**实验报告**

专业： 电子信息工程

姓名： 冯静怡

学号： 3220104119

日期： 2024.3.28

地点： 紫金港东三406

课程名称： 微机原理与应用实验 指导老师： 胡斯登

实验名称： 单片机指令系统BCD控制

Lab3 单片机指令系统BCD控制

1. 课内完成内容
2. 进制转换（2/16进制转换为10进制）
3. ORG 0000H
4. SJMP MAIN
5. ORG 0080H
6. MAIN:
7. MOV R0,#8H     ;一共8位数据
8. MOV 30H,#00H   ;30H存储结果的低二位BCD码
9. MOV 31H,#00H   ;31H存储结果的高一位BCD码
10. LOOP:
11. MOV A, R1   ;R1存储的是待转换的二进制码
12. RLC A       ;取出最高位，在CY中
13. MOV R1,A
14. MOV A,30H      ;30H存储结果的低二位BCD码(已经计算的部分)
15. ADDC A,ACC     ;相当于将原BCD码×2，并加上CY中的新数据
16. DA A           ;将结果转换为BCD码
17. MOV 30H,A      ;存储新的低二位BCD码
18. MOV A, 31H     ;31H存储结果的高一位BCD码(已经计算的部分)
19. ADDC A,ACC     ;相当于将高位数据×2
20. MOV 31H,A      ;存储新的高一位BCD码
21. DJNZ R0,LOOP   ;循环8次
22. SJMP $
23. END
24. BCD码加法
25. ORG 0000H
26. SJMP START
27. ORG 0030H
28. START:
29. MOV R3, #10H   ;R3代表BCD码最高为16位
30. MOV R0, #40H   ;R0代表第一个BCD码的起始位置
31. MOV R1, #50H   ;R1代表第二个BCD码的起始位置
32. MOV R6, #60H   ;R6代表结果的起始位置
33. MOV R7,#00H    ;R7代表进位
34. LOOP:
35. MOV A,@R0  ;将第一个BCD码的值取出
36. ADD A,R7  ;加上进位
37. ADD A,@R1 ;加上第二个BCD码的值
38. DA A      ;十进制调整，此时的A的范围为#00H~#19H
39. SWAP A    ;将A的高低四位进行交换
40. MOV 71H,R1     ;将R1的值暂时存入71H
41. MOV R1,#30H
42. MOV @R1,#00H   ;将30H的值清零
43. XCHD A,@R1     ;将结果的进位位存入30H
44. SWAP A         ;将A的高低四位进行交换,即将结果的未进位部分放入地位
45. MOV R1,71H     ;将R1的值恢复
46. MOV 70H,R0     ;将R0的值暂时存入70H
47. MOV 72H,A      ;将结果的未进位值暂时存入72H
48. MOV A,R6
49. MOV R0,A       ;将R6的值存入R0
50. MOV @R0,72H    ;将未进位结果存入R0
51. MOV R0,70H     ;将R0的值恢复
52. MOV R7,30H     ;将进位位存入R7
53. INC R0
54. INC R1
55. INC R6
56. DJNZ R3,LOOP   ;循环
57. SJMP $
58. END

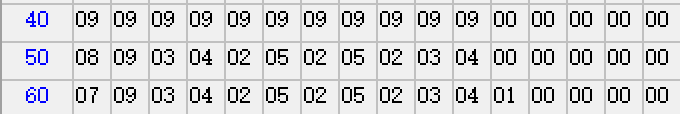
关键点在于标黄部分的内容，也就是说进行相加的过程中要考虑加数1和加数2包括之前的进位，加法之后可以直接进行DA操作。由于个位数加法不会超过100，所以不使用DA的CY改变来控制，而是增加R7作为进位标识符。

