Advanced Parallel Architecture

Lesson 1

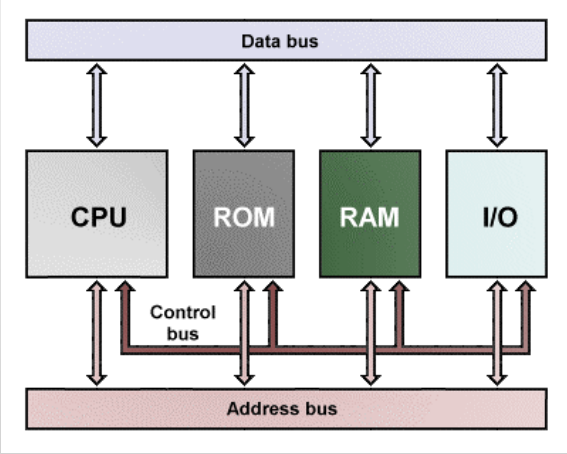
Annalisa Massini - 2016/2017

Introduction

Aim of course

- The aim of this course is
  - to acquire an **understanding** and **appreciation** of a computer system
  - to learn to harness **parallelism to sustain performance improvements**

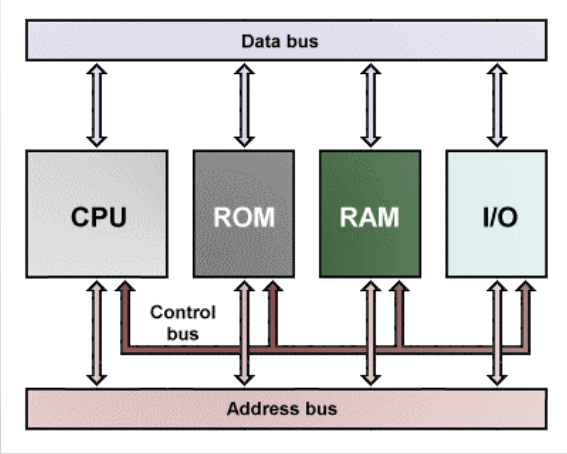
starting from the knowledge of the computer architecture from undergraduate courses



Aim of course

In fact the design of *parallel algorithms* and the study *of strategies for problem decomposition* **are sustained by**

- A deep **knowledge of the computer architecture**,
- A careful **use of parallelism** and
- The **performance analysis**



Syllabus

- Von Neumann's Architecture and its limitations.
- Instruction pipeline. Arithmetic operations pipeline. Number representations.
- Classification of parallel architectures: Flynn’s Taxonomy and other classifications. Forms of parallelism. SIMD and MIMD architectures.
- Interconnection topologies and interconnection networks.
- Vector processors.

Syllabus

- Non Von Neumann Architectures. Dataflow architecture.
- Cache Coherence: Snooping Protocols and Directory-based Protocols. Memory Consistency. Message Passing Systems.
- Manycore Architectures: GPU (and CUDA).
- Performance metrics and measurement. Amdahl's Law.
- Performance optimization: work distribution and load balance, locality, communication.

Textbooks

- *Parallel Computer Architecture: A Hardware/Software Approach*

  D.E. Culler, J. P. Singh, A. Gupta - Morgan Kaufmann, 1998

- *Multicore and GPU Programming An Integrated Approach*

  G. Barlas - Morgan Kaufmann, 2014

- *Computer Architecture: A Quantitative Approach*

  J. L. Hennessy, D. A. Patterson - Morgan Kaufmann, 2011

- *Programming Massively Parallel Processors: A Hands-on Approach*

  D. B. Kirk, W-M. W. Hwu - Morgan Kaufmann, 2010

Lessons and exams

- Aula Alfa - Tuesdays and Thursday, 12:00 to 14:00
- Two mid-term exams or a final exam

  Mid-term and final exams consist in (written) test and exercises
- Project or oral exam

Lessons and exams

- Project proposals will be presented in collaboration with Tiziana Castrignanò from Cineca

- **CINECA** is the largest Italian computing centre, one of the most important worldwide
- The **High Performance Systems department (SCAI: Super Computing Applications and Innovation)** offers support to scientific and technological research through supercomputing and its applications

Today's Goal

- Answer your questions about the course

- Introduce you to Parallel Computer Architecture
- Provide you a sense of the trends that shape the field

Lecture from*:*

Parallel Computer Architecture: A Hardware/Software Approach - Chapter 1

D. E. Culler, J. P. Singh, A. Gupta

Motivations to Parallel Architectures

- **Parallel computer architecture** forms an important thread in the evolution of computer architecture
- It rooted in the beginnings of computing exploits *advancement* over what the *base technology* can provide
- Parallel computer designs have demonstrated a rich *diversity of structure*, usually motivated by specific higher level parallel programming models

Motivations to Parallel Architectures

- We have seen an explosive **growth** in **performance** and **capability** of computer systems for decades
- The **speed** with which computer can process information has been **increasing exponentially** over the time
- The advance of the underlying VLSI technology allowed
  - Larger and larger numbers of components to fit on a chip
  - Clock rates to increase

