
The Other Face of On-Chip Interconnect

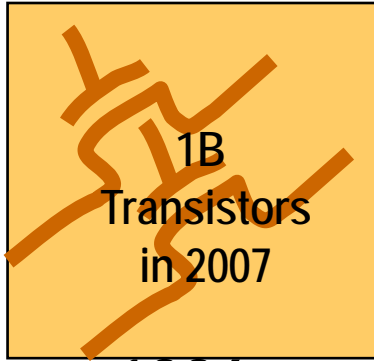
Anant Agarwal
MIT CSAIL

Throughput
Latency
Wires
Routing



Programming ease
Energy efficiency
Scalar transport
Protection
Streaming
Demultiplexing
Sender occupancy

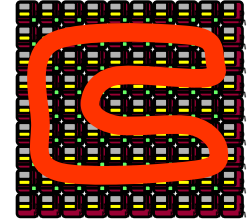
Stages of Reality



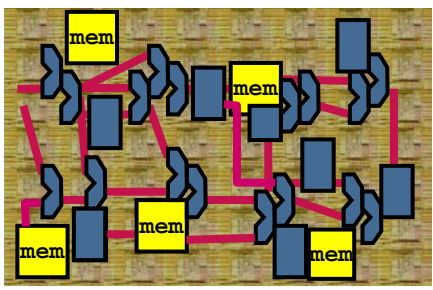
1996



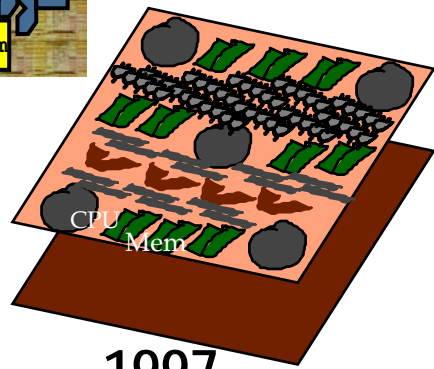
2018



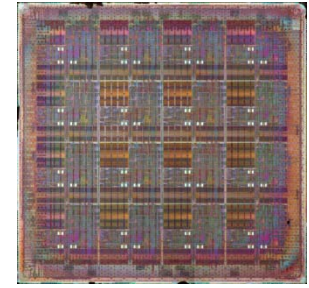
2014



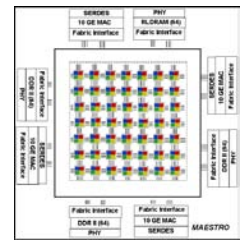
1996



1997



2002



2009



2007

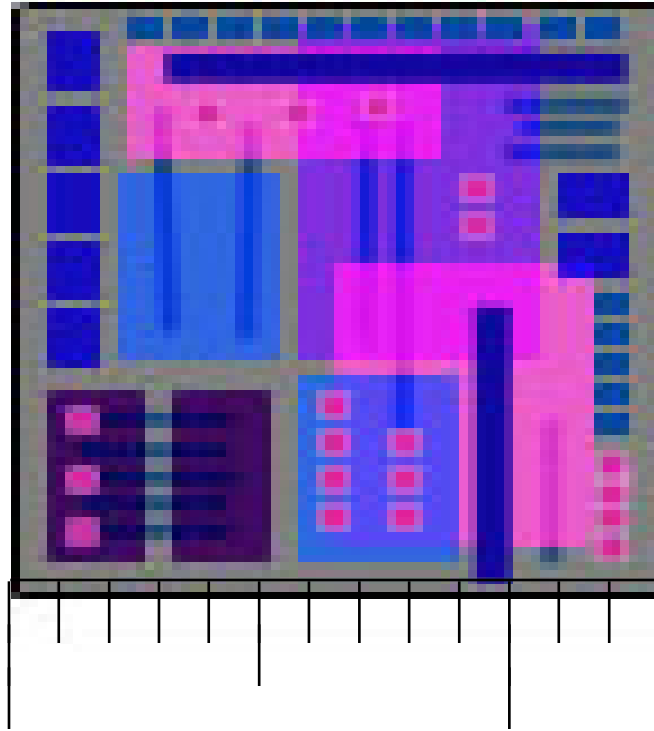
Virtual reality

Prototype reality
Product reality
Virtual Reality

The Opportunity

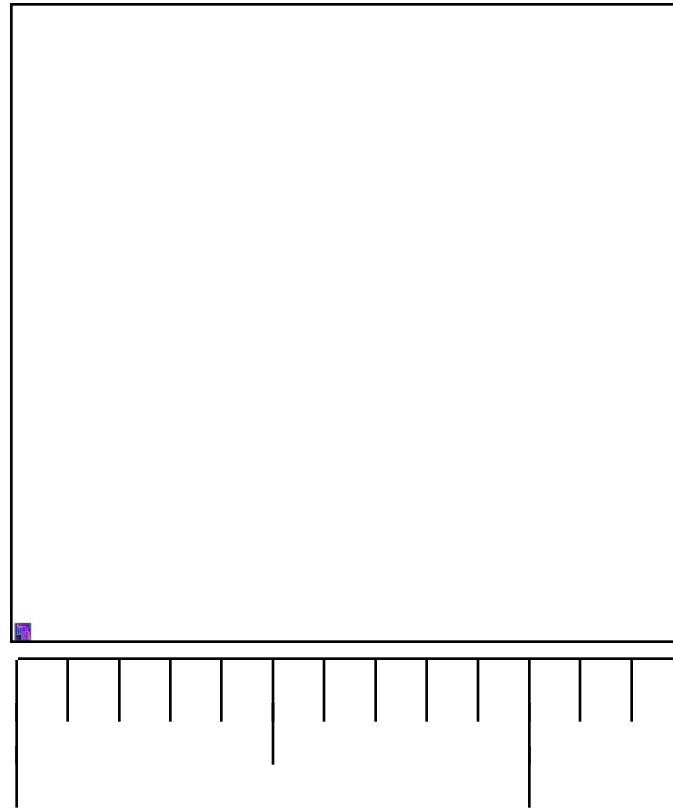
1996...

20MIPS cpu
in 1987



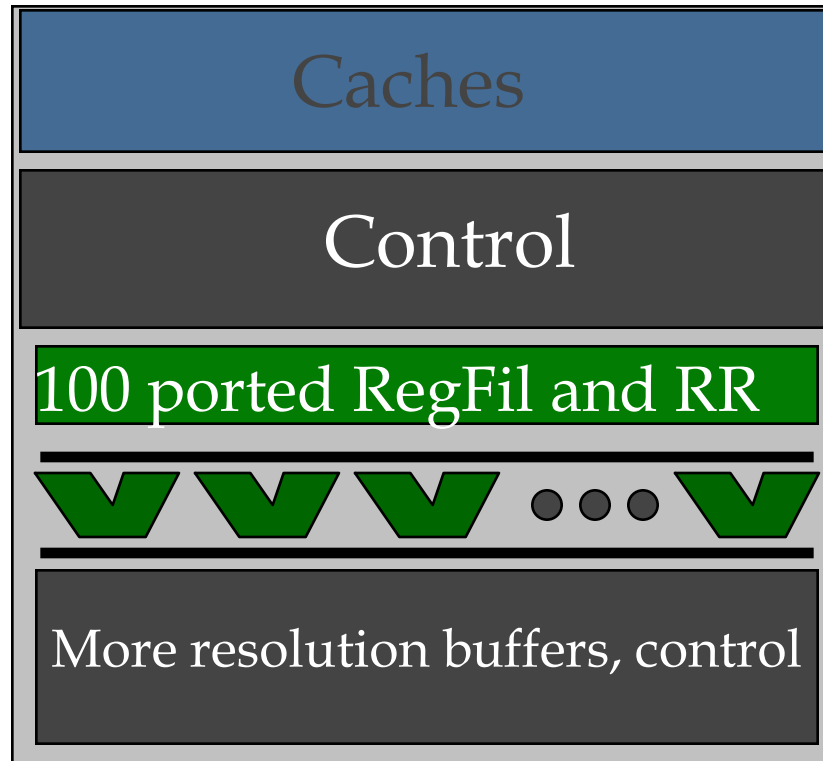
Few thousand gates

The Opportunity



The billion transistor chip of 2007

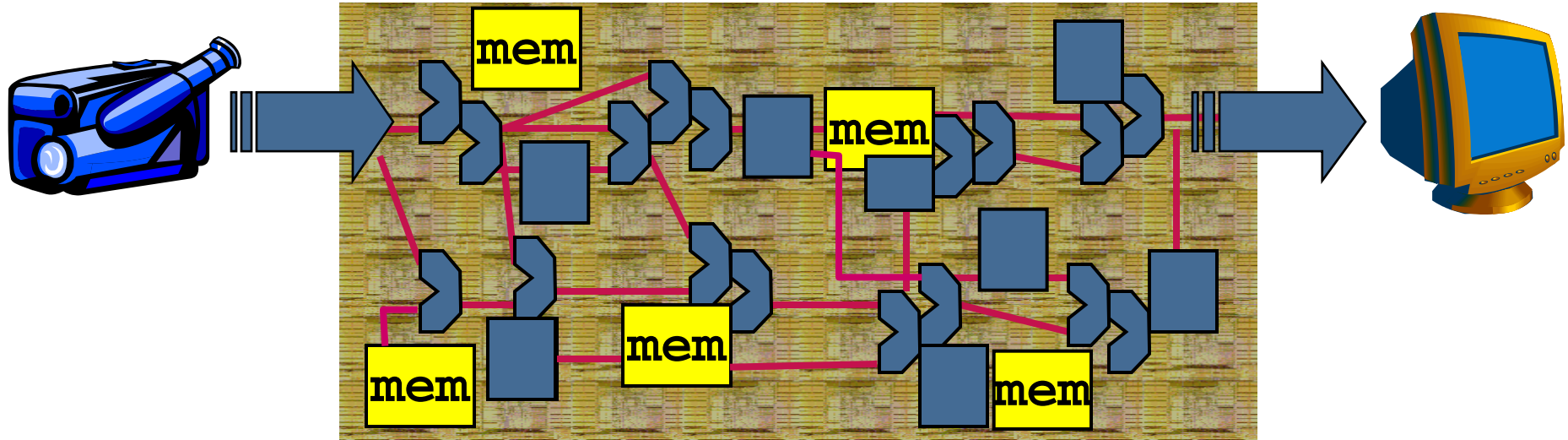
How to Fritter Away Opportunity



the x1786?
does not scale

“1/10 ns”

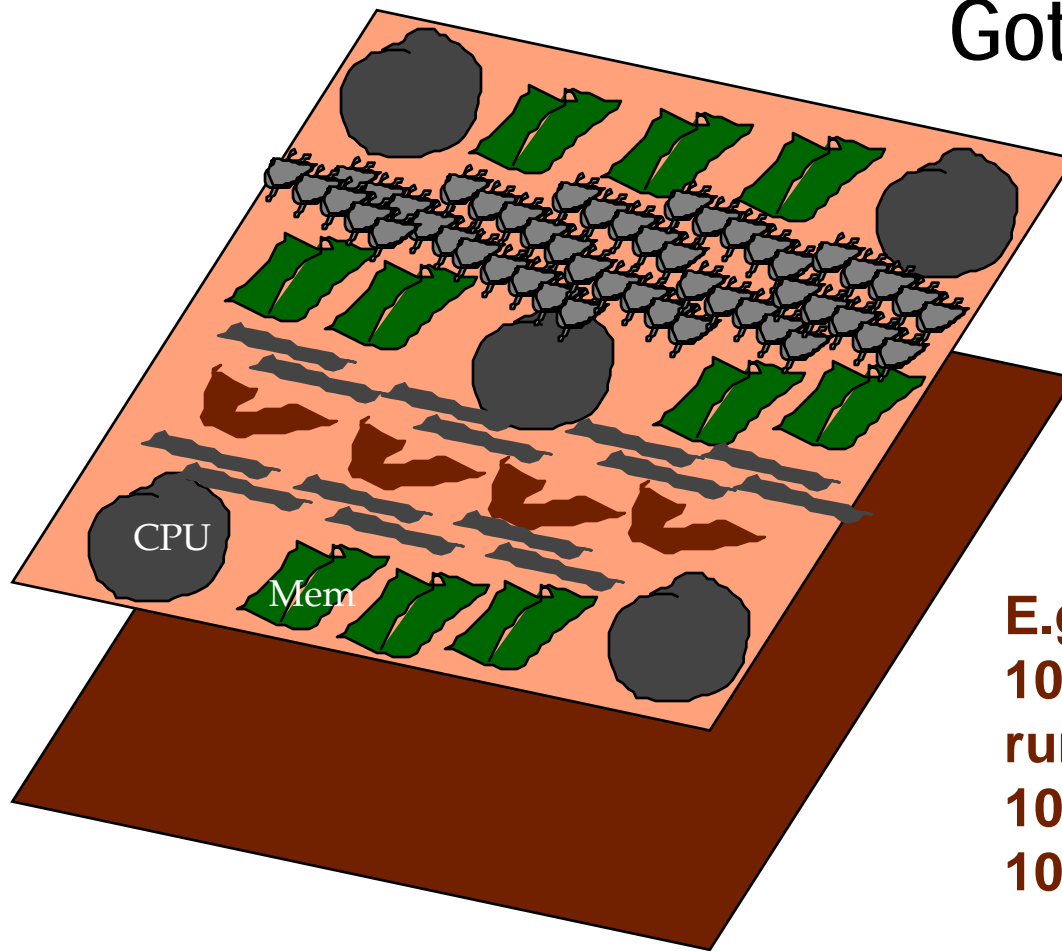
Take Inspiration from ASICs



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

*Fast, low power, area efficient
But not programmable*

Our Early Raw Proposal



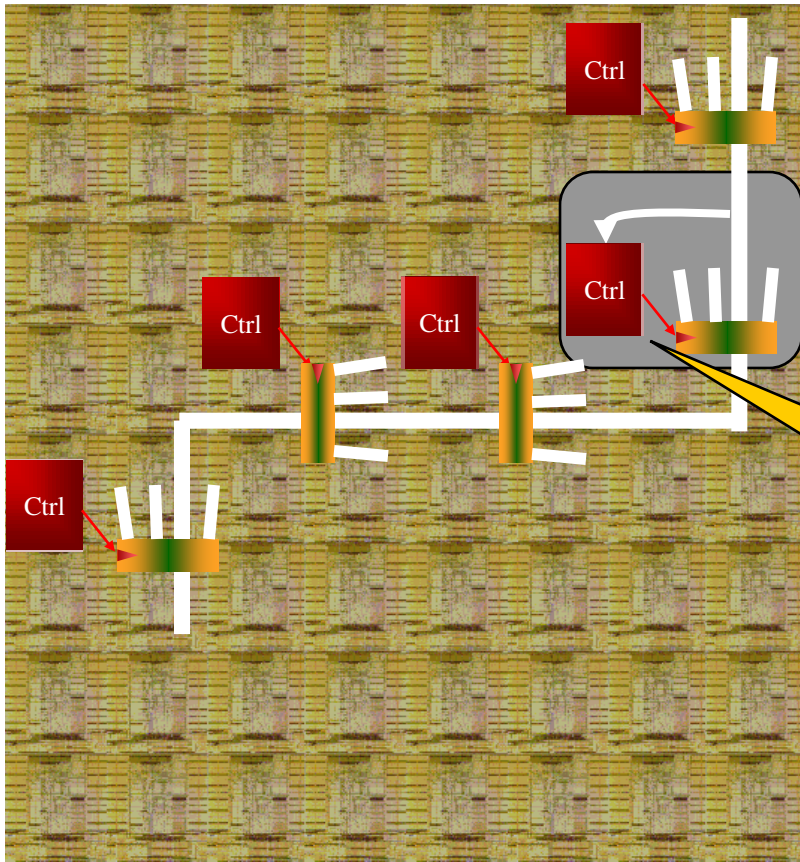
Got parallelism?

