

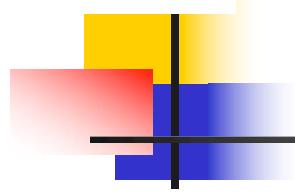


**KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN MẠNG VÀ CÁC HỆ THỐNG THÔNG TIN**

CHƯƠNG 5

HỆ THỐNG VÀO RA

Kiến trúc và Tổ chức máy tính

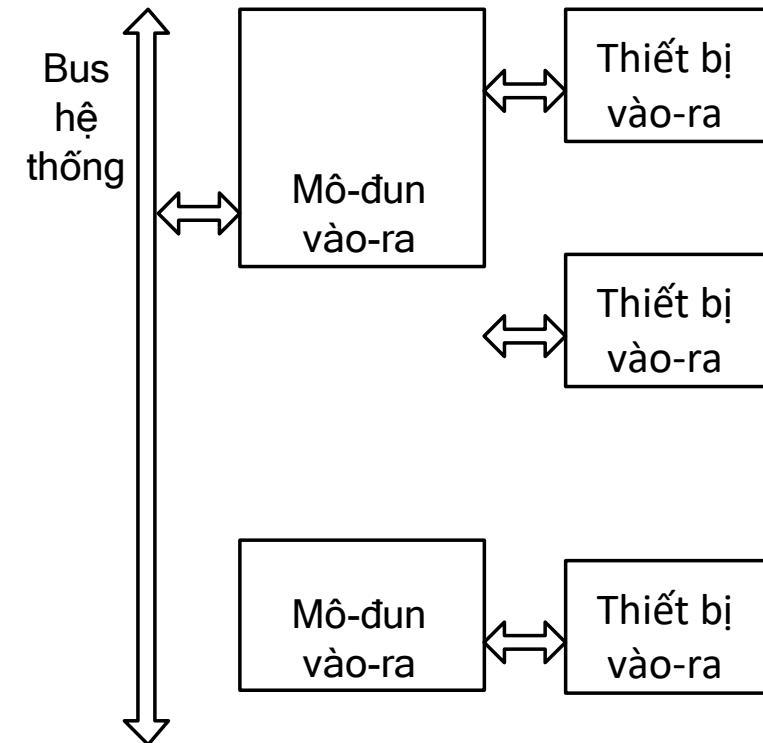


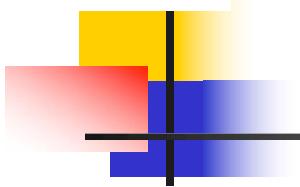
Nội dung

- 5.1. Tổng quan về hệ thống vào/ra
- 5.2. Các phương pháp điều khiển vào/ra
- 5.3. Nối ghép thiết bị ngoại vi
- 5.4. Các cổng vào/ra thông dụng trên PC

5.1. Tổng quan về hệ thống vào/ra

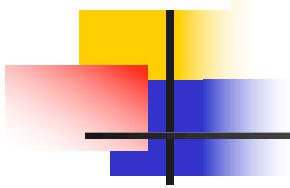
- Chức năng: Trao đổi thông tin giữa máy tính với bên ngoài
- Các thao tác cơ bản:
 - Vào dữ liệu (Input)
 - Ra dữ liệu (Output)
- Các thành phần chính:
 - Các thiết bị vào-ra
 - Các mô-đun vào-ra





Đặc điểm của hệ thống vào/ra

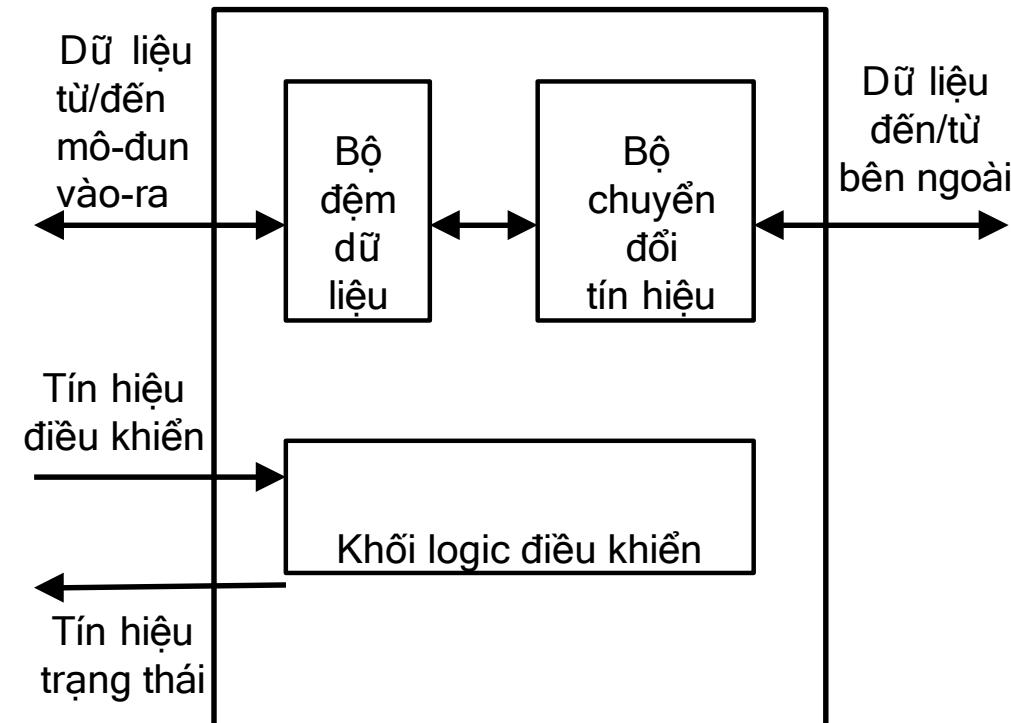
- Tồn tại đa dạng các thiết bị vào-ra khác nhau về:
 - Nguyên tắc hoạt động
 - Tốc độ
 - Khuôn dạng dữ liệu
- Tất cả các thiết bị vào-ra đều chậm hơn CPU và RAM
→ Cần có các mô-đun vào-ra để nối ghép các thiết bị với CPU và bộ nhớ chính

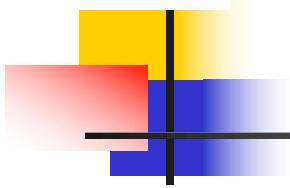


Thiết bị vào/ra

- Còn gọi là thiết bị ngoại vi (Peripherals)
- Chức năng: chuyển đổi dữ liệu giữa bên trong và bên ngoài máy tính
- Phân loại:
 - Thiết bị vào (Input Devices)
 - Thiết bị ra (Output Devices)
 - Thiết bị lưu trữ (Storage Devices)
 - Thiết bị truyền thông (Communication Devices)
- Giao tiếp:
 - Người - máy
 - Máy - máy

