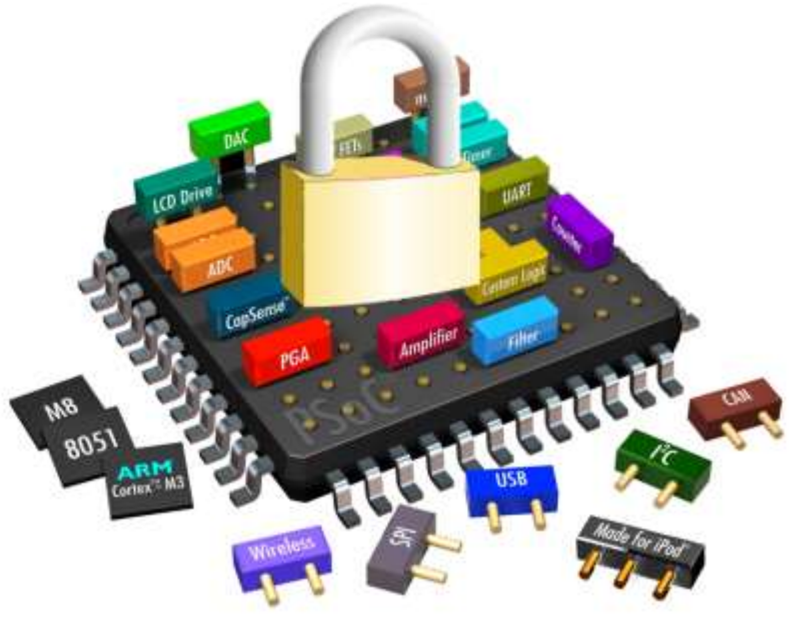
KOREA UNIVERSITY

Advanced Computer Architecture

Thread-Level Parallelism



Shift to Multiprocessing

"We are dedicating all of our future product development to multicore designs. We believe this is a key inflection point for the industry."

— Intel President Paul Otellini, Intel Developer Forum (2005)

Historical Context

While some researchers predicted the end of uniprocessor advances as early as the 1960s, the period of 1986-2003 actually saw the highest performance growth since the first transistorized computers.

The Inflection Point

By 2005, the industry recognized that multiprocessors would play a major role from low-end to high-end computing, marking a clear shift in architectural focus.

This transition wasn't just a technical choice—it represented a fundamental rethinking of how to advance computing performance in the face of physical and practical limitations.

Why Multiprocessing Became Essential

Diminishing Returns

Attempts to extract more instruction-level parallelism (ILP) led to dramatically lower efficiencies in silicon and energy use between 2000-2005, as power and silicon costs grew faster than performance.

Cloud Computing Growth

Increasing interest in high-end servers as cloud computing and software-as-a-service became more important to the computing landscape.

Data-Intensive Applications

Growth in applications driven by the availability of massive amounts of data on the Internet requiring parallel processing capabilities.

