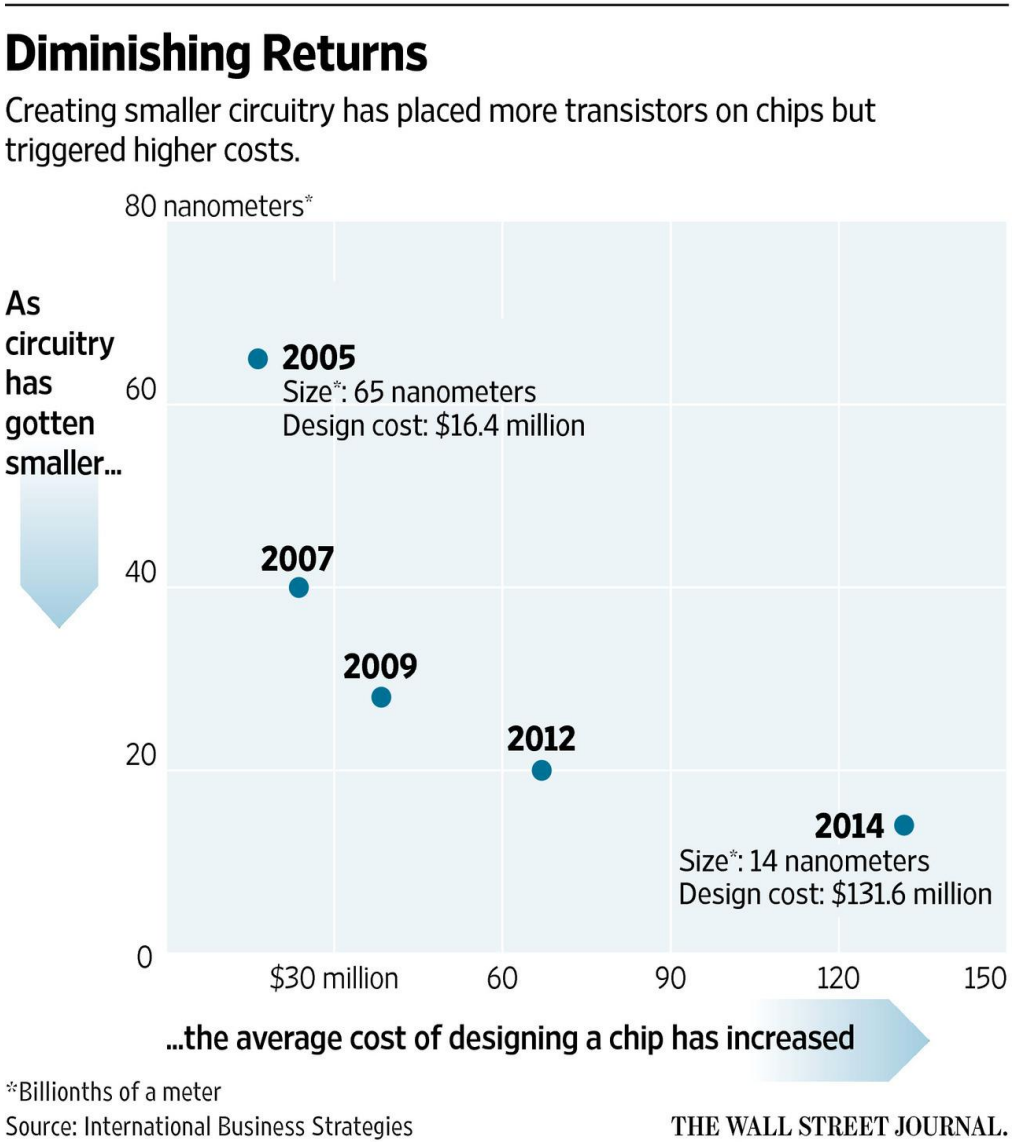
How Open Source Hardware Will Drive the Next Generation of HPC Systems

