## Instrucciones y PseudoInstrucciones

o otnominibuto of co	Sala Collabolamente a la	Salta condicionalmente a la	of a standardicipand of all a	Salta condicionalmente a la	Salta condicionalmente a la	bgezal Rs, etiqueta	odica contractoramiente a la cuqueta si nos 72 o y además almacena la dirección de la siguiente instrucción en el registro \$31
0 0	etiqueta si Rs = Rt	beqz Rs, etiqueta	ordered or the ordered	bge Rs, Rt, enqueta etiqueta si Rs >= Rt	bgez Rs, etiqueta etiqueta si Rs >= 0	bgezal Rs, etiqueta	además almacena la dire en el registro \$31
Conjunto de Instrucciones	Aritméticas	Operaciones de Suma Rd <- Rs + Rt	add Rd, Rs, Rt suma con overflow add \$t1,	ətz, ətə addi Rd, Rs, Inm suma Inmediata addi \$t1, \$12.5	Operaciones de Resta Rd <- Rs - Rt	sub Rd, Rs, Rt resta con overflow add \$t1, \$t2,	subi Rd, Rs, Inm resta Inmediata addi \$t1,\$t2, 5

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

Rdest	Operaciones de Mutiplicación y División	Rdest, Rsrc1, Src2 Rd <- Rs * Rt 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	tro Hi.
registro Rdest	Operaciones	mul Rdest,	Aultiplica los orden inferior	el reaistro Hi.
_	_			•

	registro Rt y deja e	gistro Hi.
	registro	o en el re
	(D	rest
	entre el	y e
	Rs	12 12
Rs, Rt	el registro	cociente en el registro Lo y el resto en el registro Hi.
Rs,	Œ	e
− ≥	Divide	cocient

		2		
insferencia	Rdest <- Rsrc	Instrucciones que operan sobre los registros Hi y Lo	Rd <- Hi Rd <- Lo	H.
Instrucciones de Transferencia	Rdest, Rsrc	ucciones que ol	Rd	Rd
Instr	move	Instr	mfhi mflo	m

Hi <- Rd Lo <- Rd	as	conjunction Rd <- Rs AND Rt conjunction Inmediata Rd <- Rs	disjunction Rd <-Rs Or Rt disjunction Inmediata Rd <- Rs
mthi Rd mtlo Rd	Intrucciones Lógicas	and Rd, Rs, Rt andi Rd, Rs, Inm AND Inm	or Rd, Rs, Rt ori Rd, Rs, Inm Or Inm

Rd <- NOT (Rs)	Rd <- Rs Xor Rt Inmediato Rd <- Rs
negación	or exclusivo Rd <- or exclusivo Inmediato
not Rd, Rs	xor Rd, Rs, Rt xori Rd, Rs, Inm Xor Inm
_	* * *

ón
cac
ij
m
>
2
a
(J)
de
Jes
.0
ဗ
Ž
nst
-

Salta incondicionalmente a la	
b etiqueta	dirección de la etiqueta

<u>a</u>	<u>a</u>	a
Ø	σ	a
Salta condicionalmente a la	Salta condicionalmente a la	Salta condicionalmente a la
Salta	Salta	Salta
beq Rs, Rt, etiqueta etiqueta si Rs = Rt	oeqz Rs, etiqueta etiqueta si Rs = 0	bge Rs, Rt, etiqueta
22 0	22 W	עב

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

Salta condicionalmente a la	Salta condicionalmente a la	Salta condicionalmente a la
bgt Rs, Rt, etiqueta etiqueta si Rs > Rt	bgtz Rs, etiqueta etiqueta si Rs > 0	ble Rs, Rt, etiqueta etiqueta si Rs <= Rt

g g g	nte a la	ite a la
Salta condicionalmente a la	Salta condicionalmente a la	Salta condicionalmente a la
Salta	Salta	Salta
blez Rs, ettqueta etiqueta si Rs <= 0	blt Rs, Rt, etiqueta etiqueta si Rs < Rt	bltz Rs, etiqueta etiqueta si Rs < 0

