Sistemi embedded a multiprocessore

Massimo Bocchi

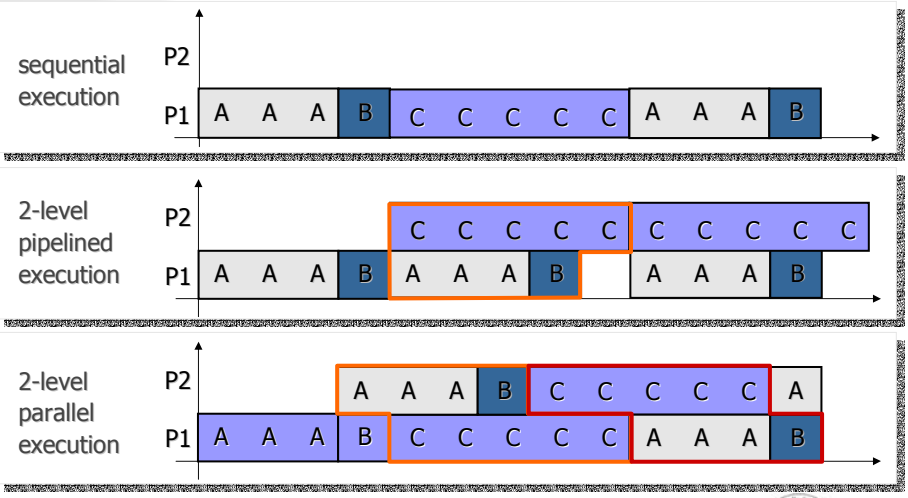
Architetture Digitali per l'Elaborazione dei Segnali LS

A.A. 2003/2004

Motivation

- Performance increase
  - Pipelined execution
  - Parallel execution
- Cost effectiveness
  - Custom uniprocessors are less cost effective
  - Microprocessor design cost can be amortized if the processor core can be reused on the same system
- Multi processor System On Chip (MPSoC)
  - Growing transistor count on a single chip should be exploited
  - Parallelism becomes effective use of chip density

Processor-time diagram



Main topics

- Inter-processor communication and data sharing
- Inter-processor synchronization
- Interconnection architecture

Multiprocessor classification

In 1966 Flynn proposed a multiprocessor classification:

- SISD (single instruction stream – single data stream): a normal single processor based on the von Neumann model
- SIMD (single instruction stream – multiple data stream): a single instruction stream is generated by a single control unit but used by several processors; each processor executes the same instruction stream but acts upon different data, generating multiple data streams
- MISD (multiple instruction stream – single data stream): no existing implementations can be referred to this model
- MIMD (multiple instruction stream – multiple data stream): one instruction stream is generated for each computer; each instruction acts upon different data

MIMD multiprocessor classification

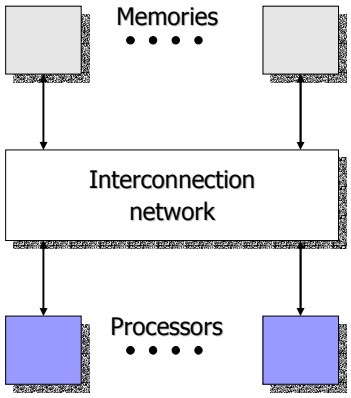
A MIMD multiprocessor system needs a mechanism to load instruction and data memories and to pass information between the processors.

Two main models:

- shared memory multiprocessor systems
  - Uniform Memory Access (UMA) multiprocessors or Symmetric Multiprocessors (SMP)
  - Non-Uniform Memory Access (NUMA) multiprocessors or Asymmetric Multiprocessors (AMP)
- multiprocessor systems without shared memory
  - dataflow multiprocessor systems
  - message-passing multiprocessor systems

Shared memory model

- All the processors have access to a common memory space
- Cache memories create consistency problems on the shared data

