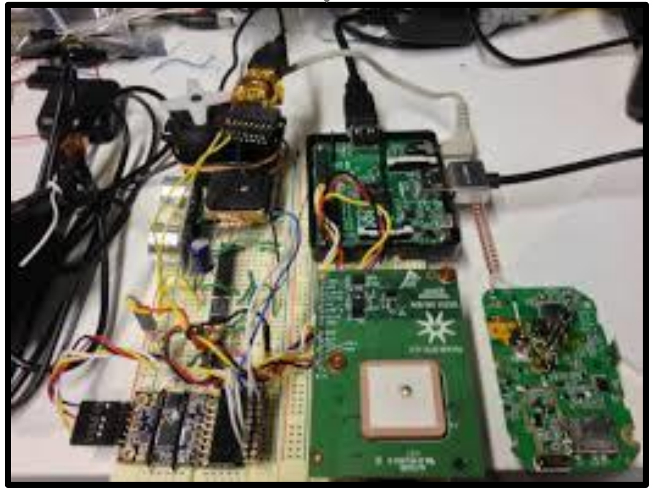


Clocking

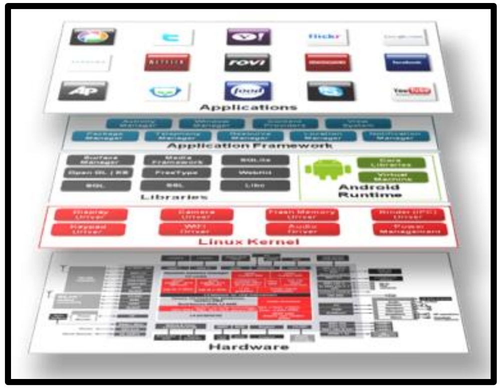


Memories

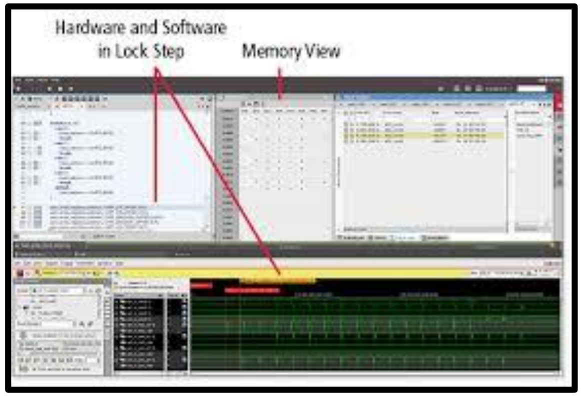




Software



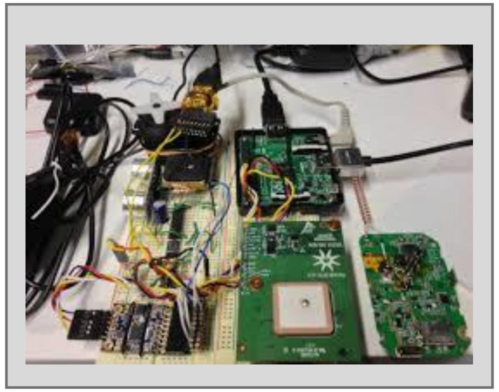
Debug



FPGA-Based Prototyping Is Fragmented

Disjointed, lacking integrated flow and automation

- FPGA-based prototyping

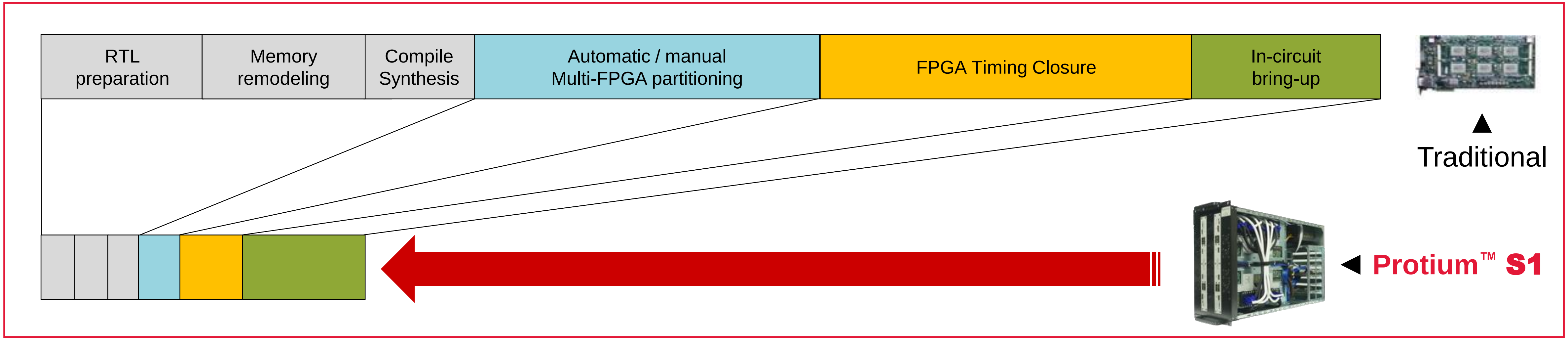


Challenges:

- Fragmented
  - Requires RTL modifications
- Lack auto compilation
  - Memory and clocks
  - Partitioning
- Lack of flow integration
  - Emulation and prototyping
  - Configuration reuse
  - FPGA P&R

Really Hard to Do … Or is It?

Protium S1 – Addressing the prototyping challenges



RTL preparation | Memory remodeling | Compile Synthesis | Automatic / manual Multi-FPGA partitioning | FPGA Timing Closure | In-circuit bring-up

