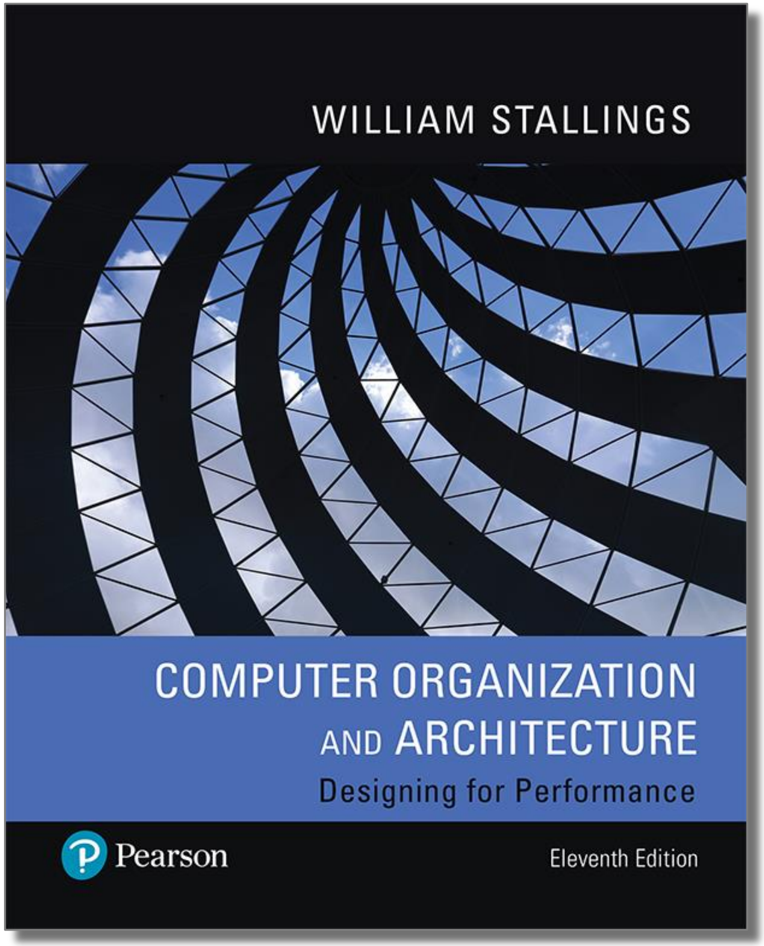
Computer Organization and Architecture

Designing for Performance

11th Edition


WILLIAM STALLINGS
COMPUTER ORGANIZATION AND ARCHITECTURE
Designing for Performance
Pearson
Eleventh Edition

Chapter 2

Performance Concepts

Designing for Performance

- The cost of computer systems continues to drop dramatically, while the performance and capacity of those systems continue to rise equally dramatically
- Today's laptops have the computing power of an IBM mainframe from 10 or 15 years ago
- Processors are so inexpensive that we now have microprocessors we throw away
- Desktop applications that require the great power of today's microprocessor-based systems include:
  - Image processing
  - Three-dimensional rendering
  - Speech recognition
  - Videoconferencing
  - Multimedia authoring
  - Voice and video annotation of files
  - Simulation modeling
- Businesses are relying on increasingly powerful servers to handle transaction and database processing and to support massive client/server networks that have replaced the huge mainframe computer centers of yesteryear
- Cloud service providers use massive high-performance banks of servers to satisfy high-volume, high-transaction-rate applications for a broad spectrum of clients

Designing for Performance

- Businesses are relying on increasingly powerful servers to handle transaction and database processing and to support massive client/server networks that have replaced the huge mainframe computer centers of yesteryear
- Cloud service providers use massive high-performance banks of servers to satisfy high-volume, high-transaction-rate applications for a broad spectrum of clients

The need for high performance computer is rising!

Solution?

