

William Stallings

Computer Organization

and Architecture

8th Edition

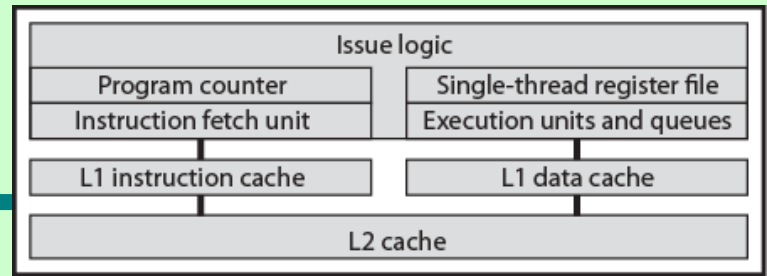
Chapter 18

Multicore Computers

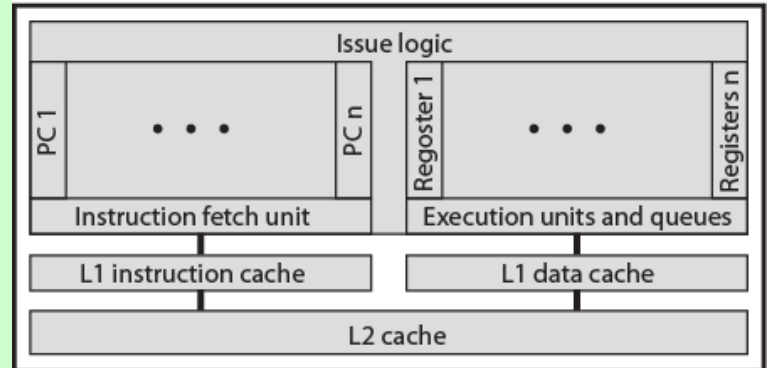
Hardware Performance Issues

- 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

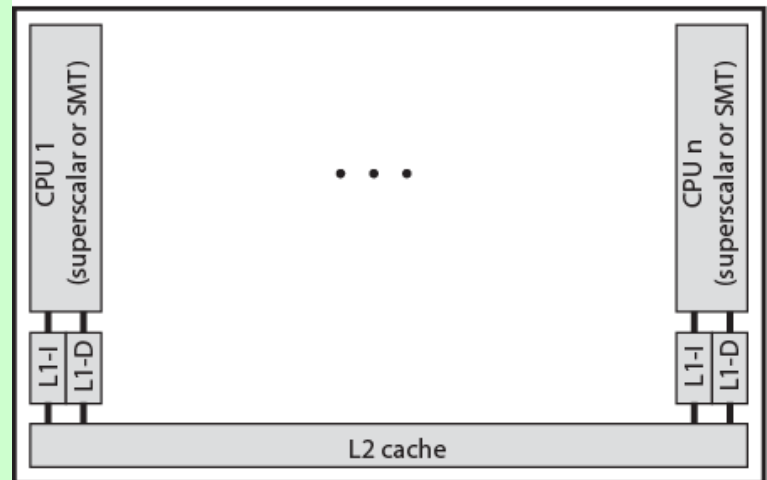
Alternative Chip Organizations



(a) Superscalar

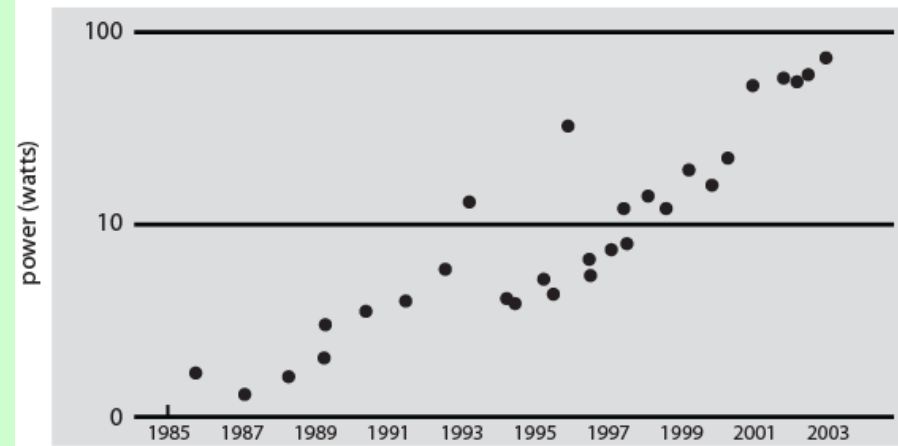
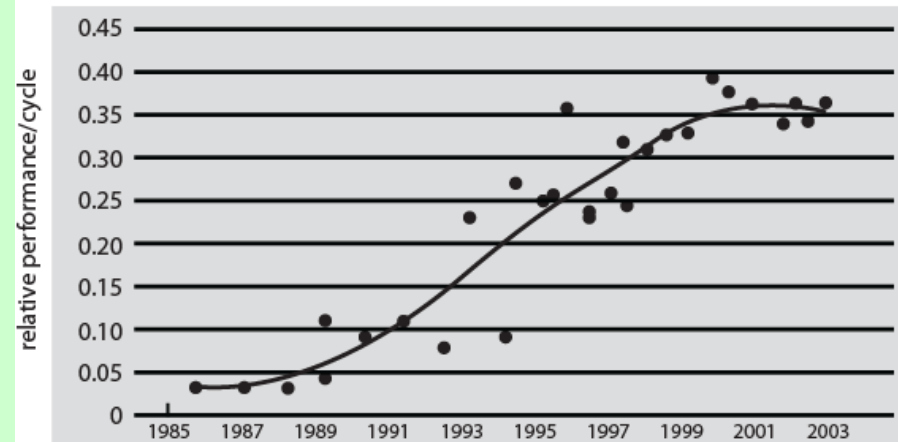
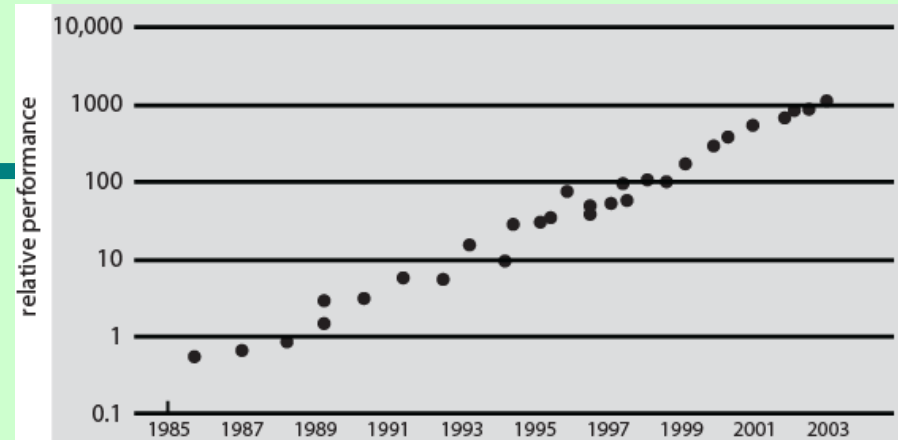