▲ Traditional

◀ Protium™ S1

⬜ No RTL modifications needed

- Clocking / number of clocks
- Automated memory compilation and modeling

🟦 Fully automatic, multi-FPGA partitioning

- Optional manual optimization

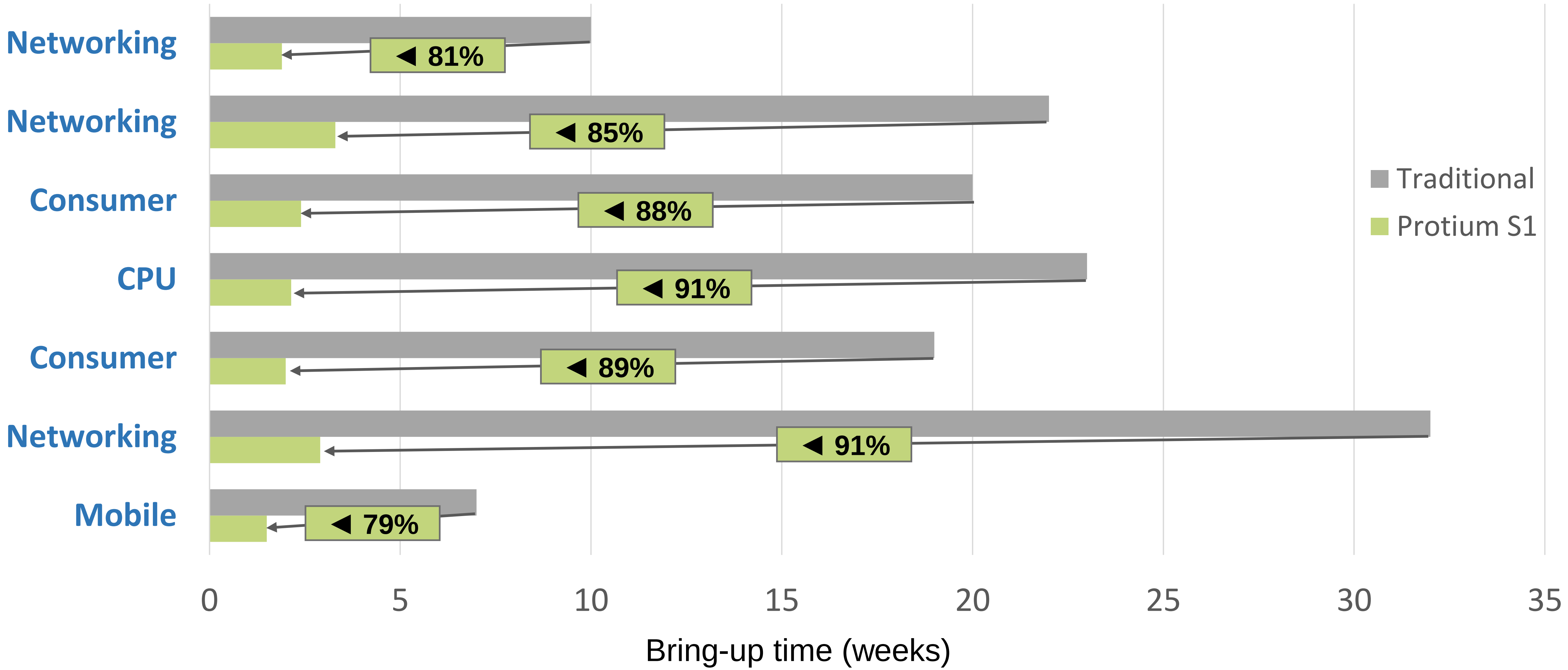
🟨 FPGA Timing Closure

- Multiple design integrations per day
- Avoids time-consuming FPGA P&R

🟩 Fully integrated FPGA P&R

- Automatic constraint generation
- Guaranteed P&R success

Fast Time-to-Prototype (TTP)

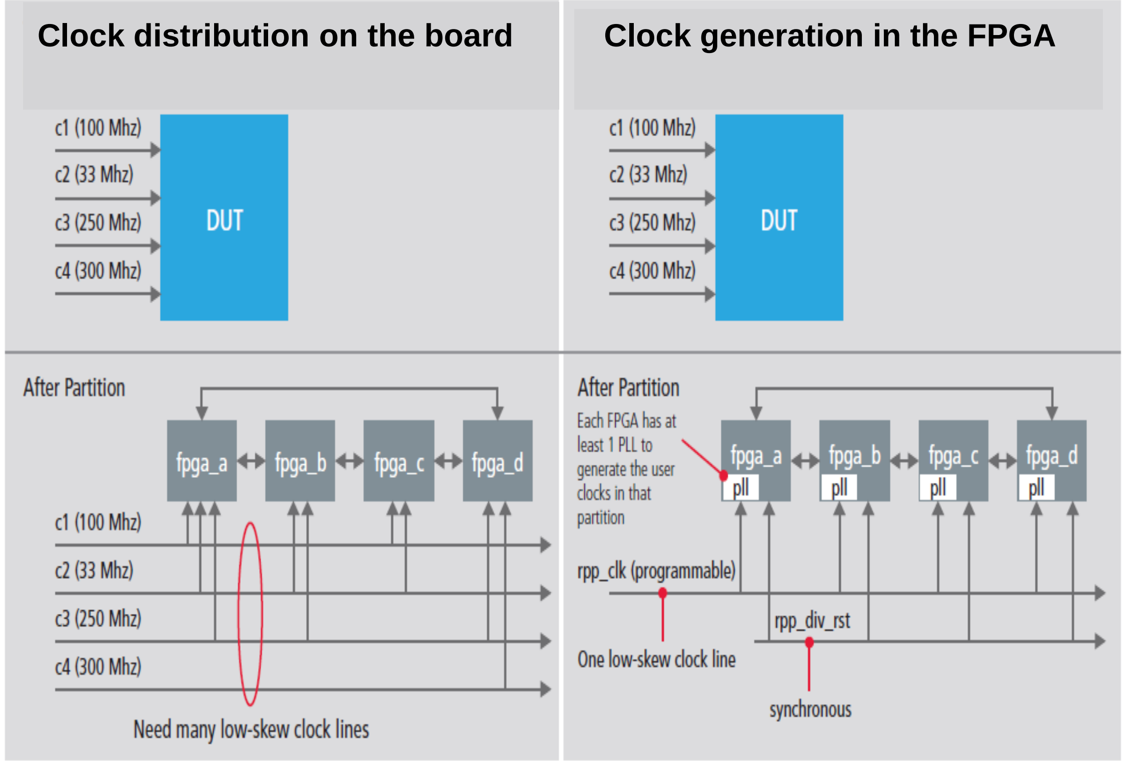


Note: Sample customer bring-up gains over traditional FPGA-based Prototyping solutions

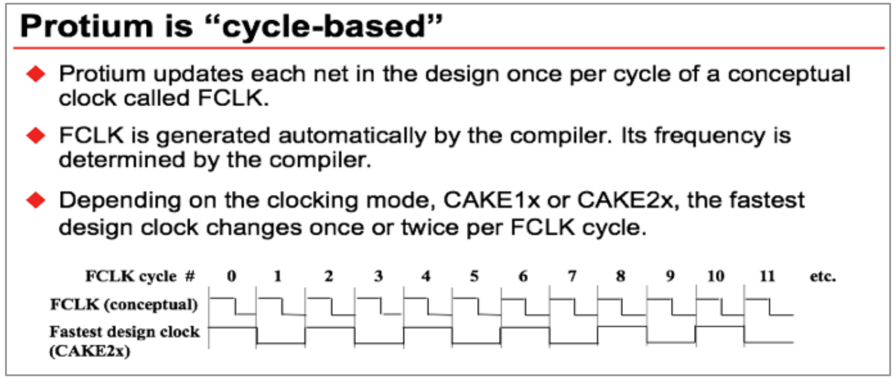
cādence®

No RTL Modifications – Clocking

- Traditional imitations:
  - Gated clock, multiplexed clocks
  - # of clocks
  - **Difficult to achieve FPGA timing closure**
  - **Long iteration times / long FPGA P&R times**
  - **Unpredictable results & prototype behavior**



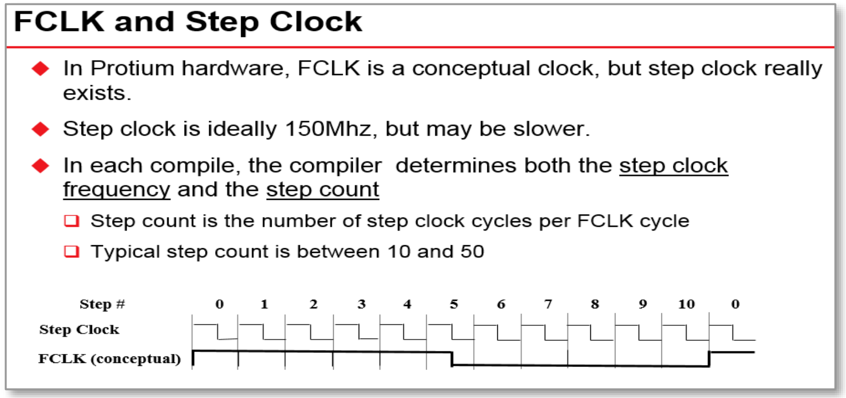
- Protium S1 benefits:
  - No hold-time violations in user clock domains
  - Removes any FPGA-specific clock limitations
  - **Supports unlimited # of design clocks**
  - **Improves FPGA timing closure**
  - **Accelerates FPGA P&R times**



Protium is "cycle-based"

- Protium updates each net in the design once per cycle of a conceptual clock called FCLK.
- FCLK is generated automatically by the compiler. Its frequency is determined by the compiler.
- Depending on the clocking mode, CAKE1x or CAKE2x, the fastest design clock changes once or twice per FCLK cycle.

