TILERA®

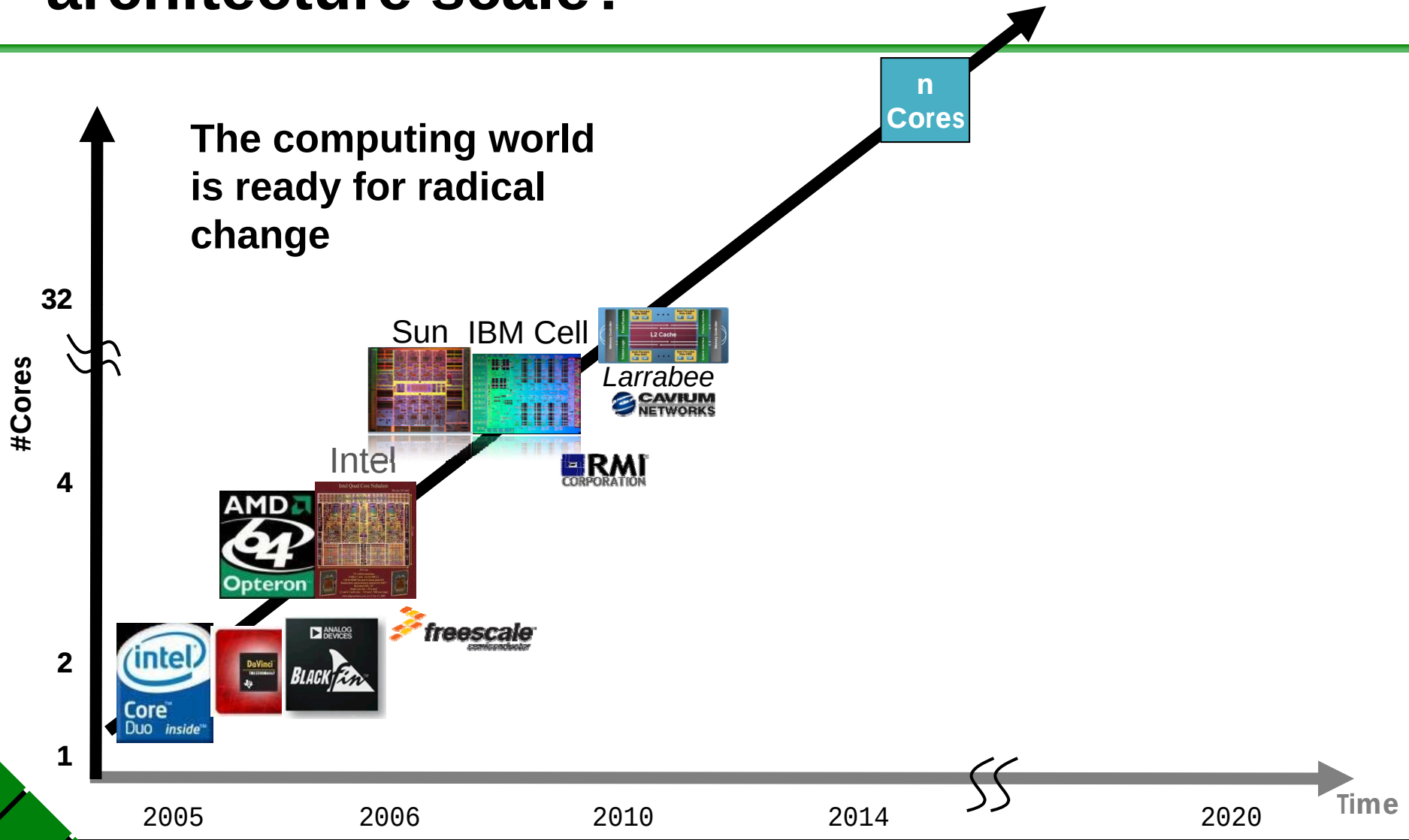
Tile Processors: Many-Core for Embedded and Cloud Computing

Richard Schooler
VP Software Engineering
Tilera Corporation
rschooler@tilera.com

Exploiting Natural Parallelism

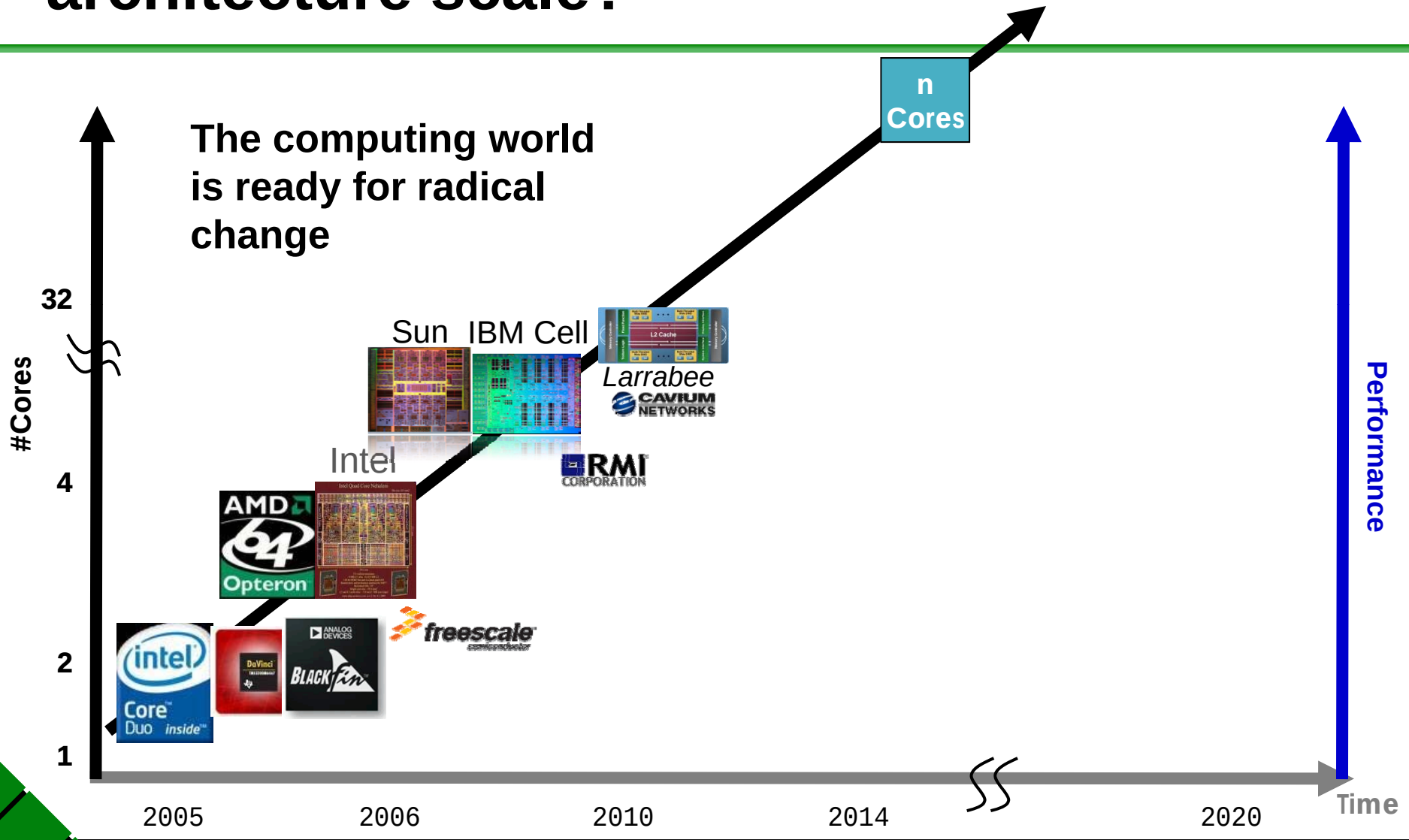
- High-performance applications have lots of parallelism!
  - Embedded apps:
    - Networking: packets, flows
    - Media: streams, images, functional & data parallelism
  - Cloud apps:
    - Many clients: network sessions
    - Data mining: distributed data & computation
- Lots of different levels:
  - SIMD (fine-grain data parallelism)
  - Thread/process (medium-grain task parallelism)
  - Distributed system (coarse-grain job parallelism)

Every one is going Manycore, but can the architecture scale?



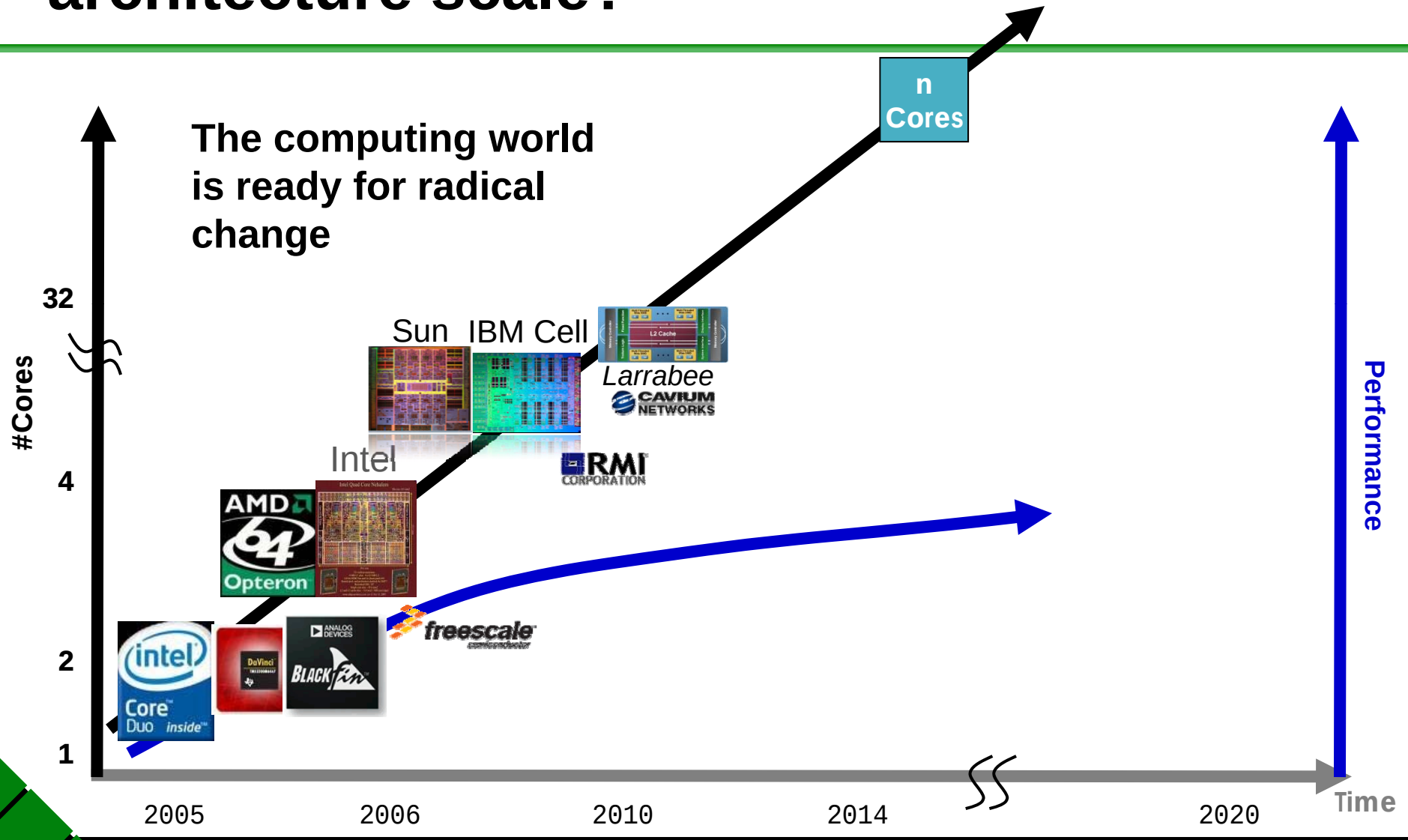
TILERA

Every one is going Manycore, but can the architecture scale?



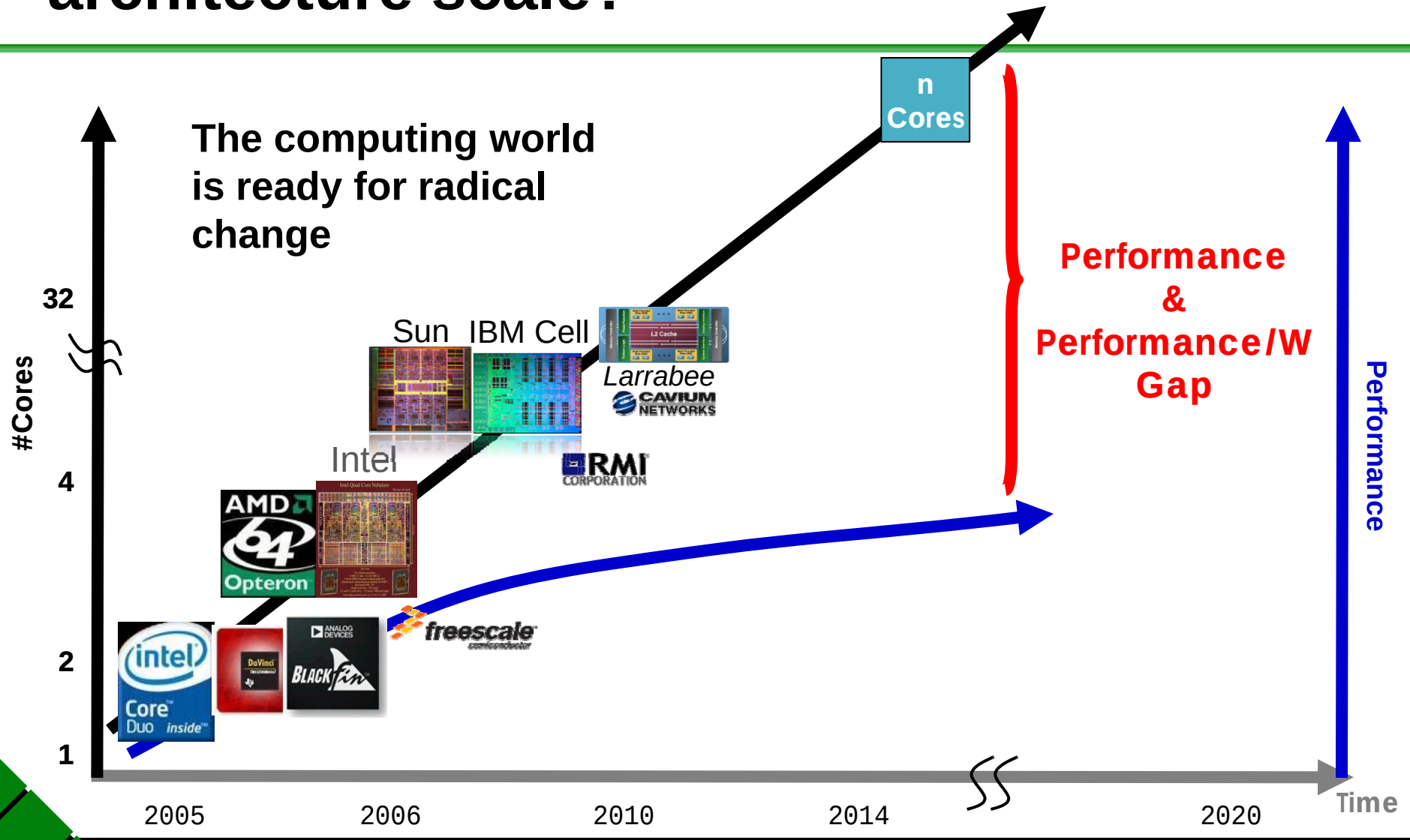
TILERA

Every one is going Manycore, but can the architecture scale?



