

DESENSAMBLADOR EN BASIC PARA EL MICROPROCESADOR Z-80

M.A. MURRAY-LASSO*

* PROFESOR DE LA SECCION DE CONTROL DE LA DIVISION DE
ESTUDIOS DE POSGRADO DE LA FACULTAD DE INGENIERIA
U. N. A. M.



F
DEPR 2
1-1
10-0
ES. 7

INTRODUCCION

Al principio del uso de las computadoras digitales electrónicas, éstas se tenían que programar en lenguaje de máquina, lo cual resultaba sumamente tedioso y estaba sujeto a errores. Pronto aparecieron los lenguajes ensambladores que sustituían cadenas de ceros y unos por líneas con mnemónicos mucho más fáciles de entender y manejar. Con la disponibilidad de los ensambladores, los programadores escribían sus programas en dichos lenguajes y éstos se encargaban de traducirlos a las cadenas de ceros y unos correspondientes al lenguaje de la máquina.

Hay ocasiones en que es de interés el proceso contrario, es decir, se cuenta con un programa en lenguaje de máquina y se desea obtener el mismo escrito en lenguaje ensamblador. A un programa que logre esta traducción se le llama un desensamblador. Una de las aplicaciones posibles de un desensamblador es la revisión de un programa que fue introducido a la memoria usando instrucciones POKE o la decodificación de una memoria-sólo-para-lectura (ROM; Read Only Memory) que contiene, entre otras cosas, subrutinas de entrada-salida y un intérprete como BASIC escrito en lenguaje de máquina.

En este artículo presentamos un programa escrito en BASIC el cual opera sobre un vector de números hexadecimales de dos cifras cada uno y produce el listado de un programa en el ensamblador del microprocesador Z-80 correspondiente al vector de números hexadecimales. Cuando el vector de números hexadecimales no representa un programa, el desensamblador señala el primer byte en error.

El proceso de desensamblado se puede llevar a cabo a mano con ayuda de una lista en orden lexicográfico de las instrucciones de la máquina. Se trata de un proceso simple pero laborioso. El desensamblador automatiza este proceso eliminando el tedio.

METODOLOGIA DEL DESENSAMBLADOR

Para desensamblar un programa escrito en lenguaje de máquina para el microprocesador Z-80 hemos comenzado por clasificar las instrucciones del Z-80 en varios grupos. En la fig 1 se muestra un listado en orden lexicográfico de las instrucciones del Z-80 (ref 2). En la Tabla I se muestra una lista de variables y en la fig 4 los contenidos de los vectores OB\$,SS\$ y TA. Tomando la columna titulada OBJ CODE en la fig 1 como llave distinguimos los siguientes grupos:

- Grupo 1: Instrucciones que comienzan con CB
- Grupo 2: Instrucciones de un byte que están entre 40 y BF
- Grupo 3: Instrucciones que comienzan con FDCB o con DDCB
- Grupo 4: Instrucciones cuyo primer byte está entre 00 y 3F
- Grupo 5: Instrucciones cuyo primer byte es DD y cuyo segundo byte no es CB
- Grupo 6: Instrucciones cuyo primer byte es ED
- Grupo 7: Instrucciones cuyo primer byte es FD y cuyo segundo byte no es CB
- Grupo 8: Instrucciones cuyo primer byte está entre C0 y FF y que no pertenecen a ningún grupo anterior

El ensamblador toma un byte e intenta encontrar a qué grupo pertenece dicho byte ensayando los grupos en el orden dado arriba. Una vez encontrado el grupo, hace un análisis especial para cada uno de ellos, mismo que a continuación detallamos.

DETALLES DEL PROCESO DESENSAMBLADOR

Grupo 1. Se lee un segundo byte (el primero vale CB). Si el segundo byte está entre "38" y "FF" se transforma el segundo byte a octal con tres cifras significativas (poniendo ceros a la izquierda en caso necesario). Si las dos cifras más significativas de este número octal son "07", se escribe "SRL"; si son 1i con i entre 0 y 7, se escribe "BIT i," si son 2i con i entre 0 y 7, se escribe "RES i"; y si son 3i con i entre 0 y 7, se escribe

"SET i". Además se ve el valor del dígito menos significativo del número octal y se agrega después de la coma: B,C,D,E,H,L,(HL),A según dicho dígito valga 0,1,2,3,4,5,6,7 respectivamente.

Si al leer el segundo byte éste está entre 00 y 2F se transforma el segundo byte a octal con dos cifras significativas ji y se escribe RLC,RRC,RL,RR,SLA,SRA según j valga 0,1,2,3,4,5, respectivamente, seguido de un espacio y seguido de B,C,D,E,H,L,(HL),A según i valga 0,1,2,3,4,5,6,7 respectivamente.

En el listado de la fig 2 el desensamblado de las instrucciones del grupo 1 se logra con las instrucciones 110 a 140.

Grupo 2. Si la instrucción es 76 se escribe HALT. En caso contrario, se convierte el byte de la instrucción a octal con tres cifras. Si las dos cifras más significativas son "li" con i entre 0 y 7 se escribe "LDx," con x representando "B","C","D","E","H","L","(HL)","A" según i valga 0,1,2,3,4,5,6,7 respectivamente. Si las dos cifras más significativas son "20" o "21" se escribe "ADD A","ADC A," respectivamente. Si son "22" o "23" se escribe "SUB A" o "SBC A," respectivamente. Si son "24","25","26" o "27" se escribe "AND","XOR","OR" o "CP" respectivamente. Si son "24","25","26" o "27" se escribe "AND","XOR","OR" o "CP" respectivamente.

Para aquellas instrucciones cuyos dos dígitos más significativos están entre "20" y "27" se imprime un espacio vacío,

Finalmente, se escribe "B","C","D","E","H","L","(HL)","A" según el dígito menos significativo del número octal valga 0,1,2,3,4,5,6,7 respectivamente.

El desensamblado de las instrucciones del grupo 2 se logra en las líneas 160 a 200 del listado de la fig 2.

Grupo 3. Todas las instrucciones de este grupo son de 4 bytes. Se reconocen 3 casos de acuerdo al rango de valores del segundo byte menos significativo.

Caso 1: el byte vale "3E"

Caso 2: el byte está entre "06" y "2E"

Caso 3: el byte está entre "3E" y "FE"

En el primer caso se imprime "SRL(h+g)" donde h es IX o IY, dependiendo si el byte más significativo de la instrucción es "DD" o "FD" respectivamente, y g es el segundo byte menos significativo de la instrucción.

Para el segundo caso se imprime "RLC(h+g)", "RRC(h+g)", "RL(h+g)", "RR(h+g)", "SLA(h+g)", "SRA(h+g)" dependiendo si el byte menos significativo de la instrucción vale 06,0E, 16,1E,26 o 2E respectivamente; g es el segundo byte menos significativo y h es IX o IY si el byte más significativo de la instrucción es "DD" o "FD".

Para el caso 3 imprimimos "BIT i, (h+g)", "RES i, (h+g)", "SET i, (h+g)" dependiendo si la cifra hexadecimal más significativa del byte menos significativo de la instrucción está en los rangos (4-7), (8-B), (C-F); donde g es el valor del segundo byte menos significativo; h es IX o IY si el byte más significativo es DD o FD respectivamente; i se calcula con la fórmula $i = H - B * 8$; $B = \text{INT}(H/8)$; $H = (K-4) * 2$ si la última cifra hexadecimal de la instrucción es 6 o $H = (K-4) * 2 + 1$ si la última cifra es E, donde K es valor de la cifra hexadecimal más significativa del byte menos significativo de la instrucción.

El desensamblado de las instrucciones del grupo 3 se logra con las líneas 210 a 269 del listado de la fig 2.

Grupo 4. El desensamblado de las instrucciones de este grupo se logra aplicando una búsqueda binaria en las primeras 64 componentes del vector OB\$ que contienen los códigos de las instrucciones ordenadas lexicográficamente e imprimiendo la instrucción desensamblada correspondiente que está en un vector auxiliar SS\$. Distinguimos 4 casos diferentes.

Caso 1: Instrucción de 1 byte con desensamble fijo

Caso 2: Instrucción de 2 bytes con el byte menos significativo variable que se escribe al final

Caso 3: Instrucción de 3 bytes con los 2 bytes menos significa-

tivos variables que se escriben al final

Caso 4: Instrucciones de 4 bytes con 2 bytes menos significativos variables que no se escriben al final

Para las instrucciones del caso 1 simplemente se imprime la instrucción desensamblada fija.

Para las instrucciones del caso 2 se escribe la parte fija de la instrucción, la cual termina en una coma seguida por el byte menos significativo de la instrucción.

Para las instrucciones del caso 3 se escribe la parte fija de la instrucción, la cual termina en una coma seguida por el byte menos significativo y después por el segundo byte menos significativo de la instrucción.

Las instrucciones del caso 4 se manejan en forma específica cada una. (Ver líneas 330 a 360 del listado de la fig 2). Para distinguir los casos se tiene un vector auxiliar TA cuyos componentes son el número del caso en cuestión para el 1, 2 y 3. Para el caso 4 se utiliza el valor 8.

El desensamblado de las instrucciones del grupo 4 se logra con las líneas 270 a 370 del listado de la fig 2. En esas líneas se llama a la subrutina de búsqueda binaria que está en las líneas 1040 a 1048 del mismo listado.

Grupo 5. El desensamblado de las instrucciones de este grupo también se logra aplicando una búsqueda binaria en las componentes 125 a 163 del vector OB\$ e imprimiendo la instrucción desensamblada correspondiente del vector SS\$. Se distinguen 4 casos.

Caso 1: Instrucciones de 2 bytes con desensamble fijo

Caso 2: Instrucciones de 3 bytes con el byte menos significativo variable, el cual va al final de la instrucción desensamblada y termina con el símbolo ")]"

Caso 3: Instrucciones de 3 bytes con el byte menos significativo

variable pero que no va al final de la instrucción desensamblada.

Caso 4: Instrucciones de 4 bytes con 2 de ellos variables.

Para el caso 1 simplemente se imprime la instrucción desensamblada. Estos casos están marcados en el vector auxiliar TA con 4.

Para el caso 2 se imprime la parte fija de la instrucción desensamblada, después el byte menos significativo de la instrucción seguido del símbolo ")). Estos casos están marcados con 5 en el vector auxiliar TA.

El caso 3 se maneja en forma específica para cada instancia. (Ver líneas 430 a 470 en el listado de la fig 2).

En el caso 4 las instrucciones desensambladas son del tipo LD(IX+g), i donde g es el byte menos significativo de la instrucción. El símbolo i es el que se guarda en el vector SS\$ y se localiza por medio de la búsqueda binaria.

El desensamblado de las instrucciones del grupo 5 se logra con las líneas 380 a 480 del listado de la fig 2. En esas líneas se llama a la subrutina de búsqueda binaria de las líneas 1050 a 1058 del mismo listado.

Grupo 6. El desensamblado de las instrucciones de este grupo se logra por medio de una búsqueda binaria en las componentes 164 a 217 del vector OB\$ con ayuda del vector auxiliar SS\$ en forma análoga a las de los grupos 4 y 5. Se distinguen 2 casos:

Caso 1: Instrucciones de 4 bytes

Caso 2: Instrucciones de 2 bytes con desensamblado fijo

Las instrucciones del caso 1 se manejan en forma específica. Ver líneas 580 a 630 del listado de la fig 2.

