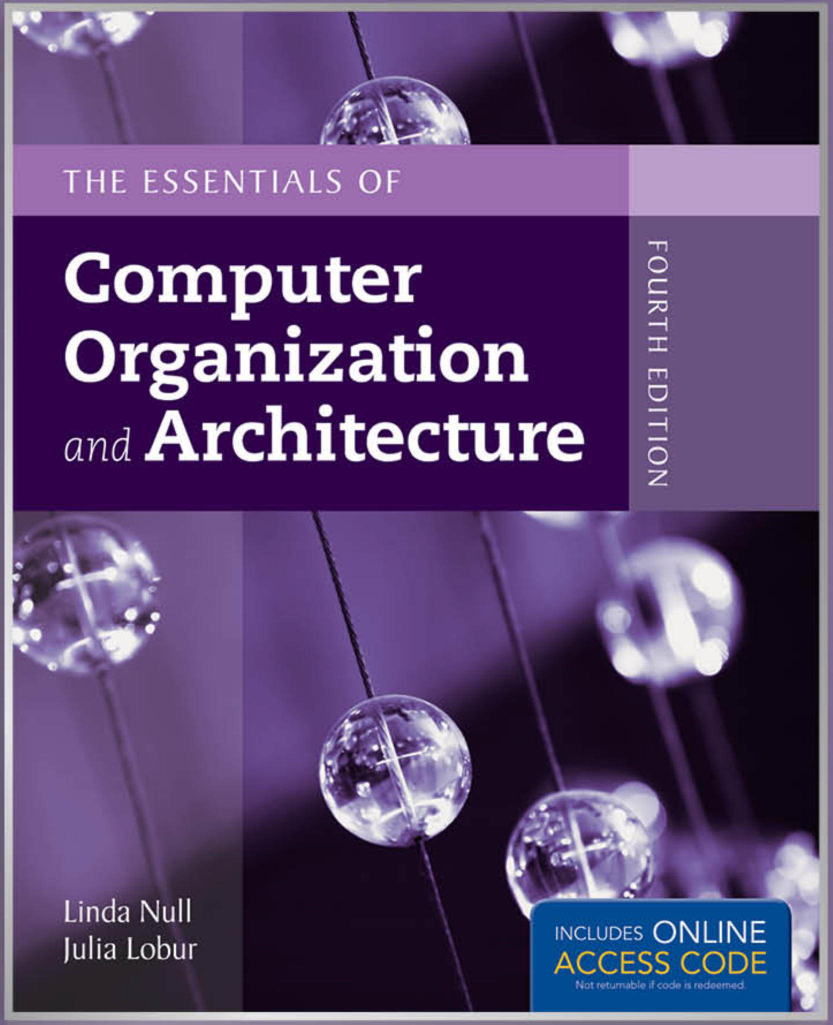
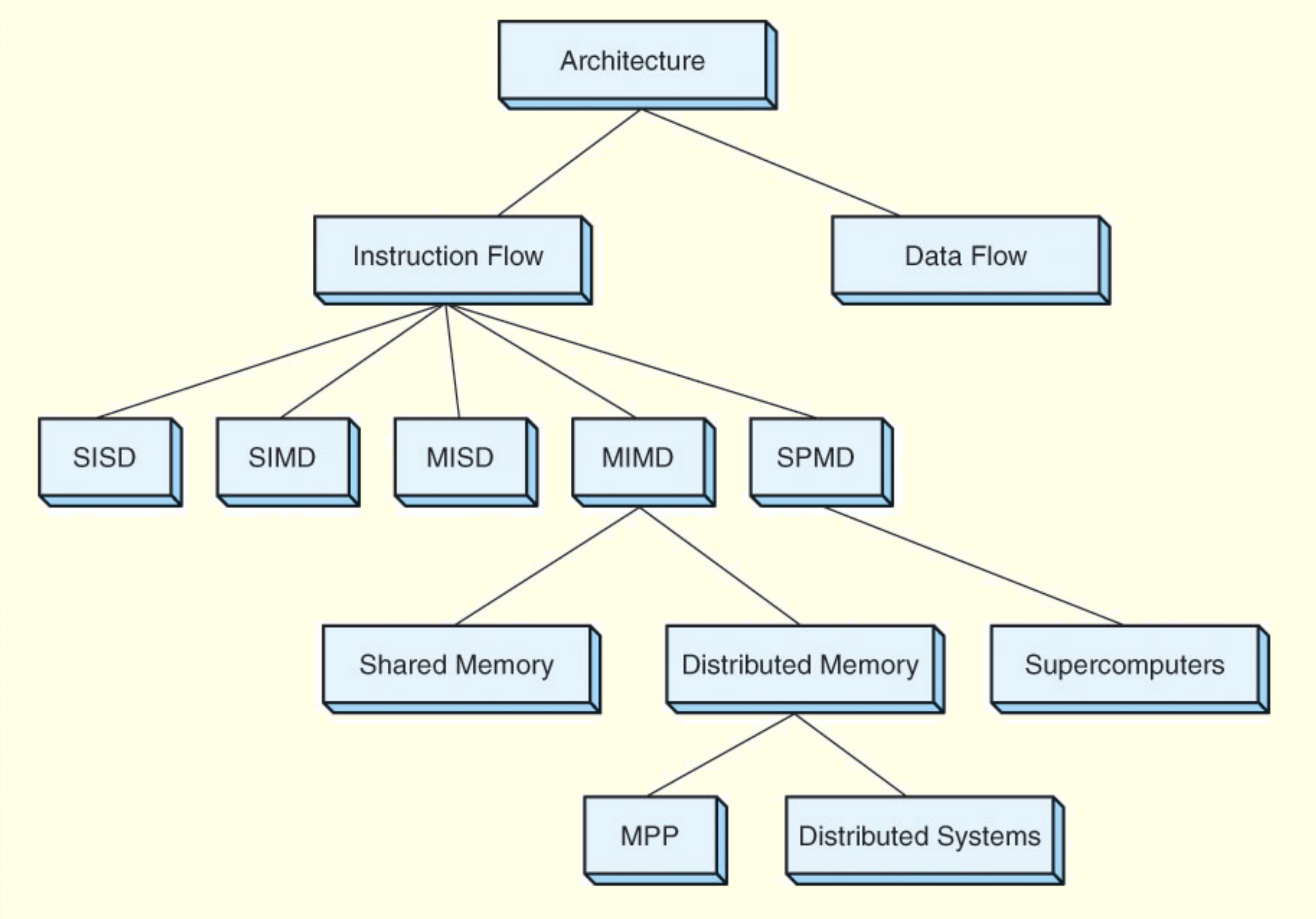
Chapter 9

