

## Trabalho Pratico #2

*Professor:* Omar Paranaíba Vilela Neto e Daniel Fernandes Macedo

*Monitor:* Matheus H. Silva

Antes de começar seu trabalho, leia todas as instruções abaixo.

- O trabalho pode ser feito em grupos compostos por **até 3 alunos**.
- Cópias de trabalho acarretarão em devida penalização às partes envolvidas.
- Entregas após o prazo serão aceitas, porem haverá uma penalização. Quanto maior o atraso maior a penalização.
- O objetivo desse trabalho é te familiarizar com a Linguagem de Descrição de *Hardware Verilog*. Será disponibilizado no moodle um arquivo .ipynb com uma implementação do RISC-V 5 estágios em Verilog. Sua tarefa nesse trabalho será alterar o caminho de dados fornecido a fim de incluir mais operações e módulos. É **necessário executar esse arquivo no google Colab**, sendo está a plataforma que será utilizada para avaliar as submissões dos trabalhos.
- Você deve entregar um único arquivo zip, contendo um arquivo .ipynb com a implementação do caminho de dados com as funções pedidas a seguir, em Verilog. Note que todas as funções devem estar no mesmo caminho de dados, ou seja, o trabalho é incremental, você deve entregar **somente um caminho de dados contendo todas as funções solicitadas**.
- Deverão ser implementados os arquivos Verilog e os códigos de teste em assembly das instruções. **As suas modificações devem ser feitas em uma cópia** do código Verilog fornecido. É recomendado que você mostre as formas de onda, assim como mostradas nos exemplos do arquivo .ipynb fornecido.
- No mesmo arquivo zip contendo o caminho de dados, você deve **enviar um relatório**, em pdf, explicando suas decisões de projeto e contendo **nome e matrícula de todos os integrantes do grupo**.
- Cada grupo deve fazer somente uma submissão.

### Problema 1: ANDI - Bitwise or immediate

(5.0 pontos)

#Para mais informações sobre o funcionamento dessa instrução confira a documentação do RISC-V

<https://github.com/riscv/riscv-isa-manual/releases/latest>

### Problema 2: SRLI - Shift Right Logical Immediate

(2.5 pontos)

#Para mais informações sobre o funcionamento dessa instrução confira a documentação do RISC-V

<https://github.com/riscv/riscv-isa-manual/releases/latest>

### Problema 3: J – Jump

(5.0 pontos)

#Para mais informações sobre o funcionamento dessa instrução confira a documentação do RISC-V

<https://github.com/riscv/riscv-isa-manual/releases/latest>

### Problema 4: BGT– Branch on Greater Than

(2.5 pontos)

#Para mais informações sobre o funcionamento dessa instrução confira a documentação do RISC-V

<https://github.com/riscv/riscv-isa-manual/releases/latest>

31	25	24	20	19	15	14	12	11	7	6	0	
imm[31:12]								rd	opcode			Type-U
imm[20:10:11:19:12]								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

