



13.4 汇编语言程序设计基础

1 顺序程序设计

例， 利用学号查学生的数学成绩表。

```
DATA    SEGMENT
```

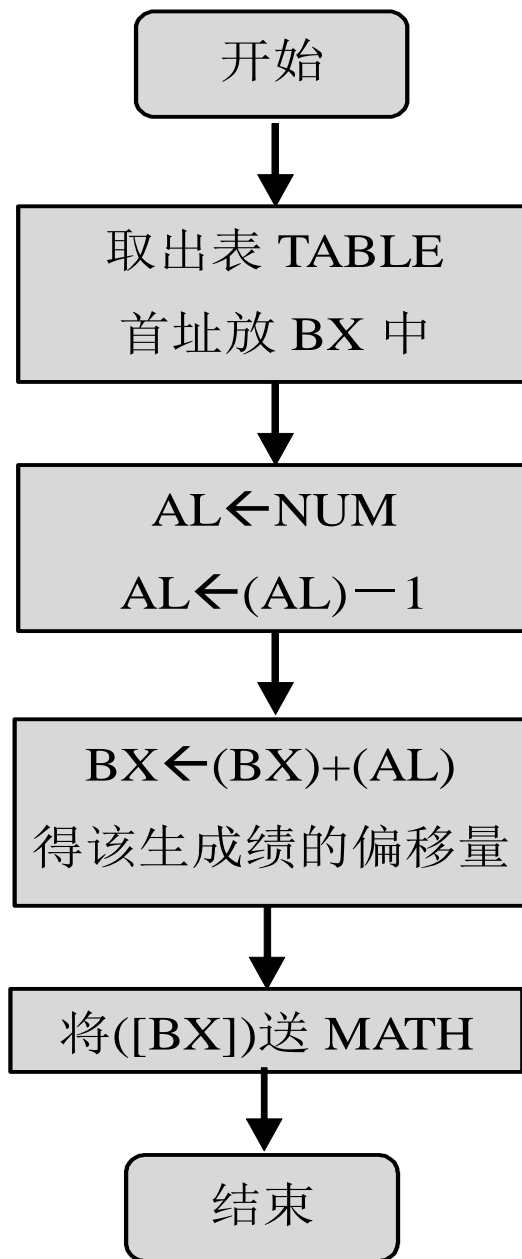
```
TABLE   DB    81, 78, 90, 64, 85, 76, 93, 82, 57, 80
```

```
         DB    73, 62, 87, 77, 74, 86, 95, 91, 82, 71
```

```
NUM      DB    8
```

```
MATH     DB    ?
```

```
DATA     ENDS
```





COSEG SEGMENT

ASSUME CS:COSEG, DS:DATA

START: MOV AX, DATA ;

