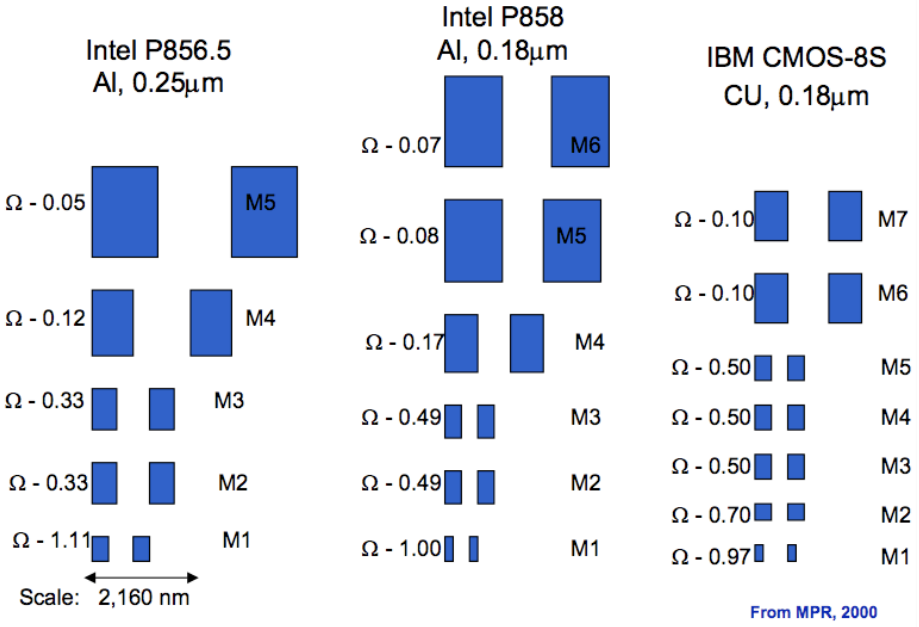
Lecture 04
Interconnect Overhead

Specific topics include a short review of logic scaling, the impact of technology scaling on interconnect, how interconnect scaling impacts the current solution to problems associated with logic scaling (multi-core architectures), and information processing "tokens"

Background Slides

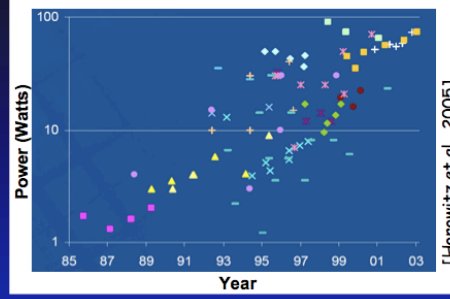
Wire Spacing Comparisons



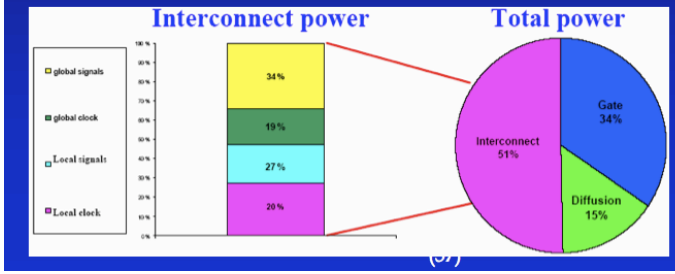
From MPR, 2000

Interconnect Power Dissipation

- Power dissipation is arguably the most critical problem in high-performance chip design
- Over last two decades microprocessor power dissipation grows exponentially and primary contribution from interconnects

