Multicore Computers

ICT 2203 Computer Architecture

Based on William Stallings, Computer Organization and Architecture, 8th Edition

Why Parallel Computers?

- Computer architects have long sought to create parallel computers by combining a large number of processing elements into a single system
- Parallel computers allow a large computation to be carried out in orders of magnitude shorter execution time
- Scientists and engineers have relied on parallel computers to solve important and complex scientific problems
- Parallel computers have also found a broader audience outside science, such as serving search engines, Web servers and databases
- There have been needs for parallel computers, and there will be even more needs for them in the future

Hardware Performance Issues

- Microprocessors have seen an exponential increase in performance
  - Improved organization
  - Increased clock frequency
- Increase in Parallelism
  - Pipelining
  - Superscalar
  - Simultaneous multithreading (SMT)
- Diminishing returns
  - More complexity requires more logic
  - Increasing chip area for coordinating and signal transfer logic
    - Harder to design, make and debug

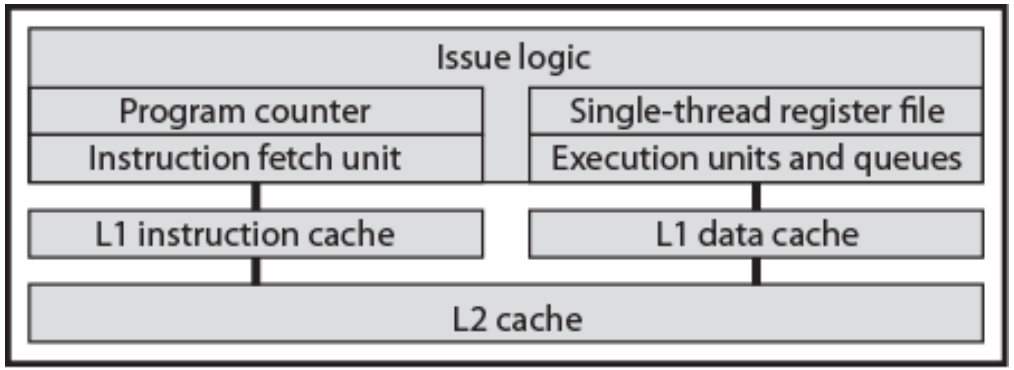
From Scalar to Superscalar

- The increasing miniaturization of transistors has resulted in various processor architecture changes
- Fitting more and more components in a single processor (e.g., integrating FPU in Intel 486 in 1989)
- Adding more features to a single processor
  - Parallelism at the instruction level and out-of-order execution (Intel Pentium Pro in 1995)
  - L1 & L2 caches (Intel Pentium III in 1999)
  - Simultaneous multithreading (SMT) or parallelism at the thread level (Intel Pentium III)

