

RV64I 基本整数说明，按字母顺序排列

助记符	FMT '姓名	描述（在Verilog中）	注释
添加, 添加	R ADD（单词）	R[rd] = R[rs1]+R[rs2]	1)
addi, addiw	I ADD立即数（Word）	R[rd] = R[rs1] + imm	1)
和	R 和	R[rd] = R[rs1] & R[rs2]	
安迪	I 与立即数	R[rd] = R[rs1] & imm	
auipc	U 将上立即数添加到 PC	R[rd] = PC + {imm, 12'b0}	
贝克	小型 分支 EQual	如果 {R[rs1]=R[rs2]} PC=PC+{imm,lb'0}	
贝格	小型 分支大于或等于	if {R[rs1]>=R[rs2]} PC=PC+{imm,lb'0}	
伯格	小型 分支 > 无符号	如果 {R[rs1]>=R[rs2]} PC=PC+{imm,lb'0}	2)
钻头	小型 分支小于	如果 {R[rs1]<R[rs2]} PC=PC+{imm,lb'}	
蓝色的	小型 小于无符号的分支	如果 {R[rs1]<R[rs2]} PC=PC+{imm,lb'}	2)
贝思	小型 分支不相等	if {R[rs1]!=R[rs2]} PC=PC+{imm,lb'0}	
中国证监会	I Cont./Stat.RegRead&Clear	R[rd] = CSR;CSR = CSR & ~R[rs1]	
csrrci	I Cont./Stat.RegRead&Clear Imm	R[rd] = CSR;CSR = CSR & -imm	
csrrs	I Cont./Stat.RegRead&Set（连	R[rd] = CSR;CSR = CSR   R[rs1]	
csrrsi	I Cont./Stat.RegRead&Set Imm	R[rd] = CSR;CSR = CSR   imm	
csrrw	I Cont./Stat.Reg读写	R[rd] = CSR;CSR = R[rs1]	
csrrwi	I Cont./Stat.Reg读写Imm	R[rd] = CSR;CSR = imm	
裂口	I 环境中断	将控制权转移到调试器	
召唤	I 环境调用	将控制权转移到操作系统	
栅栏	I 同步螺旋	同步线程	
围栏.i	I 同步指令和数据	同步写入指令流	
贾尔	乌杰 跳转和链接	R[rd] = PC+4;PC = PC + {imm,lb'0}	
贾尔	I 跳转和链接寄存器	R[rd] = PC+4;PC = R[rs1]+imm	3)
磅	I 加载字节	R[rd] = [56bM[]（7），M[R[rs1]+imm]（7; 0	4)
伊布	I 加载无符号字节	R[rd] =	
编号	I 加载双字	R[rd] = M[R[rs1]+imm]（63; 0）	
伊赫	I 加载半字	R[rd] = [48bM[]（15），M[R[rs1]+imm]（15	4)
伊胡	I 加载无符号半字	R[rd] =	
吕伊	U 加载立即数上限	R[rd] = {32bimm<31>, imm,12'b0}	
伊瓦	I 加载字	R[rd] = [32bM[]（3	4)
伊武	I 加载无符号字	R[rd] =	
或	R 或	R[rd] = R[rs1]   R[rs2]	
ori	I OR 立即	R[rd] = R[rs1]   imm	
某人	S 存储字节	M[R[rs1]+imm]（7; 0） =	
标准差	S 存储双字	M[R[rs1]+imm]（63; 0） =	
什	S 存储半字	M[R[rs1]+imm]（15; 0） =	
sll, sllw	R 左移（Word）	R[rd] = R[rs1] << R[rs2]	1)
slli, slliw	I 左移立即数（Word）	R[rd] = R[rs1] <<imm	1)
坐	R 设置小于	R[rd] = {R[rs1] < R[rs2]} ?1: 0	
索尔蒂	I 设置小于立即数	R[rd] = {R[rs1] < imm} ?1: 0	
萨尔蒂尤	I 设置 < 无符号立即数	R[rd] = {R[rs1] <imm} ?1: 0	2)
原位的	R 设置小于无符号	R[rd] = {R[rs1]<R[rs2]} ?1: 0	2)
sra, sraw	R 算术右移（Word）	R[rd] = R[rs1] » R[rs2]	1,5)
srai, sraiw	I 右移算子Imm（单词）	R[rd] = R[rs1] » imm	1,5)
srl, srlw	R 右移（Word）	R[rd] = R[rs1] » R[rs2]	1)
slli, srlwi	I 右移立即数（Word）	R[rd] = R[rs1] »imm	1)
sub, subw	R SUBtract（Word）	R[rd] = R[rs1] - R[rs2]	1)
SW	S 存储 Word	M[R[rs1] +imm]（31; 0） =	
异或	R 异或	R[rd] = R[rs1] ^ R[rs2]	
肖里	I XOR 立即数	R[rd] = R[rs1] ^ imm	

