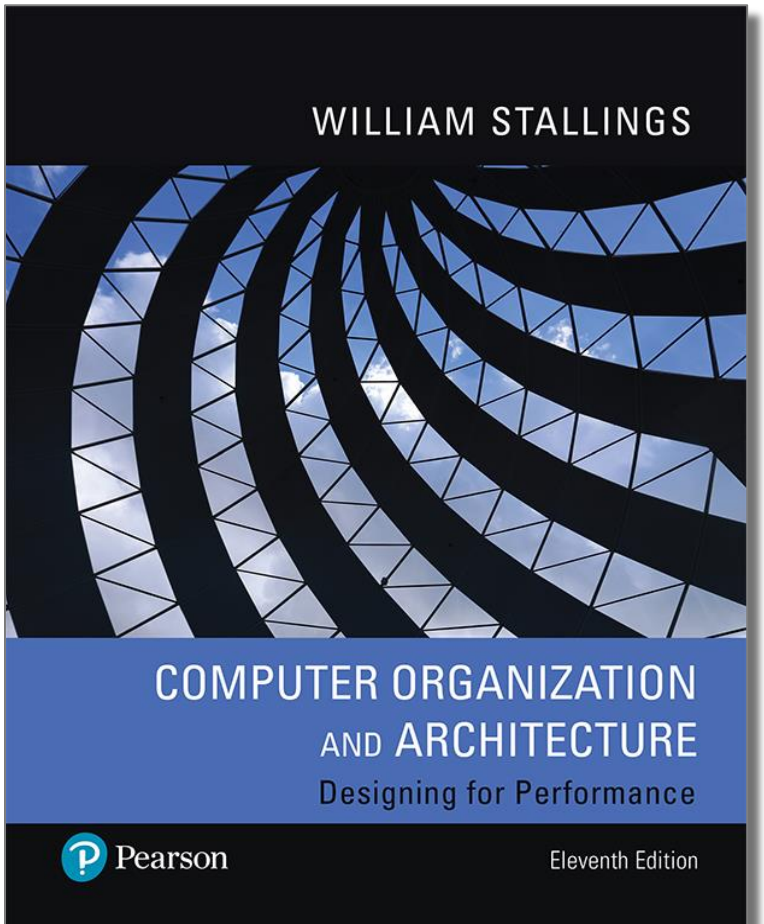
Computer Organization and Architecture
Designing for Performance

