31	25	24	20	19	15	14	12	11	7	6	0	
imm[31:12]								rd		opcode		**Type-U**
imm[20:10:1|11|19:12]								rd		opcode		**Type-UJ**
imm[11:0]				rs1		funct3		rd		opcode		**Type-I**
imm[12|10:5]		rs2		rs1		funct3		imm[4:1|11]		opcode		**Type-SB**
imm[11:5]		rs2		rs1		funct3		imm[4:0]		opcode		**Type-S**
funct5	funct2	rs2		rs1		funct3		rd		opcode		**Type-R**

RV32I Base Integer Instruction Set

31	25	24	20	19	15	14	12	11	7	6	0	
imm[31:12]								rd		0110111		LUI rd, imm
imm[31:12]								rd		0010111		AUIPC rd, offset
imm[20:10:1|11|19:12]								rd		1101111		JAL rd, offset
imm[11:0]				rs1		000		rd		1100111		JALR rd, rs1, offset
imm[12|10:5]		rs2		rs1		000		imm[4:1|11]		1100011		BEQ rs1, rs2, offset
imm[12|10:5]		rs2		rs1		001		imm[4:1|11]		1100011		BNE rs1, rs2, offset
imm[12|10:5]		rs2		rs1		100		imm[4:1|11]		1100011		BLT rs1, rs2, offset
imm[12|10:5]		rs2		rs1		101		imm[4:1|11]		1100011		BGE rs1, rs2, offset
imm[12|10:5]		rs2		rs1		110		imm[4:1|11]		1100011		BLTU rs1, rs2, offset
imm[12|10:5]		rs2		rs1		111		imm[4:1|11]		1100011		BGEU rs1, rs2, offset
imm[11:0]				rs1		000		rd		0000011		LB rd, offset(rs1)
imm[11:0]				rs1		001		rd		0000011		LH rd, offset(rs1)
imm[11:0]				rs1		010		rd		0000011		LW rd, offset(rs1)
imm[11:0]				rs1		100		rd		0000011		LBU rd, offset(rs1)
imm[11:0]				rs1		101		rd		0000011		LHU rd, offset(rs1)
imm[11:5]		rs2		rs1		000		imm[4:0]		0100011		SB rs2, offset(rs1)
imm[11:5]		rs2		rs1		001		imm[4:0]		0100011		SH rs2, offset(rs1)
imm[11:5]		rs2		rs1		010		imm[4:0]		0100011		SW rs2, offset(rs1)
imm[11:0]				rs1		000		rd		0010011		ADDI rd, rs1, imm
imm[11:0]				rs1		010		rd		0010011		SLTI rd, rs1, imm
imm[11:0]				rs1		011		rd		0010011		SLTIU rd, rs1, imm
imm[11:0]				rs1		100		rd		0010011		XORI rd, rs1, imm
imm[11:0]				rs1		110		rd		0010011		ORI rd, rs1, imm
imm[11:0]				rs1		111		rd		0010011		ANDI rd, rs1, imm
000000	0	shamt[4:0]		rs1		001		rd		0010011		SLLI rd, rs1, imm
000000	0	shamt[4:0]		rs1		101		rd		0010011		SRLI rd, rs1, imm
010000	0	shamt[4:0]		rs1		101		rd		0010011		SRAI rd, rs1, imm
00000	00	rs2		rs1		000		rd		0110011		ADD rd, rs1, rs2
01000	00	rs2		rs1		000		rd		0110011		SUB rd, rs1, rs2
00000	00	rs2		rs1		001		rd		0110011		SLL rd, rs1, rs2
00000	00	rs2		rs1		010		rd		0110011		SLT rd, rs1, rs2
00000	00	rs2		rs1		011		rd		0110011		SLTU rd, rs1, rs2
00000	00	rs2		rs1		100		rd		0110011		XOR rd, rs1, rs2
00000	00	rs2		rs1		101		rd		0110011		SRL rd, rs1, rs2
01000	00	rs2		rs1		101		rd		0110011		SRA rd, rs1, rs2
00000	00	rs2		rs1		110		rd		0110011		OR rd, rs1, rs2
00000	00	rs2		rs1		111		rd		0110011		AND rd, rs1, rs2
0000	pred	pred	succ	00000		000		00000		0001111		FENCE
0000000		00000		00000		001		00000		0001111		FENCE.I

