Kiến trúc máy tính

Các thiết bị vào/ra

NGUYỄN Ngọc Hoá

Bộ môn Hệ thống thông tin, Khoa CNTT

Trường Đại học Công nghệ,

Đại học Quốc gia Hà Nội

Hoa.Nguyen@vnu.edu.vn

Nội dung

1. Giới thiệu

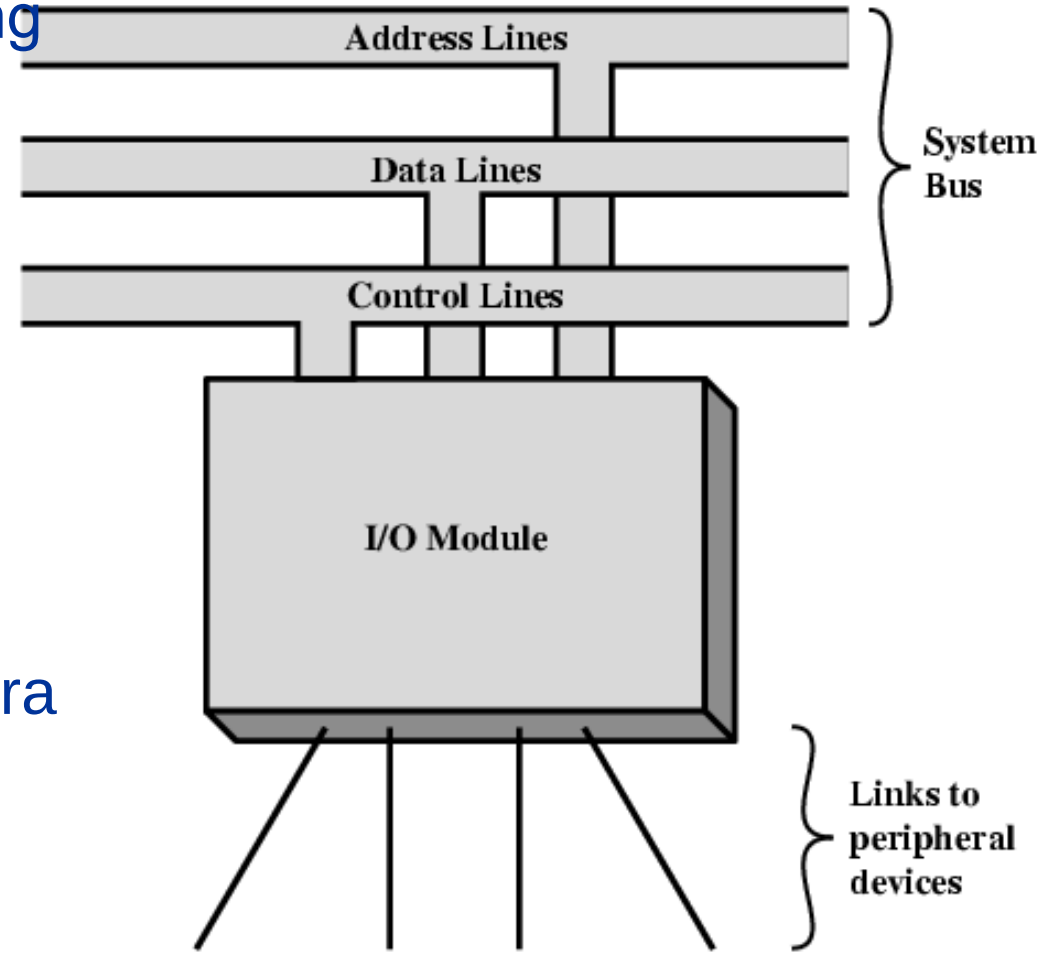
2. Khối kiểm soát vào/ra
   - Chức năng
   - Cấu trúc

3. Kỹ thuật kiểm soát vào/ra
   - Programmed I/O
   - Interrupt driven I/O
   - Direct Memory Access – DMA
   - Kênh vào/ra & CPU

4. Interfacing

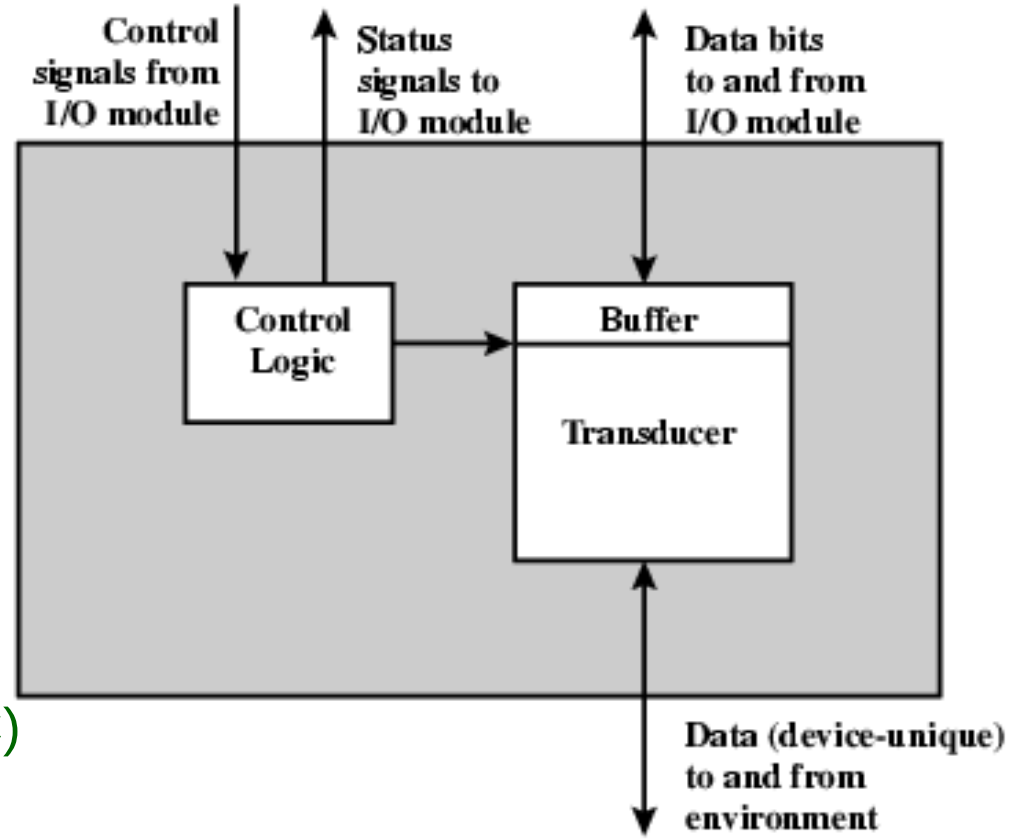
1. Quản lý vào/ra (I/O)

- Rất nhiều thiết bị ngoại vi – peripherals, dữ liệu thường
  - Khối lượng khác nhau
  - Tốc độ khác nhau
  - Định dạng khác nhau
- Tốc độ xử lý chậm hơn nhiều so với CPU và MM
- Cần phải có các khối vào/ra



Thiết bị ngoại vi

- Human readable
  - Screen, printer, keyboard
- Machine readable
  - Monitoring and control
- Communication
  - Modem
  - Network Interface Card (NIC)



2. Chức năng của I/O Module

- Kiểm soát và điều phối (timing)
- Truyền thông với CPU/RAM
- Truyền thông với thiết bị ngoại vi
- Tạo cơ chế đệm dữ liệu (Data Buffering)
- Phát hiện và kiểm soát lỗi

