



# Module

# 1

# Introduction



## *Lesson 1*

### *Giới thiệu về thời gian thực của hệ thống nhúng*

Chủ đề:

- Biết hệ thống nhúng là gì
- Sự khác biệt hệ thống nhúng thời gian thực và các hệ thống khác
- Sự khác biệt giữa thời gian thực và không phải thời gian thực
- Học những thông tin về 1 chiếc mobile phone
- Biết dc cấu trúc
- Các bộ phận cơ bản của 1 hệ thống nhúng

Definition:

**A real time embedded system ( hệ thống nhúng thời gian thực)**  
: là một hệ thống hoàn thành các công việc hoặc các tác vụ trong khoảng thời gian cho phép trước (deadline). Khoảng thời gian có thể là 1ns hoặc vài phút

Có 2 loại hệ thống thời gian thực:

- 1) Hard real-time: nếu không đáp ứng thời gian thực thì hệ thống sẽ sụp đổ hoàn toàn và gây hậu quả khôn lường. VD: hệ thống túi khí oto
- 2) Soft real-time: nếu không đáp ứng thời gian thực thì hệ thống sẽ bị suy giảm về chất lượng của dịch vụ. VD: hệ thống kiểm soát đóng mở cửa trên oto

Các đặc điểm của một RTES:

- 1) Đơn chức năng: một hệ thống thời gian thực sẽ chỉ thực hiện từng chức năng một. Khi thực hiện chức năng khác thì nó sẽ đưa chức năng cũ ra ngoài và nạp chức năng mới vào. Tất cả các hoạt động này đều được thực hiện và giám sát bởi hệ điều hành nhúng Real time operating system
- 2) Hạn chế chặt chẽ:



## 3) Thời gian phản ứng và thời gian thực

Kiến trúc chung của hệ thống nhúng thời gian thực :

### Question and answers:

**Câu 1: Điều nào sau đây là hệ thống thời gian thực ? Giải thích sự lựa chọn của bạn**

- a) Quạt trần
- b) Lò vi sóng
- c) TV
- d) Bàn phím desktop
- e) Máy ảnh kỹ thuật số

**Đáp án:** câu b và e

**Giải thích:** a) quạt trần ko phải lập trình

b) và e) tuân thủ hết điều kiện của 1 hệ thống nhúng như

+ ) Làm việc trong thời gian thực

+ ) lập trình trên nó

+ ) một số hệ thống cùng tồn tại trên 1 nền tảng duy nhất để chạy 1 chức năng duy nhất

c) TV chỉ lập trình 1 phần nhỏ trên nó. Nó có thể làm việc mà ko cần lập trình cho nó. Không bị ràng buộc chặt chẽ

d) Nó có 1 vi xử lý thông thường chứ ko phải lập trình

**Câu 2: Viết 5 ưu và nhược điểm của hệ thống nhúng thời gian thực**

a) 5 ưu điểm:

- Kích thước nhỏ



- Trọng lượng nhỏ
- Tiêu thụ điện năng thấp
- Nhiều điện từ thấp
- Giá thấp hơn

b) 5 nhược điểm:

- Sửa chữa và bảo trì là không thể
- Obsolesce nhanh hơn
- Khó quản lý dc nhiệt độ
- Khó thiết kế

***Câu 3: Hoạt động thời gian thực theo ý nghĩa là để làm gì ? cho ví dụ***

Nhiều hệ thống nhúng liên tục phải phản ứng với những thay đổi trong môi trường của hệ thống và phải tính toán kết quả nhất định trong thời gian thực mà không có sự chậm trễ.

Ví dụ:

Hành trình điều khiển của xe liên tục giám sát và phản ứng để tăng tốc và cảm biến phanh. Nó phải tính toán tăng tốc hoặc giảm tốc số liên tục trong một thời gian hạn chế; một tính toán chậm trễ có thể dẫn đến thất bại trong việc duy trì sự kiểm soát của chiếc xe.

Ngược lại một hệ thống máy tính để bàn chủ yếu tập trung vào các tính toán, với phản ứng tương đối không thường xuyên (từ góc nhìn của máy tính) với các thiết bị đầu vào. Ngoài ra, sự chậm trễ trong những tính toán, trong khi có vẻ bất tiện với máy tính của người sử dụng, thường không dẫn đến sự thất bại của hệ thống.

***Câu 4: Cung cấp ít nhất 5 hệ thống nhúng mà bạn đang sử dụng ? xem xét trong cuộc sống hằng ngày của bạn***

Điện thoại di động, máy ảnh kỹ thuật số, laptop , ipod



# Lesson

# 2

## Introduction to Real Time Embedded Systems Part II



## Giới thiệu hệ thống nhúng thời gian thực part 2

### Cấu trúc và thiết kế:

#### Mục tiêu đạt được:

- Tìm hiểu thêm về các hệ thống nhúng thời gian thực
- Tìm hiểu cấu trúc phần cứng của một chiếc điện thoại
- Tìm hiểu về các thành phần quan trọng của một RTES
- Tìm hiểu về thành phần của một chiếc điện thoại
- Tìm hiểu về các vấn đề thiết kế quan trọng khác nhau

### Mobile phone:

Một chiếc điện thoại gồm những thành phần sau :

- Một bảng mạch
- Anten
- Microphone
- Speaker
- LCD
- Keyboard
- Battery

### Các thành phần của một hệ thống nhúng:

#### 1) Microprocessor

Đây là trung tâm của bất kì một hệ thống nhúng nào. Mỗi một hệ thống nhúng đều có một microprocessor có chức năng khác nhau tùy mục đích của hệ thống mình sử dụng

#### 2) Memory

Nhỏ gọn, tốc độ và mức tiêu thụ điện năng thấp là những đặc điểm cần thiết cho bộ nhớ được sử dụng trong hệ thống nhúng thời gian thực

#### 3) Input output devices and interfaces

Input /output là cần thiết để làm cho RTES tương tác với thiết bị bên ngoài như màn hình LCD, anten, micro,....

#### 4) Software

Hệ thống RTES sau khi thiết kế chỉ là 1 phần cứng vô tri ko có lập trình là ko có sự sống gì hết vì thế chưa thể thực hiện được chức năng

# Hệ thống nhúng : Lesson 2

---



gì cả. vì vậy mà trong RTES còn có hệ điều hành nhúng để thực hiện các chức năng đó

Kết luận:

Phạm vi của hệ thống nhúng đã được bao quát hơn và đa dạng hơn trong ngành công nghệ ngày này so ngày khác. Ngày này ngày các hệ thống ngày càng nhỏ hơn hiệu quả hơn

Question:

**Câu 1: Cho 1 hệ thống nhúng trong bài lesson vẽ sơ đồ khối và thảo luận về chức năng của các khối khác nhau?**

**Câu 2: thảo luận về ổ đĩa cứng đặt trong máy tính của bạn. Nó có phải là RTES không ?**

Ổ đĩa cứng có 2 thành phần là bên trong và bên ngoài. Thành phần bên ngoài được đặt trên 1 bảng mạch in gọi là logic board. Trong khi đó các thành phần bên trong được đặt trong 1 buồng kín dc gọi là HDA hoặc ổ cứng hội

Các mạch lớn là bộ điều khiển. nó chịu trách nhiệm về mọi thứ : trao đổi dữ liệu giữa các ổ đĩa cứng và máy tính, kiểm soát các hoạt động cơ trên ổ đĩa cứng. Vì thế nó là 1 RTES

**Câu 3: Xây dựng về thiết kế số liệu thời gian để đưa ra thị trường**

Thời gian cần thiết để phát triển một hệ thống quan trọng đến mức nó có thể được phát hành và bán cho khách hàng tạo lợi nhuận lớn

Những thứ đóng góp chính là thời gian thiết kế , thời gian sản xuất và thời gian thử nghiệm. Số liệu này đã trở nên đặc biệt đòi hỏi trong những năm gần đây. Giới thiệu 1 hệ thống nhúng vào thị trường sớm có thể tạo 1 sự khác biệt lớn trong lợi nhuận của hệ thống

**Câu 4: định luật moore là gì ? Làm thế nào nó dc hình thành?**

Định luật moore là các quan sát thực nghiệm rằng sự phức tạp của các mạch tích hợp , đối với chi phí tối thiểu với thành phần , gấp đôi mỗi 24 tháng . đó là do



# Lesson

3

## Embedded Systems Components Part I





## Lesson 3

### Linh kiện hệ thống nhúng

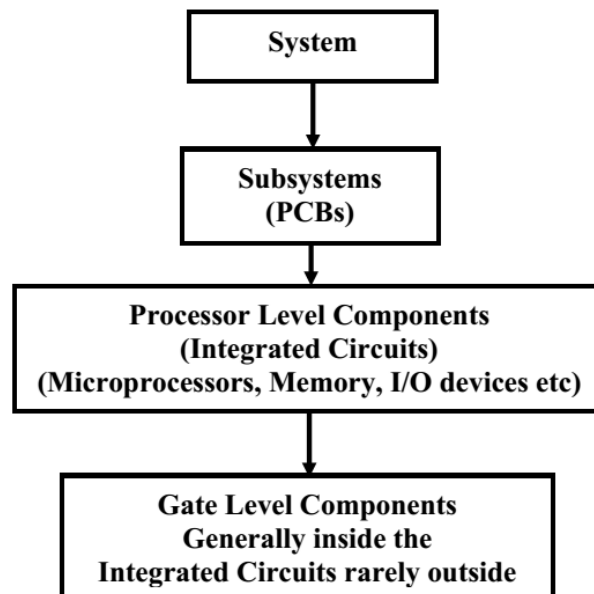
#### Giao diện cơ cấu và ví dụ

#### Mục tiêu của lesson 3:

- Biết cách bố trí cơ cấu
- Các thông số kỹ thuật của các thành phần chính của một hệ thống nhúng
- Đặc biệt là tìm hiểu về một board máy tính

#### Question:

**Câu 1: Cấu trúc các thành phần trong 1 hệ thống nhúng là gì:**



**The Hierarchical Components**

**Câu 2: LVDS là gì ?**

**Câu 3: một vài bộ truyền động trong điện thoại của bạn ?**

Có một máy rung trong thiết bị di động có thể được kích hoạt để nhận 1 cuộc gọi hoặc 1 tin nhắn. Nói chung có động cơ **coreless** được điều hành bởi vi điều khiển để tạo ra các rung động



# Lesson

# 4

# Embedded Systems Components Part II



## Lesson 4

### Linh kiện hệ thống nhúng part 2

Tổng quan về thành phần:

Mục tiêu lesson 4:

- Tổng quan những vấn đề sau:
  - Processors
  - Memory
  - Input/output devices

## ĐỊNH NGHĨA VỀ PROCESSER .

Bộ xử lý trung tâm là thành phần quan trọng nhất trong một hệ thống nhúng nó tồn tại hoạt động bằng cách tích hợp với bộ nhớ và các thiết bị ngoại vi khác .

### CÁC LOẠI PROCESER :

Tùy thuộc vào loại ứng dụng mà các bộ xử lý được chia làm 3 loại chính :

1. Bộ vi xử lý mục đích chung.
2. Vi điều khiển
3. Bộ xử lý tín hiệu số

Thực chất các bộ xử lý còn có thể tùy chỉnh dựa trên cầu sản xuất để chạy một ứng dụng cụ thể nào đó nhưng làm vậy sẽ tốn rất nhiều chi phí . Nên thường một ứng dụng sẽ được xây dựng dựa trên những bộ vi xử lý đã có sẵn trên thị trường.

### BỘ XỬ LÝ MỤC ĐÍCH CHUNG LÀ GÌ :

Đây là bộ xử lý được thiết kế để phục vụ nhiều ứng dụng thực tế như đa truyền thông , ô tô, công nghiệp ....Các bộ xử lý được sản xuất với số lượng lớn và chi phí rẻ phù hợp thị trường . Và được thường xuyên nâng cấp tối ưu , cải thiện thiết kế bởi nhà sản xuất



# **ĐỊNH NGHĨA VỀ MICRO CONTROLLER.**

Thường bạn đặt tất cả thành phần của máy tính lên trên một board mạch , nhưng nếu bạn đặt tất cả các thành phần trên board mạch vào một con chip duy nhất thì nó được gọi là vi điều khiển . Nó còn được ví như là một máy tính thu nhỏ do hạn chế trong thiết kế để đáp ứng được đa phần các yêu cầu thực tế nên các chức năng vào và ra tồn tại một cách đơn giản.



# SO SÁNH MICROPROCESSORS VÀ MICROCONTROLLERS

## + Đặc tính MICROPROCESSORS :

Một bộ vi xử lý là đơn vị xử lý số trung tâm cho những mục đích chung . Để thực hiện một máy vi tính đầy đủ, bạn có thêm bộ nhớ ( ROM và RAM ) giải mã bộ nhớ , một xung thạch dao động , và một số thiết bị I / O . Việc sử dụng chính của một bộ vi xử lý là để đọc dữ liệu , thực hiện tính toán sâu rộng trên dữ liệu đó , và lưu trữ các kết quả trong một thiết bị lưu trữ hoặc hiển thị kết quả . Những bộ vi xử lý có kiến trúc phức tạp với nhiều giai đoạn pipelining và xử lý song song . Các bộ nhớ được chia thành các giai đoạn như cache đa cấp và RAM . Thời gian phát triển của bộ vi xử lý Mục đích chung là cao vì rất phức tạp trong thiết kế .

## + Đặc tính MICROCONTROLLER :

Các thiết kế của vi điều khiển được thúc đẩy bởi mong muốn làm cho nó càng mở rộng và linh hoạt càng tốt . Những thành phần gồm có này là lõi CPU, bộ nhớ chương trình (thông thường là ROM hoặc bộ nhớ Flash), bộ nhớ dữ liệu (RAM), một hoặc vài bộ định thời và các cổng vào/ra để giao tiếp với các thiết bị ngoại vi và các môi trường bên ngoài tất cả chúng được thiết kế **trong một CHIP vi mạch tích hợp**. Vi điều khiển khác với các bộ vi xử lý đa năng ở chỗ là nó có thể hoạt động chỉ với vài vi mạch hỗ trợ bên ngoài. . Vi điều khiển cũng thường có timer để tạo ra ngắt và do đó có thể được sử dụng với các CPU và các chip A/D D/A hoặc cổng song song để có được thường xuyên hẹn giờ cho các thiết bị . Việc bạn dùng một vi điều khiển để kiểm soát các hoạt động của một loại máy móc theo một cách nào đó dựa trên một chương trình cố định được lưu trữ sẵn trong ROM và không thay đổi trong suốt cuộc quãng đời của hệ thống. Các vi điều khiển rất quan tâm đến dữ liệu truyền nhận vào các chân của mình . các kiến trúc cũng như tập lệnh được tối ưu hóa để xử lý những dữ liệu nằm trong khoảng Byte và Bit

## + Sự tương phản giữa chúng :

Các bộ vi xử lý thì có rất nhiều mã opcode để thực thi việc chuyển dữ liệu từ bộ nhớ ngoài vào CPU , vi điều khiển thì chỉ có 1 hoặc 2

**Vi xử lý thường thì sẽ có rất ít các bit xử lý mã lệnh , vi điều khiển thì sẽ có rất nhiều**



# ĐỊNH NGHĨA VỀ BỘ XỬ LÝ TÍN HIỆU SỐ

Đây là những bộ vi xử lý được thiết kế dựa trên các biến đổi của kiến trúc Harvard để xử lý thời gian thực . Các tính năng của bộ vi xử lý phù hợp để thực hiện các thuật toán xử lý tín hiệu.

( Một trong các tổ chức hoạt động quan trọng của bộ xử lý tín hiệu số là mảng nhân , Sử dụng để thực hiện các phép chập tín hiệu . Để thực hiện việc này thường có một đơn vị hỗ trợ gọi là MAC ( Multiplier and Accumulator ) hay là MACD ( Multiplier and Accumulator Data ) . Tất cả các instruction ( lệnh) đều được thực hiện trong một chu kỳ đơn.

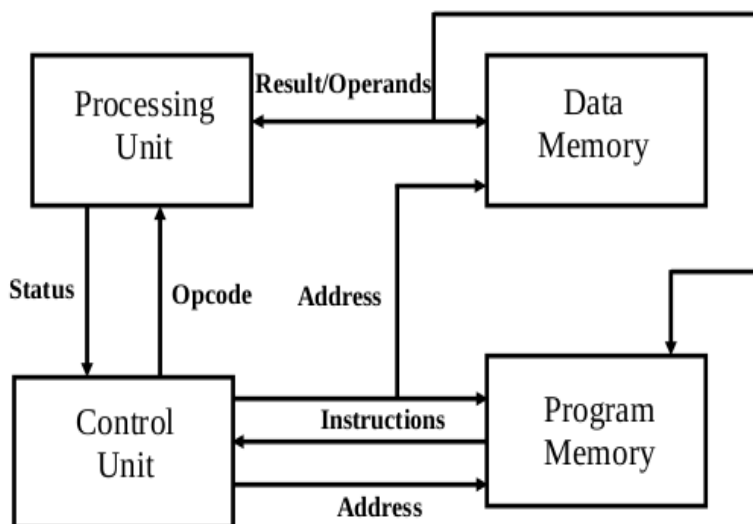


Fig. 4.3 The modified Harvard architecture

The MACD type of instructions can be executed faster by parallel implementation. This is possible by separately accessing the program and data memory in parallel. This can be

Biến đổi dựa trên kiến trúc havard

Microprocessors may have one or two types of bit-handling instructions; microcontrollers will have many.

Vi xử lý có thể có một hoặc hai loại “lệnh xử lý bit”, vi điều khiển có nhiều hơn



## MEMORY

- Lưu giữ lần chương trình và dữ liệu
    - **Princeton** Architecture: (Kiến trúc Princeton ):  
Chương trình và dữ liệu được lưu cùng memory
    - **Harvard** Architecture (Kiến trúc Harvard) : Chương trình và dữ liệu có các khối lưu riêng
  - **General purpose processors : có kiến trúc Princeton**
  - Bộ nhớ có thể là **RAM hoặc ROM**, có thể on-chip hay bên ngoài, bộ nhớ on-chip thì nhanh hơn. Để giảm thời gian đọc ghi, người ta làm ra bộ nhớ **Cache** nhỏ nhưng rất nhanh
  - Bộ nhớ có thể là bộ nhớ động (**Dynamic**) hay bộ nhớ tĩnh (**Static**):
    - **Bộ nhớ động** : Ít tiêu tốn năng lượng, nhỏ gọn, giá rẻ
      - **Trong DRAM** (Dynamic RAM): giá trị được lưu lại nhờ hoạt động làm mới định kỳ
    - **Bộ nhớ tĩnh**: Nhanh hơn bộ nhớ động gấp nhiều lần
      - **Trong SRAM** (Static Memory) : dữ liệu được nguồn giữ liên tục do đó sẽ truy xuất nhanh hơn và tốn điện hơn
- ⇒ Cache là SRAM**
- Khi cần dữ liệu CPU sẽ tìm trong **thanh ghi** của mình, nếu không có sẽ tìm **Cache level 1** nếu không có nữa sẽ tìm ở **Cache level 2** nếu không có nữa sẽ tìm trong bộ nhớ chính **RAM** nếu vẫn không có nó sẽ tìm cuối cùng tại **Disk** !. Trong khi đó CPU vẫn nằm đợi

**Register => Cache lv1 => Cache lv2 => RAM => Disk**

## Input/Output Devices and Interface Chips



- Để giao tiếp với người sử dụng hoặc môi trường thì hệ thống RTES cần một số phần cứng sẵn có .Ví dụ như : bàn phím , màn hình hiển thị ,màn hình , ăng-ten , micro , loa , đèn LED chỉ số vv
- Các tín hiệu này có thể là Analog hoặc Digital ,để tạo ra tín hiệu Analog từ bộ xử lý chúng ta cần bộ **Digital to Analog Converter(DAC)**, để tạo tín hiệu Digital cho bộ xử lý cần **bộ Analog to Digital Converter (ADC)**.Những bộ này hoạt động riêng lẻ với tốc độ khác nhau nên chúng ta cần một con **Chip giao diện** để điều khiển. Tương tự cho các thiết bị khác.
- **Chip giúp cho CPU không phải chờ đợi thực hiện các thao tác vào/ra do đó tăng tốc độ xử lý CPU**

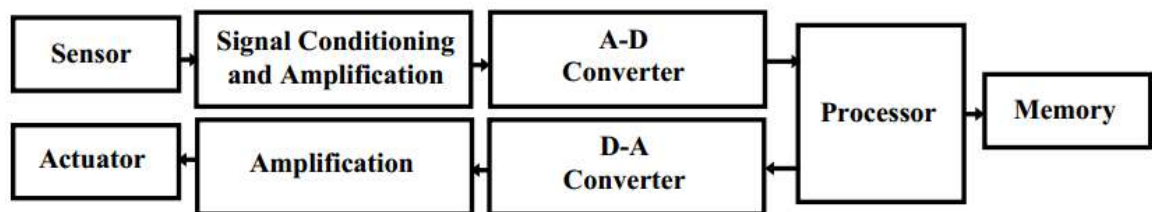


Fig. 4.10 The typical input/output interface blocks

Question:

