# Computer Architecture

Dr: Alshaimaa Mostafa Mohammed

# Lecture Rules

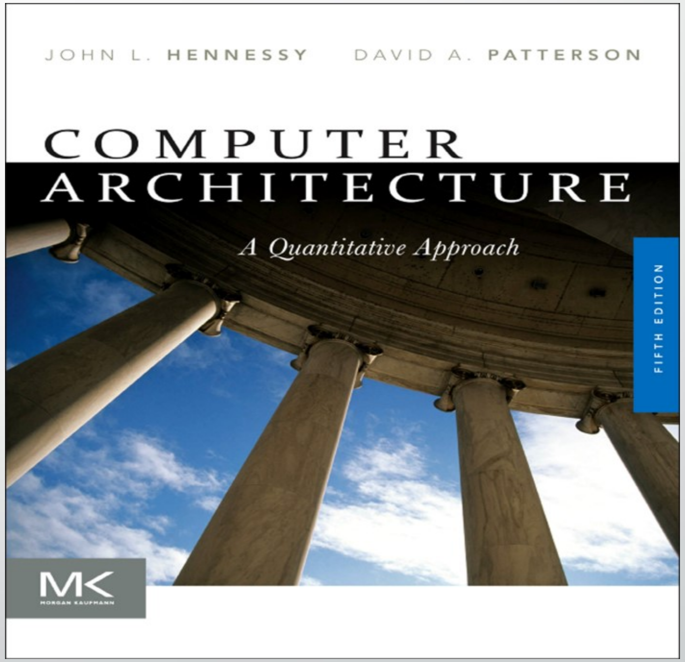
1. Arrive on Time.
2. Turn Off Cell Phones (Silent).
3. If You Have a Question, Ask for Help
4. Do not have private conversations.

# Course Aims

- The Computer Architecture course aims to describe a broad range of architectural designs and to contrast them, highlighting the design decisions they incorporate.

# Resources and References

- **Computer Architecture:** A Quantitative Approach (*Fifth Edition)*
- **John L. Hennessy - David A. Patterson**



# Evaluation

- Total Degree: 150
- Mid Term: 15
- Practical: 30
- Oral: 15
- Final Exam: 90

# Practical Section

Dr Heba

# Lecture 1

# Chapter 1:Fundamentals of Quantitative Design and Analysis

# What is Computer Architecture?

- **Computer architecture** can be defined as a set of rules and methods that describe the functionality, management and implementation of computers.
- To be precise, it is nothing but rules by which a system performs and operates

# What is Computer Architecture?

- Computer architecture provides the functional details and behavior of a computer system.
- It involves the design of the instruction set, the microarchitecture, and the memory hierarchy, as well as the design of the hardware and software components that make up the system.
- Computer Architecture mainly deals with the functional behavior of a computer system and covers the "**What to do?**" part.
- It gives the functional description of requirements, design, and implementation of the different parts of a computer system.

# What is Computer Organization?

- **Computer Organization** refers to the way in which the hardware components of a computer system are arranged and interconnected.
- It implements the provided computer architecture and covers the "**How to do**?" part.
- It just provides information that how operational attributes of a computer system are linked together and help in realizing the architectural specification of the computer.
- It involves the design of the interconnections between the various hardware components, as well as the design of the memory and I/O systems.

# What is Computer Organization?

- **Computer Organization** refers to the way in which the hardware components of a computer system are arranged and interconnected.
- It implements the provided computer architecture and covers the "**How to do**?" part.
- It just provides information that how operational attributes of a computer system are linked together and help in realizing the architectural specification of the computer.
- It involves the design of the interconnections between the various hardware components, as well as the design of the memory and I/O systems.

