CSCI 344/715
Lecturer: Dr. Simina Fluture

Topics: Hardware Implementation of Mutual Exclusion
TTS

Readings:	**Web Lecture**	Bibliography:	[AT] ch. 1
	In-class notes		[JH]

UMA – Uniform Memory Access

- **Bus-Based Multiprocessors**

A number of CPUs all connected to a common bus and a common memory module.

Fig. A bus-based multiprocessor

Read a word: CPU puts the address of the word on the bus address lines, and a signal on the appropriate control lines that will indicate the type of the operation. The memory responds by putting the value of the word on the data lines.

Write a word:

With even only 5 CPUs, the bus becomes overloaded and the performance decreases.

- **Multiprocessors Using Crossbar Switches**

The simplest circuit for connecting *n* CPUs to *k* memories is the **crossbar switch.**

+) if the memory module is available, no CPU is ever denied the connection it needs – **nonblocking network.**

-) The number of crosspoints grows as n^2.

- **Multiprocessors Using Multistage Switching Networks**

Is based on a 2 x 2 switches – **omega network**.

CSCI 344/715
Lecturer: Dr. Simina Fluture

Cache, Cache coherence

Cache Hit:

Cache Miss:

If a multiprocessor system has an independent cache in each processor, the possibility exists for two or more caches to contain different versions of the same information at the same time. This is called the ***cache coherence*** or ***multi-cache consistency*** problem.

CPU1 – K1		CPU2 – K2
A = A + 1;	**A = 2**	**A = A – 1;**

Solution:
Write-through cache policy:

Note: in order to work, the processor is forced to bypass the cache and access the main memory, always acquiring the most recent version of the shared cache variable.

Snooping Cache:

TS (test and set) vs. TTS (test and test and set)