Quy trình hoạt động mô đun I/O

1. CPU kiểm tra trạng thái của thiết bị được kiểm soát bởi mô đun I/O

2. Mô đun I/O trả kết quả trạng thái của thiết bị I/O đó

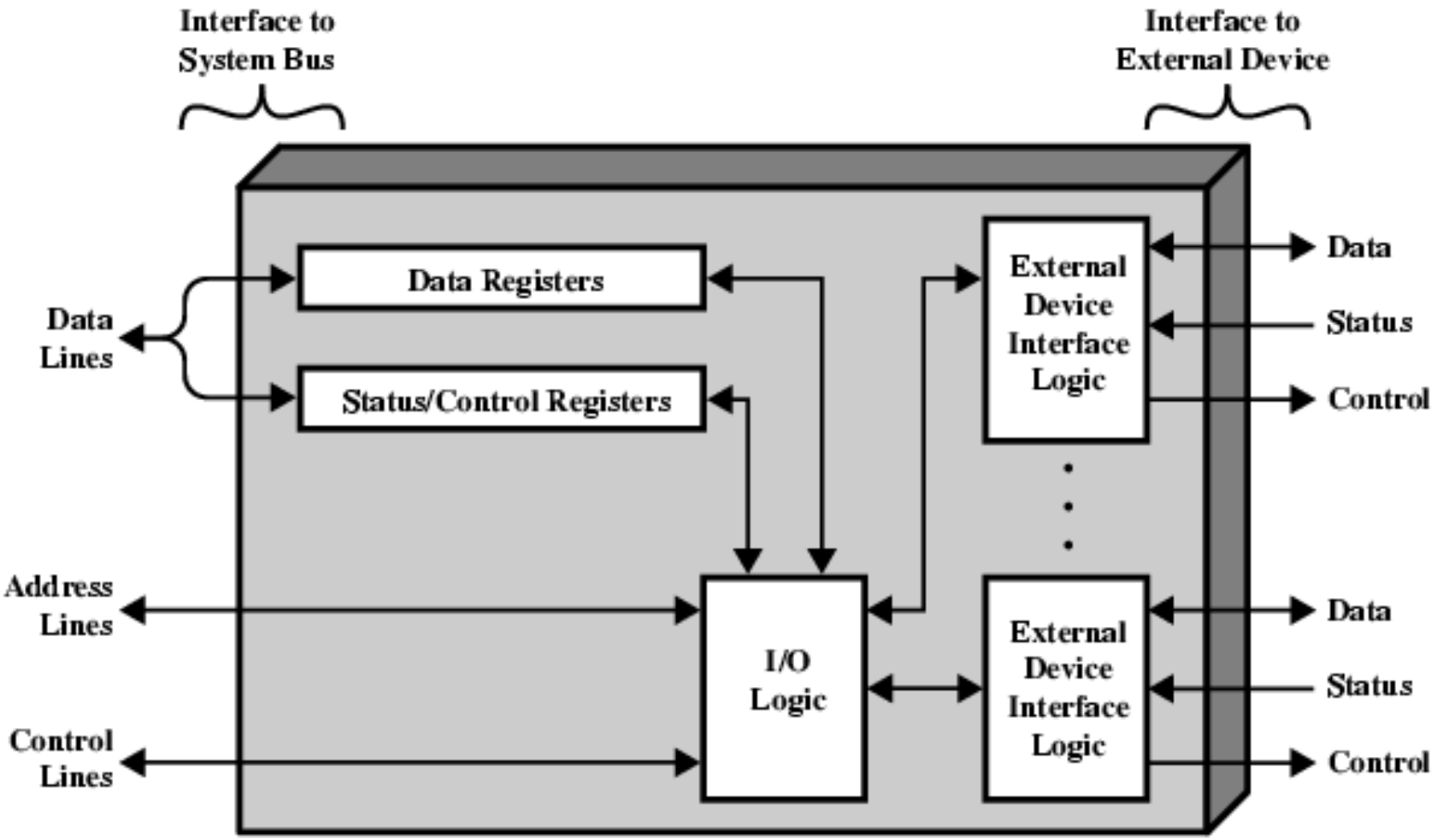
3. Nếu sẵn sàng, CPU tiến hành truyền/nhận dữ liệu

4. Mô đun I/O lấy dữ liệu từ thiết bị

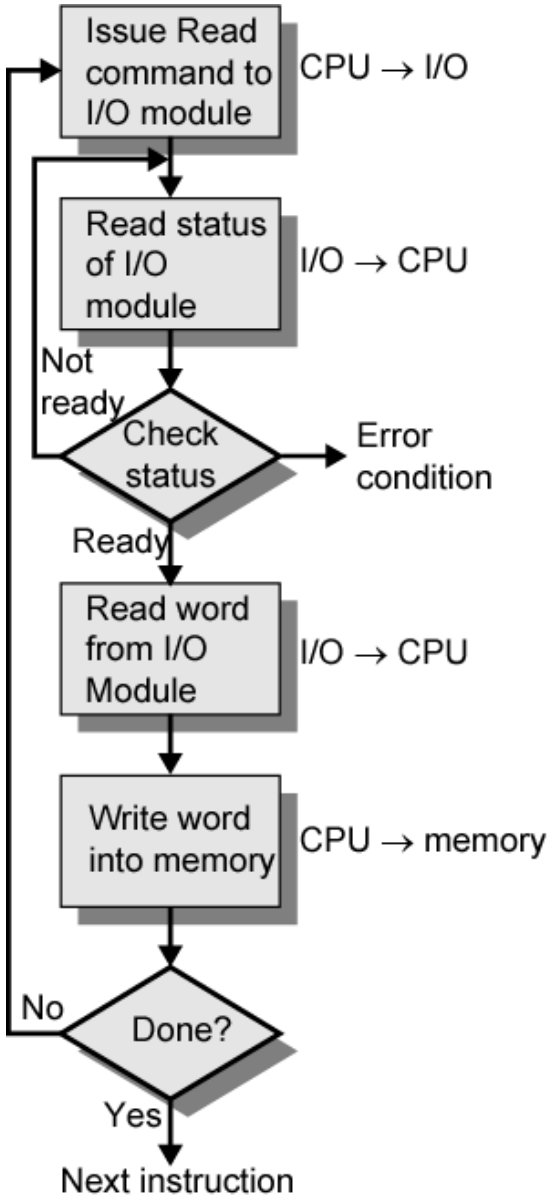
5. Mô đun I/O truyền dữ liệu đến CPU

Ngoài ra còn có thêm những kỹ thuật truyền dữ liệu khác nữa như DMA, etc.

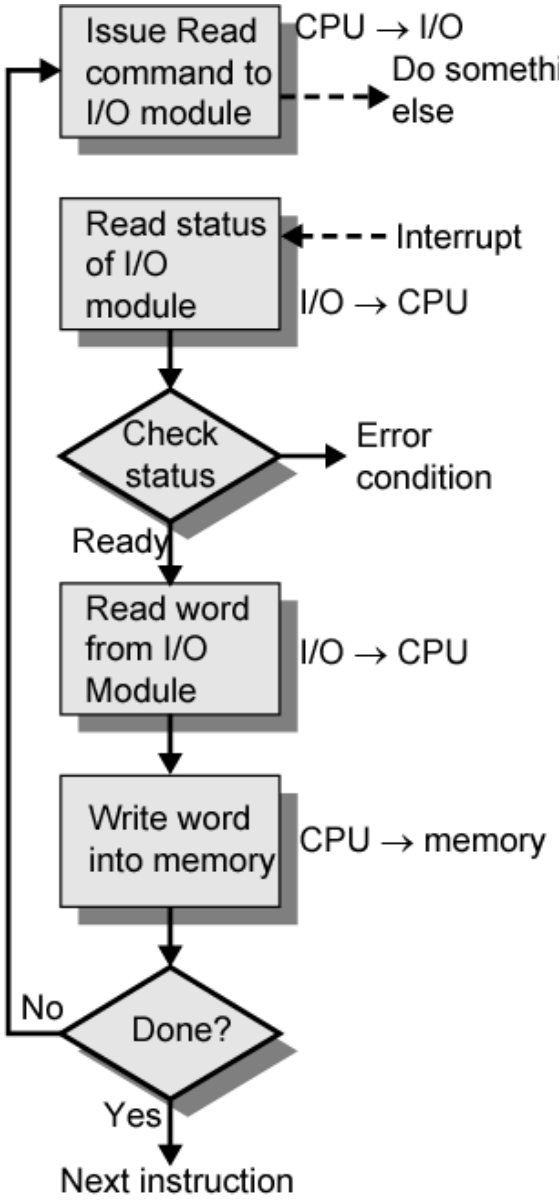
Sơ đồ khối I/O


Interface to System Bus
Interface to External Device
Data Registers
Status/Control Registers
Data Lines
Address Lines
Control Lines
I/O Logic
External Device Interface Logic
External Device Interface Logic
Data
Status
Control
Data
Status
Control

