OpenSoC Fabric

An open source, parameterized, network generation tool

David Donofrio, Farzad Fatollahi-Fard
George Michelogiannakis, John Shalf

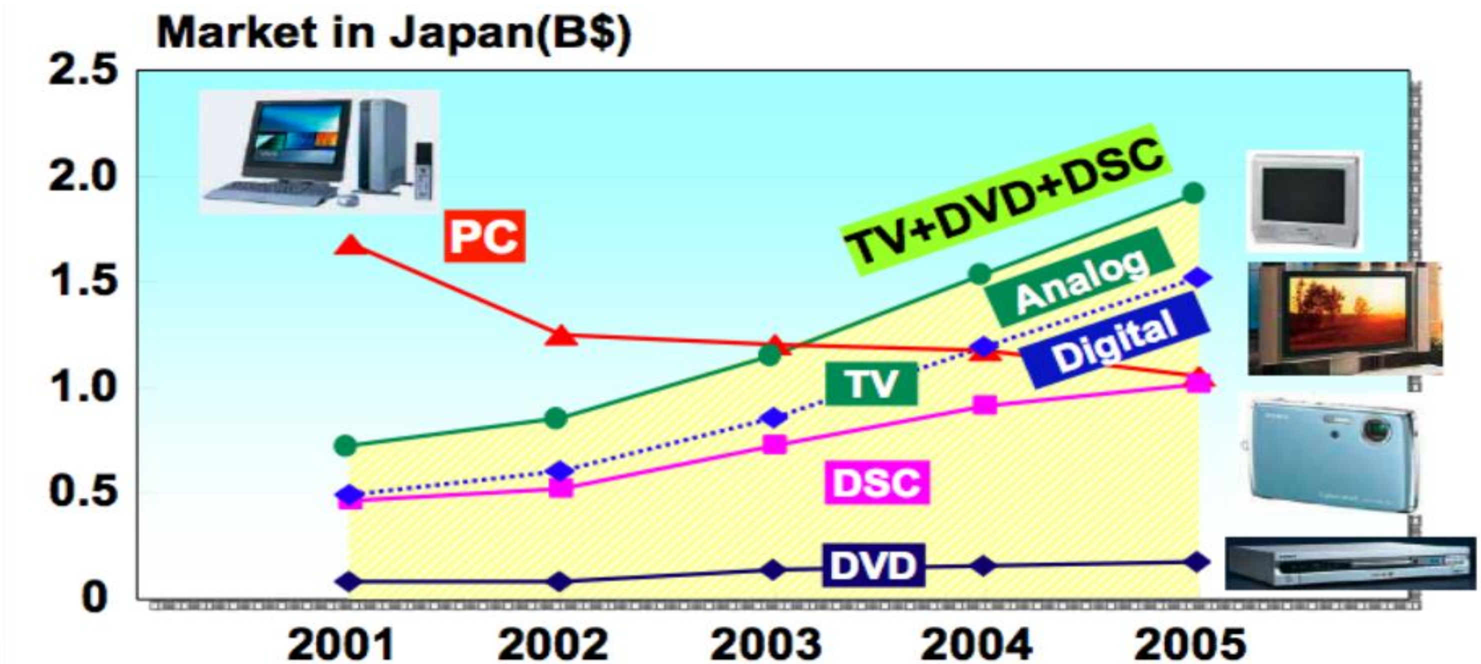
CoDEx
COMPUTER ARCHITECTURE LABORATORY
EXASCALE DESIGN SPACE EXPLORATION

OpenFabrics Alliance – Monterey, UCA
March 17, 2015

Technology Investment Trends

Image below From Tsugio Makimoto: ISC2006

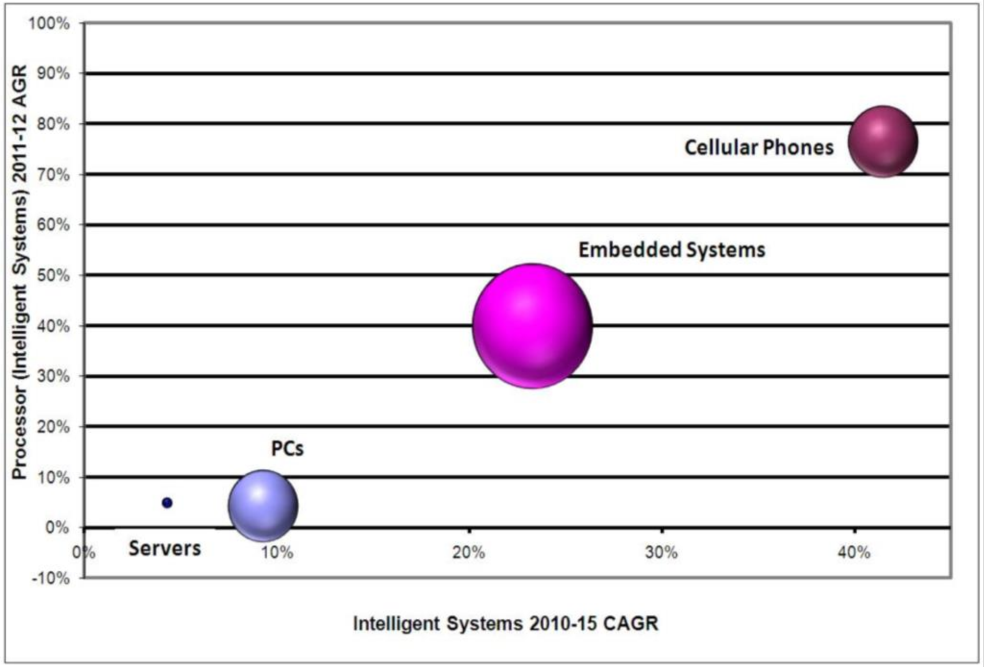
- **1990s – R&D in computing dominated by desktop market**
- **2000's – R&D investments moving rapidly towards consumer electronics and embedded**



Trends continue today…

IDC 2010 Market Study

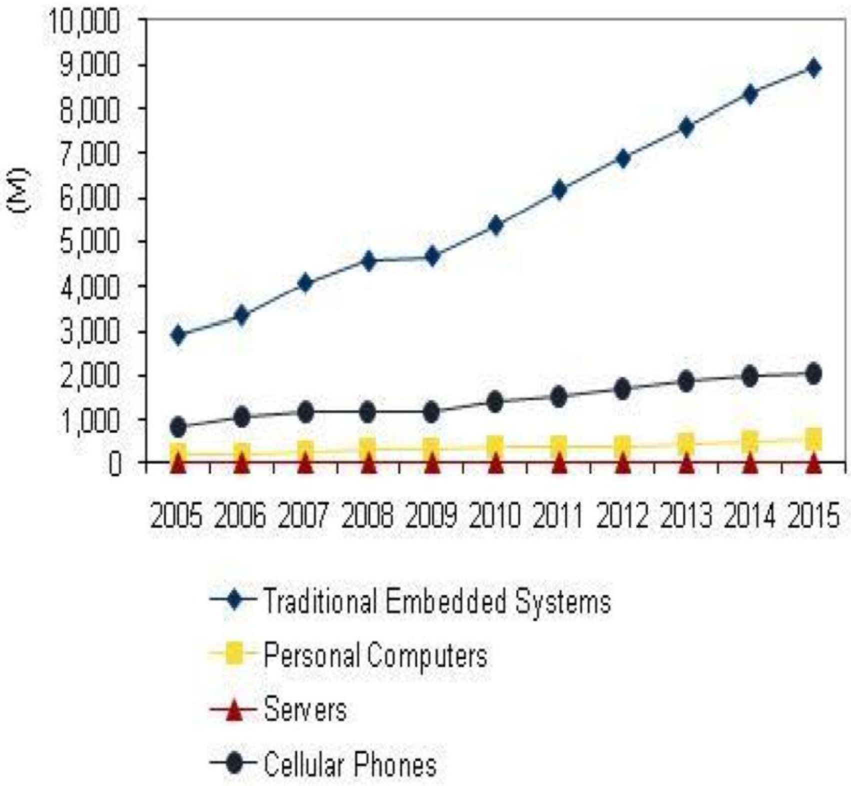
Worldwide Intelligent Systems Unit Shipments Comparison - Embedded Systems vs. Mainstream Systems 2011 Share and Growth



Notes:

Size of bubble equals 2011 share of system shipments. Growth of cell phone system shipments is driven by smartphones and multi core processor designs.

Worldwide Systems Unit Shipments - Traditional Embedded Systems vs. Mainstream Systems, 2005-2015 (Millions)



Building an SoC for HPC

Is this a good idea?

