



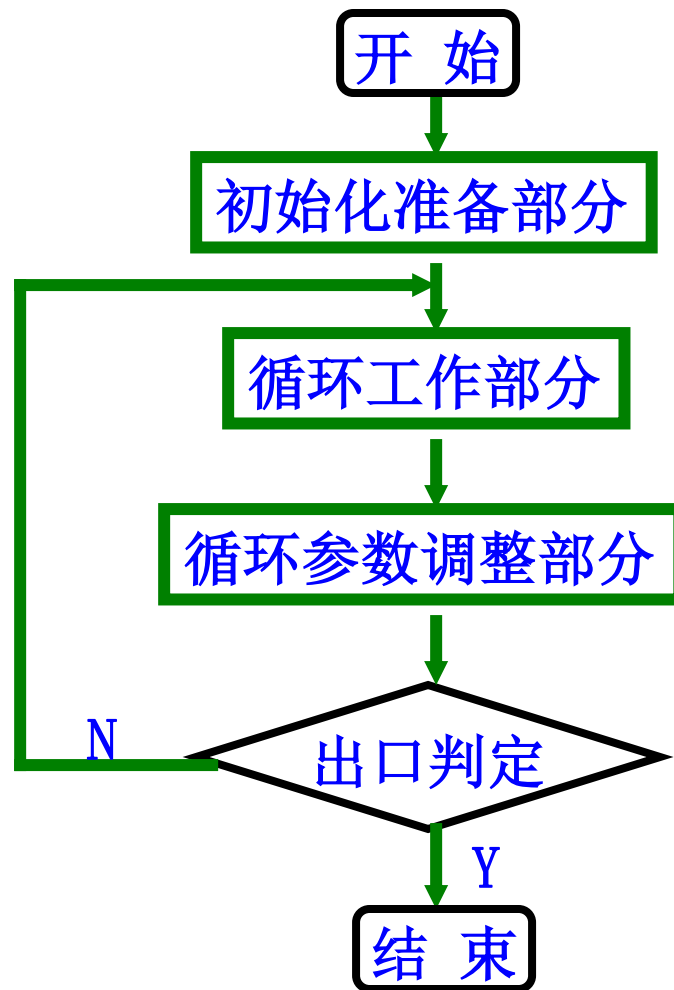
4.3 循环程序设计技术

1. 循环程序的基本结构

循环程序有以下两种结构。

(A) 先循环工作，后循环判定 (DO-UNTIL结构)

- 应用场合：适合于循环次数已知的情況；
- 循环判定条件：用CX 作减1 计数控制循环

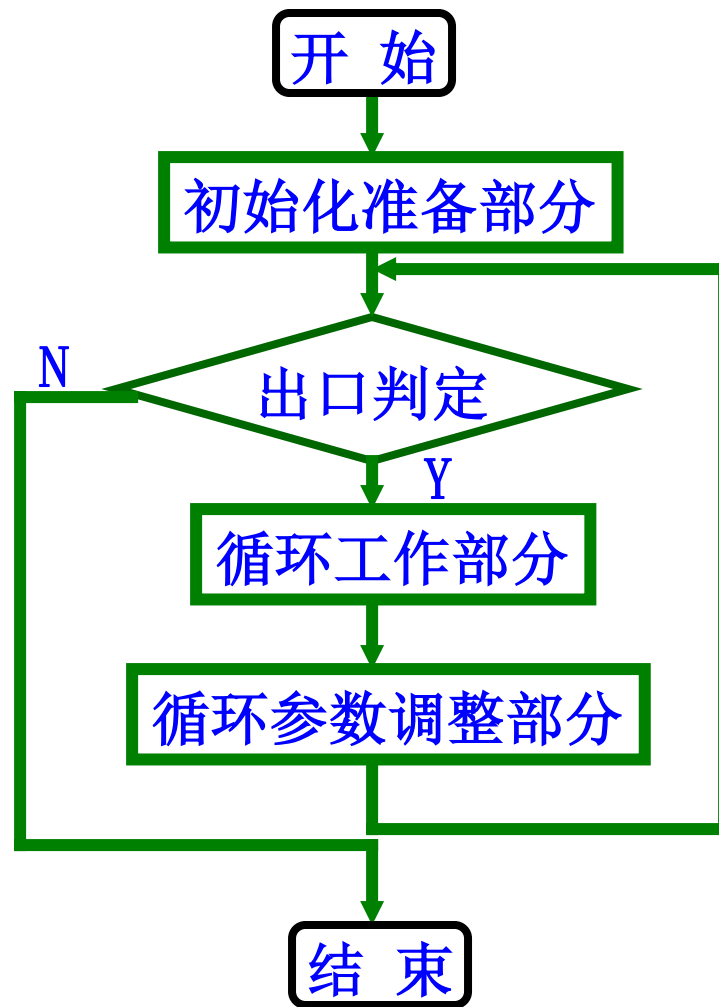




4.3 循环程序设计技术

(B) 先循环判定，后循环工作
(DO-WHILE结构)

