



## KIẾN TRÚC MÁY TÍNH



## Chương 3

Tổng quan về chức năng và kết nối trong máy tính

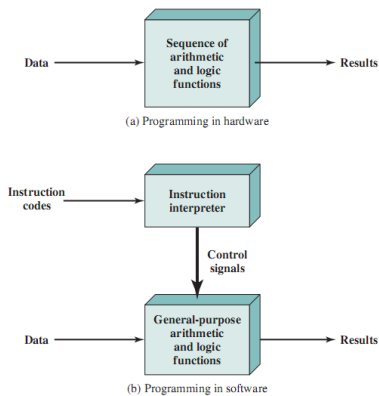
### Chương 3. Tổng quan về chức năng và kết nối trong máy tính

- 3.1 Các thành phần của máy tính
- 3.2 Chức năng của máy tính
- 3.3 Cấu trúc các kết nối
- 3.4 Hệ thống bus
- 3.5 Kết nối điểm-điểm (Point-To-Point)
- 3.6 PCI Express

## 1. Các thành phần của máy tính

- Máy tính hiện đại ngày nay được thiết kế dựa trên ý tưởng của von Neumann tại Viện nghiên cứu Princeton
- Kiến trúc này được gọi là kiến trúc Von Neumann và có 3 điểm chính
  - Dữ liệu và lệnh được lưu trữ trên cùng một bộ nhớ đọc-viết
  - Nội dung của dữ liệu được định vị theo vị trí (đánh địa chỉ) mà không phụ thuộc vào kiểu dữ liệu.
  - Các lệnh được thực thi một cách tuần tự (trừ trong một số trường hợp yêu cầu gọi đến câu lệnh khác).
- Lắp ghép phần cứng (Hardwired program)
  - Là kết quả của quá trình kết nối các thành phần khác nhau tạo nên một cấu hình mong muốn

Tiếp cận  
phần cứng  
và phần  
mềm



### Phần mềm (Software)

- Một chuỗi các mã lệnh hoặc chỉ thị
- Một bộ phận của phần cứng làm chức năng phiên dịch từng lệnh và tạo ra tín hiệu điều khiển
- Đưa ra chuỗi mã mới cho mỗi chương trình khác nhau thay vì đi lại dây nối phần cứng

### Các thành phần chính:

- CPU
  - Bộ biên dịch lệnh
  - Thiết bị tính toán số học và logic đa nhiệm
- Các thành phần xuất nhập I/O
  - Module nhập
    - Bao gồm các thành phần cơ bản cho việc nhận vào dữ liệu và lệnh; chuyển đổi chúng thành dạng tín hiệu sử dụng bên trong hệ thống
  - Module xuất
    - Công cụ để hiện thị kết quả

### Phần mềm

Các thành  
phần



- Chương trình không phải lúc nào cũng thực hiện theo thứ tự; nó có thể nhảy đến lệnh khác (ví dụ, câu lệnh JUMP trong IAS)

- Các phép toán trên dữ liệu có thể yêu cầu truy cập nhiều hơn một số

→ Module lưu trữ tạm thời cả lệnh và dữ liệu: **Bộ nhớ bộ nhớ chính (main memory)**

- Thanh ghi **MAR (Memory Address Register)** chứa địa chỉ trong bộ nhớ cho lần đọc hoặc ghi tiếp theo

- Thanh ghi **MBR (Memory Buffer Register)** dữ liệu được ghi vào bộ nhớ hoặc nhận dữ liệu được đọc từ bộ nhớ.

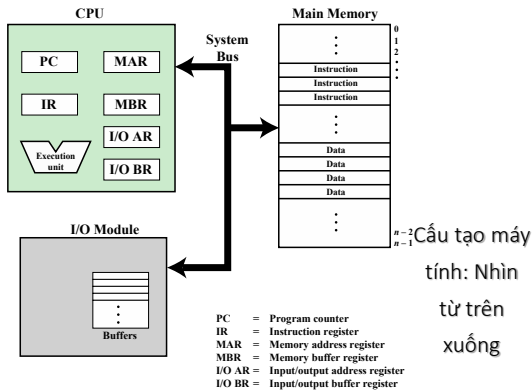
- Thanh ghi **I/OAR (I/O Address Register)** xác định một thiết bị I/O cụ thể.

- Thanh ghi **I/O BR (I/O Buffer Register)** được sử dụng để trao đổi dữ liệu giữa một mô-đun I/O và CPU.

