Input and Output System

Introduction to I/O Module

An I/O module is not simply a set of mechanical connectors that wire a device into the system bus. Rather, the I/O module contains logic for performing a communication function between the peripheral and the bus.

why one does not connect peripherals directly to the system bus. The reasons are as follows:

- There are a wide variety of peripherals with various methods of operation. It would be impractical to incorporate the necessary logic within the processor to control a range of devices.
- The data transfer rate of peripherals is often much slower than that of the memory or processor. Thus, it is impractical to use the high-speed system bus to communicate directly with a peripheral.
- On the other hand, the data transfer rate of some peripherals is faster than that of the memory or processor. Again, the mismatch would lead to inefficiencies if not managed properly.
- Peripherals often use different data formats and word lengths than the computer to which they are attached.

Generic Model of an I/O Module

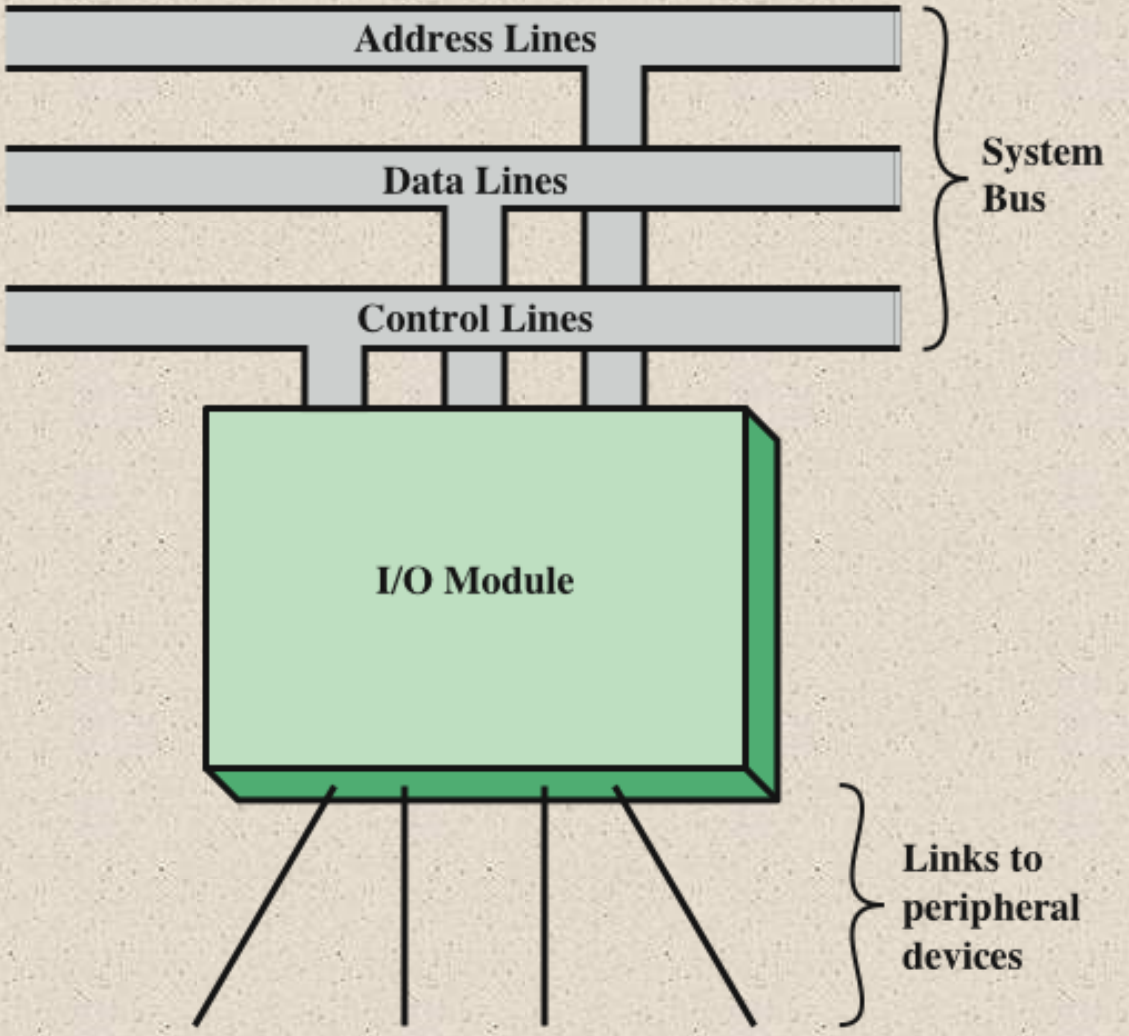


Figure 7.1 Generic Model of an I/O Module

External Devices

- Provide a means of exchanging data between the external environment and the computer
- Attach to the computer by a link to an I/O module
  - The link is used to exchange control, status, and data between the I/O module and the external device
- *peripheral device*
  - An external device connected to an I/O module

- Three categories:
- Human readable
  - Suitable for communicating with the computer user
  - Video display terminals (VDTs), printers
- Machine readable
  - Suitable for communicating with equipment
  - Magnetic disk and tape systems, sensors and actuators
- Communication
  - Suitable for communicating with remote devices such as a terminal, a machine readable device, or another computer

External Device Block Diagram

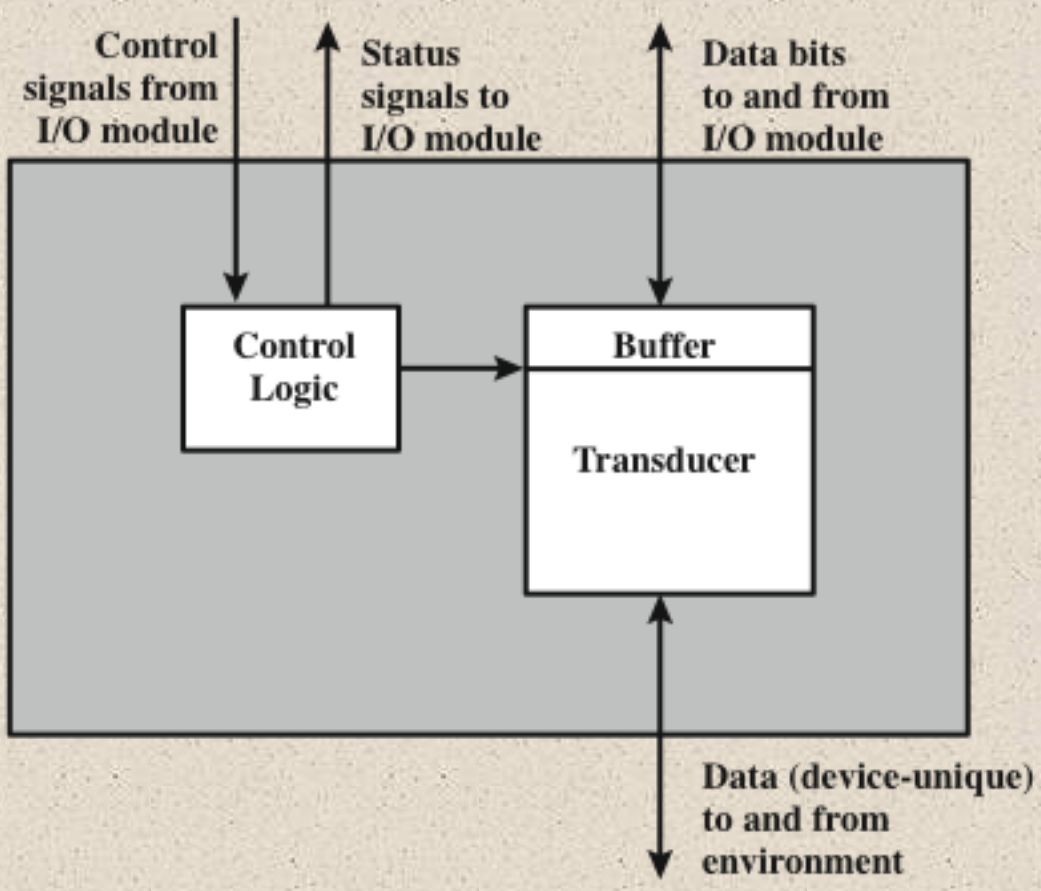


Figure 7.2 Block Diagram of an External Device

Keyboard/Monitor

- Most common means of computer/user interaction
- User provides input through the keyboard
- The monitor displays data provided by the computer

