

RESUMEN INSTRUCCIONES 6809

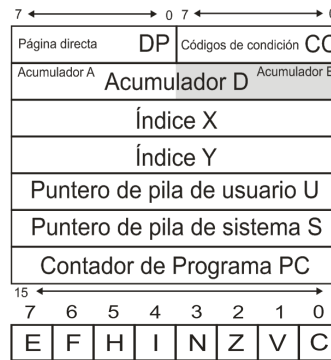
ENSAMBLADO

```
as6809 -o [-l] <nombre_fichero.asm>
aslink -s <nombre_fichero.rel>
m6809-run [-C] [-d] <nombre_fichero.s19>
```

MÁQUINA SIMULADA



REGISTROS



E (entire state)
F (FIRQ interrupt mask)
H (half-carry)
I (IRQ interrupt mask)
N (negative)
Z (zero)
V (overflow)
C (carry)


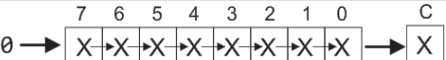
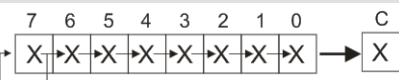
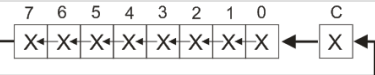
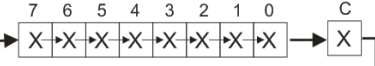
CARGA Y ALMACENAMIENTO

LDx operando	Carga de registros (inmediato #valor, dirección de memoria) <i>LDA, LDB, LDD, LDX, LDY, LDS, LDU</i>	E F H I N Z V C [] [] [] [x] [x] [0] []
STx operando	Carga de registros (inmediato #valor, dirección de memoria) <i>STA, STB, STD, STX, STY, STS, STU</i>	E F H I N Z V C [] [] [] [x] [x] [0] []
TFR R1,R2	Transferencia de registro a registro ($R2 \leftarrow R1$) <i>A,B,D,X,Y,U,S,PC,A,B,CC*,DP</i>	E F H I N Z V C [] [] [] [] [] [] []
EXG R1,R2	Intercambio de registro a registro ($R2 \leftrightarrow R1$) <i>A,B,D,X,Y,U,S,PC,A,B,CC*,DP</i>	E F H I N Z V C [] [] [] [] [] [] []

OPERACIONES ARITMÉTICAS

INCx	Suma una unidad al registro A,B o a una dirección de memoria <i>INCA, INCB ó INC dirección</i>	E F H I N Z V C [] [] [] [x] [x] [x] []
DECx	Resta una unidad al registro A,B o a una dirección de memoria <i>DECA, DECB ó DEC dirección</i>	E F H I N Z V C [] [] [] [x] [x] [x] []
CLRx	Limpia el contenido del registro A,B o una posición de memoria <i>CLRA, CLRB ó CLR dirección</i>	E F H I N Z V C [] [] [] [0] [1] [0] [0]
NEGx	Cambia de signo el registro A,B o de una posición de memoria (Complemento a 2) <i>NEGA, NEGB ó NEG dirección</i>	E F H I N Z V C [] [] [x] [x] [x] [x] [x]
ADDx operando	Suma al registro A,B ó D el valor inmediato o el contenido de una posición de memoria <i>ADDA, ADDB, ADDD</i>	E F H I N Z V C [] [] [x] [x] [x] [x] [x]
ABX	Suma al registro X el contenido de B ($X=X+B$) <i>ABX</i>	E F H I N Z V C [] [] [] [] [] [] []
ADCx	Suma al registro A ó B el valor inmediato o el contenido de una posición de memoria más el acarreo. <i>ADCA, ADCB</i>	E F H I N Z V C [] [] [x] [x] [x] [x] [x]
SUBx operando	Resta al registro A,B ó D el valor inmediato o el contenido de una posición de memoria <i>SUBA, SUBB, SUBD</i>	E F H I N Z V C [] [] [x] [x] [x] [x] [x]
SBCx	Resta al registro A ó B el valor inmediato o el contenido de una posición de memoria más el acarreo. <i>SBCA, SBCB</i>	E F H I N Z V C [] [] [x] [x] [x] [x] [x]
SEX	Extensión de signo de B en A de modo que D exprese el mismo número en Complemento a 2 <i>SEX</i>	E F H I N Z V C [] [] [] [x] [x] [0] []
MUL	Multiplica sin signo $D = A \times B$ <i>MUL</i>	E F H I N Z V C [] [] [] [] [x] [] [x]

