Registros del MIPS y usos

Nombre del registro	**Número**	**Uso**
zero	0	Constante 0
at	1	Reservada para ensamblador
v0	2	Evaluación de expresiones y resultado de funciones
v1	3	Evaluación de expresiones y resultado de funciones
a0	4	Argumento 1
a1	5	Argumento 2
a2	6	Argumento 3
a3	7	Argumento 4
t0..t7	8..15	Temporal (no se guarda valor entre llamadas)
s0..s7	16..23	Temporal (el valor se guarda entre llamadas)
t8, t9	24, 25	Temporal (no se guarda valor entre llamadas)
k0, k1	26, 27	Reservado para el kernel del sistema operativo
gp	28	Puntero al área global
sp	29	Puntero de pila
fp	30	Puntero de marco de pila
ra	31	Dirección de retorno, usada por llamadas a función

Servicios del sistema

Servicio	**Código de llamada**	**Argumentos**	**Resultado**
print_int	1	$a0 = entero	
print_float	2	$f12 = real (32 bits)	
print_double	3	$f12 = real (64 bits)	
print_string	4	$a0 = cadena	
read_int	5		Entero (en $v0)
read_float	6		Real 32 bits (en $f0)
read_double	7		Real 64 bits (en $f0)
read_string	8	$a0=buffer, $a1 = longitud	
sbrk	9	$a0 = cantidad	Dirección (en $v0)
exit	10		

Directivas del ensamblador

.ascii cadena → Almacena la cadena en memoria, pero no termina con null (‘\0’).
.assciiz cadena → Almacena la cadena en memoria y pone un null (‘\0’) al final de esta.
.byte b1, ..., bn → Almacena n valores en bytes sucesivos de memoria.
.data → Las siguientes definiciones de datos que aparezcan se almacenan en el segmento de datos. Puede llevar un argumento que indica la dirección a partir de donde se empezarán a almacenar los datos.
.double d1, ..., dn → Almacena n valores reales de doble precisión en direcciones consecutivas de memoria.
.extern etiqueta n→ Declara que los datos almacenados a partir de *etiqueta* ocupan *n* bytes y que *etiqueta* es un símbolo global. Esta directiva permite al ensamblador almacenar datos en una zona del segmento de datos que puede ser accedida a través del registro $gp.
.flota f1, ..., fn → Almacena n reales de precisión simple en posiciones consecutivas de memoria.
.globl símbolo → Declara un símbolo global que se puede referenciar desde otros programas.
.half h1, ..., hn → Almacena n números de 16 bits en medias palabras consecutivas.
.text → Las instrucciones que siguen a esta directiva se ponen en el segmento de código. Pude llevar un parámetro que indica donde empieza la zona de código.
.word w1, ..., wn → Almacena n cantidades de 32 bits (una palabra) en posiciones consecutivas de memoria.

Juego de instrucciones del SPIM

Instrucciones aritméticas y lógicas

En todas las instrucciones siguientes, Src2 puede ser tanto un registro como un valor inmediato (un entero de 16 bits).

abs Rdest, Rsrc	Valor absoluto
add Rdest, Rsrc1, Src2	Suma con desbordamiento
addu Rdest, Rsrc1, Src2	Suma sin desbordamiento
and Rdest, Rsrc1, Src2	Operación lógica AND
div Rsrc1, Rsrc2	Divide con desbordamiento. Deja el cociente en el registro *lo* y el resto en el registro *hi*
divu Rsrc1, Rsrc2	Divide sin desbordamiento. Deja el cociente en el registro *lo* y el resto en el registro *hi*
div Rdest, Rsrc1, Src2	Divide con desbordamiento
div Rdest, Rsrc1, Src2	Divide sin desbordamiento
mul Rdest, Rsrc1, Src2	Multiplica sin desbordamiento
mulo Rdest, Rsrc1, Src2	Multiplica con desbordamiento
mulou Rdest, Rsrc1, Src2	Multiplicación con signo y con desbordamiento
mult Rsrc1, Rsrc2	Multiplica, la parte baja del resultado se deja en el registro *lo* y la parte alta en el registro *hi*
mult Rsrc1, Rsrc2	Multiplica con signo, la parte baja del resultado se deja en el registro *lo* y la parte alta en el registro *hi*
neg Rdest, Rsrc	Niega el valor (detecta desbordamiento)
negu Rdest, Rsrc	Niega el valor (sin desbordamiento)
nor Rdest, Rsrc1, Src2	Operación Lógica NOR
not Rdest, Rsrc	Operación Lógica NOT
or Rdest, Rsrc1, Src2	Operación Lógica OR
rem Rdest, Rsrc1, Src2	Resto (Módulo), pone el resto de dividir Rsrc1 por Src2 en el registro Rdest.
rol Rdest, Rsrc1, Src2	Rotar a la izquierda
ror Rdest, Rsrc1, Src2	Rotar a la derecha
sll Rdest, Rsrc1, Src2	Desplazamiento lógico de bits a la izquierda
srl Rdest, Rsrc1, Src2	Desplazamiento lógico de bits a la derecha
sra Rdest, Rsrc1, Rsrc2	Desplazamiento aritmético de bits a la derecha
sub Rdest, Rsrc1, Src2	Resta (con desbordamiento)
subu Rdest, Rsrc1, Src2	Resta (sin desbordamiento)
xor Rdest, Rsrc1, Src2	Operación Lógica XOR

Instrucciones manipulación de constantes

li Rdest, inmediato	Cargar valor inmediato
lui Rdest, inmediato	Cargar los 16 bits de la parte baja del valor inmediato en la parte alta del registro. Los bits de la parte baja se pone a 0.