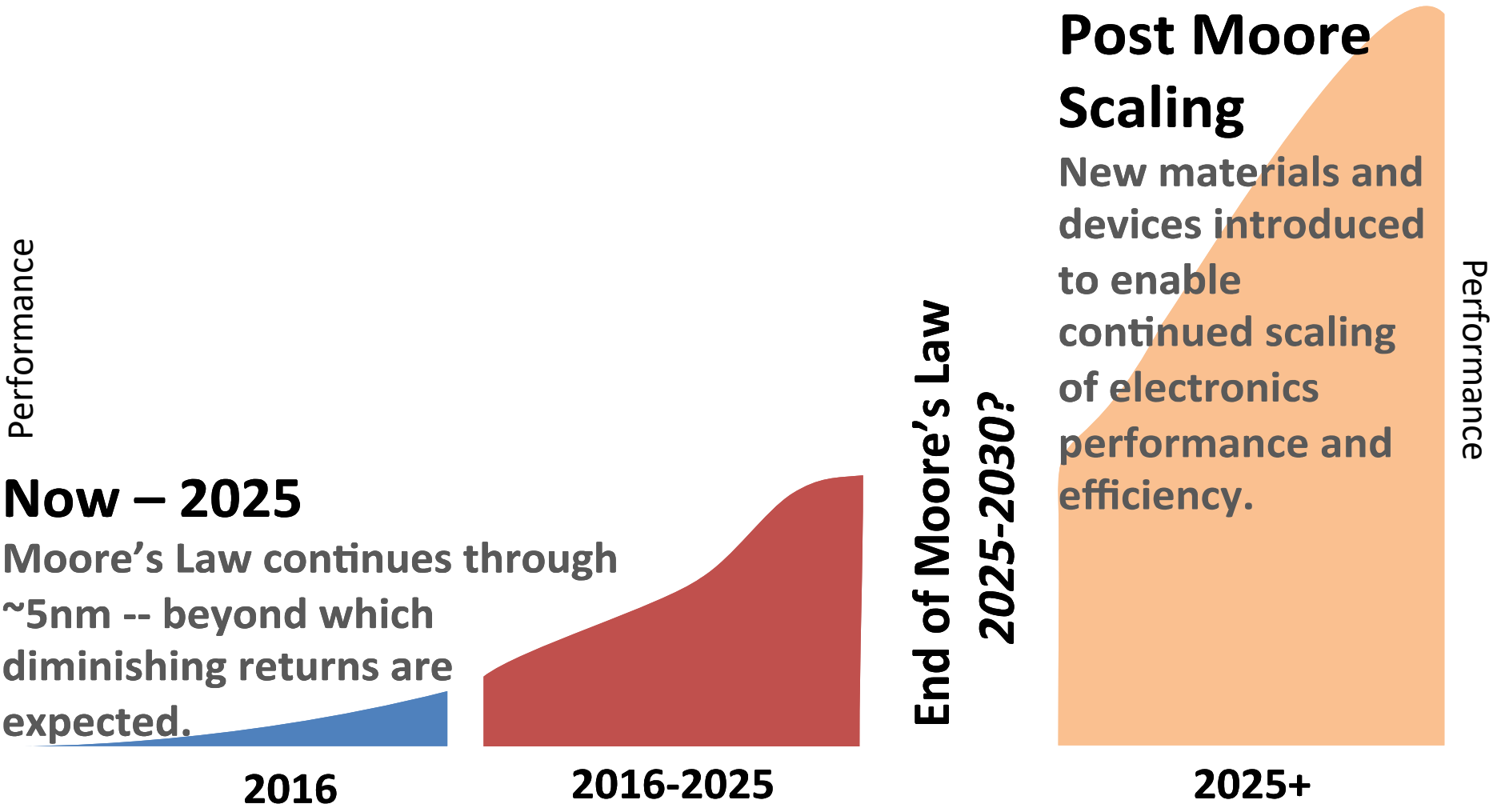
George Michelogiannakis

Research scientist
Lawrence Berkeley National Laboratory

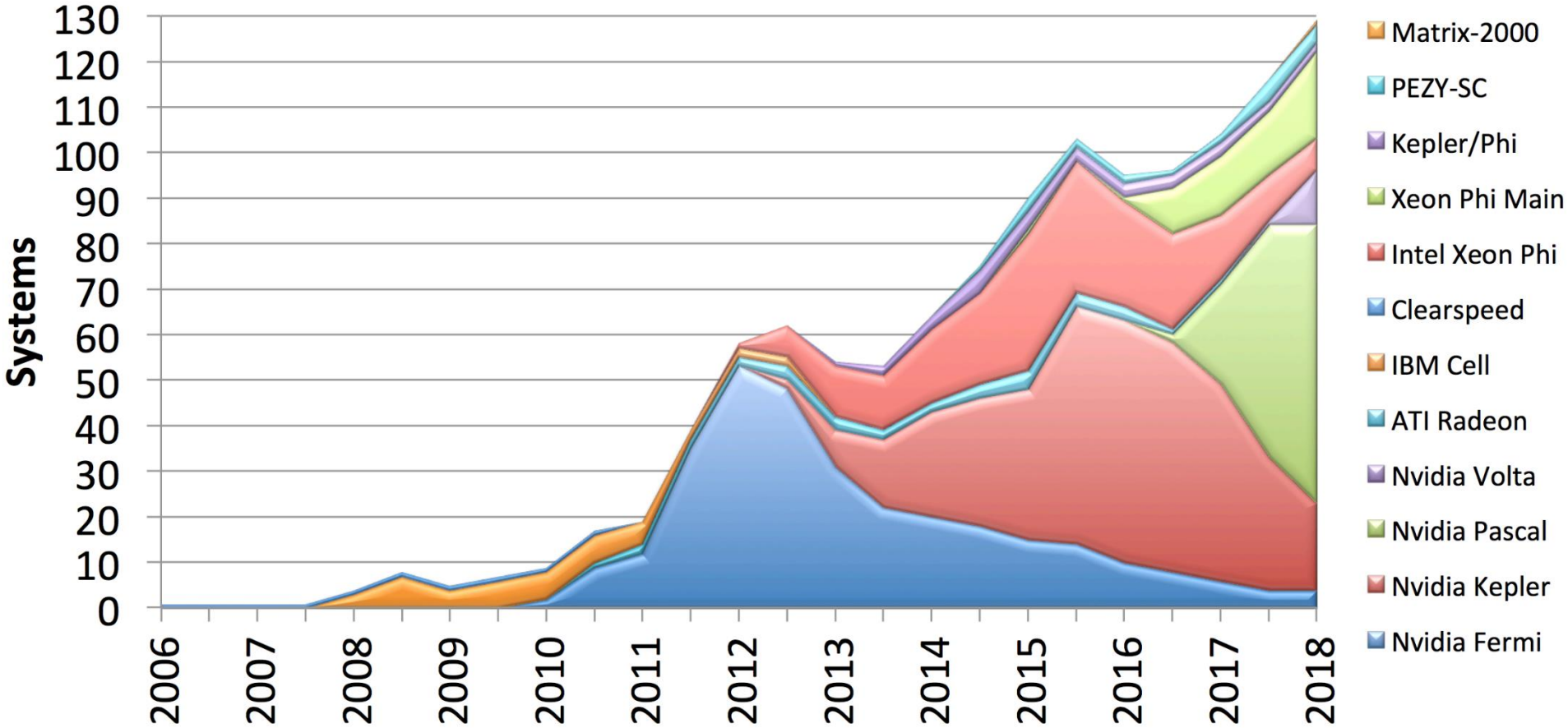
Moore's Law – A Quick Review



