微机原理与嵌入式系统

实验报告

信息科学技术学院

姓名：胡睿 PB17061124

**【实验题目】实验1 基于ASM的Project**

**【实验目的】**

1、掌握µVision IDE基本使用、了解一个项目编译、连接、调试的工作过程

2、汇编代码编写的一般语法，掌握编写子程序的方法

3、掌握常规代码调试技巧

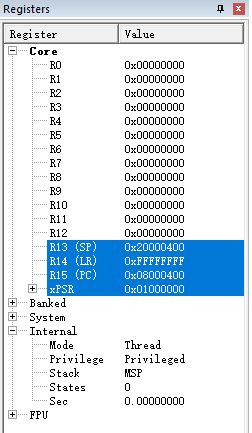
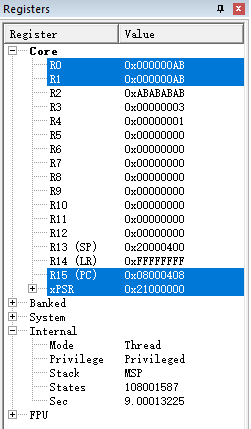
4、理解编程者模型

**【实验内容】**

1.2.1、Project的建立、编译、连接

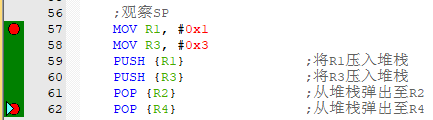
1.2.2、Project的调试（Debug）

运行前寄存器数值： 运行后寄存器数值：

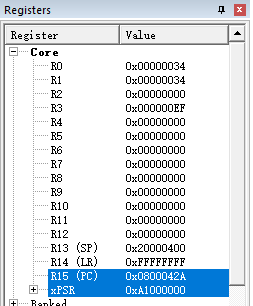
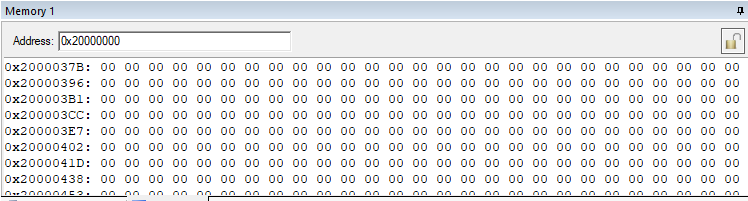
 

