

# **Multi Processors, their Memory organizations and Implementations by Intel & AMD**

**Abstract:** Multi-core processors represent a major evolution in computing technology and are becoming very popular today. Multi-core processors will eventually become the pervasive computing model because they offer performance and productivity benefits beyond the capabilities of today's single-core processors. This paper initially looks at the motivation and the advantages offered by multiprocessors over uni processors. We then look at the multiprocessors' memory organizations, cache coherence and its implementations by today's leading chip manufacturers Intel and AMD and how they deal with these problems.

## **Introduction:**

### **1. Motivation and the need for Multiprocessors:**

In today's digital world the demands of complex 3D simulations, streaming media files, added levels of security, more sophisticated user interfaces, larger databases, and more on-line users are beginning to exceed single-core processor capabilities.

Multi-core processors enable true multitasking. On single-core systems, multitasking can max out CPU utilization, resulting in decreased performance as operations have to wait to be processed. On multi-core systems, since each core has its own cache, the operating system has sufficient resources to handle most compute intensive tasks in parallel.

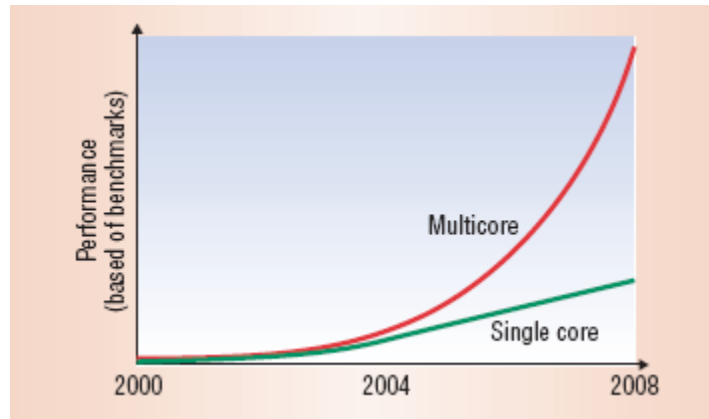
In todays and future world, parallel processors will definitely have a bigger role. There are three reasons to this.

- 1) Since Microprocessors are likely to remain the dominant uniprocessor technology, the logical way to improve performance beyond a single processor is by connecting multiple microprocessors together.
- 2) The combination is likely to be more cost effective.
- 3) Exploitation of Instruction level parallelism.

Server and embedded applications exhibit natural parallelism that can be exploited.

### **A Fundamental Theorem of Multi-Core Processors:**

Multi-core processors take advantage of a fundamental relationship between power and frequency. By incorporating multiple cores, each core is able to run at a lower frequency, dividing among them the power normally given to a single core. The result is a big performance increase over a single core processor. The following illustration—based on lab experiments with commonly used workloads—illustrates this key advantage.



**Figure 1 - Multi core performance compared to single core.**

Multi-core technology can improve system efficiency and application performance for computers running multiple applications at the same time.

### **2. Benefits of Multi-Core Technology: The Multi-Core Advantage**

- Improved system efficiency and application performance for computers running multiple applications.
- Enhanced performance for multi-threaded applications.
- Support for more users or tasks for transaction-intensive applications.
- Superior performance for compute-intensive applications.
- Simplified overall computing infrastructure requirements helping to save you money.
- Helps to eliminate thermal and environmental issues.

### **3. Multiprocessors Characteristics:**

Multiple Instruction Multiple data (MIMD): Each processor fetches its own instructions and operates its own data.

MIMD offers flexibility. With an MIMD, each processor is executing its own instruction stream. We categorize these MIMD multiprocessors depending on number of processors involved, which in turn dictate a memory organization and interconnect strategy.

