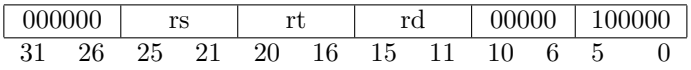
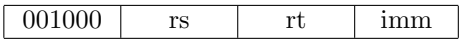
add



Add instruction (trap on overflow)
Reg[rd] ← Reg[rs] + Reg[rt]

addi

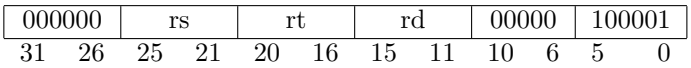


Add immediate (trap on overflow)
Reg[rt] ← Reg[rs] + $\text{imm}^{\pm}$

addiu

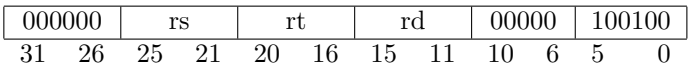
Add immediate (ignore overflow, note imm is still sign extended)
Reg[rt] ← Reg[rs] + $\text{imm}^{\pm}$

addu



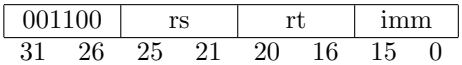
Add instruction (ignore overflow)
Reg[rd] ← Reg[rs] + Reg[rt]

and



Bitwise logical AND
Reg[rd] ← Reg[rs] AND Reg[rt]

andi



Bitwise logical AND with immediate
Reg[rt] ← Reg[rs] AND $\text{imm}^{\emptyset}$

bc1f

010001	01000	cc	0	0	imm
31 26	25 21	20 16	15	0	

Branch if coprocessor 1 (FPU) false

PC $\leftarrow$ (cc = 0) ? PC + offset$^{\pm}$: PC + 1

bc1t

010001	01000	cc	0	1	imm
31 26	25 21	20 16	15	0	

Branch if coprocessor 1 (FPU) true

PC $\leftarrow$ (cc = 1) ? PC + offset$^{\pm}$: PC + 1

beq

000100	rs	rt	offset
31 26	25 21	20 16	15 0

Branch if equal

PC $\leftarrow$ (rs = rt) ? PC + offset$^{\pm}$: PC + 1

bgez

000001	rs	0001	offset
31 26	25 21	20 16	15 0

Branch if greater than or equal to zero

PC $\leftarrow$ (rs $\geq$ 0) ? PC + offset$^{\pm}$: PC + 1

bgtz

000111	rs	0000	offset
31 26	25 21	20 16	15 0

Branch if greater than zero

PC $\leftarrow$ (rs $>$ 0) ? PC + offset$^{\pm}$: PC + 1

blez

000110	rs	0000	offset
31 26	25 21	20 16	15 0

Branch if less than or equal to zero

PC $\leftarrow$ (rs $\leq$ 0) ? PC + offset$^{\pm}$: PC + 1

bltz

000001	rs	0000	offset
31 26	25 21	20 16	15 0

Branch if less than zero

PC $\leftarrow$ (rs $<$ 0) ? PC + offset$^{\pm}$: PC + 1

bne

Branch if less not equal

PC $\leftarrow$ (rs $\neq$ rt) ? PC + offset$^{\pm}$: PC + 1

ceq

cfc1

cge

cgt

cle

clt

cne

ctc1

fabs

fadd

fdiv

fma

fmul

fneg

fsqrt

fsub

j

jr

lb

lbu

lh

lhu

lui

lw

lwc0

lwc1

mfc0

mfc1

mov

mtc0

mtc1

nor

or

ori

pre