电子科技大学信息与软件工程学院

**实 验 报 告**

学 号 2016220304031

姓 名 邓巧

（实验） 课程名称 ARM处理器体系结构及应用

理论教师 何兴高

实验教师 朱相印

**电子科技大学教务处制表**

**电 子 科 技 大 学**

**实 验 报 告**

**学生姓名：邓巧 学号：2016220304031 指导教师：朱相印**

**实验地点：信软楼305 实验时间：2018.5.7-5.15**

**一、实验名称：《ARM处理器体系结构及应用》实验**

**二、实验学时：16学时**

**三、实验目的：**

1.掌握基本ARM指令的使用

2.掌握ARM基本编程

**四、实验原理：**

1. C语言中的函数

* int Function（int param0，int param1）
  + 函数名称
  + 函数形参
  + 函数返回值
  + 局部变量

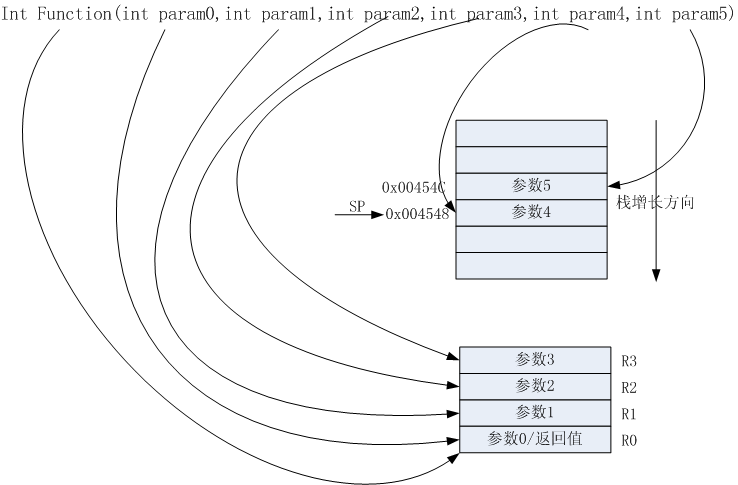
2. ARM汇编程序中的函数

* 函数名称是跳转目标地址，在汇编语言中是表明函数存储位置的标号
* 汇编语言调用函数使用BL指令
  + BL指令在跳转之前会把BL指令的下一条指令地址保存到连接寄存器LR（R14）中
* 函数的形式参数和返回值以及函数中的局部变量存储在ARM寄存器和堆栈中
* 函数返回
  + 调用mov指令将连接寄存器的值传递给程序计数器
  + 返回值存储在寄存器中

3. ARM汇编程序对寄存器的使用

|  |  |  |  |
| --- | --- | --- | --- |
| **寄存器** | **别名1** | **别名2** | **用途** |
| R0-R3 | A1-A4 |  | 参数寄存器，其中R0又被用作函数返回值寄存器 |
| R4-R8 | V1-V5 |  | 函数局部变量寄存器 |
| R9 | V6 | sb | 在RWPI情况下保存静态基地址 |
| R10 | V7 | sl | 用来保存堆栈边界地址 |
| R11 | V8 | fp | 保存结构指针 |
| R12 |  | IP | 过渡寄存器 |
| R13 |  | SP | 堆栈指针 |
| R14 |  | LR | 连接寄存器 |
| R15 |  | PC | 程序计数器 |

4. ATPCS规则



**五、实验内容：**

**1.实验环境搭建及指令熟悉与使用；**

用立即数寻址、寄存器寻址、寄存器间接寻址、基址变址寻址、堆栈寻址、多寄存器寻址、相对寻址，加S标志，加条件来使用以下指令，并通过查看寄存器、存储器、程序状态寄存器的内容，检查是否与期望一致。

**（1）寻址方式指令系统一**

（1）MOV指令；（2）MVN指令；（3）AND指令；（4）EOR指令；（5）SUB指令；（6）RSB指令；（7）ADD指令；（8）ADC指令；（9）SBC指令；（10）RSC指令；（11）TST指令；（12）TEQ指令；（13）CMP指令；（14）CMN指令；（15）ORR指令；（16）BIC指令；（17）MUL指令；（18）MLA指令；（19）UMULL指令；（20）UMLAL指令；（21）SMULL指令；（22）SMLAL指令。

**（2）寻址方式指令系统二**

（1）LDR指令；（2）STR指令；（3）LDRB指令；（4）STRB指令；（5）LDRH指令；（6）STRH指令；（7）LDMIA指令；（8）LDMIB指令；(9）LDMDA指令；（10）LDMDB指令；（11）LDMFA指令；（12）STMFA指令；（13）LDMFD指令；（14）STMFD指令；（15）LDMEA指令；（16）STMEA指令；（17）LDMED指令；（18）STMED指令；（19）SWP指令；（20）B指令；（21）BL指令；（22）BX指令；（23）BLX指令；（24）MRS指令；（25）MSR指令。

