MULTICORE COMPUTERS

HARDWARE PERFORMANCE ISSUES

- Power Consumption

SOFTWARE PERFORMANCE ISSUES

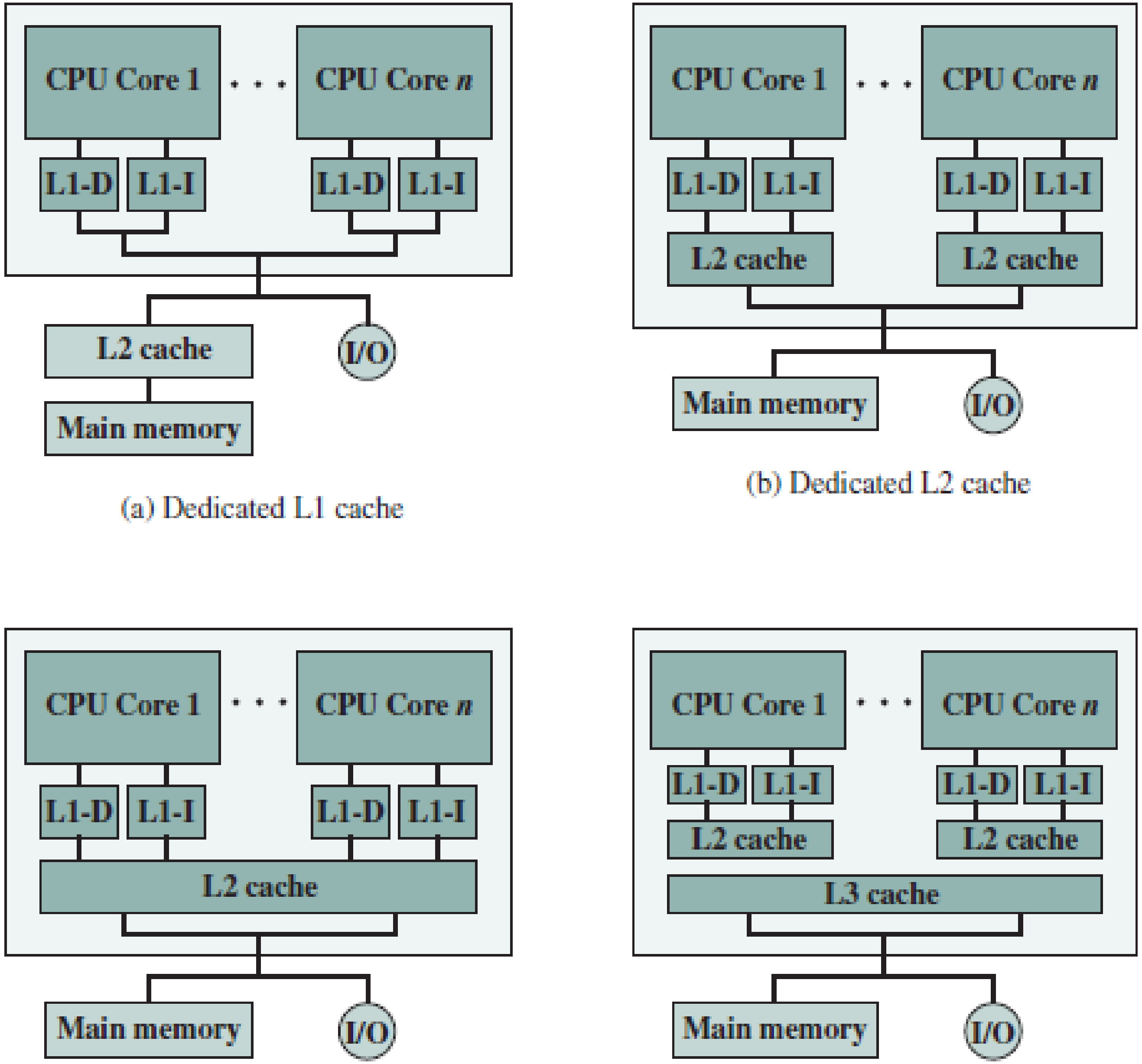
- Software on Multicore

MULTICORE ORGANIZATION

At a top level of description, the main variables in a multicore organization are as follows:

- □ The number of core processors on the chip
- □ The number of levels of cache memory
- □ How cache memory is shared among cores
- □ Whether simultaneous multithreading (SMT) is employed
- □ The types of cores

Levels of Cache



(a) Dedicated L1 cache

(b) Dedicated L2 cache

(c) Shared L2 cache

(d) Shared L3 cache

Figure 18.6 Multicore Organization Alternatives

HETEROGENEOUS MULTICORE ORGANIZATION

- A typical case for the use of multiple cores is a chip with multiple identical cores, known as homogenous multicore organization.
- To achieve better results, in terms of performance and/or power consumption, an increasingly popular design choice is heterogeneous multicore organization, which refers to a processor chip that includes more than one kind of core.

GPU

- The graphics processor unit (GPU) is designed specifically to be optimized for fast three- dimensional (3D) graphics rendering and video processing. GPUs can be found in almost all of today’s workstations, laptops, tablets, and smartphones [OWEN08].
- The GPU comes in many sizes. The larger units have several hundred to thousands of parallel processor cores on a single integrated circuit (IC).
- Over the past several years, the GPU has found its way into massively parallel programming environments for a wide range of applications, such as bioinformatics, molecular dynamics, oil and gas exploration, computational finance, signal and audio processing, statistical modeling, computer vision, and medical imaging.

GPU VERSUS CPU

Architectures

- In the CPU, the control logic and cache memory make up the majority of the CPU. This is as expected for an architecture which is tuned to process sequential code as quickly as possible.
- A GPU uses a massively parallel SIMD (single instruction multiple data) architecture to perform mainly mathematical operations. A GPU doesn’t require the same complex capabilities of the CPU’s control logic, nor does it require large amounts of cache memory.

GPU VERSUS CPU

Performance

- The GPU is designed to maximize the number of floating-point operations per second (FLOPs) it can perform.

Performance per Watt Comparison

- Newer NVIDIA architectures, such as the Kepler and Maxwell architectures, have focused on increasing the performance per watt ratio (FLOPs/watt) over previous GPU architectures by decreasing the power required by each GPU processor core.