El caso 2 se maneja simplemente imprimiendo el desensamblado fijo que se localiza por medio de la búsqueda binaria. Las instrucciones de este caso son todas aquellas que no pertenecen al caso 1 y fueron encontradas en la búsqueda binaria.

El desensamblado de las instrucciones de este grupo se logra con las líneas 540 a 640 del listado de la fig 2. En dichas líneas se llama a la subrutina de búsqueda binaria de las líneas 1050 a 1058 del mismo listado.

Grupo 7. El desensamblado de las instrucciones de este grupo se logra por medio de una búsqueda binaria en las componentes 218 a 256 del vector OB\$ con ayuda del vector auxiliar SS\$ en forma análoga a la de los grupos 4 a 6. Se distinguen 4 casos:

Caso 1: Instrucciones de 2 bytes con desensamblado fijo

Caso 2: Instrucciones de 3 bytes con el byte menos significativo variable que va al final de la instrucción desensamblada seguido del símbolo ")"

Caso 3: Instrucciones de 4 bytes

Caso 4: Instrucciones de 3 bytes con el byte menos significativo variable que no va al final de la instrucción desensamblada.

Las instrucciones del caso 1 están marcadas en el vector auxiliar TA con 4. Simplemente se imprime su desensamblado fijo.

Las instrucciones del caso 2 están marcadas en el vector auxiliar TA con 5. Se imprime la parte fija del desensamblado seguido del byte menos significativo de la instrucción y del símbolo "B".

Las instrucciones del caso 3 se manejan en forma específica. Ver líneas 700 a 730 del listado de la fig 2.

Las instrucciones del caso 4 están marcadas con 8 en el vector auxiliar TA. Son del tipo LD(IY+g), i donde g es el byte menos

significativo de la instrucción. El símbolo *i* es el que se guarda en el vector SS\$ y el valor adecuado se localiza por búsqueda binaria.

Las instrucciones del grupo 7 se desensamblan con las líneas 650 a 750 del listado de la fig 2. Se llama a la subrutina de búsqueda binaria de las líneas 1050 a 1058.

Grupo 8. El desensamblado de las instrucciones de este grupo se logra también por búsqueda binaria en las componentes 65 a 124 del vector OB\$ con ayuda del vector auxiliar SS\$ como en los grupos 4 a 7. Se distinguen 4 casos:

- Caso 1: Instrucciones de un byte con desensamblado fijo
- Caso 2: Instrucciones de 2 bytes con el byte menos significativo variable, el cual va al final de la instrucción desensamblada
- Caso 3: Instrucciones de tres bytes con los dos bytes menos significativos variables, los cuales van al final de la instrucción desensamblada
- Caso 4: Instrucciones de dos bytes con uno variable que no va al final de la instrucción desensamblada

El manejo de las instrucciones del caso 1 se hace simplemente imprimiendo la instrucción desensamblada fija del vector SS\$.

Las instrucciones del caso 2 se manejan imprimiendo la parte fija de la instrucción obtenida del vector SS\$ seguida del byte menos significativo de la instrucción.

Las instrucciones del caso 3 se manejan imprimiendo la parte fija de la instrucción obtenida del vector SS\$, seguida del byte menos significativo y seguido del segundo byte menos significativo de la instrucción.

Sólo hay una instrucción del caso 4; ésta se maneja en forma específica. Ver línea 810 del listado de la fig 2.

El desensamblado de las instrucciones del grupo 8 se logra con las líneas 760 a 830 del listado de la fig 2. Se llama la subrutina de búsqueda binaria que está en las líneas 1040 a 1048.

COMENTARIOS ADICIONALES SOBRE EL DESENSAMBLADOR

La parte esencial del desensamblador ha sido comentada en la sección anterior. Aquí solamente agregaremos algunos comentarios adicionales para completar la discusión.

En la línea 10 se reserva memoria para 2000 caracteres alfanuméricos que se manejan en variables y arreglos alfanuméricos y se limpia la pantalla.

Todas las variables numéricas se declaran enteras en la línea 15*. En la línea 20 se dimensionan los arreglos alfanuméricos del programa.

En la línea 21 se introducen, de las instrucciones DATA (líneas 501 a 539) las componentes 125 a 163 (requeridas por las instrucciones del grupo 5) a los vectores OB\$, TA y SS\$.

En las líneas 22 a 26, se introducen de DATA (línea 2000) las componentes del vector Y\$,W\$,y V\$ quedando estos vectores como sigue:

	0	1	2	3	4	5	6	7	
Y\$=	B	C	D	E	H	L	(HL)	A	
	1	2	3	4	5	6	7	8	9
W\$=	RLC	RRC	RL	RR	SLA	SRA	SRL	BIT	RES
									SET
	0	1	2	3	4	5	6	7	8
V\$=	LD	ADD	ADC	SUB	SBC	AND	XOR	OR	CP

* Sólo las variables que empiezan con B y con S no se declaran enteras debido al reducido rango de las mismas para que la modificación de la fig 3 funcione correctamente.

En las líneas 42 a 46 se introducen de las instrucciones DATA (líneas 2193 a 2252) las componentes 65 a 124 (requeridas por las instrucciones del grupo 8) a los vectores OB\$, TA y SS\$.

En la línea 47 se define el vector Z\$ que se utiliza para calcular el valor de un byte o una cifra hexadecimal en decimal en varias partes del programa.

En la línea 48 se introducen, de las instrucciones DATA (líneas 2571 a 2624), las componentes 164 a 217 (requeridas por las instrucciones del grupo 6) a los vectores OB\$ y SS\$. (Nótese que en vista de que sólo hay 2 casos en el grupo 6 y que el primer caso se trata específicamente, estas componentes no requieren marcas en el vector TA).

En la línea 59 se lee la longitud en bytes L del programa que se quiere desensamblar. En las líneas 60 a 80 se introduce, de las instrucciones DATA, el programa que se va a desensamblar guardándolo en el vector ST\$.

En la línea 100 se inicia el proceso de desensamblado poniendo el apuntador I en el primer byte del programa que se va a desensamblar. Después de desensamblar cada instrucción y de actualizar el apuntador I, el programa regresará a la línea 110 donde se empieza a localizar el grupo al que pertenece la instrucción bajo proceso.

Los datos del programa se escriben en instrucciones DATA numeradas de 2665 en adelante. El primer dato debe ser el número de bytes que tiene el programa y los demás los bytes del programa mismo.

En las líneas 1030 a 1038; 1040 a 1048 y 1050 a 1058, aparecen tres subrutinas para hacer búsqueda binaria en el arreglo OB\$. La primera es para buscar una cadena X\$ que coincida en todos sus símbolos con alguna componente de OB\$ entre la componente II y la FF. La segunda busca que la cadena de símbolos X\$ coin-

cida en los 2 primeros símbolos con alguna componente del vector OB\$ ente la componente II y la FF. La tercera busca que la cadena de símbolos X\$ coincida en los 4 primeros símbolos con alguna componente del vector OB\$ entre la componente II y la FF. En las tres subrutinas, al entrar, se hace la bandera 0 igual a cero. En caso de éxito en la búsqueda, al regreso de la subrutina, se tiene 0=1 y en X se tiene el índice del vector OB\$ de la componente que coincidió con la cadena buscada. En caso de no encontrarse, la subrutina regresa la bandera 0 con el valor cero.

EJEMPLO

El siguiente programa de 26 bytes está tomado de la ref 1.
CD,0E,45,3E,3D,83,5F,30,01,24,CD,0E,45,C9,01,03,00,ED,B0,C9,A0,
AB,A1,8A,83,8B.

Se introdujo a la computadora, por medio de las líneas DATA,
2900 y 2910 siguientes:

2900 DATA 26

2910 DATA CD,0E,45,3E,3D,83,5F,30,01,24,CE,0E,45,C9,01,03,00,ED,
B0,C9,A0,AB,A1,8A,83,8B.

El programa desensamblado producido por el desensamblador es el siguiente:

CALL 45 0E	AND B
LD A,3D	XOR E
ADD A,E	AND C
LDE,A	ADC A,D
JR NC,01	ADD A,E
INC H	ADC A,E
CALL 45 0E	BREAK IN 820
RET	READY
LD BC,0003	
LDIR	
RET	

MODIFICACIONES PARA DESENSAMBLAR PORCIONES DE MEMORIA INCLUYENDO EL ROM

Cuando ya se tiene escrita una porción de la memoria con un programa, éste se puede leer directamente de la memoria en vez de introducirlo al desensamblador tecleándolo en instrucciones DATA. En la fig 3 aparecen las modificaciones que hay que hacerle al desensamblador para desensamblar un programa que está en posiciones contiguas de la memoria del byte BI al BF (dados en notación hexadecimal). La subrutina en las líneas 850 a 870 convierte una cadena de símbolos a decimal. La subrutina en las líneas 900 a 918 convierte un número decimal a una cadena de símbolos que representan al número en notación hexadecimal.

Para ejemplificar el uso del desensamblador para desensamblar un pedazo del ROM de la Radio Shack TRS-80 mostramos a continuación lo producido en la pantalla al desensamblar lo contenido en la memoria entre las localidades 0033 y 0043 (ambos números en hexadecimal).

RUN

LOCALIDAD EN HEXADECIMAL INICIAL Y FINAL? 0033,0043

PROGRAMA OBJETO

11 1D 40 18 E3 C3 12 40 11 25 40 18 DB C3

D9 05 C9

PROGRAMA FUENTE

LD DE, 401D

JR E3

JP 4012

LD DE, 4025

JR DB

JP 05 D9

RET

BREAK IN 820

READY

CONCLUSION

Se ha presentado un desensamblador para el microprocesador Z-80 escrito en BASIC. Dicho desensamblador es una herramienta útil en el desarrollo de programas de máquina, los cuales son laboriosos de manejar a mano. La ventaja de que el desensamblador esté en BASIC es que se puede correr en cualquier máquina que tenga este lenguaje. También se presentaron modificaciones para desensamblar directamente el contenido de un segmento de memoria. El desensamblador ha sido instalado y probado en una computadora Radio Shack TRS-80 en la cual se ejecutaron los ejemplos y se obtuvieron los listados mostrados en este artículo.