(b) Simultaneous multithreading



(c) Multicore

Intel Hardware Trends

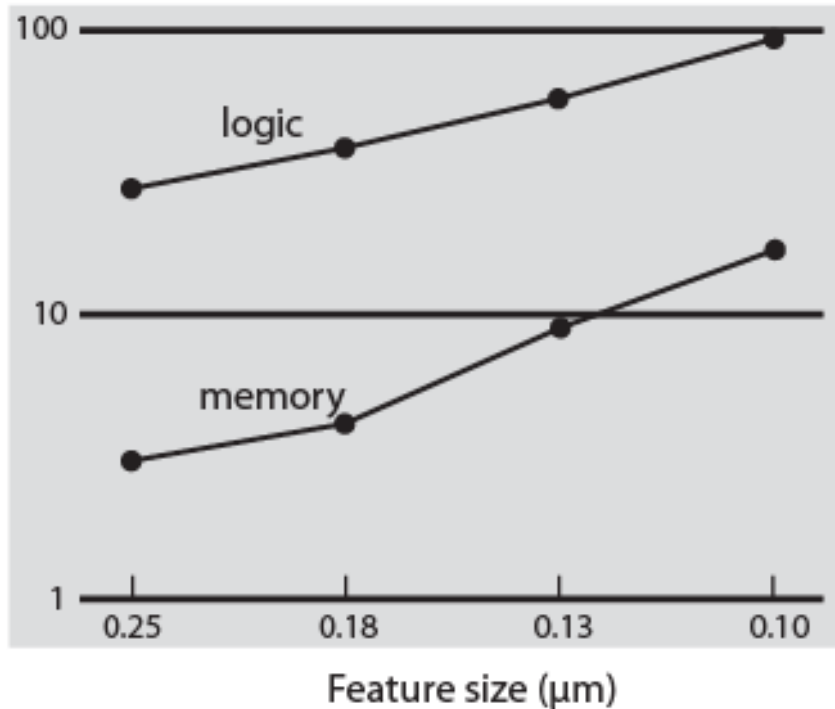


Increased Complexity

- Power requirements grow exponentially with chip density and clock frequency
 - Can use more chip area for cache
 - Smaller
 - Order of magnitude lower power requirements
- By 2015
 - 100 billion transistors on 300mm² die
 - Cache of 100MB
 - 1 billion transistors for logic
- 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