DESPLAZAMIENTO Y ROTACIÓN DE BITS

ASLx	Desplazamiento a la izquierda aritmético y lógico																	
ASLA, ASLB ó ASL dirección		<table><tr><th>E</th><th>F</th><th>H</th><th>I</th><th>N</th><th>Z</th><th>V</th><th>C</th></tr><tr><td></td><td></td><td>x</td><td></td><td>x</td><td>x</td><td>x</td><td>x</td></tr></table>	E	F	H	I	N	Z	V	C			x		x	x	x	x
E	F	H	I	N	Z	V	C											
		x		x	x	x	x											
LSRx	Desplazamiento a la derecha lógico																	
LSRA, LSRB ó LSR dirección		<table><tr><th>E</th><th>F</th><th>H</th><th>I</th><th>N</th><th>Z</th><th>V</th><th>C</th></tr><tr><td></td><td></td><td>x</td><td></td><td>0</td><td>x</td><td></td><td>x</td></tr></table>	E	F	H	I	N	Z	V	C			x		0	x		x
E	F	H	I	N	Z	V	C											
		x		0	x		x											
ASRx	Desplazamiento a la derecha aritmético																	
ASRA, ASRB ó ASR dirección		<table><tr><th>E</th><th>F</th><th>H</th><th>I</th><th>N</th><th>Z</th><th>V</th><th>C</th></tr><tr><td></td><td></td><td>x</td><td></td><td>x</td><td>x</td><td></td><td>x</td></tr></table>	E	F	H	I	N	Z	V	C			x		x	x		x
E	F	H	I	N	Z	V	C											
		x		x	x		x											
ROLx	Rotación 1 bit a la izquierda																	
ROLA, ROLB ó ROL dirección		<table><tr><th>E</th><th>F</th><th>H</th><th>I</th><th>N</th><th>Z</th><th>V</th><th>C</th></tr><tr><td></td><td></td><td></td><td></td><td>x</td><td>x</td><td>x</td><td>x</td></tr></table>	E	F	H	I	N	Z	V	C					x	x	x	x
E	F	H	I	N	Z	V	C											
				x	x	x	x											
RORx	Rotación 1 bit a la derecha																	
RORA, RORB ó ROR dirección		<table><tr><th>E</th><th>F</th><th>H</th><th>I</th><th>N</th><th>Z</th><th>V</th><th>C</th></tr><tr><td></td><td></td><td></td><td></td><td>x</td><td>x</td><td></td><td>x</td></tr></table>	E	F	H	I	N	Z	V	C					x	x		x
E	F	H	I	N	Z	V	C											
				x	x		x											