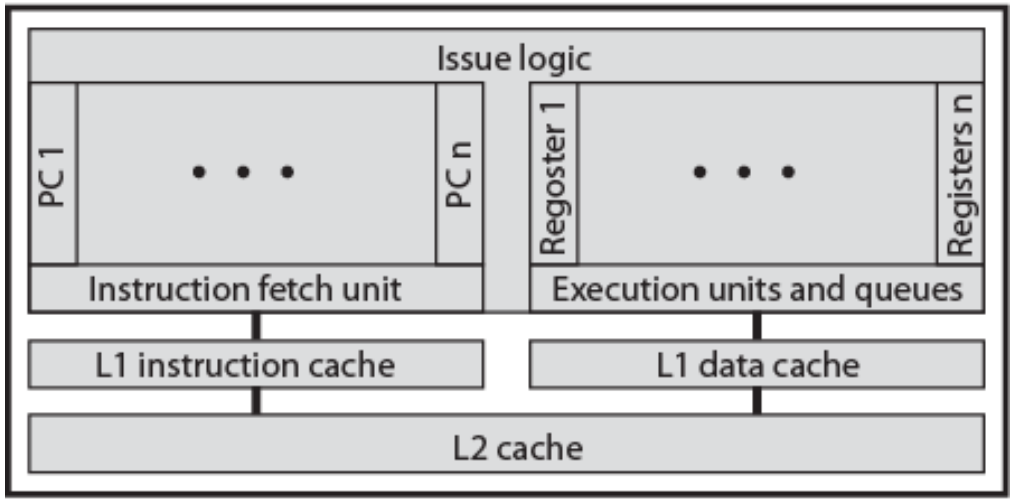
Moving to Multicore

- Instruction Level Parallelism (ILP) started to become harder
- Deepening the processor pipeline
- Branch prediction exceeded 95% accuracy
- Caches started showing diminishing returns
- Power consumption increased
- Nevertheless, Moore's law continued on, leaving a big question:
- What is the most power-performance efficient way to use the extra transistors available?
- Answer: **Multicore or Chip Multiprocessors (CMPs)!**

