UNIVERSIDAD
DE BURGOS

Lecture 1

Multiprocessor Architectures: shared memory MIMD computers

V 1.1

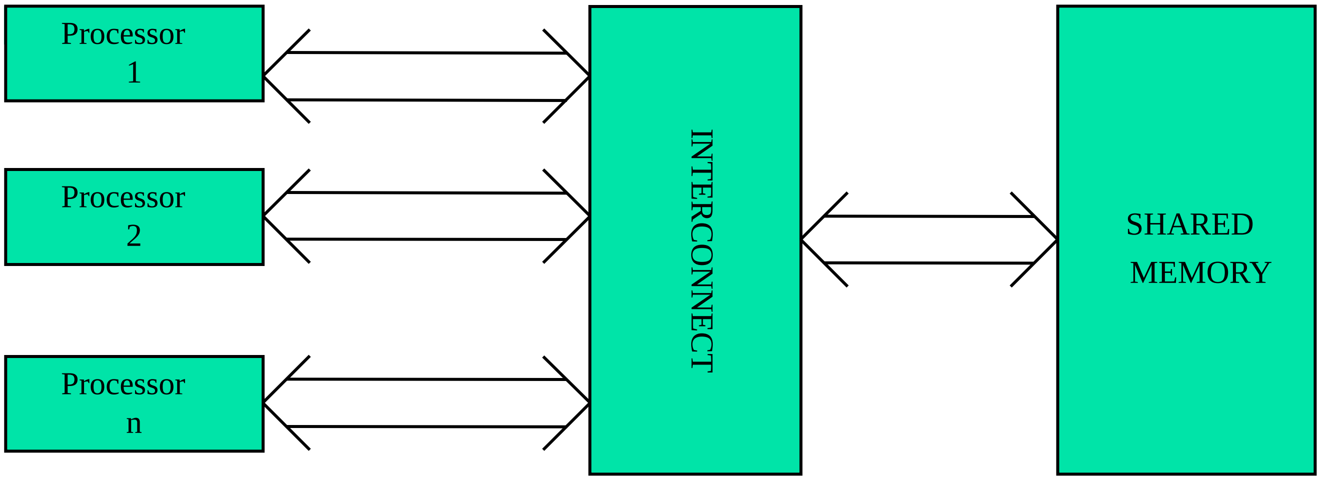
José M. Cámara (checam@ubu.es)

Multiprocessors & multicomputers

- Multiprocessors: integrated by a number of processors working in parallel. Communication is achieved by common variables in a shared memory.
- Multicomputers: each processor in the system owns a private memory unreachable by the rest. This is known as a distributed memory system. Communication is achieved by a message passing mechanism.