**Bộ nhớ  
MEMORY**

MAR

MBR

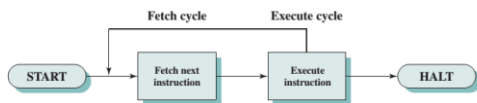


## 2. Chức năng của máy tính

### - Truy xuất và thực thi chương trình

2 bước:

- Truy xuất (Fetch)
- Thực thi (Execute)



Chu kỳ lệnh cơ bản

## Chu kỳ truy xuất

- Vào đầu mỗi chu kỳ lệnh, bộ xử lý truy xuất một lệnh từ bộ nhớ
- Thanh ghi PC (Program Counter) giữ địa chỉ của lệnh được truy xuất tiếp theo
- Bộ xử lý tăng PC sau mỗi lần truy xuất lệnh do đó nó sẽ truy xuất được lệnh tiếp theo vào lần sau.
- Lệnh vừa được truy xuất được tải vào thanh ghi IR (Instruction Register)
- Bộ xử lý biên dịch lệnh và thi hành những hành động cần thiết




---

---

---

---

---

---

---

---

## Phân loại các hoạt động (action)

Bộ xử lý – bộ nhớ	<ul style="list-style-type: none"> <li>• Dữ liệu truyền từ bộ xử lý đến bộ nhớ hoặc ngược lại</li> </ul>
Bộ xử lý – I/O	<ul style="list-style-type: none"> <li>• Dữ liệu truyền đến/đi từ thiết bị ngoại vi bằng cách truyền thông tin giữa bộ xử lý và module I/O</li> </ul>
Xử lý dữ liệu	<ul style="list-style-type: none"> <li>• Bộ xử lý có thể thực hiện một số phép toán số học hoặc logic trên dữ liệu</li> </ul>
Điều khiển	<ul style="list-style-type: none"> <li>• Đưa ra lệnh chỉ rõ thứ tự thực hiện các lệnh bị thay đổi</li> </ul>

---

---

---

---

---

---

---

---



(a) Instruction format



(b) Integer format

Program Counter (PC) = Address of instruction  
 Instruction Register (IR) = Instruction being executed  
 Accumulator (AC) = Temporary storage

(c) Internal CPU registers

0001 = Load AC from Memory  
 0010 = Store AC to Memory  
 0101 = Add to AC from Memory

(d) Partial list of opcodes

Đặc tính của  
Máy giả thiết

---

---

---

---

---

---

---

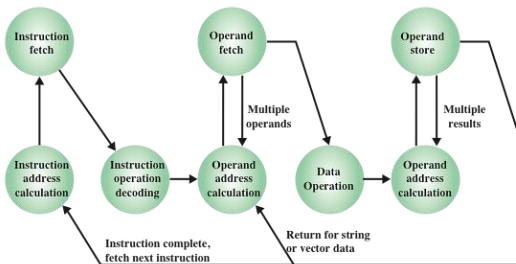
---

Ví dụ  
thi hành  
chương trình

Step 1		Step 2	
Memory	CPU Registers	Memory	CPU Registers
300 1 9 4 0	3 0 0 PC	300 1 9 4 0	3 0 1 PC
301 5 9 4 1	0 0 0 AC	301 5 9 4 1	0 0 0 AC
302 2 9 4 1	1 9 4 IR	302 2 9 4 1	1 9 4 IR
940 0 0 0 3		940 0 0 0 3	
941 0 0 0 2		941 0 0 0 2	
Step 3		Step 4	
Memory	CPU Registers	Memory	CPU Registers
300 1 9 4 0	3 0 1 PC	300 1 9 4 0	3 0 2 PC
301 5 9 4 1	0 0 0 AC	301 5 9 4 1	0 0 0 AC
302 2 9 4 1	5 9 4 IR	302 2 9 4 1	5 9 4 IR
940 0 0 0 3		940 0 0 0 3	
941 0 0 0 2		941 0 0 0 2	
Step 5		Step 6	
Memory	CPU Registers	Memory	CPU Registers
300 1 9 4 0	3 0 2 PC	300 1 9 4 0	3 0 3 PC
301 5 9 4 1	0 0 0 AC	301 5 9 4 1	0 0 0 AC
302 2 9 4 1	2 9 4 IR	302 2 9 4 1	2 9 4 IR
940 0 0 0 3		940 0 0 0 3	
941 0 0 0 2		941 0 0 0 5	

