RISC-V RV32IM binarni zapis instrukcija

Ime i prezime studenta: _____ Potpis studenta: _____

Potpis dežurnog: _____ Datum ispita: _____

R tip: registar, registar

	31..25	24..20	19..15	14..12	11..7	6..0	
R	funct7	rs2	rs1	funct3	rd	opcode	ZNAČENJE
add rd, rs1, rs2	0000000	*rs2*	*rs1*	000	*rd*	0110011	rd = rs1 + rs2
sub rd, rs1, rs2	0100000	*rs2*	*rs1*	000	*rd*	0110011	rd = rs1 − rs2
sll rd, rs1, rs2	0000000	*rs2*	*rs1*	001	*rd*	0110011	rd = rs1 << rs2[4:0]
slt rd, rs1, rs2	0000000	*rs2*	*rs1*	010	*rd*	0110011	rd = (rs1 < rs2) ? 1 : 0 (označeno)
sltu rd, rs1, rs2	0000000	*rs2*	*rs1*	011	*rd*	0110011	rd = (rs1 < rs2) ? 1 : 0 (neoznačeno)
xor rd, rs1, rs2	0000000	*rs2*	*rs1*	100	*rd*	0110011	rd = rs1 ^ rs2
srl rd, rs1, rs2	0000000	*rs2*	*rs1*	101	*rd*	0110011	rd = rs1 >> rs2[4:0] (logičko)
sra rd, rs1, rs2	0100000	*rs2*	*rs1*	101	*rd*	0110011	rd = rs1 >> rs2[4:0] (aritmetičko)
or rd, rs1, rs2	0000000	*rs2*	*rs1*	110	*rd*	0110011	rd = rs1 | rs2
and rd, rs1, rs2	0000000	*rs2*	*rs1*	111	*rd*	0110011	rd = rs1 & rs2
mul rd, rs1, rs2	0000001	*rs2*	*rs1*	000	*rd*	0110011	rd = (rs1 × rs2)[31:0]
mulh rd, rs1, rs2	0000001	*rs2*	*rs1*	001	*rd*	0110011	rd = (rs1 × rs2)[63:32] (označeno)
mulhsu rd, rs1, rs2	0000001	*rs2*	*rs1*	010	*rd*	0110011	rd = (rs1 × rs2)[63:32] (rs1 označeno, rs2 neoznačeno)
mulhu rd, rs1, rs2	0000001	*rs2*	*rs1*	011	*rd*	0110011	rd = (rs1 × rs2)[63:32] (neoznačeno)
div rd, rs1, rs2	0000001	*rs2*	*rs1*	100	*rd*	0110011	rd = rs1 / rs2 (označeno)
divu rd, rs1, rs2	0000001	*rs2*	*rs1*	101	*rd*	0110011	rd = rs1 / rs2 (neoznačeno)
rem rd, rs1, rs2	0000001	*rs2*	*rs1*	110	*rd*	0110011	rd = rs1 % rs2 (označeno)
remu rd, rs1, rs2	0000001	*rs2*	*rs1*	111	*rd*	0110011	rd = rs1 % rs2 (neoznačeno)

mul ... remu (funct7 = 0000001) su M ekstenzija.

I tip: registar, konstanta

	31..25	24..20	19..15	14..12	11..7	6..0	
I	imm[11:0]		rs1	funct3	rd	opcode	ZNAČENJE
addi rd, rs1, imm	*imm[11:0]*		*rs1*	000	*rd*	0010011	rd = rs1 + imm
slti rd, rs1, imm	*imm[11:0]*		*rs1*	010	*rd*	0010011	rd = (rs1 < imm) ? 1 : 0 (označeno)
sltiu rd, rs1, imm	*imm[11:0]*		*rs1*	011	*rd*	0010011	rd = (rs1 < imm) ? 1 : 0 (neoznačeno)
xori rd, rs1, imm	*imm[11:0]*		*rs1*	100	*rd*	0010011	rd = rs1 ^ imm
ori rd, rs1, imm	*imm[11:0]*		*rs1*	110	*rd*	0010011	rd = rs1 | imm
andi rd, rs1, imm	*imm[11:0]*		*rs1*	111	*rd*	0010011	rd = rs1 & imm
slli rd, rs1, shamt	0000000	*shamt*	*rs1*	001	*rd*	0010011	rd = rs1 << shamt
srli rd, rs1, shamt	0000000	*shamt*	*rs1*	101	*rd*	0010011	rd = rs1 >> shamt (logičko)
srai rd, rs1, shamt	0100000	*shamt*	*rs1*	101	*rd*	0010011	rd = rs1 >> shamt (aritmetičko)

I tip: čitanje iz memorije i jalr

	31..25	24..20	19..15	14..12	11..7	6..0	
I	imm[11:0]		rs1	funct3	rd	opcode	ZNAČENJE
lb rd, imm(rs1)	*imm[11:0]*		*rs1*	000	*rd*	0000011	rd = sext(mem8[rs1 + imm])
lh rd, imm(rs1)	*imm[11:0]*		*rs1*	001	*rd*	0000011	rd = sext(mem16[rs1 + imm])
lw rd, imm(rs1)	*imm[11:0]*		*rs1*	010	*rd*	0000011	rd = mem32[rs1 + imm]
lbu rd, imm(rs1)	*imm[11:0]*		*rs1*	100	*rd*	0000011	rd = zext(mem8[rs1 + imm])
lhu rd, imm(rs1)	*imm[11:0]*		*rs1*	101	*rd*	0000011	rd = zext(mem16[rs1 + imm])
jalr rd, imm(rs1)	*imm[11:0]*		*rs1*	000	*rd*	1100111	rd = pc + 4; pc = (rs1 + imm) & ~1