Motivations to Parallel Architectures

- The leading character is **parallelism**
- A larger volume of available resources means that more operations can be done at once, in **parallel**
- **Parallel computer architecture** is about organizing these resources so that they work well together
- **Computers of all types** have harnessed parallelism more and more effectively to gain performance from the technology

Motivations to Parallel Architectures

- The other key character is **storage**
- The **data** that is operated on at an ever faster rate must be held somewhere in the machine
- Thus, the story of parallel processing is deeply intertwined with data *locality* and *communication*

Motivations to Parallel Architectures

Role of a computer architect:

- To design and engineer the various levels of a computer system to maximize *performance* and *programmability* within limits of *technology* and *cost*

Parallelism:

- Provides an *interesting perspective* from which to understand computer architecture
- Provides *alternative to faster clock* for performance
- Applies at *all levels of system design*
- Is *increasingly central* in information processing

Motivations to Parallel Architectures

- **A *parallel computer* is a collection of processing elements** that cooperate to solve large problems fast

- This simple definition raises *many questions*

Motivations to Parallel Architectures

- **A *parallel computer* is a collection of processing elements** that cooperate to solve large problems fast

- This simple definition raises *many questions*

- Resource Allocation:
  - how large a collection?
  - how powerful are the elements?
  - how much memory?

Motivations to Parallel Architectures

- **A *parallel computer* is a collection of processing elements** that cooperate to solve large problems fast

- This simple definition raises *many questions*

- 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?

Motivations to Parallel Architectures

- **A *parallel computer* is a collection of processing elements** that cooperate to solve large problems fast

- This simple definition raises *many questions*

- Performance and Scalability
  - how does it all translate into performance?
  - how does it scale?

Motivations to Parallel Architectures

- To understand parallel architectures it is important to examine*:*
  - the principles of computer design at the processor level
  - the design issues present for each of the system components
    - memory systems
    - processors
    - networks
  - the relationships between these components
  - the division of responsibilities between hardware and software

Motivations to Parallel Architectures

- There is a **variety of important architectural styles** which give different contributes to the understanding of parallel machines

- Within this diversity of design, we can individuate **a common set of design principles and trade-offs**, driven by the same advances in the underlying technology

Motivations to Parallel Architectures

- Traditionally, **computer architecture**, technology, and applications evolve together and have very strong interactions
- Parallel computer architecture is no exception
- A *new dimension* is added to the design space: ***the number of processors***
- The design is even more strongly driven by the demand for **performance** at acceptable cost

Motivations to Parallel Architectures

- Whatever the performance of a single processor at a given time → **higher performance** can be achieved by utilizing many such processors

- How much additional performance is gained and at what additional cost *depends on a number of factors*

Motivations to Parallel Architectures

Observation

- The advantages of using small, inexpensive, low power, mass produced **processors** have been used **as the building blocks** for computer systems with many processors

- **But**, usually the performance of the processor best suited to parallel architecture was **far** behind that of the *fastest single processor system*

- **This is no longer so**, with exceptions

Motivations to Parallel Architectures

- Parallel machines have been built at various scales since the earliest days of computing
- But the approach is more viable today than ever before
- In fact the basic **processor** building block is **better suited** to the job

Motivations to Parallel Architectures

Observation

Change, even dramatic change, is the norm in computer architecture

- The continuing process of change has profound implications for the study of computer architecture
- We need to understand not only *how things are*, but ***how they might evolve, and why***

- **But**, the prevalence of change suggests that one should be ***cautious in extrapolating toward the future***

Motivations to Parallel Architectures

- In the “late 1990s”
  - the **single-chip microprocessor** started to dominate every sector of computing
  - **parallel computing** took hold in many areas of mainstream computing

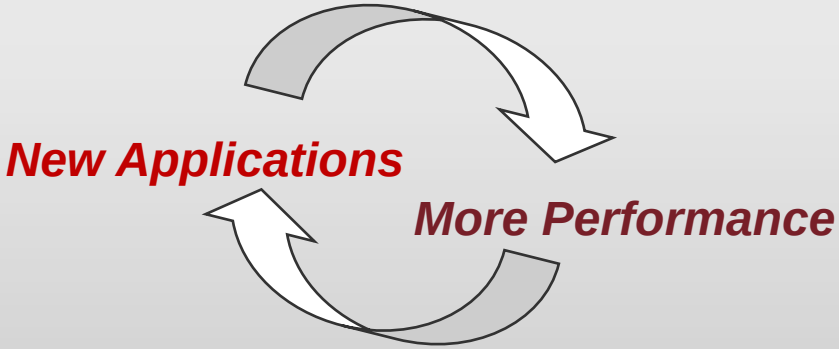
- We will explore:
  - Which **forces** and **trends** are giving parallel architectures an increasingly important role throughout the computing field

