

# Summary of Application Trends

Transition to parallel computing has occurred for scientific and engineering computing

In rapid progress in commercial computing

- Database and transactions as well as financial
- Usually smaller-scale, but large-scale systems also used

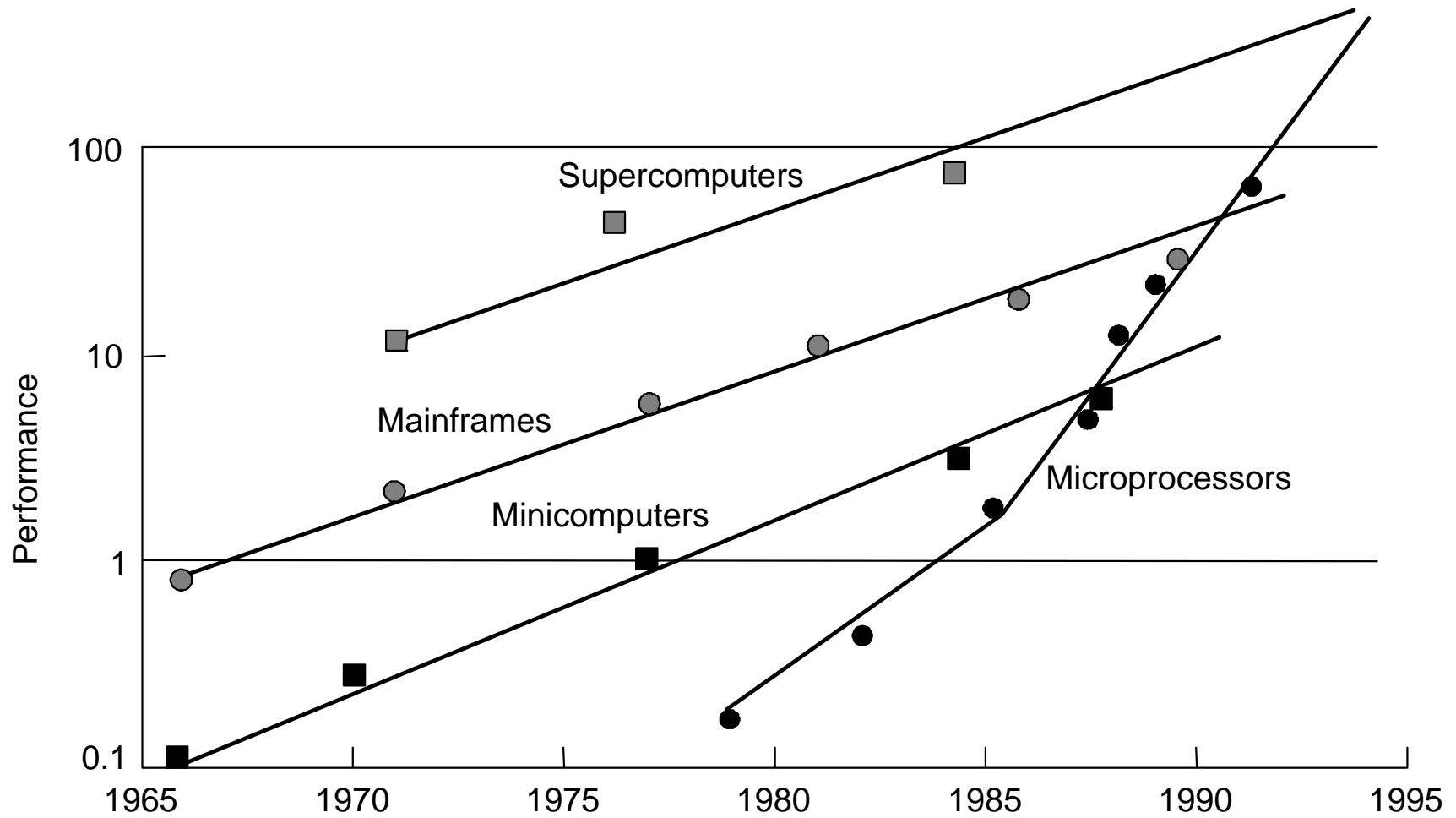
Desktop also uses multithreaded programs, which are a lot like parallel programs

