Apéndice: Cartilla de Assembler Z-80

1. Características del Procesador Z-80

RDM de 2 bytes (0000h al FFFFh)

RBM de 1 bytes (8 bits en cada Transferencia de Memoria)

El almacenamiento se realiza usando Little-Endian

1.1. Registros

Existen *tres grupos* de registros: grupo GR, grupo GR' y grupo de Registros Especiales. La diferencia entre los del grupo GR' con los del GR es que sus contenido no son directamente accesibles, sólo pueden ser utilizados para intercambiarse con los del grupo GR a alta velocidad.

Registros Especiales:

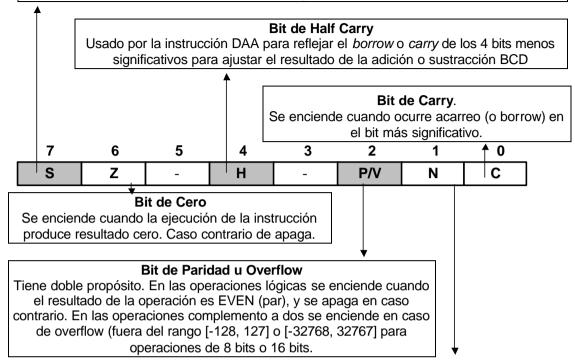
- > I: Registro de Interrupción (8 bits). Indica el lugar del vector de Interrupción.
- ➤ R: Registro Contador ó Refresh (8 bits). Los 7 bits menos significativos indican la cantidad de instrucciones ejecutadas por el procesador. Es incrementado por el procesador en cada fetch.
- ➤ IX, IY: Registros Índices (16 bits). Se usan para manipular datos y direcciones. Cuando se usa en direcciones, el contenido de un desplazamiento especificado en la instrucción es sumado o restado del registro índice para calcular la dirección efectiva de un operando.
- > SP: Registro Stack Pointer (16 bits). Contiene la dirección del Stack.
- ▶ PC: Registro Contador de Programa (16 bits). Contiene la dirección de la próxima instrucción a ejecutar. Es actualizado automáticamente después de cada fetch.

Registros del grupo GR:

- A: Registro Acumulador (8 bits). Se utiliza para operaciones aritméticas, lógicas e instrucciones de E/S.
- ▶ B, C, D, E, H, L: Registros de Uso General para manipular datos de 8 bits. Pueden ser tratados en forma individual (B, C, D, E, H, L) o aparearse en tres pares (BC, DE, HL) para manipular 16 bits.
- ➤ F: Registro de Flag (8 bits). Guarda los bits de estados que guardan los resultados de la ejecución de instrucciones. Se consulta para controlar el flujo de ejecución del programa y la operación de la instrucción. El significado de cada uno de sus bits se detalla en el siguiente cuadro:

Bit de Signo

Usado en operaciones de números con signo donde si está encendido es interpretado como negativo



Bit Negativo

Se enciende si el resultado de la ultima operación fue de sustracción (SUB, DEC, CP, etc.) y se apaga si fue de adición (ADD, INC, etc.)

1.2. Modos de Direccionamiento

- Registro Implícito: Ciertas instrucciones se aplican automáticamente a un registro predeterminado. Por ejemplo las instrucciones aritméticas y lógicas utilizan direccionamiento implícito, ya que todas ellas realizan operaciones con el contenido del registro acumulador A.
- Registro Directo: El código de operación de la instrucción contiene cuál o cuáles registros están implicados en la ejecución de la instrucción.
- Registro Indirecto: Alguno de los registros de uso general (16 bits: BC, DE o HL) indican la dirección donde se encuentra el operando de memoria.
- Registro Indexado: La dirección del operando en memoria es calculada usando el contenido de uno de los registros índice IX o IY más un desplazamiento de 8 bits con signo que se especifica en las instrucción.
- **Extendido:** La dirección del operando en memoria es especificada por dos bytes contenidos en la instrucción. La dirección se coloca entre paréntesis.
- Inmediato: El operando está contenido en 1 ó 2 bytes de la instrucción.
- Relativo: Se utiliza sólo para instrucciones de bifurcación condicional o incondicional. El desplazamiento es sumado al registro PC.
- I/O: Se utiliza sólo para instrucciones de E/S.

2. Directivas al Compilador (Pseudoinstrucciones)

Son acciones que debe tomar el compilador, pero NO GENERAN CÓDIGO.

Definición de Constantes Simbólicas

Asignan un nombre simbólico a una expresión. Existen dos formas de hacerlo:

nombre	equ	expresión
nombre	defl	expresión

• Definición de Datos

Reservan posiciones de memoria para guardar variables. Existen 5 formas de hacerlo, alguna de ellas permiten inicializar con un valor y otras sólo reservar espacio:

nombre	db	listaDeExpresiones	(cada dato en 1 byte)
nombre	dw	listaDeExpresiones	(cada dato en un doble byte)
nombre	defm	mensajeDeCracteres	(cada caracter en 1 byte)
nombre	deff	listaDeNrosReales	(IEEE 754 de 32 bits)
nombre	ds	cantidadBytes	(sólo reserva espacio)

• Control de Ensamblado

Indica que la primera instrucción a ejecutar (no necesariamente la primera ejecutable) es la que se encuentra en la dirección indicada por rótulo:

Referencias Externas

Para indicar que pueden ser usados por otros módulos (análogo a colocar el "::" haciendo el rótulo o constante simbólica pública) se usa:

public listaDeldentificadores

ent listaDeldentificadores

Para indicar que los rótulos o constantes simbólicas están definidas en otros módulos y es el linkeditor el que debe resolverlas, se coloca:

extern listaDeldentificadores

external listaDeldentificadores

Directivas de Segmento

Segmento absoluto:

Aseg

org direccionDeMemoria

Segmento Reubicable:

Cseg

zonaDeCodigo

.

Dseg

zonaDeDatos

.

3. Set de Instrucciones Detalladas



Explica el modo de direccionamiento de los operandos que intervienen en la instrucción y el fuincionamiento de la misma.



Indica el formato de la instrucción (forma de uso para incluirla en un código fuente)



Indica la cantidad de ciclos de memoria utilizados en la ejecución de la instrucción



Indica la cantidad de ciclos de CPU utilizados en la ejecución de la instrucción



Muestra el código de operación (código de máquina) asociado a la instrucción (y por lo tanto la longitud de la instrucción)



Muestra el estado de los bits del registro F después de la ejecución de la instrucción, según la siguiente convención:

- ☑ si es alterado por la operación
- si no es alterado por la operación
- 0 si es apagado por la operación
- 1 si es encendido por la operación
- ? si es indefinido después de la operación

En el caso de que una instrucción setee el bit P/V (Paridad/Overflow), se indica con una P o con una V según esté indicando Paridad u Overflow, respectivamente.

TAB	LA 1
Regist ro	Códig o
В	000
С	001
D	010
E	011
Н	100
L	101
-	110
Α	111

TAB	LA 2
Regist	Códig
ro	0
ВС	00
DE	01
HL	10
AF	11

TAB	LA 3
Regist	Códig
ro	0
BC	00
DE	01
HL	10
SP	11

TAB	LA 4
Regist	Códig
ro	ο
ВС	00
DE	01
IX	10
SP	11

TAB	LA 5
Regist	Códig
ro	0
ВС	00
DE	01
IY	10
SP	11

TABLA 6							
Flag	Significado	Códig o					
NZ	No zero	000					
Z	Zero	001					
NC	No carry	010					
С	Carry	011					
РО	Paridad impar	100					
PE	Paridad par	101					
Р	Signo mas	110					
M	Signo menos	111					

3.1. Instrucciones Aritméticas (8 y 16 bits)

Add (8 bits)

La suma de 8 bits utiliza *el modo de direccionamiento implícito*, ya que siempre deja el resultado en el registro acumulador A.

Además, el otro operando que interviene en la suma puede tener diferentes modos de direccionamiento, que se indican en la tabla. La longitud de instrucción varía según el operando en cuestión.

