Unit 5 Interconnection Structures

Structure:

5.1 Introduction

Objectives

5.2 Types of exchange of information

Modules of a System

Different types of transfers

5.3 Types of Buses

5.4 Elements of Bus Design

Bus Types

Method of arbitration

Bus Timing

Bus width

Bus Speed

5.5 Bus Structure

Single Bus System

Two Bus Organization

The Bus Standard

5.6 Summary

5.7 Terminal Questions

5.8 Answers

5.1 Introduction

Different types of exchanges are required for communication in a computer. Figure 5.1 shows the Memory Module, I/O Module and CPU Module and also indicates the major forms of inputs and outputs of these modules.

Objectives:

By the end of Unit 5, you should be able to:

1. Discuss the different means of data transfer.
2. Define BUS, and explain its structure.
3. Explain the elements of Bus design.
4. Explain with neat sketches single and two bus structure.

5.2 Types of exchange of information

A computer consists of three basic types of modules (processor, memory, I/O) that communicate with each other. Thus, there must be paths for connecting the modules. The collection of paths is called the **"interconnection structure"**. The design of this structure will depend on the exchanges that must be made between modules.

Modules of a system

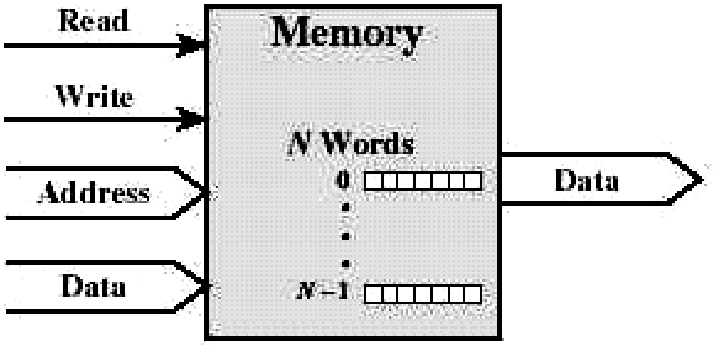

Figure 5.1: a) Memory module

Memory module consists of N words of equal length. Each word is assigned a unique numerical value (0, 1, . . ., N-1). Figure 5.1(a) shows the possible inputs and output of a memory module. A word of data can be read from or written into the memory depending on whether the control signal is Memory

Read (MR) or Memory Write (MW). The location of the memory is provided by the input called as Address.

Example: List the inputs and outputs that are necessary to write the data to the memory.

Solution:

The inputs that are required are

1. Control signal which is memory write,
2. Address that gives the location of the memory where the data is to be written and the data itself.

There is no output required for this particular task.

The sequence of operations that take place:

1. The position of the memory is located using the input address.
2. Then it sees the input control signal is MW and then the input which is data is placed in the location of the memory.

I/O Module

I/O is functionally similar to memory. The possible inputs and outputs for a typical I/O module is as shown in figure 5.1(b).

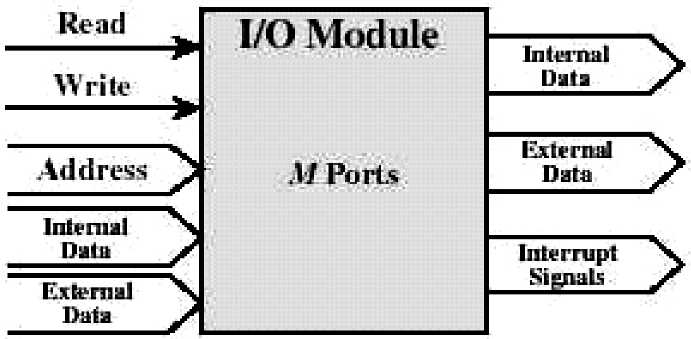

Figure 5.1: b) I/O Module

There are two operations **read** and **write**. I/O module may control more than one external device. We usually refer to each of the interface of the external device as a ***port***. Each port is given a unique address as 0, 1, ………, M-1. Also there are external data paths for the input and output of data with an external device. Also an I/O module may be able to send interrupt signals to the CPU.

