**西安电子科技大学**

**微机系统综合实验 课程实验报告**

## **实验名称** 实验二 数制转换与数值运算编程

网络与信息安全学院 2118021 班

成 绩

姓名 盖乐 学号 21009200991

同作者

实验日期 2023 年 4 月 15 日

|  |
| --- |
| 指导教师评语：  指导教师：  年 月 日 |

1. 实验要求
2. 将ASCII码表示的十进制数转换为二进制数，自行绘制流程图并编写程序。

十进制表示为：

 （1）

Di代表十进制数0，1，2，…，9；

上式转换为：

 （2）

由式（2）可归纳十进制数转换为二进制数的方法：从十进制数的最高位Dn开始作乘10加次位的操作，依次类推，则可求出二进制数的结果。

表1.1 数制对应关系表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 十六进制 | BCD码 | 二进制  机器码 | ASCII码 | 七段码 | |
| 共阳 | 共阴 |
| 0 | 0000 | 0000 | 30H | 40H | 3FH |
| 1 | 0001 | 0001 | 31H | 79H | 06H |
| 2 | 0010 | 0010 | 32H | 24H | 5BH |
| 3 | 0011 | 0011 | 33H | 30H | 4FH |
| 4 | 0100 | 0100 | 34H | 19H | 66H |
| 5 | 0101 | 0101 | 35H | 12H | 6DH |
| 6 | 0110 | 0110 | 36H | 02H | 7DH |
| 7 | 0111 | 0111 | 37H | 78H | 07H |
| 8 | 1000 | 1000 | 38H | 00H | 7FH |
| 9 | 1001 | 1001 | 39H | 18H | 67H |
| A |  | 1010 | 41H | 08H | 77H |
| B |  | 1011 | 42H | 03H | 7CH |
| C |  | 1100 | 43H | 46H | 39H |
| D |  | 1101 | 44H | 21H | 5EH |
| E |  | 1110 | 45H | 06H | 79H |
| F |  | 1111 | 46H | 0EH | 71H |

1. BCD码转换为二进制数

将四个二位十进制数的BCD码存放于3500H起始的内存单元中，将转换的二进制数存入3510H起始的内存单元中，自行绘制流程图并编写程序。

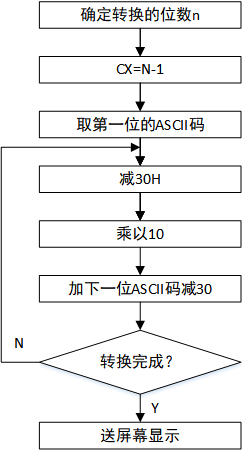
1. 两个非压缩BCD数加法程序。

要求：被加数和加数均为4位数，从键盘输入，分别放在SBCD1和SBCD2开始的地址单元中。运算结果放在SSUM开始的地址单元中。并在虚拟终端上以“SBCD1+SBCD2=SSUM”格式进行显示，自行绘制流程图并编写程序。

1. 从键盘上输入任意两个不大于2位数的正整数，计算其乘积。结果在屏幕上显示，自行绘制流程图并编写程序。
2. 实验目的
3. 掌握不同进制数及编码相互转换的程序设计方法，加深对数制转换的理解。
4. 熟悉在EMU8086集成环境中程序调试的方法。
5. 掌握使用运算类指令编程及调试方法。
6. 掌握运算类指令对各状态标志位的影响及其测试方法。
7. 学习使用软件监视变量的方法。
8. 实验代码及实验结果