注: 1) Word 版本只在 64 位寄存器的最右 32 位上运行  
2) 运算假定无符号整数（而不是 2 的补码）  
3) jalr 中分支地址的最低有效位设置为0  
4)（有符号）加载指令扩展数据的符号位以填充 64 位寄存器  
5) 在右移过程中，复制符号位以填充结果的最左边位  
6) 与一个有符号操作数和一个无符号操作数相乘  
7) Single 版本使用 64 位 F 寄存器的最右 32 位执行单精度运算  
8) Classify 写入 1 个 0 位掩码以显示哪些属性为真（例如，-inf,-0,+0,+inf denorm,...）  
9) 原子存储器操作: 在存储器位置的读取和写入之间没有任何其它东西可以插入自身，立即字段在 RISC-V 中是符号扩展的

算术核心指令集（D  
RV64M 乘法扩展

助记符	FMT 名称	描述（在Verilog中）	注释
mul, mulw	R MULiPly（单词）	R[rd] = R[rs1] * R[rs2]（63; 0）	1
灌木	R MUL极高	R[rd] = {R[rs1] * R[rs2]}（127; 64）	
木依湖	R MUL极高无符号	R[rd] = {R[rs1] * R[rs2]}（127; 64）	2
穆尔苏	R MULiPly 上半符号/Uns	R[rd] = {R[rs1] * R[rs2]}（127; 64）	6
div, divw	R 划分（Word）	R[rd] = R[rs1]/R[rs2]	1
草皮	R 无符号除法	R[rd] = {R[rs1]/R[rs2]}	2
拆除, 拆除	R 剩余部分（单词）	R[rd] = {R[rs1] %R[rs2]}	1
雷姆, 雷穆	R	R[rd] = {R[rs1] % R[rs2]}	1,2
	剩余部分无符号（Word）		