**Câu 1: liệt kê những điểm tương đồng và khác biệt giữa vi điều khiển và xử lý tín hiệu kỹ thuật số**

**Câu 2: Tên của 1 vài con chip trong gia đình processors:**

Microcontroller: Intel 8051, Intel 80196, Motorola 68705

Digital Signal Processors: TI 3206711, TI 3205000

General Purpose Processor: Intel Pentium IV, Power PC

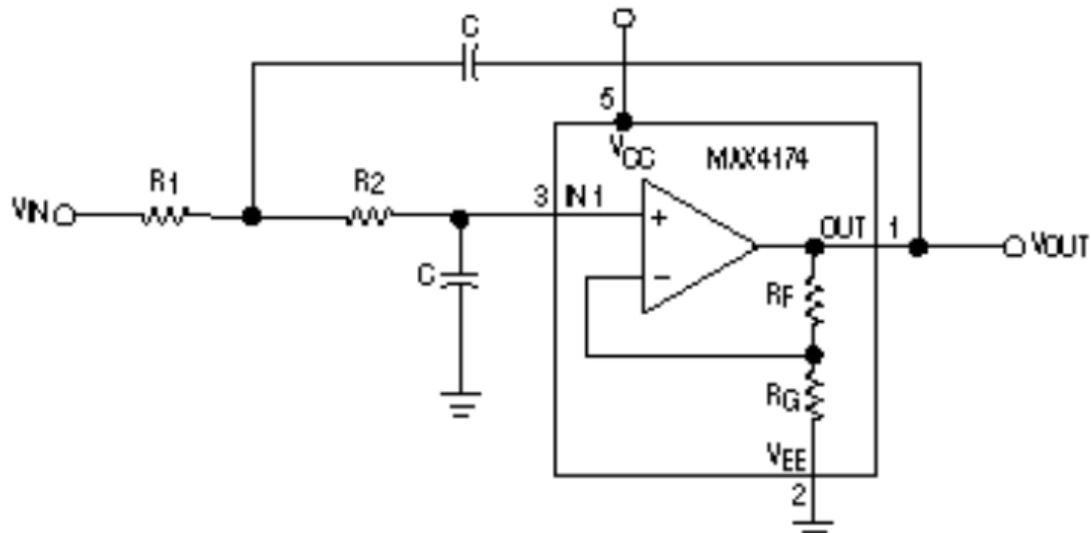
**Câu 3: Sắp xếp theo thứ tự tốc độ truy cập của Flash Memory, Dynamic Memory, Cache Memory, CDRom, Hard Disk, Magnetic Tape, Processor Memory**





Sắp xếp: Magnetic Tape, CDROM, Hard Disk, Dynamic Memory, Flash Memory, Cache Memory, Processor Memory

**Câu 4: vẽ mạch một bộ lọc chống rung của sử dụng bộ khuếch đại**



**Low Pass Sallen Key Butterworth Filter**



# Lesson

# 5

# Memory-I



## ➤ CẤU TRÚC:

- Có ba kiến trúc bộ nhớ khác nhau :
  - Bộ nhớ xử lý.
  - Bộ nhớ chính.
  - Bộ nhớ tương tác.

## ➤ MÔ HÌNH TỔ TỔNG QUÁT CÁCH HOẠT ĐỘNG :

- Mô hình hệ thống máy tính và bộ nhớ cơ bản phổ biến :

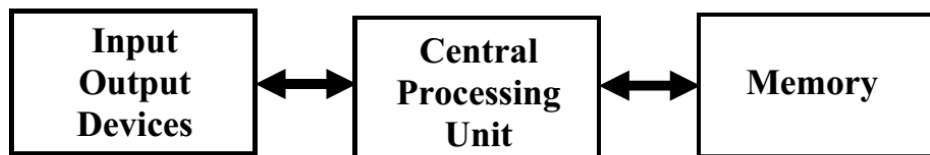
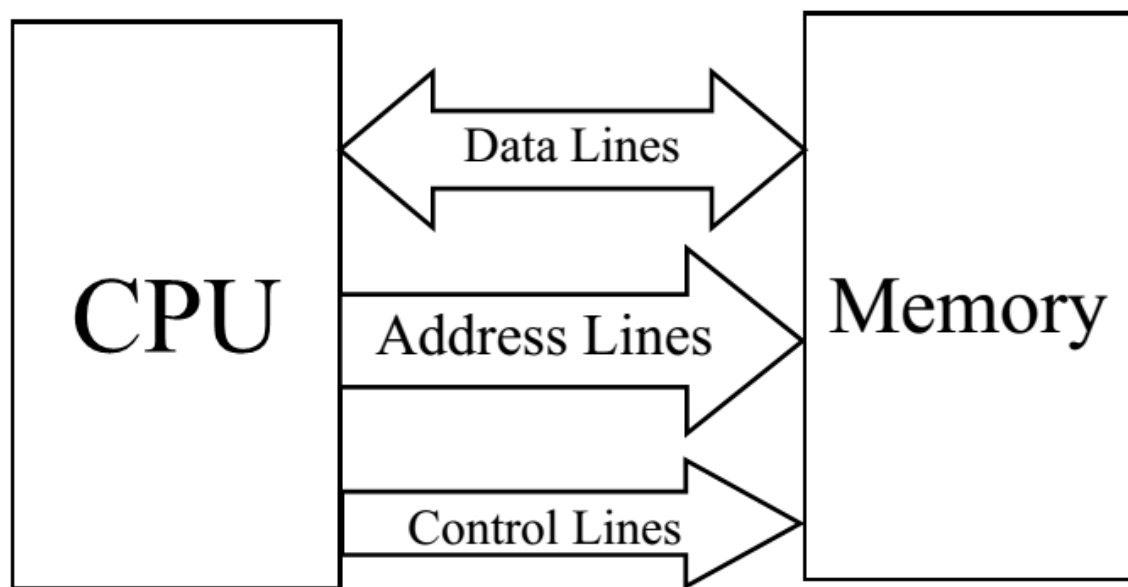


Fig. 5.1 The Von Neumann Architecture

- Bộ nhớ Memory kết nối với CPU qua 3 đường :
  1. Address
  2. Data
  3. Control



## + Cách hoạt động :

- Chế độ đọc dữ liệu : CPU sẽ load giá trị địa chỉ cần truy xuất lên đường Address Bus đi tới bộ giải mã và giải mã ra địa chỉ thực để truy xuất lên RAM. Sau đó CPU gửi tín hiệu READ SIGNAL qua đường Control lines. Dữ liệu sẽ

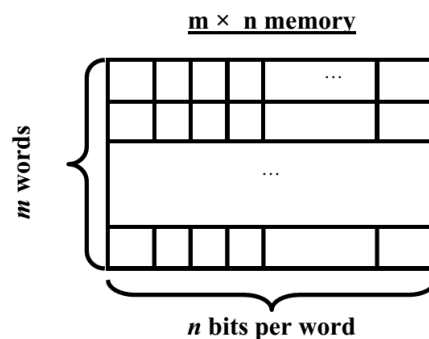


được chuyển từ memory sang CPU bằng đường DATA LINES

- Tương tự cho chế độ ghi dữ liệu chỉ thay đổi tín hiệu thành WRITE SIGNAL và dữ liệu sẽ được chuyển từ CPU tới MEMORY.

## ➤ CÁCH THỨC LƯU TRỮ BỘ NHỚ :

- Ví dụ một ram :  $(m = 4,096) \times (n = 8)$  memory:
  - Có  $m * n = 32,768$  bits.
  - Có  $K = \log_2(m) = 12$  address input signals ( 12 tín hiệu địa chỉ truy xuất )
  - Độ rộng bit là 8.



## ➤ TRUY XUẤT BỘ NHỚ :

- Các bộ nhớ có thể được truy xuất bằng cách định giá trị địa chỉ lên các đường address line của bộ nhớ, và đường control sẽ kích hoạt tín hiệu đọc hay ghi cho bộ nhớ đó . Một số bộ nhớ có nhiều cổng , nhiều đường truy xuất để có thể đi đến các vùng nhớ khác nhau trong cùng một thời điểm ...!

## ➤ THÔNG SỐ KỸ THUẬT :

- The storage capacity ( Dung lượng lưu trữ ) : là giá trị bit, byte hay word có thể lưu trữ.
- The memory access time : (Thời gian truy xuất bộ nhớ) : là thời gian nó truy xuất để đẩy dữ liệu ra đường data line khi đã nhận được địa chỉ truy xuất . hoặc là khả năng lưu trữ dữ liệu của nó nhanh hay chậm. Những đáp ứng về thời gian trên được hiểu như là : Thời gian truy xuất bộ nhớ.
- Bandwidth ( Băng thông ) : ??? WTF ???
- Có 2 thông số kỹ thuật quan trọng nữa của bộ nhớ khi ta thành lập một hệ thống nhúng cần quan tâm :
  - Write Ability (Khả năng ghi) :
    - ✓ High end ( Cao cấp ) : Loại có khả năng ghi rất nhanh và đơn giản....RAM....



- ✓ Middle Range ( loại trung bình ) : Loại có khả năng ghi vào nhưng chậm hơn. ,, FLASH....
  - ✓ Cấp thấp ( Loại tầm thấp ) : Dành cho các thiết bị đặt biệt , lập trình vào bộ nhớ ... EPROM, OTP ROM
  - ✓ Low end ( Loại thấp ) : các bit và dữ liệu được lưu trữ cố định trong quá trình sản xuất .... Mask-programmed ROM
  - Storage Performance (Hiệu năng lưu trữ):
    - ✓ High end ( Cao cấp ) : Không bao giờ mất bit
    - ✓ Middle Range ( loại trung bình ) : có khả năng lưu trữ các bit qua nhiều ngày, nhiều tháng , nhiều năm sau khi ngồn bị mất.
    - ✓ Cấp thấp ( Loại tầm thấp ) : Các bit được giữ khi nguồn điện còn.
    - ✓ Low end ( Loại thấp ) : bị mất bit ngay lập tức sau khi ghi.
- **PHÂN LOẠI THEO CHỨC NĂNG BỘ NHỚ :**
- Dựa vào các chỉ số như Điện năng tiêu thụ, cách thức lưu trữ , vân vân người ta có thể phân loại bộ nhớ thành 5 loại cơ bản :
    - Processor Memory ( Kiểu register thanh ghi):
      - Nơi dùng để lưu trữ các toán tử và kết quả cho CPU, tốc độ truyền dữ liệu rất nhanh ! nhưng bị hạn chế về số lượng.
    - Internal on-chip Memory ( Bộ nhớ nội bộ on-chip)
      - Trong một số bộ vi xử lý có thể có một khối bộ nhớ xác định , có cùng kiểu như bộ nhớ ngoài nhưng nó thực thi xử lý nhanh hơn
    - Primary Memory ( Bộ nhớ chính, Bộ nhớ sơ cấp )
      - Bộ nhớ sơ cấp: hay bộ nhớ chính hoặc bộ nhớ trong, Ví dụ: RAM. ROM, cache  
Là bộ nhớ được truy cập trực tiếp bởi CPU. CPU liên tục đọc các lệnh được lưu trong bộ nhớ này, thực hiện các lệnh. Các dữ liệu được xử lý thường xuyên cũng được lưu trong bộ nhớ này.
    - Secondary Memory ( Bộ thứ cấp )
      - Là bộ nhớ mà CPU truy cập phải thông qua bộ nhớ sơ cấp. Khi mất điện dữ liệu vẫn được lưu trữ



lại và không mất. Những loại bộ nhớ này thường rẻ chẳng hạn như ổ cứng, ổ CD ROM v.v..v

- Cache memory
  - Cache CPU là bộ nhớ đệm tốc độ cao sử dụng công nghệ SRAM và được đặt nằm giữa bộ xử lý CPU và bộ nhớ chính RAM với mục đích làm tăng tốc độ truy cập và xử lý dữ liệu.

## ➤ PHÂN LOẠI THEO KHẢ NĂNG HOẠT ĐỘNG BỘ NHỚ :

### ○ ĐẶC CHủng ROM (Bộ nhớ chỉ đọc) :

Là một loại thiết bị lưu trữ dùng trong máy tính và các thiết bị khác thông tin trên ROM vẫn được duy trì dù nguồn điện cấp không còn. ROM cho phép ghi dữ liệu một lần duy nhất hoặc nhiều lần.

- **Mask ROM ( MROM )** là một loại bộ nhớ chỉ đọc ( ROM ) mà nội dung được lập trình bởi các nhà sản xuất mạch tích hợp. Nó lưu dữ liệu mãi mãi và không thay đổi.
  - **OTP ROM: One-time programmable ROM ( Bộ nhớ chỉ lập trình một lần )**: khả năng ghi rất thấp, nhưng khả năng lưu trữ cao rất khó bị thay đổi bit trừ phi được nhà sản xuất Lập trình lại và thời chi phần cứng. Sử dụng trong giai đoạn sản xuất cuối cùng ( Thành sản phẩm để khó bị thay đổi CODE bởi người khác )
  - **EPROM: Erasable programmable ROM ( ROM có khả năng xóa chương trình )** bằng cách sử dụng tia cực tím .
  - **EEPROM** : Electronic Erasable programmable ROM ( ROM có khả năng xóa chương trình bằng cách sử dụng điện).
  - **Flash Memory** : *Đây là dạng đặc biệt mở rộng của EEPROM ( ROM xóa bằng điện ) có khả năng ghi và lưu trữ lâu dài . Nó có thể xóa đi với tốc độ nhanh hơn bằng cách xóa nhiều WORD một lúc thay vì xóa từng WORD như EEPROM ( ROM xóa bằng điện ).*
- ### ○ ĐẶC CHủng RAM ( B/N truy cập ngẫu nhiên):

Nó có đặc tính: thời gian thực hiện thao tác đọc hoặc ghi đối với mỗi ô nhớ là như nhau, cho dù đang ở bất kỳ vị trí nào trong bộ nhớ.



- **SRAM (Static RAM)** : Loại này sử dụng các Flip-Flop để lưu trữ , với một cell nhớ gồm có 6 transistor , dữ liệu còn khi nguồn điện được duy trì.
- **DRAM (Dynamic RAM)** : Bộ nhớ này sử dụng MOS transistor và tụ điện để lưu trữ dữ liệu , nhỏ gọn tiện hơn SRAM, nhưng chậm hơn SRAM và cần phải làm tươi ram trong quá trình sử dụng vì các bit nhớ sẽ bị mất (do tụ điện bị xả) phải có bộ điều khiển nạp lại các bit nhớ này liên tục . Tốc độ làm tươi thông thường là 15.625 micro(s).

### ➤ MỘT SỐ BIẾN THỂ CẢI TIẾN CỦA RAM :

- **PSRAM: (Pseudo-static RAM )** : là một dạng DRAM tích hợp với bộ điều khiển làm tươi cho RAM.Loại này phổ biến giá rẻ có thể sử dụng trong một số trường hợp thay thế cho SRAM.
- **NVRAM: (Nonvolatile RAM)** : Là một dạng SRAM nhưng tích hợp thêm một pin nguồn luôn duy trì và kết nối với chính nó. ➔ Khi mất nguồn dữ liệu vẫn có thể được giữ lại trong một khoảng thời gian. TỐC ĐỘ GHI VÀ ĐỌC NHƯ NHAU. KHÔNG BỊ GIỚI HẠN SỐ LẦN GHI NHƯ ROM. CÓ THỂ THAY ĐỔI DỮ LIỆU RẤT NHANH. ( Quá đỉnh 😊 ^^!)
- **FPM-DRAM (Fast Page Mode DRAM)**  
Đây là một dạng cải tiến của DRAM, về nguyên lý thì FPM DRAM sẽ chạy lẹ hơn DRAM một tí do cải tiến cách dò địa chỉ trước khi truy cập thông tin. Những loại RAM như FPM hầu như không còn sản xuất trên thị trường hiện nay nữa.
- **EDO-DRAM (Extended Data Out DRAM)**  
Là một dạng cải tiến của FPM DRAM, nó chạy lẹ hơn FPM DRAM một nhờ vào một số cải tiến cách dò địa chỉ trước khi truy cập data. Một đặc điểm nữa của EDO DRAM



là nó cần support của system chipset. Loại memory này chạy với máy 486 trở lên (tốc độ dưới 75MHz). EDO DRAM cũng đã quá cũ so với kỹ thuật hiện nay. EDO-DRAM chạy lẹ hơn FPM-DRAM từ 10 - 15%.

- **BDEO-DRAM (Burst Extended Data Out DRAM)**

Là thế hệ sau của EDO DRAM, dùng kỹ thuật "pipeline technology" để rút ngắn thời gian dò địa chỉ của data. Nếu các bạn để ý những mẫu RAM tôi giới thiệu trên theo trình tự kỹ thuật thì thấy là hầu hết các nhà chế tạo tìm cách nâng cao tốc độ truy cập thông tin của RAM bằng cách cải tiến cách dò địa chỉ hoạt cách chế tạo hardware.

- **SDRAM (Synchronous DRAM)**

Đây là một loại RAM có nguyên lý chế tạo khác hẳn với các loại RAM trước. Như tên gọi của nó là "synchronous" DRAM, synchronous có nghĩa là đồng bộ, nếu bạn học về điện tử số thì sẽ rõ hơn ý nghĩ của tính đồng bộ.

*Synchronous là một khái niệm rất quan trọng trong lĩnh vực digital, trong giới hạn về chuyên môn tôi cũng rất lấy làm khó giải thích. Bạn chỉ cần biết là RAM hoạt động được là do một memory controller (hay clock controller), thông tin sẽ được truy cập hay cập nhật mỗi khi clock (dòng điện) chuyển từ 0 sang 1, "synchronous" có nghĩa là ngay lúc clock nhảy từ 0 sang 1 chứ không hẳn là clock qua 1 hoàn toàn (khi clock chuyển từ 0 sang 1 hay ngược lại, nó cần 1 khoảng thời gian interval, tuy vô cùng ngắn nhưng cũng mất 1 khoảng thời gian, SDRAM không cần chờ khoảng interval này kết thúc hoàn toàn rồi mới cập nhật*





*thông tin, mà thông tin sẽ được bắt đầu cập nhật ngay trong khoảng interval). Do kỹ thuật chế tạo mang tính bước ngoặt này, SDRAM và các thế hệ sau có tốc độ cao hơn hẳn các loại DRAM trước.*

*Đây là loại RAM thông dụng nhất trên thị trường hiện nay, tốc độ 66-100-133Mhz*

- **ESDRAM** ((S)ynchronous and Enhanced Synchronous (ES) DRAM) : Có thêm bộ đệm dữ liệu cho phép ghi đè địa chỉ , xung nhanh hơn và hạ thấp thời gian trễ do việc ghi và đọc dữ liệu
- **DDR SDRAM (Double Data Rate SDRAM)**  
Đây là loại memory cải tiến từ SDRAM. Nó nhân đôi tốc độ truy cập của SDRAM bằng cách dùng cả hai quá trình đồng bộ khi clock chuyển từ 0 sang 1 và từ 1 sang 0. Ngay khi clock của memory chuyển từ 0 sang 1 hoặc từ 1 sang 0 thì thông tin trong memory được truy cập.  
Loại RAM này được CPU Intel và AMD hỗ trợ, tốc độ hiện tại vào khoảng 266Mhz. (DDR-SDRAM đã ra đời trong năm 2000).
- **RDRAM** (viết tắt từ Rambus Dynamic RAM), thường được giới chuyên môn gọi tắt là “rambus”. Đây là một loại DRAM được thiết kế kỹ thuật hoàn toàn mới so với kỹ thuật SDRAM. RDRAM hoạt động đồng bộ theo một hệ thống lập và truyền dữ liệu theo một hướng. Một kênh bộ nhớ RDRAM có thể hỗ trợ đến 32 chip DRAM. Mỗi chip được ghép nối tuần tự trên một module gọi là RIMM (rambus inline memory module) nhưng việc truyền dữ liệu được thực hiện giữa các mạch điều khiển và từng chip riêng biệt chứ không truyền giữa các chip với nhau. Tốc độ Rambus đạt từ 400-800MHz.



➤ **MMU ( Manager Memory Unit ):** ( Đơn vị quản lý ram ) :

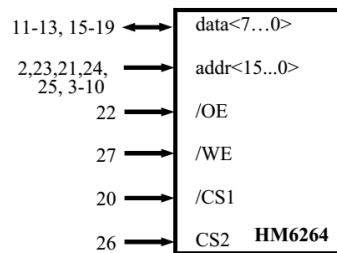
- Xử lý refresh DRAM , tương tác ram và các thiết bị
- Quản lý chia sẻ bộ nhớ giữa nhiều bộ vi xử lý
- Dịch các địa chỉ bộ nhớ logic từ bộ vi xử lý để thành các địa chỉ bộ nhớ vật lý
- của DRAM.
- Các CPU hiện đại thường đi kèm với MMU được xây dựng kèm theo.
- Việc xử lý đơn mục đích có thể được sử dụng.

