

Fiche de référence des instructions assembleur RISC-V RV32I *

Jérémie Crenne

2021

31	30	25	24	21	20	19	15	14	12	11	8	7	6	0			
funct7				rs2			rs1		funct3		rd			opcode		R-type	
imm[11 :0]						rs1		funct3		rd			opcode		I-type		
imm[11 :5]				rs2			rs1		funct3		imm[4 :0]			opcode		S-type	
imm[12]		imm[10 :5]			rs2			rs1		funct3		imm[4 :1]		imm[11]		opcode	B-type
imm[31 :12]										rd			opcode		U-type		
imm[20]		imm[10 :1]			imm[11]		imm[19 :12]				rd			opcode		J-type	

FIGURE 1 – Formats des instructions RISC-V R, I, S, B, U et J

*La spécification RISC-V complète est disponible au format PDF à l'adresse suivante : <https://riscv.org/technical/specifications/> dans la rubrique ISA Specification, Volume 1, Unprivileged Spec v.

Instruction	Opérandes	Type	Description	Note(s)
add	rd, rs1, rs2	R	$R[rd] = R[rs1] + R[rs2]$	-
sub	rd, rs1, rs2	R	$R[rd] = R[rs1] - R[rs2]$	-
slt	rd, rs1, rs2	R	$R[rd] = R[rs1] <_s R[rs2]$	-
sltu	rd, rs1, rs2	R	$R[rd] = R[rs1] <_u R[rs2]$	-
and	rd, rs1, rs2	R	$R[rd] = R[rs1] \& R[rs2]$	-
or	rd, rs1, rs2	R	$R[rd] = R[rs1] R[rs2]$	-
xor	rd, rs1, rs2	R	$R[rd] = R[rs1] \wedge R[rs2]$	-
sll	rd, rs1, rs2	R	$R[rd] = R[rs1] << R[rs2]$	-
srl	rd, rs1, rs2	R	$R[rd] = R[rs1] >> R[rs2]$	-
sra	rd, rs1, rs2	R	$R[rd] = R[rs1] >>_s R[rs2]$	-
addi	rd, rs1, immediate	I	$R[rd] = R[rs1] + \text{sext}(\text{immediate})$	-
slti	rd, rs1, immediate	I	$R[rd] = R[rs1] <_s \text{sext}(\text{immediate})$	-
sltiu	rd, rs1, immediate	I	$R[rd] = R[rs1] <_u \text{sext}(\text{immediate})$	-
andi	rd, rs1, immediate	I	$R[rd] = R[rs1] \& \text{sext}(\text{immediate})$	-
ori	rd, rs1, immediate	I	$R[rd] = R[rs1] \text{sext}(\text{immediate})$	-
xori	rd, rs1, immediate	I	$R[rd] = R[rs1] \wedge \text{sext}(\text{immediate})$	-
slli	rd, rs1, shamt	I	$R[rd] = R[rs1] << \text{shamt}$	-
srli	rd, rs1, shamt	I	$R[rd] = R[rs1] >>_u \text{shamt}$	-
srai	rd, rs1, shamt	I	$R[rd] = R[rs1] >>_s \text{shamt}$	-
lb	rd, offset(rs1)	I	$R[rd] = \text{sext}(M[R[rs1] + \text{sext}(\text{offset})][7:0])$	-
lbu	rd, offset(rs1)	I	$R[rd] = \text{sext}(M[R[rs1] + \text{sext}(\text{offset})][7:0])$	non signé
lh	rd, offset(rs1)	I	$R[rd] = \text{sext}(M[R[rs1] + \text{sext}(\text{offset})][15:0])$	-
lhu	rd, offset(rs1)	I	$R[rd] = \text{sext}(M[R[rs1] + \text{sext}(\text{offset})][15:0])$	non signé
lw	rd, offset(rs1)	I	$R[rd] = \text{sext}(M[R[rs1] + \text{sext}(\text{offset})][31:0])$	-
mv	rd, rs1	I	$R[rd] = R[rs1]$	pseudo-instruction
li	rd, immediate	I	$R[rd] = \text{immediate}$	pseudo-instruction
ble	rs1, rs2, offset	I	if (rs1 \leq_s rs2) pc += sext(offset)	pseudo-instruction
beqz	rs1, offset	I	if (rs1 = 0) pc += sext(offset)	pseudo-instruction
bnez	rs1, offset	I	if (rs1 \neq rs2) pc += sext(offset)	pseudo-instruction
jalr	rd, offset(rs1)	I	t = pc + 4; pc = (R[rs1] + sext(offset)) & ~1; R[rd] = t	-
sb	rs2, offset(rs1)	S	$M[R[rs1] + \text{sext}(\text{offset})] = R[rs2][7:0]$	-
sh	rs2, offset(rs1)	S	$M[R[rs1] + \text{sext}(\text{offset})] = R[rs2][15:0]$	-
sw	rs2, offset(rs1)	S	$M[R[rs1] + \text{sext}(\text{offset})] = R[rs2][31:0]$	-
beq	rs1, rs2, offset	B	if (rs1 == rs2) pc += sext(offset)	-
bne	rs1, rs2, offset	B	if (rs1 \neq rs2) pc += sext(offset)	-
blt	rs1, rs2, offset	B	if (rs1 $<_s$ rs2) pc += sext(offset)	-
bltu	rs1, rs2, offset	B	if (rs1 $<_u$ rs2) pc += sext(offset)	-
bge	rs1, rs2, offset	B	if (rs1 \geq_s rs2) pc += sext(offset)	-
bgeu	rs1, rs2, offset	B	if (rs1 \geq_u rs2) pc += sext(offset)	-
lui	rd, immediate	U	$R[rd] = \text{sext}(\text{immediate}[31:12] << 12)$	-
auipc	rd, immediate	U	$R[rd] = \text{pc} + \text{sext}(\text{immediate}[31:12] << 12)$	-
j	offset	J	pc += sext(offset)	pseudo-instruction
jal	rd, offset	J	$R[rd] = \text{pc} + 4$; pc += sext(offset)	-
call	rd, symbol	J	$R[rd] = \text{pc} + 8$; pc = @symbol. Si pas de rd, x1 est utilisé	pseudo-instruction
ret		J	équivalent à jalr x0, 0(x1)	pseudo-instruction

