R-type: 8:6 = opcode, 5:2 = reg\_file read address A, 1:0 = funct

I-type: 8:6 = opcode, 5:0 = immediate

B-type: 8:6 = opcode, 5:2 = target address, 1:0 = funct

|  |  |  |  |
| --- | --- | --- | --- |
| Instruction | Description | Format | Opcode / Funct |
| Add | add RX  R0 <- R0 + RX | R-type | 000 / 00 |
| Sub | sub R1  R0 <- R0 – RX | R-type | 000 / 01 |
| Cmp | cmp RX  R0 = status flags that result from R0 – RX | R-type | 101 / 01 |
| And | and RX  R0 <- R0 & RX | R-type | 001 / 00 |
| Or | or RX  R0 <- R0 | RX | R-type | 001 / 01 |
| Xor | xor R1  R0 <- R0 ^ RX | R-type | 001 / 10 |
| Not | not R1  R0 <- ~RX | R-type | 001 / 11 |
| mov | mov RX  RX <- R0 | R-type | 000 / 10 |
| set | set RX  R0 <- RX | R-type | 000 / 11 |
| lsl | lsl RX  R0 <- R0 << RX | R-type | 011 / 00 |
| lsr | lsr RX  R0 <- R0 >> RX | R-type | 011 / 01 |
| rxr | rxr RX  R0 <- ^RX | R-type | 011 / 10 |
| stri | seti #imm  R0 <- #imm | I-type | 010 |
| ld | lw RX  R0 <- Mem[RX] | R-type | 110 / 00 |
| str | sw RX  Mem[RX] <- R0 | R-type | 110 / 01 |
| beq | beq .label  branches to label based on cmp | B-type | 100 / 00 |
| bne | bne .label | B-type | 100 / 01 |
| ble | ble .label | B-type | 100 / 10 |
| blt | blt .label | B-type | 100 / 11 |
| b | b .label  Unconditional branch to label | B-type | 101 / 00 |