



Computer IS:

add: adds 2 registers, saves result to a register

sub: subtracts 2 registers, saves result to a register

or: preforms bitwise OR on 2 registers, saves result to a register

and: preforms bitwise AND on 2 registers, saves result to a register

xor: preforms bitwise XOR on 2 registers, saves result to a register

sll: preforms a logical left shift on a register, saves result to a register

srl: preforms a logical right shift on a register, saves result to a register

jr: jumps to the line of code held in a register, mostly used to return from subroutine calls

beq: reads 2 register values and branches if they are equal

bne: reads 2 register values and branches if they are not equal

bgt: reads 2 register values and branches if register A is greater than register B

blt: reads 2 register values and branches if register A is less than register B

bge: reads 2 register values and branches if register A is greater than or equal to register B

ble: reads 2 register values and branches if register A is less than or equal to register B

addi: reads 1 register value and adds it to an immediate, saves to a register

ori: reads 1 register value and preforms bitwise OR on it with an immediate, saves result to a register

andi: reads 1 register value and performs bitwise AND on it with an immediate, saves result to a register

lw: loads a value off stack memory, stack pointer-1= address, saves value to a register

sw: loads a value into stack memory, stack pointer= address

j: preforms an unconditional jump

jal: preforms a jump and link (subroutine call), stores return address (PC+1) in the 15th register