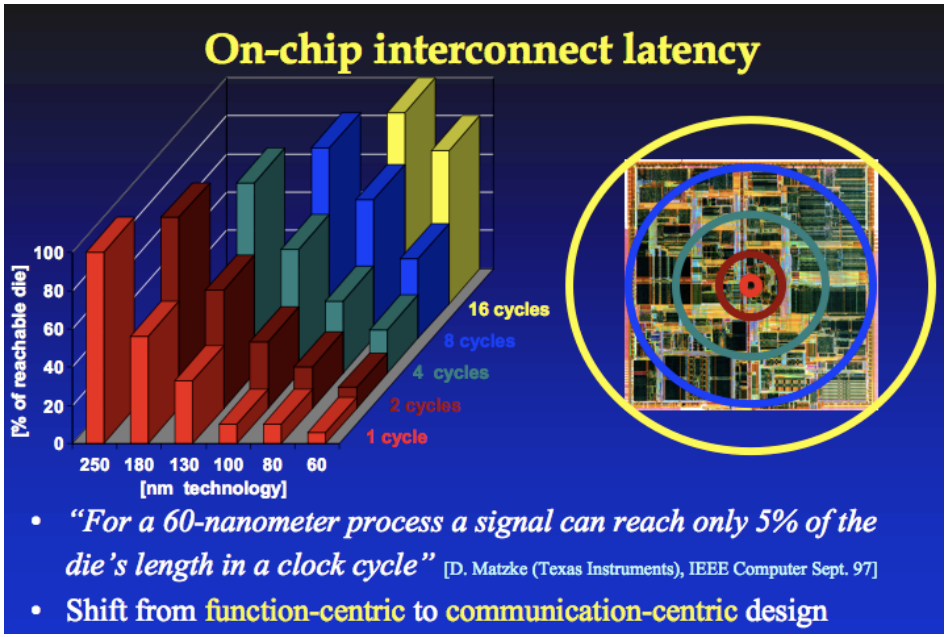


[Horowitz et al., 2005]



Interconnect responsible for 50% of dynamic power dissipation [Magen et al., 2004]

COLUMBIA UNIVERSITY



Dally Paper Slides

NW topologies

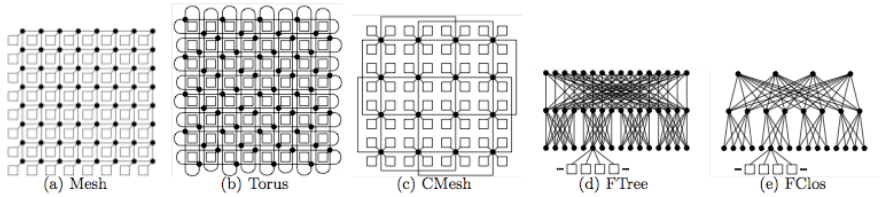


Figure 8: Network Topologies

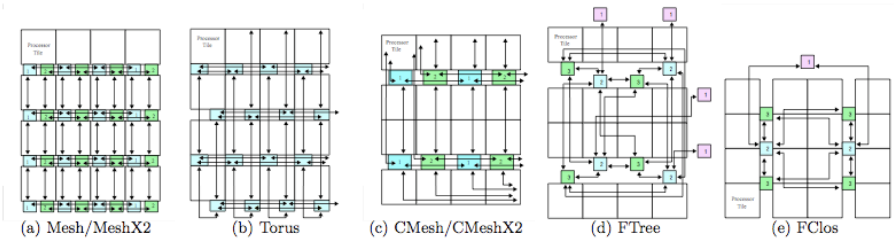


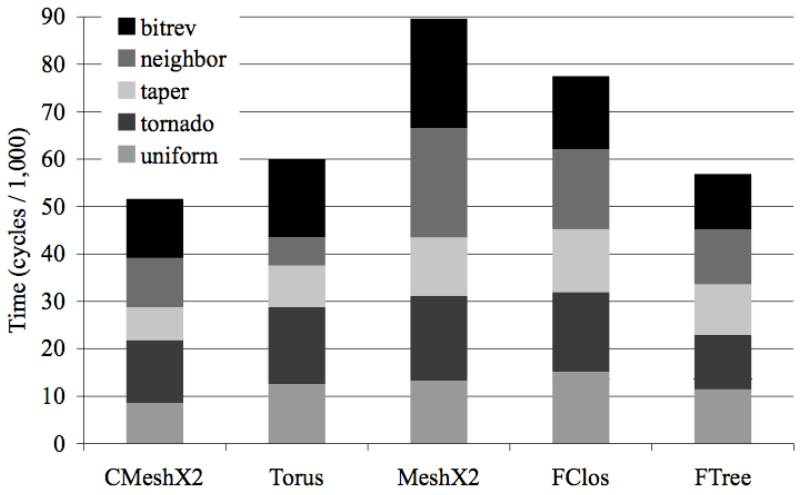
Figure 9: Placement of Routers used to Estimate Area (Lower Left Quadrant)