FCLK cycle # 0 1 2 3 4 5 6 7 8 9 10 11 etc.
FCLK (conceptual)
Fastest design clock (CAKE2x)



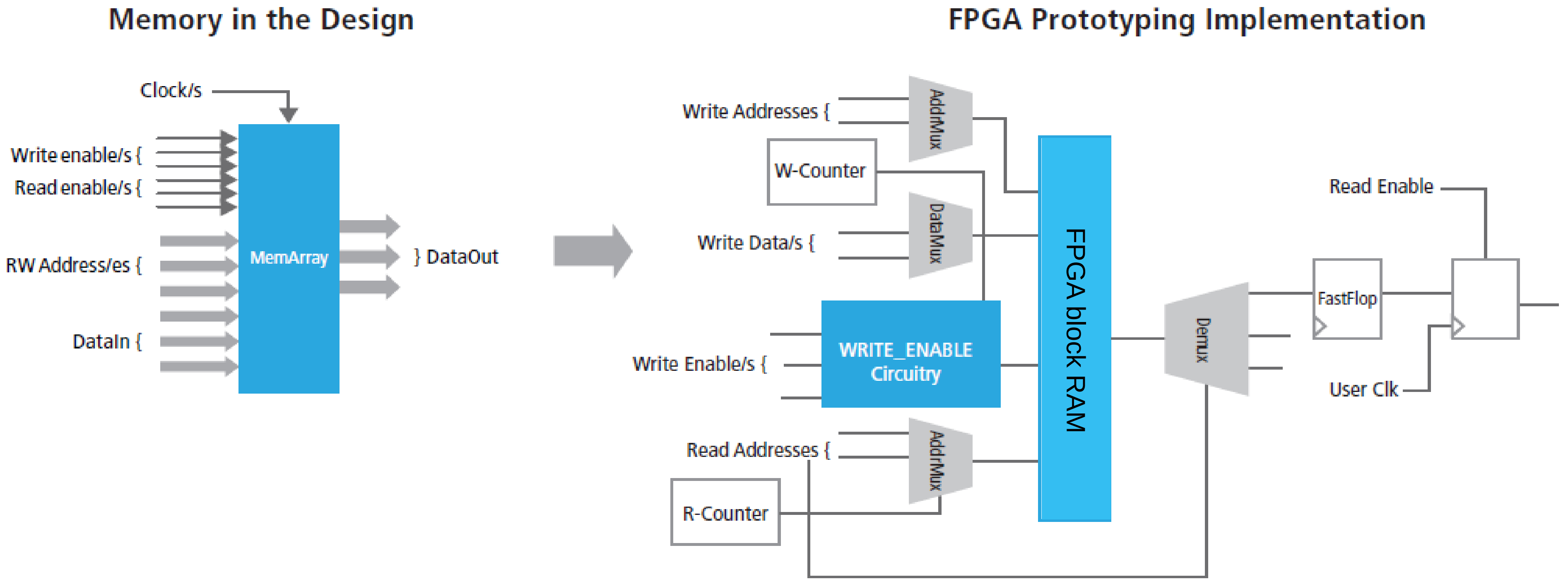
FCLK and Step Clock

- In Protium hardware, FCLK is a conceptual clock, but step clock really exists.
- Step clock is ideally 150Mhz, but may be slower.
- In each compile, the compiler determines both the step clock frequency and the step count
  - ❑ Step count is the number of step clock cycles per FCLK cycle
  - ❑ Typical step count is between 10 and 50

Step # 0 1 2 3 4 5 6 7 8 9 10 0
Step Clock
FCLK (conceptual)

No RTL Modifications - Memories

- **No ASIC RTL changes**
  - Automatic conversion of latches and tri-states
  - Automatic memory compilation and modeling
  - Fully automated clock tree transformation
    - Automatic conversion of gated and multiplexed clocks



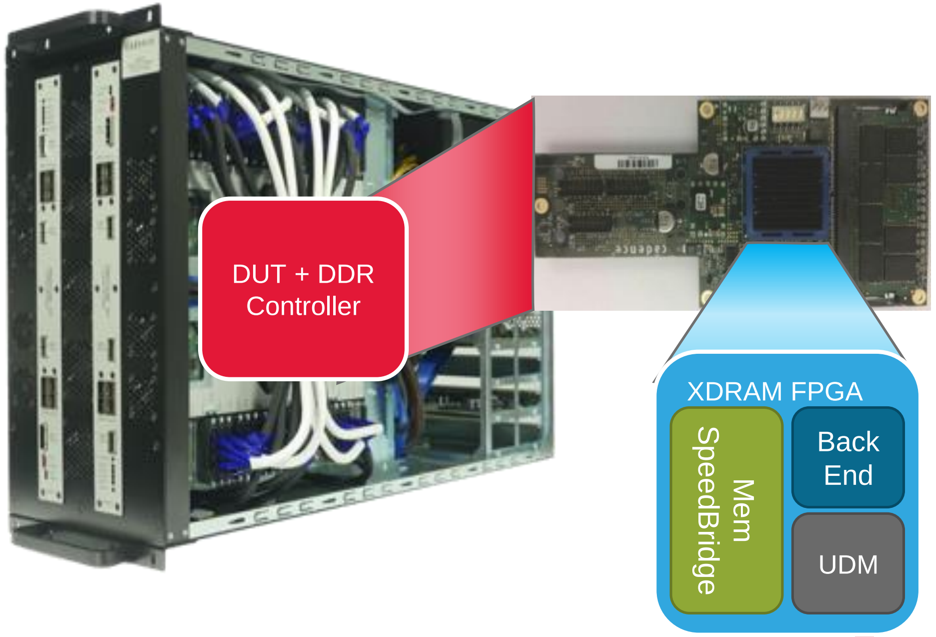
Memory in the Design

FPGA Prototyping Implementation

Innovative XDRAM & XSRAM Solution

- XSRAM
  - Benefits:
    - Increases FPGA internal memory from 80Mbits to 128MBytes (>10x)
    - Automatic mapping of any memory type
    - Support for multi-port memories
    - Support for backdoor upload/download

- XDRAM
  - Benefits:
    - Adds DDRx bulk memories
    - Supports LPDDR2/3/4; DDR3/4; HBM
    - No change to design memory controller and firmware
    - Support for backdoor upload/download
    - Acts as memory SpeedBridge (timing, refresh, etc.)



Comprehensive, automated memory support

Conversion and **implementation** of memories is one of the **most challenging** and **time-consuming** steps in bring-up of an FPGA-based prototype (often taking many weeks to complete).

Type	Size	Palladium MMP	Upload/ Download	Perform.	Comments
FPGA-internal	~50Mbits / FPGA	Yes	Yes	Full design speed	• Fully automatic compile
XSRAM (automated small external memory)	128 Mbytes per memory card	Some	Yes	Full design speed	• Fully automatic compile • Extends 'FPGA-internal' memory to external SRAM • Useful for Serial Parallel Interface (SPI)-flash and other small memories (e.g. boot ROM)
XDRAM (automated bulk memory)	8/16 GBytes per XDRAM card	DDR family models	Yes	<8MHz	• semi automatic compile • Leverages XDRAM hardware Support for DDR3/4, LPDDR3/4
DCMC (Direct Connected Memory Card)	x GBytes (depending on memories used)	No	No	Full design speed	• Design change may be required, depending on memory type • App notes available
FCMC (Full-custom Memory Card)	Custom	No	No	Full design speed	• Fully custom development

Protium S1
Memory compile capabilities :

– Smaller memories are **automatically compiled** into FPGA-internal resources

– For larger, off-FPGA memories, the Protium platform offers **several automated solutions**, see table

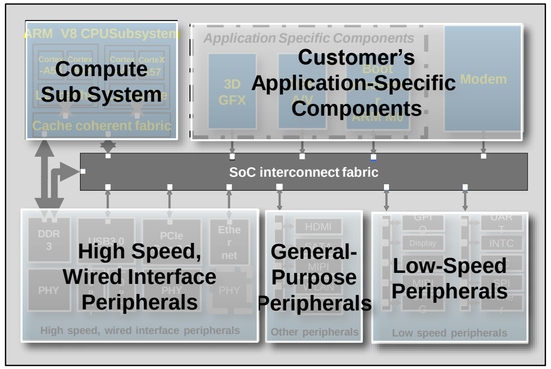
Hardware and Software Debug

- Waveforms across partitions
  - Design-centric view vs. FPGA-centric
- Force/release
  - predefined signals (at compile time) to “0” or “1” during runtime
- Monitor signal
  - real-time monitoring of predefined (at compile time) signals

