Accelerating Embedded Software Development
Protium - Ease-of-Use in FPGA-based Prototyping

Juergen Jaeger, Director, Product Management

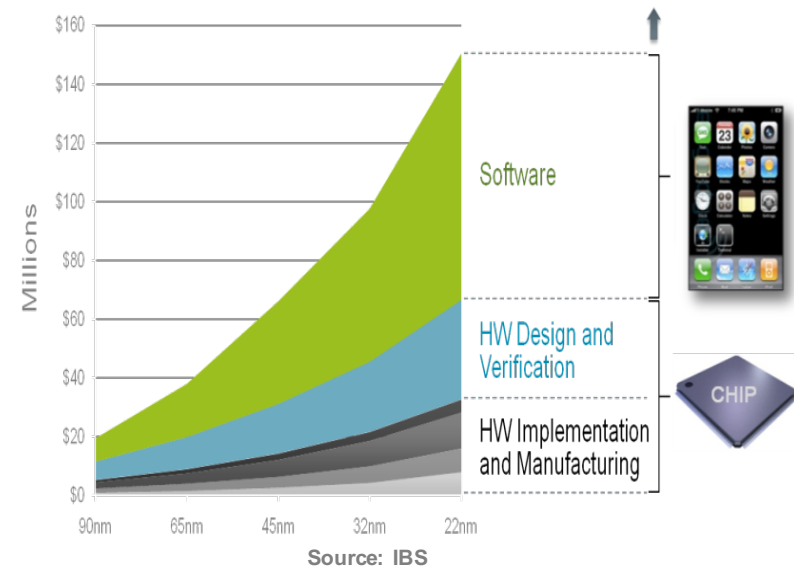September 2015

cādence®

It's All About The Software

- **Getting to chip tape-out … is not the biggest problem anymore!**
  - Software dominates development cost and schedules
  - Software is a necessity to successfully sell silicon
  - Delay of software delivery delays time to revenue for semiconductor providers

