华清远见 FARSIGHT 始于2004 | 嵌入式培训专家

10年口碑积累，成功培养50000多名研发工程师，铸就专业品牌形象

华清远见的企业理念是不仅要做良心教育、做专业教育，更要做受人尊敬的职业教育。

《ARM 系列处理器应用技术完全手册》

作者：华清远见

专业始于专注 卓识源于远见

第 8 章 跳转指令

本章目标

跳转（B）和跳转连接（BL）指令是改变指令执行顺序的标准方式。ARM 一般按照字地址顺序执行指令，需要时使用条件执行跳过某段指令。只要程序必须偏离顺序执行，就要使用控制流指令来修改程序计数器。尽管在特定情况下还有其他几种方式实现这个目的，但转移和转移连接指令是标准的方式。

专业始于专注 卓识源于远见

跳转指令改变程序的执行流程或者调用子程序。这种指令使得一个程序可以使用子程序、if-then-else 结构以及循环。执行流程的改变迫使程序计数器 PC 指向一个新的地址，ARMv5 架构指令集包含的跳转指令如表 8.1 所示。

表 8.1　　ARMv5 架构跳转指令

助记符	说　　明	操　　作
B	跳转指令	pc←label
BL	带返回的连接跳转	pc←label(lr←BL 后面的第一条指令)
BX	跳转并切换状态	pc←Rm&0xfffffffe, T←Rm&1
BLX	带返回的跳转并切换状态	pc←lable, T←1 pc←Rm&0xfffffffe, T←Rm&1 lr←BL 后面的第一条指令

另一种实现指令跳转的方式是通过直接向 PC 寄存器中写入目标地址值，实现在 4GB 地址空间中任意跳转，这种跳转指令又被称为长跳转。如果在长跳转指令之前使用“MOV　LR”或“MOV　PC”等指令，可以保存将来返回的地址值，也就实现了在 4GB 的地址空间中的子程序调用。

在 ARMv5 以前的版本中，传送到 PC 寄存器中的目标地址值的低两位 bits[1：0]被忽略，跳转指令只能在 ARM 指令集中执行，即程序不能从 ARM 状态切换到 Thumb 状态。在非 T 系列版本 5 的 ARM 体系不含 Thumb 指令，当程序试图切换到 Thumb 状态时，将产生未定义指令异常中断。

在 ARMv5 以后的版本中，有两种类型的带连接的跳转切换指令（BLX），叙述如下。

（1）形式如“BLX　<Rm>”，它是一种类似于带寄存器 Rm 的 BX 指令。指令执行 BX 操作，同时将返回地址放到 LR 寄存器中。这种形式的带状态切换的跳转连接指令，方便了 ARM/Thumb 互交的子程序调用。

（2）另一种类型的 BLX 指令类似于 BL 指令，指令使程序跳转到指定地址，并将返回地址保存到 LR 寄存器中，该指令能够实现 32MB 地址空间的跳转。与 BL 指令的不同之处在于它返回到 Thumb 状态，而不是 ARM 状态。

华清远见

8.1　跳转指令 B 及带连接的跳转指令 BL

1．指令编码格式

跳转指令 B 使程序跳转到指定的地址执行程序。带连接的跳转指令 BL 将下一条指令的地址拷贝到 r14（即返回地址连接寄存器 LR）寄存器中，然后跳转到指定地址运行程序。需要注意的是，这两条指令和目标地址处的指令都要属于 ARM 指令集。两条指令都可以根据 CPSR 中的条件标志位的值决定指令是否执行。指令的编码格式如图 8.1 所示。