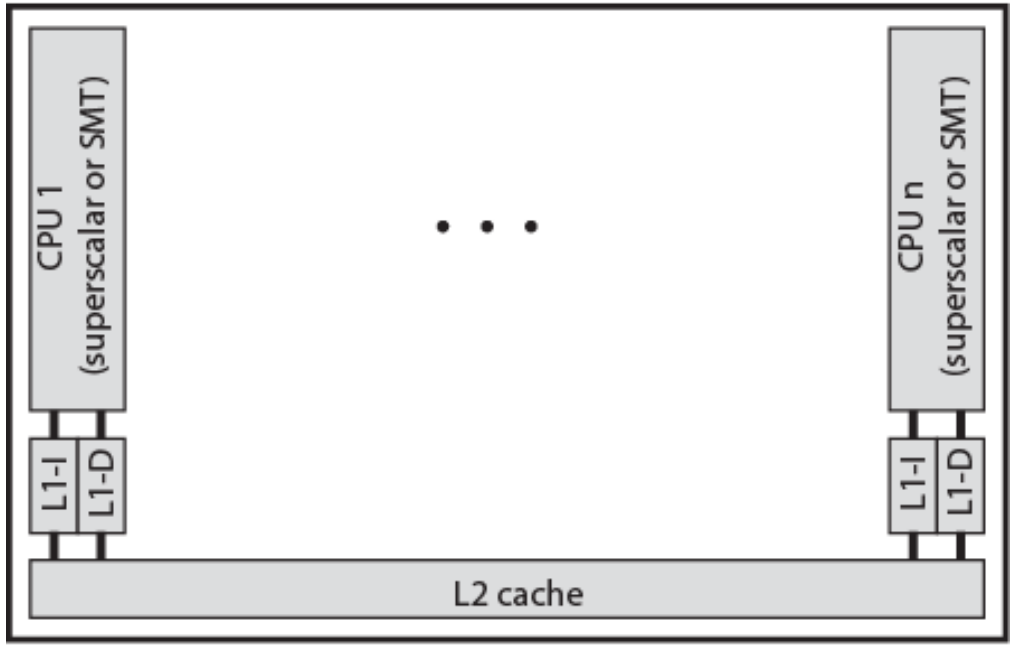
Alternative Chip Organizations



(a) Superscalar



(b) Simultaneous multithreading



(c) Multicore

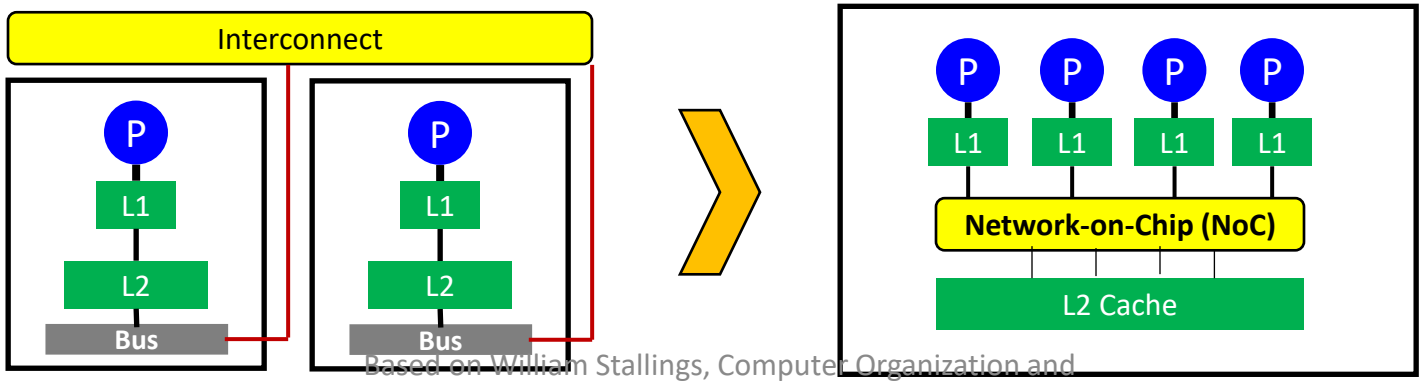
Based on William Stallings, Computer Organization and Architecture, 8th Edition

Parallel Computers Today

- Before 2001, parallel computers were mainly used in the server and supercomputer markets
- Since 2001, parallel computers have evolved as an architecture in which multiple processor cores are implemented on a single chip
- Such an architecture is popularly known as **multicore** or **chip multiprocessors (CMPs)**
- With the adoption of multicore processors, virtually all new laptops, desktops, and servers are now parallel computers
- Today, multicore/CMP is the architecture of choice

What are multicores?

- In contrast to traditional multiprocessor architectures, multicores have multiple cores implemented on a single die
- Multicores provide an attractive approach as long as performance can scale linearly with the increase in the number of cores
- This depends on the software to utilize the cores effectively
- Generally, data parallel applications with high thread level parallelism