➤ **Tích hợp ram vào các hệ thống xử lý.**

- *SRAM giải quyết được vấn đề khá đơn giản dễ dàng tích hợp trên CHIP và Các bộ xử lý.*
- *DRAM thì gặp nhiều vấn đề về việc chuyển địa chỉ thực sang logic*
- *Mục tiêu tích hợp của các nhà sản xuất IC có DRAM : Giảm thiểu sự kí sinh điện dung để hạn chế sự chậm trễ trong việc truyền dữ liệu và giảm điện năng tiêu thụ*
- *Mục tiêu của các nhà sản xuất DRAM : Tạo ra các tế bào điện để lưu giữ lại các thông tin một cách hiệu quả*



## ➤ MỘT VÍ DỤ NHỎ VỀ PHÂN TÍCH RAM



Data : Đường data 8 bit trao đổi dữ liệu

Addr : Đường địa chỉ truy xuất cho RAM

OE: output enable bar: Cho phép đọc khi chân này được set ở mức thấp

WE: write enable bar: Cho phép ghi dữ liệu

CS : các chân này dùng để mở rộng cho RAM khi kết nối với các thành phần khác.



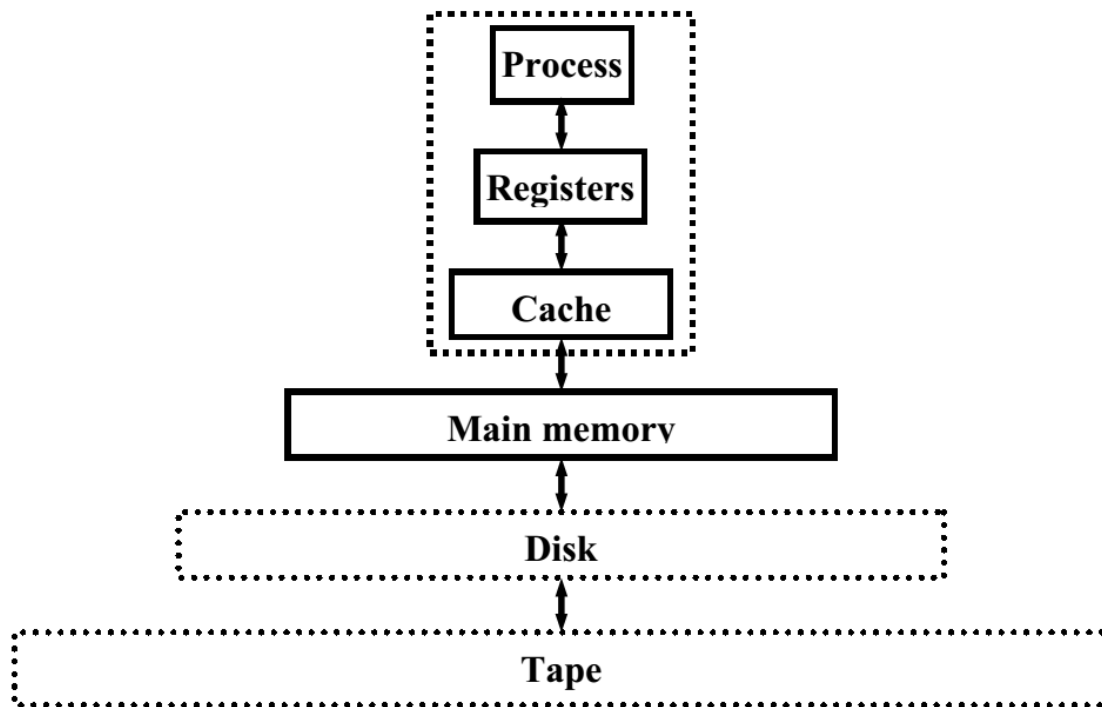
# Lesson

# 6

# Memory-II



### ➤ TỔ CHỨC CÁC BỘ NHỚ TRONG MỘT HỆ THỐNG :



**Fig. 6.1 The memory Hierarchy**

#### ○ Bộ nhớ Cache :

- Thường được thiết kế theo kiểu SRAM, Thường đi cùng với chip trong bộ xử lý , truy cập nhanh hơn bộ nhớ chính
- **Hoạt động :**
  - Nếu CPU có yêu cầu truy xuất vào bộ nhớ chính để đọc hoặc ghi, Đầu tiên nó sẽ kiểm tra bộ nhớ cache cho việc sao chép . Nếu như dữ liệu đó nằm trên cache ( Cache hit ) thì copy dữ liệu đó.
  - Nếu như không có dữ liệu cần nằm trên Cache (Cache MISS) , Bộ điều khiển sẽ lấy dữ liệu đưa vào Cache và CPU sẽ lấy dữ liệu đó.

# Hệ thống nhúng : Lesson 6



- Một số kiểu thiết kế Cache : ( Ánh xạ trực tiếp , Ánh xạ liên kết toàn phần , Ánh xạ tập hợp)

## Đặc điểm chung của cache / bộ nhớ chính (tiếp)

Một số Block của bộ nhớ chính được nạp vào các Line của cache.  
Mỗi dòng Tag (thẻ nhớ) cho biết Block nào của bộ nhớ chính hiện đang được lưu ở Line đó.  
Khi CPU truy nhập (đọc/ghi) một từ nhớ, hai khả năng xảy ra:  
Từ nhớ đó có trong cache (cache hit)  
Từ nhớ đó không có trong cache (cache miss).

Bài giảng Kiến trúc Máy tính

49

## 2. Các phương pháp ánh xạ

(Chính là các phương pháp tổ chức nhớ cache)

- Ánh xạ trực tiếp  
(Direct mapping)
- Ánh xạ liên kết toàn phần  
(Fully associative mapping)
- Ánh xạ liên kết tập hợp  
(Set associative mapping)

18 March 2007

Bài giảng Kiến trúc Máy tính

## Ánh xạ trực tiếp

Mỗi Block của bộ nhớ chính chỉ có thể được nạp vào một Line của cache:

$$3_0 \rightarrow L_0$$

$$3_1 \rightarrow L_1$$

....

$$3_{m-1} \rightarrow L_{m-1}$$

$$3_m \rightarrow L_0$$

$$3_{m+1} \rightarrow L_1$$

....

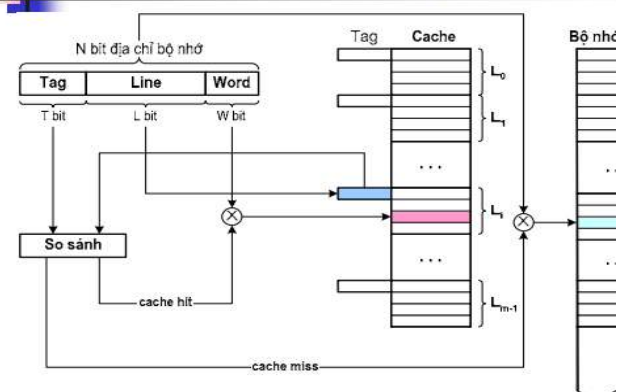
Tổng quát

$3_j$  chỉ có thể nạp vào  $L_{j \bmod m}$

$m$  là số Line của cache.

NKK-HUT

## Minh họa ánh xạ trực tiếp





## Đặc điểm của ánh xạ trực tiếp

Chỉ một địa chỉ N bit của bộ nhớ chính gồm địa chỉ trường:

Trường Word gồm W bit xác định một từ nhớ trong Block hay Line:

$$2^W = \text{kích thước của Block hay Line}$$

Trường Line gồm L bit xác định một trong số các Line trong cache:

$$2^L = \text{số Line trong cache} = m$$

Trường Tag gồm T bit:

$$T = N - (W+L)$$

Độ so sánh đơn giản

Xác suất cache hit thấp

Bài giảng Kiến trúc Máy tính

53

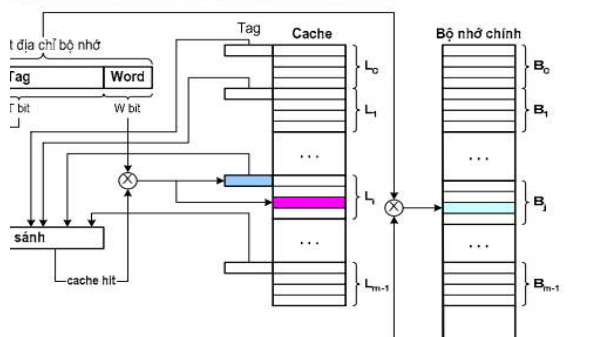
## Ánh xạ liên kết toàn phần

- Mỗi Block có thể nạp vào bất kỳ Line nào của cache.
- Địa chỉ của bộ nhớ chính bao gồm hai trường:
  - Trường Word giống như trường hợp ở trên.
  - Trường Tag dùng để xác định Block của bộ nhớ chính.
- Tag xác định Block đang nằm ở Line c

18 March 2007

Bài giảng Kiến trúc Máy tính

## Biểu đồ ánh xạ liên kết toàn phần



NKK-HUT

## Đặc điểm của ánh xạ liên kết toàn phần

- So sánh đồng thời với tất cả các Tag mất nhiều thời gian
- Xác suất cache hit cao.
- Bộ so sánh phức tạp.



## Ánh xạ liên kết tập hợp

Cache được chia thành các Tập (Set)

Mỗi một Set chứa một số Line

Ví dụ:

4 Line/Set  $\rightarrow$  4-way associative mapping

Ánh xạ theo nguyên tắc sau:

$B_0 \rightarrow S_0$

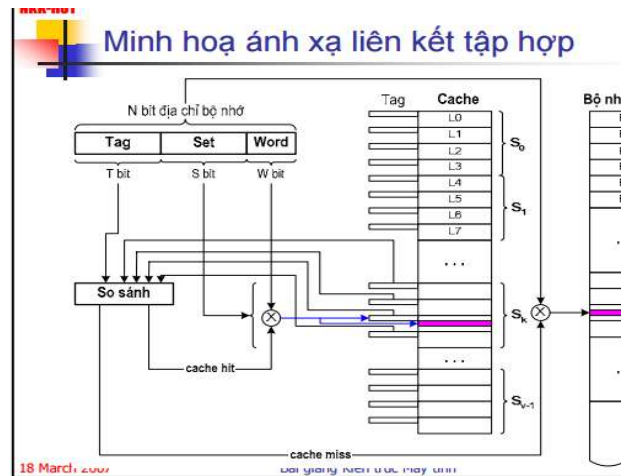
$B_1 \rightarrow S_1$

$B_2 \rightarrow S_2$

.....

Bài giảng Kiến trúc Máy tính

57



## Đặc điểm của ánh xạ liên kết tập hợp

Kích thước  $Block = 2^W Word$

Số lượng Set có S bit dùng để xác định

Số lượng Set trong số  $V = 2^S Set$

Số lượng Tag có T bit:  $T = N - (W+S)$

Ánh xạ quát cho cả hai phương pháp trên

Số lượng thường 2,4,8,16 Lines/Set

**Ví dụ về ánh xạ địa chỉ**

- Không gian địa chỉ bộ nhớ chính = 4
- Dung lượng bộ nhớ cache là 256KE
- Kích thước Line (Block) = 32byte.
- Xác định số bit của các trường địa chỉ cho ba trường hợp tổ chức:
  - Ánh xạ trực tiếp
  - Ánh xạ liên kết toàn phần
  - Ánh xạ liên kết tập hợp 4 đường





## Với ánh xạ trực tiếp

Bộ nhớ chính = 4GB =  $2^{32}$  byte  $\rightarrow N = 32$  bit

Cache = 256 KB =  $2^{18}$  byte.

Line = 32 byte =  $2^5$  byte  $\rightarrow W = 5$  bit

Số Line trong cache =  $2^{18} / 2^5 = 2^{13}$  Line  
 $\rightarrow L = 13$  bit

$T = 32 - (13 + 5) = 14$  bit

Tag	Line	Word
14 bit	13 bit	5 bit

Bài giảng Kiến trúc Máy tính

61

NKK-HUT

## Với ánh xạ liên kết toàn phần

■ Bộ nhớ chính = 4GB =  $2^{32}$  byte  $\rightarrow N = 32$  bit

■ Line = 32 byte =  $2^5$  byte  $\rightarrow W = 5$  bit

■ Số bit của trường Tag sẽ là:  $T = 32 - 5 = 27$  bit

Tag	Word
27 bit	5 bit

18 March 2007

Bài giảng Kiến trúc Máy tính

## Với ánh xạ liên kết tập hợp 4 đường

Bộ nhớ chính = 4GB =  $2^{32}$  byte  $\rightarrow N = 32$  bit

Line = 32 byte =  $2^5$  byte  $\rightarrow W = 5$  bit

Số Line trong cache =  $2^{18} / 2^5 = 2^{13}$  Line

Một Set có 4 Line =  $2^2$  Line

Số Set trong cache =  $2^{13} / 2^2 = 2^{11}$  Set  $\rightarrow$

$S = 11$  bit

Số bit của trường Tag sẽ là:  $T = 32 - (11 + 5)$

$T = 16$  bit

NKK-HUT

## 3. Thuật giải thay thế (1): Ánh xạ trực tiếp

- Không phải lựa chọn
- Mỗi Block chỉ ánh xạ vào một Line xác định
- Thay thế Block ở Line đó



## Đài thay thế (2): Ảnh xạ liên kết

Hiện bằng phần cứng (nhánh)

Thay thế ngẫu nhiên

(In First Out): Thay thế *Block* nào  
đầu tiên ở trong *Set* đó

(Most Frequently Used): Thay thế *Block*  
*Set* có số lần truy nhập ít nhất trong  
khoảng thời gian

(Least Recently Used): Thay thế *Block* ở  
trạng thái có thời gian lâu nhất không  
được truy nhập.

Thuật: LRU

Bài giảng Kiến trúc Máy tính

65

NKK-HUT



## 4. Phương pháp ghi dữ liệu khi

- Ghi xuyên qua (Write-through):
  - ghi cả cache và cả bộ nhớ chính
  - tốc độ chậm
- Ghi trả sau (Write-back):
  - chỉ ghi ra cache
  - tốc độ nhanh
  - khi *Block* trong cache bị thay thế  
ghi trả cả *Block* về bộ nhớ chính

18 March 2007

Bài giảng Kiến trúc Máy tính

## Trên các bộ xử lý Intel

Cache L1 trên chip

Có hai cache L1 trên chip

Cache = 8KB

Cache = 8KB

Pentium 4 (2000): hai mức cache L1 và L2

Cache:

Cache 8KB

Cache Line = 64 byte

Cache liên kết tập hợp 4 đường

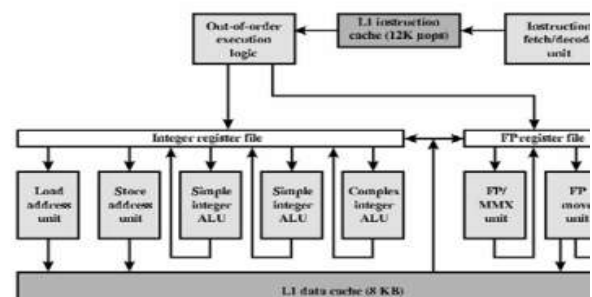
2

3

NKK-HUT



## Sơ đồ Pentium 4





### ▪ Các yếu tố liên quan đến hiệu năng của Cache:

- Tổng kích thước bộ nhớ cache.
- Kích cỡ một block \ line.
- Cache lớn đạt được hit lớn và miss thấp nhưng bù lại giá thành lại cao .
- Phương pháp ánh xạ tổ chức.

### **BÀI TẬP NHỎ VỀ VIỆC SO SÁNH BỘ SIZE BỘ NHỚ CACHE :**

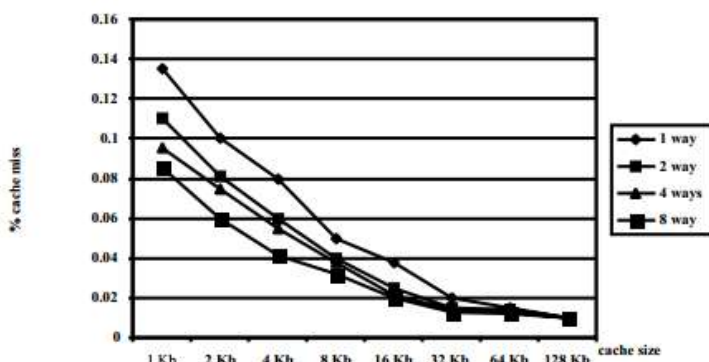
**Lưu ý :** Không phải lúc nào bộ nhớ Cache càng lớn thì càng tốt đôi lúc nếu bộ nhớ cache quá lớn thì máy tính phải truy xuất vô không gian nhớ quá lớn có thể tốn nhiều chi kì xung hơn bình thường => Chưa chắc hiệu quả : Ví dụ :

- e.g.,
  - 2 Kbyte cache: miss rate = 15%, hit cost = 2 cycles, miss cost = 20 cycles
    - avg. cost of memory access  
 $= (0.85 * 2) + (0.15 * 20) = 4.7 \text{ cycles}$
  - 4 Kbyte cache: miss rate = 6.5%, hit cost = 3 cycles, miss cost will not change
    - avg. cost of memory access  $= (0.935 * 3) + (0.065 * 20) = 4.105 \text{ cycles}$  **(improvement)**
  - 8 Kbyte cache: miss rate = 5.565%, hit cost = 4 cycles, miss cost will not change
    - avg. cost of memory access  $= (0.94435 * 4) + (0.05565 * 20) = 4.8904 \text{ cycles}$

Q1: Thảo luận một số Cache Mapping khác?

Ans: Direct, Fully Associative, Set Associative  
(trực tiếp, kết hợp đầy đủ, Thiết lập kết hợp)

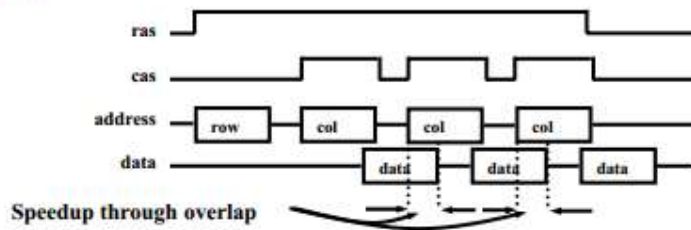
Q2: Thảo luận về kích thước và hiệu năng cache



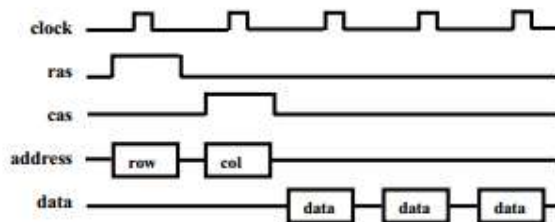


## Q3: So sánh EDO ROM và SDRAM

### EDO RAM



### SDRAM



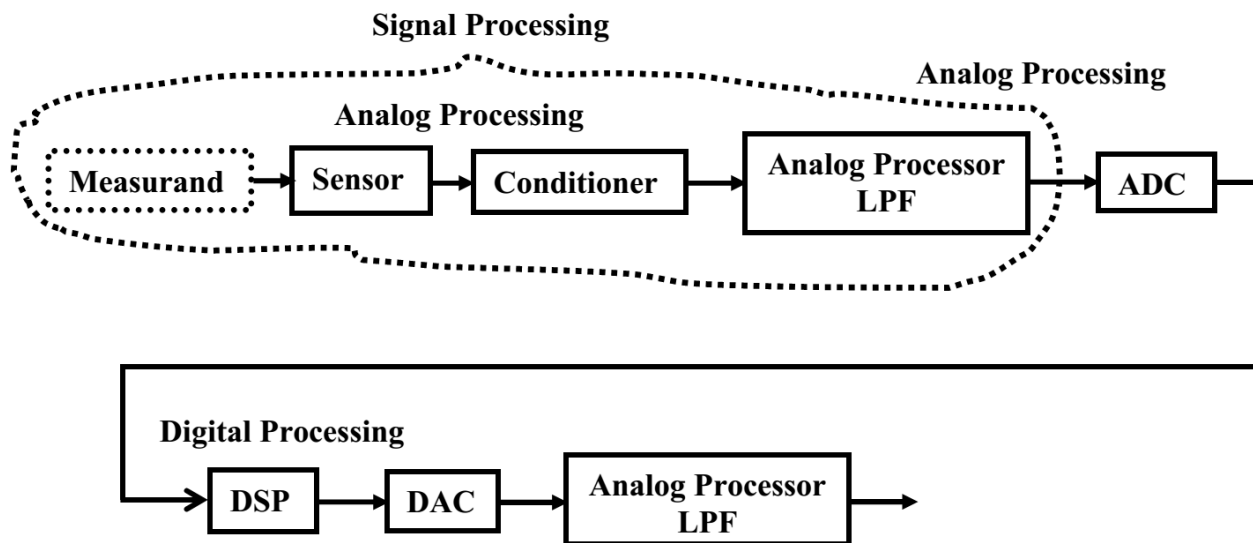


# Lesson 7

## Digital Signal Processors



## ➤ TÍN HIỆU SỐ ĐƯỢC XỬ LÝ NHƯ THẾ NÀO :



➤ Như mô hình ở trên thì ta có thể tóm tắt hoạt động như sau :

- Tín hiệu bên ngoài vào sẽ đi qua bộ cảm biến → tín hiệu từ cảm biến được đưa qua bộ Conditioner để thực hiện việc lọc tín hiệu , khuếch đại tín hiệu , chọn lọc tín hiệu v.v → tín hiệu tiếp tục đi tới bộ ANALOG PROCESSOR LPF khối này mang tính chất giống bộ lọc Thông thấp để chống hiện tượng răng cưa tín hiệu → tín hiệu đã được xử lý sẽ được đưa vào bộ ADC “ Khối chuyển đổi tín hiệu ANALOG sang dạng DIGITAL.
- Từ tín hiệu ADC chúng ta chuyển ngược về dạng ANALOG tín hiệu từ bộ ADC sẽ đi vào bộ phân giải tín hiệu DSP → sau đó tới bộ DAC “ khối chuyển tín hiệu DIGITAL dạng dạng ANALOG → tín hiệu sau cùng được đưa qua khối ANALOG PROCESSOR LPF để khử răng cưa và xuất ra ngoài trở lại.



