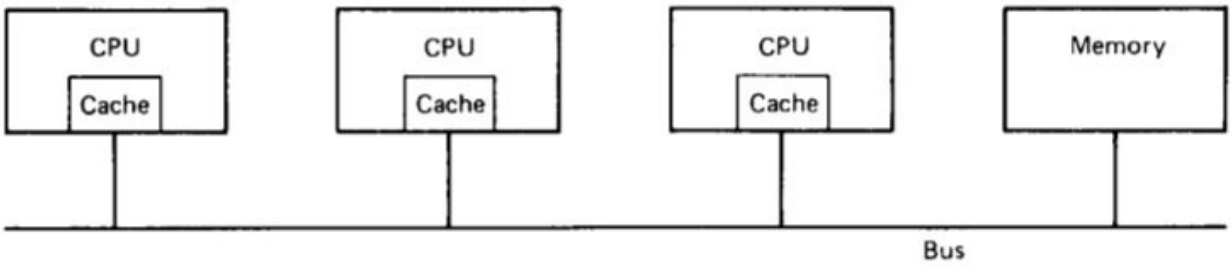
BUS BASED AND SWITCHED MICROPROCESSOR

Bus based multiprocessor

Bus based multiprocessor consists of some number of CPUs all connected to a common bus, along with a memory module.



A simple configuration is to have a high speed backplane or motherboard into which CPU or memory cards can be inserted. A typical bus has 32 or 64 address lines, 32 or 64 data lines, and perhaps 32 or more control lines, all of which operate in parallel.

The problem with this scheme is that with as few as 4 or 5 CPUs , the bus will usually be overloaded and performance will drop drastically. The solution is to add the high speed cache memory between the CPUs and the bus as shown in the fig. The cache holds the most recently accessed words. All memory requests goes though the cache. If the word requested is in the cache, the cache itself responds to the CPU and no bus request is made. If the cache is large enough, the probability of success, called the hit rate, will be high and the amount of bus traffic per CPU will drop dramatically, allowing many more CPUs in the system.
However the introduction of cache also bring the serious problem with it suppose that two CPUs A and B each read the same word into their respective caches. Then A overwrites the word. When B next reads that word, it gets the old value from its cache, not the value A just wrote. The memory is now incoherent and the system is difficult to program.
One of the solution for it is to implement write-through cache and snoopy cache. In write-through cache, cache memories are designed so that whenever a word is written to the cache, it is written through to memory as well. In addition, all caches constantly monitor the bus. Whenever a cache sees a write occurring to a memory address present in its cache, it either removes that entry from its cache or updates the cache entry with the new value. Such a cache is called **snoopy cache.** because it is always snooping on the bus.

Using it, it is possible to put about 32 or possibly 64 CPUs on a single bus.

Switched Multiprocessor

To build a multiprocessor with more than 64 processors, a different method is needed to connect the CPUs with the memory. Two switching techniques are employed for it.
a. Crossbar Switch
b. Omega Switch

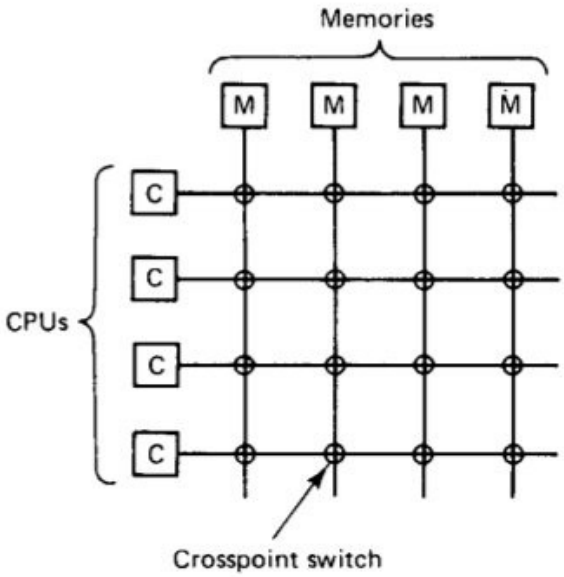


Fig a). Crossbar Switch

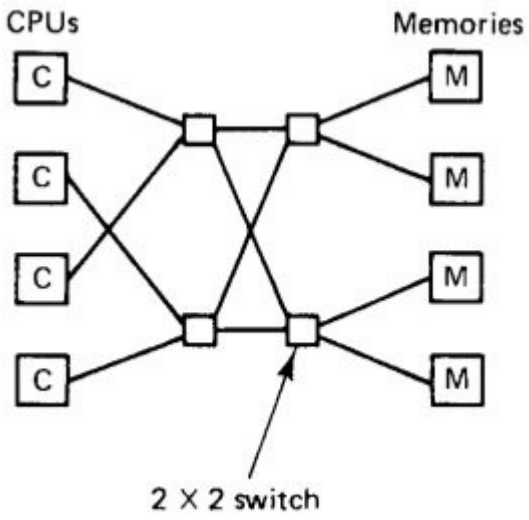


Fig b). An Omega Switching Network

Crossbar Switch:

Memory is divided into the modules and are connected to the CPUs with the crossbar switch. Each CPU and each memory has a connection coming out of it, as shown. At every intersection is a tiny electronic crosspoint switch that can be opened and closed in hardware. When a CPU wants to access a particular memory, the crosspoint switch connecting them is closed, to allow the access to take place. If two CPUs try to access the same memory simultaneously, one of them will have to wait.
The downside of the crossbar switch is that with n CPUs and n memories, n^2 crosspoint switches are needed. For large n this number can be prohibitive.

Omega Switching network

It contains 2x2 switches, each having two inputs and two outputs. Each switch can route either input to either output. A careful look at the figure will show that with proper setting of the switches, every CPU can access every memory. In general case, with n CPUs and n memories, the omega network requires $\log_2 n$ switching stages, each containing n/2 switches, for a total of $(n\log_2 n)/2$ switches.