Shared memory

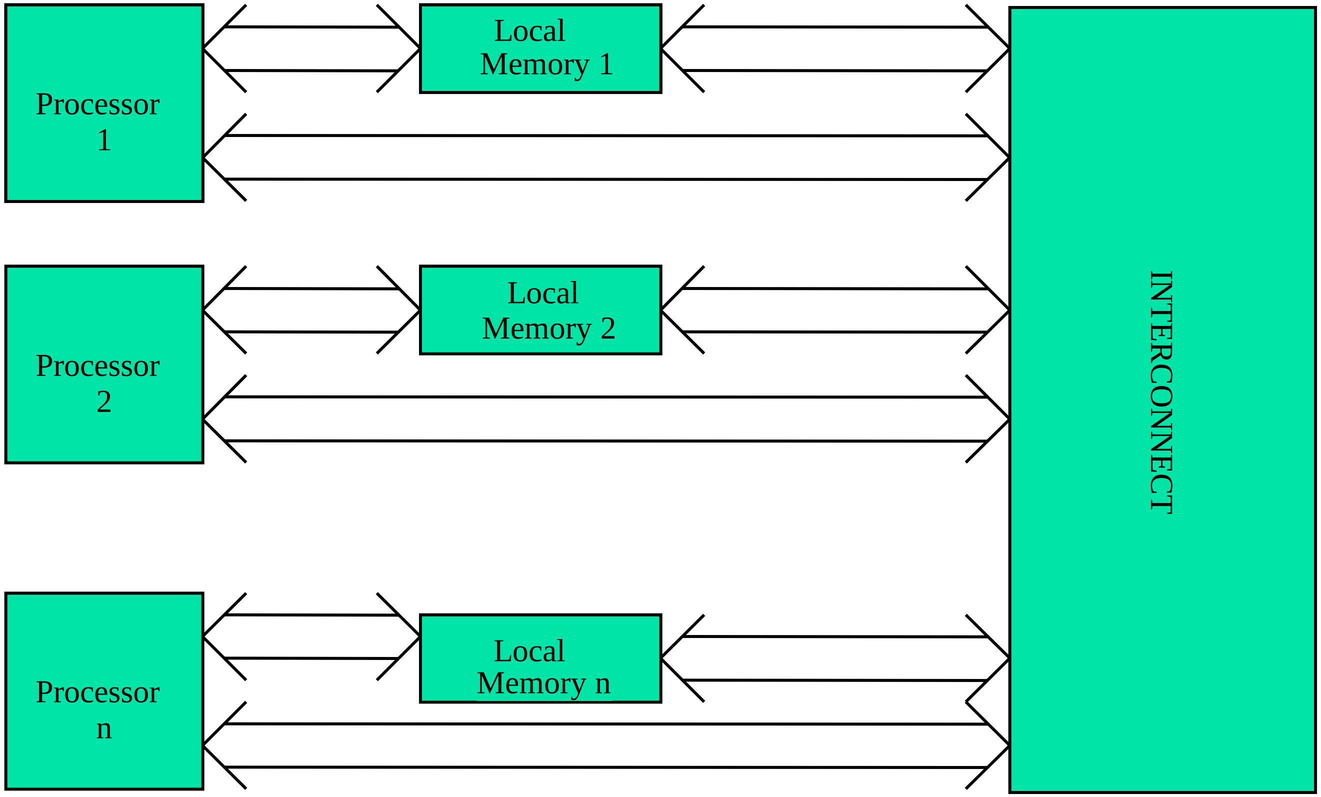
- UMA



UNIVERSIDAD
DE BURGOS

Shared memory

- NUMA



Shared memory

- Coherence conflicts due to:
  - Data sharing
  - Process migration
  - Input - output
- To accomplish:
  - Write propagation
  - Write serialization

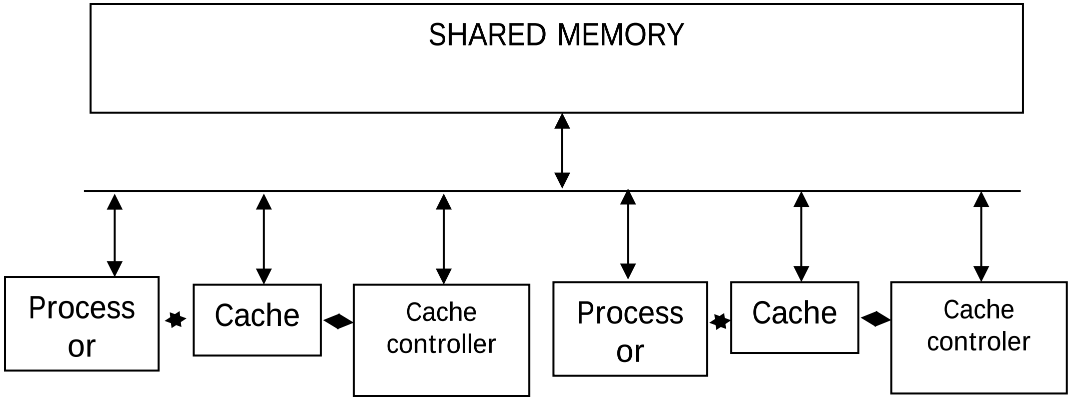
Cache Coherence

- Bus snoopy protocols:
  - Write invalidate
  - Write update
- Source snoopy protocols:
- Directory based protocols:
  - Full mapping
  - Limited
  - Chained

