

Lập trình hợp ngữ 8086/88 và các chế độ địa chỉ

Giới thiệu chung về tập lệnh

- Mỗi bộ xử lý có một tập lệnh xác định
- Tập lệnh thường có hàng chục đến hàng trăm lệnh
- Mỗi lệnh là một chuỗi số nhị phân mà bộ xử lý hiểu được để thực hiện một thao tác xác định.
- Các lệnh được mô tả bằng các ký hiệu gọi nhớ → chính là các lệnh của hợp ngữ

1. Nhắc lại về tập thanh ghi

- Thanh ghi đa năng
 - Thanh ghi AX: Lưu kết quả tính toán
 - Thanh ghi BX: Thường dùng để lưu địa chỉ cơ sở
 - Thanh ghi CX: Lưu số lần lặp trong lệnh LOOP
 - Thanh ghi DX: Kết hợp với thanh ghi AX để lưu kết quả của phép nhân hoặc chia số 16 bit. Ngoài ra còn dùng để lưu địa chỉ 16 bit của cổng vào ra

1. Nhắc lại về thanh ghi (tiếp)

- Thanh ghi đoạn
 - Thanh ghi CS: Để ghi địa chỉ mã lệnh
 - Thanh ghi DS: Chứa địa chỉ dữ liệu
 - Thanh ghi SS: Chứa địa chỉ ngăn xếp
 - Thanh ghi ES: Thanh ghi phụ, thường dùng để chứa địa chỉ chuỗi

1. Nhắc lại về thanh ghi (tiếp)

- Thanh ghi con trỏ và chỉ số
 - IP (PC): Thanh ghi trỏ tới lệnh tiếp theo
 - BP: Thanh ghi trỏ về dữ liệu bộ nhớ
 - SP: Trỏ tới đỉnh ngăn xếp
 - SI: Thanh ghi chỉ số nguồn, trỏ vào dữ liệu, kết hợp với thanh ghi DS để chỉ địa chỉ của dữ liệu là DS:SI
 - DI: Thanh ghi chỉ số đích, trỏ vào dữ liệu, kết hợp với ES để có ES:DI

Cấu trúc của chương trình hợp ngữ

```
.MODEL    Kiểu kích thước bộ nhớ ; Khai báo quy mô sử dụng bộ nhớ
.STACK   Kích thước                ; Khai báo dung lượng đoạn stack
.DATA    msg DB 'Hello$'           ; Khai báo đoạn dữ liệu

.CODE
main PROC                                ; Khai báo đoạn mã
...
main ENDP
END main
```

Ví dụ một chương trình hợp ngữ

;Viết ra màn hình Hello World

```
.MODEL small
.STACK 100h
.DATA
Message db 'Hello World!$'
.CODE
ProgramStart:
    Mov AX,@DATA ; nạp địa chỉ đoạn dữ liệu vào AX
    Mov DS,AX ; DS chứa địa chỉ đoạn dữ liệu
    Mov DX, OFFSET Message ; hoặc LEA DX, Message
    Mov AH,9h ; hàm 9h hiển thị chuỗi kí tự
    Int 21h ; gọi ngắt 21h của DOS
    Mov AH,4Ch ; hàm 4Ch trở về DOS
    Int 21h
End ProgramStart
```

7

Ngắt 21h

- Ngắt 21h được dùng để gọi rất nhiều hàm của DOS, mỗi hàm được gọi bằng cách đặt số hàm vào trong thanh ghi AH và gọi INT 21h.
- Một số hàm hay dùng:

Số hiệu hàm	Chương trình
1	Vào một phím
2	Đưa một kí tự ra màn hình
9	Đưa ra một chuỗi kí tự

Hàm 1: Vào một phím

Vào: AH=1
Ra: AL=Mã ASCII nếu một phím kí tự được nhấn
=0 nếu một phím điều khiển hay chức năng được nhấn

Ví dụ:

```
MOV AH,1
INT 21h ; mã ASCII trong AL
```

Hàm 2: Hiển thị một kí tự hay thi hành một chức năng điều khiển

Vào: AH=2
DL=Chứa mã ASCII của kí tự hiển thị hay kí tự điều khiển
Ra: AL=Mã ASCII của kí tự hiển thị hay kí tự điều khiển

8

Ngắt 21h (tiếp)