="		£;3				ħ.					
=		٠٫٦	۵	(2000)	S	Z	Н	P V	N	С	
Registro Directo	Add A, r	22	4	8	V	\square	V	N.	0	V	
A ←A + r	Donde r puede ser alguno de los siguientes registros de 8 bits: B, C, D, E, H, L, A			1000 donde codifica alguno de los registros de 8 bits con la convención de la tabla 1				V			
Registro Indirecto	Add A, (HL)	2	6	86	Ø	Ø	V	V	0	☑	
A ←A + (HL)								٧			
Registro Indexado	Add A, (IX + d)	6	3	DD 86 d	Ø	Ø	Ø		0	V	
A ←A + (IX + d)	donde d es un desplazamiento de 8 bits							٧			
Registro Indexado	Add A, (IY + d)	6	3	FD 86 d	V	V	V	V	0	V	
A ←A + (IY + d)	donde d es un desplazamiento de 8 bits							>			
Inmediato	Add A, v	2	6	C6 v	V	V	V	K	0	\sqrt	
A ←A + v	donde v es un dato de 8 bits							٧			

Add (16 bits)

La suma de 16 bits utiliza *el modo de direccionamiento implícito* y el de *registro directo*. Sólo se pueden hacer sumas entre ciertos pares de registros.

="	~ f	£3					r	h		
=		ئہا		[s	Z	Н	P V	N	С
Registro Directo	Add HL, r	5	7	9	×	×	?	×	0	☑
HL ←HL + r	donde r puede ser alguno de los siguientes pares registros de 16 bits: BC, DE, HL, SP			001001 donde codifica alguno de los pares de registros de 16 bits con la convención de la tabla 3						
Registro Directo	Add IX, r	6	10	DD9	×	×	?	×	0	V
IX ←IX + r	donde r puede ser alguno de los siguientes pares registros de 16 bits: BC, DE, IX, SP			11011101 001001 donde codifica alguno de los pares de registros de 16 bits con la convención de la tabla 4						
Registro Directo	Add IY, r	6	10	FD9	×	×	?	×	0	V
IY ←IY + r	donde r puede ser alguno de los siguientes pares registros de 16 bits: BC, DE, IY, SP			1111101 00—1001 donde codifica alguno de los pares de registros de 16 bits con la convención de la tabla 5						

Adc (8 bits)

La suma con carry utiliza *el modo de direccionamiento implícito*, ya que siempre deja el resultado en el registro acumulador A.

Además, el otro operando que interviene en la suma puede tener diferentes modos de direccionamiento, que se indican en la tabla. La longitud de instrucción varía según el operando en cuestión.

="	A D			H						
=		لہا		[50000]	s	Z	Н	P V	N	С
Registro Directo	Adc A, r	2	4	8	Ø	Ø	Ø	V	0	V
A ←A + r + c	donde r puede ser alguno de los siguientes registros de 8 bits: B, C, D, E, H, L, A			10001 donde codifica alguno de los registros de 8 bits con la convención de la tabla 1				>		
Registro Indirecto	Adc A, (HL)	6	6	8E	V	Ø	Ø	V	0	V
A ←A + (HL) + c								٧		
Registro Indexado	Adc A, (IX + d)	6	14	DD 8E d	V	\square	V	V	0	V
A←A+(IX+d) + c	donde d es un desplazamiento de 8 bits							٧		
Registro Indexado	Adc A, (IY + d)	6	14	FD 8E d	V	Ø	V	V	0	V
A←A+(IY+d) + c	donde d es un desplazamiento de 8 bits							>		
Inmediato	Adc A, v	2	6	CE v	V	Ø	V	V	0	V
A ←A + v + c	donde v es un dato de 8 bits							٧		

Adc (16 bits)

La suma con carry de 16 bits utiliza el **modo de direccionamiento implícito** y el **directo**. Sólo se pueden hacer sumas entre ciertos pares de registros.

="	~	ťή					r	h		
		لہا		[FEEST]	s	Z	H	P V	N	С
Registro Directo	Adc HL, r	6	10	EDA		V	?	<u>N</u>	0	V
HL ←HL + r + c	donde r puede ser alguno de los siguientes pares registros de 16 bits: BC, DE, HL, SP			11101101 011010 donde codifica alguno de los pares de registros de 16 bits con la convención de la tabla 3				>		

Sub (8 bits)

La diferencia de 8 bits utiliza *el modo de direccionamiento implícito*, ya que siempre deja el resultado en el registro acumulador A.

Además, el otro operando que interviene en la diferencia puede tener distintos modos de direccionamiento. La longitud de instrucción varía según el operando en cuestión.

=	~	£;3					h	h		
		רָ			S	Z	Н	P V	N	С
Registro Directo	Sub r	2	4	9	Ø	Ø	☑	V	1	☑
A ←A - r	donde r puede ser alguno de los siguientes registros de 8 bits: B, C, D, E, H, L, A			10010 donde codifica alguno de los registros de 8 bits con la convención de la tabla 1				>		
Registro Indirecto A ←A - (HL)	Sub (HL)	2	6	96	V	V	V	∑ >	1	V
Registro Indexado A ←A - (IX + d)	Sub (IX + d) donde d es un desplazamiento de 8 bits	6	14	DD 96 d	\S		V	< 🛚	1	
Registro Indexado A ←A - (IY + d)	Sub (IY + d) donde d es un desplazamiento de 8 bits	6	14	FD 96 d	Ø	\square	\square	>	1	
Inmediato A ←A - v	Sub v donde v es un dato de 8 bits	2	6	D6 v	V	V	V	>	1	\

Sbc (8 bits)

La diferencia de 8 bits con carry utiliza *el modo de direccionamiento implícito*, ya que siempre deja el resultado en el registro acumulador A.

Además, el otro operando que interviene en la diferencia puede tener diferentes modos de direccionamiento. La longitud de instrucción varía según el operando en cuestión.

="	~	£3					h	h		
=		لبا		[ESSECTION 100	S	Z	Н	P V	N	С
Registro Directo	Sbc A, r	2	4	9	Ø	Ø	V	V	1	V
A ←A - r - c	donde r puede ser alguno de los siguientes registros de 8 bits: B, C, D, E, H, L, A			10011 donde codifica alguno de los registros de 8 bits con la convención de la tabla 1				V		
Registro Indirecto	Sbc A, (HL)	2	6	9E	Ø	Ø	Ø	Ø	1	V
A ←A - (HL) - c								V		
Registro Indexado	Sbc A, (IX + d)	6	14	DD 9E d	Ø	V	Ø	Ø	1	☑
A←A -(IX + d) - c	donde d es un desplazamiento de 8 bits							V		
Registro Indexado	Sbc A, (IY + d)	6	14	FD 9E d	Ø	V	V	V	1	V
A←A -(IY + d) - c	donde d es un desplazamiento de 8 bits							٧		
Inmediato	Sbc A, v	2	6	DE v	V	Ø	Ø	Ø	1	V
A ←A - v - c	donde v es un dato de 8 bits							٧		

Sbc (16 bits)

La diferencia de 16 bits con carry utiliza e*l modo de direccionamiento implícito* y el *directo*. Sólo puede realizarse entre determinado pares de registros.

="	~	ťΊ					h	h		
		ربا		[SSSSS]	S	Z	H	P >	N	С
Registro Directo	Sbc HL, r	6	10	ED2	Ø	Ø	?	V	1	V
HL ←HL - r - c	donde r puede ser alguno de los siguientes pares de registros de 16 bits: BC, DE, HL, SP			11101101 010010 donde codifica alguno de los pares de registros de 16 bits con la convención de la tabla 3				>		

Inc (8 bits)

Esta es una operación unaria, ya que tiene un único operando el cual incrementa en 1.El operando funciona como origen y destino, ya que en el mismo se guarda el valor modificado. La longitud de instrucción varía según el operando en cuestión.

