

# Set de instrucciones del simulador WinMIPS64

Instrucciones de transferencias de datos		
Instrucción	Comentario	
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	$r_f, \text{Inm}(r_i)$	Guarda los 8 bits menos significativos de $r_f$ 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 dirección $(\text{Inm}+r_i)$ (con extensión del signo)
lhu	$r_d, \text{Inm}(r_i)$	Copia en $r_d$ un half-word (16 bits) desde la dirección $(\text{Inm}+r_i)$ (sin extensión del signo)
sh	$r_f, \text{Inm}(r_i)$	Guarda los 16 bits menos significativos de $r_f$ en la dirección $(\text{Inm}+r_i)$
lw	$r_d, \text{Inm}(r_i)$	Copia en $r_d$ un word (32 bits) desde la dirección $(\text{Inm}+r_i)$ (con extensión del signo)
lwu	$r_d, \text{Inm}(r_i)$	Copia en $r_d$ un word (32 bits) desde la dirección $(\text{Inm}+r_i)$ (sin extensión del signo)
sw	$r_f, \text{Inm}(r_i)$	Guarda los 32 bits menos significativos de $r_f$ en 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	$r_f, \text{Inm}(r_i)$	Guarda $r_f$ en 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	$f_f, \text{Inm}(r_i)$	Guarda $f_f$ a partir de la dirección $(\text{Inm}+r_i)$
mov.d	$f_d, f_f$	Copia el valor del registro $f_f$ al registro $f_d$
mtc1	$r_f, f_d$	Copia los 64 bits del registro entero $r_f$ al registro $f_d$ de punto flotante
mfc1	$r_d, f_f$	Copia los 64 bits del registro $f_f$ de punto flotante al registro $r_d$ entero
cvt.d.l	$f_d, f_f$	Convierte a punto flotante el valor entero copiado al registro $f_f$ , dejándolo en $f_d$
cvt.l.d	$f_d, f_f$	Convierte a entero el valor en punto flotante contenido en $f_f$ , dejándolo en $f_d$

Instrucciones lógicas		
Instrucción	Comentario	
and	$r_d, r_f, r_g$	Realiza un AND entre $r_f$ y $r_g$ (bit a bit), dejando el resultado en $r_d$
andi	$r_d, r_f, N$	Realiza un AND entre $r_f$ y el valor inmediato $N$ (bit a bit), dejando el resultado en $r_d$
or	$r_d, r_f, r_g$	Realiza un OR entre $r_f$ y $r_g$ (bit a bit), dejando el resultado en $r_d$
ori	$r_d, r_f, N$	Realiza un OR entre $r_f$ y el valor inmediato $N$ (bit a bit), dejando el resultado en $r_d$
xor	$r_d, r_f, r_g$	Realiza un XOR entre $r_f$ y $r_g$ (bit a bit), dejando el resultado en $r_d$
xori	$r_d, r_f, N$	Realiza un XOR entre $r_f$ y el valor inmediato $N$ (bit a bit), dejando el resultado en $r_d$

Instrucciones de desplazamiento de bits		
Instrucción	Comentario	
dsl	$r_d, r_f, N$	Desplaza a la izquierda $N$ veces los bits del registro $r_f$ , dejando el resultado en $r_d$
dslv	$r_d, r_f, r_n$	Desplaza a la izquierda $r_n$ veces los bits del registro $r_f$ , dejando el resultado en $r_d$
sr	$r_d, r_f, N$	Desplaza a la derecha $N$ veces los bits del registro $r_f$ , dejando el resultado en $r_d$
sr	$r_d, r_f, r_n$	Desplaza a la derecha $r_n$ veces los bits del registro $r_f$ , dejando el resultado en $r_d$
dsra	$r_d, r_f, N$	Igual que <b>dsrl</b> pero mantiene el signo del valor desplazado
dsrav	$r_d, r_f, r_n$	Igual que <b>dsrlv</b> pero mantiene el signo del valor desplazado