simm[31:12]					rd	0110111	LUI rd, imm		
simm[31:12]					rd	0010111	AUIPC rd, offset		
simm[20 10:1 11 19:12]					rd	1101111	JAL rd, offset		
simm[11:0]			rs1	000	rd	1100111	JALR rd, rs1, offset		
simm[12 10:5]		rs2	rs1	000	simm[4:1 11]	1100011	BEQ rs1, rs2, offset		
simm[12 10:5]		rs2	rs1	001	simm[4:1 11]	1100011	BNE rs1, rs2, offset		
simm[12 10:5]		rs2	rs1	100	simm[4:1 11]	1100011	BLT rs1, rs2, offset		
simm[12 10:5]		rs2	rs1	101	simm[4:1 11]	1100011	BGE rs1, rs2, offset		
simm[12 10:5]		rs2	rs1	110	simm[4:1 11]	1100011	BLTU rs1, rs2, offset		
simm[12 10:5]		rs2	rs1	111	simm[4:1 11]	1100011	BGEU rs1, rs2, offset		
simm[11:0]			rs1	000	rd	0000011	LB rd, offset(rs1)		
simm[11:0]			rs1	001	rd	0000011	LH rd, offset(rs1)		
simm[11:0]			rs1	010	rd	0000011	LW rd, offset(rs1)		
simm[11:0]			rs1	100	rd	0000011	LBU rd, offset(rs1)		
simm[11:0]			rs1	101	rd	0000011	LHU rd, offset(rs1)		
simm[11:5]		rs2	rs1	000	simm[4:0]	0100011	SB rs2, offset(rs1)		
simm[11:5]		rs2	rs1	001	simm[4:0]	0100011	SH rs2, offset(rs1)		
simm[11:5]		rs2	rs1	010	simm[4:0]	0100011	SW rs2, offset(rs1)		
simm[11:0]			rs1	000	rd	0010011	ADDI rd, rs1, imm		
simm[11:0]			rs1	010	rd	0010011	SLTI rd, rs1, imm		
simm[11:0]			rs1	011	rd	0010011	SLTIU rd, rs1, imm		
simm[11:0]			rs1	100	rd	0010011	XORI rd, rs1, imm		
simm[11:0]			rs1	110	rd	0010011	ORI rd, rs1, imm		
simm[11:0]			rs1	111	rd	0010011	ANDI rd, rs1, imm		
00000	00	shamt[4:0]	rs1	001	rd	0010011	SLLI rd, rs1, imm		
00000	00	shamt[4:0]	rs1	101	rd	0010011	SRLI rd, rs1, imm		
01000	00	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	pred	succ	00000	000	00000	0001111	FENCE pred, succ
0000000		00000		00000	001	00000	0001111		FENCE.I

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
funct5		funct2	rs2	rs1	funct3		rd		opcode			Type-R

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

simm[11:0]			rs1	110	rd	0000011	LWU rd, offset(rs1)
simm[11:0]			rs1	011	rd	0000011	LD rd, offset(rs1)
simm[11:5]		rs2	rs1	011	simm[4:0]	0100011	SD rs2, offset(rs1)
00000	0	shamt[5:0]	rs1	001	rd	0010011	SLLI rd, rs1, imm
00000	0	shamt[5:0]	rs1	101	rd	0010011	SRLI rd, rs1, imm
01000	0	shamt[5:0]	rs1	101	rd	0010011	SRAI rd, rs1, imm
simm[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
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

#### RV128I Base Integer Instruction Set (in addition to RV64I)

simm[11:0]			rs1	111	rd	0000011	LDU rd, offset(rs1)
simm[11:0]			rs1	010	rd	0001111	LQ rd, offset(rs1)
simm[11:5]		rs2	rs1	100	simm[4:0]	0100011	SQ rs2, offset(rs1)
00000	shamt[6:0]		rs1	001	rd	0010011	SLLI rd, rs1, imm
00000	shamt[6:0]		rs1	101	rd	0010011	SRLI rd, rs1, imm
01000	shamt[6:0]		rs1	101	rd	0010011	SRAI rd, rs1, imm
simm[11:0]			rs1	000	rd	1011011	ADDID rd, rs1, imm
000000		shamt[5:0]	rs1	001	rd	1011011	SLLID rd, rs1, imm
000000		shamt[5:0]	rs1	101	rd	1011011	SRLID rd, rs1, imm
010000		shamt[5:0]	rs1	101	rd	1011011	SRAID rd, rs1, imm
00000	00	rs2	rs1	000	rd	1111011	ADDD rd, rs1, rs2
01000	00	rs2	rs1	000	rd	1111011	SUBD rd, rs1, rs2
00000	00	rs2	rs1	001	rd	1111011	SLLD rd, rs1, rs2
00000	00	rs2	rs1	101	rd	1111011	SRLD rd, rs1, rs2
01000	00	rs2	rs1	101	rd	1111011	SRAD rd, rs1, rs2

#### RV32M Standard Extension for Integer Multiply and Divide

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

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

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

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

#### RV128M Standard Extension for Integer Multiply and Divide (in addition to RV64M)

00000	01	rs2	rs1	000	rd	1111011	MULD rd, rs1, rs2
00000	01	rs2	rs1	100	rd	1111011	DIVD rd, rs1, rs2
00000	01	rs2	rs1	101	rd	1111011	DIVUD rd, rs1, rs2
00000	01	rs2	rs1	110	rd	1111011	REMD rd, rs1, rs2
00000	01	rs2	rs1	111	rd	1111011	REMUW rd, rs1, rs2

#### RV32A Standard Extension for Atomic Instructions

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)