Hàm 9: Hiển thị một chuỗi

Vào: AH=9
DX=địa chỉ tương đối (offset) của chuỗi. Chuỗi phải kết thúc bằng kí tự \$
Ra: AL=Mã ASCII nếu một phím kí tự được nhấn
=0 nếu một phím điều khiển hay chức năng được nhấn

Ví dụ:

```
.MODEL small
.DATA
Mess DB 'Hello$'
.CODE
MOV AX,@DATA
MOV DS,AX
LEA DX,Mess ; nạp offset của Mess vào DX
MOV AH,9
INT 21h
```

9

2. Cú pháp của hợp ngữ

Mỗi lệnh hay dẫn hướng biên dịch thường có 4 trường:

[Tên] Toán tử Toán hạng [Lời bình]

Ví dụ một lệnh:

START: MOV CX,5 ; khởi tạo bộ đếm

Ví dụ một dẫn hướng biên dịch:

Main PROC

Main là tên và toán hạng là PROC, khai báo một chương trình con có tên là Main

10

Khai báo dữ liệu

• Khai báo biến:

```
A1 DB 1 ; A1 dài 1 byte, A1=1
A2 DB ? ; A2 dài 1 byte và không có giá trị
A3 DW 1 ; A3 dài 2 byte và A3=1
A4 DD ? ; A4 dài 4 byte, không có giá trị ban đầu
A5 DB 1,2,3 ; Mảng 3 phần tử
A6 DB 10 DUP(?) ; Mảng 10 phần tử, không gán giá trị ban đầu
A7 DB 'A' ; Biến kiểu ký tự
```

• Khai báo hằng số:

```
Muoi EQU 10
BaA EQU 'AAA'
```

11

Lệnh trong hợp ngữ

Dạng lệnh

<Mã gọi nhớ> <Toán hạng đích> [<Toán hạng nguồn>]

- Mã gọi nhớ:** Giúp người sử dụng biết hoạt động của lệnh
- Toán hạng đích:**
 - Giữ kết quả (nếu có yêu cầu) sau khi thi hành lệnh.
 - Có thể là thanh ghi hay bộ nhớ.
- Toán hạng nguồn:** có thể là thanh ghi, bộ nhớ hay một giá trị tức thời

Ví dụ:

```
MOV AX,WORD1
```

Lệnh trên chuyển nội dung của ô nhớ WORD1 vào thanh ghi AX (tương đương lệnh gán)

12

Các nhóm lệnh

- Chuyển dữ liệu
- Xử lý số học với số nguyên
- Xử lý logic
- Điều khiển vào-ra
- Chuyển điều khiển (rẽ nhánh)
- Điều khiển hệ thống
- Xử lý số dấu phẩy động
- Xử lý các dữ liệu chuyên dụng

13

2.1 Nhóm lệnh chuyển dữ liệu

- Nhóm lệnh chuyển dữ liệu đa dụng
- Nhóm lệnh chuyển địa chỉ
- Nhóm lệnh chuyển cờ

14

Nhóm lệnh chuyển dữ liệu đa dụng

- **MOVE** Copy dữ liệu từ nguồn đến đích
- **LOAD** Nạp dữ liệu từ bộ nhớ đến bộ xử lý
- **STORE** Cát dữ liệu từ bộ xử lý đến bộ nhớ
- **EXCHANGE** Trao đổi nội dung của nguồn và đích
- **CLEAR** Chuyển các bit 0 vào toán hạng đích
- **SET** Chuyển các bit 1 vào toán hạng đích
- **PUSH** Cát nội dung toán hạng nguồn vào ngăn xếp
- **POP** Lấy nội dung đỉnh ngăn xếp đưa đến toán hạng đích

15

Lệnh MOV

MOV đích, nguồn

Đích	Nguồn	Ví dụ	Giải thích
1 Bộ nhớ	Thanh ghi	MOV DS:100H, AX	- chuyển nội dung trong AX vào vị trí nhớ 100H.
2 Thanh ghi	Bộ nhớ	MOV AX, MEM1	-Chuyển nội dung trong vị trí nhớ do nhãn MEM1 chỉ ra vào thanh ghi AX.
3 Thanh ghi	Thanh ghi	MOV AX, BX	— Chuyển nội dung trong BX vào thanh ghi AX.
4 Thanh ghi	Tức thời	MOV AX, 0FFFFH	- Chuyển giá trị hằng số FFFFH vào thanh ghi AX

