HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

# LẬP TRÌNH HỆ THỐNG VÀ ĐIỀU KHIỂN THIẾT BỊ

*(Dùng cho sinh viên hệ đào tạo đại học từ xa)*

Lưu hành nội bộ

HÀ NỘI - 2006

# LẬP TRÌNH HỆ THỐNG
# VÀ ĐIỀU KHIỂN THIẾT BỊ

**Biên soạn :** THS. PHẠM VĂN CƯỜNG

# LỜI NÓI ĐẦU

Cuốn giáo trình Lập trình hợp ngữ và điều khiển thiết bị được chia thành 4 chương. Mỗi chương bao gồm các nội dung cơ bản, tóm tắt chương, các câu hỏi và bài tập cho mỗi chương.

Chương 1: trình bày về vấn đề liên quan đến bộ vi xử lý 8088 : kiến trúc, chức năng các thành phần và tập lệnh. Ngoài ra, 1 trong các ngắt được sử dụng phổ biến trong lập trình hệ thống- ngắt 21h của hệ điều hành DOS cũng được giới thiệu trong chương này.

Chương 2: trình bày về các vấn đề liên quan đến lập trình hợp ngữ: cách thức viết và thực hiện một chương trình, cách thức cài đặt các cấu trúc lập trình trong hợp ngữ và các vấn đề liên quan đến chương trình con và macro.

Chương 3: giới thiệu về công cụ gỡ rối debug, chương trình mô phỏng Emu 8086. Liên kết chương trình viết bằng hợp ngữ với chương trình được viết bằng các ngôn ngữ bậc cao như C và Pascal cũng được đề cập ở chương này. Ngoài ra, chương này có giới thiệu về một số ngắt của BIOS phục vụ thiết bị ngoại vi, chương trình thường trú và chương trình con ngắt.

Chương 4: Trình bày về lập trình phối ghép: lập trình modem, bàn phím và màn hình. Đồng thời chương này cũng giới thiệu về một môi trường RadASM để phát triển các ứng dụng viết bằng hợp ngữ trên Windows.

Do thời gian có hạn và kinh nghiệm còn hạn chế, cuốn giáo trình sẽ không tránh khỏi các sai sót. Tác giả biên soạn rất mong nhận được ý kiến đóng góp từ các độc giả.

Mọi ý kiến góp ý xin gửi về email : pcuongcntt@yahoo.com

*Xin chân thành cảm ơn!*

**Hà Nội, tháng 11/2006**

*Tác giả*

CHƯƠNG TRÌNH PTIT ĐÀO TẠO ĐẠI HỌC TỪ XA

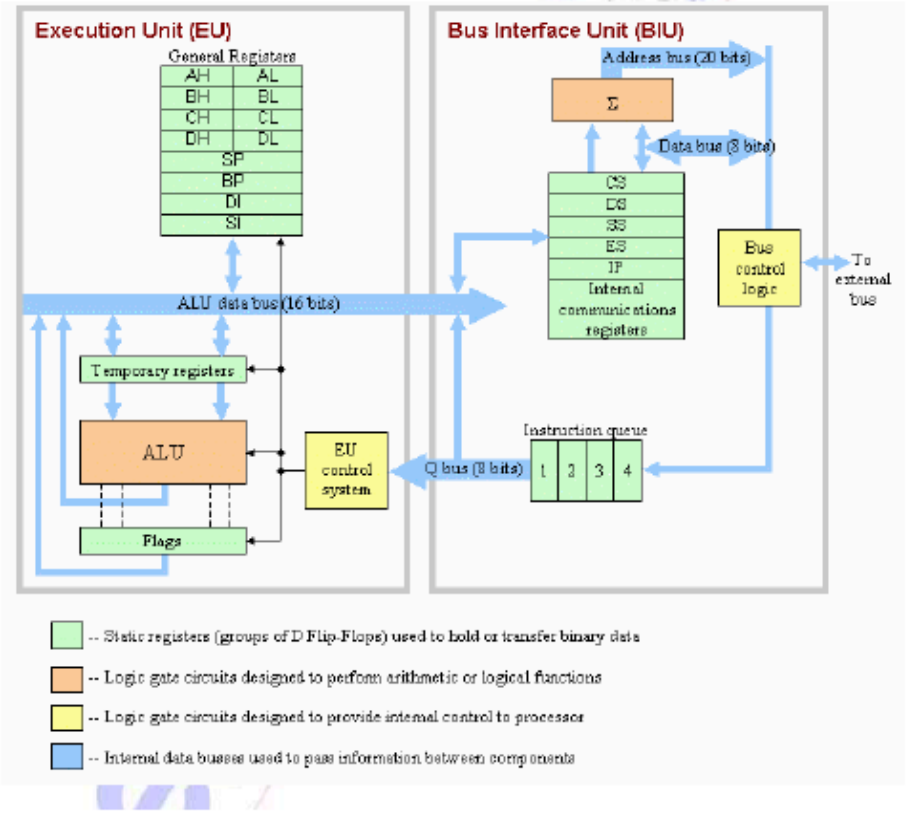
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
Km10 Đường Nguyễn Trãi, Hà Đông-Hà Tây
Tel: (04).5541221; Fax: (04).5540587
Website: http://www.vnpt.edu.vn; E-mail: dhtx@e-ptit.edu.vn
CHƯƠNG TRÌNH PTIT
ĐÀO TẠO ĐẠI HỌC TỪ XA