E.g.,
100-way unrolled loop,
running on 100 ALUs,
1000 regs,
100 memory banks

But how to build programmable, yet custom, wires?

A digital wire

Pipeline it! Multiplex it!



Uh! What were we smoking!

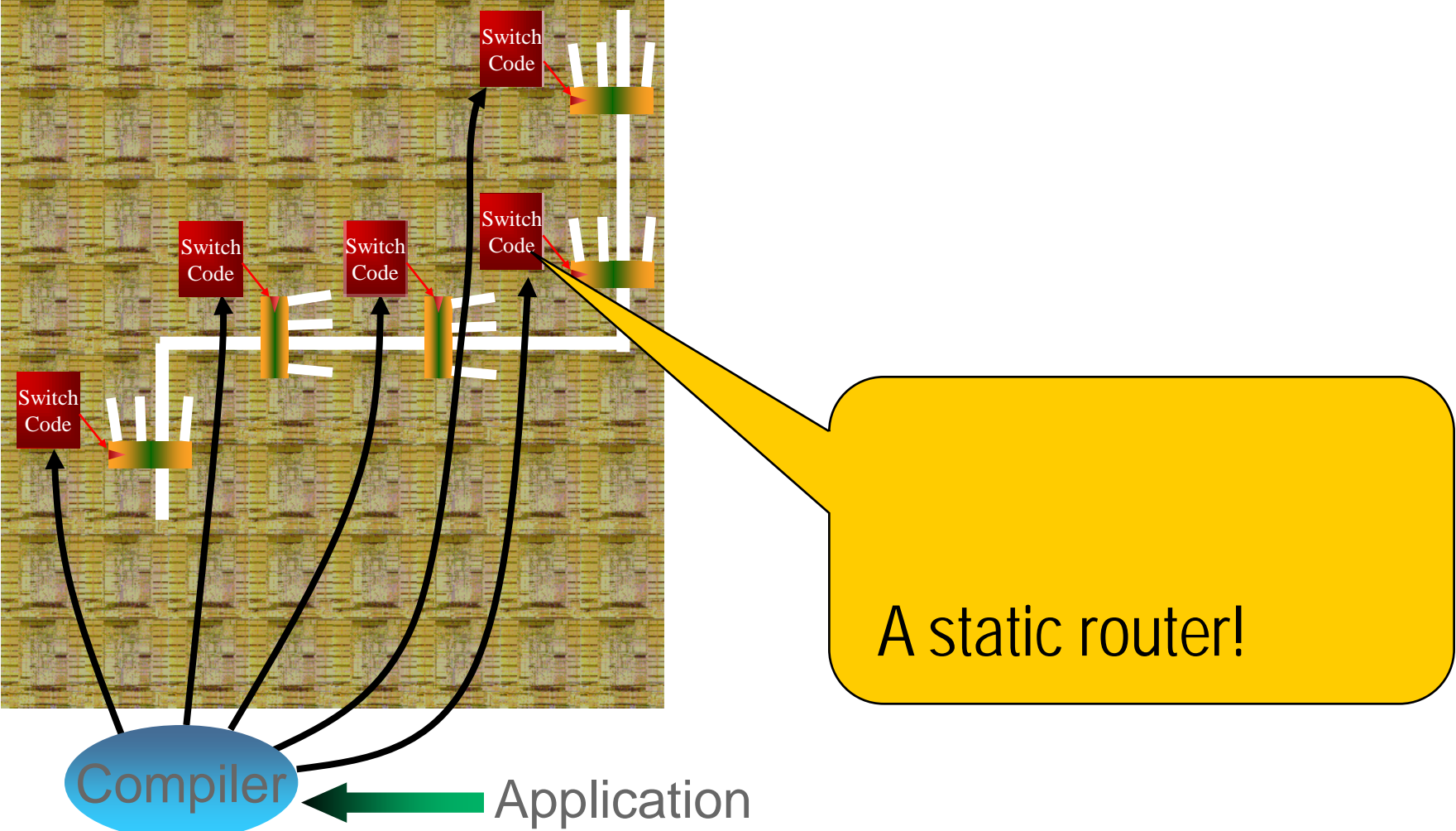
rate it!

- Fast clock (10GHz in 2010)
- Improve utilization
- Customize to application and optimize utilization

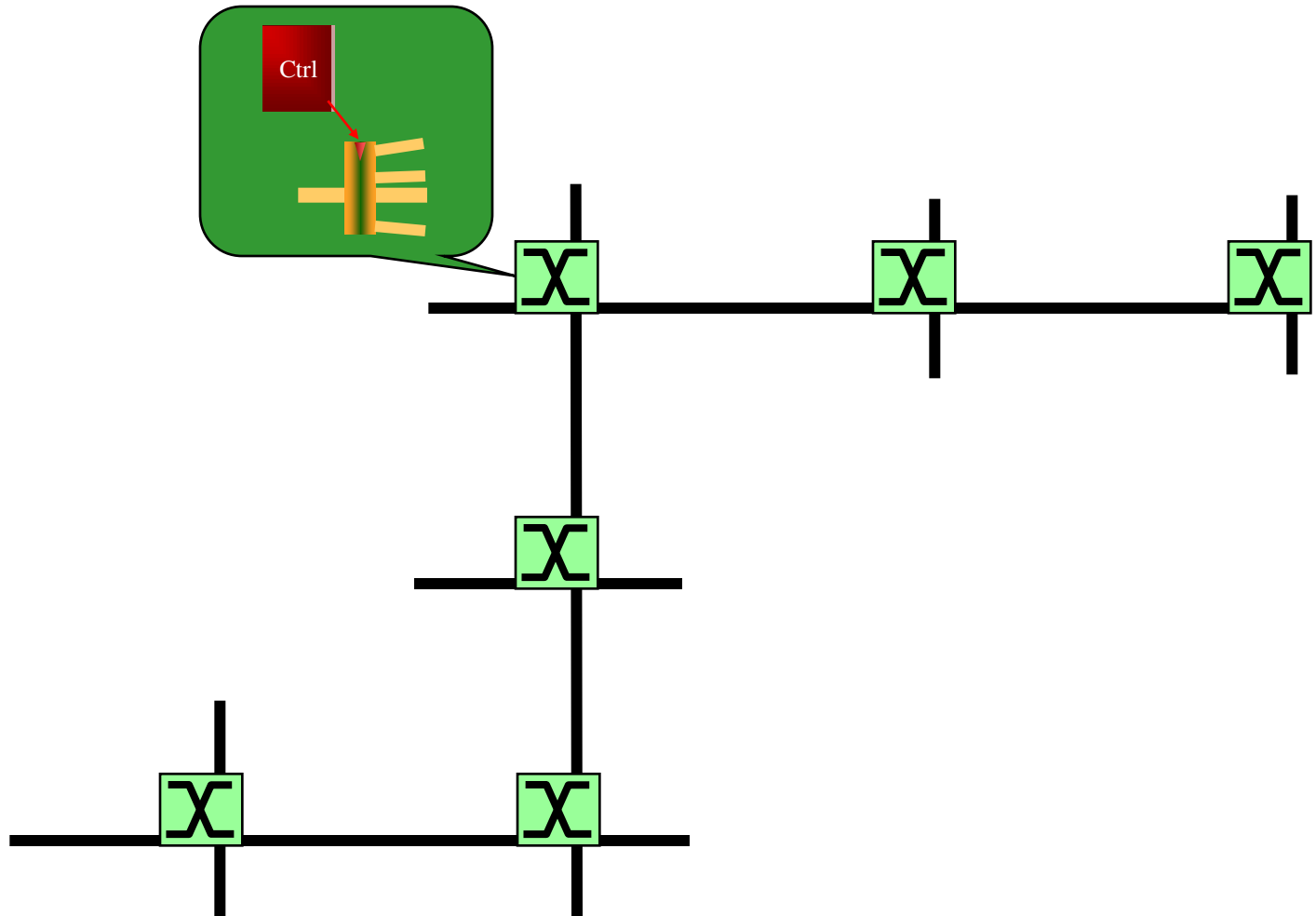
A dynamic router!