International Reference Alphabet (IRA)

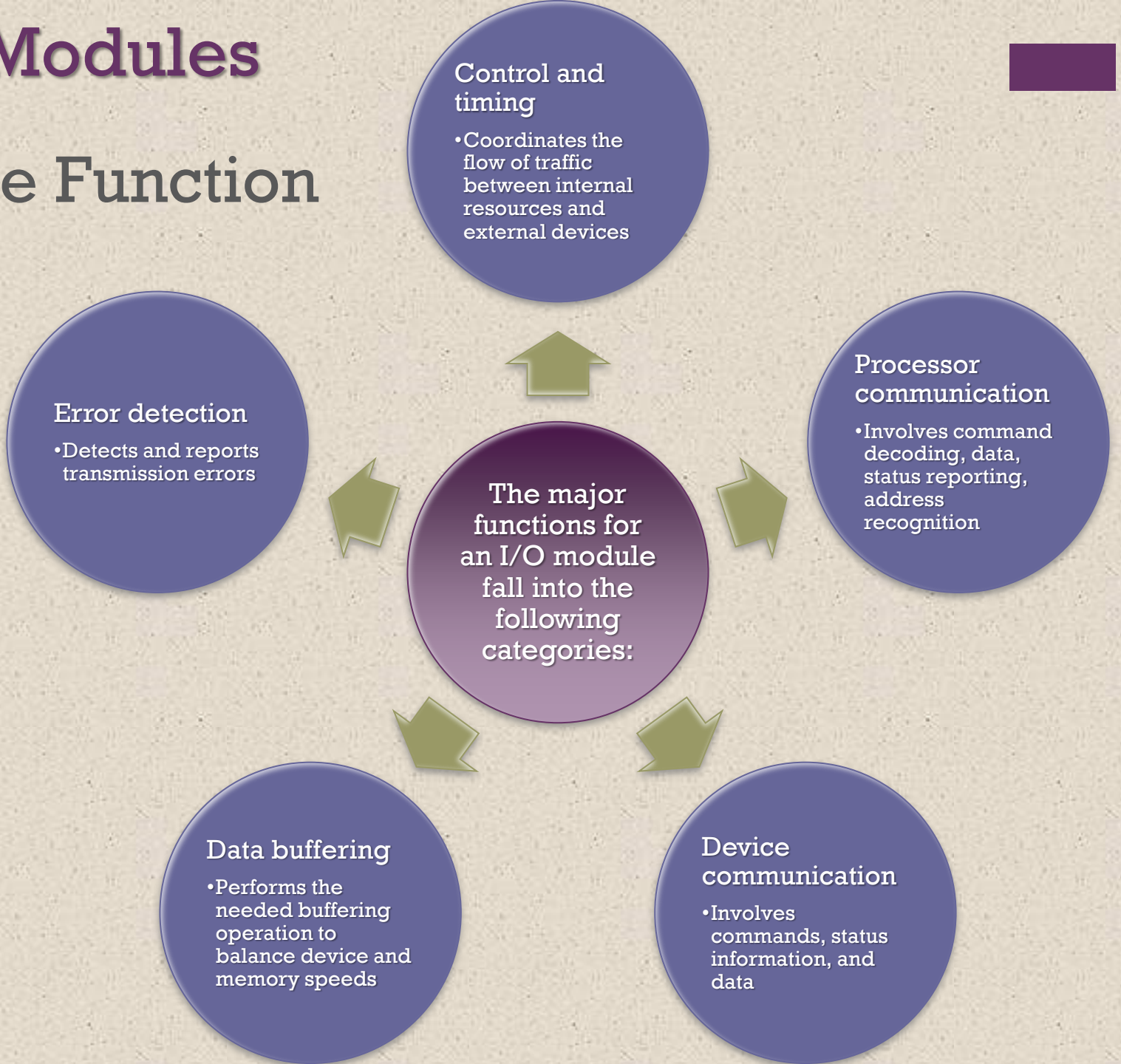
- Basic unit of exchange is the character
  - Associated with each character is a code
  - Each character in this code is represented by a unique 7-bit binary code
    - 128 different characters can be represented
- Characters are of two types:
  - Printable
    - Alphabetic, numeric, and special characters that can be printed on paper or displayed on a screen
  - Control
    - Have to do with controlling the printing or displaying of characters
    - Example is carriage return