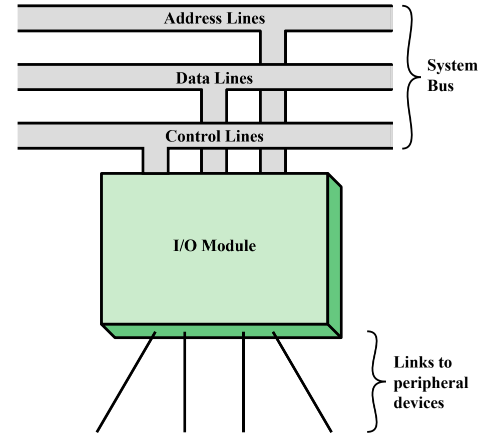
11th Edition

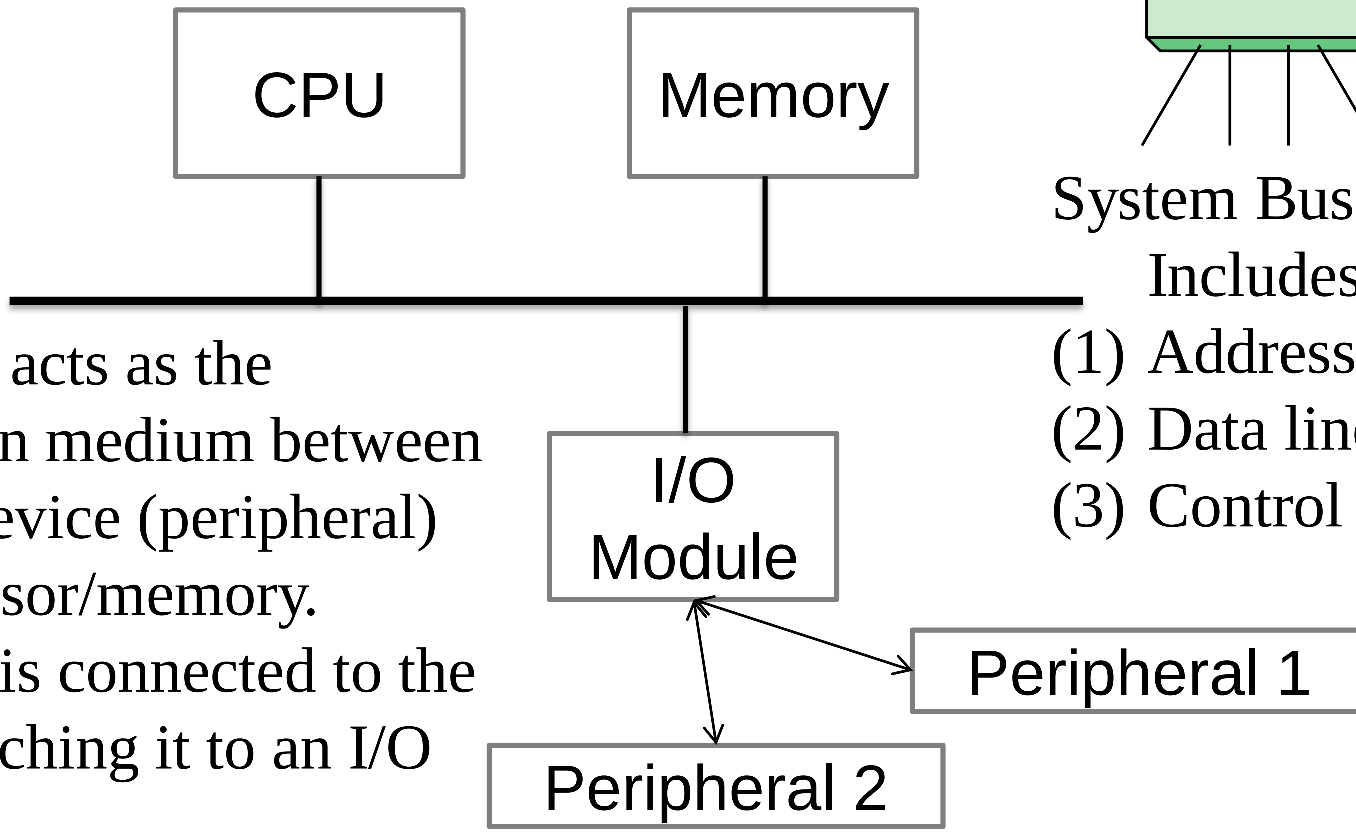


Chapter 8

Input/Output

Input / Output



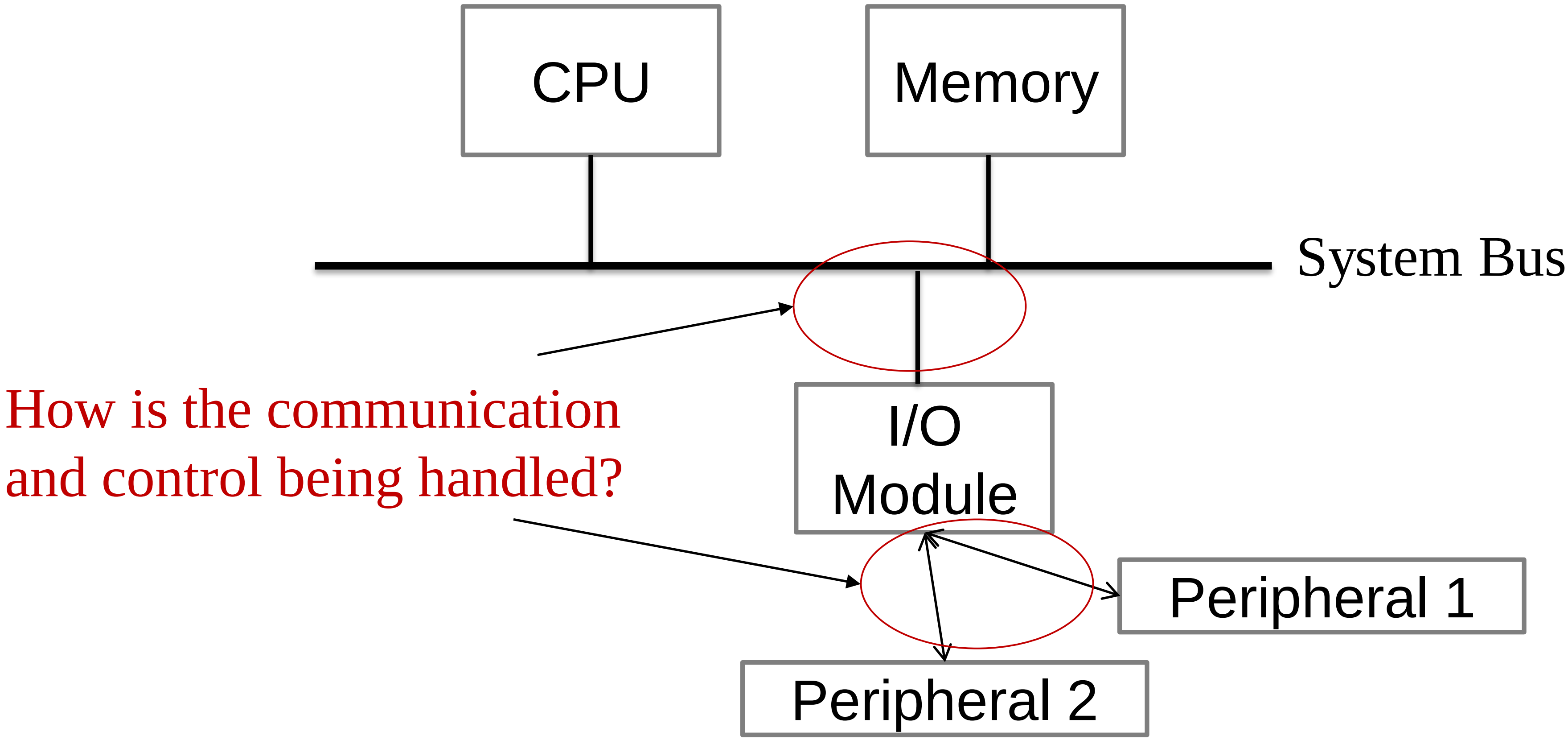


System Bus Includes:

(1) Address lines
(2) Data lines
(3) Control lines

- I/O Modules acts as the communication medium between external I/O device (peripheral) and the processor/memory.
