COS 598A:
Parallel Computer Architecture

Instructor:	J.P. Singh, jps@cs
Office:	423 CS Bldg, 258-5329
Class Hours:	TTh 1:30-2:50
Office Hours:	Door, what's that?
Prerequisite:	Uniprocessor Architecture
Assignments:	3 Programs, some homeworks

Text: Draft of *Paralllel Computer Architecture*, by instructor with David Culler and Anoop Gupta.

Goal: In-depth understanding of modern parallel architecture---fundamental design issues and engineering tradeoffs---as well as parallel programs and performance evaluation.

Course Outline

• Fundamentals	1 week
• Parallel Programs and Performance Evaluation	2
• Small-scale Shared Memory	2
• Large-scale Distributed Memory	1
• Large-scale Shared Address Space	1.5
• Latency Tolerance	1.5
• Network Design	1
• Synchronization	1
• Input/Output	0.5

Fundamentals

Fundamentals

What is Parallel Architecture?

Why Parallel Architecture?

Design Issues

Current Parallel Architectures

Design Issues Revisted

Historical Perspective

What is Parallel Architecture?

- A Parallel Computer is a collection of processing elements that cooperate to solve large problems them fast

- Some broad issues:
  - Resource Allocation:
    - how large a collection?
    - how powerful are the elements?
    - how much memory?
  - Data access, Communication and Synchronization
    - how do the elements cooperate and communicate?
    - how are data transmitted between processors?
    - what are the abstractions and primitives for cooperation?
  - Performance and Scalability
    - how does it all translate into performance?
    - how does it scale?

Why Study Parallel Architecture?

- *Role of a computer architect:*

  To design and engineer the various levels of a computer system to maximize <u>performance</u> and <u>programmability</u> within limits of <u>technology</u> and <u>cost</u>.

- Parallelism:
  - provides alternative to faster clock for performance
  - applies at all levels of system design
  - is a fascinating perspective from which to view architecture
  - is increasingly central in information processing

Why Study it Today?

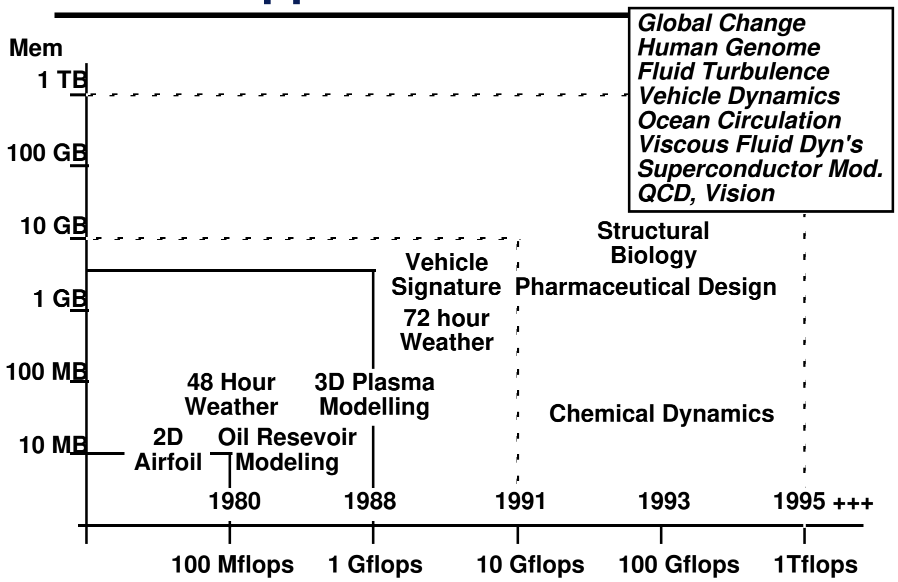
- History: diverse and innovative organizational structures, often tied to novel programming models
- Rapidly maturing under strong technological constraints
  - The “killer micro” is ubiquitous
  - laptops and supercomputers are fundamentally similar!
  - technological trends cause diverse approaches to converge
- Technological trends make parallel computing inevitable
  - in the mainstream
- We need to understand the fundamental principles and design tradeoffs, not just taxonomies
  - Naming, Synchronization, Latency, Bandwidth

