

MOVIMIENTO DE DATOS

Mnemónico	Función	Modos permitidos	Tamaño del operando	Función
EXG	Intercambiar registros	Rm,Rn	L	Rm ↔ Rn
LEA	Cargar una dirección efectiva	<cea>,An	L	<ea> → An
LINK	Asignar espacio	An.#<d16>	Sin tamaño	An → —(SP); SP → An; SP + d16 → SP
MOVE	Mover datos	<ea>,<adea>	L,W,{B}	Fnte → Dest
MOVEA	Mover una dirección	<ea>,An	L,W	Fnte → Dest
MOVEM	Mover varios registros	<lista,reg>,<acea>+,<cea>+,<lista.reg>	L,W [+ —(An)], L,W [+ +(An)]	Reg.list → Dest Fnte → Reg.list
MOVEP	Mover datos desde/hacia los periféricos	Dn.d(An) o d(An).Dn	L,W	Fnte → Dest
MOVEQ	Mover rápidamente	#<d8>,Dn	L (extensión de signo a 32 bits)	d8 → Dest
PEA	Obtener una dirección efectiva de control	<cea>	L	<ea> → —(SP)
SWAP	Intercambia registros	Dn	W	Dn[31:16] ↔ Dn[15:0]
UNLK	Desasignar	An	Sin tamaño	An → SP; (SP) + → An

ARITMETICA ENTERA

ADD	Sumar en binario	<ea>,Dn o Dn,<amea>	L,W,B	Fnte + Dest → Dest
ADDA	Sumar direcciones	<ea>,An	L,W	Fnte + Dest → Dest
ADDI	Suma inmediata	#<datos>,<adea>	L,W,B	dato + Dest → Dest
ADDQ	Suma rápida	#<d3>,<aea>	L,W,{B}	d3 + Dest → Dest
ADDX	Suma extendida	Dm,Dn o —(Am),—(An)	L,W,B	Fnte + Dest + X → Dest
CLR	Poner a 0 un operando	<adea>	L,W,B	0 → Dest
CMP	Comparar	<ea>,Dn	L,W,{B}	Dest — Fnte; activa CCR
CMPA	Comparar direcciones	<ea>,An	L,W	Dest — Fnte; activa CCR
CMPI	Comparación inmediata	#<dato>,<adea>	L,W,B	Dest — dato; activa CCR
CMPM	Comparar con la memoria	(Am) + , (An) +	L,W,B	Dest — Fnte; activa CCR
DIVS/DIVU	División con y sin signo	<dea>,Dn	W	Dest/Fnte → Dest Cociente = inferior; Rem = palabra superior
EXT	Extensión de signo	Dn	L,W	Dn/sign — extendido → Dn
MULS/MULU	Multiplicar con y sin signo	<dea>,Dn	W	Fnte × Dest → Dest
NEG	Negar	<adea>	L,W,B	0 — Dest → Dest
NEGX	Negar en modo extendido	<adea>	L,W,B	0 — Dest — X → Dest
SUB	Resta binaria	Igual que ADD	L,W,B	Dest — Fnte → Dest
SUBA	Restar direcciones	Igual que ADDA	L,W	Dest — Fnte → Dest
SUBI	Resta inmediata	Igual que ADDI	L,W,B	Dest — dato → Dest
SUBQ	Resta rápida	Igual que ADDQ	L,W,{B}	Dest — d3 → Dest
SUBX	Resta extendida	Igual que ADDX	L,W,B	Dest — Fnte — X → Dest
TAS	Comprueba y pone a 1 un operando	<adea>	B	Test Dest → CCR; 1 → Dest[bit 7]
TST	Comprobar un operando	<adea>	L,W,B	Test Dest → CCR

OPERACIONES LOGICAS

AND	Y lógico	<dea>,Dn o Dn,<amea>	L,W,B	Fnte ∧ Dest → Dest
ANDI	Y inmediato	#<datos>,<adea>	L,W,B	dato ∧ Dest → Dest
EOR	OR exclusivo	Dn.<adea>	L,W,B	Fnte <OR exclusivo> Dest → Dest
EORI	OR exclusivo inmediato	#<datos>,<adea>	L,W,B	dato <OR exclusivo> Dest → Dest
NOT	Complemento lógico	<adea>	L,W,B	~Dest → Dest
OR	OR lógico inclusivo	Igual que AND	L,W,B	Fnte <OR> Dest → Dest
ORI	OR inclusivo inmediato	Igual que ANDI	L,W,B	dato <OR> Dest → Dest

ROTACIONES Y DESPLAZAMIENTOS

Mnemónico	Función	Modos permitidos	Tamaño del operando	Función
ASL/ASR	Desplazamientos aritméticos	Dm,Dn o #<d3>,Dn	L,W,B (Dm mod 64)	Dest Arit shift <cont> → Dest
ASL/ASR	Desplazamientos aritméticos	<amea>	W (contador de desplazamiento = 1)	Dest Arit shift b 1 vez → Dest
LSL/LSR	Desplazamientos lógicos	Igual que ASL/ASR		
ROL/ROR	Rotar a la izquierda o a la derecha	Igual que ASL/ASR		
ROXL/ROXR	Rotación con extensión	Igual que ASL/ASR		