CPU Module

The possible inputs and outputs for a typical CPU module is as shown in figure 5.1(c). The CPU reads in the instructions and data and writes out the data after processing. It uses control signals to control the overall operation of the system. It also receives the interrupt signals and appropriate actions to be taken in outputs data as well as control signals.

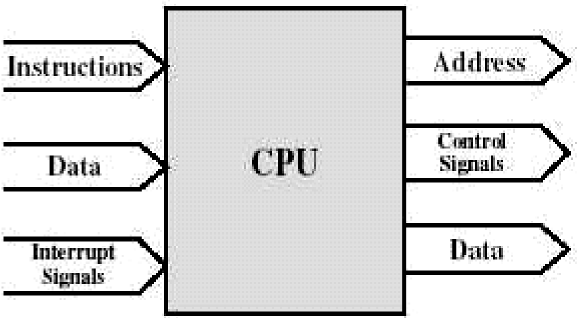

Figure 5.1: c) CPU Module

Different types of transfers

There are two types of transfers that are classified; one depending on the devices that are to be interconnected and the other depending on whether it carries information one bit or several bits.

Types of transfers between different modules

Following are the possible ways of transfer of information between the three modules of the system:

- **Memory to Processor (CPU):** The processor reads an instruction or a unit of data from memory.
- **CPU to Memory:** The processor writes a unit of data to memory.
- **I/O to CPU:** The processor reads data from an I/O device via an I/O module.
- **CPU to I/O:** The processor sends data to the I/O device.
- **I/O to or from memory:** For these two cases, an I/O module is allowed to exchange data directly with memory, without going through the processor, using **Direct Memory Access (DMA).**

Serial & Parallel transfer

A bus consists of multiple communication lines (or pathways). Each line is capable of transmitting signals representing binary 1 or binary 0. Bits are transmitted in Serial & Parallel. When only one bit is carried at a time it requires only a single wire as shown in figure 5.2. When we consider many bits to be transmitted at a time it requires many wires as shown in figure 5.3. These many wires together constitute a bus and the mode of transmission is termed as parallel transmission.

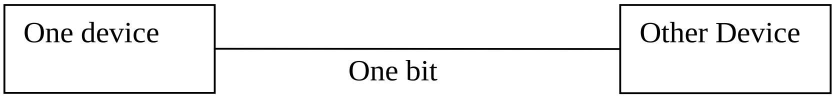

Figure 5.2: Serial transmission - one bit at a time

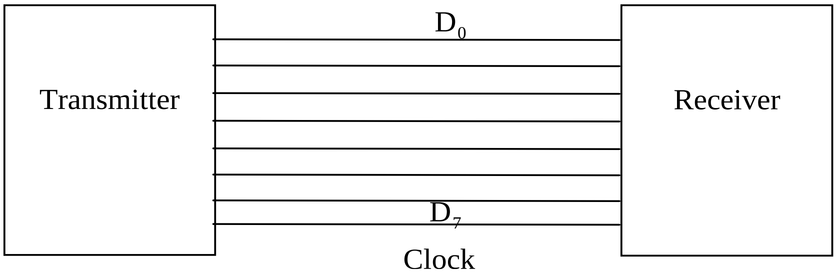

Figure 5.3: Parallel transmission - several (8 bits here) bits at a time

Table gives the Comparison of serial & parallel transmission

Factor	***Serial mode***	***Parallel mode***
Cost	Less costly (only one wire)	More costly (many wires)
Speed	Low (only 1 bit at a time)	High (more bits at a time)
Throughput	Low	High

5.3 Types of Buses

As discussed earlier in unit 1, a system bus consists of a ***data bus*****, a *****memory address bus*** **and a** ***control bus****.* The interconnection between the modules of a system is as shown in figure 5.4

