

Tujuan: Memahami konsep pipeline pada computer sybsystem serta konsep perancangannya

Pokok bahasan: Arsitektur Sinkron

Reference: Buku 1 (Chapter 10.3)

Tugas: None

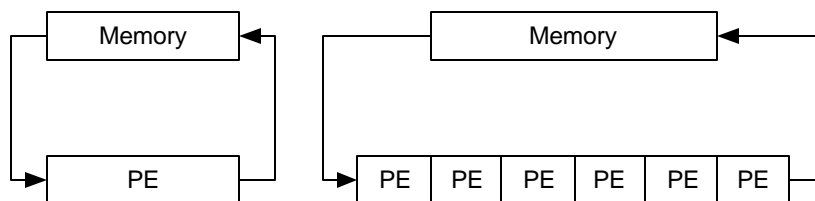
VLSI COMPUTING STRUCTURE

SYSTOLIC ARRAY ARCHITECTURE

Systolic architecture concept was developed by **Kung** and associates at Carnegie-Mellon University. The architecture consist of a set of interconnected cells, each capable of performing some simple operation. Because of the simplicity, only regular communication and control structures are required, giving substantial advantage over the more complex ones. Cells in the systolic system are typically interconnected to form a **systolic array** or **systolic tree** structure.

Information flows between cells in a pipelined fashion and communication with the outside world occurs only at the **boundary-cells**. Thus the I/O ports of the system are placed at those boundaries.

In basics systolic architecture replace single powerful PE with an array of simplified PEs. Higher computational throughput can be achieved without increasing memory bandwidth. The memory module is used to push the data though the array of PEs, ensuring that once a data item is brought from the memory it can be used effectively at each cell it passed.



The advantage of the systolic system is that it is very useful especially in a **compute-bound** data, where the same data is manipulated in different ways. The system can accommodate such usage with a very simple and efficient fashion with just one fetch and one store operation.

Systolic system also have high degree of expansion-ability, simple and regular data and control flow, simple and uniform cells, elimination of global broadcasting, limited fan-in and fast response time.

[POP QUIZ] Give argument on how systolic system give all those advantage mentioned above?

DAY 8 – SESSION 16

Tujuan: Memahami hubungan antara processor dan memory

Pokok bahasan: Functional Structures

Reference: Buku 1 (Chapter 7.1)

Tugas:: None

FUNCTIONAL STRUCTURES

MULTIPROCESSORS VS. MULTICOMPUTERS

Multiprocessor is characterized by at least 2 attributes:

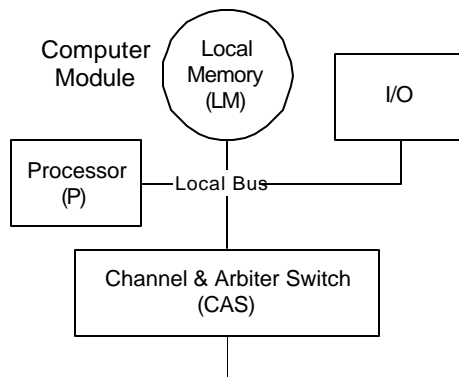
- A multiprocessor is a single computer that includes multiple processors
- Processors may communicate and cooperate at different levels in solving a given problem. The communication occurs by sending message from one processor to another or by sharing common memory space.

What is the difference between multi-computer and multi-processor? A multi-computer is a multiple computer system consisting of several autonomous computers, which may or may not communicate with each other. A multi-processor system consists of several processor controller by one operating system, which provide interaction between processors and their programs at the process, data set and data element levels.

ARCHITECTURE SET OF MULTIPROCESSORS

- **Tightly coupled** multiprocessors.
- **Loosely coupled** multiprocessors.

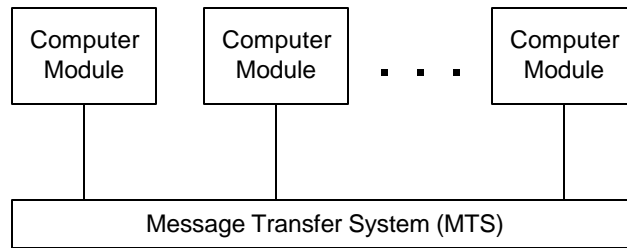
ARCHITECTURE 1: LOOSELY COUPLED MULTIPROCESSORS



Each processor has a set of input-output devices and a large local memory where it access most of the instructions and data. We called the processor, its local memory and I/O interfaces as **computer module**.

