

William Stallings Computer Organization and Architecture 10th Edition

© 2016 Pearson Education, Inc., Hoboken,
NJ. All rights reserved.



+Chapter 2

Performance Issues




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



+ Microprocessor Speed

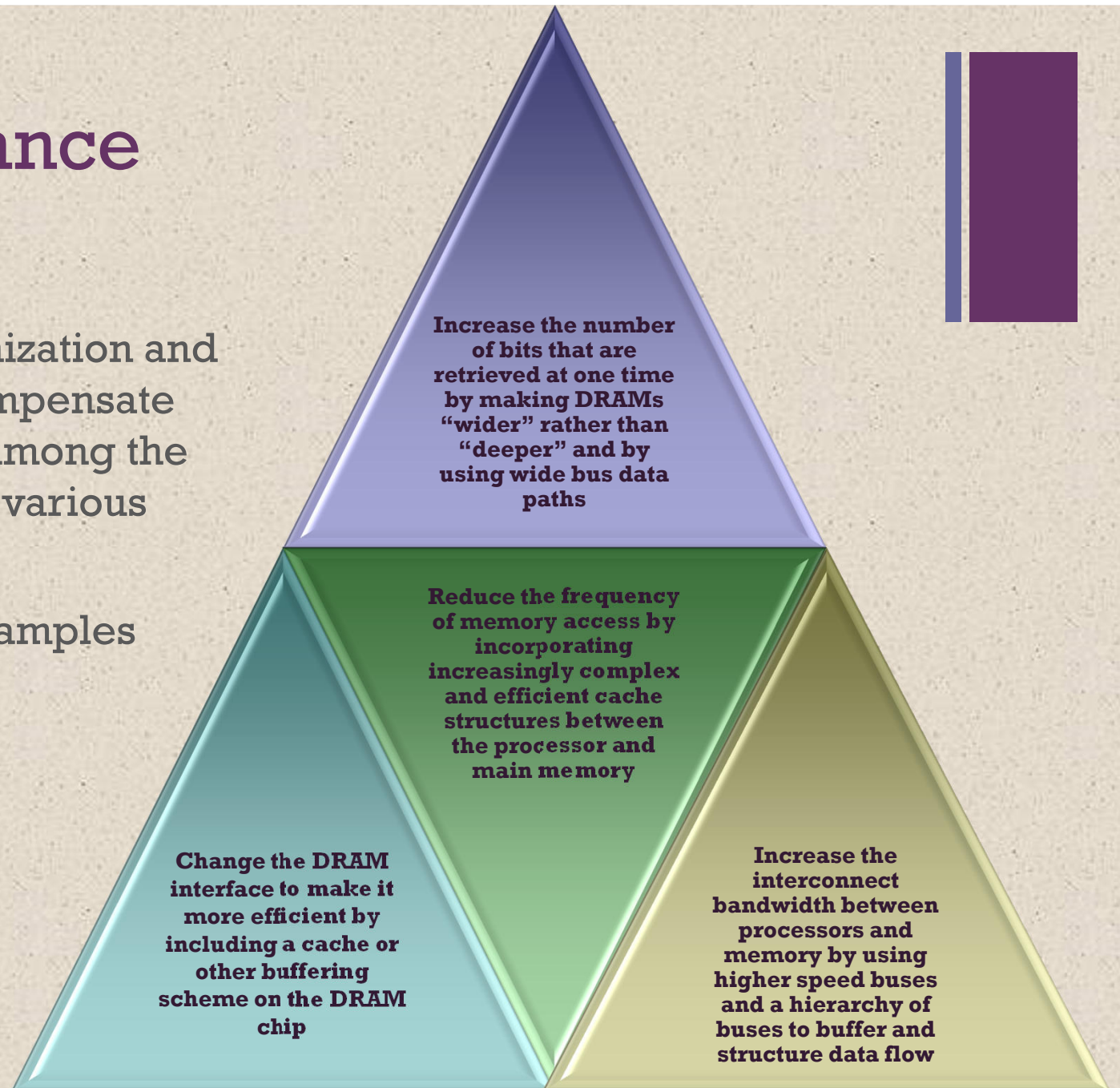
Techniques built into contemporary processors include:



Pipelining	<ul style="list-style-type: none">• Processor moves data or instructions into a conceptual pipe with all stages of the pipe processing simultaneously
Branch prediction	<ul style="list-style-type: none">• 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	<ul style="list-style-type: none">• 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	<ul style="list-style-type: none">• Processor analyzes which instructions are dependent on each other's results, or data, to create an optimized schedule of instructions
Speculative execution	<ul style="list-style-type: none">• 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:



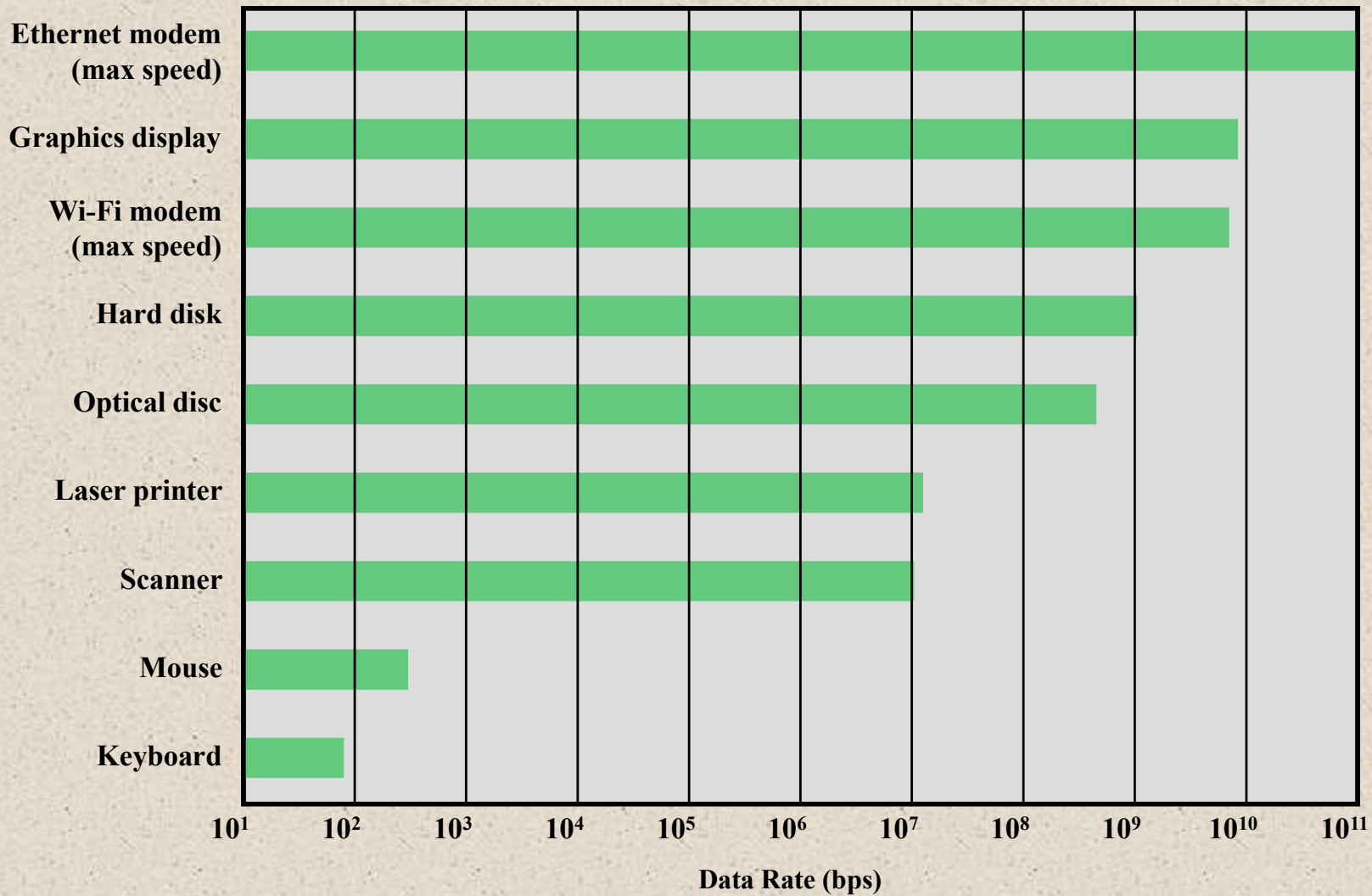
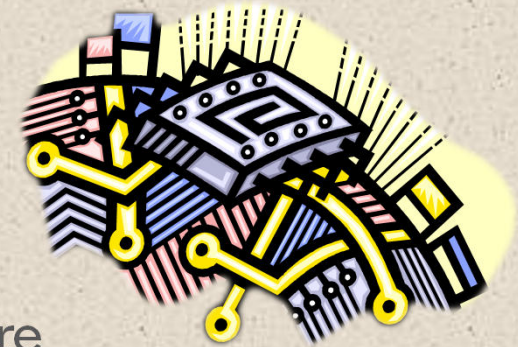


