

实验一：汇编指令实验

1. 实验目的

- 了解 Keil uVision 集成开发环境及软件仿真功能使用。
- 掌握 ARM7TDMI 汇编指令的用法，并能编写简单的汇编程序。
- 掌握指令的条件执行和使用 LDR/STR 指令完成存储器的访问。
- 掌握基本的汇编程序调试。

2. 实验设备

- 硬件：PC 机一台
- 软件：WindowsXP 系统，Keil uVision 4.0 集成开发环境

3. 实验内容

(1) 使用 LDR 指令读取 0x40000100 地址上的数据，将数据加 1。若结果小于 10，则使用 STR 指令把结果写回原地址；若结果大于等于 10，则把 0 写回原地址。

(2) 使用 Keil uVision 软件仿真，单步、全速运行程序，设置断点，打开寄存器窗口监视 R0 和 R1 的值，打开存储器观察窗口监视 0x40000100 地址上的值。

4. 实验预习要求

- (1) 学习 ARM 指令系统的内容，重点掌握 LDR/STR 指令和指令条件执行；
- (2) 查阅 Keil uVision 软件的介绍，了解软件的功能和操作方法。

5. 实验步骤

- (1) 启动 Keil uVision，新建一个工程 ex01。见图 1-1、图 1-2、图 1-3。

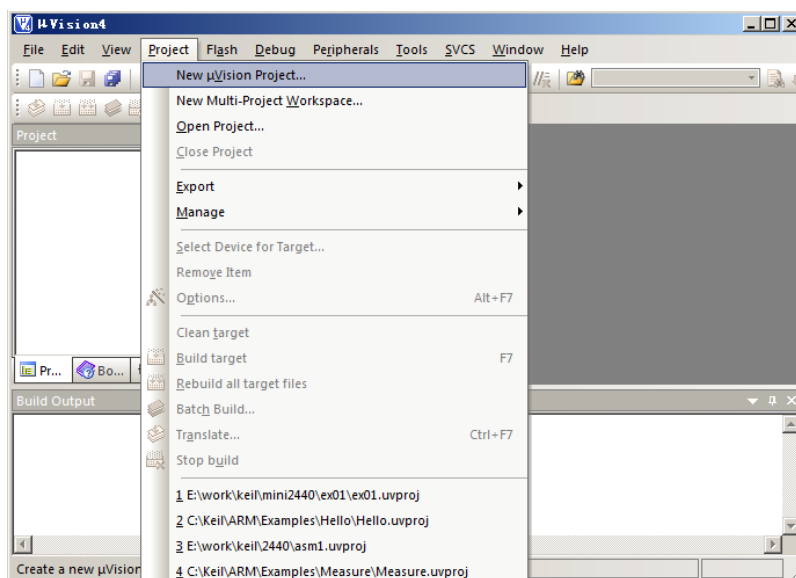


图 1-1 建立工程

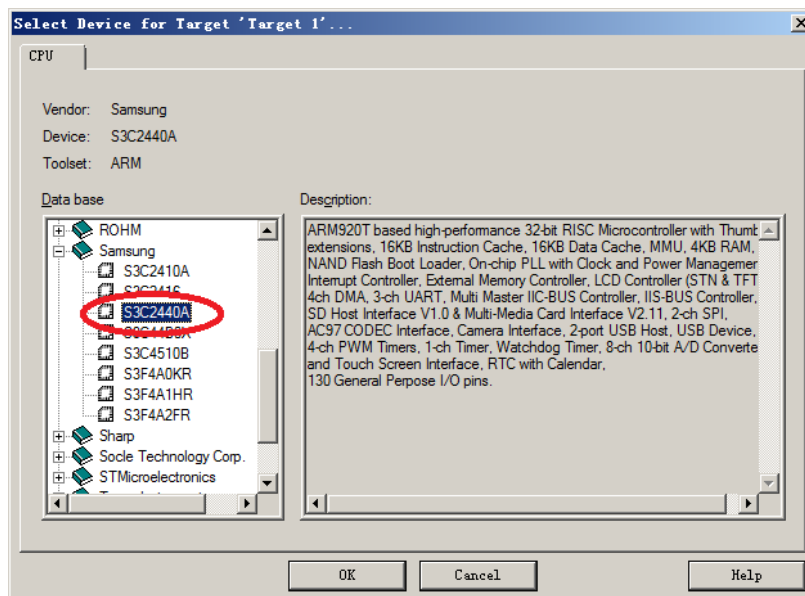