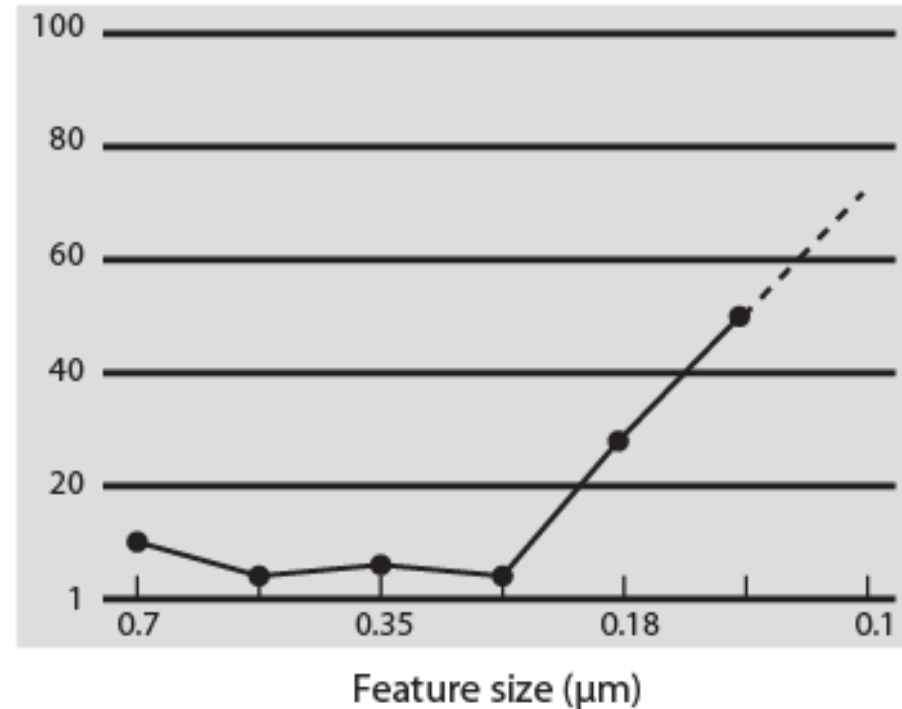
Power and Memory Considerations

Power density
(watts/cm²)



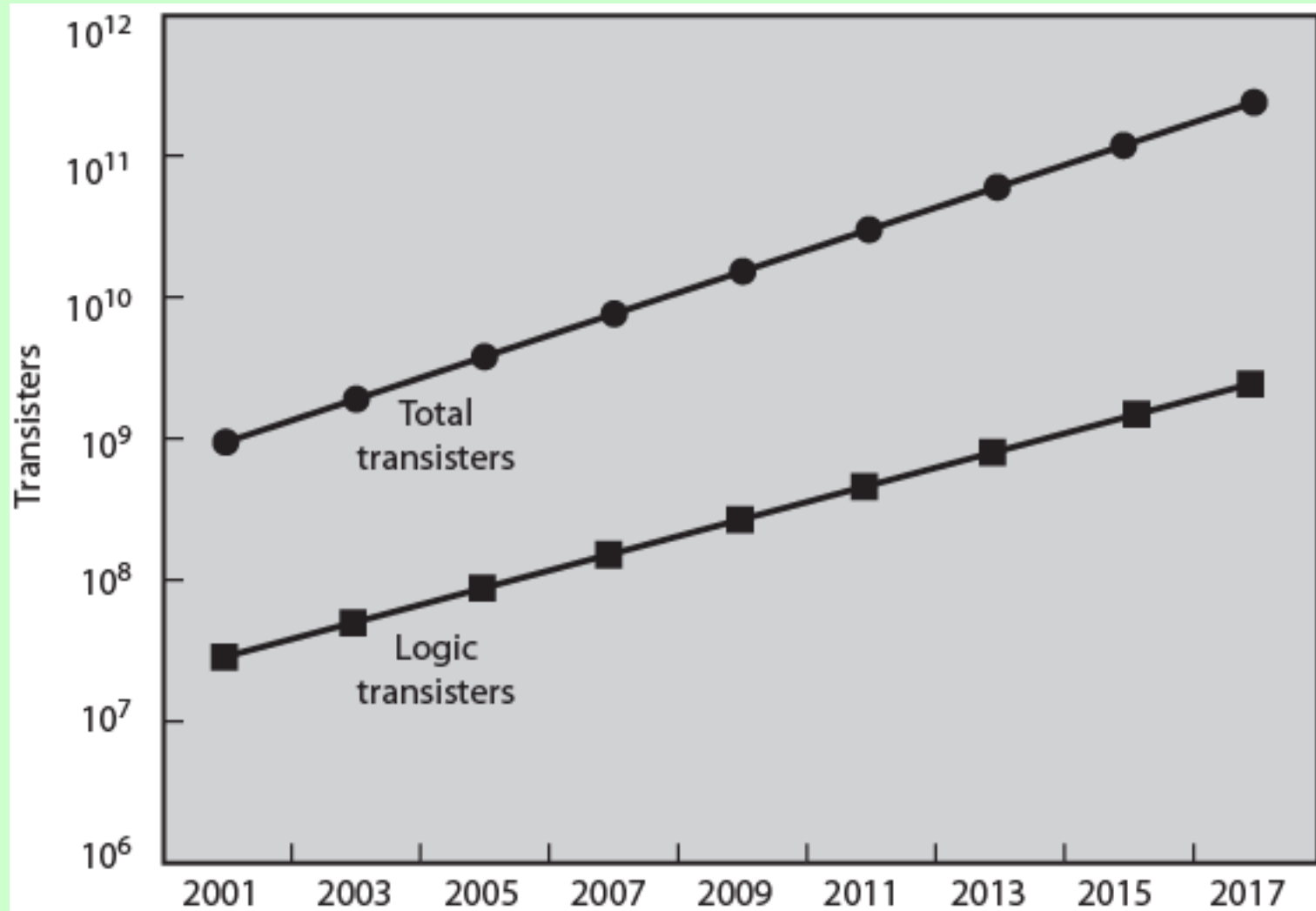
(a) Power density

cache percent
of full chip area



(b) Chip area

Chip Utilization of Transistors



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

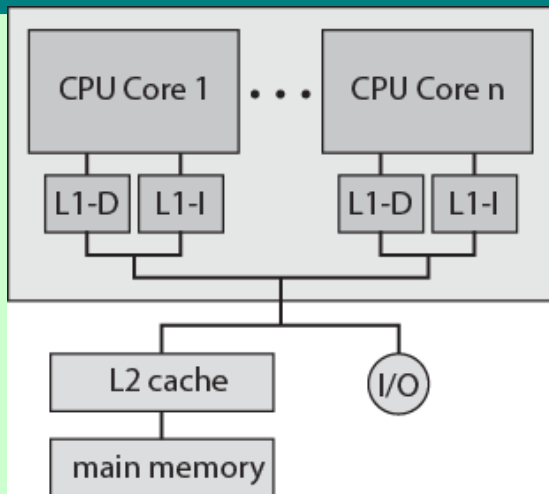
Effective Applications for Multicore Processors