Inevitability of Parallel Computing

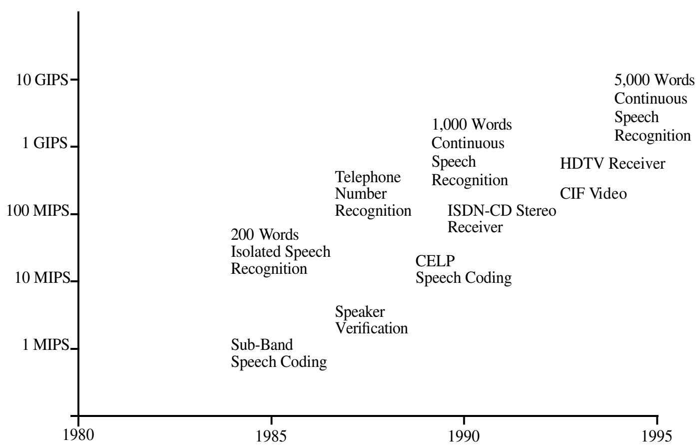
- **Application demands:** Our insatiable need for computing cycles
  - *Scientific computing*: CFD, Biology, Chemistry, Physics, ...
  - *General-purpose computing*: Video, Graphics, CAD, Databases, TP...
- **Trends in general-purpose high-performance computing:**
  - *Technology:*
    - # of transistors on chip growing rapidly
    - Clock rates expected to go up only slowly
  - *Architecture:*
    - instruction-level parallelism valuable but limited

  ==> Coarser-level parallelism, as in MPs, the most viable approach
- **Current trends**:
  - today’s microprocessors have multiprocessor support
  - servers and workstations becoming MPs: SUN, SGI, DEC, ... COMPAQ!...
  - tomorrow’s microprocessors are multiprocessors

Application Demand



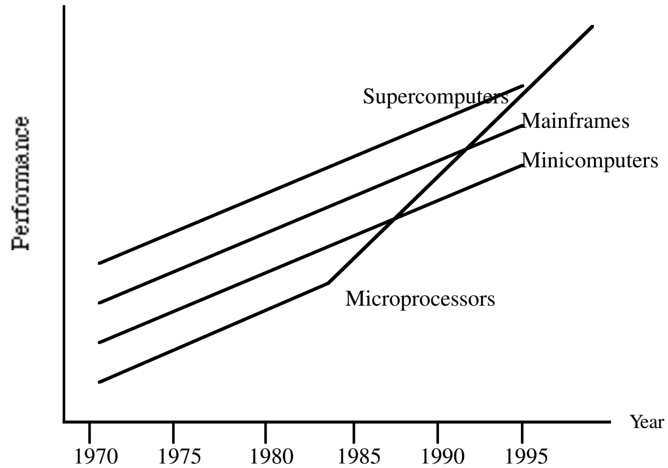
Appls: Speech and Image Processing



Also CAD, Databases, . . .

100 processors gets you 10 years, 1000 gets you 20 !

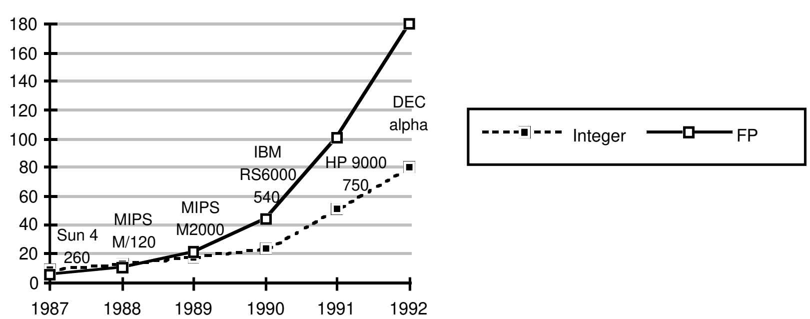
Technology Trends



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

General Technology Trends

- **Microprocessor performance** increases 50% - 100% per year
- **Transitor 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

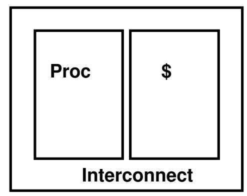
- Basic advance is *decreasing feature size* (λ)
  - **circuits become either faster or lower in power**
- Die size is growing too