Preserve Performance Scaling With Emerging Technologies

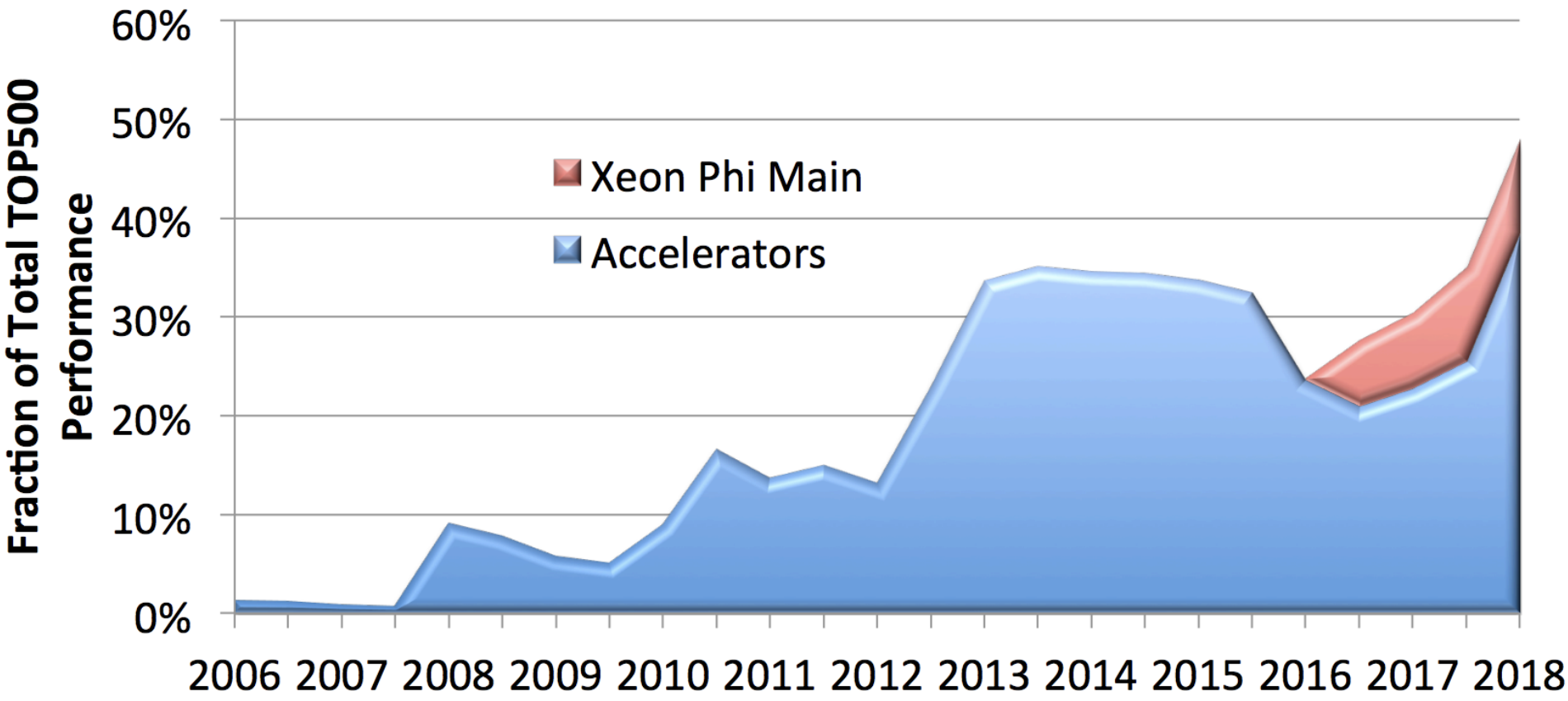


More Accelerators in HPC



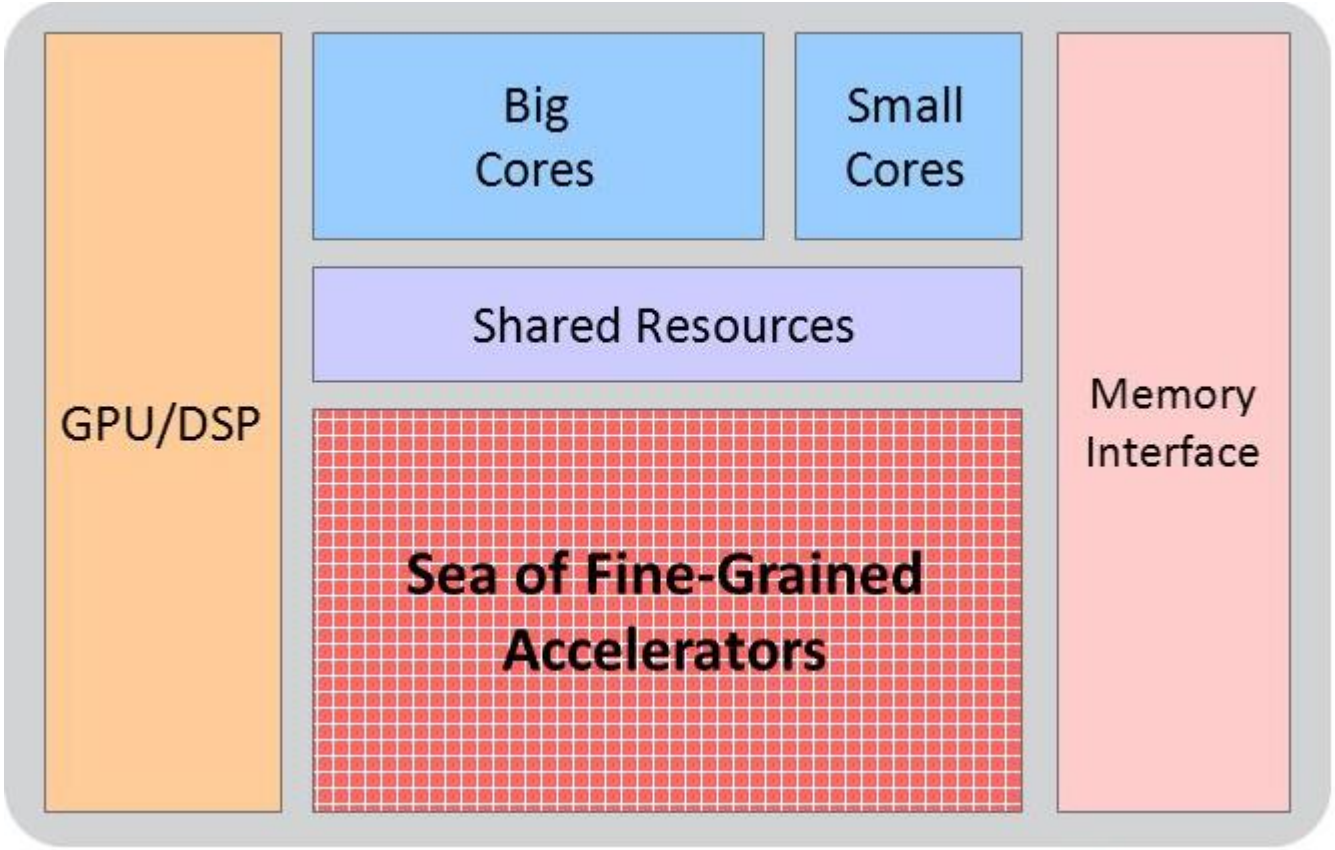
TOP 500