Alternative Architectures


THE ESSENTIALS OF
Computer Organization and Architecture
FOURTH EDITION
Linda Null
Julia Lobur
INCLUDES ONLINE ACCESS CODE
Not returnable if code is redeemed.

9.3 Flynn's Taxonomy



9.4 Parallel and Multiprocessor Architectures

- **Parallel processing** is capable of economically **increasing system performance**.

- The **limiting factor** is that no matter how well an algorithm is parallelized, there is always **some portion that must be done sequentially**.

- It is important to keep in mind that an ***n*-fold increase in processing** power does **not necessarily result in an *n* -fold increase in performance**.

9.4 Parallel and Multiprocessor Architectures

- **Pipelining divides** the **fetch-decode-execute** cycle **into single-cycle stages** that each carry out a small part of the process on a set of instructions.
- **Super pipelining** occurs when a **pipeline has substages that require less than half a clock cycle** to complete.
- The super pipeline is equipped with a **separate clock** running at a **frequency that is at least double** that of the main system clock.

9.4 Parallel and Multiprocessor Architectures

- **Superscalar architectures** include **multiple execution units** such as specialized integer and floating-point adders and multipliers.

- The superscalar **instruction fetch unit** can **simultaneously retrieve several instructions** from memory.

- The **hardware decoding unit determines which** of these **instructions can be executed in parallel and combines them**.

