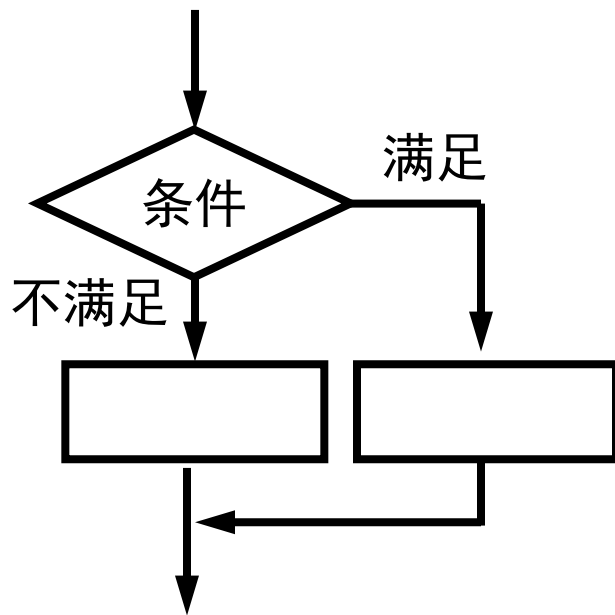


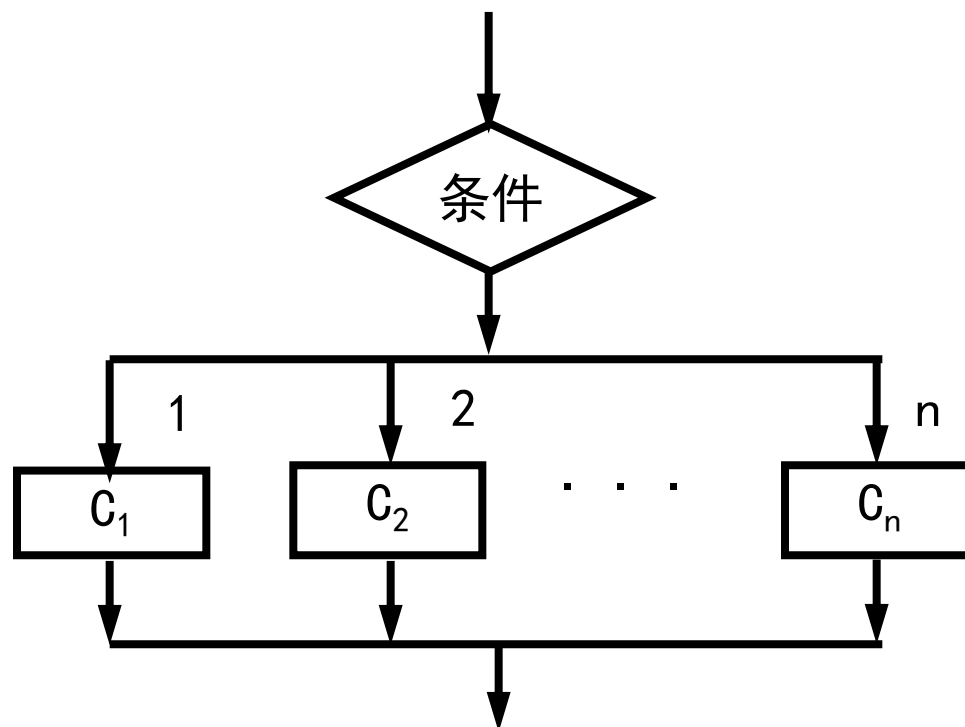
- 控制转移指令
- 循环结构程序设计
- 分支结构程序设计

分枝结构程序设计

- 在程序中，往往需要对不同的情况或条件进行不同的处理，这样的程序就不再是简单的顺序结构，而要采用分枝结构。
- 分支程序结构可以有两种基本形式，即双分支结构和多分支结构。
- 不论哪种形式，共同点是：运行方向是向前的，在某一特定条件下，只能执行多个分支中的一个分支。