- A peripheral is connected to the system by attaching it to an I/O Module.

Input / Output



External Devices

- Provide a means of exchanging data between the external environment and the computer

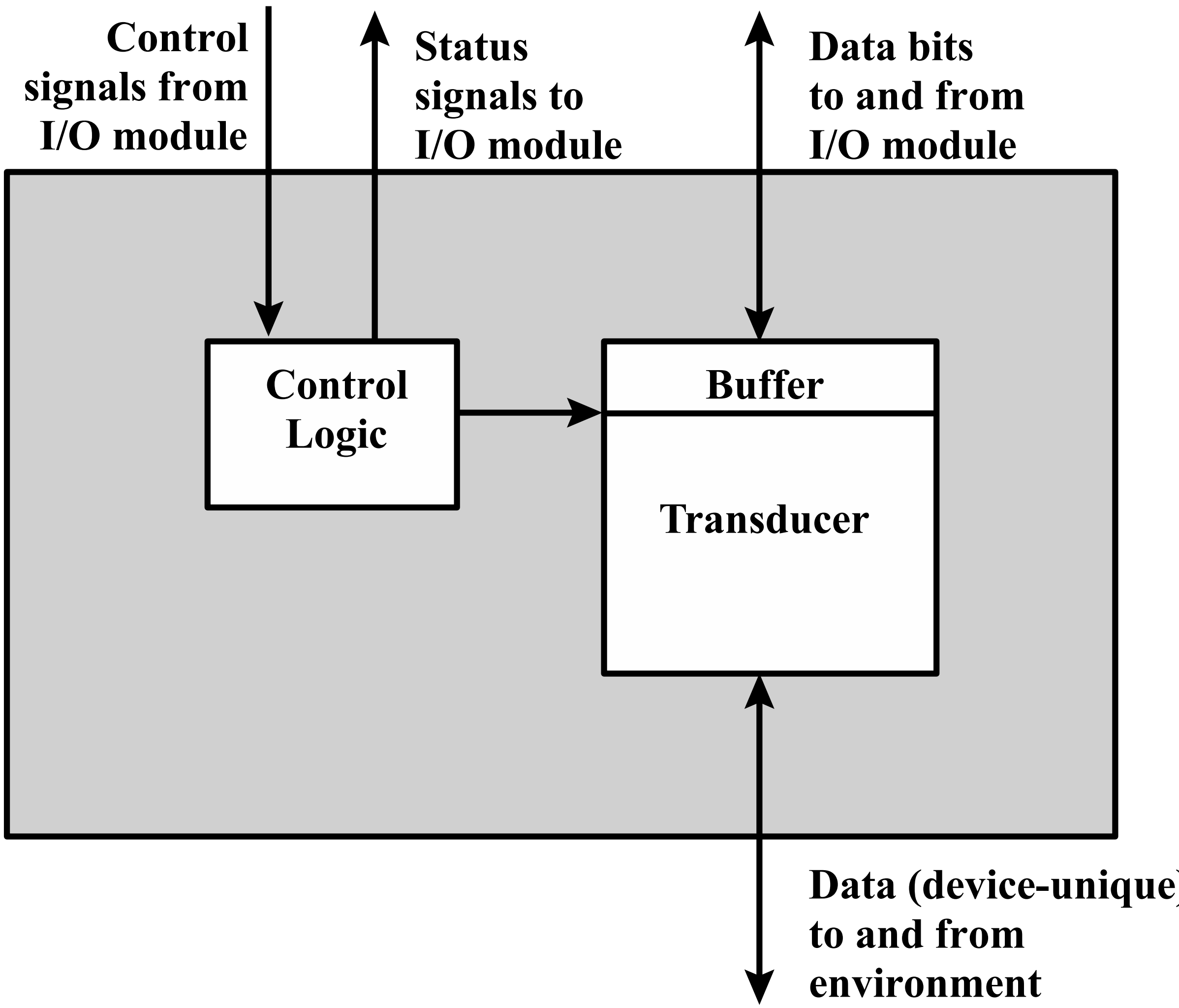
- Attach to the computer by a link to an I/O module
  - The link is used to exchange control, status, and data between the I/O module and the external device

