



## 汇编语言程序设计

### 4.0 概述

### 4.1 汇编语言程序格式

### 4.2 MASM中的体现式

### 4.3 伪指令语句

### 4.4 程序设计措施

### 4.5 DOS系统功能调用

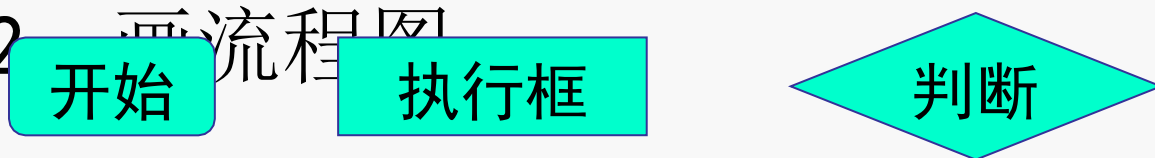


# 4.4 程序设计措施

- 一、汇编语言源程序设计基本环节

- 1、明确任务，拟定算法。

- 2、画流程图



- 3、分配存储空间及工作单元

分析需要定义的段、变量、变量类型、长度；规划各寄存器的使用等

- 4、编写源程序(.ASM)、汇编(.OBJ)、连接



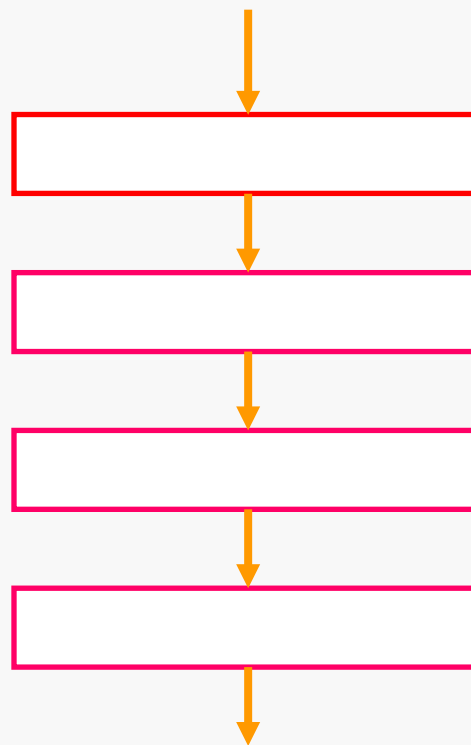
- 二、基本程序构造
- 顺序构造
- 分支构造
- 循环构造
- 子程序构造

```
DATA    SEGMENT
.....
DATA    ENDS
CODE    SEGMENT
ASSUME  CS:CODE, DS:DATA
START:  MOV  AX, DATA
        MOV  DS, AX
        .....
        MOV  AH, 4CH
        INT  21H
CODE    ENDS
        END  START
```