## ➤ KHỐI ADC XỬ LÝ NHƯ THỂ NÀO :

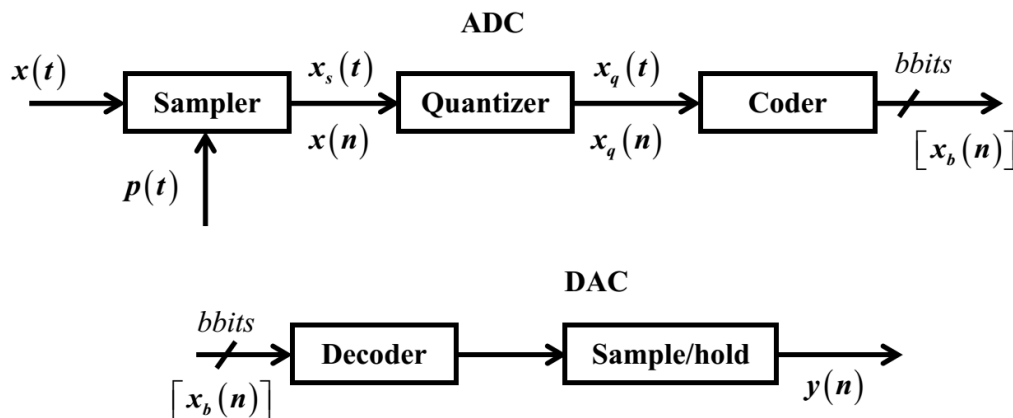


Fig. 7.2 D-A and A-D Conversion Process

## ➤ NHƯ MÔ HÌNH Ở TRÊN THÌ TA CÓ THỂ TÓM TẮT HOẠT ĐỘNG NHƯ SAU :

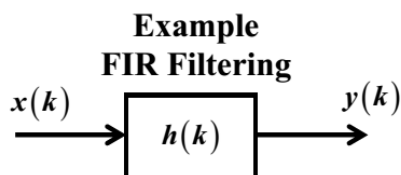
- Học sinh học môn xử lý số rồi tự giải thích ☺ .Cho biết chữ Quantizer có nghĩa là LƯỢNG TỬ HÓA( Bộ Làm tròn).

## ➤ ĐẶC ĐIỂM QUAN TÂM CỦA VIỆC CHUYỂN ĐỔI TRÊN :

- Ảnh hưởng của bộ ADC : Chất lượng của hệ thống xử lý tín hiệu số phụ thuộc rất lớn vào bộ ADC . Bộ ADC được xác định bởi số lượng bit hay nói theo cách khác là độ phân giải. 1 lần chuyển đổi được quyết định bởi một lần lấy mẫu. => ***lỗi phát sinh từ bộ ADC thường bắt nguồn từ việc chỉ có hữu hạn bit số phân giải và hữu hạn lần lấy mẫu.***
- Ảnh hưởng của bộ DAC : Tương tự như trên bộ DAC cũng có ảnh hưởng tới hệ thống thông qua chỉ tiêu về số lượng bit và chế độ thời gian ở đầu ra.
- Ảnh hưởng của DSP (công nghệ xử lý tín hiệu số): là một công nghệ được sử dụng để thiết lập các vị trí lọc khác nhau và nhằm tránh nhiễu.
  - Bộ DSP yêu cầu : Tính toán các số mang tính chấp lập đi lập lại , Chú ý đến độ chính xác của số, Có bộ nhớ băng thông rộng, Là hệ thống xử lý thời gian thực.
  - Khi thiết kế cần thiết kế tối ưu nhất :
    - Về mặt giá thành
    - Năng lượng tiêu thụ
    - Bộ nhớ sử dụng
    - Thời gian phát triển.



## VÍ DỤ VỀ MỘT BỘ DSP : BỘ LỌC ĐÁP ỨNG XUNG HỮU HẠN



$$\begin{aligned}
 y(k) &= (h_0 + h_1 z^{-1} + h_2 z^{-2} + \dots + h_{N-1} z^{N-1}) x(k) \\
 &= h_0 x(k) + h_1 x(k-1) + h_2 x(k-2) + \dots + h_{N-1} x(k-N+1) \\
 &= \sum_{i=0}^{N-1} h_i x(k-i) = h(k) * x(k)
 \end{aligned}$$

Các đầu ra của bộ lọc là một sự kết hợp tuyến tính của các giá trị hiện tại và quá khứ của đầu vào. nó có một số ưu điểm như :

Giai đoạn tuyến tính

Tính ổn định

Cải thiện thời gian tính toán

### DỰA THEO CÔNG THỨC TA CÓ ĐOẠN CODE

loop:

```

lw x0, (r0)
lw y0, (r1)
mul a, x0,y0
add b,a,b
inc r0
inc r1
dec ctr
tst ctr
jnz loop
sw b,(r2)
inc r2
    
```





## **BÀI TẬP NHỎ VỀ VIỆC KIỂM CHỨNG HỆ THỐNG SỐ**

Các lỗi hay xảy ra ở hệ thống thời gian thực

- Các lỗi phổ biến xảy ra trong bộ ADC :
  - i. Sampling error ( Lấy mẫu sai )
  - ii. Quantization ( Lượng tử hóa )
  - iii. Coding ( Mã hóa )
- Các lỗi phổ biến xảy ra trong bộ số học : Algorithm
  - i. in accurate modeling ( trong mô hình chính xác )
  - ii. Finite word length (chiều hữu hạn \_tràn số )
  - iii. Round of errors ( vòng lỗi )
  - iv. Delay due to finite execution time of the processor (chậm trễ do giới hạn thời gian xử lý ở các bộ vi xử lý )
- Các lỗi phổ biến xảy ra trong bộ DAC :
  - i. Decoding ( giải mã )
  - ii. Transients in sampling time ( Thời gian lấy mẫu quá ngắn )



# Lesson 8

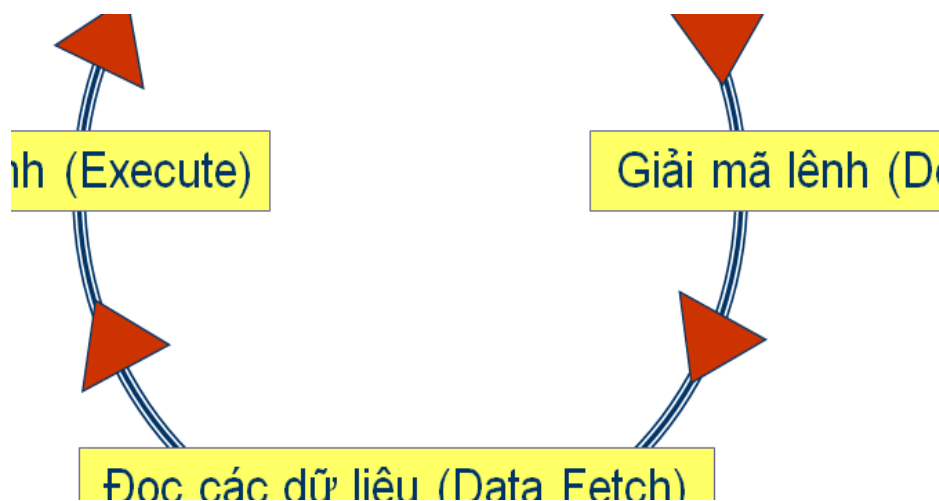
## General Purpose Processors - I



**Table 1 Development History of Intel Microprocessors**

<i>Intel Processor</i>	<b>Year of Introduction</b>	<b>Initial Clock Speed</b>	<b>Number of Transistors</b>	<b>Circuit Line Width</b>
4004	1971	108 kHz	2300	10 micron
8008	1972	500-800 KHz	3500	10 micron
8080	1974	2 MHz	4500	6 micron
8086	1978	5 MHz	29000	3 micron
8088	1979	5 MHz	29000	3 micron
Intel286 <sup>TM</sup>	1982	6 MHz	134,000	1.5 micron
Intel386 <sup>TM</sup>	1985	16 MHz	275,000	1.5 micron
Intel486 <sup>TM</sup>	1989	25 MHz	1.2 Million	1 Micron
Pentium <sup>TM</sup>	1993	66 MHz	3.1 Million	0.8 Micron
Pentium <sup>TM</sup> Pro	1995	200 MHz	5.5 Million	0.35 Micron
Pentium <sup>TM</sup> II	1997	300 MHz	7.5 Million	0.25 Micron
Celeron <sup>TM</sup>	1998	266 MHz	7.5 Million	0.25 Micron
Pentium <sup>TM</sup> III	1999	500 MHz	9.5 Million	0.25 Micron
Pentium <sup>TM</sup> IV	2000	1.5MHz	42 Million	0.18 Micron
Itanium <sup>TM</sup>	2001	800 MHz	25 Million	0.18 Micron
Intel® Xeon <sup>TM</sup>	2001	1.7 GHz	42 million	0.18 micron
Itanium <sup>TM</sup> 2	2002	1 GHz	220 million	0.18 micron
Pentium <sup>TM</sup> M	2005	1.5 GHz	140 Million	90 nm

## ➤ HOẠT ĐỘNG :



## ➤ CHÚNG TA SẼ ĐI XÉT MỘT SỐ TIÊU CHÍ TIÊU BIỂU :

### ○ Pipe line :

- Đây là thông số quan trọng của bộ vi xử lý hiện đại để tăng tốc độ thực thi chương trình. Để tăng tốc quá trình xử lý người ta chia nó vào những ô đơn vị khác nhau . Trong khi ô này đang Fetch ( instruction ) lệnh thì đơn



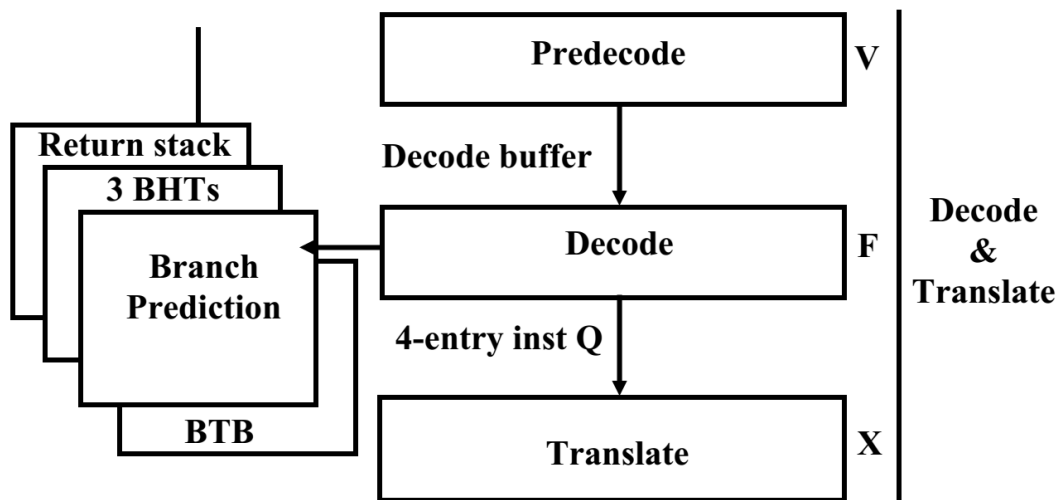
vị khác đang thực hiện việc giải mã lệnh và một đơn vị khác lại đang thực hiện việc thực thi lệnh v.v.v....

### ➤ CHÚNG TA SẼ ĐI XÉT CÁC KHỐI TRÊN SƠ ĐỒ :

#### ○ KHỐI FETCH INSTRUCTION UNIT :

- Tác dụng đưa lệnh từ bộ nhớ vào trong bộ đệm giải mã lệnh ( Decode ) để chuẩn bị thực hiện giải mã lệnh .
- Vì mỗi lần truy xuất vào bộ nhớ phải mất 2 chu kì ( Bài từ hệ điều hành : truy xuất vào bảng trang + truy xuất từ bảng trang tới memory ) . Để hạn chế việc này chúng ta sử dụng kĩ thuật TLB - Vùng đệm hỗ trợ chuyển đổi .

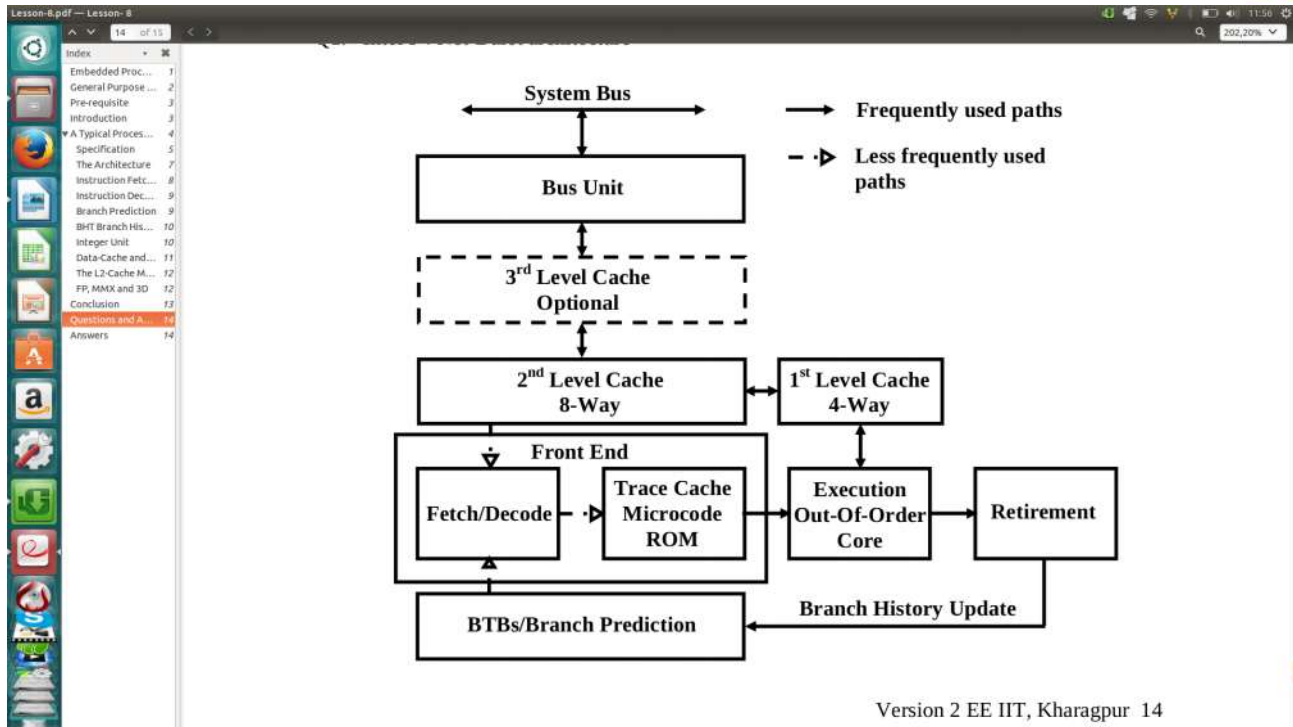
#### ○ Khối giả mã lệnh :



- Các dữ liệu Instructions byte sẽ được giải mã và biên dịch từ bên trong 2 giai đoạn F và X , giai đoạn F giải mã và định dạng khối lệnh thành một định dạng trung gian. Sau đó dữ liệu được đặt vào một hàng đợi trước khi vào giai đoạn X . Tại giai đoạn X các định dạng trung gian sẽ được biên dịch thành các Micro Instruction format ( Vi lệnh )
- Việc Fetch lệnh giải mã lệnh và biên dịch lệnh là thực hiện không đồng bộ tới nhau .

Trả lời câu hỏi bài tập 8 :

Câu 1 : vẽ kiến trúc Intel P4 Net-Burst :



**Câu 2: Kiến trúc superscalar của vi xử lý intel là gì ? Nó có giống gì với kiến trúc pipeline hay không ?**

Kiến trúc Superscalar : đề cập đến việc xử lý nhiều đơn vị cùng một lúc , cụ thể hơn ở đây là xử lý nhiều lệnh trong cùng một thời điểm , và đây thực sự là xử lý song song bởi nhiều CPU cùng tham gia hoạt động , và các bộ vi xử lý hiện đại đa phần đều có nhiều luồng Pipeline



Câu 4 : Làm thế nào để hiện thực hóa lệnh  $\text{SIN}(x)$  bằng phần cứng

- Bước 1 : Tìm trong bảng lệnh
- Bước 2 : Xác định loại thuật toán để tính  $\sin(x)$  là (Các hệ thức Taylor )
- Bước : Tính toán dựa trên biểu thức Taylor.

***“Ở đây khi tính  $\sin(x)$  máy tính sẽ tính theo công thức của Taylor cho phép biến đổi dạng lượng giác về dạng mũ ! để máy tính dễ thực thi hơn “***

Câu 5: làm thế nào để có thể tính toán được thời gian thực thi của một lệnh cụ thể trên nhân xử lý ?

- Họ sẽ sử dụng phương pháp tính trung bình :

Cho lệnh thực hiện trên nhiều chương trình khác nhau so đó lấy trung bình ( thời gian thực thi + thời gian trễ ) của nhiều lần chạy khác nhau .

Câu 6 : Lấy ví dụ một tập kiến trúc lệnh trên một nhân xử lý mà em biết :

- Kiến trúc x32 của Intel

Câu 7 : Công suất tiêu thụ của : VIA C3 processor. là bao nhiêu ?

Khoảng 7.5 watts

Câu 8 : xác định điện thế mức logic của VIA C3 processor.



includes latency and overhead. This is a statistical measure.

Q6. All x86 family instructions will work.

Q7. around 7.5 watts

Q8.

Parameter	Min	Max	Units	Notes
$V_{IL}$ – Input Low Voltage	-0.58	0.700	V	
$V_{IH,1.5}$ – Input High Voltage	$V_{REF} + 0.2$	$V_{TT}$	V	(2)
$V_{IH,2.5}$ – Input High Voltage	2.0	3.18	V	(3)
$V_{OL}$ – Low Level Output Voltage		0.40	V	@ $I_{OL}$
$V_{OH}$ – High Level Output Voltage		$V_{CMOS}$	V	(1)
$I_{OL}$ – Low Level Output Current	9		mA	@ $V_{CL}$
$I_{IL}$ – Input Leakage Current		$\pm 100$	$\mu A$	
$I_{LO}$ – Output Leakage Current		$\pm 100$	$\mu A$	

Q9. Refer Text

Q10. Refer Text

Câu 9 : Làm thế nào để đánh số pin lên con CHIP ??? ! ( Tao chịu ) hỏi ngu VCL ! How do you number the pins in an EPGA chip?