16

PUSH/POP

- **Lệnh PUSH:** Cát dữ liệu vào ngăn xếp
 - Viết lệnh: **PUSH nguồn**
 - Mô tả: **SP ← SP - 2**
Nguồn → {SP}.
 - Ví dụ:
 - PUSH BX; Cát BX vào ngăn xếp tại vị trí do SP chỉ ra
 - PUSH Table[BX] ;Cát 2 byte của vùng dữ liệu DS có địa chỉ đầu tại Table+BX

17

Ví dụ lệnh PUSH/POP

- **Lệnh POP:** Lấy dữ liệu từ ngăn xếp
 - Viết lệnh: **POP đích**
 - Mô tả: **Đích ← {SP}.**
SP ← SP + 2
 - Ví dụ:
 - POP BX; lấy 2 byte từ đỉnh ngăn xếp đưa vào BX
 - POP Table[BX] ;lấy 2 byte từ đỉnh ngăn xếp ;đưa vào DS có địa chỉ đầu ;tại Table+BX

18

Nhóm lệnh chuyển địa chỉ

- Lệnh

LEA reg, mem

chuyển địa chỉ offset của toán hạng bộ nhớ vào thanh ghi reg16

19

Nhóm lệnh chuyển cờ

- Lệnh LAHF (Load AH from flag)
- Lệnh SAHF (Store AH into flag)

20

Nhóm lệnh chuyển dữ liệu qua cổng

- INPUT** Copy dữ liệu từ một cổng xác định đến đích
- OUTPUT** Copy dữ liệu từ nguồn đến một cổng xác định

21

Lệnh IN/OUT

- Lệnh IN:** Đọc dữ liệu từ cổng vào thanh ghi Acc
 - Viết lệnh: **IN Acc, Port**
 - Port: từ 00h .. FFh
 - Nếu Acc là AL: dữ liệu 8 bit được đưa vào từ cổng port
 - Nếu Acc là AX: Dữ liệu 16 bit được đưa vào từ cổng port, port+1
 - Cách khác: **IN Acc, DX**
 - DX: Được gán giá trị cổng (0000h .. FFFFh) từ trước

22

Lệnh IN/OUT

- Lệnh OUT:** Đưa dữ liệu từ thanh ghi Acc ra cổng
 - Viết lệnh: **OUT Port, Acc**
 - Port: từ 00h .. FFh
 - Nếu Acc là AL: dữ liệu 8 bit được đưa ra cổng
 - Nếu Acc là AX: Dữ liệu 16 bit được đưa ra cổng port, port+1
 - Cách khác: **OUT DX, Acc**
 - DX: Được gán giá trị cổng (0000h .. FFFFh) từ trước

23

Nhóm lệnh số học

- ADD** Cộng hai toán hạng
- SUBTRACT** Trừ hai toán hạng
- MULTIPLY** Nhân hai toán hạng
- DIVIDE** Chia hai toán hạng
- ABSOLUTE** Lấy trị tuyệt đối toán hạng
- NEGATE** Đổi dấu toán hạng (lấy bù 2)
- INCREMENT** Tăng toán hạng thêm 1
- DECREMENT** Giảm toán hạng đi 1
- COMPARE** Trừ hai toán hạng để lập cờ

24

Lệnh ADD/SUB

- Viết lệnh:
 - ADD** **đích, nguồn**; đích←đích + nguồn
 - SUB** **đích, nguồn**; đích←đích - nguồn
- Đích và nguồn phải có cùng kích thước
- Bảng loại địa chỉ khác nhau của đích và nguồn

Đích (nơi đến)	Nguồn (gốc)
Thanh ghi	Thanh ghi
Thanh ghi	Bộ nhớ
Bộ nhớ	Thanh ghi
Bộ nhớ	Tức thời(hằng số)
Thanh ghi	Tức thời (hằng số)

25

Ví dụ

- Viết đoạn chương trình cộng 5h với 3h dùng hai thanh ghi AL và BL

```
MOV AL, 05H ; AL ← 05H
MOV BL, 03H ; BL ← 03H
ADD AL, BL ; AL ← 05H+03H =08H
MOV DS:100H, AL ; Di chuyển kết quả từ AL vào vị trí nhớ DS:100H.
```

26

Lệnh MUL

- Viết lệnh:
 - MUL** **số nhân nguồn**
- Mô tả
 - Nhân 2 số không dấu
 - Toán hạng nguồn 8 bit:** AX ← AL*số nhân nguồn
 - Toán hạng nguồn 16 bit:** DXAX ← AX*số nhân nguồn