Process which executed on different computer modules communicate by exchanging messages through a **message-transfer-system** (MTS). The degree of coupling in such a system is very loose, thus it sometimes called **distributed system**.

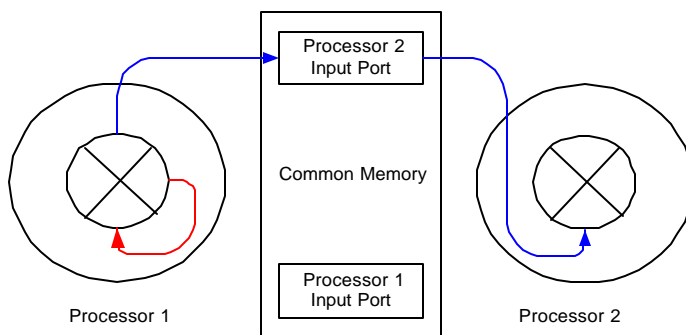
Loosely Coupled System (LCS) is efficient when the interaction between tasks are minimal, thus no communication overhead exist in the system's MTS.



If request from two or more computer module collide in accessing a physical segment of the MTS, the arbiter is responsible for choosing one of the simultaneous requests according to a given service discipline. It is also responsible for delaying other requests until the servicing of the selected request is completed. The channels in CAS may have high-speed communication memory which is used for buffering block transfer of messages.

The MTS can be a simple time shared bus or a shared memory system. For a **time shared bus** implementation, the performance is limited by the message arrival rate on the bus, the message length and the bus capacity. Contentions for the bus increase as the number of computer modules increases. For a **shared memory system** implementation, there are a set of memory modules and a processor-memory interconnection network or a multi-ported main memory. The limiting factor in the shared memory implementation is the memory conflict problem which imposed by the processor-memory interconnection network. The MTS is one of the most important factors that determine the performance of the multiprocessor system.

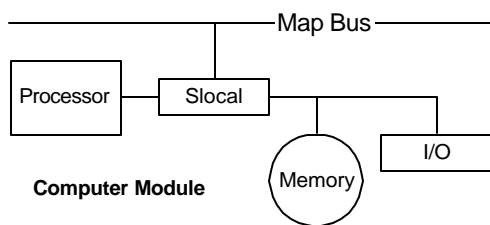
Process or tasks in one processor can communicate with other processes allocated at the same



processor, or with tasks allocated on another processors. Each task has their own input port stored in the local memory of the processor where the task is allocated on. Every message issued by the task is directed to the input port of the destination task. Communication between tasks in the same processor takes place through the local memory only. While

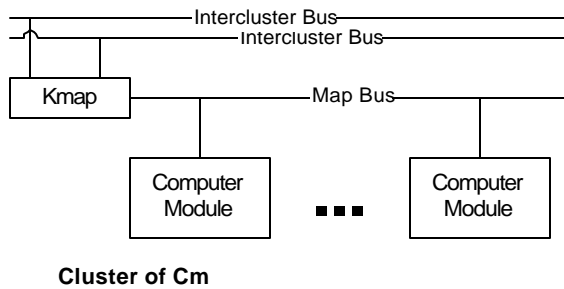
communication between tasks on the different processors utilize the communication port on the communication memory, which one port is associated with each processor as its input port.

The Cm* Architecture



The architecture above is a non-hierarchical LCS, for a hierarchical LCS we take example from a computer system project at Carnegie Mellon University called the **Cm* Architecture**. Each computer module in the Cm* includes a local switch called the **Slocal**. Slocal is similar in principal with CAS, it intercepts and routes the processor's requests to the memory and I/O devices outside

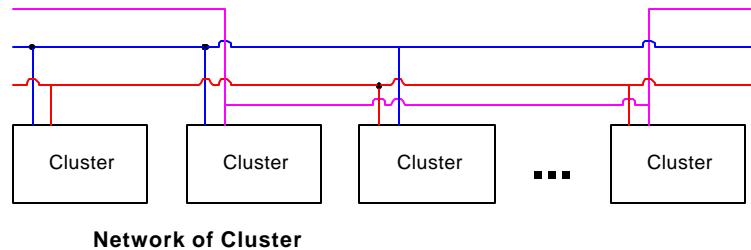
the computer module via a map bus. It also accepts references from other computer modules to its local memory and I/O devices.