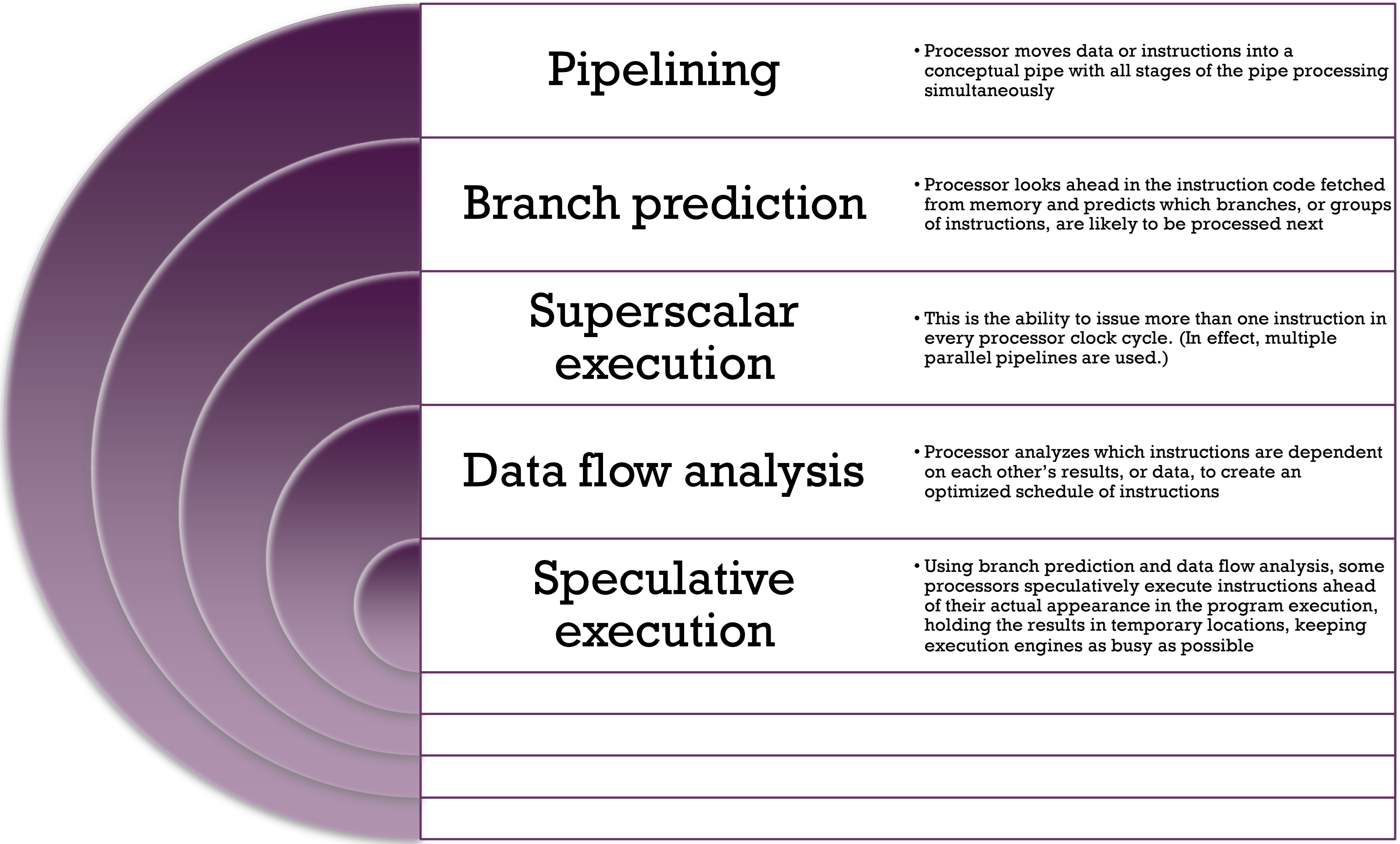
- Hardware innovation
- Implementation techniques

Improvements in Chip Organization and Architecture

- Increase hardware speed of processor
  - Fundamentally due to shrinking logic gate size
    - More gates, packed more tightly, increasing clock rate
    - Propagation time for signals reduced
- Increase size and speed of caches
  - Dedicating part of processor chip
    - Cache access times drop significantly
- Change processor organization and architecture
  - Increase effective speed of instruction execution
  - Parallelism

Microprocessor Speed

Techniques built into contemporary processors include:



Pipelining
- Processor moves data or instructions into a conceptual pipe with all stages of the pipe processing simultaneously

Branch prediction
- Processor looks ahead in the instruction code fetched from memory and predicts which branches, or groups of instructions, are likely to be processed next