Keyboard Codes

- When the user depresses a key it generates an electronic signal that is interpreted by the transducer in the keyboard and translated into the bit pattern of the corresponding IRA code
- This bit pattern is transmitted to the I/O module in the computer
- On output, IRA code characters are transmitted to an external device from the I/O module
- The transducer interprets the code and sends the required electronic signals to the output device either to display the indicated character or perform the requested control function

I/O Modules

Module Function



I/O Module Structure

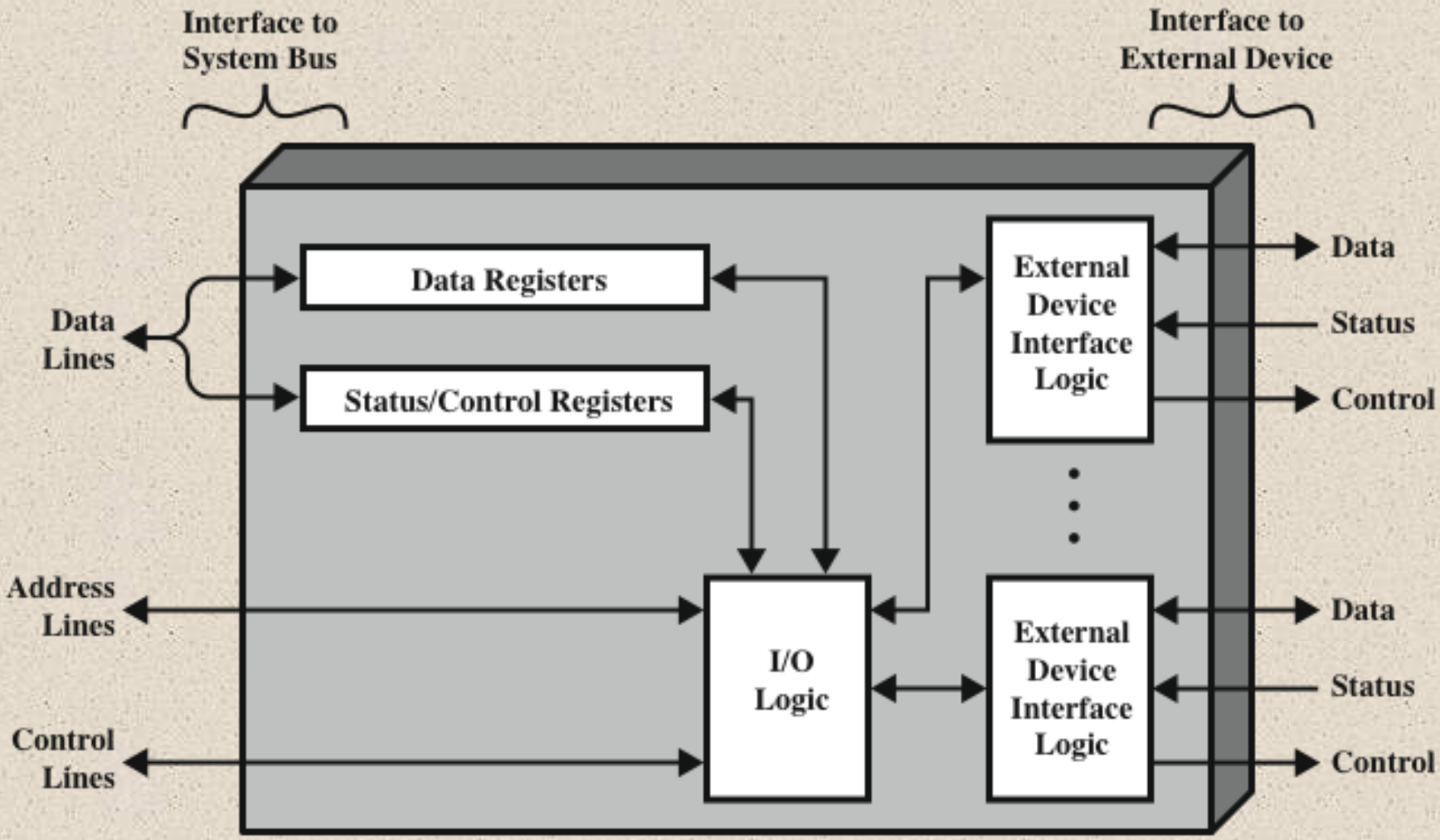


Figure 7.3 Block Diagram of an I/O Module

I/O operations

- Three techniques are possible for I/O operations:
- Programmed I/O
  - Data are exchanged between the processor and the I/O module
  - Processor executes a program that gives it direct control of the I/O operation
  - When the processor issues a command it must wait until the I/O operation is complete
  - If the processor is faster than the I/O module this is wasteful of processor time
- Interrupt-driven I/O
