

# 实验1\_清零汇编程序实验

原创

刘一五 于 2020-11-09 20:40:54 发布 4966 收藏 57

分类专栏: [# 51单片机编程与仿真](#) 文章标签: [编程语言 c语言](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/Jessaly/article/details/109554416>

版权



[51单片机编程与仿真](#) 专栏收录该内容

3 篇文章 1 订阅

订阅专栏

## 一、实验要求

- (1) 把外部数据存储区 2000H~20FFH 的内容清零
- (2) 把外部数据存储区 2000H~20FFH 的内容全部填入FF

## 二、代码

- (1) 将把外部数据存储区 2000H~20FFH 的内容清零的汇编程序

```
ORG 0000H
LJMP MAIN
ORG 0640H
MAIN: MOV R0, #00H           ;R0从00开始计数
      MOV DPTR, #2000H
LOOP: CLR A
      MOVX @DPTR, A
      INC R0
      INC DPTR               ;指向下一个地址
      CJNE R0, #00H, LOOP   ;R0记完FF个数, 跳出循环
      SJMP $
      END
```

- (2) 将把外部数据存储区 2000H~20FFH 的内容清零的C语言程序

```

#include <reg51.h>
#include <absacc.h>
void main()
{
    unsigned char i;
    unsigned int j;
    for(i=0;i<=0xff;i++)
    {
        j = 0x2000 + i; //起始地址为外部数据存储区:0x2000;
        XBYTE[j]=0x00; //清零
    }

    while(1);
}

```

(3) 把外部数据存储区 2000H~20FFH 的内容全部填入FF的汇编程序

```

ORG 0000H
LJMP MAIN
ORG 0640H
MAIN: MOV R0,#00H
      MOV DPTR,#2000H
LOOP: MOV A,#0FFH      ;A中写入FF
      MOVX @DPTR,A
      INC R0
      INC DPTR
      CJNE R0,#00H,LOOP
      SJMP $
END

```

(4) 把外部数据存储区 2000H~20FFH 的内容全部填入FF的C语言程序

```

#include <reg51.h>
#include <absacc.h>
void main()
{
    unsigned char i;
    unsigned int j;
    for(i=0;i<=0xff;i++)
    {
        j = 0x2000 + i; //起始地址为外部数据存储区:0x2000;
        XBYTE[j]=0xff; //写入FF
    }

    while(1);
}

```

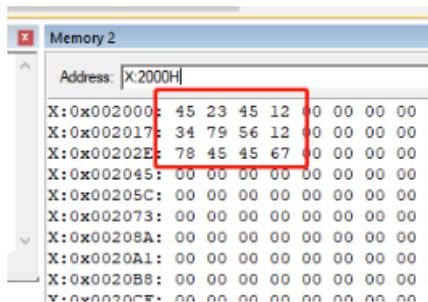
### 三、调试过程及结果

点击此放大镜进入调试



在Memory窗口填入地址X:2000H回车即可查看该片区域的情况

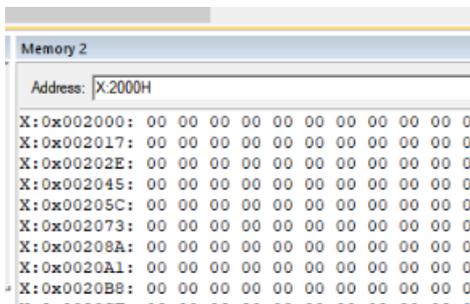
(1) 手动改一下某几个地址的值



(2) 然后点击程序运行

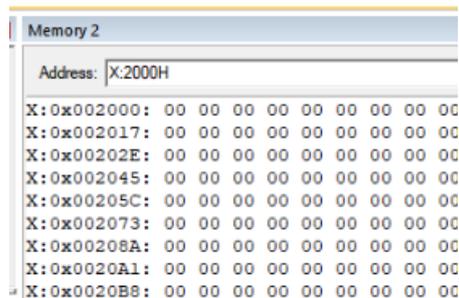


程序运行后结果如下:

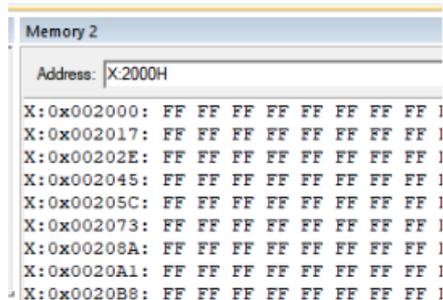


(3)打开第二个程序并且进入调试

可以看到2000H~20FFH对应的初始内容为00



执行完程序后



#### 四、收获与思考

- 1、16进制的立即数最高位大于等于A时，需要再前面再加上0（如写#FFH会报错，正确的形式应该是#0FFH）
- 2、在memory窗口查看ROM或RAM对应地址的内容（查看ROM是C:，查看片内RAM是D:，查看片外RAM是X:）
- 3、汇编程序文件中不可以像C51文件那样包含其他头文件，所有程序必须写在同一个程序文件中。因此也不适用于大的项目文件的开发。

话说汇编语言有这么大的局限性我们为什么还要学汇编呢？

（某度来的，觉得讲得很有道理）

（1）汇编语言操作直接面向硬件。所以我们在使用汇编语言的时候，我们能够感知计算机的运行过程和原理，从而能够对计算机硬件和应用程序之间的联系和交互形成一个清晰的认识。（2）汇编语言能够让你更好的理解高级语言，尤其是高级语言中的C语言。汇编语言对于内存的操作都是基于内存地址的，而C语言中最令人头疼的指针概念，说白了就是内存的地址。

- 4、运行了一次要rst复位后才能进行下一次的运行

其实对单片机中某地址的清零操作就是写入0，通过这个实验不仅知道如何对某些单元清零还知道如对具体地址写入数据。