Chương 5: Vi xử lý 8086-3

Seattle Computer Products SCP-200B 8086 CPU Board

Mục tiêu

- Hiểu được tập lệnh assemply của vi xử lý 8086
- Biết được các bước lập trình với VXL 8086
- Biết được cấu trúc chung của một chương trình ASM trên 8086
- Biết và sử dụng được các cấu trúc điều khiển cơ bản

COMPUTER ENGINEERING

Nội dung

- Lập trình hợp ngữ cho bộ vi xử lý 8086
  - Tập lệnh ASM
  - Cấu trúc chung của chương trình
  - Các cấu trúc điều khiển cơ bản

COMPUTER ENGINEERING

Tập lệnh Assemply

- Các lệnh vận chuyển dữ liệu
- Các lệnh tính toán số học
- Các lệnh tính toán logic
- Các lệnh dịch quay toán hạng
- Các lệnh nhảy
- Các lệnh lặp
- Các lệnh điều khiển, đặc biệt khác

COMPUTER ENGINEERING

Các lệnh vận chuyển dữ liệu

1. MOV Đích, Nguồn

- Toán hạng Đích và Nguồn có thể định địa chỉ theo nhiều kiểu khác nhau nhưng phải có cùng độ dài và không được phép đồng thời là 2 ô nhớ hoặc hai thanh ghi đoạn
- VD:
  - MOV AL,AH
  - MOV CX, 50
  - MOV DL, [SI]

COMPUTER ENGINEERING

Các lệnh vận chuyển dữ liệu (tt)

2. OUT Port, Acc

- Trong đó Port là địa chỉ cổng, có 8bit. Nếu Acc là AL thì dữ liệu 8 bit được đưa ra ở Port. Nếu Acc là AX thì dữ liệu 16 bit được đưa ra ở Port và Port + 1.
- Có thể dùng DX để thay thế cho Port -> 0000H < Port <FFFFH
- VD:
  - OUT 45H, AL
  - MOV DX, 005FH
  - OUT DX, AX

COMPUTER ENGINEERING

Các lệnh vận chuyển dữ liệu (tt)

3. IN Acc, Port

- Trong đó Port là địa chỉ cổng, có 8bit. Truyền 1 byte hoặc 1 word từ cổng tới thanh ghi
- Có thể dùng DX để thay thế cho Port -> 0000H < Port <FFFFH
- VD:
  - OUT 45H, AL
  - MOV DX, 005FH
  - MOV DX, AX

COMPUTER ENGINEERING

Các lệnh vận chuyển dữ liệu (tt)

4. POP Đích

- Lấy 1 từ ở đỉnh ngăn xếp vào thanh ghi

5. PUSH Nguồn

- Cất 1 từ vào đỉnh ngăn xếp
- VD:
  - POP DX
  - PUSH BX

COMPUTER ENGINEERING

Các lệnh tính toán số học

6/7. ADC/ADD Đích, Nguồn

- Cộng 2 toán hạng Đích và Nguồn (ADC với cờ CF) kết quả lưu vào đích
- Các cờ bị thanh đổi: AF, CF, OF, PF, SF, ZF
- VD:
  - ADC AL, 74H
  - ADC CL,BL
  - ADC DL,[SI]

COMPUTER ENGINEERING

Các lệnh tính toán số học (tt)

8. SUB Đích, Nguồn

- Toán hạng Đích và Nguồn phải cùng loại dữ liệu và không được đồng thời là hai ô nhớ hoặc thanh ghi đoạn
- Các cờ bị thanh đổi: AF, CF, OF, PF, SF, ZF
- VD:
  - SUB AL, 74H
  - SUB CL,BL
  - SUB DL,[SI]

COMPUTER ENGINEERING

Các lệnh tính toán số học (tt)

9. MUL Nguồn

- Thực hiện phép nhân không dấu thanh ghi tích lũy với toán hạng nguồn
- Nếu nguồn là số 16bit thì tích lưu vào DXAX
- Các cờ bị thanh đổi: CF, OF
- VD:
  - MUL CX
  - MUL BL

COMPUTER ENGINEERING

Các lệnh tính toán số học (tt)

10. DIV Nguồn

- Toán hạng Nguồn là số chia.
- Nếu Nguồn là 8bit thì phép chia là AX/Nguồn, thương là AL, số dư là AH
- Nếu Nguồn là 16bit thì phép chia là DXAX/Nguồn, thương là AX và số dư là DX
- Nếu Nguồn = 0 hoặc thương lớn hơn FFH (FFFFH) thì 8086 thực hiện INT0
- VD:
  - MOV AX, 0033H
  - MOV BL, 25
  - DIV BL

COMPUTER ENGINEERING

Các lệnh tính toán số học (tt)

11. DEC Đích

- Trừ toán hạng Đích đi 1

12. INC Đích

- Tăng toán hạng Đích thêm 1
- Các cờ bị thanh đổi: AF, OF, PF, SF, ZF
- VD:
  - MOV BX, 1200H
  - DEC BX
  - INC BH

COMPUTER ENGINEERING

Các lệnh tính toán số học (tt)

13. NEG Đích

- Lấy 0 trừ đi toán hạng Đích, lưu kết quả vào đích
- Các cờ bị thanh đổi: AF, CF, OF, PF, SF, ZF
- VD:
  - MOV BX, 1234H
  - NEG BX