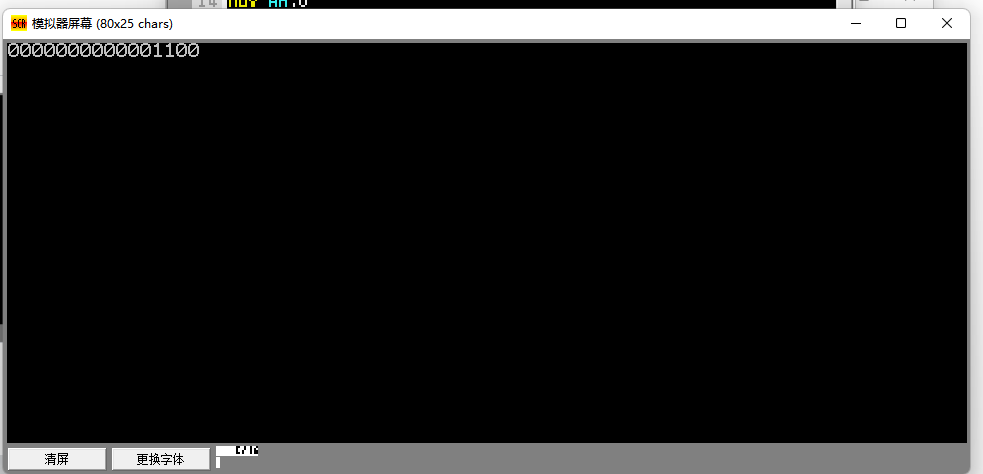
实验一

1. 实验流程图：



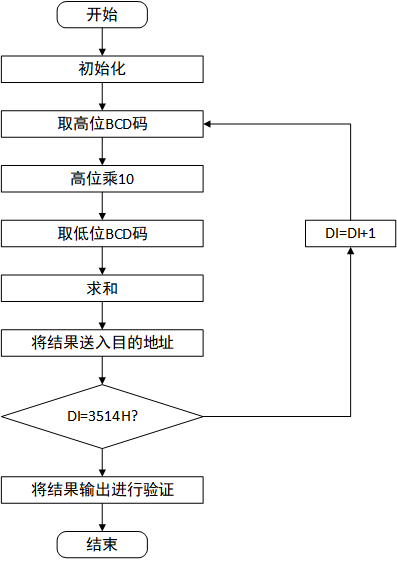
1. 实验代码：
2. DATA SEGMENT ;   定义源数据段
3. MSR DB '00012';     将十进制数00012放入DB中
4. LEN EQU $-MSR
5. DATA ENDS
7. CODE SEGMENT ;定义代码段
8. ASSUME CS:CODE, DS:DATA
10. START: MOV AX, DATA
11. MOV DS,AX
12. MOV SI, OFFSET MSR
13. MOV CX,LEN-1
14. MOV AL, DS:[SI] ;       开始传输数据
15. MOV AH,0
16. SUB AX,30H

19. NEXT1: MOV BX, 0AH
20. MUL BX ;   将AX中的数乘以10，所得的结果高位保存到DX中，低位保存到AX中
21. INC SI
22. MOV DL,DS:[SI]
23. MOV DH,0
24. SUB DX,30H ;  将数字对应的ASCII码转换成响应的二进制数
25. ADD AX,DX
26. DEC CX
27. JNZ NEXT1
28. MOV CL,16 ;   CL用作循环NEXT2的计数器，因为5位十进制数转化成二进制只有十六位，所以循环次数为16次
29. MOV BX,AX
31. NEXT2: ROL BX,1
32. MOV DL,BL
33. AND DL,01H
34. ADD DL,30H ;  把BX中的数以二进制形式输出
35. MOV AH,2H
36. INT 21H
37. DEC CL
38. JNZ NEXT2
39. MOV AX,4C00H
40. INT 21H
42. CODE ENDS
43. END START
44. 实验结果：