Software

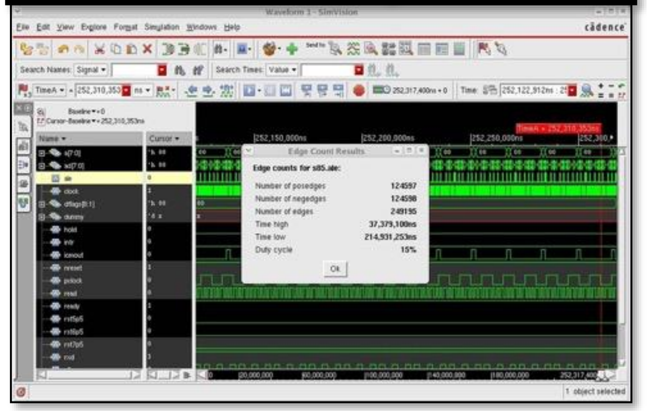
- Applications
- Middleware
- Operating Systems (OS)
- Drivers
- Firmware / HAL

SoC, Sub-System or IP



- Backdoor memory access
  - quickly change boot code, software, etc.
- Clock control
  - start/stop the clock on demand
- Fully scriptable runtime environment
- Remote access
  - Network resource anytime from anywhere
- High-performance link to software model

Hardware Debug: RTL



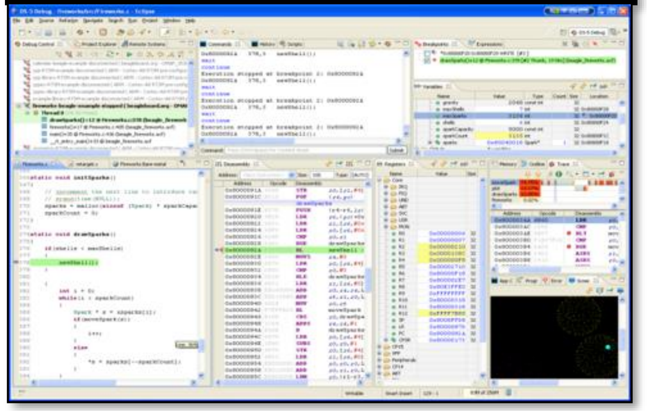
Edge counts for s05.aie:

Number of posedges 124597
Number of negedges 124598
Number of edges 249195
Time high 37,379,100ns
Time low 214,931,253ns
Duty cycle 15%

Probes

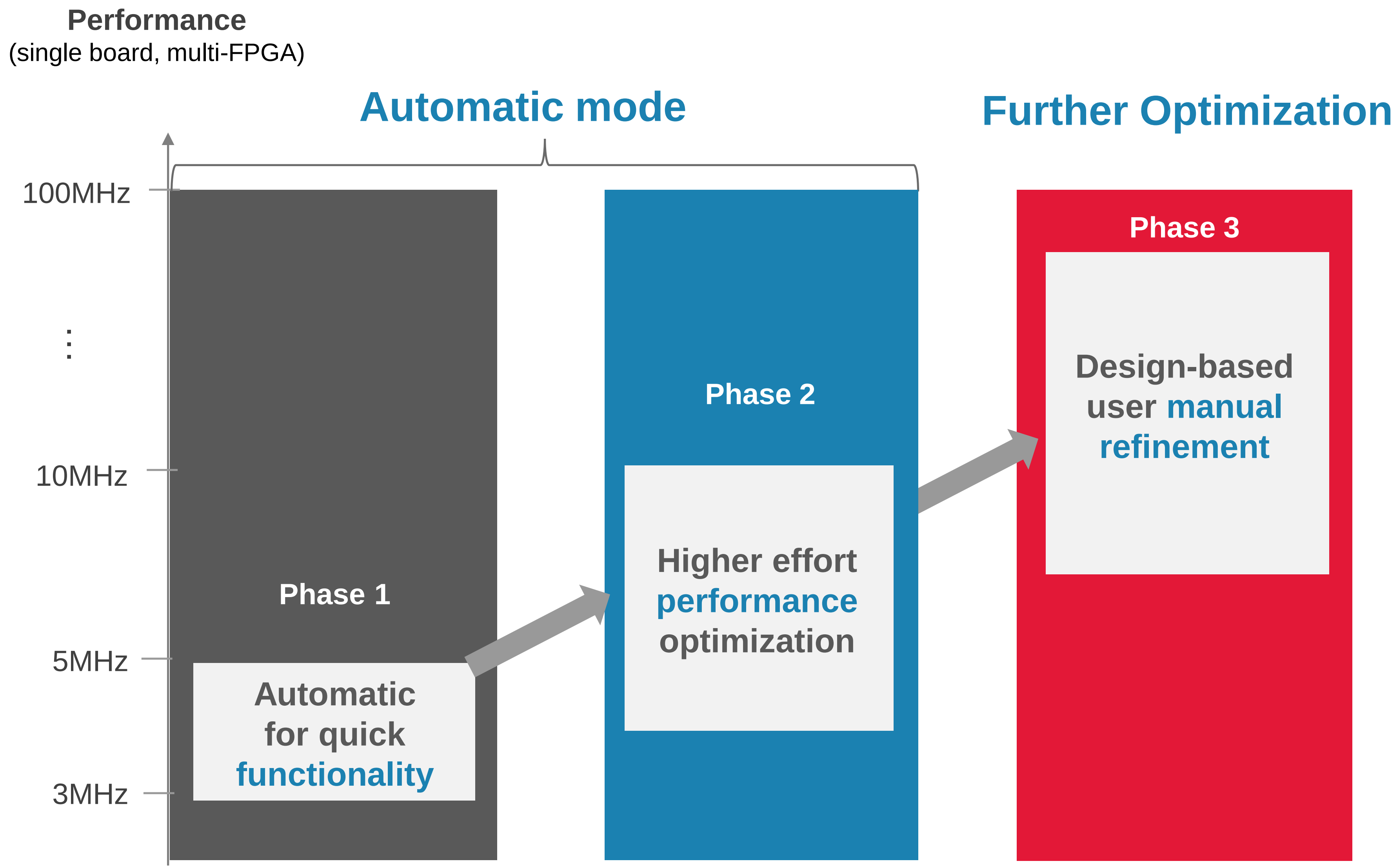
JTAG

Software Debug: C Code



Daughter-cards & peripherals

Scalable performance



Performance
(single board, multi-FPGA)

Automatic mode

Further Optimization

Phase 1

Automatic for quick functionality

Phase 2

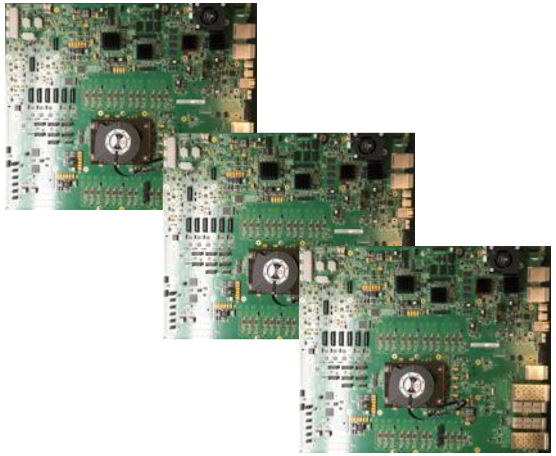
Higher effort performance optimization

Phase 3

Design-based user manual refinement

Scalable hardware

Protium S1-G



Single board

- 1 FPGA
- Up to 25M ASIC gates
- Affordable and scalable
- Highest performance
- Early software development
- IP verification

