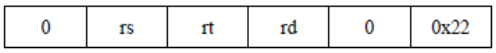
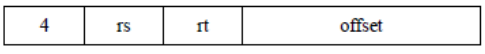
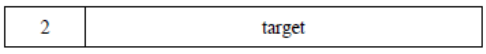
Load word lw rt, address		Load 32-bit word at address into register rt.
Store word sw rt, address		Store the word from register rt at address.
Addition (without overflow) addu rd, rs, rt		Put the sum of the register rs and rt into register rd
Subtract (with overflow) sub rd, rs, rt		
AND immediate andi rt, rs, imm		Put the logical AND of register rs and the zero-extended immediate into register rt
OR or rd, rs, rt		Put the logical OR of registers rs and rt into register rd.
Set less than slt rd, rs, rt		
Branch on equal beq rs, rt, label		Conditionally branch the number of instructions specified by the offset if register rs equals rt.
Jump j target		Unconditionally jump to the instruction at target.