Figure 2.1 Typical I/O Device Data Rates

+ 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



+ Problems with Clock Speed and Login Density

■ Power

- Power density increases with density of logic and clock speed
- Dissipating heat

■ RC delay

- Speed at which electrons flow 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

- Memory speeds lag processor speeds

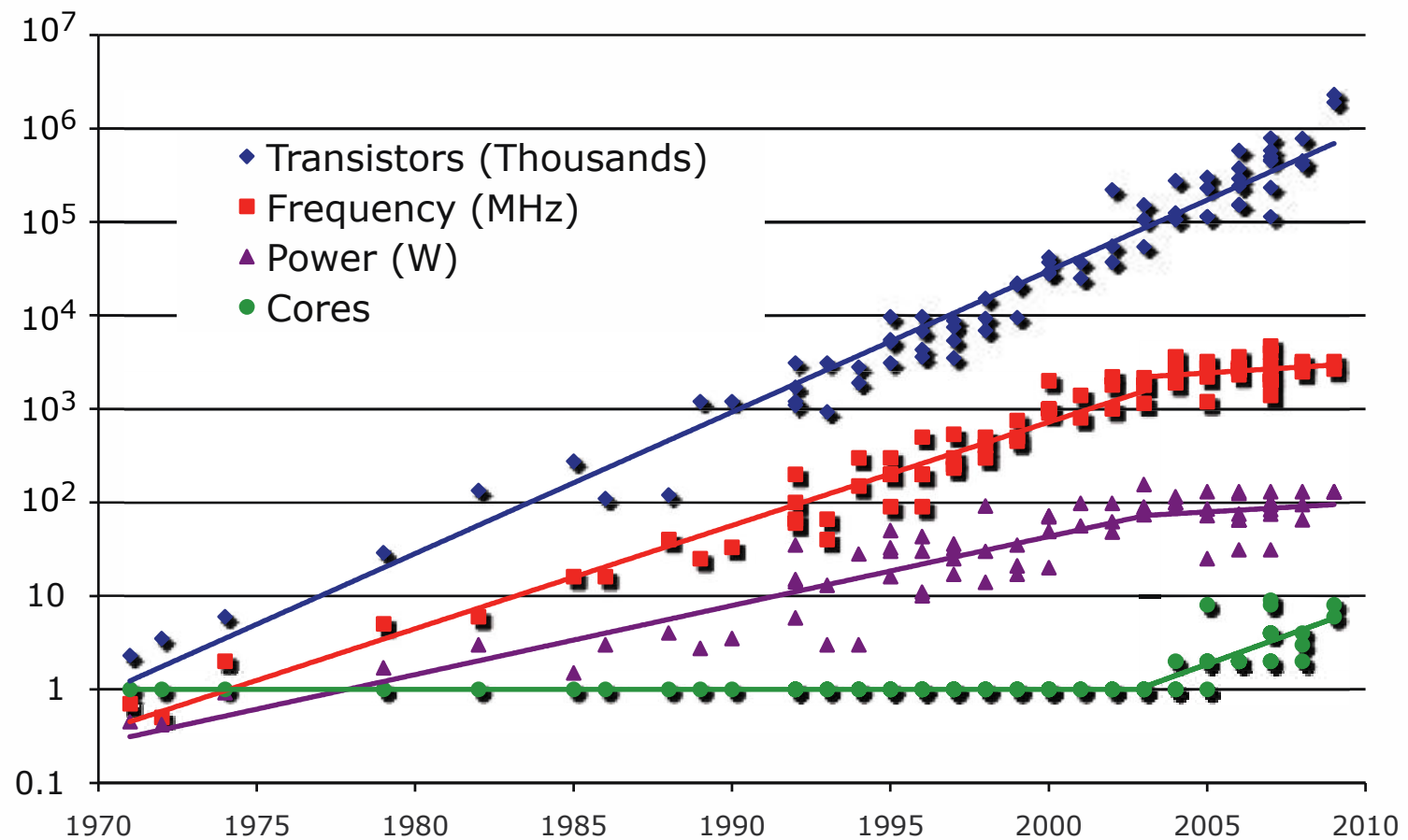