RV64I Base Integer Instruction Set (in addition to RV32I)

31	25	24	20	19	15	14	12	11	7	6	0	
imm[11:0]				rs1		110		rd		0000011		LWU rd, offset(rs1)
imm[11:0]				rs1		011		rd		0000011		LD rd, offset(rs1)
imm[11:5]		rs2		rs1		011		imm[4:0]		0100011		SD rs2, offset(rs1)
000000		shamt[5:0]		rs1		001		rd		0010011		SLLI rd, rs1, imm
000000		shamt[5:0]		rs1		101		rd		0010011		SRLI rd, rs1, imm
010000		shamt[5:0]		rs1		101		rd		0010011		SRAI rd, rs1, imm
imm[11:0]				rs1		000		rd		0011011		ADDIW rd, rs1, imm
0000000		shamt[4:0]		rs1		001		rd		0011011		SLLIW rd, rs1, imm
0000000		shamt[4:0]		rs1		101		rd		0011011		SRLIW rd, rs1, imm
0100000		shamt[4:0]		rs1		101		rd		0011011		SRAIW rd, rs1, imm

31		25 24	20 19	15 14	12 11	7 6	0	
funct5	funct2	rs2	rs1	funct3	rd	opcode		**Type-R**

RV64I Base Integer Instruction Set (in addition to RV32I) contd

funct5	funct2	rs2	rs1	funct3	rd	opcode	
00000	00	rs2	rs1	000	rd	0111011	ADDW rd, rs1, rs2
01000	00	rs2	rs1	000	rd	0111011	SUBW rd, rs1, rs2
00000	00	rs2	rs1	001	rd	0111011	SLLW rd, rs1, rs2
00000	00	rs2	rs1	101	rd	0111011	SRLW rd, rs1, rs2
01000	00	rs2	rs1	101	rd	0111011	SRAW rd, rs1, rs2

RV32M Standard Extension for Integer Multiply and Divide

funct5	funct2	rs2	rs1	funct3	rd	opcode	
00000	01	rs2	rs1	000	rd	0110011	MUL rd, rs1, rs2
00000	01	rs2	rs1	001	rd	0110011	MULH rd, rs1, rs2
00000	01	rs2	rs1	010	rd	0110011	MULHSU rd, rs1, rs2
00000	01	rs2	rs1	011	rd	0110011	MULHU rd, rs1, rs2
00000	01	rs2	rs1	100	rd	0110011	DIV rd, rs1, rs2
00000	01	rs2	rs1	101	rd	0110011	DIVU rd, rs1, rs2
00000	01	rs2	rs1	110	rd	0110011	REM rd, rs1, rs2
00000	01	rs2	rs1	111	rd	0110011	REMU rd, rs1, rs2

RV64M Standard Extension for Integer Multiply and Divide (in addition to RV32M)

funct5	funct2	rs2	rs1	funct3	rd	opcode	
00000	01	rs2	rs1	000	rd	0111011	MULW rd, rs1, rs2
00000	01	rs2	rs1	100	rd	0111011	DIVW rd, rs1, rs2
00000	01	rs2	rs1	101	rd	0111011	DIVUW rd, rs1, rs2
00000	01	rs2	rs1	110	rd	0111011	REMW rd, rs1, rs2
00000	01	rs2	rs1	111	rd	0111011	REMUW rd, rs1, rs2