Cấu trúc chung của thiết bị vào-ra

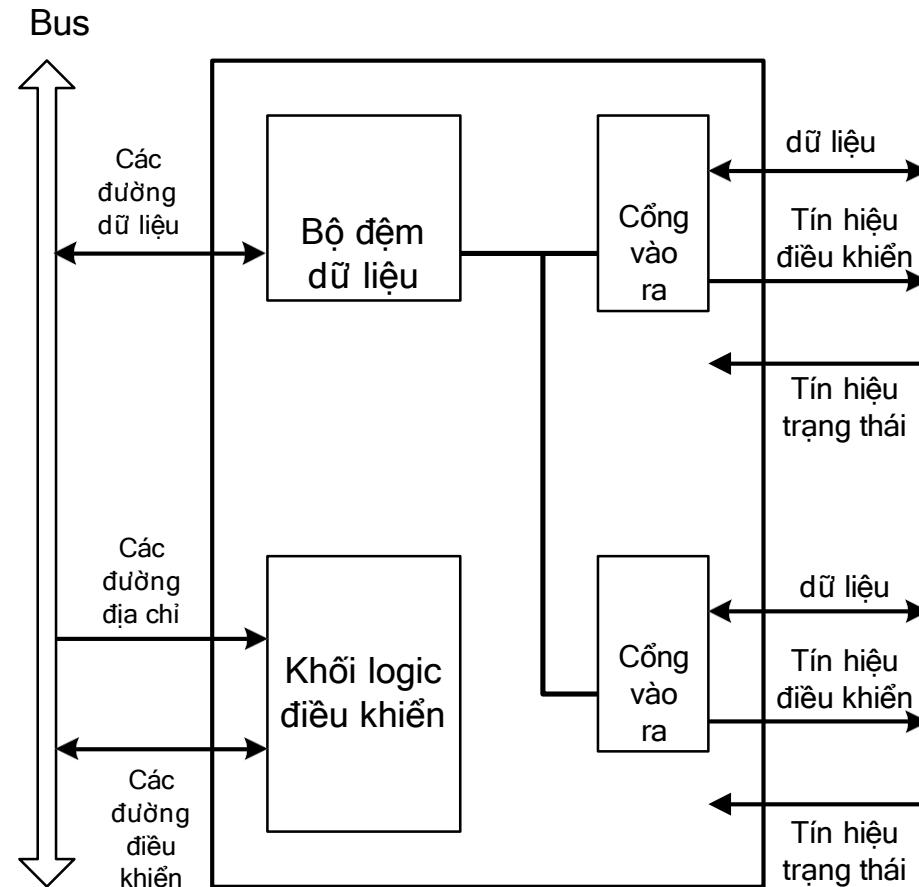


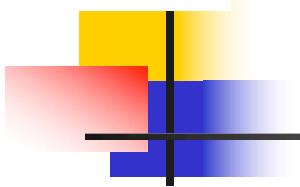


Mô-đun vào/ra

- Chức năng:
 - Điều khiển và định thời
 - Trao đổi thông tin với CPU hoặc bộ nhớ chính
 - Trao đổi thông tin với thiết bị vào-ra
 - Đệm giữa bên trong máy tính với thiết bị vào-ra
 - Phát hiện lỗi của thiết bị vào-ra

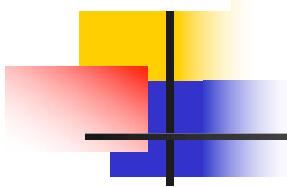
Cấu trúc của mô-đun vào-ra



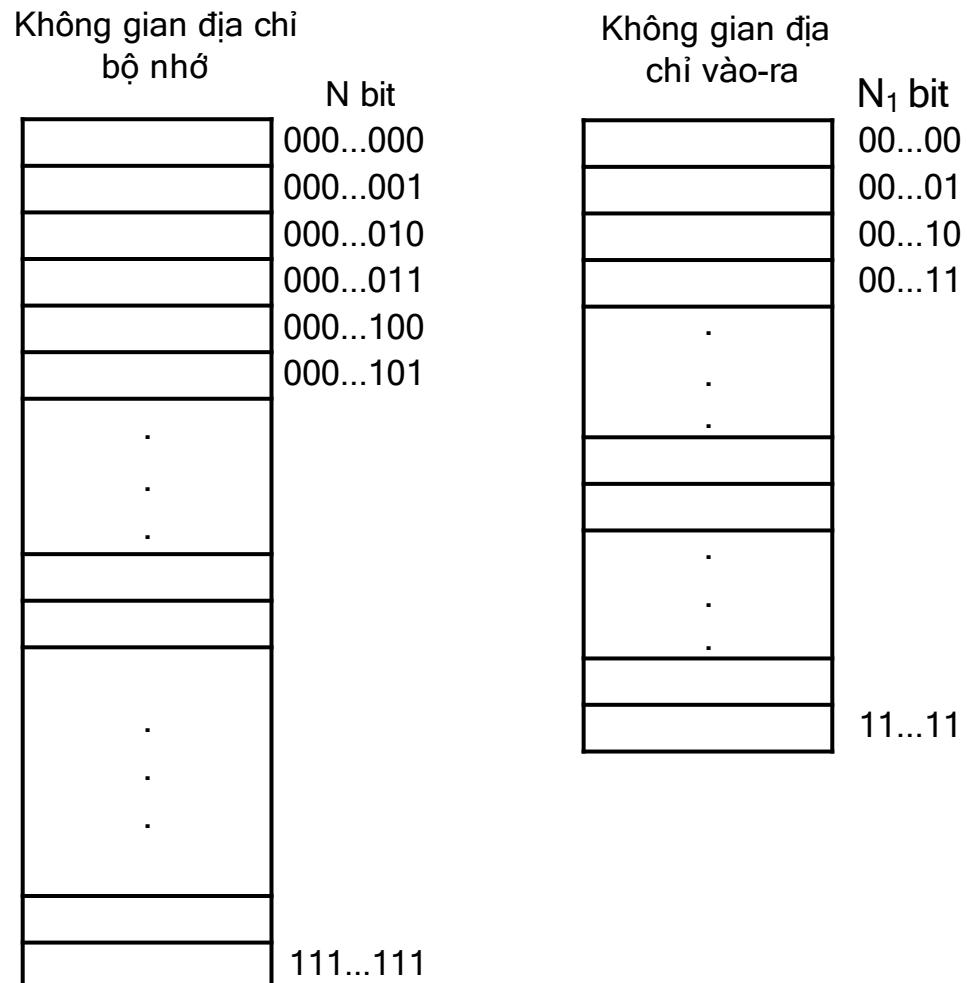


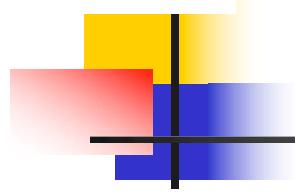
4. Địa chỉ hóa cổng vào-ra (IO addressing)

- Hầu hết các bộ xử lý chỉ có một không gian địa chỉ chung cho cả các ngăn nhớ và các cổng vào-ra
 - Các bộ xử lý 680x0 của Motorola
 - Các bộ xử lý theo kiến trúc RISC: MIPS, ARM, ...
- Một số bộ xử lý có hai không gian địa chỉ tách biệt:
 - Không gian địa chỉ bộ nhớ
 - Không gian địa chỉ vào-ra
 - Ví dụ: Intel x86



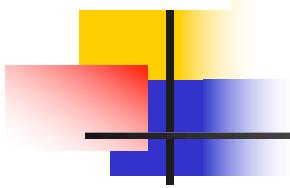
Không gian địa chỉ tách biệt





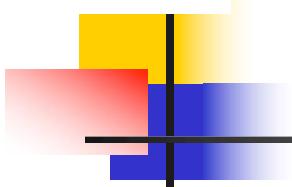
Các phương pháp địa chỉ hoá cổng vào-ra

- Vào-ra theo bản đồ bộ nhớ
(Memory mapped IO)
- Vào-ra riêng biệt
(Isolated IO hay IO mapped IO)



Vào-ra theo bản đồ bộ nhớ

- Cổng vào-ra được đánh địa chỉ theo không gian địa chỉ bộ nhớ
- CPU coi cổng vào-ra như ngăn nhớ
- Lập trình trao đổi dữ liệu với cổng vào-ra bằng các lệnh truy nhập dữ liệu bộ nhớ
- Có thể thực hiện trên mọi hệ thống
- Ví dụ: Bộ xử lý MIPS
 - 32-bit địa chỉ cho một không gian địa chỉ chung cho cả các ngăn nhớ và các cổng vào-ra
 - Các cổng vào-ra được gắn các địa chỉ thuộc vùng địa chỉ dự trữ
 - Vào/ra dữ liệu: sử dụng lệnh load/store



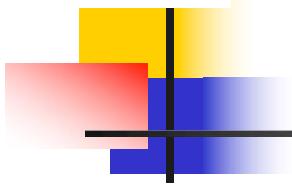
Ví dụ lập trình vào-ra cho MIPS

- Ví dụ: Có hai cổng vào-ra được gán địa chỉ:
 - Cổng 1: 0xFFFFFFFF4
 - Cổng 2: 0xFFFFFFFF8
- Ghi giá trị 0x41 ra cổng 1

```
addi $t0, $0, 0x41      # đưa giá trị 0x41  
sw    $t0, 0xFFFF4($0)  # ra cổng 1
```