Performance Share



Fixed-Function Hardware

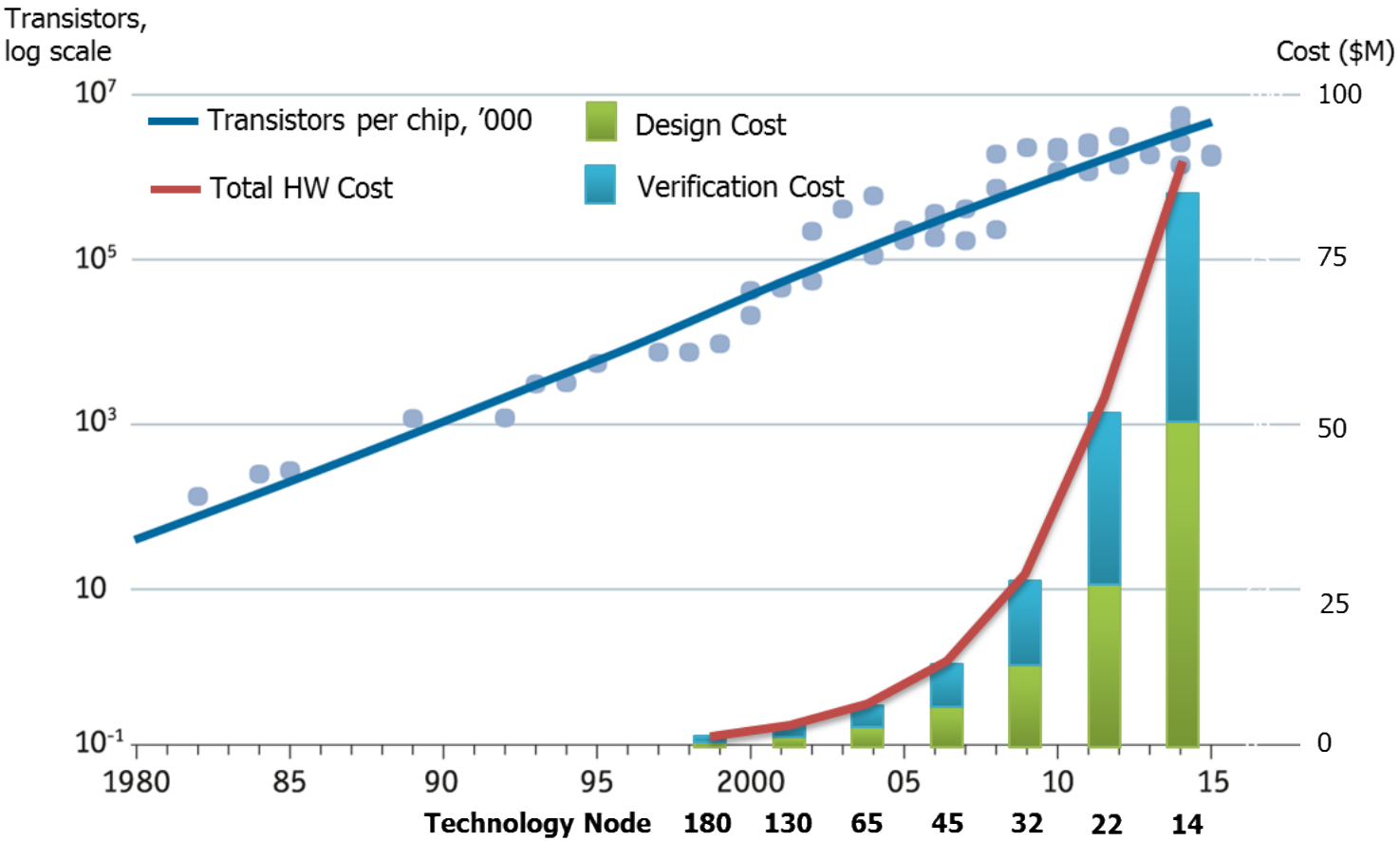
- How do we design accelerators for a wide variety of applications?



Yakun S et al “Aladdin”