- 应用场合： 适合于循环次数未知或可能会出现0次循环的情况
- 循环判定条件： 用条件标志





4.3 循环程序设计技术

2. 循环程序的组成

不管是哪一种循环结构，一个标准的循环程序应由以下四部分组成：

- 初始化准备部分
- 循环工作部分
- 参数调整部分
- 出口判定部分



4.3 循环程序设计技术

3. 循环控制指令

为了便于循环控制，8086/8088CPU 专门设置了一类循环控制类指令，其指令书写格式和指令完成的功能如下表所示。

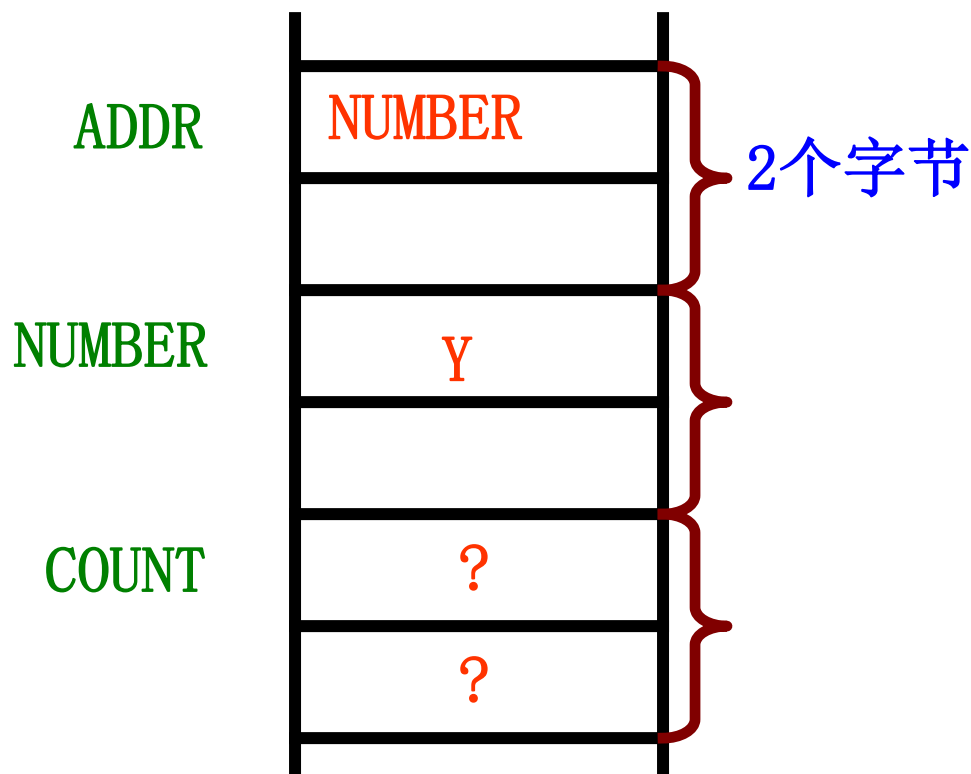
助记符格式	功能说明
LOOP 标号	$(CX) \leftarrow (CX) - 1$, $(CX) \neq 0$ 时, 转到标号处
LOOPZ/LOOPE 标号	$(CX) \leftarrow (CX) - 1$, $(CX) \neq 0$ 且 $ZF=1$ 时, 转到标号处
LOOPNZ/LOOPNE 标号	$(CX) \leftarrow (CX) - 1$, $(CX) \neq 0$ 且 $ZF=0$ 时, 转到标号处
JCXZ 标号	$CX=0$ 时, 转到标号处



4.3 循环程序设计技术

例：在存储器ADDR单元中存放着数Y的地址，试编制一程序统计数Y中的1的个数，并将结果存入COUNT单元中。

思路：





4.3 循环程序设计技术

- 要测出数Y中1的个数，一个比较简单的方法就是逐位测试，把数Y用移位的方法，逐次移到CF中去，根据CF是否为1来计数，共循环移位计数16次。
- 在很多情况下，数Y可能一开始就为0，或在循环移位计数的过程中，在16次循环未进行完之前，已经变为0，那么用上述循环移位的逐位测试法，势必使程序执行效率降低。为提高程序的效率，在这类程序设计中，常采用先循环判断后循环工作的循环结构。
- 循环结束控制条件为：ZF=1。



4.3 循环程序设计技术

程序:

```
DATA    SEGMENT
```

```
    ADDR        DW    NUMBER
```

```
    NUMBER      DW    Y
```

```
    COUNT       DW    ?
```

```
DATA    ENDS
```

```
CODE    SEGMENT
```

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

```
START:
```

```
    MOV     AX, DATA
```

```
    MOV     DS, AX
```

```
    MOV     BX, ADDR
```



4.3 循环程序设计技术

```
        MOV     AX, [BX]
        MOV     DX, 0
LOP:
        CMP     AX, 0
        JZ      EXIT
        SHL     AX, 1
        JNC     LOP
        INC     DX
        JMP     LOP
EXIT:
        MOV     COUNT, DX
        MOV     AH, 4CH
        INT     21H
CODE    ENDS
        END     START
```