IF__THEN__ELSE结构



CASE结构

分枝结构程序设计

双分枝结构：

使用条件转移指令与无条件转移指令JMP来实现分支：一般必须先进行比较或算术、逻辑运算等影响标志位的指令，然后用条件转移指令判断条件，以实现分支转移。

IF-THEN结构：

```
cmp AX, BX
```

```
JE EndOfIF
```

```
<THEN 程序段>
```

EndOfIF:

注意：程序隐含是顺序执行的，在THEN分支体执行后，不会自动跳过ELSE分支体，而是继续执行其后的代码。

IF-THEN-ELSE结构：

```
cmp AX, BX
```

```
JE ElseCode
```

```
<THEN 程序段>
```

```
jmp EndOfIF
```

ElseCode:

```
<ELSE 程序段>
```

EndOfIF:

分枝结构程序设计

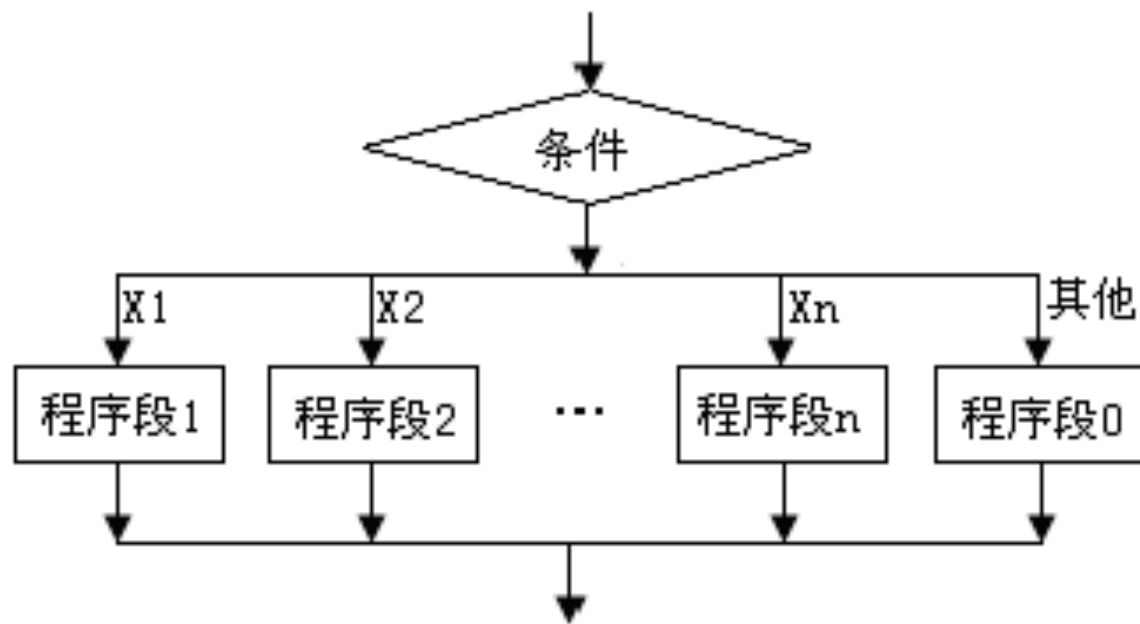
例： 已知在内存中有一个字节单元NUM，存有带符号数据，要求计算出它的绝对值后，放入RESULT单元中。

```
DATA    SEGMENT
        X      DB    -25
        RESULT DB    ?
DATA ENDS
CODE SEGMENT
        ASSUME DS:DATA,CS:CODE
START:   MOV     AX, DATA
         MOV     DS, AX           ;初始化
         MOV     AL, X           ;X取到AL中
         TEST    AL, 80H         ;测试AL正负
         JZ      NEXT           ;为正，转NEXT
         NEG     AL              ;否则AL求补
NEXT:    MOV     RESULT, AL      ;送结果
         MOV     AH, 4CH
         INT     21H             ;返回DOS
CODE ENDS
        END     START           ;汇编结束
```

分枝结构程序设计

多分支结构：

- 多分支结构是指有两个以上的分支。在程序设计时，有时要求对多个条件同时进行判断，根据判断的结果，可能有多个分支要进行处理。
- 在汇编语言中，多分支只能由多次使用单分支方式予以实现。
- 设计方法：双分支法、逻辑分解法、跳跃表法、转移表法。