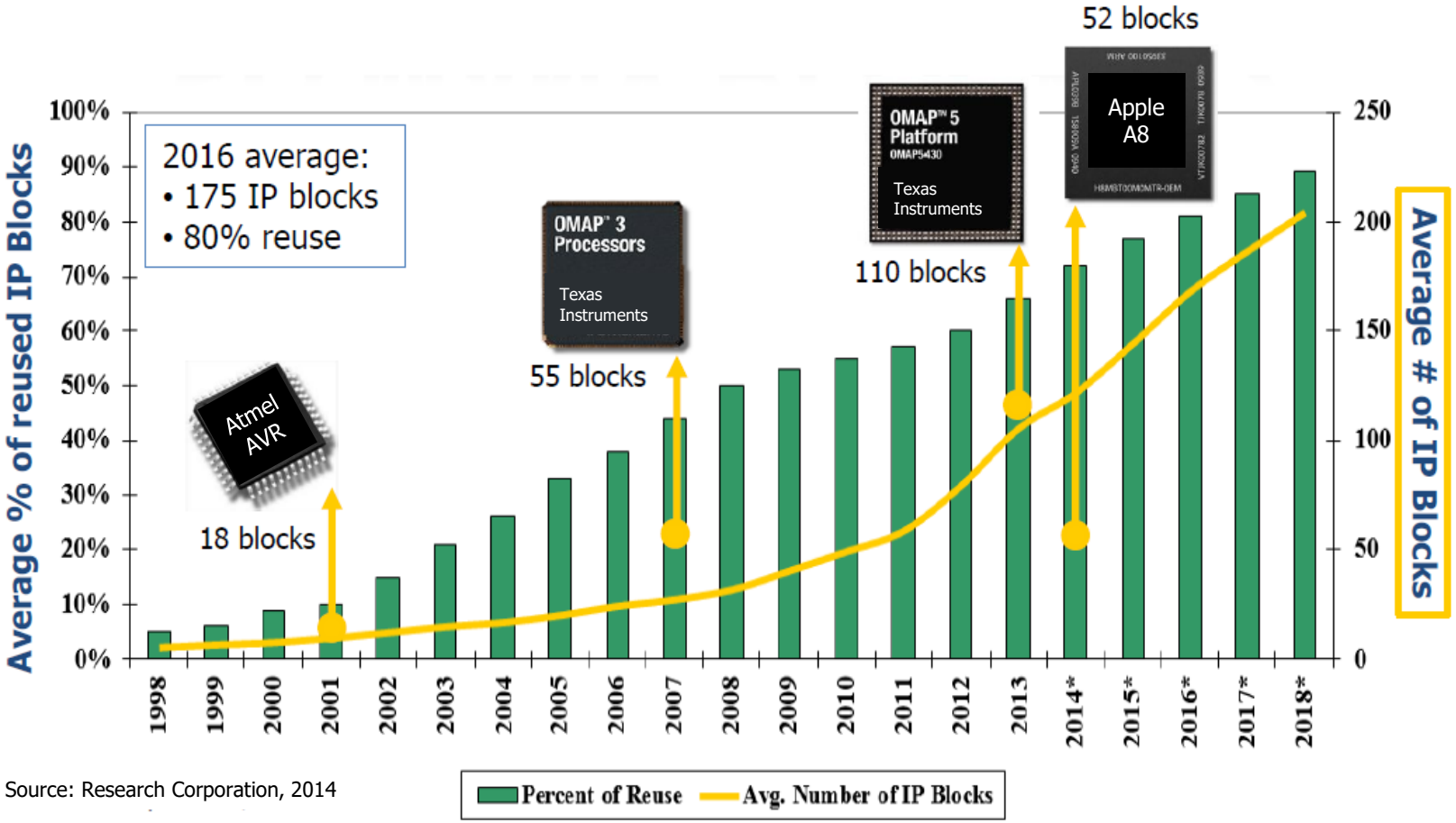
But This Will Further Increase Cost

The curse of Moore's Law



Because Complexity Already High

Root cause: complexity growth



Source: Research Corporation, 2014

Reduce Hardware Development Effort to Explore the Specialization Spectrum with:

Open-Source Hardware

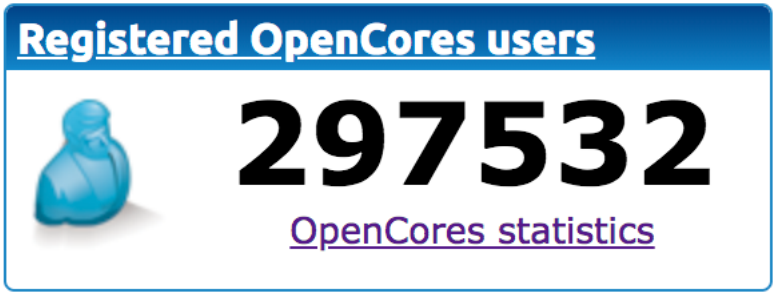
High-Level Synthesis Languages

Why Open Source Hardware?

- Closed-source IP major drag to innovation
  - High barrier to entry
  - Open nature enables customization
- Create a community
- Shorten design cycles
  - Share hardware and software stack
- Open-source hardware can form the basis of generators

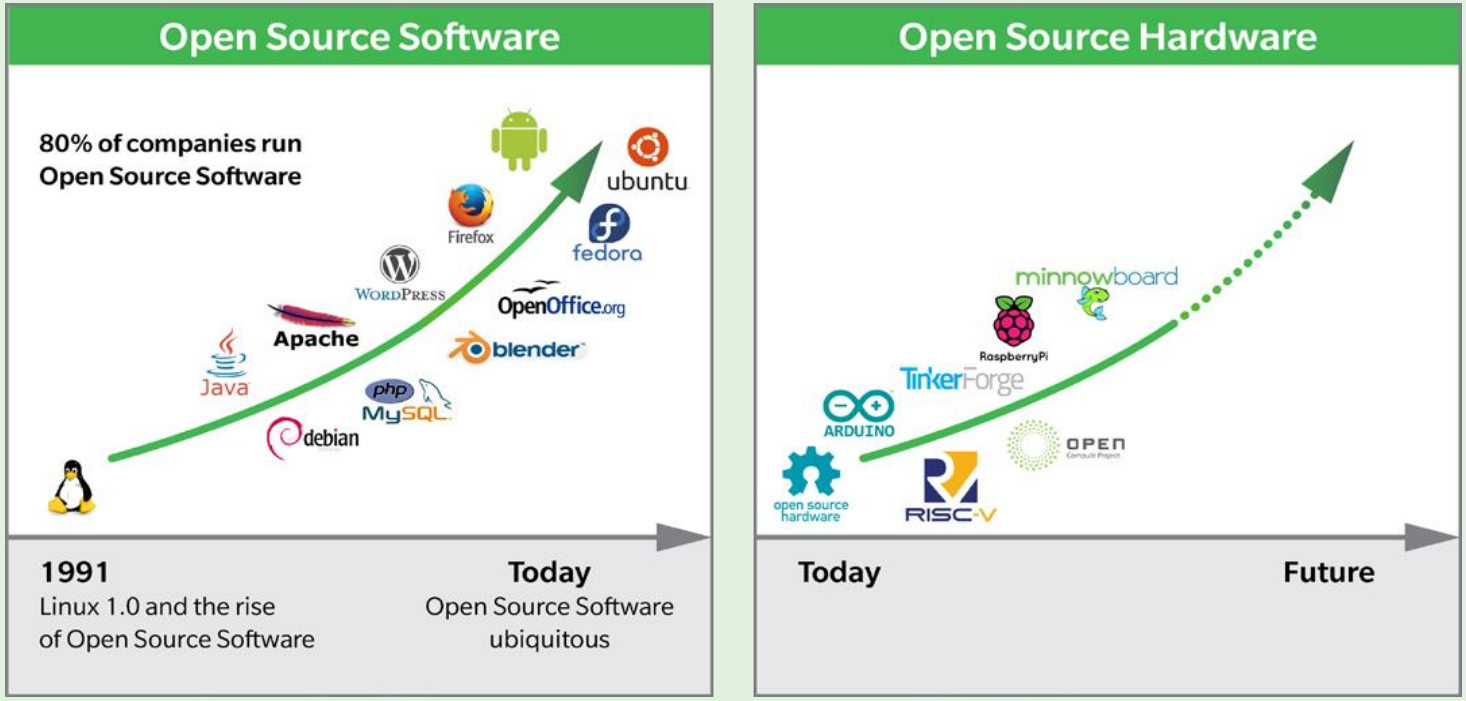
OpenCores

- Shows there is a large community interest
- Does not go far enough
  - Majority are point designs
- 1190 projects
  - 55 labeled “mature”