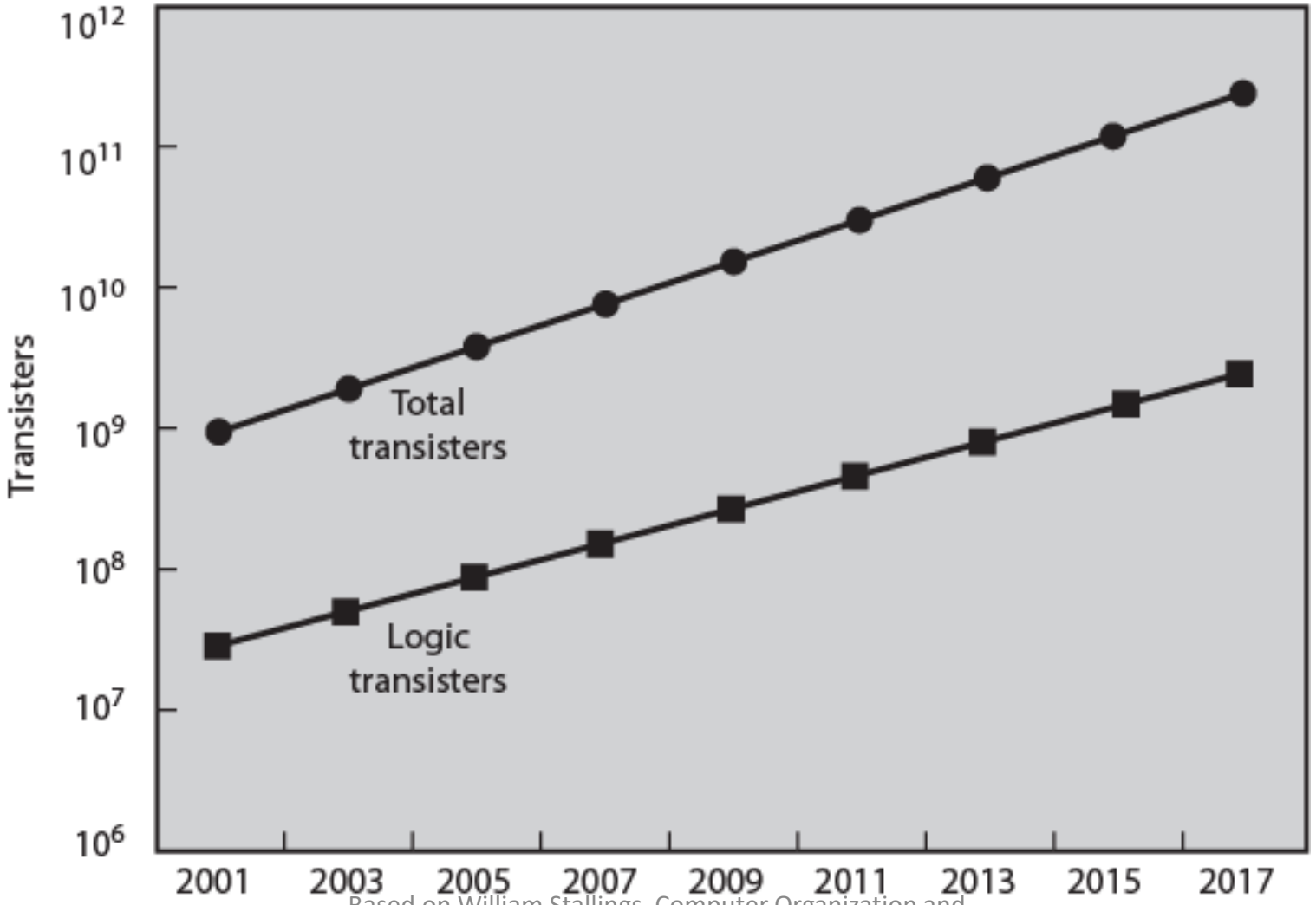
A Multiprocessor System

A Multicore System

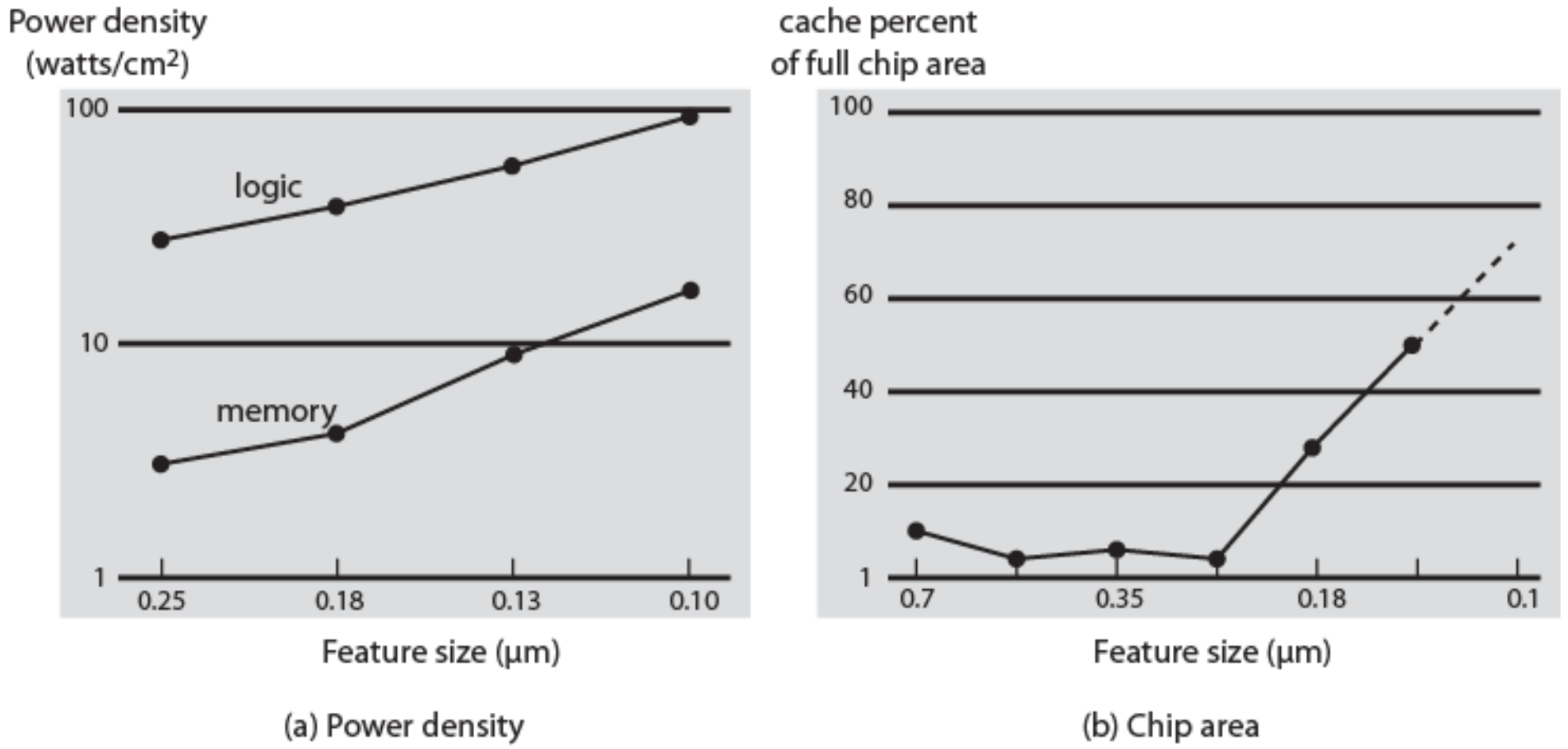
Increased Complexity

- Power requirements grow exponentially with chip density and clock frequency
  - Can use more chip area for cache
- By 2015
  - 100 billion transistors on 300mm^2 die; 1 billion transistors for logic
  - Cache of 100MB;
- Pollack's rule: Performance is roughly proportional to square root of increase in complexity
  - Double complexity gives 40% more performance
- Multicore has potential for near-linear improvement
- Unlikely that one core can use all cache effectively

Chip Utilization of Transistors



Power and Memory Considerations



Software Performance Issues

- Performance benefits dependent on effective exploitation of parallel resources
- Even small amounts of serial code impact performance
