MODULE 13: Selected Topics

Lecture 13.2
Parallel and Multiprocessor Architectures

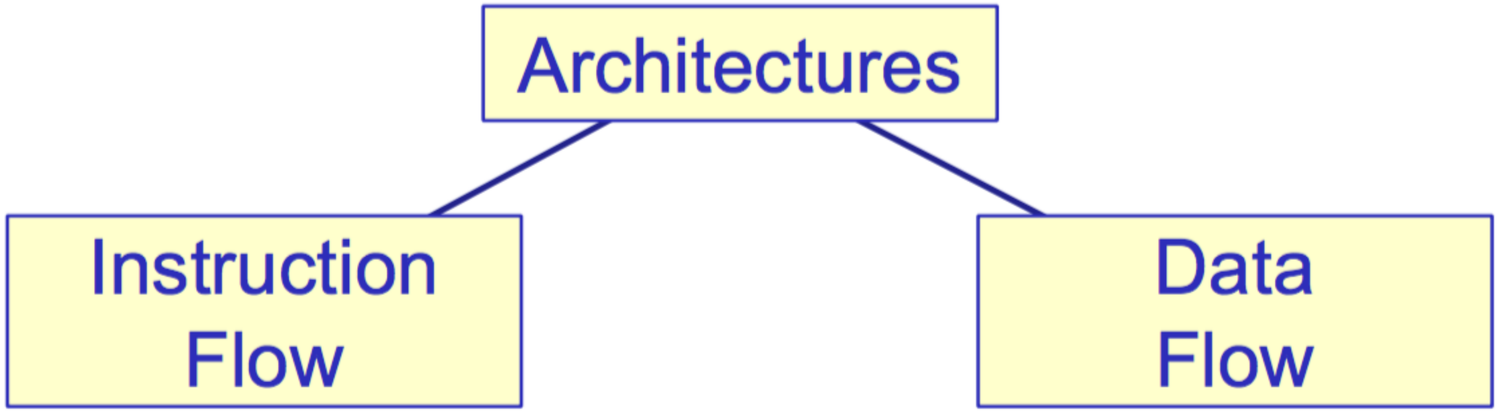
Prepared By:
- Scott F. Midkiff, PhD
- Luiz A. DaSilva, PhD
- Kendall E. Giles, PhD

Electrical and Computer Engineering
Virginia Tech

Lecture 13.2 Objectives

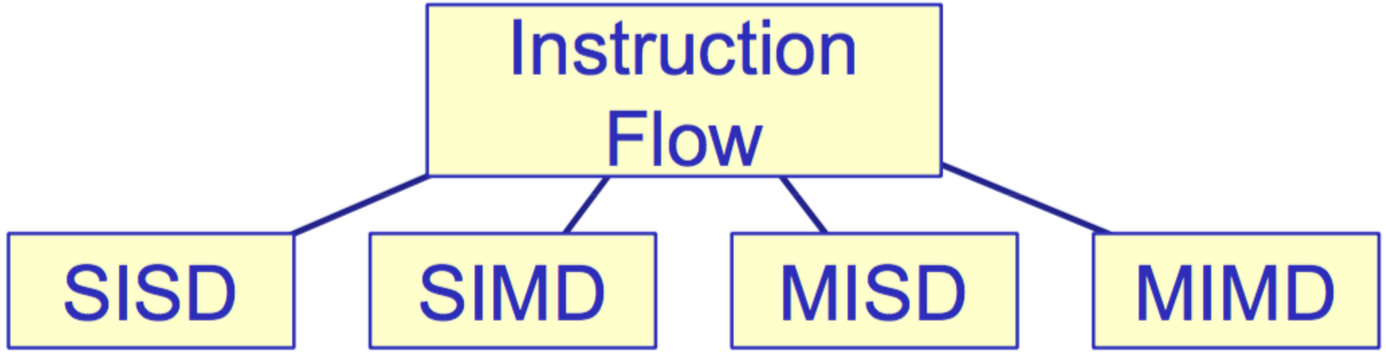
- Use Flynn's taxonomy to describe computer architectures in terms of their parallelism in data processing and/or instruction processing
- Describe basic properties of different types of parallel and multiprocessor architectures, including:
  - Superscalar and VLIW processors
  - Vector processors
  - Shared memory multiprocessors
  - Distributed computing systems

