



# Listado de instrucciones del MC68HC11

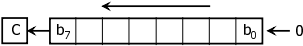
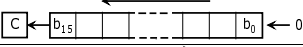
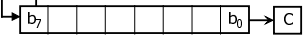
## Inst. de Transferencia

Nemónico	Operación	Descripción	Registro de Condiciones							
			S	X	H	I	N	Z	V	C
LDAA/B (opr)	Load Accumulo A/B	$M \rightarrow A/B$	•	•	•	•	↑	↑	0	•
LDD (opr)	Load Double Accumulator D	$M \rightarrow A; M + 1 \rightarrow B$	•	•	•	•	↑	↑	0	•
LDS/X/Y (opr)	Load Stack Pointer/IX/IY	$M : M + 1 \rightarrow SP/IX/IY$	•	•	•	•	↑	↑	0	•
PSHA/B	Push A / B onto Stack	$A / B \rightarrow (SP); (SP) - 1 \rightarrow SP$	•	•	•	•	•	•	•	•
PSHX/Y	Push IX / IY onto Stack (Lo First)	$(IX / IY) \rightarrow (SP) - 1; (SP) - 2; (SP) - 2 \rightarrow SP$	•	•	•	•	•	•	•	•
PULA/B	Pull A / B from Stack	$((SP) + 1) \rightarrow A / B; (SP) + 1 \rightarrow SP$	•	•	•	•	•	•	•	•
PULX/Y	Pull IX / IY from Stack (Hi First)	$((SP + 1); (SP + 2) \rightarrow IX/IY; (SP) + 2 \rightarrow SP$	•	•	•	•	•	•	•	•
STAA/B (opr)	Store Accumulator A/B	$A / B \rightarrow M$	•	•	•	•	↑	↑	0	•
STD (opr)	Store Accumulator D	$A \rightarrow M; B \rightarrow M + 1$	•	•	•	•	↑	↑	0	•
STS/X/Y (opr)	Store Stack Pointer / IX/ IY	$SP / IX / IY \rightarrow M : M + 1$	•	•	•	•	↑	↑	0	•
TAB	Transfer A to B	$A \rightarrow B$	•	•	•	•	↑	↑	0	•
TAP	Transfer A to CC register	$A \rightarrow CCR$	↑	↓	↑	↑	↑	↑	↑	↑
TBA	Transfer B to A	$B \rightarrow A$	•	•	•	•	↑	↑	0	•
TPA	Transfer CC register to A	$CCR \rightarrow A$	•	•	•	•	•	•	•	•
TSX/Y	Transfer Stack Pointer to IX / IY	$SP \rightarrow IX / IY$	•	•	•	•	•	•	•	•
TXS	Transfer IX to Stack Pointer	$IX \rightarrow SP$	•	•	•	•	•	•	•	•
TYS	Transfer IY to Stack Pointer	$IY \rightarrow SP$	•	•	•	•	•	•	•	•
XGDX	Exchange D with IX	$IX \rightarrow D, D \rightarrow IX$	•	•	•	•	•	•	•	•
XGDY	Exchange D with IY	$IY \rightarrow D, D \rightarrow IY$	•	•	•	•	•	•	•	•