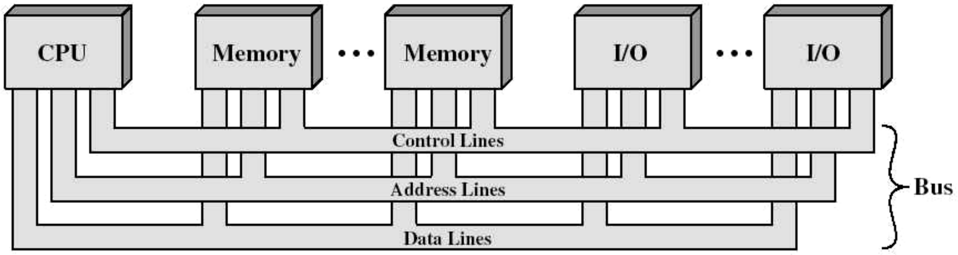

Figure 5.4: Bus Interconnection scheme

Data Bus: A bus, which carries a word to or from memory, is called ***data bus***. Its width is equal to the word length of the memory. Also, it provides a means for moving data between the different modules of a system. The data bus usually consists of 8, 16 or 32 separate lines. The number of lines implies the data bus

Address bus: A bus that is used to carry the address of the data in the memory and its width is equal to the number of bits in the **Memory Address Register (MAR)** of the memory.

Example: If a computer memory has 64K, 32-bit words, then the data bus will be 32-bits wide and the address bus will be 16-bits wide.

Control bus: A bus that is used to control the access carries the control signals between the various units of the computer. The processor has to send commands READ and WRITE to the memory which requires single wire. A START command is necessary for the I/O units. All these signals are carried by the control bus.

Types of Control Lines

- **Memory Write:** Causes data on the bus (data bus) to be written into the addressed location.
- **Memory Read:** Causes data from the addressed location to be placed on the bus (data bus).
- **I/O Write:** Causes data on the data 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 (data bus).
- **Transfer ACK:** Indicates that data has 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:** Used to synchronize operations.
- **Reset:** Initializes all modules.

5.4 Elements of Bus Design

Bus Types: Bus lines can be separated into two types.

- **Dedicated:** Permanently assigned to either one function or to a physical subset of components.
  - ♦ **Functional dedication**: Bus has a specific function.

    Example: Three busses identified for carrying address, data, and control signals as seen earlier. They are **Address Bus, Data Bus**, and **Control Bus**.
  - ♦ **Physical dedication**: Refers to the use of multiple buses, each of which connects only a subset of components using the bus.

    Example: I/O buses are used only to interconnect all I/O modules. And this bus is then connected to the main bus through some type of an I/O adapter module.

Advantage of Physical dedication:

It offers high throughput because there is less bus contention.

Disadvantage of Physical dedication:

Increased size and cost of the system

- **Multiplexed:** These are also referred to as **non-dedicated**. Same bus may be used for various functions. The method of using the same bus for multiple purposes is known as **Time Multiplexing**.

Example: Discuss the steps of actions that are to be performed so that address and data information may be transmitted over the same set of lines.

Solution: Assume an additional control signal called address line activation line or **Address Line Enable (ALE)** line is used.

List of operations are as follows:

1. At the beginning of the data transfer the address is placed on the bus with the ALE line activated.

2. Each module is given sufficient period of time to copy the address and determine if it is the addressed module.
3. The address is then removed from the bus, and then same bus connections are used for subsequent read and write data transfer with the ALE signal deactivated.

Advantages: Multiplexing uses fewer lines, which saves space and cost.

Disadvantages of multiplexing:

1. More complex circuitry required in each module.
2. There is potential reduction in the performance as certain events that share the same bus cannot take place in parallel.

Method of Arbitration:

- **Centralized:** A single hardware device (referred to as **bus controller** or **arbiter**) is responsible for allocating time on the bus to various components.
- **Distributed:** In a distributed scheme, there is no central controller. Each module contains access control logic and the modules act together to share the bus.

In both methods, the purpose is to designate one device (either the processor or an I/O module) as master. The master may then initiate a data transfer with some other device which acts as slave for this particular exchange.