实验二

1. 实验流程图：

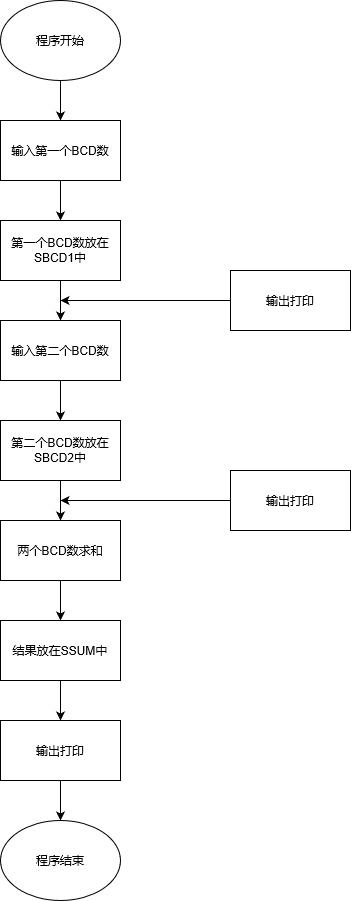


1. 实验代码：
2. CODE SEGMENT
3. ASSUME CS:CODE
4. START:
5. MOV AX,0
6. MOV DS,AX
7. MOV SI,3500H;将 SI 置为 3500H,以便后续将数据送入 3500H 开始的内存单元
8. MOV [SI],0000B
9. MOV [SI+1],0001B
10. MOV [SI+2],0010B
11. MOV [SI+3],0100B
12. MOV [SI+4],0100B
13. MOV [SI+5],1000B
14. MOV [SI+6],1001B
15. MOV [SI+7],0111B;将四个二位十进制数的 BCD 码依次送入
16. MOV DI,3510H;将 DI 置为 3510H,以便后续将数据送入 3510H 开始的内存单元
17. MOV CX,4;共四个数循环次数位 4
18. LOOP\_1:
19. MOV AH,[SI];取一个数的高位
20. MOV AL,[SI+1];取一个数的低位
21. AAD;将 AX 中两个未压缩十进制数进行处理
22. MOV [DI],AL;将最终结果送回 3510 开始的内存单元
23. INC SI;
24. INC SI;指向下一个数的高位
25. INC DI;指向下一个位置
26. LOOP LOOP\_1
27. MOV DI,3510H
28. LOOP\_2:
29. MOV BL,[DI]
30. INC DI
31. MOV CX,8;循环次数为 8
32. LOOP\_3:
33. MOV AL,BL
34. AND AL,80H;每次只取最高位进行输出
35. ROL AL,1
36. ADD AL,30H;获取 ASCII 码
37. MOV DL,AL
38. MOV AH,02H
39. INT 21H
40. SAL BL,1;将数字左移一位
41. LOOP LOOP\_3
42. MOV DL,0AH
43. MOV AH,02H
44. INT 21H;输出回车
45. MOV DL,0DH
46. MOV AH,02H
47. INT 21H;输出换行
48. CMP DI,3514H
49. JNE LOOP\_2
50. CODE ENDS
51. END START
52. 实验结果：