="	~	£;3					h	h		
=		لہا			s	Z	Н	P V	N	С
Registro Directo	Inc r	2	4		V	V	V	V	0	V
r ←r + 1	donde r puede ser alguno de los siguientes registros de 8 bits: B, C, D, E, H, L, A			00100 donde codifica alguno de los registros de 8 bits con la convención de la tabla 1				V		
Registro Indirecto (HL) ← (HL) + 1	Inc (HL)	4	10	34	\square	Ø	\square	✓	0	V
Registro Indexado (IX + d) ← (IX + d) + 1	Inc (IX + d) donde d es un desplazamiento de 8 bits	8	18	DD 34 d	\square	Ø	\square	>	0	Ø
Registro Indexado (IY + d) ← (IY + d) + 1	Inc (IY + d) donde d es un desplazamiento de 8 bits	8	18	FD 34 d	\square	V		>	0	I

Inc (16 bits)

Esta es una operación unaria, ya que tiene un único operando el cual incrementa en 1.El operando funciona como origen y destino, ya que en el mismo se guarda el valor modificado. La longitud de instrucción varía según el operando en cuestión.

="	~	£;3					h	h		_
		لہا		[30000]	s	Z	Н	P V	N	С
Registro Directo	Inc r	2	4	3	×	×	×	×	×	×
r ←r + 1	donde r puede ser alguno de los siguientes pares de registros de 16 bits: BC, DE, HL, SP			000011 donde codifica alguno de los pares de registros de 16 bits con la convención de la tabla 3						
Registro Implícito IX ← IX + 1	Inc IX	3	7	DD 23	×	×	×	×	×	×
Registro Implícito IY ← IY + 1	Inc IY	3	7	FD 23	×	×	×	×	×	×

Dec (8 bits)

Esta es una operación unaria, ya que tiene un único operando el cual se decrementa en 1. El operando funciona como origen y destino, ya que en el mismo se guarda el valor modificado. La longitud de instrucción varía según el operando en cuestión.

="	~	£;3					h	h		
=		لہا			s	Z	Н	P V	N	С
Registro Directo	Dec r	2	4		Ø	Ø	Ø		1	V
r ←r - 1	donde r puede ser alguno de los siguientes registros de 8 bits: B, C, D, E, H, L, A			00101 donde codifica alguno de los registros de 8 bits con la convención de la tabla 1				V		
Registro Indirecto (HL) ← (HL) - 1	Dec (HL)	4	10	35	\square	V	\square	✓	1	V
Registro Indexado (IX + d) ← (IX + d) - 1	Dec (IX + d) donde d es un desplazamiento de 8 bits	8	18	DD 35 d	\square	I	\square	>	1	Ø
Registro Indexado (IY + d) ← (IY + d) - 1	Dec (IY + d) donde d es un desplazamiento de 8 bits	8	18	FD 35 d	Ø	V	V	>	1	✓

Dec (16 bits)

Esta es una operación unaria, ya que tiene un único operando el cual se decrementa en 1. El operando funciona como origen y destino, ya que en el mismo se guarda el valor modificado. La longitud de instrucción varía según el operando en cuestión.

="	~	£;3					h	h		
=		لہا			s	Z	Н	P V	N	С
Registro Directo	Dec r	2	4	В	×	×	×	×	×	×
r ←r − 1	donde r puede ser alguno de los siguientes pares de registros de 16 bits: BC, DE, HL, SP			001011 donde codifica alguno de los pares de registros de 16 bits con la convención de la tabla 3						
Registro Implícito IX ← IX - 1	Dec IX	3	7	DD 2B	×	×	×	×	×	×
Registro Implícito IY ← IY - 1	Dec IY	3	7	FD 2B	×	×	×	×	×	×

Neg

La negación sólo se puede hacer en el registro acumulador A. Representa el complemento a 2 del acumulador (aplicar NEG es lo mismo que hacer 0-A)

="	~	£-,J					h	h		
		لہا		1000	S	Z	Н	P >	N	С
Registro Implícito	Neg	2	6	ED 44	Ø	V	V	S	1	V
A ← - A								٧		

MIt

La multiplicación aparece en assembler (es una extensión al Z80). Opera sólo con ciertos registros de 16 bits. Multiplica un byte por el otro byte dejando el resultado en el registro de 16 bits.

El operando funciona como origen y destino, ya que en el mismo se guarda el valor modificado.

Sólo sirve para multiplicar números positivos.

="	~	£.t					h	h		
		رب			S	Z	Н	P V	Z	С
Registro Directo	Mlt r	13	17	EDC	×	×	×	×	×	×
r ←rh * rl	donde r puede ser alguno de los siguientes pares de registros de 16 bits: BC, DE, HL, SP			11101101 011100 donde codifica alguno de los pares de registros de 16 bits con la convención de la tabla 3						

3.2. Instrucciones de Transferencia de Datos

Ld (8 bits)

		ָרָ <u>י</u>					r	h		
="		£3		(Passe)	s	Z	Н	P V	N	С
Registro Implícito en fuente y destino A ←I	Ld A, I	2	6	ED 57	Ø	☑	0	×	0	×
Registro Implícito en fuente y destino A ←R	Ld A, R	2	6	ED 5F	Ø	☑	0	×	0	×
Registro Implícito en fuente y destino I ←A	Ld I, A	2	6	ED 47	×	×	×	×	×	×
Registro Implícito en fuente y destino R ←A	Ld R, A	2	6	ED 4F	×	×	×	×	×	×
Registro Directo en fuente y en destino r ← r'	Ld r, r' donde r y r' son cualquier registro de la tabla 1	2	4	los primeros codifican al registro r y los otros codifican al registro r', según la tabla 1	×	×	×	×	×	×
Registro Implícito para el destino, registro Indirecto para el origen A ← (BC)	Ld A, (BC)	2	6	0A	×	×	×	×	×	×

="	~	£; }					h	h		
=		لہا		(50000)	s	Z	Н	P V	N	С
Registro Implícito para el destino, registro Indirecto para el origen A ← (DE)	Ld A, (DE)	2	6	1A	×	×	×	×	×	×
Registro Indirecto para el destino, Registro Implícito para el origen (BC) ← A	Ld (BC), A	6	7	02	×	×	×	×	×	×
Registro Indirecto para el destino, Registro Implícito para el origen (DE) ← A	Ld (DE), A	3	7	12	×	×	×	×	X	X
Registro Implícito para el destino, Extendido para el fuente A ←(mn)	Ld A, (mn) donde mn es un número de 16 bits	4	12	3A n m	×	×	×	×	×	×
Extendido para el destino, Registro Implícito para el fuente (mn) ←A	Ld (mn), A donde mn es un número de 16 bits	5	13	32 n m	×	×	×	×	×	×
Registro Directo para el destino, registro Indirecto para el fuente r ←(HL)	Ld r, (HL) donde r es alguno de los registros de la tabla 1	2	6	01110 donde codifica alguno de los registro de la tabla 1	×	×	×	×	×	×

="	~	£3					ļ.	b		
=		ئہا		(2000C)	s	Z	Н	P V	N	С
Registro Indirecto para el destino, Registro Directo para el fuente (HL) ← r	Ld (HL), r donde r es alguno de los registros de la tabla 1	3	7	7 01110 donde codifica alguno de los registro de tabla 1	×	×	×	×	×	×
Registro Directo para el destino, registro Indexado para el fuente r ←(IX +d)	Ld r, (IX+d) donde r es alguno de los registros de la tabla 1, y d es un desplazamiento de 8 bits	6	14	DD d 11011101 01110 d donde codifica algún registro de la tabla 1	×	×	×	×	×	×
Registro Directo para el destino, registro Indexado para el fuente r ←(IY +d)	Ld r, (IY+d) donde r es alguno de los registros de la tabla 1, y d es un desplazamiento de 8 bits	6	14	FD d 11111101 01110 d donde codifica algún registro de la tabla 1	×	×	×	K	X	×
Registro Indexado para el destino, registro Directo para el fuente (IX +d) ←r	Ld (IX+d), r donde r es alguno de los registros de la tabla 1, y d es un desplazamiento de 8 bits	7	15	DD 7 d 11011101 01110 d donde codifica algún registro de la tabla 1	×	×	×	×	×	×
Registro Indexado para el destino, registro Directo para el fuente (IY +d) ←r	Ld (IY+d), r donde r es alguno de los registros de la tabla 1, y d es un desplazamiento de 8 bits	7	15	FD 7 d 11111101 01110 d donde codifica algún registro de la tabla 1	×	×	×	×	×	×