From Balfour, Dally, *Supercomputing*

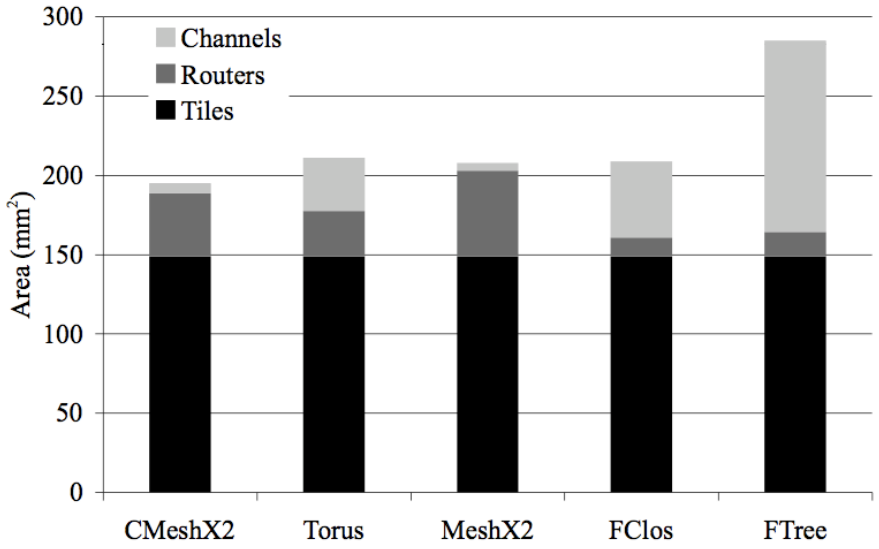
Preferred NW configurations

Table 3: Preferred Network Configurations

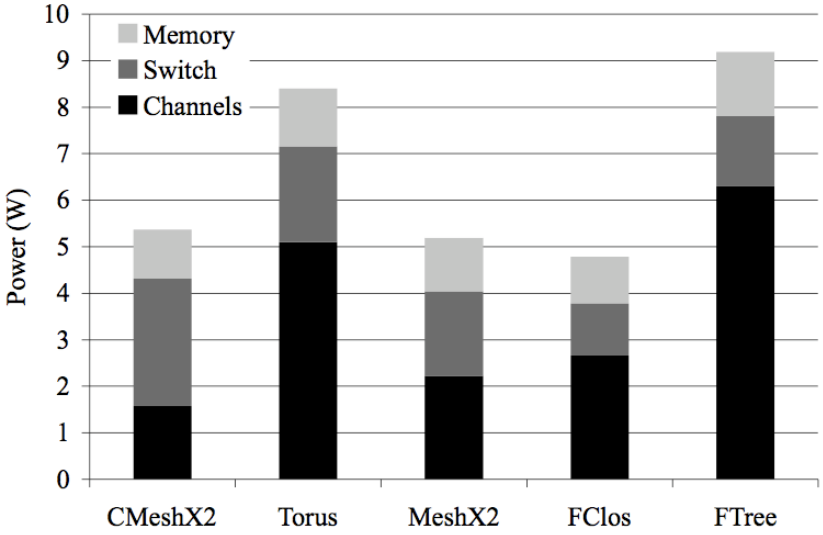
	H	t_r	B_C	w	B_B	T_c	T_s	T_0
Mesh	$6\frac{1}{4}$	2	16	192	3,072	5.3	3	17.8
MeshX2	$6\frac{1}{4}$	2	32	192	6,144	5.3	3	17.8
Torus	5	2	32	288	9,216	4.0	2	14.0
CMesh	$3\frac{1}{8}$	3	16	288	4,608	2.1	2	11.5
CMeshX2	$3\frac{1}{8}$	3	32	288	9,216	2.1	2	11.5
FTree	$4\frac{3}{8}$	2	64	144	9,216	4.4	4	13.1
FClos	$4\frac{3}{8}$	2	32	144	4,608	3.5	4	12.2