Bus Timing

- **Synchronous:** The occurrence of events on the bus is determined by a clock. The bus includes a clock line. A single 1-0 transmission on clock signal is referred to as 1 clock cycle" or "bus cycle", and defines a time slot. Most events occupy a single clock cycle, but some requires more cycles.

- **Asynchronous:** The occurrence of one event on a bus follows and depends on the occurrence of a previous event.

  Synchronous timing is simpler to implement and test; however it is less flexible.

  With asynchronous timing, a mixture of slow and fast devices can share a bus.

Bus Width:

- **Bus Width of Address Lines:** Number of memory units that can be addressed.
- **Bus Width of Data Lines:** Size of memory units that can be addressed. (8, 12, 16, 32, 64 bits)

  Ask and give examples: How many memory addresses with 24 bits, 32 bits, etc.

Bus Speed:

One of the important attribute of busses is its speed. The speed of the bus refers to how fast you can change the data on the bus, and still have devices to be able to read the values correctly. Bus speed can limit how fast a CPU can communicate with memory. The size of a bus can also limit the speed too.

Example: The speed can be measured in say, MHz that is up to 10^6 changes per second.

Data Transfer Type:

- **Read:** (Slave to Master)
- **Write:** (Master to Slave)
- **Read-Modify-Write:** A read followed immediately by a write to the same address. Usually indivisible operation to prevent any access to data by other potential bus masters.

- **Read-After-Write:** Indivisible operation consisting of a write followed immediately by a read of the same address. Generally for checking purposes.
- **Block:** In this case, one address cycle is followed by n data cycles. The first data item is transferred to/from the specified address, the remaining data items are transferred to/from subsequent addresses.

5.5 Bus Structure

If a large number of devices are connected to the bus, the performance will suffer. There are two main causes:

1. In general, the more devices attached to the bus, the greater is the bus length, and hence the greater is the propagation delay.
2. The bus may become a bottleneck as the aggregate data transfer demand approaches the capacity of the bus.

   To overcome these problems, in most of the modern computers there are multiple buses.

Single Bus System

In this type of inter-connection, the three units share a single bus. Hence the information can be transferred only between two units at a time. Here the I/O units use the same memory address space. This simplifies programming of I/O units as no special I/O instructions are needed. This is one of advantages of single bus organization.

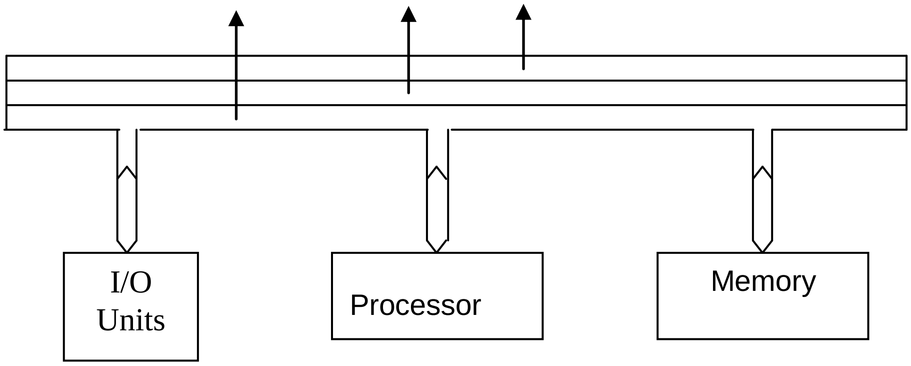

Figure 5.5: Single-Bus Organization

The transfer of information over a bus cannot be done at a speed comparable to the operating speed of all the devices connected to the bus. Some electromechanical devices such as keyboards and printers are very slow whereas disks and tapes are considerably faster. Main memory and processors operate at electronic speeds. Since all the devices must communicate over the bus, it is necessary to smooth out the differences in timings among all the devices.