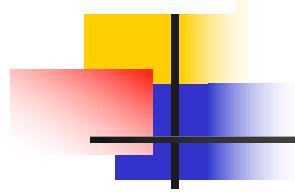
Chú ý: giá trị 16-bit 0xFFFF4 được sign-extended thành 32-bit 0xFFFFFFFF4

- Đọc dữ liệu từ cổng 2 đưa vào \$t3
- ```
lw $t3, 0xFFFF8($0) # đọc dữ liệu cổng 2 đưa vào $t3
```



## Vào-ra riêng biệt (Isolated IO)

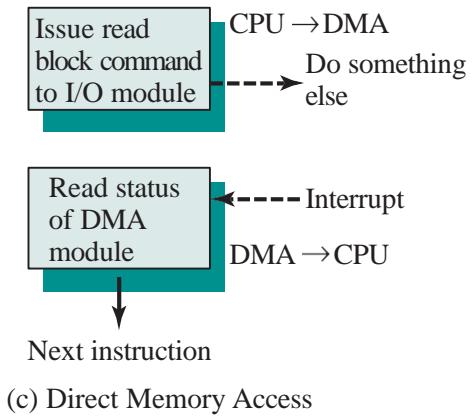
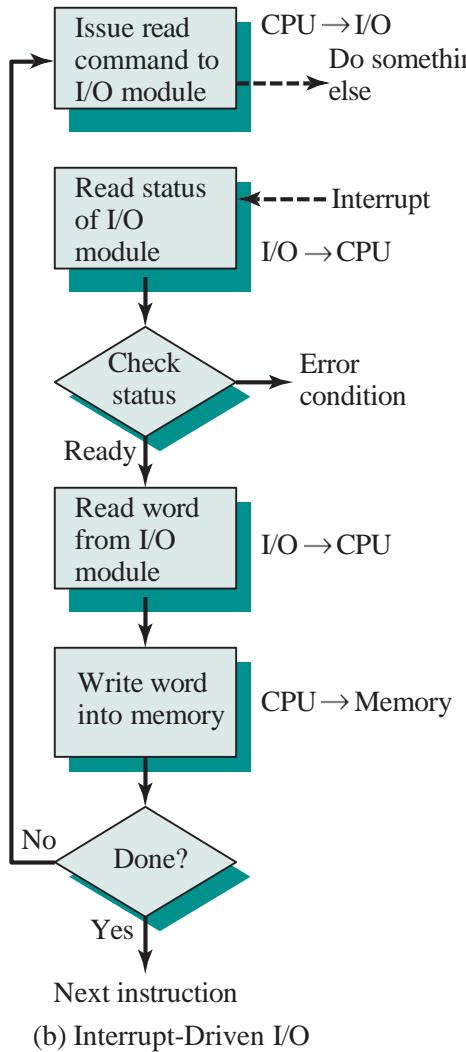
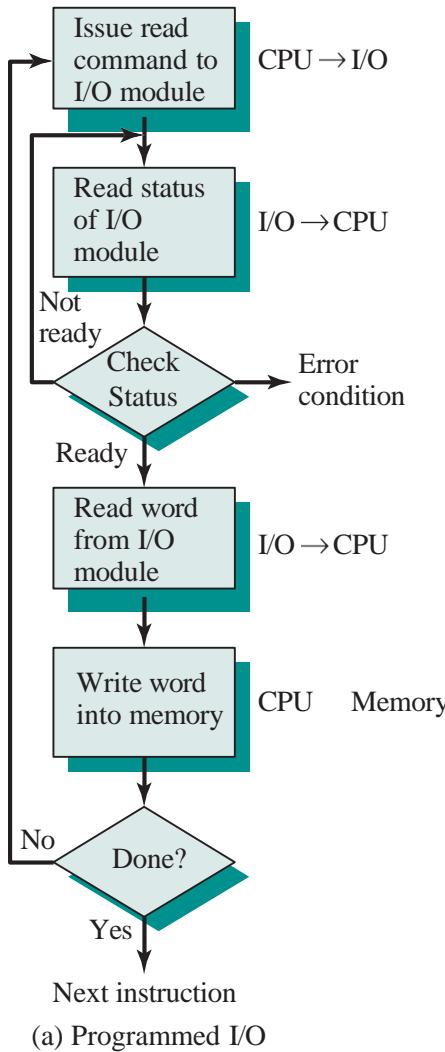
- Cổng vào-ra được đánh địa chỉ theo không gian địa chỉ vào-ra riêng
- Lập trình trao đổi dữ liệu với cổng vào-ra bằng các lệnh vào-ra chuyên dụng
- Ví dụ: Intel x86
  - Dùng 8-bit hoặc 16-bit địa chỉ cho không gian địa chỉ vào-ra riêng
  - Có hai lệnh vào-ra chuyên dụng
    - Lệnh IN: nhận dữ liệu từ cổng vào
    - Lệnh OUT: đưa dữ liệu đến cổng ra



## 8.2. Các phương pháp điều khiển vào-ra

- Vào-ra bằng chương trình  
(Programmed IO)
- Vào-ra điều khiển bằng ngắt  
(Interrupt Driven IO)
- Truy nhập bộ nhớ trực tiếp - DMA  
(Direct Memory Access)

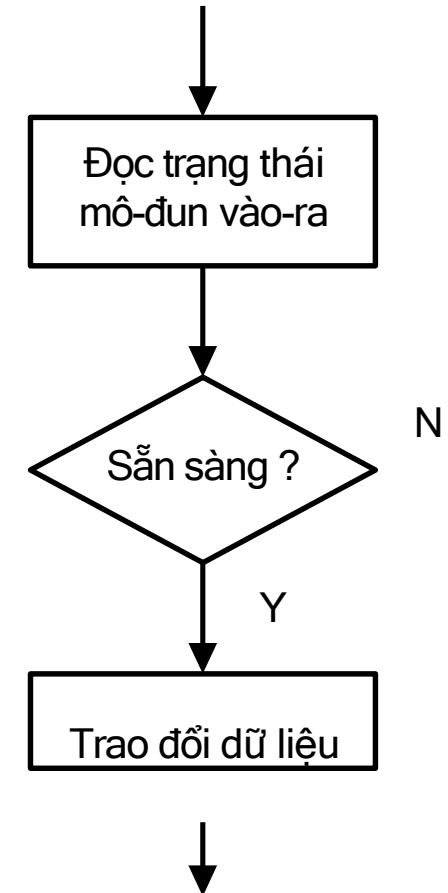
# Ba kỹ thuật thực hiện vào một khối dữ liệu

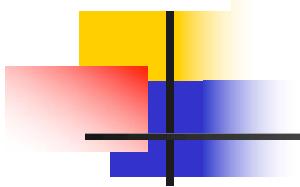


# 1. Vào-ra bằng chương trình

## ■ Nguyên tắc chung:

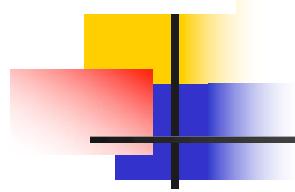
- CPU điều khiển trực tiếp vào-ra bằng chương trình → cần phải lập trình vào-ra để trao đổi dữ liệu giữa CPU với mô-đun vào-ra
- CPU nhanh hơn thiết bị vào-ra rất nhiều lần, vì vậy trước khi thực hiện lệnh vào-ra, chương trình cần đọc và kiểm tra trạng thái sẵn sàng của mô-đun vào-ra