REFERENCIAS

1. George H. Yeager, "Exploring TRS-80 Graphics", Byte, Agosto 1979, pp 82-84.
2. Radio Shack TRS-80 EDITOR/ASSEMBLER User Instruction Manual, Tandy Corporation, Fort Worth, TX., 1978

07/09/76 10:20:50		OPCODE LISTING					
LOC	OBJ CODE	STMT	SOURCE STATEMENT	LOC	OBJ CODE	STMT	SOURCE STATEMENT
0000	00	1	NOP	0065	47	72	LD B,A
0001	018405	2	LD BC,NN	0066	48	73	LD C,B
0004	02	3	LD (BC),A	0067	49	74	LD C,C
0005	03	4	INC BC	0068	4A	75	LD C,D
0006	04	5	INC B	0069	4B	76	LD C,E
0007	05	6	DEC B	006A	4C	77	LD C,H
0008	0620	7	LD B,N	006B	4D	78	LD C,L
000A	07	8	RLCA	006C	4E	79	LD C,(HL)
000B	08	9	EX AF,AF'	006D	4F	80	LD C,A
000C	09	10	ADD HL,BC	006E	50	81	LD D,B
000D	0A	11	LD A,(BC)	006F	51	82	LD D,C
000E	0B	12	DEC BC	0070	52	83	LD D,D
000F	0C	13	INC C	0071	53	84	LD D,E
0010	0D	14	DEC C	0072	54	85	LD D,H
0011	0E20	15	LD C,N	0073	55	86	LD D,L
0013	0F	16	RRCA	0074	56	87	LD D,(HL)
0014	102E	17	DJNZ DIS	0075	57	88	LD D,A
0016	118405	18	LD DE,NN	0076	58	89	LD E,B
0019	12	19	LD (DE),A	0077	59	90	LD E,C
001A	13	20	INC DE	0078	5A	91	LD E,D
001B	14	21	INC D	0079	5B	92	LD E,E
001C	15	22	DEC D	007A	5C	93	LD E,H
001D	1620	23	LD D,N	007B	5D	94	LD E,L
001F	17	24	RLA	007C	5E	95	LD E,(HL)
0020	182E	25	JR DIS	007D	5F	96	LD E,A
0022	19	26	ADD HL,DE	007E	60	97	LD H,B
0023	1A	27	LD A,(DE)	007F	61	98	LD H,C
0024	1B	28	DEC DE	0080	62	99	LD H,D
0025	1C	29	INC E	0081	63	100	LD H,E
0026	1D	30	DEC E	0082	64	101	LD H,H
0027	1E20	31	LD E,N	0083	65	102	LD H,L
0029	1F	32	RRA	0084	66	103	LD H,(HL)
002A	202E	33	JR NZ,DIS	0085	67	104	LD H,A
002C	218405	34	LD HL,NN	0086	68	105	LD L,B
002F	228405	35	LD (NN),HL	0087	69	106	LD L,C
0032	23	36	INC HL	0088	6A	107	LD L,D
0033	24	37	INC H	0089	6B	108	LD L,E
0034	25	38	DEC H	008A	6C	109	LD L,H
0035	2620	39	LD H,N	008B	6D	110	LD L,L
0037	27	40	DAA	008C	6E	111	LD L,(HL)
0038	282E	41	JR Z,DIS	008D	6F	112	LD L,A
003A	29	42	ADD HL,HL	008E	70	113	LD (HL),B
003B	2A8405	43	LD HL,(NN)	008F	71	114	LD (HL),C
003E	2B	44	DEC HL	0090	72	115	LD (HL),D
003F	2C	45	INC L	0091	73	116	LD (HL),E
0040	2D	46	DEC L	0092	74	117	LD (HL),H
0041	2E20	47	LD L,N	0093	75	118	LD (HL),L
0043	2F	48	CPL	0094	76	119	HALT
0044	302E	49	JR NC,DIS	0095	77	120	LD (HL),A
0046	318405	50	LD SP,NN	0096	78	121	LD A,B
0049	328405	51	LD (NN),A	0097	79	122	LD A,C
004C	33	52	INC SP	0098	7A	123	LD A,D
004D	34	53	INC (HL)	0099	7B	124	LD A,E
004E	35	54	DEC (HL)	009A	7C	125	LD A,H
004F	3620	55	LD (HL),N	009B	7D	126	LD A,L
0051	37	56	SCF	009C	7E	127	LD A,(HL)
0052	382E	57	JR C,DIS	009D	7F	128	LD A,A
0054	39	58	ADD HL,SP	009E	80	129	ADD A,B
0055	3A8405	59	LD A,(NN)	009F	81	130	ADD A,C
0058	3B	60	DEC SP	00A0	82	131	ADD A,D
0059	3C	61	INC A	00A1	83	132	ADD A,E
005A	3D	62	DEC A	00A2	84	133	ADD A,H
005B	3E20	63	LD A,N	00A3	85	134	ADD A,L
005D	3F	64	CCF	00A4	86	135	ADD A,(HL)
005E	40	65	LD B,B	00A5	87	136	ADD A,A
005F	41	66	LD B,C	00A6	88	137	ADC A,B
0060	42	67	LD B,D	00A7	89	138	ADC A,C
0061	43	68	LD B,E	00A8	8A	139	ADC A,D
0062	44	69	LD B,H,NN	00A9	8B	140	ADC A,E
0063	45	70	LD B,L	00AA	8C	141	ADC A,H
0064	46	71	LD B,(HL)	00AB	8D	142	ADC A,L

FIGURA 1 +

LOC	OBJ CODE	STMT	SOURCE STATEMENT	LOC	OBJ CODE	STMT	SOURCE STATEMENT
00AC	8E	143	ADC A,(HL)	010B	DA8405	218	JP C,NN
00AD	8F	144	ADC A,A	010E	DB20	219	IN A,N
00AE	90	145	SUB B	0110	DC8405	220	CALL C,NN
00AF	91	146	SUB C	0113	DE20	221	SBC A,N
00B0	92	147	SUB D	0115	DF	222	RST 18H
00B1	93	148	SUB E	0116	E0	223	RET PO
00B2	94	149	SUB H	0117	E1	224	POP HL
00B3	95	150	SUB L	0118	E28405	225	JP PO,NN
00B4	96	151	SUB (HL)	011B	E3	226	EX (SP),HL
00B5	97	152	SUB A	011C	E48405	227	CALL PO,NN
00B6	98	153	SBC A,B	011F	E5	228	PUSH HL
00B7	99	154	SBC A,C	0120	E620	229	AND N
00B8	9A	155	SBC A,D	0122	E7	230	RST 20H
00B9	9B	156	SBC A,E	0123	E8	231	RET PE
00BA	9C	157	SBC A,H	0124	E9	232	JP (HL)
00BB	9D	158	SBC A,L	0125	EA8405	233	JP PE,NN
00BC	9E	159	SBC A,(HL)	0128	EB	234	EX DE,HL
00BD	9F	160	SBC A,A	0129	EC8405	235	CALL PO,NN
00BE	A0	161	AND B	012C	EE20	236	XOR N
00BF	A1	162	AND C	012E	EF	237	RST 28H
00C0	A2	163	AND D	012F	F0	238	RET P
00C1	A3	164	AND E	0130	F1	239	POP AF
00C2	A4	165	AND H	0131	F28405	240	JP P,NN
00C3	A5	166	AND L	0134	F3	241	DI
00C4	A6	167	AND (HL)	0135	F48405	242	CALL P,NN
00C5	A7	168	AND A	0138	F5	243	PUSH AF
00C6	A8	169	XOR B	0139	F620	244	OR N
00C7	A9	170	XOR C	013B	F7	245	RST 30H
00C8	AA	171	XOR D	013C	F8	246	RET M
00C9	AB	172	XOR E	013D	F9	247	LD SP,HL
00CA	AC	173	XOR H	013E	FA8405	248	JP M,NN
00CB	AD	174	XOR L	0141	FB	249	EI
00CC	AE	175	XOR (HL)	0142	FC8405	250	CALL M,NN
00CD	AF	176	XOR A	0145	FE20	251	CP N
00CE	B0	177	OR B	0147	FF	252	RST 38H
00CF	B1	178	OR C	0148	CB00	253	RLC B
00D0	B2	179	OR D	014A	CB01	254	RLC C
00D1	B3	180	OR E	014C	CB02	255	RLC D
00D2	B4	181	OR H	014E	CB03	256	RLC E
00D3	B5	182	OR L	0150	CB04	257	RLC H
00D4	B6	183	OR (HL)	0152	CB05	258	RLC L
00D5	B7	184	OR A	0154	CB06	259	RLC (HL)
00D6	B8	185	CP B	0156	CB07	260	RLC A
00D7	B9	186	CP C	0158	CB08	261	RRC B
00D8	BA	187	CP D	015A	CB09	262	RRC C
00D9	BB	188	CP E	015C	CB0A	263	RRC D
00DA	BC	189	CP H	015E	CB0B	264	RRC E
00DB	BD	190	CP L	0160	CB0C	265	RRC H
00DC	BE	191	CP (HL)	0162	CB0D	266	RRC L
00DD	BF	192	CP A	0164	CB0E	267	RRC (HL)
00DE	C0	193	RET NZ	0166	CB0F	268	RRC A
00DF	C1	194	POP BC	0168	CB10	269	RL B
00E0	C28405	195	JP NZ, NN	016A	CB11	270	RL C
00E3	C38405	196	JP NN	016C	CB12	271	RL D
00E6	C48405	197	CALL NZ,NN	016E	CB13	272	RL E
00E9	C5	198	PUSH BC	0170	CB14	273	RL H
00EA	C620	199	ADD A,N	0172	CB15	274	RL L
00EC	C7	200	RST 0	0174	CB16	275	RL (HL)
00ED	C8	201	RET Z	0176	CB17	276	RL A
00EE	C9	202	RET	0178	CB18	277	RR B
00EF	CA8405	203	JP Z,NN	017A	CB19	278	RR C
00F2	CC8405	204	CALL Z,NN	017C	CB1A	279	RR D
00F5	CD8405	205	CALL NN	017E	CB1B	280	RR E
00F8	CE20	206	ADC A,N	0180	CB1C	281	RR H
00FA	CF	207	RST 8	0182	CB1D	282	RR L
00FB	D0	208	RET NC	0184	CB1E	283	RR (HL)
00FC	D1	209	POP DE	0186	CB1F	284	RR A
00FD	D28405	210	JP NC,NN	0188	CB20	285	SLA B
0100	D320	211	OUT N,A	018A	CB21	286	SLA C
0102	D48405	212	CALL NC,NN	018C	CB22	287	SLA D
0105	D5	213	PUSH DE	018E	CB23	288	SLA E
0106	D620	214	SUB N	0190	CB24	289	SLA H
0108	D7	215	RST 10H	0192	CB25	290	SLA L
0109	D8	216	RET C	0194	CB26	291	SLA (HL)
010A	D9	217	EXX	0196	CB27	292	SLA A