分枝结构程序设计

双分支法：由多个双分支实现多分枝程序设计

例. 实现符号函数Y的功能。

其中： $-128 \leq X \leq +127$

$$Y = \begin{cases} 1 & \text{当 } X > 0 \text{ 时} \\ 0 & \text{当 } X = 0 \text{ 时} \\ -1 & \text{当 } X < 0 \text{ 时} \end{cases}$$

分枝结构程序设计

双分支法：由多个双分支实现多分枝程序设计

例. 实现符号函数Y的功能。

其中： $-128 \leq X \leq +127$

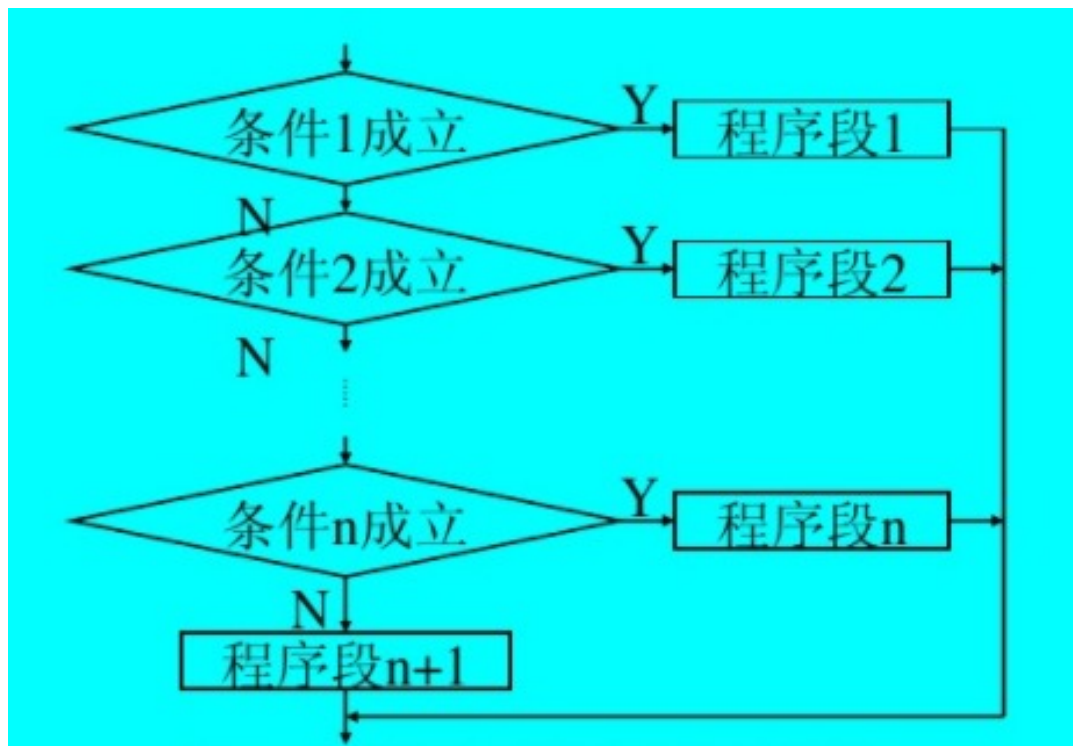
$$Y = \begin{cases} 1 & \text{当 } X > 0 \text{ 时} \\ 0 & \text{当 } X = 0 \text{ 时} \\ -1 & \text{当 } X < 0 \text{ 时} \end{cases}$$

代码：

```
X      DB      ?           ;被测数据
Y      DB      ?           ;函数值单元
...
      MOV      AL, 0
      CMP      X, AL
      JG      BIG           ;大于0
      JZ      SAV           ;等于0
      MOV      AL, 0FFH    ;小于0
      JMP      SAV
BIG:   MOV      AL, 1       ;大于0
SAV:   MOV      Y, AL      ;保存结果
```