=> clock rate improves roughly proportional to improvement in λ

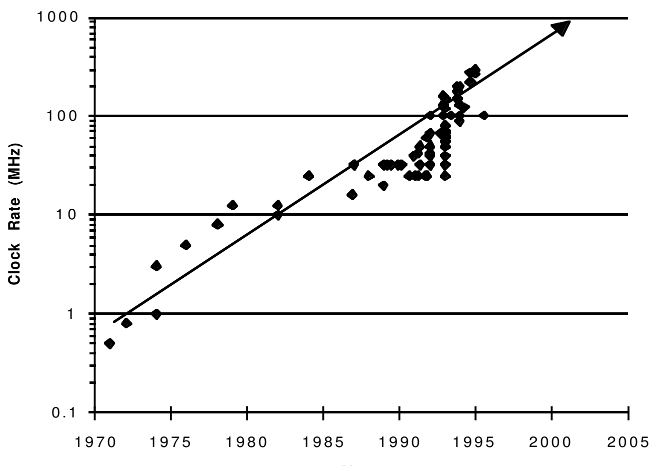
=> number of transistors improves like the square of λ (or faster)

How to use more transistors?

- **parallelism**
  - » **multiple operations per cycle reduces CPI**
- **locality**
  - » **avoids latency and reduces CPI**
  - » **also improves processor utilization**
- Both need resources (fundamental tradeoff)



Clock Frequency Growth Rate

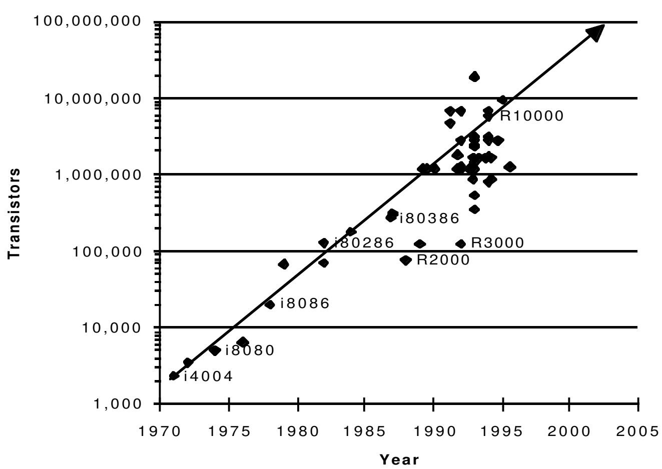


Clock Rate and Performance

- Performance > 100x per decade
- Clock rate 10x per decade

⇒ rest (>10x per decade) is from transistor count
- parallelism
- locality

Transistor Count Growth Rate



• We should have a 100 million transistors on chip by 2003

• Transistor count grows much faster than clock rate

Architectural Trends: ILP

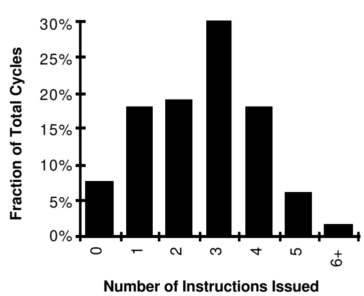
• Reported speedups for superscalar processors

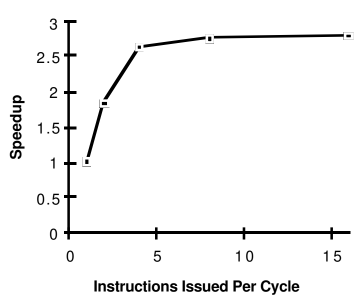
- Horst, Harris, and Jardine [1990] 1.37
- Wang and Wu [1988] 1.70
- Smith, Johnson, and Horowitz [1989] 2.30
- Murakami et al. [1989] 2.55
- Chang et al. [1991] 2.90
- Jouppi and Wall [1989] 3.20
- Lee, Kwok, and Briggs [1991] 3.50
- Wall [1991] 5
- Melvin and Patt [1991] 8
- Butler et al. [1991] ....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

• Large variance due to difference in

- application domain investigated (numerical vs. non-numerical)
- capabilities of machine modelled