10	Design Engineers
20	Verification Engineers
10	Test Engineers
200	Software Engineers
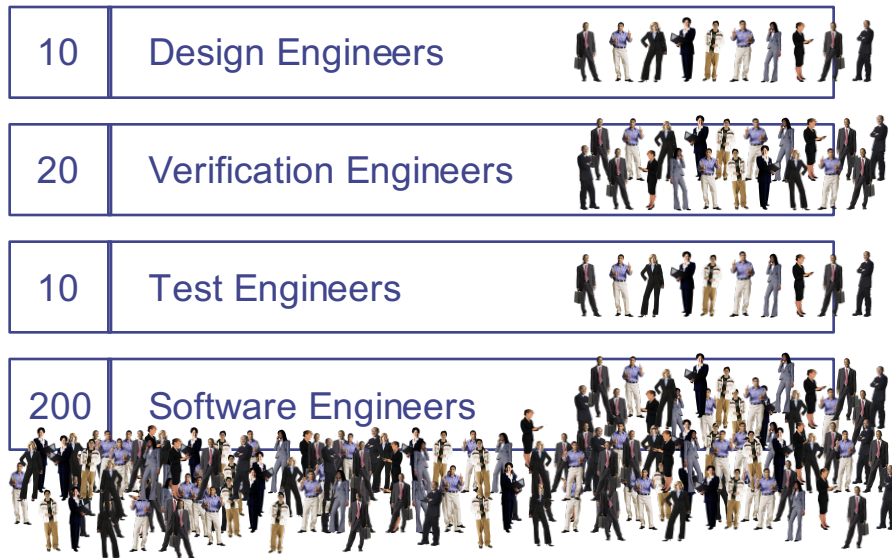
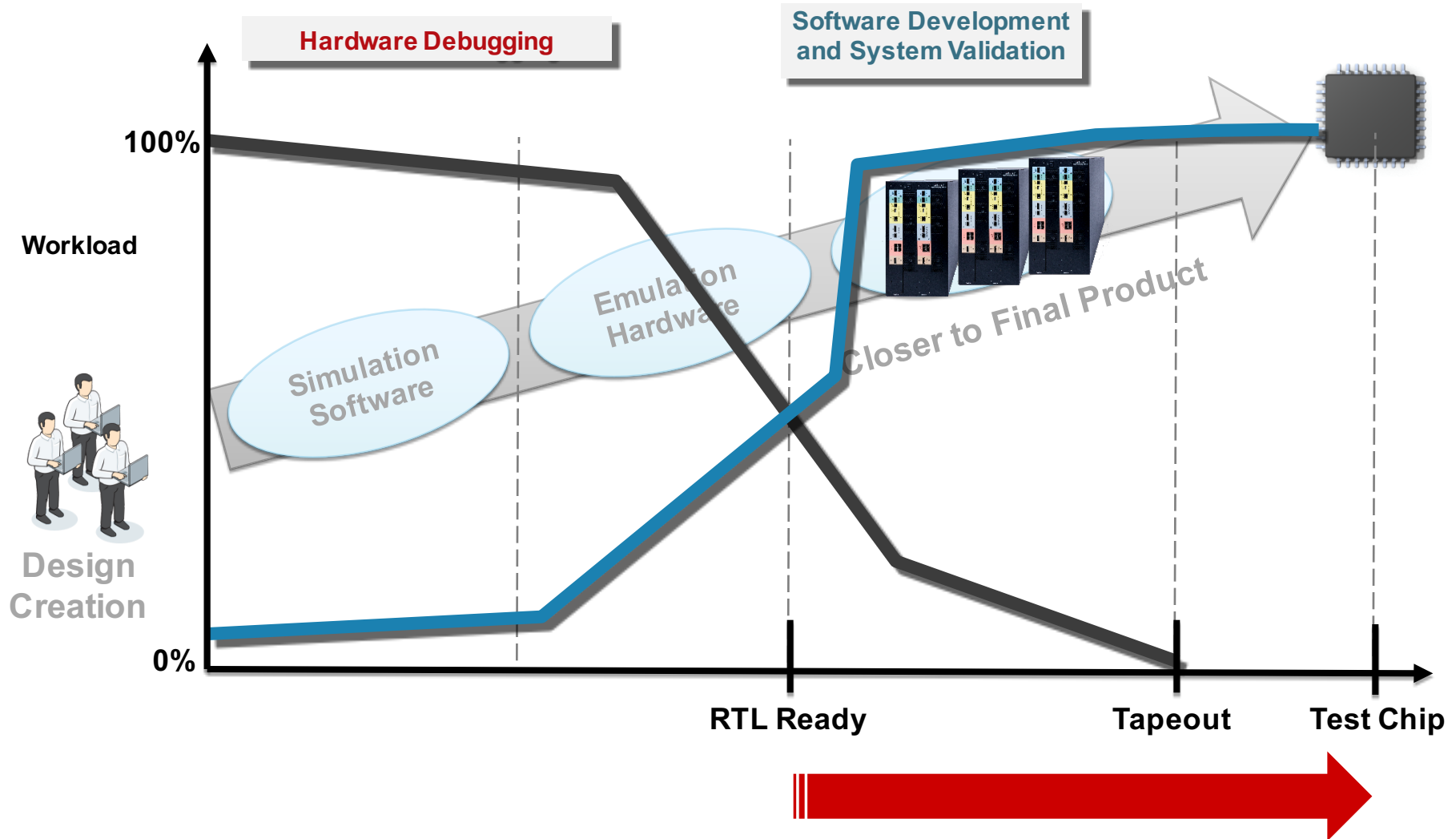Source: Raouf Halim, CEO Mindspeed, 2010 SoC Conference

Source: IBS

cādence®

SW/FW Development Care About

- Ease of use
  - Network resource
  - "one click" download, configure, run
  - Reset, re-run