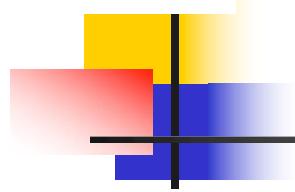
## Các tín hiệu điều khiển vào-ra

- Tín hiệu **điều khiển** (*Control*): kích hoạt thiết bị vào-ra
- Tín hiệu **kiểm tra** (*Test*): kiểm tra trạng thái của mô-đun vào-ra và thiết bị vào-ra
- Tín hiệu điều khiển **đọc** (*Read*): yêu cầu mô-đun vào-ra nhận dữ liệu từ thiết bị vào-ra và đưa vào bộ đệm dữ liệu, rồi CPU nhận dữ liệu đó
- Tín hiệu điều khiển **ghi** (*Write*): yêu cầu mô-đun vào-ra lấy dữ liệu trên bus dữ liệu đưa đến bộ đệm dữ liệu rồi chuyển ra thiết bị vào-ra



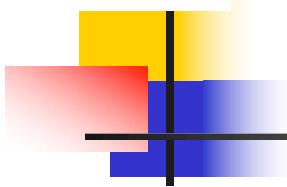
## Các lệnh vào-ra

- Với vào-ra theo bản đồ bộ nhớ: sử dụng các lệnh trao đổi dữ liệu với bộ nhớ để trao đổi dữ liệu với cổng vào-ra
- Với vào-ra riêng biệt: sử dụng các lệnh vào-ra chuyên dụng (IN, OUT)



## Đặc điểm

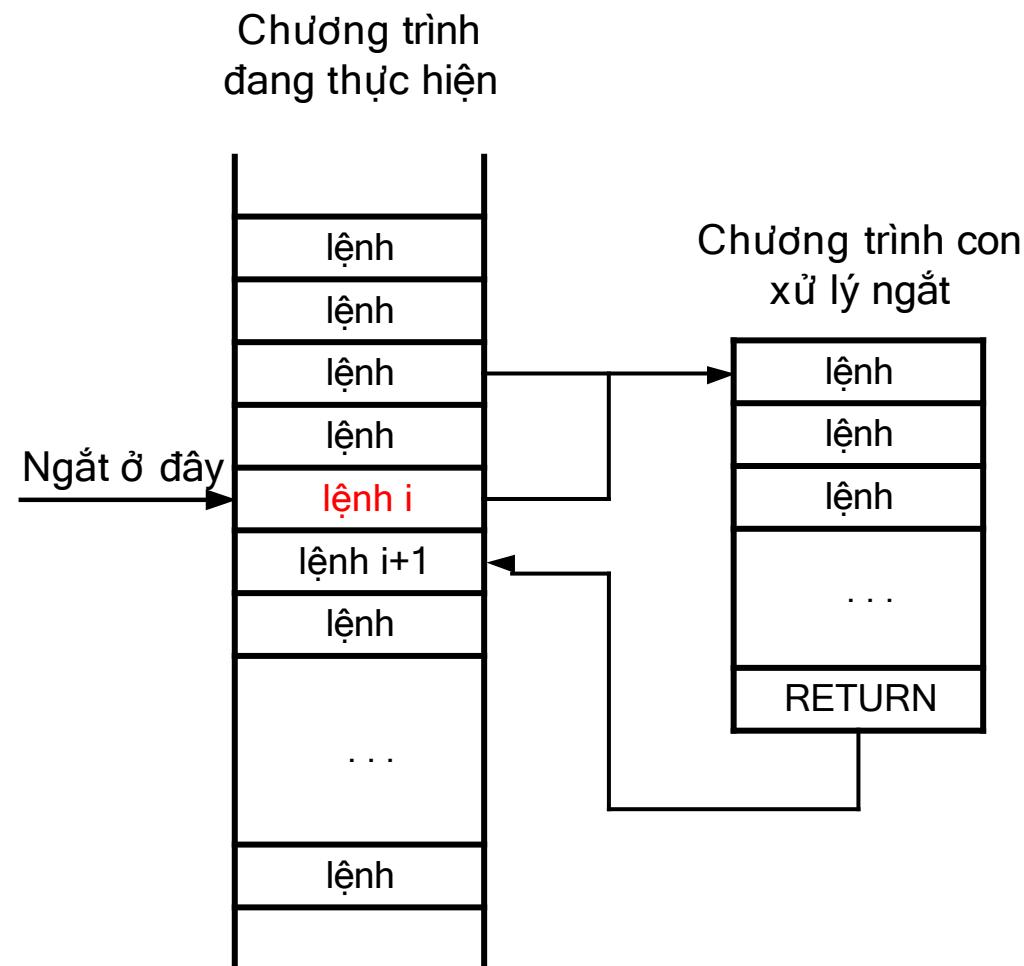
- Vào-ra do ý muốn của người lập trình
- CPU trực tiếp điều khiển trao đổi dữ liệu giữa CPU với mô-đun vào-ra
- CPU đợi trạng thái sẵn sàng của mô-đun vào-ra (qua vòng lặp) → tiêu tốn nhiều thời gian của CPU

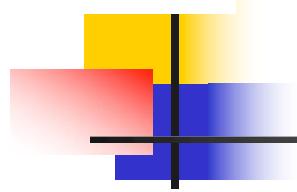


## 2. Vào-ra điều khiển bằng ngắt

- Nguyên tắc chung:
  - CPU không phải đợi trạng thái sẵn sàng của mô-đun vào-ra, CPU thực hiện một chương trình nào đó
  - Khi mô-đun vào-ra sẵn sàng thì nó phát tín hiệu ngắt CPU
  - CPU thực hiện chương trình con xử lý ngắt vào-ra tương ứng để trao đổi dữ liệu
  - CPU trở lại tiếp tục thực hiện chương trình đang bị ngắt

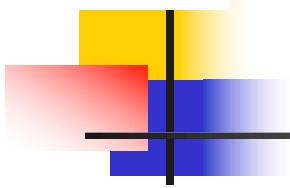
# Chuyển điều khiển đến chương trình con ngắt





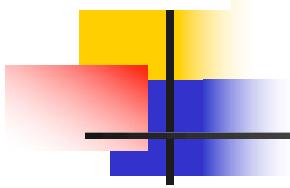
## Hoạt động vào dữ liệu: nhìn từ mô-đun vào-ra

- Mô-đun vào-ra nhận tín hiệu điều khiển đọc từ CPU
- Mô-đun vào-ra nhận dữ liệu từ thiết bị vào-ra, trong khi đó CPU làm việc khác
- Khi đã có dữ liệu → mô-đun vào-ra phát tín hiệu ngắt CPU
- CPU yêu cầu dữ liệu
- Mô-đun vào-ra chuyển dữ liệu đến CPU



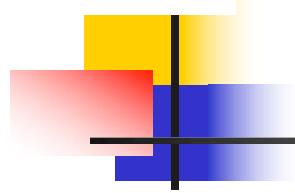
# Hoạt động vào dữ liệu: nhìn từ CPU

- Phát tín hiệu điều khiển **đọc**
- Làm việc khác
- Cuối mỗi chu trình lệnh, kiểm tra tín hiệu yêu cầu ngắt
- Nếu bị ngắt:
  - Cắt ngũ cảnh (nội dung các thanh ghi liên quan)
  - Thực hiện chương trình con xử lý ngắt để vào dữ liệu
  - Khôi phục ngũ cảnh của chương trình đang thực hiện