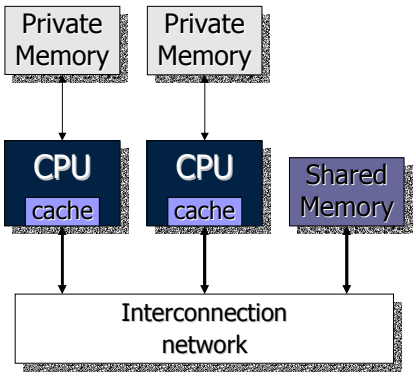


Shared memory multiprocessors

- Implicit communication performed through shared variables
- Explicit synchronization based on lock/unlock mechanism (semaphores and monitors)
- Two types of shared memory systems:
  - Uniform Memory Access (UMA)
  - Non-Uniform Memory Access (NUMA)

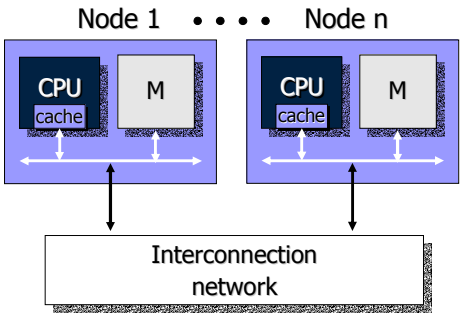
UMA Multiprocessor Systems

- All the memory resources feature the same access latency
- CPUs can access to RAM via bus or interconnection network
- Cache memories and local (private) memories can be used to reduce bus contention or network traffic.
- Straightforward programmability



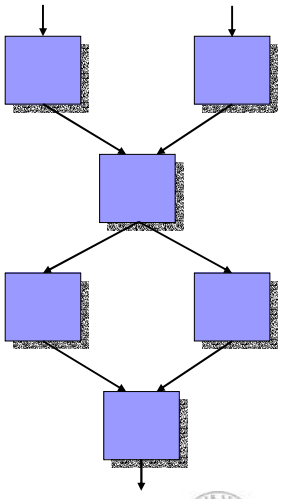
NUMA Multiprocessor Systems

- Remote memory resources have higher access latency then local memories
- The system features a single address space visible to all CPUs
- Potentially higher performance due to higher scalability
- Difficult to program



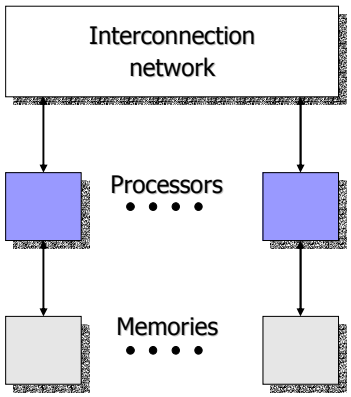
Dataflow model

- This model directly derives from DFG representations
- An instruction is executed within a module when the operands required are available
- The computation is *data driven* since the data stream activates the nodal operations



Message-passing model

- Direct links are used to pass data between the processors
- There are only local memories; each memory can be accessed only by one processor



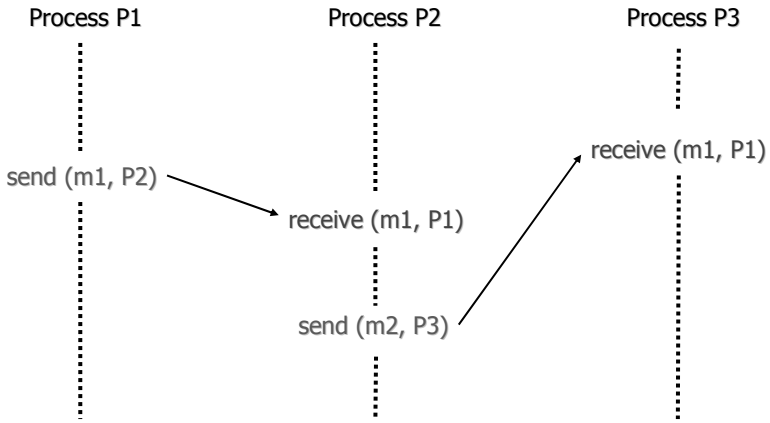
Message-passing systems elements

- Node: an independent subsystem containing a processor and local memories and peripherals
- Link: a physical communication path between two nodes
- Channel: a communication path between two processes in one processor or between two processes executing on different processors

Message Passing Systems

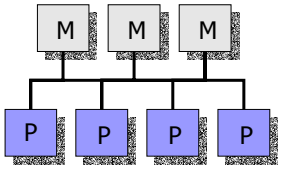
- Explicit communication based on *send* and *receive* primitives
- Implicit synchronization through blocking methods
- Since each node runs as an independent cluster, these system are also referred as *multicomputers*.