TABLE 1 – Descriptif des instructions RV32I. En gris, les instructions non étudiées en cours

Notation	Signification	Note(s)
rd	registre de destination	adresse du registre sur 5 bits
rs	registre source	adresse du registre sur 5 bits
R[addr]	file de registres	adressage sur 5 bits par <i>addr</i>
M[addr]	mémoire de données	adressage sur n bits par <i>addr</i>
CSR[addr]	registres de contrôle et d'état	adressage sur n bits par <i>addr</i>
[x:y]	opérateur d'extraction des bits y à x	-
offset(a)	décalage par rapport à a	-
immediate	valeur constante codée dans l'instruction elle même	voir champ imm Figure 1
sext(a)	fonction d'extension de signe de a (sign extension)	-
shamt	nombre de décalages en bits (shift amount)	-
pc	registre du compteur programme (program counter)	pointe sur la prochaine instruction à exécuter
<i>u</i>	opérateur non signé	-
<i>s</i>	opérateur signé	-
(a)&~1	masque le bit le moins significatif de a	-
@a	adresse de a	-
&	et logique	-
 	ou logique	-
^	ou exclusif logique	-
a<<n	décalage de a de n bit vers la gauche	-
a>>n	décalage de a de n bit vers la droite	-

TABLE 2 – Signification des notations employées dans la Table 1

Nom symbolique	Registre	Description	Sauvegarde
a0-a7	x10-x17	Arguments de fonction	Appelant
a0-a1	x10-x11	Valeurs de retour de fonction	Appelant
ra	x1	Adresse de retour	Appelant
t0-t6	x5-x7, x28-x31	Temporaires	Appelant
s0/fp	x8	Registre sauvegardé/pointeur de frame	Appelé
s1-s11	x9, x18-x27	Registres sauvegardés	Appelé
sp	x2	Pointeur de pile	Appelé
gp	x3	Pointeur global	-
tp	x4	Pointeur de thread	-
zero	x0	'0' câblé	-

TABLE 3 – Noms donnés aux registres RISC-V et leur rôle dans la convention d'appel (ABI)