University of Babylon, College of science for women
Dept. of Computer science

Computer Architecture

Second year

Dr. Salah Al-Obaidi

Lecture #10: Interconnection Structures Spring 2024

Contents

11. Interconnection Structures

11.1 Computer Components

Virtually all contemporary computer designs are based on concepts developed by John von Neumann at the Institute for Advanced Studies, Princeton. Such a design is referred to as the *von Neumann architecture* and is based on three key concepts:

- Data and instructions are stored in a single read–write memory.
- The contents of this memory are addressable by location, without regard to the type of data contained there.
- Execution occurs in a sequential fashion from one instruction to the next.

A computer consists of a set of components or modules of three basic types (processor, memory, I/O) that communicate with each other (Figure 11.1). In effect, a computer is a network of basic modules. Thus, there must be paths for connecting the modules. The collection of paths connecting the various modules is called the **interconnection structure**. The design of this structure will depend on the exchanges that must be made among modules.

Over the years, a number of interconnection structures have been tried. By far the most common are (1) the **bus** and various multiple-bus structures, and (2) **point-to-point interconnection** structures with packetized data transfer.

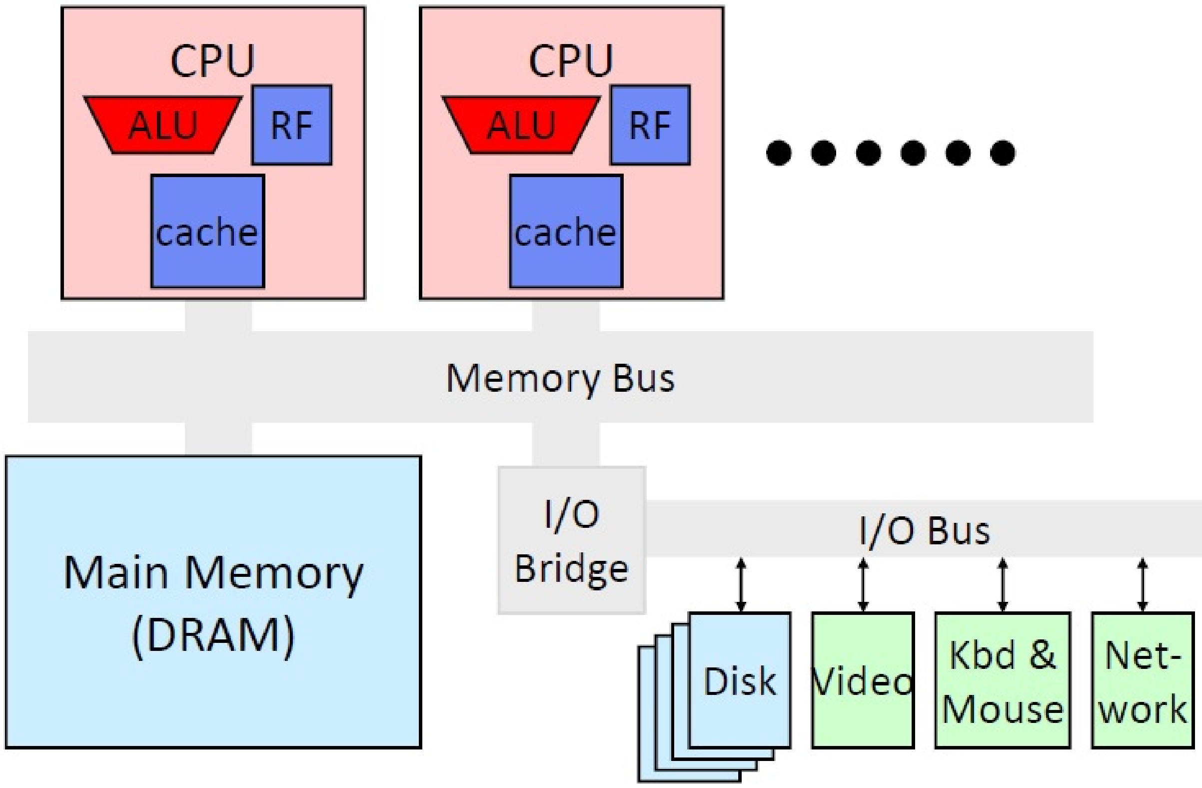

Figure 11.1: The components of the computer and how they communicate with each other.

11.2 Bus Interconnection

The **bus** was the dominant means of computer system component interconnection for decades. For general-purpose computers, it has gradually given way to various point-to-point interconnection structures, which now dominate computer system design. However, bus structures are still commonly used for embedded systems, particularly microcontrollers.