- **Consumer market dominates PC and server market**
  - Smartphone and tablets are in control
  - Huge investments in IP, design practices, etc.
- **HPC is power limited (delivered performance/watt)**
  - Embedded has always been driven by max performance/watt (max battery life) and minimizing cost
- **HPC and embedded requirements are now aligned**
  - …and now we have a very large commodity ecosystem
- ***Why not leverage technologies for the embedded and consumer for HPC?***

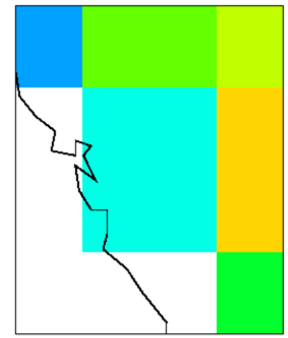
Looking back…

Some previous HPC system designs based on semi-custom SoCs

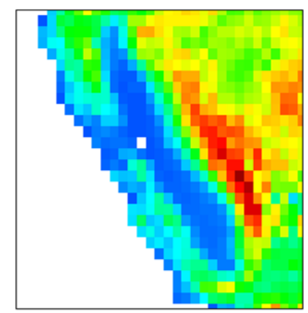
Applying Embedded to HPC (climate)

Must maintain 1000x faster than real time for practical climate simulation

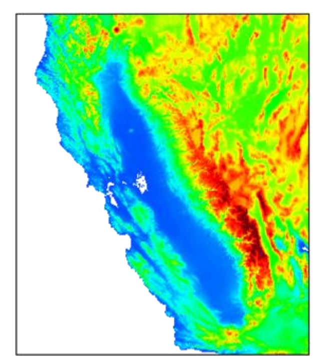
- **~2 million horizontal subdomains**
- **100 Terabytes of Memory**
  - 5MB memory per subdomain
- **~20 million total subdomains**
  - Nearest-neighbor communication
- ***New discretization for climate model***
  - *CSU Icosahedral Code*



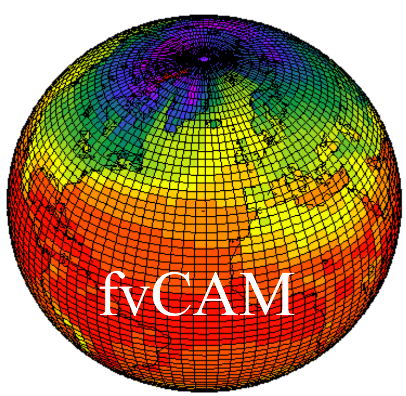
200km
Typical resolution of IPCC AR4 models

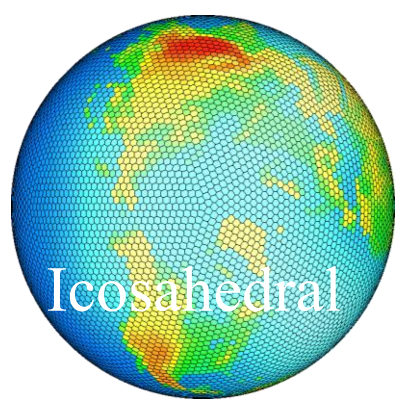


25km
Upper limit of climate models with cloud param



~2km
Cloud system resolving models
transformational





Green Flash

A full system design

System Arch	45nm	22nm
Cores per Chip	128	512
Clock Freq	650 MHz	650 MHz
Gflops / core	1.3	1.3
Cache / core	256 KB	256 KB
Gflops / chip	166	666
Subdomains / chip	4 x 4 x 8	8 x 8 x 8
Total Cores	20,971,520	20,971,520
Total Chip count	163,840	40,960

- **167,772,162 vertices at ~2 km**
- **Rectangular, 2-D decomposition**
  - 2,621,440 horizontal domains
  - 20,971,520 vertical domains
- **28 PF Sustained**
- **1.8 PB Memory**

Berkeley Sumpercomputer Predicts Your Doom

Written by whatsrequired

0 tweet | Share 0 | Digg

Photo:

Image from Blatch

The University of California at Berkeley is rolling out a new breed of supercomputer, specially designed to predict the challenges presented by climate change, ultimately leading humanity to our doom and the computers to their rightful place as the masters of our earthly domain.

The idea driving the claim that supercomputers can be revolutionized is the radical notion

Search!

with permission by photographer Lane Hartwell

conteurs Performing Carolina Dram

About

Spidey Senses is written by Ted Rheingold, a passionate thirty-something living in San Francisco. He's started and runs both the biggest dog info, care and community site and cat info and community site (aka Dogster and Catster =) and posts articles about online communities and business development at the Dogster, Inc. company blog.

Recent Mini-Updates

- Now that's a perfect fit! Big congrats @dshen http://ds.ly/IKNVnR Help more great things grow. about 2 hours ago
- Sleeping lamb, smiling monkey http://plixi.com/p/96517546 about 3 hours ago
- @sarahkunst are u on instagram on path? Been posting more there. Mabel was conceived just before or after (oops) the 4 of us had drinks ;) about 3 hours ago
- @Aloisius happy birthday! And invite me when u do! 1 day ago

Firefox Extension

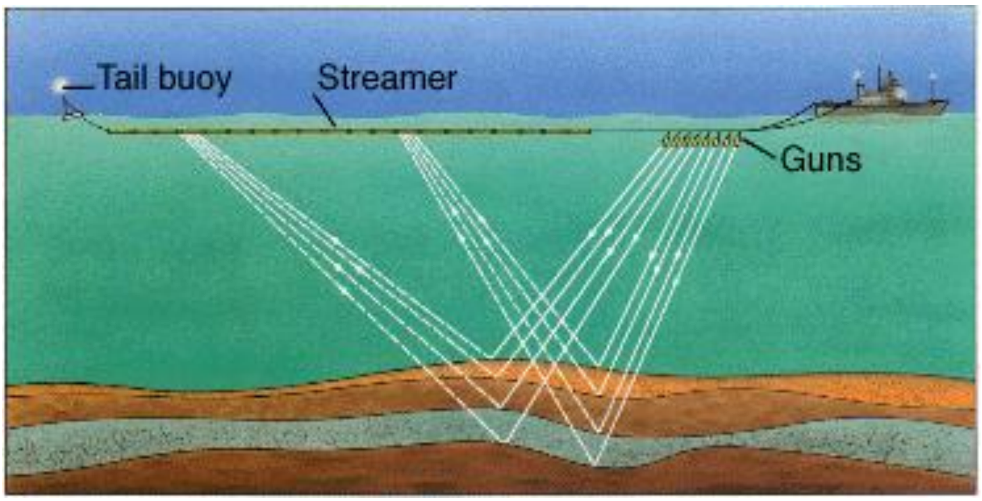
- Who Is This Person? Research a person by searching their name against relevant websites.

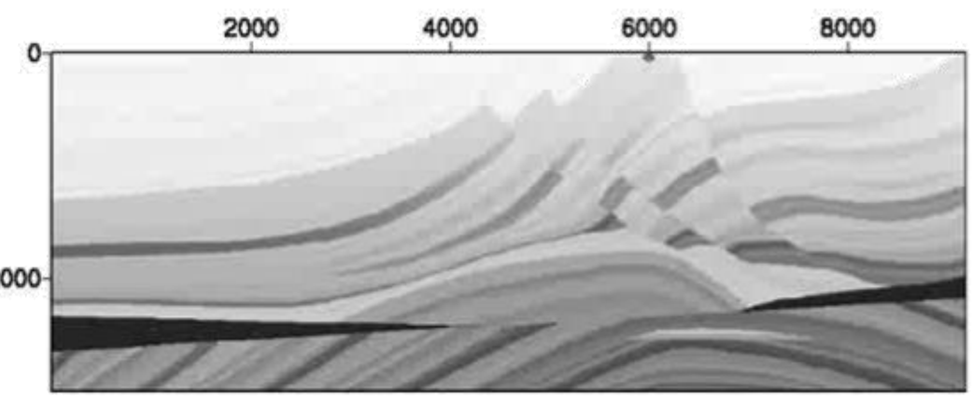
Recent Site Readers

You!

Green Wave

Apply principles of Green Flash to a new problem – 2009-2012





- **Seismic imaging used extensively by oil and gas industry**
  - Dominant method is RTM (Reverse Time Migration)
- **RTM models acoustic wave propagation through rock strata using explicit PDE solve for elastic equation in 3D**
  - High order (8th or more) stencils
  - High computational intensity