FIGURA 1 →

LOC	OBJ CODE	STMT	SOURCE STATEMENT	LOC	OBJ CODE	STMT	SOURCE STATEMENT
0198	CB28	293	SRA B	0230	CB7C	369	BIT 7,H
019A	CB29	294	SRA C	0232	CB7D	370	BIT 7,L
019C	CB2A	295	SRA D	0234	CB7E	371	BIT 7,(HL)
019E	CB2B	296	SRA E	0236	CB7F	372	BIT 7,A
01A0	CB2C	297	SRA H	0238	CB80	373	RES 0,B
01A2	CB2D	298	SRA L	023A	CB81	374	RES 0,C
01A4	CB2E	299	SRA (HL)	023C	CB82	375	RES 0,D
01A6	CB2F	300	SRA A	023E	CB83	376	RES 0,E
01A8	CB38	301	SRL B	0240	CB84	377	RES 0,H
01AA	CB39	302	SRL C	0242	CB85	378	RES 0,L
01AC	CB3A	303	SRL D	0244	CB86	379	RES 0,(HL)
01AE	CB3B	304	SRL E	0246	CB87	380	RES 0,A
01B0	CB3C	305	SRL H	0248	CB88	381	RES 1,B
01B2	CB3D	306	SRL L	024A	CB89	382	RES 1,C
01B4	CB3E	307	SRL (HL)	024C	CB8A	383	RES 1,D
01B6	CB3F	308	SRL A	024E	CB8B	384	RES 1,E
01B8	CB40	309	BIT 0,B	0250	CB8C	385	RES 1,H
01BA	CB41	310	BIT 0,C	0252	CB8D	386	RES 1,L
01BC	CB42	311	BIT 0,D	0254	CB8E	387	RES 1,(HL)
01BE	CB43	312	BIT 0,E	0256	CB8F	388	RES 1,A
01C0	CB44	313	BIT 0,H	0258	CB90	389	RES 2,B
01C2	CB45	314	BIT 0,L	025A	CB91	390	RES 2,C
01C4	CB46	315	BIT 0,(HL)	025C	CB92	391	RES 2,D
01C6	CB47	316	BIT 0,A	025E	CB93	392	RES 2,E
01C8	CB48	317	BIT 1,B	0260	CB94	393	RES 2,H
01CA	CB49	318	BIT 1,C	0262	CB95	394	RES 2,L
01CC	CB4A	319	BIT 1,D	0264	CB96	395	RES 2,(HL)
01CE	CB4B	320	BIT 1,E	0266	CB97	396	RES 2,A
01D0	CB4C	321	BIT 1,H	0268	CB98	397	RES 3,B
01D2	CB4D	322	BIT 1,L	026A	CB99	398	RES 3,C
01D4	CB4E	323	BIT 1,(HL)	026C	CB9A	399	RES 3,D
01D6	CB4F	324	BIT 1,A	026E	CB9B	400	RES 3,E
01D8	CB50	325	BIT 2,B	0270	CB9C	401	RES 3,H
01DA	CB51	326	BIT 2,C	0272	CB9D	402	RES 3,L
01DC	CB52	327	BIT 2,D	0274	CB9E	403	RES 3,(HL)
01DE	CB53	328	BIT 2,E	0276	CB9F	404	RES 3,A
01E0	CB54	329	BIT 2,H	0278	CBA0	405	RES 4,B
01E2	CB55	330	BIT 2,L	027A	CBA1	406	RES 4,C
01E4	CB56	331	BIT 2,(HL)	027C	CBA2	407	RES 4,D
01E6	CB57	332	BIT 2,A	027E	CBA3	408	RES 4,E
01E8	CB58	333	BIT 3,B	0280	CBA4	409	RES 4,H
01EA	CB59	334	BIT 3,C	0282	CBA5	410	RES 4,L
01EC	CB5A	335	BIT 3,D	0284	CBA6	411	RES 4,(HL)
01EE	CB5B	336	BIT 3,E	0286	CBA7	412	RES 4,A
01F0	CB5C	337	BIT 3,H	0288	CBA8	413	RES 5,B
01F2	CB5D	338	BIT 3,L	028A	CBA9	414	RES 5,C
01F4	CB5E	339	BIT 3,(HL)	028C	CBAA	415	RES 5,D
01F6	CB5F	340	BIT 3,A	028E	CBAB	416	RES 5,E
01F8	CB60	341	BIT 4,B	0290	CBAC	417	RES 5,H
01FA	CB61	342	BIT 4,C	0292	CBAD	418	RES 5,L
01FC	CB62	343	BIT 4,D	0294	CBAE	419	RES 5,(HL)
01FE	CB63	344	BIT 4,E	0296	CBAF	420	RES 5,A
0200	CB64	345	BIT 4,H	0298	CBBO	421	RES 6,B
0202	CB65	346	BIT 4,L	029A	CBB1	422	RES 6,C
0204	CB66	347	BIT 4,(HL)	029C	CBB2	423	RES 6,D
0206	CB67	348	BIT 4,A	029E	CBB3	424	RES 6,E
0208	CB68	349	BIT 5,B	02A0	CBB4	425	RES 6,H
020A	CB69	350	BIT 5,C	02A2	CBB5	426	RES 6,L
020C	CB6A	351	BIT 5,D	02A4	CBB6	427	RES 6,(HL)
020E	CB6B	352	BIT 5,E	02A6	CBB7	428	RES 6,A
0210	CB6C	353	BIT 5,H	02A8	CBB8	429	RES 7,B
0212	CB6D	354	BIT 5,L	02AA	CBB9	430	RES 7,C
0214	CB6E	355	BIT 5,(HL)	02AC	CBBA	431	RES 7,D
0216	CB6F	356	BIT 5,A	02AE	CBBB	432	RES 7,E
0218	CB70	357	BIT 6,B	0280	CBBC	433	RES 7,H
021A	CB71	358	BIT 6,C	0282	CBBD	434	RES 7,L
021C	CB72	359	BIT 6,D	0284	CBBE	435	RES 7,(HL)
021E	CB73	360	BIT 6,E	0286	CBBF	436	RES 7,A
0220	CB74	361	BIT 6,H	0288	CBC0	437	SET 0,B
0222	CB75	362	BIT 6,L	02BA	CBC1	438	SET 0,C
0224	CB76	363	BIT 6,(HL)	02BC	CBC2	439	SET 0,D
0226	CB77	364	BIT 6,A	02BE	CBC3	440	SET 0,E
0228	CB78	365	BIT 7,B	02C0	CBC4	441	SET 0,H
022A	CB79	366	BIT 7,C	02C2	CBC5	442	SET 0,L
022C	CB7A	367	BIT 7,D	02C4	CBC6	443	SET 0,(HL)
022E	CB7B	368	BIT 7,E	02C6	CBC7	444	SET 0,A

FIGURA 1 →

LOC	OBJ CODE	STMT	SOURCE	STATEMENT	LOC	OBJ CODE	STMT	SOURCE	STATEMENT
02C8	CBC8	445	SET 1,B		036F	DD7105	520	LD (IX+IND),C	
02CA	CBC9	446	SET 1,C		0372	DD7205	521	LD (IX+IND),D	
02CC	CBCA	447	SET 1,D		0375	DD7305	522	LD (IX+IND),E	
02CE	CBCB	448	SET 1,E		0378	DD7405	523	LD (IX+IND),H	
02D0	CBCC	449	SET 1,H		037B	DD7505	524	LD (IX+IND),L	
02D2	CBCE	450	SET 1,L		037E	DD7705	525	LD (IX+IND),A	
02D4	CBCE	451	SET 1,(HL)		0381	DD7E05	526	LD A,(IX+IND)	
02D6	CBCF	452	SET 1,A		0384	DD8605	527	ADD A,(IX+IND)	
02D8	CBD0	453	SET 2,B		0387	DD8E05	528	ADC A,(IX+IND)	
02DA	CBD1	454	SET 2,C		038A	DD9605	529	SUB (IX+IND)	
02DC	CBD2	455	SET 2,D		038D	DD9E05	530	SBC A,(IX+IND)	
02DE	CBD3	456	SET 2,E		0390	DDA605	531	AND (IX+IND)	
02E0	CBD4	457	SET 2,H		0393	DDAE05	532	XOR (IX+IND)	
02E2	CBD5	458	SET 2,L		0396	DD8605	533	OR (IX+IND)	
02E4	CBD6	459	SET 2,(HL)		0399	DD8E05	534	CP (IX+IND)	
02E6	CBD7	460	SET 2,A		039C	DDE1	535	POP IX	
02E8	CBD8	461	SET 3,B		039E	DDE3	536	EX (SP),IX	
02EA	CBD9	462	SET 3,C		03A0	DDE5	537	PUSH IX	
02EC	CBDA	463	SET 3,D		03A2	DDE9	538	JP (IX)	
02EE	CBDB	464	SET 3,E		03A4	DDF9	539	LD SP,IX	
02F0	CBDC	465	SET 3,H		03A6	DDCB0506	540	RLC (IX+IND)	
02F2	CBDD	466	SET 3,L		03AA	DDCB050E	541	RRC (IX+IND)	
02F4	CBDE	467	SET 3,(HL)		03AE	DDCB0516	542	RL (IX+IND)	
02F6	CBDF	468	SET 3,A		03B2	DDCB051E	543	RR (IX+IND)	
02F8	CBE0	469	SET 4,B		03B6	DDCB0526	544	SLA (IX+IND)	
02FA	CBE1	470	SET 4,C		03BA	DDCB052E	545	SRA (IX+IND)	
02FC	CBE2	471	SET 4,D		03BE	DDCB053E	546	SRL (IX+IND)	
02FE	CBE3	472	SET 4,E		03C2	DDCB0546	547	BIT 0,(IX+IND)	
0300	CBE4	473	SET 4,H		03C6	DDCB054E	548	BIT 1,(IX+IND)	
0302	CBE5	474	SET 4,L		03CA	DDCB0556	549	BIT 2,(IX+IND)	
0304	CBE6	475	SET 4,(HL)		03CE	DDCB055E	550	BIT 3,(IX+IND)	
0306	CBE7	476	SET 4,A		03D2	DDCB0566	551	BIT 4,(IX+IND)	
0308	CBE8	477	SET 5,B		03D6	DDCB056E	552	BIT 5,(IX+IND)	
030A	CBE9	478	SET 5,C		03DA	DDCB0576	553	BIT 6,(IX+IND)	
030C	CBEA	479	SET 5,D		03DE	DDCB057E	554	BIT 7,(IX+IND)	
030E	CBEB	480	SET 5,E		03E2	DDCB0586	555	RES 0,(IX+IND)	
0310	CBEC	481	SET 5,H		03E6	DDCB058E	556	RES 1,(IX+IND)	
0312	CBED	482	SET 5,L		03EA	DDCB0596	557	RES 2,(IX+IND)	
0314	CBEE	483	SET 5,(HL)		03EE	DDCB059E	558	RES 3,(IX+IND)	
0316	CBEF	484	SET 5,A		03F2	DDCB05A6	559	RES 4,(IX+IND)	
0318	CBF0	485	SET 6,B		03F6	DDCB05AE	560	RES 5,(IX+IND)	
031A	CBF1	486	SET 6,C		03FA	DDCB05B6	561	RES 6,(IX+IND)	
031C	CBF2	487	SET 6,D		03FE	DDCB05BE	562	RES 7,(IX+IND)	
031E	CBF3	488	SET 6,E		0402	DDCB05C6	563	SET 0,(IX+IND)	
0320	CBF4	489	SET 6,H		0406	DDCB05CE	564	SET 1,(IX+IND)	
0322	CBF5	490	SET 6,L		040A	DDCB05D6	565	SET 2,(IX+IND)	
0324	CBF6	491	SET 6,(HL)		040E	DDCB05DE	566	SET 3,(IX+IND)	
0326	CBF7	492	SET 6,A		0412	DDCB05E6	567	SET 4,(IX+IND)	
0328	CBF8	493	SET 7,B		0416	DDCB05EE	568	SET 5,(IX+IND)	
032A	CBF9	494	SET 7,C		041A	DDCB05F6	569	SET 6,(IX+IND)	
032C	CBFA	495	SET 7,D		041E	DDCB05FE	570	SET 7,(IX+IND)	
032E	CBFB	496	SET 7,E		0422	ED40	571	IN B,(C)	
0330	CBFC	497	SET 7,H		0424	ED41	572	OUT (C),B	
0332	CBFD	498	SET 7,L		0426	ED42	573	SBC HL,BC	
0334	CBFE	499	SET 7,(HL)		0428	ED438405	574	LD (NN),BC	
0336	CBFF	500	SET 7,A		042C	ED44	575	NEG	
0338	DD09	501	ADD IX,BC		042E	ED45	576	RETN	
033A	DD19	502	ADD IX,DE		0430	ED46	577	IM 0	
033C	DD218405	503	LD IX,NN		0432	ED47	578	LD I,A	
0340	DD228405	504	LD (NN),IX		0434	ED48	579	IN C,(C)	
0344	DD23	505	INC IX		0436	ED49	580	OUT (C),C	
0346	DD29	506	ADD IX,IX		0438	ED4A	581	ADC HL,BC	
0348	DD2A8405	507	LD IX,(NN)		043A	ED4B8405	582	LD BC,(NN)	
034C	DD2B	508	DEC IX		043E	ED4D	583	RETI	
034E	DD3405	509	INC (IX+IND)		0440	ED50	584	IN D,(C)	
0351	DD3505	510	DEC (IX+IND)		0442	ED51	585	OUT (C),D	
0354	DD360520	511	LD (IX+IND),N		0444	ED52	586	SBC HL,DE	
0358	DD39	512	ADD IX,SP		0446	ED538405	587	LD (NN),DE	
035A	DD4605	513	LD B,(IX+IND)		044A	ED56	588	IM 1	
035D	DD4E05	514	LD C,(IX+IND)		044C	ED57	589	LD A,I	
0360	DD5605	515	LD D,(IX+IND)		044E	ED58	590	IN E,(C)	
0363	DD5E05	516	LD E,(IX+IND)		0450	ED59	591	OUT (C),E	
0366	DD6605	517	LD H,(IX+IND)		0452	ED5A	592	ADC HL,DE	
0369	DD6E05	518	LD L,(IX+IND)		0454	ED5B8405	593	LD DE,(NN)	
036C	DD7005	519	LD (IX+IND),B		0458	ED5E	594	IM 2	