TILERA

Every one is going Manycore, but can the architecture scale?

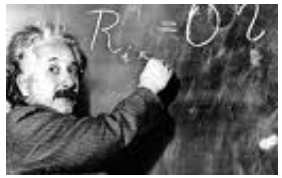


TILERA

Key “Many-Core” Challenges: The 3 P’s

- **P**erformance challenge
  - How to scale from 1 to 1000 cores – the number of cores is the new Megahertz
- **P**ower efficiency challenge
  - Performance per watt is the new metric – systems are often constrained by power & cooling
- **P**rogramming challenge
  - How to provide a converged many core solution in a standard programming environment

“Problems cannot be solved by the same level of thinking that created them.”



- Current technologies fail to deliver
  - Incremental performance increase
  - High power
  - Low level of Integration
  - Increasingly bigger cores
- We need to have a new thinking to get
  - 10 x performance
  - 10 x performance per watt
  - Converged computing
  - Standard programming models

Stepping Back: How Did We Get Here?

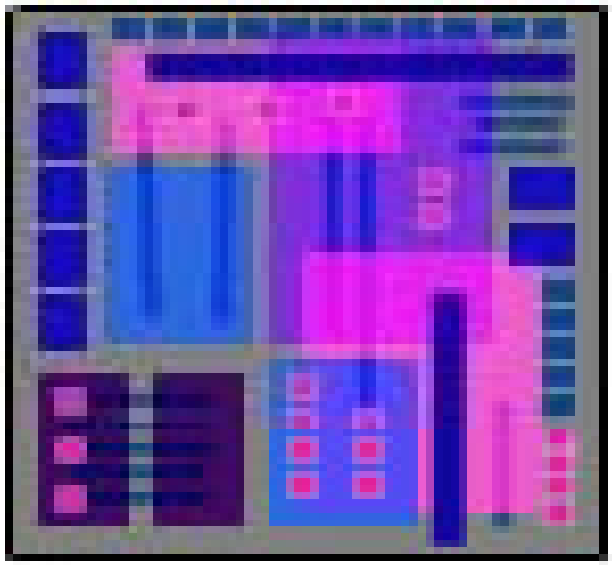
- Moore's Conundrum:

  More devices =>? More performance

- Old answers: More complex cores; bigger caches
  - But power-hungry
- New answers: More cores
  - But do conventional approaches scale?
- Diminishing returns!

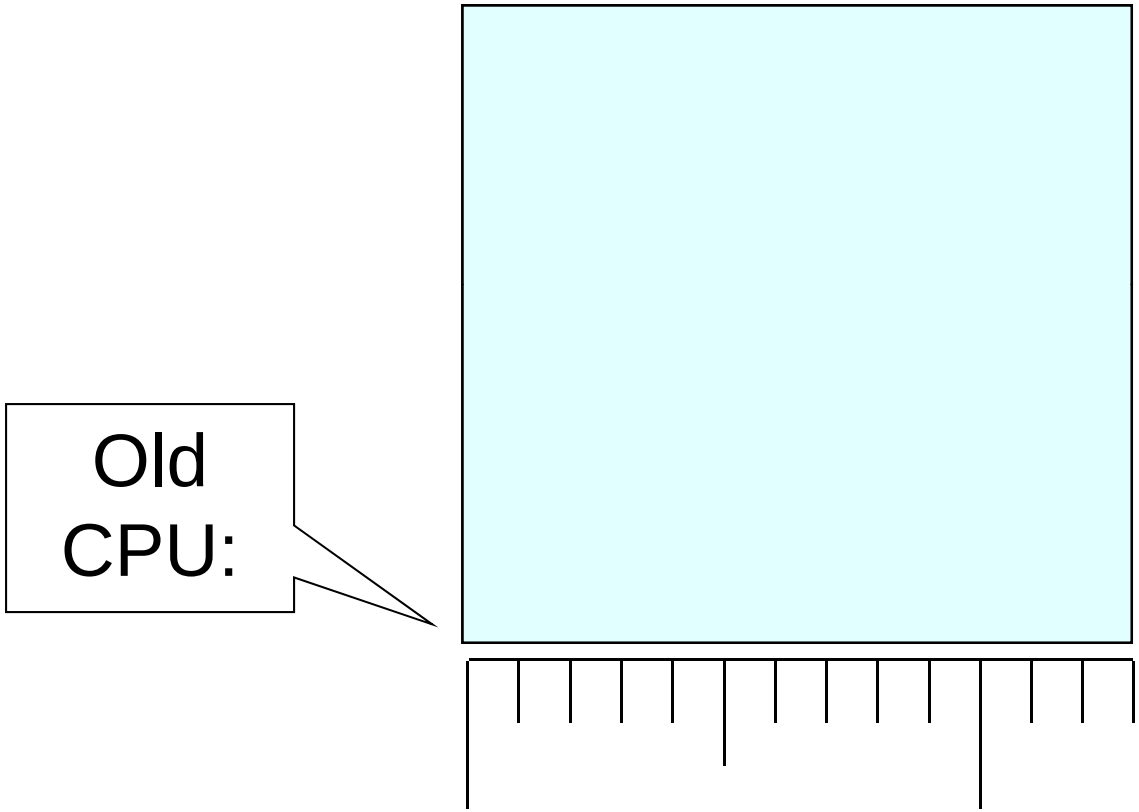
The Old Challenge: CPU-on-a-chip

20 MIPS CPU in 1987



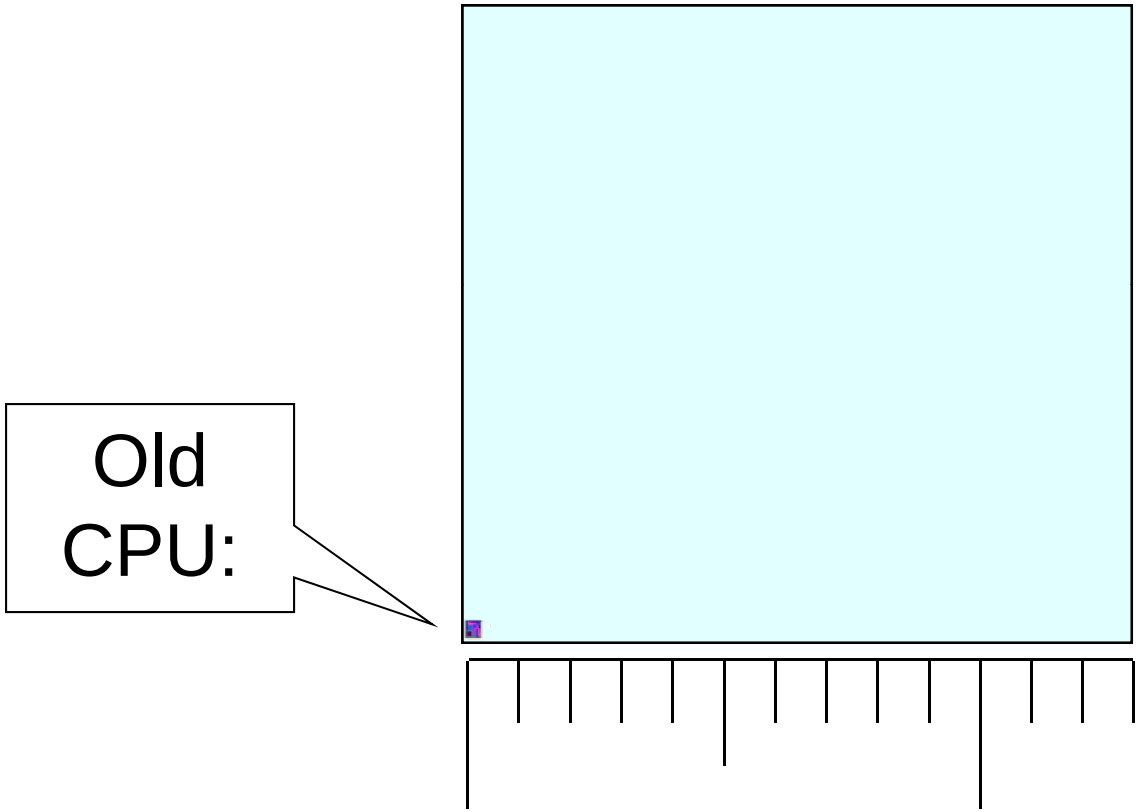
Few thousand gates

The Opportunity: Billions of Transistors



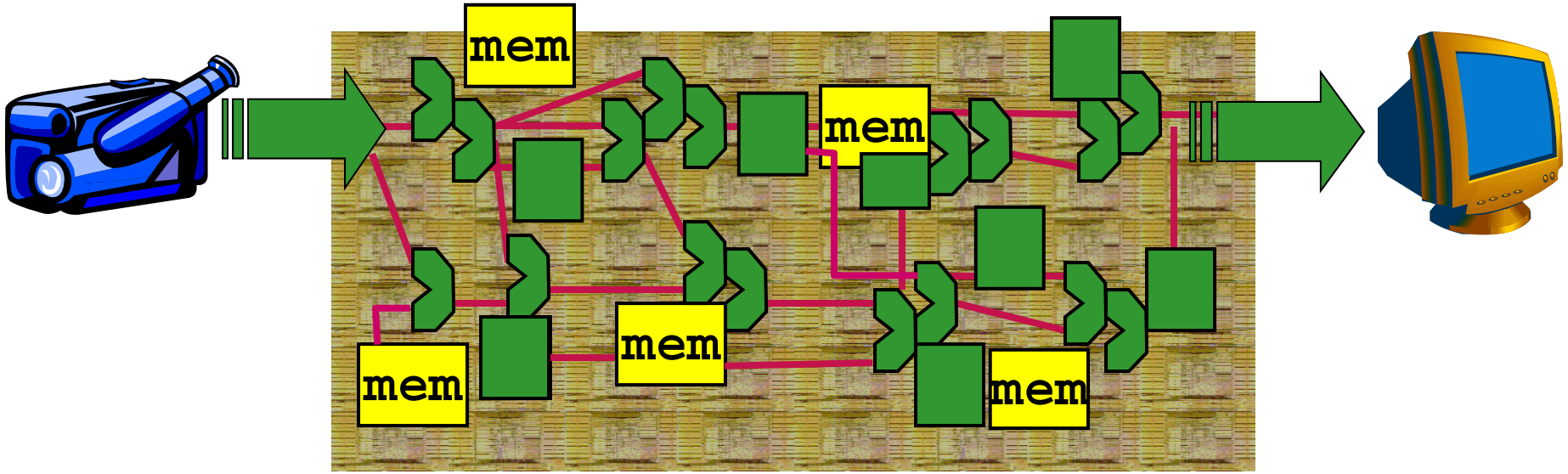
What to do with all those transistors?

The Opportunity: Billions of Transistors



What to do with all those transistors?

Take Inspiration from ASICs

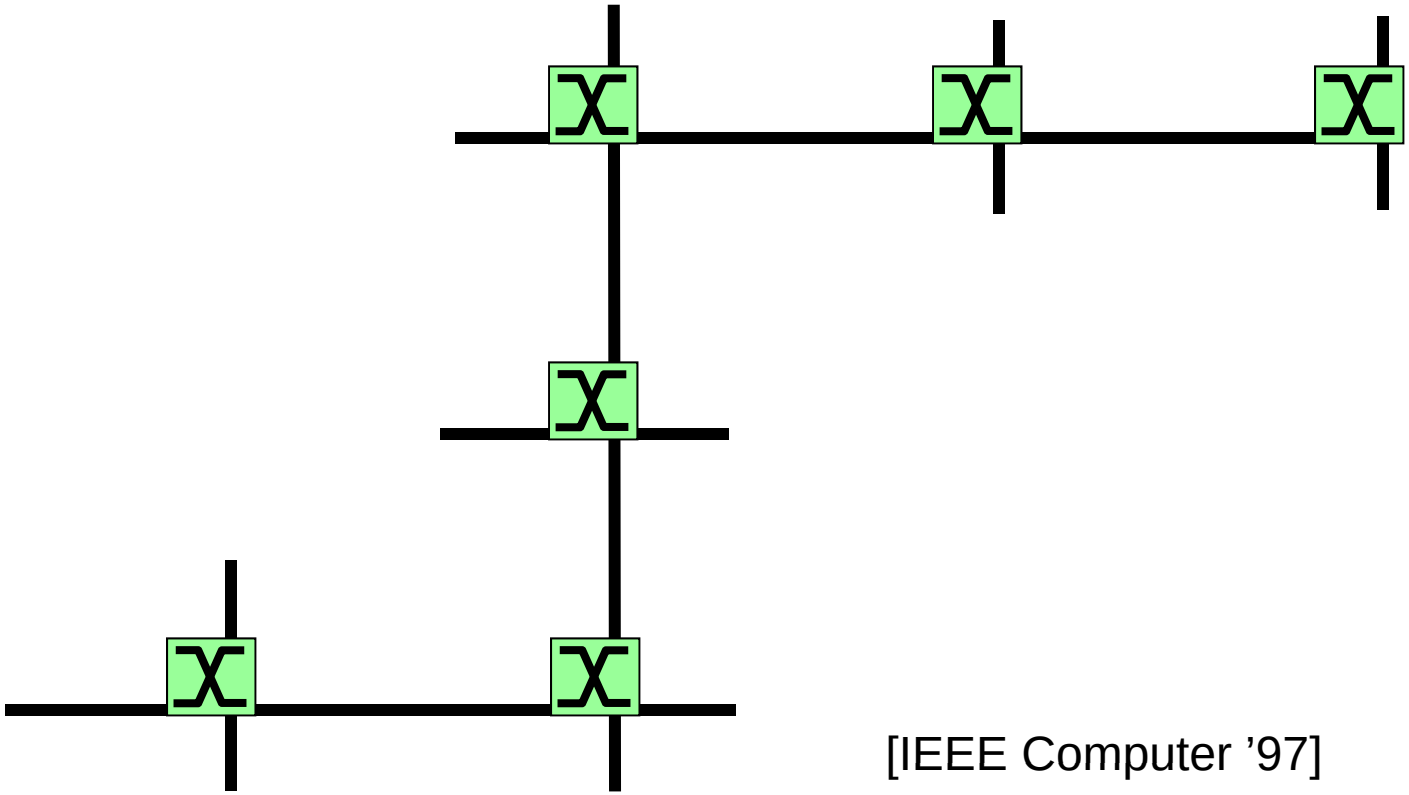


ASICs have high performance and low power

- Custom-routed, short wires
- Lots of ALUs, registers, memories – huge on-chip parallelism