9.4 Parallel and Multiprocessor Architectures

- **Very long instruction word** (VLIW) architectures **use** a special **compiler** that **combines independent instructions into one long instruction** that is sent down the pipeline.

- One could argue that this is **better** than the hardware-based superscalar approach **because the compiler can better identify dependencies**.

- A counterargument is that **compilers good with static data but cannot have a view of the run time code**.

9.4 Parallel and Multiprocessor Architectures

- **Vector processors** are processors that **operate on entire vectors** or matrices at once.
- They are highly pipelined so that **arithmetic instructions can be overlapped**.
- Vector processors can be **categorized based on how operands are accessed**:
  - **Register-register** vector processors require all operands to be in registers.
  - **Memory-memory** vector processors allow operands to be sent from memory directly to the arithmetic units.

9.4 Parallel and Multiprocessor Architectures

- A **disadvantage** of **register-register** vector computers is that **large vectors must be divided into segments** to fit into the registers.

- **Memory-memory** vector computers have a **longer startup time until the pipeline becomes full**.

- Vector processors are **efficient because sets of values can be prefetched** since the processor knows it will have a continuous stream of data.

9.4 Parallel and Multiprocessor Architectures

- **MIMD** systems can **communicate through shared memory or** through an **interconnection network**.

- Interconnection networks are **classified** based on their **topology**, **routing** strategy, and **switching** technique.

- The **topology** is a **major determining factor** in the **overhead cost of message passing**.

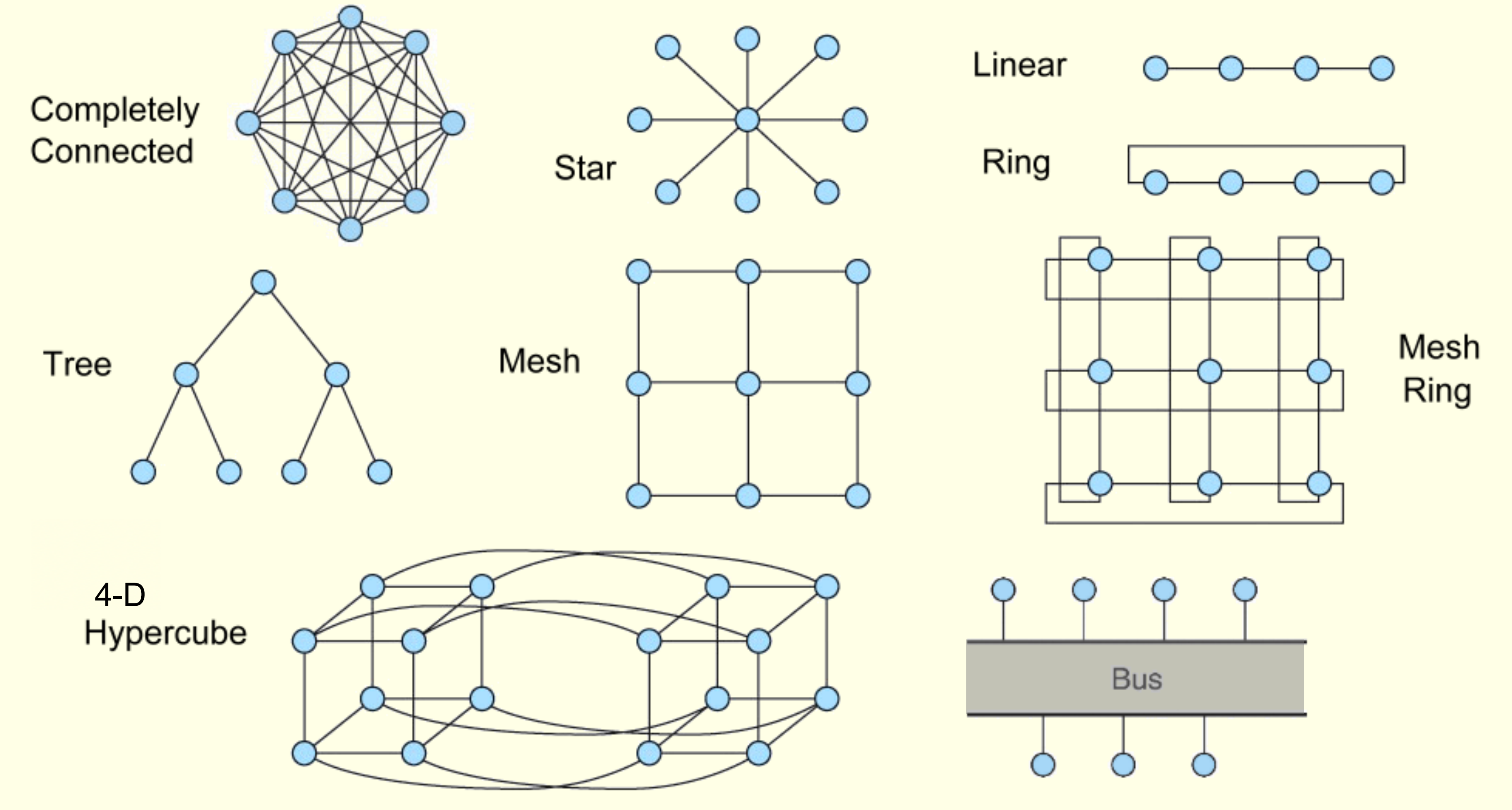
9.4 Parallel and Multiprocessor Architectures