BERKELEY LAB

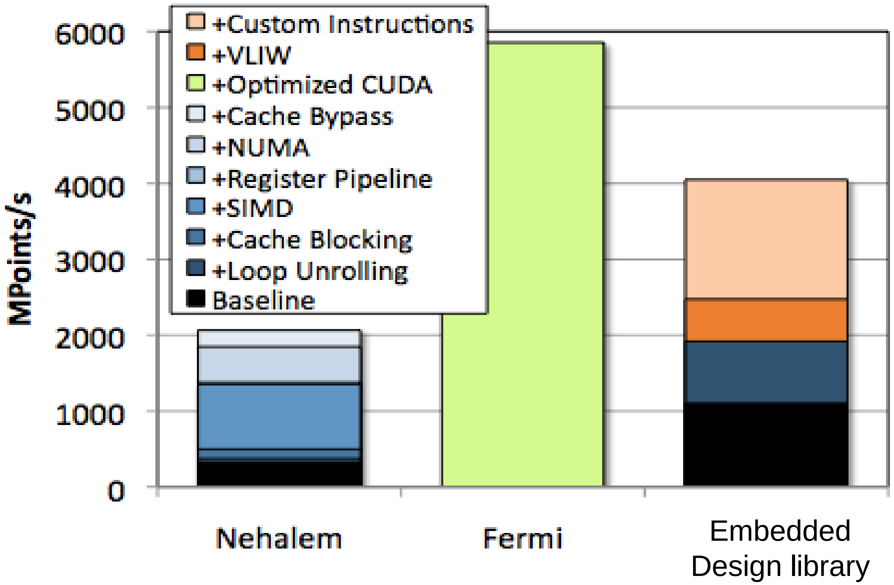
Fraunhofer ITWM

U.S. DEPARTMENT OF ENERGY | Office of Science

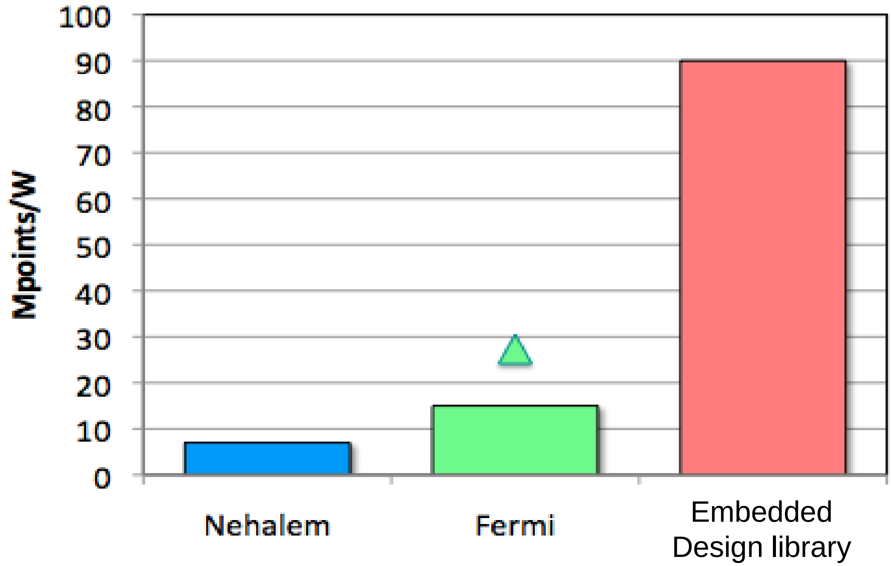
Green Wave Design Study
Seismic Imaging

Green Wave Inc. 2010

Performance

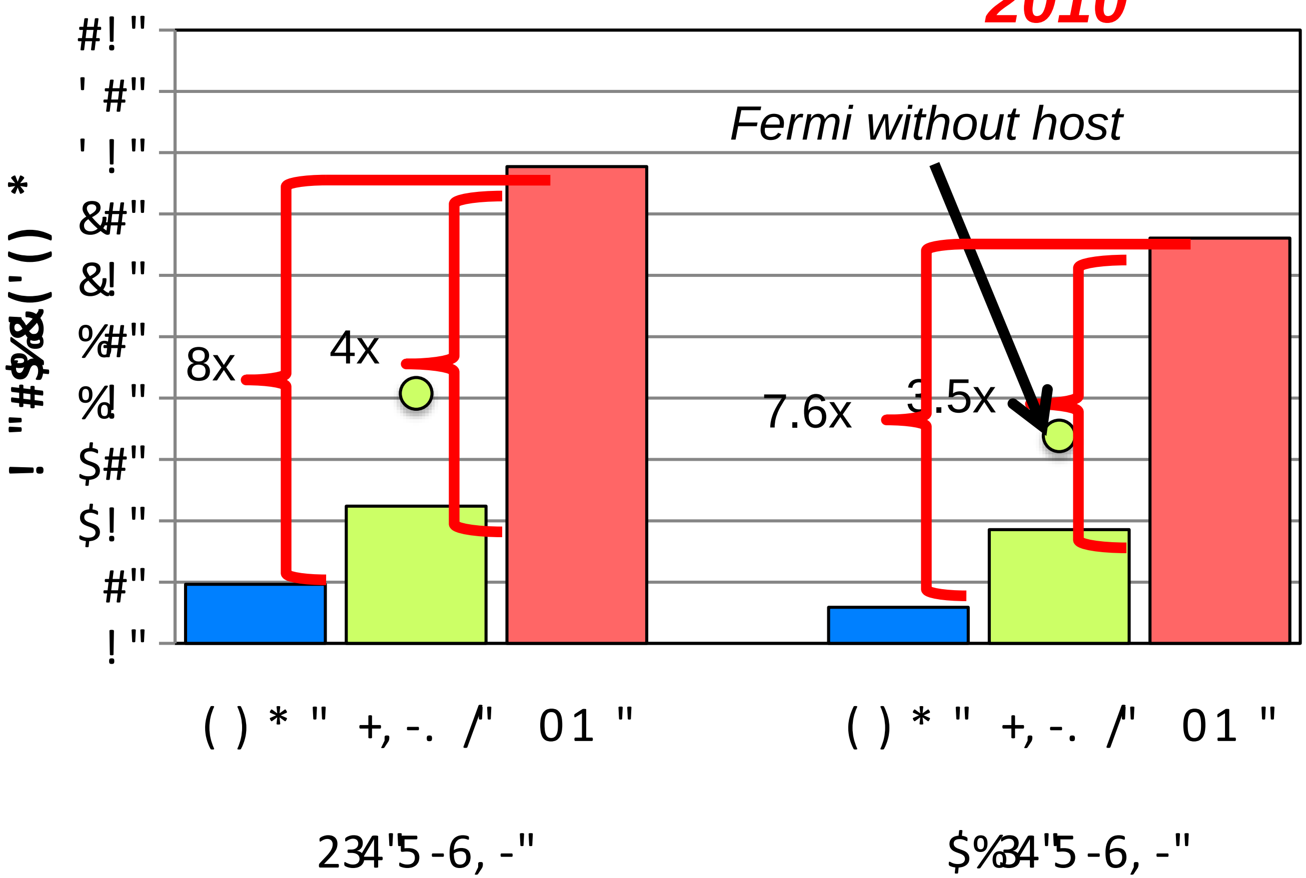


Energy Efficiency



Embedded SoC Efficiency Competitive with cutting-edge designs

Green Wave Inc. 2010



So what does this cost?

Total cost: **$20 Mil** using the assumptions below:

(Courtesy Marty Deneroff, Green Wave, Inc.)

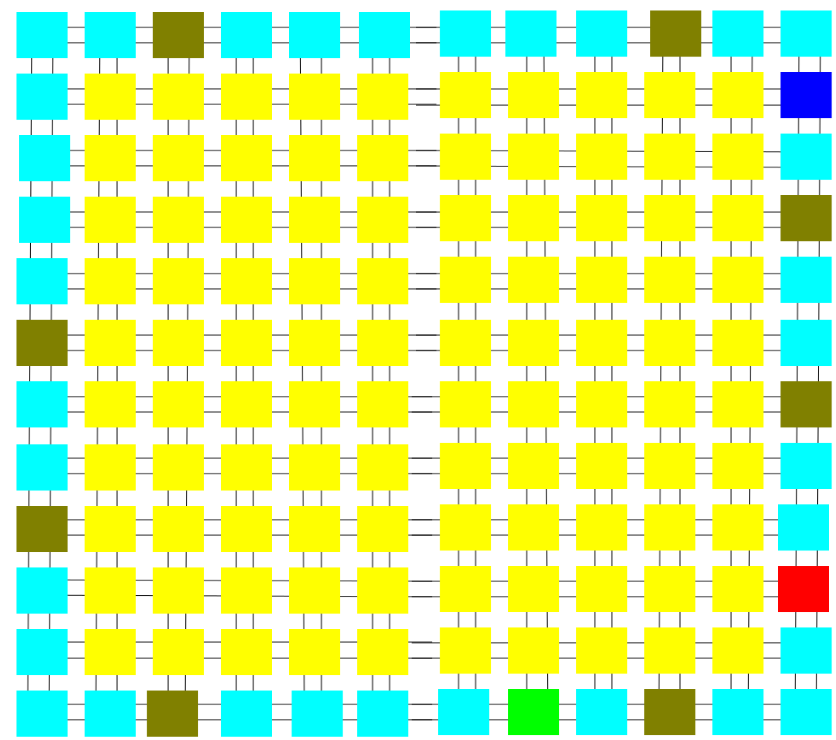
- **Current established (Not Bleeding Edge!) process**
- **Large (near reticle limit) die size**
- **Vendors understand what you are doing, trust your competence**
- **$5M NRE to Silicon Integrator (eSilicon, GUC, etc.)**
  - Physical design
  - Package design
  - Test design
  - Mask & proto charges

- **$5M for IP**
- **$2M for CAD tools**
- **$8M for engineering salaries and expenses**
  - 20% architecture / logic design
  - 20% system software development
  - 30% Design Verification
  - 30% Floorplanning / placement / vendor engagement

Green Wave Chip Block Diagram

Courtesy Marty Deneroff, Green Wave, Inc.