RV32A Standard Extension for Atomic Instructions

funct5	aq	rl	rs2	rs1	funct3	rd	opcode	
00010	aq	rl	00000	rs1	010	rd	0101111	LR.W aqrl, rd, (rs1)
00011	aq	rl	rs2	rs1	010	rd	0101111	SC.W aqrl, rd, rs2, (rs1)
00001	aq	rl	rs2	rs1	010	rd	0101111	AMOSWAP.W aqrl, rd, rs2, (rs1)
00000	aq	rl	rs2	rs1	010	rd	0101111	AMOADD.W aqrl, rd, rs2, (rs1)
00100	aq	rl	rs2	rs1	010	rd	0101111	AMOXOR.W aqrl, rd, rs2, (rs1)
01000	aq	rl	rs2	rs1	010	rd	0101111	AMOOR.W aqrl, rd, rs2, (rs1)
01100	aq	rl	rs2	rs1	010	rd	0101111	AMOAND.W aqrl, rd, rs2, (rs1)
10000	aq	rl	rs2	rs1	010	rd	0101111	AMOMIN.W aqrl, rd, rs2, (rs1)
10100	aq	rl	rs2	rs1	010	rd	0101111	AMOMAX.W aqrl, rd, rs2, (rs1)
11000	aq	rl	rs2	rs1	010	rd	0101111	AMOMINU.W aqrl, rd, rs2, (rs1)
11100	aq	rl	rs2	rs1	010	rd	0101111	AMOMAXU.W aqrl, rd, rs2, (rs1)

RV64A Standard Extension for Atomic Instructions (in addition to RV32A)

funct5	aq	rl	rs2	rs1	funct3	rd	opcode	
00010	aq	rl	00000	rs1	011	rd	0101111	LR.D aqrl, rd, (rs1)
00011	aq	rl	rs2	rs1	011	rd	0101111	SC.D aqrl, rd, rs2, (rs1)
00001	aq	rl	rs2	rs1	011	rd	0101111	AMOSWAP.D aqrl, rd, rs2, (rs1)
00000	aq	rl	rs2	rs1	011	rd	0101111	AMOADD.D aqrl, rd, rs2, (rs1)
00100	aq	rl	rs2	rs1	011	rd	0101111	AMOXOR.D aqrl, rd, rs2, (rs1)
01000	aq	rl	rs2	rs1	011	rd	0101111	AMOOR.D aqrl, rd, rs2, (rs1)
01100	aq	rl	rs2	rs1	011	rd	0101111	AMOAND.D aqrl, rd, rs2, (rs1)
10000	aq	rl	rs2	rs1	011	rd	0101111	AMOMIN.D aqrl, rd, rs2, (rs1)
10100	aq	rl	rs2	rs1	011	rd	0101111	AMOMAX.D aqrl, rd, rs2, (rs1)
11000	aq	rl	rs2	rs1	011	rd	0101111	AMOMINU.D aqrl, rd, rs2, (rs1)
11100	aq	rl	rs2	rs1	011	rd	0101111	AMOMAXU.D aqrl, rd, rs2, (rs1)

31　　　　　25	24　　　　20	19　　　15	14　　12	11　　　　7	6　　　　　　0	
imm[11:0]		rs1	funct3	rd	opcode	**Type-I**
imm[11:5]	rs2	rs1	funct3	imm[4:0]	opcode	**Type-S**
rs3	funct2	rs2	rs1	funct3	rd	opcode	**Type-R4**
funct5	funct2	rs2	rs1	funct3	rd	opcode	**Type-R**

RV32S Standard Extension for Supervisor-level Instructions

0000000	00000	00000	000	00000	1110011	ECALL
0000000	00001	00000	000	00000	1110011	EBREAK
0000000	00010	00000	000	00000	1110011	URET
0001000	00000	00000	000	00000	1110011	SRET
0010000	00010	00000	000	00000	1110011	HRET
0011000	00010	00000	000	00000	1110011	MRET
0111101	10010	00000	000	00000	1110011	DRET
0001000	00001	rs1	000	00000	1110011	SFENCE.VM
0001000	00010	00000	000	00000	1110011	WFI
csr[11:0]		rs1	001	rd	1110011	CSRRW rd, csr, rs1
csr[11:0]		rs1	010	rd	1110011	CSRRS rd, csr, rs1
csr[11:0]		rs1	011	rd	1110011	CSRRC rd, csr, rs1
csr[11:0]		zimm[4:0]	101	rd	1110011	CSRRWI rd, csr, zimm
csr[11:0]		zimm[4:0]	110	rd	1110011	CSRRSI rd, csr, zimm
csr[11:0]		zimm[4:0]	111	rd	1110011	CSRRCI rd, csr, zimm

