

第五章 循环与分支程序设计

【教学目标】

- 1) 掌握程序设计中的基本结构划分
- 2) 学会顺序结构程序设计的方法
- 3) 学会分解、构造分支判定条件
- 3) 理解掌握循环构造要素
- 4) 掌握程序流程图画法，对典型算法要能灵活变形应用
- 5) 程序调试技巧

【重点要点】

- 1、数据段的定义；
- 2、数据的存储与读取方法；
- 3、循环要素、循环程序设计的方法；

- 编程求以下运算： $Y=X1-(X2+X3)$
(X, Y均为字节变量)

- 分析：
 - 将字节型变量X1、X2和X3存放在数据段开始的3个单元，Y单元定义为存放运算结果的单元
 - 题目的运算顺序是先完成X2和X3所对应数据的相加运算，结果暂时存在寄存器，再用X1减去这个和，得到的差存放到内存Y单元

参考例程

ASSUME CS:CODE,DS:DATA

DATA SEGMENT

X1 DB 39H

X2 DB 21H

X3 DB 13H

Y DB ?

DATA ENDS

CODE SEGMENTS

START: MOV AX, DATA

MOV DS, AX

MOV BX, OFFSET X2

MOV CL, [BX]

ADD CL, [BX+1]

MOV AL, [BX-1]

SUB AL, CL

MOV [BX+2], AL

MOV AH, 4CH

MOV 21H

CODE ENDS

END START

地址 存储单元

DS:0 | 39

1 | 21

2 | 13

3 | ?

顺序结构程序设计

- 首先将**FIRST**字变量与**SECOND**字变量相加，结果存至**THIRD1**存储字中，然后将**FIRST**与**SECOND**两个字变量相乘，结果存至**SECOND2**开始的两个字中
- 分析：
 - **FIRST**与**SECOND**都是字变量，因此存放和的**THIRD1**单元也必须是字变量。而存放积的单元**THIRD2**必须是双字
 - 此题只要变量类型定义正确，很容易编写

参考例程

地址 存储单元

DS:0 6C

1 | **D5**

208

3	1F
---	----

?

?

?

?

?

?

--	--

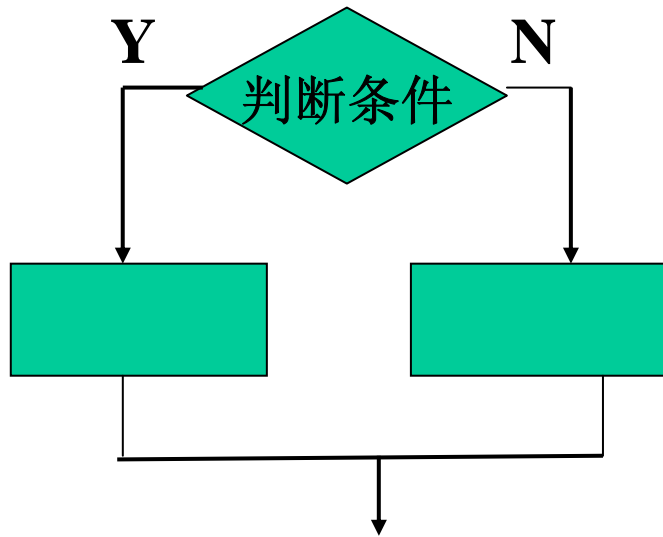
--	--

--	--

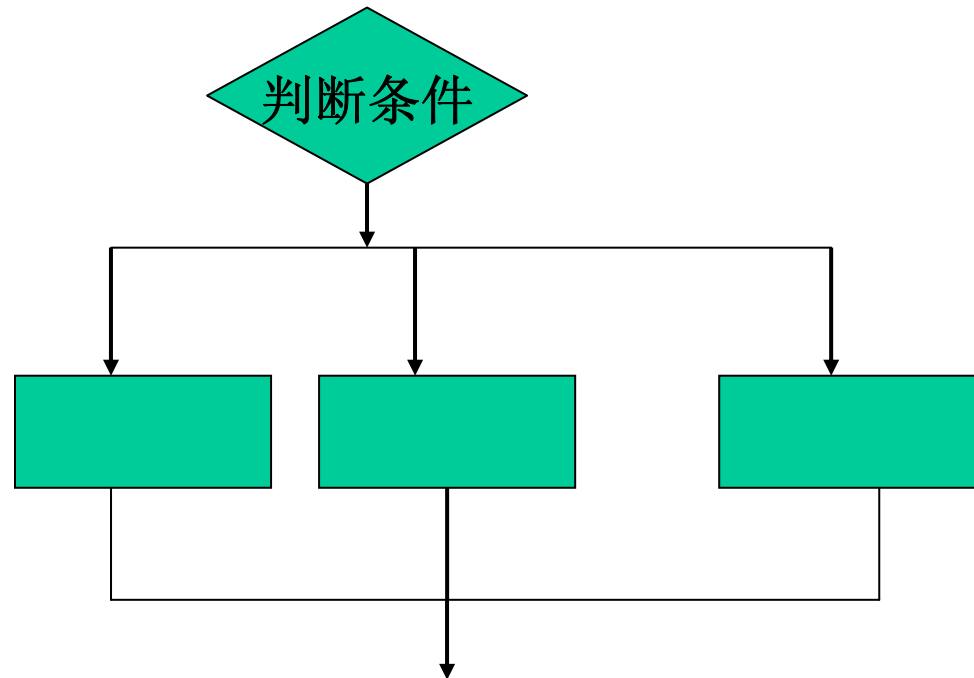
1. *Journal of the American Medical Association*, 2000; 283: 2689-2696.