实验结果如下图：

从40H输入数字（高位在前）：999 9999 9999

从50H输入数字（高位在前）：432 5252 4398

得到60H的结果（高位在前）：1432 5252 4397



1. 课后作业一

BCD码倒序排列：

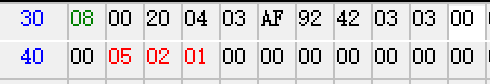
1. ORG 0000H
2. SJMP START
3. ORG 0030H
4. START:
5. MOV R0,#30H ;取出30H的值
6. MOV R1,#31H ;取出31H的值
7. XCH A, @R0  ;交换A和30H的值
8. SWAP A      ;交换A的高低位，即30H的高低位互换
9. XCH A,@R1   ;将变换后的30H放到31H的位置，并将31H的值放到A中
10. SWAP        ;再次交换A的高低位，即31H的高低位互换
11. XCH A,@R0   ;将变换后的31H放到30H的位置
12. SJMP $
13. END

处理前：

处理后：

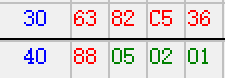
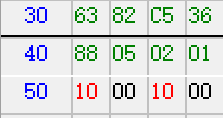
1. 课后作业二
2. ORG 0000H
3. SJMP START
4. ORG 0030H
5. START:
6. MOV R3,30H
7. MOV R0,#31H ;数据指针
8. MOV 41H,#00H
9. MOV 42H,#00H
10. MOV 43H,#00H
11. LOOP:
12. CJNE @R0,#00H,NOTZERO ;若不为0则跳转
13. INC 43H
14. SJMP NEXT
15. NOTZERO:
16. CJNE @R0,#80H,NEQUAL    ;若不为80H则跳转
17. INC 42H ;对于该数80H，代表的是-128，为负数
18. SJMP NEXT
19. NEQUAL:
20. JC POSITIVE ;若比80H小，则为正数，CY=1，则跳转
21. INC 42H     ;CY=0,否则为负数
22. SJMP NEXT
23. POSITIVE:
24. INC 41H
25. NEXT:
26. INC R0
27. DJNZ R3,LOOP
28. SJMP $
29. END

这是分支结构的运用，若该数第7位为1则为负数，即该数大于#80H则为负数。



实验结果如上如所示

1. 课后作业三
2. ORG 0000H
3. SJMP START
4. ORG 0030H
5. START:
6. MOV 30H,#63H    ;存储数据A
7. MOV 31H,#82H    ;存储数据B
8. MOV 32H,#0C5H   ;存储数据C
9. MOV 33H,#36H    ;存储数据D
10. MOV A,32H
11. CPL A       ;对C取反
12. ANL A,31H   ;C反 与 B 相与
13. XRL A,30H   ;与A异或
14. CPL A       ;取反
15. MOV 40H,A
16. MOV A,33H
17. ORL A,30H   ;D或A
18. CPL A       ;取反
19. ANL A,40H   ;与40H相与
20. MOV 40H,A
21. HTOB:       ;十六进制表示转二进制
22. MOV R3,#08H
23. MOV R0,#50H
24. LOOP:
25. RLC A    ;取A的最高位
26. MOV R6,A
27. MOV A,R3
28. JNB ACC.0,SETHIGH   ;如果R3的最低位为0，跳转到SETHIGH，设置结果字节的高位
29. MOV A,#00H  ;设置结果字节的低位
30. MOV ACC.0,C ;将进位标志位C存入ACC.0
31. ORL A,@R0   ;将结果字节的低位与已经处理过高位的结果字节相或
32. MOV @R0,A
33. INC R0      ;指向下一个结果字节
34. SJMP NEXT
35. SETHIGH:
36. MOV A,#00H  ;设置结果字节的高位
37. MOV ACC.4,C ;将进位标志位C存入ACC.4
38. MOV @R0,A
39. NEXT:
40. MOV A,R6
41. DJNZ R3,LOOP
42. SJMP $
43. END

在HTOB后的代码处设置断点，观察得到40H的值为：，说明结果为88。对十六进制数进行转二进制操作后，结果为：，最后结果存储在50H为起始位置处，且高位位于左边，结果为1000 1000 B，符合88H。