Instrucciones y PseudoInstrucciones

Conjunto de Instrucciones

Aritméticas

Operaciones de Suma Rd <- Rs + Rt

add Rd, Rs, Rt suma con overflow add $t1, $t2, $t3
addi Rd, Rs, Inm suma Inmediata addi $t1, $t2, 5

Operaciones de Resta Rd <- Rs - Rt

sub Rd, Rs, Rt resta con overflow add $t1, $t2, $t3
subi Rd, Rs, Inm resta Inmediata addi $t1, $t2, 5

neg Rdest, Rsrc
Pone el negativo del contenido del registro Rsrc en el registro Rdest

Operaciones de Mutiplicación y División

mul Rdest, Rsrc1, Src2 Rd <- Rs * Rt
mult Rs, Rt
Multiplica los registros Rs y Rt y deja la palabra de orden inferior en el registro Lo y la de orden superior en el registro Hi.

div Rs, Rt
Divide el registro Rs entre el registro Rt y deja el cociente en el registro Lo y el resto en el registro Hi.

Instrucciones de Transferencia

move Rdest, Rsrc Rdest <- Rsrc

Instrucciones que operan sobre los registros Hi y Lo

mfhi Rd Rd <- Hi
mflo Rd Rd <- Lo

mthi Rd Hi <- Rd
mtlo Rd Lo <- Rd

Intrucciones Lógicas

and Rd, Rs, Rt conjunction Rd <- Rs AND Rt
andi Rd, Rs, Inm conjunction Inmediata Rd <- Rs AND Inm

or Rd, Rs, Rt disjunction Rd <- Rs Or Rt
ori Rd, Rs, Inm disjunction Inmediata Rd <- Rs Or Inm

not Rd, Rs negación Rd <- NOT (Rs)

xor Rd, Rs, Rt or exclusivo Rd <- Rs Xor Rt
xori Rd, Rs, Inm or exclusivo Inmediato Rd <- Rs Xor Inm

Instrucciones de Salto y Bifurcación

b etiqueta Salta incondicionalmente a la dirección de la etiqueta

beq Rs, Rt, etiqueta Salta condicionalmente a la etiqueta si Rs = Rt

beqz Rs, etiqueta Salta condicionalmente a la etiqueta si Rs = 0

bge Rs, Rt, etiqueta Salta condicionalmente a la etiqueta si Rs >= Rt

bgez Rs, etiqueta Salta condicionalmente a la etiqueta si Rs >= 0

bgezal Rs, etiqueta
Salta condicionalmente a la etiqueta si Rs >= 0 y además almacena la dirección de la siguiente instrucción en el registro $31

bgt Rs, Rt, etiqueta Salta condicionalmente a la etiqueta si Rs > Rt

bgtz Rs, etiqueta Salta condicionalmente a la etiqueta si Rs > 0

ble Rs, Rt, etiqueta Salta condicionalmente a la etiqueta si Rs <= Rt

blez Rs, etiqueta Salta condicionalmente a la etiqueta si Rs <= 0

blt Rs, Rt, etiqueta Salta condicionalmente a la etiqueta si Rs < Rt

bltz Rs, etiqueta Salta condicionalmente a la etiqueta si Rs < 0

bne Rs, Rt, etiqueta Salta condicionalmente a la etiqueta si Rs <> Rt

bnez Rs, etiqueta Salta condicionalmente a la etiqueta si Rs <> 0

j etiqueta Bifurca incondicionalmente a la dirección de la etiqueta

jr Rdest Bifurca incondicionalmente a la dirección que contiene el registro Rdest

jal etiqueta
Bifurca incondicionalmente a la etiqueta y además almacena la dirección de la siguiente instrucción en el registro $31

jalr Rdest
Bifurca incondicionalmente a la a la dirección que contiene el registro Rdest y además almacena la dirección de la siguiente instrucción en el registro $31

Desplazamiento y rotación

rol Rdest, Rsrc1, Rsrc2 desplazamiento cíclico (rotación) a la izquierda
ror Rdest, Rsrc1, Rsrc2 desplazamiento cíclico (rotación) a la derecha

Desplaza circularmente el registro Rsrc1 a la izquierda (derecha) la cantidad de bits indicada por el contenido del registro Rsrc2 y coloca el resultado en el registro Rdest.

sll Rdest, Rsrc1, Rsrc2 desplazamiento lógico a la izquierda
srl Rdest, Rsrc1, Rsrc2 desplazamiento lógico a la derecha

Desplaza los bits del registro Rsrc1 a la izquierda (derecha) la cantidad de bits indicada por el contenido del registro Rsrc2 y coloca el resultado en el registro Rdest. No toma en cuenta el bit de signo.

sla Rdest, Rsrc1, Rsrc2 desplazamiento lógico a la izquierda
sra Rdest, Rsrc1, Rsrc2 desplazamiento lógico a la derecha

Desplaza los bits del registro Rsrc1 a la izquierda (derecha) la cantidad de bits indicada por el contenido del registro Rsrc2 y coloca el resultado en el registro Rdest. Toma en cuenta el bit de signo.

Instrucciones de carga y almacenamiento

li Rdest, Inm Rdest <- Inm Load Inmediately

la Rdest, dir Rdest <- Dirección calculada de dir

lb Rdest, direccion
Carga el byte que esta en direccion en el Registro Rdest, extendiendo el signo.

lbu Rdest, direccion lb que no extiende el bit de signo.

lh Rdest, direccion
Carga la media palabra (halfword) que esta en direccion en el Registro Rdest, extendiendo el signo.

lhu Rdest, direccion lh que no extiende el bit de signo.
lw Rdest, direccion
Carga la palabra (word) que esta en direccion en el Registro Rdest.

ld Rdest, direccion
Carga la doble palabra (double word - 64 bits) que está partir de direccion en el Registro Rdest.

sb Rdest, direccion
Almacena el byte inferior del Registro Rdest en direccion.

sh Rdest, direccion
Almacena la media palabra (halfword) inferior del Registro Rdest en direccion.

sw Rdest, direccion
Almacena la palabra (word) que esta en el Registro Rdest en direccion.

sd Rdest, direccion
Almacena el valor de la doble palabra (double word - 64 bits) que está en los registro Rdest y Rdest+1 a partir de direccion.

Service	System call code	Arguments	Result
print_int	1	$a0 = integer	
print_float	2	$f12 = float	
print_double	3	$f12 = double	
print_string	4	$a0 = string	
read_int	5		integer (in $v0)
read_float	6		float (in $f0)
read_double	7		double (in $f0)
read_string	8	$a0 = buffer, $a1 = length	
sbrk	9	$a0 = amount	address (in $v0)
exit	10		
print_char	11	$a0 = char	
read_char	12		char (in $a0)
open	13	$a0 = filename (string), $a1 = flags, $a2 = mode	file descriptor (in $a0)
read	14	$a0 = file descriptor, $a1 = buffer, $a2 = length	num chars read (in $a0)
write	15	$a0 = file descriptor, $a1 = buffer, $a2 = length	num chars written (in $a0)
close	16	$a0 = file descriptor	
exit2	17	$a0 = result	