UNIVERSIDAD
DE BURGOS

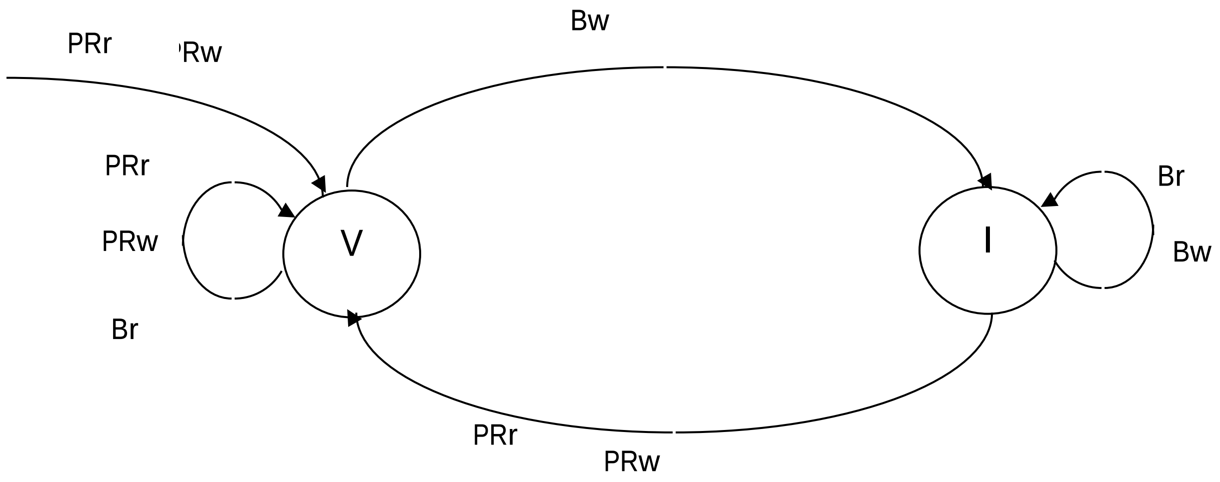
Snoopy protocols:

- Invalidate:
  - Write-trough
  - Write-back
    - MSI
    - MESI (Intel)
    - MOESI (AMD)*
    - MESIF (Intel)*
- Update:
  - Firefly
  - Dragon