Fig. 8.6 The Bottom View of the Processor

Câu 10: Lợi thế của công nghệ làm nhân chip mới ( EPGA ) so với công nghệ nhân chip cũ là (PGA) là nó tiết kiệm được không gian phần cứng



# Lesson 9

## General Purpose Processors – II





## Nội dung:

- Tín hiệu của một bộ xử lý tổng thể (General Purpose Processor )
  - Mul
  - Add
  - Control
    - Bus
    - Status
  - Sleep
  - Interrupts

## 1. Mul

Một phương pháp truyền tải dữ liệu kỹ thuật số mà mất tín hiệu từ nhiều nguồn khác nhau, phân chia chúng thành mảnh mà sau đó được đặt định kỳ vào các khe thời gian , ( cái xung clock ) truyền tín hiệu xuống một con đường duy nhất và ráp các khe thời gian trở lại thành tín hiệu phức tạp trên cái "điều khiển kết thúc truyền

P/S : như truyền ra LCD 8051

## 2. Add

## 3. Control

### a. Bus

- i. Address Signals
- ii. Data Signals
- iii. Control Signals
  1. Memory Write
  2. Memory Read
  3. I/O Read
  4. I/O Write

### b. Status

- i. Bus Transaction Control ( điều khiển)
  1. Gồm Master và Slave



2. Master yêu cầu bằng cách gửi địa chỉ, Slave gửi Data tại địa chỉ lại cho Master hoặc nhận Data ghi vào địa chỉ của Master gửi
- ii. Bus Arbitration Control (Phân xử)
  1. Bus yêu cầu
  2. Bus lấy tín hiệu
  3. Lock : khóa khi yêu cầu ko thành công

## 4. Sleep

### a. "Stop Grant"

- i. CPU có power duy trì, nhưng ko làm vc
- ii. Standby trong windows

### b. "Suspend to RAM"

- i. Nội dung trong thanh ghi được đưa vào RAM, tắt điện CPU
- ii. Tắt cung cấp điện cho CPU, chỉ cung cấp điện cho RAM
- iii. Hệ thống dễ xảy ra lỗi
- iv. Sleep trong windows

### c. Suspend to Disk

- i. Nội dung trong thanh ghi đưa vào ram, rồi RAM đưa vào đĩa cứng, nên giá trị tiến trình vẫn còn
- ii. Ngưng cung cấp điện cho CPU lẫn RAM
- iii. Hibernate trong windows

### d. Soft Off

- i. Hệ thống ngừng hoạt động
- ii. Một số nguồn điện còn cung cấp cho các thiết bị nhất định để tạo xung đánh thức
- iii. VD: hỗ trợ khởi động lại từ USB hoặc LAN
- iv. Shutdown trong windows

### e. Processor "C" power states

- i. Là bộ xử lý điện năng thường thấy trong laptop
- ii. CPU điều chỉnh giữa tiêu thụ điện và hiệu suất làm việc (không phải hiệu năng hay độ trễ), nghĩa là CPU có thể làm việc lâu hơn, chứ không phải mạnh lên



## 5. Interrupts

- a. Là tín hiệu bên ngoài CPU cho trường hợp khẩn cấp

### Q&A

#### Q2: POST là gì ?

Là chế độ tự động kiểm tra sự hoạt động đúng đắn của Đĩa cứng, CDROM, đĩa mềm..v.v. Khi khởi động

#### Q3: Mô tả các chế độ tiết kiệm năng lượng ?

Phần trên

#### Q4: Laptop, desktop và motor control khác nhau ntn?

Laptop: có bộ xử lý phức tạp với mức tiêu thụ điện ít, có nhiều chế độ tiết kiệm điện khác nhau

Desktop: Bộ xử lý ở cường độ cao, không giới hạn nguồn điện.

Motor control: Là một bộ xử lý cực kì đơn giản và ít tốn điện cho một chức năng nào đó với hệ thống xử lý thời gian thực

#### Q5: Tại sao lại giảm điện từ 5 xuống còn 3.5V

Giảm nhiễu nhưng tăng tiếng ồn

#### Q6: Tại sao phải sử dụng nguồn tốt?

Vì nó ảnh hưởng trực tiếp lên CPU, nếu nguồn ko tốt có thể gây ra lỗi dẫn đến mất dữ liệu



# Lesson 10

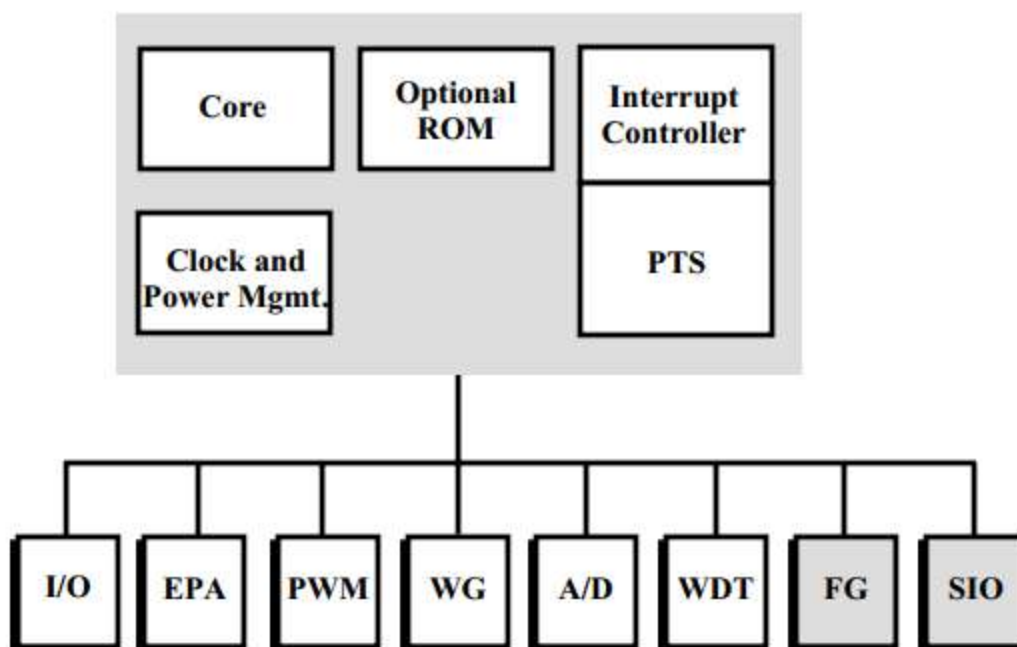
## Embedded Processors – I



## 1. Introduction

- a. Phân biệt “Microcontroller” và “general purpose microprocessors”
  - i. **Microcontroller** thường được kết hợp với các ứng dụng nhúng
  - ii. *Microprocessor* thường được kết hợp như máy tính desktop
  - iii. **Microcontroller** có phân cấp bộ nhớ đơn giản
  - iv. **Microcontroller** yêu cầu hạn chế điện năng
  - v. **Microcontroller** thường là 8 bit tới 16bit đơn giản, *General purpose Microprocessors* thường là 32bit-64bit phức tạp

## 2. Kiến trúc 1 vi điều khiển điển hình (80x96)



PTS: Peripheral Transaction Server; (Giao tiếp ngoại vi)

I/O: Input/Output Interface; ( Input/Output )

EPA: Event Processor Array; (Xử lý mảng )

PWM: Pulse Width Modulated Outputs; (Điều chế xung đầu ra)

WG: Waveform Generator; (Tạo xung)

A/D- Analog to Digital Converter; (Chuyển Analog sang Digital )

FG: Frequency Generator; ( Phát tần số )

SIO: Serial Input/Output Port ( Cổng vào ra nối tiếp )

## 3. Kết luận



## 4. Q&A

### a. Micro Engine là gì ?

Là nơi mà instructions được chia là làm nhiều micro-instruction nhỏ hơn để thực thi

Microprogramming là một trong những bước đột phá quan trọng cho phép các kiến trúc hệ thống thực hiện các hướng dẫn phức tạp trong phần cứng. Để hiểu microprogramming là gì, đầu tiên xem xét một phương pháp khác: thực hiện trực tiếp

Với thực hiện trực tiếp, máy tìm một instruction trong memory và tự nạp nó vào một bộ điều khiển phần cứng. Vùng điều khiển này lấy instruction như là một input và làm hoạt động một số sơ đồ mạch theo yêu cầu cụ thể

ưu điểm của thực hiện lệnh trực tiếp là :

nhANH, không có lệnh trừu tượng hoặc quá trình dịch lệnh, máy chỉ được giải mã và thực hiện các instruction bên trong phần cứng

Nhược điểm: tốn diện tích, không gian. Cần nhiều khối thực thi cho nhiều lệnh, do đó thư viện chương trình sẽ nặng, phức tạp, tốn nhiều thời gian thực hiện. Do đó các lệnh trực tiếp cho CISC không khả thi khi số lượng transistor có giới hạn

-----

với "vi chương trình" (microprogramming), nó giống như là một CPU mini bên trong CPU. Khối điều khiển là những "khối vi lệnh" mà nó thực hiện những "vi lệnh". Nhà thiết kế CPU dùng những "vi lệnh" để viết những "vi chương trình" và được cất vào một "bộ nhớ điều khiển đặc biệt" (!). Khi một lệnh chương trình bình thường được gọi từ bộ nhớ và nạp vào "khối vi lệnh", "khối vi lệnh" thực hiện các "vi chương trình con" đã được tích hợp. "Vi chương trình con" sẽ hướng dẫn các đơn vị chức năng khác nhau việc cần phải làm

Như bạn có thể đoán, trong microcode bắt đầu là một cách khá chậm chạp trong cách làm việc. Các ROM dùng cho bộ nhớ kiểm soát đã được nhanh hơn từ bộ nhớ chính core-based khoảng 10 lần, do đó "khối vi lệnh" có thể ở đủ xa về phía trước để cung cấp đúng hiệu



suất. Khi công nghệ vi phát triển , tuy nhiên , nó đã nhanh hơn và nhanh hơn . ("khối vi lện " trên các CPU hiện nay là khoảng 95 % nhanh như thực hiện trực tiếp ).từ khi công nghệ "vi" đã nhận được tốt hơn và tốt hơn, nó có ý nghĩa nhiều hơn và nhiều hơn nữa để chỉ chuyển chức năng từ phần mềm (chậm hơn và đắt tiền hơn ) đến ( nhanh hơn và rẻ hơn) phần cứng . Vì vậy, số lượng lệnh ISA ngày càng tăng , và số lượng chương trình chương trình đã giảm

Như microprograms đã lớn hơn và lớn hơn để chứa các instruction ngày càng tăng ,Tuy nhiên , một số vấn đề nghiêm trọng bắt đầu xuất hiện. Để giữ cho hiệu suất tăng , vi code phải được tối ưu hóa cao nhưng không được thiếu hiệu quả , và nó phải cực kỳ nhỏ gọn để giữ cho bộ nhớ giảm chi phí. Và kể từ khi chương trình vi là quá lớn lúc này , nó đã trở thành nhiều khó khăn hơn để kiểm tra và sửa lỗi . Kết quả là , các vi code được vận chuyển với máy thường có lỗi và phải được vá nhiều lần trong lĩnh vực này . Đó là khó khăn liên quan đến việc sử dụng vi điều khiển đó thúc đẩy Patterson và những người khác bắt đầu đặt câu hỏi liệu việc áp dụng tất cả các, instruction tỉ mỉ phức tạp trong microcode đã thực sự sử dụng tốt nhất nguồn lực transistor giới hạn hay chưa .

## b. WDT là gì ?

Đó là một cơ chế can thiệp không an toàn nếu một hệ thống ngừng hoạt động . Một giờ phần cứng đó là định kỳ thiết lập lại bằng phần mềm . Nếu lỗi phần mềm hay bị treo, WDT sẽ được kích để toàn bộ hệ thống sẽ được thiết lập lại tự động

***watch dog unit chứa một watch dog timer***

WDT : là một **thiết bị hoặc thẻ điện tử** mà thực hiện một hoạt động cụ thể sau một thời gian nhất định nếu có điều gì sai với một hệ thống điện tử và các hệ thống không tự khắc phục lỗi

Một vấn đề phổ biến là để một máy tính hoặc hệ điều hành ngừng lại nếu hai bộ phận hoặc các chương trình xung đột , hoặc , trong một hệ thống điều hành , nếu xảy ra khó khăn quản lý bộ nhớ . trong một



số trường hợp , hệ thống sẽ phục hồi chính nó , nhưng điều này có thể mất một khoảng thời gian dài không xác định. Một WDT có thể được lập trình để thực hiện một "**warm boot**" ( khởi động lại hệ thống) sau một số giây trong đó một chương trình hay máy tính không đáp ứng sau hành động click chuột hoặc gõ phím bàn phím gần nhất . các timer cũng có thể được sử dụng cho các mục đích khác

*ví dụ, để kích hoạt nút refresh ( hoặc tải lại ) trong một trình duyệt Web nếu một trang web không đầy đủ tải sau một thời gian nhất định sau sự xâm nhập của một Uniform Resource Locator ( URL )*

Một WDT chứa một truy cập kỹ thuật số mà đếm xuống 0 ở một tốc độ không đổi từ một số định sẵn . Tốc độ truy cập được giữ ổn định bởi một mạch tạo xung . Nếu đếm về zero trước khi hồi phục máy tính , một tín hiệu được gửi đến các mạch được chỉ định để thực hiện các hành động mong muốn





# Lesson

# 11

# Embedded Processors - II



## 1. Introduction

- Vi điều khiển được yêu cầu hoạt động trong thế giới thực mà không cần nhiều giao diện mạch.
- Tín hiệu in-out put gồm cả analog và digital. Tín hiệu digital có thể được truyền song song hoặc nối tiếp, các mức điện áp có thể khác nhau.
- Bộ vi xử lý thông thường sẽ có Digital Input / Output , Timer và đường nối tiếp Digital Input/Output. Một số vi điều khiển cũng hỗ trợ bộ chuyển đa kênh analog to digital converter ( ADC ) cũng như bộ chuyển DAC
- Như vậy tín hiệu analog đầu vào và đầu ra chân cũng có mặt trong các đơn vị vi điều khiển điển hình . Đối với bộ nhớ bên ngoài và I / O chip địa chỉ cũng như các dòng dữ liệu cũng được hỗ trợ.

## 2. Tín hiệu của Intel MCS 96

### Some Specifications of the Processor

một số thông số kĩ thuật của bộ vi xử lý

Frequency of Operation: 40 MHz

2 Mbytes of linear address space

1 Kbyte of register RAM

3 Kbytes of code RAM

8 Kbytes of ROM

2 peripheral interrupt handlers (PIH) ngoại vi xử lý ngắt

6 peripheral interrupts ngắt ngoại vi

83 I/O port pins

2 full-duplex serial ports with baud-rate generators

2 cổng nối tiếp song công hoàn toàn với máy phát điện baud -rate

Synchronous serial unit

8 pulse-width modulator (PWM) outputs with 8-bit resolution

16-bit watchdog timer

Sixteen 10-bit A/D channels

Programmable clock output signal

## 3. Kết luận

## 4. Q&A

- Q2: Tại sao phải cung cấp điện cho chip tại nhiều điểm mà không phải tại một điểm duy nhất ?**

Để đảm bảo các tính chất sau :



- điện áp tại các thiết bị ( transistors và cells) tốt hơn là thiết lập một mục tiêu đặc biệt để thay thay đổi các điều kiện trong thiết kế . Điều này là để đảm bảo hiệu suất hoạt động đúng đắn của các mạch ở cấp độ dự kiến.
- hiện nay được cung cấp bởi một pad , pin , hoặc bộ điều chỉnh điện áp trong giới hạn quy định dưới bất kỳ các điều kiện quy định tải. Các yêu cầu cần thiết :
  - -a)cho không vượt quá không gian thiết kế
  - -b)để phân phối các dòng điện thống nhất hơn trong số các thiết bị, do đó L di / dt biến điện áp do điện cảm ký sinh trong các gói của chất nền , mảng lưới bóng (trans ), và đường điện được giảm thiểu



# Lesson

# 12

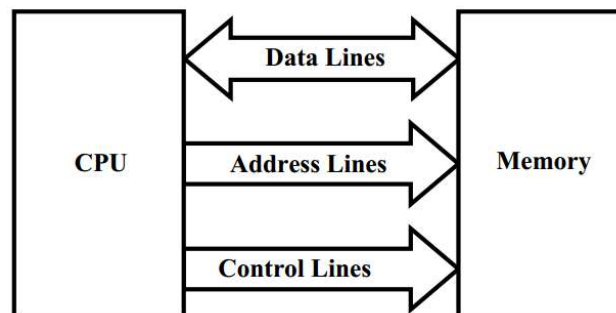
# Memory- Interfacing



## Giao tiếp Memory

### 1. Introduction

- Hầu hết memory nằm bên trong VĐK do chương trình ít đòi hỏi vùng nhớ lớn nhưng một số VĐK có bộ nhớ nằm bên ngoài gây tăng điện và tăng kích thước
- CPU giao tiếp với Memory qua 3 đường



### 2. Giao diện bộ nhớ ngoài họ vi điều khiển PIC18F8XXX

Nhiều họ vđk có cả bộ nhớ onchip và bộ nhớ ngoài. Khi đó thì on-chip memory có thể được lập trình như là một loại bộ nhớ flash.

Lập trình thanh ghi đặc biệt bên trong vđk để có thể sử dụng bộ nhớ ngoài theo chế độ khác nhau

#### a. Microcontroller Mode

- Enable on-chip, Disable External nhưng đọc đc bit 0 ở External

#### b. Microprocessor Mode

- Disable on-chip, Enable External

#### c. Microprocessor with Boot Block mode

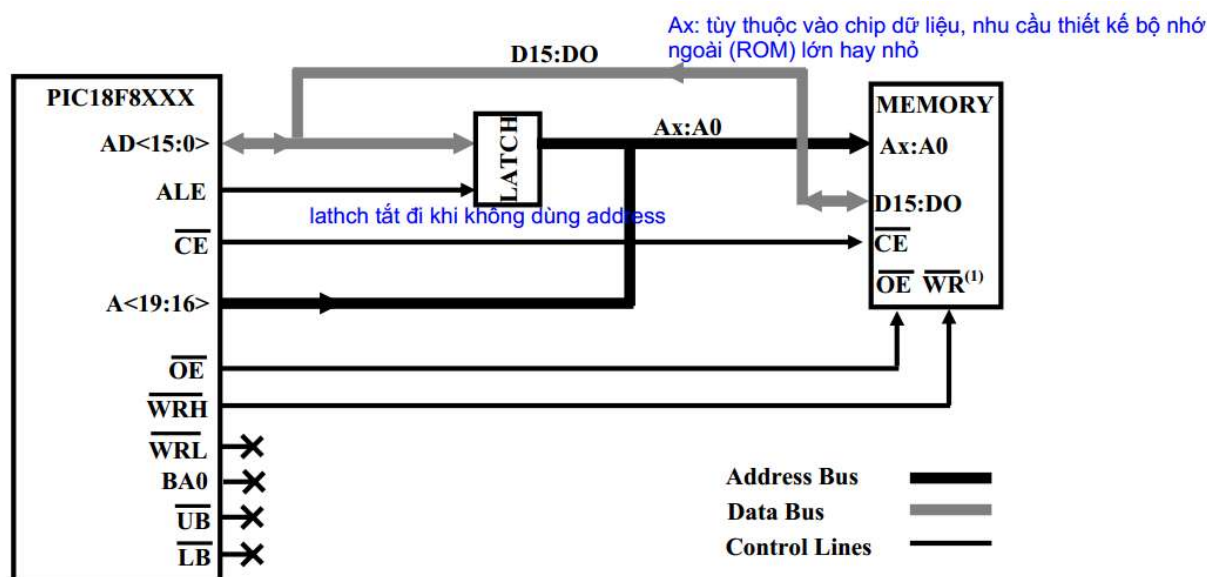
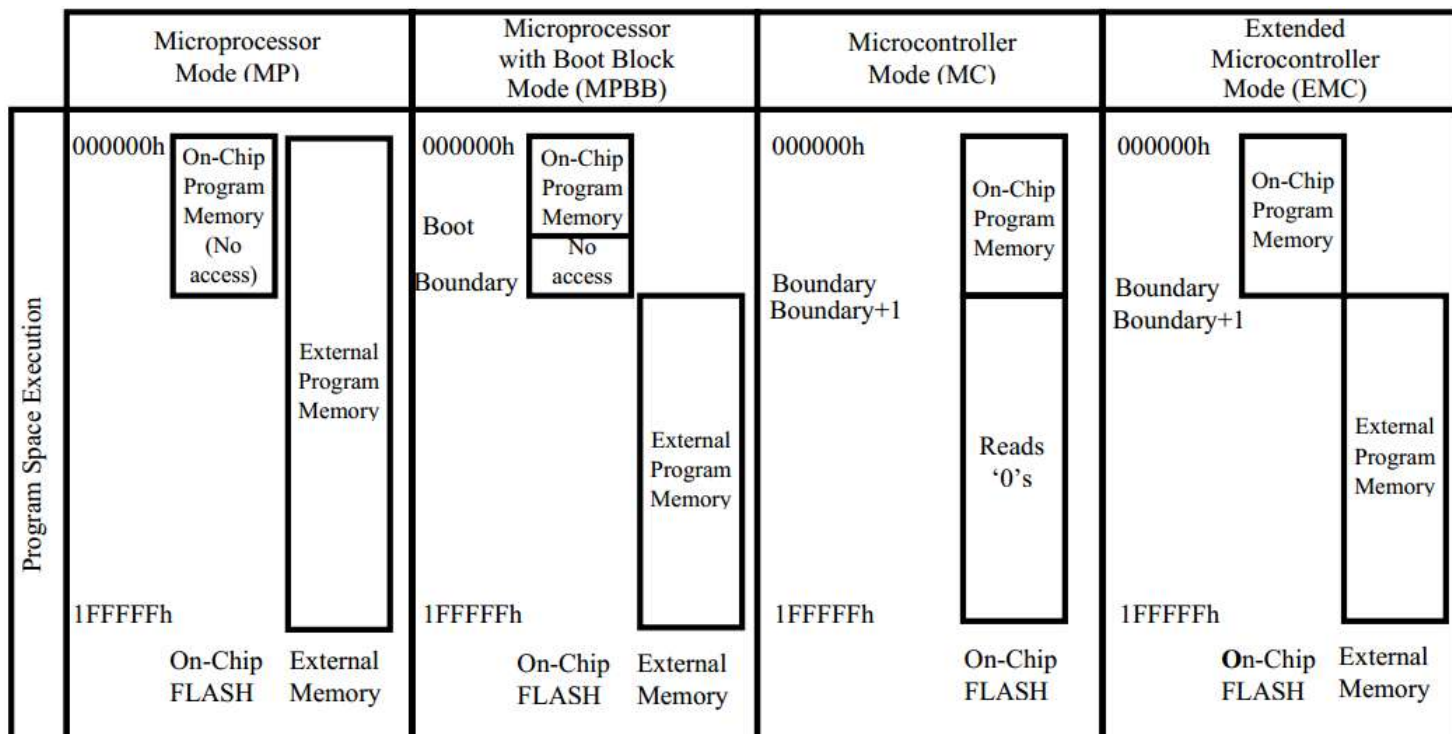
- Enable một phần (phần đầu ) on-chip ,Enable external

#### d. Extended Microcontroller Mode

- Enable on-chip, Enable External

Chế độ	VXL	VXL với khối Boot	VĐK	VĐK mở rộng
Vùng truy cập	External	External và phần đầu on-chip ghép lại (on-chip nằm trên )	On-chip nhưng đọc đc bit 0 ở External	On-chip và External

# Hệ thống nhúng : Lesson 12







# Lesson

# 14

# Timers





## Mục tiêu bài giảng:

- Timer và cơ bản về counter
- Các chế độ khác nhau của Timer operation
- Timer 8051
- Một chương trình timer 8253
- Watchdog Timer and Watchdog circuit

## Giới thiệu

Các thiết bị ngoại vi của một bộ xử lý nhúng có thể ở trên cùng một chip như bộ vi xử lý hoặc có thể được kết nối bên ngoài

### *Timer*

Là một thiết bị ngoại vi rất phổ biến và hữu ích. Nó được sử dụng để tạo ra các sự kiện tại thời điểm cụ thể hoặc đo thời gian của các sự kiện cụ thể ở bên ngoài để xử lý. Nó là một thiết bị lập trình, tức là khoảng thời gian có thể điều chỉnh bằng cách viết mẫu bit cụ thể đối với một số thanh ghi gọi là thanh ghi timer điều khiển

### *Counter*

Là một phiên bản tổng quát hơn của bộ đếm thời gian. Nó được sử dụng để đếm các sự kiện ở dạng xung.