="	~	£3					h	h		
=		ئ.		20000	s	Z	Н	P V	N	С
Registro Directo para el destino, Inmediato para el fuente r ←v	Ld r, v donde r es un registro de la tabla 1 y v es un dato de 8 bits	2	6	v 00110 v donde codifica alguno de los registros de la tabla 1	×	×	×	×	×	×
Registro Indirecto para el destino, Inmediato para el fuente (HL) ←v	Ld (HL), v donde v es un dato de 8 bits	3	9	36 v	×	×	×	X	X	X
Registro Indexado para el destino, Inmediato para el fuente (IX +d) ←v	Ld (IX+d), v donde v es un dato de 8 bits	5	15	DD 36 d v	×	×	×	×	×	×
Registro Indexado para el destino, Inmediato para el fuente (IY +d) ←v	Ld (IY+d), v donde v es un dato de 8 bits	5	15	FD 36 d v	×	×	×	×	×	×

Ld (16 bits)

="	~	£; }					h	h		
		ربا			s	Z	Н	P V	N	С
Registro Implícito en fuente y destino SP ←HL	Ld SP, HL	2	4	F9	×	×	×	×	×	×
Registro Implícito en fuente y destino SP ←IX	Ld SP,IX	3	7	DD F9	×	×	×	×	×	×
Registro Implícito en fuente y destino SP ←IY	Ld SP, IY	3	7	FD F9	×	×	×	×	×	×
Registro Implícito para el destino, Extendido para el fuente HL ←(mn)	Ld HL, (mn) donde mn es un número de 16 bits	3	15	2A n m	X	×	×	X	X	X
Registro Implícito para el destino, Extendido para el fuente IX ←(mn)	Ld IX, (mn) donde mn es un número de 16 bits	6	18	DD 2A n m	×	×	×	×	×	×
Registro Implícito para el destino, Extendido para el fuente IY ←(mn)	Ld IY, (mn) donde mn es un número de 16 bits	6	18	FD 2A n m	×	×	×	×	×	×
Extendido para el destino, Registro Implícito para el fuente (mn)←HL	Ld (mn), HL donde mn es un número de 16 bits	6	16	22 n m	×	×	×	×	×	×

="	~	£3					h	h		_
=		ئہا		20000	s	Z	Н	P V	N	С
Extendido para el destino, Registro Implícito para el fuente (mn) ←IX	Ld (mn), IX donde mn es un número de 16 bits	7	19	DD 22 n m	×	×	×	×	×	×
Extendido para el destino, Registro Implícito para el fuente (mn) ←IY	Ld (mn), IY donde mn es un número de 16 bits	7	19	FD 22 n m	×	×	×	×	×	×
Registro Directo para el destino, Extendido para el fuente r ←(mn)	Ld r, (mn) donde r es alguno de los registros de la tabla 3, y mn es un dato de 16 bits	6	9	EDB n m 11101101 011011 n m	×	×	×	×	×	×
Extendido para el destino, Registro Directo para el fuente (mn) ←r	Ld (mn), r donde r es alguno de los registros de la tabla 3 (≠HL), y mn es un dato de 16 bits	7	19	ED3 n m 11101101 010011 n m	×	×	×	×	×	X
Registro Implícito para el destino, Inmediato para el fuente IX ←mn	Ld IX, mn donde mn es un dato de 16 bits	4	12	DD 21 n m	×	×	×	×	×	×
Registro Implícito para el destino, Inmediato para la fuente IY ←mn	Ld IY, mn donde mn es un dato de 16 bits	4	12	FD 21 n m	×	×	×	×	×	×
Registro Directo para el destino, Inmediato para el fuente r ←mn	Ld r, mn donde r es alguno de los registros de la tabla 3, y mn es un dato de 16 bits	3	9	1 n m 000001 n m	×	×	×	×	×	×

3.3. Instrucciones Lógicas

And

La operación lógica de conjunción es de 8 bits y utiliza el **modo de** direccionamiento implícito en el destino ya que siempre deja el resultado en el registro acumulador A. El otro operando que interviene en la conjunción puede tener diferentes modos de direccionamiento.

="	~/	£3					h	h		
=		ل.ا		20000	s	Z	Н	P V	N	С
Registro Directo en fuente A ←A & r	And r donde r es alguno de los siguientes registros de 8 bits: B, C, D, E, H, L, A	1	4	A 10100 donde codifica alguno de los registros de 8 bits	Ø	☑	1	N P	0	0
				con la convención de la tabla 1						
Registro Indirecto	And (HL)	1	6	A6	Ø	Ø	1	⊠ P	0	0
A ←A & (HL)								•		
Registro Indexado	And (IX + d)	3	14	DD A6 d	Ø	Ø	1	N P	0	0
A ←A & (IX + d)								Р		
Registro Indexado	And (IY + d)	3	14	FD A6 d	Ø	Ø	1	V	0	0
A ←A & (IY + d)								Р		
Inmediato A ←A & v	And v	2	6	E6 v	\square	V	1	N P	0	0

La operación lógica de disyunción es de 8 bits y utiliza el **modo de direccionamiento implícito en el destino** ya que siempre deja el resultado en el registro acumulador A. El otro operando que interviene en la disyunción puede tener diferentes modos de direccionamiento.

="	~	£;3					h	h		
		ربا			s	Z	Н	P V	N	С
Registro Directo en fuente A ←A r	Or r donde r es alguno de los siguientes registros de 8 bits: B, C, D, E, H, L, A	1	4	B 10110 donde codifica alguno de los registros de 8 bits con la convención de la tabla 1	Ø	\square	0	P	0	0
Registro Indirecto A ←A (HL)	Or (HL)	1	6	В6	Ø	\square	0	☑ P	0	0
Registro Indexado A ←A (IX + d)	Or (IX + d)	3	14	DD B6 d	Ø	☑	0	☑ P	0	0
Registro Indexado A ←A (IY + d)	Or (IY + d)	3	14	FD B6 d	Ø	\square	0	☑ P	0	0
Inmediato A ←A v	Or v	2	6	F6 v	V	\square	0	☑ P	0	0

Xor

La operación lógica de disyunción exclusiva es de 8 bits y utiliza el **modo de direccionamiento implícito en el destino** ya que siempre deja el resultado en el registro acumulador A. El otro operando que interviene en la disyunción exclusiva puede tener diferentes modos de direccionamiento.

="	~	£;3					h	H		
		رجا			s	Z	Н	P V	N	С
Registro Directo en fuente A ←A ⊕ r	XOR r donde r es alguno de los siguientes registros de 8 bits: B, C, D, E, H, L, A	1	4	A 10101 donde codifica alguno de los registros de 8 bits con la convención de la tabla 1	V	\square	0	P	0	0
Registro Indirecto $A \leftarrow \! A \oplus (HL)$	XOR (HL)	1	6	AE	Ø	V	0	Ø P	0	0
Registro Indexado $A \leftarrow A \oplus (IX + d)$	XOR (IX + d)	3	14	DD AE d	Ø	Ø	0	Ø P	0	0
Registro Indexado $A \leftarrow A \oplus (IY + d)$	XOR (IY + d)	3	14	FD AE d	V	Ø	0	Ø P	0	0
Inmediato A ←A ⊕ v	XOR v	2	6	EE v	V	Ø	0	☑ P	0	0

Cpl

Esta instrucción realiza una negación bit a bit del registro acumulador. Es lo que se llama **complemento a 1.**

_•	~	Ł.J					h	b		
=		لہا			S	Z	Н	P V	N	С
Registro Implícito	СрІ	1	3	2F	×	×	1	×	1	×
<u>-</u> A ← A										

Ср

La operación de comparar es de 8 bits y utiliza el *modo de direccionamiento implícito en el destino*. El otro operando que interviene puede tener diferentes modos de direccionamiento.

=	~/	[]					ļ	h		
=		لہا			s	Z	Н	P V	N	С
Registro Directo	Cp r	1	4	В	Ø	V	V	V	1	V
A – r	donde r es alguno de los siguientes registros de 8 bits: B, C, D, E, H, L, A			10111 donde codifica alguno de los registros de 8 bits con la convención de la tabla 1				V		
Registro Indirecto	Cp (HL)	1	6	BE	Ø	Ø	Ø	V	1	V
A - (HL)								٧		
Registro Indexado	Cp (IX + d)	3	14	DD BE d	Ø	\square	Ø	∑ >	1	Ø
A - (IX + d)										
Registro Indexado	Cp (IY + d)	3	14	FD BE d	V		Ø	∑ >	1	☒
A - (IY + d)										
Inmediato A – v	Ср v	2	6	FE v	Ø	☑	Ø	✓	1	☑

3.4. Instrucciones de Salto y de Ciclo

JP (Incondicional)

Con esta instrucción se puede alterar la secuencia de ejecución. Existen diversas instrucciones según el modo de direccionamiento del operando.