# Differences between Computer Architecture and Computer Organization

| Key | Computer Architecture | Computer Organization |
|---|---|---|
| Role | Computer architecture assists in understanding the functionality of the computer. | Computer organization helps to understand the exact arrangement of component of a computer. |
| Actors | Actors in Computer architecture are hardware parts. | Actor in computer organization is performance. |

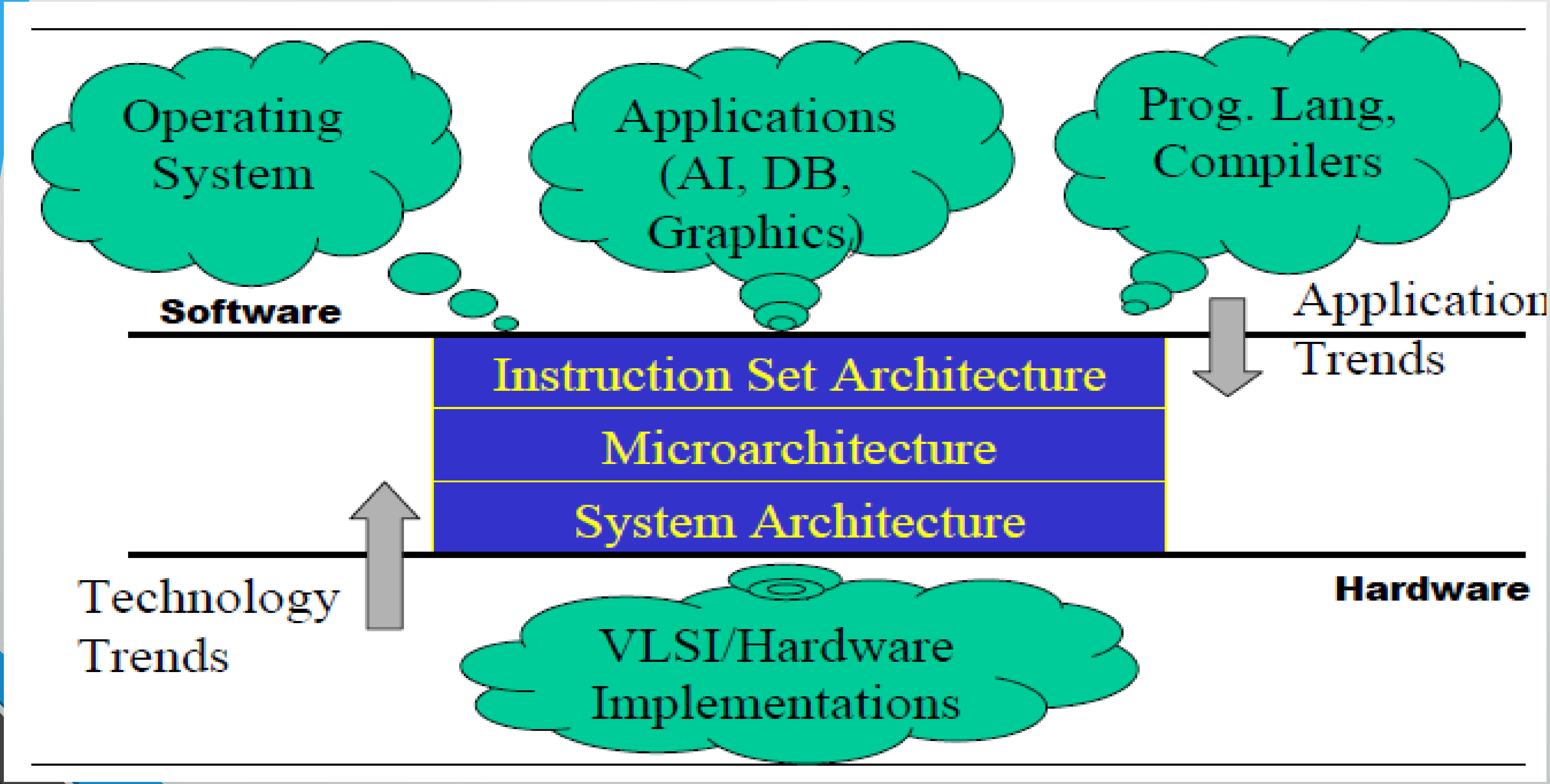
# Differences between Computer Architecture and Computer Organization

| Key | Computer Architecture | Computer Organization |
|---|---|---|
| Purpose | Computer architecture explains what a computer should do. | Computer organization explains how a computer works. |
| Target | Computer architecture provides functional behavior of computer system. | Computer organization provides structural relationships between parts of computer system. |
| Design | Computer architecture deals with high level design. | Computer organization deals with low level design. |