（4）观察调试过程中通用寄存器的变化

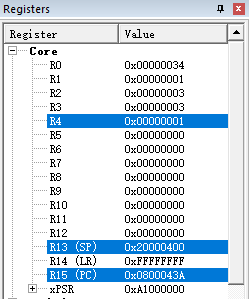
（5）观察如下代码执行前后栈指针的变化

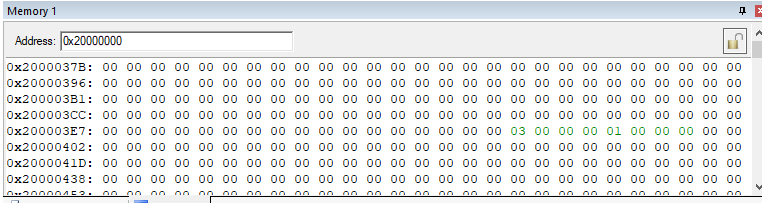


运行前：

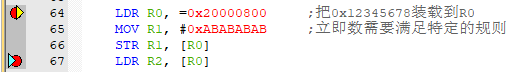
运行后：



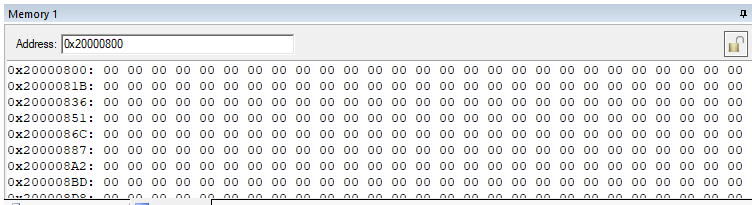


可以看出栈的指针是递减的，同时指向栈顶的元素。

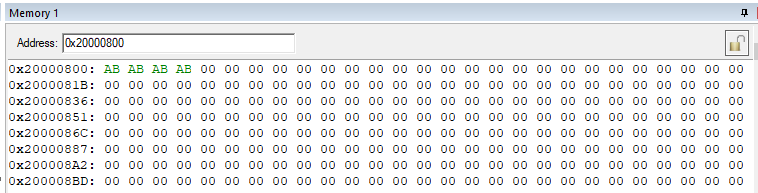
（6）在如下代码执行前后观察存储器0x20000800地址的内容



执行前：



执行后：



**【1.3 思考题】**

**（7） 异常处理子程序Reset\_Handler的入口地址是？**

答：0x08000400

**（8） 添加一行代码，使xPSR寄存器的Z标志位为1。**

答：MOV R1,#0x34

CMP R1,#0x34

**（9） 示例代码中，为何使用的是MSP而不是PSP。**

答：因为是工作在handler模式下，只允许使用主堆栈指针MSP。

**（10） 请解释执行至Reset\_Handler中第一行代码时，为何MSP为“0x20000400”？**

答：因为在程序开头，定义了栈的大小为0x00000400，然后开始执行时，会指向栈的顶部，即为0x20000400，同时为递减堆栈。

**（11） 请依据代码调试中观察到的机器指令解释伪指令“LDR R0, =0x20000800”被翻译为机器指令的执行过程？**

答：首先执行机器码4803，LDR，然后将pc寄存器向下偏移12个字节的数据存入R0中。

**（12） 伪指令“LDR R0, =0x20000800”中数值“0x20000800”，被存放在哪个地址？**

答：0x08000448。

**（13） 解释指示符（伪指令）“EXPORT”和“DCD”的作用？**

答：export伪指令用于程序中声明一个全局的标号，该标号可在其他的文件中引用。DCD伪指令用于分配一片连续的字存储单元并用伪指令中指定的表达式初始化。

**（14） 观察代码执行过程中，PC变化的规律。并写出一条使PC递增2的指令，再写出一条使PC递增4的指令。**

答：递增2的指令：MOV R0, R1

递增4的指令：MOV R1, #0xAB。

**【实验题目】实验2 基于C的Project**

**【实验目的】**

1、掌握µVision IDE下创建C语言工程的基本步骤

2、了解µVision IDE自带CMSIS库和device的启动文件

3、掌握联机帮助查询技巧

4、掌握代码分析技巧

5、理解ARM汇编程序中的伪指令（指示符，Directive）

6、掌握C和汇编混合编程方法

**【实验内容】**

**2.2.1、建立基于C程序的Project**

**2.2.2、代码功能验证**

**（16） 阅读启动文件“startup\_ARMCM3.s”和“startup\_ARMCM3.c”，分析启动的大致过程。**

答：首先设置了栈和堆的大小并进行设置，然后设置了向量表，然后调用reset handler，引导程序进入\_\_main，最后进行用户堆栈的初始化，完成启动的过程。

**（17） 分析示例代码中C调用汇编子程序的过程。**

答：示例中，c程序调用的是strcopy子程序，strcopy首先将R1指向源字符串地址，取出字符内容存入R2，然后R0指向目的字符串地址，R2中内容存入R0指向内存单元，然后比较R2里面是不是全空，就可以实现复制字符串的功能。

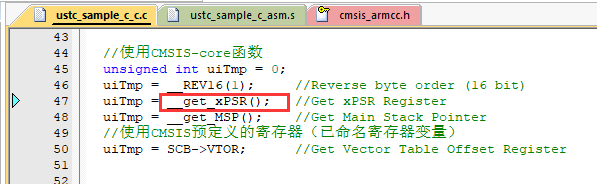
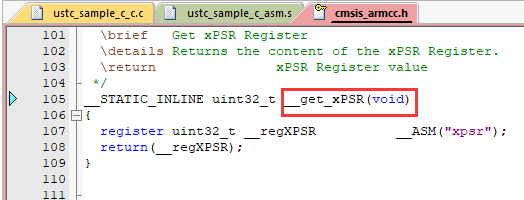
**（18） 分析示例代码中汇编调用C子程序的过程。**