The computer modules are connected in hierarchical clusters by 2 level buses, they are the Kmap and the map bus. A cluster is regarded as the lowest level which made up of computer modules. Clustering can enhance cooperative ability of the processors in a cluster to operate on shared data with low communication overhead. It also facilitates the execution of a group of tightly coupled cooperating processes. Any

non local reference to memory is handled by the Kmap in the cluster of the target memory module.

The map bus may create a bottleneck problem since only one transaction at a time can take place. Clusters communicate via inter-cluster buses, which are connected between Kmap.



The Kmap provides the address mapping, communication and synchronization functions within the system. The key operating system primitives can be moved into the Kmap and relieving the computer modules from major supervisor functions. There are 3 processors in the Kmap, they are Kbus, the Linc and the Pmap. [Further discussion on the Kmap can be found at Book 1, Chapter 7 in the reference]

ARCHITECTURE 2: TIGHTLY COUPLED MULTIPROCESSORS

Processors communicate with each other though a shared main memory, thus the rate of communication from one processor to the other depends on the bandwidth of the memory. A small local memory or buffer (cache) may exist in each processor to improve performance.

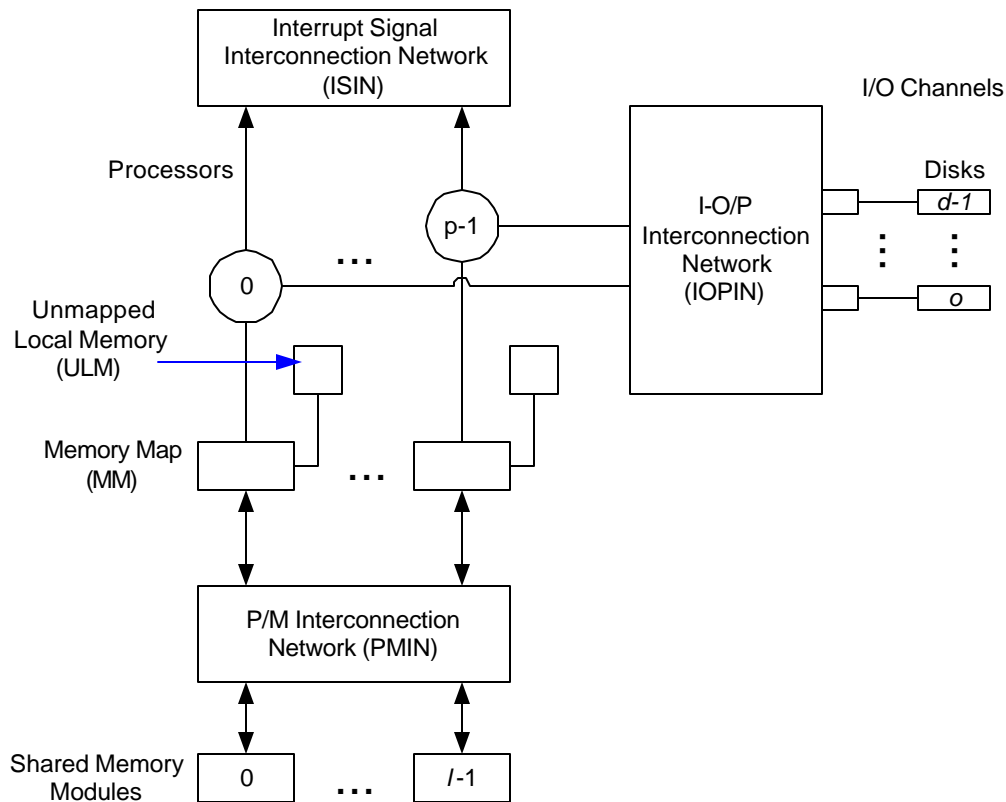
The connectivity between processor and memory is achieved by inserting an **interconnection network** between them or by using a **multi-ported memory**. Serious performance degradation may be suffered due to the memory contentions when two or more processor attempt to access the same memory space.

Tightly Coupled System (TCS) can tolerate higher degree of interaction between tasks running in one processor with the other tasks running on the other processor. The communication overhead rarely shows significant impact on TCS overall performance.

First Model of TCS

There are 2 models of TCS. The first one model consists of p processors, I memory modules and d input-output channels. These units are connected through a set of 3 interconnection networks, called the Processor-Memory Interconnection Network (PMIN), the I/O processor interconnection network (IOPIN) and the interrupt signal interconnection network (ISIN).

PMIN is a switch, which can connect every processor to every memory module, implemented using a p by I crossbar which has $p \cdot I$ set of cross points, or PMIN could also be a multistage network.