分支程序的设计

分支程序的结构形式



IF-THEN-ELSE结构



CASE 结构

汇编程序中经常采用条件转移和无条件转移语句实现程序的分支结构。注意，判定过程中的状态标志应该来自于其前面的最近一条影响该标志位（或者几位）的指令。

分支程序设计的步骤：

- 1、利用比较指令**CMP**、测试指令**TEST**、加减运算、逻辑运算等影响状态标志的指令形成条件
- 2、选择合适的条件转移指令，利用条件转移指令判断由标志表达的条件
- 3、根据标志状态控制程序转移到不同的程序段

分支结构程序设计

- 编制实现变量X的符号函数程序，该符号函数可用下式表示：

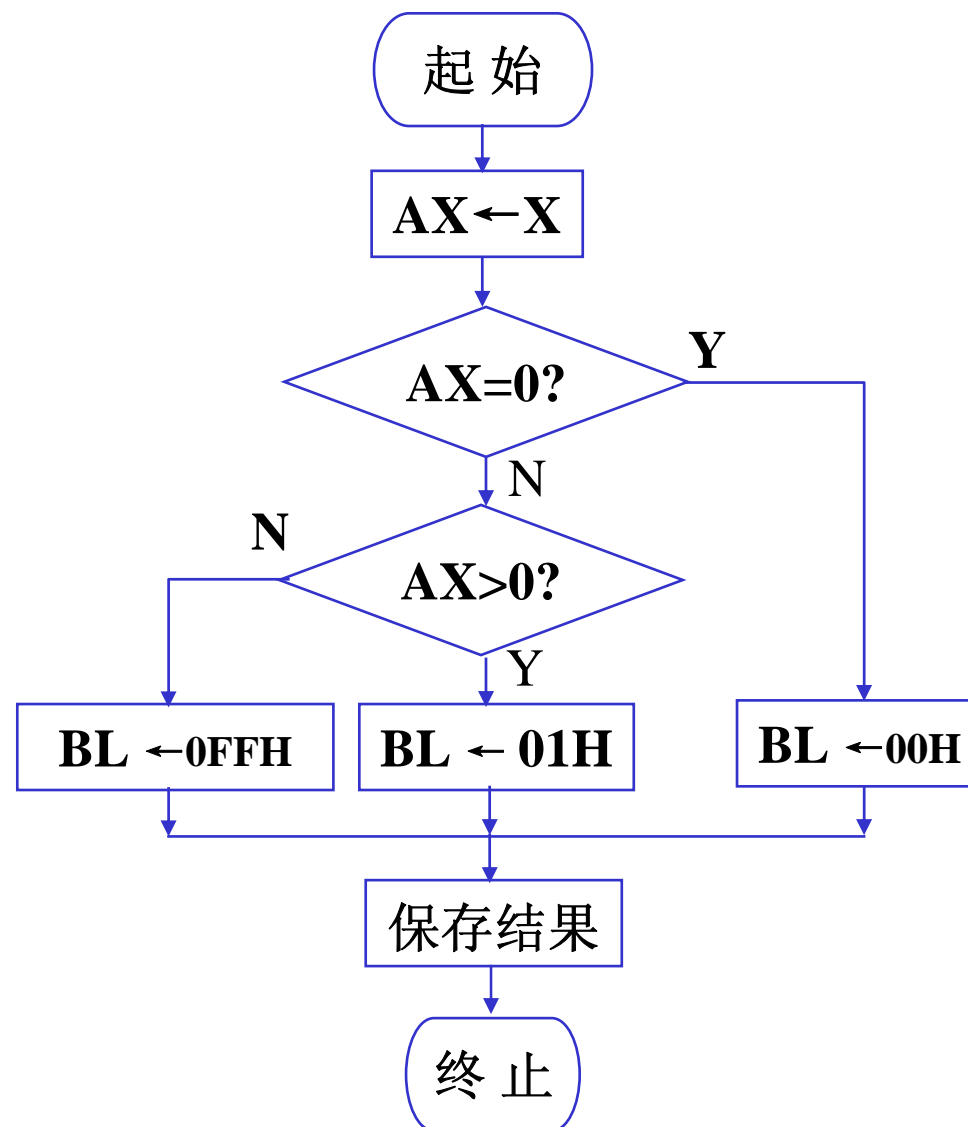
$$Y = \begin{cases} 1 & \text{当 } X > 0 \\ 0 & \text{当 } X = 0 \\ -1 & \text{当 } X < 0 \end{cases}$$