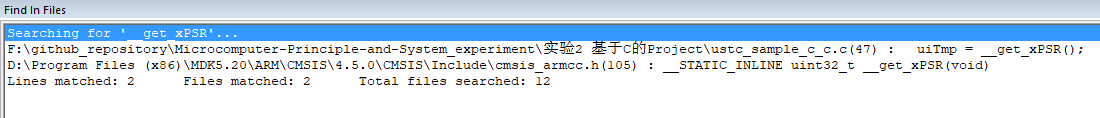
答：示例中，汇编调用的是MY\_C\_FUNCTION子程序，通过传入两个int数，然后返回两个数的和。

**2.2.3、代码分析技巧**

**（19） 在当前Project内所有文件中查找“\_\_get\_xPSR”关键字。**

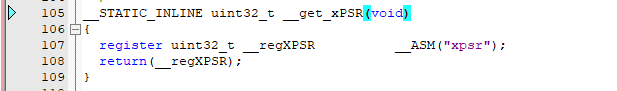
**逐个文件查看关键字“\_\_get\_xPSR”出现的位置。**



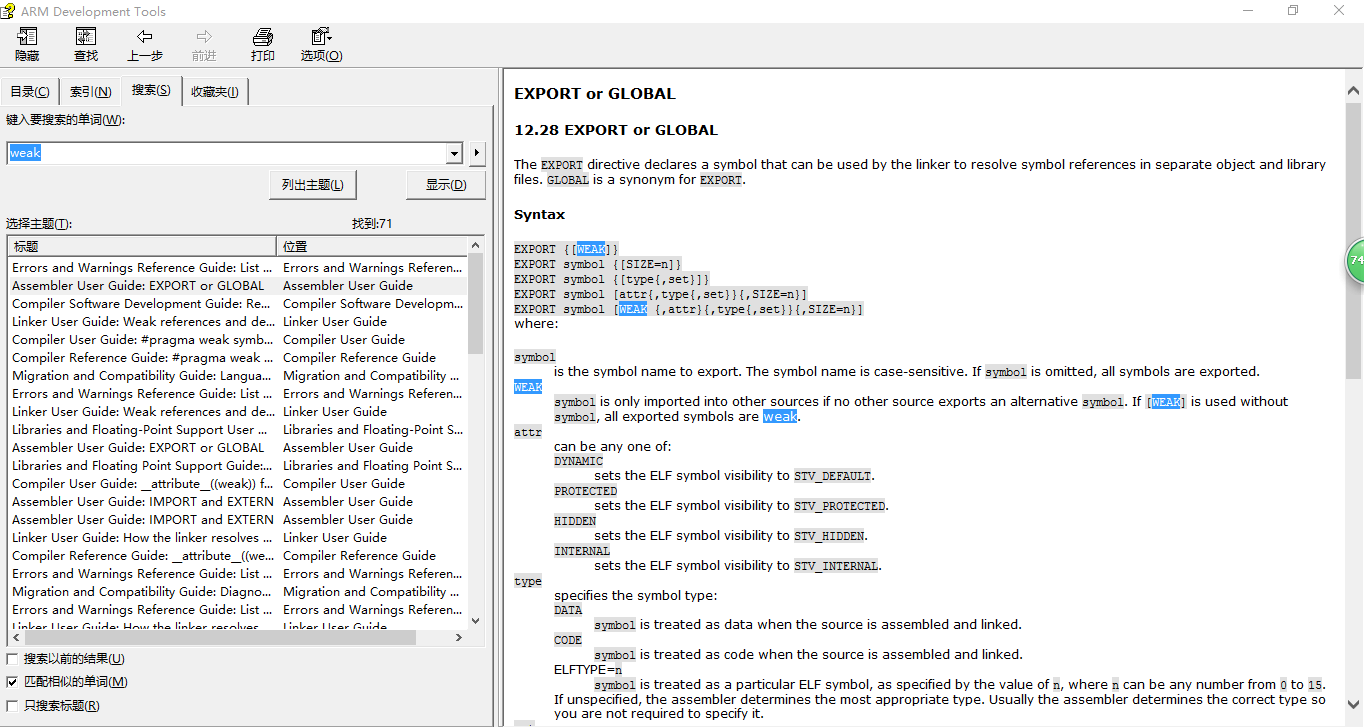
**（20） 分析示例代码中所调用的CMSIS-core函数的定义。如下图所示在函数位置点击鼠标右键（或热键F12），跳转至\_get\_xPSR()的定义文件。浏览所打开的“cmsis\_armcc.h”文件。**