(đoạn counter này tao ko biết nó đếm sao)

### *Timer của vi điều khiển 8051*

8051 được trang bị 2 timer, cả 2 đều có thể được kiểm soát, thiết lập, đọc và cấu hình riêng. 8051 giờ có 3 chức năng chung:

- 1) Giữ thời gian và/ hoặc tính toán khoảng thời gian giữa các sự kiện
- 2) Đếm lại các sự kiện
- 3) Tạo ra baud rates cho cổng nối tiếp

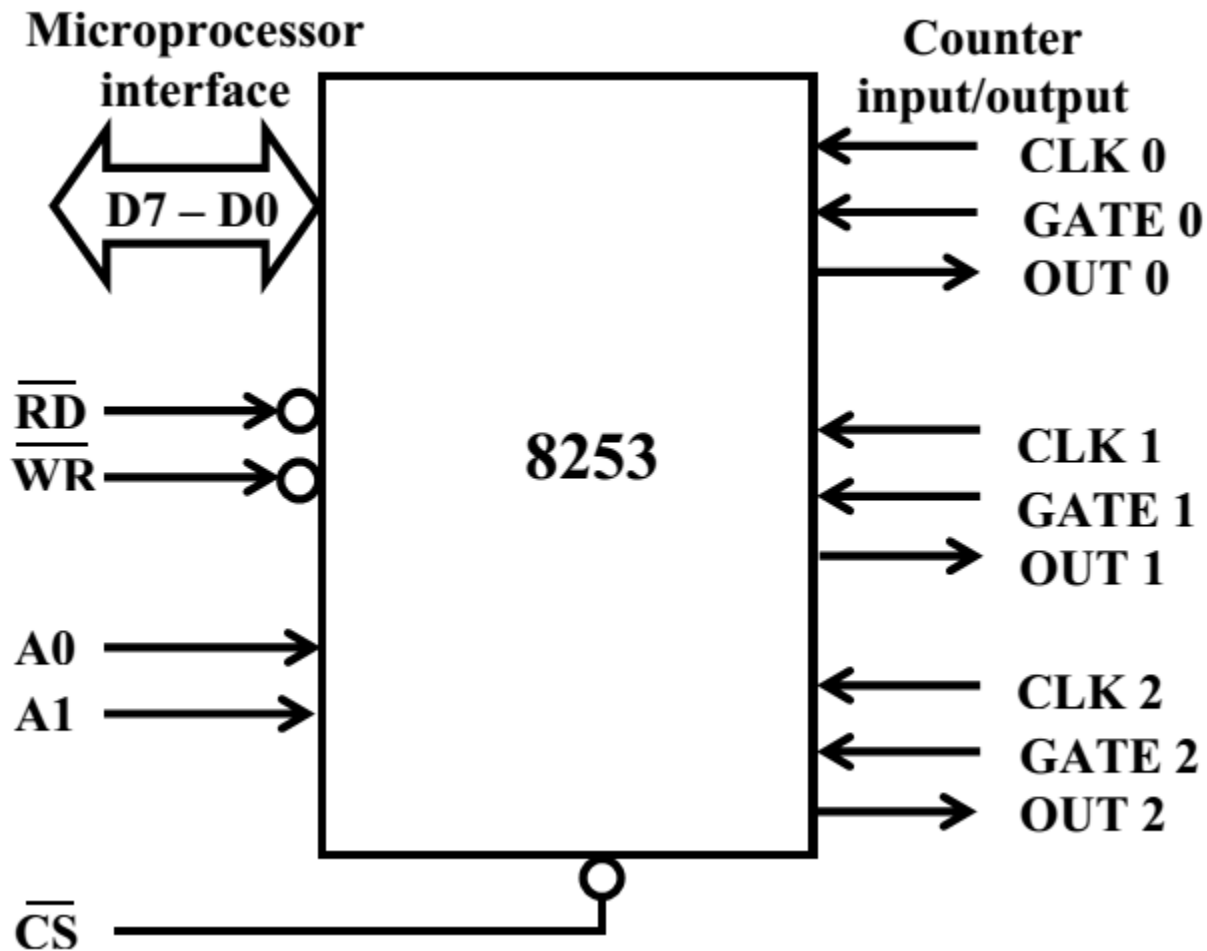
8051 có 2 timer là timer0 và timer1. Hai timer chia sẻ hai thanh ghi chức năng đặc biệt (SFR) (TMOD và TCON) để điều khiển tính giờ. Và mỗi timer cũng có 2 SFR chỉ dành riêng cho bản thân (TH0/TL0 và TH1/TL1)

Các chế độ của timer 8051 như lúc trước nên khỏi nói

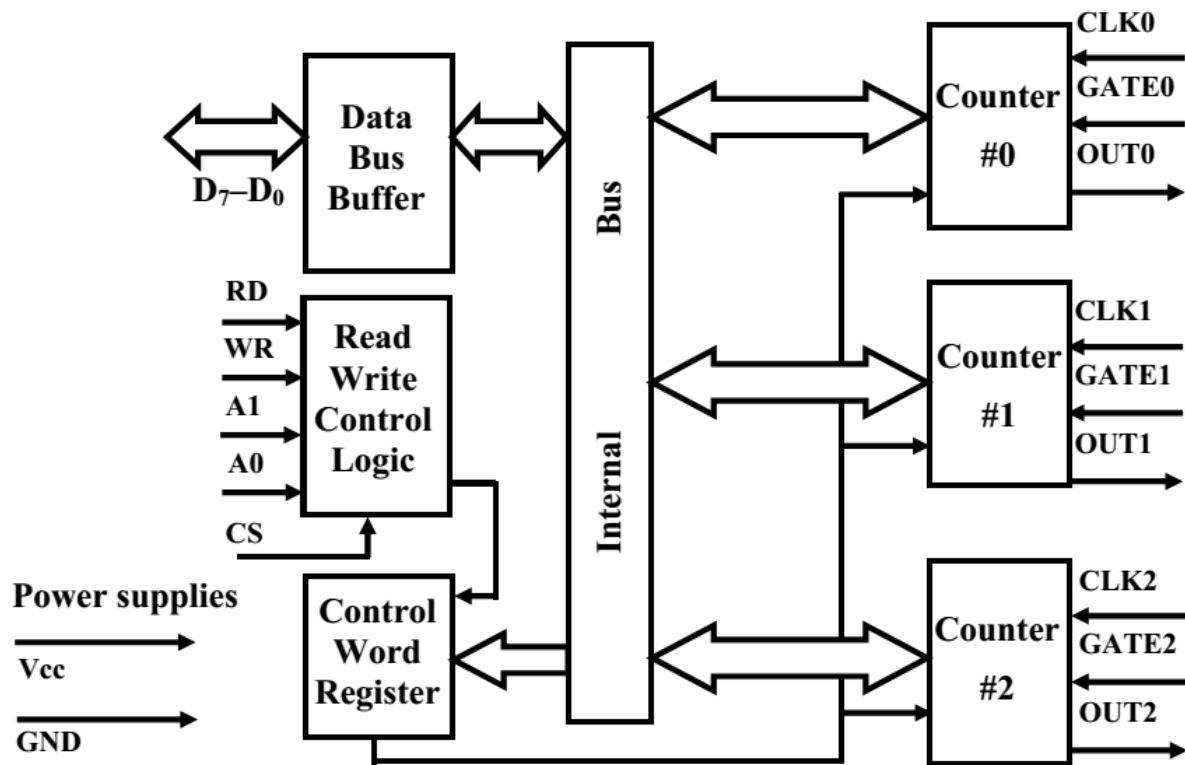


## *The Programmable Interval Timer 8253*

Các chân của timer



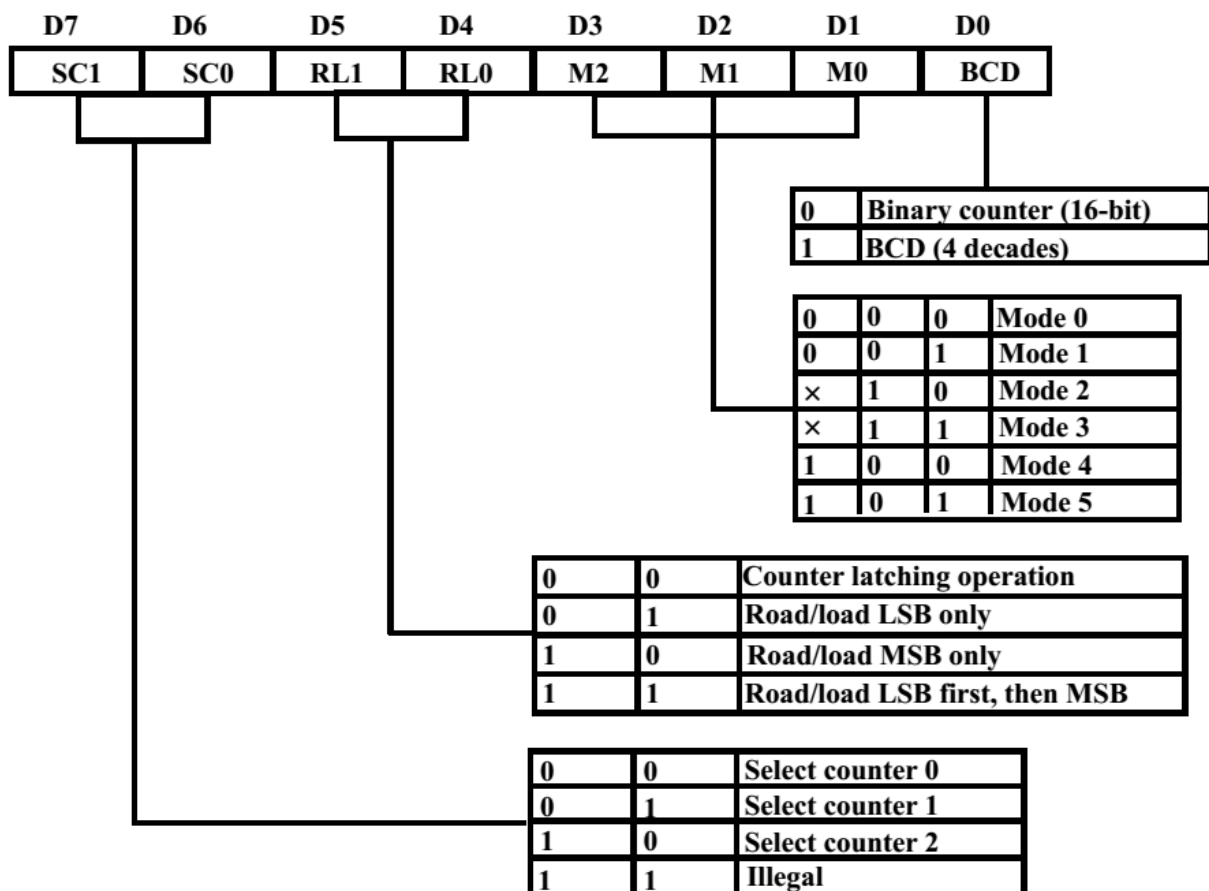
Mỗi counter có 2 đầu vào, clock và gate và 1 đầu ra. Clock là tín hiệu giúp đếm bằng cách giảm một giá trị được tải trước trong thanh ghi counter tương ứng. Các gate server như là 1 input enable. Nếu gate được duy trì thấp thì counter được disable. Biểu đồ thời gian sẽ giải thích chi tiết về các chế độ khác nhau về hoạt động của timer.



*Thanh ghi điều khiển timer của 8253*



$\overline{\text{CS}}$	A1	A0	Port
0	0	0	Counter 0
0	0	1	Counter 1
0	1	0	Counter 2
0	1	1	Control register





## ***Các chế độ hoạt động của 8235***

Mode 0 – ngắt của thiết bị đầu cuối

Mode 1 - lập trình 1 lần

Mode 2 – rate generator

Mode 3 –sóng vuông rate generator

Mode 4 –phần mềm strobe được kích hoạt

Mode 5 –phần cứng strobe được kích hoạt

## ***Chi tiết mỗi mode:***

### **Mode 0:**

- Đầu ra cao sau khi các thiết bị đầu cuối đạt
- Counter dừng nếu gate thấp
- Thanh ghi timer count được nạp vào 1 số (số 6) khi WR thấp
- Các counter bắt đầu đếm theo cạnh xuống của mỗi xung clock
- Các đầu ra (out) cao khi đếm tới 0
- Như hình b nếu gate thấp thì bị treo tại 3 và đếm tiếp khi gate cao

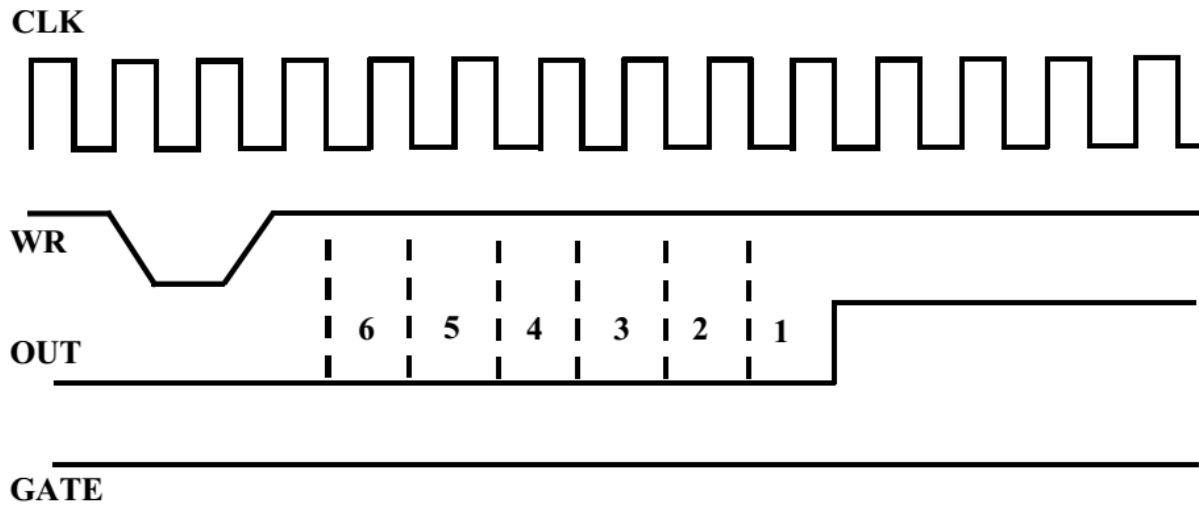


Fig. 14.10(a) Mode 0 count when Gate is high (enabled)

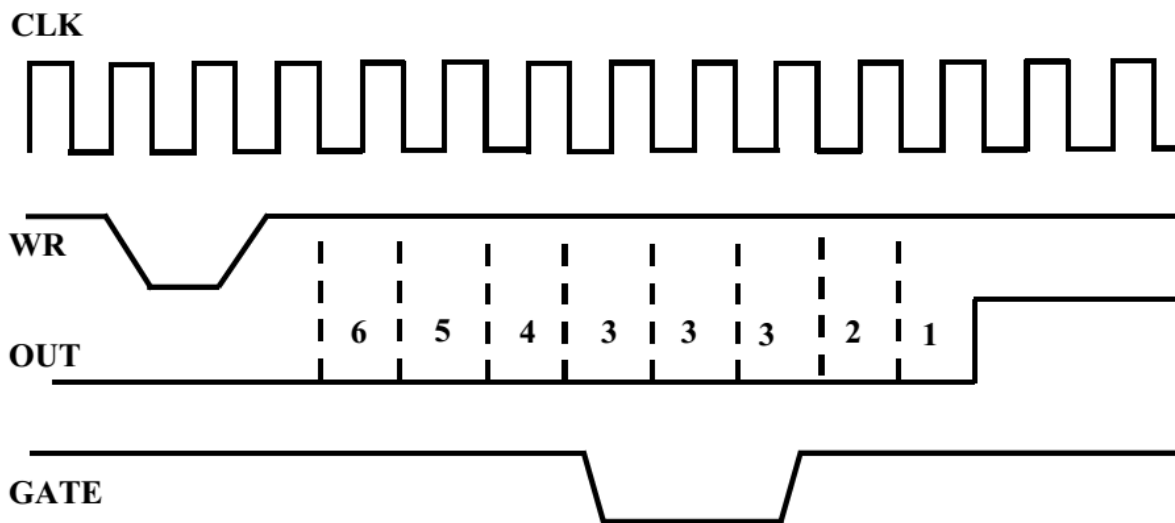
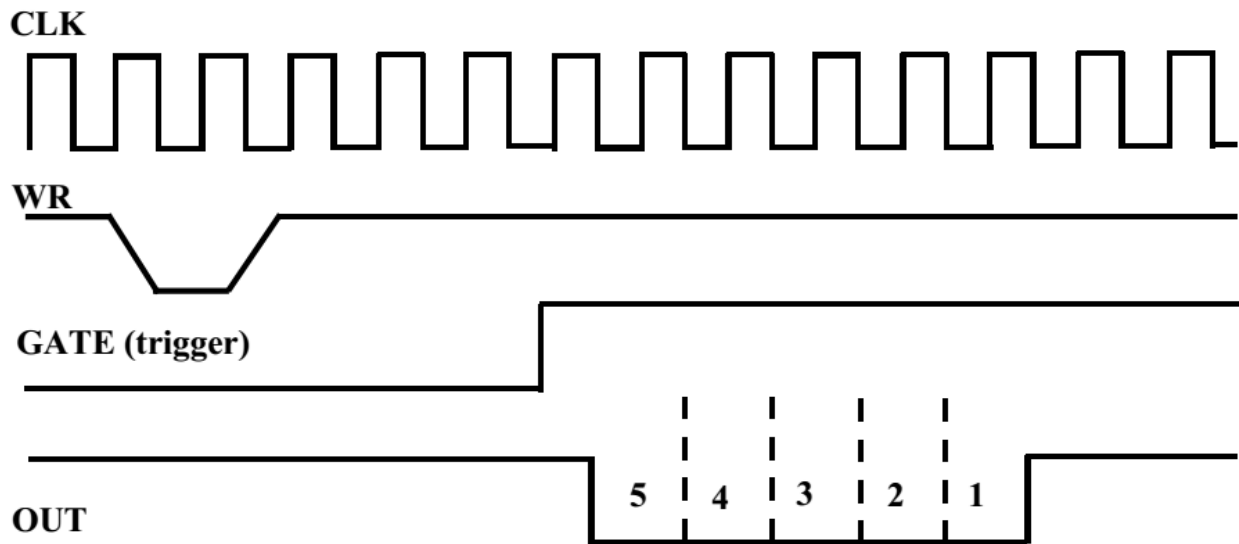