分析：Y的取值由X的符号决定，因此只要能判断出X的符号，即可得Y的值。

判断X的符号可让X直接与0比较，也可以用一条能影响标志位的指令来进行，例如“与”或“或”操作，就可将X的符号和是否为0反映到ZF与SF标志上。

如果X=0，则进入第1个分支，Y=0；如果X>0，则进入第2个分支，Y=1；否则，进入第3个分支，Y=-1。

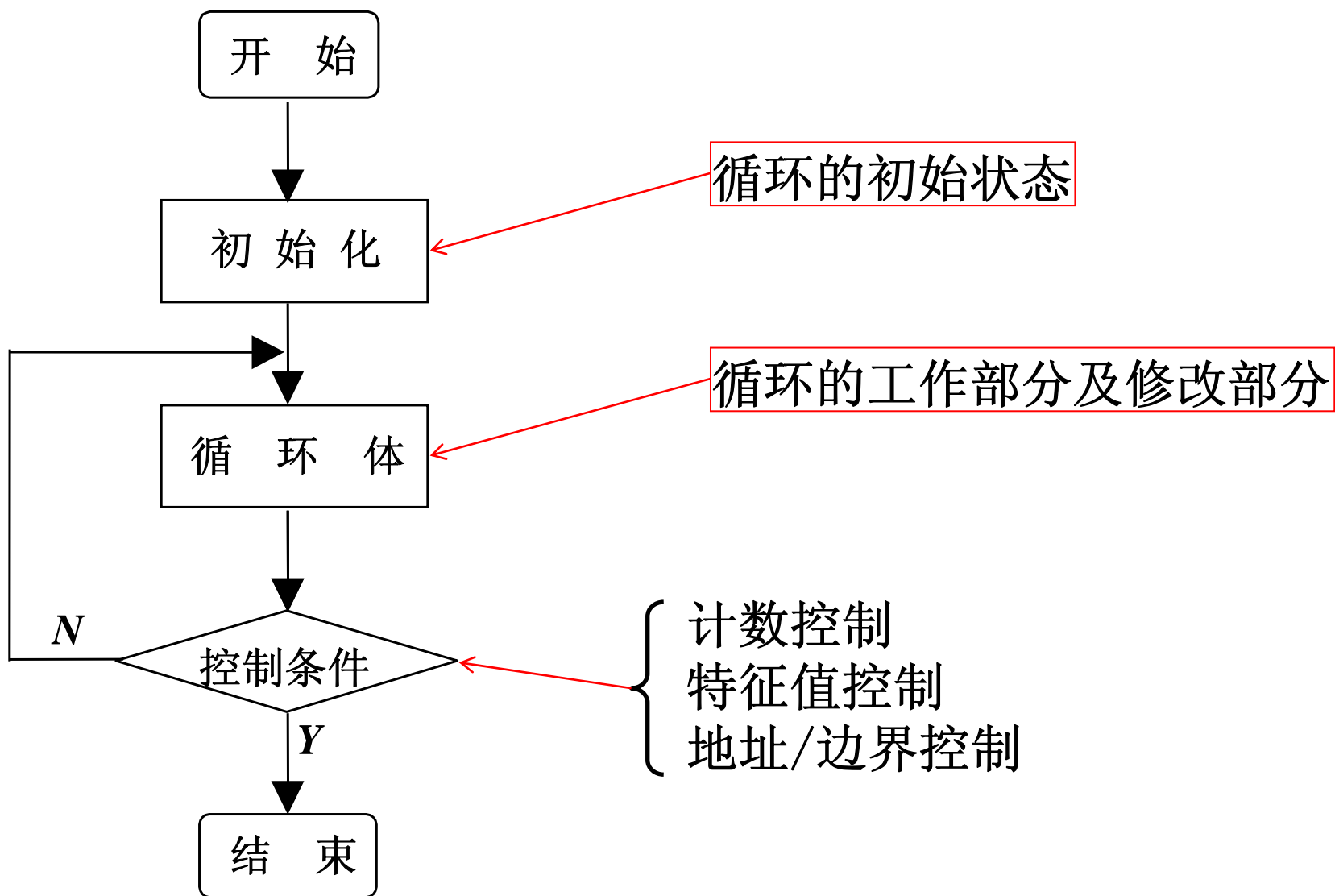
- 流程图



参考例程

```
STACK SEGMENT
    DB 256 DUP(?)
STACK ENDS
CODE SEGMENTS
    ASSUME CS:CODE,DS:CODE,SS:STACK
BEGIN: PUSH CS
        POP  DS
        MOV  AX, X
        OR   AX, AX
        JE   ZERO      ;为0
        JNE  PLUS      ;为正
        MOV  BL, 0FFH   ;为负
        JMP  EXIT
ZERO:   MOV  BL, 0
        JMP  EXIT
PLUS:   MOV  BL, 1
EXIT:   MOV  Y, BL
        MOV  AH, 4CH
        MOV  21H
        X    DW  300H    ;定义数据X
        Y    DB  ?       ;定义Y单元
CODE ENDS
        END BEGIN
```