Motivations to Parallel Architectures

- We will consider:
  - application demands (for increased performance)
  - technological trends
  - architectural trends
  - economics

Application Trends

- The demand for ever greater application performance is a familiar feature of every aspect of computing
- Application demand for performance fuels advances in hardware, which enables new applications, which...
- We see that this cycle:
  - drives exponential increase in microprocessor performance
  - drives parallel architecture harder
    - most demanding applications



Application Trends

- Prior to the microprocessor era, greater performance was obtained through *exotic circuit technologies* and *machine organizations*

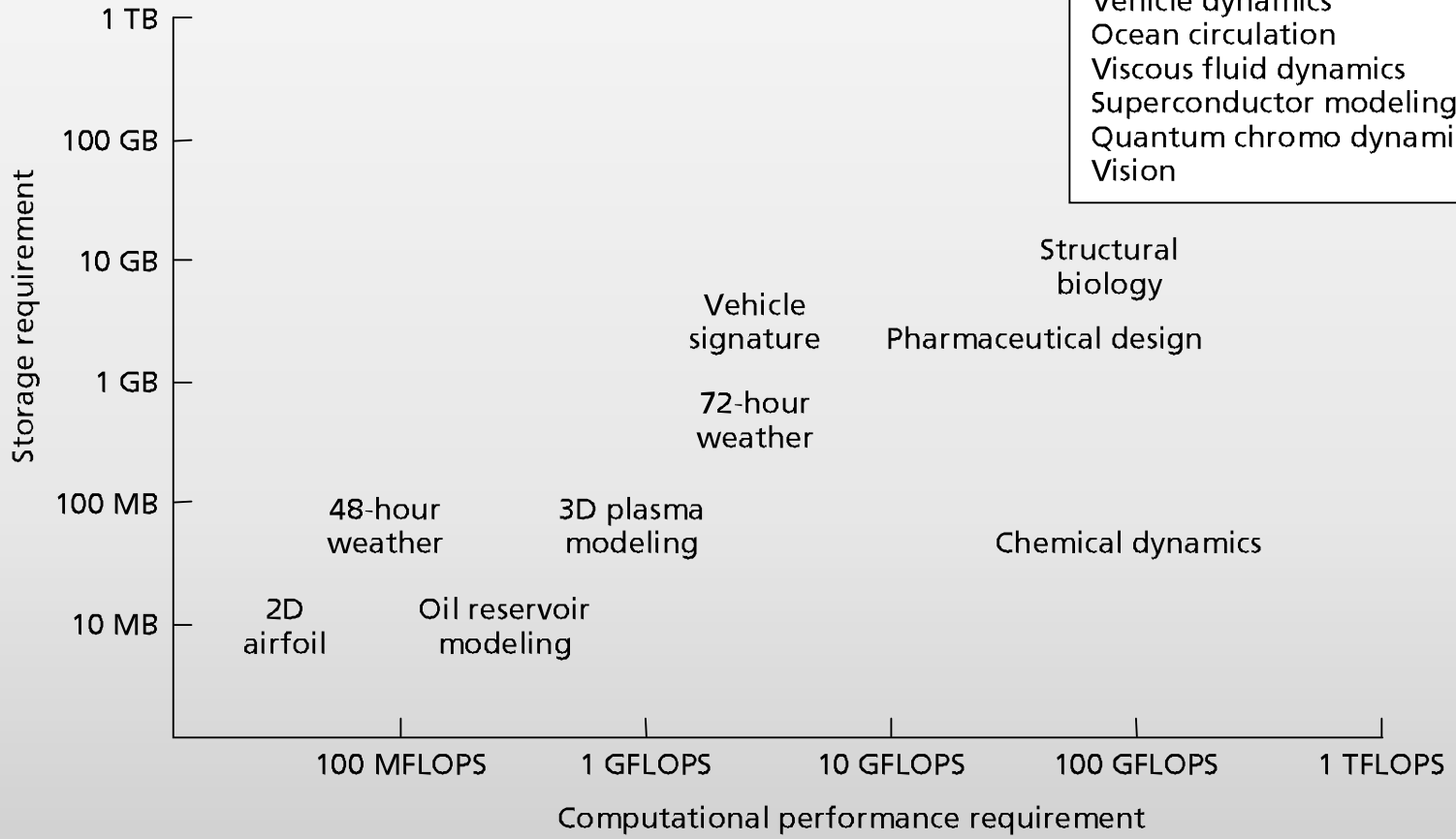
- Then, to obtain performance significantly greater than the state-of-the-art microprocessor, the primary option is **multiple processors**, and the most demanding applications are written as **parallel programs**
- Thus, *parallel architectures* and *parallel applications* are subject to the most acute demands for greater **performance**

Scientific Computing Demand

- The direct reliance on increasing levels of performance is most apparent in the field of computational science and engineering
- Computers are used to simulate physical phenomena that are *impossible* or *very costly* to observe empirically
- Typical examples include:
  - modeling global climate change over long periods
  - the evolution of galaxies
  - the atomic structure of materials
  - the flow of air over surfaces of vehicles
  - the damage due to impacts
  - ……