MOV DS, AX ;装入DS

MOV BX, OFFSET TABLE ;BX指向表首址

MOV AH, 0

MOV AL, NUM

DEC AL ;实际学号是从1开始的

ADD BX, AX ;BX加上学号指向要查的成绩

MOV AL, [BX] ;查到成绩送AL

MOV MATH, AL ;存结果

MOV AH, 4CH ;返回DOS

INT 21H

COSEG ENDS

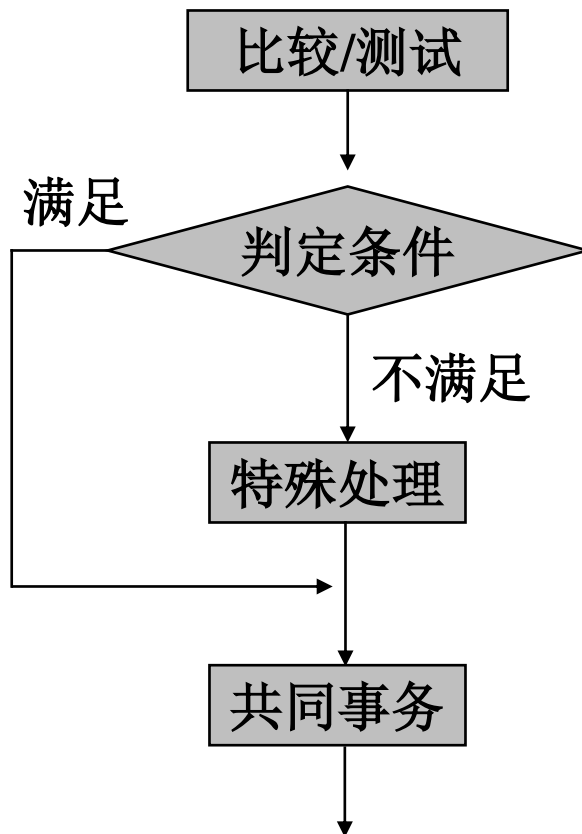
END START



2 分支程序设计

1、用比较/测试指令+条件转移指令实现分支

(1) 单分支结构





例，计算字单元**Dmem**中带符号数的绝对值，并将结果存储于数据段**RESULT**中。

```
DATA SEGMENT
```

```
    DMem    DW    -5
```

```
    RESULT  DW    0
```

```
DATA END
```

```
STACK1 SEGMENT PARA STACK
```

```
    DW 20H DUP(0)
```

```
STACK1 ENDS
```

共同事务：将数的绝对值存储到数据段

特殊处理：负数需要求补



```
CODE  SEGMENT
```

```
    ASSUME CS:CODE,DS:DATA,SS:STACK1
```

```
BEGIN:
```

```
    MOV  AX,  DATA
```

```
    MOV  DS,  AX
```

```
    MOV  AX, DMem
```

```
    CMP  AX,  0
```

```
    JGE  NONEG
```

```
    NEG  AX
```

```
NONEG: MOV  RESULT, AX
```

```
    MOV  AH,4CH
```

```
    INT  21H
```

```
CODE  ENDS
```

```
    END  BEGIN
```



需要注意的是，条件满足时转移，否则执行下一条语句。

```
CMP  AX, 0
```

```
JL   YESNEG
```

```
JMP  NONEG
```

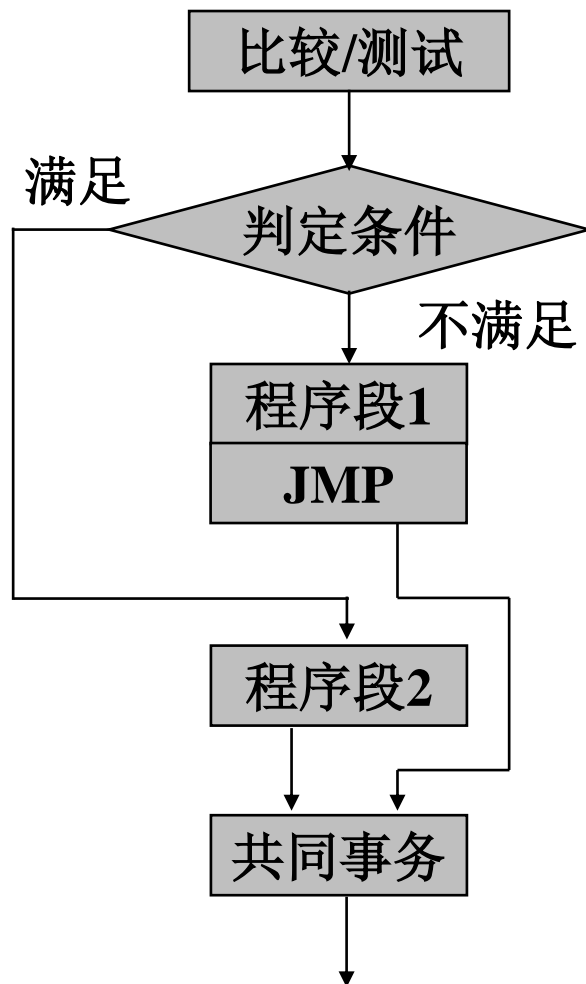
```
YESNEG: NEG  AX
```

```
NONEG:  MOV  RESULT, AX
```

因此，需要合理选择分支条件，同时理解与高级语言的差别。



(2) 双分支结构





例，编写一个能够显示BX二进制最高位的程序段。

```

                                ⋮
                                SHL    BX,    1
                                JC      ONE
                                MOV     DL,    '0'
                                JMP     DISPLAY
ONE:                            MOV     DL,    '1'
DISPLAY:                       MOV     AH,    2
                                INT      21H
                                ⋮
```