**2.子程序及调用；**

**（1）ARM主程序调用ARM/C语言子程序**

·ARM指令 主程序 调用 ARM指令 子程序；

·ARM指令 主程序 调用 C语言子程序；

·子程序的参数个数要求 至少6个，子程序功能自定，观察参数是如何传递的。

**（2）C语言主程序调用ARM子程序**

·C语言 主程序 调用 ARM指令子程序；

·子程序的参数个数要求 至少6个，子程序功能自定，观察参数是如何传递的。

**3.数字滤波；**

**（1）中值滤波及编程实验**

将 N 个无符号数（N为奇数）进行排序，中间那个数即为本次滤波所得的结果。

为了便于检查实验过程和结果，实验要求如下：

·关于 N（N为奇数）个数的值在程序中能任意、方便设置，并且放在 R0 中；

·原始数据放在内存 0X40000000 开始的地址空间；

·中值滤波的结果放在寄存器 R1 中；

·实验中先画程序流程图，再写代码。

**（2）均值滤波及编程实验**

将 N 个无符号数进行排序，去掉最大值和最小值，剩余的数求平均值，平均值即为本次滤波所得的结果。

为了便于检查实验过程和结果，实验要求如下：

·关于 N（N为偶数）个数的值在程序中能任意、方便设置，如 4、6、10、18 等，该值放在 R0 中；

·原始数据放在内存中 0X40000000 开始的地址空间；

·均值滤波的结果放在寄存器 R1 中；

·实验中先画程序流程图，再写程序。

**4.数值变换。**

**（1）数值变换及编程实验**

将寄存器中的一个无符号 32 位二进制数先转换为 10 进制数

为了便于检查实验过程和结果，要求如下：

·要转换的数在 R0 中，该数能任意设置；

·十进制数放在 0X40000000 开始地址；

·实验中先画程序流程图，再写代码

**（2）数码变换及编程实验**

将内存单元中的 10 进制数转换为 LED 码。

为了便于检查实验过程和结果，要求如下：

·十进制数放在0X40000000开始地址；

·LED 码数放在 0X40000010 开始地址；

·实验中先画程序流程图，再写实验代码。

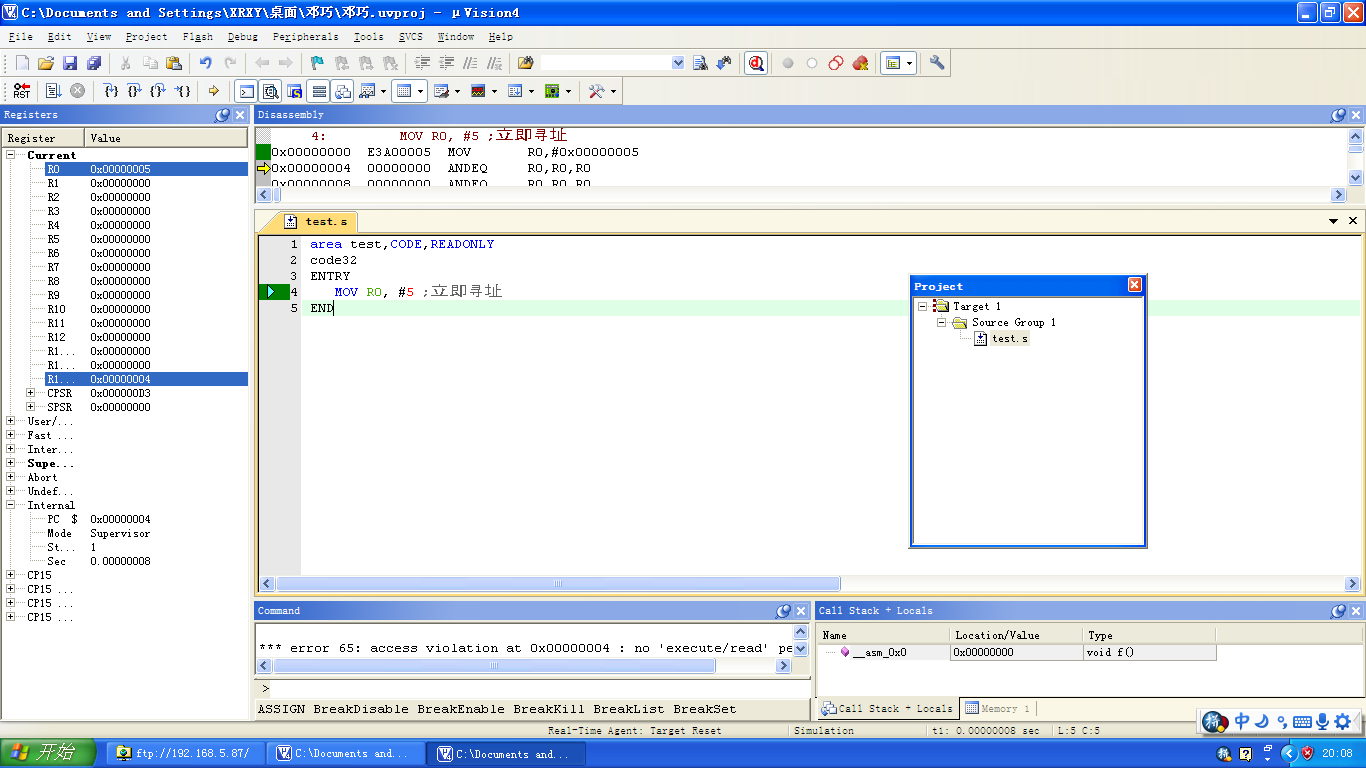
**六、实验器材（设备、元器件）：**

PC机一台

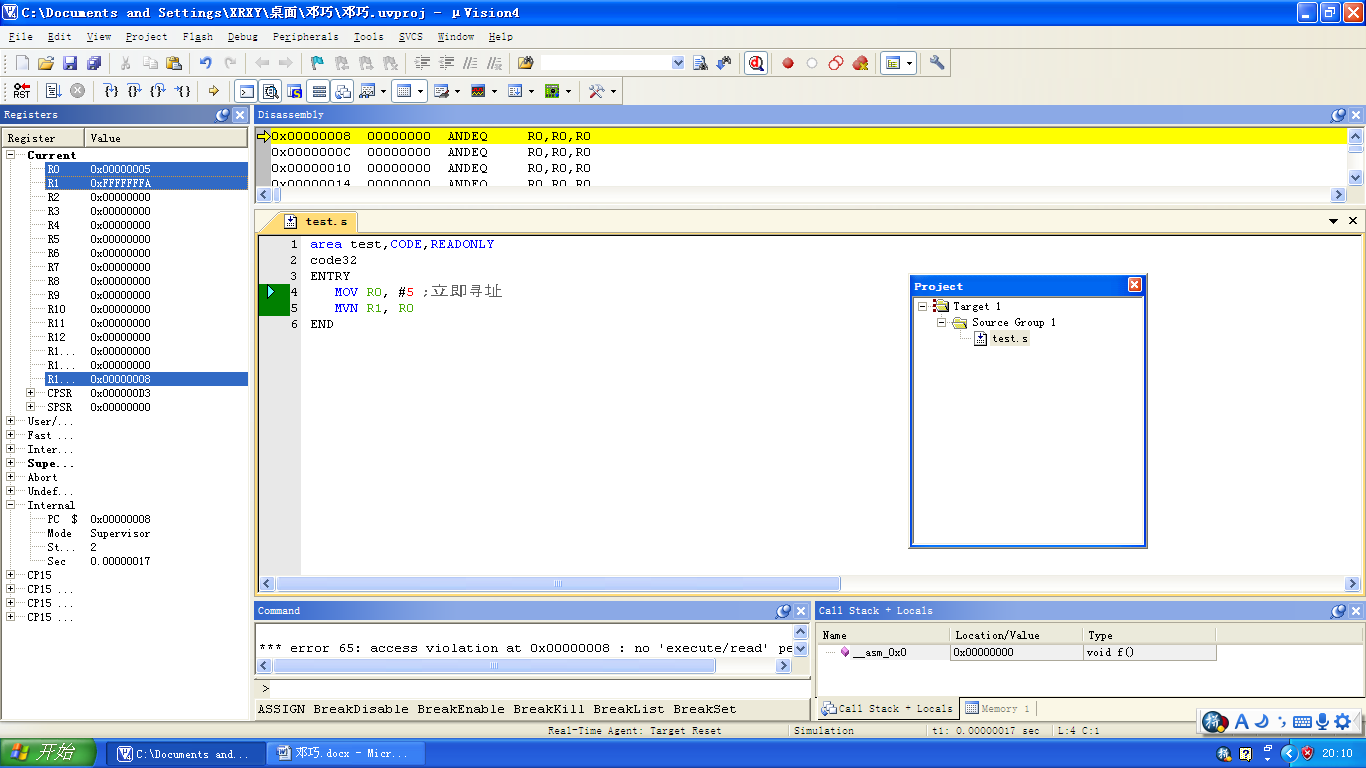
**七、实验结果与分析（含重要数据结果分析或核心代码流程分析）**

**实验一，寻址方式指令系统一**

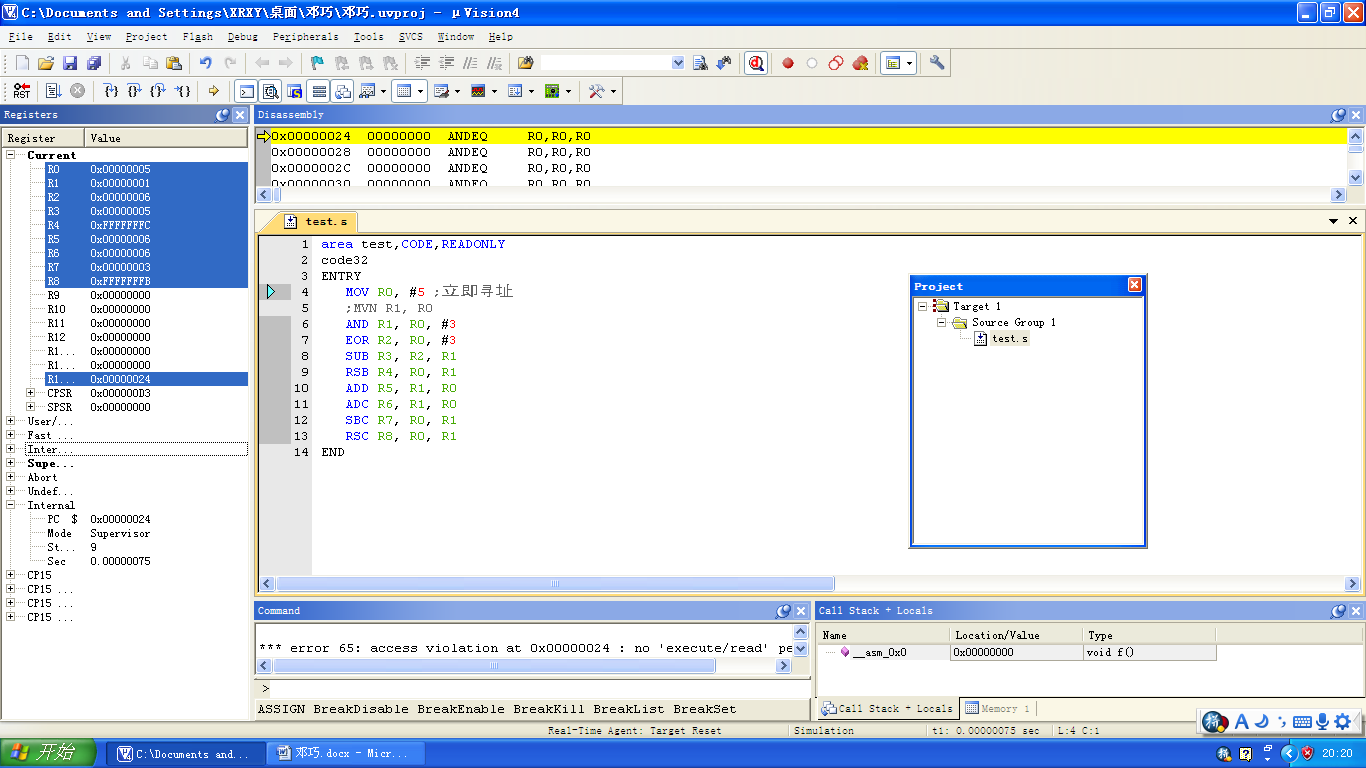
|  |
| --- |
| area test,CODE,READONLY  code32  ENTRY  MOV R0, #5 ;Á¢¼´Ñ°Ö·  MOV R1, #10  ;MVN R1, R0  ;AND R1, R0, #3  ;EOR R2, R0, #3  ;SUB R3, R2, R1  ;RSB R4, R0, R1  ;ADD R5, R1, R0  ;ADC R6, R1, R0  ;SBC R7, R0, R1  ;RSC R8, R0, R1  ;TST R0, 0x00000001  ;TEQ R1, R0  ;MOVNE R2, #1  ;CMP R0, #5  ;CMN R0, #0  ;MOVNE R1, #1  ORR R1, R1, #3  BIC R1, R0, #0xB  MOV R2, #10  MUL R3, R2, R0  MLA R4, R2, R0, R3  UMULL R5, R6, R2, R0  MOV R7, #1  UMLAL R7, R8, R2, R0  SMLAL R8, R9, R2, R3    END |