FIGURA 1 →

LOC	OBJ CODE	STMT	SOURCE STATEMENT
045A	ED60	595	IN H,(C)
045C	ED61	596	OUT (C),H
045E	ED62	597	SBC HL,HL
0460	ED67	598	RRD
0462	ED68	599	IN L,(C)
0464	ED69	600	OUT (C),L
0466	ED6A	601	ADC HL,HL
0468	ED6F	602	RLD
046A	ED72	603	SBC HL,SP
046C	ED738405	604	LD (NN),SP
0470	ED78	605	IN A,(C)
0472	ED79	606	OUT (C),A
0474	ED7A	607	ADC HL,SP
0476	ED7B8405	608	LD SP, (NN)
047A	EDA0	609	LDI
047C	EDA1	610	CPI
047E	EDA2	611	INI
0480	EDA3	612	OUTI
0482	EDA8	613	LDD
0484	EDA9	614	CPD
0486	EDAA	615	IND
0488	EDAB	616	OUTD
048A	EDB0	617	LDIR
048C	EDB1	618	CFIR
048E	EDB2	619	INIR
0490	EDB3	620	OTIR
0492	EDB8	621	LDDR
0494	EDB9	622	CPDR
0496	EDBA	623	INDR
0498	EDBB	624	OTDR
049A	FD09	625	ADD IY,BC
049C	FD19	626	ADD IY,DE
049E	FD218405	627	LD IY,NN
04A2	FD228405	628	LD (NN),IY
04A6	FD23	629	INC IY
04A8	FD29	630	ADD IY,IY
04AA	FD2A8405	631	LD IY,(NN)
04AE	FD2B	632	DEC IY
04B0	FD3405	633	INC (IY+IND)
04B3	FD3505	634	DEC (IY+IND)
04B6	FD360520	635	LD (IY+IND),N
04BA	FD39	636	ADD IY,SP
04BC	FD4605	637	LD B,(IY+IND)
04BF	FD4E05	638	LD C,(IY+IND)
04C2	FD5605	639	LD D,(IY+IND)
04C5	FD5E05	640	LD E,(IY+IND)
04C8	FD6605	641	LD H,(IY+IND)
04CB	FD6E05	642	LD L,(IY+IND)
04CE	FD7005	643	LD (IY+IND),B
04D1	FD7105	644	LD (IY+IND),C
04D4	FD7205	645	LD (IY+IND),D
04D7	FD7305	646	LD (IY+IND),E
04DA	FD7405	647	LD (IY+IND),H
04DD	FD7505	648	LD (IY+IND),L
04E0	FD7705	649	LD (IY+IND),A
04E3	FD7E05	650	LD A,(IY+IND)
04E6	FD8605	651	ADD A,(IY+IND)
04E9	FD8E05	652	ADC A,(IY+IND)
04EC	FD9605	653	SUB (IY+IND)
04EF	FD9E05	654	SBC A,(IY+IND)
04F2	FDA605	655	AND (IY+IND)
04F5	FDAE05	656	XOR (IY+IND)
04F8	FDB605	657	OR (IY+IND)
04FB	FDBE05	658	CP (IY+IND)
04FE	FDE1	659	POP IY
0500	FDE3	660	EX (SP),IY
0502	FDE5	661	PUSH IY
0504	FDE9	662	JP (IY)
0506	FDE9	663	LD SP,IY
0508	FDCB0506	664	RLC (IY+IND)
050C	FDCB050E	665	RRC (IY+IND)
0510	FDCB0516	666	RL (IY+IND)
0514	FDCB051E	667	RR (IY+IND)
0518	FDCB0526	668	SLA (IY+IND)
051C	FDCB052E	669	SRA (IY+IND)

LOC	OBJ CODE	STMT	SOURCE STATEMENT
0520	FDCB053E	670	SRL (IY+IND)
0524	FDCB0546	671	BIT 0,(IY+IND)
0528	FDCB054E	672	BIT 1,(IY+IND)
052C	FDCB0556	673	BIT 2,(IY+IND)
0530	FDCB055E	674	BIT 3,(IY+IND)
0534	FDCB0566	675	BIT 4,(IY+IND)
0538	FDCB056E	676	BIT 5,(IY+IND)
053C	FDCB0576	677	BIT 6,(IY+IND)
0540	FDCB057E	678	BIT 7,(IY+IND)
0544	FDCB0586	679	RES 0,(IY+IND)
0548	FDCB058E	680	RES 1,(IY+IND)
054C	FDCB0596	681	RES 2,(IY+IND)
0550	FDCB059E	682	RES 3,(IY+IND)
0554	FDCB05A6	683	RES 4,(IY+IND)
0558	FDCB05AE	684	RES 5,(IY+IND)
055C	FDCB05B6	685	RES 6,(IY+IND)
0560	FDCB05BE	686	RES 7,(IY+IND)
0564	FDCB05C6	687	SET 0,(IY+IND)
0568	FDCB05CE	688	SET 1,(IY+IND)
056C	FDCB05D6	689	SET 2,(IY+IND)
0570	FDCB05DE	690	SET 3,(IY+IND)
0574	FDCB05E6	691	SET 4,(IY+IND)
0578	FDCB05EE	692	SET 5,(IY+IND)
057C	FDCB05F6	693	SET 6,(IY+IND)
0580	FDCB05FE	694	SET 7,(IY+IND)

Nota: NN quedó representado con 0584 (8405 con los bytes en orden inverso)

N quedó representado con 20

M quedó representado con 10

IND quedó representado con 05

DIS quedó representado con 30

FIGURA 1 (concluye)



TABLA I

LISTA DE VARIABLES

OB\$	Vector con componentes alfanuméricas en el cual ciertos segmentos de la lista de códigos objeto son almacenados en orden lexicográfico para buscarlos por búsqueda binaria y con el índice se consulta el vector SS\$ para ayudarse en el desensamblado
SS\$	Vector con componentes alfanuméricas que contiene pedazos de instrucciones desensambladas. (En ocasiones está la instrucción completa)
ST\$	Vector alfanumérico en el que se guarda byte por byte el código objeto del programa que se va a desensamblar
TA	Vector entero que guarda para algunas instrucciones objeto su tipo en la clasificación dentro de su grupo
Y\$	Vector alfanumérico auxiliar útil en el desensamblado de ciertas instrucciones. Su contenido se da en el artículo
W\$	Vector alfanumérico auxiliar de naturaleza similar a Y\$
V\$	Vector alfanumérico auxiliar similar a W\$ y Y\$
X	Valor del índice en el que termina la búsqueda binaria en el vector OB\$
O	Bandera para determinar si una búsqueda binaria tuvo éxito. O=1 significa éxito. O=0 significa fracaso
Z\$	Variable alfanumérica útil para la decodificación de números hexadecimales
II,FF	Indices inicial y final en el segmento del vector OB\$ en el que se lleva a cabo una búsqueda binaria
L	Longitud en bytes del programa a desensamblar
I	Índice del byte que se está desensamblando en el vector ST\$
H,B,K, C,P\$, Q\$,X\$	Variables auxiliares con propósitos varios

```

10 CLEAR 2000:CLS
15 DEFINT C-R,T-Z
20 DIM OB$(256),SS$(100),TA(256),Y$(7),W$(18),V$(8)
21 FOR I=125 TO 163:READ OB$(I),TA(I),SS$(I):NEXT I
22 FOR I=0 TO 7:READ V$(I):NEXT I
24 FOR I=1 TO 10:READ W$(I):NEXT I
26 FOR I=0 TO 8:READ Y$(I):NEXT I
28 FOR I=1 TO 64
29 READ OB$(I),TA(I),SS$(I)
40 NEXT I
42 FOR I=65 TO 124
44 READ OB$(I),TA(I),SS$(I)
46 NEXT I
47 Z$="0123456789ABCDEF"
48 FOR I=164 TO 217:READ OB$(I),SS$(I):NEXT I
50 FOR I=218 TO 256:READ OB$(I),TA(I),SS$(I):NEXT I
59 READ L
60 FOR I=1 TO L
61 READ ST$(I)
62 NEXT I
200 I=1
110 IF ST$(I)="CB" THEN 120
115 GOTO 160
120 IF ST$(I+1)="D8"AND ST$(I+2)="FF" THEN 122
122 IF ST$(I+1)<="2F"AND ST$(I+2)="00" THEN 126
124 PRINT "ERROR":I:STOP
126 H=0:FOR K=1 TO 16:IF MID$(Z$,K,1)=MID$(ST$(I+1),1,1) THEN H=(K-1)*16:GOTO 128
127 NEXT K
128 FOR K=1 TO 16:IF MID$(Z$,K,1)=MID$(ST$(I+1),2,1) THEN H=H+K-1:GOTO 130
129 NEXT K
130 Y=INT(H/8)+1:B=H-INT(H/8)*8:PRINT W$(Y); " ";Y$(B):I=I+2:GOTO 110
131 PRINT "ERROR":I:STOP
132 H=0:FOR K=1 TO 16:IF MID$(Z$,K,1)=MID$(ST$(I+1),1,1) THEN H=(K-1)*16:GOTO 134
133 NEXT K
134 FOR K=1 TO 16:IF MID$(Z$,K,1)=MID$(ST$(I+1),2,1) THEN H=H+K-1:GOTO 136
135 NEXT K
136 Y=INT(H/8)+1:B=H-INT(H/8)*8:IF H=0 AND H=7 THEN PRINT W$(7); " ";Y$(B):I=I+2:GOTO 110
137 IF H=8 AND H=71 THEN PRINT W$(8); " ";Y-2; " ";Y$(B):I=I+2:GOTO 110
138 IF H=72 AND H=135 THEN PRINT W$(9); " ";Y-10; " ";Y$(B):I=I+2:GOTO 110
139 IF H=136 AND H=200 THEN PRINT W$(10); " ";Y-18; " ";Y$(B):I=I+2:GOTO 110
140 PRINT "ERROR":I:STOP
160 IF ST$(I)="40"AND ST$(I) <="BF" THEN 168
165 GOTO 210
168 IF ST$(I)="76" THEN PRINT "HALT":I=I+1:GOTO 110
170 FOR K=1 TO 16:IF MID$(Z$,K,1)=MID$(ST$(I),1,1) THEN H=(K-5)*16:GOTO 172
171 NEXT K
172 FOR K=1 TO 16:IF MID$(Z$,K,1)=MID$(ST$(I),2,1) THEN H=H+K-1:GOTO 174
173 NEXT K
174 B=INT(H/8):C=H-B*8
175 IF ST$(I) <="7F" THEN PRINT V$(0); " ";Y$(B); " ";Y$(C):I=I+1:GOTO 110
180 PRINT V$(B-7); " ";:IF B=8 OR B=9 OR B=11 THEN PRINT "A ";
181 PRINT V$(C):I=I+1:GOTO 110
200 PRINT "ERROR":I:STOP