Types of Architectures



- Instruction flow architectures
  - Control is driven by an instruction stream (or a set of instruction streams)
  - Common processors are instruction flow architectures
- Data flow architectures
  - Data driven – organized around data

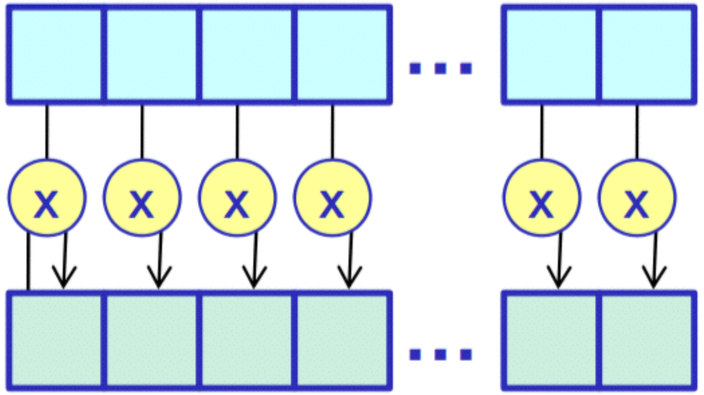
Flynn's Taxonomy



- Classification of instruction flow systems based on number of concurrent instructions for control and data operations
- Four categories:
  - Single instruction stream, single data (SISD)
  - Single instruction stream, multiple data (SIMD)
  - Multiple instruction stream, single data (MISD)
  - Multiple instruction stream, multiple data (MIMD)

Flynn's Taxonomy (cont'd)

- Single instruction, single data (SISD)
  - Single instruction operating on a single data item, such as ADD AX, BX
  - Conventional processors have an SISD architecture
- Single instruction, multiple data (SIMD)
  - Single instruction stream, but operating simultaneously on multiple data items, e.g. in an array or vector



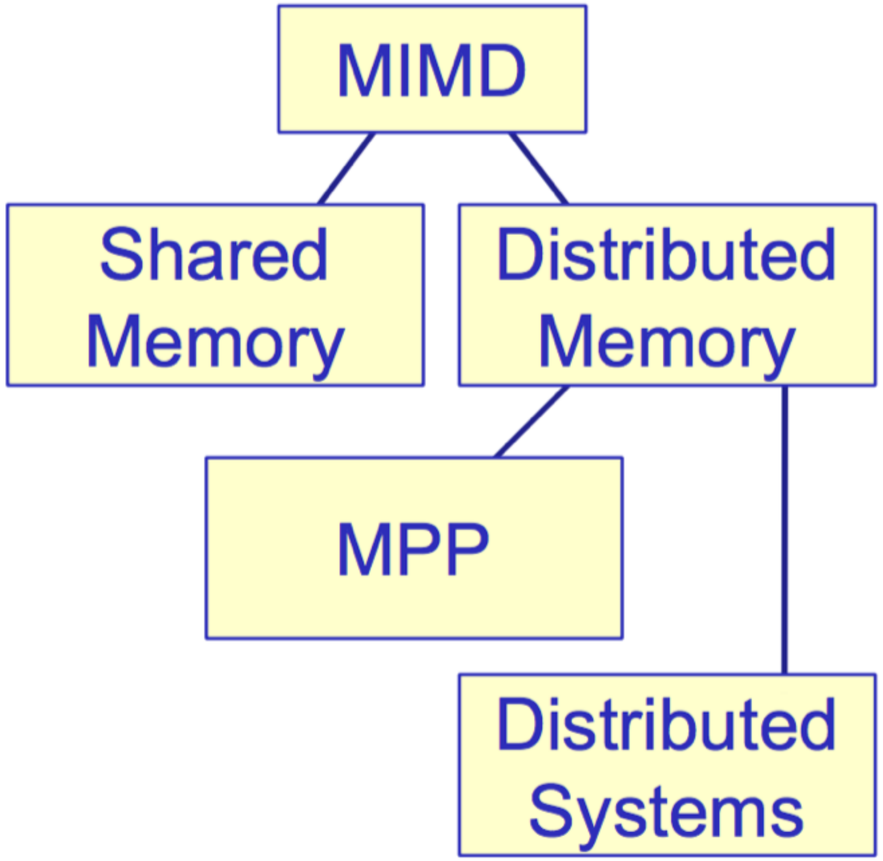
Array Multiplication (element-by-element)