(contents of memory and registers in hexadecimal)

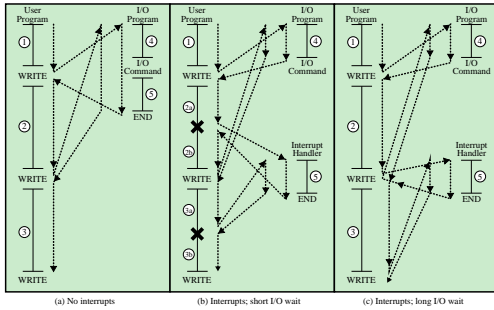
Sơ đồ trạng thái chu kỳ lệnh



Phân loại các gián đoạn

- **Gián đoạn chương trình:** Gây ra bởi lỗi thi hành lệnh, ví dụ như tràn số học, lỗi chia cho 0, cố tình thực hiện các lệnh máy không hợp lệ, hoặc tham chiếu ngoài phạm vi bộ nhớ mà người sử dụng được phép
- **Gián đoạn định thời:** Gây ra bởi đồng hồ nằm trong bộ xử lý. Nó cho phép hệ điều hành thực hiện các chức năng cơ bản nhất định.
- **Gián đoạn I/O:** Gây ra bởi bộ điều khiển I/O, để báo hiệu hoàn thành một thao tác, yêu cầu dịch vụ từ bộ xử lý, hoặc báo hiệu các trường hợp lỗi
- **Gián đoạn lỗi phần cứng:** Gây ra bởi một số lỗi như lỗi nguồn hay lỗi bộ nhớ

## Điều khiển dòng chương trình



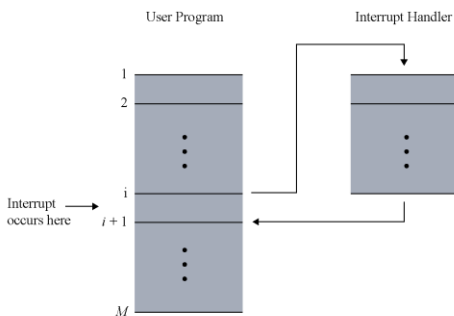
✕ = interrupt occurs during course of execution of user program

Figure 3.7 Program Flow of Control Without and With Interrupts

## Chu kỳ gián đoạn

- Thêm vào chu kỳ lệnh
- bộ vi xử lý kiểm tra để ngắt
  - Được chỉ thị bằng tín hiệu gián đoạn
- Nếu không có gián đoạn, truy xuất lệnh tiếp theo
- Nếu gián đoạn đang chờ xử lý:
  - Tạm ngưng thực hiện chương trình hiện tại
  - Lưu ngữ cảnh
  - Đặt PC về địa chỉ bắt đầu của trình xử lý gián đoạn
  - Ngắt quá trình
  - Khởi phục ngữ cảnh và tiếp tục chương trình gián đoạn

## Truyền điều khiển qua các gián đoạn



Chu kỳ lệnh có gián đoạn

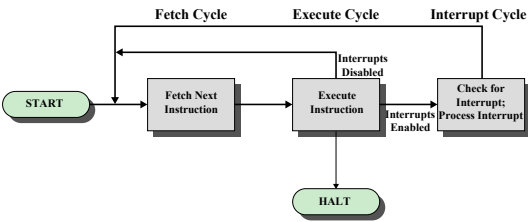
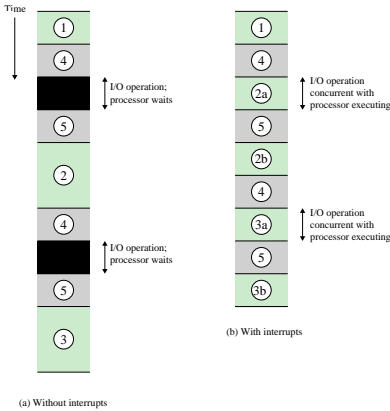
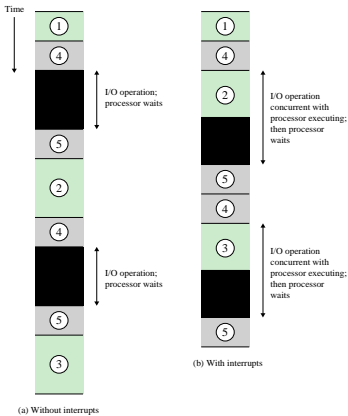