Demand for improving throughput on sequential workloads

- Greatest use of small-scale multiprocessors

Solid application demand exists and will increase

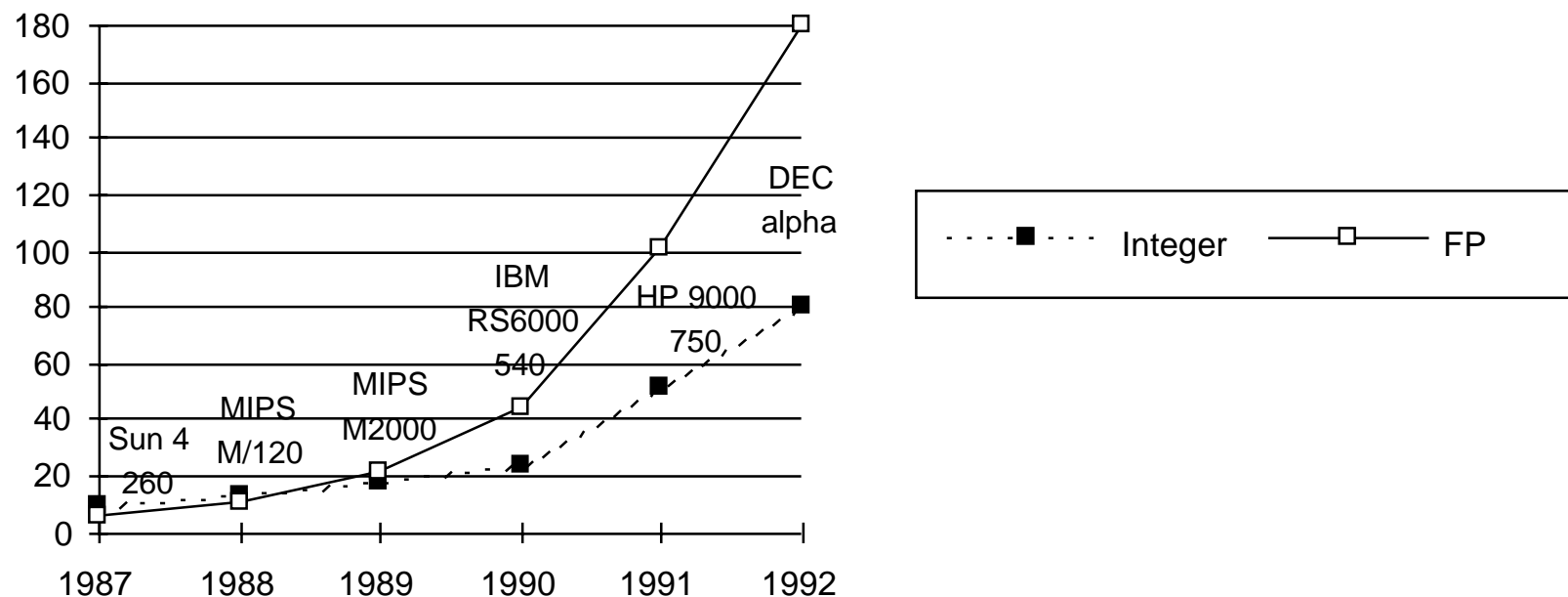
# Technology Trends



The natural building block for multiprocessors is now also about the fastest!

# General Technology Trends

- *Microprocessor performance* increases 50% - 100% per year
- *Transistor count* doubles every 3 years
- *DRAM size* quadruples every 3 years
- Huge investment per generation is carried by huge commodity market



- Not that single-processor performance is plateauing, but that parallelism is a natural way to improve it.