RV64F 和 RV64D 浮点扩展

fid, flw	I 负载（Word）	F[rd] = M[R[rs1]+Imm]	1
fsd, fsw	S 存储（Word）	M[R[rs1]+i+Imm] = F[rd]	1
fadd.s, fadd.d	R 添加	F[rd]=F[rs1]+F[rs2]	7
fsub.s, fsub.d	R 减法	F[rd] = F[rs1]-F[rs2]	7
fmul.s, fmul.d	R MULiPly	F[rd] = F[rs1] * F[rs2]	7
fdiV.s, fdiV.d	R 划分	F[rd] = F[rs1]/F[rs2]	7
fsqrt.s, fsqrt.d	R 平方RooT	F[rd] = 平方根（F[rs1]）	7
fmadd.s, fmadd.d	R 乘加	F[rd] = F[rs1] * F[rs2] + F[rs3]	7
fmsub.s, fmsub.d	R 乘减	F[rd] = F[rs1] * F[rs2] - F[rs3]	7
fnmadd.s, fnmadd.d	R 负乘法-加法	F[rd] = <-F[rs1] * F[rs2] + F[rs3]	7
f nmsub.s, s, f nmsub.d	R 负乘减 F[rd] = -（F[rs 1] * F[rs2] - F[rs3]）		7
fsgnj.s, fsgnj.d	R SIGN源	F[rd] = { F[rs2]<63>, F[rs1]<62:0>}	7
fsgnjn.s, fsgnjn.d	R 负SIGN源	F[rd] = {（-F[rs2]<63>），F[rs1]<62:0>}	7
fsgnjx.s, fsgnjx.d	R X或 SIGN 源	F[rd] = {F[rs2]<63>房侧[rs1]<63>, F[rs1]<62:0>}	7
fmin.s, fmin.d	R 最小值	F[rd] = {F[rs1] < F[rs2]} ?F[rs1]: F[rs2]	7
fmax.s, fmax.d	R 最大值	F[rd] = {F[rs1] > F[rs2]} ?F[rs1]: F[rs2]	7
次, 次	R 比较浮点等值	R[rd] = {比较浮点== F[rs2]} ?1: 0	7
拟合 s, 拟合 d	R 比较小于的浮点值	R[rd] = {F[rs1]<F[rs2]} ?1: 0	7
fie.s, fie.d	R 比较浮点小于或等于分类类型	R[rd] = {F[rs1]<= F[rs2]} ?1: 0	7
fclass.s, fclass.d	R 从整数移动	R[rd] = 类别（F[rs1]）	7,8
fmv.s,x, fmv.d,x	R 移至整数	F[rd] = R[rs1]	7
fcvt.s,d	R 从 DP 转换为 SP	R[rd]=F[rs1]	7
fcvt.d,s	R 从 SP 转换为 DP	F[rd] = 双倍（F[rs1]）	
fcvt.s,w,fcvt.d,w	R 从 32b 整数转换	F[rd] = 浮点（R[rs1]（31; 0））	7
fcvt.s,l,fcvt.d,l	R 从 64b 整数转换	F[rd] = 浮点（R[rs1]（63; 0））	7
fcvt.s,wu,fcvt.d,wu	R 从 32b 内部无符号转换	F[rd] = 浮点（R[rs1]（31; 0））	2,7
fcvt.s,lu,fcvt.d,lu	R 从 64b 内部无符号转换	F[rd] = 浮点（R[rs1]（63; 0））	2,7
fcvt.宽.s,fcvt.宽.d	R 转换为 32b 整数	R[rd]（31; 0）= 整数（F[rs1]）	7
fcvt.l,s,fcvt.l,d	R 转换为 64b 整数	R[rd]（63; 0）= 整数（F[rs1]）	7
fcvt.wu.s,fcvt.wu.d	R 转换为 32b Int 无符号 R[rd]（31; 0）= integer（F[rs1]）		2,7
fcvt.lu.s,fcvt.lu.d	R 转换为 64b Int 无符号 R[rd]（63; 0）= integer（F[rs1]）		2,7

RV64A 原子扩展

amoadd.w, amoadd.d	R 添加	R[rd] = M[R[rs1]], M[R[rs1]] = M[R[rs1]] + R[rs2]	9
amoand.w, amoand.d	R 和	R[rd] = M[R[rs1]], M[R[rs1]] = M[R[rs1]] & R[rs2]	9
amomax.w, amomax.d	R 最大值	R[rd]=M[R[rs1]], 如果 {R[rs2] > M[R[rs1]]} M[R[rs1]] = R[rs2]	9
amomaxu.w, amomaxu.d	R 最大无符号	R[rd] = M[R[rs1]], 如果 {R[rs2] >M[R[rs1]]} M[R[rs1]] = R[rs2]	2,9
amomin.w, amomin.d	R 最小值	R[rd] = M[R[rs1]], 如果 {R[rs2] < M[R[rs1]]} M[R[rs1]] = R[rs2]	9
amominu.w, amominu.d	R 最小无符号	R[rd] = M[R[rs1]], 如果 {R[rs2] < M[R[rs1]]} M[R[rs1]] = R[rs2]	2,9
aoor.w, amoor.d	R 或	R[rd]=M[R[rs1]], M[R[rs1]] = M[R[rs1]]   R[rs2]	9
amoswap.w, amoswap.d	R 交换	R[rd] = M[R[rs1]], M[R[rs1]] = R[rs2]	9
阿莫素 .w, 阿莫素 .d	R 异或	R[rd]=M[R[rs1]], M[R[rs1]] = M[R[rs1]] ^ R[rs2]	9
Ir.w, Ir.d	R 预留负载	R[rd]=M[R[rs1]], M[R[rs1]]上的预留 如果保留, 则M[R[rs1]]=R[rs2], R[rd]=0;否则R[rd] = 1	
标准差, 标准差	R 存储条件		