  - 10% inherently serial on 8 processor system gives only 4.7 times performance
- Communication, distribution of work and cache coherence overheads
- Some applications effectively exploit multicore processors

Effective Applications for Multicore Processors

- Database
- Servers handling independent transactions
- Multi-threaded native applications
  - Lotus Domino, Siebel CRM
- Multi-process applications
  - Oracle, SAP, PeopleSoft
- Java applications
  - Java VM is multi-thread with scheduling and memory management
  - Sun's Java Application Server, BEA's Weblogic, IBM Websphere, Tomcat
- Multi-instance applications
  - One application running multiple times
  - E.g. Value Game Software

Multicore Organization

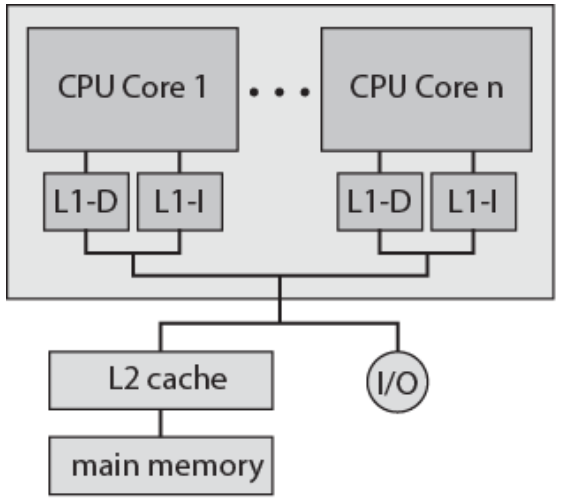
- Number of core processors on chip
- Number of levels of cache on chip
- Amount of shared cache