- **Interconnection networks** can be either **static or dynamic**.

- **Processor-to-memory connections** usually employ **dynamic** interconnections **based on switches**.

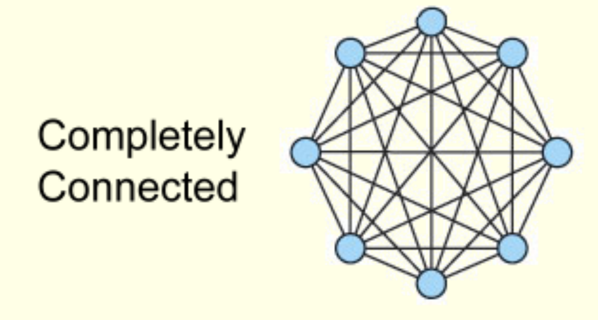
- **Processor-to-processor message-passing interconnections** are usually **static** and can employ any of several **different topologies**.

9.4 Processor-to-Processor Message-Passing Topologies



What is the max. length of the shortest path between arbitrary two nodes in these topologies?

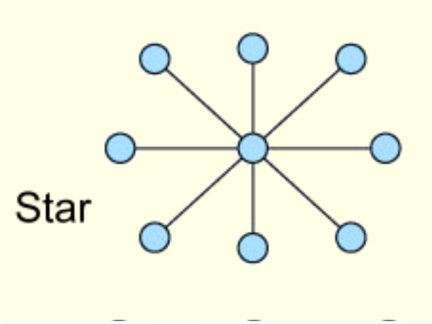
9.4 Processor-to-Processor Message-Passing Topologies



- Direct communication between any pair of nodes possible
- Durable network that isn’t dependent on any one node
- Very secure due to 1-to-1 communication
- Suitable for small-sized networks

- Requires an extremely large amount of cable
- Takes a long time to set up
- Requires meticulous planning
- There is a limit to the number of cables each node can accommodate
- Difficult to add further nodes

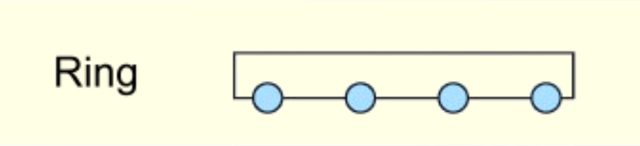
9.4 Processor-to-Processor Message-Passing Topologies



- At most two 'hops' for each pair of communicating nodes
- Easy to add and remove nodes
- Low cable usage
- Failure of an outer node will not affect other nodes in a network

- Requires specialist network hardware (hub node)
- A finite number of hub ports limits the network's size
- Performance of transmission depends on the hub.
- Failure of the hub will stop any transmission.