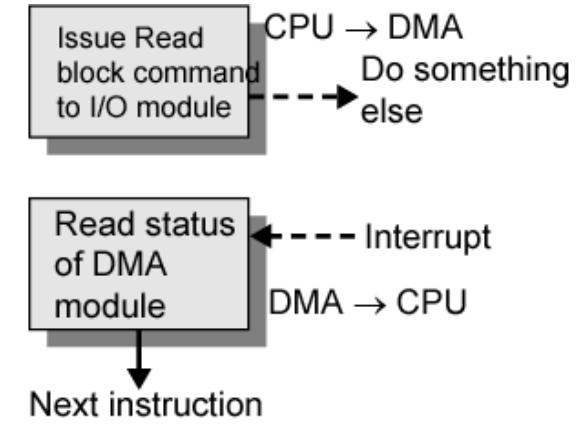
3. Kỹ thuật kiểm soát vào/ra



(a) Programmed I/O



(b) Interrupt-driven I/O



(c) Direct memory access

i. Programmed I/O

- Idea: CPU kiểm soát trực tiếp các thiết bị ngoại vi
  - Phát hiện thay đổi (sensing status)
  - Gửi các lệnh read/write
  - Truyền dữ liệu

➔ CPU phải đợi các I/O module hoàn tất các thao tác ➔ lãng phí tài nguyên CPU

Các bước thực hiện

- CPU requests I/O operation
- I/O module performs operation
- I/O module sets status bits
- CPU checks status bits periodically
- I/O module does not inform CPU directly
- I/O module does not interrupt CPU
- CPU may wait or come back later

I/O Commands

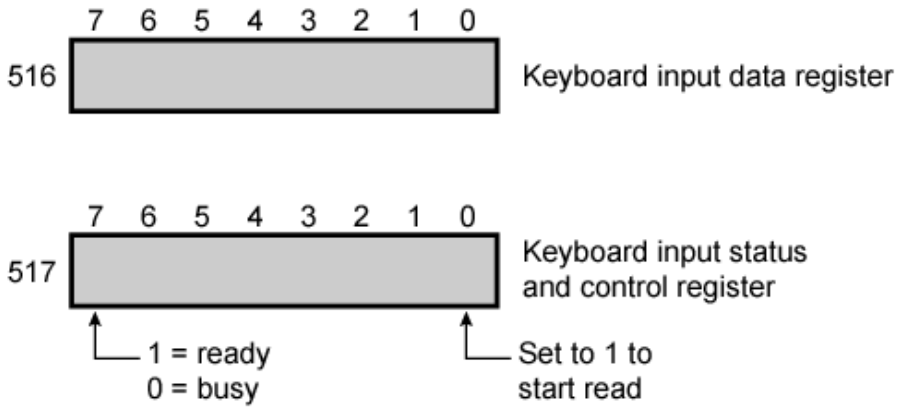
- CPU gửi địa chỉ
  - Xác định bộ vào/ra (& thiết bị nếu >1 mỗi bộ)

- CPU gửi command
  - Control : yêu cầu module thực hiện thao tác
    - Ví dụ: quay đĩa, di chuyển đầu từ, …
  - Test – kiểm tra trạng thái thiết bị
    - Ví dụ: đã được cung cấp nguồn? Có lỗi hay không?...
  - Read/Write
    - Module truyền dữ liệu sử dụng buffer từ/đến thiết bị

Đánh địa chỉ với các thiết bị I/O

- Với kỹ thuật *programmed I/O*, được được truyền giống như việc truy cập bộ nhớ chính (CPU viewpoint)
- Mỗi thiết bị có một định danh duy nhất (port ID)
- Các lệnh từ CPU sẽ chưa định danh này
- Phân loại
  - I/O được ánh xạ vào MM: các thiết bị có địa chỉ nằm trong không gian địa chỉ của MM
    - Các thao tác với I/O tương tự như đọc/ghi bộ nhớ,
    - Không cần lệnh đặc biệt
  - I/O độc lập so với MM: các thiết bị có địa chỉ độc lập so với không gian địa chỉ MM
    - Cần cơ chế liên kết riêng cho I/O
    - Cần có lệnh riêng thao tác với I/O

Minh hoạ



ADDRESS	INSTRUCTION	OPERAND	COMMENT
200	Load AC	"1"	Load accumulator
	Store AC	517	Initiate keyboard read
202	Load AC	517	Get status byte
	Branch if Sign = 0	202	Loop until ready
	Load AC	516	Load data byte

(a) Memory-mapped I/O

ADDRESS	INSTRUCTION	OPERAND	COMMENT
200	Load I/O	5	Initiate keyboard read
201	Test I/O	5	Check for completion
	Branch Not Ready	201	Loop until complete
	In	5	Load data byte