图 1-2 选择目标芯片



图 1-3 不需要系统提供的 Startup 文件

(2) 建立汇编源文件 ex01.s, 编写实验程序, 然后添加到工程中。见图 1-4、图 1-5、图 1-6、图 1-7。

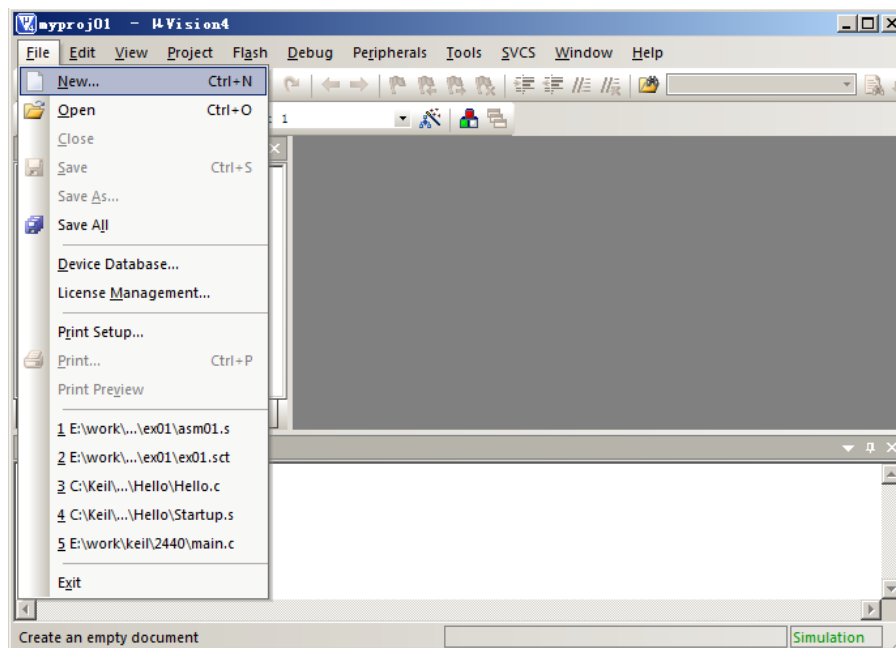


图 1-4 建立新文件

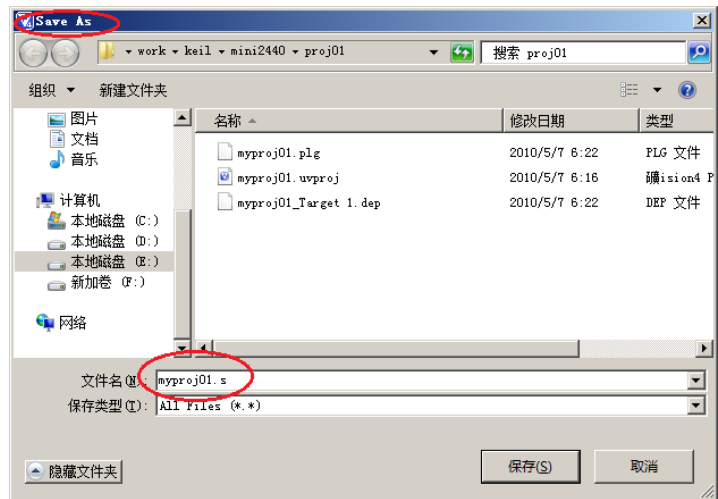


图 1-5 保存扩展名为.s 的汇编程序文件

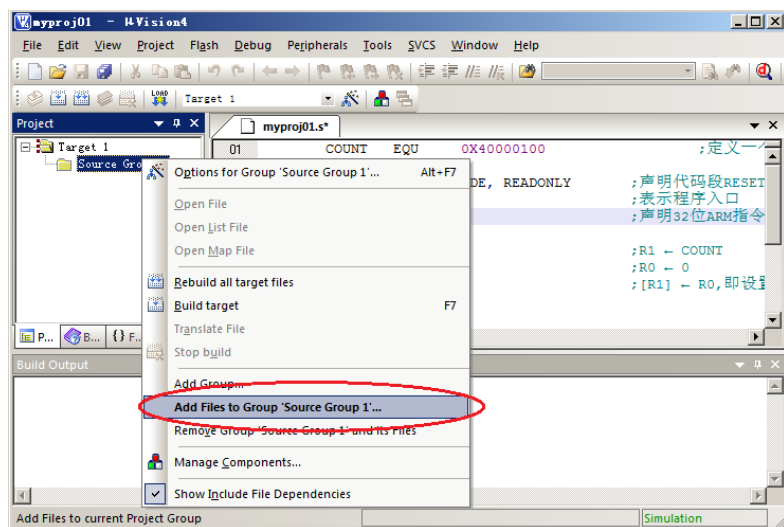


图 1-6 把文件添加到工程

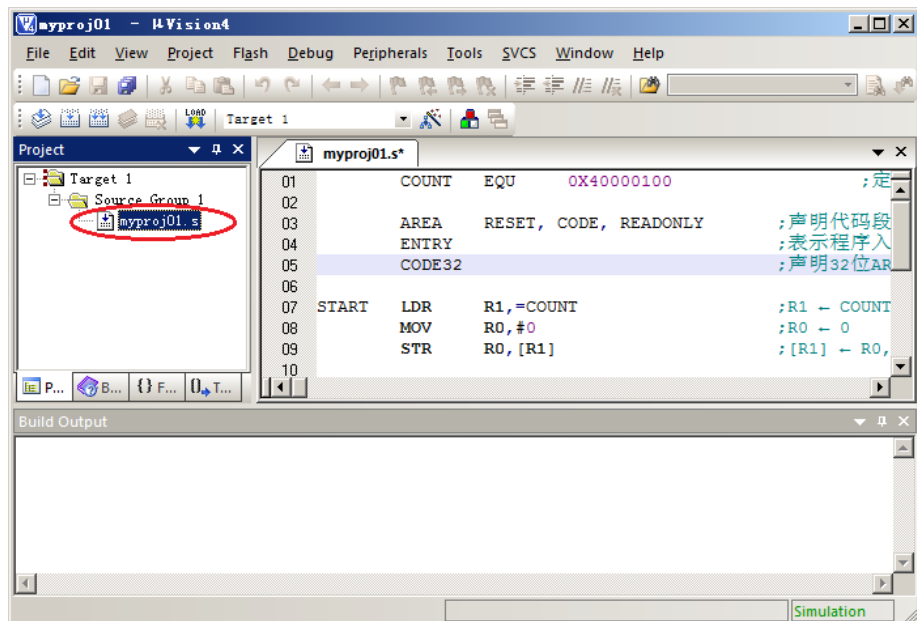