Replace custom wires with routed on-chip networks

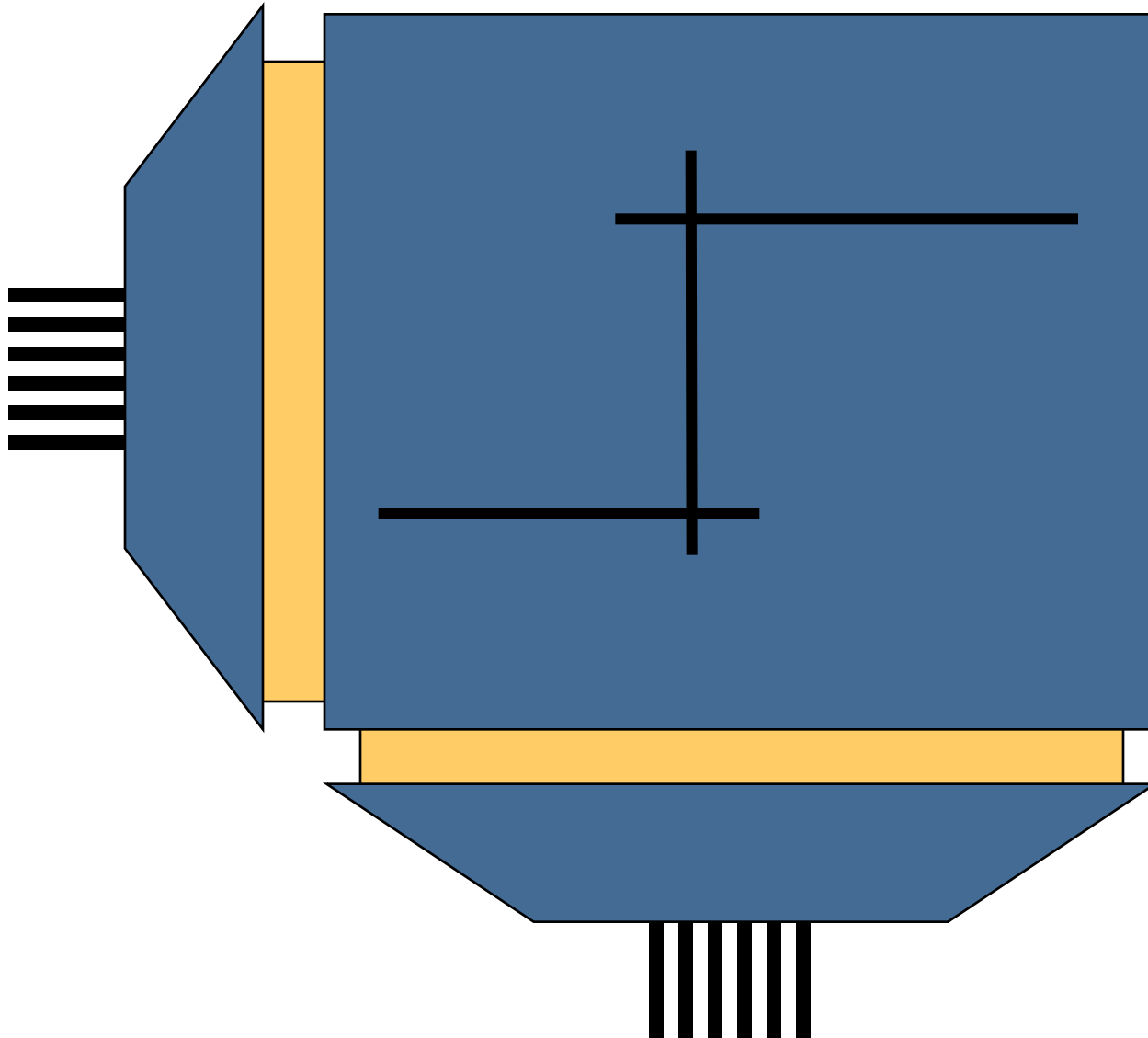
Static Router



Replace Wires with Routed Networks

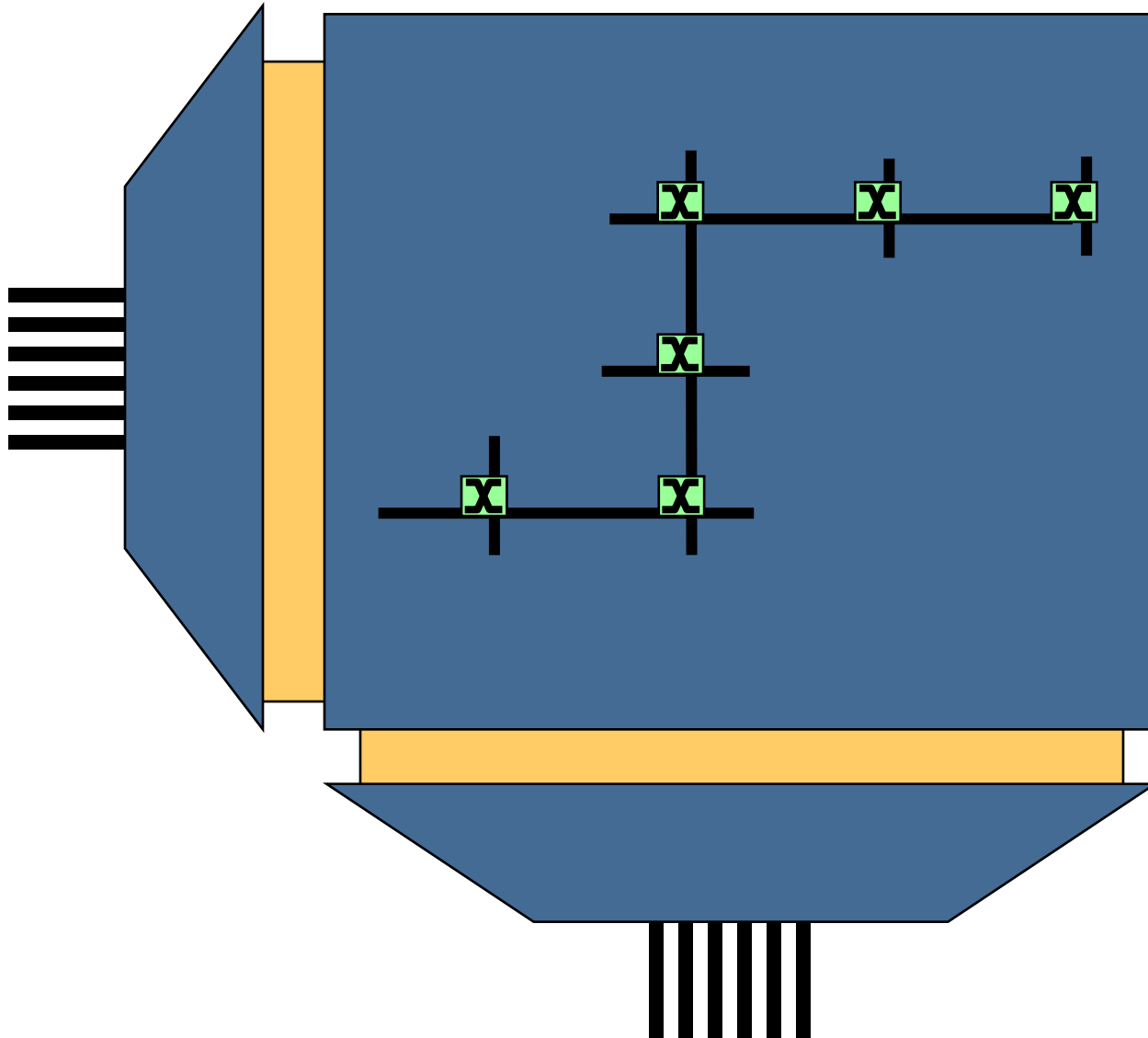


50-Ported Register File → Distributed Registers



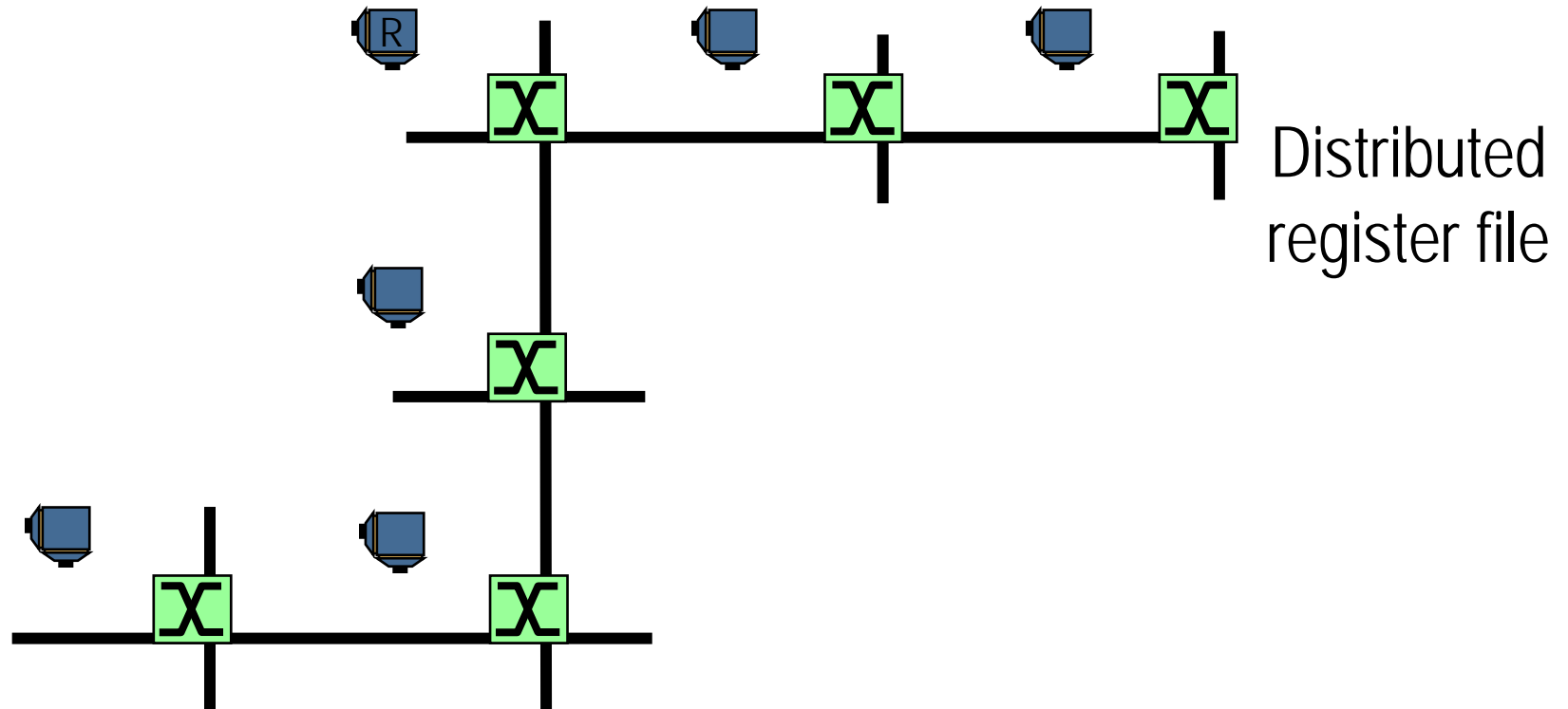
Gigantic
50
ported
register
file

50-Ported Register File → Distributed Registers



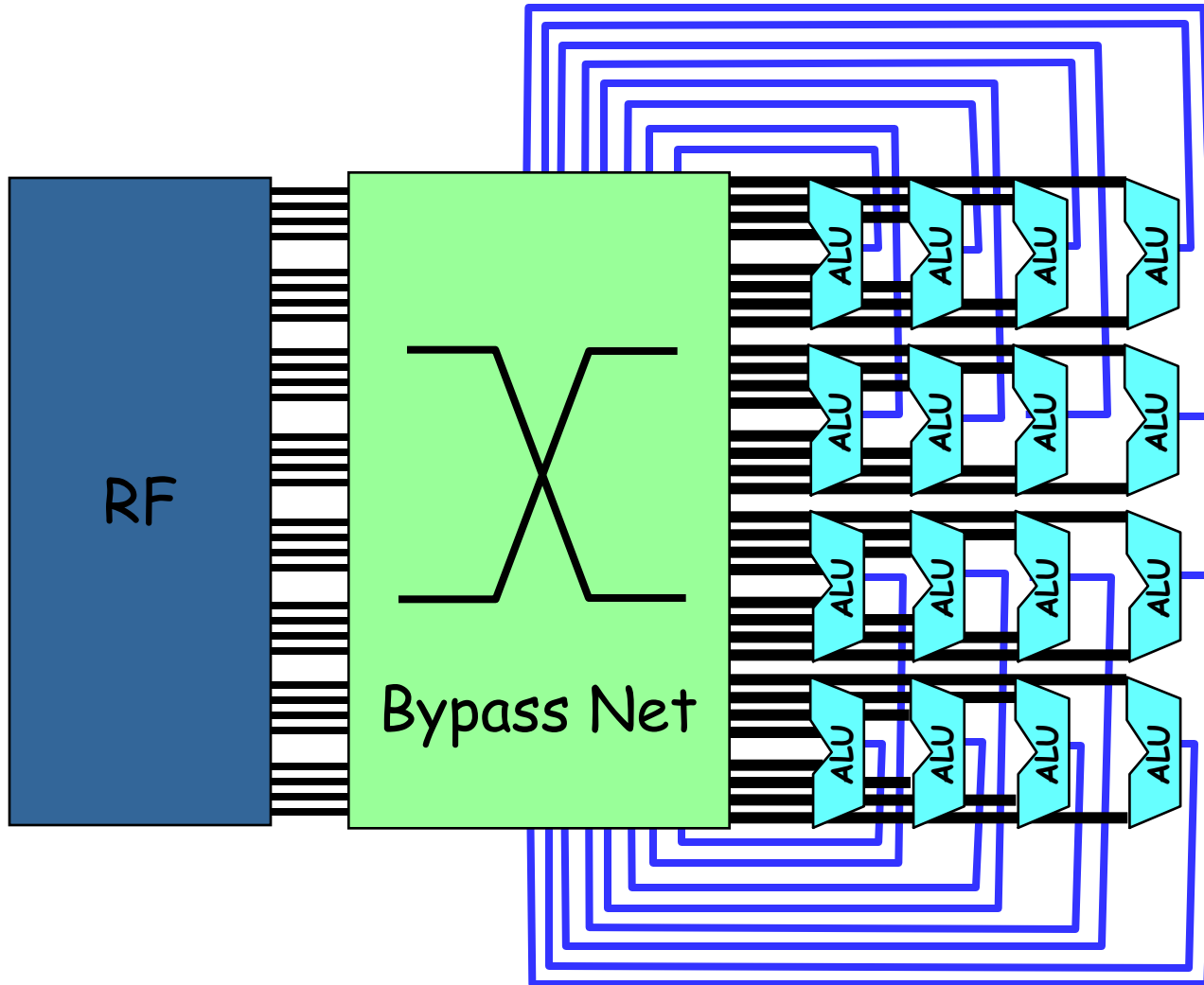
Gigantic
50
ported
register
file

Distributed Registers + Routed Network

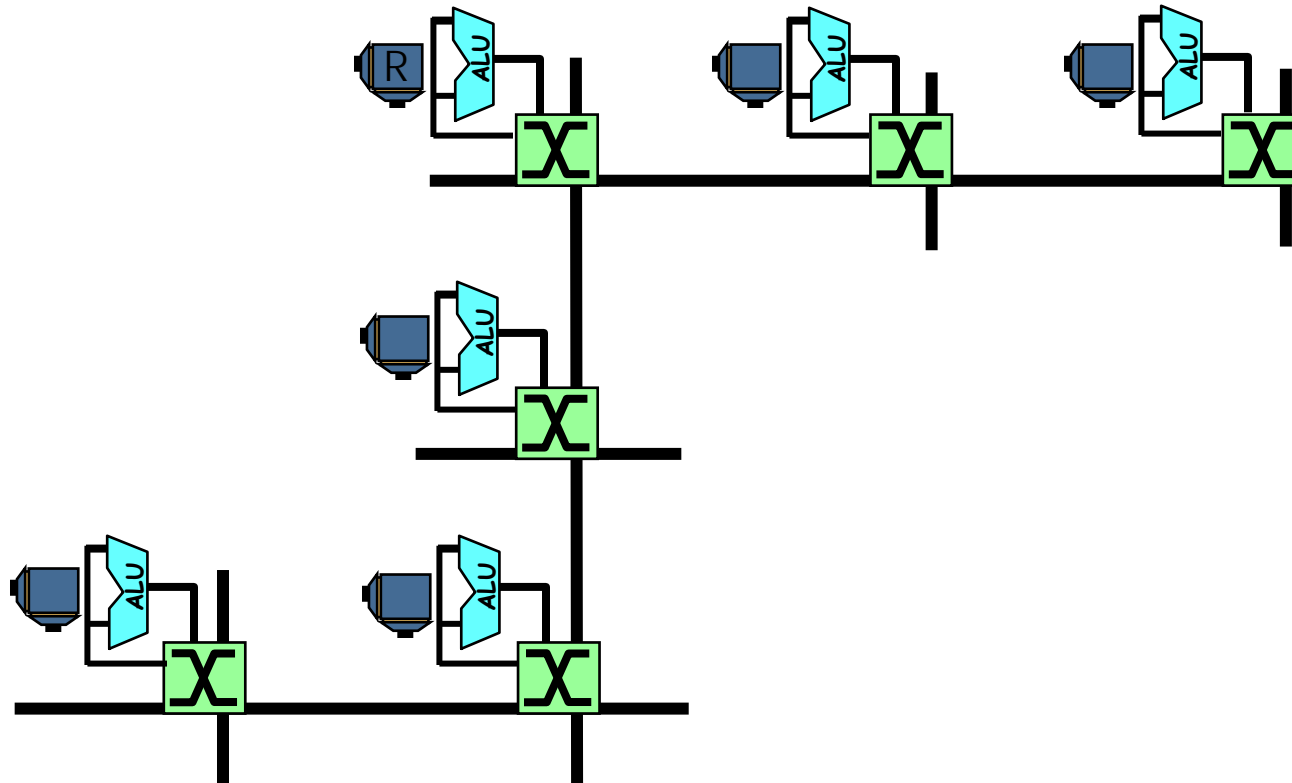


Called NURA [ASPLOS 1998]