- **12 x 12 2D on-chip torus network**
- **676 Compute cores (500 in compute clusters, 176 in peripheral clusters)**
- **33 Supervisory cores**
- **1 PCI express interface**
- **8 Hybrid Memory Cube (HMC) interfaces**
- **1 Flash controller**
- **1 1000BaseT Ethernet controller**
- **It is not anticipated that all cores will be utilized – some are spares for yield enhancement.**



Actual network connections form folded torus, not open mesh
Torus connection not shown.

- Compute cluster (5 FLIX cores + DMA)
- HMC Cluster (4 FLIX Cores + DMA + HMC)
- Supervisory Cluster (4 FLIX cores + DMA + 1 TLB Core)
- Enet Cluster (4 FLIX Cores + DMA + Enet)
- PCIexpress Cluster (4 FLIX Cores + DMA + PCIe)
- Flash Cluster (4 FLIX Cores + DMA + Flash

Inspiration from the Embedded Market

- **Have most of the IP and experience with for low-power technology**
  - Have sophisticated tools for rapid turn-around of designs
- **Vibrant commodity market in IP components**
  - *Change your notion of “commodity”!*
  - *It’s commodity IP on the chip (not the chip itself!)*
- **Design validation / verification dominate cost**
- **Convergence with HPC requirements**
  - Need better computational efficiency and lower power with greater parallelism

Integration is Key

What if we had method of quickly integrating the IP that is readily available for the embedded market?

Embracing Integration

What happens when you stop caring about core power

- **Future chips will have many lightweight cores for computation**
  - Power per core will drop to mW – does *not* imply energy efficiency
  - Similar to embedded cores…
- **Integrated IP will differentiate processors**
  - Also efficiency gains in what we do *not* include
- **Need powerful networks to connect cores to memory(s), external IO and each other**

Building an SoC from IP Logic Blocks

It's Legos with a some extra integration and verification cost

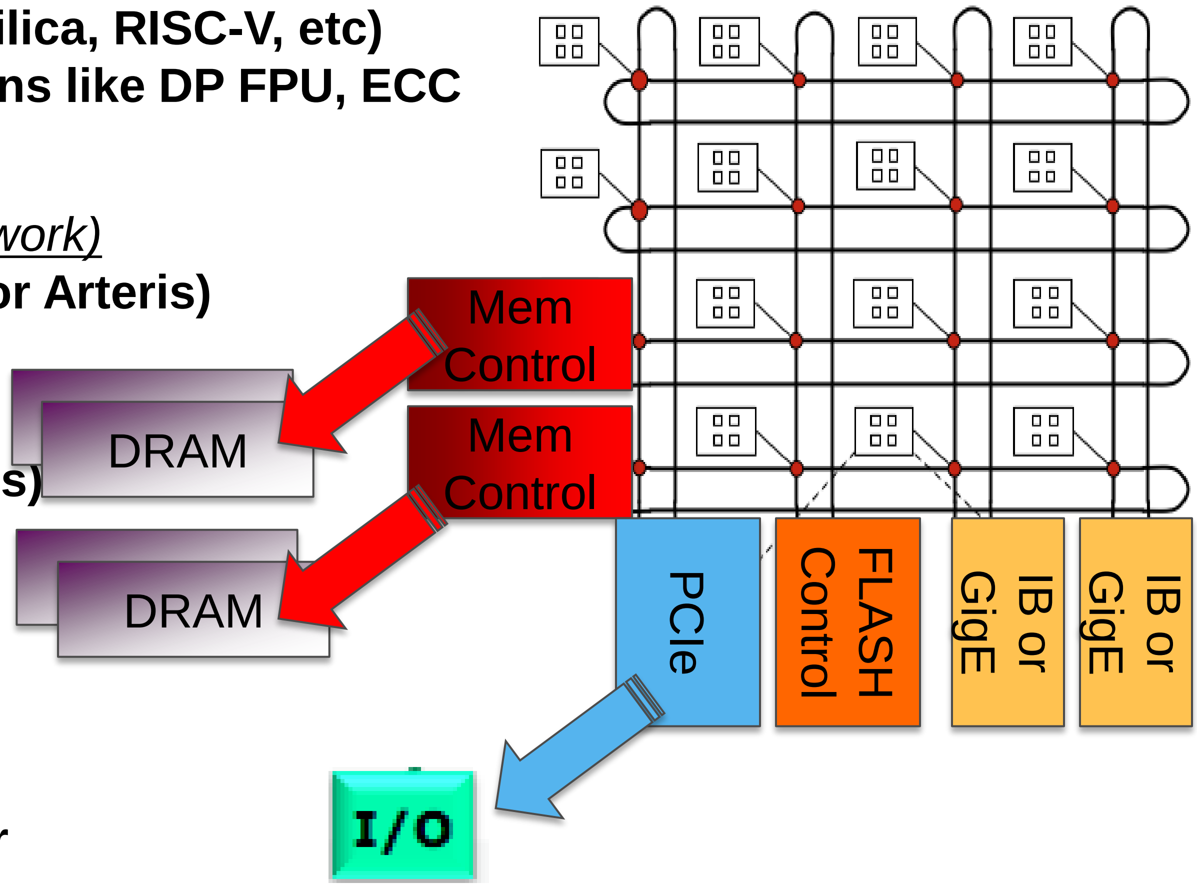
Processor Core (ARM, Tensilica, RISC-V, etc)
With possible HPC extensions like DP FPU, ECC

<u>OpenSoC Fabric (on-chip network)</u>
(currently proprietary ARM or Arteris)