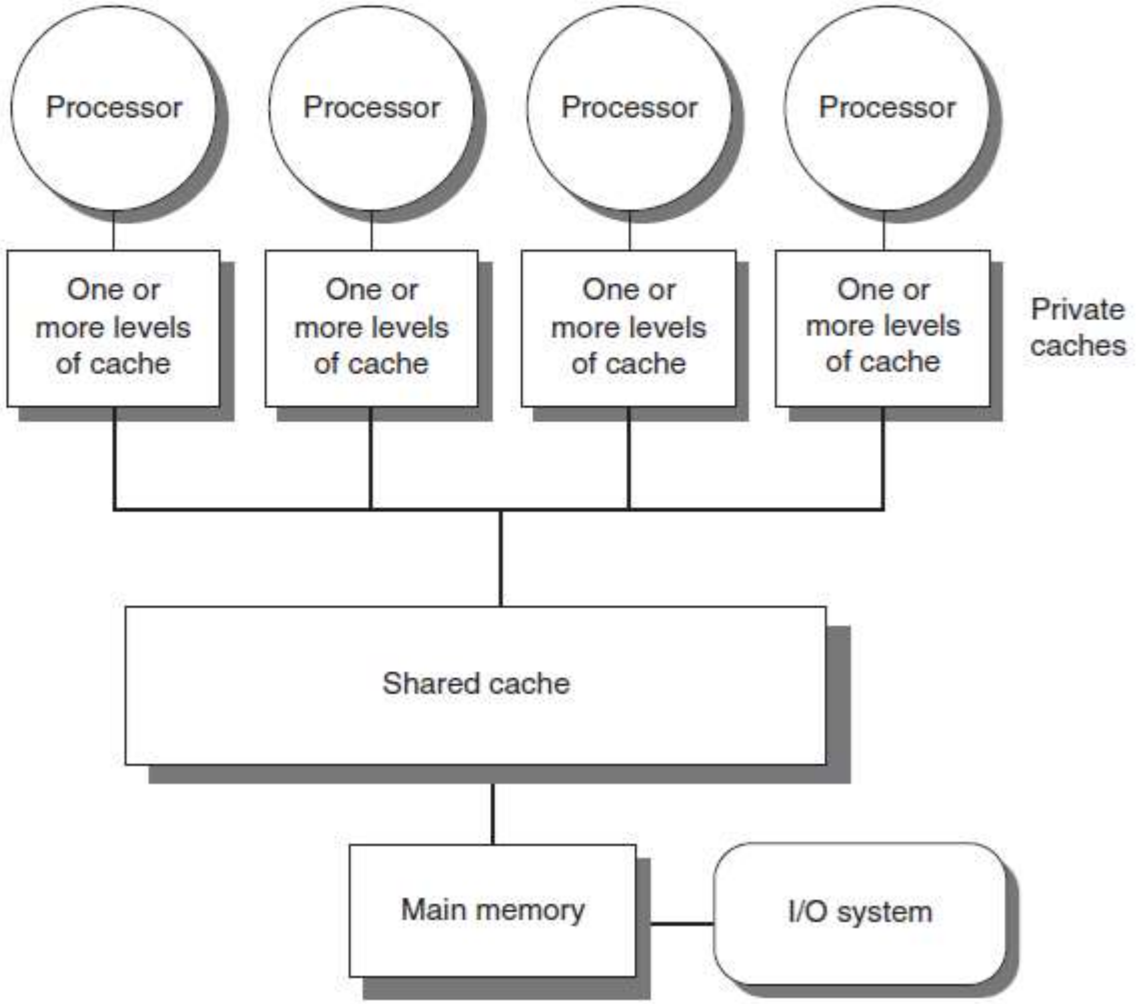
Desktop Performance Plateau

Recognition that increasing desktop performance was less important (outside of graphics), as current performance became acceptable or compute-intensive tasks moved to the cloud.