## Inst. Aritméticas

ABA	Add Accumulators	$A + B \rightarrow A$	•	•	↑	•	↑	↑	↑	↑
ABX	Add B to X	$IX + (00:B) \rightarrow IX$	•	•	•	•	•	•	•	•
ABY	Add B to Y	$IY + (00:B) \rightarrow IY$	•	•	•	•	•	•	•	•
ADCA/B (opr)	Add Memory with carry to A/B	$A/B + M + C \rightarrow A/B$	•	•	↑	•	↑	↑	↑	↑
ADDA/B (opr)	Add Memory to A/B	$A/B + M \rightarrow A/B$	•	•	↑	•	↑	↑	↑	↑
ADDD (opr)	Add 16-bits to D	$D + (M : M + 1) \rightarrow D$	•	•	•	•	↑	↑	↑	↑
CLR (opr)	Clear Memory Byte/A/B	$0 \rightarrow M/A/B$	•	•	•	•	0	1	0	0
COM (opr)	1's Complement Memory byte/A/B	$\$FF - M/A/B \rightarrow M/A/B$	•	•	•	•	↑	↑	0	1
COMA/B			•	•	•	•	•	•	•	•
DAA	Decimal Adjust A	Adjust Sum to BCD	•	•	•	•	↑	↑	↑	↑
DEC (opr)	Decrement Memory Byte/A/B	$M/A/B - 1 \rightarrow M/A/B$	•	•	•	•	↑	↑	↑	•
DECA/B			•	•	•	•	•	•	•	•
DES	Decrement SP	$SP - 1 \rightarrow SP$	•	•	•	•	•	•	•	•
DEX/Y	Decrement IX/IY	$IX/IY - 1 \rightarrow IX/IY$	•	•	•	•	•	↑	•	•
FDIV	Fractional Divide 16 by 16	$D / IX \rightarrow IX; r \rightarrow D$	•	•	•	•	•	↑	↑	↑
IDIV	Integer Divide 16 by 16	$D / IX \rightarrow IX; r \rightarrow D$	•	•	•	•	•	↑	0	↑
INC (opr)	Increment Memory Byte/A/B	$M/A/B + 1 \rightarrow M/A/B$	•	•	•	•	↑	↑	↑	•
INCA/B			•	•	•	•	•	•	•	•
INS	Increment Stack Pointer	$SP + 1 \rightarrow SP$	•	•	•	•	•	•	•	•
INX/Y	Increment Index Register IX/IY	$IX/IY + 1 \rightarrow IX/IY$	•	•	•	•	•	↑	•	•
MUL	Multiply 8 by 8	$A * B \rightarrow D$	•	•	•	•	•	•	•	↑
NEG (opr)	2's Complement Memory Byte / A / B	$0 - M/A/B \rightarrow M/A/B$	•	•	•	•	↑	↑	↑	↑
NEGA/B			•	•	•	•	•	•	•	•
SBA	Subtract B from A	$A - B \rightarrow A$	•	•	•	•	↑	↑	↑	↑
SBCA/B (opr)	Subtract with Carry from A / B	$A/B - M \rightarrow A/B$	•	•	•	•	↑	↑	↑	↑
SUBA/B (opr)	Subtract Memory from A/B	$A / B - M \rightarrow A / B$	•	•	•	•	↑	↑	↑	↑
SUBD (opr)	Subtract Memory from D	$D - M : M + 1 \rightarrow D$	•	•	•	•	↑	↑	↑	↑

## Inst. Lógicas

ANDA/B (opr)	AND A/B with Memory	$A/B \bullet M \rightarrow A/B$	•	•	•	•	↑	↑	0	•
EORA/B (opr)	Exclusive OR A/B with Memory	$A/B \oplus M \rightarrow A/B$	•	•	•	•	↑	↑	0	•
ORAA/B (opr)	Or Accumulator A/B with Memory	$A/B \vee M \rightarrow A/B$	•	•	•	•	↑	↑	0	•
ASL (opr)	Arithmetic Shift Left		•	•	•	•	↑	↑	↑	↑
ASLA/B			•	•	•	•	•	•	•	•
ASLD	Arithmetic Shift Left		•	•	•	•	↑	↑	↑	↑
ASR (opr)	Arithmetic Shift Right		•	•	•	•	↑	↑	↑	↑
ASRA/B			•	•	•	•	•	•	•	•