SALTOS

NOP	No realiza nada	<table><tr><td>E</td><td>F</td><td>H</td><td>I</td><td>N</td><td>Z</td><td>V</td><td>C</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	E	F	H	I	N	Z	V	C								
E	F	H	I	N	Z	V	C											
JMP operando	Carga el PC con la dirección de memoria indicada en el operador	<table><tr><td>E</td><td>F</td><td>H</td><td>I</td><td>N</td><td>Z</td><td>V</td><td>C</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	E	F	H	I	N	Z	V	C								
E	F	H	I	N	Z	V	C											
Bxx operando	Salta a la dirección de memoria indicada en el operador. La dirección debe de estar en el rango [PC-128,PC+127] y puede ser una dirección o una etiqueta. Las distintas variantes realizan el salto o no dependiendo del estado de los flags. BCC, BCS: C=0 ó C=1, respectivamente BNE, BEQ: Z=0 ó Z=1, respectivamente BVC, BVS: V=0 ó V=1, respectivamente BPL, BMI: N=0 ó N=1, respectivamente BRA, BRN: siempre o nunca, respectivamente	<table><tr><td>E</td><td>F</td><td>H</td><td>I</td><td>N</td><td>Z</td><td>V</td><td>C</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	E	F	H	I	N	Z	V	C								
E	F	H	I	N	Z	V	C											
	Variantes en función de los flags, solamente tienen sentido después de una diferencia o una operación CMP BHI: minuyendo > sustraendo sin signo (Z=0 y C=0) BHS: minuyendo >= sustraendo sin signo (C=0) BLS: minuyendo <= sustraendo sin signo (Z=1 ó C=1) BLO: minuyendo<sustraendo sin signo (C=1) BGT: minuyendo >sustraendo con signo (Z=0 y C=N) BGE: minuyendo >=sustraendo con signo (C=N) BLE: minuyendo <= sustraendo con signo (Z=1 ó C<>N) BLT: minuyendo < sustraendo con signo (C<>N)	<table><tr><td>E</td><td>F</td><td>H</td><td>I</td><td>N</td><td>Z</td><td>V</td><td>C</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	E	F	H	I	N	Z	V	C								
E	F	H	I	N	Z	V	C											
LBxx operando	Igual que Bxx pero con desplazamientos respecto del PC de 16 bits con signo LBCC, LBSC, LBNE, LBEQ, LBVC, LBVS, LBPL, LBMI, LBRA, LBRN LBHI, LBHS, LBLS, LBLO, LBGT, LBGE, LBLE, LBLT	<table><tr><td>E</td><td>F</td><td>H</td><td>I</td><td>N</td><td>Z</td><td>V</td><td>C</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>	E	F	H	I	N	Z	V	C								
E	F	H	I	N	Z	V	C											
CMPx operando	Resta el contenido x menos el operando indicado (inmediato o contenido de una posición de memoria). El resultado se descarta, pero queda los flags afectados. CMPA, CMPB, CMPD, CMPS, CMPU, CMPX, CMPY	<table><tr><td>E</td><td>F</td><td>H</td><td>I</td><td>N</td><td>Z</td><td>V</td><td>C</td></tr><tr><td></td><td></td><td>x</td><td></td><td>x</td><td>x</td><td>x</td><td>x</td></tr></table>	E	F	H	I	N	Z	V	C			x		x	x	x	x
E	F	H	I	N	Z	V	C											
		x		x	x	x	x											
TSTx	Establece el valor de los flags N y Z en función del contenido del registro A, B o del contenido de una posición de memoria TSTA, TSTB ó TST dirección	<table><tr><td>E</td><td>F</td><td>H</td><td>I</td><td>N</td><td>Z</td><td>V</td><td>C</td></tr><tr><td></td><td></td><td></td><td></td><td>x</td><td>x</td><td>0</td><td></td></tr></table>	E	F	H	I	N	Z	V	C					x	x	0	
E	F	H	I	N	Z	V	C											
				x	x	0												

OPERACIONES CON BITS

ANDx operando	Operación lógica AND del registro x con un valor inmediato o el contenido de una dirección de memoria	E F H I N Z V C [][][][][X][X][0][]
	ANDA, ANDB	
ANDCC	Operación lógica AND del registro CC con un valor inmediato	E F H I N Z V C [X][X][X][X][X][X][X][X]
	ANDCC	
ORx operando	Operación lógica OR del registro x con un valor inmediato o el contenido de una dirección de memoria	E F H I N Z V C [][][][][X][X][0][]
	ORA, ORB	
ORCC	Operación lógica AND del registro CC con un valor inmediato	E F H I N Z V C [X][X][X][X][X][X][X][X]
	ORCC	
EORx	Operación lógica OR del registro x con un valor inmediato o el contenido de una dirección de memoria	E F H I N Z V C [][][][][X][X][0][]
	EORA, EORB	
COMx	Operación lógica NOT del registro x, o del contenido de una dirección de memoria	E F H I N Z V C [][][][][X][X][0][1]
	COMA, COMB ó COM dirección	
BITx	Operación lógica AND del registro x con un valor inmediato o el contenido de una dirección de memoria. Se descarta el resultado	E F H I N Z V C [][][][][X][X][0][]
	BITA, BITB	

