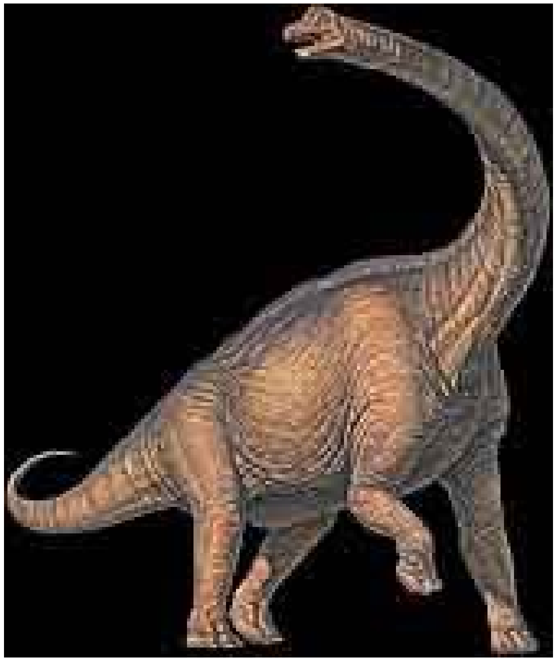
Whither goeth the microprocessor?

Anant Agarwal
MIT and
Tilera Corp.

What is a microprocessor?



Whatever's "Inside"