## Các vấn đề này sinh khi thiết kế

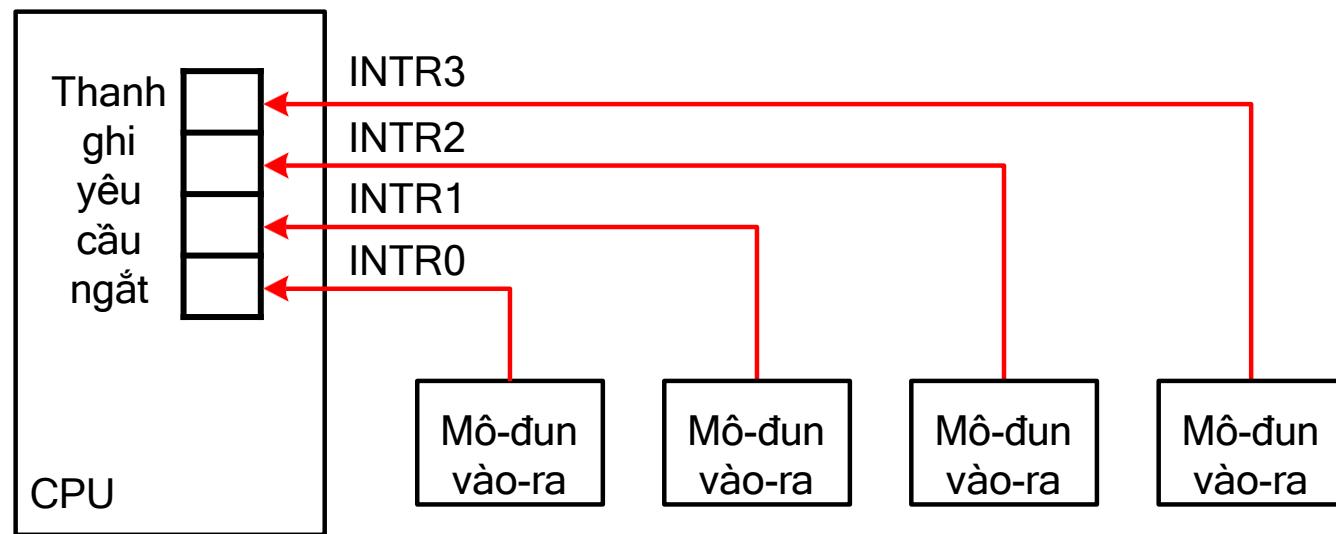
- Làm thế nào để xác định được mô-đun vào-ra nào phát tín hiệu ngắt ?
- CPU làm như thế nào khi có nhiều yêu cầu ngắt cùng xảy ra ?



## Các phương pháp nối ghép ngắt

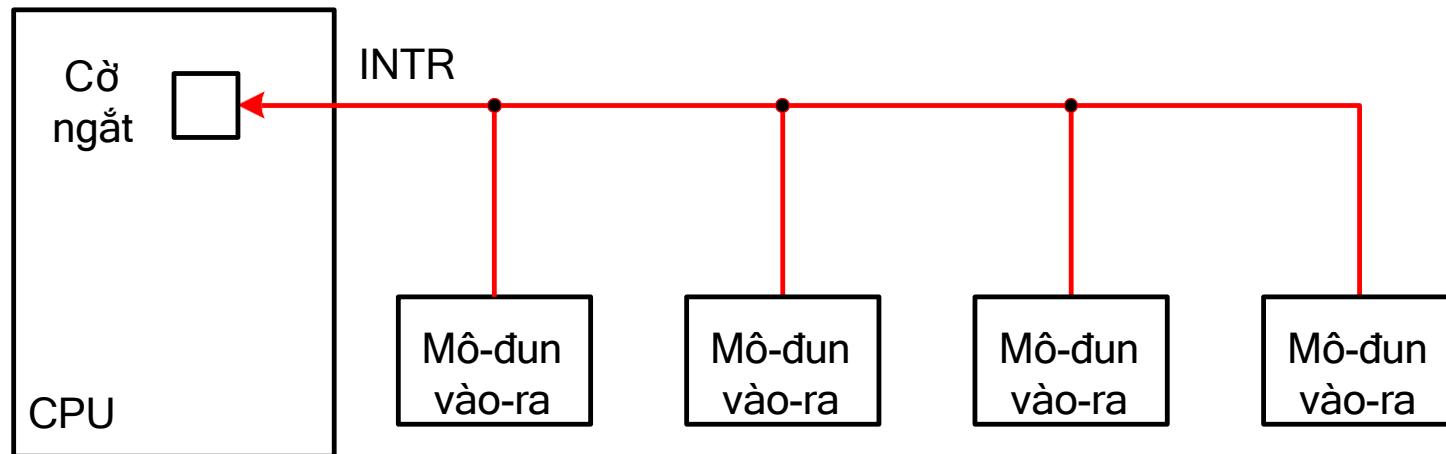
- Sử dụng nhiều đường yêu cầu ngắt
- Hỏi vòng bằng phần mềm (Software Poll)
- Hỏi vòng bằng phần cứng (Daisy Chain or Hardware Poll)
- Sử dụng bộ điều khiển ngắt (PIC)

# Nhiều đường yêu cầu ngắt



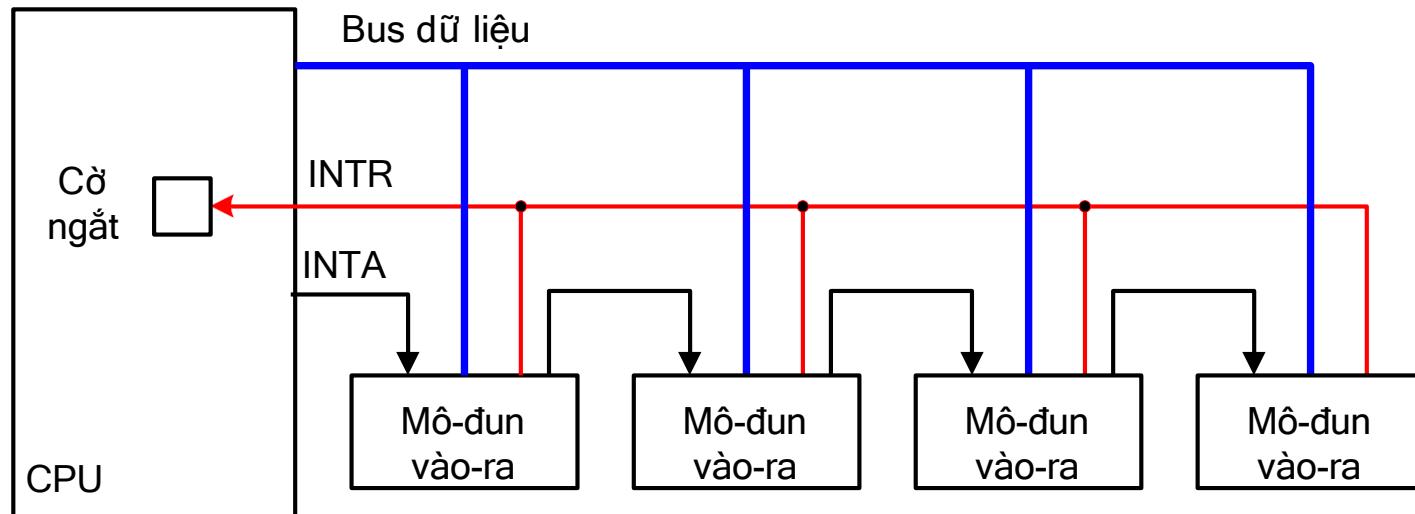
- Mỗi mô-đun vào-ra được nối với một đường yêu cầu ngắt
- CPU phải có nhiều đường tín hiệu yêu cầu ngắt
- Hạn chế số lượng mô-đun vào-ra
- Các đường ngắt được qui định mức ưu tiên

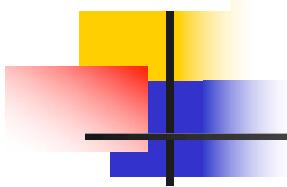
# Hỏi vòng bằng phần mềm



- CPU thực hiện phần mềm hỏi lân lượt từng mô-đun vào-ra
- Chậm
- Thứ tự các mô-đun được hỏi vòng chính là thứ tự ưu tiên

