

实验 1 ARM 汇编指令实验

实验目的

- 初步学会使用 ADS1.2 开发环境及 ARM 软件模拟器；
- 通过实验掌握简单 ARM 汇编指令的使用方法。

实验设备

- 硬件：PC 机
- 软件：ADS1.2+WIN XP。

实验内容

- 熟悉开发环境的使用并使用 `ldr/str`, `mov` 等指令访问寄存器或存储单元。
- 使用 `add/sub/lsl/lsr/and/orr` 等指令，完成基本数学/逻辑运算。

实验原理

ARM 处理器共有 37 个寄存器：31 个通用寄存器，包括程序计数器(PC)。这些寄存器都是 32 位的。6 个状态寄存器。这些寄存器也是 32 位的，但是只是使用了其中的 12 位。这里简要介绍通用寄存器。

1. ARM 通用寄存器

通用寄存器 (R0~R15) 可分为 3 类：

不分组寄存器 R0~R7；

分组寄存器 R8~R14；

程序计数器 R15；

1) 不分组寄存器 R0~R7；

R0~R7 是不分组寄存器。这意味着在所有处理器模式下，它们每一个都访问一样的 32 位寄存器。它们是真正的通用寄存器，没有体系结构所隐含的特殊用途。

2) 分组寄存器 R8~R14

R8~R14 是分组寄存器。它们每一个访问的物理寄存器取决于当前的处理器模式。若要访问特定的物理寄存器而不依赖当前的处理器模式，则使用规定的名字。

寄存器 R8~R12 各有两组物理寄存器：一组为 FIQ 模式，另一组为除了 FIQ 以外的所有模式。寄存器 R8~R12 没有任何指定的特殊用途。只是使用 R8~R14 来简单地处理中断。寄存器 R13,R14 各有 6 个分组的物理寄存器。1 个用于用户模式和系统模式，其它 5 个分别用于 5 种异常模式。寄存器 R13 通常用做堆栈指针，称为 SP。每种异常模式都有自己的 R13。寄存器 R14 用作子程序链接寄存器，也称为 LR。

3) 程序计数器 R15

寄存器 R15 用做程序计数器 (PC)。

在本实验中，我们认为 ARM 核工作在用户模式，R0~R15 可用。

2. 存储器格式

ARM 体系结构将存储器看作是从零地址开始的字节的线性组合。字节零到字节三放置第一个字（WORD），字节四到字节七存储第二个字，以此类推。

ARM 体系结构可以用两种方法存储字数据，分别称为大端格式和小端格式。

1) 大端格式

在这种格式中，字数据的高位字节存储在低地址中，而字数据的低位字节则存放在高地址中，如图 2-1 所示。

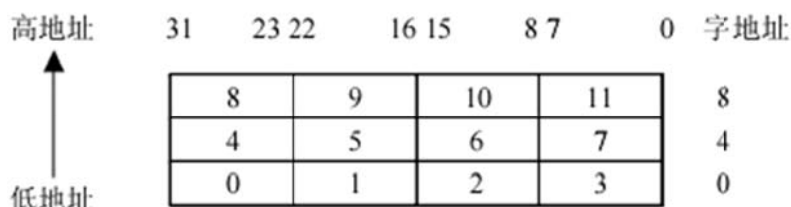


图 0-1 存储器大端格式

2) 小端格式

在这种格式中，字数据的高位字节存储在高地址中，而字数据的低位字节则存放在低地址中，如图 2-2 所示。

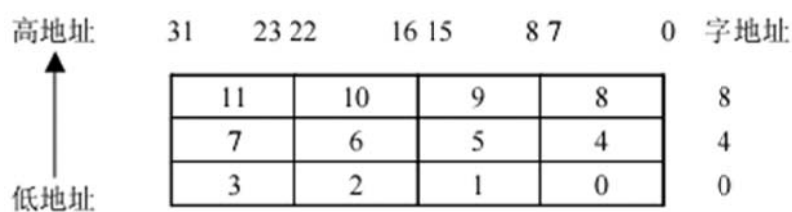


图 0-2 存储器小端格式

实验步骤

- 1) 新建工程：
- 2) 编辑源文件：
- 3) 添加源文件：
- 4) 配置。RO_Base 设置为 0X8000
- 5) 编译生成目标代码：
- 6) 调试运行（注意：第一次使用 AXD 调试器时，应先对其进行配置，参考理论课 PPT）
- 7) 单步执行程序并观察和记录寄存器 r0,r1 与 memory@0x1000 的值变化。

数据记录表

指令	指令执行后变化情况		
	寄存器		存储器
	R0	R1	0x1000
movsp, #stack_add			

8)理解和掌握实验后，完成实验练习题(必做)。

实验参考程序

```
x EQU 45
y EQU 64
stack_add EQU 0x1000
        AREA SY,CODE,READONLY
        ENTRY
START
        mov    sp, #stack_add
        mov    r0, #x
        str    r0, [sp]
        mov    r0, #y
        ldr    r1, [sp]
        add    r0, r0, r1
        str    r0, [sp]
STOP
        b     STOP
        END
```

练习题

1. 编写程序循环对 R4~R11 进行累加 8 次操作，R4~R11 起始值为 1~8，每次加操作后把 R4~R11 的内容放入 SP 栈中，SP 初始设置为 0x800。最后把 R4~R11 用 LDMFD 指令清空赋值为 0。