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	
funct5	funct2	rs2	rs1	funct3	rd	opcode						<b>Type-R</b>
	imm[11:0]		rs1	funct3	rd	opcode						<b>Type-I</b>
	imm[11:5]	rs2	rs1	funct3	imm[4:0]	opcode						<b>Type-S</b>
rs3	funct2	rs2	rs1	funct3	rd	opcode						<b>Type-R4</b>

#### RV128A Standard Extension for Atomic Instructions (in addition to RV64A)

00010	aq	rl	00000	rs1	100	rd	0101111	LR.Q aqrl, rd, (rs1)
00011	aq	rl	rs2	rs1	100	rd	0101111	SC.Q aqrl, rd, rs2, (rs1)
00001	aq	rl	rs2	rs1	100	rd	0101111	AMOSWAP.Q aqrl, rd, rs2, (rs1)
00000	aq	rl	rs2	rs1	100	rd	0101111	AMOADD.Q aqrl, rd, rs2, (rs1)
00100	aq	rl	rs2	rs1	100	rd	0101111	AMOXOR.Q aqrl, rd, rs2, (rs1)
01000	aq	rl	rs2	rs1	100	rd	0101111	AMOOR.Q aqrl, rd, rs2, (rs1)
01100	aq	rl	rs2	rs1	100	rd	0101111	AMOAND.Q aqrl, rd, rs2, (rs1)
10000	aq	rl	rs2	rs1	100	rd	0101111	AMOMIN.Q aqrl, rd, rs2, (rs1)
10100	aq	rl	rs2	rs1	100	rd	0101111	AMOMAX.Q aqrl, rd, rs2, (rs1)
11000	aq	rl	rs2	rs1	100	rd	0101111	AMOMINU.Q aqrl, rd, rs2, (rs1)
11100	aq	rl	rs2	rs1	100	rd	0101111	AMOMAXU.Q aqrl, rd, rs2, (rs1)

#### 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		00010	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
00010	00	00100	rs1	000	00000	1110011	SFENCE.VM rs1
0001000		00101	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]			uimm[4:0]	101	rd	1110011	CSRRWI rd, csr, zimm
csr[11:0]			uimm[4:0]	110	rd	1110011	CSRRSI rd, csr, zimm
csr[11:0]			uimm[4:0]	111	rd	1110011	CSRRCI rd, csr, zimm

#### RV32F Standard Extension for Single-Precision Floating-Point

simm[11:0]			rs1	010	frd	0000111	FLW frd, offset(rs1)
simm[11:5]		frs2	rs1	010	simm[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



31	25	24	20	19	15	14	12	11	7	6	0	
funct5	funct2	rs2	rs1	funct3	rd	opcode						
imm[11:0]			rs1	funct3	rd	opcode						
imm[11:5]		rs2	rs1	funct3	imm[4:0]	opcode						
rs3	funct2	rs2	rs1	funct3	rd	opcode						