分枝结构程序设计

逻辑分解法：按条件成立的先后，依次逻辑分解成下图所示的一串双分支结构，然后使用双分支的方法来进行程序设计。



分枝结构程序设计

例：根据输入值（0-4）的不同，执行不同的操作。

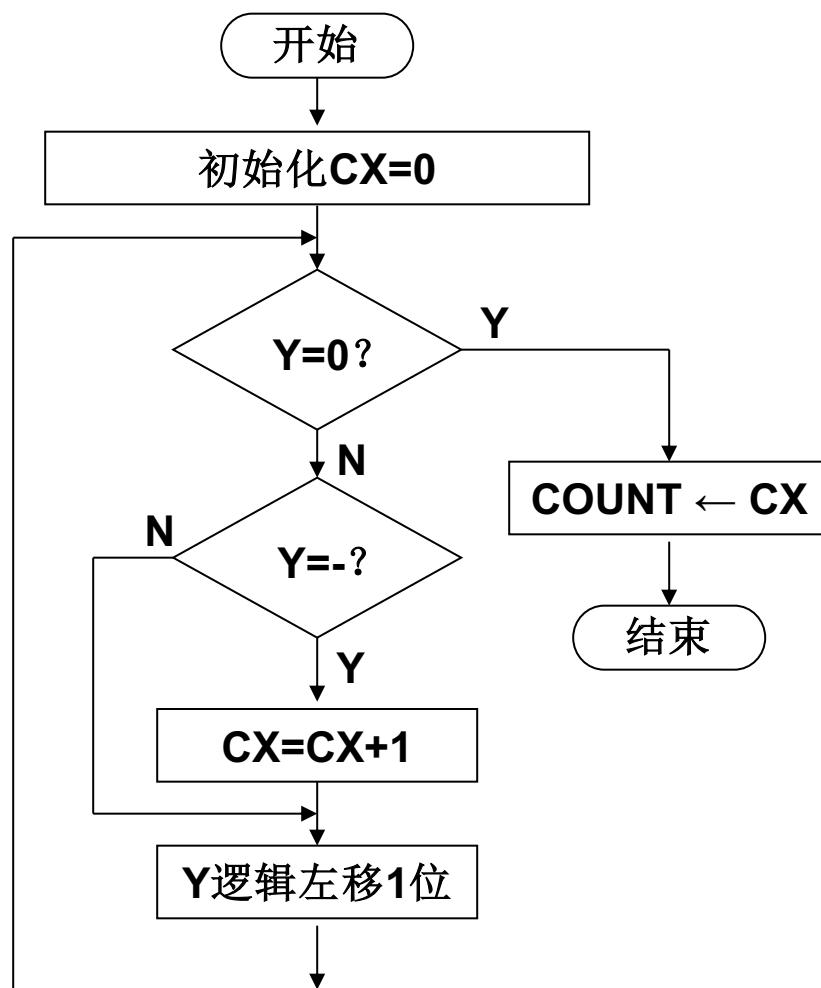
```
DATA SEGMENT
NUM DB 2
DATA ENDS
CODE SEGMENT
        ASSUME CS:CODE,DS:DATA
START:MOV AX,DATA
        MOV DS,AX
        MOV AL,NUM
        CMP AL,0
        JZ NEXT0
        CMP AL,1
        JZ NEXT1
        CMP AL,2
        JZ NEXT2
        CMP AL,3
        JZ NEXT3
        CMP AL,4
        JZ NEXT4
NEXT0:MOV DL,30H
        JMP EXIT
NEXT1:MOV DL,31H
        JMP EXIT
NEXT2:MOV DL,32H
        JMP EXIT
NEXT3:MOV DL,33H
        JMP EXIT
NEXT4:MOV DL,34H
EXIT:MOV AH,2
        INT 21H
        MOV AH,4CH
        INT 21H
CODE ENDS
        END START
```

