|  |  |  |
| --- | --- | --- |
| ARM Reference  This page contains a reference for the ARMv6 instruction set, register set, and also the GNU Assembler program syntax.   |  | | --- | | **Contents**   * [1 Tutorial ARM Instructions](http://www.cl.cam.ac.uk/projects/raspberrypi/tutorials/os/armv6.html#tutorial) |   1 Tutorial ARM Instructions  The following is a list of all the instruction boxes in the courses in order.  ldr reg,=val puts the number val into the register named reg.  mov reg,#val puts the number val into the register named reg.  lsl reg,#val shifts the binary representation of the number in reg by val places to the left.  str reg,[dest,#val] stores the number in reg at the address given by dest + val.  name: labels the next line name.  b label causes the next line to be executed to be label.  sub reg,#val subtracts the number val from the value in reg.  cmp reg,#val compares the value in reg with the number val.  Suffix ne causes the command to be executed only if the last comparison determined that the numbers were not equal.  .globl lbl makes the label lbl accessible from other files.  mov reg1,reg2 copies the value in reg2 into reg1.  Suffix ls causes the command to be executed only if the last comparison determined that the first number was less than or the same as the second. Unsigned.  Suffix hi causes the command to be executed only if the last comparison determined that the first number was higher than the second. Unsigned.  push {reg1,reg2,...} copies the registers in the list reg1,reg2,... onto the top of the stack. Only general purpose registers and lr can be pushed.  bl lbl sets lr to the address of the next instruction and then branches to the label lbl.  add reg,#val adds the number val to the contents of the register reg.  Argument shift reg,lsl #val shifts the binary representation of the number in reg left by val before using it in the operation before.  lsl reg,amt shifts the binary representation of the number in reg left by the number in amt.  str reg,[dst] is the same as str reg,[dst,#0].  pop {reg1,reg2,...} copies the values from the top of the stack into the register list reg1,reg2,.... Only general purpose registers and pc can be popped.  alias .req reg sets alias to mean the register reg.  .unreq alias removes the alias alias.  lsr dst,src,#val shifts the binary representation of the number in src right by val, but stores the result in dst.  and reg,#val computes the Boolean and function of the number in reg with val.  teq reg,#val checks if the number in reg is equal to val.  ldrd regLow,regHigh,[src,#val] loads 8 bytes from the address given by the number in src plus val into regLow and regHigh.  .align num ensures the address of the next line is a multiple of 2num.  .int val outputs the number val.  tst reg,#val computes and reg,#val and compares the result with 0.  mla dst,reg1,reg2,reg3 multiplies the values from reg1 and reg2, adds the value from reg3 and places the least significant 32 bits of the result in dst.  strh reg,[dest] stores the low half word number in reg at the address given by dest. | ARM 参考  这一页包含一个针对ARMv6指令集、寄存器集的参考，同时还包括了GNU汇编器程序的语法。  目录  1 教程中用到的ARM指令   1. 教程中用到的ARM指令   下面所列是按照循序排列的课程中使用到的所有指令的列表。  ldr reg, =val 会把val数值存入名为reg的寄存器中。  mov reg, #val 会把val数值存入名为reg的寄存器中。  lsl reg, #val 会把保存在寄存器reg中的数值的二进制表示左移val个位置。  str reg, [dest, #val] 会把寄存器reg中的数值存储在由dest + val给定的地址中去。  name: 会把下一行标记为name标签。  b label 将会跳转到标签为label的地方去执行。  sub reg, #val 会把寄存器reg中的数值减去val数值。  cmp reg, #val 会把寄存器reg里的数值和val数值进行比较。  ne 后缀将会导致命令在满足下面条件执行：仅当最近一次比较的结果不相等时。  .globl lbl 将会让lbl标签可以被别的文件获取。  mov reg1, reg2 会把寄存器reg2的数值复制到寄存器reg1中。  ls 后缀将会导致命令在以下条件执行：仅当最近一次比较的结果为第一个数值小于或者等于第二个。无符号的比较。  hi 后缀将会导致命令在以下条件执行：仅当最近一次比较的结果为第一个数值高于第二个时。  push {reg1, reg2, …} 将会寄存器列表中的reg1, reg2, … 的数值复制到堆栈的顶部。只有通用目的寄存器和lr寄存器里的数值可以被压栈。  bl lbl 会把下一个指令的地址存入lr寄存器中，然后跳转到标签lbl处执行。  add reg, #val 会把寄存器reg中的数值加入数值val。  参数化 shift reg, lsl #val将会在一个操作之前，先对寄存器reg里的数值的二进制表示进行val个左移操作。  lsl reg, amt 将会把寄存器reg里的数值的二进制表示进行amt个左移。  str reg, [dst] 和指令str reg, [dst, #0]相同。  pop {reg1, reg2, …} 将会把保存在堆栈顶部的数值拷贝到寄存器列表reg1, reg2, …中。只有通用目的寄存器和pc寄存器可以被出栈。  alias .req reg 会把寄存器reg标记一个别名。  .unreq alias 会把别名alias移除。  lsr dst, src, #val 将会把保存在src中的数值的二进制表示进行val个单位右移，把结果保存在dst中。  and reg, #val 将会计算寄存器reg和val的布尔值。  teq reg, #val 将会检测寄存器reg里的数值是否等于val。  ldrd regLow, regHigh, [src, #val] 将会把src里的数值加上val所得结果作为地址，并从此处加载一个8字节数据到地址regLow和regHigh处。  .align num 将会确保下一行的地址是2的倍数。  .int val 输出val的数值。  tst reg, #val 将会计算寄存器reg里的数值和val数值的并，然后把结果和0进行比较。  mla dst, reg1, reg2, reg3 将会把寄存器reg1和reg2的数值进行相乘，然后加上reg3的数值，并加上符号位，把结果放在dst中。  strh reg, [dest] 将会寄存器reg的低半字保存在由dest给出的地址处。 |