**Type-R**  
**Type-I**  
**Type-S**  
**Type-R4**

#### RV32F Standard Extension for Single-Precision Floating-Point contd

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

#### RV32D Standard Extension for Double-Precision Floating-Point

rs1		rs2		rs3		rs4		rs5		rs6		rs7		rs8		rs9		rs10		rs11		rs12		rs13		rs14		rs15		rs16		rs17		rs18		rs19		rs20		rs21		rs22		rs23		rs24		rs25		rs26		rs27		rs28		rs29		rs30		rs31		rs32		rs33		rs34		rs35		rs36		rs37		rs38		rs39		rs40		rs41		rs42		rs43		rs44		rs45		rs46		rs47		rs48		rs49		rs50		rs51		rs52		rs53		rs54		rs55		rs56		rs57		rs58		rs59		rs60		rs61		rs62		rs63		rs64		rs65		rs66		rs67		rs68		rs69		rs70		rs71		rs72		rs73		rs74		rs75		rs76		rs77		rs78		rs79		rs80		rs81		rs82		rs83		rs84		rs85		rs86		rs87		rs88		rs89		rs90		rs91		rs92		rs93		rs94		rs95		rs96		rs97		rs98		rs99		rs100		rs101		rs102		rs103		rs104		rs105		rs106		rs107		rs108		rs109		rs110		rs111		rs112		rs113		rs114		rs115		rs116		rs117		rs118		rs119		rs120		rs121		rs122		rs123		rs124		rs125		rs126		rs127		rs128		rs129		rs130		rs131		rs132		rs133		rs134		rs135		rs136		rs137		rs138		rs139		rs140		rs141		rs142		rs143		rs144		rs145		rs146		rs147		rs148		rs149		rs150		rs151		rs152		rs153		rs154		rs155		rs156		rs157		rs158		rs159		rs160		rs161		rs162		rs163		rs164		rs165		rs166		rs167		rs168		rs169		rs170		rs171		rs172		rs173		rs174		rs175		rs176		rs177		rs178		rs179		rs180		rs181		rs182		rs183		rs184		rs185		rs186		rs187		rs188		rs189		rs190		rs191		rs192		rs193		rs194		rs195		rs196		rs197		rs198		rs199		rs200		rs201		rs202		rs203		rs204		rs205		rs206		rs207		rs208		rs209		rs210		rs211		rs212		rs213		rs214		rs215		rs216		rs217		rs218		rs219		rs220		rs221		rs222		rs223		rs224		rs225		rs226		rs227		rs228		rs229		rs230		rs231		rs232		rs233		rs234		rs235		rs236		rs237		rs238		rs239		rs240		rs241		rs242		rs243		rs244		rs245		rs246		rs247		rs248		rs249		rs250		rs251		rs252		rs253		rs254		rs255		rs256		rs257		rs258		rs259		rs260		rs261		rs262		rs263		rs264		rs265		rs266		rs267		rs268		rs269		rs270		rs271		rs272		rs273		rs274		rs275		rs276		rs277		rs278		rs279		rs280		rs281		rs282		rs283		rs284		rs285		rs286		rs287		rs288		rs289		rs290		rs291		rs292		rs293		rs294		rs295		rs296		rs297		rs298		rs299		rs300		rs301		rs302		rs303		rs304		rs305		rs306		rs307		rs308		rs309		rs310		rs311		rs312		rs313		rs314		rs315		rs316		rs317		rs318		rs319		rs320		rs321		rs322		rs323		rs324		rs325		rs326		rs327		rs328		rs329		rs330		rs331		rs332		rs333		rs334		rs335		rs336		rs337		rs338		rs339		rs340		rs341		rs342		rs343		rs344		rs345		rs346		rs347		rs348		rs349		rs350		rs351		rs352		rs353		rs354		rs355		rs356		rs357		rs358		rs359		rs360		rs361		rs362		rs363		rs364		rs365		rs366		rs367		rs368		rs369		rs370		rs371		rs372		rs373		rs374		rs375		rs376		rs377		rs378		rs379		rs380		rs381		rs382		rs383		rs384		rs385		rs386		rs387		rs388		rs389		rs390		rs391		rs392		rs393		rs394		rs395		rs396		rs397		rs398		rs399		rs400		rs401		rs402		rs403		rs404		rs405		rs406		rs407		rs408		rs409		rs410		rs411		rs412		rs413		rs414		rs415		rs416		rs417		rs418		rs419		rs420		rs421		rs422		rs423		rs424		rs425		rs426		rs427		rs428		rs429		rs430		rs431		rs432		rs433		rs434		rs435		rs436		rs437		rs438		rs439		rs440		rs441		rs442		rs443		rs444		rs445		rs446		rs447		rs448		rs449		rs450		rs451		rs452		rs453		rs454		rs455		rs456		rs457		rs458		rs459		rs460		rs461		rs462		rs463		rs464		rs465		rs466		rs467		rs468		rs469		rs470		rs471		rs472		rs473		rs474		rs475		rs476		rs477		rs478		rs479		rs480		rs481		rs482		rs483		rs484		rs485		rs486		rs487		rs488		rs489		rs490		rs491		rs492		rs493		rs494		rs495		rs496		rs497		rs498		rs499		rs500		rs501		rs502		rs503		rs504		rs505		rs506		rs507		rs508		rs509		rs510		rs511		rs512		rs513		rs514		rs515		rs516		rs517		rs518		rs519		rs520		rs521		rs522		rs523		rs524		rs525		rs526		