(b) Isolated I/O

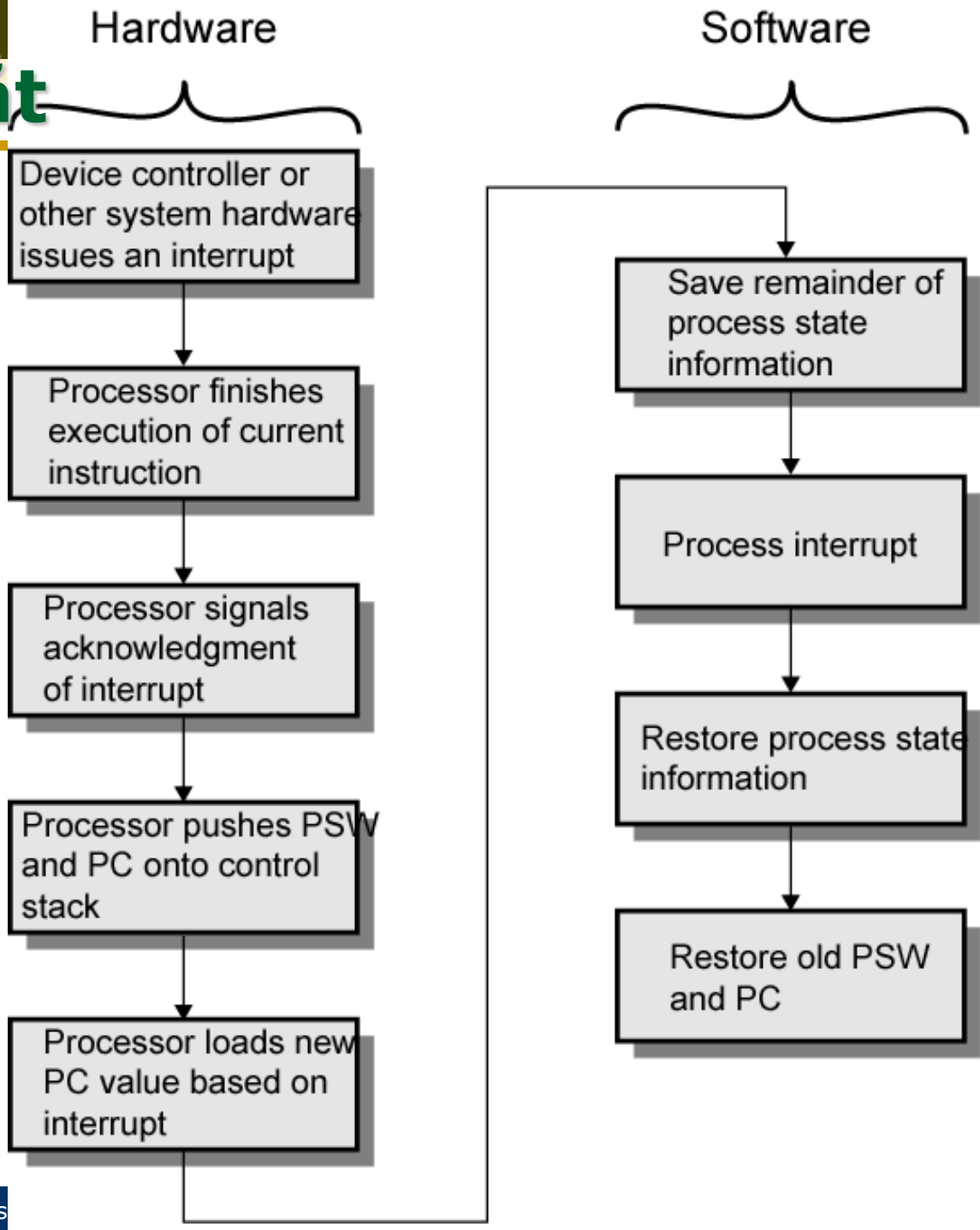
ii. Interrupt Driven I/O

- Tránh được sự lãng phí CPU
- CPU không cần phải kiểm tra định kỳ trạng thái thiết bị
- Bộ vào/ra sinh ngắt khi đã sẵn sàng

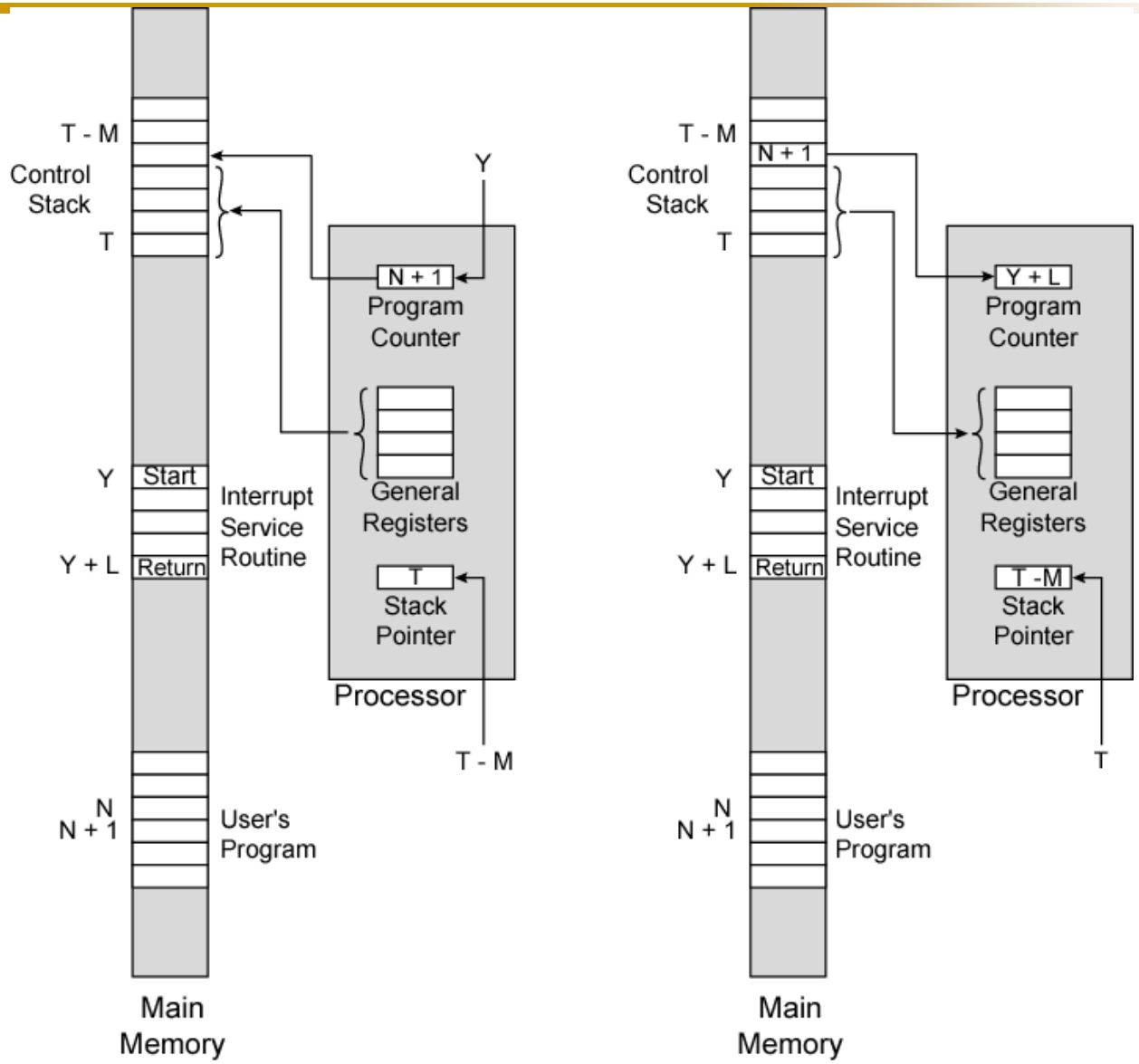
Các bước thực hiện

- CPU issues read command
- I/O module gets data from peripheral whilst CPU does other work
- I/O module interrupts CPU
- CPU requests data
- I/O module transfers data

Sơ đồ xử lý ngắt



Thay đổi trong MM và Registers khi xử lý ngắt



Vấn đề cần quan tâm

- Làm thế nào để xác định được module nào yêu cầu ngắt?
- Việc xử lý đa ngắt được thực hiện như thế nào?

Xác định module ngắt

- Sử dụng nhiều đường ngắt (multiple interrupt lines)
  - Số pins của CPU là giới hạn ➔ số lượng thiết bị cũng bị giới hạn
  - Không được sử dụng trong thực tế
- Software poll
  - Sử dụng chương trình con (routine) làm nhiệm vụ xác định thiết bị yêu cầu ngắt mỗi khi CPU nhận được yêu cầu ngắt
  - ➔ mất nhiều chu kỳ cho việc xác định thiết bị yêu cầu ngắt
- Daisy Chain (hardware poll): dùng chung đường interrupt
  - CPU gửi lại tín hiệu *Interrupt Acknowledge* khi nhận được yêu cầu ngắt và sẽ được truyền lần lượt các thiết bị cho đến khi đến được thiết bị yêu cầu
  - Thiết bị yêu cầu ngắt sẽ gửi dữ liệu trên bus dưới dạng vector chứa địa chỉ
  - CPU sử dụng vector để định danh thiết bị
- Bus Master
  - I/O module phải đăng ký sử dụng bus trước khi gửi ngắt
  - Được sử dụng trong PCI & SCSI