循环程序的结构形式与构成要素



1. 循环的初始化

循环准备阶段，包括：建立地址指针、设置循环次数、必要的数据保护、循环的初始状态

2. 循环体

循环要完成的具体操作，需要重复执行的部分，是循环的核心部分

3. 循环的控制要素

a. 循环的条件: 值可变的逻辑表达式



b. 循环控制变量: 至少含有一个变量

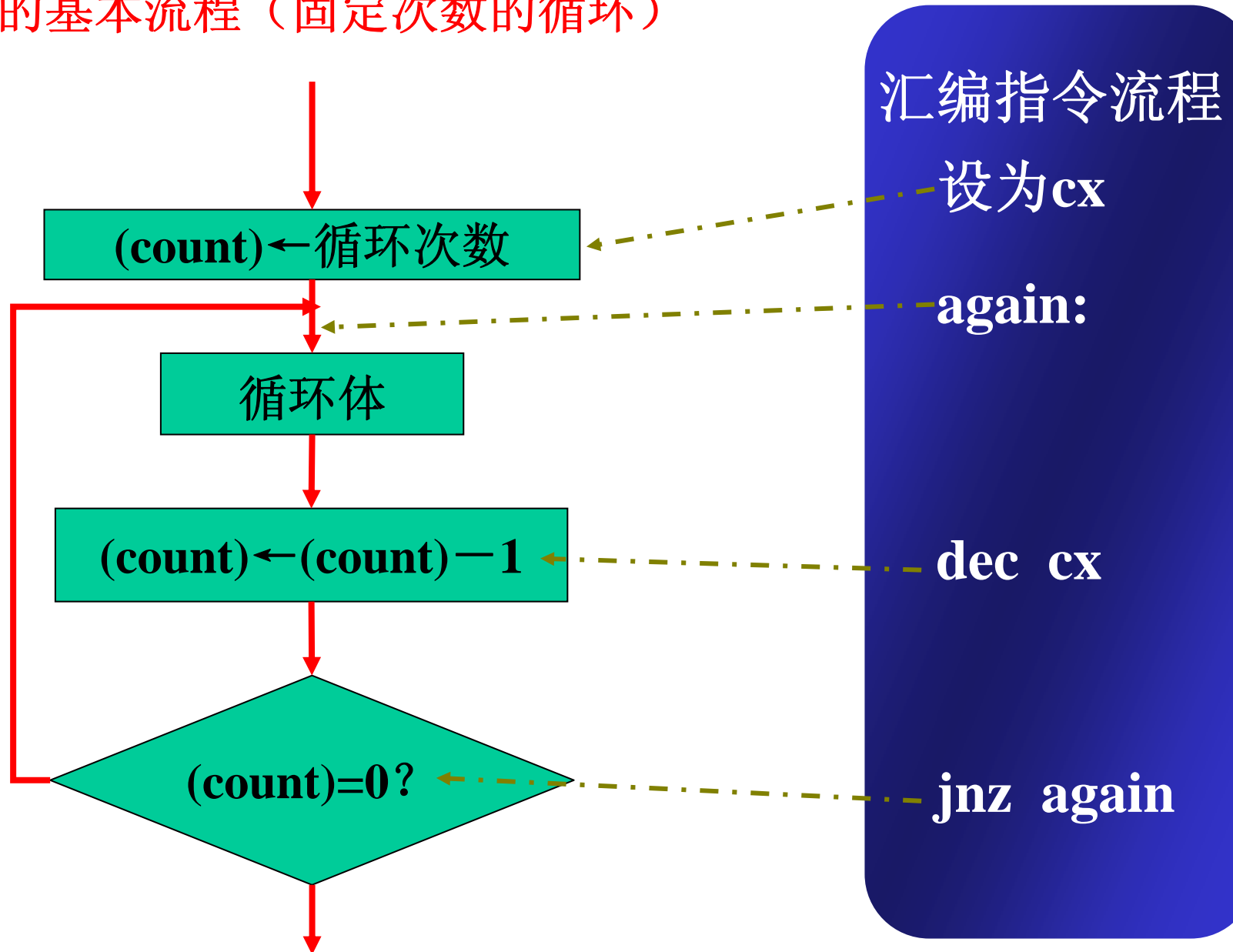
c. 循环(变量)的修正(表达式)

初值
中间态
终值

循环控制的目的:

使得循环控制变量由初值经过中间态到达终值

循环的基本流程（固定次数的循环）



循环程序常用到的指令: **LOOP**

LOOP的作用是什么?

例程

```
                MOV DI, M
AGAIN:  ....
                MOV CX, N
NEXT:    ....
                LOOP NEXT
                ....
                DEC DI
                JNZ AGAIN
```

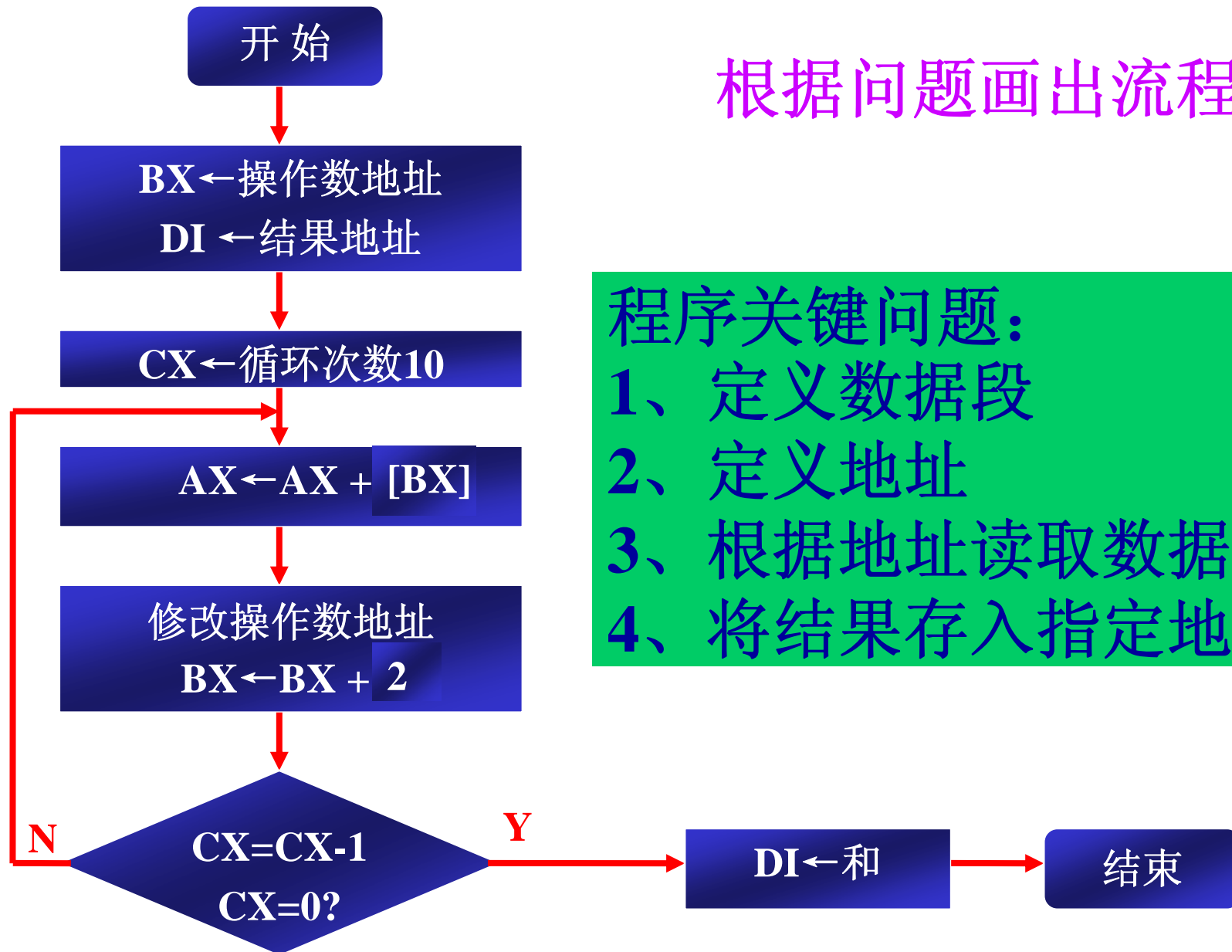
- (1)、 $(CX) = (CX) - 1$;
- (2)、如果 $(CX) \neq 0$,
 $(IP) = (IP) + 8$ 位位移。
 - 8位位移=“标号”处的地址
 - 当 $(CX) = 0$, 什么也不做
 (程序向下执行)