A **bus is a communication pathway connecting two or more devices**. A key characteristic of a bus is that *it is a shared transmission medium*. Multiple devices connect to the bus, and a signal transmitted by any one device is available for reception by all other devices attached to the bus.

Typically, a bus consists of **multiple communication pathways**, or **lines**. Each line is capable of transmitting signals representing binary **1** and binary **0**. Over time, a sequence of binary digits can be transmitted across a single line. Taken together, several lines of a bus can be used to transmit binary digits simultaneously (in parallel). For example, an 8-bit unit of data can be transmitted over eight bus lines.

Computer systems contain a number of different buses that provide pathways between components at various levels of the computer system hierarchy. A bus that connects major computer components (processor, memory, I/O) is called a **system bus**. The most common computer interconnection structures are based on the use of one or more system buses.

There are many different bus designs, on any bus the lines can be classified into three functional groups (Figure 11.2): *data*, *address*, and *control lines*. In addition, there may be *power distribution lines* that supply power to the attached modules.

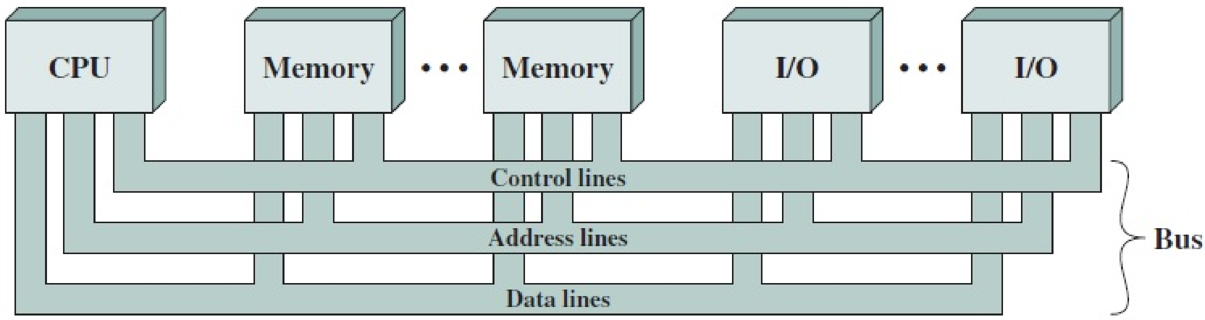

Figure 11.2: Bus Interconnection Scheme.

The **data lines** provide a path for moving data among system modules. These lines, collectively, are called the data bus. The data bus may consist of 32, 64, 128, or even more separate lines, the number of lines being referred to as the width of the data bus.

The **address lines** are used to designate the source or destination of the data on the data bus. For example, if the processor wishes to read a word (8, 16, or 32 bits) of data from memory, it puts the address of the desired word on the address lines. Clearly, the width of the address bus determines the maximum possible memory capacity of the system.

The **control lines** are used to control the access to and the use of the data and address lines. Because the data and address lines are shared by all components, there must be a means of controlling their use. Control signals transmit both command and timing information among system modules. Timing signals indicate the validity of data and address information. Command signals specify operations to be performed. Typical control lines include:

- **Memory write**: causes data on the bus to be written into the addressed location.
- **Memory read**: causes data from the addressed location to be placed on the bus.
- **I/O write**: causes data on the bus to be output to the addressed I/O port.
- **I/O read**: causes data from the addressed I/O port to be placed on the bus.
- **Transfer ACK**: indicates that data have been accepted from or placed on the bus.
- **Bus request**: indicates that a module needs to gain control of the bus.
- **Bus grant**: indicates that a requesting module has been granted control of the bus.
- **Interrupt request**: indicates that an interrupt is pending.
- **Interrupt ACK**: acknowledges that the pending interrupt has been recognized.
- **Clock**: is used to synchronize operations.
- **Reset**: initializes all modules.

The operation of the bus is as follows:

- If one module wishes to send data to another, it must do two things: (1) obtain the use of the bus, and (2) transfer data via the bus.
- If one module wishes to request data from another module, it must (1) obtain the use of the bus, and (2) transfer a request to the other module over the appropriate control and address lines. It must then wait for that second module to send the data.

11.3 Point-to-Point Interconnect

The shared bus architecture was the standard approach to interconnection between the processor and other components (memory, I/O, and so on) for decades. But contemporary systems increasingly rely on point-to-point interconnection rather than shared buses.