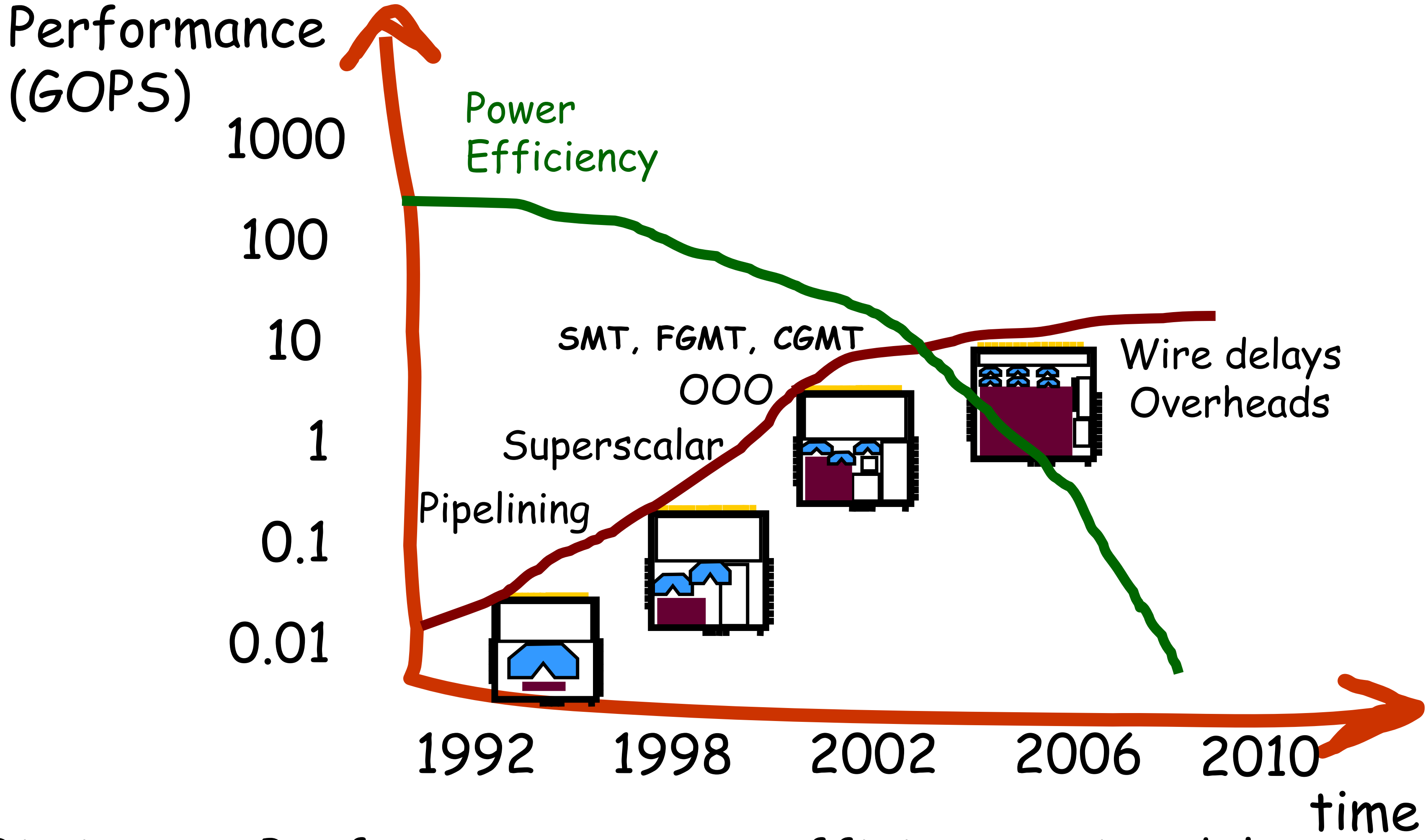
Whatever Intel sells

Whatever ISCA accepts

Whatever's in your desktop

A better definition:
