|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pc | MIPS instruction | Functionality | Instructions | Machine code | Range |
| Pc++ | init | Rx = imm | init Rx, imm | 0 xx iiii | Rx[r0,r3]  Imm[0,15] |
| pc++ | add | Rx=Rx+Ry | addreg  Rx, Ry | 010 xx yy | Rx[r0,r3] Ry[r0,r3] |
| Pc++ | slt | Rx=1 f Ry<Rx  Else Rx=0 | lt Rx, Ry | 101 xx yy | Rx[r0,r3] Ry[r0,r3] |
| Pc++ | srl | Shift right one bit | Shift | 0011 xx i | Rx[r0,r3] imm[0,1] |
| Pc++ | beq | Rx==Ry  Pc=pc+mem[Rx]  else pc++ | branch  Rx, Ry | 011 xx yy | Rx[r0,r3] Ry[r0,r3] |
| Pc++ | lw | Rx=Mem[imm] | load Rx, imm | 10 xx iiii | Rx[r0,r3] imm[] |
| Pc++ | sw | Mem[imm]=Rx | store Rx,imm | 0000 xx i | Rx[r0,r3] imm[0,1] |
| Pc+imm | j | imm [] | jump imm | 1 iiiiii | Imm[] |
| Pc++ | andi | Rx = Ry•imm | and Rx, Ry, imm | 11 xx yy i | Rx[r0,r3]  Ry[r0,r3] imm[0,1] |
| Pc++ | xor | Rx = RyRz | Xor Rx, Ry, Rz | 00 xx yy z | Rx[r0,r3] Ry[r0,r3]  Rz[r0] |