- Productive
  - Backdoor memory access
  - Start/stop clock
  - High performance

- (SW) flow integration
  - JTAG
  - UART
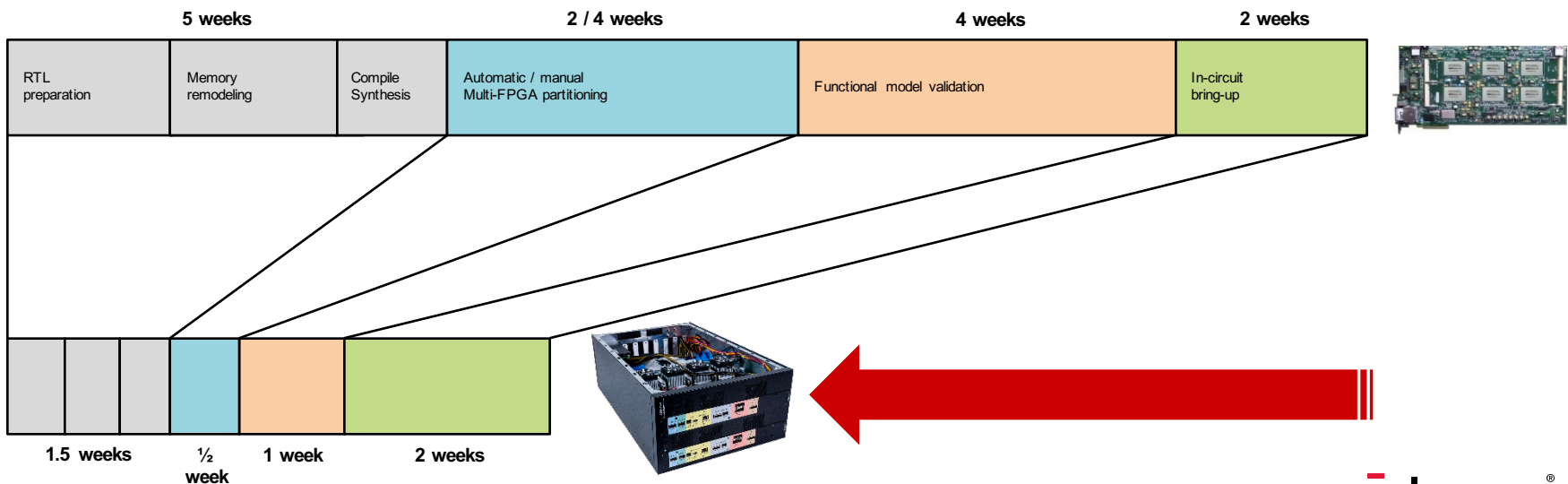 - Design-specific interfaces (e.g. PCIe, Ethernet, etc.)

What else?