16-Way ALU Clump → Distributed ALUs

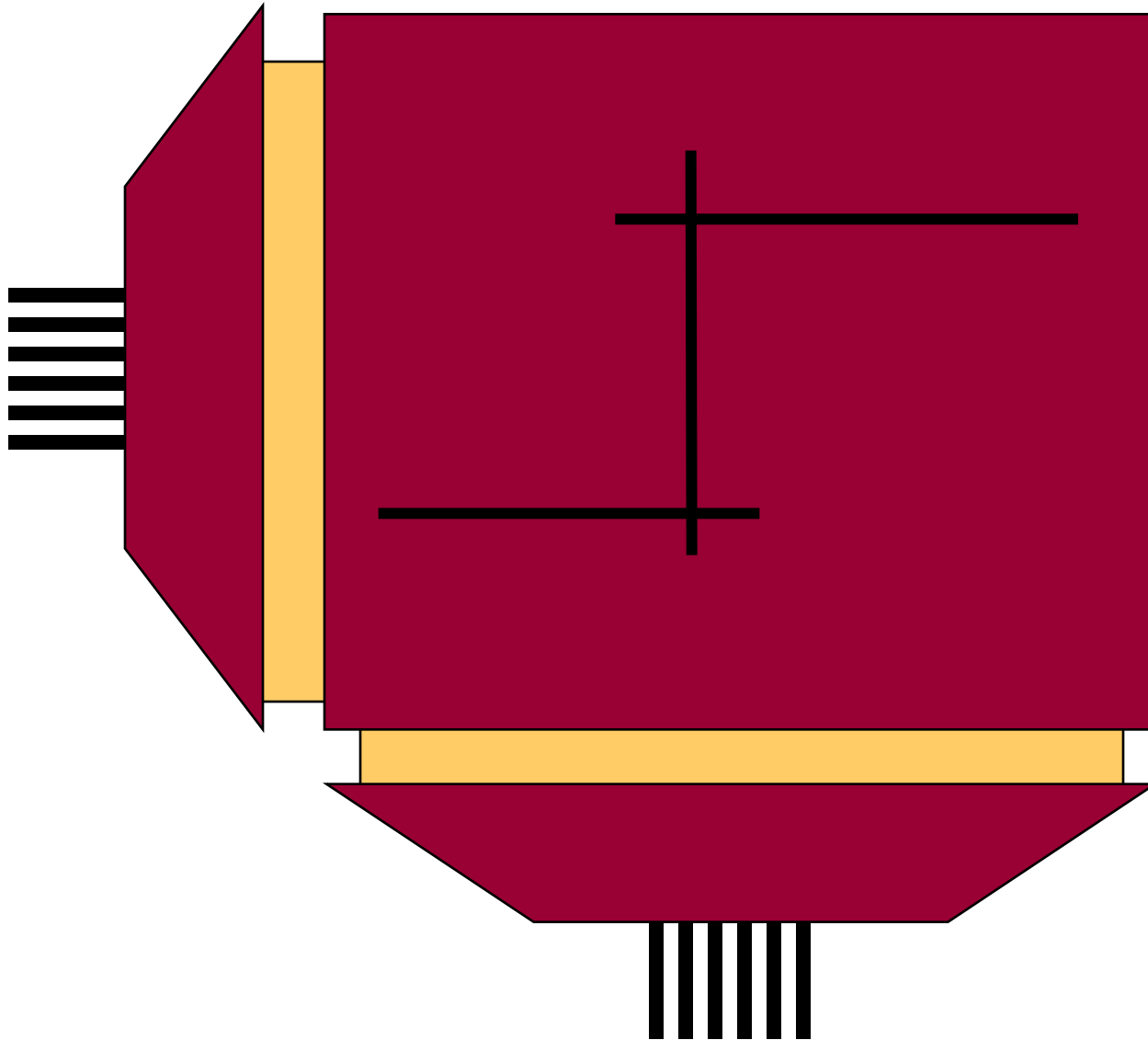


Distributed ALUs, Routed Bypass Network



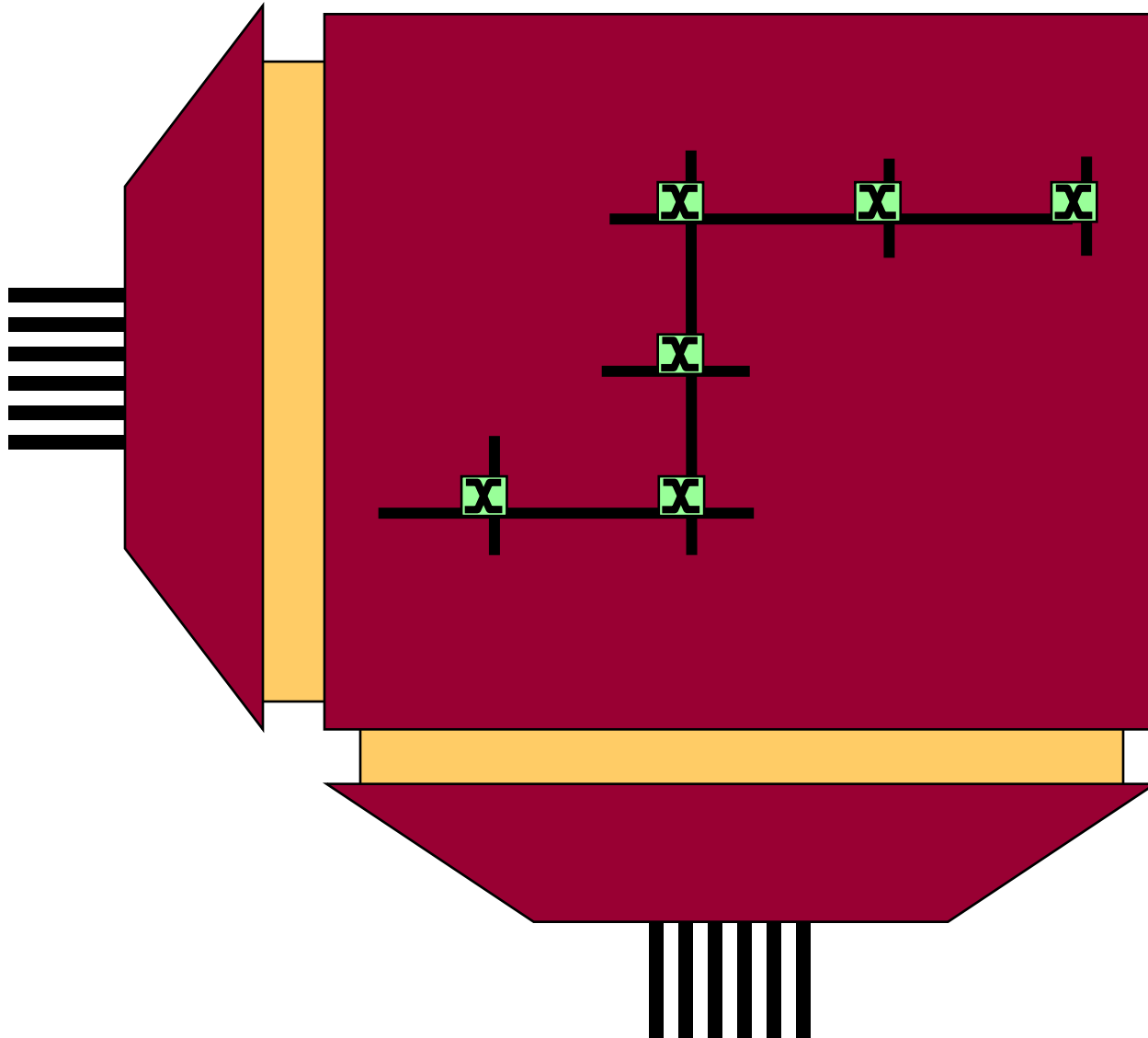
Scalar Operand Network (SON) [TPDS 2005]

Mongo Cache → Distributed Cache

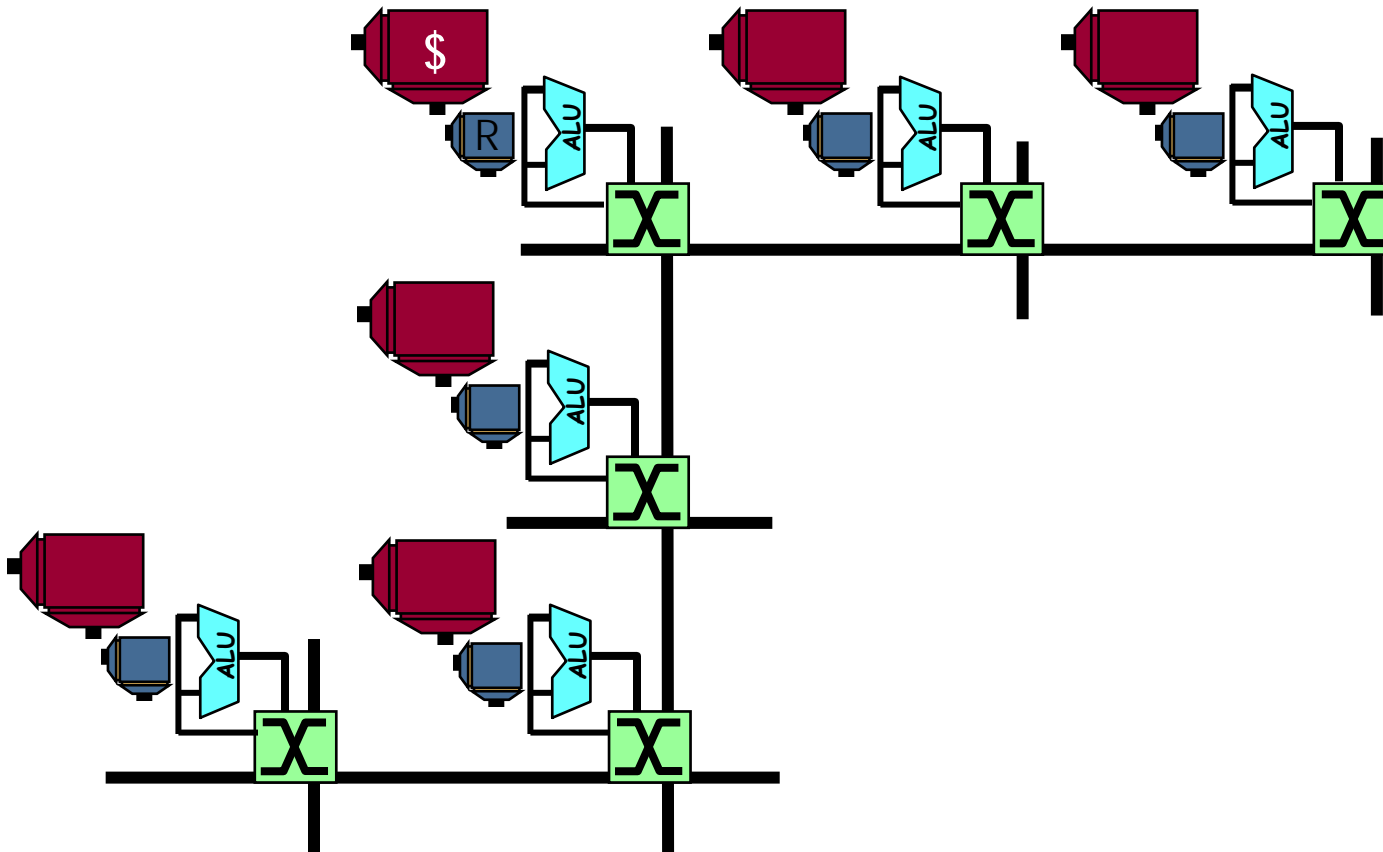


Gigantic
10
ported
cache

Distributing the Cache



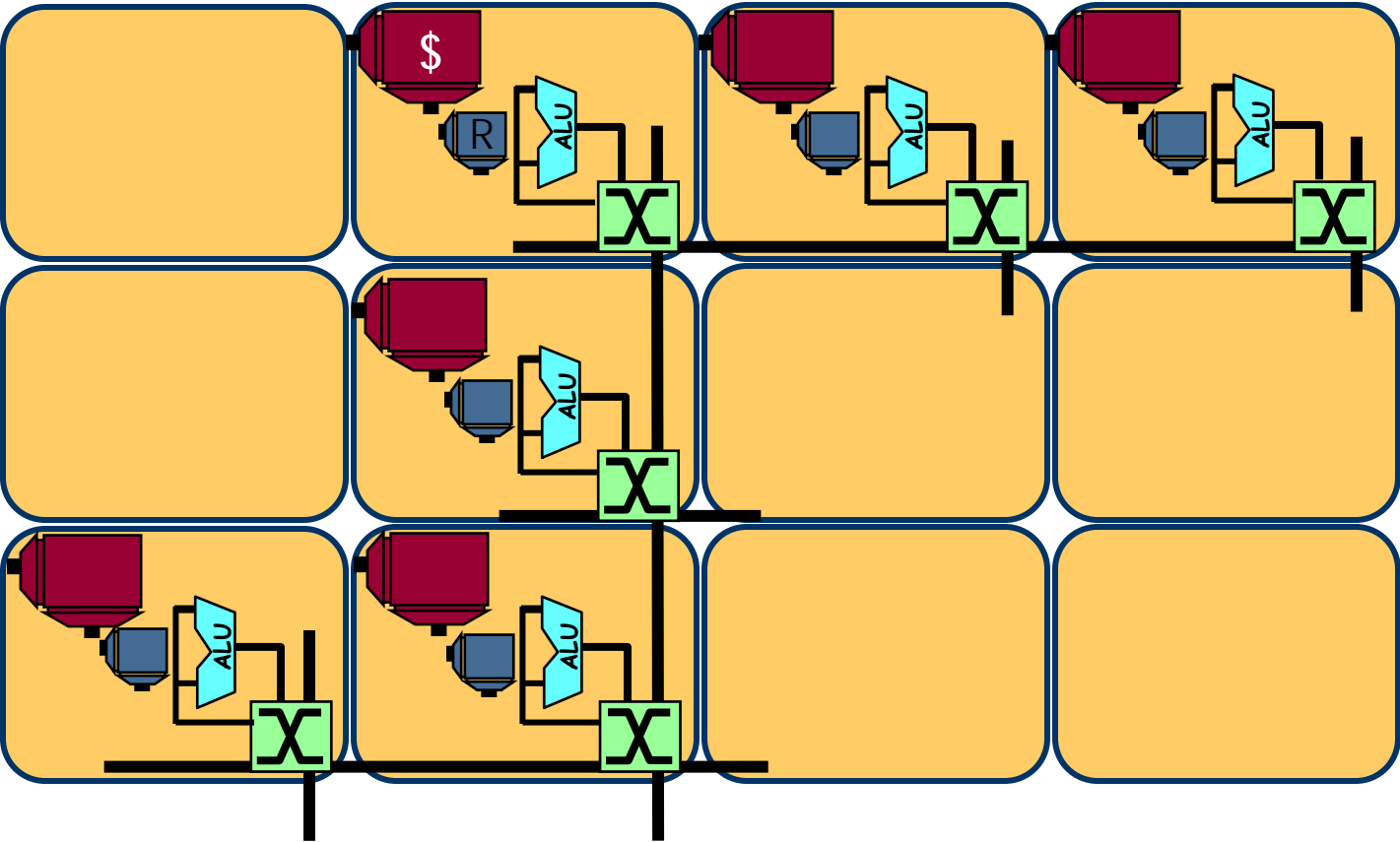
Distributed Shared Cache



Like DSM (distributed shared memory), cache is distributed;
But, unlike NUCA, caches are local to processors, not far away

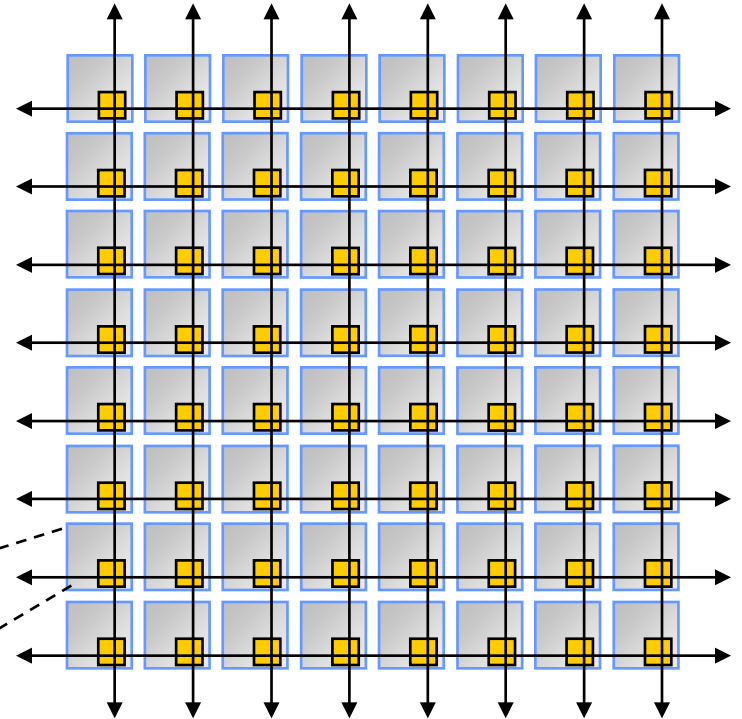
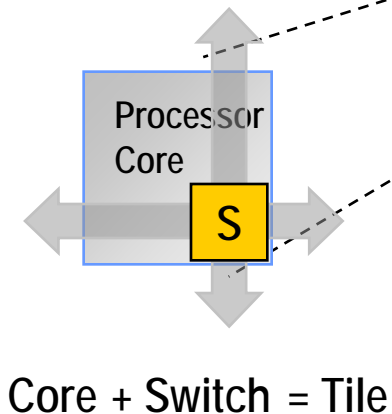
[ISCA 1999]