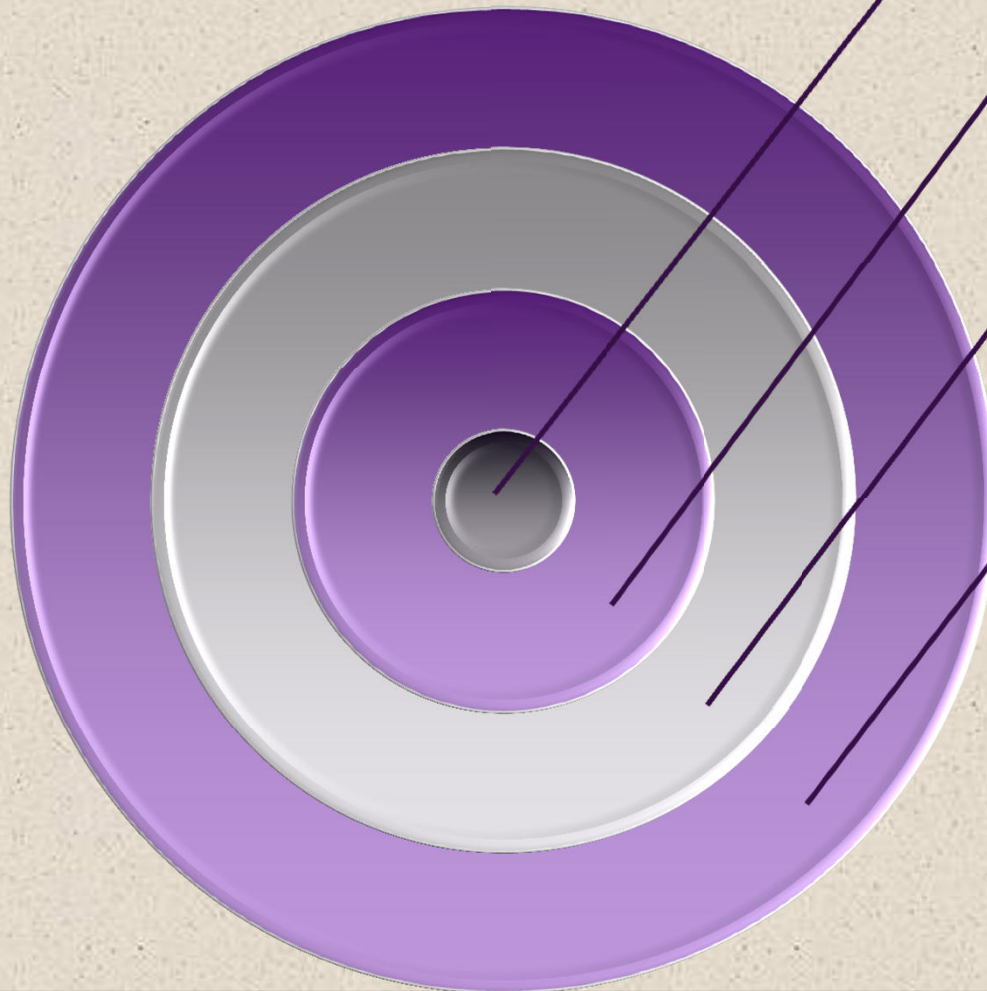


Figure 2.2 Processor Trends

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

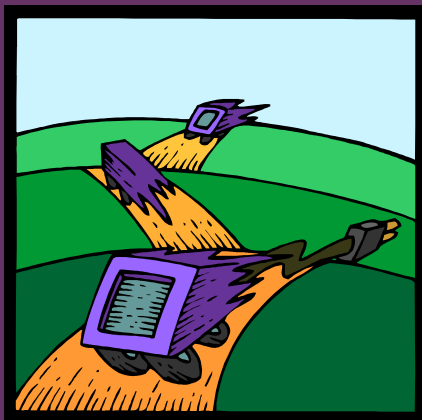
- Leap in performance as well as the challenges in developing software to exploit such a large number of cores
- The multicore and MIC strategy involves a homogeneous collection of general purpose processors on a single chip