cādence®

Time to Prototype
Key to success

Hardware Debugging

Software Development and System Validation

100%

Workload

Simulation Software

Emulation Hardware

Closer to Final Product

Design Creation

0%

RTL Ready

Tapeout

Test Chip

cādence®
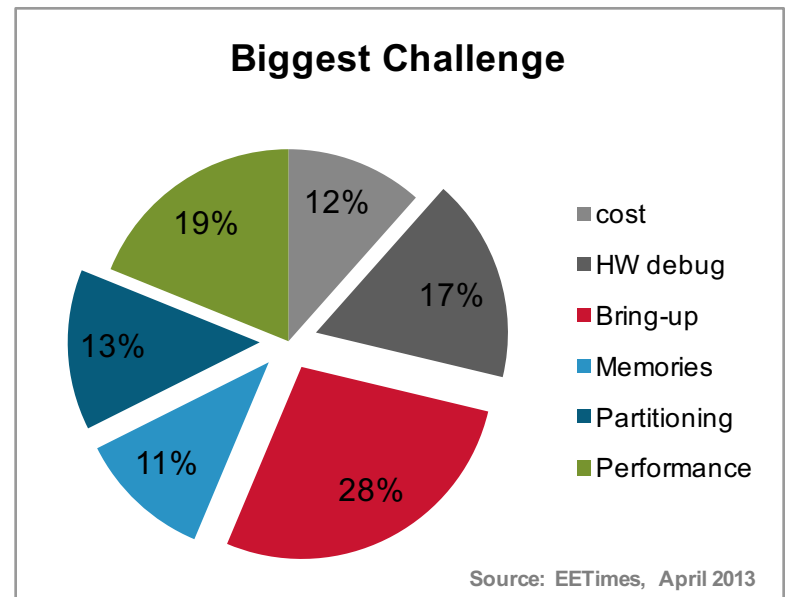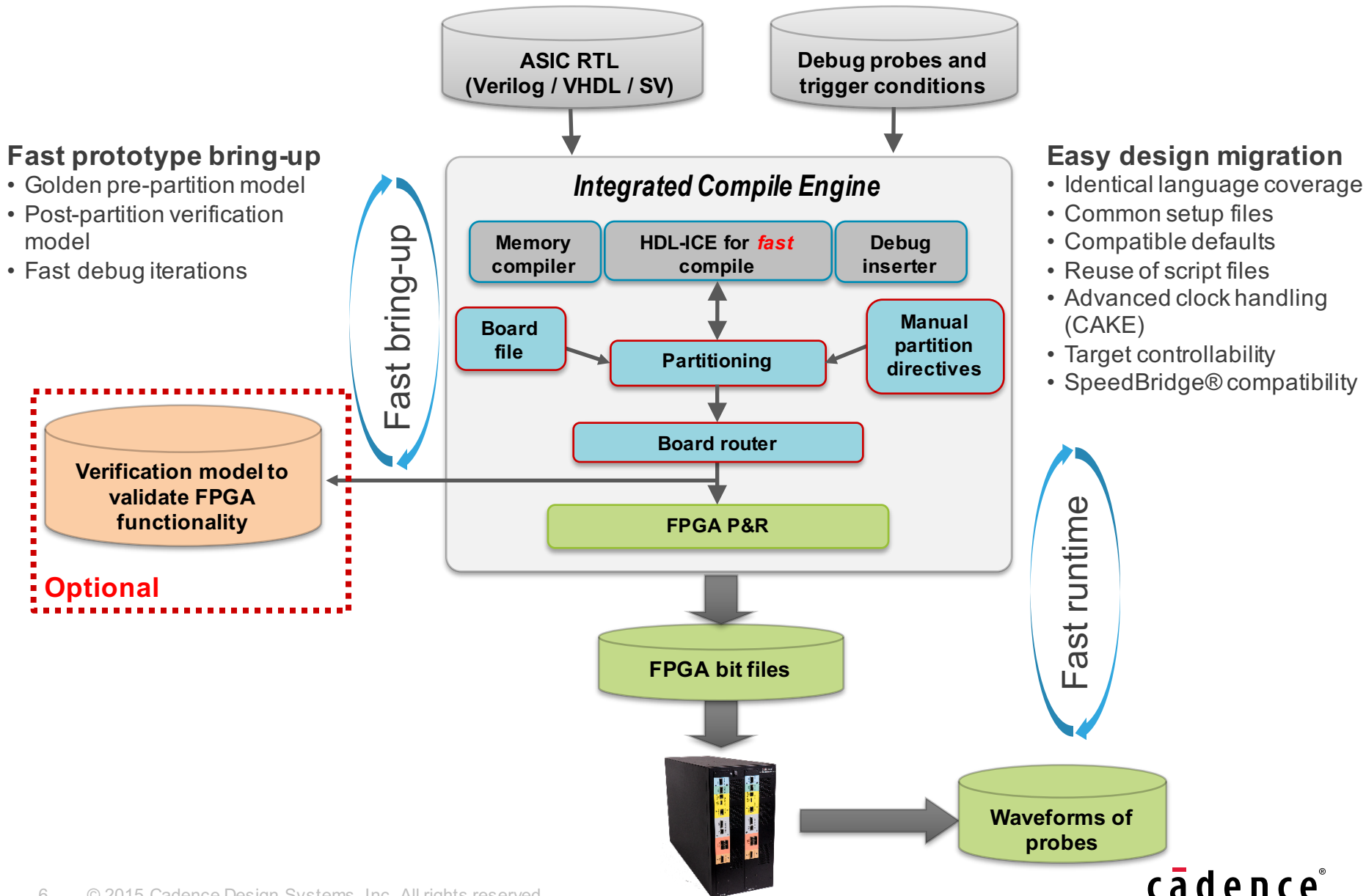
Prototyping User Challenges