MANEJO DE BITS

BCHG	Comprobar/Cambiar un bit	Dm,Dn o #<d5>,Dn	L (Dm mod 32)	~Dest[bit] → flag Z ~Dest[bit] → Dest[bit]
BCHG	Comprobar/Cambiar un bit	Dm,<amea> o #<d3>,<amea>		Igual
BCLR	Probar/Poner a cero un bit	Igual que BCHG		~Dest[bit] → flag Z:0 → Dest[bit]
BSET	Comprobar/Poner a 1 un bit	Igual que BCHG		~Dest[bit] → flag Z:1 → Dest[bit]
BTST	Comprobar un bit	Dm,Dn o #<d15>,Dn	L (Dm mod 32)	~Dest[bit] → flag Z
BTST	Comprobar un bit	Dm,<mea> o #<d3>,<mea>	B (Dm mod 8)	

OPERACIONES EN BCD

ABCD	Sumar en decimal	Dm,Dn o —(Am),—(An)	B	Fnte(10) + Dest(10) + X → Dest(10)
NBCD	Negar en decimal	<adea>	B	0 — Dest(10) — X → Dest(10)
SBCD	Resta decimal	Igual que ABCD	B	Fnte(10) — Dest(10) — X → Dest(10)

OPERACIONES DE CONTROL DE PROGRAMA

Condicionales:

Bcc	Condición de bifurcación	<etiqueta>	Desplazamiento de 16 bits	IF cc true PC + disp → PC
Bcc.S	Condición de desplazamiento corto	<etiqueta>	Desplazamiento de 8 bits	Igual
DBcc	Salto condicional decimal	Dm,<etiqueta>	Desplazamiento de 16 bits	IF cc false THEN Dm — 1 → Dm IF Dm <> — 1 THEN PC + disp → PC ELSE PC + 2 → PC (siguiente instrucción)
Scc	Poner a 1 condicionalmente	<adea>	B	IF cc true \$FF → Dest ELSE \$00 → Dest

Incondicionales:

BRA	Salto incondicional	<etiqueta>	Desplazamiento de 8 ó 16 bits	PC + desp → PC
BSR	Saltar a una subrutina	Igual que BRA		PC → —(SP); PC + desp → PC
JMP	Salto	<cea>	Sin tamaño	Dest → PC
JSR	Salto a subrutina	<cea>	Sin tamaño	PC → —(SP); Dest → PC

Mnemónico	Función	Modos permitidos	Tamaño del operando	Función
Retornos:				
RTD ^^	Retornar/Desasignar	#<d16>	Sin tamaño (extensión de signo a 32 bits)	(SP) + → PC; SP + desp → SP
RTR	Retornar y restaurar el CCR	Sin operando	Sin tamaño	(SP) + → CCR; (SP) + → PC
RTS	Retornar de una subrutina	Sin operando	Sin tamaño	(SP) + → PC

OPERACIONES DE CONTROL DEL SISTEMA

ANDI → SR*	Suma directa con el SR	#<d16>,SR	W	IF S = 1 Fnte ∧ SR → SR ELSE TRAP
EORI → SR*	Registro de estado de EORI	#<d16>,SR	W	IF S = 1 Fnte <OR exclusivo> SR → SR ELSE TRAP
MOVE → SR*	Mover al SR	<adea>,SR	W (* todos los modelos)	S = 1 SR → Dest ELSE TRAP
MOVE ← SR ^^*		SR,<adea>	W (* sólo para el 68010)	IF S = 1 Fnte → SR ELSE TRAP
MOVE → USP*	Mover al USP	USP,An o An,USP	L	IF S = 1 Fnte → Dest ELSE TRAP
MOVEC ^^*	Mover a los registros de control	Rc,Rn o Rn,Rc	L	IF S = 1 Fnte → Dest ELSE TRAP
MOVES ^^*	Mover espacios de direcciones	RN,DFC<amea> o SFC <amea>,Rn	L,W,B	IF S = 1 Fnte → Dest ELSE TRAP
ORI → SR*	ORI con el registro de estado	Igual que ANDI → SR		IF S = 1 Fnte <OR> SR → SR ELSE TR
RESET*	Reinicia un dispositivo externo	Sin operandos	Sin tamaño	IF S = 1 activa RESET ELSE TRAP
RTE*	Retornar de una excepción	Sin operando	Sin tamaño	IF S = 1 (SP) + → SR; (SP) + → PC ELSE TRAP
RTE ^^*		Sin operando	Sin tamaño	IF S = 1 (SP) + → SR; (SP) + → PC IF (SP) + = long THEN recuperación completa ELSE TRAP
STOP*	Carga un SR/STOP	#<d16>	Sin tamaño	IF S = 1 d16 → SR; PC + 4 → PC; Pausa hasta excepción ELSE TRAP