RISC-V RV32IM binarni zapis instrukcija

Ime i prezime studenta: _____ Potpis studenta: _____

Potpis dežurnog: _____ Datum ispita: _____

I tip: sistem i CSR

	31..25	24..20	19..15	14..12	11..7	6..0	
I	imm[11:0]		rs1	funct3	rd	opcode	ZNAČENJE
ecall	000000000000		00000	000	00000	1110011	poziv okruženja
ebreak	000000000001		00000	000	00000	1110011	prekidna tačka
mret	001100000010		00000	000	00000	1110011	pc = mepc
fence	*fm|pred|succ*		00000	000	00000	0001111	redosled pristupa
csrrw rd, csr, rs1	*csr*		*rs1*	001	*rd*	1110011	rd = csr; csr = rs1
csrrs rd, csr, rs1	*csr*		*rs1*	010	*rd*	1110011	rd = csr; csr |= rs1
csrrc rd, csr, rs1	*csr*		*rs1*	011	*rd*	1110011	rd = csr; csr &= ~rs1
csrrwi rd, csr, uimm	*csr*		*uimm*	101	*rd*	1110011	rd = csr; csr = uimm
csrrsi rd, csr, uimm	*csr*		*uimm*	110	*rd*	1110011	rd = csr; csr |= uimm
csrrci rd, csr, uimm	*csr*		*uimm*	111	*rd*	1110011	rd = csr; csr &= ~uimm

S tip: upis u memoriju

	31..25	24..20	19..15	14..12	11..7	6..0	
S	imm[11:5]	rs2	rs1	funct3	imm[4:0]	opcode	ZNAČENJE
sb rs2, imm(rs1)	*imm[11:5]*	*rs2*	*rs1*	000	*imm[4:0]*	0100011	mem8[rs1 + imm] = rs2[7:0]
sh rs2, imm(rs1)	*imm[11:5]*	*rs2*	*rs1*	001	*imm[4:0]*	0100011	mem16[rs1 + imm] = rs2[15:0]
sw rs2, imm(rs1)	*imm[11:5]*	*rs2*	*rs1*	010	*imm[4:0]*	0100011	mem32[rs1 + imm] = rs2

Konstanta (imm) po formatu

TIP	IMM	NAPOMENA
I	sext(IR[31:20])	12 bita, ±2 KiB
S	sext(IR[31:25] : IR[11:7])	kao I, ali rd mesto nosi imm[4:0]
B	sext(IR[31] : IR[7] : IR[30:25] : IR[11:8] : 0)	pomeraj u bajtovima, uvek paran
U	IR[31:12] : 000000000000	gornjih 20 bita
J	sext(IR[31] : IR[19:12] : IR[20] : IR[30:21] : 0)	±1 MiB, uvek paran

„:“ spaja bitove (levo su viši); sext = proširenje znakom, bit 31 instrukcije je uvek znak konstante.

B tip: uslovni skok

	31..25	24..20	19..15	14..12	11..7	6..0	
B	imm[12|10:5]	rs2	rs1	funct3	imm[4:1|11]	opcode	ZNAČENJE
beq rs1, rs2, imm	*imm[12|10:5]*	*rs2*	*rs1*	000	*imm[4:1|11*	1100011	if (rs1 == rs2) pc = pc + imm
bne rs1, rs2, imm	*imm[12|10:5]*	*rs2*	*rs1*	001	*imm[4:1|11*	1100011	if (rs1 != rs2) pc = pc + imm
blt rs1, rs2, imm	*imm[12|10:5]*	*rs2*	*rs1*	100	*imm[4:1|11*	1100011	if (rs1 < rs2) pc = pc + imm (označeno)
bge rs1, rs2, imm	*imm[12|10:5]*	*rs2*	*rs1*	101	*imm[4:1|11*	1100011	if (rs1 >= rs2) pc = pc + imm (označeno)
bltu rs1, rs2, imm	*imm[12|10:5]*	*rs2*	*rs1*	110	*imm[4:1|11*	1100011	if (rs1 < rs2) pc = pc + imm (neoznačeno)
bgeu rs1, rs2, imm	*imm[12|10:5]*	*rs2*	*rs1*	111	*imm[4:1|11*	1100011	if (rs1 >= rs2) pc = pc + imm (neoznačeno)

Važi i za jal. pc je adresa instrukcije skoka; u asembleru se umesto imm piše labela ili adresa, a asembler izračuna imm = adresa − pc.

U tip: gornjih 20 bita

	31..25	24..20	19..15	14..12	11..7	6..0	
U	imm[31:12]				rd	opcode	ZNAČENJE
lui rd, imm	*imm[31:12]*				*rd*	0110111	rd = imm << 12
auipc rd, imm	*imm[31:12]*				*rd*	0010111	rd = pc + (imm << 12)

J tip: skok

	31..25	24..20	19..15	14..12	11..7	6..0	
J	imm[20|10:1|11|19:12]				rd	opcode	ZNAČENJE
jal rd, imm	*imm[20|10:1|11|19:12]*				*rd*	1101111	rd = pc + 4; pc = pc + imm

Opcode (IR[6:0])

OPCODE	GRUPA	TIP	OPCODE	GRUPA	TIP
0110011	OP	R	1101111	JAL	J
0010011	OP-IMM	I	0110111	LUI	U
0000011	LOAD	I	0010111	AUIPC	U
0100011	STORE	S	1110011	SYSTEM	I
1100011	BRANCH	B	0001111	MISC-MEM	I
1100111	JALR	I			

Najniža dva bita su uvek 11 (instrukcije od 32 bita). Opcode bira format, funct3 i funct7 biraju operaciju.