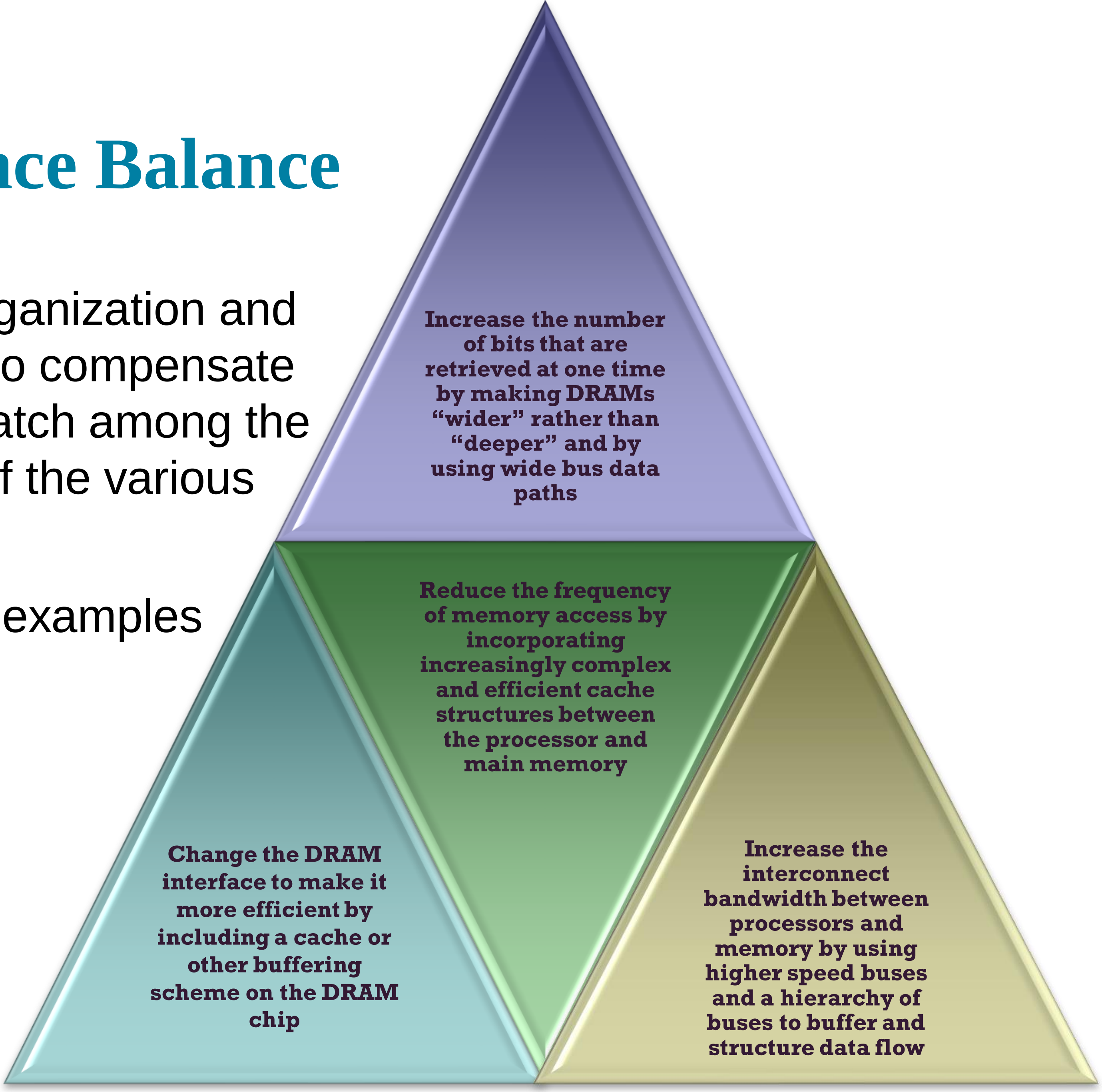
Superscalar execution
- This is the ability to issue more than one instruction in every processor clock cycle. (In effect, multiple parallel pipelines are used.)

Data flow analysis
- Processor analyzes which instructions are dependent on each other's results, or data, to create an optimized schedule of instructions

Speculative execution
- Using branch prediction and data flow analysis, some processors speculatively execute instructions ahead of their actual appearance in the program execution, holding the results in temporary locations, keeping execution engines as busy as possible

Performance Balance

- Adjust the organization and architecture to compensate for the mismatch among the capabilities of the various components
- Architectural examples include:



Increase the number of bits that are retrieved at one time by making DRAMs "wider" rather than "deeper" and by using wide bus data paths

Change the DRAM interface to make it more efficient by including a cache or other buffering scheme on the DRAM chip

Reduce the frequency of memory access by incorporating increasingly complex and efficient cache structures between the processor and main memory

Increase the interconnect bandwidth between processors and memory by using higher speed buses and a hierarchy of buses to buffer and structure data flow

