# 实验三 Keil 的使用与汇编语言仿真操作

一、实验目的

熟悉**Keil**环境，通过在**Keil**环境下调试字节加法程序，掌握汇编语言程序的调试方法，加深对**加法**、堆栈、寄存器、数据指针、汇编语言指令、机器码等基本概念的理解，为后续程序编制和调试打下基础。

二、 实验内容

1.掌握Keil环境的使用

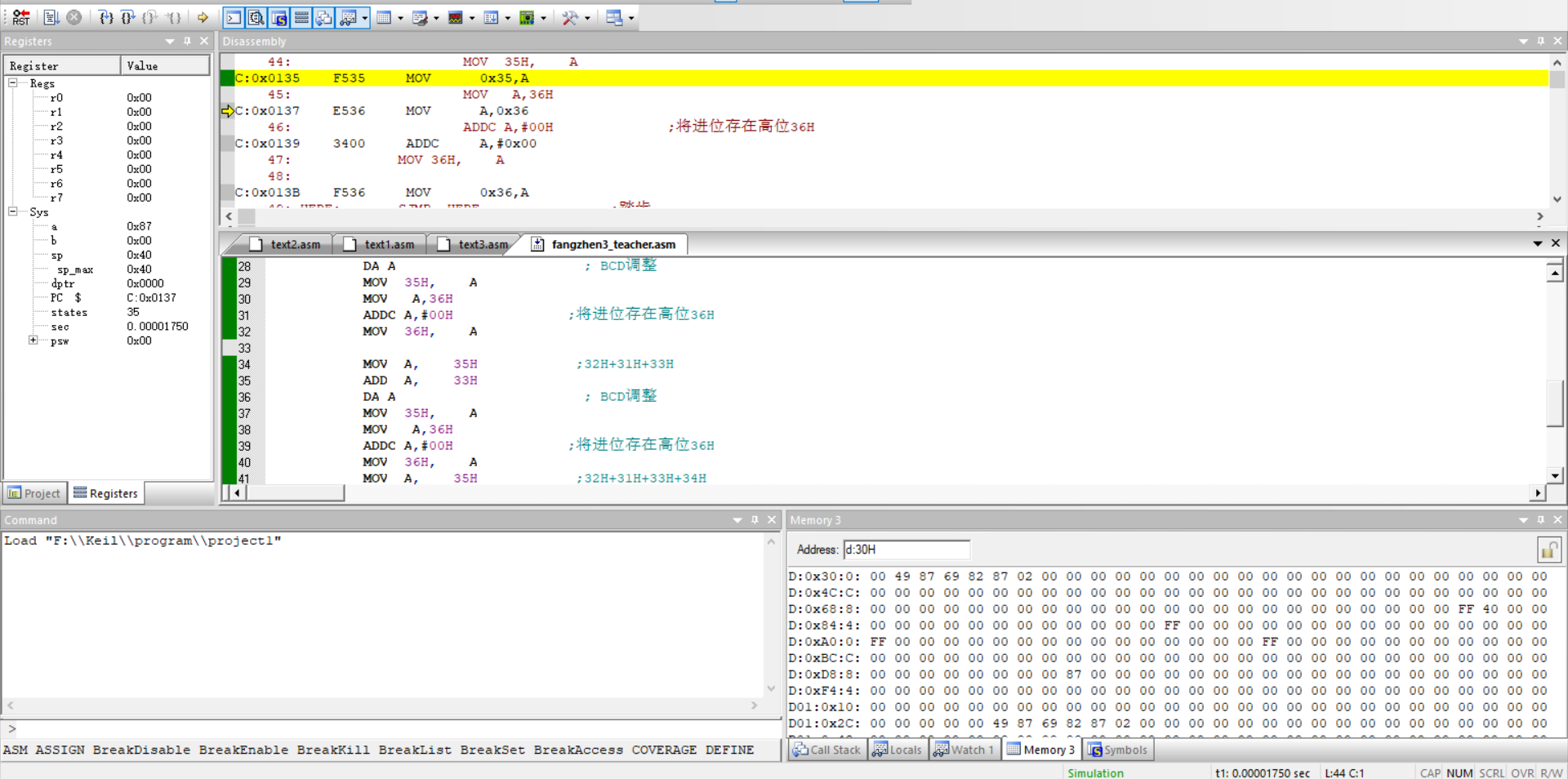
字节加法：调试e2.asm程序， 观察相关寄存器和单元的内容。

2. 编写多个十六位数的加法程序

有3个十六位压缩BCD码，连续存放在20H开始的片上数据区中，低八位在前，高八位在后。要求：将3个数的和存于30H（高八位）和31H（低八位），进位位存于32H中。

三、实验结果

1、调试e2.asm程序， 观察相关寄存器和单元的内容。

复制e2.asm程序，得到程序运行时相关寄存器和单元内容如下图所示：

由上图左上角寄存器工作区可以看到相关寄存器得内容。在地址栏中输入d:30H,可以看到从31H地址处依次存入数据：49H、87H、69H、82H、87、02。其中加数为49H、87H、69H、82H，其和结果存入地址35H，其值为87；进位信息存入36H，其值为02。且根据程序运行结果可知程序运行正确，实现了字节加法。  
 2、编写多个十六位数的加法程序

本次仿真考虑首先设置数据存入相应地址处，因为是十六位数，所以考虑采用先低八位相加，获取和以及进位信息。接着将进位数据和高八位进行相加得到高八位得和以及最终的和进位信息，最终可以得到三位十六位数的和。

