**实验项目名称：第二次实验**

姓名： 赵 磊 学号： 176002104 班级： 通信19 实验时间： 3.18

姓名： 熊 康 学号： 206001232 班级： 阳明2班 实验地点：实验楼105

**一、实验目的**

* 1. 熟悉基于KEIL下的实验环境。
  2. 掌握程序调试的单步、断点、连续运行方法。
  3. 熟悉在KEIL环境下观察内部RAM , 特殊功能寄器的方法。

**二、实验内容**

1. 调试已给的程序，熟悉KEIL的使用方法，掌握程序调试的单步、断点、连续运行方法。
2. 编写并调试一个双字节无符号十进制数加法程序。其功能为将被加数写入30H、31H单元，加数写入40H、41H单元，运行程序结果写入50H、51H、52H单元中，加法程序功能为：(30H)(31H)+(40H)(41H)=(50H)(51H)(52H)（都是高位在前、低位在后）。

**三、实验说明**

实验的第一部分内容，在老师的讲解后，能较快的发现错误并改正，最后调试成功，只要观察第一部分的代码编写的格式，根据C语言，第二部分程序编写也较简单。

**四、实验仪器和设备**

PC机、Keil uVision4软件。

**五、实验原理**

C51语言。

**六、程序流程图**

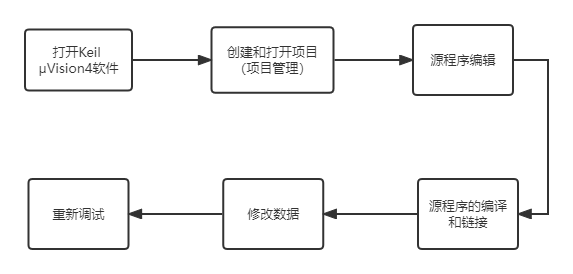


图1 程序流程图

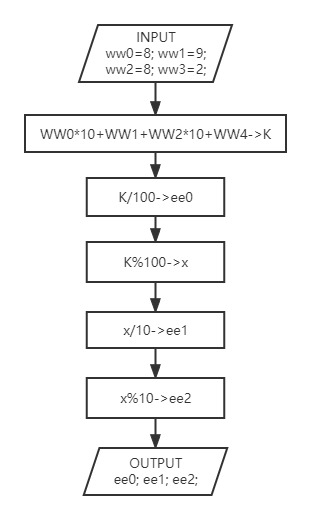


图2 伪代码流程图

**伪代码流程图分析：**

1. **程序初始化：**被加数，，加数，。
2. **程序运算：**
   1. 将被加数和加数的和放到变量中；
   2. 对和整除10取和的百位放到变量；
   3. 将和的十位和百位放到变量中暂存；
   4. 对变量整除10取和的十位；
   5. 对变量取余10取和的个位。
3. **程序结束：**得到和的百位、十位和个位上的数。