="	~/	£,J					h	h		
=		€,3		(2000)	s	Z	Н	P V	N	С
Inmediato	JP rotulo	3	9	C3 n m	×	×	×	×	×	×
PC←mn	donde rótulo es un número mn de 16 bits									
Registro Indirecto	JP (HL)	1	3	E 9	×	×	×	×	×	×
PC←HL										
Registro Indexado	JP (IX)	2	6	DD E9	×	×	×	×	X	×
PC←IX										
Registro Indexado	JP (IY)	2	6	FD E9	×	×	×	×	×	×
PC←IY										
Relativo	JR rotulo	2	8	18 (j-2)	×	×	×	×	×	×
PC←PC + (j-2)										

j: rotulo - dirección de la instrucción

JP (Condicional)

Con esta instrucción se puede cortar la secuencia de ejecución según esté encendido o apagado alguno de los flags del registro F (carry, zero). Existen diversas instrucciones según el modo de direccionamiento del operando.

=		ב יח					ŗ	h		
=		[]	۵	(20000)	s	z	Н	P V	N	С
Inmediato	JP f, rotulo	3	6	n m	×	×	×	×	×	×
Si f es TRUE PC←mn sino CONTINUAR	donde rótulo es un valor mn de 16 bits y f puede ser alguno de los siguiente valores: NZ, Z, C, NC, PO, PE, P, M (Tabla 6)		(9)	11010 n m donde codifica según la tabla 6						
Relativo	JR C, rotulo	2	6	38 (j-2)	×	×	×	×	×	×
Si C = 1 PC←PC + (j -2) sino CONTINUAR			(8)							
Relativo	JR NC, rotulo	2	6	30 (j-2)	×	×	×	×	×	×
si C = 0 PC←PC + (j -2) sino CONTINUAR			(8)							
Relativo	JR Z, rotulo	2	6	28 (j-2)	×	×	×	×	×	×
si Z = 1 PC←PC + (j-2) sino CONTINUAR			(8)							
Relativo	JR NZ, rotulo	2	6	20 (j-2)	×	×	×	×	×	×
si Z = 0 PC←PC + (j-2) sino CONTINUAR			(8)							

j: rotulo - dirección de la instrucción

DJNZ (ciclo)

Con esta instrucción se puede repetir un ciclo la cantidad de veces que se cargue inicialmente en el registro B.

="	~	£; 3					Ţ	h		
=		٦٦		[3000]	S	Z	Н	P V	N	С
Registro Relativo	DJNZ rotulo	5	7	10 (j-2)	×	×	×	×	×	×
B←B-1			(9)							
Luego, si B ≠ 0 PC←PC+(j-2) sino CONTINUAR con próxima instrucción										

j: rotulo - dirección de la instrucción

3.5. Instrucciones de Manipulación de Bits

Shift (Decalaje)

Permiten reubicar los bits moviéndolos hacia izquierda o derecha.

="	Ø	£3			H						
				(Sec.)	s	Z	Н	P V	N	С	
Registro Directo para fuente y destino □ ← □□□□□□ ← 0 c shift left arithmetic	SLA r donde r puede ser alguno de los siguientes registros de 8 bits: A,B, C, D, E, H, L	3	7	CB 2 11001011 00100 donde codifica al registro según la tabla 1	\square	\square	0	D P	0		
Registro Indirecto para fuente y destino ídem al anterior	SLA (HL)	5	13	CB 26	\square	Ø	0	∑ P	0	Ø	
Registro Indexado para fuente y destino ídem al anterior	SLA (IX+d) donde d es un desplazamiento de 8 bits	7	19	DD CB d 26	\square	I	0	D P	0	Ø	
Registro Indexado para fuente y destino ídem al anterior	SLA (IY+d) donde d es un desplazamiento de 8 bits	7	19	FD CB d 26	\square	V	0	D P	0	I	

="	Ø	C			H						
				(Second)	S	Z	Н	P V	N	С	
Registro Directo para fuente y destino → □ □ □ □ C	SRA r donde r puede ser alguno de los siguientes registros de 8 bits: A, B, C, D, E, H, L	3	7	CB 2 11001011 00101 donde codifica al registro según la tabla 1			0	D L	0		
Registro Indirecto para fuente y destino ídem al anterior	SRA (HL)	5	13	CB 2E	☑	Ø	0	P	0		
Registro Indexado para fuente y destino ídem al anterior	SRA (IX+d) donde d es un desplazamiento de 8 bits	7	19	DD CB d 2E	☑	V	0	D P	0	Ø	
Registro Indexado para fuente y destino ídem al anterior	SRA (IY+d) donde d es un desplazamiento de 8 bits	7	19	FD CB d 2E	V	V	0	N P	0		

=	Ø	£ 3			H						
				[FEEDE]	S	Z	Н	P V	N	С	
Registro Directo para fuente y destino 0 → TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT	SRL r donde r puede ser alguno de los siguientes registros de 8 bits: A, B, C, D, E, H, L	3	7	CB 3 11001011 00111 donde codifica al registro según la tabla 1			0	P	0		
Registro Indirecto para fuente y destino ídem al anterior	SRL (HL)	5	13	CB 3E		I	0	∑ P	0	\S	
Registro Indexado para fuente y destino	SRL (IX+d) donde d es un desplazamiento de 8 bits	7	19	DD CB d 3E			0	∑ P	0	\square	
Registro Indexado para fuente y destino ídem al anterior	SRL (IY+d) donde d es un desplazamiento de 8 bits	7	19	FD CB d 3E		I	0	∑ P	0		

Rotación

Permiten parecido al anterior reubicar los bits moviéndolos dentro del operando, pero realizando un tratamiento circular de los mismos.

="	Ø	£3									
					s	Z	Н	P V	N	С	
Registro Implícito para fuente y destino	RLA	1	3	17	×	×	0	P	0		
Registro Directo para fuente y destino ídem al anterior	RL r donde r puede ser alguno de los siguientes registros de 8 bits: A, B, C, D, E, H, L	თ	7	CB 1 11001011 00010 donde codifica al registro según la tabla 1		V	0	P	0	Ŋ	
Registro Indirecto para fuente y destino ídem al anterior	RL (HL)	5	13	CB 16	\square	V	0	☑ P	0		
Registro Indexado para fuente y destino ídem al anterior	RL (IX+d) donde d es un desplazamiento de 8 bits	7	19	DD CB d 16	Ø	Ø	0	☑ P	0		
Registro Indexado para fuente y destino ídem al anterior	RL (IY+d) donde d es un desplazamiento de 8 bits	7	19	FD CB d 16		Ø	0	☑ P	0	Ŋ	

="	~1	ָרָ י					h	b		
[=]		£3		(20000)	s	Z	Н	P V	N	С
Registro Implícito para fuente y destino C	RLCA	1	3	07	×	×	0	P	0	
Registro Directo para fuente y destino ídem al anterior	RLC r donde r puede ser alguno de los siguientes registros de 8 bits: A, B, C, D, E, H, L	3	7	CB 0 11001011 00000 donde codifica al registro según la tabla 1	V	\square	0	□ □	0	
Registro Indirecto para fuente y destino ídem al anterior	RLC (HL)	5	13	CB 06	Ø	Ø	0	P	0	
Registro Indexado para fuente y destino ídem al anterior	RLC (IX+d) donde d es un desplazamiento de 8 bits	7	19	DD CB d 06	Ø	☑	0	∑ P	0	
Registro Indexado para fuente y destino ídem al anterior	RLC (IY+d) donde d es un desplazamiento de 8 bits	7	19	FD CB d 06	Ø	V	0	P	0	