  - Processor issues an I/O command, continues to execute other instructions, and is interrupted by the I/O module when the latter has completed its work
- Direct memory access (DMA)
  - The I/O module and main memory exchange data directly without processor involvement

Table 7.1
I/O Techniques

	No Interrupts	Use of Interrupts
I/O-to-memory transfer through processor	Programmed I/O	Interrupt-driven I/O
Direct I/O-to-memory transfer		Direct memory access (DMA)

Overview of Programmed I/O

- When the processor is executing a program and encounters an instruction related to I/O, It executes that instruction by issuing a command to the appropriate I/O module.
- With programmed I/O, the I/O module will perform the requested action and then set the appropriate bits in I/O status register.

I/O Commands

- There are four types of I/O commands that an I/O module may receive when it is addressed by a processor:

1) Control

- used to activate a peripheral and tell it what to do

2) Test

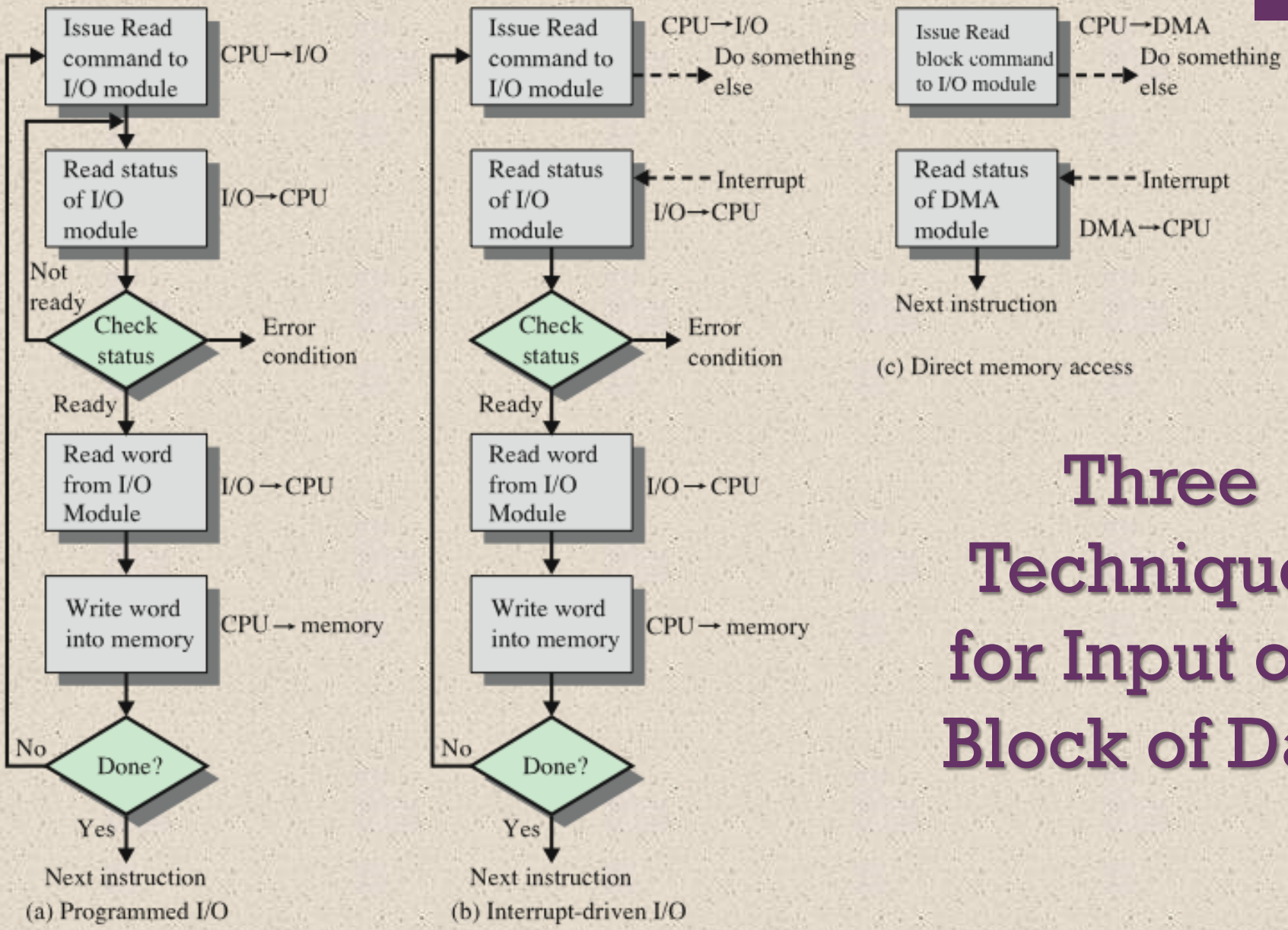
- used to test various status conditions associated with an I/O module and its peripherals

