Supercomputers of today consist of hundreds or thousands of processors connected to hundreds of gigabytes of memory.

Today we have no single, universal supercomputer architecture capable of satisfying all users Indeed, we have some doubts whether such architecture could exist.

We still aim at satisfying users, who demand:

- **High performance**
- **Low cost**
- **Sustained productivity**

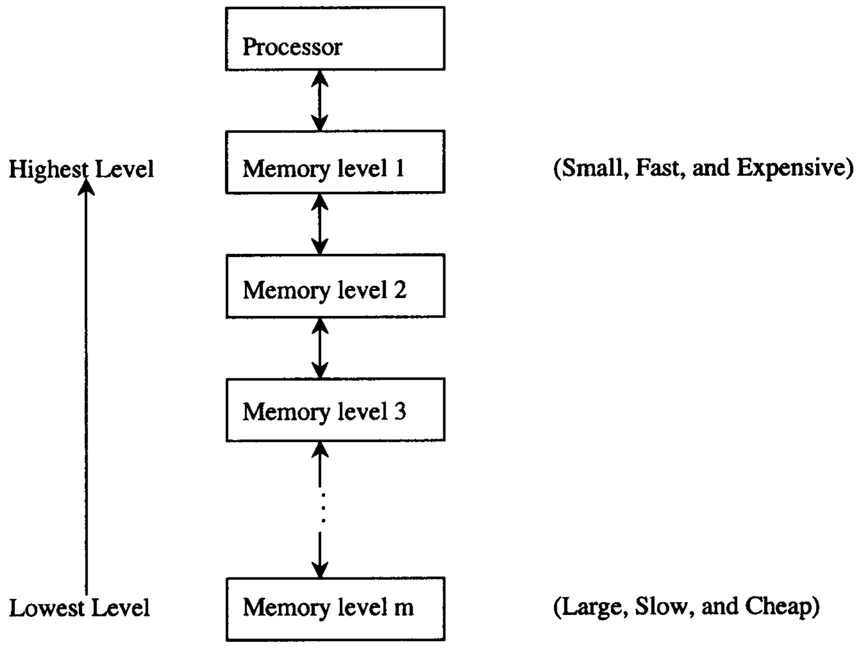
Regarding Low cost: How about allowing users to build their own supercomputers from widely available components, much like we build today home theatre systems?

Issues of concern in designing parallel architectures:

- **Large register set** – decreases delays caused by interrupts
- **Large physical address space** – decreases delays due to swapping
- **Processor scheduling** – allows dynamic allocation of processors to tasks
- **Processor synchronization** (of concern to SIMD only) – prevents processors from simultaneous modification of shared memory locations. Efficient primitives needed to handle synchronization and interrupts
- **Interconnection network topology** – allows efficient processor-to-processor and processor-to-memory connections (the most expensive component of any supercomputer!)
- **Partitioning** – identifies parallelism in algorithms to specify the parallel computing streams. Various levels of partitioning are possible – at a statement, procedure, or program levels
- **Reliability** – refers to graceful degradation of performance and better fault tolerance. As some processors fail, the workload should be picked up by “healthy” processors
- **High performance** – measured in peak performance or sustained performance

Memory system design is one of the most important design issues.

Memories are hierarchical – this exploits the locality of references:



- All data transfers between the neighbouring memory levels are controlled by activity in memory level 1
- Each level is considered a subset of the next level
- Main memory is the largest random access memory in the system

Two kinds of RAM technologies exist:

- **Static RAM** (SRAM) – of non-destructive read property
- **Dynamic RAM** (DRAM) - of destructive read property. Each memory cell that was read must be refreshed. The cost of refreshing circuits is considered an acceptable and unavoidable evil

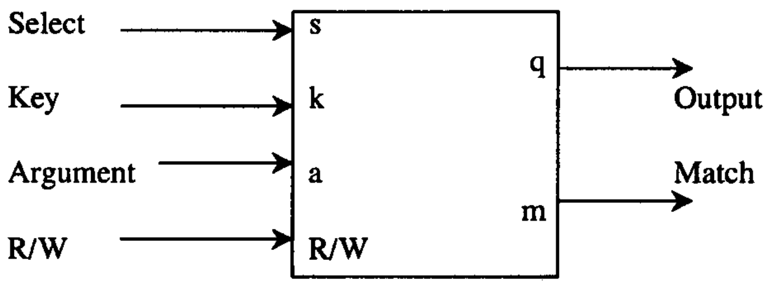
Associative Memory (AM) a.k.a. **Content Addressable Memory** (CAM)

AM consists of memory locations and associative logic (data gathering logic)

AM is used to locate data by parallel searching of cells for specific bit patterns, so this is **search by contents** rather than **search by address!**

Each AM memory location contains sufficient logic circuits to determine if it does or does not hold data that match some search pattern, broadcast from a control unit. Search pattern consists of an argument and a mask.
Argument is to contain value to be searched
Mask specifies which argument bits to use when searching

Each AM memory location contains a number of AM cells of the following structure:

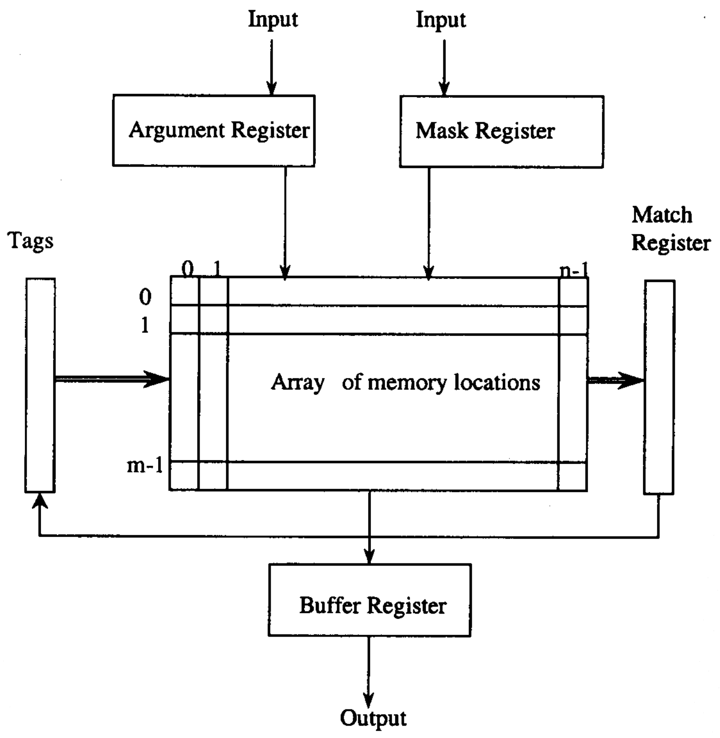


Each AM cell holds one bit, and has four (one bit) inputs and two (one bit) outputs.

AM types:

- Exact match AMs (or CAMs)
- Comparison AMs (or CAMs)

AM memory locations are organized in banks of m words and n bits in each word, viz.:



There are mxn searchable memory locations.

AMs are much faster than traditional memories, but also much more expensive.

MEMORY MODELS FOR PRAMs

PRAM = Parallel Random Access Machine consists of M memory locations (RAM) shared by P processors

Note the dual role of RAM:

- To store data
- To act as communications area between processors

RAM access primitives:

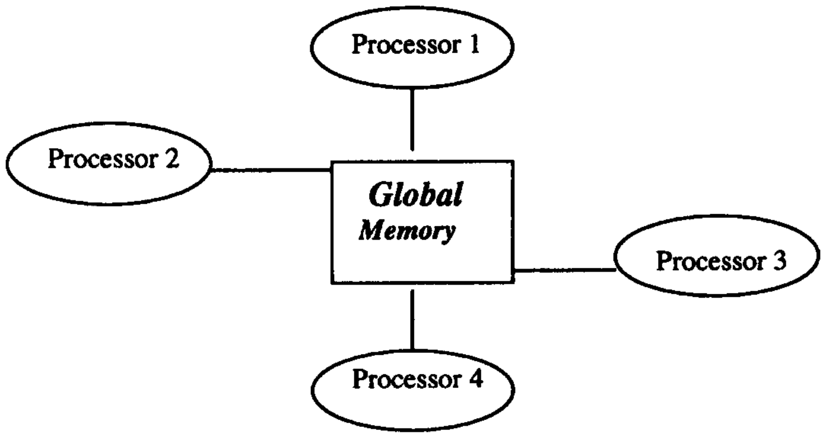
- **Concurrent read** (CR) – allows two or more processors to read the same RAM location simultaneously
- **Exclusive read** (ER) – allows p processors to read simultaneously p distinct RAM locatons
- **Concurrent write** (CW) - allows two or more processors to write to the same RAM location simultaneously, subject to some protocol (see below)
- **Exclusive write** (EW) - allows p processors to write simultaneously to p distinct RAM locatons

Possible CW protocols:

- **Priority CW** – processors are assigned priorities (statically or dynamically) and the processor of the highest priority is allowed to write to the RAM location
- **Common CW** – all processors are allowed to write, provided they intend to write the same value (one processor is doing the actual writing)
- **Arbitrary CW** – several processors attempt to write, but one is allowed to write. Which one? – This info must be provided somehow
- **Random CW** – succeeding processor is chosen at random
- **Combining CW** – all values intended to be written by the processors are somehow combined into a single value that is actually written

MEMORY MODELS FOR PRAMs

Multiport Memory:



This is an attempt to overcome bus limitations. Large-capacity multiport memories are not available due to packing costs and pin-bandwidth limitations. However, complex interconnection networks may be built using multiport memories.

MEMORY MODELS FOR PRAMs (continued)

Multiprocessor Memory:
Processors communicate via shared variables in common memory.

The categories of shared memories:

- **Uniform Memory Access model** (UMA) – shared memory is centralized. Some central switching mechanism is needed to access memory (bus, or crossbar switch, or or packer-switching network, etc.). When all processors have access to all peripherals, we call it the symmetric multiprocessing system (SMP)

- **Non-Uniform Memory Access model** (NUMA) – shared memory is distributed and partitioned into a number of modules. Local modules are directly accessible by their processors. Collection of all modules form a global memory accessible to all processors. Processors are allowed simultaneous access to one or more memory modules, and operate independently of each other

- **Cache-Only Memory Architecture** (COMA) – distributed memories are treated as caches – all caches form a global address space

MEMORY MODELS FOR PRAMs (continued)

Multicomputer Memory:

- Each processing node of a multicomputer system is a self-contained computer with unshared, local memory. It communicates with other processing nodes by message passing. Advantages:
- Interconnecting network can be simpler than in shared-memory systems, because …
- Messages are sent less frequently, resulting in lower traffic
- Memory management systems are simpler, similar to single-CPU machines