循环结构程序设计

【例5.2】 在Y中存放着16位数，试编制一个程序把Y中1的个数存入COUNT单元中。

循环结构程序设计

【例5.2】在Y中存放着16位数，试编制一个程序把Y中1的个数存入COUNT单元中。

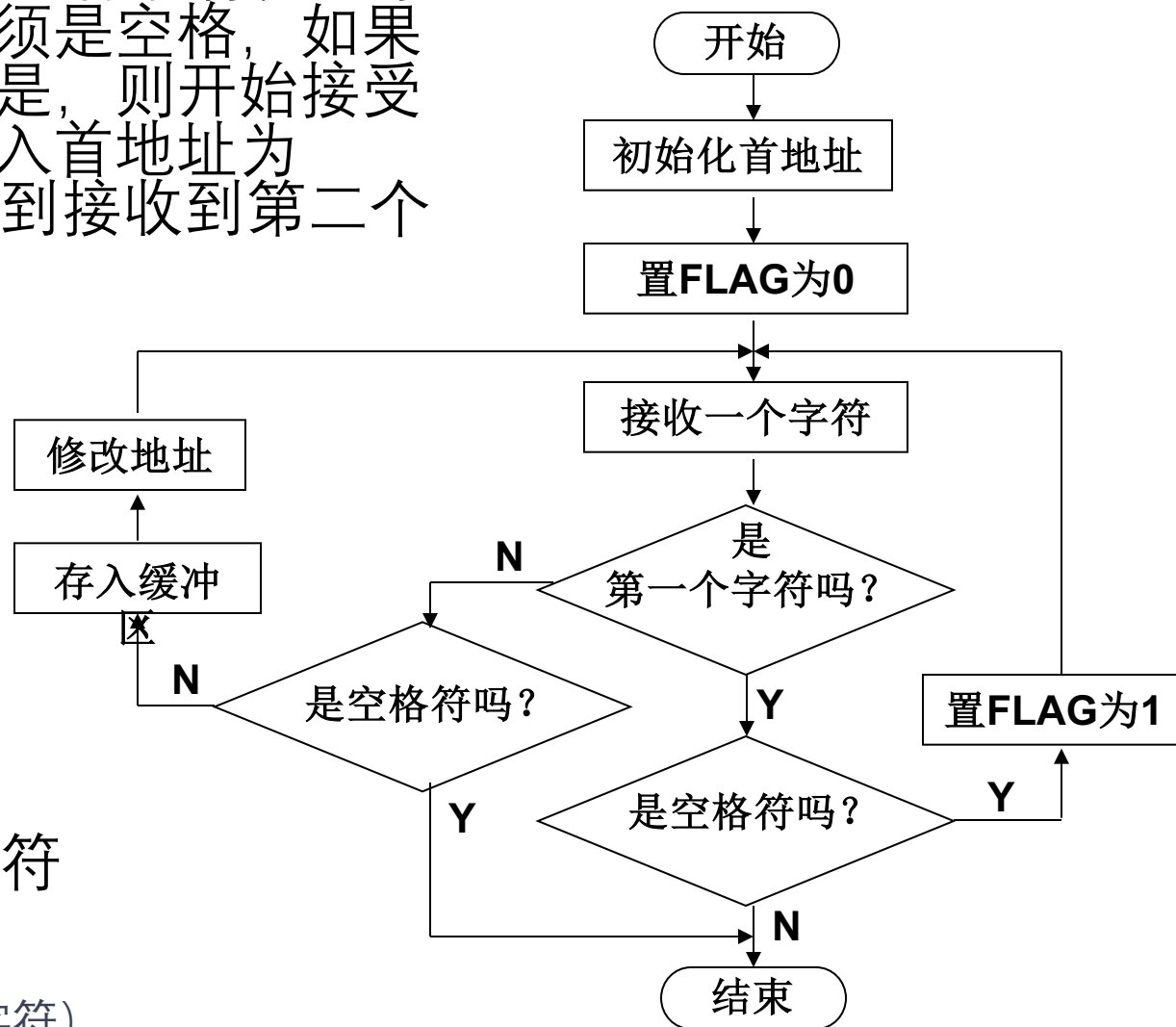


循环结构程序设计

```
data    segment
        Y      dw      06F3CH
        count  dw      ?
data    ends
prog    segment
        assume  cs: prog, ds: data
start:  mov     ax, data    ;数据段地址送ax
        mov     ds, ax
        mov     cx, 0      ;初始化CX=0
        mov     ax, Y      ;把数放到ax
testing: and    ax, ax     ;测试Y
        jz      exit      ;如果Y=0, 结束
        jns     shift     ;如果为正数, 不变
        inc     cx        ;否则CX=CX+1
shift:  shl     ax, 1      ;左移Y
        jmp     testing
exit:   mov     count, cx
        mov     ah, 4ch
        int     21h
prog    ends
        end     start
```

