

Set de instrucciones del Simulador WinMIPS64

Instrucciones de Transferencia de Datos		
lb	$r_d, \text{Inm}(r_i)$	Copia en $r_d$ un byte (8 bits) desde la dirección $(\text{Inm}+r_i)$ (con extensión del signo)
lbu	$r_d, \text{Inm}(r_i)$	Copia en $r_d$ un byte (8 bits) desde la dirección $(\text{Inm}+r_i)$ (sin extensión del signo)
sb	$r_{\bar{e}}, \text{Inm}(r_i)$	Guarda los 8 bits menos significativos de $r_{\bar{e}}$ en la dirección $(\text{Inm}+r_i)$
lh	$r_d, \text{Inm}(r_i)$	Copia en $r_d$ un half-word (16 bits) desde la dir. $(\text{Inm}+r_i)$ (con extensión del signo)
lhu	$r_d, \text{Inm}(r_i)$	Copia en $r_d$ un half-word (16 bits) desde la dir. $(\text{Inm}+r_i)$ (sin extensión del signo)
sh	$r_{\bar{e}}, \text{Inm}(r_i)$	Guarda los 16 bits menos significativos de $r_{\bar{e}}$ a partir de la dirección $(\text{Inm}+r_i)$
lw	$r_d, \text{Inm}(r_i)$	Copia en $r_d$ un word (32 bits) desde la dir. $(\text{Inm}+r_i)$ (con extensión del signo)
lwu	$r_d, \text{Inm}(r_i)$	Copia en $r_d$ un word (32 bits) desde la dir. $(\text{Inm}+r_i)$ (sin extensión del signo)
sw	$r_{\bar{e}}, \text{Inm}(r_i)$	Guarda los 32 bits menos significativos de $r_{\bar{e}}$ a partir de la dirección $(\text{Inm}+r_i)$
ld	$r_d, \text{Inm}(r_i)$	Copia en $r_d$ un double word (64 bits) desde la dirección $(\text{Inm}+r_i)$
sd	$r_{\bar{e}}, \text{Inm}(r_i)$	Guarda $r_{\bar{e}}$ a partir de la dirección $(\text{Inm}+r_i)$
l.d	$f_d, \text{Inm}(r_i)$	Copia en $f_d$ un valor en punto flotante (64 bits) desde la dirección $(\text{Inm}+r_i)$
s.d	$f_{\bar{e}}, \text{Inm}(r_i)$	Guarda $f_{\bar{e}}$ a partir de la dirección $(\text{Inm}+r_i)$
mov.d	$f_d, f_{\bar{e}}$	Copia el valor del registro $f_{\bar{e}}$ al registro $f_d$
mtc1	$r_{\bar{e}}, f_d$	Copia los 64 bits del registro entero $r_{\bar{e}}$ al registro $f_d$ de punto flotante
mfc1	$r_d, f_{\bar{e}}$	Copia los 64 bits del registro $f_{\bar{e}}$ de punto flotante al registro $r_d$ entero
cvt.d.l	$f_d, f_{\bar{e}}$	Convierte a punto flotante el valor entero copiado al registro $f_{\bar{e}}$ , dejándolo en $f_d$
cvt.l.d	$f_d, f_{\bar{e}}$	Convierte a entero el valor en punto flotante contenido en $f_{\bar{e}}$ , dejándolo en $f_d$

Instrucciones Aritméticas		
dadd	$r_d, r_{\bar{e}}, r_g$	Suma $r_{\bar{e}}$ con $r_g$ , dejando el resultado en $r_d$ (valores con signo)
daddi	$r_d, r_{\bar{e}}, N$	Suma $r_{\bar{e}}$ con el valor inmediato $N$ , dejando el resultado en $r_d$ (valores con signo)
daddu	$r_d, r_{\bar{e}}, r_g$	Suma $r_{\bar{e}}$ con $r_g$ , dejando el resultado en $r_d$ (valores sin signo)
daddui	$r_d, r_{\bar{e}}, N$	Suma $r_{\bar{e}}$ con el valor inmediato $N$ , dejando el resultado en $r_d$ (valores con signo)
add.d	$f_d, f_{\bar{e}}, f_g$	Suma $f_{\bar{e}}$ con $f_g$ , dejando el resultado en $f_d$ (en punto flotante)
dsub	$r_d, r_{\bar{e}}, r_g$	Resta $r_g$ a $r_{\bar{e}}$ , dejando el resultado en $r_d$ (valores con signo)
dsubu	$r_d, r_{\bar{e}}, r_g$	Resta $r_g$ a $r_{\bar{e}}$ , dejando el resultado en $r_d$ (valores sin signo)
sub.d	$f_d, f_{\bar{e}}, f_g$	Resta $f_g$ a $f_{\bar{e}}$ , dejando el resultado en $f_d$ (en punto flotante)
dmul	$r_d, r_{\bar{e}}, r_g$	Mutliplica $r_{\bar{e}}$ con $r_g$ , dejando el resultado en $r_d$ (valores con signo)
dmulu	$r_d, r_{\bar{e}}, r_g$	Mutliplica $r_{\bar{e}}$ con $r_g$ , dejando el resultado en $r_d$ (valores sin signo)
mul.d	$f_d, f_{\bar{e}}, f_g$	Multiplica $f_{\bar{e}}$ con $f_g$ , dejando el resultado en $f_d$ (en punto flotante)
ddiv	$r_d, r_{\bar{e}}, r_g$	Divide $r_{\bar{e}}$ por $r_g$ , dejando el resultado en $r_d$ (valores con signo)
ddivu	$r_d, r_{\bar{e}}, r_g$	Divide $r_{\bar{e}}$ por $r_g$ , dejando el resultado en $r_d$ (valores sin signo)
div.d	$f_d, f_{\bar{e}}, f_g$	Divide $f_{\bar{e}}$ por $f_g$ , dejando el resultado en $f_d$ (en punto flotante)
slt	$r_d, r_{\bar{e}}, r_g$	Compara $r_{\bar{e}}$ con $r_g$ , dejando $r_d=1$ si $r_{\bar{e}}$ es menor que $r_g$ (valores con signo)
slti	$r_d, r_{\bar{e}}, N$	Compara $r_{\bar{e}}$ con el valor inmediato $N$ , dejando $r_d=1$ si $r_{\bar{e}}$ es menor que $N$ (valores signo)
c.lt.d	$f_d, f_{\bar{e}}$	Compara $f_d$ con $f_{\bar{e}}$ , dejando flag $FP=1$ si $f_d$ es menor que $f_{\bar{e}}$ (en punto flotante)
c.le.d	$f_d, f_{\bar{e}}$	Compara $f_d$ con $f_{\bar{e}}$ , dejando flag $FP=1$ si $f_d$ es menor o igual que $f_{\bar{e}}$ (en punto flotante)
c.eq.d	$f_d, f_{\bar{e}}$	Compara $f_d$ con $f_{\bar{e}}$ , dejando flag $FP=1$ si $f_d$ es igual que $f_{\bar{e}}$ (en punto flotante)

