NAME	Type	OpCode	R1	R2	R3	Shamt	Funct	lmm	26-Bit	Description
abs	. , , , ,	Pseudo								AbsoluteValue
add	R	0	R1	R2	RD	0	32			Addition
addi	I	8	RS	RD				lmm		AdditionImmediate
addu	R	0	R1	R2	RD	0	33			AdditionUnsigned(withoutoverflow)
addiu	- 1	9	RS	RD				lmm		AdditionImmediateUnsigned
and	R	0	R1	R2	RD	0	36			AND
andi	I	12	RS	RD				lmm		ANDImmediate
b		Pseudo								Branch
beq	ı	4 .	R1	R2				Offset/4		BranchifEqual(ifR1=R2)
beqz		Pseudo								BranchonEqualZero
bge		Pseudo Pseudo								BranchifGreaterThanorEqual
bgequ	12	1	RS	1				Offset/4		BranchifGreaterThanorEqualUnsigned BranchifRSGreaterorEqualtoZero
bgez bgezal	12	1	RS	17				Offset/4		BranchandLinkifRSGreaterorEqualtoZero
bgezai	12	Pseudo	NO	17				Oliseu4		BranchifGreaterThan
bgtu		Pseudo								BranchifGreaterThanUnsigned
bgtz	12	7	RS	0				Offset/4		BranchifRSGreaterThanZero
ble		Pseudo								BranchifLessThanorEqual
bleu		Pseudo								BranchifLessThanorEqualUnsigned
blez	12	6	RS	0				Offset/4		BranchifRSLessorEqualtoZero
blt		Pseudo								BranchifLessThan
bltu		Pseudo								BranchifLessThanUnsigned
bltz	12	1	RS	0				Offset/4		BranchifRSLessThanZero
bltzal	l2	1	RS	16				Offset/4		BranchandLinkifRSLessThanZero
bne	I	5	R1	R2				Offset/4		BranchifNotEqual(ifR1<>R2)
bnez	-	Pseudo								BranchifNotEqualZero
div	R6	0	R1	R2	0	0	26			DivideR1/R2(HI-remainder,LO-quotient)
divu	R6	0	R1	R2	0	0	27		T	DivideR1/R2Unsigned
J iol	J	2 3							Target	
jal jalr	J R5	0	Addr	0	RD	0	9		rarget	JumpAndLink JumpandLinkRegister
jan jr	R4	0	Addr	0	0	0	8			JumpRegister
la	114	Pseudo	Addi	0	-	<u> </u>				LoadAddress
lb	- 1	32	Addr	RD				Offset		LoadByte
lbu	Ī	36	Addr	RD				Offset		LoadByteUnsigned
ld		Pseudo								LoadDoubleWord
lh	- 1	33	Addr	RD				Offset		LoadHalfWord
lhu	- 1	37	Addr	RD				Offset		LoadHalfWordUnsigned
li		Pseudo								LoadImmediate
lui	I1	15	0	RD				lmm		LoadUpperImmediate(Immediatex2^16)
lw	l l	35	Addr	RD				Offset		LoadWord
lwl	!	34	Addr	RD				Offset		LoadWordLeft(4highbytesof8-byteword)
lwr	I D4	38	Addr	RD		•	40	Offset		LoadWordRight(4lowbytesof8-byteword)
mfhi	R1	0	0	0	RD	0	16			MoveFromHI
mflo move	R1	Pseudo	U	0	RD	0	18			MoveFromLO Move
mthi	R4	0	RS	0	0	0	17			MoveToHI
mtlo	R4	0	RS	0	0	0	19			MoveToLO
mul	114	Pseudo	ING.	0	-	<u> </u>	13			Multiply(withoutoverflow)
mulo		Pseudo								Multiply(withoverflow)
mulou		Pseudo								MultiplyUnsigned(withoverflow)
mult	R6	0	R1	R2	0	0	24			Multiply
multu	R6	0	R1	R2	0	0	25			MultiplyUnsigned
neg		Pseudo								Negate(withoverflow)
negu		Pseudo								Negate(withoutoverflow)
nor	R	0	R1	R2	RD	0	39			NOR
not		Pseudo								NOT
or	R	0	R1	R2	RD	0	37			OR
ori	I	_ 13	RS	RD				lmm		ORImmediate
rem		Pseudo								Remainder

NAME	Туре	OpCode	R1	R2	R3	Shamt	Funct	lmm	26-Bit	Description
remu		Pseudo								UnsignedRemainder
rol		Pseudo								RotateLeft
ror		Pseudo								RotateRight
sb	1	40	Addr	RS				Offset		StoreByte
sd		Pseudo								StoreDoubleWord
seq		Pseudo								SetifEqual
sge		Pseudo								SetifGreaterThanorEqual
sgeu		Pseudo								SetifGreaterThanorEqualUnsigned
sgt		Pseudo								SetifGreaterThan
sgtu		Pseudo								SetifGreaterThhanUnsigned
sh	I	41	Addr	RS				Offset		StoreHalfWord
sll	R3	0	-	RS	RD	SHAMT	0			ShiftLeftLogicalSHAMT
sllv	R	0	Shift	RS	RD	0	4			ShiftLeftLogicalVariable
slt	R	0	R1	R2	RD	0	42			SetifLessThan(RD=1ifR1 <r2)< td=""></r2)<>
slti	I	10	RS	RD				lmm		SetifRSLessThanImmediate
sltiu	<u> </u>	11	RS	RD				lmm		SetifRSLessThanImmediateUnsigned
sltu	R	0	R1	R2	RD	0	43			SetifLessThanUnsigned(RD=1ifR1 <r2)< td=""></r2)<>
sne		Pseudo								SetifNotEqual
sra	R3	0		RS	RD	SHAMT	3			ShiftRightArithmeticSHAMT
srav	R	0	Shift	RS	RD	0	7			ShiftRightArithmeticVariable
srl	R3	0		RS	RD	SHAMT	2			ShiftRightLogicalSHAMT
srlv	R	0	Shift	RS	RD	0	6			ShiftRightLogicalVariable
sub	R	0	R1	R2	RD	0	34			SubtractR1-R2
subu	R	0	R1	R2	RD	0	35			SubtractR1-R2(withoutoverflow)
SW	!	43	Addr	RS				Offset		StoreWord
swl		42	Addr	RS				Offset		StoreWordLeft(4highbytesof8-byteword)
swr	ı	_ 46	Addr	RS				Offset		StoreWordRight(4lowbytesof8-byteword)
ulh		Pseudo								UnalignedLoadHalfWord
ulhu		Pseudo								UnalignedLoadHalfWordUnsigned
ulw		Pseudo								UnalignedLoadWord
ush		Pseudo								UnalignedStoreHalfWord
usw	_	Pseudo	D4	D0	D.D.	0	00			UnalignedStoreWord
xor .	R	0	R1	R2	RD	0	38			XOR
xori		14	RS	RD				lmm		XORImmediate