图 1-7 添加文件到工程后

(3) 设置工程选项，存储器映射。见图 1-8、图 1-9。

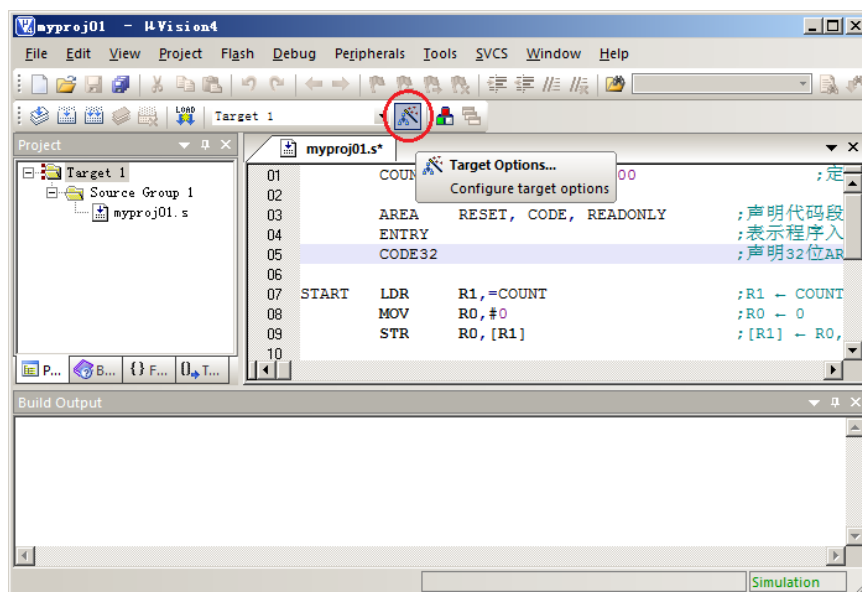


图 1-8 设置工程选项

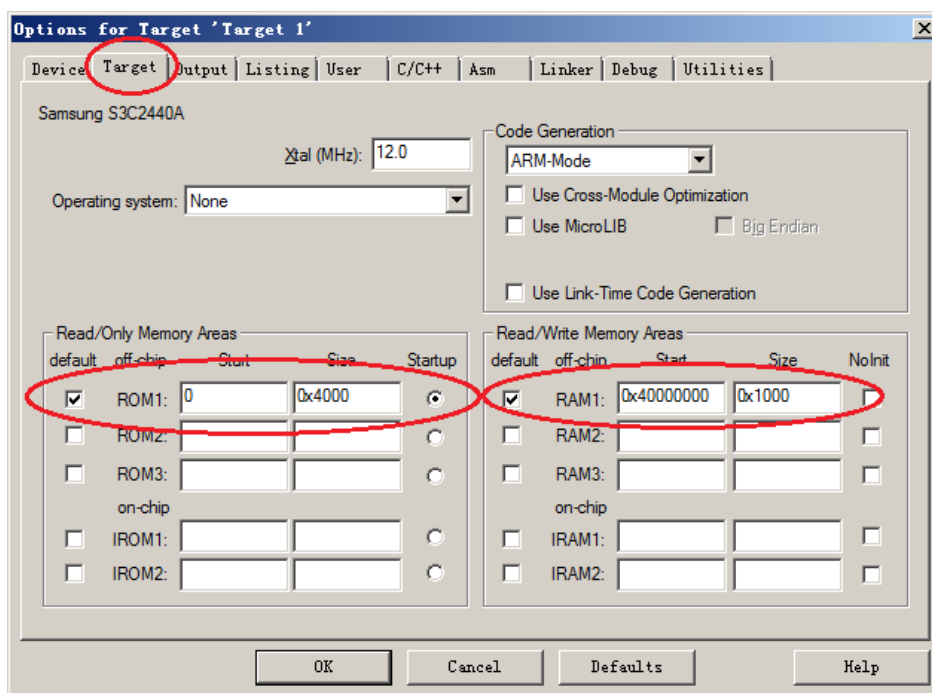


图 1-9 设置存储器映射

(4) 编译链接工程。见图 1-10。

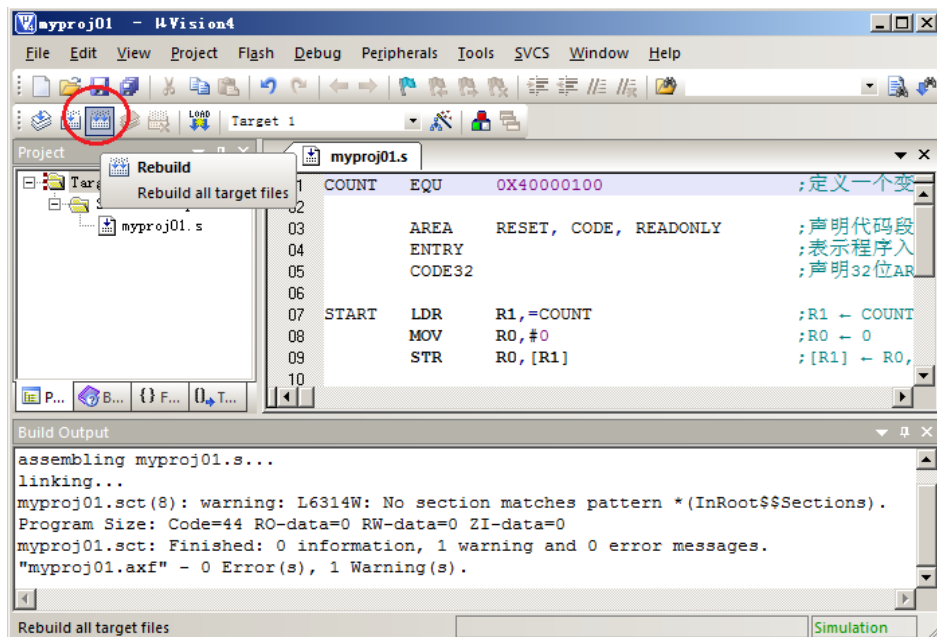


图 1-10 编译工程

(5) 进行软件仿真调试。见图 1-11、图 1-12、图 1-13、图 1-14。

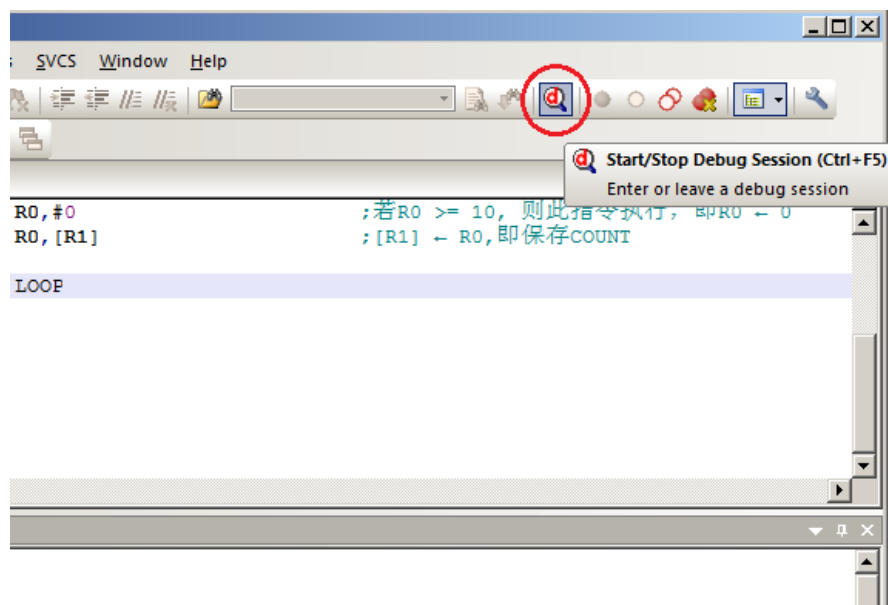