- FPGA-based prototype bring-up can be both challenging and time consuming
- Unique capabilities in the following areas are addressing these challenges:
  1. Design import and compilation
  2. Memory support and modeling
  3. Clock handling
  4. Multi-FPGA partitioning
  5. Functional validation and debug (bring-up)
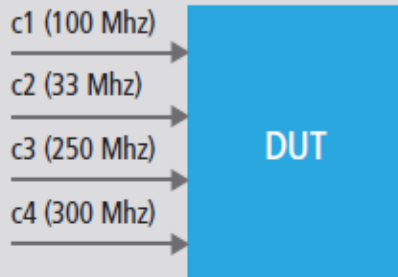 6. Runtime debug
  7. Flow integration

Biggest Challenge

■ cost	12%
■ HW debug	17%
■ Bring-up	28%
■ Memories	11%
■ Partitioning	13%
■ Performance	19%

Source: EETimes, April 2013

5 weeks			2 / 4 weeks	4 weeks	2 weeks
RTL preparation	Memory remodeling	Compile Synthesis	Automatic / manual Multi-FPGA partitioning	Functional model validation	In-circuit bring-up

1.5 weeks   ½ week   1 week   2 weeks

cādence®

Protium Compile Flow

ASIC RTL (Verilog / VHDL / SV)

Debug probes and trigger conditions

Fast prototype bring-up
- Golden pre-partition model
- Post-partition verification model
- Fast debug iterations

Integrated Compile Engine

Memory compiler	HDL-ICE for *fast* compile	Debug inserter

Board file

Partitioning

Manual partition directives

Board router

FPGA P&R

Fast bring-up

Verification model to validate FPGA functionality

Optional

Easy design migration
- Identical language coverage
- Common setup files
- Compatible defaults
- Reuse of script files
- Advanced clock handling (CAKE)
- Target controllability
- SpeedBridge® compatibility

