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'b00. **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". rd shamt func op rt rs $Next_PC[31:0] = R[rs]$ int sum = 0, i = 0; int sum = 0; for (int i = 0; $i != x ; i ++) {$ do { sum += i; sum += i; i++ } } while (i != 10)