a. ARM11 MPCore

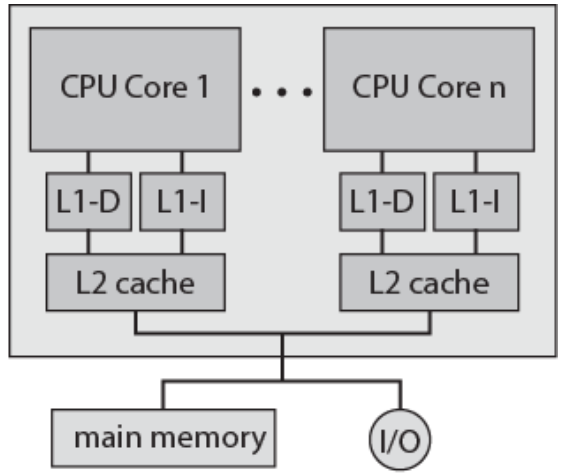
b. AMD Opteron

c. Intel Core Duo

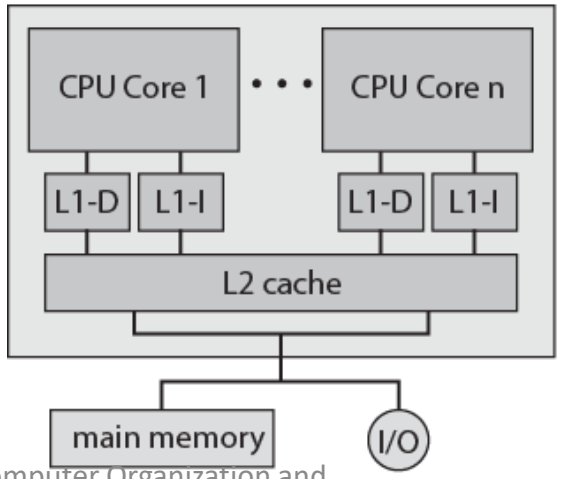
d. Intel Core i7



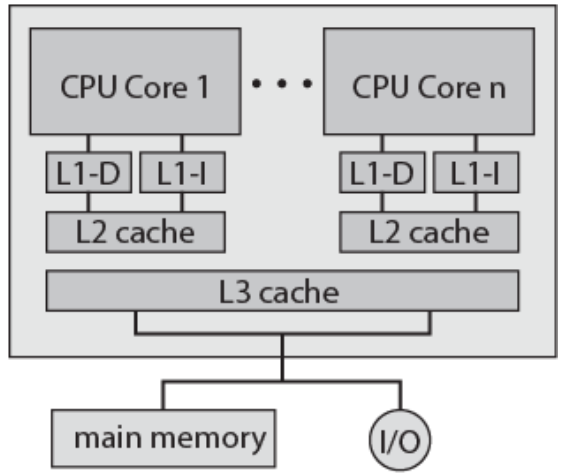
(a) Dedicated L1 cache



(b) Dedicated L2 cache



(c) Shared L2 cache



(d) Shared L3 cache

Advantages of shared L2 Cache

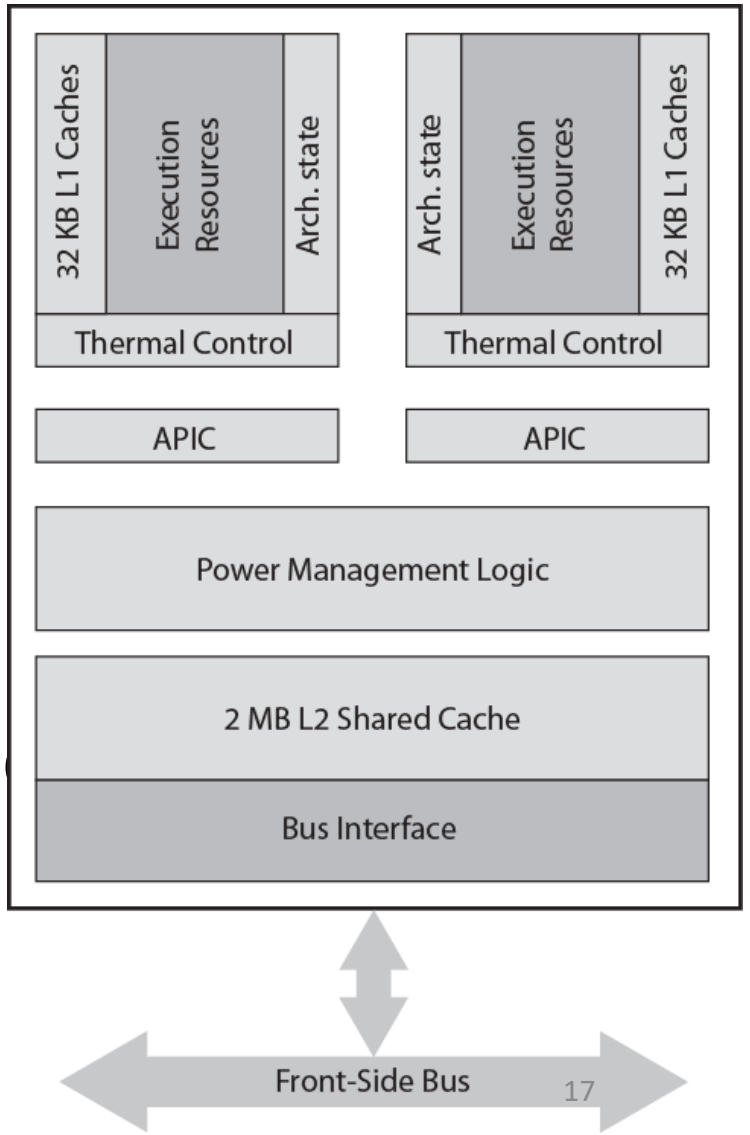
- Constructive interference reduces overall miss rate
- Data shared by multiple cores not replicated at cache level
- With proper frame replacement algorithms mean amount of shared cache dedicated to each core is dynamic
- Easy inter-process communication through shared memory
- Cache coherency confined to L1
- Dedicated L2 cache gives each core more rapid access
- Shared L3 cache may also improve performance

Individual Core Architecture

- Intel Core Duo uses superscalar cores
- Intel Core i7 uses simultaneous multi-threading (SMT)
  - Scales up number of threads supported
    - 4 SMT cores, each supporting 4 threads appears as 16 core