# Technology: A Closer Look

Basic advance is *decreasing feature size* ( $\lambda$ )

- Circuits become either faster or lower in power

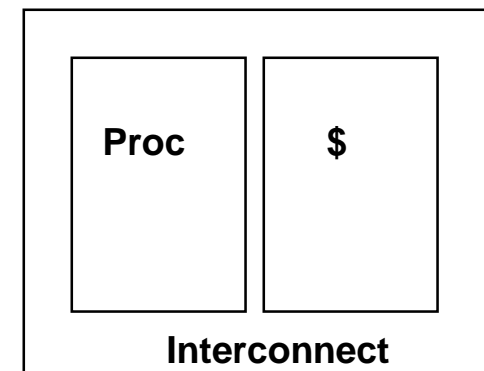
Die size is growing too

- Clock rate improves roughly proportional to improvement in  $\lambda$
- Number of transistors improves like  $\lambda^2$  (or faster)

Performance > 100x per decade; clock rate 10x, rest transistor count

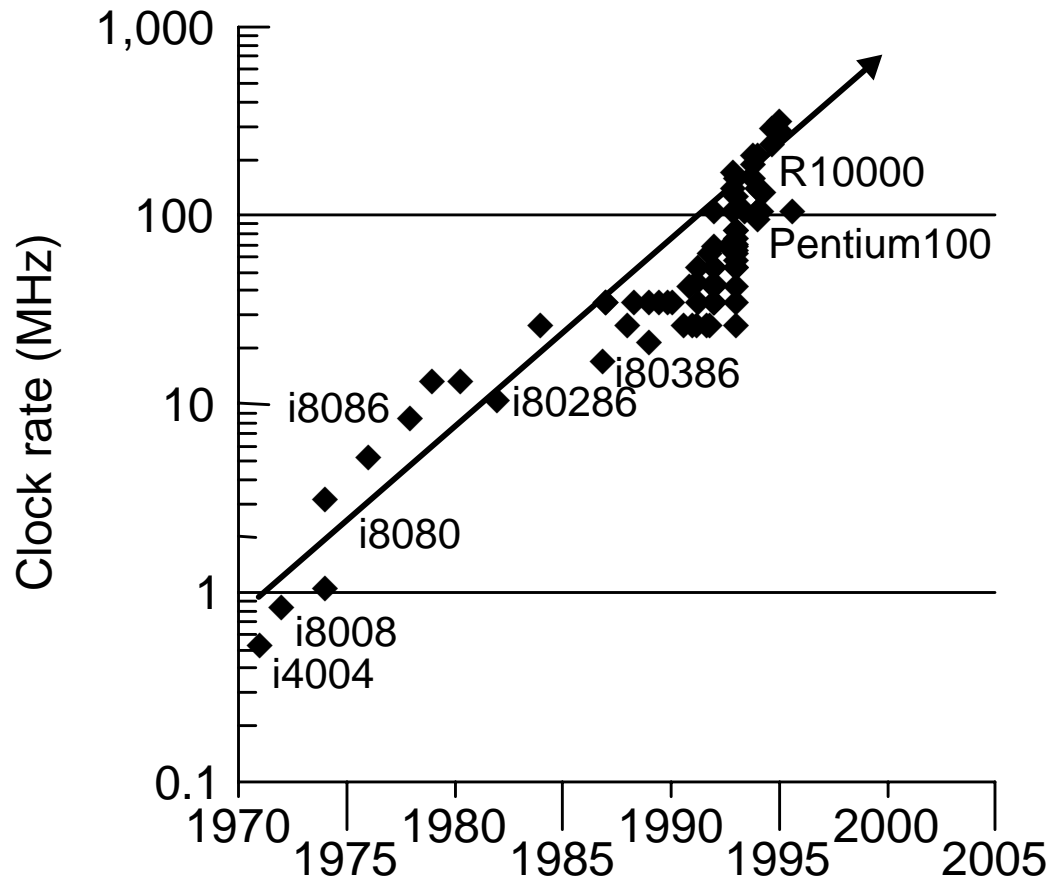
*How to use more transistors?*

- Parallelism in processing
  - multiple operations per cycle reduces CPI
- Locality in data access
  - avoids latency and reduces CPI
  - also improves processor utilization
- Both need resources, so tradeoff