**七、实验结果与分析**

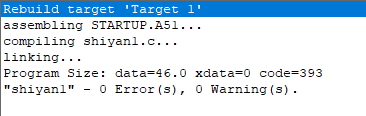


图3 程序编译成功界面图

**分析：**“0 Error(s), 0 Warning(s)”程序编译成功，没有错误和警告。

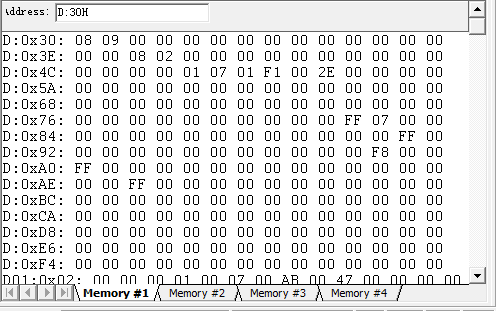


图4 片内存储器数据图

**分析：**被加数89，8存在0x30，9存在0x31中；加数82，8存在0x40，2存在0x41中；和的百位存在0x50，十位存在0x51，个位存在0x52中。和为171，所以0x50处为1，0x51处为7，0x52处为1。

**八、源代码**

1. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
2. File name: Demo
3. Description: 学习观察内部RAM和调试程序
4. Author: 熊康
5. Version: V1.0
6. Date: 2022.3.18
7. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
8. #include <stdio.h>
9. #include <intrins.h>
11. unsigned **char** a[5] \_at\_ 0x50;
13. unsigned **char** ww0 \_at\_ 0x30;
14. unsigned **char** ww1 \_at\_ 0x31;
15. unsigned **char** ww2 \_at\_ 0x32;
16. unsigned **char** ww3 \_at\_ 0x33;
17. unsigned **char** ww4 \_at\_ 0x34;
19. unsigned **char** ee0 \_at\_ 0x40;
20. unsigned **char** ee1 \_at\_ 0x41;
21. unsigned **char** ee2 \_at\_ 0x42;
23. unsigned **char** bdata BZW \_at\_ 0x20;  //标志位
25. sbit BB0 = BZW^0;   //标志位0
26. sbit BB1 = BZW^1;   //标志位1
27. sbit BB2 = BZW^2;   //标志位2
29. unsigned **char**  xx;yy;zz;ACC;B;TL0;TH0;
30. **int** i,k,y,z,x;
32. **void** main()
33. {
35. ww0=0x95;   //存储器窗口    察看30h
36. ww1=0xa6;                // 察看31h
37. ww2=0xb7;                // 察看32h
38. ww3=0x88;                // 察看33h
39. ww4=0x08;                // 察看34h
41. k=ww0+ww1+ww2+ww3+ ww4;  // 观察和堆栈窗口 察看  k=0x282
42. ww4=k/4;                     // 察看34h   =0xa0
44. ee0=ww4/100;         //察看40h   =0x01
45. xx=ww4%100;         //察看  xx  观察和堆栈窗口 察看  xx=0x3c
46. ee1=xx/10;            //察看41h  =0x06
47. ee2=xx%10;     //察看42h  =0x00
48. **if**  (ee0>2)
49. BB0=1;     //察看20h  位0
50. **if**  (ee1>=5)
51. BB1=1;     //察看20h  位1
52. **if**  (ee2<9)
53. BB2=1;     //察看20h  位2
55. **for**(i=0,k=0x55;i<5;i++,k++)
56. {
57. a[i]=k;    // 察看 存储器窗口 (50h---54h）
58. }
59. ACC=0xa0;    //专用寄存器窗口   察看 A
60. B=0xbb;       //专用寄存器窗口   察看 B
61. **switch**(ACC+1){
62. **case** 0xa1:
63. TL0=0x77;TH0=0x88;   //存储器窗口   察看8ah,8ch
64. **break**;
65. **case** 0xa2:
66. TL0=0x33;TH0=0x44;   //存储器窗口   察看8ah, 8ch
67. **break**;
68. **default**:
69. TL0=0x11;  TH0=0x22;   //存储器窗口   察看8ah, 8ch
70. **break**;
71. }
72. **while**(1);
73. }

相关流程图如图2所示。

1. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
2. File name: Test2
3. Description: 学习观察内部RAM和调试程序
4. Author: 熊康
5. Version: V1.0
6. Date: 2022.3.18
7. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
8. #include "stdio.h"
9. #include "intrins.h"
10. #include "reg51.h"
12. unsigned **char** ww0 \_at\_ 0x30;
13. unsigned **char** ww1 \_at\_ 0x31;
14. unsigned **char** ww2 \_at\_ 0x40;
15. unsigned **char** ww3 \_at\_ 0x41;
17. unsigned **char** ee0 \_at\_ 0x50;
18. unsigned **char** ee1 \_at\_ 0x51;
19. unsigned **char** ee2 \_at\_ 0x52;    //变量写入各自的单元中
21. **int** k,x;
23. **void** main()
24. {
25. //给两个两位数赋值
26. ww0=0x08;
27. ww1=0x09;
28. ww2=0x08;
29. ww3=0x02;
31. //求出总和
32. k=10\*ww0+ww1+10\*ww2+ww3;
33. TI=1;
34. printf("k=%d",k);
35. TI=0;
37. //分别求出百位十位个位的值
38. ee0=k/100;
39. x=k%100;
40. ee1=x/10;
41. ee2=x%10;
42. }