例题1: 计算 $y = \sum_{i=1}^{10} a_i$

分析:

- 1、由题意可知, 求 $a_1 + a_2 + \dots + a_n$ 的和
- 2、需要10条加法指令, 且数据有规律的存放, 可用循环来重复执行10次完成
- 3、题目没有给出具体数据, 可以自行定义为10个字型数据
- 4、需要解决数据的存放和读取问题, 没有10个寄存器保存10个数, 只能使用存储器方式存取
- 5、需要解决循环累加的程序代码设计问题

根据问题画出流程图



```

DATA    SEGMENT
    AA   DW  10H, 20H, 30H, ..., 100H
    BB   DW  ?
DATA    ENDS

CODE    SEGMENT
    ASSUME  CS:CODE, DS: DATA
START:
    MOV  AX, DATA
    MOV  DS, AX
    MOV  AX, 0
    MOV  CX, 10
    MOV  BX, OFFSET AA
    MOV  DI, OFFSET BB
LOP:
    ADD  AX, [BX]
    INC  BX
    INC  BX
    LOOP LOP

```

根据流程图写出代码

代码共计9个组成部分

```

    MOV  [DI], AX

    MOV  AH, 4CH
    INT  21H

CODE  ENDS

    END  START

```

循环程序例程

例2： 编制程序把BX寄存器的二进制数用十六进制的形式在屏幕上表示出来

例如： 设BX中的二进制数为

显示：

1010	0011	0101	1110
↓	↓	↓	↓
A	3	5	E

分析可知：

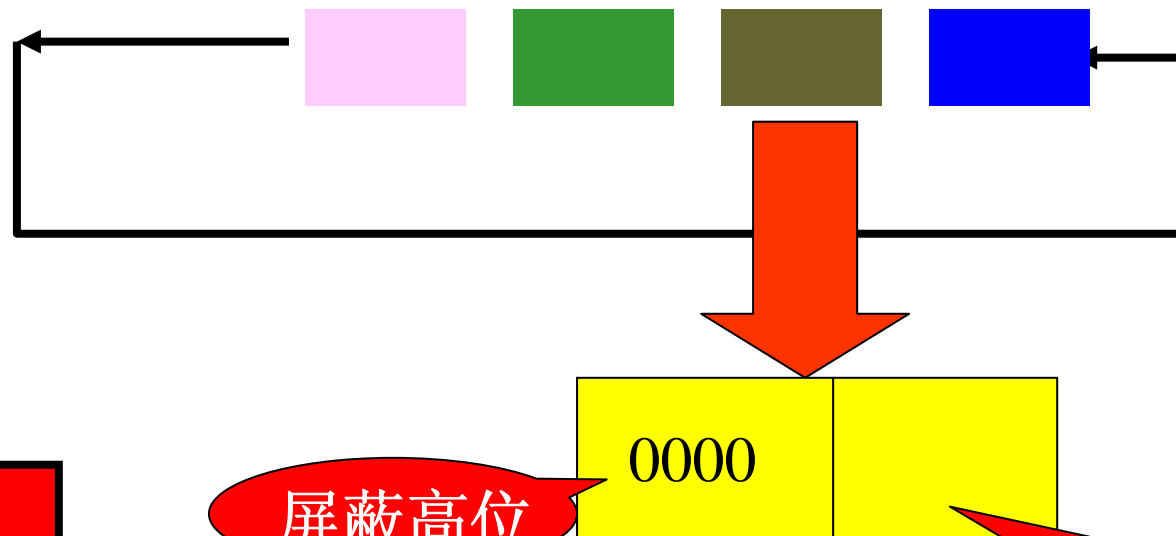
- 1、需要每4为显示输出一次
- 2、显示输出需要使用ASCII码

循环程序例程

例2：编制程序把BX寄存器的二进制数用十六进制的形式在屏幕上表示出来

分析题意：

高4位

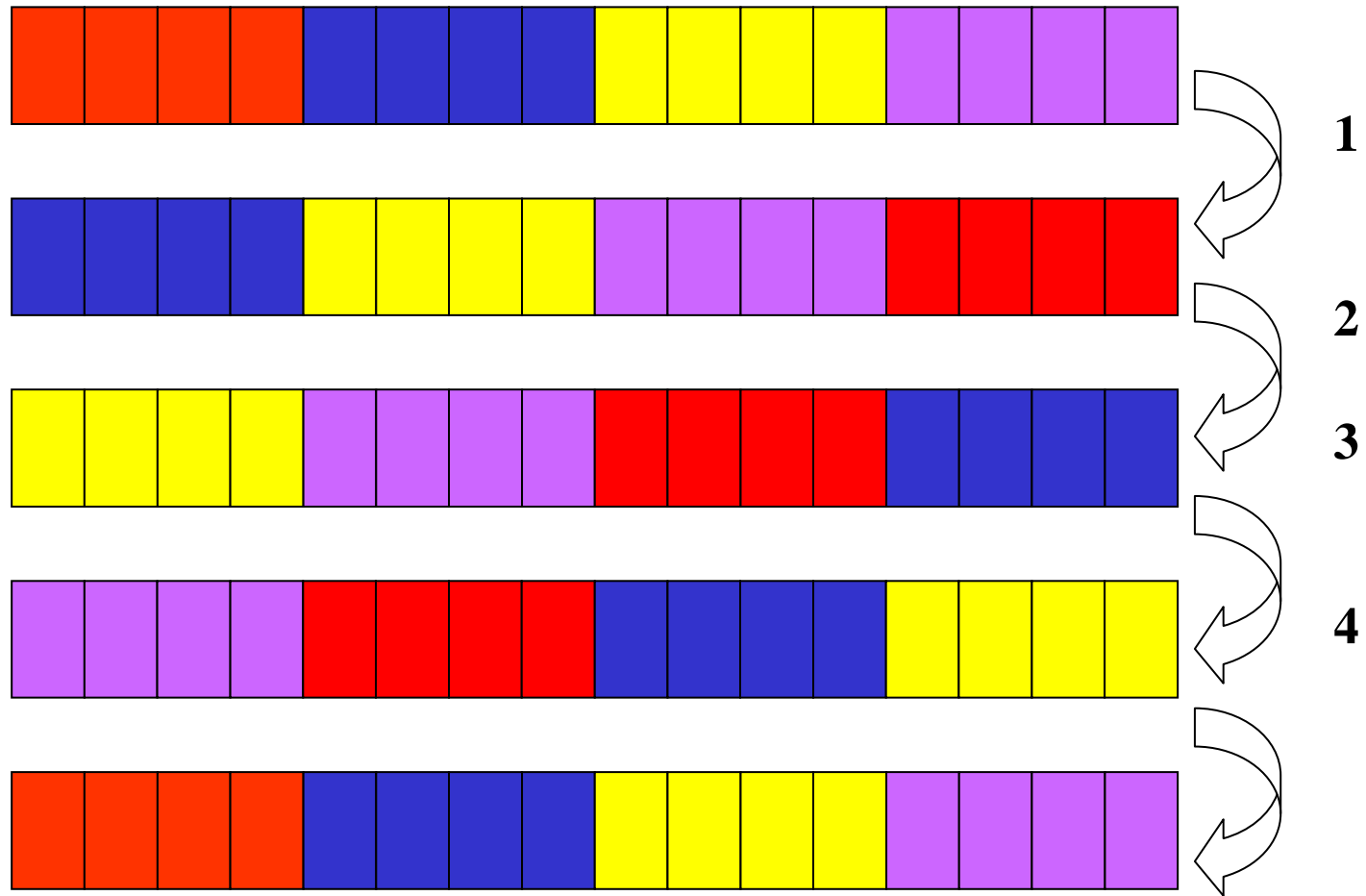


ASCII码

0-9: 30-39

A-F: 41-46

BX



回顾：逻辑指令—逻辑运算指令

把(BX)中的16位数每4位压入堆栈

MOV CH, 4 ; 循环次数

MOV CL, 4 ; 移位次数

NEXT: ROL BX, CL

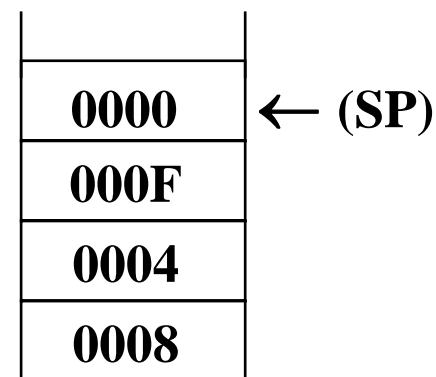
MOV AX, BX

AND AX, 0FH

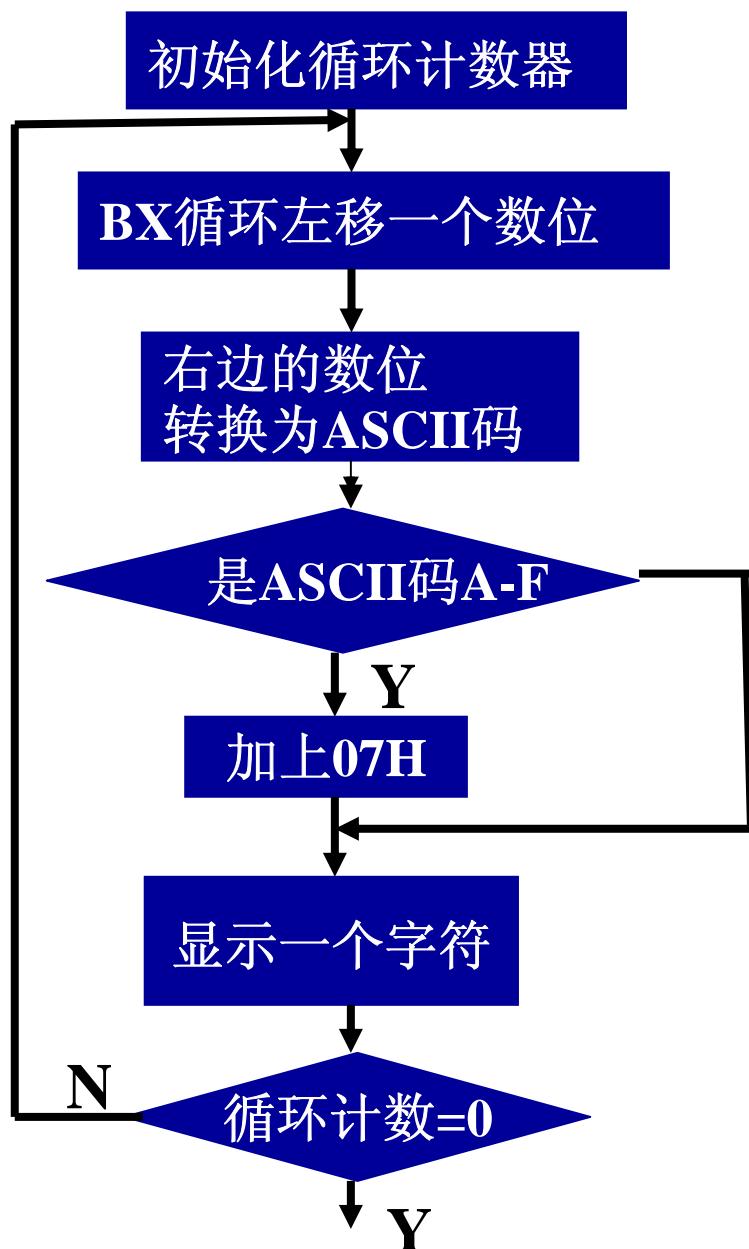
PUSH AX

DEC CH

JNZ NEXT



流程图



- 1、一个字单元有16位二进制数，即4位十六进制数
- 2、要按顺序显示必须把每4位二进制移至最右边，即分离出每个4位