Additional Drivers of Multiprocessing

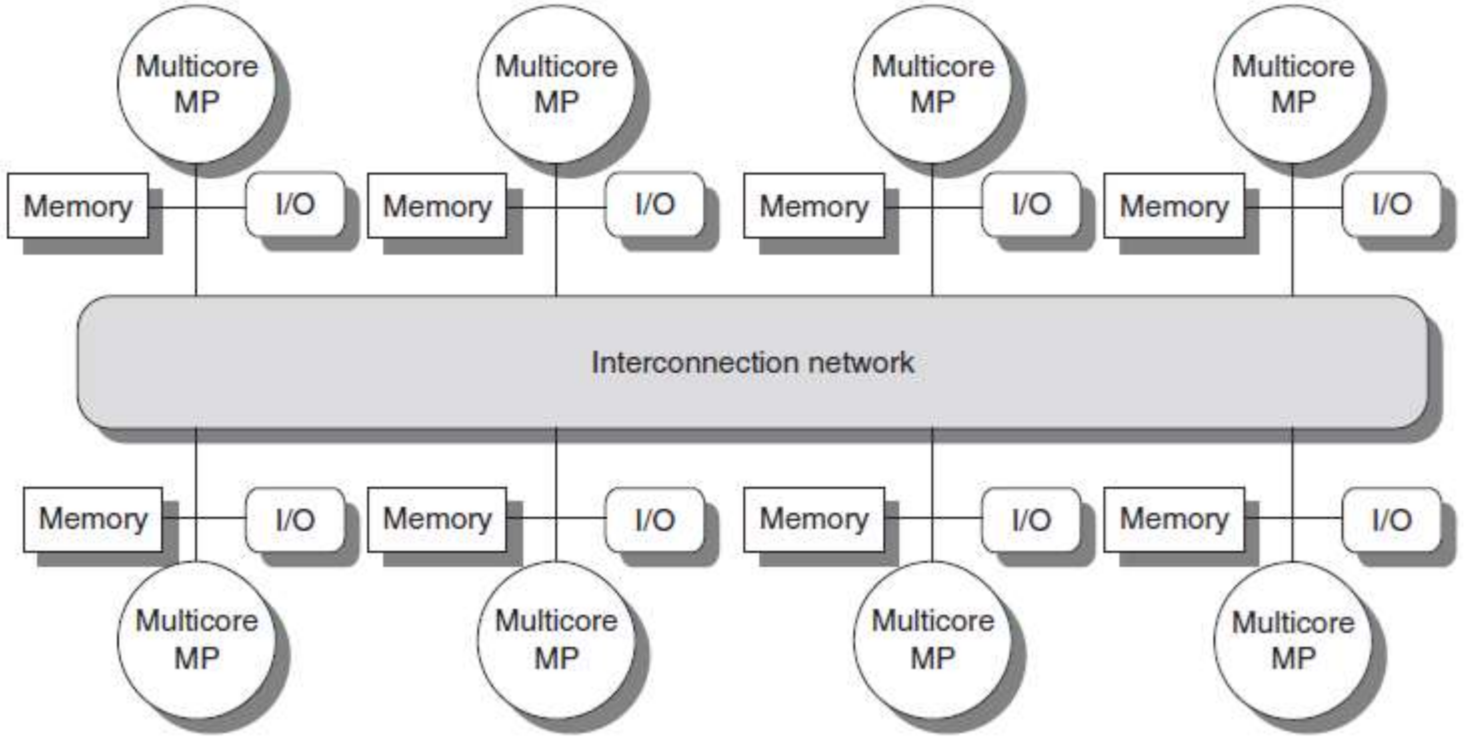
- Improved Understanding
  - Better knowledge of how to use multiprocessors effectively, especially in server environments with significant natural parallelism from large datasets or independent requests.
- Design Leverage
  - The advantages of leveraging a design investment by replication rather than unique design; all multiprocessor designs provide such leverage.
- These factors collectively pushed the industry toward thread-level parallelism as the primary means of performance scaling, fundamentally changing how processors are designed and programmed.

Symmetric Multiprocessors (SMPs)


Processor
Processor
Processor
Processor
One or more levels of cache
One or more levels of cache
One or more levels of cache
One or more levels of cache
Private caches
Shared cache
Main memory
I/O system

- Small number of cores (typically ≤8)
- Centralized shared memory with equal access
- Also called Uniform Memory Access (UMA)
- All existing multicores are SMPs

Distributed Shared Memory (DSM)



- Physically distributed memory among processors
- Also called Non-Uniform Memory Access (NUMA)
- Required for larger processor counts
- Multi-chip multiprocessors use distributed memory

Challenge #1: Limited Parallelism

80x	0.25%	99.75%
Target Speedup	**Sequential Limit**	**Parallel Requirement**
With 100 processors	Maximum sequential portion allowed	Portion that must be parallel

- Amdahl's Law shows that even a small sequential portion of a program severely limits potential speedup. To achieve 80x speedup with 100 processors, only 0.25% of the original computation can be sequential.
- This fundamental limitation makes it difficult to achieve good speedups in any parallel processor, regardless of architecture.

Challenge #2: Communication Latency

1

Latency Costs

Communication between cores may cost 35-50 clock cycles, and between chips 100-500+ cycles depending on the mechanism and scale.

2

Performance Impact

Even with just 0.2% of instructions requiring remote communication, performance can drop by 3.4x compared to all-local references.

3

Contention Effects

In practice, contention for the interconnect can make remote access latency even worse than the base values.

These long-latency remote communications represent one of the biggest performance challenges in multiprocessor systems, requiring both architectural and software solutions.