# Hỏi vòng bằng phàn cứng

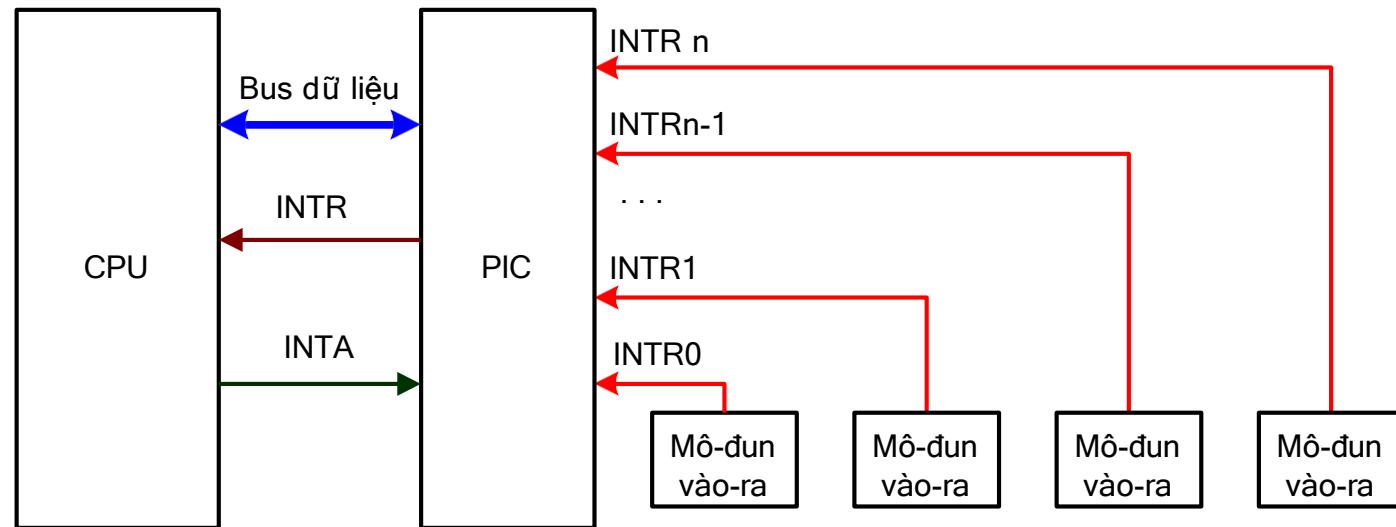




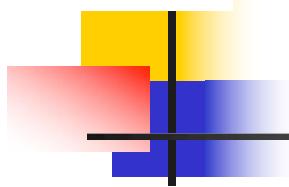
## Hỏi vòng bằng phần cứng (tiếp)

- CPU phát tín hiệu chấp nhận ngắn (INTA) đến mô-đun vào-ra đầu tiên
- Nếu mô-đun vào-ra đó không gây ra ngắn thì nó gửi tín hiệu đến mô-đun kế tiếp cho đến khi xác định được mô-đun gây ngắn
- Thứ tự các mô-đun vào-ra kết nối trong chuỗi xác định thứ tự ưu tiên

# Bộ điều khiển ngắt lập trình được

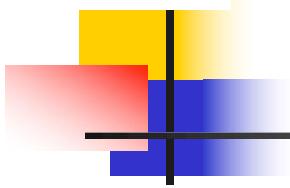


- PIC - Programmable Interrupt Controller
- PIC có nhiều đường vào yêu cầu ngắt có qui định mức ưu tiên
- PIC chọn một yêu cầu ngắt không bị cấm có mức ưu tiên cao nhất gửi tới CPU



## Đặc điểm của vào-ra điều khiển bằng ngắt

- Có sự kết hợp giữa phần cứng và phần mềm
  - Phần cứng: gây ngắt CPU
  - Phần mềm: trao đổi dữ liệu giữa CPU với mô-đun vào-ra
- CPU trực tiếp điều khiển vào-ra
- CPU không phải đợi mô-đun vào-ra, do đó hiệu quả sử dụng CPU tốt hơn

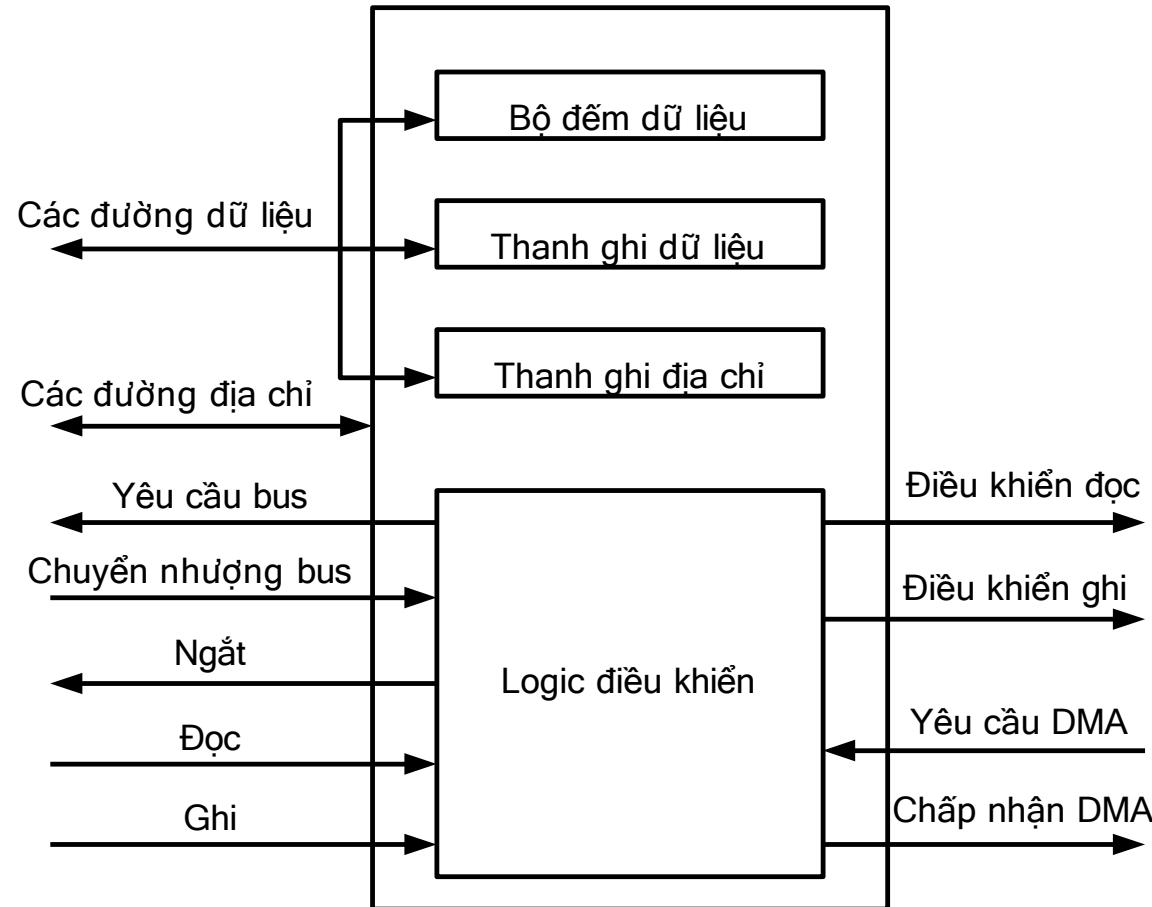


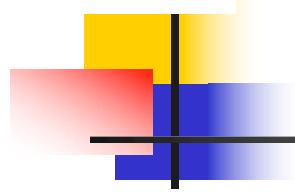
### 3. DMA (Direct Memory Access)

---

- Vào-ra bằng chương trình và bằng ngắt do CPU trực tiếp điều khiển:
  - Chiếm thời gian của CPU
- Để khắc phục dùng kỹ thuật DMA
  - Sử dụng mô-đun điều khiển vào-ra chuyên dụng, gọi là DMAC (Controller), điều khiển trao đổi dữ liệu giữa mô-đun vào-ra với bộ nhớ chính

