

MIPS Reference Sheet

Arithmetic Instructions			
Instruction	Opcode/Function	Syntax	Operation
add	100000	\$d, \$s, \$t	\$d = \$s + \$t
addu	100001	\$d, \$s, \$t	\$d = \$s + \$t
addi	001000	\$t, \$s, i	\$t = \$s + SE(i)
addiu	001001	\$t, \$s, i	\$t = \$s + SE(i)
div	011010	\$s, \$t	lo = \$s / \$t; hi = \$s % \$t
divu	011011	\$s, \$t	lo = \$s / \$t; hi = \$s % \$t
mult	011000	\$s, \$t	hi:lo = \$s * \$t
multu	011001	\$s, \$t	hi:lo = \$s * \$t
sub	100010	\$d, \$s, \$t	\$d = \$s - \$t
subu	100011	\$d, \$s, \$t	\$d = \$s - \$t
Logical Instructions			
Instruction	Opcode/Function	Syntax	Operation
and	100100	\$d, \$s, \$t	\$d = \$s & \$t
andi	001100	\$t, \$s, i	\$t = \$s & ZE(i)
nor	100111	\$d, \$s, \$t	\$d = ~(\$s \$t)
or	100101	\$d, \$s, \$t	\$d = \$s \$t
ori	001101	\$t, \$s, i	\$t = \$s ZE(i)
xor	100110	\$d, \$s, \$t	\$d = \$s ^ \$t
xori	001110	\$t, \$s, i	\$t = \$s ^ ZE(i)
Shift Instructions			
Instruction	Opcode/Function	Syntax	Operation
sll	000000	\$d, \$t, a	\$d = \$t << a
sllv	000100	\$d, \$t, \$s	\$d = \$t << \$s
sra	000011	\$d, \$t, a	\$d = \$t >> a
srav	000111	\$d, \$t, \$s	\$d = \$t >> \$s
srl	000010	\$d, \$t, a	\$d = \$t >>> a
srlv	000110	\$d, \$t, \$s	\$d = \$t >>> \$s
Data Movement Instructions			
Instruction	Opcode/Function	Syntax	Operation
mfhi	010000	\$d	\$d = hi
mflo	010010	\$d	\$d = lo
mthi	010001	\$s	hi = \$s
mtlo	010011	\$s	lo = \$s
Branch Instructions			
Instruction	Opcode/Function	Syntax	Operation
beq	000100	\$s, \$t, label	if (\$s == \$t) pc <- label
bgtz	000111	\$s, label	if (\$s > 0) pc <- label
blez	000110	\$s, label	if (\$s <= 0) pc <- label
bne	000101	\$s, \$t, label	if (\$s != \$t) pc <- label

Last Name: _____ First Name: _____

Jump Instructions			
Instruction	Opcode/Function	Syntax	Operation
j	000010	label	pc <- label
jal	000011	label	\$ra = pc; pc <- label
jalr	001001	\$s	\$ra = pc; pc = \$s
jr	001000	\$s	pc = \$s
Comparison Instructions			
Instruction	Opcode/Function	Syntax	Operation
slt	101010	\$d, \$s, \$t	\$d = (\$s < \$t)
sltu	101001	\$d, \$s, \$t	\$d = (\$s < \$t)
slti	001010	\$t, \$s, i	\$t = (\$s < SE(i))
sltiu	001001	\$t, \$s, i	\$t = (\$s < SE(i))
Memory Instructions			
Instruction	Opcode/Function	Syntax	Operation
lb	100000	\$t, i (\$s)	\$t = SE (MEM [\$s + i]:1)
lbu	100100	\$t, i (\$s)	\$t = ZE (MEM [\$s + i]:1)
lh	100001	\$t, i (\$s)	\$t = SE (MEM [\$s + i]:2)
lhu	100101	\$t, i (\$s)	\$t = ZE (MEM [\$s + i]:2)
lw	100011	\$t, i (\$s)	\$t = MEM [\$s + i]:4
sb	101000	\$t, i (\$s)	MEM [\$s + i]:1 = LB (\$t)
sh	101001	\$t, i (\$s)	MEM [\$s + i]:2 = LH (\$t)
sw	101011	\$t, i (\$s)	MEM [\$s + i]:4 = \$t
Pseudo Instructions			
Instruction	Opcode/Function	Syntax	Operation
la	N/A	\$t, label	\$t = address of label
li	N/A	\$t, i	\$t = i
blt	N/A	\$s, \$t, label	if (\$s < \$t) pc <- label
bltu	N/A	\$s, \$t, label	if (\$s < \$t) pc <- label
bgt	N/A	\$s, \$t, label	if (\$s > \$t) pc <- label
bgtu	N/A	\$s, \$t, label	if (\$s > \$t) pc <- label
ble	N/A	\$s, \$t, label	if (\$s <= \$t) pc <- label
bleu	N/A	\$s, \$t, label	if (\$s <= \$t) pc <- label
bge	N/A	\$s, \$t, label	if (\$s >= \$t) pc <- label
bgeu	N/A	\$s, \$t, label	if (\$s >= \$t) pc <- label
syscall	N/A		Call system trap, trapcode is in \$v0

Registers		Registers	
0	\$zero	16-23	\$s0 - \$s7
1	\$at	24-25	\$t8 - \$s9
2-3	\$v0 - \$v1	26-27	\$k0 - \$k1
4-7	\$a0 - \$a3	28-29	\$gp , \$sp
8-15	\$t0 - \$t7	30-31	\$fp , \$ra