Figure 2.1

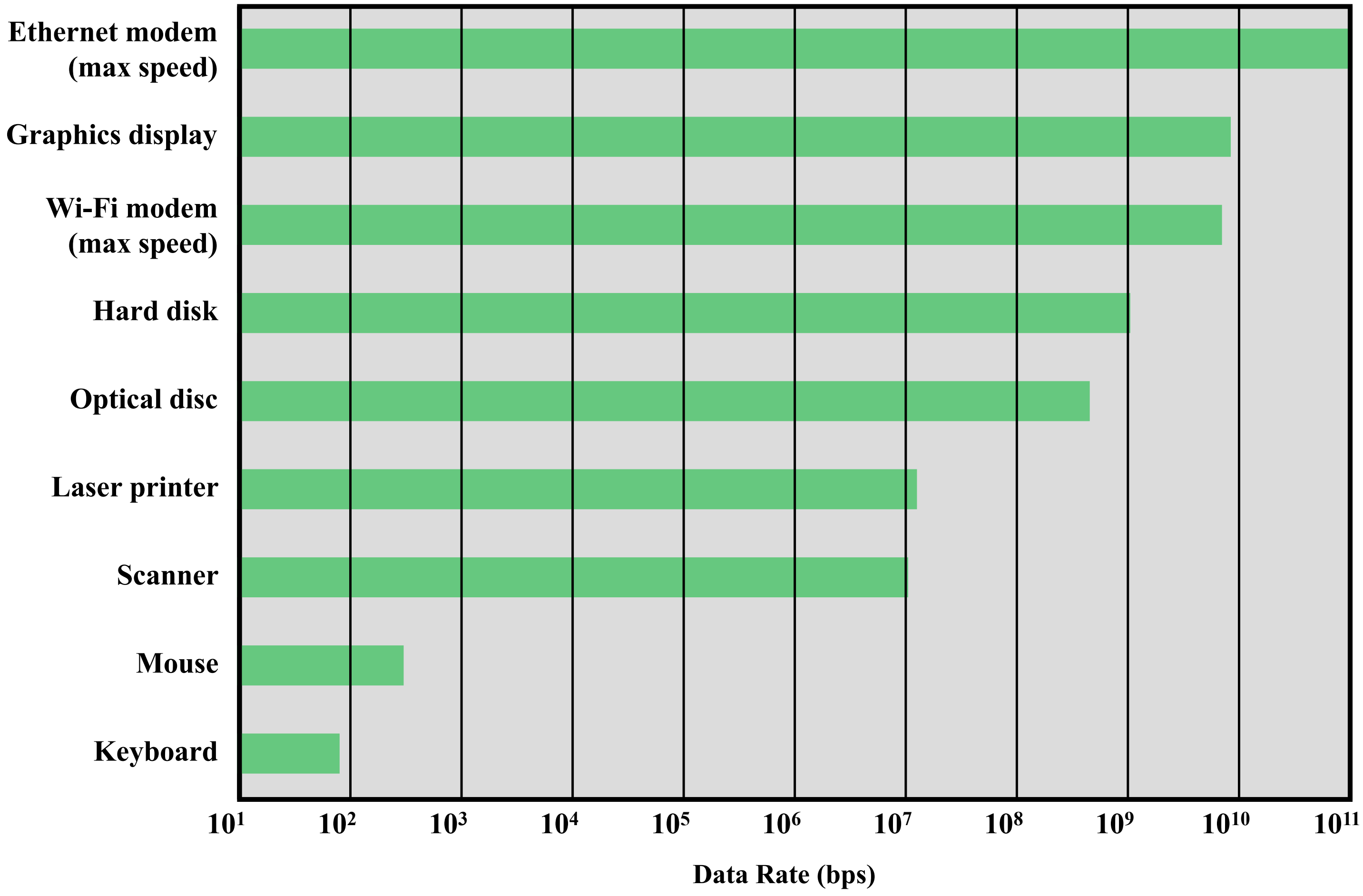


Figure 2.1 Typical I/O Device Data Rates

Problems with further improvements

- Power
  - Power density increases with density of logic and clock speed
  - Dissipating heat
- RC delay
  - Speed at which electrons flow is limited by resistance and capacitance of metal wires connecting them
  - Delay increases as the RC product increases
  - As components on the chip decrease in size, the wire interconnects become thinner, increasing resistance
  - Also, the wires are closer together, increasing capacitance
- Memory latency and throughput
  - Memory access speed (latency) and transfer speed (throughput) lag processor speeds