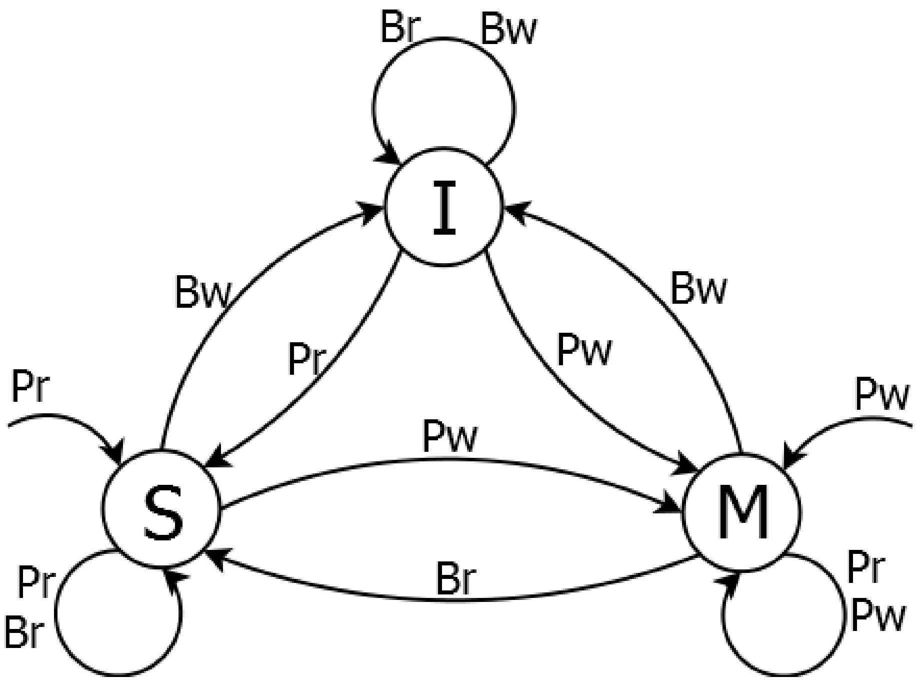


* Are not associated to a bus but rather to point to point connections

Write through

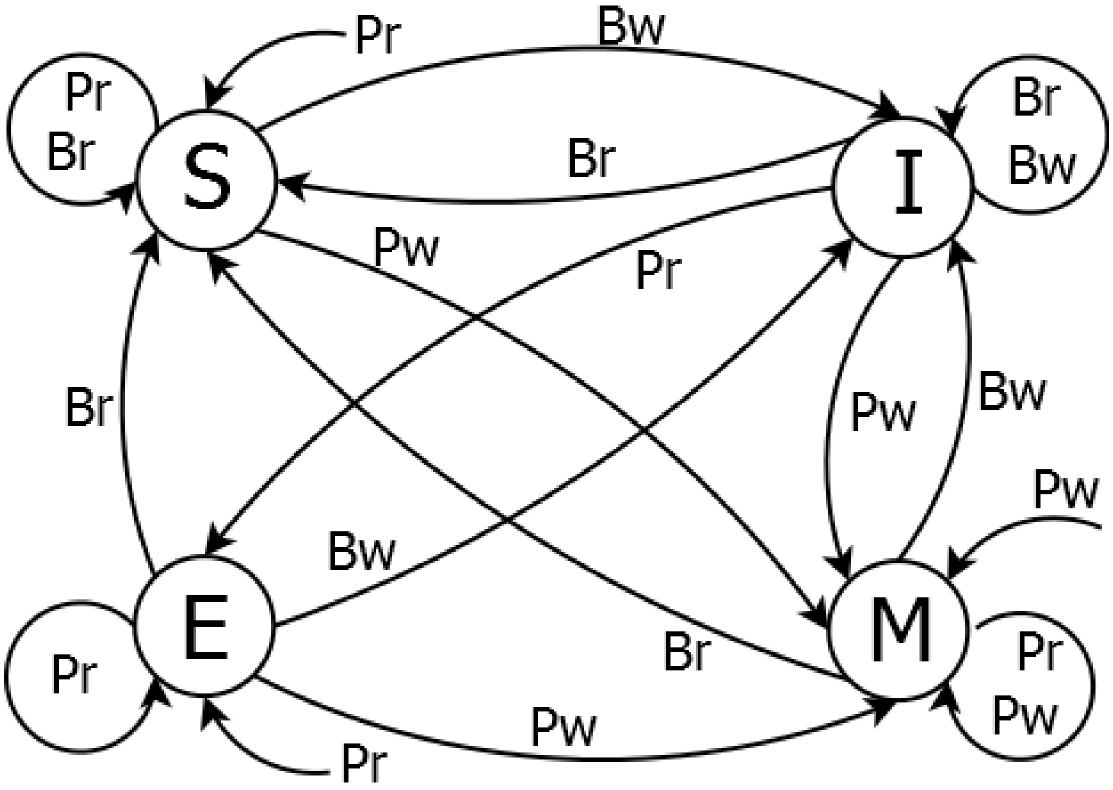


MSI protocol



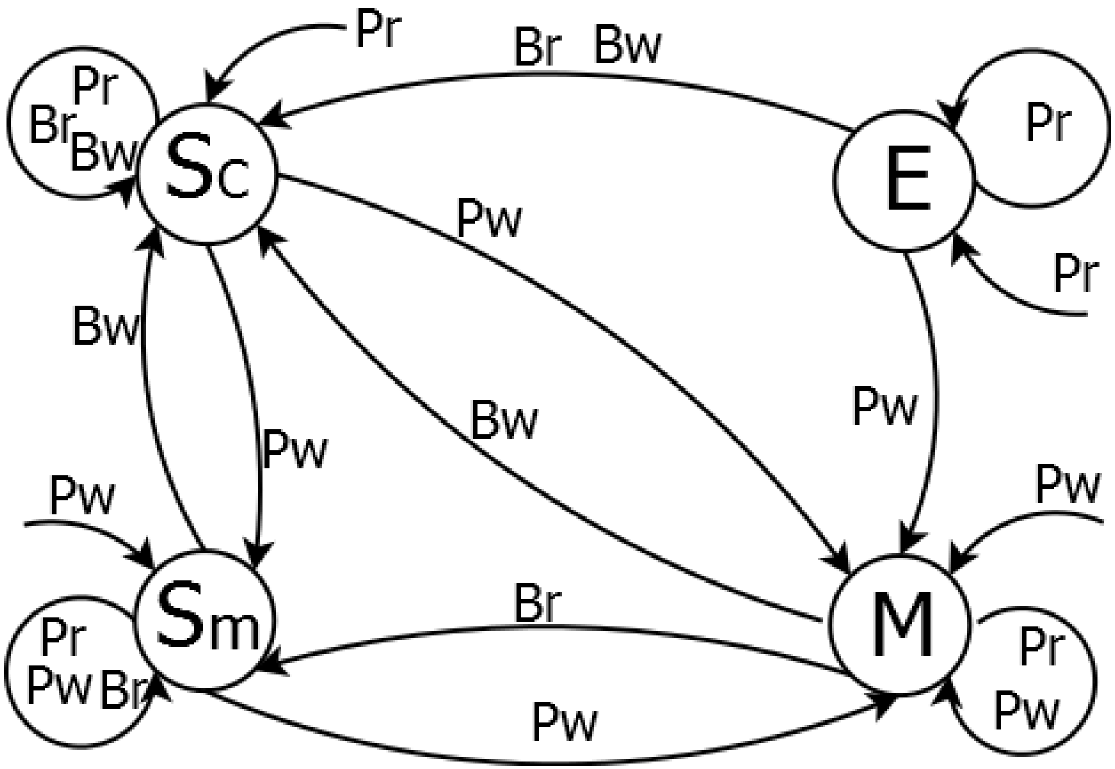
Multimedia content available

MESI protocol



Multimedia content available

Dragon protocol



Multimedia content available

Directories

- Stored in main memory.
- Provide the memory controller information about all copies of cache lines present in local caches.
- Each cache line has an attached tag.
- There are 3 types of directories:
  - Full map directories
  - Limited directories
  - Chained directories

Full map directories

- Tag includes a single bit field for each possible owner of copy (normally all processors).
- An additional single field bit indicates if write rights have been granted to any processor. In that case, only the requester's bit on the tag will be active.
- Before granting write access, the memory controlled hass to invalidate all other copies.
- Full map directories do not scale well:
    - Tags' sizes grow proportionally to the number of nodes > too much space in main memory to store the directory.

Full map directory example

- 512 node computer; 2 Gbytes main memory on each node.
- 64 bytes cache lines.
  - 1024GB/64bytes = $2^{40}/2^{6}=2^{34}$ cache lines= tags
  - Each tag 512 + 1 bits ≈ 2^{9} bits = 2^{6} bytes
  - 2^{34} tags * 2^{6} bytes/tag = 2^{40} directory bytes

Limited directories

- Tag includes only some fields to keep track of copies on a limited number of nodes.
- Although less in number, fields have to be big enough to point to any node on the network: $\log_2$ (number of nodes).
- The write permission field does exist as well.
- Total tag's size is reduced if the number of fields is severely restricted.
- In compensation, the number of copies in local caches is accordingly reduced. If all fields are being occupied and a new processor request a copy of the cache line, another one has to be removed. This leads to unnecessary swaps and a swapping algorithm has to be implemented.