Đa ngắt

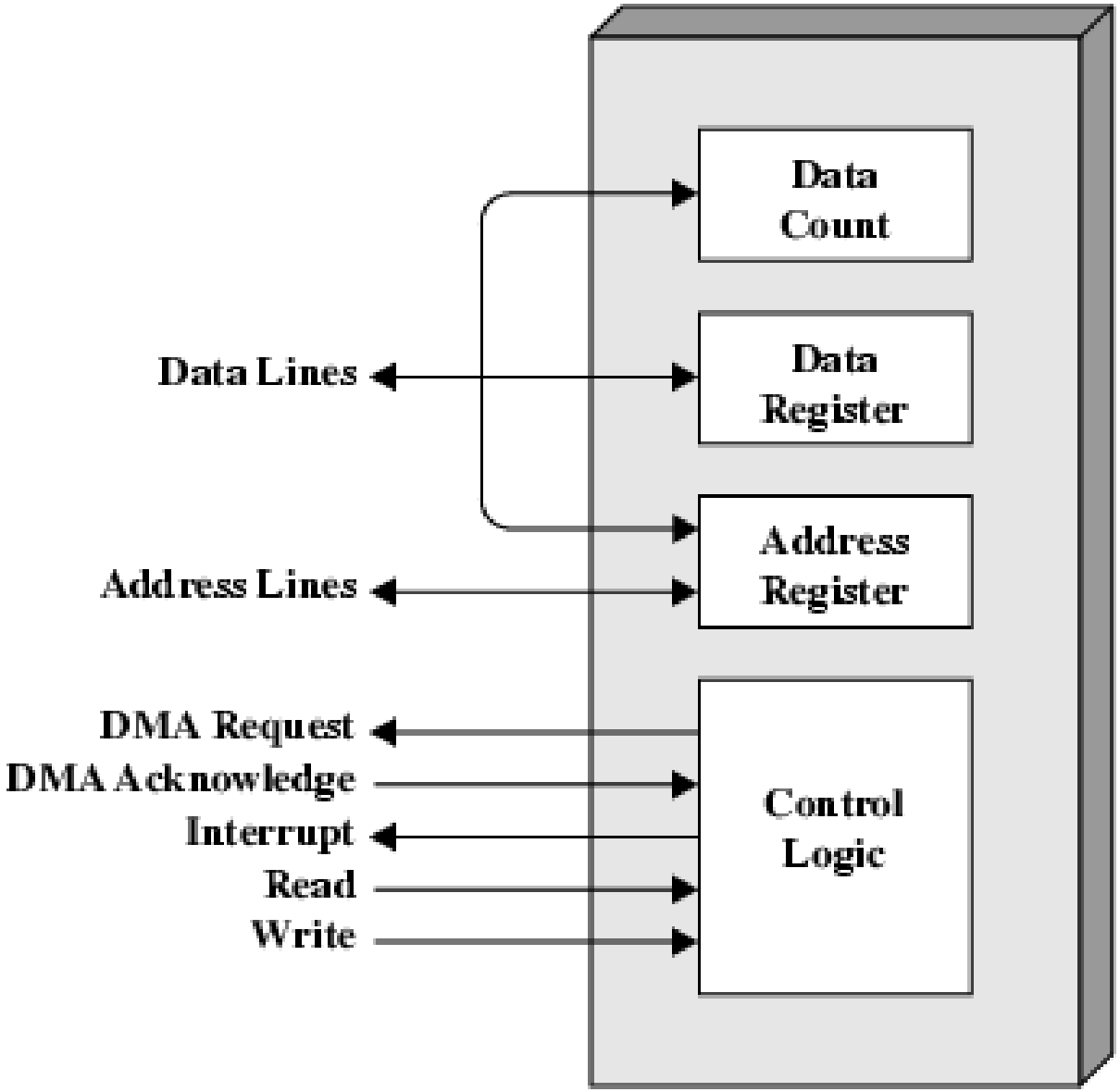
- Multiple lines: Mỗi đường ngắt được gán một độ ưu tiên (priority), đường ngắt có độ ưu tiên cao hơn có thể ngắt thấp hơn
- Software polling/daisy chain: đánh độ ưu tiên thứ tự poll cho các modules
- Bus mastering: chỉ duy nhất thiết bị có quyền master được xử lý ngắt

iii. Direct Memory Access

- Cả Interrupt-driven và programmed I/O đều cần sự "can thiệp" của CPU
  - ➔ Tốc độ truyền giữa các thiết bị sẽ bị giới hạn
  - ➔ CPU có thể bị quá tải

- DMA là kỹ thuật sẽ cho phép giải quyết được những nhược điểm nêu trên khi truyền khối lượng dữ liệu lớn
  - Sử dụng thêm module (hardware) DMA Controller
  - DMA controller giữ vai trò điều phối vào/ra