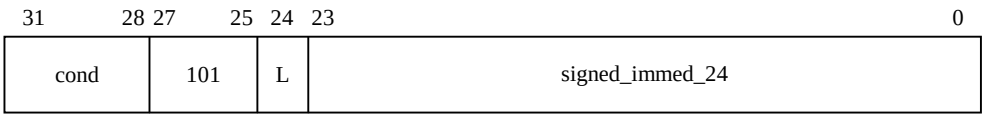

图 8.1　B&BL 指令编码格式

2．指令的语法格式

```
B{L}{<cond>}  <target_address>
```

①　<cond>

为指令编码中的条件域。它指示指令在什么条件下执行。当<cond>忽略时，指令为无条件执行（cond=AL（Alway））。

② L

L 位（bit[24]）=1，指令存储返回地址到 LR；L 位（bit[24]）=0，指令仅实现跳转，不保存返回指令。

③ <target_addrss>

指令跳转的目标地址。指令通过下面的方法计算目标地址。

- 将 24 位的立即数符号扩展为 32 位。
- 将扩展后的 32 位立即数左移两位。
- 将得到的值加到 PC 寄存器中，即得到跳转的目标地址。

注意 由于以上原因，B 和 BL 指令只能实现±32MB 空间的跳转。

3．指令操作的伪代码

指令操作的伪代码如下面程序段所示。

```
If  conditionPassed{cond}  then
    If  L==1  then
        LR = address of the instruction after the branch instruftion
    PC = PC + (SignExtend(signed_immed_24)<<2)
```

4．指令的使用

BL 指令用于实现子程序调用。子程序的返回可以通过将 LR 寄存器的值复制到 PC 寄存器来实现。下面三种指令可以实现子程序返回。

- BX　r14（如果体系结构支持 BX 指令）。
- MOV　PC，r14。
- 当子程序在入口处使用了压栈指令：

```
STMFD  r13!，{<registers>，r14}，
```

可以使用指令。

```
LDMFD  r13!，{<registers>，PC}
```

将子程序返回地址放入 PC 中。

ARM 汇编器通过以下步骤计算指令编码中的 signed_immed_24。

（1）将 PC 寄存器的值作为本跳转指令的基地址值。

（2）从跳转的目标地址中减去上面所说的跳转的基地址，生成字节偏移量。由于 ARM 指令是字对齐的，该字节偏移量为 4 的倍数。

（3）当上面生成的字节偏移量超过－33554432～+33554430 时，不同的汇编器使用不同的代码产生策略。

（4）否则，将指令编码字中的 signed_immed_24 设置成上述字节偏移量的 bits[25：2]。

注意 在一些 RISC 体系结构的处理器中，存在延时跳转（delayed branch）模式，即在程序执行跳转指令跳转到目标地址之前，程序会执行跳转指令之后的指令。但在 ARM 体系中，没有这种延时跳转机制。

5．指令举例

（1）程序跳转到 LABLE 标号处。

```
    B  LABLE ;
    ADD  r1，r2，＃4
    ADD  r3，r2，＃8
    SUB  r3，r3，r1
LABLE
    SUB  r1，r2，#8
```

（2）跳转到绝对地址 0x1234 处。

```
B  0x1234
```

（3）跳转到子程序 func 处执行，同时将当前 PC 值保存到 LR 中。

```
BL  func
```

（4）条件跳转：当 CPSR 寄存器中的 C 条件标志位为 1 时，程序跳转到标号 LABLE 处执行。

```
BCC  LABLE
```

（5）通过跳转指令建立一个无限循环。

```
LOOP
    ADD  r1，r2，＃4
    ADD  r3，r2，＃8
    SUB  r3，r3，r1
    B  LOOP
```

（6）通过使用跳转使程序体循环 10 次。

```
    MOV  r0，＃10
LOOP
    SUBS  r0，＃1
    BNE   LOOP
```

（7）条件子程序调用示例。

```
……
CMP  r0，＃5                ;如果 r0<5
BLLT  SUB1                  ;则调用
BLGE  SUB2                  ;否则调用 SUB2
```

注意 只有 SUB1 不改变条件码，本例才能正确执行，因为如果 BLLT 执行了转移，将返回到 BLGE 指令。如果条件码被 SUB1 子程序改变，则 SUB2 可能又会被执行，从而达不到指令的预期效果。

8.2 带状态切换的跳转指令 BX

1．指令编码格式

带状态切换的跳转指令 BX 使程序跳转到指令中指定的参数 Rm 指定的地址执行程序，Rm 的第 0 位拷贝到 CPSR 中 T 位，位[31：1]移入 PC。若 Rm 的 bit[0]为 1，则跳转时自动将 CPSR 中的标志位 T 置位，即把目标地址的代码解释为 Thumb 代码；若 Rm 的位 bit[0]为 0，则跳转时自动将 CPSR 中的标志位 T 复位，即把目标地址代码解释为 ARM 代码。