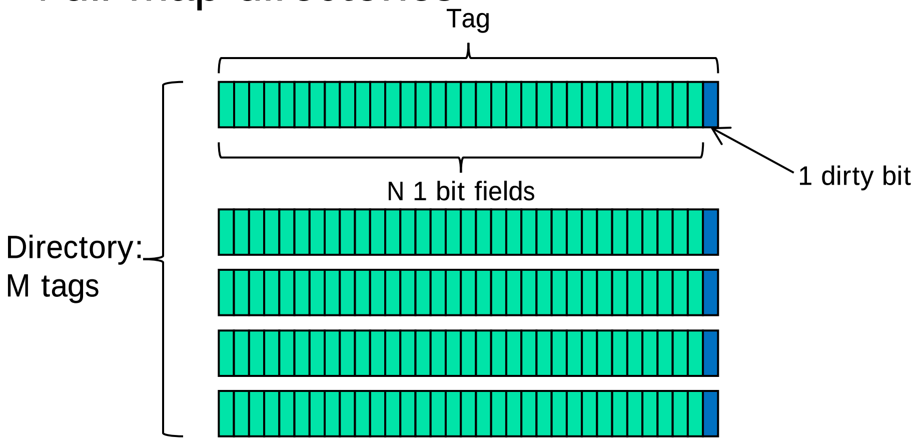
Chained directories

- Tags in main memory point only to the last copy owner to join the list.
- Each one on the list has a pointer to the previous one.
- When a new node joins the list, it is placed at its end and given a pointer to its predecessor.
- It a write request is issued, it has to be propagated to the beginning so all nodes are aware and invalidate their copies. Eventually, the memory controller grants write access.
- This is a space saving but slow procedure. Directory space in main memory is optimized but some of as well as management capabilities have to be assumed by the nodes.