答：访问一些汇编指令和特殊寄存器。

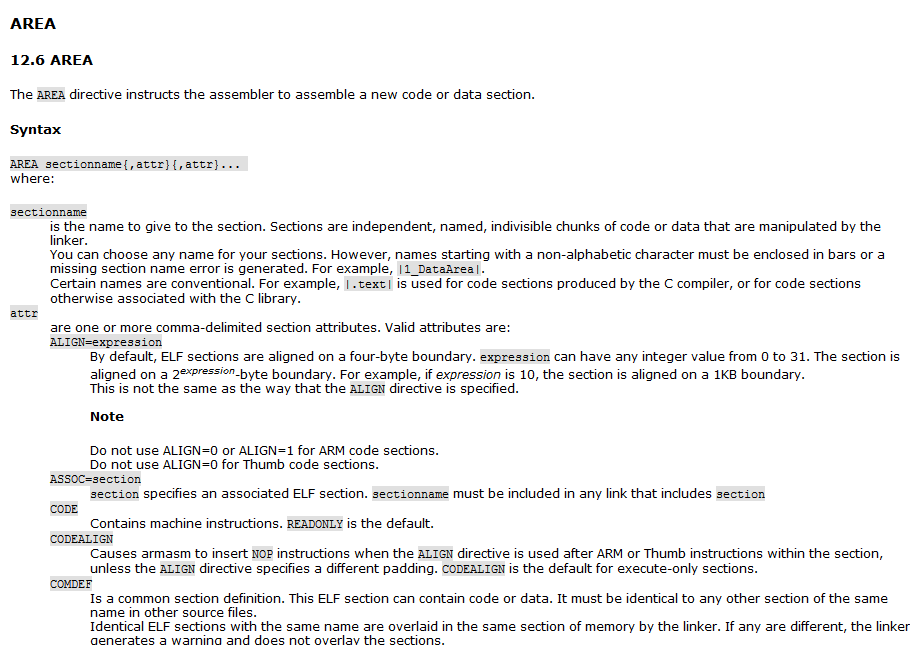


**2.2.4 µVision联机资源使用**

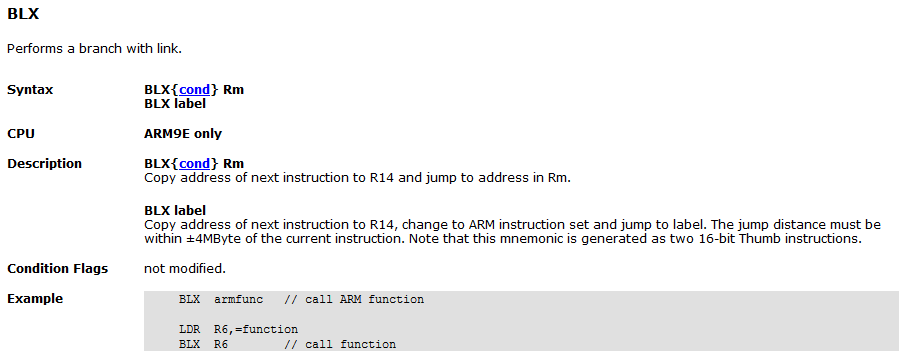
**（21） 打开联机帮助文档，搜索指示符“WEAK”的作用。**



**（22） 将光标停留在某个指示符（伪指令关键字）上，如“AREA”，按键“F1”，查阅自动弹出的帮助信息。**

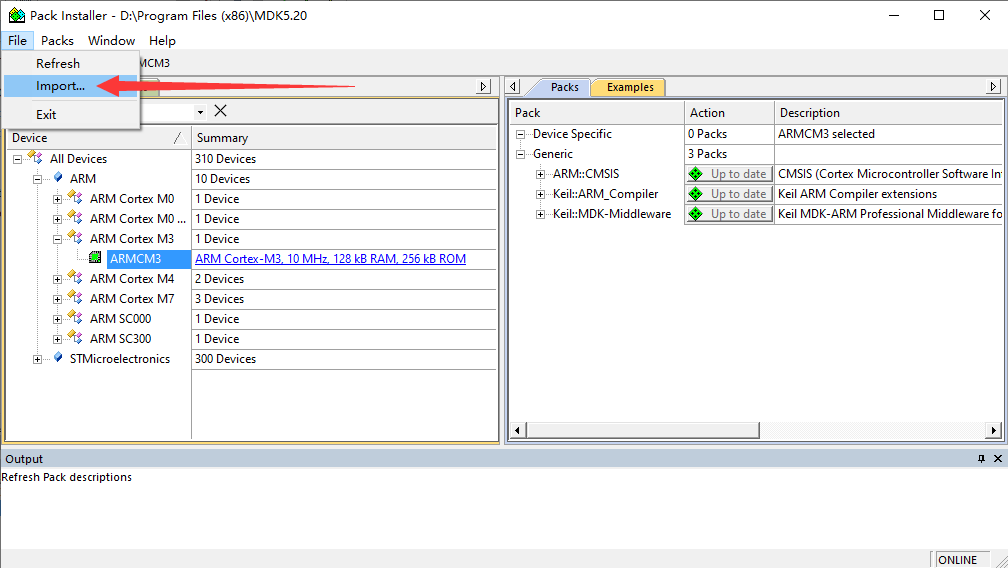