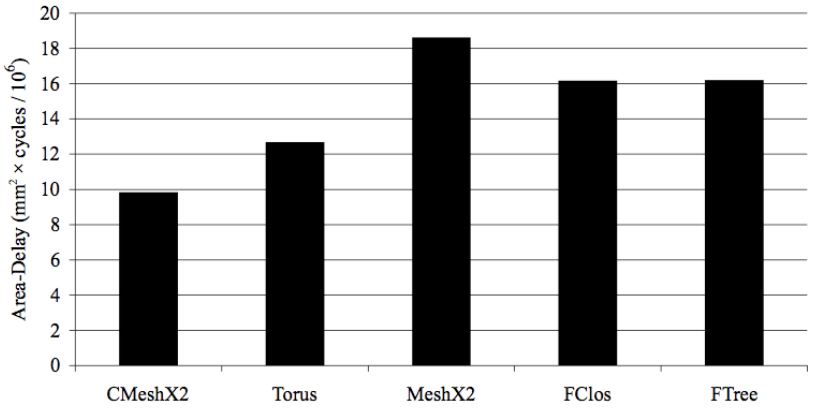
(a) Completion Time by Pattern



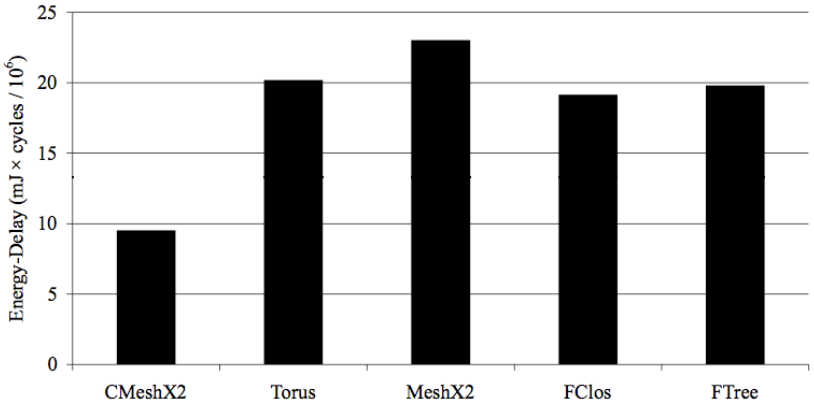
(b) Chip Area



(c) Network Power Dissipation



(d) Area Delay Metric



(e) Energy Delay Metric

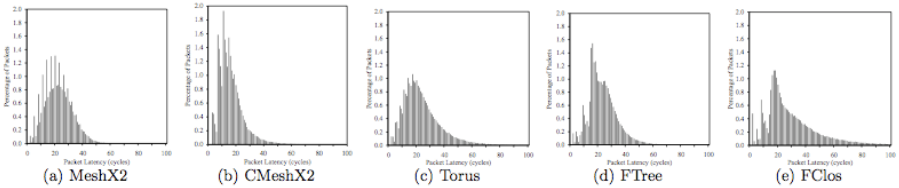


Figure 11: Workload Packet Latency Distribution for Uniform Random Traffic Pattern