初始数据设置：8949H、8269H、1413H，低八位在前，高八位在后。

理论结果分析：和为8431，进位信息为01。分别存入地址30H、31H（高低八位）和32H。

结合前面算法思路，编写相应程序如下：

ORG 0000H

LJMP MAIN

ORG 0100H

MAIN: MOV SP, #40H ;赋堆栈指针

CLR C ;进位位清零

MOV 20H, #49H ;20H单元赋值

MOV 21H, #87H ;21H单元赋值

MOV 22H, #69H ;22H单元赋值

MOV 23H, #82H ;23H单元赋值

MOV 24H, #13H ;24H单元赋值

MOV 25H, #14H ;25H单元赋值

MOV A, 20H ;20H+22H

ADD A, 22H

DA A ; BCD调整

MOV 31H, A ;前两个低位结果存入31H

MOV A,32H

ADDC A,#00H ;将进位存在32H

MOV 32H, A

MOV A, 31H ;22H+20H+24H

ADD A, 24H

DA A ; BCD调整

MOV 31H, A ;低位结果存入31H

MOV A,32H

ADDC A,#00H ;将进位存在高位32H

MOV 32H, A

在Keil中编写上述程序得到如下所示结果：

//开始加高八位

CLR C

MOV A, 21H ;21H+低八位进位

ADD A, 32H

DA A ; BCD调整

MOV 30H, A ;高位结果存入30H

MOV A, #00H

ADDC A,#00H ;将进位存在32H

MOV 32H, A

MOV A, 30H ;21H+23H

ADD A, 23H

DA A ; BCD调整

MOV 30H, A

MOV A,32H

