Modos de Direccionamiento del MC68HC11

Son diferentes formas en las que el CPU de una computadora accede a memoria externa para ejecutar la instrucción.

Soporta 6 modos detonados como:

- Direccionamiento inherente (INH)
- Direccionamiento inmediato (IMM)
- Direccionamiento directo(DIR)
- Direccionamiento extendido (EXT)
- Direccionamiento indexado (IND, X) o (IND, Y)
- Direccionamiento relativo (REL)



Suelen ser usados por otros MC.

De lo mas sencillo a lo mas complicado.

Direccionamiento inherente (INH)

- -instrucciones sencillas (no requiere parámetros adicionales para que el CPU funcione)
- -carecen de operando
- -codigo de instrucción de 8 a 16 bits.
- -no comparten mnemónico con instrucciones con otros métodos de direccionamiento.

								SET	DEINS	TRUCC	IONES D	EL MCC	BHC1	1													
1	MNEMONICO		мм			Dek		IND.X			TX3 Y.ON				PM			REL			Benderas de estado						
		OPCDDE	Ciclo	Byte	OPCODE	Ciclo	Byte	онсове	Ciclo	Byte	oecope	Cicle	Byte	oncome	Grie	Byte	орсове	Cicle	Byte	OPCDDE	Ckdo	Byte	s	×	1	N Z	v
2	aba	-	-	-	111	-	-	***	77		-	~	-	**	~	77	18	2	1	**	~	~	П	- >		X >	×
2	abx	-	-	-	-	-	-				-	-			-		3A	3	1		-	-	H	-			-
1	aby	-	-	-		-	-	***	-		-	-	**		-	-	18 3A	4	2	**	++	-		-			-
1	adca	89	2	2	99	3	2	A9	4	2	18 A9	5	3	B9	4	3			-	41		44	Н	-)		X >	
	adcb	C9	2	2	D9	3	2	E9	4	2	18 E9	5	3	F9.	4	3	-	-	-	**	-71	-		-)		X >	
Ц	adda	88	2	2	98	3	2	AB	4	2	18 AB	5	3	88	4	3		117	-	**	100		H	-)		X 3	
Ц	addb	CB	2	2	D8	3	2	EB	4	2	18 EB	5	3	FB	4	3	-	-	-	44.	-			-)	111	X	
Ц	addd	C3	4	3	Di	5	2	£3	6	2	18 E3	7	3	F3	6	3			=	**	**		H	-	-	X	
	anda		2	2	94	3	2	84	4	2	18 A4	5	3	84	4	3	-	116	-	**		-	H		**	X 3	
D	andib	C4	2	2	D4	3	2	E4	4	2	18 E4	5	3	F4	4	3	-	-		41	-77		H	-		X >	
	asl	-	-				-	68	6	2	18 68	7	1	78	6	3	~	-	-	44.		-	H	-	-	X X	
4	asla	-	-	200		-	-	**	***		++	-		**	- 00	-	48	2	1	**	10.	-	H	-	-	X >	
8	asib		- 41		44	-	-		14-	-44	44	140	44	4.0		40	58	2	1	4.0	144	04	H	-	-	X >	
4	asld		-	**		-	-	**	-		++	140		**			5	. 1	1	**	14		1	-	-	х э	
5	ase	- 10		-			-	67	6	2	18 67	7	3	77	6	3	47	-	-	-	**	-	H	-	11	X >	
7	asra	-	~	50-	-4	-	-	- 44	- +		++		**	. 44	-	44		2	1	**	. 11	++	н	9 4	111	X >	
B	aseb	-	-	-	(44	-	-	***	-		***	100	**	**	-		57	2	1	**	-		H	-	175	X >	4
9	betr		-	-	-11	-	-	**	ter		94	140		- 11		**	-	-	-	24	3	2	H	-	111		4
		100	-	-	15	6	3	10	7	3	18 10	8	4		- 00		94	-	-		0.4		H	-		X >	4
0	hes	-	-	-	100	-	-	77			10	-	**	11		-	~:	- 30	-	25	3	2	H		100		4
2	beq				- 14			**	10		-	-	**	**	14		-	100	-	27	3	2	H	-	+		#
1	bge	-		-	**		-	**			10-	**				_		-	-	2C	3	2	H	-	~		+
1	bgt	. 10	-	-	.94		-	**	-	. ++	-	-		- 14	110	**	211	-	-	2E	3	2	H	+	11	100	+
5		- 10	-		-11	-			100		- 10		-	. 10	. 00	-	-	- 10	~				H	-	1"	-	4
5	bhs		-			-	-		-			100	**				70	-	-	24	1	2	H	-	1.		1
7	bita	85 CS	2	2	95 DS	3	2	A5	4	2	18 A5	5	3	BS	4	3	-	-71	-	**	100		H	-	1	XX	
, B			2	2	21.5	3	2	ES		2	18 ES	5	3	FS	4	3		-	-	OF.			H	+	1	A >	4
,	ble	na.	-	40	- 11		-		140		-	44	**	**		-	-	A.A.	-	25	3	2	H	-	1		Ŧ
	blo		-		268	-	-	**	740		100	**		***		-01		-	-	25			H	-	-		#
	bls	-	-	**	044	-	-		10-	**		100	**	**	- 10		200	- 10	-	23 2D	3	2	Ħ	+	100	-	+
		_	-	-	**	-			- 10-		-	-		-			77.		-		3	2	H	-	1	F	+
	bmi	-	-		- 11	-	-									-	-		-	28	3	2	H	+	-	-	+
	bne	- 60	-	-	-44	-	-	44	- 00-	. 00		-	**	. 16	140	44	-		-	26	3		FF	-	**	-	4
	bpl	-	-	-	-	-	-	**	- 44		.+-	-			-		-	1,99	-	2A		2	H	-	-	H	+
6	bra		-	-			77	40	~	**		~		-10	44	24	-	-	-	20	3	2	H	+	**	-	+
à	brcir	0.0	-	80	13	6	4	1F	7	4	18 1F	- 8	5	44	- 40	44			-	40	0.00	-	ger je			4	-

