ELEC6234
Embedded Processor Design

Inter processor communication
Moore’s Gap due to many core

- Diminishing returns from single CPU mechanisms (pipelining, caching, etc.)
- Wire delays
- Power envelopes

Source: Jason Miller, Carbon Research Group, MIT
Embedded multicore trends

**Yesterday**

- A few large cores on each chip
- Shared global structures: bus, L2 caches
- Only option for future scaling is to add more cores

**Today & tomorrow**

- 100’s to 1000’s of simpler cores
  - [S. Borkar, Intel, 2007]
- Simple cores are more power and area efficient

**Major communication challenges**
Typical inter-processor communication systems
Bus-based communication

Bus

Shared memory

CPU

CPU

M

Cache

CPU

CPU

M

Private memory

Shared memory

CPU

CPU

M
Crossbar-based communication

- **Diagram Description**: The diagram illustrates a crossbar communication system with CPUs and Memories. Each CPU (000, 001, 010, 011, 100, 101, 110, 111) is connected to Memories through crossbar switches. The figure shows different states of the crossbar switches:
  - **Closed crosspoint switch**: Indicates a closed switch allowing communication between CPUs and Memories.
  - **Open crosspoint switch**: Indicates an open switch preventing communication.

- **Figure Notes**:
  - (b) Crosspoint switch is open
  - (c) Crosspoint switch is closed
In this configuration there is exactly one path from each source to any particular destination.
OS distribution in traditional multi-processor systems

1. Each processor has its own OS
2. Only master runs an OS
Modern approach in many-core embedded systems

A few large cores run OS.

Many small cores run tasks

Small cores are heterogeneous, i.e. application specific
Tightly-coupled CPUs that do not share memory

- Interconnect topologies
  - (a) single switch
  - (b) ring
  - (c) grid
  - (d) double torus
  - (e) cube
  - (f) hypercube
Multi-port memory systems

Source: J LaRue and D Tseng “Using dual port interconnect to resolve multiprocessor system bottlenecks, EETimes, 18 Jan 2006.
Dual-port memory depth expansion

Source: J LaRue and D Tseng “Using dual port interconnect to resolve multiprocessor system bottlenecks, EETimes, 18 Jan 2006.
Cache coherence problem

• The primary advantage of caches is their ability to reduce the average access time in processors.
• When the processor finds a word in cache during a read operation, the main memory is not involved in the transfer.
• If the operation is to write, there are two commonly used procedures to update memory.
  • In the write-through policy, both cache and main memory are updated with every write operation.
  • In the write-back policy, only the cache is updated and the location is marked so that it can be copied later into main memory.

However, in multi-core cached systems with a shared memory, cache coherence needs to be maintained.
Cache coherence problem

• In a shared memory multiprocessor system, all the processors share a common memory. In addition, each processor may have a local memory, part or all of which may be a cache.

• The same information may reside in a number of copies in some caches and main memory. To ensure the ability of the system to execute memory operations correctly, the multiple copies must be kept identical.

• This requirement imposes a cache coherence problem.

• A memory scheme is coherent if the value returned on a load instruction is always the value given by the latest store instruction with the same address.
Cache coherence problem

Shared memory contents after processor P1 writes X. P2 and P3 caches must be updated to maintain coherence.
Summary

- Many forms of multiple processor communication systems
- Architecture must be carefully designed to fully utilize the multiple resources
- Programming is a major challenge
- Communication, memory, energy consumption are important factors influencing the system cost and performance
- Embedded On-chip Multi-Processor (MPSoc) technology appears to be growing