27

MUL: Ví dụ

- Ví dụ1: Viết đoạn chương trình nhân 5h với 3h dùng thanh ghi CL

```
MOV AL, 05H ; AL ← 5H (số bị nhân)
MOV CL, 03H ; CL ← 3H (số nhân)
MUL CL ; AX ← FH (kết quả)
MOV MEM1, AX ; chuyển kết quả(0FH) ; từ AX vào vị trí nhớ có nhãn MEM1
```

28

MUL: Ví dụ

- Ví dụ 2: Viết đoạn chương trình nhân 3A62H với 2B14H.

```
MOV AX, 3A62H ; AX ← 3A62H
MOV CX, 2B14H ; CX ← 2B14H
MUL CX ; DXAX ← tích = 289C63A8H
```

29

Lệnh DIV

- Viết lệnh:
 - DIV** **số chia nguồn**
- Mô tả
 - Chia 2 số không dấu
 - Toán hạng nguồn 8 bit:**
 - AL ← AX/số chia nguồn,
 - AH ← số dư AX/số chia nguồn
 - Toán hạng nguồn 16 bit:**
 - AX ← DXAX/số chia nguồn
 - DX ← số dư DXAX/số chia nguồn

30

DIV: Ví dụ

- Ví dụ 1: Viết đoạn chương trình chia 6h cho 3h, dùng thanh ghi CL

```
MOV AX, 0006H ; AX ← 6H
MOV CL, 03H   ; CL ← 3H
DIV CL        ; AH:AL ← 0H (số dư), 2H (thương số)
```

- Chú ý: 6H được đưa vào thành 0006H để lấp đầy toàn bộ thanh ghi AX. Như vậy các byte trọng số cao của AX sẽ bị xoá để tránh bị lỗi.

31

DIV: Ví dụ

- Ví dụ 2: Viết đoạn chương trình chia 1A034H cho 1002H, dùng thanh ghi BX

```
MOV AX, A034H ; AX ← A034H
MOV DX, 0001H ; DX ← 0001H
MOV BX, 1002H ; BX ← 1002H
DIV BX        ; DX:AX ← 00H (số dư) 1AH
              ; (thương số)
```

32

Nhóm lệnh logic

AND	Thực hiện phép AND hai toán hạng
OR	Thực hiện phép OR hai toán hạng
XOR	Thực hiện phép XOR hai toán hạng
NOT	Đảo bit của toán hạng (lấy bù 1)
TEST	Thực hiện phép AND hai toán hạng để lập cờ
SHIFT	Dịch trái (phải) toán hạng
ROTATE	Quay trái (phải) toán hạng

33

Lệnh NOT

- Viết lệnh:
NOT Đích ; Đích ← $\overline{\text{Đích}}$
- Ví dụ: Xác định kết quả của đoạn chương trình sau:

```
MOV BL, 00110011B
NOT BL
MOV MEM1, BL
```

34

Lệnh AND/OR/XOR

- Viết lệnh:
AND đích, nguồn
OR đích, nguồn
XOR đích, nguồn

- Ví dụ:
AND BL, 0FH ; che 4 bit cao của BL.
OR BL, 30H ; Thiết lập bit b4 và b5 bằng 1.
XOR BL, BL ; Xoá các bit của BL

35

Nhóm lệnh xử lý chuỗi

- Lệnh LODS
LODSB ; AL ← DS:SI, SI ← SI+1
LODSW ; AX ← [DS:SI+1:DS:SI], SI ← SI+2
- Lệnh STOS
STOSB ; ES:DI ← AL, DI ← DI+1
STOSW ; [ES:DI+1, ES:DI] ← AX, DI ← DI+2
- MOVS: Chuyển nội dung tại địa chỉ trỏ bởi DS:SI đến địa chỉ trỏ bởi ES:DI. Giá trị của SI và DI thay đổi giống trường hợp trên
- CMPS: So sánh byte hay từ tại vị trí DS:SI và ES:DI. Giá trị của SI và DI thay đổi giống trường hợp trên

36

Nhóm lệnh chuyển điều khiển

- Lệnh so sánh
- Lệnh nhảy không điều kiện
- Lệnh nhảy có điều kiện
- Vòng lặp

37

Lệnh so sánh

- Lệnh

CMP Toán hạng đích, Toán hạng nguồn

- Trong đó:** [Toán hạng đích], [Toán hạng nguồn] có thể là hằng, biến, thanh ghi hay ô nhớ. [Toán hạng đích] không thể là hằng số. [Toán hạng đích] và [Toán hạng nguồn] không thể đồng thời là ô nhớ.
- Tác dụng:** Lệnh **CMP** (**Compare**) được sử dụng để so sánh giá trị/nội dung của [Toán hạng đích] so với [Toán hạng nguồn]. Tương tự như lệnh **Sub**, nó lấy [Toán hạng đích] trừ đi [Toán hạng nguồn] nhưng kết quả không làm thay đổi [Toán hạng đích] mà chỉ làm thay đổi giá trị của một số cờ hiệu: CF, ZF, OF,...