MOV：

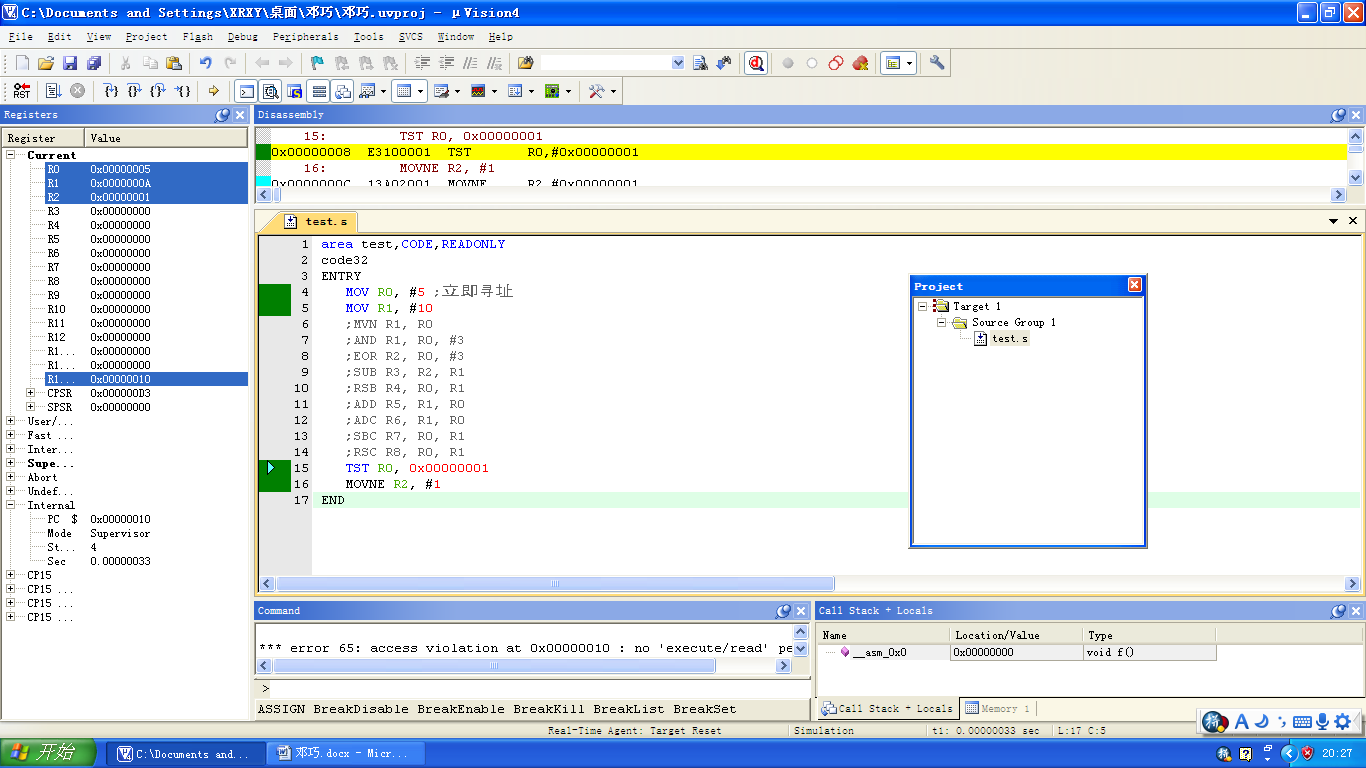
MVN：



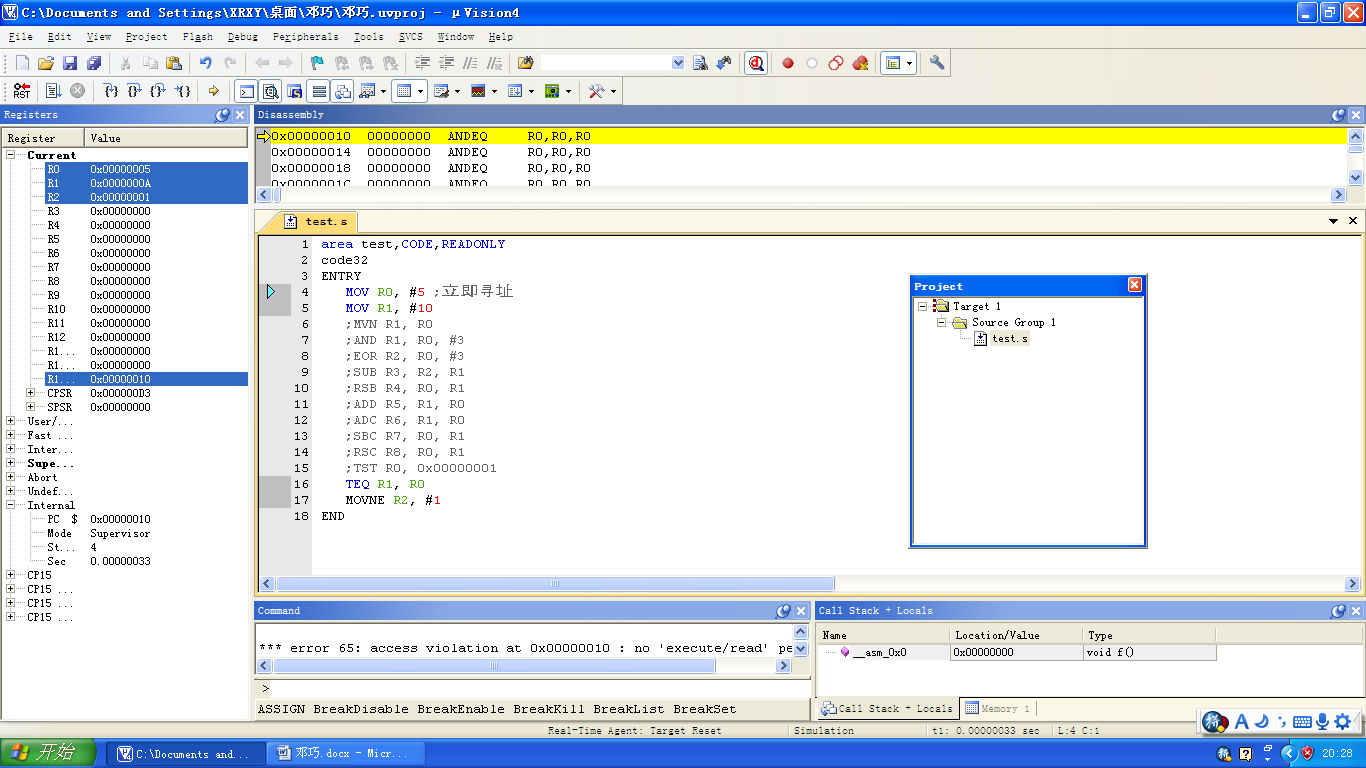
AND EOR SUBRSBADD ADC SBC RSC：

****

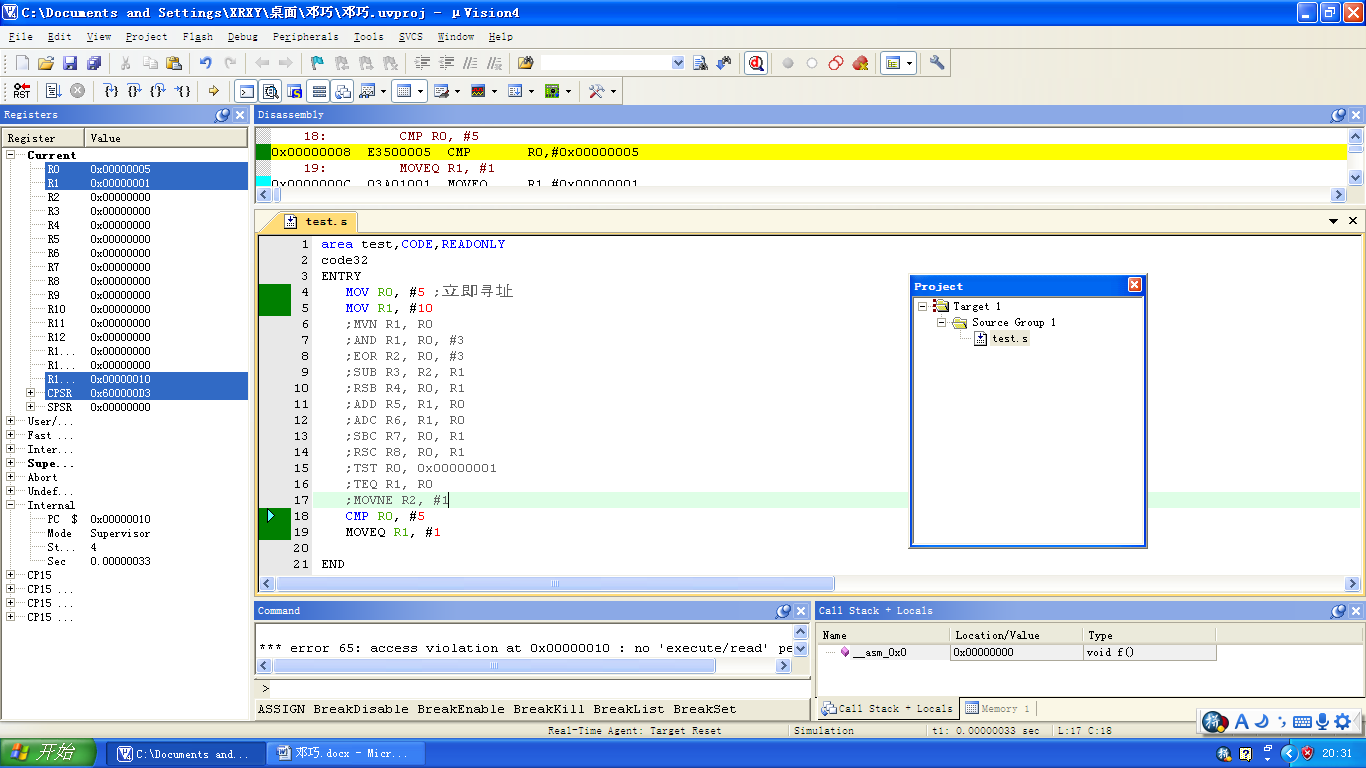
TST：



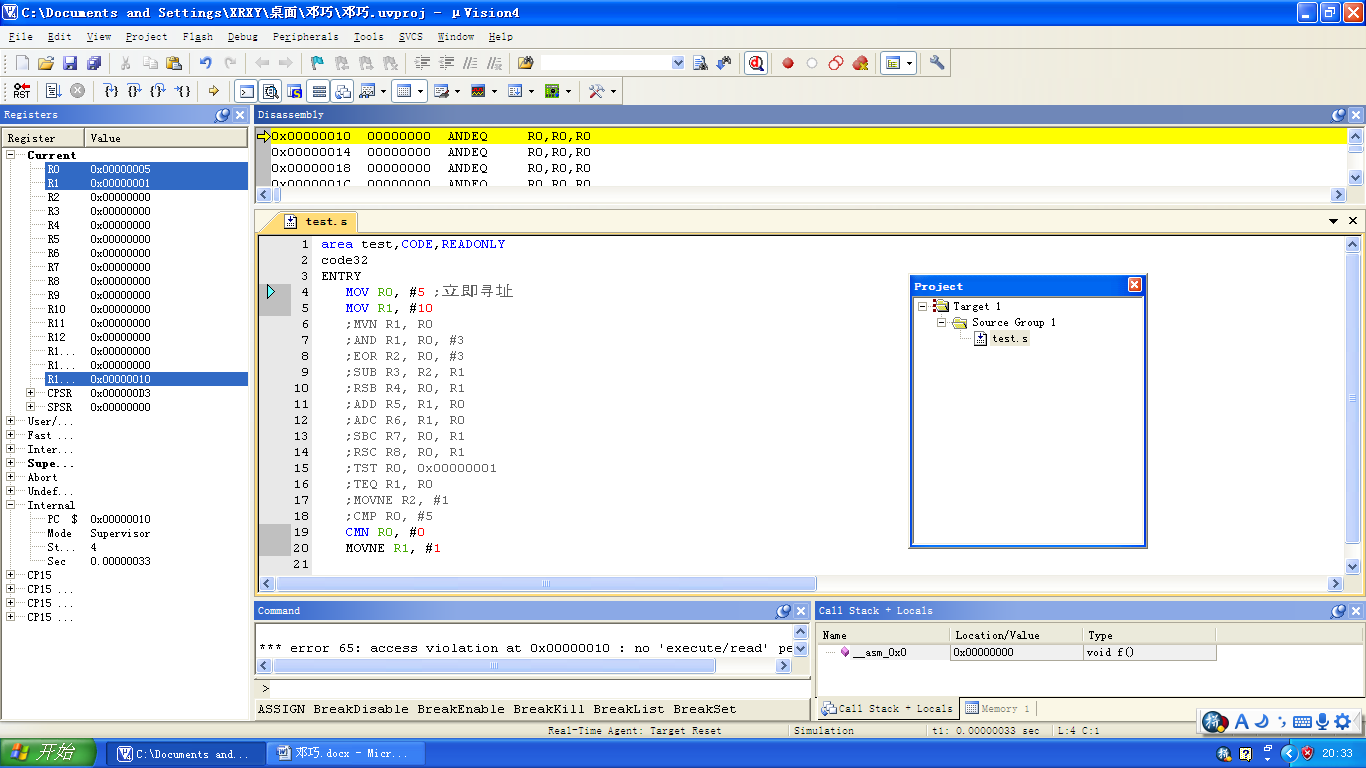
TEQ：

****

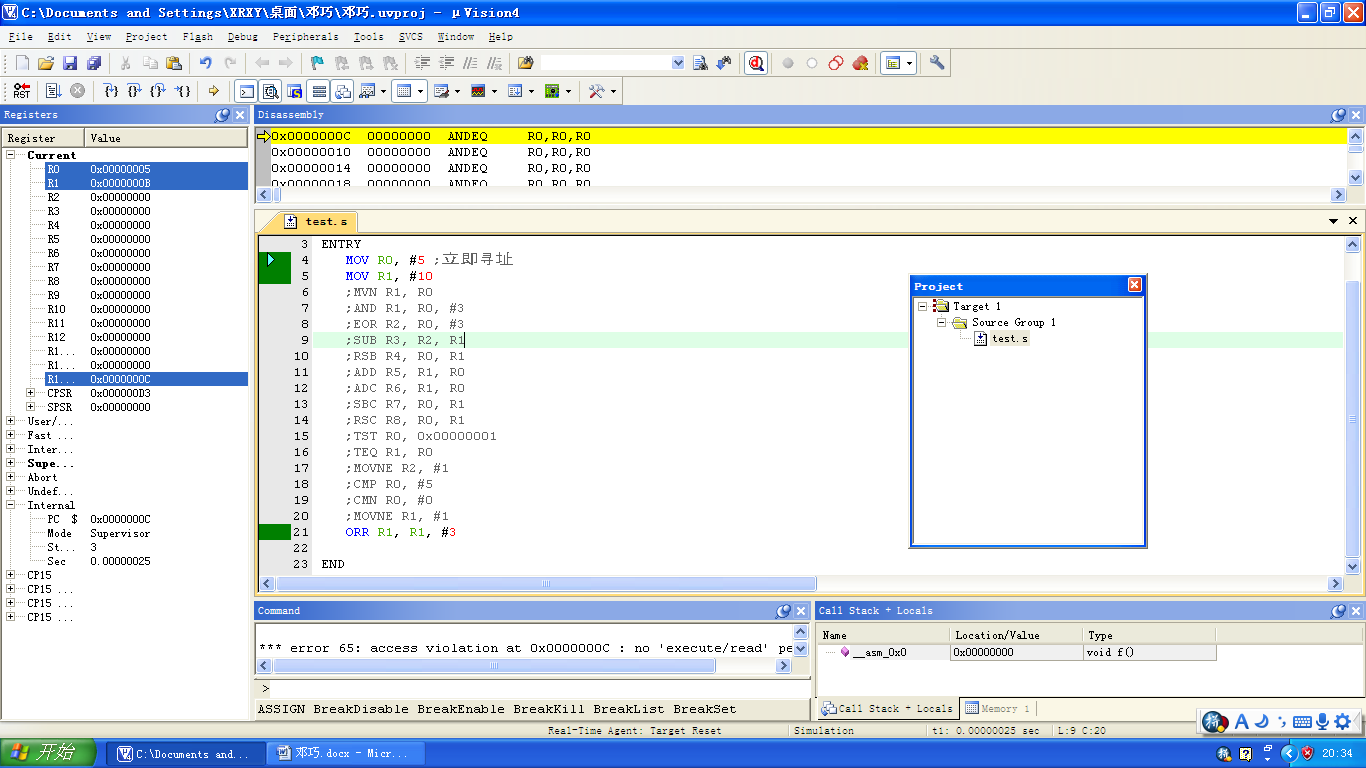
CMP：

****

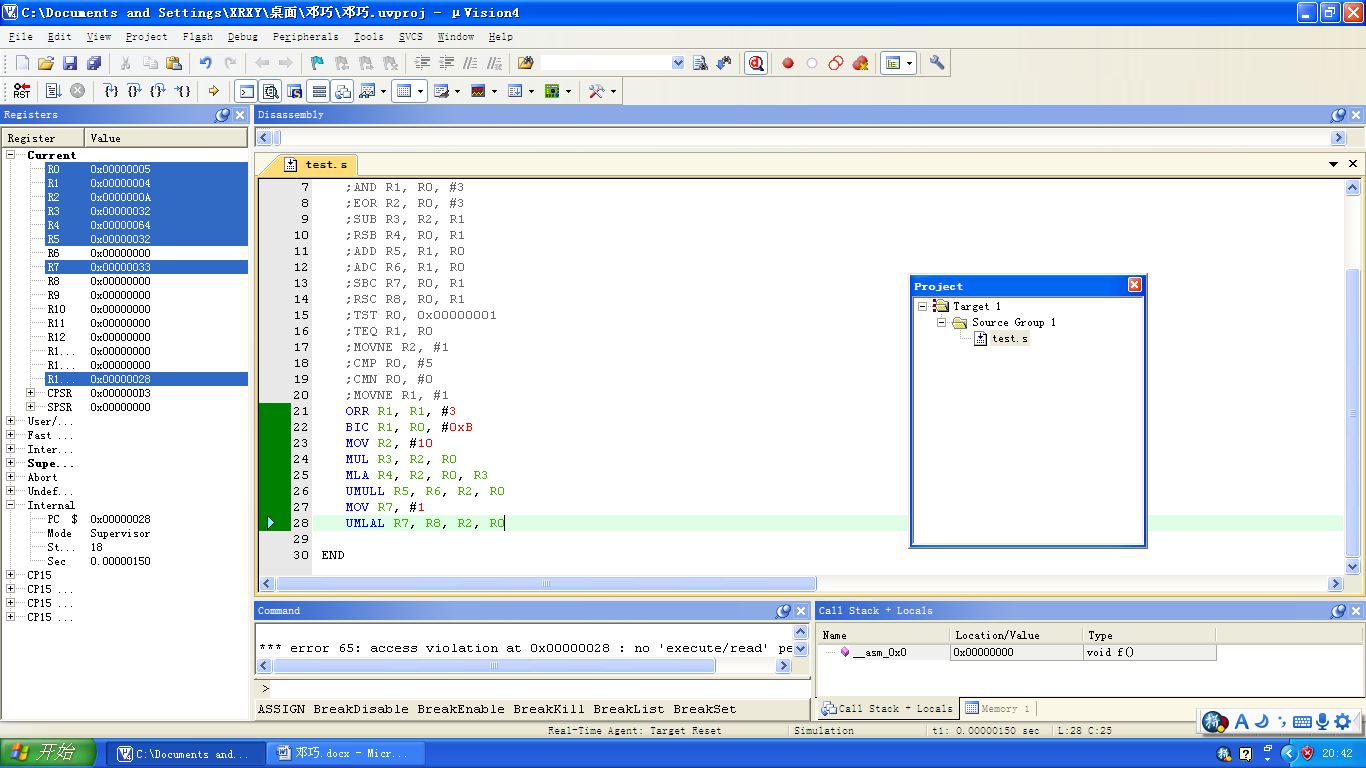
CMN：

****

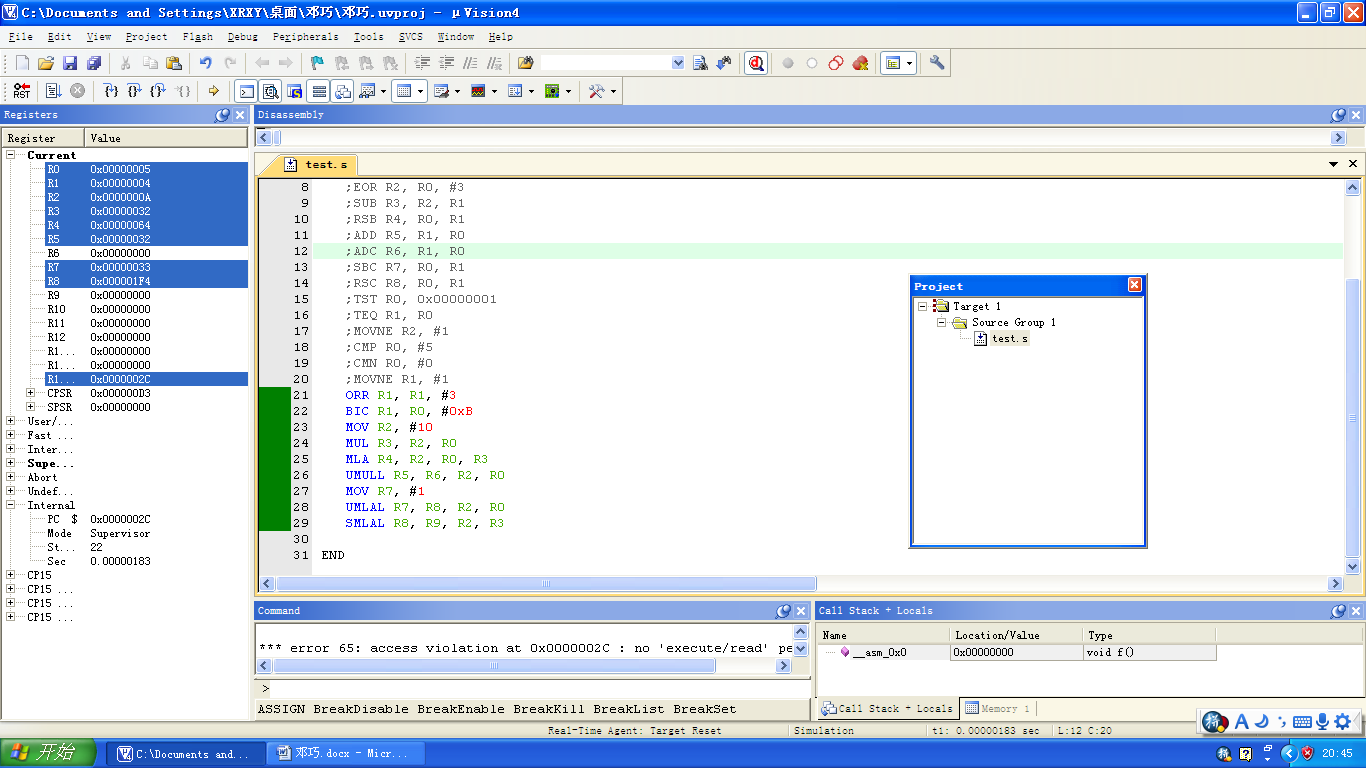
ORR：

****

BIC MUL MLA UMULL UMLAL SMULL：

****

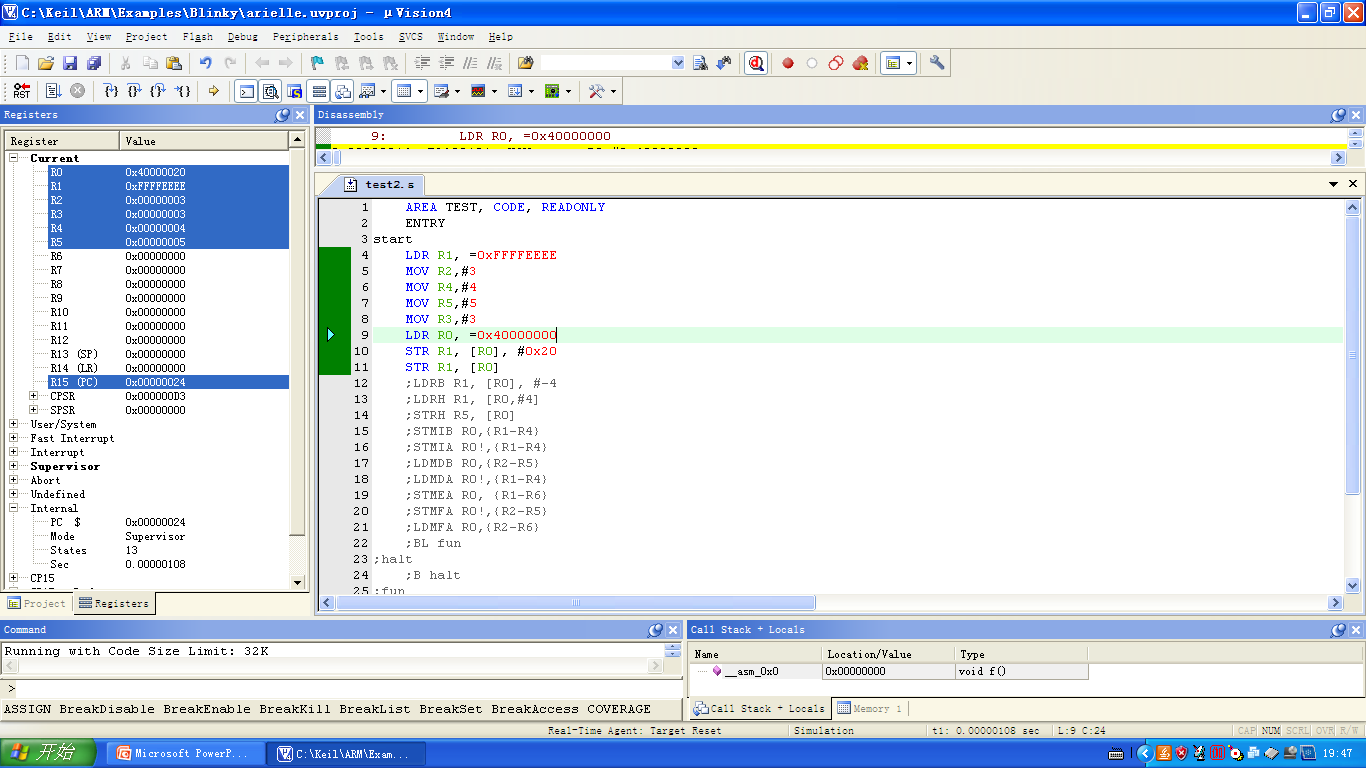
SMLAL：