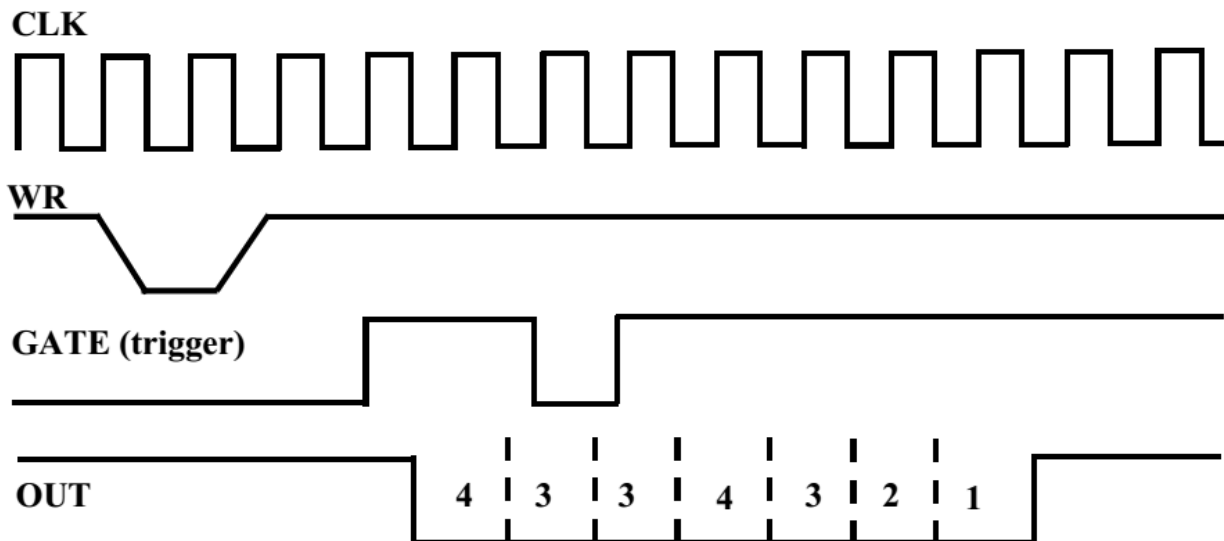
Fig. 14.10(b) Mode 0 count when Gate is low temporarily (disabled)

## Mode 1:

- Các bộ đếm bị vô hiệu hóa nên xung gate thấp đi trong giây lát
- Khi gate cao để bắt đầu đếm thì phải đợi xung clock kích cạnh xuống nó mới đếm
- Thanh ghi counter được nạp 1 giá trị đếm như trước đó là 5
- Khi gate thấp thì sẽ bị delay dừng đếm giữa chừng
- Bằng cách thay đổi giá trị này đếm thời gian của xung đầu ra có thể được thay đổi. nếu gate thấp trước khi đếm xuống



**Fig. 14.11(a) Mode 1** The Gate goes high. The output goes low for the period depending on the count



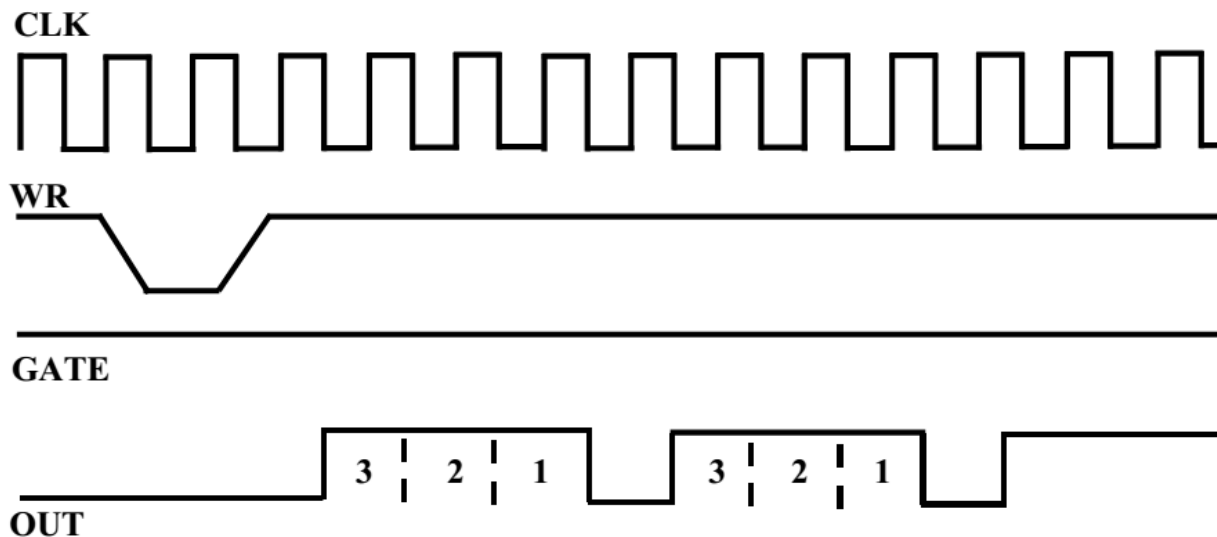
**Fig. 14.11(b) Mode 1** The Gate pulse is disabled momentarily causing the counter to stop.

## Mode 2:

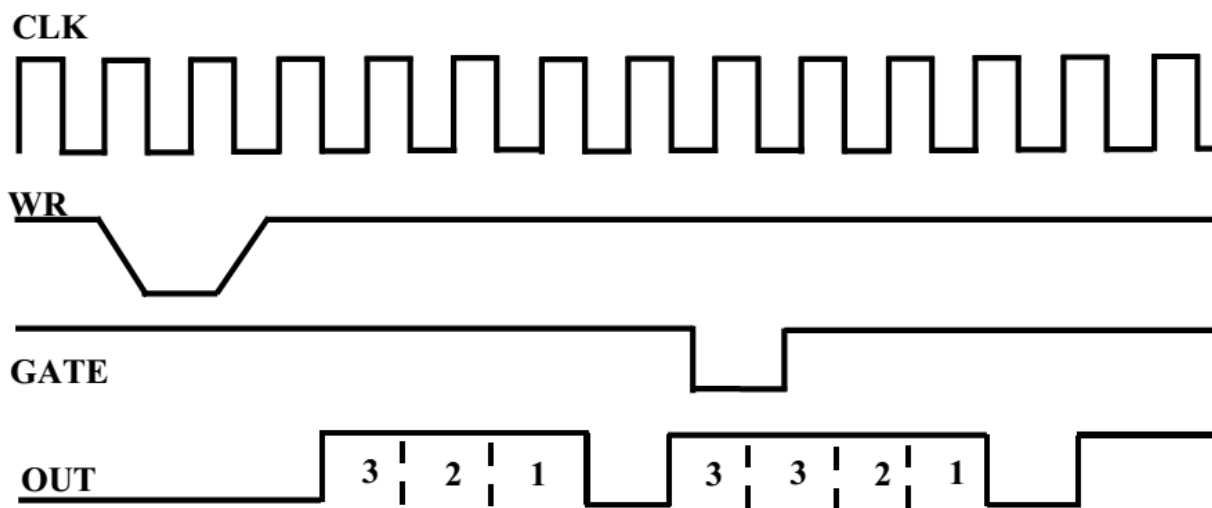
- Trong chế độ này nó hoạt động như 1 máy phát rate
- Bắt đầu đếm khi WR thấp
- Nạp giá trị đếm cho thanh ghi count trong TH này là 3



- Nó sẽ đếm ở mức cao và đếm khi xung clock kích cạnh xuống. khi đếm xong nó sẽ nghỉ 1 chu kì và đếm tiếp
- Đây là 1 hoạt động định kỳ
- Nó cũng sẽ bị delay nếu gate thấp



**Fig. 14.12(a) Mode 2 Operation when the GATE is kept high**



**Fig. 14.12(b) Mode 2 operation when the GATE is disabled momentarily.**

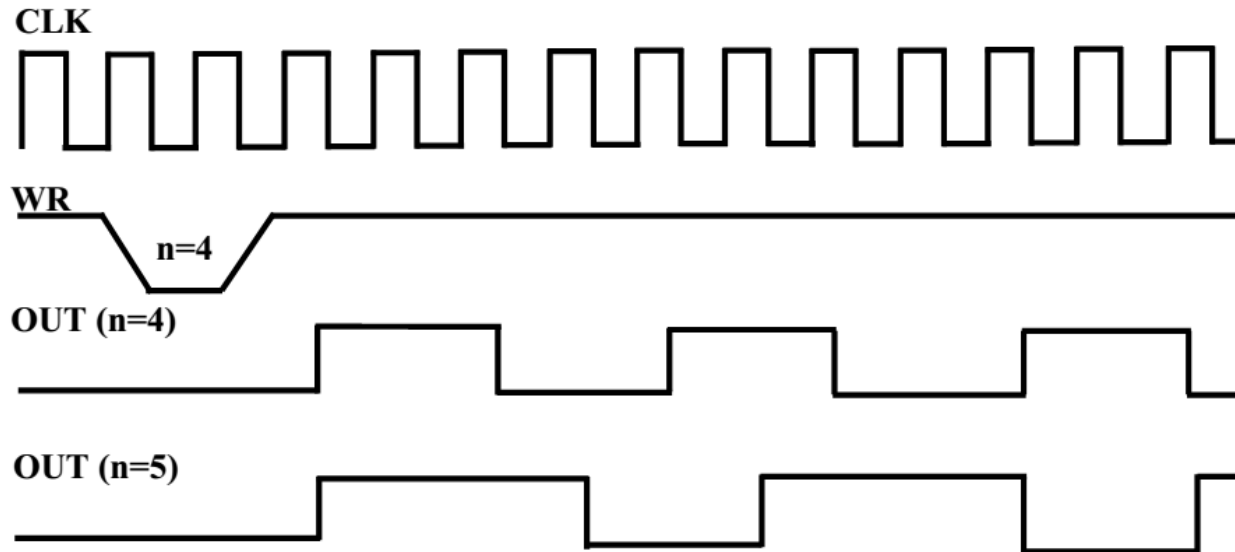
### Mode 3:

- Tương tự chế độ 2





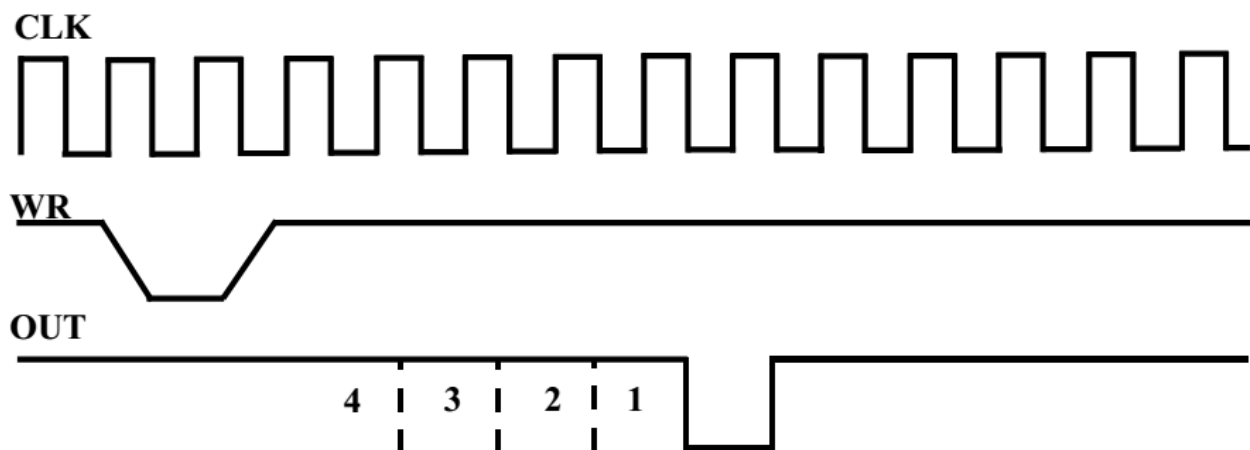
- Chỉ khác là nó tạo ra 1 lần song vuông đối xứng không giống như chế độ 2
- Gate ko có vai trò gì ở đây



**Fig. 14.13 Mode3 Operation: Square Wave generator**

### Mode 4:

- Giống các chế độ trên
- Chỉ khác là đầu ra sẽ thấp trong 1 chu kỳ khi đếm xong
- Nó vẫn bị delay nếu gate thấp



**Fig. 14.14(a) Mode 4 Software Triggered Strobe when GATE is high**



## Mode 5:

- Chế độ này khác với mấy chế độ trên là bắt đầu đếm khi gate mức cao
- Các đầu ra vẫn sẽ mức thấp 1 chu kỳ clock khi đếm xong

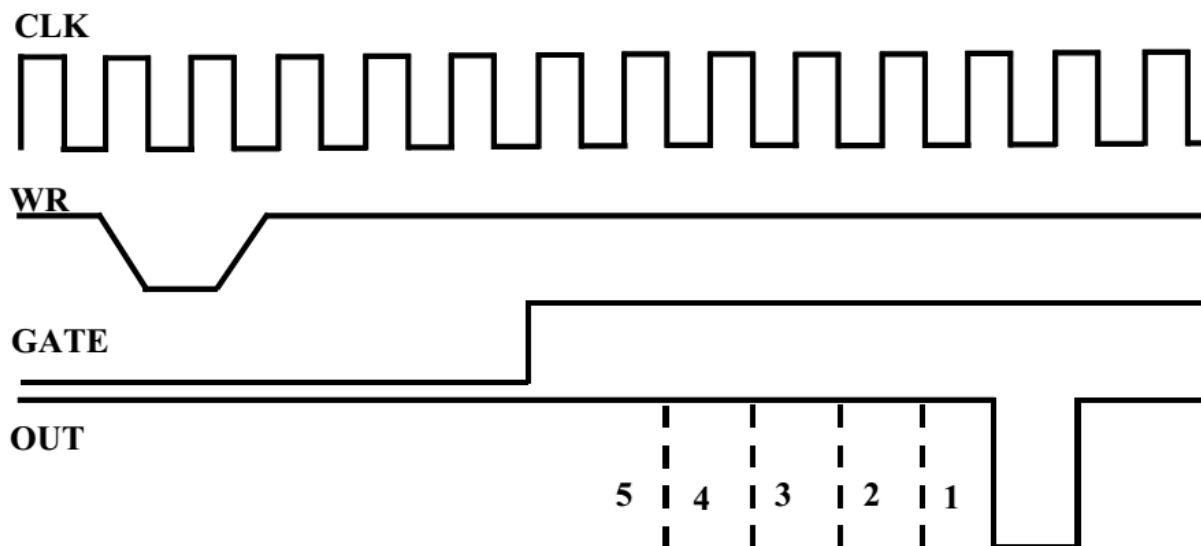


Fig. 14.15 Mode 5 Hardware Triggered Strobe

## Watchdog timer

Là một mạch tự động kích hoạt một thiết lập lại trừ khi hệ thống đang được theo dõi thường xuyên gửi các tín hiệu hold-off để watchdog

## Watchdog Circuit

Để đảm bảo một chương trình hoạt động bình thường thì một Watchdog Circuit được sử dụng

Chương trình được thiết lập lại định kì bằng các flip flop bên ngoài

Khi Watch Dog xảy tức là chương trình đó đang hoạt động không đúng cách và ngắt ngoại lệ xảy ra

Watch Dog Timer là một bộ đếm độc lập bên ngoài VXL

Khi bật bộ đếm Watch Dog Timer thì nó sẽ đếm từ 00 đến FF (tùy từng VXL) khi giá trị FF nhảy sang 0 (nghĩa là FF + 1) thì ngắt WD được bật reset lại chương trình.

Để không bị reset lại thì chương trình cần một ngoại lệ ngắt thiết lập lại giá trị đếm của WDT về 00 trong suốt chương trình, nếu chương trình lỗi, treo thì WDT sẽ không được set về 00 do đó sẽ bật chức năng reset của WD reset lại chương trình



# Lesson 15 Interrupts

Version 2 EE IIT, Kharagpur 2

---



## I. Nội dung

### II. .

- Giới thiệu
- Các loại ngắt
- PC-Program counter, P1-Port P2-Port,  $\mu$ C-Microcontroller
  - ❖ Ngắt điển hình (VD: 8051 )
- Interrupt Enables (Cho phép ngắt )
- Interrupt Priorities ( Độ ưu tiên ngắt )
- The Bus Arbitration ( Bus trọng tài )
- Priority Arbiter ( Trọng tài phân xử độ ưu tiên )
- Daisy Chain Interrupt ( ngắt vòng hoa cúc )
  - ❖ Trình bày một số ngắt của bộ điều khiển ngắt 8259
- Interrupt Request Register (IRR) and In-Service Register (ISR) (thanh ghi yêu cầu ngắt và thanh ghi phục vụ ngắt )
  - Resolver ( Bộ phân giải ngắt )
  - Interrupt Mask Register (IMR) ( Thanh ghi mặt nạ ngắt )
  - Data Bus Buffer (Bus dữ liệu đệm )
  - Read/Write Control Logic ( Điều khiển đọc/ ghi )
  - Phần mềm ngắt
  - Kết luận

### III. Q&A

---

## I. Nội dung

### II. .

- Giới thiệu
  - **Phương pháp cũ** : polled approach (thăm dò ý kiến ) : CPU “ask” từng ngoại vi xem có cần sử dụng CPU hay không.  
**Nhược điểm chính** : Nếu có nhiều ngoại vi thì vòng quét “ask” sẽ tăng lên nhiều làm giảm hiệu năng
  - **Phương pháp mới: ngắt** : CPU có thể làm việc của nó mà không cần phải tạo vòng lặp “ask”

Một tín hiệu thông báo cho một chương trình hay một thiết bị kết nối với bộ xử lý một sự kiện có xảy ra . Khi một bộ xử lý nhận được một tín hiệu gián đoạn, phải mất một hành động cụ thể tùy thuộc vào **độ ưu tiên và tầm quan trọng** của thực thể tạo ra các tín hiệu .