rs527		rs528		rs529		rs530		rs531		rs532		rs533		rs534		rs535		rs536		rs537		rs538		rs539		rs540		rs541		rs542		rs543		rs544		rs545		rs546		rs547		rs548		rs549		rs550		rs551		rs552		rs553		rs554		rs555		rs556		rs557		rs558		rs559		rs560		rs561		rs562		rs563		rs564		rs565		rs566		rs567		rs568		rs569		rs570		rs571		rs572		rs573		rs574		rs575		rs576		rs577		rs578		rs579		rs580		rs581		rs582		rs583		rs584		rs585		rs586		rs587		rs588		rs589		rs590		rs591		rs592		rs593		rs594		rs595		rs596		rs597		rs598		rs599		rs600		rs601		rs602		rs603		rs604		rs605		rs606		rs607		rs608		rs609		rs610		rs611		rs612		rs613		rs614		rs615		rs616		rs617		rs618		rs619		rs620		rs621		rs622		rs623		rs624		rs625		rs626		rs627		rs628		rs629		rs630		rs631		rs632		rs633		rs634		rs635		rs636		rs637		rs638		rs639		rs640		rs641		rs642		rs643		rs644		rs645		rs646		rs647		rs648		rs649		rs650		rs651		rs652		rs653		rs654		rs655		rs656		rs657		rs658		rs659		rs660		rs661		rs662		rs663		rs664		rs665		rs666		rs667		rs668		rs669		rs670		rs671		rs672		rs673		rs674		rs675		rs676		rs677		rs678		rs679		rs680		rs681		rs682		rs683		rs684		rs685		rs686		rs687		rs688		rs689		rs690		rs691		rs692		rs693		rs694		rs695		rs696		rs697		rs698		rs699		rs700		rs701		rs702		rs703		rs704		rs705		rs706		rs707		rs708		rs709		rs710		rs711		rs712		rs713		rs714		rs715		rs716		rs717		rs718		rs719		rs720		rs721		rs722		rs723		rs724		rs725		rs726		rs727		rs728		rs729		rs730		rs731		rs732		rs733		rs734		rs735		rs736		rs737		rs738		rs739		rs740		rs741		rs742		rs743		rs744		rs745		rs746		rs747		rs748		rs749		rs750		rs751		rs752		rs753		rs754		rs755		rs756		rs757		rs758		rs759		rs760		rs761		rs762		rs763		rs764		rs765		rs766		rs767		rs768		rs769		rs770		rs771		rs772		rs773		rs774		rs775		rs776		rs777		rs778		rs779		rs780		rs781		rs782		rs783		rs784		rs785		rs786		rs787		rs788		rs789		rs790		rs791		rs792		rs793		rs794		rs795		rs796		rs797		rs798		rs799		rs800		rs801		rs802		rs803		rs804		rs805		rs806		rs807		rs808		rs809		rs810		rs811		rs812		rs813		rs814		rs815		rs816		rs817		rs818		rs819		rs820		rs821		rs822		rs823		rs824		rs825		rs826		rs827		rs828		rs829		rs830		rs831		rs832		rs833		rs834		rs835		rs836		rs837		rs838		rs839		rs840		rs841		rs842		rs843		rs844		rs845		rs846		rs847		rs848		rs849		rs850		rs851		rs852		rs853		rs854		rs855		rs856		rs857		rs858		rs859		rs860		rs861		rs862		rs863		rs864		rs865		rs866		rs867		rs868		rs869		rs870		rs871		rs872		rs873		rs874		rs875		rs876		rs877		rs878		rs879		rs880		rs881		rs882		rs883		rs884		rs885		rs886		rs887		rs888		rs889		rs890		rs891		rs892		rs893		rs894		rs895		rs896		rs897		rs898		rs899		rs900		rs901		rs902		rs903		rs904		rs905		rs906		rs907		rs908		rs909		rs910		rs911		rs912		rs913		rs914		rs915		rs916		rs917		rs918		rs919		rs920		rs921		rs922		rs923		rs924		rs925		rs926		rs927		rs928		rs929		rs930		rs931		rs932		rs933		rs934		rs935		rs936		rs937		rs938		rs939		rs940		rs941		rs942		rs943		rs944		rs945		rs946		rs947		rs948		rs949		rs950		rs951		rs952		rs953		rs954		rs955		rs956		rs957		rs958		rs959		rs960		rs961		rs962		rs963		rs964		rs965		rs966		rs967		rs968		rs969		rs970		rs971		rs972		rs973		rs974		rs975		rs976		rs977		rs978		rs979		rs980		rs981		rs982		rs983		rs984		rs985		rs986		rs987		rs988		rs989		rs990		rs991		rs992		rs993		rs994		rs995		rs996		rs997		rs998		rs999		rs1000		rs1001		rs1002		rs1003		rs1004		rs1005		rs1006		rs1007		rs1008		rs1009		rs1010		rs1011		rs1012		rs1013		rs1014		rs1015		rs1016		rs1017		rs1018		rs1019		rs1020		rs1021		rs1022		rs1023		rs1024		rs1025		rs1026		rs1027		rs1028		rs1029		rs1030		rs1031		rs1032		rs1033		rs1034		rs1035		rs1036		rs1037		rs	
-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	-----	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	-------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	--------	--	----	--