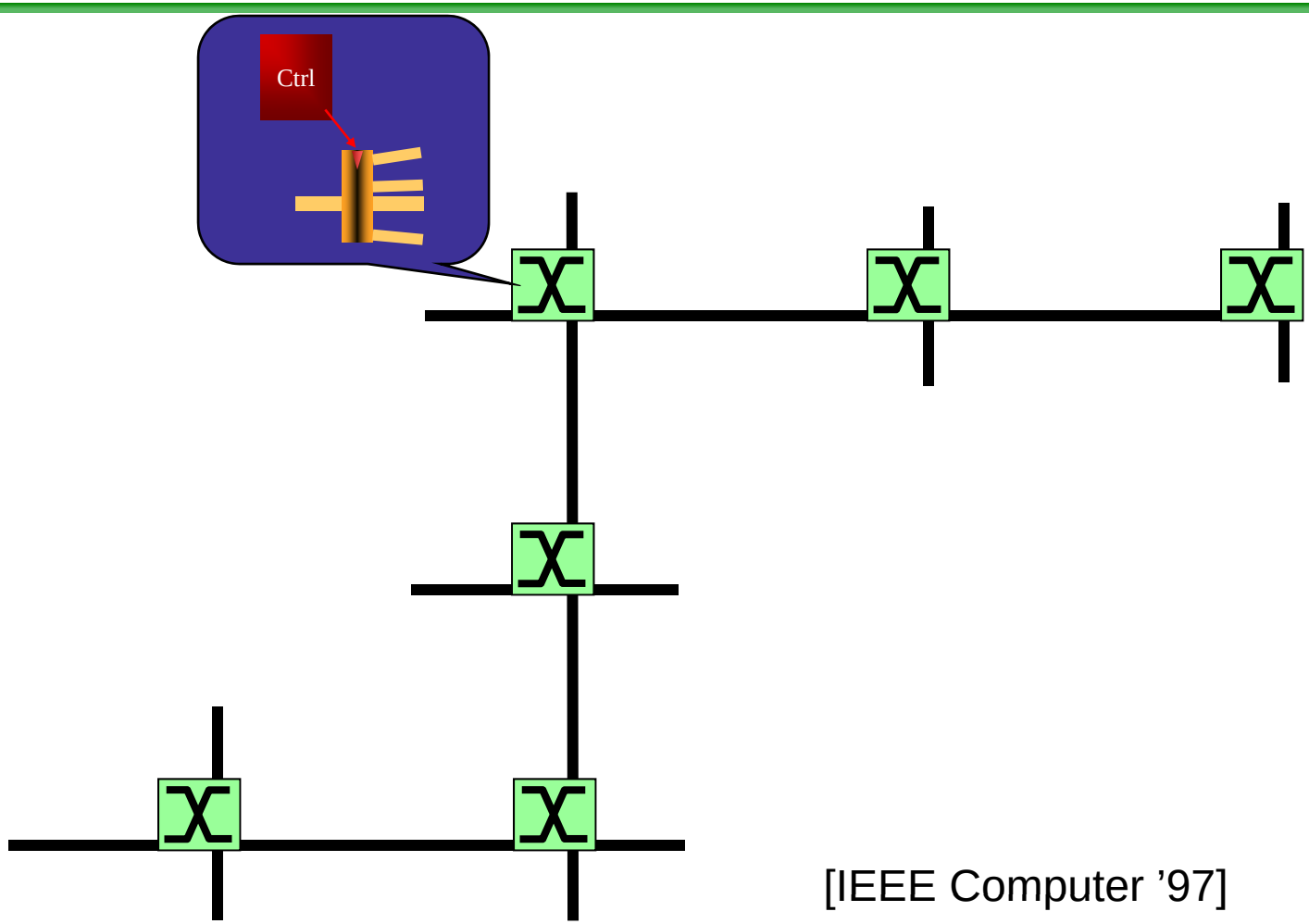
But how to build a programmable chip?

Replace Long Wires with Routed Interconnect



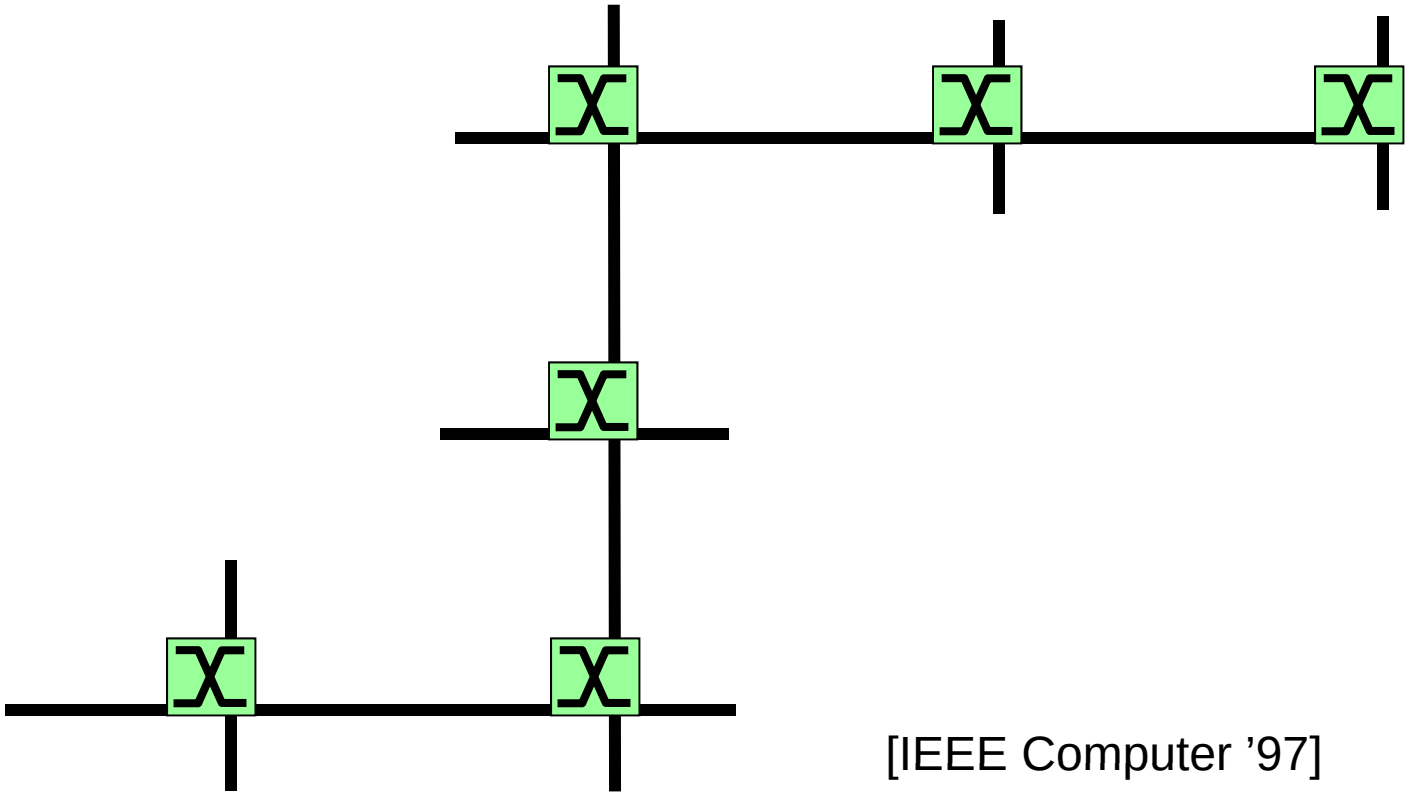
[IEEE Computer ’97]

Replace Long Wires with Routed Interconnect



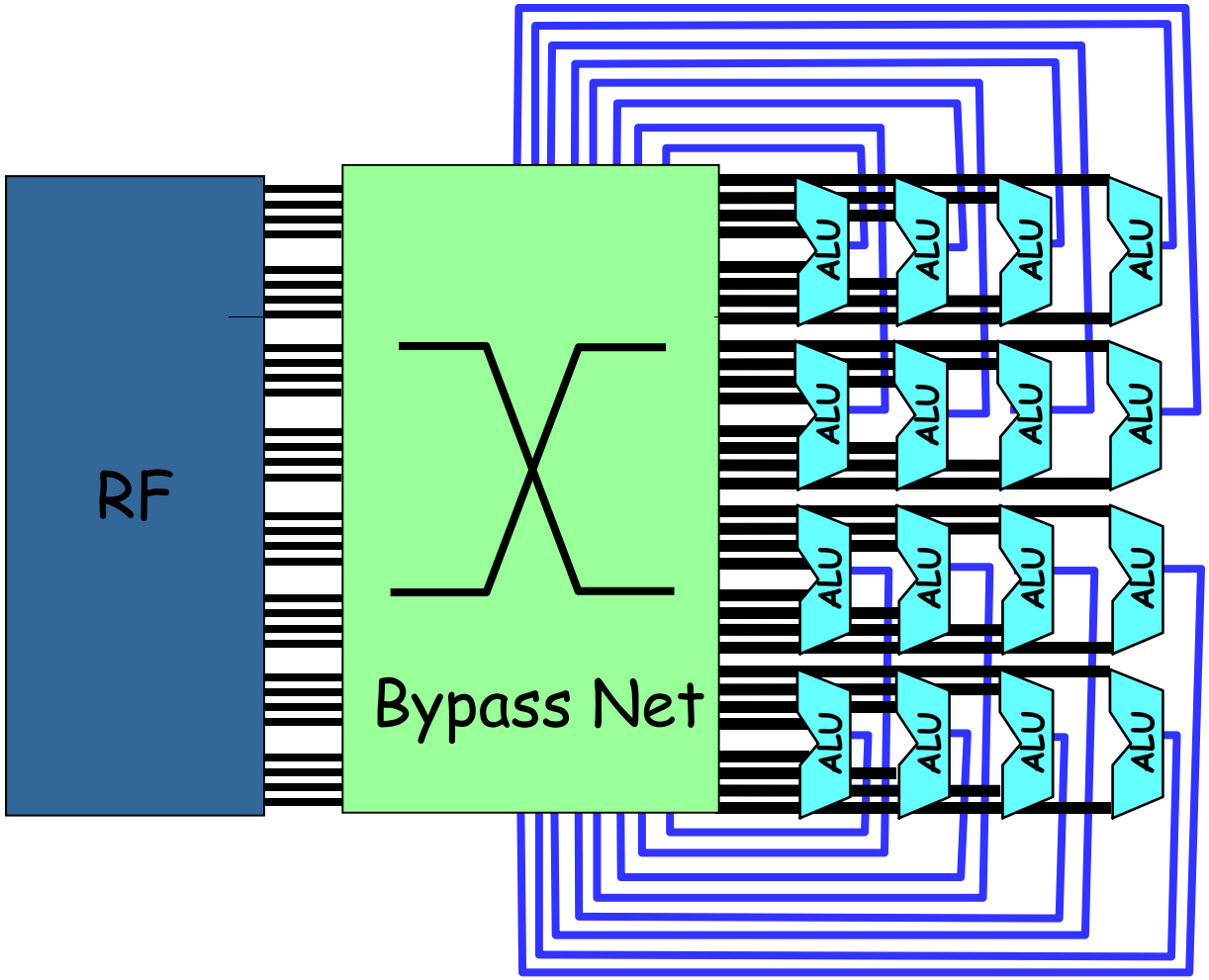
[IEEE Computer ’97]

Replace Long Wires with Routed Interconnect

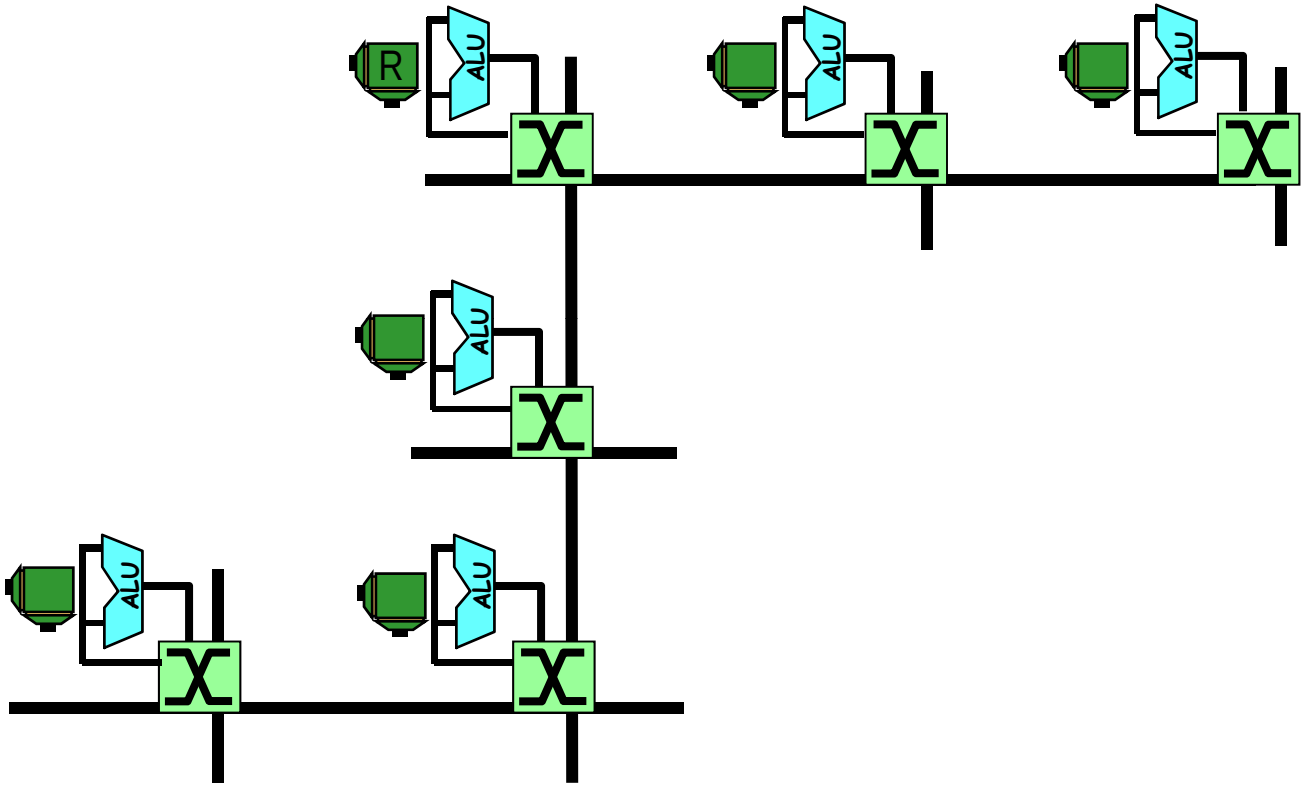


[IEEE Computer ’97]