- 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

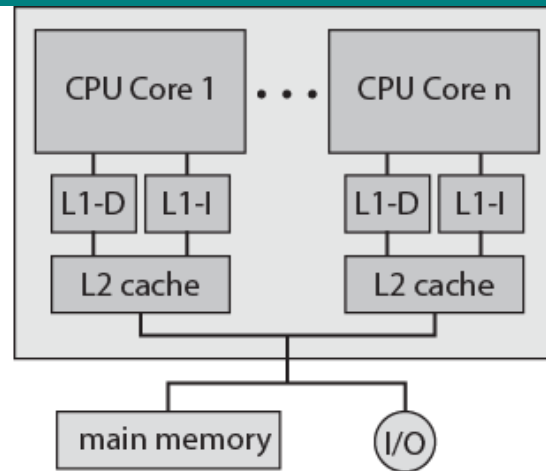
Multicore Organization

- Number of core processors on chip
- Number of levels of cache on chip
- Amount of shared cache
- Next slide examples of each organization:
 - (a) ARM11 MPCore
 - (b) AMD Opteron
 - (c) Intel Core Duo
 - (d) Intel Core i7

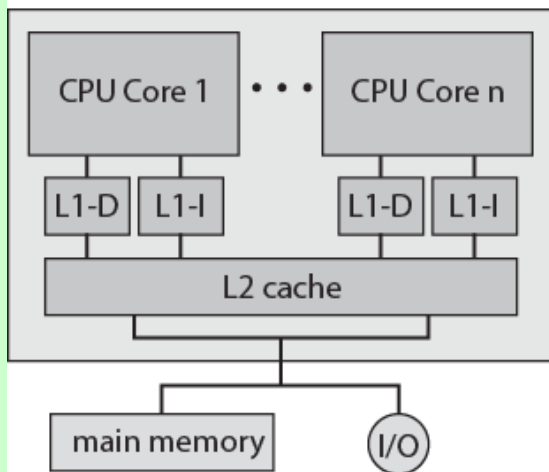
Multicore Organization Alternatives



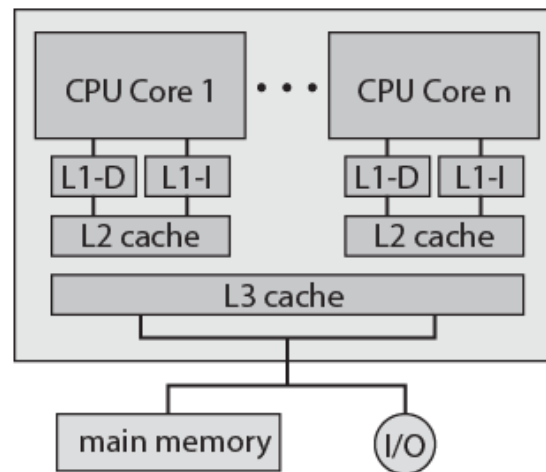
(a) Dedicated L1 cache



(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
 - Threads with less locality can have more cache
- Easy inter-process communication through shared memory
- Cache coherency confined to L1
- Dedicated L2 cache gives each core more rapid access
 - Good for threads with strong locality
- 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 (1)

- 2006
- Two x86 superscalar, shared L2 cache
- Dedicated L1 cache per core
 - 32KB instruction and 32KB data
- Thermal control unit per core
 - Manages chip heat dissipation
 - Maximize performance within constraints
 - Improved ergonomics
- Advanced Programmable Interrupt Controlled (APIC)
 - Inter-process interrupts between cores
 - Routes interrupts to appropriate core
 - Includes timer so OS can interrupt core

Intel x86 Multicore Organization - Core Duo (2)

- Power Management Logic
 - Monitors thermal conditions and CPU activity
 - Adjusts voltage and power consumption
 - Can switch individual logic subsystems
- 2MB shared L2 cache
 - Dynamic allocation
 - MESI support for L1 caches
 - Extended to support multiple Core Duo in SMP
 - L2 data shared between local cores or external
- Bus interface