A common approach is to include ***buffer register*** with the devices to hold the information during transfers. To illustrate this let us take one example. Consider the transfer of an encoded character from the processor to a character printer where it is to be printed. The processor sends the character to the printer output register over the bus. Since buffer is an electronic register this transfer requires relatively little time. Now the printer starts printing. At this time bus and the processor are no longer needed and can be released for other activities. Buffer register is not available for other transfers until the process is completed. Thus buffer register smoothes out the timing differences between the processor, memory and I/O devices. This allows the processor to switch rapidly from one device to another interweaving its processing activity with data transfer involving several I/O devices.

Two Bus Organization

Figure 5.6 shows inter-connection of various computer units through two independent system buses. Here the I/O units are connected to the processor through an I/O bus and the processor is connected to the memory through the memory bus.

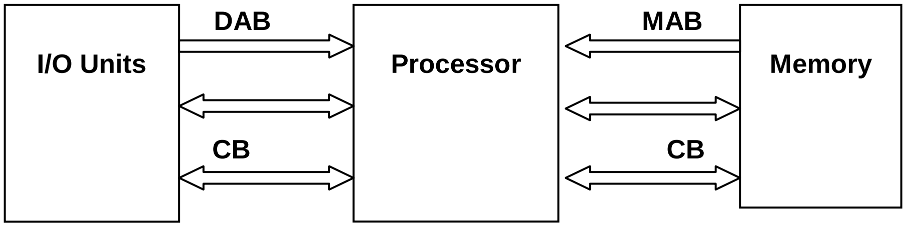

Figure 5.6: Two Bus Organization

The I/O bus consists of device address bus, data bus and a control bus. Device address bus carries the address of the I/O units to be accessed by the processor. The data bus carries a word from the addressed input unit to the processor and from the processor to the addressed output unit. The control bus carries control commands such as START, STOP etc., from the processor to I/O units and it also carries status information of I/O units to the processor. Memory bus also consists of a Memory Address Bus (MAB), data bus and a control bus.

In this type of inter-connection the processor completely supervises the transfer of information to and from the I/O units. All the information is first taken to the processor and from there to the memory. Such a data transfer is called as program controlled transfer.

An alternative two bus structure:

There is one more method to connect processor to the I/O units. Figure 5.7 shows an alternative two bus structure. Here the I/O units are directly connected to the memory.

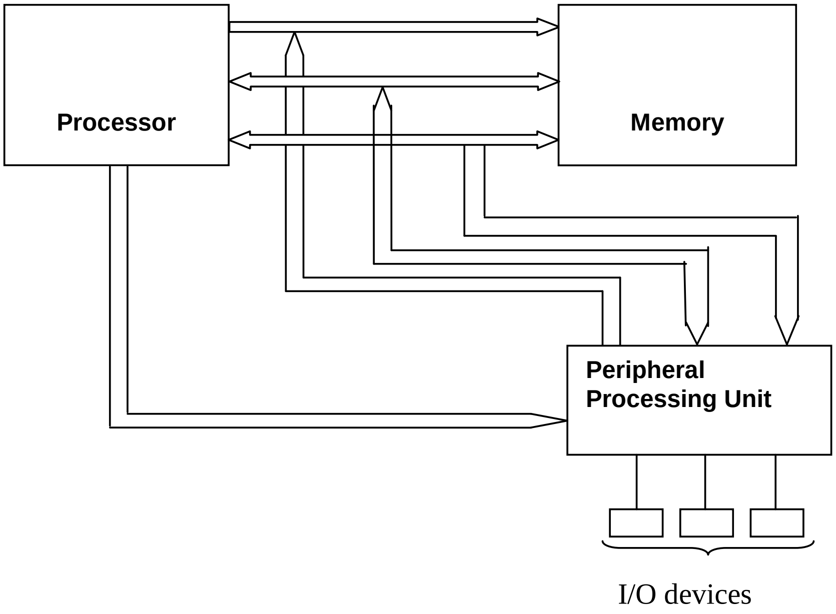

Figure 5.7: Alternative Two-Bus Organization

Here the I/O devices are connected to special interface logic known as ***Direct Memory Access*** **(DMA)** logic or an ***I/O channel***. This is also called as ***Peripheral Processor Unit (PPU).*** The processor issues a READ or WRITE command giving the device address, the address of the memory location where the data read from the input unit is to be stored or from where the data is to be taken to output units, and the number of data words to be transferred. This command is accepted by the PPU, which now takes the responsibility of data transfer.

