

1 I-Type

In the following table: $m = m_a(ea(c))$, where $ea(c) = rs(c) +_{32} sxtimm(c)$

opc	Mnemonic	Assembler-Syntax	d	Effect
Data Transfer				
100 011	lw	lw <i>rt rs imm</i>	4	rt = m
101 011	sw	sw <i>rt rs imm</i>	4	m = rt
Arithmetic, Logical Operation, Test-and-Set				
001 000	addi	addi <i>rt rs imm</i>		rt = rs + sxt(imm)
001 001	addiu	addiu <i>rt rs imm</i>		rt = rs + sxt(imm)
001 010	slti	slti <i>rt rs imm</i>		rt = (rs < sxt(imm) ? 1 : 0)
001 011	sltui	sltui <i>rt rs imm</i>		rt = (rs < sxt(imm) ? 1 : 0)
001 100	andi	andi <i>rt rs imm</i>		rt = rs \wedge zxt(imm)
001 101	ori	ori <i>rt rs imm</i>		rt = rs \vee zxt(imm)
001 110	xori	xori <i>rt rs imm</i>		rt = rs \oplus zxt(imm)
001 111	lui	lui <i>rt imm</i>		rt = imm0 ¹⁶
opc	rt	Mnemonic	Assembler-Syntax	Effect
Branch				
000 001	00000	bltz	bltz <i>rs imm</i>	pc = pc + (rs < 0 ? imm00 : 4)
000 001	00001	bgez	bgez <i>rs imm</i>	pc = pc + (rs \geq 0 ? imm00 : 4)
000 100		beq	beq <i>rs rt imm</i>	pc = pc + (rs = rt ? imm00 : 4)
000 101		bne	bne <i>rs rt imm</i>	pc = pc + (rs \neq rt ? imm00 : 4)
000 110	00000	blez	blez <i>rs imm</i>	pc = pc + (rs \leq 0 ? imm00 : 4)
000 111	00000	bgtz	bgtz <i>rs imm</i>	pc = pc + (rs > 0 ? imm00 : 4)

2 R-type

opcode	fun	Mnemonic	Assembler-Syntax	Effect
Shift Operation				
000000	000 000	sll	sll <i>rd rt sa</i>	rd = sll(rt,sa)
000000	000 010	srl	srl <i>rd rt sa</i>	rd = srl(rt,sa)
000000	000 011	sra	sra <i>rd rt sa</i>	rd = sra(rt,sa)
000000	000 100	sllv	sllv <i>rd rt rs</i>	rd = sll(rt,rs)
000000	000 110	srlv	srlv <i>rd rt rs</i>	rd = srl(rt,rs)
000000	000 111	srav	srav <i>rd rt rs</i>	rd = sra(rt,rs)
Arithmetic, Logical Operation				
000000	100 000	add	add <i>rd rs rt</i>	rd = rs + rt
000000	100 001	addu	addu <i>rd rs rt</i>	rd = rs + rt
000000	100 010	sub	sub <i>rd rs rt</i>	rd = rs - rt
000000	100 011	subu	subu <i>rd rs rt</i>	rd = rs - rt
000000	100 100	and	and <i>rd rs rt</i>	rd = rs \wedge rt
000000	100 101	or	or <i>rd rs rt</i>	rd = rs \vee rt
000000	100 110	xor	xor <i>rd rs rt</i>	rd = rs \oplus rt
000000	100 111	nor	nor <i>rd rs rt</i>	rd = rs $\bar{\vee}$ rt
Test Set Operation				
000000	101 010	slt	slt <i>rd rs rt</i>	rd = (rs < rt ? 1 : 0)
000000	101 011	sltu	sltu <i>rd rs rt</i>	rd = (rs < rt ? 1 : 0)
Jumps				
000000	001 000	jr	jr <i>rs</i>	pc = rs
000000	001 001	jalr	jalr <i>rd rs</i>	rd = pc + 4 pc = rs

3 J-type

opc	Mnemonic	Assembler-Syntax	Effect
Jumps			
000 010	j	j <i>iindex</i>	pc = $bin_{32}(pc)[31:28]iindex00$
000 011	jal	jal <i>iindex</i>	R31 = pc + 4 pc = $bin_{32}(pc)[31:28]iindex00$