#### **Centralized shared memory:**

Multiple processor cache access the same physical memory and are often called symmetric (shared memory) multiprocessors and this style of architecture is called as Uniform memory access (UMA)

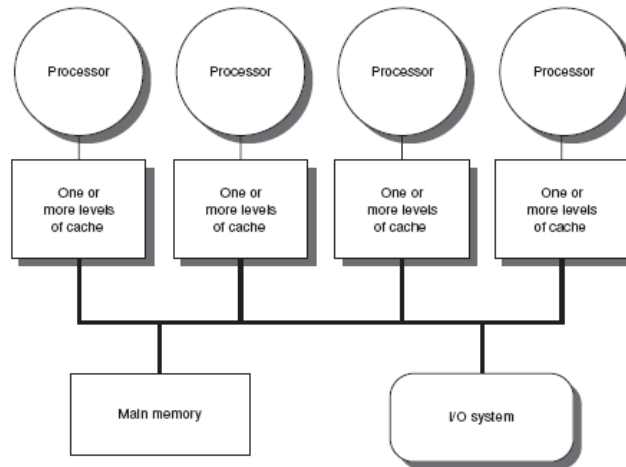


Figure 2 - Multiple processor cache subsystems share the same physical memory, typically connected to a bus.

**Distributed Memory:**

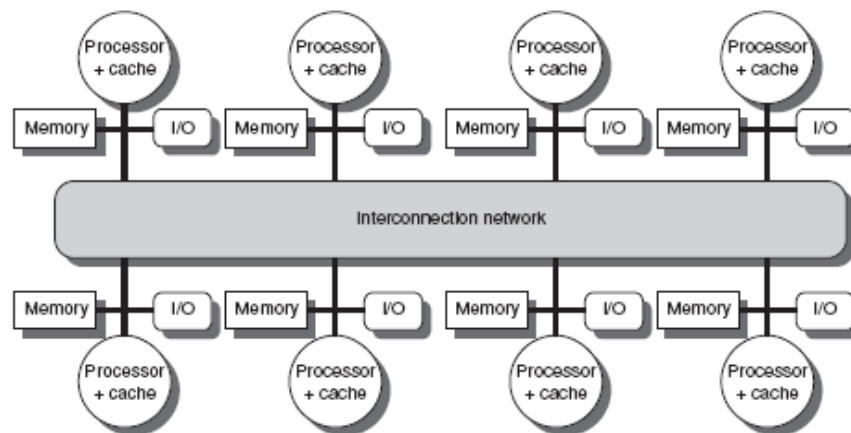


Figure 3 - Basic architecture of a distributed memory system MP consisting of individual nodes.

To support larger processor counts, memory must be distributed among the processors rather than centralized; otherwise the memory system would not be able to support the bandwidth demands of a larger number of processors without incurring excessively long access latency.

**Advantages:** Distributing the memory among the nodes has two major benefits. First, it is a cost-effective way to scale the memory bandwidth if most of the accesses are to the local memory in the node. Second, it reduces the latency for accesses to the local memory. These two advantages make distributed memory attractive at smaller processor counts as processors get ever faster and require more memory bandwidth and lower memory latency.

**Disadvantage:** Communication between processors becomes more complex and has higher latency because the processors no longer share a single, centralized memory.

**Importance of Cache:** As the number of the core increase, the communications among cores also become complex and difficult. Caches are used in multicore processors for sharing data and increasing performance. It becomes a channel for cores to communicate with each other.

**Cache coherence:** Two or more different processors having two different values for the same location is referred to as cache coherence. Caches are critical to modern high-speed processors as multiple copies of a block can easily get inconsistent. As there are many cores on the chip, and different cores could read/write the same address, it will cause the cache coherence problem.

The common way to solve the problem is using the MESI (Modified, Exclusive, Shared, Invalid) protocol. The MESI protocol has four states:

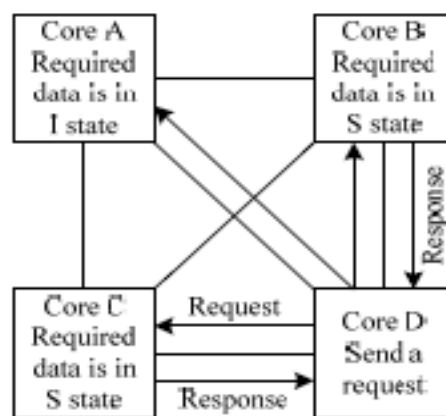


Figure 4 - MESI protocol

**Modified:** The cache line resides exclusively in this cache only, and the content is modified relative to memory

**Exclusive:** The cache line resides exclusively in this cache only, and the content is same as memory

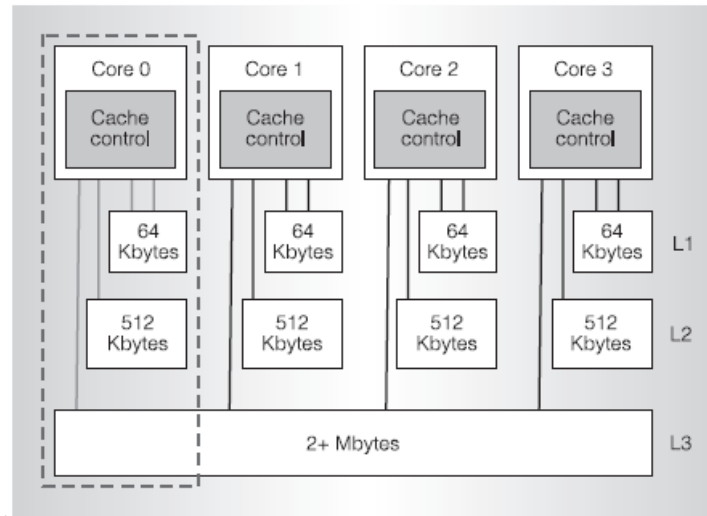
**Shared:** The cache line resides in this cache is shared with other caches. And content is same as memory

**Invalid:** The cache line contains no valid memory copy.

Now that we have defined the memory organizations in general multiprocessors and looked at how cache coherence is taken care of by the MESI protocol, let us look at the cache organization and the way cache coherence is solved by today's leading chip manufacturers Intel and AMD. We will consider the latest multi-core models i.e. the AMD Opteron and Intel Nehalem.

#### 4. Implementations:

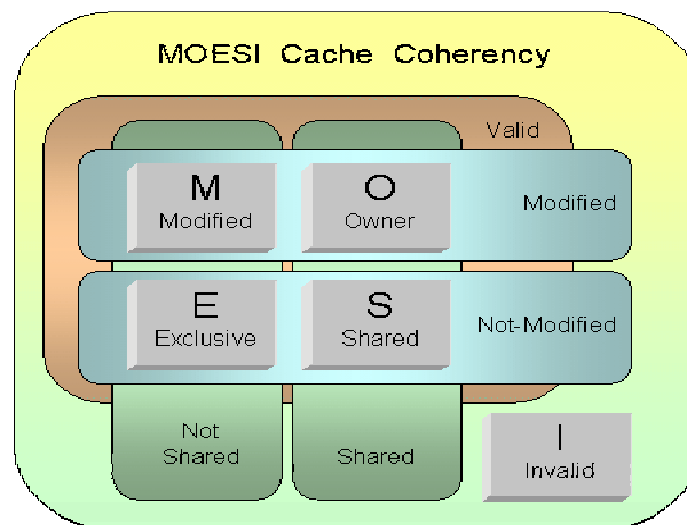
## AMD Opteron:



**Figure 5 - AMD's three level cache hierarchy**

The design has a three-level cache hierarchy as shown in Figure 5. Each core has separate L1 data and instruction caches of 64 Kbytes each. Each core also has a dedicated 512- Kbyte L2 cache, sized to accommodate most workloads. All cores share a common L3 victim cache that resides logically in the Northbridge SRI unit. The L3 cache is *noninclusive*, allowing a line to be present in an upper level L1 or L2 cache and not be present in the L3. This increases the maximum number of unique cache lines that can be cached on a node to the sum of the individual L3, L2, and L1 cache capacities (in contrast, the maximum number of distinct cache lines that can be cached with an inclusive L3 is simply the L3 capacity).

## **Cache Coherency problem in AMD Opteron:**



**Figure 6 - MOESI protocol in AMD Opteron used for Cache Coherence**

A cache line in the owned state holds the most recent, correct copy of the data. The owned state is similar to the shared state in that other processors can hold a copy of the most recent, correct data. Unlike the shared state, however, the copy in main memory can be stale (incorrect). Only one processor can hold the data in the owned state—all other processors must hold the data in the shared state.

