Open-Source Hardware in the Post Moore Era

Dr. George Michelogiannakis
Research scientist
Computer architecture group
Lawrence Berkeley National Laboratory

These are not DOE’s or LBNL’s official views
Technology Scaling Trends

Figure courtesy of Kunle Olukotun, Lance Hammond, Herb Sutter, and Burton Smith
Moore’s Law of Documentation

- IBM 250 nm
- GF 130 nm
- IBM 90 nm
- TSMC 65 nm

new “Moore’s Law” on documentation volume seen from the 14th floor at Fermilab perspective
Scaling Already Slowing Down

Peter Bright “Intel retires “tick-tock” development model, extending the life of each process “, 2016
Preserve Performance Scaling With Emerging Technologies

Now – 2025
Moore’s Law continues through ~5nm -- beyond which diminishing returns are expected.

Post Moore Scaling
New materials and devices introduced to enable continued scaling of electronics performance and efficiency.

2016
2016-2025
End of Moore’s Law
2025-2030?
2025+

Performance
Performance
Some Paths Forward in Post Moore

New Devices

- Carbon Nanotubes
- Spintronics
- TFETs
- Photonics
- MRAM
- General Purpose
- Exascale
- CMOS

New Models of Computation

- Modeling / Simulation
- Asynchronous (Async)
- Neuromorphic
- Big Data
- Dataflow
- Superconducting
- Specialized Architectures
- SoC

Specialized Hardware

- 3D Stacking
- Advanced Packages
- Dark Silicon
- First Line of Defense: More Efficient Architectures

10 Years +10 Year Lead Time

Revolutionary Heterogeneous HPC Architectures

Y

X
3D Integration of Tomorrow

Enabled by Emerging Nanotechnologies

Massive Sensing

Data Storage (NV memory)

Computing Logic

Fine-grained 3D integration (not TSVs)

Shulaker “Transforming Emerging Technologies into Working Systems”
Let's Get The Most Out of CMOS
Before we Jump Ship
General-Purpose Architectures Trade Overhead for Programmability

Superscalar out-of-order pipeline

Architectures Trade Overhead for Programmability

Compare against 12-core 1.9 GHz Intel Xeon E5-2420 processor

Accelerators Have Been Growing in HPC

Top 500 systems

Strohmaier “Top 500”, SC17
 Hardware that is more suited for specific kinds of computation
 Can also have accelerators for data transfer

General purpose

Accelerators

Fixed function

Programmability

High

Low
GPU Acceleration is Popular

How GPU Acceleration Works

Application Code

Compute-Intensive Functions
5% of Code

Rest of Sequential CPU Code

GPU

CPU

NVIDIA
FPGA Acceleration Just Beginning

- FPGA accelerators used as programmable array of soft cores – more like a GPU model

- Parallels early days of GPGPU computing
  - Capable hardware
  - New languages raising abstraction levels
  - Tools lacking

Wadler, Intel’s Response to ARM Servers: Xeon D Processors with a Twist, 2016
Fixed-Function Hardware

* How fine-grain accelerators?
* How to schedule and transfer data?

Yakun S et al “Aladdin”
Hardware Development Effort Is a Challenge

- **Behavioral simulators**: Fast but less accurate
  - Typically used to prune design space
  - No substitute for real hardware

- Lets make hardware development faster!
  - High level synthesis languages
  - Open-source hardware

12-18 month cycle

Zipcpu, “FPGAs vs ASICs”, 2017
Reduce Hardware Development Effort to Explore the Specialization Spectrum With Open-Source Hardware
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?
Accelerating the Design Process

*A complete set of tools*

- OpenSoC Fabric
- OpenSoC Compiler
- OpenSoC Cores & Open2C
- OpenSoC System Architect
OpenSoC System Architect

- Chisel
- Spec
- Verilog

Frontend & CoreGen

LLVM Compiler
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
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

![Projected Rates Graph](image-url)
<table>
<thead>
<tr>
<th>7 Giants of Data (NRC)</th>
<th>7 Motifs of Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic statistics</td>
<td>Monte Carlo methods</td>
</tr>
<tr>
<td>Generalized N-Body</td>
<td>Particle methods</td>
</tr>
<tr>
<td>Graph-theory</td>
<td>Unstructured meshes</td>
</tr>
<tr>
<td>Linear algebra</td>
<td>Dense Linear Algebra</td>
</tr>
<tr>
<td>Optimizations</td>
<td>Sparse Linear Algebra</td>
</tr>
<tr>
<td>Integrations</td>
<td>Spectral methods</td>
</tr>
<tr>
<td>Alignment</td>
<td>Structured Meshes</td>
</tr>
</tbody>
</table>
Spatial Specialization

* Architecture to match data set shape to help communication

**PDEcell / PICcell:** Ultra-simple compute engine (50k gates) calculates finite-difference updates, and particle forces from neighbors. Microinstructions specify the PDE equation, stencil, and PIC operators. **Novel features:** variable length streaming integer arithmetic and novel PIC particle virtualization scheme.

**Computational Lattice:** PDECells are tiles in a lattice/array on each 2D planar chip layer. Target 120x120 tiles per mm² @28nm lithography. **Novel Features:** each tile represents single cell of computational domain (pushes to limit of strong-scaling).

**Monolithic 3D Integration:** Integrate layers of compute elements using emerging monolithic 3D chip stacking.

- **Novel Features:** 1000 layer stacking (20x more than current practice).
- Area efficient inter-layer connectivity and new energy efficient transistor logic (ncFET).
- 1 Petaflop equivalent performance in 300mm^2 for < 200Watts.
* 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
Conclusion

- Open-source projects rely on community
  - Need a collection of accelerators

- Open-source hardware may be the key to ubiquitous specialization

- Programmability and compilers must not be neglected

- It is an exciting time to be an architect
Questions