="	~/	[]					h	h		
=		لہا		[EEEE]	s	Z	Н	P V	N	С
Registro Implícito para fuente y destino	RRA	1	3	1F	×	×	0	D P	0	
Registro Directo Para fuente y destino ídem al anterior	RR r donde r puede ser alguno de los siguientes registros de 8 bits: A, B, C, D, E, H, L	3	7	CB 1 11001011 00011 donde codifica al registro según la tabla 1	V	\square	0	N P	0	
Registro Indirecto para fuente y destino ídem al anterior	RR (HL)	5	13	CB 1E	Ø	V	0	D P	0	▼
Registro Indexado para fuente y destino ídem al anterior	RR (IX+d) donde d es un desplazamiento de 8 bits	7	19	DD CB d 1E	Ø	V	0	D P	0	
Registro Indexado para fuente y destino ídem al anterior	RR (IY+d) donde d es un desplazamiento de 8 bits	7	19	FD CB d 1E	V	V	0	Ŋ P	0	\

=	~	[]					h	b		
		لہا		[******	s	Z	Н	P V	N	С
Registro Implícito para fuente y destino	RRCA	1	3	0F	×	×	0	N P	0	
Registro Directo para fuente y destino ídem al anterior	RRC r donde r puede ser alguno de los siguientes registros de 8 bits: A, B, C, D, E, H, L	3	7	CB 0 11001011 00001 donde codifica al registro según la tabla 1	V	\square	0	N P	0	
Registro Indirecto para fuente y destino ídem al anterior	RRC (HL)	5	13	CB 0E	Ø	V	0	∑ P	0	✓
Registro Indexado para fuente y destino ídem al anterior	RRC (IX+d) donde d es un desplazamiento de 8 bits	7	19	DD CB d 0E	Ø	Ø	0	D P	0	N
Registro Indexado para fuente y destino ídem al anterior	RRC (IY+d) donde d es un desplazamiento de 8 bits	7	19	FD CB d 0E	Ø	V	0	Ŋ P	0	▽

="	~	[]					Ļ	b		
		ربا		(50000)	S	Z	Н	P V	N	С
Registro Implícito para fuentes y destinos entre Registro A y (HL)	RLD	8	16	ED 6F	V	Ø	0	∑ P	0	V
Sean A = N_3N_2 y (HL) = N_1N_0 esta operación mueve nibbles dejando A = N_3N_1 y (HL) = N_0N_2										
Registro Implícito para fuentes y destinos entre Registro A y (HL)	RRD	8	16	ED 67	V	\sqrt	0	Б	0	
Sean A = N_3N_2 y (HL) = N_1N_0 esta operación mueve nibbles dejando A = N_3N_0 y (HL) = N_2N_1										

Seteo de Bits

Permite setear o apagar bits dentro del operando.

	~/	€3					r	h		
="		ئہا		20000	s	Z	Н	P V	N	С
Registro Directo para r y Dir. a bit para b bit b de r ← 1	Set b, r donde b es la posición del bit dentro del registro y r puede ser alguno de los siguientes registros de 8 bits: A, B, C, D, E, H, L	3	7	CB 11001011 11 donde las primeras codifican a b y las segundas al registro según la tabla 1	X	×	×	X	×	×
Registro Indirecto para r y Dir. a bit para b bit b de (HL) ← 1	Set b,(HL)	5	13	CB 11001011 11 110 donde codifica a b	×	×	×	×	×	×
Registro Indexado para r y Dir. a bit para b bit b de (IX+d) ← 1	Set b, (IX+d) donde d es un desplazamiento de 8 bits	7	19	11011101 11001011 d 11110 donde codifica a b	×	×	×	×	×	×
Registro Indexado para r y Dir. a bit para b bit b de (IY+d) ← 1	Set b, (IY+d) donde d es un desplazamiento de 8 bits	7	19	FD CB d 11111101 11001011 d 11110 donde codifica a b	×	×	×	×	×	×

		רָּיַ					h	h		
=		£3		(Passes)	s	Z	Н	P V	N	С
Registro Directo para r y Dir. a bit para b bit b de r ← 0	Res b, r donde b es la posición del bit dentro del registro y r puede ser alguno de los siguientes registros de 8 bits: A, B, C, D, E, H, L	3	7	CB 11001011 10 donde las primeras codifican a b y las segundas al registro según la tabla 1	×	×	×	×	×	×
Registro Indirecto para r y Dir. a bit para b bit b de (HL) ← 0	Res b,(HL)	5	13	CB 11001011 10 110 donde codifica a b	×	×	×	×	×	×
Registro Indexado para r y Dir. a bit para b bit b de (IX+d) ← 0	Res b, (IX+d) donde d es un desplazamiento de 8 bits	7	19	11011101 11001011 d 10110 donde codifica a b	×	×	×	×	×	×
Registro Indexado para r y Dir. a bit para b bit b de (IY+d) ← 0	Res b, (IY+d) donde d es un desplazamiento de 8 bits	7	19	FD CB d 11111101 11001011 d 10110 donde codifica a b	×	×	×	×	×	×

3.6. Instrucciones para el Manejo de Stack

Push

Coloca el contenido del registro de 16 bits especificado en el stack y actualiza el **stack pointer**.

="	~/	£,J					ļ	h		
=		£3		2000	s	Z	Н	P V	N	С
Registro Directo para el fuente e Implícito para el destino (SP-1) ← hi r (SP-2) ← lo r SP ← SP -2	PUSH r donde r es alguno de los registros de 16 bits de la tabla 2	5	11	5 110101 donde codifica alguno de los registros de la tabla 2	X	X	X	X	×	×
Registro Implícito para fuente y destino ídem	PUSH IX	6	14	DD E5	×	×	×	×	×	×
Registro Implícito para fuente y destino ídem	PUSH IY	6	14	FD E5	×	×	×	×	×	×

Pop

Saca del stack 16 bits y los deja en el registro de 16 bits especificado, y actualiza el **stack pointer**.

="	~	ťή					h	h		
=		£3		20000	s	z	Н	P V	N	С
Registro Directo para el fuente e Implícito para el destino hi r ← (SP+1) lo r ← (SP) SP ← SP +2	POP r donde r es alguno de los registros de 16 bits de la tabla 2	1	0	110001 donde codifica alguno de los registros de la tabla 2	×	×	×	×	X	×
Registro Implícito para fuente y destino ídem	POP IX	2	12	DD E1	×	×	×	×	×	×
Registro Implícito para fuente y destino ídem	POP IY	2	12	FD E1	×	×	×	×	×	X

Call

Realiza un salto hacia el rótulo especificado, previo salvar la dirección del registro PC (Program Counter) en el stack.

="	~	[;]					h	h		
=		لہا			s	Z	Н	P >	N	С
Inmediato	CALL mn	6	16	CD nm	×	×	×	×	×	×
(SP-1) ← hi PC										
(SP-2) ← lo PC										
PC ← mn										
SP ← SP -2										

Ret

Realiza un salto hacia la dirección que se encuentra en el tope del stack.

="	~	£; 3					h			
		لہا			s	Z	н	P V	N	С
Implícito	RET	3	9	C9	×	×	×	×	×	×
hi PC ← (SP+1)										
lo PC ← (SP)										
SP ← SP + 2										

3.7. Instrucciones Especiales de Control

DAA

Esta instrucción ajusta el acumulador de manera tal que se obtenga la representación numérica BCD.

=	~	ťή					h	h		
=		ربا			s	Z	Н	P V	N	С
Registro Implícito en A para fuente y destino Ajuste Decimal Aritmético	DAA	2	4	27	☑	V	☑	☑ P	×	V

NOP

Esta instrucción no tiene efecto alguno sobre memoria y registros, pero consume tiempo. Se la utiliza en rutinas de manejo de tiempo.