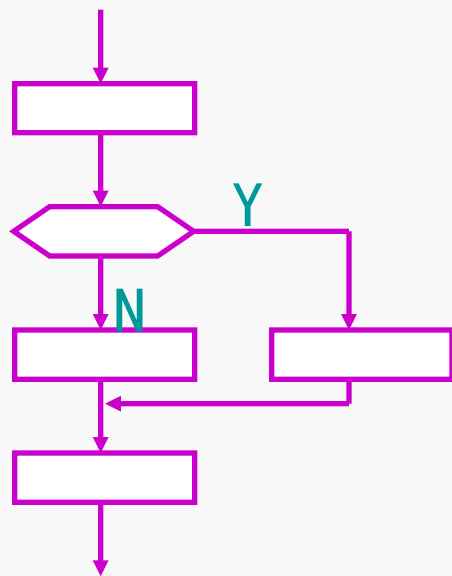
- 顺序构造
- 分支构造
- 循环构造
- 子程序构造

顺序构造

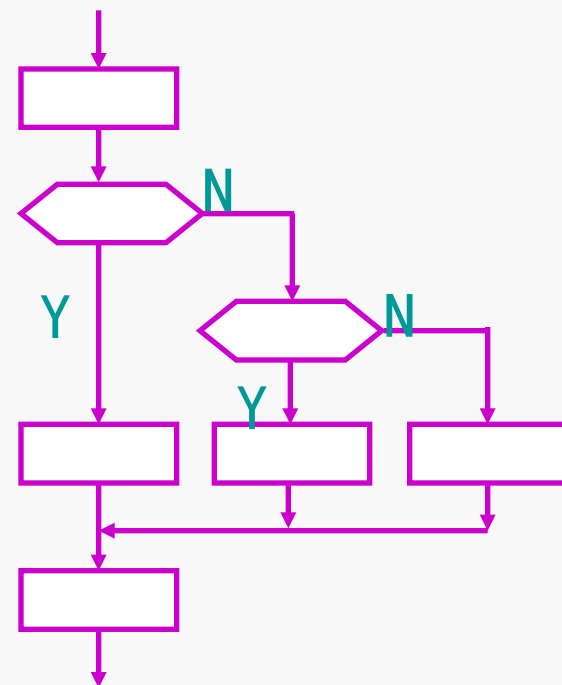




- 顺序构造
- 分支构造
- 循环构造
- 子程序构造



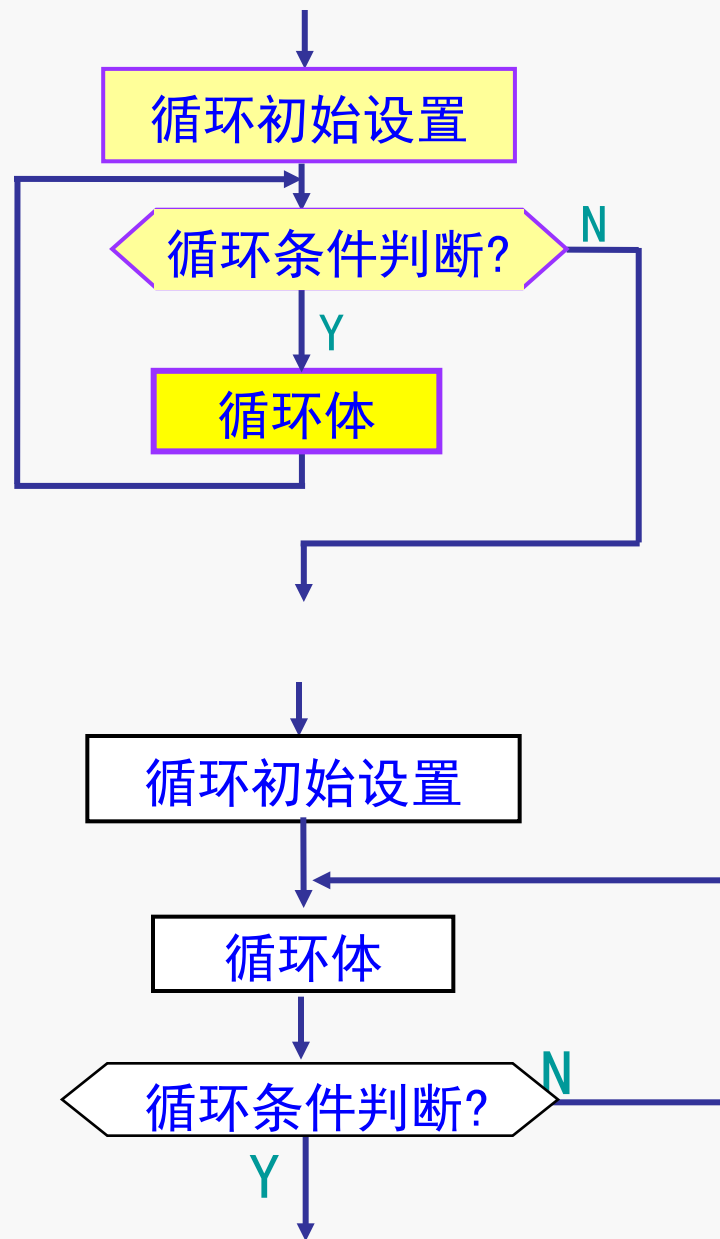
两个分支



三个分支



- 顺序构造
- 分支构造
- 循环构造
- 子程序构造





- 顺序构造
- 分支构造
- 循环构造
- 子程序构造

```
code SEGMENT
    ...
begin: ...
    CALL sub1
    ...
    CALL sub2
    ...
    MOV AH, 4CH
    INT 21H
-----
sub1 PROC
    ...
    RET
sub1 ENDP
-----
sub2 PROC
    ...
    RET
sub2 ENDP
-----
code ENDS
    END begin
```

- 1、顺序构造

- 例1：已知目前数据段中有一种十进制数字0~9的七段代码表，其数值依次为40H、79H、24H、30H、19H、12H、02H、78H、00H、18H。要求用XLAT指令将十进制数57转换成相应的七段代码值，存到BX寄存器中，编写完整的汇编源程序。

- 分析：先求5的七段代码值，再求数字7的；

- 需定义数据段、七段代码表：TAB，字变





## 源程序清单：

DATA SEGMENT