Scientific Computing Demand

Figure indicates the computational rate and storage capacity required to tackle a number of important science and engineering problems (1993)



Engineering Computing Demand

- Engineering applications for modeling physical phenomena are essential to many industries
  - Petroleum (reservoir analysis)
  - Automotive (crash simulation, drag analysis, combustion efficiency)
  - Aeronautics (airflow analysis, engine efficiency, structural mechanics, electromagnetism)
  - Computer-aided design
  - Pharmaceuticals (molecular modeling)
  - Visualization (in all of the above, entertainment - animation films, architecture - walk-throughs and rendering)
  - Financial modeling (yield and derivative analysis)….

Example

- Let us consider an example from the Grand Challenge program to understand the interaction between applications, architecture, and technology in the context of parallel machines
- A 1995 study examined the effectiveness of a wide range of parallel machines on a variety of applications

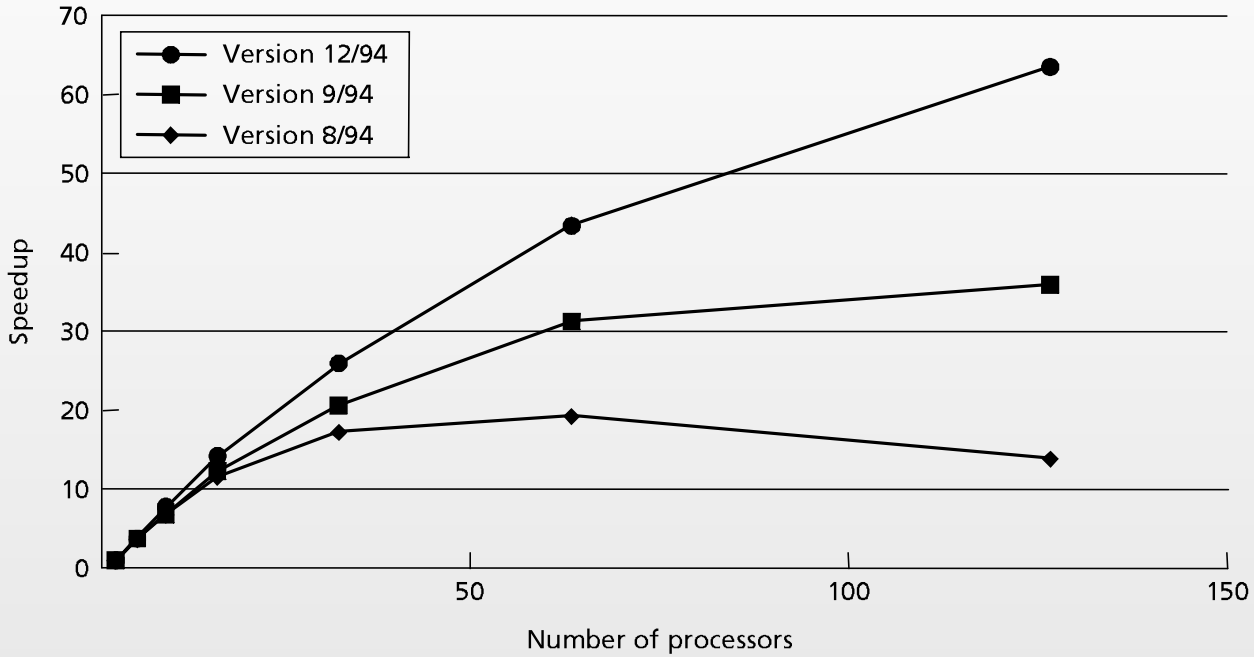
- **AMBER** (Assisted Model Building through Energy Refinement) - a molecular dynamics package
- AMBER is widely used to simulate the motion of large biological models such as proteins and DNA

Example

- The code was developed on Cray vector supercomputers, which employ:
  - custom ECL-based processors
  - large expensive SRAM memories (instead of caches)
  - machine instructions that perform arithmetic or data movement on *vector* of data values