Protium S1-SC

Single chassis system

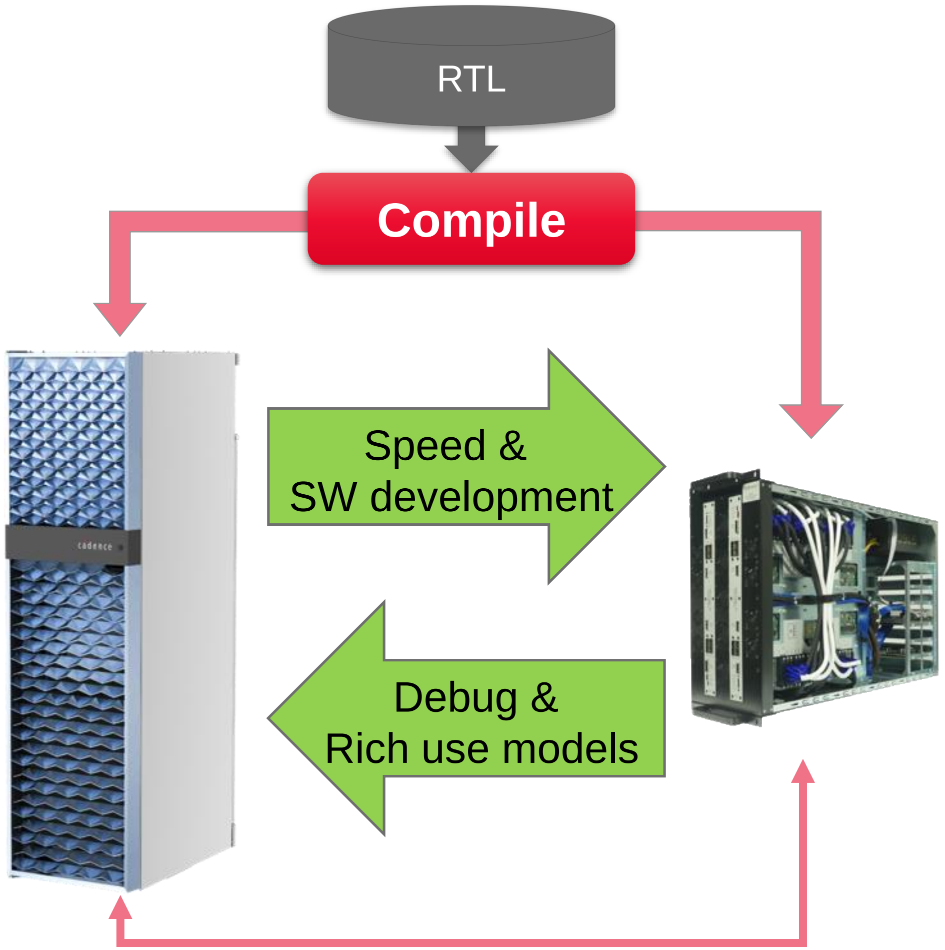
- 2-8 FPGAs
- Up to 200M ASIC gates
- Flexible and scalable
- Fastest bring-up
- Unique SW debug capabilities
- Early software development
- HW/SW integration

Protium S1-MC (2H 2017)

Multi-chassis configuration

- 8-24 FPGAs
- Up to 600M ASIC gates
- Highest capacity
- Flexible use modes
- Advanced debug
- High performance regression
- Full system validation