```
TAB DB 40H, 79H, 24H, 30H
     DB 19H, 12H, 02H, 78H
     DB 00H, 18H
```

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE, DS:DATA

```
START: MOV AX, DATA
        MOV DS, AX
        (转右)
```

```
MOV BX, OFFSET TAB
```

```
MOV AL, 5
```

```
XLAT
```

```
MOV AH, AL
```

```
MOV AL, 7
```

```
XLAT
```

```
MOV BX, AX
```

```
MOV AH, 4CH ;返回DOS
```

```
INT 21H
```

```
CODE ENDS
```

```
END START
```

- 2、分支构造

- 程序根据不同条件选择不同处理措施

例2: 求内存中数X的绝对值, 并送回原处

1. 分析:

$$|X| = \begin{cases} X & X \geq 0 \\ 0 - X & X < 0 \end{cases}$$

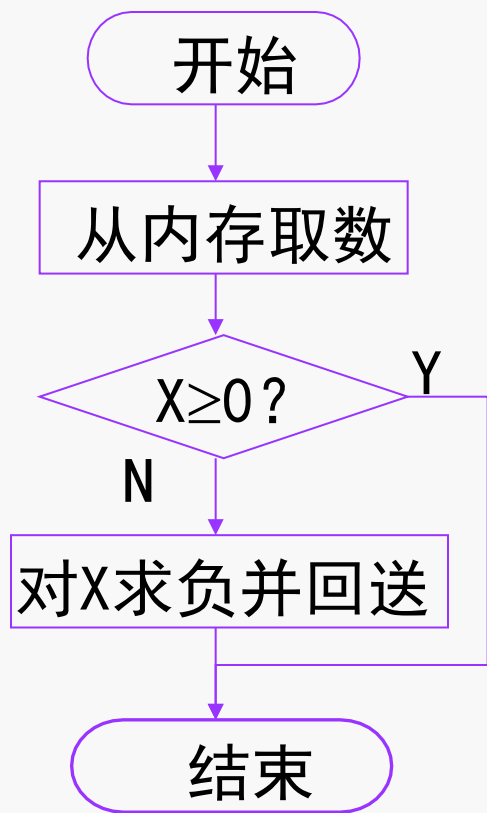
2. 根据以上分析画出流程框图

3. 根据流程图拟定所用寄存器, 并编写源程序;

需定义数据段、变量X、使用寄存器AL



## 程序框图及清单：



DATA SEGMENT

X DB ?