OPCODE : código de instrucción (numero que le da el fabricante a cada instrucción)

Ciclo: ciclos que tarda

Byte: numero de bytes que tendremos en la instrucción

Directiva de ensamblador: SIRVEN PARA DAR ESTRUCTURA AL PROGRAMA

ORG \$8000, END

Localidad (hexadecimal)

-Se pueden tener varios ORG, porque puede haber varios sectores.

PROGRAMA SENCILLO, COMPILANDO

ODO 60000	\$8000	01	NOP
ORG \$8000	\$8001	08] INX
	\$8002	09	DEX
NOP	\$8003	18	1
INX	\$8004	80	INY
DEX	\$8005	18	DEY
INY	\$8006	09	JULI
DEY	\$8007	3D	MUL
MUL	\$8008	8F) XGDX
XGDX	\$8009	18	1
XGDY	\$800A	8F	→ XGDY
NEGA	\$800B	40) NEGA
NEGB	\$800C	50	NEGB
III OB	\$800D		
END	\$800E		
END	\$800F		

INY: usa dos bytes por eso se usan dos localidades de memoria

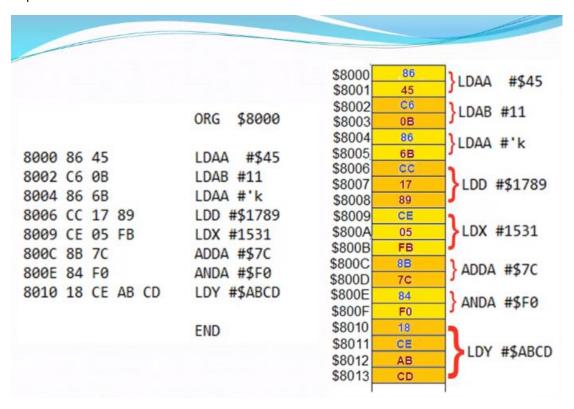
ASI TIENE QUE APARECER EN EL PROYECTO.

