The MIPS Blue Card

Mnemonic	Format	Opcode / Funct	Syntax	Name	0
sll	R	0 / 0	$rd, $rt, shamt	Shift Left Logical	$rs = 0
srl	R	0 / 2	$rd, $rt, shamt	Shift Right Logical	$rs = 0
jr	R	0 / 8	$rs	Jump Register	$rd, $rt, shamt = 0
jalr	R	0 / 9	$rd, $rs	Jump and Link Register	$rt, shamt = 0
mfhi	R	0 / 10 (hex)	$rd	Move from HI	$rs, $rt, shamt = 0
mflo	R	0 / 12 (hex)	$rd	Move from LO	$rs, $rt, shamt = 0
mult	R	0 / 18 (hex)	$rs, $rt	Multiply	$rd, shamt = 0
and	R	0 / 24 (hex)	$rd, $rs, $rt	And	shamt = 0
or	R	0 / 25 (hex)	$rd, $rs, $rt	Or	shamt = 0
xor	R	0 / 26 (hex)	$rd, $rs, $rt	Xor	shamt = 0
nor	R	0 / 27 (hex)	$rd, $rs, $rt	Nor	shamt = 0
add	R	0 / 20 (hex)	$rd, $rs, $rt	Add	shamt = 0
addu	R	0 / 21 (hex)	$rd, $rs, $rt	Add Unsigned	shamt = 0
sub	R	0 / 22 (hex)	$rd, $rs, $rt	Subtract	shamt = 0
subu	R	0 / 23 (hex)	$rd, $rs, $rt	Subtract Unsigned	shamt = 0
slt	R	0 / 2A (hex)	$rd, $rs, $rt	Set on Less Than	shamt = 0
sltu	R	0 / 2B (hex)	$rd, $rs, $rt	Set on Less Than Unsigned	shamt = 0
clz	R	1C / 20 (hex)	$rd, $rs	Count Leading Zeros	$rt, shamt = 0
j	J	2	target	Jump	
jal	J	3	target	Jump and Link	
beq	I	4	$rs, $rt, offset	Branch on Equal	
bne	I	5	$rs, $rt, offset	Branch on Not Equal	
blez	I	6	$rs, offset	Branch on Less Than or Equal Zero	$rt = 0
bgtz	I	7	$rs, offset	Branch on Greater Than Zero	$rt = 0
addi	I	8	$rt, $rs, immediate	Add Immediate	
addiu	I	9	$rt, $rs, immediate	Add Immediate Unsigned	
slti	I	A (hex)	$rt, $rs, immediate	Set on Less Than Immediate	
sltiu	I	B (hex)	$rt, $rs, immediate	Set on Less Than Imm. Unsigned	
andi	I	C (hex)	$rt, $rs, immediate	And Immediate	
ori	I	D (hex)	$rt, $rs, immediate	Or Immediate	
lui	I	F (hex)	$rt, immediate	Load Upper Immediate	$rs = 0
lb	I	20 (hex)	$rt, offset($rs)	Load Byte	
lh	I	21 (hex)	$rt, offset($rs)	Load Half Word	
lw	I	23 (hex)	$rt, offset($rs)	Load Word	
lbu	I	24 (hex)	$rt, offset($rs)	Load Byte Unsigned	
lhu	I	25 (hex)	$rt, offset($rs)	Load Half Word Unsigned	
sb	I	28 (hex)	$rt, offset($rs)	Store Byte	
sh	I	29 (hex)	$rt, offset($rs)	Store Half Word	
sw	I	2B (hex)	$rt, offset($rs)	Store Word	
sc	I	2C (hex)	$rt, offset($rs)	Store Conditional	
ldlc	I	32 (hex)	$rt, offset($rs)	Load Doubleword Left Coprocessor	