Tín hiệu gián đoạn có thể gây ra một chương trình để tự dừng nó tạm thời để phục vụ ngắt bằng cách **phân nhánh** vào một chương trình khác gọi là **Interrupt Service Subroutines** **thủ tục con** ( ISS ) cho các thiết bị quy định đó gây ra các ngắt

### ➤ Các loại ngắt

Ngắt được phân loại như sau:

- **Tùy vào cái làm ngắt**
  - Ngắt phần cứng
    1. Là ngắt gây ra bởi thiết bị kết nối
  - Ngắt phần mềm
    1. Là ngắt bằng những phần mềm để tạo ra ngắt theo mong muốn của người sử dụng
  - Trap (Bẫy)
    1. Phát hiện trường hợp ngoại lệ ( vd như chia cho 0)
- **Tùy thuộc vào cách phục vụ ngắt:**
  - **Ngắt cố định**
    1. Địa chỉ thanh ghi ngắt (ISR) cố định trong vxl, không thể thay đổi
    2. ISR được lưu trong địa chỉ **hoặc** con trỏ trỏ tới địa chỉ thực nếu không đủ byte lưu trữ
  - **Ngắt theo hướng**
    1. Ngoại vi cung cấp địa chỉ ngắt
    2. Thường dùng khi có nhiều ngoại vi dùng chung một đường bus
  - **Kết hợp ngắt cố định và ngắt theo hướng**
    1. Chỉ một chân ngắt
    2. Table trong bộ nhớ chứa địa chỉ thanh ghi ISR
    3. Thiết bị ngoại vi không cung cấp địa chỉ ngắt **nhưng** nó trỏ và bảng
      - Bit gửi từ ngoại vi ít hơn
      - Có thể thay đổi vị trí ISR mà không ảnh hưởng ngoại vi



- **Ngắt có mặt nạ và không có mặt nạ (Maskable vs. Non-maskable interrupts)**
  - Có mặt nạ : Lập trình viên có thể set bit để bộ xử lý bỏ qua ngắt
    1. Quan trọng khi khẩn khe về thời gian xử lý
  - Không có mặt nạ : pin ngắt không thể bỏ qua
    1. Dùng trong tình huống cấp bách, như mất điện, sao lưu dữ liệu ngay lập tức

## **Ví dụ : Interrupt Driven Data Transfer (Fixed Interrupt)**

### ➤ **PC-Program counter, P1-Port P2-Port, $\mu$ C-Microcontroller**

#### ❖ **Ngắt điển hình (VD: 8051 )**

- 8051 có 5 ngắt : 2 ngắt ngoài, 2 ngắt timer, 1 ngắt truyền thông nối tiếp
  - Do đó các điều kiện xảy ra ngắt là:
    - Vượt bộ đếm
    - Có yêu cầu nhận truyền thông nối tiếp
    - Có yêu cầu truyền truyền thông nối tiếp
    - 2 sự kiện ngắt ngoài
- **Interrupt Enables (Cho phép ngắt )**
  - Bit EA trong thanh ghi ngắt IE
- **Interrupt Priorities ( Độ ưu tiên ngắt )**
  - **Nhắc lại:** Ngắt cao có quyền ngắt ngắt thấp, 2 ngắt xảy ra đồng thời thì ngắt có độ ưu tiên cao làm trước, 2 ngắt có cùng độ ưu tiên xảy ra đồng thời thì xét yếu tố phụ (tùy từng vxl)

Interrupt Number	Interrupt Vector Address	Description
0	0003h	EXTERNAL 0
1	000Bh	TIMER/COUNTER 0
2	0013h	EXTERNAL 1
3	001Bh	TIMER/COUNTER 1
4	0023h	SERIAL PORT

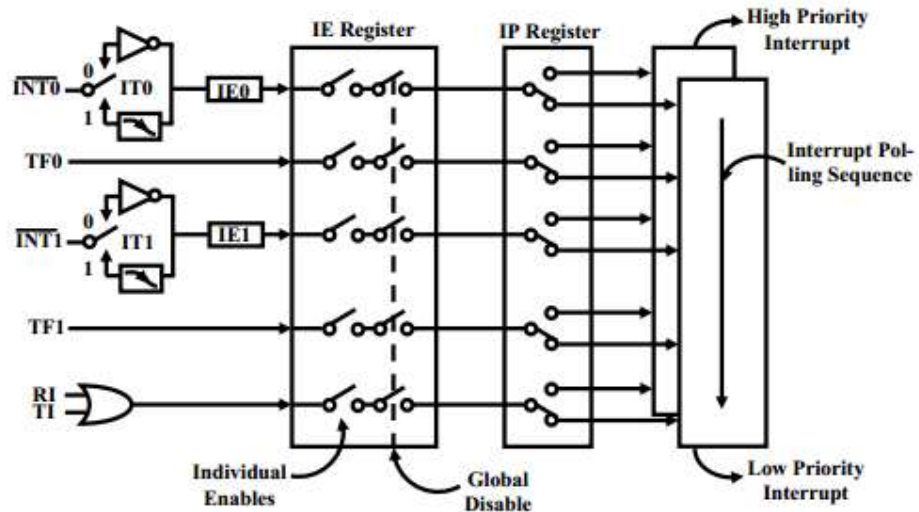


Fig. 15.4 8051 Interrupt Control System

➤ **The Bus Arbitration ( Bus trọng tài )**

- Cần thiết khi có nhiều ngoại vi : có 2 loại
  - Thứ tự Ngắt theo độ ưu tiên **Priority Arbiter**
  - Thứ tự Ngắt theo vòng **Daisy Chain Arbiter**
- **Priority Arbiter ( Trọng tài phân xử độ ưu tiên )**
  - Xét độ ưu tiên, cùng độ ưu tiên thì xét điều kiện phụ

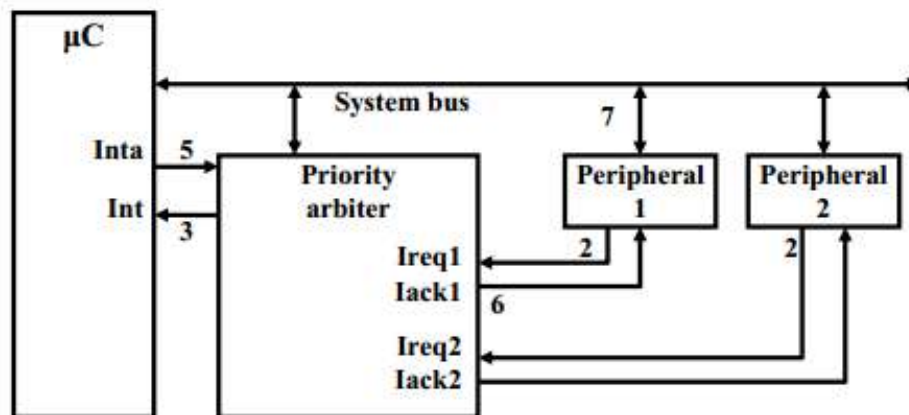


Fig. 15.5 The Priority Arbitration

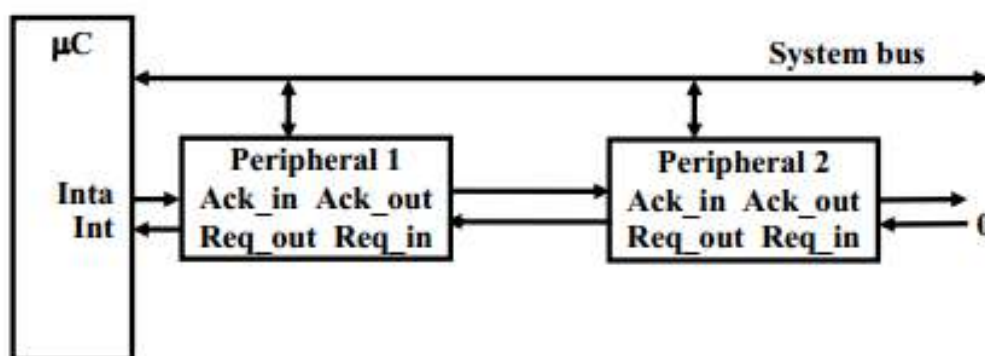
1. VXL chạy chương trình của nó
2. Peripheral 1 yêu cầu ngắt Ireq1, Peripheral2 yêu cầu ngắt Ireq2
3. Priority arbiter thấy rằng có ít nhất một Ireq yêu cầu ngắt, nó kích hoạt chân Int
4. VXL ngừng tiếp trình của nó và cất cả stack



5. VXL xác nhận Inta (bảo nó đã sẵn sàng)
6. Priority arbiter yêu cầu ngắt 1 bằng chân Iack1 đáp lại Peripheral1
7. Peripheral1 đưa địa chỉ ngắt vào system bus
8. Bộ VXL nhảy tới địa chỉ ngắt của ISR đọc bus data, ISR thực thi và trả về giá trị (tới đây hoàn tất phân xử ngắt )

○ **Daisy Chain Interrupt ( ngắt vòng hoa cúc )**

- Không xét độ ưu tiên mà xét xem thằng nào gần CPU nhất ( giống như xếp hàng )



**Fig. 15.6 The Daisy Chain Arbitration**

1. VXL chạy chương trình của nó
2. Bất kì thằng Peripheral nào cần phục vụ ngắt thì yêu cầu qua **Req out**. Thằng **Req out** này đưa tín hiệu sang thằng **Req in** của thiết bị đứng sau nó ( đầu ra **Req out** của thằng này là đầu vào **Req in** của thằng kia nó nối tiếp nhau)
3. Khi có ít nhất 1 thằng yêu cầu thì gửi đến chân **Int** yêu cầu ngắt
4. VXL dừng chương trình đang chạy và lưu lại
5. VXL gửi **Inta** báo sẵn sàng ngắt
6. **Inta** đi qua các chuỗi **Ack\_in**, **Ack\_out** cho đến khi nó thấy một cờ ngắt
7. Thiết bị ngắt gửi **Địa chỉ ngắt** tới VXL để ngắt
8. VXL nhảy tới địa chỉ ngắt trong thanh ghi ISR thông qua dữ liệu bus, ISR được thực thi và trả về
9. **Reset lại cờ ngắt**





### ❖ Trình bày một số ngắt của bộ điều khiển ngắt 8259

- **Interrupt Request Register (IRR) and In-Service Register (ISR) (thanh ghi yêu cầu ngắt và thanh ghi phục vụ ngắt )**
- **Priority Resolver ( Bộ phân giải ngắt )**  
Khởi logic này sẽ xác định các ưu tiên của các bit được đặt trong IRR . Các ưu tiên cao nhất là lựa chọn và đưa vào các bit tương ứng của các LSR trong chuỗi INTA .
- **Interrupt Mask Register (IMR) ( Thanh ghi mặt nạ ngắt)**  
Lưu trữ các bit vô hiệu hóa các đường ngắt được che đậy . Các IMR hoạt động trên đầu ra của IRR . “Mặt nạ” của một đầu vào ưu tiên cao hơn sẽ không ảnh hưởng đến yêu cầu ngắt của ưu tiên thấp hơn .
- **Data Bus Buffer (Bus dữ liệu đệm )**  
Có 3 trạng thái, Bộ đệm 8 bit 2 chiều được sử dụng để giao tiếp với PIC tới các System Data Bus  
Các word điều khiển và thông tin được truyền thông qua Data Bus Buffer
- **Read/Write Control Logic ( Điều khiển đọc/ ghi )**
- **Phần mềm ngắt**  
Đây được khởi xướng bởi các chương trình bằng cách thiết lập lệnh cụ thể . Khi gặp phải lệnh này CPU thực hiện một chương trình con phục vụ ngắt .
- **Kết luận**



## III. Q&A

### Question Answers

Q1. A computer system has three devices whose characteristics are summarized in the following table:

Device	Service Time	Interrupt Frequency	Allowable Latency
D1	150 $\mu$ s	1/(800 $\mu$ s)	50 $\mu$ s
D2	50 $\mu$ s	1/(1000 $\mu$ s)	50 $\mu$ s
D3	100 $\mu$ s	1/(800 $\mu$ s)	100 $\mu$ s

Service time indicates how long it takes to run the interrupt handler for each device. The maximum time allowed to elapse between an interrupt request and the start of the interrupt handler is indicated by allowable latency. If a program P takes 100 seconds to execute when interrupts are disabled, how long will P take to run when interrupts are enabled?

**Ans:**

Cho các thiết bị có thời gian ngắt và độ trễ, tần số ngắt như trên ... ! Chương trình P thực hiện trong 100s nếu không có ảnh hưởng của 3 thiết bị trên xen vào .

Nếu có 3 thiết bị ngắt ở trên xen vào và thực hiện cùng với chương trình P thì hỏi ? chương trình P phải mất bao lâu để thực thi xong ?

Các thiết bị trên sẽ chiếm thời gian lần lượt là :

The Device 1 shall take  $(150+50)/800 = 1/4$  unit

The Device 2 shall take  $(50+50)/1000 = 1/10$  unit

The Device 3 shall take  $(100+100)/800 = 1/4$  unit

Tổng cộng ta được 0.6 unit ( Có nghĩa trong một đơn vị thực thi thì 3 thiết bị trên sẽ chiếm 0.6 khoảng thời gian để thực thi còn 0.4 còn lại thì dành cho chương trình P :

=> dựa theo công thức tỉ lệ nghịch ta có thời gian thực thi của P bây giờ là :

1 unit : 100s ( Tỉ lệ nghịch )

0.4 unit : 250s

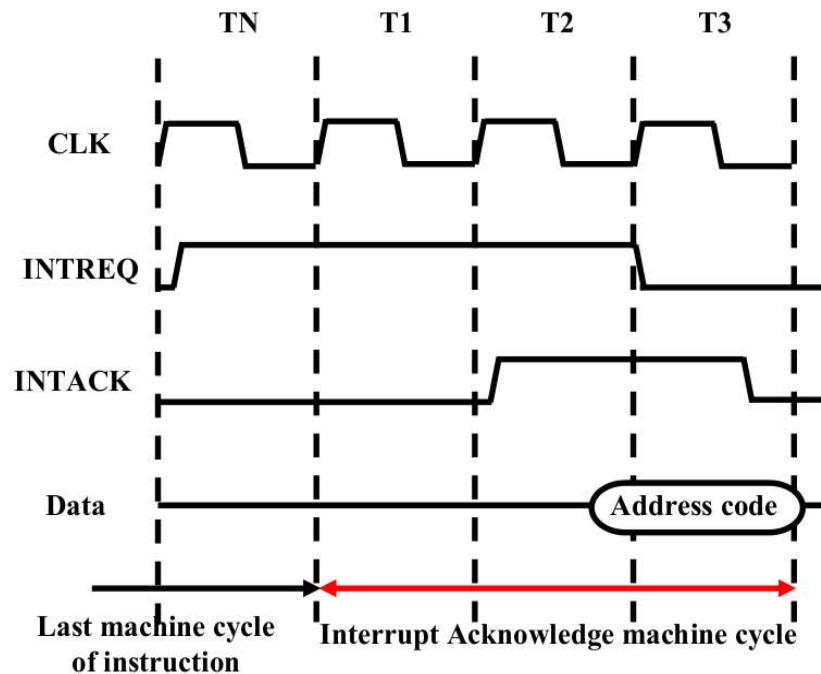
=> P mất 250s để thực thi nếu như có 3 thiết bị ngắt xen vào thực thi cùng với nó .

Câu 2 : TRAP trong interrupt là gì ... !



Nó được hiểu như là một ngắt được thực thi bằng phần mềm ... ! và nó ngắt theo mong muốn và ý định của người sử dụng để thực thi một công việc nào đó , TRAP và nó được coi như một chương trình con đặt biệt , ngắt TRAP thường là ngắt vô điều kiện khi thực hiện ngắt con trỏ sẽ nhảy đến vùng chứa chương trình và thực thi theo một hướng xác định do người dùng đề ra ... ! nó có thể lặp đi lặp lại trong chương trình như một thói quen để xử lý công việc nào đó .

Câu 3 : Ngắt thừa nhận là gì ? Interrupt Acknowledge Machine Cycle.



Khi một thiết bị ngoài kết nối vào thì tín hiệu INTREQ sẽ được đưa lên cao và bắt khởi động tín hiệu ngắt INTACK ( thừa nahaanj) và chuyển địa chỉ ngắt qua đường data line !



# Lesson

# 16

# DMA



## Định nghĩa về DMA là gì :

Chế độ DMA: thường thì hoạt động chuyển thông tin giữa thiết bị I/O với bộ nhớ máy tính hoặc từ bộ nhớ này tới bộ nhớ được thực hiện thông qua CPU theo cơ chế tuần tự: CPU ra lệnh "import" để nhập 1 đơn vị thông tin (byte/word) từ thiết bị I/O vào thanh ghi CPU rồi ra lệnh ghi thông tin từ thanh ghi CPU vào ô nhớ RAM nào đó. Qui trình này rất chậm vì CPU phải tốn nhiều chu kỳ máy để thực hiện các lệnh di chuyển thông tin, hơn nữa thông tin còn phải đi vòng qua CPU trước khi đến được vị trí cuối cùng. Để tăng tốc độ chuyển dữ liệu giữa I/O và RAM, người ta đã thiết kế mạch cứng DMA (Direct Memory Access) cho phép chuyển dữ liệu trực tiếp giữa I/O và RAM thông qua một đường bus mà không cần đi ngang qua CPU nữa. Như vậy, nếu driver thiết bị I/O hỗ trợ tốt chế độ hoạt động DMA thì bạn nên cho phép chế độ này hoạt động. Và trong lúc này CPU có thể đi thực hiện những công việc khác trên những đường bus khác.

## DMA controller là bộ phận gì ?

DMA controller thường là một thiết bị ngoại vi với một CPU được lập trình để thực hiện việc chuyển các dữ liệu thay cho CPU. Một bộ điều khiển DMA có thể truy cập trực tiếp bộ nhớ và được sử dụng nó để chuyển dữ liệu sang bộ nhớ khác hoặc các thiết bị I/O và ngược lại

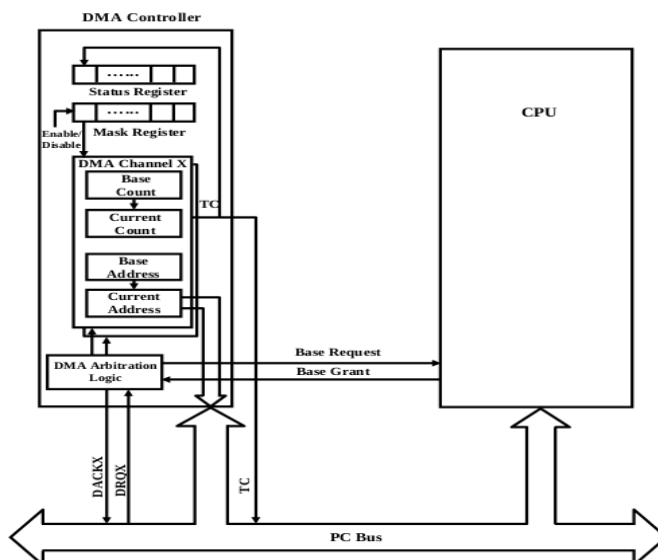


Fig. 16.1 The DMA controller architecture



## **Loại chuyển dữ liệu của DMA :**

### **1. Có hai loại :**

+ DMA transfers are flyby ( Chuyển dữ liệu ngang qua) hay còn được gọi là chuyển dữ liệu trên đường địa chỉ đơn , chuyển dữ liệu chu kì đơn . Trong một lần chuyển dữ liệu (tương ứng với 1 chu kì) thì một đường bus duy nhất sẽ được sử dụng để truyền dữ liệu từ nguồn đến đích

+ DMA transfers and fetch-and-deposit ( Chuyển dữ liệu theo kiểu lấy và gửi) hay còn được gọi là chuyển dữ liệu trên đường địa chỉ kép , chuyển dữ liệu chu kì kép. Trong chu kì đầu tiên Dữ liệu từ các thiết bị truyền nhận I/O hay các vùng nhớ sẽ được đưa vào một thanh ghi tạm nằm bên trong bộ DMA controller và dữ liệu đó sẽ được chuyển đến đích vào chu kì tiếp theo.

## **Chế độ chuyển dữ liệu của DMA :**

Chế độ truyền duy nhất

Chế độ truyền theo khối

chế độ truyền theo yêu cầu



**Câu 1 :** Con truyền dữ liệu 8237 ở chế độ High performance với nguồn cung cấp là 5v và hiệu điện thế có thể đạt tới 16MA tương ứng công suất tiêu thụ là 80 mW

**Câu 2 :** Các chế độ truyền của con 8237 bao gồm

+ Chế độ truyền đơn :

Thiết bị được lập trình để chỉ thực hiện 1 thao tác truyền.

Từ đếm giảm dần, địa chỉ giảm dần (hoặc tăng) sau mỗi thao tác truyền. Khi bộ đếm giảm từ 0 sang FFFFH, quá trình truyền kết thúc

+ Chế độ truyền theo khối :

Bộ điều khiển DMA được kích hoạt bởi DREQ và liên tục truyền trong quá trình phục vụ yêu cầu cho đến khi kết thúc do bộ đếm từ chuyển từ 0 về FFFFH hoặc do tín hiệu EOP từ bên ngoài.

+ Chế độ truyền theo yêu cầu :

Thực hiện việc truyền liên tục cho đến khi bộ đếm chuyển sang FFFFH hoặc nhận được EOP hoặc DREQ chuyển sang thụ động.

+ Chế độ xếp tầng :

Chủ yếu dùng để mở rộng hệ thống !

# Hệ thống nhúng : Lesson 16



## Câu 3 : vẽ cái con 8237

Về thầy bà luôn :

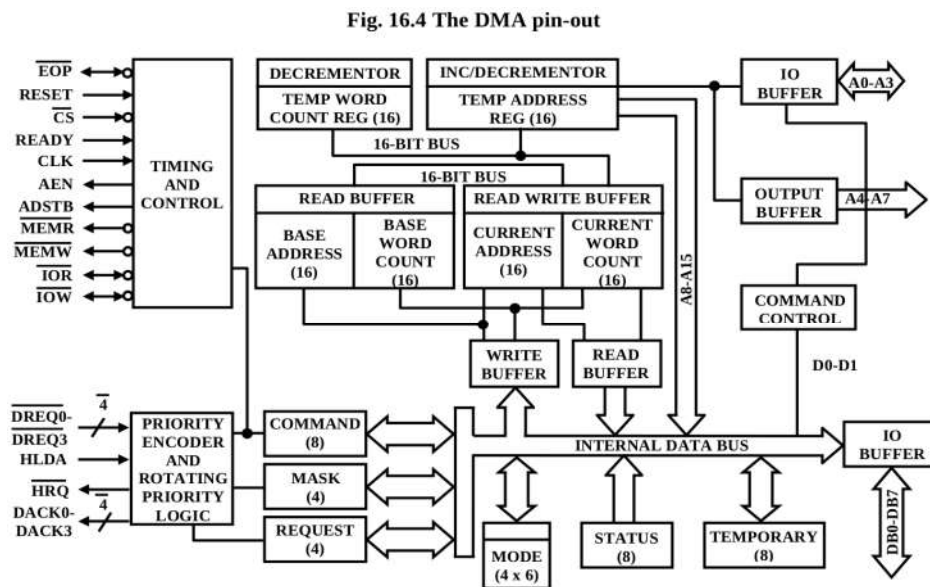


Fig. 16.5 The 8237 Architecture

## Các tín hiệu 8237