From Centralized Clump of CPUs …

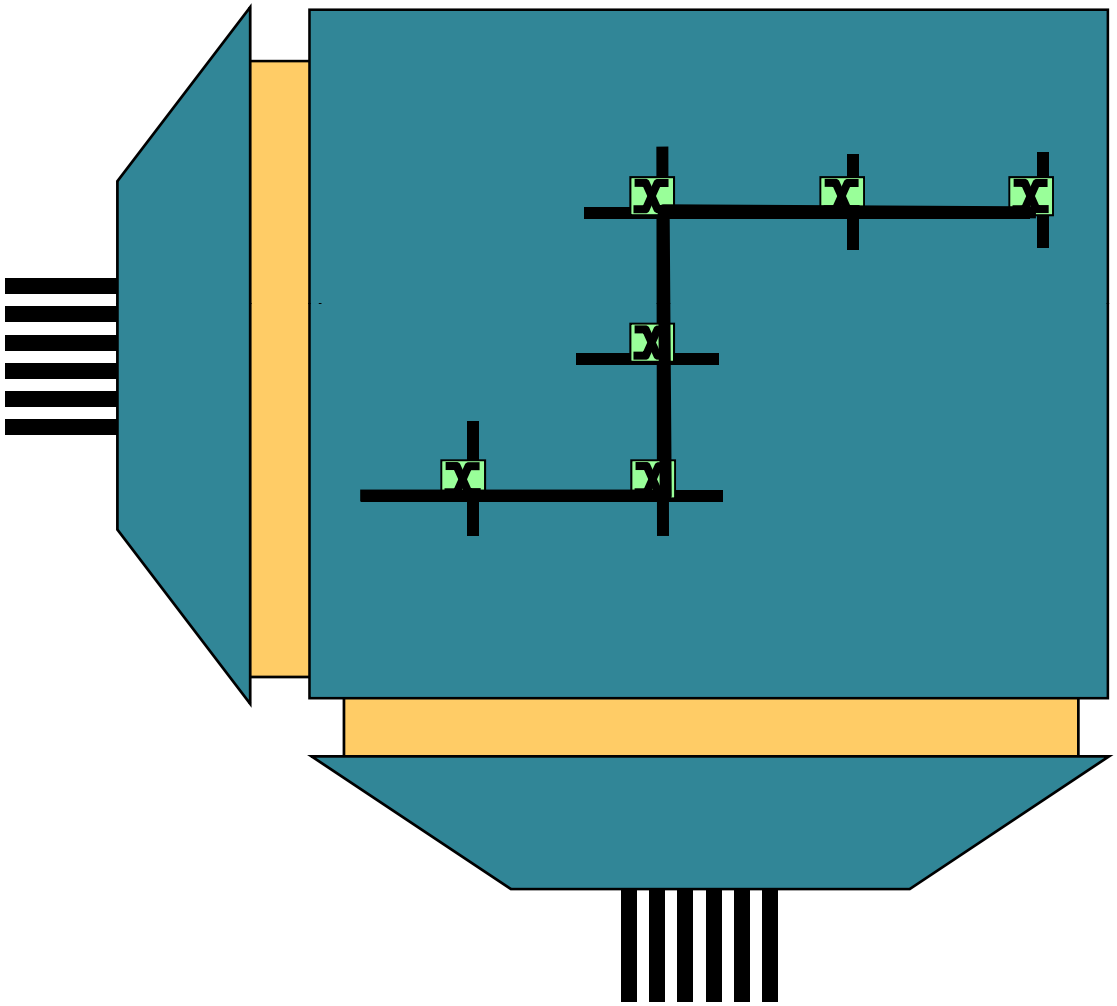


… To Distributed ALUs, Routed Bypass Network

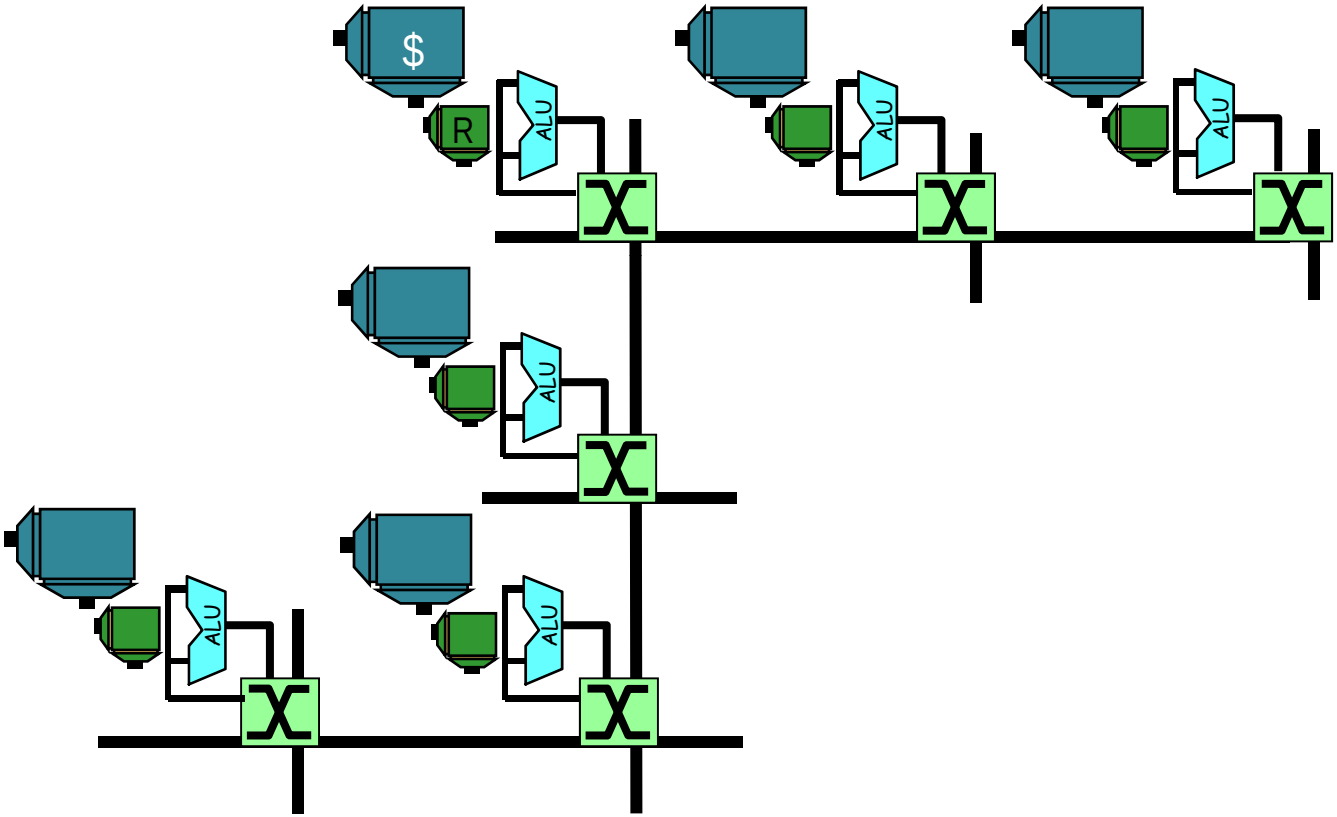


Scalar Operand Network (SON) [TPDS 2005]