指令的编码格式如图 8.2 所示。

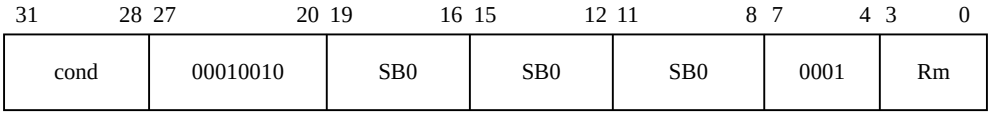

图 8.2　BX 指令编码格式

2．指令的语法格式

```
BX{<cond>}  <Rm>
```

① <cond>

为指令编码中的条件域。它指示指令在什么条件下执行。当<cond>忽略时，指令为无条件执行（cond=AL（Alway））。

② <Rm>

包含跳转指令的目标地址。如果 Rm 的 bit[0]=0，目标地址处指令为 ARM 指令；如果 Rm 的 bit[0]=1，目标地址处指令为 Thumb 指令。

3．指令操作的伪代码

指令操作的伪代码如下面程序段所示。

```
If  conditionPassed{cond}  then
    T  Flag=Rm[0]
    PC = Rm AND 0xfffffffe
```

4．指令的使用

- 当 Rm[1：0]=0b10 时，指令的执行结果不可预知。因为在 ARM 状态下，指令是 4 字节对齐的。
- PC 可以作为 Rm 寄存器使用，但这种用法不推荐使用。当 PC 作为<Rm>使用时，指令“BX　PC”将程序跳转到当前指令下面第二条指令处执行。虽然这样跳转可以实现，但最好使用下面的指令完成这种跳转。

```
MOV   PC, PC
```

或，

```
ADD   PC, PC, #0
```

5．指令举例

（1）转移到 r0 中的地址，如果 r0[0]=1，则进入 Thumb 状态。

```
BX   r0;
```

（2）跳转到 r0 指定的地址，并根据 r0 的最低位来切换处理器状态。

```
ADRL   r0，ThumbFun+1 ;
BX   r0;
```

8.3　带状态切换的连接跳转指令 BLX（1）

1．指令编码格式

带连接和状态切换的跳转指令 BLX（Branch with Link Exchange）使用标号，用于使程序跳转到 Thumb 状态或从 Thumb 状态返回。该指令为无条件执行指令，并用分支寄存器的最低位来更新 CPSR 中的 T 位，将返回地址写入到连接寄存器 LR 中。
指令编码格式如图 8.3 所示。

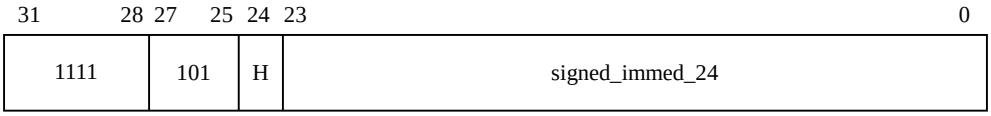

图 8.3　BLX(1)指令编码格式

2．语法格式

```
BLX  <target_add>
```

其中，<target_add>为指令的跳转目标地址。该地址根据以下规则计算。
① 将指令中指定的 24 位偏移量进行符号扩展，形成 32 位立即数。
② 将结果左移两位。
③ 位 H（bit[24]）加到结果地址的第一位（bit[1]）。
④ 将结果累加进程序计数器 PC 中。
计算偏移量的工作一般由 ARM 汇编器来完成。这种形式的跳转指令只能实现±32MB 空间的跳转。
左移两位形成字偏移量，然后将其累加进程序计数器 PC 中。这时，程序计数器的内容为 BX 指令地址加 8 字节。位 H（bit[24]）也加到结果地址的第一位（bit[1]），使目标地址成为半字地址，以执行接下来的 Thumb 指令。计算偏移量的工作一般由 ARM 汇编器来完成。这种形式的跳转指令只能实现±32MB 空间的跳转。