核心指令格式

31	27	26	25	24	20	19	15	14	12	11	7	6	0
功能7				rs2		无线电		功能3		第三次		操作码	
imm[]ll: 0]						无线电		功能3		第三次		操作码	
imm[]ll: 5]				rs2		无线电		功能3		imm[]4: 0]		操作码	
imm[]12[]0: 5]				rs2		无线电		功能3		imm[]4: 1[]]		操作码	
				imm[]31: 12]						第三次		操作码	
				imm[]20[]10: 1[]]		19: 12]				第三次		操作码	

伪说明

助记符	姓名	描述	用途
贝克兹	分支	if ( R[rs1] != 0 ) PC=PC+(imm,sh0)	贝克
布内兹	分支 # 0	if ( R[rs1]==0 ) PC=PC+(imm,sh0)	贝恩
fab.s, fab.s	i 绝对值	F[rd] = ( F[rs1] < 0 ) ? -F[rs1]: F[rs1]	fsgnux 系
f mv.s, fmv.d	FP 移动	F[rd] = F[rs1]	f sgnj
fneg.s, fneg.	i FP取反	F[rd] = -F[rs1]	
j	跳跃	PC = (imm,sh0)	杰尔
■	跳转寄存器	PC = R[rs1]	贾尔
拉	加载地址	R[rd] = 地址	auipc
里	加载 imm	R[rd] = imm	addi
毫伏	移动	R[rd] = R[rs1]	addi
阴性	否定	R[rd] = -R[rs1] I	次级
诺普	无操作	R[0] = R[0]	addi
不	否	R[rd] = ~R[rs1] I	肖里
保留	返回	PC = R[0]	贾尔
序号	设置 = zero	R[rd] = ( R[rs1] != 0 ) ? 1: 0	萨尔蒂尤
偷偷摸摸	置零	R[rd] = ( R[rs1] == 0 ) ? 1: 0	原位的

按OPC码按数字顺序排列的OPCODE

FMT	OPCODE	功能3	功能70RIMM	十六进制
磅	I	0000011	000	03/0
伊赫	I	0000011	001	03/1 日
伊瓦	I	0000011	010	2 月 3 日
编号	I	0000011	油	3 月 3 日
伊布	I	0000011	100	03/4
伊胡	I	0000011	101	03/5
伊武	I	0000011	谎吉	03 年 6 月
栅栏	I	0001111	000	0F/0
围栏.i	I	0001111	001	0F/1
addi	I	0010011	000	0 月 13 日
索利	I	0010011	001	0000 0000 2000 年 1 月
索尔蒂	I	0010011	010	2 月 13 日
萨尔蒂尤	I	0010011	油	3 月 13 日
	I	0010011	100	四月 13 日
斯里利	I	0010011	101	0000 0000 2000 年 5 月
斯拉伊	I	0010011	101	0100 0000 2020 年 5 月
ori	I	0010011	110	6 月 13 日
安迪	I	0010011	111	7 月 13 日
auipc	U	0010111		17
附加	I	0011011	000	1 个月/0
萨利尤	I	0011011	001	0000 0000 1B/4/00
srliw	I	0011011	101	0000 0000 1B/5/00
srariw	I	0011011	101	0100 0000 1B/5/20
某人	I	0100011	000	0 月 23 日
什	S	0100011	001	23/1
SW	S	0100011	010	23/2
标准差	C	0100011	油	3 月 23 日
加上	R	0110011	000	0000 0000 2000 年 3 月 3
次级	R	0110011	000	0100 0000 2020 年 3 月 3
sll	R	0110011	001	0000 0000 2000 年 1 月
坐	R	0110011	010	0000 0000 2000 年 2 月
原位的	R	0110011	油	0000 0000 2000 年 3 月
异或	R	0110011	100	0000 0000 2000 年 4 月
srl	R	0110011	101	0000 0000 2000 年 5 月
sra	R	0110011	101	0100 0000 2020 年 5 月
或	R	0110011	谎吉	0000 0000 2000 年 6 月
和	R	0110011	111	0000 0000 2000 年 7 月
吕伊	U	0110111		37
添加	R	0111011	000	0000 0000 2000 年 2 月 3
潜艇	R	0111011	000	0100 0000 2000 年 2 月 3
slilw	R	0111011	001	0000 0000 2000 年 1 月 3
srilw	R	0111011	101	0000 0000 3B/5/00
稻草	R	0111011	101	0100 0000 2020 年 5 月 3
贝克	小型	1100011	000	63/0
贝恩	小型	1100011	001	第 63/1 页
钻头	小型	1100011	100	63/4 号
贝格	小型	1100011	101	63/5
蓝色的	小型	1100011	110	第 63/6 页
伯格	小型	1100011	111	63/7
贾尔	I	1100111	000	67/0
井西	乌杰	1101111		6F
召唤	I	1110011	000	0000 0000 0000 0000 2000 年 3 月
裂口	I	1110011	000	0000 0000 0000 0001 2001 年 3 月
科学研究院	I	1110011	001	73/1
CSRRS	I	1110011	010	73/2
中国社会责任委	I	1110011	油	73/3
CSRRWI	I	1110011	: j	73/5
CSRRSI	I	1110011	110	73/6 年
中国科学院	I	1110011	11 -	73/7 年