Tiled Multicore Architecture



Tiled Multicore

- Scales to large numbers of cores
- Modular – design and verify one tile
- Power efficient
 - Short wires CV^2f
 - Chandrakasan effect CV^2f
 - Dynamic and compiler scheduled routing



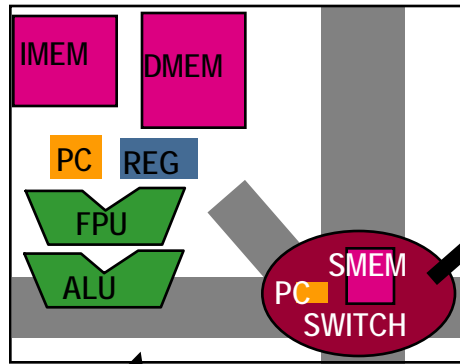
Besides the usual metrics of latency and bandwidth, energy efficiency has become an important metric of interconnect goodness

A Prototype Tiled Architecture: The Raw Microprocessor

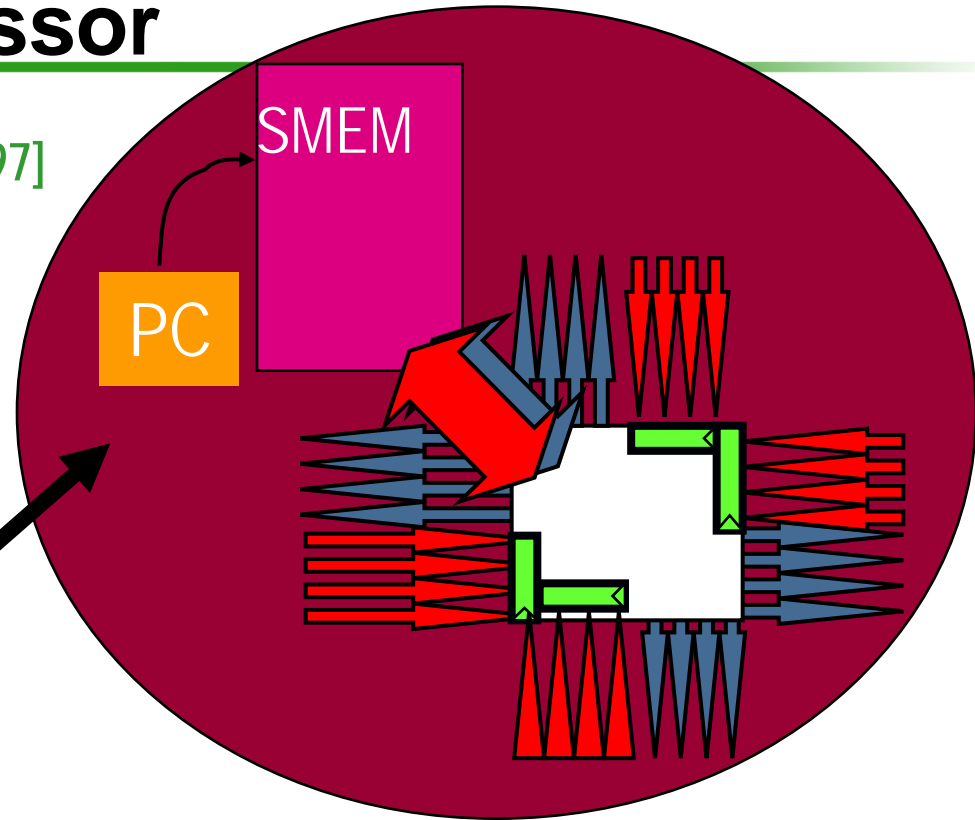
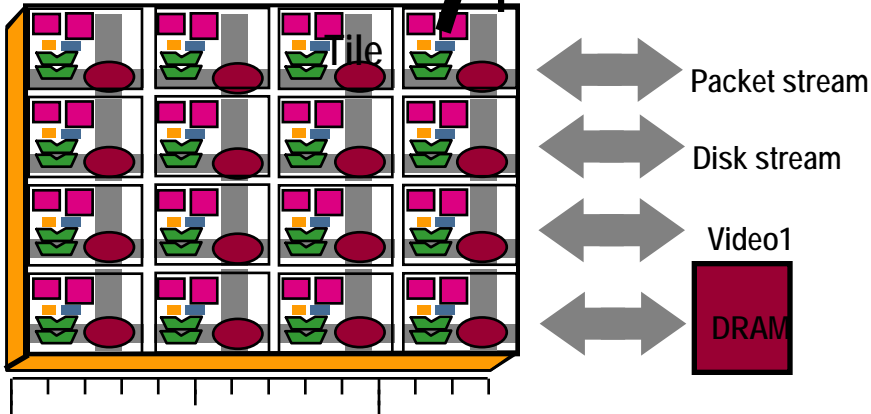
Raw Switch

[Billion transistor IEEE Computer Issue '97]

A Raw Tile



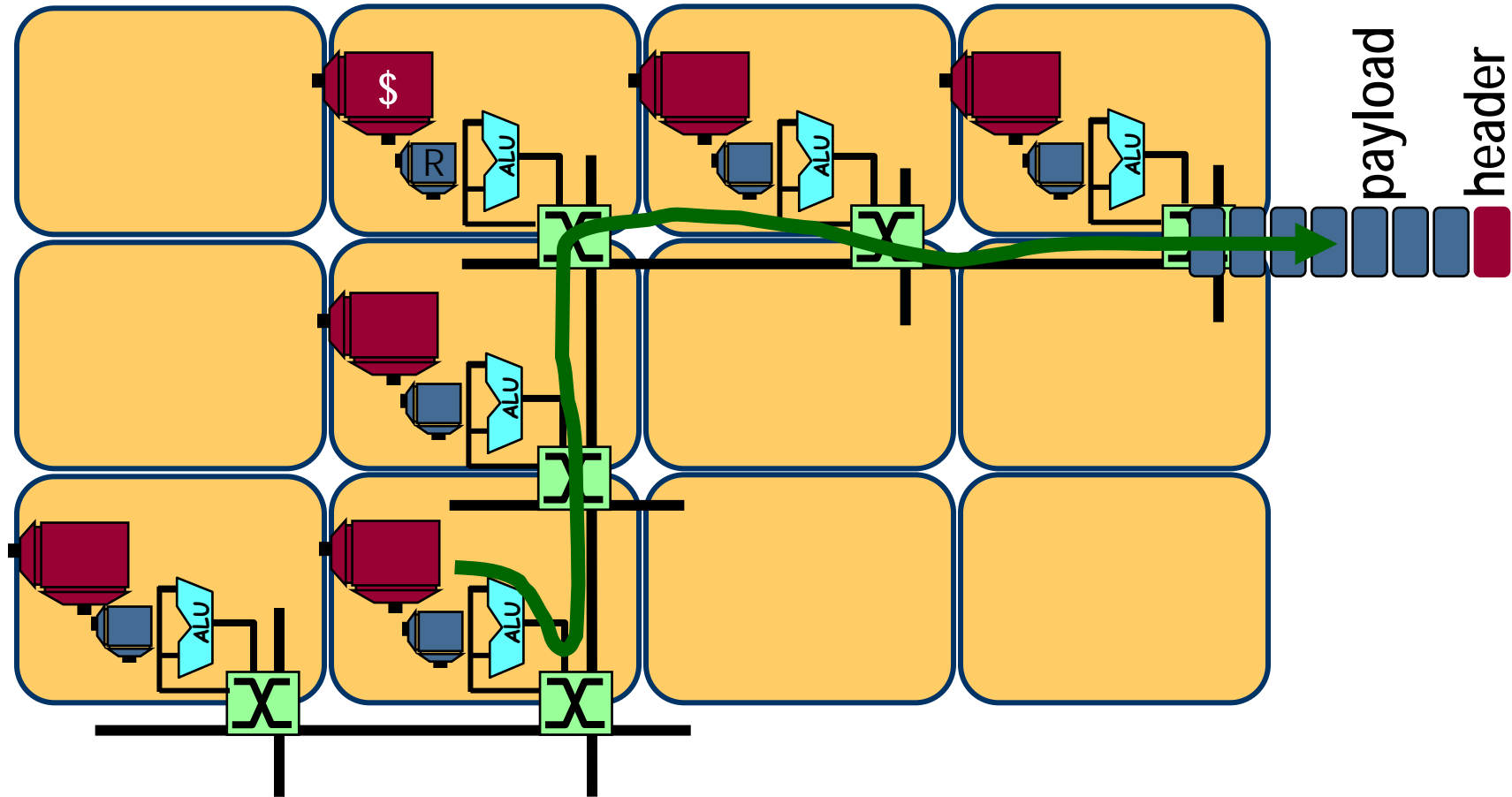
The Raw Chip



Scalar operand network (SON):
Capable of low latency transport of
small (or large) packets

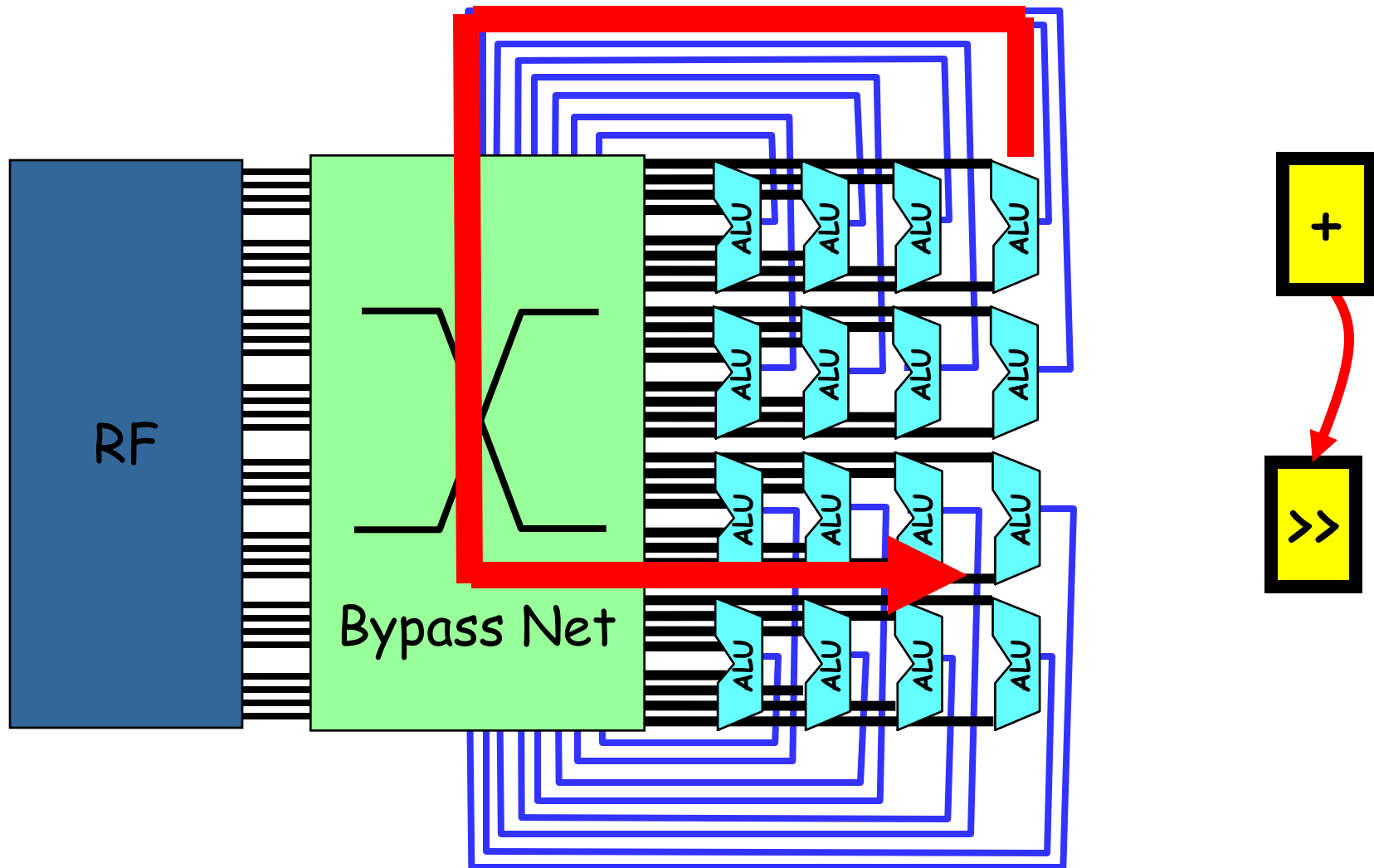
[IEEE TPDS 2005]

On-Chip Interconnect Routes Messages



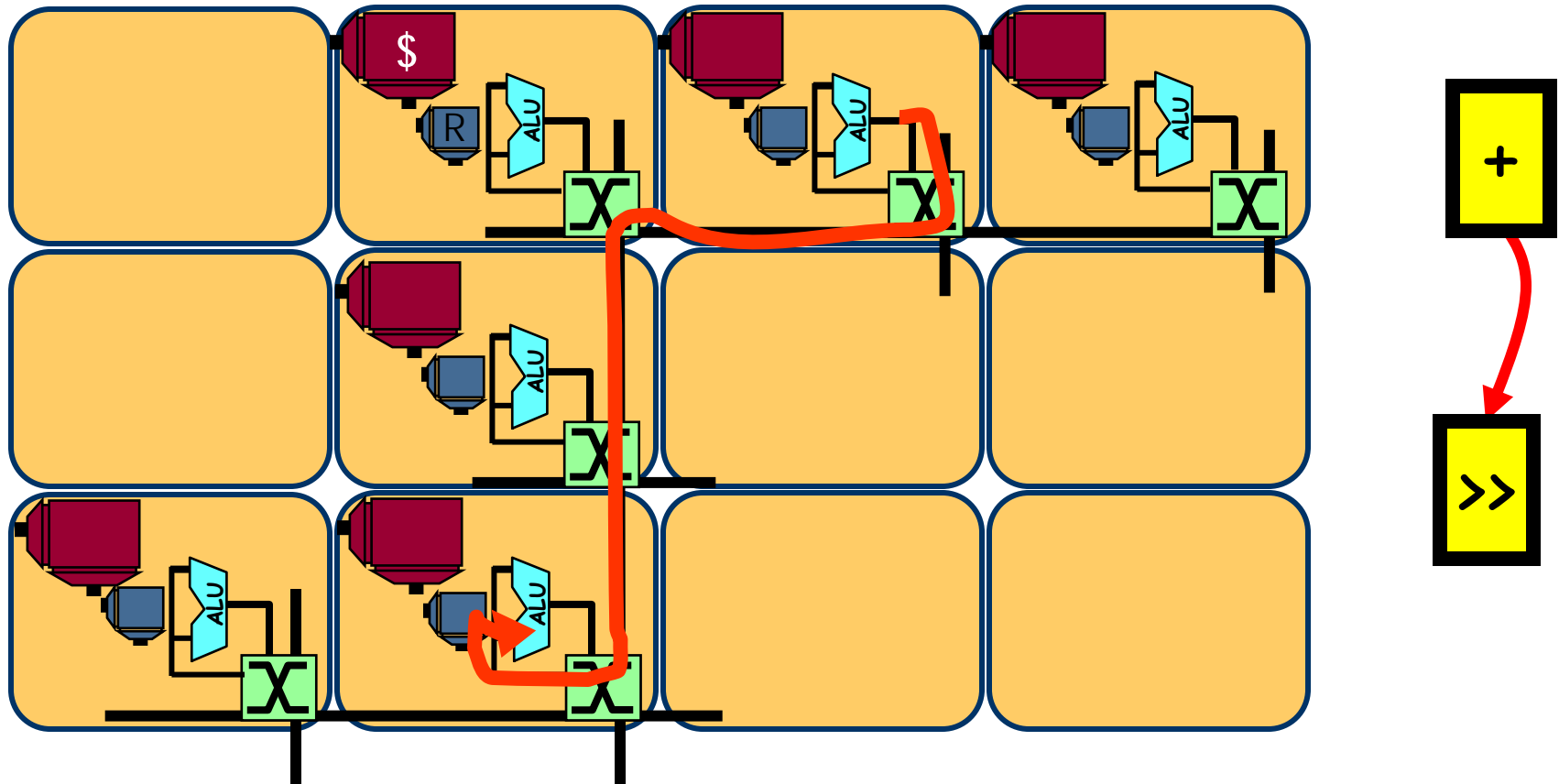
**For distributed cache access, off-chip misses, I/O,
user-level messages**