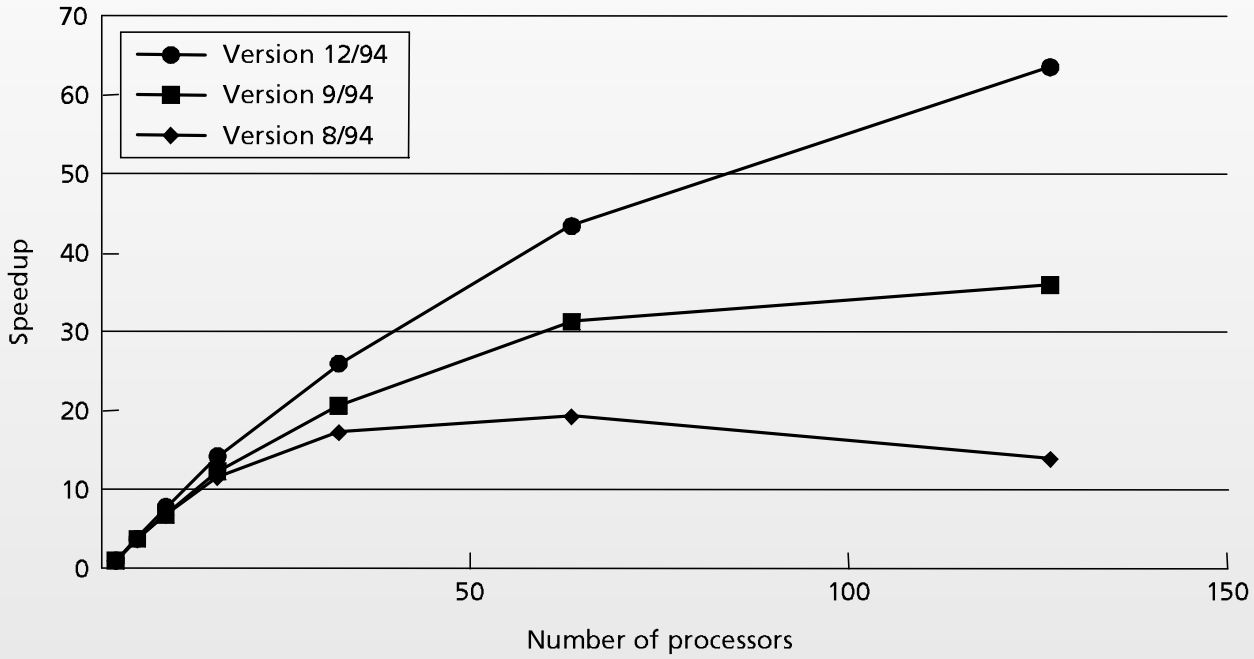
- The test involves the simulation of a protein solvated by water: 99 amino acids, 3,375 water molecules for a total of about 11,000 atoms

Example



- Figure shows the speedup obtained on *three versions* of this code on the Intel Paragon a 128-processor microprocessor-based machine
- **vers. 8/94 -** *initial parallelization* → good speedup for small configurations, but poor speedup on larger configurations

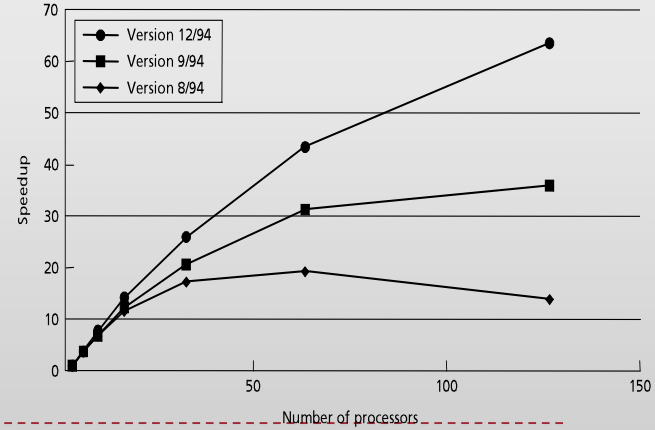
Example



- **vers. 9/94 -** the *balance of work* done by each processor improved the scaling of the application significantly
- **vers. 12/94 -** *optimization of the communication* produced a highly scalable version

Example

- This sort of *learning curve* is quite typical in the parallelization of important applications, as is the *interaction between application and architecture*
- The **application writer** studies the application to understand the demands it places on the available architectures and how to improve its performance on a given set of machines
- The **architect** studies these demands to understand how to make the machine more effective on a given set of applications
- The **end user** of the application enjoys the benefits of both efforts.



Motivations to Parallel Architectures

- The question is:
  - Which **forces** and **trends** are giving parallel architectures an increasingly important role throughout the computing field?

- We have:
  - application demands (for increased performance)
  - technological trends
  - architectural trends
  - economics

Technology Trends

- The critical issues in parallel computer architecture are fundamentally similar to those that we wrestle with in “sequential” computers:

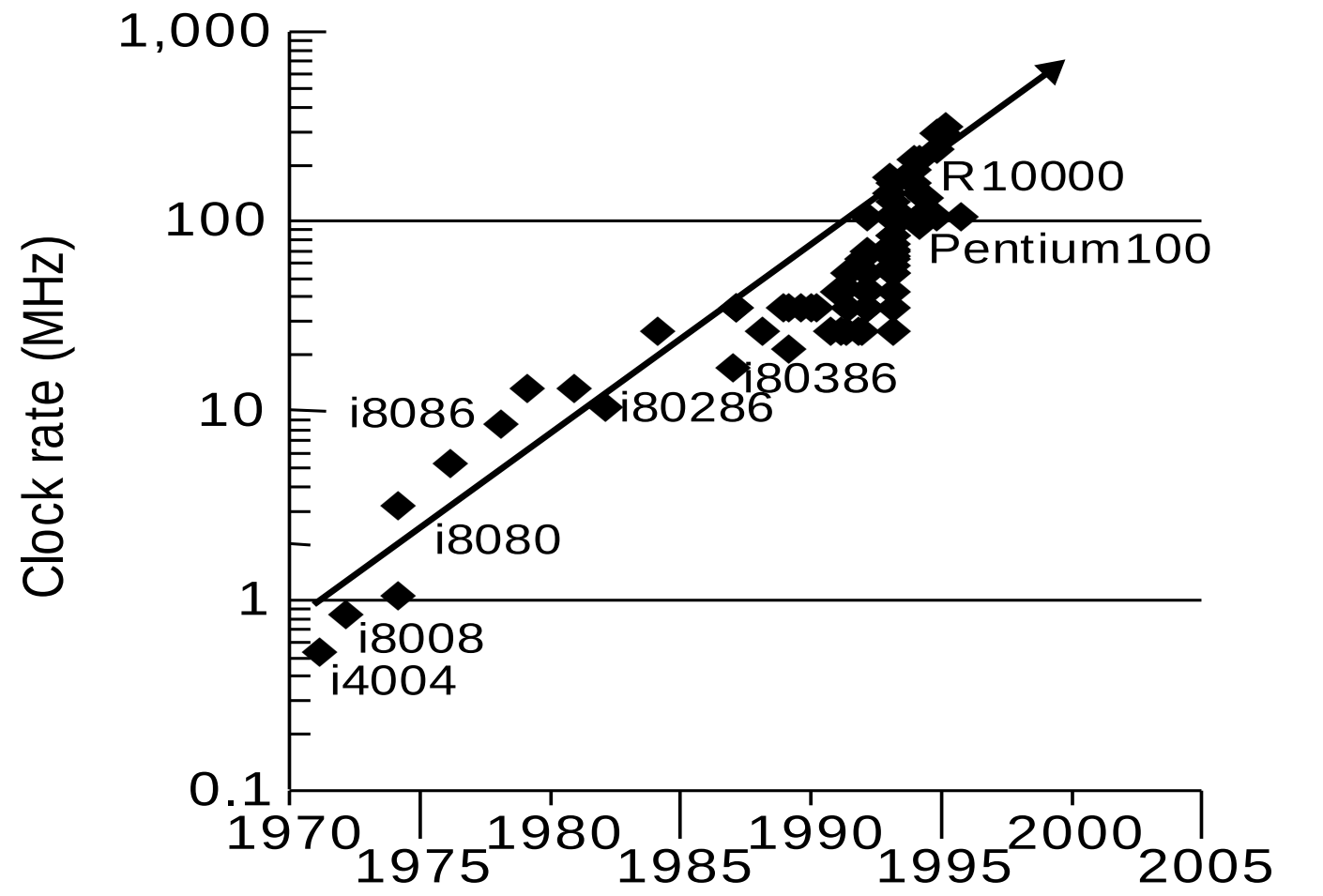
- The problem is how the **resource** budget should be divided up among
  - Functional units that do the work
  - **Caches** to exploit locality
  - Wires to provide **bandwidth**

Technology Trends

- The primary **technological advance** is reduction in the VLSI feature size
- Transistors, gates, and circuits become faster and smaller, so ***more fit in the same area***