2、采用跳转表实现多路分支

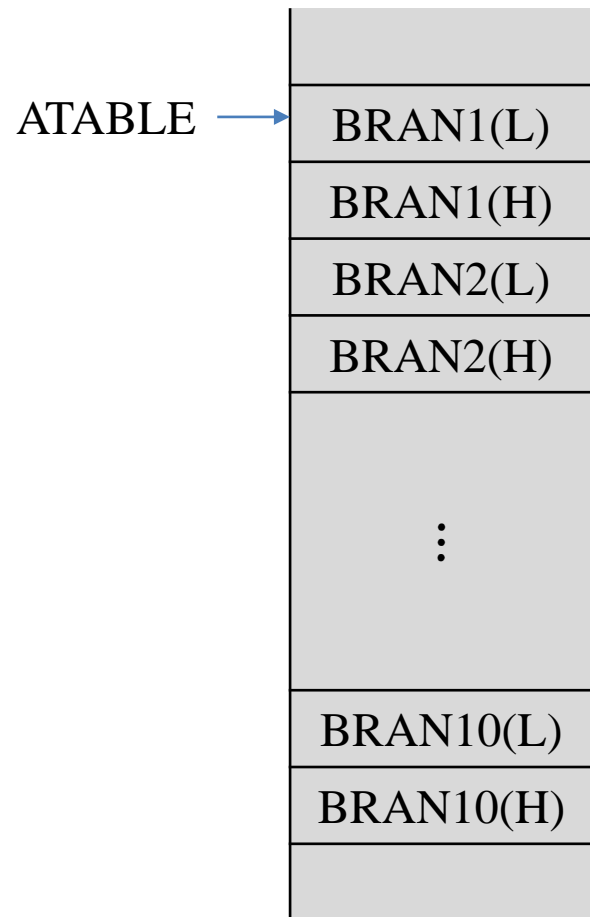
例，设某程序有10路分支，试根据变量N的值（1~10），将程序转移到其中的一路分支去。

设10路分支程序段的入口地址分别为：
BRAN1、BRAN2.....BRAN10。

当变量N为1时，转移到**BRAN1**；N为2时，转移到**BRAN2**，依次类推。

在跳转表中每两个字节存放一个入口地址的偏移量，如右图所示。

程序中，先根据N的值形成查表地址：
 $(N-1) \times 2 + \text{表首址}$ 。





DATA SEGMENT

 ATABLE DW BRAN1,BRAN2,BRAN3,...,BRAN10

 N DB 3

DATA ENDS

STACK1 SEGMENT PARA STACK

 DW 20H DUP (0)