Ví dụ:

```
MOV AX,A
CMP AX,B
JA label1
JB label2
JMP label3
```

38

Lệnh nhảy không điều kiện

- Lệnh

JMP <ví trí đích>

- Trong đó:** <Ví trí đích> có thể là nhân của một lệnh, tên của một thủ tục hoặc có thể là một thanh ghi, một ô nhớ (đã được định nghĩa) nào đó. <Ví trí đích> cũng có thể là một biến nào đó, giá trị của nó thường là địa chỉ của một ô nhớ trong đoạn Code.
- Tác dụng:** Khi gặp lệnh này chương trình chuyển điều khiển (nhảy đến) đến thực hiện lệnh sau <Ví trí đích> mà không phụ thuộc vào bất kỳ điều kiện nào.

Ví dụ:

Top:

```
Start:
Jmp Main
TieuDe DB 'Khoa CNTT – DHGTVT'
Main PROC
```

.....

```
Main ENDP
END Start
```

39

Lệnh nhảy có điều kiện

- JA label:** Nhảy nếu lớn hơn
- JB label:** Nhảy nếu nhỏ hơn
- JE label:** Nhảy nếu bằng

Ví dụ 1:

```
CMP al,13          ;xem al=13?, ki tu xuong dong
JE  dungnhap
```

Ví dụ 2:

```
CMP ax,bx
JA nhan           ; nhảy nếu ax>bx
```

40

Các cấu trúc rẽ nhánh

IF ELSE

```
;if AX<0
    CMP     AX,0          ;AX<0 ?
    JNL     END_IF        ;không, thoát ra, else!
;then
    NEG     AX            ;đúng, đổi dấu AX
END_IF:
```

Ví dụ 2:

```
;if
    CMP     AL,'A'
    JB      khongphaichucaiaZ ;else
;then
    CMP     AL,'Z'
    JA      khongphaichucaiaZ ;else
; la chu cai in hoa A-Z
    MOV     AH,2          ;in chu cai in hoa
    MOV     DL,AL
    INT     21h
khongphaichucaiaZ:
```

;in thong bao không phải kí tự in hoa

41

If else (tiếp)

Ví dụ 2:

```
cmp     al,'A'
jb      khongphaichucaia
cmp     al,'Z'
ja      khongphaichucaiahoa
mov     ah,2
mov     dl,al
int     21h
```

42

Vòng lặp “for”

```
;khởi tạo CX=số lần lặp
Top:
;thân vòng lặp
LOOP Top
```

Ví dụ: In các chữ cái hoa A-Z

```
MOV     CX,26      ;CX chứa số lần lặp
MOV     AH,2
MOV     DL,'A'

InKiTu:
INT     21H
INC     DL
LOOP    InKiTu
```

43

Vòng lặp “while”

Ví dụ: In các chữ cái hoa A-Z

```
MOV     AH,2
MOV     DL,'A'

While_:
INT     21H
INC     DL
CMP     DL,'Z'
JBE     While_

End_While:
;các lệnh sau while
```

44

Chương trình con

TênCT PROC

;Các lệnh

RET ;thoát CTC trở về nơi gọi

TênCTC ENDP

Ví dụ:

```
.DATA
Xuongdong      10,13,"$"
.CODE
.....
Enter          PROC
    Mov        AH,9
    Lea        DX,xuongdong
    Int        21h
    Ret
Enter          ENDP
```

45

Thao tác với chuỗi ký tự

Nhập chuỗi kết thúc bằng Enter:

– Hàm 0Ah, ngắt 21h

Ví dụ:

```
.data
S db 100,?,101 dup('$') ;chuỗi dài tối đa 100 kí tự
;vị trí ? chứa độ dài thực sự
;khởi gán 101 kí tự $, nơi chứa kí tự nhập

.code
    mov        ah,0Ah
    lea        dx,s      ;dx trở vào đầu chuỗi
    int        21h

Giả sử ta nhập chuỗi Hello rồi enter, khi đó
```