Inst. de Rotación  
y Desplazamiento

Inst. de  
Comparación

Inst. de Salto y Bifurcación

Inst. de Control

Nemónico	Operación	Descripción	Registro de Condiciones							
			S	X	H	I	N	Z	V	C
LSL (opr) LSLA/B	Logical Shift Left Memory Byte / A / B		•	•	•	•	↑	↑	↑	↑
LSLD	Logical Shift Left Double		•	•	•	•	↑	↑	↑	↑
LSR (opr) LSRA/B	Logical Shift Right Memory Byte / A / B		•	•	•	•	0	↑	↑	↑
LSRD	Logical Shift Right Double		•	•	•	•	0	↑	↑	↑
ROL (opr) ROLA/B	Rotate Left Memory Byte / A / B		•	•	•	•	↑	↑	↑	↑
ROR (opr) RORA/B	Rotate Right Memory Byte / A / B		•	•	•	•	↑	↑	↑	↑
BITA/B (opr)	Bits Test A/B with Memory	A/B • M	•	•	•	•	↑	↑	0	•
CBA	Compare A to B	A - B	•	•	•	•	↑	↑	↑	↑
CMPA/B (opr)	Compare A/B to Memory	A/B - M	•	•	•	•	↑	↑	↑	↑
CPD/X/Y (opr)	Compare D/X/Y to Memory 16-bits	D/IX/IY - M:M + 1	•	•	•	•	↑	↑	↑	↑
TST (opr) TSTA/B	Test Memory Byte / A / B for Zero or Minus	M / A / B - 0	•	•	•	•	↑	↑	0	0
BCC (rel)	Branch if Carry Clear	? C = 0	•	•	•	•	•	•	•	•
BCS (rel)	Branch if Carry Set	? C = 1	•	•	•	•	•	•	•	•
BEQ (rel)	Branch if = Zero	? Z = 1	•	•	•	•	•	•	•	•
BGE (rel)	Branch if ≥ Zero	? N ⊕ V = 0	•	•	•	•	•	•	•	•
BGT (rel)	Branch if > Zero	? Z + (N ⊕ V) = 0	•	•	•	•	•	•	•	•
BHI (rel)	Branch if Higher	? C + Z = 0	•	•	•	•	•	•	•	•
BHS (rel)	Branch if Higher or Same	? C = 0	•	•	•	•	•	•	•	•
BLE (rel)	Branch if ≤ Zero	? Z + (N ⊕ V) = 1	•	•	•	•	•	•	•	•
BLO (rel)	Branch if Lower	? C = 1	•	•	•	•	•	•	•	•
BLS (rel)	Branch if Lower or Same	? C + Z = 1	•	•	•	•	•	•	•	•
BLT (rel)	Branch if < Zero	? N ⊕ V = 1	•	•	•	•	•	•	•	•
BMI (rel)	Branch if Minus	? N = 1	•	•	•	•	•	•	•	•
BNE (rel)	Branch if not = Zero	? Z = 0	•	•	•	•	•	•	•	•
BPL (rel)	Branch if Plus	? N = 0	•	•	•	•	•	•	•	•
BRA (rel)	Branch Always	? 1 = 1	•	•	•	•	•	•	•	•
BRCLR (opr) (msk) (rel)	Branch if Bit(s) Clear	? M • mm = 0	•	•	•	•	•	•	•	•
BRN (rel)	Branch Never	? 1 = 0	•	•	•	•	•	•	•	•
BRSET (opr) (msk) (rel)	Branch if Bit(s) Set	? (M) • mm = 0	•	•	•	•	•	•	•	•
BSR (rel)	Branch to Subrutine	PC → Stack	•	•	•	•	•	•	•	•
BVC (rel)	Branch if Overflow Clear	? V = 0	•	•	•	•	•	•	•	•
BVS (rel)	Branch if Overflow Set	? V = 1	•	•	•	•	•	•	•	•
JMP (opr)	Jump to an Address		•	•	•	•	•	•	•	•
JSR (opr)	Jump to Subrutine	PC → Stack	•	•	•	•	•	•	•	•
RTI	Return From Interrupt		↑	↓	↑	↑	↑	↑	↑	↑
RTS	Return From Subrutine		•	•	•	•	•	•	•	•
BCLR (opr) (msk)	Clear Bits	M • (mm) → M	•	•	•	•	↑	↑	0	•
BSET (opr) (msk)	Set Bit(s)	M + mm → M	•	•	•	•	↑	↑	0	•
CLC	Clear Carry Bit	0 → C	•	•	•	•	•	•	•	0
CLI	Clear Interrupt Mask	0 → I	•	•	•	0	•	•	•	•
CLV	Clear Overflow Flag	0 → V	•	•	•	•	•	•	0	•
SEC	Set Carry	1 → C	•	•	•	•	•	•	•	1
SEI	Set Interrupt Mask	1 → I	•	•	•	1	•	•	•	•
SEV	Set Overflow	1 → V	•	•	•	•	•	•	1	•
STOP	Stop Internal Clocks	--	•	•	•	•	•	•	•	•
WAI	Wait for Interrupt	Stak Register and WAIT	•	•	•	•	•	•	•	•

#### Registro de Condición; Símbolos:

- No afectado
- ↑ Afectado (puede ser cero o uno)
- 0 Puesto a cero (cleared)
- 1 Puesto a uno (set)
- ↓ Puede ser puesto a cero pero nunca a 1

#### Operaciones Lógicas:

- AND Lógica
- ✓ OR Lógica
- ⊕ OR Exclusiva