A general purpose computing engine programmed in a high-level language

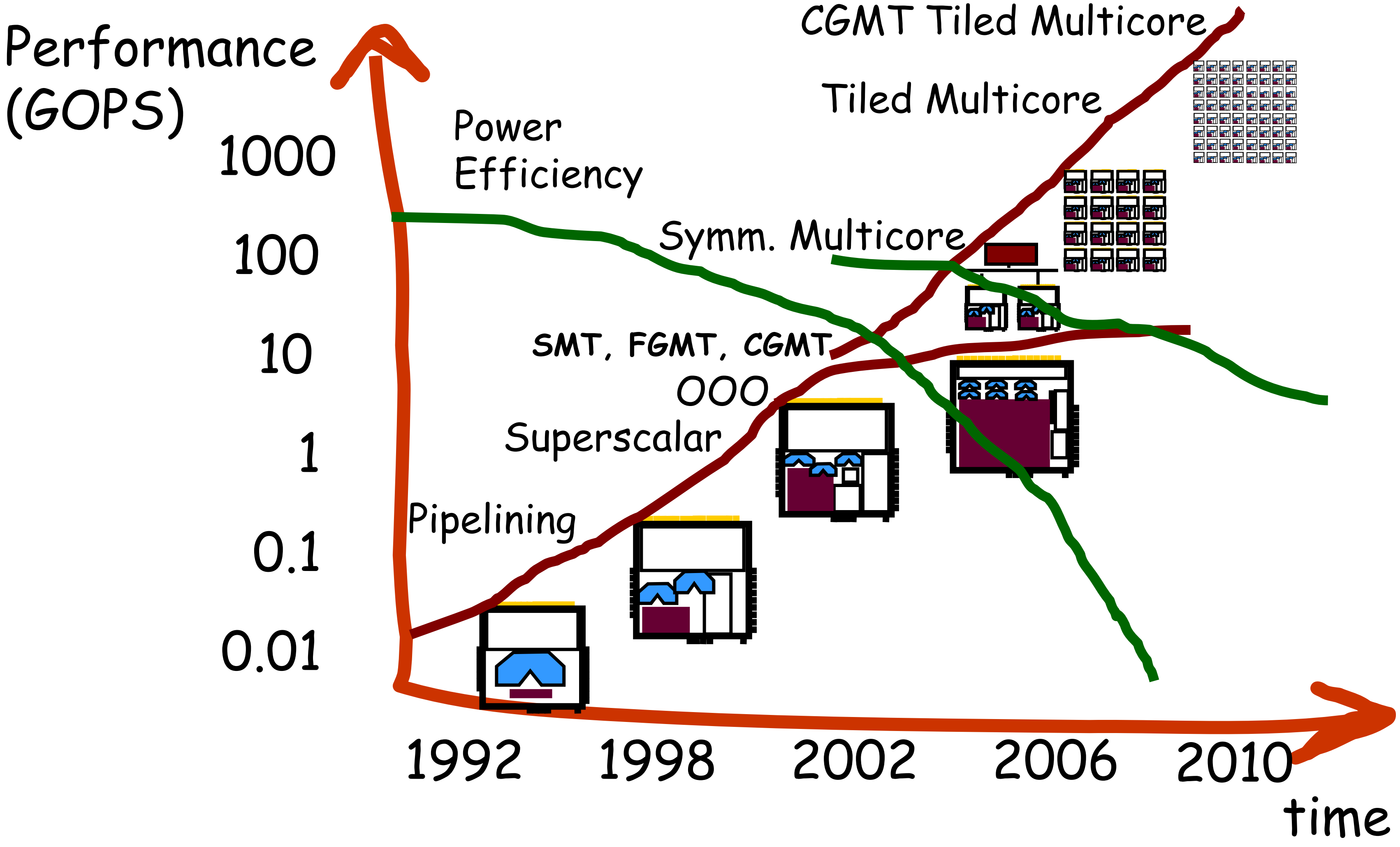
Let's take stock of where we stand with the desktop processor