图 1-11 调试运行

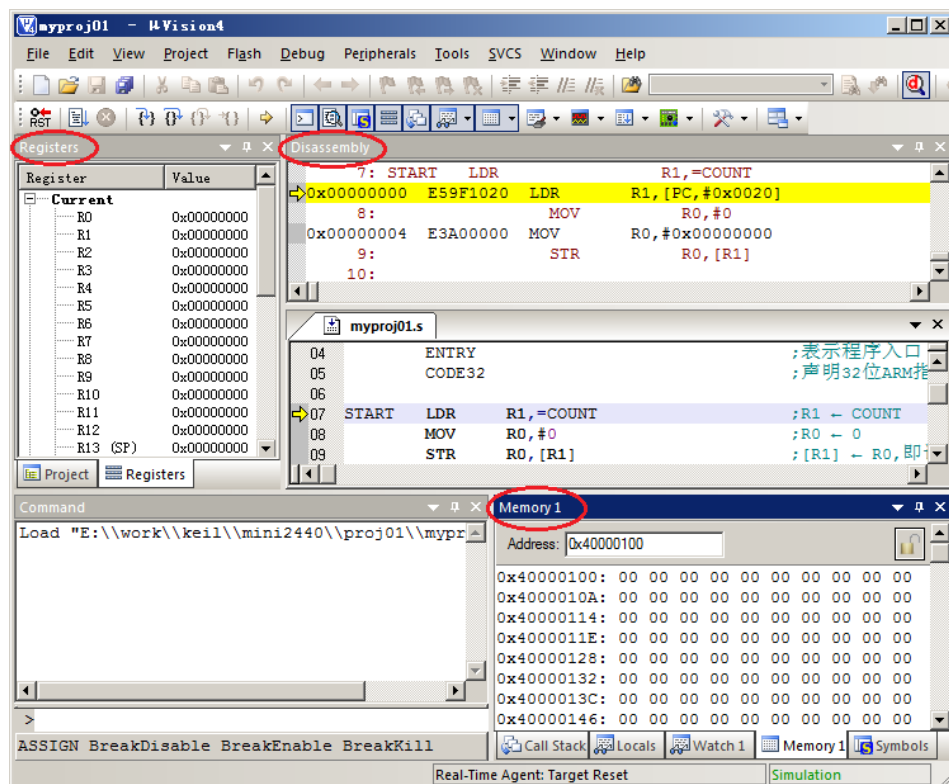


图 1-12 调试界面

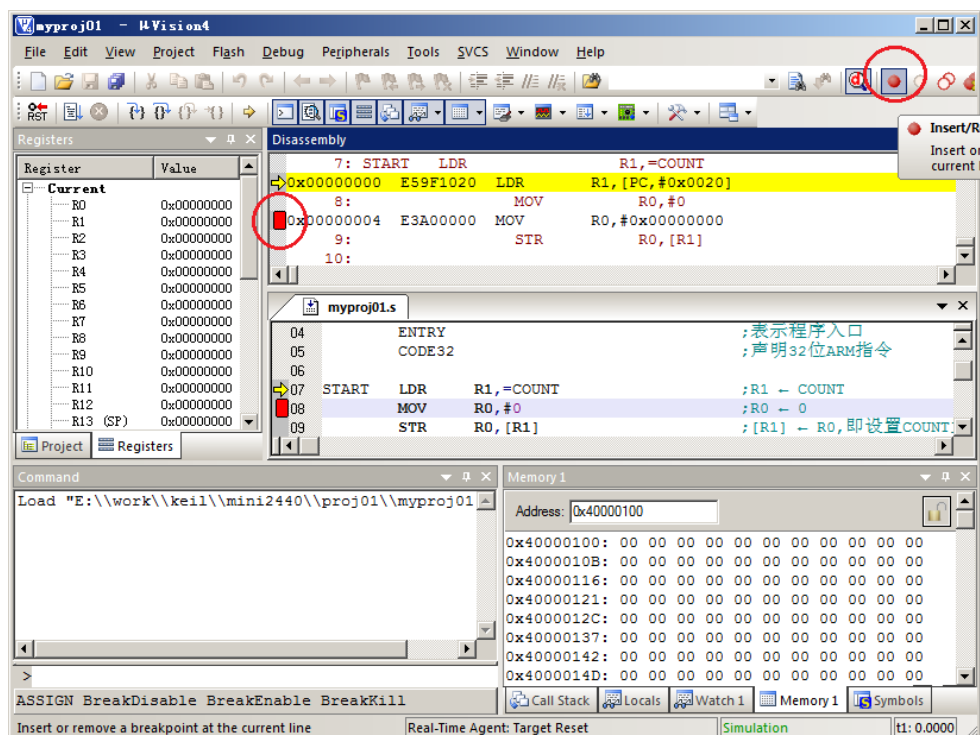


图 1-13 设置断点

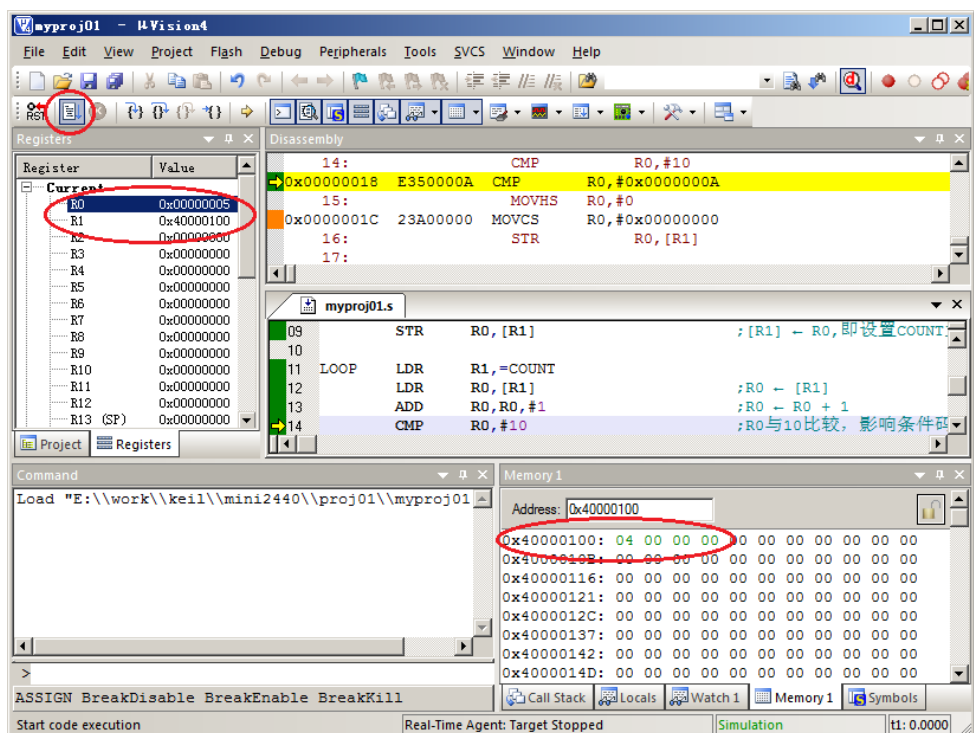


图 1-14 运行程序

6. 实验参考程序

汇编指令实验的参考程序见程序清单 1.1。

程序清单 1.1 汇编指令实验参考程序

COUNT	EQU	0X40000100	; 定义一个变量, 地址为 0x40000100
AREA	RESET, CODE, READONLY		; 声明代码段 RESET