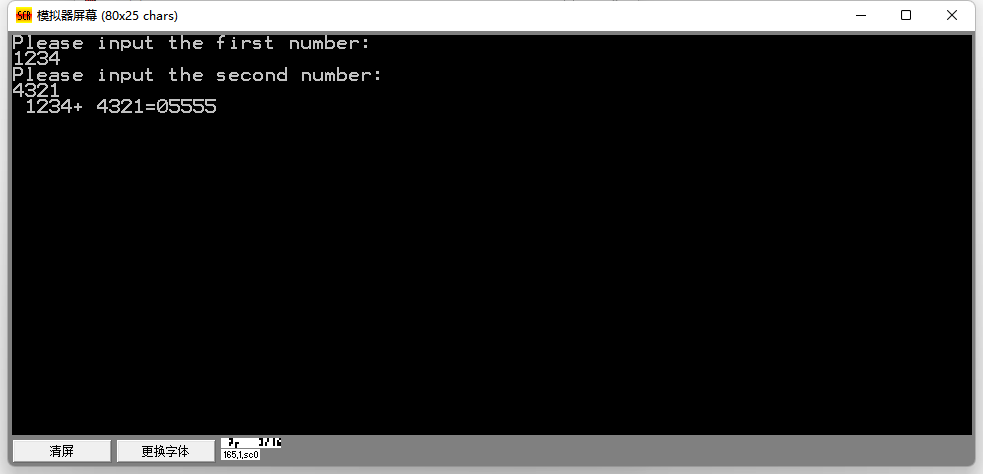


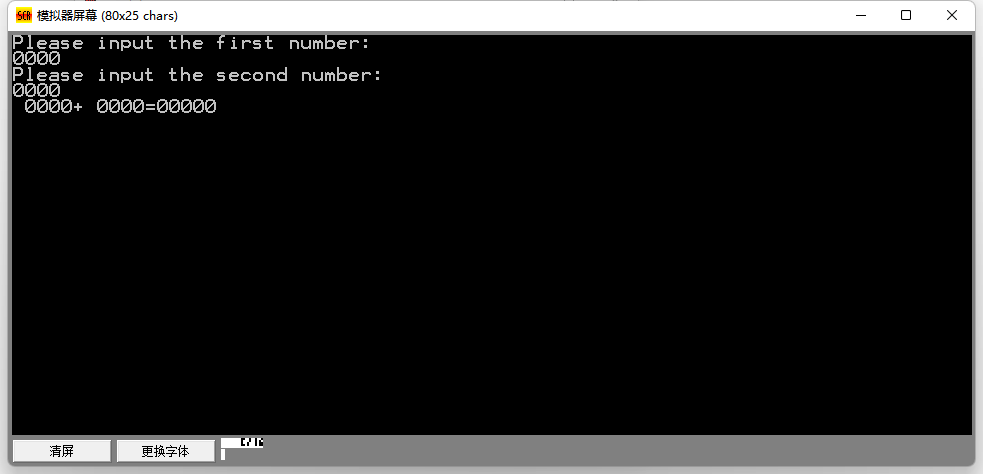
实验三

1. 实验流程图：



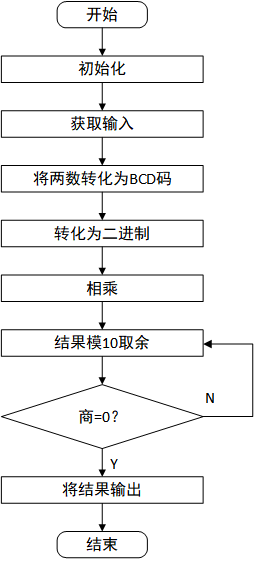
1. 实验代码：
2. DATA SEGMENT
3. STRING\_1 DB 'Please input the first number:',0AH,0DH,'$'
4. STRING\_2 DB 0AH,0DH,'Please input the second number:',0AH,0DH,'$'
5. SBCD1 DB 0AH,0DH,0,0,0,0,0,'+','$'
6. SBCD2 DB 0,0,0,0,0,'=','$'
7. SSUM DB 0,0,0,0,0,'$'
8. DATA ENDS
10. STACK SEGMENT
11. STA DB 10 DUP(0)
12. STACK ENDS
14. CODE SEGMENT
15. ASSUME CS:CODE,DS:DATA,SS:STACK
16. START:
17. MOV AX,DATA
18. MOV DS,AX
19. MOV DX,OFFSET STRING\_1;
20. MOV AH,09H
21. INT 21H;输出第一个数的提示信息
22. MOV SI,OFFSET SBCD1
23. ADD SI,3;前面有 0AH,ODH,0
24. CALL INPUT;调用 input 过程获取输入并送入 SI 指向位置
25. MOV DX,OFFSET STRING\_2;
26. MOV AH,09H
27. INT 21H;输出第二个数的提示信息
28. MOV SI,OFFSET SBCD2
29. INC SI;前面有一个 0
30. CALL INPUT;调用 input 过程获取输入并送入 SI 指向位置
31. MOV DX,OFFSET SBCD1
32. MOV AH,09H
33. INT 21H;输出换行符，第一个数和加号
34. MOV DX,OFFSET SBCD2
35. MOV AH,09H
36. INT 21H;输出第二个数和等号
37. MOV SI,OFFSET SBCD1
38. ADD SI,3;前面有 0AH,ODH,0
39. CALL GET;将第一个四位数中的每一位换成 BSD 码
40. MOV SI,OFFSET SBCD2
41. INC SI;前面有一个 0
42. CALL GET;将第二个四位数中的每一位换成 BCD 码
43. DEC SI;将 SI 指向第二个四位数的最后一位（GET 过程中多加了一次）
44. MOV DI,OFFSET SBCD1
45. ADD DI,6;将 DI 指向第一个四位数的最后一位
46. MOV CX,5;循环 5 次，进行 5 次加法
47. CLC;清除 CF 位
48. MOV BX,OFFSET SSUM;
49. ADD BX,4;将 BX 指向和的最后一位
50. LOOP\_3:
51. MOV AL,[SI]
52. ADC AL,[DI];用最低位进行带进位加法
53. AAA;将 AL 中未压缩的十进制进行调整
54. MOV [BX],AL;送回
55. DEC SI
56. DEC DI
57. DEC BX
58. LOOP LOOP\_3
59. MOV SI,OFFSET SSUM
60. CALL GET\_1;将每一位数字转化为 ASCII 码
61. OUTPUT:
62. MOV DX,OFFSET SSUM
63. MOV AH,09H
64. INT 21H;输出最终结果
65. JMP OVER
67. INPUT PROC
68. PUSH CX
69. PUSH AX
70. MOV CX,4;循环 4 次
71. LOOP\_1:
72. MOV AH,01H
73. INT 21H
74. MOV [SI],AL;获取一位输入
75. INC SI;SI+1
76. LOOP LOOP\_1
77. POP AX;恢复使用过的 AX 和 CX
78. POP CX;恢复使用过的 AX 和 CX
79. RET;返回
80. INPUT ENDP
82. GET PROC
83. XOR AX,AX
84. MOV CX,4;循环 4 次
85. LOOP\_2:
86. MOV AL,[SI]
87. SUB AL,30H;对每一位减 30H
88. MOV [SI],AL
89. INC SI
90. LOOP LOOP\_2
91. RET;返回
92. GET ENDP
94. GET\_1 PROC
95. XOR AX,AX
96. MOV CX,5;循环 5 次
97. LOOP\_4:
98. MOV AL,[SI]
99. ADD AL,30H;对每一位加 30H 将 BCD 转化为 ASCII 码
100. MOV [SI],AL
101. INC SI
102. LOOP LOOP\_4
103. RET;返回
104. GET\_1 ENDP
106. OVER:
108. CODE ENDS
109. END START
110. 实验结果：