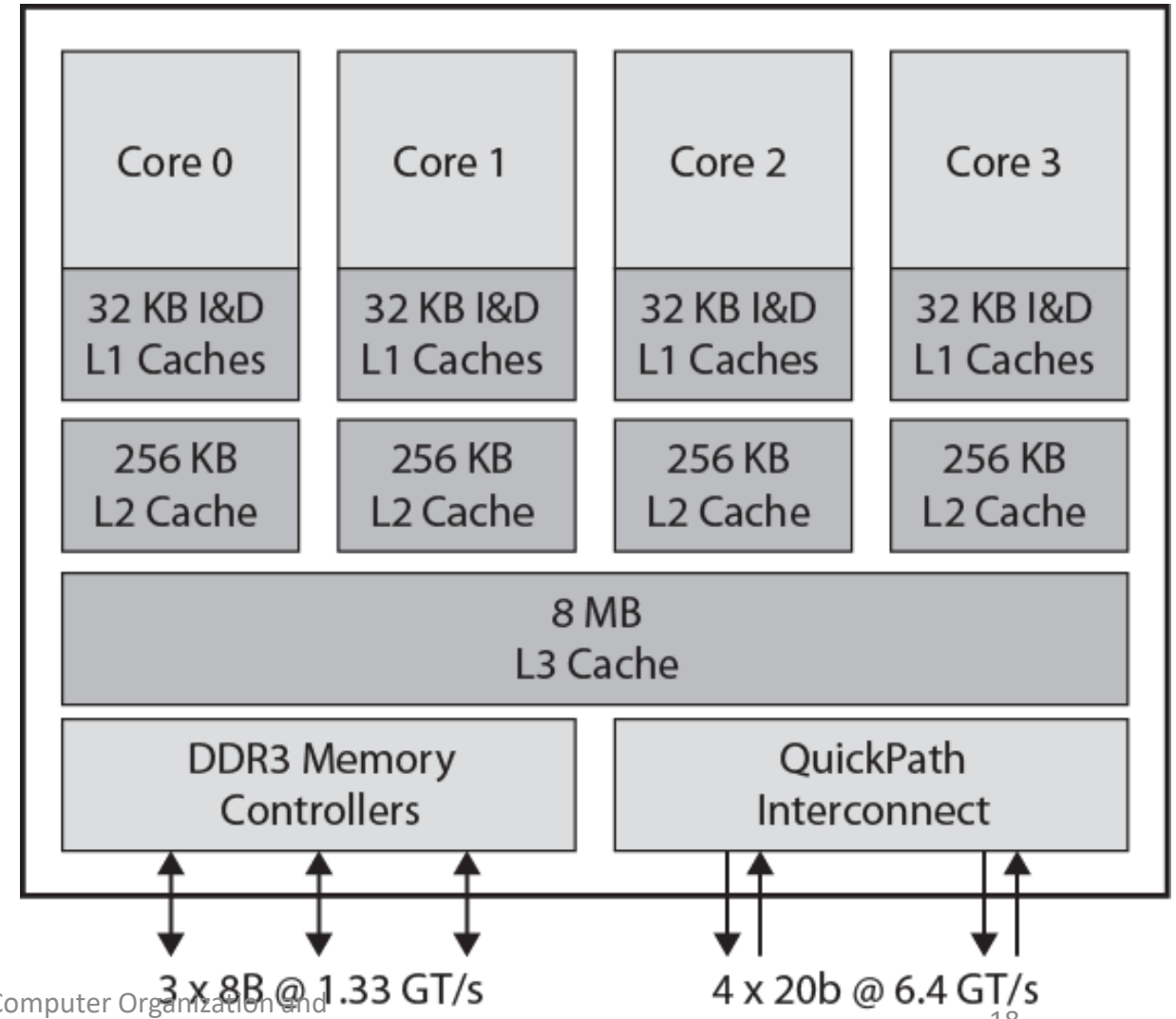
Intel x86 Multicore Organization - Core Duo

- 2006
- Two x86 superscalar, shared L2 cache
- Dedicated L1 cache per core
  - 32KB instruction and 32KB data
- 2MB shared L2 cache
- Thermal control unit per core
- Advanced Programmable Interrupt Controlled (API
- Power Management Logic
- Bus interface



Intel x86 Multicore Organization - Core i7

- November 2008
- Four x86 SMT processors
- Dedicated L2, shared L3 cache
- Speculative pre-fetch for caches
- On chip DDR3 memory controller
- QuickPath Interconnection
  - Cache coherent point-to-point link
  - High speed communications between processor chips



Main Memory

- There are main memory concerns when scaling CMP cores
- Specifically, scaling up the number of cores per a chip requires at least a proportional increase in the size of the main memory
  - This is to keep the size of the main memory per core constant
- However, the main memory size may need to grow faster than Moore's law
  - This is due to the continuing increase in software complexity (i.e., larger working sets would require larger main memories)
- If the main memory capacity grows at Moore's law while the demand for that capacity grows faster than Moore's law, the cost of the main memory will increase in the future
  - This makes memory design an increasingly essential issue

Next:

Revision