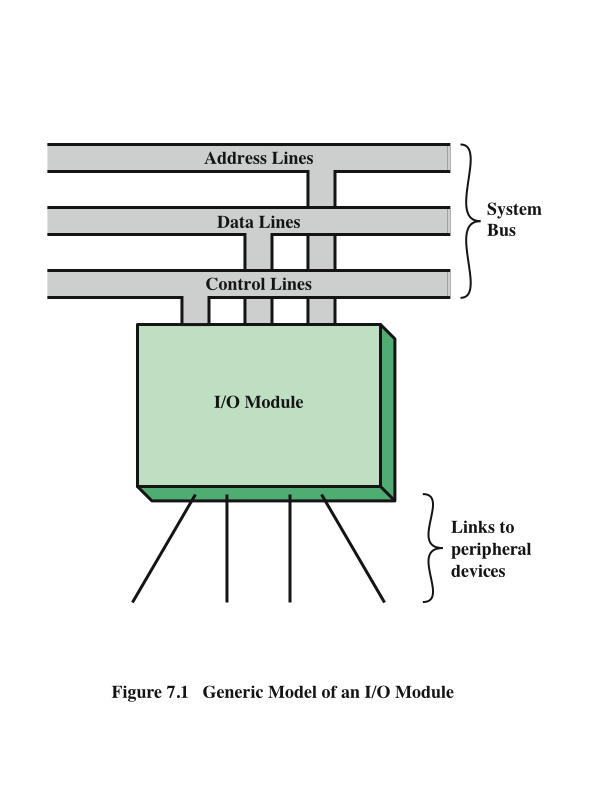
**CHAPTER 7**

**Why are devices not connected to system bus?**Có rất nhiều loại thiết bị ngoại vi với các phương thức hoạt động khác nhau. Sẽ không thực tế khi tích hợp logic cần thiết vào bộ xử lý để điều khiển một loạt các thiết bị. Tốc độ truyền dữ liệu của các thiết bị ngoại vi thường chậm hơn nhiều so với bộ nhớ hoặc bộ xử lý. Vì vậy, việc sử dụng bus hệ thống tốc độ cao để giao tiếp trực tiếp với một thiết bị ngoại vi là không thực tế. Tốc độ truyền dữ liệu của một số thiết bị ngoại vi có thể nhanh hơn so với bộ nhớ hoặc bộ xử lý. Một lần nữa, sự không khớp này sẽ dẫn đến sự kém hiệu quả nếu không được quản lý đúng cách. Các thiết bị ngoại vi thường sử dụng các định dạng dữ liệu và độ dài từ khác với máy tính mà chúng được kết nối.

**Generic Model of an I/O Module**

**Why an IO module is needed?**• Interface to the processor and memory via the system bus or central switch• Interface to one or more peripheral devices by tailored data links



**External DevicesProvide a means of exchanging data between the external environment and the computerAttach 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 userVideo display terminals (VDTs), printers**Machine readable**Suitable for communicating with equipmentMagnetic disk and tape systems, sensors and actuators (thiết bị khởi phát)**Communication**Suitable for communicating with remote devices such as a terminal, a machine readable device, or another computer

**External Device Block Diagram**

A diagram of a computer system

AI-generated content may be incorrect.

**Keyboard/Monitor**

**International Reference Alphabet (IRA)**

Basic unit of exchange is the characterAssociated with each character is a codeEach character in this code is represented by a unique 7-bit binary code128 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 charactersExample is carriage returnOther control characters are concerned with communications procedures

Most common means of computer/user interactionUser provides input through the keyboardThe monitor displays data provided by the computer

**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 codeThis bit pattern is transmitted to the I/O module in the computerOn output, IRA code characters are transmitted to an external device from the I/O moduleThe 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

**7.2-I/O Modules**

**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**Error detection**Detects and reports transmission errors**Data buffering**Performs the needed buffering operation to balance device and memory speeds**Device communication**Involves commands, status information, and data**Processor communication**Involves command decoding, data, status reporting, address recognition**Control and timing**Coordinates the flow of traffic between internal resources and external devices

**Programmed I/OThree techniques**

A close-up of a sign

AI-generated content may be incorrect.

**Programmed I/O**Data are exchanged between the processor and the I/O moduleProcessor executes a program that gives it direct control of the I/O operationWhen the processor issues a command it must wait until the I/O operation is completeIf 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

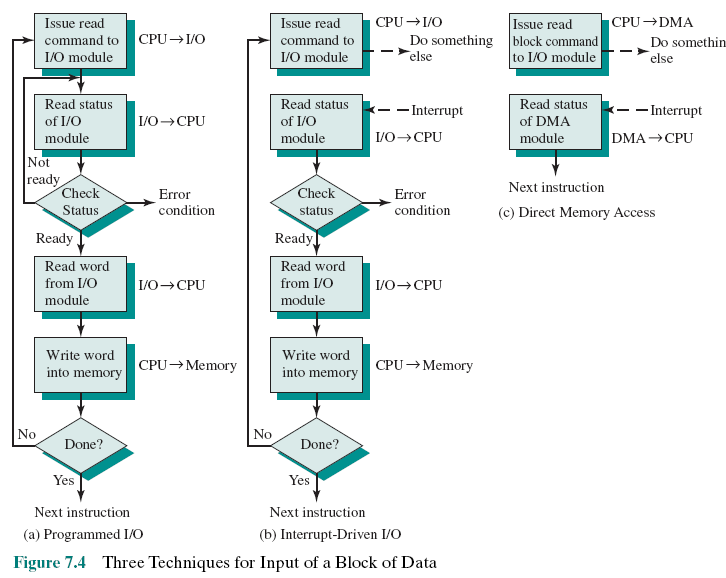
**I/O Module Structure**A diagram of a computer system

AI-generated content may be incorrect.