```

FIGURA 2 +

```

210 IF ST$(I)+ST$(I+1)="FDCB" THEN 250
240 IF ST$(I)+ST$(I+1)="DCCB" THEN 250
245 GOTO 270
250 P$="(IX+"; Q$="(IY+"; IF ST$(I+3)="3E" THEN 268
251 IF ST$(I+3))="06" AND ST$(I+3)<="2E" THEN 254
252 IF ST$(I+3))="3E" AND ST$(I+3)<="FE" THEN 259
253 PRINT "ERROR"; I: STOP
254 H=VAL(MID$(ST$(I+3), 1, 1))*2: IF MID$(ST$(I+3), 2, 1)="6" THEN 257
255 IF MID$(ST$(I+3), 2, 1)="E" THEN H=H+1: GOTO 257
256 PRINT "ERROR"; I: STOP
257 PRINT W$(H+1); " "; : IF ST$(I)="D0" THEN PRINT P$; ST$(I+2); " " ELSE PRINT Q$; ST$(I+2); " "
258 GOTO 269
259 FOR K=1 TO 16: IF MID$(Z$, K, 1)=MID$(ST$(I+3), 1, 1) THEN H=(K-5)*2: GOTO 262
260 NEXT K
261 PRINT "ERROR"; I: STOP
262 IF MID$(ST$(I+3), 2, 1)="6" THEN 265
263 IF MID$(ST$(I+3), 2, 1)="E" THEN H=H+1: GOTO 265
264 PRINT "ERROR"; I: STOP
265 B=INT(H/8): C=H-B*8: PRINT W$(B+8); " "; C; " "; : IF ST$(I)="D0" THEN PRINT P$; ST$(I+2); " " ELSE PRINT Q$; ST$(I+2); " "
266 GOTO 269
268 PRINT W$(7); " "; : IF ST$(I)="D0" THEN PRINT P$; ST$(I+2); " " ELSE PRINT Q$; ST$(I+2); " "
269 I=I+4: GOTO 110
270 IF ST$(I)>="00" AND ST$(I)<="3F" THEN 280
275 GOTO 300
280 X$=ST$(I)
282 II=1: FF=64
290 GOSUB 1040
295 IF 0=1 AND TA(X)=1 THEN PRINT SS$(X): I=I+1: GOTO 110
300 IF 0=1 AND TA(X)=2 THEN PRINT SS$(X); ST$(I+1): I=I+2: GOTO 110
310 IF 0=1 AND TA(X)=3 THEN PRINT SS$(X); ST$(I+2); ST$(I+1): I=I+3: GOTO 110
320 IF 0=1 AND TA(X)=8 THEN 330
325 PRINT "ERROR"; I: STOP
330 IF X=35 THEN PRINT "LD ("; ST$(I+2); ST$(I+1); ")", HL: I=I+3: GOTO 110
340 IF X=43 THEN PRINT "LD HL, ("; ST$(I+2); ST$(I+1); ")", I=I+3: GOTO 110
350 IF X=51 THEN PRINT "LD ("; ST$(I+2); ST$(I+1); ")", A: I=I+3: GOTO 110
360 IF X=59 THEN PRINT "LD R, ("; ST$(I+2); ST$(I+1); ")", I=I+3: GOTO 110
370 PRINT "ERROR"; I: STOP
380 IF ST$(I)="D0" THEN 390
385 GOTO 540
390 X$=ST$(I)+ST$(I+1)
400 II=125: FF=163: GOSUB 1050
410 IF 0=1 AND TA(X)=4 THEN PRINT SS$(X): I=I+2: GOTO 110
420 IF 0=1 AND TA(X)=5 THEN PRINT SS$(X); ST$(I+2); " ": I=I+3: GOTO 110
430 IF 0=1 AND X=127 THEN PRINT "LD IX, "; ST$(I+3); ST$(I+2): I=I+4: GOTO 110
440 IF 0=1 AND X=128 THEN PRINT "LD ("; ST$(I+3); ST$(I+2); ")", IX: I=I+4: GOTO 110
450 IF 0=1 AND X=131 THEN PRINT "LD IX, ("; ST$(I+3); ST$(I+2); ")", I=I+4: GOTO 110
460 IF 0=1 AND X=135 THEN PRINT "LD (IX+"; ST$(I+2); ")", ST$(I+3): I=I+4: GOTO 110
470 IF 0=1 AND TA(X)=8 THEN PRINT "LD (IX+"; ST$(I+2); ")", SS$(X): I=I+3: GOTO 110

```

FIGURA 2 →

```

488 PRINT "ERROR"; I:STOP
501 DATA D009, 4, "ADD IX, BC"
502 DATA D019, 4, "ADD IX, DE"
503 DATA D0218405, 6, "LD IX, "
504 DATA D0228405, 6, X
505 DATA D023, 4, INC IX
506 DATA D029, 4, "ADD IX, IX"
507 DATA D02A8405, 6, X
508 DATA D02B, 4, DEC IX
509 DATA D003405, 5, INC (IX+
510 DATA D03505, 5, DEC (IX+
511 DATA D0360520, 6, X
512 DATA D039, 4, "ADD IX, SP"
513 DATA D04605, 5, "LD B, (IX+"
514 DATA D04E05, 5, "LD C, (IX+"
515 DATA D05605, 5, "LD D, (IX+"
516 DATA D05E05, 5, "LD E, (IX+"
517 DATA D06605, 5, "LD H, (IX+"
518 DATA D06E05, 5, "LD L, (IX+"
519 DATA D07005, 8, B
520 DATA D07105, 8, C
521 DATA D07205, 8, D
522 DATA D07305, 8, E
523 DATA D07405, 8, H
524 DATA D07505, 8, L
525 DATA D07705, 8, A
526 DATA D07E05, 5, "LD A, (IX+"
527 DATA D08605, 5, "ADD A, (IX+"
528 DATA D08E05, 5, "ADC A, (IX+"
529 DATA D09605, 5, "SUB (IX+"
530 DATA D09E05, 5, "SBC A, (IX+"
531 DATA D0A605, 5, "AND (IX+"
532 DATA D0AE05, 5, XOR (IX+
533 DATA D0B605, 5, OR (IX+
534 DATA D0BE05, 5, CP (IX+
535 DATA D0E1, 4, POP IX
536 DATA D0E3, 4, "EX (SP), IX"
537 DATA D0E5, 4, PUSH IX
538 DATA D0E9, 4, JP (IX)
539 DATA D0F9, 4, "LD SP, IX"

```

FIGURA 2 +


```

540 IF ST$(I)="ED" THEN 550
545 GOTO 650
550 X$=ST$(I)+ST$(I+1)
560 II=164:FF=217:GOSUB 1050
580 IF 0=1ANDX=167 THEN PRINT "LD (", ST$(I+3); ST$(I+2); ")", BC": I=I+4: GOTO 110
590 IF 0=1ANDX=175 THEN PRINT "LD BC, (", ST$(I+3); ST$(I+2); ")", I=I+4: GOTO 110
600 IF 0=1 ANDX=180 THEN PRINT "LD (", ST$(I+3); ST$(I+2); ")", DE": I=I+4: GOTO 110
610 IF 0=1ANDX=186 THEN PRINT "LD DE, (", ST$(I+3); ST$(I+2); ")", I=I+4: GOTO 110
620 IF 0=1ANDX=197 THEN PRINT "LD (", ST$(I+3); ST$(I+2); ")", SP": I=I+4: GOTO 110
630 IF 0=1ANDX=201 THEN PRINT "LD SP, (", ST$(I+3); ST$(I+2); ")", I=I+4: GOTO 110
635 IF 0=1 THEN PRINT SS$(X): I=I+2: GOTO 110
640 PRINT "ERROR"; I: STOP
650 IF ST$(I)="FD" THEN 660
655 GOTO 760
660 X$=ST$(I)+ST$(I+1)
670 II=218:FF=256:GOSUB 1050
680 IF 0=1ANDTR(X)=4 THEN PRINT SS$(X): I=I+2: GOTO 110
690 IF 0=1ANDTR(X)=5 THEN PRINT SS$(X); ST$(I+2); "": I=I+3: GOTO 110
700 IF 0=1ANDX=220 THEN PRINT "LD IV, (", ST$(I+3); ST$(I+2); ")", I=I+4: GOTO 110
710 IF 0=1ANDX=221 THEN PRINT "LD (", ST$(I+3); ST$(I+2); ")", IV": I=I+4: GOTO 110
720 IF 0=1ANDX=224 THEN PRINT "LD IV, (", ST$(I+3); ST$(I+2); ")", I=I+4: GOTO 110
730 IF 0=1ANDX=228 THEN PRINT "LD (IV+", ST$(I+2); ")", ST$(I+3): I=I+4: GOTO 110
740 IF 0=1ANDTR(X)=8 THEN PRINT "LD (IV+", ST$(I+2); ")", SS$(X): I=I+3: GOTO 110
750 PRINT "ERROR"; I: STOP
760 IF ST$(I)>="C0" AND ST$(I)<="FF" THEN 770
765 GOTO 820
770 X$=ST$(I): II=65: FF=124: GOSUB 1040
780 IF 0=1ANDTR(X)=1 THEN PRINT SS$(X): I=I+1: GOTO 110
790 IF 0=1AND TR(X)=2 THEN PRINT SS$(X); ST$(I+1): I=I+2: GOTO 110
800 IF 0=1ANDTR(X)=3 THEN PRINT SS$(X); ST$(I+2); ST$(I+1): I=I+3: GOTO 110
810 IF 0=1ANDX=83 THEN PRINT "OUT ", ST$(I+1); ")", R": I=I+2: GOTO 110
820 IF ST$(I)="" THEN STOP
830 PRINT "ERROR"; I: STOP
1030 O=0
1032 IF II>FF THEN RETURN
1033 X=INT((II+FF)/2)
1034 IF OB$(X)=X$ THEN O=1: RETURN
1036 IF OB$(X)<X$ THEN II=X+1: GOTO 1032
1038 IF OB$(X)>X$ THEN FF=X-1: GOTO 1032
1040 O=0
1042 IF II>FF THEN RETURN
1043 X=INT((II+FF)/2)
1044 IF MID$(OB$(X), 1, 2)=X$ THEN O=1: RETURN
1046 IF MID$(OB$(X), 1, 2)<X$ THEN II=X+1: GOTO 1042
1048 IF MID$(OB$(X), 1, 2)>X$ THEN FF=X-1: GOTO 1042
1050 O=0
1052 IF II>FF THEN RETURN
1053 X=INT((II+FF)/2)
1054 IF MID$(OB$(X), 1, 4)=X$ THEN O=1: RETURN
1056 IF MID$(OB$(X), 1, 4)<X$ THEN II=X+1: GOTO 1052
1058 IF MID$(OB$(X), 1, 4)>X$ THEN FF=X-1: GOTO 1052
2000 DATAB, C, D, E, H, L, (HL), A, RLC, RRC, RL, RR, SLA, SRA, SRL, BIT, RES, SET, LD, ADD, ADC, SUB, SBC, AND, XOR, OR, CP

