Instruction		Description	Operation	Type	Opcode	Funct
add	rd,rs,rt	Add	rd	R	000000	100000
sub	rd,rs,rt	Subtract	rd = rs - rt	R	000000	100010
addi	rt,rs,imm	Add Immediate	rt = rs + imm	I	001000	
addu	rd,rs,rt	Add Unsigned	rd = rs + rt	R	000000	100001
subu	rd,rs,rt	Subtract Unsigned	rd = rs - rt	R	000000	100011
addiu	rt,rs,imm	Add Immediate Unsigned	rt = rs + imm	I	001001	
mult	rs,rt	Multiply	{hi, lo} = rs * rt	R	000000	011000
div	rs,rt	Divide	lo = rs / rt; hi = rs % rt	R	000000	011010
multu	rs,rt	Multiply Unsigned	{hi, lo} = rs * rt	R	000000	011001
divu	rs,rt	Divide Unsigned	lo = rs / rt; hi = rs % rt	R	000000	011011
mfhi	rd	Move From Hi	rd = hi	R	000000	010000
mflo	rd	Move From Lo	rd = lo	R	000000	010010
and	rd,rs,rt	And	rd = rs & rt	R	000000	100100
or	rd,rs,rt	Or	rd = rs | rt	R	000000	100101
nor	rd,rs,rt	Nor	rd = ~(rs | rt)	R	000000	100111
xor	rd,rs,rt	Exclusive Or	rd = rs ^ rt	R	000000	100110
andi	rt,rs,imm	And Immediate	rt = rs & imm0	I	001100	
ori	rt,rs,imm	Or Immediate	rt = rs | imm0	I	001101	
xori	rt,rs,imm	Exclusive Or Immediate	rt = rs ^ imm0	I	001110	
sll	rd,rt,sh	Shift Left Logical	rd = rt << sh	R	000000	000000
srl	rd,rt,sh	Shift Right Logical	rd = rt >>> sh	R	000000	000010
sra	rd,rt,sh	Shift Right Arithmetic	rd = rt >> sh	R	000000	000011
sllv	rd,rt,rs	Shift Left Logical Variable	rd = rt << rs	R	000000	000100
srlv	rd,rt,rs	Shift Right Logical Variable	rd = rt >>> rs	R	000000	000110
srav	rd,rt,rs	Shift Right Arithmetic Variable	rd = rt >> rs	R	000000	000111
slt	rd,rs,rt	Set if Less Than	rd = rs < rt ? 1 : 0	R	000000	101010
sltu	rd,rs,rt	Set if Less Than Unsigned	rd = rs < rt ? 1 : 0	R	000000	101011
slti	rt,rs,imm	Set if Less Than Immediate	rt = rs < imm ? 1 : 0	I	001010	
sltiu	rt,rs,imm	Set if Less Than Immediate Unsigned	rt = rs < imm ? 1 : 0	I	001011	
j	addr	Jump	PC = PC & 0xF0000000 | (addr<< 2)	J	000010	
jal	addr	Jump And Link	ra = PC + 8; PC = PC & 0xF0000000 | (addr<< 2)	J	000011	
jr	rs	Jump Register	PC = rs	R	000000	001000
jalr	rs	Jump And Link Register	ra = PC + 8; PC = rs	R	000000	001001
beq	rt,rs,imm	Branch if Equal	if (rs == rt) PC += 4 + (imm << 2)	I	000100	
bne	rt,rs,imm	Branch if Not Equal	if (rs != rt) PC += 4 + (imm << 2)	I	000101	
syscall		System Call		R	000000	001100
lui	rt,imm	Load Upper Immediate	rt = imm << 16	I	001111	
lb	rt,imm(rs)	Load Byte	rt = SignExt(MB[rs + imm])	I	100000	
lbu	rt,imm(rs)	Load Byte Unsigned	rt = MB[rs + imm] & 0xFF	I	100100	
lh	rt,imm(rs)	Load Half	rt = SignExt(MH[rs + imm])	I	100001	
lhu	rt,imm(rs)	Load Half Unsigned	rt = MH[rs + imm] & 0xFFFF	I	100101	
lw	rt,imm(rs)	Load Word	rt = MW[rs + imm]	I	100011	
sb	rt,imm(rs)	Store Byte	MB[rs + imm] = rt	I	101000	
sh	rt,imm(rs)	Store Half	MH[rs + imm] = rt	I	101001	
sw	rt,imm(rs)	Store Word	M4[rs + imm] = rt	I	101011	
ll	rt,imm(rs)	Load Linked	rt = MW[rs + imm]	I	110000	
Sc	rt,imm(rs)	Store Conditional	MW[rs + imm] = rt; rt = atomic ? 1 : 0	I	111000	

Pseudo instruction		Description	Operation	Type	Opcode	Funct
bge	rx,ry,imm	Branch if Greather Than or Equal	if (rs >= rt) PC += 4 + (imm << 2)			
bgt	rx,ry,imm	Branch if Greather Than	if (rs > rt) PC += 4 + (imm << 2)			
ble	rx,ry,imm	Branch if Less Than or Equal	if (rs <= rt) PC += 4 + (imm << 2)			
blt	rx,ry,imm	Branch if Less Than	if (rs < rt) PC += 4 + (imm << 2)			
la	rx,label	Load Address	rx = Address(label)			
li	rx,imm	Load 32-bit Immediate	rx = imm			
move	rx,ry	Move	rx = ry			
nop		No Operation				

Register Number	Register name
$0	$zero
$1	$at
$2 – $3	$v0–$v1
$4 – $7	$a0–$a3
$8 – $15	$t0–$t7
$16 – $23	$s0–$s7
$24 – $25	$t8–$t9
$26 – $27	$k0–$k1
$28	$gp
$29	$sp
$30	$fp
$31	$ra
	hi
	lo
	pc