Big issues: Performance, power efficiency, wire delays

Post 2000: SMT (PowerPC, Hyperthreaded x86s), FGMT (Niagara), CGMT (PowerPC, SOE Itanium)

Take stock of where we stand with the desktop processor



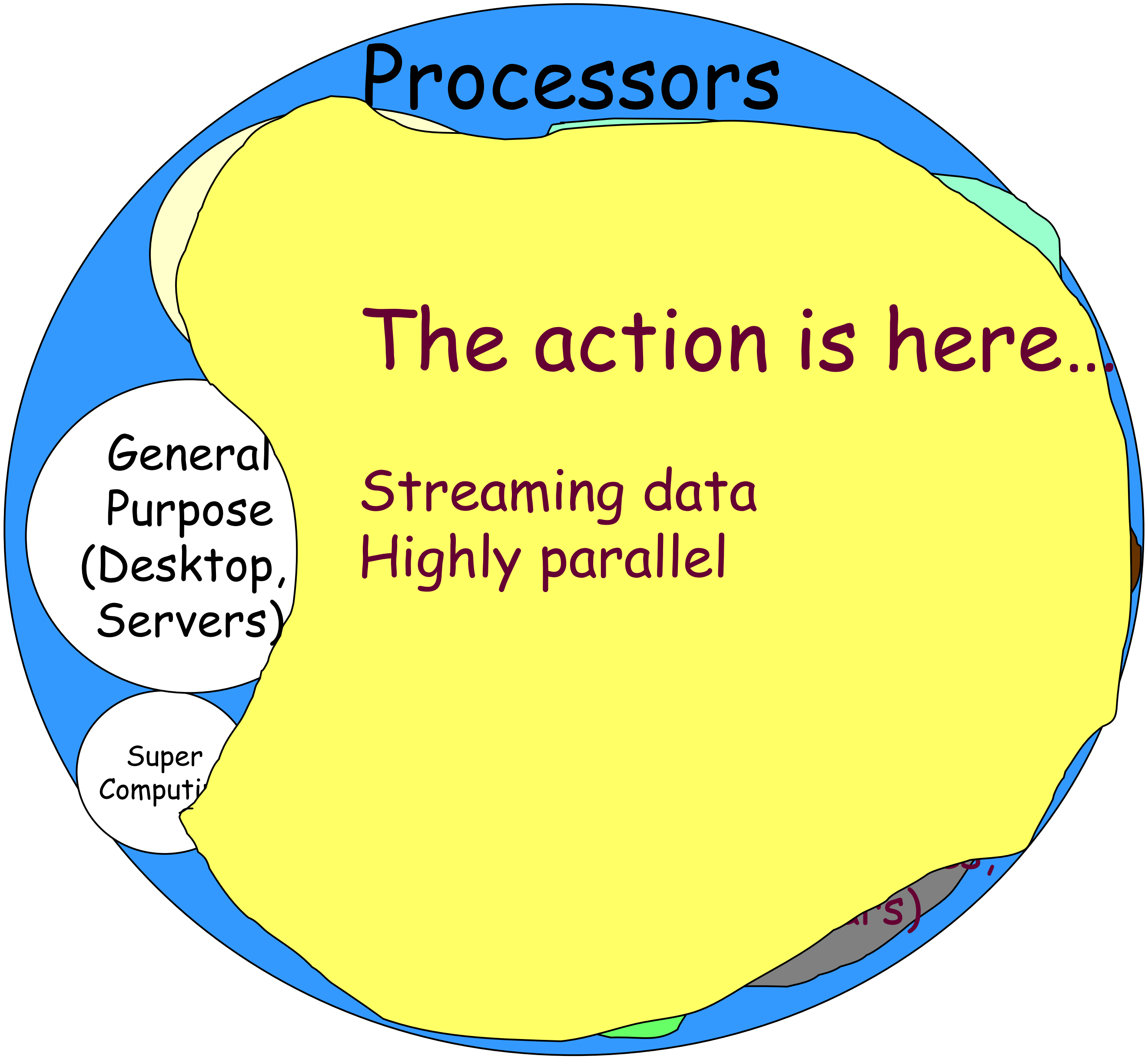
Big issues: Performance, power efficiency, wire delays