- *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

Figure 8.2
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

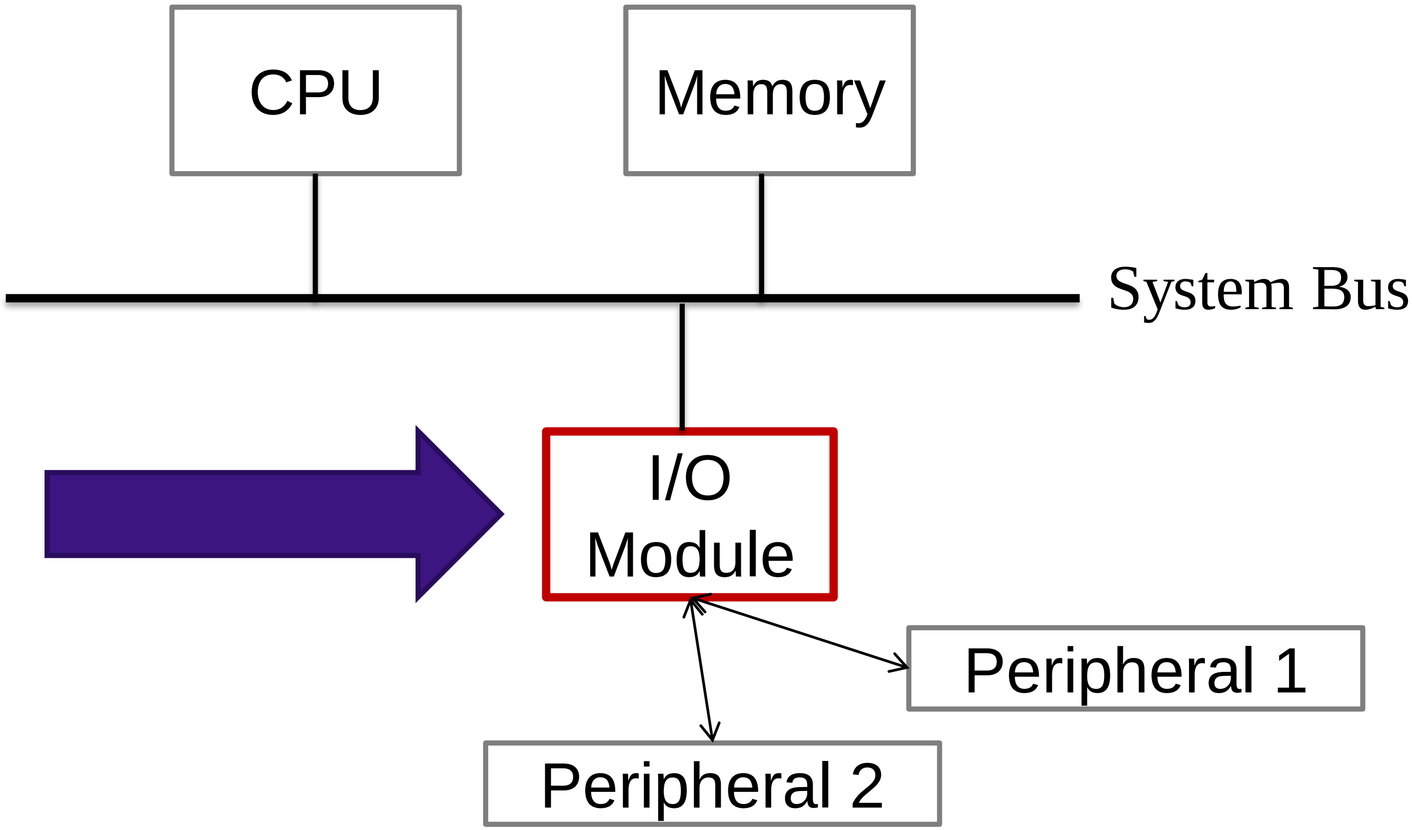
International Reference Alphabet (IRA)