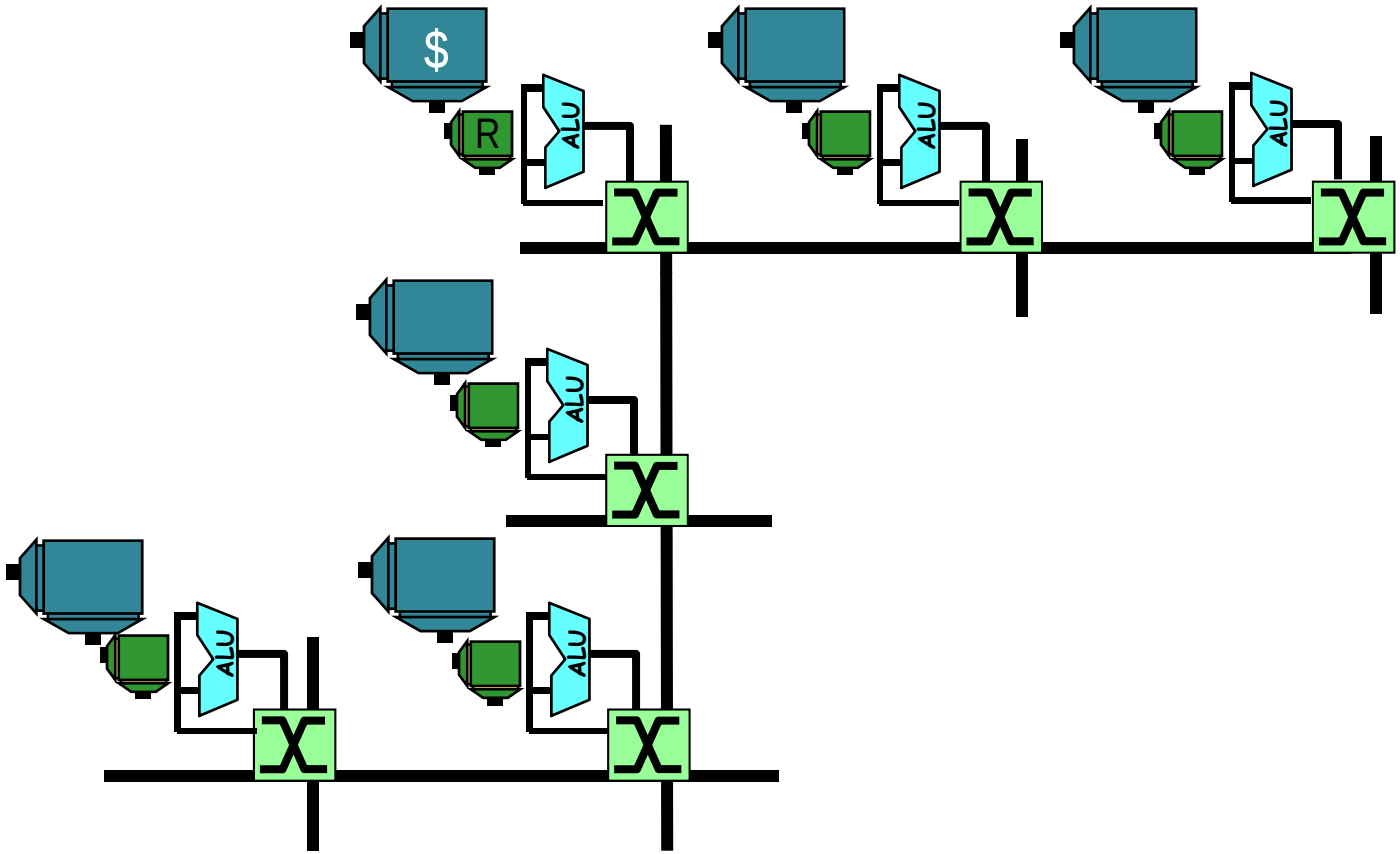
From a Large Centralized Cache…



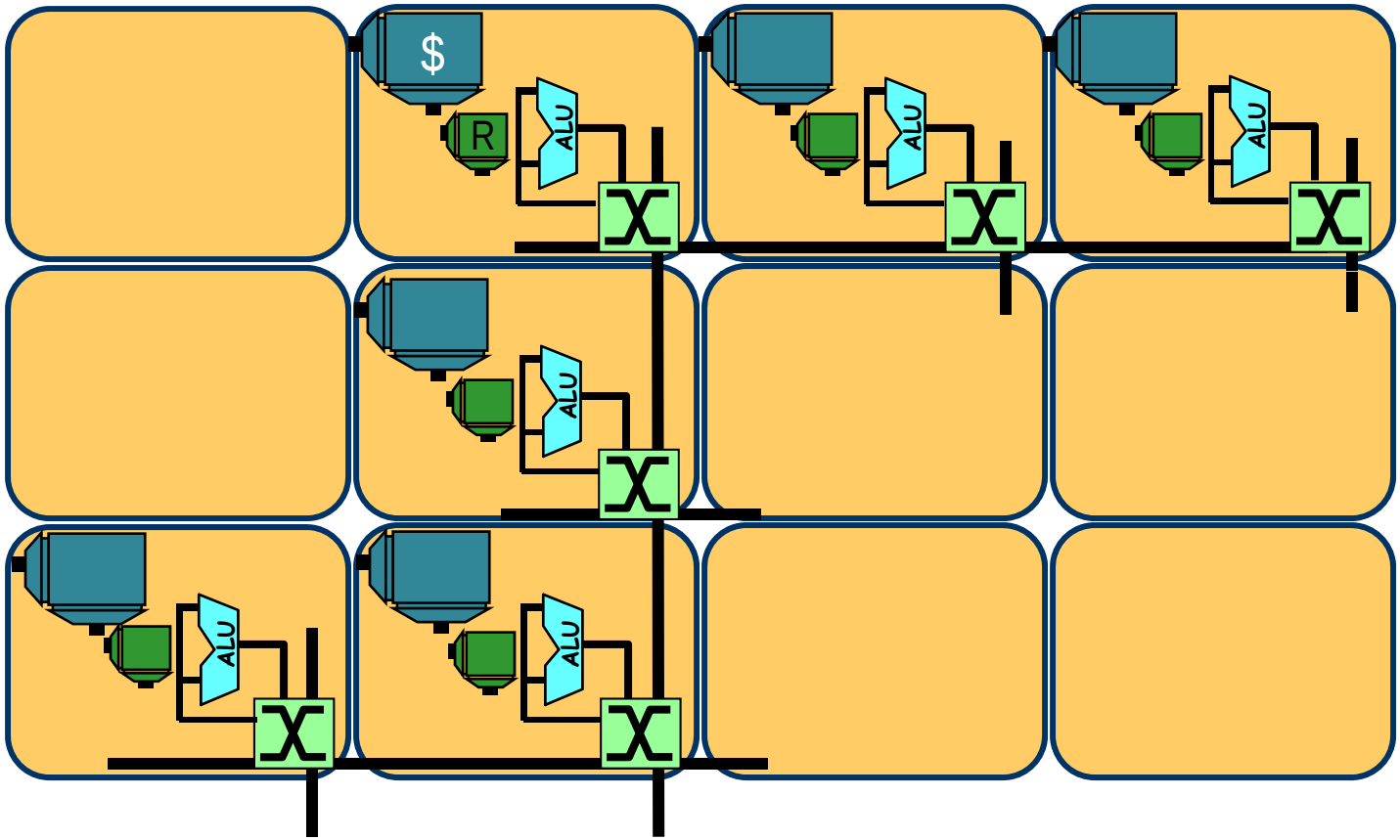
…to a Distributed Shared Cache



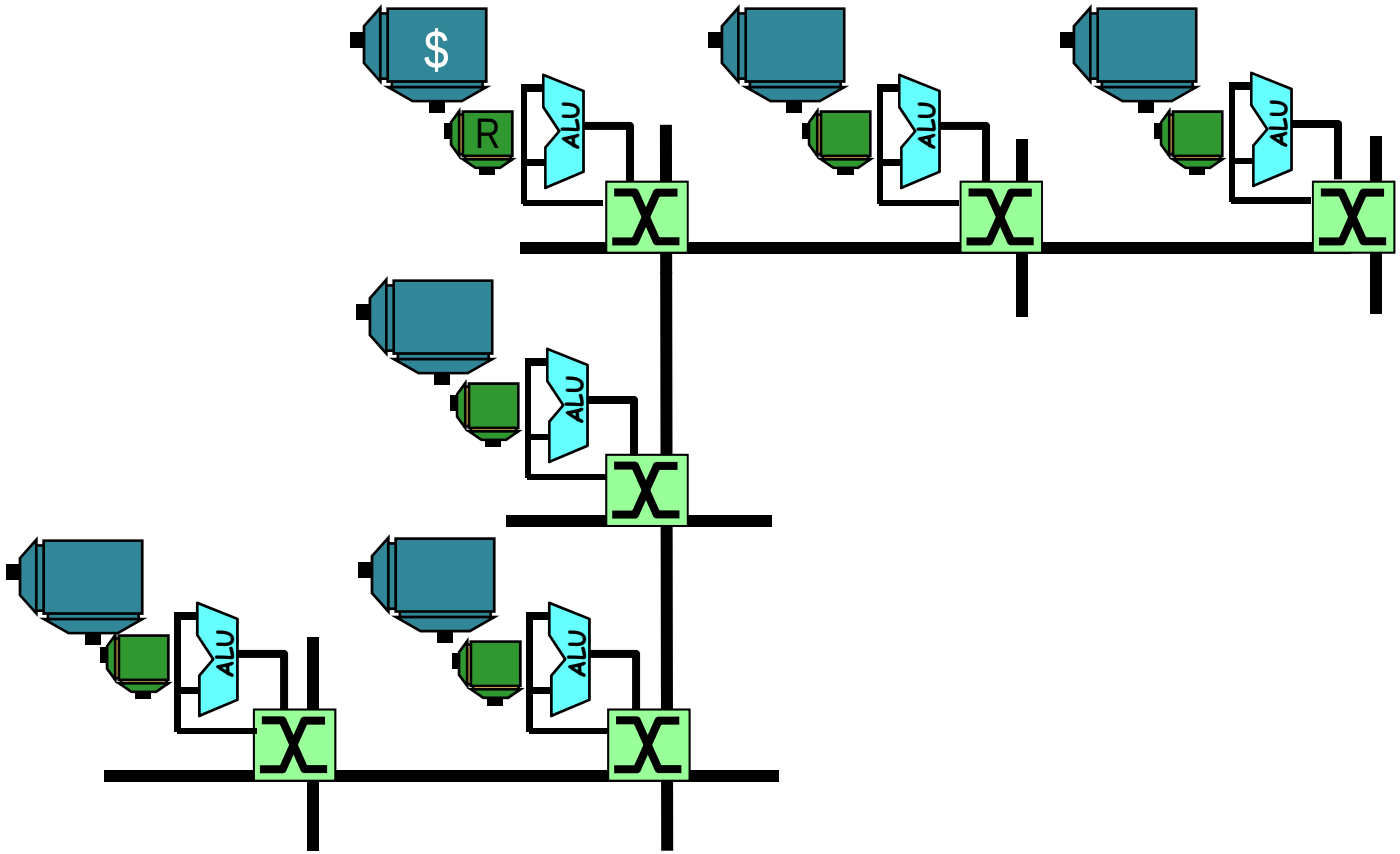
Distributed Everything + Routed Interconnect → Tiled Multicore



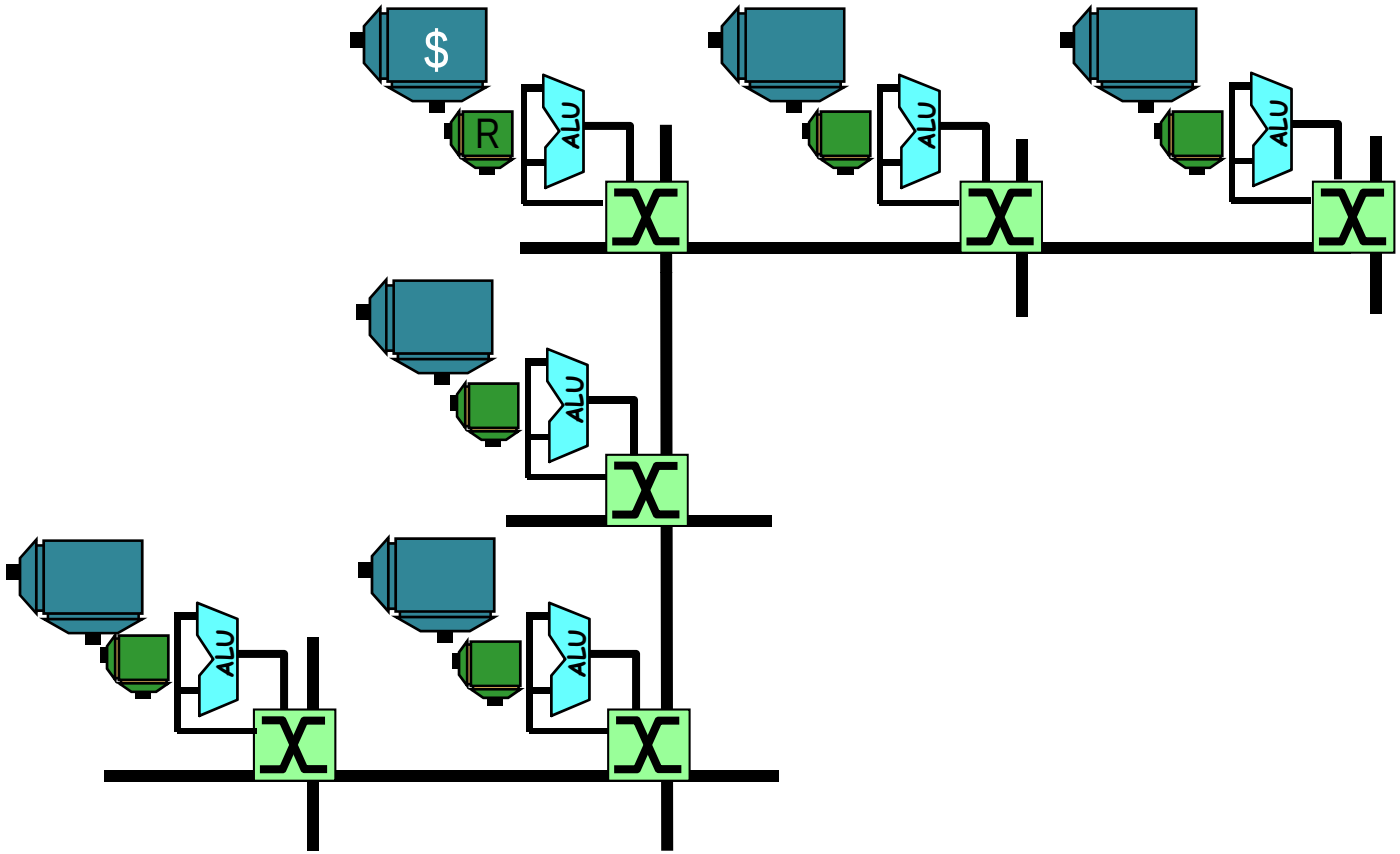
Distributed Everything + Routed Interconnect → Tiled Multicore



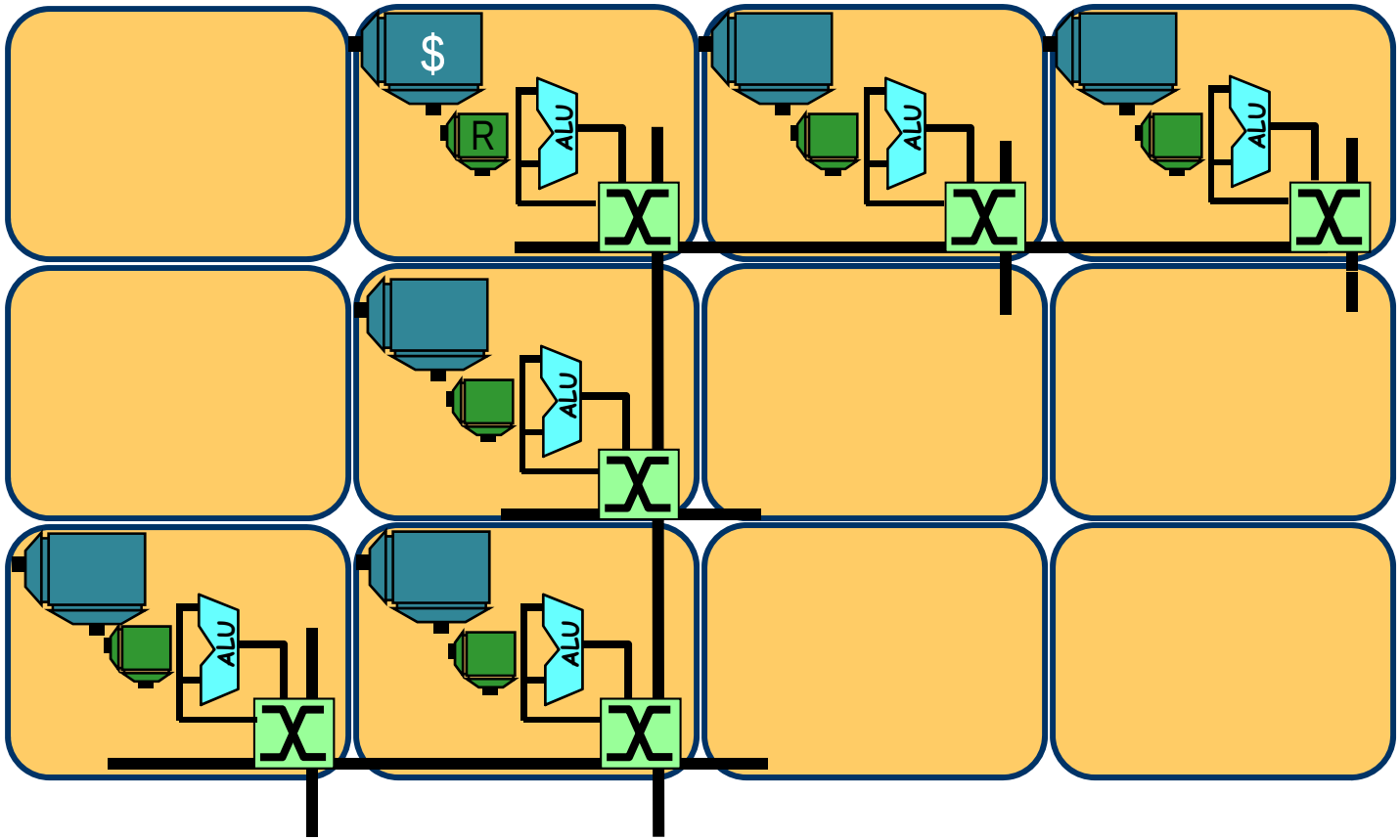
Distributed Everything + Routed



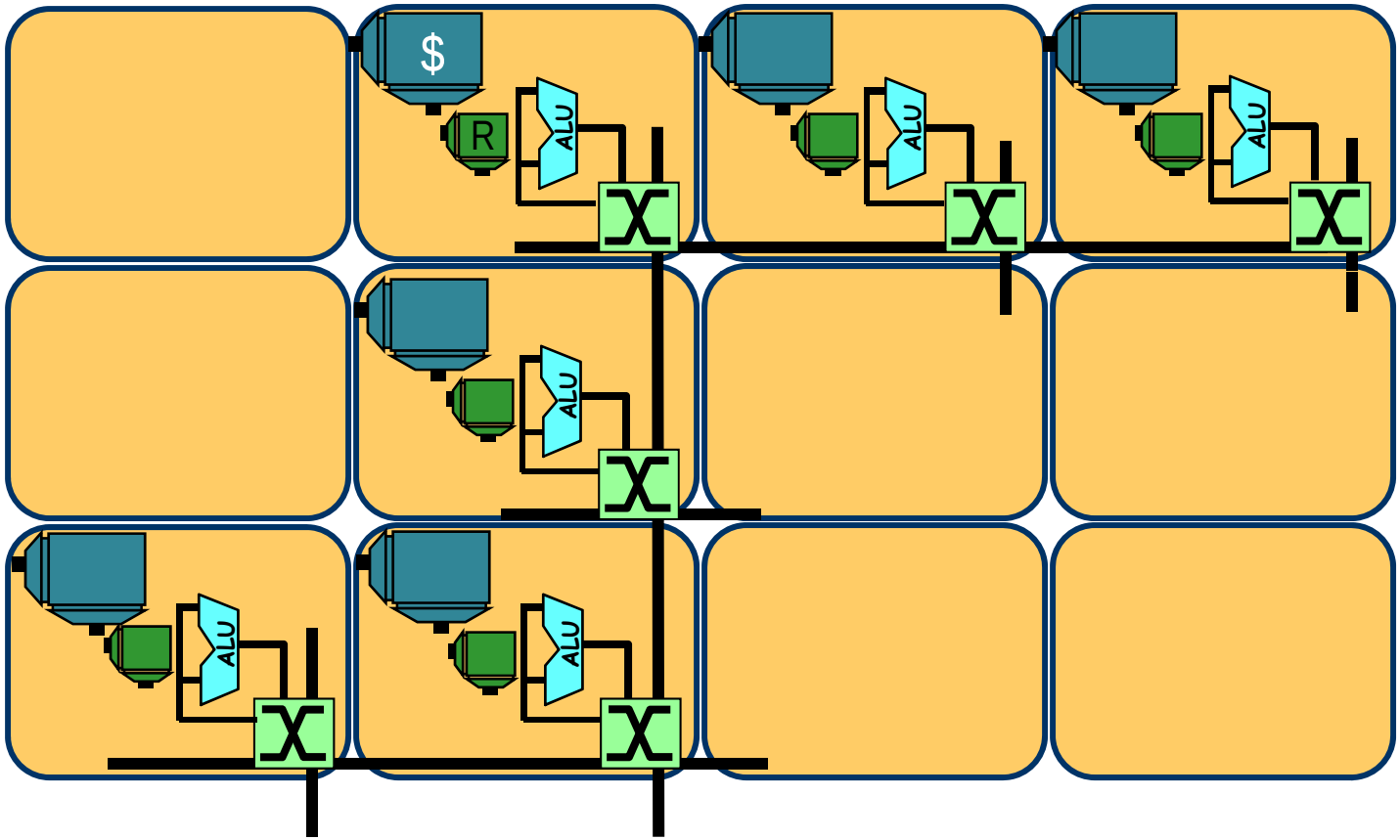
Distributed Everything + Routed Interconnect → Tiled Multicore