	ORG \$8000
8000 01	NOP
8001 08	INX
8002 09	DEX
8003 18 08	INY
8005 18 09	DEY
8007 3D	MUL
8008 8F	XGDX
8009 18 8F	XGDY
800B 40	NEGA
800C 50	NEGB

Un compilador genera dos archivos: 1 codigo fuente, 1 codigo objeto.

Direccionamiento inmediato (IMM).

- -direccionamientos que poseen un código
- por un #



Los operandos se tienen que convertir a hexadecimal

Los caracteres se ponen de la siguiente manera en código ascii : 'f, 'k. y solo soporta un caracter.

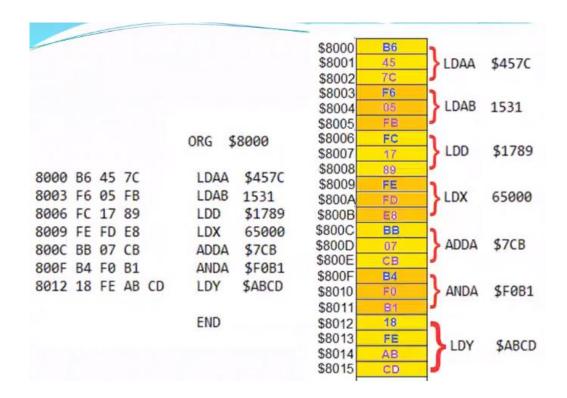
Direccionamiento directo (DIR).

- -un operando de 8 bits.
- -Codigo de instrucción de 8 16 bits.
- -Solo puede accedr a la memoria RAM (256 localidades)
- -Desde la localida \$0000 \$00FF
- -no van precedidos por un #



Direccionamiento extendido (EXT)

- -Un operando de 16 bits; se le da tratamiento de dirección
- -Codigo de instrucción de 8 16 bits.
- Accede a cualquier parte de memoria (ROM Y RAM).
- -Un poco mas lento y mas memoria.
- no le antecede un #.
- Tenemos que saber en que parte de la memoria ram estamos almacenando datos.



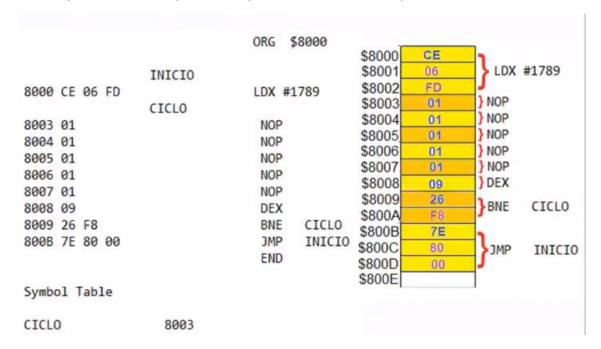
Direccionamiento indexado

- -Un operando de 8 bits, se le da tratamiento de desplazamiento algebraico.
- -precedido por una ", " y una x ó y.
- código de instrucción de 8 a 16 bits.

LISTANO NEI BLOZIAIIIA \$8000 A6 LDAA \$45,X \$8001 45 \$8002 18 LDAB \$67,Y \$8003 E6 67 ORG \$8000 \$8004 \$8005 EC LDD \$17,X \$8006 17 3000 A6 45 LDAA \$45,X CD \$8007 3002 18 E6 67 LDAB \$67,Y \$8008 EE LDX \$F1,Y 3005 EC 17 LDD F1 \$17,X \$8009 \$800A 18 3007 CD EE F1 LDX \$F1,Y \$800B ADDA \$07,Y AB 300A 18 AB 07 ADDA \$07,Y \$800C 07 300D A4 F0 ANDA \$F0,X \$800D A4 ANDA \$F0,X 300F 18 EE AB LDY \$AB,Y \$800E \$800F 18 LDY \$AB,Y \$8010 EE END \$8011 AB \$8012

Direccionamiento relativo (REL).

- -Codigo de instrucción de 8 16 bits.
- Operando de 8 bits.
- se emplea para hacer saltos.
- su operando siempre se expresa como un etiqueta.



¿Cómo se calculo F8?

