

## CHƯƠNG IV: CÁC LỆNH LOGIC, DỊCH VÀ QUAY

### 4.1 Các lệnh logic

#### 4.1.1 Các phép toán logic

##### a) Phép toán AND (và)

Quy tắc thực hiện phép toán AND giữa hai số nhị phân A và B được trình bày trong bảng sau:

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

Ví dụ:

Cho M = 16h, N = 0Dh, hãy tính M AND N = ?

Giải:

$$\begin{array}{rcl}
 M & = & 0001\ 0110b \quad (16h) \\
 N & = & 0000\ 1101b \quad (0Dh) \\
 M \text{ AND } N & = & 0000\ 0100b = 04h
 \end{array}$$

##### b) Phép toán OR (hoặc)

Quy tắc thực hiện phép toán OR giữa hai số nhị phân A và B được trình bày trong bảng sau:

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

Ví dụ:

Cho M = 16h, N = 0Dh, hãy tính M OR N = ?

Giải:

$$\begin{array}{rcl}
 M & = & 0001\ 0110b \quad (16h) \\
 N & = & 0000\ 1101b \quad (0Dh) \\
 M \text{ OR } N & = & 0001\ 1111b = 1Fh
 \end{array}$$

##### c) Phép toán NOT (phủ định)

Quy tắc thực hiện phép toán NOT giữa hai số nhị phân A và B được trình bày trong bảng sau:

A	NOT A
0	1
1	0

Ví dụ:

Cho M = 16h, hãy tính NOT M = ?

Giải:

$$\begin{array}{rcl}
 M & = & 0001\ 0110b \quad (16h) \\
 \text{NOT } M & = & 1110\ 1001b = E9h
 \end{array}$$

d) *Phép toán XOR (hoặc - phủ định)*

Quy tắc thực hiện phép toán XOR giữa hai số nhị phân A và B được trình bày trong bảng sau:

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Ví dụ:

Cho M = 16h, N = 0Dh, hãy tính M XOR N = ?

Giải:

$$\begin{array}{rcl}
 M & = & 0001\ 0110b \quad (16h) \\
 N & = & 0000\ 1101b \quad (0Dh) \\
 M \text{ XOR } N & = & \underline{0001\ 1011b} = 1Bh
 \end{array}$$

#### 4.1.2 Các lệnh logic trong hợp ngữ

Tương ứng với các phép toán logic trên, hợp ngữ có các lệnh sau đây:

AND <Đích>, <Nguồn>

OR <Đích>, <Nguồn>

XOR <Đích>, <Nguồn>

NOT <Đích>

<Đích>: là một thanh ghi hay một ô nhớ

<Nguồn>: là một thanh ghi, một ô nhớ, hoặc một hằng số

<Đích>, <Nguồn> không đồng thời là hai ô nhớ.

Ví dụ:

AND AX, 002Ah

OR AL, 3Dh

NOT BX

#### 4.1.3 Ứng dụng các lệnh logic

Ví dụ 1:

Hãy thay đổi bit dấu trong thanh ghi AX.

XOR AX, 8000h

Ví dụ 2:

Hãy xoá bit LSB trong thanh ghi BH.

AND BH, 0FEh

Ví dụ 3:

Nhập một kí tự số từ bàn phím ('0', '1', ... , '9'), đổi nó sang số thập phân tương ứng.

Giải:

Ta sẽ sử dụng các lệnh logic để chuyển đổi kí tự sang số.

```

TITLE  VI DU 3
.MODEL  SMALL
.STACK  100H
.CODE
MAIN    PROC
NhapLai:
    MOV    AH, 1          ;Nhập một kí tự
    INT     21h

    CMP     AL, '0'
    JB      NhapLai       ;Nếu AL < '0' thì nhập lại
    CMP     AL, '9'
    JA      NhapLai       ;Nếu AL > '9' thì nhập lại

    AND     AL, 0Fh       ;Đổi sang số thập phân tương ứng
                        ; (xoá 4 bit cao của AL)
    ...           ;Các lệnh khác

    MOV     AH, 4Ch       ;Kết thúc
    INT     21h

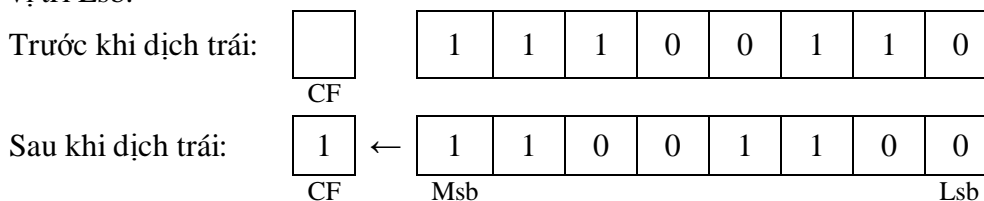
MAIN ENDP
END MAIN
    
```