- 3、对分离出的1位十六进制进行转换
- 5、需要特别注意：
0~9的ASCII码为30~39H
A~F的ASCII码为41~46H

```

DATA    SEGMENT
    AA    DW    3C6DH
DATA    ENDS

CODE    SEGMENT
    ASSUME    CS:CODE, DS; DATA
START:
    MOV    AX, DATA
    MOV    DS, AX
    LEA    SI, AA
    MOV    BX, [SI]
    MOV    CX, 10
    MOV    CH, 4
A1:
    MOV    CL, 4
    ROL    BX, CL
    MOV    AL, BL
    AND    AL, 0FH
    ADD    AL, 30H
    CMP    AL, 3AH

```

根据流程图写出代码

```

    JAL    A2
    ADD    AL, 7H
A2:
    MOV    DL, AL
    MOV    AH, 2
    INT    21H
    DEC    CH
    JNZ    A1

    MOV    AH, 4CH
    INT    21H

CODE    ENDS

    END    START

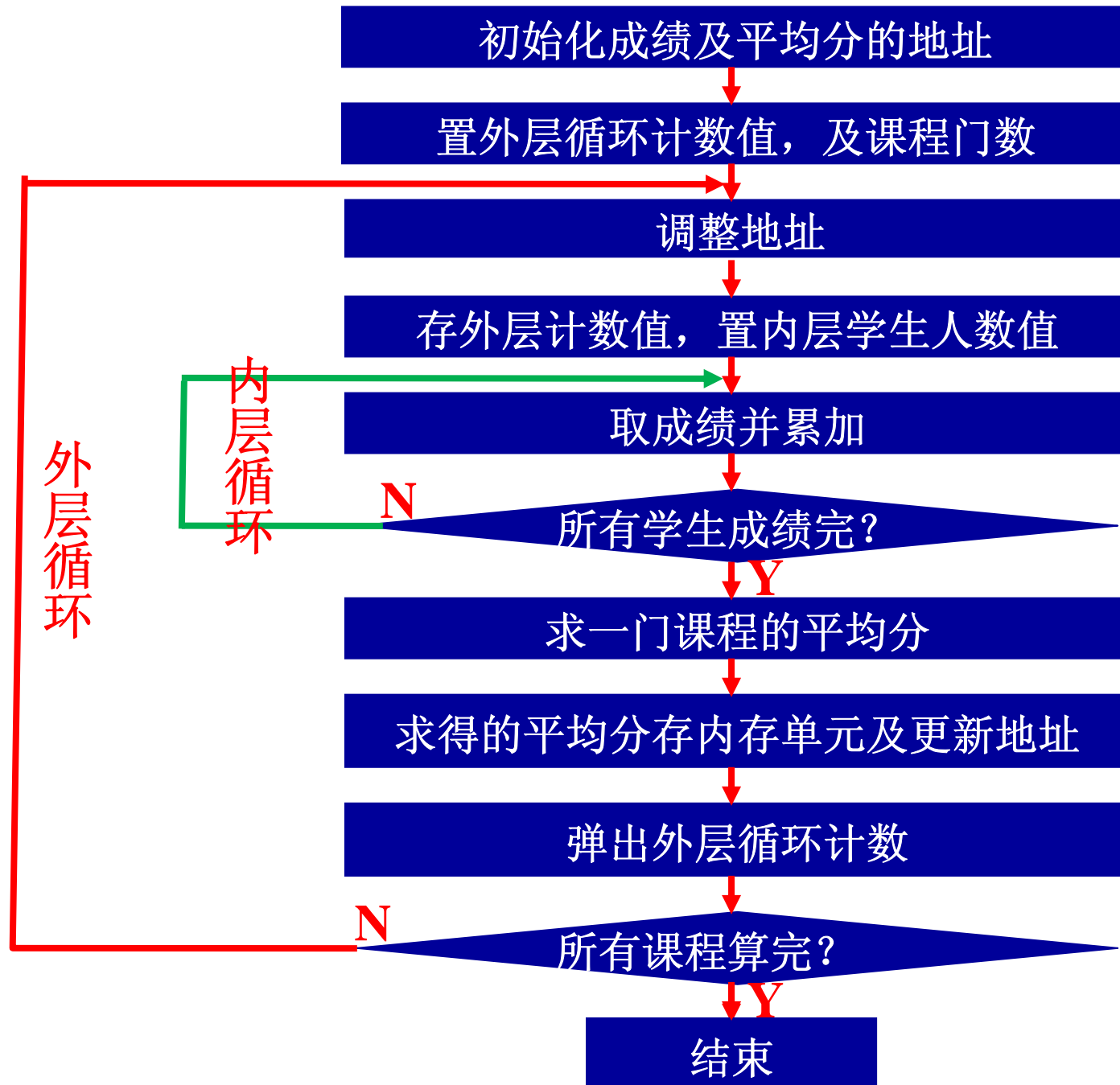
```