Figure 3.9 Instruction Cycle with Interrupts



Định thời  
chương  
trình:  
Đợi I/O ngắn



Định thời  
chương trình:  
Đợi I/O dài

# Sơ đồ trạng thái chu kỳ lệnh Cổ gián đoạn

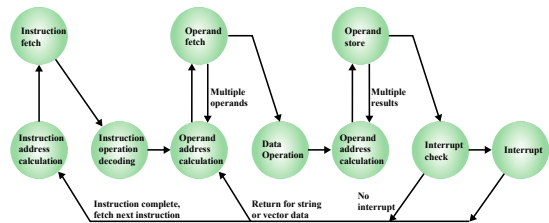


Figure 3.12 Instruction Cycle State Diagram, With Interrupts

---

---

---

---

---

---

---

---

## Đa gián đoạn

- Tắt ngắt
  - Bộ vi xử lý sẽ bỏ qua các ngắt khác trong khi xử lý một gián đoạn
  - Ngắt vẫn còn đang chờ xử lý và được kiểm tra sau khi gián đoạn đầu tiên đã được xử lý
  - Ngắt được xử lý theo trình tự khi chúng xảy ra
- Xác định ưu tiên
  - Các ngắt ưu tiên thấp có thể bị gián đoạn bởi các ngắt ưu tiên cao hơn
  - Khi ngắt ưu tiên cao hơn đã được xử lý, bộ xử lý quay trở lại ngắt trước

---

---

---

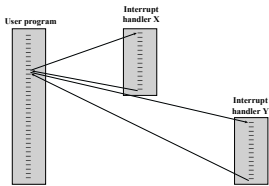
---

---

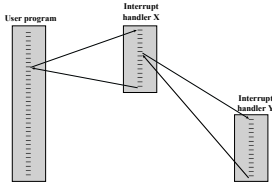
---

---

---



(a) Sequential interrupt processing



(b) Nested interrupt processing

Truyền tín hiệu điều khiển

Đa gián đoạn

---

---

---

---

---

---

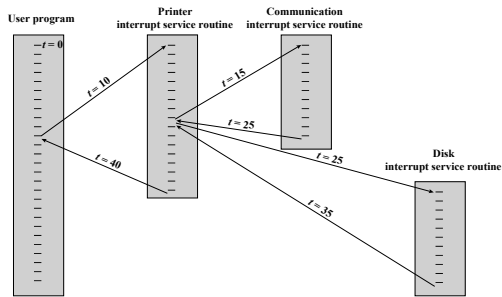
---

---



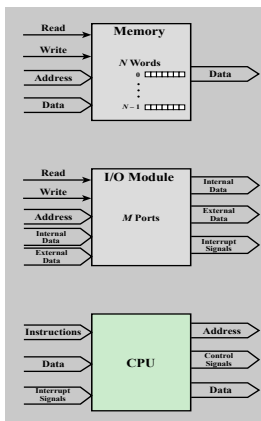
## Trình tự thời gian của đa gián đoạn

Ví dụ

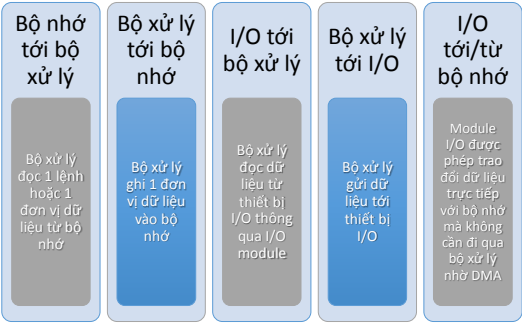


## Chức năng I/O

- Module I/O có thể chuyển dữ liệu trực tiếp với bộ xử lý
- Bộ xử lý có thể đọc dữ liệu từ hoặc ghi dữ liệu lên module I/O
  - Bộ xử lý xác định thiết bị nào được điều khiển bởi module I/O nào
  - I/O đưa ra lệnh chứ không phải là bộ nhớ đưa ra lệnh
- Trong một số trường hợp, cần thiết cho phép I/O trao đổi trực tiếp với bộ nhớ
  - Bộ xử lý cấp cho module I/O quyền đọc/ghi vào bộ nhớ do đó việc truyền tin giữa module I/O và bộ nhớ có thể diễn ra trực tiếp mà không cần thông qua bộ xử lý
  - Module I/O đưa ra yêu cầu đọc/ghi tới bộ nhớ và giải phóng nhiệm vụ chuyển dữ liệu cho bộ nhớ
  - Được gọi là truy cập bộ nhớ trực tiếp (Direct Memory Access - DMA)



