表格 6-1 ARM 常用汇编指令讲解

|  |  |
| --- | --- |
| 指令名称 | 作用 |
| EQU | 给数字常量取一个符号名，相当于C语言中的define |
| AREA | 汇编一个新的代码段或者数据段 |
| SPACE | 分配内存空间 |
| PRESERVE8 | 当前文件栈需按照8字节对齐 |
| EXPORT | 声明一个标号具有全局属性，可被外部的文件使用 |
| DCD | 以字为单位分配内存，要求4字节对齐，并要求初始化这些内存 |
| PROC | 定义子程序，与ENDP成对使用，表示子程序结束。 |
| WEAK | 弱定义，如果外部文件声明了一个标号，则优先使用外部文件定义的  标号，如果外部文件没有定义也不出错。要注意的是：这个不是 ARM  的指令，是编译器的，这里放在一起只是为了方便。 |
| IMPORT | 声明标号来自外部文件，跟 C 语言中的 EXTERN 关键字类似 |
| B | 跳转到一个标号 |
| ALIGN | 编译器对指令或者数据的存放地址进行对齐，一般需要跟一个立即  数，缺省表示 4 字节对齐。要注意的是：这个不是 ARM 的指令，是  编译器的，这里放在一起只是为了方便。 |
| END | 到达文件的末尾，文件结束 |
| IF,ELSE,ENDIF | 汇编条件分支语句 |
|  |  |
| MRS | 加载特殊功能寄存器的值到通用寄存器 |
| MSR | 存储通用寄存器的值到特殊寄存器 |
| CBZ | 比较，如果结果为0就转移 |
| CBNZ | 比较，如果结果非0就转移 |
| LDR | 从存储器中加载字到一个寄存器中 |
| LDR[伪指令] | 加载一个立即数或者一个地址值到一个寄存器。举例：LDR Rd, = label，  如果 label 是立即数，那 Rd 等于立即数，如果 label 是一个标识符，比如指针，那存到 Rd 的就是 label 这个标识符的地址 |
| LDRH | 从存储器中加载半字到一个存储器中 |
| LDRB | 从存储器中加载字节到一个存储器中 |
| STR | 把一个寄存器按字存储到存储器中 |
| STRH | 把一个寄存器的低半字存储到存储器中 |
| STRB | 把一个寄存器的低字节存储到存储器中 |
| LDMIA | 加载多个字，并且在加载后自增基址寄存器 |
| STMIA | 存储多个字，并且在存储后自增基址寄存器 |
| ORR | 按位或 |
| BX | 直接跳转到寄存器给定的地址 |
| BL | 跳转到标号对应的地址，并且把跳转前的下条指令地址保存到LR |
| BLX | 跳转到由寄存器 REG 给出的的地址，并根据 REG 的 LSB 切换处理器要 把 转 移 前 的 下 条 指 令 地 址 保 存 到 LR 。ARM(LSB=0) ，  Thumb(LSB=1)。Cortex-M3 只在 Thumb 中运行，就必须保证 reg 的  LSB=1，否则一个 fault 打过来 |