

Instruction Formats

a) Register-Reference Instruction

Opcode	R1	R2	R3/SA	X
0 4	5 7	8 10	11 14	15

b) Immediate Instruction

Opcode	R1	R2	IMM	X(Unused)
0 5	6 8	9 11	12 27	28 31

c) Memory-reference Instruction

Opcode	R1	R2	Address	X(Unused)
0 5	6 8	9 11	12 27	28 31

Computer Instructions

Opcode	Operation	Description
00000	No operation	-
00001	Addition with carry	$Reg1 = Reg2 + Reg3 + CF$
00010	Subtraction	$Reg1 = Reg2 - Reg3$
00011	Indirect Addition	$Reg1 = Mem[Reg2] + Mem[Reg3]$
00100	Nand	$Reg1 = Reg2 (Nand) Reg3$
00101	Nor	$Reg1 = Reg2 (Nor) Reg3$
00110	Shift Right Arithmetic	$Reg1 = Reg2 \gg SA$
00111	Shift Left Logical	$Reg1 = Reg2 \ll SA$
10000	Addition	$Reg1 = Reg2 + IMM$
10001	Nand	$Reg1 = Reg2 (Nand) IMM$
10010	Xor	$Reg1 = Reg2 (Xor) IMM$
10011	Load Immediate	$Reg1 = IMM$
100100	Store Immediate	$Mem[Reg1] = IMM$
100101	Push Immediate	$Mem[SP] = IMM, SP--$
11000	Branch if equal	If $Reg1 == Reg2$ then $PC = Address$
11001	Branch if greater than	If $Reg1 > Reg2$ then $PC = Address$
110010	Call if equal	If $Reg1 == Reg2$ then $PC = Address$, Push PC
110011	Call if greater than	If $Reg1 > Reg2$ then $PC = Address$, Push PC
110100	Return	Pop PC
110111	Jump	$PC = Address$
110101	Load	$Reg1 = Mem[Address]$
110110	Store	$Mem[Address] = Reg1$
110111	Pop	$Reg1 = Mem[SP], SP ++$
111000	Push	$Mem[SP] = Reg1, SP --$
111001	Swap	$Swap(Reg1, Reg2)$
111010	SPA	Skip next instruction if AC positive
111011	SNA	Skip next instruction if AC negative
111100	SZA	Skip next instruction if AC zero