

MIPS ASSEMBLY LANGUAGE

<u>Instruction</u>	<u>Example</u>	<u>Meaning</u>
Add	Add \$1,\$2,\$3	$\$1 = \$2 + \$3$
Subtract	Sub \$1,\$2,\$3	$\$1 = \$2 - \$3$
Add immediate	Addi \$1,\$2,100	$\$1 = \$2 + 100$
Add unsigned	Addu \$1,\$2,\$3	$\$1 = \$2 + \$3$
Subtract unsigned	Subu \$1,\$2,\$3	$\$1 = \$2 - \$3$
Add immediate unsigned	Addiu \$1,\$2,100	$\$1 = \$2 + 100$
Multiply	mult \$2,\$3	Hi, lo = $\$2 * \3
Multiply unsigned	Multu \$2,\$3	Hi, lo = $\$2 * \3
Divide	Div \$2,\$3	Lo = $\$2 / \3 ; hi remainder
Divide unsigned	Divu \$2,\$3	Lo = $\$2 / \3 ; hi remainder
Move from Hi	Mfhi \$1	$\$1 = \text{Hi}$
Move from Lo	Mflo \$1	$\$1 = \text{Lo}$
And	And \$1,\$2,\$3	$\$1 = \$2 \& \$3$
Or	Or \$1,\$2,\$3	$\$1 = \$2 \$3$
And immediate	Andi \$1,\$2,100	$\$1 = \$2 \& 100$
Or immediate	Ori \$1,\$2,100	$\$1 = \$2 100$
Shift left logical	Sll \$1,\$2,5	$\$1 = \$2 \ll 5$
Shift right logical	Srl \$1,\$2,5	$\$1 = \$2 \gg 5$
Load word	Lw \$1,100(\$2)	$\$1 = \text{memory} [\$2 + 100]$
Load address	la \$1,addr	$\$1 = \text{value of addr label}$
Store word	Sw \$1, 100(\$2)	$\text{memory} [\$2 + 100] = \1
Load upper immediate	Lui \$1,100	$\$1 = 100 \times 2^{16}$
Branch if equal	Beq \$1,\$2,100	if ($\$1 = \2), $\text{PC} = \text{PC} + 4 + 100$
Branch if not equal	Bne \$1,\$2,100	if ($\$1 \neq \2), $\text{PC} = \text{PC} + 4 + 100$
Set if less than	Slt \$1,\$2,\$3	if ($\$2 < \3), $\$1 = 1$ else $\$1 = 0$
Set if less than immediate	Slti \$1,\$2,100	if ($\$2 < 100$), $\$1 = 1$ else $\$1 = 0$
Set if less than unsigned	Sltu \$1,\$2,\$3	if ($\$2 < \3), $\$1 = 1$ else $\$1 = 0$
Set if less than imm. uns.	Sltiu \$1,\$2,100	if ($\$2 < 100$), $\$1 = 1$ else $\$1 = 0$
Jump	J 10000	$\text{PC} = 10000$
Jump register	Jr \$31	$\text{PC} = \$31$
Jump and link	Jal 10000	$\$31 = \text{PC} + 4$; $\text{PC} = 10000$

Additional ones that we have considered:

subi, xor,xori, nor, nand, nori, nandi, sa (shift arithmetic), sl (shift logical), rot (shift circular)

Procedure Call Naming Convention

Register	Naming Convention	Use
0	\$zero or \$0	Integer value zero
1	\$at	Reserved for assembler
2,3	\$v0-\$v1	Return from function
4-7	\$a0-\$a3	Pass first 4 arguments to routines
8-15,24,25	\$t0-\$t9	Temporary values
16-23	\$s0-\$s7	Saved values
26-27	\$k0-\$k1	Reserved for assembler and operating system
28	\$gp	Global pointer
29	\$sp	Stack pointer
30	\$fp	Frame pointer
31	\$ra	Return address

\$s0-\$s7: the caller does not expect these registers to have changed when returning from a call

\$t0-\$t9: these can be changed by the called procedure