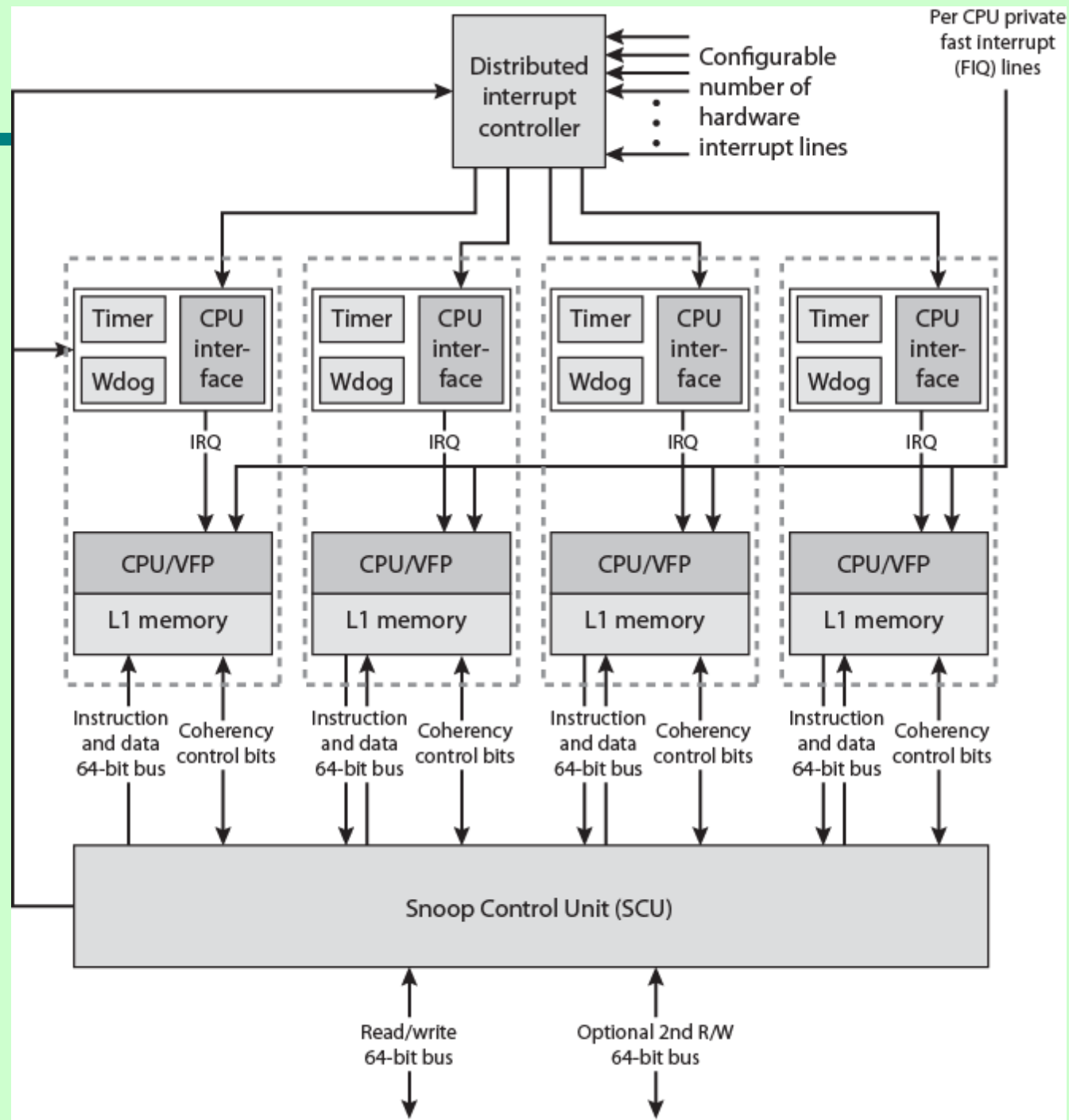
Intel x86 Multicore Organization - Core i7

- November 2008
- Four x86 SMT processors
- Dedicated L2, shared L3 cache
- Speculative pre-fetch for caches
- On chip DDR3 memory controller
 - Three 8 byte channels (192 bits) giving 32GB/s
 - No front side bus
- QuickPath Interconnection
 - Cache coherent point-to-point link
 - High speed communications between processor chips
 - 6.4G transfers per second, 16 bits per transfer
 - Dedicated bi-directional pairs
 - Total bandwidth 25.6GB/s

ARM11 MPCore

- Up to 4 processors each with own L1 instruction and data cache
- Distributed interrupt controller
- Timer per CPU
- Watchdog
 - Warning alerts for software failures
 - Counts down from predetermined values
 - Issues warning at zero
- CPU interface
 - Interrupt acknowledgement, masking and completion acknowledgement
- CPU
 - Single ARM11 called MP11
- Vector floating-point unit
 - FP co-processor
- L1 cache
- Snoop control unit
 - L1 cache coherency

ARM11 MPCore Block Diagram



ARM11 MPCore Interrupt Handling

- Distributed Interrupt Controller (DIC) collates from many sources
- Masking
- Prioritization
- Distribution to target MP11 CPUs
- Status tracking
- Software interrupt generation
- Number of interrupts independent of MP11 CPU design
- Memory mapped
- Accessed by CPUs via private interface through SCU
- Can route interrupts to single or multiple CPUs
- Provides inter-process communication
 - Thread on one CPU can cause activity by thread on another CPU

