<table>
<thead>
<tr>
<th>Instruction type/opcode</th>
<th>Instruction meaning</th>
</tr>
</thead>
</table>
| **Data transfers**     | \begin{align} &{\text{Move data between registers and memory, or between the integer and FP or special registers; only memory address mode is 16-bit displacement + contents of a GPR}} \\
&{\text{LB, LBU, SB}} \quad \text{Load byte, load byte unsigned, store byte} \\
&{\text{LH, LHU, SH}} \quad \text{Load half word, load half word unsigned, store half word} \\
&{\text{LW, SW}} \quad \text{Load word, store word (to/from integer registers)} \\
&{\text{LF, LD, SF, SD}} \quad \text{Load SP float, load DP float, store SP float, store DP float} \\
&{\text{MOVI2S, MOVS2I}} \quad \text{Move from/to GPR to/from a special register} \\
&{\text{MOVF, MOVDF}} \quad \text{Copy one FP register or a DP pair to another register or pair} \\
&{\text{MOVP2I, MOVF2FP}} \quad \text{Move 32 bits from/to FP registers to/from integer registers} \end{align} |
| **Arithmetic/logical**  | \begin{align} &{\text{Operations on integer or logical data in GPRs; signed arithmetic trap on overflow}} \\
&{\text{ADD, ADDI, ADDU, ADDUI}} \quad \text{Add, add immediate (all immediates are 16 bits); signed and unsigned} \\
&{\text{SUB, SUBI, SUBU, SUBUI}} \quad \text{Subtract, Subtract immediate; signed and unsigned} \\
&{\text{MULT, MULTU, DIV, DIVU}} \quad \text{Multiply and divide; sign and unsigned; operands must be FP registers; all operands take and yield 32-bit values} \\
&{\text{AND, ANDI}} \quad \text{And, and immediate} \\
&{\text{OR, ORI, XOR, XORI}} \quad \text{Or, or immediate, exclusive or, exclusive or immediate} \\
&{\text{LHI}} \quad \text{Load high immediate - loads upper half of register with immediate} \\
&{\text{SLL, SRL, SRA, SLLI, SRLI, SRAI}} \quad \text{Shifts: both immediate (S_I) and variable (S_); shifts are shift left logical, right logical, right arithmetic} \\
&{\text{S__, S__I}} \quad \text{Set conditional: "__" may be LT, GT, LE, GE, EQ, NE} \end{align} |
| **Control**            | \begin{align} &{\text{Conditional branches and jumps; PC-relative or through register}} \\
&{\text{BEQZ, BNEQ}} \quad \text{Branch GPR equal/not equal to zero: 16-bit offset from PC+4} \\
&{\text{BFPT, BFPF}} \quad \text{Test comparison bit in the FP status register and branch: 16-bit offset from PC+4} \\
&{\text{J, JR}} \quad \text{Jumps: 26-bit offset from PC+4 (J) or target in register (JR)} \\
&{\text{IAL, IALR}} \quad \text{Jump and link: save PC+4 in R31, target is PC-relative (JAL) or a register (JALR)} \\
&{\text{TRAP}} \quad \text{Transfer to operating system at a vectored address} \\
&{\text{RFE}} \quad \text{Return to user code from an exception; restore user mode} \end{align} |
| **Floating point**     | \begin{align} &{\text{FP operations on DP and SP formats}} \\
&{\text{ADDD, ADDF}} \quad \text{Add DP, SP numbers} \\
&{\text{SUBD, SUBF}} \quad \text{Subtract DP, SP numbers} \\
&{\text{MULTD, MULTF}} \quad \text{Multiply DP, SP floating point numbers} \\
&{\text{DIVD, DIVF}} \quad \text{Divide DP, SP floating point numbers} \\
&{\text{CVTF2D, CVTF2I, CVTD2F, CVTD2I, CVTIF2, CVTID2}} \quad \text{Convert instructions: CVT}x2y\text{ converts type }x\text{ to type }y\text{, where }x\text{ and }y\text{ are }I\text{ (integer), }D\text{ (double precision), or }F\text{ (single precision).} \\
&{\text{__D, __F}} \quad \text{DP and SP compares: "__" = LT, GT, LE, GE, EQ, NE; sets bit in FP status register} \end{align} |