Two questions for multicore

Where do we find the parallelism?

Why is it more power efficient?

Where's the action?



Processors

General Purpose (Desktop, Servers)

Super Computi

The action is here...

Streaming data
Highly parallel

s,

rs)

Do we really need more performance?

- Embedded systems
  - H.264, HD, wireless 50gpbs, networking 10Gbps
- Servers
  - Enterprises have 1,000 to 30,000 servers!
- Desktops
  - Nah! My powerpoint runs plenty fast enough

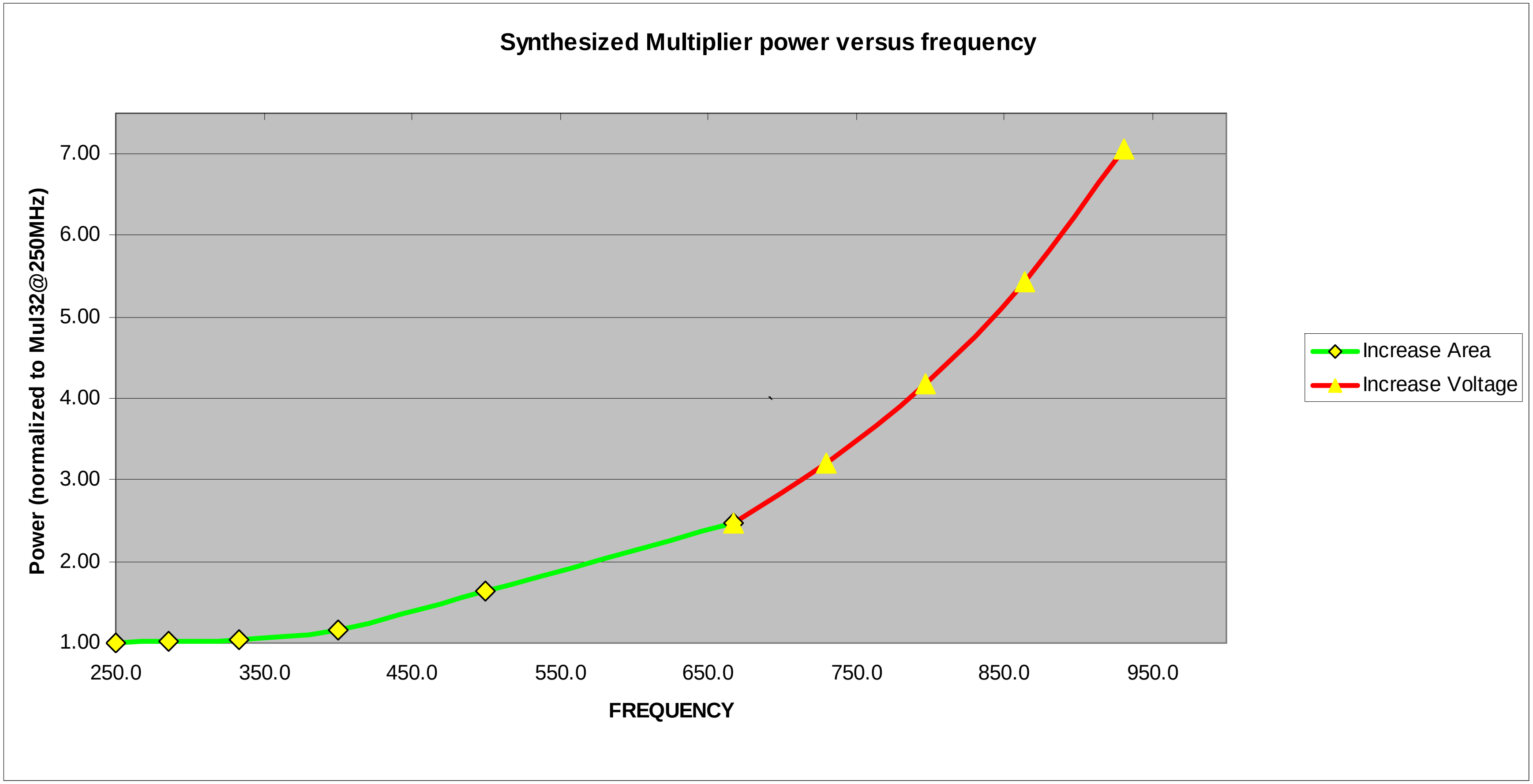
Apps generally highly parallel – streaming or throughput oriented