Flynn's Taxonomy (cont'd 2)

- Multiple instruction, single data (MISD)
  - Not used in practice
- Multiple instruction, multiple data (MIMD)
  - Realized by a variety of different parallel and multiprocessor architectures
- Single program, multiple data (SPMD)
  - Sometimes added as an additional category
  - Is a specialized MIMD architecture with instruction control being replicated so different processors can be executing at different parts of the same program

Types of MIMD Machines

- Shared memory machines have a single memory that is shared by multiple processors
- Distributed memory machines have memory assigned to each instruction execution unit (processor)
  - Massively parallel processing (MPP)
  - Distributed computing systems



Multiprocessor and Parallel Architectures

- Typical multiprocessor and parallel architectures
  - Super scalar and VLIW processors
  - Vector processors
  - Shared memory multiprocessors
  - Distributed computers
- Interconnect networks are essential for some architectures
  - Processor-to-processor connections, e.g. for message passing
  - Processor-to-memory connections, e.g. for accessing shared memory

CHECK POINT

As a checkpoint of your understanding, please pause the video and make sure you can do the following:

- Use Flynn's taxonomy to describe computer architectures in terms of their parallelism in data processing and/or instruction processing

If you have any difficulties, please review the lecture video before continuing.

Superscalar and VLIW

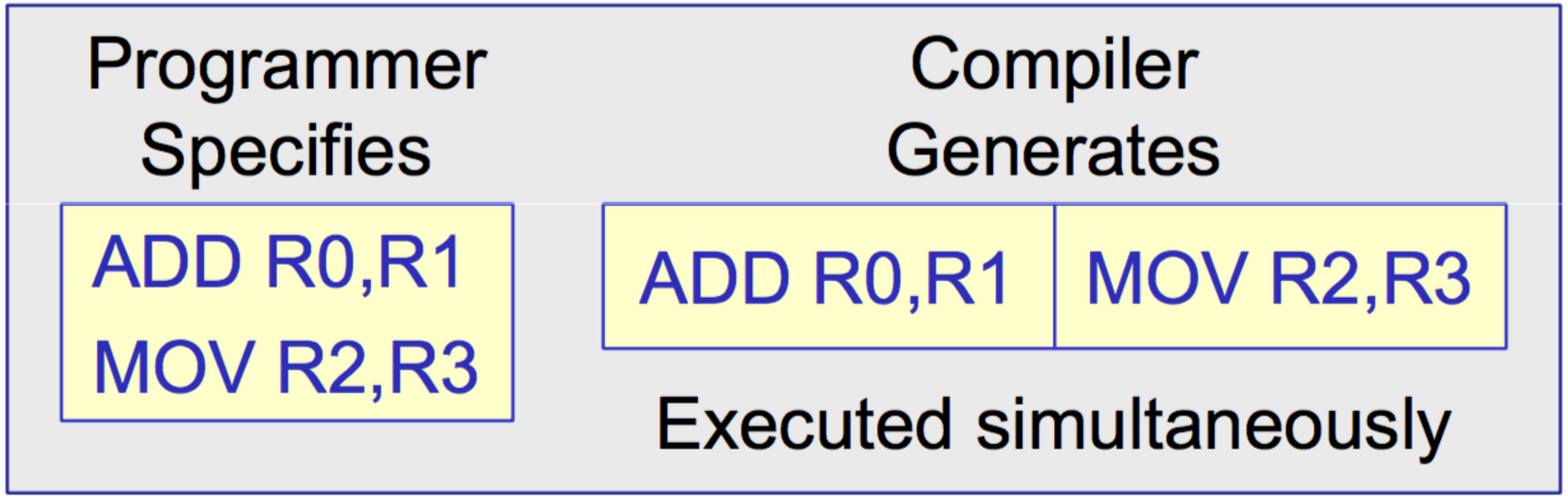
- Superscalar and very-long instruction word (VLIW) processors both extract parallelism at the instruction level
- Superscalar processors
  - Hardware design enables the execution (sometimes) of more than one instruction at a time
  - The processor, not the programmer or compiler, extracts parallelism (although compiler can help)

```
ADD R0,R1
MOV R2,R3
```

The two instructions can be fetched, decoded and executed in parallel

Superscalar and VLIW (cont'd)

- Very-long instruction word (VLIW) processors
  - An instruction can specify multiple operations
  - The compiler extracts parallelism and specifies it in the (very long) instruction word
  - Of course, parallel execution is not always possible