# Differences between Computer Architecture and Computer Organization

| Key | Computer Architecture | Computer Organization |
|---|---|---|
| Involves | Computer architecture involves the relationship among logical attributes of the system like instruction sets, data types, addressing modes, etc. | Computer organization involves the relationship among physical parts of the system like circuits, peripherals, etc. |

# What is Computer Architecture?



# What is Computer Architecture?

- computer architecture defines the design and functionality of a computer system.
- the term *computer architecture* often referred only to instruction set design.
- The components of a microcomputer are designed to interact with one another, and this interaction plays an important role in the overall system operation.
- Computer technology has made incredible progress has more performance, more main memory, and more disk storage technological improvements have been fairly steady, progress arising from better computer architectures has been much less consistent.

# What is Computer Architecture?

- The ability of the microprocessor to ride the improvements in integrated circuit technology led to a higher rate of performance improvement
- This growth rate, combined with the cost advantages of a mass-produced microprocessor, led to an increasing fraction of the computer business being based on microprocessors
- An architect must plan for technology changes that can increase the lifetime of a successful computer

# Application Areas

- Commercial Servers

– Database, transaction processing, search engines

– Performance, Availability, Scalability

– Server downtime could cost a brokerage company more than $6M/hour

– Examples: Sun Fire 15K, IBM p690, Google Cluster

- Scientific Applications

– Protein Folding, Weather Modeling, CompBio, Defense

– Floating-point arithmetic, Huge Memories

– Examples: IBM DeepBlue, BlueGene, Cray T3E, etc.

# Software Trends

- No longer just executing C/FORTRAN code
- Object Oriented Programming
- Java
- Architectural features to assist security
- Middleware

– Layer(s) between client and server applications

– Hides complexity of client/server communications

# Moore’s Law

- Moore’s observation in 1965:
  - Number of transistors per square inch on ***integrated circuits had doubled every Two years***
- Moore’s revised observation in 1975:
  - The space slowed down a bit, but ***data density had doubled approximately every 18 months***