Instrucciones aritméticas		
Instrucción	Comentario	
dadd	$r_d, r_f, r_g$	Suma $r_f$ con $r_g$ , dejando el resultado en $r_d$ (valores con signo)
daddi	$r_d, r_f, N$	Suma $r_f$ con el valor inmediato $N$ , dejando el resultado en $r_d$ (valores con signo)
daddu	$r_d, r_f, r_g$	Suma $r_f$ con $r_g$ , dejando el resultado en $r_d$ (valores sin signo)
daddui	$r_d, r_f, N$	Suma $r_f$ con el valor inmediato $N$ , dejando el resultado en $r_d$ (valores sin signo)
add.d	$f_d, f_f, f_g$	Suma $f_f$ con $f_g$ , dejando el resultado en $f_d$ (en punto flotante)
dsub	$r_d, r_f, r_g$	Resta $r_f$ a $r_g$ , dejando el resultado en $r_d$ (valores con signo)
dsubu	$r_d, r_f, r_g$	Resta $r_f$ a $r_g$ , dejando el resultado en $r_d$ (valores sin signo)
sub.d	$f_d, f_f, f_g$	Resta $f_f$ a $f_g$ , dejando el resultado en $f_d$ (en punto flotante)
dmul	$r_d, r_f, r_g$	Multiplica $r_f$ con $r_g$ , dejando el resultado en $r_d$ (valores con signo)
dmulu	$r_d, r_f, r_g$	Multiplica $r_f$ con $r_g$ , dejando el resultado en $r_d$ (valores sin signo)
mul.d	$f_d, f_f, f_g$	Multiplica $f_f$ con $f_g$ , dejando el resultado en $f_d$ (en punto flotante)
ddiv	$r_d, r_f, r_g$	Divide $r_f$ por $r_g$ , dejando el resultado en $r_d$ (valores con signo)
ddivu	$r_d, r_f, r_g$	Divide $r_f$ por $r_g$ , dejando el resultado en $r_d$ (valores sin signo)
div.d	$f_d, f_f, f_g$	Divide $f_f$ por $f_g$ , dejando el resultado en $f_d$ (en punto flotante)
slt	$r_d, r_f, r_g$	Compara $r_f$ con $r_g$ , dejando $r_d = 1$ si $r_f < r_g$ (valores con signo)
slti	$r_d, r_f, N$	Compara $r_f$ con el valor inmediato $N$ , dejando $r_d = 1$ si $r_f < N$ (valores con signo)
c.lt.d	$f_d, f_f$	Compara $f_d$ con $f_f$ , dejando la <i>flag</i> $FP = 1$ si $f_d < f_f$ ; si no, $FP = 0$ (en punto flotante)
c.le.d	$f_d, f_f$	Compara $f_d$ con $f_f$ , dejando la <i>flag</i> $FP = 1$ si $f_d \leq f_f$ ; si no, $FP = 0$ (en punto flotante)
c.eq.d	$f_d, f_f$	Compara $f_d$ con $f_f$ , dejando la <i>flag</i> $FP = 1$ si $f_d = f_f$ ; si no, $FP = 0$ (en punto flotante)

Instrucciones de transferencia de control		
Instrucción	Comentario	
j	offN	Salta a la dirección rotulada offN
jal	offN	Salta a la dirección rotulada offN y copia en r31 la dirección de retorno
jr	$r_d$	Salta a la dirección contenida en el registro $r_d$
beq	$r_d, r_f, \text{offN}$	Si $r_d = r_f$ , salta a la dirección rotulada offN
bne	$r_d, r_f, \text{offN}$	Si $r_d \neq r_f$ , salta a la dirección rotulada offN
beqz	$r_d, \text{offN}$	Si $r_d = 0$ , salta a la dirección rotulada offN
bnez	$r_d, \text{offN}$	Si $r_d \neq 0$ , salta a la dirección rotulada offN
bc1f	offN	Salta a la dirección rotulada offN si la <i>flag</i> $FP = 0$ (en punto flotante)
bc1t	offN	Salta a la dirección rotulada offN si la <i>flag</i> $FP = 1$ (en punto flotante)

Instrucciones de control	
Instrucción	Comentario
nop	Operación nula
halt	Detiene el simulador