The Bus Standard

In 1984 IBM was shipping its PC AT model. The CPU, memory and I/O bus all shared a common 8MHz clock. This became the basis for all subsequent clone computers. The term “AT” is a registered trademark of IBM, so this I/O bus became known as the **ISA (Industry Standard Architecture)** bus.

Every currently marketed PC supports some ISA interface slots. The bus and matching adapter cards are simple and cheap. ISA is a 16-bit interface, which means that data can be transferred only two bytes at a time. More importantly, the ISA bus runs at only 8 MHz and it typically requires two or three clock ticks to transfer those two bytes of data. This is not a problem for devices that are inherently slow like the COM port (modem), the printer port, the sound card or the CD-ROM. However, the ISA bus is too slow for high performance disk access and therefore is not acceptable in Servers. It is also too slow for modern Windows display adapters.

The **PCI (Peripheral Connection Interface)** bus was developed by Intel. Although it is mostly known for its CPUs, Intel also has a historical association with Ethernet, multimedia and some disk interfaces. So Intel was unhappy with the VLB concentration on just the video interface and wanted to develop a general purpose bus. The objective was an interface that was fast and inexpensive. It did not have to be simple (advances in chip technology took care of that) and could achieve a low cost by high volume production.

PCI is a 64-bit interface in a 32-bit package. Figuring this out requires a bit of arithmetic. The PCI bus runs at 33 MHz and can transfer 32-bits of data (four bytes) every clock tick. That sounds like a 32-bit bus. However, a clock tick at 33 MHz is 30 nanoseconds and memory only has a speed of 70 nanoseconds. When the CPU fetches data from RAM, it has to wait at least three clock ticks for the data. By transferring data every clock tick, the PCI bus can deliver the same throughout on a 32-bit interface that other parts of the machine deliver through a 64-bit path.

The PCI bus connects at one end to the CPU/memory bus and at the other end to a more traditional I/O bus. The PCI interface chip may support the video adapter, the EIDE disk controller chip and may be two external adapter cards. A desktop machine will have only one PCI chip and so it will add a number of extra ISA only slots. A server may add additional PCI chips and extra server slots will usually be EISA.

While ISA and EISA are exclusively PC interfaces, the PCI bus is now used in Power Macintosh systems and PowerPC machines. It may be attractive for minicomputers and other RISC workstations.

5.6 Summary

There must be paths for connecting the processor, Memory or I/O modules. The collection of paths is called the interconnection structure or the Bus structure. We have introduced different types of transfers between different modules like Memory to processor (CPU), CPU to memory, or I/O to CPU. There are different buses called address, data or control bus depending upon information they carry, that is, address, data, or controlling signals respectively. These buses can be dedicated to one type of information they carry or can be multiplexed; they can be centralized or distributed, or synchronous or asynchronous.

Self Assessment Questions

1. The collection of paths for connecting the modules is called the ____________.
2. The location of the memory is provided by the input called as ____________.
3. A bus which carries a word to or from memory is called ____________.
4. A bus that is used to carry control signals is ____________.
5. The method of using the same bus for multiple purposes is known as ____________.

5.7 Terminal Questions

1. Discuss the different types of Bus.
2. Give the advantages and disadvantages of physical and functional buses.
3. Explain the Single bus structure.
4. Discuss the relative merits and de-merits of single, two and three bus structure.
5. Discuss the methods of arbitration.

5.8 Answers

Self Assessment Questions:

1. interconnection structure
2. address
3. data bus
4. control bus
5. time multiplexing

Terminal Questions:

1. Refer Section 5.2
2. Refer Section 5.3
3. Refer Section 5.4
4. Refer Section 5.4
5. Refer Section 5.3