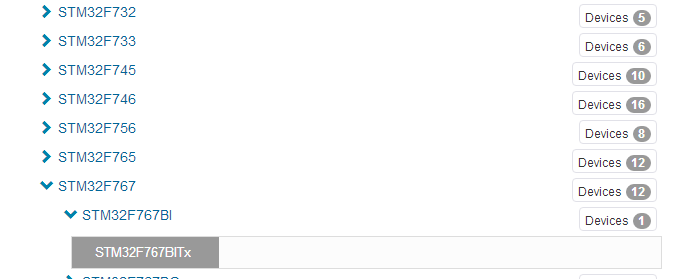


**（23） 将光标停留在某条指令上，如“BLX”，按键“F1”，查阅自动弹出的帮助信息。**



**（24） 了解包管理器“Pack installer”的基本功能。**

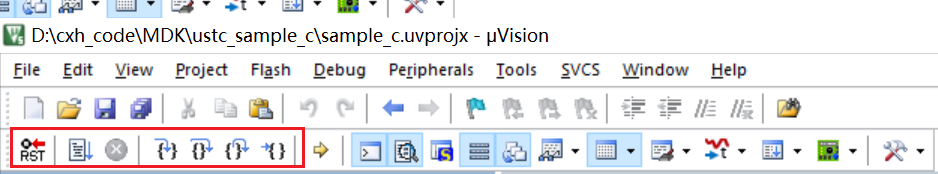
keil中如果找不到自己要使用的芯片，可以使用keil的“Pack installer”找到目标芯片进行下载安装，首先我们要到keil的官网http://www.keil.com/dd2找到我们需要的芯片并下载对应的.pack文件如下左图所示：