- 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
    - Other control characters are concerned with communications procedures

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

Input / Output



I/O Module Functions

The major functions for an I/O module fall into the following categories:

Control and timing
- Coordinates the flow of traffic between internal resources and external devices

Processor communication
- Involves command decoding, data, status reporting, address recognition

Device communication
- Involves commands, status information, and data

Data buffering
- Performs the needed buffering operation to balance device and memory speeds

Error detection
- Detects and reports transmission errors

I/O Module Functions

- At any point of time, the processor may communicate with one or more external devices in unpredictable patterns, depending on the program's need for I/O.
- The I/O function includes a control and timing requirement, to coordinate the flow of traffic between internal resources (e.g., shared bus) and external devices
- Thus, the I/O function includes a **control and timing requirement**, to coordinate the flow of traffic between internal resources and external devices.

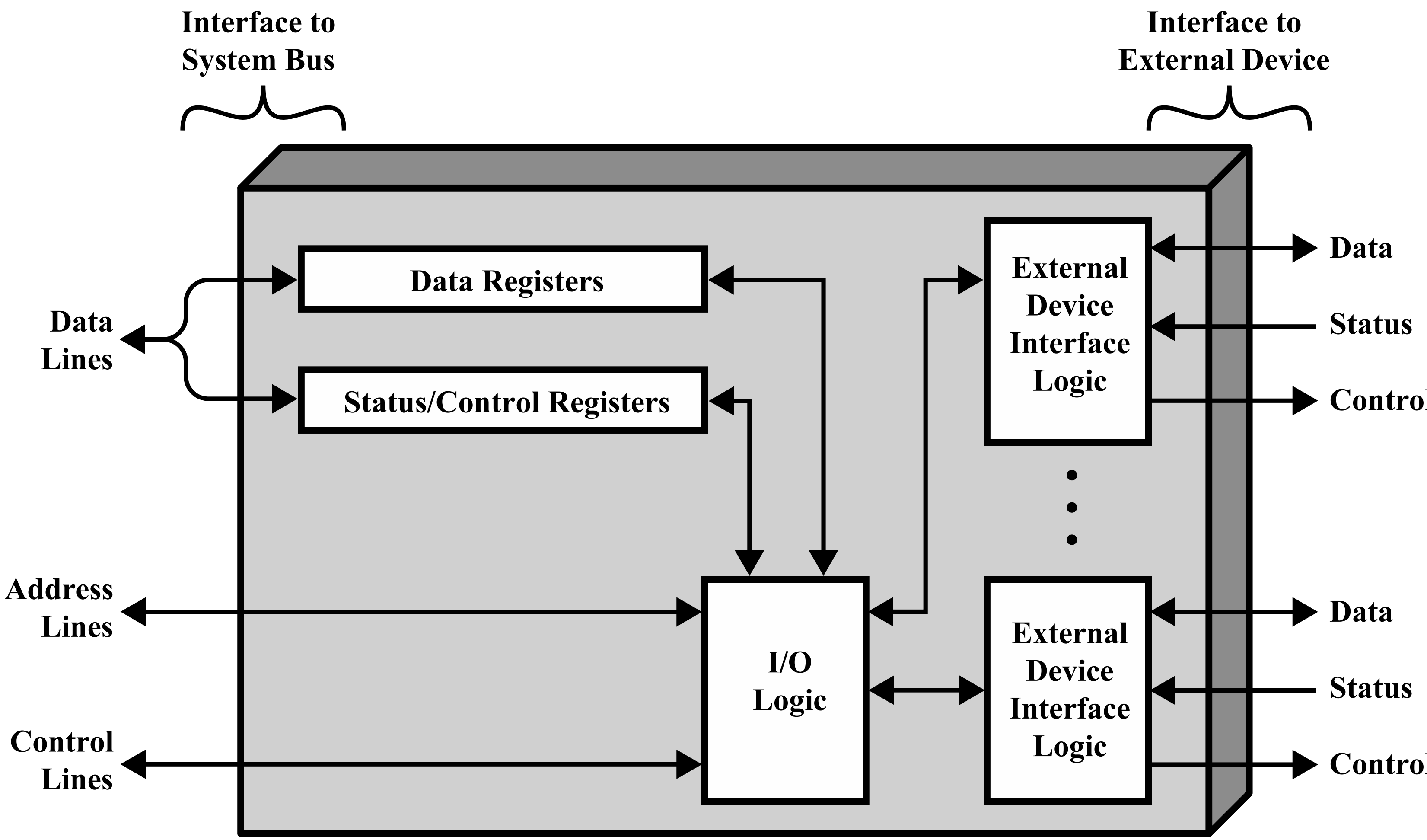
I/O Module Functions

- The I/O module must also communicate with the processor and with the external device. **Processor communication** involves the following:
  - Command decoding: I/O module accepts and understands commands from the processor.
  - Data: data are exchanged between the processor and the I/O module over the data bus.
  - Status reporting: it is important to know the status of each peripheral (busy, ready, etc.)
  - Address recognition: I/O module must recognize one unique address for each peripheral it controls.

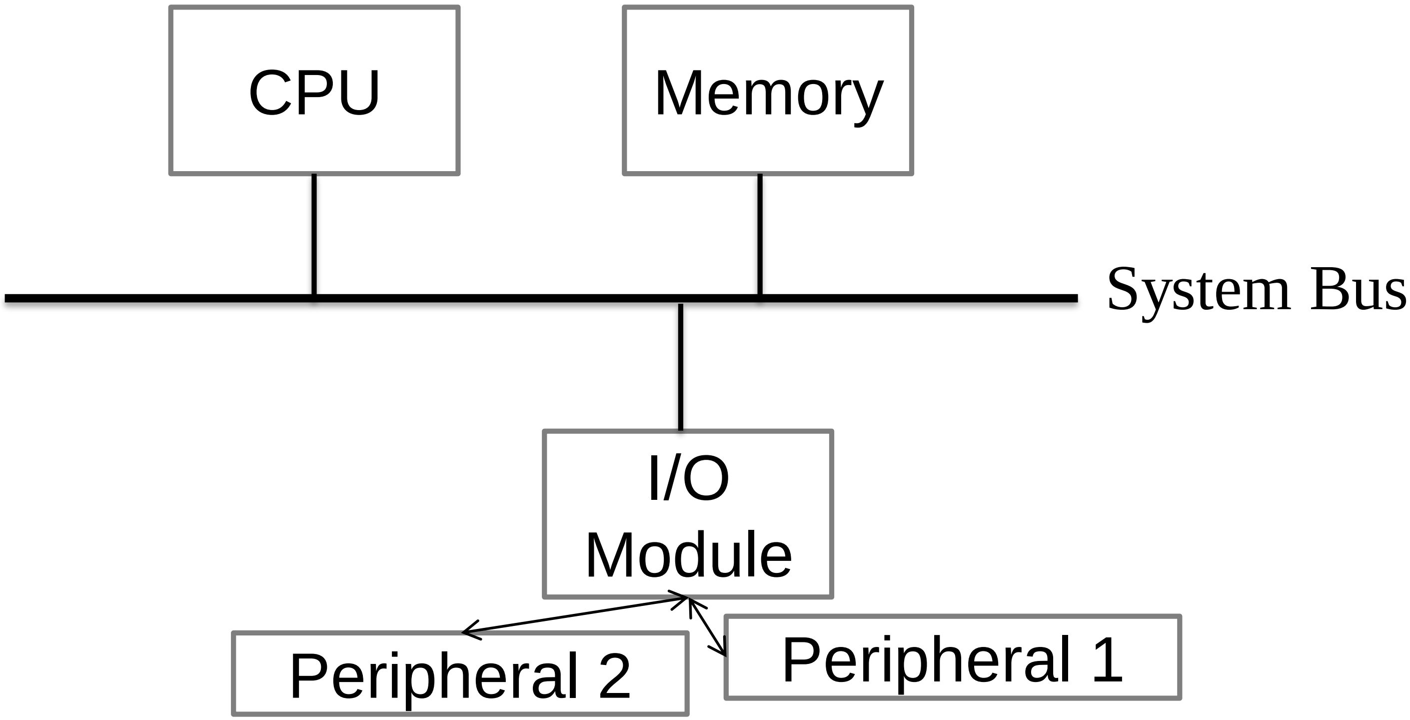
I/O Module Functions

- An essential task of an I/O module is **data buffering**.
- Finally, an I/O module is often responsible for **error detection** and for subsequently reporting errors to the processor

Figure 8.3
Block Diagram of an I/O Module



I/O Operations



A very important question is:

How much work is done by the processor?

