|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Instruction Format** | **Syntax** | **Semantics** |
| 1 | 00000 xxxxxxxxxxx | HALT | Cease instruction issue, dump memory state to file |
| 2 | 00001 xxxxxxxxxxx | NOP | None |
|  |  |  |  |
| 3 | 01000 sss ddd iiiii | ADDI Rd, Rs, immediate | Rd <- Rs + I(sign ext.) |
| 4 | 01001 sss ddd iiiii | SUBI Rd, Rs, immediate | Rd <- I(sign ext.) - Rs |
| 5 | 01010 sss ddd iiiii | XORI Rd, Rs, immediate | Rd <- Rs XOR I(zero ext.) |
| 6 | 01011 sss ddd iiiii | ANDNI Rd, Rs, immediate | Rd <- Rs AND ~I(zero ext.) |
| 7 | 10100 sss ddd iiiii | ROLI Rd, Rs, immediate | Rd <- Rs <<(rotate) I(lowest 4 bits) |
| 8 | 10101 sss ddd iiiii | SLLI Rd, Rs, immediate | Rd <- Rs << I(lowest 4 bits) |
| 9 | 10110 sss ddd iiiii | RORI Rd, Rs, immediate | Rd <- Rs >>(rotate) I(lowest 4 bits) |
| 10 | 10111 sss ddd iiiii | SRLI Rd, Rs, immediate | Rd <- Rs >> I(lowest 4 bits) |
| 11 | 10000 sss ddd iiiii | ST Rd, Rs, immediate | Mem[Rs + I(sign ext.)] <- Rd |
| 12 | 10001 sss ddd iiiii | LD Rd, Rs, immediate | Rd <- Mem[Rs + I(sign ext.)] |
| 13 | 10011 sss ddd iiiii | STU Rd, Rs, immediate | Mem[Rs + I(sign ext.)] <- Rd  Rs <- Rs + I(sign ext.) |
|  |  |  |  |
| 14 | 11001 sss xxx ddd xx | BTR Rd, Rs | Rd[bit i] <- Rs[bit 15-i] for i=0..15 |
| 15 | 11011 sss ttt ddd 00 | ADD Rd, Rs, Rt | Rd <- Rs + Rt |
| 16 | 11011 sss ttt ddd 01 | SUB Rd, Rs, Rt | Rd <- Rt - Rs |
| 17 | 11011 sss ttt ddd 10 | XOR Rd, Rs, Rt | Rd <- Rs XOR Rt |
| 18 | 11011 sss ttt ddd 11 | ANDN Rd, Rs, Rt | Rd <- Rs AND ~Rt |
| 19 | 11010 sss ttt ddd 00 | ROL Rd, Rs, Rt | Rd <- Rs << (rotate) Rt (lowest 4 bits) |
| 20 | 11010 sss ttt ddd 01 | SLL Rd, Rs, Rt | Rd <- Rs << Rt (lowest 4 bits) |
| 21 | 11010 sss ttt ddd 10 | ROR Rd, Rs, Rt | Rd <- Rs >> (rotate) Rt (lowest 4 bits) |
| 22 | 11010 sss ttt ddd 11 | SRL Rd, Rs, Rt | Rd <- Rs >> Rt (lowest 4 bits) |
| 23 | 11100 sss ttt ddd xx | SEQ Rd, Rs, Rt | if (Rs == Rt) then Rd <- 1 else Rd <- 0 |
| 24 | 11101 sss ttt ddd xx | SLT Rd, Rs, Rt | if (Rs < Rt) then Rd <- 1 else Rd <- 0 |
| 25 | 11110 sss ttt ddd xx | SLE Rd, Rs, Rt | if (Rs <= Rt) then Rd <- 1 else Rd <- 0 |
| 26 | 11111 sss ttt ddd xx | SCO Rd, Rs, Rt | if (Rs + Rt) generates carry out  then Rd <- 1 else Rd <- 0 |
|  |  |  |  |
| 27 | 01100 sss iiiiiiii | BEQZ Rs, immediate | if (Rs == 0) then  PC <- PC + 2 + I(sign ext.) |
| 28 | 01101 sss iiiiiiii | BNEZ Rs, immediate | if (Rs != 0) then  PC <- PC + 2 + I(sign ext.) |
| 29 | 01110 sss iiiiiiii | BLTZ Rs, immediate | if (Rs < 0) then  PC <- PC + 2 + I(sign ext.) |
| 30 | 01111 sss iiiiiiii | BGEZ Rs, immediate | if (Rs >= 0) then  PC <- PC + 2 + I(sign ext.) |
| 31 | 11000 sss iiiiiiii | LBI Rs, immediate | Rs <- I(sign ext.) |
| 32 | 10010 sss iiiiiiii | SLBI Rs, immediate | Rs <- (Rs << 8) | I(zero ext.) |
|  |  |  |  |
| 33 | 00100 ddddddddddd | J displacement | PC <- PC + 2 + D(sign ext.) |
| 34 | 00101 sss iiiiiiii | JR Rs, immediate | PC <- Rs + I(sign ext.) |
| 35 | 00110 ddddddddddd | JAL displacement | R7 <- PC + 2  PC <- PC + 2 + D(sign ext.) |
| 36 | 00111 sss iiiiiiii | JALR Rs, immediate | R7 <- PC + 2  PC <- Rs + I(sign ext.) |
|  |  |  |  |
| 37 | 00010 | siic Rs | produce IllegalOp exception. Must provide one source register. |
| 38 | 00011 xxxxxxxxxxx | NOP / RTI | PC <- EPC |