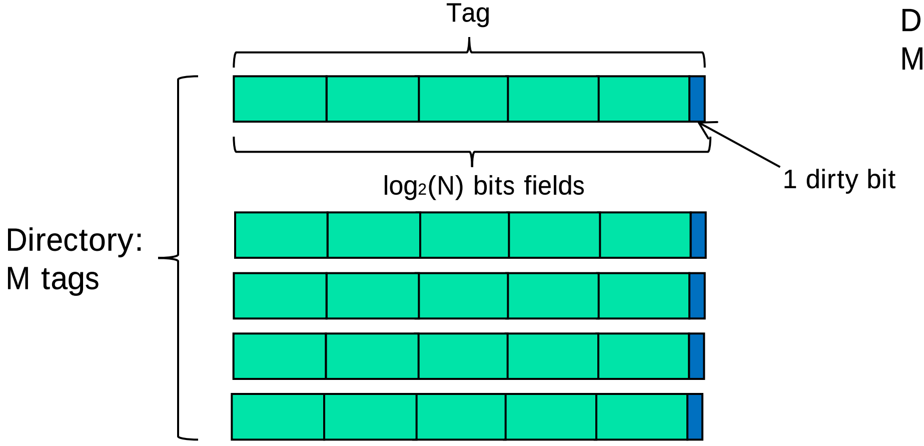
Directories

- N processors in the system
- M cache lines in main memory

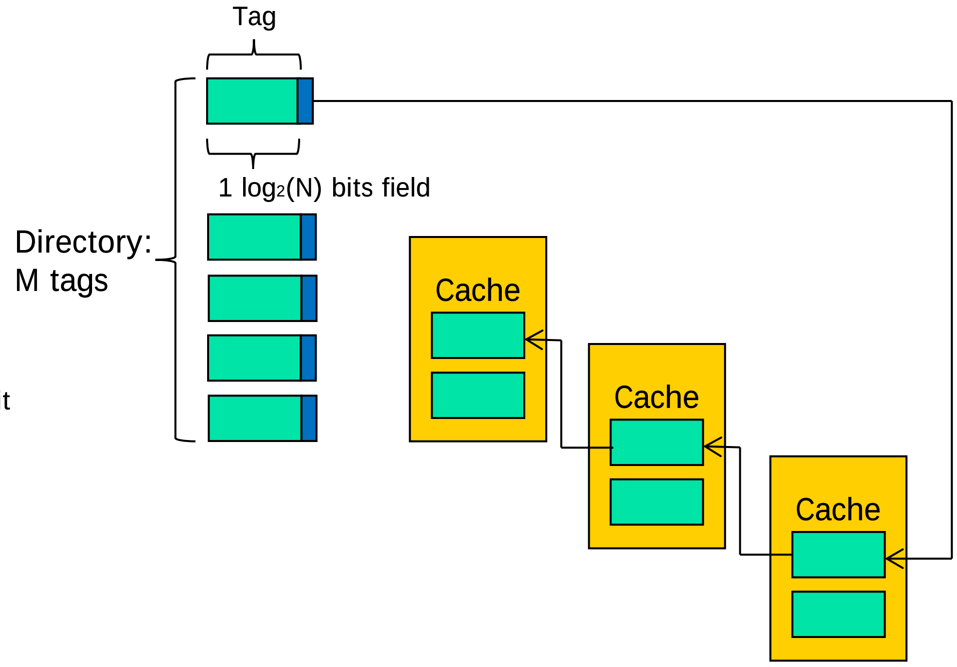
Full-map directories



Limited directories



Chained directories



Actual bus example: TLSB

- Introduced in Alpha computers such as Alphaserver 8400.
- Second half of the 90s.
- Synchronous bus with separated address and data buses.
- 256 bits data bus.
- BW up to 3,2 Gbytes/s: 32bytes / 100MHz.

TLSB: Addressing I

- Three address lines for geographical addressing of the modules. Up to 9 modules can be connected since 000 address is shared by device 0 and the required I/O module.
- Virtual addressing scheme for devices such as memory banks and CPUs whom, in this way are given an address within the system. Each module can hold up to 8 virtual addresses.

TLSB: Addressing II

- Up to 1TB main memory can be addresses via a 40 bits address scheme.
- The address is decoded by the requester to extract the bank’s virtual address.
- Up to 16 memory banks can be supported.
- Cache line size is 64 bytes.
- The requester may launch a bus request and check local cache alongside. If a cache hit happens, the request is invalidated.

TLSB: Addressing III

- Bits 6 to 39 are used as cache line address.
- Bit 5 is used to determine the order in which the two 32 bytes parts of the line are delivered: High>low or Low>high.
- The 5 bits remaining are used to code the virtual address (up to 16 CPUs & up to 16 memory banks).

TLSB: Arbitrage

- Any transaction on the bus must be initiated as a request from the module will to communicate.
- TLSB implements a distributed arbitrage mechanism. That means that there is no arbiter. Conflicts have to be solve by the competing modules with no external intervention.
- Priority is set according to the geographical address at star up.
- At run time a Round Robin mechanism is used to guarantee fairness.

TLSB: Transfers

- When a slave node is ready to deliver the requested data, it takes over the bus.
- The slave activates TLSB_SEND_DATA line, forcing the nodes storing copies of the same cache line to set TLSB_SHARED or TLSB_DIRTY lines.
- TLSB_SHARED set means that there is another node possessing a valid copy and wants to keep it.
- TLSB_DIRTY set means that a more recent copy of the line exists and, in this case, the node that set the line will complete the transaction.
- Bus specifications do not reference any coherence protocol in particular. They just set up the basis to make it possible.

References

- [1] K. Whang. Advances Computer Architectures. McGraw Hill.
- [2] Alphaserver8400 system handbook.