Two questions for multicore

Where do we find the parallelism?

Why is it more power efficient?

Frequency vs power tradeoff



Frequency increases as V

Power increases as V^3

For a 1% increase in freq, we suffer 3% increase in power

How to get 50% more performance with 20% less power

	Cores	V	Freq	Perf	Power
OOO	1	1	1	1	1
New OOO	1X	1.5X	1.5X	1.5X*	3.3X
Multicore	2X	0.75X	0.75X	1.5X**	0.8X

*Optimistic

**Assuming parallel application

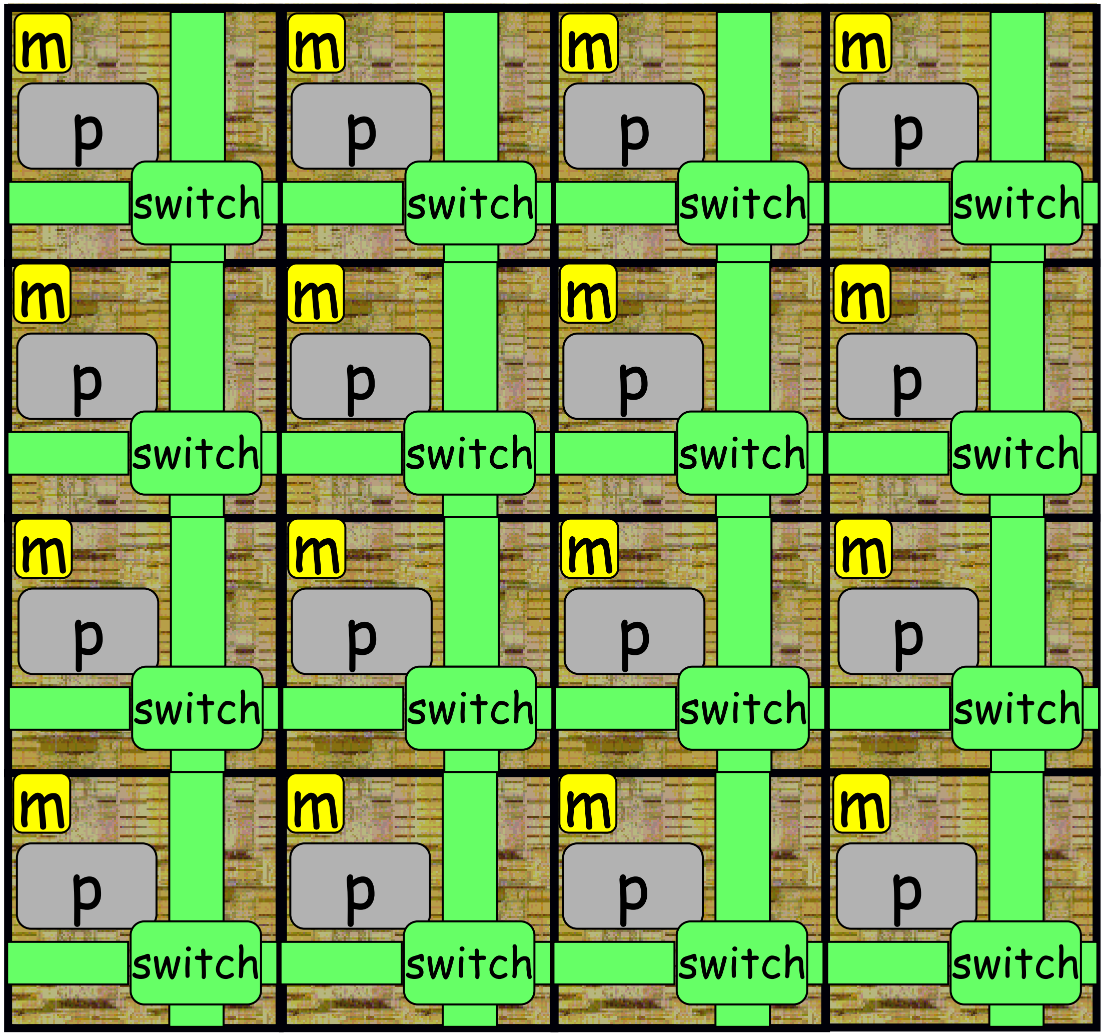
Assuming same technology

Scalability of various processor architectures

	Perf.	Power Eff.	Wire Delay	Prog. Ease
Wider issue OOO	✗	✗	✗	✓ ILP
OOO SMT	✗	✗	✗	— ILP, threads
Clustered VLIW	✗	—	—	✓ ILP
Busbased Multicore	—	—	✗	✗ threads
Tiled Multicore (Mesh)	✓	✓	✓	— ILP, threads
Tiled CGMT	✓	✓	✓	— ILP, threads

ILP on tiled multicore: Trips, Wavescalar, Raw

This is the future…



Tiled multicore

Looks like Borkar's (Intel) future multicore platform

What should academia work on wrt multicore?
Assume 3 to 10 year horizon

Remember:
Number of cores will double every 18 months

	‘02	‘05	‘08	‘11	‘14
Academia	16	64	256	1024	4096
Industry	2	8	32	128	512

Bus or ring does not cut it

Let’s look at some multicore research questions

Power efficiency

- The power problem has not gone away, it has just been postponed
  - Low power interconnects – e.g., signalling methods
  - Microarchitecture of banked caches for low power
  - Hierarchical memories for low power
  - How to build the smallest, most power efficient core/tile (as opposed to the most powerful core at whatever cost)
  - CGMT architectures – idle resources are bad when there is leakage

Interconnect

- What to do
  - Resurrect good old interconnection network research
  - New twist – can get 1-3 cycle latency as opposed to 1-3 microseconds between tiles. How to exploit this? ILP!
  - On-chip interconnection network architectures
  - Processor-interconnect interfaces for low latency
  - Processor-interconnect interfaces for streaming
- What not to do
  - How to make buses go faster (industry is way ahead!)

Buses don't have a good ring to them

Memory and comms systems

- What to do
  - How to get the most out of small caches – caches consume lots of power and leak like sieves
  - Scalable coherence methods – directory based, software based, page-based DSMs
  - Message based communications
  - Stream based communications
  - Streaming memories
- What not to do
  - How to build a better snoop filter
  - Other twists on snoop coherence

Programming and apps

- What to do
  - How to use a 1000 cores
  - Shared memory based approaches
  - Message based approaches
  - Stream approaches
  - Hybrid approaches

Exploiting cores

- What to do
  - DMR and TMR for reliability
  - "Helper cores"
  - How to make power-performance tradeoffs – use more cores at lower frequency versus few cores at higher frequency
  - CGMT techniques – parallelism within a core

Parallel architecture all over again... so what is new?

The constant factors are much better

The sequential processor juggernaut has ended.
We have no choice

Intel is doing it too!