Salta condicionalmente a la	Salta condicionalmente a la
Salta c	Salta o
bne Rs, Rt, etiqueta etiqueta si Rs <> Rt	bnez Rs, etiqueta etiqueta si Rs <> 0

Bifurca incondicionalmente a la	Bifurca incondicionalmente a la	2000
j etiqueta dirección de la etiqueta	jr Rdest dirección que contiene el	-

registro Rdest

jal etiqueta	2000	900		_			3
	DII CA III COI INCOI IN INCOI INCOI INCOI IN INC			U	ש	anhna	ğ
almacen	almacena la dirección de la siguiente inst	de	π	0	len	ė	ns.

Bifurca incondicionalmente a la etiqueta y además almacena la dirección de la siguiente instrucción en el registro \$31 in Rdesti Pares incondicionalmente o la cinculta con el conditionalmente o la cinculta con el conditionalmente o la cinculta cinculta con el conditionalmente o la cinculta
Birurca incondicionalmente a la a la dirección que
contiene el registro Rdest y además almacena la
direction of the circuitante incention of the chicagon

Desplazamiento y rotación	Rdest, Rsrc1, Rsrc2 desplazamiento cíclico	Rdest, Rsrc1, Rsrc2 desplazamiento cíclico
<b>Desplazamient</b>	rol Rdest, Rsrc1, I	ror Rdest, Rsrc1,

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.

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

Ihu Rdest, direccion Ih que no extiende el bit de signo.
W Rdest, direccion
Carga la palatra (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.

SII.	sll Rdest,	Rdest, Rsrc1, Rsrc2	Rsrc2	desplazamiento lógico a la	lógico	<u>a</u>	
srl derect	Rdest, I	Rsrc1, Rsrc2	Rsrc2	desplazamiento lógico a la	lógico	<u>а</u>	

stro Rsrc1 a la izquierda s indicada por el contenido el resultado en el registro oit de signo.	Rdest, Rsrc1, Rsrc2 desplazamiento lógico a la
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 Reset. No toma en cuenta el bit de signo.	sla Rdest, Rsrc1, Rsrc2 izquierda

m	<ul><li>la Rdest, Rsrc1, Rsrc2</li></ul>	Rsrc1,	Rsrc2	desplazamiento lógico a la	lógico	a	m
zquie	:quierda						
g	Rdest,	Rdest, Rsrc1, Rsrc2	Rsrc2	desplazamiento lógico a la	lógico	a	m
erecha	ha						

erda	nido	istro	
izqui	conte	el reg	
a a	or el	en e	
Desplaza los bits del registro Rsrc1 a la izquierda	dicada p	esultado	signo.
registro	bits inc	oca el r	l bit de s
de	ad de	8	enta e
bits	cantid	src2	en cu
sol a	<u>a</u>	tro F	oma (
Desplaza	(derecha	del registro Rsrc2 y coloca el resultado en el registro	Rdest. Toma en cuenta el bit de signo.

sh Rdest, direction 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.

## Instrucciones de carga y almacenamiento

Load Inmediate	Rdest <- Dirección calculada de dir
mul	Direcci
Rdest <- Inm	Rdest <-
Rdest, Inm	Rdest, dir
=	a

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

Ib que no extiende el bit de

lbu Rdest, direccion signo.

ad Indeat, dileccion
Almacena el valor de la doble palabra (double word
bits) que está en los registro Rdest y Rdest+1 a part
direccion.

sd Kdest, 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	Arguments	Result
print int	1	\$a0 = integer	
print_float	2	\$f12 = float	
print_double	6	\$f12 = double	
print_string	4	\$a0 = string	
read_int	ro.		integer (in \$v0)
read_float	9		float (in \$f0)
read_double	7		double (in \$f0)
read_string	80	\$a0 = buffer, \$a1 = length	
sbrk	o	\$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 \$30)
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	\$30 = result	