Dostępne instrukcje:

|  |  |  |
| --- | --- | --- |
| addi rs1, rd, imm |  |  |
| lui rd, imm |  |  |
| auipc rd, imm |  |  |
| xor rs1, rs2, rd |  |  |
| sltu rs1, rs2, rd |  |  |
| jal rd, imm |  |  |
| jalr rs1, rd, imm |  |  |
| beq rs1, rs2, imm |  |  |
| lw rs1, rd, imm |  |  |
| sw rs1, rs2, imm |  |  |
| sb rs1, rs2, imm |  |  |
| nop | addi x0, x0, 0 | No operation |
| mv rd, rs | addi rd, rs, 0 | Copy register |
|  |  |  |
| la rd, symbol (non-PIC) | auipc rd, delta[31 : 12] + delta[11] | Load absolute address, |
|  | addi rd, rd, delta[11:0] | where delta = symbol − pc |
| la rd, symbol (PIC) | auipc rd, delta[31 : 12] + delta[11] | Load absolute address, |
|  | lw rd, rd, delta[11:0] | where delta = GOT[symbol] − pc |
| lla rd, symbol | auipc rd, delta[31 : 12] + delta[11] | Load local address, |
|  | addi rd, rd, delta[11:0] |  |
| lw rd, symbol | auipc rd, delta[31 : 12] + delta[11] | Load global |
|  | lw rd, delta[11:0](rd) |  |
| s{b|w} rd, symbol, rt | auipc rt, delta[31 : 12] + delta[11] | Store global |
|  | s{b|w} rd, delta[11:0](rt) |  |
|  |  |  |
|  |  |  |
| j offset | jal x0, offset | Jump |
| jal offset | jal x1, offset | Jump and link |
| jr rs | jalr x0, 0(rs) | Jump register |
| jalr rs | jalr x1, 0(rs) | Jump and link register |
| ret | jalr x0, 0(x1) | Return from subroutine |
| call offset | auipc x1, offset[31 : 12] + offset[11] | Call far-away subroutine |
|  | jalr x1, offset[11:0](x1) |  |
| tail offset | auipc x6, offset[31 : 12] + offset[11] | Tail call far-away subroutine |