****

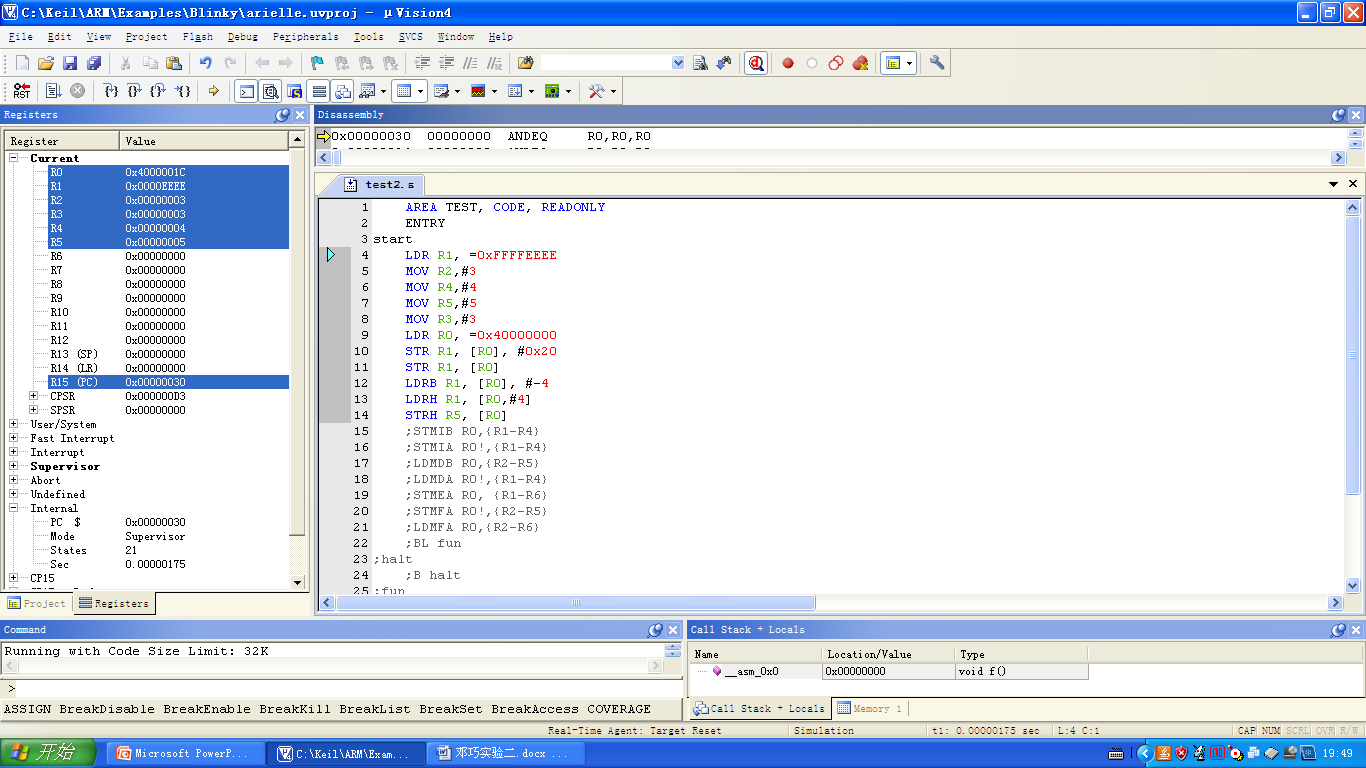
**实验二，寻址方式指令系统二**

|  |
| --- |
| AREA TEST, CODE, READONLY  ENTRY  start  LDR R1, =0xFFFFEEEE  MOV R2,#3  MOV R4,#4  MOV R5,#5  MOV R3,#3  LDR R0, =0x40000000  STR R1, [R0], #0x20  STR R1, [R0]  LDRB R1, [R0], #-4  LDRH R1, [R0,#4]  STRH R5, [R0]  STMIB R0,{R1-R4}  STMIA R0!,{R1-R4}  LDMDB R0,{R2-R5}  LDMDA R0!,{R1-R4}  STMEA R0, {R1-R6}  STMFA R0!,{R2-R5}  LDMFA R0,{R2-R6}  BL fun  halt  B halt  fun  MRS R3,CPSR  ADD R2,R3,R4  BX LR  END |

