Labels:

L1: # mark a place in the code to reference 
# from other instructions

Unconditional Jump: j target_label # the next instruction to execute is the 
# one labeled with the “target_label”

op address

Next_PC[31:0] = PC[31:28], “address” [25:0], 2’boo.

Conditional Branch: beq rs, rt, label # if R[rs] == R[rt] execute instruction at 
# “label”, otherwise execute next inst.

op rs rt address

Next_PC[31:0] = PC[31:0] + 4 + sign_extend(address << 2)

Jump Register: jr rs # execute the instruction at the address 
# held in register “rs”.

op rs rt rd shamt func

Next_PC[31:0] = R[rs]

int sum = 0; i = 0;
do {
    sum += i;
i++
} while (i != 10)

int sum = 0;
for (int i = 0 ; i != x ; i ++) {
    sum += i;
}