This protocol, a more elaborate version of the simpler MESI protocol, avoids the need to write modifications back to main memory when another processor tries to read it. Instead, the Owned state allows a processor to supply the modified data directly to the other processor. This is beneficial when the communication latency and bandwidth between two CPUs is significantly better than to main memory.

If a processor wishes to write to an Owned cache line, it must notify the other processors that are sharing that cache line. Depending on the implementation it may simply tell them to invalidate their copies (moving its own copy to the Modified state), or it may tell them to update their copies with the new contents (leaving its own copy in the Owned state).

#### Intel Nehalem:

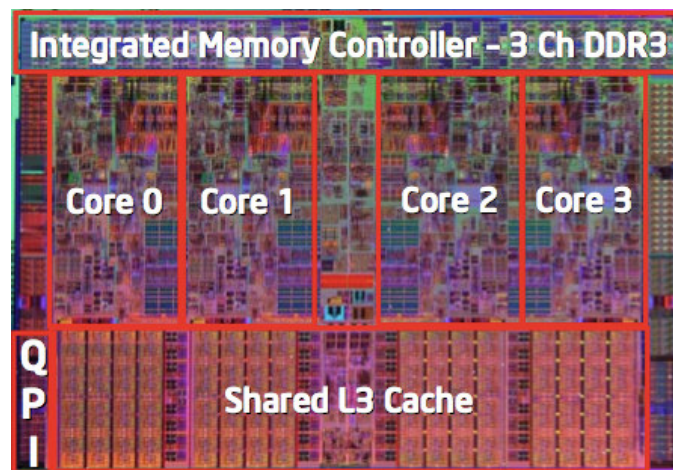
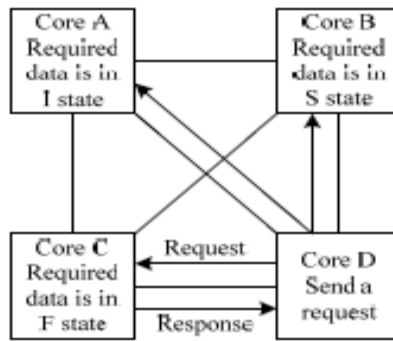


Figure 7 - Intel Nehalem's three level cache hierarchy

The characteristic of the inclusive cache is indicated by the operation. There is a data request send to L3 cache when one of the four cores has a miss. For example if the data is not in the L1 and L2 cache of the core A which will cause a miss and send the request to L3 cache. The L3 cache will reply a hit, when the requested data is in the L3 cache, or it will give a miss. If it's a miss, as there has 3 other cores, the *inclusive* L3 cache can guarantee the data is not in other cores, as all the data on the chip has a copy in the inclusive L3 cache. It means the data should load from the main memory. Intel's next generation multi-core processor Nehalem which using an inclusive L3 cache to enhance the performances.



**Figure 8 - MESIF protocol**

Intel's multi-core processor Nehalem uses an advanced way called MESIF. They adapted the standard MESI protocol to include an additional state, the Forwarding (F) state, and changed the role of the Shared (S) state. In the MESIF protocol, only a single instance of a cache line may be in the F state and that instance is the only one that may be duplicated. Other caches may hold the data, but it will be in the shared state and cannot be copied. In other words, the cache line in the F state is used to respond to any read requests, while the S state cache lines are now silent.

## **5. Conclusion:**

The trend of the processor development is multi-core. This paper brings forward the need for multiprocessors in today's computing and its advantages over single core. We saw the memory organization in multiprocessors and the cache coherence problem. We then saw the implementation of multiprocessors by today's chip manufacturers Intel and AMD and how they deal with cache coherence.

## **6. References:**

1. *Advanced Micro Devices website, [www.amd.com](http://www.amd.com), white papers.*
2. *Intel Website, [www.intel.com](http://www.intel.com), white papers.*
3. *Chip manufacturers turn to Multiprocessors, David Geer*
4. *Hennessey and Patterson, Computer Architecture, a Quantitative approach.*
5. *www.wikipedia.org*
6. *Inclusive cache of Nehalem, Intel*