实验 2　　汇编指令实验

1. 实验目的

●掌握 ARM 乘法指令的使用方法

●了解子程序编写及调用

2. 实验设备

●硬件：计算机一台

●软件：Windows XP 系统，ADS1.2 集成开发环境

3. 实验内容

使用 STMFD/LDMFD、MUL 指令编写一个整数乘方的子程序，然后使用 BL 指令调用子程序计算。

4. 实验预习要求

（1）　仔细阅读《ARM 嵌入式系统基础教程（第 2 版）》第 3 章 ARM 指令系统的内容。

（2）　仔细阅读文献 1 中 ADS 工程编辑和 AXD 调试的内容。（本实验使用软件仿真）

5. 实验原理

X^n=X*X*X*X·····*X，其中相乘的 X 的个数为 n 个。先将 X 的值装入 R0 和 R1，使用寄存器 R2 进行计数，循环 n-1 次 R0=R0*R1，运算结果就保存在 R0 中。（不考虑结果溢出问题。）

注意：若 n 为 0，则运算结果直接赋值 1；若 n 为 1，则运算结果直接赋值 X。

6. 实验步骤

（1）　启动 ADS1.2，使用 ARM Executable Image 工程模板建立一个工程 Instruction3。

（2）　建立汇编源文件 TEST4.S，编写实验程序，然后添加到工程中。

（3）　设置工程连接地址 RO Base 为 0X4000 0000，RW Base 为 0X4000 3000。设置调试入口地址 Image entry point 为 0X4000 0000。

（4）　编译连接工程，选择 Project→Debug，启动 AXD 进行软件仿真调试。

（5）　打开寄存器窗口（Processor Registers），选择 Current 项监视寄存器 R0，R1，R13（SP）和 R14（LR）的值。

（6）　打开存储器观察窗口（Memory），设置观察地址为 0X4000 3EA0，显示方式 Size 为 32bit，监视从 0X4000 3F00 起始的满递减堆栈区。

（7）　单步运行程序，跟踪程序执行的流程，观察寄存器值得变化和堆栈区的数据变化，判断执行结果是否正确。

（8）　调试程序时，更改参数 X 和 n 来测试程序，观察是否得到正确的结果。例如：先复位程序（选择 File→Reload Current Image），接着单步执行到 “BL POW” 指令，在寄存器窗口中将 R0 和 R1 的值进行修改，然后继续运行程序。

说明：双击寄存器窗口的寄存器，即可修改寄存器的值。输入数据可以是十进制数（如 136/198），也可以是十六进制数（如 0X123、0XF0），输入数据后回车确定。

7. 实验参考程序

汇编指令实验的参考程序见程序清单 2.1。

程序清单 2.1　汇编指令实验 2 的参考程序

```
; 文件名：TEST4.S
; 功能：计算X的n次方的值
; 说明：X和n均为无符号整数

X       EQU     9               ; 定义X的值为9
n       EQU     8               ; 定义n的值为8

        AREA    Example4,CODE,READONLY ; 声明代码段Example4
        ENTRY                   ; 标识程序入口
        CODE32                  ; 声明32位ARM指令

START   LDR     SP,=0x40003F00      ; 设置堆栈(满递减堆栈，使用STMFD/LMDFD指令)
        LDR     R0,=X
        LDR     R1,=n
        BL      POW                 ; 调用子程序POW，返回值为R0

HALT    B       HALT


; 名称：POW
; 功能：整数乘方运算。
; 入口参数：R0      底数
;           R1      指数
; 出口参数：R0      运算结果
; 占用资源：R0、R1
; 说明：本子程序不考虑溢出问题
POW
        STMFD   SP!,{R1-R12,LR}     ; 寄存器入栈保护
        MOVS    R2,R1               ; 将指数值复制到R2，并影响条件码标志

        MOVEQ   R0,#1               ; 若指数为0，则设置R0=1
        BEQ     POW_END             ; 若指数为0，则返回

        CMP     R2,#1
        BEQ     POW_END             ; 若指数为1，则返回。(此时R0没有被更改)

        MOV     R1,R0               ; 设置DO_MUL子程序的入口参数R0和R1
        SUB     R2,R2,#1            ; 计数器R2 = 指数值减1
POW_L1  BL      DO_MUL              ; 调用DO_MUL子程序，R0 = R1 * R0
```

```
            SUBS    R2,R2,#1                ; 每循环一次，计数器R2减1
            BNE     POW_L1                  ; 若计数器R2不为0，跳转到POW_L1
            
            
POW_END     LDMFD   SP!,{R1-R12,PC}         ; 寄存器出栈，返回

; 名称：DO_MUL
; 功能：32位乘法运算。
; 入口参数：R0      乘数
;           R1      被乘数
; 出口参数：R0      计算结果
; 占用资源：R0、R1
; 说明：本子程序不会破坏R1
DO_MUL      MUL     R0,R1,R0                ; R0 = R1 * R0
            MOV     PC,LR                   ; 返回
            
            END
```

8. 思考题

（1）若需要考虑溢出问题（使用 32 位运算结果，判断运算是否溢出），如何修改实验参考程序？（提示：使用 UMULL 指令）

（2）在实验参考程序中的 DO_MUL 子程序，是否可以使用 B、ADD、和 SUB 指令返回？（提示：修改程序进行调试）