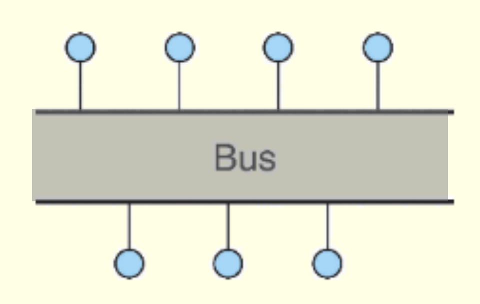
9.4 Processor-to-Processor Message-Passing Topologies



- Cheap to set up
- Easy to expand
- Dual ring option provides continuity through redundancy

- One faulty node or cable will bring the entire network down
- Performance declines with each additional node
- Reorganizing the network requires a full system shutdown

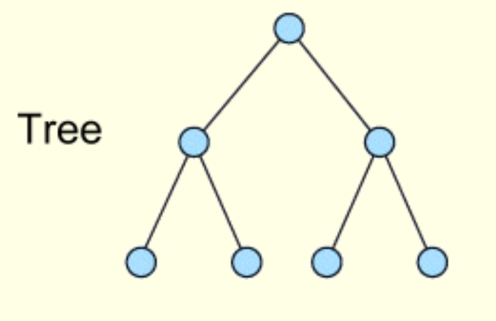
9.4 Processor-to-Processor Message-Passing Topologies



- Easy to connect a device and handle
- Takes less time to set up
- Best-suited for small networks.
- Easy to expand.

- If the bus cable fails, then the whole network will be down.
- Data can only travel in one direction at any point in time
- No communication can happen in parallel unless expensive parallel bus is used

9.4 Processor-to-Processor Message-Passing Topologies

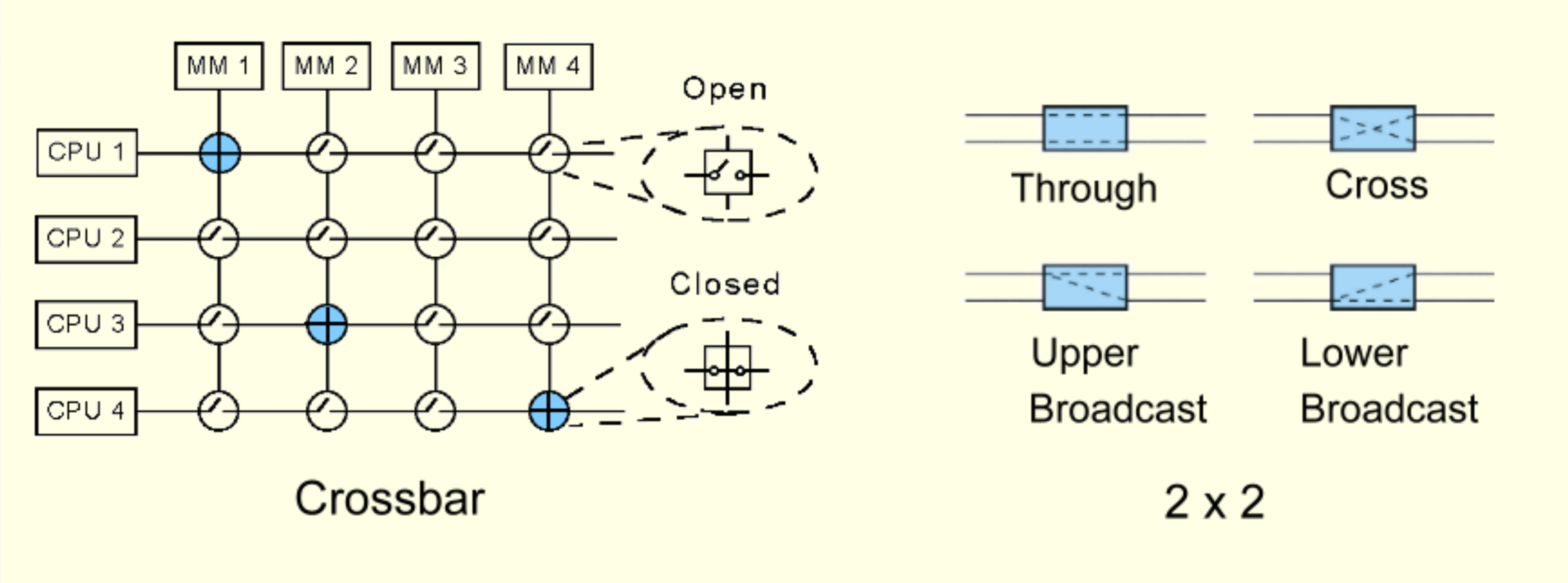


- Hybrid of bus and star topology
- Expandable
- Suitable for implementing divide-and-conquer approaches

