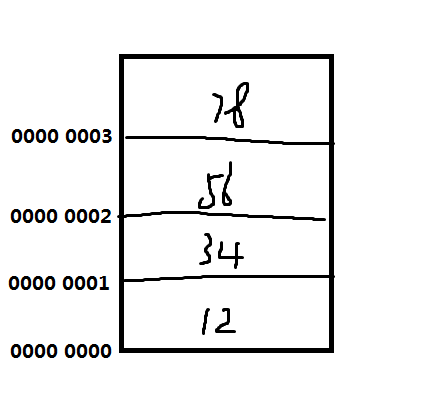
* 1. ARM处理器架构概述
     1. 大小端

16位是两个字节，前八位是高字节，后八位是低字节。当一个逻辑上长于一个字节的整形数据放置在内存中时（比如16位，32位，和64位的整数），计算机设计者需要考虑这些字节的存储顺序。一些体系结构的设计者选择了将字节的逻辑顺序与物理顺序一致，即将逻辑上较低的字节放置在物理上较低的字节上；另外一些设计者则选择了将字节的逻辑顺序与物理顺序相反，即将逻辑上较低的字节放置在物理上较高的字节上。前者被称为“little endian”低字节优先，比如Intel x86系列；后者则被称为“big endian”高字节优先，比如Motorola的PowerPC以及Sun Sparc。

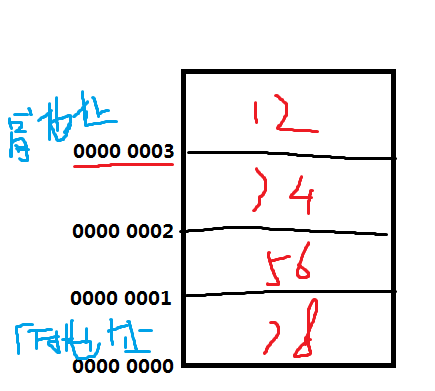
比如16进制的数据0x12345678，逻辑上（数理范畴）12是高位，78是最低位。

小端系统中，0x12345678内存结构图：



1. 小端系统0x12345678内存图

在大端系统中，0x12345678内存结构图：



1. 大端系统0x12345678内存图
   * 1. 带进位的循环右移

问题：带进位循环右移是什么意思

组成原理中带进位的循环右移是什么意思，怎么个移法？比如：10110101 ？

回答：

带进位的循环右移，若最初的进位位C为0，：

如果移1为的话..变成01011010..进位标志就为1

如果移2为的话..变成10101101..进位标志就为0

如果移3为的话..变成01010110..进位标志就为1

* + 1. ARM指令的寻址方式
       1. 概述

ARM指令的寻址主要分为两类：立即数寻址和Load/Store指令，立即数各类寻址方式大同小异，主要分为常数（即常数的循环右移）和寄存器（及寄存器移位）两种；类似，Load/Store指令的寻址方式也大同小异，具体表现为：基址+偏移的方式寻址，主要区别体现在偏移上——可以是常数或者寄存器。

此外，比较特别的是批量内存访问和栈访问，两者是在普通的Load/Store基础上增加自动改变基址的机制，实现批量（或者栈）的内存数据访问。

* + - 1. Load/Store指令

Load/Store指令分成以下两类：



寻址方式大体为：基址+偏移