3) Read

- causes the I/O module to obtain an item of data from the peripheral and place it in an internal buffer

4) Write

- causes the I/O module to take an item of data from the data bus and subsequently transmit that data item to the peripheral



Three Techniques for Input of a Block of Data

Figure 7.4 Three Techniques for Input of a Block of Data

I/O Instructions

With programmed I/O there is a close correspondence between the I/O-related instructions that the processor fetches from memory and the I/O commands that the processor issues to an I/O module to execute the instructions

The form of the instruction depends on the way in which external devices are addressed

Each I/O device connected through I/O modules is given a unique identifier or address

When the processor issues an I/O command, the command contains the address of the desired device

Thus each I/O module must interpret the address lines to determine if the command is for itself

Memory-mapped I/O

There is a single address space for memory locations and I/O devices

A single read line and a single write line are needed on the bus

Interrupt-Driven I/O

The problem with programmed I/O is that the processor has to wait a long time for the I/O module to be ready for either reception or transmission of data

An alternative is for the processor to issue an I/O command to a module and then go on to do some other useful work

The I/O module will then interrupt the processor to request service when it is ready to exchange data with the processor

The processor executes the data transfer and resumes its former processing

Design Issues

Two design issues arise in implementing interrupt I/O:

- Because there will be multiple I/O modules how does the processor determine which device issued the interrupt?
- If multiple interrupts have occurred how does the processor decide which one to process?

Drawbacks of Programmed and Interrupt-Driven I/O

- Both forms of I/O suffer from two inherent drawbacks:

  1) The I/O transfer rate is limited by the speed with which the processor can test and service a device

  2) The processor is tied up in managing an I/O transfer; a number of instructions must be executed for each I/O transfer

- When large volumes of data are to be moved a more efficient technique is *direct memory access* (DMA)