循环结构程序设计

【例5.6】从键盘输入一行字符，要求输入的~~第一个~~字符必须是空格，如果不是，则退出，如果是，则开始接受输入的字符并顺序存入首地址为BUFFER的缓冲区，直到接收到第二个空格为止。



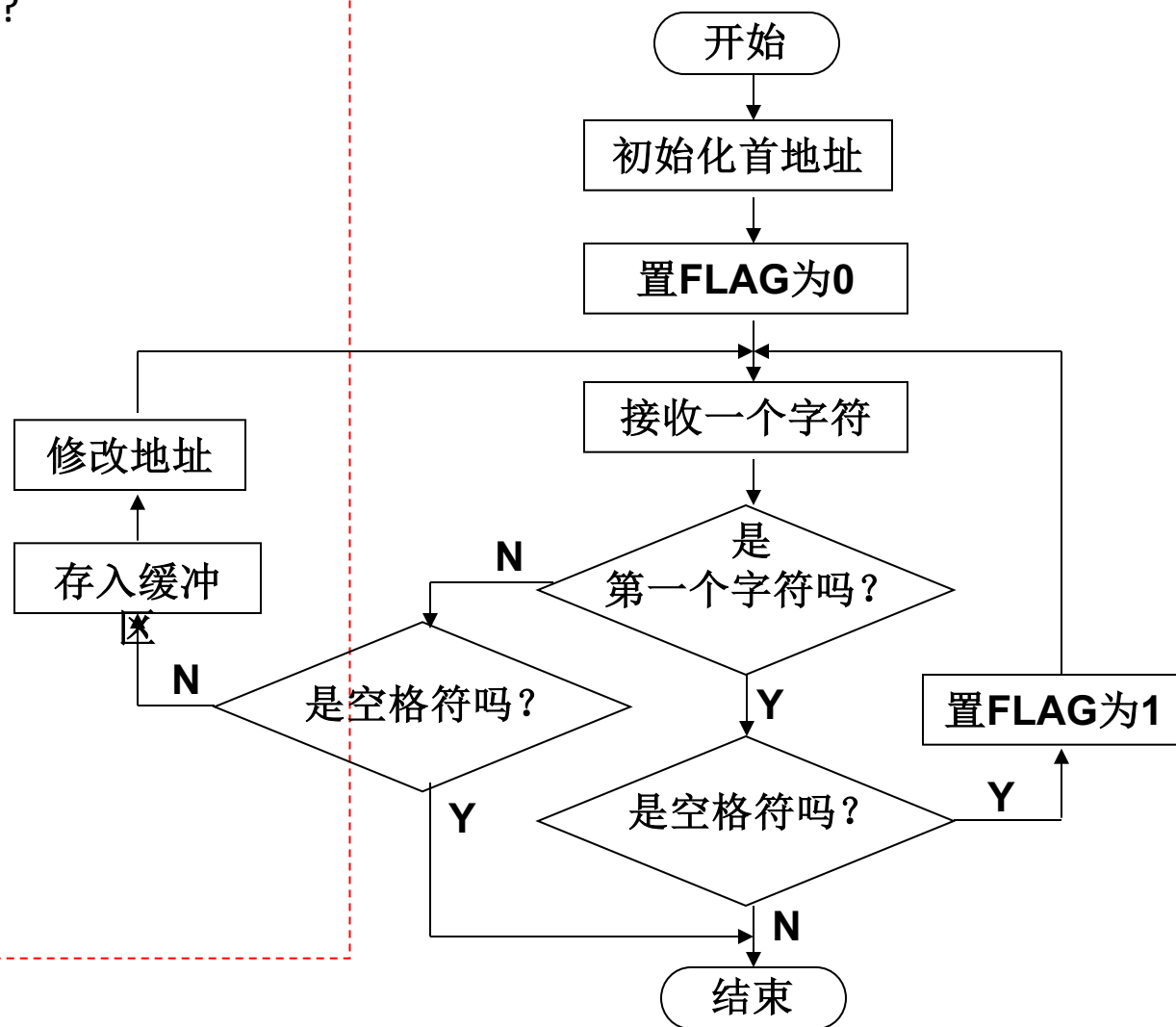
Flag=0: 第1个字符

Flag=1: 不是第1个字符

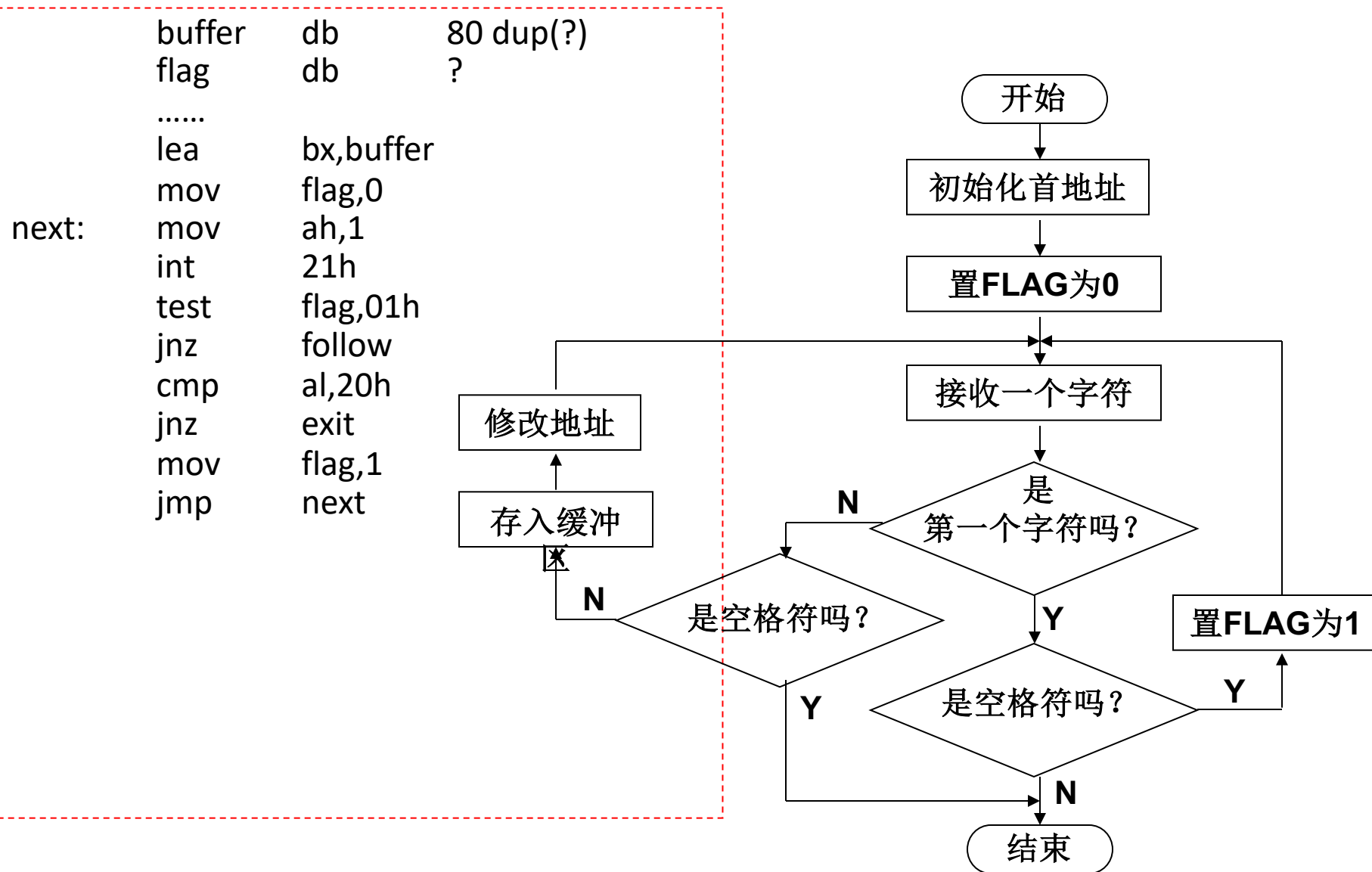
20h (十六进制, 等于空格字符)