Sơ đồ bộ DMA điển hình

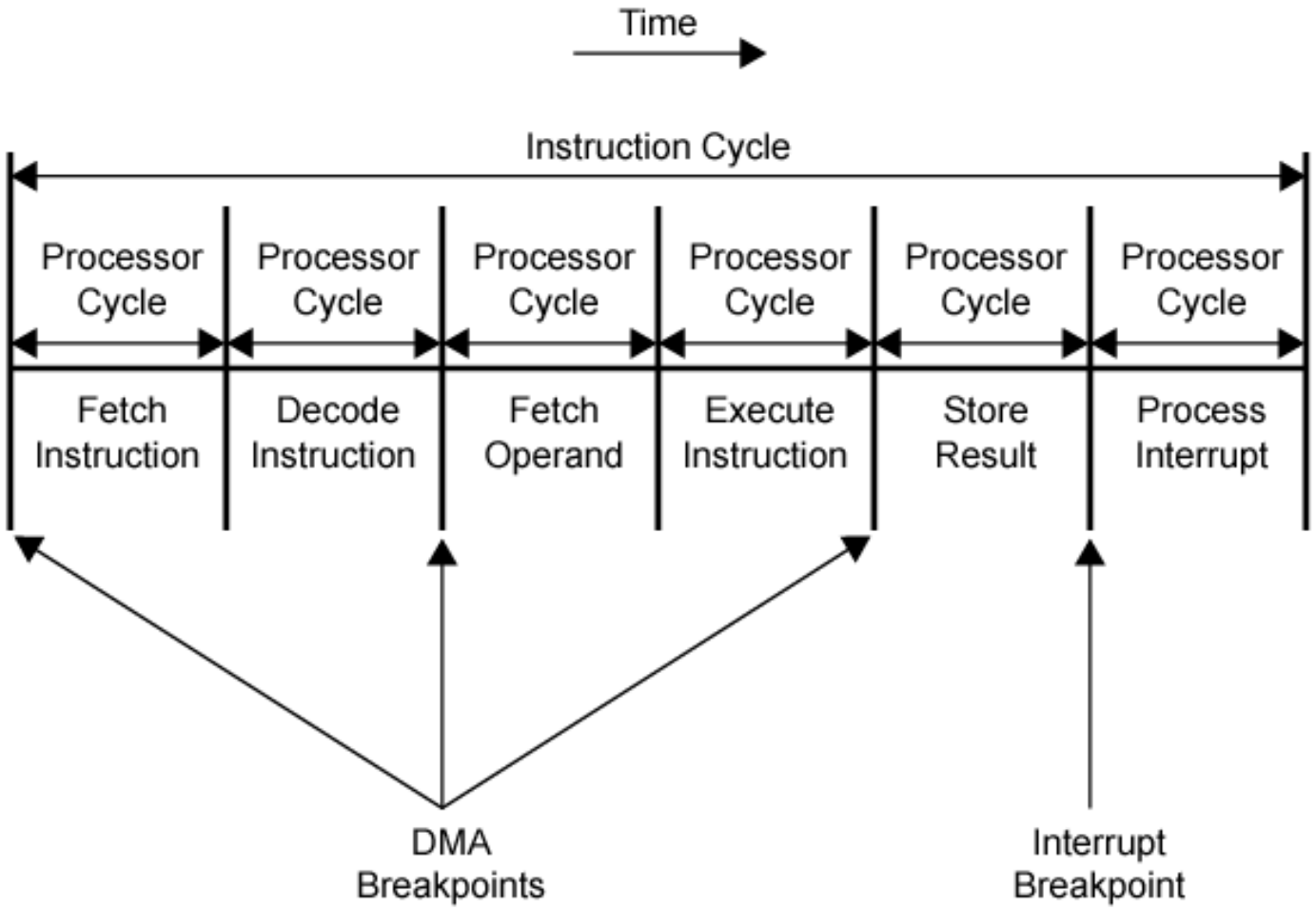


Quy trình hoạt động

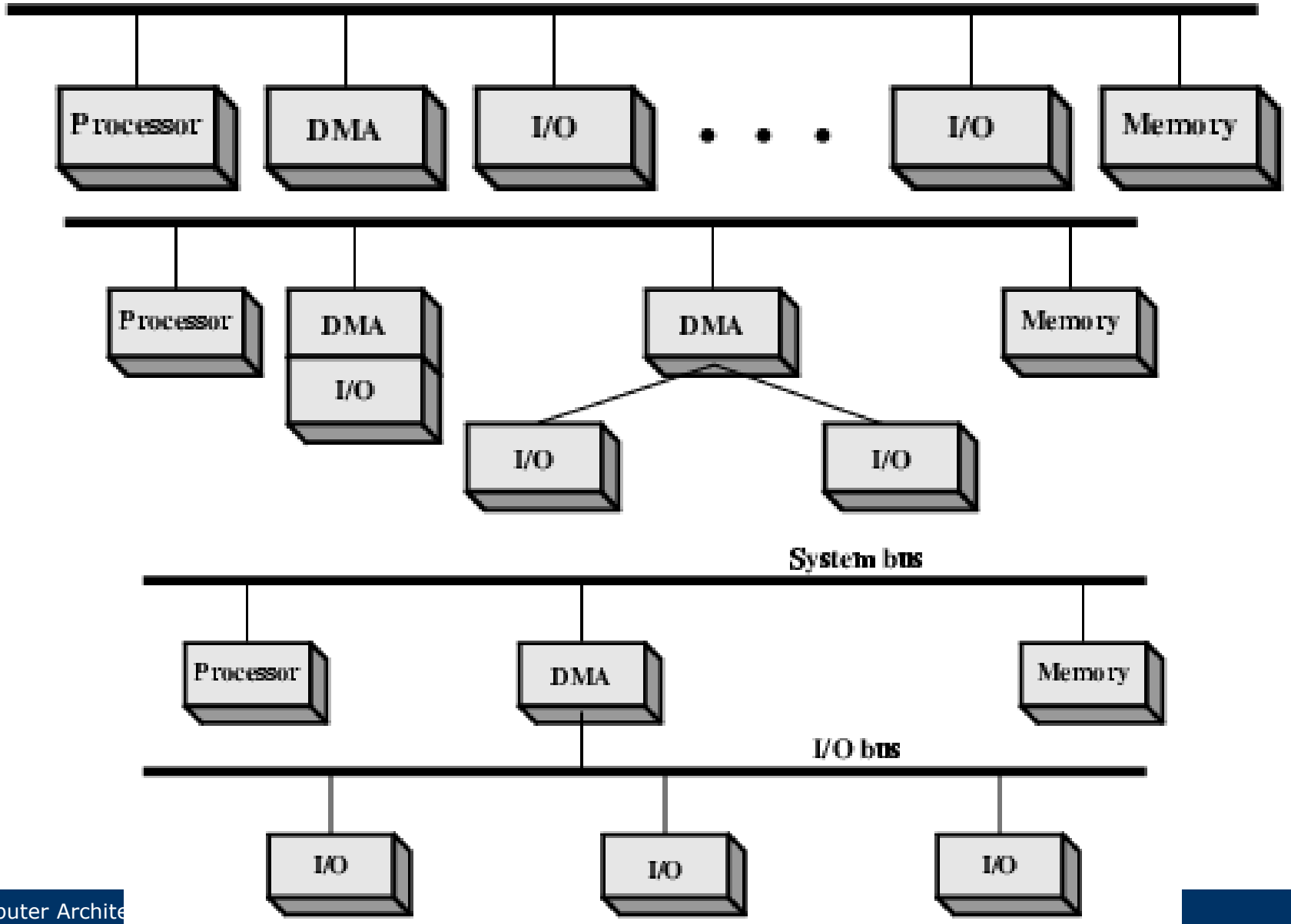
- CPU gửi lệnh đến DMA controller:
  - Read/Write
  - Device address
  - Starting address of memory block for data
  - Amount of data to be transferred
- CPU tiến hành các lệnh khác
- DMA controller đảm nhiệm điều phối truyền dữ liệu
- DMA controller gửi interrupt khi kết thúc truyền

DMA truyền với kỹ thuật Cycle Stealing

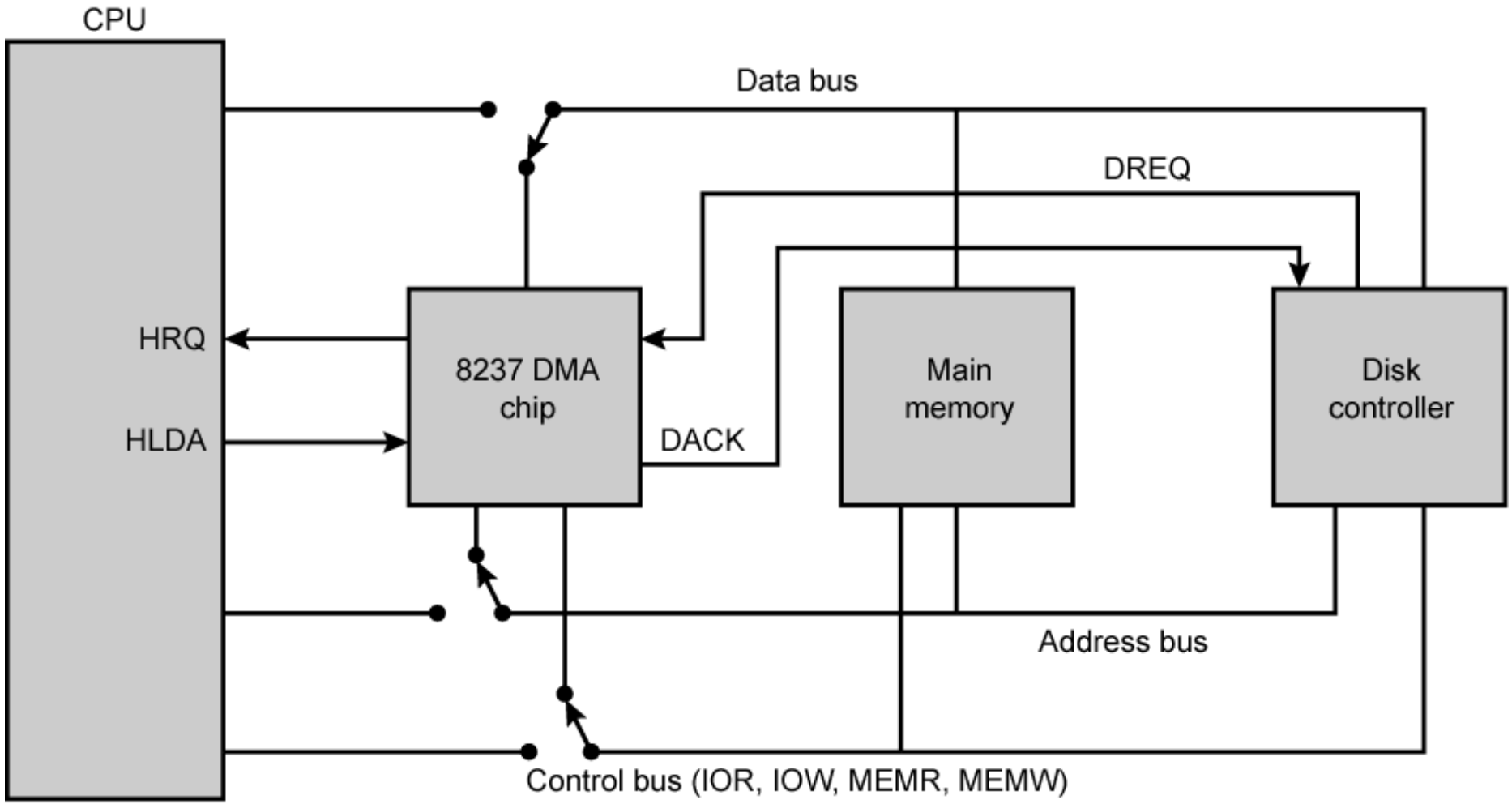
- DMA controller tận dụng các cycles mà CPU không sử dụng data bus để truyền dữ liệu (one word/cycle)