然后点击keil的“Pack installer”进入如上图右图界面，点击file-import找到下载好的.pack文件即可安装。

**2.2.5、Project的调试（Debug）**

**（25） 通过试验分析下图所示Debug工具栏（红色框内）各个图标对应功能的区别。**



答：第一个是重新设置cpu，相当于清零大部分寄存器；

第二个是向下运行；

第三个是停止代码的执行；

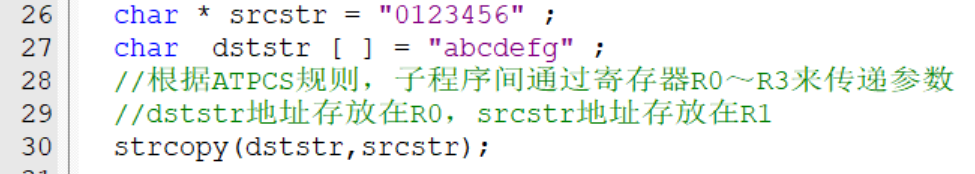
第四个是进入函数的单步执行；

第五个是不进入函数的单步执行；

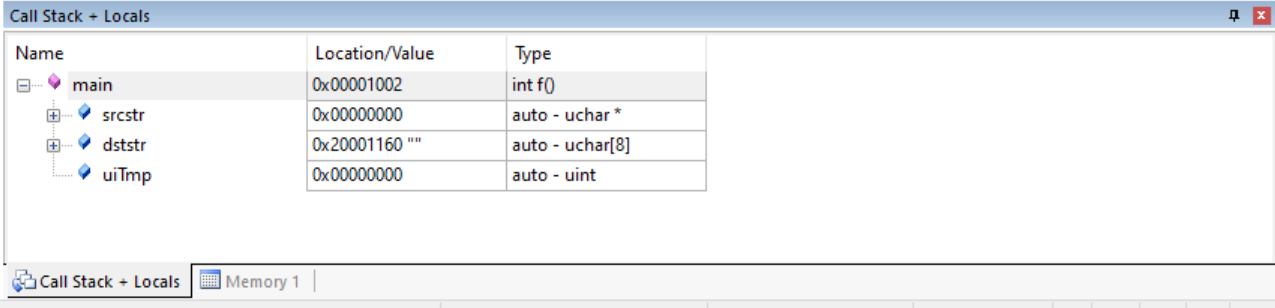
第六个是跳过当前函数执行；

第七个是执行到光标位置。

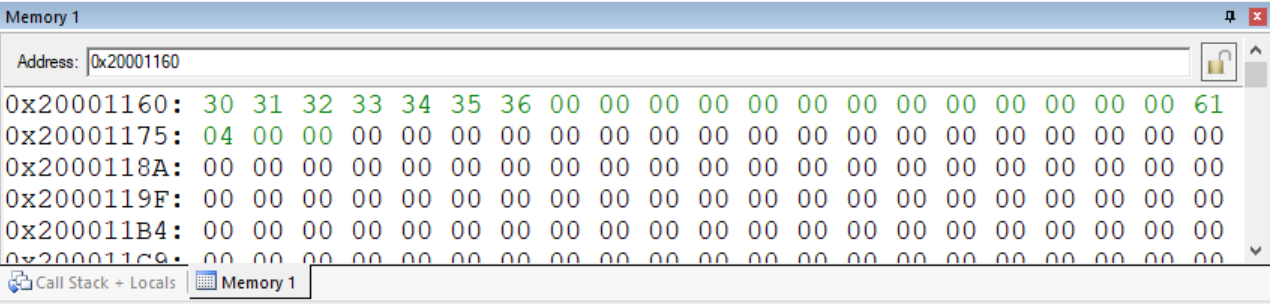
**（26） 通过Watch窗口观察一下代码执行前后变量值。**

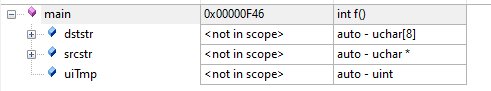


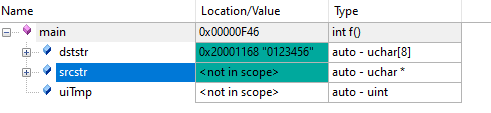
观察变量“srcstr”和“dststr”