DATA ENDS

CODE SEGMENT

ASSUME CS: CODE, DS: DATA

```

START: MOV AX, DATA } ;将数据段段
      MOV DS, AX    } 地址送入DS
  
```

MOV AL, [X]

CMP AL, 00H ;与0比较

JGE RETURN

NEG AL ;求负

MOV [X], AL ;回送成果

RETURN: MOV AH, 4CH ;返回DOS

INT 21H

CODE ENDS

END START ;结束源程序

- 3、循环构造

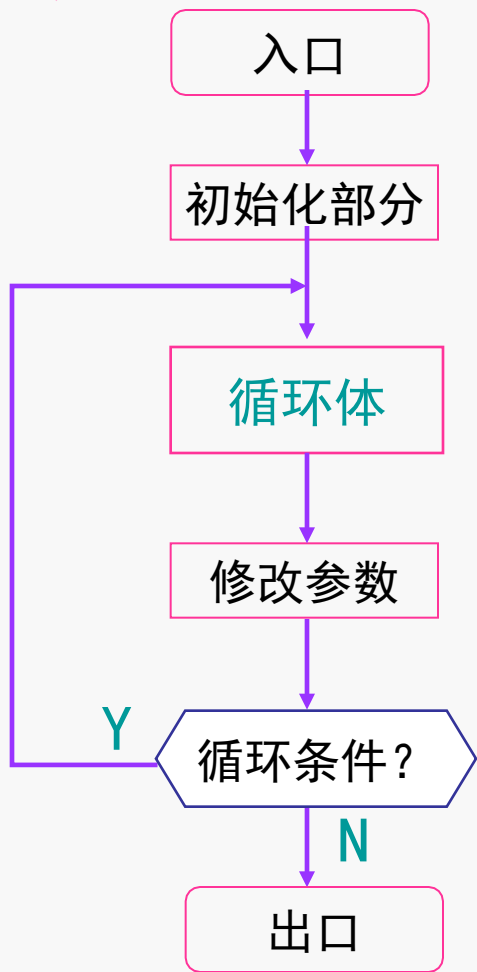
- 两种形式：

- 1) “先执行，后判断”：进入循环后至少要执行一次循环体，再判断循环是否结束。

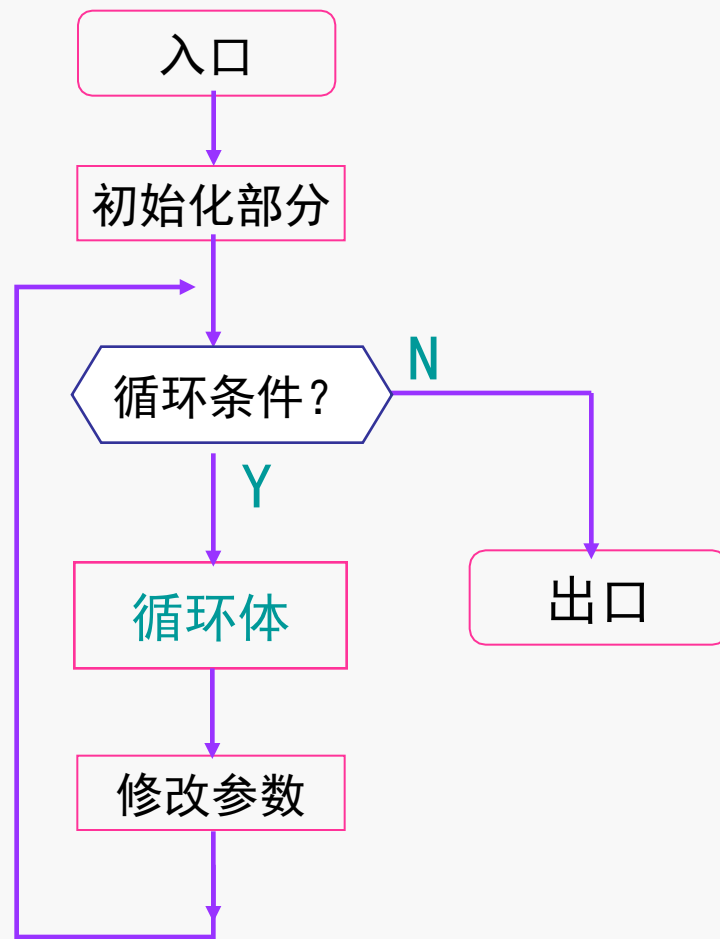
- ---用于循环次数固定的程序（简朴）  
（掌握）

- 2) “先判断，后执行”：进入循环后，先

判断循环结束条件，<sup>12</sup>再决定是否执行循环



(a) “先执行，后判断” 构造



(b) “先判断，后执行” 构造



- 两种循环构造都涉及四部分：
- 1) 初始化---设置计数值、地址指针
- 2) 循环体
- ---循环部分的关键：循环的全部执行指令
- 3) 修改参数---改地址指针，为下次循环做准备
- 4) 循环控制---修改计数值，进行判断



例3：将附加段起始地址为DA1的100个字节的存储区内容清0。

DATA SEGMENT

DA1 DB 100 DUP (?)

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE, ES:DATA

START: MOV AX, DATA

MOV ES, AX

(转右边)

MOV AL, 00H

MOV CX, 0064H

LEA BX, DA1 ; 初始化

LP: MOV ES: [BX], AL ; 循环体

INC BX ; 修改参数

LOOP LP ; 循环控制

MOV AH, 4CH ; 返回DOS

INT 21H

CODE ENDS

END START



- 例4：编程在一串给定个数的数据区BUFF中寻找最小数，成果存储至MIN单元中！（循环+分支构造）

```
DATA SEGMENT
BUFF DB 42H, 56, 34, -5, -9, -78
CONT EQU $-BUFF
MIN DB 00H
DATA ENDS
CODE SEGMENT
ASSUME CS: CODE, DS: DATA
START: MOV AX, DATA
MOV DS, AX
```

(转下页)