Cấu trúc kết nối hỗ trợ các hình thức truyền sau:



---

---

---

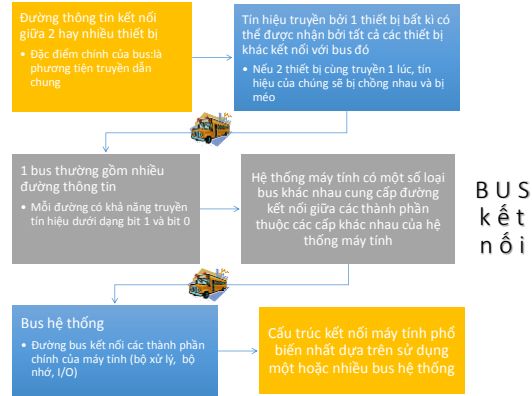
---

---

---

---

---



---

---

---

---

---

---

---

---

### Bus dữ liệu

- Bus dữ liệu là đường kết nối dùng để truyền dữ liệu giữa các module hệ thống
- Bao gồm 32, 64, 128 đường hay nhiều hơn
- Số lượng đường nối được xem là *độ rộng* của bus dữ liệu
- Số lượng đường nối quyết định bao nhiêu bit có thể truyền đi cùng một lúc
- Độ rộng bus dữ liệu là yếu tố chính quyết định hiệu suất toàn hệ thống



---

---

---

---

---

---

---

---

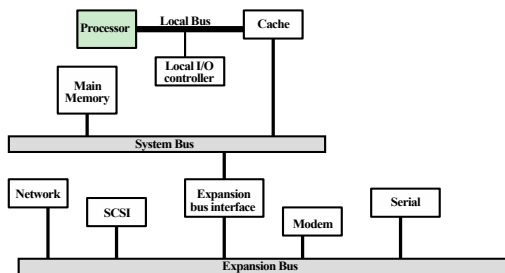
- Được sử dụng để xác định địa chỉ nguồn/đích của dữ liệu trên bus dữ liệu.
  - Nếu bộ xử lý muốn đọc 1 word từ bộ nhớ, nó sẽ địa chỉ của word đó lên đường bus địa chỉ.
- Độ rộng bus xác định đường lượng nhớ tối đa của hệ thống
- Cũng được sử dụng để xác định cổng vào/ra.
  - Các bit báo cáo được sử dụng để lựa chọn module cụ thể trên con bit báo thức dùng để chọn vị trí bit nhớ hoặc cổng vào/ra trong module.

- Được sử dụng để điều khiển việc truy nhập và sử dụng dữ liệu và bus địa chỉ.
- Bởi vì dữ liệu và bus địa chỉ được chia sẻ cho tất cả các thành phần nên cần phải có một công cụ kiểm soát việc sử dụng chúng.
- Các tín hiệu điều khiển truyền cả thông tin lệnh và định thời giữa các mô đun hệ thống.
- Tín hiệu **định thời** xác định tính hợp lệ của dữ liệu và thông tin địa chỉ.
- Tín hiệu lệnh chi ra thao tác (operation) cần được thực hiện.

The diagram illustrates a system bus architecture. At the top, there are four green rectangular blocks representing system components: 'CPU', 'Memory', 'Memory', and 'I/O'. Ellipses between the second 'Memory' and 'I/O' blocks indicate that there can be multiple instances of these components. Below these blocks, a horizontal grey bar represents the 'Bus'. Three distinct lines connect the components to the bus: a top line for 'Control lines', a middle line for 'Addresses lines', and a bottom line for 'Data lines'. On the right side of the bus, a bracket groups these three lines and is labeled 'Bus'.

