Instruction	Operation	Source Registers	Destination Register	Description
FLW	Load a 32-bit floating-point value from memory into a floating-point register	`rs1` (integer), `imm`	`rd` (floating-point)	Load a 32-bit floating-point value from memory (`Mem[rs1 + imm] = rd`)
FSW	Store a 32-bit floating-point value to memory	`rs1` (integer), `rs2` (floating-point), `imm`	None	Store the value of `rs2` to memory at address `rs1 + imm`
FMADD.S	Floating-point multiply-add: `rd = (rs1 * rs2) + rs3`	`rs1`, `rs2`, `rs3` (floating-point)	`rd` (floating-point)	Floating-point multiplication and addition
FMSUB.S	Floating-point multiply-subtract: `rd = (rs1 * rs2) - rs3`	`rs1`, `rs2`, `rs3` (floating-point)	`rd` (floating-point)	Floating-point multiplication and subtraction
FNMSUB.S	Floating-point negative multiply-subtract: `rd = -((rs1 * rs2) - rs3)`	`rs1`, `rs2`, `rs3` (floating-point)	`rd` (floating-point)	Floating-point negative multiply and subtraction
FNMADD.S	Floating-point negative multiply-add: `rd = -((rs1 * rs2) + rs3)`	`rs1`, `rs2`, `rs3` (floating-point)	`rd` (floating-point)	Floating-point negative multiply and addition
FADD.S	Floating-point addition: `rd = rs1 + rs2`	`rs1`, `rs2` (floating-point)	`rd` (floating-point)	Floating-point addition
FSUB.S	Floating-point subtraction: `rd = rs1 - rs2`	`rs1`, `rs2` (floating-point)	`rd` (floating-point)	Floating-point subtraction
FMUL.S	Floating-point multiplication: `rd = rs1 * rs2`	`rs1`, `rs2` (floating-point)	`rd` (floating-point)	Floating-point multiplication
FDIV.S	Floating-point division: `rd = rs1 / rs2`	`rs1`, `rs2` (floating-point)	`rd` (floating-point)	Floating-point division
FSQRT.S	Floating-point square root: `rd = sqrt(rs1)`	`rs1` (floating-point)	`rd` (floating-point)	Floating-point square root
FSGNJ.S	Floating-point sign insertion: `rd = sign(rs1) + magnitude(rs2)`	`rs1`, `rs2` (floating-point)	`rd` (floating-point)	Set the sign of `rd` to that of `rs1` and magnitude of `rs2`
FSGNJN.S	Floating-point negative sign insertion: `rd = -sign(rs1) + magnitude(rs2)`	`rs1`, `rs2` (floating-point)	`rd` (floating-point)	Set the sign of `rd` to the negative of `rs1` and magnitude of `rs2`
FSGNJX.S	Floating-point exclusive OR sign insertion: `rd = xor_sign(rs1, rs2)`	`rs1`, `rs2` (floating-point)	`rd` (floating-point)	Set the sign of `rd` to the XORed signs of `rs1` and `rs2`
FMIN.S	Floating-point minimum: `rd = min(rs1, rs2)`	`rs1`, `rs2` (floating-point)	`rd` (floating-point)	Set `rd` to the minimum of `rs1` and `rs2`

Instruction	Operation	Source Registers	Destination Register	Description
FMAX.S	Floating-point maximum: `rd = max(rs1, rs2)`	`rs1`, `rs2` (floating-point)	`rd` (floating-point)	Set `rd` to the maximum of `rs1` and `rs2`
FCVT.W.S	Convert a floating-point value to a signed integer: `rd = (int) rs1`	`rs1` (floating-point)	`rd` (integer)	Convert floating-point to signed integer (rounding mode controlled by `rm`)
FCVT.WU.S	Convert a floating-point value to an unsigned integer: `rd = (uint) rs1`	`rs1` (floating-point)	`rd` (integer)	Convert floating-point to unsigned integer (rounding mode controlled by `rm`)
FMV.X.W	Move floating-point register to integer register: `rd = rs1`	`rs1` (floating-point)	`rd` (integer)	Move the bit pattern of `rs1` (floating-point) to `rd` (integer)
FEQ.S	Floating-point equality comparison: `rd = (rs1 == rs2)`	`rs1`, `rs2` (floating-point)	`rd` (integer)	Set `rd` to 1 if `rs1 == rs2`, otherwise set `rd` to 0
FLT.S	Floating-point less-than comparison: `rd = (rs1 < rs2)`	`rs1`, `rs2` (floating-point)	`rd` (integer)	Set `rd` to 1 if `rs1 < rs2`, otherwise set `rd` to 0
FLE.S	Floating-point less-than or equal comparison: `rd = (rs1 <= rs2)`	`rs1`, `rs2` (floating-point)	`rd` (integer)	Set `rd` to 1 if `rs1 <= rs2`, otherwise set `rd` to 0
FCLASS.S	Floating-point classification: classify `rs1`	`rs1` (floating-point)	`rd` (integer)	Classify the value of `rs1` (e.g., NaN, Infinity, zero) and store the result in `rd`
FCVT.S.W	Convert signed integer to floating-point: `rd = (float) rs1`	`rs1` (integer)	`rd` (floating-point)	Convert signed integer to floating-point (rounding mode controlled by `rm`)
FCVT.S.WU	Convert unsigned integer to floating-point: `rd = (float) rs1`	`rs1` (integer)	`rd` (floating-point)	Convert unsigned integer to floating-point (rounding mode controlled by `rm`)
FMV.W.X	Move integer to floating-point register: `rd = rs1`	`rs1` (integer)	`rd` (floating-point)	Move the bit pattern of `rs1` (integer) to `rd` (floating-point)