## 4.2 Các lệnh dịch

### 4.2.1 Các phép toán dịch

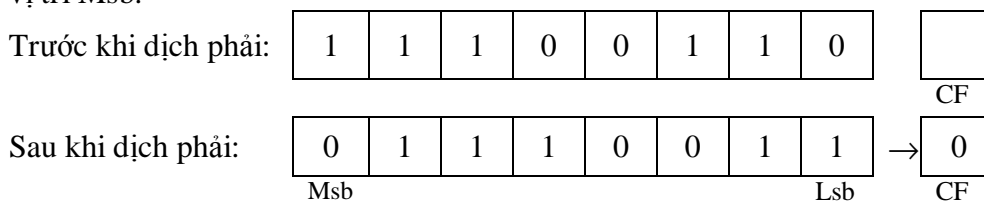
#### a) Phép dịch trái

Xét một dãy bit trong một thanh ghi hoặc một ô nhớ: phép dịch trái sẽ dịch chuyển toàn bộ các bit trong dãy về bên trái, giá trị của Msb được đưa vào cờ CF, thêm bit 0 vào vị trí Lsb.



#### b) Phép dịch phải

Xét một dãy bit trong một thanh ghi hoặc một ô nhớ: phép dịch phải sẽ dịch chuyển toàn bộ các bit trong dãy về bên phải, giá trị của Lsb được đưa vào cờ CF, thêm bit 0 vào vị trí Msb.



### 4.2.2 Các lệnh dịch trong hợp ngữ

#### a) Lệnh dịch trái

Có thể sử dụng lệnh SHL (Shift Left) hoặc lệnh SAL (Shift Arithmetic Left), hai lệnh này tạo ra cùng một mã máy.

Cú pháp lệnh:

- Dạng 1: SHL <Đích>, 1
- Dạng 2: SHL <Đích>, CL

<Đích>: là một thanh ghi hay một ô nhớ

Dạng 1 sẽ dịch các bit của toán hạng đích sang trái 1 lần, dạng 2 sẽ dịch các bit của toán hạng đích sang trái nhiều lần, số lần dịch chứa trong thanh ghi CL.

Ví dụ 1:

SHL AX, 1 ;Dịch các bit của thanh ghi AX sang trái 1 lần

Ví dụ 2:

MOV CL, 3

SHL AX, CL ;Dịch các bit của thanh ghi AX sang trái 3 lần

- Ứng dụng của lệnh dịch trái:

Một trong số các ứng dụng của lệnh dịch trái là thực hiện phép nhân với 2. Xét ví dụ sau đây:

Giả sử AH = 0001 1010b =  $1 \times 2^4 + 1 \times 2^3 + 1 \times 2^1 = 26$

Sau khi dịch trái 1 lần thì AH = 0011 0100b =  $1 \times 2^5 + 1 \times 2^4 + 1 \times 2^2$   
 $= 2 \times (1 \times 2^4 + 1 \times 2^3 + 1 \times 2^1) = 52$

Như vậy, phép dịch trái 1 lần tương đương phép nhân toán hạng đích với 2.

Tổng quát: Phép dịch trái N lần tương đương phép nhân toán hạng đích với  $2^N$ .

Chú ý: Kết luận trên chỉ đúng khi không có hiện tượng tràn xảy ra.

Ví dụ:

Giả sử AH = 1000 0001b = 129

Sau khi dịch trái 1 lần thì AH = 0000 0010b = 2: Không phải là phép nhân với 2.

Bit Msb của AH được chuyển vào cờ CF: cờ CF = 1 báo hiệu hiện tượng tràn xảy ra, kết quả không còn đúng nữa.

#### b) Lệnh dịch phải

Có hai loại lệnh dịch phải: SHR (Shift Right) và SAR (Shift Arithmetic Right), tác dụng của chúng không hoàn toàn giống nhau.

Cú pháp lệnh SHR:

- Dạng 1: SHR <Đích>, 1
- Dạng 2: SHR <Đích>, CL

<Đích>: là một thanh ghi hay một ô nhớ

Dạng 1 sẽ dịch các bit của toán hạng đích sang phải 1 lần, dạng 2 sẽ dịch các bit của toán hạng đích sang phải nhiều lần, số lần dịch chứa trong thanh ghi CL.

Ví dụ 1:

SHR BX, 1 ;Dịch các bit của thanh ghi BX sang phải 1 lần

Ví dụ 2:

MOV CL, 3

SHR BX, CL ;Dịch các bit của thanh ghi BX sang phải 3 lần

- Ứng dụng của lệnh dịch phải:

Một trong số các ứng dụng của lệnh dịch phải là thực hiện phép chia cho 2. Xét ví dụ sau đây:

Giả sử  $AL = 0001\ 0000b = 1 \times 2^4 = 16$

Sau khi dịch phải 1 lần thì  $AL = 0000\ 1000b = 1 \times 2^3 = 8$

Như vậy, phép dịch phải 1 lần tương đương phép chia toán hạng đích cho 2.

Tổng quát: Phép dịch phải N lần tương đương phép chia toán hạng đích cho  $2^N$ .

Chú ý: Kết luận trên chỉ đúng nếu không làm thay đổi bit dấu khi dịch phải.

