

SIMD Architectures

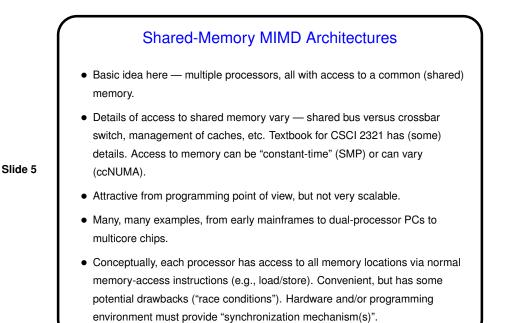
 Basic idea sort of implied by name (Single Instruction, Multiple Data) — many identical arithmetic units all executing the same instruction stream in lockstep (via single control unit), each on its own data. Can have separate memory for each AU or all can share.

Slide 3

- Vector processor addition(s) to CPU meant to speed up operations on arrays (vectors) by using pipelining and/or multiple AUs. Can be thought of as a special case of (pipelined) SIMD.
- Both used more widely in early supercomputers than now, except in special-purpose hardware.

MIMD Architectures

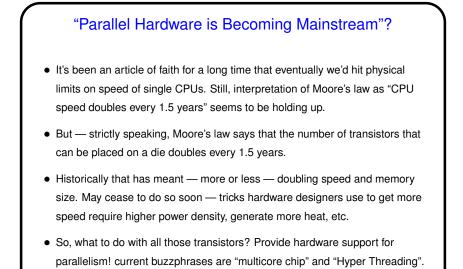
- Again, basic idea implied by name (Multiple Instruction, Multiple Data) many processors, each executing its own stream of instructions on its own data.
- Category is broad enough, and popular enough, to consider two subcategories (shared and distributed memory).



Distributed-Memory MIMD Architectures

- Basic idea here multiple processors, each with its own memory, communicating via some sort of interconnect network.
- Details of interconnect network vary can be custom-built "backplane" or standard network. Various "topologies" possible. Textbook for CSCI 2321 has (some) details.

- Not initially as attractive from a programming point of view, but very scalable.
- Examples include "massively parallel" supercomputers, Beowulf clusters, networks of PCs/workstations, etc.
- Conceptually, each processor has access only to its own memory via normal memory-access instructions (e.g., load/store). Communication between processors is via "message-passing" (details depending on type of interconnect network). Not so convenient, but much less potential for race conditions.



One Approach — Multicore Chips Key idea here — chip includes several (usually two or four) "cores", all sharing one connection to memory. Each "core" is a CPU in the sense we talk about in Computer Design; each typically has its own first-level cache. To fully exploit this for a single application, probably need multiple threads.

Slide 7



Slide 9

"Parallel Hardware is Becoming Mainstream"?, Continued

 In addition to hardware support for shared-memory parallelism — Ubiquity of networking makes almost any PC part of a "cluster".