DDR memory controller
(Denali/Cadence, SiCreations)
+ Phy & Programmable PLL

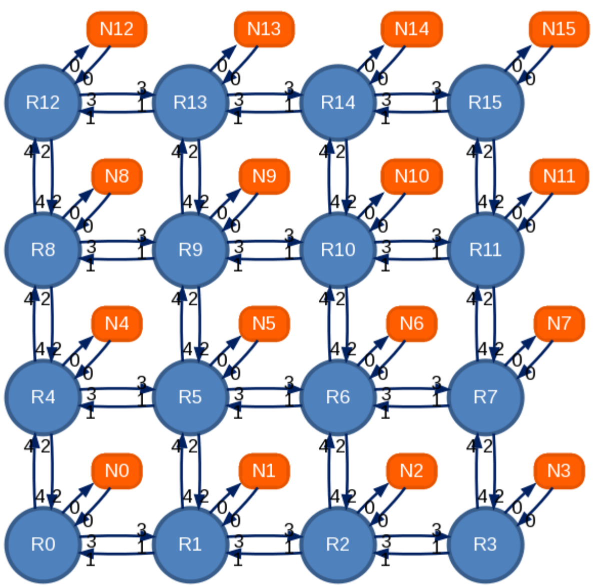
PCIe Gen3 Root complex

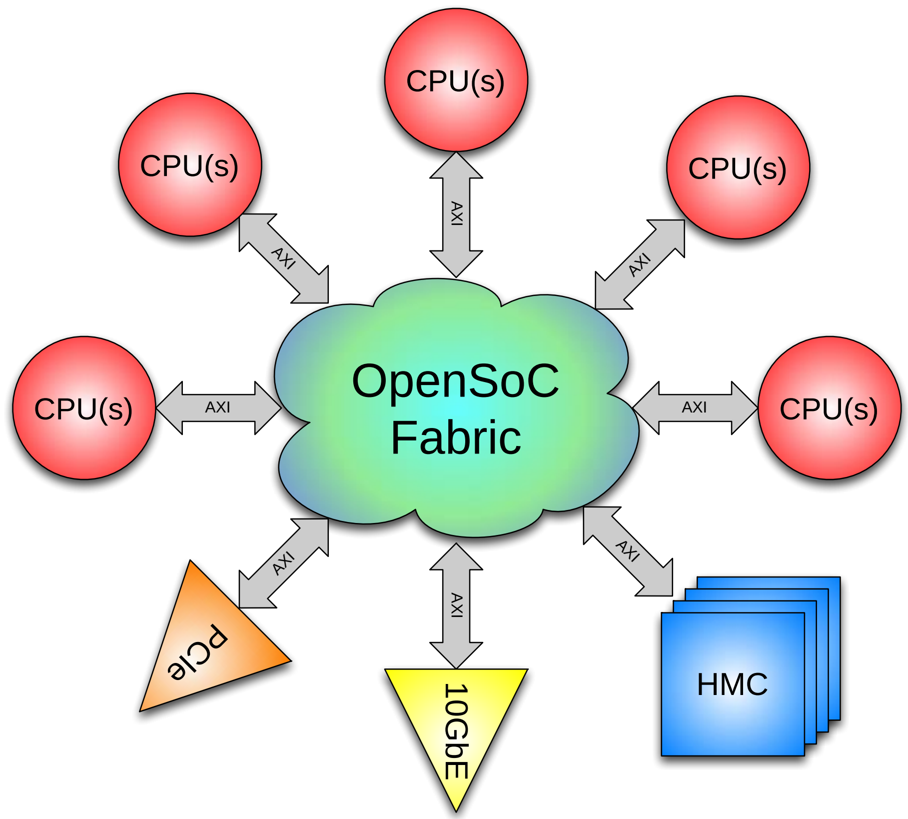
Integrated FLASH Controller



10GigE or IB DDR 4x Channel

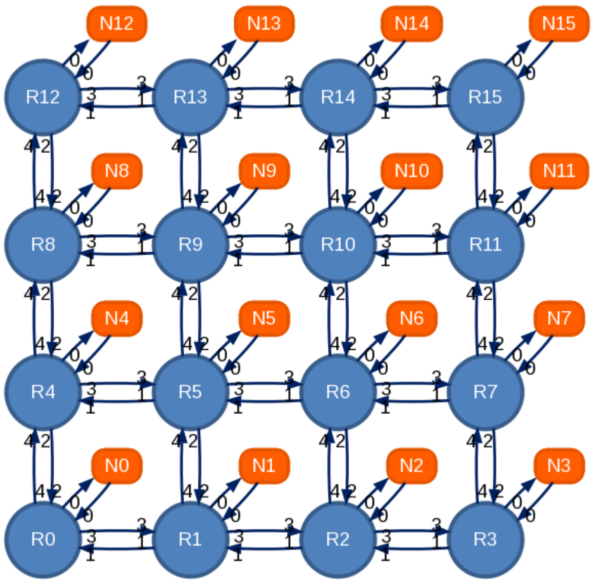
Network on Chip Overview

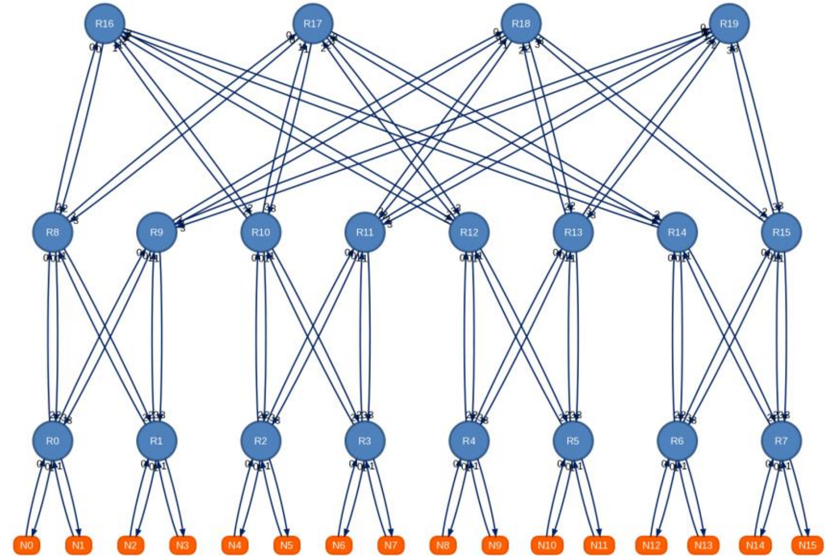


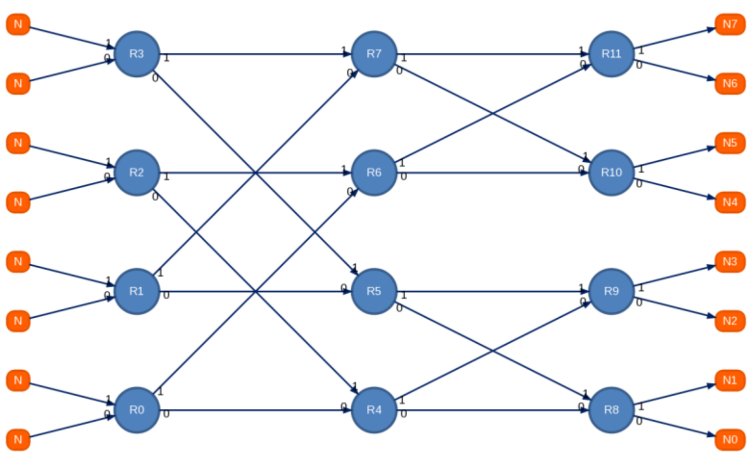


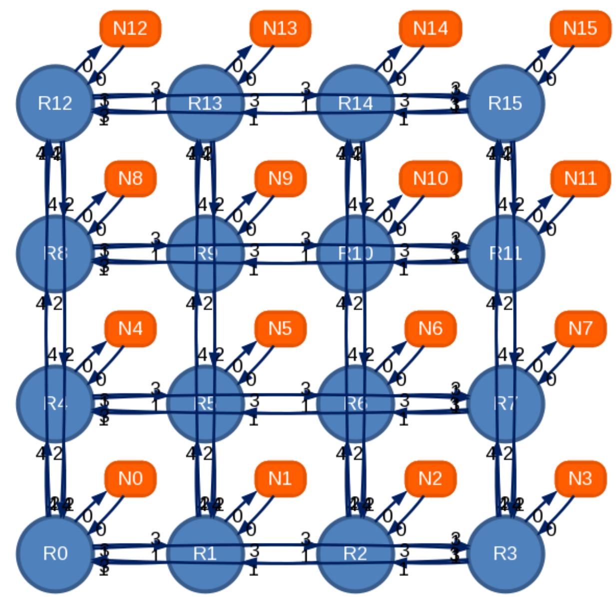
SoC - NoC Topology Examples

Some common topologies





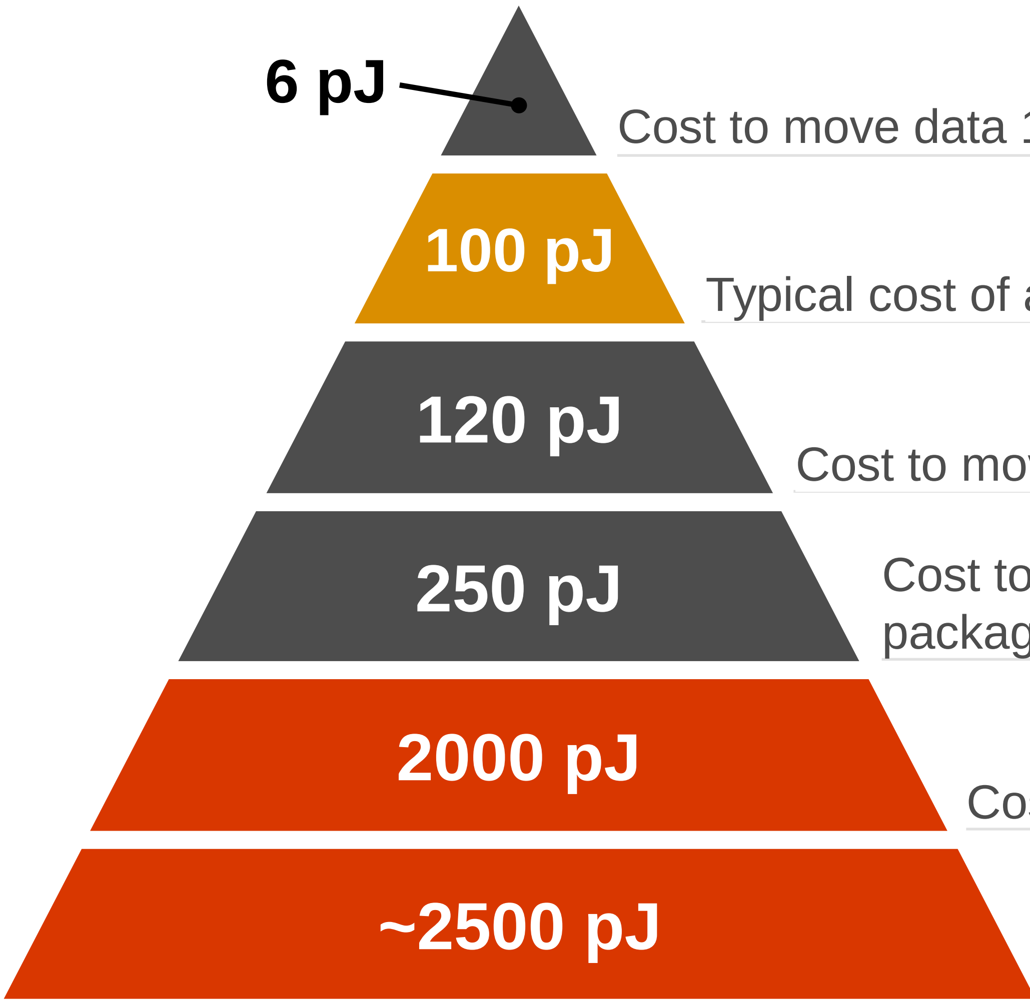




of
ce

Hierarchical Power Costs

Data movement is the dominant power cost



Cost to move data 1 mm on-chip

Typical cost of a single floating point operation

Cost to move data 20 mm on chip

Cost to move off-chip, but stay within the package (SMP)