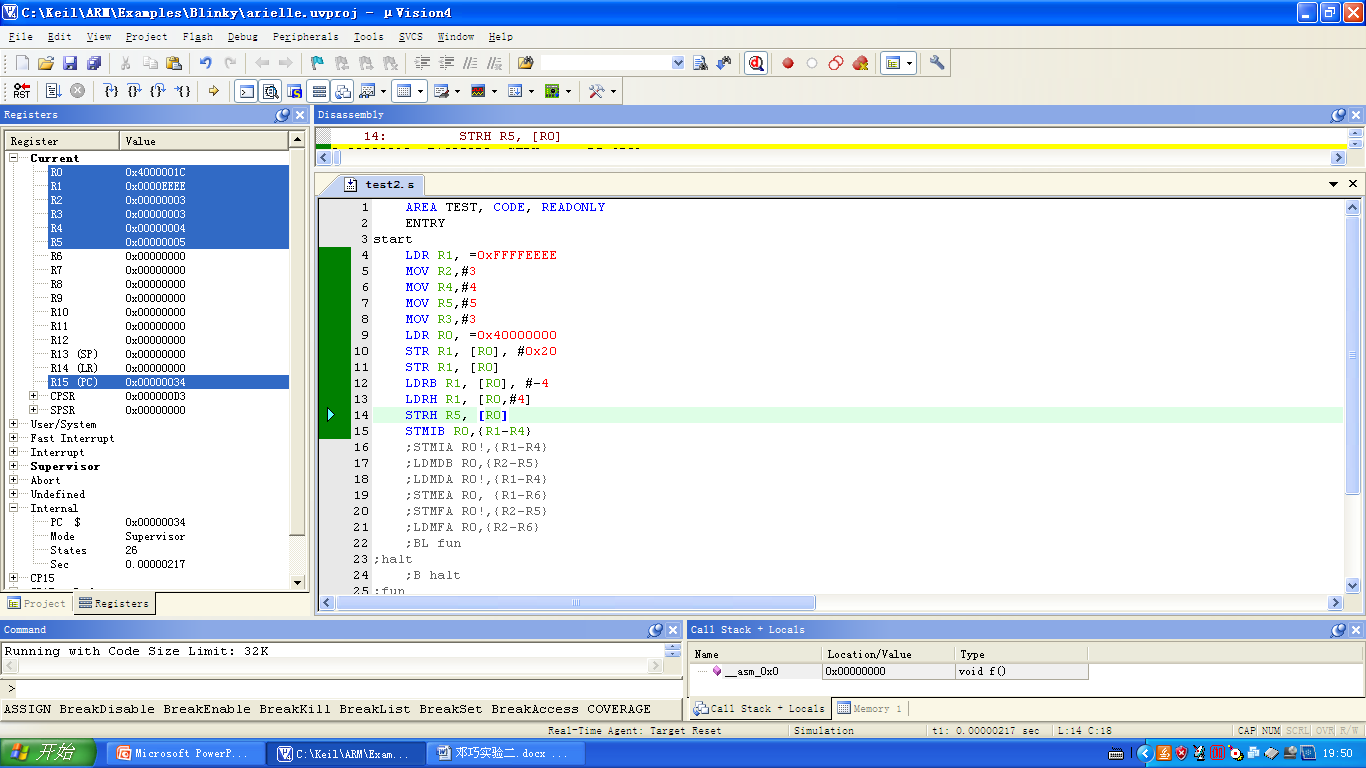
LDR STR



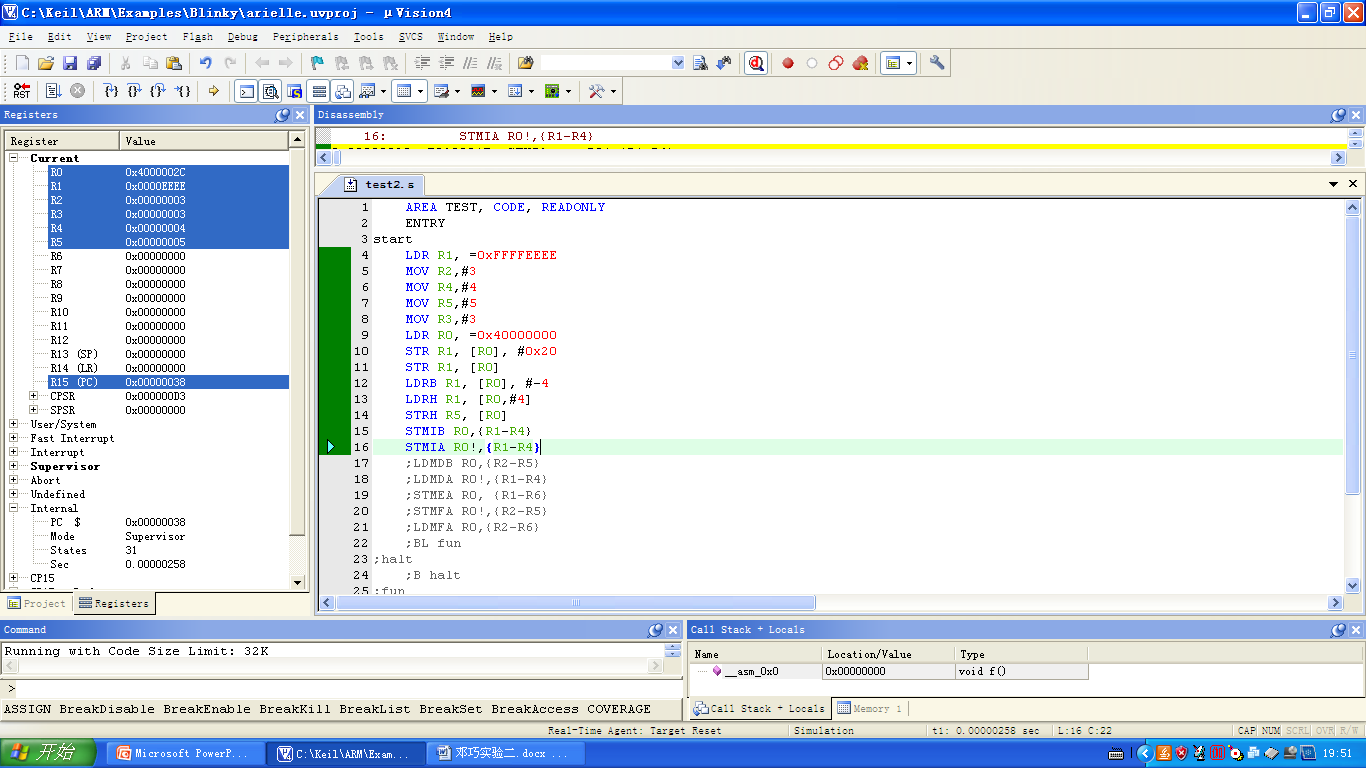
LDRB LDRH STRH



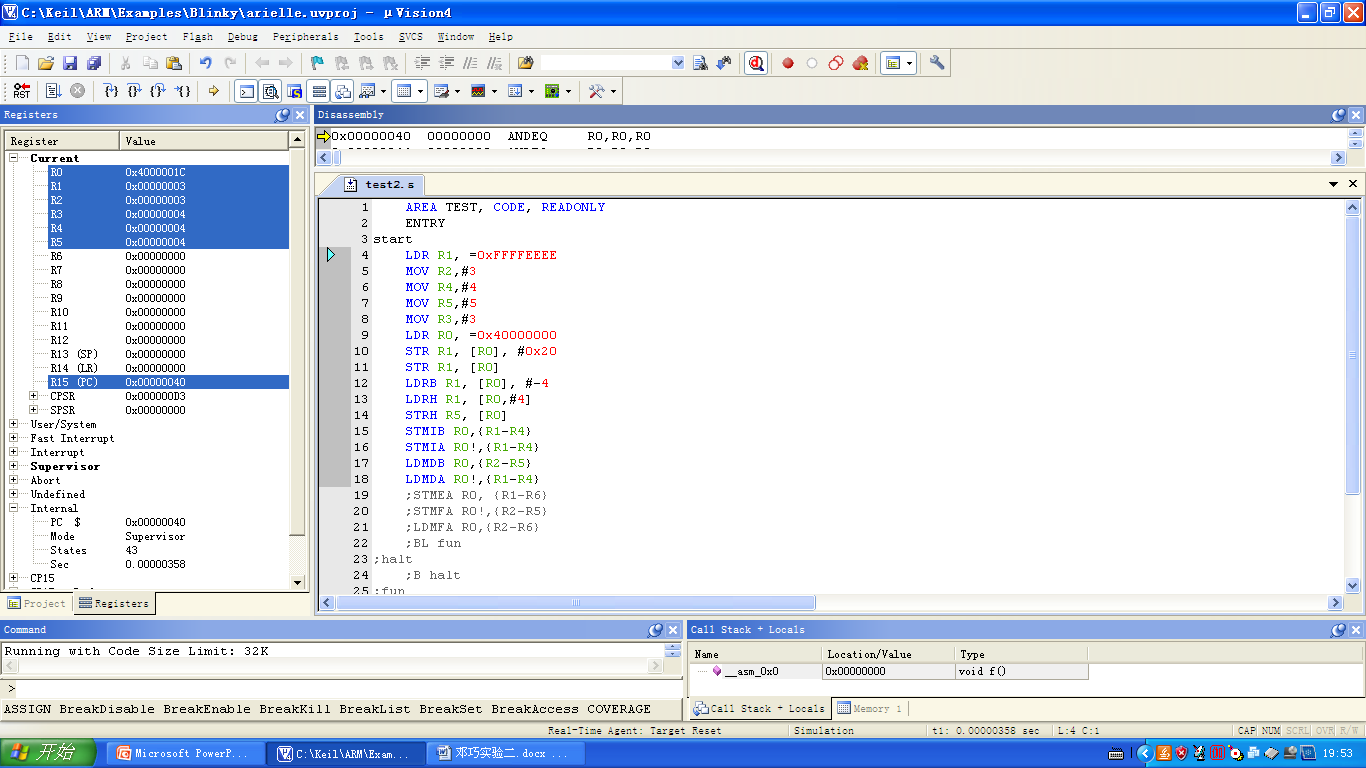
STMIB



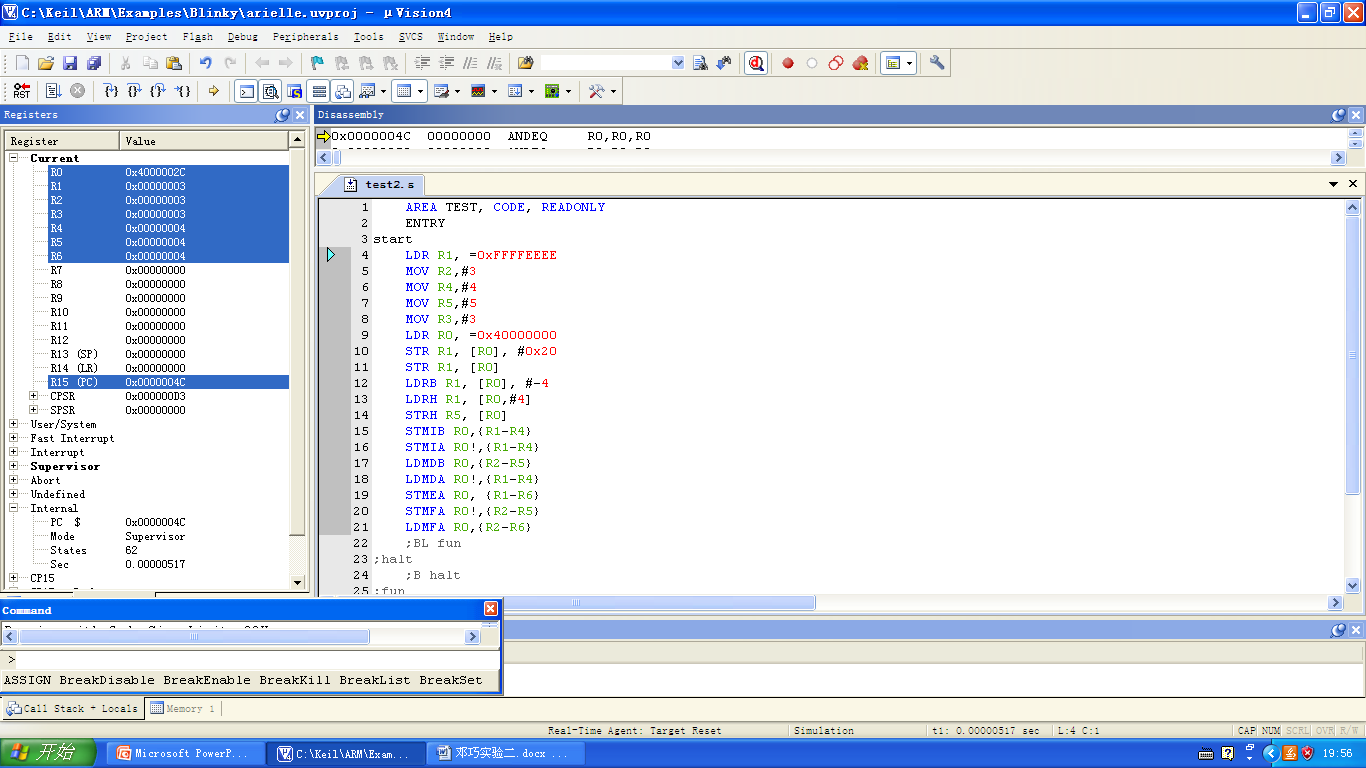
STMIA



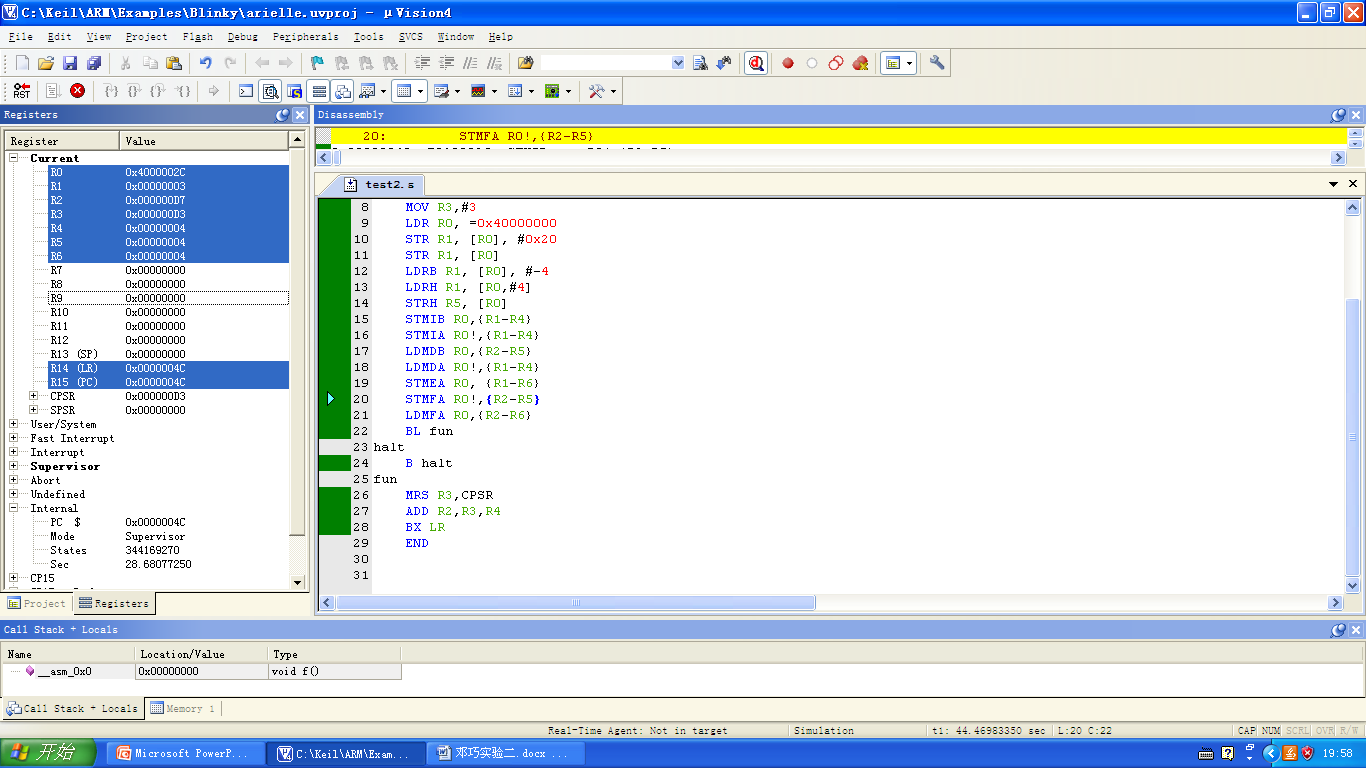
LDMDA LDMDB



STMFA



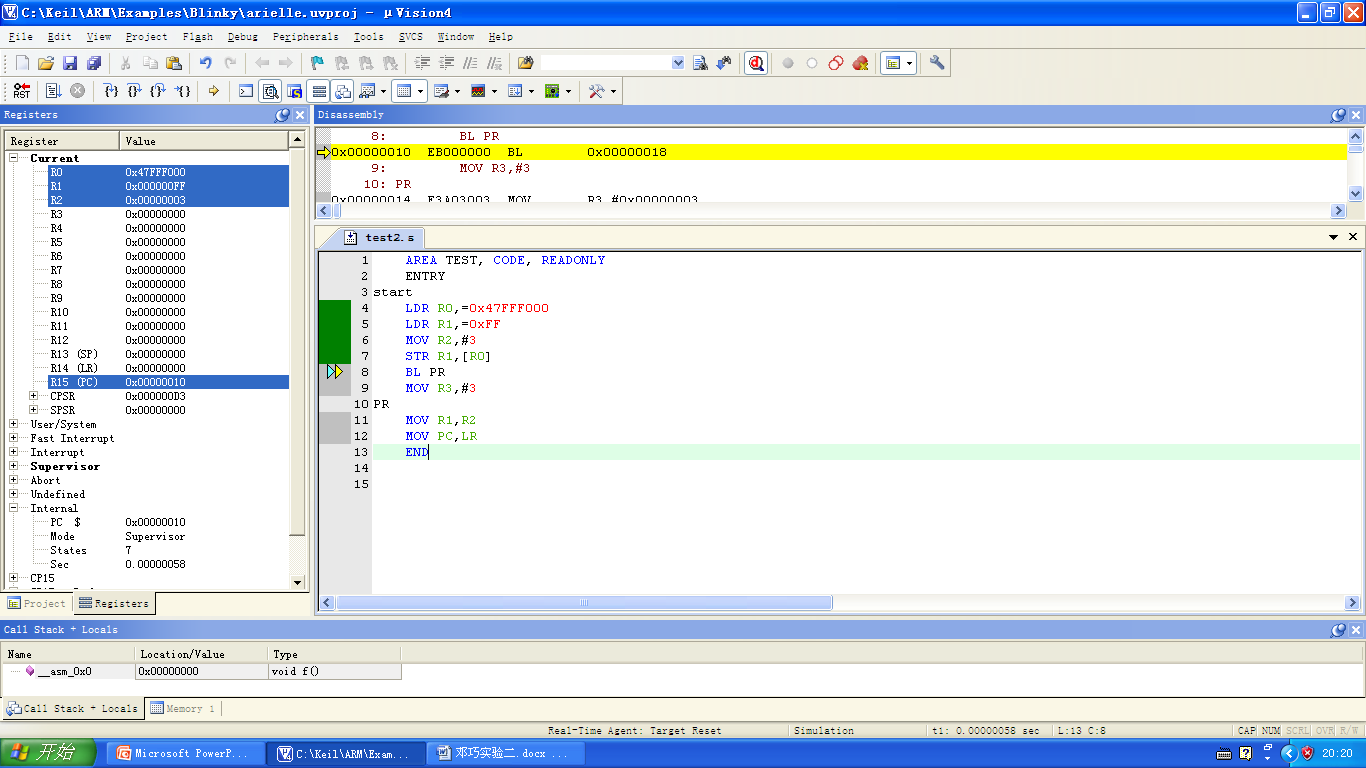
BL B MSR



**实验三，ARM主程序调用ARM/C语言子程序**

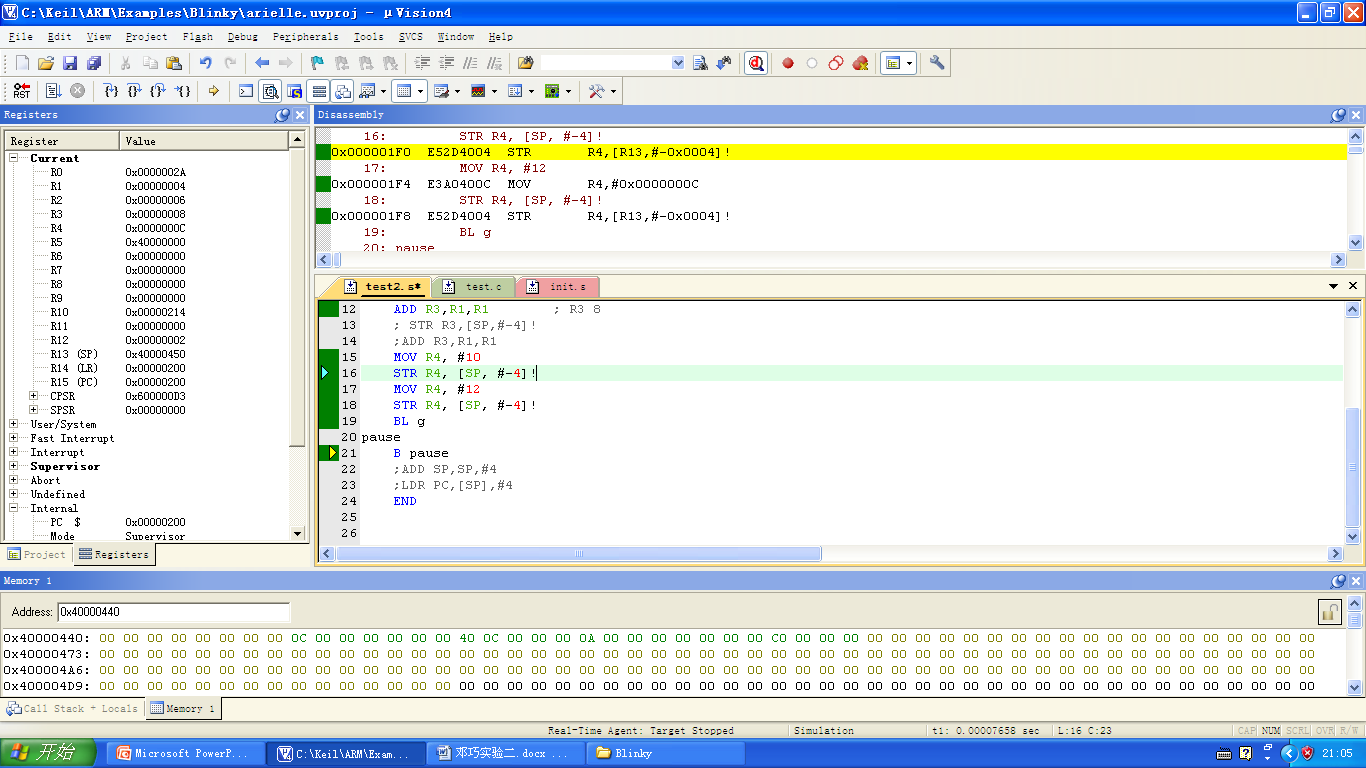
1.arm 调用子程序

|  |
| --- |
| **Arm 调用子程序**  AREA TEST, CODE, READONLY  ENTRY  start  LDR R0,=0x47FFF000  LDR R1,=0xFF  MOV R2,#3  STR R1,[R0]  BL PR  MOV R3,#3  PR  MOV R1,R2  MOV PC,LR  END |