31	25	24	20	19	15	14	12	11	7	6	0			
funct5		funct2		rs2		rs1		funct3		rd		opcode		Type-R
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-R <sub>4</sub>

#### RV32D Standard Extension for Double-Precision Floating-Point contd

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

#### RV32Q Standard Extension for Quadruple-Precision Floating-Point

simm[11:0]			rs1	100	frd	0000111	FLQ frd, offset(rs1)
simm[11:5]		frs2	rs1	100	simm[4:0]	0100111	FSQ frs2, offset(rs1)
frs3	11	frs2	frs1	rm	frd	1000011	FMADD.Q rm, frd, frs1, frs2, frs3
frs3	11	frs2	frs1	rm	frd	1000111	FMSUB.Q rm, frd, frs1, frs2, frs3
frs3	11	frs2	frs1	rm	frd	1001011	FNMSUB.Q rm, frd, frs1, frs2, frs3
frs3	11	frs2	frs1	rm	frd	1001111	FNMADD.Q rm, frd, frs1, frs2, frs3
00000	11	frs2	frs1	rm	frd	1010011	FADD.Q rm, frd, frs1, frs2
00001	11	frs2	frs1	rm	frd	1010011	FSUB.Q rm, frd, frs1, frs2
00010	11	frs2	frs1	rm	frd	1010011	FMUL.Q rm, frd, frs1, frs2
00011	11	frs2	frs1	rm	frd	1010011	FDIV.Q rm, frd, frs1, frs2
00100	11	frs2	frs1	000	frd	1010011	FSGNJ.Q frd, frs1, frs2
00100	11	frs2	frs1	001	frd	1010011	FSGNJN.Q frd, frs1, frs2
00100	11	frs2	frs1	010	frd	1010011	FSGNJX.Q frd, frs1, frs2
00101	11	frs2	frs1	000	frd	1010011	FMIN.Q frd, frs1, frs2
00101	11	frs2	frs1	001	frd	1010011	FMAX.Q frd, frs1, frs2
01000	00	00011	frs1	rm	frd	1010011	FCVT.S.Q rm, frd, frs1
01000	11	00000	frs1	rm	frd	1010011	FCVT.Q.S rm, frd, frs1
01000	01	00011	frs1	rm	frd	1010011	FCVT.D.Q rm, frd, frs1
01000	11	00001	frs1	rm	frd	1010011	FCVT.Q.D rm, frd, frs1
01011	11	00000	frs1	rm	frd	1010011	FSQRT.Q rm, frd, frs1
10100	11	frs2	frs1	000	rd	1010011	FLE.Q rd, frs1, frs2
10100	11	frs2	frs1	001	rd	1010011	FLT.Q rd, frs1, frs2
10100	11	frs2	frs1	010	rd	1010011	FEQ.Q rd, frs1, frs2
11000	11	00000	frs1	rm	rd	1010011	FCVT.W.Q rm, rd, frs1
11000	11	00001	frs1	rm	rd	1010011	FCVT.WU.Q rm, rd, frs1
11010	11	00000	rs1	rm	frd	1010011	FCVT.Q.W rm, frd, rs1
11010	11	00001	rs1	rm	frd	1010011	FCVT.Q.WU rm, frd, rs1
11100	11	00000	frs1	001	rd	1010011	FCLASS.Q rd, frs1