ENTRY			; 表示程序入口
CODE32			; 声明 32 位 ARM 指令
START	LDR	R1, =COUNT	; R1 ← COUNT
	MOV	R0, #0	; R0 ← 0
	STR	R0, [R1]	; [R1] ← R0, 即设置 COUNT 为 0
LOOP	LDR	R1, =COUNT	
	LDR	R0, [R1]	; R0 ← [R1]
	ADD	R0, R0, #1	; R0 ← R0 + 1
	CMP	R0, #10	; R0 与 10 比较, 影响条件码标志
	MOVHS	R0, #0	; 若 R0 >= 10, 则此指令执行, 即 R0 ← 0
	STR	R0, [R1]	; [R1] ← R0, 即保存 COUNT

B LOOP

END

7. 思考

(1) 若使用 LDRB/STRB 代替程序清单中的所有加载/存储指令 (LDR/STR), 程序会得到正确的执行吗?

(2) LDR 伪指令与 LDR 加载指令的功能和应用有何区别, 举例说明? (提示: LDR 伪指令的形式为 “LDR Rn, =expr” 。)

(3) LDR/STR 指令的前索引偏移指令如何编写? 指令是怎样操作的?

8. 选作内容

- (1) 使用 ARM 汇编指令实现 if 条件执行。
- (2) 使用 ARM 汇编指令实现 for 循环结构。
- (3) 使用 ARM 汇编指令实现 while 循环结构。
- (4) 使用 ARM 汇编指令实现 do...while 循环结构。