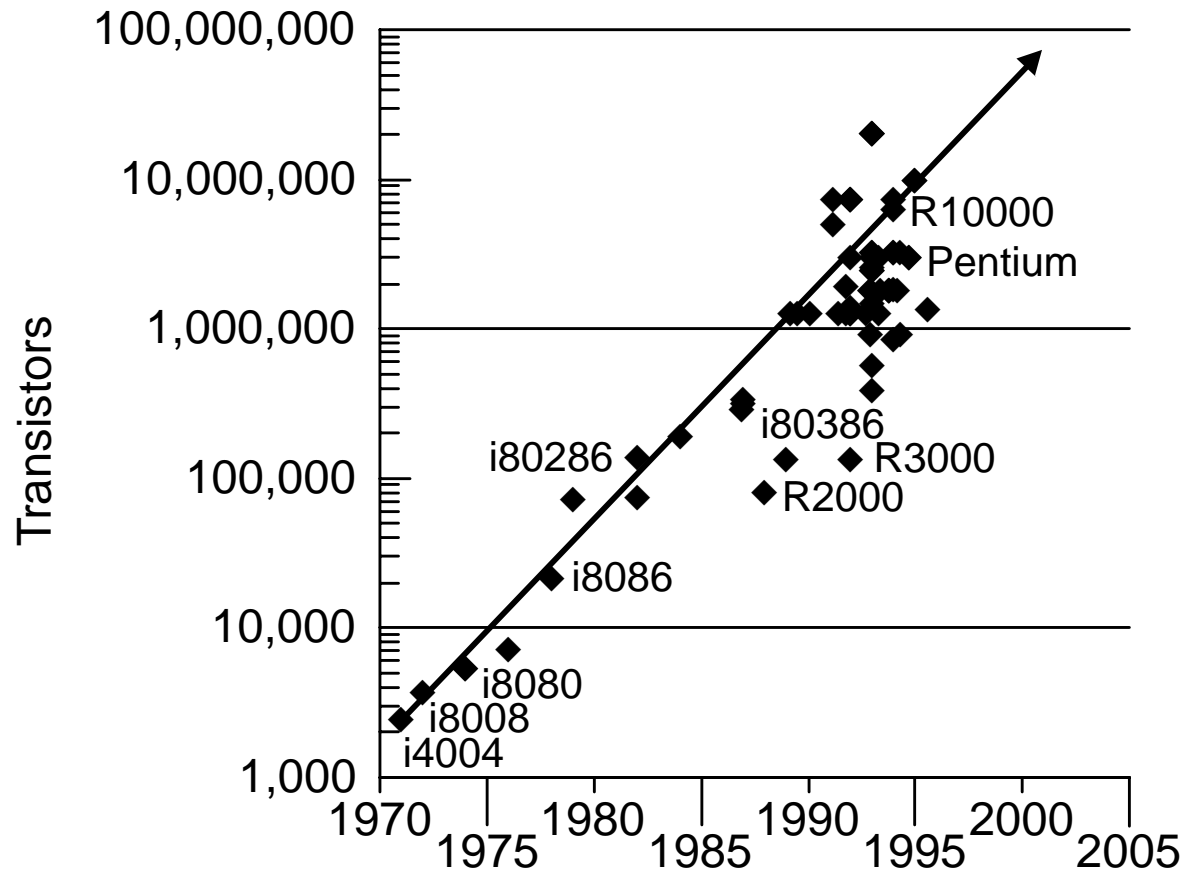
*Fundamental issue is resource distribution, as in uniprocessors*

# Clock Frequency Growth Rate



- 30% per year

# Transistor Count Growth Rate



- 100 million transistors on chip by early 2000's A.D.
- Transistor count grows much faster than clock rate
  - 40% per year, order of magnitude more contribution in 2 decades

# Similar Story for Storage

Divergence between memory capacity and speed more pronounced

- Capacity increased by 1000x from 1980-95, speed only 2x
- Gigabit DRAM by c. 2000, but gap with processor speed much greater

Larger memories are slower, while processors get faster

- Need to transfer more data in parallel
- Need deeper cache hierarchies
- How to organize caches?