# CHƯƠNG 1: GIỚI THIỆU

## 1.1 CẤU TRÚC BỘ VI XỬ LÝ

Phần này trình bày kiến trúc bên trong của bộ Vi xử lý 8088 và bộ Vi xử lý Pentium IV

### 1.1.1 Sơ đồ kiến trúc bộ Vi xử lý 8088




**Hình 1.1: Kiến trúc bên trong của bộ Vi xử lý 8088**

Bộ vi xử lý 8088 được chia làm 2 khối chính: Khối giao diện bus (BIU) và khối thực hiện lệnh (EU).

Các thành phần bên trong của CPU giao tiếp với nhau thông qua các bus trong. Giữa khối giao diện bus và khối thực hiện lệnh được liên hệ với nhau thông qua hàng đợi dữ liệu và hệ thống bus trong.

### 1.1.2 Chức năng các thành phần

*1. Thành phần điều khiển Bus (Bus Control Logic)*

Điều khiển các loại tín hiệu trên các bus bao gồm: các tín hiệu trên bus địa chỉ (20 bit), các tín hiệu trên bus dữ liệu (8 bit) và các tín hiệu trên bus điều khiển. Ngoài ra, thành phần này còn làm nhiệm vụ hỗ trợ giao tiếp giữa hệ thống bus trong và bus ngoài. Hệ thống bus ngoài là hệ thống bus kết nối giữa các thành phần của hệ vi xử lý với nhau: CPU, Bộ nhớ trong và Thiết bị vào/ra.

*2. Hàng đợi lệnh (Prefetch Queue)*

Chứa mã lệnh chờ được xử lý. Hàng đợi lệnh có kích thước 4 byte đối với 8088 và 6 byte đối với 8086. Sở dĩ có điều này là vì hàng đợi lệnh phải có kích thước có thể chứa được ít nhất một lệnh có độ dài bất kỳ (dài nhất) của bộ vi xử lý. Mã tập lệnh của 8086 chứa các lệnh có độ dài từ 1-6 byte.

Hàng đợi lệnh làm việc theo cơ chế FIFO (First In First Out), nghĩa là lệnh nào được đưa vào hàng đợi lệnh trước sẽ được xử lý trước

*3. Khối điều khiển (Control Unit)*

Khối điều khiển có hai chức năng chính: giải mã lệnh và tạo xung điều khiển . Đầu vào của khối điều khiển là mã lệnh được đọc từ hàng đợi lệnh và đầu ra là các xung điều khiển gửi đến các bộ phận khác nhau bên trong bộ vi xử lý. Quá trình này được thực hiện nhờ hai mạch giải mã lệnh và mạch tạo xung.