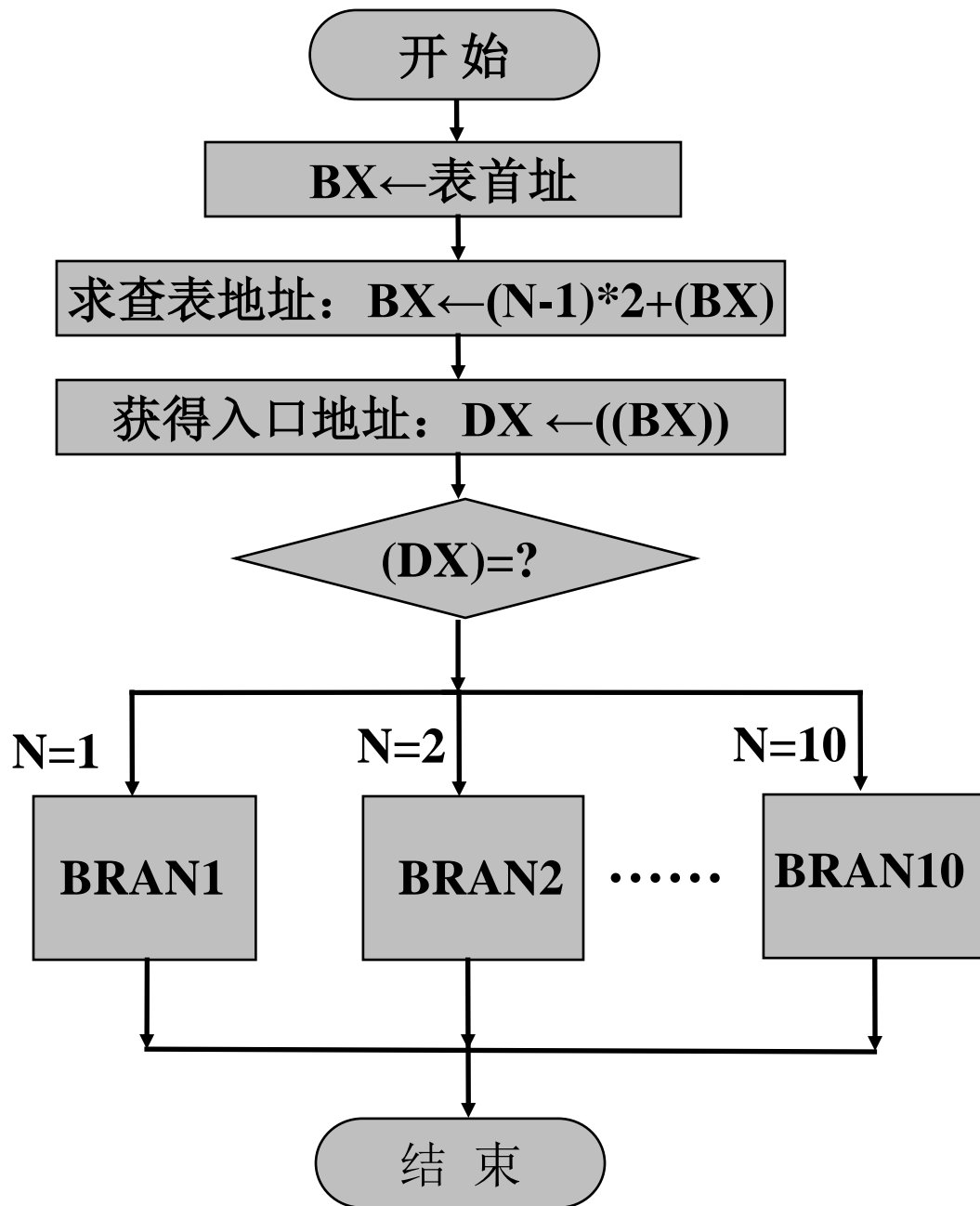
STACK1 ENDS

CODE SEGMENT

 ASSUME CS:CODE, DS:DATA, SS:STACK1

START: MOV AX, DATA

 MOV DS, AX



多路分支结构流程图



MOV AH, 0

MOV AL, N

DEC AL

SHL AL, 1 ; $2^*(N-1)$

MOV BX, OFFSET ATABLE ;BX指向表首址

ADD BX, AX ;BX指向查表地址

MOV DX, [BX] ;将N对应的分支入口地址送到CX中

JMP DX ;转移到N对应的分支入口地址



```
BRAN1:  ⋮  
        JMP    END1  
BRAN2:  ⋮  
        JMP    END1  
BRAN3:  ⋮  
        JMP    END1  
  
BRAN10: ⋮  
  
END1:   ⋮  
        MOV    AH, 4CH  
        INT    21H  
CODE    ENDS  
        END    START
```



3 循环程序设计

1、计数循环程序设计

例，计算1~100数字之和，并将结果存入字变量SUM的程序段。

	MOV	AX, 0	← 初始化部分
	MOV	CX, 100	
AGAIN:	ADD	AX, CX	← 循环体
	LOOP	AGAIN	← 控制条件
	MOV	SUM, AX	← 结果处理