PILAS

PSHx operando	Introduce en la pila x la lista de registros que se especifica como operando. El orden en el que se meten en la pila está prefijado	E F H I N Z V C [][][][][][][][]
	PSHU PC,S,Y,X,DP,B,A,CC	
	PSHS PC,U,Y,X,DP,B,A,CC	
PULx	Saca de la pila x la lista de registros que se especifica como operando. El orden en el que se saca en la pila está prefijado.	Si se saca CC E F H I N Z V C [X][X][X][X][X][X][X][X]
	PSHU CC,A,B,DP,X,Y,S,PC	
	PSHS CC,A,B,DP,X,Y,U,PC	

SUBROUTINAS

JSR operando	Sata a una subrutina. Almacena el PC en la pila S y lo carga con el operando (puede ser una dirección de memoria o una etiqueta)	E F H I N Z V C [][][][][][][][]
BSR operando	Salta a una subrutina. Almacena el PC en la pila S y lo carga con el operando (puede ser una dirección de memoria o una etiqueta). La dirección de memoria debe de estar en el rango [PC-128, PC+127].	E F H I N Z V C [][][][][][][][]
LBSR operando	Igual que BSR , pero permitiendo desplazamientos de 16 bits con signo	E F H I N Z V C [][][][][][][][]
RTS	Vuelve de una subrutina: Saca el PC de la pila S	E F H I N Z V C [][][][][][][][]

CALCULO DE DIRECCIONES

LEAx operando	Calcula la dirección efectiva a la que hace referencia el operando y la almacena en x. x puede ser X,Y,U o S	E F H I N Z V C [][][][][X][][][]
	LEAX, LEAY	
	LEAU, LEAS	E F H I N Z V C [][][][][][][][]

INTERRUPCIONES

SWI, SWI2, SWI3	Produce una interrupción software de tipo especificado	<div>E F H I N Z V C</div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div>
RTI	Vuelve de una interrupción, restaurando los registros si el flag E vale 1	<div>E F H I N Z V C</div> <div> <div>X</div> <div>X</div> <div>X</div> <div>X</div> <div>X</div> <div>X</div> <div>X</div> <div>X</div> </div>
	Si se restaura CC	
	Si no se restaura CC	<div>E F H I N Z V C</div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div>

DIRECTIVAS DE ENSAMBLADO

.INCLUDE "x"	Incluye, a la hora de ensamblar, el fichero <i>x</i> con los mismos efectos que si se hubiese copiado su contenido en ese punto del programa
.MODULE x	Declara el nombre del módulo actual
.AREA x (f)	<p>Declara el nombre del área donde debe ir el código que sigue. <i>f</i> son los flags de esa área, que deben coincidir en todas las apariciones del área en el proyecto. Los flags pueden ser:</p> <ul style="list-style-type: none"> REL/ABS: Reubicable o absoluta CON/OVR: Concatenable o superponible NOPAG/PAG: no página o página <p>Solamente puede aparecer una de las parejas y, de no aparecer, se considera la primera. ABS implica forzosamente OVR.</p>
.ORG x	En las áreas ABS, establece el punto de ensamblado en la dirección indicada en el operando
.GLOBL <i>x₁,..., x_n</i>	Establece que las etiquetas <i>xS</i> son etiquetas exportables o compartidas entre los módulos (y definida una en cada módulo, o en varios si coincide la definición)
.LOCAL <i>x₁,...,x_n</i>	Establece que las etiquetas <i>xS</i> son etiquetas locales. Solamente conoce de cada una el módulo donde se ha definido y usado. Puede haber varias etiquetas locales en distintos módulos con el mismo nombre y no interfieren entre sí.
<i>e</i>.EQU <i>x</i>	Da a la etiqueta <i>e</i> el valor arbitrario <i>x</i>
.BYTE <i>x₁,...,x_n</i>	Almacena la lista de bytes en la posición actual de ensamblado
.WORD <i>x₁,...,x_n</i>	Almacena la lista de palabras en la posición actual de ensamblado
.ASCII "x"	Almacena la cadena <i>x</i> en la posición actual de ensamblado
.ASCIIZ "x"	Almacena la cadena <i>x</i> en la posición de ensamblado, añadiendo un carácter nulo al final de ella.
.DS <i>x</i>	Avanza la posición de ensamblado <i>x</i> bytes
.SETDP <i>d,a</i>	Indica al ensamblador/enlazador que, a partir de ese punto, la página directa comienza en el área <i>a</i> , desplazamiento <i>d</i> (normalmente cero)