Vector Processors

- Vector processors use hardware specifically designed to operate on vectors and/or two-dimensional matrices
  - Vector registers to hold multiple vector elements
  - Execution units to simultaneously operate on multiple vector elements
- Application in vector- and array-based scientific computing

```
for i = 1 to N {
   A[i] = B[i] + C[i];
}
```

Conventional

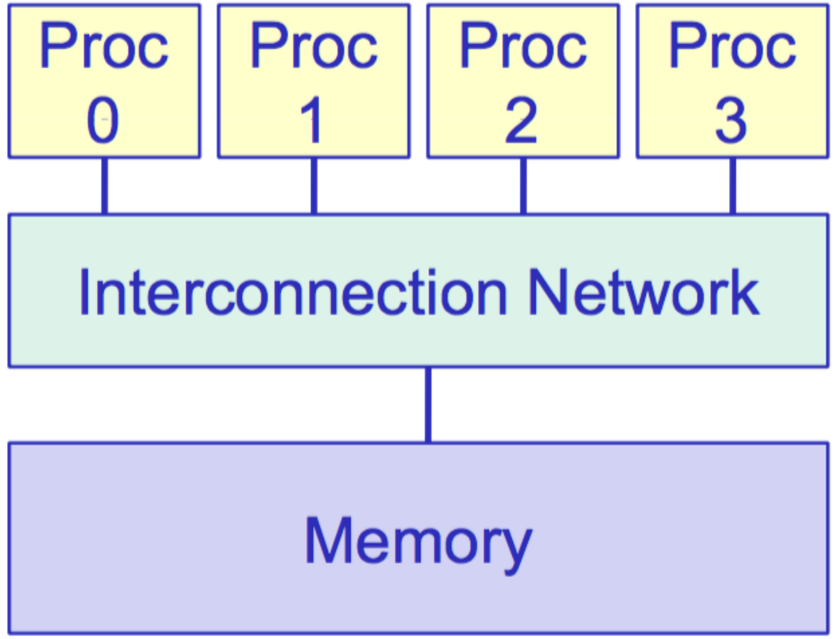
```
LOADV    B,R1
LOADV    C,R2
ADDV     R0,R1,R2
STOREV   R0,A
```

Vector

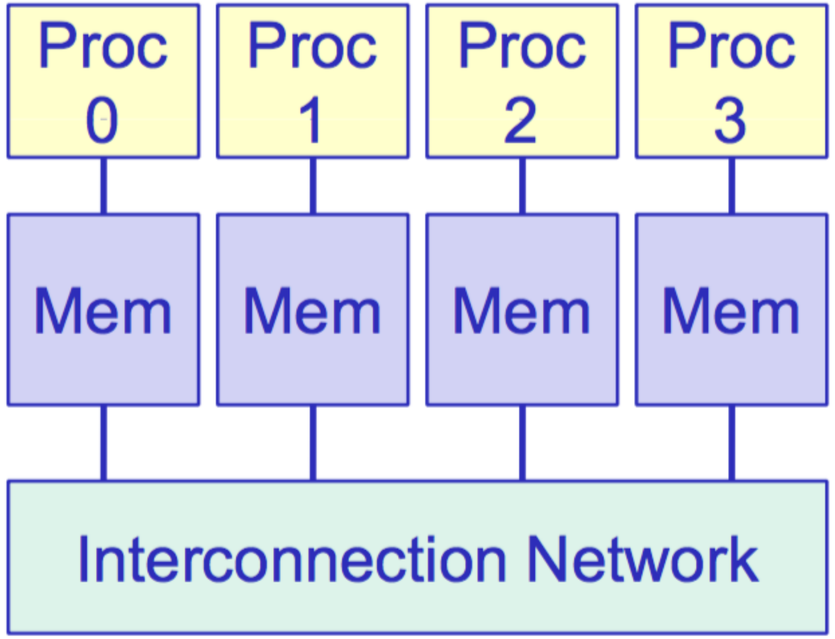
Shared Memory Multiprocessors

- A shared memory multiprocessor is an MIMD computer where multiple processing units execute instruction streams on different data elements with the data being in a shared or common memory
- Sharing maybe actual or logical
  - Global shared memory – the sharing may be physical in that a single physical memory unit is accessed by all processors
  - Distributed shared memory – the sharing may be logical in that memory is associated with each processor, but can be accessed by any processor