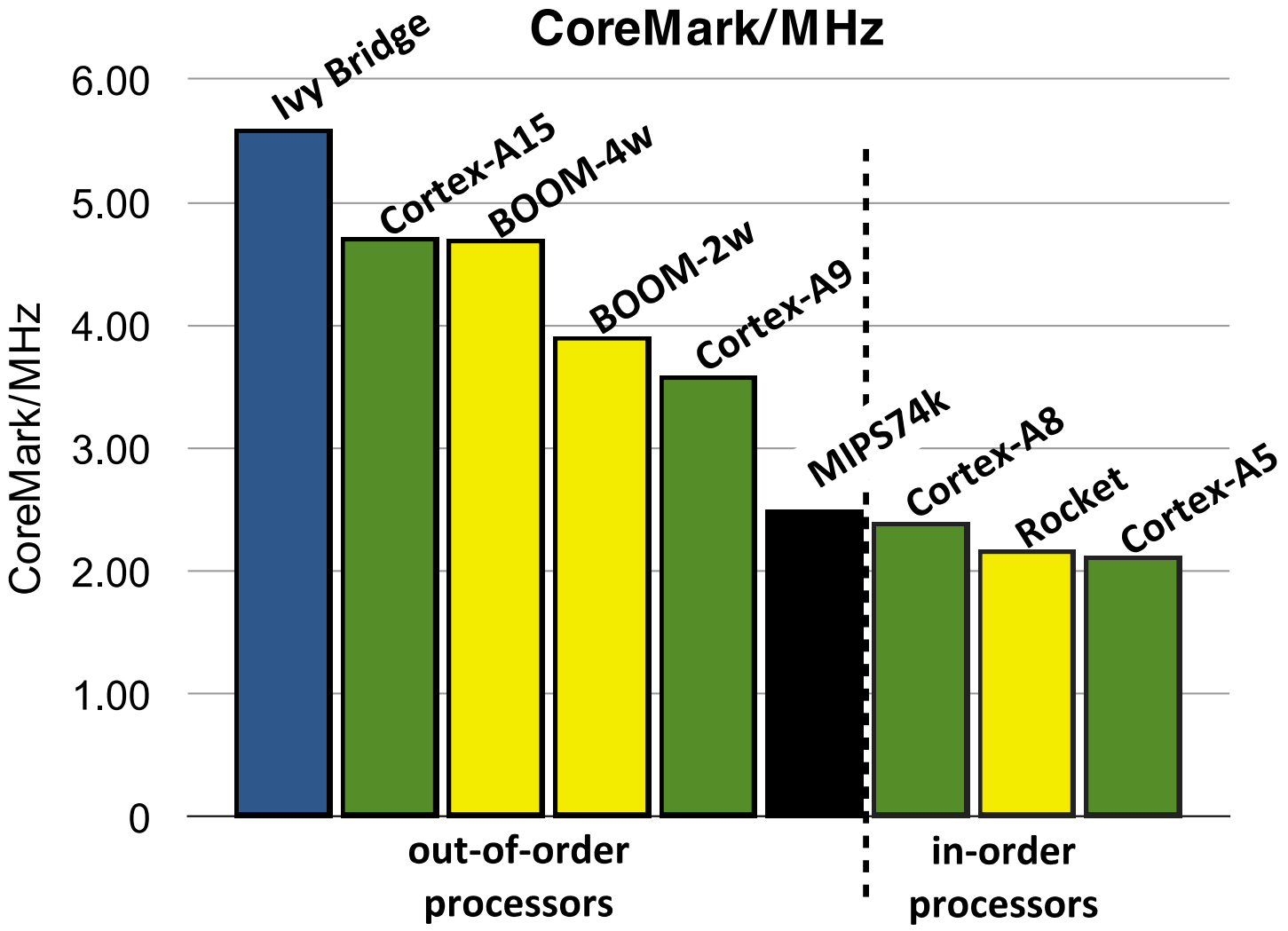
The Rise of Open-Source Hardware

The Rise of Open Source Software: Will Hardware Follow Suit?



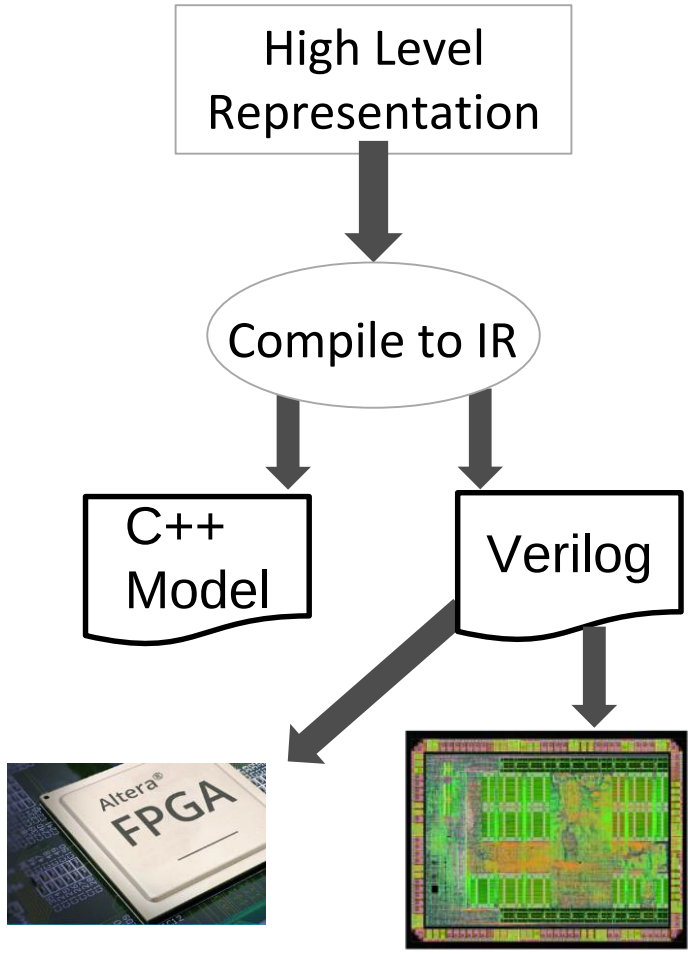
- Rapid growth in the adoption and number of open source software projects
- More than 95% of web servers run Linux variants, approximately 85% of smartphones run Android variants
- Will open source hardware ignite the semiconductor industry?

