

**Tabla\_1.1**

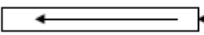
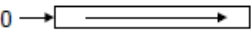
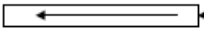
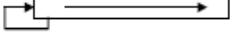
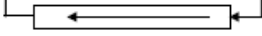
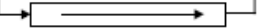
<i><b>Instrucción</b></i>	<i><b>Operación</b></i>	<i><b>Descripción</b></i>
MOVE    fuente,destino	$\text{destino} \leftarrow \text{fuente}$	Transfiere palabra de reg. a reg., reg. a mem, mem. a reg o mem. a mem. (fuente= mem. o reg.; destino = mem. o reg.)
LOAD    Ri, dir	$R_i \leftarrow \text{dir}$	Transfiere palabra de memoria a registro
STORE   dir, Ri	$R_i \leftarrow \text{dir}$	Transfiere palabra de registro a memoria
PUSH    fuente	$\text{Pila} \leftarrow \text{fuente}$	Transfiere palabra de mem. o reg. a la cabecera de pila
POP      destino	$\text{destino} \leftarrow \text{Pila}$	Transfiere palabra de cabecera de pila a mem. o reg.
MOVEM   fnte., dest., n	$\text{fnte}_0 \leftarrow \text{dest}_0 \dots\dots$ $\dots\dots \text{fnte}_{n-1} \leftarrow \text{dest}_{n-1}$	Transfiere n palabras a partir de una dir. inicial fuente y una dir. inicial destino

<i><b>Instrucción</b></i>	<i><b>Operación</b></i>	<i><b>Descripción</b></i>
ADD    fnte1,fnte2,dest	$\text{dest} \leftarrow \text{fnt1} + \text{fnt2}$	Suma dos operandos
SUB    fnte1,fnte2,dest	$\text{dest} \leftarrow \text{fnt1} - \text{fnt2}$	Resta dos operandos
MULT   fnte1,fnte2,dest	$\text{dest} \leftarrow \text{fnt1} * \text{fnt2}$	Multiplica dos operandos
DIV    fnte1,fnte2,dest	$\text{dest} \leftarrow \text{fnt1} / \text{fnt2}$	Divide dos operandos
NEG    fnte,dest	$\text{dest} \leftarrow - \text{fnt}$	Cambia de signo al operando
ABS    fnte,dest	$\text{dest} \leftarrow \text{Abs}(\text{fnt})$	Obtiene el valor absoluto del operando
INC    fnte,dest	$\text{dest} \leftarrow \text{fnt} + 1$	Suma 1 al operando
DEC    fnte,dest	$\text{dest} \leftarrow \text{fnt} - 1$	Resta 1 al operando
COMP   fnte1,fnte2	$\text{fnt1} - \text{fnt2}$ Activa Estado	Compara dos operandos y activa los bits de estado según el resultado de la comparación

**Tabla\_1.2**

<i><b>Instrucción</b></i>	<i><b>Operación</b></i>	<i><b>Descripción</b></i>
AND    fnte1,fnte2,dest	$\text{dest} \leftarrow \text{fnt1} \text{ AND } \text{fnt2}$	Realiza la Y lógica de dos operandos
OR      fnte1,fnte2,dest	$\text{dest} \leftarrow \text{fnt1} \text{ OR } \text{fnt2}$	Realiza la O lógica de dos operandos
NOT    fnte,dest	$\text{dest} \leftarrow \text{NOT } \text{fnt}$	Realiza la negación lógica del operando
XOR    fnte1,fnte2,dest	$\text{dest} \leftarrow \text{fnt1} \text{ XOR } \text{fnt2}$	Realiza la O exclusiva de dos operandos
BCLR   dest, n	$\text{dest}(n) \leftarrow 0$	Pone a cero el bit especificado
BSET   dest, n	$\text{dest}(n) \leftarrow 1$	Pone a uno el bit especificado
BTST   fnte, n	$\text{Estado} \leftarrow \text{fnte}(n)$	Activa estado según valor del bit indicado

**Tabla\_1.3**

<i><b>Instrucción</b></i>	<i><b>Operación</b></i>	<i><b>Descripción</b></i>
LSL    fnte,n		Desplaz. lógico izda. n bits (Logic Shift Left)
LSR    fnte,n		Desplaz. lógico decha. n bits (Logic Shift Right)
ASL    fnte,n		Desplaz. aritm. izda. n bits (Arith. Shift Left)
ASR    fnte,n		Desplaz. aritm. decha. n bits (Arith. Shift Right)
RL      dest, n		Rotación izda. n bits (Rotate Left)
RR      dest, n		Rotación decha. n bits (Rotate Right)

**Tabla 1.4**

<i><b>Instrucción</b></i>	<i><b>Operación</b></i>	<i><b>Descripción</b></i>
JMP    dir	$\text{PC} \leftarrow \text{dir}$	Salta (Jump) a la dirección especificada
Bcc    dir	If cc=TRUE then $\text{PC} \leftarrow \text{dir}$ Else $\text{PC} \leftarrow \text{PC} + \text{long\_instr}$	Bífurca (Branch) a la dirección especificada si la condición es cierta
JSR    dir	Estado, $\text{PC} \rightarrow \text{Pila}$ ; $\text{PC} \leftarrow \text{dir}$	Salto a subrutina; guarda en pila PC y estado
RTS	Estado, $\text{PC} \leftarrow \text{Pila}$	Retorno de subrutina; recupera de pila PC y estado
SKIP   n	$\text{PC} \leftarrow \text{PC} + n * \text{long\_instr}$	Salta el n° de instrucciones especificado
NOP	$\text{PC} \leftarrow \text{PC} + \text{long\_instr}$	No ejecuta ninguna operación, pero la ejecución continúa secuencialmente

**Tabla 1.5**