_•	~	t-j					h	h		
틸		لہا		(2000)	s	Z	Н	P V	N	С
Sin operandos No opera	NOP	1	3		×	×	×	×	×	×

4. Cartilla Reducida de Instrucciones (Ordenada por Nombre de Instrucción)

C	-/	۲,		П			F			
Grupo	<u> </u>	£3	۵		S	Z	Н	P V	N	С
	Adc A, r	2	4	8 10001 tabla 1	V		V	∀ >	0	V
	Adc A, (HL)	6	6	8E	\square	V	☑	∑>	0	V
ADC (8 bits)	Adc A, (IX + d)	6	14	DD 8E d	V	V	V	∑>	0	V
	Adc A, (IY + d)	6	14	FD 8E d	V	V	V	∑>	0	V
	Adc A, v	2	6	CE v	V	V	V	∀ ∨	0	V
ADC (16 bits)	Adc HL, r	6	10	EDA 11101101 011010 tabla 3	Ø	☑	?	∀ ∨	0	Ø
	Add A, r	22	4	8 10000 tabla 1	V	\square	V	∑>	0	V
	Add A, (HL)	2	6	86	V	\square	V	∀ ∨	0	I
ADD (8 bits)	Add A, (IX + d)	6	3	DD 86 d	Ø	V	Ø	☑ V	0	Ø
	Add A, (IY + d)	6	3	FD 86 d	V	Ø	V	>	0	Ø
	Add A, v	2	6	C6 v	Ø	V	Ø	∀	0	V

Crupo	~	£3					F			
Grupo	<u> </u>	ب	۵		S	Z	Н	P V	N	С
	Add HL, r	5	7	9	×	×	?	×	0	☑
				001001 tabla 3						
ADD	Add IX, r	6	10	DD9	×	×	?	×	0	☑
(16 bits)				11011101 001001 tabla 4						
	Add IY, r	6	10	FD9	×	×	?	×	0	V
				11111101 00—1001 tabla 5						
	And r	1	4	Α		Ø	1	☑ P	0	0
				10100 tabla 1				Ρ		
	And (HL)	1	6	A 6	Ø	V	1	☑ P	0	0
AND	And (IX + d)	3	14	DD A6 d	Ø	Ø	1	☑ P	0	0
	And (IY + d)	3	14	FD A6 d		Ø	1	☑ P	0	0
	And v	2	6	E6 v	V	V	1	∑ P	0	0
CALL	CALL mn	6	16	CD n m	×	×	×	×	×	×
	Срг	1	4	В	V	V	V	V	1	V
				10111 tabla 1				V		
СР	Cp (HL)	1	6	BE	Ø	Ø	Ø	☑ V	1	V
OF .	Cp (IX + d)	3	14	DD BE d	V	Ø	☑	√ V	1	V
	Cp (IY + d)	3	14	FD BE d	Ø	V	☑	✓ V	1	V
	Ср v	2	6	FE v	Ø	Ø	☑	☑ V	1	☑

Crumo	~/	£3	2				ļ.			
Grupo	Ø.	ڻ.	۵		S	Z	Н	P V	N	С
CPL	СрІ	1	3	2F	×	×	1	×	1	×
DAA	Daa	2	4	27	V			☑ P	×	V
	Dec r	2	4	 00101 tabla 1	I	☑	\	∑ >	1	☑
DEC (8 bits)	Dec (HL)	4	10	35	Ø	Ø	V	☑ V	1	V
	Dec (IX + d)	8	18	DD 35 d	☑	Ø	\square	√ V	1	V
	Dec (IY + d)	8	18	FD 35 d	Ø	Ø	V	∀	1	☑
DEC	Dec r	2	4	B 001011 tabla 3	×	×	×	×	×	×
(16 bits)	Dec IX	3	7	DD 2B	×	×	×	×	×	×
	Dec IY	3	7	FD 2B	×	×	×	×	×	×
DJNZ	Djnz rotulo	5	7 (9)	10 (j-2)	×	×	×	×	×	×
	Inc r	2	4	 00100 tabla 1	V	V		\\ >	0	\square
INC (8 bits)	Inc (HL)	4	10	34	V	V	N	√ >	0	v
	Inc (IX + d)	8	18	DD 34 d	V	Ø	N	>	0	V
	Inc (IY + d)	8	18	FD 34 d	V	7	☑	√ V	0	Ø

C******		£;3					F			
Grupo	Ø.	ئہا	۵		S	Z	Н	P V	N	С
	Inc r	2	4	3	×	×	×	×	×	×
				000011 tabla 3						
INC (16 bits)	Inc IX	3	7	DD 23	×	×	×	×	×	×
	Inc IY	3	7	FD 23	×	×	×	×	×	×
	JP rotulo	3	9	C3 n m	×	×	×	×	×	X
	JP (HL)	1	3	E9	×	×	×	×	×	×
JP	JP (IX)	2	6	DD E9	×	×	×	×	×	×
	JP (IY)	2	6	FD E9	×	×	×	×	×	×
	JP f, rotulo	3	6	n m	×	×	×	×	×	×
	f es valor de tabla 6		(9)	11010 n m tabla 6						
	JR rotulo	2	8	18 (j-2)	×	×	×	×	×	×
	JR C, rotulo	2	6 (8)	38 (j-2)	×	×	×	×	×	×
JR	JR NC, rotulo	2	6 (8)	30 (j-2)	×	×	×	×	×	×
	JR Z, rotulo	2	6 (8)	28 (j-2)	×	×	×	×	×	×
	JR NZ, rotulo	2	6 (8)	20 (j-2)	×	×	×	×	×	×

				-			F	b		
Grupo	Ø	€3	2		s	Z	Н	P V	N	С
	Ld A, I	2	6	ED 57	V	Ø	0	×	0	×
	Ld A, R	2	6	ED 5F	Ø	Ø	0	×	0	×
	Ld I, A	2	6	ED 47	×	×	×	×	×	×
	Ld R, A	2	6	ED 4F	×	×	×	×	×	×
	Ld A, (BC)	2	6	0A	×	×	×	×	×	×
	Ld A, (DE)	2	6	1A	×	×	×	×	×	×
LD (8 bits)	Ld (BC), A	6	7	02	×	×	×	×	×	×
	Ld (DE), A	3	7	12	×	×	×	×	×	×
	Ld r, r'	2	4	 01 r en tabla 1 r' en tabla 1	×	×	×	×	×	×
	Ld A, (mn)	4	12	3A n m	×	×	×	×	×	×
	Ld (mn), A	5	13	32 n m	×	×	×	×	×	×
	Ld r, (HL)	2	6	 01110 tabla 1	×	×	×	×	×	×
continúa ↓	Ld (HL), r	3	7	7 01110 tabla 1	×	×	×	×	×	×

0		ر ب					h	b		
Grupo	Ø.	€3	۵		S	Z	Н	P V	N	С
	Ld r, (IX+d)	6	14	DD d 11011101 01110 d tabla 1	×	×	×	×	×	×
	Ld r, (IY+d)	6	14	FD d 11111101 01110 d la tabla 1	×	×	×	×	×	×
	Ld (IX+d), r	7	15	DD 7 d 11011101 01110 d la tabla 1	×	×	×	×	×	×
LD (8 bits)	Ld (IY+d), r	7	15	FD 7 d 11111101 01110 d tabla 1	×	×	×	×	×	×
	Ld r, v	2	6	v 00110 v tabla 1	×	×	×	×	×	×
	Ld (HL), v	3	9	36 v	×	×	×	×	×	×
	Ld (IX+d), v	5	15	DD 36 d v	×	×	×	×	×	×
	Ld (IY+d), v	5	15	FD 36 d v	×	×	×	×	×	×

			0				r	b		
Grupo	K	€3			s	Z	Н	P V	N	С
	Ld SP, HL	2	4	F9	×	×	×	×	×	×
	Ld SP, IX	3	7	DD F9	×	×	×	×	×	×
	Ld SP, IY	3	7	FD F9	×	×	×	×	×	×
	Ld HL, (mn)	3	15	2A n m	×	×	×	×	×	×
	Ld IX, (mn)	6	18	DD 2A n m	×	×	×	×	×	×
	Ld IY, (mn)	6	18	FD 2A n m	×	×	×	×	×	×
	Ld (mn), HL	6	16	22 n m	×	×	×	×	×	×
	Ld (mn), IX	7	19	DD 22 n m	×	×	×	×	×	×
LD (16 bits)	Ld (mn), IY	7	19	FD 22 n m	×	×	×	×	×	×
	Ld r, (mn)	6	9	EDB n m	×	×	×	×	×	×
				11101101 011011 n m						
	Ld (mn), r	7	19	ED3 n m	×	×	×	×	×	×
				11101101 010011 n m						
	Ld IX, mn	4	12	DD 21 n m	×	×	×	×	×	×
	Ld IY, mn	4	12	FD 21 n m	×	×	×	×	×	×
	Ld r, mn	3	9	1 n m	×	×	×	×	×	×
				000001 n m tabla 3						