Message communication mechanisms



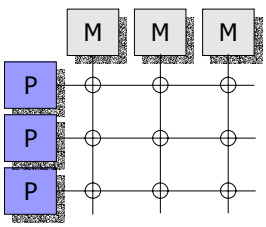
Interconnection architectures

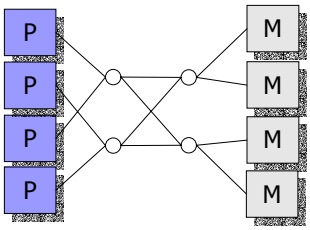
- Bus-based interconnection



Interconnection architectures (2)

- Switch-based interconnection (network)
  - Crossbar switch
  - Omega switching network

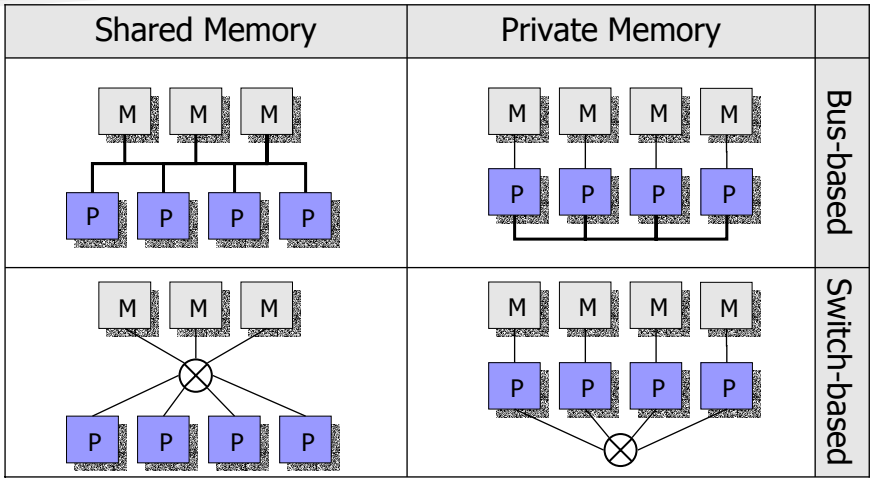




Bus-based multiprocessor systems

- A typical implementation of a shared memory model consists of a bus-based multiprocessor system (e.g. an AMBA AHB-based system)
- These systems are not easily expandable to contain a large number of bus masters
- Synchronization mechanisms are necessary to maintain consistency on the shared data
- The shared bus and memory generate contentions between bus masters, thus reducing the system speed and the data throughput

Parallel systems architectures



How many processors?

Category	Type		Number of processors
Communication model	Message passing		8-256
	Shared memory	NUMA	8-256
		UMA	2-64
Physical connection	Switch-based		8-256
	Bus-based		2-32

MPI standard

- Message Passing Interface is a standard for writing message-passing programs
- Supports an extended message-passing model
- It is not targeted to a specific platform
- Both Fortran 77 and C languages are supported
- Main MPI features are intended to improve performance on scalable parallel computers with specialized inter-processor communication hardware
- MPICH is a free portable implementation of MPI standard

MPI Programming

- Single Program Multiple Data (SPMD) model
- Communication:
  - Point-to-point
  - Collective (broadcast primitives)
- Synchronization
  - Point-to-point synchronization is performed by message passing and blocking primitives
  - Global synchronization is done by broadcast communication paradigm

SPMD Programming Model

- The same program is loaded by all the processors
- Each processor can identify itself
- Total number of processors is known
- Execution can be different on each CPU:

```
My_ID = getID();
if (my_ID == 1)
        execute task1;
if (my_ID == 2)
        execute task2;
…
if (my_ID == N)
        execute taskN;
```