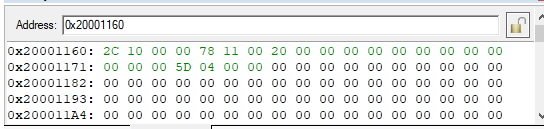


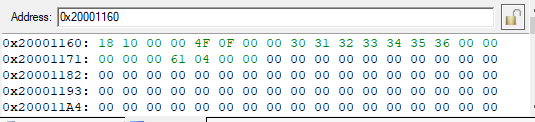
同时观察Memory窗口对应地址的值



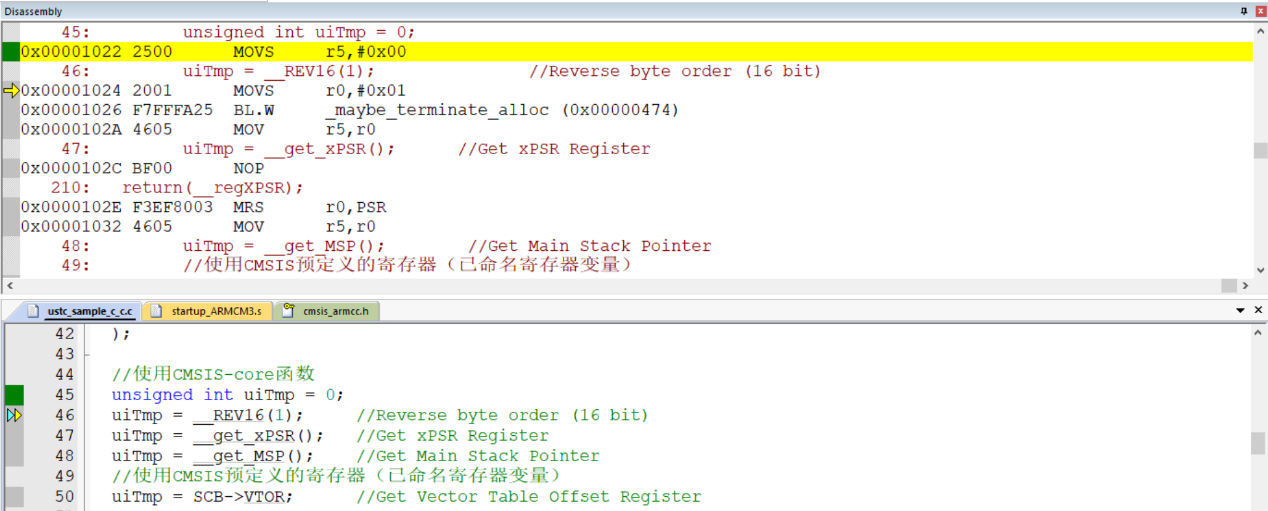








**（27） 分析下图所示反汇编窗口中机器指令与源代码窗口中C代码的对应关系。**



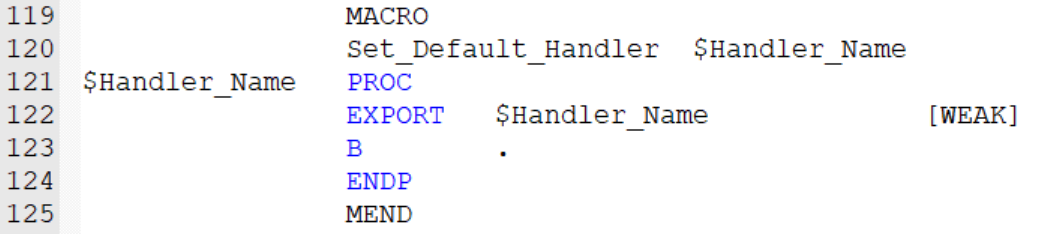
答：第45行对应第一条机器指令。

第46行对应后面三条机器指令。

第47行对应后面四条机器指令。

**【2.3】思考题（内容需要上机调试才能完成）**

**（28） 分析启动文件“startup\_ARMCM3.s”中图下宏定义的含义？并写出$Handler\_Name等于NMI\_Handler时，该宏定义展开后的代码。**



答：表示如果有多个地方出现了$Handler\_Name，那么就不改变原来的量的定义。

NMI\_Handler PROC

EXPORT NMI\_Handler [WEAK]

B .