**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: **Control**- used to activate a peripheral and tell it what to do Test- used to test various status conditions associated with an I/O module and its peripherals

**Test**- used to test various status conditions associated with an I/O module and its peripherals **Read**- causes the I/O module to obtain an item of data from the peripheral and place it in an internal buffer **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



**I/O InstructionsWith 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**

**I/O Mapping SummaryMemory mapped I/ODevices and memory share an address spaceI/O looks just like memory read/writeNo special commands for I/OLarge selection of memory access commands availableIsolated I/OSeparate address spacesNeed I/O or memory select linesSpecial commands for I/OLimited set**

**7.4- Interrupt-Driven I/O** Hạn chế của Programmed I/O: CPU phải chờ I/O module sẵn sàng.

 Giải pháp: CPU gửi lệnh I/O rồi làm việc khác.

 Khi I/O module sẵn sàng, nó gửi interrupt đến CPU.

 CPU xử lý dữ liệu xong, rồi tiếp tục công việc trước đó

**Design IssuesTwo 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 IdentificationFour general categories of techniques are in common use:-** **Multiple interrupt lines:  Kết nối giữa CPU và các mô-đun I/O.**

** Giải pháp đơn giản nhất để giao tiếp I/O.**

** Một đường bus có thể kết nối nhiều mô-đun I/O.**

**-** **Software poll:  Khi CPU phát hiện interrupt, nó chuyển đến interrupt-service routine.**

** CPU kiểm tra từng mô-đun I/O để tìm ra mô-đun gây ngắt.**

** Nhược điểm: Tốn thời gian khi có nhiều thiết bị I/O- Daisy chain (hardware poll, vectored):** ** Dây tín hiệu xác nhận ngắt (interrupt acknowledge) được mắc chuỗi (daisy chain) qua các mô-đun.**

** Vector: Địa chỉ mô-đun I/O hoặc mã nhận diện duy nhất.**

** Vectored Interrupt: CPU dùng vector để gọi đúng routine xử lý thiết bị, không cần kiểm tra từng mô-đun, giúp xử lý nhanh hơn.**

**-** **Bus arbitration (vectored):** ** Mô-đun I/O phải giành quyền kiểm soát bus trước khi gửi yêu cầu ngắt.**

** CPU phát hiện ngắt và phản hồi qua interrupt acknowledge line.**

** Mô-đun yêu cầu đặt vector của nó lên bus dữ liệu, giúp CPU xác định nhanh thiết bị cần xử lý.**

**Drawbacks of Programmed and Interrupt-Driven I/OBoth 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 device2.The processor is tied up in managing an I/O transfer; a number of instructions must be executed for each I/O transferWhen large volumes of data are to be moved a more efficient technique is direct memory access (DMA)**

**7.5- Direct Memory Access**

**DMA Operation**A diagram of a breakpoint

AI-generated content may be incorrect.

**** CPU bị tạm dừng trước khi sử dụng bus, nhường quyền cho DMA.

 DMA truyền một từ (word) dữ liệu, sau đó trả lại quyền kiểm soát cho CPU.

 Không phải ngắt (interrupt) → CPU không lưu context mà chỉ tạm dừng một chu kỳ bus.

 Giảm tốc độ CPU nhưng hiệu quả hơn interrupt-driven/programmed I/O khi truyền nhiều dữ liệu

**Fly-By DMA Controller**Data does not pass through and is not stored in DMA chipDMA only between I/O port and memoryNot between two I/O ports or two memory locationsCan do memory to memory via regist

8237 contains four DMA channelsProgrammed independentlyAny one active Numbered 0, 1, 2, and 3

**7.6- IO Channels and Processors  
Evolution of the I/O Function**

+The CPU directly controls a peripheral device. +A controller or I/O module is added. The CPU uses **programmed I/O** without interrupts.+Same configuration as in step 2 is used, but now interrupts are employed. The CPU need not spend time waiting for an I/O operation to be performed, thus increasing efficiency.+The I/O module is given direct access to memory via DMA. It can now move a block of data to or from memory without involving the CPU, except at the beginning and end of the transfer.+The I/O module is enhanced to become a processor in its own right, with a specialized instruction set tailored for I/O+The I/O module has a local memory of its own and is, in fact, a computer in its own right. With this architecture a large set of I/O devices can be controlled with minimal CPU involvement.