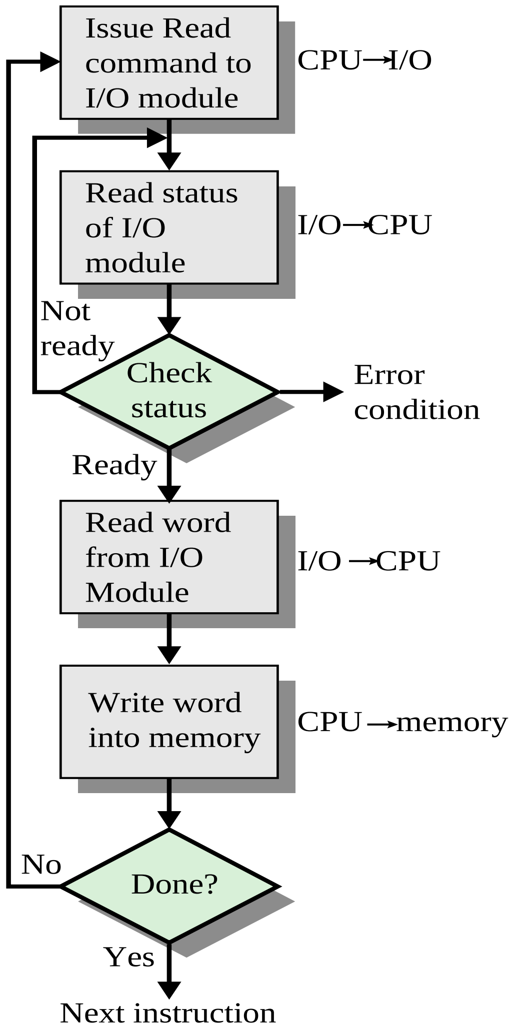
Goal: as less as possible

Programmed I/O

Three techniques are possible for I/O operations:

1. Programmed I/O
2. Interrupt-driven I/O
3. Direct memory access (DMA)

Figure 8.4 - Programmed I/O

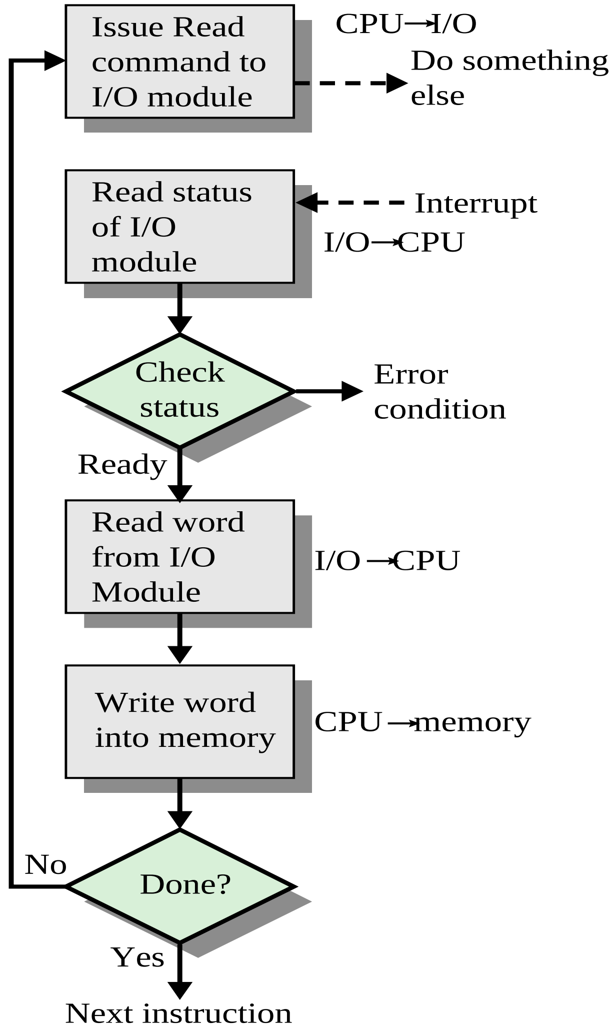


(a) Programmed I/O

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

Figure 8.4 - Interrupt-driven I/O



(b) Interrupt-driven I/O

Interrupt-Driven I/O

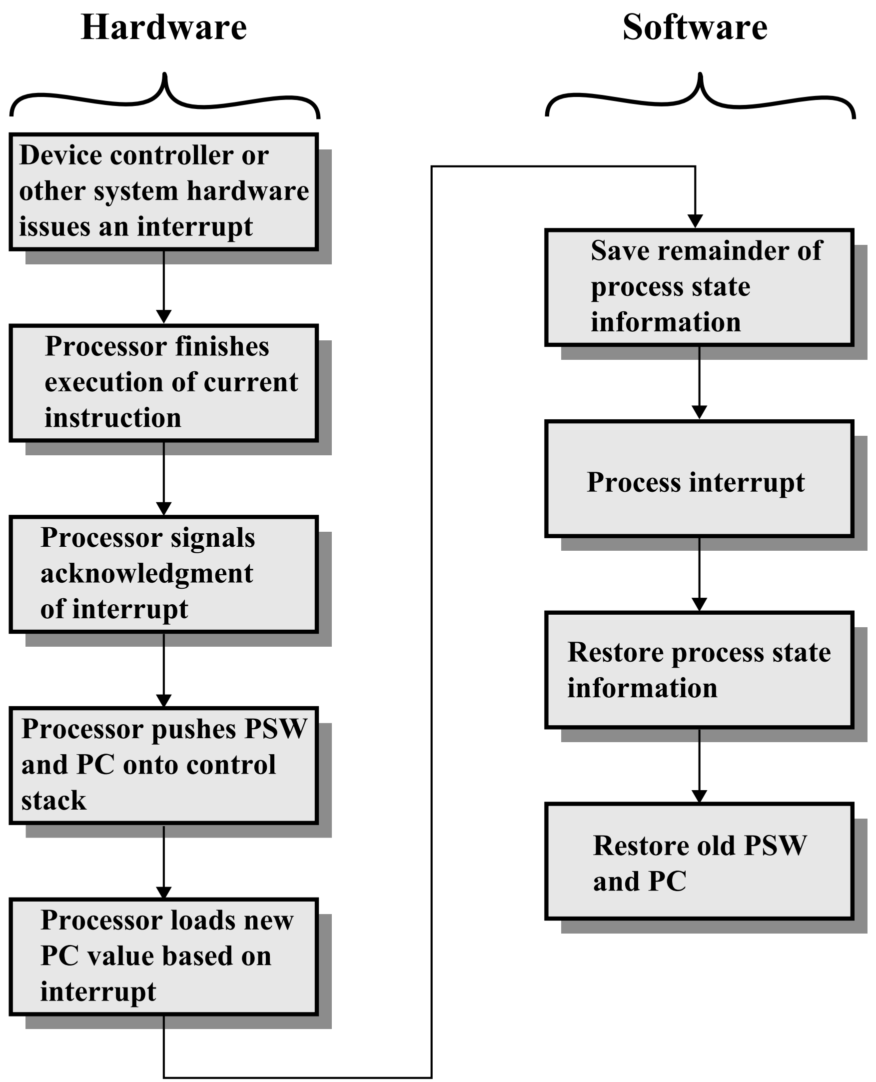
The problem with programmed I/O is that the processor has to wait a long time for the I/O module to be ready for either reception or transmission of data

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

Figure 8.6
Simple Interrupt Processing



Pearson

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?

Device Identification

Four general categories of techniques are in common use:

- **Multiple interrupt lines**
  - Between the processor and the I/O modules
  - Most straightforward approach to the problem
  - Consequently even if multiple lines are used, it is likely that each line will have multiple I/O modules attached to it
- **Software poll**
  - When the processor detects an interrupt it branches to an interrupt-service routine whose job is to poll each I/O module to determine which module caused the interrupt
  - Time consuming
- **Daisy chain (hardware poll, vectored)**
  - The interrupt acknowledge line is daisy chained through the modules
  - Vector – address of the I/O module or some other unique identifier
  - Vectored interrupt – processor uses the vector as a pointer to the appropriate device-service routine, avoiding the need to execute a general interrupt-service routine first
- **Bus arbitration (vectored)**
  - An I/O module must first gain control of the bus before it can raise the interrupt request line
  - When the processor detects the interrupt it responds on the interrupt acknowledge line
  - Then the requesting module places its vector on the data lines

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)

Copyright

This work is protected by United States copyright laws and is provided solely for the use of instructions in teaching their courses and assessing student learning. dissemination or sale of any part of this work (including on the World Wide Web) will destroy the integrity of the work and is not permitted. The work and materials from it should never be made available to students except by instructors using the accompanying text in their classes. All recipients of this work are expected to abide by these restrictions and to honor the intended pedagogical purposes and the needs of other instructors who rely on these materials.