Shared Memory Multiprocessors (cont'd)



Global Shared Memory



Distributed Shared Memory

CHECK POINT

As a checkpoint of your understanding, please pause the video and make sure you can do the following:

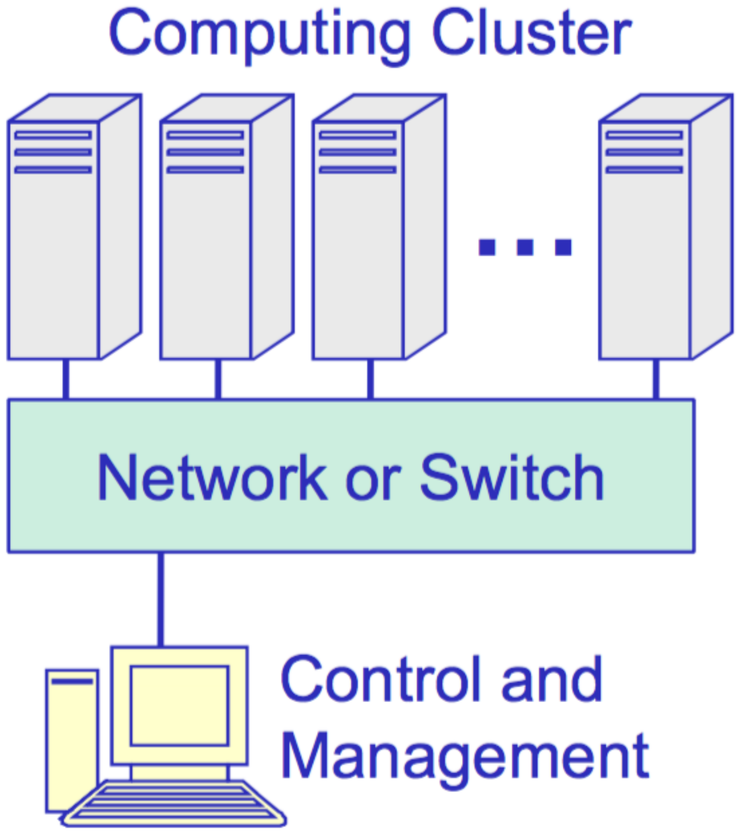
- Describe basic properties of Superscalar and VLIW processors, Vector Processors, and Shared Memory Multiprocessors

If you have any difficulties, please review the lecture video before continuing.

Distributed Computing Systems

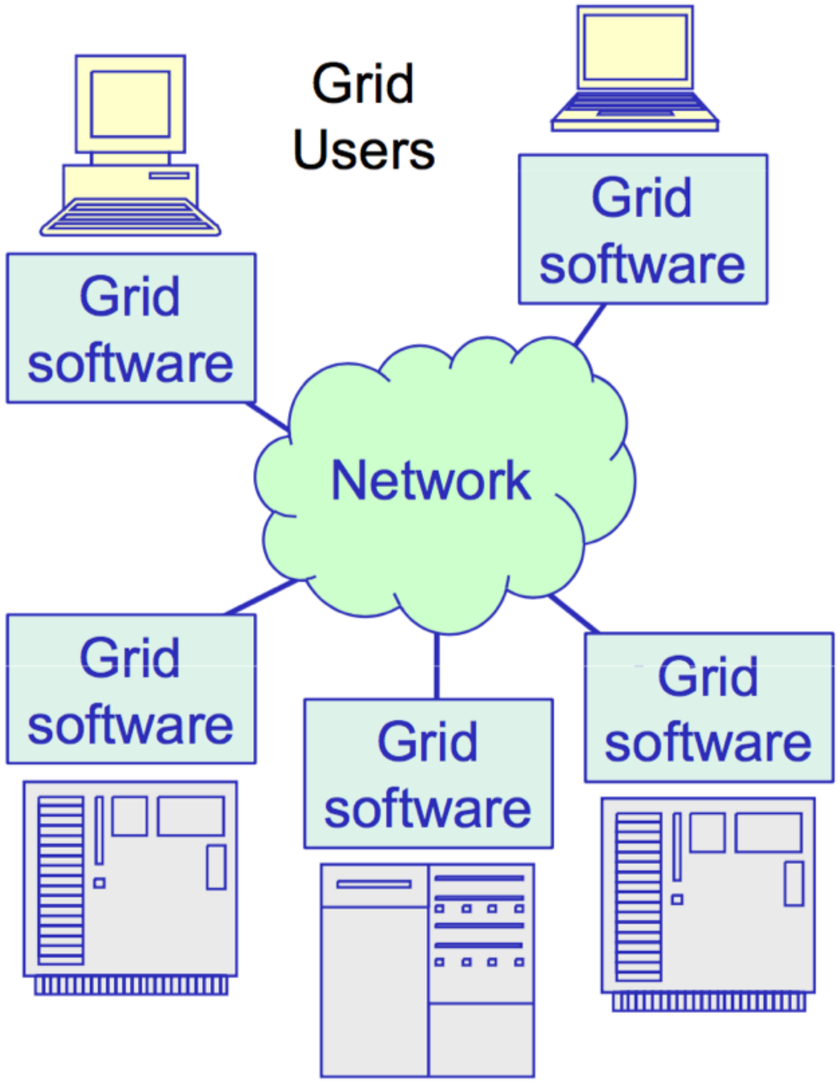
- Distributed computing has been propelled by:
  - Low-cost, powerful general-purpose computers
  - Low-cost, high data rate networks
- As an alternative to specialized, high-performance vector or shared memory supercomputers, distributed computing systems offer high-levels of parallelism with conventional components at relatively low cost
- Challenges
  - Network latency (not data rate)
  - Course-grain parallelism to reduce communication
  - Reliability
  - Input/output rates

