

Tipo de Instrucción	Estándar	MC88110	MC68000
Transferencia	MOVE .R1,.R2 .R2 ← .R1	or r2,r1,r1 r2 ← r1	MOVE.L D1,D2 D2 ← D1
	ST .R1,/var mem(var) ← .R1	LEA(r20,var) r20 ← dir(var) st r1,r20,r0 mem(r20) ← r1	MOVE.L D1,VAR mem(var) ← D1
	LD .R1,/var .R1 ← mem(var)	LEA(r20,var) r20 ← dir(var) ld r1,r20,r0 r1 ← mem(r20)	MOVE.L VAR,D1 D1 ← mem(var)
	MOVE /V1,/V2 mem(V2) ← mem(V1)	LEA(r20,V1) r20 ← dir(V1) ld.hu r20,r20,r0 r20 ← mem(r20) LEA(r21,V2) r21 ← dir(V2) st.h r20,r21,r0 mem(r21) ← r20	MOVE.W V1,V2 mem(V2) ← mem(V1)
	MOVE.B .R1,/V2 mem(V2) ← R1 ₇₋₀	LEA(r20,V1) r20 ← dir(V2) st.b r1,r20,r0 mem(r20) ← r1 ₇₋₀	MOVE.B D1,V2 mem(V2) ← D1 ₇₋₀
	PUSH .R2 mem(--SP) ← R2	PUSH(r2) r30 ← r30-4 mem(r30) ← r2	MOVE.L D1,-(A7) A7 ← A7-4 mem(A7) ← D1
	POP .R2 R2 ← mem(SP++)	POP(r2) r2 ← mem(r30) r30 ← r30+4	MOVE.L (A7)+,D1 D1 ← mem(A7) A7 ← A7+4
Aritméticas	ADD .R1,.R2 R1 ← R1 + R2	add r1,r1,r2 r1 ← r1+r2	ADD D2,D1 D1 ← D1+D2
	ADD .R1,#4 R1 ← R1 + 4	add r1,r1,4 r1 ← r1+4	ADD #4,D1 D1 ← D1+4
	ADD [.R1],[.R2] mem(R1) ← mem(R1) + mem(R2)	ld r7,r1,r0 r7 ← mem(r1) ld r8,r2,r0 r8 ← mem(r2) add r7,r7,r8 r7 ← r7+r8 st r7,r2,r0 mem(r2) ← r7	ADD (A2),(A1) mem(A1) ← mem(A1)+mem(A2)
Bifurcación	BR /ETIQ PC ← dir(ETIQ)	br ETIQ PC ← dir(ETIQ)	BRA ETIQ PC ← dir(ETIQ)
	BR [.R1] PC ← R1	jmp (r1) PC ← r1	JMP (A1) PC ← A1
	CALL /ETIQ Salva dir. retorno	bsr ETIQ r1 ← PC	BSR ETIQ A7 ← A7 - 4 mem(A7) ← PC PC ← dir(ETIQ)
	RET Recupera dir. retorno	jmp (r1) PC ← r1	RTS PC ← mem(A7) A7 ← A7 + 4
	BZ /ETIQ Si Z = 1 PC ← dir(ETIQ)	—————	BEQ ETIQ ^a Si Z = 1 PC ← dir(ETIQ)

^aTodas las condiciones de salto están especificadas en la pág. 10 del manual

Tipo de Instrucción	Estándar	MC88110	MC68000
Bit	BSET.I #3,.R1 $R1_3 \leftarrow 1$	set r1,r1,1<3> $r1_3 \leftarrow 1$	BSET #3,D1 $D1_3 \leftarrow 1$
	BCLR.I #5,.R1 $R1_5 \leftarrow 0$	clr r1,r1,1<5> $r1_5 \leftarrow 0$	BCLR #5,D1 $D1_5 \leftarrow 0$
	BTST.I #0,.R1 $Z \leftarrow \overline{R1_0}$	—————	BTST #0,D1 $Z \leftarrow \overline{D1_0}$
Comparación y bifurcación	CMP .R1,.R2 Pregunta si dos regs. no son iguales	cmp r5,r1,r2 Pregunta si dos regs. no son iguales	CMP D1,D2 Pregunta si dos regs. no son iguales
	BNZ ETIQ	bb1 ne,r5,eti	BNE ETIQ
	CMP .R1,#0 Pregunta si R1 es cero	cmp r5,r1,0 Pregunta si r1 es cero	CMP #0,D1 Pregunta si D1 es cero
	BZ ETIQ	bb1 eq,r5,eti	BEQ ETIQ

Pseudoinstrucciones

Estándar	MC88110	MC68000
ORG Define el origen del código	org	org
EQU Define una constante	—	EQU
RES Reserva espacio en memoria	res 200	DS.B 200
DATA Inicializa un espacio en memoria	data	DC
	data 43, 44, 0x45	DC 43, 44, \$45
	data "arquitectura"	DC.B 'arquitectura'