DIC Routing

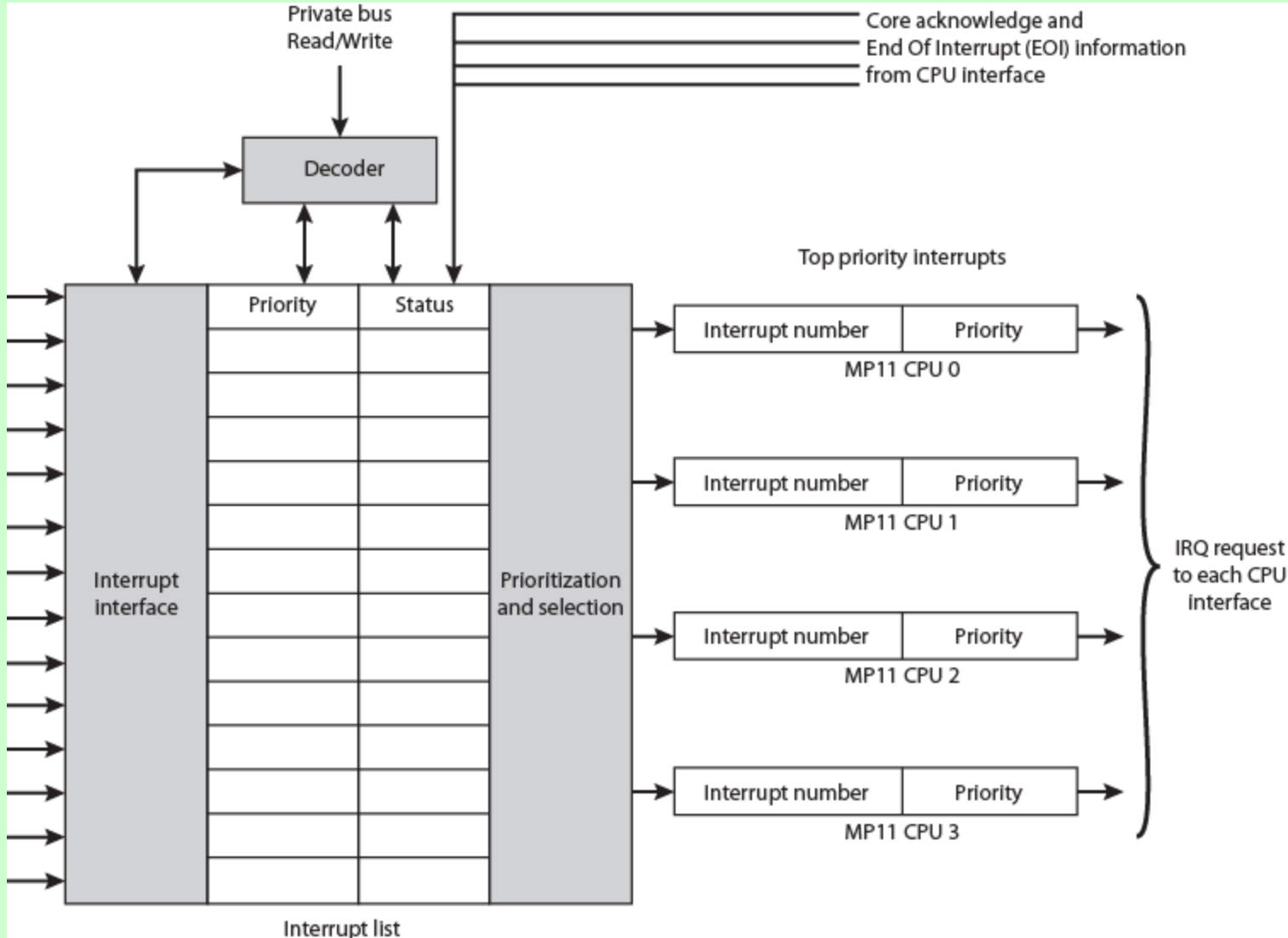
- Direct to specific CPU
- To defined group of CPUs
- To all CPUs
- OS can generate interrupt to:
 - All but self
 - Self
 - Other specific CPU
- Typically combined with shared memory for inter-process communication
- 16 interrupt ids available for inter-process communication

Interrupt States

- Inactive
 - Non-asserted
 - Completed by that CPU but pending or active in others
- Pending
 - Asserted
 - Processing not started on that CPU
- Active
 - Started on that CPU but not complete
 - Can be pre-empted by higher priority interrupt

Interrupt Sources

- Inter-process Interrupts (IPI)
 - Private to CPU
 - ID0-ID15
 - Software triggered
 - Priority depends on target CPU not source
- Private timer and/or watchdog interrupt
 - ID29 and ID30
- Legacy FIQ line
 - Legacy FIQ pin, per CPU, bypasses interrupt distributor
 - Directly drives interrupts to CPU
- Hardware
 - Triggered by programmable events on associated interrupt lines
 - Up to 224 lines
 - Start at ID32



