

Lab3 BCD 码

3220104119 冯静怡

一、 课内完成内容

1. 进制转换 (2/16 进制转换为 10 进制)

1	ORG 0000H	15	ADDC A,ACC ;相当于将原
2	SJMP MAIN		BCD 码×2, 并加上 CY 中的新数据
3	ORG 0080H	16	DA A ;将结果转换
4			为 BCD 码
5	MAIN:	17	MOV 30H,A ;存储新的低
6	MOV R0,#8H ;一共 8 位数据		二位 BCD 码
7	MOV 30H,#00H ;30H 存储结果的低	18	
	二位 BCD 码	19	MOV A, 31H ;31H 存储结
8	MOV 31H,#00H ;31H 存储结果的高		果的高一位 BCD 码(已经计算的部分)
	一位 BCD 码	20	ADDC A,ACC ;相当于将高
9			位数据×2
10	LOOP:	21	MOV 31H,A ;存储新的高
11	MOV A, R1 ;R1 存储的是待		一位 BCD 码
	转换的二进制码	22	
12	RLC A ;取出最高位, 在	23	DJNZ R0,LOOP ;循环 8 次
	CY 中	24	
13	MOV R1,A	25	SJMP \$
14	MOV A,30H ;30H 存储结	26	
	果的低二位 BCD 码(已经计算的部分)	27	END

2. BCD 码加法

1	ORG 0000H	9	MOV R6, #60H ;R6 代表结果的起始
2	SJMP START		位置
3	ORG 0030H	10	MOV R7,#00H ;R7 代表进位
4		11	
5	START:	12	LOOP:
6	MOV R3, #10H ;R3 代表 BCD 码最高	13	MOV A,@R0 ;将第一个 BCD 码
	为 16 位		的值取出
7	MOV R0, #40H ;R0 代表第一个 BCD	14	ADD A,R7 ;加上进位
	码的起始位置	15	ADD A,@R1 ;加上第二个 BCD 码
8	MOV R1, #50H ;R1 代表第二个 BCD		的值
	码的起始位置	16	DA A ;十进制调整, 此时
			的 A 的范围为#00H~#19H

17	SWAP A ;将 A 的高低四位进行交换	27	MOV A,R6
18	MOV 71H,R1 ;将 R1 的值暂时存入 71H	28	MOV R0,A ;将 R6 的值存入 R0
19	MOV R1,#30H	29	MOV @R0,72H ;将未进位结果存入 R0
20	MOV @R1,#00H ;将 30H 的值清零	30	MOV R0,70H ;将 R0 的值恢复
21	XCHD A,@R1 ;将结果的进位位存入 30H	31	
22	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	32	MOV R7,30H ;将进位位存入 R7
23	MOV R1,71H ;将 R1 的值恢复	33	INC R0
24		34	INC R1
25	MOV 70H,R0 ;将 R0 的值暂时存入 70H	35	INC R6
26	MOV 72H,A ;将结果的未进位值暂时存入 72H	36	DJNZ R3,LOOP ;循环
		37	
		38	SJMP \$
		39	
		40	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。

40	09	09	09	09	09	09	09	09	09	09	09	00	00	00	00
50	08	09	03	04	02	05	02	05	02	03	04	00	00	00	00
60	07	09	03	04	02	05	02	05	02	03	04	01	00	00	00

二、 课后作业一

BCD 码倒序排列：

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

```

13
14 SJMP $
15 END

```

处理前:

30	12	34
----	----	----

处理后:

30	43	21
----	----	----

三、 课后作业二

```

1  ORG 0000H
2  SJMP START
3  ORG 0030H
4
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
12 LOOP:
13 CJNE @R0,#00H,NOTZERO ;若不为0
   则跳转
14 INC 43H
15 SJMP NEXT
16
17 NOTZERO:
18 CJNE @R0,#80H,NEQUAL ;若不为
   80H 则跳转
19 INC 42H ;对于该数 80H, 代表的是-
   128, 为负数
20 SJMP NEXT
21
22 NEQUAL:
23 JC POSITIVE ;若比 80H 小, 则为正
   数, CY=1, 则跳转
24 INC 42H ;CY=0, 否则为负数
25 SJMP NEXT
26
27 POSITIVE:
28 INC 41H
29
30 NEXT:
31 INC R0
32 DJNZ R3, LOOP
33
34 SJMP $
35
36 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。

30	08	00	20	04	03	AF	92	42	03	03	00
40	00	05	02	01	00	00	00	00	00	00	00

实验结果如上如所示

四、 课后作业三

```

1  ORG 0000H
2  SJMP START
3  ORG 0030H
4  START:
5  MOV 30H,#63H ;存储数据 A
6  MOV 31H,#82H ;存储数据 B
7  MOV 32H,#0C5H ;存储数据 C
8  MOV 33H,#36H ;存储数据 D
9
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	31	MOV ACC.0,C ;将进位标志位 C 存入 ACC.0
16	MOV A,33H	32	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
17	ORL A,30H ;D 或 A	33	MOV @R0,A
18	CPL A ;取反	34	INC R0 ;指向下一个结果字节
19	ANL A,40H ;与 40H 相与	35	SJMP NEXT
20	MOV 40H,A	36	SETHIGH:
21		37	MOV A,#00H ;设置结果字节的高位
22	HTOB: ;十六进制表示转二进制	38	MOV ACC.4,C ;将进位标志位 C 存入 ACC.4
23	MOV R3,#08H	39	MOV @R0,A
24	MOV R0,#50H	40	
25	LOOP:	41	NEXT:
26	RLC A ;取 A 的最高位	42	MOV A,R6
27	MOV R6,A	43	DJNZ R3,LOOP
28	MOV A,R3	44	
29	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	45	SJMP \$
30	MOV A,#00H ;设置结果字节的低位	46	END

在 HTOB 后的代码处设置断点，观察得到 40H 的值为：

30	63	82	C5	36
40	88	05	02	01

，说明结果

为 88。对十六进制数进行转二进制操作后，结果为：

30	63	82	C5	36
40	88	05	02	01
50	10	00	10	00

，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。