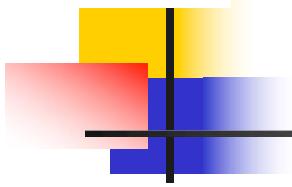
# Sơ đồ cấu trúc của DMAC





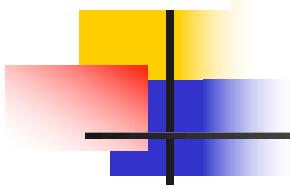
## Các thành phần của DMAC

- Thanh ghi dữ liệu: chứa dữ liệu trao đổi
- Thanh ghi địa chỉ: chứa địa chỉ ngăn nhớ dữ liệu
- Bộ đếm dữ liệu: chứa số từ dữ liệu cần trao đổi
- Logic điều khiển: điều khiển hoạt động của DMAC



# Hoạt động DMA

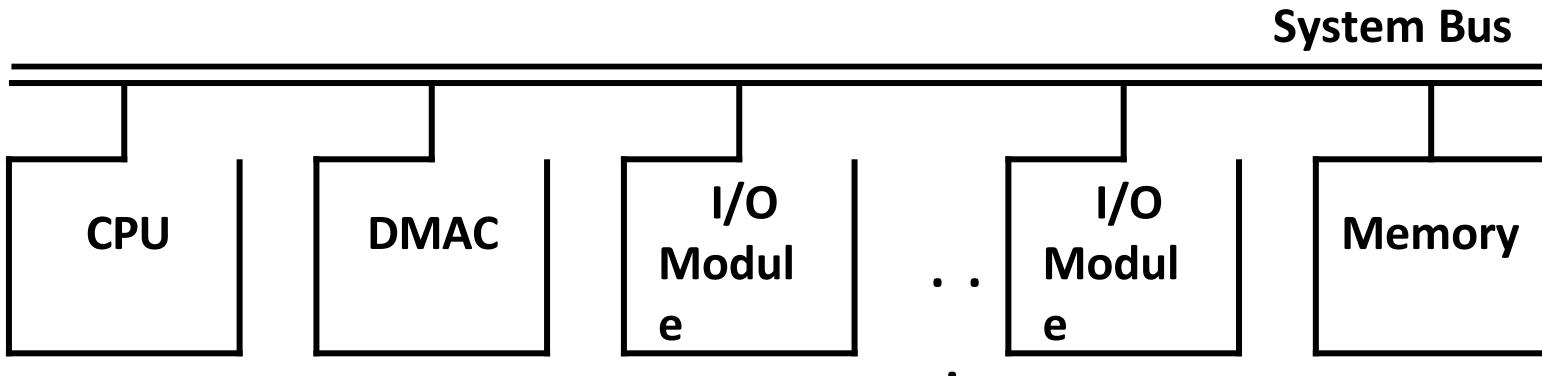
- CPU “nói” cho DMAC
  - Vào hay Ra dữ liệu
  - Địa chỉ thiết bị vào-ra (cổng vào-ra tương ứng)
  - Địa chỉ đầu của mảng nhớ chứa dữ liệu → nạp vào thanh ghi địa chỉ
  - Số từ dữ liệu cần truyền → nạp vào bộ đếm dữ liệu
- CPU làm việc khác
- DMAC điều khiển trao đổi dữ liệu
- Sau khi truyền được một từ dữ liệu thì:
  - nội dung thanh ghi địa chỉ tăng
  - nội dung bộ đếm dữ liệu giảm
- Khi bộ đếm dữ liệu = 0, DMAC gửi tín hiệu ngắt CPU để báo kết thúc DMA



## Các kiểu thực hiện DMA

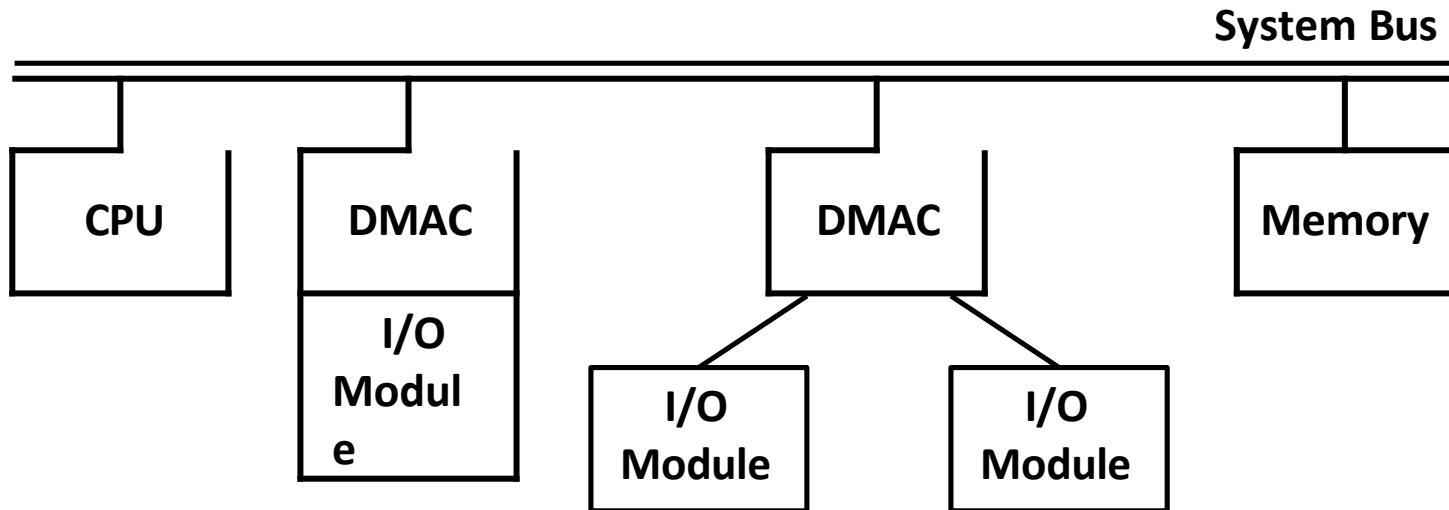
- DMA truyền theo khối (Block-transfer DMA): DMAC sử dụng bus để truyền xong cả khối dữ liệu
- DMA lấy chu kỳ (Cycle Stealing DMA): DMAC cưỡng bức CPU treo tạm thời từng chu kỳ bus, DMAC chiếm bus thực hiện truyền một từ dữ liệu.
- DMA trong suốt (Transparent DMA): DMAC nhận biết những chu kỳ nào CPU không sử dụng bus thì chiếm bus để trao đổi một từ dữ liệu.

# Cấu hình DMA (1)



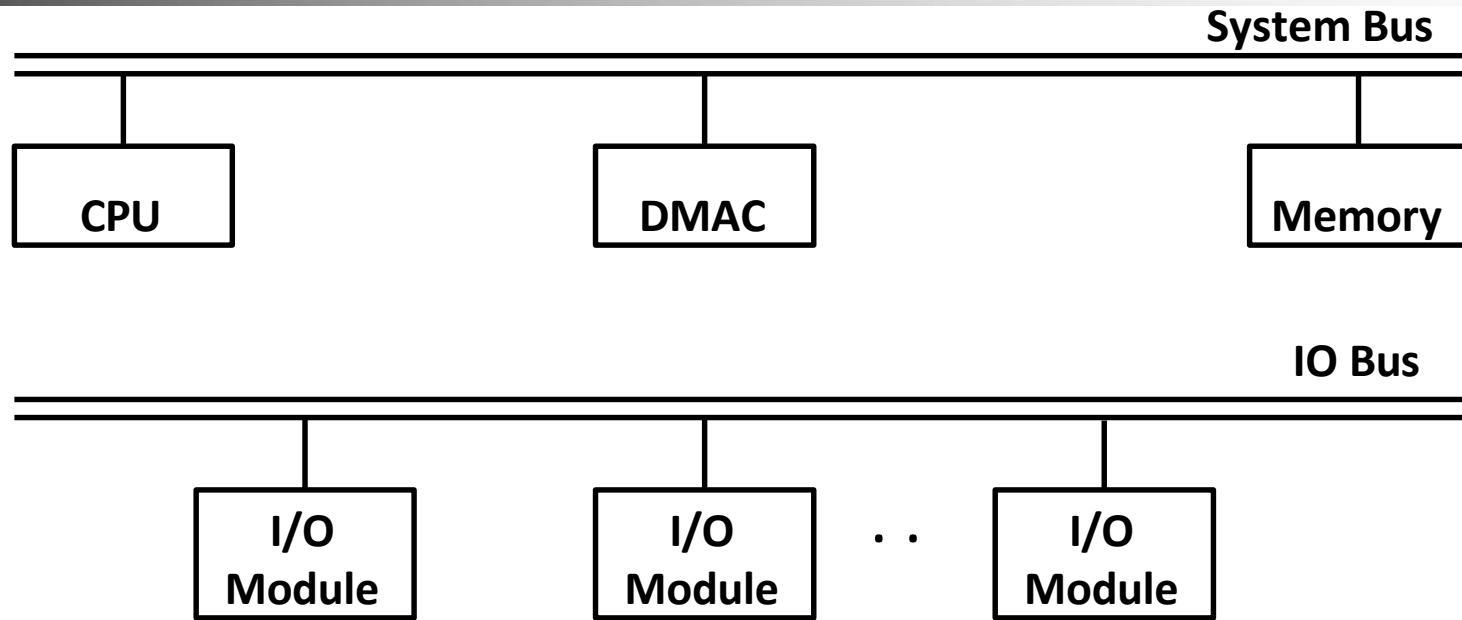
- Mỗi lần trao đổi một dữ liệu, DMAC sử dụng bus hai lần
  - Giữa mô-đun vào-ra với DMAC
  - Giữa DMAC với bộ nhớ