COMPUTER ENGINEERING

Các lệnh tính toán logic

14. AND Đích, Nguồn

- AND toán hạng Đích và Nguồn, lưu kết quả vào Đích

15. OR Đích, Nguồn

- OR toán hạng Đích và Nguồn, lưu kết quả vào Đích
- Các cờ bị thanh đổi: CF, OF, PF, SF, ZF
- VD:
  - AND AL, 0FH
  - OR CL, 30H

COMPUTER ENGINEERING

Các lệnh tính toán logic

16. NOT Đích

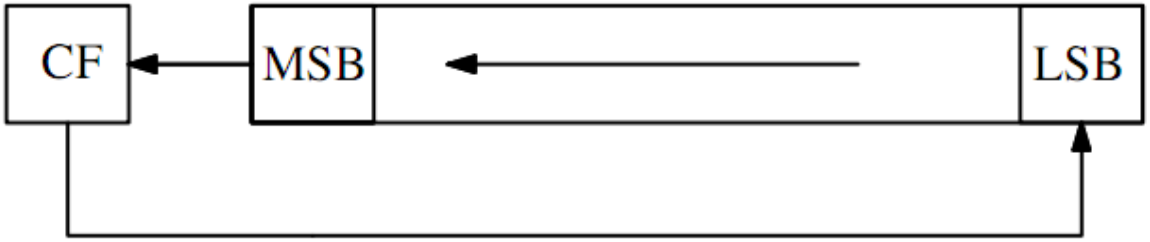
- Đảo giá trị các bit của toán hạng Đích
- Các cờ bị thanh đổi: không có cờ nào bị thay đổi
- VD:
  - MOV AL, 53H
  - NOT AL

COMPUTER ENGINEERING

Các lệnh dịch quay toán hạng

17. RCL Đích, CL

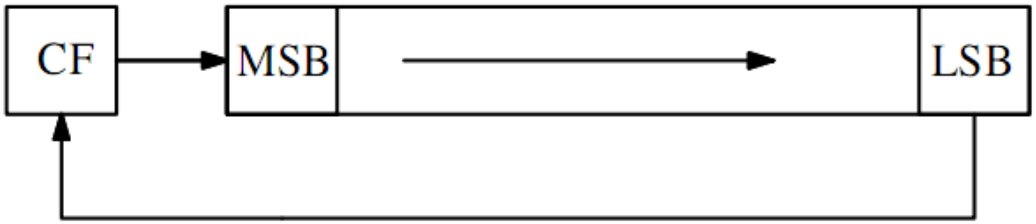
- Quay toán hạng Đích sang trái thông qua cờ CF; CL là số lần quay.
- Các cờ bị thanh đổi: CF, OF, SF, ZF, PF
- VD:
  - MOV CL, 5
  - RCL AL, CL
  - RCL BL, 1
  - ~~RCL BL, 2~~



Các lệnh dịch quay toán hạng (tt)

18. RCR Đích, CL

- Quay toán hạng Đích sang phải thông qua cờ CF; CL là số lần quay.
- Các cờ bị thanh đổi: CF, OF, SF, ZF, PF
- VD:
  - MOV CL, 4
  - RCR AL, CL
  - RCR BL, 1
  - ~~RCR BL, 2~~

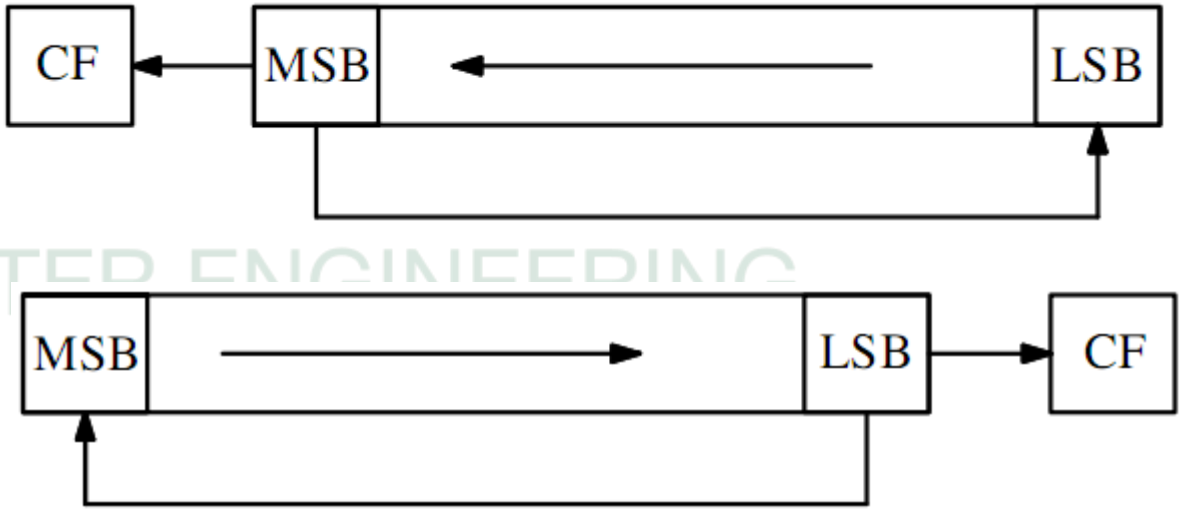


COMPUTER ENGINEERING

Các lệnh dịch quay toán hạng (tt)

19/20. ROL/ROR Đích, CL

- Quay toán hạng Đích sang trái/ phải; CL là số lần quay.
- CF = MSB/LSB sau mỗi lần quay
- Các cờ bị thanh đổi: CF, OF, SF, ZF, PF
- VD:
  - MOV CL, 4
  - ROL AL, CL
  - ROR BL, 1
  - ~~ROR BL, 2~~