The principal reason driving the change from bus to point-to-point interconnect was **the electrical constraints encountered with increasing the frequency of wide synchronous buses**. At higher and higher data rates, it becomes increasingly difficult to perform the synchronization and arbitration functions in a timely fashion. Further, with the advent of multicore chips, with multiple processors and significant memory on a single chip, it was found that the use of a conventional shared bus on the same chip magnified the difficulties of increasing bus data rate and reducing bus latency to keep up with the processors. Compared to the shared bus, the point-to point interconnect has *lower latency*, *higher data rate*, and *better scalability*.

In this section, we look at an important and representative example of the point-to-point interconnect approach: Intel's **QuickPath Interconnect (QPI)**, which was introduced in 2008.

The following are significant characteristics of QPI and other point-to-point interconnect schemes:

Figure 11.3 illustrates a typical use of QPI on a multicore computer. The QPI links (indicated by the green arrow pairs in the figure) form a switching fabric that enables data to move throughout the network. Direct QPI connections can be established between each pair of core processors. If core A in Figure 11.3 needs to access the memory controller in core B, it sends its request through either cores C or D, which must in turn forward that request on to the memory controller in core B. Similarly, larger systems with eight or more processors can be built using processors with three links and routing traffic through intermediate processors.

In addition, QPI is used to connect to an I/O module, called an I/O hub (**IOH**). The IOH acts as a switch directing traffic to and from I/O devices. A core also links to a main memory module (typically the memory uses dynamic access random memory (DRAM) technology) using a dedicated memory bus.

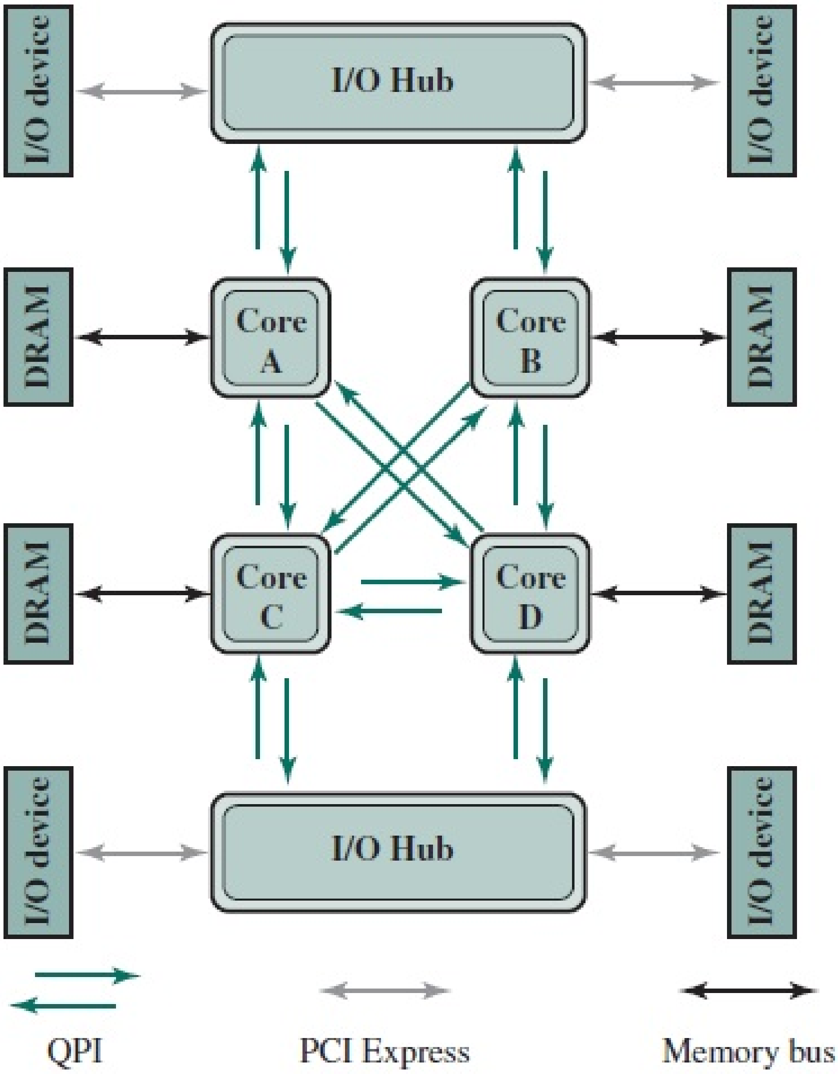

Figure 11.3: Multicore Configuration Using QPI.