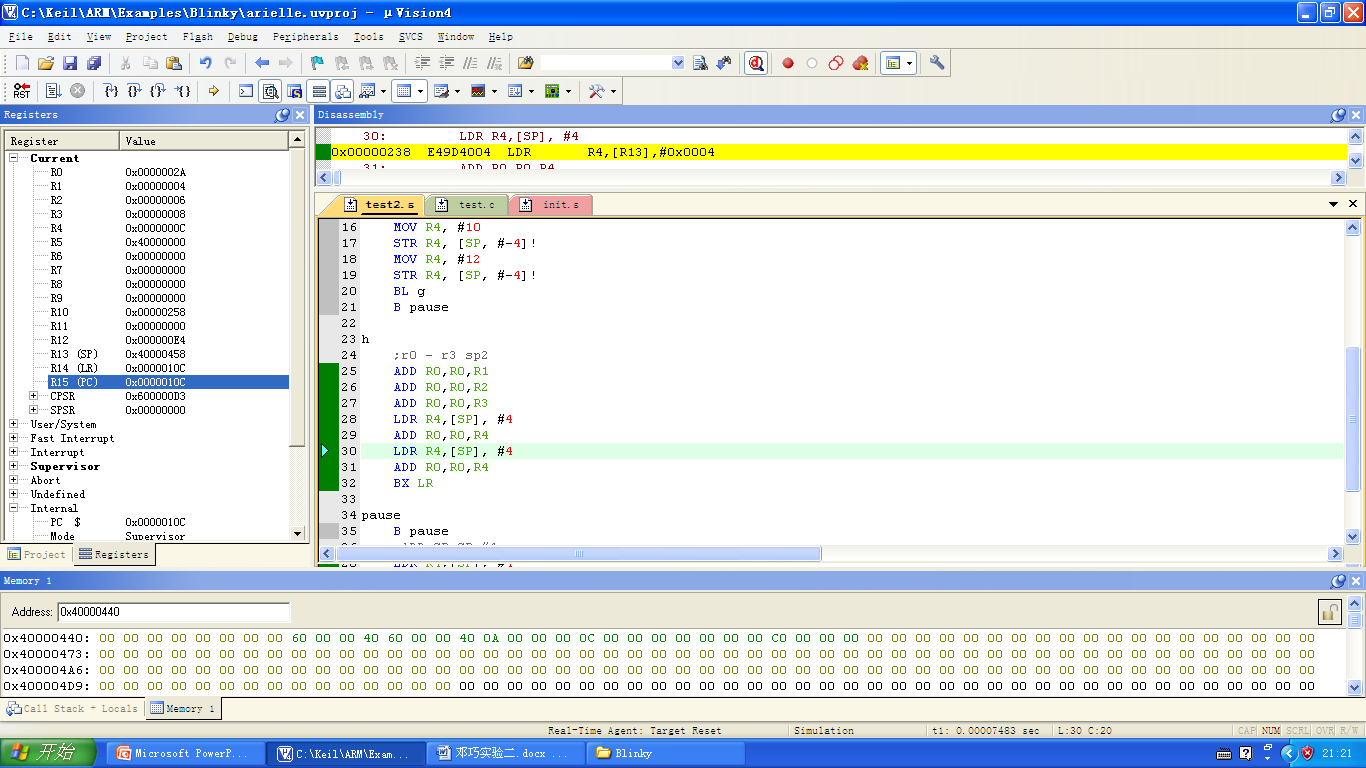
2.arm 调用C

|  |
| --- |
| **ARM调用C**  PRESERVE8  EXPORT f  AREA TEST, CODE, READONLY  code32  IMPORT g  ENTRY  f  MOV R0,#2 ; R0 2  ADD R1,R0,R0 ; R1 4  ADD R2,R1,R0 ; R2 6  ADD R3,R1,R1 ; R3 8  MOV R4, #10  STR R4, [SP, #-4]!  MOV R4, #12  STR R4, [SP, #-4]!  BL g  pause  B pause  END    **C程序**  int g(int a, int b, int c, int d, int e, int f);  extern void f(void);  int main(){  f();  return 0;  }  int g(int a,int b,int c,int d,int e,int f){  return a + b + c + d + e + f;  } |

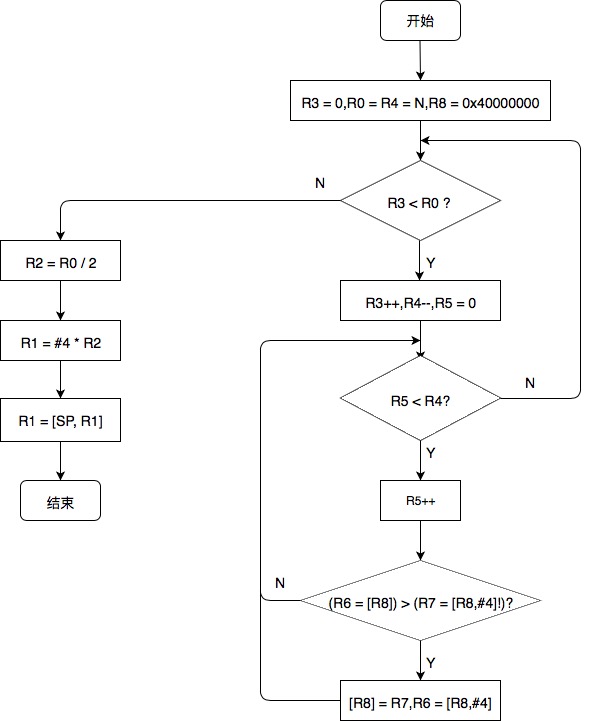


**实验四，C语言主程序调用ARM子程序**

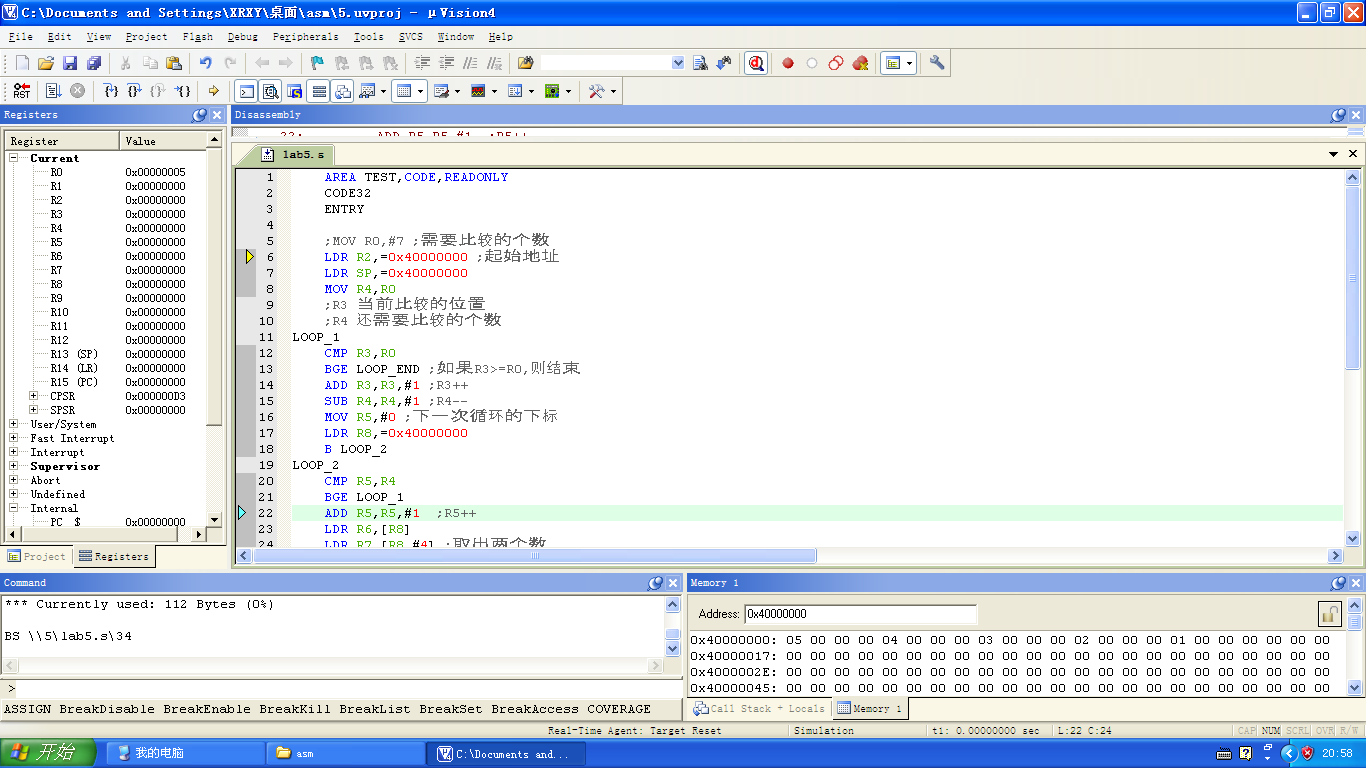
|  |
| --- |
| PRESERVE8  EXPORT f  EXPORT h  AREA TEST, CODE, READONLY  code32  IMPORT g  ENTRY  f  MOV R0,#2 ; R0 2  ; STR LR,[SP,#-4]!  ADD R1,R0,R0 ; R1 4  ADD R2,R1,R0 ; R2 6  ADD R3,R1,R1 ; R3 8  ; STR R3,[SP,#-4]!  ;ADD R3,R1,R1  MOV R4, #10  STR R4, [SP, #-4]!  MOV R4, #12  STR R4, [SP, #-4]!  BL g  B pause  h  ;r0 - r3 sp2  ADD R0,R0,R1  ADD R0,R0,R2  ADD R0,R0,R3  LDR R4,[SP], #4  ADD R0,R0,R4  LDR R4,[SP], #4  ADD R0,R0,R4  BX LR  pause  B pause  ;ADD SP,SP,#4  ;LDR PC,[SP],#4  END  **C程序：**  int g(int a, int b, int c, int d, int e, int f);  extern void f(void);  extern int h(int a, int b, int c, int d, int e, int f);  int main(){  h(2,4,6,8,10,12);  return 0;  }  int g(int a,int b,int c,int d,int e,int f){  return a + b + c + d + e + f;  } |