Encouraging Performance Results



More Productive H/W Design Path

- New DSLs raise abstraction level
  - Increase productivity and code re-use
- Hardware generators more efficient
  - Reduce cost, risk, design time



Code Re-Use

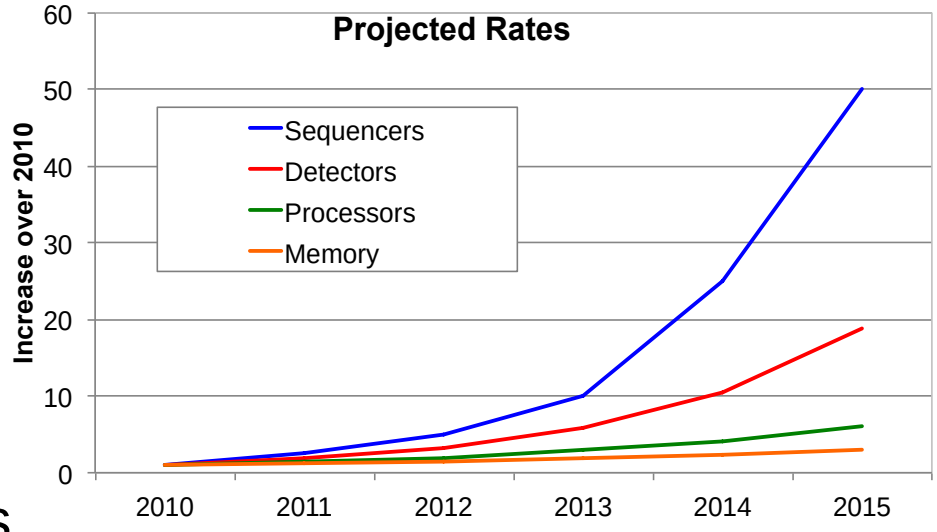
Reuse: Shared Lines of RTL Code (Chisel)

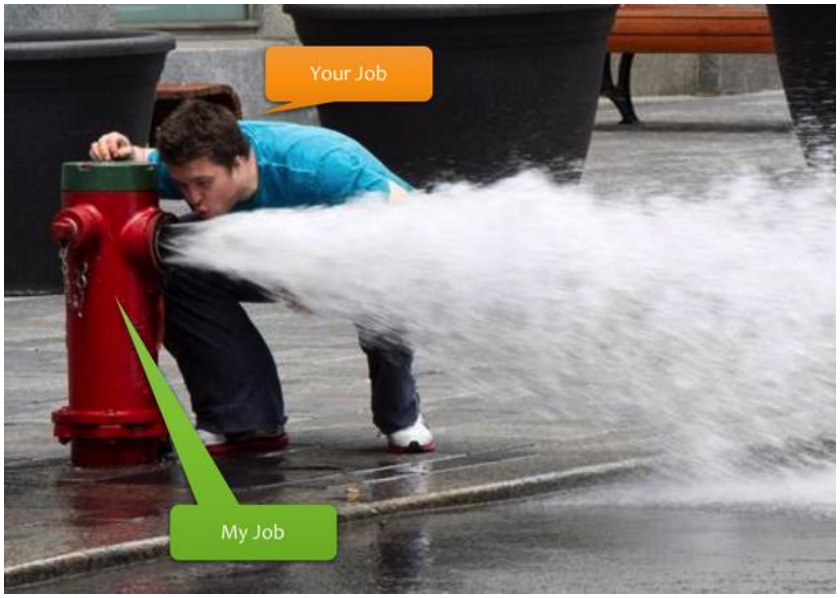
RISC-V Core	Z-scale	Rocket	BOOM
Description	32-bit 3-stage pipeline in-order 1-instruction issue L1 caches (≈ ARM Cortex-M0)	64-bit, FPU, MMU 5-stage pipeline in-order 1-instruction issue L1 & L2 caches (≈ ARM Cortex-A5)	64-bit, FPU, MMU 5-stage pipeline out-of-order 2-, 3-, or 4- instruction issue L1 &L2 caches (≈ ARM Cortex-A9)
Unique LOC	600 (40%)	1,400 (10%)	9,000 (45%)
LOC all 3 share	500 (30%)	500 (5%)	500 (5%)
LOC Z-scale & Rocket share	500 (30%)	500 (5%)	---
LOC Rocket & BOOM share	---	10,000 (80%)	10,000 (50%)
Total LOC	1,600	12,400	19,500

Use Open-Source Hardware: Specialization Opportunities

A Specialization Opportunity

- On-detector processing
- Future detectors have data rates exceeding **1 Tb/s**
- Proposed solution:
  - Process data before it leaves the sensor
  - Application-tailored, programmable processing
  - Programmability allows processing to be tailored to the experiment