**Figure 3.16 Bus Inter connection Scheme**

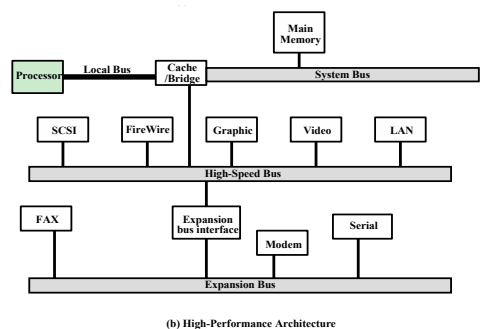
## Kiến trúc truyền thống (Có cache)



### (a) Traditional Bus Architecture

# Cấu hình BUS

## Kiến trúc hiệu suất cao



### Các yếu tố trong thiết kế Bus

Type	Bus Width
Dedicated	Address
Multiplexed	Data
Method of Arbitration	Data Transfer Type
Centralized	Read
Distributed	Write
Timing	Read-modify-write
Synchronous	Read-after-write
Asynchronous	Block

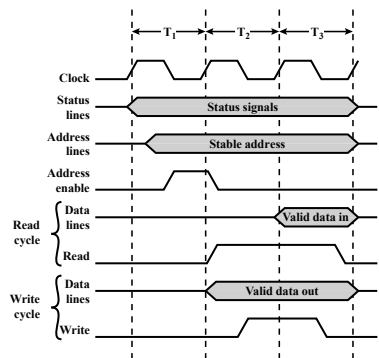
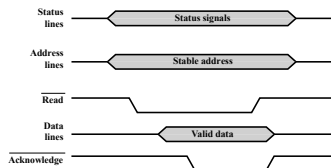
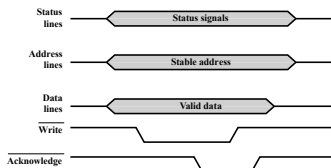


Figure 3.18 Timing of Synchronous Bus Operations

Phân chia  
thời gian hoạt  
động trên  
BUS đồng bộ



(a) System bus read cycle



(b) System bus write cycle

Phân chia thời  
gian hoạt  
động trên BUS  
đồng bộ

---

---

---

---

---

---

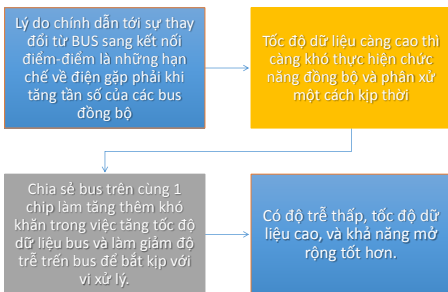
---

---

---

---

## Kết nối điểm - điểm




---

---

---

---

---

---

---

---

---

---

## Đường dẫn nhanh (Quick Path Interconnect)

QPI

- 
- 
- 




---

---

---

---

---

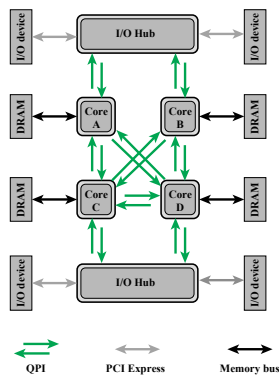
---

---

---

---

---



Cấu hình chip  
đa nhân sử  
dụng  
QPI

Figure 3.20 Multicore Configuration Using QPI

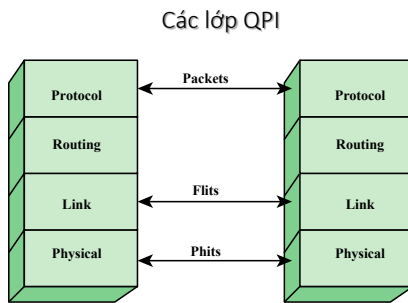
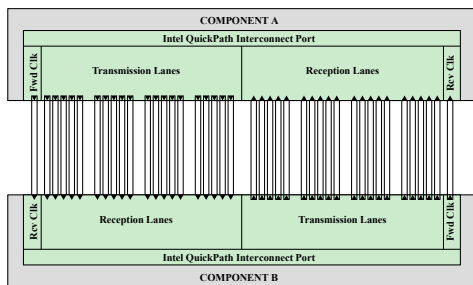