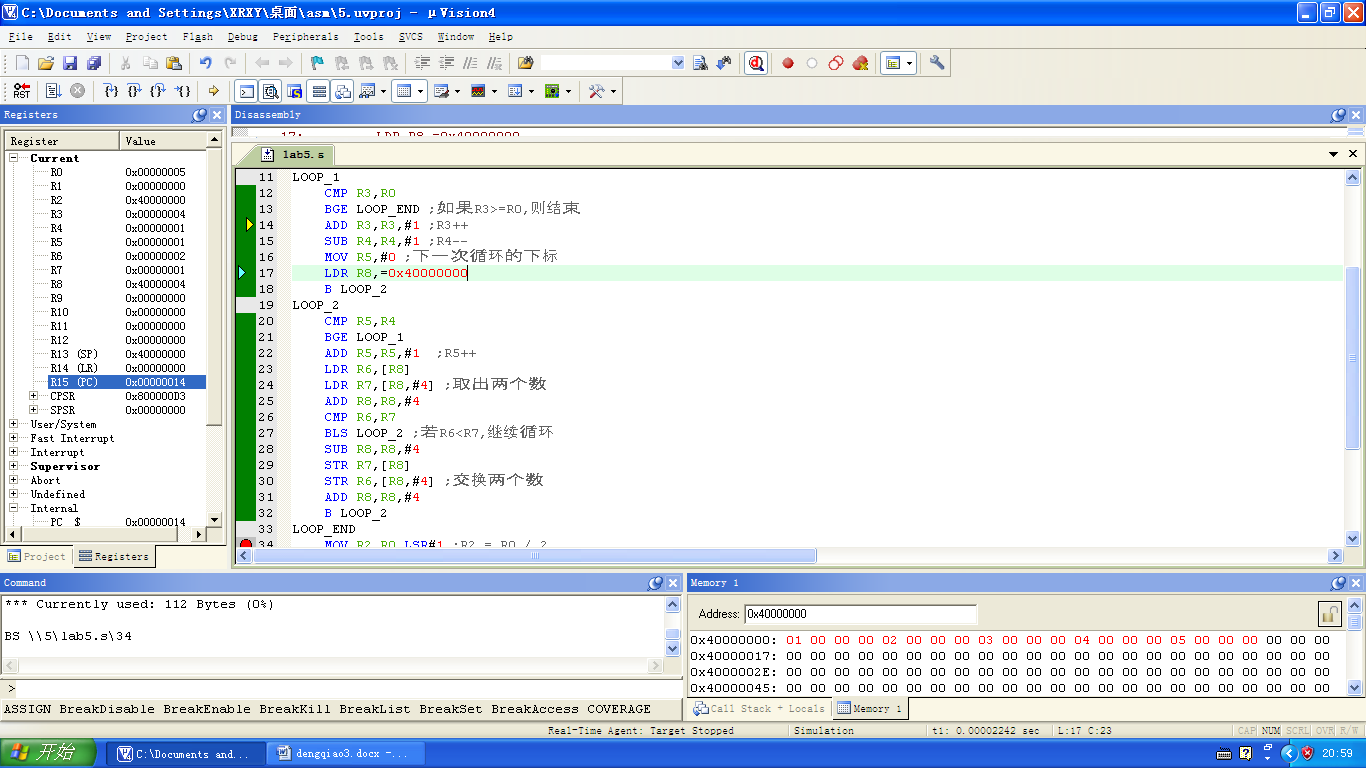


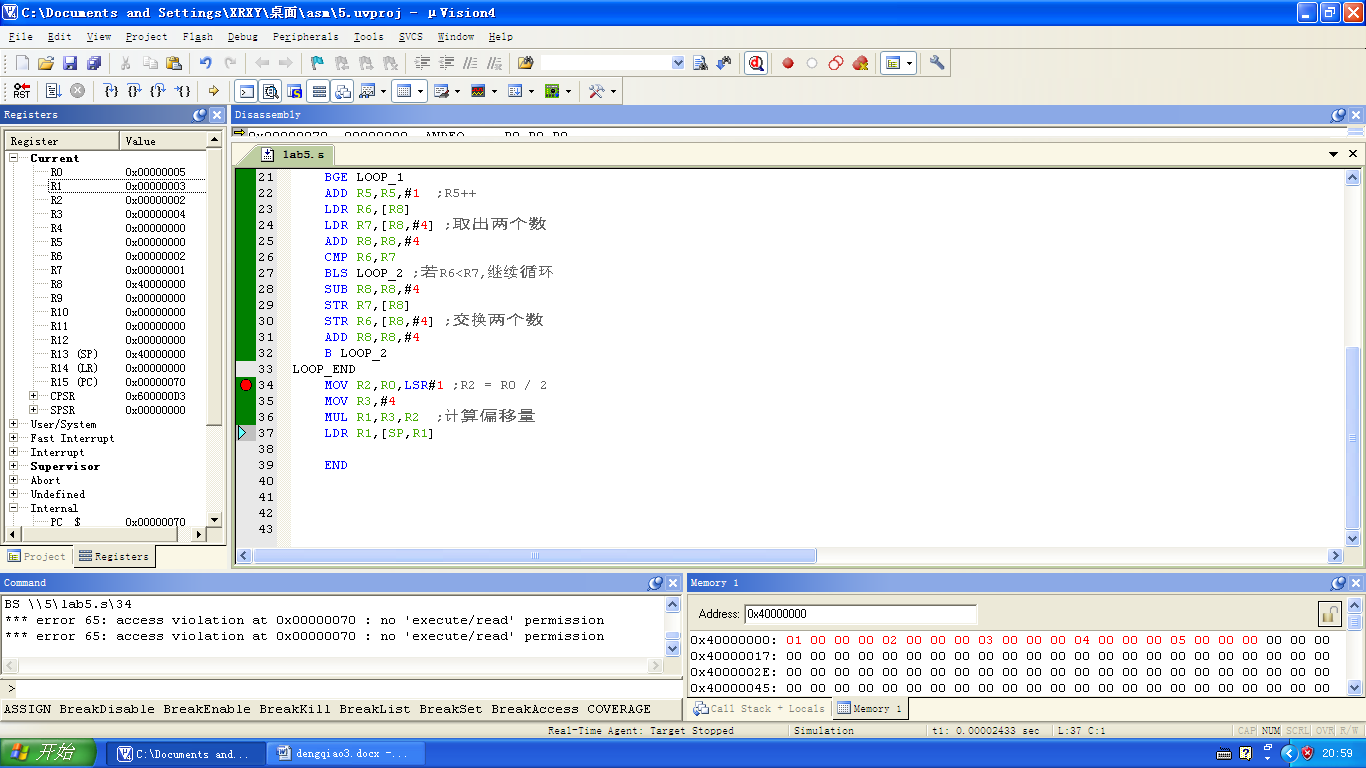
**实验五，中值滤波及编程实验**

****

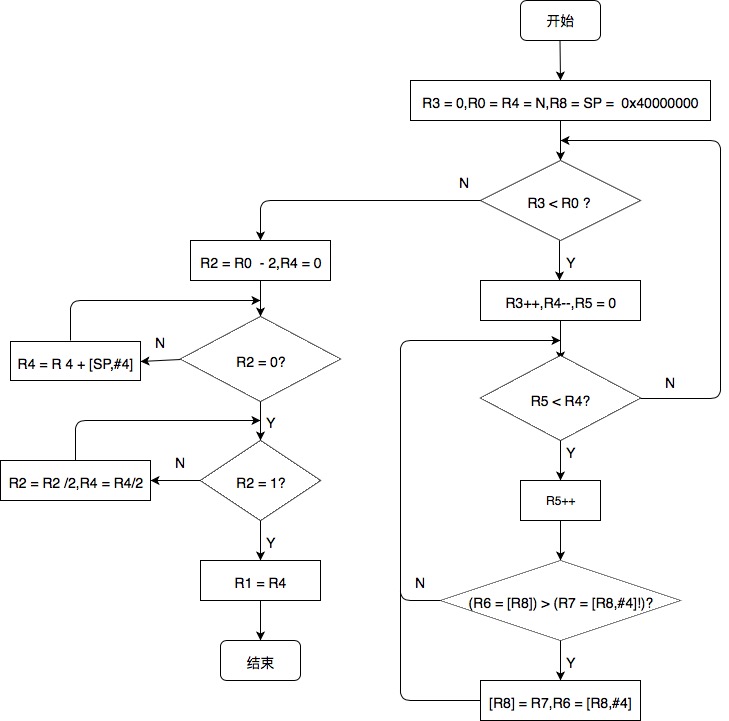
|  |
| --- |
| AREA TEST,CODE,READONLY  CODE32  ENTRY    LDR R2,=0x40000000  LDR SP,=0x40000000  MOV R4,R0    LOOP\_1  CMP R3,R0  BGE LOOP\_END  ADD R3,R3,#1 ;R3++  SUB R4,R4,#1 ;R4--  MOV R5,#0  LDR R8,=0x40000000  B LOOP\_2  LOOP\_2  CMP R5,R4  BGE LOOP\_1  ADD R5,R5,#1 ;R5++  LDR R6,[R8]  LDR R7,[R8,#4]  ADD R8,R8,#4  CMP R6,R7  BLS LOOP\_2  SUB R8,R8,#4  STR R7,[R8]  STR R6,[R8,#4]  ADD R8,R8,#4  B LOOP\_2  LOOP\_END  MOV R2,R0,LSR#1 ;R2 = R0 / 2  MOV R3,#4  MUL R1,R3,R2  LDR R1,[SP,R1]    END |



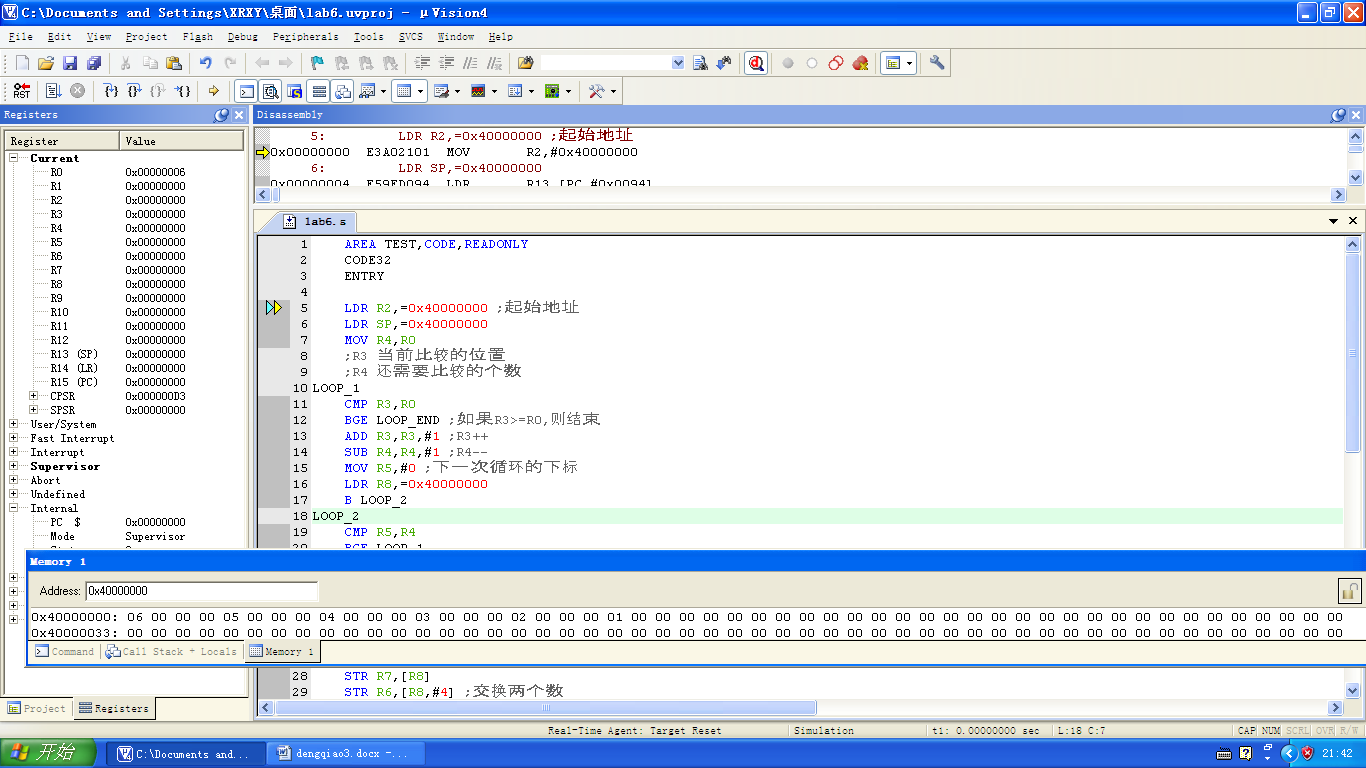
****

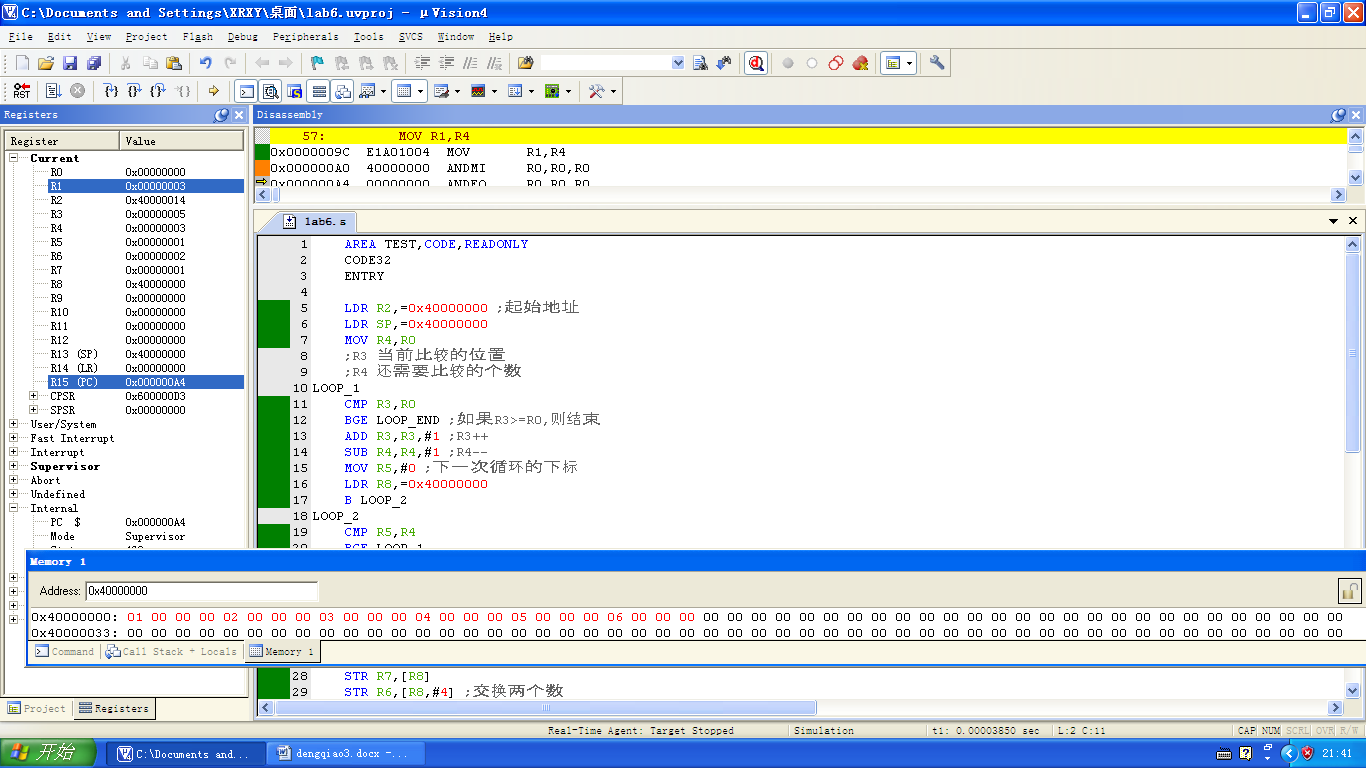
****

**实验六，均值滤波及编程实验**

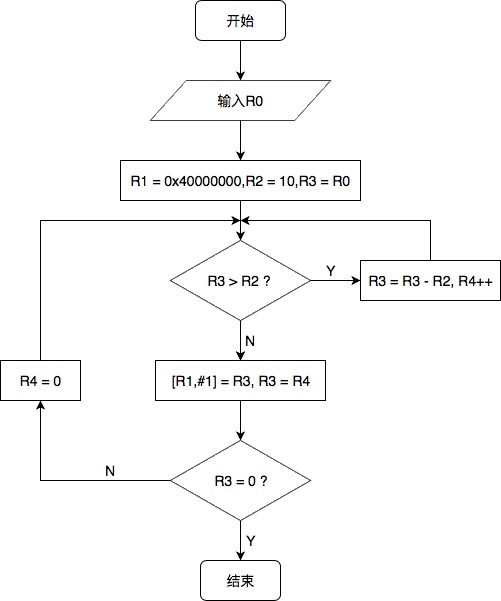
****

|  |
| --- |
| AREA TEST,CODE,READONLY  CODE32  ENTRY    LDR R2,=0x40000000 ;ÆðÊ¼µØÖ·  LDR SP,=0x40000000  MOV R4,R0  ;R3 µ±Ç°±È½ÏµÄÎ»ÖÃ  ;R4 »¹ÐèÒª±È½ÏµÄ¸öÊý  LOOP\_1  CMP R3,R0  BGE LOOP\_END ;Èç¹ûR3>=R0,Ôò½áÊø  ADD R3,R3,#1 ;R3++  SUB R4,R4,#1 ;R4--  MOV R5,#0 ;ÏÂÒ»´ÎÑ­»·µÄÏÂ±ê  LDR R8,=0x40000000  B LOOP\_2  LOOP\_2  CMP R5,R4  BGE LOOP\_1  ADD R5,R5,#1 ;R5++  LDR R6,[R8]  LDR R7,[R8,#4] ;È¡³öÁ½¸öÊý  ADD R8,R8,#4  CMP R6,R7  BLS LOOP\_2 ;ÈôR6<R7,¼ÌÐøÑ­»·  SUB R8,R8,#4  STR R7,[R8]  STR R6,[R8,#4] ;½»»»Á½¸öÊý  ADD R8,R8,#4  B LOOP\_2  LOOP\_END  ;MOV R2,R0,LSR#1 ;R2 = R0 / 2  ;MOV R3,#4  ;MUL R1,R3,R2 ;¼ÆËãÆ«ÒÆÁ¿ã  ;LDR R1,[SP,R1]  SUB R0,R0,#2  MOV R5,R0  ADD R2,R2,#4  MOV R4,#0  B LOOP\_ADD  LOOP\_ADD  LDR R3,[R2]  ADD R2,#4  ADD R4,R4,R3  SUB R0,R0,#1  CMP R0,#0  BNE LOOP\_ADD  B LOOP\_DIV  LOOP\_DIV  MOV R5,R5,LSR#1  MOV R4,R4,LSR#1  CMP R5,#1  BGT LOOP\_DIV  MOV R1,R4    END |