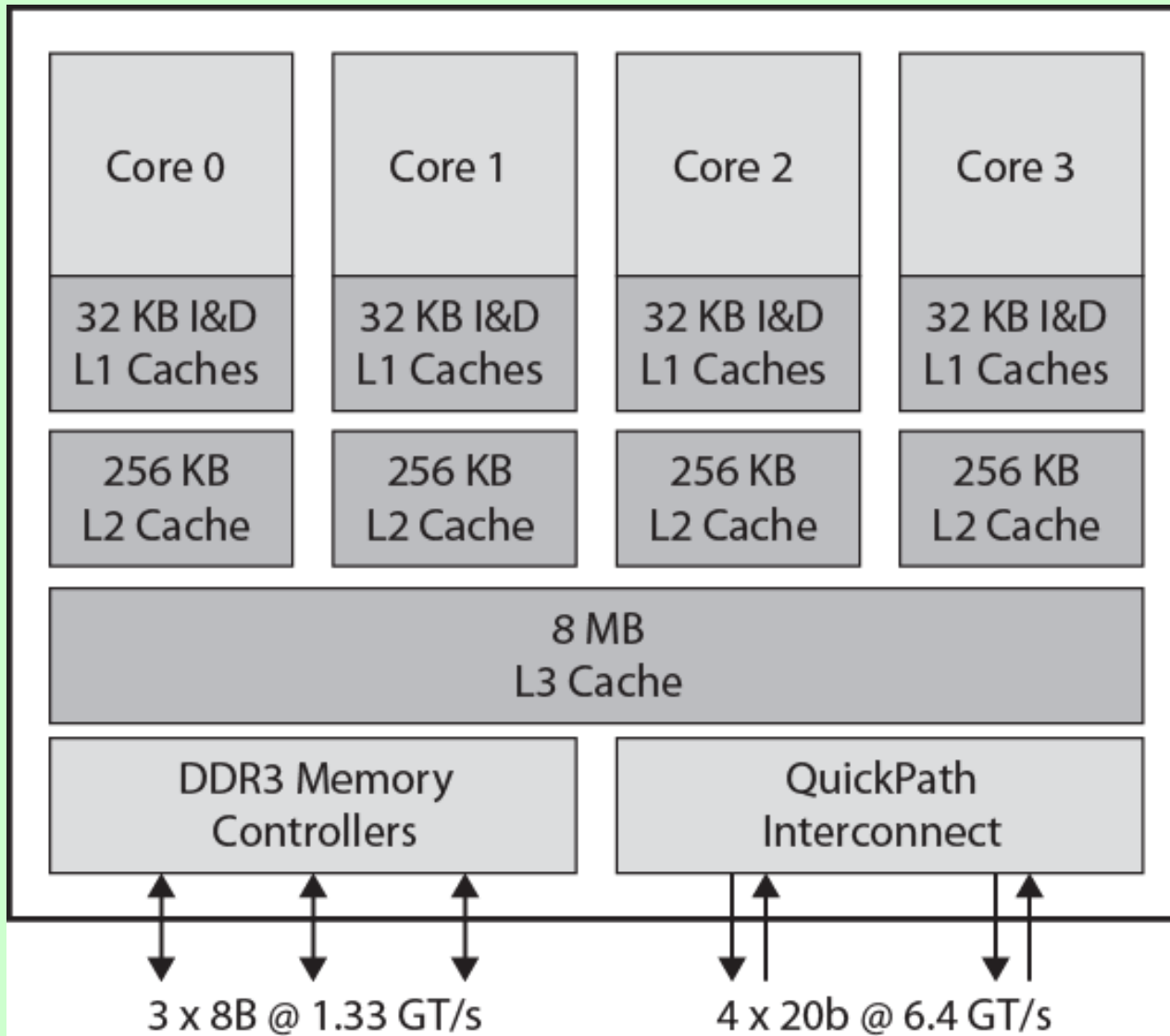
Cache Coherency

- Snoop Control Unit (SCU) resolves most shared data bottleneck issues
- L1 cache coherency based on MESI
- Direct data Intervention
 - Copying clean entries between L1 caches without accessing external memory
 - Reduces read after write from L1 to L2
 - Can resolve local L1 miss from remote L1 rather than L2
- Duplicated tag RAMs
 - Cache tags implemented as separate block of RAM
 - Same length as number of lines in cache
 - Duplicates used by SCU to check data availability before sending coherency commands
 - Only send to CPUs that must update coherent data cache
- Migratory lines
 - Allows moving dirty data between CPUs without writing to L2 and reading back from external memory