Figure 3.21 QPI Layers

Giao diện vật lý của kết nối QPI Intel



## Phân phối đa tuyến QPI

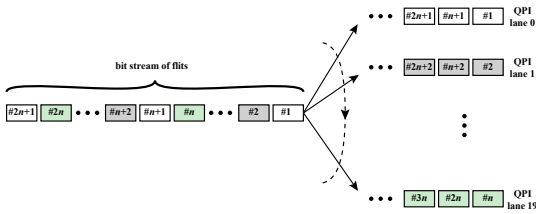


Figure 3.23 QPI Multilane Distribution

## Tầng kết nối QPI

- Thực hiện hai chức năng chính: *điều khiển luồng* và *điều khiển lỗi*.
  - Vận hành trên cấp flit (flow control unit – đơn vị điều khiển luồng)
  - Mỗi flit gồm tài tin 72-bit và một mã kiểm soát lỗi 8-bit được gọi là *cyclic redundancy check* (CRC)
- Chức năng điều khiển luồng
  - Cần thiết để đảm bảo rằng 1 thực thể QPI gửi không áp đảo 1 thực thể QPI nhận bằng cách gửi dữ liệu nhanh hơn khả năng xử lý dữ liệu và xóa bộ đệm để nhiều dữ liệu mới đến của phía nhận
- Chức năng điều khiển lỗi
  - Phát hiện và khắc phục lỗi bit, do đó tránh cho các lớp cao hơn gặp lỗi bit

## Lớp Giao thức và Định tuyến QPI

### Lớp Định tuyến

- Được sử dụng để xác định đường đi mà một gói sẽ đi qua các kết nối hệ thống có sẵn
- Xác định bởi phần sụn và mô tả các đường dẫn mà một gói tin có thể đi theo

### Lớp Giao thức

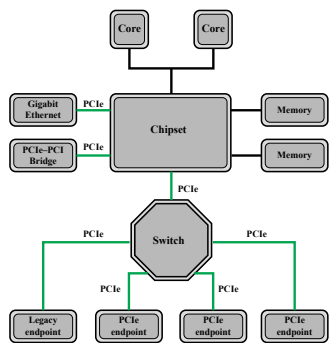
- Gói (packet) là đơn vị truyền
- Một chức năng quan trọng được thực hiện ở lớp này là giao thức liên kết bộ nhớ cache - đảm bảo rằng các giá trị bộ nhớ chính được giữ trong nhiều cache là phù hợp
- Một tài gói dữ liệu điển hình là một khối dữ liệu được gửi đến hoặc từ một bộ nhớ cache

# Kết nối thiết bị ngoại vi

## Peripheral Component Interconnect (PCI)

- Một bus băng thông cao, độc lập với bộ xử lý, có thể hoạt động như một bus ngoại vi
- Cung cấp hiệu năng hệ thống tốt hơn cho các hệ thống con I / O tốc độ cao
- Nhóm quan tâm đặc biệt PCI (Special Interest Group - SIG)
  - Được tạo ra để phát triển và duy trì tính tương thích của các đặc tính PCI
- PCI Express (PCIe)
  - Cơ chế kết nối điểm-điểm nhằm thay thế cơ chế dựa trên bus như PCI
  - Yêu cầu chính là dung lượng cao để hỗ trợ nhu cầu của thiết bị I / O tốc độ dữ liệu cao hơn, như Gigabit Ethernet
  - Một yêu cầu khác là phải hỗ trợ các luồng dữ liệu phụ thuộc thời gian

Cấu hình PCIe



Lớp giao thức PCIe

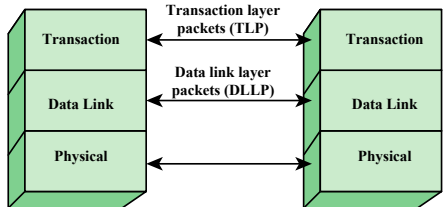


Figure 3.25 PCIe Protocol Layers



## Phân phối đa tuyến PCIe

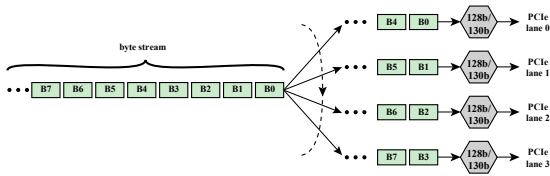
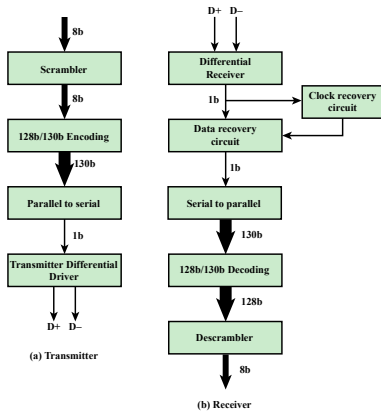