Cost to move data off chip into DRAM

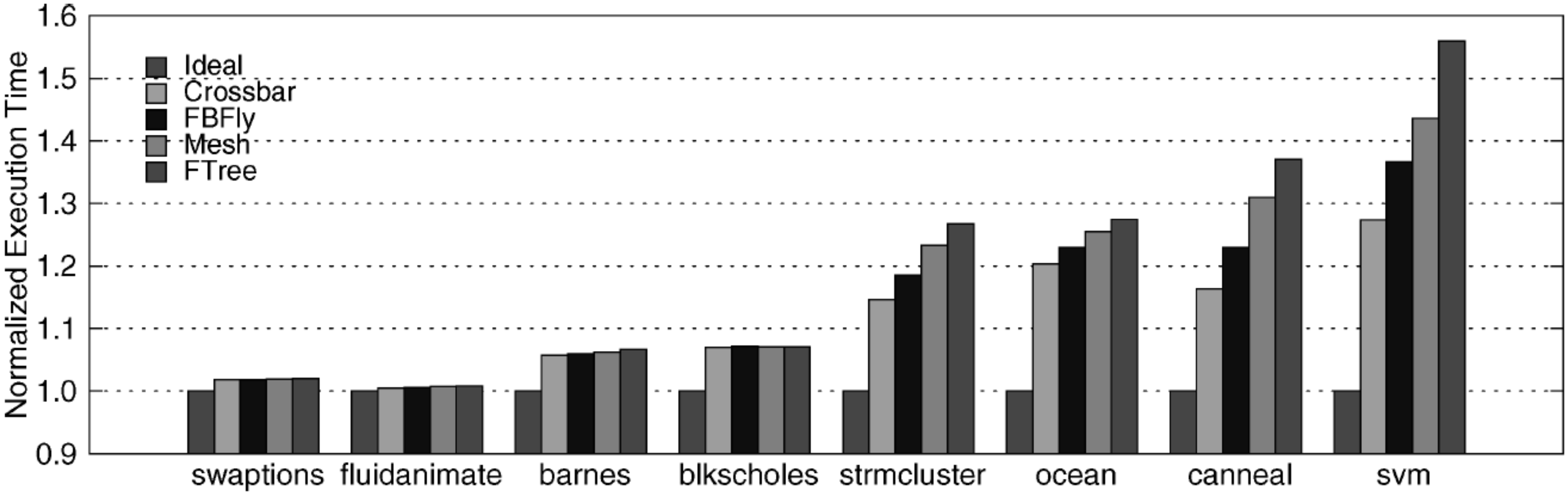
Cost to move data off chip to a neighboring node

BERKELEY LAB

U.S. DEPARTMENT OF ENERGY | Office of Science

Network Architecture Impact

Topology choice influences application performance

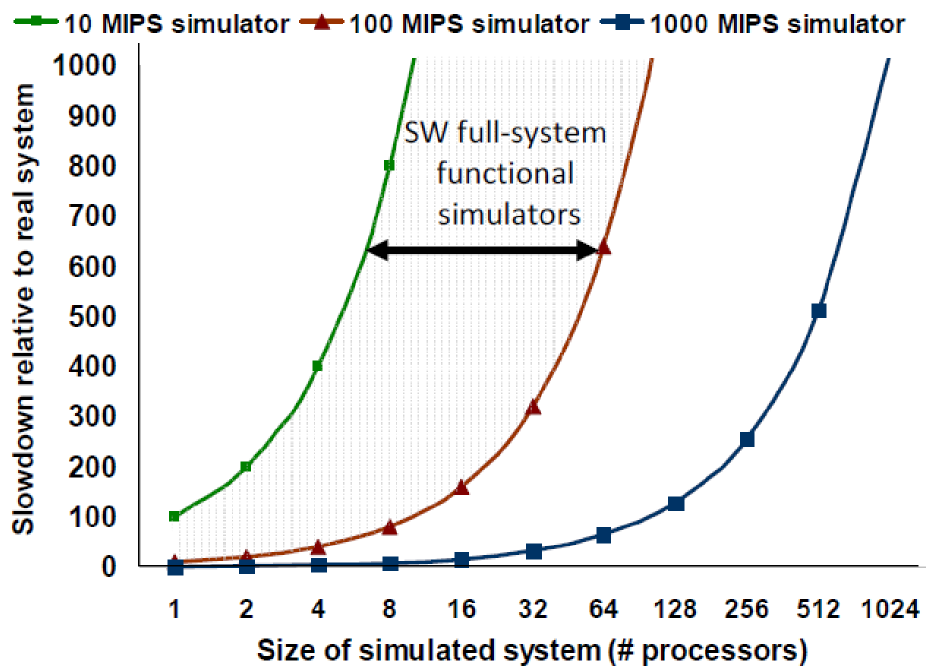


An analysis of on-chip interconnection networks for large-scale chip multiprocessors
ACM Transactions on computer architecture and code optimization (TACO), April 2010

What tools exist for NoC research

What Tools Do We Have to Evaluate Large, Complex Networks of Cores?

- **Software models**
  - Fast to create, but plagued by long runtimes as system size increases
- **Hardware emulation**
  - Fast, accurate evaluate that scales with system size but suffers from long development time

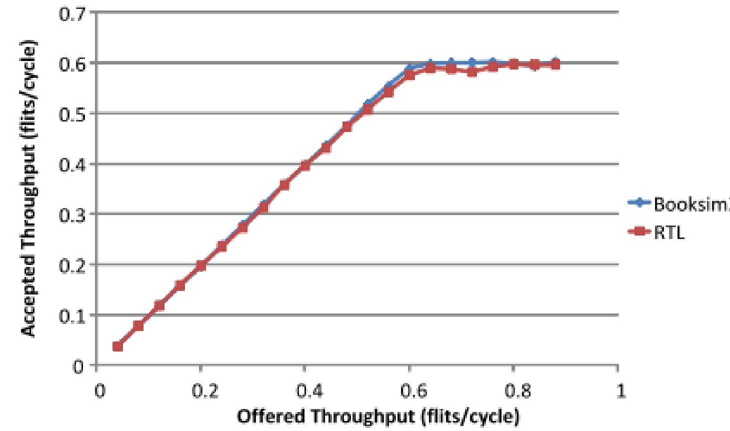


A complexity-effective architecture for accelerating full-system multiprocessor simulations using FPGAs. FPGA 2008

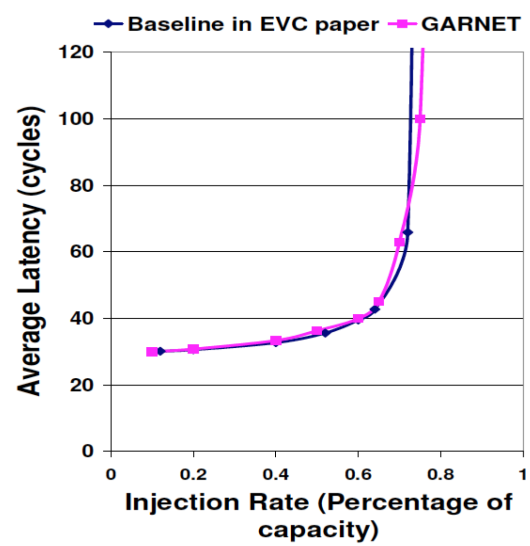
Software Models

C++ based on-chip network simulators

- **Booksim**
  - Cycle-accurate
  - Verified against RTL
  - Few thousand cycles per second
- **Garnet**
  - Event driven
  - Simulation speed limits designs to 100’s of cores