Cấu hình DMA


Processor
DMA
I/O
I/O
Memory
Processor
DMA
I/O
DMA
Memory
I/O
I/O
System bus
Processor
DMA
Memory
I/O bus
I/O
I/O
I/O

Ví dụ: Intel 8237A DMA Controller



DACK = DMA acknowledge
DREQ = DMA request
HLDA = HOLD acknowledge
HRQ = HOLD request

iv. I/O Channels

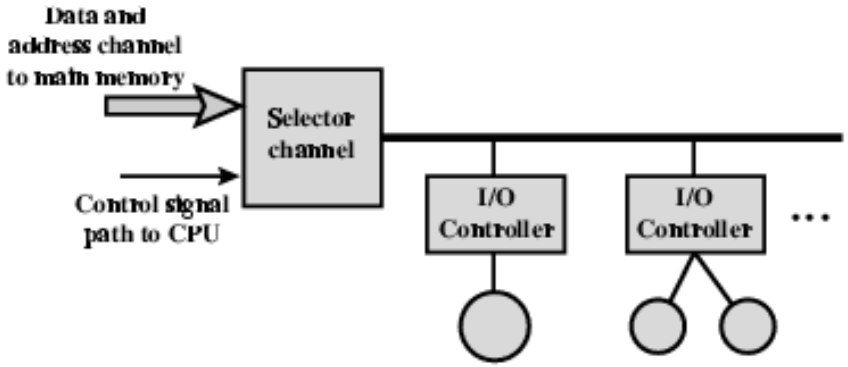
- Thiết bị I/O càng ngày càng phức tạp, có xu hướng tích hợp luôn bộ vi xử lý riêng
  - VD: Card đồ hoạ (GPU), …

- CPU gửi lệnh đến I/O controller và giao toàn quyền việc truyền dữ liệu

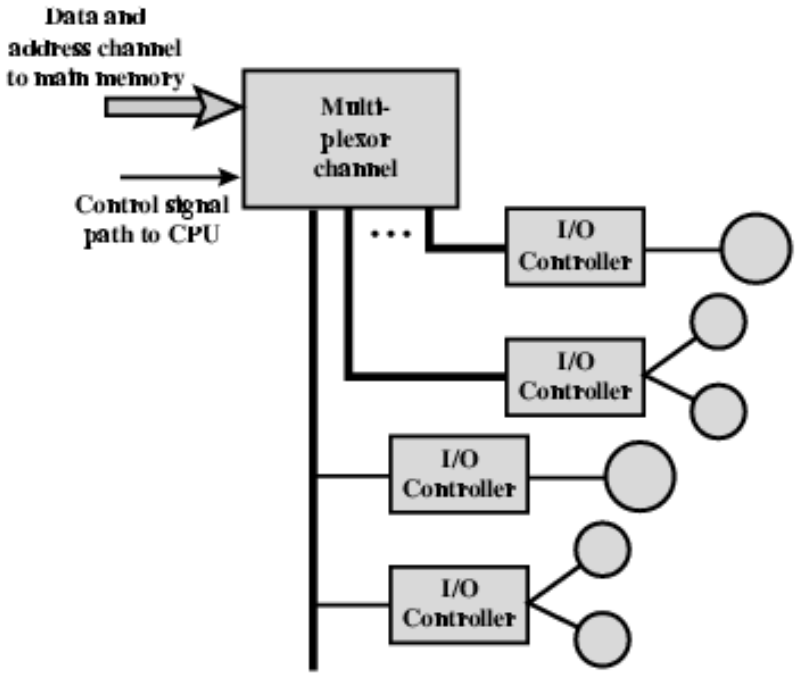
➔ Cải thiện được tốc độ truyền dữ liệu I/O

- I/O channels: thể hiện mở rộng của kỹ thuật DMA

Kiến trúc kênh vào/ra



(a) Selector



(b) Multiplexor

4. Interfacing

- Giao tiếp giữa các thiết bị:
  - Serial <> Parallel
  - Dedicated processor/memory/buses?

- Ví dụ với FireWire, InfiniBand

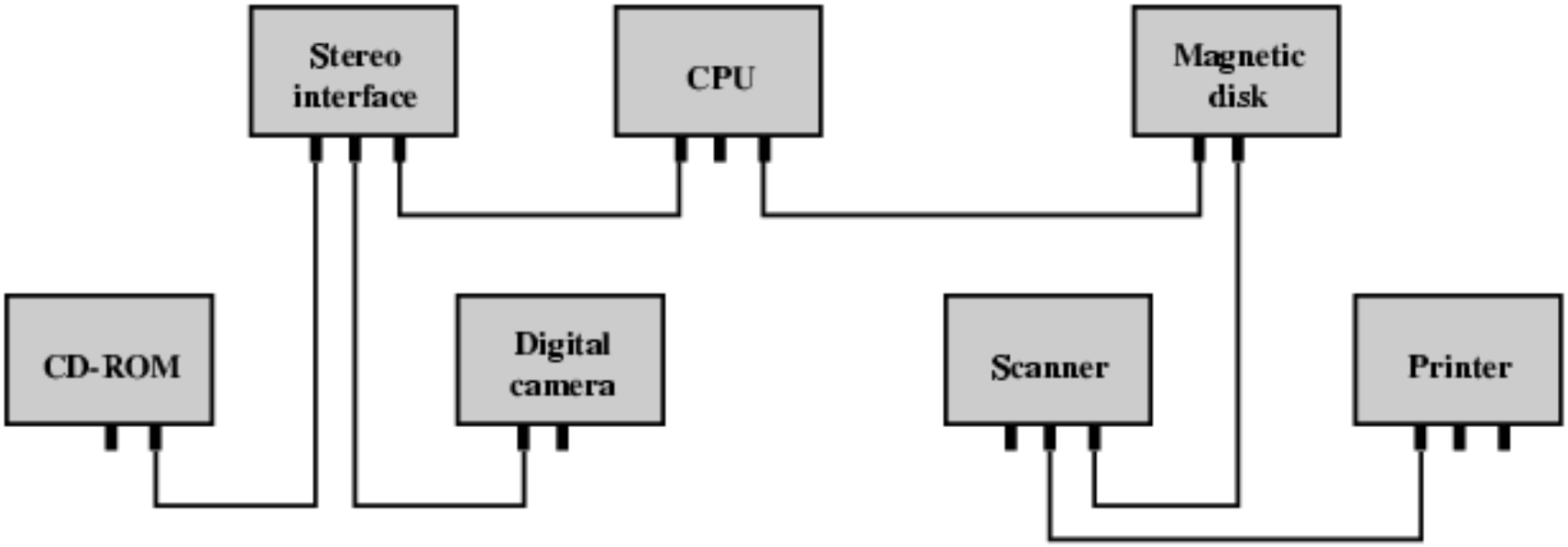
IEEE 1394 FireWire

- Sử dụng bus tuần tự tốc độ cao
- Chi phí thấp, dễ cài đặt
- Thường được sử dụng kết nối với các thiết bị gia đình digital cameras, VCRs, TV, …

FireWire Configuration

- Daisy chain
- Tối đa 63 devices với mỗi cổng
  - Really 64 of which one is the interface itself
- Up to 1022 buses can be connected with bridges
- Automatic configuration
- No bus terminators
- May be tree structure