COMPUTER ENGINEERING

Các lệnh dịch quay toán hạng (tt)

21/22. SHL/SHR Đích, CL

- Dịch trài/phải toán hạng Đích; CL là số lần dịch.
- CF = MSB/LSB sau mỗi lần dịch
- Bit 0 sẽ đưa vào LSB/MSB
- Các cờ bị thanh đổi: CF, OF, SF, ZF, PF
- VD:
  - MOV CL, 4
  - SHL AL, CL
  - SHR BL, 1
  - ~~SHR BL, 2~~

COMPUTER ENGINEERING

Các lệnh dịch quay toán hạng (tt)

23. XOR Đích, Nguồn

- Thực hiện lệnh XOR giữa 2 toán hạng
- Kết quả = 1 nếu 2 toán hạng đối nhau
- Kết quả = 0 nếu 2 toán hạng bằng nhau
- Các cờ bị thanh đổi: CF, OF, SF, ZF, PF, PF
- VD:
  - XOR AX, AX
  - XOR BX, BX
  - MOV AX, 5857H
  - XOR AX, BX

COMPUTER ENGINEERING

Các lệnh nhảy

24. CMP Đích, Nguồn

- Thực hiện trừ toán hạng Đích cho toán hạng Nguồn như không lưu kết quả
- Các cờ bị thanh đổi: AF, CF, OF, SF, ZF, PF

So sánh	CF	ZF
Đích = Nguồn	0	1
Đích > Nguồn	0	0
Đích < Nguồn	1	0

Các lệnh nhảy (tt)

25. JA/JNBE Nhãn

- Nhảy đến Nhãn nếu CF + ZF = 0 (lớn hơn)
- Nhãn nằm trong khoảng -128 đến +127
- VD:
  - MOV AX, 12ABH
  - CMP AX, 12ABH
  - JA HET GIO

COMPUTER ENGINEERING

Các lệnh nhảy (tt)

26. JAE/JNB/JNC Nhãn

- Nhảy đến Nhãn nếu CF = 0 (lớn hơn hoặc bằng)
- Nhãn nằm trong khoảng -128 đến +127
- VD:
  - MOV AX, 12AAH
  - CMP AX, 12ABH
  - JA E HET GIO

COMPUTER ENGINEERING

Các lệnh nhảy (tt)

27. JB/JC/JNAE Nhãn

- Nhảy đến Nhãn nếu CF = 1 (nhỏ hơn)
- Nhãn nằm trong khoảng -128 đến +127
- VD:
  - MOV AX, 12ACH
  - CMP AX, 12ABH
  - JB HET GIO

COMPUTER ENGINEERING

Các lệnh nhảy (tt)

28. JBE/JNA Nhãn

- Nhảy đến Nhãn nếu CF + ZF = 1 (nhỏ hơn)
- Nhãn nằm trong khoảng -128 đến +127
- VD:
  - MOV AX, 12ACH
  - CMP AX, 12ABH
  - JBE HET GIO

COMPUTER ENGINEERING

Các lệnh nhảy (tt)

29. JE/JZ Nhãn

- Nhảy đến Nhãn nếu ZF = 1 (bằng nhau)
- Nhãn nằm trong khoảng -128 đến +127
- VD:
  - MOV AX, 12ACH
  - CMP AX, 12ABH
  - JE HET GIO

COMPUTER ENGINEERING

Các lệnh nhảy (tt)

30. JNE/JNZ Nhãn

- Nhảy đến Nhãn nếu ZF = 0 (bằng nhau)
- Nhãn nằm trong khoảng -128 đến +127
- VD:
  - MOV AX, 12ABH
  - CMP AX, 12ABH
  - JNE HET GIO

COMPUTER ENGINEERING

Các lệnh nhảy (tt)

31. JMP Nhãn

- Nhảy đến Nhãn không điều kiện
- Nhảy ngắn : -128 đến + 127
- Nhảy gần: - 32768 đến + 32767
- Nhảy xa: đoạn mã khác
- VD:
  - JMP SHORT HET GIO
  - JMP NEAR HET GIO
  - JMP HET GIO

COMPUTER ENGINEERING

Các lệnh lặp

32. LOOP Nhãn

- Dùng để lặp lại đoạn chương trình từ Nhãn đến hết lệnh loop Nhãn khi CX #0
- Trước khi vào vòng lặp cần phải nạp CX, sau mỗi lần lặp CX giảm 1
- Nhãn phải nằm các LOOP tối đa -128 byte
- VD:
  - MOV AL, 20
  - MOV CX, 10
  - LAP: INC AL
  - LOOP LAP

Đoạn lệnh này lặp bao nhiêu lần?

COMPUTER ENGINEERING

Các lệnh lặp (tt)

33. LOOPE/LOOPZ Nhãn

- Dùng để lặp lại đoạn chương trình từ Nhãn đến hết lệnh loop Nhãn khi CX # 0 và ZF = 1
- Trước khi vào vòng lặp cần phải nạp CX, sau mỗi lần lặp CX giảm 1
- Nhãn phải nằm các LOOP tối đa -128 byte
- VD:
  - MOV AX, 2015H
  - MOV CX, 30
  - LAP: DEC AH
  - COMP AL, AH
  - LOOPE LAP

Đoạn lệnh này lặp bao nhiêu lần?