Distributed Everything + Routed Interconnect → Tiled Multicore



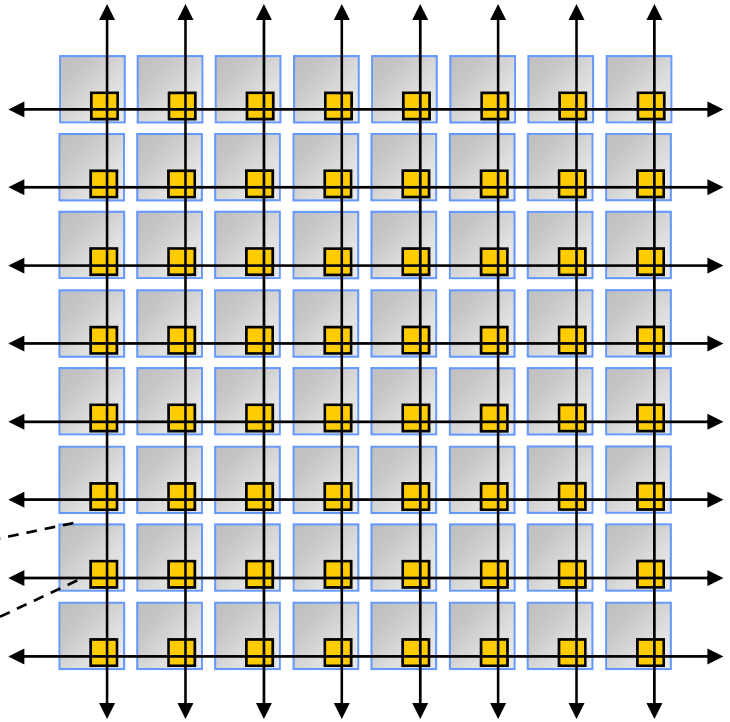
Distributed Everything + Routed Interconnect → Tiled Multicore

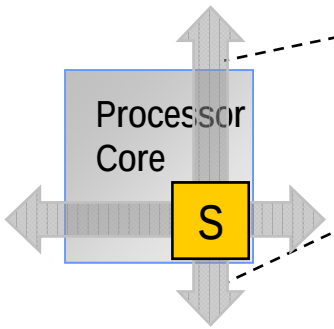


Each tile is a processor, so programmable

Tiled Multicore Captures ASIC Benefits and is Programmable

- Scales to large numbers of cores
- Modular – design and verify 1 tile
- Power efficient
  - Short wires plus locality opts – CV^2f
  - Chandrakasan effect, more cores at lower freq and voltage – CV^2f

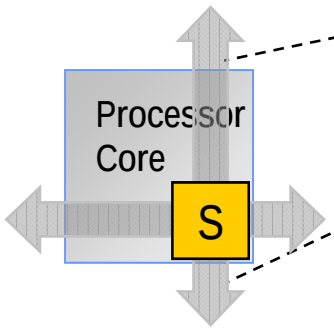




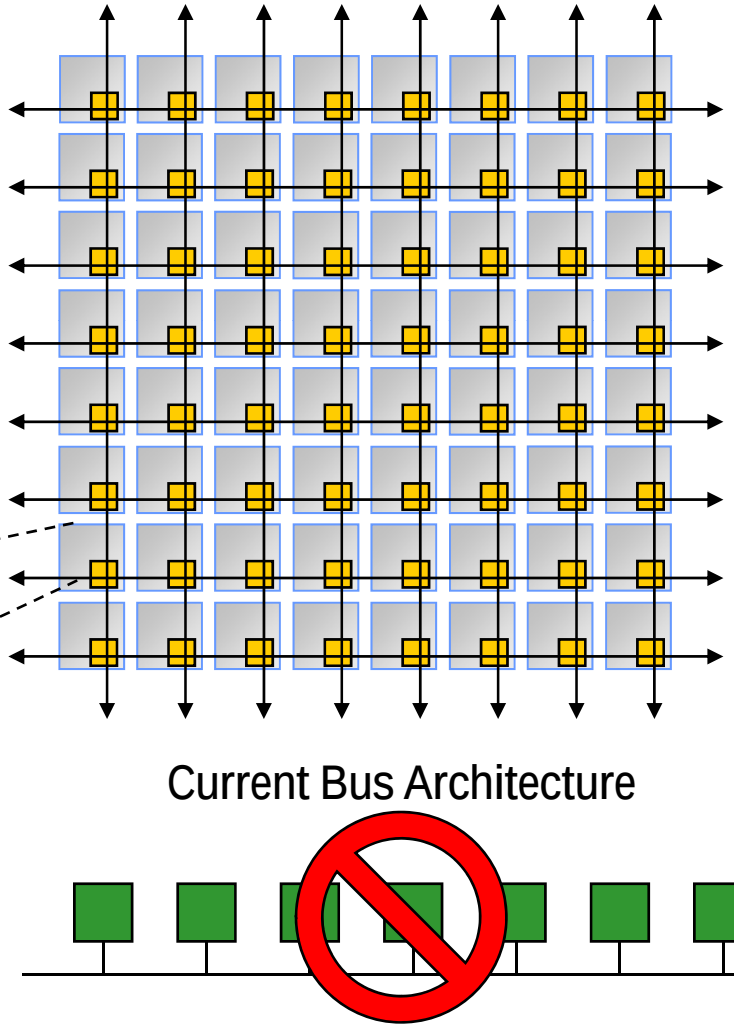
Core + Switch = Tile

Tiled Multicore Captures ASIC Benefits and is Programmable

- Scales to large numbers of cores
- Modular – design and verify 1 tile
- Power efficient
  - Short wires plus locality opts – CV^2f
  - Chandrakasan effect, more cores at lower freq and voltage – CV^2f

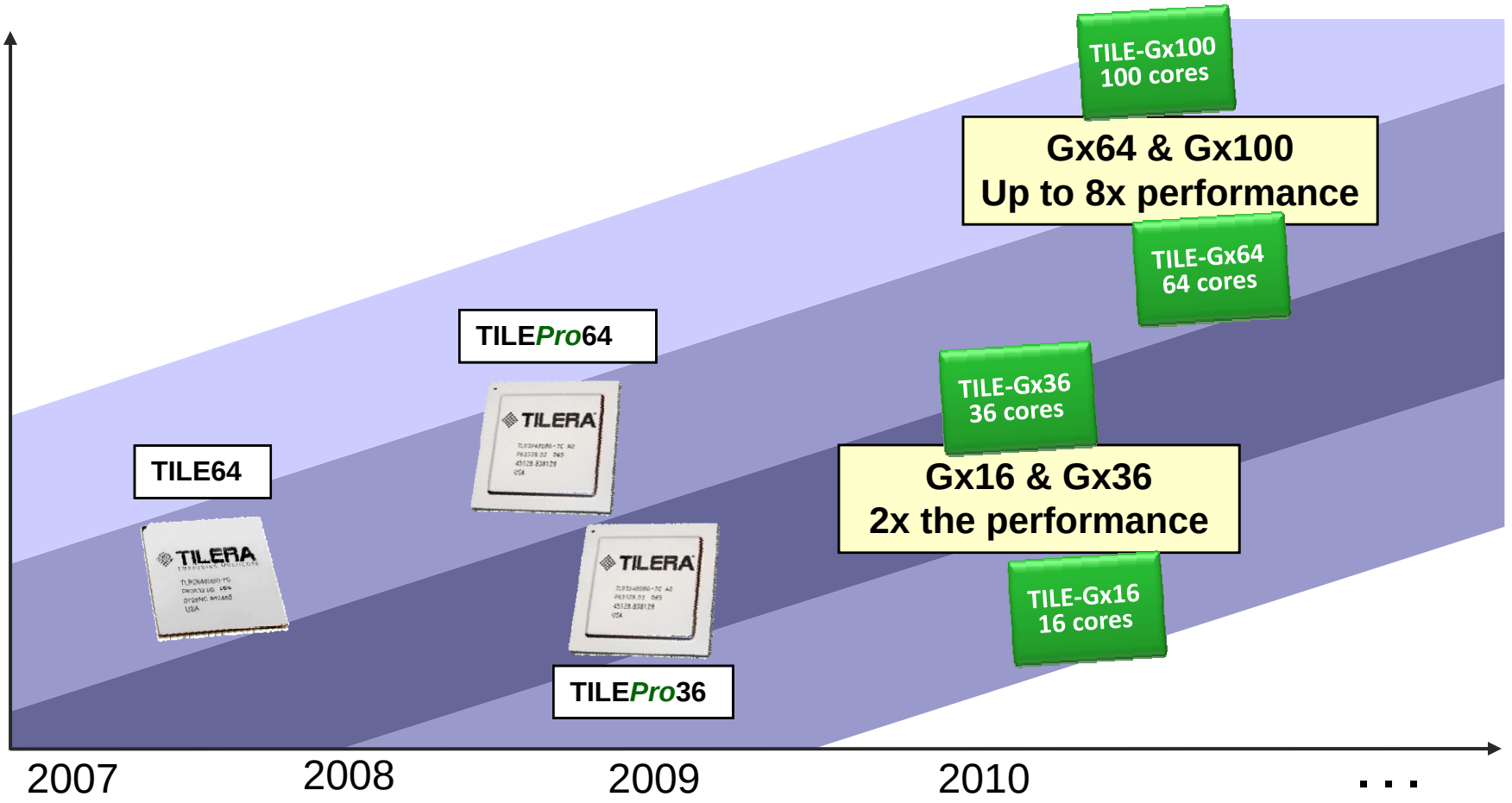


Core + Switch = Tile



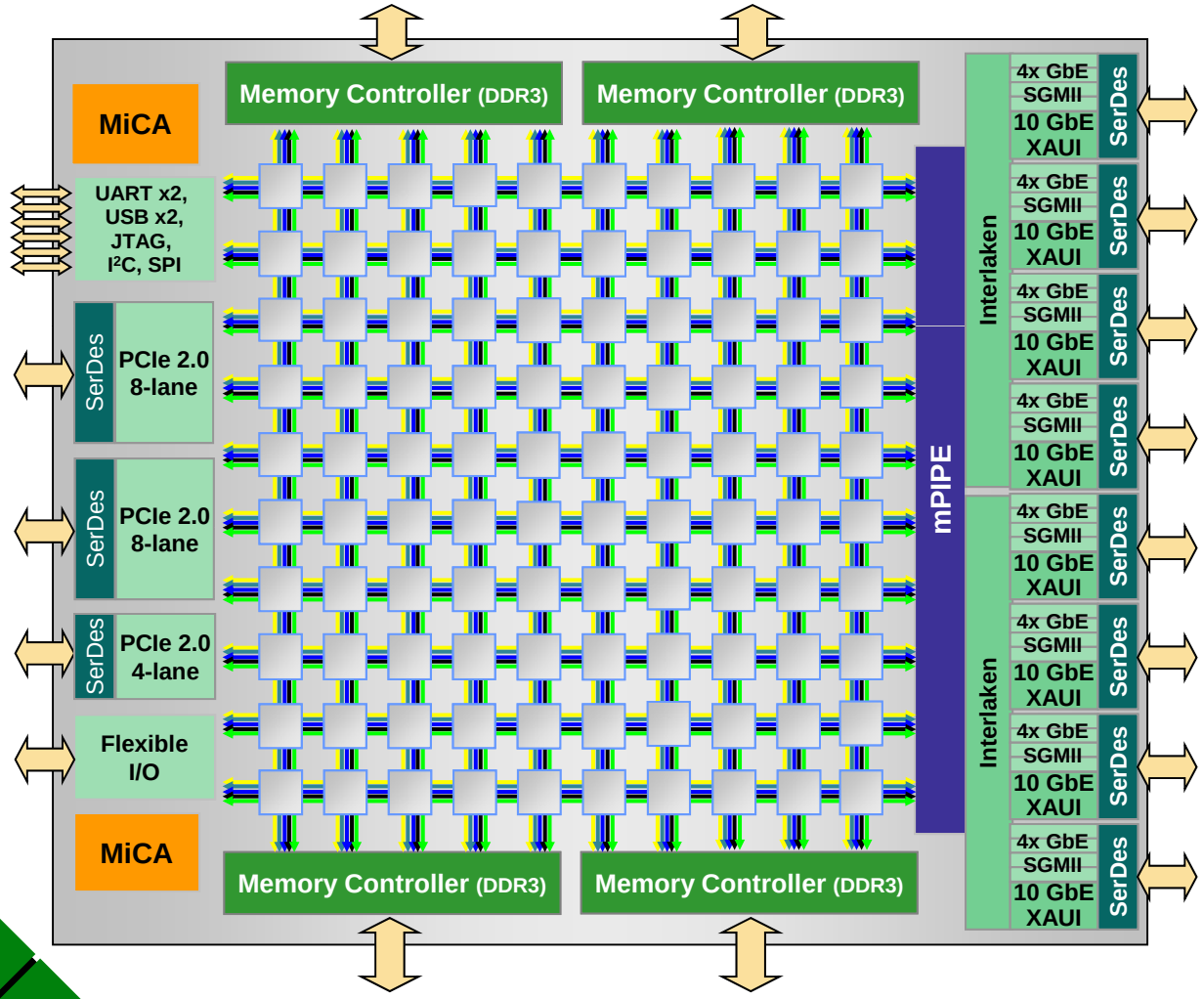
Tilera processor portfolio

Demonstrating the scale of many-core



TILE-Gx100™:

Complete System-on-a-Chip with 100 64-bit cores

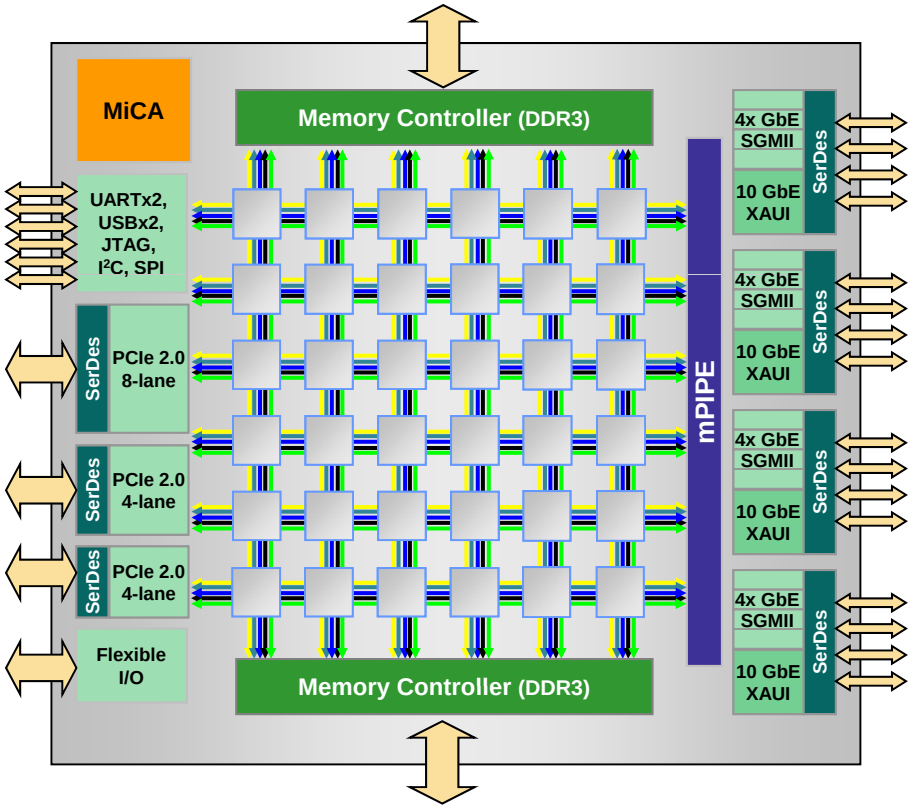


- 1.2GHz – 1.5GHz
- 32 MBytes total cache
- 546 Gbps peak mem BW
- 200 Tbps iMesh BW
- 80-120 Gbps packet I/O
  - 8 ports XAUI / 2 XAUI
  - 2 40Gb Interlaken
  - 32 ports 1GbE (SGMII)
- 80 Gbps PCIe I/O
  - 3 StreamIO ports (20Gb)
- Wire-speed packet eng.
  - 120Mpps
- MiCA engines:
  - 40 Gbps crypto
  - compress & decompress

TILERA®

TILE-Gx36™:

Scaling to a broad range of applications



- 36 Processor Cores
- 866M, 1.2GHz, 1.5GHz clk
- 12 MBytes total cache

- 40 Gbps total packet I/O
  - 4 ports 10GbE (XAUI)
  - 16 ports 1GbE (SGMII)
- 48 Gbps PCIe I/O
  - 2 16Gbps Stream IO ports
- Wire-speed packet engine
  - 60Mpps

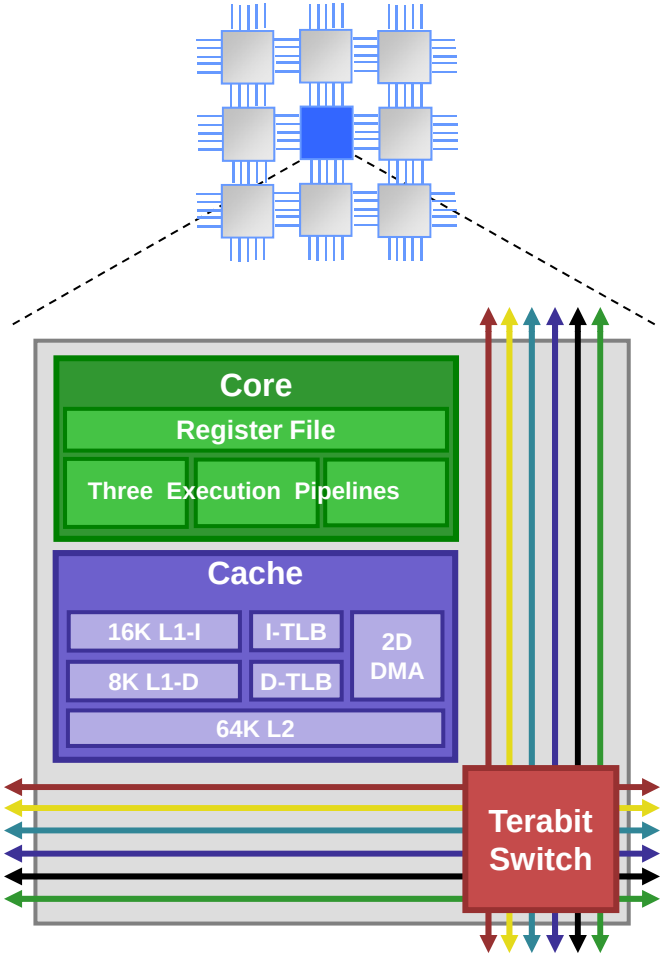
- MiCA engine:
  - 20 Gbps crypto
  - Compress & decompress

TILERA®

Full-Featured General Converged Cores

- Processor
  - Each core is a complete computer
  - 3-way VLIW CPU
  - SIMD instructions: 32, 16, and 8-bit ops
  - Instructions for video (e.g., SAD) and networking
  - Protection and interrupts
- Memory
  - L1 cache and L2 Cache
  - Virtual and physical address space
  - Instruction and data TLBs
  - Cache integrated 2D DMA engine

- **Runs SMP Linux**
- **Runs off-the-shelf C/C++ programs**
- **Signal processing and general apps**



Software must complement the hardware

Enable re-use of existing code-bases

- Standards-based development environment
  - e.g. gcc, C, C++, Java, Linux
  - Comprehensive command-line & GUI-based tools
- Support multiple OS models
  - One OS running SMP
  - Multiple virtualized OS's with protection
  - Bare metal or "zero-overhead" with background OS environment
- Support range of parallel programming styles
  - Threaded programming (pThreads, TBB)
  - Run-to-Completion with load-balancing
  - Decomposition & Pipelining
  - Higher-level frameworks (Erlang, OpenMP, Hadoop etc.)

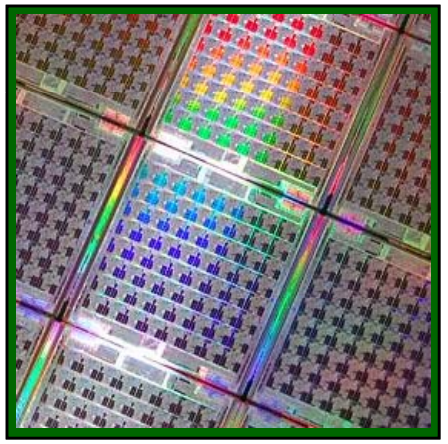
Software Roadmap

- Standards & open source integration
  - Compiler: gcc, g++ 4.4+
  - Linux:
    - Kernel: Tile architecture integrated to 2.6.36
    - User-space: glibc, broader set of standard packages
- Extended programming and runtime environments
  - Java: porting OpenJDK
  - Virtualization: porting KVM

Tile architecture:

The future of many-core computing

- Multicore is the way forward
  - But we need the right architecture to utilize it
- The Tile architecture addresses the challenges
  - Scales to 100's of cores
  - Delivers very low power
  - Runs your existing code



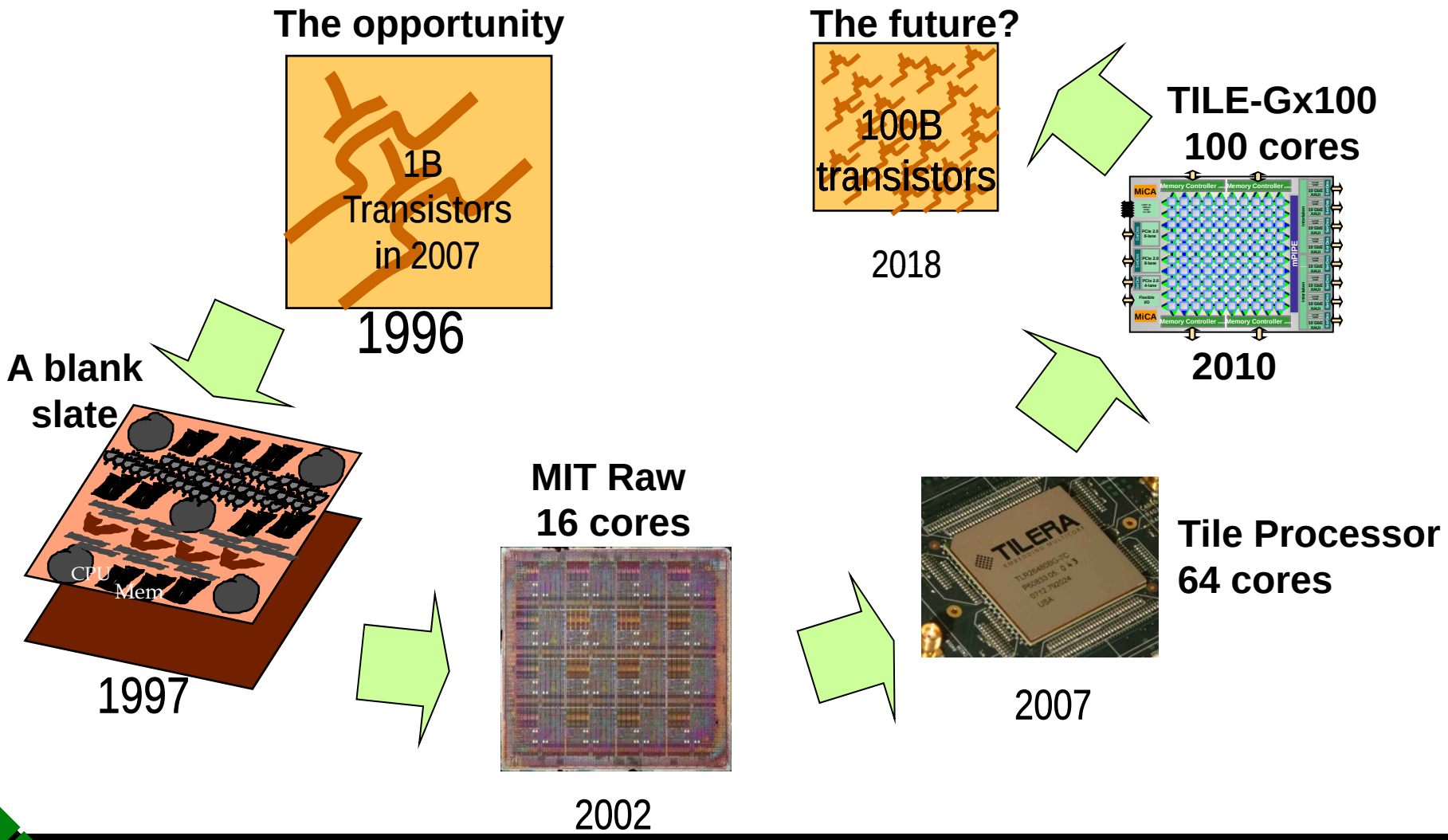
- Standards-based software
  - Familiar tools
  - Full range of standard programming environments

TILERA®

Thank you!

Questions?