FPGA bit files

Fast runtime

Waveforms of probes

cādence®

Traditional Clocking Approach

cādence®
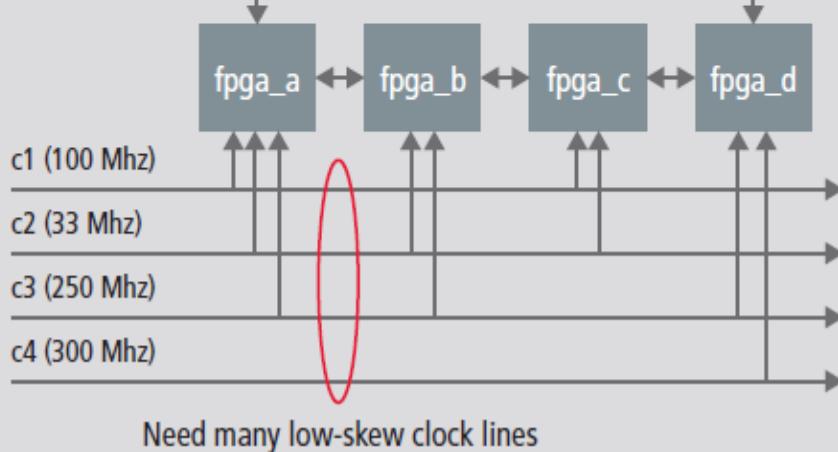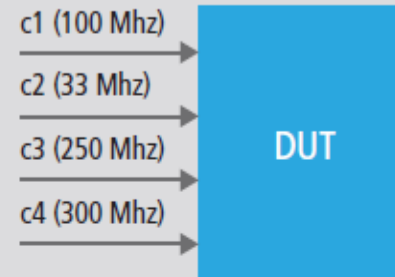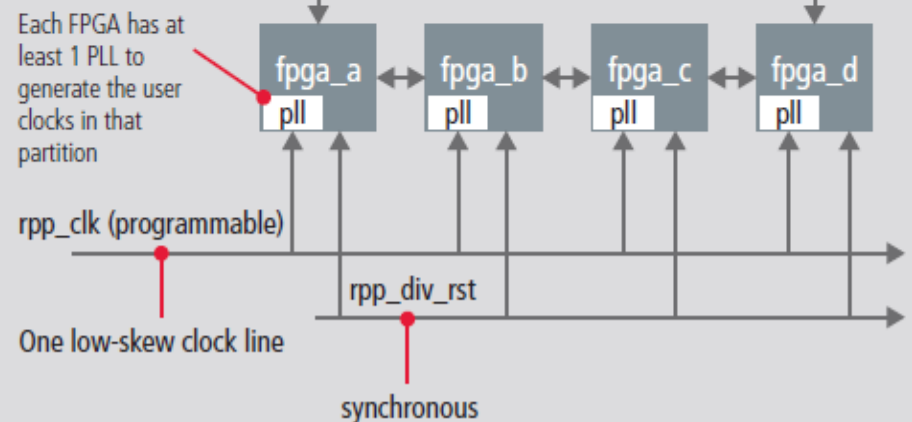
Advanced Clocking

- ## Complete clock tree transformation
  - Automatic conversion of gated and multiplexed clocks
  - Eliminates FPGA and board clocking restrictions
    - Support unlimited # of design clocks
  - Converts latches and tri-states
  - Removes FPGA hold-time violations
  - Reduces complexity of clock trees, which speeds up FPGA place and route
    - Faster P&R times – better quality of results

- ## Benefits:
  - No hold-time violations in user clock domains
  - Removes any FPGA-specific clock limitations
  - 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.
- ◆ The fastest design clock changes once per FCLK cycle, so it runs 2X slower than FCLK.

FCLK cycle # 0 1 2 3 4 5 6 7 8 9 10 11 etc.

FCLK (conceptual)

Fastest design clock
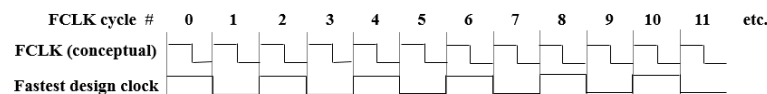
FCLK and Step Clock

- ◆ 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)

cādence ®
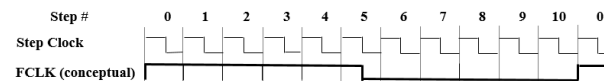
Adjustable Performance

Fully automatic: 3-10MHz*
- Clock-tree transformation
- ASIC memory mapping
- Partitioning
- FPGA P&R

Manual guidance: 10-20MHz*
- Partitioning input
- Directly connected bulk memories
- FPGA P&R options and constraints
- Logic replication
- Clock-tree simplification

Black-boxing: >100MHz*
- Single FPGA scope
- FPGA-specific optimization
- Direct clock mapping
- Directly connected bulk memories
- FPGA P&R options and constraints