Palladium Z1 + Protium S1



Palladium® Z1

- Best debug
- SoC integration & accel
- Rich use models

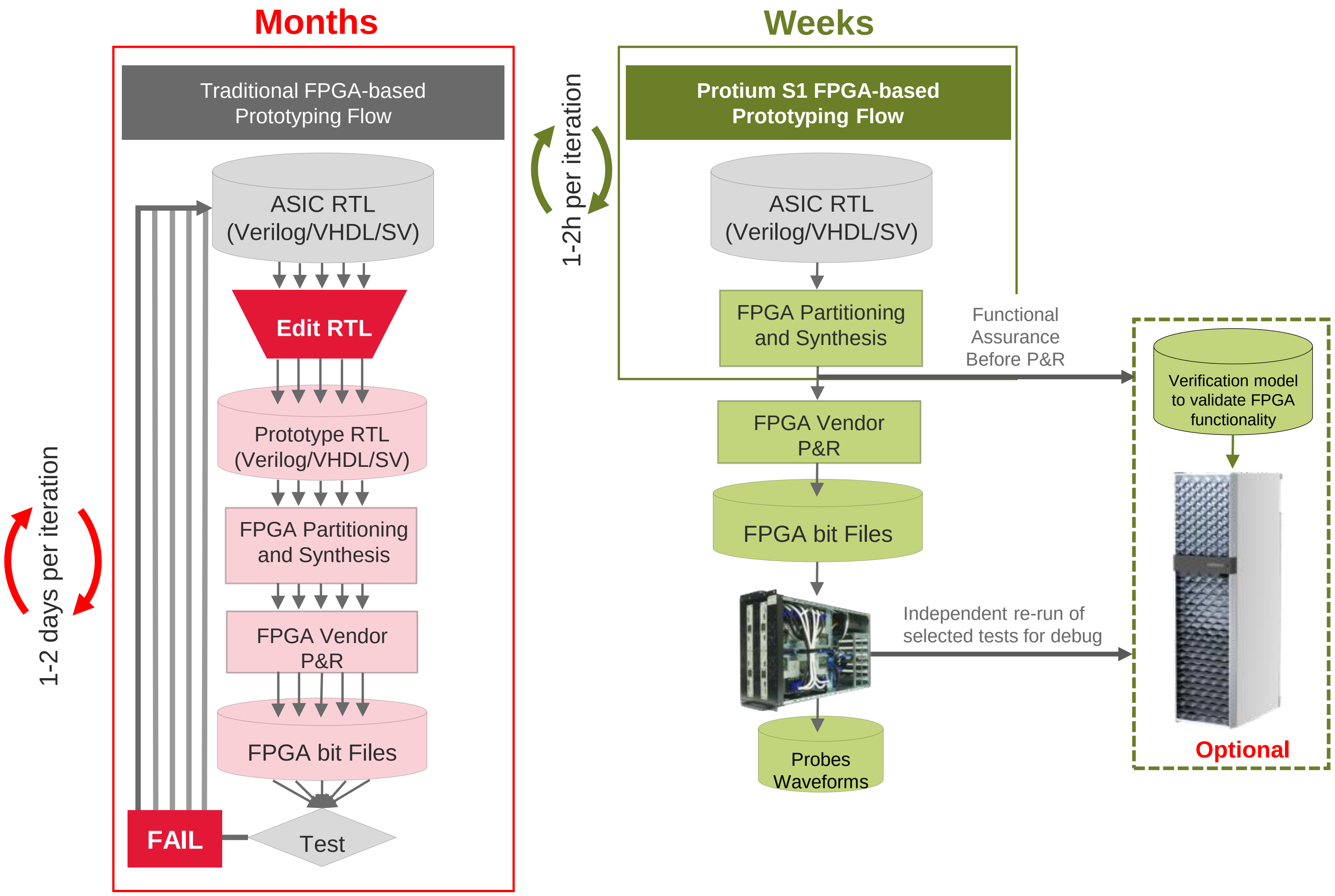
Protium™ S1

- Highest performance
- SW development
- Regressions

Congruency and **common environment**

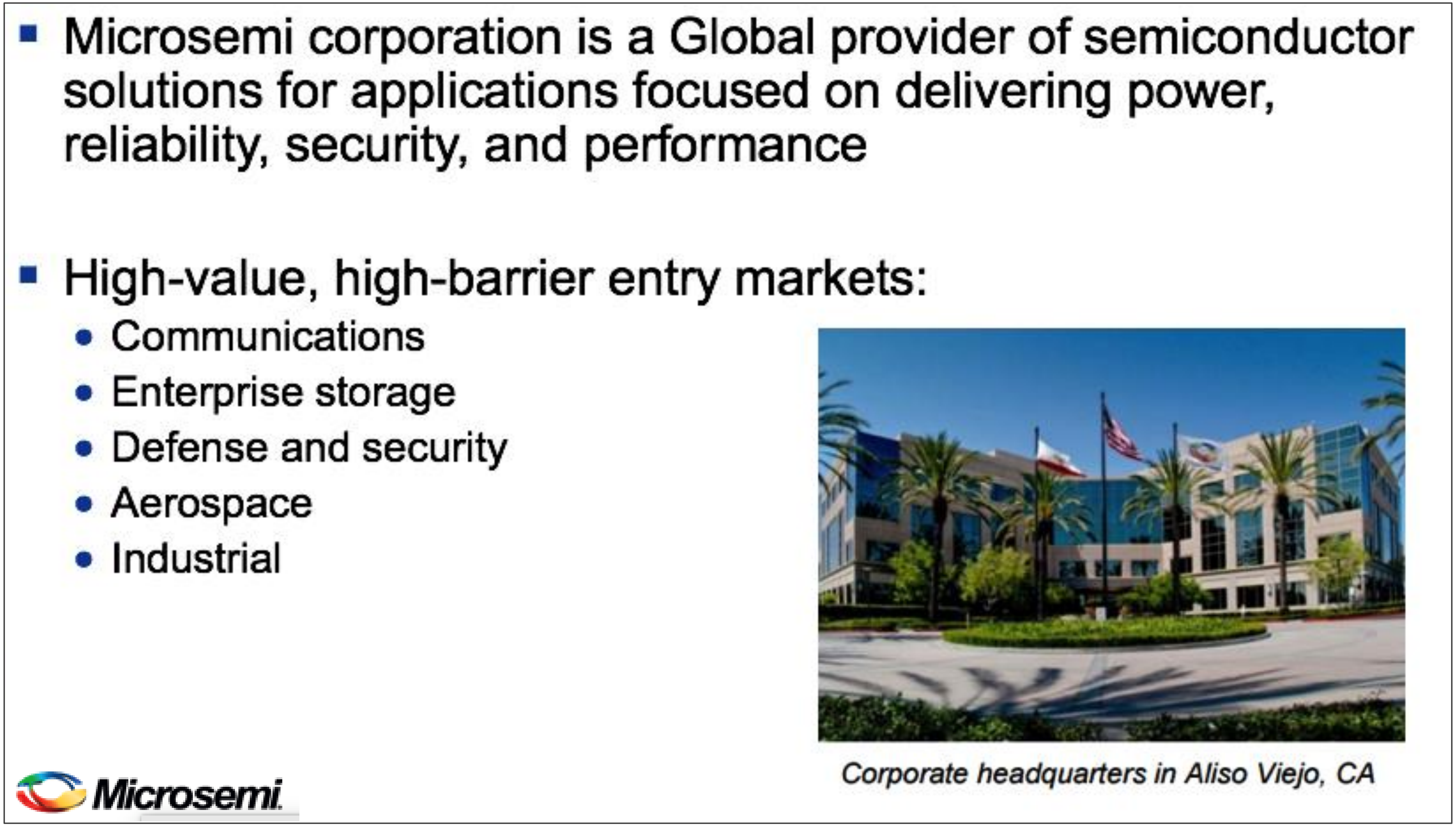
Protium S1 Fully Integrated Implementation Flow

- **Automated prototyping flow reduces time-to-prototype (TTP) from months to weeks**
    - Design changes have much lesser time impact on iterations
    - Simpler single pass flow iterations are run in hours not days
    - Software development gets a head start measured in months not days



Months
Traditional FPGA-based Prototyping Flow
ASIC RTL (Verilog/VHDL/SV)
Edit RTL
Prototype RTL (Verilog/VHDL/SV)
FPGA Partitioning and Synthesis
FPGA Vendor P&R
FPGA bit Files
Test
FAIL
1-2 days per iteration
Weeks
Protium S1 FPGA-based Prototyping Flow
1-2h per iteration
ASIC RTL (Verilog/VHDL/SV)
FPGA Partitioning and Synthesis
Functional Assurance Before P&R
FPGA Vendor P&R
FPGA bit Files
Independent re-run of selected tests for debug
Probes Waveforms
Verification model to validate FPGA functionality
Optional

Case study: Microsemi

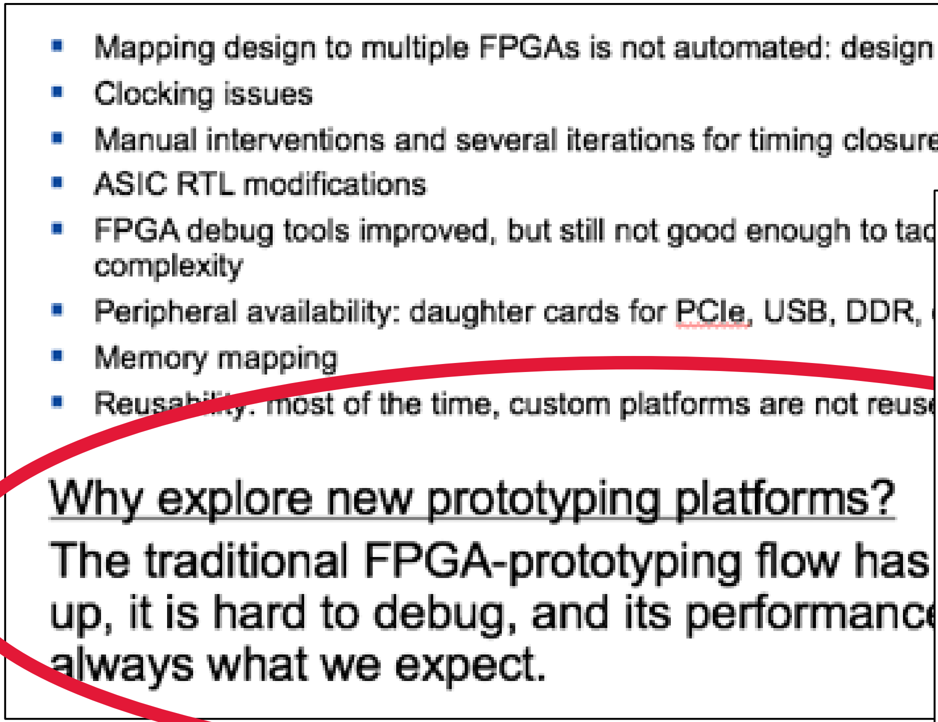


- Microsemi corporation is a Global provider of semiconductor solutions for applications focused on delivering power, reliability, security, and performance

- High-value, high-barrier entry markets:
  - Communications
  - Enterprise storage
  - Defense and security
  - Aerospace
  - Industrial