Instrucciones de comparación

En todas las instrucciones siguientes, Src2 puede ser un registro o un valor inmediato (de 16 bits).

seq Rdest, Rsrc1, Src2	Pone Rdest a 1 si Rsrc1 y Src2 son iguales, en otro caso pone 0.
sge Rdest, Rsrc1, Src2	Pone Rdest a 1 si Rsrc1 es mayor o igual a Src2, y 0 en otro caso (para números con signo).
sgeu Rdest, Rsrc1, Src2	Pone Rdest a 1 si Rsrc1 es mayor o igual a Src2, y 0 en otro caso (para números sin signo).
sgt Rdest, Rsrc1, Src2	Pone Rdest a 1 si Rsrc1 es mayor que Src2, y 0 en otro caso (para números con signo).
sgtu Rdest, Rsrc1, Src2	Pone Rdest a 1 si Rsrc1 es mayor que Src2, y 0 en otro caso (para números sin signo).
sle Rdest, Rsrc1, Src2	Pone Rdest a 1 si Rsrc1 es menor o igual a Src2, en otro caso pone 0 (para números con signo).
sleu Rdest, Rsrc1, Src2	Pone Rdest a 1 si Rsrc1 es menor o igual a Src2, en otro caso pone 0 (para números sin signo).
slt Rdest, Rsrc1, Src2	Pone Rdest a 1 si Rsrc1 es menor a Src2, en otro caso pone 0 (para números con signo).
sltu Rdest, Rsrc1, Src2	Pone Rdest a 1 si Rsrc1 es menor a Src2, en otro caso pone 0 (para números sin signo).
sne Rdest, Rsrc1, Src2	Pone Rdest to 1 si el registro Rsrc1 no es igual a Src2 y 0 en otro caso.

Instrucciones de bifurcación y salto

En todas las instrucciones siguientes, Src2 puede ser un registro o un valor inmediato. Las instrucciones de bifurcación (branch) usan un desplazamiento de 16 bits con signo; por lo que se puede saltar 2^15-1 instrucciones hacia delante o 2^15 instrucciones hacia atrás. Las instrucciones de salto (jump) contienen un campo de dirección de 26 bits.

b etiqueta	Bif. incondicional a la instrucción que está en etiqueta.
beq Rsrc1, Src2, etiqueta	Bif. condicional si Rsrc1 es igual a Src2.
beqz Rsrc, etiqueta	Bif. condicional si el registro Rsrc es igual a 0.
bge Rsrc1, Src2, etiqueta	Bif. condicional si el registro Rsrc1 es mayor o igual a Src2 (con signo).
bgeu Rsrc1, Src2, etiq	Bif.. condicional si el registro Rsrc1 es mayor o igual a Src2 (sin signo).
bgez Rsrc, etiqueta	Bif. condicional si el registro Rsrc es mayor o igual a 0.
bgezal Rsrc, etiqueta	Bif. condicional si el registro Rsrc es mayor o igual a 0. Guarda la dirección actual en el registro $ra ($31).
bgt Rsrc1, Src2, etiqueta	Bif. condicional si el registro Rsrc1 es mayor que Src2 (con signo).
bgtu Rsrc1, Src2, etiqueta	Bif. condicional si el registro Rsrc1 es mayor que Src2 (sin signo.
bgtz Rsrc, etiqueta	Bif. condicional si Rsrc es mayor que 0.
ble Rsrc1, Src2, etiqueta	Bif. Condicional si Rsrc1 es menor o igual a Src2 (con signo).
bleu Rsrc1, Src2, etiqueta	Bif. Condicional si Rsrc1 es menor o igual a Src2 (sin signo).
blez Rsrc, etiqueta	Bif. Condicional si Rsrc es menor o igual a 0.
bltzal Rsrc, etiqueta	Bif. Condicional si Rsrc es menor que 0. Guarda la dirección actual en el registro $ra ($31).
blt Rsrc1, Src2, etiqueta	Bif. Condicional si Rsrc1 es menor que Src2 (con signo).
bltu Rsrc1, Src2, etiqueta	Bif. Condicional si Rsrc1 es menor que Src2 (sin signo).
bltz Rsrc, etiqueta	Bif. Condicional si Rsrc es menor que 0.
bne Rsrc1, Src2, etiqueta	Bif. Condicional si Rsrc1 no es igual a Src2.
bnez Rsrc, etiqueta	Bif. Condicional si Rsrc no es igual a 0.
j etiqueta	Salto incondicional.
jal etiqueta	Salto incondicional, almacena la dirección actual en $ra ($31).
jalr Rsrc	Salto incondicional, almacena la dirección actual en $ra ($31).
jr Rsrc	Salto incondicional.