Parallelism increases effective size of each level of hierarchy, without increasing access time

Parallelism and locality within memory systems too

- New designs fetch many bits within memory chip; follow with fast pipelined transfer across narrower interface
- Buffer caches most recently accessed data

Disks too: Parallel disks plus caching

# Architectural Trends

Architecture translates technology's gifts to performance and capability

Resolves the tradeoff between parallelism and locality

- Current microprocessor: 1/3 compute, 1/3 cache, 1/3 off-chip connect
- Tradeoffs may change with scale and technology advances

Understanding microprocessor architectural trends

- Helps build intuition about design issues or parallel machines
- Shows fundamental role of parallelism even in “sequential” computers

Four generations of architectural history: tube, transistor, IC, VLSI

- Here focus only on VLSI generation

Greatest delineation in VLSI has been in type of parallelism exploited

# Architectural Trends

Greatest trend in VLSI generation is increase in parallelism

- Up to 1985: bit level parallelism: 4-bit -> 8 bit -> 16-bit
  - slows after 32 bit
  - adoption of 64-bit now under way, 128-bit far (not performance issue)
  - great inflection point when 32-bit micro and cache fit on a chip
- Mid 80s to mid 90s: instruction level parallelism
  - pipelining and simple instruction sets, + compiler advances (RISC)
  - on-chip caches and functional units => superscalar execution
  - greater sophistication: out of order execution, speculation, prediction
    - to deal with control transfer and latency problems
- Next step: thread level parallelism