注册名称、用途、召集会议

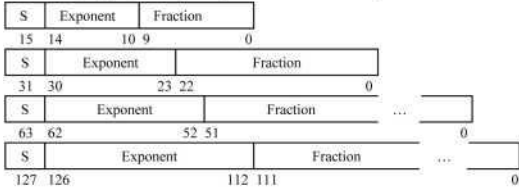
④

注册	姓名	用途	拯救者
x0	零	常数0	不适用
扩展逻辑	瑞安	寄存器地址	呼叫者
x2	sp	堆栈指针	被呼叫者
x3	总计	全局指针	
x4	转换点	螺纹指针	
x5-x7	t0-t2	临时	呼叫者
x8	e0/ep	保存的寄存器/帧指针	被呼叫者
x9	单位	已保存的寄存器	被呼叫者
十一~十一	a0-a1	函数参数/保留值	呼叫者
X12-X17	a2-a7	函数参数	呼叫者
x18-x27	s2-s11	已保存的寄存器	被呼叫者
x28 至 x31	t3-t6	临时	呼叫者
f0-f7	英尺-英尺 7	FP临时工	呼叫者
f8-f9	fs0-fs1	FP 保存的寄存器	被呼叫者
非奥-菲尔	fa0-fa1	FP 函数参数/保留值	呼叫者
f12-f17	fa2-fa7	FP 函数参数	呼叫者
f18-f27	fs2-fs11	FP 保存的寄存器	被呼叫者
128-女性 31	ft0-ft11	Rfnd = R[rs1] + R[rs2]	呼叫者

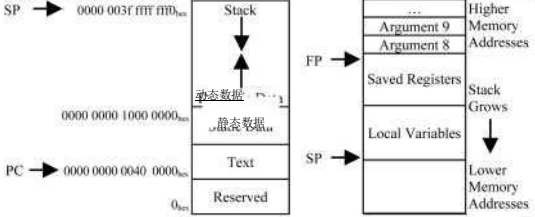
IEEE 754浮点标准

(-1) s x (1 + 分数) x 2<r-P"nc", -B""  
其中，半精度偏倚 = 15,单精度偏倚 = 127,双精度偏倚 = 1023,四精度偏倚 = 16383

IEEE 半精度、单精度、双精度和四精度格式:



MEMORY ALLOCATION



SIZE PREFIXES AND SYMBOLS

尺寸	前缀	符号	尺寸	前缀	符号
103	千瓦	K	20	基比	基
106	巨型	M	2211	海比	米
10英尺	千兆	G	230	吉比	基
1012	特拉	T	240	特比	狄
101'	佩塔	P	2英寸	百比	皮
IOIS公司	Exa-	E	260	Exbi-	埃伊
1021	泽塔	Z	2TM	泽比	紫
第024条	约塔	Y	2TM	尤比	易
10英尺3英	毫微米	m	10英寸	毫微微	f
106	微小的	p	10英寸	atto	a
10英尺	纳米	n	IO21	zepto	z
2012年10	微微	p	IO24	yocto	y

RISC-V参考数据卡(“绿卡”) 1.沿穿孔拉动, 分离卡2. 将底侧(第3列和第4列) 折叠在一起