- Network depends on the root node
- More cables needed in comparison to most other topologies

9.4 Parallel and Multiprocessor Architectures

- **Dynamic routing** is achieved **through switching** networks that consist of **crossbar** switches **or 2 × 2** switches.

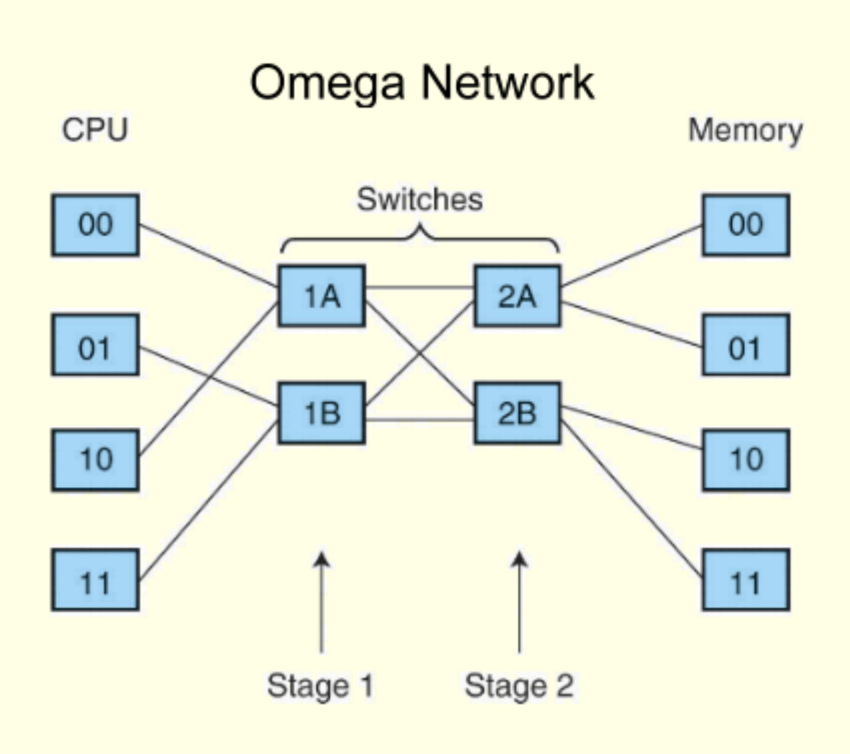


Quadratic number of switches required

9.4 Parallel and Multiprocessor Architectures

- **Multistage interconnection networks based on 2 x 2 switches**:

- $\log_2 n$ stages with n/2 switches per stage required



Chapter 9 Conclusion

- The common **distinctions between RISC and CISC systems** include RISC's short, fixed-length instructions. RISC ISAs are load-store architectures. These things permit RISC systems to be highly pipelined.

- **Flynn's Taxonomy** provides a way to **classify multiprocessor systems** based upon the number of processors and data streams.

Chapter 9 Conclusion

- **Massively parallel processors** have many processors, distributed memory, and computational elements communicate through a network. **Symmetric multiprocessors** have fewer processors and communicate through shared memory.

- Characteristics of **superscalar design** include **super pipelining**, and specialized instruction fetch and decoding units.

Chapter 9 Conclusion

- **Very long instruction word** (VLIW) architectures differ from superscalar architectures because the compiler, instead of a decoding unit, creates long instructions.
- **Vector computers** are highly-pipelined processors that operate on entire vectors or matrices at once.
- **MIMD systems** communicate through networks. The **network topology** often determines throughput.