Generadores de Traps:

CHK	Comprobar los límites de los registros	<dea>,Dn	W	IF Dn <0 or Dn>Dest THEN TRAP
ILLEGAL	No permitido	Sin operando		PC → —(SSP); SR → —(SSP); Vector #4 → PC
TRAP	Trap	#<d4>	Sin tamaño	PC → —(SSP); SR → —(SSP); Vector #d4 → PC
TRAPV	Trap cuando se produce un <i>overflow</i>	Sin operando	Sin tamaño	IF V = 1 THEN TRAP

Registro de código de condición:

ANDI → CCR	Suma inmediata con el CCR	#<d8>,CCR	B	d8 ∧ CCR → CCR
EORI → CCR	Códigos de condiciones de EORI	#<d8>,CCR	B	d8 <OR exclusivo> CCR → CCR
MOVE ← CCR	Mover desde el CCR	CCR,<adea>	W (palabra más baja)	CCR → Dest
MOVE → CCR	Mover al CCR	<adea>,CCR	W (palabra más baja)	Fnte → CCR
MOVE ← SR**	Desde SR	SR,<adea>	W	SR → Dest
ORI → CCR	ORI con los códigos de condición	Igual que ANDI → CCR		d8 <OR> CCR → CCR

VARIOS

NOP	No efectúa operación alguna	Sin operandos	Sin tamaño	PC + 2 → PC
-----	-----------------------------	---------------	------------	-------------

Modos de direccionamiento

- <ea> = Cualquier dirección efectiva
 <rea> = Dirección efectiva de un registro
 <dea> = Dirección efectiva de los datos
 <mea> = Dirección efectiva de la memoria
 <cea> = Dirección efectiva de control
 <aea> = Dirección efectiva alterable (datos o memoria)
 <adea> = Dirección efectiva alterable de datos
 <amea> = Dirección efectiva alterable de la memoria
 <aceu> = Dirección efectiva alterable de control

Modo	ea	rea	dea	mea	cea	aea	adea	amea	aceu	
Dn	*	*	*			*	*			
An	*	*				*				
(An)	*		*	*	*	*	*	*	*	
(An) +	*		*	*		*	*	*		
-(An)	*		*	*		*	*	*		
d(An)	*		*	*	*	*	*	*	*	
d(An,Xi)	*		*	*	*	*	*	*	*	
Abs.W	*		*	*	*	*	*	*	*	
Abs.L	*		*	*	*	*	*	*	*	
d(PC)	*		*	*	*					
d(PC,Xi)	*		*	*	*					
Inmed	*		*	*						
bd(An,Xi)	*		*	*	*	*	*	*	*	MC68020
bd(PC,Xi)	*		*	*	*					MC68020
[bd,An],Xi,od	*		*	*	*	*	*	*	*	MC68020
[bd,An,Xi],od	*		*	*	*	*	*	*	*	MC68020
[bd,PC],Xi,od	*		*	*	*					MC68020
[bd,PC,Xi],od	*		*	*	*					MC68020

Símbolos

{B}	Cantidad de datos Byte no permitidos en un operando An.
#<dato>	Hasta 32 bits de datos inmediatos.
#<dn>	n bits de datos inmediatos.
*	Instrucción con privilegio del MC68000.
**	Instrucción con privilegio del MC68010/20.
^^	Instrucción sólo del MC68010.
Rc	Registro de control (SFC, DFC, USP o VBR).
SP	Puntero de pila (A7).
SSP	Puntero de pila supervisor (A7').
Fnte	Operando fuente.
Dest	Operando destino.
<>	Distinto que.
^	AND (Y lógico).
~	NOT (NO lógico).
→	Sustituye.
↔	Se intercambia con.
[m:n]	Rango de bits.

Complemento al formulario:

Multiplicación con signo: Destino = Destino*Origen.

MULS	.W	<dea>,Dn	$16 \times 16 \Rightarrow 32$
	L	<dea>,Dn	$32 \times 32 \Rightarrow 32$
	L	<dea>,Dh:Dl	$32 \times 32 \Rightarrow 64$

Multiplicación sin signo: MULU.

División con signo: Destino = Destino/Origen.

DIVS	.W	<dea>,Dn	$32/16 \Rightarrow 16r:16q$
	L	<dea>,Dq	$32 \times 32 \Rightarrow 32$
	L	<dea>,Dr:Dq	$64/32 \Rightarrow 32r:32q$
DIVSL	L	<dea>,Dr:Dq	$32/32 \Rightarrow 32r:32q$

Multiplicación sin signo: DIVU, DIVUL.

Extensión de signo:

EXT	.W	Dn	$8 \Rightarrow 16$
	L	Dn	$16 \Rightarrow 32$
EXTB	L	Dn	$8 \Rightarrow 32$