```

2001 DATA 00,1,NOP
 2002 DATA 010405,3,"LD BC,"
 2003 DATA 02,1,"LD (BC),A"
 2004 DATA 03,1,INC BC
 2005 DATA 04,1,INC B
 2006 DATA 05,1,DEC B
 2007 DATA 0620,2,"LD B,"
 2008 DATA 07,1,RLOA
 2009 DATA 08,1,"EX AF,AF"
 2010 DATA 09,1,"ADD HL,BC"
 2011 DATA 0A,1,"LD A,(BC)"
 2012 DATA 0B,1,DEC BC
 2013 DATA 0C,1,INC C
 2014 DATA 0D,1,DEC C
 2015 DATA 0E20,2,"LD C,"
 2016 DATA 0F,1,RROA
 2017 DATA 102E,2,"DJNZ "
 2018 DATA 110405,3,"LD DE,"
 2019 DATA 12,1,"LD (DE),A"
 2020 DATA 13,1,INC DE
 2021 DATA 14,1,INC D
 2022 DATA 15,1,DEC D
 2023 DATA 1620,2,"LD D,"
 2024 DATA 17,1,RLOA
 2025 DATA 182E,2,"JR "
 2026 DATA 19,1,"ADD HL,DE"
 2027 DATA 1A,1,"LD A,(DE)"
 2028 DATA 1B,1,DEC DE
 2029 DATA 1C,1,INC E
 2030 DATA 1D,1,DEC E
 2031 DATA 1E20,2,"LD E,"
 2032 DATA 1F,1,RROA
 2033 DATA 202E,2,"JR NZ,"
 2034 DATA 210405,3,"LD HL,"
 2035 DATA 220405,8,X
 2036 DATA 23,1,INC HL
 2037 DATA 24,1,INC H
 2038 DATA 25,1,DEC H
 2039 DATA 2620,2,"LD H,"
 2040 DATA 27,1,RROA
 2041 DATA 282E,2,"JR Z,"
 2042 DATA 29,1,"ADD HL,HL"
 2043 DATA 2A0405,8,X
 2044 DATA 2B,1,DEC HL
 2045 DATA 2C,1,INC L
 2046 DATA 2D,1,DEC L
 2047 DATA 2E20,2,"LD L,"
 2048 DATA 2F,1,CPL
 2049 DATA 302E,2,"JR NC,"
 2050 DATA 310405,3,"LD SP,"
 2051 DATA 320405,8,X
 2052 DATA 33,1,INC SP
 2053 DATA 34,1,INC (HL)
 2054 DATA 35,1,DEC (HL)
 2055 DATA 3620,2,"LD (HL),"
 2056 DATA 37,1,SCF
 2057 DATA 382E,2,"JR C,"
 2058 DATA 39,1,"ADD HL,SP"
 2059 DATA 3A0405,8,X
 2060 DATA 3B,1,DEC SP
 2061 DATA 3C,1,INC A
 2062 DATA 3D,1,DEC A
 2063 DATA 3E20,2,"LD A,"
 2064 DATA 3F,1,CCF
 2193 DATA C0,1,RET NZ
 2194 DATA C1,1,POP BC
 2195 DATA C20405,3,"JP NZ,"
 2196 DATA C30405,3,"JP "
 2197 DATA C40405,3,"CALL NZ,"
 2198 DATA C5,1,PUSH BC
 2199 DATA C620,2,"ADD A,"
 2200 DATA C7,1,RST 0
 2201 DATA C8,1,RET Z
 2202 DATA C9,1,RET
 2203 DATA CA0405,3,"JP Z,"
 2204 DATA CB0405,3,"CALL Z,"
 2205 DATA CC0405,3,"CALL "
 2206 DATA CE20,2,"ADC A,"
 2207 DATA CF,1,RST 8
 2208 DATA D0,1,RET NC
 2209 DATA D1,1,POP DE
 2210 DATA D20405,3,"JP NC,"
 2211 DATA D320,8,X
 2212 DATA D40405,3,"CALL NC,"
 2213 DATA D5,1,PUSH DE
 2214 DATA D620,2,"SUB "
 2215 DATA D7,1,RST 10H
 2216 DATA D8,1,RET C
 2217 DATA D9,1,EXX
 2218 DATA DA0405,3,"JP C,"
 2219 DATA DB20,2,"IN A,"
 2220 DATA DC0405,3,"CALL C,"
 2221 DATA DE20,2,"SBC A,"
 2222 DATA DF,1,RST 18H
 2223 DATA E0,1,RET PO
 2224 DATA E1,1,POP HL
 2225 DATA E20405,3,"JP PO,"
 2226 DATA E3,1,"EX (SP),HL"
 2227 DATA E40405,3,"CALL PO,"
 2228 DATA E5,1,PUSH HL
 2229 DATA E620,2,"AND "
 2230 DATA E7,1,RST 20H
 2231 DATA E8,1,RET PE
 2232 DATA E9,1,JP (HL)
 2233 DATA EA0405,3,"JP PE,"
 2234 DATA EB,1,"EX DE,HL"
 2235 DATA EC0405,3,"CALL PE,"
 2236 DATA EE20,2,"XOR "

FIGURA 2 →

2237 DATA EF, 1, RST 38H	2609 DATA ED98, LD1
2238 DATA F0, 1, RET P	2610 DATA ED9L, CPI
2239 DATA F1, 1, POP AF	2611 DATA ED92, INI
2240 DATA F28485, 3, "JP P, "	2612 DATA ED93, OUTI
2241 DATA F3, 1, DI	2613 DATA ED98, LDO
2242 DATA F48485, 3, "CALL P, "	2614 DATA ED99, CPD
2243 DATA F5, 1, PUSH AF	2615 DATA ED9A, IND
2244 DATA F520, 2, "OR "	2616 DATA ED9B, OUTD
2245 DATA F7, 1, RST 38H	2617 DATA ED90, LDIR
2246 DATA F8, 1, RET M	2618 DATA ED8L, CPIR
2247 DATA F9, 1, "LD SP, HL "	2619 DATA ED82, INIR
2248 DATA FA8485, 3, "JP M, "	2620 DATA ED83, OTIR
2249 DATA FB, 1, EI	2621 DATA ED88, LDOR
2250 DATA FC8485, 3, "CALL M, "	2622 DATA ED89, CPDR
2251 DATA FE20, 2, "CP "	2623 DATA ED8A, INDR
2252 DATA FF, 1, RST 38H	2624 DATA ED8B, OTDR
2571 DATA ED40, "IN B, (C)"	2625 DATA FD09, 4, "ADD IV, BC"
2572 DATA ED4L, "OUT (C), B"	2626 DATA FD19, 4, "ADD IV, DE"
2573 DATA ED42, "SBC HL, BC"	2627 DATA FD218485, 6, X
2574 DATA ED438485, X	2628 DATA FD228485, 6, X
2575 DATA ED44, NEG	2629 DATA FD23, 4, INC IV
2576 DATA ED45, RETH	2630 DATA FD29, 4, "ADD IV, IV"
2577 DATA ED46, IN 0	2631 DATA FD2A8485, 6, X
2578 DATA ED47, "LD I, A"	2632 DATA FD2B, 4, DEC IV
2579 DATA ED48, "IN C, (C)"	2633 DATA FD3485, 5, INC (IV+
2580 DATA ED49, "OUT (C), C"	2634 DATA FD3585, 5, DEC (IV+
2581 DATA ED4A, "ADC HL, BC"	2635 DATA FD368520, 6, X
2582 DATA ED488485, X	2636 DATA FD39, 4, "ADD IV, SP"
2583 DATA ED4D, RETI	2637 DATA FD4685, 5, "LD B, (IV+"
2584 DATA ED50, "IN D, (C)"	2638 DATA FD4E85, 5, "LD C, (IV+"
2585 DATA ED51, "OUT (C), D"	2639 DATA FD5685, 5, "LD D, (IV+"
2586 DATA ED52, "SBC HL, DE"	2640 DATA FD5E85, 5, "LD E, (IV+"
2587 DATA ED538485, X	2641 DATA FD6685, 5, "LD H, (IV+"
2588 DATA ED56, IN 1	2642 DATA FD6E85, 5, "LD L, (IV+"
2589 DATA ED57, "LD A, I"	2643 DATA FD7085, 8, B
2590 DATA ED58, "IN E, (C)"	2644 DATA FD7185, 8, C
2591 DATA ED59, "OUT (C), E"	2645 DATA FD7285, 8, D
2592 DATA ED5A, "ADC HL, DE"	2646 DATA FD7385, 8, E
2593 DATA ED5B8485, "LD DE, (C"	2647 DATA FD7485, 8, H
2594 DATA ED5E, IN 2	2648 DATA FD7585, 8, L
2595 DATA ED60, "IN H, (C)"	2649 DATA FD7785, 8, A
2596 DATA ED61, "OUT (C), H"	2650 DATA FD7E85, 5, "LD A, (IV+"
2597 DATA ED62, "SBC HL, HI"	2651 DATA FD8685, 5, "ADD A, (IV+"
2598 DATA ED67, RRD	2652 DATA FD8E85, 5, "ADC A, (IV+"
2599 DATA ED68, "IN L, (C)"	2653 DATA FD9685, 5, "SUB (IV+"
2600 DATA ED69, "OUT (C), L"	2654 DATA FD9E85, 5, "SBC A, (IV+"
2601 DATA ED6A, "ADC HL, HL"	2655 DATA FDA685, 5, "AND (IV+"
2602 DATA ED6F, RLD	2656 DATA FDAE85, 5, XOR (IV+
2603 DATA ED72, "SBC HL, SP"	2657 DATA FDB685, 5, OR (IV+
2604 DATA ED738485, X	2658 DATA FDBE85, 5, CP (IV+
2605 DATA ED78, "IN A, (C)"	2659 DATA FDE1, 4, POP IV
2606 DATA ED79, "OUT (C), A"	2660 DATA FDE3, 4, "EX (SP), IV"
2607 DATA ED7A, "ADC HL, SP"	2661 DATA FDE5, 4, PUSH IV
2608 DATA ED7B8485, X	2662 DATA FDE9, 4, JP (IV)
	2663 DATA FDF9, 4, "LD SP, IV"