On-Chip Interconnect can Also Route Scalar Operands



Source: [Taylor ISCA 2004]

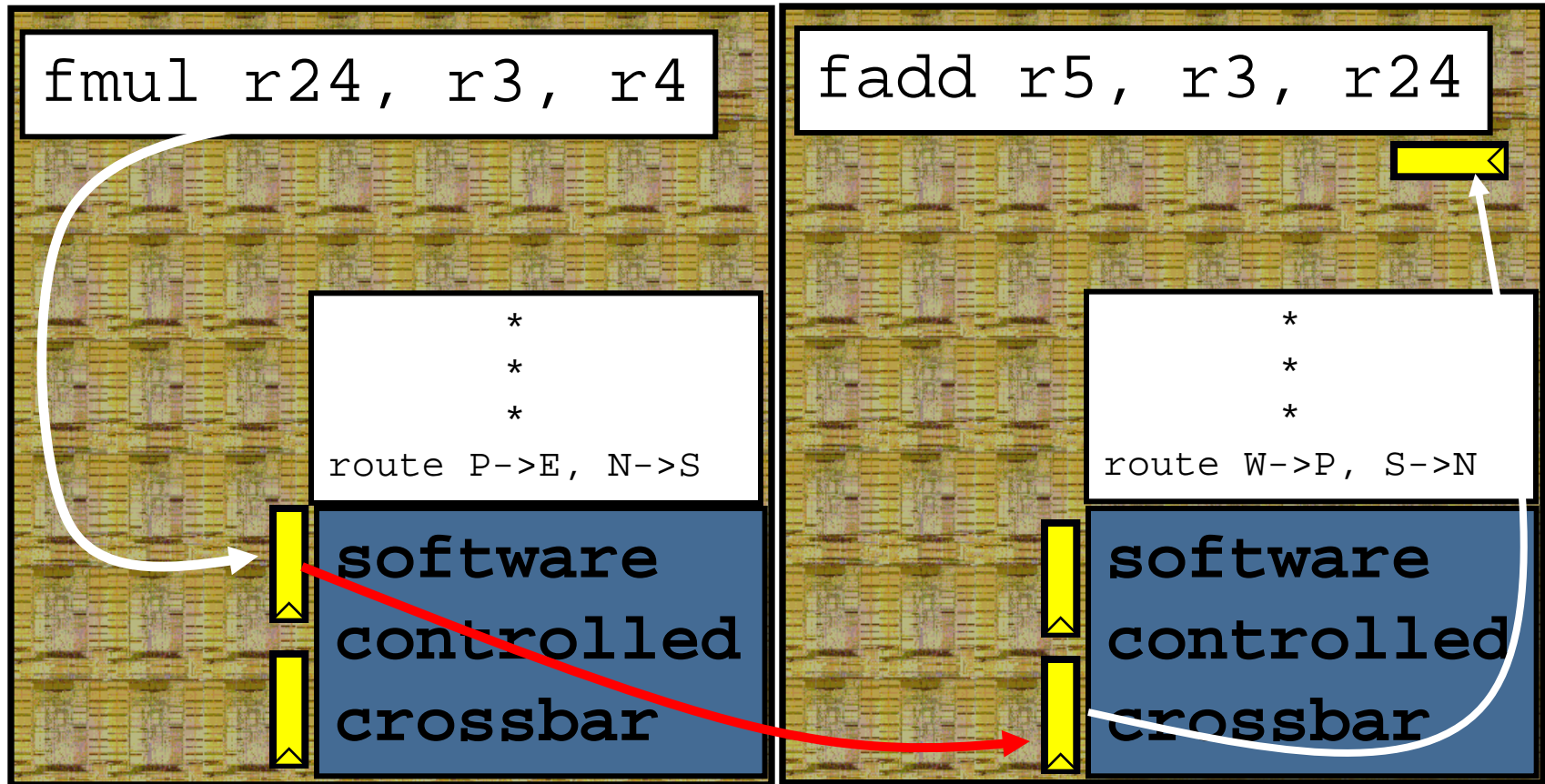
Operand Routing in a Tiled Architecture



Supports fine-grain instruction-level parallelism (ILP)
Or, How to make 2 cores look like one faster core

Scalar Operand Transport in Raw using Static Network

Goal: flow controlled, in order delivery of operands



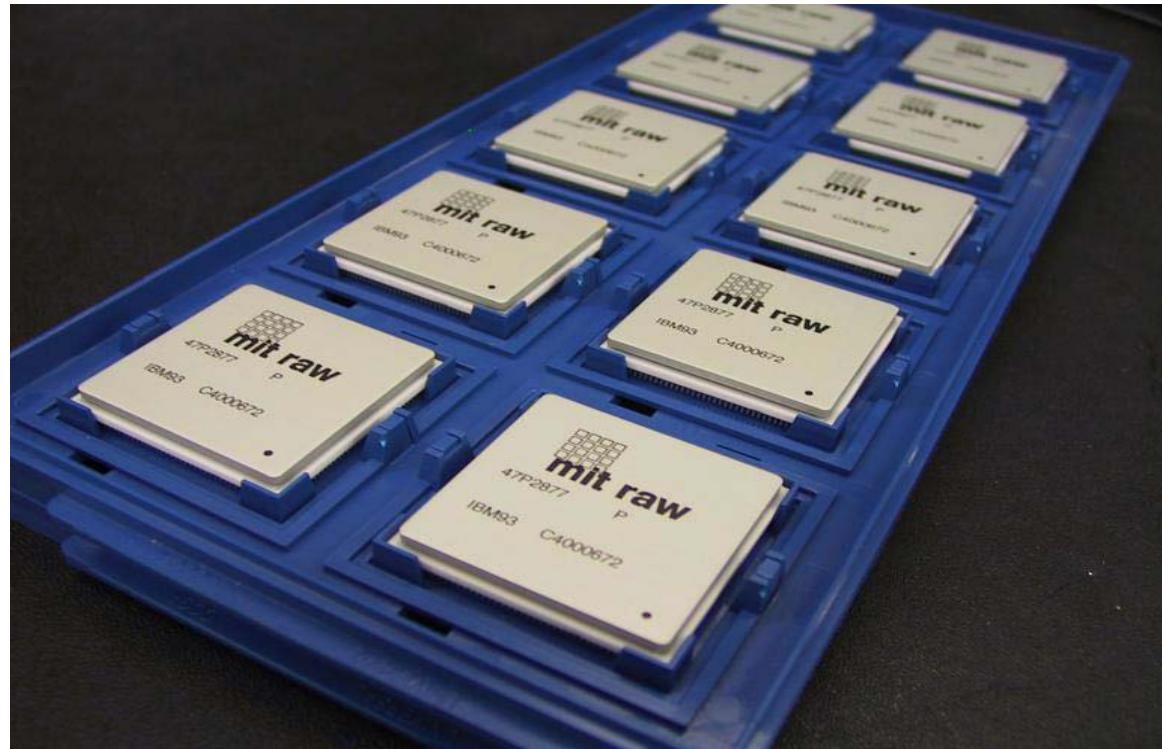
Virtual reality

Prototype reality

Product reality

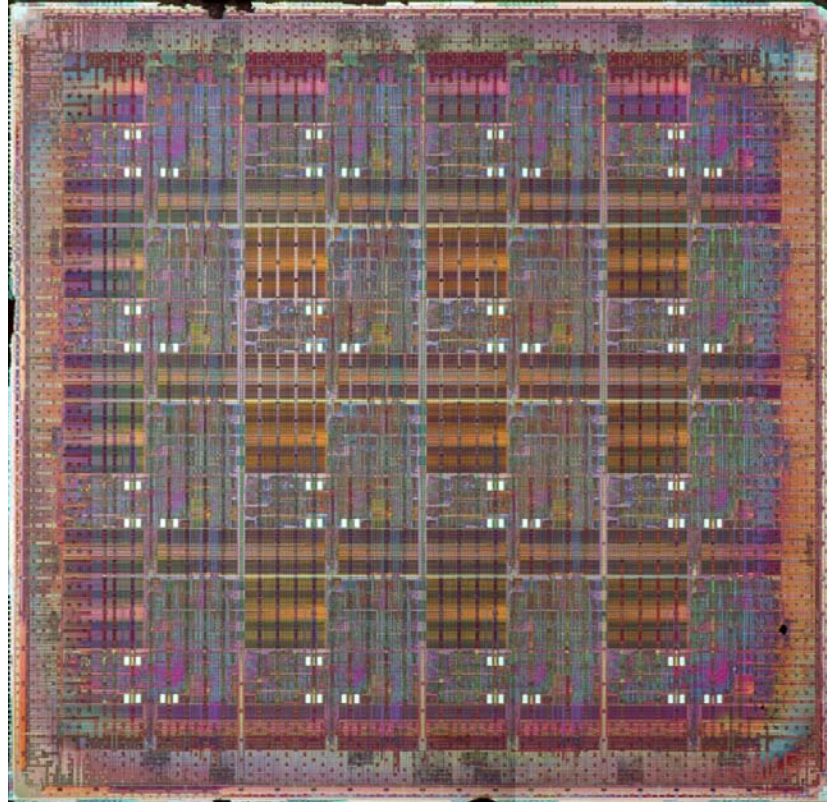
A Tiled Processor Architecture Prototype: the Raw Microprocessor

Michael Taylor
Walter Lee
Jason Miller
David Wentzlaff
Ian Bratt
Ben Greenwald
Henry Hoffmann
Paul Johnson
Jason Kim
James Psota
Arvind Saraf
Nathan Shnidman
Volker Strumpfen
Matt Frank
Rajeev Barua
Elliot Waingold
Jonathan Babb
Sri Devabhaktuni
Saman Amarasinghe
Anant Agarwal



October 02

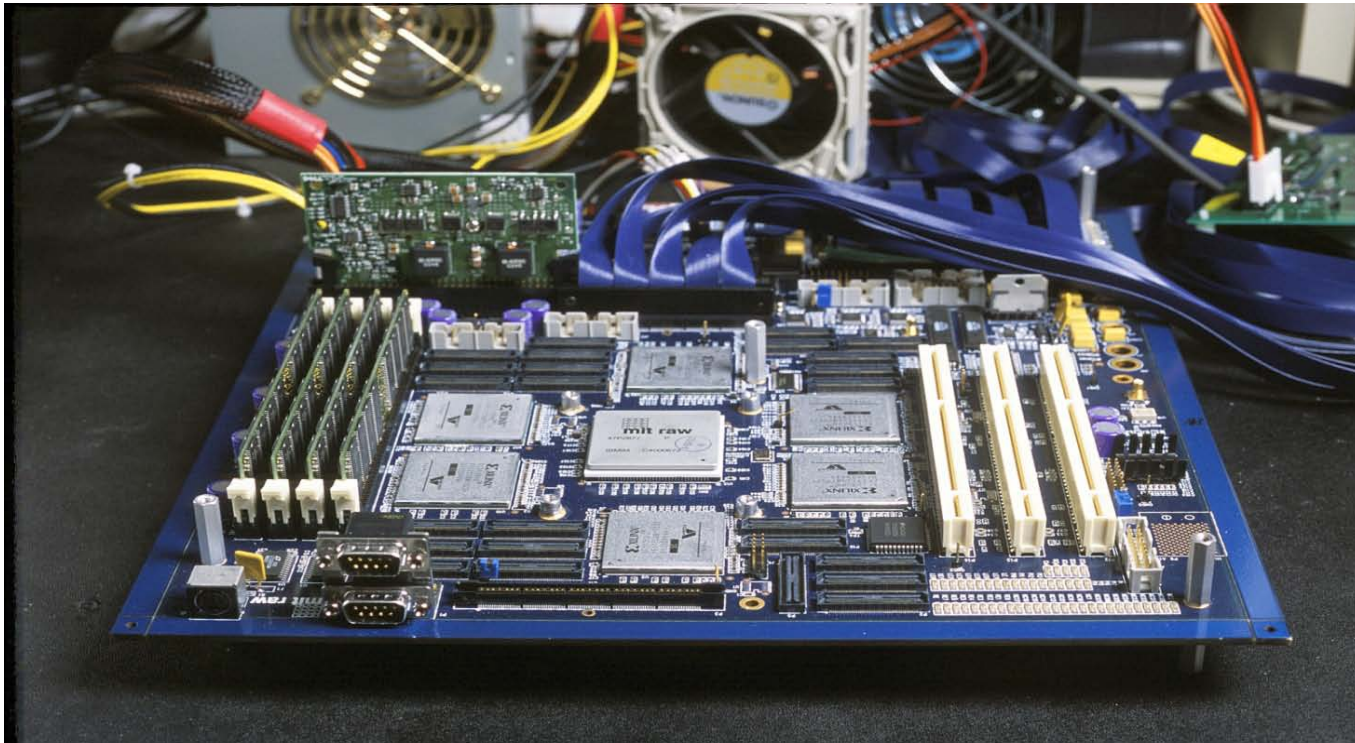
Raw Die Photo



IBM .18 micron process,
16 tiles,
425MHz,
18 Watts (vpenta)

[ISCA 2004]

Raw Motherboard



Raw Ideas and Decisions: What Worked, What Did Not

- Build a complete prototype system
- Simple processor with single issue cores
- Distributed ILP and static network
- Static network for streaming
- Multiple types of computation – ILP, streams, TLP, server
- Program counter in every tile

Raw Ideas and Decisions: What Worked, What Did Not

- Build a complete prototype system **Yes**
- Simple processor, single issue **1GHz, 2-way, inorder in 2016**
- Distributed ILP **Yes '02, No '06, Yes '12**
- Static network for streaming **No**
- Multiple types of computation – ILP, streams, TLP, server **Yes**
- PC in every tile **Yes**

Why Build?