Instrucciones Lógicas		
and	$r_d, r_{\bar{e}}, r_g$	Realiza un AND entre $r_{\bar{e}}$ y $r_g$ (bit a bit), dejando el resultado en $r_d$
andi	$r_d, r_{\bar{e}}, N$	Realiza un AND entre $r_{\bar{e}}$ y el valor inmediato $N$ (bit a bit), dejando el resultado en $r_d$
or	$r_d, r_{\bar{e}}, r_g$	Realiza un OR entre $r_{\bar{e}}$ y $r_g$ (bit a bit), dejando el resultado en $r_d$
ori	$r_d, r_{\bar{e}}, N$	Realiza un OR entre $r_{\bar{e}}$ y el valor inmediato $N$ (bit a bit), dejando el resultado en $r_d$
xor	$r_d, r_{\bar{e}}, r_g$	Realiza un XOR entre $r_{\bar{e}}$ y $r_g$ (bit a bit), dejando el resultado en $r_d$
xori	$r_d, r_{\bar{e}}, N$	Realiza un XOR entre $r_{\bar{e}}$ y el valor inmediato $N$ (bit a bit), dejando el resultado en $r_d$

Instrucciones de desplazamiento de bits		
dsll	$r_d, r_{\bar{e}}, N$	Desplaza a izquierda $N$ veces los bits del registro $r_{\bar{e}}$ , dejando el resultado en $r_d$
dsllv	$r_d, r_{\bar{e}}, r_N$	Desplaza a izquierda $r_N$ veces los bits del registro $r_{\bar{e}}$ , dejando el resultado en $r_d$
dsrl	$r_d, r_{\bar{e}}, N$	Desplaza a derecha $N$ veces los bits del registro $r_{\bar{e}}$ , dejando el resultado en $r_d$
dsrlv	$r_d, r_{\bar{e}}, r_N$	Desplaza a derecha $r_N$ veces los bits del registro $r_{\bar{e}}$ , dejando el resultado en $r_d$
dsra	$r_d, r_{\bar{e}}, N$	Igual que <b>dsrl</b> pero mantiene el signo del valor desplazado
dsrav	$r_d, r_{\bar{e}}, r_N$	Igual que <b>dsrlv</b> pero mantiene el signo del valor desplazado

Instrucciones de Transferencia de Control		
j	offN	Salta a la dirección rotulada offN
jal	offN	Salta a la dirección rotulada offN y copia en $r_{31}$ la dirección de retorno
jr	$r_d$	Salta a la dirección contenida en el registro $r_d$
beq	$r_d, r_{\bar{e}}, \text{offN}$	Si $r_d$ es igual a $r_{\bar{e}}$ , salta a la dirección rotulada offN
bne	$r_d, r_{\bar{e}}, \text{offN}$	Si $r_d$ no es igual a $r_{\bar{e}}$ , salta a la dirección rotulada offN
beqz	$r_d, \text{offN}$	Si $r_d$ es igual a 0, salta a la dirección rotulada offN
bnez	$r_d, \text{offN}$	Si $r_d$ no es igual a 0, salta a la dirección rotulada offN
bclf	offN	Salta a la dirección rotulada offN si flag $FP=0$ (ó false) (en punto flotante)
bclt	offN	Salta a la dirección rotulada offN si flag $FP=1$ (ó true) (en punto flotante)

Instrucciones de Control	
nop	Operación nula
halt	Detiene el simulador