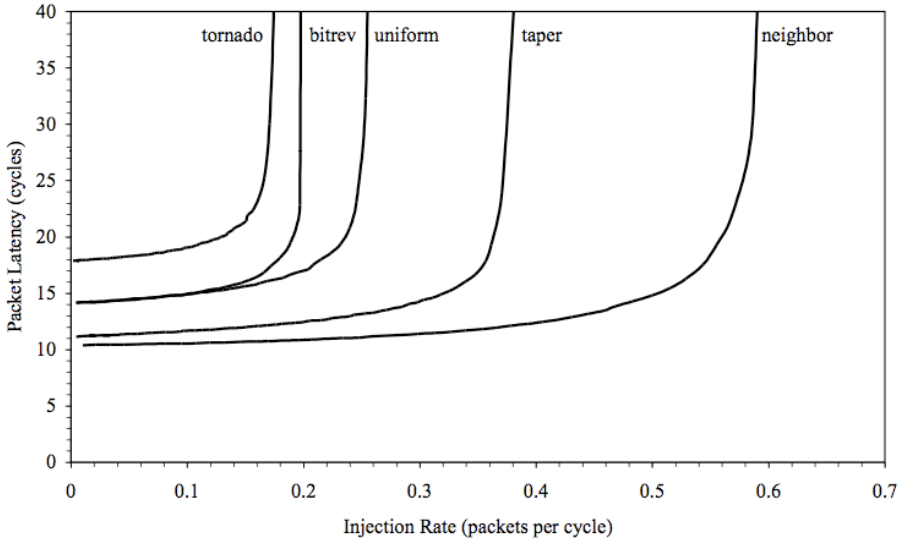


Figure 12: Offered Latency for CMeshX2 Network

Amdahl's Law Slides

Impediments to Parallel Performance

★ Contention for access to shared resources
- i.e. multiple accesses to limited # of memory banks may dominate system scalability

★ Programming languages, environments, & methods:
- Need simple semantics that can expose computational properties to be exploited by large-scale architectures

★ Algorithms
- What if you write good code for a 4-core chip, and then get an 8-core chip?

★ Cache coherency
- P1 writes, P2 can read
  - Protocols can enable $ coherency but add overhead

★ Overhead where no actual processing is done.

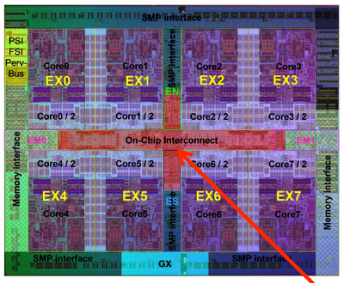
Impediments to Parallel Performance

- Latency ★
  - Is already a major source of performance degradation
  - Architecture charged with hiding local latency
    - (that's why we talked about registers & caches)
  - Hiding global latency is also task of programmer
    - (I.e. manual resource allocation)
- Today:
  - access to DRAM in 100s of CCs
  - round trip remote access in 1000s of CCs
  - multiple clock cycles to cross chip or to communicate from core-to-core
    - Not "free"

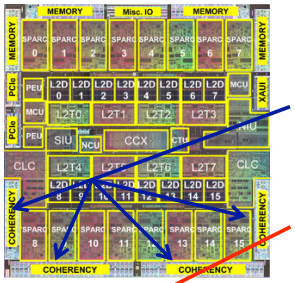
★ Overhead where no actual processing is done.

Recent multi-core die photos

(Route packets, not wires?)



http://dx.doi.org/10.1109/ISSCC.2010.5434074

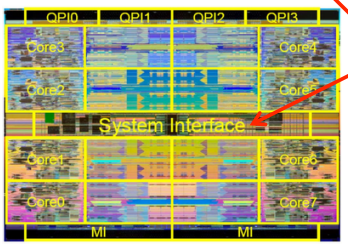


http://dx.doi.org/10.1109/ISSCC.2010.5434030

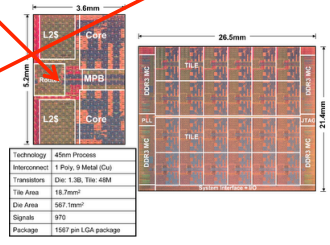
Likely to see HW support for parallel processor configurations:
- Coherency

+

- On-chip IC NWs



http://dx.doi.org/10.1109/ASSCC.2009.5357230



http://dx.doi.org/10.1109/ISSCC.2010.5434077

…takes advantage of 8 voltage and 28 frequency islands to allow independent DVFS of cores and mesh. As performance scales, the processor dissipates between 25 W and 125 W. … 567 mm² processor on 45 nm CMOS integrates 48 IA-32 cores and 4 DDR3 channels in a 2D-mesh network. Cores communicate through message passing using 384 KB of on-die shared memory. Fine-grain power management

Impediments to Parallel Performance

- All ★'ed items also affect $\text{Fraction}_{\text{parallelizable}}$
  - (and hence speedup)

$$\text{Speedup} = \frac{1}{\left[1 - \text{Fraction}_{\text{parallelizable}}\right] + \frac{\text{Fraction}_{\text{parallelizable}}}{N}}$$

Multi-core only as good as algorithms that use it