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



- Gene Amdahl
- 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

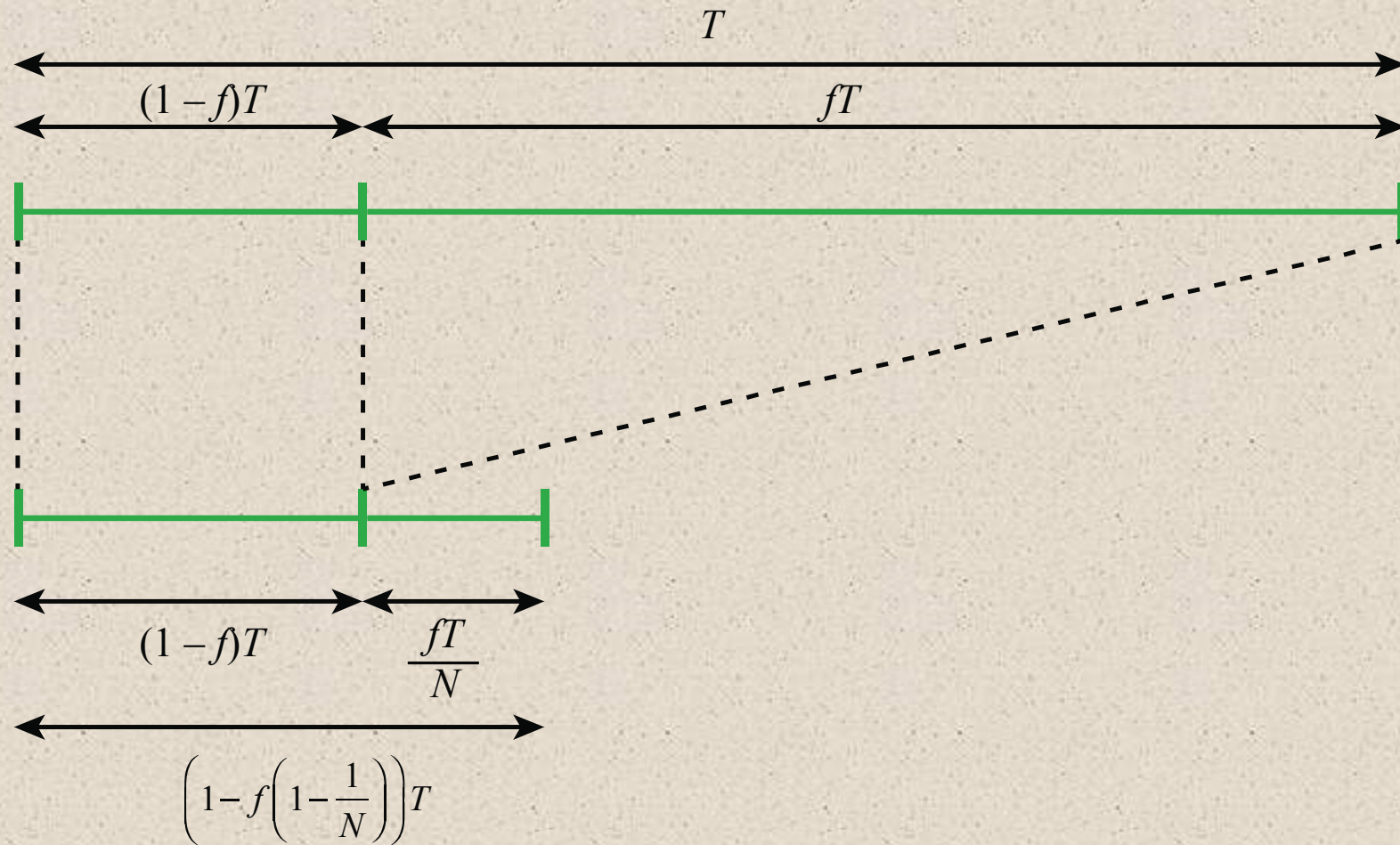


Figure 2.3 Illustration of Amdahl's Law