Babylon University

College of Information Technology

Software Department

Shared Memory Architecture
Part One

By
Dr. Asaad Sabah Hadi
Classification Of Shared Memory Systems

• The simplest shared memory system consists of one memory module (M) that can be accessed from two processors (P1 and P2);
• Requests arrive at the memory module through its two ports. An arbitration unit within the memory module passes requests through to a memory controller.
• Requests arrive at the memory module through its two ports. An arbitration unit within the memory module passes requests through to a memory controller.
• If the memory module is not busy and a single request arrives, then the arbitration unit passes that request to the memory controller and the request is satisfied. The module is placed in the busy state while a request is being serviced. If a new request arrives while the memory is busy servicing a previous request, the memory module sends a wait signal, through the memory controller, to the processor making the new request.
• In response, the requesting processor may hold its request on the line until the memory becomes free or it may repeat its request some time later.
• If the arbitration unit receives two requests, it selects one of them and passes it to the memory controller. Again, the denied request can be either held to be served next or it may be repeated some time later.
• Based on the interconnection network shared memory systems can be categorized into:
  ➢ Uniform Memory Access (UMA)
  ➢ Nonuniform Memory Access (NUMA)
  ➢ Cache-Only Memory Architecture (COMA)
Uniform Memory Access (UMA)

- In the UMA system a shared memory is accessible by all processors through an interconnection network in the same way a single processor accesses its memory.
- All processors have equal access time to any memory location.
- The interconnection network used in the UMA can be a single bus, multiple buses, or a crossbar switch.
- Because access to shared memory is balanced, these systems are also called SMP (symmetric multiprocessor) systems.
- Each processor has equal opportunity to read/write to memory, including equal access speed.
• A typical bus-structured SMP (Symmetric Multiprocessor), shown below, attempt to reduce contention for the bus by fetching instructions and data directly from each individual cache.

• The bus contention might be reduced to zero after the cache memories are loaded from the global memory, because it is possible for all instructions and data to be completely contained within the cache.
Nonuniform Memory Access (NUMA)

- In the NUMA system, each processor has part of the shared memory attached.
- Therefore, any processor could access any memory location directly using its real address.
- However, the access time to modules depends on the distance to the processor. This results in a nonuniform memory access time.
- A number of architectures are used to interconnect processors to memory modules in a NUMA. Among these are the tree and the hierarchical bus networks.
Cache-Only Memory Architecture (COMA)

• Similar to the NUMA, each processor has part of the shared memory in the COMA.

• However, in this case the shared memory consists of cache memory.

• A COMA system requires that data be migrated to the processor requesting it. There is no memory hierarchy and the address space is made of all the caches. There is a cache directory (D) that helps in remote cache access.
COMA Shared memory system
Bus-based Symmetric Multiprocessors

• Shared memory systems can be designed using bus-based or switch-based interconnection networks. The simplest network for shared memory systems is the bus.

• The bus/cache architecture alleviates the need for expensive multiported memories and interface circuitry as well as the need to adopt a message-passing paradigm when developing application software.

• However, the bus may get saturated if multiple processors are trying to access the shared memory (via the bus) simultaneously.

• A typical bus-based design uses caches to solve the bus contention problem.

• High speed caches connected to each processor on one side and the bus on the other side mean that local copies of instructions and data can be supplied at the highest possible rate.

• If the local processor finds all of its instructions and data in the local cache, we say the hit rate is 100%.

• The miss rate of a cache is the fraction of the references that cannot be satisfied by the cache, and so must be copied from the global memory, across the bus, into the cache, and then passed on to the local processor.
• One of the goals of the cache is to maintain a high hit rate, or low miss rate under high processor loads.
• A high hit rate means the processors are not using the bus as much.
• Hit rates are determined by a number of factors, ranging from the application programs being run to the manner in which cache hardware is implemented.
• A processor goes through a duty cycle, where it executes instructions a certain number of times per clock cycle.
• Typically, individual processors execute less than one instruction per cycle, thus reducing the number of times it needs to access memory.
• Subscalar processors execute less than one instruction per cycle, and superscalar processors execute more than one instruction per cycle.
• In any case, we want to minimize the number of times each local processor tries to use the central bus. Otherwise, processor speed will be limited by bus bandwidth.
We define the variables for hit rate, number of processors, processor speed, bus speed, and processor duty cycle rates as follows:

- \( N \) = number of processors;
- \( h \) = hit rate of each cache, assumed to be the same for all caches;
- \( 1 - h \) = miss rate of all caches;
- \( B \) = bandwidth of the bus, measured in cycles/second;
- \( I \) = processor duty cycle, assumed to be identical for all processors, in fetches/cycle; and
- \( V \) = peak processor speed, in fetches/second.

The effective bandwidth of the bus is \( (B \times I) \) fetches/second.

If each processor is running at a speed of \( V \), then misses are being generated at a rate of \( V \times (1-h) \).

For an \( N \)-processor system, misses are simultaneously being generated at a rate of \( N \times (1-h) \times V \).

This leads to saturation of the bus when \( N \) processors simultaneously try to access the bus. That is, \( N \times (1-h) \times V \leq B \times I \).

The maximum number of processors with cache memories that the bus can support is given by the relation:

\[
N \leq \frac{BI}{(1 - h)V}
\]
• **Example**: Suppose a shared memory system is constructed from processors that can execute $V = 107$ instructions/s and the processor duty cycle $I = 1$. The caches are designed to support a hit rate of 97%, and the bus supports a peak bandwidth of $B = 106$ cycles/s.

- The miss rate is $\text{miss rate} = 1 - h = (1 - 0.97) = 0.03$.
- The maximum number of processors $N$ is $N \leq \left( \frac{106}{0.03 \times 107} \right) = 3.33$, thus the system we have in mind can support three processors.
- Question: what is the value of hit rate needed to support 30-processors system?

  In this case $h = 1 - \frac{B I}{N V} = 1 - \left( \frac{106 \times 1}{30 \times 107} \right) = 1 - \frac{1}{300}$ then $h = 0.9967$. 

Dr. Asaad Sabah Hadi