3．指令操作的伪代码

指令操作的伪代码如下面程序段所示。
第一种格式 BLX 指令。

```
LR=address of the instruction after the BLX instruction
T Flag=1
PC=PC + PC = PC + (SignExtend(signed_immed_24)<<2) + (H<<1)
```

4．指令的使用

- 从 Thumb 状态返回到 ARM 状态，使用 BX 指令。

```
BX  r14
```

- 可以在子程序的入口和出口增加栈操作指令。

```
PUSH  {<registers>，r14}
……
POP  {<registers>，PC}
```

8.4　带状态切换的连接跳转指令 BLX（2）

华清远见

1．指令编码格式

带连接和状态切换的跳转指令 BLX（Branch with Link Exchange）使用一个寄存器中的绝对地址，用于使程序跳转到 Thumb 状态或从 Thumb 状态返回。该指令用分支寄存器的最低位来更新 CPSR 中的 T 位，将返回地址写入到连接寄存器 LR 中。
指令编码格式如图 8.4 所示。

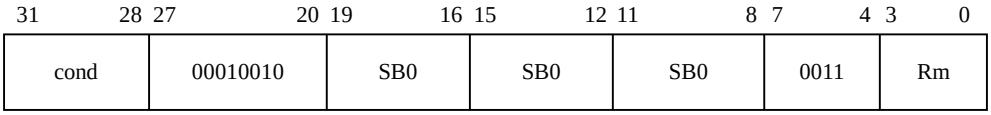

图 8.4　BLX(2)指令编码格式

2．语法格式

```
BLX{<cond>}  <Rm>
```

① <cond>
为指令编码中的条件域。它指示指令在什么条件下执行。当<cond>忽略时，指令为无条件执行（cond=AL（Alway））。
② <Rm>
寄存器 Rm 指定转移目标，Rm 的第 0 位拷贝到 CPSR 中的 T 位，bit[31：0]移入 PC。

- 如果 Rm 的 bit[0]=1，则跳转时自动将 CPSR 中的标志位 T 置位，即把目标地址的代码解释为 Thumb 代码。
- 如果 Rm 的 bit[0]=0，则跳转时自动将 CPSR 中的标志位 T 复位，即把目标地址代码解释为 ARM 代码。

3．指令操作的伪代码

指令操作的伪代码如下面程序段所示。

```
If  ConditionPass{cond}  then
      LR = address of the instruction after the branch instruction
      T Flag=Rm[0]
      PC=Rm AND 0xfffffffe
```

注意　在这种情况下，如果 Rm 的 bit[1：0]=0b10，指令的执行结果不可预知，因为这将导致在 ARM 状态下非对齐的字访问。

4．指令举例

调用 Thumb 子程序。

```
    CODE32                  ;ARM 代码
    ……
    BLX     TSUB            ;调用 Thumb 子程序
    ……
    CODE16                  ;Thumb 代码开始
TSUB
```

华清远见

```
……
BX      r14             ;返回 ARM 状态
```

注意 （1）一些不支持 Thumb 指令集的 ARM 处理器将捕获这些指令，允许软件仿真 Thumb 指令。
（2）只有实现 ARMv5 版本以上的处理器支持 BLX 指令的两种格式。

联系方式

集团官网：www.hqyj.com 嵌入式学院：www.embedu.org 移动互联网学院：www.3g-edu.org

企业学院：www.farsight.com.cn 物联网学院：www.topsight.cn 研发中心：dev.hqyj.com

集团总部地址：北京市海淀区西三旗悦秀路北京明园大学校内 华清远见教育集团

北京地址：北京市海淀区西三旗悦秀路北京明园大学校区，电话：010-82600386/5

上海地址：上海市徐汇区漕溪路银海大厦 A 座 8 层，电话：021-54485127

深圳地址：深圳市龙华新区人民北路美丽 AAA 大厦 15 层，电话：0755-22193762

成都地址：成都市武侯区科华北路 99 号科华大厦 6 层，电话：028-85405115

南京地址：南京市白下区汉中路 185 号鸿运大厦 10 层，电话：025-86551900

武汉地址：武汉市工程大学卓刀泉校区科技孵化器大楼 8 层，电话：027-87804688

西安地址：西安市高新区高新一路 12 号创业大厦 D3 楼 5 层，电话：029-68785218

华清远见