RV32F Standard Extension for Single-Precision Floating-Point

imm[11:0]		rs1	010	frd	0000111	FLW frd, offset(rs1)
imm[11:5]	frs2	rs1	010	imm[4:0]	0100111	FSW frs2, offset(rs1)
frs3	00	frs2	frs1	rm	frd	1000011	FMADD.S rm, frd, frs1, frs2, frs3
frs3	00	frs2	frs1	rm	frd	1000111	FMSUB.S rm, frd, frs1, frs2, frs3
frs3	00	frs2	frs1	rm	frd	1001011	FNMSUB.S rm, frd, frs1, frs2, frs3
frs3	00	frs2	frs1	rm	frd	1001111	FNMADD.S rm, frd, frs1, frs2, frs3
00000	00	frs2	frs1	rm	frd	1010011	FADD.S rm, frd, frs1, frs2
00001	00	frs2	frs1	rm	frd	1010011	FSUB.S rm, frd, frs1, frs2
00010	00	frs2	frs1	rm	frd	1010011	FMUL.S rm, frd, frs1, frs2
00011	00	frs2	frs1	rm	frd	1010011	FDIV.S rm, frd, frs1, frs2
00100	00	frs2	frs1	000	frd	1010011	FSGNJ.S frd, frs1, frs2
00100	00	frs2	frs1	001	frd	1010011	FSGNJN.S frd, frs1, frs2
00100	00	frs2	frs1	010	frd	1010011	FSGNJX.S frd, frs1, frs2
00101	00	frs2	frs1	000	frd	1010011	FMIN.S frd, frs1, frs2
00101	00	frs2	frs1	001	frd	1010011	FMAX.S frd, frs1, frs2
01011	00	00000	frs1	rm	frd	1010011	FSQRT.S rm, frd, frs1
10100	00	frs2	frs1	000	rd	1010011	FLE.S rd, frs1, frs2
10100	00	frs2	frs1	001	rd	1010011	FLT.S rd, frs1, frs2
10100	00	frs2	frs1	010	rd	1010011	FEQ.S rd, frs1, frs2
11000	00	00000	frs1	rm	rd	1010011	FCVT.W.S rm, rd, frs1
11000	00	00001	frs1	rm	rd	1010011	FCVT.WU.S rm, rd, frs1
11010	00	00000	rs1	rm	frd	1010011	FCVT.S.W rm, frd, rs1
11010	00	00001	rs1	rm	frd	1010011	FCVT.S.WU rm, frd, rs1
11100	00	00000	frs1	000	rd	1010011	FMV.X.S rd, frs1
11100	00	00000	frs1	001	rd	1010011	FCLASS.S rd, frs1
11110	00	00000	rs1	000	frd	1010011	FMV.S.X frd, rs1

RV64F Standard Extension for Single-Precision Floating-Point (in addition to RV32F)

11000	00	00010	frs1	rm	rd	1010011	FCVT.L.S rm, rd, frs1
11000	00	00011	frs1	rm	rd	1010011	FCVT.LU.S rm, rd, frs1
11010	00	00010	rs1	rm	frd	1010011	FCVT.S.L rm, frd, rs1
11010	00	00011	rs1	rm	frd	1010011	FCVT.S.LU rm, frd, rs1

31	25	24	20	19	15	14	12	11	7	6	0	
imm[11:0]				rs1		funct3		rd		opcode		**Type-I**
imm[11:5]		rs2		rs1		funct3		imm[4:0]		opcode		**Type-S**
rs3	funct2	rs2		rs1		funct3		rd		opcode		**Type-R4**
funct5	funct2	rs2		rs1		funct3		rd		opcode		**Type-R**