FIGURA 2 (concluye)

```

59 END
60 INPUT "LOCALIDAD EN HEXADECIMAL INICIAL Y FINAL";BI$,BF$
62 X$=BI$:GOSUB850:BI=5
64 X$=BF$:GOSUB850:BF=5
66 L=BF-BI+1
68 PRINT "PROGRAMA OBJETO"
70 FOR I=0 TO L-1
72 X=PEEK(BI+I)
74 GOSUB 900:ST$(I+1)=X$
76 PRINT W$; " ";
78 NEXT I
80 PRINT:PRINT:PRINT "PROGRAMA FUENTE"
850 N1=LEN(X$)
852 FOR II=1 TO 16
854 FOR JJ=1 TO N1
856 IF MID$(X$,JJ,1)=MID$(Z$,II,1) THEN XX(JJ)=II-1
858 NEXT JJ
860 NEXT II
862 S=0
864 FOR JJ=1 TO N1
866 S=S+XX(JJ)*16I (N1-JJ)
868 NEXT JJ
870 RETURN
900 XX=X:W$="":X$=""
902 R1=INT(XX/16)
904 R1=XX-R1*16
906 X$=X$+MID$(Z$,R1+1,1)
908 IF R1=0 THEN 912
910 XX=R1:GOTO902
912 FOR II=LEN(X$) TO 1 STEP -1
914 W$=W$+MID$(X$,II,1)
916 NEXT II
918 RETURN

```

FIGURA 3

1	084(1)	SS4(1)	TA(1)	51	328485	X	8	181	E620	AND	2
1	08	NOP	1	52	33	INC SP	1	182	E7	RST 28H	1
2	018485	LD BC,	3	53	34	INC (HL)	1	183	E8	RET PE	1
3	02	LD (BC),A	1	54	35	DEC (HL)	1	184	E9	JP (HL)	1
4	03	INC BC	1	55	3620	LD (HL),	2	185	EA8485	JP PE,	3
5	04	INC B	1	56	37	SCF	1	186	EB	EX DE,HL	1
6	05	DEC B	1	57	382E	JR C,	2	187	EC8485	CALL PE,	3
7	0620	LD B,	2	58	39	ADD HL,SP	1	188	EE20	XOR	2
8	07	RLCA	1	59	3A8485	X	8	189	EF	RST 28H	1
9	08	EX AF,AF'	1	60	3B	DEC SP	1	110	F0	RET P	1
10	09	ADD HL,BC	1	61	3C	INC A	1	111	F1	POP AF	1
11	0A	LD A,(BC)	1	62	3D	DEC A	1	112	F28485	JP P,	3
12	0B	DEC BC	1	63	3E20	LD A,	2	113	F3	DI	1
13	0C	INC C	1	64	3F	CCF	1	114	F48485	CALL P,	3
14	0D	DEC C	1	65	00	RET NZ	1	115	F5	PUSH AF	1
15	0E20	LD C,	2	66	C1	POP BC	1	116	F620	OR	2
16	0F	RRCR	1	67	C28485	JP NZ,	3	117	F7	RST 30H	1
17	102E	DJNZ	2	68	C38485	JP	3	118	F8	RET M	1
18	118485	LD DE,	3	69	C48485	CALL NZ,	3	119	F9	LD SP,HL	1
19	12	LD (DE),A	1	70	C5	PUSH BC	1	120	FA8485	JP M,	3
20	13	INC DE	1	71	C620	ADD A,	2	121	FB	EI	1
21	14	INC D	1	72	C7	RST 0	1	122	FC8485	CALL M,	3
22	15	DEC D	1	73	C8	RET Z	1	123	FE20	CP	2
23	1620	LD D,	2	74	C9	RET	1	124	FF	RST 30H	1
24	17	RLA	1	75	CA8485	JP Z,	3	125	D009	ADD IX,BC	4
25	182E	JR	2	76	CC8485	CALL Z,	3	126	D019	ADD IX,DE	4
26	19	ADD HL,DE	1	77	CD8485	CALL	3	127	D0218485	LD IX,	6
27	1A	LD A,(DE)	1	78	CE20	ADC A,	2	128	D0228485	X	6
28	1B	DEC DE	1	79	CF	RST 8	1	129	D023	INC IX	4
29	1C	INC E	1	80	D0	RET NC	1	130	D029	ADD IX,IX	4
30	1D	DEC E	1	81	D1	POP DE	1	131	D02A8485	X	6
31	1E20	LD E,	2	82	D28485	JP NC,	3	132	D02B	DEC IX	4
32	1F	RRA	1	83	D320	X	8	133	D003485	INC (IX+	5
33	202E	JR NZ,	2	84	D48485	CALL NC,	3	134	D03585	DEC (IX+	5
34	218485	LD HL,	3	85	D5	PUSH DE	1	135	D0368520	X	6
35	228485	X	8	86	D620	SUB	2	136	D039	ADD IX,SP	4
36	23	INC HL	1	87	D7	RST 10H	1	137	D04685	LD B,(IX+	5
37	24	INC H	1	88	D8	RET C	1	138	D04E85	LD C,(IX+	5
38	25	DEC H	1	89	D9	EXX	1	139	D05685	LD D,(IX+	5
39	2620	LD H,	2	90	DA8485	JP C,	3	140	D05E85	LD E,(IX+	5
40	27	DAA	1	91	DB20	IN A,	2	141	D06685	LD H,(IX+	5
41	282E	JR Z,	2	92	DC8485	CALL C,	3	142	D06E85	LD L,(IX+	5
42	29	ADD HL,HL	1	93	DE20	SBC A,	2	143	D07085	B	8
43	2A8485	X	8	94	DF	RST 10H	1	144	D07185	C	8
44	2B	DEC HL	1	95	E0	RET PO	1	145	D07285	D	8
45	2C	INC L	1	96	E1	POP HL	1	146	D07385	E	8
46	2D	DEC L	1	97	E28485	JP PO,	3	147	D07485	H	8
47	2E20	LD L,	2	98	E3	EX (SP),HL	1	148	D07585	L	8
48	2F	CPL	1	99	E48485	CALL PO,	3	149	D07785	A	8
49	302E	JR NC,	2	100	E5	PUSH HL	1	150	D07E85	LD A,(IX+	5
50	318485	LD SP,	3								

FIGURA 4 →

151	D08605	ADD A, (IX+	5	201	ED7B8405	X	0
152	D08E05	ADC A, (IX+	5	202	ED90	LDI	0
153	D09605	SUB (IX+	5	203	ED91	CPI	0
154	D09E05	SBC A, (IX+	5	204	ED92	INI	0
155	D0A605	AND (IX+	5	205	ED93	OUTI	0
156	D0AE05	XOR (IX+	5	206	ED98	LDD	0
157	D0B605	OR (IX+	5	207	ED99	CPD	0
158	D0BE05	CP (IX+	5	208	ED9A	IND	0
159	D0E1	POP IX	4	209	ED9B	OUTD	0
160	D0E3	EX (SP), IX	4	210	ED9B	LDIR	0
161	D0E5	PUSH IX	4	211	EDB1	CFIR	0
162	D0E9	JP (IX)	4	212	EDB2	INIR	0
163	D0F9	LD SP, IX	4	213	EDB3	OTIR	0
164	ED40	IN B, (C)	0	214	EDB8	LDOR	0
165	ED41	OUT (C), B	0	215	EDB9	CPDR	0
166	ED42	SBC HL, BC	0	216	EDBA	INDR	0
167	ED438405	X	0	217	EDBB	OTDR	0
168	ED44	NEG	0	218	FD09	ADD IV, BC	4
169	ED45	RETN	0	219	FD19	ADD IV, DE	4
170	ED46	IN 0	0	220	FD218405	X	6
171	ED47	LD I, A	0	221	FD228405	X	6
172	ED48	IN C, (C)	0	222	FD23	INC IV	4
173	ED49	OUT (C), C	0	223	FD29	ADD IV, IV	4
174	ED4A	ADC HL, BC	0	224	FD2A8405	X	6
175	ED4B8405	X	0	225	FD2B	DEC IV	4
176	ED4D	RETI	0	226	FD3405	INC (IV+	5
177	ED50	IN D, (C)	0	227	FD3505	DEC (IV+	5
178	ED51	OUT (C), D	0	228	FD368520	X	6
179	ED52	SBC HL, DE	0	229	FD39	ADD IV, SP	4
180	ED538405	X	0	230	FD4605	LD B, (IV+	5
181	ED56	IN 1	0	231	FD4E05	LD C, (IV+	5
182	ED57	LD A, I	0	232	FD5605	LD D, (IV+	5
183	ED58	IN E, (C)	0	233	FD5E05	LD E, (IV+	5
184	ED59	OUT (C), E	0	234	FD6605	LD H, (IV+	5
185	ED5A	ADC HL, DE	0	235	FD6E05	LD L, (IV+	5
186	ED5B8405	LD DE, (	0	236	FD7005	B	8
187	ED5E	IN 2	0	237	FD7105	C	8
188	ED60	IN H, (C)	0	238	FD7205	D	8
189	ED61	OUT (C), H	0	239	FD7305	E	8
190	ED62	SBC HL, HI	0	240	FD7405	H	8
191	ED67	RRO	0	241	FD7505	L	8
192	ED68	IN L, (C)	0	242	FD7705	A	8
193	ED69	OUT (C), L	0	243	FD7E05	LD A, (IV+	5
194	ED6A	ADC HL, HL	0	244	FD8605	ADD A, (IV+	5
195	ED6F	RLD	0	245	FD8E05	ADC A, (IV+	5
196	ED72	SBC HL, SP	0	246	FD9605	SUB (IV+	5
197	ED738405	X	0	247	FD9E05	SBC A, (IV+	5
198	ED78	IN A, (C)	0	248	FDA605	AND (IV+	5
199	ED79	OUT (C), A	0	249	FDAE05	XOR (IV+	5
200	ED7A	ADC HL, SP	0	250	FD8605	OR (IV+	5
				251	FD8E05	CP (IV+	5
				252	FDE1	POP IV	4
				253	FDE3	EX (SP), IV	4
				254	FDE5	PUSH IV	4
				255	FDE9	JP (IV)	4
				256	PDF9	LD SP, IV	4

FIGURA 4 (concluye)

El le

UCIO.

F/DEPFI/I-1/1980/EJ.4



702232