Research Vision to Commercial Product



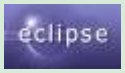
Standard tools and programming model

Multicore Development Environment

Standards-based tools

Standard programming
- SMP Linux 2.6
- ANSI C/C++
- Java, PHP


Integrated tools
- GCC compiler
- Standard gdb gprof
- Eclipse IDE



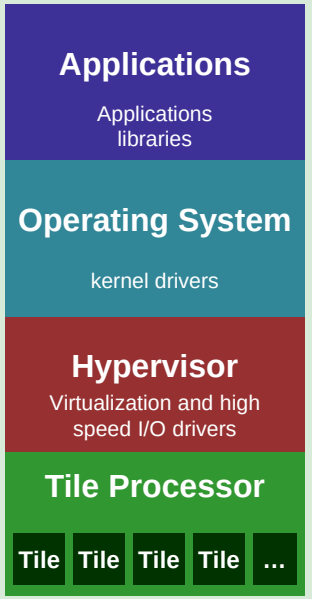
Innovative tools
- Multicore debug
- Multicore profile

Standard application stack

Application layer
- Open source apps
- Standard C/C++ libs

Operating System layer
- 64-way SMP Linux
- Zero Overhead Linux
- Bare metal environment

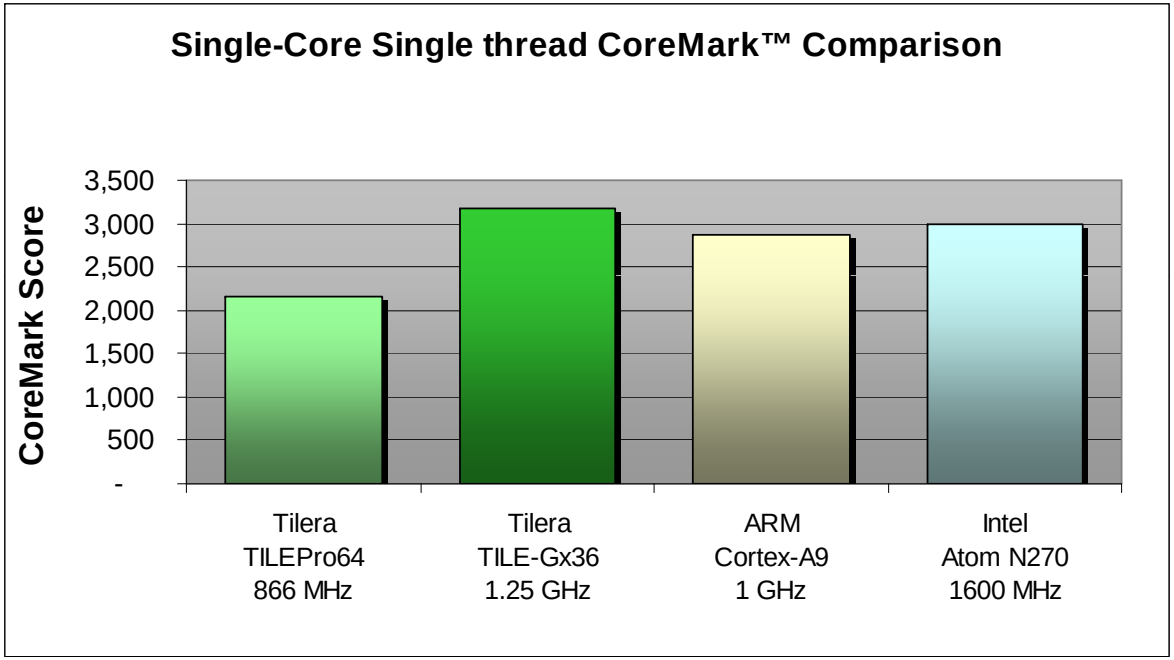
Hypervisor layer
- Virtualizes hardware
- I/O device drivers



Standard Software Stack

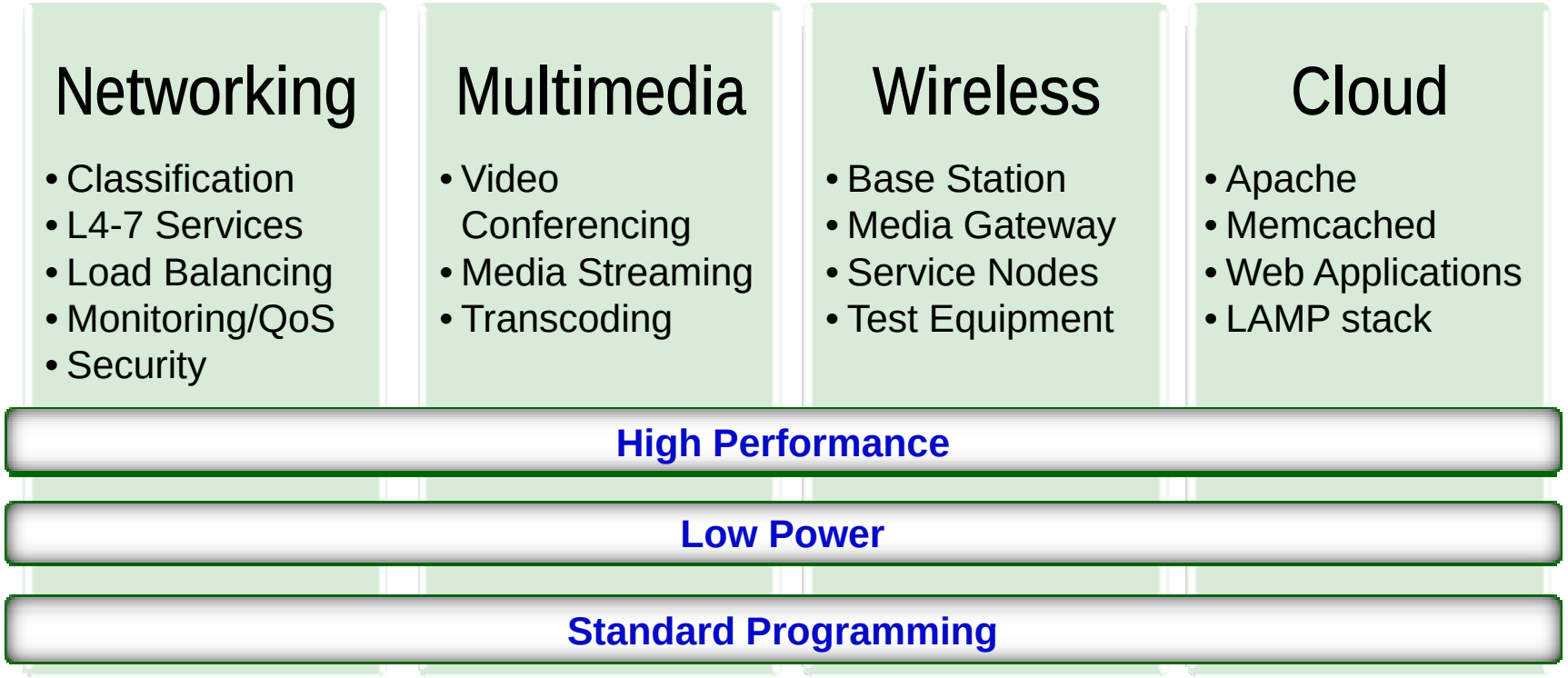
Management Protocols	Nagios	Ganglia	IPMI 2.0	MRTG	NET-SNMP
Infrastructure Apps	LIGHTTPD fly light.	hadoop, APACHE	Transcoding	MySQL, MEMCACHED	Network Monitoring
Language Support	php	python	Ruby A Programmer's Best Friend	Perl	Java
Compiler, OS Hypervisor	GNU gcc & g++	KVM			Commercial Linux Distribution

High single core performance
Comparable to Atom & ARM Cortex-A9 cores



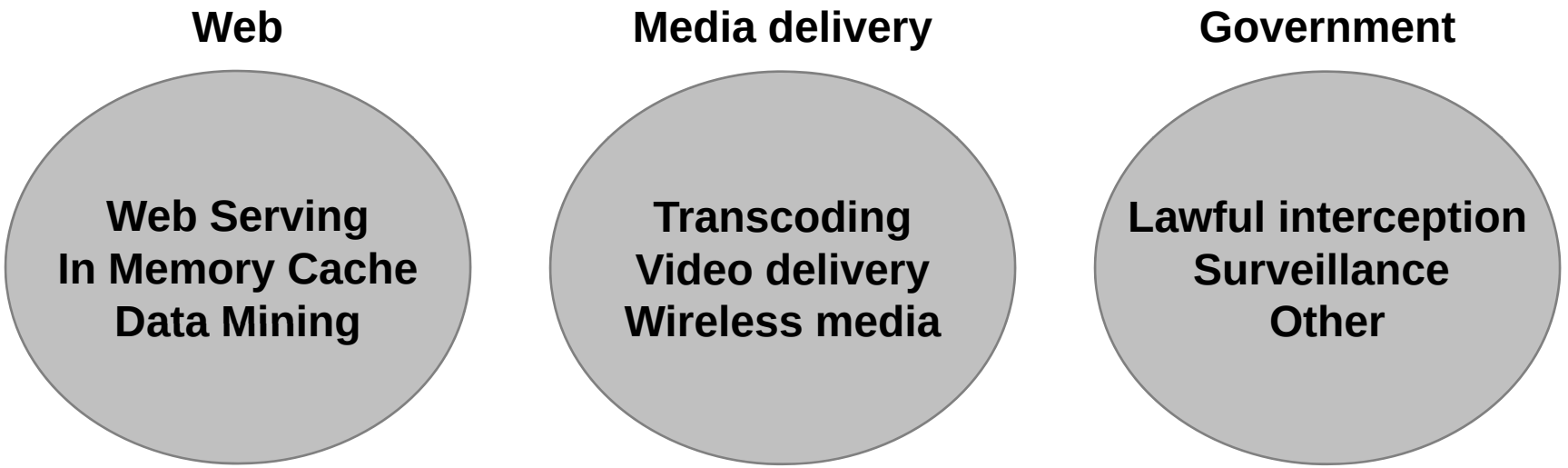
- Data for TILEPro, ARM Cortex-A9, Atom N270 is available on the CoreMark website http://coremark.org/home.php
- TILE-Gx and single thread Atom results were measured in Tilera labs
- Single core, single thread result for ARM is calculated based on chip scores

Significant value across multiple markets



Over 100 customers
Over 40 customers going into production
Tier 1 customers in all target markets

Targeting markets with highly parallel applications



Common Themes

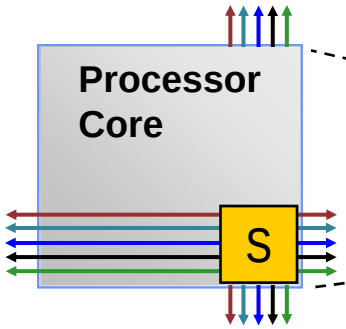
Hundreds and Thousands of servers running each application

thousands of parallel transactions

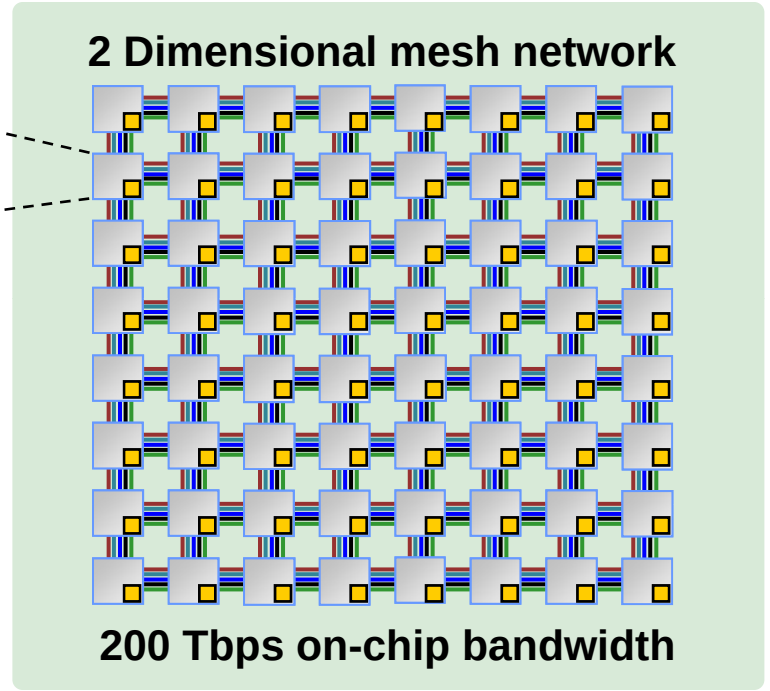
All need better performance and power efficiency

The Tile Processor Architecture

Mesh interconnect, power-optimized cores

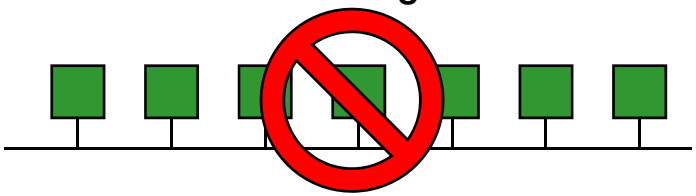


Core + Switch = Tile



- Scales to large numbers of cores
- Modular: Design-and-verify 1 tile
- Power efficient:
  - Short wires & locality optimize CV^2f
  - Chandrakasan effect, more cores at lower freq and voltage – CV^2f

Traditional Bus/Ring Architecture

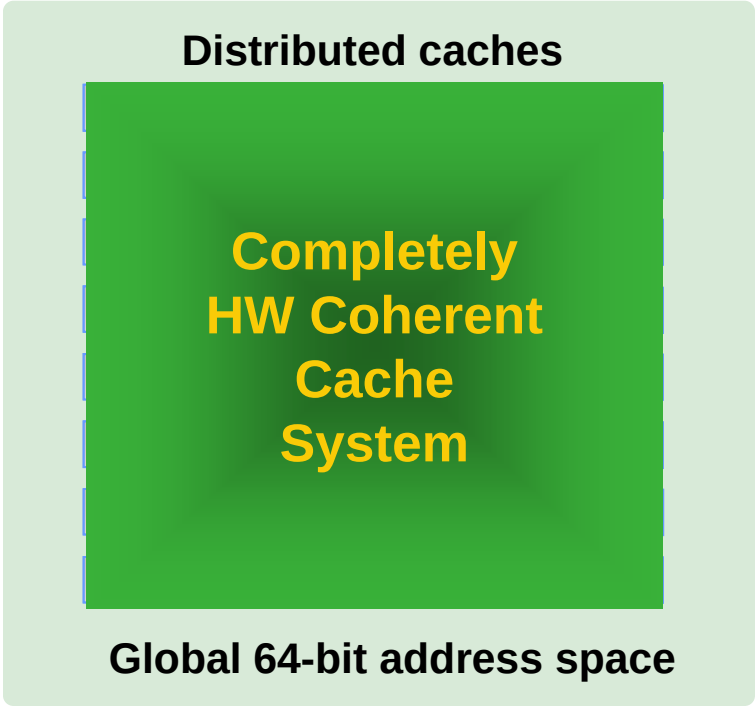


TILERA

Distributed “everything”

Cache, memory management, connectivity

- Big centralized caches don’t scale
  - Contention
  - Long latency
  - High power
- Distributed caches have numerous benefits
  - Lower power (less logic lit-up per access)
  - Exploit locality (local L1 & L2)
  - Can exploit various cache placement mechanisms to enhance performance



Highest compute density

- 2U form factor
- 4 hot pluggable modules
- 8 Tilera TILEPro processors
- 512 general purpose cores
- 1.3 trillion operations /sec

Power efficient and eco-friendly server

- 10,000 cores in a 8 Kilowatt rack
- 35-50 watts max per node
- Server power of 400 watts
- 90%+ efficient power supplies
- Shared fans and power supplies

Coherent distributed cache system

- **Globally Shared Physical Address Space**
  - Full Hardware Cache Coherence
  - Standard shared memory programming model
- **Distributed cache**
  - Each tile has local L1 and L2 caches
  - Aggregate of L2 serves as a globally shared L3
  - Any cache block can be replicated locally
  - Hardware tracks sharers, invalidates stale copies
- **Dynamic Distributed Cache (DDC™)**
  - Memory pages distributed across all cores or homed by allocating core
- **Coherent I/O**
  - Hardware maintains coherence
  - I/O reads/writes coherent with tile caches
  - Reads/writes delivered by HW to home cache
  - Header/packet delivered directly to tile caches