RV32D Standard Extension for Double-Precision Floating-Point

imm[11:0]				rs1		011		frd		0000111		FLD frd, offset(rs1)
imm[11:5]		frs2		rs1		011		imm[4:0]		0100111		FSD frs2, offset(rs1)
frs3	01	frs2		frs1		rm		frd		1000011		FMADD.D rm, frd, frs1, frs2, frs3
frs3	01	frs2		frs1		rm		frd		1000111		FMSUB.D rm, frd, frs1, frs2, frs3
frs3	01	frs2		frs1		rm		frd		1001011		FNMSUB.D rm, frd, frs1, frs2, frs3
frs3	01	frs2		frs1		rm		frd		1001111		FNMADD.D rm, frd, frs1, frs2, frs3
00000	01	frs2		frs1		rm		frd		1010011		FADD.D rm, frd, frs1, frs2
00001	01	frs2		frs1		rm		frd		1010011		FSUB.D rm, frd, frs1, frs2
00010	01	frs2		frs1		rm		frd		1010011		FMUL.D rm, frd, frs1, frs2
00011	01	frs2		frs1		rm		frd		1010011		FDIV.D rm, frd, frs1, frs2
00100	01	frs2		frs1		000		frd		1010011		FSGNJ.D frd, frs1, frs2
00100	01	frs2		frs1		001		frd		1010011		FSGNJN.D frd, frs1, frs2
00100	01	frs2		frs1		010		frd		1010011		FSGNJX.D frd, frs1, frs2
00101	01	frs2		frs1		000		frd		1010011		FMIN.D frd, frs1, frs2
00101	01	frs2		frs1		001		frd		1010011		FMAX.D frd, frs1, frs2
01000	00	00001		frs1		rm		frd		1010011		FCVT.S.D rm, frd, frs1
01000	01	00000		frs1		rm		frd		1010011		FCVT.D.S rm, frd, frs1
01011	01	00000		frs1		rm		frd		1010011		FSQRT.D rm, frd, frs1
10100	01	frs2		frs1		000		rd		1010011		FLE.D rd, frs1, frs2
10100	01	frs2		frs1		001		rd		1010011		FLT.D rd, frs1, frs2
10100	01	frs2		frs1		010		rd		1010011		FEQ.D rd, frs1, frs2
11000	01	00000		frs1		rm		rd		1010011		FCVT.W.D rm, rd, frs1
11000	01	00001		frs1		rm		rd		1010011		FCVT.WU.D rm, rd, frs1
11010	01	00000		rs1		rm		frd		1010011		FCVT.D.W rm, frd, rs1
11010	01	00001		rs1		rm		frd		1010011		FCVT.D.WU rm, frd, rs1
11100	01	00000		frs1		001		rd		1010011		FCLASS.D rd, frs1

RV64D Standard Extension for Double-Precision Floating-Point (in addition to RV32D)

11000	01	00010		frs1		rm		rd		1010011		FCVT.L.D rm, rd, frs1
11000	01	00011		frs1		rm		rd		1010011		FCVT.LU.D rm, rd, frs1
11100	01	00000		frs1		000		rd		1010011		FMV.X.D rd, frs1
11010	01	00010		rs1		rm		frd		1010011		FCVT.D.L rm, frd, rs1
11010	01	00011		rs1		rm		frd		1010011		FCVT.D.LU rm, frd, rs1
11110	01	00000		rs1		000		frd		1010011		FMV.D.X frd, rs1

15	13	12	10	9	7	6	5	4	2	1	0	
funct3		imm8						rd'		op		**Type-CIW**
funct3		imm3		rs1'		imm2		rd'		op		**Type-CL**
funct3		imm3		rs1'		imm2		rs2'		op		**Type-CS**
funct3		imm1		rd/rs1			imm5			op		**Type-CI**
funct3				imm11						op		**Type-CJ**
funct3		imm3		rs1'			imm5			op		**Type-CB**
funct4				rd/rs1			rs2			op		**Type-CR**
funct3		imm6					rs2			op		**Type-CSS**