COMPUTER ENGINEERING

Các lệnh lặp (tt)

34. LOOPNE/LOOPNZ Nhãn

- Dùng để lặp lại đoạn chương trình từ Nhãn đến hết lệnh loop Nhãn khi CX =#0 và ZF = 0
- Trước khi vào vòng lặp cần phải nạp CX, sau mỗi lần lặp CX giảm 1
- Nhãn phải nằm các LOOP tối đa -128 byte
- VD:
  - MOV AX, 2015H
  - MOV CX, 30
  - LAP: DEC AH
  - CMP AL, AH
  - LOOPNE LAP

Đoạn lệnh này lặp bao nhiêu lần?

COMPUTER ENGINEERING

Các lệnh điều khiển, đặc biệt khác

35. CALL Nhãn chương trình con

- Dùng để chuyển hoạt động của vi xử lý từ chương trình chính sang chương trình con
- VD:
  - ORG 100h ; for COM file.
  - CALL p1
  - ADD AX, 1
  - RET ; return to OS.
  - p1 PROC ; procedure declaration. MOV AX, 1234h
  - RET ; return to caller.
  - p1 ENDP

COMPUTER ENGINEERING

Các lệnh điều khiển, đặc biệt khác (tt)

36. INT N

- Lệnh gọi chương trình con phục vụ ngắt
- Bảng vector ngắt 1KB từ 00000H đến 003FFH
- Có 256 ngắt (N từ 00H – FFH)
  - Mỗi vector có 4 byte chứa IP và CS của CT phục vụ ngắt
  - 32 vector đầu dành cho Intel
  - 224 vector tiếp theo dành cho người dùng

COMPUTER ENGINEERING

Các lệnh điều khiển, đặc biệt khác (tt)

Vector ngắt	Công dụng
00h	CPU: tác động khi chia cho 0
01h	CPU: chương trình thực thi từng bước
02h	CPU: ngắt không che được
03h	CPU: tạo điểm dừng chochương trình
04h	CPU: tác động khi kết quả số học tràn
05h	Tác động khi nhấn Print Screen
06h - 07h	Dành riêng
08h	Tác động bởi nhịp đồng hồ (18.2 lần/s)
09h	Tác động khi có phím nhấn
0Ah	Dành riêng
0Bh - 0Ch	Tác động phần cứng liên lạc nối tiếp
0Dh	Đĩa cứng
0Eh	Đĩa mềm
0Fh	Máy in
10h	BIOS: màn hình
11h	BIOS: xác định cấu hình máy tính

http://www.scribd.com/doc/82189065/Cac-ngat-cua-8086

Các lệnh điều khiển, đặc biệt khác (tt)

37. IRET

- Trở về chương trình chính từ chương trình con phục vụ ngắt

COMPUTER ENGINEERING

Quy trình thực hiện lệnh INT và IRET

THỰC HIỆN LỆNH INT

- Cất thanh ghi cờ vào Stack
- IF = 0 cấm các ngắt khác tác động
- TF = 0 chạy suốt chương trình ngắt
- Cất CS và IP vào Stack
- IP = [N*4], CS = [N*4 + 2]
- Thực hiện chương trình ngắt

COMPUTER ENGINEERING

Quy trình thực hiện lệnh INT và IRET

THỰC HIỆN LỆNH IRET

- Lấy IP từ Stack
- Lấy CS từ Stack
- Lấy thanh ghi cờ từ Stack
- Thực hiện chương trình chính trước đó

COMPUTER ENGINEERING

Các lệnh điều khiển, đặc biệt khác (tt)

38. RET / RET N

- Trở về chương trình chính từ chương trình con
- RET N dùng để nhảy qua các giá trị của chương trình con trong stack

COMPUTER ENGINEERING

Các lệnh điều khiển, đặc biệt khác (tt)

39. STC

- Dùng để thiết lập cờ nhớ CF = 1

40. NOP

- Không làm gì
- Tăng nội dung IP và tiêu tốn 3 chu kỳ clock

COMPUTER ENGINEERING

Các bước lập trình

- Cú pháp của chương trình hợp ngữ
- Dữ liệu cho chương trình
- Biến và hằng
- Khung của một chương trình hợp ngữ

COMPUTER ENGINEERING

Cú pháp của chương trình hợp ngữ

```
1.  .Model Small
2.  .Stack 100
3.  .Data
4.          Tbao DB 'Chuoi da sap xep:', 10, 13
5.          MGB DB 'a', 'Y', 'G', 'T', 'y', 'Z', 'U', 'B', 'D', 'E',
6.              DB '$'
7.  .Code
8.  MAIN Proc
9.          MOV AX, @Data                   ;khoi dau DS
10.         MOV DS, AX
11.         MOV BX, 10                      ;BX: so phan tu cua mang
12.         LEA     DX, MGB                 ;DX chi vao dau mang byte
13.         DEC     BX                      ;so vong so sanh phai lam
14. LAP:    MOV SI, DX                      ; SI chi vao dau mang
15.         MOV     CX, BX                  ; CX so lan so cua vong so
16.         MOV     DI, SI                  ;gia su ptu dau la max
17.         MOV     AL, [DI]                ;AL chua phan tu max
18. TIMMAX:
19.         INC SI                          ;chi vao phan tu ben canh
20.         CMP     [SI], AL                ; phan tu moi > max?
21.         JNG     TIEP                    ;khong, tim max
22.         MOV     DI, SI                  ; dung, DI chi vao max
23.       MOV AL, [DI]                      ;AL chua phan tu max
24. TIEP:   LOOP TIMMAX                     ;tim max cua mot vong so
25.         CALL  DOICHO                    ;doi cho max voi so moi
26.         DEC     BX                      ;so vong so con lai
27.         JNZ     LAP                     ;lam tiep vong so moi
28.         MOV   AH, 9                     ; hien thi chuoi da sap xep
29.         MOV     DX, Tbao
30.         INT     21H
31.         MOV     AH, 4CH                 ;ve DOS
32.         INT     21H
33. MAIN Endp
34. DOICHO Proc
35.         PUSH AX
36.         MOV     AL, [SI]
37.         XCHG    AL, [DI]
38.         MOV     [SI], AL
39.         POP     AX
40.         RET
41. DOICHO Endp
42.         END MAIN
```