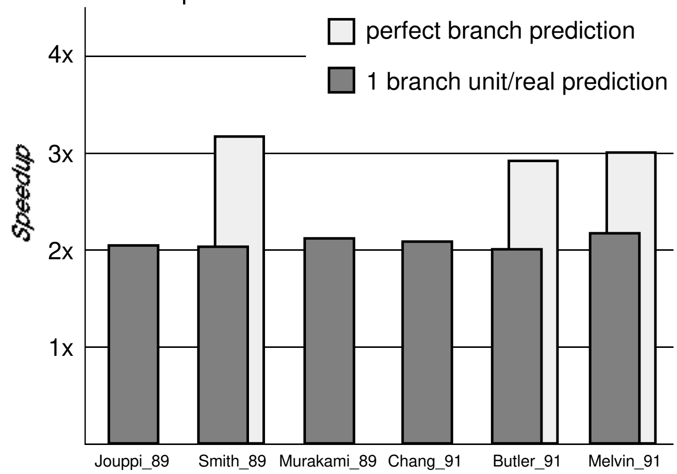
ILP Ideal Potential





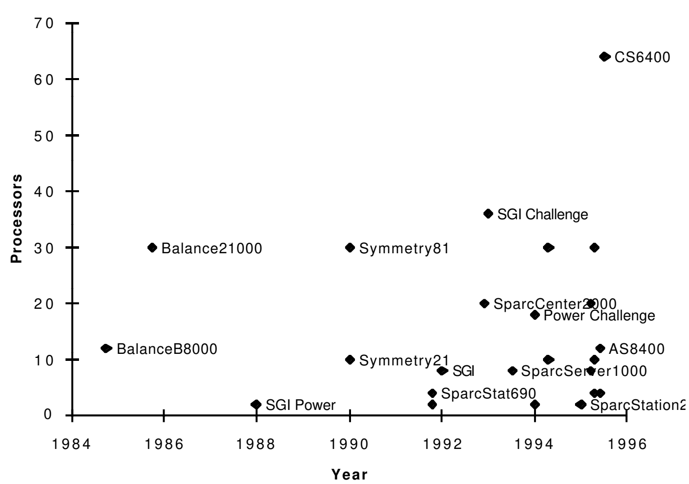
Results of ILP Studies

• Concentrate on parallelism for 4-issue machines



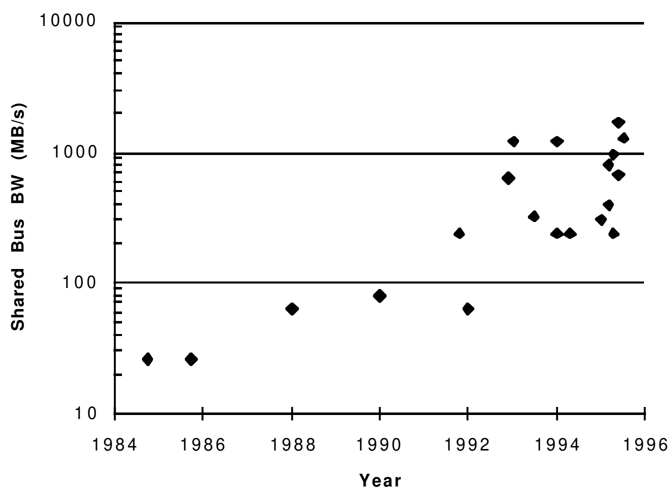
• **Realistic studies show only 2-fold speedup**