# Architectural Trends: ILP

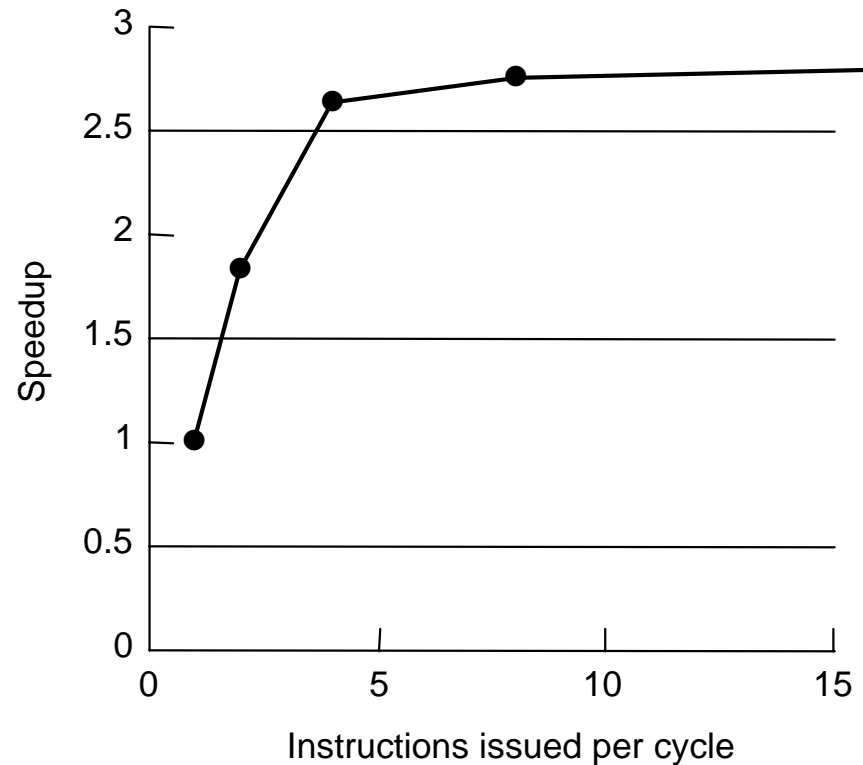
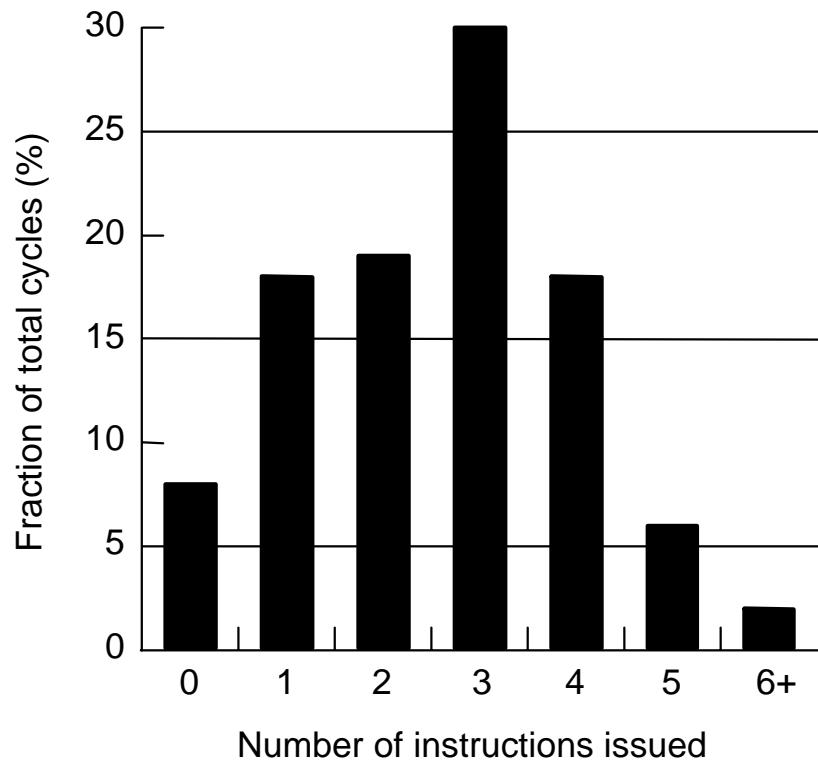
- Reported speedups for superscalar processors

• Horst, Harris, and Jardine [1990] .....	1.37
• Wang and Wu [1988] .....	1.70
• Smith, Johnson, and Horowitz [1989] .....	2.30
• Murakami et al. [1989] .....	2.55
• Chang et al. [1991] .....	2.90
• Jouppi and Wall [1989] .....	3.20
• Lee, Kwok, and Briggs [1991] .....	3.50
• Wall [1991] .....	5
• Melvin and Patt [1991] .....	8
• Butler et al. [1991] .....	17+

- Large variance due to difference in

- application domain investigated (numerical versus non-numerical)
- capabilities of processor modeled

# ILP Ideal Potential



- Infinite resources and fetch bandwidth, perfect branch prediction and renaming
  - real caches and non-zero miss latencies

# Homework 1 for CSE613

Assigned Wed 10 Sept 2003

**Due Wed 13 Sept 2003**

Exercise 1.4 page 72:

- 1.4 Given a histogram of available parallelism such as that shown in Figure 1.7, where  $f_i$  is the fraction of cycles on an ideal machine in which  $i$  instruction issue, derive a generalization of Amdahl's Law to estimate the potential speedup on a  $k$ -issue superscalar machine. Apply your formula to the histogram data in Figure 1.7 to produce the speedup curve shown in that figure.

Histogram shows fractions of cycles while running the program on a theoretical infinite-issue machine, so limit on number of instructions issued per cycle are those inherent to the dependencies of the program.