# Moore's Law

- Every 18-24 months

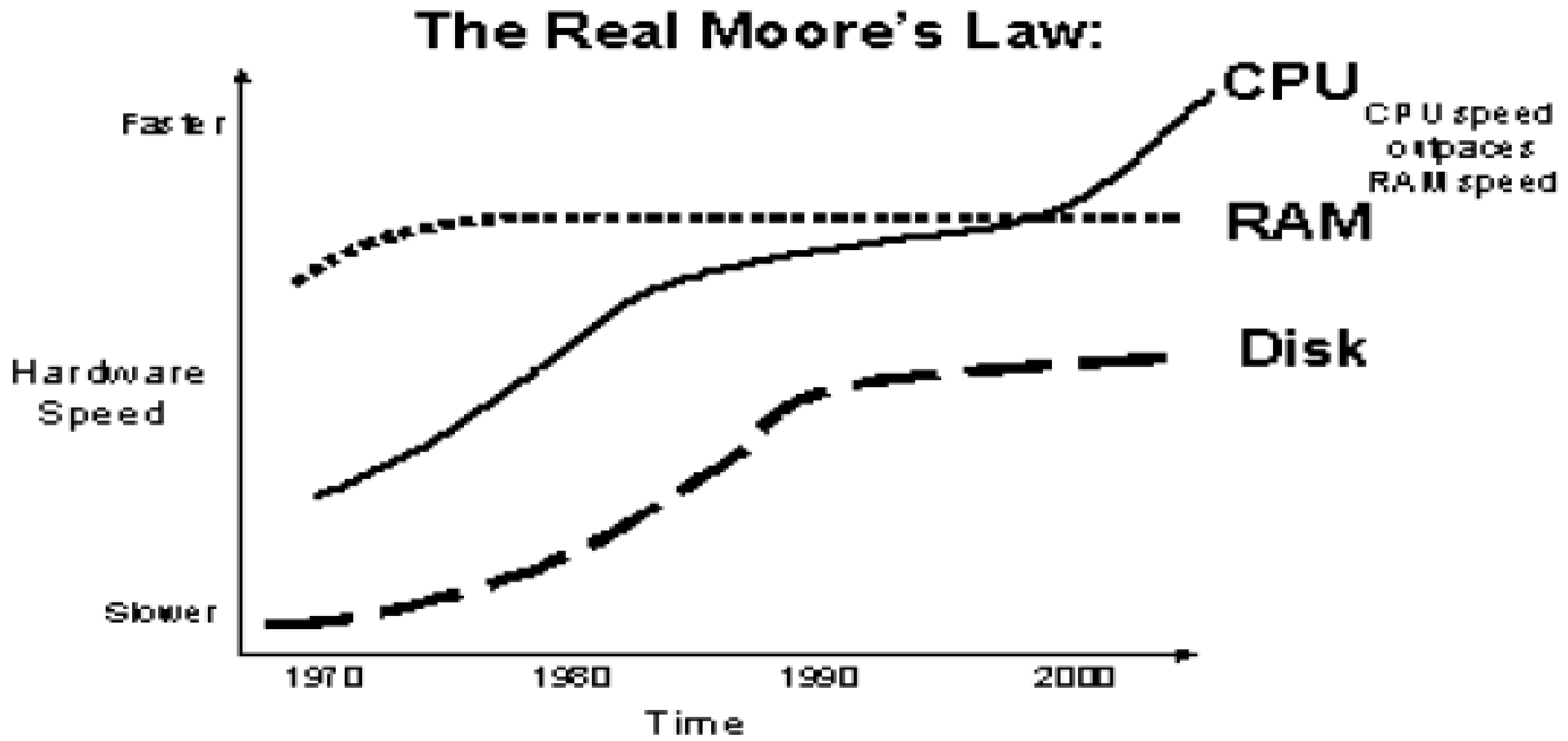
– Feature sizes shrink by 0.7x

– Number of transistors per die increases by 2x

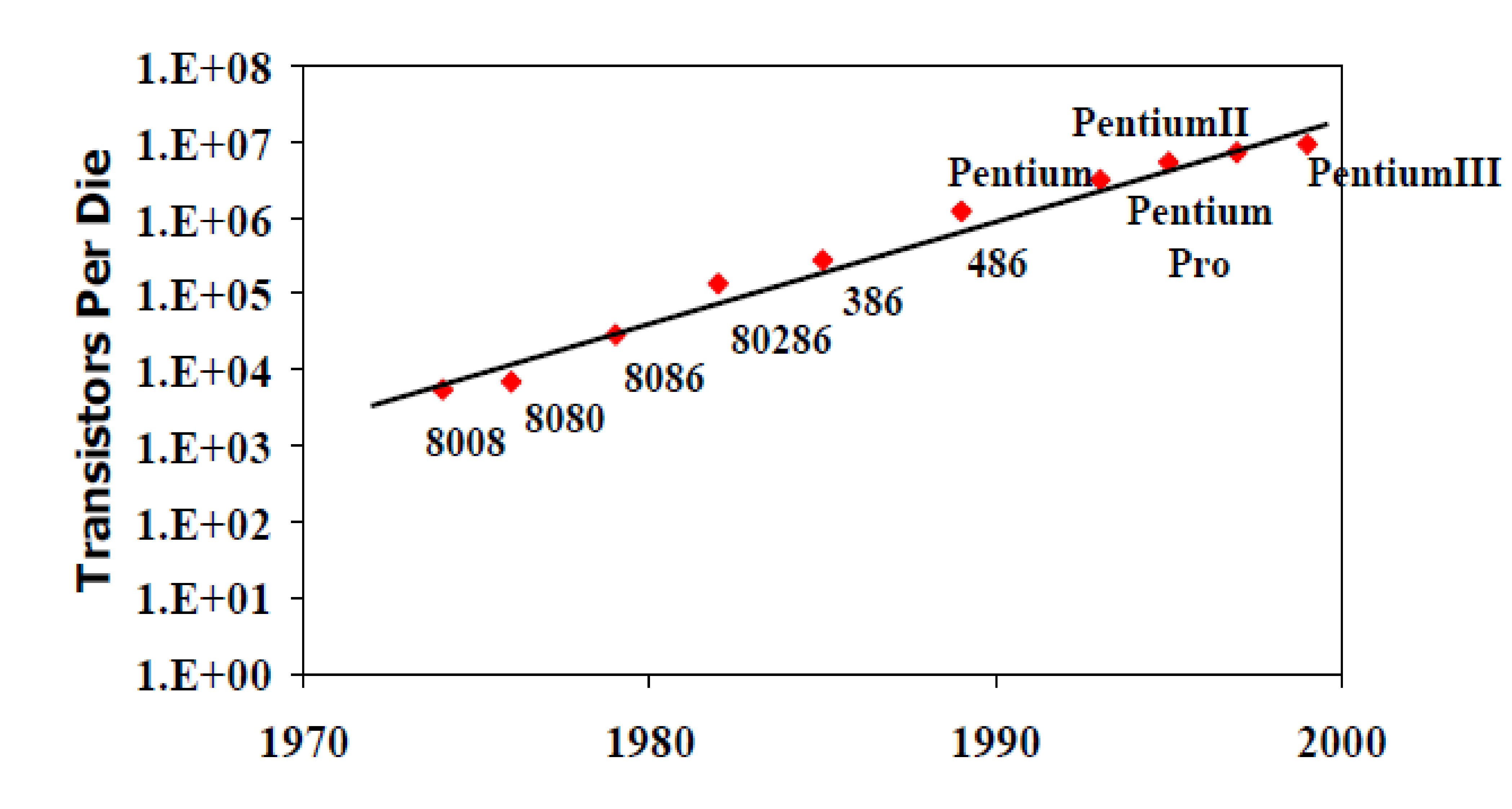
– Speed of transistors increases by 1.4x

- But we are starting to hit some roadblocks…
- Also, what to do with all of these transistors???

# CPU, Memory, and Disk Speed



# Moore's Law (Density)



# How have we used these transistors?

- More functionality on one chip

– Early 1980s – 32-bit microprocessors

– Late 1980s – On Chip Level 1 Caches

– Early/Mid 1990s – 64-bit microprocessors, superscalar (ILP)