Pearson

Figure 2.2

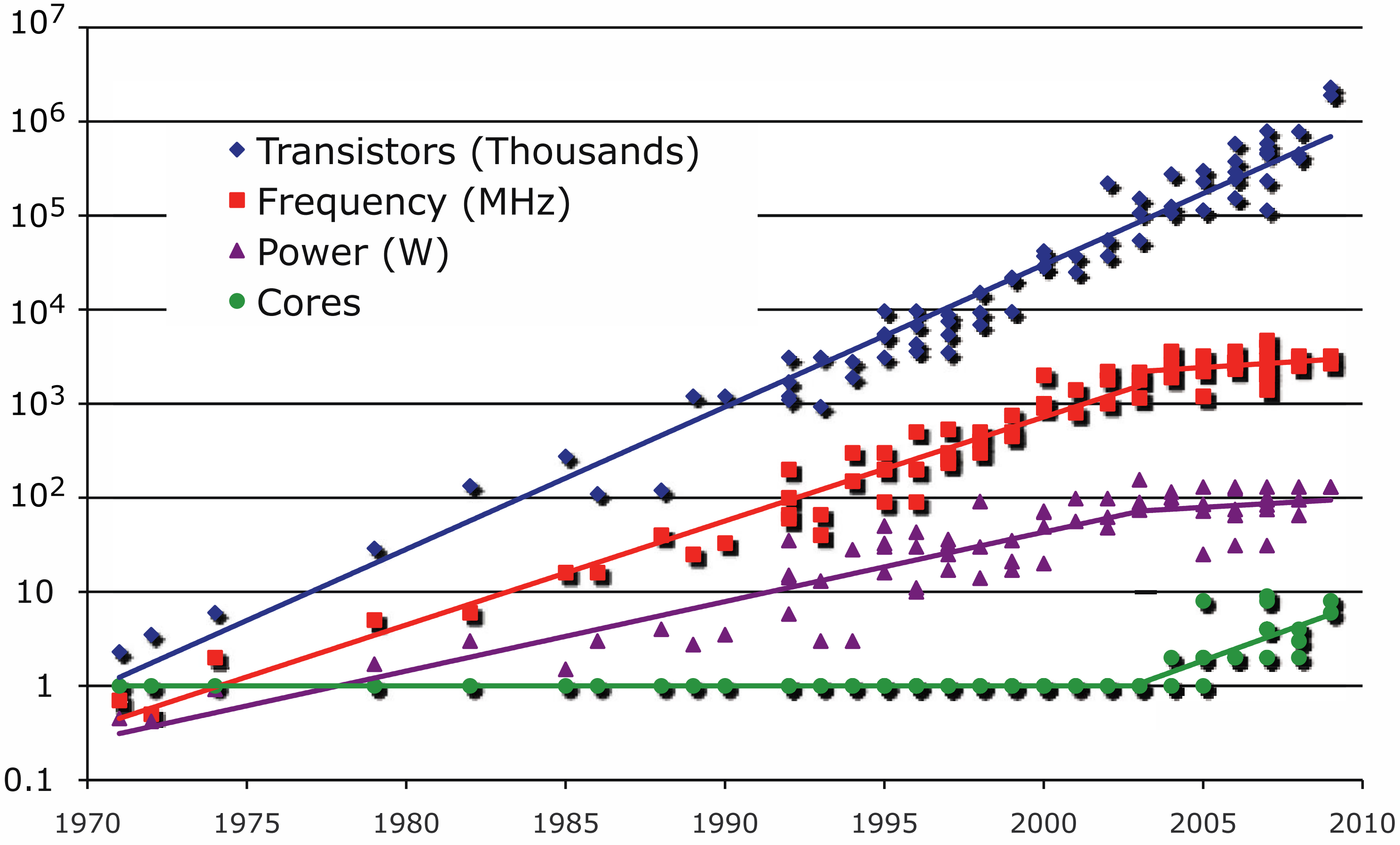


Figure 2.2 Processor Trends

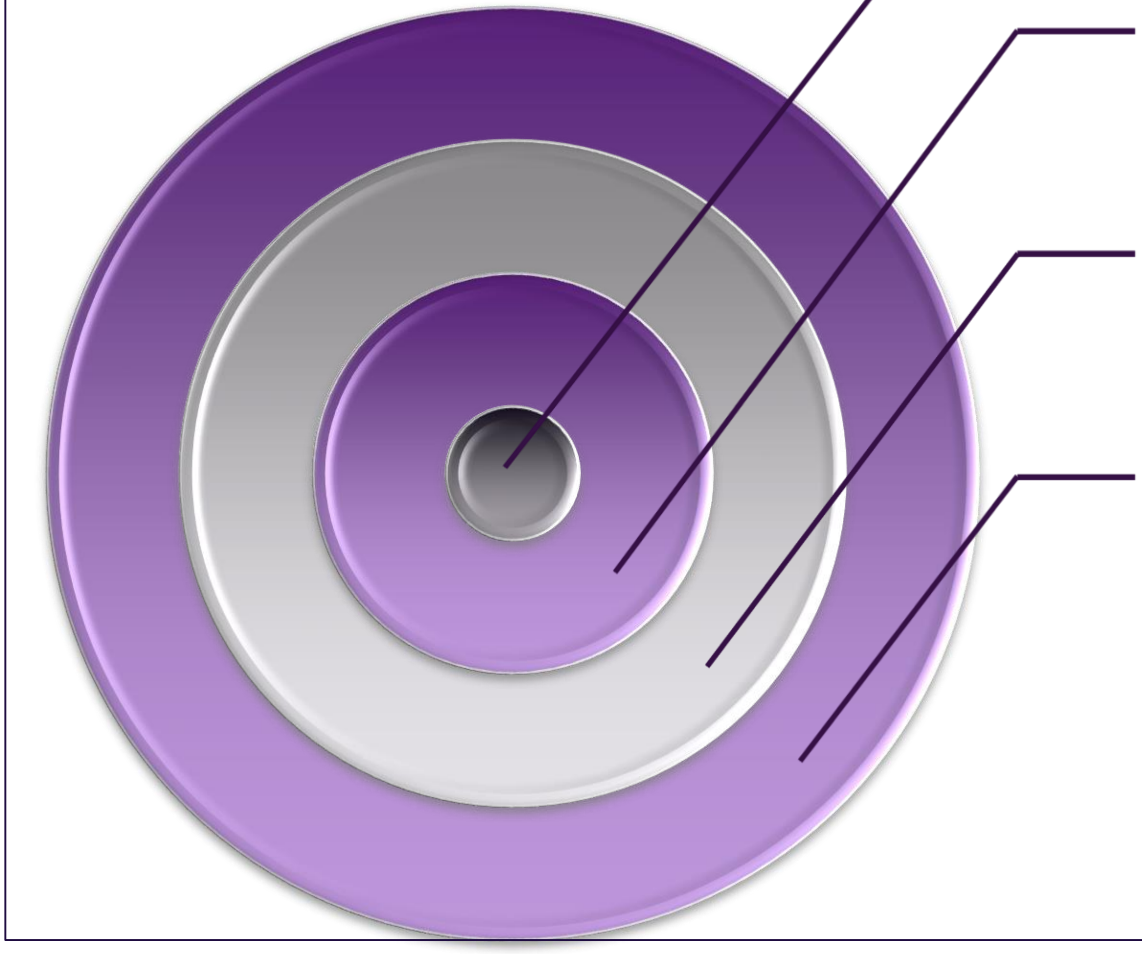
Problems with further improvements

- Meanwhile, clock speed has levelled off to prevent a further rise in power.
- To continue to increase performance, designers have had to find ways of exploiting the growing number of transistors other than simply building a more complex processor
- Next steps?

In recent year, the response has been the development of

Multicore computer chip.

Multicore



The use of multiple processors on the same chip provides the potential to increase performance without increasing the clock rate

Strategy is to use two simpler processors on the chip rather than one more complex processor

With two processors larger caches are justified

As caches became larger it made performance sense to create two and then three levels of cache on a chip

Many Integrated Core (MIC) Graphics Processing Unit (GPU)

MIC

- Leap in performance as well as the challenges in developing software to exploit such a large number of cores
- The multicore and MIC strategy involves a homogeneous collection of general purpose processors on a single chip

GPU

- Core designed to perform parallel operations on graphics data
- Traditionally found on a plug-in graphics card, it is used to encode and render 2D and 3D graphics as well as process video
- Used as vector processors for a variety of applications that require repetitive computations

Amdahl’s Law

- Improvement in technology or design doesn’t result in corresponding improvement in performance.
- This limitation is addressed by Amdahl’s law (1967).
- Deals with the **potential speedup of a program using multiple processors compared to a single processor**
- Illustrates the problems facing industry in the development of multi-core machines
  - Software must be adapted to a highly parallel execution environment to exploit the power of parallel processing
- Can be generalized to evaluate and design technical improvement in a computer system