## Cấu hình DMA (2)

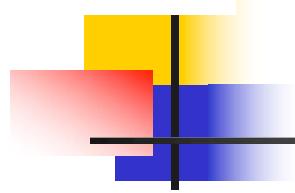


- DMAC điều khiển một hoặc vài mô-đun vào-ra
- Mỗi lần trao đổi một dữ liệu, DMAC sử dụng bus một lần
  - Giữa DMAC với bộ nhớ

## Cấu hình DMA (3)

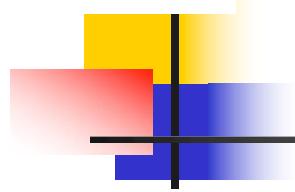


- Bus vào-ra tách rời hỗ trợ tất cả các thiết bị cho phép DMA
- Mỗi lần trao đổi một dữ liệu, DMAC sử dụng bus một lần
  - Giữa DMAC với bộ nhớ



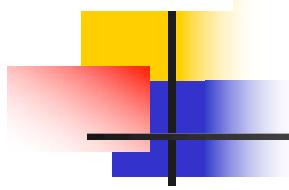
## Đặc điểm của DMA

- CPU không tham gia trong quá trình trao đổi dữ liệu
- DMAC điều khiển trao đổi dữ liệu giữa bộ nhớ chính với mô-đun vào-ra (hoàn toàn bằng phần cứng) → tốc độ nhanh
- Phù hợp với các yêu cầu trao đổi mảng dữ liệu có kích thước lớn



## 4. Bộ xử lý vào-ra

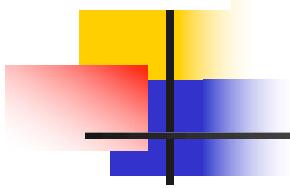
- Việc điều khiển vào-ra được thực hiện bởi một bộ xử lý vào-ra chuyên dụng
- Bộ xử lý vào-ra hoạt động theo chương trình của riêng nó
- Chương trình của bộ xử lý vào-ra có thể nằm trong bộ nhớ chính hoặc nằm trong một bộ nhớ riêng



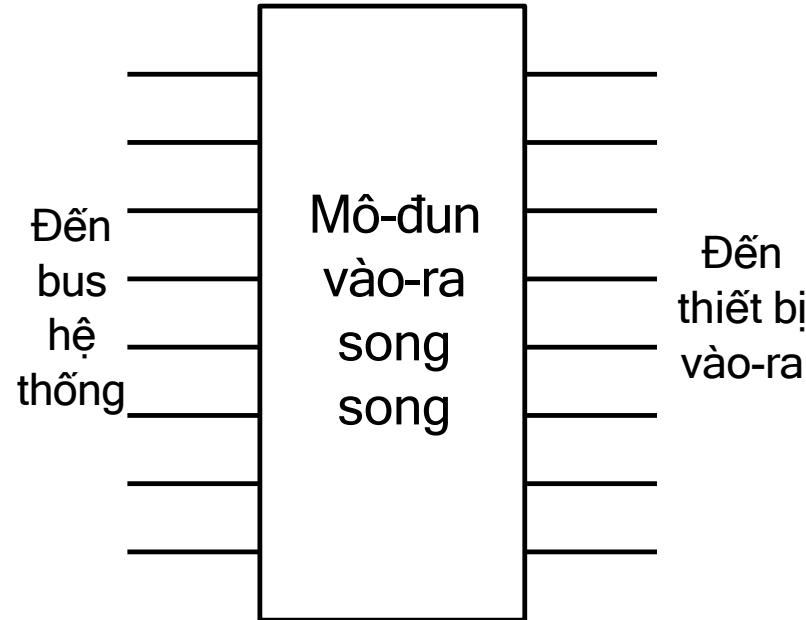
## 5.3. Nối ghép thiết bị ngoại vi

### 1. Các kiểu nối ghép vào-ra

- Nối ghép song song
- Nối ghép nối tiếp

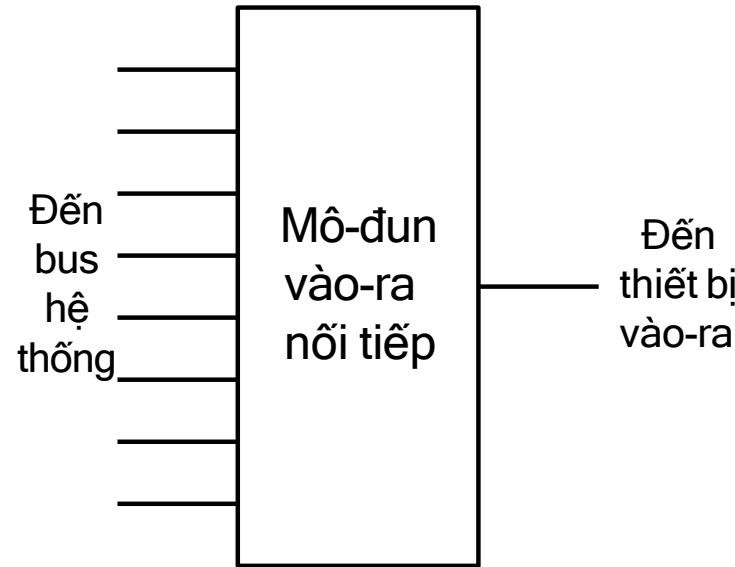


# Nối ghép song song

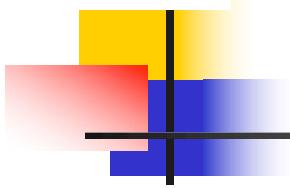


- Truyền nhiều bit song song
- Tốc độ nhanh
- Cần nhiều đường truyền dữ liệu

# Nối ghép nối tiếp



- Truyền lần lượt từng bit
- Cần có bộ chuyển đổi từ dữ liệu song song sang nối tiếp hoặc/và ngược lại
- Tốc độ chậm hơn
- Cần ít đường truyền dữ liệu



## 2. Các cấu hình nối ghép

- **Điểm tới điểm (Point to Point)**
  - Thông qua một cổng vào-ra nối ghép với một thiết bị
- **Điểm tới đa điểm (Point to Multipoint)**
  - Thông qua một cổng vào-ra cho phép nối ghép được với nhiều thiết bị
  - **Ví dụ:**
    - USB (Universal Serial Bus): 127 thiết bị
    - IEEE 1394 (FireWire): 63 thiết bị
    - Thunderbolt

# Thunderbolt