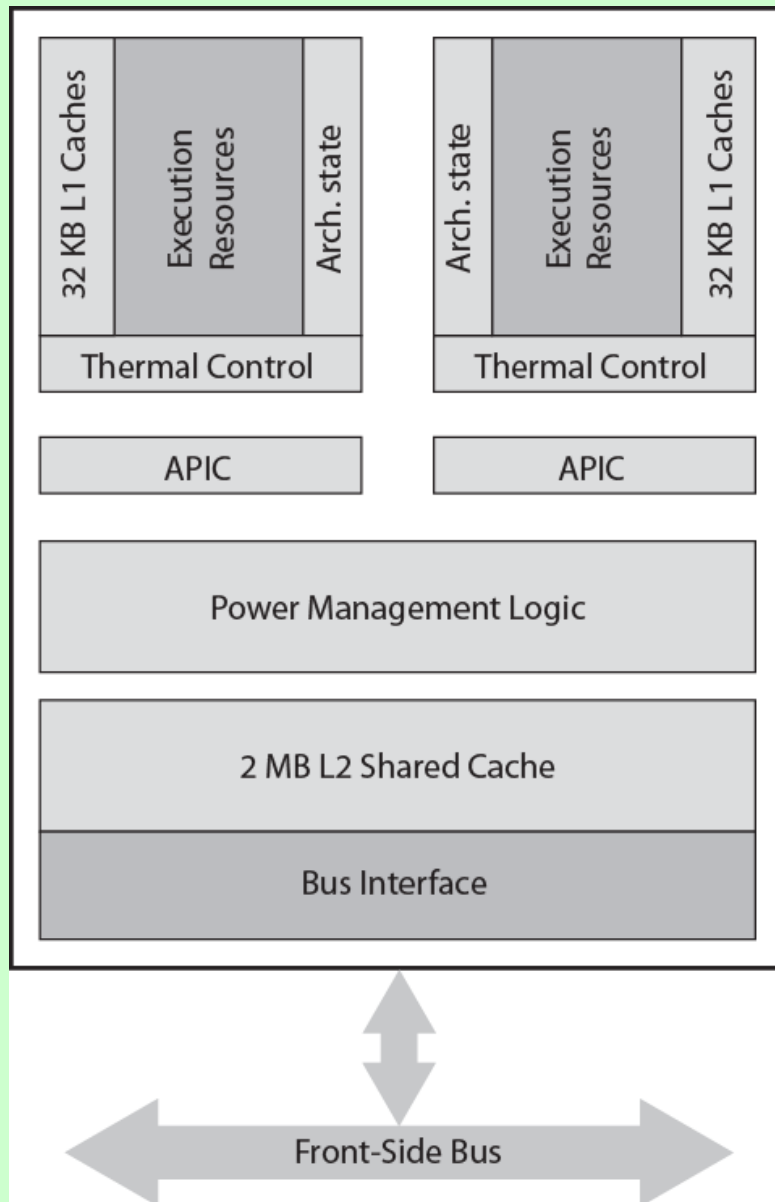
Recommended Reading

- Stallings chapter 18
- ARM web site

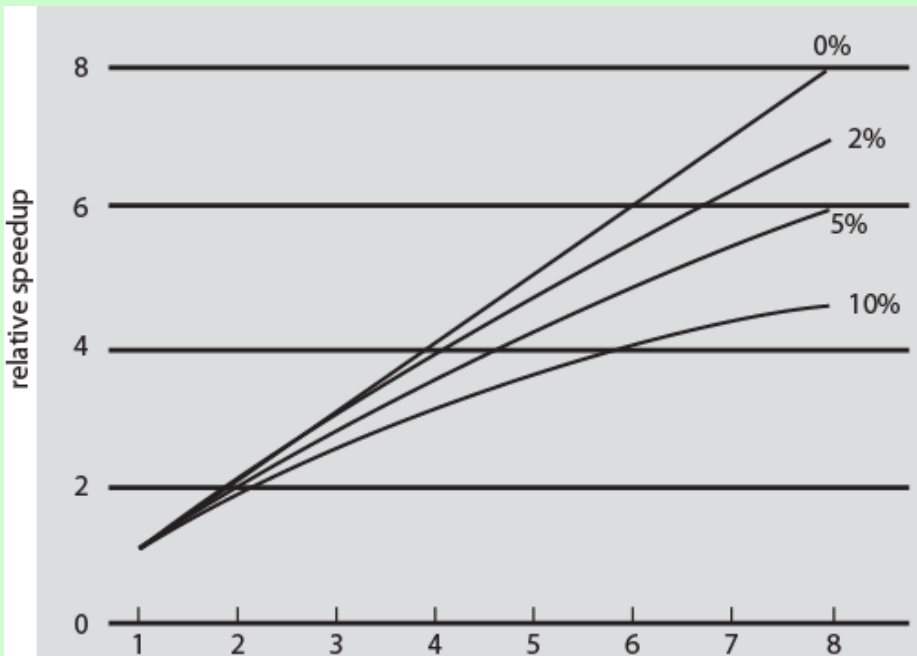
Intel Core i& Block Diagram



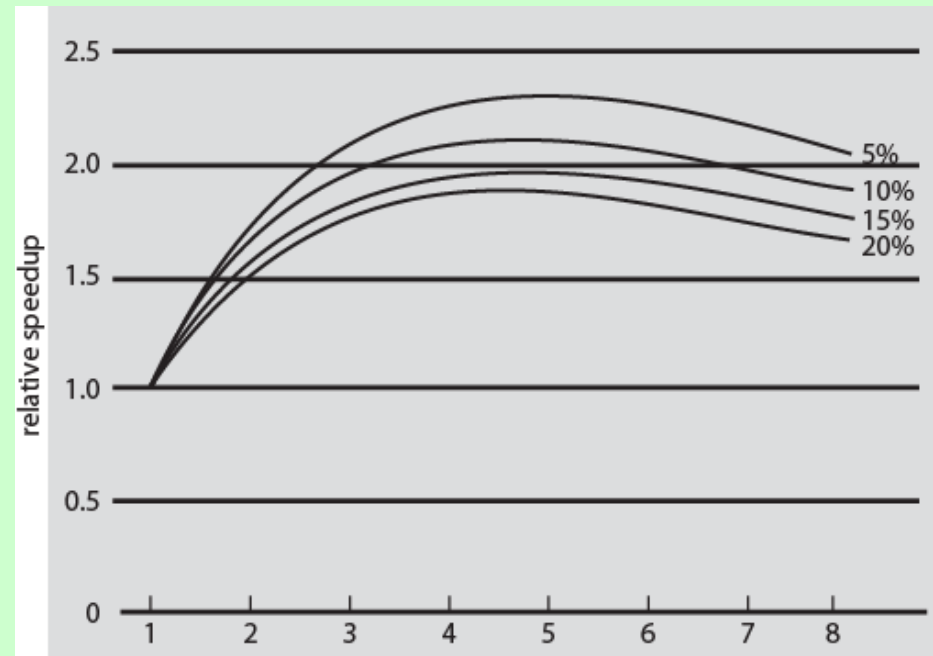
Intel Core Duo Block Diagram



Performance Effect of Multiple Cores



(a) Speedup with 0%, 2%, 5%, and 10% sequential portions



(b) Speedup with overheads

Recommended Reading

- Multicore Association web site
- ARM web site