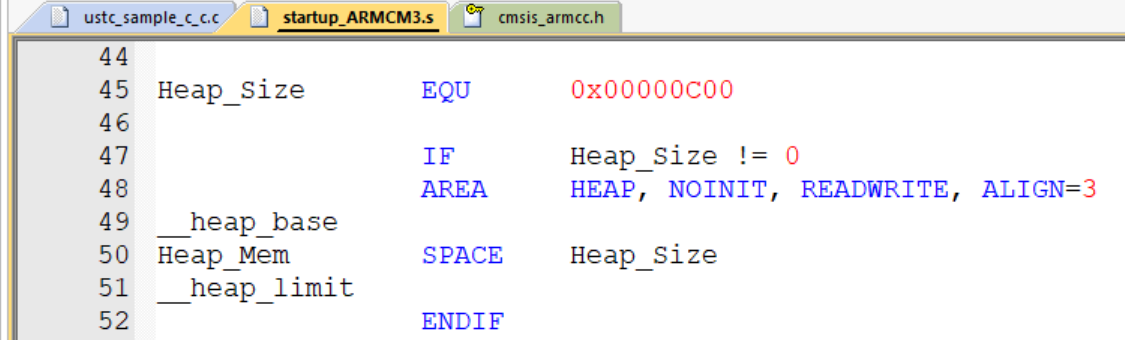
ENDP

**（29） 依据调试结果，示例c程序中如下行中字符串"Hello USTCer\n"被保存在存储器的什么位置（写出存储器地址）？**



答：0x00000F80

**（30） 请结合“startup\_ARMCM3.s”文件中如下代码分析示例中“srcstr”的地址为何是“0x20001160”？**



答：初始化的堆栈头地址为0x20000178，堆空间为0x00000C00，栈空间为0x00000400，初始化的栈顶为0x20001178.栈是向下增长的，所以dststr存放的地址是0x20001168.

**【实验题目】实验3 基于STM32库的GPIO与定时器**

**【实验目的】**

（1） 掌握µVision IDE中基于ST公司STM32库建立project的流程

（2） 了解ST公司提供的TIM、GPIO相关库函数

（3） 了解STM32F10X系列芯片定时器相关的寄存器功能。

（4） 掌握µVision IDE中外设仿真模块（GPIO）的使用

a) 学会利用外设仿真模块（GPIO）观察I/O引脚输出

b) 学会利用外设仿真模块（GPIO）模拟I/O引脚的输入

（5） 掌握µVision IDE逻辑分析模块（Logic Analyzer）的使用

**【实验内容】**

**3.2.1 下载ST公司STM32库及芯片手册**

**（31） 下载ST公司关于STM32F103系列芯片的库文件**

**（32） 下载ST公司关于STM32F103系列芯片的文档**

**3.2.2 建立基于STM32库的Project**

**3.2.3 配置Project的头文件目录、预编译参数、Simulator**

**3.2.4 Debug时使用外设仿真功能验证GPIO输入和输出**

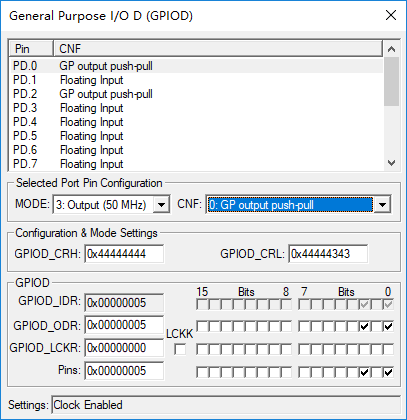
**3.2.5 Debug时使用Logic Analyzer观察GPIO输出**

**3.2.6 使用外设仿真和Logic Analyzer验证GPIO输入和输出**

**3.2.7 定时器的配置**

**【3.3 思考题（部分内容需要上机调试才能完成）】**

**（44） 阅读讲义8.5.3小节，及STM32芯片手册，解释下图中“CRL、CRH、IDR、ODR、LCKR”几个寄存器的作用。**



答：首先设置了栈和堆的大小并进行设