实验四

1. 实验流程图：



1. 实验代码：
2. DATA SEGMENT
3. STR1 DB 'PLEASE INPUT THE FIRST NUMBER:$'
4. STR2 DB 'PLEASE INPUT THE SECOND NUMBER:$'
5. STR3 DB 'RESULT=$'
6. A DW ?
7. B DW ?
8. C DW ?
9. DATA ENDS
10. CODE SEGMENT
11. ASSUME CS:CODE,DS:DATA
12. MAIN PROC FAR
13. START:
14. MOV AX,DATA
15. MOV DS,AX
16. LEA DX,STR1
17. MOV AH,9H
18. INT 21H
19. CALL INPUT
20. MOV A,BX
21. LEA DX,STR2
22. MOV AH,9H
23. INT 21H
24. CALL INPUT
25. MOV B,BX
26. LEA DX,STR3
27. MOV AH,9H
28. INT 21H
29. MOV AX,A
30. AAD
31. MOV BX,AX
32. MOV AX,B
33. AAD
34. MUL BX
35. MOV WORD PTR C,AX
36. MOV DI,0AH
37. MOV CX,0
38. TOTEN:
39. DIV DI
40. PUSH DX
41. INC CX
42. CMP AX,0
43. JE PRINT
44. CWD
45. JMP TOTEN
46. PRINT:
47. POP DX
48. ADD DL,30H
49. MOV AH,2H
50. INT 21H
51. LOOP PRINT
52. RET
53. MAIN ENDP
54. INPUT PROC
55. MOV BX,0
56. INPUTA:
57. MOV AH,1
58. INT 21H
59. SUB AL,30H
60. MOV BL,AL
61. MOV AH,1
62. INT 21H
63. CMP AL,0DH
64. JE EXIT
65. SUB AL,30H
66. MOV CL,8
67. ROL BX,CL
68. MOV BL,AL
69. JMP EXIT
70. EXIT:
71. CALL CRLF
72. RET
73. INPUT ENDP
74. CRLF PROC NEAR
75. MOV DL,0AH
76. MOV AH,2H
77. INT 21H
78. MOV DL,0DH
79. MOV AH,2H
80. INT 21H
81. RET
82. CRLF ENDP
84. CODE ENDS
85. END START
86. 实验结果：



1. 实验总结

本次实验中，我们学习了8086汇编语言的基础知识和一些常用的指令，以及如何在汇编程序中进行输入输出。在实践过程中，我们也发现了一些需要注意的点：

1.在8086汇编中，如果要实现换行输出，需要同时使用“回车”和“换行”两个控制字符。

2.汇编程序中不同寄存器有不同的作用，特别是DX和AX寄存器经常用于存储运算结果和源数据，因此需要特别小心处理，避免数据丢失。

3.如果需要输出字符串，必须在字符串末尾加上‘$’符号来表示字符串的结束位置。

4.关于栈的使用，需要注意栈的大小，防止溢出，同时在使用DI和SI指针时也要小心处理，避免越界出现错误。