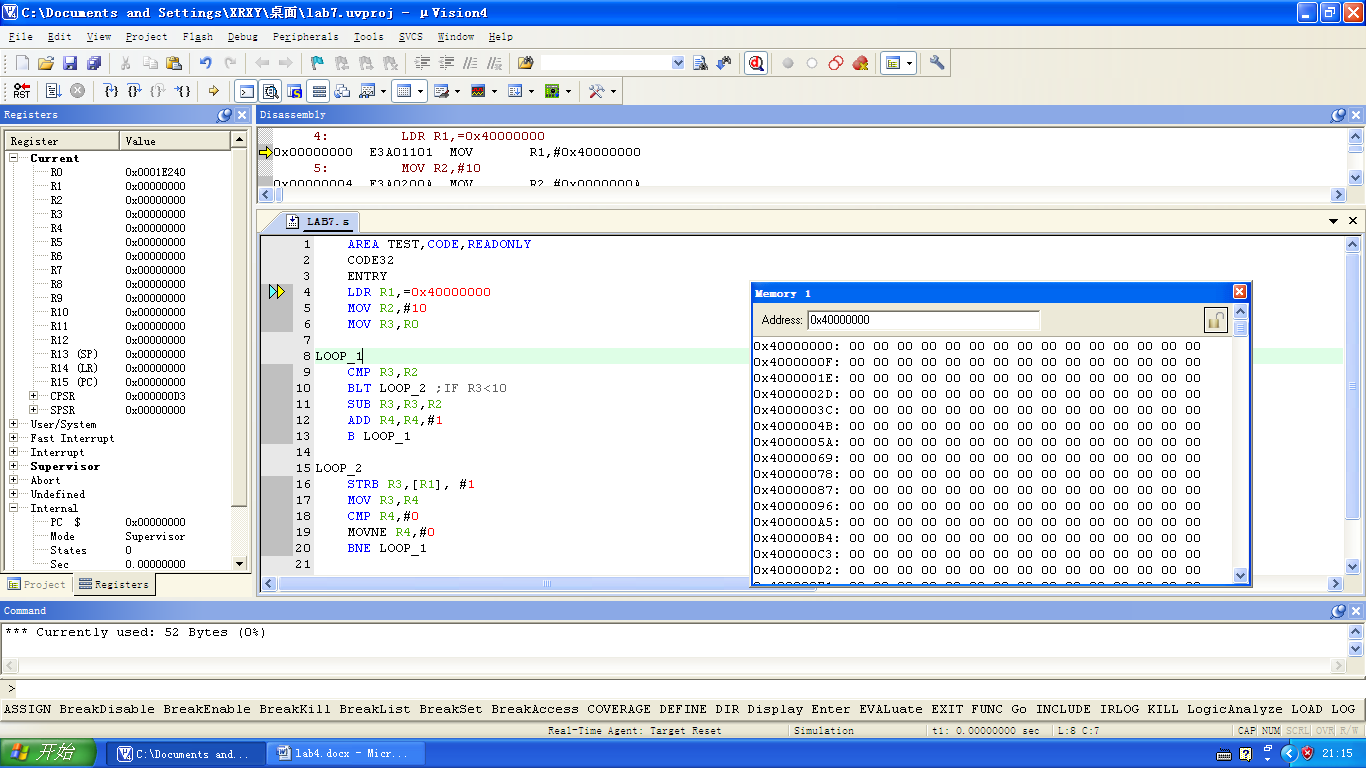
****

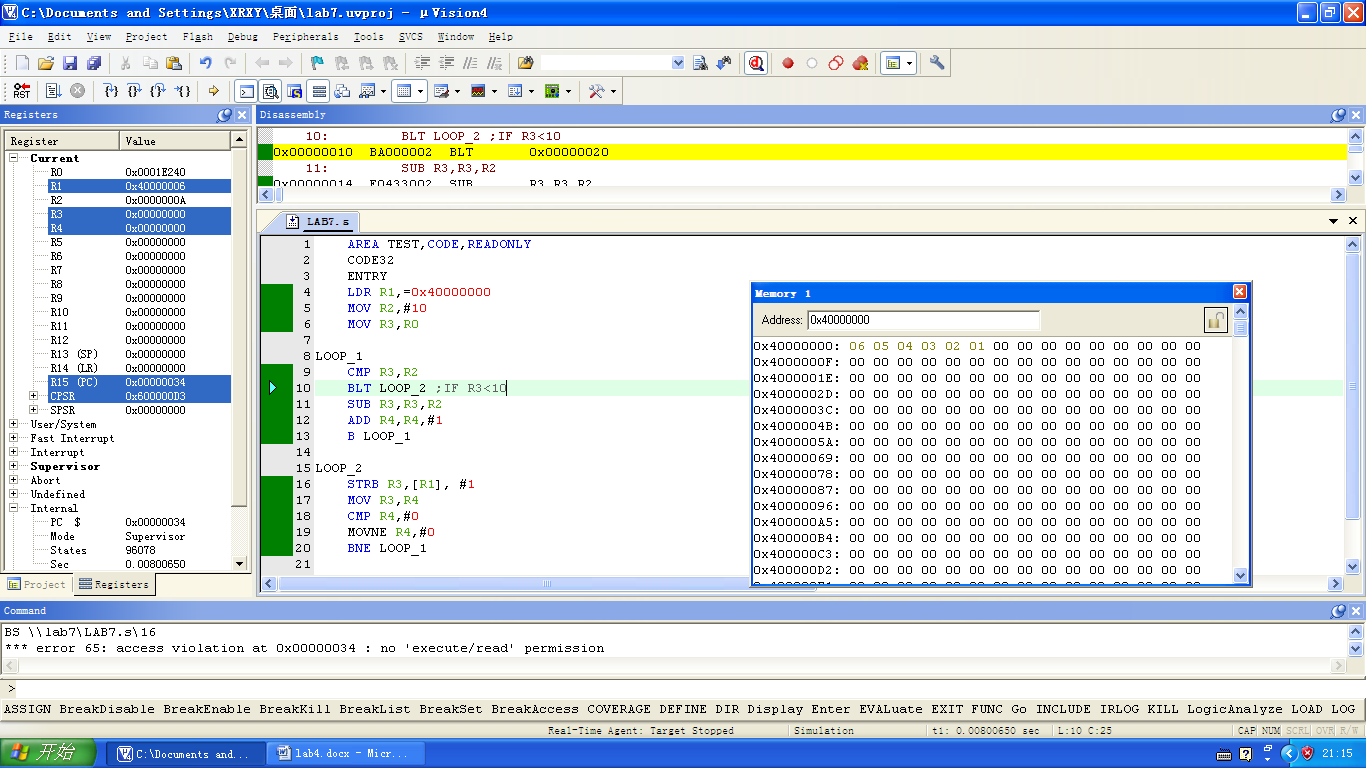
****

**实验七，数值变换及编程实验**

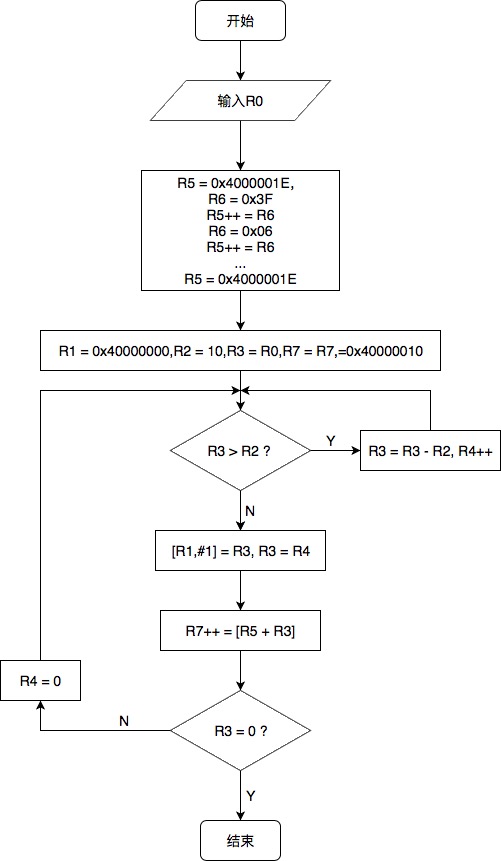
****

|  |
| --- |
| AREA TEST,CODE,READONLY  CODE32  ENTRY  LDR R1,=0x40000000  MOV R2,#10  MOV R3,R0    LOOP\_1  CMP R3,R2  BLT LOOP\_2 ;IF R3<10  SUB R3,R3,R2  ADD R4,R4,#1  B LOOP\_1  LOOP\_2  STRB R3,[R1],#1  MOV R3,R4  CMP R4,#0  MOVNE R4,#0  BNE LOOP\_1 |

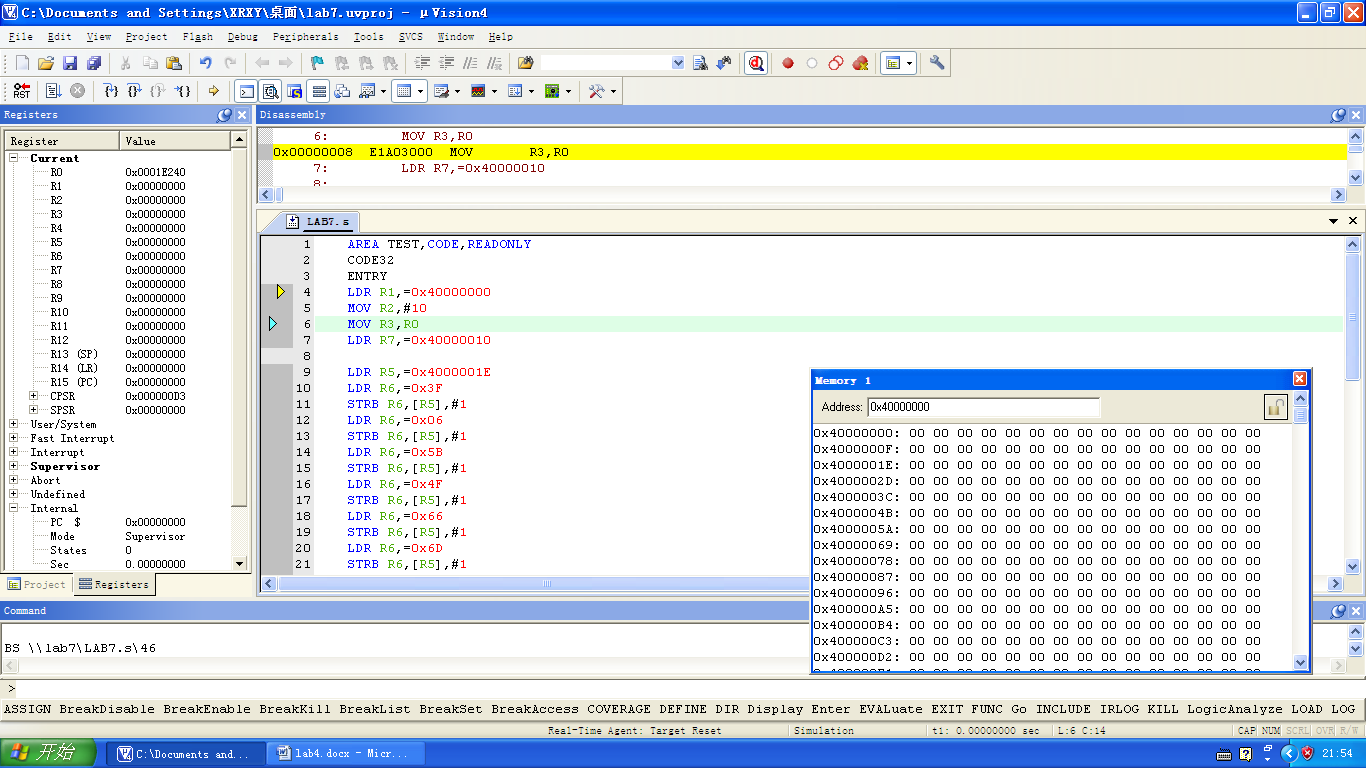


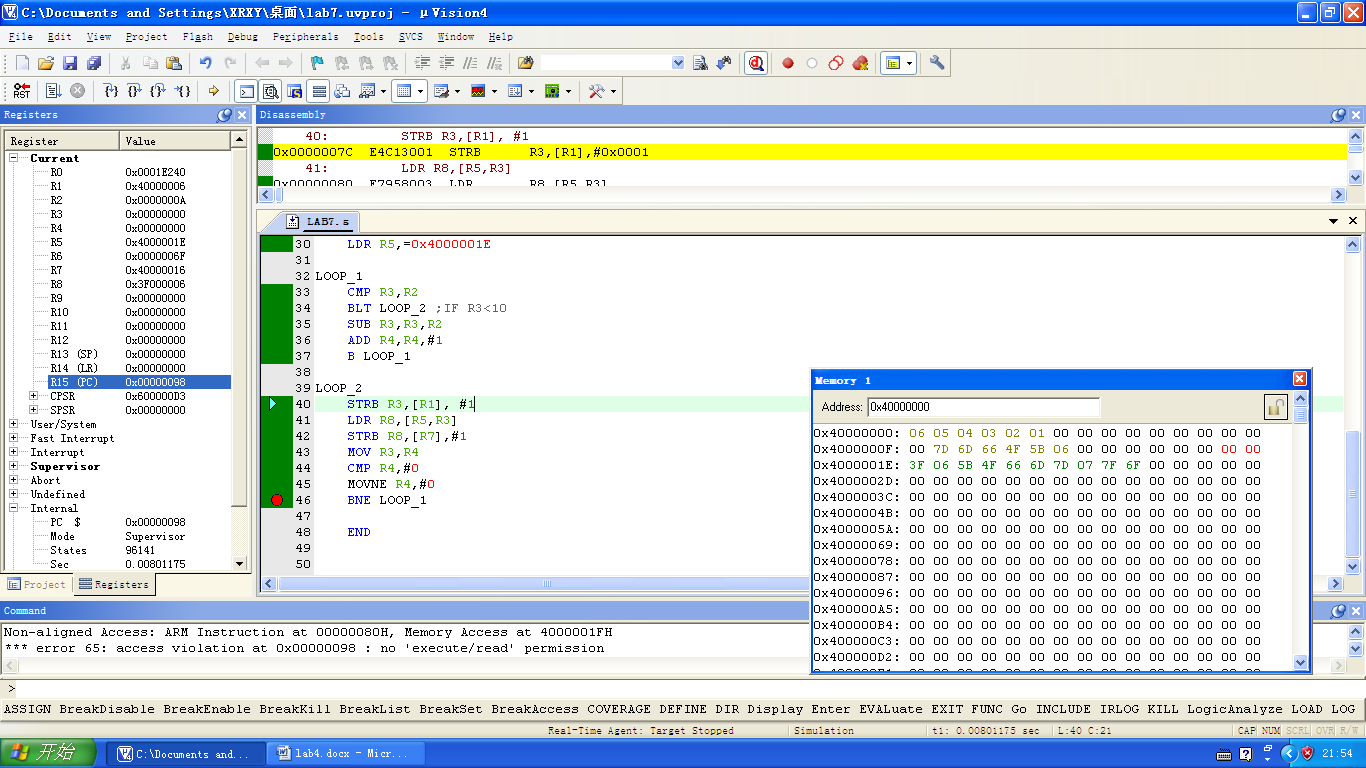


**实验八，数码变换及编程实验**

****

|  |
| --- |
| AREA TEST,CODE,READONLY  CODE32  ENTRY  LDR R1,=0x40000000  MOV R2,#10  MOV R3,R0  LDR R7,=0x40000010    LDR R5,=0x4000001E  LDR R6,=0x3F  STRB R6,[R5],#1  LDR R6,=0x06  STRB R6,[R5],#1  LDR R6,=0x5B  STRB R6,[R5],#1  LDR R6,=0x4F  STRB R6,[R5],#1  LDR R6,=0x66  STRB R6,[R5],#1  LDR R6,=0x6D  STRB R6,[R5],#1  LDR R6,=0x7D  STRB R6,[R5],#1  LDR R6,=0x07  STRB R6,[R5],#1  LDR R6,=0x7F  STRB R6,[R5],#1  LDR R6,=0x6F  STRB R6,[R5],#1  LDR R5,=0x4000001E    LOOP\_1  CMP R3,R2  BLT LOOP\_2 ;IF R3<10  SUB R3,R3,R2  ADD R4,R4,#1  B LOOP\_1  LOOP\_2  STRB R3,[R1], #1  LDR R8,[R5,R3]  STRB R8,[R7],#1  MOV R3,R4  CMP R4,#0  MOVNE R4,#0  BNE LOOP\_1    END |





**八、总结及心得体会：**

1.应该多加练习，提高对指令的熟悉度

2.勤动手

**九、对本实验过程及方法、手段的改进建议：**

无

**报告评分：**

**指导教师签字：**