Ví dụ:

Giả sử  $AL = 1000\ 0001b = -127$

Sau khi dịch phải 1 lần thì  $AL = 0100\ 0000b = 64$ : Không phải là phép chia cho 2.

Do đó, đối với các số có dấu ta không được sử dụng lệnh SHR mà phải sử dụng lệnh SAR. Lệnh SAR sẽ giữ nguyên bit dấu khi dịch phải.

Ví dụ:

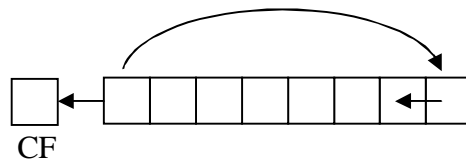
Ban đầu  $AL = 1000\ 0001b = -127$

Sau lệnh SAR AL, 1 thì  $AL = 1100\ 0000b = -64$  (bit dấu vẫn được giữ nguyên).

### 4.3 Các lệnh quay

#### 4.3.1 Lệnh quay trái (ROL - Rotate Left)

Lệnh quay trái cũng gần giống với lệnh dịch trái, chỉ khác ở chỗ bit Msb vừa được đưa vào cờ CF, vừa được đưa trở lại vị trí Lsb.

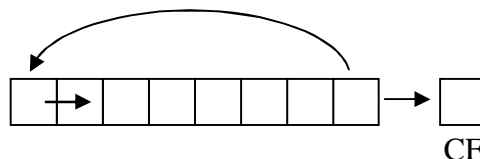


Cú pháp lệnh:

- Dạng 1: `ROL <Đích>, 1` ;Quay trái một lần
  - Dạng 2: `ROL <Đích>, CL` ;Quay trái nhiều lần, CL chứa số lần quay
- <Đích>: là một thanh ghi hay một ô nhớ

#### 4.3.2 Lệnh quay phải (ROR - Rotate Right)

Lệnh quay phải cũng gần giống với lệnh dịch phải, chỉ khác ở chỗ bit Lsb vừa được đưa vào cờ CF, vừa được đưa trở lại vị trí Msb.

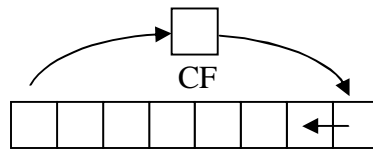


Cú pháp lệnh:

- Dạng 1: ROR <Đích>, 1 ;Quay phải một lần
  - Dạng 2: ROR <Đích>, CL ;Quay phải nhiều lần, CL chứa số lần quay
- <Đích>: là một thanh ghi hay một ô nhớ

#### 4.3.3 Lệnh quay trái qua cờ CF (RCL – Rotate Carry Left)

Lệnh này cũng gần giống với lệnh dịch trái, chỉ khác ở chỗ bit Msb được đưa vào cờ CF, còn nội dung cờ CF lại được đưa vào vị trí Lsb.

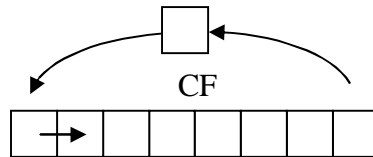


Cú pháp lệnh:

- Dạng 1: RCL <Đích>, 1 ;Quay một lần
  - Dạng 2: RCL <Đích>, CL ;Quay nhiều lần, CL chứa số lần quay
- <Đích>: là một thanh ghi hay một ô nhớ

#### 4.3.4 Lệnh quay phải qua cờ CF (RCR – Rotate Carry Right)

Lệnh này cũng gần giống với lệnh dịch phải, chỉ khác ở chỗ bit Lsb được đưa vào cờ CF, còn nội dung cờ CF lại được đưa vào vị trí Msb.



Cú pháp lệnh:

- Dạng 1: RCR <Đích>, 1 ;Quay một lần
  - Dạng 2: RCR <Đích>, CL ;Quay nhiều lần, CL chứa số lần quay
- <Đích>: là một thanh ghi hay một ô nhớ

- Ứng dụng lệnh quay:

Lệnh quay có nhiều ứng dụng khác nhau, dưới đây chỉ xét một trường hợp đơn giản:  
Ví dụ: Đếm số bit 1 trong thanh ghi AX

Giải:

Ta sẽ thực hiện lệnh quay AX 16 lần (quay trái hay quay phải đều được). Mỗi lần quay thì một bit sẽ được đưa vào cờ CF, nếu bit đó bằng 1 thì tăng BX lên 1 (BX chứa giá trị đếm được). Sau 16 lần quay thì thanh ghi AX sẽ trở lại giá trị ban đầu.

```
MOV    CX, 16          ;CX chứa số lần lặp
XOR     BX              ;Xoá BX để chuẩn bị chứa số lượng bit 1
```

Lap:

ROL	AX, 1	;Quay trái AX 1 lần
JNC	TiepTuc	;Nếu CF = 0 (gặp بیت 0) thì nhảy
INC	BX	;Nếu gặp بیت 1 thì tăng BX
TiepTuc:		

LOOP Lap