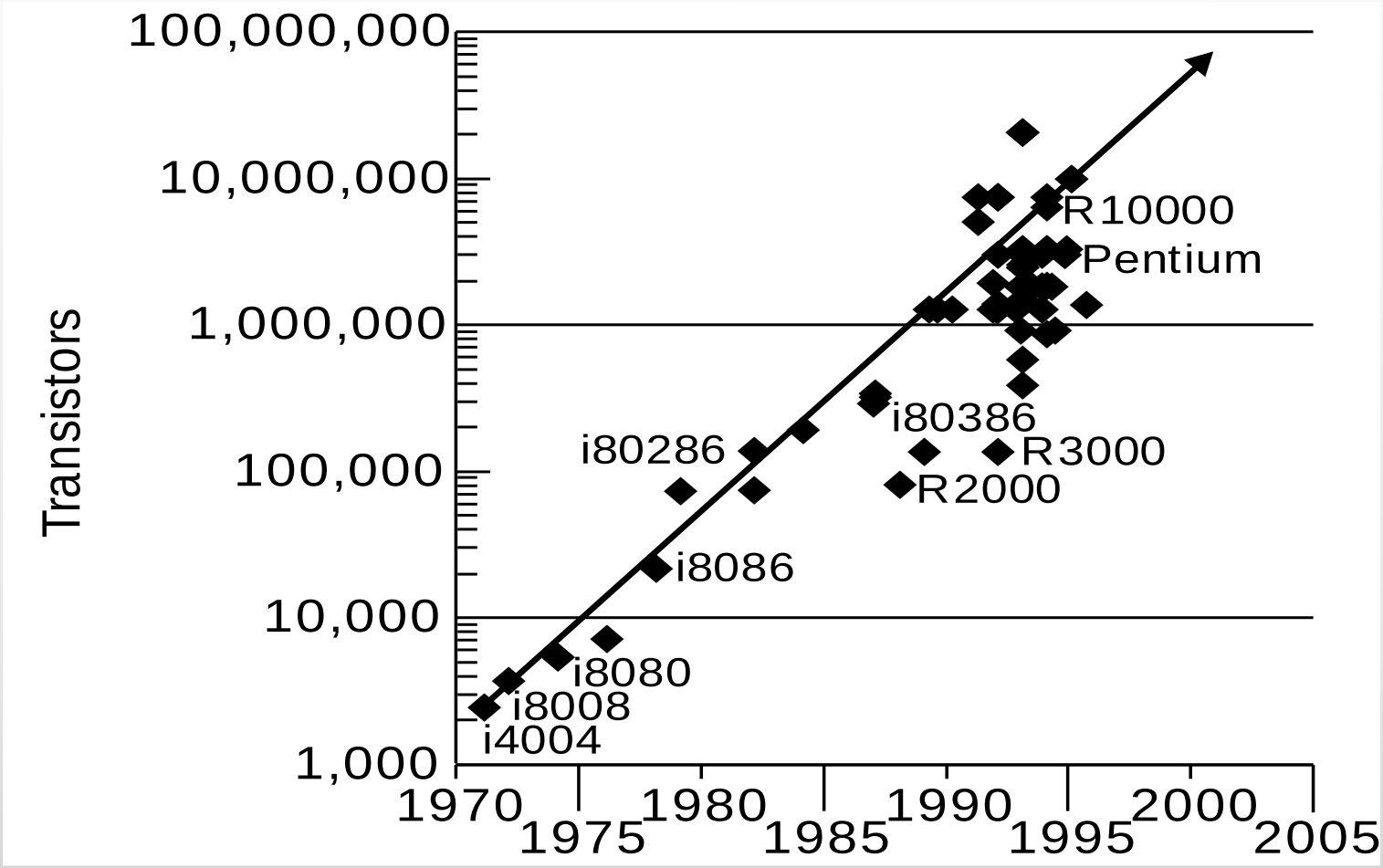
- We can observe that:
  - **Clock rate** improves in proportion to the improvement in feature size
  - The **number of transistors** grows as the square (or faster) due to increasing overall die area
  - The use of many transistors, **parallelism**, contributes more than clock rate to the performance improvement of the single-chip building block