循环结构程序设计

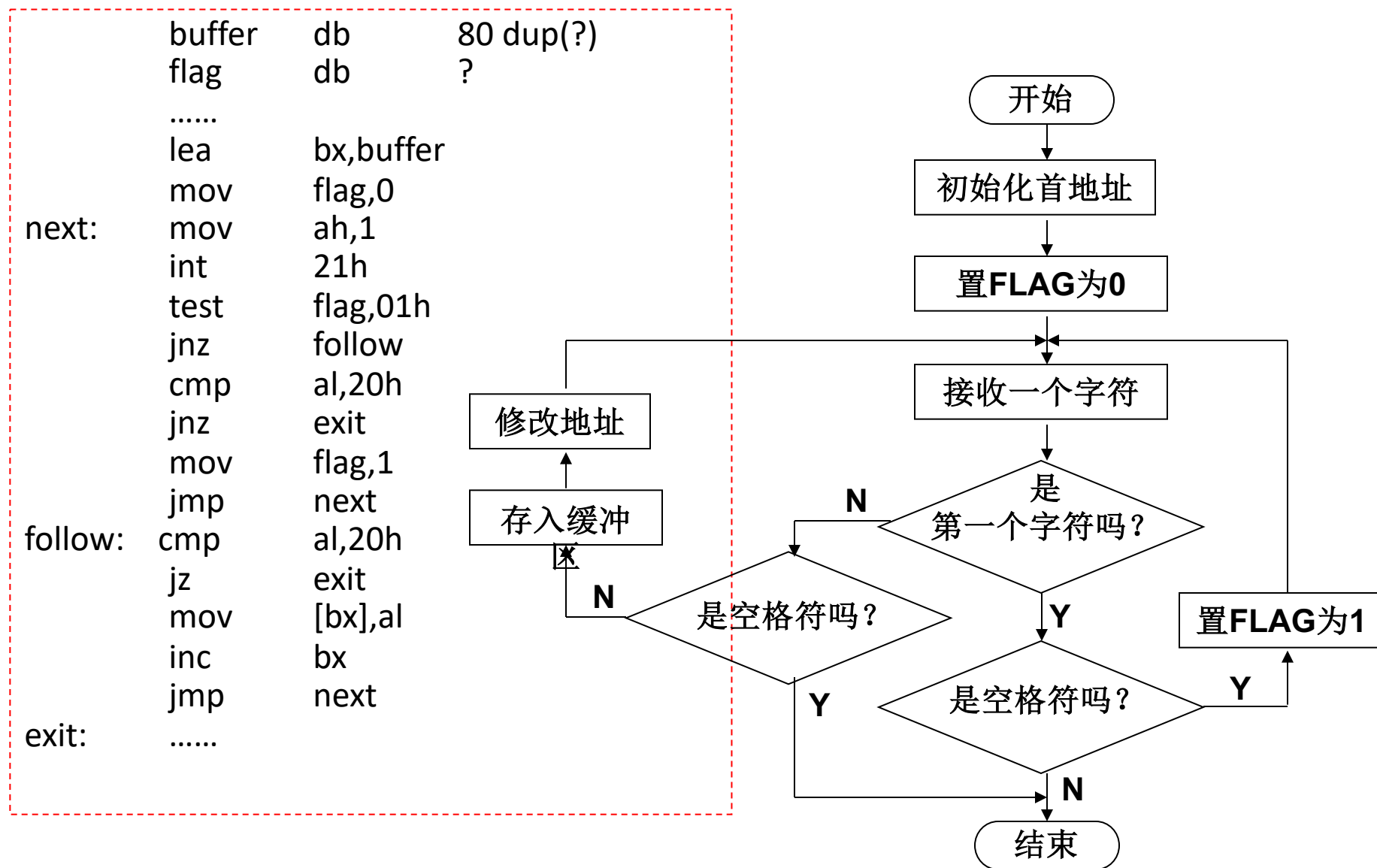
```
buffer    db    80 dup(?)
flag      db    ?
.....
lea       bx,buffer
mov       flag,0
```



循环结构程序设计



循环结构程序设计



Q&A



Fall 2023