khai báo kiểu kích thước bộ nhớ

khai báo đoạn ngăn xếp

khai báo đoạn dữ liệu

khai báo đoạn mã lệnh

bắt đầu chương trình chính

chú thích bắt đầu bằng dấu ;

kết thúc chương trình chính

bắt đầu chương trình con

kết thúc đoạn mã

Dữ liệu cho chương trình

- Hệ số 2: 00110B
- Hệ số 10: 1234
- Hệ số 16: 1EFDH, 0ABCEH
- Ký tự, chuỗi ký tự: 'A', 'mhd'

COMPUTER ENGINEERING

Biến và hằng

- DB (define byte): định nghĩa biến kiểu byte
- DW (define word): định nghĩa biến kiểu từ
- DD (define double word): định nghĩa biến kiểu từ kép

- Biến byte:
  - Tên DB gia_tri_khởi đầu
  - Ví dụ:

```
⇨B1   DB   4
⇨B1   DB   ?
⇨C1   DB   '$'
⇨C1   DB   34
```

```
MOV AL, B1

LEA BX, B1
MOV AL, [BX]
```

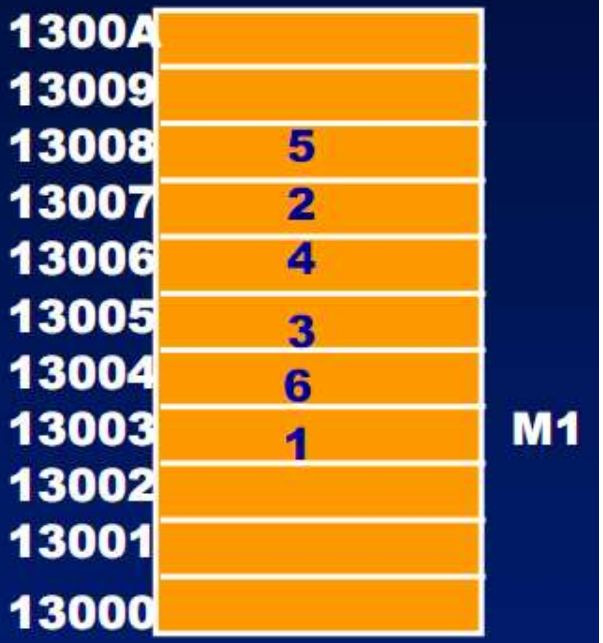
Biến và hằng (tt)

- **Biến từ:**
  - Tên DW gia_trị_khởi đầu
  - Ví dụ:
    - ⇨W1 DW 4
    - ⇨W2 DW ?
- **Biến mảng:**
  - M1 DB 4, 5, 6, 7, 8, 9
  - M2 DB 100 DUP(0)
  - M3 DB 100 DUP(?)
  - M4 DB 4, 3, 2, 2 DUP (1, 2 DUP(5), 6)
  - M4 DB 4, 3, 2, 1, 5, 5, 6, 1, 5, 5, 6

Biến và hằng (tt)

- **Biến mảng 2 chiều:**

$$\begin{pmatrix} 1 & 6 & 3 \\ 4 & 2 & 5 \end{pmatrix}$$



- M1 DB 1, 6, 3
  DB 4, 2, 5

- M2 DB 1, 4
  DB 6, 2
  DB 3, 5

```
MOV AL, M1 ; copy 1 vao AL
MOV AH, M1[2]
MOV BX, 1
MOV SI, 1
MOV CL, M1[BX+SI]
MOV AX, Word Ptr M1[BX+SI+2]
MOV DL, M1[BX][SI]
```

Biến và hằng (tt)

- **Biến kiểu xâu ký tự**
  - STR1 DB 'string'
  - STR2 DB 73h, 74h, 72h, 69h, 6Eh, 67h
  - STR3 DB 73h, 74h, 'r', 'i', 6Eh, 67h

- **Hằng có tên**
  - Có thể khai báo hằng ở trong chương trình
  - Thường được khai báo ở đoạn dữ liệu
  - Ví dụ:
    - ⇨ CR EQU 0Dh ;CR là carriage return
    - ⇨ LF EQU 0Ah ; LF là line feed
    - ⇨ CHAO EQU 'Hello'
    - ⇨ MSG DB CHAO, '$'

Khung của một chương trình hợp ngữ

- Khai báo quy mô sử dụng bộ nhớ

  .MODEL kiểu kích thước bộ nhớ

- Ví dụ: .MODEL Small

Kiểu	Mô tả
Tiny (hẹp)	mã lệnh và dữ liệu gói gọn trong một đoạn
Small (nhỏ)	mã lệnh nằm trong 1 đoạn, dữ liệu 1 đoạn
Medium (tB)	mã lệnh nằm trong nhiều đoạn, dữ liệu 1 đoạn
Compact (gọn)	mã lệnh nằm trong 1 đoạn, dữ liệu trong nhiễu đoạn
Large (lớn)	mã lệnh nằm trong nhiều đoạn, dữ liệu trong nhiều đoạn, không có mảng nào lớn hơn 64 K
Huge (đồ sộ)	mã lệnh nằm trong nhiều đoạn, dữ liệu trong nhiều đoạn, các mảng có thể lớn hơn 64 K