Simple FireWire Configuration



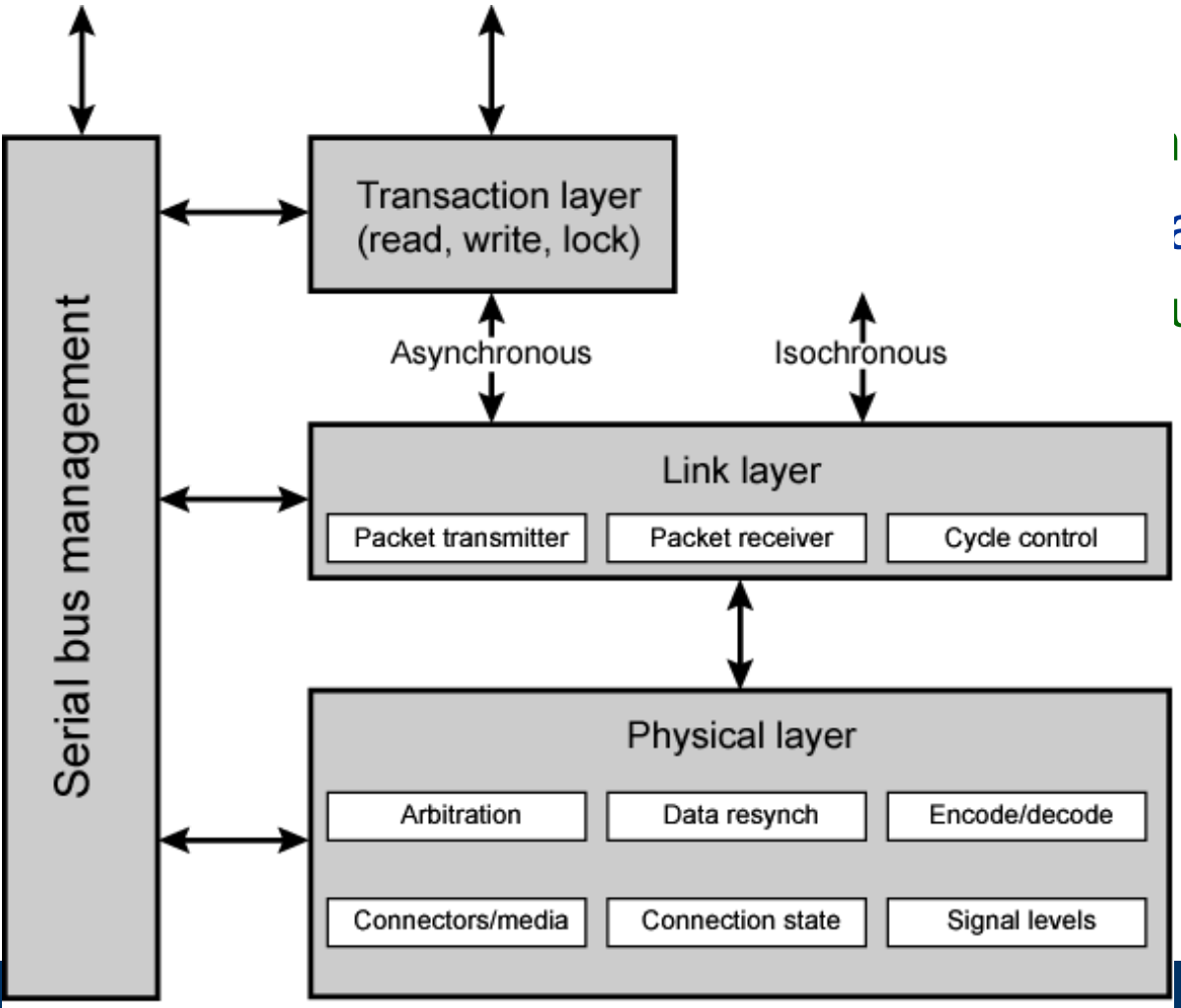
FireWire 3 Layer Stack

- Physical
  - Transmission medium, electrical and signaling characteristics

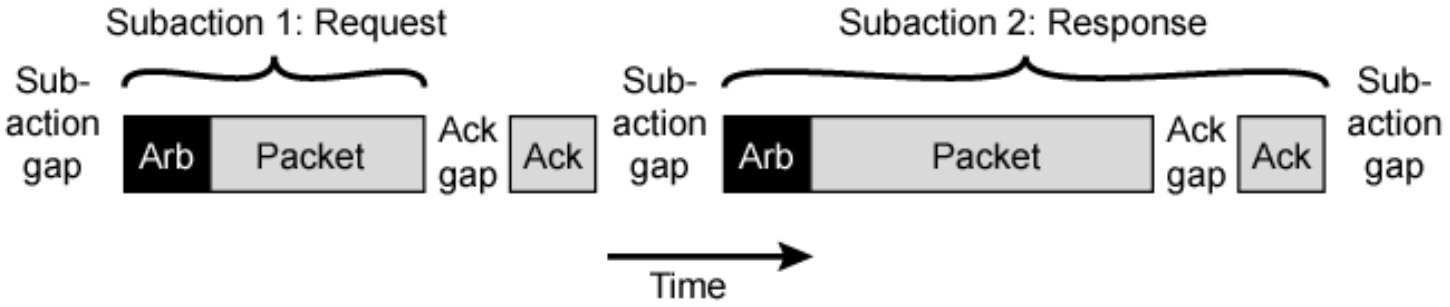
ısmission of data in packets

action

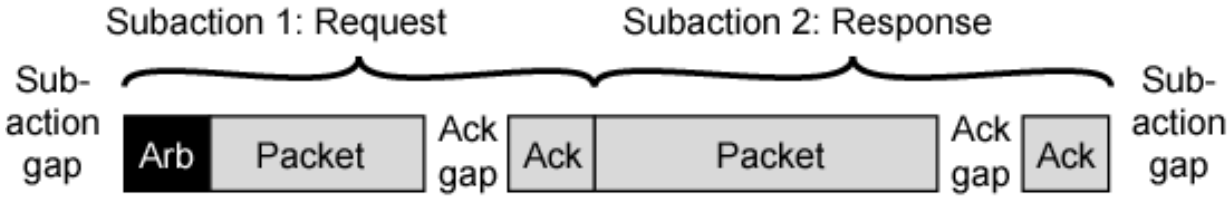
uest-response protocol



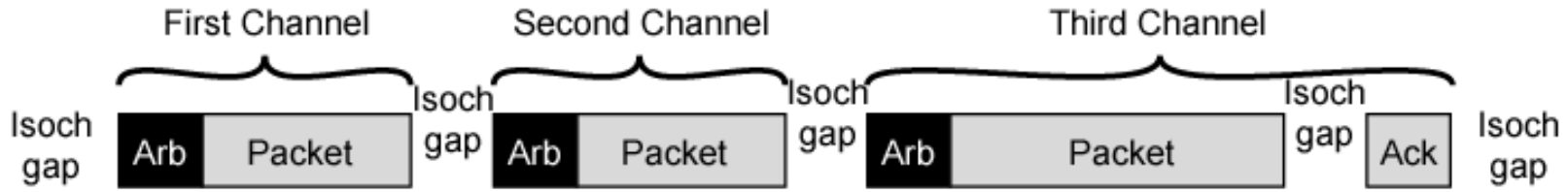
FireWire Subactions



(a) Example asynchronous subaction



(b) Concatenated asynchronous subactions



(c) Example isochronous subactions

Kết luận

- Kiến trúc vào/ra cho phép máy tính "giao tiếp" được với thế giới bên ngoài
- Có 3 kỹ thuật vào/ra chính: programmed I/O, interrupt driven I/O và DMA
- Giao tiếp với các thiết bị ngoài vi được thực hiện theo các chuẩn: Firewire 1394, USB, …