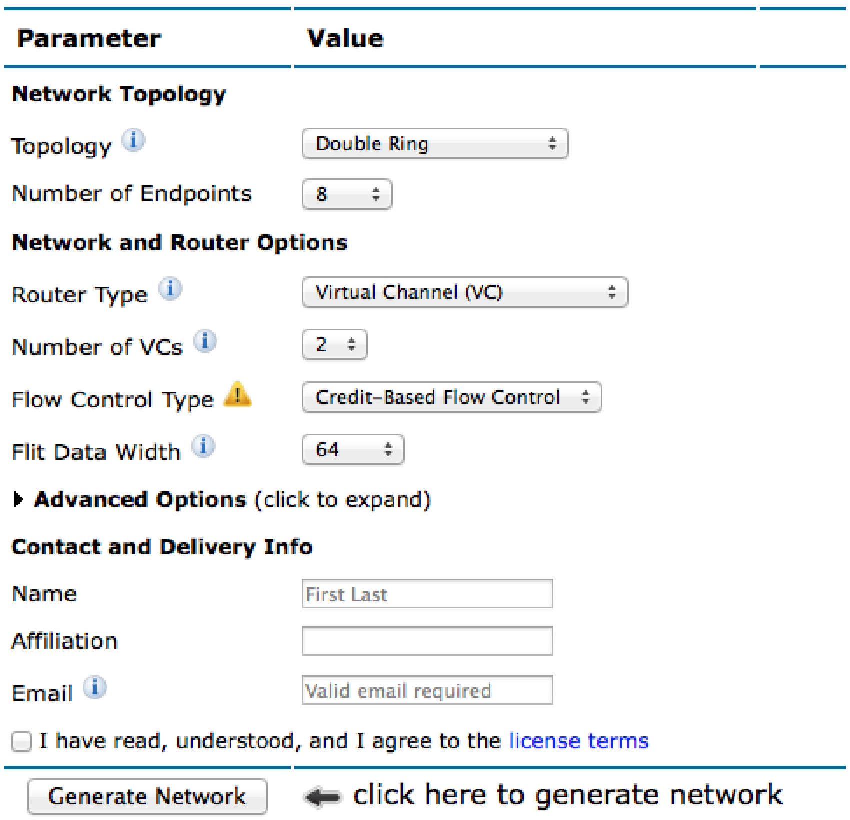
Booksim ISPASS 2013



GARNET ISPASS 2009

Hardware Models

HDL network generators and implementations



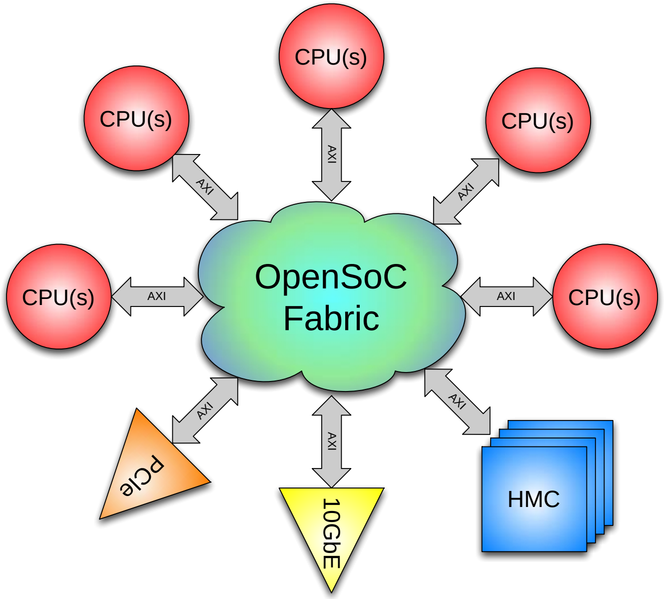
CONNECT: fast flexible FPGA-tuned networks-on-chip.
CARL 2012

- **Stanford open-source NoC router**
  - Verilog
  - Precise but long simulation times
- **Connect network generation**
  - Bluespec
  - FPGA Optimized

OpenSoC Fabric

An Open-Source, Flexible, Parameterized, NoC Generator

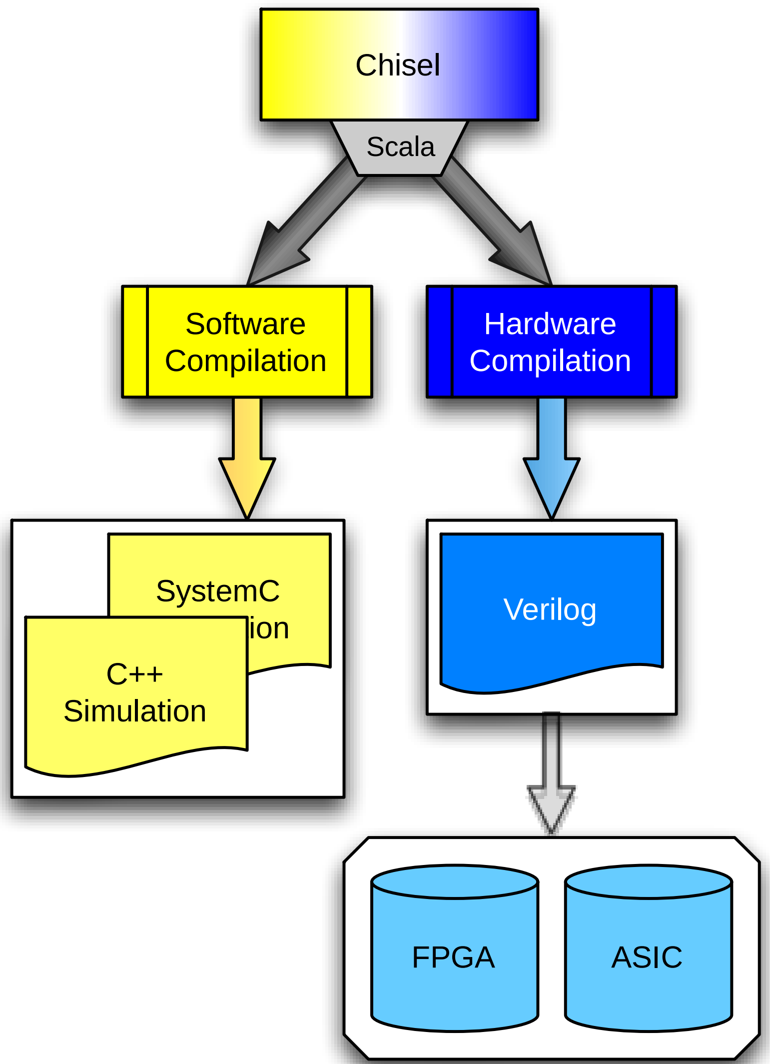
- **SoC technology gaining momentum in HPC**
  - On-chip networks evolving from simple crossbar to sophisticated networks
  - Need new tools and techniques to evaluate tradeoffs
- **Chisel-based**
  - Allows high level of parameterization
    - Dimensions, topology, VCs, etc. all configurable
  - Fast, functional SW model with SystemC integration
  - Verilog model for FPGA and ASIC flows
- **Multiple Network Interfaces**
  - Integrate with Tensillica, RISC-V, ARM, etc.