*4. Khối số học và logic (Arithmetic Logic Unit)*

Khối số học và logic có chức năng thực hiện các phép tính toán như phép cộng, trừ... hay các phép logic như AND, OR, NOT. Đầu vào ALU là hai thanh ghi tạm thời chứa dữ liệu của cho phép tính được lấy từ bus dữ liệu. Kết quả đầu ra của ALU được đưa trở lại bus dữ liệu và phản ánh vào thanh ghi cờ (flag register).

*5. Các thanh ghi đoạn (Segment registers)*

Ta hãy thử xem đoạn chương trình được viết bằng ngôn ngữ C sau:

```
int Cong(int a, int b)
{
   Return (a+b);
}
void main()
{
     int x=3; int  y=4;
     printf("Tong: %d", Cong(x,y));
  }
```

Trong chương trình trên có 2 phần: phần khai báo và phần lệnh của chương trình. Trong phần lệnh có thể có lời gọi chương trình con.

Như vậy để thực hiện được một chương trình (dạng .EXE) thì người ta cần ít nhất 3 đoạn bộ nhớ (segment). Đoạn dành chứa dữ liệu được khai báo, đoạn chứa mã chương trình, đoạn ngăn

xếp phục vụ cho các lời gọi chương trình con. Mỗi đoạn có kích thước 64KB. Khi chương trình được thực hiện, mỗi đoạn bộ nhớ này được trỏ bởi các thanh ghi đoạn. Đó là:

- Thanh ghi đoạn mã CS (Code Segment): trỏ đến đoạn bộ nhớ chứa mã của chương trình.
- Thanh ghi đoạn dữ liệu DS (Data Segment): trỏ đến đoạn bộ nhớ chứa các khai báo của chương trình.
- Thanh ghi đoạn ngăn xếp SS (Stack Segment): trỏ đến đoạn bộ nhớ dành cho stack.
- Ngoài ra, trong nhiều trường hợp người ta sử dụng thêm một đoạn dữ liệu phụ dùng trong trường hợp các dữ liệu cần khai báo vượt quá kích thước cho phép của 1 đoạn (các khai báo mảng, file…). Khi đó thanh ghi đoạn dữ liệu phụ ES (Extra Segment) sẽ trỏ đến đoạn này

*6. Các thanh ghi con trỏ và chỉ số (pointers and index registers)*

Các thanh ghi con trỏ và chỉ số là các thanh ghi 16 bit. Chúng thường được lưu địa chỉ lệch (offset) và kết hợp với thanh ghi đoạn tương ứng tạo thành cặp thanh ghi chứa địa chỉ xác định của mã lệnh, mục dữ liệu, hoặc mục dữ liệu lưu trong stack. Nhờ vào cặp thanh ghi này, người ta có thể tính địa chỉ vật lý cụ thể theo công thức sau:

*Địa chỉ vật lý = địa chỉ đoạn * 16 + địa chỉ lệch*

Dưới đây là các thanh ghi con trỏ và chỉ số:

- Thanh ghi con trỏ lệnh IP (Instruction Pointer): trỏ vào lệnh kế tiếp sẽ được thực hiện nằm trong đoạn mã do con trỏ CS trỏ tới. Địa chỉ đầy đủ của lệnh là CS:IP.
- Thanh ghi con trỏ cơ sở BP (Base Pointer): trỏ vào một mục dữ liệu nằm trong đoạn ngăn xếp SS. Địa chỉ đầy đủ của mục dữ liệu là CS:IP.
- Thanh ghi con trỏ ngăn xếp SP (Stack Pointer): trỏ vào đỉnh hiện thời ngăn xếp nằm trong đoạn ngăn xếp SS. Địa chỉ đầy đủ của đỉnh ngăn xếp là SS:SP.
- Thanh ghi chỉ số nguồn SI (Source Index): trỏ vào một mục dữ liệu trong đoạn DS. Địa chỉ đầy đủ của mục dữ liệu là DS:SI.
- Thanh ghi chỉ số đích DI (Destination Index): trỏ vào một mục dữ liệu trong đoạn DS. Địa chỉ đầy đủ của mục dữ liệu là DS:DI.