Figure 2.3

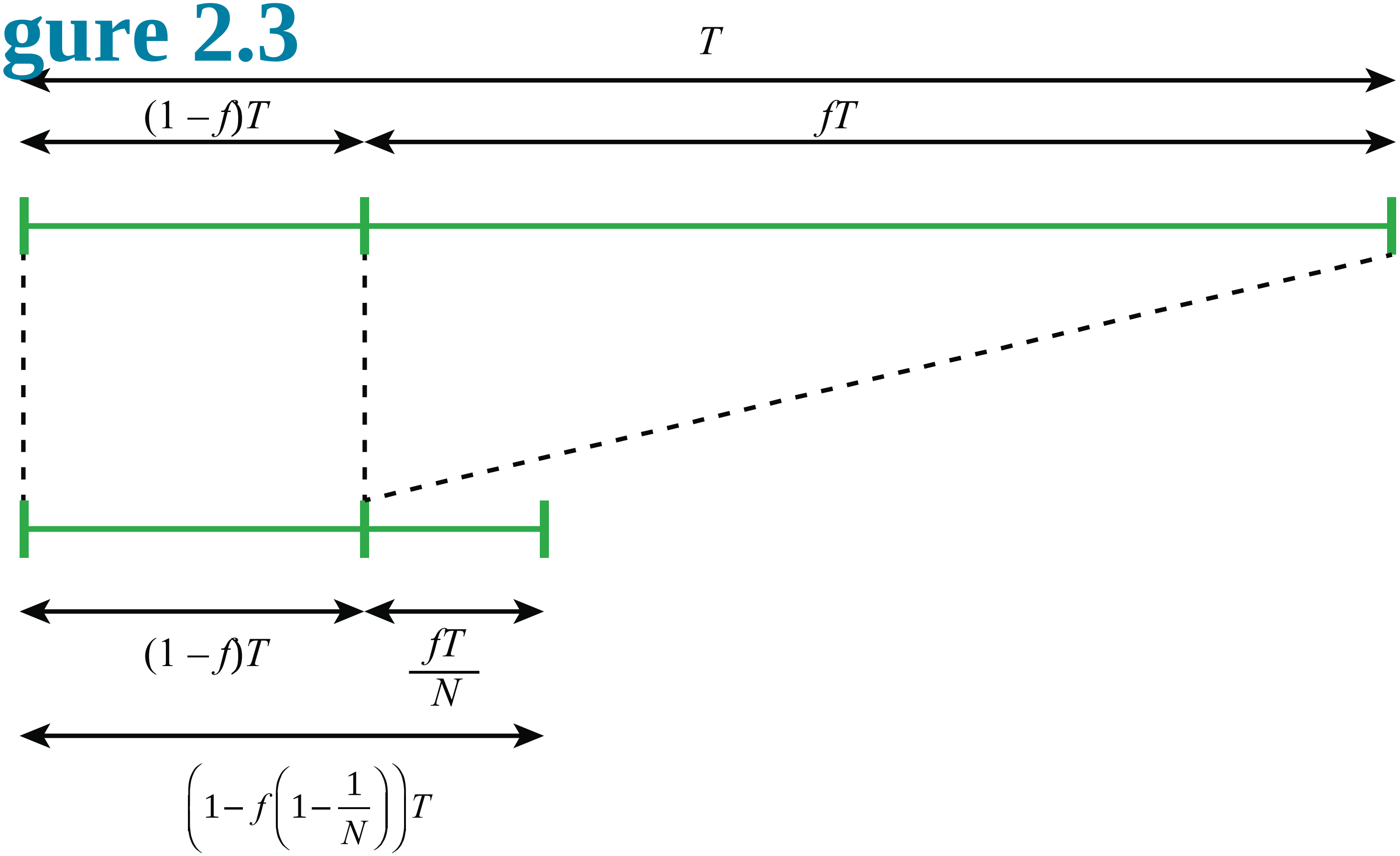


Figure 2.3 Illustration of Amdahl’s Law

Figure 2.4

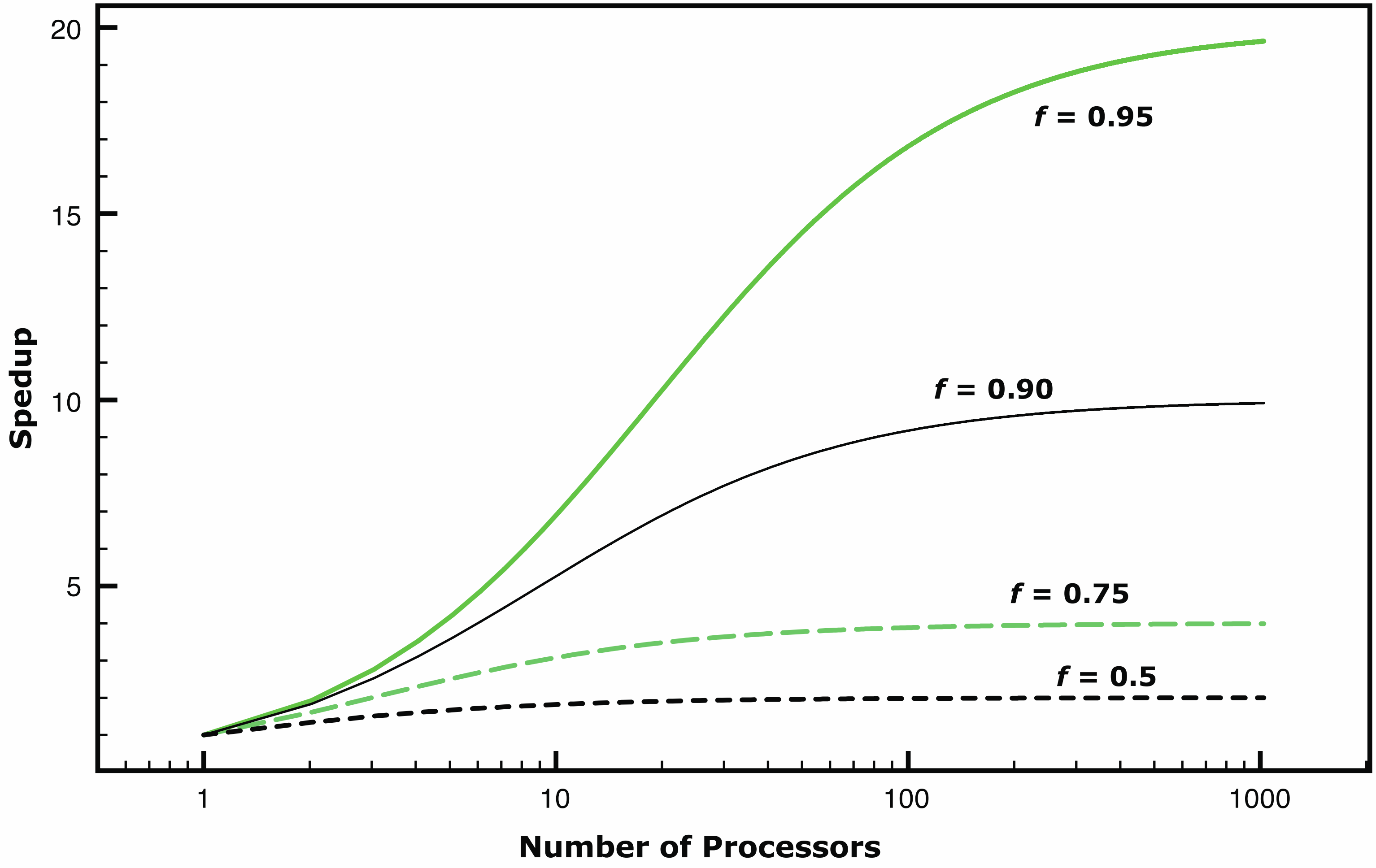


Figure 2.4 Amdahl’s Law for Multiprocessors

Amdahl's Law Example

- Suppose that a task takes extensive use of floating-point operations, with 40% of the time consumed by floating-point operations. With a new hardware design, the floating-point module is sped up by a factor of K. What is the **max** overall speed up?

$$Speedup = \frac{1}{(1 - f) + \frac{f}{SU_f}} = \frac{1}{0.6 + \frac{0.4}{K}}$$

Independent of K, the maximum speedup is 1.67

Pearson

Processor Performance Measures

- **Clock speed:** Operations performed by a processor, such as fetching an instruction, decoding the instruction, performing an arithmetic operation, etc., are governed by a **system clock**.
- Typically, all operations begin with the pulse of the clock. Thus, at the most fundamental level, the speed of a processor is dictated by the pulse frequency produced by the clock, measured in cycles per second, or Hertz (Hz).
- A CPU with a clock speed of 3.2 GHz executes 3.2 billion cycles per second.
- During each cycle, billions of transistors within the processor open and close.

Processor Performance Measures

- Higher clock speed = a faster CPU? Yes, BUT other factors come into play.
- CPU with a higher clock might be outperformed CPU with a lower clock speed, as its architecture deals with instructions more efficiently.
- **Clock cycles per instruction (CPI):** the average number of clock cycles per instruction for a program.

Processor Performance Measures

- A common measure of performance for a processor is the rate at which instructions are executed, expressed as millions of instructions per second (MIPS), referred to as the **MIPS rate.**
- Another common performance measure deals only with floating-point instructions. These are common in many scientific and game applications. Floating-point performance is expressed as millions of floating-point operations per second (**MFLOPS**).
- Typically, they measure performance using **benchmark programs** to use it as a point of reference.

Copyright