Cluster Computing



- Distributed memory, although a shared memory model can be presented to the programmer
- Interconnect network is a standard high-data rate local area network or a specialized switch to achieve low latency
- System software (e.g., Beowulf) needed to coordinate and manage resources
- Programming is often a challenge

Grid Computing



- Grid computing shares heterogeneous resources (computation, storage, etc.) are connected via a network (high-performance Internet)
- System software provides a coherent programming model and management tools
- Goal is to “virtualize” computing resources

Interconnection Networks

- Shared memory systems and distributed computing systems require connectivity
- Processor-to-processor
- Processor-to-memory
- Local area network and wide area network technologies are sometimes used
- Off-the-shelf, good performance at low cost
- High latency
- Throughput may be limited
- Alternatives are needed for many applications

CHECK POINT

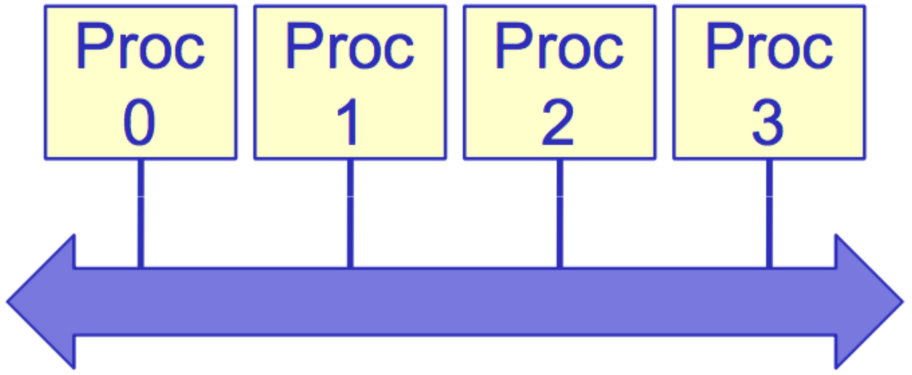
As a checkpoint of your understanding, please pause the video and make sure you can do the following:

- Describe basic properties of Distributed computing systems

If you have any difficulties, please review the lecture video before continuing.

Bus-based Networks

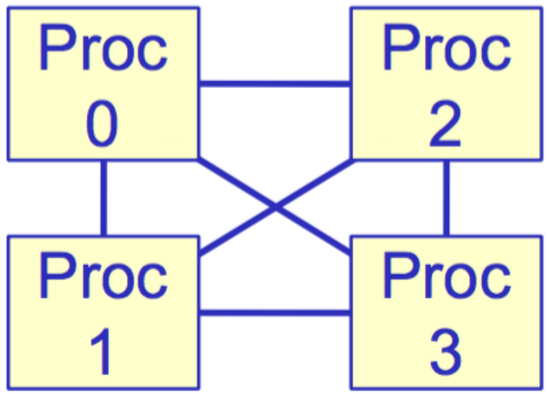
- Bus-based networks use a shared bus to connect processors to each other (below) or processors to memory
- Relatively low cost
- Does not scale well – poor performance if heavily used or if there are a large number of devices to connect