- Compiler (Amarasinghe), OS and runtimes (ISI), apps (ISI, Lincoln Labs, Durand) folks will not work with you unless you are serious about building hardware
- Need motivation to build software tools -- compilers, runtimes, debugging, visualization – many challenges here
- Run large data sets (simulation takes forever even with 100 servers!)
- Many hard problems show up or are better understood after you begin building (how to maintain ordering for distributed ILP, slack for streaming codes)
- Have to solve hard problems – no magic!
- The more radical the idea, the more important it is to build
 - World will only trust end-to-end results since it is too hard to dive into details and understand all assumptions
 - Would you believe this: “Prof. John Bull has demonstrated a simulation prototype of a 64-way issue out-of-order superscalar”
- Cycle simulator became cycle *accurate* simulator only after HW got precisely defined
- Don't bother to commercialize unless you have a working prototype

Virtual reality

Prototype reality

Product reality

Why Do We Care?

Markets Demanding More Performance

Cloud

- Demands higher performance density
 - Performance per watt per foot²
- Demands general purpose compute

Wireless infrastructure market

- Demands higher throughput
 - More channels, more services, lower power
- Demands more services

Networking market

- Demands higher performance
 - Faster speeds 10-40 Gbps
- Demands more services

Digital multimedia market

- Demands higher quality
 - H.264 encoding for High Definition
- Demands more services



Cloud server rack



GGSN



Base Station



Security Appliances



Switches



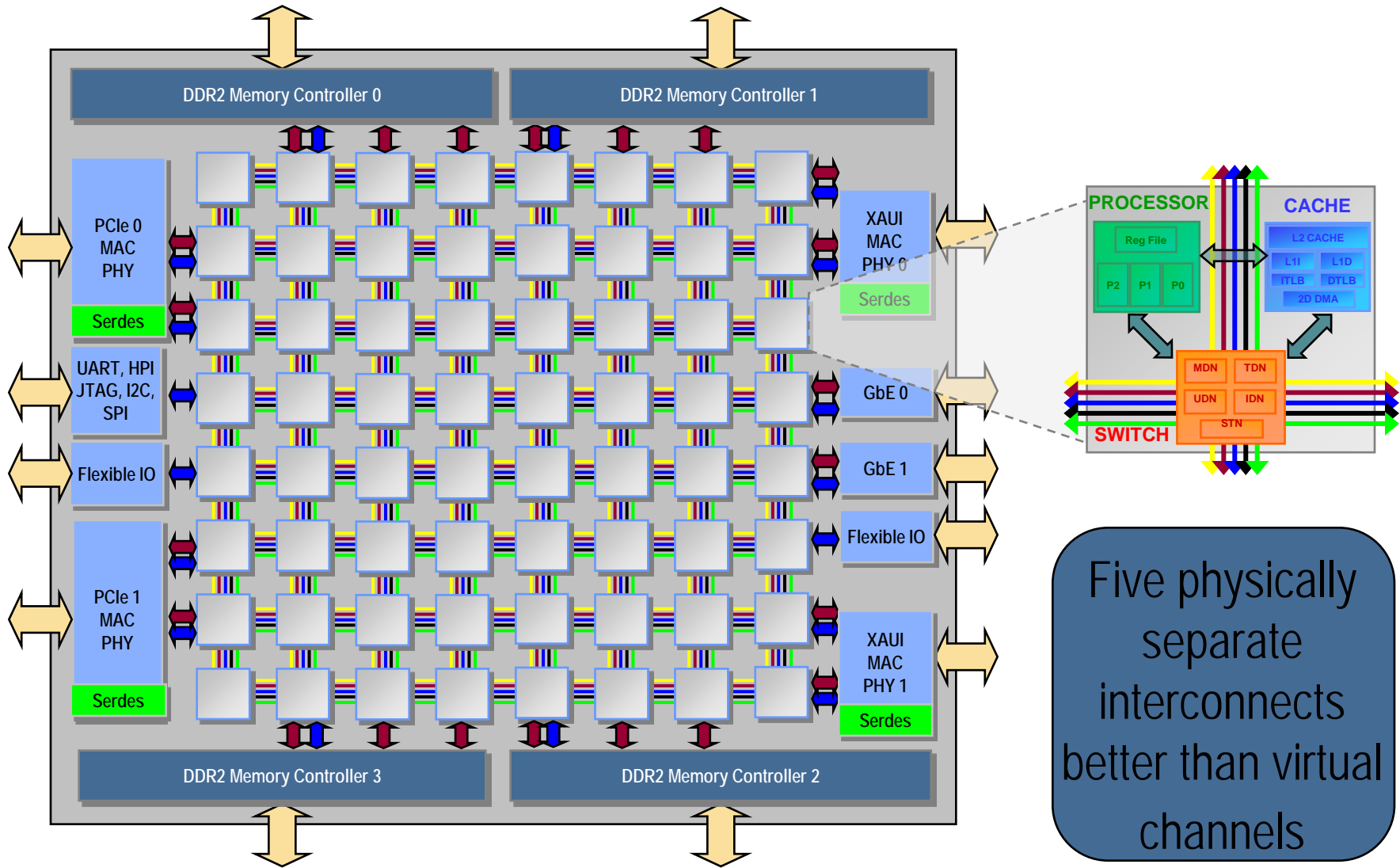
Video Conferencing



Cable & Broadcast

Tile Processor Block Diagram

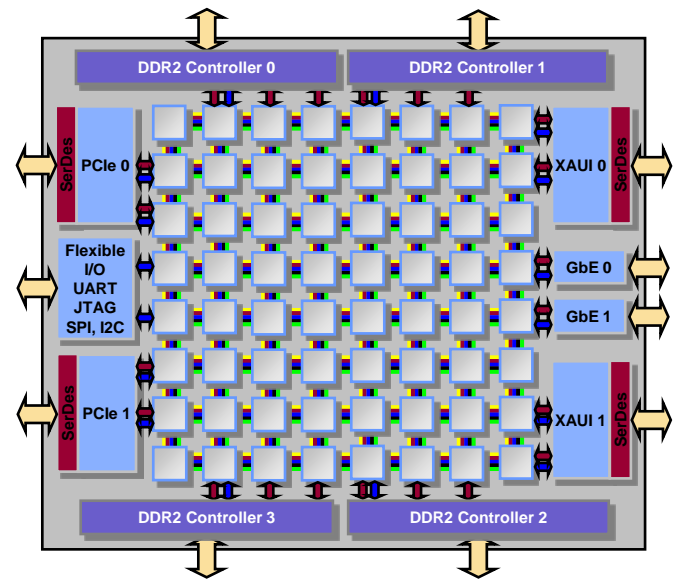
A Complete System on a Chip



Five physically separate interconnects better than virtual channels

System-on-a-chip in all Tile Processors

Performance	TILEPro36	TILE64	TILEPro64
# of cores	36	64	64
On-chip cache (MB)	3.2	5	5.6
Cache coherency	Yes w/ DDC	Yes	Yes w/ DDC
Operations (16/32-bit BOPS)	144/54	221/166	221/166
On chip bandwidth (Terabit/s)	12	32	38
Clock speed (MHz)	500	700, 866	700, 866
Power			
Typical power -5 device (W)	10-12	N/A	N/A
Typical power -7 device (W)	N/A	16-20	17-21
Typical power -9 device (W)	N/A	26-32	27-34
I/O and Memory			
PCIe and Ethernet bandwidth (Gbps)	20	40	40
Ethernet interfaces	1 XAUI, 2GbE	2 XAUI, 2GbE	2 XAUI, 2GbE
PCIe interfaces	1x 4-lanes	2 x 4-lanes	2 x 4-lanes
Flexible I/O pins	64	64	64
DDR2 bandwidth (peak Gbps)	100	200	200

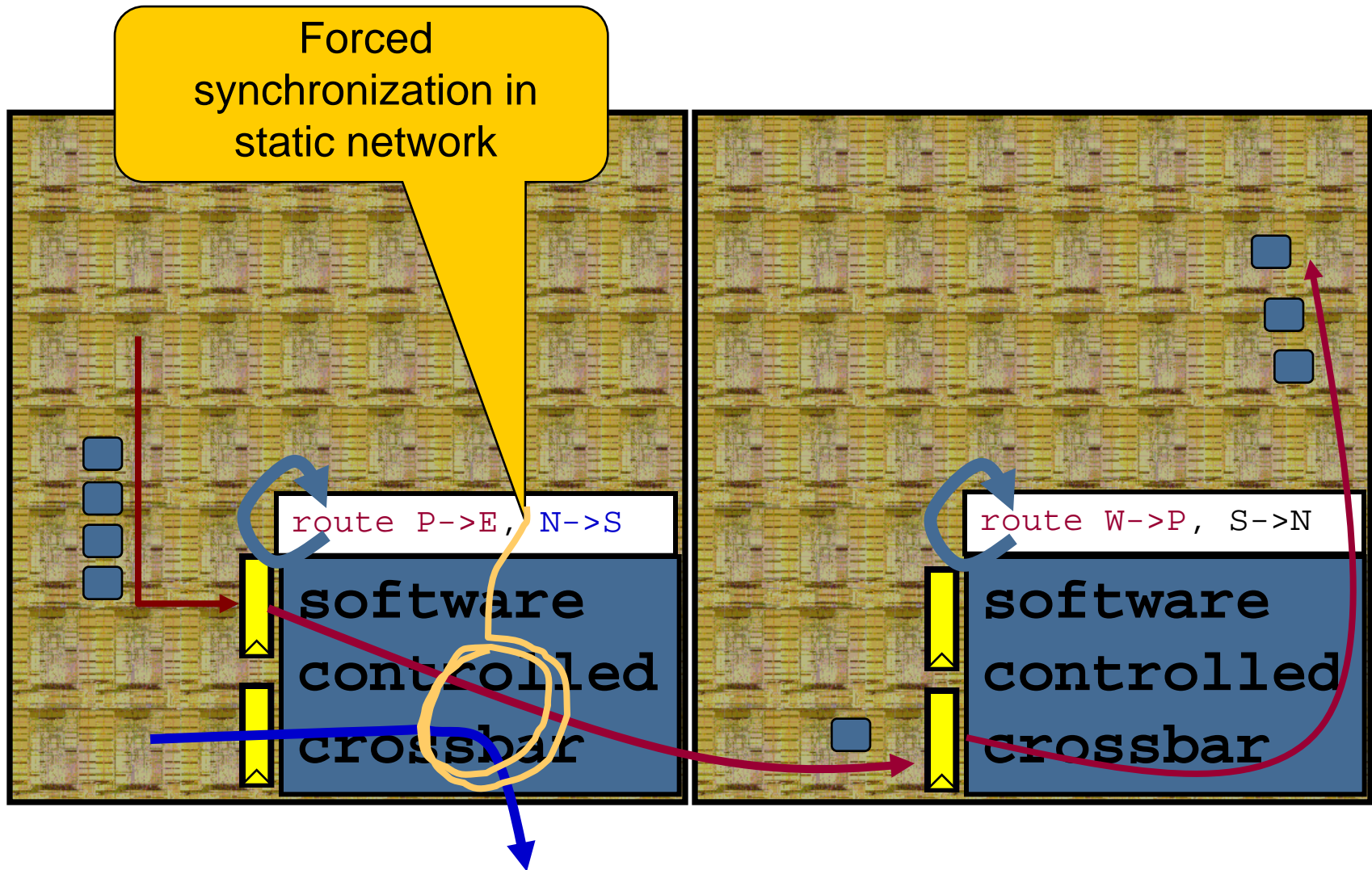


TILEPro64 Block Diagram

Product Reality Differences

- Market forces
 - Need crisper answer to “who cares”
 - SMP Linux programming with pthreads – fully cache coherent
 - C + API approach to streaming vs new language Streamit in Raw (Tilera’s iLib and TMC)
 - Special instructions for video, networking
 - Floating point needed in research project, but not in product for embedded market
- Lessons from Raw
 - Dynamic network for streams
 - HW instruction cache
 - Protected interconnects
- More substantial engineering
 - 3-way VLIW CPU, subword arithmetic
 - Engineering for clock speed and power efficiency
 - Completeness – I/O interfaces on chip – complete system chip. Just add DRAM for system
 - Support for virtual memory, 2D DMA
 - Runs SMP Linux (can run multiple OSes simultaneously)

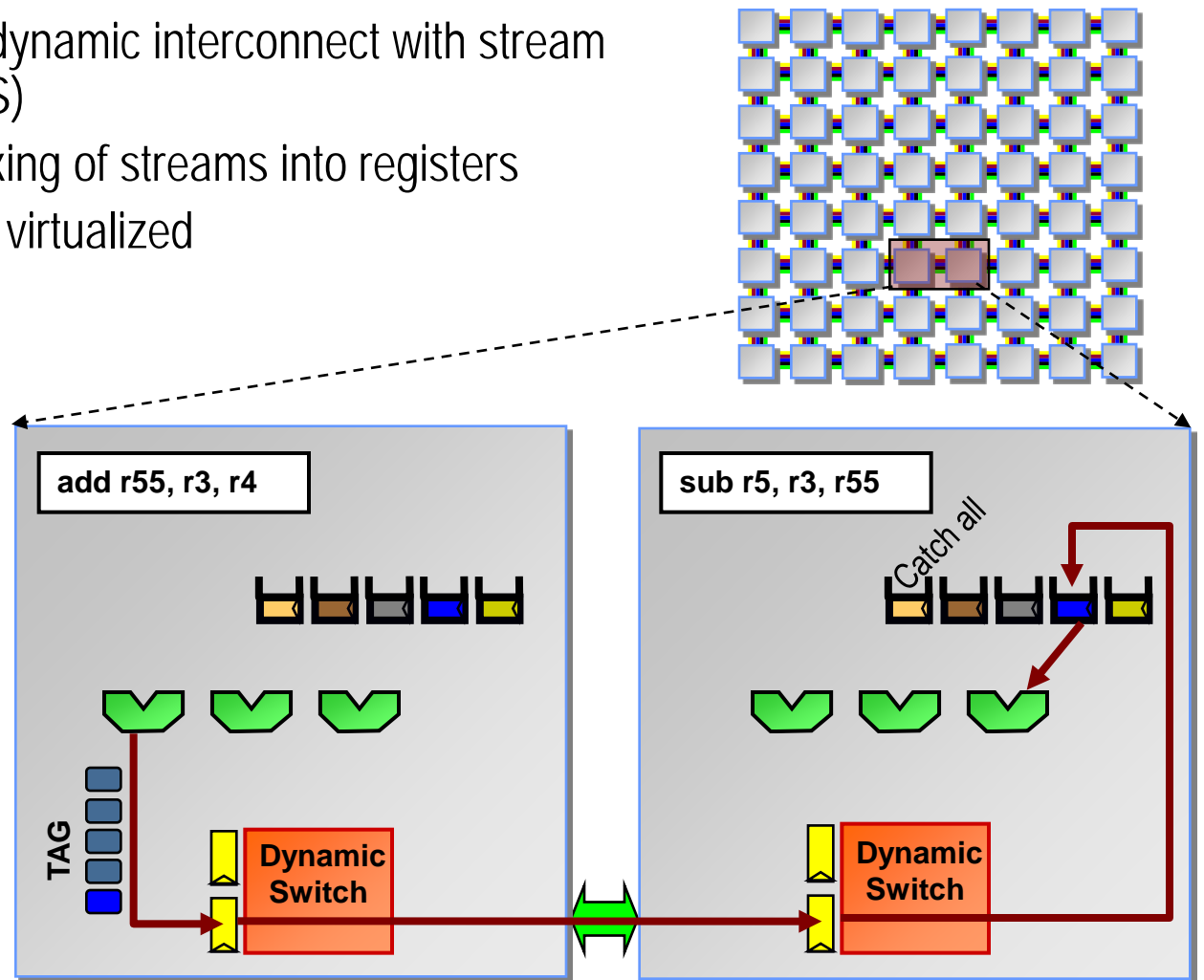
Raw Ideas and Decisions: Streaming – Interconnect Support