```
        LEA  BX,  BUFF
        MOV  CX,  CONT
        MOV  AL,  [BX]      ; 初始化
LP:     CMP  AL,  [BX]      ; 循环体
        JLE  NEXT
        MOV  AL,  [BX]
NEXT:   INC  BX            ; 修改参数
        LOOP LP           ; 循环控制
        MOV  MIN,  AL

        MOV  AH,  4CH
        INT  21H

CODE   ENDS
        END  START
```



# 作业

- 补充
- 在BLOCK开始有一种字节型数组，共有15个元素，利用汇编语言编写完整的程序，计算这个数组中奇数的个数，并将成果存入内存SUM单元。
-



- **例5:** 数据段从**BUFF**开始有一字数组，存储**N**个有符号数据，编程将数据按从小到大排序！

(双重循环)

```
DATA SEGMENT
BUFF DW 42H, 56, 34, -5, -9, -78
N EQU ($-BUFF)/2
DATA ENDS

CODE SEGMENT
ASSUME CS: CODE, DS: DATA

START: MOV AX, DATA
MOV DS, AX
```

(转下页)

```

MOV  CX, N-1
LPW: MOV  BX, 00H
MOV  DX, CX      ; 初始化
LPN: MOV  AX, BUFF[BX] ; 循环体
CMP  AX, BUFF[BX+2]
JLE  NEXT
XCHG AX, BUFF[BX+2]
MOV  BUFF[BX], AX
NEXT: ADD  BX, 2      ; 修改参数
DEC  DX              ; 循环控制
JNZ  LPN
LOOP LPW             ; 循环控制

```

(转右边)

```

MOV  AH, 4CH
INT  21H
CODE  ENDS
      END  START

```

**思索：**

- 1、若为无符号数？
- 2、若要求原数据不变？

- 4、子程序构造
- 子程序：能完毕特定功能而又经常用到的程序段，编写成独立的模块，供其他程序调用，称为子程序或过程。
- 子程序构造的定义：
  - 使用过程定义语句**PROC...ENDP**。
- 子程序属性：
  - 调用程序和过程在同一代码段中，则用

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/558017060130006132>