Figure 3.26 PCIe Multilane Distribution



Sơ đồ khối  
Truyền và  
nhận PCIe

## Lớp giao dịch PCIe

Transaction Layer (TL)



- Nhận các yêu cầu đọc và ghi từ phần mềm phía trên TL và tạo ra các gói tin yêu cầu để truyền tới đích qua lớp liên kết (link layer)
- Hầu hết các giao dịch sử dụng kỹ thuật giao dịch phân chia
  - Một thiết bị PCIe nguồn gửi 1 gói tin yêu cầu đi, sau đó đợi 1 phản hồi gọi là *gói tin hoàn thành*
- Bản tin TL và một số giao dịch ghi là giao dịch gửi (nghĩa là không cần có phản hồi)
- Định dạng gói TL hỗ trợ địa chỉ bộ nhớ 32-bit và địa chỉ bộ nhớ 64-bit mở rộng

TL hỗ trợ bốn không gian địa chỉ:

- Bộ nhớ
    - Không gian bộ nhớ bao gồm bộ nhớ chính của hệ thống và thiết bị I / O PCIe
    - Các khoảng địa chỉ bộ nhớ nhất định được ánh xạ vào các thiết bị I / O
  - Cấu hình
    - Không gian địa chỉ này cho phép TL đọc / ghi các thanh ghi cấu hình kết hợp với các thiết bị I / O
- I / O
    - Không gian địa chỉ này được sử dụng cho thiết bị PCI kế thừa, với dải địa chỉ dành riêng dùng để xác định các thiết bị I / O kế thừa
  - Message
    - Không gian địa chỉ này dành cho các tín hiệu điều khiển liên quan đến gián đoạn, xử lý lỗi, và quản lý năng lượng

---

---

---

---

---

---

---

---

Các kiểu giao dịch TLP PCIe

Address Space	TLP Type	Purpose
Memory	Memory Read Request	Transfer data to or from a location in the system memory map.
	Memory Read Lock Request	
	Memory Write Request	
I/O	I/O Read Request	Transfer data to or from a location in the system memory map for legacy devices.
	I/O Write Request	
Configuration	Config Type 0 Read Request	Transfer data to or from a location in the configuration space of a PCIe device.
	Config Type 0 Write Request	
	Config Type 1 Read Request	
	Config Type 1 Write Request	
Message	Message Request	Provides in-band messaging and event reporting.
	Message Request with Data	
Memory, I/O, Configuration	Completion	Returned for certain requests.
	Completion with Data	
	Completion Locked	
	Completion Locked with Data	

---

---

---

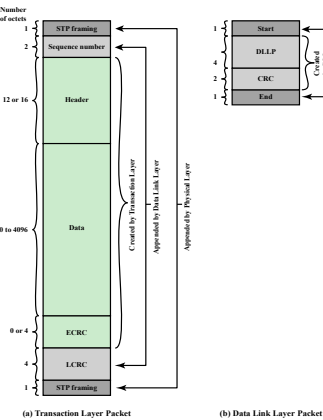
---

---

---

---

---



Định dạng Đơn vị dữ liệu Giao thức PCIe

---

---

---

---

---

---

---

---

Định dạng yêu cầu bộ nhớ TLP

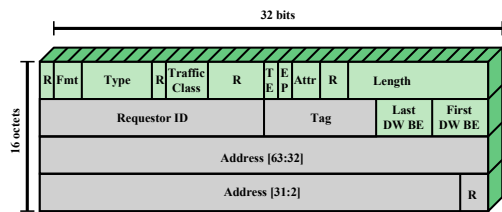


Figure 3.29 TLP Memory Request Format

Tổng kết

Chương 3

- Thành phần máy tính
- Chức năng máy tính
  - Lệnh truy xuất và thi hành
  - Gián đoạn
  - Chức năng I / O
- Cấu trúc kết nối
- Kết nối bus
  - Cấu trúc bus
  - Nhiều phân cấp bus
  - Các yếu tố thiết kế bus

Chức năng máy tính và kết nối

- Kết nối điểm-điểm
  - Lớp vật lý QPI
  - Lớp liên kết QPI
  - Lớp định tuyến QPI
  - Lớp giao thức QPI
- PCI Express
  - Kiến trúc vật lý và logic PCI
  - Lớp vật lý PCIe
- Lớp giao dịch PCIe
  - Lớp liên kết dữ liệu PCIe