* Actual performance is design-dependent

cādence®

Unique Control & Debug Capabilities

- ## Runtime
  - Start/stop clock capability (run "N" cycles)
- ## Memory (backdoor) upload and download
- ## Monitor signal
  - Real-time monitoring of predefined (at compile time) signals
- ## Force/release signal
  - Forces predefined signals (at compile time) into "0" or "1" during runtime

- ## Probes
  - Runtime data capture of predefined signals for offline waveform viewing
  - Waveforms across partitions
    - Provides a design-centric view rather than an FPGA-centric view

cādence®

Comprehensive Memory Support

- The conversion and implementation of memories is one of the most challenging and time-consuming steps in the bring-up of an FPGA-based prototype system, often taking many weeks to complete

Type	Size	Palladium MMP	Upload/download	Performance	Comments
FPGA-internal	~50Mbits/FPGA	Yes	Yes	Full design speed	• Fully automatic compile • Upload/download limited to 32 memory blocks max. per FPGA
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 • Currently supported: DDR3, DDR4, SRAM, SD-FLASH
FCMC (Full-custom memory card)	custom	No	No	Full design speed	• Full custom development
XSRAM (automated small external memory)	144 Mbytes per memory card	some	Yes	Full design speed	• Fully automatic compile • Extends 'FPGA-internal' memory modeling capabilities to a 144M external SRAM • Useful for SPI-flash and other small memories (e.g. boot ROM)
XDRAM (automated bulk memory)	8 GBytes per XDRAM card	DDR family models	Yes	>4.5MHz	• semi automatic compile • Leverages XDRAM hardware (daughter card) • Support for DDR3/4, LPDDR3/4 • Additional protocols may be added on customer demand and technical feasibility

The memory compile capabilities in the Protium™ platform are comprehensive and easy to use:

- Smaller memories are fully automatically compiled into FPGA-internal memory resources

- For larger, off-FPGA memories, Protium platform offers several solutions; which one to use depends on specific requirements and objectives

cādence®

Protium Rapid Prototyping Platform

- Gate capacity: up to 100M

- Adjustable performance: full automation to user tuned

- Fast compile time

- **Fast Time-toPrototype**

- Highly productive implementation flow
  – Automated memory compilation
  – Terminal timing controllability
  – Fully integrated FPGA place&route tool

- Automatic, emulation-like clock tree transformation
  – Unlimited # of user clocks
  – Gated-clock, latch, internal tri-state conversion
  – Elimination of any hold-time violations

cādence®

Example – Tensilica Fusion on Protium

cādence®

Tensilica Fusion

- **Single-precision FPU**
  - Floating Point instructions issued concurrently with 64-bit load/store
  - Speeds S/W porting
- **AVS (Audio/Voice/Speech)**
  - SW compatibility with HiFi 3 Audio DSP
  - Access to 125+ HiFi Audio/Voice software packages
- **Quad 16x16 MAC**
  - Accelerates communications standards like BLE/Wi-Fi
  - Accelerates voice algorithm performance
- **BLE/Wi-Fi AES-128**
  - Encryption acceleration for wireless
- **Baseband bit operations**
  - Accelerates performance of bit operations for implementation of Baseband MAC/PHY

FPU	Audio/Voice/Speech	AES-128
Xtensa Core ISA		
2-way VLIW FLIX		
Single/Dual MAC DSP (32b, 24b, 16b)		
Quad 16 MAC		Baseband Bit Ops