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

**RV64Q Standard Extension for Quadruple-Precision Floating-Point (in addition to RV32Q)**

11000	11	00010	frs1	rm	rd	1010011	FCVT.L.Q rm, rd, frs1
11000	11	00011	frs1	rm	rd	1010011	FCVT.LU.Q rm, rd, frs1
11010	11	00010	rs1	rm	frd	1010011	FCVT.Q.L rm, frd, rs1
11010	11	00011	rs1	rm	frd	1010011	FCVT.Q.LU rm, frd, rs1
11100	11	00000	frs1	000	rd	1010011	FMV.X.Q rd, frs1
11110	11	00000	rs1	000	frd	1010011	FMV.Q.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	nzuimm[5:4 9:6 2 3]				rd'	00	C.ADDI4SPN rd, rs1, imm
001	uimm[5:3]		rs1'	uimm[7:6]	frd'	00	C.FLD frd, offset(rs1)
010	uimm[5:3]		rs1'	uimm[2 6]	rd'	00	C.LW rd, offset(rs1)
011	uimm[5:3]		rs1'	uimm[2 6]	frd'	00	C.FLW frd, offset(rs1)
101	uimm[5:3]		rs1'	uimm[7:6]	frs2'	00	C.FSD frs2, offset(rs1)
110	uimm[5:3]		rs1'	uimm[2 6]	rs2'	00	C.SW rs2, offset(rs1)
111	uimm[5:3]		rs1'	uimm[2 6]	frs2'	00	C.FSW frs2, offset(rs1)
000	0	00000			00000	01	C.NOP
000	nzsimm[5]	rs1/rd/= 0			nzsimm[4:0]	01	C.ADDI rd, rs1, imm
001	simm[11 4 9:8 10 6 7 3:1 5]					01	C.JAL rd, offset
010	simm[5]	rs1/rd/= 0			simm[4:0]	01	C.LI rd, rs1, imm
011	nzsimm[9]	rs1/rd= 2			nzsimm[4 6 8:7 5]	01	C.ADDI16SP rd, rs1, imm
011	nzsimm[17]	rd/= {0, 2}			nzsimm[16:12]	01	C.LUI rd, imm
100	0	00	rs1'/rd'		nzuimm[4:0]	01	C.SRLI rd, rs1, imm
100	0	01	rs1'/rd'		nzuimm[4:0]	01	C.SRAI rd, rs1, imm
100	nzsimm[5]	10	rs1'/rd'		nzsimm[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	simm[11 4 9:8 10 6 7 3:1 5]					01	C.J rd, offset
110	simm[8 4:3]	rs1'		simm[7:6 2:1 5]		01	C.BEQZ rs1, rs2, offset
111	simm[8 4:3]	rs1'		simm[7:6 2:1 5]		01	C.BNEZ rs1, rs2, offset
000	0	rs1/rd≠ 0			nzuimm[4:0]	10	C.SLLI rd, rs1, imm
001	uimm[5]	frd			uimm[4:3 8:6]	10	C.FLDSP frd, offset(rs1)
010	uimm[5]	rd/= 0			uimm[4:2 7:6]	10	C.LWSP rd, offset(rs1)
011	uimm[5]	frd			uimm[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	uimm[5:3 8:6]			frs2		10	C.FSDSP frs2, offset(rs1)
110	uimm[5:2 7:6]			rs2		10	C.SWSP rs2, offset(rs1)
111	uimm[5:2 7:6]			frs2		10	C.FSWSP frs2, offset(rs1)

15	13	12	10	9	7	6	5	4	2	1	0	
funct3		imm3		rs1'		imm2		rd'		op		<b>Type-CL</b>
funct3		imm3		rs1'		imm2		rs2'		op		<b>Type-CS</b>
funct3		imm1		rd/rs1				imm5		op		<b>Type-CI</b>
funct3		imm3		rs1'				imm5		op		<b>Type-CB</b>
funct3			imm6					rs2		op		<b>Type-CSS</b>

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

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

#### RV128C Standard Extension for Compressed Instructions (in addition to RV64C)

001	uimm[5:4 8]	rs1'	uimm[7:6]	rd'	00	C.LQ rd, offset(rs1)
101	uimm[5:4 8]	rs1'	uimm[7:6]	rs2'	00	C.SQ rs2, offset(rs1)
001	uimm[5]	rd		uimm[4 9:6]	10	C.LQSP rd, offset(rs1)
101		uimm[5:4 9:6]		rs2	10	C.SQSP rs2, offset(rs1)