2、条件判断循环程序设计

例， 数据段的ARY字节数组中存放有10个无符号数， 试找出其中最大者送入MAX字节单元。

DATA SEGMENT

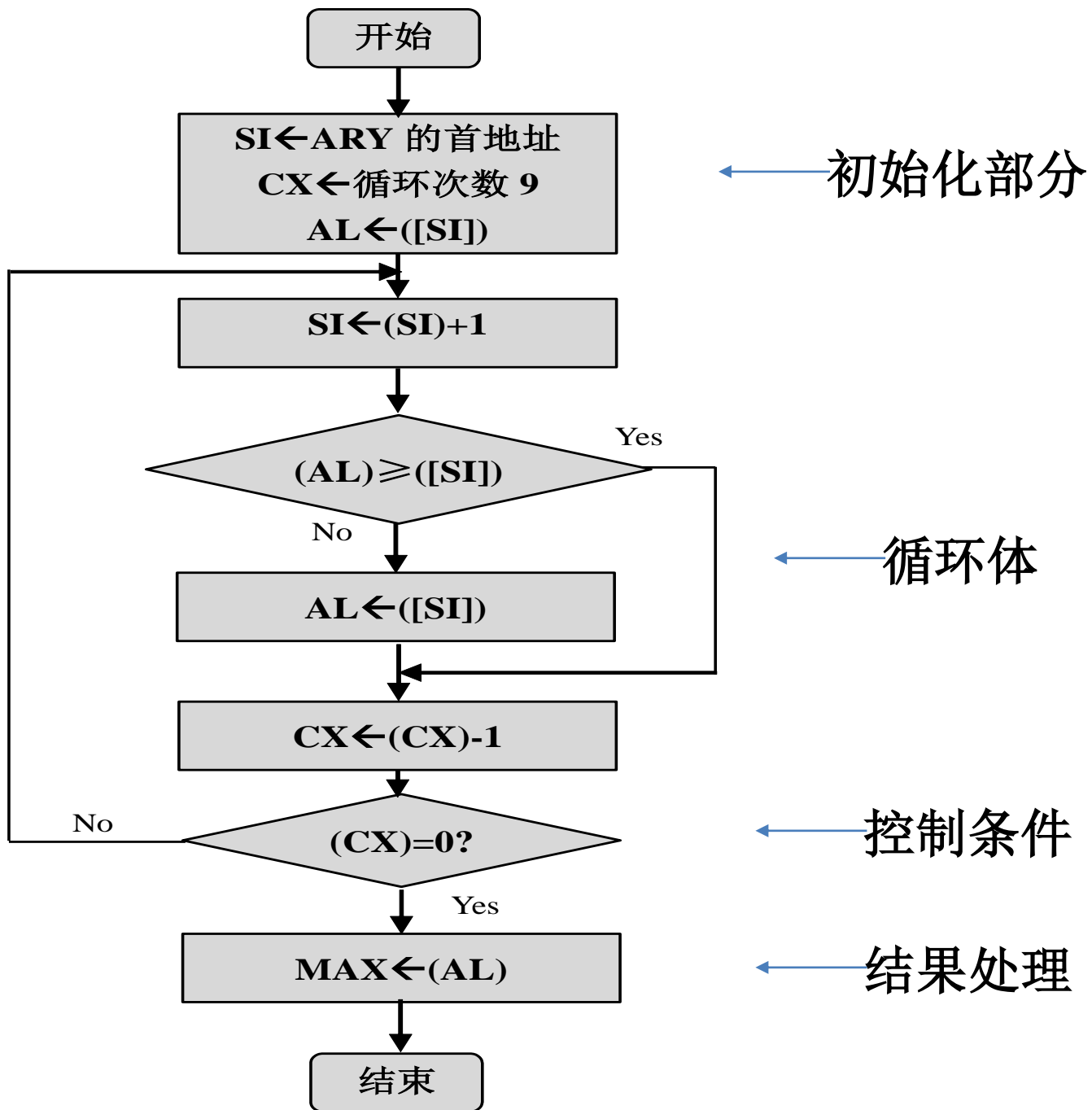
```
ARY    DB    17, 5, 40, 0, 67
        DB    12, 34, 78, 32, 10
MAX    DB    ?
```

DATA END

STACK1 SEGMENT PARA STACK

```
DW 20H DUP(0)
```

STACK1 ENDS





CODE SEGMENT

ASSUME CS:CODE,DS:DATA,SS:STACK1

BEGIN:

MOV AX,DATA

MOV DS,AX

MOV SI, OFFSET ARY ; SI指向ARY的第一个元素

MOV CX, 9 ;CX作次数计数器

MOV AL, [SI] ;取第一个元素到AL

LOP: INC SI ;SI指向后一个元素

CMP AL, [SI] ;比较两个数

JAE BIGER ;前元素 \geq 后元素转移

MOV AL, [SI] ;取较大数到AL

BIGER: DEC CX ;减1计数

JNZ LOP ;未比较完转回去，否则顺序执行

MOV MAX, AL ;存最大数

MOV AH,4CH

INT 21H

CODE ENDS

END BEGIN



4 子程序设计

- 子程序是程序的一部分，是完成特定功能的程序段，它能够在程序中的任何地方被调用。
- 子程序的调用与返回是由指令**CALL**和**RET**来完成的。
- 子程序中一般都要使用寄存器，除了要返回参数的寄存器外，在子程序设计的开始部分，要将用到的寄存器进行压栈保护，在子程序结束返回调用程序之前要进行出栈恢复。
- 子程序和调用程序直接的信息传送称为参数传递。



(1) 子程序设计举例

例，实现一个回车和换行功能的子程序。

DPCRLF	PROC		
	PUSH	AX	
	PUSH	DX	
	MOV	DL,	0DH
	MOV	AH,	2
	INT	21H	
	MOV	DL,	0AH
	MOV	AH,	2
	INT	21H	
	POP	DX	
	POP	AX	
	RET		
DPCRLF	ENDP		



(2) 主/子程序设计举例

例，将两个给定的二进制数BIN1和BIN2(8位和16位)逐位转换为ASCII码字符串，并存储于ASCBUF中。

DATA SEGMENT

BIN1 DB 35H

BIN2 DW 0AB48H

ASCBUF DB 20H DUP(?)

DATA ENDS

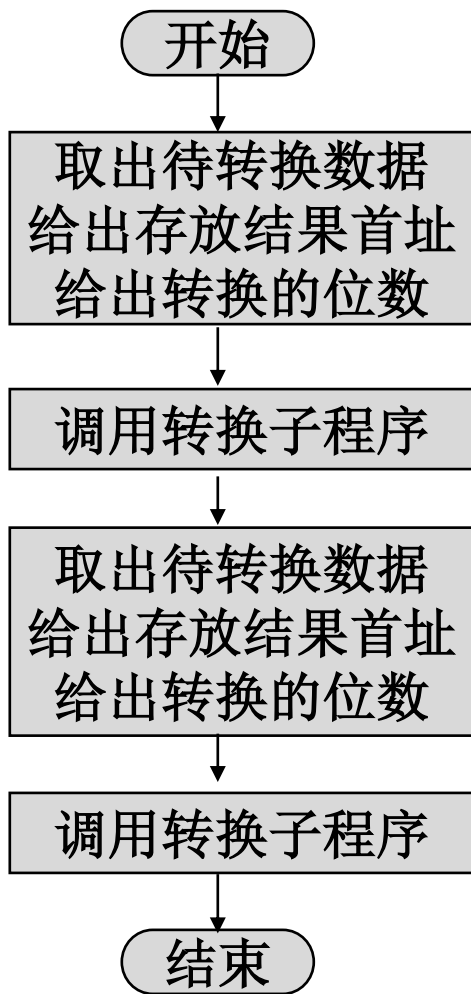
STACK1 SEGMENT PARA STACK

DW 20H DUP(0)

STACK1 ENDS



主程序框图



(寄存器参数传递)

子程序：





COSEG SEGMENT

ASSUME CS:COSEG,DS:DATA,SS:STACK1

START: MOV AX,DATA

MOV DS,AX

XOR DX,DX

LEA DI,ASCBUF;存放ASCII码的单元首址送DI

主程序:

MOV DH,BIN1 ;待转换的第1个数据送DH

MOV CX,8 ;待转换的二进制数的位数

CALL BINASC ;调用转换子程序

MOV DX,BIN2 ;待转换的第二个数据送DX

MOV CX,16

CALL BINASC

MOV AH,4CH

INT 21H



转换子程序

BINASC PROC

PUSH AX

LOP: ROL DX, 1 ; 最高位移入最低位

MOV AL, DL

AND AL, 1 ; 保留最低位, 屏蔽其它位

ADD AL, 30H ; AL中即为该数字字符(0或1)的ASCII码

MOV [DI], AL ; 存结果

INC DI ; 修改地址指针

LOOP LOP

POP AX

RET

BINASC ENDP

COSEG ENDS

END START