– Late 1990s – On Chip Level 2 Caches

– Early 2000s – Chip Multiprocessors, On Chip Level 3 Caches

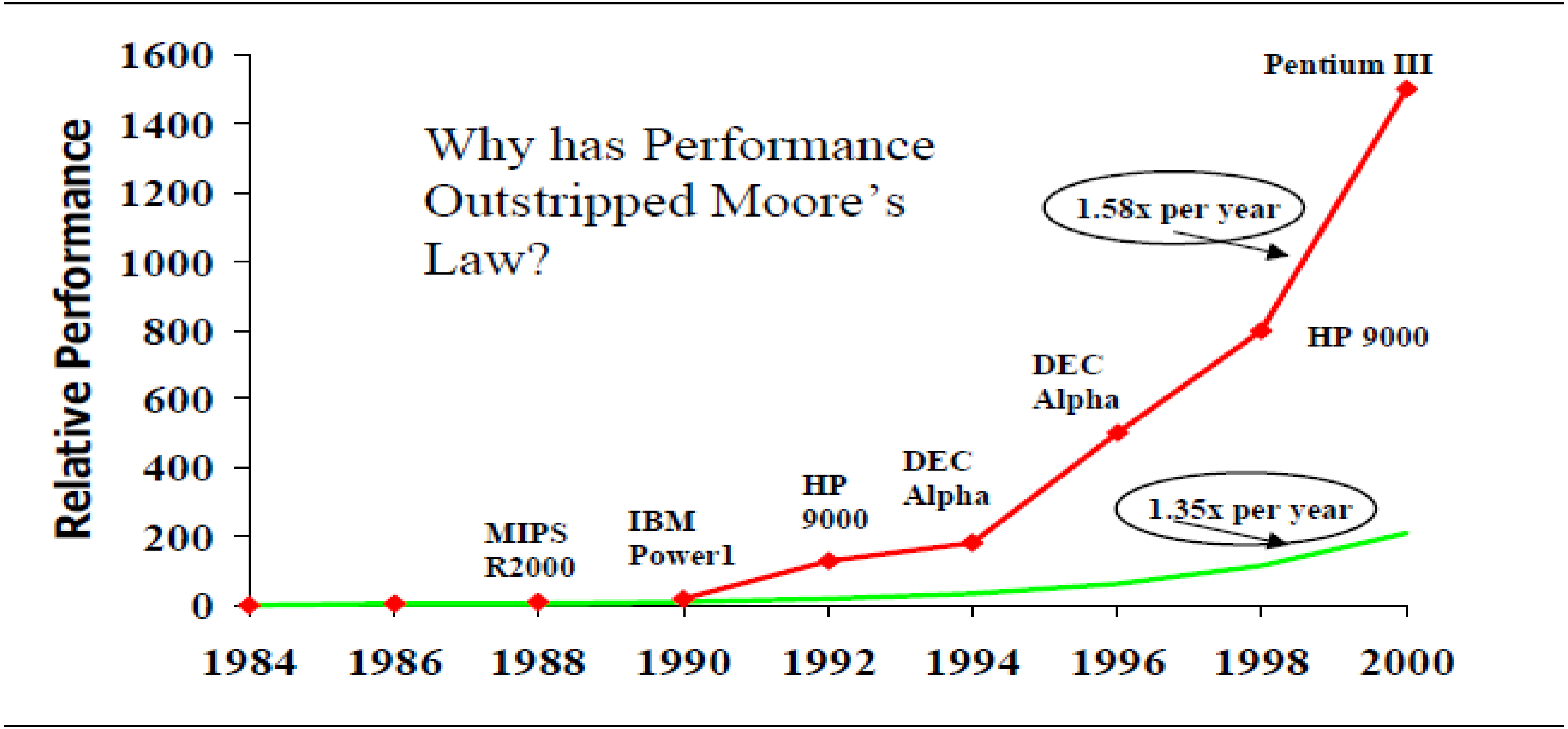
- • What is next?

– How much more cache can we put on a chip? (IRAM)

– How many more cores can we put on a chip? (Piranha, etc)

– What else can we put on chips?

# Moore’s Law (Performance)



# Performance Trends: Bandwidth over Latency

- ***Bandwidth*** **or** ***throughput*** is the total amount of work done in a given time, such as megabytes per second for a disk transfer.
- ***latency*** **or** ***response time*** is the time between the start and the completion of an event, such as milliseconds for a disk access.

# Performance vs. Technology Scaling

- Architectural Innovations

– Massive pipelining (good and bad!)

– Huge caches

– Branch Prediction, Register Renaming, OOO-issue/execution,

Speculation (hardware/software versions of all of these)

- • Circuit Innovations

– New logic circuit families (dynamic logic)

– Better CAD tools

– Advanced computer arithmetic

# Moore’s Law (Power)