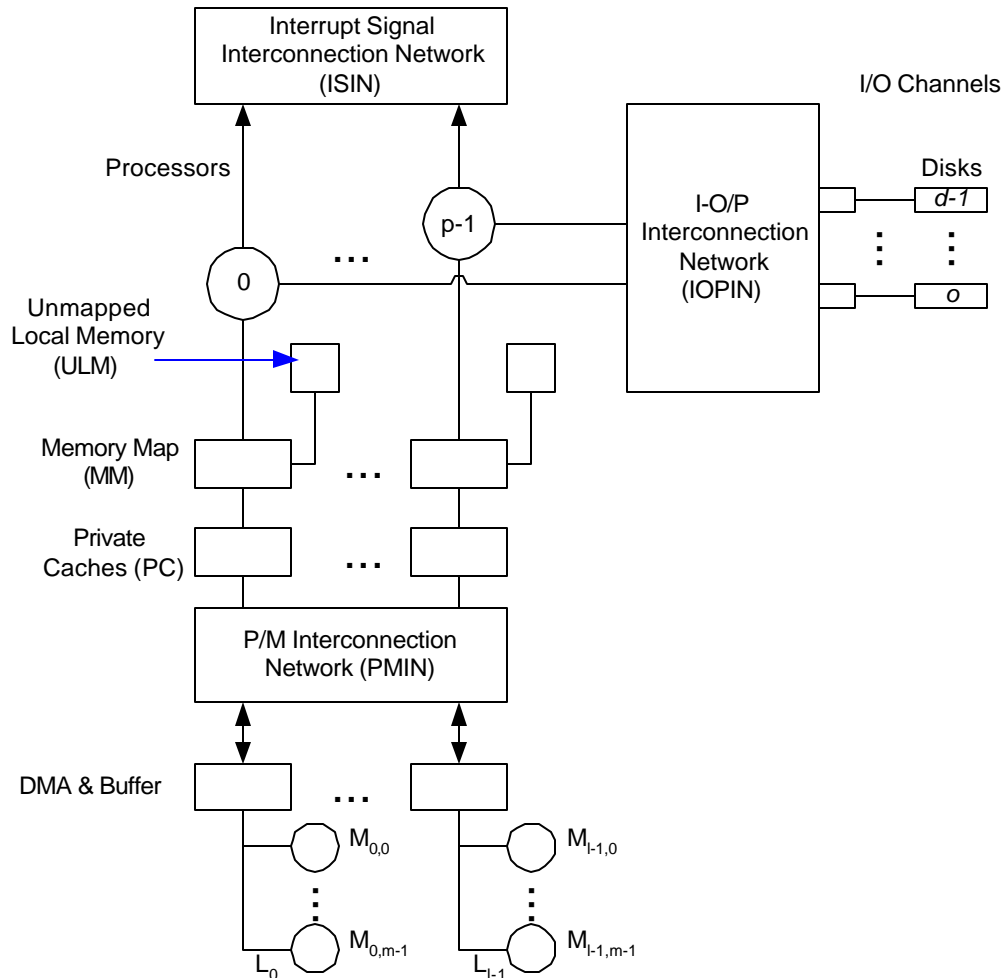


A memory module can satisfy only one processor's request in a given memory cycle. Hence if two or more processors attempt to access the same memory module, a conflict occurs which PMIN has to resolved. To avoid excessive conflicts, the number of memory modules is usually as large as the number of processor or more. Another method to reduce conflicts is to associate a reserved storage area with each processor, which is the Unmapped Local Memory (ULM). ULM store Kernel code and OS tables often used by the processes running on that processor, thus the ULM helps to reduce traffic to the PMIN and hence reduce the degree of conflict.

Each processor in the first model of TCS are allowed to make memory references which are accessed in main memory. These memory references contribute to the memory conflict at the memory modules and since memory references goes through PMIN, it also suffer delays and thus increase instruction cycle time and reduce the system throughput.

Second Model of TCS

To overcome the disadvantage of the first model, a cache is introduced to the system. A **cache** is associated with each processor to capture most of the references made by a processor and thus reducing the traffic through the crossbar switch.



Despite the improved performance, this model encounter the cache coherence problem. More than one inconsistent copy of data may exist in the system.

There is a module attached to each processor that directs the memory references to either the ULM or the private cache of that processor. This module is called the memory map and is similar in operation to Slocal in C_m^* Architecture.