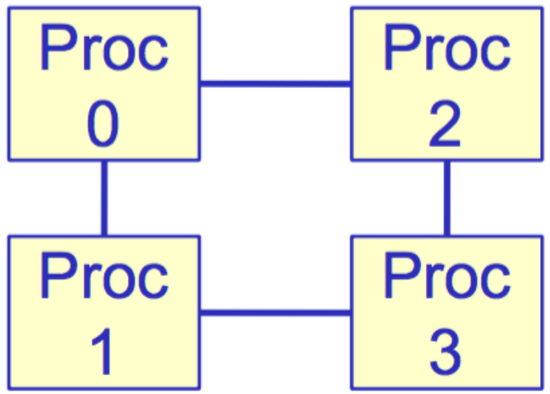
Bus-based Network

Point-to-Point Networks

- Processors (and memories) can be directly connected by point-to-point links
- A fully-connected mesh network provides complete connectivity, but is expensive
- Partially-connected topologies are cheaper and can be matched to communication requirements



Fully Connected

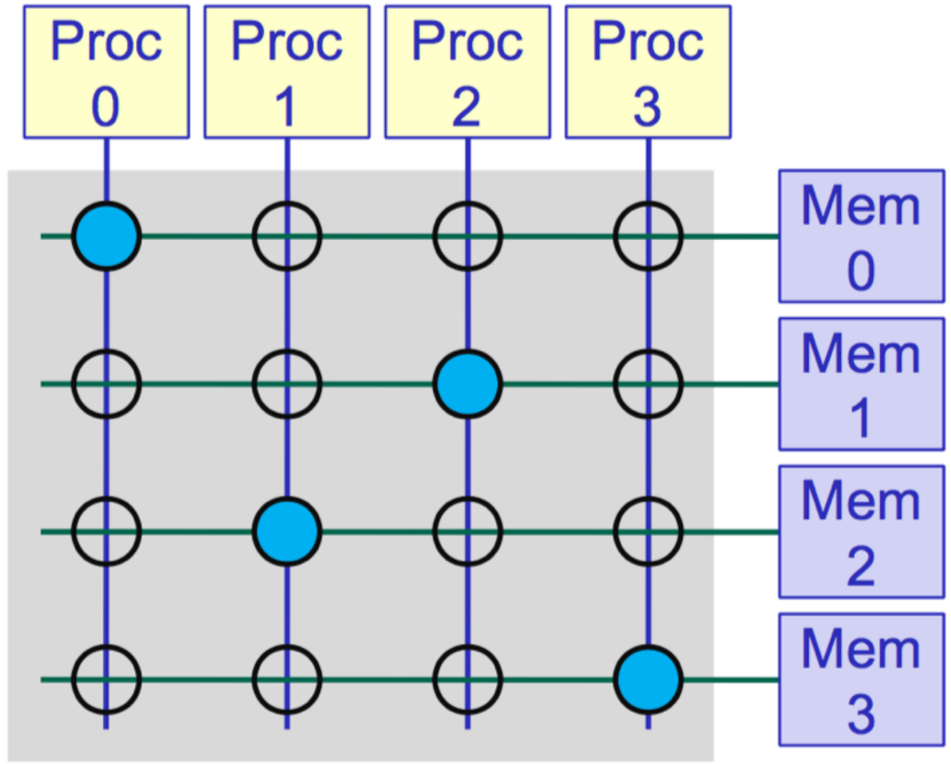


Partially Connected

Switched Networks

- Switches can connect processors to processors or to memory
- Point-to-point links replaced by a more structured switch organization
- Full connectivity via a crossbar switch, but expensive
- Multi-stage switches can reduce cost
  - Non-blocking – any end-to-end connection is possible at any given time
  - Blocking – some connections may not be possible depending on other active connections