			0				<u></u>	b		
Grupo	Ø.	€3			s	Z	Н	P V	N	С
	Mlt r	13	17	EDC	×	×	×	×	×	×
MLT				11101101 011100 tabla 3						
NEG	Neg	2	6	ED 44	\square	Ø	\square	>	1	\square
NOP	NOP	1	3	00	×	×	×	×	×	×
	Orr	1	4	B 10110 tabla 1	V	V	0	IDР	0	0
	Or (HL)	1	6	В6	V	V	0	ID	0	0
OR	Or (IX + d)	3	14	DD B6 d	V	V	0	IDР	0	0
	Or (IY + d)	3	14	FD B6 d	V	V	0	ŊΩ	0	0
	Or v	2	6	F6 v	V	V	0	ŊΩ	0	0
	POP r	1	9	 110001 tabla 2	×	×	×	X	×	×
POP	POP IX	2	12	DD E1	×	×	×	×	×	×
	POP IY	2	12	FD E1	×	×	×	×	×	×
	PUSH r	5	11	5	×	×	×	×	×	×
				110101 tabla 2						
PUSH	PUSH IX	6	14	DD E5	×	×	×	×	×	×
	PUSH IY	6	14	FD E5	×	×	×	×	×	×

0		۲۰					F	b		
Grupo	Ø.	€3			S	Z	Н	P V	N	С
	Res b, (HL)	5	13	СВ	×	×	×	×	×	×
	b es lposición del bit			11001011 10 110						
	Res b, (IX+d)	7	19	DD CB d	×	×	×	×	×	×
	b es posición del bit			11011101 11001011 d 10110						
RES	Res b, (IY+d)	7	19	FD CB d	×	×	×	×	×	×
	b es posición del bit			11111101 11001011 d						
	Doo h r	2	7	10110 CB	E I	[P	[F]	[F]	E I	u u
	Res b, r b es posición del bit	3	7	11001011 10 primero b segundo tabla 1	×	×	X	×	×	×
RET	RET	3	9	С9	×	×	×	×	×	×
	RLA	1	3	17	×	×	0	☑ P	0	V
	RL r	3	7	CB 1	\square	Ø	0	☑ P	0	V
				11001011 00010 tabla 1				Г		
Rotation	RL (HL)	5	13	CB 16	V	Ø	0	☑ P	0	V
	RL (IX+d)	7	19	DD CB d 16	V	V	0	☑ P	0	V
	RL (IY+d)	7	19	FD CB d 16	V	Ø	0	☑ P	0	V
oontin/-	RLCA	1	3	07	×	×	0	☑ P	0	V
continúa ↓	RLC (HL)	5	13	CB 06	Ø	V	0	☑ P	0	Ø

			0										
Grupo	<u> </u>	€3			S	Z	Н	P V	N	С			
	RLC r	3	7	CB 0 11001011 00000 tabla 1	\square	\square	0		0	Ø			
	RLC (IX+d)	7	19	DD CB d 06	Ø	Ø	0		0	V			
	RLC (IY+d)	7	19	FD CB d 06	Ø	Ø	0		0	Ø			
	RLD	8	16	ED 6F	Ø	Ø	0	☑ P	0	Ø			
	RRA	1	3	1F	×	×	0	☑ P	0	Ø			
	RR r	3	7	CB 1 11001011 00011 tabla 1	Ø	Ø	0	Ø P	0	Ø			
Rotation	RR (HL)	5	13	CB 1E	Ø	Ø	0	☑ P	0	Ø			
Notation	RR (IX+d)	7	19	DD CB d 1E	V	Ø	0	☑ P	0	V			
	RR (IY+d)	7	19	FD CB d 1E	Ø	Ø	0	☑ P	0	Ø			
	RRCA	1	3	0F	×	×	0	☑ P	0	Ø			
	RRC r	3	7	CB 0 11001011 00001 tabla 1	\square	Ø	0	Ø P	0	V			
	RRC (HL)	5	13	CB 0E	V	Ø	0	☑ P	0	Ø			
	RRC (IX+d)	7	19	DD CB d 0E	Ø	Ø	0	☑ P	0	Ø			
	RRC (IY+d)	7	19	FD CB d 0E	V	V	0	☑ P	0	V			
	RRD	8	16	ED 67	V	V	0	☑ P	0	V			

0							F	b		
Grupo	Ø.		8		s	Z	н	P V	N	С
	Set b, r	3	7	СВ	×	×	×	×	×	×
	b es posición del bit			11001011 11 tabla 1						
	Set b,(HL)	5	13	СВ	×	×	×	×	×	×
	b es posición del bit			11001011 11 110						
SET	Set b, (IX+d)	7	19	DD CB d	×	×	×	×	×	×
	b es posición del bit			11011101 11001011 d 11110						
	Set b, (IY+d)	7	19	FD CB d	×	×	×	×	×	×
	b es posición del bit			11111101 11001011 d						
				11110						
	Sbc A, r	2	4	9 10011 tabla 1	✓		V	✓	1	
	Sbc A, (HL)	2	6	9E	Ø	V	V	V	1	\(\)
SBC (8 bits)	Sbc A, (IX + d)	6	14	DD 9E d	V	V	Ø	V	1	<u>S</u>
	Sbc A, (IY + d)	6	14	FD 9E d	V	V	V	✓	1	<u>S</u>
	Sbc A, v	2	6	DE v	V	V	Ø	V	1	Ø
SBC	Sbc HL, r	6	10	ED2	V	Ø	?	☑ V	1	V
(16 bits)				11101101 010010 tabla 3				V		

			0				F	b		
Grupo	Ø.	€3	2		s	Z	Н	P V	N	С
	SLA r	3	7	CB 2 11001011 00100 tabla 1	Ø	Ø	0	Ø P	0	Ø
	SLA (HL)	5	13	CB 26	V	Ø	0	☑ P	0	V
	SLA (IX+d)	7	19	DD CB d 26	V	Ø	0	☑ P	0	V
	SLA (IY+d)	7	19	FD CB d 26	V	Ø	0	☑ P	0	V
	SRA r	3	7	CB 2 11001011 00101 tabla 1	Ø	I	0	Ø P	0	Q
Shift	SRA (HL)	5	13	CB 2E	V	Ø	0	☑ P	0	V
	SRA (IX+d)	7	19	DD CB d 2E	V	Ø	0	☑ P	0	V
	SRA (IY+d)	7	19	FD CB d 2E	V	Ø	0	☑ P	0	V
	SRL r	3	7	CB 3 11001011 00111 tabla 1	Ø	☑	0	Ø P	0	☑
	SRL (HL)	5	13	CB 3E	V	Ø	0	☑ P	0	V
	SRL (IX+d)	7	19	DD CB d 3E	V	Ø	0	☑ P	0	V
	SRL (IY+d)	7	19	FD CB d 3E	V	Ø	0	☑ P	0	Ø

Grupo	Æ	[]	8		Pri l'alla					
					S	Z	Н	P V	N	С
SUB (8 bits)	Sub r	2	4	9	V	V	V	>	1	S
				tabla 1						
	Sub (HL)	2	6	96	V	\square	\square	∑>	1	\square
	Sub (IX + d)	6	14	DD 96 d	V	V	V	V >	1	V
	Sub (IY + d)	6	14	FD 96 d	V	V	V	∀ >	1	V
	Sub v	2	6	D6 v	Ø	☑	Ø	☑ V	1	\square
XOR	XOR r	1	4	A 10101 tabla 1	V	\square	0	☑ P	0	0
	XOR (HL)	1	6	AE	V	\square	0	☑ P	0	0
	XOR (IX + d)	3	14	DD AE d	V	\square	0	☑ P	0	0
	XOR (IY + d)	3	14	FD AE d	V	Ø	0	☑ P	0	0
	XOR v	2	6	EE v	v	Ø	0	☑ P	0	0