*7. Các thanh ghi đa năng (Multi-purposed registers)*

Bộ xử lý 8088 có 4 thanh ghi đa năng 16 bit đó là: AX, BX, CX và DX. Các thanh ghi này cũng có thể được tách ra thành 2 nửa gồm 8 bit cao (nửa cao) gồm bit thứ 8 đến bit thứ 15 và 8 bit thấp (nửa thấp) gồm các bit thứ 0 đến 7. Các nửa thanh ghi này có thể được sử dụng một cách độc lập để chứa các dữ liệu 8 bit. Đó là các nửa thanh ghi: AH và AL, BH và BL, CH và CL, và DH và DL. Trong đó AH, BH, CH, DH la các nửa cao còn AL,BL, CL, DL là các nửa thấp.

Ngoài chức năng "đa năng", mỗi thanh ghi 16 bit thường được sử dụng trong các tác vụ đặc biệt, giống như tên của chúng:

- AX (Accumulator) thanh chứa: các kết quả của các phép toán thường được lưu vào thanh ghi này. Ngoài ra, AX còn là toán hạng ẩn cho 1 số phép toán như nhân (AX là thừa số) hoặc chia (AX là số bị chia).

- BX (Base) thanh ghi cơ sở: thường được dùng để chứa các địa chỉ cơ sở.
- CX (Count) bộ đếm: CX thường dung để chứa số lần lặp trong trường hợp dùng lệnh LOOP. Ngoài ra, CL còn chứa số lần dịch chuyển, quay trái, quay phải của các toán hạng.
- DX (Data) thanh ghi dữ liệu: DX thường được chứa địa chỉ offset của xâu kí tự khi có các thao tác nhập vào xâu hoặc in xâu. DX (cùng với AX) còn tham gia chứa kết quả của phép nhân các số 16 bit hoặc làm số bị chia cho phép chia các số 16 bit. Ngoài ra, DX còn dùng để chứa địa chỉ của các cổng vào/ra trong trường hợp thực hiện các lệnh IN hoặc OUT.

*8. Thanh ghi cờ (flag register)*

Thanh ghi cờ là thanh ghi lưu trữ trạng thái của CPU tại mỗi thời điểm. Thanh ghi cờ có 16 bit, trong đó có 7 bit dự trữ cho tương lai (CPU 8088 chưa dùng đến các bit này). Còn lại 9 bit và mỗi bit tương ứng là một cờ. Kết hợp các lệnh nhảy có điều kiện (conditional jump) với các cờ này, người lập trình dễ dàng hơn

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x | x | x | x | OF | DF | IF | TF | SF | ZF | x | AF | x | PF | x | CF |

Hình 1.2: Cấu trúc của thanh ghi cờ của CPU 8088.

Các bit được đánh dấu x là các cờ chưa được dùng đến.

- *Cờ CF* (Carry Flag): cờ nhớ CF=1 khi có nhớ hoặc trừ có mượn từ bit có trọng số cao nhất (Most Significant Bit). Ngoài ra, cờ CF=1 trong trường hợp khi thao tác với file hoặc thư mục gây ra lỗi như các lỗi tạo, xóa file và thư mục.
- *Cờ PF* (Parity Flag): cờ chẵn lẻ PF=1 khi tổng số các bit bằng 1 trong kết quả của phép tính là một số chẵn.
- *Cờ AF* (Auxiliary Carry Flag): cờ nhớ phụ AF =1 khi có nhớ từ bit thứ 4 sang bit thứ 5 hoặc có mượn từ bit 5 sang bit thứ 4 trong biểu diễn BCD của 1 số.
- *Cờ ZF* (Zero Flag): cờ Zero ZF=1 khi kết quả tính toán bằng 0.
- *Cờ SF* (Sign Flag): cờ dấu SF=1 khi kết quả tính toán là một số âm.
- *Cờ TF* (Trap Flag): cờ bẫy TF=1 khi CPU đang làm việc ở chế độ chạy từng lệnh. Chế độ này được sử dụng cần thiết khi tìm lỗi (defect) và gỡ lỗi (debug) chương trình.
- *Cờ IF* (Interrupt enable Flag): cờ cho phép ngắt IF=1, cho phép tác động đến yêu cầu ngắt che được (maskable interrupts).
- *Cờ DF* (Direction Flag): cờ hướng DF=1 khi CPU xử lý chuỗi kí tự theo thứ tự từ phải sang trái.
- *Cờ OF* (Overflow Flag): cờ tràn OF=1 khi kết quả là một số bù hai vượt ra ngoài giới hạn biểu diễn dành cho nó.