Create an Architecture per Motif

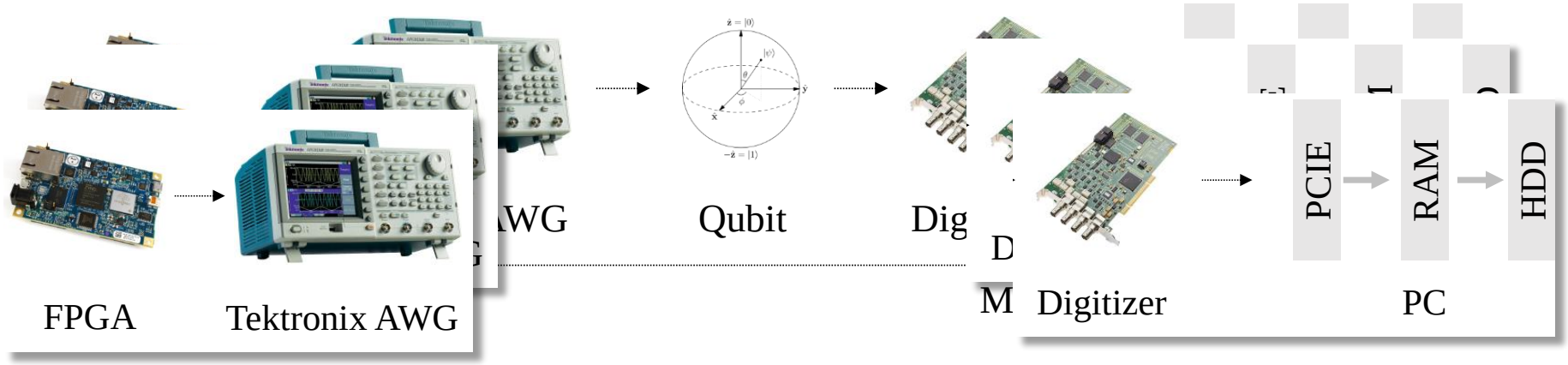
7 Giants of Data (NRC)	7 Motifs of Simulation
Basic statistics	Monte Carlo methods
Generalized N-Body	Particle methods
Graph-theory	Unstructured meshes
Linear algebra	Dense Linear Algebra
Optimizations	Sparse Linear Algebra
Integrations	Spectral methods
Alignment	Structured Meshes

Quantum Control Processor

- $Quantum\ Computer = Quantum\ PU + Control\ Hardware$

Off the shelf and high cost

Large amount of data and slow speed



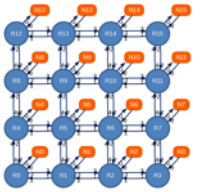
1000 qubits,
gate time 10ns,
3 ops/qubit
300 billion ops per second

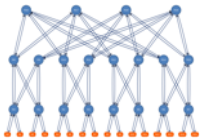
Some Current Projects

Accelerating the Design Process

- A complete set of tools

OpenSoC Fabric



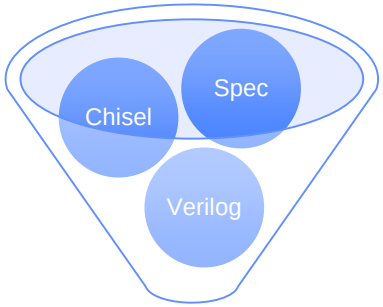


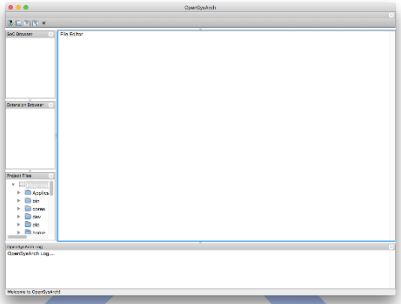
OpenSoC Compiler

OpenSoC Cores & Open2C

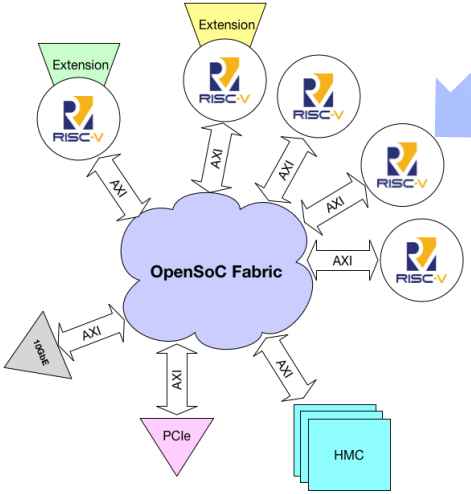
OpenSoC System Architect

OpenSoC System Architect





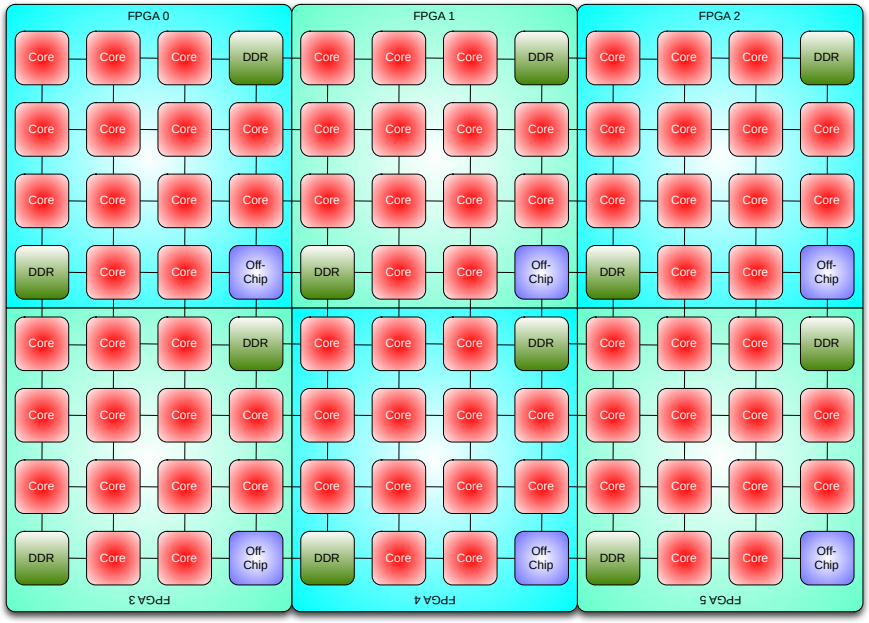
Frontend &
CoreGen



LLVM
Compiler

SC15 Demo: 96-core SoC for HPC

- Shockingly but accidentally similar to Sunway node architecture
- 4 Z-Scale processors connected on a 4x4 mesh and Micron HMC memory
- Two people spent two months to create



Questions