Technology Trends



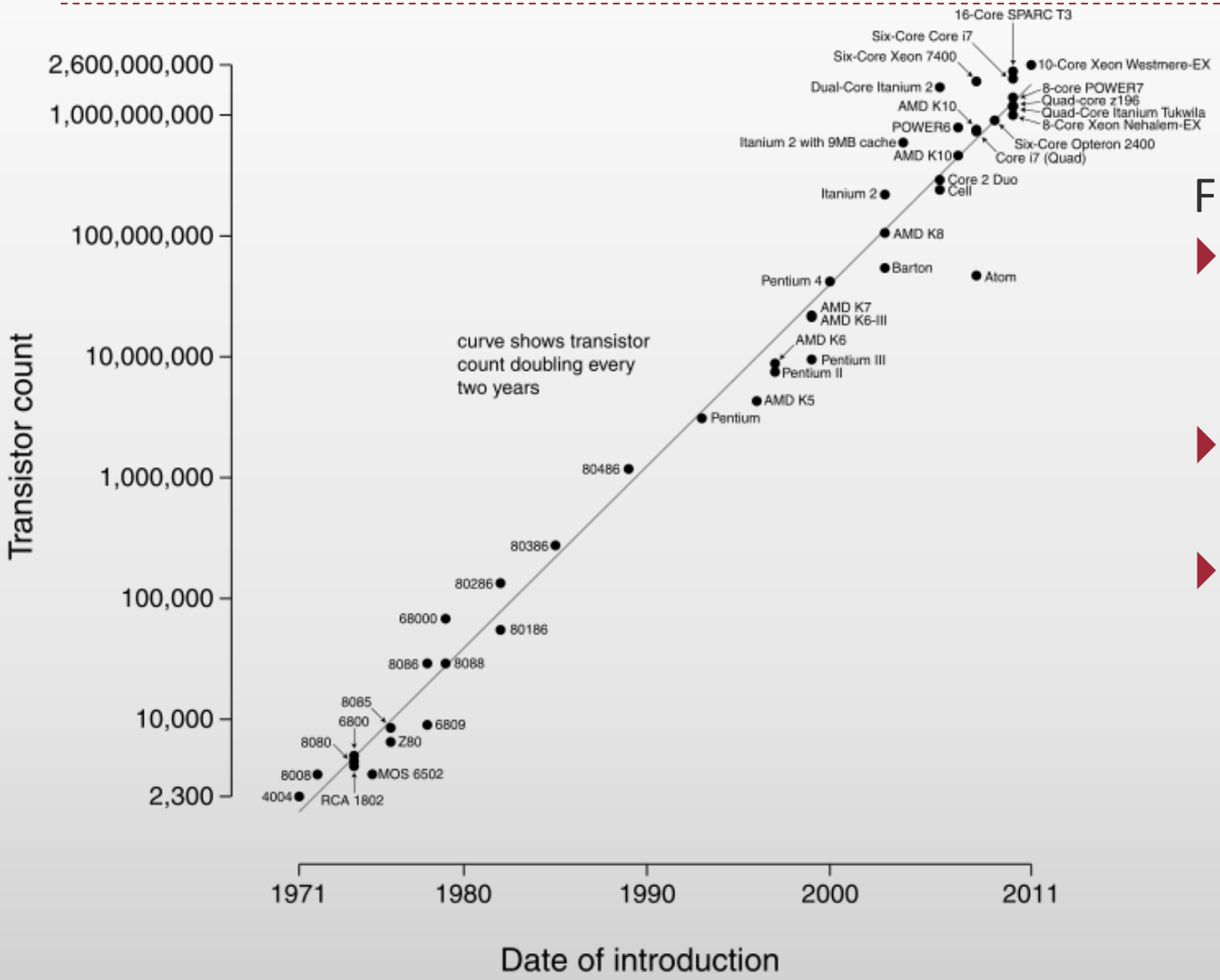
30% per year

Technology Trends



40% per year Transistor count grows much faster than clock rate

Technology Trends

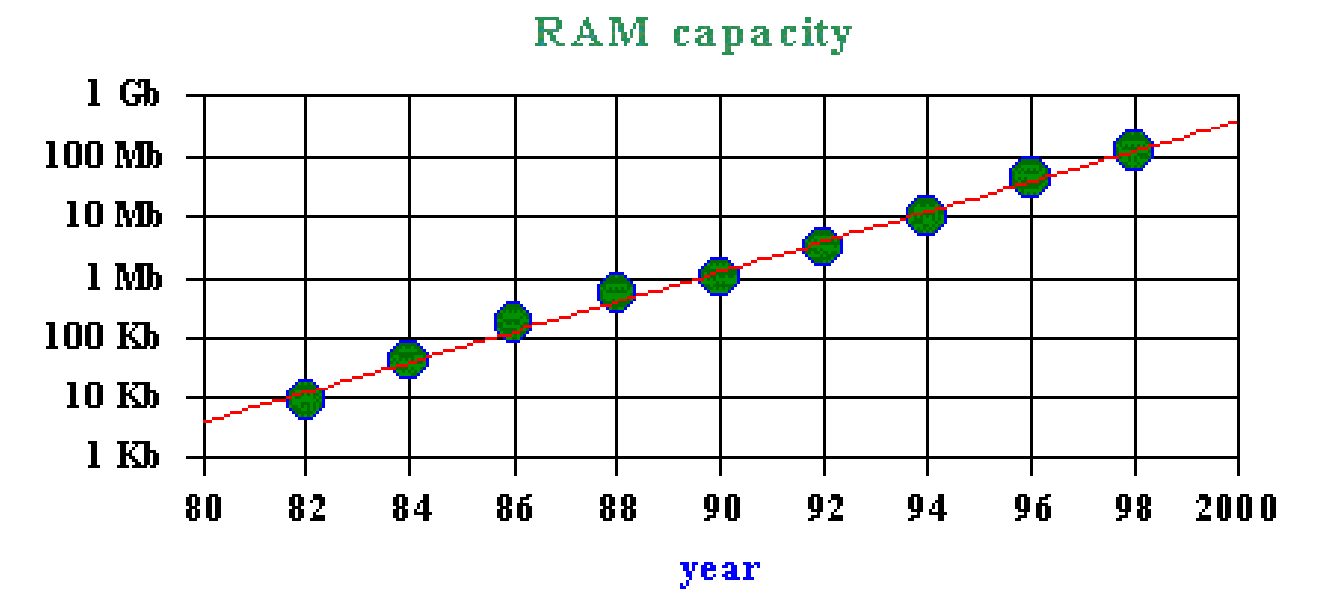


From Wikipedia:

- CPU transistor counts against dates of introduction
- Logarithmic vertical scale
- The line corresponds to exponential growth with transistor count doubling every two years

Technology Trends

- The divergence between capacity and speed is much more pronounced in **memory** technology
- The memory bandwidth demanded by the processor (bytes per memory cycle) is growing rapidly and we have to transfer more data in parallel



Current new PCs (and similar devices) range from around the 2GB to 16GB or more.

Technology Trends

- PCs, workstations, servers (based on conventional DRAMs) are using wider paths into the memory and greater interleaving of memory banks → *parallelism*