46

Bài tập

- Bài 1. Nhập vào 1 ký tự và hiển thị ra màn hình
 Bài 2. Nhập vào một chuỗi và hiển thị chuỗi này ra màn hình
 Bài 3. Đọc hiểu bài “Nhập một chuỗi ký tự và chuyển chữ thường thành chữ hoa” trong tài liệu
 Bài 4. Lập trình thực hiện các công việc sau:
- Hiển thị thông báo : 'Hãy gõ vào một chữ cái hoa: '
 - Vào một chữ cái hoa
 - Xuống dòng, về đầu dòng
 - Hiển thị thông báo : 'Chữ cái thường là: '
 - Hiển thị chữ cái thường tương ứng
 - Xuống dòng về đầu dòng
 - Hiển thị thông báo : 'Mã ASCII của chữ cái là: '
 - Hiển thị mã ASCII của chữ cái.
 - Thoát về DOS.

47

3.Các phương pháp định địa chỉ

Khái niệm về định địa chỉ (addressing)

- Toán hạng của lệnh có thể là:
 - Một giá trị cụ thể nằm ngay trong lệnh
 - Nội dung của thanh ghi
 - Nội dung của ngăn nhớ hoặc cổng vào-ra
- Phương pháp định địa chỉ là cách thức địa chỉ hóa trong trường địa chỉ của lệnh để xác định toán hạng

48

Các phương pháp định địa chỉ thông dụng

- Định địa chỉ tức thì
- Định địa chỉ thanh ghi
- Định địa chỉ trực tiếp
- Định địa chỉ gián tiếp qua thanh ghi
- Định địa chỉ dịch chuyển

49

Định địa chỉ tức thì

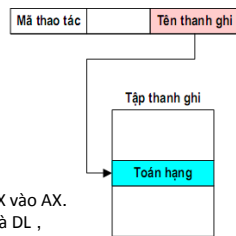


- Toán hạng đích: Thanh ghi, ô nhớ
- Toán hạng nguồn: Hằng số

50

Định địa chỉ thanh ghi

- Sử dụng các thanh ghi như là toán hạng để chứa dữ liệu cần để thao tác

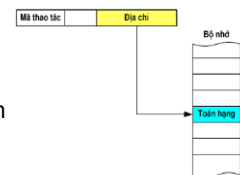


MOV AX, BX ; chuyển nội dung BX vào AX.
ADD DS, DL ; cộng nội dung AL và DL ,
; kết quả giữ trong AL.

51

Định địa chỉ trực tiếp

- Toán hạng: Địa chỉ ô nhớ để chứa dữ liệu
- Toán hạng kia: Thanh ghi

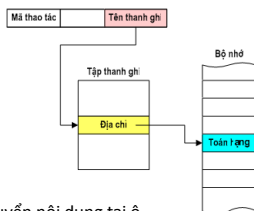


MOV AL, DS:1234H ; chuyển nội dung ô nhớ
;DS:1234H vào AL.
MOV DS:4321H, CX ; chuyển nội dung CX vào 2 vị trí nh
; liên tiếp là DS:4321 và DS:4322.

52

Định địa chỉ gián tiếp qua thanh ghi

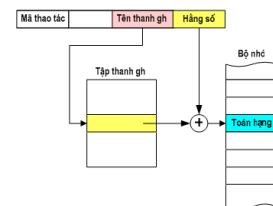
- Toán hạng: Thanh ghi chứa địa chỉ ô nhớ của dữ liệu
- Toán hạng kia: Thanh ghi



MOV AL, DS:BX ; chuyển nội dung tại ô
; nhớ DS:BX vào AL.
MOV DS:SI, CL ; chuyển nội dung CL vào
; ô nhớ DS:SI.

53

Định địa chỉ tương đối



- Để xác định toán hạng gồm 2 thành phần
 - Tên thanh ghi và hằng số
 - Địa chỉ toán hạng = Nội dung thanh ghi + hằng số

54

Ví dụ: Định địa chỉ tương đối

Tương đối chỉ số

MOV AL, DS:[SI+10] ; chuyển nội dung ô nhớ DS:(SI+10)
; vào AL.

Tương đối cơ sở

MOV CX, DS:[BX + 10] ; chuyển nội dung 2 ô nhớ liên tiếp có
; địa chỉ DS:(BX+10) và DS:(BX+11)
; vào CX.

55