Instrucciones de carga

la Rdest, dirección	Carga dirección en Rdest (el valor de dirección, no el contenido)
lb Rdest, dirección	Carga el byte de la dirección especificada y extiende el signo
lbu Rdest, dirección	Carga el byte de la dirección especificada, no extiende el signo
ld Rdest, dirección	Carga Rdest y Rdest + 1 con el valor del double (64 bits) que se encuentra a partir de la dirección especificada.
lh Rdest, dirección	Carga 16 bits de la dirección especificada, se extiende el signo
lhu Rdest, dirección	Carga 16 bits de la dirección especificada, no se extiende signo
lw Rdest, dirección	Carga una palabra de la dirección especificada.
lwcz Rdest, dirección	Carga una palabra en el registro Rdest del coprocesador z.
lwl Rdest, dirección	Carga bytes en la palabra por la izquierda de la palabra desalineada.
lwr Rdest, dirección	Carga bytes en la palabra por la derecha de la palabra desalineada.
ulh Rdest, dirección	Carga media palabra (16 bits) de palabras desalineadas.
ulhu Rdest, dirección	Carga media palabra (16 bits) de palabras desalineadas (extiende el signo).
ulw Rdest, dirección	Carga una palabra (32 bits) de direcciones de memoria desalineadas.

Instrucciones de almacenamiento

sb Rsrc, dirección	Almacena el byte más bajo de Rsrc en la dirección indicada.
sd Rsrc, dirección	Almacena un double (64 bits) en la dirección indicada, el valor de 64 bits es proviene de Rsrc y Rsrc + 1.
sh Rsrc, dirección	Almacena la media palabra (16 bits) baja de un registro en la dirección de memoria indicada.
sw Rsrc, dirección	Almacena la Rsrc en la dirección indicada.
swcz Rsrc, dirección	Almacena el valor del registro Rsrc del coprocesador z.
swl Rsrc, dirección	Almacena los bytes del registro empezando por la derecha en la dirección de memoria indicada.
swr Rsrc, dirección	Almacena los bytes del registro empezando por la izquierda en la dirección de memoria indicada.
ush Rsrc, dirección	Almacena la parte baja del registro en una dirección desalineada.
usw Rsrc, dirección	Almacena la palabra del registro en una dirección desalineada.

Instrucciones de movimiento de datos

move Rdest, Rsrc	Mueve el contenido del registro Rsrc al registro Rdest.
mfhi Rdest	Mueve el contenido del registro HI al registro Rdest.
mflo Rdest	Mueve el contenido del registro LO al registro Rdest.
mthi Rsrc	Mueve el contenido del registro Rsrc al registro HI.
mtlo Rsrc	Mueve el contenido del registro Rsrc al registro LO.
mfcz Rdest, CPsrc	Mueve el contenido del registro CPsrc del coprocesador z al registro de la CPU Rdest.
mfc1.d Rdest, FRsrc1	Mueve el contenido de los registros FRrc1 y FRsrc1+1 a los registros de la CPU Rdest y Rdest + 1.
mtcz Rsrc, CPdest	Mueve los contenidos del registro Rsrc de la CPU al registro Cpdest del coprocesador z.

Instrucciones de excepción y trap

rfe	Retorno de una excepción, restaura el registro de estado.
syscall	Llamada al sistema, el registro v0 contiene el número de la llamada al sistema.
break n	Provoca una excepción de valor n.
nop	No operation, no hace nada.

Instrucciones del coprocesador – 32 bits

abs.s fd, fs	Valor absoluto
add.s fd, fs, ft	Suma de los registros fs y ft
c.eq.s fs, ft	Compara fs y ft, si son iguales pone el flag de condición de coma flotante true. Se deben utilizar bc1t o bclf para comprobar el valor del flag.
c.le.s fs, ft	Si fs es menor o igual que ft pone el flag de condición de coma flotante true. Se deben utilizar bc1t o bclf para comprobar el valor del flag.
c.lt.s fs, ft	Si fs es menor que ft pone el flag de condición de coma flotante true. Se deben utilizar bc1t o bclf para comprobar el valor del flag.
div.s fd, fs, ft	Divide fs entre ft y deja el resultado en fd
l.s fdest, direc	Lee de la dirección de memoria indicada un real y lo almacena en fd.
mov.s fd, fs	Mueve el contenido del registro fs al registro fd.
mul.s fd, fs, ft	Multiplica los registros fs y ft y deja su resultado en fd.
neg.s fd, fs	Niega el valor del real contenido en fs y lo almacena el fd
s.s fd, direc	Escribe en la dirección de memoria indicada el real contenido en el registro fd
sub.s fd, fs, ft	Resta el contenido de dos registros y almacen el resultado en fd.
bc1t dir_relativa	Pasa a ejecutar el código de la dirección indicada si el flag de coma flotante es 1 (true).
bc1f dir_relativa	Pasa a ejecutar el código de la dirección indicada si el flag de coma flotante es 0 (false).