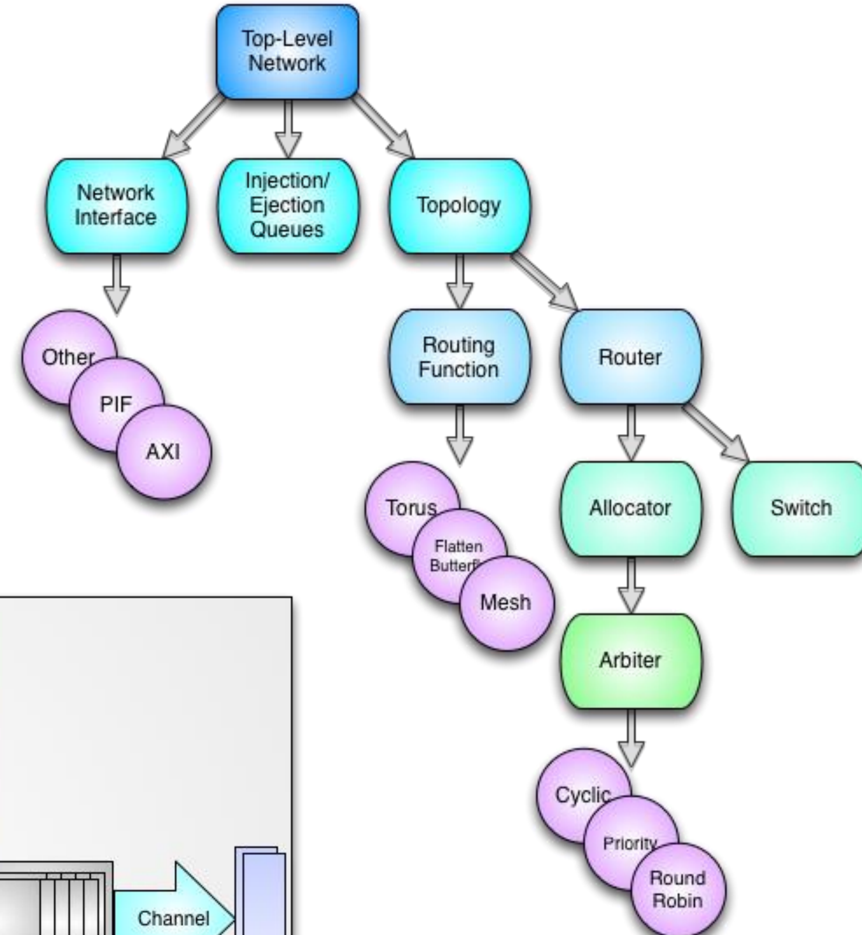


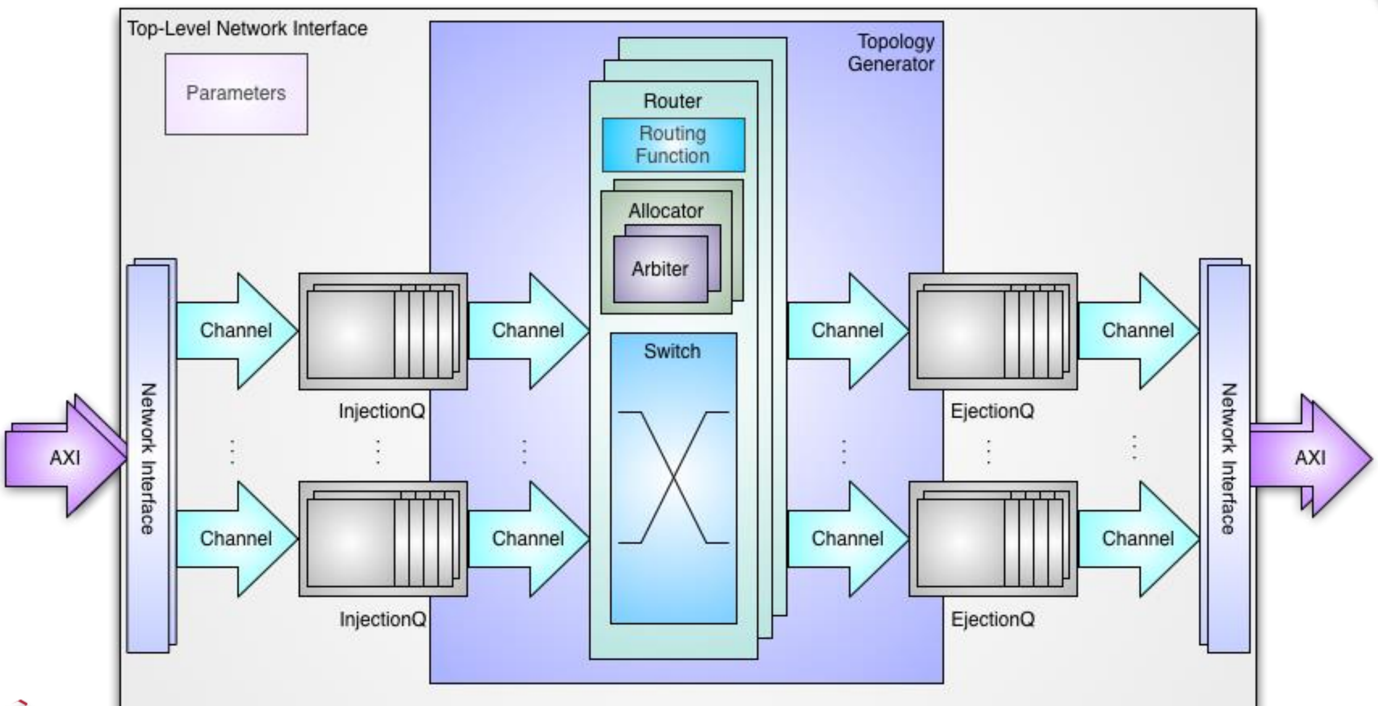
Chisel: Hardware DSL

Constructing Hardware In a Scala Embedded Language

- **Chisel provides both software and hardware models from the same codebase**
- **Object-oriented hardware development**
  - Allows definition of structs and other high-level constructs
- **Powerful libraries and components ready to use**
- **Working processors fabricated using Chisel**




Top-Level Network
Network Interface
Injection/ Ejection Queues
Topology
Other
PIF
AXI
Routing Function
Router
Torus
Flatten
Mesh
Allocator
Switch
Arbiter
Cyclic
Priority
Round Robin


Top-Level Network Interface
Parameters
Topology Generator
Router
Routing Function
Allocator
Arbiter
Switch
AXI
Network Interface
Channel
Channel
Channel
Channel
InjectionQ
InjectionQ
Channel
Channel
Channel
Channel
EjectionQ
EjectionQ
Network Interface
AXI

OpenSoC Configuration

OpenSoC is a fully configurable hardware generator

- **OpenSoC configured at run time through *Parameters* class**
  - Declared at top level, sub modules can add / change parameters tree
- **Not limited to just integer values**
  - Leverage Scala to pass functions to parameterize module creation
    - Example: Routing Function constructor passed as parameter to router

Configuration options

A few of the current run time configuration parameters

- **Network Parameters**
  - Dimension
  - Routers per dimension
  - Concentration
  - Virtual Channels
  - Topology
  - Queue depths
  - Routing Function

- **Packet / Flit Parameters**
  - Flit widths
  - Packet types / lengths
- **Testing Parameters**
  - Pattern
    - Neighbor, random, tornado, etc
  - Injection Rate

Highly modular architecture supports FUB replacement

Developing

Incredibly Fast Development Time

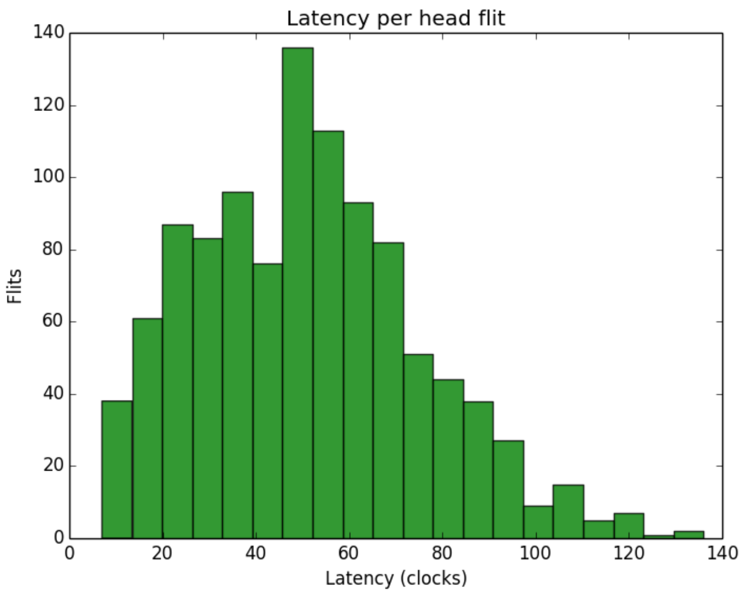
- **Modules have a standard interface that you inherit**
- **Development of modules is very quick**
  - Flattened Butterfly took 2 hours of development

```
abstract class Allocator(parms: Parameters)
    extends Module(parms) {
  val numReqs = parms.get[Int]("numReqs")
  val numRes = parms.get[Int]("numRes")
  val arbCtor = parms.get[Parameters=>Arbiter]
    ("arbCtor")
  val io = new Bundle {
    val requests = Vec.fill(numRes)
      { Vec.fill(numReqs)
        { new RequestIO }.flip }
    val resources = Vec.fill(numRes)
      { new ResourceIO }
    val chosens = Vec.fill(numRes)
      { UInt(OUTPUT, Chisel.log2Up(numReqs)) }
  }
}
class SwitchAllocator(parms: Parameters)
    extends Allocator(parms) {
  // Implementation
}
```

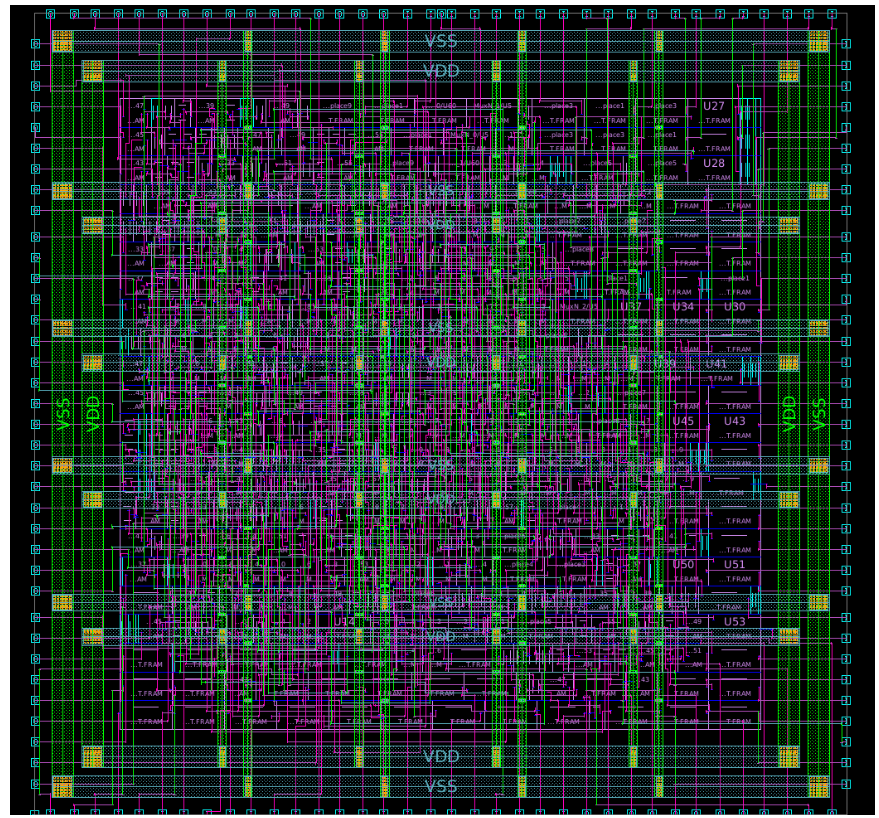
Results

4x4 DOR Mesh of Single Concentration with Uniform Random Traffic

Head Flit Latency



8x8 Dimension-Ordered Mesh
Concentration 1



More Information and Download

http://www.opensocfabric.org

Join us for an SoC for HPC workshop at DAC 2015