Crossbar Switch

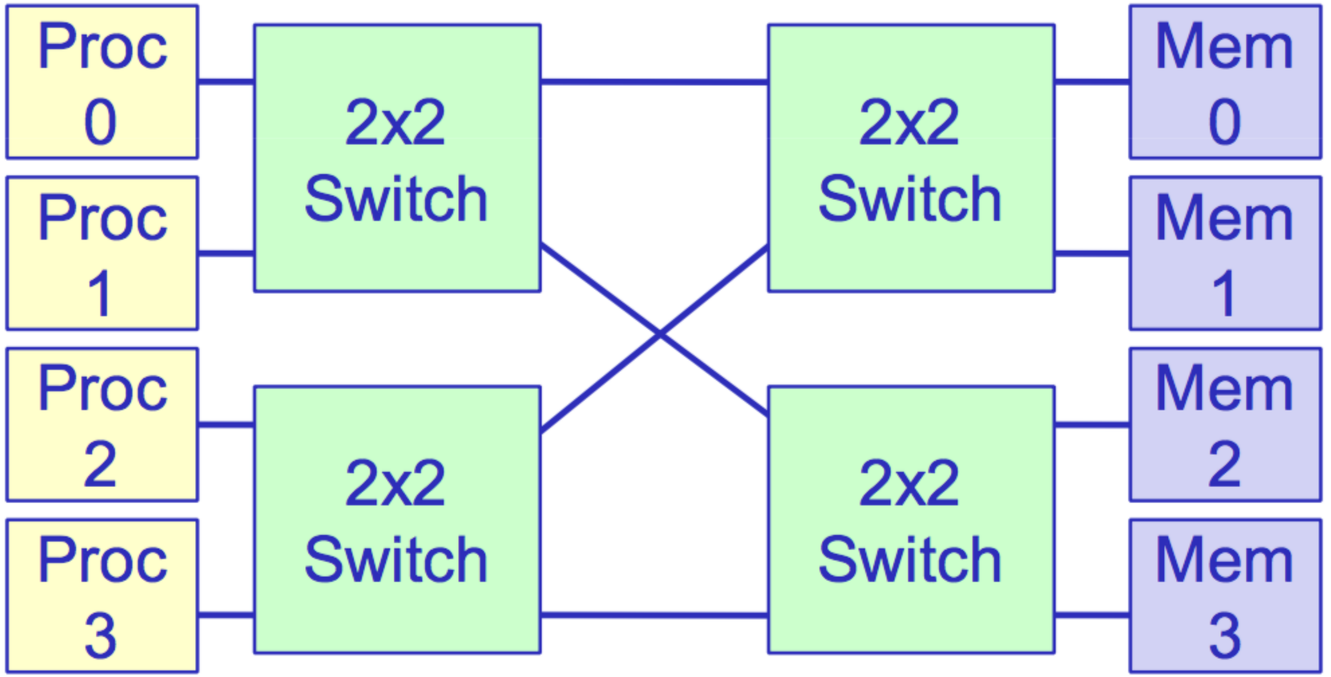


Example Connections

Proc 0 – Mem 0
Proc 1 – Mem 2
Proc 2 – Mem 1
Proc 3 – Mem 3

- Complexity grows a N^2
- Usually impractical

Multistage Switch



- Complexity is reduced from crossbar
- But may add latency or introduce blocking

CHECK POINT

As a checkpoint of your understanding, please pause the video and make sure you can do the following:

- Describe basic properties of Point-to-Point and Switched Networks

If you have any difficulties, please review the lecture video before continuing.

Summary

- Computers may be driven by instructions (instruction flow) or data (data flow), with instruction flow systems being the most common by far
- Flynn's taxonomy considers the number of instructions and number of data elements being operated on to yield SISD, SIMD, MISD, and MIMD types of architectures
- MIMD machines may have shared or distributed memory
- Superscalar processors extract instruction-level parallelism directly, while VLIW processors rely on the compiler to extract the parallelism

Summary (cont'd)

- Vector and array processors are designed for vector and array operations, common in scientific computation
- Shared memory multiprocessors allow multiple processors to access a common memory, using either an actual shared memory or a logically shared memory that is physically distributed
- Cluster computing and grid computing are examples of distributed computing systems
- Bus, crossbar, or multistage interconnect networks are used to connect processors-to-processors and processors- to-memory in parallel and multiprocessor architectures

MODULE 13: Selected Topics

Lecture 13.2
Parallel and Multiprocessor Architectures

Prepared By:
- Scott F. Midkiff, PhD
- Luiz A. DaSilva, PhD
- Kendall E. Giles, PhD

Electrical and Computer Engineering
Virginia Tech