

## 第十八章

# 调用约定

本章描述了RV32和RV64程序的C编译器标准和两个约定：基本ISA和标准通用扩展（RV32G/RV64G），以及用于实现浮点单元的浮点单元（例如RV32I/RV64I）。

---

使用ISA扩展的实现可能需要结束调用约定。

### 18.1 C数据类型和对齐

表18.1总结了RISC-Vc程序原生支持的数据类型。在bothRV32和RV64c编译器中，C类型是32位转换。另一方面，长指针和指针都和整数寄存器一样宽，所以在RV32，都是32位宽，而在RV64，都是64位宽。同样，RV32采用ilp32整数模型，而RV64是LP64。在bothRV32和RV64中，C类型长是64位整数，浮动是32位IEEE754-2008浮点数，双是64位IEEE754-2008浮点数，长双是128位IEEE浮点数。

C类型字符和无符号字符是8位无符号整数，存储在aRISC-V整数寄存器时为零扩展。无符号短是16位无符号整数，存储在aRISC-V整数寄存器时为零扩展。有符号字符是一个8位有符号整数，在aRISC-V整数寄存器存储时为符号扩展，即位（XLEN-1）..7是等的。short是一个16位有符号的整数，当存储在寄存器中时是有符号扩展。

InRV64,32位类型，这样的s int，被存储在整数寄存器中，作为其32位值的适当符号扩展；也就是说，位63..31是正确的。这个限制对无符号的32位类型保留了7个。

rv32和rv64C编译器和兼容软件在存储上述所有数据类型时自然对齐。

棉型	描述	rv32中的字节	rv64中的字节
焦油	字节特征值	1	1
短的	短整数	2	2
int	整数	4	4
长时间地	长整数	4	8
长长	长整数	8	8
void*	建议	4	8
使漂浮	单先决条件浮动	4	4
两倍的	双先决条件浮动	8	8
长双	扩展的先决条件浮动	16	16

表18.1：baseRISC-VISA的C编译器数据类型。

## 18.2 RVG呼叫约定

risc-v调用约定在可能的情况下在寄存器中传递参数。最多使用8个积分寄存器，0-a7和8个浮点器fa0-fa7。

如果函数的参数是C结构的域，每个都有点对齐，参数寄存器是该结构的前八个指针字的影子。如果参数i<8是浮点类型，则传递浮点寄存器fai；否则，它将以整数寄存器ai的形式传递。但是，作为联合或结构域的浮点参数在整数寄存器中传递。此外，对并进函数的浮点参数（在参数列表中显式命名的参数除外）将在整数寄存器中传递。

较小的参数指针字在参数寄存器的最小重要位中传递。相应地，子指针字参数在堆栈中出现在指针字的较低地址中，因为RISC-V具有小端记忆系统。

当在堆栈上传递指针字大小的原始参数时，它们按顺序对齐。当它们在整数寄存器中传递时，它们位于一个连接偶数注册对中，偶数寄存器持有最显著位。例如，InRV32，函数（int，长）的第一个参数为0，第二个参数为2and3。在1中什么都没有通过。

参数传递的大小是指针字的两倍多。

未在参数寄存器中传递的概念结构的部分将在堆栈上传递。堆栈点rsp指向未在区域中传递的第一个参数。

值从整数寄存器sa0and1和浮点注册寄存器sfa0和fa1中的函数返回。只有当浮点值是启动项或只包含一个或两个加载点值的结构体c的成员时，才会返回入浮点寄存器。其他适合两个两个的返回值以0和1返回。较大的返回值完全在内存中传递；调用者a1定位这个内存区域，并将一个指针作为第一个隐含参数传递给它。

在标准化的risc-v调用约定中，堆栈gro向下移动，堆栈指针总是保持16个字节对齐。

除了参数和返回值寄存器之外，7个整数注册器0-t6和12个浮点注册器sft0-ft11是临时寄存器，它们在调用之间是不稳定的，如果以后使用，必须由调用器保存。12个整数注册ss0-s11和12个浮点注册sfs0-fs11在调用中保存，并且必须由被调用者保存。表18.2显示了呼叫约定中每个整数和浮点寄存器的角色。

注册	ABI名称	描述	保护程序
x0	零	硬接线零	—
x1	ra	返回地址	呼叫者
x2	sp	堆栈指针	卡莱
x3	gp	全局指针	—
x4	tp	线程指针	—
x5-7	t0-2	时间	呼叫者
x8	s0/ fp	SaveDregister/帧指针	卡莱
x9	s1	已保存的寄存器	卡莱
x10-11	a0-1	函数参数/返回值	呼叫者
x12-17	a2-7	函数参数	呼叫者
x18-27	s2-11	保存的寄存器	卡莱
x28-31	t3-6	时间	呼叫者
f0-7	ft0-7	FP时间	呼叫者
f8-9	fs0-1	FP保存寄存器	卡莱
f10-11	fa0-1	F参数/返回值	呼叫者
f12-17	fa2-7	FP参数	呼叫者
f18-27	fs2-11	FP保存寄存器	卡莱
f28-31	ft8-11	FP时间	呼叫者

Table18.2：RISC-V调用约定注册表的用法。

18.3软浮子呼叫约定

这种频率-浮点调用约定在RV32和RV64实现上使用，它们缺乏浮点硬件。它避免了F、D和Q标准扩展中的教学使用，从而避免了分寄存器。

积分参数以与RVG约定相同的方式传递和返回，并且堆栈规则是相同的。浮点参数被经过并返回到整数寄存器中，使用相同大小的整数参数句点的规则。例如，在rv32中，双函数（int，双，长双）的第一个参数为0，第二个参数为2和3，第三个参数通过引用viaa4；它的结果以0和da1返回。在InRV64中，参数以0、a1和a2-a3对传递，结果以0返回。

通过提供的例程访问动态舍入模式和应计的异常标志

由c99headerfenv.h。