RV32C Standard Extension for Compressed Instructions

000		nzimm[5:4|9:6|2|3]				rd'	00	C.ADDI4SPN rd, rs1, imm
001	imm[5:3]		rs1'		imm[7:6]	frd'	00	C.FLD frd, offset(rs1)
010	imm[5:3]		rs1'		imm[2|6]	rd'	00	C.LW rd, offset(rs1)
011	imm[5:3]		rs1'		imm[2|6]	frd'	00	C.FLW frd, offset(rs1)
101	imm[5:3]		rs1'		imm[7:6]	frs2'	00	C.FSD frs2, offset(rs1)
110	imm[5:3]		rs1'		imm[2|6]	rs2'	00	C.SW rs2, offset(rs1)
111	imm[5:3]		rs1'		imm[2|6]	frs2'	00	C.FSW frs2, offset(rs1)
000	0		00000			00000	01	C.NOP
000	nzimm[5]		rs1/rd≠ 0			nzimm[4:0]	01	C.ADDI rd, rs1, imm
001			imm[11|4|9:8|10|6|7|3:1|5]				01	C.JAL rd, offset
010	imm[5]		rs1/rd≠ 0			imm[4:0]	01	C.LI rd, rs1, imm
011	nzimm[9]		rs1/rd= 2			nzimm[4|6|8:7|5]	01	C.ADDI16SP rd, rs1, imm
011	nzimm[17]		rd≠ {0, 2}			nzimm[16:12]	01	C.LUI rd, imm
100	000		rs1'/rd'			nzimm[4:0]	01	C.SRLI rd, rs1, imm
100	001		rs1'/rd'			nzimm[4:0]	01	C.SRAI rd, rs1, imm
100	nzimm[5]	10	rs1'/rd'			nzimm[4:0]	01	C.ANDI rd, rs1, imm
100	011		rs1'/rd'		00	rs2'	01	C.SUB rd, rs1, rs2
100	011		rs1'/rd'		01	rs2'	01	C.XOR rd, rs1, rs2
100	011		rs1'/rd'		10	rs2'	01	C.OR rd, rs1, rs2
100	011		rs1'/rd'		11	rs2'	01	C.AND rd, rs1, rs2
100	111		rs1'/rd'		00	rs2'	01	C.SUBW rd, rs1, rs2
100	111		rs1'/rd'		01	rs2'	01	C.ADDW rd, rs1, rs2
101			imm[11|4|9:8|10|6|7|3:1|5]				01	C.J rd, offset
110	imm[8|4:3]		rs1'			imm[7:6|2:1|5]	01	C.BEQZ rs1, rs2, offset
111	imm[8|4:3]		rs1'			imm[7:6|2:1|5]	01	C.BNEZ rs1, rs2, offset
000	0		rs1/rd≠ 0			nzimm[4:0]	10	C.SLLI rd, rs1, imm
001	imm[5]		frd			imm[4:3|8:6]	10	C.FLDSP frd, offset(rs1)
010	imm[5]		rd≠ 0			imm[4:2|7:6]	10	C.LWSP rd, offset(rs1)
011	imm[5]		frd			imm[4:2|7:6]	10	C.FLWSP frd, offset(rs1)
100	rd,		rs1			00000	10	C.JR rd, rs1, offset
1000			rd≠ 0			rs2≠ 0	10	C.MV rd, rs1, rs2
100	1		00000			00000	10	C.EBREAK
100	rd,		rs1			00000	10	C.JALR rd, rs1, offset
1001			rs1/rd≠ 0			rs2≠ 0	10	C.ADD rd, rs1, rs2
101	imm[5:3|8:6]					frs2	10	C.FSDSP frs2, offset(rs1)
110	imm[5:2|7:6]					rs2	10	C.SWSP rs2, offset(rs1)
111	imm[5:2|7:6]					frs2	10	C.FSWSP frs2, offset(rs1)

RV64C Standard Extension for Compressed Instructions (in addition to RV32C)

011	imm[5:3]		rs1'		imm[7:6]	rd'	00	C.LD rd, offset(rs1)
111	imm[5:3]		rs1'		imm[7:6]	rs2'	00	C.SD rs2, offset(rs1)
001	imm[5]		rs1/rd≠ 0			imm[4:0]	01	C.ADDIW rd, rs1, imm
100	nzimm[5]	00	rs1'/rd'			nzimm[4:0]	01	C.SRLI rd, rs1, imm
100	nzimm[5]	01	rs1'/rd'			nzimm[4:0]	01	C.SRAI rd, rs1, imm
000	nzimm[5]		rs1/rd≠ 0			nzimm[4:0]	10	C.SLLI rd, rs1, imm
011	imm[5]		rd≠ 0			imm[4:3|8:6]	10	C.LDSP rd, offset(rs1)
111	imm[5:3|8:6]					rs2	10	C.SDSP rs2, offset(rs1)