Corporate headquarters in Aliso Viejo, CA

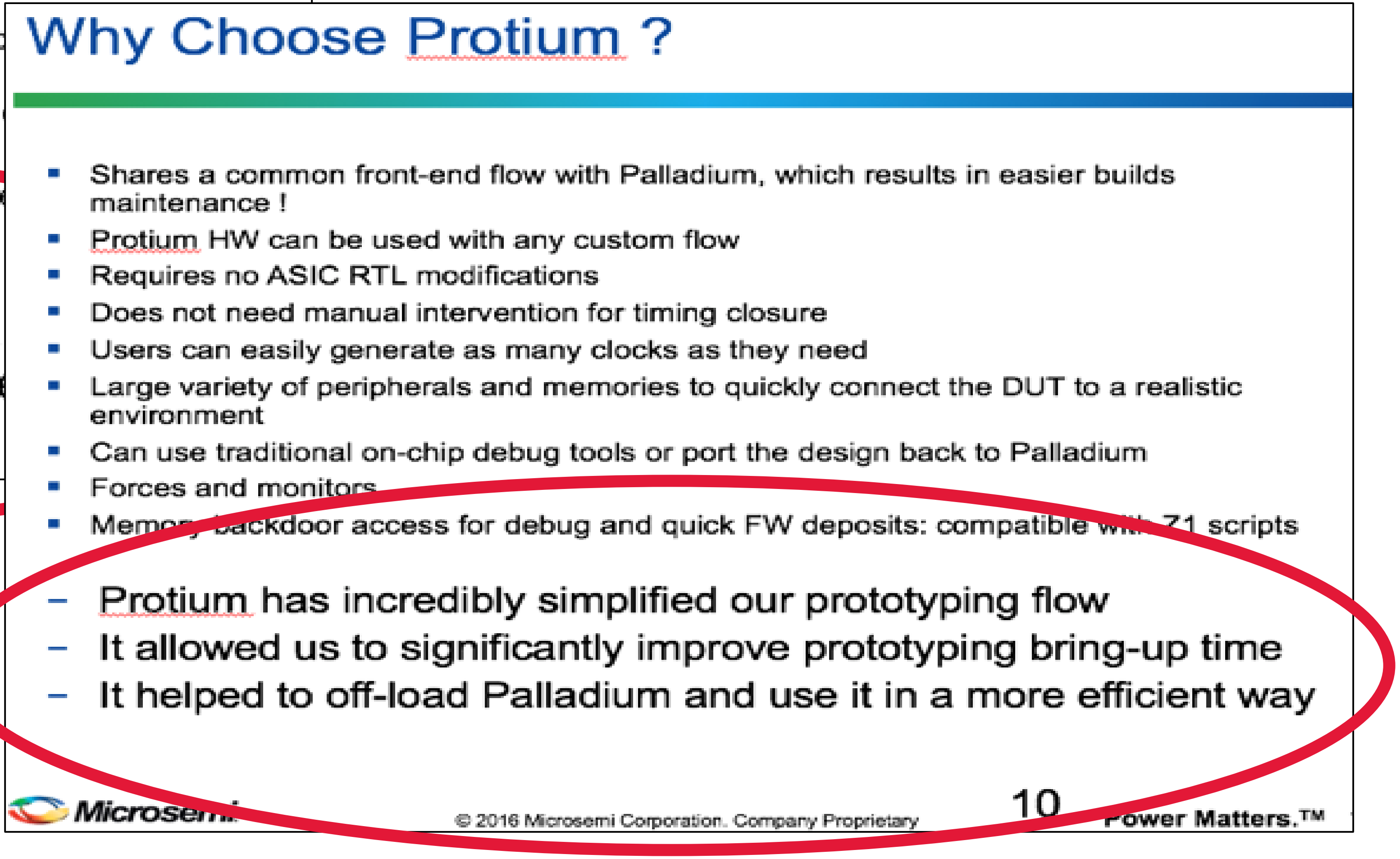
Case study: Microsemi (cont.)



- Mapping design to multiple FPGAs is not automated: design partitioning
- Clocking issues
- Manual interventions and several iterations for timing closure
- ASIC RTL modifications
- FPGA debug tools improved, but still not good enough to tac complexity
- Peripheral availability: daughter cards for PCIe, USB, DDR, (
- Memory mapping
- Reusability: most of the time, custom platforms are not reuse

Why explore new prototyping platforms?

The traditional FPGA-prototyping flow has up, it is hard to debug, and its performance always what we expect.



Why Choose Protium ?

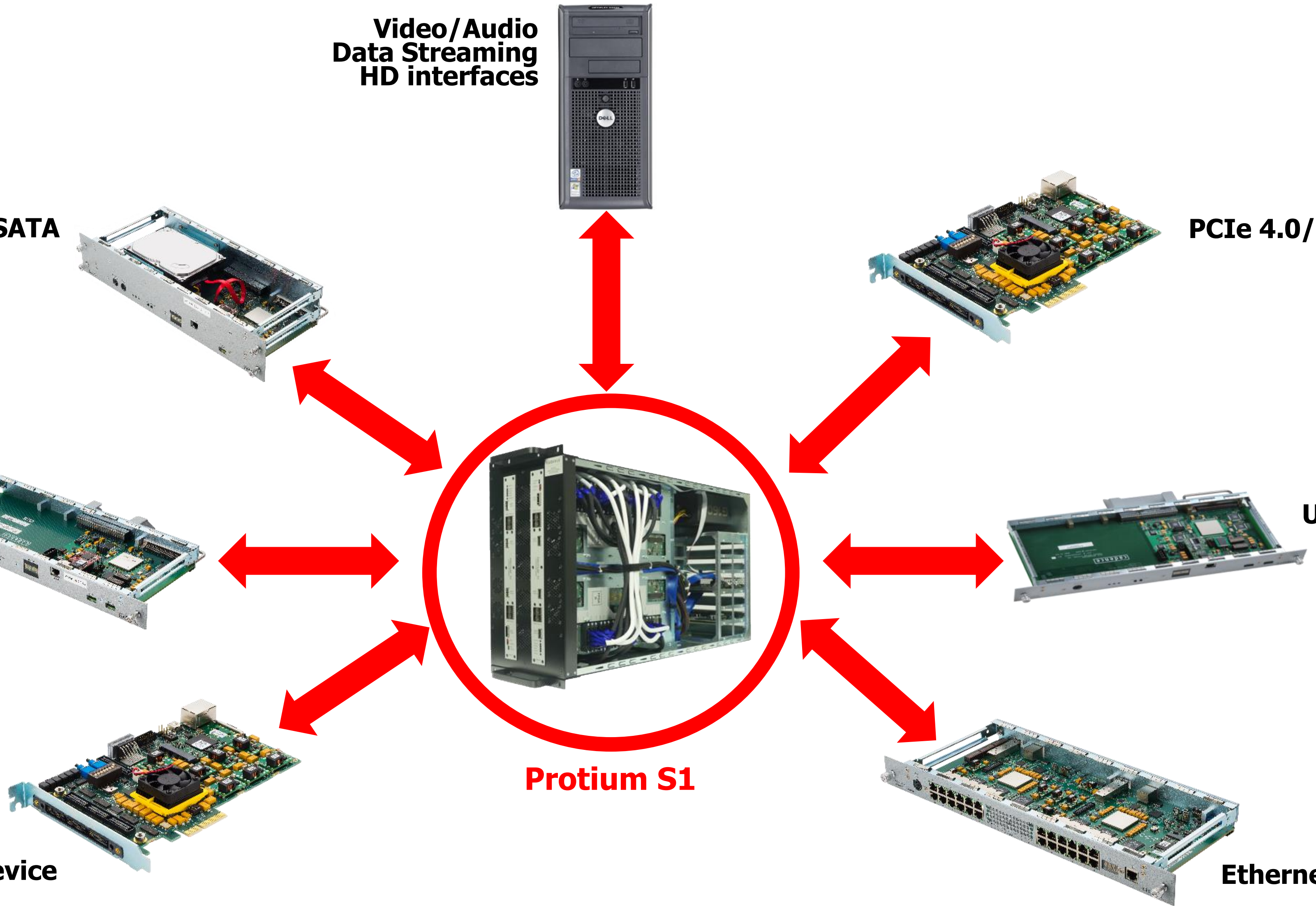
- Shares a common front-end flow with Palladium, which results in easier builds maintenance !
- Protium HW can be used with any custom flow
- Requires no ASIC RTL modifications
- Does not need manual intervention for timing closure
- Users can easily generate as many clocks as they need
- Large variety of peripherals and memories to quickly connect the DUT to a realistic environment
- Can use traditional on-chip debug tools or port the design back to Palladium
- Forces and monitors
- Memory backdoor access for debug and quick FW deposits: compatible with Z1 scripts

– Protium has incredibly simplified our prototyping flow
– It allowed us to significantly improve prototyping bring-up time
– It helped to off-load Palladium and use it in a more efficient way