Khung của một chương trình hợp ngữ (tt)

- Khai báo đoạn ngăn xếp

  .Stack kích thước (bytes)

- Khai báo đoạn dữ liệu

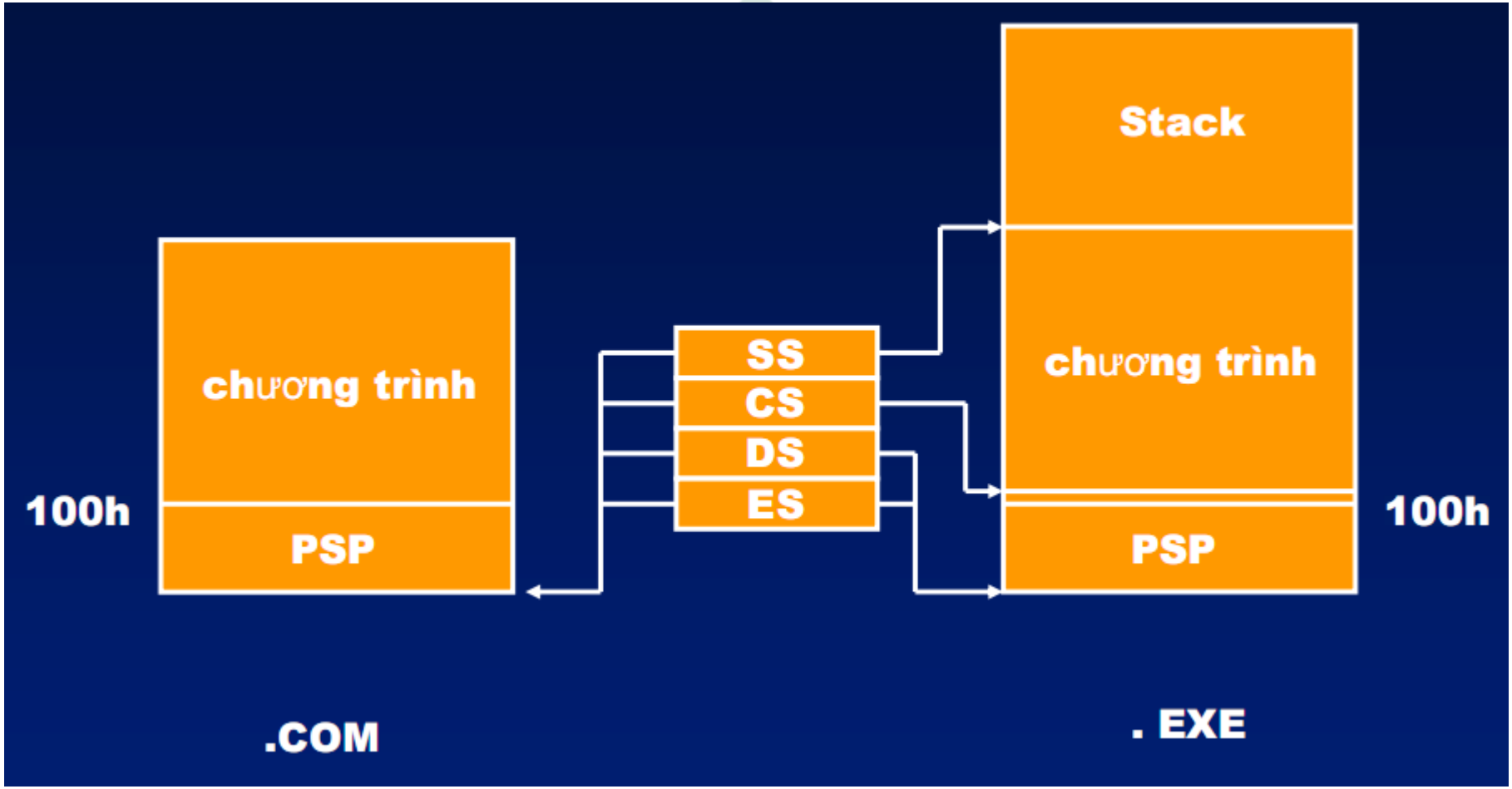
  .Data

- Khai báo đoạn mã

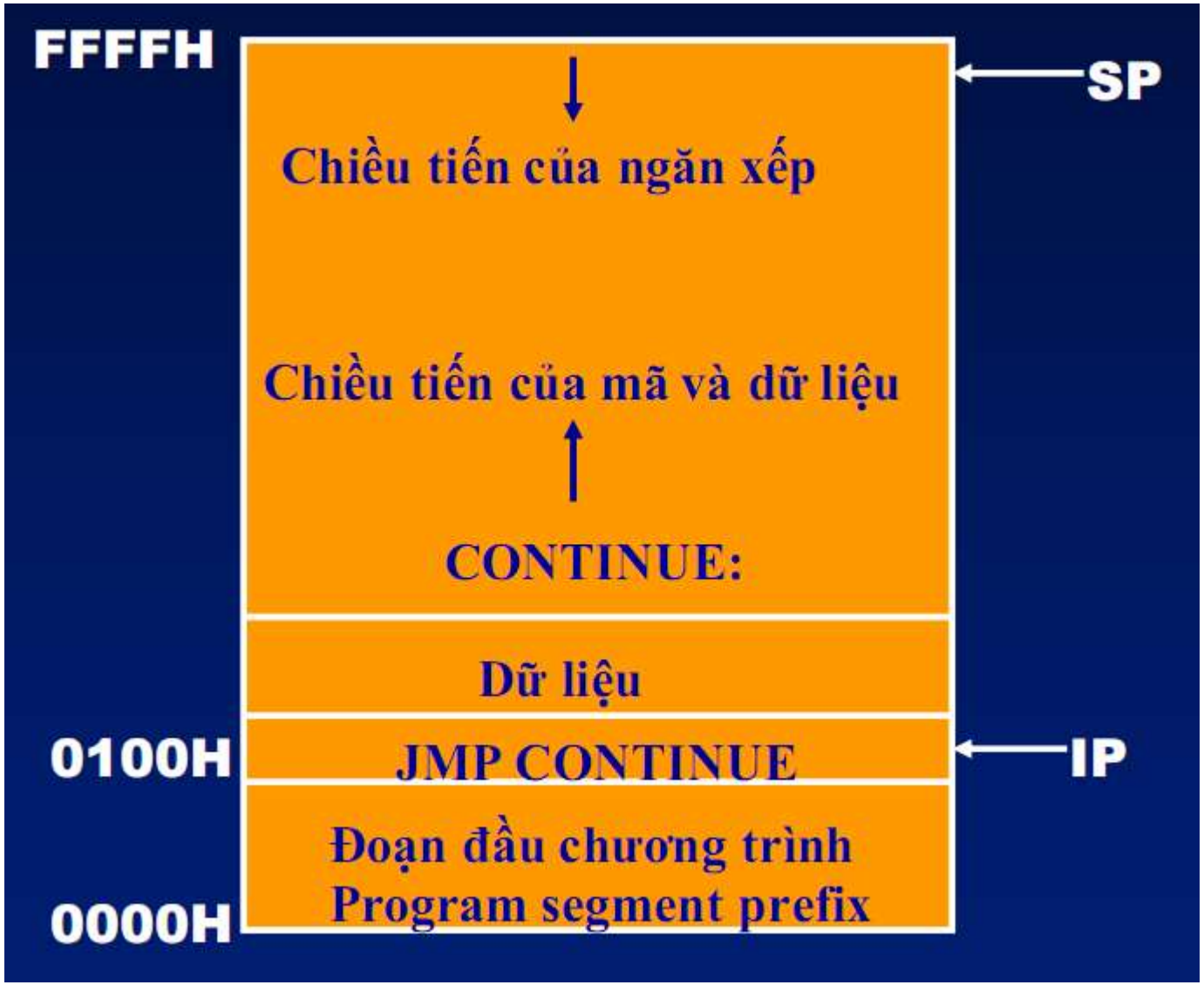
  . Code

COMPUTER ENGINEERING

Khung của một chương trình hợp ngữ (tt)



Khung của một chương trình hợp ngữ (tt)



Các cấu trúc điều khiển cơ bản

If điều kiện **then** công việc

- Ví dụ: Gán cho BX giá trị tuyệt đối của AX

```
; If AX<0
CMP        AX, 0     ; AX<0 ?
JNL        End_if    ; không, thoát ra
; then
NEG        AX        ; đúng, đảo dấu
End_if: MOV    BX, AX     ;gán
```

Các cấu trúc điều khiển cơ bản

If điều kiện **then** công việc1 **else** công việc 2

- Ví dụ: Nếu AX<BX thì CX = 0 ngược lại CX =1

```
    ; if AX<BX
    CMP       AX, BX          ; AX<BX ?
    JL        Then_   ; đúng, CX=0
    ;else
    MOV       CX, 1   ; sai, CX=1
    JMP       End_if
Then_: MOV    CX, 0;
End_if:
```