ADDC A,#00H ;将进位存在高位32H

MOV 32H, A

MOV A, 30H ;21H+23H+25H

ADD A, 25H

DA A ; BCD调整

MOV 30H, A

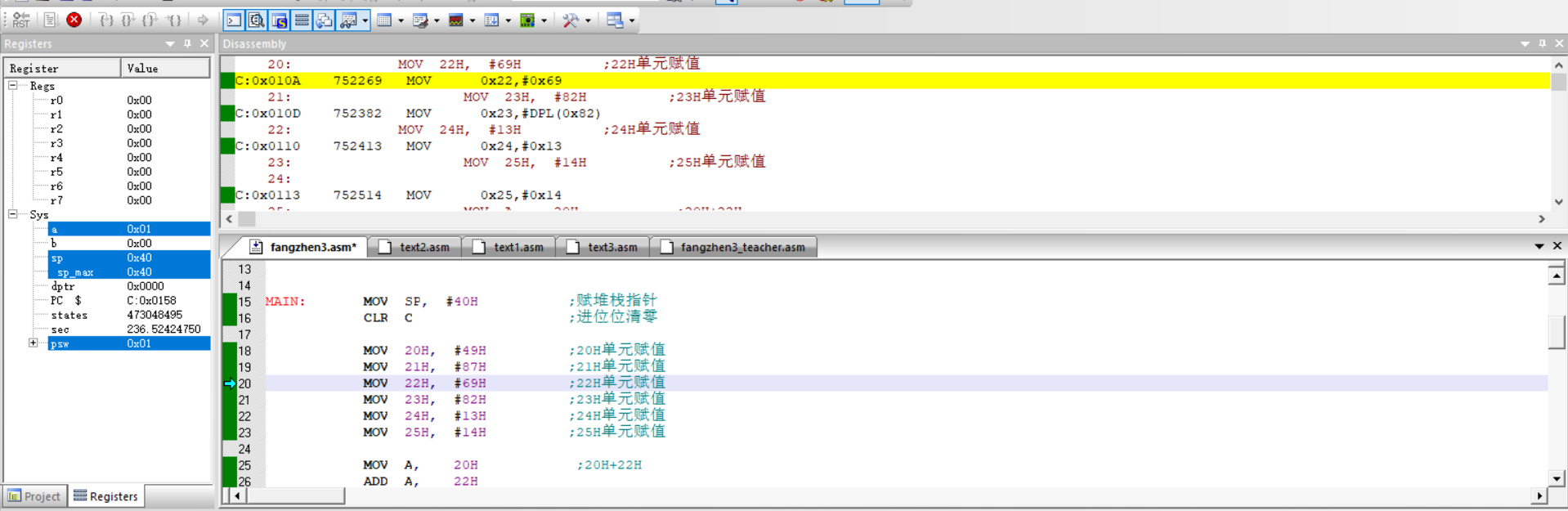
MOV A,32H

ADDC A,#00H ;将进位存在高位32H

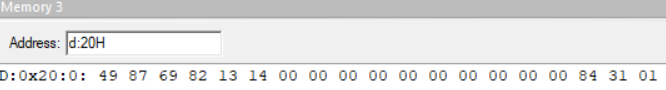
MOV 32H, A

HERE: SJMP HERE ;踏步

END



内存中相应数据如下所示：



由上图可知：初始设置数据存入相应数据，开始低八位在前高八位在后，所得结果和高八位在前低八位在后，进位信息依后存储，且所得结果和理论数据相同，由此可见仿真实验完成。

四、思考题

1. 怎样查看工作寄存器、SFR、片内RAM、片外RAM及程序代码空间内容？

进行调试后，左侧Register窗口中可以看到寄存器工作状态，工作寄存器R0-R7，特殊功能寄存器SYS下的内容。

右下角Memory窗口中，输入d：0x\_\_\_\_可以看片内RAM的内容；

输入x：0x\_\_\_\_可以查看片外RAM的内容。

输入c：0x\_\_\_\_可以查看程序代码空间内容。

1. 字节加法能实现减法吗？举一例说明。

能，根据减去一个数等于加上该数的补码，因此可以将被减数先取补码，再利用字节加法来进行相加，进而可以实现字节减法运算。

五、实验总结

通过本次仿真实验加深了自己对于51系列单片机指令系统的理解，尤其是是对字节加法和进位信息的相关理解。同时更加熟悉了Keil的使用方法，能够熟练的掌握工作寄存器状态的观察方法和相应内存信息的获取，同时对调试工作进一步熟悉，过程中遇到的相关问题也加强了自己排错纠错的能力！