Microsemi © 2016 Microsemi Corporation. Company Proprietary 10 Power Matters.™

Cadence SpeedBridge Adapter Solutions

Palladium® Series



Video/Audio
Data Streaming
HD interfaces

SATA

PCIe 4.0/3.0/2.0/1.1/1.0a

SAS

USB 2.0 Host

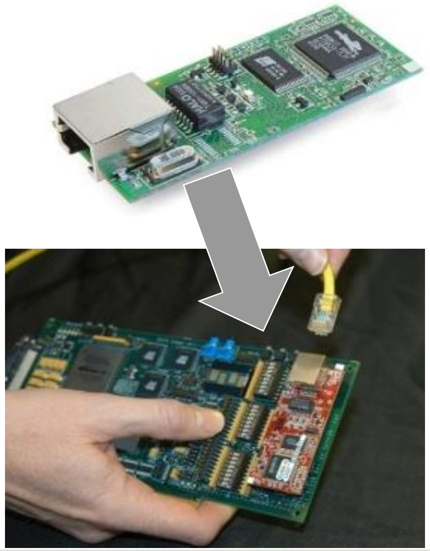
Protium S1

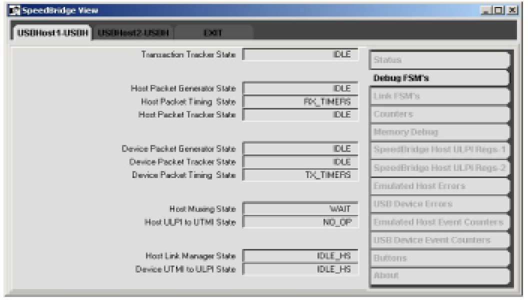
USB 3.0/2.0/1.1 Device

Ethernet 10/100, 1G, 10G, 40G

SpeedBridge Configuration Manager (SCM)

- Error injection during System-level test*
- The SpeedBridge™ Configuration Manager components
  - – Hardware: SCM; Software/GUI: SpeedBridge view
- Benefits of Remote Configurability
  - – Confirmation of setup and configuration
    - – A complete environment for remote monitoring & use
      - – Protocol specific debug and SpeedBridge information
      - – DIP switches, LED status
      - – Remote SpeedBridge reset capability
  - – Ability to monitor multiple SpeedBridges from a single GUI





Transaction Tracker State IDLE

Host Packet Generator State IDLE
Host Packet Timing State RX_TIMERS
Host Packet Tracker State IDLE

Device Packet Generator State IDLE
Device Packet Tracker State IDLE
Device Packet Timing State TX_TIMERS

Host Muxing State WAIT
Host ULPI to UTMI State NO_OP

Host Link Manager State IDLE_HS
Device UTMI to ULPI State IDLE_HS

*New! – (EA) capability available on some of the SpeedBridge adapters

Comprehensive portfolio of accessories and interfaces

- Please see datasheet for comprehensive list of what is natively supported.
  - https://www.cadence.com/content/dam/cadence-www/global/en_US/documents/tools/system-design-verification/protium-s1-fpga-based-prototyping-platform-ds.pdf

Protium S1 – the Most Efficient Way to Prototype Your SoC

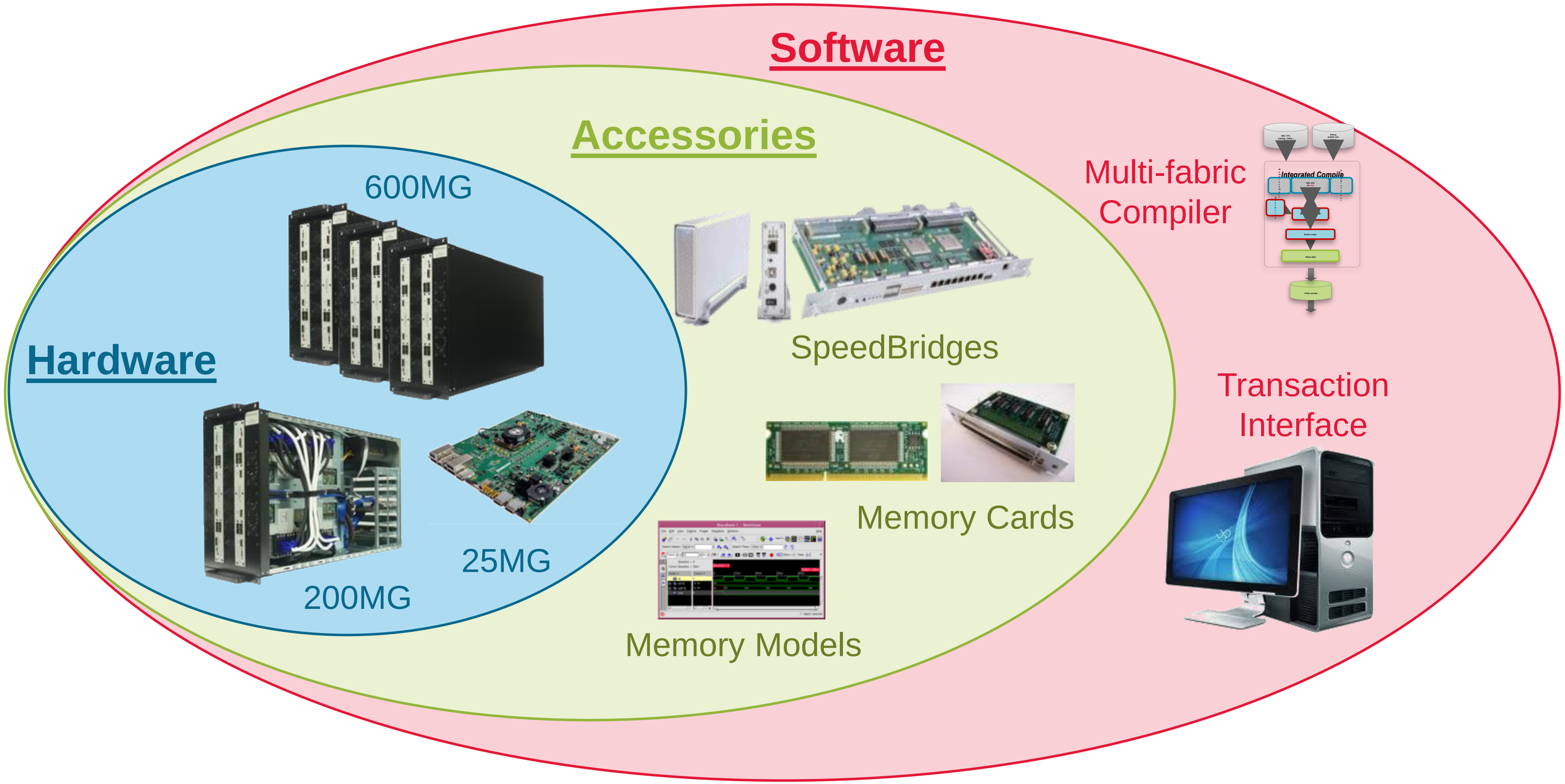
- **Fast time-to-prototyping** (months to 1-2 weeks)
  - No RTL changes
  - Automatic partitioning/memory compilation
  - Fully integrated FPGA place-and-route

- **Scalable performance** (3-100MHz)
  - From fully automatic to fully manual
  - Advanced black-box methodology

- Advanced **software debug**
  - Memory upload/download
  - Force and release
  - Data Capture card
  - Assertion checkers
  - State read-back

Protium S1 Prototyping Solution

Industry’s first comprehensive, fully integrated solution



Software

Accessories

Hardware

600MG

200MG

25MG

SpeedBridges

Memory Cards

Memory Models

Multi-fabric Compiler

Transaction Interface

… and Xilinx Likes it Too

“The Cadence Protium S1 platform ensures scalability to hundreds of software developers at the earliest possible point during the development flow, and allows developers to focus on design validation and software development rather than prototype bring-up.

The common flow with the Cadence Palladium Z1 emulation platform enables a smooth transition from emulation to prototyping, which greatly improves productivity.”

Peter Ryser, Senior Director for System Software, Integration, and Validation, Xilinx

cadence®