Streaming in Tiler's Tile Processor

- Streaming done over dynamic interconnect with stream demuxing (AsTrO SDS)
- Automatic demultiplexing of streams into registers
- Number of streams is virtualized

On-Chip Interconnect is not just about wires and routers; need to think about receive side demultiplexing, send occupancy



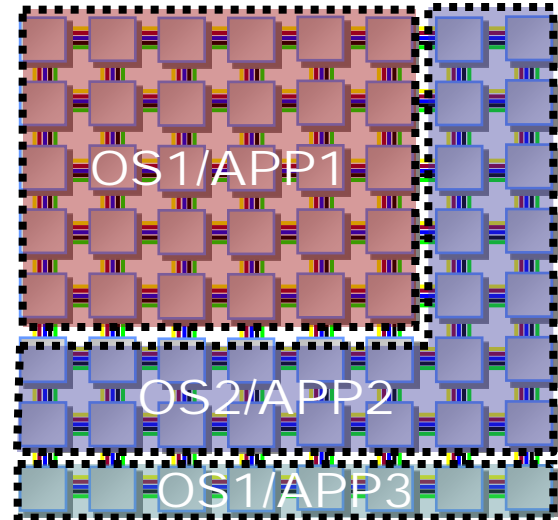
Tile Processor's Multicore Hardwall Technology for Protection and Virtualization

The protection and virtualization challenge

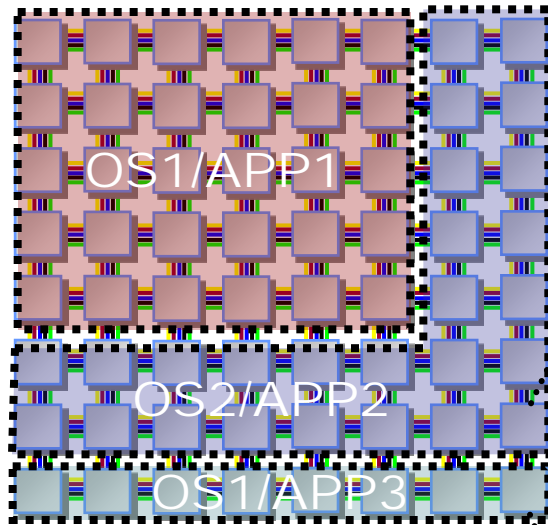
- Multicore interactions make traditional architectures hard to debug and protect
- Memory based protection will not work with direct IO interfaces and messaging
- Multiple OS's and applications exacerbate this problem
- E.g., By observing cache miss timing variations in a multiprocess core, an adversary can determine your secret key [Tromer Osvik Shamir 09]

Multicore Hardwall technology

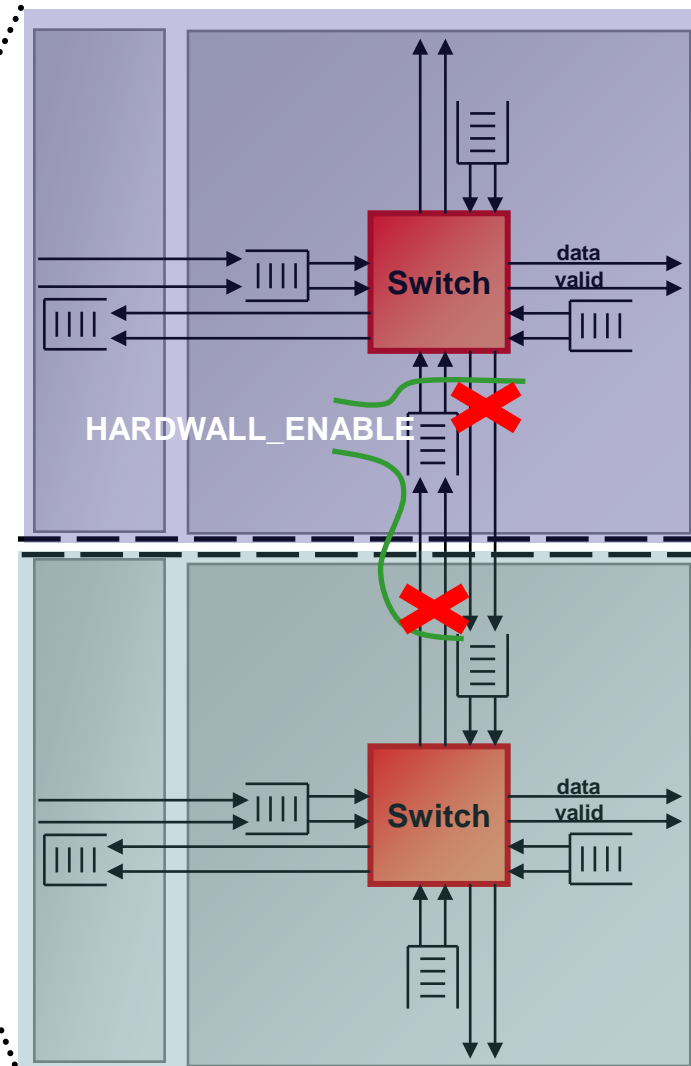
- Protects applications and OS by prohibiting unwanted interactions
- Configurable to include one or many tiles in a protected area



Multicore Hardwall Implementation



Interconnects can play a key role in support for virtualization and security



Virtual reality

Prototype reality

Product reality

What Does the Future Look Like?

Corollary of Moore's law: Number of cores will double every 18 months

	'02	'05	'08	'11	'14
Research	16	64	256	1024	4096
Industry	4	16	64	256	1024

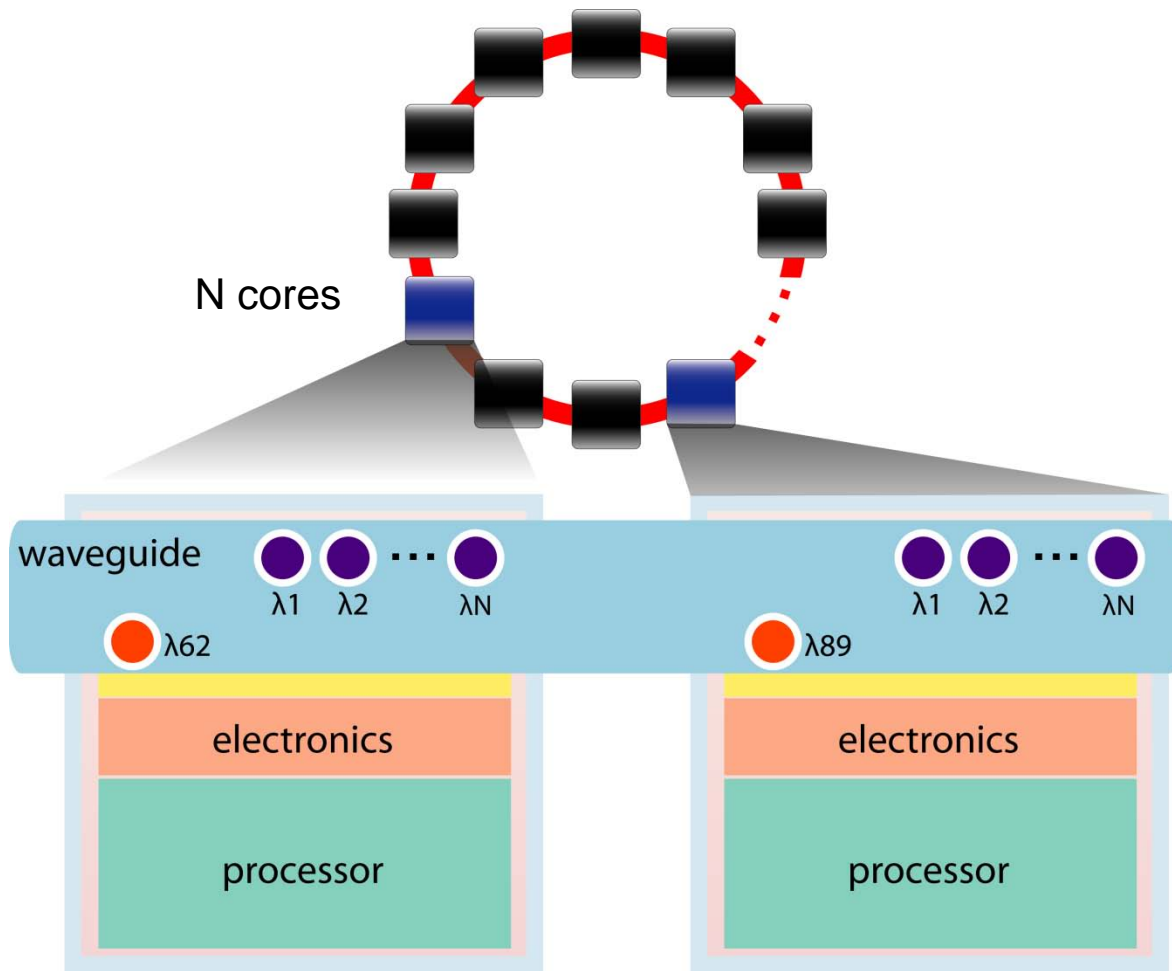
1K cores by 2014! Are we ready?

(Cores minimally big enough to run a self respecting OS!)

Research Challenges for 1K Cores

- 4-16 cores not interesting. Industry is there. University must focus on “1K cores”; Everything will change!
- Can we use 4 cores to get 2X through DILP? Remember cores will be 1GHz and simple! What is the interconnect?
- How should we program 1K cores? **Can interconnect help with programming?**
- Locality and reliability WILL matter for 1K cores. Spatial view of multicore?
- Can we add architectural support for programming ease? E.g., suppose I told you cores are free. Can you discover mechanisms to make programming easier?
- What is the right grain size for a core?
- How must our computational models change in the face of small memories per core?
- How to “feed the beast”? I/O and external memory bandwidth
- Can we assume perfect reliability any longer?

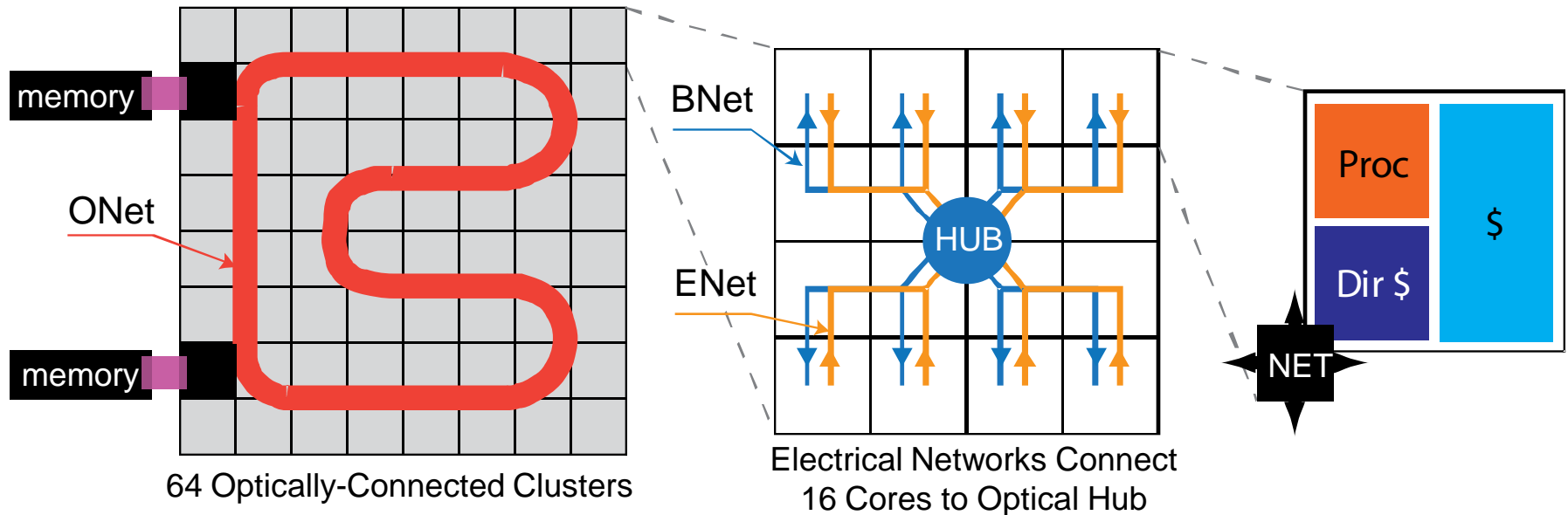
Optical Broadcast Network



- Electronic-photonic integration using standard CMOS process
- Cores communicate via optical WDM broadcast and select network
- Each core sends on its own dedicated wavelength using modulators ●
- Cores can receive from some set of senders using optical filters ●

Can cheap broadcast facilitated by optical interconnect improve programmability?

Scaling to 1000 Cores



- Purely optical design scales to about 64 cores
- After that, clusters of cores share optical hubs
 - ENet and BNet move data to/from optical hub
 - Dedicated, special-purpose electrical networks

Vision for the Future

- The 'core' is the logic gate of the 21st century



The following are trademarks of Tiler Corporation: Tiler, the Tiler Logo, Tile Processor, TILE64, Embedding Multicore, Multicore Development Environment, Gentle Slope Programming, iLib, iMesh and Multicore Hardwall. All other trademarks and/or registered trademarks are the property of their respective owners.