*9. Hệ thống bus trong (Internal bus system)*

Hệ thống bus bên trong của CPU 8088 bao gồm 3 loại:

- Bus *dữ* liệu: 16 bít, cho phép di chuyển 2 byte dữ liệu tại một thời điểm
- Bus địa chỉ: 20 bít, có thể địa chỉ hóa được $2^{20}$ bytes và vì thế không gian địa chỉ nhớ của CPU 8088 là 1MB.
- Bus điều khiển: truyền tải các tín hiệu điều khiển như RD, WR …

## 1.2 MỘT SỐ CHỨC NĂNG CỦA NGẮT 21H

Phần này trình bày các hàm thông dụng của ngắt 21h. Đó là các hàm thao tác vào/ra đối với kí tự, chuỗi ký tự, file, thư mục, kết thúc chương trình và trả lại quyền điều khiển cho Hệ điều hành DOS.

**Hàm 01:** đọc 1 kí tự (có hiện) từ bàn phím

Input: AH=01

Output: AL= mã ASCII của ký tự

AL=0 nếu gõ vào phím chức năng.

**Hàm 02:** hiện 1 kí tự lên màn hình

Input: AH=02

DL= mã ASCII của ký tự cần hiển thị

Output:

**Hàm 08:** đọc 1 kí tự (không hiện) từ bàn phím

Input: AH=08

Output: AL= mã ASCII của ký tự

AL=0 nếu gõ vào phím chức năng.

**Hàm 09:** hiện xâu kí tự kết thúc bởi '$' lên màn hình

Input: AH = 09

DX = địa chỉ offset của xâu kí tự

**Hàm 0Ah:** đọc xâu kí tự từ bàn phím

Input: AH = 09

DX = địa chỉ offset của vùng đệm chứa xâu kí tự

Output: DX = địa chỉ offset của xâu kí tự

**Hàm 39h:** tạo thư mục

Input: AH = 39h

DX = địa chỉ offset của tên thư mục

Output:

- Nếu thành công, thư mục được tạo ra
- Nếu không thành công, CF=1 và AX= mã lỗi.

**Hàm 3Ah:** xóa thư mục

Input: AH = 3Ah

DX = địa chỉ offset của tên thư mục

Output:

- Nếu thành công, thư mục được xóa
- Nếu không thành công, CF=1 và AX=mã lỗi.

**Hàm 3Ch:** tạo file

Input: AH = 3Ch

DX = địa chỉ offset của tên file

CX = thuộc tính file

Output:

- Nếu thành công, file được tạo ra, CF=0 và AX= thẻ file (file handle)
- Nếu không thành công, CF=1 và AX= mã lỗi.

Thuộc tính file được định nghĩa như sau:

00h: file bình thường (plain old file)

01h: file chỉ đọc (Read Only)

02h: file ẩn (Hidden from searches)

04h: file hệ thống (system)

08h: thuộc tính cho nhãn đĩa.

10h: thuộc tính cho thư mục con.

**Hàm 3Dh:** mở file

Input: AH = 3Dh

AL = mode

Output:

- Nếu thành công, file được tạo ra, CF=0 và AX= thẻ file (file handle)
- Nếu không thành công, CF=1 và AX= mã lỗi.

**Hàm 3Eh:** đóng file

Input: AH = 3Eh

BX = thẻ file

Output:

- Nếu thành công, file được đóng lại và CF=0
- Nếu không thành công, CF=1 và AX= mã lỗi.

**Hàm 3Fh:** đọc từ file

Input: AH = 3Fh

DS:DX = địa chỉ offset của vùng đệm

CX = số byte cần đọc

BX = thẻ file

Output:

- Nếu thành công, CF=0 và AX= số byte đã đọc được
- Nếu không thành công, CF=1 và AX= mã lỗi.