NEMÓNICO	OPERANDOS	DESCRIPCIÓN	31 26	25 21	20 16	15 10	9 5	4
and	rD, rS1, rS2	rD ← rS1 *op* rS2	111101	D	S1	010000	00000	
and.c	rD, rS1, rS2					1		
and	rD, rS1, IMM16	rD ← rS1 *op* IMM16 (los otros	010000			IMM16		
and.u	rD, rS1, IMM16	16 bits sin cambiar)	1					
xor	rD, rS1, rS2					01010		
xor	rD, rS1, IMM16		01010					
or	rD, rS1, rS2					01011		
or	rD, rS1, IMM16		01011					
mask	rD, rS1, IMM16	rD ← rS1 and IMM16 (los otros 16 a 0)	01001					
add	rD, rS1, rS2	rD ← rS1 ± rS2	111101	D	S1	011100	00 000	
add.ci	rD, rS1, rS2						10	
add.co	rD, rS1, rS2						01	
add.cio	rD, rS1, rS2						11	
addu	rD, rS1, rS2					011000		
sub	rD, rS1, rS2					011101		
subu	rD, rS1, rS2					011001		
add	rD, rS1, SIMM16	rD ← rS1 ± IMM16	011100			IMM16 ó SIMM1		
addu	rD, rS1, IMM16	u indica sin signo	011000					
sub	rD, rS1, SIMM16		011101					
subu	rD, rS1, IMM16		011001					
muls	rD, rS1, rS2	rD ← rS1 * rS2	111101	D	S1	011011	10000	
divs	rD, rS1, rS2	rD ← rS1 / rS2				011110	00000	
divs	rD, rS1, SIMM16		011110			SIMM16		
mulu	rD, rS1, rS2	u indica sin signo	111101			011011	00000	
mulu.d	rD, rS1, rS2						1	
mulu	rD, rS1, IMM16		011011			IMM16		
divu	rD, rS1, rS2		111101			011010	00000	
divu.d	rD, rS1, rS2						1	
divu	rD, rS1, IMM16		011010			IMM16		
clr	rD, rS1, W5<O5>	rD ← rS1 con bits O5 a O5 + W5 –1 a 0	111100	D	S1	100000	W5	
clr	rD, rS1, rS2		111101			100000	00000	
set	Se opera	rD ← rS1 con bits O5 a O5 + W5 – 1 a 1				100010		
ext	con los dos	rD ← los bits O5 a				100100		
	tipos de	O5 + W5 – 1 de rS1 (con ext. signo)						
extu	formatos de	instr. ext sin extensión de signo				100110		
mak	instrucción	bits O5 a O5 + W5 – 1 de rD ← bits				101000		
	utilizados	0 a W5 de rS1. resto a 0						
rot	en clr	rD ← O5 rotaciones dcha. de rS1				101010	00000	

NEMÓNICO	OPERANDOS	DESCRIPCIÓN	31 26	25 21	20 16	15 11	10 5	4
fadd.xxx	rD, rS1, rS2	rD ← rS1 *op* rS2	100001	D	S1	00101	T1T2TD	
fsub.xxx	rD, rS1, rS2	T1, T2, TD son:				00110		
fmul.xxx	rD, rS1, rS2	00: Simple precisión				00000		
fdiv.xxx	rD, rS1, rS2	01: Doble precisión				01110		
fcvt.x$\overline{x}$	rD, rS2	Cambio de precisión			00000	00001	00T2TD	
flt.xs	rD, rS2	Coma flotante ← entero con signo				00100	0000TD	
int.sx	rD, rS2	entero con signo ← Coma flotante (s ó d)				01001	00T200	
bb0	B5, rS1, D16	PC ← PC + 4*D16 si $RS1_{B5} = 0$	110100	B5	S1	D16		
bb0.n	B5, rS1, D16		110101					
bb1	B5, rS1, D16	PC ← PC + 4*D16 si $RS1_{B5} = 1$	11011					
br	D26	PC ← PC + 4*D26 y	110000	D26				
br.n	D26	r1 ← PC + 4 si es bsr	1					
bsr	D26		11001					
jmp	(rS2)	PC ← rS2 con $rS2_0 = rS2_1 = 0$ y	111101	00000	00000	11000	000000	
jmp.n	(rS2)	r1 ← PC + 4 si es jsr					1	
jsr	(rS2)					11001		
ld.b	rD, rS1, SI16	rD ← (rS1 + SI16)	000111	D	S1	SI16		
ld.h	rD, rS1, SI16		10					
ld	rD, rS1, SI16		01					
ld.d	rD, rS1, SI16		00					
st.y	rD, rS1, SI16	(rS1 + SI16) ← rD	0010					
ld.b	rD, rS1, rS2	rD ← (rS1 + rS2)	111100			00011	100000	
ld.h	rD, rS1, rS2					1	0	
ld	rD, rS1, rS2					0	1	
ld.d	rD, rS1, rS2					0	0	
st.y	rD, rS1, rS2	(rS1 + rS2) ← rD				0010	00000	
st.y.wt	rD, rS1, rS2	(rS1 + rS2) ← rD					10000	
ld.bu	rD, rS1, SI16		000011			SI16		
ld.hu	rD, rS1, SI16		0					
ld.bu	rD, rS1, rS2		111101			00001	100000	
ld.hu	rD, rS1, rS2						0	
ldcr	rD	rD ← reg. control	100000		00000	00000	000000	0
stcr	rD	rD → reg. control	100000		00000	10000	000000	0
xmem	rD, rS1, rS2	rD ↔ (rS1 + rS2)	111101		S1	00000	000000	
cmp		VÉASE EL MANUAL						
fcmp		VÉASE EL MANUAL						