cādence®
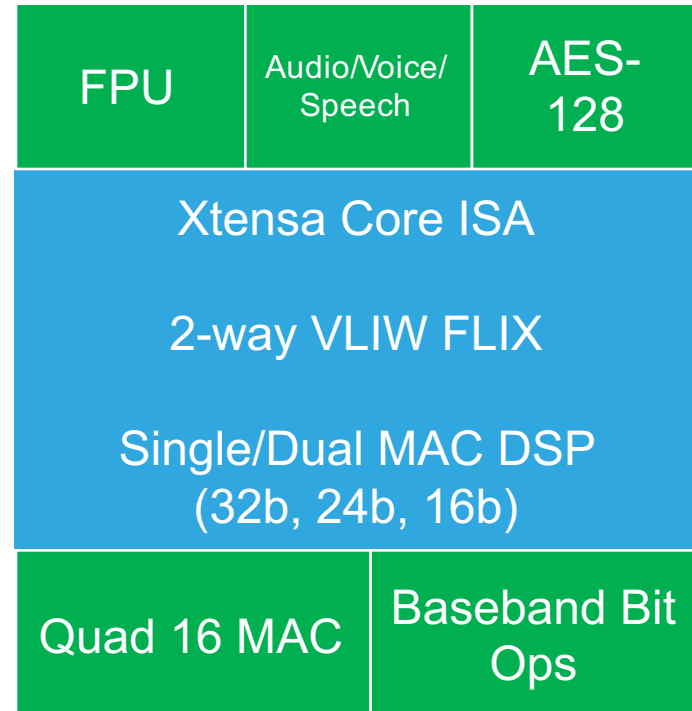
Debugging on Hardware via JTAG

May be the same workstation, or separate workstations

Full support for JTAG-based debugging and real-time trace

Xplorer	XOCD Daemon

TCP/IP
USB

JTAG
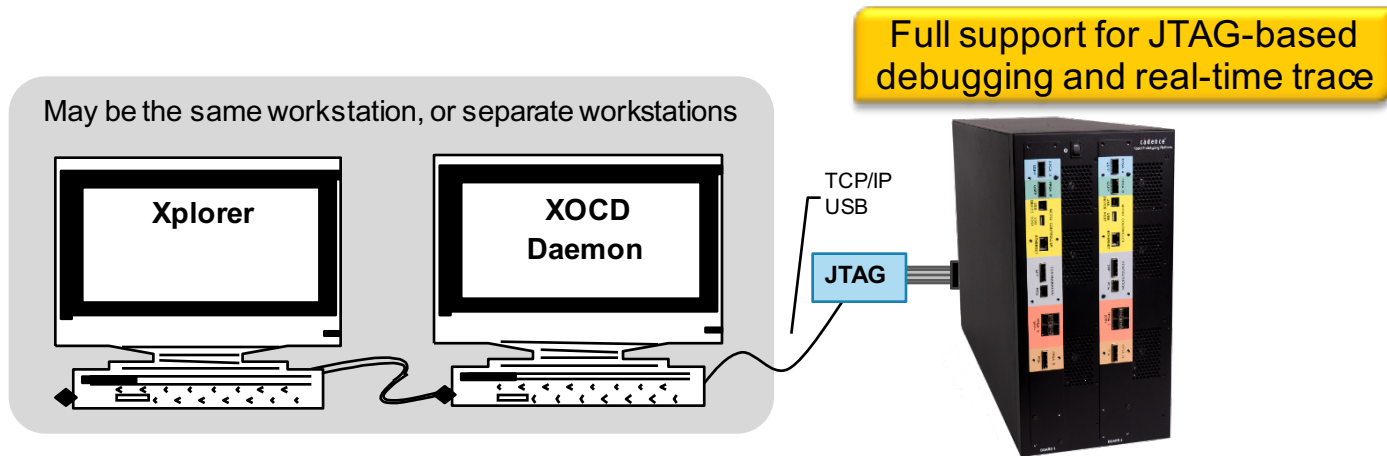
Debug Host	Target Host	Debug Probe	Target hardware
Windows or Linux host running : • Xtensa Xplorer with integrated xt-gdb • Command-line xt-gdb	Windows or Linux host running: • XOCD • debug probe driver software	• Flyswatter • Catapult • J-Link • DStream • Wiggler • USB2 Daemon • …and more	• Protium / Palladium • Emulation board • Customer target hardware with OCD

cādence®

Protium HW setup

cādence

Debugging

cadence®

Tack Tack

cādence®

cadence®