Các cấu trúc điều khiển cơ bản

Case Biểu thức

Giá trị 1: công việc 1

…

Giá trị n: công việc n

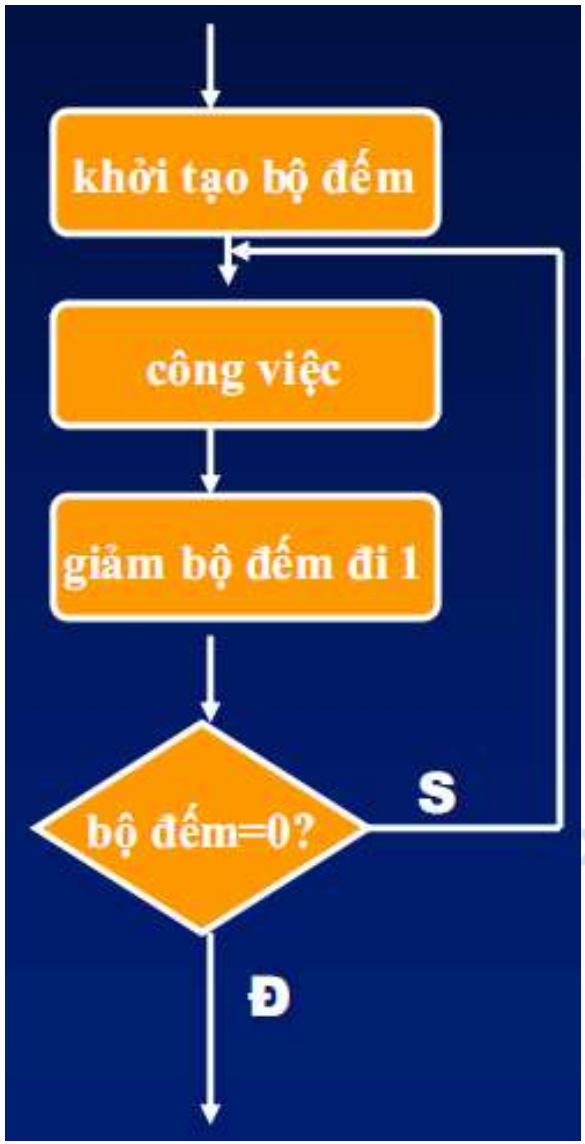
END CASE

- Ví dụ:
- Nếu AX<0 thì CX = -1
- Nếu AX=0 thì CX = 0
- Nếu AX>0 thì CX = 1

```
    CMP      AX, 0     ;
    JL       AM        ; AX<0
    JE       Khong     ; AX=0
    JG       DUONG;    AX>0
AM: MOV      CX, -1
    JMP      End_case
Khong: MOV   CX, 0
    JMP      End_case

DUONG: MOV   CX, 1
End_case:
```

Các cấu trúc điều khiển cơ bản



For số lần lặp **Do** công việc

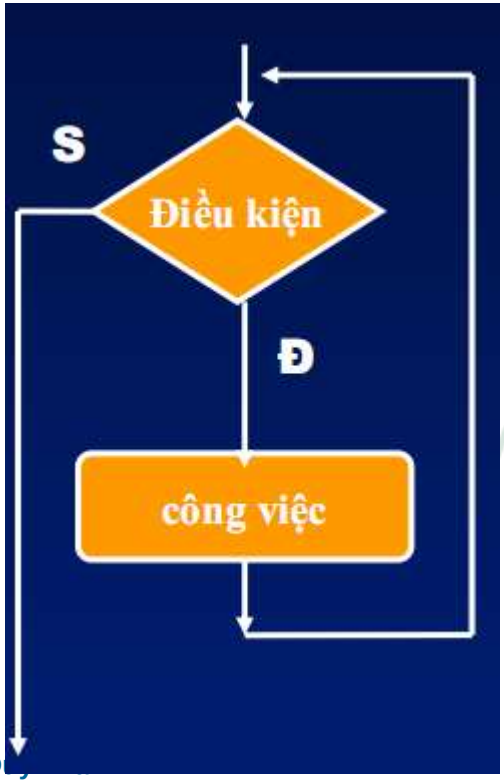
- Ví dụ: Hiển thị 1 dòng ký tự $ trên màn hình

```
    MOV CX, 80          ;số lần lặp
    MOV  AH,2           ;hàm hiển thị
    MOV   DL,'$'        ;DL chứa ký tự cần hiển thị
HIEN: INT 21H           ; Hiển thị
    LOOP   HIEN
End_for
```

Các cấu trúc điều khiển cơ bản

While điều kiện **Do** công việc

- Ví dụ: Đếm số ký tự đọc được từ bàn phím đến khi gặp ký tự CR thì thôi

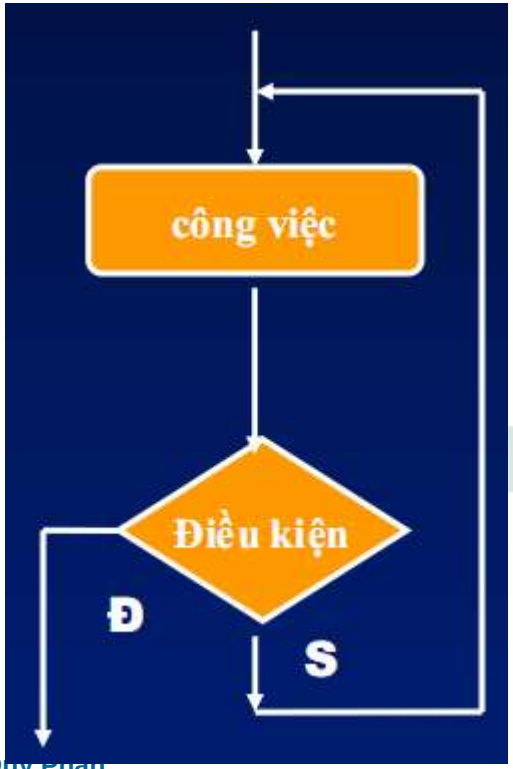



```
      XOR CX, CX          ;CX=0
      MOV  AH,1           ;hàm đọc ký tự từ bàn phím
TIEP:    INT 21H          ; đọc một ký tự vào AL
      CMP  AL, 13         ; đọc CR?
     JE End_while         ; đúng, thoát
     INC  CX              ; sai, thêm 1 ký tự vào tổng
     JMP TIEP             ; đọc tiếp
End_while:
```

Các cấu trúc điều khiển cơ bản

Repeat điều kiện **Until** công việc

- Ví dụ: Đọc từ bàn phím cho tới khi gặp ký tự CR thì thôi




```
      MOV  AH,1          ;hàm đọc ký tự từ bàn phím
TIEP:    INT 21H         ; đọc một ký tự vào AL
      CMP  AL, 13        ; đọc CR?
      JNE TIEP           ; chưa, đọc tiếp
End_:
```

Ôn tập

- Tập lệnh ASM
- Cấu trúc chung của chương trình
- Các cấu trúc điều khiển cơ bản

COMPUTER ENGINEERING

Kết thúc chương 5-3

Seattle Computer Products SCP-200B 8086 CPU Board