Typical DMA Module Diagram

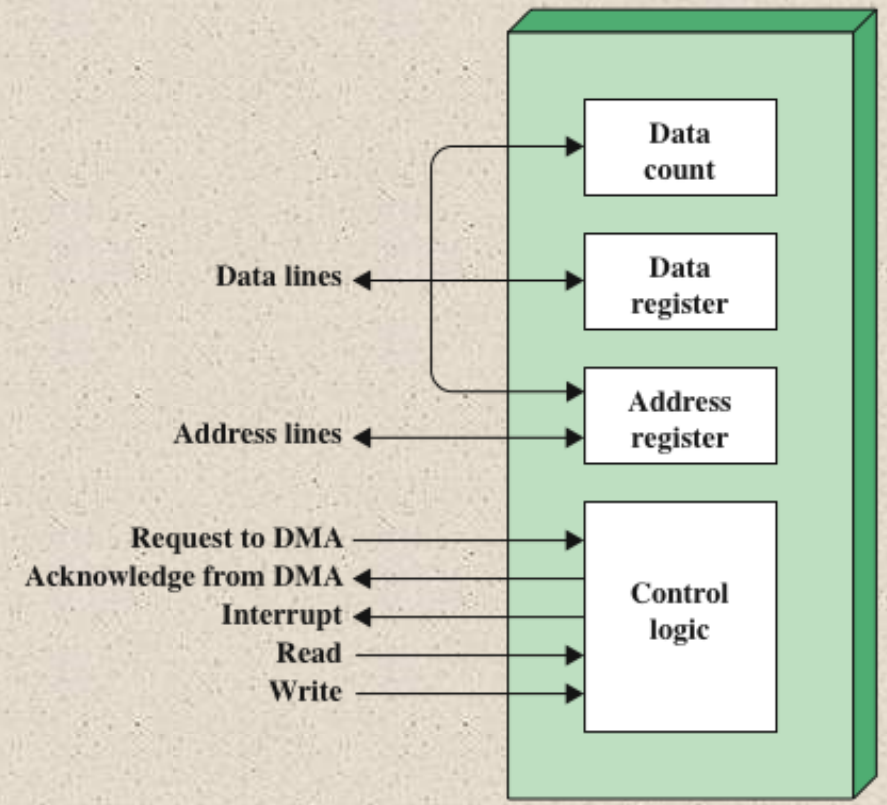


Figure 7.11 Typical DMA Block Diagram

DMA Controller is a hardware device that allows I/O devices to directly access memory with less participation of the processor.

DMA controller needs the same old circuits of an interface to communicate with the CPU and Input/Output devices.

The DMA controller has three registers as follows.

- **Address register –** It contains the address to specify the desired location in memory.
- **Word count register –** It contains the number of words to be transferred.
- **Control register –** It specifies the transfer mode.

The CPU initializes the DMA by sending the given information through the data bus.

1. The starting address of the memory block where the data is available (to read) or where data are to be stored (to write).
2. It also sends word count which is the number of words in the memory block to be read or write.
3. Control to define the mode of transfer such as read or write.
4. A control to begin the DMA transfer.

Memory Mapped I/O

- In memory mapping of I/O devices, the I/O ports are assigned 16-bit address within the memory.
- Here each bus is common thus the same set of instructions is used for memory and I/O devices. Thus, I/O is considered as memory and the same address space is used by both memory and I/O devices. This reduces the addressing capability of the memory.
- In this case, the processor considers the I/O ports as memory locations for the purpose of reading and writing. So, whenever, an address is generated on the address bus then resultantly control signal is generated for memory read.
- In such a case, the processor is not concerned whether the responding data is coming from a memory device or an I/O device. The same is the case with the memory write operation.

I/O Mapped I/O

It is also known as **Isolated I/O mapping** and the reason for the same is that here the address space of memory and I/O are separated from each other. Thus, different read and write instructions are used for I/O and memory.

In this approach, there is a common bus for I/O devices and memory however, individual read and write control lines are used for I/O.

Here the operation takes place in a way that, if the data over which operation is to be performed is to be collected from the I/O devices then address is placed on the address line and I/O read and I/O write control lines will get activated so that data transfer can be performed between the processor and I/O.

For the transfer of data between the processor and I/O devices, only IN and OUT instructions are used in the isolated mapping. The required chip select signals in this case are generated by an individual decoding unit.

Difference between Memory mapped I/O and I/O mapped I/O

Features	Memory Mapped IO	IO Mapped IO
Addressing	IO devices are accessed like any other memory location.	They cannot be accessed like any other m location.
Address Size	They are assigned with 16-bit address values.	They are assigned with 8-bit address valu
Instructions Used	The instruction used are LDA and STA, etc.	The instruction used are IN and OUT.
Cycles	Cycles involved during operation are Memory Read, Memory Write.	Cycles involved during operation are IO IO writes
Registers Communicating	Any register can communicate with the IO device	Only Accumulator can communicate wit devices
Space Involved	2^{16} IO ports are possible to be used for interfacing in case of Memory Mapped IO.	Only 256 I/O ports are available for inte case of IO Mapped IO.
Control Signal	No separate control signal required since we have unified memory space in the case of Memory Mapped IO.	Special control signals are used in the cas Mapped IO.