Architectural Trends: Bus-based MPs



No. of processors in fully configured commercial shared-memory systems

Bus Bandwidth



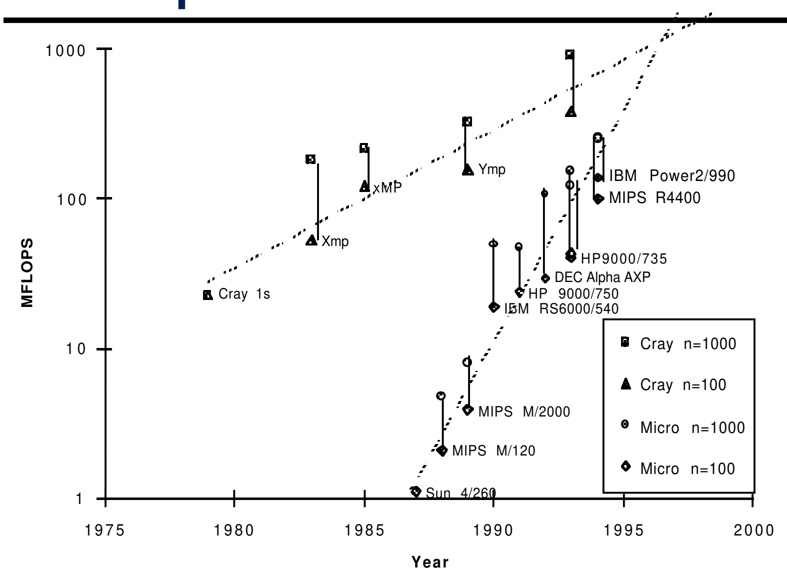
Performance and Economics

- Scientific computing:
  - proving ground for innovative architectures
  - uPs have made massive gains in FP performance
    - clock rates (e.g., DEC-alpha 200MHz)
    - pipelined FPUs (e.g., mult-add every cycle)
    - effective use of caches (e.g., automatic blocking)
- These commodity uPs are CHEAP
    - development cost is tens of millions of dollars
    - BUT, many more are sold as compared to supercomputers
  - Crucial to take advantage of the investment, and use the commodity building block

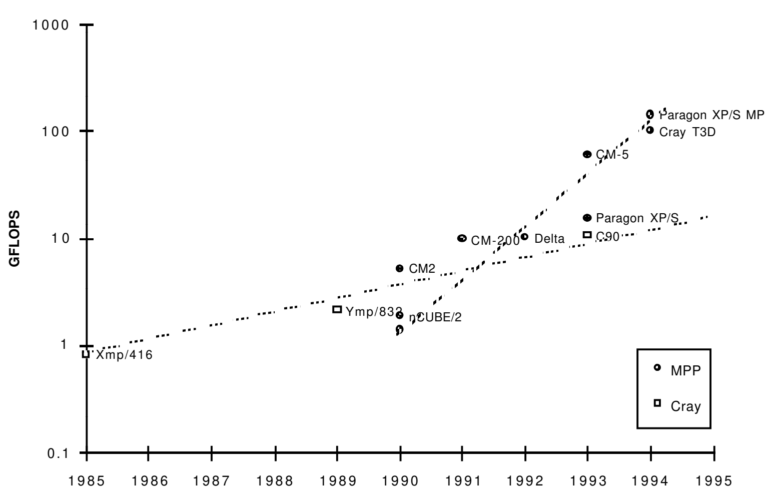
==> Large-scale multiprocessors will replace traditional supercomputers

- Trend seen in products from Cray, intel, TMC, IBM, Convex, ...

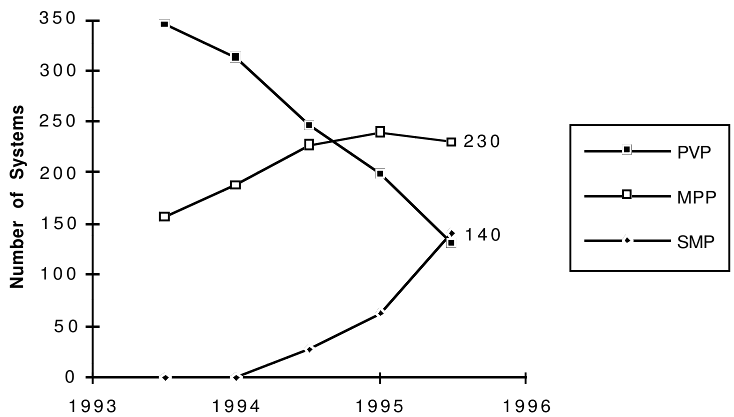
Raw Uniproc. Performance: LINPACK



Raw Performance: LINPACK



500 Fastest Computers



Summary: Why Parallel Architecture?

- Increasingly attractive
  - economics, technology, architecture, application demand
- Increasingly central and mainstream
- Parallelism exploited at many levels
  - ILP
  - MP Servers
  - MPPs
- Wide range of parallel architectures make sense
  - different cost, performance and scalability