例题3：求学生的平均成绩 设5个学生、5门课程

问题分析：

- 1、5个学生5门课程定义一张成绩表
- 2、因为成绩采用百分制，所以数据类型为字节型
- 3、算出的每门课程的平均成绩存入字节型变量AA中
- 4、需要采用双重循环，外层控制课程门数。内层用于计算每门课程的累加分

流程图

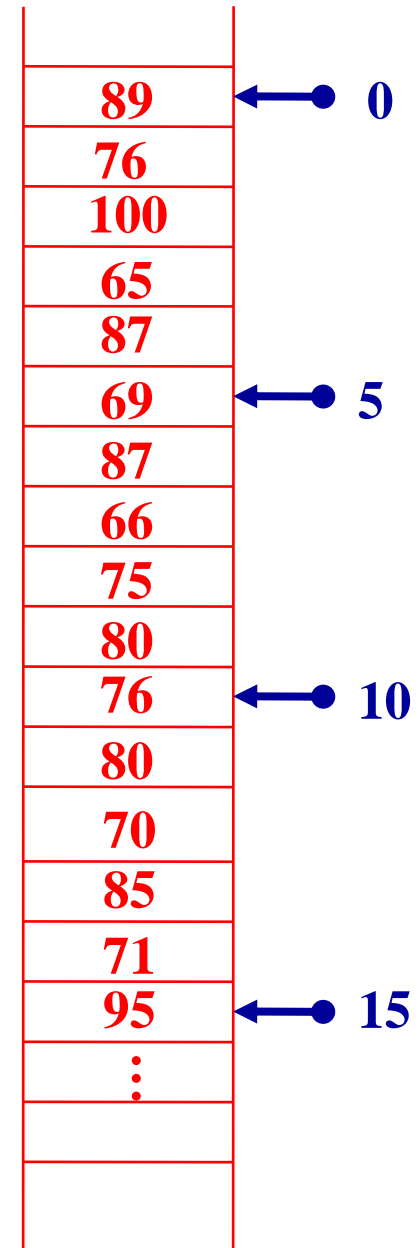


根据流程图写出代码

①定义数据段

```
DATA SEGMENT
    GRADE DB 89, 76, 100, 65, 87
           DB 69, 87, 66, 75, 80
           DB 76, 80, 70, 85, 71
           DB 95, 98, 96, 84, 90
           DB 80, 97, 72, 95, 88
    AA DB 5 DUP ( ? )
DATA ENDS
```

分析：
定义的数据段在内存中存储情况！



根据流程图写出代码

```
DATA SEGMENT
    GRADE DB 89, 76, 100, 65, 87
           DB 69, 87, 66, 75, 80
           DB 76, 80, 70, 85, 71
           DB 95, 98, 96, 84, 90
           DB 80, 97, 72, 95, 88
    AA     DB 5 DUP ( ? )
DATA ENDS

CODE SEGMENT
    ASSUME CS:CODE, DS; DATA
START:
    MOV AX, DATA
    MOV DS, AX

    MOV BX, OFFSET GRADE
    MOV DI, OFFSET AA
    MOV SI, BX
```

①定义数据段

②关联相关段

③取数据段物理地址

④取数据段偏移地址

```
MOV CX, 5 ;外层循环计数值，即课程门数
A1: MOV BX, SI
    SUB BX, 5
    PUSH CX
    MOV AL, 0
    MOV CX, 5 ;内层循环计数值，即学生人数
    A2: ADD BX, 5 ;同一门课程地址更新
        ADD AL, [BX]
        LOOP A2
```

⑤外循环

⑥内循环

```
MOV    AH, 0
MOV    DL, 5    ;人数赋给DL
DIV    DL        ;求一门课程的平均成绩
MOV    [DI], AL    ;存平均分

INC    SI
INC    DI
POP    CX    ;弹出外层循环计数值
LOOP   A1

MOV    AH, 4CH
INT    21H

CODE   ENDS

END    START
```

⑦求平均

⑧外循环控制

⑨程序返回

⑩程序结束

分析理解完整代码

DATA SEGMENT

```
GRADE DB 89, 76, 100, 65, 87
      DB 69, 87, 66, 75, 80
      DB 76, 80, 70, 85, 71
      DB 95, 98, 96, 84, 90
      DB 80, 97, 72, 95, 88
AA     DB 5 DUP ( ? )
```

DATA ENDS

CODE SEGMENT

```
ASSUME CS:CODE, DS; DATA
```

START:

```
MOV AX, DATA
MOV DS, AX
```

```
MOV BX, OFFSET GRADE
MOV DI, OFFSET AA
MOV SI, BX
```

```
MOV CX, 5 ;外层循环计数值，即课程门数
```

A1:

```
MOV BX, SI
SUB BX, 5
PUSH CX
MOV AL, 0
```

```
MOV CX, 5 ;内层循环计数值，即学生人数
A2:
ADD BX, 5 ;同一门课程地址更新
ADD AL, [BX]
LOOP A2
```

```
MOV AH, 0
MOV DL, 5 ;人数赋给DL
DIV DL ;求一门课程的平均成绩
MOV [DI], AL ;存平均分
```

```
INC SI
INC DI
POP CX ;弹出外层循环计数值
LOOP A1
```

```
MOV AH, 4CH
INT 21H
```

CODE ENDS

END START